Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

CY8C22x45
CY8C21345

PSoC® Programmable System-on-Chip™
Technical Reference Manual (TRM)

PSoC TRM, Spec. # 001-48461 Rev. *H

Cypress Semiconductor
An Infineon Technologies Company
198 Champion Court
San Jose, CA 95134-1709
www.cypress.com
www.infineon.com

Copyrights

© Cypress Semiconductor Corporation, 2008-2021. This document is the property of Cypress Semiconductor Corporation and its subsidiaries ("Cypress"). This document, including any software or firmware included or referenced in this document ("Software"), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress's patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress shall have no liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. CYPRESS DOES NOT REPRESENT, WARRANT, OR GUARANTEE THAT CYPRESS PRODUCTS, OR SYSTEMS CREATED USING CYPRESS PRODUCTS, WILL BE FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (collectively, "Security Breach"). Cypress disclaims any liability relating to any Security Breach, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any Security Breach. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. "High-Risk Device" means any device or system whose failure could cause personal injury, death, or property damage. Examples of High-Risk Devices are weapons, nuclear installations, surgical implants, and other medical devices. "Critical Component" means any component of a High-Risk Device whose failure to perform can be reasonably expected to cause, directly or indirectly, the failure of the High-Risk Device, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any use of a Cypress product as a Critical Component in a High-Risk Device. You shall indemnify and hold Cypress, its directors, officers, employees, agents, affiliates, distributors, and assigns harmless from and against all claims, costs, damages, and expenses, arising out of any claim, including claims for product liability, personal injury or death, or property damage arising from any use of a Cypress product as a Critical Component in a High-Risk Device. Cypress products are not intended or authorized for use as a Critical Component in any High-Risk Device except to the limited extent that (i) Cypress's published data sheet for the product explicitly states Cypress has qualified the product for use in a specific High-Risk Device, or (ii) Cypress has given you advance written authorization to use the product as a Critical Component in the specific High-Risk Device and you have signed a separate indemnification agreement.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com. Other names and brands may be claimed as property of their respective owners.

Contents Overview

Section A: Overview **18**
1. Pin Information 25

Section B: PSoC Core **29**
2. CPU Core (M8C) 33
3. Supervisory ROM (SROM) 43
4. RAM Paging 53
5. Interrupt Controller 60
6. General Purpose I/O (GPIO) 68
7. Internal Main Oscillator (IMO) 75
8. Internal Low Speed Oscillator (ILO) 79
9. External Crystal Oscillator (ECO) 80
10. Phase-Locked Loop (PLL) 85
11. Sleep and Watchdog 88

Section C: Register Reference **100**
12. Register Details 103

Section D: Digital System **268**
13. Global Digital Interconnect (GDI) 272
14. Array Digital Interconnect (ADI) 278
15. Row Digital Interconnect (RDI) 280
16. Digital Blocks 288

Section E: Analog System **334**
17. Two Column Limited Analog System 338
18. Two Column Analog Compare System 358

Section F: System Resources **373**
19. Digital Clocks 377
20. Multiply Accumulate (MAC) 387
21. I2C 392
22. Internal Voltage Reference 410
23. System Resets 412
24. POR and LVD 419
25. I/O Analog Multiplexer 422
26. CSD Logic System 429
27. Real Time Clock (RTC) 436
28. 10-Bit SAR ADC Controller 439
29. MISC Logic 446

Section G: Glossary **452**

Contents

Section A: Overview 18

Document Organization 18
Top-Level Architecture 19
PSoC Core 19
Digital System 19
Analog System 19
System Resources 19
PSoC Device Characteristics 21
Getting Started 21
Support 21
Product Upgrades 21
Development Kits 21
Document History 22
Documentation Conventions 23
Register Conventions 23
Numeric Naming 23
Units of Measure 23
Acronyms 23

1. Pin Information 25

1.1 Pinouts 25
1.1.1 28-Pin Part Pinout 26
1.1.2 44-Pin Part Pinout 27
1.1.3 56-Pin Part Pinout 28

Section B: PSoC Core 29

Top Level Core Architecture 29
Interpreting the Core Documentation 29
Core Register Summary 30

2. CPU Core (M8C) 33

2.1 Overview 33
2.2 Internal Registers 33
2.3 Address Spaces 33
2.4 Instruction Set Summary 34
2.5 Instruction Formats 36
2.5.1 One-Byte Instructions 36
2.5.2 Two-Byte Instructions 36
2.5.3 Three-Byte Instructions 37
2.6 Addressing Modes 37
2.6.1 Source Immediate 37
2.6.2 Source Direct 38

2.6.3 Source Indexed 38
2.6.4 Destination Direct 39
2.6.5 Destination Indexed 39
2.6.6 Destination Direct Source Immediate 39
2.6.7 Destination Indexed Source Immediate 40
2.6.8 Destination Direct Source Direct 40
2.6.9 Source Indirect Post Increment 41
2.6.10 Destination Indirect Post Increment 41
2.7 Register Definitions 42
2.7.1 CPU_F Register 42

3. Supervisory ROM (SROM) 43

3.1 Architectural Description 43
3.1.1 Additional SROM Feature 44
3.1.2 SROM Function Descriptions 44
3.1.2.1 SWBootReset Function 44
3.1.2.2 ReadBlock Function 45
3.1.2.3 WriteBlock Function 45
3.1.2.4 EraseBlock Function 46
3.1.2.5 ProtectBlock Function 46
3.1.2.6 TableRead Function 46
3.1.2.7 EraseAll Function 47
3.1.2.8 Checksum Function 47
3.1.2.9 Calibrate0 Function 47
3.1.2.10 Calibrate1 Function 47
3.2 PSoC Device Distinctions 48
3.3 Register Definitions 49
3.3.1 STK_PP Register 49
3.3.2 MVR_PP Register 49
3.3.3 MVW_PP Register 50
3.3.4 CPU_SCR1 Register 50
3.3.5 FLS_PR1 Register 51
3.4 Clocking 52
3.4.1 DELAY Parameter 52
3.4.2 CLOCK Parameter 52

4. RAM Paging 53

4.1 Architectural Description 53
4.1.1 Basic Paging 53
4.1.2 Stack Operations 54
4.1.3 Interrupts 54
4.1.4 MVI Instructions 54
4.1.5 Current Page Pointer 54
4.1.6 Index Memory Page Pointer 55
4.2 Register Definitions 56
4.2.1 TMP_DRx Registers 56
4.2.2 CUR_PP Register 56
4.2.3 STK_PP Register 57
4.2.4 IDX_PP Register 57
4.2.5 MVR_PP Register 58
4.2.6 MVW_PP Register 58
4.2.7 CPU_F Register 59

5. Interrupt Controller 60

5.1 Architectural Description 60
5.1.1 Posted versus Pending Interrupts 61
5.2 Application Description 62
5.3 Register Definitions 63
5.3.1 INT_CLRx Registers 63
5.3.1.1 INT_CLR0 Register 63
5.3.1.2 INT_CLR1 Register 63
5.3.1.3 INT_CLR2 Register 64
5.3.1.4 INT_CLR3 Register 64
5.3.2 INT_MSKx Registers 65
5.3.2.1 INT_MSK3 Register 65
5.3.2.2 INT_MSK2 Register 65
5.3.2.3 INT_MSK0 Register 65
5.3.2.4 INT_MSK1 Register 66
5.3.3 INT_VC Register 66
5.3.4 CPU_F Register 66

6. General Purpose I/O (GPIO) 68

6.1 Architectural Description 68
6.1.1 Digital I/O 68
6.1.2 Global I/O 69
6.1.3 Analog Input 70
6.1.4 GPIO Block Interrupts 71
6.2 Register Definitions 72
6.2.1 PRTxDR Registers 72
6.2.2 PRTxIE Registers 72
6.2.3 PRTxGS Registers 72
6.2.4 PRTxDMx Registers 73
6.2.5 PRTxICx Registers 74

7. Internal Main Oscillator (IMO) 75

7.1 Architectural Description 75
7.2 PSoC Device Distinctions 75
7.3 Application Description 76
7.3.1 Trimming the IMO 76
7.3.2 Engaging Slow IMO 76
7.4 Register Definitions 77
7.4.1 CPU_SCR1 Register 77
7.4.2 OSC_CR2 Register 78
7.4.3 IMO_TR Register 78

8. Internal Low Speed Oscillator (ILO) 79

8.1 Architectural Description 79
8.2 Register Definitions 79
8.2.1 ILO_TR Register 79

9. External Crystal Oscillator (ECO) 80

9.1 Architectural Description 80
9.1.1 ECO External Components 81
9.2 PSoC Device Distinctions 81

9.3 Register Definitions 82
9.3.1 CPU_SCR1 Register 82
9.3.2 OSC_CR0 Register 83
9.3.3 ECO_TR Register 84

10. Phase-Locked Loop (PLL) 85

10.1 Architectural Description 85
10.2 Register Definitions 85
10.2.1 OSC_CR0 Register 86
10.2.2 OSC_CR2 Register 87

11. Sleep and Watchdog 88

11.1 Architectural Description 88
11.1.1 32 kHz Clock Selection 88
11.1.2 Sleep Timer 88
11.2 Application Description 89
11.3 Register Definitions 90
11.3.1 INT_MSK0 Register 90
11.3.2 RES_WDT Register 90
11.3.3 CPU_SCR1 Register 91
11.3.4 CPU_SCR0 Register 92
11.3.5 OSC_CR0 Register 93
11.3.6 OSC_CR2 Register 94
11.3.7 ILO_TR Register 95
11.3.8 ECO_TR Register 95
11.4 Timing Diagrams 96
11.4.1 Sleep Sequence 96
11.4.2 Wake Up Sequence 97
11.4.3 Bandgap Refresh 98
11.4.4 Watchdog Timer 98
11.5 Power Consumption 99

Section C: Register Reference 100

Register General Conventions 100
Register Naming Conventions 100
Register Mapping Tables 100
Register Map Bank 0 Table: User Space 101
Register Map Bank 1 Table: Configuration Space 102

12. Register Details 103

12.1 Maneuvering Around the Registers 103
Register Conventions 103
12.1.1 Register Naming Conventions 103
12.2 Bank 0 Registers 105
12.2.1 PRTxDR 105
12.2.2 PRTxIE 106
12.2.3 PRTxGS 107
12.2.4 PRTxDM2 108
12.2.5 DxCxxDR0 109
12.2.6 DxCxxDR1 110
12.2.7 DxCxxDR2 111
12.2.8 DxCxxCR0 (Timer Control:000) 112

12.2.9 DxCxxCR0 (Counter Control:001) 113
12.2.10DxCxxCR0 (Dead Band Control:100) 114
12.2.11DxCxxCR0 (CRCPRS Control:010) 115
12.2.12DxCxxCR0 (IPWM Control:011) 116
12.2.13DCCxxCR0 (SPIM Control:0-110) 117
12.2.14DCCxxCR0 (SPIS Control:1-110) 118
12.2.15DxCxxCR0 (DSM Control:111) 119
12.2.16DCCxxCR0 (UART Transmitter Control) 120
12.2.17DCCxxCR0 (UART Receiver Control) 121
12.2.18CSDx_DR0_L, CSDx_DR0_H 122
12.2.19CSDx_DR1_L, CSDx_DR1_H 123
12.2.20CSDx_CNT_L, CSDx_CNT_H 124
12.2.21CSDx_CR0 125
12.2.22CSDx_CR1 126
12.2.23AMX_IN 127
12.2.24AMUX_CFG 128
12.2.25PWM_CR 129
12.2.26ARF_CR 130
12.2.27CMP_CR0 131
12.2.28CMP_CR1 132
12.2.29ADCx_CR 133
12.2.30ADC_DH 134
12.2.31ADC_DL 135
12.2.32TMP_DRx 136
12.2.33ACExxCR1 137
12.2.34ACExxCR2 138
12.2.35ASExxCR0 139
12.2.36RDIxRI 140
12.2.37RDIxSYN 141
12.2.38RDIxIS 142
12.2.39RDIxLT0 143
12.2.40RDIxLT1 144
12.2.41RDIxRO0 145
12.2.42RDIxRO1 146
12.2.43RDIxDSM 147
12.2.44PWMVREF0 148
12.2.45PWMVREF1 149
12.2.46IDAC_MODE 150
12.2.47PWMSRC 151
12.2.48TS_CR0 152
12.2.49TS_CMPH 153
12.2.50TS_CMPL 154
12.2.51TS_CR1 155
12.2.52CUR_PP 156
12.2.53STK_PP 157
12.2.54IDX_PP 158
12.2.55MVR_PP 159
12.2.56MVW_PP 160
12.2.57I2C_CFG 161
12.2.58I2C_SCR 162
12.2.59I2C_DR 164
12.2.60I2C_MSCR 165
12.2.61INT_CLR0 166

12.2.62INT_CLR1 168
12.2.63INT_CLR2 170
12.2.64INT_CLR3 171
12.2.65INT_MSK3 172
12.2.66INT_MSK2 173
12.2.67INT_MSK0 174
12.2.68INT_MSK1 175
12.2.69INT_VC 176
12.2.70RES_WDT 177
12.2.71DEC_CR0 178
12.2.72DEC_CR1 179
12.2.73MULx_X 180
12.2.74MULx_Y 181
12.2.75MULx_DH 182
12.2.76MULx_DL 183
12.2.77MACx_X/ACCx_DR1 184
12.2.78MACx_Y/ACCx_DR0 185
12.2.79MACx_CL0/ACCx_DR3 186
12.2.80MACx_CL1/ACCx_DR2 187
12.2.81CPU_F 188
12.2.82IDACx_D 189
12.2.83CPU_SCR1 190
12.2.84CPU_SCR0 191
12.3 Bank 1 Registers 192
12.3.1 PRTxDM0 192
12.3.2 PRTxDM1 193
12.3.3 PRTxIC0 194
12.3.4 PRTxIC1 195
12.3.5 DxCxxFN 196
12.3.6 DxCxxIN 198
12.3.7 DxCxxOU 200
12.3.8 DxCxxCR1 (Timer Control:000) 202
12.3.9 DxCxxCR1 (Counter Control:001) 203
12.3.10DxCxxCR1 (CRCPRS Control:010) 204
12.3.11DxCxxCR1 (IPWM Control:011) 205
12.3.12DxCxxCR1 (Dead Band Control:100) 206
12.3.13DxCxxCR1 (SPIM Control:0-110) 207
12.3.14DxCxxCR1 (SPIS Control:0-110) 208
12.3.15DxCxxCR1 (DSM Control:111) 209
12.3.16CMPxCR1 210
12.3.17CMPxCR2 211
12.3.18VDAC5xCR0 212
12.3.19CSCMPCR0 213
12.3.20CSCMPGOEN 214
12.3.21CSLUTCR0 215
12.3.22CMPCOLMUX 216
12.3.23CMPPWMCR 217
12.3.24CMPFLTCR 218
12.3.25CMPCLK1 219
12.3.26CMPCLK0 220
12.3.27CLK_CR0 221
12.3.28CLK_CR1 222
12.3.29ABF_CR0 223

12.3.30AMD_CR0 224
12.3.31CMP_GO_EN 225
12.3.32AMD_CR1 226
12.3.33ALT_CR0 227
12.3.34AMUX_CFG1 229
12.3.35CLK_CR3 230
12.3.36GDI_O_IN_CR 231
12.3.37GDI_E_IN_CR 232
12.3.38GDI_O_OU_CR 233
12.3.39GDI_E_OU_CR 234
12.3.40RTCH 235
12.3.41RTCM 236
12.3.42RTCS 237
12.3.43RTCCR 238
12.3.44ADC_CR0 239
12.3.45ADC_CR1 240
12.3.46ADC_CR2 241
12.3.47ADC_CR3TRIM 242
12.3.48ADC_CR4 243
12.3.49I2C_ADDR 244
12.3.50GDI_O_IN 245
12.3.51GDI_E_IN 246
12.3.52GDI_O_OU 247
12.3.53GDI_E_OU 248
12.3.54MUX_CRx 249
12.3.55IDAC_CR1 250
12.3.56OSC_GO_EN 251
12.3.57OSC_CR4 252
12.3.58OSC_CR3 253
12.3.59OSC_CR0 254
12.3.60OSC_CR1 255
12.3.61OSC_CR2 256
12.3.62VLT_CR 257
12.3.63VLT_CMP 258
12.3.64ADCx_TR 259
12.3.65VDAC_TRIM 260
12.3.66IMO_TR 261
12.3.67ILO_TR 262
12.3.68BDG_TR 263
12.3.69ECO_TR 264
12.3.70FLS_PR1 265
12.3.71IDAC_CR0 266

Section D: Digital System 268

Top-Level Digital Architecture 268
Interpreting the Digital Documentation 268
Digital Register Summary 269

13. Global Digital Interconnect (GDI) 272

13.1 Architectural Description 272
13.1.1 28-Pin Global Interconnect....272
13.1.2 44-Pin Global Interconnect....273

13.1.3 56-Pin Global Interconnect 274
13.2 Register Definitions 275
13.2.1 GDI_x_IN Registers/GDI_x_IN_CR Registers 275
13.2.2 GDI_x_OU/GDI_x_OU_CR Registers 276

14. Array Digital Interconnect (ADI) 278
14.1 Architectural Description 278

15. Row Digital Interconnect (RDI) 280
15.1 Architectural Description 280
15.2 PSoC Device Distinctions 281
15.3 Register Definitions 282
15.3.1 RDIxRI Register 282
15.3.2 RDIxSYN Register 283
15.3.3 RDIxIS Register 284
15.3.4 RDIxLTx Registers 285
15.3.5 RDIxROx Registers 286
15.3.5.1 RDIxRO0 Register 286
15.3.5.2 RDIxRO1 Register 286
15.3.6 RDIxDSM Register 287
15.4 Timing Diagram 287

16. Digital Blocks 288
16.1 Architectural Description 288
16.1.1 Input Multiplexers 289
16.1.2 Input Clock Resynchronization 289
16.1.2.1 Clock Resynchronization Summary 290
16.1.3 Output De-Multiplexers 290
16.1.4 Block Chaining Signals 290
16.1.5 Input Data Synchronization 290
16.1.6 Timer Function 290
16.1.6.1 Usability Exceptions 291
16.1.6.2 Block Interrupt 291
16.1.7 Counter Function 291
16.1.7.1 Counter Timing 291
16.1.7.2 Usability Exceptions 292
16.1.7.3 Block Interrupt 292
16.1.8 Dead Band Function 292
16.1.8.1 Usability Exceptions 293
16.1.8.2 Block Interrupt 293
16.1.9 IPWM Function 293
16.1.9.1 Usability Exceptions 294
16.1.9.2 Block Interrupt 294
16.1.10 CRCPRS Function 294
16.1.10.1 Usability Exceptions 295
16.1.10.2 Block Interrupt 295
16.1.11 SPI Protocol Function 295
16.1.11.1 SPI Protocol Signal Definitions 296
16.1.12 SPI Master Function 296
16.1.12.1 Usability Exceptions 297
16.1.12.2 Block Interrupt 297
16.1.13 SPI Slave Function 297

16.1.13.1 Usability Exceptions 297
16.1.13.2 Block Interrupt 297
16.1.14 Asynchronous Transmitter and Receiver Functions 298
16.1.14.1 Asynchronous Transmitter Function 298
16.1.14.2 Usability Exceptions 298
16.1.14.3 Block Interrupt 298
16.1.14.4 Asynchronous Receiver Function 298
16.1.14.5 Usability Exceptions 299
16.1.14.6 Block Interrupt 299
16.1.15 DSM function 299
16.1.15.1 Usability Exception 299
16.1.15.2 Block interrupt 299
16.2 Register Definitions 300
16.2.1 DxCxxDRx Registers 301
16.2.1.1 Timer Register Definitions 301
16.2.1.2 Counter Register Definitions 302
16.2.1.3 Dead Band Register Definitions 302
16.2.1.4 IPWM Register Definitions 303
16.2.1.5 CRCPRS Register Definitions 304
16.2.1.6 SPI Master Register Definitions 304
16.2.1.7 SPI Slave Register Definitions 305
16.2.1.8 Transmitter Register Definitions 305
16.2.1.9 Receiver Register Definitions 305
16.2.1.10 DSM Register Definitions 306
16.2.2 DxCxxCR0 Register 307
16.2.3 DxCxxCR1 Register 310
16.2.4 INT_MSK1 Register 313
16.2.5 DxCxxFN Registers 314
16.2.6 DxCxxIN Registers 315
16.2.7 DxCxxOU Registers 315
16.3 Timing Diagrams 317
16.3.1 Timer Timing 317
16.3.2 Counter Timing 320
16.3.3 Dead Band Timing 320
16.3.3.1 Changing the PWM Duty Cycle 321
16.3.3.2 Kill Operation 321
16.3.4 IPWM Timing 322
16.3.5 CRCPRS Timing 323
16.3.6 SPI Mode Timing 323
16.3.7 SPIM Timing 323
16.3.8 SPIS Timing 327
16.3.9 Transmitter Timing 329
16.3.10 Receiver Timing 331
16.3.11 DSM Timing 333

Section E: Analog System 334

Top-Level Analog Architecture 334
Interpreting the Analog Documentation 334
Application Description 335
Defining the Analog Blocks 335
Analog Functionality 335
Analog Register Summary 336

17. Two Column Limited Analog System 338
17.1 Architectural Description 338
17.1.1 Analog Interface 338
17.1.1.1 Analog Comparator Bus Interface 339
17.1.1.2 Analog Column Clock Generation 339
17.1.1.3 Single Slope ADC 339
17.1.1.4 PWM ADC Interface 341
17.1.1.5 Analog Modulator Interface (Mod Bits) 341
17.1.1.6 Sample and Hold Feature 341
17.1.2 Analog Array 342
17.1.2.1 NMux Connections 342
17.1.2.2 PMux Connections 343
17.1.2.3 Temperature Sensing Capability 343
17.1.3 Analog Input Configuration 343
17.1.4 Analog Reference 347
17.1.5 Continuous Time PSoC Block 347
17.1.6 Switched Capacitor PSoC Block 348
17.1.6.1 Application Description for the SC Block 348
17.2 PSoC Device Distinctions 348
17.3 Register Definitions 349
17.3.1 Summary Table for 2 Column Limited Analog System Registers 349
17.3.2 Analog Interface Registers 350
17.3.3 Analog Input Configuration Registers 355
17.3.4 Continuous Time PSoC Block Registers 356
17.3.5 Switched Capacitor PSoC Block Register 357

18. Two Column Analog Compare System 358
18.1 Architectural Description 358
18.1.1 Analog Interface 358
18.1.1.1 Analog Comparator Bus Interface 359
18.1.1.2 Analog Column Clock Generation 360
18.1.2 Analog Array 360
18.1.2.1 NMux Connections 361
18.1.2.2 PMux Connections 361
18.1.3 Analog Input Configuration 362
18.1.4 Analog Reference 364
18.1.5 Continuous Time PSoC Block 364
18.1.6 VDAC 365
18.2 PSoC Device Distinctions 365
18.3 Register Definitions 366
18.3.1 Summary Table for Analog System Registers 366
18.3.2 Analog Interface Registers 367
18.3.3 Analog Input Configuration Registers 370
18.3.4 Continuous Time PSoC Block Registers 371
18.3.5 VDAC5 Block Register 372

Section F: System Resources 373
Top-Level System Resources Architecture 373
Interpreting the System Resources Documentation 373
System Resources Register Summary 374

19. Digital Clocks 377

19.1 Architectural Description 377
19.1.1 Internal Main Oscillator 377
19.1.2 Internal Low Speed Oscillator 377
19.1.3 32.768 kHz Crystal Oscillator 379
19.1.4 External Clock 379
19.1.4.1 Clock Doubler 379
19.1.4.2 Switch Operation 379
19.2 PSoC Device Distinctions 380
19.3 Register Definitions 381
19.3.1 INT_CLR0 Register 381
19.3.2 INT_MSK0 Register 381
19.3.3 OSC_GO_EN Register 381
19.3.4 OSC_CR4 Register 382
19.3.5 OSC_CR3 Register 382
19.3.6 OSC_CR0 Register 384
19.3.7 OSC_CR1 Register 385
19.3.8 OSC_CR2 Register 386

20. Multiply Accumulate (MAC) 387

20.1 Architectural Description 387
20.2 Application Description 388
20.2.1 Multiplication with No Accumulation 388
20.2.2 Accumulation After Multiplication 388
20.3 Register Definitions 388
20.3.1 MULx_X Register 388
20.3.2 MULx_Y Register 389
20.3.3 MULx_DH Register 390
20.3.4 MULx_DL Register 390
20.3.5 MACx_X/ACCx_DR1 Register 390
20.3.6 MACx_Y/ACCx_DR0 Register 391
20.3.7 MACx_CL0/ACCx_DR3 Register 391
20.3.8 MACx_CL1/ACCx_DR2 Register 391

21. I2C 392

21.1 Architectural Description 392
21.1.1 Basic I2C Data Transfer 392
21.2 Application Description 393
21.2.1 Slave Operation 393
21.2.2 Master Operation 395
21.3 Register Definitions 396
21.3.1 I2C_ADDR Register 396
21.3.2 I2C_CFG Register 397
21.3.3 I2C_SCR Register 399
21.3.4 I2C_DR Register 401
21.3.5 I2C_MSCR Register 402
21.4 Timing Diagrams 404
21.4.1 Clock Generation 404
21.4.2 Basic Input/Output Timing 404
21.4.3 Status Timing 405
21.4.4 Master Start Timing 406
21.4.5 Master Restart Timing 407

21.4.6 Master Stop Timing 407
21.4.7 Master/Slave Stall Timing 408
21.4.8 Master Lost Arbitration Timing 408
21.4.9 Master Clock Synchronization....409

22. Internal Voltage Reference 410

22.1 Architectural Description....410
22.2 PSoC Device Distinctions....410
22.3 Register Definitions 410
22.3.1 BDG_TR Register 410

23. System Resets 412

23.1 Architectural Description....412
23.2 Pin Behavior During Reset 412
23.2.1 GPIO Behavior on Power Up 412
23.2.2 GPIO Behavior on External Reset 412
23.3 Register Definitions 414
23.3.1 CPU_SCR1 Register 414
23.3.2 CPU_SCR0 Register 415
23.4 Timing Diagrams 416
23.4.1 Power On Reset 416
23.4.2 External Reset 416
23.4.3 Watchdog Timer Reset 416
23.4.4 Reset Details....418
23.5 Power Consumption 418

24. POR and LVD 419

24.1 Architectural Description....419
24.2 PSoC Device Distinctions....419
24.3 Register Definitions 419
24.3.1 VLT_CR Register 420
24.3.2 VLT_CMP Register 421

25. I/O Analog Multiplexer 422

25.1 Architectural Description 422
25.1.1 IOMUX and GPIO 422
25.1.2 Dual Channel 8-Bit IDAC 423
25.2 PSoC Device Distinctions....423
25.3 Application Description 424
25.3.1 Capacitive Sensing 424
25.3.2 Chip-Wide Analog Input 425
25.3.3 Crosspoint Switch 425
25.3.4 Charging Current....425
25.4 Register Definitions 426
25.4.1 AMUX_CFG Register 426
25.4.2 IDACx_D Registers 426
25.4.3 AMUX_CFG1 Register 427
25.4.4 MUX_CRx Registers 427
25.4.5 IDAC_CR0 Register....428
25.4.6 IDAC_CR1 Register 428

26. CSD Logic System 429
26.1 Architectural Description 429
26.1.1 System Bus Interface 429
26.1.2 Local Data Bus DBDRV signal 429
26.1.3 CSD Clocks Generation 430
26.1.4 16-bit Timer/PRS Function 430
26.1.4.1 Timer Function 430
26.1.4.2 PRS Function 430
26.1.5 4-bit Multi-shot Function 430
26.1.6 16-bit Counter/Capture Function 430
26.1.6.1 Counter mode 430
26.1.6.2 Capture mode 430
26.1.7 Interrupt and Primary Output 430
26.2 Application Description 431
26.2.1 CSD Usage 431
26.2.2 General Timer Usage 432
26.3 Register Definitions 433
26.3.1 CSDx_DR0_L, CSDx_DR0_H 433
26.3.2 CSDx_DR1_L, CSDx_DR1_H 433
26.3.3 CSDx_CNT_L, CSDx_CNT_H 433
26.3.4 CSDx_CR0 434
26.3.5 CSDx_CR1 434
26.4 Clocking 434
26.5 Reuse Information 434
26.6 Symbol/Representation 435
26.7 Block Pin Description 435
26.8 Block Level Interfaces 435
26.8.1 CSD Logic Register Space Assignment 435
26.8.2 PRS related Block Level Interface Configuration 435
26.8.2.1 Seed[15:0] 435
26.8.2.2 DR0_O[15:0] and DR0_I[15:0] 435
26.8.2.3 SEL[7:0] and MASK_I[7:0] 435

27. Real Time Clock (RTC) 436
27.1 Architectural Description 436
27.1.1 BCD Code Counter 436
27.1.2 User Data Writing 436
27.1.3 RTC Data Reading 436
27.1.4 General Timer 437
27.2 Register Definitions 437
27.2.1 RTC_H 437
27.2.2 RTC_M 437
27.2.3 RTC_S 437
27.2.4 RTC_CR 438

28. 10-Bit SAR ADC Controller 439
28.1 Architectural Description 439
28.1.1 System Bus Interface and Local Data Bus DBDRV signal 439
28.1.2 ADC Clock Generation 439
28.1.3 Voltage Doubler Clock Generation 440
28.1.4 ADC FSM 440
28.1.5 SAR Algorithm and Data Process 440

28.1.6 A-D-C Operation Mode 440
28.1.7 ‘Ready’ Bit, ‘Ongoing’ Bit and Interrupt 441
28.2 Application Description 441
28.2.1 ADC Sample Rate and Clock Selection 441
28.2.2 Enable Voltage Doubler 441
28.3 Register Definitions 442
28.3.1 ADC_DH 442
28.3.2 ADC_DL 442
28.3.3 ADC_CR0 442
28.3.4 ADC_CR1 443
28.3.5 ADC_CR2 443
28.3.6 SARADC_CR3TRIM 443
28.3.7 SARADC_CR4 444
28.4 PSoC Device Distinctions 444
28.5 Clocking 444
28.6 Test Modes 445
28.6.1 Test Mode in ADC Comparator 445
28.6.2 Test Mode Bits in ADC Controller 445
28.6.3 Test for ADC Controller 445
28.6.4 ADC Controller Configuration for ADC Comparator Test 445
28.6.4.1 Run ADC clock and set D[13:0] 445
28.6.4.2 Do not run ADC clock and keep D[13:0] stable 445

29. MISC Logic 446

29.1 Architectural Description 446
29.1.1 System Bus Interface and Local Data Bus DBDRV signal 446
29.1.2 Vref PWM Signal Generator 446
29.1.3 IDAC ON/OFF Control 447
29.1.4 Dig-Blocks PWM Selection 447
29.1.5 Auto-Trig Selection 448
29.2 Register Definitions 449
29.2.1 PWMVREF0 449
29.2.2 PWMVREF1 449
29.2.3 IDACMODE 449
29.2.4 PWMSRC 450
29.2.5 TSCR0 450
29.2.6 TSCMPH 450
29.2.7 TSCMPL 450
29.2.8 TSCR1 451

Section G: Glossary 452

Section A: Overview

The PSoC® family consists of many Programmable System-on-Chip™ devices with on-chip controller devices. As described in this technical reference manual (TRM), a PSoC device includes configurable blocks of analog circuits and ***digital logic***, as well as programmable interconnect. This architecture allows the user to create customized peripheral configurations, to match the requirements of each individual application. Additionally, a fast CPU, Flash program memory, SRAM data memory, and configurable input/output (I/O) are included in a range of pinouts.

This document is a technical reference manual for all PSoCs with a base part number of CY8C2xxxx, except for the CY8C25122 and CY8C26xxx PSoC devices. It also applies to CY7C64215, CY7C603xx, and CYWUSB6953. To use this manual effectively, you must know how many digital rows and how many analog columns your PSoC device has (see the PSoC Device Characteristics table on page 21) and be aware of your PSoC device's distinctions (see the PSoC Device Distinctions on page 48). For the most up-to-date Ordering, Pinout, Packaging, or Electrical Specification information, refer to the individual PSoC device's data sheet. For the most current technical reference manual information, refer to the addendum. To obtain the newest product documentation, go to the Cypress web site at http://www.cypress.com/psoc. This section encompasses the following chapter:

- Pin Information on page 25

Document Organization

This manual is organized into sections and chapters, according to ***PSoC***® functionality. Each section begins with documentation interpretation, a top-level architectural explanation, PSoC device distinctions (if relevant), and a register summary (if applicable). Most chapters within the sections have an introduction, an architectural/application description, PSoC device distinctions (if relevant), register definitions, and timing diagrams. The sections are as follows:

- **Overview** – Presents the PSoC top-level architecture, PSoC device characteristics and distinctions, how to get started with helpful information, and document history and conventions. The PSoC device ***pinouts*** are detailed in the Pin Information chapter on page 25.
- **PSoC Core** – Describes the heart of the PSoC device in various chapters, beginning with an architectural overview and a summary list of registers pertaining to the PSoC core. See "PSoC Core" on page 29.
- **Register Reference** – Lists all PSoC device registers in Register Mapping Tables, on page 100, and presents bit-level detail of each PSoC register in its own Register Details chapter on page 103. Where applicable, detailed register descriptions are also located in each chapter.
- **Digital System** – Describes the configurable PSoC digital system in various chapters, beginning with an architectural overview and a summary list of registers pertaining to the digital system. See the "Digital System" on page 268.
- **Analog System** – Describes the configurable PSoC analog system in various chapters, beginning with an architectural overview and a summary list of registers pertaining to the analog system. See the "Analog System" on page 334.
- **System Resources** – Presents additional PSoC system resources, depending on the PSoC device, beginning with an overview and a summary list of registers pertaining to system resources. See "System Resources" on page 373.
- **Glossary** – Defines the specialized terminology used in this manual. Glossary terms are presented in ***bold, italic font*** throughout this manual. See the "Glossary" on page 452.
- **Index** – Lists the location of key topics and elements that constitute and empower the PSoC device. See the "Index" in the TRM.

Top-Level Architecture

The PSoC block diagram on the next page illustrates the top-level architecture of the family of PSoC devices. Each major grouping in the diagram is covered in this manual in its own section: PSoC Core, Digital System, Analog System, and the System Resources. Banding these four main areas together is the communication network of the system ***bus***.

PSoC Core

The PSoC Core is a powerful engine that supports a rich instruction set. It encompasses the ***SRAM*** for data storage, an ***interrupt*** controller for easy program execution to new addresses, sleep and watchdog timers, and multiple ***clock*** sources that include the phase locked loop (PLL), IMO (internal main oscillator), ILO (internal low speed oscillator), and ECO (32.768 kHz external crystal oscillator) for precision, programmable clocking. The clocks, together with programmable clock dividers (as a System Resource), provide the flexibility to integrate almost any timing requirement into the PSoC device.

The CPU core, called the M8C, is a powerful processor with speeds up to 24 MHz. The M8C is a four MIPS 8-***bit*** Harvard architecture microprocessor. Within the CPU core are the ***SROM*** and ***Flash*** memory components that provide flexible programming. The smallest PSoC devices have a slightly different analog configuration.

PSoC GPIOs provide connection to the CPU, digital and analog resources of the device. Each pin's drive mode may be selected from eight options, allowing great flexibility in external interfacing. Every pin also has the capability to generate a system interrupt on high level, low level, and change from last read.

Digital System

The Digital System is composed of digital rows in a block ***array***, and the Global, Array, and Row Digital Interconnects (GDI, ADI, and RDI, respectively). Digital blocks are provided in two rows of four blocks. This allows you the optimum choice of system resources for your application.

The digital blocks can be connected to any GPIO through a series of global buses that can route any signal to any pin. The buses also allow for signal multiplexing and for performing logic operations. This configurability frees your designs from the constraints of a fixed peripheral controller.

Analog System

The Analog System is composed of analog columns in a block array, analog references, analog ***input*** muxing, and analog drivers. The analog system block is composed of up to four analog columns with up to six analog blocks. Each configurable block is comprised of an opamp circuit allowing the creation of complex analog signal flows.

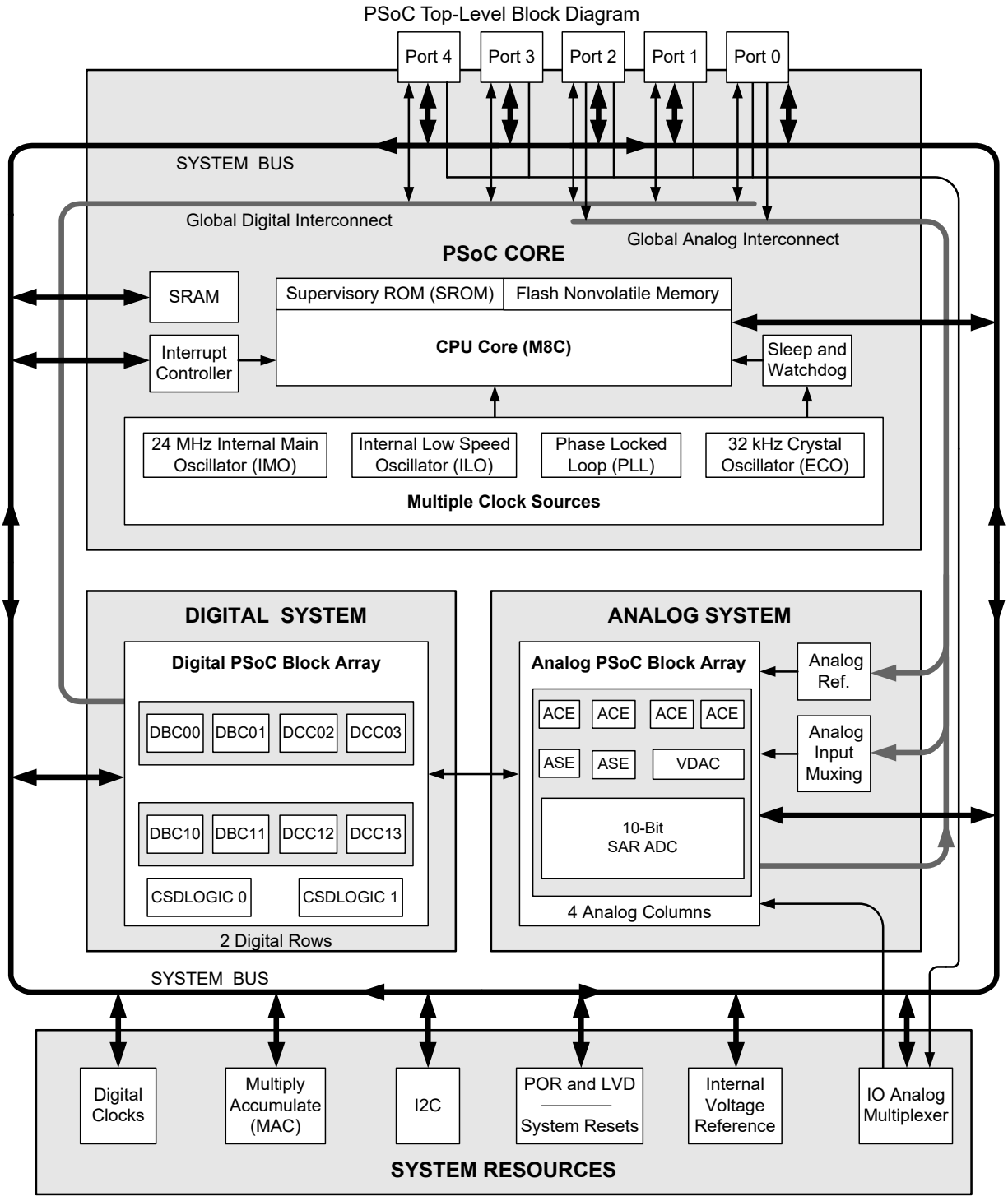
Each analog column contains one Continuous Time (CT) block, Type E (ACE); one Switched Capacitor (SC) block, Type E (ASE); or a VDAC.

System Resources

The System Resources provide additional PSoC capability, depending on the features of your PSoC device (see the table titled "Availability of System Resources for PSoC Devices" on page 21). These system resources include:

- Digital clocks to increase the flexibility of the PSoC ***mixed-signal*** arrays.
- One multiply accumulate (MAC) that provides a fast 8-bit multiplier or a fast 8-bit multiplier with 32-bit accumulate.
- ***I2C*** functionality for implementing either I2C slave or master.
- An internal voltage reference that provides an absolute value of 1.3 V to a variety of PSoC subsystems.
- An enhanced analog multiplexer (mux) that allows every I/O pin to connect to a common internal analog mux bus.
- Various system resets supported by the M8C.

PSoC Top-Level Block Diagram



PSoC Device Characteristics

There are many chip groups in the PSoC Programmable System-on-Chip Family. Besides differentiating these groups by way of PSoC part numbers, each PSoC group is easily distinguished by the unique number of digital rows and analog columns it has. This unique characteristic is the foundation for how this manual presents information.

The ***digital*** system can have 2 or 1 digital rows. The ***analog*** system can have 4 analog columns. Each PSoC device has a unique combination of digital rows and analog columns. The following table lists the resources available for specific PSoC device groups. Remember the particular PSoC device characteristics when referencing its functionality in this manual.

PSoC Device Characteristics

PSoC Device Group	Digital I/O (max)	Digital Rows	Digital Blocks	Analog Inputs	Analog Outputs	Analog Columns	Analog Blocks	Amount of SRAM	Amount of Flash
CY8C22x45	38	2	8	10	0	4	6	1 KB	16 KB
CY8C21345	24	1	4	10	0	4	6	512 Bytes	8 KB

Getting Started

The quickest path to understanding PSoC is by reading the PSoC device's data sheet and using the *PSoC Designer Integrated Development Environment (IDE)*. This manual is useful for understanding the details of the PSoC integrated circuit.

Important Note For the most up-to-date Ordering, Packaging, or Electrical Specification information, refer to the individual PSoC device's data sheet or go to http://www.cypress.com/psoc.

Support

Free support for PSoC products is available online at http://www.cypress.com. Resources include Training Seminars, Discussion Forums, Application Notes, PSoC Consultants, TightLink Technical Support Email/Knowledge Base, and Application Support Technicians.

Technical Support can be reached at http://www.cypress.com/support/ or can be contacted by phone at:
1-800-541-4736.

Product Upgrades

Cypress provides scheduled upgrades and version enhancements for PSoC Designer free of charge. You can order the upgrades from your distributor on CD-ROM or download them directly from http://www.cypress.com under the Software tab. Also provided are critical updates to system documentation under the Documentation tab.

Development Kits

Development Kits are available from the following distributors: Digi-Key, Avnet, Arrow, and Future. The Cypress Online Store contains development kits, ***C*** compilers, and all accessories for PSoC development. Go to the Cypress Online Store web site at http://www.cypress.com/shop/. Under Product Categories click *PSoC (Programmable System-on-Chip)* to view a current list of available items.

Document History

This section serves as a chronicle of the *PSoC® Programmable System-on-Chip™ Technical Reference Manual*.

PSoC Technical Reference Manual History

Version/ Release Date	Description of Change
** August 2008	First release of the *PSoC® Programmable System-on-Chip™ Technical Reference Manual*. This release encompasses the following PSoC devices: CY8C22x45 and CY8C21345.
*A December 2008	Overall updates for Product Release.
*B July 2011	Corrected register names: DAC_CR0 to IDAC_CR0 and DAC_CR1 to IDAC_CR1. Updated Register (CMPxCR1; 1,50h) description table.
*C June 2012	Updated IDAC_CR entries in the index
*D July 2012	Updated sections 12.2.82 IDACx_D and 25.4.2 IDACx_D Registers to state that the IDACL_D register is available in both CY8C21x45 and CY8C22x45 PSoC devices.
*E July 2013	Removed reference to IMODIS bit
*F June 2014	Updated the “Current Range” column on page 251, page 267, and page 433
*G October 2016	Updated to new template. Completing Sunset Review.
*H January 05, 2021	Updated Table 3-5. Modified “Checksum Function” on page 47. Added information about “no glitch protection” in “External Clock” on page 379. Updated logo and copyright information.

Documentation Conventions

There are only four distinguishing font types used in this manual, besides those found in the headings.

- The first is the use of *italics* when referencing a document title or file name.
- The second is the use of ***bold italics*** when referencing a term described in the Glossary of this manual.
- The third is the use of Times New Roman font, distinguishing equation examples.
- The fourth is the use of `Courier New` font, distinguishing code examples.

Register Conventions

The following table lists the register conventions that are specific to this manual. A detailed set of register conventions is located in the Register Details chapter on page 103.

Register Conventions

Convention	Example	Description
'x' in a register name	ACExxCR1	Multiple instances/address ranges of the same register
R	R : 00	Read register or bit(s)
W	W : 00	Write register or bit(s)
L	RL : 00	Logical register or bit(s)
C	RC : 00	Clearable register or bit(s)
00	RW : 00	Reset value is 0x00 or 00h
XX	RW : XX	Register is not reset
0,	0,04h	Register is in bank 0
1,	1,23h	Register is in bank 1
x,	x,F7h	Register exists in register bank 0 and register bank 1
Empty, grayed-out table cell		Reserved bit or group of bits, unless otherwise stated

Numeric Naming

Hexadecimal numbers are represented with all letters in uppercase with an appended lowercase 'h' (for example, '14h' or '3Ah') and ***hexadecimal*** numbers may also be represented by a '0x' prefix, the ***C*** coding convention. Binary numbers have an appended lowercase 'b' (for example, 01010100b' or '01000011b'). Numbers not indicated by an 'h' or 'b' are ***decimal***.

Units of Measure

The following table lists units of measure used herein.

Units of Measure

Symbol	Unit of Measure
dB	decibels
Hz	hertz
k	kilo, 1000
K	2^{10}, 1024
KB	1024 bytes
Kbit	1024 bits
kHz	kilohertz (32.000)
MHz	megahertz
µA	microampere
µF	microfarad
µs	microsecond
µV	microvolts
mA	milli-ampere
ms	milli-second
mV	milli-volts
ns	nanosecond
pF	picofarad
ppm	parts per million
V	volts

Acronyms

The following table lists the acronyms that are used in this manual.

Acronyms

Acronym	Description
ABUS	analog output bus
AC	alternating current
ADC	analog-to-digital converter
API	Application Programming Interface
BC	broadcast clock
BR	bit rate
BRA	bus request acknowledge
BRQ	bus request
CBUS	comparator bus
CI	carry in
CMP	compare
CO	carry out
CPU	central processing unit
CRC	cyclic redundancy check
CT	continuous time
DAC	digital-to-analog converter
DC	direct current
DI	digital or data input

Acronyms *(continued)*

Acronym	Description
DMA	direct memory access
DO	digital or data output
ECO	external crystal oscillator
FB	feedback
GIE	global interrupt enable
GPIO	general purpose I/O
ICE	in-circuit emulator
IDE	integrated development environment
ILO	internal low speed oscillator
IMO	internal main oscillator
I/O	input/output
IOR	I/O read
IOW	I/O write
IPOR	imprecise power on reset
IRQ	interrupt request
ISR	interrupt service routine
ISSP	in-system serial programming
IVR	interrupt vector read
LFSR	linear feedback shift register
LRb	last received bit
LRB	last received byte
LSb	least significant bit
LSB	least significant byte
LUT	look-up table
MISO	master-in-slave-out
MOSI	master-out-slave-in
MSb	most significant bit
MSB	most significant byte
PC	program counter
PCH	program counter high
PCL	program counter low
PD	power down
PMA	PSoC memory arbiter
POR	power on reset
PPOR	precision power on reset
PRS	pseudo random sequence
PSoC®	Programmable System-on-Chip™
PSSDC	power system sleep duty cycle
PWM	pulse width modulator
RAM	random access memory
RETI	return from interrupt
RI	row input
RO	row output
ROM	read only memory
RW	read/write
SAR	successive approximation register
SC	switched capacitor
SIE	serial interface engine
SE0	single-ended zero
SOF	start of frame
SP	stack pointer

Acronyms *(continued)*

Acronym	Description
SPI	serial peripheral interconnect
SPIM	serial peripheral interconnect master
SPIS	serial peripheral interconnect slave
SRAM	static random access memory
SROM	supervisory read only memory
SSADC	single slope ADC
SSC	supervisory system call
TC	terminal count
USB	universal serial bus
WDT	watchdog timer
WDR	watchdog reset
XRES	external reset

1. Pin Information

This chapter lists, describes, and illustrates PSoC device pins and pinout configurations. For up-to-date Ordering, Pinout, and Packaging information, refer to the individual PSoC device's data sheet or go to: http://www.cypress.com/psoc.

1.1 Pinouts

The PSoC devices are available in a variety of packages. Refer to the following information for details on individual devices. Every port pin (labeled with a “P”), except for Vss, Vdd, and XRES in the following tables and illustrations, is capable of Digital I/O and Analog Mux Bus (L or R).

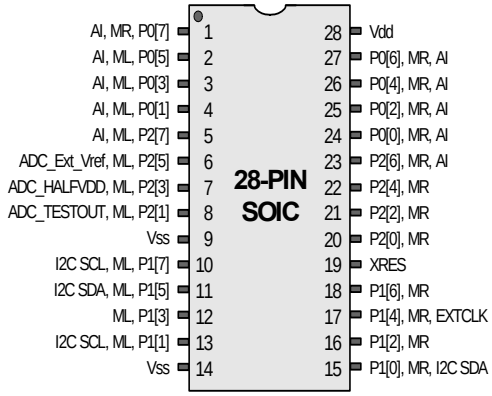
1.1.1 28-Pin Part Pinout

Table 1-1. 28-Pin Part Pinout (SOIC)

Pin No.	All Devices			
	Digital	Analog	Name	Description
1	I/O	I, MR	P0[7]	Integration Cap for MR
2	I/O	I, ML	P0[5]	Integration Cap for ML
3	I/O	I, ML	P0[3]	
4	I/O	I, ML	P0[1]	
5	I/O	I, ML	P2[7]	To Compare Column 0
6	I/O	ML	P2[5]	Optional ADC External Vref
7	I/O	ML	P2[3]	ADC Half Vdd Input (TM only)
8	I/O	ML	P2[1]	ADC Test Out (TM only)
9	Power		Vss	Ground connection.
10	I/O	ML	P1[7]	I2C Serial Clock (SCL)
11	I/O	ML	P1[5]	I2C Serial Data (SDA)
12	I/O	ML	P1[3]	
13	I/O	ML	P1[1]*	I2C Serial Clock (SCL), ISSP-SCLK
14	Power		Vss	Ground connection.
15	I/O	MR	P1[0]*	I2C Serial Clock (SCL), ISSP-SDATA
16	I/O	MR	P1[2]	
17	I/O	MR	P1[4]	Optional External Clock Input (EXTCLK)
18	I/O	MR	P1[6]	
19	Input		XRES	Active high pin reset with internal pull down.
20	I/O	MR	P2[0]	
21	I/O	MR	P2[2]	
22	I/O	MR	P2[4]	
23	I/O	I, MR	P2[6]	To Compare Column 1
24	I/O	I, MR	P0[0]	
25	I/O	I, MR	P0[2]	
26	I/O	I, MR	P0[4]	
27	I/O	I, MR	P0[6]	
28	Power		Vdd	Supply voltage.

LEGEND I = Input, O = Output, ML = Analog Mux Left Input,
MR = Analog Mux Right Input
* ISSP pin which is not High Z at POR.

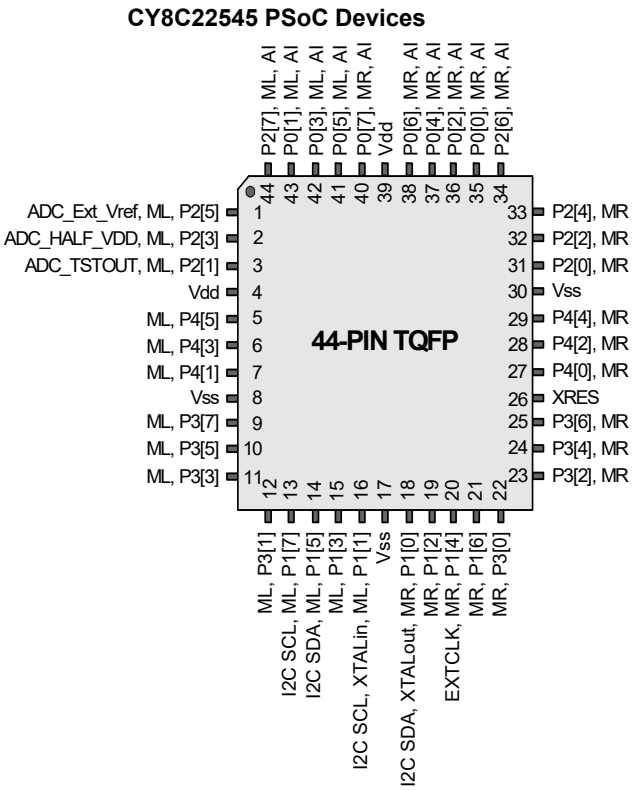
CY8C22345 and CY8C21345 PSoC Devices



1.1.2 44-Pin Part Pinout

Table 1-2. 44-Pin Part Pinout (TQFP)

Pin No.	Digital	Analog	Name	Description
1	I/O	ML	P2[5]	Optional ADC ExternalVref
2	I/O	ML	P2[3]	ADC Half Vdd Input (TM only)
3	I/O	ML	P2[1]	ADC TESTOUT (TM only)
4	Power		Vdd	Supply voltage
5	I/O	ML	P4[5]	
6	I/O	ML	P4[3]	
7	I/O	ML	P4[1]	
8	Power		Vss	Ground connection.
9	I/O	ML	P3[7]	
10	I/O	ML	P3[5]	
11	I/O	ML	P3[3]	
12	I/O	ML	P3[1]	
13	I/O	ML	P1[7]	I2C Serial Clock (SCL)
14	I/O	ML	P1[5]	I2C Serial Data (SDA)
15	I/O	ML	P1[3]	
16	I/O	ML	P1[1]*	Crystal (XTALin), I2C Serial Clock (SCL)
17	Power		Vss	Ground connection.
18	I/O	MR	P1[0]*	Crystal (XTALout), I2C Serial Data (SDA)
19	I/O	MR	P1[2]	
20	I/O	MR	P1[4]	Optional External Clock Input (EXT-CLK)
21	I/O	MR	P1[6]	
22	I/O	MR	P3[0]	
23	I/O	MR	P3[2]	
24	I/O	MR	P3[4]	
25	I/O	MR	P3[6]	
26	Input		XRES	Active high pin reset with internal pull down.
27	I/O	MR	P4[0]	
28	I/O	MR	P4[2]	
29	I/O	MR	P4[4]	
30	I/O	MR	Vss	Ground connection.
31	I/O	MR	P2[0]	
32	I/O	MR	P2[2]	
33	I/O	MR	P2[4]	
34	I/O	I, MR	P2[6]	To Compare Column 1
35	I/O	I, MR	P0[0]	
36	I/O	I, MR	P0[2]	



Pin No.	Digital	Analog	Name	Description
37	I/O	I, MR	P0[4]	
38	I/O	I, MR	P0[6]	
39	Power		Vdd	Supply voltage.
40	I/O	I, MR	P0[7]	Integration Cap for MR
41	I/O	I, ML	P0[5]	Integration Cap for ML
42	I/O	I, ML	P0[3]	
43	I/O	I, ML	P0[1]	
44	I/O	I, ML	P2[7]	To Compare Column 0

LEGEND A = Analog, I = Input, O = Output, TC/TM: Test, ML = Analog Mux Left Input, MR = Analog Mux Right Input.
* ISSP pin which is not High Z at POR.

1.1.3 56-Pin Part Pinout

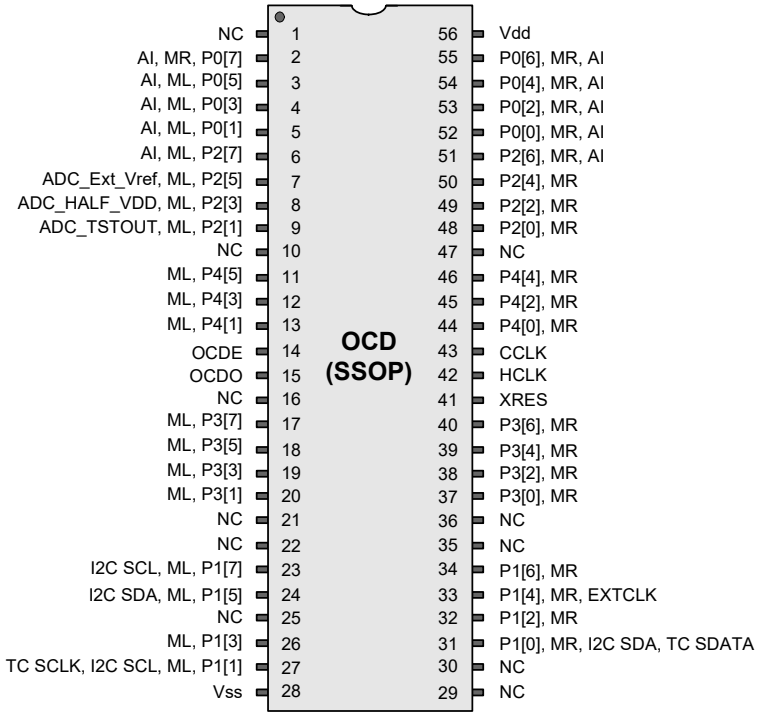
The 56-pin OCD part for the PSoC devices is shown below. Note that the CY8C21345 PSoC device uses the CY8C22x45 PSoC device for OCD.

Note OCD parts are only used for in-circuit debugging. OCD parts are NOT available for production.

Table 1-3. 56-Pin OCD Part Pinout (SSOP)

Pin No.	Name	Description
1	NC	No internal connection.
2	P0[7]	AI, MR, Integration Cap For MR
3	P0[5]	AI, ML, Integration Cap For ML
4	P0[3]	AI, ML
5	P0[1]	AI, ML
6	P2[7]	AI to Compare Column 0, ML
7	P2[5]	Optional ADC External Vref, ML
8	P2[3]	ADC Half Vdd Input (TM only), ML
9	P2[1]	ADC TEST OUT (TM only), ML
10	NC	No internal connection.
11	P4[5]	ML
12	P4[3]	ML
13	P4[1]	ML
14	OCDE	OCD even data I/O
15	OCDO	OCD odd data output
16	NC	No internal connection.
17	P3[7]	ML
18	P3[5]	ML
19	P3[3]	ML
20	P3[1]	ML
21	NC	No internal connection.
22	NC	No internal connection.
23	P1[7]	I2C Serial Clock (SCL), ML
24	P1[5]	I2C Serial Data (SDA), ML
25	NC	No internal connection.
26	P1[3]	ML
27	P1[1]*	I2C Serial Clock (SCL), ML
28	Vss	Ground connection.
29	NC	No internal connection.
30	NC	No internal connection.
31	P1[0]*	I2C Serial Data (SDA), MR
32	P1[2]	MR
33	P1[4]	Optional External Clock Input (EXTCLK), MR
34	P1[6]	MR
35	NC	No internal connection.
36	NC	No internal connection.
37	P3[0]	MR
38	P3[2]	MR
39	P3[4]	MR
40	P3[6]	MR
41	XRES	Active high pin reset with internal pull down.
42	HCLK	OCD high speed clock output
43	CCLK	OCD CPU clock output
44	P4[0]	MR
45	P4[2]	MR
46	P4[4]	MR
47	NC	No internal connection.
48	P2[0]	MR
49	P2[2]	MR
50	P2[4]	MR
51	P2[6]	AI to Compare Column 1, MR
52	P0[0]	AI, MR
53	P0[2]	AI, MR
54	P0[4]	AI, MR
55	P0[6]	AI, MR
56	Vdd	Supply Power.

CY8C22x45 and CY8C21345 OCD PSoC Devices



NOT FOR PRODUCTION

LEGEND A = Analog, I = Input, O = Output, OCD: On-chip Debugger, TC/TM: Test.
* IISSP pin which is not High Z at POR.

Section B: PSoC Core

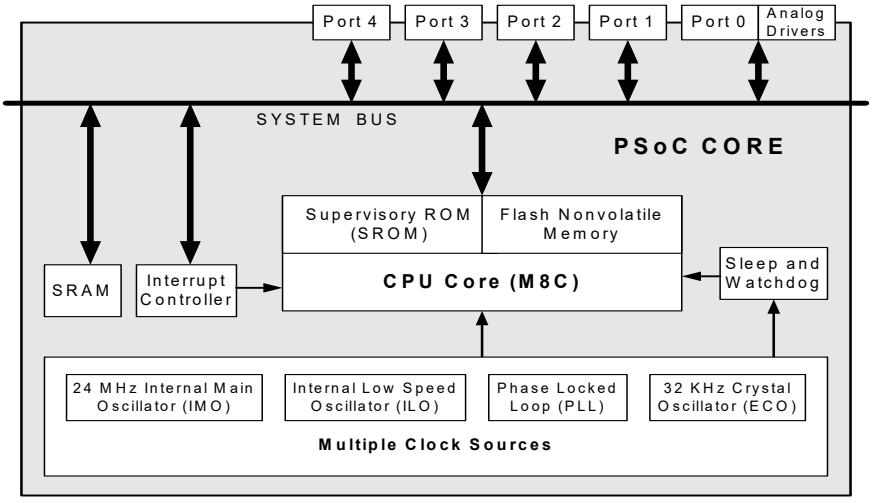
The PSoC Core section discusses the core components of a PSoC device with a base part number of CY8C22x45 and CY8C21345. This section contains these chapters:

- CPU Core (M8C) on page 33
- Supervisory ROM (SROM) on page 43
- RAM Paging on page 53
- Interrupt Controller on page 60
- General Purpose I/O (GPIO) on page 68
- Internal Main Oscillator (IMO) on page 75
- Internal Low Speed Oscillator (ILO) on page 79
- External Crystal Oscillator (ECO) on page 80
- Phase-Locked Loop (PLL) on page 85
- Sleep and Watchdog on page 88

Top Level Core Architecture

The figure below displays the top-level architecture of the PSoC's core. Each component of the figure is discussed at length in this section.

PSoC Core Block Diagram



Interpreting the Core Documentation

The core section covers the heart of the PSoC device which includes the M8C ***microcontroller***; SROM, interrupt controller, GPIO, analog output drivers, and ***SRAM*** paging; multiple clock sources such as IMO, ILO, ECO, and PLL; and sleep and watchdog functionality.

The ***analog output*** drivers are described in this section and not the Analog System section because they are part of the PSoC core input and ***output*** signals.

Core Register Summary

The table below lists all the PSoC registers for the CPU core in ***address*** order within their system resource configuration. The bits that are grayed out are reserved bits. If these bits are written, they should always be written with a value of '0'. For the core registers, the first 'x' in some ***register*** addresses represents either bank 0 or bank 1. These registers are listed throughout this manual in bank 0, even though they are also available in bank 1.

Note that all PSoC devices have a combination of 4, 2, or 1 analog columns and 4, 2 or 1 digital rows. The registers that are specifically constrained by the number of analog columns have the number of analog columns (Cols.) listed within the Address column of the table. The registers specifically pertaining to digital rows have the number of rows (Rows) listed within the Address column of the table. To determine the number of analog columns and digital rows in your PSoC device, refer to the table titled "PSoC Device Characteristics" on page 21.

Summary Table of the Core Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
M8C REGISTER (page 42)										
x,F7h	CPU_F	PgMode[1:0]			XIO		Carry	Zero	GIE	RL : 02
SUPERVISORY ROM (SROM) REGISTERS (page 49)										
0,D1h	STK_PP						Page Bits[2:0]			RW : 00
0,D4h	MVR_PP						Page Bits[2:0]			RW : 00
0,D5h	MVW_PP						Page Bits[2:0]			RW : 00
x,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW	ECO EX		IRAMDIS	# : 00
1,FAh	FLS_PR1							Bank[1:0]		RW:00
RAM PAGING (SRAM) REGISTERS (page 56)										
x,6Ch	TMP_DR0	Data[7:0]								RW : 00
x,6Dh	TMP_DR1	Data[7:0]								RW : 00
x,6Eh	TMP_DR2	Data[7:0]								RW : 00
x,6Fh	TMP_DR3	Data[7:0]								RW : 00
0,D0h	CUR_PP						Page Bits[2:0]			RW : 00
0,D1h	STK_PP						Page Bits[2:0]			RW : 00
0,D3h	IDX_PP						Page Bits[2:0]			RW : 00
0,D4h	MVR_PP						Page Bits[2:0]			RW : 00
0,D5h	MVW_PP						Page Bits[2:0]			RW : 00
x,F7h	CPU_F	PgMode[1:0]			XIO		Carry	Zero	GIE	RL : 02
INTERRUPT CONTROLLER REGISTERS (page 63)										
0,DAh 4 Cols.	INT_CLR0	VC3	Sleep	GPIO	Compare 1	Compare 0	Analog 1	Analog 0	V Monitor	RW : 00
0,DBh 2 Rows	INT_CLR1	DCC13	DCC12	DBC11	DBC10	DCC03	DCC02	DBC01	DBC00	RW : 00
1 Row						DCC03	DCC02	DBC01	DBC00	
0,DCh	INT_CLR2	RTC	CSD1	CSD0	SARADC					RW : 00
0,DDh	INT_CLR3								I2C	RW : 00
0,DEh	INT_MSK3	ENSWINT							I2C	RW : 00
0,DFh	INT_MSK2	RTC	CSD1	CSD0	SARADC					RW : 00
0,E0h 4 Cols.	INT_MSK0	VC3	Sleep	GPIO	Compare 1	Compare 0	Analog 1	Analog 0	V Monitor	RW : 00
0,E1h 2 Rows	INT_MSK1	DCC13	DCC12	DBC11	DBC10	DCC03	DCC02	DBC01	DBC00	RW : 00
1 Row						DCC03	DCC02	DBC01	DBC00	
0,E2h	INT_VC	Pending Interrupt[7:0]								RC : 00
x,F7h	CPU_F	PgMode[1:0]			XIO		Carry	Zero	GIE	RL : 02
GENERAL PURPOSE I/O (GPIO) REGISTERS (page 72)										
0,00h	PRT0DR	Data[7:0]								RW : 00
0,01h	PRT0IE	Interrupt Enables[7:0]								RW : 00
0,02h	PRT0GS	Global Select[7:0]								RW : 00
0,03h	PRT0DM2	Drive Mode 2[7:0]								RW : FF
1,00h	PRT0DM0	Drive Mode 0[7:0]								RW : 00

Summary Table of the Core Registers *(continued)*

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,01h	PRT0DM1	Drive Mode 1[7:0]								RW : FF
1,02h	PRT0IC0	Interrupt Control 0[7:0]								RW : 00
1,03h	PRT0IC1	Interrupt Control 1[7:0]								RW : 00
0,04h	PRT1DR	Data[7:0]								RW : 00
0,05h	PRT1IE	Interrupt Enables[7:0]								RW : 00
0,06h	PRT1GS	Global Select[7:0]								RW : 00
0,07h	PRT1DM2	Drive Mode 2[7:0]								RW : FF
1,04h	PRT1DM0	Drive Mode 0[7:0]								RW : 00
1,05h	PRT1DM1	Drive Mode 1[7:0]								RW : FF
1,06h	PRT1IC0	Interrupt Control 0[7:0]								RW : 00
1,07h	PRT1IC1	Interrupt Control 1[7:0]								RW : 00
0,08h	PRT2DR	Data[7:0]								RW : 00
0,09h	PRT2IE	Interrupt Enables[7:0]								RW : 00
0,0Ah	PRT2GS	Global Select[7:0]								RW : 00
0,0Bh	PRT2DM2	Drive Mode 2[7:0]								RW : FF
1,08h	PRT2DM0	Drive Mode 0[7:0]								RW : 00
1,09h	PRT2DM1	Drive Mode 1[7:0]								RW : FF
1,0Ah	PRT2IC0	Interrupt Control 0[7:0]								RW : 00
1,0Bh	PRT2IC1	Interrupt Control 1[7:0]								RW : 00
0,0Ch	PRT3DR	Data[7:0]								RW : 00
0,0Dh	PRT3IE	Interrupt Enables[7:0]								RW : 00
0,0Eh	PRT3GS	Global Select[7:0]								RW : 00
0,0Fh	PRT3DM2	Drive Mode 2[7:0]								RW : FF
1,0Ch	PRT3DM0	Drive Mode 0[7:0]								RW : 00
1,0Dh	PRT3DM1	Drive Mode 1[7:0]								RW : FF
1,0Eh	PRT3IC0	Interrupt Control 0[7:0]								RW : 00
1,0Fh	PRT3IC1	Interrupt Control 1[7:0]								RW : 00
0,10h	PRT4DR	Data[7:0]								RW : 00
0,11h	PRT4IE	Interrupt Enables[7:0]								RW : 00
0,12h	PRT4GS	Global Select[7:0]								RW : 00
0,13h	PRT4DM2	Drive Mode 2[7:0]								RW : FF
1,10h	PRT4DM0	Drive Mode 0[7:0]								RW : 00
1,11h	PRT4DM1	Drive Mode 1[7:0]								RW : FF
1,12h	PRT4IC0	Interrupt Control 0[7:0]								RW : 00
1,13h	PRT4IC1	Interrupt Control 1[7:0]								RW : 00
INTERNAL MAIN OSCILLATOR (IMO) REGISTERS (page 77)										
x,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW	ECO EX		IRAMDIS	# : 00
1,E2h	OSC_CR2	PLLGAIN			SLP_EXTEND	WDR32_SE	EXTCLKEN	RSVD	SYSCLKX2 DIS	RW : 00
1,E8h	IMO_TR	Trim[7:0]								W : 00
INTERNAL LOW SPEED OSCILLATOR (ILO) REGISTER (page 79)										
1,E9h	ILO_TR			Bias Trim[1:0]		Freq Trim[3:0]				W : 00
EXTERNAL CRYSTAL OSCILLATOR (ECO) REGISTERS (page 82)										
x,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW	ECO EX		IRAMDIS	# : 00
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00
1,EBh	ECO_TR	PSSDC[1:0]								W : 00
PHASE-LOCKED LOOP (PLL) REGISTERS (page 85)										
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00
1,E2h	OSC_CR2	PLLGAIN			SLP_EXTEND	WDR32_SE	EXTCLKEN	RSVD	SYSCLKX2 DIS	RW : 00

Summary Table of the Core Registers *(continued)*

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
SLEEP AND WATCHDOG REGISTERS (page 90)										
0,E0h 4 Cols.	INT_MSK0	VC3	Sleep	GPIO	Analog 3	Analog 2	Analog 1	Analog 0	V Monitor	RW : 00
0,E3h	RES_WDT	WDSL_Clear[7:0]								W : 00
x,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW	ECO EX		IRAMDIS	# : 00
x,FFh	CPU_SCR0	GIES		WDRS	PORS	Sleep			STOP	# : XX
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00
1,E2h	OSC_CR2	PLLGAIN			SLP_EXTEND	WDR32_SE	EXTCLKEN	RSVD	SYSCLKX2 DIS	RW : 00
1,E9h	ILO_TR			Bias Trim[1:0]		Freq Trim[3:0]				W : 00
1,EBh	ECO_TR	PSSDC[1:0]								W : 00

LEGEND
L The and f, expr; or f, expr; and xor f, expr instructions can be used to modify this register.
Access is bit specific. Refer to the Register Details chapter on page 103 for additional information.
X The value for power on reset is unknown.
x An “x” before the comma in the address field indicates that this register can be accessed or written to no matter what bank is used.
C Clearable register or bit(s).
R Read register or bit(s).
W Write register or bit(s).

2. CPU Core (M8C)

This chapter explains the CPU Core, called M8C, and its associated register. It covers the internal M8C registers, address spaces, ***instruction*** formats, and addressing modes. For additional information concerning the M8C instruction set, refer to the *PSoC Designer Assembly Language User Guide* available at the Cypress web site (http://www.cypress.com/psoc). For a complete table of the CPU Core registers, refer to the "Summary Table of the Core Registers" on page 30. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

2.1 Overview

The ***M8C*** is a four MIPS 8-bit Harvard architecture microprocessor. Selectable processor clock speeds from 93.7 kHz to 24 MHz allow the M8C to be tuned to a particular application's performance and power requirements. The M8C supports a rich instruction set which allows for efficient low level language support.

2.2 Internal Registers

The M8C has five internal registers that are used in program execution. The following is a list of these registers.

- Accumulator (A)
- Index (X)
- Program Counter (PC)
- Stack Pointer (SP)
- Flags (F)

All of the internal M8C registers are eight bits in width, except for the PC which is 16 bits wide. Upon ***reset***, A, X, PC, and SP are reset to 00h. The Flag register (F) is reset to 02h, indicating that the Z ***flag*** is ***set***.

With each ***stack*** operation, the SP is automatically incremented or decremented so that it always points to the next stack ***byte*** in RAM. If the last byte in the stack is at address FFh, the ***stack pointer*** will wrap to RAM address 00h. It is the ***firmware*** developer's responsibility to ensure that the stack does not overlap with user-defined variables in RAM.

With the exception of the F register, the M8C internal registers are not accessible via an explicit register address. The internal M8C registers are accessed using the following instructions:

- `MOV A, expr`
- `MOV X, expr`
- `SWAP A, SP`
- `OR F, expr`
- `JMP LABEL`

The F register can be read by using address F7h in either register bank.

2.3 Address Spaces

The M8C has three address spaces: ***ROM***, ***RAM***, and registers. The ROM address space includes the supervisory ROM (SROM) and the Flash. The ROM address space is accessed via its own address and ***data bus***.

The ROM address space is composed of the Supervisory ROM and the on-chip Flash program store. Flash is organized into 64-byte blocks. The user need not be concerned with program store page boundaries, as the M8C automatically increments the 16-bit PC on every instruction making the block boundaries invisible to user code. Instructions occurring on a 256-byte Flash page boundary (with the exception of jmp instructions) incur an extra M8C clock cycle, as the upper byte of the PC is incremented.

The register address space is used to configure the PSoC microcontroller's programmable blocks. It consists of two banks of 256 bytes each. To switch between banks, the XIO bit in the Flag register is set or cleared (set for Bank1, cleared for Bank0). The common convention is to leave the bank set to Bank0 (XIO cleared), switch to Bank1 as needed (set XIO), then switch back to Bank0.

2.4 Instruction Set Summary

The instruction set is summarized in both Table 2-1 and Table 2-2 (in numeric and ***mnemonic*** order, respectively), and serves as a quick reference. If more information is needed, the Instruction Set Summary tables are described in detail in the *PSoC Designer Assembly Language User Guide* (refer to the http://www.cypress.com/psoc web site).

Table 2-1. Instruction Set Summary Sorted Numerically by Opcode

Opcode Hex	Cycles	Bytes	Instruction Format	Flags	Opcode Hex	Cycles	Bytes	Instruction Format	Flags	Opcode Hex	Cycles	Bytes	Instruction Format	Flags
00	15	1	SSC		2D	8	2	OR [X+expr], A	Z	5A	5	2	MOV [expr], X	
01	4	2	ADD A, expr	C, Z	2E	9	3	OR [expr], expr	Z	5B	4	1	MOV A, X	Z
02	6	2	ADD A, [expr]	C, Z	2F	10	3	OR [X+expr], expr	Z	5C	4	1	MOV X, A	
03	7	2	ADD A, [X+expr]	C, Z	30	9	1	HALT		5D	6	2	MOV A, reg[expr]	Z
04	7	2	ADD [expr], A	C, Z	31	4	2	XOR A, expr	Z	5E	7	2	MOV A, reg[X+expr]	Z
05	8	2	ADD [X+expr], A	C, Z	32	6	2	XOR A, [expr]	Z	5F	10	3	MOV [expr], [expr]	
06	9	3	ADD [expr], expr	C, Z	33	7	2	XOR A, [X+expr]	Z	60	5	2	MOV reg[expr], A	
07	10	3	ADD [X+expr], expr	C, Z	34	7	2	XOR [expr], A	Z	61	6	2	MOV reg[X+expr], A	
08	4	1	PUSH A		35	8	2	XOR [X+expr], A	Z	62	8	3	MOV reg[expr], expr	
09	4	2	ADC A, expr	C, Z	36	9	3	XOR [expr], expr	Z	63	9	3	MOV reg[X+expr], expr	
0A	6	2	ADC A, [expr]	C, Z	37	10	3	XOR [X+expr], expr	Z	64	4	1	ASL A	C, Z
0B	7	2	ADC A, [X+expr]	C, Z	38	5	2	ADD SP, expr		65	7	2	ASL [expr]	C, Z
0C	7	2	ADC [expr], A	C, Z	39	5	2	CMP A, expr	if (A=B) Z=1 if (A<B) C=1	66	8	2	ASL [X+expr]	C, Z
0D	8	2	ADC [X+expr], A	C, Z	3A	7	2	CMP A, [expr]		67	4	1	ASR A	C, Z
0E	9	3	ADC [expr], expr	C, Z	3B	8	2	CMP A, [X+expr]		68	7	2	ASR [expr]	C, Z
0F	10	3	ADC [X+expr], expr	C, Z	3C	8	3	CMP [expr], expr		69	8	2	ASR [X+expr]	C, Z
10	4	1	PUSH X		3D	9	3	CMP [X+expr], expr		6A	4	1	RLC A	C, Z
11	4	2	SUB A, expr	C, Z	3E	10	2	MVI A, [[expr]++]	Z	6B	7	2	RLC [expr]	C, Z
12	6	2	SUB A, [expr]	C, Z	3F	10	2	MVI [[expr]++], A		6C	8	2	RLC [X+expr]	C, Z
13	7	2	SUB A, [X+expr]	C, Z	40	4	1	NOP		6D	4	1	RRC A	C, Z
14	7	2	SUB [expr], A	C, Z	41	9	3	AND reg[expr], expr	Z	6E	7	2	RRC [expr]	C, Z
15	8	2	SUB [X+expr], A	C, Z	42	10	3	AND reg[X+expr], expr	Z	6F	8	2	RRC [X+expr]	C, Z
16	9	3	SUB [expr], expr	C, Z	43	9	3	OR reg[expr], expr	Z	70	4	2	AND F, expr	C, Z
17	10	3	SUB [X+expr], expr	C, Z	44	10	3	OR reg[X+expr], expr	Z	71	4	2	OR F, expr	C, Z
18	5	1	POP A	Z	45	9	3	XOR reg[expr], expr	Z	72	4	2	XOR F, expr	C, Z
19	4	2	SBB A, expr	C, Z	46	10	3	XOR reg[X+expr], expr	Z	73	4	1	CPL A	Z
1A	6	2	SBB A, [expr]	C, Z	47	8	3	TST [expr], expr	Z	74	4	1	INC A	C, Z
1B	7	2	SBB A, [X+expr]	C, Z	48	9	3	TST [X+expr], expr	Z	75	4	1	INC X	C, Z
1C	7	2	SBB [expr], A	C, Z	49	9	3	TST reg[expr], expr	Z	76	7	2	INC [expr]	C, Z
1D	8	2	SBB [X+expr], A	C, Z	4A	10	3	TST reg[X+expr], expr	Z	77	8	2	INC [X+expr]	C, Z
1E	9	3	SBB [expr], expr	C, Z	4B	5	1	SWAP A, X	Z	78	4	1	DEC A	C, Z
1F	10	3	SBB [X+expr], expr	C, Z	4C	7	2	SWAP A, [expr]	Z	79	4	1	DEC X	C, Z
20	5	1	POP X		4D	7	2	SWAP X, [expr]		7A	7	2	DEC [expr]	C, Z
21	4	2	AND A, expr	Z	4E	5	1	SWAP A, SP	Z	7B	8	2	DEC [X+expr]	C, Z
22	6	2	AND A, [expr]	Z	4F	4	1	MOV X, SP		7C	13	3	LCALL	
23	7	2	AND A, [X+expr]	Z	50	4	2	MOV A, expr	Z	7D	7	3	LJMP	
24	7	2	AND [expr], A	Z	51	5	2	MOV A, [expr]	Z	7E	10	1	RETI	C, Z
25	8	2	AND [X+expr], A	Z	52	6	2	MOV A, [X+expr]	Z	7F	8	1	RET	
26	9	3	AND [expr], expr	Z	53	5	2	MOV [expr], A		8x	5	2	JMP	
27	10	3	AND [X+expr], expr	Z	54	6	2	MOV [X+expr], A		9x	11	2	CALL	
28	11	1	ROMX	Z	55	8	3	MOV [expr], expr		Ax	5	2	JZ	
29	4	2	OR A, expr	Z	56	9	3	MOV [X+expr], expr		Bx	5	2	JNZ	
2A	6	2	OR A, [expr]	Z	57	4	2	MOV X, expr		Cx	5	2	JC	
2B	7	2	OR A, [X+expr]	Z	58	6	2	MOV X, [expr]		Dx	5	2	JNC	
2C	7	2	OR [expr], A	Z	59	7	2	MOV X, [X+expr]		Ex	7	2	JACC	
										Fx	13	2	INDEX	Z

Note 1 Interrupt acknowledge to Interrupt Vector table = 13 cycles.

Note 2 The number of cycles required by an instruction is increased by one for instructions that span 256 byte page boundaries in the Flash memory space.

Table 2-2. Instruction Set Summary Sorted Alphabetically by Mnemonic

Opcode Hex	Cycles	Bytes	Instruction Format	Flags	Opcode Hex	Cycles	Bytes	Instruction Format	Flags	Opcode Hex	Cycles	Bytes	Instruction Format	Flags
09	4	2	ADC A, expr	C, Z	76	7	2	INC [expr]	C, Z	20	5	1	POP X	
0A	6	2	ADC A, [expr]	C, Z	77	8	2	INC [X+expr]	C, Z	18	5	1	POP A	Z
0B	7	2	ADC A, [X+expr]	C, Z	Fx	13	2	INDEX	Z	10	4	1	PUSH X	
0C	7	2	ADC [expr], A	C, Z	Ex	7	2	JACC		08	4	1	PUSH A	
0D	8	2	ADC [X+expr], A	C, Z	Cx	5	2	JC		7E	10	1	RETI	C, Z
0E	9	3	ADC [expr], expr	C, Z	8x	5	2	JMP		7F	8	1	RET	
0F	10	3	ADC [X+expr], expr	C, Z	Dx	5	2	JNC		6A	4	1	RLC A	C, Z
01	4	2	ADD A, expr	C, Z	Bx	5	2	JNZ		6B	7	2	RLC [expr]	C, Z
02	6	2	ADD A, [expr]	C, Z	Ax	5	2	JZ		6C	8	2	RLC [X+expr]	C, Z
03	7	2	ADD A, [X+expr]	C, Z	7C	13	3	LCALL		28	11	1	ROMX	Z
04	7	2	ADD [expr], A	C, Z	7D	7	3	LJMP		6D	4	1	RRC A	C, Z
05	8	2	ADD [X+expr], A	C, Z	4F	4	1	MOV X, SP		6E	7	2	RRC [expr]	C, Z
06	9	3	ADD [expr], expr	C, Z	50	4	2	MOV A, expr	Z	6F	8	2	RRC [X+expr]	C, Z
07	10	3	ADD [X+expr], expr	C, Z	51	5	2	MOV A, [expr]	Z	19	4	2	SBB A, expr	C, Z
38	5	2	ADD SP, expr		52	6	2	MOV A, [X+expr]	Z	1A	6	2	SBB A, [expr]	C, Z
21	4	2	AND A, expr	Z	53	5	2	MOV [expr], A		1B	7	2	SBB A, [X+expr]	C, Z
22	6	2	AND A, [expr]	Z	54	6	2	MOV [X+expr], A		1C	7	2	SBB [expr], A	C, Z
23	7	2	AND A, [X+expr]	Z	55	8	3	MOV [expr], expr		1D	8	2	SBB [X+expr], A	C, Z
24	7	2	AND [expr], A	Z	56	9	3	MOV [X+expr], expr		1E	9	3	SBB [expr], expr	C, Z
25	8	2	AND [X+expr], A	Z	57	4	2	MOV X, expr		1F	10	3	SBB [X+expr], expr	C, Z
26	9	3	AND [expr], expr	Z	58	6	2	MOV X, [expr]		00	15	1	SSC	
27	10	3	AND [X+expr], expr	Z	59	7	2	MOV X, [X+expr]		11	4	2	SUB A, expr	C, Z
70	4	2	AND F, expr	C, Z	5A	5	2	MOV [expr], X		12	6	2	SUB A, [expr]	C, Z
41	9	3	AND reg[expr], expr	Z	5B	4	1	MOV A, X	Z	13	7	2	SUB A, [X+expr]	C, Z
42	10	3	AND reg[X+expr], expr	Z	5C	4	1	MOV X, A		14	7	2	SUB [expr], A	C, Z
64	4	1	ASL A	C, Z	5D	6	2	MOV A, reg[expr]	Z	15	8	2	SUB [X+expr], A	C, Z
65	7	2	ASL [expr]	C, Z	5E	7	2	MOV A, reg[X+expr]	Z	16	9	3	SUB [expr], expr	C, Z
66	8	2	ASL [X+expr]	C, Z	5F	10	3	MOV [expr], [expr]		17	10	3	SUB [X+expr], expr	C, Z
67	4	1	ASR A	C, Z	60	5	2	MOV reg[expr], A		4B	5	1	SWAP A, X	Z
68	7	2	ASR [expr]	C, Z	61	6	2	MOV reg[X+expr], A		4C	7	2	SWAP A, [expr]	Z
69	8	2	ASR [X+expr]	C, Z	62	8	3	MOV reg[expr], expr		4D	7	2	SWAP X, [expr]	
9x	11	2	CALL		63	9	3	MOV reg[X+expr], expr		4E	5	1	SWAP A, SP	Z
39	5	2	CMP A, expr	if (A=B) Z=1 if (A<B) C=1	3E	10	2	MVI A, [[expr]++]	Z	47	8	3	TST [expr], expr	Z
3A	7	2	CMP A, [expr]		3F	10	2	MVI [[expr]++], A		48	9	3	TST [X+expr], expr	Z
3B	8	2	CMP A, [X+expr]		40	4	1	NOP		49	9	3	TST reg[expr], expr	Z
3C	8	3	CMP [expr], expr		29	4	2	OR A, expr	Z	4A	10	3	TST reg[X+expr], expr	Z
3D	9	3	CMP [X+expr], expr		2A	6	2	OR A, [expr]	Z	72	4	2	XOR F, expr	C, Z
73	4	1	CPL A	Z	2B	7	2	OR A, [X+expr]	Z	31	4	2	XOR A, expr	Z
78	4	1	DEC A	C, Z	2C	7	2	OR [expr], A	Z	32	6	2	XOR A, [expr]	Z
79	4	1	DEC X	C, Z	2D	8	2	OR [X+expr], A	Z	33	7	2	XOR A, [X+expr]	Z
7A	7	2	DEC [expr]	C, Z	2E	9	3	OR [expr], expr	Z	34	7	2	XOR [expr], A	Z
7B	8	2	DEC [X+expr]	C, Z	2F	10	3	OR [X+expr], expr	Z	35	8	2	XOR [X+expr], A	Z
30	9	1	HALT		43	9	3	OR reg[expr], expr	Z	36	9	3	XOR [expr], expr	Z
74	4	1	INC A	C, Z	44	10	3	OR reg[X+expr], expr	Z	37	10	3	XOR [X+expr], expr	Z
75	4	1	INC X	C, Z	71	4	2	OR F, expr	C, Z	45	9	3	XOR reg[expr], expr	Z
										46	10	3	XOR reg[X+expr], expr	Z

Note 1 Interrupt acknowledge to Interrupt Vector table = 13 cycles.

Note 2 The number of cycles required by an instruction is increased by one for instructions that span 256 byte page boundaries in the Flash memory space.

2.5 Instruction Formats

The M8C has a total of seven instruction formats which use instruction lengths of one, two, and three bytes. All instruction bytes are fetched from the program memory (Flash), using an address and data bus that are independent from the address and data buses used for register and RAM access.

While examples of instructions are given in this section, refer to the *PSoC Designer Assembly Language User Guide* for detailed information on individual instructions.

2.5.1 One-Byte Instructions

Many instructions, such as some of the MOV instructions, have single-byte forms, because they do not use an address or data as an operand. As shown in Table 2-3, one-byte instructions use an 8-bit opcode. The set of one-byte instructions can be divided into four categories, according to where their results are stored.

Table 2-3. One-Byte Instruction Format

Byte 0
8-Bit Opcode

The first category of one-byte instructions are those that do not update any registers or RAM. Only the one-byte NOP and SSC instructions fit this category. While the ***program counter*** is incremented as these instructions execute, they do not cause any other internal M8C registers to be updated, nor do these instructions directly affect the register space or the RAM address space. The SSC instruction will cause SROM code to run, which will modify RAM and the M8C internal registers.

The second category has only the two PUSH instructions in it. The PUSH instructions are unique, because they are the only one-byte instructions that cause a RAM address to be modified. These instructions automatically increment the SP.

The third category has only the HALT instruction in it. The HALT instruction is unique, because it is the only a one-byte instruction that causes a user register to be modified. The HALT instruction modifies user register space address FFh (CPU_SCR register).

The final category for one-byte instructions are those that cause updates of the internal M8C registers. This category holds the largest number of instructions: ASL, ASR, CPL, DEC, INC, MOV, POP, RET, RETI, RLC, ROMX, RRC, SWAP. These instructions can cause the A, X, and SP registers or SRAM to update.

2.5.2 Two-Byte Instructions

The majority of M8C instructions are two bytes in length. While these instructions can be divided into categories identical to the one-byte instructions, this would not provide a useful distinction between the three two-byte instruction formats that the M8C uses.

Table 2-4. Two-Byte Instruction Formats

Byte 0	Byte 1
4-Bit Opcode	12-Bit Relative Address
8-Bit Opcode	8-Bit Data
8-Bit Opcode	8-Bit Address

The first two-byte instruction format, shown in the first row of Table 2-4, is used by short jumps and calls: `CALL`, `JMP`, `JACC`, `INDEX`, `JC`, `JNC`, `JNZ`, `JZ`. This instruction format uses only four bits for the instruction opcode, leaving 12 bits to store the relative destination address in a two's-complement form. These instructions can change program execution to an address relative to the current address by –2048 or +2047.

The second two-byte instruction format, shown in the second row of Table 2-4, is used by instructions that employ the Source Immediate addressing ***mode*** (see "Source Immediate" on page 37). The destination for these instructions is an internal M8C register, while the source is a constant value. An example of this type of instruction would be `ADD A, 7`.

The third two-byte instruction format, shown in the third row of Table 2-4, is used by a wide range of instructions and addressing modes. The following is a list of the addressing modes that use this third two-byte instruction format:

- Source Direct (`ADD A, [7]`)
- Source Indexed (`ADD A, [X+7]`)
- Destination Direct (`ADD [7], A`)
- Destination Indexed (`ADD [X+7], A`)
- Source Indirect Post Increment (`MVI A, [7]`)
- Destination Indirect Post Increment (`MVI [7], A`)

For more information on addressing modes see "Addressing Modes" on page 37.

2.5.3 Three-Byte Instructions

The three-byte instruction formats are the second most prevalent instruction formats. These instructions need three bytes because they either move data between two addresses in the user-accessible address space (registers and RAM) or they hold 16-bit absolute addresses as the destination of a long jump or long call.

Table 2-5. Three-Byte Instruction Formats

Byte 0	Byte 1	Byte 2
8-Bit Opcode	16-Bit Address (MSB, LSB)	
8-Bit Opcode	8-Bit Address	8-Bit Data
8-Bit Opcode	8-Bit Address	8-Bit Address

The first instruction format, shown in the first row of Table 2-5, is used by the LJMP and LCALL instructions. These instructions change program execution unconditionally to an absolute address. The instructions use an 8-bit opcode, leaving room for a 16-bit destination address.

The second three-byte instruction format, shown in the second row of Table 2-5, is used by the following two addressing modes:

- Destination Direct Source Immediate (`ADD [7], 5`)
- Destination Indexed Source Immediate (`ADD [X+7], 5`)

The third three-byte instruction format, shown in the third row of Table 2-5, is for the Destination Direct Source Direct addressing mode, which is used by only one instruction. This instruction format uses an 8-bit opcode followed by two 8-bit addresses. The first address is the destination address in RAM, while the second address is the source address in RAM. The following is an example of this instruction:

```
MOV [7], [5]
```

2.6 Addressing Modes

The M8C has ten addressing modes. These modes are detailed and located on the following pages:

- “Source Immediate” on page 37.
- “Source Direct” on page 38.
- “Source Indexed” on page 38.
- “Destination Direct” on page 39.
- “Destination Indexed” on page 39.
- “Destination Direct Source Immediate” on page 39.
- “Destination Indexed Source Immediate” on page 40.
- “Destination Direct Source Direct” on page 40.
- “Source Indirect Post Increment” on page 41.
- “Destination Indirect Post Increment” on page 41.

2.6.1 Source Immediate

For these instructions, the source value is stored in operand 1 of the instruction. The result of these instructions is placed in either the M8C A, F, or X register as indicated by the instruction’s opcode. All instructions using the Source Immediate addressing mode are two bytes in length.

Table 2-6. Source Immediate

Opcode	Operand 1
Instruction	Immediate Value

Source Immediate Examples:

Source Code	Machine Code	Comments
`ADD A, 7`	`01 07`	The immediate value 7 is added to the Accumulator. The result is placed in the Accumulator.
`MOV X, 8`	`57 08`	The immediate value 8 is moved into the X register.
`AND F, 9`	`70 09`	The immediate value of 9 is logically AND’ed with the F register and the result is placed in the F register.

2.6.2 Source Direct

For these instructions, the source address is stored in operand 1 of the instruction. During instruction execution, the address will be used to retrieve the source value from RAM or register address space. The result of these instructions is placed in either the M8C A or X register as indicated by the instruction's opcode. All instructions using the Source Direct addressing mode are two bytes in length.

Table 2-7. Source Direct

Opcode	Operand 1
Instruction	Source Address

Source Direct Examples:

Source Code	Machine Code	Comments
`ADD A, [7]`	`02 07`	The value in memory at address 7 is added to the Accumulator and the result is placed into the Accumulator.
`MOV A, REG[8]`	`5D 08`	The value in the register space at address 8 is moved into the Accumulator.

2.6.3 Source Indexed

For these instructions, the source offset from the X register is stored in operand 1 of the instruction. During instruction execution, the current X register value is added to the signed offset, to determine the address of the source value in RAM or register address space. The result of these instructions is placed in either the M8C A or X register as indicated by the instruction's opcode. All instructions using the Source Indexed addressing mode are two bytes in length.

Table 2-8. Source Indexed

Opcode	Operand 1
Instruction	Source Index

Source Indexed Examples:

Source Code	Machine Code	Comments
`ADD A, [X+7]`	`03 07`	The value in memory at address X+7 is added to the Accumulator. The result is placed in the Accumulator.
`MOV X, [X+8]`	`59 08`	The value in RAM at address X+8 is moved into the X register.

2.6.4 Destination Direct

For these instructions, the destination address is stored in the machine code of the instruction. The source for the operation is either the M8C A or X register as indicated by the instruction's opcode. All instructions using the Destination Direct addressing mode are two bytes in length.

Table 2-9. Destination Direct

Opcode	Operand 1
Instruction	Destination Address

Destination Direct Examples:

Source Code	Machine Code	Comments
`ADD [7], A`	`04 07`	The value in the Accumulator is added to memory at address 7. The result is placed in memory at address 7. The Accumulator is unchanged.
`MOV REG[8], A`	`60 08`	The Accumulator value is moved to register space at address 8. The Accumulator is unchanged.

2.6.5 Destination Indexed

For these instructions, the destination offset from the X register is stored in the machine code for the instruction. The source for the operation is either the M8C A register or an immediate value as indicated by the instruction's opcode. All instructions using the Destination Indexed addressing mode are two bytes in length.

Table 2-10. Destination Indexed

Opcode	Operand 1
Instruction	Destination Index

Destination Indexed Example:

Source Code	Machine Code	Comments
`ADD [X+7], A`	`05 07`	The value in memory at address X+7 is added to the Accumulator. The result is placed in memory at address X+7. The Accumulator is unchanged.

2.6.6 Destination Direct Source Immediate

For these instructions, the destination address is stored in operand 1 of the instruction. The source value is stored in operand 2 of the instruction. All instructions using the Destination Direct Source Immediate addressing mode are three bytes in length.

Table 2-11. Destination Direct Source Immediate

Opcode	Operand 1	Operand 2
Instruction	Destination Address	Immediate Value

Destination Direct Source Immediate Examples:

Source Code	Machine Code	Comments
`ADD [7], 5`	`06 07 05`	The value in memory at address 7 is added to the immediate value 5. The result is placed in memory at address 7.
`MOV REG[8], 6`	`62 08 06`	The immediate value 6 is moved into register space at address 8.

2.6.7 Destination Indexed Source Immediate

For these instructions, the destination offset from the X register is stored in operand 1 of the instruction. The source value is stored in operand 2 of the instruction. All instructions using the Destination Indexed Source Immediate addressing mode are three bytes in length.

Table 2-12. Destination Indexed Source Immediate

Opcode	Operand 1	Operand 2
Instruction	Destination Index	Immediate Value

Destination Indexed Source Immediate Examples:

Source Code	Machine Code	Comments
`ADD [X+7], 5`	`07 07 05`	The value in memory at address X+7 is added to the immediate value 5. The result is placed in memory at address X+7.
`MOV REG[X+8], 6`	`63 08 06`	The immediate value 6 is moved into the register space at address X+8.

2.6.8 Destination Direct Source Direct

Only one instruction uses this addressing mode. The destination address is stored in operand 1 of the instruction. The source address is stored in operand 2 of the instruction. The instruction using the Destination Direct Source Direct addressing mode is three bytes in length.

Table 2-13. Destination Direct Source Direct

Opcode	Operand 1	Operand 2
Instruction	Destination Address	Source Address

Destination Direct Source Direct Example:

Source Code	Machine Code	Comments
`MOV [7], [8]`	`5F 07 08`	The value in memory at address 8 is moved to memory at address 7.

2.6.9 Source Indirect Post Increment

Only one instruction uses this addressing mode. The source address stored in operand 1 is actually the address of a pointer. During instruction execution, the pointer's current value is read to determine the address in RAM where the source value is found. The pointer's value is incremented after the source value is read. For PSoC microcontrollers with more than 256 bytes of RAM, the Data Page Read (MVR_PP) register is used to determine which RAM page to use with the source address. Therefore, values from pages other than the current page can be retrieved without changing the Current Page Pointer (CUR_PP). The pointer is always read from the current RAM page. For information on the MVR_PP and CUR_PP registers, see the Register Details chapter on page 103. The instruction using the Source Indirect Post Increment addressing mode is two bytes in length.

Table 2-14. Source Indirect Post Increment

Opcode	Operand 1
Instruction	Source Address Pointer

Source Indirect Post Increment Example:

Source Code	Machine Code	Comments
`MVI A, [8]`	`3E 08`	The value in memory at address 8 (the indirect address) points to a memory location in RAM. The value at the memory location, pointed to by the indirect address, is moved into the Accumulator. The indirect address, at address 8 in memory, is then incremented.

2.6.10 Destination Indirect Post Increment

Only one instruction uses this addressing mode. The destination address stored in operand 1 is actually the address of a pointer. During instruction execution, the pointer's current value is read to determine the destination address in RAM where the Accumulator's value is stored. The pointer's value is incremented, after the value is written to the destination address. For PSoC microcontrollers with more than 256 bytes of RAM, the Data Page Write (MVW_PP) register is used to determine which RAM page to use with the destination address. Therefore, values can be stored in pages other than the current page without changing the Current Page Pointer (CUR_PP). The pointer is always read from the current RAM page. For information on the MVR_PP and CUR_PP registers, see the Register Details chapter on page 103. The instruction using the Destination Indirect Post Increment addressing mode is two bytes in length.

Table 2-15. Destination Indirect Post Increment

Opcode	Operand 1
Instruction	Destination Address Pointer

Destination Indirect Post Increment Example:

Source Code	Machine Code	Comments
`MVI [8], A`	`3F 08`	The value in memory at address 8 (the indirect address) points to a memory location in RAM. The Accumulator value is moved into the memory location pointed to by the indirect address. The indirect address, at address 8 in memory, is then incremented.

2.7 Register Definitions

The following register is associated with the CPU Core (M8C). The register description has an associated register table showing the bit structure. The bits that are grayed out in the table are reserved bits and are not detailed in the register description that follows. Reserved bits should always be written with a value of ‘0’.

2.7.1 CPU_F Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,F7h	CPU_F	PgMode[1:0]			XIO		Carry	Zero	GIE	RL : 02

LEGEND
L The AND F, expr; OR F, expr; and XOR F, expr flag instructions can be used to modify this register.
x An “x” before the comma in the address field indicates that this register can be read or written to no matter what bank is used.

The M8C Flag Register (CPU_F) provides read access to the M8C flags.

Bits 7 and 6: PgMode[1:0]. PgMode determines how the CUR_PP, STK_PP, and IDX_PP registers are used in forming effective RAM addresses for Direct Address mode and Indexed Address mode operands. PgMode also determines whether the stack page is determined by the STK_PP or IDX_PP register.

Bit 4: XIO. The I/O Bank Select bit, also known as the register bank select bit, is used to select the register bank that is active for a register read or write. This bit allows the PSoC device to have 512 8-bit registers and therefore, can be thought of as the ninth address bit for registers. The address space accessed when the XIO bit is set to ‘0’ is called the ***user space***, while the address space accessed when the XIO bit is set to ‘1’ is called the ***configuration space***.

Bit 2: Carry. The Carry flag bit is set or cleared in response to the result of several instructions. It can also be manipulated by the flag-logic opcodes (for example, OR F, 4). See the *PSoC Designer Assembly Guide User Manual* for more details.

Bit 1: Zero. The Zero flag bit is set or cleared in response to the result of several instructions. It can also be manipulated by the flag-logic opcodes (for example, OR F, 2). See the *PSoC Designer Assembly Guide User Manual* for more details.

Bit 0: GIE. The state of the Global Interrupt Enable bit determines whether interrupts (by way of the interrupt request (IRQ)) will be recognized by the M8C. This bit is set or cleared by the user, using the flag-logic instructions (for example, OR F, 1). GIE is also cleared automatically when an interrupt is processed, after the flag byte has been stored on the stack, preventing nested interrupts. If desired, the bit can be set in an ***interrupt service routine (ISR)***.

For GIE=1, the M8C samples the IRQ input for each instruction. For GIE=0, the M8C ignores the IRQ.

For additional information, refer to the CPU_F register on page 188.

3. Supervisory ROM (SROM)

This chapter discusses the Supervisory ROM (SROM) functions and its associated registers. For a complete table of the SROM registers, refer to the "Summary Table of the Core Registers" on page 30. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

3.1 Architectural Description

The SROM holds code that is used to boot the PSoC device, calibrate circuitry, and perform Flash operations. The functions provided by the SROM are called from code stored in the Flash or by device programmers.

The SROM is used to boot the part and provide ***interface*** functions to the Flash banks. (Table 3-1 lists the SROM functions.) The SROM functions are accessed by executing the Supervisory System Call instruction (SSC) which has an opcode of 00h. Prior to executing the SSC, the M8C's ***accumulator*** needs to load with the desired SROM function code from Table 3-1. Attempting to access undefined functions will cause a HALT. The SROM functions execute code with calls; therefore, the functions require stack space. With the exception of Reset, all of the SROM functions have a ***parameter block*** in SRAM that must be configured before executing the SSC. Table 3-2 lists all possible parameter block variables. The meaning of each ***parameter***, with regards to a specific SROM function, is described later in this chapter. Because the SSC instruction clears the CPU_F PgMode bits, all parameter block variable addresses are in SRAM Page 0. The CPU_F value is automatically restored at the end of the SROM function.

Note For PSoC devices with more than 256 bytes of SRAM (that is, more than 1 page of SRAM, see the table titled "PSoC Device SRAM Availability" on page 53), the MVR_PP and the MVW_PP pointers are not disabled by clearing the CPU_F PgMode bits. Therefore, the POINTER parameter is interpreted as an address in the page indicated by the MVI page pointers, when the supervisory operation is called. This allows the data ***buffer*** used in the supervisory operation to be located in any SRAM page. (See the RAM Paging chapter on page 53 for more details regarding the MVR_PP and MVW_PP pointers.)

Table 3-1. List of SROM Functions

Function Code	Function Name	Stack Space Needed	Page
00h	SWBootReset	0	44
01h	ReadBlock	7	45
02h	WriteBlock	10	45
03h	EraseBlock	9	46
06h	TableRead	3	46
07h	CheckSum	3	47
08h	Calibrate0	4	47
09h	Calibrate1	3	47

Note ProtectBlock (described on page 46) and EraseAll (described on page 47) SROM functions are not listed in the table above because they are dependent on external programming.

Two important variables that are used for all functions are KEY1 and KEY2. These variables are used to help discriminate between valid SSCs and inadvertent SSCs. KEY1 must always have a value of 3Ah, while KEY2 must have the same value as the stack pointer when the SROM function begins execution. This would be the SP (Stack Pointer) value when the SSC opcode is executed, plus three. For all SROM functions except SWBootReset, if either of the keys do not match the expected values, the M8C will halt. The SWBootReset function does not check the key values. It only checks to see if the accumulator's value is 0x00. The following code example puts the correct value in KEY1 and KEY2. The code is preceded by a HALT, to force the program to jump directly into the setup code and not accidentally run into it.

```
1.        halt
2. SSC0P: mov [KEY1], 3ah
3.        mov X, SP
4.        mov A, X
5.        add A, 3
6.        mov [KEY2], A
```

Table 3-2. SROM Function Variables

Variable Name	SRAM Address
KEY1 / RETURN CODE	0,F8h
KEY2	0,F9h
BLOCKID	0,FAh
POINTER	0,FBh
CLOCK	0,FCh
Reserved	0,FDh
DELAY	0,FEh
Reserved	0,FFh

3.1.1 Additional SROM Feature

The SROM has the following additional feature.

Return Codes: These aid in the determination of success or failure of a particular function. The return code is stored in KEY1's position in the parameter block. The CheckSum and TableRead functions do not have return codes because KEY1's position in the parameter block is used to return other data.

Table 3-3. SROM Return Code Meanings

Return Code Value	Description
00h	Success
01h	Function not allowed due to level of protection on the block.
02h	Software reset without hardware reset.
03h	Fatal error, SROM halted.

Note Read, write, and erase operations may fail if the target block is read or write protected. Block protection levels are set during device programming and can not be modified from code in the PSoC device.

3.1.2 SROM Function Descriptions

3.1.2.1 SWBootReset Function

The SROM function SWBootReset is responsible for transitioning the device from a reset state to running ***user*** code. See "System Resets" on page 412 for more information on what events will cause the SWBootReset function to execute.

The SWBootReset function is executed whenever the SROM is entered with an M8C accumulator value of 00h; the SRAM parameter block is not used as an input to the function. This will happen, by design, after a ***hardware*** reset, because the M8C's accumulator is reset to 00h or when user code executes the SSC instruction with an accumulator value of 00h.

If the ***checksum*** of the calibration data is valid, the SWBootReset function ends by setting the internal M8C registers (CPU_SP, CPU_PC, CPU_X, CPU_F, CPU_A) to 00h writing 00h to most SRAM addresses in SRAM Page 0 and then begins to execute user code at address 0000h. (See Table 3-4 and the following paragraphs for more information on which SRAM addresses are modified.) If the checksum is not valid, an internal reset is executed and the boot process starts over. If this condition occurs, the internal reset status bit (IRESS) is set in the CPU_SCR1 register.

In PSoC devices with more than 256 bytes of SRAM, no SRAM is modified by the SWBootReset function in SRAM pages numbered higher than '0'.

Table 3-4 documents the value of all the SRAM addresses in Page 0 after a successful SWBootReset. A cell in the table with "xx" indicates that the SRAM address is not modified by the SWBootReset function. A hex value in a cell indicates that the address should always have the indicated value after a successful SWBootReset. A cell with a "??" in it indicates that the value, after a SWBootReset, is determined by the value of IRAMDIS bit in the CPU_SCR1 register. If IRAMDIS is not set, these addresses will be initialized to 00h. If IRAMDIS is set, these addresses will not be modified by a SWBootReset after a watchdog reset. The IRAMDIS bit allows variables to be preserved even if a watchdog reset (WDR) occurs. The IRAMDIS bit is reset by all system resets except watchdog reset. Therefore, this bit is only useful for watchdog resets and not general resets.

Table 3-4. SRAM Map Post SWBootReset (00h)

Address	0	1	2	3	4	5	6	7
	8	9	A	B	C	D	E	F
0x0_	0x00	0x00	0x00	??	??	??	??	??
	??	??	??	??	??	??	??	??
0x1_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0x2_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0x3_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0x4_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0x5_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0x6_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0x7_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0x8_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0x9_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0xA_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0xB_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0xC_	??	??	??	??	??	??	??	??
	??	??	??	??	??	??	??	??
0xD_	??	??	??	??	??	??	??	??
	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0xE_	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
	0x00	0x00	0x00	0x00	0x00	0x00	0x00	0x00
0xF_	0x00	0x00	0x00	0x00	0x00	0x00	??	??
	0x00 0x02	xx	0x00	0x00	0xn	xx	0x00	0x00

Address F8h is the return code byte for all SROM functions (except Checksum and TableRead); for this function, the only acceptable values are 00h and 02h. Address FCh is the fail count variable. After POR (Power on Reset), WDR, or XRES (External Reset), the variable is initialized to 00h by the SROM. Each time the checksum fails, the fail count is incremented. Therefore, if it takes two passes through SWBootReset to get a good checksum, the fail count would be 01h.

3.1.2.2 ReadBlock Function

The ReadBlock function is used to read 64 contiguous bytes from Flash: a ***block***. The number of blocks in a device is the total number of bytes divided by 64. Refer to Table 3-5 to determine the amount of space in your PSoC device.

Table 3-5. Flash Memory Organization

PSoC Device	Amount of Flash	Amount of SRAM	Number of Blocks per Bank	Number of Banks
CY8C29x66	32 KB	2 KB	128	4
CY8C21345	8KB	512 Bytes	128	1
CY8C21x34	8 KB	512 Bytes	128	1
CY8C21x23	4 KB	256 Bytes	64	1
CY7C64215	16 KB	1 KB	128	2
CY7C603xx	8 KB	512 Bytes	128	1
CYWUSB6953	8 KB	512 Bytes	128	1
CY8C22X45	16 KB	1 KB	128	2

The first thing the ReadBlock function does is check the protection bits to determine if the desired BLOCKID is readable. If read protection is turned on, the ReadBlock function will exit setting the accumulator and KEY2 back to 00h. KEY1 will have a value of 01h, indicating a read failure.

If read protection is not enabled, the function will read 64 bytes from the Flash using a ROMX instruction and store the results in SRAM using an MVI instruction. The 64 bytes are stored in SRAM, beginning at the address indicated by the value of the POINTER parameter. When the ReadBlock completes successfully, the accumulator, KEY1, and KEY2 will all have a value of 00h.

If the PSoC device has more than one bank of Flash, the bank value in the FLS_PR1 register must be set prior to executing the SSC instruction. Refer to Table 3-5.

Note A MVI [expr], A is used to store the Flash block contents in SRAM; thus, the MVW_PP register can be set to indicate which SRAM pages will receive the data.

Table 3-6. ReadBlock Parameters (01h)

Name	Address	Type	Description
MVW_PP	0,D5h	Register	MVI write page pointer register
KEY1	0,F8h	RAM	3Ah
KEY2	0,F9h	RAM	Stack Pointer value+3, when SSC is executed.
BLOCKID	0,FAh	RAM	Flash block number
POINTER	0,FBh	RAM	Addresses in SRAM where returned data should be stored.
FLS_PR1	1,FAh	Register	Flash bank number.

3.1.2.3 WriteBlock Function

The WriteBlock function is used to store data in the Flash. Data is moved 64 bytes at a time from SRAM to Flash using this function. Before a write can be performed, either an EraseAll or an EraseBlock must be completed successfully.

The first thing the WriteBlock function does is check the protection bits and determine if the desired BLOCKID is writeable. If write protection is turned on, the WriteBlock function will exit, setting the accumulator and KEY2 back to 00h. KEY1 will have a value of 01h, indicating a write failure. Write protection is set when the PSoC device is programmed externally and cannot be changed through the SSC function.

The BLOCKID of the ***Flash block***, where the data is stored, must be determined and stored at SRAM address FAh. For valid BLOCKID values, refer to Table 3-5.

An MVI A, [expr] instruction is used to move data from SRAM into Flash. Therefore, the MVI read pointer (MVR_PP register) can be used to specify which SRAM page data is pulled from. Using the MVI read pointer and the parameter blocks POINTER value allows the SROM WriteBlock function to move data from any SRAM page into any Flash block, in either Flash bank.

The SRAM address, of the first of the 64 bytes to be stored in Flash, must be indicated using the POINTER variable in the parameter block (SRAM address FBh).

Finally, the CLOCK and DELAY value must be set correctly. The CLOCK value determines the length of the write ***pulse*** that will be used to store the data in the Flash. The CLOCK and DELAY values are dependent on the CPU speed and must be set correctly. Refer to “Clocking” on page 52 for additional information.

If the PSoC device you are using has more than one bank of Flash, the bank value in the FLS_PR1 register must be set prior to executing the SSC instruction. Refer to Table 3-5.

Table 3-7. WriteBlock Parameters (02h)

Name	Address	Type	Description
MVR_PP	0,D4h	Register	MVI read page pointer register.
KEY1	0,F8h	RAM	3Ah
KEY2	0,F9h	RAM	Stack Pointer value+3, when SSC is executed.
BLOCKID	0,FAh	RAM	Flash block number.
POINTER	0,FBh	RAM	First of 64 addresses in SRAM, where the data to be stored in Flash is located prior to calling WriteBlock.
CLOCK	0,FCh	RAM	Clock divider used to set the write pulse width.
DELAY	0,FEh	RAM	For a CPU speed of 12 MHz set to 56h.
FLS_PR1	1,FAh	Register	Flash bank number.

3.1.2.4 EraseBlock Function

The EraseBlock function is used to erase a block of 64 contiguous bytes in Flash.

The first thing the EraseBlock function does is check the protection bits and determine if the desired BLOCKID is writeable. If write protection is turned on, the EraseBlock function will exit, setting the accumulator and KEY2 back to 00h. KEY1 will have a value of 01h, indicating a write failure.

To set up the parameter block for the EraseBlock function, correct key values must be stored in KEY1 and KEY2. The block number to be erased must be stored in the BLOCKID variable, and the CLOCK and DELAY values must be set based on the current CPU speed. For more information on setting the CLOCK and DELAY values, see “Clocking” on page 52.

If the PSoC device you are using has more than one bank of Flash, the bank value in the FLS_PR1 register must be set prior to executing the SSC instruction. Refer to Table 3-5.

Table 3-8. EraseBlock Parameters (03h)

Name	Address	Type	Description
KEY1	0,F8h	RAM	3Ah
KEY2	0,F9h	RAM	Stack Pointer value+3, when SSC is executed.
BLOCKID	0,FAh	RAM	Flash block number.
CLOCK	0,FCh	RAM	Clock divider used to set the erase pulse width.
DELAY	0,FEh	RAM	For a CPU speed of 12 MHz set to 56h.
FLS_PR1	1,FAh	Register	Flash bank number.

3.1.2.5 ProtectBlock Function

The PSoC devices offer Flash protection on a block-by-block basis. Table 3-9 lists the protection modes available. In the table, ER and EW are used to indicate the ability to perform external reads and writes (that is, by an external programmer). For internal writes, IW is used. Internal reading is always permitted by way of the ROMX instruction. The ability to read by way of the SROM ReadBlock function is indicated by SR.

In the table below, note that all protection is removed by EraseAll.

Table 3-9. Protect Block Modes

Mode	Settings	Description	In PSoC Designer
00b	SR ER EW IW	Unprotected	U = Unprotected
01b	$\overline{\text{SR}}$ $\overline{\text{ER}}$ EW IW	Read protect	F = Factory upgrade
10b	SR ER $\overline{\text{EW}}$ IW	Disable external write	R = Field upgrade
11b	$\overline{\text{SR}}$ $\overline{\text{ER}}$ $\overline{\text{EW}}$ $\overline{\text{IW}}$	Disable internal write	W = Full protection

3.1.2.6 TableRead Function

The TableRead function gives the user access to part-specific data stored in the Flash during manufacturing. The Flash for these tables is separate from the program Flash and is not directly accessible.

One of the uses of the SROM TableRead function is to retrieve the values needed to optimize Flash programming for temperature. More information about how to use these

values may be found in the section titled “Clocking” on page 52.

Table 3-10. TableRead Parameters (06h)

Name	Address	Type	Description
KEY1	0,F8h	RAM	3Ah
KEY2	0,F9h	RAM	Stack Pointer value+3, when SSC is executed.
BLOCKID	0,FAh	RAM	Table number to read.

3.1.2.7 EraseAll Function

The EraseAll function performs a series of steps that destroys the user data in the Flash banks and resets the protection block in each Flash bank to all zeros (the unprotected state). This function may only be executed by an external programmer. If EraseAll is executed from code, the M8C will HALT without touching the Flash or protections.

Table 3-11. Flash Tables with Assigned Values in Flash Bank 0

	F8h	F9h	FAh	FBh	FCh	FDh	FEh	FFh
Table 0	Silicon ID							
Table 1	Voltage Reference Trim for 3.3 V reg[1,EA]	IMO Trim for 3.3 V reg[1,E8]	Room Temperature Calibration for 3.3 V	Hot Temperature Calibration for 3.3 V	Voltage Reference Trim for 5 V reg[1,EA]	IMO Trim for 5 V reg[1,E8]	Room Temperature Calibration for 5 V	Hot Temperature Calibration for 5 V
Table 2	Voltage Reference Trim for 2.7 V reg[1,EA]	IMO Slow Trim 12 MHz Vdd = 2.7 V	Room Temperature Calibration for 2.7 V *	Hot Temperature Calibration for 2.7 V *	IMO Slow Trim 6 MHz Vdd = 3.3 V	IMO Slow Trim 6 MHz Vdd = 2.7 V	IMO Slow Trim 6 MHz Vdd = 5.0 V	
Table 3	M (cold)	B (cold)	Mult (cold)	M (hot)	B (hot)	Mult (hot)	00h	01h

* CY8C24x94 and CY7C64215 Table 2: FAh = IMO Trim 2 for 3.3 V, FBh = IMO Trim 2 for 5 V.

3.1.2.8 Checksum Function

The Checksum function calculates a 16-bit checksum over a user specifiable number of blocks, within a single ***Flash bank*** starting at block zero. The BLOCKID parameter is used to pass in the number of blocks to checksum. A BLOCKID value of '1' calculates the checksum of only block0, a BLOCKID of '2' calculates the checksum of block 0 and block 1, and so on. A BLOCKID value of '0' calculates the checksum of the entire Flash bank. Note that if the BLOCKID is greater than the number of blocks that the device has in a Flash bank, the function calculates checksum for the entire Flash bank and repeats the process of checksum from block 0 in that Flash bank. For example, if the BLOCKID is equal to 150, the function calculates checksum for block 0 to block 127 (if the device has 128 blocks in a Flash bank) and again for block 0 to block 21.

The 16-bit checksum is returned in KEY1 and KEY2. The parameter KEY1 holds the lower 8 bits of the checksum and the parameter KEY2 holds the upper 8 bits of the checksum. For devices with multiple Flash banks, the checksum function must be called once for each Flash bank. The SROM Checksum function will operate on the Flash bank indicated by the Bank bit in the FLS_PR1 register.

Table 3-12. Checksum Parameters (07h)

Name	Address	Type	Description
KEY1	0,F8h	RAM	3Ah
KEY2	0,F9h	RAM	Stack Pointer value+3, when SSC is executed.
BLOCKID	0,FAh	RAM	Number of Flash blocks to calculate checksum on.
FLS_PR1	1,FAh	Register	Flash bank number.

3.1.2.9 Calibrate0 Function

The Calibrate0 function transfers the calibration values stored in a special area of the Flash to their appropriate registers. This function may be executed at any time to set all calibration values back to their 5 V values. However, it should not be necessary to call this function. This function is simply documented for completeness. 3.3 V calibration values are accessed by way of the TableRead function, which is described in the section titled “TableRead Function” on page 46.

Table 3-13. Calibrate0 Parameters (08h)

Name	Address	Type	Description
KEY1	0,F8h	RAM	3Ah
KEY2	0,F9h	RAM	Stack Pointer value+3, when SSC is executed.

3.1.2.10 Calibrate1 Function

While the Calibrate1 function is a completely separate function from Calibrate0, they perform the same function, which is to transfer the calibration values stored in a special area of the Flash to their appropriate registers. What is unique about Calibrate1 is that it calculates a checksum of the calibration data and, if that checksum is determined to be invalid, Calibrate1 will cause a ***hardware reset*** by generating an internal reset. If this occurs, it is indicated by setting the Internal Reset Status bit (IRESS) in the CPU_SCR1 register.

The Calibrate1 function uses SRAM to calculate a checksum of the calibration data. The POINTER value is used to indicate the address of a 30-byte buffer used by this func-

tion. When the function completes, the 30 bytes will be set to 00h.

An MVI A, [expr] and an MVI [expr], A instruction are used to move data between SRAM and Flash. Therefore, the MVI write pointer (MVW_PP) and the MVI read pointer (MVR_PP) must be specified to the same SRAM page to control the page of RAM used for the operations.

Calibrate1 was created as a sub-function of SWBootReset and the Calibrate1 function code was added to provide ***direct access***. For more information on how Calibrate1 works, see the SWBootReset section.

This function may be executed at any time to set all calibration values back to their 5 V values. However, it should not be necessary to call this function. This function is simply documented for completeness. This function has no argument to select between 5 V and 3.3 V calibration values; therefore, it always defaults to 5 V values. 3.3 V calibration values are accessed by way of the TableRead function, which is described in the section titled "TableRead Function" on page 46.

Table 3-14. Calibrate1 Parameters (09h)

Name	Address	Type	Description
KEY1	0,F8h	RAM	3Ah
KEY2	0,F9h	RAM	Stack Pointer value+3, when SSC is executed.
POINTER	0,FBh	RAM	First of 30 SRAM addresses used by this function.
MVR_PP	0,D4h	Register	MVI write page pointer.
MVW_PP	0,D5h	Register	MVI read page pointer.

3.2 PSoC Device Distinctions

For the CY8C22x45, CY8C21345, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices, a BLOCKID value of '0' will cause all available Flash to be checksummed. In all other PSoC devices, a BLOCKID value of '0' will checksum 256 blocks.

3.3 Register Definitions

The following registers are associated with the Supervisory ROM (SROM) and are listed in address order. The register descriptions have an associated register table showing the bit structure for that register. The bits in the tables that are grayed out are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of '0'. For a complete table of SROM registers, refer to the "Summary Table of the Core Registers" on page 30.

3.3.1 STK_PP Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,D1h	STK_PP						Page Bits[2:0]			RW : 00

The Stack Page Pointer Register (STK_PP) is used to set the effective SRAM page for stack memory accesses in a multi-SRAM page PSoC device. This register is only used when a device has more than one page of SRAM.

Bits 2 to 0: Page Bits[2:0]. This register has the potential to affect two types of memory access. The first type of memory access of the STK_PP register is to determine which SRAM page the stack will be stored on. In the reset state, this register's value is 0x00 and the stack will therefore be in SRAM Page 0. However, if the STK_PP register value is changed, the next stack operation will occur on the SRAM page indicated by the new STK_PP value. Therefore, the value of this register should be set early in the program and never be changed. If the program changes the STK_PP value after the stack has grown, the program must ensure that the STK_PP value is restored when needed.

Note The impact that the STK_PP has on the stack is independent of the SRAM Paging bits in the CPU_F register.

The second type of memory access of the STK_PP register affects indexed memory access when the CPU_F[7:6] bits are set to 11b. In this mode, source indexed and destination indexed memory accesses are directed to the stack SRAM page, rather than the SRAM page indicated by the IDX_PP register or SRAM Page 0.

For additional information, refer to the STK_PP register on page 157.

3.3.2 MVR_PP Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,D4h	MVR_PP						Page Bits[2:0]			RW : 00

The MVI Read Page Pointer Register (MVR_PP) is used to set the effective SRAM page for MVI read memory accesses in a multi-SRAM page PSoC device.

Note This register is only used when a device has more than one page of SRAM. Refer to the table titled "PSoC Device SRAM Availability" on page 53 to determine the number of SRAM pages for your PSoC device.

Bits 2 to 0: Page Bits[2:0]. This register is only used by the MVI A, [expr] instruction, not to be confused with the MVI [expr], A instruction covered by the MVW_PP register. This instruction is considered a read because data is transferred from SRAM to the microprocessor's A register (CPU_A).

When an MVI A, [expr] instruction is executed in a device with more than one page of SRAM, the SRAM address that is read by the instruction is determined by the value of the least significant bits in this register. However, the pointer for the MVI A, [expr] register is always located in the current SRAM page. See the *PSoC Designer Assembly Language User Guide* for more information on the MVI A, [expr] instruction.

The function of this register and the MVI instructions are independent of the SRAM Paging bits in the CPU_F register.

For additional information, refer to the MVR_PP register on page 159.

3.3.3 MVW_PP Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,D5h	MVW_PP						Page Bits[2:0]			RW : 00

The MVI Write Page Pointer Register (MVW_PP) is used to set the effective SRAM page for MVI write memory accesses in a multi-SRAM page PSoC device.

Note This register is only used when a device has more than one page of SRAM.

Bits 2 to 0: Page Bits[2:0]. This register is only used by the MVI [expr], A instruction, not to be confused with the MVI A, [expr] instruction covered by the MVR_PP register. This instruction is considered a write because data is transferred from the microprocessor's A register (CPU_A) to SRAM.

When an MVI [expr], A instruction is executed in a device with more than one page of SRAM, the SRAM address that is written by the instruction is determined by the value of the least significant bits in this register. However, the pointer for the MVI [expr], A register is always located in the current SRAM page. See the *PSoC Designer Assembly Language User Guide* for more information on the MVI [expr], A instruction.

The function of this register and the MVI instructions are independent of the SRAM Paging bits in the CPU_F register.

For additional information, refer to the MVW_PP register on page 160.

3.3.4 CPU_SCR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW *	ECO EX *		IRAMDIS	# : 00

LEGEND
x An “x” before the comma in the address field indicates that this register can be read or written to no matter what bank is used.
Access is bit specific. Refer to the Register Details chapter on page 103 for additional information.
* Bits 3 and 2 (ECO EXW and ECO EX, respectively) cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24x23 and CY8C22x13 PSoC devices.

The System Status and Control Register 1 (CPU_SCR1) is used to convey the status and control of events related to internal resets and watchdog reset.

Bit 7: IRESS. The Internal Reset Status bit is a read only bit that can be used to determine if the booting process occurred more than once.

When this bit is set, it indicates that the SROM SWBootReset code was executed more than once. If this bit is not set, the SWBootReset was executed only once. In either case, the SWBootReset code will not allow execution from code stored in Flash until the M8C Core is in a safe operating mode with respect to supply voltage and Flash operation. There is no need for concern when this bit is set. It is provided for systems which may be sensitive to boot time, so that they can determine if the normal one-pass boot time was exceeded.

Bit 4: SLIMO. When set, the Slow IMO bit allows the active power dissipation of the PSoC device to be reduced by slowing down the IMO from 24 MHz to 6 MHz. The IMO trim value must also be changed when SLIMO is set (see “Engaging Slow IMO” on page 76). When not in external clocking mode, the IMO is the source for SYSCLK; therefore, when the speed of the IMO changes, so will SYSCLK.

Bit 3: ECO EXW. The ECO Exists Written bit is used as a status bit to indicate that the ECO EX bit has been previously written to. It is read only. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24x23 and CY8C22x13 PSoC devices.

Bit 2: ECO EX. The ECO Exists bit serves as a flag to the hardware, to indicate that an external crystal ***oscillator*** exists in the system. Just after boot, it may be written only once to a value of ‘1’ (crystal exists) or ‘0’ (crystal does not exist). If the bit is ‘0’, a switch-over to the ECO is locked out by hardware. If the bit is ‘1’, hardware allows the firmware to freely switch between the ECO and ILO. It should be written as early as possible after a ***Power On Reset (POR)*** or ***External Reset (XRES)*** event, where it is assumed that program execution integrity is high. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24x23 and CY8C22x13 PSoC devices.

Bit 0: IRAMDIS. The Initialize RAM Disable bit is a control bit that is readable and writeable. The ***default value*** for this bit is ‘0’, which indicates that the maximum amount of SRAM should be initialized on watchdog reset to a value of 00h. When the bit is ‘1’, the minimum amount of SRAM is initialized after a watchdog reset. For more information on this bit, see the “SROM Function Descriptions” on page 44.

For additional information, refer to the CPU_SCR1 register on page 190.

3.3.5 FLS_PR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,FAh	FLS_PR1							Bank[1:0]		RW : 00

The Flash Program Register 1 (FLS_PR1) is used to specify which Flash bank should be used for SROM operations.

Note This register has no effect on products with one Flash bank. Refer to the table titled "Flash Memory Organization" on page 45 to determine the number of Flash banks in PSoC devices.

Bits 1 and 0: Bank[1:0]. The Bank bits in this register indicate which Flash bank the SROM Flash functions should operate on. The default value for the Bank bit is zero. Flash bank 0 holds up to the first 8K of user code, as well as the cal table. Note that the CY8C27x43 PSoC device holds 16K in bank 0. The optional Flash banks 1, 2, and 3 hold additional user code.

For additional information, refer to the FLS_PR1 register on page 265.

3.4 Clocking

Successful programming and erase operations, on the Flash, require that the CLOCK and DELAY parameters be set correctly. To determine the proper value for the DELAY parameter only, the CPU speed must be considered. However, three factors should be used to determine the proper value for CLOCK: operating temperature, CPU speed, and characteristics of the individual device. Equations and additional information on calculating the DELAY and CLOCK values follow.

3.4.1 DELAY Parameter

To determine the proper value for the DELAY parameter, the CPU speed during the Flash operation must be considered. Equation 1 displays the equation for calculating DELAY based on a CPU speed value. In this equation the units for CPU are hertz (Hz).

$$DELAY = \frac{100 \times 10^{-6} \cdot CPU - 80}{13}, \quad 3MHz \le CPU \le 12MHz$$

Equation 1

Equation 2 shows the calculation of the DELAY value for a CPU speed of 12 MHz. The numerical result of this calculation should be rounded to the nearest whole number. In the case of a 12 MHz CPU speed, the correct value for DELAY is 86 (0x56).

$$DELAY = \frac{100 \times 10^{-6} \cdot 12 \times 10^{6} - 80}{13}$$

Equation 2

3.4.2 CLOCK Parameter

The CLOCK parameter must be calculated using different equations for erase and write operations. The erase value for CLOCK must be calculated first. In Equation 3, the erase CLOCK value is indicated by a subscript E after the word CLOCK and the write CLOCK value is indicated by a subscript W after the word CLOCK.

Before either CLOCK value can be calculated, the values for M, B, and Mult must be determined. These are device specific values that are stored in the Flash table 3 and are accessed by way of the TableRead SROM function (see the "TableRead Function" on page 46). If the operating temperature is at or below 0°C, the cold values should be used. For operating temperatures at or above 0°C, the hot values should be used. See Table 3-11 for more information. Equations for calculating the correct value of CLOCK for write operations are first introduced with the assumption that the CPU speed is 12 MHz.

The equation for calculating the CLOCK value for an erase Flash operation is shown in Equation 3. In this equation the T has units of °C.

$$CLOCK_E = B - \frac{2M \cdot T}{256}$$

Equation 3

Using the correct values for B, M, and T, in the equation above, is required to achieve the endurance specifications of the Flash. However, for device programmers, where this calculation may be difficult to perform, the equation can be simplified by setting T to 0°C and using the hot value for B and M. This simplification is acceptable only if the total number of erase write cycles are kept to less than 10 and the operation is performed near room temperature. When T is set to 0, Equation 3 simplifies to the following.

$$CLOCK_E = B$$

Equation 4

Once a value for the erase CLOCK value has been determined, the write CLOCK value can be calculated. The equation to calculate the CLOCK value for a write is as follows.

$$CLOCK_W = \frac{CLOCK_E \cdot Mult}{64}$$

Equation 5

In the equation above, the correct value for Mult must be determined, based on temperature, in the same way that the B and M values were determined for Equation 3.

4. RAM Paging

This chapter explains the PSoC device's use of RAM Paging and its associated registers. For a complete table of the RAM Paging registers, refer to the "Summary Table of the Core Registers" on page 30. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

4.1 Architectural Description

The M8C is an 8-bit CPU with an 8-bit address bus. The 8-bit memory address bus allows the M8C to access up to 256 bytes of SRAM, to increase the amount of available SRAM and preserve the M8C ***assembly*** language. PSoC devices with more than 256 bytes of SRAM have a paged memory architecture.

Table 4-1. PSoC Device SRAM Availability

PSoC Device	Amount of SRAM	Number of Pages
CY8C29x66 CY8CPLC20 CY8CLED16P01 CY8CNP1xx	2 KB	8 Pages
CY8C27x43	256 Bytes	1 Page
CY8C24x94	1 KB	4 Pages
CY8C24x23	256 Bytes	1 Page
CY8C24x23A	256 Bytes	1 Page
CY8C22x13	256 Bytes	1 Page
CY8C21x34	512 Bytes	2 Pages
CY8C21x34B	512 Bytes	2 Pages
CY8C21x23	256 Bytes	1 Page
CY7C64215	1 KB	4 Pages
CY7C603xx	512 Bytes	2 Pages
CYWUSB6953	512 Bytes	2 Pages

To take full advantage of the paged memory architecture of the PSoC device, several registers must be used and two CPU_F register bits must be managed. However, the Power On Reset (POR) value for all of the paging registers and CPU_F bits is zero. This places the PSoC device in a mode identical to PSoC devices with only 256 bytes of SRAM. It is not necessary to understand all of the Paging registers to take advantage of the additional SRAM available in some devices. Very simple modifications to the reset state of the memory paging logic can be made, to begin to take advantage of the additional SRAM pages.

The memory paging architecture consists of five areas:

- Stack Operations
- Interrupts
- MVI Instructions
- Current Page Pointer
- Indexed Memory Page Pointer

The first three of these areas have no dependency on the CPU_F register's PgMode bits and are covered in the next subsections after Basic Paging. The function of the last two depend on the CPU_F PgMode bits and will be covered last.

4.1.1 Basic Paging

The M8C is an 8-bit CPU with an 8-bit memory address bus. The memory address bus allows the M8C to access up to 256 bytes of SRAM. To increase the amount of SRAM, the M8C accesses memory page bits. The memory page bits are located in the CUR_PP register and allow for selection of one of eight SRAM pages. In addition to setting the page bits, Page mode must be enabled by setting the CPU_F[7] bit. If Page mode is not enabled, the page bits are ignored and all non-stack memory access is directed to Page 0.

Once Page mode is enabled and the page bits are set, all instructions that operate on memory access the SRAM page indicated by the page bits. The exceptions to this are the instructions that operate on the stack and the MVI instructions: PUSH, POP, LCALL, RETI, RET, CALL, and MVI. See the description of Stack Operations and MVI Instructions below for a more detailed discussion.

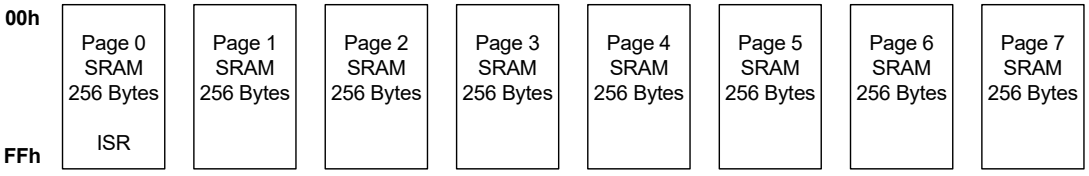


Figure 4-1. Data Memory Organization

4.1.2 Stack Operations

As mentioned previously, the paging architecture's reset state puts the PSoC in a mode that is identical to that of a 256 byte PSoC device. Therefore, upon reset, all memory accesses will be to Page 0. The SRAM page that stack operations will use is determined by the value of the three least significant bits of the stack page pointer register (STK_PP). Stack operations have no dependency on the PgMode bits in the CPU_F register. Stack operations are those that use the Stack Pointer (SP) to calculate their affected address. Refer to the *PSoC Designer Assembly Language User Guide* for more information on all M8C instructions.

Stack memory accesses must be treated as a special case. If they are not, the stack could be fragmented across several pages. To prevent the stack from becoming fragmented, all instructions that operate on the stack automatically use the page indicated by the STK_PP register. Therefore, if a CALL is encountered in the program, the PSoC device will automatically push the program counter onto the stack page indicated by STK_PP. Once the program counter is pushed, the SRAM paging mode automatically switches back to the pre-call mode. All other stack operations, such as RET and POP, follow the same rule as CALL. The stack is confined to a single SRAM page and the Stack Pointer will wrap from 00h to FFh and FFh to 00h. The user code must ensure that the stack is not damaged due to stack wrapping.

Because the value of the STK_PP register can be changed at any time, it is theoretically possible to manage the stack in such a way as to allow it to grow beyond one SRAM page or manage multiple stacks. However, the only supported use of the STK_PP register is when its value is set prior to the first stack operation and not changed again.

4.1.3 Interrupts

Interrupts, in a multi-page SRAM PSoC device, operate the same as interrupts in a 256 byte PSoC device. However, because the CPU_F register is automatically set to 0x00 on an interrupt and because of the non-linear nature of interrupts in a system, other parts of the PSoC memory paging architecture can be affected.

Interrupts are an abrupt change in program flow. If no special action is taken on interrupts by the PSoC device, the ***interrupt service routine (ISR)*** could be thrown into any SRAM page. To prevent this problem, the special addressing modes for all memory accesses, except for stack and MVI, are disabled when an ISR is entered. The special addressing modes are disabled when the CPU_F register is cleared. At the end of the ISR, the previous SRAM addressing mode is restored when the CPU_F register value is restored by the RETI instruction.

Therefore, all interrupt service ***routine*** code will start execution in SRAM Page 0. If it is necessary for the ISR to change to another SRAM page, it can be accomplished by changing the values of the CPU_F[7:6] bits to enable the special SRAM addressing modes. However, any change made to the CUR_PP, IDX_PP, or STK_PP registers will persist after the ISR returns. Therefore, the ISR should save the current value of any paging register it modifies and restore its value before the ISR returns.

4.1.4 MVI Instructions

MVI instructions use data page pointers of their own (MVR_PP and MVW_PP). This allows a data buffer to be located away from other program variables, but accessible without changing the Current Page Pointer (CUR_PP).

An MVI instruction performs three memory operations. Both forms of the MVI instruction access an address in SRAM that holds the data pointer (a memory read 1st access), incrementing that value and then storing it back in SRAM (a memory write 2nd access). This pointer value must reside in the current page, just as all other non-stack and non-indexed operations on memory must. However, the third memory operation uses the MVx_PP register. This third memory access can be either a read or a write, depending on which MVI instruction is used. The MVR_PP pointer is used for the MVI instruction that moves data into the accumulator. The MVW_PP pointer is used for the MVI instruction that moves data from the accumulator into SRAM. The MVI pointers are always enabled, regardless of the state of the Flag register page bits (CPU_F register).

4.1.5 Current Page Pointer

The Current Page Pointer is used to determine which SRAM page should be used for all memory accesses. Normal memory accesses are those not covered by other pointers including all non-stack, non-MVI, and non-indexed memory access instructions. The normal memory access instructions have the SRAM page they operate on determined by the value of the CUR_PP register. By default, the CUR_PP register has no affect on the SRAM page that will be used for normal memory access, because all normal memory access is forced to SRAM Page 0.

The upper bit of the PgMode bits in the CPU_F register determine whether or not the CUR_PP register affects normal memory access. When the upper bit of the PgMode bits is set to ‘0’, all normal memory access is forced to SRAM Page 0. This mode is automatically enabled when an Interrupt Service Routine (ISR) is entered. This is because, before the ISR is entered, the M8C pushes the current value of the CPU_F register onto the stack and then clears the CPU_F register. Therefore, by default, any normal memory access in an ISR is guaranteed to occur in SRAM Page 0.

When the RETI instruction is executed, to end the ISR, the previous value of the CPU_F register is restored, restoring the previous page mode. Note that this ISR behavior is the default and that the PgMode bits in the CPU_F register can

be changed while in an ISR. If the PgMode bits are changed while in an ISR, the pre-ISR value is still restored by the RETI; but if the CUR_PP register is changed in the ISR, the ISR is also required to restore the value before executing the RETI instruction.

When the upper bit of the PgMode bits is set to '1', all normal memory access is forced to the SRAM page indicated by the value of the CUR_PP register. Table 4-2 gives a summary of the PgMode bit values and the corresponding Memory Paging mode.

4.1.6 Index Memory Page Pointer

The source indexed and destination indexed addressing modes to SRAM are treated as a unique addressing mode in a PSoC device, with more than one page of SRAM. An example of an indexed addressing mode is the MOV A, [X+expr] instruction. Note that register access also has indexed addressing; however, those instructions are not affected by the SRAM paging architecture.

Important Note If you are not using assembly to program a PSoC device, be aware that the ***compiler*** writer may restrict the use of some memory paging modes. Review the conventions in your compiler's user guide for more information on restrictions or conventions associated with memory paging modes.

Indexed SRAM accesses operate in one of three modes:

- Index memory access modes are forced to SRAM Page 0.
- Index memory access modes are directed to the SRAM page indicated by the value in the STK_PP register.
- Index memory access is forced to the SRAM page indicated by the value in the IDX_PP register.

The mode is determined by the value of the PgMode bits in the CPU_F register. However, the final SRAM page that is used also requires setting either the Stack Page Pointer (STK_PP) register or the Index Page Pointer (IDX_PP) register. Table 4-2 shows the three indexed memory access modes. The third column of the table is provided for reference only.

Table 4-2. CPU_F PgMode Bit Modes

CPU_F PgMode Bits	Current SRAM Page	Indexed SRAM Page	Typical Use
00b	0	0	ISR*
01b	0	STK_PP	ISR with variables on stack
10b	CUR_PP	IDX_PP	
11b	CUR_PP	STK_PP	

* Mode used by SROM functions initiated by SSC instruction.

After reset, the PgMode bits are set to 00b. In this mode, index memory accesses are forced to SRAM Page 0, just as they would be in a PSoC device with only 256 bytes of SRAM. This mode is also automatically enabled when an interrupt occurs in a PSoC device and is therefore considered the default ISR mode. This is because before the ISR is entered, the M8C pushes the current value of the CPU_F register on to the stack and then clears the CPU_F register. Therefore, by default, any indexed memory access in an ISR is guaranteed to occur in SRAM Page 0. When the RETI instruction is executed to end the ISR, the previous value of the CPU_F register is restored and the previous page mode is then also restored. Note that this ISR behavior is the default and that the PgMode bits in the CPU_F register may be changed while in an ISR. If the PgMode bits are changed while in an ISR, the pre-ISR value is still restored by the RETI; but if the STK_PP or IDX_PP registers are changed in the ISR, the ISR is also required to restore the values before executing the RETI instruction.

The most likely PgMode bit change, while in an ISR, is from the default value of 00b to 01b. In the 01b mode, indexed memory access is directed to the SRAM page indicated by the value of the STK_PP register. By using the PgMode, the value of the STK_PP register is not required to be modified. The STK_PP register is the register that determines which SRAM page the stack is located on. The 01b paging mode is intended to provide easy access to the stack, while in an ISR, by setting the CPU_X register (just X in the instruction format) equal to the value of SP using the MOV X, SP instruction.

The two previous paragraphs covered two of the three indexed memory access modes: STK_PP and forced to SRAM Page 0. Note, as shown in Table 4-2, that the STK_PP mode for indexed memory access is available under two PgMode settings. The 01b mode is intended for ISR use and the 11b mode is intended for non-ISR use. The third indexed memory access mode requires the PgMode bits to be set to 10b. In this mode indexed memory access is forced to the SRAM page indicated by the value of the IDX_PP register.

4.2 Register Definitions

The following registers are associated with RAM Paging and are listed in address order. The register descriptions have an associated register table showing the bit structure for that register. The bits in the tables that are grayed out are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of '0'. For a complete table of RAM Paging registers, refer to the "Summary Table of the Core Registers" on page 30.

4.2.1 TMP_DRx Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,6xh	TMP_DRx	Data[7:0]								RW : 00

LEGEND

x An 'x' before the comma in the address field indicates that this register can be read or written to no matter what bank is used. An "x" after the comma in the address field indicates that there are multiple instances of the register.

The Temporary Data Registers (TMP_DR0, TMP_DR1, TMP_DR2, and TMP_DR3) are used to enhance the performance in multiple SRAM page PSoC devices.

These registers have no pre-defined function (for example, the compiler and hardware do not use these registers) and exist for the user to use as desired.

Bits 7 to 0: Data[7:0]. Due to the paged SRAM architecture of PSoC devices with more than 256 bytes of SRAM, a value in SRAM may not always be accessible without first changing the current page. The TMP_DRx registers are readable and writable and are provided to improve the performance of multiple SRAM page PSoC devices, by supplying some register space for data that is always accessible.

For an expanded listing of the TMP_DRx registers, refer to the "Summary Table of the Core Registers" on page 30. For additional information, refer to the TMP_DRx register on page 136.

4.2.2 CUR_PP Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,D0h	CUR_PP						Page Bits[2:0]			RW : 00

The Current Page Pointer Register (CUR_PP) is used to set the effective SRAM page for normal memory accesses in a multi-SRAM page PSoC device.

Note This register is only used when a device has more than one page of SRAM. Refer to the table titled "PSoC Device SRAM Availability" on page 53 to determine the number of SRAM pages in PSoC devices.

Bits 2 to 0: Page Bits[2:0]. These bits affect the SRAM page that is accessed by an instruction when the CPU_F[7:0] bits have a value of either 10b or 11b. Source indexed and destination indexed addressing modes, as well as stack instructions, are never affected by the value of the CUR_PP register. (See the STK_PP and IDX_PP registers for more information.)

The source indirect post increment and destination indirect post increment addressing modes, better know as MVI, are only partially affected by the value of the CUR_PP register. For MVI instructions, the pointer address is in the SRAM page indicated by CUR_PP, but the address pointed to may be in another SRAM page. See the MVR_PP and MVW_PP register descriptions for more information.

For additional information, refer to the CUR_PP register on page 156.

4.2.3 STK_PP Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,D1h	STK_PP						Page Bits[2:0]			RW : 00

The Stack Page Pointer Register (STK_PP) is used to set the effective SRAM page for stack memory accesses in a multi-SRAM page PSoC device.

Note This register is only used when a device has more than one page of SRAM. Refer to the table titled "PSoC Device SRAM Availability" on page 53 to determine the number of SRAM pages in PSoC devices.

Bits 2 to 0: Page Bits[2:0]

These bits have the potential to affect two types of memory access.

The purpose of this register is to determine which SRAM page the stack will be stored on. In the reset state, this register's value will be 0x00 and the stack will therefore be in SRAM Page 0. However, if the STK_PP register value is changed, the next stack operation will occur on the SRAM page indicated by the new STK_PP value. Therefore, the value of this register should be set early in the program and never be changed. If the program changes the STK_PP value after the stack has grown, the program must ensure that the STK_PP value is restored when needed.

Note that the impact that the STK_PP register has on the stack is independent of the SRAM Paging bits in the CPU_F register.

The second type of memory accesses that the STK_PP register affects are indexed memory accesses when the CPU_F[7:6] bits are set to 11b. In this mode, source indexed and destination indexed memory accesses are directed to the stack SRAM page, rather than the SRAM page indicated by the IDX_PP register or SRAM Page 0.

For additional information, refer to the STK_PP register on page 157.

4.2.4 IDX_PP Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,D3h	IDX_PP						Page Bits[2:0]			RW : 00

The Index Page Pointer Register (IDX_PP) is used to set the effective SRAM page for indexed memory accesses in a multi-SRAM page PSoC device.

Note This register is only used when a device has more than one page of SRAM. Refer to the table titled "PSoC Device SRAM Availability" on page 53 to determine the number of SRAM pages in PSoC devices.

Bits 2 to 0: Page Bits[2:0].

These bits allow instructions, which use the source indexed and destination indexed address modes, to operate on an SRAM page that is not equal to the current SRAM page. However, the effect this register has on indexed addressing modes is only enabled when the CPU_F[7:6] is set to 10b.

When CPU_F[7:6] is set to 10b and an indexed memory access is made, the access is directed to the SRAM page indicated by the value of the IDX_PP register.

See the STK_PP register description for more information on other indexed memory access modes.

For additional information, refer to the IDX_PP register on page 158.

4.2.5 MVR_PP Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,D4h	MVR_PP						Page Bits[2:0]			RW : 00

The MVI Read Page Pointer Register (MVR_PP) is used to set the effective SRAM page for MVI read memory accesses in a multi-SRAM page PSoC device.

Note This register is only used when a device has more than one page of SRAM. Refer to the table titled "PSoC Device SRAM Availability" on page 53 to determine the number of SRAM pages in PSoC devices.

Bits 2 to 0: Page Bits[2:0]. These bits are only used by the MVI A, [expr] instruction, not to be confused with the MVI [expr], A instruction covered by the MVW_PP register. This instruction is considered a read because data is transferred from SRAM to the microprocessor's A register (CPU_A).

When an MVI A, [expr] instruction is executed in a device with more than one page of SRAM, the SRAM address that is read by the instruction is determined by the value of the least significant bits in this register. However, the pointer for the MVI A, [expr] instruction is always located in the current SRAM page. See the *PSoC Designer Assembly Language User Guide* for more information on the MVI A, [expr] instruction.

The function of this register and the MVI instructions are independent of the SRAM Paging bits in the CPU_F register. For additional information, refer to the MVR_PP register on page 159.

4.2.6 MVW_PP Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,D5h	MVW_PP						Page Bits[2:0]			RW : 00

The MVI Write Page Pointer Register (MVW_PP) is used to set the effective SRAM page for MVI write memory accesses in a multi-SRAM page PSoC device.

Note This register is only used when a device has more than one page of SRAM. Refer to the table titled "PSoC Device SRAM Availability" on page 53 to determine the number of SRAM pages in PSoC devices.

Bits 2 to 0: Page Bits[2:0]. These bits are only used by the MVI [expr], A instruction, not to be confused with the MVI A, [expr] instruction covered by the MVR_PP register. This instruction is considered a write because data is transferred from the microprocessor's A register (CPU_A) to SRAM.

When an MVI [expr], A instruction is executed in a device with more than one page of SRAM, the SRAM address that is written by the instruction is determined by the value of the least significant bits in this register. However, the pointer for the MVI [expr], A instruction is always located in the current SRAM page. See the *PSoC Designer Assembly Language User Guide* for more information on the MVI [expr], A instruction.

The function of this register and the MVI instructions are independent of the SRAM Paging bits in the CPU_F register. For additional information, refer to the MVW_PP register on page 160.

4.2.7 CPU_F Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,F7h	CPU_F	PgMode[1:0]			XIO		Carry	Zero	GIE	RL : 02

LEGEND

L The AND F, expr; OR F, expr; and XOR F, expr flag instructions can be used to modify this register.
x An 'x' before the comma in the address field indicates that this register can be read or written to no matter what bank is used.

The M8C Flag Register (CPU_F) provides read access to the M8C flags.

Bits 7 and 6: PgMode[1:0]. PgMode determines how the CUR_PP and IDX_PP registers are used in forming effective RAM addresses for Direct Address mode and Indexed Address mode operands.

Bit 4: XIO. The IO Bank Select bit, also know as the register bank select bit, is used to select the register bank that is active for a register read or write. This bit allows the PSoC device to have 512 8-bit registers and therefore, can be thought of as the ninth address bit for registers. The address space accessed when the XIO bit is set to '0' is called the ***user space***, while the address space accessed when the XIO bit is set to '1' is called the ***configuration space***.

Bit 2: Carry. The Carry Flag bit is set or cleared in response to the result of several instructions. It can also be manipulated by the flag-logic opcodes (for example, OR F, 4). See the *PSoC Designer Assembly Guide User Manual* for more details.

Bit 1: Zero. The Zero Flag bit is set or cleared in response to the result of several instructions. It can also be manipulated by the flag-logic opcodes (for example, OR F, 2). See the *PSoC Designer Assembly Guide User Manual* for more details.

Bit 0: GIE. The state of the Global Interrupt Enable bit determines whether interrupts (by way of the IRQ) will be recognized by the M8C. This bit is set or cleared by the user, using the flag-logic instructions (for example, OR F, 1). GIE is also cleared automatically by the M8C upon entering the interrupt service routine (ISR), after the flag byte has been stored on the stack, preventing nested interrupts. Note that the bit can be set in an ISR if desired.

For GIE=1, the M8C samples the IRQ input for each instruction. For GIE=0, the M8C ignores the IRQ. For additional information, refer to the CPU_F register on page 188.

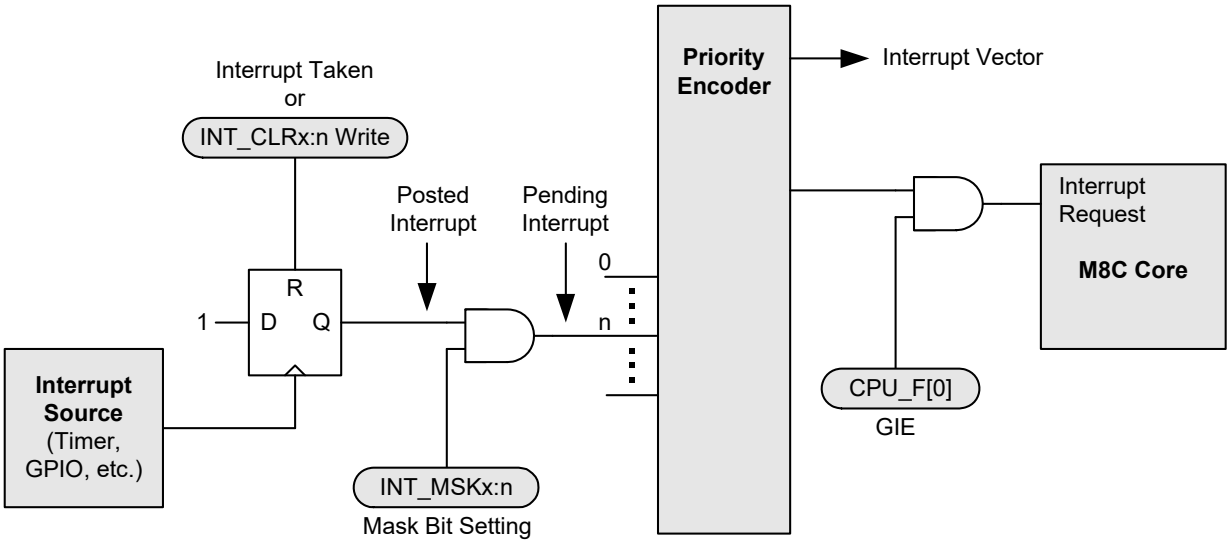
5. Interrupt Controller

This chapter presents the Interrupt Controller and its associated registers. The interrupt controller provides a mechanism for a hardware resource in PSoC Programmable System-on-Chip devices, to change program execution to a new address without regard to the current task being performed by the code being executed. For a complete table of the Interrupt Controller registers, refer to the "Summary Table of the Core Registers" on page 30. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

5.1 Architectural Description

A block diagram of the PSoC Interrupt Controller is shown in Figure 5-1, illustrating the concepts of ***posted interrupts*** and ***pending interrupts***.

Figure 5-1. Interrupt Controller Block Diagram



The sequence of events that occur during interrupt processing is as follows.

1. An interrupt becomes active, either because (a) the interrupt condition occurs (for example, a timer expires), (b) a previously posted interrupt is enabled through an update of an interrupt ***mask*** register, or (c) an interrupt is pending and GIE is set from '0' to '1' in the CPU Flag register.
2. The current executing instruction finishes.
3. The internal interrupt routine executes, taking 13 cycles. During this time, the following actions occur:
   - The PCH, PCL, and Flag register (CPU_F) are pushed onto the stack (in that order).
   - The CPU_F register is then cleared. Because this clears the GIE bit to 0, additional interrupts are temporarily disabled.
   - The PCH (PC[15:8]) is cleared to zero.
   - The interrupt vector is read from the interrupt controller and its value is placed into PCL (PC[7:0]). This sets the program counter to point to the appropriate address in the interrupt table (for example, 001Ch for the GPIO interrupt).
4. Program execution vectors to the interrupt table. Typically, a LJMP instruction in the interrupt table sends execution to the user's interrupt service routine (ISR) for this interrupt. (See "Instruction Set Summary" on page 34.)

5. The ISR executes. Note that interrupts are disabled because GIE = 0. In the ISR, interrupts can be re-enabled if desired, by setting GIE = 1 (take care to avoid stack overflow in this case).
6. The ISR ends with a RETI instruction. This pops the Flag register, PCL, and PCH from the stack, restoring those registers. The restored Flag register re-enables interrupts, because GIE = 1 again.
7. Execution resumes at the next instruction, after the one that occurred before the interrupt. However, if there are more pending interrupts, the subsequent interrupts will be processed before the next normal program instruction.

Interrupt Latency. The time between the assertion of an enabled interrupt and the start of its ISR can be calculated using the following equation:

$$\begin{aligned} Latency = \\ &Time\ for\ current\ instruction\ to\ finish + \\ &Time\ for\ M8C\ to\ change\ program\ counter\ to\ interrupt\ address + \\ &Time\ for\ LJMP\ instruction\ in\ interrupt\ table\ to\ execute. \end{aligned}$$

Equation 1

For example, if the 5-cycle JMP instruction is executing when an interrupt becomes active, the total number of CPU clock cycles before the ISR begins would be as follows:

$$\begin{aligned} &(1\ to\ 5\ cycles\ for\ JMP\ to\ finish) + \\ &(13\ cycles\ for\ interrupt\ routine) + \\ &(7\ cycles\ for\ LJMP) = 21\ to\ 25\ cycles. \end{aligned}$$

Equation 2

In the example above, at 24 MHz, 25 clock cycles take 1.042 µs.

Interrupt Priority. The priorities of the interrupts only come into consideration if more than one interrupt is pending during the same instruction cycle. In this case, the priority encoder (see Figure 5-1) generates an interrupt vector for the highest priority interrupt that is pending.

5.1.1 Posted versus Pending Interrupts

An interrupt is posted when its interrupt conditions occur. This results in the flip-flop in Figure 5-1 clocking in a ‘1’. The interrupt will remain posted until the interrupt is taken or until it is cleared by writing to the appropriate INT_CLRx register.

A posted interrupt is not pending unless it is enabled by setting its interrupt mask bit (in the appropriate INT_MSKx register). All pending interrupts are processed by the Priority Encoder to determine the highest priority interrupt which will be taken by the M8C if the Global Interrupt Enable bit is set in the CPU_F register.

Disabling an interrupt by clearing its interrupt mask bit (in the INT_MSKx register) does not clear a posted interrupt, nor does it prevent an interrupt from being posted. It simply prevents a posted interrupt from becoming pending.

It is especially important to understand the functionality of clearing posted interrupts, if the configuration of the PSoC device is changed by the application.

For example, if a digital PSoC block is configured as a counter and has posted an interrupt but is later reconfigured to a serial communications receiver, the posted interrupt from the counter will remain. Therefore, if the digital PSoC block's INT_MSKx bit is set after configuring the block as a serial communications receiver, a pending interrupt is generated immediately. To prevent the carryover of posted interrupts from one configuration to the next, the INT_CLRx registers should be used to clear posted interrupts prior to enabling the digital PSoC block.

5.2 Application Description

The interrupt controller and its associated registers allow the user's code to respond to an interrupt from almost every functional block in the PSoC devices. Interrupts for all the digital blocks and each of the analog columns are available, as well as interrupts for supply voltage, sleep, variable clocks, and a general GPIO (pin) interrupt.

The registers associated with the interrupt controller allow interrupts to be disabled either globally or individually. The registers also provide a mechanism by which a user can clear all pending and posted interrupts, or clear individual posted or pending interrupts. A software mechanism is provided to set individual interrupts. Setting an interrupt by way of software is very useful during code development, when one may not have the complete hardware system necessary to generate a real interrupt.

The following table lists the interrupts for all PSoC devices and the priorities that are available in each PSoC device.

Interrupt Priority	Interrupt Address	PSoC Devices CY8 –												Interrupt Name
		C29x66	C28xxx	C27x43	C24x94	C24x23	C24x23A	C24x33	C22x45	C22x13	C21345	C21x34	C21x23	
0 (Highest)	0000h									✓		✓	✓	Reset
1	0004h									✓		✓	✓	Supply Voltage Monitor
2	0008h											✓	✓	Analog Column 0
3	000Ch									✓		✓	✓	Analog Column 1
4	0010h													Analog Column 2
5	0014h													Analog Column 3
6	0018h									✓		✓	✓	VC3
7	001Ch									✓		✓	✓	GPIO
8	0020h									✓		✓	✓	PSoC Block DBC00
9	0024h									✓		✓	✓	PSoC Block DBC01
10	0028h									✓		✓	✓	PSoC Block DCC02
11	002Ch									✓		✓	✓	PSoC Block DCC03
12	0030h													PSoC Block DBC10
13	0034h													PSoC Block DBC11
14	0038h													PSoC Block DCC12
15	003Ch													PSoC Block DCC13
16	0040h													PSoC Block DBC20
17	0044h													PSoC Block DBC21
18	0048h													PSoC Block DCC22
19	004Ch													PSoC Block DCC23
20	0050h													PSoC Block DBC30
21	0054h													PSoC Block DBC31
22	0058h													PSoC Block DCC32
23	005Ch													PSoC Block DCC33
24	0060h									✓		✓	✓	1st I2C
25	0064h									✓		✓	✓	Sleep Timer
26	0068h													SAR ADC
27	006Ch													RTC
28	0070h													AE Column 0
29	0074h													AE Column 1
30	0078h													Reserved
31 (Lowest)	007Ch													Sleep Timer

5.3 Register Definitions

The following registers are associated with the Interrupt Controller and are listed in address order. The register descriptions have an associated register table showing the bit structure for that register. The bits in the tables that are grayed out are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of '0'. For a complete table of Interrupt Controller registers, refer to the “Summary Table of the Core Registers” on page 30.

Depending on the PSoC device you have, only certain bits are accessible to be read or written, such as the INT_CLR0 and INT_MSK0 registers that are analog column and digital row dependent. The analog column dependent registers have the column number listed to the right of the Address column. The digital row dependent registers are set up the same way, only with the term “Row” in the Address column. To determine your PSoC's characteristics, refer to the table titled “PSoC Device Characteristics” on page 21.

5.3.1 INT_CLRx Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,DAh CY8C22x45	INT_CLR0	VC3	Sleep	GPIO	Compare 1	Compare 0	Analog 1	Analog 0	V Monitor	RW : 00
0,DBh CY8C22x45	INT_CLR1	DCC13	DCC12	DBC11	DBC10	DCC03	DCC02	DBC01	DBC00	RW : 00
0,DCh CY8C22x45	INT_CLR2	RTC	CSD1	CSD0	SARADC					RW : 00
0,DDh CY8C22x45	INT_CLR3								I2C	RW : 00

The Interrupt Clear Registers (INT_CLRx) are used to enable the individual interrupt sources’ ability to clear posted interrupts.

There are four interrupt clear registers (INT_CLR0, INT_CLR1, INT_CLR2, and INT_CLR3) which may be referred to in general as INT_CLRx.The INT_CLRx registers are similar to the INT_MSKx registers in that they hold a bit for each interrupt source. Functionally the INT_CLRx registers are similar to the INT_VC register, although their operation is completely independent. When an INT_CLRx register is read, any bits that are set indicates an interrupt has been posted for that hardware resource. Therefore, reading these registers gives the user the ability to determine all posted interrupts.

The Enable Software Interrupt (ENSWINT) bit in INT_MSK3[7] determines the way an individual bit value written to an INT_CLR0 register is interpreted. When ENSWINT is cleared (the default state), writing 1's to an INT_CLRx register has no effect. However, writing 0's to an INT_CLRx register, when ENSWINT is cleared, will cause the corresponding interrupt to clear. If the ENSWINT bit is set, any 0's written to the INT_CLRx registers are ignored. However, 1's written to an INT_CLRx register, while ENSWINT is set, will cause an interrupt to post for the corresponding interrupt.

Note When using the INT_CLRx register to post an interrupt, the hardware interrupt source, such as a digital clock, must not have its interrupt output high. Therefore, it may be difficult to use software interrupts with interrupt sources that do not have enables such as VC3.

Software interrupts can aid in debugging interrupt service routines by eliminating the need to create system level interactions that are sometimes necessary to create a hardware-only interrupt.

5.3.1.1 INT_CLR0 Register

Depending on the analog column configuration of your PSoC device (see the table titled “PSoC Device Characteristics” on page 21), some bits may not be available in the INT_CLR0 register.

Bit 7: VC3. This bit allows posted VC3 interrupts to be read, cleared, or set.

Bit 6: Sleep. This bit allows posted sleep interrupts to be read, cleared, or set.

Bit 5: GPIO. This bit allows posted GPIO interrupts to be read, cleared, or set.

Bit 4: Compare 1. This bit allows posted compare column 1 interrupts to be read, cleared, or set.

Bit 3: Compare 0. This bit allows posted compare column 0 interrupts to be read, cleared, or set.

Bit 2: Analog 1. This bit allows posted analog column 1 interrupts to be read, cleared, or set.

Bit 1: Analog 0. This bit allows posted analog column 0 interrupts to be read, cleared, or set.

Bit 0: V Monitor. This bit allows posted V monitor interrupts to be read, cleared, or set.

For additional information, refer to the INT_CLR0 register on page 166.

5.3.1.2 INT_CLR1 Register

Depending on the digital row configuration of your PSoC device (see the table titled “PSoC Device Characteristics”

on page 21), some bits may not be available in the INT_-CLR1 register.

Bit 7: DCC13. This bit allows posted DCC13 interrupts to be read, cleared, or set for row 1 block 3.

Bit 6: DCC12. This bit allows posted DCC12 interrupts to be read, cleared, or set for row 1 block 2.

Bit 5: DBC11. This bit allows posted DBC11 interrupts to be read, cleared, or set for row 1 block 1.

Bit 4: DBC10. This bit allows posted DBC10 interrupts to be read, cleared, or set for row 1 block 0.

Bit 3: DCC03. This bit allows posted DCC03 interrupts to be read, cleared, or set for row 0 block 3.

Bit 2: DCC02. This bit allows posted DCC02 interrupts to be read, cleared, or set for row 0 block 2.

Bit 1: DBC01. This bit allows posted DBC01 interrupts to be read, cleared, or set for row 0 block 1.

Bit 0: DBC00. This bit allows posted DBC00 interrupts to be read, cleared, or set for row 0 block 0.

For additional information, refer to the INT_CLR1 register on page 168.

5.3.1.3 INT_CLR2 Register

Bit 7: RTC. This bit allows posted RTC interrupts to be read, cleared, or set.

Bit 6: CSD1. This bit allows posted CSD1 interrupts to be read, cleared, or set.

Bit 5: CSD0. This bit allows posted CSD0 interrupts to be read, cleared, or set.

Bit 4: SARADC. This bit allows posted SARADC interrupts to be read, cleared, or set.

For additional information, refer to the INT_CLR2 register on page 170.

5.3.1.4 INT_CLR3 Register

Bit 0: I2C. This bit allows posted I2C interrupts to be read, cleared, or set.

For additional information, refer to the INT_CLR3 register on page 171.

5.3.2 INT_MSKx Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,DEh CY8C22x45	INT_MSK3	ENSWINT							I2C	RW : 00
0,DFh CY8C22x45	INT_MSK2	RTC	CSD1	CSD0	SARADC					RW : 00
0,E0h CY8C22x45	INT_MSK0	VC3	Sleep	GPIO	Compare 1	Compare 0	Analog 1	Analog 0	V Monitor	RW : 00
0,E1h CY8C22x45	INT_MSK1	DCC13	DCC12	DBC11	DBC10	DCC03	DCC02	DBC01	DBC00	RW : 00

The Interrupt Mask Registers (INT_MSKx) are used to enable the individual interrupt sources' ability to create pending interrupts.

There are four interrupt ***mask*** registers (INT_MSK0, INT_MSK1, INT_MSK2, and INT_MSK3) which may be referred to in general as INT_MSKx. If cleared, each bit in an INT_MSKx register prevents a posted interrupt from becoming a pending interrupt (input to the priority encoder). However, an interrupt can still post even if its mask bit is zero. All INT_MSKx bits are independent of all other INT_MSKx bits.

If an INT_MSKx bit is set, the interrupt source associated with that mask bit may generate an interrupt that will become a pending interrupt. For example, if INT_MSK0[5] is set and at least one GPIO pin is configured to generate an interrupt, the interrupt controller will allow a GPIO interrupt request to post and become a pending interrupt for the M8C to respond to. If a higher priority interrupt is generated before the M8C responds to the GPIO interrupt, the higher priority interrupt will be responded to and not the GPIO interrupt.

Each interrupt source may require configuration at a block level. Refer to the other chapters in this manual for information on how to configure an individual interrupt source.

5.3.2.1 INT_MSK3 Register

Bit 7: ENSWINT. This bit is a special non-mask bit that controls the behavior of the INT_CLRx registers. See the INT_CLRx register in this section for more information.

Bit 0: I2C. This bit allows posted I2C interrupts to be read, masked, or set.

For additional information, refer to the INT_MSK3 register on page 172.

5.3.2.2 INT_MSK2 Register

Depending on the digital row characteristics of your PSoC device (see the table titled "PSoC Device Characteristics" on page 21), you may not be able to use this register. The bits in this register are only for PSoC devices with 4 and 3 digital rows.

Bit 7: RTC. This bit allows posted RTC interrupts to be read, masked, or set.

Bit 6: CSD1. This bit allows posted CSD1 interrupts to be read, masked, or set.

Bit 5: CSD0. This bit allows posted CSD0 interrupts to be read, masked, or set.

Bit 4: SARADC. This bit allows posted SARADC interrupts to be read, masked, or set.

For additional information, refer to the INT_MSK2 register on page 173.

5.3.2.3 INT_MSK0 Register

Depending on the analog column characteristics of your PSoC device (see the table titled "PSoC Device Characteristics" on page 21), some bits may not be available in the INT_MSK0 register.

Bit 7: VC3. This bit allows posted VC3 interrupts to be read, masked, or set.

Bit 6: Sleep. This bit allows posted sleep interrupts to be read, masked, or set.

Bit 5: GPIO. This bit allows posted GPIO interrupts to be read, masked, or set.

Bit 4: Compare 1. This bit allows posted compare column 1 interrupts to be read, masked, or set.

Bit 3: Compare 0. This bit allows posted compare column 0 interrupts to be read, masked, or set.

Bit 2: Analog 1. This bit allows posted analog column 1 interrupts to be read, masked, or set.

Bit 1: Analog 0. This bit allows posted analog column 0 interrupts to be read, masked, or set.

Bit 0: V Monitor. This bit allows posted V monitor interrupts to be read, masked, or set.

For additional information, refer to the INT_MSK0 register on page 174.

5.3.2.4 INT_MSK1 Register

Depending on the digital row characteristics of your PSoC device (see the table titled "PSoC Device Characteristics" on page 21), some bits may not be available in the INT_MSK1 register. The bits in this register are available for all PSoC devices, with the exception of one digital row devices.

Bit 7: DCC13. This bit allows posted DCC13 interrupts to be read, masked, or set for row 1 block 3.

Bit 6: DCC12. This bit allows posted DCC12 interrupts to be read, masked, or set for row 1 block 2.

Bit 5: DBC11. This bit allows posted DBC11 interrupts to be read, masked, or set for row 1 block 1.

Bit 4: DBC10. This bit allows posted DBC10 interrupts to be read, masked, or set for row 1 block 0.

Bit 3: DCC03. This bit allows posted DCC03 interrupts to be read, masked, or set for row 0 block 3.

Bit 2: DCC02. This bit allows posted DCC02 interrupts to be read, masked, or set for row 0 block 2.

Bit 1: DBC01. This bit allows posted DBC01 interrupts to be read, masked, or set for row 0 block 1.

Bit 0: DBC00. This bit allows posted DBC00 interrupts to be read, masked, or set for row 0 block 0.

For additional information, refer to the INT_MSK1 register on page 175.

5.3.3 INT_VC Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E2h	INT_VC	Pending Interrupt[7:0]								RC : 00

LEGEND
C Clearable register or bits.

The Interrupt Vector Clear Register (INT_VC) returns the next pending interrupt and clears all pending interrupts when written.

Bits 7 to 0: Pending Interrupt[7:0]. When the register is read, the ***least significant byte (LSB)***, of the highest priority pending interrupt, is returned. For example, if the GPIO and I2C interrupts were pending and the INT_VC register was read, the value 1Ch would be read. However, if no interrupt were pending, the value 00h would be returned. This is the reset vector in the interrupt table; however, reading 00h from the INT_VC register should not be considered an indication that a system reset is pending. Rather, reading 00h from the INT_VC register simply indicates that there are no pending interrupts. The highest priority interrupt, indicated by the value returned by a read of the INT_VC register, is removed from the list of pending interrupts when the M8C services an interrupt.

Reading the INT_VC register has limited usefulness. If interrupts are enabled, a read to the INT_VC register would not be able to determine that an interrupt was pending before the interrupt was actually taken. However, while in an interrupt, a user may wish to read the INT_VC register to see what the next interrupt will be. When the INT_VC register is written, with any value, all pending and posted interrupts are cleared by asserting the clear line for each interrupt.

For additional information, refer to the INT_VC register on page 176.

5.3.4 CPU_F Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,F7h	CPU_F	PgMode[1:0]			XIO		Carry	Zero	GIE	RL : 02

LEGEND
L The AND F, expr; OR F, expr; and XOR F, expr flag instructions can be used to modify this register.
x An "x" before the comma in the address field indicates that this register can be read or written to no matter what bank is used.

The M8C Flag Register (CPU_F) provides read access to the M8C flags. Note that only the GIE (Global Interrupt Enable) bit is related to the interrupt controller.

Bits 7 to 1. The CPU_F register holds bits that are used by different resources. For information on the other bits in this register, refer to the CPU Core (M8C) chapter on page 33.

Bit 0: GIE. The state of the Global Interrupt Enable bit determines whether interrupts (by way of the IRQ) will be recognized by the M8C. This bit is set or cleared by the user, using the flag-logic instructions (for example, OR F, 1). GIE is also cleared automatically by the M8C upon entering the interrupt service routine (ISR), after the flag byte has been stored on the stack, preventing nested interrupts. Note that the bit can be set in an ISR if desired.

For GIE=1, the M8C samples the IRQ input for each instruction. For GIE=0, the M8C ignores the IRQ.

For additional information, refer to the CPU_F register on page 188.

6. General Purpose I/O (GPIO)

This chapter discusses the General Purpose I/O (GPIO) and its associated registers, which is the circuit responsible for interfacing to the I/O pins of a PSoC device. The GPIO blocks provide the interface between the M8C core and the outside world. They offer a large number of configurations to support several types of ***input/output (I/O)*** operations for both digital and analog systems. For a complete table of the GPIO registers, refer to the "Summary Table of the Core Registers" on page 30. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

6.1 Architectural Description

The GPIO contains input buffers, output drivers, register bit storage, and configuration logic for connecting the PSoC device to the outside world.

I/O Ports are arranged with (up to) 8 bits per port. Each full port contains eight identical GPIO blocks, with connections to identify a unique address and register bit number for each block. Each GPIO block can be used for the following types of I/O:

- Digital I/O (digital input and output controlled by software)
- Global I/O (digital PSoC block input and output)
- Analog I/O (analog PSoC block input and output)

Each I/O pin also has several drive modes, as well as interrupt capabilities. While all GPIO pins are identical and provide digital I/O, some pins may not connect internally to analog functions.

The main block diagram for the GPIO block is shown in Figure 6-1. Note that some pins do not have all of the functionality shown, depending on internal connections.

The CY8C22x45, CY8C21345, CY8C21x34, CY7C603xx, and CYWUSB6953 PSoC devices contain an enhanced capability to connect any GPIO to an internal analog bus. This is described in detail in the I/O Analog Multiplexer chapter on page 422.

6.1.1 Digital I/O

One of the basic operations of the GPIO ports is to allow the M8C to send information out of the PSoC device and get information into the M8C from outside the PSoC device. This is accomplished by way of the port data register (PRTxDR). Writes from the M8C to the PRTxDR register store the data state, one bit per GPIO. In the standard non-bypass mode, the pin drivers drive the pin in response to this data bit, with a drive strength determined by the Drive mode setting (see Figure 6-1). The actual voltage on the pin depends on the Drive mode and the external ***load***.

The M8C can read the value of a port by reading the PRTxDR register address. When the M8C reads the PRTxDR register address, the current value of the pin voltage is translated into a logic value and returned to the M8C. Note that the pin voltage can represent a different logic value than the last value written to the PRTxDR register. This is an important distinction to remember in situations such as the use of a read modify write to a PRTxDR register. Examples of read modify write instructions include ***AND***, ***OR***, and ***XOR***.

The following is an example of how a read modify write, to a PRTxDR register, could have an unexpected and even indeterminate result in certain systems. Consider a scenario where all bits of Port 1 on the PSoC device are in the strong 1 resistive 0 drive mode; so that in some cases, the system the PSoC is in may pull up one of the bits.

```
mov   reg[PRT1DR], 0x00
or    reg[PRT1DR], 0x80
```

In the first line of code above, writing a 0x00 to the port will not affect any bits that happen to be driven by the system the PSoC is in. However, in the second line of code, it can not guarantee that only bit 7 will be the one set to a strong 1. Because the OR instruction will first read the port, any bits that are in the pull up state will be read as a '1'. These ones will then be written back to the port. When this happens, the pin will go in to a strong 1 state; therefore, if the pull up condition ends in the system, the PSoC will keep the pin value at a logic 1.

6.1.2 Global I/O

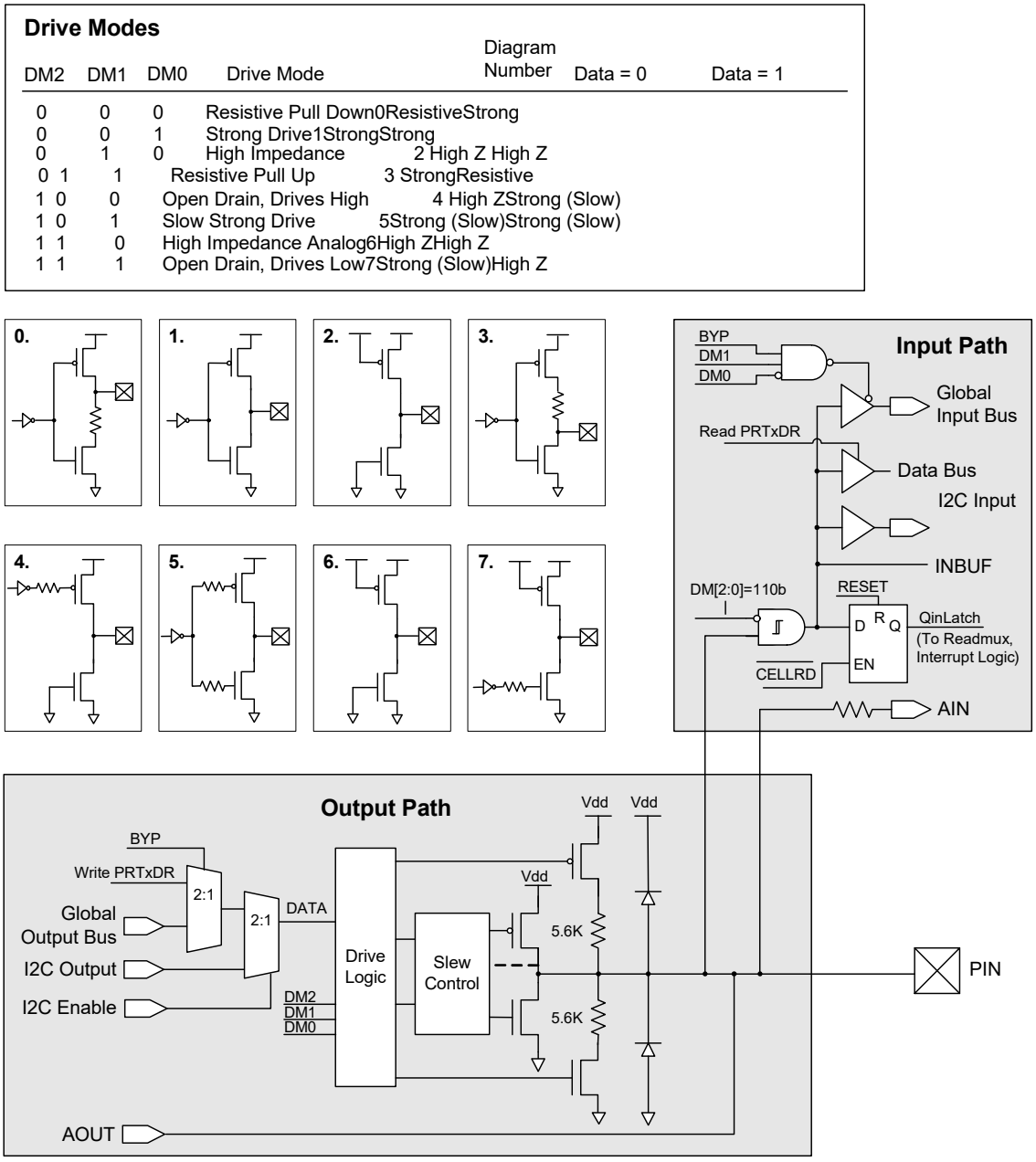
The GPIO ports are also used to interconnect signals to and from the digital PSoC blocks, as global inputs or outputs.

The global I/O feature of each GPIO (port pin) is off by default. To access the feature, two parameters must be changed. To configure a GPIO as a global input, the port global select bit must be set for the desired GPIO using the PRTxGS register. This sets BYP = 1 in Figure 6-1 and disconnects the output of the PRTxDR register from the pin. Also, the Drive mode for the GPIO must be set to the digital High-Z state. (Refer to the "PRTxDMx Registers" on page 73 for more information.) To configure a GPIO as a global output, the port global select bit must again be set. But in this case, the drive state must be set to any of the non-High-Z states.

6.1.3 Analog Input

Analog signals can pass into the PSoC device core from PSoC device pins through the block's AOUT pin. This provides a resistive ***path*** (~300 ohms) directly through the GPIO block. For analog modes, the GPIO block is typically configured into a High ***impedance*** Analog Drive mode (High Z). The mode turns off the Schmitt trigger on the input path, which may reduce power consumption and decrease internal switching noise when using a particular I/O as an analog input. Refer to the Electrical Specifications chapter in the individual PSoC device data sheet.

Figure 6-1. GPIO Block Diagram



6.1.4 GPIO Block Interrupts

Each GPIO block can be individually configured for interrupt capability. Blocks are configured by pin interrupt enables and also by selection of the interrupt state. Blocks can be set to interrupt when the pin is high, low, or when it changes from the last time it was read. The block provides an open-drain interrupt output (INTO) that is connected to other GPIO blocks in a wire-OR fashion.

All pin interrupts that are wire-OR'ed together are tied to the same system GPIO interrupt. Therefore, if interrupts are enabled on multiple pins, the user's interrupt service routine must provide a mechanism to determine which pin was the source of the interrupt.

Using a GPIO interrupt requires the following steps:
1. Set the Interrupt mode in the GPIO pin block.
2. Enable the bit interrupt in the GPIO block.
3. Set the mask bit for the (global) GPIO interrupt.
4. Assert the overall Global Interrupt Enable.

The first two steps, bit interrupt enable and Interrupt mode, are set at the GPIO block level (that is, at each port pin), by way of the block's configuration registers.

The last two steps are common to all interrupts and are described in the Interrupt Controller chapter on page 60.

At the GPIO block level, asserting the INTO line depends only on the bit interrupt enable and the state of the pin relative to the chosen Interrupt mode. At the PSoC device level, due to their wire-OR nature, the GPIO interrupts are neither true edge-sensitive interrupts nor true level-sensitive interrupts. They are considered edge-sensitive for asserting, but level-sensitive for release of the wire-OR interrupt line.

If no GPIO interrupts are asserting, a GPIO interrupt will occur whenever a GPIO pin interrupt enable is set and the GPIO pin transitions, if not already transitioned, appropriately high or low, to match the interrupt mode configuration. Once this happens, the INTO line will pull low to assert the GPIO interrupt. This assumes the other system-level enables are on, such as setting the global GPIO interrupt enable and the Global Interrupt Enable. Setting the pin interrupt enable may immediately assert INTO, if the Interrupt mode conditions are already being met at the pin.

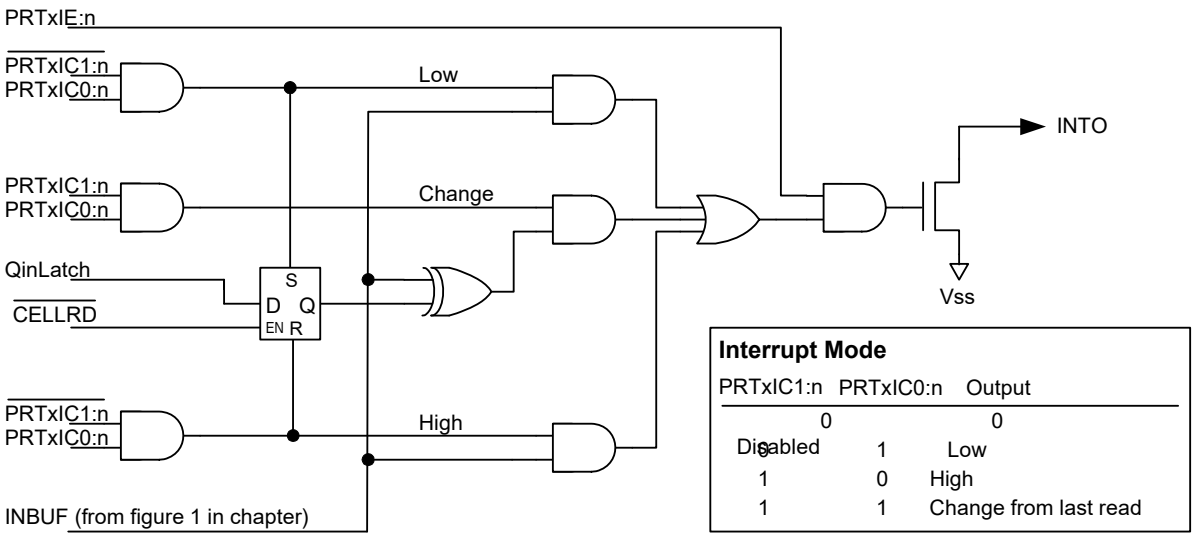
Once INTO pulls low, it will continue to hold INTO low until one of these conditions change: (a) the pin interrupt enable is cleared; (b) the voltage at pin transitions to the opposite state; (c) in interrupt-on-change mode, the GPIO data register is read, thus setting the local interrupt level to the opposite state; or (d) the Interrupt mode is changed so that the current pin state does not create an interrupt. Once one of these conditions is met, the INTO releases. At this point, another GPIO pin (or this pin again) could assert its INTO pin, pulling the common line low to assert a new interrupt.

Note that the GPIO data register state is latched during read operation. Interrupt-on-change may not behave as expected if the input signal changes during the metastability time of the latch; that is, when the GPIO is being read.

Note the following behavior from this level-release feature. If one pin is asserting INTO and then a second pin asserts its INTO, when the first pin releases its INTO, the second pin is already driving INTO and thus no change is seen (that is, no new interrupt would be asserted on the GPIO interrupt). Care must be taken, using polling or the states of the GPIO pin and Global Interrupt Enables, to catch all interrupts among a set of wire-OR GPIO blocks.

Figure 6-2 shows the interrupt logic portion of the block.

Figure 6-2. GPIO Interrupt Logic Diagram



6.2 Register Definitions

The following registers are associated with the General Purpose I/O (GPIO) and are listed in address order. The register descriptions in this section have an associated register table showing the bit structure for that register. For a complete table of GPIO registers, refer to the “Summary Table of the Core Registers” on page 30.

For a selected GPIO block, the individual registers are addressed in the Summary Table of the Core Registers. In the register names, the ‘x’ is the port number, configured at the PSoC device level (x = 0 to 7 typically). All register values are readable, except for the PRTxDR register; reads of this register return the pin state instead of the register bit state.

6.2.1 PRTxDR Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	PRTxDR	Data[7:0]								RW : 00

LEGEND
xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Core Register Summary” on page 30.

The Port Data Register (PRTxDR) allows for write or read access of the current logical equivalent of the voltage on the pin.

Note The CY8C22x45 and CY8C21345 have a 6-bit wide Port 4.

Bits 7 to 0: Data[7:0]. Writing the PRTxDR register bits set the output drive state for the pin to high (for DIN=1) or low (DIN=0), unless a bypass mode is selected (either I2C Enable=1 or the global select register written high).

Reading the PRTxDR register returns the actual pin state, as seen by the input buffer. This may not be the same as the expected output state, if the load pulls the pin more strongly than the pin’s configured output drive. See “Digital I/O” on page 68 for a detailed discussion of digital I/O.

For additional information, refer to the PRTxDR register on page 105.

6.2.2 PRTxIE Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	PRTxIE	Interrupt Enables[7:0]								RW : 00

LEGEND
xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Core Register Summary” on page 30.

The Port Interrupt Enable Register (PRTxIE) is used to enable/disable the interrupt enable internal to the GPIO block.

Note The CY8C22x45 and CY8C21345 have a 6-bit wide Port 4.

Bits 7 to 0: Interrupt Enables[7:0]. A ‘1’ enables the INTO output at the block and a ‘0’ disables INTO so it is only High-Z.

For additional information, refer to the PRTxIE register on page 106.

6.2.3 PRTxGS Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	PRTxGS	Global Select[7:0]								RW : 00

LEGEND
xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Core Register Summary” on page 30.

The Port Global Select Register (PRTxGS) is used to select the block for connection to global inputs or outputs.

Note The CY8C22x45 and CY8C21345 have a 6-bit wide Port 4.

Bits 7 to 0:Global Select[7:0]. Writing this register high enables the global bypass (BYP = 1 in Figure 6-1). If the Drive mode is set to digital High-Z (DM[2:0] = 010b), then the pin is selected for global input (PIN drives to the Global Input Bus). In non-High-Z modes, the block is selected for global output (the Global Output Bus drives to PIN), bypassing the data register value (assuming I2C Enable = 0).

If the PRTxGS register is written to zero, the global in/out function is disabled for the pin and the pin reflects the value of PRT_DR.

For additional information, refer to the PRTxGS register on page 107.

6.2.4 PRTxDMx Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	PRTxDM2	Drive Mode 2[7:0]								RW : FF
1,xxh	PRTxDM0	Drive Mode 0[7:0]								RW : 00
1,xxh	PRTxDM1	Drive Mode 1[7:0]								RW : FF

LEGEND
xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Core Register Summary” on page 30.

The Port Drive Mode Bit Registers (PRTxDMx) are used to specify the Drive mode for GPIO pins.

Note The CY8C22x45 and CY8C21345 have a 6-bit wide Port 4.

Bits 7 to 0: Drive Mode x[7:0]. In the PRTxDMx registers there are eight possible drive modes for each port pin. Three mode bits are required to select one of these modes, and these three bits are spread into three different registers (PRTxDM0, PRTxDM1, and PRTxDM2). The bit position of the effected port pin (for example, Pin[2] in Port 0) is the same as the bit position of each of the three drive mode register bits that control the Drive mode for that pin (for example, bit[2] in PRT0DM0, bit[2] in PRT0DM1, and bit[2] in PRT0DM2). The three bits from the three registers are treated as a group. These are referred to as DM2, DM1, and DM0, or together as DM[2:0]. Drive modes are shown in Table 6-1.

For analog I/O, the Drive mode should be set to one of the High-Z modes, either 010b or 110b. The 110b mode has the advantage that the block’s digital input buffer is disabled, so no ***crowbar*** current flows even when the analog input is not close to either power rail. When digital inputs are needed on the same pin as analog inputs, the 010b Drive mode should be used. If the 110b Drive mode is used, the pin will always be read as a zero by the CPU and the pin will not be able to generate a useful interrupt. (It is not strictly required that a High-Z mode be selected for analog operation.)

For global input modes, the Drive mode must be set to 010b.

Table 6-1. Pin Drive Modes

Drive Modes			Pin State	Description
DM2	DM1	DM0		
0	0	0	Resistive pull down	Strong high, resistive low
0	0	1	Strong drive	Strong high, strong low
0	1	0	High impedance	High-Z high and low, digital input enabled
0	1	1	Resistive pull up	Resistive high, strong low
1	0	0	Open drain high	Slow strong high, High-Z low
1	0	1	Slow strong drive	Slow strong high, slow strong low
1	1	0	High impedance, analog (**reset state**)	High-Z high and low, digital input disabled (for zero power) (**reset state**)
1	1	1	Open drain low	Slow strong low, High-Z high

The GPIO provides a default Drive mode of high impedance, analog (High-Z). This is achieved by forcing the reset state of all PRTxDM1 and PRTxDM2 registers to FFh.

The resistive drive modes place a ***resistance*** in series with the output, for low outputs (mode 000b) or high outputs (mode 011b). Strong Drive mode 001b gives the fastest edges at high DC drive strength. Mode 101b gives the same drive strength but with slower edges. The open-drain modes (100b and 111b) also use the slower edge rate drive. These modes enable open-drain functions such as I2C mode 111b (although the slow edge rate is not slow enough to meet the I2C fast mode specification).

For additional information, refer to the PRTxDM2 register on page 108, the PRTxDM0 register on page 192, and the PRTxDM1 register on page 193.

6.2.5 PRTxICx Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,xxh	PRTxIC0	Interrupt Control 0[7:0]								RW : 00
1,xxh	PRTxIC1	Interrupt Control 1[7:0]								RW : 00

LEGEND
xx An "x" after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the "Core Register Summary" on page 30.

The Port Interrupt Control Registers (PRTxIC1 and PRTxIC0) are used to specify the Interrupt mode for GPIO pins.

Note The CY8C22x45 and CY8C21345 have a 6-bit wide Port 4.

Bits 7 to 0: Interrupt Control x[7:0]. In the PRTxICx registers, the Interrupt mode for the pin is determined by bits in these two registers. These are referred to as IC1 and IC0, or together as IC[1:0].

There are four possible interrupt modes for each port pin. Two mode bits are required to select one of these modes and these two bits are spread into two different registers (PRTxIC0 and PRTxIC1). The bit position of the effected port pin (for example, Pin[2] in Port 0) is the same as the bit position of each of the interrupt control register bits that control the Interrupt mode for that pin (for example, bit[2] in PRT0IC0 and bit[2] in PRT0IC1). The two bits from the two registers are treated as a group.

The Interrupt mode must be set to one of the non-zero modes listed in Table 6-2, in order to get an interrupt from the pin.

The GPIO Interrupt mode "disabled" (00b) disables interrupts from the pin, even if the GPIO's bit interrupt enable is on (from the PRTxIE register).

Interrupt mode 01b means that the block will assert the interrupt line (INTO) when the pin voltage is low, providing the block's bit interrupt enable line is set (high).

Interrupt mode 10b means that the block will assert the interrupt line (INTO) when the pin voltage is high, providing the block's bit interrupt enable line is set (high).

Interrupt mode 11b means that the block will assert the interrupt line (INTO) when the pin voltage is the opposite of the last state read from the pin, providing the block's bit interrupt enable line is set high. This mode switches between low mode and high mode, depending on the last value that was read from the port during reads of the data register (PRTxDR). If the last value read from the GPIO was '0', the GPIO will subsequently be in Interrupt High mode. If the last value read from the GPIO was '1', the GPIO will then be in Interrupt Low mode.

Table 6-2. GPIO Interrupt Modes

Interrupt Modes		Description
IC1	IC0	
0	0	Bit interrupt disabled, INTO de-asserted
0	1	Assert INTO when PIN = low
1	0	Assert INTO when PIN = high
1	1	Assert INTO when PIN = change from last read

Figure 6-3. GPIO Interrupt Mode 11b

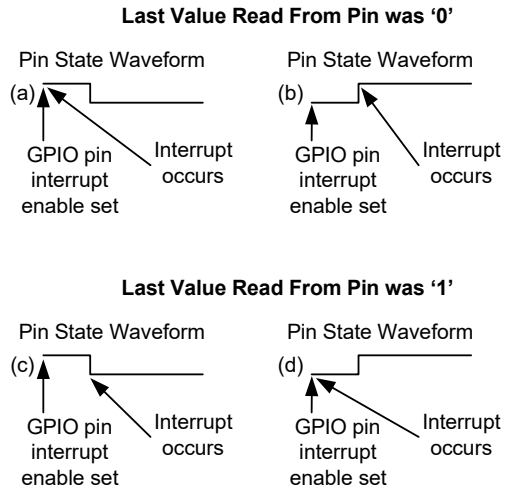


Figure 6-3 assumes that the GIE is set, GPIO interrupt mask is set, and that the GPIO Interrupt mode has been set to 11b. The Change Interrupt mode is different from the other modes, in that it relies on the value of the GPIO's read latch to determine if the pin state has changed. Therefore, the port that contains the GPIO in question must be read during every interrupt service routine. If the port is not read, the Interrupt mode will act as if it is in high mode when the latch value is '0' and low mode when the latch value is '1'.

For additional information, refer to the PRTxIC0 register on page 194 and the PRTxIC1 register on page 195.

7. Internal Main Oscillator (IMO)

This chapter presents the Internal Main Oscillator (IMO) and its associated registers. The IMO produces clock signals of 24 MHz and 48 MHz. For a complete table of the IMO registers, refer to the "Summary Table of the Core Registers" on page 30. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

7.1 Architectural Description

The Internal Main Oscillator (IMO) outputs two clocks: a SYSCLK, which can be the internal 24 MHz clock or an external clock, and a SYSCLKX2 that is always twice the SYSCLK frequency. In the absence of a high-precision input source from the 32.768 kHz ***crystal oscillator***, the accuracy of the internal 24/48 MHz clocks will be ±2.5% over temperature variation and two voltage ranges (3.3 V ± 0.3 V and 5.0 V ± 0.25%). No external components are required to achieve this level of accuracy.

There is an option to phase lock this oscillator to the External Crystal Oscillator (ECO). The choice of crystal and its inherent accuracy will determine the overall accuracy of the oscillator. The ECO must be stable prior to locking the frequency of the IMO to this reference source. Note that this ECO option is not available on the CY8C21x34, CY8C21x23, CY7C603xx, or CYWUSB6953 PSoC devices.

The ***frequency*** doubler circuit, which produces SYSCLKX2, can be disabled to save power.

On some PSoC devices (seeTable 7-1 showing check mark confirmation), lower frequency SYSCLK settings are available by setting the slow IMO (SLIMO) bit in the CPU_SCR1 register. With this bit set and the corresponding factory trim value applied to the IMO_TR register, SYSCLK can be lowered to 6 MHz. This offers lower device power consumption for systems that can operate with the reduced system clock. Slow IMO mode is discussed further in the "Application Description" on page 76.

7.2 PSoC Device Distinctions

In the CY8C27x43, CY8C24x23, CY8C22x13, CY7C603xx, and CYWUSB6953 PSoC devices, the Slow IMO mode (bit 4 in the CPU_SCR1 register on page 190) is reserved. In the table below, the slow IMO option is available for the following checked PSoC devices.

Table 7-1. Slow IMO (SLIMO) Option Availability

PSoC Device	Slow IMO Option
CY8C29x66	
CY8C27x43	
CY8C24x94	
CY8C24x23	
CY8C24x23A	✓
CY8C22x45	✓
CY8C21345	✓

7.3 Application Description

To save power, the IMO frequency can be reduced from 24 MHz to 6 MHz or 12 MHz using the SLIMO bit in the CPU_SCR1 register, in conjunction with the Trim values in the IMO_TR register. How to do this is described in the sections that follow.

7.3.1 Trimming the IMO

An 8-bit register (IMO_TR) is used to trim the IMO. Bit 0 is the LSB and bit 7 is the MSB. The trim step size is approximately 80 kHz.

A factory trim setting is loaded into the IMO_TR register at boot time for 5 V ± 0.25 V operation, except for the CY7C603xx, which is 3.3 V ± 0.25 V. For operation in the voltage ranges of 3.3 V ± 0.3 V and 2.7 V ± 0.3 V, user code must modify the contents of this register with values stored in Flash bank 0 as shown in Table 3-11 on page 47. This is done with a Table Read command to the Supervisory ROM.

7.3.2 Engaging Slow IMO

Forcing CPU_SCR1 register bit 4 high engages the Slow IMO feature. The IMO will immediately drop to a lower frequency. Factory trim settings are stored in Flash bank 0 as shown in Table 3-11 on page 47 for the following voltage/frequency combinations.

Voltage	Normal IMO Frequency	Slow IMO Frequency	Slow IMO Frequency
5.0 V ± 0.25 V	24 MHz	–	6 MHz
3.3 V ± 0.3 V	24 MHz	–	6 MHz
2.7 V ± 0.3 V	–	12 MHz	6 MHz

A Table Read command to the Supervisory ROM is performed to set the IMO to the different frequencies.

7.4 Register Definitions

The following registers are associated with the Internal Main Oscillator (IMO). The register descriptions have an associated register table showing the bit structure for that register. The bits in the tables that are grayed out are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of '0'. For a complete table showing all oscillator registers, refer to the "Summary Table of the Core Registers" on page 30.

7.4.1 CPU_SCR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW *	ECO EX *		IRAMDIS	# : 00

LEGEND
x An "x" before the comma in the address field indicates that this register can be read or written to no matter what bank is used.
Access is bit specific. Refer to the Register Details chapter on page 103 for additional information.
* Bits 3 and 2 (ECO EXW and ECO EX, respectively) cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24x23 and CY8C22x13 PSoC devices.

The System Status and Control Register 1 (CPU_SCR1) is used to convey the status and control of events related to internal resets and watchdog reset.

Bit 7: IRESS. The Internal Reset Status bit is a read only bit that may be used to determine if the booting process occurred more than once.

When this bit is set, it indicates that the SROM SWBootReset code was executed more than once. If this bit is not set, the SWBootReset was executed only once. In either case, the SWBootReset code will not allow execution from code stored in Flash until the M8C Core is in a safe operating mode with respect to supply voltage and Flash operation. There is no need for concern when this bit is set. It is provided for systems which may be sensitive to boot time, so that they can determine if the normal one-pass boot time was exceeded. For more information on the SWBootReest code see the Supervisory ROM (SROM) chapter on page 43.

Bit 4: SLIMO. When set, the Slow IMO bit allows the active power dissipation of the PSoC device to be reduced by slowing down the IMO from 24 MHz to 6 MHz. The IMO trim value must also be changed when SLIMO is set (see "Engaging Slow IMO" on page 76). When not in external clocking mode, the IMO is the source for SYSCLK; therefore, when the speed of the IMO changes, so will SYSCLK.

Bit 3: ECO EXW. The ECO Exists Written bit is used as a status bit to indicate that the ECO EX bit has been previously written to. It is read only. When this bit is a '1', this indicates that the CPU_SCR1 register has been written to and is now locked. When this bit is a '0', the register has not been written to since the last reset event. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24x23 and CY8C22x13 PSoC devices.

Bit 2: ECO EX. The ECO Exists bit serves as a flag to the hardware, to indicate that an external crystal ***oscillator*** exists in the system. Just after boot, it may be written *only once* to a value of '1' (crystal exists) or '0' (crystal does not exist). If the bit is '0', a switch-over to the ECO is locked out by hardware. If the bit is '1', hardware allows the firmware to freely switch between the ECO and ILO. It should be written as early as possible after a ***Power On Reset (POR)*** or ***External Reset (XRES)*** event, where it is assumed that program execution integrity is high. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24x23 and CY8C22x13 PSoC devices.

Bit 0: IRAMDIS. The Initialize RAM Disable bit is a control bit that is readable and writeable. The ***default value*** for this bit is '0', which indicates that the maximum amount of SRAM should be initialized on watchdog reset to a value of 00h. When the bit is '1', the minimum amount of SRAM is initialized after a watchdog reset. For more information on this bit, see the "SROM Function Descriptions" on page 44.

For additional information, refer to the CPU_SCR1 register on page 190.

7.4.2 OSC_CR2 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E2h	OSC_CR2	PLLGAIN			SLP_EXTEND	WDR32_SE	EXTCLKEN	RSVD	SYSCLKX2DIS	RW : 00

The Oscillator Control Register 2 (OSC_CR2) is used to configure various features of internal clock sources and clock nets.

Bit 7: PLLGAIN. This is the only bit in the OSC_CR2 register that directly influences the PLL. When set, this bit keeps the PLL in Low Gain mode. If this bit is held low, the lock time is less than 10 ms. If this bit is held high, the lock time is on the order of 50 ms. After lock is achieved, it is recommended that this bit be forced high to decrease the jitter on the output. If longer lock time is tolerable, the PLLGAIN bit can be held high all the time.

Bit 4: SLP_EXTEND. Long Sleep extension.

Bit 3: WDR32_SE. Watchdog clock source selection.

Bit 2: EXTCLKEN. When the EXTCLKEN bit is set, the external clock becomes the source for the internal clock tree, SYSCLK, which drives most PSoC device clocking functions. All external and internal signals, including the 32 kHz clock, whether derived from the Internal Low Speed Oscillator (ILO) or the crystal oscillator, are synchronized to this clock source. If an external clock is enabled, PLL mode should be off. The external clock input is located on port P1[4]. When using this input, the pin drive mode should be set to High-Z (not High-Z analog).

Bit 1: RSVD. This bit should always be 0.

Bit 0: SYSCLKX2DIS. When SYSCLKX2DIS is set, the IMO's doubler is disabled. This will result in a reduction of overall device power, on the order of 1 mA. It is advised that any application that does not require this doubled clock should have it turned off.

For additional information, refer to the OSC_CR2 register on page 256.

7.4.3 IMO_TR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E8h	IMO_TR	Trim[7:0]								W : 00

The Internal Main Oscillator Trim Register (IMO_TR) is used to manually center the oscillator's output to a target frequency.

The PSoC device specific value for 5 V operation is loaded into the Internal Main Oscillator Trim register (IMO_TR) at boot time. The Internal Main Oscillator will operate within specified tolerance over a voltage range of 4.75 V to 5.25 V, with no modification of this register. If the PSoC device is operated at a lower voltage, user code must modify the contents of this register. For operation in the voltage range of 3.3 V ± 0.3 V, this is accomplished with a Table Read command to the Supervisory ROM, which will supply a trim value for operation in this range. For operation between these voltage ranges, user code can interpolate the best value using both available factory trim values.

It is strongly recommended that the user not alter the register value, unless Slow IMO mode is used.

Bits 7 to 0: Trim[7:0]. These bits are used to trim the Internal Main Oscillator. A larger value in this register will increase the speed of the oscillator.

For additional information, refer to the IMO_TR register on page 261.

8. Internal Low Speed Oscillator

This chapter briefly explains the Internal Low Speed Oscillator (ILO) and its associated register. The Internal Low Speed Oscillator produces a 32 kHz clock. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

8.1 Architectural Description

The Internal Low Speed Oscillator (ILO) is an oscillator with a nominal frequency of 32 kHz. It is used to generate Sleep Wake-up interrupts and watchdog resets. This oscillator can also be used as a clocking source for the digital PSoC blocks.

The oscillator operates in three modes: normal power, low power, and off. The Normal Power mode consumes more current to produce a more accurate frequency. The Low Power mode is always used when the part is in a power down (sleep) state.

8.2 Register Definitions

The following register is associated with the Internal Low Speed Oscillator (ILO). The register description has an associated register table showing the bit structure. The bits in the table that are grayed out are reserved bits and are not detailed in the register description that follows. Note that reserved bits should always be written with a value of '0'.

8.2.1 ILO_TR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E9h	ILO_TR			Bias Trim[1:0]		Freq Trim[3:0]				W : 00

The Internal Low Speed Oscillator Trim Register (ILO_TR) sets the adjustment for the internal low speed oscillator.

The device specific value, placed in the trim bits of this register at boot time, is based on factory testing. ***It is strongly recommended that the user not alter the values in the register***.

Bits 5 and 4: Bias Trim[1:0]. These two bits are used to set the bias current in the PTAT Current Source. Bit 5 gets inverted, so that a medium bias is selected when both bits are '0'. The ***bias current*** is set according to Table 8-1.

Table 8-1. Bias Current in PTAT

Bias Current	Bias Trim [1:0]
Medium Bias	00b
Maximum Bias	01b
Minimum Bias	10b
Reserved	11b

Bits 3 to 0: Freq Trim[3:0]. These four bits are used to trim the frequency. Bit 0 is the LSb and bit 3 is the MSb. Bit 3 gets inverted inside the register.

For additional information, refer to the ILO_TR register on page 262.

9. External Crystal Oscillator

This chapter briefly explains the External Crystal Oscillator (ECO) and its associated registers. The 32.768 kHz external crystal oscillator circuit allows the user to replace the internal low speed oscillator with a more precise time source at low cost and low power. For a complete table of the External Crystal Oscillator registers, refer to the "Summary Table of the Core Registers" on page 30. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

9.1 Architectural Description

The External Crystal Oscillator (ECO) circuit uses an inexpensive watch crystal and two small value capacitors as external components, with all other components being on the PSoC device. The crystal oscillator may be configured to provide a reference to the Internal Main Oscillator (IMO) in PLL mode, for generating a 24 MHz system clock.

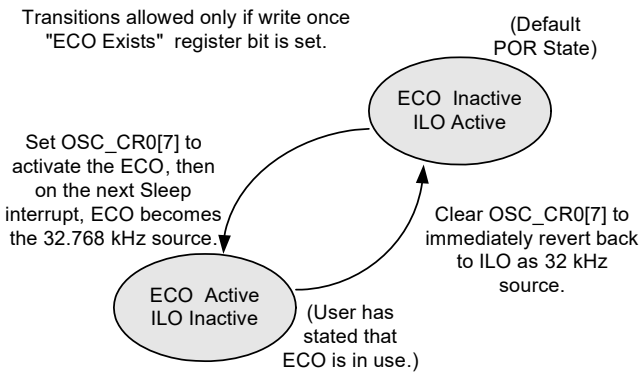
The XTALIn and XTALOut pins support connection of a 32.768 kHz watch crystal. To use the external crystal, bit 7 of the Oscillator Control 0 register (OSC_CR0) must be set (the default is off). The only external components needed are the crystal and the two capacitors that connect to Vdd. Note that transitions between the internal and external oscillator domains may produce glitches on the clock bus.

During the process of activating the ECO, there must be a hold-off period before using it as the 32.768 kHz source. This hold-off period is partially implemented in hardware using the sleep timer. Firmware must set up a sleep period of one second (maximum ECO ***settling time***), and then enable the ECO in the OSC_CR0 register. At the one second time-out (the sleep interrupt), the switch is made by hardware to the ECO. If the ECO is subsequently deactivated, the Internal Low Speed Oscillator (ILO) will again be activated and the switch is made back to the ILO immediately.

The ECO Exists bit (ECO EX, bit 2 in the CPU_SCR1 register) is used to control whether the switch-over is allowed or locked. This is a write once bit. It is written early in code execution after a Power On Reset (POR) or external reset (XRES) event. A '1' in this bit indicates to the hardware that a crystal exists in the system, and firmware is allowed to switch back and forth between ECO and ILO operation. If the bit is '0', switch-over to the ECO is locked out. The ECO Exists Written bit (ECO EXW, bit 3 in the CPU_SCR1 register) is read only and is set on the first write to this register. When this bit is '1', it indicates that the state of ECO EX is locked. This is illustrated in Figure 9-1.

Note Bits 3 and 2 (ECO EXW and ECO EX, respectively) in the CPU_SCR1 register cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24x23 and CY8C22x13 PSoC devices.

Figure 9-1. State Transition Between ECO and ILO Operation



The firmware steps involved in switching between the Internal Low Speed Oscillator (ILO) to the 32.768 kHz External Crystal Oscillator (ECO) are as follows.

1. At reset, the PSoC device begins operation, using the ILO.
2. Set the ECO EX bit to allow crystal operation.
3. Select a sleep interval of one second, using bits[4:3] in the Oscillator Control 0 register (OSC_CR0), as the oscillator stabilization interval.
4. Enable the ECO by setting bit [7] in Oscillator Control 0 register (OSC_CR0) to '1'.

5. The ECO becomes the selected source at the end of the one-second interval on the edge created by the Sleep Interrupt logic. The one-second interval gives the oscillator time to stabilize before it becomes the active source. The sleep interrupt need not be enabled for the switchover to occur. Reset the sleep timer (if this does not interfere with any ongoing real-time clock operation), to guarantee the interval length. Note that the ILO continues to run until the oscillator is automatically switched over by the sleep timer interrupt.
6. It is strongly advised to wait the one-second stabilization period prior to engaging the PLL mode to lock the IMO frequency to the ECO frequency.

Note 1 The ILO switches back instantaneously by writing the 32 kHz Select Control bit to '0'.

Note 2 If the proper settings are selected in PSoC Designer, the above steps are automatically done in *boot.asm*.

Note 3 Transitions between oscillator domains may produce glitches on the 32 kHz clock bus. Functions that require accuracy on the 32 kHz clock should be enabled after the transition in oscillator domains.

9.1.1 ECO External Components

The external component connections and selections of the External Crystal Oscillator are illustrated in Figure 9-2.

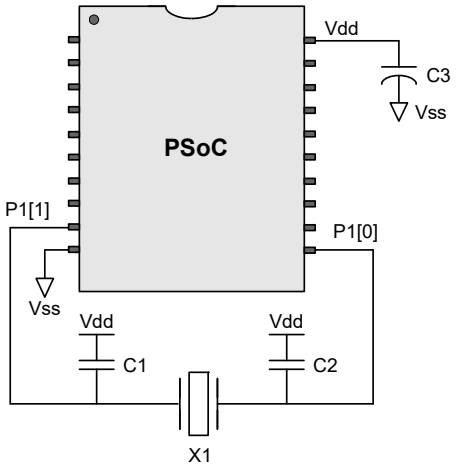
- Crystal – 32.768 kHz watch crystal such as Epson C-002RX.
- Capacitors – C1, C2 use NPO ceramic caps.

Use the equation below if you do not employ PLL mode.

$$C1 = C2 = 25\ pF - (Package\ Capacitance) - (Board\ Parasitic\ Capacitance)$$

An error of 1 pF in C1 and C2 gives about a 3 ppm error in frequency.

Figure 9-2. 20-Pin PSoC Example of the ECO External Connections



Refer to the PSoC device data sheet, in the packaging chapter, for typical package capacitances on crystal pins.

9.2 PSoC Device Distinctions

Bits 3 and 2 (ECO EXW and ECO EX, respectively) in the CPU_SCR1 register cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24x23 and CY8C22x13 PSoC devices.

9.3 Register Definitions

The following registers are associated with the External Crystal Oscillator and are listed in address order. Each register description has an associated register table showing the bit structure for that register. The bits that are grayed out in the tables below are reserved bits and are not detailed in the register descriptions. Note that reserved bits should always be written with a value of '0'. For a complete table of external crystal oscillator registers, refer to the "Summary Table of the Core Registers" on page 30.

9.3.1 CPU_SCR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW *	ECO EX *		IRAMDIS	# : 00

LEGEND
x An "x" before the comma in the address field indicates that this register can be read or written to no matter what bank is used.
Access is bit specific. Refer to the Register Details chapter on page 103 for additional information.
* Bits 3 and 2 (ECO EXW and ECO EX, respectively) cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24x23 and CY8C22x13 PSoC devices.

The System Status and Control Register 1 (CPU_SCR1) is used to convey the status and control of events related to internal resets and watchdog reset.

Bit 7: IRESS. The Internal Reset Status bit is a read only bit that may be used to determine if the booting process occurred more than once.

When this bit is set, it indicates that the SROM SWBootReset code was executed more than once. If this bit is not set, the SWBootReset was executed only once. In either case, the SWBootReset code will not allow execution from code stored in Flash until the M8C Core is in a safe operating mode with respect to supply voltage and Flash operation. There is no need for concern when this bit is set. It is provided for systems which may be sensitive to boot time, so that they can determine if the normal one-pass boot time was exceeded. For more information on the SWBootReest code see the Supervisory ROM (SROM) chapter on page 43.

Bit 4: SLIMO. When set, the Slow IMO bit allows the active power dissipation of the PSoC device to be reduced by slowing down the IMO from 24 MHz to 6 MHz. The IMO trim value must also be changed when SLIMO is set (see "Engaging Slow IMO" on page 76). When not in external clocking mode, the IMO is the source for SYSCLK; therefore, when the speed of the IMO changes, so will SYSCLK.

Bit 3: ECO EXW. The ECO Exists Written bit is used as a status bit to indicate that the ECO EX bit has been previously written to. It is read only. When this bit is a '1', this indicates that the CPU_SCR1 register has been written to and is now locked. When this bit is a '0', the register has not been written to since the last reset event. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24x23 and CY8C22x13 PSoC devices.

Bit 2: ECO EX. The ECO Exists bit serves as a flag to the hardware, to indicate that an external crystal ***oscillator*** exists in the system. Just after boot, it may be written *only once* to a value of '1' (crystal exists) or '0' (crystal does not exist). If the bit is '0', a switch-over to the ECO is locked out by hardware. If the bit is '1', hardware allows the firmware to freely switch between the ECO and ILO. It should be written as early as possible after a ***Power On Reset (POR)*** or ***External Reset (XRES)*** event, where it is assumed that program execution integrity is high. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24x23 and CY8C22x13 PSoC devices.

Bit 0: IRAMDIS. The Initialize RAM Disable bit is a control bit that is readable and writeable. The ***default value*** for this bit is '0', which indicates that the maximum amount of SRAM should be initialized on watchdog reset to a value of 00h. When the bit is '1', the minimum amount of SRAM is initialized after a watchdog reset. For more information on this bit, see the "SROM Function Descriptions" on page 44.

For additional information, refer to the CPU_SCR1 register on page 190.

9.3.2 OSC_CR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00

The Oscillator Control Register 0 (OSC_CR0) is used to configure various features of internal clock sources and clock nets.

Bit 7: 32k Select. By default, the 32 kHz clock source is the Internal Low Speed Oscillator (ILO). Optionally, the 32.768 kHz External Crystal Oscillator (ECO) may be selected.

Bit 6: PLL Mode. This is the only bit in the OSC_CR0 register that directly influences the Phase Locked Loop (PLL). When set, this bit enables the PLL. The EXTCLKEN bit in the OSC_CR2 register should be set low during PLL operation. For information on the PLL, refer to the Phase-Locked Loop (PLL) chapter on page 85.

Bit 5: No Buzz. Normally, when the Sleep bit is set in the CPU_SCR register, all PSoC device systems are powered down, including the bandgap reference. However, to facilitate the detection of ***POR*** and ***LVD*** events at a rate higher than the sleep interval, the bandgap circuit is powered up periodically for about 60 µs at the Sleep System Duty Cycle (set in ECO_TR), which is independent of the sleep interval and typically higher. When the No Buzz bit is set, the Sleep System Duty Cycle value is overridden and the bandgap circuit is forced to be on during sleep. This results in a faster response to an LVD or POR event (continuous detection as opposed to periodic detection), at the expense of higher average sleep current.

Bits 4 and 3: Sleep[1:0]. The available sleep interval selections are shown in Table 9-1. It must be remembered that when the ILO is the selected 32 kHz clock source, sleep intervals are approximate.

Table 9-1. Sleep Interval Selections

OSC_CR2[4]	Sleep Interval OSC_CR0[4:3]	Sleep Timer Clocks	Sleep Period (nominal)	Watchdog Period (nominal)
0	00b (default)	64	1.95 ms	6 ms
0	01b	512	15.6 ms	47 ms
0	10b	4,096	125 ms	375 ms
0	11b	32,768	1 sec	3 sec
1	00b (default)	65,536	2 sec	6 sec
1	01b	131,072	4 sec	12 sec
1	10b	262,144	8 sec	24 sec
1	11b	524,288	16 sec	48 sec

Bits 2 to 0: CPU Speed[2:0]. The PSoC M8C may operate over a range of CPU clock speeds (see Table 9-2), allowing the M8C's performance and power requirements to be tailored to the application.

The reset value for the CPU Speed bits is zero; therefore, the default CPU speed is one-eighth of the clock source. The Internal Main Oscillator (IMO) is the default clock source for the CPU speed circuit; therefore, the default CPU speed is 3 MHz.

The CPU frequency is changed with a write to the OSC_CR0 register. There are eight frequencies generated from a power-of-2 divide circuit, which are selected by a 3-bit code. At any given time, the CPU 8-to-1 clock mux is selecting one of the available frequencies, which is resynchronized to the 24 MHz master clock at the output.

Regardless of the CPU Speed bit's setting, if the actual CPU speed is greater than 12 MHz, the 24 MHz operating requirements apply. An example of this scenario is a device that is configured to use an external clock, which is supplying a frequency of 20 MHz. If the CPU speed register's value is 0b011, the CPU clock will be 20 MHz. Therefore, the supply voltage requirements for the device are the same as if the part was operating at 24 MHz off of the IMO. The operating voltage requirements are not relaxed until the CPU speed is at 12 MHz or less.

Table 9-2. OSC_CR0[2:0] Bits: CPU Speed

Bits	Internal Main Oscillator	External Clock
000b	3 MHz	EXTCLK/ 8
001b	6 MHz	EXTCLK/ 4
010b	12 MHz	EXTCLK/ 2
011b	24 MHz	EXTCLK/ 1
100b	1.5 MHz	EXTCLK/ 16
101b	750 kHz	EXTCLK/ 32
110b	187.5 kHz	EXTCLK/ 128
111b	93.7 kHz	EXTCLK/ 256

For additional information, refer to the OSC_CR0 register on page 254.

9.3.3 ECO_TR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,EBh	ECO_TR	PSSDC[1:0]								W : 00

The External Crystal Oscillator Trim Register (ECO_TR) sets the adjustment for the 32.768 kHz External Crystal Oscillator.

The device specific value placed in this register at boot time is based on factory testing. This register does not adjust the frequency of the external crystal oscillator.

It is strongly recommended that the user not alter the register value.

Bits 7 and 6: PSSDC[1:0]. These bits are used to set the sleep ***duty cycle***. These bits should not be altered.

For additional information, refer to the ECO_TR register on page 264.

10. Phase-Locked Loop (PLL)

This chapter presents the Phase-Locked Loop (PLL) and its associated registers. For a complete table of the PLL registers, refer to the “Summary Table of the Core Registers” on page 30. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

10.1 Architectural Description

A ***Phase-Locked Loop (PLL)*** function generates the system clock with crystal accuracy. It is designed to provide a 23.986 MHz oscillator, when utilized with an external 32.768 kHz crystal.

Although the PLL tracks crystal accuracy, it requires time to lock onto the reference frequency when first starting. The length of time depends on the PLLGAIN controlled by bit 7 of the OSC_CR2 register. If this bit is held low, the lock time is less than 10 ms. If this bit is held high, the lock time is on the order of 50 ms. After lock is achieved, it is recommended that this bit be forced high to decrease the ***jitter*** on the output. If longer lock time is tolerable, the PLLGAIN bit can be held high all the time.

After the 32.768 kHz External Crystal Oscillator (ECO) has been selected and enabled, the following procedure should be followed to enable the PLL and allow for proper frequency lock.

- Select a CPU frequency of 3 MHz or less.
- Enable the PLL.
- Wait between 10 and 50 ms, depending on bit 7 of the OSC_CR2 register.
- Set the CPU to a faster frequency, if desired. To do this, write the CPU Speed[2:0] bits in the OSC_CR0 register. The CPU frequency will immediately change when these bits are set.

If the proper settings are selected in ***PSoC Designer***, the above steps are automatically done in *boot.asm*.

10.2 Register Definitions

The following registers are associated with the Phase Locked Loop (PLL) and are listed in address order. Each register description has an associated register table showing the bit structure for that register. The bits that are grayed out in the tables below are reserved bits and are not detailed in the register descriptions. Note that reserved bits should always be written with a value of ‘0’. For a complete table of the PLL registers, refer to the “Summary Table of the Core Registers” on page 30.

10.2.1 OSC_CR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00

The Oscillator Control Register 0 (OSC_CR0) is used to configure various features of internal clock sources and clock nets.

Bit 7: 32k Select. By default, the 32 kHz clock source is the Internal Low Speed Oscillator (ILO). Optionally, the 32.768 kHz External Crystal Oscillator (ECO) may be selected.

Bit 6: PLL Mode. This is the only bit in the OSC_CR0 register that directly influences the Phase Locked Loop (PLL). When set, this bit enables the PLL. The EXTCLKEN bit in the OSC_CR2 register should be set low during PLL operation.

Bit 5: No Buzz. Normally, when the Sleep bit is set in the CPU_SCR register, all PSoC device systems are powered down, including the bandgap reference. However, to facilitate the detection of ***POR*** and ***LVD*** events at a rate higher than the sleep interval, the bandgap circuit is powered up periodically for about 60 µs at the Sleep System Duty Cycle (set in ECO_TR), which is independent of the sleep interval and typically higher. When the No Buzz bit is set, the Sleep System Duty Cycle value is overridden and the bandgap circuit is forced to be on during sleep. This results in a faster response to an LVD or POR event (continuous detection as opposed to periodic detection), at the expense of slightly higher average sleep current.

Bits 4 and 3: Sleep[1:0]. The available sleep interval selections are shown in Table 10-1. It must be remembered that when the ILO is the selected 32 kHz clock source, sleep intervals are approximate.

Table 10-1. Sleep Interval Selections

OSC_CR2[4]	Sleep Interval OSC_CR0[4:3]	Sleep Timer Clocks	Sleep Period (nominal)	Watchdog Period (nominal)
0	00b (default)	64	1.95 ms	6 ms
0	01b	512	15.6 ms	47 ms
0	10b	4,096	125 ms	375 ms
0	11b	32,768	1 sec	3 sec
1	00b (default)	65,536	2 sec	6 sec
1	01b	131,072	4 sec	12 sec
1	10b	262,144	8 sec	24 sec
1	11b	524,288	16 sec	48 sec

Bits 2 to 0: CPU Speed[2:0]. The PSoC M8C may operate over a range of CPU clock speeds (see Table 10-2), allowing the M8C's performance and power requirements to be tailored to the application.

The reset value for the CPU Speed bits is zero; therefore, the default CPU speed is one-eighth of the clock source. The Internal Main Oscillator (IMO) is the default clock source for the CPU speed circuit; therefore, the default CPU speed is 3 MHz.

The CPU frequency is changed with a write to the OSC_CR0 register. There are eight frequencies generated from a power-of-2 divide circuit, which are selected by a 3-bit code. At any given time, the CPU 8-to-1 clock mux is selecting one of the available frequencies, which is resynchronized to the 24 MHz master clock at the output.

Regardless of the CPU Speed bit's setting, if the actual CPU speed is greater than 12 MHz, the 24 MHz operating requirements apply. An example of this scenario is a device that is configured to use an external clock, which is supplying a frequency of 20 MHz. If the CPU speed register's value is 0b011, the CPU clock will be 20 MHz. Therefore, the supply voltage requirements for the device are the same as if the part was operating at 24 MHz off of the IMO. The operating voltage requirements are not relaxed until the CPU speed is at 12 MHz or less.

Some devices support the slow IMO option, as discussed in the IMO chapter in the "Architectural Description" on page 75. This offers an option to lower both system and CPU clock speed in order to save power.

An automatic protection mechanism is available for systems that need to run at peak CPU clock speed but cannot guarantee a high enough supply voltage for that clock speed. See the LVDTBEN bit in the "VLT_CR Register" on page 420 for more information.

Table 10-2. OSC_CR0[2:0] Bits: CPU Speed

Bits	6 MHz Internal Main Oscillator *	24 MHz Internal Main Oscillator	External Clock
000b	750 kHz	3 MHz	EXTCLK/ 8
001b	1.5 MHz	6 MHz	EXTCLK/ 4
010b	3 MHz	12 MHz	EXTCLK/ 2
011b	6 MHz	24 MHz	EXTCLK/ 1
100b	375 kHz	1.5 MHz	EXTCLK/ 16
101b	187.5 kHz	750 kHz	EXTCLK/ 32
110b	93.7 kHz	187.5 kHz	EXTCLK/ 128
111b	46.9 kHz	93.7 kHz	EXTCLK/ 256

* For PSoC devices that support the slow IMO option, see the "Architectural Description" on page 75.

For additional information, refer to the OSC_CR0 register on page 254.

10.2.2 OSC_CR2 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E2h	OSC_CR2	PLLGAIN			SLP_EXTEND	WDR32_SE	EXTCLKEN	RSVD	SYSCLKX2DIS	RW : 00

The Oscillator Control Register 2 (OSC_CR2) is used to configure various features of internal clock sources and clock nets.

Bit 7: PLLGAIN. This is the only bit in the OSC_CR2 register that directly influences the PLL. When set, this bit keeps the PLL in Low Gain mode.

If this bit is held low, the lock time is less than 10 ms. If this bit is held high, the lock time is on the order of 50 ms. After lock is achieved, it is recommended that this bit be forced high to decrease the jitter on the output. If longer lock time is tolerable, the PLLGAIN bit can be held high all the time.

Bit 4: SLP_EXTEND. Long Sleep extension.

Bit 3: WDR32_SE. Watchdog clock source selection.

Bit 2: EXTCLKEN. When the EXTCLKEN bit is set, the external clock becomes the source for the internal clock tree, SYSCLK, which drives most PSoC device clocking functions. All external and internal signals, including the 32 kHz clock, whether derived from the Internal Low Speed Oscillator (ILO) or the crystal oscillator, are synchronized to this clock source. If an external clock is enabled, PLL mode should be off. The external clock input is located on port P1[4]. When using this input, the pin drive mode should be set to High Z (not High Z analog).

Bit 1: RSVD. This bit should always be 0.

Bit 0: SYSCLKX2DIS. When SYSCLKX2DIS is set, the IMO's doubler is disabled. This will result in a reduction of overall device power, on the order of 1 mA. It is advised that any application that does not require this doubled clock should have it turned off. During emulation with the In-Circuit Emulator (ICE), the IMO's doubler is always active regardless of the status of SYSCLKX2DIS.

For additional information, refer to the OSC_CR2 register on page 256.

11. Sleep and Watchdog

This chapter discusses the Sleep and Watchdog operations and their associated registers. For a complete table of the Sleep and Watchdog registers, refer to the “Summary Table of the Core Registers” on page 30. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

11.1 Architectural Description

Device components that are involved in Sleep and Watchdog operation are the selected 32 kHz clock (external crystal or internal), the sleep timer, the Sleep bit in the CPU_SCR0 register, the sleep circuit (to sequence going into and coming out of sleep), the bandgap refresh circuit (to periodically refresh the reference voltage during sleep), and the ***watchdog timer***.

The goal of Sleep operation is to reduce average power consumption as much as possible. The system has a sleep state that can be initiated under firmware control. In this state, the CPU is stopped at an instruction boundary and the 24/48 MHz oscillator (IMO), the Flash memory module, and bandgap voltage reference are powered down. The only blocks that remain in operation are the 32 kHz oscillator (external crystal or internal), ***PSoC blocks*** clocked from the 32 kHz clock selection, and the supply voltage monitor circuit.

Analog PSoC blocks have individual power down settings that are controlled by firmware, independently of the sleep state. Continuous time analog blocks may remain in operation, because they do not require a clock source. Typically, switched capacitor analog blocks will not operate, because the internal sources of clocking for these blocks are stopped.

The system can only wake up from sleep as a result of an interrupt or reset event. The sleep timer can provide periodic interrupts to allow the system to wake up, poll peripherals, or do real-time functions, and then go to sleep again. The GPIO (pin) interrupt, supply monitor interrupt, analog column interrupts, and timers clocked externally or from the 32 kHz clock are examples of ***asynchronous*** interrupts that can also be used to wake the system up.

The Watchdog Timer (WDT) circuit is designed to assert a ***hardware reset*** to the device after a pre-programmed interval, unless it is periodically serviced in firmware. In the event that an unexpected execution path is taken through the code, this functionality serves to reboot the system. It can also restart the system from the CPU halt state.

Once the WDT is enabled, it can only be disabled by an External Reset (XRES) or a Power On Reset (POR). A WDT reset will leave the WDT enabled. Therefore, if the WDT is used in an application, all code (including initialization code) must be written as though the WDT is enabled.

11.1.1 32 kHz Clock Selection

By default, the 32 kHz clock source is the Internal Low Speed Oscillator (ILO). Optionally, the 32.768 kHz External Crystal Oscillator (ECO) may be activated. This selection is made in bit 7 of the OSC_CR0 register. Selecting the ECO as the source for the 32 kHz clock allows the sleep timer and sleep interrupt to be used in real-time clock applications. Regardless of the clock source selected, the 32 kHz clock plays a key role in sleep functionality. It runs continuously and is used to sequence system wakeup. It is also used to periodically refresh the bandgap voltage during sleep.

Refer to the External Crystal Oscillator (ECO) chapter on page 80, for details on activating an external crystal oscillator.

11.1.2 Sleep Timer

The sleep timer is a 15-bit up counter clocked by the currently selected 32 kHz clock source, either the ILO or ECO. This timer is always enabled. The exception to this is within an ***ICE*** (in-circuit ***emulator***) in ***debugger*** mode and when the Stop bit in the CPU_SCR0 is set; the sleep timer is disabled, so that the user will not get continual watchdog resets when a breakpoint is hit in the debugger environment.

If the associated sleep timer interrupt is enabled, a periodic interrupt to the CPU is generated based on the sleep interval selected from the OSC_CR0 register. The sleep timer functionality does not need to be directly associated with the

sleep state. It can be used as a general purpose timer interrupt regardless of sleep state.

The reset state of the sleep timer is a count value of all zeros. There are two ways to reset the sleep timer. Any hardware reset, (that is, POR, XRES, or Watchdog Reset (WDR) will reset the sleep timer. There is also a method that allows the user to reset the sleep timer in firmware. A write of 38h to the RES_WDT register clears the sleep timer.

Note Any write to the RES_WDT register also clears the watchdog timer.

Clearing the sleep timer may be done at anytime to synchronize the sleep timer operation to CPU processing. A good example of this is after POR. The CPU hold-off, due to voltage ramp and others, may be significant. In addition, a significant amount of program initialization may be required. However, the sleep timer starts counting immediately after POR and will be at an arbitrary count when user code begins execution. In this case, it may be desirable to clear the sleep timer before enabling the sleep interrupt initially, to ensure that the first sleep period is a full interval.

11.2 Application Description

The following are notes regarding sleep as it relates to firmware and application issues.

Note 1 If an interrupt is pending, enabled, and scheduled to be taken at the instruction boundary after the write to the sleep bit, the system will not go to sleep. The instruction will still execute, but it will not be able to set the SLEEP bit in the CPU_SCR0 register. Instead, the interrupt will be taken and the effect of the sleep instruction is ignored.

Note 2 The Global Interrupt Enable (CPU_F register) does not need to be enabled to wake the system out of sleep state. Individual interrupt enables, as set in the interrupt mask registers, are sufficient. If the Global Interrupt Enable is not set, the CPU will not service the ISR associated with that interrupt. However, the system will wake up and continue executing instructions from the point at which it went to sleep. In this case, the user must manually clear the pending interrupt or subsequently enable the Global Interrupt Enable bit and let the CPU take the ISR. If a pending interrupt is not cleared, it will be continuously asserted. Although the sleep bit may be written and the sleep sequence executed as soon as the device enters Sleep mode, the Sleep bit is cleared by the pending interrupt and Sleep mode is exited immediately.

Note 3 On wake up, the instruction immediately after the sleep instruction is executed before the interrupt service routine (if enabled). The instruction after the sleep instruction is pre-fetched, before the system actually goes to sleep. Therefore, when an interrupt occurs to wake the system up, the pre-fetched instruction is executed and then the interrupt service routine is executed. (If the Global Interrupt Enable is not set, instruction execution will just continue where it left off before sleep.)

Note 4 If PLL mode is enabled, CPU frequency must be reduced to 3 MHz before going to sleep. Because the PLL will overshoot as it attempts to re-lock after wakeup, the CPU frequency must be relatively low. It is recommended to wait 10 ms after wakeup, before normal CPU operating frequency may be restored.

Note 5 Analog power must be turned off by firmware before going to sleep, to achieve the smallest sleep current. The system sleep state does not control the analog array. There are individual power controls for each analog block and global power controls in the reference block. These power controls must be manipulated by firmware.

Note 6 If the Global Interrupt Enable bit is disabled, it can be safely enabled just before the instruction that writes the sleep bit. It is usually undesirable to get an interrupt on the instruction boundary, just before writing the sleep bit. This means that on the return from interrupt, the sleep command will be executed, possibly bypassing any firmware preparations that must be made in order to go to sleep. To prevent this, disable interrupts before preparations are made. After sleep preparations, enable global interrupts and write the sleep bit with the two consecutive instructions as follows.

```
and f,~01h          // disable global interrupts
                    // (prepare for sleep, could
                    // be many instructions)
or f,01h            // enable global interrupts
mov reg[ffh],08h    // Set the sleep bit
```

Due to the timing of the Global Interrupt Enable instruction, it is not possible for an interrupt to occur immediately after that instruction. The earliest the interrupt could occur is after the next instruction (write to the Sleep bit) has been executed. Therefore, if an interrupt is pending, the sleep instruction is executed; but as described in Note 1, the sleep instruction will be ignored. The first instruction executed after the ISR is the instruction after sleep.

11.3 Register Definitions

The following registers are associated with Sleep and Watchdog and are listed in address order. Each register description has an associated register table showing the bit structure for that register. The bits that are grayed out in the tables below are reserved bits and are not detailed in the register descriptions. Note that reserved bits should always be written with a value of '0'. For a complete table of the Sleep and Watchdog registers, refer to the "Summary Table of the Core Registers" on page 30.

11.3.1 INT_MSK0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E0h 4 Cols.	INT_MSK0	VC3	Sleep	GPIO	Analog 3	Analog 2	Analog 1	Analog 0	V Monitor	RW : 00

The Interrupt Mask Register 0 (INT_MSK0) is used to enable the individual sources' ability to create pending interrupts.

Depending on your PSoC device's characteristics, only certain bits are accessible to be read or written in the analog column dependent INT_MSK0 register. (Refer to the table titled "PSoC Device Characteristics" on page 21.) In the table above, the analog column numbers are listed to the right in the Address column.

Bits 7 and 5 to 0. The INT_MSK0 register holds bits that are used by several different resources. For a full discussion of the INT_MSK0 register, see the Interrupt Controller chapter on page 60.

Bit 6: Sleep. This bit controls the sleep interrupt enable.

For additional information, refer to the INT_MSK0 register on page 174.

11.3.2 RES_WDT Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E3h	RES_WDT	WDSL_Clear[7:0]								W : 00

The Reset Watchdog Timer Register (RES_WDT) is used to clear the watchdog timer (a write of any value) and clear both the watchdog timer and the sleep timer (a write of 38h).

Bits 7 to 0: WDSL_Clear[7:0]. The Watchdog Timer (WDT) write-only register is designed to timeout at three rollover events of the sleep timer. Therefore, if only the WDT is cleared, the next Watchdog Reset (WDR) will occur anywhere from two to three times the current sleep interval setting. If the sleep timer is near the beginning of its count, the watchdog timeout will be closer to three times. However, if the sleep timer is very close to its ***terminal count***, the watchdog timeout will be closer to two times. To ensure a full three times timeout, both the WDT and the sleep timer may be cleared. In applications that need a real-time clock, and thus cannot reset the sleep timer when clearing the WDT, the duty cycle at which the WDT must be cleared should be no greater than two times the sleep interval.

For additional information, refer to the RES_WDT register on page 177.

11.3.3 CPU_SCR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW *	ECO EX *		IRAMDIS	# : 00

LEGEND
x An "x" before the comma in the address field indicates that this register can be read or written to no matter what bank is used.
Access is bit specific. Refer to the Register Details chapter on page 103 for additional information.
* Bits 3 and 2 (ECO EXW and ECO EX, respectively) cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24x23 and CY8C22x13 PSoC devices.

The System Status and Control Register 1 (CPU_SCR1) is used to convey the status and control of events related to internal resets and watchdog reset.

Bit 7: IRESS. The Internal Reset Status bit is a read only bit that may be used to determine if the booting process occurred more than once.

When this bit is set, it indicates that the SROM SWBootReset code was executed more than once. If this bit is not set, the SWBootReset was executed only once. In either case, the SWBootReset code will not allow execution from code stored in Flash until the M8C Core is in a safe operating mode with respect to supply voltage and Flash operation. There is no need for concern when this bit is set. It is provided for systems which may be sensitive to boot time, so that they can determine if the normal one-pass boot time was exceeded. For more information on the SWBootReest code see the Supervisory ROM (SROM) chapter on page 43.

Bit 4: SLIMO. When set, the Slow IMO bit allows the active power dissipation of the PSoC device to be reduced by slowing down the IMO from 24 MHz to 6 MHz. The IMO trim value must also be changed when SLIMO is set (see "Engaging Slow IMO" on page 76). When not in external clocking mode, the IMO is the source for SYSCLK; therefore, when the speed of the IMO changes, so will SYSCLK.

Bit 3: ECO EXW. The ECO Exists Written bit is used as a status bit to indicate that the ECO EX bit has been previously written to. It is read only. When this bit is a '1', this indicates that the CPU_SCR1 register has been written to and is now locked. When this bit is a '0', the register has not been written to since the last reset event. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24x23 and CY8C22x13 PSoC devices.

Bit 2: ECO EX. The ECO Exists bit serves as a flag to the hardware, to indicate that an external crystal ***oscillator*** exists in the system. Just after boot, it may be written *only once* to a value of '1' (crystal exists) or '0' (crystal does not exist). If the bit is '0', a switch-over to the ECO is locked out by hardware. If the bit is '1', hardware allows the firmware to freely switch between the ECO and ILO. It should be written as early as possible after a ***Power On Reset (POR)*** or ***External Reset (XRES)*** event, where it is assumed that program execution integrity is high. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24x23 and CY8C22x13 PSoC devices.

Bit 0: IRAMDIS. The Initialize RAM Disable bit is a control bit that is readable and writeable. The ***default value*** for this bit is '0', which indicates that the maximum amount of SRAM should be initialized on watchdog reset to a value of 00h. When the bit is '1', the minimum amount of SRAM is initialized after a watchdog reset. For more information on this bit, see the "SROM Function Descriptions" on page 44.

For additional information, refer to the CPU_SCR1 register on page 190.

11.3.4 CPU_SCR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,FFh	CPU_SCR0	GIES		WDRS	PORS	Sleep			STOP	# : XX

LEGEND
X The value for power on reset is unknown.
x An “x” before the comma in the address field indicates that this register can be read or written to no matter what bank is used.
Access is bit specific. Refer to register detail for additional information.

The System Status and Control Register 0 (CPU_SCR0) is used to convey the status and control of events for various functions of a PSoC device.

Bit 7: GIES. The Global Interrupt Enable Status bit is a read only status bit and its use is discouraged. The GIES bit is a legacy bit which was used to provide the ability to read the GIE bit of the CPU_F register. However, the CPU_F register is now readable. When this bit is set, it indicates that the GIE bit in the CPU_F register is also set which, in turn, indicates that the microprocessor will service interrupts.

Bit 5: WDRS. The WatchDog Reset Status bit may not be set. It is normally ‘0’ and automatically set whenever a watchdog reset occurs. The bit is readable and clearable by writing a zero to its bit position in the CPU_SCR0 register.

Bit 4: PORS. The Power On Reset Status (PORS) bit, which is the watchdog enable bit, is set automatically by a POR or External Reset (XRES). If the bit is cleared by user code, the watchdog timer is enabled. Once cleared, the only way to reset the PORS bit is to go through a POR or XRES. Thus, there is no way to disable the watchdog timer, other than to go through a POR or XRES.

Bit 3: Sleep. The Sleep bit is used to enter Low Power Sleep mode when set. To wake up the system, this register bit is cleared asynchronously by any enabled interrupt. There are two special features of this register bit that ensures proper Sleep operation. First, the write to set the register bit is blocked, if an interrupt is about to be taken on that instruction boundary (immediately after the write). Second, there is a hardware interlock to ensure that, once set, the sleep bit may not be cleared by an incoming interrupt until the sleep circuit has finished performing the sleep sequence and the system-wide power down signal has been asserted. This prevents the sleep circuit from being interrupted in the middle of the process of system power down, possibly leaving the system in an indeterminate state.

Bit 0: STOP. The STOP bit is readable and writeable. When set, the PSoC M8C will stop executing code until a reset event occurs. This can be either a POR, WDR, or XRES. If an application wants to stop code execution until a reset, the preferred method would be to use the HALT instruction rather than a register write to this bit.

For additional information, refer to the CPU_SCR0 register on page 191.

11.3.5 OSC_CR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00

The Oscillator Control Register 0 (OSC_CR0) is used to configure various features of internal clock sources and clock nets.

Bit 7: 32k Select. By default, the 32 kHz clock source is the Internal Low Speed Oscillator (ILO). Optionally, the 32.768 kHz External Crystal Oscillator (ECO) may be selected.

Bit 6: PLL Mode. This is the only bit in the OSC_CR0 register that directly influences the Phase Locked Loop (PLL). When set, this bit enables the PLL. The EXTCLKEN bit in the OSC_CR2 register should be set low during PLL operation. For information on the PLL, refer to the Phase-Locked Loop (PLL) chapter on page 85.

Bit 5: No Buzz. Normally, when the Sleep bit is set in the CPU_SCR register, all PSoC device systems are powered down, including the bandgap reference. However, to facilitate the detection of ***POR*** and ***LVD*** events at a rate higher than the sleep interval, the bandgap circuit is powered up periodically for about 60 µs at the Sleep System Duty Cycle (set in ECO_TR), which is independent of the sleep interval and typically higher. When the No Buzz bit is set, the Sleep System Duty Cycle value is overridden and the bandgap circuit is forced to be on during sleep. This results in a faster response to an LVD or POR event (continuous detection as opposed to periodic detection), at the expense of slightly higher average sleep current.

Bits 4 and 3: Sleep[1:0]. The available sleep interval selections are shown in Table 11-1. The accuracy of the sleep intervals are dependent on the accuracy of the oscillator used.

Table 11-1. Sleep Interval Selections

OSC_CR 2[4]	Sleep Interval OSC_CR0[4:3]	Sleep Timer Clocks	Sleep Period (nominal)	Watchdog Period (nominal)
0	00b (default)	64	1.95 ms	6 ms
0	01b	512	15.6 ms	47 ms
0	10b	4,096	125 ms	375 ms
0	11b	32,768	1 sec	3 sec
1	00b (default)	65,536	2 sec	6 sec
1	01b	131,072	4 sec	12 sec
1	10b	262,144	8 sec	24 sec
1	11b	524,288	16 sec	48 sec

Bits 2 to 0: CPU Speed[2:0]. The PSoC M8C may operate over a range of CPU clock speeds (see Table 11-2), allowing the M8C's performance and power requirements to be tailored to the application.

The reset value for the CPU Speed bits is zero; therefore, the default CPU speed is one-eighth of the clock source. The Internal Main Oscillator (IMO) is the default clock source for the CPU speed circuit; therefore, the default CPU speed is 3 MHz.

The CPU frequency is changed with a write to the OSC_CR0 register. There are eight frequencies generated from a power-of-2 divide circuit, which are selected by a 3-bit code. At any given time, the CPU 8-to-1 clock mux is selecting one of the available frequencies, which is resynchronized to the 24 MHz master clock at the output.

Regardless of the CPU Speed bit's setting, if the actual CPU speed is greater than 12 MHz, the 24 MHz operating requirements apply. An example of this scenario is a device that is configured to use an external clock, which is supplying a frequency of 20 MHz. If the CPU speed register's value is 011b, the CPU clock will be 20 MHz. Therefore, the supply voltage requirements for the device are the same as if the part was operating at 24 MHz off of the IMO. The operating voltage requirements are not relaxed until the CPU speed is at 12 MHz or less.

Table 11-2. OSC_CR0[2:0] Bits: CPU Speed

Bits	Internal Main Oscillator	External Clock
000b	3 MHz	EXTCLK/ 8
001b	6 MHz	EXTCLK/ 4
010b	12 MHz	EXTCLK/ 2
011b	24 MHz	EXTCLK/ 1
100b	1.5 MHz	EXTCLK/ 16
101b	750 kHz	EXTCLK/ 32
110b	187.5 kHz	EXTCLK/ 128
111b	93.7 kHz	EXTCLK/ 256

For additional information, refer to the OSC_CR0 register on page 254.

11.3.6 OSC_CR2 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E2h	OSC_CR2	PLLGAIN			SLP_EXTEND	WDR32_SE	EXTCLKEN	RSVD	SYSCLKX2DIS	RW : 00

The Oscillator Control Register 2 (OSC_CR2) is used to configure various features of internal clock sources and clock nets.

Bit 7: PLLGAIN. This is the only bit in the OSC_CR2 register that directly influences the PLL. When set, this bit keeps the PLL in a Low Gain mode. If this bit is held low, the lock time is less than 10 ms. If this bit is held high, the lock time is on the order of 50 ms. After lock is achieved, it is recommended that this bit be forced high to decrease the jitter on the output. If longer lock time is tolerable, the PLLGAIN bit can be held high all the time.

Bit 4: SLP_EXTEND. Long Sleep extension.

Bit 3: WDR32_SE. Watchdog clock source selection.

Bit 2: EXTCLKEN. When the EXTCLKEN bit is set, the external clock becomes the source for the internal clock tree, SYSCLK, which drives most PSoC device clocking functions. All external and internal signals, including the 32 kHz clock, whether derived from the internal low speed oscillator (ILO) or the crystal oscillator, are synchronized to this clock source. If an external clock is enabled, PLL mode should be off. The external clock input is located on port P1[4]. When using this input, the pin drive mode should be set to High Z (not High Z analog).

Bit 1: RSVD. This bit should always be 0.

Bit 0: SYSCLKX2DIS. When set, the Internal Main Oscillator's doubler is disabled. This results in a reduction of overall device power, on the order of 1 mA. It is advised that any application that does not require this doubled clock should have it turned off.

For additional information, refer to the OSC_CR2 register on page 256.

11.3.7 ILO_TR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E9h	ILO_TR			Bias Trim[1:0]		Freq Trim[3:0]				W : 00

The Internal Low Speed Oscillator Trim Register (ILO_TR) sets the adjustment for the internal low speed oscillator.

The device specific value, placed in the trim bits of this register at boot time, is based on factory testing. ***It is strongly recommended that the user not alter the register value***.

Bits 5 and 4: Bias Trim[1:0]. These two bits are used to set the bias current in the PTAT Current Source. Bit 5 gets inverted, so that a medium bias is selected when both bits are '0'. The bias current is set according to Table 11-3.

Table 11-3. Bias Current in PTAT

Bias Current	Bias Trim [1:0]
Medium Bias	00b
Maximum Bias	01b
Minimum Bias	10b
Not needed *	11b

* About 15% higher than the minimum bias.

Bits 3 to 0: Freq Trim[3:0]. These four bits are used to trim the frequency. Bit 0 is the LSb and bit 3 is the MSb. Bit 3 gets inverted inside the register.

For additional information, refer to the ILO_TR register on page 262.

11.3.8 ECO_TR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,EBh	ECO_TR	PSSDC[1:0]								W : 00

The External Crystal Oscillator Trim Register (ECO_TR) sets the adjustment for the 32.768 kHz external crystal oscillator.

The value placed in this register is based on factory testing. This register does not adjust the frequency of the external crystal oscillator. ***It is strongly recommended that the user not alter the register value.***

Bits 7 and 6: PSSDC[1:0]. These bits are used to set the sleep duty cycle. These bits should not be altered.

For additional information, refer to the ECO_TR register on page 264.

11.4 Timing Diagrams

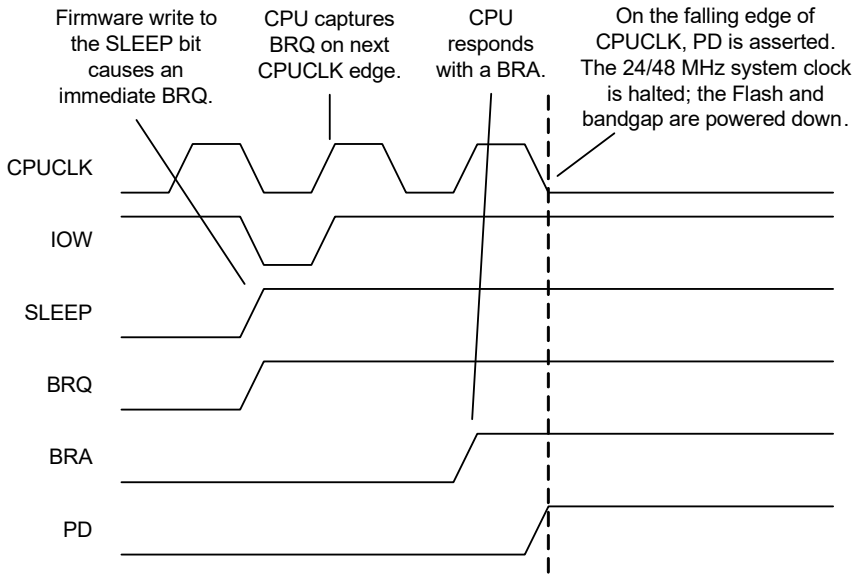
11.4.1 Sleep Sequence

The Sleep bit, in the CPU_SCR0 register, is an input into the sleep logic circuit. This circuit is designed to sequence the device into and out of the hardware sleep state. The hardware sequence to put the device to sleep is shown in Figure 11-1 and is defined as follows.

1. Firmware sets the SLEEP bit in the CPU_SCR0 register. The Bus Request (BRQ) signal to the CPU is immediately asserted: This is a request by the system to halt CPU operation at an instruction boundary.
2. The CPU issues a Bus Request Acknowledge (BRA) on the following ***positive edge*** of the CPU clock.
3. The sleep logic waits for the following ***negative edge*** of the CPU clock and then asserts a system-wide Power Down (PD) signal. In Figure 11-1, the CPU is halted and the system-wide power down signal is asserted.

The system-wide PD signal controls three major circuit blocks: the Flash memory module, the Internal Main Oscillator (24/48 MHz oscillator that is also called the IMO), and the bandgap voltage reference. These circuits transition into a zero power state. The only operational circuits on the PSoC device are the ILO (or optional ECO), the bandgap refresh circuit, and the supply voltage monitor circuit. Note that the system sleep state does not apply to the analog array. Power down settings for individual analog blocks and references must be done in firmware, prior to executing the sleep instruction.

Figure 11-1. Sleep Sequence



11.4.2 Wake Up Sequence

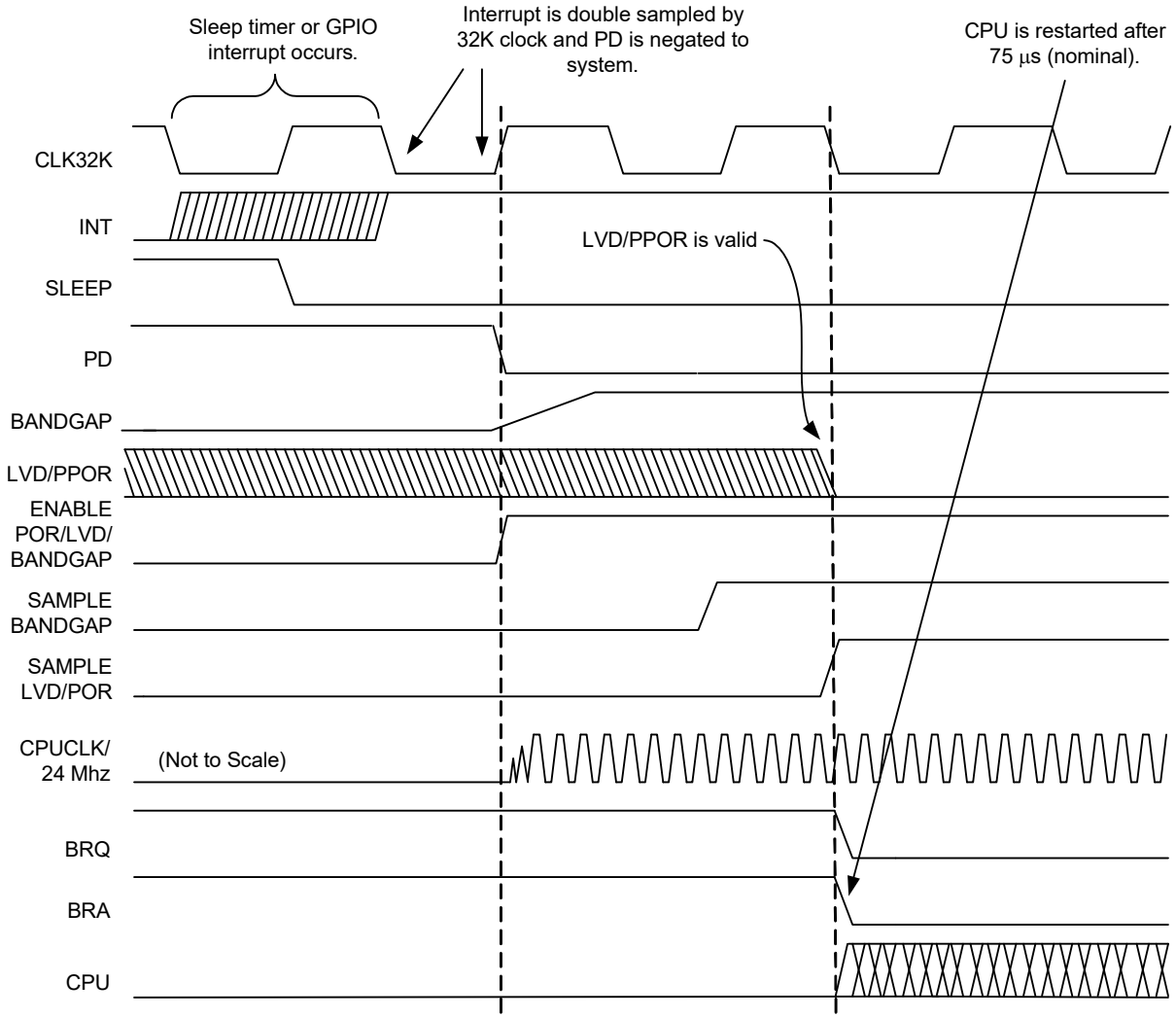
Once asleep, the only event that can wake the system up is an interrupt. The Global Interrupt Enable of the CPU flag register does not need to be set. Any unmasked interrupt will wake the system up. It is optional for the CPU to actually take the interrupt after the wakeup sequence.

The wake up sequence is synchronized to the 32 kHz clock for purposes of sequencing a startup delay, to allow the Flash memory module enough time to power up before the CPU asserts the first read access. Another reason for the delay is to allow the IMO, bandgap, and LVD/POR circuits time to settle before actually being used in the system. As shown in Figure 11-2, the wake up sequence is as follows.

1. The wake up interrupt occurs and is synchronized by the negative edge of the 32 kHz clock.
2. At the following positive edge of the 32 kHz clock, the system-wide PD signal is negated. The Flash memory module, IMO, and bandgap any POR/LVD circuits are all powered up to a normal operating state.
3. At the next positive edge of the 32 kHz clock, the values of the bandgap are settled and sampled.
4. At the following negative edge of the 32 kHz clock (after about 15 μs, nominal). The values of the POR/LVD signals have settled and are sampled. The BRQ signal is negated by the sleep logic circuit. On the following CPU clock, BRA is negated by the CPU and instruction execution resumes.

The wake up times (interrupt to CPU operational) will range from two to three 32 kHz cycles or 61 - 92 μs (nominal).

Figure 11-2. Wakeup Sequence

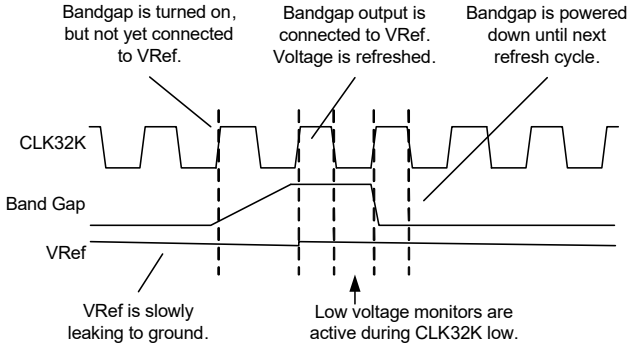


11.4.3 Bandgap Refresh

During normal operation, the bandgap circuit provides a voltage reference (VRef) to the system, for use in the analog blocks, Flash, and ***low voltage detect (LVD)*** circuitry. Normally, the bandgap output is connected directly to the VRef signal. However, during sleep, the ***bandgap reference*** generator block and LVD circuits are completely powered down. The bandgap and LVD blocks are periodically re-enabled during sleep, in order to monitor for low voltage conditions. This is accomplished by turning on the bandgap periodically, allowing it time to start up for a full 32 kHz clock period, and connecting it to VRef to refresh the reference voltage for the following 32 kHz clock period as shown in Figure 11-3.

During the second 32 kHz clock period of the refresh cycle, the LVD circuit is allowed to settle during the ***high time*** of the 32 kHz clock. During the low period of the second 32 kHz clock, the LVD interrupt is allowed to occur.

Figure 11-3. Bandgap Refresh Operation



The rate at which the refresh occurs is related to the 32 kHz clock and controlled by the Power System Sleep Duty Cycle (PSSDC), bits [7:6] of the ECO_TR register). Table 11-4 enumerates the available selections. The default setting (256 sleep timer counts) is applicable for many applications, giving a typical average device current under 5 μA.

Table 11-4. Power System Sleep Duty Cycle Selections

PSSDC	Sleep Timer Counts	Period (Nominal)
00b (default)	256	8 ms
01b	1024	31.2 ms
10b	64	2 ms
11b	16	500 μs

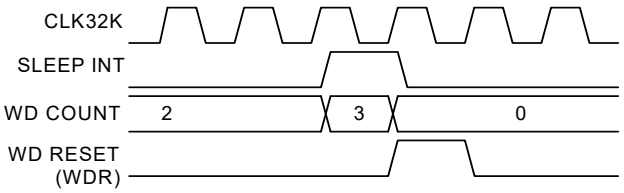
11.4.4 Watchdog Timer

On device boot up, the Watchdog Timer (WDT) is initially disabled. The PORS bit in the system control register controls the enabling of the WDT. On boot, the PORS bit is initially set to '1', indicating that either a POR or XRES event has occurred. The WDT is enabled by clearing the PORS bit. Once this bit is cleared and the watchdog timer is enabled, it cannot be subsequently disabled. (The PORS bit cannot be set to '1' in firmware; it can only be cleared.)

The only way to disable the Watchdog function, after it is enabled, is through a subsequent POR or XRES. Although the WDT is disabled during the first time through initialization code after a POR or XRES, all code should be written as if it is enabled (that is, the WDT should be cleared periodically). This is because, in the initialization code after a WDR event, the watchdog timer is enabled so all code must be aware of this.

The watchdog timer is three counts of the sleep timer interrupt output. The watchdog interval is three times the selected sleep timer interval. The available selections for the watchdog interval are shown in Table 11-1. When the sleep timer interrupt is asserted, the watchdog timer increments. When the counter reaches three, a terminal count is asserted. This terminal count is registered by the 32 kHz clock. Therefore, the WDR (Watchdog Reset) signal will go high after the following edge of the 32 kHz clock and be held asserted for one cycle (30 μs nominal). The ***flip-flop*** that registers the WDT terminal count is not reset by the WDR signal when it is asserted, but is reset by all other resets. This timing is shown in Figure 11-4.

Figure 11-4. Watchdog Reset



Once enabled, the WDT must be periodically cleared in firmware. This is accomplished with a write to the RES_WDT register. This write is data independent, so any write will clear the watchdog timer. (Note that a write of 38h will also clear the sleep timer.) If for any reason the firmware fails to clear the WDT within the selected interval, the circuit will assert WDR to the device. WDR is equivalent in effect to any other reset. All internal registers are set to their reset state, see the table titled "Details of Functionality for Various Resets" on page 418. An important aspect to remember about WDT resets is that RAM initialization can be disabled (IRAMDIS in the CPU_SCR1 register). In this case, the SRAM contents are unaffected; so that when a WDR occurs, program variables are persistent through this reset.

In practical application, it is important to know that the watchdog timer interval can be anywhere between two and three times the sleep timer interval. The only way to guarantee that the WDT interval is a full three times that of the sleep interval is to clear the sleep timer (write 38h) when clearing the WDT register. However, this is not possible in applications that use the sleep timer as a real-time clock. In the case where firmware clears the WDT register without clearing the sleep timer, this can occur at any point in a given sleep timer interval. If it occurs just before the terminal count of a sleep timer interval, the resulting WDT interval will be just over two times that of the sleep timer interval.

11.5 Power Consumption

Sleep mode power consumption consists of the items in the following tables.

In Table 11-5, the typical block currents shown do not represent maximums. These currents do not include any analog block currents that may be on during Sleep mode.

Table 11-5. Continuous Currents

IPOR	1 µA
ICLK32K (ILO/ECO)	1 µA

While the CLK32K can be turned off in Sleep mode, this mode is not useful because it makes it impossible to restart unless an imprecise power on reset (IPOR) occurs. (The Sleep bit can not be cleared without CLK32K.) During the sleep mode buzz, the bandgap is on for two cycles and the LVD circuitry is on for one cycle. Time-averaged currents from periodic sleep mode 'buzz', with periodic count of N, are listed in Table 11-6.

Table 11-6. Time-Averaged Currents

IBG (Bandgap)	(2/N) * 60 µA
ILVD (LVD comparators)	(2/N) * 50 µA

Table 11-7 lists example currents for N=256 and N=1024. Device leakage currents add to the totals in the table.

Table 11-7. Example Currents

	N=256	N=1024
IPOR	1	1
CLK32K	1	1
IBG	0.46	0.12
ILVD	0.4	0.1
Total	2.9 µA	2.2 µA

Section C: Register Reference

The Register Reference section discusses the registers of the PSoC device. It lists all the registers in mapping tables, in address order. For easy reference, each register is linked to the page of a detailed description located in the next chapter. This section encompasses the following chapter:

- Register Details on page 103

Register General Conventions

The register conventions specific to this section and the Register Details chapter are listed in the following table.

Register Conventions

Convention	Description
Empty, grayed-out table cell	Illustrates a reserved bit or group of bits.
'x' before the comma in an address	Indicates the register exists in register bank 1 and register bank 2.
'x' in a register name	Indicates that there are multiple instances/address ranges of the same register.
R	Read register or bit(s)
W	Write register or bit(s)
L	Logical register or bit(s)
C	Clearable register or bit(s)
#	Access is bit specific

Register Naming Conventions

The register naming convention specific to this section for arrays of PSoC blocks and their registers is:

<Prefix>mn<Suffix>
where m=row index, n=column index

Therefore, ACE01CR2 is a register for an analog PSoC block in row 0 column 1.

Register Mapping Tables

The PSoC device has a total register address space of 512 bytes. The register space is also referred to as I/O space and is broken into two parts. The XIO bit in the Flag register (CPU_F) determines which bank the user is currently in. When the XIO bit is set, the user is said to be in the "extended" address space or the "configuration" registers.

Refer to the individual PSoC device data sheets for device-specific register mapping information.

Register Map Bank 0 Table: User Space

Name	Addr (0,Hex)	Access	Page	Name	Addr (0,Hex)	Access	Page	Name	Addr (0,Hex)	Access	Page	Name	Addr (0,Hex)	Access	Page
PRT0DR	00	RW	105		40			ASE10CR0	80	RW	139		C0		
PRT0IE	01	RW	106		41				81				C1		
PRT0GS	02	RW	107		42				82				C2		
PRT0DM2	03	RW	108		43				83				C3		
PRT1DR	04	RW	105		44			ASE11CR0	84	RW	139		C4		
PRT1IE	05	RW	106		45				85				C5		
PRT1GS	06	RW	107		46				86				C6		
PRT1DM2	07	RW	108		47				87				C7		
PRT2DR	08	RW	105		48				88			PWMVREF0	C8	RW	148
PRT2IE	09	RW	106		49				89			PWMVREF1	C9	RW	149
PRT2GS	0A	RW	107		4A				8A			IDAC_MODE	CA	RW	150
PRT2DM2	0B	RW	108		4B				8B			PWM_SRC	CB	RW	151
PRT3DR	0C	RW	105		4C				8C			TS_CR0	CC	RW	152
PRT3IE	0D	RW	106		4D				8D			TS_CMPH	CD	RW	153
PRT3GS	0E	RW	107		4E				8E			TS_CMPL	CE	RW	154
PRT3DM2	0F	RW	108		4F				8F			TS_CR1	CF	RW	155
PRT4DR	10	RW	105	CSD0_DR0_L	50	R	122		90			CUR_PP	D0	RW	156
PRT4IE	11	RW	106	CSD0_DR1_L	51	W	123		91			STK_PP	D1	RW	157
PRT4GS	12	RW	107	CSD0_CNT_L	52	RW	124		92				D2		
PRT4DM2	13	RW	108	CSD0_CR0	53	#	125		93			IDX_PP	D3	RW	158
	14			CSD0_DR0_H	54	R	122		94			MVR_PP	D4	RW	159
	15			CSD0_DR1_H	55	W	123		95			MVW_PP	D5	RW	160
	16			CSD0_CNT_H	56	RW	124		96			I2C0_CFG	D6	RW	161
	17			CSD0_CR1	57	#	126		97			I2C0_SCR	D7	#	162
	18			CSD1_DR0_L	58	R	122		98			I2C0_DR	D8	RW	164
	19			CSD1_DR1_L	59	W	123		99			I2C0_MSCR	D9	#	165
	1A			CSD1_CNT_L	5A	RW	124		9A			INT_CLR0	DA	RW	166
	1B			CSD1_CR0	5B	#	125		9B			INT_CLR1	DB	RW	168
	1C			CSD1_DR0_H	5C	R	122		9C			INT_CLR2	DC	RW	170
	1D			CSD1_DR1_H	5D	W	123		9D			INT_CLR3	DD	RW	171
	1E			CSD1_CNT_H	5E	RW	124		9E			INT_MSK3	DE	RW	172
	1F			CSD1_CR1	5F	#	126		9F			INT_MSK2	DF	RW	173
DBC00DR0	20	#	109	AMX_IN	60	RW	127		A0			INT_MSK0	E0	RW	174
DBC00DR1	21	W	110	AMUX_CFG	61	RW	128		A1			INT_MSK1	E1	RW	175
DBC00DR2	22	RW	111	PWM_CR	62	RW	129		A2			INT_VC	E2	RC	176
DBC00CR0	23	#	112	ARF_CR	63	RW	130		A3			RES_WDT	E3	W	177
DBC01DR0	24	#	109	CMP_CR0	64	#	131		A4				E4		
DBC01DR1	25	W	110		65				A5				E5		
DBC01DR2	26	RW	111	CMP_CR1	66	RW	132		A6			DEC_CR0	E6	RW	178
DBC01CR0	27	#	112		67				A7			DEC_CR1	E7	RW	179
DCC02DR0	28	#	109	ADC0_CR	68	#	133		A8			MUL0_X	E8	W	180
DCC02DR1	29	W	110	ADC1_CR	69	#	133		A9			MUL0_Y	E9	W	181
DCC02DR2	2A	RW	111	SADC_DH	6A	RW	134		AA			MUL0_DH	EA	R	182
DCC02CR0	2B	#	112	SADC_DL	6B	RW	135		AB			MUL0_DL	EB	R	183
DCC03DR0	2C	#	109	TMP_DR0	6C	RW	136		AC			ACC0_DR1	EC	RW	184
DCC03DR1	2D	W	110	TMP_DR1	6D	RW	136		AD			ACC0_DR0	ED	RW	185
DCC03DR2	2E	RW	111	TMP_DR2	6E	RW	136		AE			ACC0_DR3	EE	RW	186
DCC03CR0	2F	#	112	TMP_DR3	6F	RW	136		AF			ACC0_DR2	EF	RW	187
DBC10DR0	30	#	109		70			RDI0RI	B0	RW	140		F0		
DBC10DR1	31	W	110		71			RDI0SYN	B1	RW	141		F1		
DBC10DR2	32	RW	111	ACE00CR1	72	RW	137	RDI0IS	B2	RW	142		F2		
DBC10CR0	33	#	112	ACE00CR2	73	RW	138	RDI0LT0	B3	RW	143		F3		
DBC11DR0	34	#	109		74			RDI0LT1	B4	RW	144		F4		
DBC11DR1	35	W	110		75			RDI0RO0	B5	RW	145		F5		
DBC11DR2	36	RW	111	ACE01CR1	76	RW	137	RDI0RO1	B6	RW	146		F6		
DBC11CR0	37	#	112	ACE01CR2	77	RW	138	RDI0DSM	B7	RW	147	CPU_F	F7	i	188
DCC12DR0	38	#	109		78			RDI1RI	B8	RW	140		F8		
DCC12DR1	39	W	110		79			RDI1SYN	B9	RW	141		F9		
DCC12DR2	3A	RW	111		7A			RDI1IS	BA	RW	142		FA		
DCC12CR0	3B	#	112		7B			RDI1LT0	BB	RW	143		FB		
DCC13DR0	3C	#	109		7C			RDI1LT1	BC	RW	144	IDACR_D	FC	RW	189
DCC13DR1	3D	W	110		7D			RDI1RO0	BD	RW	145	IDACL_D	FD	RW	189
DCC13DR2	3E	RW	111		7E			RDI1RO1	BE	RW	146	CPU_SCR1	FE	#	190
DCC13CR0	3F	#	112		7F			RDI1DSM	BF	RW	147	CPU_SCR0	FF	#	191

Gray fields are reserved. # Access is bit specific.

Register Map Bank 1 Table: Configuration Space

Name	Addr (1,Hex)	Access	Page	Name	Addr (1,Hex)	Access	Page	Name	Addr (1,Hex)	Access	Page	Name	Addr (1,Hex)	Access	Page
PRT0DM0	00	RW	192		40			ASE10CR0	80	RW	139		C0		
PRT0DM1	01	RW	193		41				81				C1		
PRT0IC0	02	RW	194		42				82				C2		
PRT0IC1	03	RW	195		43				83				C3		
PRT1DM0	04	RW	192		44			ASE11CR0	84	RW	139		C4		
PRT1DM1	05	RW	193		45				85				C5		
PRT1IC0	06	RW	194		46				86				C6		
PRT1IC1	07	RW	195		47				87				C7		
PRT2DM0	08	RW	192		48				88				C8		
PRT2DM1	09	RW	193		49				89				C9		
PRT2IC0	0A	RW	194		4A				8A				CA		
PRT2IC1	0B	RW	195		4B				8B				CB		
PRT3DM0	0C	RW	192		4C				8C				CC		
PRT3DM1	0D	RW	193		4D				8D				CD		
PRT3IC0	0E	RW	194		4E				8E				CE		
PRT3IC1	0F	RW	195		4F				8F				CF		
PRT4DM0	10	RW	192	CMP0CR1	50	RW	210		90			GDI_O_IN	D0	RW	245
PRT4DM1	11	RW	193	CMP0CR2	51	RW	211		91			GDI_E_IN	D1	RW	246
PRT4IC0	12	RW	194		52				92			GDI_O_OU	D2	RW	247
PRT4IC1	13	RW	195	VDAC50CR0	53	RW	212		93			GDI_E_OU	D3	RW	248
	14			CMP1CR1	54	RW	210		94				D4		
	15			CMP1CR2	55	RW	211		95				D5		
	16				56				96				D6		
	17			VDAC51CR0	57	RW	212		97				D7		
	18			CSCMPCR0	58	RW	213		98			MUX_CR0	D8	RW	249
	19			CSCMPGOEN	59	RW	214		99			MUX_CR1	D9	RW	249
	1A			CSLUTCR0	5A	RW	215		9A			MUX_CR2	DA	RW	249
	1B			CMPCOLMUX	5B	RW	216		9B			MUX_CR3	DB	RW	249
	1C			CMPPWMCR	5C	RW	217		9C			IDAC_CR1	DC	RW	250
	1D			CMPFLTCR	5D	RW	218		9D			OSC_GO_EN	DD	RW	251
	1E			CMPCLK1	5E	RW	219		9E			OSC_CR4	DE	RW	252
	1F			CMPCLK0	5F	RW	220		9F			OSC_CR3	DF	RW	253
DBC00FN	20	RW	196	CLK_CR0	60	RW	221	GDI_O_IN_CR	A0	RW	231	OSC_CR0	E0	RW	254
DBC00IN	21	RW	198	CLK_CR1	61	RW	222	GDI_E_IN_CR	A1	RW	232	OSC_CR1	E1	RW	255
DBC00OU	22	RW	200	ABF_CR0	62	RW	223	GDI_O_OU_CR	A2	RW	233	OSC_CR2	E2	RW	256
DBC00CR1	23	RW	202	AMD_CR0	63	RW	224	GDI_E_OU_CR	A3	RW	234	VLT_CR	E3	RW	257
DBC01FN	24	RW	196	CMP_GO_EN	64	RW	225	RTC_H	A4	RW	235	VLT_CMP	E4	R	258
DBC01IN	25	RW	198		65			RTC_M	A5	RW	236	ADC0_TR	E5	RW	259
DBC01OU	26	RW	200	AMD_CR1	66	RW	226	RTC_S	A6	RW	237	ADC1_TR	E6	RW	259
DBC01CR1	27	RW	202	ALT_CR0	67	RW	227	RTC_CR	A7	RW	238	V2BG_TR	E7	RW	260
DCC02FN	28	RW	196		68			SADC_CR0	A8	RW	239	IMO_TR	E8	W	261
DCC02IN	29	RW	198		69			SADC_CR1	A9	RW	240	ILO_TR	E9	W	262
DCC02OU	2A	RW	200	AMUX_CFG1	6A	RW	229	SADC_CR2	AA	RW	241	BDG_TR	EA	RW	263
DCC02CR1	2B	RW	202	CLK_CR3	6B	RW	230	SADC_CR3TRIM	AB	RW	242	ECO_TR	EB	W	264
DCC03FN	2C	RW	196	TMP_DR0	6C	RW	136	SADC_CR4	AC	RW	243	MUX_CR4	EC	RW	249
DCC03IN	2D	RW	198	TMP_DR1	6D	RW	136	I2C0_ADDR	AD	RW	244		ED		
DCC03OU	2E	RW	200	TMP_DR2	6E	RW	136		AE				EE		
DCC03CR1	2F	RW	202	TMP_DR3	6F	RW	136		AF				EF		
DBC10FN	30	RW	196		70			RDI0RI	B0	RW	140		F0		
DBC10IN	31	RW	198		71			RDI0SYN	B1	RW	141		F1		
DBC10OU	32	RW	200	ACE00CR1	72	RW	137	RDI0IS	B2	RW	142		F2		
DBC10CR1	33	RW	202	ACE00CR2	73	RW	138	RDI0LT0	B3	RW	143		F3		
DBC11FN	34	RW	196		74			RDI0LT1	B4	RW	144		F4		
DBC11IN	35	RW	198		75			RDI0RO0	B5	RW	145		F5		
DBC11OU	36	RW	200	ACE01CR1	76	RW	137	RDI0RO1	B6	RW	146		F6		
DBC11CR1	37	RW	202	ACE01CR2	77	RW	138	RDI0DSM	B7	RW	147	CPU_F	F7	I	188
DCC12FN	38	RW	196		78			RDI1RI	B8	RW	140		F8		
DCC12IN	39	RW	198		79			RDI1SYN	B9	RW	141		F9		
DCC12OU	3A	RW	200		7A			RDI1IS	BA	RW	142	FLS_PR1	FA	RW	265
DCC12CR1	3B	RW	202		7B			RDI1LT0	BB	RW	143		FB		
DCC13FN	3C	RW	196		7C			RDI1LT1	BC	RW	144		FC		
DCC13IN	3D	RW	198		7D			RDI1RO0	BD	RW	145	IDAC_CR0	FD	RW	266
DCC13OU	3E	RW	200		7E			RDI1RO1	BE	RW	146	CPU_SCR1	FE	#	190
DCC13CR1	3F	RW	202		7F			RDI1DSM	BF	RW	147	CPU_SCR0	FF	#	191

Gray fields are reserved. # Access is bit specific.

12. Register Details

This chapter is a reference for all the PSoC device registers in address order, for Bank 0 and Bank 1. The most detailed descriptions of the PSoC registers are in the Register Definitions section of each chapter. The registers that are in both banks are incorporated with the Bank 0 registers, designated with an 'x', rather than a '0' preceding the comma in the address. Bank 0 registers are listed first and begin on page 105. Bank 1 registers are listed second and begin on page 192. A condensed view of all the registers is shown in the "Register Map Bank 0 Table: User Space" on page 101 and the "Register Map Bank 1 Table: Configuration Space" on page 102.

12.1 Maneuvering Around the Registers

For ease-of-use, this chapter has been formatted so that there is one register per page, although some registers use two pages. On each page, from top to bottom, there are four sections:

1. Register name and address (from lowest to highest).
2. Register table showing the bit organization, with reserved bits grayed out.
3. Written description of register specifics or links to additional register information.
4. Detailed register bit descriptions.

Use the register tables, in addition to the detailed register bit descriptions, to determine which bits are reserved for some smaller PSoC devices. Reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'.

Register Conventions

The following table lists the register conventions that are specific to this chapter.

Register Conventions

Convention	Example	Description
'x' in a register name	ACExxCR1	Multiple instances/address ranges of the same register
R	R : 00	Read register or bit(s)
W	W : 00	Write register or bit(s)
L	RL : 00	Logical register or bit(s)
C	RC : 00	Clearable register or bit(s)
00	RW : 00	Reset value is 0x00 or 00h
XX	RW : XX	Register is not reset
0,	0,04h	Register is in bank 0
1,	1,23h	Register is in bank 1
x,	x,F7h	Register exists in register bank 0 and register bank 1
2L	2L Column	Register bit table designation for PSoC devices with two column limited functionality
Empty, grayed-out table cell		Reserved bit or group of bits, unless otherwise stated

12.1.1 Register Naming Conventions

There are a few register naming conventions used in this manual to abbreviate repetitious register information by using a lower case 'x' in the register name. The convention to interpret these register names is as follows.

- For all registers, an ‘x’ before the comma in the address field indicates that the register can be accessed or written to no matter what bank is used. For example, the M8C flag register’s (CPU_F) address is ‘x,F7h’ meaning it is located in bank 0 and bank 1 at F7h.
- For digital block registers, the first ‘x’ in some register names represents either “B” for basic or “C” for communication. For rows of digital PSoC blocks and their registers, the second ‘x’ set represents <Prefix>mn<Suffix>, where m=row index, n=column index. Therefore, DCC12CR0 (written DxCxxCR0) is a digital communication register for a digital PSoC block in row 1 column 2.
- For digital row registers, the ‘x’ in the digital register’s name represents the digital row index. For example, if the RDIxIS register name encompasses four registers, there is one for each digital row index and unique address (RDI0IS, RDI1IS, RDI2IS, and RDI3IS).
- For analog column registers, the naming convention for the switched capacitor and continuous time registers and their arrays of PSoC blocks is <Prefix>mn<Suffix>, where m=row index, n=column index. Therefore, ASE11CR0 (written ASCxxCR2) is a register for an analog PSoC block in row 1 column 1

12.2 Bank 0 Registers

The following registers are all in bank 0 and are listed in address order. An ‘x’ before the comma in the register’s address indicates that the register can be accessed independent of the XIO bit in the CPU_F register. Registers that are in both Bank 0 and Bank 1 are listed in address order in Bank 0. For example, the RDIxLT1 register has an address of x,B4h and is in both Bank 0 and Bank 1.

12.2.1 PRTxDR

Port Data Register

Individual Register Names and Addresses:

PRT0DR : 0,00h PRT1DR : 0,04h PRT2DR : 0,08h PRT3DR : 0,0Ch
PRT4DR : 0,10h

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Data[7:0]							

This register allows for write or read access of the current logical equivalent of the voltage on the pin.

The CY8C22x45 and CY8C21345 have a 6-bit wide Port 4. The highest 2 bits of this register will return the last data bus value when read and should be masked off prior to using this information. For additional information, refer to the “Register Definitions” on page 72 in the GPIO chapter.

Bit	Name	Description
7:0	**Data[7:0]**	Write value to port or read value from port. Reads return the state of the pin, not the value in the PRTxDR register.

12.2.2 PRTxIE

Port Interrupt Enable Register

Individual Register Names and Addresses:

PRT0IE : 0,01h PRT1IE : 0,05h PRT2IE : 0,09h PRT3IE : 0,0Dh
PRT4IE : 0,11h

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Interrupt Enables[7:0]							

This register is used to enable or disable the interrupt enable internal to the GPIO block.

The CY8C22x45 and CY8C21345 have a 6-bit wide Port 4. The highest 2 bits of this register will return the last data bus value when read and should be masked off prior to using this information. For additional information, refer to the “Register Definitions” on page 72 in the GPIO chapter.

Bit	Name	Description
7:0	**Interrupt Enables[7:0]**	A bit set in this register will enable the corresponding port pin interrupt. 0 Port pin interrupt disabled for the corresponding pin. 1 Port pin interrupt enabled for the corresponding pin.

12.2.3 PRTxGS

Port Global Select Register

Individual Register Names and Addresses:

PRT0GS : 0,02h PRT1GS : 0,06h PRT2GS : 0,0Ah PRT3GS : 0,0Eh
PRT4GS : 0,12h

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Global Select[7:0]							

This register is used to select the block for connection to global inputs or outputs.

The CY8C22x45 and CY8C21345 have a 6-bit wide Port 4. The highest 2 bits of this register will return the last data bus value when read and should be masked off prior to using this information. For additional information, refer to the “Register Definitions” on page 72 in the GPIO chapter.

Bit	Name	Description
7:0	**Global Select[7:0]**	A bit set in this register will connect the corresponding port pin to an internal global bus. This connection is used to input or output digital signals to or from the digital blocks. 0 Global function disabled. The pin value is determined by the PRTxDR bit value and port configuration registers. 1 Global function enabled. Direction depends on mode bits for the pin (registers PRTxDM0, PRTxDM1, and PRTxDM2).

12.2.4 PRTxDM2

Port Drive Mode Bit 2 Register

Individual Register Names and Addresses:

PRT0DM2 : 0,03h PRT1DM2 : 0,07h PRT2DM2 : 0,0Bh PRT3DM2 : 0,0Fh
PRT4DM2 : 0,13h

	7	6	5	4	3	2	1	0
Access : POR	RW : FF							
Bit Name	Drive Mode 2[7:0]							

This register is one of three registers whose combined value determines the unique Drive mode of each bit in a GPIO port.

In this register, there are eight possible drive modes for each port pin. Three mode bits are required to select one of these modes, and these three bits are spread into three different registers (the PRTxDM0 register on page 192, the PRTxDM1 register on page 193, and the PRTxDM2 register). The bit position of the affected port pin (for example, Pin[2] in Port 0) is the same as the bit position of each of the three drive mode register bits that control the Drive mode for that pin (for example: PRT0DM0[2], PRT0DM1[2], and PRT0DM2[2]). The three bits from the three registers are treated as a group. These are referred to as DM2, DM1, and DM0, or together as DM[2:0].

All Drive mode bits are shown in the sub-table below ([**2**10] refers to the combination (in order) of bits in a given bit position); however, this register only controls the ***most significant bit (MSb)*** of the Drive mode.

The CY8C22x45 and CY8C21345 have a 6-bit wide Port 4. The highest 2 bits of this register will return the last data bus value when read and should be masked off prior to using this information. For additional information, refer to the "Register Definitions" on page 72 in the GPIO chapter.

Bit	Name	Description
7:0	**Drive Mode 2[7:0]**	Bit 2 of the Drive mode, for each pin of an 8-bit GPIO port.

[**2**10]	Pin Output High	Pin Output Low	Notes
000b	Strong	Resistive	
001b	Strong	Strong	
010b	High Z	High Z	Digital input enabled.
011b	Resistive	Strong	
100b	Slow + strong	High Z	
101b	Slow + strong	Slow + strong	
110b	High Z	High Z	Reset state. Digital input disabled for zero power.
111b	High Z	Slow + strong	I2C Compatible mode.

Note A bold digit, in the table above, signifies that the digit is used in this register.

12.2.5 DxCxxDR0

Digital Basic/Communication Type B Block Data Register 0

Individual Register Names and Addresses:

DBC00DR0 : 0,20h	DBC01DR0 : 0,24h	DCC02DR0 : 0,28h	DCC03DR0 : 0,2Ch
DBC10DR0 : 0,30h	DBC11DR0 : 0,34h	DCC12DR0 : 0,38h	DCC13DR0 : 0,3Ch

	7	6	5	4	3	2	1	0
Access : POR	R : 00							
Bit Name	Data[7:0]							

This register is the data register for a digital block.

The use of this register is dependent on which function is selected for its block. This selection is made in the FN[2:0] bits of the DxCxxFN register on page 196. (For the Timer, Counter, Dead Band, and CRCPRS functions, a read of the DxCxxDR0 register returns 00h and transfers DxCxxDR0 to DxCxxDR2.)

The naming convention for the digital basic/communication and control registers is as follows. The first ‘x’ in the digital register’s name represents either “B” for basic or “C” for communication. For rows of digital PSoC blocks and their registers, the second ‘x’ set represents <Prefix>mn<Suffix>, where m=row index, n=column index. Therefore, DBC21DR0 is a digital basic register for a digital PSoC block in row 2 column 1. Depending on the digital row characteristics of your PSoC device (see the table titled “PSoC Device Characteristics” on page 268), some addresses may not be available. For additional information, refer to the “Register Definitions” on page 300 in the Digital Blocks chapter.

Bit	Name	Description
7:0	Data[7:0]	Data for selected function.

Block Function	Register Function	DCC Only
Timer	Count Value	No
Counter	Count Value	No
Dead Band	Count Value	No
IPWM	Count Value	No
CRCPRS	LFSR *	No
SPIM	Shifter	Yes
SPIS	Shifter	Yes
TXUART	Shifter	Yes
RXUART	Shifter	Yes
DSM	Accumulation	Yes

* ***Linear Feedback Shift Register (LFSR)***

12.2.6 DxCxxDR1

Digital Basic/Communication Type B Block Data Register 1

Individual Register Names and Addresses:

DBC00DR1 : 0,21h	DBC01DR1 : 0,25h	DCC02DR1 : 0,29h	DCC03DR1 : 0,2Dh
DBC10DR1 : 0,31h	DBC11DR1 : 0,35h	DCC12DR1 : 0,39h	DCC13DR1 : 0,3Dh

	7	6	5	4	3	2	1	0
Access : POR	W : 00							
Bit Name	Data[7:0]							

This register is the data register for a digital block.

The use of this register is dependent on which function is selected for its block. This selection is made in the FN[2:0] bits of the DxCxxFN register on page 196. Refer to the DxCxxDR0 register on page 109 for naming convention and digital row availability information. For additional information, refer to the "Register Definitions" on page 300 in the Digital Blocks chapter.

Bit	Name	Description
7:0	Data[7:0]	Data for selected function.

Block Function	Register Function	DCC Only
Timer	Period	No
Counter	Period	No
Dead Band	Period	No
IPWM	Period	No
CRCPRS	Polynomial	No
SPIM	TX Buffer	Yes
SPIS	TX Buffer	Yes
TXUART	TX Buffer	Yes
RXUART	Not applicable	Yes
DSM	Initial Data	Yes

12.2.7 DxCxxDR2

Digital Basic/Communication Type B Block Data Register 2

Individual Register Names and Addresses:

DBC00DR2 : 0,22h	DBC01DR2 : 0,26h	DCC02DR2 : 0,2Ah	DCC03DR2 : 0,2Eh
DBC10DR2 : 0,32h	DBC11DR2 : 0,36h	DCC12DR2 : 0,3Ah	DCC13DR2 : 0,3Eh

	7	6	5	4	3	2	1	0
Access : POR	RW * : 00							
Bit Name	Data[7:0]							

This register is the data register for a digital block.

The use of this register is dependent on which function is selected for its block. This selection is made in the FN[2:0] bits of the DxCxxFN register on page 196. Refer to the DxCxxDR0 register on page 109 for naming convention and digital row availability information. For additional information, refer to the "Register Definitions" on page 300 in the Digital Blocks chapter.

* If the block is configured as SPIM, SPIS, or RXUART, this register is read only.

Bit	Name	Description
7:0	**Data[7:0]**	Data for selected function.

Block Function	Register Function	DCC Only
Timer	Capture/Compare	No
Counter	Compare	No
Dead Band	Buffer	No
IPWM	Compare	No
CRCPRS	Seed/Residue	No
SPIM	RX Buffer	Yes
SPIS	RX Buffer	Yes
TXUART	Not applicable	Yes
RXUART	RX Buffer	Yes
DSM	Density	Yes

12.2.8 DxCxxCR0 (Timer Control:000)

Digital Basic/Communication Type B Block Control Register 0

Individual Register Names and Addresses:

DBC00CR0 : 0,23h
DBC01CR0 : 0,27h
DCC02CR0 : 0,2Bh
DCC03CR0 : 0,2Fh
DBC10CR0 : 0,33h
DBC11CR0 : 0,37h
DCC12CR0 : 0,3Bh
DCC13CR0 : 0,3Fh

	7	6	5	4	3	2	1	0
Access : POR	RW:0000				RW:0	RW:0	RW:0	RW:0
Bit Name	KILL[3:0]				NPS	TC Pulse Width	Capture Int	Enable

This register is the Control register for a timer, if the DxCxxFN register is configured as a ‘000’.

Refer to the DxCxxDR0 register on page 109 for naming convention and digital row availability information. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 300 in the Digital Blocks chapter.

Bit	Name	Description
7:4	**KILL[3:0]**	Select signal for kill function 0: Low 1: High 2: BC 3: VC3 4: RI[0] 5: RI[1] 6: RI[2] 7: RI[3] 8: RO[0] 9: RO[1] 10: RO[2] 11: RO[3] 12: ACMP[0] 13: ACMP[1] 14: ACMP[2] 15: ACMP[3]
3	**NPS**	Negative phase selection. The comparison output will be updated only when block clock is 0.
2	TC Pulse Width	Primary output 0 Terminal Count pulse width is one-half a block clock. Supports a period value of 00h. 1Terminal Count pulse width is one full block clock. Bit1, Capture Int, 0 Interrupt is selected with Mode bit 0 in the Function (DxBxxFN) register. 1 Block interrupt is caused by a hardware capture event (overrides Mode bit 0 selection). Bit0, Enable, 0 Timer is not enabled. 1 Timer is enabled.

12.2.9 DxCxxCR0 (Counter Control:001)

Digital Basic/Communication Type B Block Control Register 0

Individual Register Names and Addresses:

DBC00CR0: 0,23h
DBC01CR0: 0,27h
DCC02CR0: 0,2Bh
DCC03CR0: 0,2Fh
DBC10CR0: 0,33h
DBC11CR0: 0,37h
DCC12CR0: 0,3Bh
DCC13CR0: 0,3Fh

	7	6	5	4	3	2	1	0
Access : POR	RW:0000				RW:0	RW:0	RW:0	RW:0
Bit Name	KILL[3:0]				NPS		DR2BufEN	Enable

This register is the Control register for a counter, if the DxCxxFN register is configured as a '001'.

Refer to the DxCxxDR0 register on page 109 for naming convention and digital row availability information. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 300 in the Digital Blocks chapter.

Bit	Name	Description
7:4	**KILL[3:0]**	Same as Timer function.
3	**NPS**	Same meaning as in Timer.
1	**DR2BufEn**	'1' to enable DR2 update buffer; that is, update DR2 only at TC when function is running.
0	**Enable**	0 Counter is not enabled. 1 Counter is enabled.

12.2.10 DxCxxCR0 (Dead Band Control:100)

Digital Basic/Communication Type B Block Control Register 0

Individual Register Names and Addresses:

DBC00CR0: 0,23h
DBC01CR0: 0,27h
DCC02CR0: 0,2Bh
DCC03CR0: 0,2Fh
DBC10CR0: 0,33h
DBC11CR0: 0,37h
DCC12CR0: 0,3Bh
DCC13CR0: 0,3Fh

	7	6	5	4	3	2	1	0
Access : POR						RW : 0	RW : 0	RW : 0
Bit Name						Bit Bang Clock	Bit Bang Mode	Enable

This register is the Control register for a dead band, if the DxCxxFN register is configured as a ‘100’.

Refer to the DxCxxDR0 register on page 109 for naming convention and digital row availability information. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 300 in the Digital Blocks chapter.

Bit	Name	Description	
2	**Bit Bang Clock**	When Bit Bang mode is enabled, the output of this register bit is substituted for the PWM reference. This register may be toggled by user firmware to generate PHI1 and PH2 output clocks with the programmed dead time.	
1	**Bit Bang Mode**	0	Dead Band Generator uses the previous block primary output as the input reference.
		1	Dead Band Generator uses the Bit Bang Clock register as the input reference.
0	**Enable**	0	Dead Band Generator is not enabled.
		1	Dead Band Generator is enabled.

12.2.11 DxCxxCR0 (CRCPRS Control:010)

Digital Basic/Communication Type B Block Control Register 0

Individual Register Names and Addresses:

DBC00CR0: 0,23h
DBC01CR0: 0,27h
DCC02CR0: 0,2Bh
DCC03CR0: 0,2Fh
DBC10CR0: 0,33h
DBC11CR0: 0,37h
DCC12CR0: 0,3Bh
DCC13CR0: 0,3Fh

	7	6	5	4	3	2	1	0
Access : POR	RW:0000					RW:0	RW:0	RW:0
Bit Name	KILL[3:0]					Shift Mode	Pass Mode	Enable

This register is the Control register for a CRCPRS, if the DxCxxFN register is configured as a '010'.

Refer to the DxCxxDR0 register on page 109 for naming convention and digital row availability information. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 300 in the Digital Blocks chapter.

Bit	Name	Description
7:4	**KILL[3:0]**	Same as Timer function.
2	**Shift Mode**	Forces CRCPRS forward bus to zero to complete shift function.
1	**Pass Mode**	If selected, the DATA input selection is driven directly to the primary output and the block interrupt output. The CLK input selection is driven directly to the auxiliary output. 0 Normal CRC/PRS outputs. 1 Outputs are overridden.
0	**Enable**	0 CRC/PRS is not enabled. 1 CRC/PRS is enabled.

12.2.12 DxCxxCR0 (IPWM Control:011)

Individual Register Names and Addresses:

DBC00CR0: 0,23h DBC01CR0: 0,27h DCC02CR0: 0,2Bh DCC03CR0: 0,2Fh
DBC10CR0: 0,33h DBC11CR0: 0,37h DCC12CR0: 0,3Bh DCC13CR0: 0,3Fh

	7	6	5	4	3	2	1	0
Access : POR	RW:0000				RW:0	RW:0	RW:0	RW:0
Bit Name	START[3:0]				NPS	KILL_INT	SWT	Enable

For additional information, refer to the "Register Definitions" on page 300 in the Digital Blocks chapter.

Bit	Name	Description
7:4	**START[3:0]**	When SWT is 0 in PPG mode: 0: Low 1: High 2: BC 3: VC3 4: RI[0] 5: RI[1] 6: RI[2] 7: RI[3] 8: RO[0] 9: RO[1] 10: RO[2] 11: RO[3] 12: ACMP[0] 13: ACMP[1] 14: ACMP[2] 15: ACMP[3] When the shot is onging, the new trigger (rising edge of 'START') has no effect. But it will launch a new shot when START keeps high at the shot ending.
3	**NPS**	Has similar meaning as other functions but it has effects to both FO1 & FO2. **Note** The IPWM function does not support NPS mode when integrated dead band function is enabled.
2	**KILL_INT**	0 KILL is not interrupt source. The interrupt follows the rising edge for primary output. 1 Set to select KILL as interrupt; it has highest priority.
1	**SWT**	0 Not in software trigger mode 1 Set software trigger mode.
0	**Enable**	Function 'EN' bit as other functions FO1 is comparison output (< or <=); FO2 is reversed version. They support dead band.

12.2.13 DCCxxCR0 (SPIM Control:0-110)

Digital Communication Type B Block Control Register 0

Individual Register Names and Addresses:

DCC02CR0: 0,2Bh DCC03CR0: 0,2Fh
DCC12CR0: 0,3Bh DCC13CR0: 0,3Fh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	R : 0	R : 0	R : 1	R : 0	RW : 0	RW : 0	RW : 0
Bit Name	LSb First	Overrun	SPI Complete	TX Reg Empty	RX Reg Full	Clock Phase	Clock Polarity	Enable

This register is the Control register for a SPIM, if the DxCxxFN register is configured as a '110'.

The LSb First, Clock Phase, and Clock Polarity bits are configuration bits and should never be changed once the block is enabled. They can be set at the same time that the block is enabled. Refer to the DxCxxDR0 register on page 109 for naming convention and digital row availability information. For additional information, refer to the "Register Definitions" on page 300 in the Digital Blocks chapter.

Bit	Name	Description
7	**LSb First**	This bit should not be changed during an SPI transfer. 0 Data is shifted out MSb first. 1 Data is shifted out LSb first.
6	**Overrun**	0 No overrun has occurred. 1 Overrun has occurred. Indicates that a new byte is received and loaded into the RX Buffer before the previous one is read. It is cleared on a read of this (CR0) register.
5	**SPI Complete**	0 Indicates that a byte may still be in the process of shifting out, or no transmission is active. 1 Indicates that a byte is shifted out and all associated clocks are generated. It is cleared on a read of this (CR0) register. Optional interrupt.
4	**TX Reg Empty**	Reset state and the state when the block is disabled is '1'. 0 Indicates that a byte is currently buffered in the TX register. 1 Indicates that a byte is written to the TX register and cleared on write of the TX Buffer (DR1) register. This is the default interrupt. This status is initially asserted on block enable; however, the TX Reg Empty interrupt will occur only after the first data byte is written and transferred into the shifter.
3	**RX Reg Full**	0 RX register is empty. 1 A byte is received and loaded into the RX register. It is cleared on a read of the RX Buffer (DR2) register.
2	**Clock Phase**	0 Data is latched on the leading clock edge. Data changes on the trailing edge (Modes 0, 1). 1 Data changes on the leading clock edge. Data is latched on the trailing edge (Modes 2, 3).
1	**Clock Polarity**	0 Non-inverted, clock idles low (Modes 0, 2). 1 Inverted, clock idles high (Modes 1, 3).
0	**Enable**	0 SPI Master is not enabled. 1 SPI Master is enabled.

12.2.14 DCCxxCR0 (SPIS Control:1-110)

Digital Communication Type B Block Control Register 0

Individual Register Names and Addresses:

DCC02CR0: 0,2Bh DCC03CR0: 0,2Fh
DCC12CR0: 0,3Bh DCC13CR0: 0,3Fh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	R : 0	R : 0	R : 1	R : 0	RW : 0	RW : 0	RW : 0
Bit Name	LSb First	Overrun	SPI Complete	TX Reg Empty	RX Reg Full	Clock Phase	Clock Polarity	Enable

This register is the Control register for a SPIS, if the DxCxxFN register is configured as a '110'.

The LSb First, Clock Phase, and Clock Polarity bits are configuration bits and should never be changed once the block is enabled. They can be set at the same time that the block is enabled. Refer to the DxCxxDR0 register on page 109 for naming convention and digital row availability information. For additional information, refer to the "Register Definitions" on page 300 in the Digital Blocks chapter.

Bit	Name	Description
7	**LSb First**	This bit should not be changed during an SPI transfer. 0 Data is shifted out MSb first. 1 Data is shifted out LSb first.
6	**Overrun**	0 No overrun has occurred. 1 Overrun has occurred. Indicates that a new byte is received and loaded into the RX Buffer before the previous one is read. It is cleared on a read of this (CR0) register.
5	**SPI Complete**	0 Indicates that a byte may still be in the process of shifting out, or no transmission is active. 1 Indicates that a byte is shifted out and all associated clocks are generated. It is cleared on a read of this (CR0) register. Optional interrupt.
4	**TX Reg Empty**	Reset state and the state when the block is disabled is '1'. 0 Indicates that a byte is currently buffered in the TX register. 1 Indicates that a byte is written to the TX register and cleared on write of the TX Buffer (DR1) register. This is the default interrupt. This status is initially asserted on block enable; however, the TX Reg Empty interrupt will occur only after the first data byte is written and transferred into the shifter.
3	**RX Reg Full**	0 RX register is empty. 1 A byte is received and loaded into the RX register. It is cleared on a read of the RX Buffer (DR2) register.
2	**Clock Phase**	0 Data is latched on the leading clock edge. Data changes on the trailing edge (Modes 0, 1). 1 Data changes on the leading clock edge. Data is latched on the trailing edge (Modes 2, 3).
1	**Clock Polarity**	0 Non-inverted, clock idles low (Modes 0, 2). 1 Inverted, clock idles high (Modes 1, 3).
0	**Enable**	0 SPI Slave is not enabled. 1 SPI Slave is enabled.

12.2.15 DxCxxCR0 (DSM Control:111)

Individual Register Names and Addresses:

DBC00CR0: 0,23h
DBC10CR0: 0,33h
DBC01CR0: 0,27h
DBC11CR0: 0,37h
DCC02CR0: 0,2Bh
DCC12CR0: 0,3Bh
DCC03CR0: 0,2Fh
DCC13CR0: 0,3Fh

	7	6	5	4	3	2	1	0
Access : POR	RW:0000							RW:0
Bit Name	KILL_SEL[3:0]							Enable

For additional information, refer to the “Register Definitions” on page 300 in the Digital Blocks chapter.

Bit	Name	Description
7:4	**KILL_SEL[3:0]**	Used to select KILL signal source 0: Low 1: High 2: BC 3: VC3 4: RI[0] 5: RI[1] 6: RI[2] 7: RI[3] 8: RO[0] 9: RO[1] 10: RO[2] 11: RO[3] 12: ACMP[0] 13: ACMP[1] 14: ACMP[2] 15: ACMP[3]
0	**Enable**	Function 'EN' bit as other functions FO1 is multiplication output; FO2 is always local DSM output.

12.2.16 DCCxxCR0 (UART Transmitter Control)

Digital Communication Type B Block Control Register 0

Individual Register Names and Addresses:

DCC02CR0: 0,2Bh DCC03CR0: 0,2Fh
DCC12CR0: 0,3Bh DCC13CR0: 0,3Fh

	7	6	5	4	3	2	1	0
Access : POR			R : 0	R : 1		RW : 0	RW : 0	RW : 0
Bit Name			TX Complete	TX Reg Empty		Parity Type	Parity Enable	Enable

This register is the Control register for a UART transmitter, if the DxCxxFN register is configured as a '101'.

Refer to the DxCxxDR0 register on page 109 for naming convention and digital row availability information. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 300 in the Digital Blocks chapter. For the Receive mode definition, refer to section 12.2.17 on page 121.

Bit	Name	Description	
5	**TX Complete**	0	Indicates that a byte may still be in the process of shifting out.
		1	Indicates that a byte is shifted out and all associated framing bits are generated. Optional interrupt. Cleared on a read of this (CR0) register.
4	**TX Reg Empty**	Reset state and the state when the block is disabled is '1'.	
		0	Indicates that a byte is currently buffered in the TX register.
		1	Indicates that a byte is written to the TX register and cleared on write of the TX Buffer register. This is the default interrupt. TX Reg Empty interrupt will occur only after the first data byte is written and transferred into the shifter.
2	**Parity Type**	0	Even parity
		1	Odd parity
1	**Parity Enable**	0	Parity is not enabled.
		1	Parity is enabled, frame includes parity bit.
0	**Enable**	0	Serial Transmitter is not enabled.
		1	Serial Transmitter is enabled.

12.2.17 DCCxxCR0 (UART Receiver Control)

Digital Communication Type B Block Control Register 0

Individual Register Names and Addresses:

DCC02CR0: 0,2Bh DCC03CR0: 0,2Fh
DCC12CR0: 0,3Bh DCC13CR0: 0,3Fh

	7	6	5	4	3	2	1	0
Access : POR	R : 0	R : 0	R : 0	R : 0	R : 0	RW : 0	RW : 0	RW : 0
Bit Name	Parity Error	Overrun	Framing Error	RX Active	RX Reg Full	Parity Type	Parity Enable	Enable

This register is the Control register for a UART receiver, if the DxCxxFN register is configured as a '101'.

Refer to the DxCxxDR0 register on page 109 for naming convention and digital row availability information. For additional information, refer to the "Register Definitions" on page 300 in the Digital Blocks chapter. For the transmit mode definition, refer to section 12.2.16 on page 120.

Bit	Name		Description
7	**Parity Error**	0	Indicates that no parity error has occurred.
		1	Valid when RX Reg Full is set, indicating that a parity error has occurred in the received byte and cleared on a read of this (CR0) register.
6	**Overrun**	0	Indicates that no overrun has occurred.
		1	Valid when RX Reg Full is set, indicating that the byte in the RX Buffer register has not been read before the next byte is loaded. It is cleared on a read of this (CR0) register.
5	**Framing Error**	0	Indicates no framing error has occurred.
		1	Valid when RX Reg Full is set, indicating that a framing error has occurred (a logic 0 was sampled at the STOP bit, instead of the expected logic 1). It is cleared on a read of this (CR0) register.
4	**RX Active**	0	Indicates that no reception is in progress.
		1	Indicates that a reception is in progress. It is set by the detection of a START bit and cleared at the ***sampling*** of the STOP bit.
3	**RX Reg Full**	0	Indicates that the RX Buffer register is empty.
		1	Indicates that a byte is received and transferred to the RX Buffer (DR2) register. This bit is cleared when the RX Buffer register (DR2) is read by the CPU. Interrupt source.
2	**Parity Type**	0	Even parity
		1	Odd parity
1	**Parity Enable**	0	Parity is not enabled.
		1	Parity is enabled, frame includes parity bit.
0	**Enable**	0	Serial Receiver is not enabled.
		1	Serial Receiver is enabled.

12.2.18 CSDx_DR0_L, CSDx_DR0_H

Individual Register Names and Addresses:

CSD0_DR0_L : 0,50h CSD0_DR0_H : 0,54h CSD1_DR0_L : 0,58h CSD1_DR0_H : 0,5Ch

	7	6	5	4	3	2	1	0
Access : POR	R: 00							
Bit Name	Data[7:0]							

CSDx_DR0 is a 16-bit read-only register. It is main down-count register in Timer mode, or LFSRs in PRS mode. Reading it will always gets 0s on CPU data bus. Reading any DR0_L or DR0_H will transfer all 16-bit data to CSDx_CNT registers in non-CSD mode.

For additional information, see “Register Definitions” on page 433 in the CSD Logic System chapter.

Bit	Name	Description
7:0	**Data[7:0]**	The high or low byte data for CSD0 or CSD1 **Function Mode** — **Register Function** PRS Mode — LFSR Timer Mode — Down-count Register

12.2.19 CSDx_DR1_L, CSDx_DR1_H

Individual Register Names and Addresses:

CSD0_DR1_L : 0,51h CSD0_DR1_H : 0,55h CSD1_DR1_L : 0,59h CSD1_DR1_H : 0,5Dh

	7	6	5	4	3	2	1	0
Access : POR	W: 00							
Bit Name	Data[7:0]							

CSDx_DR1 is a 16-bit write-only register. It is dual purpose: either polynomial register in PRS mode, or period register in Timer mode. Writing to it in Timer mode will transfer its data to DR0 register if timer function is disabled. In PRS mode, every time polynomial register (DR1) is updated, the DR0 will be set by fixed seed value (function is disabled). Always set mode bits first, then set other registers in PRS mode.

For additional information, see “Register Definitions” on page 433 in the CSD Logic System chapter.

Bit	Name	Description
7:0	**Data[7:0]**	The high or low byte data for different function mode in CSD0 or CSD1. **Function Mode** — **Register Function** PRS Mode — Polynomial Timer Mode — Period

12.2.20 CSDx_CNT_L, CSDx_CNT_H

Individual Register Names and Addresses:

CSD0_CNT_L : 0,52h CSD0_CNT_H : 0,56h CSD1_CNT_L : 0,5Ah CSD1_CNT_H : 0,5Eh

	7	6	5	4	3	2	1	0
Access : POR	RC: 00							
Bit Name	Data[7:0]							

CSDx_CNT is a 16-bit read-only register for up-count purpose in CSD mode or capture purpose in non-CSD mode. It can be reset. As an up-counter, it counts the number of input clock when 'Gate' is open (active). As a capture register, it captures the data in DR0 register at the rising edge of 'Load' signal.

For additional information, see “Register Definitions” on page 433 in the CSD Logic System chapter.

Bit	Name	Description
7:0	**Data[7:0]**	The high or low byte data for different function mode. **Function Mode** — **Register Function** CSD Mode — CSD count data Timer Mode — Captured data

12.2.21 CSDx_CR0

Individual Register Names and Addresses:

CSD0_CR0 : 0,53h CSD1_CR0 : 0,5Bh

	7	6	5	4	3	2	1	0
Access : POR	RW: 00							
Bit Name	M_SHOT[3:0]				CNT_RST	CSD	PRS	EN

CSDx_CR0 is a configuration register.

For additional information, see “Register Definitions” on page 433 in the CSD Logic System chapter.

Bit	Name	Description
7:4	**M_SHOT[3:0]**	Multi-shot control bit. 0000b CSD_Logic block will continuously work until function is disabled. Interrupt will happen periodically. Others After 'EN' bit is set, the function starts running. The interrupt will be generated from the numbers specified by these bits. Next, the 'EN' bit will be cleared automatically, then function disabled.
3	**CNT_RST**	Not a real register bit; write-only. Writing ‘0’ will have no effect. However, writing ‘1’ will reset CSDx_CNT.
2	**CSD**	‘1’ will output MSB of CSDx_DR0 (PRS mode) or CSD_CLK (Timer mode) to MUXCLK. It should cooperate with 'EN' bits to send MSB/CSD_CLK to MUXCLK.
1	**PRS**	CSD block works as PRS generator when it is 1. Otherwise, it is Timer function.
0	**EN**	Used to enable function running. 1 function enabled. The interrupt can be generated when comparison becomes true. Comparison is CSDx_DR0 equal to seed value in PRS mode, or CSDx_DR0 equal to 0s in timer mode. There is no interrupt when 'EN' is 0.

12.2.22 CSDx_CR1

Individual Register Names and Addresses:

CSD0_CR1 : 0,57h CSD1_CR1 : 0,5Fh

	7	6	5	4	3	2	1	0
Access : POR	RW: 00							
Bit Name	INT_TYPE	CSD_CKSEL[2:0]			CNT_CKSEL[1:0]		ACOL[1:0]	

CSDx_CR1 is a configuration register.

For additional information, see “Register Definitions” on page 433 in the CSD Logic System chapter.

Bit	Name	Description
7	**INT_TYPE**	Interrupt will happen at CSDx_DR0 compare true when INT_TYPE is 0. Otherwise, the interrupt will happen at the end of the last shot.
6:4	**CSD_CKSEL[2:0]**	Used to set CSD_CLK frequency (similar divider as CPUCLK generator): 000 SYSCLK/8 001 SYSCLK/4 010 SYSCLK/2 011 SYSCLK/1 100 SYSCLK/16 101 SYSCLK/32 110 SYSCLK/128 111 SYSCLK/256
3:2	**CNT_CKSEL[1:0]**	Used to set CNT_CLK frequency: 00 CSD_CLK/1 01 CSD_CLK/2 10 CSD_CLK/4 11 CSD_CLK/8
1:0	**ACOL[1:0]**	Indicates the analog column chosen for counter 'Gate' signal. **Note** The dividers for CSD_CLK and CNT_CLK are reset when function is disabled.

12.2.23 AMX_IN

Analog Input Select Register

Individual Register Names and Addresses:

AMX_IN: 0,60h

	7	6	5	4	3	2	1	0
Access : POR					RW : 0		RW : 0	
Bit Name					ACI1[1:0]		ACI0[1:0]	

This register controls the analog muxes that feed signals in from port pins into the analog column 0/1.

Note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 349 in the Two Column Limited Analog System chapter.

Bits	Name	Description
3:2	**ACI1[1:0]**	Selects the Analog Column Mux 1. For 1 column, these are even inputs. 00b ACM1 P0[0] 01b ACM1 P0[2] 10b ACM1 P0[4] 11b ACM1 P0[6] **Note** ACol1Mux (ABF_CR0, Address 1,62h) 0 AC1 = ACM1 1 AC1 = ACM0
1:0	**ACI0[1:0]**	Selects the Analog Column Mux 0. For 0 column, these are odd inputs. 00b ACM0 P0[1] 01b ACM0 P0[3] 10b ACM0 P0[5] 11b ACM0 P0[7]

12.2.24 AMUX_CFG

Analog Mux Configuration Register

Individual Register Names and Addresses:

AMUX_CFG : 0,61h

	7	6	5	4	3	2	1	0
Access : POR			RW : 0		RW : 0			RW : 0
Bit Name			INTCAP[1:0]		MUXCLK[2:0]			EN

This register is used to configure the clocked pre-charge mode of the analog multiplexer system.

Note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 426 in the I/O Analog Multiplexer chapter.

Bits	Name	Description
5:4	**INTCAP[1:0]**	Selects pins for static operation, even when the precharge clock is selected with MUXCLK[2:0]. The CY8C22x45 and CY8C21345 use pins P0[7] (connects to Mux Bus Right) and P0[5] (connects to Mux Bus Left) for this function. 00b Both P0[7] and P0[5] are in normal precharge configuration. 01b P0[5] pin selected for static mode only. 10b P0[7] pin selected for static mode only. 11b Both P0[7] and P0[5] are selected for static mode only.
3:1	**MUXCLK[2:0]**	Selects a precharge clock source for analog mux bus connections. It could be suppressed by the setting in Register AMUX_CFG1 in CY8C22x45. 000b Precharge clock is off, no switching. 001b VC1 010b VC2 011b Row0 Broadcast 100b Analog column clock 0 101b Analog column clock 1 110b Reserved 111b Reserved
0	**EN**	0 Disable MUXCLK output (disable all MUXCLK in CY8C22x45) 1 Enable MUXCLK output (enable all MUXCLK in CY8C22x45, but any MUXCLK still could be low by the setting in Register AMUX_CFG1)

12.2.25 PWM_CR

ADC PWM Control Register

Individual Register Names and Addresses:

PWM_CR: 0,62h

2L COLUMN	7	6	5	4	3	2	1	0
Access : POR			RW : 0			RW : 0		RW : 0
Bit Name			HIGH[2:0]			LOW[1:0]		PWMEN

This register controls the parameters for the dedicated ADC PWM. This PWM signal can be selected to gate one or more comparator bus signals (as enabled by bits 5:4 of the DEC_CR0 register).

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 349 in the Two Column Limited Analog System chapter.

When the HIGH[2:0] bits are configured with a value other than zero, this PWM source overrides the digital block sources for gating as defined by ICLKS3, ICLKS2, ICLKS1, and ICLKS0 in the DEC_CR0 and DEC_CR1 registers.

Bits	Name	Description	
5:3	HIGH[2:0]	000b	The dedicated PWM is not in use. The gating signal reverts to a digital block output as selected by the ICLKS bits in the DEC_CR0 and DEC_CR1 registers.
		001b	High time is 1 VC3 period.
		010b	High time is 2 VC3 periods.
		011b	High time is 4 VC3 periods.
		100b	High time is 8 VC3 periods.
		101b	High time is 16 VC3 periods.
		110b	Reserved
		111b	Reserved
2:1	LOW[1:0]	00b	No PWM low time, only the terminal count is generated.
		01b	Low time is 1 VC3 period.
		10b	Low time is 2 VC3 periods.
		11b	Low time is 3 VC3 periods.
0	PWMEN	0	Disable the dedicated PWM.
		1	Enable the dedicated PWM.

12.2.26 ARF_CR

Analog Reference Control Register

Individual Register Names and Addresses:

ARF_CR: 0,63h

	7	6	5	4	3	2	1	0
Access : POR								
Bit Name								

This register is used to configure various features of the configurable analog references.

In the table above, note that the reserved bit is a gray table cell and is not described in the bit description section below. Reserved bits should always be written with a value of '0'.

Bits	Name	Description

12.2.27 CMP_CR0

Analog Comparator Bus 0 Register

Individual Register Names and Addresses:

CMP_CR0: 0,64h

	7	6	5	4	3	2	1	0
Access : POR			R : 0				RW : 0	
Bit Name			COMP[1:0]				AINT[1:0]	

This register is used to poll the analog column comparator bits and select column interrupts.

Note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'.

Bits	Name	Description
5	**COMP[1]**	Comparator bus state for column 1. This bit is updated on the rising edge of PHI2, unless the comparator latch disable bits are set (refer to the CLDISx bits in the CMP_CR1 register). If the comparator latch disable bits are set, then this bit is transparent to the comparator bus in the analog array.
4	**COMP[0]**	Comparator bus state for column 0. This bit is updated on the rising edge of PHI2, unless the comparator latch disable bits are set (refer to the CLDISx bits in the CMP_CR1 register). If the comparator latch disable bits are set, then this bit is transparent to the comparator bus in the analog array.
1	**AINT[1]**	Controls the selection of the analog comparator interrupt for column 1. 0 The comparator data bit from the column is the input to the interrupt controller. 1 The terminal count for the dedicated incremental PWM is the input to the interrupt controller.
0	**AINT[0]**	Controls the selection of the analog comparator interrupt for column 0. 0 The comparator data bit from the column is the input to the interrupt controller. 1 The terminal count for the dedicated incremental PWM is the input to the interrupt controller.

12.2.28 CMP_CR1

Analog Comparator Bus 1 Register

Individual Register Names and Addresses:

CMP_CR1: 0,66h

	7	6	5	4	3	2	1	0
Access : POR			RW : 0	RW : 0				
Bit Name			CLDIS[1]	CLDIS[0]				

This register is used to override the analog column comparator synchronization, or select direct column clock synchronization for the CY8C22x45 and CY8C21345 PSoC devices.

By default, the analog comparator bus is synchronized by the column clock and driven to the digital comparator bus for use in the digital array and the interrupt controller. The CLDIS bits are used to bypass the synchronization. This bypass mode can be used in power down operation to wake the device out of sleep, as a result of an analog column interrupt. The comparator bus is updated to the rising edge of the selected column clock.

Note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'.

Bits	Name	Description
5	**CLDIS[1]**	Controls the comparator output latch, column 1. 0 Comparator bus synchronization is enabled. 1 Comparator bus synchronization is disabled.
4	**CLDIS[0]**	Controls the comparator output latch, column 0. 0 Comparator bus synchronization is enabled. 1 Comparator bus synchronization is disabled.*(continued on next page)*

12.2.29 ADCx_CR

ADC Column 0 and Column 1 Configuration Register

Individual Register Names and Addresses:

ADC0_CR : 0,68h ADC1_CR : 0,69h

	7	6	5	4	3	2	1	0
Access : POR	R : 0	RW : 0	RW : 0		RW : 0	RW : 0		RW : 0
Bit Name	CMPST	LOREN	SHEN		CBSRC	AUTO		ADCEN

This register controls the single slope ADC in each column.

ADC0_CR is the ADC column 0 configuration register and ADC1_CR is the ADC column 1 configuration register. Reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 349 in the Two Column Limited Analog System chapter.

Bit	Name	Description
7	**CMPST**	This bit is the state of the comparator at the end of an ADC conversion period (as defined by the falling edge of the gating PWM). It is read only. 0 The comparator tripped during the previous conversion ramp. 1 The comparator did not trip during the previous conversion ramp.
6	**LOREN**	This bit controls an approximate 4-to-1 range on the ADC current source. 0 Normal current range 1 Low current range
5	**SHEN**	Sample and Hold Enable. The sample and hold function is only applicable to the PMUX (positive) comparator input. 0 Disabled 1 Enabled
3	**CBSRC**	Digital Comparator Bus Source. There are two possible sources for the digital comparator bus in conjunction with ADC operation. 0 Digital comparator bus is driven with synchronized and gated analog comparator output. Implements a Counter Enable interface. 1 Digital comparator bus is driven with the selected PWM terminal count. Implements a Timer Capture interface.
2	**AUTO**	Auto ADC Mode. The bit allows for a periodic signal to control ADC sequencing. 0 Auto mode off. 1 Auto mode on. Set this bit for ADC operation. The voltage ramp generator and sample and hold circuitry are controlled by the selected PWM signal (digital block or dedicated PWM).
0	**ADCEN**	Enable. Configures the ADC for operation, power up. 0 Disabled, Powered Down. 1 Enabled

12.2.30 ADC_DH

Individual Register Names and Addresses:

ADC_DH : 0,6Ah

	7	6	5	4	3	2	1	0
Access : POR	RW:00							
Bit Name	ADC_DH							

For additional information, see “Register Definitions” on page 442 in the 10-bit SAR ADC Controller chapter.

Bit	Name	Description
7:0	**ADC_DH**	The highest eight bits of ADC data. In normal mode, it is read-only byte. In ADC test mode, it is readable/writeable byte. The ADC is treated as a 8-bit ADC if only read as ADC data.

12.2.31 ADC_DL

Individual Register Names and Addresses:

ADC_DL : 0,6Bh

	7	6	5	4	3	2	1	0
Access : POR							RW:00	
Bit Name							ADC_DL	

For additional information, see “Register Definitions” on page 442 in the 10-bit SAR ADC Controller chapter.

Bit	Name	Description
1:0	**ADC_DL**	The lowest two bits of ADC data. In normal mode, it is read-only byte. In ADC test mode, it is readable/writeable byte.

12.2.32 TMP_DRx

Temporary Data Register

Individual Register Names and Addresses:

TMP_DR0 : x,6Ch TMP_DR1 : x,6Dh TMP_DR2 : x,6Eh TMP_DR3 : x,6Fh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Data[7:0]							

This register is used to enhance the performance in multiple SRAM page PSoC devices.

All bits in this register are reserved for PSoC devices with 256 bytes of SRAM. Refer to the table titled “PSoC Device SRAM Availability” on page 53. For additional information, refer to the “Register Definitions” on page 56 in the RAM Paging chapter.

Bit	Name	Description
7:0	**Data[7:0]**	General purpose register space.

12.2.33 ACExxCR1

Analog Continuous Time Type E Block Control Register 1

Individual Register Names and Addresses:

ACE00CR1 : x,72h ACE01CR1 : x,76h

	7	6	5	4	3	2	1	0
Access : POR		RW : 0	RW : 0			RW : 0		
Bit Name		CompBus	NMux[2:0]			PMux[2:0]		

This register is one of two registers used to configure the type E continuous time PSoC block.

The register naming convention for arrays of PSoC blocks and their registers is <Prefix>mn<Suffix>, where m=row index, n=column index; therefore, ACE01CR1 is a register for an analog PSoC block in row 0 column 1.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 349 in the Two Column Limited Analog System chapter on page 338.

Bits **Name** **Description**

6 **CompBus**

Enable output to the comparator bus. The comparator bus is always driven from the CT block.
0 Disable output to comparator bus.
1 Enable output to comparator bus.

5:3 **NMux[2:0]**

Encoding for negative input select. Note that available mux inputs vary by individual PSoC block.

	ACE00	ACE01	
000b	ACE01	ACE00	
001b	VBG	VBG	
010b	Switch 5 & 7	Switch 5 & 7	
011b	Mux Bus[1]	Mux Bus[2]	Chip-wide analog mux bus.
100b	FB#	FB#	
101b	ASE10	ASE11	
110b	ASE11	ASE10	
111b	Port Inputs	Port Inputs	

1 Mux Bus Left for the CY8C22x45 and CY8C21345.
2 Mux Bus Right for the CY8C22x45 and CY8C21345.
Feedback. Gain = 1, configuration only.

2:0 **PMux[2:0]**

Encoding for positive input select. Note that available mux inputs vary by individual PSoC block.

	ACE00	ACE01	
000b	Reserved	V_{TEMP}	
001b	Port Inputs	Port Inputs	
010b	ACE01	ACE00	
011b	VBG	VBG	
100b	ASE10	ASE11	
101b	ASE11	ASE10	
110b	Switch 1 & 4	Switch 1 & 4	
111b	Mux Bus[1]	Mux Bus[2]	For the CY8C22x45, CY8C21345, CY8C21x34, CY7C603xx, and CYWUSB6953 only: Chip-wide analog mux bus.

1 Mux Bus Left for the CY8C22x45 and CY8C21345.
2 Mux Bus Right for the CY8C22x45 and CY8C21345.

12.2.34 ACExxCR2

Analog Continuous Time Type E Block Control Register 2

Individual Register Names and Addresses:

ACE00CR2 : x,73h ACE01CR2 : x,77h

	7	6	5	4	3	2	1	0
Access : POR							RW : 0	RW : 0
Bit Name							FullRange	PWR

This register is one of two registers used to configure the type E continuous time PSoC block.

The register naming convention for arrays of PSoC blocks and their registers is <Prefix>mn<Suffix>, where m=row index, n=column index; therefore, ACE01CR2 is a register for an analog PSoC block in row 0 column 1.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 349 in the Two Column Limited Analog System chapter on page 338.

Bits	Name	Description	
1	**FullRange**	0	Input range includes Vss but not Vdd.
		1	Rail-to-rail input range, with approximately 10 µA additional cell current.
0	**PWR**	0	Powers off both the CT and SC blocks in the column.
		1	Enables the column's analog blocks.

12.2.35 ASExxCR0

Analog Switch Cap Type E Block Control Register 0

Individual Register Names and Addresses:

ASE10CR0 : x,80h ASE11CR0 : x,84h

2L COLUMN	7	6	5	4	3	2	1	0
Access : POR	RW : 0							
Bit Name	FVal							

This register is used to configure a type E switched capacitor PSoC block.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 349 in the Two Column Limited Analog System chapter.

Bits	Name	Description
7	**FVal**	F Capacitor value selection bit. 0 Slower integration in the SC block (higher accuracy) 1 Faster integration (lower accuracy)

12.2.36 RDIxRI

Row Digital Interconnect Row Input Register

Individual Register Names and Addresses:

RDI0RI : x,B0h RDI1RI : x,B8h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0		RW : 0		RW : 0		RW : 0	
Bit Name	RI3[1:0]		RI2[1:0]		RI1[1:0]		RI0[1:0]	

This register is used to control the input mux that determines which global inputs will drive the row inputs.

The 'x' in the digital register's name represents the digital row index. Depending on the digital row characteristics of your PSoC device (see the table titled "PSoC Device Characteristics" on page 268), some addresses may not be available. For additional information, refer to the "Register Definitions" on page 282 in the Row Digital Interconnect chapter.

Bit	Name	Description
7:6	**RI3[1:0]**	Select source for row input 3. 00b GIE[3] 01b GIE[7] 10b GIO[3] 11b GIO[7]
5:4	**RI2[1:0]**	Select source for row input 2. 00b GIE[2] 01b GIE[6] 10b GIO[2] 11b GIO[6]
3:2	**RI1[1:0]**	Select source for row input 1. 00b GIE[1] 01b GIE[5] 10b GIO[1] 11b GIO[5]
1:0	**RI0[1:0]**	Select source for row input 0. 00b GIE[0] 01b GIE[4] 10b GIO[0] 11b GIO[4]

12.2.37 RDIxSYN

Row Digital Interconnect Synchronization Register

Individual Register Names and Addresses:

RDI0SYN : x,B1h RDI1SYN : x,B9h

	7	6	5	4	3	2	1	0
Access : POR					RW : 0	RW : 0	RW : 0	RW : 0
Bit Name					RI3SYN	RI2SYN	RI1SYN	RI0SYN

This register is used to control the input synchronization.

The 'x' in the digital register's name represents the digital row index. Depending on the digital row characteristics of your PSoC device (see the table titled "PSoC Device Characteristics" on page 268), some addresses may not be available. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 282 in the Row Digital Interconnect chapter.

Bit	Name		Description
3	**RI3SYN**	0	Row input 3 is synchronized to the SYSCLK system clock.
		1	Row input 3 is passed without synchronization.
2	**RI2SYN**	0	Row input 2 is synchronized to the SYSCLK system clock.
		1	Row input 2 is passed without synchronization.
1	**RI1SYN**	0	Row input 1 is synchronized to the SYSCLK system clock.
		1	Row input 1 is passed without synchronization.
0	**RI0SYN**	0	Row input 0 is synchronized to the SYSCLK system clock.
		1	Row input 0 is passed without synchronization.

12.2.38 RDIxIS

Row Digital Interconnect Input Select Register

Individual Register Names and Addresses:

RDI0IS : x,B2h RDI1IS : x,BAh

	7	6	5	4	3	2	1	0
Access : POR			RW : 0		RW : 0	RW : 0	RW : 0	RW : 0
Bit Name			BCSEL[1:0]		IS3	IS2	IS1	IS0

This register is used to configure the inputs to the digital row LUTS and select a broadcast driver from another row if present.

The ‘x’ in the digital register’s name represents the digital row index. Depending on the digital row characteristics of your PSoC device (see the table titled “PSoC Device Characteristics” on page 268), some addresses may not be available. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 282 in the Row Digital Interconnect chapter.

Bit	Name	Description
5:4	**BCSEL[1:0]**	When the BCSEL value is equal to the row number, the ***tri-state*** buffer that drives the row broadcast ***net*** from the input select mux is disabled, so that one of the row’s blocks may drive the local row broadcast net. 00b Row 0 drives row broadcast net. 01b Row 1 drives row broadcast net. Reserved for 1 row PSoC devices. 10b Row 2 drives row broadcast net. Reserved for 1 and 2 row PSoC devices. 11b Row 3 drives row broadcast net. Reserved for 1, 2, and 3 row PSoC devices.
3	**IS3**	0 The ‘A’ input of LUT3 is RO[3]. 1 The ‘A’ input of LUT3 is RI[3].
2	**IS2**	0 The ‘A’ input of LUT2 is RO[2]. 1 The ‘A’ input of LUT2 is RI[2].
1	**IS1**	0 The ‘A’ input of LUT1 is RO[1]. 1 The ‘A’ input of LUT1 is RI[1].
0	**IS0**	0 The ‘A’ input of LUT0 is RO[0]. 1 The ‘A’ input of LUT0 is RI[0].

12.2.39 RDIxLT0

Row Digital Interconnect Logic Table Register 0

Individual Register Names and Addresses:

RDI0LT0 : x,B3h RDI1LT0 : x,BBh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0				RW : 0			
Bit Name	LUT1[3:0]				LUT0[3:0]			

This register is used to select the logic function of the digital row LUTS.

The 'x' in the digital register's name represents the digital row index. Depending on the digital row characteristics of your PSoC device (see the table titled "PSoC Device Characteristics" on page 268), some addresses may not be available. For additional information, refer to the "Register Definitions" on page 282 in the Row Digital Interconnect chapter.

Bit	Name	Description
7:4	**LUT1[3:0]**	Select ***logic function*** for LUT1. **Function** 0h FALSE 1h A AND B 2h A AND $\overline{B}$ 3h A 4h $\overline{A}$ AND B 5h B 6h A XOR B 7h A OR B 8h A NOR B 9h A XNOR B Ah $\overline{B}$ Bh A OR $\overline{B}$ Ch $\overline{A}$ Dh $\overline{A}$ OR B Eh A NAND B Fh TRUE
3:0	**LUT0[3:0]**	Select logic function for LUT0. **Function** 0h FALSE 1h A AND B 2h A AND $\overline{B}$ 3h A 4h $\overline{A}$ AND B 5h B 6h A XOR B 7h A OR B 8h A NOR B 9h A XNOR B Ah $\overline{B}$ Bh A OR $\overline{B}$ Ch $\overline{A}$ Dh $\overline{A}$ OR B Eh A NAND B Fh TRUE

12.2.40 RDIxLT1

Row Digital Interconnect Logic Table Register 1

Individual Register Names and Addresses:

RDI0LT1 : x,B4h RDI1LT1 : x,BCh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0				RW : 0			
Bit Name	LUT3[3:0]				LUT2[3:0]			

This register is used to select the logic function of the digital row LUTS.

The 'x' in the digital register's name represents the digital row index. Depending on the digital row characteristics of your PSoC device (see the table titled "PSoC Device Characteristics" on page 268), some addresses may not be available. For additional information, refer to the "Register Definitions" on page 282 in the Row Digital Interconnect chapter.

Bit	Name	Description
7:4	**LUT3[3:0]**	Select logic function for LUT3. **Function** 0h FALSE 1h A AND B 2h A AND $\overline{B}$ 3h A 4h $\overline{A}$ AND B 5h B 6h A XOR B 7h A OR B 8h A NOR B 9h A XNOR B Ah $\overline{B}$ Bh A OR $\overline{B}$ Ch $\overline{A}$ Dh $\overline{A}$ OR B Eh A NAND B Fh TRUE
3:0	**LUT2[3:0]**	Select logic function for LUT2. **Function** 0h FALSE 1h A AND B 2h A AND $\overline{B}$ 3h A 4h $\overline{A}$ AND B 5h B 6h A XOR B 7h A OR B 8h A NOR B 9h A XNOR B Ah $\overline{B}$ Bh A OR $\overline{B}$ Ch $\overline{A}$ Dh $\overline{A}$ OR B Eh A NAND B Fh TRUE

12.2.41 RDIxRO0

Row Digital Interconnect Row Output Register 0

Individual Register Names and Addresses:

RDI0RO0 : x,B5h RDI1RO0 : x,BDh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	GOO5EN	GOO1EN	GOE5EN	GOE1EN	GOO4EN	GOO0EN	GOE4EN	GOE0EN

This register is used to select the global nets that the row outputs drive.

The 'x' in the digital register's name represents the digital row index. Depending on the digital row characteristics of your PSoC device (see the table titled "PSoC Device Characteristics" on page 268), some addresses may not be available. For additional information, refer to the "Register Definitions" on page 282 in the Row Digital Interconnect chapter.

Bit	Name	Description	
7	**GOO5EN**	0	Disable Row's LUT1 output to global output.
		1	Enable Row's LUT1 output to GOO[5].
6	**GOO1EN**	0	Disable Row's LUT1 output to global output.
		1	Enable Row's LUT1 output to GOO[1].
5	**GOE5EN**	0	Disable Row's LUT1 output to global output.
		1	Enable Row's LUT1 output to GOE[5].
4	**GOE1EN**	0	Disable Row's LUT1 output to global output.
		1	Enable Row's LUT1 output to GOE[1].
3	**GOO4EN**	0	Disable Row's LUT0 output to global output.
		1	Enable Row's LUT0 output to GOO[4].
2	**GOO0EN**	0	Disable Row's LUT0 output to global output.
		1	Enable Row's LUT0 output to GOO[0].
1	**GOE4EN**	0	Disable Row's LUT0 output to global output.
		1	Enable Row's LUT0 output to GOE[4].
0	**GOE0EN**	0	Disable Row's LUT0 output to global output.
		1	Enable Row's LUT0 output to GOE[0].

12.2.42 RDIxRO1

Row Digital Interconnect Row Output Register 1

Individual Register Names and Addresses:

RDI0RO1 : x,B6h RDI1RO1 : x,BEh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	GOO7EN	GOO3EN	GOE7EN	GOE3EN	GOO6EN	GOO2EN	GOE6EN	GOE2EN

This register is used to select the global nets that the row outputs drive.

The 'x' in the digital register's name represents the digital row index. Depending on the digital row characteristics of your PSoC device (see the table titled "PSoC Device Characteristics" on page 268), some addresses may not be available. For additional information, refer to the "Register Definitions" on page 282 in the Row Digital Interconnect chapter.

Bit	Name		Description
7	**GOO7EN**	0 1	Disable Row's LUT3 output to global output. Enable Row's LUT3 output to GOO[7].
6	**GOO3EN**	0 1	Disable Row's LUT3 output to global output. Enable Row's LUT3 output to GOO[3].
5	**GOE7EN**	0 1	Disable Row's LUT3 output to global output. Enable Row's LUT3 output to GOE[7].
4	**GOE3EN**	0 1	Disable Row's LUT3 output to global output. Enable Row's LUT3 output to GOE[3].
3	**GOO6EN**	0 1	Disable Row's LUT2 output to global output. Enable Row's LUT2 output to GOO[6].
2	**GOO2EN**	0 1	Disable Row's LUT2 output to global output. Enable Row's LUT2 output to GOO[2].
1	**GOE6EN**	0 1	Disable Row's LUT2 output to global output. Enable Row's LUT2 output to GOE[6].
0	**GOE2EN**	0 1	Disable Row's LUT2 output to global output. Enable Row's LUT2 output to GOE[2].

12.2.43 RDIxDSM

Row Digital Interconnect Delta Sigma Modulator Function Select Register

Individual Register Names and Addresses:

RDI0DSM : x,B7h RDI1DSM : x,BFh

	7	6	5	4	3	2	1	0
Access : POR		RW:000			RW:0000			
Bit Name		AVS_SEL[2:0]			AVG_EN[3:0]			

The Row Digital Interconnect Delta Sigma Modulator Register (RDIxDSM) is used to select the Delta Sigma Modulator function on the row outputs.

The ‘x’ in the digital register’s name represents the digital row index. Depending on the digital row characteristics of your PSoC device (see the table titled “PSoC Device Characteristics” on page 268), some addresses may not be available. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 282 in the Row Digital Interconnect chapter.

Bit	Name	Description
6:4	**AVS_SEL[2:0]**	Select 1 from 8 dig blocks's primary output as average-control signal.
3:0	**AVG_EN[3:0]**	Enables average function on corresponding row output channel.

12.2.44 PWMVREF0

Individual Register Names and Addresses:

PWMVREF0: 0,C8h

	7	6	5	4	3	2	1	0
Access : POR	RW:0	R:0	R:0	R:0	RW:0000			
Bit Name	EN	MODBIT[0]	SADC[0]	ADC_SOLSB	PWMDUTY			

For additional information, refer to the “Register Definitions” on page 449 in the MISC Logic chapter.

Bit	Name	Description
7	**EN**	‘1’ to enable the PWMVref 0 output. Otherwise it will keep 0.
6	**MODBIT[0]**	The MODBIT[0] signal to analog column0. It is only readable in TEST mode. It is for DFT only. Reading it will always get 0 in normal mode.
5	**SADC[0]**	The SADC[0] signal to analog column0 block. It is only readable in TEST mode. It is for DFT only. Reading it will always get 0 in normal mode.
4	**ADC_SOLSB**	The ADC_SOLSB signal. It is only readable in TEST mode. It is for DFT only. Reading it will always get 0 in normal mode.
3:0	**PWMDUTY**	From MIN 0 to MAX 15/16. The PWM period is fixed SYSCLK/8.

12.2.45 PWMVREF1

Individual Register Names and Addresses:

PWMVREF1: 0,C9h

	7	6	5	4	3	2	1	0
Access : POR	RW:0	R:0	R:0	R:0	RW:0000			
Bit Name	EN	MODBIT[1]	SADC[1]	ADC_S_H	PWMDUTY			

For additional information, refer to the “Register Definitions” on page 449 in the MISC Logic chapter.

Bit	Name	Description
7	**EN**	‘1’ to enable the PWMVref 1 output. Otherwise it will keep 0.
6	**MODBIT[1]**	The MODBIT[1] signal to analog column 1. It is only readable in TEST mode. It is for DFT only. Reading it will always get 0 in normal mode.
5	**SADC[1]**	The SADC[1] signal to analog column 1 block. It is only readable in TEST mode. It is for DFT only. Reading it will always get 0 in normal mode.
4	**ADC_S_H**	The ADC_S_H signal. It is only readable in TEST mode. It is for DFT only. Reading it will always get 0 in normal mode.
3:0	**PWMDUTY**	From MIN 0 to MAX 15/16. The PWM period is fixed SYSCLK/8.

12.2.46 IDAC_MODE

Individual Register Names and Addresses:

IDAC_MODE: 0,CAh

	7	6	5	4	3	2	1	0
Access : POR	RW:00		RW:00		RW:00		RW:00	
Bit Name	CSD1_GO_EN[1:0]		CSD0_GO_EN[1:0]		IDACR_MD		IDACL_MD	

For additional information, refer to the “Register Definitions” on page 449 in the MISC Logic chapter.

Bit	Name	Description	
7:6	**CSD1_GO_EN[1]**	1	enable CSD_CNTEN[1] go to GOO[7].
7:6	**CSD1_GO_EN[0]**	1	enable CSD_CNTEN[1] go to GOO[5].
5:4	**CSD0_GO_EN[1]**	1	enable CSD_CNTEN[0] go to GOO[6].
5:4	**CSD0_GO_EN[0]**	1	enable CSD_CNTEN[0] go to GOO[4].
3:2/1:0	**IDACR_MD[1:0]/IDACL_MD[1:0]**		
		00B	IDAC ON/OFF switch is always ON.
		01B	IDAC ON/OFF switch is following CSD_CNTEN in current CSD block.
		10B	IDAC ON/OFF switch is following CSD_EN in current CSD block.
		11B	IDAC ON/OFF switch is following CSD_CNTEN in another CSD block.

12.2.47 PWMSRC

Individual Register Names and Addresses:

PWMSRC: 0,CBh

	7	6	5	4	3	2	1	0
Access : POR	RWT:0	RT:0	RT:000			RW:000		
Bit Name	TSYNC	SAMP2	ADC_MUX_SEL[2:0]			DBSEL[2:0]		

For additional information, refer to the “Register Definitions” on page 449 in the MISC Logic chapter.

Bit	Name	Description
7	**TSYNC**	Only TMODE accessible 1 use test clock as external clock.
6	**SAMP2**	SAMP2 signal from TCSLEEP block. It is only readable in TMODE accessible; reading it in normal mode will always get 0. It is only for pure DFT purpose.
5:3	**ADC_MUX_SEL[2:0]**	ADC_MUX_SEL[2:0] signals to ADC comparator block. They are only readable in TMODE accessible; reading them in normal mode will always get 0s. They are only for pure DFT purpose.
2:0	**DBSEL[2:0]**	Select one DB FO1 as PWM signal.

12.2.48 TS_CR0

Individual Register Names and Addresses:

TS_CR0: 0,CCh

	7	6	5	4	3	2	1	0
Access : POR	RW:0000				RW:0	RW:0	RW:0	RW:0
Bit Name	TS_INCMP_SEL				INCMP_INV	INCMP_EN	CMPH_EN	CMPL_EN

For additional information, refer to the “Register Definitions” on page 449 in the MISC Logic chapter.

Bit	Name	Description
7:4	**TS_INCMP_SEL**	Indicates the external source (GIE[7:0) or internal source (CSCMP[1:0],ACMP[1:0]). 0H - 7H select GIE[0:7]. 8H - 9H select ACMP[0:1]. AH - BH select CSCMP[0:1].
3	**INCMP_INV**	1 use inverted version of INCMP.
2	**INCMP_EN**	1 enable INCMP trigger source.
1	**CMPH_EN**	1 enable high channel trigger source.
0	**CMPL_EN**	1 enable low channel trigger source. **Note** Enable both CMPH_EN and CMPL_EN to use 16-bit trigger source.

12.2.49 TS_CMPH

Individual Register Names and Addresses:

TS_CMPH: 0,CDh

	7	6	5	4	3	2	1	0
Access : POR	RW:00H							
Bit Name	TS_CMPH							

This register is used to set the compare value of high channel.

For additional information, refer to the “Register Definitions” on page 449 in the MISC Logic chapter.

Bit	Name	Description
7:0	**TS_CMPH**	The compare value of high channel

12.2.50 TS_CMPL

Individual Register Names and Addresses:

TS_CMPL: 0,CEh

	7	6	5	4	3	2	1	0
Access : POR	RW:00H							
Bit Name	TS_CMPL							

This register is used to set the compare value of low channel.

For additional information, refer to the “Register Definitions” on page 449 in the MISC Logic chapter.

Bit	Name	Description
7:0	**TS_CMPL**	The compare value of low channel

12.2.51 TS_CR1

Individual Register Names and Addresses:

TS_CR1: 0,CFh

	7	6	5	4	3	2	1	0
Access : POR		RW:000				RW:000		
Bit Name		TS_CMPH_SEL				TS_CMPL_SEL		

This register is used to select the digital block for high channel and low channel comparison.

For additional information, refer to the “Register Definitions” on page 449 in the MISC Logic chapter.

Bit	Name	Description
6:4	**TS_CMPH_SEL**	Indicates which digital block is selected for high channel comparison.
2:0	**TS_CMPL_SEL**	Indicates which digital block is selected for low channel comparison.

12.2.52 CUR_PP

Current Page Pointer Register

Individual Register Names and Addresses:

CUR_PP: 0,D0h

	7	6	5	4	3	2	1	0
Access : POR						RW : 0		
Bit Name						Page Bits[2:0]		

This register is used to set the effective SRAM page for normal memory accesses in a multi-SRAM page PSoC device.

This register is only used when a device has more than one page of SRAM, see the table titled “PSoC Device SRAM Availability” on page 53. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 56 in the RAM Paging chapter.

Bit	Name	Description
2:0	**Page Bits[2:0]**	These bits determine which SRAM Page is used for generic SRAM access. See the RAM Paging chapter on page 53 for more information. 000b SRAM Page 0 001b SRAM Page 1 010b SRAM Page 2 011b SRAM Page 3 100b SRAM Page 4 101b SRAM Page 5 110b SRAM Page 6 111b SRAM Page 7 **Note** A value beyond the available SRAM, for a specific PSoC device, should not be set.

12.2.53 STK_PP

Stack Page Pointer Register

Individual Register Names and Addresses:

STK_PP: 0,D1h

	7	6	5	4	3	2	1	0
Access : POR						RW : 0		
Bit Name						Page Bits[2:0]		

This register is used to set the effective SRAM page for stack memory accesses in a multi-SRAM page PSoC device.

This register is only used when a device has more than one page of SRAM, see the table titled “PSoC Device SRAM Availability” on page 53. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 56 in the RAM Paging chapter.

Bit	Name	Description
2:0	**Page Bits[2:0]**	These bits determine which SRAM Page is used to hold the stack. See the RAM Paging chapter on page 53 for more information. 000b SRAM Page 0 001b SRAM Page 1 010b SRAM Page 2 011b SRAM Page 3 100b SRAM Page 4 101b SRAM Page 5 110b SRAM Page 6 111b SRAM Page 7 **Note** A value beyond the available SRAM, for a specific PSoC device, should not be set.

12.2.54 IDX_PP

Indexed Memory Access Page Pointer Register

Individual Register Names and Addresses:

IDX_PP: 0,D3h

	7	6	5	4	3	2	1	0
Access : POR						RW : 0		
Bit Name						Page Bits[2:0]		

This register is used to set the effective SRAM page for indexed memory accesses in a multi-SRAM page PSoC device.

This register is only used when a device has more than one page of SRAM, see the table titled "PSoC Device SRAM Availability" on page 53. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 56 in the RAM Paging chapter.

Bit	Name	Description
2:0	**Page Bits[2:0]**	These bits determine which SRAM Page an indexed memory access operates on. See the "Register Definitions" on page 56 for more information on when this register is active. 000b SRAM Page 0 001b SRAM Page 1 010b SRAM Page 2 011b SRAM Page 3 100b SRAM Page 4 101b SRAM Page 5 110b SRAM Page 6 111b SRAM Page 7 **Note** A value beyond the available SRAM, for a specific PSoC device, should not be set.

12.2.55 MVR_PP

MVI Read Page Pointer Register

Individual Register Names and Addresses:

MVR_PP: 0,D4h

	7	6	5	4	3	2	1	0
Access : POR						RW : 0		
Bit Name						Page Bits[2:0]		

This register is used to set the effective SRAM page for MVI read memory accesses in a multi-SRAM page PSoC device.

This register is only used when a device has more than one page of SRAM, see the table titled “PSoC Device SRAM Availability” on page 53. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 56 in the RAM Paging chapter.

Bit	Name	Description
2:0	**Page Bits[2:0]**	These bits determine which SRAM Page a MVI Read instruction operates on. 000b SRAM Page 0 001b SRAM Page 1 010b SRAM Page 2 011b SRAM Page 3 100b SRAM Page 4 101b SRAM Page 5 110b SRAM Page 6 111b SRAM Page 7 **Note** A value beyond the available SRAM, for a specific PSoC device, should not be set.

12.2.56 MVW_PP

MVI Write Page Pointer Register

Individual Register Names and Addresses:

MVW_PP: 0,D5h

	7	6	5	4	3	2	1	0
Access : POR						RW : 0		
Bit Name						Page Bits[2:0]		

This register is used to set the effective SRAM page for MVI write memory accesses in a multi-SRAM page PSoC device.

This register is only used when a device has more than one page of SRAM, see the table titled “PSoC Device SRAM Availability” on page 53. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 56 in the RAM Paging chapter.

Bit	Name	Description
2:0	**Page Bits[2:0]**	These bits determine which SRAM Page a MVI Write instruction operates on. 000b SRAM Page 0 001b SRAM Page 1 010b SRAM Page 2 011b SRAM Page 3 100b SRAM Page 4 101b SRAM Page 5 110b SRAM Page 6 111b SRAM Page 7 **Note** A value beyond the available SRAM, for a specific PSoC device, should not be set.

12.2.57 I2C_CFG

I²C Configuration Register

Individual Register Names and Addresses:

I2C_CFG: 0,D6h

	7	6	5	4	3	2	1	0
Access : POR		RW : 0	RW : 0	RW : 0	RW : 0		RW : 0	RW : 0
Bit Name		PSelect	Bus Error IE	Stop IE	Clock Rate[1:0]		Enable Master	Enable Slave

This register is used to set the basic operating modes, baud rate, and selection of interrupts.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 396 in the I2C chapter.

Bit	Name	Description
6	**PSelect**	I2C Pin Select 0 P1[5] and P1[7] 1 P1[0] and P1[1] **Note** Read the I2C chapter for a discussion of the side effects of choosing the P1[0] and P1[1] pair of pins.
5	**Bus Error IE**	Bus Error Interrupt Enable 0 Disabled 1 Enabled. An interrupt is generated on the detection of a Bus Error.
4	**Stop IE**	Stop Interrupt Enable 0 Disabled 1 Enabled. An interrupt is generated on the detection of a Stop Condition.
3:2	**Clock Rate[1:0]**	00b 100K Standard Mode 01b 400K Fast Mode 10b 50K Standard Mode 11b Reserved
1	**Enable Master**	Writing a '0' to both the Enable Master and Enable Slave bits will hold the I2C hardware in reset. 0 Disabled 1 Enabled
0	**Enable Slave**	Writing a '0' to both the Enable Master and Enable Slave bits will hold the I2C hardware in reset. 0 Disabled 1 Enabled

12.2.58 I2C_SCR

I²C Status and Control Register

Individual Register Names and Addresses:

I2C_SCR: 0,D7h

	7	6	5	4	3	2	1	0
Access : POR	RC : 0	RC : 0	RC : 0	RW : 0	RC : 0	RW : 0	RC : 0	RC : 0
Bit Name	Bus Error	Lost Arb	Stop Status	ACK	Address	Transmit	LRB	Byte Complete

This register is used by both master and slave to control the flow of data bytes and to keep track of the bus state during a transfer.

Bits in this register are held in reset until one of the enable bits in I2C_CFG is set. For additional information, refer to the "Register Definitions" on page 396 in the I2C chapter.

Bit	Name	Description
7	**Bus Error**	0 This status bit must be cleared by firmware by writing a '0' to the bit position. It is never cleared by the hardware. 1 A misplaced Start or Stop condition was detected.
6	**Lost Arb**	0 This bit is set immediately on lost arbitration; however, it does not cause an interrupt. This status may be checked after the following Byte Complete interrupt. Any Start detect or a write to the Start or Restart generate bits (I2C_MSCR register), when operating in Master mode, will also clear the bit. 1 Lost Arbitration
5	**Stop Status**	0 This status bit must be cleared by firmware with write of '0' to the bit position. It is never cleared by the hardware. 1 A Stop condition was detected.
4	**ACK**	Acknowledge Out. This bit is automatically cleared by hardware on a Byte Complete event. 0 NACK the last received byte. 1 ACK the last received byte
3	**Address**	0 This status bit must be cleared by firmware with write of '0' to the bit position. 1 The received byte is a slave address.
2	**Transmit**	Transmit bit is set by firmware to define the direction of the byte transfer. Any Start detect or a write to the Start or Restart generate bits, when operating in Master mode, will also clear the bit. 0 Receive mode 1 Transmit mode
1	**LRB**	Last Received Bit. The value of the 9th bit in a Transmit sequence, which is the acknowledge bit from the receiver. Any Start detect or a write to the Start or Restart generate bits, when operating in Master mode, will also clear the bit. 0 Last transmitted byte was ACK'ed by the receiver. 1 Last transmitted byte was NACK'ed by the receiver.

(continued on next page)

12.2.58 I2C_SCR *(continued)*

0 **Byte Complete**

Transmit/Receive Mode:
0 No completed transmit/receive since last cleared by firmware. Any Start detect or a write to the Start or Restart generate bits, when operating in Master mode, will also clear the bit.

Transmit Mode:
1 Eight bits of data have been transmitted and an ACK or NACK has been received.

Receive Mode:
1 Eight bits of data have been received.

12.2.59 I2C_DR

I²C Data Register

Individual Register Names and Addresses:

I2C_DR: 0,D8h

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Data[7:0]							

This register provides read/write access to the Shift register.

This register is read only for received data and write only for transmitted data. For additional information, refer to the “Register Definitions” on page 396 in the I2C chapter.

Bit	Name	Description
7:0	**Data[7:0]**	Read received data or write data to transmit.

12.2.60 I2C_MSCR

I²C Master Status and Control Register

Individual Register Names and Addresses:

I2C_MSCR: 0,D9h

	7	6	5	4	3	2	1	0
Access : POR					R : 0	R : 0	RW : 0	RW : 0
Bit Name					Bus Busy	Master Mode	Restart Gen	Start Gen

This register implements I2C framing controls and provides Bus Busy status.

Bits in this register are held in reset until one of the enable bits in I2C_CFG is set. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 396 in the I2C chapter.

Bit	Name	Description
3	**Bus Busy**	This bit is set to the following. 0 When a Stop condition is detected (from any bus master). 1 When a Start condition is detected (from any bus master).
2	**Master Mode**	This bit is set/cleared by hardware when the device is operating as a master. 0 Stop condition detected, generated by this device. 1 Start condition detected, generated by this device.
1	**Restart Gen**	This bit is cleared by hardware when the Restart generation is complete. 0 Restart generation complete. 1 Generate a Restart condition.
0	**Start Gen**	This bit is cleared by hardware when the Start generation is complete. 0 Start generation complete. 1 Generate a Start condition and send a byte (address) to the I2C bus, if bus is not busy.

12.2.61 INT_CLR0

Interrupt Clear Register 0

Individual Register Names and Addresses:

INT_CLR0: 0,DAh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	VC3	Sleep	GPIO	Compare 1	Compare 0	Analog 1	Analog 0	V Monitor

This register is used to enable the individual interrupt sources' ability to clear posted interrupts.

When bits in this register are read, a '1' will be returned for every bit position that has a corresponding posted interrupt. When bits in this register are written with a '0' and ENSWINT is not set, posted interrupts will be cleared at the corresponding bit positions. If there was not a posted interrupt, there is no effect. When bits in this register are written with a '1' and ENSWINT is set, an interrupt is posted in the interrupt controller. Note that the ENSWINT bit is in the INT_MSK3 register on page 172.

Note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 63 in the Interrupt Controller chapter.

Bit	Name	Description	
7	**VC3**	Read 0	No posted interrupt for Variable Clock 3.
		Read 1	Posted interrupt present for Variable Clock 3.
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for Variable Clock 3.
6	**Sleep**	Read 0	No posted interrupt for sleep timer.
		Read 1	Posted interrupt present for sleep timer.
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for sleep timer.
5	**GPIO**	Read 0	No posted interrupt for general purpose inputs and outputs (pins).
		Read 1	Posted interrupt present for GPIO (pins).
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for general purpose inputs and outputs (pins).

(continued on next page)

12.2.61 INT_CLR0 *(continued)*

4	**Compare 1**	Read 0	No posted interrupt for compare columns.
		Read 1	Posted interrupt present for compare columns
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for compare columns.
3	**Compare 0**	Read 0	No posted interrupt for compare columns.
		Read 1	Posted interrupt present for compare columns
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for compare columns.
2	**Analog 1**	Read 0	No posted interrupt for analog columns.
		Read 1	Posted interrupt present for analog columns
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for analog columns.
1	**Analog 0**	Read 0	No posted interrupt for analog columns.
		Read 1	Posted interrupt present for analog columns
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for analog columns.
0	**V Monitor**	Read 0	No posted interrupt for supply voltage monitor.
		Read 1	Posted interrupt present for supply voltage monitor.
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for supply voltage monitor.

12.2.62 INT_CLR1

Interrupt Clear Register 1

Individual Register Names and Addresses:

INT_CLR1: 0,DBh

2 Rows	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	DCC13	DCC12	DBC11	DBC10	DCC03	DCC02	DBC01	DBC00

1 Row	7	6	5	4	3	2	1	0
Access : POR					RW : 0	RW : 0	RW : 0	RW : 0
Bit Name					DCC03	DCC02	DBC01	DBC00

This register is used to clear posted interrupts for digital blocks or generate interrupts.

When bits in this register are read, a ‘1’ will be returned for every bit position that has a corresponding posted interrupt. When bits in this register are written with a ‘0’ and ENSWINT is not set, posted interrupts will be cleared at the corresponding bit positions. If there was not a posted interrupt, there is no effect. When bits in this register are written with a ‘1’ and ENSWINT is set, an interrupt is posted in the interrupt controller. Note that the ENSWINT bit is in the INT_MSK3 register on page 172.

Use the register tables above, in addition to the detailed register bit descriptions below, to determine which bits are reserved for some smaller PSoC devices. Note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of ‘0’. For additional information, refer to the “Register Definitions” on page 63 in the Interrupt Controller chapter.

Bit	Name	Description
7	DCC13	Digital Communications Block type B, row 1, position 3. Read 0 No posted interrupt. Read 1 Posted interrupt present. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt.
6	DCC12	Digital Communications Block type B, row 1, position 2. Read 0 No posted interrupt. Read 1 Posted interrupt present. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt.
5	DBC11	Digital Basic Block type B, row 1, position 1. Read 0 No posted interrupt. Read 1 Posted interrupt present. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt.

(continued on next page)

12.2.62 INT_CLR1 *(continued)*

4	**DBC10**	Digital Basic Block type B, row 1, position 0. Read 0 No posted interrupt. Read 1 Posted interrupt present. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt.
3	**DCC03**	Digital Communications Block type B, row 0, position 3. Read 0 No posted interrupt. Read 1 Posted interrupt present. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt.
2	**DCC02**	Digital Communications Block type B, row 0, position 2. Read 0 No posted interrupt. Read 1 Posted interrupt present. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt.
1	**DBC01**	Digital Basic Block type B, row 0, position 1. Read 0 No posted interrupt. Read 1 Posted interrupt present. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt.
0	**DBC00**	Digital Basic Block type B, row 0, position 0. Read 0 No posted interrupt. Read 1 Posted interrupt present. Write 0 AND ENSWINT = 0 Clear posted interrupt if it exists. Write 1 AND ENSWINT = 0 No effect. Write 0 AND ENSWINT = 1 No effect. Write 1 AND ENSWINT = 1 Post an interrupt.

12.2.63 INT_CLR2

Interrupt Clear Register 2

Individual Register Names and Addresses:

INT_CLR2: 0,DCh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0				
Bit Name	RTC	CSD1	CSD0	SARADC				

This register is used to enable the individual interrupt sources' ability to clear posted interrupts for digital blocks.

When bits in this register are read, a '1' will be returned for every bit position that has a corresponding posted interrupt. When bits in this register are written with a '0' and ENSWINT is not set, posted interrupts will be cleared at the corresponding bit positions. If there was not a posted interrupt, there is no effect. When bits in this register are written with a '1' and ENSWINT is set, an interrupt is posted in the interrupt controller. Note that the ENSWINT bit is in the INT_MSK3 register on page 172. For additional information, refer to the "Register Definitions" on page 63 in the Interrupt Controller chapter.

Bit	Name	Description	
7	**RTC**	Read 0	No posted interrupt for RTC.
		Read 1	Posted interrupt present for RTC.
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for RTC.
6	**CSD1**	Read 0	No posted interrupt for CSD1.
		Read 1	Posted interrupt present for CSD1.
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for CSD1.
5	**CSD0**	Read 0	No posted interrupt for CSD0.
		Read 1	Posted interrupt present for CSD0.
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for CSD0.
4	**SARADC**	Read 0	No posted interrupt for SARADC.
		Read 1	Posted interrupt present for SARADC.
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for SARADC.

12.2.64 INT_CLR3

Interrupt Clear Register 3

Individual Register Names and Addresses:

INT_CLR3: 0,DDh

	7	6	5	4	3	2	1	0
Access : POR								RW : 0
Bit Name								I2C

This register is used to enable the I2C interrupt sources' ability to clear posted interrupts.

When bits in this register are read, a '1' will be returned for every bit position that has a corresponding posted interrupt. When bits in this register are written with a '0' and ENSWINT is cleared, any posted interrupt will be cleared. If there was not a posted interrupt, there is no effect. When bits in this register are written with a '1' and ENSWINT is set, an interrupt is posted in the interrupt controller. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 63 in the Interrupt Controller chapter.

Bit	Name	Description	
0	**I2C**	Read 0	No posted interrupt for I2C.
		Read 1	Posted interrupt present for I2C.
		Write 0 AND ENSWINT = 0	Clear posted interrupt if it exists.
		Write 1 AND ENSWINT = 0	No effect.
		Write 0 AND ENSWINT = 1	No effect.
		Write 1 AND ENSWINT = 1	Post an interrupt for I2C.

12.2.65 INT_MSK3

Interrupt Mask Register 3

Individual Register Names and Addresses:

INT_MSK3: 0,DEh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0							RW : 0
Bit Name	ENSWINT							I2C

This register is used to enable the I2C's ability to create pending interrupts and enable software interrupts.

When an interrupt is masked off, the mask bit is '0'. The interrupt will still post in the interrupt controller. Therefore, clearing the mask bit only prevents a posted interrupt from becoming a pending interrupt. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 63 in the Interrupt Controller chapter.

Bit	Name	Description	
7	**ENSWINT**	0	Disable software interrupts.
		1	Enable software interrupts.
0	**I2C**	0	Mask I2C interrupt
		1	Unmask I2C interrupt

12.2.66 INT_MSK2

Interrupt Mask Register 2

Individual Register Names and Addresses:

INT_MSK2: 0,DFh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	RTC	CSD1	CSD0	SARADC				

This register is used to enable the individual sources' ability to create pending interrupts for digital blocks.

When an interrupt is masked off in this register, the mask bit is '0'. The interrupt will still post in the interrupt controller. Therefore, clearing the mask bit only prevents a posted interrupt from becoming a pending interrupt. For additional information, refer to the "Register Definitions" on page 63 in the Interrupt Controller chapter.

Bit	Name	Description	
7	**RTC**	0	Mask RTC interrupt.
		1	Unmask RTC interrupt.
6	**CSD1**	0	Mask CSD1 interrupt.
		1	Unmask CSD1 interrupt.
5	**CSD0**	0	Mask CSD0 interrupt.
		1	Unmask CSD0 interrupt.
4	**SARADC**	0	Mask SARADC interrupt.
		1	Unmask SARADC interrupt.

12.2.67 INT_MSK0

Interrupt Mask Register 0

Individual Register Names and Addresses:

INT_MSK0: 0,E0h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	VC3	Sleep	GPIO	Compare 1	Compare 0	Analog 1	Analog 0	V Monitor

This register is used to enable the individual sources' ability to create pending interrupts.

This register is used to enable the individual sources' ability to create pending interrupts. When an interrupt is masked off, the mask bit is '0'. The interrupt will still post in the interrupt controller. Therefore, clearing the mask bit only prevents a posted interrupt from becoming a pending interrupt. Note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 63 in the Interrupt Controller chapter.

Bit	Name	Description	
7	**VC3**	0 1	Mask VC3 interrupt. Unmask VC3 interrupt.
6	**Sleep**	0 1	Mask sleep interrupt. Unmask sleep interrupt.
5	**GPIO**	0 1	Mask GPIO interrupt. Unmask GPIO interrupt.
4	**Compare 1**	0 1	Mask compare interrupt, column 1. Unmask compare interrupt.
3	**Compare 0**	0 1	Mask compare interrupt, column 0. Unmask compare interrupt.
2	**Analog 1**	0 1	Mask analog interrupt, column 1. Unmask analog interrupt.
1	**Analog 0**	0 1	Mask analog interrupt, column 0. Unmask analog interrupt.
0	**V Monitor**	0 1	Mask voltage monitor interrupt. Unmask voltage monitor interrupt.

12.2.68 INT_MSK1

Interrupt Mask Register 1

Individual Register Names and Addresses:

INT_MSK1: 0,E1h

2 COLUMN	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	DCC13	DCC12	DBC11	DBC10	DCC03	DCC02	DBC01	DBC00

1 COLUMN	7	6	5	4	3	2	1	0
Access : POR					RW : 0	RW : 0	RW : 0	RW : 0
Bit Name					DCC03	DCC02	DBC01	DBC00

This register is used to enable the individual sources' ability to create pending interrupts for digital blocks.

When an interrupt is masked off, the mask bit is '0'. The interrupt will still post in the interrupt controller. Therefore, clearing the mask bit only prevents a posted interrupt from becoming a pending interrupt. Use the register tables above, in addition to the detailed register bit descriptions below, to determine which bits are reserved for some smaller PSoC devices. Note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 63 in the Interrupt Controller chapter.

Bit	Name	Description	
7	DCC13	0	Mask Digital Communication Block, row 1, position 3 interrupt.
		1	Unmask Digital Communication Block, row 1, position 3 interrupt.
6	DCC12	0	Mask Digital Communication Block, row 1, position 2 interrupt.
		1	Unmask Digital Communication Block, row 1, position 2 interrupt.
5	DBC11	0	Mask Digital Basic Block, row 1, position 1interrupt.
		1	Unmask Digital Basic Block, row 1, position 1 interrupt.
4	DBC10	0	Mask Digital Basic Block, row 1, position 0 interrupt.
		1	Unmask Digital Basic Block, row 1, position 0 interrupt.
3	DCC03	0	Mask Digital Communication Block, row 0, position 3 off.
		1	Unmask Digital Communication Block, row 0, position 3.
2	DCC02	0	Mask Digital Communication Block, row 0, position 2 off.
		1	Unmask Digital Communication Block, row 0, position 2.
1	DBC01	0	Mask Digital Basic Block, row 0, position 1 off.
		1	Unmask Digital Basic Block, row 0, position 1.
0	DBC00	0	Mask Digital Basic Block, row 0, position 0 off.
		1	Unmask Digital Basic Block, row 0, position 0.

12.2.69 INT_VC

Interrupt Vector Clear Register

Individual Register Names and Addresses:

INT_VC: 0,E2h

	7	6	5	4	3	2	1	0
Access : POR	RC : 00							
Bit Name	Pending Interrupt[7:0]							

This register returns the next pending interrupt and clears all pending interrupts when written.

For additional information, refer to the “Register Definitions” on page 63 in the Interrupt Controller chapter.

Bit	Name	Description	
7:0	**Pending Interrupt[7:0]**	Read	Returns vector for highest priority pending interrupt.
		Write	Clears all pending and posted interrupts.

12.2.70 RES_WDT

Reset Watchdog Timer Register

Individual Register Names and Addresses:

RES_WDT: 0,E3h

	7	6	5	4	3	2	1	0
Access : POR	W : 00							
Bit Name	WDSL_Clear[7:0]							

This register is used to clear the watchdog timer and clear both the watchdog timer and the sleep timer.

For additional information, refer to the “Register Definitions” on page 90 in the Sleep and Watchdog chapter.

Bit	Name	Description
7:0	**WDSL_Clear[7:0]**	Any write clears the watchdog timer. A write of 38h clears both the watchdog and sleep timers.

12.2.71 DEC_CR0

Decimator Control Register 0

Individual Register Names and Addresses:

DEC_CR0: 0,E6h

	7	6	5	4	3	2	1	0
Access : POR			RW : 0		RW : 0			
Bit Name			IGEN[1:0]		ICLKS0			

This register contains control bits to access hardware support for ADC operation.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'.

Bits	Name	Description
5:4	**IGEN[1:0]**	Incremental/SSADC Gate Enable. Selects on a column basis which comparator outputs will be gated with the SSADC selected PWM source. 1h Analog Column 0 2h Analog Column 1
3	**ICLKS0**	Incremental/SSADC Gate Source. Along with bits ICLKS2 and ICLKS1 in the DEC_CR1 register, this bit selects any one of the digital blocks in your device. The bit value for a digital block number that does not exist in a specific PSoC should be considered reserved. For example, a PSoC device with 1 row may choose any block numbered 0x, but not a block numbered 1x. **Note** The CY8C22x45 and CY8C21345 PSoC devices also contain a dedicated ADC PWM. When this PWM source is configured, it overrides the ICLKS0 through ICLKS2 digital block source.
3	*(cont.)*	**ICLKS2, ICLKS1 (see the DEC_CR1 register), ICLKS0** 000b Digital block 02 001b Digital block 12 010b Digital block 01 011b Digital block 11 100b Digital block 00 101b Digital block 10 110b Digital block 03 111b Digital block 13

12.2.72 DEC_CR1

Decimator Control Register 1

Individual Register Names and Addresses:

DEC_CR1: 0,E7h

	7	6	5	4	3	2	1	0
Access : POR				RW : 0	RW : 0			
Bit Name				ICLKS2	ICLKS1			

This register is used to configure signals for ADC operation.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'.

Bits	Name	Description
4:3	**ICLKSx**	Incremental/SSADC Gate Source. Along with ICLKS0 in DEC_CR0, selects any one of the digital blocks in your device. The bit value for a digital block number that does not exist in a specific PSoC should be considered reserved. For example, a PSoC device with 1 row may choose any block numbered 0x, but not a block numbered 1x. **Note** The CY8C22x45 and CY21345 PSoC devices also contain a dedicated ADC PWM. When this PWM source is configured, it overrides the ICLKS0 through ICLKS2 digital block source. **ICLKS2, ICLKS1, ICLKS0 (see the DEC_CR0 register)** 000b Digital block 02 001b Digital block 12 010b Digital block 01 011b Digital block 11 100b Digital block 00 101b Digital block 10 110b Digital block 03 111b Digital block 13

12.2.73 MULx_X

Multiply Input X Register

Individual Register Names and Addresses:

MUL0_X : 0,E8h

	7	6	5	4	3	2	1	0
Access : POR	W : XX							
Bit Name	Data[7:0]							

This register is one of two multiplicand registers for the signed 8-bit multiplier in the PSoC MAC.

This register is for 2 MAC block PSoC devices only. For additional information, refer to the “Register Definitions” on page 388 in the Multiply Accumulate chapter.

Bit	Name	Description
7:0	**Data[7:0]**	X multiplicand for MAC 8-bit multiplier.

12.2.74 MULx_Y

Multiply Input Y Register

Individual Register Names and Addresses:

MUL0_Y : 0,E9h

	7	6	5	4	3	2	1	0
Access : POR	W : XX							
Bit Name	Data[7:0]							

This register is one of two multiplicand registers for the signed 8-bit multiplier in the PSoC MAC.

This register is for 2 MAC block PSoC devices only. For additional information, refer to the “Register Definitions” on page 388 in the Multiply Accumulate chapter.

Bit	Name	Description
7:0	**Data[7:0]**	Y multiplicand for MAC 8-bit multiplier.

12.2.75 MULx_DH

Multiply Result High Byte Register

Individual Register Names and Addresses:

MUL0_DH : 0,EAh

	7	6	5	4	3	2	1	0
Access : POR	R : XX							
Bit Name	Data[7:0]							

This register holds the most significant byte of the 16-bit product.

This register is for 2 MAC block PSoC devices only. For additional information, refer to the “Register Definitions” on page 388 in the Multiply Accumulate chapter.

Bit	Name	Description
7:0	**Data[7:0]**	High byte of MAC multiplier 16-bit product.

12.2.76 MULx_DL

Multiply Result Low Byte Register

Individual Register Names and Addresses:

MUL0_DL : 0,EBh

	7	6	5	4	3	2	1	0
Access : POR	R : XX							
Bit Name	Data[7:0]							

This register holds the least significant byte of the 16-bit product.

This register is for 2 MAC block PSoC devices only. For additional information, refer to the “Register Definitions” on page 388 in the Multiply Accumulate chapter.

Bit	Name	Description
7:0	**Data[7:0]**	Low byte of MAC multiplier 16-bit product.

12.2.77 MACx_X/ACCx_DR1

Accumulator Data Register 1

Individual Register Names and Addresses:

MAC0_X/ACC0_DR1 : 0,ECh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Data[7:0]							

This is the multiply accumulate X register and the second byte of the accumulated value.

This register is for 2 MAC block PSoC devices only. For additional information, refer to the “Register Definitions” on page 388 in the Multiply Accumulate chapter.

Bit	Name	Description	
7:0	**Data[7:0]**	Read	Returns the 2nd byte of the 32-bit accumulated value. The 2nd byte is next to the least significant byte for the accumulated value.
		Write	X multiplicand for the MAC 16-bit multiply and 32-bit accumulator.

12.2.78 MACx_Y/ACCx_DR0

Accumulator Data Register 0

Individual Register Names and Addresses:

MAC0_Y/ACC0_DR0 : 0,EDh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Data[7:0]							

This is the multiply accumulate Y register and the first byte of the accumulated value.

This register is for 2 MAC block PSoC devices only. For additional information, refer to the “Register Definitions” on page 388 in the Multiply Accumulate chapter.

Bit	Name	Description	
7:0	**Data[7:0]**	Read	Returns the 1st byte of the 32-bit accumulated value. The 1st byte is the least significant byte for the accumulated value.
		Write	Y multiplicand for the MAC 16-bit multiply and 32-bit accumulate.

12.2.79 MACx_CL0/ACCx_DR3

Accumulator Data Register 3

Individual Register Names and Addresses:

MAC0_CL0/ACC0_DR3 : 0,EEh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Data[7:0]							

This is an accumulator clear register and the fourth byte of the accumulated value.

This register is for 2 MAC block PSoC devices only. For additional information, refer to the “Register Definitions” on page 388 in the Multiply Accumulate chapter.

Bit	Name	Description	
7:0	**Data[7:0]**	Read	Returns the 4th byte of the 32-bit accumulated value. The 4th byte is the ***most significant byte (MSB)*** for the accumulated value.
		Write	Writing any value to this address will clear all four bytes of the Accumulator.

12.2.80 MACx_CL1/ACCx_DR2

Accumulator Data Register 2

Individual Register Names and Addresses:

MAC0_CL1/ACC0_DR2 : 0,EFh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Data[7:0]							

This is an accumulator clear register and the third byte of the accumulated value.

This register is for 2 MAC block PSoC devices only. For additional information, refer to the “Register Definitions” on page 388 in the Multiply Accumulate chapter.

Bit	Name	Description	
7:0	**Data[7:0]**	Read	Returns the 3rd byte of the 32-bit accumulated value. The 3rd byte is the next to most significant byte for the accumulated value.
		Write	Writing any value to this address will clear all four bytes of the Accumulator.

12.2.81 CPU_F

M8C Flag Register

Individual Register Names and Addresses:

CPU_F: x,F7h

	7	6	5	4	3	2	1	0
Access : POR	RL : 0			RL : 0		RL : 0	RL : 0	RL : 0
Bit Name	PgMode[1:0]			XIO		Carry	Zero	GIE

This register provides read access to the M8C flags.

The AND f, expr; OR f, expr; and XOR f, expr flag instructions can be used to modify this register. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 42 in the M8C chapter and the "Register Definitions" on page 63 in the Interrupt Controller chapter.

Bit	Name	Description
7:6	**PgMode[1:0]**	00b Direct Address mode and Indexed Address mode operands are referred to RAM Page 0, regardless of the values of CUR_PP and IDX_PP. Note that this condition prevails on entry to an Interrupt Service Routine when the CPU_F register is cleared. 01b Direct Address mode instructions are referred to page 0. Indexed Address mode instructions are referred to the RAM page specified by the stack page pointer, STK_PP. 10b Direct Address mode instructions are referred to the RAM page specified by the current page pointer, CUR_PP. Indexed Address mode instructions are referred to the RAM page specified by the index page pointer, IDX_PP. 11b Direct Address mode instructions are referred to the RAM page specified by the current page pointer, CUR_PP. Indexed Address mode instructions are referred to the RAM page specified by the stack page pointer, STK_PP.
4	**XIO**	0 Normal register address space 1 Extended register address space. Primarily used for configuration.
2	**Carry**	Set by the M8C CPU Core to indicate whether there has been a carry in the previous logical/arithmetic operation. 0 No carry 1 Carry
1	**Zero**	Set by the M8C CPU Core to indicate whether there has been a zero result in the previous logical/arithmetic operation. 0 Not equal to zero 1 Equal to zero
0	**GIE**	0 M8C will not process any interrupts. 1 Interrupt processing enabled.

12.2.82 IDACx_D

Analog Mux DAC Data Register

Individual Register Names and Addresses:

IDACR_D : 0,FCh IDACL_D : 0,FDh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	DACDATA[7:0]							

The Analog Mux right DAC Data Register (IDACR_D) and the Analog Mux left DAC Data Register (IDACL_D) specify the 8-bit multiplying factor that determines the output DAC current.

This register is used by the CY8C21x45 and CY8C22x45 PSoC devices. For additional information, refer to the “Register Definitions” on page 426 in the I/O Analog Multiplexer chapter.

Bits	Name	Description
7:0	**DACDATA[7:0]**	This 8-bit value selects the number of current units that combine to form the DAC current. This current then drives the analog mux bus when DAC mode is enabled in the MXDACCR register. For example, a setting of 80h means that the charging current will be 128 current units. The current unit size depends on the range setting in the IDAC_CR0 register.

12.2.83 CPU_SCR1

System Status and Control Register 1

Individual Register Names and Addresses:

CPU_SCR1: x,FEh

	7	6	5	4	3	2	1	0
Access : POR	R : 0			RW : 0	R : 0	RW : 0		RW : 0
Bit Name	IRESS			SLIMO	ECO EXW	ECO EX		IRAMDIS

This register is used to convey the status and control of events related to internal resets and watchdog reset.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 49 in the SROM chapter or "Register Definitions" on page 82 of the External Crystal Oscillator (ECO) chapter.

Notes

1. Refer to the "PSoC Device Distinctions" on page 75, in the Internal Main Oscillator chapter, for more information on bit 4, SLIMO.
2. Bits 3 and 2 (ECO EXW and ECO EX, respectively) cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24x23 and CY8C22x13 PSoC devices.

Bit	Name	Description
7	**IRESS**	This bit is read only. 0 Boot phase only executed once. 1 Boot phase occurred multiple times.
4	**SLIMO**	Reduces frequency of the internal main oscillator (IMO). This bit is reserved on PSoC devices that do not support the slow IMO (see the "Architectural Description" on page 75). 0 IMO produces 24 MHz 1 Slow IMO (6 MHz)
3	**ECO EXW**	ECO Exists Written. 1 The ECO Exists Written bit has been written with a '1' or '0' and is now locked. 0 The ECO Exists Written bit has never been written in User mode.
2	**ECO EX**	ECO Exists (write once – see the explanation in "Register Definitions" on page 90). 1 ECO operation exists (set/reset OSC_CR[7] to enable/disable). 0 ECO operation does not exist. 32 kHz clock source is locked to operate from the ILO.
0	**IRAMDIS**	0 SRAM is initialized to 00h after POR, XRES, and WDR. 1 Address 03h - D7h of SRAM Page 0 are not modified by WDR.

12.2.84 CPU_SCR0

System Status and Control Register 0

Individual Register Names and Addresses:

CPU_SCR0: x,FFh

	7	6	5	4	3	2	1	0
Access : POR	R : 0		RC : 0	RC : 1	RW : 0			RW : 0
Bit Name	GIES		WDRS	PORS	Sleep			STOP

This register is used to convey the status and control of events for various functions of a PSoC device.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 90 in the Sleep and Watchdog chapter.

Bit	Name	Description
7	**GIES**	Global interrupt enable status. It is recommended that the user read the Global Interrupt Enable Flag bit from the CPU_F register on page 188. This bit is Read Only for GIES. Its use is discouraged, as the Flag register is now readable at address x,F7h (read only).
5	**WDRS**	Watchdog Reset Status. This bit may not be set by user code; however, it may be cleared by writing it with a '0'. 0 No Watchdog Reset has occurred. 1 Watchdog Reset has occurred.
4	**PORS**	Power On Reset Status. This bit may not be set by user code; however, it may be cleared by writing it with a '0'. 0 Power On Reset has not occurred and watchdog timer is enabled. 1 Will be set after external reset or Power On Reset.
3	**Sleep**	Set by the user to enable the CPU sleep state. CPU will remain in Sleep mode until any interrupt is pending. 0 Normal operation 1 Sleep
0	**STOP**	0 M8C is free to execute code. 1 M8C is halted. Can only be cleared by POR, XRES, or WDR.

12.3 Bank 1 Registers

The following registers are all in bank 1 and are listed in address order. Registers that are in both Bank 0 and Bank 1 are listed in address order in the section titled “Bank 0 Registers” on page 105.

12.3.1 PRTxDM0

Port Drive Mode Bit Register 0

Individual Register Names and Addresses:

PRT0DM0 : 1,00h PRT1DM0 : 1,04h PRT2DM0 : 1,08h PRT3DM0 : 1,0Ch
PRT4DM0 : 1,10h

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Drive Mode 0[7:0]							

This register is one of three registers whose combined value determines the unique Drive mode of each bit in a GPIO port.

In register PRTxDM0 there are eight possible drive modes for each port pin. Three mode bits are required to select one of these modes, and these three bits are spread into three different registers (PRTxDM0, “PRTxDM1” on page 193, and “PRTxDM2” on page 108). The bit position of the effected port pin (for example, Pin[2] in Port 0) is the same as the bit position of each of the three Drive Mode register bits that control the Drive mode for that pin (for example, Bit[2] in PRT0DM0, bit[2] in PRT0DM1, and bit[2] in PRT0DM2). The three bits from the three registers are treated as a group. These are referred to as DM2, DM1, and DM0, or together as DM[2:0].

All Drive mode bits are shown in the sub-table below ([21**0**] refers to the combination (in order) of bits in a given bit position); however, this register only controls the ***least significant bit (LSb)*** of the Drive mode.

The CY8C22x45 and CY8C21345 have a 6-bit wide Port 4. The highest 2 bits of this register will return the last data bus value when read and should be masked off prior to using this information. For additional information, refer to the “Register Definitions” on page 72 in the GPIO chapter.

Bit	Name	Description
7:0	**Drive Mode 0[7:0]**	Bit 0 of the Drive mode, for each of 8-port pins, for a GPIO port.

[21**0**]	Pin Output High	Pin Output Low	Notes
00**0**b	Strong	Resistive	
00**1**b	Strong	Strong	
01**0**b	High Z	High Z	Digital input enabled.
01**1**b	Resistive	Strong	
10**0**b	Slow + strong	High Z	
10**1**b	Slow + strong	Slow + strong	
11**0**b	High Z	High Z	Reset state. Digital input disabled for zero power.
11**1**b	High Z	Slow + strong	I2C Compatible mode.

Note A bold digit, in the table above, signifies that the digit is used in this register.

12.3.2 PRTxDM1

Port Drive Mode Bit Register 1

Individual Register Names and Addresses:

PRT0DM1 : 1,01h PRT1DM1 : 1,05h PRT2DM1 : 1,09h PRT3DM1 : 1,0Dh
PRT4DM1 : 1,11h

	7	6	5	4	3	2	1	0
Access : POR	RW : FF							
Bit Name	Drive Mode 1[7:0]							

This register is one of three registers whose combined value determines the unique Drive mode of each bit in a GPIO port.

In register PRTxDM1 there are eight possible drive modes for each port pin. Three mode bits are required to select one of these modes, and these three bits are spread into three different registers ("PRTxDM0" on page 192, PRTxDM1, and "PRTxDM2" on page 108). The bit position of the effected port pin (for example, Pin[2] in Port 0) is the same as the bit position of each of the three Drive Mode register bits that control the Drive mode for that pin (for example, Bit[2] in PRT0DM0, bit[2] in PRT0DM1, and bit[2] in PRT0DM2). The three bits from the three registers are treated as a group. These are referred to as DM2, DM1, and DM0, or together as DM[2:0].

All Drive mode bits are shown in the sub-table below ([2**1**0] refers to the combination (in order) of bits in a given bit position); however, this register only controls the middle bit of the Drive mode.

The CY8C22x45 and CY8C21345 have a 6-bit wide Port 4. The highest 2 bits of this register will return the last data bus value when read and should be masked off prior to using this information. For additional information, refer to the "Register Definitions" on page 72 in the GPIO chapter.

Bit	Name	Description
7:0	**Drive Mode 1[7:0]**	Bit 1 of the Drive mode, for each of 8-port pins, for a GPIO port.

[2**1**0]	Pin Output High	Pin Output Low	Notes
0**0**0b	Strong	Resistive	
0**0**1b	Strong	Strong	
0**1**0b	High Z	High Z	Digital input enabled.
0**1**1b	Resistive	Strong	
1**0**0b	Slow + strong	High Z	
1**0**1b	Slow + strong	Slow + strong	
1**1**0b	High Z	High Z	Reset state. Digital input disabled for zero power.
1**1**1b	High Z	Slow + strong	I2C Compatible mode.

Note A bold digit, in the table above, signifies that the digit is used in this register.

12.3.3 PRTxIC0

Port Interrupt Control Register 0

Individual Register Names and Addresses:

PRT0IC0 : 1,02h PRT1IC0 : 1,06h PRT2IC0 : 1,0Ah PRT3IC0 : 1,0Eh
PRT4IC0 : 1,12h

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Interrupt Control 0[7:0]							

This register is one of two registers whose combined value determine the unique Interrupt mode of each bit in a GPIO port.

In register PRTxIC0 there are four possible interrupt modes for each port pin. Two mode bits are required to select one of these modes and these two bits are spread into two different registers (PRTxIC0 and “PRTxIC1” on page 195). The bit position of the effected port pin (for example, Pin[2] in Port 0) is the same as the bit position of each of the interrupt control register bits that control the Interrupt mode for that pin (for example, Bit[2] in PRT0IC0 and bit[2] in PRT0IC1). The two bits from the two registers are treated as a group. In the sub-table below, “[**0**]” refers to the combination (in order) of bits in a given position, one bit from PRTxIC1 and one bit from PRTxIC0.

The CY8C22x45 and CY8C21345 have a 6-bit wide Port 4. The highest 2 bits of this register will return the last data bus value when read and should be masked off prior to using this information. For additional information, refer to the “Register Definitions” on page 72 in the GPIO chapter.

Bit	Name	Description	
7:0	**Interrupt Control 0[7:0]**	[1**0**]	**Interrupt Type**
		0**0**b	Disabled
		0**1**b	Low
		1**0**b	High
	‘	1**1**b	Change from last read
		Note A bold digit, in the table above, signifies that the digit is used in this register.	

12.3.4 PRTxIC1

Port Interrupt Control Register 1

Individual Register Names and Addresses:

PRT0IC1 : 1,03h PRT1IC1 : 1,07h PRT2IC1 : 1,0Bh PRT3IC1 : 1,0Fh
PRT4IC1 : 1,13h

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	Interrupt Control 1[7:0]							

This register is one of two registers whose combined value determine the unique Interrupt mode of each bit in a GPIO port.

In register PRTxIC1 there are four possible interrupt modes for each port pin. Two mode bits are required to select one of these modes and these two bits are spread into two different registers ("PRTxIC0" on page 194 and PRTxIC1). The bit position of the effected port pin (for example, Pin[2] in Port 0) is the same as the bit position of each of the interrupt control register bits that control the Interrupt mode for that pin (for example, Bit[2] in PRT0IC0 and bit[2] in PRT0IC1). The two bits from the two registers are treated as a group. In the sub-table below, "[**1**]" refers to the combination (in order) of bits in a given position, one bit from PRTxIC1 and one bit from PRTxIC0.

The CY8C22x45 and CY8C21345 have a 6-bit wide Port 4. The highest 2 bits of this register will return the last data bus value when read and should be masked off prior to using this information. For additional information, refer to the "Register Definitions" on page 72 in the GPIO chapter.

Bit	Name	Description	
7:0	**Interrupt Control 1[7:0]**	[**1**0]	**Interrupt Type**
		00b	Disabled
		01b	Low
		10b	High
	'	**1**1b	Change from last read
		Note A bold digit, in the table above, signifies that the digit is used in this register.	

12.3.5 DxCxxFN

Digital Basic/Communications Type B Block Function Register

Individual Register Names and Addresses:

DBC00FN : 1,20h
DBC10FN : 1,30h
DBC01FN : 1,24h
DBC11FN : 1,34h
DCC02FN : 1,28h
DCC12FN : 1,38h
DCC03FN : 1,2Ch
DCC13FN : 1,3Ch

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0		RW : 0		
Bit Name	Data Invert	BCEN	End Single	Mode[1:0]		Function[2:0]		

This register contains the primary Mode and Function bits that determine the function of the block.

Before changing any of the configuration registers (DxCxxFN, DxCxxIN, and DxCxxOU), disable the corresponding digital block by setting bit 0 in the CR0 or DxCxxCR0 register to '0'. The values in the DxCxxFN register should not be changed while the block is enabled. After all configuration changes are made, enable the block by setting bit 0 in the DxCxxCR0 register to '1'.

The naming convention for this register is as follows. The first 'x' in the digital register's name represents either "B" for basic or "C" for communication. For rows of digital PSoC blocks and their registers, the second 'x' set represents <Prefix>mn<Suffix>, where m=row index, n=column index. Therefore, DCC12FN is a digital communication register for a digital PSoC block in row 1 column 2. Depending on the digital row characteristics of your PSoC device, some addresses may not be available. For additional information, refer to the "Register Definitions" on page 300 in the Digital Blocks chapter.

Bit	Name	Description
7	**Data Invert**	0 Data input is non-inverted. 1 Data input is inverted.
6	**BCEN**	Enable Primary Function Output to drive the broadcast net. 0 Disable 1 Enable
5	**End Single**	0 Block is not the end of a chained function or the function is not chainable. 1 Block is the end of a chained function or a standalone block in a chainable function.
4:3	**Mode[1:0]**	These bits are function dependent and are described by function as follows.
	Timer or Counter:	Mode[0] signifies the interrupt type. 0 Interrupt on Terminal Count 1 Interrupt on Compare True Mode[1] signifies the compare type. 0 Compare on Less Than or Equal 1 Compare on Less Than
	CRCPRS:	Mode[1:0] are encoded as the Compare Type. 00b Compare on Equal 01b Compare on Less Than or Equal 10b Reserved 11b Compare on Less Than

(continued on next page)

12.3.5 DxCxxFN *(continued)*

4:3 *(cont.)*	IPWM	Same as Dead Band that follows:
	Dead Band:	Mode[1:0] are encoded as the Kill Type. 00b Synchronous Restart KILL mode 01b Disable KILL mode 10b Asynchronous KILL mode 11b Reserved
	UART:	Mode[0] signifies the Direction. 0 Receiver 1 Transmitter Mode[1] signifies the Interrupt Type. 0 Interrupt on TX Reg Empty 1 Interrupt on TX Complete
	SPI:	Mode[0] signifies the Type. 0 Master 1 Slave Mode[1] signifies the Interrupt Type. 0 Interrupt on TX Reg Empty 1 Interrupt on SPI Complete
	DSM	DSM Kill Mode 0B: “KILL Async” Mode 1B: “Kill Disable” Mode
	(Mode[0])	DSM Multiplication Mode 0B: Density Multiplier for Single Reference 1B: Density Multiplier for Bipolar Reference
2:0	**Function[2:0]**	000b Timer (chainable) 001b Counter (chainable) 010b CRCPRS (chainable) 011b IPWM 100b Dead Band 101b UART (DCCxx blocks only) 110b SPI (DCCxx blocks only) 111b DSM

12.3.6 DxCxxIN

Digital Basic/Communications Type B Block Input Register

Individual Register Names and Addresses:

DBC00IN : 1,21h	DBC01IN : 1,25h	DCC02IN : 1,29h	DCC03IN : 1,2Dh
DBC10IN : 1,31h	DBC11IN : 1,35h	DCC12IN : 1,39h	DCC13IN : 1,3Dh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0				RW : 0			
Bit Name	Data Input[3:0]				Clock Input[3:0]			

These registers are used to select the data and clock inputs.

Before changing any of the configuration registers (DxCxxFN, DxCxxIN, and DxCxxOU), disable the corresponding digital block by setting bit 0 in the CR0 or DxCxxCR0 register to ‘0’. The values in this register should not be changed while the block is enabled. After all configuration changes are made, enable the block by setting bit 0 in the CR0 register to ‘1’.

The naming convention for this register is as follows. The first ‘x’ in the digital register’s name represents either “B” for basic or “C” for communication. For rows of digital PSoC blocks and their registers, the second ‘x’ set represents <Prefix>mn<Suffix>, where m=row index, n=column index. Therefore, DCC12IN is a digital communication register for a digital PSoC block in row 1 column 2. Depending on the digital row characteristics of your PSoC device, some addresses may not be available. For additional information, refer to the “Register Definitions” on page 300 in the Digital Blocks chapter.

Bit	Name	Description	
7:4	**Data Input[3:0]**	0h	Low (0)
		1h	High (1)
		2h	Row broadcast net
		3h	Chain function to previous block (low (0) in block DBC00IN)
		4h	Analog column comparator 0
		5h	Analog column comparator 1
		6h	Analog column comparator 2
		7h	Analog column comparator 3
		8h	Row output 0
		9h	Row output 1
		Ah	Row output 2
		Bh	Row output 3
		Ch	Row input 0
		Dh	Row input 1
		Eh	Row input 2
		Fh	Row input 3

(continued on next page)

12.3.6 DxCxxIN *(continued)*

3:0	**Clock Input[3:0]**	0h	Clock disabled (low)
		1h	VC3
		2h	Row broadcast net
		3h	Previous block primary output (low for DBC00)
		4h	SYSCLKX2
		5h	VC1
		6h	VC2
		7h	CLK32K
		8h	Row output 0
		9h	Row output 1
		Ah	Row output 2
		Bh	Row output 3
		Ch	Row input 0
		Dh	Row input 1
		Eh	Row input 2
		Fh	Row input 3

12.3.7 DxCxxOU

Digital Basic/Communications Type B Block Output Register

Individual Register Names and Addresses:

DBC00OU : 1,22h	DBC01OU : 1,26h	DCC02OU : 1,2Ah	DCC03OU : 1,2Eh
DBC10OU : 1,32h	DBC11OU : 1,36h	DCC12OU : 1,3Ah	DCC13OU : 1,3Eh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0		RW : 0	RW : 0		RW : 0	RW : 0	
Bit Name	AUXCLK		AUXEN	AUX I/O Select[1:0]		OUTEN	Output Select[1:0]	

This register is used to control the connection of digital block outputs to the available row interconnect and control clock resynchronization.

Before changing any of the configuration registers (DxCxxFN, DxCxxIN, and DxCxxOU), disable the corresponding digital block by setting bit 0 in the CR0 or DxCxxCR0 register to '0'. The values in this register should not be changed while the block is enabled. After all configuration changes are made, enable the block by setting bit 0 in the DxCxxCR0 register to '1'.

The naming convention for this register is as follows. The first 'x' in the digital register's name represents either "B" for basic or "C" for communication. For rows of digital PSoC blocks and their registers, the second 'x' set represents <Prefix>mn<Suffix>, where m=row index, n=column index. Therefore, DBC12OU is a digital basic register for a digital PSoC block in row 1 column 2. Depending on the digital row characteristics of your PSoC device, some addresses may not be available. For additional information, refer to the "Register Definitions" on page 300 in the Digital Blocks chapter.

Bit	Name	Description
7:6	**AUXCLK**	00b No sync 16-to-1 clock mux output 01b Synchronize Output of 16-to-1 clock mux to SYSCLK 10b Synchronize Output of 16-to-1 clock mux to SYSCLKX2 11b SYSCLK Directly connect SYSCLK to block clock input
5	**AUXEN**	Auxiliary I/O Enable (function dependent) All Functions except SPI Slave: Enable Auxiliary Output Driver 0 Disabled 1 Enabled SPI Slave: Input Source for SS_ 0 Row Input [3:0], as selected by the AUX I/O Select bits 1 Force SS_ Active
4:3	**AUX I/O Select[1:0]**	Auxiliary I/O Select Function Output (function dependent) All Functions except SPI Slave: Row Output Select 00b Row Output 0 01b Row Output 1 10b Row Output 2 11b Row Output 3 SPI Slave Source for SS_ Input if AUXEN =0. 00b Row Input 0 01b Row Input 1 10b Row Input 2 11b Row Input 3

(continued on next page)

12.3.7 DxCxxOU *(continued)*

4:3 *(cont.)*	**AUX I/O Select[1:0]**	SPI Slave Source for SS_ Input if AUXEN =1. 00b Force SS_ Active 01b Reserved 10b Reserved 11b Reserved
2	**OUTEN**	Enable Primary Function Output Driver 0 Disabled 1 Enabled
1:0	**Output Select[1:0]**	Row Output Select for Primary Function Output 00b Row Output 0 01b Row Output 1 10b Row Output 2 11b Row Output 3

12.3.8 DxCxxCR1 (Timer Control:000)

Individual Register Names and Addresses:

DBC00CR1 : 1,23h	DBC01CR1 : 1,27h	DCC02CR1 : 1,2Bh	DCC03CR1 : 1,2Fh
DBC10CR1 : 1,33h	DBC11CR1 : 1,37h	DCC12CR1 : 1,3Bh	DCC13CR1 : 1,3Fh

	7	6	5	4	3	2	1	0
Access : POR	RW:0000				RW:0	RW:00		RW:0
Bit Name	Multi-Shot				KILL_INV	KILL_MD[1:0]		KILL_INT

For additional information, refer to the “Register Definitions” on page 300 in the Digital Blocks chapter.

Bit	Name	Description
7:4	**Multi-Shot == 0** **Multi-Shot > 0**	No Multi-shot function, and function will run repeatedly. Multi-shot number.
3	**KILL_INV**	Invert KILL signal.
2:1	**KILL_MD[1:0]**	KILL mode. 0XB NO KILL mode. 10B “KILL Sync” mode: the function block is always in reload when block clock is coming and KILL is high, and the FO1 and FO2 are gated to low. The function block returns to normal operation when KILL goes away. 11B “KILL Disable” mode: the function block is disabled when KILL is asserted.
0	**KILL_INT**	Set ‘1’ to select KILL as interrupt; it has highest priority.

12.3.9 DxCxxCR1 (Counter Control:001)

Individual Register Names and Addresses:

DBC00CR1 : 1,23h
DBC01CR1 : 1,27h
DCC02CR1 : 1,2Bh
DCC03CR1 : 1,2Fh
DBC10CR1 : 1,33h
DBC11CR1 : 1,37h
DCC12CR1 : 1,3Bh
DCC13CR1 : 1,3Fh

	7	6	5	4	3	2	1	0
Access : POR	RW:0000				RW:0	RW:00		RW:0
Bit Name	Multi-Shot				KILL_INV	KILL_MD[1:0]		KILL_INT

For additional information, refer to the “Register Definitions” on page 300 in the Digital Blocks chapter.

Bit	Name	Description
7:4	**Multi-Shot**	Has same meaning as in Timer.
3	**KILL_INV**	Same as Timer.
2:1	**KILL_MD[1:0]**	Same as Timer.
0	**KILL_INT**	Same as Timer.

12.3.10 DxCxxCR1 (CRCPRS Control:010)

Individual Register Names and Addresses:

DBC00CR1 : 1,23h
DBC01CR1 : 1,27h
DCC02CR1 : 1,2Bh
DCC03CR1 : 1,2Fh
DBC10CR1 : 1,33h
DBC11CR1 : 1,37h
DCC12CR1 : 1,3Bh
DCC13CR1 : 1,3Fh

	7	6	5	4	3	2	1	0
Access : POR	RW:0000				RW:0	RW:00		RW:0
Bit Name	Multi-Shot				KILL_INV	KILL_MD[1:0]		KILL_INT

For additional information, refer to the “Register Definitions” on page 300 in the Digital Blocks chapter.

Bit	Name	Description
7:4	**Multi-Shot**	Has same meaning as in Timer.
3	**KILL_INV**	Same as Timer.
2:1	**KILL_MD[1:0]**	Same as Timer.
0	**KILL_INT**	Same as Timer.

12.3.11 DxCxxCR1 (IPWM Control:011)

Individual Register Names and Addresses:

DBC00CR1 : 1,23h	DBC01CR1 : 1,27h	DCC02CR1 : 1,2Bh	DCC03CR1 : 1,2Fh
DBC10CR1 : 1,33h	DBC11CR1 : 1,37h	DCC12CR1 : 1,3Bh	DCC13CR1 : 1,3Fh

	7	6	5	4	3	2	1	0
Access : POR	RW:0000			RW:0	RW:000			
Bit Name	Multi-Shot				STARTINV	DBW[2:0]		

For additional information, refer to the “Register Definitions” on page 300 in the Digital Blocks chapter.

Bit	Name	Description
7:4	**Multi-Shot**	Has same meaning as in Timer. 0 means the function is not in PPG mode. Otherwise, the function is in PPG mode and the iteration time by one trigger is specified by these 4 multi-shot bits.
3	**STARTINV**	‘1’ to invert START signal
2:0	**DBW[2:0]**	Dead Band Width: 000: no DeadBand 001: 1 BLKCLK DeadBand 010: 2 BLKCLK DeadBand 011: 4 BLKCLK DeadBand 100: 8 BLKCLK DeadBand 101: 16 BLKCLK DeadBand 110: 32 BLKCLK DeadBand 111: 64 BLKCLK DeadBand

12.3.12 DxCxxCR1 (Dead Band Control:100)

Individual Register Names and Addresses:

DBC00CR1 : 1,23h	DBC01CR1 : 1,27h	DCC02CR1 : 1,2Bh	DCC03CR1 : 1,2Fh
DBC10CR1 : 1,33h	DBC11CR1 : 1,37h	DCC12CR1 : 1,3Bh	DCC13CR1 : 1,3Fh

	7	6	5	4	3	2	1	0
Access : POR								RW:0
Bit Name								KILL_INT

For additional information, refer to the “Register Definitions” on page 300 in the Digital Blocks chapter.

Bit	Name	Description
0	**KILL_INT**	Set ‘1’ to select KILL as interrupt; it has highest priority.

12.3.13 DxCxxCR1 (SPIM Control:0-110)

Individual Register Names and Addresses:

DBC00CR1 : 1,23h	DBC01CR1 : 1,27h	DCC02CR1 : 1,2Bh	DCC03CR1 : 1,2Fh
DBC10CR1 : 1,33h	DBC11CR1 : 1,37h	DCC12CR1 : 1,3Bh	DCC13CR1 : 1,3Fh

	7	6	5	4	3	2	1	0
Access : POR	RW:0	RW:0		RW:00000				
Bit Name	Chain	LSB		SPI Length				

For additional information, refer to the “Register Definitions” on page 300 in the Digital Blocks chapter.

Bit	Name	Description
7	**Chain**	1 means the block is in chain in SPI.
6	**LSB**	Indicates the LSB block in chain mode.
4:0	**SPI Length**	Specifies the SPI length in chain mode.

12.3.14 DxCxxCR1 (SPIS Control:0-110)

Individual Register Names and Addresses:

DBC00CR1 : 1,23h	DBC01CR1 : 1,27h	DCC02CR1 : 1,2Bh	DCC03CR1 : 1,2Fh
DBC10CR1 : 1,33h	DBC11CR1 : 1,37h	DCC12CR1 : 1,3Bh	DCC13CR1 : 1,3Fh

	7	6	5	4	3	2	1	0
Access : POR	RW:0	RW:0		RW:00000				
Bit Name	Chain	LSB		SPI Length				

For additional information, refer to the “Register Definitions” on page 300 in the Digital Blocks chapter.

Bit	Name	Description
7	**Chain**	‘1’ means the block is in chain in SPI.
6	**LSB**	Indicates the LSB block in chain mode.
4:0	**SPI Length**	Specifies the SPI length in chain mode.

12.3.15 DxCxxCR1 (DSM Control:111)

Individual Register Names and Addresses:

DBC00CR1 : 1,23h
DBC10CR1 : 1,33h
DBC01CR1 : 1,27h
DBC11CR1 : 1,37h
DCC02CR1 : 1,2Bh
DCC12CR1 : 1,3Bh
DCC03CR1 : 1,2Fh
DCC13CR1 : 1,3Fh

	7	6	5	4	3	2	1	0
Access : POR					RW:0			RW:0
Bit Name					KILL_INV			KILL_INT

For additional information, refer to the “Register Definitions” on page 300 in the Digital Blocks chapter.

Bit	Name	Description
3	**KILL_INV**	‘1’ to invert KILL signal.
0	**KILL_INT**	‘1’ to select KILL as interrupt; ‘0’ to select CO as interrupt.

12.3.16 CMPxCR1

Individual Register Names and Addresses:

CMP0CR1 : 1,50h CMP1CR1 : 1,54h

	7	6	5	4	3	2	1	0
Access : POR	RW:0	RW:0	RW:0			RW:0		
Bit Name	BINC	CBUS	NMux[2:0]			PMux[2:0]		

This register is one of two registers used to configure the type E continuous time PSoC block in compare columns.

This register is only used by the CY8C22x45 and CY8C21345 PSoC devices.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 366 in the Two Column Analog Compare System chapter.

Bits	Name	Description
6	**CompBus**	Enable output to the comparator bus. The comparator bus is always driven from the CT block. 0 Disable output to comparator bus. 1 Enable output to comparator bus.
5:3	**NMux[2:0]**	Encoding for negative input select. Note that available mux inputs vary by individual PSoC block. **CMP0** **CMP1** 000b CMP1 CMP0 001b VBG VBG 010b Short VDACL Short VDACR Port inputs Port Inputs 011b Mux Bus Left Mux Bus Right 100b FB# FB# 101b VDACL VDACR 110b VDACR VDACL 111b Port Inputs Port Inputs # Feedback. Gain = 1, configuration only.
2:0	**PMux[2:0]**	Encoding for positive input select. Note that available mux inputs vary by individual PSoC block. **CMP0** **CMP1** 000b P2[7] P2[6] 001b Port Inputs Port Inputs 010b CMP1 CMP0 011b VBG VBG 100b VDACL VDACR 101b VDACR VDACL 110b Short VDACL Short VDACR Port inputs Port Inputs 111b Mux Bus Left Mux Bus Right

12.3.17 CMPxCR2

Individual Register Names and Addresses:

CMP0CR2 : 1,51h CMP1CR2 : 1,55h

	7	6	5	4	3	2	1	0
Access : POR							RW:0	RW:0
Bit Name							FullRange	PWR

This register is one of two registers used to configure the type E continuous time PSoC block in compare columns.

This register is only used by the CY8C22x45 and CY8C21345 PSoC devices.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 366 in the Two Column Analog Compare System chapter.

Bits	Name		Description
1	**FullRange**	0	Input range includes Vss but not Vdd.
		1	Rail-to-rail input range, with approximately 10 μA additional cell current.
0	**PWR**	0	Powers off both the CT and SC blocks in the column.
		1	Enables the column's analog blocks.

12.3.18 VDAC5xCR0

Individual Register Names and Addresses:

VDAC50CR0 : 1,53h VDAC51CR0 : 1,57h

	7	6	5	4	3	2	1	0
Access : POR				RW : 0				
Bit Name				VDACIN[4:0]				

These two registers are used to configure the VDAC5 output voltage.

This register is only used by the CY8C22x45 and CY8C21345 PSoC devices.

For additional information, refer to the “Register Definitions” on page 366 in the Two Column Analog Compare System chapter.

Bits	Name	Description
4:0	**VDACIN[4:0]**	5-bit R2R VDAC selector signal

12.3.19 CSCMPCR0

Individual Register Names and Addresses:

CSCMPCR0: 1,58h

	7	6	5 4	3 2	1 0
Access : POR	RW : 0	RW : 0	R : 0	RW : 0	RW : 0
Bit Name	CSSEL1[2]	CSSEL0[2]	CSCMP[1:0]	CSIGEN[1:0]	CSCLKDIS[1:0]

This register controls the selection of signals driven to GOO and also for gating function.

This register is only used by the CY8C22x45 and CY8C21345 PSoC devices.

For additional information, refer to the “Register Definitions” on page 366 in the Two Column Analog Compare System chapter.

Bits	Name	Description
7/6	**CSSEL1[2]/CSSEL0[2]**	The MSB of CSCEL1/CSCEL0. See CSCMPGOEN.
5:4	**CSCMP[1:0]**	Read only; can read out the output of the CS comparator.
3:2	**CSIGEN[1:0]**	Reserved for gated function.
1:0	**CSCLKDIS[1:0]**	CS Comparator bus sync enable/disable.

12.3.20 CSCMPGOEN

Individual Register Names and Addresses:

CSCMPGOEN: 1,59h

	7	6	5	4	3	2	1	0
Access : POR	RW:0	RW:0	RW : 0		RW:0	RW:0	RW : 0	
Bit Name	GOO7	GOO3	CSSEL1[1:0]		GOO6	GOO2	CSSEL0[1:0]	

This register controls options for driving the analog comparator bus and clock to the global bus.

This register is only used by the CY8C22x45 and CY8C21345 PSoC devices.

For additional information, refer to the “Register Definitions” on page 366 in the Two Column Analog Compare System chapter.

Bits	Name	Description
7	**GOO7**	Drives the selected CS comparator column 1 signal to GOO7
6	**GOO3**	Drives the selected CS comparator column 1 signal to GOO3
5:4	**CSSEL1[1:0]**	Selects the CS comparator column 1 signal to global bus. MSB is in CSCMPCR0 (1,58) 000 comparator bus output 001 unsynced Column Clock 010 comparator output after signal sync 011 unsynced Column Clock gated by comparator bus output 100 RS-latch slave output 101 CMPFF2 110 positive one-shot 111 positive or negative one-shot
3	**GOO6**	Drives the selected CS comparator column 0 signal to GOO6
2	**GOO2**	Drives the selected CS comparator column 0 signal to GOO2
1:0	**CSSEL0[1:0]**	Selects the CS comparator column 0 signal to global bus. MSB is in CSCMPCR0 (1,58) 000 comparator bus output 001 unsynced Column Clock 010 comparator output after signal sync 011 unsynced Column Clock gated by comparator bus output 100 RS-latch master output 101 CMPFF2 110 positive one-shot 111 positive or negative one-shot

12.3.21 CSLUTCR0

Individual Register Names and Addresses:

CSLUTCR0: 1,5Ah

	7	6	5	4	3	2	1	0
Access : POR	RW : 0				RW : 0			
Bit Name	CSLUT1[3:0]				CSLUT0[3:0]			

This register is used to select the logic function.

This register is only used by the CY8C22x45 and CY8C21345 PSoC devices.

For additional information, refer to the “Register Definitions” on page 366 in the Two Column Analog Compare System chapter.

Bits	Name	Description
7:4	**CSLUT1[3:0]**	Select 1 of 16 logic functions for output of CS comparator bus 1.
3:0	**CSLUT0[3:0]**	Select 1 of 16 logic functions for output of CS comparator bus 0.

12.3.22 CMPCOLMUX

Individual Register Names and Addresses:

CMPCOLMUX: 1,5Bh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00		RW:0	RW:0	RW : 0		RW : 0	
Bit Name	ACMPS1[1:0]		LMXR	RMXL	ACPI1[1:0]		ACPI0[1:0]	

This register controls the analog muxes that feed signals in from port pins into the analog compare column.

This register is only used by the CY8C22x45 and CY8C21345 PSoC devices.

For additional information, refer to the “Register Definitions” on page 366 in the Two Column Analog Compare System chapter.

Bits	Name	Description
7:6	**ACMPS1[1:0]**	The last stage selection for CMPCOL1 compare bus output 00B CMPSYNC1 (LUT out or RS-Lat slave output) 01B CMPSYNC1 pass double sync 10B positive one-shot from CMPSYNC1 11B positive or negative one-shot from CMPSYNC1
5	**LMXR**	Select CS CMP column 0 input to column I/O input mux output. 0:P0[7:1:2], 1:P0[6:0:2]
4	**RMXL**	Select CS CMP column 1 input to column I/O input mux output. 1:P0[7:1:2], 0:P0[6:0:2]
3:2	**ACPI1[1:0]**	Select the CS comparator analog mux 1. PO[0:6:2]
1:0	**ACPI0[1:0]**	Select the CS comparator analog mux 0. PO[1:7:2]

12.3.23 CMPPWMCR

Individual Register Names and Addresses:

CMPPWMCR: 1,5Ch

	7	6	5	4	3	2	1	0
Access : POR	RW : 0		RW:0	RW:00		RW:0	RW:00	
Bit Name	ACMPS0[1:0]		PWMDLY_EN[1]	PWMCLK_SEL1[1:0]		PWMDLY_EN[0]	PWMCLK_SEL0[1:0]	

This register is used to control the PWM function.

This register is only used by the CY8C22x45 and CY8C21345 PSoC devices.

For additional information, refer to the “Register Definitions” on page 366 in the Two Column Analog Compare System chapter.

Bits	Name	Description
7:6	**ACMPS0[1:0]**	The last stage selection for CMPCOL0 compare bus output 00B CMPSYNC1 (LUT out or RS-Lat master out) 01B CMPSYNC1 pass double sync 10B positive one-shot from CMPSYNC1 11B positive or negative one-shot from CMPSYNC1
5/2	**PWMDLY_EN[1:0]**	Enable the PWM Delay function
4:3/1:0	**PWMCLK_SEL[1:0]/PWMCLK_SEL0[1:0]:**	Select trigger signals 00B TG_L or TG_H 01B TG_L 10B TG_H 11B TG_16BIT

12.3.24 CMPFLTCR

Individual Register Names and Addresses:

CMPFLTCR: 1,5Dh

	7	6	5	4	3	2	1	0
Access : POR	RW:0		RW:0	RW:0	RW:0		RW:0	RW:0
Bit Name	FLT1_WD[1:0]		FLT1_CSL	FLT1_EN	FLT0_WD[1:0>		FLT0_CSL	FLT0_EN

This register controls the pulse filter feature for analog column 2 and column 3.

This register is only used by the CY8C22x45 and CY8C21345 PSoC devices.

For additional information, refer to the “Register Definitions” on page 366 in the Two Column Analog Compare System chapter.

Bits	Name	Description
7:6/3:2	**FLT1_WD[1:0]/FLT0_WD[1:0]:**	Filter width 00B 1 selected clock 01B 2 selected clock 10B 4 selected clock 11B 8 selected clock
5/1	**FLT1_CSL/FLT0_CSL**	Filter Clock Select 0 Sysclk 1 Column Clock
4/0	**FLT1_EN/FLT0_EN**	1 enable filter function

12.3.25 CMPCLK1

Individual Register Names and Addresses:

CMPCLK1: 1,5Eh

	7	6	5	4	3	2	1	0
Access : POR	RW:0	RW:0	RW:0		RW:0	RW:0	RW:0	
Bit Name	RS_SEL[1]	SYS1	DIVCLK1		RS_SEL[0]	SYS0	DIVCLK0	

This register controls the analog column clock selection and RS feature with CMPCLK0.

This register is only used by the CY8C22x45 and CY8C21345 PSoC devices.

For additional information, refer to the “Register Definitions” on page 366 in the Two Column Analog Compare System chapter.

Bits	Name	Description	
7/3	**RS_SEL[1:0]**	Set 1 to select RS output to compare bus for Column1/Column0.	
6/2	**SYS1/SYS0:**		
		0	Column 1/0 clock selection is controlled by CMPCLK0.
		1	Column 1/0 clock selection is controlled by SYSCLK.
5:4/1:0	**DIVCLK1[1:0]/DIVCLK0[1:0]:**		
		00b	No divide on selected column 0 clock.
		01b	Divide by 2 on selected column 0 clock.
		10b	Divide by 4 on selected column 0 clock.
		11b	Divide by 8 on selected column 0 clock.

12.3.26 CMPCLK0

Individual Register Names and Addresses:

CMPCLK0: 1,5Fh

	7	6	5	4	3	2	1	0
Access : POR	RW:0		RW:00		RW:00		RW:00	
Bit Name	RS_EN		SEL_CSD[1:0]		CMPCOL1[1:0]		CMPCOL0[1:0]	

This register controls the analog column clock selection and RS feature with CMPCLK1.

This register is only used by the CY8C22x45 and CY8C21345 PSoC devices.

For additional information, refer to the “Register Definitions” on page 366 in the Two Column Analog Compare System chapter.

Bits	Name	Description
7	**RS_EN**	1 Enable RS function. CMPCOL0's LUT output is master and CMPCOL1's LUT output is slave.
3:2/1:0	**CMPCOL1[1:0]/CMPCOL0[1:0]:**	Clock Selection for CMPCOL1/0 If related SEL_CSD bit is 0, then: 00b VC1 01b VC2 10b Analog Clock 0 (decided by 1,61, CLK_CR1) 11b Analog Clock 1 (decided by 1,61, CLK_CR1) If related SEL_CSD bit is 1, then corresponding clock comes from related CSD_CNTCK.

12.3.27 CLK_CR0

Analog Column Clock Control Register 0

Individual Register Names and Addresses:

CLK_CR0: 1,60h

	7	6	5	4	3	2	1	0
Access : POR					RW : 0		RW : 0	
Bit Name					AColumn1[1:0]		AColumn0[1:0]	

This register is used to select the clock source for an individual analog column.

Each column has two bits that select the column clock input source. The clock dividing is controlled by the CLK_CR3 register. Note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'.

Bits	Name	Description
3:2	**AColumn1[1:0]**	Clock selection for column 1. 00b Variable Clock 1 (VC1) 01b Variable Clock 2 (VC2) 10b Analog Clock 0 (ACLK0) 11b Analog Clock 1 (ACLK1)
1:0	**AColumn0[1:0]**	Clock selection for column 0. 00b Variable Clock 1 (VC1) 01b Variable Clock 2 (VC2) 10b Analog Clock 0 (ACLK0) 11b Analog Clock 1 (ACLK1)

12.3.28 CLK_CR1

Analog Clock Source Control Register 1

Individual Register Names and Addresses:

CLK_CR1: 1,61h

	7	6	5	4	3	2	1	0
Access : POR			RW : 0			RW : 0		
Bit Name			ACLK1[2:0]			ACLK0[2:0]		

This register is used to select the clock source for an individual analog column.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'.

Bits	Name	Description
5:3	**ACLK1[2:0]**	Select the clocking source for Analog Clock 1. 000b Digital Basic Block 00 001b Digital Basic Block 01 010b Digital Communication Block 02 011b Digital Communication Block 03 100b Digital Basic Block 10, Reserved for CY8C21345 101b Digital Basic Block 11, Reserved for CY8C21345 110b Digital Communication Block 12, Reserved for CY8C21345 111b Digital Communication Block 13, Reserved for CY8C21345
2:0	**ACLK0[2:0]**	Select the clocking source for Analog Clock 0. 000b Digital Basic Block 00 001b Digital Basic Block 01 010b Digital Communication Block 02 011b Digital Communication Block 03 100b Digital Basic Block 10, Reserved for CY8C21345 101b Digital Basic Block 11, Reserved for CY8C21345 110b Digital Communication Block 12, Reserved for CY8C21345 111b Digital Communication Block 13, Reserved for CY8C21345

12.3.29 ABF_CR0

Analog Output Buffer Control Register 0

Individual Register Names and Addresses:

ABF_CR0: 1,62h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0							
Bit Name	ACol1Mux							

This register controls analog input muxes from Port 0.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of ‘0’.

Bits	Name	Description	
7	**ACol1Mux**	0	Set column 1 input to column 1 input mux output. (1 Column: selects among P0[6,4,2,0])
		1	Set column 1 input to column 0 input mux output. (1 Column: selects among P0[7,5,3,1])

12.3.30 AMD_CR0

Analog Modulation Control Register 0

Individual Register Names and Addresses:

AMD_CR0: 1,63h

	7	6	5	4	3	2	1	0
Access : POR					RW : 0			
Bit Name					AMOD0[3:0]			

This register is used to select the modulator bits used with each column.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'.

Bits	Name	Description
3:0	**AMOD0[3:0]**	These bits are specific to the CY8C22x45 and CY8C21345 PSoC devices. These devices have two column limited and two compare column analog devices. 0000b Zero (off) 0001b Global Output Bus, even bus bit 1 (GOE[1]) 0010b Global Output Bus, even bus bit 0 (GOE[0]) 0011b Row 0 Broadcast Bus 0100b Analog Column Comparator 0 0101b Analog Column Comparator 1 0110b CS Compare Column Comparator 0 0111b CS Compare Column Comparator 1 1000b PWMVref 0 1001b PWMVref 1 1010b Reserved 1011b Reserved 1100b Analog Column Comparator 0, single synchronized 1101b Analog Column Comparator 1, single synchronized 1110b CS Compare Column Comparator 0, single synchronized 1111b CS Compare Column Comparator 1, single synchronized

12.3.31 CMP_GO_EN

Comparator Bus to Global Outputs Enable Register

Individual Register Names and Addresses:

CMP_GO_EN: 1,64h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0		RW : 0	RW : 0	RW : 0	
Bit Name	GOO5	GOO1	SEL1[1:0]		GOO4	GOO0	SEL0[1:0]	

This register controls options for driving the analog comparator bus and column clock to the global bus.

For additional information, refer to the "Register Definitions" on page 349 in the Two Column Limited Analog System chapter.

Bits	Name	Description
7	**GOO5**	Drives the selected column 1 signal to GOO5.
6	**GOO1**	Drives the selected column 1 signal to GOO1.
5:4	**SEL1[1:0]**	Selects the column 1 signal to output. 00b Comparator bus output 01b Column clock 10b Comparator output after single sync 11b Column clock gated with the synchronized comparator bus
3	**GOO4**	Drives the selected column 0 signal to GOO4.
2	**GOO0**	Drives the selected column 0 signal to GOO0.
1:0	**SEL0[1:0]**	Selects the column 0 signal to output. 00b Comparator bus output 01b Column clock 10b Comparator output after single sync 11b Column clock gated with the synchronized comparator bus

12.3.32 AMD_CR1

Analog Modulation Control Register 1

Individual Register Names and Addresses:

AMD_CR1: 1,66h

	7	6	5	4	3	2	1	0
Access : POR					RW : 0			
Bit Name					AMOD1[3:0]			

This register is used to select the modulator bits used with each column.

Note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'.

Bits	Name	Description
3:0	**AMOD1[3:0]**	These bits are specific to the CY8C22x45 and CY8C21345 PSoC devices. These devices are two column limited analog and two compare column devices. 0000b Zero (off) 0001b Global Output Bus, even bus bit 1 (GOE[1]) 0010b Global Output Bus, even bus bit 0 (GOE[0]) 0011b Row 0 Broadcast Bus 0100b Analog Column Comparator 0 0101b Analog Column Comparator 1 0110b CS Compare Column Comparator 0 0111b CS Compare Column Comparator 1 1000b PWMVref 0 1001b PWMVref 1 1010b Reserved 1011b Reserved 1100b Analog Column Comparator 0, single synchronized 1101b Analog Column Comparator 1, single synchronized 1110b CS Compare Column Comparator 0, single synchronized 1111b CS Compare Column Comparator 1, single synchronized

12.3.33 ALT_CR0

Analog LUT Control Register 0

Individual Register Names and Addresses:

ALT_CR0: 1,67h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0				RW : 0			
Bit Name	LUT1[3:0]				LUT0[3:0]			

This register is used to select the logic function.

Note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'.

Bits	Name	Description
7:4	**LUT1[3:0]**	Select 1 of 16 logic functions for output of comparator bus 1. LUT input B=0.

	Function
0h	FALSE
1h	A AND $\overline{B}$
2h	A AND $\overline{B}$
3h	$\overline{A}$
4h	$\overline{A}$ AND B
5h	B
6h	A XOR B
7h	A OR B
8h	A NOR B
9h	$\overline{A}$ XNOR B
Ah	$\overline{B}$
Bh	$\overline{A}$ OR $\overline{B}$
Ch	$\overline{A}$
Dh	$\overline{A}$ OR B
Eh	A NAND B
Fh	TRUE

(continued on next page)

12.3.33 ALT_CR0 *(continued)*

3:0 LUT0[3:0] Select 1 of 16 logic functions for output of comparator bus 0.

	Function
0h	FALSE
1h	A AND $\overline{B}$
2h	A AND $\overline{B}$
3h	$\overline{A}$
4h	$\overline{A}$ AND B
5h	B
6h	A XOR B
7h	A OR B
8h	A NOR B
9h	$\overline{A}$ XNOR B
Ah	$\overline{B}$
Bh	$\overline{A}$ OR $\overline{B}$
Ch	$\overline{A}$
Dh	$\overline{A}$ OR B
Eh	A NAND B
Fh	TRUE

12.3.34 AMUX_CFG1

Analog Mux Config Register 1

Individual Register Names and Addresses:

AMUX_CFG1: 1,6Ah

	7	6	5	4	3	2	1	0
Access : POR			RW : 1	RW : 1	RW : 00		RW : 00	
Bit Name			GOINVR	GOINVL	AMUXMODR		AMUXMODL	

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 426 in the I/O Analog Multiplexer chapter.

Bits	Name	Description
5/4	**GOINVR/GOINVL**	1 AMUXCLK is inverted when it goes to GOO. **Note** The GOO output enable control bit is in IDAC_CRx.
3:2/1:0	**AMUXMODR/AMUXMODL**	Used to select AMUX clock source for left side and right side. 00b Select original AMUX clock decided by 0,61. 01b Select CSD_PO<0> as AMUX clock. 10b Select CSD_PO<1> as AMUX clock. 11b Precharge clock is off, no switching.

12.3.35 CLK_CR3

Analog Clock Source Control Register 3

Individual Register Names and Addresses:

CLK_CR3: 1,6Bh

	7	6	5	4	3	2	1	0
Access : POR		RW : 0	RW : 0			RW : 0	RW : 0	
Bit Name		SYS1	DIVCLK1[1:0]			SYS0	DIVCLK0[1:0]	

This register controls additional options for analog column clock generation.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 349 in the Two Column Limited Analog System chapter.

Bits	Name	Description	
6	**SYS1**	0	Column 1 clock selection is controlled by CLK_CR0.
		1	Column 1 clock selection is SYSCLK direct.
5:4	**DIVCLK1[1:0]**	00b	No divide on selected column 1 clock.
		01b	Divide by 2 on selected column 1 clock.
		10b	Divide by 4 on selected column 1 clock.
		11b	Divide by 8 on selected column 1 clock.
2	**SYS0**	0	Column 0 clock selection is controlled by CLK_CR0.
		1	Column 0 clock selection is SYSCLK direct.
1:0	**DIVCLK0[1:0]**	00b	No divide on selected column 0 clock.
		01b	Divide by 2 on selected column 0 clock.
		10b	Divide by 4 on selected column 0 clock.
		11b	Divide by 8 on selected column 0 clock.

12.3.36 GDI_O_IN_CR

Individual Register Names and Addresses:

GDI_O_IN_CR: 1,A0h

	7	6	5	4	3	2	1	0
Access : POR	RW: 00							
Bit Name	GDIOICR[7]	GDIOICR[6]	GDIOICR[5]	GDIOICR[4]	GDIOICR[3]	GDIOICR[2]	GDIOICR[1]	GDIOICR[0]

For additional information, refer to the “Register Definitions” on page 275 in the Global Digital Interconnect chapter.

Bit	Name	Description
7	**GDIOICR[7]**	0: GIO[7] --> GOO[7] or 1: GIO[6] --> GOO[7] when bit 7 is 1 in GDI_O_IN register.
6	**GDIOICR[6]**	0: GIO[6] --> GOO[6] or 1: GIO[5] --> GOO[6] when bit 6 is 1 in GDI_O_IN register.
5	**GDIOICR[5]**	0: GIO[5] --> GOO[5] or 1: GIO[4] --> GOO[5] when bit 5 is 1 in GDI_O_IN register.
4	**GDIOICR[4]**	0: GIO[4] --> GOO[4] or 1: GIO[3] --> GOO[4] when bit 4 is 1 in GDI_O_IN register.
3	**GDIOICR[3]**	0: GIO[3] --> GOO[3] or 1: GIO[2] --> GOO[3] when bit 3 is 1 in GDI_O_IN register.
2	**GDIOICR[2]**	0: GIO[2] --> GOO[2] or 1: GIO[1] --> GOO[2] when bit 2 is 1 in GDI_O_IN register.
1	**GDIOICR[1]**	0: GIO[1] --> GOO[1] or 1: GIO[0] --> GOO[1] when bit 1 is 1 in GDI_O_IN register.
0	**GDIOICR[0]**	0: GIO[0] --> GOO[0] or 1: GIO[7] --> GOO[0] when bit 0 is 1 in GDI_O_IN register.

12.3.37 GDI_E_IN_CR

Individual Register Names and Addresses:

GDI_E_IN_CR: 1,A1h

	7	6	5	4	3	2	1	0
Access : POR	RW: 00							
Bit Name	GDIEICR[7]	GDIEICR[6]	GDIEICR[5]	GDIEICR[4]	GDIEICR[3]	GDIEICR[2]	GDIEICR[1]	GDIEICR[0]

For additional information, refer to the “Register Definitions” on page 275 in the Global Digital Interconnect chapter.

Bit	Name	Description
7	**GDIEICR[7]**	0: GIE[7] --> GOE[7] or 1: GIE[6] --> GOE[7] when bit 7 is 1 in GDI_E_IN register.
6	**GDIEICR[6]**	0: GIE[6] --> GOE[6] or 1: GIE[5] --> GOE[6] when bit 6 is 1 in GDI_E_IN register.
5	**GDIEICR[5]**	0: GIE[5] --> GOE[5] or 1: GIE[4] --> GOE[5] when bit 5 is 1 in GDI_E_IN register.
4	**GDIEICR[4]**	0: GIE[4] --> GOE[4] or 1: GIE[3] --> GOE[4] when bit 4 is 1 in GDI_E_IN register.
3	**GDIEICR[3]**	0: GIE[3] --> GOE[3] or 1: GIE[2] --> GOE[3] when bit 3 is 1 in GDI_E_IN register.
2	**GDIEICR[2]**	0: GIE[2] --> GOE[2] or 1: GIE[1] --> GOE[2] when bit 2 is 1 in GDI_E_IN register.
1	**GDIEICR[1]**	0: GIE[1] --> GOE[1] or 1: GIE[0] --> GOE[1] when bit 1 is 1 in GDI_E_IN register.
0	**GDIEICR[0]**	0: GIE[0] --> GOE[0] or 1: GIE[7] --> GOE[0] when bit 0 is 1 in GDI_E_IN register.

12.3.38 GDI_O_OU_CR

Individual Register Names and Addresses:

GDI_O_OU_CR: 1,A2h

	7	6	5	4	3	2	1	0
Access : POR	RW: 00							
Bit Name	GDIOOCR[7]	GDIOOCR[6]	GDIOOCR[5]	GDIOOCR[4]	GDIOOCR[3]	GDIOOCR[2]	GDIOOCR[1]	GDIOOCR[0]

For additional information, refer to the “Register Definitions” on page 275 in the Global Digital Interconnect chapter.

Bit	Name	Description
7	**GDOOICR[7]**	0: GOO[7] --> GIO[7] or 1: GOO[6] --> GIO[7] when bit 7 is 1 in GDI_O_OU register.
6	**GDOOICR[6]**	0: GOO[6] --> GIO[6] or 1: GOO[5] --> GIO[6] when bit 6 is 1 in GDI_O_OU register.
5	**GDOOICR[5]**	0: GOO[5] --> GIO[5] or 1: GOO[4] --> GIO[5] when bit 5 is 1 in GDI_O_OU register.
4	**GDOOICR[4]**	0: GOO[4] --> GIO[4] or 1: GOO[3] --> GIO[4] when bit 4 is 1 in GDI_O_OU register.
3	**GDOOICR[3]**	0: GOO[3] --> GIO[3] or 1: GOO[2] --> GIO[3] when bit 3 is 1 in GDI_O_OU register.
2	**GDOOICR[2]**	0: GOO[2] --> GIO[2] or 1: GOO[1] --> GIO[2] when bit 2 is 1 in GDI_O_OU register.
1	**GDOOICR[1]**	0: GOO[1] --> GIO[1] or 1: GOO[0] --> GIO[1] when bit 1 is 1 in GDI_O_OU register.
0	**GDOOICR[0]**	0: GOO[0] --> GIO[0] or 1: GOO[7] --> GIO[0] when bit 0 is 1 in GDI_O_OU register.

12.3.39 GDI_E_OU_CR

Individual Register Names and Addresses:

GDI_E_OU_CR: 1,A3h

	7	6	5	4	3	2	1	0
Access : POR	RW: 00							
Bit Name	GDIEOCR[7]	GDIEOCR[6]	GDIEOCR[5]	GDIEOCR[4]	GDIEOCR[3]	GDIEOCR[2]	GDIEOCR[1]	GDIEOCR[0]

For additional information, refer to the “Register Definitions” on page 275 in the Global Digital Interconnect chapter.

Bit	Name	Description
7	**GDEOICR[7]**	0: GOE[7] --> GIE[7] or 1: GOE[6] --> GIE[7] when bit 7 is 1 in GDI_E_OU register.
6	**GDEOICR[6]**	0: GOE[6] --> GIE[6] or 1: GOE[5] --> GIE[6] when bit 6 is 1 in GDI_E_OU register.
5	**GDEOICR[5]**	0: GOE[5] --> GIE[5] or 1: GOE[4] --> GIE[5] when bit 5 is 1 in GDI_E_OU register.
4	**GDEOICR[4]**	0: GOE[4] --> GIE[4] or 1: GOE[3] --> GIE[4] when bit 4 is 1 in GDI_E_OU register.
3	**GDEOICR[3]**	0: GOE[3] --> GIE[3] or 1: GOE[2] --> GIE[3] when bit 3 is 1 in GDI_E_OU register.
2	**GDEOICR[2]**	0: GOE[2] --> GIE[2] or 1: GOE[1] --> GIE[2] when bit 2 is 1 in GDI_E_OU register.
1	**GDEOICR[1]**	0: GOE[1] --> GIE[1] or 1: GOE[0] --> GIE[1] when bit 1 is 1 in GDI_E_OU register.
0	**GDEOICR[0]**	0: GOE[0] --> GIE[0] or 1: GOE[7] --> GIE[0] when bit 0 is 1 in GDI_E_OU register.

12.3.40 RTCH

Individual Register Names and Addresses:

RTCH: 1,A4h

	7	6	5	4	3	2	1	0
Access : POR			RW:00		RW:0000			
Bit Name			HR1[1:0]		HR0[3:0]			

This register is used to read and write the current hour value in BCD format. Writing to this register will reset count32768 to all zeros. (Will be displayed as “XY”; the legal range is from 00 to 23.)

For additional information, see “Register Definitions” on page 437 in the Real Time Clock chapter.

Bit	Name	Description
5:4	**HR1[1:0]**	Hour time decal number; BCD code.
3:0	**HR0[3:0]**	Hour time units number; BCD code.

12.3.41 RTCM

Individual Register Names and Addresses:

RTCM: 1,A5h

	7	6	5	4	3	2	1	0
Access : POR		RW:00			RW:0000			
Bit Name		MIN1[2:0]			MIN0[3:0]			

This register is used to read and write the current minute value in BCD format. Writing to this register will reset count32768 to all zeros. (Will be displayed as “XY”; the legal range is from 00 to 59.)

For additional information, see “Register Definitions” on page 437 in the Real Time Clock chapter.

Bit	Name	Description
6:4	**MIN1[2:0]**	Minute time decal number; BCD code.
3:0	**MIN0[3:0]**	Minute time units number; BCD code.

12.3.42 RTCS

Individual Register Names and Addresses:

RTCS: 1,A6h

	7	6	5	4	3	2	1	0
Access : POR		RW:00			RW:0000			
Bit Name		SEC1[2:0]			SEC0[3:0]			

This register is used to read and write the current second value in BCD format. Writing to this register will reset count32768 to all zeros. (Will be displayed as “XY”; the legal range is from 00 to 59.)

For additional information, see “Register Definitions” on page 437 in the Real Time Clock chapter.

Bit	Name	Description
6:4	**SEC1[2:0]**	Second time decal number; BCD code.
3:0	**SEC0[3:0]**	Second time units number; BCD code.

12.3.43 RTCCR

Individual Register Names and Addresses:

RTCCR: 1,A7h

	7	6	5	4	3	2	1	0
Access : POR	RW:00		RW:0	RW:0	RW:00		RW:0	RW:0
Bit Name	TREG[1:0]		INT_EN	CLKSE	INT_SEL[1:0]		SYNCRD_EN	RTC_EN

For additional information, see “Register Definitions” on page 437 in the Real Time Clock chapter.

Bit	Name	Description
7:6	**TREG[1:0]**	Only TMOD accessible (for DFT purpose only). 00B in normal mode, RTC runs on 32K or VC1 clock. 01B bypass 8th to 14th registers in 1 second generator; 15th register is still there. RTC runs on VC1. 10B bypass 1st to 7th registers in 1 second generator. RTC runs on VC1. 01B bypass 1st to 14th registers in 1 second generator; 15th register is still there. RTC runs on VC1.
5	**INT_EN**	‘1’ for enable interrupt
4	**CLKSE**	‘1’ for VC1; ‘0’ for CLK32S. However, it will always use VC1 in test mode. In normal mode, if VC1 is selected, the RTC module acts as a fixed period interrupt source.
3:2	**INT_SEL[1:0]**	Interrupt Select 00B interrupt per second 01B interrupt per minute 10B interrupt per hour 11B interrupt per day
1	**SYNCRD_EN**	When this bit is set to ‘1’, RTC_M/RTC_S will read data from its data buffer. The data buffer will be updated from its real data register when RTC_H is read.
0	**RTC_EN**	‘1’ to enable RTC function

12.3.44 ADC_CR0

Individual Register Names and Addresses:

ADC_CR0: 1,A8h

	7	6	5	4	3	2	1	0
Access : POR	RW:0	RW:0000				RW:0	RW:0	RW:0
Bit Name	ADC_TST1	ADC_CHS[3:0]				READY	Start/ONGOING	ADCEN

For additional information, see “Register Definitions” on page 442 in the 10-bit SAR ADC Controller chapter.

Bit	Name	Description
7	**ADC_TST1**	The ADC data registers are write-only in TEST mode when this bit is set.
6:3	**ADC_CHS[3:0]**	Channel selection 0: P0.0 1: P0.1 2: P0.2 3: P0.3 4: P0.4 5: P0.5 6: P0.6 7: P0.7 8: 9: A: B: VBG C: AMUXL D: AMUXR E: VDAC0 F: V2bg
2	**READY**	‘1’ means there is a new data never been read.
1	**Start/ONGOING**	Reading 1 means the A-D conversion started, and is not finished yet. Writing 1 to it in SW trigger mode will trigger a new conversion.
0	**ADC_EN**	Enable ADC function.

12.3.45 ADC_CR1

Individual Register Names and Addresses:

ADC_CR1: 1,A9h

	7	6	5	4	3	2	1	0
Access : POR	RW:00		RW:00		RW:000			RW:0
Bit Name	CVTMD[1:0]		TIGSEL[1:0]		CLKSEL[2:0]			ALIGN_EN

For additional information, see “Register Definitions” on page 442 in the 10-bit SAR ADC Controller chapter.

Bit	Name	Description
7:6	**CVTMD[1:0]**	The conversion mode 00b the default mode that only the extra cycle for 6th bit conversion 01b the extra cycle for 6th bit conversion with add-on weak Vref buffer 10b the extra cycle for 7th bit conversion with add-on weak Vref buffer 11b the extra cycle for 1st bit conversion with add-on weak Vref buffer
5:4	**TIGSEL[1:0]**	Auto-trigger source selection 00b: TGL 01b: TGH 10b: TG16BIT 11b: TGINCMP
3:1	**CLKSEL[3:0]**	ADC Clock Selection 0: /2 1: /4 2: /6 3: /8 4: /12 5: /16 6: /32 7: /64
0	**ALIGN_EN**	‘1’ to enable auto-align function. The A-D-C will be driven by outside-block trigger signal. Refer to CHIPMISC 0, CD/0, CE/0, CF. **Note** When both ALIGN_EN and FREERUN are zero, the ADC is in software trigger mode; that is, if you write 1 to Start/ONGOING bit, it triggers one time A-D-C.

12.3.46 ADC_CR2

Individual Register Names and Addresses:

ADC_CR2: 1,AAh

	7	6	5	4	3	2	1	0
Access : POR	RW:0	RW:0	RW:0	RW:0	RW:0	RW:0	RW:00	
Bit Name	REFSEL	BUFEN	VDBEN	VDB_CLK	FREERUN	ADC_EXT_HALFVDD	ADC_MODE[1:0]	

For additional information, see “Register Definitions” on page 442 in the 10-bit SAR ADC Controller chapter.

Bit	Name	Description
7	**REFSEL**	‘1’ to select external Vref other than Vdd.
6	**BUFEN**	‘1’ to enable Vref buffer. Otherwise will bypass Vref buffer.
5	**VDBEN**	‘1’ to enable voltage doubler in ADC comparator.
4	**VDB_CLK**	‘1’ to select SYSLCK as VDB clock. Otherwise select SYSCLK/4 as VDB clock.
3	**FREERUN**	‘1’ means ADC in FREERUN mode if ADC is not in auto-align mode.
2	**ADC_EXT_HALFVDD**	‘1’ to select Vdd/2 from external pad. Otherwise Vdd/2 will come from internal source.
1:0	**ADC_MODE[1:0]**	00B Normal mode; 'Test_out' is Hi-Z output. 01B Normal mode; 'Test_out' is comparator output (digital signal). 10B Normal mode; 'Test_out' is internal Vref (analog signal). 11B Normal mode; 'Test_out' is internal Vdd/2 (analog signal).

12.3.47 ADC_CR3TRIM

Individual Register Names and Addresses:

ADC_CR3TRIM: 1,ABh

	7	6	5	4	3	2	1	0
Access : POR						RW:0_0000		
Bit Name						ADC_TRIM0[2:0]		

For additional information, see “Register Definitions” on page 442 in the 10-bit SAR ADC Controller chapter.

Bit	Name	Description
2:0	**ADC_TRIM0[2:0]**	Sent to ADC comparator block directly.

12.3.48 ADC_CR4

Individual Register Names and Addresses:

ADC_CR4: 1,ACh

	7	6	5	4	3	2	1	0
Access : POR		RW:0	RW:0	RW:0	RW:0	RW:0	RW:0	RW:0
Bit Name		D[13]	D[11]	D[9]	VDBCLK	ADCCLK	ADC_TST2	ADC_CMP

For additional information, see “Register Definitions” on page 442 in the 10-bit SAR ADC Controller chapter.

Bit	Name	Description
6/5/4	**D[13]/D[11]/D[9]**	Read-only registers. In chip-wide TEST mode, you read them as the value on D[13]/D[11]/D[9] which directly go to ADC comparator for data input. In non TEST mode, you read them always as 0s. They are for pure DFT purpose.
3/2	**VDBCLK/ADCCLK**	Read-only register. In chip-wide TEST mode, you read them as the value on voltage doubler clock and ADC comparator clock. In non TEST mode, you read them always as 0s. These two bits are for pure DFT purpose.
1	**ADC_TST2**	Can be set to 1 only in TEST mode.
0	**ADC_CMP**	Read-only bit and always 0 when ADC is in normal mode. In ADC_TST2 mode, you read it as the data from ADC comparator output. When ADC_TST2 is set to 1, the data written to it replaces the role of ADC comparator output data; that is, it becomes pseudo compare output.

12.3.49 I2C_ADDR

Individual Register Names and Addresses:

I2C_ADDR: 1,ADh

	7	6	5	4	3	2	1	0
Access : POR	RW:0	RW:000000						
Bit Name	HwAddrEn	Addr[6:0]						

The I2C address register is used to configure the hardware address automatic comparison feature so that the microcontroller will not be disturbed by an unwanted slave request. When HwAddrEn is enabled, the 7-bit address should be stored in Addr[6:0]; there is no interrupt only when the received address matches the stored address.

The hardware address automatic compare feature is available in slave only mode; master/slave mode is not supported. For additional information, see “Register Definitions” on page 396 in the I2C chapter.

Bit	Name	Description	
7	**HwAddrEn**	1	Enable hardware address comparison feature. Only supports 7-bit address. When you enable the hardware address comparison feature, I2C block will not support the special system address definition which is listed in I2C V2.1 spec, section 10 (for example: general call address, CBUS address, 10-bit slave address, and so on).
		0	Disable hardware address comparison feature.
6:0	**Addr**	Slave Address bits hold the slave's own device address.	

12.3.50 GDI_O_IN

Global Digital Interconnect Odd Inputs Register

Individual Register Names and Addresses:

GDI_O_IN: 1,D0h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	GIONOUT7	GIONOUT6	GIONOUT5	GIONOUT4	GIONOUT3	GIONOUT2	GIONOUT1	GIONOUT0

This register is used to configure a global input to drive a global output.

For additional information, refer to the "Register Definitions" on page 275 in the Global Digital Interconnect chapter.

Bit	Name	Description	
7	**GIONOUT7**	0	No connection between GIx[7]/GIx[6] to GOx[7].
		1	Allow GIx[7]/GIx[6] to drive GOx[7].
6	**GIONOUT6**	0	No connection between GIx[6]/GIx[5] to GOx[6].
		1	Allow GIx[6]/GIx[5] to drive GOx[6].
5	**GIONOUT5**	0	No connection between GIx[5]/GIx[4] to GOx[5].
		1	Allow GIx[5]/GIx[4] to drive GOx[5].
4	**GIONOUT4**	0	No connection between GIx[4]/GIx[3] to GOx[4].
		1	Allow GIx[4]/GIx[3] to drive GOx[4].
3	**GIONOUT3**	0	No connection between GIx[3]/GIx[2] to GOx[3].
		1	Allow GIx[3]/GIx[2] to drive GOx[3].
2	**GIONOUT2**	0	No connection between GIx[2]/GIx[1] to GOx[2].
		1	Allow GIx[2]/GIx[1] to drive GOx[2].
1	**GIONOUT1**	0	No connection between GIx[1]/GIx[0] to GOx[1].
		1	Allow GIx[1]/GIx[0] to drive GOx[1].
0	**GIONOUT0**	0	No connection between GIx[0]/GIx[7] to GOx[0].
		1	Allow GIx[0]/GIx[7] to drive GOx[0].

12.3.51 GDI_E_IN

Global Digital Interconnect Even Inputs Register

Individual Register Names and Addresses:

GDI_E_IN: 1,D1h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	GIENOUT7	GIENOUT6	GIENOUT5	GIENOUT4	GIENOUT3	GIENOUT2	GIENOUT1	GIENOUT0

This register is used to configure a global input to drive a global output.

For additional information, refer to the "Register Definitions" on page 275 in the Global Digital Interconnect chapter.

Bit	Name		Description
7	**GIENOUT7**	0	No connection between GIx[7]/GIx[6] to GOx[7].
		1	Allow GIx[7]/GIx[6] to drive GOx[7].
6	**GIENOUT6**	0	No connection between GIx[6]/GIx[5] to GOx[6].
		1	Allow GIx[6]/GIx[5] to drive GOx[6].
5	**GIENOUT5**	0	No connection between GIx[5]/GIx[4] to GOx[5].
		1	Allow GIx[5]/GIx[4] to drive GOx[5].
4	**GIENOUT4**	0	No connection between GIx[4]/GIx[3] to GOx[4].
		1	Allow GIx[4]/GIx[3] to drive GOx[4].
3	**GIENOUT3**	0	No connection between GIx[3]/GIx[2] to GOx[3].
		1	Allow GIx[3]/GIx[2] to drive GOx[3].
2	**GIENOUT2**	0	No connection between GIx[2]/GIx[1] to GOx[2].
		1	Allow GIx[2]/GIx[1] to drive GOx[2].
1	**GIENOUT1**	0	No connection between GIx[1]/GIx[0] to GOx[1].
		1	Allow GIx[1]/GIx[0] to drive GOx[1].
0	**GIENOUT0**	0	No connection between GIx[0]/GIx[7] to GOx[0].
		1	Allow GIx[0]/GIx[7] to drive GOx[0].

12.3.52 GDI_O_OU

Global Digital Interconnect Odd Outputs Register

Individual Register Names and Addresses:

GDI_O_OU: 1,D2h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	GOOUTIN7	GOOUTIN6	GOOUTIN5	GOOUTIN4	GOOUTIN3	GOOUTIN2	GOOUTIN1	GOOUTIN0

This register is used to configure a global output to drive a global input.

For additional information, refer to the "Register Definitions" on page 275 in the Global Digital Interconnect chapter.

Bit	Name	Description	
7	**GOOUTIN7**	0	No connection between GOx[7]/GOx[6] to GIx[7].
		1	Allow GOx[7]/GOx[6] to drive GIx[7].
6	**GOOUTIN6**	0	No connection between GOx[6]/GOx[5] to GIx[6].
		1	Allow GOx[6]/GOx[5] to drive GIx[6].
5	**GOOUTIN5**	0	No connection between GOx[0]/GOx[4] to GIx[5].
		1	Allow GOx[5]/GOx[4] to drive GIx[5].
4	**GOOUTIN4**	0	No connection between GOx[4]/GOx[3] to GIx[4].
		1	Allow GOx[4]/GOx[3] to drive GIx[4].
3	**GOOUTIN3**	0	No connection between GOx[3]/GOx[2] to GIx[3].
		1	Allow GOx[3]/GOx[2] to drive GIx[3].
2	**GOOUTIN2**	0	No connection between GOx[2]/GOx[1] to GIx[2].
		1	Allow GOx[2]/GOx[1] to drive GIx[2].
1	**GOOUTIN1**	0	No connection between GOx[1]/GOx[0] to GIx[1].
		1	Allow GOx[1]/GOx[0] to drive GIx[1].
0	**GOOUTIN0**	0	No connection between GOx[0]/GOx[7] to GIx[0].
		1	Allow GOx[0]/GOx[7] to drive GIx[0].

12.3.53 GDI_E_OU

Global Digital Interconnect Even Outputs Register

Individual Register Names and Addresses:

GDI_E_OU: 1,D3h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	GOEUTIN7	GOEUTIN6	GOEUTIN5	GOEUTIN4	GOEUTIN3	GOEUTIN2	GOEUTIN1	GOEUTIN0

This register is used to configure a global output to drive a global input.

For additional information, refer to the "Register Definitions" on page 275 in the Global Digital Interconnect chapter.

Bit	Name	Description	
7	**GOEUTIN7**	0	No connection between GOx[7]/GOx[6] to GIx[7].
		1	Allow GOx[7]/GOx[6] to drive GIx[7].
6	**GOEUTIN6**	0	No connection between GOx[6]/GOx[5] to GIx[6].
		1	Allow GOx[6]/GOx[5] to drive GIx[6].
5	**GOEUTIN5**	0	No connection between GOx[0]/GOx[4] to GIx[5].
		1	Allow GOx[5]/GOx[4] to drive GIx[5].
4	**GOEUTIN4**	0	No connection between GOx[4]/GOx[3] to GIx[4].
		1	Allow GOx[4]/GOx[3] to drive GIx[4].
3	**GOEUTIN3**	0	No connection between GOx[3]/GOx[2] to GIx[3].
		1	Allow GOx[3]/GOx[2] to drive GIx[3].
2	**GOEUTIN2**	0	No connection between GOx[2]/GOx[1] to GIx[2].
		1	Allow GOx[2]/GOx[1] to drive GIx[2].
1	**GOEUTIN1**	0	No connection between GOx[1]/GOx[0] to GIx[1].
		1	Allow GOx[1]/GOx[0] to drive GIx[1].
0	**GOEUTIN0**	0	No connection between GOx[0]/GOx[7] to GIx[0].
		1	Allow GOx[0]/GOx[7] to drive GIx[0].

12.3.54 MUX_CRx

Analog Mux Port Bit Enables Register

Individual Register Names and Addresses:

MUX_CR0 : 1,D8h MUX_CR1 : 1,D9h MUX_CR2 : 1,DAh MUX_CR3 : 1,DBh
MUX_CR4 : 1,ECh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	ENABLE[7:0]							

This register is used to control the connection between the analog mux bus and the corresponding pin.

The CY8C22x45 has a 6-bit wide Port 4 and the highest 2 bits of the MUX_CR4 register are reserved and will return zeros when read. For additional information, refer to the “Register Definitions” on page 426 in the I/O Analog Multiplexer chapter.

Bits	Name	Description
7:0	**ENABLE[7:0]**	Each bit controls the connection between the analog mux bus and the corresponding port pin. For example, MUX_CR2[3] controls the connection to bit 3 in Port 2. Any number of pins may be connected at the same time. Note that if a precharge clock is selected in the AMUX_CFG register, the connection to the mux bus will be switched on and off under hardware control. 0 No connection between port pin and analog mux bus. 1 Connect port pin to analog mux bus.

12.3.55 IDAC_CR1

Analog Mux DAC Control Register 1

Individual Register Names and Addresses:

IDAC_CR1 : 1,DCh

	7	6	5	4	3	2	1	0
Access : POR	RW:0	RW:0	RW:00	RW:1000				RW:0
Bit Name	ENR	MuxClkGER	ICEN	IDAC_TRIM				Double_Current

This register contains the control bits for the DAC current that drives the analog mux bus.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 426 in the I/O Analog Multiplexer chapter.

Bits	Name	Description
7	**ENR**	1 Enables right IDAC.
6	**MuxClkGER**	1 Enables driving right side MUXCLK to GOO[7].
5	**ICEN**	1 Enables ICEN. Both IDAC outputs are enabled. AGOEL: 0 selects IDACL_D to Left IDAC 1 se lets IDACR_D to Left IDAC AGOER: 0 se lets IDACL_D to Right IDAC 1 se lets IDACR_D to Right IDAC When ICEN is disabled, both IDAC will be controlled separately by their data registers and output enable signal AGOEL/AGOER.
4:1	**IDAC_TRIM[3:0]**	These signals go to PLL block and are used to trim IUNIT32 current output. The default value is 4'b1000 (ideally it is 10uA). Each step will change the current approximately by 3%.
0	**Double_Current**	This bit is used for IDAC current range control; it will combine with IDAC_CR0[3].IRANGE to define three different IDAC current ranges.

Double_Current	IDAC_CR0[3].IRANGE	Current Range
0 (Reserved)	0 (Reserved)	Reserved
1	0	4x
0	1	16x
1	1	32x

12.3.56 OSC_GO_EN

Oscillator to Global Outputs Enable Register

Individual Register Names and Addresses:

OSC_GO_EN: 1,DDh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	SLPINT	VC3	VC2	VC1	SYSCLKX2	SYSCLK	CLK24M	CLK32K

This register is used to enable tri-state buffers that connect specific system clocks to specific global output even nets.

For additional information, refer to the "Register Definitions" on page 381 in the Digital Clocks chapter. Also refer to the "PSoC Device Distinctions" on page 380, in the Digital Clocks chapter, for more information on availability of the OSC_GO_EN register bits.

Bit	Name	Description	
7	**SLPINT**	0	The sleep interrupt is not driven onto a global net.
		1	The sleep interrupt is driven onto GOE[7].
6	**VC3**	0	The VC3 clock is not driven onto a global net
		1	The VC3 clock is driven onto GOE[6]
5	**VC2**	0	The VC2 clock is not driven onto a global net
		1	The VC2 clock is driven onto GOE[5]
4	**VC1**	0	The VC1 clock is not driven onto a global net
		1	The VC1 clock is driven onto GOE[4]
3	**SYSCLKX2**	0	The 2 times system clock is not driven onto a global net
		1	The 2 times system clock is driven onto GOE[3]
2	**SYSCLK**	0	The system clock is not driven onto a global net
		1	The system clock is driven onto GOE[2]
1	**CLK24M**	0	The 24 MHz clock is not driven onto a global net
		1	The 24 MHz system clock is driven onto GOE[1]
0	**CLK32K**	0	The 32 kHz clock is not driven onto a global net
		1	The 32 kHz system clock is driven onto GOE[0]

12.3.57 OSC_CR4

Oscillator Control Register 4

Individual Register Names and Addresses:

OSC_CR4: 1,DEh

	7	6	5	4	3	2	1	0
Access : POR							RW : 0	
Bit Name							VC3 Input Select[1:0]	

This register selects the input clock to variable clock 3 (VC3).

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 381 in the Digital Clocks chapter.

Bit	Name	Description
1:0	**VC3 Input Select[1:0]**	Selects the clocking source for the VC3 Clock Divider. 00b SYSCLK 01b VC1 10b VC2 11b SYSCLKX2

12.3.58 OSC_CR3

Oscillator Control Register 3

Individual Register Names and Addresses:

OSC_CR3: 1,DFh

	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	VC3 Divider[7:0]							

This register selects the divider value for variable clock 3 (VC3).

The output frequency of the VC3 Clock Divider is the input frequency divided by the value in this register, plus one. For example, if this register contains 07h, the clock frequency output from the VC3 Clock Divider will be one eighth the input frequency. For additional information, refer to the “Register Definitions” on page 381 in the Digital Clocks chapter.

Bit	Name	Description
7:0	**VC3 Divider[7:0]**	Refer to the OSC_CR4 register. 00h Input Clock 01h Input Clock / 2 02h Input Clock / 3 03h Input Clock / 4 ... ... FCh Input Clock / 253 FDh Input Clock / 254 FEh Input Clock / 255 FFh Input Clock / 256

12.3.59 OSC_CR0

Oscillator Control Register 0

Individual Register Names and Addresses:

OSC_CR0: 1,E0h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0	RW : 0		RW : 0		
Bit Name	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]		

This register is used to configure various features of internal clock sources and clock nets.

Bits 7 and 6 in this register cannot be used by the CY8C21xxx, CY8C24x94, and CY7C64215 PSoC devices. For additional information, refer to the “Register Definitions” on page 381 in the Digital Clocks chapter.

Bit	Name	Description
7	**32k Select**	0 Internal low precision 32 kHz oscillator 1 External crystal 32.768 kHz oscillator
6	**PLL Mode**	0 Disabled 1 Enabled. Internal main oscillator is frequency locked to External Crystal Oscillator.
5	**No Buzz**	0 BUZZ bandgap during power down. 1 Bandgap is always powered even during sleep.
4:3	**Sleep[1:0]**	Sleep Interval when SLP_EXTEND=0 00b 1.95 ms (512 Hz) 01b 15.6 ms (64 Hz) 10b 125 ms (8 Hz) 11b 1 s (1 Hz) Sleep Interval when SLP_EXTEND=1 00b 2s (1/2 Hz) 01b 4s (1/4 Hz) 10b 10b 8s (1/8 Hz) 11b 11b 16 s (1/16 Hz)
2:0	**CPU Speed[2:0]**	These bits set the CPU clock speed, based on the system clock (SYSCLK). SYSCLK is 24 MHz by default, but it can optionally be set to 6 MHz on some PSoC devices (see the “Architectural Description” on page 75), or driven from an external clock.

	6 MHz IMO	24 MHz IMO	External Clock	
000b	750 kHz	3 MHz	EXTCLK / 8	
001b	1.5 MHz	6 MHz	EXTCLK / 4	
010b	3 MHz	12 MHz	EXTCLK / 2	
011b	6 MHz	24 MHz	EXTCLK / 1	Not available for CY7C603xx due to lower operating voltage.
100b	375 kHz	1.5 MHz	EXTCLK / 16	
101b	187.5 kHz	750 kHz	EXTCLK / 32	
110b	46.9 kHz	187.5 kHz	EXTCLK / 128	
111b	23.4 kHz	93.7 kHz	EXTCLK / 256	

12.3.60 OSC_CR1

Oscillator Control Register 1

Individual Register Names and Addresses:

OSC_CR1: 1,E1h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0				RW : 0			
Bit Name	VC1 Divider[3:0]				VC2 Divider[3:0]			

This register selects the divider value for variable clocks 1 and 2 (VC1 and VC2).

For additional information, refer to the “Register Definitions” on page 381 in the Digital Clocks chapter.

Bit	Name	Description		
7:4	**VC1 Divider[3:0]**		**Internal Main Oscillator**	**External Clock**
		0h	24 MHz	EXTCLK / 1
		1h	12 MHz	EXTCLK / 2
		2h	8 MHz	EXTCLK / 3
		3h	6 MHz	EXTCLK / 4
		4h	4.8 MHz	EXTCLK / 5
		5h	4 MHz	EXTCLK / 6
		6h	3.43 MHz	EXTCLK / 7
		7h	3 MHz	EXTCLK / 8
		8h	2.67 MHz	EXTCLK / 9
		9h	2.40 MHz	EXTCLK / 10
		Ah	2.18 MHz	EXTCLK / 11
		Bh	2.00 MHz	EXTCLK / 12
		Ch	1.85 MHz	EXTCLK / 13
		Dh	1.71 MHz	EXTCLK / 14
		Eh	1.6 MHz	EXTCLK / 15
		Fh	1.5 MHz	EXTCLK / 16
3:0	**VC2 Divider[3:0]**		**Internal Main Oscillator**	**External Clock**
		0h	(24 / (OSC_CR1[7:4]+1)) / 1	(EXTCLK / (OSC_CR1[7:4]+1)) / 1
		1h	(24 / (OSC_CR1[7:4]+1)) / 2	(EXTCLK / (OSC_CR1[7:4]+1)) / 2
		2h	(24 / (OSC_CR1[7:4]+1)) / 3	(EXTCLK / (OSC_CR1[7:4]+1)) / 3
		3h	(24 / (OSC_CR1[7:4]+1)) / 4	(EXTCLK / (OSC_CR1[7:4]+1)) / 4
		4h	(24 / (OSC_CR1[7:4]+1)) / 5	(EXTCLK / (OSC_CR1[7:4]+1)) / 5
		5h	(24 / (OSC_CR1[7:4]+1)) / 6	(EXTCLK / (OSC_CR1[7:4]+1)) / 6
		6h	(24 / (OSC_CR1[7:4]+1)) / 7	(EXTCLK / (OSC_CR1[7:4]+1)) / 7
		7h	(24 / (OSC_CR1[7:4]+1)) / 8	(EXTCLK / (OSC_CR1[7:4]+1)) / 8
		8h	(24 / (OSC_CR1[7:4]+1)) / 9	(EXTCLK / (OSC_CR1[7:4]+1)) / 9
		9h	(24 / (OSC_CR1[7:4]+1)) / 10	(EXTCLK / (OSC_CR1[7:4]+1)) / 10
		Ah	(24 / (OSC_CR1[7:4]+1)) / 11	(EXTCLK / (OSC_CR1[7:4]+1)) / 11
		Bh	(24 / (OSC_CR1[7:4]+1)) / 12	(EXTCLK / (OSC_CR1[7:4]+1)) / 12
		Ch	(24 / (OSC_CR1[7:4]+1)) / 13	(EXTCLK / (OSC_CR1[7:4]+1)) / 13
		Dh	(24 / (OSC_CR1[7:4]+1)) / 14	(EXTCLK / (OSC_CR1[7:4]+1)) / 14
		Eh	(24 / (OSC_CR1[7:4]+1)) / 15	(EXTCLK / (OSC_CR1[7:4]+1)) / 15
		Fh	(24 / (OSC_CR1[7:4]+1)) / 16	(EXTCLK / (OSC_CR1[7:4]+1)) / 16

12.3.61 OSC_CR2

Oscillator Control Register 2

Individual Register Names and Addresses:

OSC_CR2: 1,E2h

	7	6	5	4	3	2	1	0
Access : POR	RW : 0			RW : 0	RW : 0	RW : 0	RW : 0	RW : 0
Bit Name	PLLGAIN			SLP_EXTEND	WDR32_SE	EXTCLKEN	RSVD	SYSCLKX2DIS

This register is used to configure various features of internal clock sources and clock nets.

In OCD mode (OCDM=1), bits [1:0] have no effect. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 381 in the Digital Clocks chapter.

Bit	Name	Description
7	**PLLGAIN**	Phase-locked loop gain. 0 Recommended value, normal gain. 1 Reduced gain to make PLL more tolerant to noisy or jittery crystal input.
4	**SLP_EXTEND**	Long Sleep extension. 0 Disables the long sleep feature; max sleep period is 1s, default mode. 1 Enables the long sleep feature; max sleep period is 16s.
3	**WDR32_SE**	Watchdog clock source selection. 0 The same 32K clock source as system setting, default mode. 1 Uses internal 32K oscillator as clock source, even if external 32K clock source is enabled.
2	**EXTCLKEN**	External clock mode enable. 0 Disabled. Operate from internal main oscillator. 1 Enabled. Operate from clock supplied at port P1[4].
1	**RSVD**	Reserved bit - This bit should always be 0.
0	**SYSCLKX2DIS**	48 MHz clock source disable. 0 Enabled. If enabled, system clock net is forced on. 1 Disabled for power reduction.

12.3.62 VLT_CR

Voltage Monitor Control Register

Individual Register Names and Addresses:

VLT_CR: 1,E3h

	7	6	5	4	3	2	1	0
Access : POR			RW : 0		RW : 0	RW : 0		
Bit Name			PORLEV[1:0]		LVDTBEN	VM[2:0]		

This register is used to set the trip points for POR, LVD, and the supply pump.

Use the register tables above, in addition to the detailed register bit descriptions below, to determine which bits are reserved for some smaller PSoC devices. Note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 419 in the POR and LVD chapter.

Bit	Name	Description
5:4	**PORLEV[1:0]**	Sets the POR level per the DC electrical specifications in the PSoC device data sheet. See the table titled "PSoC Device Distinctions" on page 48 for PSoC device specific distinctions. 00b POR level for 2.4 V or 3 V operation (refer to the PSoC device data sheet) 01b POR level for 3.0 V or 4.5 V operation (refer to the PSoC device data sheet) 10b POR level for 4.75 V operation 11b Reserved
3	**LVDTBEN**	Enables reset of CPU speed register by LVD comparator output. 0 Disables CPU speed throttle-back. 1 Enables CPU speed throttle-back.
2:0	**VM[2:0]**	Sets the LVD and pump levels per the DC electrical specifications in the PSoC device data sheet, for those PSoC devices with this feature. 000b Lowest voltage setting 001b 010b . 011b . 100b . 101b 110b 111b Highest voltage setting

12.3.63 VLT_CMP

Voltage Monitor Comparators Register

Individual Register Names and Addresses:

VLT_CMP: 1,E4h

	7	6	5	4	3	2	1	0
Access : POR						R : 0	R : 0	R : 0
Bit Name						PUMP	LVD	PPOR

This register is used to read the state of internal supply voltage monitors.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 419 in the POR and LVD chapter.

Bit	Name	Description
2	**PUMP**	Read state of pump comparator. 0 Vdd is above trip point. 1 Vdd is below trip point.
1	**LVD**	Reads state of LVD comparator. 0 Vdd is above LVD trip point. 1 Vdd is below LVD trip point.
0	**PPOR**	Reads state of Precision POR comparator (only useful with PPOR reset disabled, with PORLEV[1:0] in VLT_CR register set to 11b). 0 Vdd is above PPOR trip voltage. 1 Vdd is below PPOR trip voltage.

12.3.64 ADCx_TR

ADC Column 0 and Column 1 Trim Register

Individual Register Names and Addresses:

ADC0_TR : 1,E5h ADC1_TR : 1,E6h

2L* Column	7	6	5	4	3	2	1	0
Access : POR	RW : 00							
Bit Name	CAPVAL_[7:0]							

* This table shows the two column limited functionality of the CY8C22x45, CY8C21345, CY8C21x34, CY8C21x23, CY7C603xx, CYWUSB6953 PSoC devices for this register.

This register controls a combination of capacitor and current values that determine the slope of the ADC voltage ramp.

ADC0_TR is the ADC column 0 trim register and ADC1_TR is the ADC column 1 trim register. For additional information, refer to the “Register Definitions” on page 349 in the Two Column Limited Analog System chapter.

Bits	Name	Description
7:0	CAPVAL_[7:0]	Controls, in binary weighted segments, the capacitor trim for ADC and general analog operation. This trim has a 16-1 range. By default (0000b), all capacitors are switched into the circuit, which is the maximum capacitance. 0 Switches that binary weighted capacitor segment into the circuit (more capacitance). 1 Switches that binary weighted capacitor segment out of the circuit (less capacitance).

12.3.65 VDAC_TRIM

VDAC 2BG Register

Individual Register Names and Addresses:

VDAC_TRIM: 1,E7h

	7	6	5	4	3	2	1	0
Access : POR		RW:0	RW:0	RW:0	RW:0100			
Bit Name		EN	V24	SAMPLE	TRIM[3:0]			

For additional information, refer to the “Register Definitions” on page 366 in the Two Column Analog Compare System chapter.

Bits	Name	Description
6	**EN**	1 Enables VDAC, otherwise the VDAC outputs vgnd.
5	**V24**	1 Sets regulated voltage default as 2.11 V; otherwise the default is 2*Vbg=2.6 V.
4	**SAMPLE**	1 Sets regulated voltage as VDAC reference input; otherwise uses noisy Vpwr as reference input.
3:0	**TRIM[3:0]**	Trim bits for regulated; default is 8. When V24=1: (1+(45+TRIM*2)/(112-TRIM*2))Vbg When V24=0: (1+(80+TRIM*2)/(112-TRIM*2))Vbg

12.3.66 IMO_TR

Internal Main Oscillator Trim Register

Individual Register Names and Addresses:

IMO_TR: 1,E8h

	7	6	5	4	3	2	1	0
Access : POR	W : 00							
Bit Name	Trim[7:0]							

This register is used to manually center the oscillator's output to a target frequency.

It is strongly recommended that the user not alter this register's values. The value in this register should not be changed. For additional information, refer to the "Register Definitions" on page 77 in the Internal Main Oscillator chapter.

Bit	Name	Description
7:0	**Trim[7:0]**	The value of this register is used to trim the Internal Main Oscillator. Its value is set to the best value for the device during boot. ***The value of these bits should not be changed***. 00h Lowest frequency setting 01h ... ... 7Fh 80h Design center setting 81h ... ... FEh FFh Highest frequency setting

12.3.67 ILO_TR

Internal Low Speed Oscillator Trim Register

Individual Register Names and Addresses:

ILO_TR: 1,E9h

	7	6	5	4	3	2	1	0
Access : POR			W : 0		W : 0			
Bit Name			Bias Trim[1:0]		Freq Trim[3:0]			

This register sets the adjustment for the Internal Low Speed Oscillator (ILO).

It is strongly recommended that the user not alter this register's values. The trim bits are set to factory specifications and should not be changed. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 79 in the Internal Low Speed Oscillator chapter.

Bit	Name	Description
5:4	**Bias Trim[1:0]**	The value of this register is used to trim the Internal Low Speed Oscillator. Its value is set to the device specific, best value during boot. ***The value of these bits should not be changed.*** 00b Medium bias 01b Maximum bias (recommended) 10b Minimum bias 11b Intermediate Bias * * About 15% higher than the minimum bias.
3:0	**Freq Trim[3:0]**	The value of this register is used to trim the Internal Low Speed Oscillator. Its value is set to the device specific, best value during boot. ***The value of these bits should not be changed.***

12.3.68 BDG_TR

Bandgap Trim Register

Individual Register Names and Addresses:

BDG_TR: 1,EAh

	7	6	5	4	3	2	1	0
Access : POR			RW : 1		RW : 8			
Bit Name			TC[1:0]		V[3:0]			

This register is used to adjust the bandgap and add an RC filter to AGND.

Note that reserved bits are grayed table cells and are not described in the bit description section. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 410 in the Internal Voltage Reference chapter.

Bit	Name	Description
5:4	**TC[1:0]**	The value of these bits is used to trim the temperature coefficient. Their value is set to the best value for the device during boot. ***The value of these bits should not be changed.***
3:0	**V[3:0]**	The value of these bits is used to trim the bandgap reference. Their value is set to the best value for the device during boot. ***The value of these bits should not be changed.***

12.3.69 ECO_TR

External Crystal Oscillator Trim Register

Individual Register Names and Addresses:

ECO_TR: 1,EBh

	7	6	5	4	3	2	1	0
Access : POR	W : 0							
Bit Name	PSSDC[1:0]							

This register sets the adjustment for the 32.768 kHz External Crystal Oscillator.

The value in this register should not be changed. The value is used to trim the 32.768 kHz external crystal oscillator and is set to the device specific, best value during boot. In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the "Register Definitions" on page 82 in the External Crystal Oscillator (ECO) chapter.

Bit	Name	Description
7:6	**PSSDC[1:0]**	Sleep duty cycle. Controls the ratios (in numbers of 32.768 kHz clock periods) of "on" time versus "off" time for PORLVD, Bandgap reference, and pspump. ***These bits should not be changed.*** 00b 1 / 128 01b 1 / 512 10b 1 / 32 11b 1 / 8

12.3.70 FLS_PR1

Flash Program Register 1

Individual Register Names and Addresses:

FLS_PR1: 1,FAh

	7	6	5	4	3	2	1	0
Access : POR							RW : 0	
Bit Name							Bank[1:0]	

This register is used to specify which Flash bank should be used for SROM operations.

In the table above, note that reserved bits are grayed table cells and are not described in the bit description section below. Reserved bits should always be written with a value of '0'. For additional information, refer to the Supervisory ROM (SROM) chapter on page 43.

Bit	Name	Description
1:0	**Bank[1:0]**	Selects the active Flash bank for supervisory operations. No affect in User mode. 00b Flash Bank 0 01b Flash Bank 1 10b Flash Bank 2, Reserved 11b Flash Bank 3, Reserved

12.3.71 IDAC_CR0

Analog Mux DAC Control Register 0

Individual Register Names and Addresses:

IDAC_CR0 : 1,FDh

	7	6	5	4	3	2	1	0
Access : POR	RW : 0	RW : 0	RW : 0		RW : 0	RW : 0		RW : 0
Bit Name	SplitMux	MuxClkGEL	OSCMDR[1:0]		IRANGE	OSCMDL[1:0]		ENL

This register contains the control bits for the DAC current that drives the analog mux bus and for selecting the split configuration.

For additional information, refer to the "Register Definitions" on page 426 in the I/O Analog Multiplexer chapter.

Bits	Name	Description
7	**SplitMux**	Configures the analog mux bus for the CY8C24x94 and CY7C64215 PSoC devices. Left side connects to odd pins (P0[1], P5[5]) and right side connects to even pins (P0[2], P5[6]) with one exception: P0[7] is a right side pin. 0 Split analog mux bus: left side pins connect to Analog Mux Bus Left and right side pins connect to Analog Mux Bus Right. 1 Single analog mux bus.
6	**MuxClkGEL**	Global enable connection for MUXCLK in the CY8C24x94 and CY7C64215 PSoC devices, or MUXCLK Left in the CY8C22x45 PSoC devices. 0 Analog mux bus clock not connected to global. 1 Connect analog mux bus clock to global GOO[6].
5:4	**OSCMDR[1:0]**	When set, these bits enable the analog mux bus right to reset to Vss whenever the comparator trip point is reached. These bits are only used in the CY8C22x45 and CY8C21345 PSoC devices. 00b No automatic reset. 01b Reset whenever GOO[6 is high. 10b Reset whenever GOO[7] is high. 11b Reset whenever either GOO[6] or GOO[7] is high.
3	**IRANGE**	This bit is used for IDAC current range control; it will combine with IDAC_CR1[0].Double_Current to define three different IDAC current ranges. **IDAC_CR1[0].Double_Current** / **IRANGE** / **Current Range** 0(Reserved) / 0 (Reserved) / Reserved 1 / 0 / 4x 0 / 1 / 16x 1 / 1 / 32x
2:1	**OSCMDL[1:0]**	When set, these bits enable the analog mux bus left (or analog mux bus right) to reset to Vss whenever the comparator trip point is reached. These bits are only used in the CY8C22x45 and CY8C21345 PSoC devices. 00b No automatic reset. 01b Reset whenever GOO[4] is high. 10b Reset whenever GOO[5] is high. 11b Reset whenever either GOO[4] or GOO[5] is high.

(continued on next page)

12.3.71 IDAC_CR0 *(continued)*

0	**ENL**	0	DAC function disabled (left DAC function disabled in the CY8C22x45 and CY8C21345).
		1	DAC function enabled (left DAC function enabled in the CY8C22x45 and CY8C21345). The DAC current charges the analog mux bus (or analog mux bus left in the CY8C22x45 and CY8C21345). In the CY8C24x94 and CY7C64215, if the SplitMux is set high, the charging current only charges the mux bus right.

Section D: Digital System

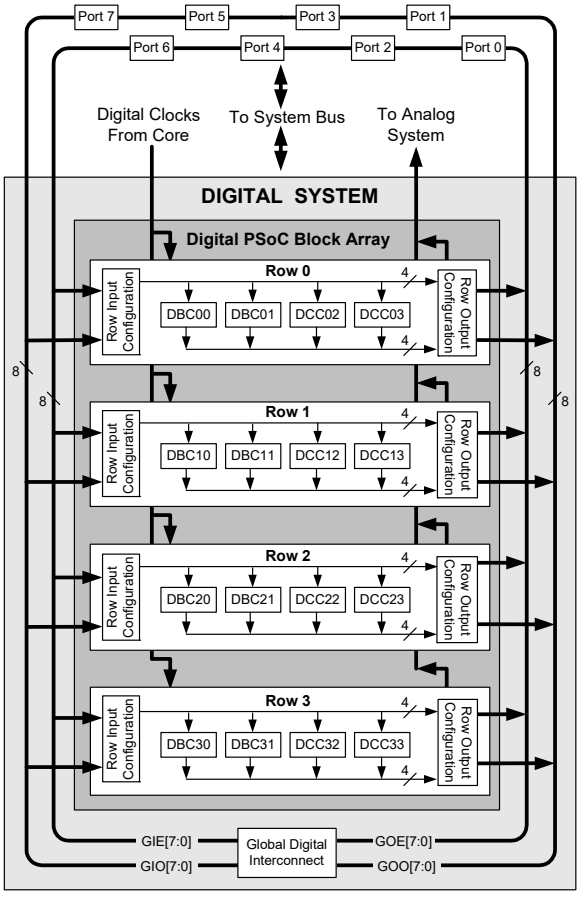
The configurable Digital System section discusses the digital components of the PSoC device and the registers associated with those components. This section encompasses the following chapters:

- Global Digital Interconnect (GDI) on page 272
- Array Digital Interconnect (ADI) on page 278
- Row Digital Interconnect (RDI) on page 280
- Digital Blocks on page 288

Top-Level Digital Architecture

The figure below displays the top-level architecture of the PSoC's digital system. Each component of the figure is discussed at length in this section.

Figure 13-1. PSoC Digital System Block Diagram



Interpreting the Digital Documentation

Information in this section covers all PSoC devices with a base part number of CY8C2xxxx (except for the CY8C25122 and CY8C26xxx PSoC devices). It also applies to CY7C64215, CY7C603xx, and CYWUSB6953. The primary digital distinction between these devices is the number of digital rows. This can be either 1, 2, or 4 rows. The following table lists the resources available for specific device groups. While reading the digital system section, determine and keep in mind the number of digital rows that are in your device, to accurately interpret this documentation.

PSoC Device Characteristics

PSoC Part Number	Digital I/O (max)	Digital Rows	Digital Blocks	Analog Inputs	Analog Outputs	Analog Columns	Analog Blocks
CY8C29x66	**64**	**4**	**16**	12	4	4	12
CY8C27x43	**44**	**2**	**8**	12	4	4	12
CY8C24x94	**50**	**1**	**4**	48	2	2	6
CY8C24x23	**24**	**1**	**4**	12	2	2	6
CY8C24x23A	**24**	**1**	**4**	12	2	2	6
CY8C22x45	**38**	**2**	**8**	10	0	4	6
CY8C22x13	**16**	**1**	**4**	10	1	1	3
CY8C21345	**24**	**1**	**4**	10	0	4	6
CY8C21x34	**28**	**1**	**4**	28	0	2	4 *
CY8C21x23	**16**	**1**	**4**	8	0	2	4 *
CY7C64215	**50**	**1**	**4**	48	2	2	6
CY7C603xx	**28**	**1**	**4**	28	0	2	4 *
CYWUSB6953	**28**	**1**	**4**	28	0	2	4 *

* Limited analog functionality.

Digital Register Summary

The table below lists all the PSoC registers for the digital system in address order (Add. column) within their system resource configuration. The bits that are grayed out are reserved bits. If these bits are written, they should always be written with a value of '0'. The naming conventions for the digital row registers and the digital block registers are detailed in their respective table title rows.

Note that all PSoC devices with a base part number of CY8C2xxxx (except for the CY8C25122 and CY8C26xxx PSoC devices), fall into one of the following categories with respect to their digital PSoC rows: 4 row device, 2 row device, or 1 row device. It also applies to CY7C64215, CY7C603xx, and CYWUSB6953. The "PSoC Digital System Block Diagram" at the beginning of this section illustrates this.

In the table below, the third column from the left titled "Digital Rows" indicates which of the three PSoC device categories the register falls into. To determine the number of digital rows in your PSoC device, refer to the table titled "PSoC Device Characteristics" on page 268.

Summary Table of the Digital Registers

Add.	Name	Digital Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
GLOBAL DIGITAL INTERCONNECT (GDI) REGISTERS (page 275)											
1,A0h	GDI_O_IN_CR	4, 3, 2, 1	GDIOICR7	GDIOICR6	GDIOICR5	GDIOICR4	GDIOICR3	GDIOICR2	GDIOICR1	GDIOICR0	RW : 00
1,A1h	GDI_E_IN_CR	4, 3, 2, 1	GDIEICR7	GDIEICR6	GDIEICR5	GDIEICR4	GDIEICR3	GDIEICR2	GDIEICR1	GDIEICR0	RW : 00
1,A2h	GDI_O_OU_CR	4, 3, 2, 1	GDIOOCR7	GDIOOCR6	GDIOOCR5	GDIOOCR4	GDIOOCR3	GDIOOCR2	GDIOOCR1	GDIOOCR0	RW : 00
1,A3h	GDI_E_OU_CR	4, 3, 2, 1	GDIEOCR7	GDIEOCR6	GDIEOCR5	GDIEOCR4	GDIEOCR3	GDIEOCR2	GDIEOCR1	GDIEOCR0	RW : 00
1,D0h	GDI_O_IN	4, 3, 2, 1	GIONOUT7	GIONOUT6	GIONOUT5	GIONOUT4	GIONOUT3	GIONOUT2	GIONOUT1	GIONOUT0	RW : 00
1,D1h	GDI_E_IN	4, 3, 2, 1	GIENOUT7	GIENOUT6	GIENOUT5	GIENOUT4	GIENOUT3	GIENOUT2	GIENOUT1	GIENOUT0	RW : 00
1,D2h	GDI_O_OU	4, 3, 2, 1	GOOUTIN7	GOOUTIN6	GOOUTIN5	GOOUTIN4	GOOUTIN3	GOOUTIN2	GOOUTIN1	GOOUTIN0	RW : 00
1,D3h	GDI_E_OU	4, 3, 2, 1	GOEUTIN7	GOEUTIN6	GOEUTIN5	GOEUTIN4	GOEUTIN3	GOEUTIN2	GOEUTIN1	GOEUTIN0	RW : 00
DIGITAL ROW REGISTERS (page 282)											
x,B0h	RDI0RI	4, 3, 2, 1	RI3[1:0]		RI2[1:0]		RI1[1:0]		RI0[1:0]		RW : 00
x,B1h	RDI0SYN	4, 3, 2, 1					RI3SYN	RI2SYN	RI1SYN	RI0SYN	RW : 00
x,B2h	RDI0IS	4, 3, 2, 1			BCSEL[1:0]		IS3	IS2	IS1	IS0	RW : 00
x,B3h	RDI0LT0	4, 3, 2, 1	LUT1[3:0]				LUT0[3:0]				RW : 00
x,B4h	RDI0LT1	4, 3, 2, 1	LUT3[3:0]				LUT2[3:0]				RW : 00
x,B5h	RDI0RO0	4, 3, 2, 1	GOO5EN	GOO1EN	GOE5EN	GOE1EN	GOO4EN	GOO0EN	GOE4EN	GOE0EN	RW : 00
x,B6h	RDI0RO1	4, 3, 2, 1	GOO7EN	GOO3EN	GOE7EN	GOE3EN	GOO6EN	GOO2EN	GOE6EN	GOE2EN	RW : 00
x,B7h	RDI0DSM	4, 3, 2, 1		AVS_SEL[2:0]			AVG_EN[3:0]				RW : 00
x,B8h	RDI1RI	4, 3, 2	RI3[1:0]		RI2[1:0]		RI1[1:0]		RI0[1:0]		RW : 00
x,B9h	RDI1SYN	4, 3, 2					RI3SYN	RI2SYN	RI1SYN	RI0SYN	RW : 00
x,BAh	RDI1IS	4, 3, 2			BCSEL[1:0]		IS3	IS2	IS1	IS0	RW : 00
x,BBh	RDI1LT0	4, 3, 2	LUT1[3:0]				LUT0[3:0]				RW : 00
x,BCh	RDI1LT1	4, 3, 2	LUT3[3:0]				LUT2[3:0]				RW : 00
x,BDh	RDI1RO0	4, 3, 2	GOO5EN	GOO1EN	GOE5EN	GOE1EN	GOO4EN	GOO0EN	GOE4EN	GOE0EN	RW : 00
x,BEh	RDI1RO1	4, 3, 2	GOO7EN	GOO3EN	GOE7EN	GOE3EN	GOO6EN	GOO2EN	GOE6EN	GOE2EN	RW : 00
x,BFh	RDI1DSM	4, 3, 2		AVS_SEL[2:0]			AVG_EN[3:0]				RW : 00

Summary Table of the Digital Registers *(continued)*

Add.	Name	Digital Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
DIGITAL BLOCK REGISTERS (page 300)											
Digital Block Data and Control Registers (page 300)											
0,20h	DBC00DR0	4, 3, 2, 1	Data[7:0]								# : 00
0,21h	DBC00DR1	4, 3, 2, 1	Data[7:0]								W : 00
0,22h	DBC00DR2	4, 3, 2, 1	Data[7:0]								# : 00
0,23h	DBC00CR0	4, 3, 2, 1	Function control/status bits for selected function[6:0]							Enable	# : 00
1,20h	DBC00FN	4, 3, 2, 1	Data Invert	BCEN	End Single	Mode[1:0]		Function[2:0]			RW : 00
1,21h	DBC00IN	4, 3, 2, 1	Data Input[3:0]				Clock Input[3:0]				RW : 00
1,22h	DBC00OU	4, 3, 2, 1	AUXCLK		AUXEN	AUX I/O Select[1:0]		OUTEN	Output Select[1:0]		RW : 00
1,23h	DBC00CR1	4, 3, 2, 1	Function control/status bits for selected function[7:0]								RW : 00
0,24h	DBC01DR0	4, 3, 2, 1	Data[7:0]								# : 00
0,25h	DBC01DR1	4, 3, 2, 1	Data[7:0]								W : 00
0,26h	DBC01DR2	4, 3, 2, 1	Data[7:0]								# : 00
0,27h	DBC01CR0	4, 3, 2, 1	Function control/status bits for selected function[6:0]							Enable	# : 00
1,24h	DBC01FN	4, 3, 2, 1	Data Invert	BCEN	End Single	Mode[1:0]		Function[2:0]			RW : 00
1,25h	DBC01IN	4, 3, 2, 1	Data Input[3:0]				Clock Input[3:0]				RW : 00
1,26h	DBC01OU	4, 3, 2, 1	AUXCLK		AUXEN	AUX I/O Select[1:0]		OUTEN	Output Select[1:0]		RW : 00
1,27h	DBC01CR1	4, 3, 2, 1	Function control/status bits for selected function[7:0]								RW : 00
0,28h	DCC02DR0	4, 3, 2, 1	Data[7:0]								# : 00
0,29h	DCC02DR1	4, 3, 2, 1	Data[7:0]								W : 00
0,2Ah	DCC02DR2	4, 3, 2, 1	Data[7:0]								# : 00
0,2Bh	DCC02CR0	4, 3, 2, 1	Function control/status bits for selected function[6:0]							Enable	# : 00
1,28h	DCC02FN	4, 3, 2, 1	Data Invert	BCEN	End Single	Mode[1:0]		Function[2:0]			RW : 00
1,29h	DCC02IN	4, 3, 2, 1	Data Input[3:0]				Clock Input[3:0]				RW : 00
1,2Ah	DCC02OU	4, 3, 2, 1	AUXCLK		AUXEN	AUX I/O Select[1:0]		OUTEN	Output Select[1:0]		RW : 00
1,2Bh	DCC02CR1	4, 3, 2, 1	Function control/status bits for selected function[7:0]								RW : 00
0,2Ch	DCC03DR0	4, 3, 2, 1	Data[7:0]								# : 00
0,2Dh	DCC03DR1	4, 3, 2, 1	Data[7:0]								W : 00
0,2Eh	DCC03DR2	4, 3, 2, 1	Data[7:0]								# : 00
0,2Fh	DCC03CR0	4, 3, 2, 1	Function control/status bits for selected function[6:0]							Enable	# : 00
1,2Ch	DCC03FN	4, 3, 2, 1	Data Invert	BCEN	End Single	Mode[1:0]		Function[2:0]			RW : 00
1,2Dh	DCC03IN	4, 3, 2, 1	Data Input[3:0]				Clock Input[3:0]				RW : 00
1,2Eh	DCC03OU	4, 3, 2, 1	AUXCLK		AUXEN	AUX I/O Select[1:0]		OUTEN	Output Select[1:0]		RW : 00
1,2Fh	DCC03CR1	4, 3, 2, 1	Function control/status bits for selected function[7:0]								RW : 00
0,30h	DBC10DR0	4, 3, 2	Data[7:0]								# : 00
0,31h	DBC10DR1	4, 3, 2	Data[7:0]								W : 00
0,32h	DBC10DR2	4, 3, 2	Data[7:0]								# : 00
0,33h	DBC10CR0	4, 3, 2	Function control/status bits for selected function[7:1]							Enable	# : 00
1,30h	DBC10FN	4, 3, 2	Data Invert	BCEN	End Single	Mode[1:0]		Function[2:0]			RW : 00
1,31h	DBC10IN	4, 3, 2	Data Input[3:0]				Clock Input[3:0]				RW : 00
1,32h	DBC10OU	4, 3, 2	AUXCLK		AUXEN	AUX I/O Select[1:0]		OUTEN	Output Select[1:0]		RW : 00
1,33h	DBC10CR1	4, 3, 2	Function control/status bits for selected function[7:0]								RW : 00
0,34h	DBC11DR0	4, 3, 2	Data[7:0]								# : 00
0,35h	DBC11DR1	4, 3, 2	Data[7:0]								W : 00
0,36h	DBC11DR2	4, 3, 2	Data[7:0]								# : 00
0,37h	DBC11CR0	4, 3, 2	Function control/status bits for selected function[7:1]							Enable	# : 00
1,34h	DBC11FN	4, 3, 2	Data Invert	BCEN	End Single	Mode[1:0]		Function[2:0]			RW : 00
1,35h	DBC11IN	4, 3, 2	Data Input[3:0]				Clock Input[3:0]				RW : 00
1,36h	DBC11OU	4, 3, 2	AUXCLK		AUXEN	AUX I/O Select[1:0]		OUTEN	Output Select[1:0]		RW : 00
1,37h	DBC11CR1	4, 3, 2	Function control/status bits for selected function[7:0]								RW : 00

Summary Table of the Digital Registers *(continued)*

Add.	Name	Digital Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,38h	DCC12DR0	4, 3, 2	Data[7:0]								# : 00
0,39h	DCC12DR1	4, 3, 2	Data[7:0]								W : 00
0,3Ah	DCC12DR2	4, 3, 2	Data[7:0]								# : 00
0,3Bh	DCC12CR0	4, 3, 2	Function control/status bits for selected function[7:1]							Enable	# : 00
1,38h	DCC12FN	4, 3, 2	Data Invert	BCEN	End Single	Mode[1:0]		Function[2:0]			RW : 00
1,39h	DCC12IN	4, 3, 2	Data Input[3:0]				Clock Input[3:0]				RW : 00
1,3Ah	DCC12OU	4, 3, 2	AUXCLK		AUXEN	AUX I/O Select[1:0]		OUTEN	Output Select[1:0]		RW : 00
1,3Bh	DCC12CR1	4, 3, 2	Function control/status bits for selected function[7:0]								RW : 00
0,3Ch	DCC13DR0	4, 3, 2	Data[7:0]								# : 00
0,3Dh	DCC13DR1	4, 3, 2	Data[7:0]								W : 00
0,3Eh	DCC13DR2	4, 3, 2	Data[7:0]								# : 00
0,3Fh	DCC13CR0	4, 3, 2	Function control/status bits for selected function[7:1]							Enable	# : 00
1,3Ch	DCC13FN	4, 3, 2	Data Invert	BCEN	End Single	Mode[1:0]		Function[2:0]			RW : 00
1,3Dh	DCC13IN	4, 3, 2	Data Input[3:0]				Clock Input[3:0]				RW : 00
1,3Eh	DCC13OU	4, 3, 2	AUXCLK		AUXEN	AUX I/O Select[1:0]		OUTEN	Output Select[1:0]		RW : 00
1,3Fh	DCC13CR1	4, 3, 2	Function control/status bits for selected function[7:0]								RW : 00
Digital Block Interrupt Mask Registers (page 313)											
0,DEh	INT_MSK3		ENSWINT							I2C	RW : 00
0,DFh	INT_MSK2	4, 3	RTC	CSD1	CSD0	SARADC					RW : 00
0,E0h	INT_MSK0		VC3	Sleep	GPIO	Compare 1	Compare 0	Analog 1	Analog 0	V Monitor	RW : 00
			VC3	Sleep	GPIO			Analog 1	Analog 0	V Monitor	
			VC3	Sleep	GPIO			Analog 1		V Monitor	
0,E1h	INT_MSK1	4, 3, 2	DCC13	DCC12	DBC11	DBC10	DCC03	DCC02	DBC01	DBC00	RW : 00
		1					DCC03	DCC02	DBC01	DBC00	

LEGEND
x An 'x' before the comma in the address field indicates that this register can be read or written to no matter what bank is used. R: Read register or bit(s).
Access is bit specific. Refer to the Register Details chapter on page 103 for additional information.
R Read register or bit(s).
W Write register or bit(s).

13. Global Digital Interconnect (GDI)

This chapter discusses the Global Digital Interconnect (GDI) and its associated registers. All PSoC CY8C2xxxx devices (except for the CY8C25122 and CY8C26xxx PSoC devices) have the exact same global digital interconnect options, varying only in the number of 8-bit ports connected to the globals. For a complete table of the GDI registers, refer to the "Summary Table of the Digital Registers" on page 269. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

13.1 Architectural Description

Global Digital Interconnect (GDI) consists of four 8-bit buses (refer to the figures that follow). Two of the buses are input buses, which allow signals to pass from the device pins to the core of the PSoC device. These buses are called Global Input Odd (GIO[7:0]) and Global Input Even (GIE[7:0]). The other two buses are output buses that allow signals to pass from the core of the PSoC device to the device pins. They are called Global Output Odd (GOO[7:0]) and Global Output Even (GOE[7:0]). The word "odd" or "even" in the bus name indicates which device ports the bus connects to. Busses with odd in their name connect to all odd numbered ports. Busses with even in their name connect to all even numbered ports.

There are two ends to the global digital interconnect core signals and port pins. An end may be configured as a source or a destination. For example, a GPIO pin may be configured to drive a global input or receive a global output and drive it to the package pin. Globals cannot "loop through" a GPIO. Currently, there are two types of core signals connected to the global buses. The digital blocks, which may be a source or a destination for a global ***net***, and system clocks, which may only drive global nets.

Many of the digital clocks may also be driven on to the global bus to allow the clocks to route directly to I/O pins. This is shown in the global interconnect block diagrams on the following pages. For more information on this feature, see the Digital Clocks chapter on page 377.

Each global input and global output has a ***keeper*** on it. The keeper sets the value of the global to '1' on system reset and holds the last driven value of the global should it become un-driven.

The primary goal, of the architectural block diagrams that follow, is to communicate the relationship between global buses (GOE, GOO, GIE, GIO) and pins. Note that any global input may be connected to its corresponding global output, using the tri-state buffers located in the corners of the figures. Also, global outputs may be shorted to global inputs using these tri-state buffers. The rectangle in the center of the figure represents the array of digital PSoC blocks.

13.1.1 28-Pin Global Interconnect

For 28-pin PSoC devices , there are three 8-bit ports. Therefore, there are two ports connected to the even global buses and one port connected to the odd global buses. Table 13-1 lists the mapping between global buses and ports.

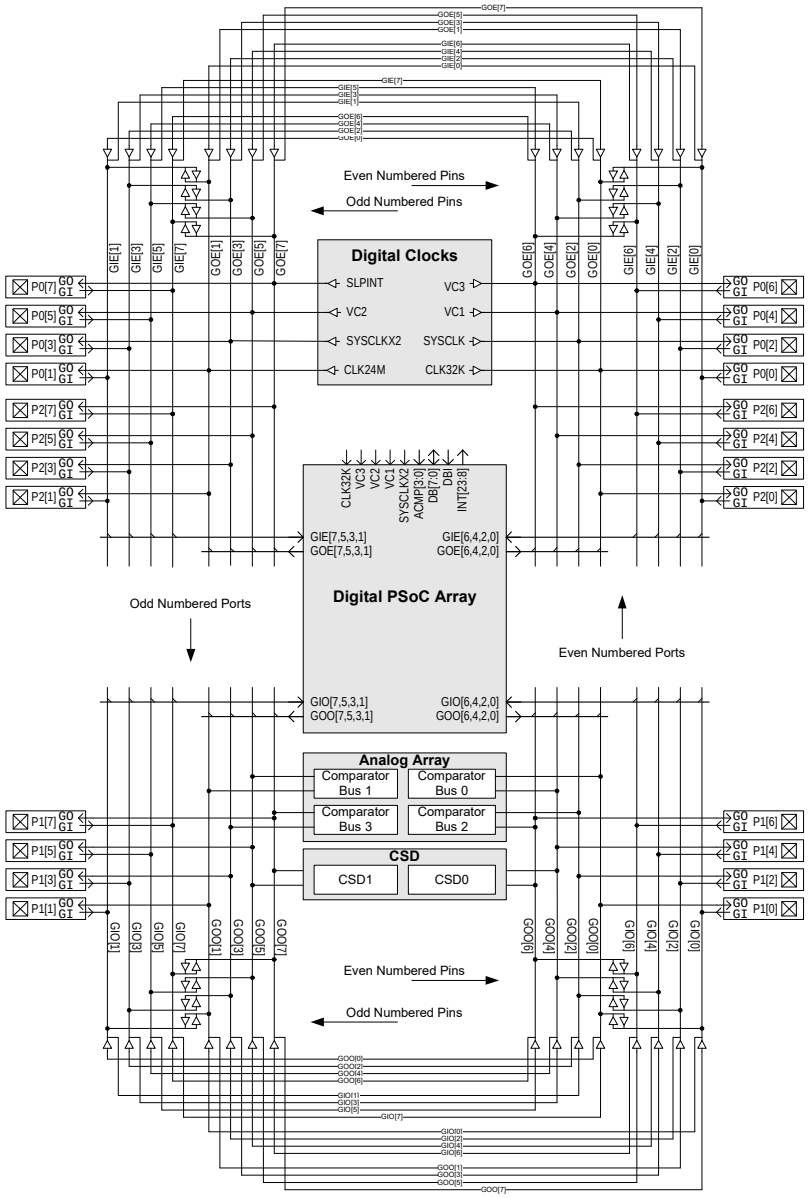
Table 13-1. 28-Pin Global Bus to Port Mapping

Global Bus	Ports
GIO[7:0], GOO[7:0]	P1
GIE[7:0], GOE[7:0]	P0, P2

Because up to two ports are connected to a single global bus, there is a one-to-many mapping between individual nets in a global bus and port pins. For example, if GIE[1] is used to bring an input signal into a digital PSoC block, either pin P0[1] or P2[1] may be used. The same is true for the outputs. For example, if GOE[3] is used to carry a signal from a digital PSoC block to a port pin, either or both of the following pins may be used: P0[3] or P2[3]. Only Port 1 pins connect to the GIO/GOO globals in these 28-pin PSoC devices.

To determine the number of digital rows and digital blocks in your PSoC device, refer to the table titled "PSoC Device Characteristics" on page 268.

Figure 13-1. Global Interconnect Block Diagram for a 28-Pin Package



13.1.2 44-Pin Global Interconnect

For 44-pin PSoC devices, there are five 8-bit ports. Therefore, there are up to three ports connected to the even global buses and two ports connected to the odd global buses. Table 13-2 lists the mapping between global buses and ports.

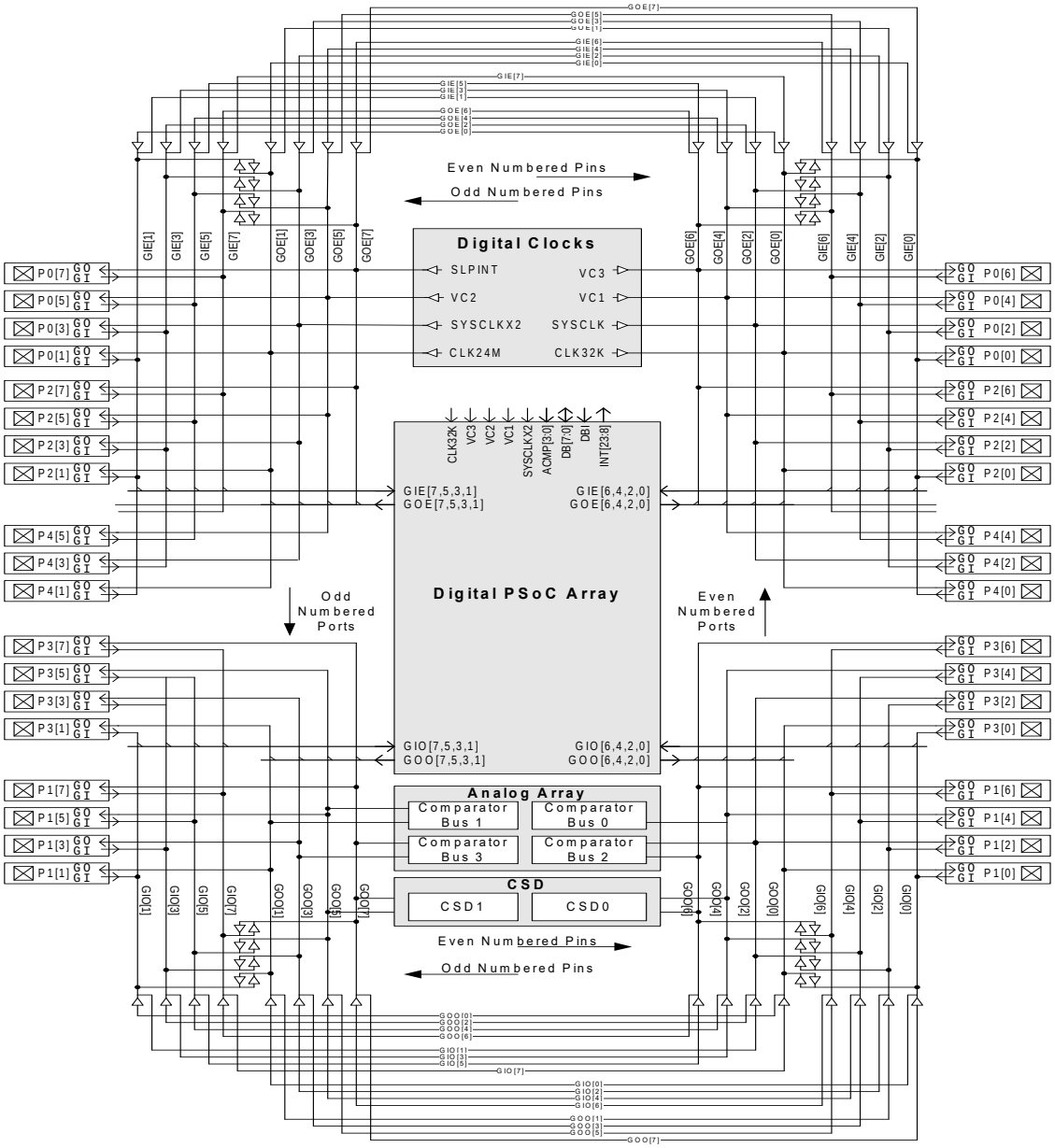
Table 13-2. 44-Pin Global Bus to Port Mapping

Global Bus	Ports
GIO[7:0], GOO[7:0]	P1, P3
GIE[7:0], GOE[7:0]	P0, P2, P4

Because several ports are connected to a single global bus, there is a one-to-many mapping between individual nets in a global bus and port pins. For example, if GIO[1] is used to bring an input signal into a digital PSoC block, either pin P1[1] or P3[1] may be used. The same is true for the outputs. For example, if GOE[3] is used to carry a signal from a digital PSoC block to a port pin, any or all of the following pins may be used: P0[3], P2[3], or P4[3].

To determine the number of digital rows and digital blocks in your PSoC device, refer to the table titled “PSoC Device Characteristics” on page 268.

Figure 13-2. Global Interconnect Block Diagram for a 44-Pin Package



13.1.3 56-Pin Global Interconnect

The 56-pin PSoC device is only for on-chip debugging (OCD) purposes. Therefore, the 56-pin global connection is the same as the 44-pin package.

13.2 Register Definitions

The following registers are associated with the Global Digital Interconnect and are listed in address order. Each register description has an associated register table showing the bit structure for that register. For a complete table of GDI registers, refer to the "Summary Table of the Digital Registers" on page 269.

In the PSoC device with two digital rows, the configurable GDI is used to resynchronize the ***feedback*** between two digital PSoC blocks. This is accomplished by connecting a digital PSoC block's output to a global output that has been configured to drive its corresponding global input. The global input is chosen to drive one of the row inputs. The row input is configured to synchronize the signal to the device's 24 MHz system clock. Finally, the row input is used by the second digital PSoC block.

13.2.1 GDI_x_IN Registers/GDI_x_IN_CR Registers

GDI_x_IN										
Add.	**Name**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**Access**
1,D0h	GDI_O_IN	GIONOUT7	GIONOUT6	GIONOUT5	GIONOUT4	GIONOUT3	GIONOUT2	GIONOUT1	GIONOUT0	RW : 00
1,D1h	GDI_E_IN	GIENOUT7	GIENOUT6	GIENOUT5	GIENOUT4	GIENOUT3	GIENOUT2	GIENOUT1	GIENOUT0	RW : 00
GDI_x_IN_CR										
Add.	**Name**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**Access**
1,A0h	GDI_O_IN_CR	GDIOICR7	GDIOICR6	GDIOICR5	GDIOICR4	GDIOICR3	GDIOICR2	GDIOICR1	GDIOICR0	RW : 00
1,A1h	GDI_E_IN_CR	GDIEICR7	GDIEICR6	GDIEICR5	GDIEICR4	GDIEICR3	GDIEICR2	GDIEICR1	GDIEICR0	RW : 00

The Global Digital Interconnect Odd and Even Input Registers (GDI_x_IN/GDI_x_IN_CR) are used to configure a global input to drive a global output.

The PSoC device has a configurable Global Digital Interconnect (GDI). Note that the GDI_x_IN and GDI_x_OU registers should never have the same bits connected. This would result in multiple drivers of one bus.

Bits 7 to 0: GIxNOUTx. Using the configuration bits in the GDI_x_IN registers, a global input net may be configured to drive its corresponding global output net. For example,

$$GIE[7] \rightarrow GOE[7]$$

The configurability of the GDI does not allow odd and even nets or nets with different indexes to be connected. The following are examples of connections that are not possible in the PSoC devices.

$$GOE[7] \not\rightarrow GIO[7]$$
$$GOE[0] \not\rightarrow GIE[7]$$
$$GOE[0] \rightarrow GIE[1]$$

There are a total of 16 bits that control the ability of global inputs to drive global outputs. These bits are in the GDI_x_IN registers. Table 13-3 enumerates the meaning of each bit position in either of the GDI_O_IN or GDI_E_IN registers.

Table 13-3. GDI_x_IN Register

GDI_x_IN[0]	0: No connection between GIx[0]/GIx[7] to GOx[0] 1: Allow GIx[0]/GIx[7] to drive GOx[0]
GDI_x_IN[1]	0: No connection between GIx[1]/GIx[0] to GOx[1] 1: Allow GIx[1]/GIx[0] to drive GOx[1]
GDI_x_IN[2]	0: No connection between GIx[2]/GIx[1] to GOx[2] 1: Allow GIx[2]/GIx[1] to drive GOx[2]
GDI_x_IN[3]	0: No connection between GIx[3]/GIx[2] to GOx[3] 1: Allow GIx[3]/GIx[2] to drive GOx[3]
GDI_x_IN[4]	0: No connection between GIx[4]/GIx[3] to GOx[4] 1: Allow GIx[4]/GIx[3] to drive GOx[4]
GDI_x_IN[5]	0: No connection between GIx[5]/GIx[4] to GOx[5] 1: Allow GIx[5]/GIx[4] to drive GOx[5]
GDI_x_IN[6]	0: No connection between GIx[6]/GIx[5] to GOx[6] 1: Allow GIx[6]/GIx[5] to drive GOx[6]
GDI_x_IN[7]	0: No connection between GIx[7]/GIx[6] to GOx[7] 1: Allow GIx[7]/GIx[6] to drive GOx[7]

For additional information, refer to the GDI_O_IN register on page 245 and the GDI_E_IN register on page 246.

Bits 7 to 0: GDIxICRx. Using the configuration bits in the GDI_x_IN_CR registers, a global input net may be configured to drive its corresponding *next* global output net. For example,

$$GIE[7] \rightarrow GOE[0]$$

Therefore, it is possible to drive 2 global nets with same data source, or shift data to other global nets (see Section 13.2.2). For example,

$$GIE[7] \rightarrow GOE[0]$$
$$GIE[7] \rightarrow GOE[7]$$
$$GIO[0] \rightarrow GOO[1] \rightarrow GIO[2]$$

There are a total of 16 bits that control the data source of global inputs to drive global outputs. These bits are in the GDI_x_IN_CR registers. Table 13-4 enumerates the meaning of each bit position in either of the GDI_O_IN_CR or GDI_E_IN_CR registers.

Table 13-4. GDI_x_IN_CR Register

GDI_xICR[0]	0: Data source is GIx[0] 1: Data source is GIx[7]
GDI_xICR[1]	0: Data source is GIx[1] 1: Data source is GIx[0]
GDI_xICR[2]	0: Data source is GIx[2] 1: Data source is GIx[1]
GDI_xICR[3]	0: Data source is GIx[3] 1: Data source is GIx[2]
GDI_xICR[4]	0: Data source is GIx[4] 1: Data source is GIx[3]
GDI_xICR[5]	0: Data source is GIx[5] 1: Data source is GIx[4]
GDI_xICR[6]	0: Data source is GIx[6] 1: Data source is GIx[5]
GDI_xICR[7]	0: Data source is GIx[7] 1: Data source is GIx[6]

For additional information, refer to the GDI_O_IN_CR register on page 231 and the GDI_E_IN_CR register on page 232.

13.2.2 GDI_x_OU/GDI_x_OU_CR Registers

GDI_x_OU										
Add.	**Name**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**Access**
1,D2h	GDI_O_OU	GOOUTIN7	GOOUTIN6	GOOUTIN5	GOOUTIN4	GOOUTIN3	GOOUTIN2	GOOUTIN1	GOOUTIN0	RW : 00
1,D3h	GDI_E_OU	GOEUTIN7	GOEUTIN6	GOEUTIN5	GOEUTIN4	GOEUTIN3	GOEUTIN2	GOEUTIN1	GOEUTIN0	RW : 00
GDI_x_OU_CR										
Add.	**Name**	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**	**Access**
1,A2h	GDI_O_OU_CR	GDIOOCR7	GDIOOCR6	GDIOOCR5	GDIOOCR4	GDIOOCR3	GDIOOCR2	GDIOOCR1	GDIOOCR0	RW : 00
1,A3h	GDI_E_OU_CR	GDIEOCR7	GDIEOCR6	GDIEOCR5	GDIEOCR4	GDIEOCR3	GDIEOCR2	GDIEOCR1	GDIEOCR0	RW : 00

The Global Digital Interconnect Odd and Even Output Registers (GDI_x_OU/GDI_x_OU_CR) are used to configure a global output to drive a global input.

The PSoC device has a configurable Global Digital Interconnect (GDI). Note that the GDI_x_IN and GDI_x_OU registers should never have the same bits connected. This would result in multiple drivers of one bus.

Bits 7 to 0: GOxUTINx. Using the configuration bits in the GDI_x_OU registers, a global output net may be configured to drive its corresponding global input. For example,

$$GOE[7] \rightarrow GIE[7]$$

The configurability of the GDI does not allow odd and even nets or nets with different indexes to be connected (however from N to N+1 is allowed and decided by GDI_x_OU_CR). The following are examples of connections that are not possible in the PSoC devices.

$$GOE[7] \not\rightarrow GIO[7]$$
$$GOE[0] \not\rightarrow GIE[7]$$
$$GOE[0] \rightarrow GIE[1]$$

There are a total of 16 bits that control the ability of global outputs to drive global inputs. These bits are in the GDI_x_OU registers. Table 13-5 enumerates the meaning of each bit position in either of the GDI_O_OU or GDI_E_OU registers.

Table 13-5. GDI_x_OU Register

GDI_x_OU[0]	0: No connection between GOx[0]/GOx[7] to GIx[0] 1: Allow GOx[0]/GOx[7] to drive GIx[0]
GDI_x_OU[1]	0: No connection between GOx[1]/GOx[0] to GIx[1] 1: Allow GOx[1]/GOx[0] to drive GIx[1]
GDI_x_OU[2]	0: No connection between GOx[2]/GOx[1] to GIx[2] 1: Allow GOx[2]/GOx[1] to drive GIx[2]
GDI_x_OU[3]	0: No connection between GOx[3]/GOx[2] to GIx[3] 1: Allow GOx[3]/GOx[2] to drive GIx[3]
GDI_x_OU[4]	0: No connection between GOx[4]/GOx[3] to GIx[4] 1: Allow GOx[4]/GOx[3] to drive GIx[4]
GDI_x_OU[5]	0: No connection between GOx[0]/GOx[4] to GIx[5] 1: Allow GOx[5]/GOx[4] to drive GIx[5]
GDI_x_OU[6]	0: No connection between GOx[6]/GOx[5] to GIx[6] 1: Allow GOx[6]/GOx[5] to drive GIx[6]
GDI_x_OU[7]	0: No connection between GOx[7]/GOx[6] to GIx[7] 1: Allow GOx[7]/GOx[6] to drive GIx[7]

For additional information, refer to the GDI_O_OU register on page 247 and the GDI_E_OU register on page 248.

Bits 7 to 0: GDIxOCRx. Using the configuration bits in the GDI_x_OU_CR registers, a global output net may be configured to drive its corresponding *next* global input. For example,

$$GOE[7] \rightarrow GIE[0]$$

Therefore, it is possible to drive 2 global nets with same data source, or shift data to other global nets (see Section 13.2.1). For example,

$$GOE[4] \rightarrow GIE[4]$$
$$GOE[4] \rightarrow GIE[5]$$
$$GOO[3] \rightarrow GIO[4] \rightarrow GOO[5]$$

There are a total of 16 bits that control the ability of global outputs to drive global inputs. These bits are in the GDI_x_OU_CR registers. Table 13-6 enumerates the meaning of each bit position in either of the GDI_O_OU_CR or GDI_E_OU_CR registers.

Table 13-6. GDI_x_OU_CR Register

GDIxOCR[0]	0: Data source is GOx[0] 1: Data source is GOx[7]
GDIxOCR[1]	0: Data source is GOx[1] 1: Data source is GOx[0]
GDIxOCR[2]	0: Data source is GOx[2] 1: Data source is GOx[1]
GDIxOCR[3]	0: Data source is GOx[3] 1: Data source is GOx[2]
GDIxOCR[4]	0: Data source is GOx[4] 1: Data source is GOx[3]
GDIxOCR[5]	0: Data source is GOx[5] 1: Data source is GOx[4]
GDIxOCR[6]	0: Data source is GOx[6] 1: Data source is GOx[5]
GDIxOCR[7]	0: Data source is GOx[7] 1: Data source is GOx[6]

For additional information, refer to the GDI_O_OU_CR register on page 233 and the GDI_E_OU_CR register on page 234.

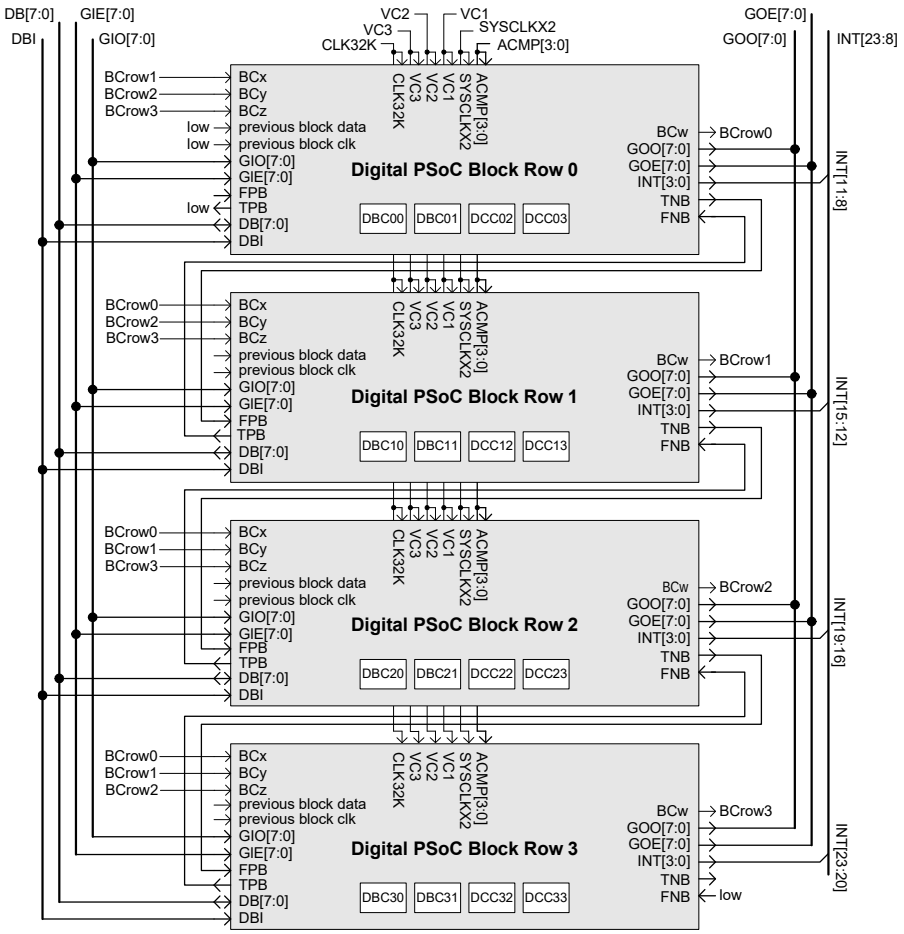
14. Array Digital Interconnect (ADI)

This chapter presents the Array Digital Interconnect (ADI). The digital PSoC array uses a scalable architecture that is designed to support from one to four digital PSoC rows, as defined in the Row Digital Interconnect (RDI) chapter on page 280. The digital PSoC array does not have any configurable interconnect; therefore, there are no associated registers in this chapter.

14.1 Architectural Description

The Array Digital Interconnect (ADI) is shown in Figure 14-1. The array structure varies depending on the number of digital rows your PSoC device has (see the table titled "PSoC Device Characteristics" on page 268). The ADI is not configurable; therefore, the information in this chapter is provided to improve the reader's understanding of the structure.

Figure 14-1. Digital PSoC Block Array Structure



In Figure 14-1, the detailed view of a Digital PSoC block row has been replaced by a box labeled digital PSoC block row x. The rest of this figure illustrates how all rows are connected to the same globals, clocks, and so on. The figure also illustrates how the broadcast clock nets (BCrowx) are connected between rows.

The different PSoC CY8C2xxxx devices (except for the CY8C25122 and CY8C26xxx PSoC devices) have varying numbers of digital PSoC blocks in the digital array. These blocks are arranged into rows and the ADI provides a regular interconnect architecture between the Global Digital Interconnect (GDI) and the Row Digital Interconnect (RDI), regardless of the number of rows available in a particular device. The most important aspect of the ADI and the digital PSoC rows is that all digital PSoC rows have the same connections to global inputs and outputs. The connections that make a row's position unique are explained as follows.

- **Register Address**: Rows and the blocks within them need to have unique register addresses.
- **Interrupt Priority**: Each digital PSoC block has its own interrupt priority and vector. A row's position in the array determines the relative priority of the digital PSoC blocks within the row. The lower the row number, the higher the interrupt priority, and the lower the interrupt vector address.
- **Broadcast**: Each digital PSoC row has an internal ***broadcast net*** that may be either driven internally, by one of the four digital PSoC blocks, or driven externally. In the case where the broadcast net is driven externally, the source may be any one of the other rows in the array. Therefore, depending on the row's position in the array, it will have different options for driving its broadcast net.
- **Chaining Position**: Rows in the array form a string of digital blocks equal in length to the number of rows multiplied by four. The first block in the first row and the last block in the last row are not connected; therefore, the array does not form a loop. The first row in the array has its previous ***chaining*** inputs tied low. If there is a second row in the array, the next chaining outputs are connected to the next row. For the last row in the array, the next inputs are tied low.

15. Row Digital Interconnect (RDI)

This chapter explains the Row Digital Interconnect (RDI) and its associated registers. This chapter discusses a single digital PSoC block row. It does not discuss the functions, inputs, or outputs for individual digital PSoC blocks; nor does it cover specific instances of multiple rows in a single part. Therefore, the information contained here is valid for 4, 2, and 1 row configurations. Information about individual digital PSoC blocks is covered in the Digital Blocks chapter on page 288. For a complete table of the RDI registers, refer to the "Summary Table of the Digital Registers" on page 269. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

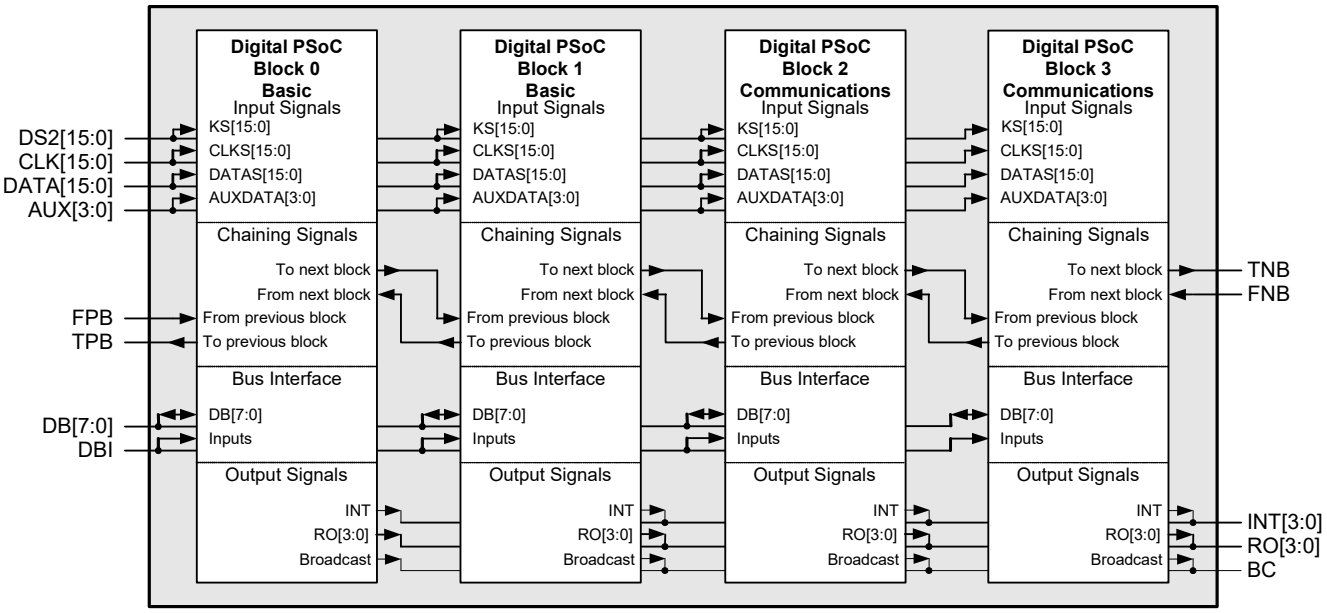
15.1 Architectural Description

Many signals pass through the digital PSoC block row on their way to or from individual ***digital blocks***. However, only a small number of signals pass though configurable circuits on their way to and from digital blocks. The configurable circuits allow for greater flexibility in the connections between digital blocks and global buses. What follows is a discussion of the signals that are configurable by way of the registers listed in the "Register Definitions" on page 282.

In Figure 15-1, within a digital PSoC block row, there are four digital PSoC Blocks. The first two blocks are of the type basic (DBC). The second two are of the type communication (DCC). This figure shows the connections between digital blocks within a row. Only the signals that pass outside the gray background box in Figure 15-1 are shown at the next level of hierarchy in Figure 15-2.

In Figure 15-2, the detailed view shown in Figure 15-1 of the four PSoC block grouping, has been replaced by the box in the center of the figure labeled "4 PSoC Block Grouping." The rest of the configurable nature of the Row Inputs (RI), Row Outputs (RO), and Broadcast clock net (BC) is shown for the next level of hierarchy.

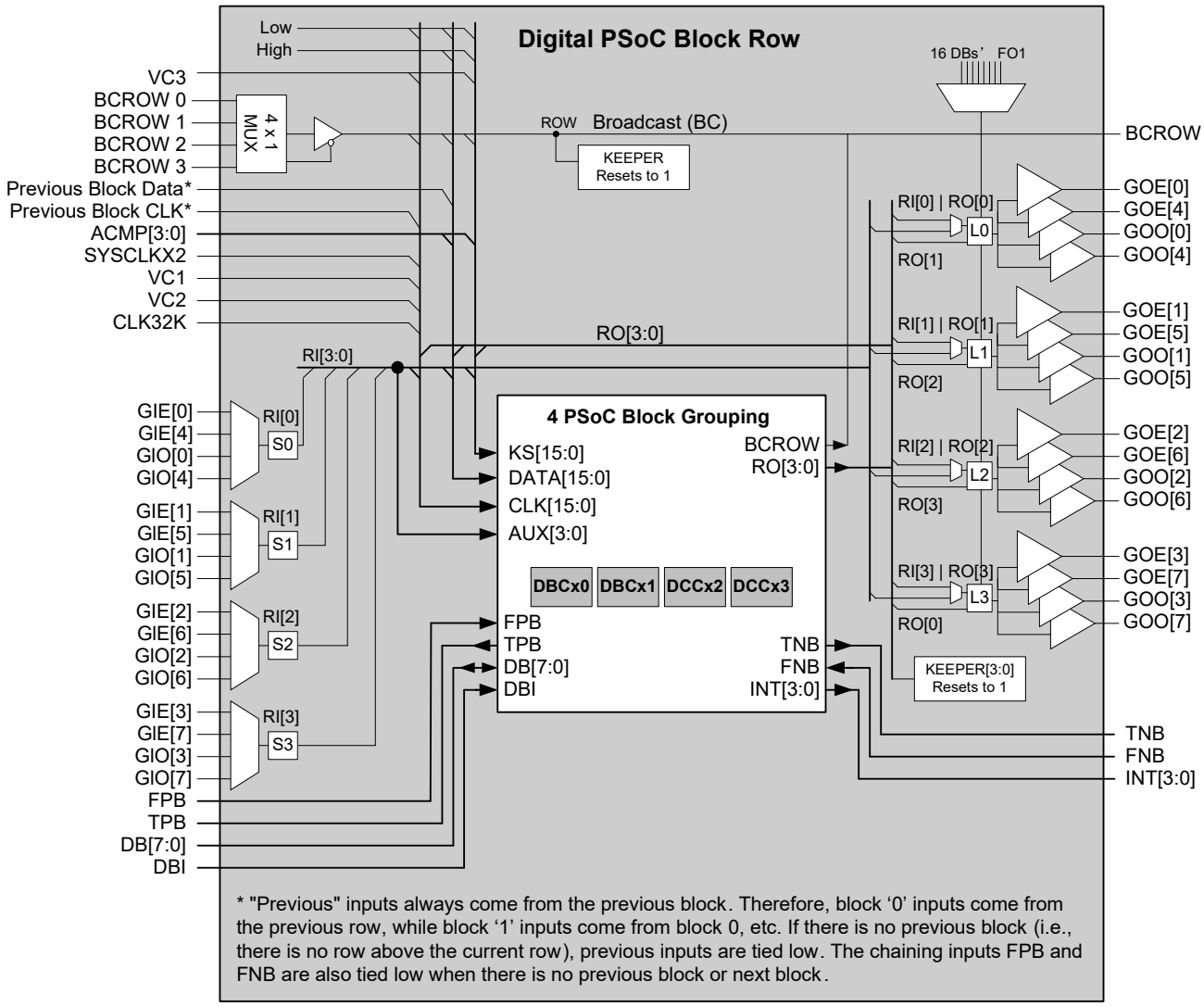
Figure 15-1. Detailed View of Four PSoC Block Grouping



As shown in Figure 15-2, there is a ***keeper*** connected to the row ***broadcast net*** and each of the row outputs. The keeper sets the value of these nets to '1' on system reset and holds the value of the net should it become un-driven.

Notice on the left side of Figure 15-2 that global inputs (GIE[n] and GIO[n]) are inputs to 4-to-1 multiplexers. The output of these muxes are Row Inputs (RI[x]). Because there are four 4-to-1 muxes, each with a unique set of inputs, a row has access to every global input line in a PSoC device.

Figure 15-2. Digital PSoC Block Row Structure



15.2 PSoC Device Distinctions

For Silicon Revision A of the CY8C27x43 PSoC device, only digital blocks 01, 02, 11, and 12 are valid in the DEC_CR1 register. See the DEC_CR1 register on page 179 for more information.

15.3 Register Definitions

The following registers are associated with the Row Digital Interconnect (RDI) and are listed in address order. Each register description has an associated register table showing the bit structure for that register. For a complete table of RDI registers, refer to the “Summary Table of the Digital Registers” on page 269.

Depending on how many digital rows your PSoC device has (see the Rows column in the register tables below and refer to the table titled “PSoC Device Characteristics” on page 268), only certain bits are accessible to be read or written. The bits that are grayed out throughout this manual are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of ‘0’.

The only configurable inputs to a digital PSoC block row are the Global Input Even and Global Input Odd 8-bit buses. The only configurable outputs from the digital PSoC block row are the Global Output Even and Global Output Odd 8-bit buses. Figure 15-2 illustrates the relationships between global signals and row signals.

15.3.1 RDIxRI Register

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,B0h	RDI0RI	4, 3, 2, 1	RI3[1:0]		RI2[1:0]		RI1[1:0]		RI0[1:0]		RW : 00
x,B8h	RDI1RI	4, 3, 2	RI3[1:0]		RI2[1:0]		RI1[1:0]		RI0[1:0]		RW : 00

LEGEND
x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Row Digital Interconnect Row Input Register (RDIxRI) is used to control the input mux that determines which global inputs will drive the row inputs.

The RDIxRI Register and the RDIxSYN Register are the only two registers that affect digital PSoC row input signals. All other registers are related to output signal configuration.

The RDIxRI register has select bits that are used to control four muxes, where “x” denotes a place holder for the row index. Table 15-1 lists the meaning for each mux’s four possible settings.

Bits 7 and 6: RI3[1:0]. These bits control the input mux for row 3.

Bits 5 and 4: RI2[1:0]. These bits control the input mux for row 2.

Bits 3 and 2: RI1[1:0]. These bits control the input mux for row 1.

Bits 1 and 0: RI0[1:0]. These bits control the input mux for row 0.

Table 15-1. RDIxRI Register

RI3[1:0]	0h: GIE[3] 1h: GIE[7] 2h: GIO[3] 3h: GIO[7]
RI2[1:0]	0h: GIE[2] 1h: GIE[6] 2h: GIO[2] 3h: GIO[6]
RI1[1:0]	0h: GIE[1] 1h: GIE[5] 2h: GIO[1] 3h: GIO[5]
RI0[1:0]	0h: GIE[0] 1h: GIE[4] 2h: GIO[0] 3h: GIO[4]

For additional information, refer to the RDIxRI register on page 140.

15.3.2 RDIxSYN Register

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,B1h	RDI0SYN	4, 3, 2, 1					RI3SYN	RI2SYN	RI1SYN	RI0SYN	RW : 00
x,B9h	RDI1SYN	4, 3, 2					RI3SYN	RI2SYN	RI1SYN	RI0SYN	RW : 00

LEGEND
x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Row Digital Interconnect Synchronization Register (RDIxSYN) is used to control the input synchronization.

The RDIxRI Register and the RDIxSYN Register are the only two registers that affect digital PSoC row input signals. All other registers are related to output signal configuration.

By default, each row input is double synchronized to the SYSCLK (system clock), which runs at 24 MHz unless external clocking mode is enabled. However, a user may choose to disable this synchronization by setting the appropriate RIxSYN bit in the RDIxSYN register. Table 15-2 lists the bit meanings for each implemented bit of the RDIxSYN register.

Bit 3: RI3SYN. This bit controls the input synchronization for row 3.

Bit 2: RI2SYN. This bit controls the input synchronization for row 2.

Bit 1: RI1SYN. This bit controls the input synchronization for row 1.

Bit 0: RI0SYN. This bit controls the input synchronization for row 0.

Table 15-2. RDIxSYN Register

RI3SYN	0: Row input 3 is synchronized to SYSCLK 1: Row input 3 is passed without synchronization
RI2SYN	0: Row input 2 is synchronized to SYSCLK 1: Row input 2 is passed without synchronization
RI1SYN	0: Row input 1 is synchronized to SYSCLK 1: Row input 1 is passed without synchronization
RI0SYN	0: Row input 0 is synchronized to SYSCLK 1: Row input 0 is passed without synchronization

For additional information, refer to the RDIxSYN register on page 141.

15.3.3 RDIxIS Register

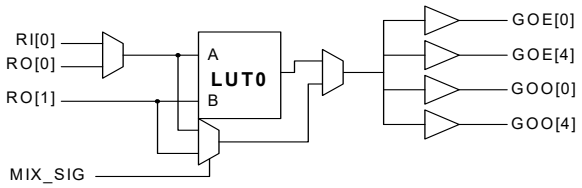
Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,B2h	RDI0IS	4, 3, 2, 1			BCSEL[1:0]		IS3	IS2	IS1	IS0	RW : 00
x,BAh	RDI1IS	4, 3, 2			BCSEL[1:0]		IS3	IS2	IS1	IS0	RW : 00

LEGEND
x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Row Digital Interconnect Input Select Register (RDIxIS) is used to configure the A inputs to the digital row LUTS and select a broadcast driver from another row if present.

Each LUT has two inputs, where one of the inputs is configurable (Input A) and the other input (Input B) is fixed to a row output. Figure 15-3 presents an example of LUT configuration

Figure 15-3. Example of LUT0 Configuration.



These bits are the Input B for the ***look-up table (LUT)***. The configurable LUT input (Input A) chooses between a single row output and a single row input. Table 15-3 lists the options for each LUT in a row. The bits are labeled IS, meaning Input Select. The LUT’s fixed input is always the RO[LUT number + 1], such as LUT0’s fixed input is RO[1], LUT1’s fixed input is RO[2], …, and LUT3’s fixed input is RO[0].

Bits 5 and 4: BCSEL[1:0]. These bits are used to determine which digital PSoC row will drive the local broadcast net. If a row number is selected that does not exist, the broadcast net is driven to a logic 1 value. If any digital PSoC block in the local row has its DxCxFN[BCEN] bit set, the broadcast select is disabled. See the “DxCxxFN Registers” on page 314.

Bit 3: IS3. This bit controls the ‘A’ input of LUT 3.

Bit 2: IS2. This bit controls the ‘A’ input of LUT 2.

Bit 1: IS1. This bit controls the ‘A’ input of LUT 1.

Bit 0: IS0. This bit controls the ‘A’ input of LUT 0.

Table 15-3. RDIxIS Register Bits

BCSEL[1:0]	0: Row broadcast net driven by row 0 broadcast net.* 1: Row broadcast net driven by row 0 broadcast net.* 2: Row broadcast net driven by row 0 broadcast net.* 3: Row broadcast net driven by row 0 broadcast net.*
IS3	0: The ‘A’ input of LUT3 is RO[3] 1: The ‘A’ input of LUT3 is RI[3]
IS2	0: The ‘A’ input of LUT2 is RO[2] 1: The ‘A’ input of LUT2 is RI[2]
IS1	0:The ‘A’ input of LUT1 is RO[1] 1: The ‘A’ input of LUT1 is RI[1]
IS0	0: The ‘A’ input of LUT0 is RO[0] 1: The ‘A’ input of LUT0 is RI[0]
* When the BCSEL value is equal to the row number, the tri-state buffer that drives the row broadcast net from the input select mux is disabled, so that one of the row’s blocks may drive the local row broadcast net. * Refer to Figure 15-2. * If the row is not present in the part, the selection provides a logic 1 value.	

For additional information, refer to the RDIxIS register on page 142.

15.3.4 RDIxLTx Registers

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,B3h	RDI0LT0	4, 3, 2, 1	LUT1[3:0]				LUT0[3:0]				RW : 00
x,B4h	RDI0LT1	4, 3, 2, 1	LUT3[3:0]				LUT2[3:0]				RW : 00
x,BBh	RDI1LT0	4, 3, 2	LUT1[3:0]				LUT0[3:0]				RW : 00
x,BCh	RDI1LT1	4, 3, 2	LUT3[3:0]				LUT2[3:0]				RW : 00

LEGEND
x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Row Digital Interconnect Logic Table Register 0 and 1 (RDIxLT0 and RDIxLT1) are used to select the logic function of the digital row LUTS.

The outputs from a digital PSoC row are a bit more complicated than the inputs. Figure 15-2 on page 281 illustrates the output circuitry in a digital PSoC row. In the figure, find a block labeled Lx. This block represents a 2-input look-up table (LUT). The LUT allows the user to specify any one of 16 logic functions that should be applied to the two inputs.

The output of the logic function will determine the value that may be driven on to the Global Output Even and Global Output Odd buses. Table 15-4 lists the relationship between a look-up table’s four configuration bits and the resulting logic function. Some users may find it easier to determine the proper configuration bits setting, by remembering that the configuration’s bits represent the output column of a two-input logic truth table. Table 15-4 lists seven examples of the relationship between the LUT’s output column for a truth table and the LUTx[3:0] configuration bits. Figure 15-3 on page 284 presents an example of LUT configuration.

Bits 7 to 4: LUTx[3:0]. These configuration bits are for a row output LUT.

Bits 3 to 0: LUTx[3:0]. These configuration bits are for a row output LUT.

Table 15-4. Example LUT Truth Tables

A	B	AND	OR	A+$\overline{B}$	A&$\overline{B}$	A	B	True
0	0	0	0	1	0	0	0	1
0	1	0	1	0	0	0	1	1
1	0	0	1	1	1	1	0	1
1	1	1	1	1	0	1	1	1
LUTx[3:0]		1h	7h	Bh	2h	3h	5h	Fh

Table 15-5. RDIxLTx Register

LUTx[3:0]	0h: 0000: FALSE 1h: 0001: A .AND. $\overline{B}$ 2h: 0010: A .AND. $\overline{B}$ 3h: 0011: $\overline{A}$ 4h: 0100: $\overline{A}$ .AND. B 5h: 0101: B 6h: 0110: A .XOR. B 7h: 0111: A .OR. B 8h: 1000: A .NOR. B 9h: 1001: $\overline{A}$ .XNOR. B Ah: 1010: $\overline{B}$ Bh: 1011: $\overline{A}$ .OR. $\overline{B}$ Ch: 1100: $\overline{A}$ Dh: 1101: $\overline{A}$ .OR. B Eh: 1110: A. NAND. B Fh: 1111: TRUE

For additional information, refer to the RDIxLT0 register on page 143 and the RDIxLT1 register on page 144.

15.3.5 RDIxROx Registers

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,B5h	RDI0RO0	4, 3, 2, 1	GOO5EN	GOO1EN	GOE5EN	GOE1EN	GOO4EN	GOO0EN	GOE4EN	GOE0EN	RW : 00
x,B6h	RDI0RO1	4, 3, 2, 1	GOO7EN	GOO3EN	GOE7EN	GOE3EN	GOO6EN	GOO2EN	GOE6EN	GOE2EN	RW : 00
x,BDh	RDI1RO0	4, 3, 2	GOO5EN	GOO1EN	GOE5EN	GOE1EN	GOO4EN	GOO0EN	GOE4EN	GOE0EN	RW : 00
x,BEh	RDI1RO1	4, 3, 2	GOO7EN	GOO3EN	GOE7EN	GOE3EN	GOO6EN	GOO2EN	GOE6EN	GOE2EN	RW : 00

LEGEND
x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Row Digital Interconnect Row Output Register 0 and 1 (RDIxRO0 and RDIxRO1) are used to select the global nets that the row outputs drive.

The final configuration bits for outputs from digital PSoC rows are in the two RDIxROx registers. These registers hold the 16 bits that can individually enable the tri-state buffers that connect to all eight of the Global Output Even lines and all eight of the Global Output Odd lines to the row LUTs.

The input to these tri-state drivers are the outputs of the row’s LUTs, as shown in Figure 15-2. This means that any row can drive any global output. Keep in mind that tri-state drivers are being used to drive the global output lines; therefore, it is possible for a part, with more than one digital PSoC row, to have multiple drivers on a single global output line. It is the user’s responsibility to ensure that the part is not configured with multiple drivers on any of the global output lines. Figure 15-3 on page 284 presents an example LUT configuration.

15.3.5.1 RDIxRO0 Register

Bits 7 to 4: GOxxEN. These configuration bits enable the tri-state buffers that connect to the global output even lines for LUT 1.

Bits 3 to 0: GOxxEN. These configuration bits enable the tri-state buffers that connect to the global output even lines for LUT 0.

For additional information, refer to the RDIxRO0 register on page 145.

15.3.5.2 RDIxRO1 Register

Bits 7 to 4: GOxxEN. These configuration bits enable the tri-state buffers that connect to the global output even lines for LUT 3.

Bits 3 to 0: GOxxEN. These configuration bits enable the tri-state buffers that connect to the global output even lines for LUT 2.

For additional information, refer to the RDIxRO1 register on page 146.

15.3.6 RDIxDSM Register

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,B7h	RDI0DSM	4, 3, 2, 1		AVS_SEL[2:0]			AVG_EN[3:0]				RW : 00
x,BFh	RDI1DSM	4, 3, 2		AVS_SEL[2:0]			AVG_EN[3:0]				RW : 00

LEGEND
x An “x” before the comma in the address field indicates that the register exists in both register banks.

The Row Digital Interconnect Delta Sigma Modulator Register (RDIxDSM) is used to select the Delta Sigma Modulator function on the row outputs.

Refer to Figure 15-2, Figure 15-3.

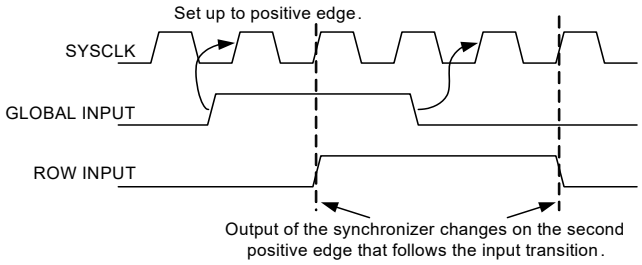
Bits 6 to 4: AVS_SEL[2:0]. These configuration bits select 1 from 8 dig-blocks's FO1 as average-control signal.

Bits 3 to 0: AVG_EN[3:0]. These configuration bits enable average function on corresponding RO channel.

For additional information, refer to the RDIxDSM register on page 147.

15.4 Timing Diagram

Figure 15-4. Optional Row Input Synchronization to SYSCLK



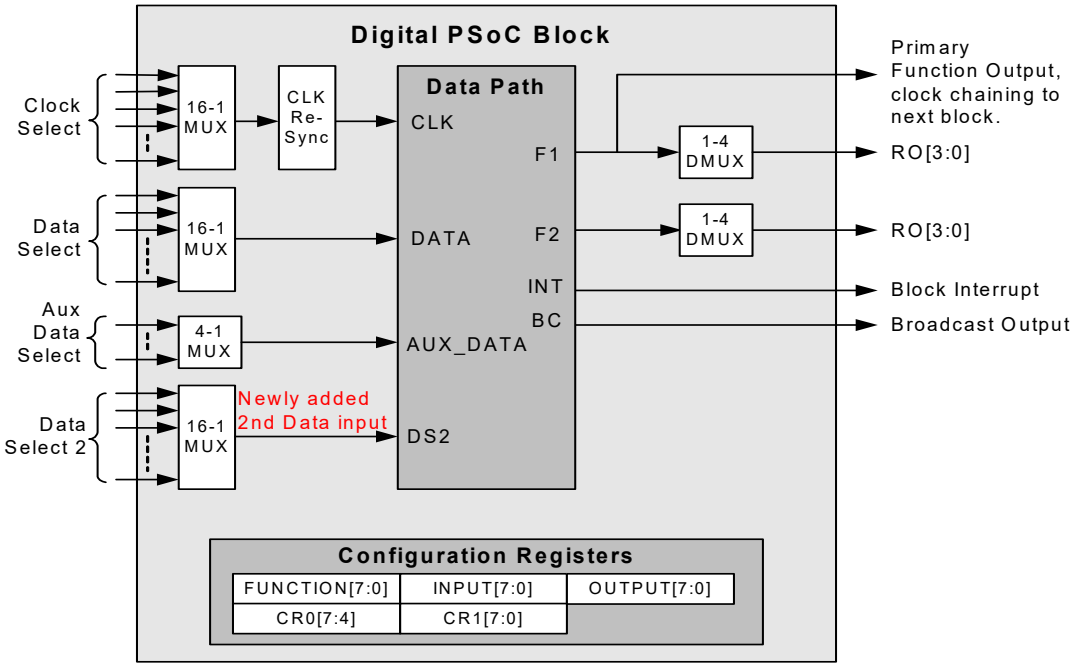
16. Digital Blocks

This chapter covers the configuration and use of the digital PSoC blocks and their associated registers. For a complete table of the Digital PSoC Block registers, refer to the "Summary Table of the Digital Registers" on page 269. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

16.1 Architectural Description

At the top level, the main components of the digital block are the data path, input multiplexers (muxes), output de-muxes, configuration registers, and chaining signals (see Figure 16-1).

Figure 16-1. Digital Blocks Top-Level Block Diagram



All digital PSoC blocks may be configured to perform any one of five basic functions: timer, counter, ***pulse width modulator (PWM)***, pseudo random sequence (PRS), or ***cyclic redundancy check (CRC)***. These functions may be used by configuring an individual PSoC block or chaining several PSoC blocks together to form functions that are greater than 8 bits. Digital communications PSoC blocks have two additional functions: master or slave SPI and a full duplex ***UART***.

Each digital PSoC block's function is independent of all other PSoC blocks. Up to seven registers are used to determine the function and state of a digital PSoC block. These registers are discussed in the Register Definitions section. Digital PSoC block function registers end with FN. The individual bit settings for a block's function register are listed in Table 16-23 on page 314. The input registers end with IN and its bit meanings are listed in Table 16-25 on page 315. Finally, the block's outputs are controlled by the output register which ends in OU.

Each digital PSoC block also has three data registers (DR0, DR1, and DR2) and one control register (CR0). The bit meanings for these registers are heavily function dependent and are discussed with each function's description.

In addition to seven registers that control the digital PSoC block's function and state, a separate interrupt mask bit is available for each digital PSoC block. Each digital PSoC block has a unique interrupt vector; therefore, it can have its own interrupt service routine.

16.1.1 Input Multiplexers

Typically, each function has a clock and a data input that may be selected from a variety of sources. Each of these inputs is selected with a 16-to-1 input mux.

In addition, there is a 4-to-1 mux which provides an auxiliary input for the SPI Slave function that requires three inputs: Clock, Data, and SS_ (unless the SS_ is forced active with the Aux I/O Enable bit). The inputs to this mux are intended to be a selection of the row inputs.

16.1.2 Input Clock Resynchronization

Digital blocks allow a clock selection from one of 16 sources. Possible sources are the system clocks (VC1, VC2, VC3, SYSCLK, and SYSCLKX2), row inputs, and other digital block outputs. To manage clock ***skew*** and ensure that the interfaces between blocks meet timing in all cases, all digital block input clocks must be resynchronized to either SYSCLK or SYSCLKX2, which are the source clocks for all the PSoC device clocking. Also, SYSCLK or SYSCLKX2 may be used directly. The AUXCLK bits in the DxCxxOU register are used to specify the input synchronization. The following rules apply to the use of input clock resynchronization.

1. If the clock input is derived (for example, divided down) from SYSCLK, resynchronize to SYSCLK at the digital block. Most the PSoC device clocks are in this category. For example, VC1 and VC2, and the output of other blocks clocked by VC1 and VC2, or SYSCLK (for setting see Table 16-1).
2. If the clock input is derived from SYSCLKX2, resynchronize to SYSCLKX2. For example, VC3 clocked by SYSCLKX2 or other digital blocks clocked by SYSCLKX2 (for setting, see Table 16-1).
3. Choose direct SYSCLK for clocking directly off of SYSCLK (for setting, see Table 16-1).
4. Choose direct SYSCLKX2 (select SYSCLKX2 in the Clock Input field of the DxCxxIN register) for clocking directly off of SYSCLKX2.
5. Bypass Synchronization. This should be a very rare selection; because if clocks are not synchronized, they may fail setup to CPU read and write commands. However, it is possible for an external pin to asynchronously clock a digital block (for example, if you want to synchronize CPU interaction through interrupts or other techniques, by setting 00 in AUXCLK). This setting is also required for blocks to remain active while in sleep.

The following note enumerates configurations that are not allowed, although the hardware does not prevent them. The clock dividers (VC1, VC2, and VC3) may not be configured in such a way as to create an output clock that is equal to SYSCLK or SYSCLKX2.

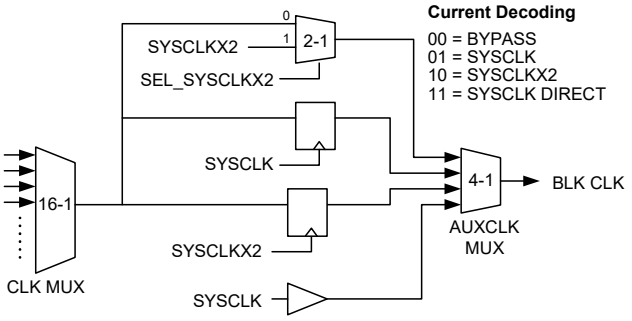
Note If the input clock frequency matches the frequency of the clock used for synchronization, the block will never receive a clock (see Figure 16-2). As for SYSCLK, this can happen in the following cases:

- Using VC1 configured as divide by one.
- Using VC2 with VC1 and VC2 both configured as divide by one.
- Using VC3 divided by one with a source of VC1 divided by one.
- Using VC3 divided by one with a source of VC2, where both VC1 and VC2 are divided by one.
- Using VC3 divided by one with SYSCLK source.

In all of these cases, select SYSCLK directly in the block. Similarly, if VC3 is configured as divide by one with a source of SYSCLKX2, then select SYSCLKX2 to clock the block directly instead of VC3.

The clock resynchronizer is illustrated in Figure 16-2.

Figure 16-2. Input Clock Resynchronization



In sleep, SYSCLK is powered down, and therefore, input synchronization is not available.

Table 16-1. AUXCLK Bit Selections

Code	Description	Usage
00	Bypass	Use this setting only when SYSCLKX2 (48 MHz) is selected. Other than this case, asynchronous clock inputs are not recommended. This setting is also required for blocks to remain active while in sleep.
01	Resynchronize to SYSCLK (24 MHz)	Use this setting for any SYSCLK-based clock. VC1, VC2, VC3 driven by SYSCLK, digital blocks with SYSCLK-based source clocks, broadcast bus with source based on SYSCLK, row input and row outputs with source based on SYSCLK.
10	Resynchronize to SYSCLKX2 (48 MHz)	Use this setting for any SYSCLKX2-based clock. VC3 driven by SYSCLKX2, digital blocks with SYSCLKX2-based source clocks, broadcast bus with source based on SYSCLKX2, row input and row outputs with source based. on SYSCLKX2.
11	SYSCLK Direct	Use this setting to clock the block directly using SYSCLK. Note that this setting is not strictly related to clock resynchronization, but because SYSCLK cannot resynchronize itself, it allows a direct skew controlled SYSCLK source.

16.1.2.1 Clock Resynchronization Summary

- Digital PSoC blocks have extremely flexible clocking configurations. To maintain reliable timing, input clocks must be resynchronized.
- The master clock for any clock in the system is either SYSCLK or SYSCLKX2. Determine the master clock for a given input clock and resynchronize to that clock.
- Do not use divide by 1 clocks derived from SYSCLK and SYSCLKX2. Use the direct SYSCLK or SYSCLKX2 clocking option available at the block.

16.1.3 Output De-Multiplexers

Most functions have two outputs: a primary and an auxiliary output, the meaning of which are function dependent. Each of these outputs may be driven onto the row output bus. Each de-mux is implemented with four tri-state drivers. There are two bits in the output register to select one of the four tri-state drivers and an additional bit to enable the selected driver.

16.1.4 Block Chaining Signals

Each digital block has the capability to be chained and to create functions with bit widths greater than eight. There are signals to propagate information, such as Compare, Carry, Enable, Capture and Gate, from one block to the next to implement higher precision functions. The selection made in the function register determines which signals are appropriate for the desired function. User Modules that have been designed to implement digital functions, with greater than 8-bit width, will automatically make the proper selections of the chaining signals, to ensure the correct information flow between blocks.

16.1.5 Input Data Synchronization

Any asynchronous input derived from an external source, such as a GPIO pin input, must be resynchronized through the row input before use into any digital block clock or data input. This is the default mode of operation (resynchronization is on).

16.1.6 Timer Function

A timer consists of a period register, a ***synchronous*** down counter, and a capture/compare register, all of which are byte wide. When the timer is disabled and a period value is written into DR1, the period value is also loaded into DR0. When the timer is enabled, the counter counts down until positive terminal count (a count of 00h) is reached. On the next clock edge, the period is reloaded and, on subsequent clocks, counting continues. The terminal count signal is the primary function output. (Refer to the timing diagram for this function on page 317.) This can be configured as a full or half clock cycle.

This function also supports multi-shot mode. When the multi-shot register is set to non-zero, the function is in multi-shot mode. For example, if the multi-shot register is set to 4'h1, the function goes to disable after it reaches the first 00 value in DR0. If the multi-shot register is set to 4'h2, when the function reaches the first 00, DR0 is reloaded and runs again and goes to disable after the second 00 in DR0 register. The multi-shot supports up to the MAX number of 15.

Hardware capture occurs on the positive edge of the data input. This event transfers the current count from DR0 to DR2. The captured value may then be read directly from DR2. A software capture function is equivalent to a hardware capture. A CPU read of DR0, with the timer enabled, triggers the same capture mechanism. The hardware and software capture mechanisms are OR'ed in the capture circuitry. Because the capture circuitry is positive edge sensitive, during an interval where the hardware capture input is high, a software capture is masked and will not occur.

The timer also implements a compare function between DR0 and DR2. The compare signal is the auxiliary function output. A limitation, in regards to the compare function, is that the capture and compare function both use the same register (DR2). Therefore, if a capture event occurs, it will overwrite the compare value.

There is another mode, called NPS mode, supported above the compare output. When it is set, the compare output is delayed half clock cycle to be finished. It is used to achieve a higher resolution when 48MHz clock is used as block clock.

This function also supports KILL function. There are two KILL modes: KILL-Disable and KILL-Reload. In KILL-Disable mode, the function is disabled when KILL arrives. In KILL-Reload mode, the DR0 register and multi-shot counter register stays in reload state when KILL is high, and the function counts down when KILL is released. The function outputs are gated to zeros when KILL is asserted.

Mode bit 1 in the function register sets the compare type (DR0 <= DR2 or DR0 < DR2) and Mode bit 0 sets the interrupt type (Terminal Count or Compare). There are also two more bits used to control interrupts, located at bit 1 in CR0 and bit 0 in CR1.

Table 16-2. Timer Interrupt Source

Interrupt Source	Non Multi-shot Mode			Multi-shot Mode		
	KILL_INT (CR1[0])	Capture INT (CR0[1])	Compare True (FN[1])	KILL_INT (CR1[0])	Capture INT (CR0[1])	Compare True (FN[1])
KILL	1	*	*	1	*	*
Capture	0	1	*	0	1	*
Compare	0	0	1	0	0	1
TC	0	0	0			
Last-Shot				0	0	0

Timers may be chained in 8-bit lengths up to 32 bits.

Table 16-3. Timer Control Signals in Chained Block

Item	Configured in
Capture	LSB Block
KILL	LSB Block
Multi-shot Period	MSB Block
Clock	All chained Blocks
KILL Mode	All chained Blocks

16.1.6.1 Usability Exceptions

The following are usability exceptions for the Timer function:
1. Capture operation is not supported at 48 MHz.
2. DR2 is not writeable when the Timer is enabled.
3. CR1 is not writeable when the Timer is enabled.

16.1.6.2 Block Interrupt

The Timer block has a selection of three interrupt sources. Interrupt on Terminal Count (TC) and Interrupt on Compare may be selected in Mode bit 0 of the function register. The third interrupt source, Interrupt on Capture, may be selected with the Capture Interrupt bit in the control register.

- **Interrupt on Terminal Count**: The positive edge of terminal count (primary output) generates an interrupt for this block. The timing of the interrupt follows the TC pulse width setting in the control register.
- **Interrupt on Compare**: The positive edge of compare (auxiliary output) generates an interrupt for this block.
- **Interrupt on Capture**: Hardware or software capture generates an interrupt for this block. The interrupt occurs at the closing of the DR2 latch on capture.
- **Interrupt on KILL**: The interrupt occurs when KILL is asserted.

16.1.7 Counter Function

A Counter consists of a period register, a synchronous down counter, and a compare register. The Counter function is identical to the Timer function, with the following exceptions:

- The data input is a counter gate (enable), rather than a capture input. Counters do not implement synchronous capture. The DR0 register in a counter should not be read when it is enabled.
- The compare output is the primary output and the Terminal Count (TC) is the auxiliary output (opposite of the Timer).
- Terminal count output is full cycle only.

When the counter is disabled and a period value is written into DR1, the period value is also loaded into DR0. When the counter is enabled, the counter counts down until terminal count (a count of 00h) is reached. On the next clock edge, the period is reloaded and, on subsequent clocks, counting continues. (Refer to the timing diagram for this function on page 319.)

16.1.7.1 Counter Timing

This function also supports multi-shot mode. When the multi-shot register is set to non-zero, the function is in multi-shot mode. For example, if the multi-shot register is set to 4'h1, the function goes to disable after it reaches the first 00 value in DR0. If the multi-shot register is set to 4'h2, when the function reaches the first 00, DR0 reloads and runs again and goes to disable after the second 00 in DR0 register. The multi-shot supports up to the MAX number of 15.

The counter implements a compare function between DR0 and DR2. The Compare signal is the primary function output. Mode bit 1 sets the compare type (DR0 <= DR2 or DR0 < DR2) and Mode bit 0 sets the interrupt type (terminal count or compare). Note that in default if you write to DR2 in function running state, DR2 data changes immediately and then the compare output may change immediately after. A configure bit in CR0[1] can be used to delay the DR2 data changing until TC occurs (that is, at DR0 reloading). Therefore, you will not see unusual changes on compare out after they change the DR2 data.

There is another mode, called NPS mode, supported above the compare output. When it is set, the compare output is delayed half clock cycle to be finished. It is used to achieve a higher resolution when 48MHz clock is used as block clock.

This function also supports KILL function. There are 2 KILL modes: KILL-Disable and KILL-Reload. In KILL-Disable mode, the function is disabled when KILL is coming. In KILL-Reload mode, the DR0 register and the multi-shot counter register stay in reload state when KILL is high, and the func-

tion counts down when KILL is released. The function outputs are gated to zeros when KILL is asserted.

The data input functions as a gate to counter operation. The counter only counts and reloads when the data input is asserted (logic 1). When the data input is negated (logic 0), counting (including the period reload) is halted.

The Interrupt is controlled by two register bits. See the following table.

Table 16-4. Counter Interrupt Source

Interrupt Source	Non Multi-Shot Mode		Multi-Shot Mode	
	KILL_INT (CR1[0])	Compare True (FN[1])	KILL_INT (CR1[0])	Compare True (FN[1])
KILL	1	*	1	*
Compare	0	1	0	1
TC	0	0		
Last-Shot			0	0

Counters may be chained in 8-bit blocks up to 32 bits.

Table 16-5. Counter Control Signals in Chained Blocks

Item	Configured in
Gate	LSB Block
KILL	LSB Block
Multi-shot Period	MSB Block
Clock	All Chained Blocks
KILL Mode	All Chained Blocks

16.1.7.2 Usability Exceptions

The following are usability exceptions for the Counter function:

1. DR0 may only be read (to transfer DR0 data to DR2) when the block is disabled.
2. CR1 is not writeable when the Counter is enabled.

16.1.7.3 Block Interrupt

The Counter block has a selection of two interrupt sources. Interrupt on Terminal Count (TC) and Interrupt on Compare may be selected in Mode bit 0 of the function register.

- **Interrupt on Terminal Count**: The positive edge of terminal count (auxiliary output) generates an interrupt for this block. The timing of the interrupt follows the TC pulse width setting in the control register.
- **Interrupt on Compare**: The positive edge of compare (primary output) generates an interrupt for this block.
- **Interrupt on KILL**: The interrupt occurs when KILL is asserted.

16.1.8 Dead Band Function

The Dead Band function generates output signals on both the primary and auxiliary outputs of the block, see Figure 16-3. Each of these outputs is one ***phase*** of a two-phase, non-overlapping clock generated by this function. The two clock phases are never high at the same time and the period between the clock phases is known as the ***dead band***. The width of the dead band time is determined by the value in the period register. This dead band function can be driven with a PWM as an input clock or it can be clocked directly by toggling a bit in software using the Bit-Bang interface. If the clock source is a PWM, this will make a two output PWM with guaranteed non-overlapping outputs. An active asynchronous signal on the KILL data input disables both outputs immediately.

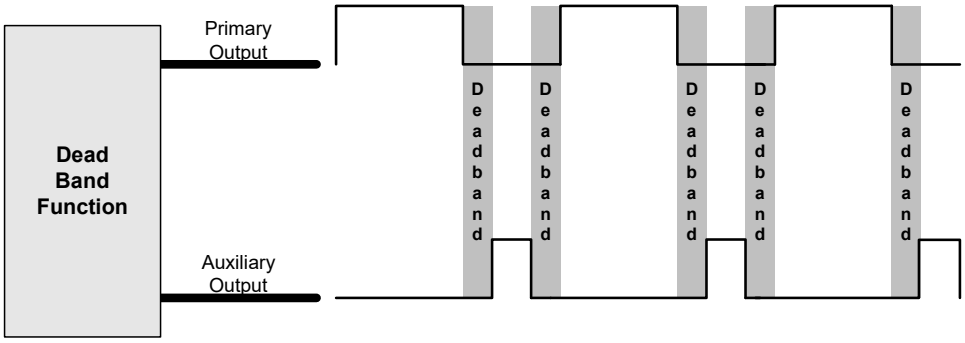
The PWM with the Dead Band User Module configures one or two blocks to create an 8- or 16-bit PWM and configures an additional block as the Dead Band function.

A dead band consists of a period register, a synchronous down counter, and a special dead band circuit. The DR2 register is only used to read the contents of DR0. As with the counter, when the dead band is disabled and a period value is written into DR1, the period value is also loaded into DR0. (Refer to the timing diagrams for this function on page 320.)

The dead band has two inputs: a PWM reference signal and a KILL signal. The PWM reference signal may be derived from one of two sources. By default, it is hardwired to be the primary output of the previous block. This previous block output is wired as an input to the 16-to-1 clock input mux. In the dead band case, as the previous block output is wired directly to the dead band reference input. If this mode is used, a PWM, or some other ***waveform*** generator, must be instantiated in the previous digital block. There is also an optional Bit Bang mode. In this mode, firmware toggles a register bit to generate a PWM reference; and therefore, the dead band may be used as a stand-alone block.

The KILL signal is derived from the data input signal to the block. Mode [1:0] is encoded as the Kill Type. In all cases when kill is asserted, the output is forced low immediately.

Figure 16-3. Dead Band Functional Overview



Mode bits are encoded for kill options and are detailed in the following table.

Table 16-6. Dead Band Kill Options

Mode[1:0]	Description
00b	Synchronous Restart KILL mode. Internal state is reset and reference edges are ignored, until the KILL signal is negated.
01b	Disable KILL mode. Block is disabled. KILL signal must be negated and user must re-enable the block in firmware to resume operation.
10b	Asynchronous KILL mode. Outputs are low only for the duration that the KILL signal is asserted, subject to a minimum disable time between one-half to one and one-half clock cycles. Internal state is unaffected.
11b	Reserved

When the block is initially enabled, both outputs are low. After enabling, a positive or negative edge of the incoming PWM reference enables the counter. The counter counts down from the period value to terminal count. At terminal count, the counter is disabled and the selected phase is asserted high. On the opposite edge of the PWM input, the output that was high is negated low and the process is repeated with the opposite phase. This results in the generation of a two phase non-overlapping clock matching the frequency and pulse width of the incoming PWM reference, but separated by a dead time derived from the period and the input clock.

There is a deterministic relationship between the incoming PWM reference and the output phases. The positive edge of the reference causes the primary output to be asserted to '1' and the negative edge of the reference causes the auxiliary output to be asserted to '1'.

16.1.8.1 Usability Exceptions

The following are usability exceptions for the Dead Band function.

1. The Dead Band function may not be chained.
2. Programming a dead band period value of 00h is not supported. The block output is undefined under this condition.
3. If the period (of either the ***high time*** or the ***low time*** of the reference input) is less than the programmed dead time, than the associated output phase will be held low.
4. DR0 may only be read (to transfer DR0 data to DR2) when the block is disabled.
5. If the asynchronous KILL signal is being used in a given application, the output of the dead band cannot be connected directly to the input of another digital block in the same row. Because the kill is asynchronous, the digital block output must be resynchronized through a row input before using it as a digital block input.

16.1.8.2 Block Interrupt

The Dead Band block has two interrupt sources. The default one is the Phase 1 primary output clock. When the KILL signal is asserted, the interrupt follows the same behavior of the Phase 1 output with respect to the various KILL modes. When KILL_INT is selected the KILL signal itself becomes interrupt. This is the second choice. **However, set KILL_INT only in KILL-Sync and KILL-Async mode**.

16.1.9 IPWM Function

The IPWM stands for Integrated dead band PWM. From a functional perspective, it combines the counter and dead band function in single block with limited dead band width selections. An IPWM consists of a period register, a synchronous down counter, a compare register, and a dead band width register. The IPWM counter function is identical to the Counter function, with the following exceptions:

- There is no counter gate input. The counting down is controlled by different sub modes.
- The multi-shot mode in IPWM is called PPG (Programmable Pulse Generator) mode. The function does not go into disable mode at last-shot but stops counting only. Hardware or software start (write one again to ‘EN’ bit) resumes the counting. Similarly the last-shot with the high of START does not stop counting, and the high of START does not affect the running counting.
- The comparison is DR0 > DR2, instead of DR0<= DR2 or DR0<DR2. Therefore, the compare out waveform is reversed.
- Writing to DR2 is always buffered when IPWM is running. Therefore, you do not need to set the register such as in the Counter function.

- TC could not be outputted, and compare out could not be outputted directly (must do through dead band function).
- KILL modes follow dead band function's setting.
- KILL does not affect the counter running state except in KILL-disable mode. The whole function is disabled when KILL is asserted in KILL-Disable mode.

The IPWM dead band function is identical to the Dead Band function, with the following exceptions:

- No need to set ref clock input from previous block. It derives from counter function's compare out in current block.
- Dead band width selections are limited. It could be 0/1/2/4/8/16/32/64 block clock cycles. 0 means there is no dead band protection. DR0 is not for dead band width register.
- Dead band function uses same one clock which is block clock. Counter function is also running on this clock.
- A new interrupt source is added, which comes from KILL signal itself.

IPWM may be chained in 8-bit blocks up to 32 bits.

Table 16-7. IPWM Control signals in Chained Blocks

Item	Configured in
KILL	LSB Block
START	LSB Block
Multi-shot Period	MSB Block
Dead Band Width	MSB Block
KILL Mode (in FN)	MSB Block
Clock	All Chained Blocks

16.1.9.1 Usability Exceptions

The following are usability exceptions for the IPWM function:

1. DR0 may only be read (to transfer DR0 data to DR2) when the block is disabled.
2. CR1 is not writeable when the IPWM is enabled.

16.1.9.2 Block Interrupt

The IPWM block has two interrupt sources. They are identical to the Dead Band function except the KILL_INT bit is in DxCxxCR0 instead of DxCxxCR1.

16.1.10 CRCPRS Function

A Cyclic Redundancy Check/Pseudo Random Sequence (CRCPRS) function consists of a polynomial register, a ***Linear Feedback Shift Register (LFSR)***, and a seed register. (See Figure 16-4 on page 295.) When the CRCPRS block is disabled and a ***seed value*** is written into DR2, the seed value is also loaded into DR0. When the CRCPRS is enabled, and synchronous clock and data are applied to the inputs, a CRC is computed on the ***serial*** data input stream. When the data input is forced to '0', then the block functions as a pseudo random sequencer (PRS) generator with the output data generated at the clock rate. The most significant bit (MSb) of the CRCPRS function is the primary output.

The CRCPRS has a selection of compare modes between DR0 and DR2. The default behavior of the compare is DR0==DR2. When the PRS function cycles through the seed value as one of the valid counts, the compare output is asserted high for one clock cycle. This is regarded as the epoch of the pseudo random sequence. The mode bits can be used to set other compare types. Setting Mode bit 0 to '1' causes the compare behavior to revert to DR0 <= DR2 or DR0 < DR2, depending upon Mode bit 1. The compare value is the auxiliary output. An interrupt is generated on compare true.

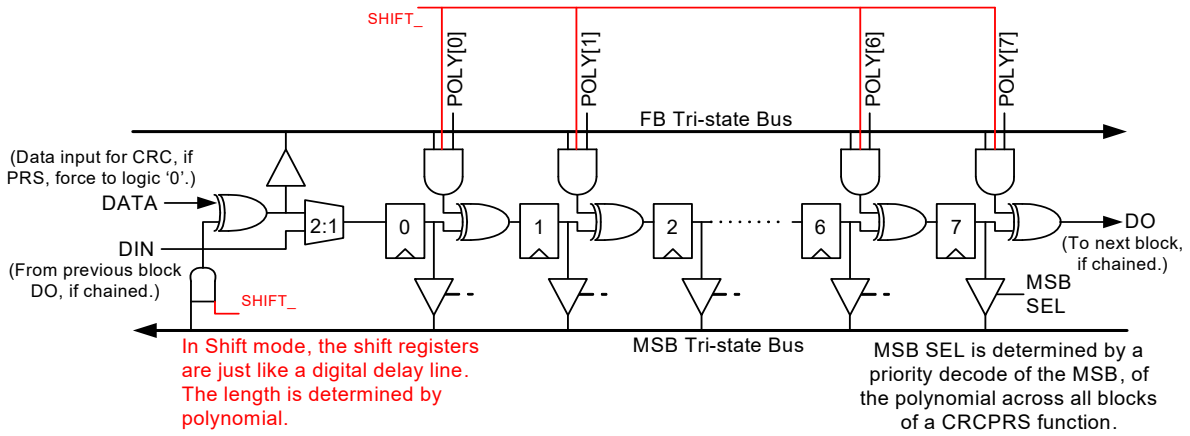
In PRS mode (that is, data input is fixed to zero), the Multi-shot and KILL functions are available. These modes are identical to the Timer/Counter function with the following exceptions:

- The multi-shot counter will count down when DR0 is equal to DR2 (seed), rather than when DR0 is equal to 00h in Timer/Counter. Note that the equivalence caused by writing DR2 in function disable mode or caused by KILL-reload will be ignored.
- KILL-Reload will reload DR2 data (seed) to DR0 instead of DR1 data (period in Timer/Counter).

CRCPRS mode offers an optional Pass function. By setting the Pass Mode bit in the CR0 register (bit 1), the CRCPRS function is overridden. In this mode, the data input is passed transparently to the primary output and an interrupt is generated on the rising of the data input. Similarly, the CLK input is passed transparently to the auxiliary output. This can only be used to pass signals to the global outputs. If the output of a pass function is needed as an input to another digital block, it must be resynchronized through the globals and row inputs.

CRCPRS supports shift mode. The LFSR acts as a digital delay line if all feedbacks are tied to zero. The shift-out data will appear on MSB bus. Note that 'Pass' mode has higher priority than 'Shift' mode.

Figure 16-4. CRCPRS LFSR Structure



LFSR Structure

The LFSR (Linear Feedback Shift register) structure, as shown in Figure 16-4, is implemented as a modular ***shift*** register generator. The least significant block in the chain inputs the MSb and XORs it with the DATA input, in the case of CRC computation. For PRS computation, the DATA input is forced to logic 0 (by input selection); and therefore, the MSb bus is directly connected to the FB bus. In the case of a chained block, the data input (DIN) comes directly from the data output (DO) of the LFSR in the previous block. The MSb selection, derived from the priority decode of the polynomial, enables one of the tri-state drivers to drive the MSb bus.

Determining the CRC Polynomial

Computation of an n-bit result is generally specified by a polynomial with n+1 terms, the last of which is X_{16}, where

$$X_0 = 1 \qquad \textbf{Equation 1}$$

As an example, the CRC-CCIT 16-bit polynomial is:

$$CRC - CCIT = X_{16} + X_{12} + X_5 + 1 \qquad \textbf{Equation 2}$$

The CRCPRS hardware assumes the presence of the X_0 term; and therefore, this polynomial can be expressed in 16 bits as 1000100000010000 or 8810h. Two consecutive digital blocks may be allocated to perform this function, with 88h as the MS block polynomial (DR1) and 10h as the LS block polynomial value.

Determining the PRS Polynomial

Generally, PRS polynomials are selected from pre-computed reference tables. It is important to note that there are two common ways to specify a PRS polynomial: simple register configuration and modular configuration. In the simple method, a ***shift register*** is implemented with a reduction XOR of the MSb and feedback taps as input into the least significant bit. In the modular method, there is an XOR operation implemented between each register bit and each tap point enables the XOR with the MSb for that given bit. The CRCPRS function implements the modular approach.

These are equivalent methods. However, there is a conversion that should be understood. If tables are specified in simple register format, then a conversion can be made to the modular format by subtracting each tap from the MS tap, as shown in the following example.

To implement a 7-bit PRS of length 127, one possible code is [7,6,4,2]s, which is in simple format. The modular format would be [7,7-6,7-4,7-2]m or [7,1,3,5]m which is equivalent to [7, 5, 3, 1]. Determining the polynomial to program is similar to the CRC example above. Set a ***binary*** bit for each tap (with bit 0 of the register corresponding to tap 1). Therefore, the code [7,5,3,1] would correspond to 01010101 or 55h.

In both the CRC and PRS cases, an appropriate seed value should be selected. All ones for PRS, or all ones or all zeros for CRC are typical values. Note that a seed value of all zeros should not be used in a PRS function, because PRS counting is inhibited by this seed.

16.1.10.1 Usability Exceptions

The following are usability exceptions for the CRCPRS function:

1. The polynomial register must only be written when the block is disabled.
2. CR1 is not writeable when the CRCPRS is enabled.

16.1.10.2 Block Interrupt

The CRCPRS block has three interrupt sources. The default one is the compare auxiliary output; that is, the compare output. The second one is data input when CRCPRS is in pass-by mode. The third one is the KILL signal when KILL_INT is selected.

16.1.11 SPI Protocol Function

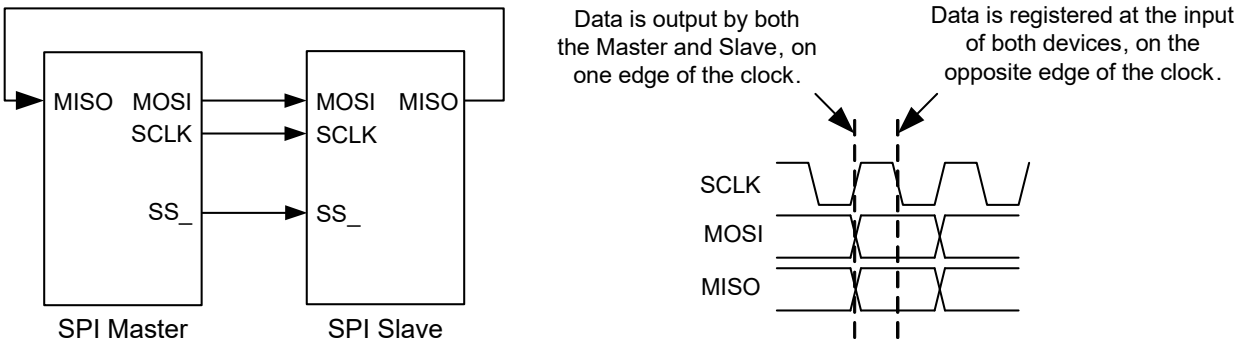
The Serial Peripheral Interface (SPI) is a Motorola™ specification for implementing full-duplex synchronous serial com-

munication between devices. The 3-wire ***protocol*** uses both edges of the clock to enable synchronous communication, without the need for stringent setup and hold requirements. Figure 16-5 shows the basic signals in a simple connection.

A device can be a master or slave. A master outputs clock and data to the ***slave device*** and inputs slave data. A slave device inputs clock and data from the ***master device*** and outputs data for input to the master. The master and slave together are essentially a circular shift register, where the master is generating the clocking and initiating data transfers.

A basic data transfer occurs when the master sends eight bits of data, along with eight clocks. In any transfer, both master and slave are transmitting and receiving simultaneously. If the master is only sending data, the received data from the slave is ignored. If the master wishes to receive data from the slave, the master must send dummy bytes to generate the clocking for the slave to send data back.

Figure 16-5. Basic SPI Configuration.



16.1.11.1 SPI Protocol Signal Definitions

The SPI Protocol signal definitions are located in Table 16-8. The use of the SS_ signal varies according to the capability of the slave device.

Table 16-8. SPI Protocol Signal Definitions

Name	Function	Description
MOSI	Master Out Slave In	Master data output.
MISO	Master In Slave Out	Slave data output.
SCLK	Serial Clock	Clock generated by the master.
SS_	Slave Select (active low)	This signal is provided to enable multi-slave connections to the MISO pin. The MOSI and SCLK pins can be connected to multiple slaves, and the SS_ input selects which slave will receive the input data and drive the MISO line.

16.1.12 SPI Master Function

The SPI Master (SPIM) offers SPI operating modes 0-3. By default, the MSb of the data byte is shifted out first. An additional option can be set to reverse the direction and shift the data byte out LSb first. (Refer to the timing diagrams for this function on page 323.)

When configured for SPIM, DR0 functions as a shift register, with input from the DATA input (MISO) and output to the primary output F1 (MOSI). DR1 is the TX Buffer register and DR2 is the RX Buffer register.

The SPI protocol requires data to be registered at the device input, on the opposite edge of the clock that operates the output shifter. An additional register (RXD), at the input to the DR0 shift register, has been implemented for this purpose. This register stores received data for one-half cycle, before it is clocked into the shift register.

The SPIM controls ***data transmission*** between master and slave, because it generates the bit clock for internal clocking and for clocking the SPIS. The bit clock is derived from the CLK input selection. Because the PSoC system clock generators produce clocks with varying duty cycles, the SPIM divides the input CLK by two to produce a bit clock with a 50 percent duty cycle. This clock is gated, to provide the SCLK output on the auxiliary output, during byte transmissions.

There are four control bits and four status bits in the control register that provide for PSoC device interfacing and synchronization.

The SPIM hardware has no support for driving the Slave Select (SS_) signal. The behavior and use of this signal is application and PSoC device dependent and, if required, must be implemented in firmware.

SPIM supports variable length from 8 bits to 16 bits. Two adjacent communication blocks are able to be chained together to achieve MAX 16-bit SPI. Note the last DCC block in one row can be chained with the first DCC block in next row. Below table shows the configurations in different length requirement. Note that the same clock setting should

be put in both blocks. And the SPI output comes from LSB Block and SPI input goes to MSB block if LSB first option is set. Otherwise SPI output comes from MSB block and SPI input goes to LSB block.

Table 16-9. Variable Length SPI Configuration

SPI Length	MSB Block				LSB Block			
	Chain	LSB	SPI Length	End Block (in FN)	Chain	LSB	SPI Length	End Block (in FN)
8-bit	0	*	*	*	N/A	N/A	N/A	N/A
12-bit	1	0	5'b0_1100	1	1	1	5'b0_1100	0

16.1.12.1 Usability Exceptions

The following are usability exceptions for the SPI Protocol function:

1. The MISO input must be resynchronized at the row inputs.
2. The DR2 (Rx Buffer) register is not writeable.
3. CR1 is not writeable when the SPIM is enabled.

16.1.12.2 Block Interrupt

The SPIM block has a selection of two interrupt sources: Interrupt on TX Reg Empty (default) or interrupt on SPI Complete. Mode bit 1 in the function register controls the selection. These mode are discussed in detail in "SPIM Timing" on page 323.

If SPI Complete is selected as the block interrupt, the control register must be read in the interrupt routine so that this status bit is cleared; otherwise, no subsequent interrupts are generated.

16.1.13 SPI Slave Function

The SPI Slave (SPIS) offers SPI operating modes 0-3. By default, the MSb of the data byte is shifted out first. An additional option can be set to reverse the direction and shift the data byte out LSb first. (Refer to the timing diagrams for this function on page 327.)

When configured for SPI, DR0 functions as a shift register, with input from the DATA input (MOSI) and output to the primary output F1 (MISO). DR1 is the TX Buffer register and DR2 is the RX Buffer register.

The SPI protocol requires data to be registered at the device input, on the opposite edge of the clock that operates the output shifter. An additional register (RXD), at the input to the DR0 shift register, is implemented for this purpose. This register stores received data for one-half cycle before it is clocked into the shift register.

The SPIS function derives all clocking from the SCLK input (typically an external SPI Master). This means that the master must initiate all transmissions. For example, to read a byte from the SPIS, the master must send a byte.

There are four control bits and four status bits in the control register that provide for PSoC device interfacing and synchronization.

In the SPIS, there is an additional data input, Slave Select (SS_), which is an ***active low*** signal. SS_ must be asserted to enable the SPIS to receive and transmit. SS_ has two high level functions: 1) To allow for the selection of a given slave in multi-slave environment, and 2) To provide additional clocking for TX data queuing in SPI modes 0 and 1.

SS_ may be controlled from an external pin through a Row Input or can be controlled by way of user firmware.

When SS_ is negated, the SPIS ignores any MOSI/SCLK input from the master. In addition, the SPIS ***state machine*** is reset, and the MISO output is forced to idle at logic 1. This allows for a wired-AND connection in a multi-slave environment. Note that if High-Z output is required when the slave is not selected, this behavior must be implemented in firmware with I/O writes to the port drive register.

SPIS also supports variable length from 8 bits to 16 bits. Two adjacent communication blocks can be chained together to achieve MAX 16-bit SPI. Note the last DCC block in one row can be chained with the first DCC block in the next row. SPIS variable length configuration is identical to variable length configuration in SPIM.

16.1.13.1 Usability Exceptions

The following are usability exceptions for the SPI Slave function:

1. The SPIS function may be chained.
2. CR1 is not writeable when the SPIS is enabled.

16.1.13.2 Block Interrupt

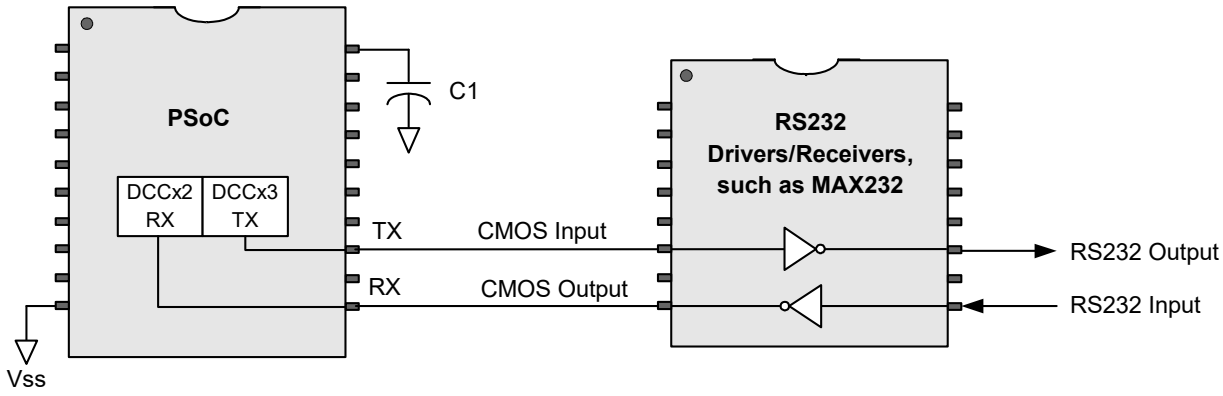
The SPIS block has a selection of two interrupt sources: Interrupt on TX Reg Empty (default) or interrupt on SPI Complete (same selection as the SPIM). Mode bit 1 in the function register controls the selection.

If SPI Complete is selected as the block interrupt, the control register must still be read in the interrupt routine so that this status bit is cleared; otherwise, no subsequent interrupts are generated.

16.1.14 Asynchronous Transmitter and Receiver Functions

The Asynchronous Transmitter and Receiver functions are illustrated in Figure 16-6.

Figure 16-6. Asynchronous Transmitter and Receiver Block Diagram



16.1.14.1 Asynchronous Transmitter Function

In the Transmitter function, DR0 functions as a shift register, with no input and with the TXD serial ***data stream*** output to the primary output F1. DR1 is a TX Buffer register and DR2 is unused in this configuration. (Refer to the timing diagrams for this function on page 329.)

Unlike SPI, which has no output latency, the TXD output has one cycle of ***latency***. This is because a mux at the output must select which bits to shift out: the shift register data, framing bits, ***parity***, or mark bits. The output of this mux is registered to remove glitches. When the block is first enabled or when it is idle, a mark bit (logic 1) is output.

The ***clock generator*** is a free running divide-by-eight circuit. Although dividing the clock is not necessary for the Transmitter function, the Receiver function does require a divide by eight for input sampling. It is also done in the Transmitter function, to allow the TX and RX functions to run off the same baud rate generator.

There are two formats supported: A 10-bit frame size including one start bit, eight data bits, and one ***stop bit*** or an 11-bit frame size including one start bit, eight data bits, one parity bit, and one stop bit.

The parity generator can be configured to output either even or odd parity on the eight data bits.

A write to the TX Buffer register (DR1) initiates a transmission and an additional byte can be buffered in this register, while transmission is in progress.

An additional feature of the Transmitter function is that a clock, generated with setup and hold time for the data bits only, is output to the auxiliary output. This allows connection to a CRC generator or other digital blocks.

16.1.14.2 Usability Exceptions

The following is a usability exception for the Transmitter function.

1. The Transmitter function may not be chained.

16.1.14.3 Block Interrupt

The Transmit block has a selection of two interrupt sources. Interrupt on TX Reg Empty (default) or interrupt on TX Complete. Mode bit 1 in the function register controls the selection.

If TX Complete is selected as the block interrupt, the control register must still be read in the interrupt routine so that this status bit is cleared; otherwise, no subsequent interrupts are generated.

16.1.14.4 Asynchronous Receiver Function

In the Receiver function, DR0 functions as the serial data shift register with RXD input from the DATA input selection. DR2 is an RX Buffer register and DR1 is unused in this configuration. (Refer to the timing diagrams for this function on page 331.)

The clock generator and START detection are integrated. The clock generator is a divide by eight which, when the system is idle, is held in reset. When a START bit (logic 0) is detected on the RXD input, the reset is negated and a ***bit rate (BR)*** clock is generated, subsequently sampling the RXD input at the center of the bit time. Every subsequent START bit resynchronizes the clock generator to the incoming bit rate.

There are two formats supported: A 10-bit frame size including one start bit, eight data bits, and one stop bit, or an 11-bit

frame size including one start bit, eight data bits, one parity bit, and one stop bit.

The received data is an input to the parity generator. It is compared with a received parity bit, if this feature is enabled. The parity generator can be configured to output either even or odd parity on the eight data bits.

After eight bits of data are received, the byte is transferred from the DR0 shifter to the DR2 RX Buffer register.

An additional feature of the Receiver function is that input data (RXD) and the synchronized clock are passed to the primary output and auxiliary output, respectively. This allows connection to a CRC generator or other digital block.

16.1.14.5 Usability Exceptions

The following are usability exceptions for the Asynchronous Receiver function.

1. The RXD input must be resynchronized through the row inputs.
2. DR2 is a read only register.

16.1.14.6 Block Interrupt

The Receiver has one fixed interrupt source, which is the RX Reg Full status.

The RX Buffer register must always be read in the RX interrupt routine, regardless of error status, and so on., so that RX Reg Full status bit is cleared; otherwise, no subsequent interrupts are generated.

16.1.15 DSM function

The Delta-Sigma-Modulator (DSM) performs density domain operation. It includes two parts in one dig-block: density signal generation and density domain multiplication.

Figure 16-7 illustrates density signal generation flow. The initial data will be loaded into DR0 by writing it into DR1. DR2 is density register. Then DR0 = DR0 – DR2 and the registered carry out is the generated density signal output. It can go to auxiliary output. For example, if you want the density at 50%, set DR2 to 80h, then CO occurs every other clock.

Figure 16-7. Density Domain Signal Generator

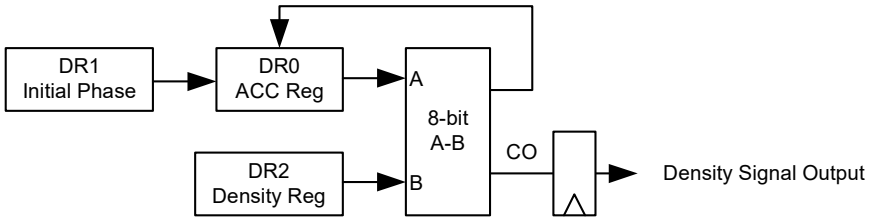
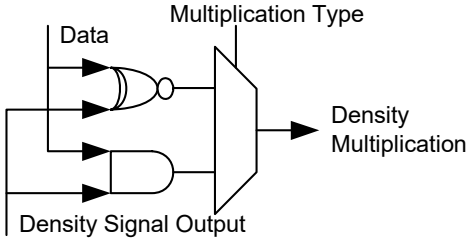


Figure 16-8 shows 2-input density signal multiplication flow. The multiplication type can be bipolar-reference type (through an XNOR) or signal-reference type (through an AND). One input comes from on-block generated density signal. Another comes from outside block through DATA selection MUX. The multiplying result can go to primary output of the block.

DSM function supports two types of KILL mode: KILL-Async or KILL-Disable. In KILL-Async mode, the block outputs are gated by KILL signal. In KILL-Disable mode, the function is disabled when KILL is asserted.

Figure 16-8. Density Signal Multiplication



16.1.15.1 Usability Exception

DSM function exists only in communication blocks, and is a signal block function.

1. DR1 is only writable when DSM function is disabled.

16.1.15.2 Block interrupt

There are two interrupt types in DSM function:

1. By default, interrupt occurs when CO goes high.
2. Interrupt occurs when KILL goes high.

16.2 Register Definitions

The following registers are associated with the Digital Blocks and listed in address order. Note that there are two banks of registers associated with the PSoC device. Bank 0 encompasses the user registers (Data and Control registers, and Interrupt Mask registers) for the device and Bank 1 encompasses the Configuration registers for the device. Both are defined below. Refer to the “Bank 0 Registers” on page 105 and the “Bank 1 Registers” on page 192 for a quick reference of PSoC registers in address order.

Each register description that follows has an associated register table showing the bit structure for that register. Depending on how many digital rows your PSoC device has (see the Rows column in the register tables below), only certain bits are accessible to be read or written (refer to the table titled “PSoC Device Characteristics” on page 268). The bits that are grayed out throughout this manual are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of ‘0’.

The Digital Block registers in this chapter are organized by function, as presented in Table 16-10. To reference timing diagrams associated with the digital block registers, see “Timing Diagrams” on page 317. For a complete table of digital block registers, refer to the “Summary Table of the Digital Registers” on page 269.

Data and Control Registers

The following table summarizes the Data and Control registers, by function type, for the digital blocks.

Table 16-10. Digital Block Data and Control Register Definitions

Function Type	DR0		DR1		DR2		CR0		CR1	
	Function	Access	Function	Access	Function	Access	Function	Access	Function	Access
Timer	Down Counter	R *	Period	W	Capture/Compare	RW	Control	RW	Control	RW
Counter	Down Counter	R *	Period	W	Compare	RW	Control	RW	Control	RW
Dead Band	Down Counter	R *	Period	W	N/A	N/A	Control	RW	Control	RW
IPWM	Down Counter	R *	Period	W	Compare	RW	Control	RW	Control	RW
CRCPRS	LFSR	R *	Polynomial	W	Seed	RW	Control	RW	Control	RW
SPIM	Shifter	N/A	TX Buffer	W	RX Buffer	R	Control/Status	RW **	Control	RW
SPIS	Shifter	N/A	TX Buffer	W	RX Buffer	R	Control/Status	RW **	Control	RW
TXUART	Shifter	N/A	TX Buffer	W	N/A	N/A	Control/Status	RW **	N/A	N/A
RXUART	Shifter	N/A	N/A	N/A	RX Buffer	R	Control/Status	RW **	N/A	N/A
DSM	Subtract	R *	Init-Phase	W	Density	RW	Control	RW	Control	RW

LEGEND

* In Timer, Counter, Dead Band, CRCPRS, IPWM, and DSM functions, a read of the DR0 register returns 00h and transfers DR0 to DR2.

** In the Communications functions, control bits are read/write accessible and status bits are read only accessible.

16.2.1 DxCxxDRx Registers

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	DxCxxDR0	4, 3, 2, 1	Data[7:0]								# : 00
0,xxh	DxCxxDR1	4, 3, 2, 1	Data[7:0]								W : 00
0,xxh	DxCxxDR2	4, 3, 2, 1	Data[7:0]								# : 00

LEGEND
Access is bit specific. Refer to the register detail for additional information.
xx An "x" after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the "Digital Register Summary" on page 269.

The DxCxxDRx Registers are the digital blocks' Data registers.

Bits 7 to 0: Data[7:0]. The Data registers and bits presented in this section encompass the DxCxxDR0, DxCxxDR1, and DxCxxDR2 registers. They are discussed according to which bank they are located in and then detailed in the tables that follow by function type.

For additional information, refer to the Register Details chapter for the following registers:

- DxCxxDR0 register on page 109.
- DxCxxDR1 register on page 110.
- DxCxxDR2 register on page 111.

16.2.1.1 Timer Register Definitions

There are three 8-bit Data registers and two 8-bit Control registers. Table 16-11 explains the meaning of the data registers in the context of timer operation. The Control registers are described beginning with section 16.2.2 DxCxxCR0 Register.

Note DR2 is not writeable when the Timer is enabled.

Table 16-11. Timer Data Register Descriptions

Name	Function	Description
DR0	Count Value	Not directly readable or writeable. During normal operation, DR0 stores the current count of a synchronous down counter. When disabled, a write to the DR1 period register is also simultaneously loaded into DR0 from the data bus. In KILL-Reload mode DR1 period data is loaded into DR0 at each rising edge of block clock when KILL is asserted. When disabled, a read of DR0 returns 00h to the data bus and transfers the contents of DR0 to DR2. This transfer only occurs in the addressed block. When enabled, a read of DR0 returns 00h to the data bus and synchronously transfers the contents of DR0 to DR2. It operates simultaneously on the byte addressed and all higher bytes in a multi-block timer. Note that when the hardware capture input is high, the read of DR0 (software capture) will be masked and will not occur. The hardware capture input must be low for a software capture to occur.
DR1	Period	Write only register. Data in this register sets the period of the count. The actual number of clocks counted is Period + 1. In the default one-half cycle Terminal Count mode (TC), a period value of 00h results in the primary output to be the inversion of the input clock. In the optional full cycle TC mode, a period of 00h gives a constant logic high on the primary output. When disabled, a write to this register also transfers the period value directly into DR0. When enabled, if the block frequency is 24 MHz or below, this register may be written to at any time, but the period will only be reloaded into DR0 in the clock following a TC. If the block frequency is 48 MHz, the Terminal Count or Compare Interrupt should be used to synchronize the new period register write; otherwise, the counter could be incorrectly loaded.
DR2	Capture/ Compare	Read write register (see **Exception** below). DR2 has multiple functions in a timer configuration. It is typically used as a capture register, but it also functions as a compare register. When enabled and a capture event occurs, the current count in DR0 is synchronously transferred into DR2. When enabled, the compare output is computed using the compare type (set in the function register mode bits) between DR0 and DR2. The result of the compare is output to the Auxiliary output. When disabled, a read of DR0 transfers the contents of DR0 into DR2 for the addressed block only. **Exception**: When enabled, DR2 is not writeable.

16.2.1.2 Counter Register Definitions

There are three 8-bit Data registers and two Control registers (a 7-bit and an 8-bit). Table 16-12 explains the meaning of these registers in the context of the Counter operation. Note that the descriptions of the registers are dependent on the enable/disable state of the block. This behavior is only related to the enable bit in the Control register, not the data input that provides the counter gate (unless otherwise noted). The Control registers are described beginning with section 16.2.2 DxCxxCR0 Register.

Note DR0 may only be read (to transfer DR0 data to DR2) when the block is disabled.

Table 16-12. Counter Data Register Descriptions

Name	Function	Description
DR0	Count Value	Not directly readable or writeable. During normal operation, DR0 stores the current count of a synchronous down counter. When disabled, a write to the DR1 period register is also simultaneously loaded into DR0 from the data bus. In KILL-Reload mode DR1 period data is loaded into DR0 at each rising edge of block clock when KILL is asserted. When disabled or the data input (counter gate) is low, a read of DR0 returns 00h to the data bus and transfers the contents of DR0 to DR2. This register should not be read when the counter is enabled and counting.
DR1	Period	Write only register. Data in this register sets the period of the count. The actual number of clocks counted is Period + 1. A period of 00h gives a constant logic high on the auxiliary output. When disabled, a write to this register also transfers the period value directly into DR0. When enabled, if the block frequency is 24 MHz or below, this register may be written to at any time, but the period will only be reloaded into DR0 in the clock following a TC. If the block frequency is 48 MHz, the Terminal Count or Compare Interrupt should be used to synchronize the new period register write; otherwise, the counter could be incorrectly loaded.
DR2	Compare	Read write register. DR2 functions as a Compare register. When enabled, the compare output is computed using the compare type (set in the function register mode bits) between DR0 and DR2. The result of the compare is output to the primary output. When disabled or the data input (counter gate) is low, a read of DR0 will transfer the contents of DR0 into DR2. DR2 may be written to when the function is enabled or disabled. In DR2-buffer mode in counter running, the data written to DR2 is stored first, then transferred to DR2 register when DR0 is being reloaded.

16.2.1.3 Dead Band Register Definitions

There are three 8-bit Data registers and a 3-bit Control register. Table 16-13 explains the meaning of these registers in the context of Dead Band operation. The Control registers are described beginning with section 16.2.2 DxCxxCR0 Register.

Note DR0 may only be read (to transfer DR0 data to DR2) when the block is disabled.

Table 16-13. Dead Band Register Descriptions

Name	Function	Description
DR0	Count Value	Not directly readable or writeable. During normal operation, DR0 stores the current count of a synchronous down counter. When disabled, a write to the DR1 period register is also simultaneously loaded into DR0 from the data bus. When disabled, a read of DR0 returns 00h to the data bus and transfers the contents of DR0 to DR2.
DR1	Period	Write only register. Data in this register sets the period of the dead band count. The actual number of clocks counted is Period + 1. The minimum period value is 00h, which sets a dead band time of one clock. When disabled, a write to this register also transfers the period value directly into DR0. When enabled, if the block frequency is 24 MHz or below, this register may be written to at any time, but the period will only be reloaded into DR0 in the clock following a Terminal Count (TC). If the block frequency is 48 MHz, the Terminal Count or Compare Interrupt should be used to synchronize the new period register write; otherwise, the counter could be incorrectly loaded.
DR2	Buffer	When disabled, a read of DR0 transfers the contents of DR0 into DR2.

16.2.1.4 IPWM Register Definitions

There are three 8-bit Data registers and two Control registers (a 7-bit and an 8-bit). Table 16-14 explains the meaning of these registers in the context of the IPWM operation. The Control registers are described beginning with section 16.2.2 DxCxxCR0 Register.

Note Read DR0 (to transfer DR0 data to DR2) only when the block is disabled.

Table 16-14. IPWM Data Register Descriptions

Name	Function	Description
DR0	Count Value	Not directly readable or writeable. During normal operation, DR0 stores the current count of a synchronous down counter. When disabled, a write to the DR1 period register is also simultaneously loaded into DR0 from the data bus. When disabled, a read of DR0 returns 00h to the data bus and transfers the contents of DR0 to DR2. This register should not be read when the counter is enabled and counting.
DR1	Period	Write only register. Data in this register sets the period of the count. The actual number of clocks counted is Period + 1. A period of 00h gives a constant logic high on the auxiliary output. When disabled, a write to this register also transfers the period value directly into DR0. When enabled, if the block frequency is 24 MHz or below, this register may be written to at any time, but the period will only be reloaded into DR0 in the clock following a TC. If the block frequency is 48 MHz, the Terminal Count or Compare Interrupt should be used to synchronize the new period register write; otherwise, the counter could be incorrectly loaded.
DR2	Compare	Read write register. DR2 functions as a Compare register. When enabled, the compare output is computed using the compare type (set in the function register mode bits) between DR0 and DR2. The result of the compare is output to the primary output. When disabled or the data input (counter gate) is low, a read of DR0 will transfer the contents of DR0 into DR2. DR2 may be written to when the function is enabled or disabled. When counter is running, the data written to DR2 is stored first, then transferred to DR2 register when DR0 is being reloaded.

16.2.1.5 CRCPRS Register Definitions

There are three 8-bit Data registers and two Control registers (a 7-bit and an 8-bit). Table 16-15 explains the meaning of these registers in the context of CRCPRS operation. Note that in the CRCPRS function a write to the DR2 Seed register is also loaded simultaneously into DR0. The Control registers are described beginning with section 16.2.2 DxCxxCR0 Register.

Table 16-15. CRCPRS Register Descriptions

Name	Function	Description
DR0	LFSR	Not directly readable or writeable. During normal operation, DR0 stores the state of a synchronous Linear Feedback Shift register. When disabled, a write to the DR2 Seed register is also simultaneously loaded into DR0 from the data bus. In KILL-Reload mode DR2 seed data is loaded into DR0 at each rising edge of block clock when KILL is asserted. When disabled, a read of DR0 returns 00h to the data bus and transfers the contents of DR0 to DR2. This register should not be read while the block is enabled.
DR1	Polynomial	Write only register. Data in this register sets the polynomial for the CRC or PRS function. **Exception**: This register must only be written when the block is disabled.
DR2	Seed/Residue	Read write register. DR2 functions as a Seed and Residue register. When disabled, a write to this register also transfers the seed value directly into DR0. When enabled, DR2 may be written to at any time. The value written will be used in the Compare function. When enabled, the compare output is computed using the compare type (set in the function register mode bits) between DR0 and DR2. The result of the compare is output to the auxiliary output. When disabled, a read of DR0 will transfer the contents of DR0 into DR2. This feature can be used to read out the residue, after a CRC operation is complete.

16.2.1.6 SPI Master Register Definitions

There are three 8-bit Data registers and two Control/Status registers (an 8-bit and a 7-bit). Table 16-16 explains the meaning of these registers in the context of SPIM operation. The Control registers are described beginning with section 16.2.2 DxCxx-CR0 Register.

Table 16-16. SPIM Data Register Descriptions

Name	Function	Description
DR0	Shifter	Not readable or writeable. During normal operation, DR0 implements a Shift register for shifting serial data.
DR1	TX Buffer	Write only register. If no transmission is in progress and this register is written to, the data from this register (DR1) is loaded into the Shift register (DR0), on the following clock edge, and a transmission is initiated. If a transmission is currently in progress, this register serves as a buffer for TX data. This register should only be written to when TX Reg Empty status is set, and this write clears the TX Reg Empty status bit in the Control register. When the data is transferred from this register (DR1) to the Shift register (DR0), then TX Reg Empty status is set.
DR2	RX Buffer	Read only register. When a byte transmission/reception is complete, the data in the shifter (DR0) is transferred into the RX Buffer register and RX Reg Full status in the Control register is set. A read from this register (DR2) clears the RX Reg Full status bit in the Control register.

16.2.1.7 SPI Slave Register Definitions

There are three 8-bit Data registers and two Control/Status registers (an 8-bit and a 7-bit). Table 16-17 explains the meaning of these registers in the context of SPIS operation. The Control registers are described beginning with section 16.2.2 DxCxx-CR0 Register.

Table 16-17. SPIS Data Register Descriptions

Name	Function	Description
DR0	Shifter	Not readable or writeable. During normal operation, DR0 implements a Shift register for shifting serial data.
DR1	TX Buffer	Write only register. This register should only be written to when TX Reg Empty status is set and the write clears the TX Reg Empty status bit in the Control register. When the data is transferred from this register (DR1) to the Shift register (DR0), then TX Reg Empty status is set.
DR2	RX Buffer	Read only register. When a byte transmission/reception is complete, the data in the shifter (DR0) is transferred into the RX Buffer register and RX Reg Full status in the Control (CR0) register is set. A read from this register (DR2) clears the RX Reg Full status bit in the Control register.

16.2.1.8 Transmitter Register Definitions

There are three 8-bit Data registers and one 5-bit Control/Status register. Table 16-18 explains the meaning of these registers in the context of Transmitter operation. The Control registers are described beginning with section 16.2.2 DxCxxCR0 Register.

Table 16-18. Transmitter Data Register Descriptions

Name	Function	Description
DR0	Shifter	Not readable or writeable. During normal operation, DR0 implements a shift register for shifting out serial data.
DR1	TX Buffer	Write only register. If no transmission is in progress and this register is written to, subject to the setup time requirement, the data from this register (DR1) is loaded into the Shift register (DR0) on the following clock edge and a transmission is initiated. If a transmission is currently in progress, this register serves as a buffer for TX data. This register should only be written to when TX Reg Empty status is set and this write clears the TX Reg Empty status bit in the Control (CR0) register. When the data is transferred from this register (DR1) to the Shift register (DR0), then TX Reg Empty status is set.
DR2	NA	Not used in this function.

16.2.1.9 Receiver Register Definitions

There are three 8-bit Data registers and one 8-bit Control/Status register. Table 16-19 explains the meaning of these registers in the context of Receiver operation. The Control registers are described beginning with section 16.2.2 DxCxxCR0 Register.

Table 16-19. Receiver Data Register Descriptions

Name	Function	Description
DR0	Shifter	Not readable or writeable. During normal operation, DR0 implements a Shift register for shifting in serial data from the RXD input.
DR1	NA	Not used in this function.
DR2	RX Buffer	Read only register. After eight bits of data are received, the contents of the shifter (DR0) is transferred into the RX Buffer register and the RX Reg Full status is set. The RX Reg Full status bit in the Control register is cleared when this register is read.

16.2.1.10 DSM Register Definitions

There are three 8-bit Data registers and two Control registers (a 5-bit and a 2-bit). Table 16-20 explains the meaning of these registers in the context of DSM operation. The Control registers are described beginning with section 16.2.2 DxCxxCR0 Register.

Table 16-20. DSM Data Register Descriptions

Name	Function	Description
DR0	Difference	Not directly readable or writeable. During normal operation, DR0 stores the current value of a synchronous subtractor. When disabled, a write to the DR1 initial minuend register is also simultaneously loaded into DR0 from the data bus. When disabled, a read of DR0 returns 00h to the data bus and transfers the contents of DR0 to DR2. This register should not be read when the subtractor is enabled.
DR1	Initial phase	Write only register. Data in this register sets the initial data of the subtractor. DR1 may only be written to when the function is disabled. When disabled, a write to this register also transfers the initial value directly into DR0.
DR2	Density Value	Read write register. DR2 functions as a subtrahend register. When enabled, DR0 = DR0 – DR2 is performed. The carry out is outputted to the auxiliary output. When disabled, a read of DR0 will transfer the contents of DR0 into DR2. DR2 may be written to when the function is enabled or disabled. When enabled, if the block frequency is 24 MHz or below, this register may be written to at any time, If the block frequency is 48 MHz, the DSM function should be disabled first.

16.2.2 DxCxxCR0 Register

The DxCxxCR0 Registers are the digital blocks' Control registers.

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	DxCxxCR0 (Timer Control:000)	4, 3, 2, 1	KILL[3:0]				NPS	TC Pulse Width	Capture Int	Enable	RW : 00

LEGEND
xx An "x" after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the "Digital Register Summary" on page 269.

Bits 7 to 1: . The bits for this register are described by function in Table 16-21.

Bit 0: Enable. This bit is used to synchronously enable or disable the programmed function.

For a complete description of bit functionality, refer to the DxCxxCR0 (Timer Control:000) register on page 112.

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	DxCxxCR0 (Counter Control:001)	4, 3, 2, 1	KILL[3:0]				NPS		DR2BufEN	Enable	RW : 00

LEGEND
xx An "x" after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the "Digital Register Summary" on page 269.

Bits 7 to 1: The bits for this register are described by function in Table 16-21.

Bit 0: Enable. This bit is used to synchronously enable or disable the programmed function.

For a complete description of bit functionality, refer to the DxCxxCR0 (Counter Control:001) register on page 113.

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	DxCxxCR0 (CRCPRS Control:010)	4, 3, 2, 1	KILL[3:0]					Shift Mode	Pass Mode	Enable	RW : 00

LEGEND
xx An "x" after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the "Digital Register Summary" on page 269.

Bits 7 to 1: The bits for this register are described by function in Table 16-21.

Bit 0: Enable. This bit is used to synchronously enable or disable the programmed function.

For a complete description of bit functionality, refer to the DxCxxCR0 (CRCPRS Control:010) register on page 115.

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	DxCxxCR0 (IPWM Control:011)	4, 3, 2, 1	START[3:0]				NPS	KILL_INT	SWT	Enable	RW : 00

LEGEND
xx An "x" after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the "Digital Register Summary" on page 269.

Bits 7 to 1: The bits for this register are described by function in Table 16-21.

Bit 0: Enable. This bit is used to synchronously enable or disable the programmed function.

For a complete description of bit functionality, refer to the DxCxxCR0 (IPWM Control:011) register on page 116.

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	DxCxxCR0 (Dead Band Control:100)	4, 3, 2, 1						Bit Bang Clock	Bit Bang Mode	Enable	RW : 00

LEGEND
xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Digital Register Summary” on page 269.

Bits 7 to 1: The bits for this register are described by function in Table 16-21.

Bit 0: Enable. This bit is used to synchronously enable or disable the programmed function.

For a complete description of bit functionality, refer to the DxCxxCR0 (Dead Band Control:100) register on page 114.

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	DCCxxCR0 (SPIM Control:0-110)	4, 3, 2, 1	LSb First	Overrun	SPI Complete	TX Reg Empty	RX Reg Full	Clock Phase	Clock Polarity	Enable	# : 00

LEGEND
Access is bit specific. Refer to the register detail for additional information.
xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Digital Register Summary” on page 269.

Bits 7 to 1: The bits for this register are described by function in Table 16-21.

Bit 0: Enable. This bit is used to synchronously enable or disable the programmed function.

For a complete description of bit functionality, refer to the DCCxxCR0 (SPIM Control:0-110) register on page 117.

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	DCCxxCR0 (SPIS Control:1-110)	4, 3, 2, 1	LSb First	Overrun	SPI Complete	TX Reg Empty	RX Reg Full	Clock Phase	Clock Polarity	Enable	# : 00

LEGEND
Access is bit specific. Refer to the register detail for additional information.
xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Digital Register Summary” on page 269.

Bits 7 to 1: The bits for this register are described by function in Table 16-21.

Bit 0: Enable. This bit is used to synchronously enable or disable the programmed function.

For a complete description of bit functionality, refer to the DCCxxCR0 (SPIS Control:1-110) register on page 118.

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	DCCxxCR0 (UART Transmitter Control)	4, 3, 2, 1			TX Complete	TX Reg Empty		Parity Type	Parity Enable	Enable	# : 00

LEGEND
Access is bit specific. Refer to the register detail for additional information.
xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Digital Register Summary” on page 269.

Bits 7 to 1: The bits for this register are described by function in Table 16-21.

Bit 0: Enable. This bit is used to synchronously enable or disable the programmed function.

For a complete description of bit functionality, refer to the DCCxxCR0 (UART Transmitter Control) register on page 120.

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	DCCxxCR0 (UART Receiver Control)	4, 3, 2, 1	Parity Error	Overrun	Framing Error	RX Active	RX Reg Full	Parity Type	Parity Enable	Enable	# : 00

LEGEND
Access is bit specific. Refer to the register detail for additional information.
xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Digital Register Summary” on page 269.

Bits 7 to 1: The bits for this register are described by function in Table 16-21.

Bit 0: Enable. This bit is used to synchronously enable or disable the programmed function.

For a complete description of bit functionality, refer to the DCCxxCR0 (UART Receiver Control) register on page 121.

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,xxh	DxCxxCR0 (DSM Control:111)	4, 3, 2, 1	KILL_SEL[3:0]							Enable	RW : 00

LEGEND
xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Digital Register Summary” on page 269.

Bits 7 to 1: The bits for this register are described by function in Table 16-21.

Bit 0: Enable. This bit is used to synchronously enable or disable the programmed function.

For a complete description of bit functionality, refer to the DxCxxCR0 (DSM Control:111) register on page 119.

Table 16-21. DxCxxCR0 Control Register Descriptions

Function	Description
Timer	There are eight bits in the Control (CR0) register: one to enable the block, one to set the optional interrupt on capture, one to select between one-half and a full clock for Terminal Count (TC) output, one to select between extending or not extending compare output half cycle, and four bits for KILL signal selection.
Counter	There are eight bits in the Control (CR0) register: one to enable the block, one to enable DR2 update buffer, one to select between extending or not extending compare output half cycle, and four bits for KILL signal selection.
Dead Band	There are three bits in the Control (CR0) register: one bit to enable the block, and two bits to enable and control Dead Band Bit Bang mode. When Bit Bang mode is enabled, the output of this register is substituted for the PWM reference. This register may be toggled by user firmware, to generate PHI1 and PHI2 output clock with the programmed dead time. The options for Bit Bang mode are as follows: 0 Function uses the previous clock primary output as the input reference. 1 Function uses the Bit Bang Clock register as the input reference.
IPWM	There are seven bits in the Control (CR0) register: one to enable the block, one to set software trigger mode, one to select between extending or not extending compare output half cycle, and four bits for START signal selection. **Note** The IPWM function does not support NPS mode when integrated deadband function is enabled.
CRCPRS	There are seven bits in the Control (CR0) register: one to enable the block, one for bypass mode, one for shift mode, and four bits for KILL signal selection.
SPIM	The SPI Control (CR0) register contains both control and status bits. There are four control bits that are read/write: Enable, Clock Phase and Clock Polarity to set the mode, and LSb First which controls bit ordering. There are two read only status bits: Overrun and SPI Complete. There are two additional read only status bits to indicate TX and RX Buffer status.
SPIS	The SPI Control (CR0) register contains both control and status bits. There are four control bits that are read/write: Enable, Clock Phase and Clock Polarity to set the mode, and LSb First which controls bit ordering. There are two read only status bits: Overrun and SPI Complete. There are two additional read only status bits to indicate TX and RX Buffer status.
TXUART	The Transmitter Control (CR0) register contains three control bits and two status bits. The control bits are Enable, Parity Enable, and Parity Type, and have read/write access. The status bits, TX Reg Empty and TX Complete, are read only.
RXUART	The Receiver Control (CR0) register contains both control and status bits. The three control bits are read/write: Enable, Parity Enable, and Parity Type. There are five read only status bits: RX Reg Full, RX Active, Framing Error, Overrun, and Parity Error.
DSM	There are five bits in the Control (CR0) register: one to enable the block, and four bits for KILL signal selection.

16.2.3 DxCxxCR1 Register

The DxCxxCR1 registers are the digital blocks' Control registers (located in bank 1 of the PSoC device's memory map).

The bits for the following registers are described by function in Table 16-22.

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,xxh	DxCxxCR1 (Timer Control:000)	4, 3, 2, 1	Multi-Shot				KILL_INV	KILL_MD[1:0]		KILL_INT	RW : 00

LEGEND
xx An "x" after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the "Digital Register Summary" on page 269.

For a complete description of bit functionality, refer to the DxCxxCR1 (Timer Control:000) register on page 202.

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,xxh	DxCxxCR1 (Counter Control:001)	4, 3, 2, 1	Multi-Shot				KILL_INV	KILL_MD[1:0]		KILL_INT	RW : 00

LEGEND
xx An "x" after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the "Digital Register Summary" on page 269.

For a complete description of bit functionality, refer to the DxCxxCR1 (Counter Control:001) register on page 203.

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,xxh	DxCxxCR1 (CRCPRS Control:010)	4, 3, 2, 1	Multi-Shot				KILL_INV	KILL_MD[1:0]		KILL_INT	RW : 00

LEGEND
xx An "x" after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the "Digital Register Summary" on page 269.

For a complete description of bit functionality, refer to the DxCxxCR1 (CRCPRS Control:010) register on page 204.

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,xxh	DxCxxCR1 (IPWM Control:011)	4, 3, 2, 1	Multi-Shot				STARTINV	DBW[2:0]			RW : 00

LEGEND
xx An "x" after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the "Digital Register Summary" on page 269.

For a complete description of bit functionality, refer to the DxCxxCR1 (IPWM Control:011) register on page 205.

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,xxh	DxCxxCR1 (Dead Band Control:100)	4, 3, 2, 1								KILL_INT	RW : 00

LEGEND
xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Digital Register Summary” on page 269.

For a complete description of bit functionality, refer to the DxCxxCR1 (Dead Band Control:100) register on page 206.

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,xxh	DxCxxCR1 (SPIM Control:0-110)	4, 3, 2, 1	Chain	LSB		SPI Length					RW : 00

LEGEND
xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Digital Register Summary” on page 269.

For a complete description of bit functionality, refer to the DxCxxCR1 (SPIM Control:0-110) register on page 207.

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,xxh	DxCxxCR1 (SPIS Control:0-110)	4, 3, 2, 1	Chain	LSB		SPI Length					RW : 00

LEGEND
xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Digital Register Summary” on page 269.

For a complete description of bit functionality, refer to the DxCxxCR1 (SPIS Control:0-110) register on page 208.

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,xxh	DxCxxCR1 (DSM Control:111)	4, 3, 2, 1					KILL_INV			KILL_INT	RW : 00

LEGEND
xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Digital Register Summary” on page 269.

For a complete description of bit functionality, refer to the DxCxxCR1 (DSM Control:111) register on page 209.

Table 16-22. DxCxxCR1 Control Register Descriptions

Function	Description
Timer	There are 8 bits in the Control (CR1) register: one for KILL interrupt select, two to select KILL mode, one to decide whether invert KILL signal, and four to set multi-shot times.
Counter	There are 8 bits in the Control (CR1) register: one for KILL interrupt select, two to select KILL mode, one to decide whether invert KILL signal, and four to set multi-shot times.
Dead Band	There is 1 bit in the CR1 register for KILL interrupt select.
IPWM	There are 8 bits in the Control (CR1) register: three for dead band width selection, one to decide whether invert START signal, and four to set multi-shot times.
CRCPRS	There are 8 bits in the Control (CR1) register: one for KILL interrupt select, two to select KILL mode, one to decide whether invert KILL signal, and four to set multi-shot times.
SPIM	There are 7 bits in the Control (CR1) register: five to set SPI length, one to set whether it is LSB block, and one to set it is chained block.
SPIS	There are 7 bits in the Control (CR1) register: five to set SPI length, one to set whether it is LSB block, and one to set it is chained block.
TXUART	N/A
RXUART	N/A
DSM	There are 2 bits in the Control (CR1) register: one for KILL interrupt select, and one to decide whether invert KILL signal.

Interrupt Mask Registers

The following are the interrupt mask registers for the digital blocks.

16.2.4 INT_MSK1 Register

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E1h	INT_MSK1	4, 3, 2	DCC13	DCC12	DBC11	DBC10	DCC03	DCC02	DBC01	DBC00	RW : 00

The Interrupt Mask Register 1 (INT_MSK1) is used to enable the individual sources' ability to create pending interrupts for digital blocks.

Depending on the digital row configuration of your PSoC device (see the table titled "PSoC Device Characteristics" on page 21), some bits may not be available in the INT_MSK1 register.

Bit 7: DCC13. Digital communications block interrupt enable for row 1 block 3.

Bit 6: DCC12. Digital communications block interrupt enable for row 1 block 2.

Bit 5: DBC11. Digital basic block interrupt enable for row 1 block 1.

Bit 4: DBC10. Digital basic block interrupt enable for row 1 block 0.

Bit 3: DCC03. Digital communications block interrupt enable for row 0 block 3.

Bit 2: DCC02. Digital communications block interrupt enable for row 0 block 2.

Bit 1: DBC01. Digital basic block interrupt enable for row 0 block 1.

Bit 0: DBC00. Digital basic block interrupt enable for row 0 block 0.

For additional information, refer to the INT_MSK1 register on page 175.

Configuration Registers

The configuration block contains 3 registers: Function (DxCxxFN), Input (DxCxxIN), and Output (DxCxxOU). The values in these registers should not be changed while the block is enabled. Note that the Digital Block Configuration registers are all located in bank 1 of the PSoC device's memory map.

16.2.5 DxCxxFN Registers

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,xxh	DxCxxFN	4, 3, 2, 1	Data Invert	BCEN	End Single	Mode[1:0]		Function[2:0]			RW : 00

LEGEND
xx An "x" after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the "Digital Register Summary" on page 269.

The Digital Basic/Communications Type B Block Function Registers (DxCxxFN) contain the primary Mode and Function bits that determine the function of the block.

All bits in these registers are common to all functions, except those specified in Table 16-24.

Bit 7: Data Invert. This bit inverts the selected data input.

Bit 6: BCEN. This bit enables the primary output of the block, to drive the row broadcast block. The BCEN bit is set independently in each block; and therefore, care must be taken to ensure that only one BCEN bit, in a given row, is enabled. However, if any of the blocks in a given row have the BCEN bit set, the input that allows the broadcast net from other rows to drive the given row's broadcast net is disabled (see Figure 15-2 on page 281).

Bit 5: End Single. This bit is used to indicate the last or most significant block in a chainable function. This bit must also be set if the chainable function only consists of a single block.

Bits 4 and 3: Mode[1:0]. The mode bits select the options available for the selected function. These bits should only be changed when the block is disabled.

Bits 2 to 0: Function[2:0]. The function bits configure the block into one of the available block functions (six for the Comm block, four for the Basic block).

For additional information, refer to the DxCxxFN register on page 196.

Table 16-23. DxCxxFN Function Registers

[7]: Data Invert	1 == Invert block's data input 0 == Do not invert block's data input
[6]: BCEN	1 == Enable 0 == Disable
[5]: End Single	1 == Block is not chained or is at the end of a chain 0 == Block is at the start of or in the middle of a chain
[4:3]: Mode	Function specific
[2:0]: Function	000b: Timer 001b: Counter 010b: CRCPRS 011b: IPWM 100b: Dead band for PWM 101b: UART (DCCxx blocks only) 110b: SPI (DCCxx blocks only) 111b: DSM

Table 16-24. Digital Block Configuration Register Functional Descriptions

Function	Description
Timer	The mode bits in the Timer block control the Interrupt Type and the Compare Type.
Counter	The mode bits in the Counter block control the Interrupt Type and the Compare Type (same as the Timer function).
Dead Band	The mode bits are encoded as the kill type. See the table titled "Dead Band Kill Options" on page 293 for an explanation of Kill options.
IPWM	The mode bits are encoded as the kill type. It is identical to Dead Band function.
CRCPRS	The mode bits are encoded to determine the Compare type.
SPIM	Mode bit 1 selects interrupt type. Mode bit 0 selects master or slave (for SPIM, it is '0').
SPIS	Mode bit 1 selects interrupt type. Mode bit 0 selects master or slave (for SPIS, it is '1').
TXUART	Mode bit 0 selects between Transmitter and Receiver (in this case Mode bit 0 is set to '1' for TX) and Mode bit 1 selects the interrupt type.
RXUART	Mode bit 0 selects between Transmitter and Receiver (in this case Mode bit 0 is set to '0' for RX) and Mode bit 1 selects the interrupt type.
DSM	Mode bit 1 selects KILL mode. Mode bit 0 selects multiplication type.

16.2.6 DxCxxIN Registers

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,xxh	DxCxxIN	4, 3, 2, 1	Data Input[3:0]				Clock Input[3:0]				RW : 00

LEGEND
xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Digital Register Summary” on page 269.

The Digital Basic/Communications Type B Block Input Registers (DxCxxIN) are used to select the data and clock inputs.

These registers are common to all functional types, except the SPIS. The SPIS is unique in that it has three function inputs and one function output defined. Refer to the DxCxxOU registers.

The input registers are eight bits and consist of two 4-bit fields to control each of the 16-to-1 Clock and Data input muxes. The meaning of these fields depends on the external clock and data connections, which is context specific. See Table 16-25.

Bits 7 to 4:Data Input[3:0]. These bits control the data input.

Bits 3 to 0: Clock Input[3:0]. These bits control the clock input.

Table 16-25. Digital Block Input Definitions

Function	Inputs		
	DATA	CLK	Auxiliary
Timer	Capture	CLK	N/A
Counter	Enable	CLK	N/A
Dead Band	Kill	CLK	Reference *
CRCPRS	Serial Data **	CLK	N/A
SPIM	MISO	CLK	N/A
SPIS	MOSI	SCLK	SS_
Transmitter	N/A	8X Baud CLK	N/A
Receiver	RXD	8X Baud CLK	N/A

* The Dead Band reference input does not use the auxiliary input mux. It is hardwired to be the primary output of the previous block.

** For CRC computation, the input data is a serial data stream synchronized to the clock. For PRS mode, this input should be forced to logic 0.

For additional information, refer to the DxCxxIN register on page 198.

16.2.7 DxCxxOU Registers

Add.	Name	Rows	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,xxh	DxCxxOU	4, 3, 2, 1	AUXCLK		AUXEN	AUX I/O Select[1:0]		OUTEN	Output Select[1:0]		RW : 00

LEGEND
xx An “x” after the comma in the address field indicates that there are multiple instances of the register. For an expanded address listing of these registers, refer to the “Digital Register Summary” on page 269.

The Digital Basic/Communications Type B Block Output Registers (DxCxxOU) are used to control the connection of digital block outputs to the available row interconnect and control clock resynchronization.

When the selected function is SPI Slave (SPIS), the AUXEN and AUX I/O bits change meaning, and select the input source and control for the Slave Select (SS_) signal.

The Digital Block Output register is common to all functional types, except the SPIS. The SPIS function is unique in that it has three function inputs and one function output defined. When the Aux I/O Enable bit is '0', the Aux I/O Select bits are used to select one of four inputs from the auxiliary data input mux to drive the SS_ input. Alternatively, when the Aux I/O Enable bit is a '1', the SS_ signal is driven directly from the value of the Aux I/O Select[0] bit. Thus, the SS_ input can be controlled in firmware, eliminating the need to use an additional GPIO pin for this purpose. Regardless of how the SS_ bit is configured, a SPIS block has the auxiliary row output drivers forced off; and therefore, the auxiliary output is not available in this block.

The following table enumerates the Primary and Auxiliary outputs that are defined for a given block function. Most functions have two outputs defined (the exception is the SPI Slave, which has only one). One or both of these outputs can optionally be enabled for output. When output, these signals can be routed to other block inputs through row or global interconnect, or output to chip pins.

Table 16-26. Digital Block Output Definitions

Function	Outputs		
	Primary	Auxiliary	Interrupt
Timer	Terminal Count	Compare	Terminal Count or Last-shot or Compare True or Capture or KILL
Counter	Compare	Terminal Count	Terminal Count or Last-shot or Compare True or KILL
Dead Band	Phase 1	Phase 2	Phase 1 or KILL
IPWM	Phase 1	Phase 2	Phase 1 or KILL
CRCPRS	MSB	Compare	Compare or DS or KILL
SPIM	MOSI	SCLK	TX Reg Empty or SPI Complete
SPIS	MISO	N/A **	TX Reg Empty or SPI Complete
Transmitter	TXD	SCLK *	TX Reg Empty or TX Compete
Receiver	RXD	SCLK *	RX Reg Full
DSM	Multiplication	Density Signal	Density Signal or KILL

* The UART blocks generate an SPI mode 3 style clock that is only active during the data bits of a received or transmitted byte.

** In the SPIS, the field that is used to select the auxiliary output is used to control the auxiliary input to select the SS_.

Bits 7 and 6: AUXCLK. All digital block clock inputs must be resynchronized. The digital blocks have numerous selections for clocking. In addition to the system clocks such as VC1, VC2, and VC3, clocks generated by other digital blocks may be selected through row or global interconnect. To maintain the integrity of block timing, all clocks are resynchronized at the input to the digital block.

The two AUXCLK bits are used to enable the input clock resynchronization. When enabled, the input clock is resynchronized to the selected system clock, which occurs after the 16-to-1 multiplexing. The rules for selecting the value for this register are as follows:

- If the input clock is based on SYSCLK (for example, VC1, VC2, VC3 based on SYSCLK) or the output of other blocks whose clock source is based on SYSCLK, synchronize to SYSCLK.
- If the input clock is based on SYSCLKX2 (for example, VC3 based on SYSCLKX2) or another digital block clocked by SYSCLKX2, or a SYSCLKX2 based clock, synchronize to SYSCLKX2.
- If you want to clock the block at 24 MHz (SYSCLK), choose SYSCLK direct in the resynchronized bits (the 16-to-1 input clock selection is ignored).
- If you want to clock the block at 48 MHz (SYSCLKX2), choose SYSCLKX2 as the clock input selection and leave the resynchronized bits in bypass mode.

The following table summarizes the available selections of the AUXCLK bits.

Table 16-27. AUXCLK Bit Selections

Code	Description	Usage
00	Bypass	Use this selection only when SYSCLKX2 (48 MHz) is selected by the 16-to-1 clock multiplexer (see the DxCxxIN register).
01	Resynchronize to SYSCLK (24 MHz)	This is a typical selection. Use this setting for any SYSCLK-based clock: VC1, VC2, VC3 driven by SYSCLK, digital blocks with SYSCLK based source clocks, broadcast bus with source based on SYSCLK, row input and row outputs with source based on SYSCLK.
10	Resynchronize to SYSCLKX2 (48 MHz)	Use this setting for any SYSCLKX2-based clock: VC3 driven by SYSCLKX2, digital blocks with SYSCLKX2 based source clocks, broadcast bus with source based on SYSCLKX2, row input and row outputs with source based on SYSCLKX2.
11	SYSCLK Direct	Use this setting to clock the block directly using SYSCLK. Note that this setting is not strictly related to clock resynchronization: but because SYSCLK cannot resynchronize itself, it allows a direct skew controlled SYSCLK source.

Note Selecting VC1/1 or VC2/1 (when VC1 is 1), or VC3/1 when the input is SYSCLK, or SYSCLKX2 is not allowed.

Bit 5: AUXEN. The AUXEN bit enables the Auxiliary output to be driven onto the selected row output. If the selected function is SPI Slave, the meaning of this bit is different. The SPI Slave does not have a defined Auxiliary output, so this bit is used, in conjunction with the AUX I/O Select bits to control the Slave Select input signal (SS_). When this bit is set, the SS_ input is forced active; and therefore, ***routing*** SS_ from an input pin is unnecessary.

Bits 4 and 3: AUX I/O Select[1:0]. These two bits select one (out of the four) row outputs to drive the Auxiliary output onto. In SPI Slave mode, these bits are used in conjunction with the AUXEN bit to control the Slave Select (SS_) signal. In this mode, these two bits are used to select one of four row inputs for use as SS_. If no SS_ is required in a given application, the AUXEN bit can be used to force the SS_ input active; and therefore, routing SS_ in through a Row Input would not be required.

Bit 2: OUTEN. This bit enables the Primary output to be driven onto the selected row output.

Output Select[1:0]. These two bits indicate which of the four row outputs the Primary output will be driven onto.

For additional information, refer to the DxCxxOU register on page 200.

16.3 Timing Diagrams

The timing diagrams in this section are presented according to their functionality and are in the following order.

- "Timer Timing" on page 317
- "Counter Timing" on page 320
- "Dead Band Timing" on page 320
- "IPWM Timing" on page 322
- "CRCPRS Timing" on page 323
- "SPI Mode Timing" on page 323
- "SPIM Timing" on page 323
- "SPIS Timing" on page 327
- "Transmitter Timing" on page 329
- "Receiver Timing" on page 331
- "DSM Timing" on page 333

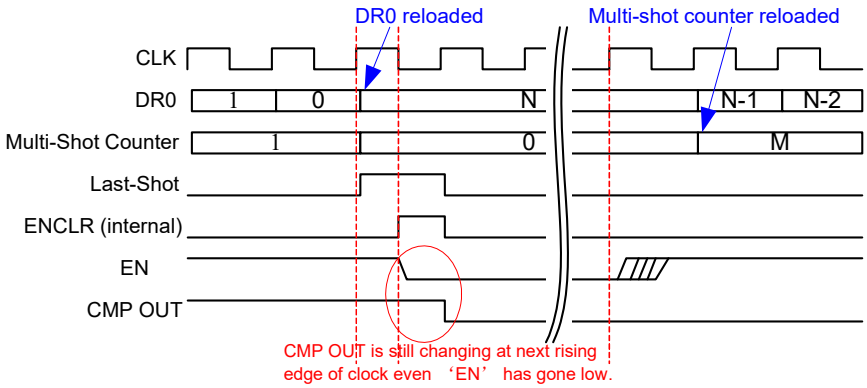
16.3.1 Timer Timing

Enable/Disable Operation. When the block is disabled, the clock is immediately gated low. All outputs are gated low, including the interrupt output. All internal state is reset to its configuration-specific reset state, except for DR0, DR1, and DR2 which are unaffected.

Terminal Count/Compare Operation. In the clock cycle following the count of 00h, the Terminal Count (TC) output is asserted. It is one-half cycle or a full cycle depending on the TC Pulse Width mode, as set in the block Control register. If this block stands alone or is the least significant block in a chain, the Carry Out (CO) signal is also asserted. If the period is set to 00h and the TC Pulse Width mode is one-half cycle, the output is the inversion of the input clock. The Compare (CMP) output will be asserted in the cycle following the compare true and will be negated one cycle after compare false.

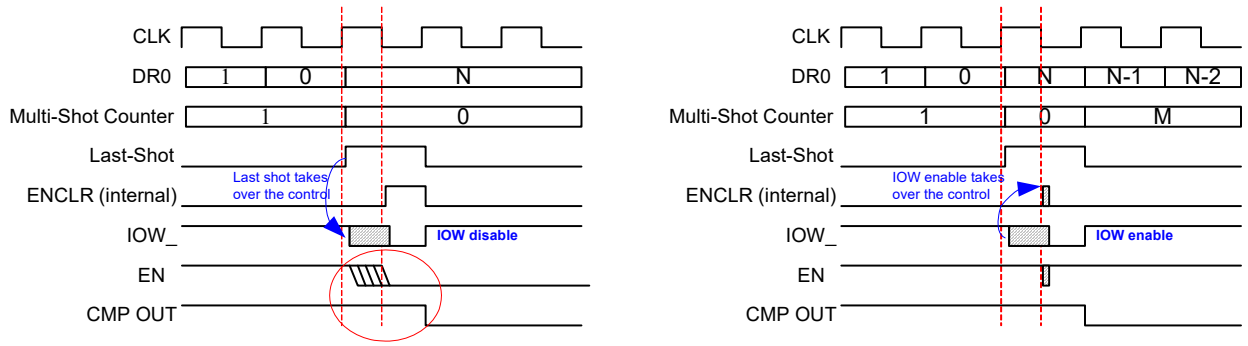
Multi-Shot Operation. There is a 4-bit multi-shot down-counter to count shot times. If you write multi-shot period register in CR1, it does not trigger writing data to this down-counter. The multi-shot counter reloading occurs in function enable state at the first clock after last-shot occurred or multi-shot period was written. In multi-shot mode, the last-shot is generated at rising edge of block clock when terminal count is one and multi-shot counter is one. The DR0 is reloaded simultaneously. Then at the next falling edge of the clock, the block enable bit is cleared. The multi-shot counter reloading must wait until the function is re-enabled.

Figure 16-9. Last-shot Generation and DR0/Multi-Shot Counter Reloaded



If the ENCLR and IOW enable occur simultaneously, the system IOW enable has higher priority. However if the ENCLR and IOW disable occur simultaneously, ENCLR takes over, shown as follows.

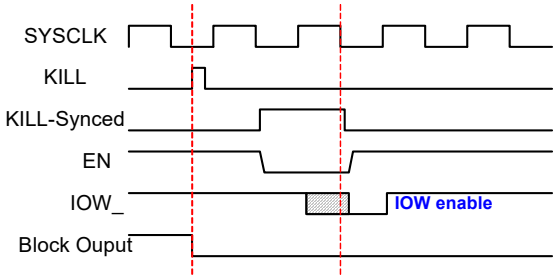
Figure 16-10. Last-Shot Meets IOW



In the special case when DR1 is zero, the function is not in multi-shot mode, whether or not the multi-shot period register in CR1 is written zero. Therefore, the minimal count period in multi-shot mode is two cycles.

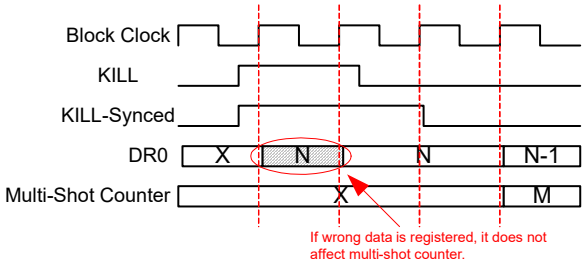
KILL Disable Operation. In KILL-Disable mode, the KILL signal is first synchronized at the falling edge of SYSCLK, and then the synchronized KILL is used to clear EN bit. If you synchronize at falling edge of SYSCLK, it keeps a safe timing when an IOW enable just follows it. Note that the potential block out is gated to low along with assertion of KILL immediately.

Figure 16-11. KILL-Disable and IOW Timing



KILL Reload Operation. In KILL-Reload mode, the KILL signal is synchronized at the rising edge of block clock and is extended one block clock cycle. DR0 reloads at rising edge of the block clock when KILL-synced is high. The multi-shot counter reloading occurs at rising edge of the block clock after KILL-synced is released.

Figure 16-12. KILL-Reload Timing



Multi-Block Terminal Count/Compare Operation. When timers are chained, the CO signal of a given block becomes the Carry In (CI) of the next most significant block in the chain. In a chained timer, the CO output indicates that block and all lower blocks are at 00h count. The CO is set up to the next positive edge of the clock, to enable the next higher block to count once for every Terminal Count (TC) of all lower blocks.

The terminal count out of a given block becomes the terminal count in of the next least significant block in the chain. The terminal count output indicates that the block and all higher blocks are at 00h count. The terminal count in/terminal count out chaining signals provide a way for the lower blocks to know when the upper blocks are at TC. Reload occurs when all blocks are at TC, which can be determined by CI, terminal count in, and the block zero detect. Example timing for a three block timer is shown in Figure 16-13.

The compare circuit compares registers DR0 <= DR2. (When Mode[1] = 1, the comparison is DR0 < DR2.)

Each block has an internal compare condition (DR0 compared to DR2), a chaining signal to the next block called CMPO, and the chaining signal from the previous block called CMPI. In any given block of a timer, the CMPO is used to generate the auxiliary output (primary output in the counter) with a one cycle clock delay.

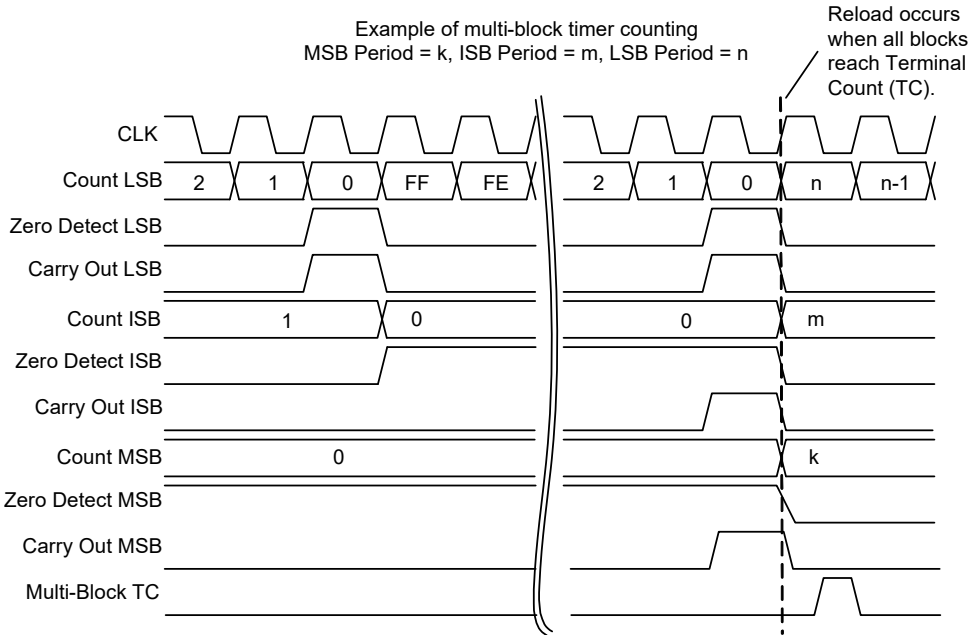
CMPO is generated from a combination of the internal compare condition and the CMPI input using the following rules:

1. For any given block, if DR0 < DR2, the CMPO condition is unconditionally asserted.
2. For any given block, if DR0 == DR2, CMPO is asserted only if the CMPI input to that block is asserted.
3. If the block is a start block, the effective CMPI depends on the compare type. If it is DR0 <= DR2, the effective CMPI input is '1'. If it is DR0 < DR2, the effective input is '0'.

Capture Operation. In the timer implementation, a rising edge of the data input or a CPU read of DR0 triggers a synchronous capture event. The result of this is to generate a latch enable to DR2 that loads the current count from DR0 into DR2. The latch enable signal is synchronized in such a way that it is not closing near an edge on which the count is changing.

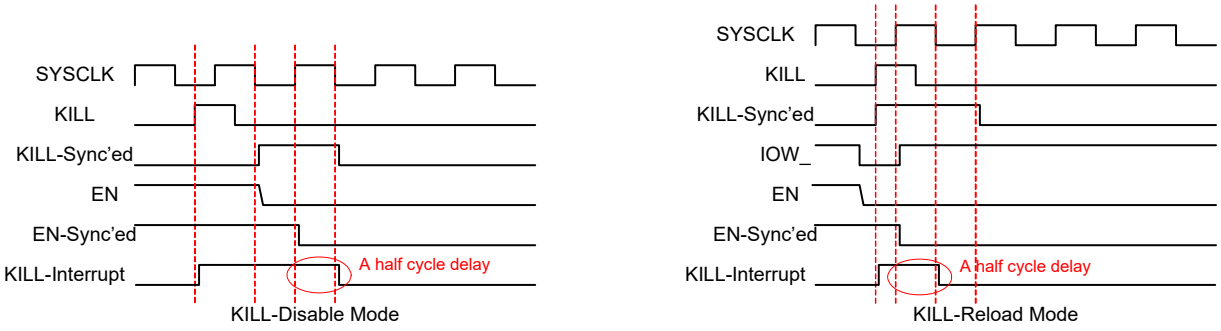
A limitation is that capture will not work with the block clock of 48 MHz. (A fundamental limitation to Timer Capture operation is the fact the GPIO inputs are currently synchronized to the 24 MHz system clock).

Figure 16-13. Multi-Block Timing



KILL Interrupt Generation . KILL interrupt occurs when the function is enabled. Therefore, no interrupt occurs if the KILL is already high in KILL-Disable mode when you write the function control register to be enabled. This is due to the function not being enabled due to KILL assertion.

Figure 16-14. Timer KILL Interrupt Generation



16.3.2 Counter Timing

Enable/Disable Operation. See Timer "Enable/Disable Operation" on page 317.

Terminal Count/Compare Operation. See Timer "Terminal Count/Compare Operation" on page 317.

Multi-Shot Operation. See Timer "Multi-Shot Operation" on page 317.

KILL-Disable Operation. See Timer "KILL Disable Operation" on page 318.

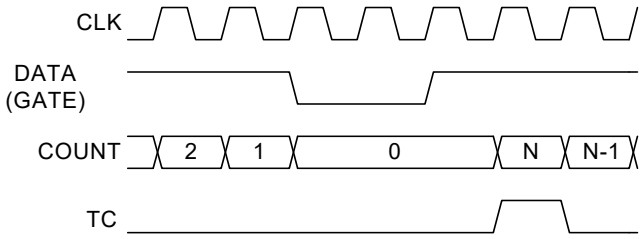
KILL-Reload Operation. See Timer "KILL Reload Operation" on page 318.

Multi-Block Operation. See Timer "Multi-Block Terminal Count/Compare Operation" on page 318.

Gate (Enable) Operation. The data input controls the counter enable. The transition on this enable must have at least one 24 MHz cycle of setup time to the block clock. This will be ensured if internal or synchronized external inputs are used.

As shown in Figure 16-15, when the data input is negated (counting is disabled) and the count is 00h, the TC output stays low. When the data input goes high again, the TC occurs on the following input clock. When the block is disabled, the clock is immediately gated low. All internal state is reset, except for DR0, DR1, and DR2, which are unaffected.

Figure 16-15. Counter Terminal Count Timing with Gate Disable



KILL Interrupt Generation. See Timer "KILL Interrupt Generation" on page 319.

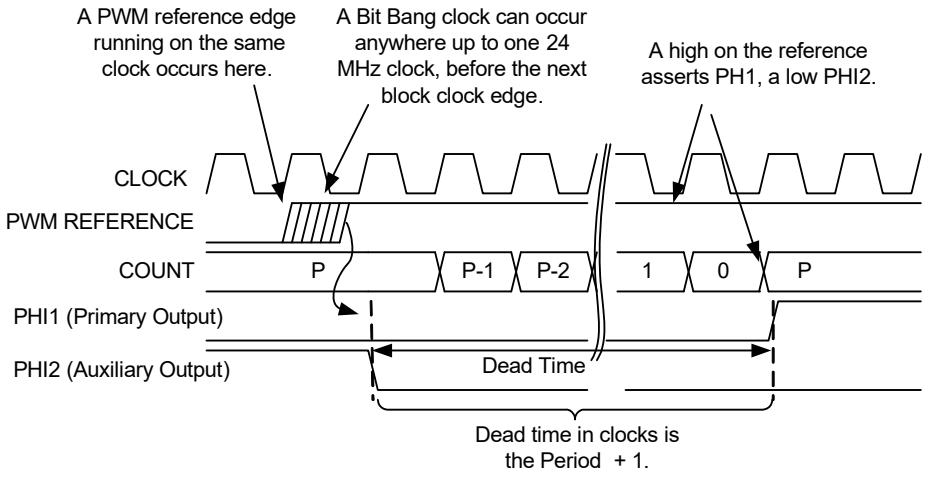
16.3.3 Dead Band Timing

Enable/Disable Operation. Initially both outputs are low. There are no critical timing requirements for enabling the block because dead band processing does not start until the first incoming positive or negative reference edge. In typical operation, it is recommended that the dead band block be enabled first, then the Pulse Width Modulator (PWM) generator block.

When the block is disabled, the clock is immediately gated low. All outputs are gated low, including the interrupt output. All internal state is reset to its configuration-specific reset state, except for DR0, DR1, and DR2 which are unaffected.

Normal Operation. Figure 16-16 shows typical dead band timing. The incoming reference edge can occur up to one 24 MHz system clock before the edge of the block clock. On the edge of the block clock, the currently asserted output is negated and the dead band counter is enabled. After Period + 1 clocks, the phase associated with the current state of the PWM reference is asserted (Reference High = Phase 1, Reference Low = Phase 2). The minimum dead time occurs with a period value of 00h and that dead time is one clock cycle.

Figure 16-16. Basic Dead Band Timing



16.3.3.1 Changing the PWM Duty Cycle

Under normal circumstances, the dead band period is less than the minimum PWM high or low time. As an example, consider Figure 16-17 where the low of the PWM is four clocks, the dead band period is two clocks, and the high time of the PHI2 is two clocks.

Figure 16-17. DB High Time is PWM Width Minus DB Period

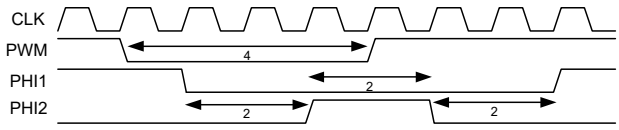
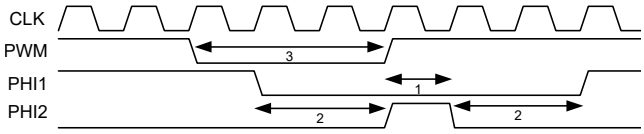


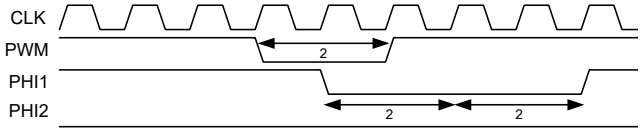
Figure 16-18 illustrates the reduction of the width of the PWM low time by one clock (to three clocks). The dead band period remains the same, but the high time for PHI2 is reduced by one clock (to one clock). Of course the opposite phase, PHI1, increases in length by one clock.

Figure 16-18. DB High Time is Reduced as PWM Width is Reduced



If the width of the PWM low time is reduced to a point where it is equal to the dead band period, the corresponding phase, PHI2, disappears altogether. Note that after the rising edge of the PWM, the opposite phase still has the programmed dead band. Figure 16-19 shows an example where the dead band period is two and the PWM width is two. In this case, the high time of PHI2 is zero clocks. Note that the Phase 1 dead band time is still two clocks.

Figure 16-19. PWM Width Equal to Dead Band Period



In the case where the dead band period is greater than the high or low of the PWM reference, the output of the associated phase will not be asserted high.

16.3.3.2 Kill Operation

It is assumed that the KILL input will not be synchronized at the row input. (This is not a requirement; however, if synchronized, the KILL operation will have up to two 24 MHz clock cycles latency which is undesirable.) To support the restart modes, the negation of KILL is internally (in the block) synchronized to the 24 MHz system clock.

There are three KILL modes supported. In all cases, the KILL signal asynchronously forces the outputs to logic 0. The differences in the modes come from how dead band processing is restarted.

1. **Synchronous Restart Mode**: When KILL is asserted high, the internal state is held in reset and the initial dead band period is reloaded into the counter. While KILL is held high, incoming PWM reference edges are ignored. When KILL is negated, the next incoming PWM reference edge restarts dead band processing. See Figure 16-20.
2. **Asynchronous Restart Mode**: When KILL is asserted high, the internal state is not affected. When KILL is negated, the outputs are restored, subject to a minimum disable time between one-half and one and one-half clock cycle. See Figure 16-21.
3. **Disable Mode**: There is no specific timing associated with Disable mode. The block is disabled and the user must re-enable the function in firmware to continue processing.

Figure 16-20. Synchronous Restart KILL Mode

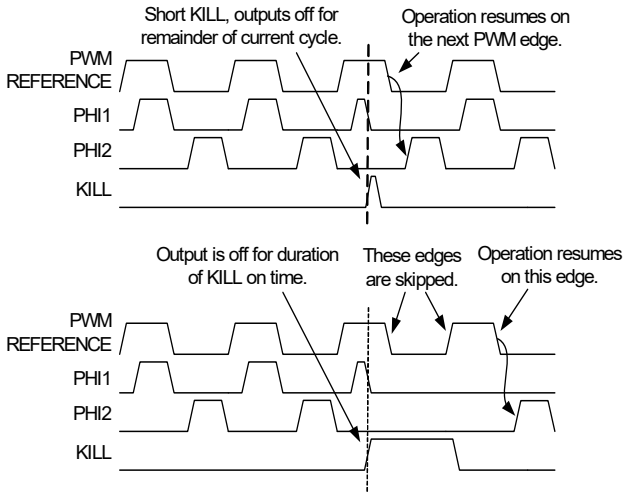
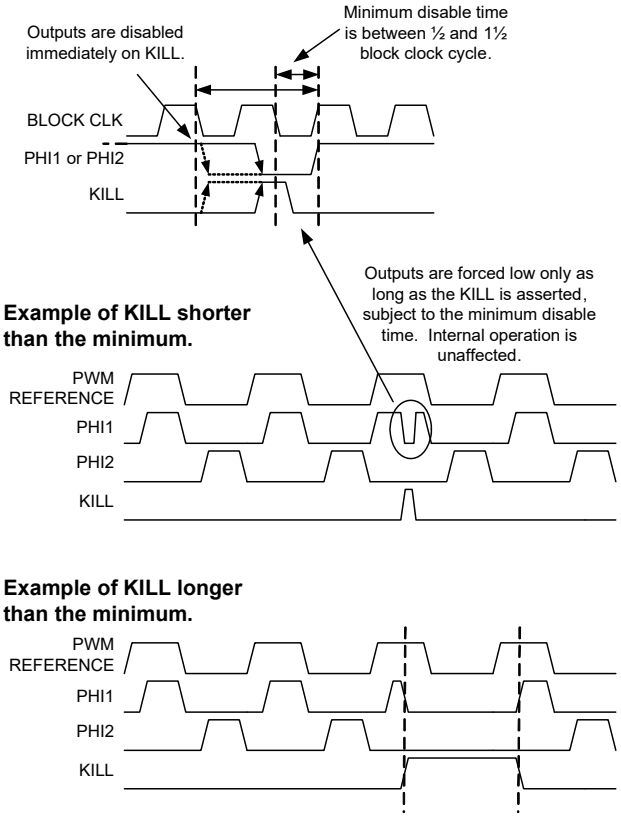


Figure 16-21. Asynchronous Restart Kill Mode



16.3.4 IPWM Timing

Enable/Disable Operation. See Timer "Enable/Disable Operation" on page 317.

PPG Start Operation. The IPWM function is in PPG mode when multi-shot period register is written with non-zero value. In non-PPG mode, the IPWM continuously outputs its PWM signal. Oppositely in PPG mode a START signal, either hardware signal or software signal (write '1' to EN bit), is used to trigger PWM output. The PWM output is stopped after the current PWM output period finishes if there is no more trigger signal. The PWM output period will start in the following conditions:

- If there is no PWM output (that is, IPWM is enabled and stopped), a STRAT starts a new PWM output period.
- A new PWM output period follows the previous one if START is high when the current PWM output will end (that is, at last-shot).

In another condition, the START signal does not affect IPWM running. To get safe timing, START is synchronized at the rising edge of block clock.

Compare Operation. The Compare operation is identical to the Timer function, except the polarity of compare output is swapped. Note that this compare output is also the reference input to the integrated dead band part. The dead band requires at least 2 cycles: high or low pulse on its reference input clock (need removal timing check in dead band generator sub-block if it is 2-cycle high or low pulse; do not need if it is larger than or equal to 3 cycles).

KILL Operation. The KILL operation has the same mechanism as the Dead Band function. The KILL mode is also identical to the Dead Band function. Refer to the Dead Band function KILL operation.

Figure 16-22. PPG Start Timing

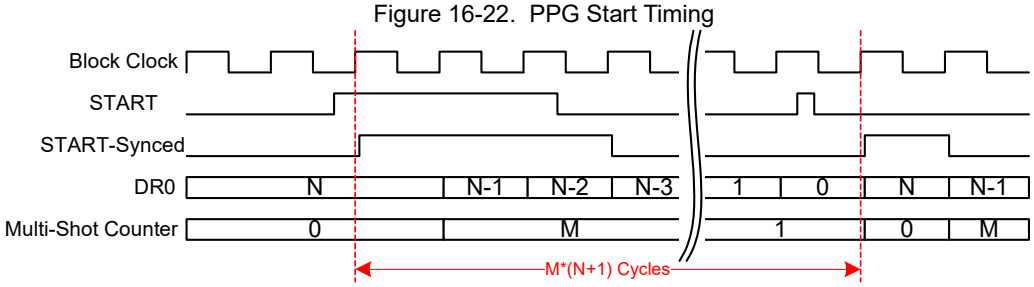
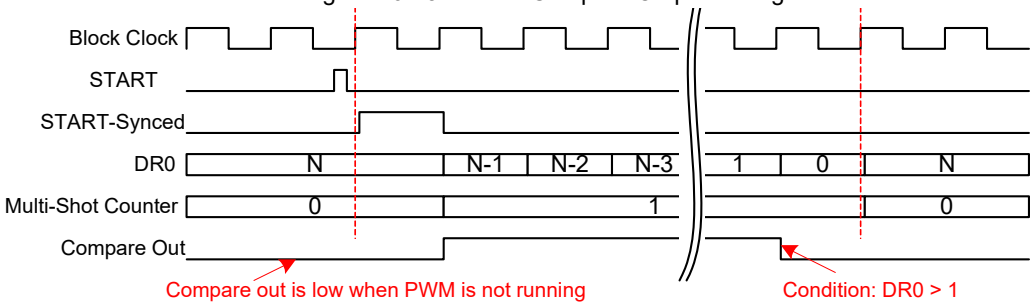


Figure 16-23. IPWM Compare Output Timing



16.3.5 CRCPRS Timing

Enable/Disable Operation. See Timer “Enable/Disable Operation” on page 317.

Multi-Shot Operation in PRS. See Timer “Multi-Shot Operation” on page 317.

KILL-Disable Operation in PRS. See Timer “KILL Disable Operation” on page 318.

KILL-Reload Operation in PRS. See Timer “KILL Reload Operation” on page 318.

KILL Interrupt Generation. See Timer “KILL Interrupt Generation” on page 319.

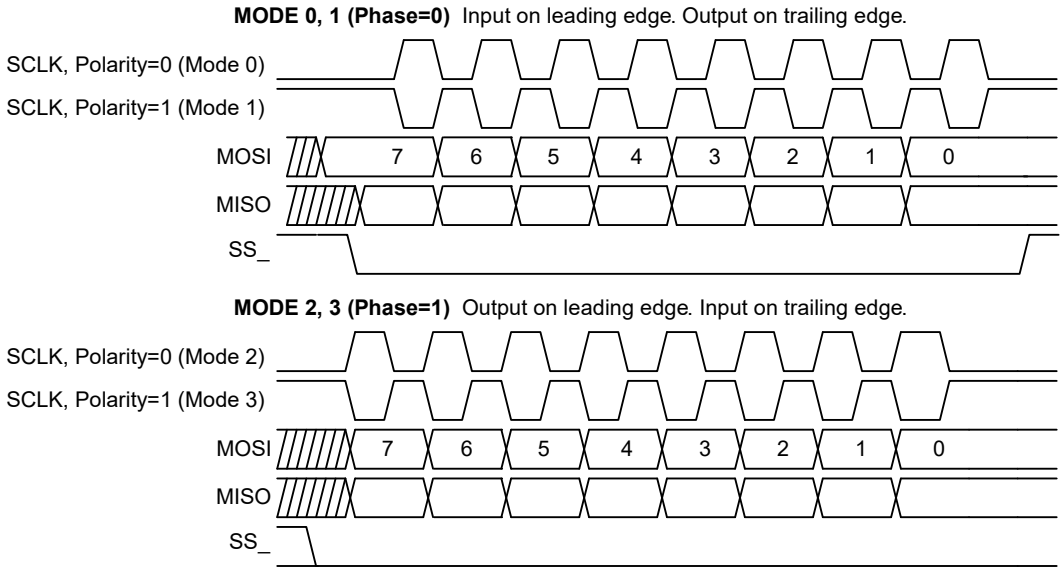
16.3.6 SPI Mode Timing

Figure 16-24 shows the SPI modes, which are typically defined as 0,1, 2, or 3. These mode numbers are an encoding of two control bits: Clock Phase and Clock Polarity.

Clock phase indicates the relationship of the clock to the data. When the clock phase is '0', it means that the data is registered as an input on the leading edge of the clock and the next data is output on the trailing edge of the clock. When the clock phase is '1', it means that the next data is output on the leading edge of the clock and that data is registered as an input on the trailing edge of the clock.

Clock polarity controls clock inversion. When clock polarity is set to '1', the clock ***idle state*** is high.

Figure 16-24. SPI Mode Timing



16.3.7 SPIM Timing

Enable/Disable Operation. As soon as the block is configured for SPIM, the primary output is the MSb or LSb of the Shift register, depending on the LSb First configuration in bit 7 of the Control register. The auxiliary output is '1' or '0' depending on the idle clock state of the SPI mode. This is the idle state.

When the SPIM is enabled, the internal reset is released on the divide-by-2 flip-flop and on the next positive edge of the selected input clock. This 1-bit divider transitions to a '1' and remains free-running thereafter.

When the block is disabled, the SCLK and MOSI outputs revert to their idle state. All internal state is reset (including CR0 status) to its configuration-specific reset state, except for DR0, DR1, and DR2 which are unaffected.

Normal Operation. Typical timing for a SPIM transfer is shown in Figure 16-25 and Figure 16-26. The user initially writes a byte to transmit when TX Reg Empty status is true. If no transmission is currently in progress, the data is loaded into the shifter and the transmission is initiated. The TX Reg Empty status is asserted again and the user is allowed to write the next byte to be transmitted to the TX Buffer register. After the last bit is output, if TX Buffer data is available with one-half clock setup time to the next clock, a new byte transmission will be initiated. A SPIM block receives a byte at the same time that it sends one. The SPI Complete or RX Reg Full can be used to determine when the input byte has been received.

Figure 16-25. Typical SPIM Timing in Mode 0 and 1

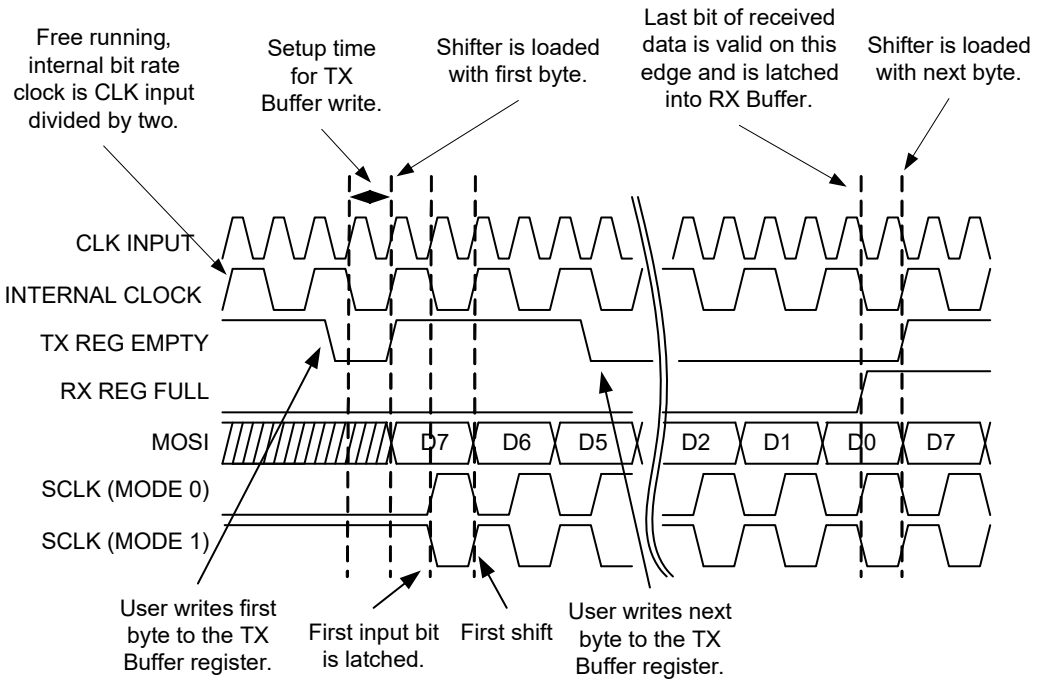
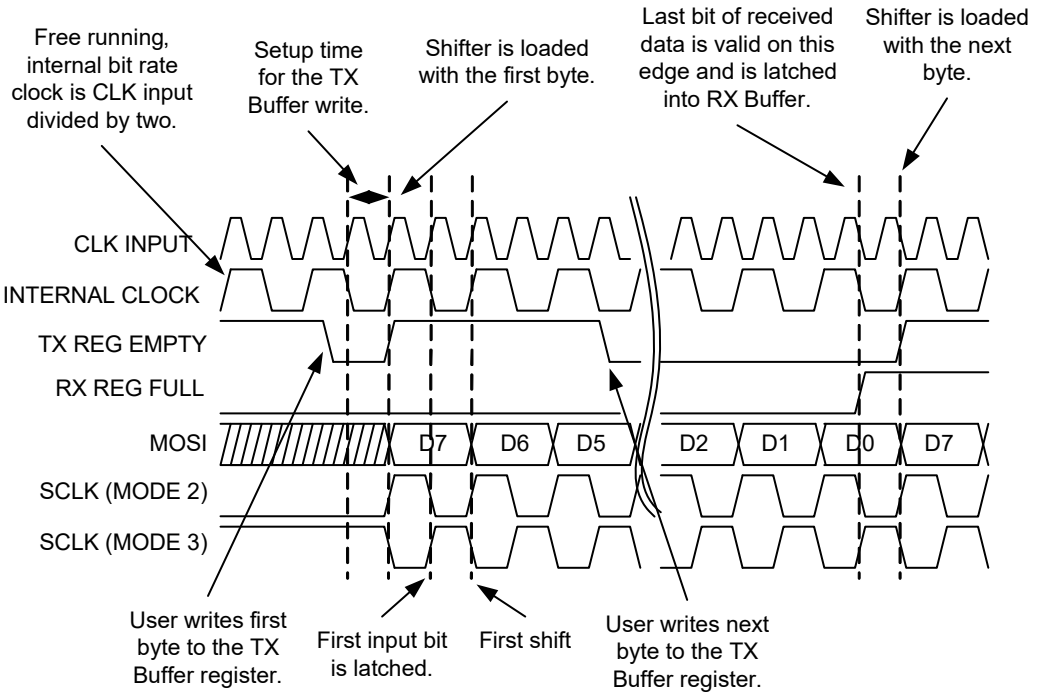


Figure 16-26. Typical SPIM Timing in Mode 2 and 3



Status Generation and Interrupts. There are four status bits in an SPI Block: TX Reg Empty, RX Reg Full, SPI Complete, and Overrun.

TX Reg Empty indicates that a new byte can be written to the TX Buffer register. When the block is enabled, this status bit is immediately asserted. This status bit is cleared when the user writes a byte of data to the TX Buffer register. TX Reg Empty is a control input to the state machine and, if a transmission is not already in progress, the assertion of this control signal initiates one. This is the default SPIM block interrupt. However, an initial interrupt is not generated when the block is enabled. The user must write a byte to the TX Buffer register and that byte must be loaded into the shifter before interrupts generated from the TX Reg Empty status bit are enabled.

RX Reg Full is asserted on the edge that captures the eighth bit of receive data. This status bit is cleared when the user reads the RX Buffer register (DR2).

SPI Complete is an optional interrupt and is generated when eight bits of data and clock have been sent. In modes 0 and 1, this occurs one-half cycle after RX Reg Full is set; because in these modes, data is latched on the leading edge of the clock and there is an additional one-half cycle remaining to complete that clock. In modes 2 and 3, this occurs at the same edge that the receive data is latched. This signal may be used to read the received byte or it may be used by the SPIM to disable the block after data transmission is complete.

Overrun status is set, if RX Reg Full is still asserted from a previous byte when a new byte is about to be loaded into the RX Buffer register. Because the RX Buffer register is implemented as a latch, Overrun status is set one-half bit clock before RX Reg Full status.

See Figure 16-27 and Figure 16-28 for status timing relationships.

Figure 16-27. SPI Status Timing for Modes 0 and 1

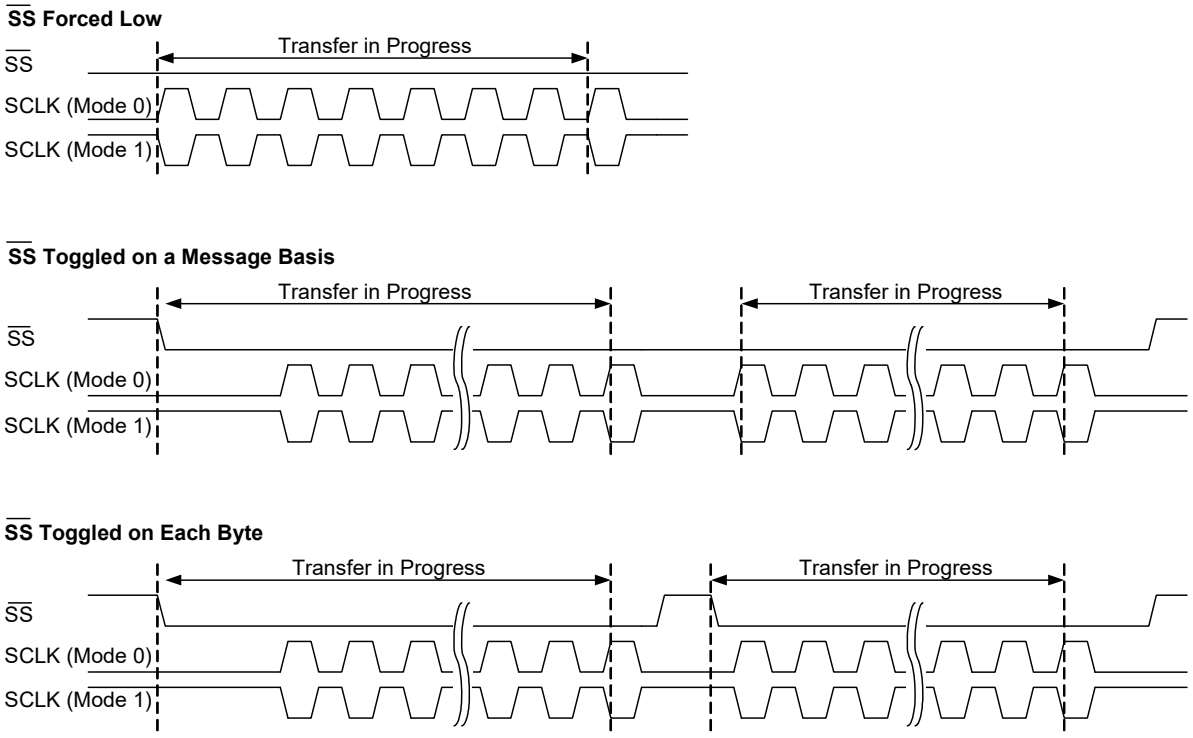
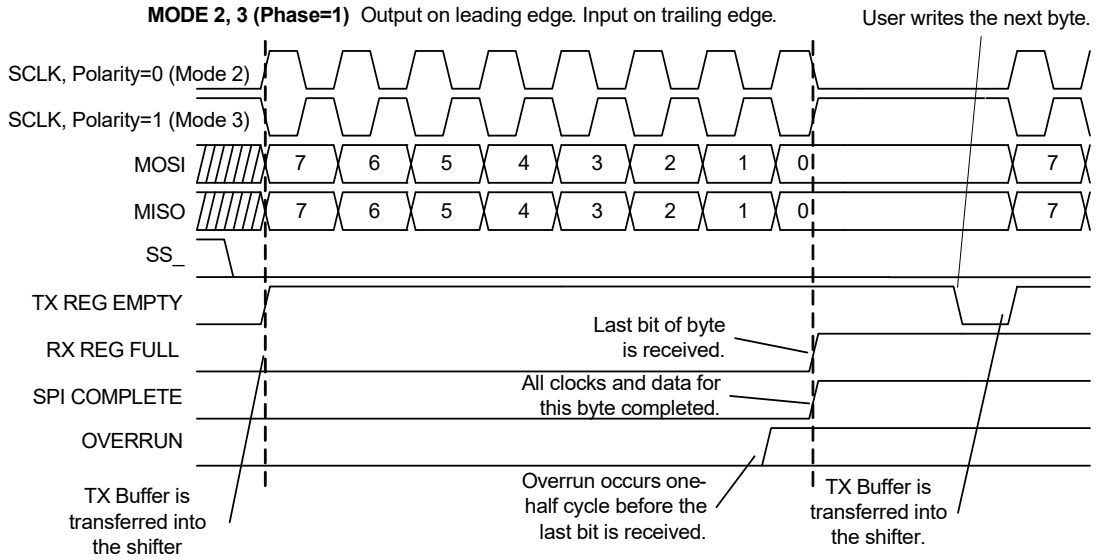


Figure 16-28. SPI Status Timing for Modes 2 and 3



Chained SPIM. When two adjacent communication blocks are chained to form a more-than-8-bit SPIM function, the preceding operations are maintained the same, with the following exceptions:

- More transmissions for more bits.
- Only need to enable LSB block to enable the function such as in chained Timer/Counter/CRCPRS/IPWM functions.
- Need to write MSB TX register first and then LSB register to initiate a transmission.
- Always read MSB RX data first and then LSB RX data.
- Always check LSB status bits for whole SPIM status if you follow above TX/RX read/write operation sequence.
- The interrupt in both blocks can be enabled and selected arbitrarily. (But if clearing SPI complete bit or TX Empty bit, still need to read the CR0 register or write DR1 register in that block).

16.3.8 SPIS Timing

Enable/Disable Operation. As soon as the block is configured for SPI Slave and before enabling, the MISO output is set to idle at logic 1. Both the enable bit must be set and the SS_ asserted (either driven externally or forced by firmware programming) for the block to output data. When enabled, the primary output is the MSb or LSb of the shift register, depending on the LSb First configuration in bit 7 of the Control register. The auxiliary output of the SPIS is always forced into tri-state.

Because the SPIS has no internal clock, it must be enabled with setup time to any external master supplying the clock. Setup time is also required for a TX Buffer register write, before the first edge of the clock or the first falling edge of SS_, depending on the mode. This setup time must be assured through the protocol and an understanding of the timing between the master and slave in a system.

When the block is disabled, the MISO output reverts to its idle '1' state. All internal state is reset (including CR0 status) to its configuration-specific reset state, except for DR0, DR1, and DR2 which are unaffected.

Normal Operation. Typical timing for a SPIS transfer is shown in Figure 16-29 and Figure 16-30. If the SPIS is primarily being used as a receiver, the RX Reg Full (polling only) or SPI Complete (polling or interrupt) status may be used to determine when a byte has been received. In this way, the SPIS operates identically with the SPIM. However, there are two main areas in which the SPIS operates differently: 1) SPIS behavior related to the SS_ signal, and 2) TX data queuing (loading the TX Buffer register).

Figure 16-29. Typical SPIS Timing in Modes 0 and 1

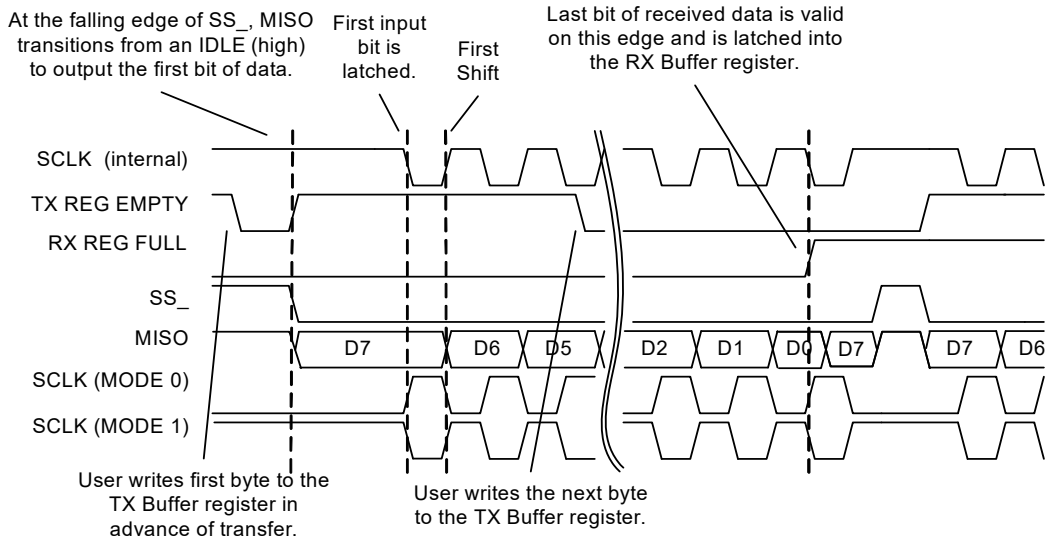
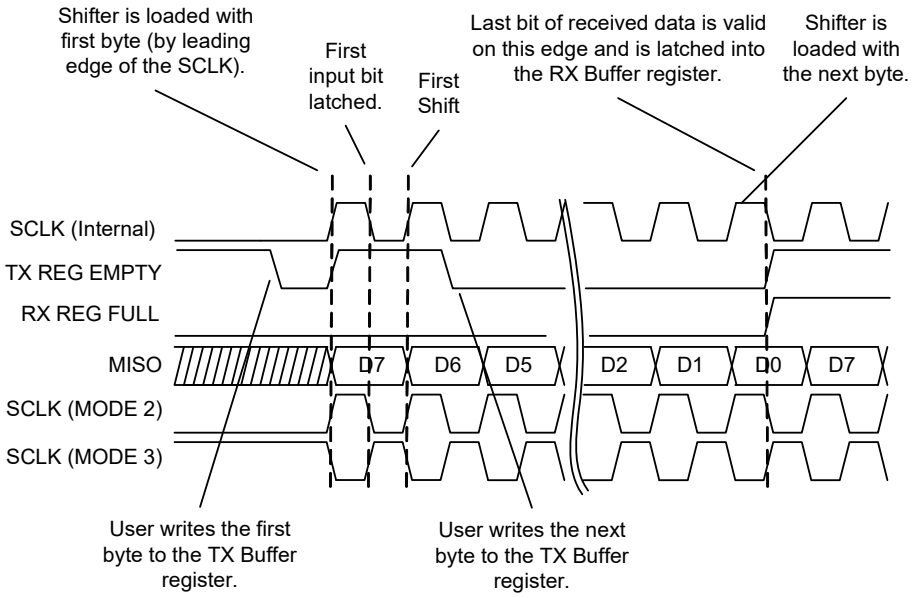


Figure 16-30. Typical SPIS Timing in Modes 2 and 3



Slave Select (SS_, active low). Slave Select must be asserted to enable the SPIS for receive and transmit. There are two ways to do this:

1. Drive the auxiliary input from a pin (selected by the Aux I/O Select bits in the output register). This gives the SPI master control of the slave selection in a multi-slave environment.
2. SS_ may be controlled in firmware with register writes to the output register. When Aux I/O Enable = 1, Aux I/O Select bit 0 becomes the SS_ input. This allows the user to save an input pin in single slave environments.

When SS_ is negated (whether from an external or internal source), the SPIS state machine is reset and the MISO output is forced to idle at logic 1. In addition, the SPIS will ignore any incoming MOSI/SCLK input from the master.

Status Generation and Interrupts. There are four status bits in the SPIS Block: TX Reg Empty, RX Reg Full, SPI Complete, and Overrun. The timing of these status bits are identical to the SPIM, with the exception of TX Reg Empty which is covered in the section on TX data queuing.

Status Clear On Read. Refer to the same subsection in “SPIM Timing” on page 323.

TX Data Queuing. Most SPI applications call for data to be sent back from the slave to the master. Writing firmware to accomplish this requires an understanding of how the Shift register is loaded from the TX Buffer register.

All modes use the following mechanism: 1) If there is no transfer in progress, 2) if the shifter is empty, and 3) if data is available in the TX Buffer register, the byte is loaded into the shifter.

The only difference between the modes is that the definition of “transfer in progress” is slightly different between modes 0 and 1, and modes 2 and 3.

Figure 16-31 illustrates TX data loading in modes 0 and 1. A transfer in progress is defined to be from the falling edge of SS_ to the point at which the RX Buffer register is loaded with the received byte. This means that in order to send a byte in the next transfer, it must be loaded into the TX Buffer register before the falling edge of SS_. This ensures a minimum setup time for the first bit, because the leading edge of the first SCLK must latch in the received data. If SS_ is not toggled between each byte or is forced low through the configuration register, the leading edge of SCLK is used to define the start of transfer. However, in this case, the user must provide the required setup time (one-half clock minimum before the leading edge), with a knowledge of system latencies and response times.

Figure 16-31. Mode 0 and 1 Transfer in Progress

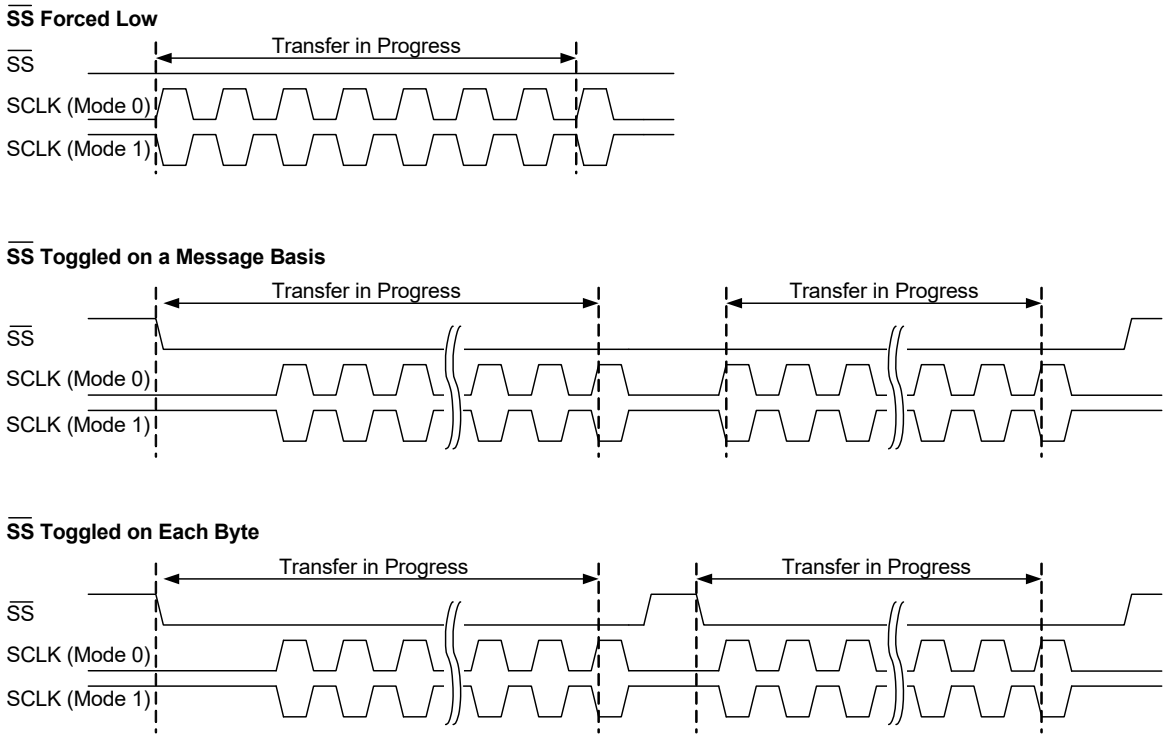
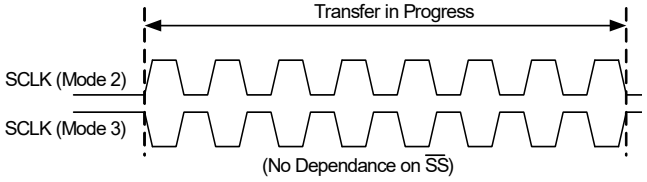


Figure 16-32 illustrates TX data loading in modes 2 and 3. In this case, there is no dependence on $\overline{SS}$ and a transfer in progress is defined to be from the leading edge of the first SCLK to the point at which the RX Buffer register is loaded with the received byte. Loading the shifter by the leading edge of the clock has the effect of providing the required one-half clock setup time, as the data is latched into the receiver on the trailing edge of the SCLK in these modes.

Figure 16-32. Mode 2 and 3 Transfer in Progress



Chained SPIS. When two adjacent communication blocks are chained to form a more-than-8-bit SPIS function, the preceding SPIS operations are maintained the same, with the following exceptions:

- More transits for more bits.
- Only need to enable LSB block to enable the function such as in chained Timer/Counter/CRCPRS/IPWM functions.
- Need to write MSB TX register first and then LSB register to set new data.
- Always read MSB RX data first and then LSB RX data.
- Always check LSB status bits for whole SPIM status if you follow above TX/RX read/write operation sequence.
- The interrupt in both blocks can be enabled and selected arbitrarily. (But if clearing SPI Complete bit or TX Empty bit, still need to read the CR0 register or write DR1 register in that block).

16.3.9 Transmitter Timing

Enable/Disable Operation. As soon as the block is configured for the Transmitter and before enabling, the primary output is set to idle at logic 1, the mark state. The output will remain '1' until the block is enabled and a transmission is initiated. The auxiliary output will also idle to '1', which is the idle state of the associated SPI mode 3 clock.

When the Transmitter is enabled, the internal reset is released on the divide-by-eight clock generator circuit. On the next positive edge of the selected input clock, this 3-bit up counter circuit, which generates the bit clock with the MSb, starts counting up from 00h, and is free-running thereafter.

When the block is disabled, the clock is immediately gated low. All internal state is reset (including CR0 status) to its configuration-specific reset state, except for DR0, DR1, and DR2 which are unaffected.

Transmit Operation. Transmission is initiated with a write to the TX Buffer register (DR1). The CPU write to this register is required to have one-half bit clock setup time for the data, to be recognized at the next positive internal bit clock edge. As shown in Figure 16-33, once the setup time is met, there is one clock of latency until the data is loaded into the shifter and the START bit is generated to the TXD (primary) output.

Figure 16-33. Typical Transmitter Timing

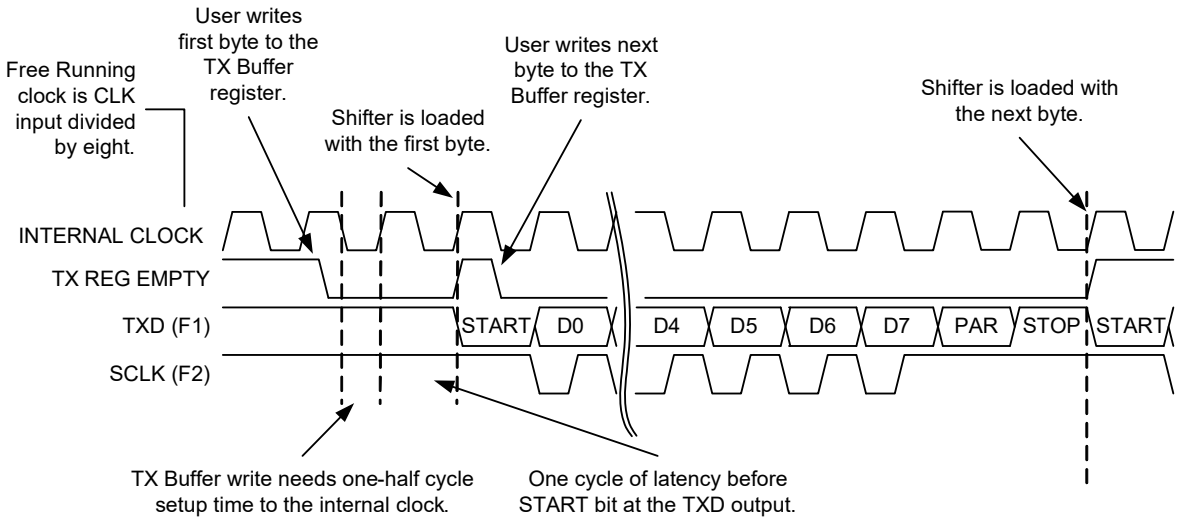
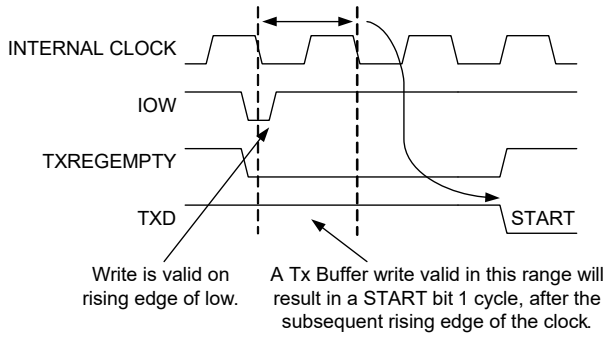


Figure 16-34 shows a detail of the Tx Buffer load timing. The data bits are shifted out on each of the subsequent clocks. Following the eighth bit, if parity is enabled, the parity bit is sent to the output. Finally, the STOP bit is multiplexed into the data stream. With one-half cycle setup to the next clock, if new data is available from the TX Buffer register, the next byte is loaded on the following clock edge and the process is repeated. If no data is available, a mark (logic 1) is output.

Figure 16-34. Tx Buffer Load Timing



The SCLK (auxiliary) output has a SPI mode 3 clock associated with the data bits (for the mode 3 timing, see Figure 16-24). During the mark (idle) and framing bits the SCLK output is high.

Status Generation. There are two status bits in the Transmitter CR0 register: TX Reg Empty and TX Complete.

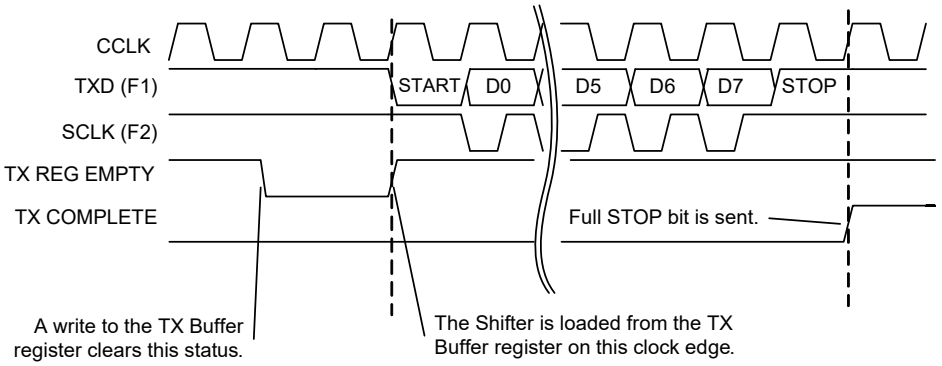
TX Reg Empty indicates that a new byte can be written to the TX Buffer register. When the block is enabled, this status bit is immediately asserted. This status bit is cleared when the user writes a byte of data to the TX Buffer register and set when the data byte in the TX Buffer register is transferred into the shifter. If a transmission is not already in progress, the assertion of this signal initiates one subject to the timing.

The default interrupt in the Transmitter is tied to TX Reg Empty. However, an initial interrupt is not generated when the block is enabled. The user must write an initial byte to the TX Buffer register. That byte must be transferred into the shifter, before interrupts generated from the TX Reg Empty status bit are enabled. This prevents an interrupt from occurring immediately on block enable.

TX Complete is an optional interrupt and is generated when all bits of data and framing bits have been sent. It is cleared on a read of the CR0 register. This signal may be used to determine when it is safe to disable the block after data transmission is complete. In an interrupt driven Transmitter application, if interrupt on TX Complete is selected, the status must be cleared on every interrupt; otherwise, the status will remain high and no subsequent interrupts are logged. See Figure 16-35 for timing relationships.

Status Clear On Read. Refer to the SPIM subsection in “SPIM Timing” on page 323.

Figure 16-35. Status Timing for the Transmitter



16.3.10 Receiver Timing

Enable/Disable Operation. As soon as the block is configured for Receiver and before enabling, the primary output is connected to the data input (RXD). This output will continue to follow the input, regardless of enable state. The auxiliary output will idle to '1', which is the idle state of the associated SPI mode 3 clock.

When the Receiver is enabled, the internal clock generator is held in reset until a START bit is detected on the input. The block must be enabled with a setup time to the first START bit input.

When the block is disabled, the clock is immediately gated low. All internal state is reset (including CR0 status) to its configuration-specific reset state, except for DR0, DR1, and DR2 which are unaffected.

Receive Operation. A clock, which must be eight times the desired baud rate, is selected as the CLK input. This clock is an input to the RX block clock divider. When the receiver is idle, the clock divider is held in reset. As shown in Figure 16-36, reception is initiated when a START bit (logic 0) is detected on the RXD input. When this occurs, the reset is negated to the clock divider and the 3-bit counter starts an up-count. The block clock is derived from the MSb of this counter (corresponding to a count of four), which serves to sample each incoming bit at the nominal center point. This clock also sequences the state machine at the specified bit rate.

The sampled data is registered into an input flip-flop. This flip-flop feeds the DR0 Shift register. Only data bits are shifted into the Shift register.

At the STOP sample point, the block is immediately (within one cycle of the 24 MHz system clock) set back into an idle state. In this way, the clock generation circuit can immediately enable the search for the next START bit, thereby re-synchronizing the bit clock with the incoming bit rate on every new data byte reception. The RX Reg Full status bit, as well as error status, is also set at the STOP sample point.

To facilitate connection to other digital blocks, the RXD input is passed directly to the RXDOUT (primary) output. The SCLK (auxiliary) output has an SPI mode 3 clock associated with the data bits (for mode 3 timing see Figure 16-36). During the mark (idle) and framing bits, the SCLK output is high.

Figure 16-36. Receiver Operation

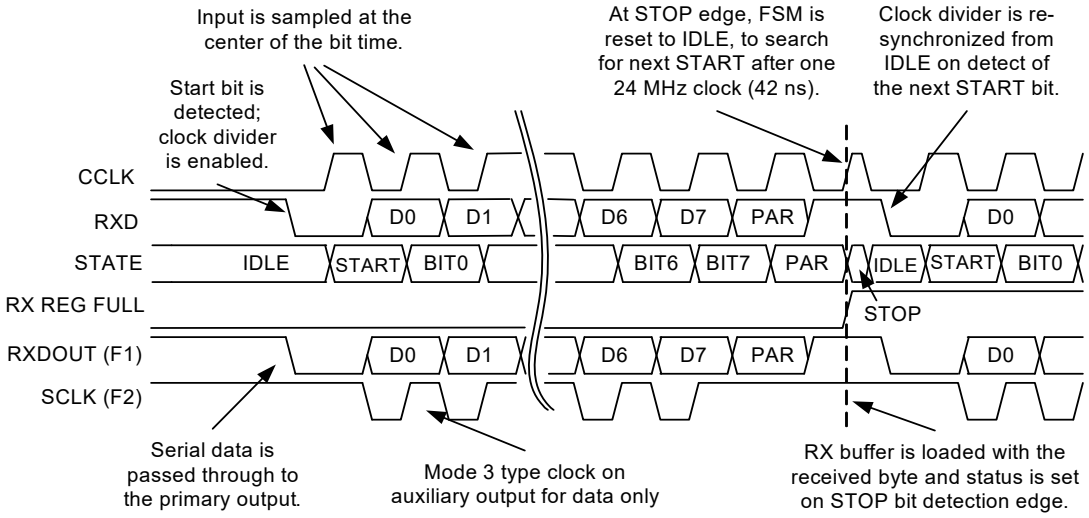
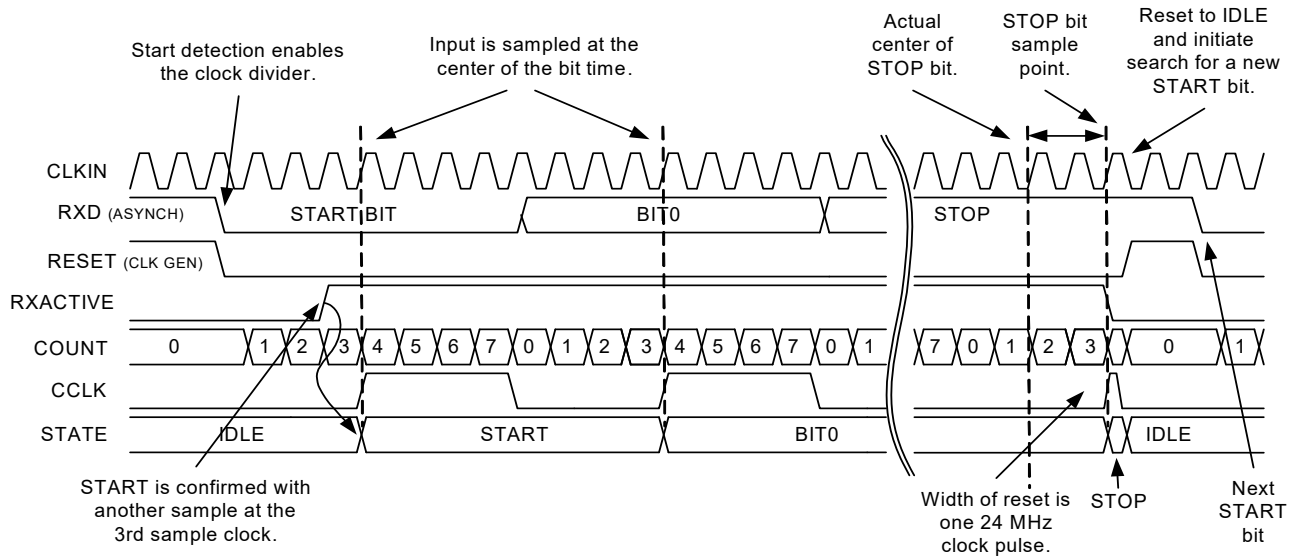


Figure 16-37. Clock Generation and Start Detection



Clock Generation and Start Detection. The input clock selection is a free running, eight times over-sampling clock. This clock is used by the clock divider circuit to generate the block clock at the bit rate. As shown in Figure 16-37, the clock block is derived from the MSb of a 3-bit counter, giving a sample point as near to the center of the bit time as possible. This block clock is used to clock all internal circuits.

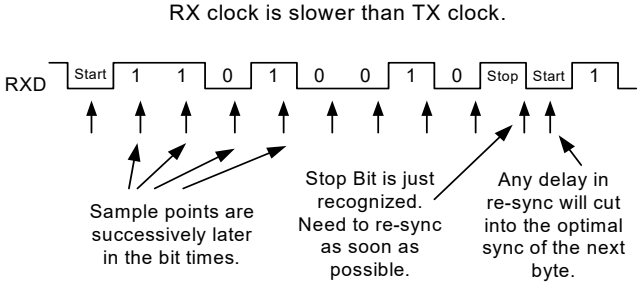
Because the RXD bit rate is asynchronous to the block bit clock, these clocks must be continually re-aligned. This is accomplished with the START bit detection.

When in IDLE state, the clock divider is held in reset. On START (when the input RXD transitions are detected as a logic 0), the reset is negated and the divider is enabled to count at the eight times rate. If the RXD input is still logic 0 after three samples of the input clock, the status RXACTIVE is asserted, which initiates a reception. If this sample of the RXD line is a logic 1, the input '0' transition was assumed to be a false start and the Receiver remains in the idle state.

Figure 16-37 shows that the internal bit clock (CCLK) is running slower than the external TX bit clock and the STOP bit is sampled later than the actual center point. After the STOP bit is sampled, the 24 MHz reset pulse forces the Receiver back to an idle state. In this state, the next START bit search is initiated, resynchronizing the RX bit clock to the TX bit clock.

This resynchronization process (forcing the state back to idle) occurs regardless of the value of the STOP bit sample. It is important to reset as soon as possible, so that maximum performance can be achieved. Figure 16-38 shows an example where the RX block clock bit rate is slower than the external TX bit rate. The sample point shifts to successively later times. In the extreme case shown, the RX samples the STOP bit at the trailing edge. In this case, the receiver has counted 9.5 bit times, while the transmitter has counted 10 bit times. Therefore, for a 10-bit message, the maximum theoretical clock offset, for the message to be received correctly, is represented by one-half bit time or five percent. If the RX and TX clocks exceed this offset, a logic 0 may be sampled for the STOP bit. In this case, the Framing Error status is set.

Figure 16-38. Example RX Resynchronization



This theoretical maximum will be degraded by the resynchronization time, which is fixed at approximately 42 ns. In a typical 115.2 Kbaud example, the bit time is 8.70 µs. In this case the new maximum offset is:

((4.35 ms - 42 ns) / 4.35 ms) x 5% or 4.95%

At slower baud rates, this value gets closer to the theoretical maximum of five percent.

Status Generation. There are five status bits in a Receiver block: RX Reg Full, RX Active, Framing Error, Overrun, and Parity Error. All status bits, except RX Active and Overrun, are set synchronously on the STOP bit sample point.

RX Reg Full indicates a byte has been received and transferred into the RX Buffer register. This status bit is cleared when the user reads the RX Buffer register (DR2). The setting of this bit is synchronized to the STOP sample point. This is the earliest point at which the Framing Error status can be set; and therefore, error status is defined to be valid when RX Reg Full is set.

RX Active can be polled to determine if a reception is in progress. This bit is set on START detection and cleared on STOP detection. This bit is not ***sticky*** and there is no way for the user to clear it.

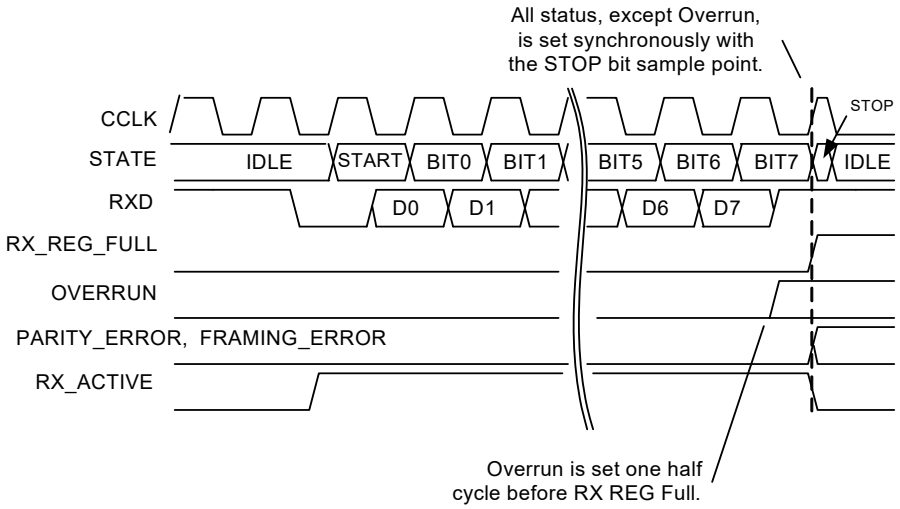
Framing Error status indicates that the STOP bit associated with a given byte was not received correctly (expecting a '1', but received a '0'). This will typically occur when the difference between the baud rates of the transmitter and receiver is greater than the maximum allowed.

Overrun occurs when there is a received data byte in the RX Buffer register and a new byte is loaded into the RX Buffer register, before the user has had a chance to read the previous one. Because the RX Buffer register is actually a latch, Overrun status is set one-half cycle before RX Reg Full. This means that although the new data is not available, the previous data has been overwritten because the latch was opened.

Parity Error status indicates that resulting parity calculation on the received byte does not match the value of the parity bit that was transmitted. This status is set on the sample point of the STOP signal.

Status Clear On Read. Refer to the SPIM subsection in "SPIM Timing" on page 323.

Figure 16-39. Status Timing for Receiver



16.3.11 DSM Timing

Enable/Disable Operation. The enable/disable timing is similar to Timer/Counter but looser because it is only single block function.

KILL Operation. DSM supports two KILL modes: KILL-Disable and KILL_Async. KILL-Disable is same as Timer/Counter KILL-Disable. KILL-Async is same as Dead Band KILL-Async mode.

Section E: Analog System

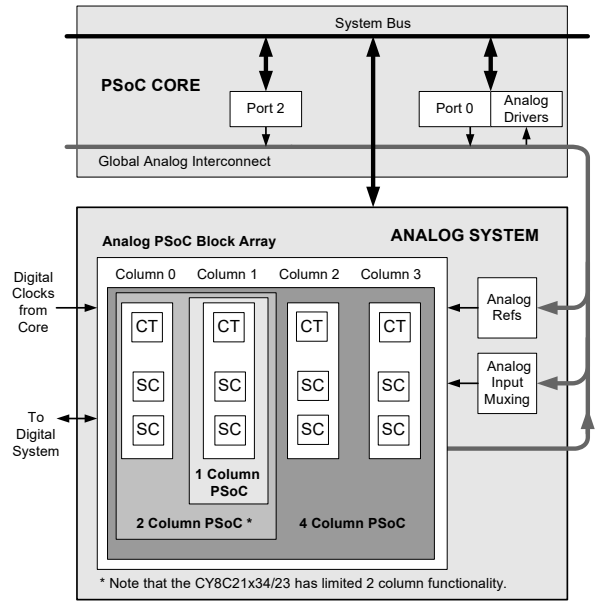
The configurable Analog System section discusses the analog components of the PSoC device and the registers associated with those components. This section encompasses the following chapters:

- Two Column Limited Analog System on page 338
- Two Column Analog Compare System chapter on page 358

Top-Level Analog Architecture

The figure below displays the top-level architecture of the PSoC's analog system. With the exception of the analog drivers, each component of the figure is discussed at length in this section.

Figure 17-1. PSoC Analog System



Interpreting the Analog Documentation

Information in this section covers the CY8C22x45 and CY8C21345 PSoC devices. The primary analog distinction between these devices and other PSoC1 devices is the number of analog columns: 1, 2, or 4 columns. The following table lists the resources available for specific device groups. While reading the analog system section, determine and keep in mind the number of analog columns that are in your device, to accurately interpret the documentation.

PSoC Device Characteristics

PSoC Part Number	Digital I/O (max)	Digital Rows	Digital Blocks	Analog Inputs	Analog Outputs	Analog Columns	Analog Blocks
CY8C29x66	64	4	16	**12**	**4**	**4**	**12**
CY8C27x43	44	2	8	**12**	**4**	**4**	**12**
CY8C24x94	50	1	4	**48**	**2**	**2**	**6**
CY8C24x23	24	1	4	**12**	**2**	**2**	**6**
CY8C24x23A	24	1	4	**12**	**2**	**2**	**6**
CY8C22x45	38	2	8	**10**	**0**	**4**	**6**
CY8C22x13	16	1	4	**8**	**1**	**1**	**3**
CY8C21345	24	1	4	**10**	**0**	**4**	**6**
CY8C21x34	28	1	4	**28**	**0**	**2**	**4 ***
CY8C21x23	16	1	4	**8**	**0**	**2**	**4 ***
CY7C64215	50	1	4	**48**	**2**	**2**	**6**
CY7C603xx	28	1	4	**28**	**0**	**2**	**4 ***
CYWUSB6953	28	1	4	**28**	**0**	**2**	**4 ***

* Limited analog functionality.

Application Description

PSoC blocks are user configurable system resources. On-chip analog PSoC blocks reduce the need for many MCU part types and external peripheral components. Analog PSoC blocks are configured to provide a wide variety of peripheral functions. The *PSoC Designer Software Integrated Development Environment* provides automated configuration of PSoC blocks by selecting the desired functions. PSoC Designer then generates the proper configuration information and prints a device data sheet unique to that configuration.

A precision internal voltage reference provides accurate analog comparisons. A temperature sensor input is provided to the analog PSoC block array, supporting applications such as battery chargers and data acquisition, without requiring external components.

Defining the Analog Blocks

There are three analog PSoC block types: Continuous Time (CT) blocks, and Type C and Type D Switch Capacitor (SC) blocks. CT blocks provide continuous time analog functions. SC blocks provide switched capacitor analog functions.

Each of the analog blocks has many potential inputs and several outputs. The inputs to these blocks include ***analog signals*** from external sources, intrinsic analog signals driven from neighboring analog blocks, or various voltage reference sources.

The analog blocks are organized into columns. Each column contains one Continuous Time (CT) block, Type E (ACE); one Switched Capacitor (SC) block, Type C (ASC); and one Switched Capacitor block, Type D (ASD). However, the number of analog columns in a specific part can either be 1, 2, or 4 columns. To determine the number of columns in your PSoC device, refer to the PSoC Device Characteristics table at the beginning of this section.

Note The CT and SC blocks in the CY8C22x45, CY8C21345, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 devices have limited functionality compared to all other PSoC devices. Analog columns in this device contain one CT block, type E (ACE), and one SC block, type E (ASE). Refer to the Two Column Limited Analog System chapter on page 338.

The blocks in a particular column all run off the same clocking source. The blocks in a column also share some output bus resources.

There are three types of outputs from each analog block and additional two discrete outputs in the Continuous Time blocks.

1. The analog output bus (ABUS) is an analog bus resource that is shared by all of the analog blocks in a column. Only one block in a column can actively drive this bus at any one time, with the user having control of this output through register settings. This is the only analog output that can be driven directly to a pin.
2. The comparator bus (CBUS) is a digital bus resource that is shared by all of the analog blocks in a column. Only one block in a column can be actively driving this bus at any one time, with the user having control of this output through register settings.
3. The local outputs (OUT, GOUT, and LOUT in the Continuous Time blocks) are routed to neighbor blocks. The various input ***multiplexer (mux)*** connections (NMux, PMux, RBotMux, AMux, BMux, and CMux) all use the output bus from one block as their input.

Analog Functionality

The following is a sampling of the functions that operate within the capability of the analog PSoC blocks, using one analog PSoC block, multiple analog blocks, a combination of more than one *type* of analog block, or a combination of analog and digital PSoC blocks. Most of these functions are currently available as ***user modules*** in *PSoC Designer*. Others will be added in the future. Refer to the *PSoC Designer* software for additional information and the most up-to-date list of user modules.

- Delta-Sigma Analog-to-Digital Converters
- Successive Approximation Analog-to-Digital Converters
- Incremental Analog-to-Digital Converters
- Digital to Analog Converters
- Programmable Gain/Loss Stage
- Analog Comparators
- Zero-Crossing Detectors
- Sample and Hold
- Low-Pass Filter
- Band-Pass Filter
- Notch Filter
- Amplitude Modulators
- Amplitude Demodulators
- Sine-Wave Generators
- Sine-Wave Detectors
- Sideband Detection
- Sideband Stripping
- Temperature Sensor
- Audio Output Drive
- DTMF Generator
- FSK Modulator
- Embedded Modem

By modifying registers, as described in this document, users can configure PSoC blocks to perform these functions and more. The philosophy of the analog functions supplied is as follows.

- Cost effective, single-ended configuration for reasonable speed and accuracy, providing a simple interface to most real-world analog inputs and outputs.
- Flexible, System-on-Chip programmability, providing variations in functions.
- Function specific, easily selected trade-offs of accuracy and resolution with speed, resources (number of analog blocks), and power dissipated for that application.

Analog Register Summary

The table below lists the CY8C22x45 and CY8C21345 PSoC registers for the analog system in address order (Add. column) within their system resource configuration. The bits that are grayed out are reserved bits. If these bits are written, they should always be written with a value of '0'. The naming conventions for the SC and CT registers and their arrays of PSoC blocks are detailed in their respective table title rows.

Note that all PSoC devices, with a base part number of CY8C2xxxx (except for the CY8C25122 and CY8C26xxx PSoC devices), fall into one of the following categories with respect to their analog PSoC arrays: 4 column device, 2 column device, or 1 column device. It also applies to CY7C64215, CY7C603xx, and CYWUSB6953. The "PSoC Analog System Block Diagram" at the beginning of this section illustrates this. However, there is one modification to this rule: The CY8C22x45, CY8C21345, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices have two column limited functionality. The CY8C22x45, CY8C21345, CY8C21x34, CY7C603xx, and CYWUSB6953 analog registers are summarized and described in the I/O Analog Multiplexer chapter on page 422 and the CY8C22x45, CY8C21345, and CY8C21x23 analog registers are summarized and described in the Two Column Limited Analog System chapter on page 338. Additionally, the CY8C22x45 and CY8C21345 analog registers have a dedicated Two Column Analog Compare System chapter on page 358.

In the table below, the third column from the left titled "Analog Columns" indicates which of the three PSoC device categories the register falls into. To determine the number of analog columns in your PSoC device, refer to the table titled "PSoC Device Characteristics" on page 334.

Summary Table of the Analog Registers

Add.	Name	Analog Cols.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
ANALOG INTERFACE REGISTERS											
0,64h	CMP_CR0	2			COMP[1:0]				AINT[1:0]		# : 00
0,66h	CMP_CR1	2			CLDIS[1]	CLDIS[0]					RW : 00
0,E6h	DEC_CR0	4, 2			IGEN[1:0]		ICLKS0				RW : 00
0,E7h	DEC_CR1	4				ICLKS2	ICLKS1				RW : 00
1,58h	CSCMPCR0		CSSEL1[2]	CSSEL0[2]	CSCMP[1:0]		CSIGEN[1:0]		CSCLKDIS[1:0]		RW : 00
1,59h	CSCMPGOEN		GOO7	GOO3	CSSEL1[1:0]		GOO6	GOO2	CSSEL0[1:0]		RW : 00
1,5Ah	CSLUTCR0		CSLUT1[3:0]				CSLUT0[3:0]				RW : 00
1,5Ch	CMPPWMCR		ACMPS0[1:0]		PWMDLY_EN[1]	PWMCLK_SEL1[1:0]		PWMDLY_EN[0]	PWMCLK_SEL0[1:0]		RW : 00
1,5Dh	CMPFLTCR		FLT1_WD[1:0]		FLT1_CSL	FLT1_EN	FLT0_WD[1:0]		FLT0_CSL	FLT0_EN	RW : 00
1,5Eh	CMPCLK1		RS_SEL[1]	SYS1	DIVCLK1		RS_SEL[0]	SYS0	DIVCLK0		RW : 00
1,5Fh	CMPCLK0		RS_EN		SEL_CSD[1:0]		CMPCOL1[1:0]		CMPCOL0[1:0]		RW : 00
1,60h	CLK_CR0	2					AColumn1[1:0]		AColumn0[1:0]		RW : 00
1,61h	CLK_CR1	4, 2			ACLK1[2:0]			ACLK0[2:0]			RW : 00
1,63h	AMD_CR0	2					AMOD0[3:0]				RW : 00
1,64h	CMP_GO_EN	2	GOO5	GOO1	SEL1[1:0]		GOO4	GOO0	SEL0[1:0]		RW : 00
1,66h	AMD_CR1	2					AMOD1[3:0]				RW : 00
1,67h	ALT_CR0	4, 2	LUT1[3:0]				LUT0[3:0]				RW : 00
ANALOG INPUT CONFIGURATION REGISTERS											
1,5Bh	CMPCOLMUX		ACMPS1[1:0]		LMXR	RMXL	ACI1[1:0]		ACI0[1:0]		RW : 00
CONTINUOUS TIME PSoC BLOCK, TYPE E, REGISTERS											
x,72h	ACE00CR1			CompBus	NMux[2:0]			PMux[2:0]			RW : 00
x,73h	ACE00CR2								FullRange	PWR	RW : 00
x,76h	ACE01CR1			CompBus	NMux[2:0]			PMux[2:0]			RW : 00
x,77h	ACE01CR2								FullRange	PWR	RW : 00
1,50h	CMP0CR1		BINC	CompBus	NMux[2:0]			PMux[2:0]			RW : 00
1,51h	CMP0CR2								FullRange	PWR	RW : 00
1,54h	CMP1CR1		BINC	CompBus	NMux[2:0]			PMux[2:0]			RW : 00
1,55h	CMP1CR2								FullRange	PWR	RW : 00

Summary Table of the Analog Registers *(continued)*

Add.	Name	Analog Cols.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
SWITCHED CAPACITOR PSoC BLOCK, TYPE E, REGISTERS											
x,80h	ASE10CR0		FVal								RW : 00
x,84h	ASE11CR0		FVal								RW : 00
VDAC5 BLOCK REGISTERS											
1,53h	VDAC50CR0					VDACIN[4:0]					RW : 00
1,57h	VDAC51CR0					VDACIN[4:0]					RW : 00
1,E7h	VDAC_TRIM			EN	V24	SAMPLE	TRIM[3:0]				RW : 00

LEGEND
x An “x” before the comma in the address field indicates that this register can be accessed or written to no matter what bank is used.
Access is bit specific. Refer to the Register Details chapter on page 103 for additional information.
R Read register or bit(s).
W Write register or bit(s).

17. Two Column Limited Analog

This chapter explains the Two Column Limited Analog System PSoC devices and their associated registers. It details the entire analog system for two column limited functionality, including the analog interface, analog array, analog input configuration, analog reference, CT and SC blocks. For a complete table of the Two Column Limited Analog System registers, refer to the "Summary Table for 2 Column Limited Analog System Registers" on page 349. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

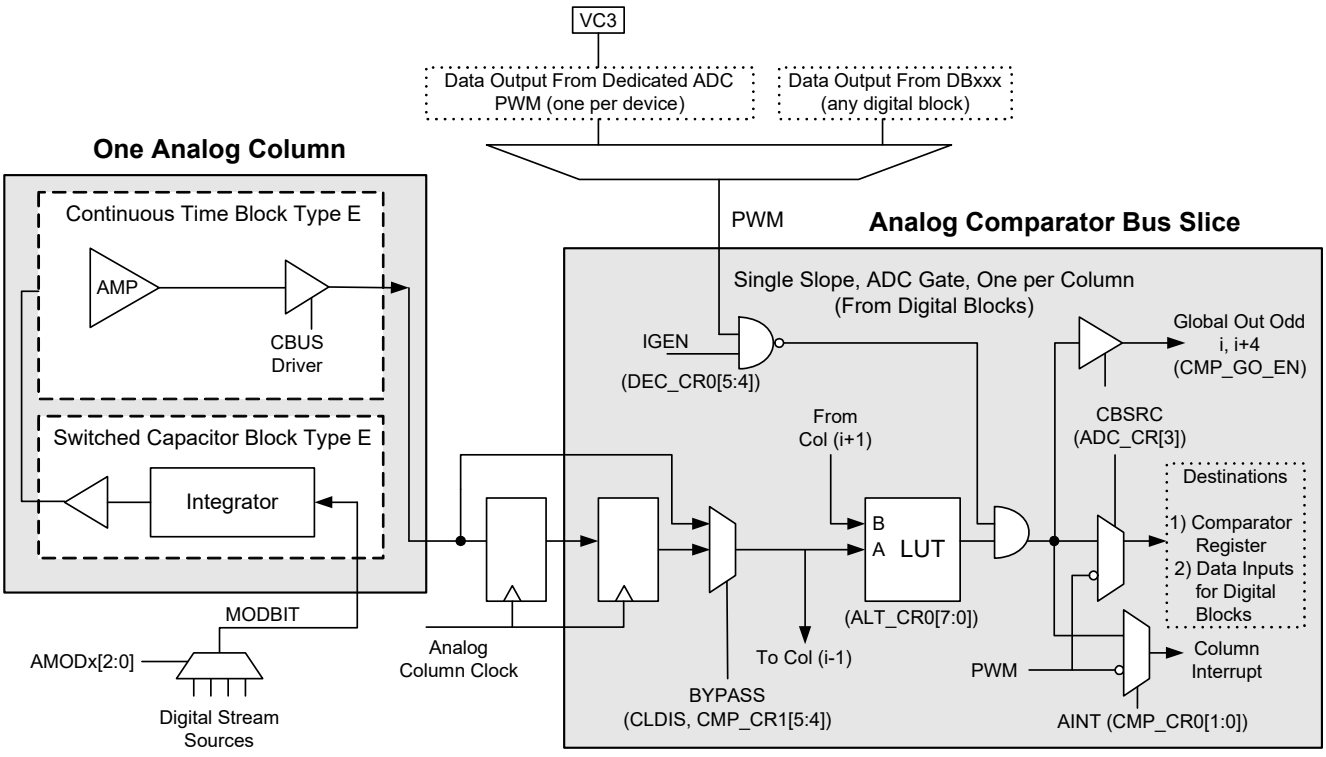
Unique to the CY8C22x45, CY8C21345, CY8C21x34, CY7C603xx, and CYWUSB6953 PSoC devices is the use of an I/O analog multiplexer system resource. The I/O Analog Multiplexer is described in the I/O Analog Multiplexer chapter on page 422. A summary of the I/O Analog Multiplexer registers are located in the section called "System Resources" on page 373.

17.1 Architectural Description

17.1.1 Analog Interface

Figure 17-1 displays the top-level diagram of the PSoC devices' analog interface system.

Figure 17-1. Analog Comparator Bus Slice of the CY8C22x45, CY8C21345, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC Devices



17.1.1.1 Analog Comparator Bus Interface

Each analog column has a dedicated comparator bus associated with it. In the CY8C22x45, CY8C21345, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices, only the Continuous Time (CT) block can drive this bus. The output on the comparator bus can drive into the digital blocks as a data input. It also serves as an input to Switched Capacitor (SC) blocks as an interrupt input, and is available as read only data in the Analog Comparator Control register (CMP_CR0). It can be driven to the global output bus by way of the Comparator to Global Output Enable register (CMP_GO_EN).

Figure 17-1 illustrates one column of the comparator bus. The comparator bus is synchronized by the selected column clock before it is available, to either drive the digital blocks, interrupt, SC blocks, or for it to be read in the CMP_CR0 register. There is also an option to bypass the synchronization in each column into a transparent mode by setting bits in the CMP_CR1 register.

As shown in Figure 17-1, the comparator bus output is gated by the primary output of a selected digital block. This feature is used to precisely control the conversion period of a single slope ADC. Any digital block can be used to drive the gate signal. This selection may be made with the ICLKS bits in registers DEC_CR0 and DEC_CR1. This function may be enabled on a column-by-column basis, by setting the IGEN bits in the DEC_CR0 register. Alternately, the dedicated ADC PWM, with VC3 as input, can be used to gate the ADC conversion period without the need for a digital block. When this dedicated PWM is configured, it overrides the ICLKS selection as defined by the DEC_CR0 and DEC_CR1 registers.

The analog comparator bus output values can be modified or combined with another analog comparator bus through the Analog Look-Up-Table (LUT) function. The LUT takes two inputs, A and B, and provides a selection of 16 possible logic functions for those inputs. The LUT A and B inputs for each column comparator output is shown in the following table.

Table 17-1. A and B Inputs for Each Column Comparator LUT Output for the CY8C22x45, CY8C21345, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 Devices

Comparator LUT Output	A	B
Column 0	ACMP0	ACMP1
Column 1	ACMP1	0

The LUT configuration is set in one control register, ALT_CR0. Each selection for each column is encoded in four bits. The function value corresponding to the bit encoding is shown in the following table.

Table 17-2. ALT_CR0 Register

LUTx[3:0]	0h: 0000: FALSE 1h: 0001: A .AND. $\overline{B}$ 2h: 0010: A .AND. $\overline{B}$ 3h: 0011: $\overline{A}$ 4h: 0100: $\overline{A}$ .AND. B 5h: 0101: B 6h: 0110: A .XOR. B 7h: 0111: A .OR. B 8h: 1000: A .NOR. B 9h: 1001: $\overline{A}$.XNOR. B Ah: 1010: $\overline{B}$ Bh: 1011: $\overline{A}$ .OR. $\overline{B}$ Ch: 1100: $\overline{A}$ Dh: 1101: $\overline{A}$ .OR. B Eh: 1110: A. NAND. B Fh: 1111: TRUE

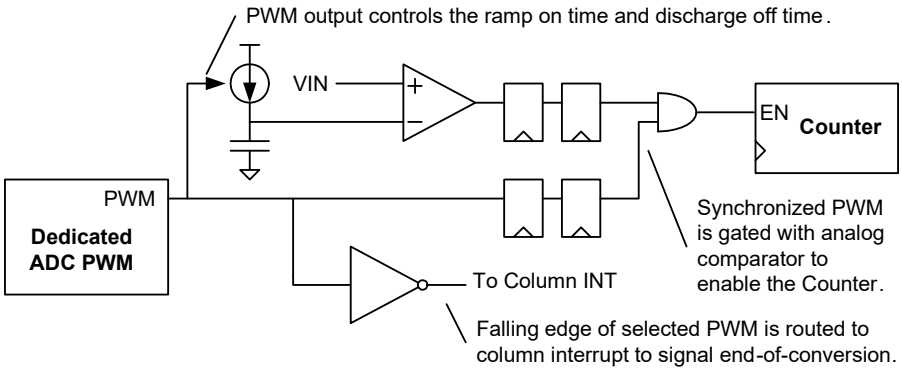
17.1.1.2 Analog Column Clock Generation

The input clock source for each column clock generator is selectable according to the CLK_CR0 register. There are four selections for each column: VC1, VC2, ACLK0, and ACLK1. An additional selection, SYSCLK, is controlled by the CLK_CR3 register. The VC1 and VC2 clock signals are global system clocks. Programming options for these system clocks can be accessed in the OSC_CR1 register. Each of the ACLK0 and ACLK1 clock selections are driven by a selection of digital block outputs. The settings for the digital block selection are located in the CLK_CR1 register. The CLK_CR3 register has additional column clock options. This register allows for a direct SYSCLK option as well as the option to divide the selected column clock by 2, 4, or 8.

17.1.1.3 Single Slope ADC

A simplified block diagram of the single slope ADC (SSADC) implementation is show in Figure 17-2. The core of the conversion algorithm involves a current source, an integrating capacitor, and a comparator. When the current source is activated, a linear voltage ramp is generated on the capacitor. This voltage is an input to an analog comparator circuit; the other input of which is the analog input voltage to be converted. With the polarity of hookup as shown, the comparator will be high until the ramp voltage equals the input voltage, at which time it will transition low. A counter gate is generated by the AND of the PWM high time (which defines the start of the ramp) and the comparator (which defines the trip point or the end of the conversion for a given voltage). When the conversion is complete, the code may be read from the counter. Each column has an ADC configuration register (ADCx_CR).

Figure 17-2. Single Slope ADC Block Diagram

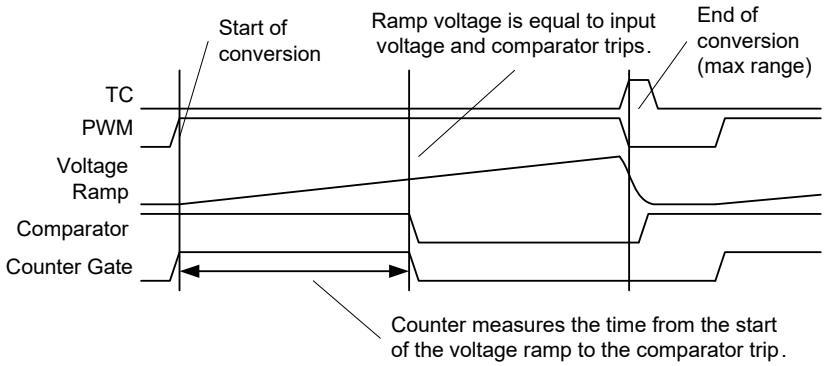


In order to interface the asynchronous analog comparator to the digital block array, a double synchronization is required. As shown in Figure 17-2, the PWM is also delayed to align with the valid comparator output.

The basic conversion waveforms are shown in Figure 17-3. The high time of the PWM is set so that the counter will count to a full-scale value. For example, for 8-bit resolution, the high time of the PWM would correspond to 255 (or 256) counter clocks. The low time of the PWM is designed to allow the capacitor to discharge. When a PWM is used for continuous conversions, the Terminal Count of the PWM can be used as a consistent interrupt to read the result of the previous conversion. If only a single conversion is desired, the comparator trip point can be used as an interrupt to signal the end of conversion.

A trim register (ADCx_TR) is provided for each column. The converter must be calibrated for a given maximum voltage, resolution, and frequency of operation before use.

Figure 17-3. Basic ADC Waveforms



17.1.1.4 PWM ADC Interface

The analog interface provides hardware support and signal routing for ***analog-to-digital (ADC)*** conversion functions, specifically the single slope ADC. The control signals for this interface are split between three registers: DEC_CR0, DEC_CR1, and PWM_CR.

The analog interface has support for the single slope ADC operation through the ability to gate the analog comparator outputs. This gating function is required in order to precisely control the digital integration period that is performed in a digital block as part of the function. A digital block PWM or the dedicated ADC PWM may be used as a source to provide the gate signal. Only one source for the gating signal can be selected. However, the gating can be applied independently to any of the column comparator outputs.

The CY8C22x45, CY8C21345, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 devices contain a dedicated block that can perform this PWM gating function using VC3. The VC3 signal, out of the VC3 divider block, can be further divided to provide for gating the incremental ADC.

The PWM_CR register controls the duty cycle selection in terms of VC3 periods, as shown in the following tables. When enabled, the PWM block becomes the source for the incremental gating, overriding the digital block selection.

Table 17-3. PWM High Time

HI[2:0]	Description
000b	Block is not selected, input to incremental gate is from selected digital block.
001b	High time is 1 VC3 period.
010b	High time is 2 VC3 period.
011b	High time is 4 VC3 period.
100b	High time is 8 VC3 period.
101b	High time is 16 VC3 period.

Table 17-4. PWM Low Time

LO[1:0]	Description
00b	No low time. Comparator gate is continually high.
01b	Low time is one VC3 period.
10b	Low time is two VC3 period.
11b	Low time is three VC3 period.

As an alternative to the PWM, the ICLKS bits, which are split between the DEC_CR0 and DEC_CR1 registers, may be used to select a digital block source for the incremental gating signal. Regardless of the source of the gating, the two IGEN bits are used to independently enable the gating function on a column-by-column basis.

17.1.1.5 Analog Modulator Interface (Mod Bits)

The Analog Modulator Interface provides a selection of signals that are routed to either of the two analog array modulation control signals. There is one modulation control signal for Switched Capacitor block. There are nine selections, which include the dedicated reference voltage generator PWM output, the analog comparator bus outputs (include CS comparator output), two global outputs, and a digital block broadcast bus. The selections for all columns are contained in the AMD_CR0 and AMD_CR1 registers.

One use of the modulator interface is to provide a selectable reference to one of the comparator inputs. This can be done by configuring a digital block as a PWM or PRS output with the desired duty cycle. The SC block will then give a low-pass filtered version of this signal, which will be a DC voltage relative to the supply with some ripple.

17.1.1.6 Sample and Hold Feature

Sample and Hold capability can be selected for improved analog-to-digital conversion accuracy. This is done by setting the SHEN bit in the ADCx_CR register.

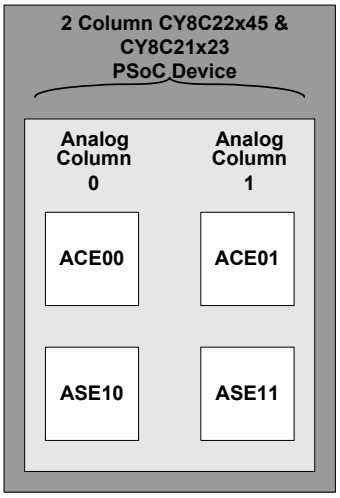
When enabled, this feature works in conjunction with the selected SSADC PWM input. During the PWM high time, the conversion is active and the sample and hold is in “hold” mode. During the PWM low time, the conversion is inactive, and the sample and hold circuit is in “sample” mode.

17.1.2 Analog Array

The analog array is designed to allow moving between families without modifying projects, except for resource limitations. The CY8C22x45, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices have limited analog array functionality. The only analog array connections available to the CY8C22x45, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 are the NMux and PMux connections. Figure 17-4 displays the analog arrays for the CY8C22x45, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 devices, containing the type E continuous time blocks (ACE) and the type E switched capacitor blocks (ASE). Each analog column has 2 analog blocks associated with it. The figures that follow illustrate the analog multiplexer (mux) connections.

Each analog column has a dedicated comparator bus associated with it. Only the CT block in each column can drive this bus. When the CT block is not configured as a comparator, a zero is driven to the comparator block. Refer to the ACExxCR1 register on page 137 and the "Analog Comparator Bus Interface" on page 339 in the Analog Interface section for more information.

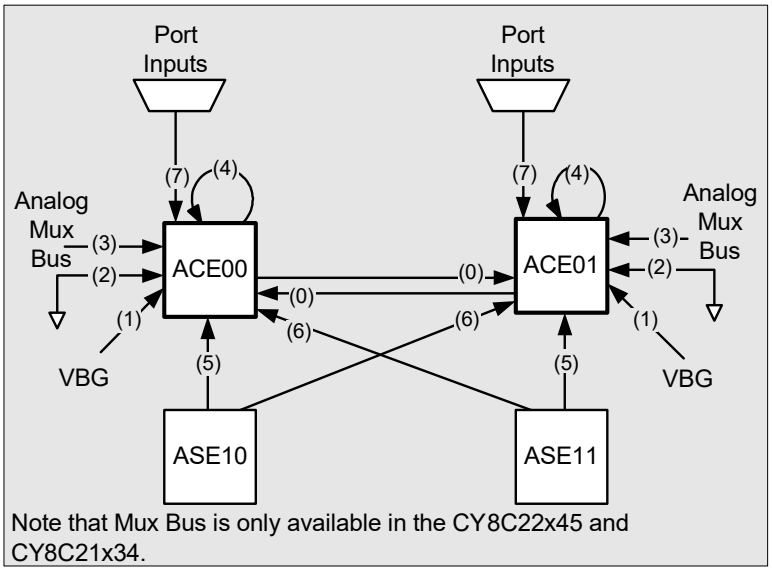
Figure 17-4. Array of Analog PSoC Blocks for the CY8C22x45, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC Devices



17.1.2.1 NMux Connections

The NMux is an 8-to-1 mux which determines the source for the inverting (also called negative) input of Continuous Time (CT) PSoC blocks. These blocks are named ACE00 and ACE01 in the CY8C22x45, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices. More details on the CT PSoC blocks are available in this chapter, in the section titled "Continuous Time PSoC Block" on page 347. The NMux connections are described in detail in the ACExxCR1 register on page 137, bits NMux[2:0]. The numbers in Figure 17-5, which are associated with each arrow, are the corresponding NMux select line values for the data in the NMux portion of the register. The call out names in the figure show nets selected for each NMux value.

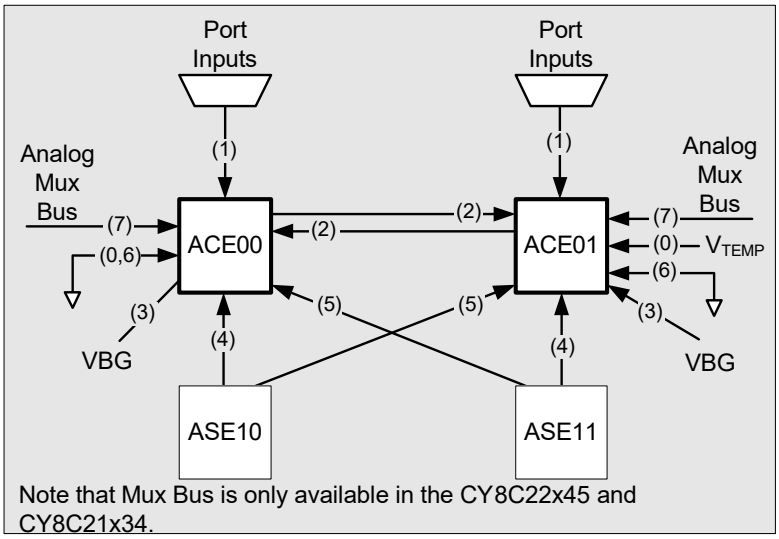
Figure 17-5. NMux Connections for the CY8C22x45, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC Devices



17.1.2.2 PMux Connections

The PMux is an 8-to-1 mux which determines the source for the non-inverting (also called positive) input of CT PSoC blocks (ACE00 and ACE01). More details on the CT PSoC blocks are available in this chapter, in the section titled “Continuous Time PSoC Block” on page 347. The PMux connections are described in detail in the ACExxCR1 register on page 137, bits PMux[2:0]. The numbers in Figure 17-6, which are associated with each arrow, are the corresponding PMux select line values for the data in the PMux portion of the register. The call out names in the figure show nets selected for each PMux value.

Figure 17-6. PMux Connections for the CY8C22x45, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC Devices



17.1.2.3 Temperature Sensing Capability

A temperature-sensitive voltage, derived from the bandgap sensing on the die, is buffered and available as an analog input into the continuous time block ACE01. Temperature sensing allows protection of device operating ranges for fail-safe applications. Temperature sensing, combined with a long sleep timer interval (to allow the die to approximate ***ambient temperature***), can give an approximate ambient temperature for data acquisition and battery charging applications. The user may also calibrate the internal temperature rise based on a known current consumption.

The temperature sensor input to the ACE01 block is labeled V_{TEMP} .

17.1.3 Analog Input Configuration

Figure 17-8 and Figure 17-9 show the analog input configuration for the CY8C22x45, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices, respectively. For a detailed description of the I/O analog multiplexer functionality illustrated in Figure 17-8, refer to the I/O Analog Multiplexer chapter on page 422.

The input multiplexer (mux) maps device inputs (package pins) to analog array columns, based on bit values in the AMX_IN and ABF_CR0 registers. Column 0 is fed by one 4-to-1 mux; column 1 is fed by one of two 4-to-1 muxes. The muxes are CMOS switches with typical resistances in the range of 2K ohms.

Figure 17-7. Limited Two Column Analog Interconnect

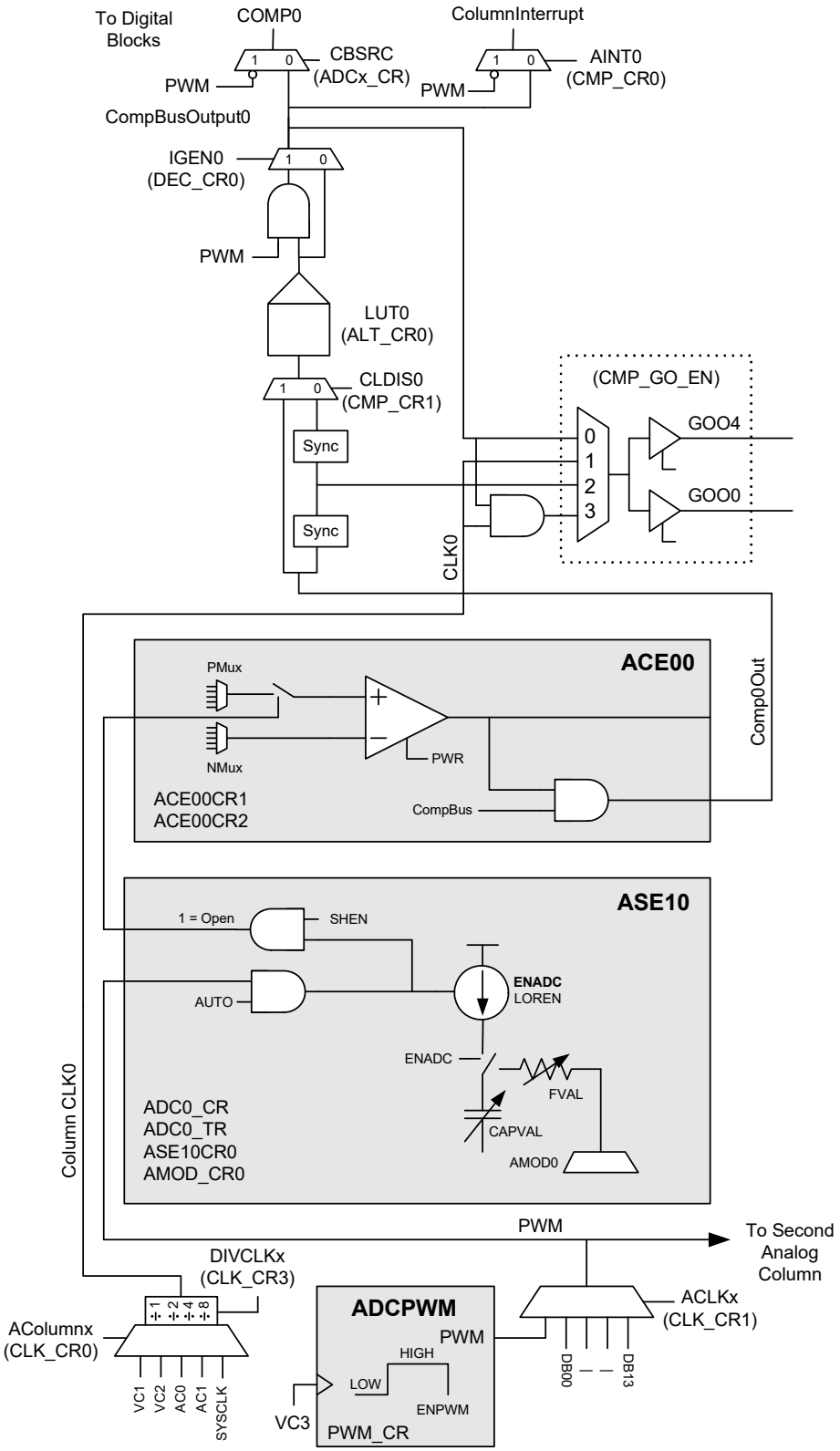


Figure 17-8. Two Column Limited Analog Pin Block Diagram for the CY8C22x45, CY8C21x34, CY7C603xx, and CYWUSB6953

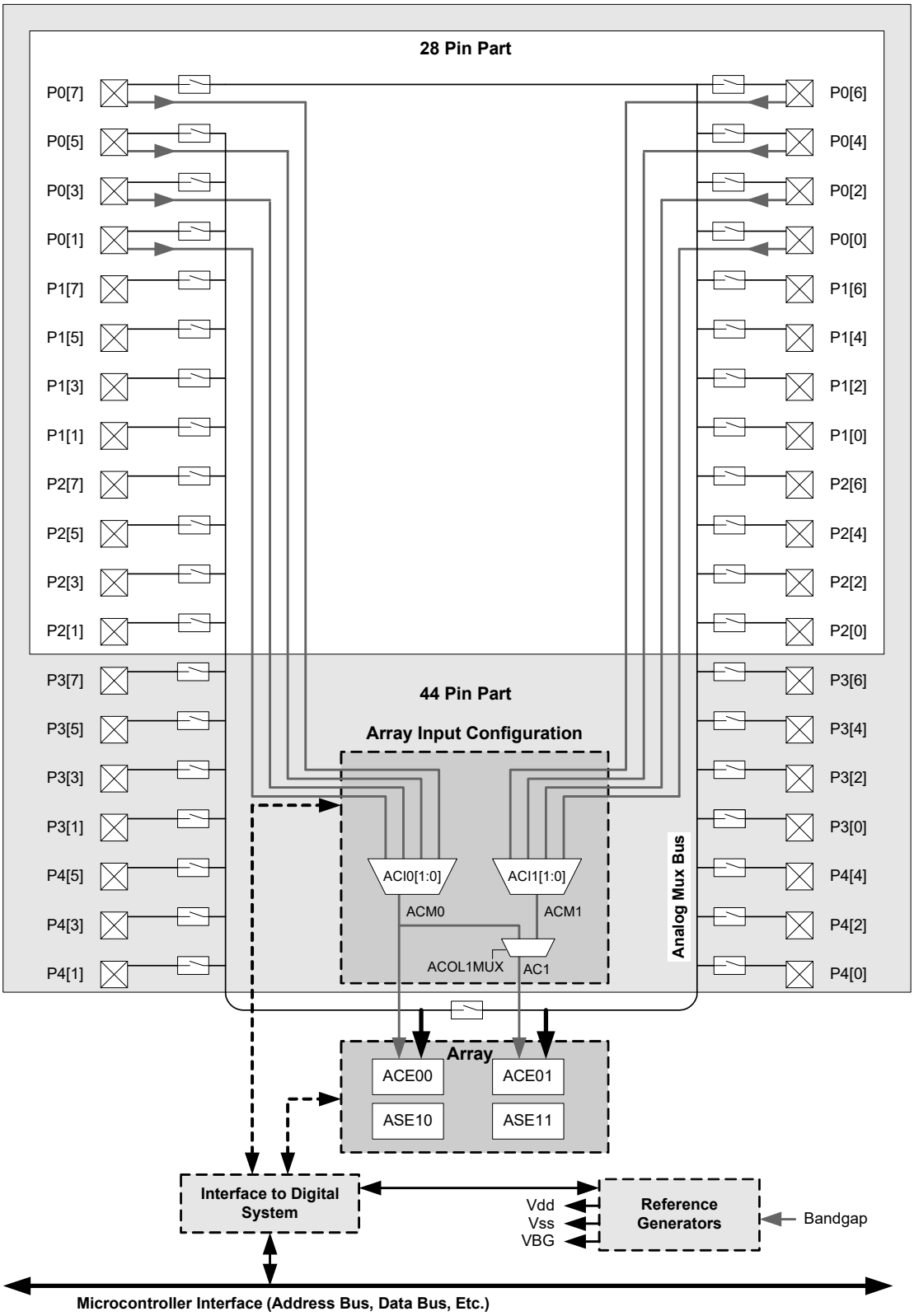
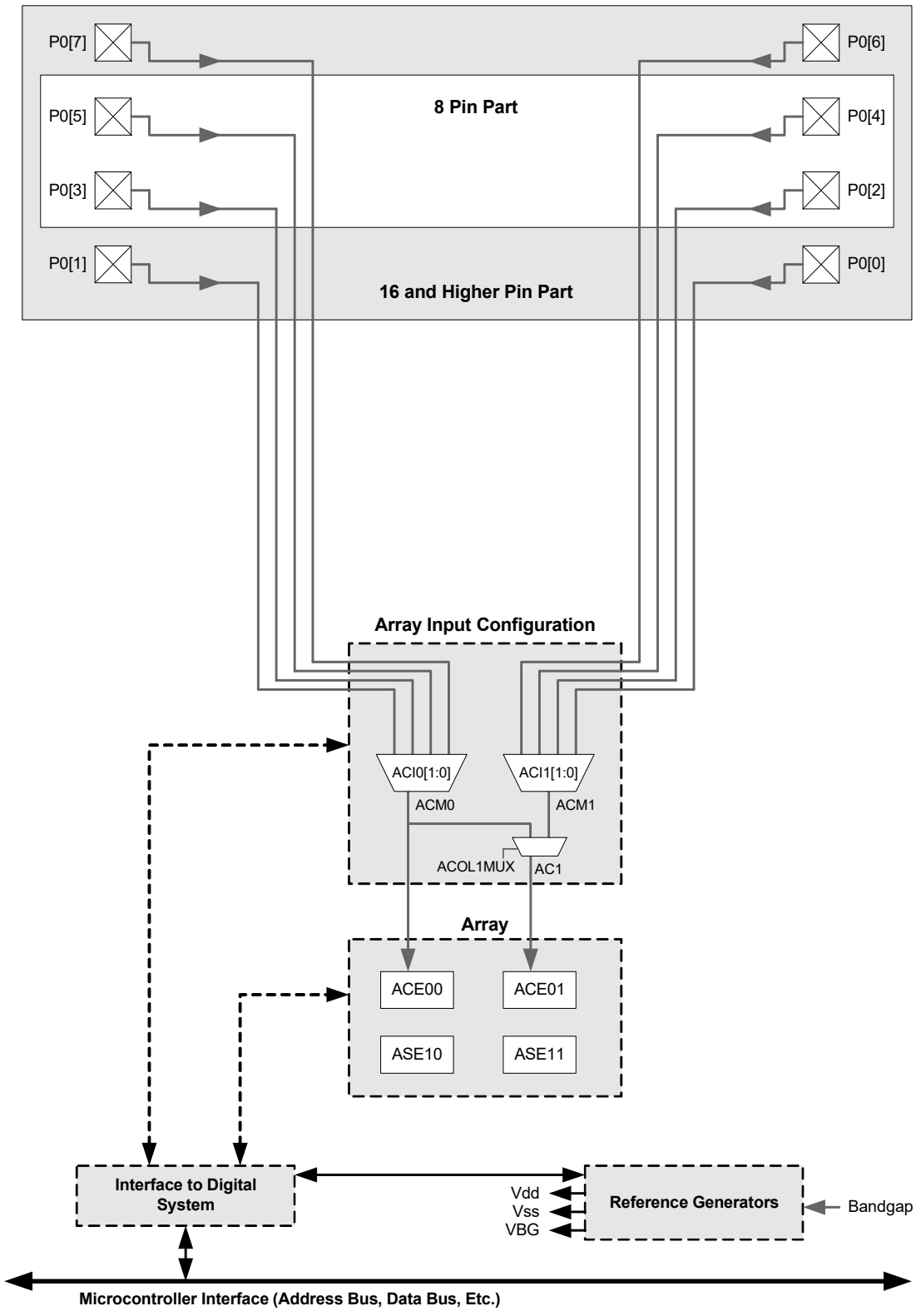


Figure 17-9. Two Column Limited Analog Pin Block Diagram for the CY8C22x45, CY8C21345, and CY8C21x23



17.1.4 Analog Reference

The PSoC device is a single supply part, with no negative voltage available or applicable. The CY8C22x45, CY8C21345, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices support only one analog reference, which is the bandgap voltage VBG. This voltage is routed to the CT blocks in each analog column. VBG is available at both positive and negative inputs of each CT amplifier.

DAC functions are relative to the power supply range (Vss to Vdd). The bandgap VBG reference can be used to calibrate the supply range. Single slope ADC operation relies on a calibration step, using the internal bandgap reference or other user-supplied reference. If the bandgap reference is used, the ADC gives absolute voltage conversions.

For CT amplifiers configured as comparators (that is, open loop), a selected analog pin can be compared against another pin (fed from the other block), VBG, or a supply-referenced DAC voltage from the SC integrator. With the analog multiplexer bus in the CY8C22x45, CY8C21345, CY8C21x34, CY7C603xx, and CYWUSB6953 PSoC devices, a Port 0 pin can be compared against another pin without using resources of the adjacent column.

17.1.5 Continuous Time PSoC Block

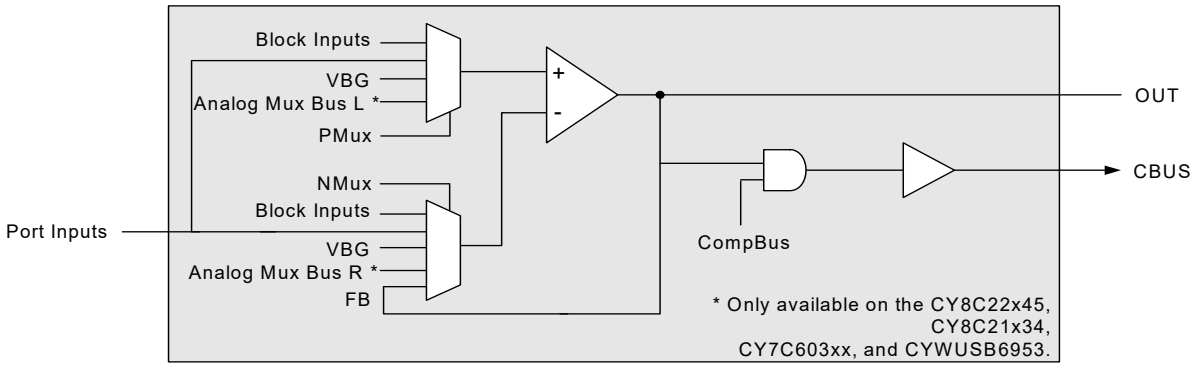
The Continuous Time blocks (Type ACE) are built around a low power, low offset amplifier. The CT block can be configured in two modes: As a unity gain buffer to drive to the other column or open loop as a comparator.

To configure as a comparator, select any NMux choice except feedback (FB). To enable the comparator bus output, the CompBus signal must be set in the ACE0xCR1 register. See Figure 17-10.

There are two discrete outputs from this block. These outputs connect to the following buses:

1. The comparator bus (CBUS), which is a digital bus that is a resource shared by all of the analog blocks in a column for that block. This output is available to system interface logic.
2. The local output bus (OUT), which is routed to the neighboring block.

Figure 17-10. Analog Continuous Time Block Diagram for PSoC Devices

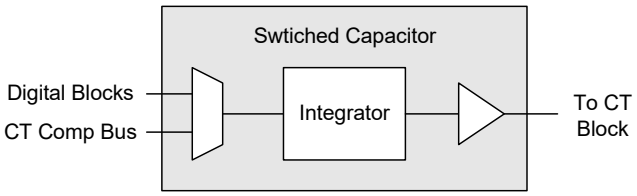


17.1.6 Switched Capacitor PSoC Block

The analog switched capacitor blocks accept a bit stream from either a digital block or a CT comparator. The SC block integrates this input and its output can then be connected to a CT block.

The low power SC block, in the CY8C22x45, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 (Type ASE), is automatically enabled whenever the CT block is powered up. Refer to the ACExxCR2 Register in this chapter.

Figure 17-11. Analog Switch Capacitor PSoC Block for the CY8C22x45, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC Devices



17.1.6.1 Application Description for the SC Block

The Analog Switched Capacitor (SC) blocks support DACs for comparator references. This application requires the use of one CT block. Analog-to-digital conversions can be done with a firmware-based successive approximation algorithm, using the SC block to provide a DAC reference.

The integrator speed can be modified to trade off accuracy for settling time.

17.2 PSoC Device Distinctions

The following are PSoC device distinctions for the CY8C22x45, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices.

1. The continuous time (CT) blocks in the CY8C22x45, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices differ from other PSoC Programmable System-on-Chip devices in the following ways:
   - The CT amplifier can only be configured as unity gain or open loop (comparator).
   - No separate low power comparator is available; however, this CT block amplifier is inherently low power and may be useful as a sleep mode comparator in many applications.
   - The column comparator bus is always driven from the CT block. When the CT amplifier is configured in Unity Gain mode, CompBus should be set to zero and the block outputs a zero on the comparator bus.
2. In the CY8C22x45, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices, the switched capacitor (SC) block consists of a low power integrator that is enabled whenever the CT block is enabled. It can be used to create a DAC reference for a CT comparator. The only configuration of the internal state of the SC block available to the user is input and output connections, and integrator speed by way of the FCap register bit.
3. The CY8C22x45, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices can use a VC3-based control for analog-to-digital conversion.
4. For the CY8C22x45, CY8C21345, CY8C21x34, CY7C603xx, and CYWUSB6953 PSoC devices, all GPIO pins can connect to the internal analog mux bus. However, there are analog mux buses in the CY8C22x45 and CY8C21345 PSoC devices. The odd bits GPIOs connect to left side analog mux bus (except P0[7]). The even bits GPIOs connect to right side analog mux bus (including P0[7]).
5. The temperature sensor input (V_{TEMP}) is connected through the ACE01 PMux. There is no special ground reference for the signal.

17.3 Register Definitions

The following registers are associated with the CY8C22x45, CY8C21345, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices and are listed in address order within their system resource configuration. The registers that are exclusive to the CY8C21x34, CY7C603xx, and CYWUSB6953 are summarized in the "Summary Table of the System Resource Registers" on page 374 and detailed in the I/O Analog Multiplexer chapter on page 422. For a complete table of all analog system registers for all other PSoC devices, refer to the "Summary Table of the Analog Registers" on page 336.

Each register description has an associated register table showing the bit structure for that register. Register bits that are grayed out throughout this document are reserved bits and are not detailed in the register descriptions that follow.

17.3.1 Summary Table for 2 Column Limited Analog System Registers

The table below lists the registers that are used by the CY8C22x45, CY8C21345, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices, in address order within their system resource configuration. Note that there are no registers associated with the CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 for the analog reference, because there are no configuration options for that function. The bits that are grayed out are reserved bits. Reserved bits should always be written with a value of '0'.

Table 17-5. Summary Table for 2 Column Limited Analog System Registers

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access : POR Value
ANALOG INTERFACE REGISTERS (page 350)										
0,62h	PWM_CR			HIGH[2:0]			LOW[1:0]		PWMEN	R : 00
0,64h	CMP_CR0			COMP[1:0]				AINT[1:0]		R : 00
0,66h	CMP_CR1			CLDIS[1:0]						RW : 00
0,68h	ADC0_CR	CMPST	LOREN	SHEN		CBSRC	AUTO		ADCEN	# : 00
0,69h	ADC1_CR	CMPST	LOREN	SHEN		CBSRC	AUTO		ADCEN	# : 00
0,E6h	DEC_CR0			IGEN[1:0]		ICLKS0				RW : 00
0,E7h	DEC_CR1				ICLKS2	ICLKS1				RW : 00
1,60h	CLK_CR0					AColumn1[1:0]		AColumn0[1:0]		RW : 00
1,61h	CLK_CR1				ACLK1[1:0]			ACLK0[1:0]		RW : 00
1,63h	AMD_CR0					AMOD0[3:0]				RW : 00
1,64h	CMP_GO_EN	GOO5	GOO1	SEL1[1:0]		GOO4	GOO0	SEL0[1:0]		RW : 00
1,66h	AMD_CR1					AMOD1[3:0]				RW : 00
1,67h	ALT_CR0	LUT1[3:0]				LUT0[3:0]				RW : 00
1,6Bh	CLK_CR3		SYS1	DIVCLK1[1:0]			SYS0	DIVCLK0[1:0]		RW : 00
1,E5h	ADC0_TR	CAPVAL_[7:0]								RW : 00
1,E6h	ADC1_TR	CAPVAL_[7:0]								RW : 00
ANALOG INPUT CONFIGURATION REGISTERS (page 355)										
0,60h	AMX_IN					ACI1[1:0]		ACI0[1:0]		RW : 00
1,62h	ABF_CR0	ACol1Mux								RW : 00
CONTINUOUS TIME PSoC BLOCK, TYPE E, REGISTERS (page 356)										
x,72h	ACE00CR1		CompBus	NMux[2:0]			PMux[2:0]			RW : 00
x,73h	ACE00CR2							FullRange	PWR	RW : 00
x,76h	ACE01CR1		CompBus	NMux[2:0]			PMux[2:0]			RW : 00
x,77h	ACE01CR2							FullRange	PWR	RW : 00
SWITCHED CAPACITOR PSoC BLOCK, TYPE E, REGISTERS (page 357)										
x,80h	ASE10CR0	FVal								RW : 00
x,84h	ASE11CR0	FVal								RW : 00

LEGEND
x An "x" before the comma in the address field indicates that this register can be accessed or written to no matter what bank is used.
Access is bit specific. Refer to the Register Details chapter on page 103.

17.3.2 Analog Interface Registers

PWM_CR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,62h	PWM_CR			HIGH[2:0]			LOW[1:0]		PWMEN	R : 00

The ADC PWM Control Register (PWM_CR) controls the parameters for the dedicated ADC PWM. This PWM function uses VC3 as its input clock so all periods are in terms of VC3 terminal counts.

Bits 5 to 3: HIGH[2:0]. These bits set the PWM high time in terms of VC3 periods.

Bits 2 and 1: LOW[1:0]. These bits set the PWM low time in terms of VC3 periods.

Bit 0: PWMEN. This bit starts and stops the PWM. When this bit is disabled, the PWM output state is always low. When this bit is enabled, the following two scenarios can occur:

1. If the low time programmed is greater than 0, the PWM waits for the first VC3 terminal count before starting the low time count.
2. If the low time programmed is 0, the PWM will wait for the first VC3 terminal count before going high, and then will start counting VC3 periods for the purpose of generating the PWM terminal count. The PWM will stay high continually until enabled.

For additional information, refer to the PWM_CR register on page 129.

CMP_CR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,64h	CMP_CR0			COMP[1:0]				AINT[1:0]		R : 00

The Analog Comparator Bus 0 Register (CMP_CR0) is used to poll the analog column comparator bits and select column interrupts.

Bits 5 and 4: COMP[1:0]. These bits are the read only bits corresponding to the comparator bits in each analog column. By default, they are synchronized to the column clock and thus may be reliably polled by the CPU.

Bits 1 and 0: AINT[1:0]. These bits choose between the analog column data and the dedicated incremental PWM terminal count as the analog interrupt source for this column.

For additional information, refer to the CMP_CR0 register on page 131.

CMP_CR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,66h	CMP_CR1			CLDIS[1:0]						RW : 00

The Analog Comparator Bus 1 Register (CMP_CR1) is used to override the analog column comparator synchronization.

Bits 5 and 4: CLDIS[1:0]. The CLDIS bits are used to override the analog column comparator synchronization. When these bits are set, the given column is not synchronized by the column clock. This capability is typically used to allow a continuous time comparator result to propagate directly to the interrupt controller during sleep. Because the master clocks (except the 32 kHz clock) are turned off during sleep, the synchronizer must be bypassed.

For additional information, refer to the CMP_CR1 register on page 132.

ADCx_CR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,68h	ADC0_CR	CMPST	LOREN	SHEN		CBSRC	AUTO		ADCEN	# : 00
0,69h	ADC1_CR	CMPST	LOREN	SHEN		CBSRC	AUTO		ADCEN	# : 00

The ADC Column 0 and Column 1 Configuration Register (ADCx_CR) controls the single slope ADC in each column.

Bit 7: CMPST. This bit is a read only status bit. It provides information at the end of an ADC conversion as to whether the analog comparator tripped or did not trip. This can be used to provide an over-range bit. For example, the range of an 8-bit conversion is 0 – 255, 256 codes. However, in order to achieve this range exactly, the PWM high time must define 255 clocks (0 clocks corresponding to a 0 ADC result). This is possible for a digital block PWM which has an arbitrary high and low time programming with respect to the input clock. However, because the dedicated ADC PWM has only a limited number of divide selections based on the VC3 period, the high time will normally be in powers of two. For example, in an 8-bit conversion, the PWM ADC will be set for 256 clocks, rather than 255, and this gives an extra code of 0 to 256. For the 256th code, the 8-bit counter value will roll over and therefore, be indistinguishable from code 0. However, the CMPST bit will indicate that this is actually the 256th code.

Bit 6: LOREN. This bit controls the range of the base current level of the ADC. A ‘0’ in this bit position is the normal current range. A ‘1’ in this bit selects the low current range that divides the current by an approximate factor of four.

Bit 5: SHEN. This bit controls sample and hold. When this bit is set to ‘1’, sample and hold is enabled and controlled by the ADC PWM selection if the AUTO mode bit is set. When this bit is set to ‘0’, sample and hold is disabled, and the comparator voltage input follows the input pin to which it is connected.

Bit 3: CBSRC. This bit controls the source of the comparator bus output to the digital blocks. By default, when this bit is ‘0’, the synchronized analog comparator output is the source for the digital comparator bus. When this bit is set to ‘1’, the selected PWM terminal count becomes the digital comparator bus source. In ADC operating mode, this bit is set to ‘1’ to implement Timer Capture digital interface and set to ‘0’ to implement Counter Enable digital interface.

Bit 2: AUTO. When enabled, this bit allows the conversion to be controlled by a selected PWM signal.

Bit 0: ADCEN. By default, the ADC circuit is powered down. When the ADCEN bit is set to ‘1’, the circuit is ready for use. The ADCEN bit must be set as part of the initial configuration, before enabling the PWM in AUTO mode.

For additional information, refer to the ADCx_CR register on page 133.

DEC_CR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E6h	DEC_CR0			IGEN[1:0]		ICLKS0				RW : 00

The Decimator Control Register 0 (DEC_CR0) contains control bits to access hardware support for ADC operation.

Bits 5 and 4: IGEN[1:0]. For single slope ADC (SSADC) support, IGEN[1:0] selects which column comparator bit will be gated by the output of the dedicated ADC PWM or a digital block. The high time of the PWM source corresponds to the ADC conversion period. This ensures that the comparator output is only processed for the precise conversion time. If a digital block is selected for the gating function, it is controlled by ICLKS0 in this register, and ICLKS2 and ICLKS1 in the DEC_CR1 register.

Bit 3: ICLKS0. In conjunction with ICLKS1 in the DEC_CR1 register, these bits select up to one of four digital blocks to provide the gating signal for an SSADC conversion.

For additional information, refer to the DEC_CR0 register on page 178.

DEC_CR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E7h	DEC_CR1				ICLKS2	ICLKS1				RW : 00

The Decimator Control Register 1 (DEC_CR1) is used to configure signals for ADC operation.

Bits 4 and 3: ICLKSx. The ICLKS1 and ICLKS2 bits in this register select the digital block sources for SSADC hardware support (see the DEC_CR0 register).

For additional information, refer to the DEC_CR1 register on page 179.

CLK_CR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,60h	CLK_CR0					AColumn1[1:0]		AColumn0[1:0]		RW : 00

The Analog Column Clock Control Register 0 (CLK_CR0) is used to select the clock source for an individual analog column.

Bits 3 to 0: AColumnx[1:0]. An analog column clock generator is provided for each column. The bits in this register select the source for each column clock generator.

There are four selections for each clock: VC1, VC2, ACLK0, and ACLK1. VC1 and VC2 are the programmable global system clocks. ACLK0 and ACLK1 sources are each selected from up to one of four digital block outputs (functioning as clock generators) as selected by CLK_CR1.

For additional information, refer to the CLK_CR0 register on page 221.

CLK_CR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,61h	CLK_CR1				ACLK1[1:0]			ACLK0[1:0]		RW : 00

The Analog Column Clock Control Register 1 (CLK_CR1) selects the clock source for an individual analog column.

Bits 4, 3 and 1, 0: ACLKx[1:0]. There are two 2-bit fields in this register that can select up to one of four digital blocks, to function as the clock source for ACLK0 and ACLK1. ACLK0 and ACLK1 are alternative clock inputs to the analog column clock generators (see the CLK_CR0 register above).

For additional information, refer to the CLK_CR1 register on page 222.

AMD_CR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,63h	AMD_CR0					AMOD0[3:0]				RW : 00

The Analog Modulation Control Register 0 (AMD_CR0) is used to select the modulator bits used with each column.

Bits 3 to 0: AMOD0[3:0]. These bits control the selection of the MODBITs for analog column 0. The MODBIT is a modulated data stream input into a Switched Capacitor block. Three bits for each column allow a one of eight selection for the MODBIT. Sources include any of the analog column comparator buses, two global buses, and one broadcast bus. The default for this function is zero or off.

For additional information, refer to the AMD_CR0 register on page 224.

CMP_GO_EN Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,64h	CMP_GO_EN	GOO5	GOO1	SEL1[1:0]		GOO4	GOO0	SEL0[1:0]		RW : 00

The Comparator Bus to Global Outputs Enable Register (CMP_GO_EN) controls options for driving the analog comparator bus and column clock to the global bus.

This register is also used by the CY8C24x94 and CY7C64215 PSoC devices.

Bit 7: GOO5. This bit drives the selected column 1 signal to GOO5.

Bit 6: GOO1. This bit drives the selected column 1 signal to GOO1.

Bits 5 and 4: SEL1[1:0]. These bits select the column 1 signal to output.

Bit 3: GOO4. This bit drives the selected column 0 signal to GOO4.

Bit 2: GOO0. This bit drives the selected column 0 signal to GOO0.

Bits 1 and 0: SEL0[1:0]. These bits select the column 0 signal to output.

For additional information, refer to the CMP_GO_EN register on page 225.

AMD_CR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,66h	AMD_CR1					AMOD1[3:0]				RW : 00

The Analog Modulation Control Register 1 (AMD_CR1) is used to select the modulator bits used with each column.

Bits 3 to 0: AMOD1[3:0]. These bits control the selection of the MODBITs for analog column 1. See the AMD_CR0 register above.

For additional information, refer to the AMD_CR1 register on page 226.

ALT_CR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,67h	ALT_CR0	LUT1[3:0]				LUT0[3:0]				RW : 00

The Analog LUT Control Register 0 (ALT_CR0) is used to select the logic function.

Bits 7 to 4 and 3 to 0: LUTx[3:0]. These bits control the selection of logic functions that may be selected for the analog comparator bits in column 0 and column 1. A one of 16 look-up table (LUT) is applied to the outputs of each column comparator bit and optionally a neighbor bit to implement two input logic functions.

Table 17-1 shows the available functions, where the A input applies to the selected column, and the B input applies to the next most significant neighbor column. Column 0 settings apply to combinations of column 0 and column 1. Column 1 settings apply to column 1 with B=0.

For additional information, refer to the ALT_CR0 register on page 227.

CLK_CR3 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,6Bh	CLK_CR3		SYS1	DIVCLK1[1:0]			SYS0	DIVCLK0[1:0]		RW : 00

The Analog Clock Source Control Register 3 (CLK_CR3) controls additional options for analog column clock generation. It allows an option for selecting SYSCLK directly as the column clock source, as well as additional divide values on the selected source.

Bit 6: SYS1. This bit selects SYSCLK as the source for the analog column 1 clocking.

Bits 5 and 4: DIVCLK1[1:0]. These bits control an optimal divide value on the selected clock in column 1.

Bit 2: SYS0. This bit selects SYSCLK as the source for the analog column 0 clocking.

Bits 1 and 0: DIVCLK0[1:0]. These bits control an optimal divide value on the selected clock in column 0.

For additional information, refer to the CLK_CR3 register on page 230.

ADCx_TR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E5h	ADC0_TR	CAPVAL_[7:0]								RW : 00
1,E6h	ADC1_TR	CAPVAL_[7:0]								RW : 00

The ADC Column 0 and Column 1 Trim Register (ADCx_TR) controls a combination of capacitor and current values that determine the slope of the ADC voltage ramp.

Bits 7 to 0: CAPVAL_[7:0]. These bits are used to calibrate the ADC. Before the converter can be used, the capacitor array must be calibrated with a known voltage (for example, the bandgap voltage). The goal of this calibration process is to tune the ramp time (slope) such that the full-scale ADC input value results in a full-scale ADC code. This is accomplished by matching the ramp time to the desired full-scale conversion period, which is dependent on clock rate and resolution. The bits of the register have an inverted sense; that is, a '1' reduces the capacitance which increases the speed of the ramp and a '0' increases the capacitance which decreases the speed of the ramp.

For additional information, refer to the ADCx_TR register on page 259.

17.3.3 Analog Input Configuration Registers

AMX_IN Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,60h	AMX_IN					ACI1[1:0]		ACI0[1:0]		RW : 00

The Analog Input Select Register (AMX_IN) controls the analog muxes that feed signals in from port pins into the analog column.

Bits 3 to 0: ACIx[1:0]. The ACI1[1:0] and ACI0[1:0] bits control the analog muxes that feed signals in from port pins into the analog column. The analog column can have up to eight port bits connected to its muxed input. ACI1 and ACI0 are used to select among even and odd pins. The AC1Mux bit field controls the bits for those muxes and is located in the Analog Output Buffer Control Register (ABF_CR0).

For additional information, refer to the AMX_IN register on page 127.

ABF_CR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,62h	ABF_CR0	ACol1Mux								RW : 00

The Analog Output Buffer Control Register 0 (ABF_CR0) controls analog input muxes from Port 0.

Bit 7: ACol1MUX. A mux selects the output of column 0 input mux or column 1 input mux. When set, this bit sets the column 1 input to column 0 input mux output.

For additional information, refer to the ABF_CR0 register on page 223.

17.3.4 Continuous Time PSoC Block Registers

The naming convention for the continuous time registers and their arrays of PSoC blocks is <Prefix>mn<Suffix>, where m=row index and n=column index. Therefore, ACE01CR2 (written ACExxCR2) is a register for an analog PSoC block in row 0 column 1.

ACExxCR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,72h	ACE00CR1		CompBus	NMux[2:0]			PMux[2:0]			RW : 00
x,76h	ACE01CR1		CompBus	NMux[2:0]			PMux[2:0]			RW : 00

LEGEND
x An “x” before the comma in the address field indicates that this register can be accessed or written to no matter what bank is used.

The Analog Continuous Time Type E Block Control Register 1 (ACExxCR1) is one of two registers used to configure the type E continuous time PSoC block.

Bit 6: CompBus. This bit determines whether the comparator bus is driven from the amplifier output or driven low. If the CT block is configured in Unity Gain mode, this bit should be set to zero so the comparator bus is driven low.

Bits 5 to 3: NMux[2:0]. These bits control the multiplexing of inputs to the inverting input of the opamp. There are several input choices from outside the block, plus an internal feedback selection.

Bits 2 to 0: PMux[2:0]. These bits control the multiplexing of the five inputs to the non-inverting input of the opamp.

For additional information, refer to the ACExxCR1 register on page 137.

ACExxCR2 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,73h	ACE00CR2							FullRange	PWR	RW : 00
x,77h	ACE01CR2							FullRange	PWR	RW : 00

LEGEND
x An “x” before the comma in the address field indicates that this register can be accessed or written to no matter what bank is used.

The Analog Continuous Time Type E Block Control Register 2 (ACExxCR2) is one of two registers used to configure the type E continuous time PSoC block.

Bit 1: FullRange. For slightly higher power, this bit enables inputs from Vss to Vdd.

Bit 0: PWR. This bit is used to power up the CT block and SC block in the column.

For additional information, refer to the ACExxCR2 register on page 138.

17.3.5 Switched Capacitor PSoC Block Register

The naming convention for the switched capacitor register and its array of PSoC blocks is <Prefix>mn<Suffix>, where m=row index and n=column index. Therefore, ASE10CR0 (written ASExxCR0) is a register for an analog PSoC block in row 1 column 0.

ASExxCR0 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,80h	ASE10CR0	FVal								RW : 00
x,84h	ASE11CR0	FVal								RW : 00

LEGEND
x An “x” before the comma in the address field indicates that this register can be accessed or written to no matter what bank is used.

The Analog Switch Capacitor Type E Block Control Register 0 (ASExxCR0) is used to configure a type E switched capacitor PSoC block.

Bit 7: FVal. This bit controls the size of the bandwidth of the filler in the Type E block.

For additional information, refer to the ASExxCR0 register on page 139.

18. Two Column Analog Compare

This chapter explains the Two Column Analog Compare System PSoC devices and their associated registers. This Two Column Analog Compare System is based on the Two Column Limited Analog System on the same die and with some chopping and enhancement. It details the entire analog system for two analog compare functionality, including the analog compare interface, analog compare array, analog compare input configuration, analog compare reference, CT blocks, and the amplifier. For a complete table of the Two Column Analog Compare System registers, refer to the "Summary Table for Analog System Registers" on page 366. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

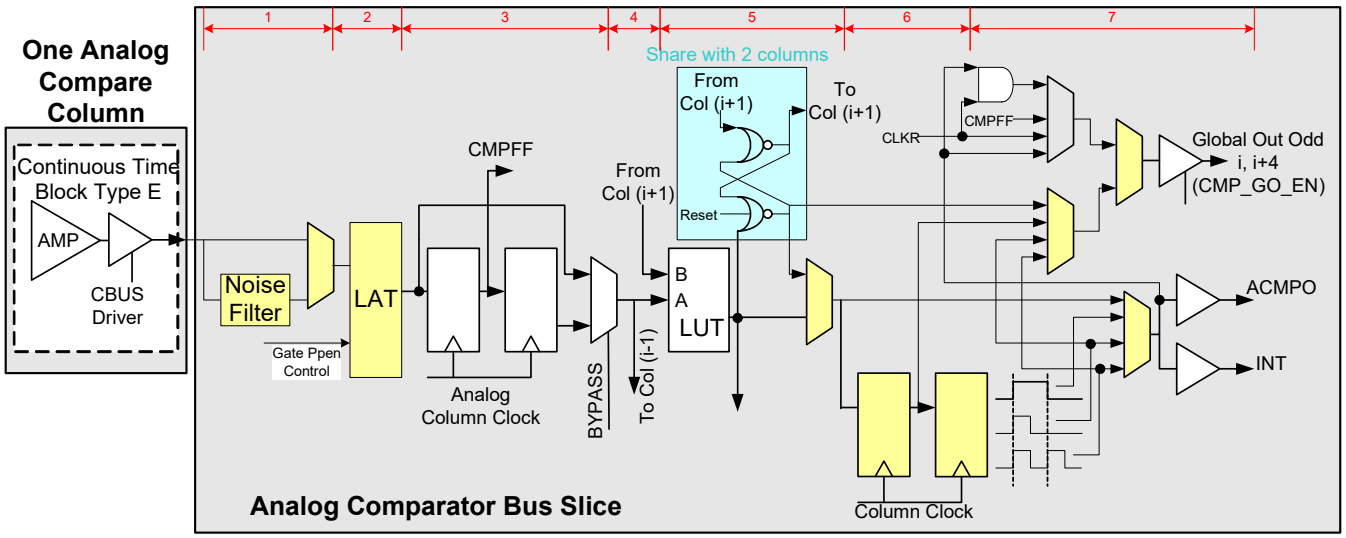
This Two Column Analog Compare System also employs the use of an I/O analog multiplexer system resource. The I/O Analog Multiplexer is described in the I/O Analog Multiplexer chapter on page 422. A summary of the I/O Analog Multiplexer registers is in the section "System Resources Register Summary" on page 374.

18.1 Architectural Description

18.1.1 Analog Interface

Figure 18-1 displays the top-level diagram of the PSoC devices' analog compare interface system.

Figure 18-1. Analog Compare Column Comparator Bus Slice of thePSoC Devices



18.1.1.1 Analog Comparator Bus Interface

Each analog column has a dedicated comparator bus associated with it. In the PSoC devices, only the Continuous Time (CT) block can drive this bus. The output on the comparator bus can drive into the digital blocks as a data input and into CSD logic as gate input. It also serves as an interrupt input, and is available as read only data in the Compare Column Comparator Control register (CSCMP_CR0). It can be driven to the global output bus by way of the Comparator to Global Output Enable register (CSCMP_GO_EN).

Figure 18-1 illustrates one column of the comparator bus. The Analog Compare output data processing includes 7 stages and the 7th is for data output, to Digital Block, or Global Output or Interrupt.

Input Noise Filter

The analog compare output signal can pass a noise filter. That is, the signal switching with width less than certain selected clock cycles will be filtered out. There is also an option to bypass this function.

Table 18-1. Filter Clock Selection

FLTx_CSL	0	1
Filter Clock	SYSCLK	Column Clock

Table 18-2. Filter Width

FLTx_WD[1:0]	0	1	2	3
Filter Width	1	2	4	8

Timed Gate

The analog compare output can only pass to the next stage within a certain period if timed gate function is enabled. It receives two inputs. One input is 1 of 8 digital blocks' FO1 output; any changing of it turns off the gate, and then the previous value is kept in its output. The second input is 1 of 4 configurable trigger signals. These trigger signals are shared with SAR ADC auto-align mode, and then they are generated in separate blocks. The rising edge of the selected trigger signal turns on the gate and then the analog compare output can go to the next stage.

Table 18-3. Gate Open Selection

Trigger Selection [1:0]	00B	01B	10B	11B
Gate Open At	TG_L or TG_H	TG_L	TG_H	TG_16BIT

Synchronization

The comparator bus is synchronized by the selected column clock before it is available, to either drive the digital blocks, interrupt, or for it to be read in the CSCMP_CR0 register. There is also an option to bypass the synchronization in each column into a transparent mode by setting bits in the CSCMP_CR1 register.

LUT

The analog comparator bus output values can be modified or combined with another analog comparator bus through the Analog Look-Up-Table (LUT) function. The LUT takes two inputs, A and B, and provides a selection of 16 possible logic functions for those inputs. The LUT A and B inputs for each column comparator output is shown in the following table.

Table 18-4. A and B Inputs for Each Column Comparator LUT Output

Comparator LUT Output	A	B
Column 0	ACMP0	ACMP1
Column 1	ACMP1	0

The LUT configuration is set in one control register, CSLUT_CR0. Each selection for each column is encoded in four bits. The function value corresponding to the bit encoding is shown in the following table.

Table 18-5. CSLUT_CR0 Register

CSLUTx[3:0]	0h: 0000: FALSE 1h: 0001: A .AND. $\overline{B}$ 2h: 0010: A .AND. $\overline{B}$ 3h: 0011: $\overline{A}$ 4h: 0100: $\overline{A}$ .AND. B 5h: 0101: B 6h: 0110: A .XOR. B 7h: 0111: A .OR. B 8h: 1000: A .NOR. B 9h: 1001: $\overline{A}$ .XNOR. B Ah: 1010: $\overline{B}$ Bh: 1011: $\overline{A}$ .OR. $\overline{B}$ Ch: 1100: $\overline{A}$ Dh: 1101: $\overline{A}$ .OR. B Eh: 1110: A. NAND. B Fh: 1111: TRUE

RS-Latch

A special RS-Lat function is added in analog comparator output data process. The block will be shared with 2 compare columns. The compare column 0 is the master input and the compare column 1 is the slave input. And the master output (Q) and slave output (QB) will go back to compare column 0 and compare column 1, respectively. If the master output is a primary output, then the 'high' signal from master channel will clear the primary output (that is, the target will be cut off for protection). The 'high' signal from slave channel will set the primary output (that is, the target will be turned on for normal operation). The RS-Latch is bypassed if RS-Latch is not selected. (Therefore, a RS_EN bit is used to enable RS function, and another RS_SEL bit in each column is used to select RS function output.)

Pulse Generator

The output from RS-Latch or LUT stage can be synced or resynced. Based on the synced version of input, you can generate on-shot pulse on both rising edge and failing edge of the input. The sync clock is running on column clock.

Output Stage

The compare bus output can be driven by the raw or re-synced version of the signal coming from LUT or RS-Latch, as well as the one-shot pulse generated by them. The selection between LUT output and RS-Latch output is made in register CMPCLK1; the selection of which signal to be driven onto compare bus is configured in register CMPPWMCR and CMPCOLMUX for column 0 and column 1 independently.

Signals from compare column 0 and 1 can also be driven to global output odd. These signals including compare bus output, unsynced column clock, comparator output after signal sync, unsynced column clock gated by compare bus output, RS-Latch output, synced version of LUT/RS-Latch output, positive one-shot, positive or negative one-shot by LUT/RS-Latch output.

The global output available for compare column 0 is GOO2 and GOO6, and for compare column 1 is GOO3 and GOO7. The bits for turning on/off this feature and signal selection resides in register CSCMPGOEN and CMPCLK1.

18.1.1.2 Analog Column Clock Generation

The input clock source for each column clock generator is selectable through the CLK_CR0 register. There are four selections for each column: VC1, VC2, ACLK0, and ACLK1. An additional selection, SYSCLK, is controlled by the CLK_CR3 register. The VC1 and VC2 clock signals are global system clocks. Programming options for these system clocks can be accessed in the OSC_CR1 register. Each of the ACLK0 and ACLK1 clock selections are driven by a selection of digital block outputs. The settings for the digital block selection are located in the CLK_CR1 register in Two Column Limited Analog System (that is, the settings are shared between Two Column Analog Compare System and Two Column Limited Analog System). The CLK_CR3 register has additional column clock options. This register allows for a direct SYSCLK option as well as the option to divide the selected column clock by 2, 4, or 8.

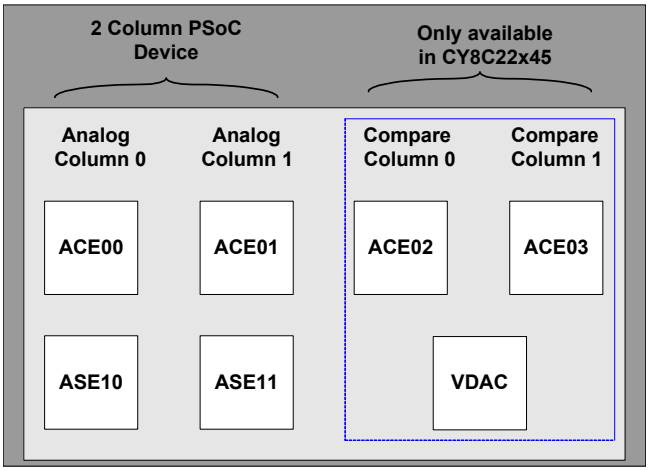
18.1.2 Analog Array

The analog array differs somewhat from the other analog array in Two Column Limited Analog System.

Figure 18-2 displays the analog arrays for the devices, containing the type E continuous time blocks (ACE) and a VDAC. The figures that follow illustrate the analog multiplexer (mux) connections.

Each compare column has a dedicated comparator bus associated with it. Only the CT block in each column can drive this bus. When the CT block is not configured as a comparator, a zero is driven to the comparator block. Refer to the CMPxCR1 register on page 210 and the “Analog Comparator Bus Interface” on page 359 in the Analog Interface section for more information.

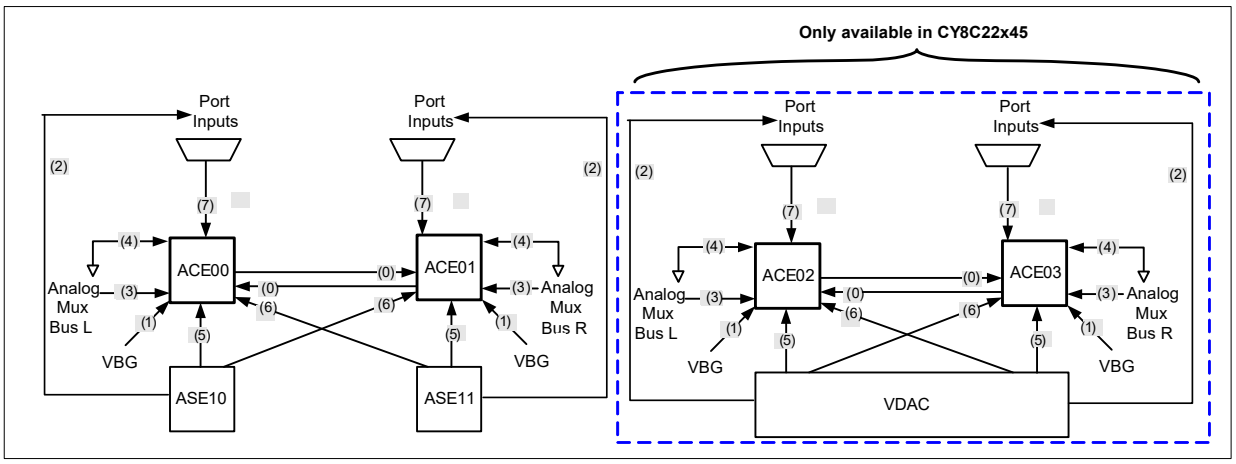
Figure 18-2. Array of Analog PSoC Blocks for the PSoC Devices



18.1.2.1 NMux Connections

The NMux is an 8-to-1 mux which determines the source for the inverting (also called negative) input of Continuous Time (CT) PSoC blocks. These blocks are named ACE0x. More details on the CT PSoC blocks are available in this chapter, in the section titled “Continuous Time PSoC Block” on page 364. The NMux connections are described in detail in the CMPxCR1 register on page 210, bits NMux[2:0]. The numbers in Figure 18-3, which are associated with each arrow, are the corresponding NMux select line values for the data in the NMux portion of the register. The call out names in the figure show nets selected for each NMux value. Note that column 2 and column 3 are only available in the CY8C22x45 PSoC devices.

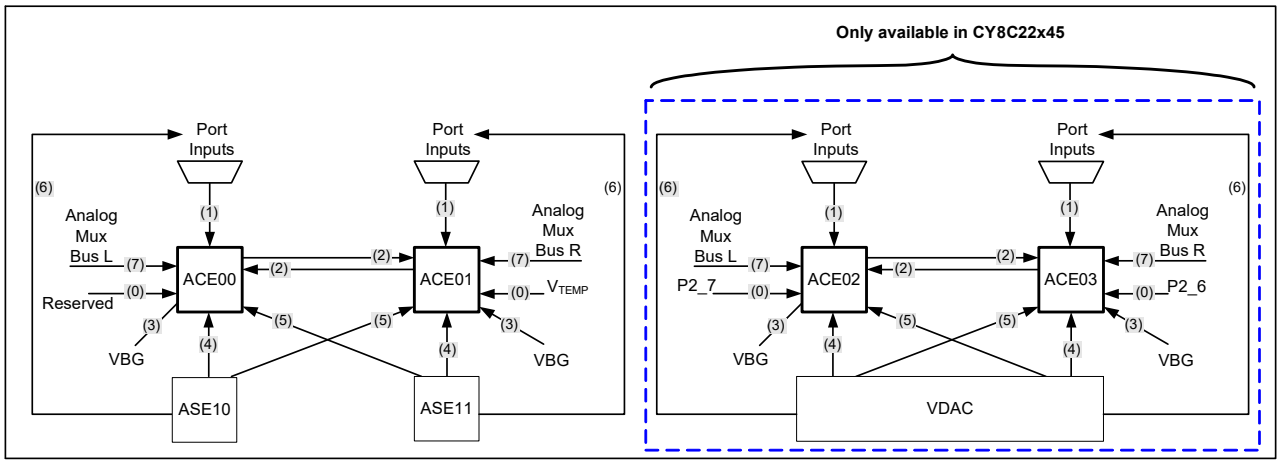
Figure 18-3. NMux Connections for the CY8C22x45 PSoC Devices



18.1.2.2 PMux Connections

The PMux is an 8-to-1 mux which determines the source for the non-inverting (also called positive) input of CT PSoC blocks (ACE0x). More details on the CT PSoC blocks are available in this chapter, in the section titled “Continuous Time PSoC Block” on page 364. The PMux connections are described in detail in the CMPxCR1 register on page 210 (bits PMux[2:0]). The numbers in Figure 18-4, which are associated with each arrow, are the corresponding PMux select line values for the data in the PMux portion of the register. The call out names in the figure show nets selected for each PMux value. Note that column 2 and column 3 are only available in the CY8C22x45 PSoC devices.

Figure 18-4. PMux Connections for the CY8C22x45 PSoC Devices



18.1.3 Analog Input Configuration

Figure 18-5 shows the analog interconnect for compare columns in the CY8C22x45 PSoC devices. For a detailed description of the I/O analog multiplexer functionality illustrated in Figure 18-6, refer to the I/O Analog Multiplexer chapter on page 422.

The input multiplexer (mux) maps device inputs (package pins) to analog compare columns, based on bit values in the CMPCOLMUX register. Each compare column is fed by one of two 4-to-1 muxes. The muxes are CMOS switches with typical resistances in the range of 2K ohms.

Figure 18-5. Column Analog Interconnect for column 2 and 3 (Only for the CY8C22x45 PSoC Devices)

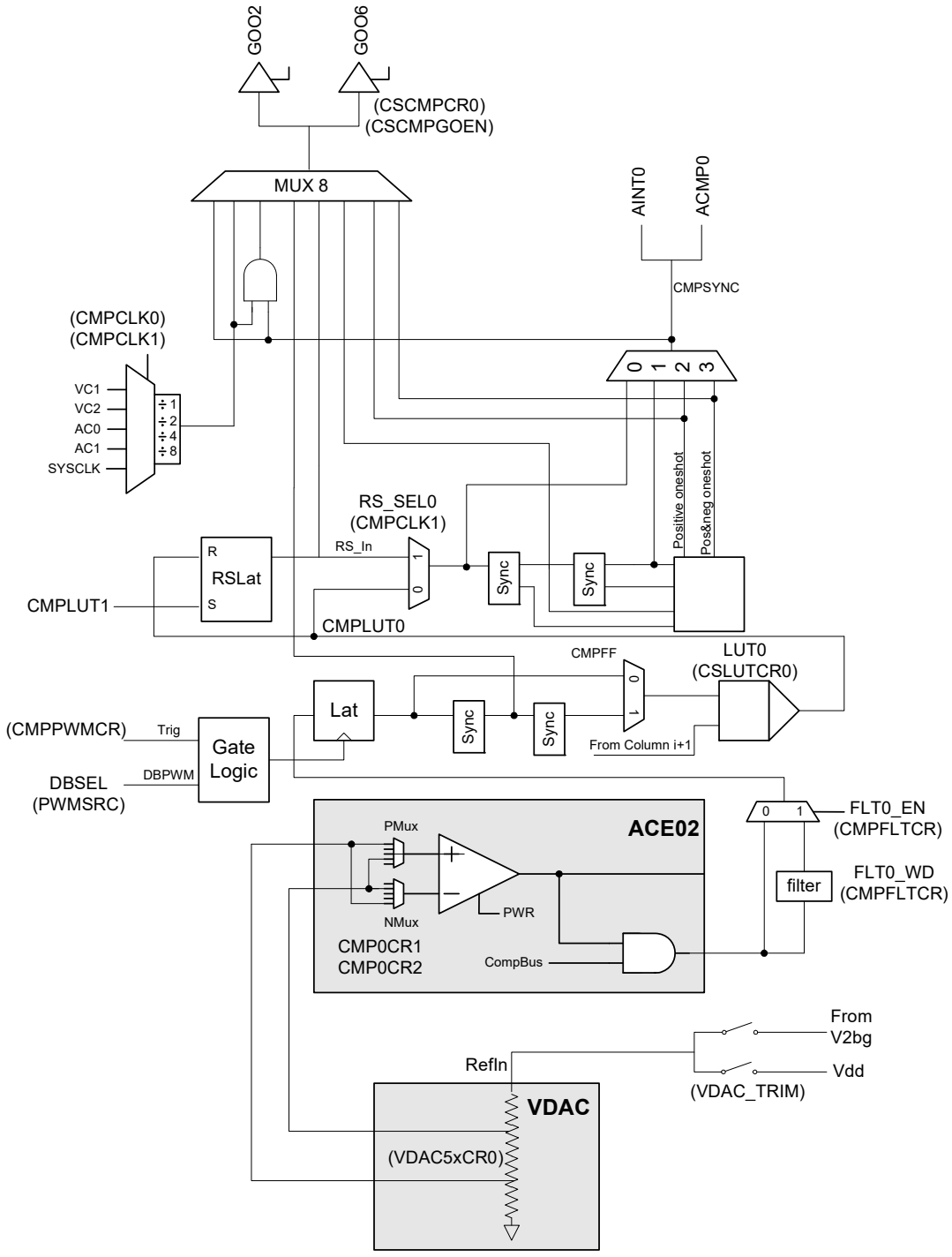
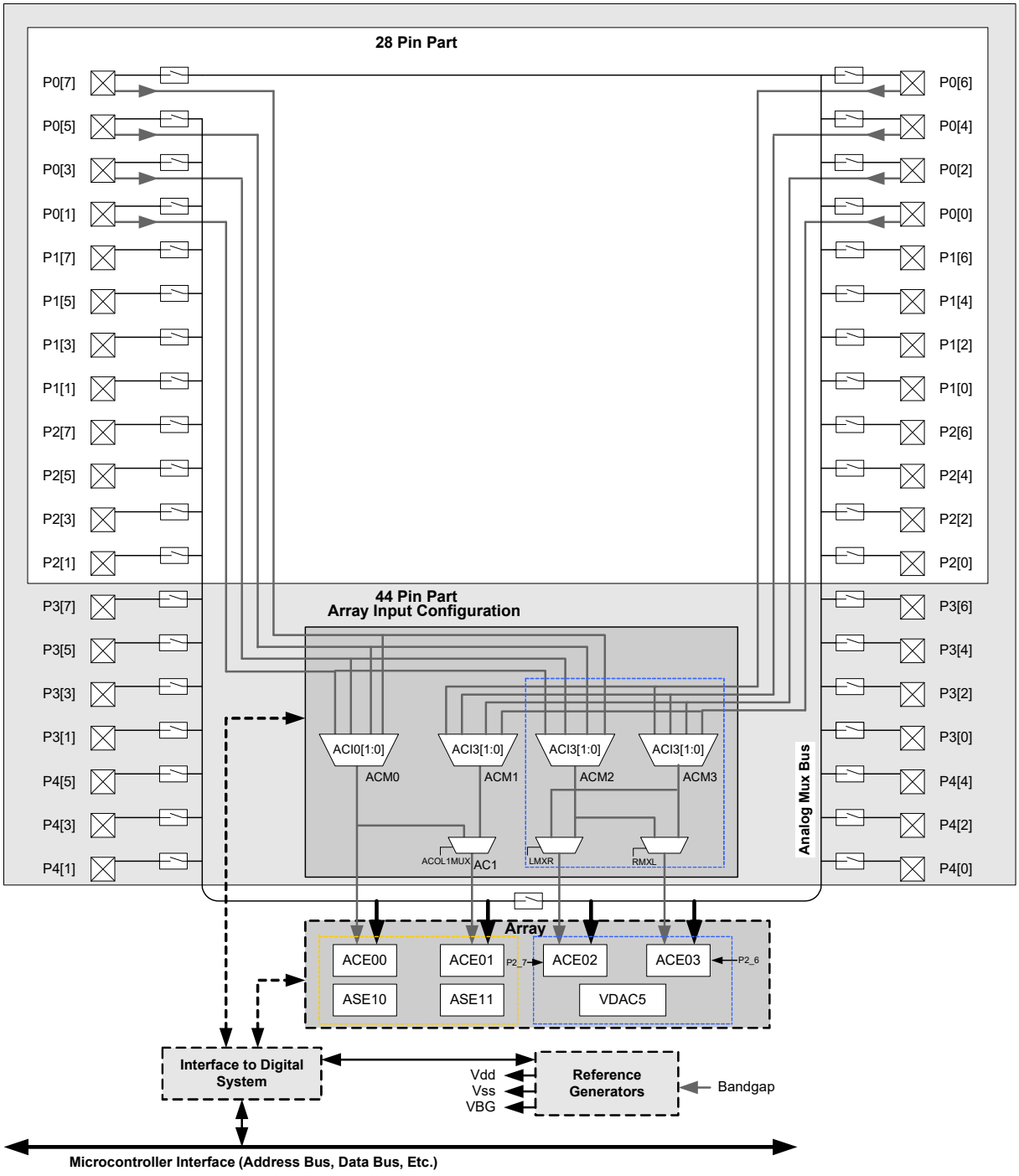


Figure 18-6. Analog Pin Block Diagram for the CY8C22x45 PSoC Devices



18.1.4 Analog Reference

The PSoC device is a single supply part, with no negative voltage available or applicable. The CY8C22x45, CY8C21345, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices support only one analog reference, which is the bandgap voltage VBG. This voltage is routed to the CT blocks in each analog column. VBG is available at both positive and negative inputs of each CT amplifier.

For CT amplifiers configured as comparators (that is, open loop), a selected analog pin can be compared against another pin (fed from the other block), VBG, or an DAC voltage from the VDAC5. With the analog multiplexer bus in the CY8C22x45 and CY8C21345 PSoC devices, a Port 0 pin can be compared against another pin without using resources of the adjacent column.

Note that in the CY8C22x45 PSoC devices, a VDAC5 can be used to generate voltage reference for ACE02 and ACE03, and is available in both NMUX and PMUX. The legal output of this VDAC5 output can be configured in register VDAC5xCR0 with a range from 0 to the reference input. Reference input of VDAC5 can be either Vdd or regulated output of bandgap by changing the values of register VDAC_TRIM.

18.1.5 Continuous Time PSoC Block

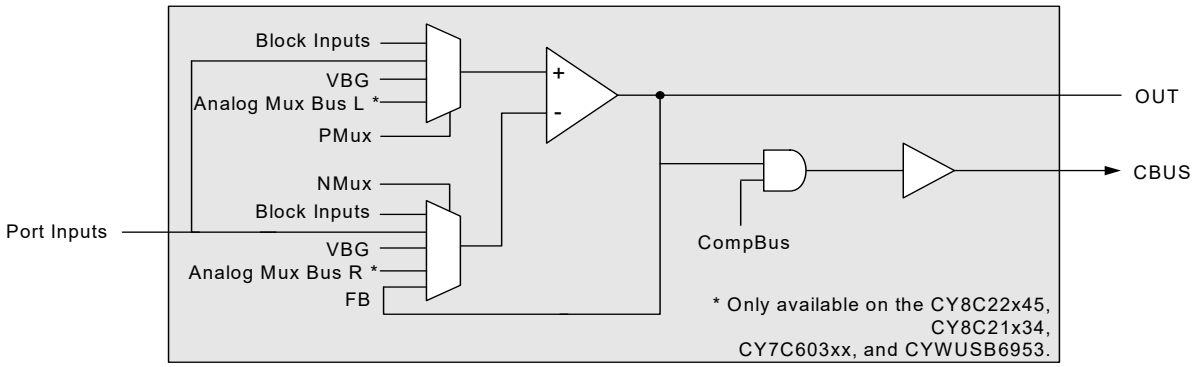
The Continuous Time blocks (Type ACE) are built around a low power, low offset amplifier. The CT block can be configured in two modes: As a unity gain buffer to drive to the other column or open loop as a comparator.

To configure as a comparator, select any NMux choice except feedback (FB). To enable the comparator bus output, the CompBus signal must be set in the CMPxCR1 register. See Figure 18-7.

There are two discrete outputs from this block. These outputs connect to the following buses:

1. The comparator bus (CBUS), which is a digital bus that is a resource shared by all of the analog blocks in a column for that block. This output is available to system interface logic.
2. The local output bus (OUT), which is routed to the neighboring block.

Figure 18-7. Analog Continuous Time Block Diagram for PSoC Devices

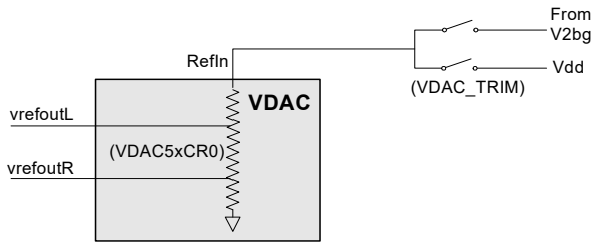


18.1.6 VDAC

The VDAC is only available in the CY8C22x45 PSoC devices and can convert a 5-bit width digital code into an analog voltage which can then be connected to a CT block. See Figure 18-8.

VDAC block can convert a 5-bit width digital code into analog voltage on vrefoutL and vrefoutR with a legal range from Re-fIn/31 to RefIn by configuring registers VDAC50CR0 and VDAC51CR0. And the Reference input can derive from regulated bandgap output or from noisy Vdd by accessing register VDAC_TRIM. When disabled, the reference output will be connected to vgnd and thus provides a ground supply on both vrefoutL and vrefoutR (this is also controlled in register VDAC_TRIM).

Figure 18-8. VDAC5 block in the CY8C22x45 PSoC Devices



18.2 PSoC Device Distinctions

The following are PSoC device distinctions for the CY8C22x45, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices.

- The continuous time (CT) blocks in the CY8C22x45, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices differ from other PSoC Mixed Signal Array devices in the following ways:
- The CT amplifier can only be configured as unity gain or open loop (comparator).
- No separate low power comparator is available; however, this CT block amplifier is inherently low power and may be useful as a sleep mode comparator in many applications.
- The column comparator bus is always driven from the CT block. When the CT amplifier is configured in Unity Gain mode, CompBus should be set to zero and the block outputs a zero on the comparator bus.
- In the CY8C22x45, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices, the switched capacitor (SC) block consists of a low power integrator that is enabled whenever the CT block is enabled. It can be used to create a DAC reference for a CT comparator. The only configuration of the internal state of the SC block available to the user is input and output connections, and integrator speed by way of the FCap register bit.
- The CY8C22x45, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices can use a VC3-based control for analog-to-digital conversion.
- In the CY8C22x45 PSoC devices, there are separate PWM signals to drive the SC block. This PWM signal frequency is fixed at 3MHz (when SYSCLK is 24MHz) and duty is from 0 to 15/16.
- For the CY8C22x45, CY8C21345, CY8C21x34, CY7C603xx, and CYWUSB6953 PSoC devices, all GPIO pins can connect to the internal analog mux bus. However, there are analog mux buses in the CY8C22x45 and CY8C21345 PSoC devices. The odd bits GPIOs connect to left side analog mux bus (except P0[7]). The even bits GPIOs connect to right side analog mux bus (including P0[7]).
- The temperature sensor input (V_{TEMP}) is connected through the ACE01 PMux. There is no special ground reference for the signal.
- In the CY8C22x45 PSoC devices, column 2 and column 3 (compare column), the reference voltage can derive from a VDAC5 block. This can provide a voltage from 0 to Vdd or regulated bandgap output voltage.
- In the CY8C22x45 PSoC devices, P2[7] / P2[6] can be connected through the ACE02 / ACE03 PMux, respectively.

18.3 Register Definitions

The following registers are associated with the CY8C22x45 and CY8C21345 PSoC devices and are listed in address order within their system resource configuration. For a complete table of all analog system registers, refer to the “Summary Table of the Analog Registers” on page 336.

Each register description has an associated register table showing the bit structure for that register. Register bits that are grayed out throughout this document are reserved bits and are not detailed in the register descriptions that follow.

18.3.1 Summary Table for Analog System Registers

The table below lists the registers that are used by the CY8C22x45 and CY8C21345 PSoC devices, in address order within their system resource configuration. Note that there are no registers for the analog reference, because there are no configuration options for that function. The bits that are grayed out are reserved bits. Reserved bits should always be written with a value of ‘0’.

Table 18-6. Summary Table for Analog System Registers

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access: POR Value
ANALOG INTERFACE REGISTERS (page 367)										
1,58h	CSCMPCR0	CSSEL1[2]	CSSEL0[2]	CSCMP[1:0]		CSIGEN[1:0]		CSCLKDIS[1:0]		RW : 00
1,59h	CSCMPGOEN	GOO7	GOO3	CSSEL1[1:0]		GOO6	GOO2	CSSEL0[1:0]		RW : 00
1,5Ah	CSLUTCR0	CSLUT1[3:0]				CSLUT0[3:0]				RW : 00
1,5Ch	CMPPWMCR	ACMPS0[1:0]		PWMDLY_EN[1]	PWMCLK_SEL1[1:0]		PWMDLY_EN[0]	PWMCLK_SEL0[1:0]		RW : 00
1,5Dh	CMPFLTCR	FLT1_WD[1:0]		FLT1_CSL	FLT1_EN	FLT0_WD[1:0]		FLT0_CSL	FLT0_EN	RW : 00
1,5Eh	CMPCLK1	RS_SEL[1]	SYS1	DIVCLK1		RS_SEL[0]	SYS0	DIVCLK0		RW : 00
1,5Fh	CMPCLK0	RS_EN		SEL_CSD[1:0]		CMPCOL1[1:0]		CMPCOL0[1:0]		RW : 00
ANALOG INPUT CONFIGURATION REGISTERS (page 370)										
1,5Bh	CMPCOLMUX	ACMPS1<1:0>		LMXR	RMXL	ACI1[1:0]		ACI0[1:0]		RW : 00
CONTINUOUS TIME PSoC BLOCK, TYPE E, REGISTERS (page 371)										
1,50h	CMP0CR1	BINC	CompBus	NMux[2:0]			PMux[2:0]			RW : 00
1,51h	CMP0CR2							FullRange	PWR	RW : 00
1,54h	CMP1CR1	BINC	CompBus	NMux[2:0]			PMux[2:0]			RW : 00
1,55h	CMP1CR2							FullRange	PWR	RW : 00
VDAC5 BLOCK REGISTERS (page 372)										
1,53h	VDAC50CR0				VDACIN<4:0>					RW : 00
1,57h	VDAC51CR0				VDACIN<4:0>					RW : 00
1,E7h	VDAC_TRIM		EN	V24	SAMPLE	TRIM[3:0]				RW : 00

LEGEND
x An “x” before the comma in the address field indicates that this register can be accessed or written to no matter what bank is used.
Access is bit specific. Refer to the Register Details chapter on page 103.

18.3.2 Analog Interface Registers

CSCMPCR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,58h	CSCMPCR0	CSSEL1[2]	CSSEL0[2]	CSCMP[1:0]		CSIGEN[1:0]		CSCLKDIS[1:0]		RW : 00

The CSCMPCR0 register controls the selection of signals driven to GOO and also for gating function.

Bit 7: CSSEL1[2]. The MSB of CSSEL1. See CSCMPGOEN (1,59h).

Bit 6: CSSEL0[2]. The MSB of CSSEL0. See CSCMPGOEN (1,59h).

Bits 5 and 4: CSCMP[1:0]. Read only; can read out the output of CS comparator.

Bit 3: CSIGEN[1:0]. Reserved for gate function.

Bits 1 and 0: CSCLKDIS[1:0]. CS comparator bus sync enable/disable.

For additional information, refer to the CSCMPCR0 register on page 213.

CSCMPGOEN Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,59h	CSCMPGOEN	GOO7	GOO3	CSSEL1[1:0]		GOO6	GOO2	CSSEL0[1:0]		RW : 00

The CSCMPGOEN register controls options for driving the analog comparator bus and clock to the global bus.

Bit 7: GOO7. Drives the selected CS comparator column 1 signal to GOO7.

Bit 6: GOO3. Drives the selected CS comparator column 1 signal to GOO3.

Bits 5 and 4: CSSEL1[1:0]. Selects the CS comparator column 1 signal to global bus. MSB is in CSCMPCR0 (1,58h).

000: comparator bus output

001: unsynced column clock

010: comparator output after signal sync

011: unsynced column clock gated by comparator bus output

100: RS-Latch slave output

101: CMPFF2

110: positive one-shot

111: positive or negative one-shot

Bit 3: GOO6. Drives the selected CS comparator column 0 signal to GOO6.

Bit 2: GOO2. Drives the selected CS comparator column 0 signal to GOO2.

Bits 1 and 0: CSSEL0[1:0]. Selects the CS comparator column 0 signal to global bus. MSB is in CSCMPCR0 (1,58h).

000: comparator bus output

001: unsynced column clock

010: comparator output after signal sync

011: unsynced column clock gated by comparator bus output

100: RS-Latch slave output

101: CMPFF2

110: positive one-shot

111: positive or negative one-shot

For additional information, refer to the CSCMPGOEN register on page 214.

CSLUTCR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,5Ah	CSLUTCR0	CSLUT1[3:0]				CSLUT0[3:0]				RW : 00

This register is used to select the logic function.

Bits 7 to 4: CSLUT1[3:0]. Select 1 of 16 logic functions for output of comparator bus 1.

Bits 3 to 0: CSLUT0[3:0]. Select 1 of 16 logic functions for output of comparator bus 0.

For additional information, refer to the CSLUTCR0 register on page 215.

CMPPWMCR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,5Ch	CMPPWMCR	ACMPS0[1:0]		PWMDLY_EN[1]	PWMCLK_SEL1[1:0]		PWMDLY_EN[0]	PWMCLK_SEL0[1:0]		RW : 00

This register is used to control the PWM function.

Bit 7 and 6: ACMPS0[1:0]. These two bits select the signal for column 2 compare bus output.

00: LUT out or RS_Lat Master out

01: double sync version of above signal

10: positive one-shot from above signal

11: positive or negative one-shot from above signal

Bit 5: PWMDLY_EN[1]. This bit controls the enable of the PWM delay function of column 3.

Bit 4 and 3: PWMCLK_SEL1[1:0]. These two bits select the trigger signals for PWM for column 3.

00: TG_L or TG_H

01: TG_L

10: TG_H

11: TG_16BIT

Bit 2: PWMDLY_EN[0]. This bit controls the enable of the PWM delay function of column 2.

Bit 1 and 0: PWMCLK_SEL0[1:0]. These two bits select the trigger signals for PWM for column 2.

00: TG_L or TG_H

01: TG_L

10: TG_H

11: TG_16BIT

For additional information, refer to the CMPPWMCR register on page 217.

CMPFLTCR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,5Dh	CMPFLTCR	FLT1_WD[1:0]		FLT1_CSL	FLT1_EN	FLT0_WD[1:0]		FLT0_CSL	FLT0_EN	RW : 00

This register controls the pulse filter feature for analog column 2 and column 3.

Bits 7 and 6: FLT1_WD[1:0]. These two bits control the filter width selection for column 3:

00: 1 selected clock

01: 2 selected clock

10: 4 selected clock

11: 8 selected clock

Bit 5: FLT1_CSL. This bit is the filter clock selection for column 3.

Bit 4: FLT1_EN. This bit controls the filter feature enable of column 3.

Bits 3 and 2: FLT0_WD[1:0]. These two bits control the filter width selection for column 2:

00: 1 selected clock

01: 2 selected clock

10: 4 selected clock

11: 8 selected clock

Bit 1: FLT0_CSL. This bit is the filter clock selection for column 2.

Bit 0: FLT0_EN. This bit controls the filter feature enable of column 2.

For additional information, refer to the CMPFLTCR register on page 218.

CMPCLK1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,5Eh	CMPCLK1	RS_SEL[1]	SYS1	DIVCLK1		RS_SEL[0]	SYS0	DIVCLK0		RW : 00

This register controls the analog column clock selection and RS feature with CMPCLK0.

Bit 7: RS_SEL[1]. If set to 1, this bit will select the RS output to compare bus for analog column 3.

Bit 6: SYS1.

0: column 3 clock selection is controlled by CMPCLK0

1: column 3 clock selection is controlled by SYSCLK

Bits 5 and 4: DIVCLK1[1:0].

00: no divide on selected column 3 clock

01: divide by 2 on selected column 3 clock

10: divide by 4 on selected column 3 clock

11: divide by 8 on selected column 3 clock

Bit 3: RS_SEL[0]. If set to 1, this bit will select the RS output to compare bus for analog column 2.

Bit 2: SYS0.

0: column 2 clock selection is controlled by CMPCLK0

1: column 2 clock selection is controlled by SYSCLK

Bits 1 and 0: DIVCLK0[1:0].

00: no divide on selected column 2 clock

01: divide by 2 on selected column 2 clock

10: divide by 4 on selected column 2 clock

11: divide by 8 on selected column 2 clock

For additional information, refer to the CMPCLK1 register on page 219.

CMPCLK0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,5Fh	CMPCLK0	RS_EN		SEL_CSD[1:0]		CMPCOL1[1:0]		CMPCOL0[1:0]		RW : 00

This register controls the analog column clock selection and RS feature with CMPCLK1.

Bit 7: RS_EN. This bit is used to enable RS function. Analog column 2's LUT output is master and column 3's output is slave.

Bit 5: SEL_CSD[1].

0: column 3 clock determined by CMPCOL1[1:0]

1: column 3 clock derives from related CSD_CNT

Bit 4: SEL_CSD[0].

0: column 2 clock determined by CMPCOL0[1:0]

1: column 2 clock derives from related CSD_CNT

Bits 3 and 2: CMPCOL1[1:0]. Clock selection for analog column 3.

00: VC1

01: VC2

10: analog clock 0

11: analog clock 1

Bits 1 and 0: CMPCOL0[1:0]. Clock selection for analog column 2.

00: VC1

01: VC2

10: analog clock 0

11: analog clock 1

For additional information, refer to the CMPCLK0 register on page 220.

18.3.3 Analog Input Configuration Registers

CMPCOLMUX Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,5Bh	CMPCOLMUX	ACMPS1[1:0]		LMXR	RMXL	ACI1[1:0]		ACI0[1:0]		RW : 00

This register controls the analog muxes that feed signals in from port pins into the analog compare column.

Bits 7 and 6: ACMPS1[1:0]. These two bits select the signal for column 3 compare bus output.

00: LUT out or RS_Lat Slave out

01: double sync version of above signal

10: positive one-shot from above signal

11: positive or negative one-shot from above signal

Bit 5: LMXR.

0: selects analog compare column 0 input from the output of the 4:1 MUX for P07/P05/P03/P01

1: selects analog compare column 0 input from the output of the 4:1 MUX for P06/P04/P02/P00

Bit 4: RMXL.

0: selects analog compare column 1 input from the output of the 4:1 MUX for P06/P04/P02/P00

1: selects analog compare column 1 input from the output of the 4:1 MUX for P07/P05/P03/P01

Bits 3 and 2: ACPI1[1:0]. These two bits select the input for a 4:1 input mux.

00: P0[0]

01: P0[2]

10: P0[4]

11: P0[6]

Bits 1 and 0: ACPI0[1:0]. These two bits select the input for a 4:1 input mux.

00: P0[1]

01: P0[3]

10: P0[5]

11: P0[7]

For additional information, refer to the CMPCOLMUX register on page 216.

18.3.4 Continuous Time PSoC Block Registers

The naming convention for the continuous time registers and their arrays of PSoC blocks is <Prefix>mn<Suffix>, where m=row index and n=column index. Therefore, ACE01CR2 (written ACExxCR2) is a register for an analog PSoC block in row 0 column 1.

CMPxCR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,50h	CMP0CR1		CompBus	NMux[2:0]			PMux[2:0]			RW : 00
1,54h	CMP1CR1		CompBus	NMux[2:0]			PMux[2:0]			RW : 00

The CMPxCR1 is one of two registers used to configure the type E continuous time PSoC block.

Bit 6: CompBus. This bit determines whether the comparator bus is driven from the amplifier output or driven low. If the CT block is configured in Unity Gain mode, this bit should be set to zero so the comparator bus is driven low.

Bits 5 to 3: NMux[2:0]. These bits control the multiplexing of inputs to the inverting input of the opamp. There are several input choices from outside the block, plus an internal feedback selection.

Bits 2 to 0: PMux[2:0]. These bits control the multiplexing of the five inputs to the non-inverting input of the opamp.

For additional information, refer to the CMPxCR1 register on page 210.

CMPxCR2 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,51h	CMP0CR2							FullRange	PWR	RW : 00
1,55h	CMP1CR2							FullRange	PWR	RW : 00

The CMPxCR2 is one of two registers used to configure the type E continuous time PSoC block.

Bit 1: FullRange. For slightly higher power, this bit enables inputs from Vss to Vdd.

Bit 0: PWR. This bit is used to power up the CT block and SC block in the column.

For additional information, refer to the CMPxCR2 register on page 211.

18.3.5 VDAC5 Block Register

The naming convention for the switched capacitor register and its array of PSoC blocks is <Prefix>mn<Suffix>, where m=row index and n=column index. Therefore, ASE10CR0 (written ASExxCR0) is a register for an analog PSoC block in row 1 column 0.

VDAC5xCR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,53h	VDAC50CR0				VDACIN[4:0]					RW : 00
1,57h	VDAC51CR0				VDACIN[4:0]					RW : 00

These two registers are used to configure the VDAC5 output voltage.

For additional information, refer to the VDAC5xCR0 register on page 212.

VDAC 2BG Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E7h	VDAC_TRIM		EN	V24	SAMPLE	TRIM[3:0]				RW : 00

For additional information, refer to the VDAC_TRIM register on page 260.

Section F: System Resources

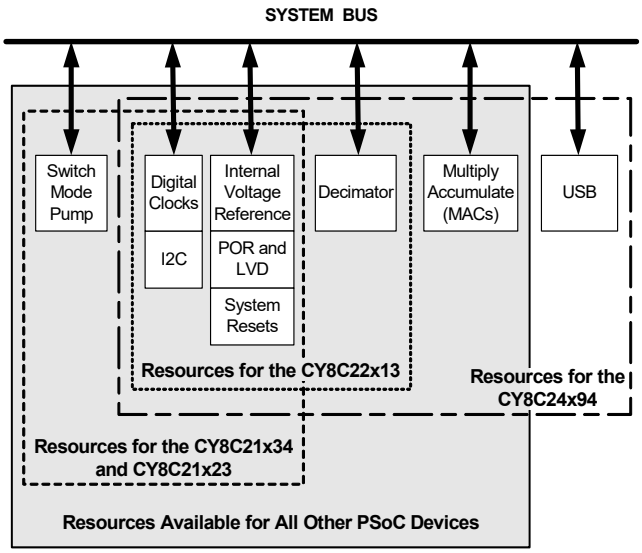
The System Resources section discusses the system resources that are available for the PSoC device and the registers associated with those resources. This section encompasses the following chapters:

- Digital Clocks on page 377
- Multiply Accumulate (MAC) on page 387
- I2C on page 392
- Internal Voltage Reference on page 410
- System Resets on page 412
- POR and LVD on page 419
- I/O Analog Multiplexer on page 422
- CSD Logic System chapter on page 429
- Real Time Clock (RTC) chapter on page 436
- 10-Bit SAR ADC Controller on page 439

Top-Level System Resources Architecture

The figure below displays the top-level architecture of the PSoC's system resources. Each component of the figure is discussed at length in this section. Note that the CY8C22x13 PSoC device does not support the Switch Mode Pump and Multiply Accumulate system resources and the CY8C21xxx PSoC devices do not support the Decimator and Multiply Accumulate system resources. All other PSoC devices support all the system resources found in this section.

PSoC System Resources



Interpreting the System Resources Documentation

Information in this section covers all PSoC devices with a base part number of CY8C2xxxx (except for the CY8C25122 and CY8C26xxx PSoC devices). It also applies to CY7C64215, CY7C603xx, and CYWUSB6953. The following table lists the resources available for specific device groups with a check mark or appropriate information. Blank fields denote that the system resource is not available. Note that the CY8C21x34, CY7C603xx, and CYWUSB6953 are the only PSoC devices that have the I/O Analog Multiplexer system resource and that the CY8C24x94 and CY7C64215 are the only PSoC devices that have USB functionality.

Availability of System Resources for PSoC Devices

PSoC Part Number	Digital Clocks	I2C	Internal Voltage Ref	POR and LVD	System Resets	Multiply Accumulate
CY8C29x66	✓	✓	✓	✓	✓	2
CY8C27x43	✓	✓	✓	✓	✓	1
CY8C24x94 **	✓	✓	✓	✓	✓	2
CY8C24x23	✓	✓	✓	✓	✓	1
CY8C24x23A	✓	✓	✓	✓	✓	1
CY8C22x45	✓	✓	✓	✓	✓	1
CY8C22x13	✓	✓	✓	✓	✓	0
CY8C21345	✓	✓	✓	✓	✓	1
CY8C21x34 **	✓	✓	✓	✓	✓	0
CY8C21x23	✓	✓	✓	✓	✓	0
CY7C64215 **	✓	✓	✓	✓	✓	2
CY7C603xx **	✓	✓	✓	✓	✓	0
CYWUSB6953 **	✓	✓	✓	✓	✓	0

** The PSoC devices that have the I/O Analog Multiplexer system resource.

System Resources Register Summary

The table below lists all the PSoC registers for the system resources, in address order, within their system resource configuration. The bits that are grayed out are reserved bits. If these bits are written, they should always be written with a value of '0'.

Note that all PSoC devices have a combination of 4, 2, or 1 analog columns and 4, 2 or 1 digital rows. The registers that are specifically constrained by the number of analog columns have the number of analog columns (Cols.) listed within the Address column of the table. The registers specifically pertaining to digital rows have the number of rows (Rows) listed within the Address column of the table. To determine the number of analog columns and digital rows in your PSoC device, refer to the table titled "PSoC Device Characteristics" on page 334.

Summary Table of the System Resource Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
DIGITAL CLOCK REGISTERS (page 381)										
0,DAh 4 Cols.	INT_CLR0	VC3	Sleep	GPIO	Analog 3	Analog 2	Analog 1	Analog 0	V Monitor	RW : 00
2 Cols.		VC3	Sleep	GPIO			Analog 1	Analog 0	V Monitor	RW : 00
1 Col.		VC3	Sleep	GPIO			Analog 1		V Monitor	RW : 00
0,E0h 4 Cols.	INT_MSK0	VC3	Sleep	GPIO	Analog 3	Analog 2	Analog 1	Analog 0	V Monitor	RW : 00
2 Cols.		VC3	Sleep	GPIO			Analog 1	Analog 0	V Monitor	RW : 00
1 Col.		VC3	Sleep	GPIO			Analog 1		V Monitor	RW : 00
1,DDh	OSC_GO_EN	SLPINT	VC3	VC2	VC1	SYSCLKX2	SYSCLK	CLK24M	CLK32K	RW : 00
1,DEh	OSC_CR4							VC3 Input Select[1:0]		RW : 00
1,DFh	OSC_CR3	VC3 Divider[7:0]								RW : 00
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00
1,E1h	OSC_CR1	VC1 Divider[3:0]				VC2 Divider[3:0]				RW : 00
1,E2h	OSC_CR2	PLLGAIN			SLP_EXTEND	WDR32_SE	EXTCLKEN	RSVD	SYSCLKX2DIS	RW : 00
MULTIPLY ACCUMULATE (MAC) REGISTERS (page 388)										
0,E8h	MUL0_X	Data[7:0]								W : XX
0,E9h	MUL0_Y	Data[7:0]								W : XX
0,EAh	MUL0_DH	Data[7:0]								R : XX
0,EBh	MUL0_DL	Data[7:0]								R : XX
0,ECh	MAC0_X/ ACC0_DR1	Data[7:0]								RW : 00
0,EDh	MAC0_Y/ ACC0_DR0	Data[7:0]								RW : 00
0,EEh	MAC0_CL0/ ACC0_DR3	Data[7:0]								RW : 00
0,EFh	MAC0_CL1/ ACC0_DR2	Data[7:0]								RW : 00
I2C REGISTERS (page 396)										
0,D6h	I2C_CFG		PSelect	Bus Error IE	Stop IE	Clock Rate[1:0]		Enable Master	Enable Slave	RW : 00
0,D7h	I2C_SCR	Bus Error	Lost Arb	Stop Status	ACK	Address	Transmit	LRB	Byte Complete	R : 00
0,D8h	I2C_DR	Data[7:0]								RW : 00
0,D9h	I2C_MSCR					Bus Busy	Master Mode	Restart Gen	Start Gen	R : 00
1,ADh	I2C_ADDR	HwAddrEn	Addr[6:0]							RW : 00
INTERNAL VOLTAGE REFERENCE REGISTER (page 410)										
1,EAh 4,2 Cols.	BDG_TR		AGNDBYP	TC[1:0]		V[3:0]				RW : 00
1 Col.				TC[1:0]		V[3:0]				RW : 00

Summary Table of the System Resource Registers *(continued)*

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
SYSTEM RESET REGISTERS (page 414)										
0,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW	ECO EX		IRAMDIS	# : 00
0,FFh	CPU_SCR0	GIES		WDRS	PORS	Sleep			STOP	# : XX
POR AND LVD REGISTERS (page 419)										
1,E3h 4,2 Cols.	VLT_CR	SMP		PORLEV[1:0]		LVDTBEN	VM[2:0]			RW : 00
1 Col.				PORLEV[1:0]		LVDTBEN	VM[2:0]			RW : 00
1,E4h 4,2 Cols.	VLT_CMP						PUMP	LVD	PPOR	R : 00
2L** Cols.						NoWrite	PUMP	LVD	PPOR	R : 00
1 Col.								LVD	PPOR	R : 00
I/O ANALOG MULTIPLEXER REGISTERS (page 426)										
0,61h	AMUX_CFG	BCol1Mux	ACol0Mux	INTCAP[1:0]		MUXCLK[2:0]			EN	RW : 00
0,FCh	IDACR_D	DACDATA[7:0]								RW : 00
0,FDh	IDACL_D	DACDATA[7:0]								RW : 00
1,6Ah	AMUX_CFG1			GOINVR	GOINVL	AMUXMODL		AMUXMODR		RW : XX
1,D8h	MUX_CR0	ENABLE[7:0]								RW : 00
1,D9h	MUX_CR1	ENABLE[7:0]								RW : 00
1,DAh	MUX_CR2	ENABLE[7:0]								RW : 00
1,DBh	MUX_CR3	ENABLE[7:0]								RW : 00
1,ECh	MUX_CR4	ENABLE[7:0]								RW : 00
1,DCh	IDAC_CR1	ENR	MuxClkGER	ICEN	IDAC_TRIM				Double_Current	RW : 00
1,FDh	IDAC_CR0	SplitMux	MuxClkGEL	OSCMDR[1:0]		IRANGEL	OSCMDL[1:0]		ENL	RW : 00
CSD LOGIC SYSTEM REGISTERS (page 429)										
0,50h	CSD0_DR0_L	Data[7:0]								R : 00
0,51h	CSD0_DR1_L	Data[7:0]								W : 00
0,52h	CSD0_CNT_L	Data[7:0]								RC : 00
0,53h	CSD0_CR0	MSHOT[3:0]				CNT_RST	CSD	PRS	EN	RW : 00
0,54h	CSD0_DR0_H	Data[7:0]								R : 00
0,55h	CSD0_DR1_H	Data[7:0]								W : 00
0,56h	CSD0_CNT_H	Data[7:0]								RC : 00
0,57h	CSD0_CR1	INT_TYPE	CSD_CKSEL[2:0]			CNT_CKSEL[1:0]		ACOL[1:0]		RW : 00
0,58h	CSD1_DR0_L	Data[7:0]								R : 00
0,59h	CSD1_DR1_L	Data[7:0]								W : 00
0,5Ah	CSD1_CNT_L	Data[7:0]								RC : 00
0,5Bh	CSD1_CR0	MSHOT[3:0]				CNT_RST	CSD	PRS	EN	RW : 00
0,5Ch	CSD1_DR0_H	Data[7:0]								R : 00
0,5Dh	CSD1_DR1_H	Data[7:0]								W : 00
0,5Eh	CSD1_CNT_H	Data[7:0]								RC : 00
0,5Fh	CSD1_CR1	INT_TYPE	CSD_CKSEL[2:0]			CNT_CKSEL[1:0]		ACOL[1:0]		RW : 00
REAL TIME CLOCK (RTC) REGISTERS (page 436)										
0,6Ah	RTCH			HR1[1:0]		HR0[3:0]				R : 00
0,6Bh	RTCM		MIN1[2:0]			MIN0[3:0]				R : 00
1,A8h	RTCS		SEC[2:0]			SEC[3:0]				RW : 00
1,A9h	RTCCR	TREG[1:0]		INT_EN	CLKSE	INT_SEL[1:0]		SYNC_RD	RT_EN	RW : 00
10-BIT SAR ADC CONTROLLER REGISTERS (page 442)										
0,6Ah	ADC_DH	ADC_DH[7:0]								R : 00
0,6Bh	ADC_DL							ADC_DL[1:0]		R : 00
1,A8h	ADC_CR0	ADC_TST1	ADC_CHS[3:0]				READY	START/ ONGOING	ADC_EN	RW : 00
1,A9h	ADC_CR1	CVTMD[1:0]		TIGSEL[1:0]		CLKSEL[2:0]			ALIGN_EN	RW : 00

Summary Table of the System Resource Registers *(continued)*

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,AAh	ADC_CR2	REFSEL	BUFEN	VDBEN	VDB_CLKSEL	FREERUN	ADC_EXT_HALFVDD	ADC_MODE[1:0]		RW : 00
1,ABh	ADC_CR3TRIM						ADC_TRIM0[2:0]			RW : 04
1,ACh	ADC_CR4		D[13]	D[11]	D[9]	VDBCLK	ADCCLK	ADC_TST2	ADC_CMP	RW : 02

LEGEND
X The value after power on reset is unknown.
C Clearable register or bits.
R Read register or bit(s).
W Write register or bit(s).
Access is bit specific. Refer to the Register Details chapter on page 103 for additional information.
** The 2L column row is only applicable to the CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices which have two column limited analog functionality.

19. Digital Clocks

This chapter discusses the Digital Clocks and their associated registers. It serves as an overview of the clocking options available in the PSoC devices. For detailed information on specific oscillators, see the individual oscillator chapters in the section called "PSoC Core" on page 29. For a complete table of the digital clock registers, refer to the "Summary Table of the System Resource Registers" on page 374. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

19.1 Architectural Description

The PSoC M8C core has a large number of clock sources that increase the flexibility of the PSoC Programmable System-on-Chip, as listed in Table 19-1 and illustrated in Figure 19-1.

Table 19-1. System Clocking Signals and Definitions

Signal	Definition
SYSCLKX2	Twice the frequency of SYSCLK.
SYSCLK	Either the direct output of the Internal Main Oscillator or the direct input of the EXTCLK pin while in external clocking mode.
CPUCLK	SYSCLK is divided down to one of eight possible frequencies, to create CPUCLK which determines the speed of the M8C. See OSC_CR0 in the Register Definitions section of this chapter.
VC1	SYSCLK is divided down to create Variable Clock 1 (VC1). See OSC_CR1 in the Register Definitions section of this chapter. Division range is from 1 to 16.
VC2	VC1 is divided down to create Variable Clock 2 (VC2). See OSC_CR1 in the Register Definitions section of this chapter. Division range is from 1 to 16.
VC3	Divides down either SYSCLK, VC1, VC2, or SYSCLKX2 to create Variable Clock 3 (VC3). Division range is from 1 to 256. See OSC_CR3 and OSC_CR4 in the Register Definitions section of this chapter.
CLK32K	Either the Internal Low Speed Oscillators or the External Crystal Oscillators output. See OSC_CR0 in the Register Definitions section of this chapter.
CLK24M	The internally generated 24 MHz clock by the IMO. By default, this clock drives SYSCLK; however, an external clock may be used by enabling EXTCLK mode. Also, the IMO may be put into a slow mode using the SLIMO bit which will change the speed of the IMO and the CLK24M to either 6 MHz or 12 MHz in some PSoC devices.
SLEEP	One of four sleep intervals may be selected from 1.95 ms to 1 second. See OSC_CR0 in the Register Definitions section of this chapter.

19.1.1 Internal Main Oscillator

The Internal Main Oscillator (IMO) is the foundation upon which almost all other clock sources in the PSoC Programmable System-on-Chip are based. The default mode of the IMO creates a 24 MHz reference clock that is used by many other circuits in the PSoC device. The IMO may also be configured to operate in a PLL mode where the oscillator is locked to a precision 32.768 kHz crystal reference. The PSoC device has an option to replace the IMO with an externally supplied clock that will become the base for all of the clocks the IMO normally serves. The internal base clock net is called SYSCLK and may be driven by either the IMO or an external clock (EXTCLK).

Whether the external clock or the internal main oscillator is selected, all PSoC device functions are clocked from a derivative of SYSCLK or are resynchronized to SYSCLK. All external asynchronous signals (through row inputs), as well as the selected 32.768 kHz crystal oscillator, are resynchronized to SYSCLK for use in the digital PSoC blocks.

Some PSoC devices contain the option to lower the internal oscillator's system clock from 24 MHz to 6 MHz. See the "Architectural Description" on page 75, in the Internal Main Oscillator chapter, for more information.

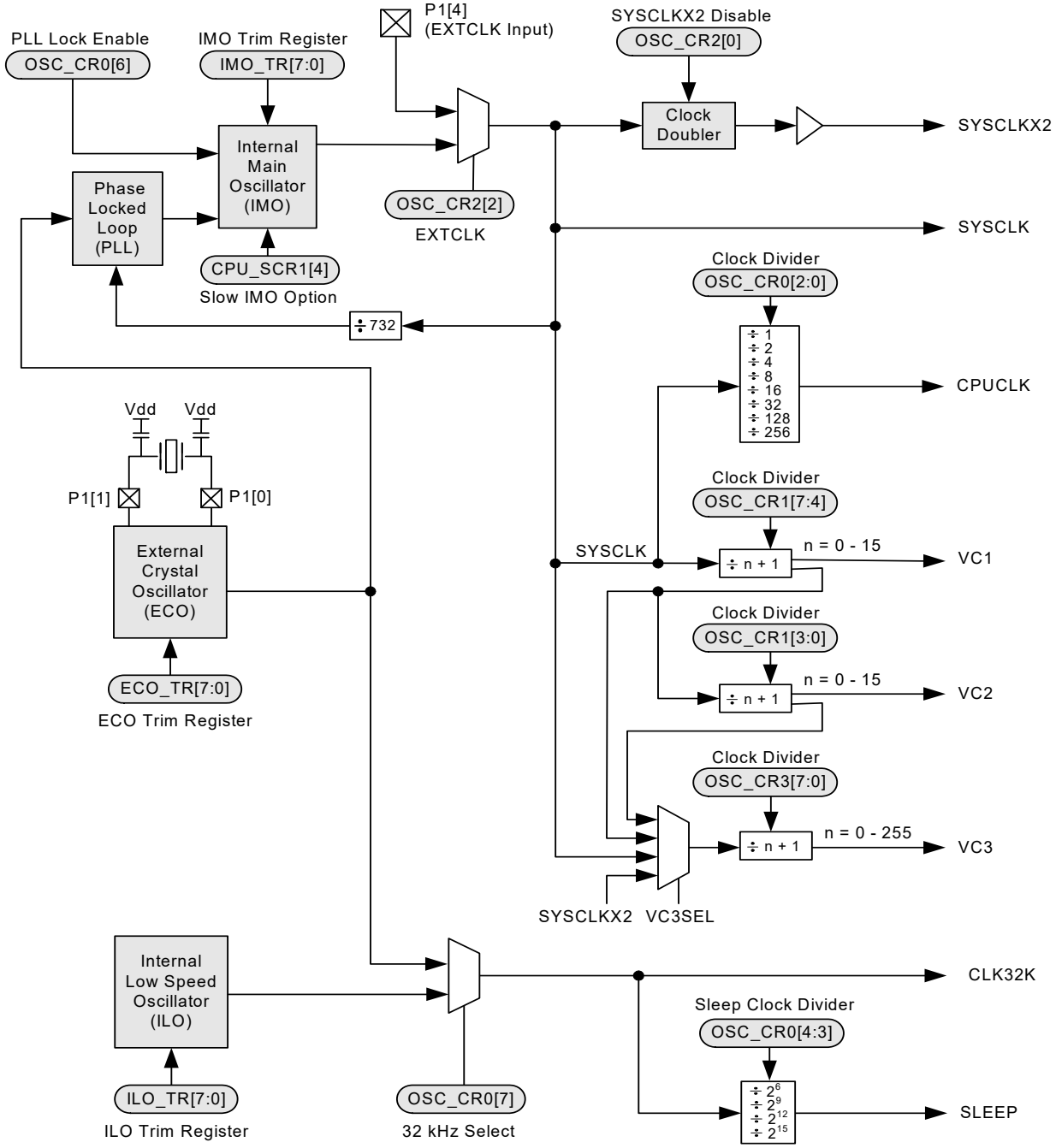
The IMO is discussed in detail in the chapter "Internal Main Oscillator (IMO)" on page 75.

19.1.2 Internal Low Speed Oscillator

The Internal Low Speed Oscillator (ILO) is always on unless the device is operating off an external crystal. The ILO is available as a general clock, but is also the clock source for the sleep and watchdog timers.

The ILO is discussed in detail in the chapter "Internal Low Speed Oscillator (ILO)" on page 79.

Figure 19-1. Overview of PSoC Clock Sources



19.1.3 32.768 kHz Crystal Oscillator

The PSoC may be configured to use an external crystal. The crystal oscillator is discussed in detail in the chapter “External Crystal Oscillator (ECO)” on page 80.

19.1.4 External Clock

The ability to replace the 24 MHz internal main oscillator (IMO), as the device master system clock (SYSCLK) with an externally supplied clock, is a feature in the PSoC Programmable System-on-Chip (see Figure 19-1).

Pin P1[4] is the input pin for the external clock. This pin was chosen because it is not associated with any special features such as analog I/O, crystal, or In-System Serial Programming (ISSP). It is also not physically close to either the P1[0] and P1[1] crystal pins. If P1[4] is selected as the external clock source, the drive mode of the pin must be set to High Z (not High Z analog).

The user is able to supply an external clock with a frequency between 1 MHz and 24 MHz. The reset state of the EXTCLKEN bit is ‘0’; and therefore, the device always boots up under the control of the IMO. There is no way to start the system from a reset state with the external clock.

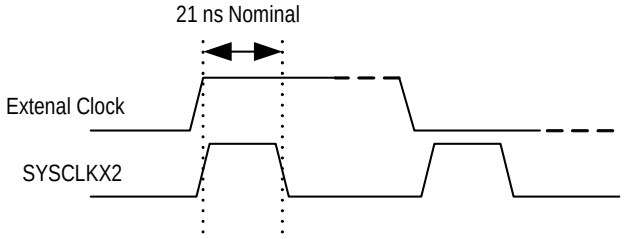
When the EXTCLKEN bit is set, the external clock becomes the source for the internal clock tree, SYSCLK, which drives most PSoC device clocking functions. All external and internal signals, including the 32 kHz clock, whether derived from the internal low speed oscillator (ILO) or the crystal oscillator, are synchronized to this clock source. Note that there is no glitch protection in the device for an external clock. User should ensure that the external clock is glitch free. See device datasheet for the clock specifications.

19.1.4.1 Clock Doubler

One of the blocks driven by the system clock is the clock doubler circuit that drives the SYSCLKX2 output. This doubled clock, which is 48 MHz when the IMO is the selected clock (at 24 MHz), may be used as a clock source for the digital PSoC blocks. When the external clock is selected, the SYSCLKX2 signal is still available and serves as a doubler for whatever frequency is input on the external clock pin.

Following the specification for the external clock input ensures that the internal circuitry of the digital PSoC blocks, which is clocked by SYSCLKX2, will meet timing requirements. However, because the doubled clock is generated from both edges of the input clock, clock jitter is introduced if the duty cycle deviates greatly from 50 percent. Also, the high time of the clock out of the doubler is fixed at 21 ns, so the duty cycle of SYSCLKX2 is proportional to the inverse of the frequency, as shown in Figure 19-2. Regardless of the input frequency, the high period of SYSCLKX2 is 21 ns nominal.

Figure 19-2. Operation of the Clock Doubler



19.1.4.2 Switch Operation

Switching between the IMO and the external clock may be done in firmware at any time and is transparent to the user. Because all PSoC device resources run on clocks derived from or synchronized to SYSCLK, when the switch is made, analog and digital functions may be momentarily interrupted.

Switch timing depends on whether the CPU clock divider is set for divide by 1, or divide by 2 or greater. In the case where the CPU clock divider is set for divide by 2 or greater, as shown in Figure 19-3, the setting of the EXTCLKEN bit occurs shortly after the rising edge of SYSCLK. The SYSCLK output is then disabled after the next falling edge of SYSCLK, but before the next rising edge. This ensures a glitch-free transition and provides a full cycle of setup time from SYSCLK to output disable. Once the current clock selection is disabled, the enable of the newly selected clock is double synchronized to that clock. After synchronization, on the subsequent negative edge, SYSCLK is enabled to output the newly selected clock.

In the 24 MHz case, as shown in Figure 19-4, the assertion of IOW_ and thus the setting of the EXTCLKEN bit occurs on the falling edge of SYSCLK. Because SYSCLK is already low, the output is immediately disabled. Therefore, the setup time from SYSCLK to disable is one-half SYSCLK.

Figure 19-3. Switch from IMO to the External Clock with a CPU Clock Divider of Two or Greater

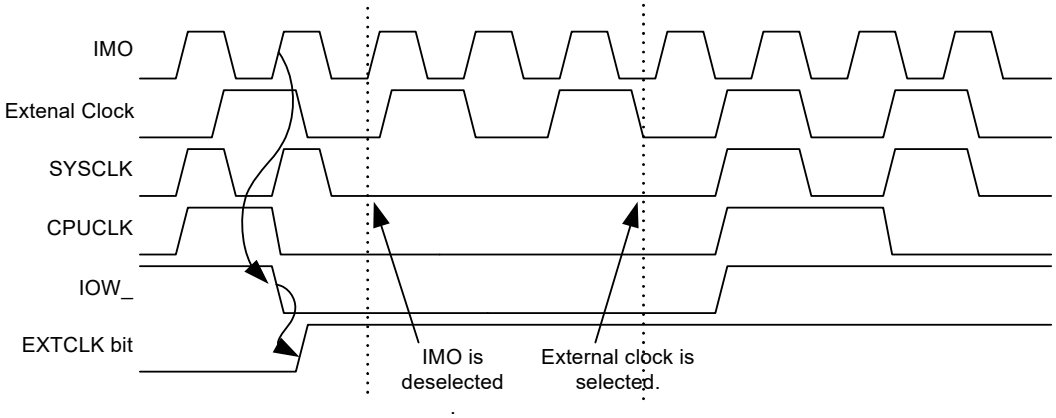
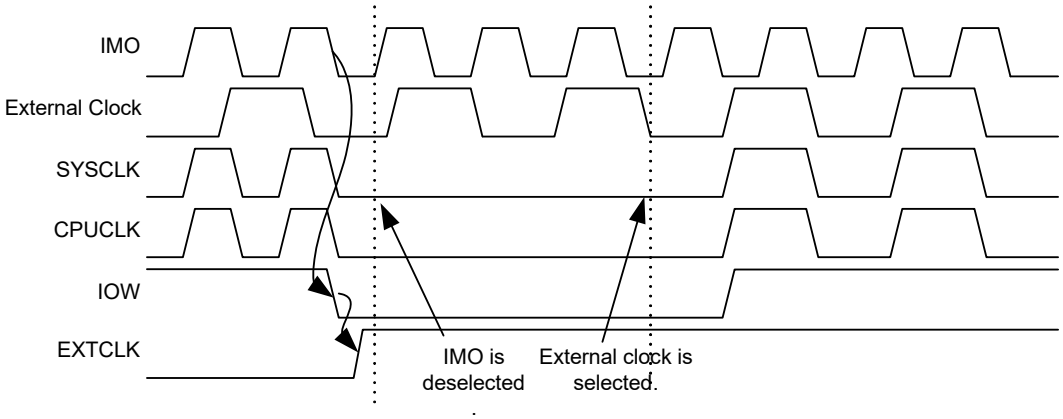


Figure 19-4. Switch from IMO to External Clock with the CPU Running with a CPU Clock Divider of One



19.2 PSoC Device Distinctions

The PSoC device distinctions that apply to the digital clocks are listed as follows.

- In PSoC devices with a part number of CY8C27x43, bits 7, 6, 5, and 4 of the OSC_GO_EN register are reserved. See the OSC_GO_EN register on page 251 for more information.
- In PSoC devices with a part number of CY8C24x23 or CY8C22x13, bit 7 of the OSC_GO_EN register is reserved. However, in PSoC devices with a part number of CY8C24x23A, bit 7 is not reserved.
- For Silicon Revision A of the CY8C27x43 PSoC device, only digital blocks DBC01, DCC02, DBC01, and DCC12 are valid in the DEC_CR1 register. See the DEC_CR1 register on page 179 for more information.

19.3 Register Definitions

The following registers are associated with the Digital Clocks and are listed in address order. Each register description has an associated register table showing the bit structure for that register. For a complete table of digital clock registers, refer to the "Summary Table of the System Resource Registers" on page 374.

Depending on your PSoC device's configuration (refer to the table titled "PSoC Device Characteristics" on page 334), only certain bits are accessible to be read or written, such as the INT_CLR0 and INT_MSK0 registers that are analog column dependent (see the "Cols." column in the tables below). The bits in the tables that are grayed out throughout this manual are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of '0'.

19.3.1 INT_CLR0 Register

Add.	Name	Cols.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,DAh	INT_CLR0	4, 3	VC3	Sleep	GPIO	Analog 3	Analog 2	Analog 1	Analog 0	V Monitor	RW : 00
		2	VC3	Sleep	GPIO			Analog 1	Analog 0	V Monitor	RW : 00
		1	VC3	Sleep	GPIO			Analog 1		V Monitor	RW : 00

The Interrupt Clear Register 0 (INT_CLR0) is used to enable the individual interrupt sources' ability to clear posted interrupts.

Bit 7: VC3. The digital clocks only use bit 7 of the INT_CLR0 register for the VC3 clock. This bit controls the VC3 clock interrupt status.

Bits 6 to 0. The INT_CLR0 register holds bits that are used by several different resources. For a full discussion of the INT_CLR0 register, see the INT_CLRx Registers in the Interrupt Controller chapter on page 60.

For additional information, refer to the INT_CLR0 register on page 166.

19.3.2 INT_MSK0 Register

Add.	Name	Cols.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E0h	INT_MSK0	4, 3	VC3	Sleep	GPIO	Analog 3	Analog 2	Analog 1	Analog 0	V Monitor	RW : 00
		2	VC3	Sleep	GPIO			Analog 1	Analog 0	V Monitor	RW : 00
		1	VC3	Sleep	GPIO			Analog 1		V Monitor	RW : 00

The Interrupt Mask Register 0 (INT_MSK0) is used to enable the individual sources' ability to create pending interrupts.

Bit 7: VC3. The digital clocks only use bit 7 of the INT_CLR0 register for the VC3 clock. This bit controls the VC3 clock interrupt enable.

Bits 6 to 0. The INT_MSK0 register holds bits that are used by several different resources. For a full discussion of the INT_MSK0 register, see the INT_MSKx Registers in the Interrupt Controller chapter on page 60.

For additional information, refer to the INT_MSK0 register on page 174.

19.3.3 OSC_GO_EN Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,DDh	OSC_GO_EN	SLPINT	VC3	VC2	VC1	SYSCLKX2	SYSCLK	CLK24M	CLK32K	RW : 00

The Oscillator to Global Outputs Enable Register (OSC_GO_EN) is used to enable tri-state buffers that connect specific system clocks to specific global output even nets.

The OSC_GO_EN register holds eight bits which independently enable a tri-state buffer to drive a clock on to a global net. In all cases, the clock is driven on to one of the nets in the Global Output Even (GOE) bus. In all cases,

these bits should only be set and the resulting clock signal on the global be used when the clock frequency is less than or equal to the maximum ***switching*** frequency of the global buses (12 MHz). Therefore, bits 2 and 3 are only useful when the PSoC device is in external clocking mode and bit 1 may never be used.

Note See "PSoC Device Distinctions" on page 380 for more information about this register.

Bit 7: SLPINT. This bit provides the option to connect the sleep interrupt signal to GOE[7]. This may be useful in real-time clock applications where very low power is required. By driving the sleep interrupt to a global, it may then be routed to a digital PSoC block. The digital PSoC block may then count several sleep interrupts before generating its own interrupt, which would be used to bring the PSoC device out of the sleep state.

Bit 6: VC3. This bit enables the driving of the VC3 clock onto GOE[6].

Bit 5: VC2. This bit enables the driving of the VC3 clock onto GOE[5].

Bit 4: VC1. This bit enables the driving of the VC3 clock onto GOE[4].

Bit 3: SYSCLKX2. This bit enables the driving of the SYSCLKX2 clock onto GOE[3].

Bit 2: SYSCLK. This bit enables the driving of the SYSCLK clock onto GOE[2].

Bit 1: CLK24M. This bit enables the driving of the 24 Mhz clock onto GOE[1].

Bit 0: CLK32K. This bit enables the driving of the 32 kHz clock onto GOE[0].

For additional information, refer to the OSC_GO_EN register on page 251.

19.3.4 OSC_CR4 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,DEh	OSC_CR4							VC3 Input Select[1:0]		RW : 00

The Oscillator Control Register 4 (OSC_CR4) selects the input clock to variable clock 3 (VC3).

Bits 1 and 0: VC3 Input Select [1:0]. The VC3 clock net is the only clock net with the ability to generate an interrupt. The input clock of VC3 comes from a configurable source. As shown in Figure 19-1 on page 378, a 4-to-1 mux determines the clock that is used in the input to the VC3 divider. The mux allows either the 48 MHz, 24 MHz, VC1, or VC2 clocks to be used as the input clock to the divider. Because the selection of a clock for the VC3 divider is performed by a simple 4-to-1 mux, ***runt pulses*** and glitches may be injected to the VC3 divider when the OSC_CR4[1:0] bits are changed. Care should be taken to ensure that blocks using the VC3 clock are either disabled when OSC_CR4[1:0] is changed or not sensitive to glitches. Unlike the VC1 and VC2 clock dividers, the VC3 clock divider is 8-bits wide. Therefore, there are 256 valid divider values as indicated by Table 19-3.

It is important to remember that even though the VC3 divider has four choices for the input clock, none of the choices have fixed frequencies for all device configurations. Both the 24 MHz and 48 MHz clocks may have very different frequencies if an external clock is in use. Also, the divider values for the VC1 and VC2 inputs to the mux must be considered.

Table 19-2. OSC_CR4[1:0] Bits: VC3

Bits	Multiplexer Output
00b	SYSCLK
01b	VC1
10b	VC2
11b	SYSCLKX2

For additional information, refer to the OSC_CR4 register on page 252.

19.3.5 OSC_CR3 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,DFh	OSC_CR3	VC3 Divider[7:0]								RW : 00

The Oscillator Control Register 3 (OSC_CR3) selects the divider value for variable clock 3 (VC3).

Bits 7 to 0: VC3 Divider[7:0]. As an example of the flexibility of the clocking structure in PSoC devices, consider a device that is running off of an externally supplied clock at a frequency of 93.7 kHz. This clock value may be divided by

the VC1 divider to achieve a VC1 clock net frequency of 5.89 kHz. The VC2 divider could reduce the frequency by another factor of 16, resulting in a VC2 clock net frequency of 366.02 Hz. Finally, the VC3 divider may choose VC2 as its input clock and divide by 256, resulting in a VC3 clock net frequency of 1.43 Hz.

Table 19-3. OSC_CR3[7:0] Bits: VC3 Divider Value

Bits	Divider Source Clock			
	SYSCLKX2	SYSCLK	VC1	VC2
00h	SYSCLKX2	SYSCLK	VC1	VC2
01h	SYSCLKX2 / 2	SYSCLK / 2	VC1 / 2	VC2 / 2
02h	SYSCLKX2 / 3	SYSCLK / 3	VC1 / 3	VC2 / 3
03h	SYSCLKX2 / 4	SYSCLK / 4	VC1 / 4	VC2 / 4
...	...	...	...	...
FCh	SYSCLKX2 / 253	SYSCLK / 253	VC1 / 253	VC2 / 253
FDh	SYSCLKX2 / 254	SYSCLK / 254	VC1 / 254	VC2 / 254
FEh	SYSCLKX2 / 255	SYSCLK / 255	VC1 / 255	VC2 / 255
FFh	SYSCLKX2 / 256	SYSCLK / 256	VC1 / 256	VC2 / 256

The VC3 clock net can generate a system interrupt. Once the input clock and the divider value for the VC3 clock are chosen, only one additional step is needed to enable the interrupt; the VC3 mask bit must be set in register INT_MSK0[7]. Once the VC3 mask bit is set, the VC3 clock generates pending interrupts every number of clock periods equal to the VC3 divider register value plus one. Therefore, if the VC3 divider register's value is 05h (divide by 6), an interrupt would occur every six periods of the VC3's input clock. Another example would be if the divider value was 00h (divide by one), an interrupt would be generated on every period of the VC3 clock. The VC3 mask bit only controls the ability of a posted interrupt to become pending. Because there is no enable for the VC3 interrupt, VC3 interrupts will always be posting. See the Interrupt Controller chapter on page 60 for more information on posting and pending.

For additional information, refer to the OSC_CR3 register on page 253.

19.3.6 OSC_CR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E0h	OSC_CR0	32k Select	PLL Mode	No Buzz	Sleep[1:0]		CPU Speed[2:0]			RW : 00

The Oscillator Control Register 0 (OSC_CR0) is used to configure various features of internal clock sources and clock nets.

Note Bits 7 and 6 in this register cannot be used by the CY8C21xxx, CY8C24x94, and CY7C64215 PSoC devices.

Bit 7: 32k Select. By default, the 32 kHz clock source is the Internal Low-Speed Oscillator (ILO). Optionally, the 32.768 kHz External Crystal Oscillator (ECO) may be selected.

Bit 6: PLL Mode. This bit is the only bit that directly influences the PLL. When set, it enables the PLL. The EXTCLK bit should be set low during PLL operation.

Bit 5: No Buzz. Normally, when the Sleep bit is set in the CPU_SCR register, all PSoC device systems are powered down, including the bandgap reference. However, to facilitate the detection of POR and LVD events at a rate higher than the sleep interval, the bandgap circuit is powered up periodically (for about 60 µs) at the Sleep System Duty Cycle (set in ECO_TR), which is independent of the sleep interval and typically higher. When the No Buzz bit is set, the Sleep System Duty Cycle value is overridden and the bandgap circuit is forced to be on during sleep. This results in faster response to an LVD or POR event (continuous detection as opposed to periodic), at the expense of slightly higher average sleep current.

Bits 4 and 3: Sleep[1:0]. The available sleep interval selections are shown in Table 19-4. Remember that when the ILO is the selected 32 kHz clock source, sleep intervals are approximate.

Table 19-4. Sleep Interval Selections

OSC_CR 2[4]	Sleep Interval OSC_CR0[4:3]	Sleep Timer Clocks	Sleep Period (nominal)	Watchdog Period (nominal)
0	00b (default)	64	1.95 ms	6 ms
0	01b	512	15.6 ms	47 ms
0	10b	4,096	125 ms	375 ms
0	11b	32,768	1 sec	3 sec
1	00b (default)	65,536	2 sec	6 sec
1	01b	131,072	4 sec	12 sec
1	10b	262,144	8 sec	24 sec
1	11b	524,288	16 sec	48 sec

Bits 2 to 0: CPU Speed[2:0]. The PSoC M8C may operate over a range of CPU clock speeds (Table 19-5), allowing the M8C's performance and power requirements to be tailored to the application.

The reset value for the CPU speed bits is zero. Therefore, the default CPU speed is one-eighth of the clock source. The internal main oscillator is the default clock source for the CPU speed circuit; therefore, the default CPU speed is 3 MHz. See "External Clock" on page 379 for more information on supported frequencies for externally supplied clocks.

The CPU frequency is changed with a write to the OSC_CR0 register. There are eight frequencies generated from a power-of-two divide circuit, which are selected by a 3-bit code. At any given time, the CPU 8-to-1 clock mux is selecting one of the available frequencies, which is resynchronized to the 24 MHz master clock at the output.

Regardless of the CPU speed bit's setting, if the actual CPU speed is greater than 12 MHz, the 24 MHz operating requirements apply. An example of this scenario is a device that is configured to use an external clock, which is supplying a frequency of 20 MHz. If the CPU speed register's value is 0x03, the CPU clock is 20 MHz. Therefore, the supply voltage requirements for the device are the same as if the part was operating at 24 MHz off of the internal main oscillator. The operating voltage requirements are not relaxed until the CPU speed is at 12.0 MHz or less.

Some devices support the slow IMO option, as discussed in the IMO chapter in the "Architectural Description" on page 75. This offers an option to lower both system and CPU clock speed in order to save power.

Table 19-5. OSC_CR0[2:0] Bits: CPU Speed

Bits	6 MHz Internal Main Oscillator *	24 MHz Internal Main Oscillator	External Clock
000b	750 kHz	3 MHz	EXTCLK/ 8
001b	1.5 MHz	6 MHz	EXTCLK/ 4
010b	3 MHz	12 MHz	EXTCLK/ 2
011b **	6 MHz	24 MHz	EXTCLK/ 1
100b	375 kHz	1.5 MHz	EXTCLK/ 16
101b	187.5 kHz	750 kHz	EXTCLK/ 32
110b	93.7 kHz	187.5 kHz	EXTCLK/ 128
111b	23.4 kHz	93.7 kHz	EXTCLK/ 256

* For PSoC devices that support the slow IMO option, see the "Architectural Description" on page 75.

** Selection 011b (24MHz) is not available for CY7C603xx due to lower operating voltage.

An automatic protection mechanism is available for systems that need to run at peak CPU clock speed but cannot guarantee a high enough supply voltage for that clock speed. See the LVDTBEN bit in the "VLT_CR Register" on page 420 for more information.

For additional information, refer to the OSC_CR0 register on page 254.

19.3.7 OSC_CR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E1h	OSC_CR1	VC1 Divider[3:0]				VC2 Divider[3:0]				RW : 00

The Oscillator Control Register 1 (OSC_CR1) selects the divider value for variable clocks 1 and 2 (VC1 and VC2).

Bits 7 to 4: VC1 Divider[3:0]. The VC1 clock net is one of the variable clock nets available in the PSoC M8C. The source for the VC1 clock net is a simple 4-bit divider. The source for the divider is the 24 MHz system clock; however, if the device is configured to use an external clock, the input to the divider is the external clock. Therefore, the VC1 clock net is not always the result of dividing down a 24 MHz clock. The 4-bit divider that controls the VC1 clock net may be configured to divide, using any integer value between 1 and 16. Table 19-6 lists all values for the VC1 clock net.

Bits 3 to 0: VC2 Divider[3:0]. The VC2 clock net is one of the variable clock nets available in the PSoC M8C. The source for the VC2 clock net is a simple 4-bit divider. The source for the divider is the VC1 clock net. The 4-bit divider that controls the VC2 clock net may be configured to divide, using any integer value between 1 and 16. Table 19-7 lists all values for the VC2 clock net.

Table 19-6. OSC_CR1[7:4] Bits: VC1 Divider Value

Bits	Divider Source Clock	
	Internal Main Oscillator at 24 MHz	External Clock
0h	24 MHz	EXTCLK / 1
1h	12 MHz	EXTCLK / 2
2h	8 MHz	EXTCLK / 3
3h	6 MHz	EXTCLK / 4
4h	4.8 MHz	EXTCLK / 5
5h	4 MHz	EXTCLK / 6
6h	3.43 MHz	EXTCLK / 7
7h	3 MHz	EXTCLK / 8
8h	2.67 MHz	EXTCLK / 9
9h	2.40 MHz	EXTCLK / 10
Ah	2.18 MHz	EXTCLK / 11
Bh	2.00 MHz	EXTCLK / 12
Ch	1.85 MHz	EXTCLK / 13
Dh	1.71 MHz	EXTCLK / 14
Eh	1.6 MHz	EXTCLK / 15
Fh	1.5 MHz	EXTCLK / 16

Table 19-7. OSC_CR1[3:0] Bits: VC2 Divider Value

Bits	Divider Source Clock	
	Internal Main Oscillator	External Clock
0h	(24 / (OSC_CR1[7:4]+1)) / 1	(EXTCLK / (OSC_CR1[7:4]+1)) / 1
1h	(24 / (OSC_CR1[7:4]+1)) / 2	(EXTCLK / (OSC_CR1[7:4]+1)) / 2
2h	(24 / (OSC_CR1[7:4]+1)) / 3	(EXTCLK / (OSC_CR1[7:4]+1)) / 3
3h	(24 / (OSC_CR1[7:4]+1)) / 4	(EXTCLK / (OSC_CR1[7:4]+1)) / 4
4h	(24 / (OSC_CR1[7:4]+1)) / 5	(EXTCLK / (OSC_CR1[7:4]+1)) / 5
5h	(24 / (OSC_CR1[7:4]+1)) / 6	(EXTCLK / (OSC_CR1[7:4]+1)) / 6
6h	(24 / (OSC_CR1[7:4]+1)) / 7	(EXTCLK / (OSC_CR1[7:4]+1)) / 7
7h	(24 / (OSC_CR1[7:4]+1)) / 8	(EXTCLK / (OSC_CR1[7:4]+1)) / 8
8h	(24 / (OSC_CR1[7:4]+1)) / 9	(EXTCLK / (OSC_CR1[7:4]+1)) / 9
9h	(24 / (OSC_CR1[7:4]+1)) / 10	(EXTCLK / (OSC_CR1[7:4]+1)) / 10
Ah	(24 / (OSC_CR1[7:4]+1)) / 11	(EXTCLK / (OSC_CR1[7:4]+1)) / 11
Bh	(24 / (OSC_CR1[7:4]+1)) / 12	(EXTCLK / (OSC_CR1[7:4]+1)) / 12
Ch	(24 / (OSC_CR1[7:4]+1)) / 13	(EXTCLK / (OSC_CR1[7:4]+1)) / 13
Dh	(24 / (OSC_CR1[7:4]+1)) / 14	(EXTCLK / (OSC_CR1[7:4]+1)) / 14
Eh	(24 / (OSC_CR1[7:4]+1)) / 15	(EXTCLK / (OSC_CR1[7:4]+1)) / 15
Fh	(24 / (OSC_CR1[7:4]+1)) / 16	(EXTCLK / (OSC_CR1[7:4]+1)) / 16

For additional information, refer to the OSC_CR1 register on page 255.

19.3.8 OSC_CR2 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E2h	OSC_CR2	PLLGAIN			SLP_EXTEND	WDR32_SE	EXTCLKEN	RSVD	SYSCLKX2DIS	RW : 00

The Oscillator Control Register 2 (OSC_CR2) is used to configure various features of internal clock sources and clock nets.

Bit 7: PLLGAIN. This is the only bit in the OSC_CR2 register that directly influences the PLL. When set, this bit keeps the PLL in a Low Gain mode. If this bit is held low, the lock time is less than 10 ms. If this bit is held high, the lock time is on the order of 50 ms. After lock is achieved, it is recommended that this bit be forced high to decrease the jitter on the output. If longer lock time is tolerable, the PLLGAIN bit can be held high all the time.

Bit 2: EXTCLKEN. When the EXTCLKEN bit is set, the external clock becomes the source for the internal clock tree, SYSCLK, which drives most PSoC device clocking functions. All external and internal signals, including the 32 kHz clock, whether derived from the internal low speed oscillator (ILO) or the crystal oscillator, are synchronized to this clock source. If an external clock is enabled, PLL mode should be off. The external clock input is located on port P1[4]. When using this input, the pin drive mode should be set to High Z (not High Z analog).

Bit 1: RSVD. Reserved bit - This bit should always be 0.

Bit 0: SYSCLKX2DIS. When set, the Internal Main Oscillator's doubler is disabled. This results in a reduction of overall device power, on the order of 1 mA. It is advised that any application that does not require this doubled clock should have it turned off.

For additional information, refer to the OSC_CR2 register on page 256.

20. Multiply Accumulate (MAC)

This chapter presents the Multiply Accumulate (MAC) and its associated registers. The MAC block is a fast 8-bit multiplier or a fast 8-bit multiplier with 32-bit accumulate. Refer to Table 20-1 for MAC availability by part number. For a complete table of the MAC registers, refer to the "Summary Table of the System Resource Registers" on page 374. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

20.1 Architectural Description

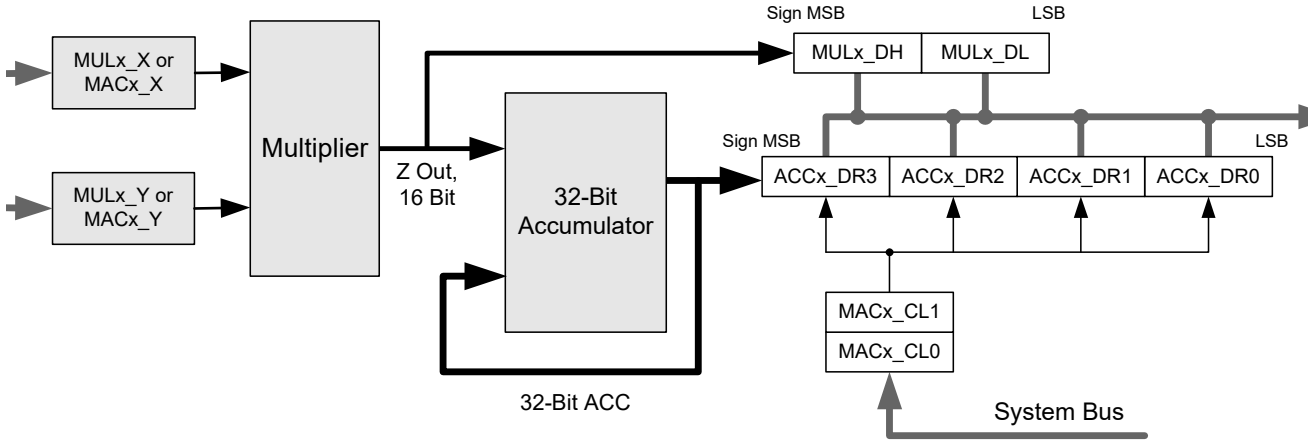
The MAC is a register-based system resource. Its only interface is the system bus; therefore, there are no special clocks or enables that are required to be sourced from digital or analog PSoC blocks. In devices with more than one MAC block, each MAC is completely independent of the other. Refer to Table 20-1 for MAC availability by part number. Note that the CY8C22x13, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 do not have MAC functionality.

The architectural presentation of the MAC is illustrated in Figure 20-1.

Table 20-1. MAC Availability for PSoC Devices

PSoC Part Number	Number of MAC Blocks
CY8C29x66	2
CY8C27x43	1
CY8C24x94	2
CY8C24x23	1
CY8C24x23A	1
CY8C22x45	1
CY8C22x13	0
CY8C21345	1
CY8C21x34	0
CY8C21x23	0
CY7C64215	2
CY7C603xx	0
CYWUSB6953	0

Figure 20-1. MAC Block Diagram



20.2 Application Description

20.2.1 Multiplication with No Accumulation

For simple multiplication, the MAC block accepts two 8-bit signed numbers as the multiplicands for a multiply operation. The product of the multiplication is stored in a 16-bit signed form. Up to four registers are involved with simple multiplication: MULx_X, MULx_Y, MULx_DH, and MULx_DL.

To execute a multiply, simply write a value to either the MULx_X or MULx_Y registers. Immediately after the write of the multiplicand, the product is available at registers MULx_DH and MULx_DL. After reset of the part at power up or after an external reset, the MAC registers will not be reset to zero. Therefore, after the write of the first multiplicand, the product is indeterminate. After the write of the second multiplicand, the product registers are updated with the product of the first and second multiplicands (assuming one of the writes was to MULx_X and the other was to MULx_Y). Multiplication is associative so the order in which you write to X and Y does not matter.

20.2.2 Accumulation After Multiplication

Accumulation of products is a feature that is implemented on top of simple multiplication. When using the MAC to accumulate the products of successive multiplications, two 8-bit signed values are used for input. The product of the multiplication is accumulated as a 32-bit signed value.

The user has the choice to either cause a multiply/accumulate function to take place or a multiply only function. The user selects which operation is performed by choosing of input register. The multiply function occurs immediately whenever the MULx_X or the MULx_Y multiplier input registers are written, and the result is available in the MULx_DH and MULx_DL multiplier result registers, as discussed in the 20.2.1 Multiplication with No Accumulation section. The multiply/accumulate function is executed whenever there is a write to the MACx_X or the MACx_Y multiply/accumulate input registers; the result is available in the ACCx_DR3, ACCx_DR2, ACCx_DR1, and ACCx_DR0 accumulator result registers. A write to the MULx_X or MACx_X registers is input as the X value to both the multiply and multiply/accumulate functions. A write to the MULx_Y or MACx_Y registers is input as the Y value to both the multiply and multiply/accumulate functions. A write to the MACx_CL0 or MACx_CL1 registers will clear the value in the four accumulate registers.

To clear the accumulated products, simply write to either of the MACx_CLx registers.

20.3 Register Definitions

In PSoC devices with more than one MAC block, there will be one of the following registers for each block. The registers in this section are listed in address order. Refer to the table titled “MAC Availability for PSoC Devices” on page 387 to determine how many MAC blocks are available for your PSoC device. Note that the CY8C22x13, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 do not have MAC functionality.

The following registers are associated with the MAC PSoC Blocks. Each register description has an associated register table showing the bit structure for that register. The ‘X’ in the Access column of some register tables signify that the value after power on reset is unknown. For a complete table of the MAC registers, refer to the “Summary Table of the System Resource Registers” on page 374.

20.3.1 MULx_X Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E8h	MUL0_X	Data[7:0]								W : XX

LEGEND
X The value after power on reset is unknown.

The Multiply Input X Register (MULx_X) is one of two multiplicand registers for the signed 8-bit multiplier in the PSoC MAC.

Bits 7 to 0: Data[7:0]. The multiply X (MULx_X) register is one of two multiplicand registers for the signed 8-bit multiplier in the PSoC MAC. When these write only registers are written, the product of the written value and the current value of the MULx_X registers are calculated.

For additional information, refer to the MULx_X register on page 180.

20.3.2 MULx_Y Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,E9h	MUL0_Y	Data[7:0]								W : XX

LEGEND
X The value after power on reset is unknown.

The Multiply Input Y Register (MULx_Y) is one of two multiplicand registers for the signed 8-bit multiplier in the PSoC MAC.

Bits 7 to 0: Data[7:0]. The multiply Y (MULx_Y) register is one of two multiplicand registers for the signed 8-bit multiplier in the PSoC MAC. When these write only registers are written, the product of the written value and the current value of the MULx_Y registers are calculated.

For additional information, refer to the MULx_Y register on page 181.

20.3.3 MULx_DH Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,EAh	MUL0_DH	Data[7:0]								R : XX

LEGEND
X The value after power on reset is unknown.

The Multiply Result High Byte Register (MULx_DH) holds the most significant byte of the 16-bit product.

Bits 7 to 0: Data[7:0]. The product of the multiply operation on the MULx_X and MULx_Y registers is stored as a signed 16-bit value. The read only multiply data high (MUL0_DH and MUL1_DH) registers hold the most significant byte of the 16-bit product.

For additional information, refer to the MULx_DH register on page 182.

20.3.4 MULx_DL Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,EBh	MUL0_DL	Data[7:0]								R : XX

LEGEND
X The value after power on reset is unknown.

The Multiply Result Low Byte Register (MULx_DL) holds the least significant byte of the 16-bit product.

Bits 7 to 0: Data[7:0]. The product of the multiply operation on the MULx_X and MULx_Y registers is stored as a signed 16-bit value. The read only multiply data low (MUL0_DL and MUL1_DL)) registers hold the least significant byte of the 16-bit product.

For additional information, refer to the MULx_DL register on page 183.

20.3.5 MACx_X/ACCx_DR1 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,ECh	MAC0_X/ ACC0_DR1	Data[7:0]								RW : 00

The Accumulator Data Register 1 (MACx_X/ACCx_DR1) is the multiply accumulate X register and the second byte of the accumulated value.

Bits 7 to 0: Data[7:0]. This register performs two distinct functions; therefore, two names are used to refer to the same address. When the address is written, a multiply operation with accumulation is performed. The multiply accumulate X (MACx_X) register is one of the two multiplicand registers for the signed 8-bit multiply with accumulate operation. When this register is written, the product of the written value and the current value of the MACx_Y register is calculated, then that product is added to the 32-bit accumulators value. When this address is read, the accumulator's data register 1 is read. This register holds the second of four bytes used to hold the accumulator's value. This byte is the most significant of the lower 16 bits of the accumulator's value.

For additional information, refer to the MACx_X/ACCx_DR1 register on page 184.

20.3.6 MACx_Y/ACCx_DR0 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,EDh	MAC0_Y/ ACC0_DR0	Data[7:0]								RW : 00

The Accumulator Data Register 0 (MACx_Y/ACCx_DR0) is the multiply accumulate Y register and the first byte of the accumulated value.

Bits 7 to 0: Data[7:0]. This register performs two distinct functions; therefore, two names are used to refer to the same address. When the address is written, a multiply operation with accumulation is performed. The multiply accumulate Y (MACx_Y) register is one of the two multiplicand registers for the signed 8-bit multiply with accumulate operation. When this register is written, the product of the written value and the current value of the MACx_X register is calculated, then that product is added to the 32-bit accumulators value. When this address is read, the accumulator's data register 0 is read. This register holds the least significant of four bytes used to hold the accumulator's value.

For additional information, refer to the MACx_Y/ACCx_DR0 register on page 185.

20.3.7 MACx_CL0/ACCx_DR3 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,EEh	MAC0_CL0/ ACC0_DR3	Data[7:0]								RW : 00

The Accumulator Data Register 3 (MACx_CL0/ACCx_DR3) is an accumulator clear register and the fourth byte of the accumulated value.

Bits 7 to 0: Data[7:0]. This register performs two distinct functions; therefore, two names are used to refer to the same address. When the address is written with any value, all 32-bits of the accumulator are reset to zero. When this address is read, the accumulator's data register 3 is read. This register holds the most significant of four bytes used to hold the accumulator's value.

For additional information, refer to the MACx_CL0/ACCx_DR3 register on page 186.

20.3.8 MACx_CL1/ACCx_DR2 Register

Add.	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,EFh	MAC0_CL1/ ACC0_DR2	Data[7:0]								RW : 00

The Accumulator Data Register 2 (MACx_CL1/ACCx_DR2) is an accumulator clear register and the third byte of the accumulated value.

Bits 7 to 0: Data[7:0]. This register performs two distinct functions; therefore, two names are used to refer to the same address. When the address is written with any value, all 32 bits of the accumulator are reset to zero. When this address is read, the accumulator's data register 2 is read. This register holds the third of four bytes used to hold the accumulator's value. This byte is the least significant of the upper 16 bits of the accumulator's value.

For additional information, refer to the MACx_CL1/ACCx_DR2 register on page 187.

21. I²C

This chapter explains the I²C™ block and its associated registers. The I2C communications block is a serial processor designed to implement a complete I2C slave or master. For a complete table of the I2C registers, refer to the "Summary Table of the System Resource Registers" on page 374. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

21.1 Architectural Description

The I2C communications block is a serial to parallel processor, designed to interface the PSoC device to a two-wire I2C serial communications bus. To eliminate the need for excessive M8C microcontroller intervention and overhead, the block provides I2C specific support for status detection and generation of framing bits.

The I2C block directly controls the data (SDA) and clock (SCL) signals to the external I2C interface, through connections to two dedicated GPIO pins. The PSoC device firmware interacts with the block through I/O (input/output) register reads and writes, and firmware synchronization will be implemented through polling and/or interrupts.

PSoC I2C features include:

- Master/Slave, Transmitter/Receiver operation
- Byte processing for low CPU overhead
- Interrupt or polling CPU interface
- Master clock rates: 50K, 100K, 400K
- Multi-master clock synchronization
- Multi-master mode arbitration support
- 7- or 10-bit addressing (through firmware support)
- Optional hardware support for automatic address comparison (7-bit address and slave mode only)
- SMBus operation (through firmware support)

Hardware functionality provides basic I2C control, data, and status primitives. A combination of hardware support and firmware command sequencing provides a high degree of flexibility for implementing the required I2C functionality.

Because receive and transmitted data are not buffered, there is no support for automatic receive acknowledge. The M8C microcontroller must intervene at the boundary of each byte and either send a byte or ACK received bytes. This is a hardware limitation of the I2C.

The I2C block is designed to support a set of primitive operations and detect a set of status conditions specific to the I2C protocol. These primitive operations and conditions are manipulated and combined at the firmware level to support the required data transfer modes. The CPU will set up control options and issue commands to the unit through I/O writes and obtain status through I/O reads and interrupts.

The block operates as either a slave, a master, or both. When enabled in Slave mode, the unit is always listening for a Start condition, or sending or receiving data. Master mode can work in conjunction with Slave mode. The master supplies the ability to generate the START or STOP condition and determine if other masters are on the bus. For Mult-Master mode, clock synchronization is supported. If Master mode is enabled and Slave mode is not enabled, the block does not generate interrupts on externally generated Start conditions.

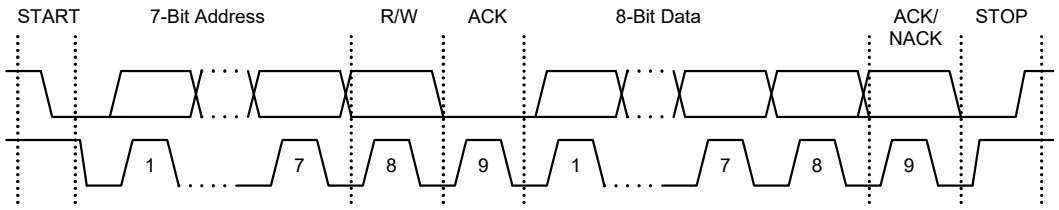
21.1.1 Basic I²C Data Transfer

Figure 21-1 shows the basic form of data transfers on the I2C bus with a 7-bit address format. (For a more detailed description, see the Phillips Semiconductors' I²C™ Specification, version 2.1.)

A Start condition (generated by the master) is followed by a data byte, consisting of a 7-bit slave address (there is also a 10-bit address mode) and a Read/Write (RW) bit. The RW bit sets the direction of data transfer. The addressed slave is required to acknowledge (ACK) the bus by pulling the data line low during the ninth bit time. If the ACK is received, the transfer may

proceed and the master can transmit or receive an indeterminate number of bytes, depending on the RW direction. If the slave does not respond with an ACK for any reason, a Stop condition is generated by the master to terminate the transfer or a Restart condition may be generated for a retry attempt.

Figure 21-1. Basic I²C Data Transfer with 7-Bit Address Format



21.2 Application Description

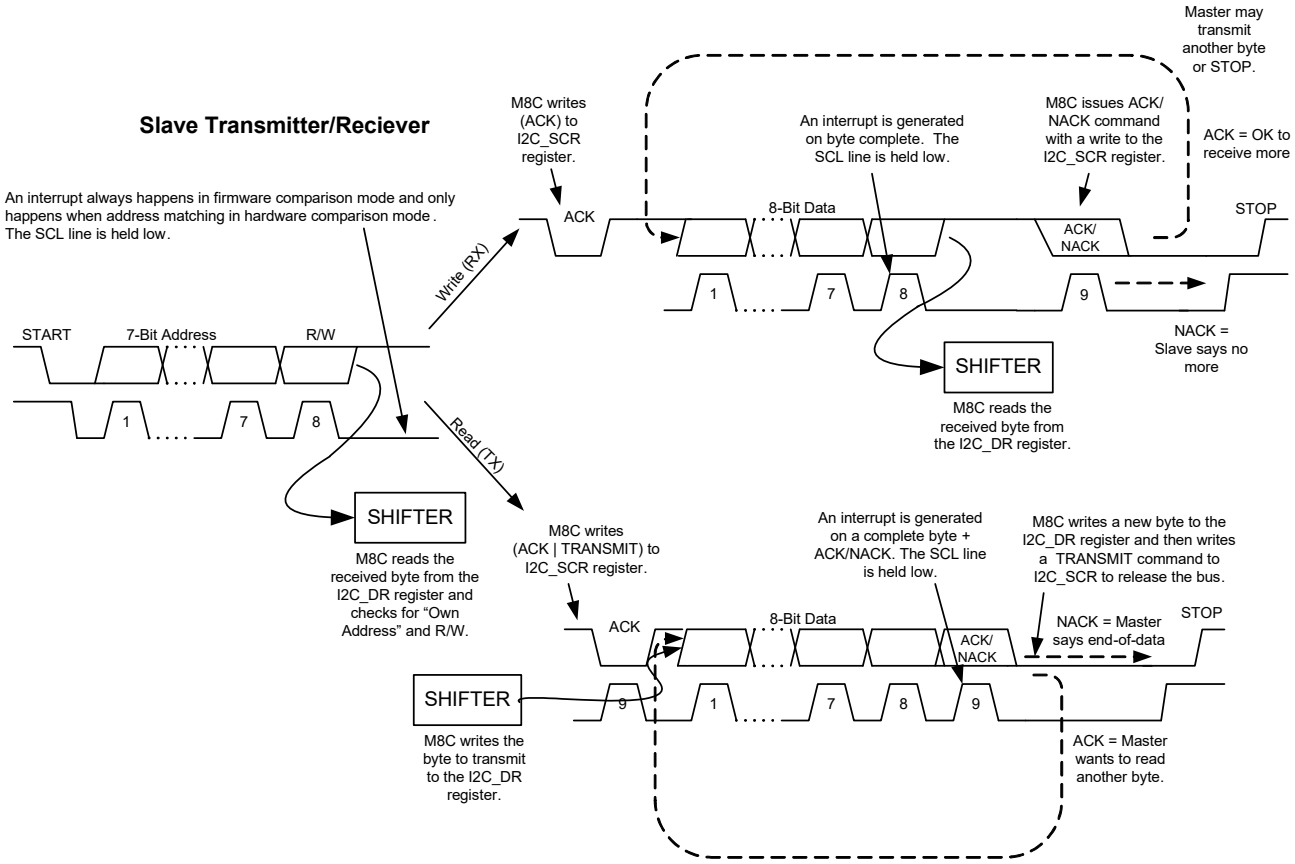
21.2.1 Slave Operation

Assuming Slave mode is enabled, it is continually listening to or on the bus for a Start condition. When detected, the transmitted Address/RW byte is received and compared by either firmware or hardware:

1. In firmware address comparison mode, at the point where eight bits of the address/RW byte are received, a byte complete interrupt is generated. Following the low of the clock, the bus is stalled by holding the SCL line low, until the PSoC device has a chance to read the address byte and compare it to its own address. It will issue an ACK or NACK command based on that comparison.
2. In hardware address comparison mode, the received address is automatically compared when the address/RW byte is completed, and the hardware will automatically issue an NACK if address mismatches. If address matches, a byte complete interrupt is generated, which is similar in the firmware comparison. Following the low of the clock, the bus is stalled by holding the SCL line low, until the PSoC device has a chance to read the address byte to determine the R/W I2C operation (firmware does not need to compare the address). Finally, the firmware needs to issue an ACK to continue I2C operation.

If there is an address match, the RW bit determines how the PSoC device will sequence the data transfer in Slave mode, as shown in the two branches of Figure 21-2. I2C handshaking methodology (slave holds the SCL line low to “stall” the bus) will be used as necessary, to give the PSoC device time to respond to the events and conditions on the bus. Figure 21-2 is a graphical representation of a typical data transfer from the slave perspective.

Figure 21-2. Slave Operation



21.2.2 Master Operation

To prepare for a Master mode transaction, the PSoC device must determine if the bus is free. This is done by polling the BusBusy status. If busy, interrupts can be enabled to detect a Stop condition. Once it is determined that the bus is available, firmware should write the address byte into the I2C_DR register and set the Start Gen bit in the I2C_MSCR register.

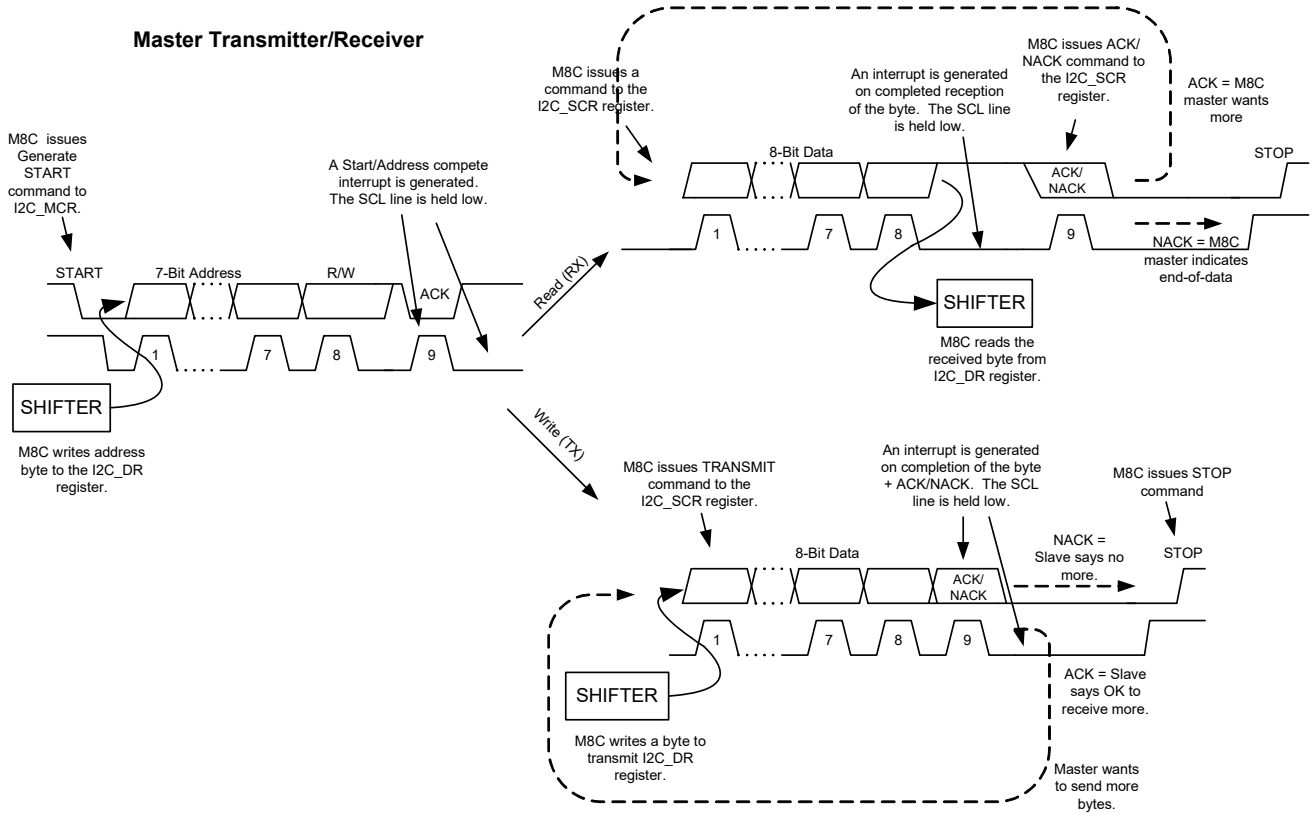
If the slave sub-unit is not enabled, the block is in Master Only mode. In this mode, the unit does not generate interrupts or stall the I2C bus on externally generated Start conditions.

In a multi-master environment there are two additional outcomes possible:

1. The PSoC device was too late to reserve the bus as a master, and another master may have generated a Start and sent an Address/RW byte. In this case, the unit as a master will fail to generate a Start and is forced into Slave mode. The Start will be pending and eventually occur at a later time when the bus becomes free. When the interrupt occurs in Slave mode, the PSoC device can determine that the Start command was unsuccessful by reading the I2C_MSCR register Start bit, which is reset on successful Start from this unit as master. If this bit is still a '1' on the Start/Address interrupt, it means that the unit is operating in Slave mode. In this case, the data register has the master's address data.
2. If another master starts a transmission at the same time as this unit, arbitration occurs. If this unit loses the arbitration, the LostArb status bit is set. In this case, the block releases the bus and switches to Slave operation. When the Start/Address interrupt occurs, the data register has the winning master's address data.

Figure 21-3 is a graphical representation of a typical data transfer from the master perspective.

Figure 21-3. Master Operation



21.3 Register Definitions

The following registers are associated with I2C and are listed in address order. Each register description has an associated register table showing the bit structure for that register. The bits in the tables that are grayed out are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of '0'. For a complete table of I2C registers, refer to the "Summary Table of the System Resource Registers" on page 374.

21.3.1 I2C_ADDR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,ADh	I2C_ADDR	HwAddrEn	Addr[6:0]							RW : 00

The I2C address register is used to configure the hardware address automatic comparison feature so that the microcontroller will not be disturbed by an unwanted slave request. When HwAddrEn is enabled, the 7-bit address should be stored in Addr[6:0]; there is no interrupt only when the received address matches the stored address.

The hardware address automatic compare feature is available in slave only mode; master/slave mode is not supported.

Table 21-1. I2C_ADDR Configuration Register

Bit	Access	Description	Mode
7	RW	HwAddrEn: 1: Enable hardware address comparison feature. Only supports 7-bit address. When you enable the hardware address comparison feature, I2C block will not support the special system address definition which is listed in I2C V2.1 spec, section 10 (for example: general call address, CBUS address, 10-bit slave address, and so on). 0: Disable hardware address comparison feature.	Slave Only
6:0	RW	Slave Address[6:0]: These 7 bits hold the slave's own device address.	Slave Only

For additional information, refer to the I2C_ADDR register on page 244.

21.3.2 I2C_CFG Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,D6h	I2C_CFG		PSelect	Bus Error IE	Stop IE	Clock Rate[1:0]		Enable Master	Enable Slave	RW : 00

The I2C Configuration Register (I2C_CFG) is used to set the basic operating modes, baud rate, and selection of interrupts.

The bits in this register control baud rate selection and optional interrupts. The values are typically set once for a given configuration. The bits in this register are all RW.

Table 21-2. I2C_CFG Configuration Register

Bit	Access	Description	Mode
6	RW	I2C Pin Select 0 = P1[7], P1[5] 1 = P1[1], P1[0]	Master/ Slave
5	RW	Bus Error IE Bus error interrupt enable. 0 = Disabled. 1 = Enabled. An interrupt is generated on the detection of a Bus Error.	Master Only
4	RW	Stop IE Stop interrupt enable. 0 = Disabled. 1 = Enabled. An interrupt is generated on the detection of a Stop Condition.	Master/ Slave
3:2	RW	Clock Rate 00 = 100K Standard Mode 01 = 400K Fast Mode 10 = 50K Standard Mode 11 = Reserved	Master/ Slave
1	RW	Enable Master 0 = Disabled 1 = Enabled	Master/ Slave
0	RW	Enable Slave 0 = Disabled 1 = Enabled	Master/ Slave

Bit 6: PSelect. With the default value of zero, the I2C pins are P1[7] for clock and P1[5] for data. When this bit is set, the pins for I2C switch to P1[1] for clock and P1[0] for data. This bit may not be changed while either the Enable Master or Enable Slave bits are set. However, the PSelect bit may be set at the same time as the enable bits. The two sets of pins that may be used on I2C are not equivalent. The default set, P1[7] and P1[5], are the preferred set. The alternate set, P1[1] and P1[0], are provided so that I2C may be used with 8-pin PSoC parts.

If In-System Serial Programming (ISSP) is to be used and the alternate I2C pin set is also used, it is necessary to take into account the interaction between the PSoC Test Controller and the I2C bus. The interface requirements for ISSP should be reviewed to ensure that they are not violated.

Even if ISSP is not used, pins P1[1] and P1[0] will respond differently to a POR or XRES event than other I/O pins. After an XRES event, both pins are pulled down to ground by going into the resistive zero Drive mode, before reaching the High Z Drive mode. After a POR event, P1[0] will drive out a one, then go to the resistive zero state for some time, and finally reach the High Z drive mode state. After POR, P1[1] will go into a resistive zero state for a while, before going to the High Z Drive mode.

Another issue with selecting the alternate I2C pins set is that these pins are also the crystal pins. Therefore, a crystal may not be used when the alternate I2C pin set is selected.

Bit 5: Bus Error IE (Interrupt Enable). This bit controls whether the detection of a bus error will generate an interrupt. A bus error is typically a misplaced Start or Stop.

This is an important interrupt with regards to Master operation. When there is a misplaced Start or Stop on the I2C bus, all slave devices (including this device, if Slave mode is enabled) will reset the bus interface and synchronize to this signal. However, when the hardware detects a bus error in Master Mode operation, the device will release the bus and transition to an idle state. In this case, a Master operation in progress will never have any further status or interrupts associated with it. Therefore, the master may not be able to determine the status of that transaction. An immediate bus error interrupt will inform the master that this transfer did not succeed.

Bit 4: Stop IE (Interrupt Enable). When this bit is set, a master or slave can interrupt on Stop detection. The status bit associated with this interrupt is the Stop Status bit in the Slave Status and Control register. When the Stop Status bit transitions from ‘0’ to ‘1’, the interrupt is generated. It is important to note that the Stop Status bit is not automatically cleared. Therefore, if it is already set, no new interrupts are generated until it is cleared by firmware.

Bits 3 and 2: Clock Rate[1:0]. These bits offer a selection of three sampling and bit rates. All block clocking is based on the SYSCLK input, which is nominally 24 MHz (unless the PSoC device is in external clocking mode). The sampling rate and the baud rate are determined as follows:

- Sample Rate = SYSCLK/Pre-scale Factor
- Baud Rate = 1/(Sample Rate x Samples per Bit)

The nominal values, when using the internal 24 MHz oscillator, are shown in Table 21-3.

Table 21-3. I²C Clock Rates

Clock Rate [1:0]	I2C Mode	SYSCLK Pre-scale Factor	Samples per Bit	Internal Sampling Freq./Period (24 MHz)	Master Baud Rate (nominal)	Start/Stop Hold Time (8 clocks)
00b	Standard	/16	16	1.5 MHz/667 ns	93.75 kHz	5.3 µs
01b	Fast	/4	16	6 MHz/167 ns	375 kHz	1.33 µs
10b	Standard	/16	32	1.5 MHz/667 ns	46.8 kHz	10.7 µs
11b	Reserved					

When clocking the input with a frequency other than 24 MHz (for example, clocking the PSOC device with an external clock), the baud rates and sampling rates will scale accordingly. Whether the block will work in a Standard Mode or Fast Mode system depends on the sample rate. The sample rate must be sufficient to resolve bus events, such as Start and Stop conditions. (See the Phillips Semiconductors' I²C™ Specification, version 2.1, for minimum Start and Stop hold times.)

Bit 1: Enable Master. When this bit is set, the Master Status and Control register is enabled (otherwise it is held in reset) and I2C transfers can be initiated in Master mode. When the master is enabled and operating, the block will clock the I2C bus at one of three baud rates, defined in the Clock Rate register. When operating in Master mode, the hardware is multi-master capable, implementing both clock synchronization and arbitration. If the Slave Enable bit is not set, the block will operate in Master Only mode. All external Start conditions are ignored (although the Bus Busy status bit will still keep track of bus activity). Block enable will be synchronized to the SYSCLK clock input (see "Timing Diagrams" on page 404).

Bit 0: Enable Slave. When the slave is enabled, the block generates an interrupt on any Start condition and an address byte that it receives, indicating the beginning of an I2C transfer. When operating as a slave, the block is clocked from an external master. Therefore, the block will work at any frequency up to the maximum defined by the currently selected clock rate. The internal clock is only used in Slave mode, to ensure that there is adequate setup time from data output to the next clock on the release of a slave stall. When the Enable Slave and Enable Master bits are both '0', the block is held in reset and all status is cleared. See Figure 21-4 for a description of the interaction between the Master/Slave Enable bits. Block enable will be synchronized to the SYSCLK clock input (see "Timing Diagrams" on page 404).

Table 21-4. Enable Master/Slave Block Operation

Enable Master	Enable Slave	Block Operation
No	No	Disabled: The block is disconnected from the GPIO pins, P1[5] and P1[7]. (The pins may be used as general purpose I/O.) When either the master or slave is enabled, the GPIO pins are under control of the I2C hardware and are unavailable. All internal registers (except I2C_CFG) are held in reset.
No	Yes	Slave Only Mode: Any external Start condition will cause the block to start receiving an address byte. Regardless of the current state, any Start resets the interface and initiates a Receive operation. Any Stop will cause the block to revert to an idle state The I2C_MSCR register is held in reset.
Yes	No	Master Only Mode: External Start conditions are ignored in this mode. No Byte Complete interrupts on external traffic are generated, but the Bus Busy status bit continues to capture Start and Stop status, and thus may be polled by the master to determine if the bus is available. Full multi-master capability is enabled, including clock synchronization and arbitration. The block will generate a clock based on the setting in the Clock Rate register
Yes	Yes	Master/Slave Mode: Both master and slave may be operational in this mode. The block may be addressed as a slave, but firmware may also initiate Master mode transfers. In this configuration, when a master loses arbitration during an address byte, the hardware will revert to Slave mode and the received byte will generate a slave address interrupt.

For additional information, refer to the I2C_CFG register on page 161.

21.3.3 I2C_SCR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,D7h	I2C_SCR	Bus Error	Lost Arb	Stop Status	ACK	Address	Transmit	LRB	Byte Complete	# : 00

LEGEND

Access is bit specific. Refer to Table 21-5 for detailed bit descriptions.

The I2C Status and Control Register (I2C_SCR) is used by both master and slave to control the flow of data bytes and to keep track of the bus state during a transfer.

This register contains status bits, for determining the state of the current I2C transfer, and control bits, for determining the actions for the next byte transfer. At the end of each byte transfer, the I2C hardware interrupts the M8C microcontroller and stalls the I2C bus on the subsequent low of the clock, until the PSoC device intervenes with the next command. This register may be read as many times as necessary; but on a subsequent write to this register, the bus stall is released and the current transfer will continue.

There are six status bits: Byte Complete, LRB, Address, Stop Status, Lost Arb, and Bus Error. These bits have Read/Clear (R/C) access, which means that they are set by hardware but may be cleared by a write of '0' to the bit position. Under certain conditions, status is cleared automatically by the hardware. These cases are noted in Table 21-5.

There are two control bits: Transmit and ACK. These bits have RW access and may be cleared by hardware.

Bit 7: Bus Error. The Bus Error status detects misplaced Start or Stop conditions on the bus. These may be due to noise, rogue devices, or other devices that are not yet synchronized with the I2C bus traffic. According to the I2C specification, all compatible devices must reset their interface on a received Start or Stop. This is a natural thing to do in Slave mode, because a Start will initiate an address reception and a Stop will idle the slave. In the case of a master, this event will force the master to release the bus and idle. However, because a master does not respond to external Start or Stop conditions, an immediate interrupt on this event allows the master to continue to keep track of the bus state.

A bus error is defined as follows. A Start is only valid if the block is idle (master or slave) or a Slave receiver is ready to receive the first bit of a new byte after an ACK. Any other timing for a Start condition causes the Bus Error bit to be set. A Stop is only valid if the block is idle or a Slave receiver is ready to receive the first bit of a new byte after an ACK. Any other timing for a Stop condition causes the Bus Error bit to be set.

Table 21-5. I2C_SCR Status and Control Register

Bit	Access	Description	Mode
7	RC	Bus Error 1 = A misplaced Start or Stop condition was detected. This status bit must be cleared by firmware with a write of '0' to the bit position. It is never cleared by the hardware.	Master Only
6	RC	Lost Arb 1 = Lost Arbitration. This bit is set immediately on lost arbitration; however, it does not cause an interrupt. This status may be checked after the following Byte Complete interrupt. Any Start detect will automatically clear this bit.	Master Only
5	RC	Stop Status 1 = A Stop condition was detected. This status bit must be cleared by firmware with a write of '0' to the bit position. It is never cleared by the hardware.	Master/ Slave
4	RW	ACK: Acknowledge Out 0 = NACK the last received byte. 1 = ACK the last received byte. This bit is automatically cleared by hardware on the following Byte Complete event.	Master/ Slave
3	RC	Address 1 = The transmitted or received byte is an address. This status bit must be cleared by firmware with a write of '0' to the bit position.	Master/ Slave
2	RW	Transmit 0 = Receive Mode. 1 = Transmit Mode. This bit is set by firmware to define the direction of the byte transfer. Any Start detect will automatically clear this bit.	Master/ Slave
1	RC	LRB: Last Received Bit The value of the ninth bit in a Transmit sequence, which is the acknowledge bit from the receiver. 0 = Last transmitted byte was ACK'ed by the receiver. 1 = Last transmitted byte was NACK'ed by the receiver. Any Start detect will automatically clear this bit.	Master/ Slave
0	RC	Byte Complete Transmit Mode: 1 = 8 bits of data have been transmitted and an ACK or NACK has been received. Receive Mode: 1 = 8 bits of data have been received. Any Start detect will automatically clear this bit.	Master/ Slave

Bit 6: Lost Arb. This bit is set when I2C bus contention is detected, during a Master mode transfer. Contention will occur when a master is writing a ‘1’ to the SDA output line and reading back a ‘0’ on the SDA input line at the given sampling point. When this occurs, the block immediately releases the SDA, but continues clocking to the end of the current byte. On the resulting byte interrupt, firmware can determine that arbitration was lost to another master by reading this bit.

The sequence occurs differently between Master transmitter and Master receiver. As a transmitter, the contention will occur on a data bit. On the subsequent Byte Complete interrupt, the Lost Arbitration status is set. In Receiver mode, the contention will occur on the ACK bit. The master that NACK’ed the last reception will lose the arbitration. However, the hardware will shift in the next byte in response to the winning master’s ACK, so that a subsequent Byte Complete interrupt occurs. At this point, the losing master can read the Lost Arbitration status. Contention is checked only at the eight data bit sampling points and one ACK bit sampling point.

Bit 5: Stop Status. Stop status is set on detection of an I2C Stop condition. This bit is sticky, which means that it will remain set until a ‘0’ is written back to it by the firmware. This bit may only be cleared if the Byte Complete status is set. If the Stop Interrupt Enable bit is set, an interrupt is also generated on Stop detection. It is never automatically cleared.

Using this bit, a slave can distinguish between a previous Stop or Restart on a given address byte interrupt. In Master mode, this bit may be used in conjunction with the Stop IE bit, to generate an interrupt when the bus is free. However, in this case, the bit must have previously been cleared prior to the reception of the Stop in order to cause an interrupt.

Bit 4: ACK. This control bit defines the acknowledge data bit that is transmitted out in response to a received byte. When receiving, a Byte Complete interrupt is generated after the eighth data bit is received. On the subsequent write to this register to continue (or terminate) the transfer, the state of this bit will determine the next bit of data that is transmitted. It is ***active high***. A ‘1’ will send an ACK and a ‘0’ will send a NACK.

A Master receiver normally terminates a transfer, by writing a ‘0’ (NACK) to this bit. This releases the bus and automatically generates a Stop condition. A Slave receiver may also send a NACK, to inform the master that it cannot receive any more bytes.

Bit 3: Address. This bit is set when an address has been received. This consists of a Start or Restart, and an address byte. This bit applies to both master and slave.

In Slave mode, when this status is set, firmware will read the received address from the data register and compare it with its own address. If the address does not match, the firmware will write a NACK indication to this register. No further interrupts will occur, until the next address is received. If the address does match, firmware must ACK the received byte, then Byte Complete interrupts are generated on subsequent bytes of the transfer.

This bit will also be set when address transmission is complete in Master mode. If a lost arbitration occurs during the transmission of a master address (indicated by the Lost Arb bit), the block will revert to Slave mode if enabled. This bit then signifies that the block is being addressed as a slave.

If Slave mode is not enabled, the Byte Complete interrupt will still occur to inform the master of lost arbitration.

Bit 2: Transmit. This bit sets the direction of the shifter for a subsequent byte transfer. The shifter is always shifting in data from the I2C bus, but a write of ‘1’ enables the output of the shifter to drive the SDA output line. Because a write to this register initiates the next transfer, data must be written to the data register prior to writing this bit. In Receive mode, the previously received data must have been read from the data register before this write. In Slave mode, firmware derives this direction from the RW bit in the received slave address. In Master mode, the firmware decides on the direction and sets it accordingly.

This direction control is only valid for data transfers. The direction of address bytes is determined by the hardware, depending on the Master or Slave mode.

The Master transmitter terminates a transfer by writing a zero to the transmit bit. This releases the bus and automatically sends a Stop condition, or a Stop/Start or Restart, depending on the I2C_MSCR control bits.

Bit 1: LRB (Last Received Bit). This is the last received bit in response to a previously transmitted byte. In Transmit mode, the hardware will send a byte from the data register and clock in an acknowledge bit from the receiver. On the subsequent byte complete interrupt, firmware will check the value of this bit. A ‘0’ is the ACK value and a ‘1’ is a NACK value. The meaning of the LRB depends on the current operating mode.

Master Transmitter:

'0': ACK. The slave has accepted the previous byte. The master may send another byte by first writing the byte to the I2C_DR register and then setting the Transmit bit in the I2C_SCR register. Optionally, the master may clear the Transmit bit in the I2C_SCR register. This will automatically send a Stop. If the Start or Restart bits are set in the I2C_MSCR register, the Stop may be followed by a Start or Restart.

'1': NACK. The slave cannot accept any more bytes. A Stop is automatically generated by the hardware on the subsequent write to the I2C_SCR register (regardless of the value written). However, a Stop/Start or Restart condition may also be generated, depending on whether firmware has set the Start or Restart bits in the I2C_MSCR register.

Slave Transmitter:

'0': ACK. The master wants to read another byte. The slave should load the next byte into the I2C_DR register and set the transmit bit in the I2C_SCR register to continue the transfer.

'1': NACK. The master is done reading bytes. The slave will revert to IDLE state on the subsequent I2C_SCR write (regardless of the value written).

Bit 0: Byte Complete. The I2C hardware operates on a byte basis. In Transmit mode, this bit is set and an interrupt is generated at the end of nine bits (the transmitted byte + the received ACK). In Receive mode, the bit is set after the eight bits of data are received. When this bit is set, an interrupt is generated at these data sampling points, which are associated with the SCL input clock rising (see details in the Timing section). If the PSoC device responds with a write back to this register before the subsequent falling edge of SCL (which is approximately one-half bit time), the transfer will continue without interruption. However, if the PSoC device is unable to respond within that time, the hardware will hold the SCL line low, stalling the I2C bus. In both Master and Slave mode, a subsequent write to the I2C_SCR register will release the stall.

For additional information, refer to the I2C_SCR register on page 162.

21.3.4 I2C_DR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,D8h	I2C_DR	Data[7:0]								RW : 00

The I2C Data Register (I2C_DR) provides read/write access to the Shift register.

Bits 7 to 0: Data[7:0]. This register is not buffered; and therefore, writes and valid data reads may only occur at specific points in the transfer. These cases are outlined as follows.

- **Master or Slave Receiver** – Data in the I2C_DR register is only valid for reading, when the Byte Complete status bit is set. Data bytes must be read from the register before writing to the I2C_SCR register, which continues the transfer.
- **Master Start or Restart** – Address bytes must be written in I2C_DR before the Start or Restart bit is set in the I2C_MSCR register, which causes the Start or Restart to generate and the address to shift out.
- **Master or Slave Transmitter** – Data bytes must be written to the I2C_DR register before the transmit bit is set in the I2C_SCR register, which causes the transfer to continue.

For additional information, refer to the I2C_DR register on page 164.

21.3.5 I2C_MSCR Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,D9h	I2C_MSCR					Bus Busy	Master Mode	Restart Gen	Start Gen	R : 00

The I2C Master Status and Control Register (I2C_MSCR) implements I2C framing controls and provides Bus Busy status.

Bit 3: Bus Busy. This read only bit is set to '1' by any Start condition and reset to '0' by a Stop condition. It may be polled by firmware to determine when a bus transfer may be initiated.

Bit 2: Master Mode. This bit indicates that the device is operating as a master. It is set in the detection of this block's Start condition and reset in the detection of the subsequent Stop condition.

Bit 1: Restart Gen. This bit is only used at the end of a master transfer (as noted in Other Cases 1 and 2 of the Start Gen bit). If an address is loaded into the data register and this bit is set prior to NACKing (Master receiver) or resetting the transmit bit (Master transmitter), or after a Master transmitter is NACK'ed by the slave, a Restart condition is generated followed by the transmission of the address byte.

Bit 0: Start Gen. Before setting this bit, firmware must write the address byte to send into the I2C_DR register. When this bit is set, the Start condition is generated followed immediately by the transmission of the address byte. (No control in the I2C_SCR register is needed for the master to initiate a transmission; the direction is inherently "transmit.") The bit is automatically reset to '0' after the Start has been generated.

There are three possible outcomes as a result of setting the Start Gen bit.

1. The bus is free and the Start condition is generated successfully. A Byte Complete interrupt is generated after the Start and the address byte are transmitted. If the address was ACK'ed by the receiver, the firmware may then proceed to send data bytes.
2. The Start command is too late. Another master in a multi-master environment has generated a valid Start and the bus is busy. The resulting behavior depends upon whether Slave mode is enabled.

   Slave mode is enabled: A Start and address byte interrupt is generated. When reading the I2C_MSCR register, the master will see that the Start Gen bit is still set and that the I2C_SCR register has the Address bit set, indicating that the block is addressed as a slave.

   Slave mode is not enabled: The Start Gen bit will remain set and the Start is queued, until the bus becomes free and the Start condition is subsequently generated. An interrupt is generated at a later time, when the Start and address byte has been transmitted.
3. The Start is generated, but the master looses arbitration to another master in a multi-master environment. The resulting behavior depends upon whether Slave mode is enabled.

   Slave mode is enabled: A Start and address byte interrupt is generated. When reading the I2C_MSCR, the master will see that the Start Gen bit cleared, indicating that the Start was generated. However, the Lost Arb bit is set in the I2C_SCR register. The Address status is also set, indicating that the block has been addressed as a slave. The firmware may then ACK or NACK the address to continue the transfer.

   Slave mode is not enabled: A Start and address byte interrupt is generated. The Start Gen bit is cleared and the Lost Arb bit is set. The hardware will wait for command input, stalling the bus if necessary. In this case, the master will clear the I2C_SCR register, to release the bus and allow the transfer to continue, and the block will idle.

Other cases where the Start bit may be used to generate a Start condition are as follows.

1. When a master is finished with a transfer, a NACK is written to the I2C_SCR register (in the case of the Master receiver) or the transmit bit is cleared (in case of a Master transmitter). Normally, the action will free the stall and generate a Stop condition. However, if the Start bit is set and an address is written into the data register prior to the I2C_SCR write, a Stop, followed immediately by a Start (minimum bus free time), is generated. In this way, messages may be chained.
2. When a Master transmitter is NACK'ed, an automatic Stop condition is generated on the subsequent I2C_SCR write. However, if the Start Gen bit has previously been set, the Stop is immediately followed by a Start condition.

Table 21-6. I2C_MSCR Master Status/Control Register

Bit	Access	Description	Mode
3	R	Bus Busy This bit is set to '1' when any Start condition is detected and reset to '0' when a Stop condition is detected.	Master Only
2	R	Master Mode This bit is set to '1' when a start condition, generated by this block, is detected and reset to '0' when a stop condition is detected.	Master Only
1	RW	Restart Gen 1 = Generate a Restart condition. This bit is cleared by hardware when the Start generation is complete.	Master Only
0	RW	Start Gen 1 = Generate a Start condition and send a byte (address) to the I2C bus. This bit is cleared by hardware when the Start generation is complete.	Master Only

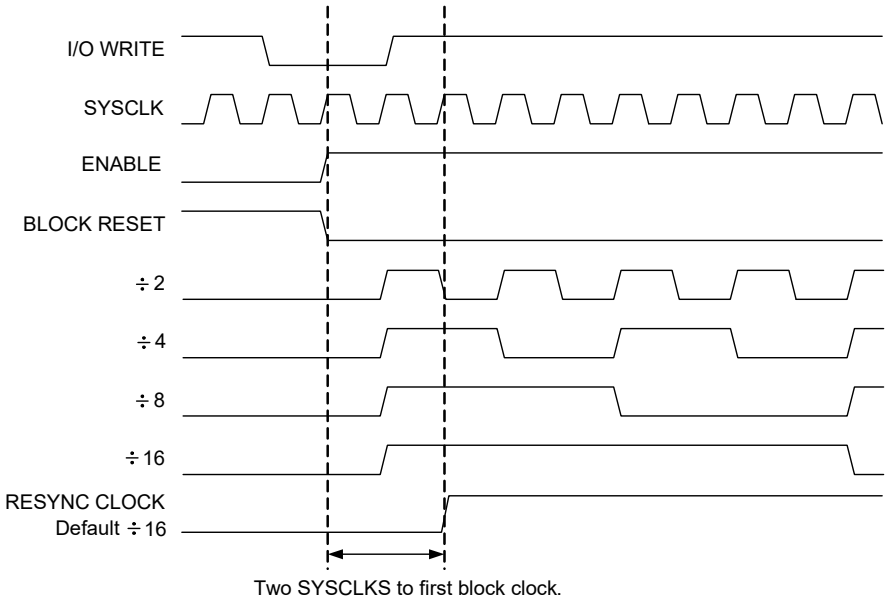
For additional information, refer to the I2C_MSCR register on page 165.

21.4 Timing Diagrams

21.4.1 Clock Generation

Figure 21-4 illustrates the I2C input clocking scheme. The SYSCLK pin is an input into a four-stage ripple divider that provides the baud rate selections. When the block is disabled, all internal state is held in a reset state. When either the Master or Slave Enable bits in the I2C_CFG register are set, the reset is synchronously released and the clock generation is enabled. Two taps from the ***ripple divider*** are selectable (/4, /16) from the clock rate bits in the I2C_CFG register. If any of the two divider taps is selected, that clock is resynchronized to SYSCLK. The resulting clock is routed to all of the synchronous elements in the design.

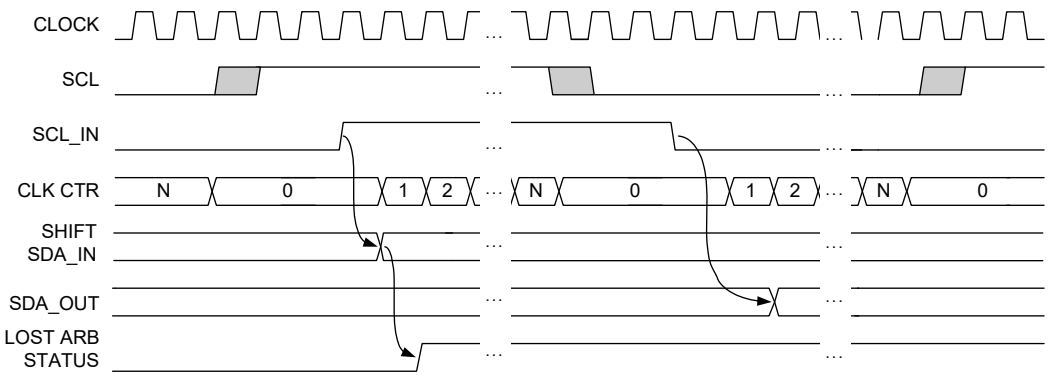
Figure 21-4. I²C Input Clocking



21.4.2 Basic Input/Output Timing

Figure 21-5 illustrates basic input output timing that is valid for both 16 times sampling and 32 times sampling. For 16 times sampling, N=4; and for 32 times sampling, N=12. N is derived from the half-bit rate sampling of eight and 16 clocks, respectively, minus the input latency of three (count of 4 and 12 correspond to 5 and 13 clocks).

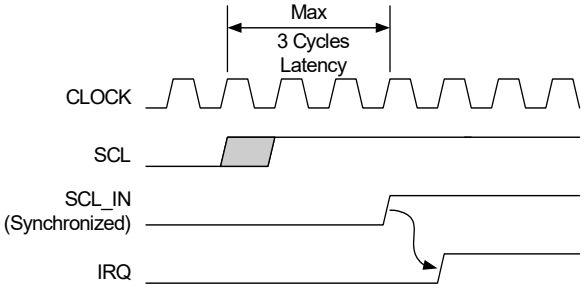
Figure 21-5. Basic Input/Output Timing



21.4.3 Status Timing

Figure 21-6 illustrates the interrupt timing for Byte Complete, which occurs on the positive edge of the ninth clock (byte + ACK/NACK) in Transmit mode and on the positive edge of the eighth clock in Receive mode. There is a maximum of three cycles of latency, due to the input synchronizer/filter circuit. As shown, the interrupt occurs on the clock following a valid SCL positive edge input transition (after the synchronizers). The Address bit is set with the same timing, but only after a slave address has been received. The LRB (Last Received Bit) status is also set with the same timing, but only on the ninth bit after a transmitted byte.

Figure 21-6. Byte Complete, Address, LRB Timing



Transmit: Ninth positive edge SCL
Receive: Eighth positive edge SCL

Figure 21-7 shows the timing for Stop Status. This bit is set (and the interrupt occurs) two clocks after the synchronized and filtered SDA line transitions to a '1', when the SCL line is high.

Figure 21-7. Stop Status and Interrupt Timing

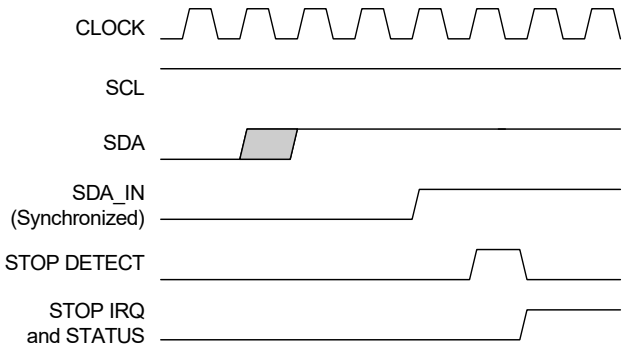
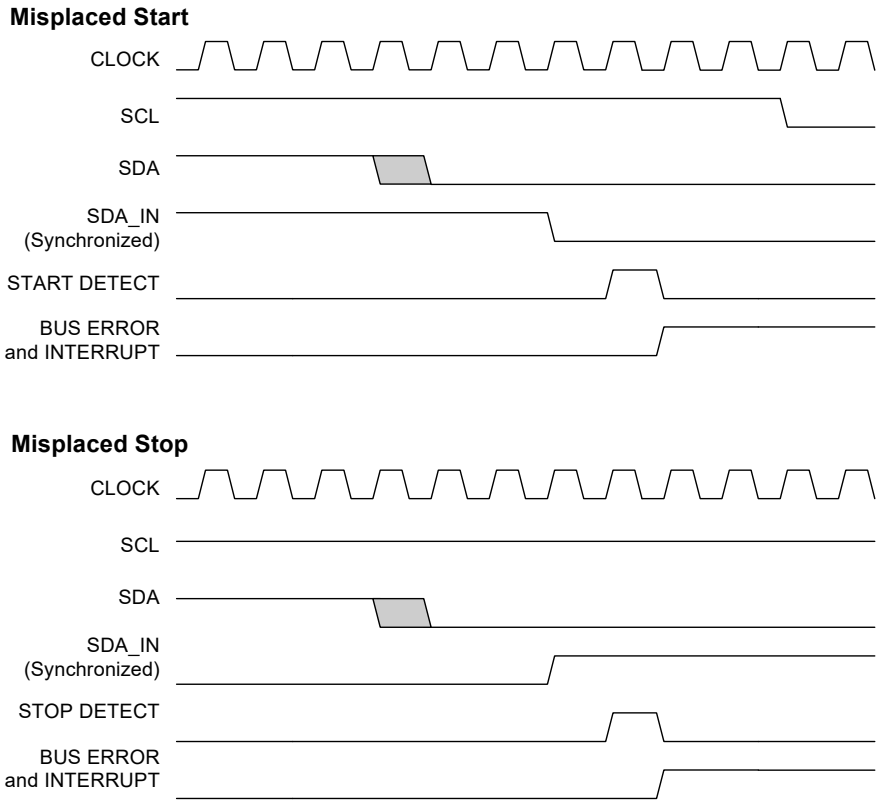


Figure 21-8 illustrates the timing for bus error interrupts. Bus Error status (and Interrupt) occurs one cycle after the internal Start or Stop Detect (two cycles after the filtered and synced SDA input transition).

Figure 21-8. Bus Error Interrupt Timing



21.4.4 Master Start Timing

When firmware writes the Start Gen command, hardware resynchronizes this bit to SYSCLK, to ensure a minimum of a full SYSCLK of setup time to the next clock edge. When the Start is initiated, the SCL line is left high for 6/14 clocks (corresponding to 16/32 times sampling rates). During this initial SCL high period, if an external Start is detected, the Start sequence is aborted and the block returns to an IDLE state. However, on the next Stop detection, the block will automatically initiate a new Start sequence.

Figure 21-9. Basic Master Start Timing

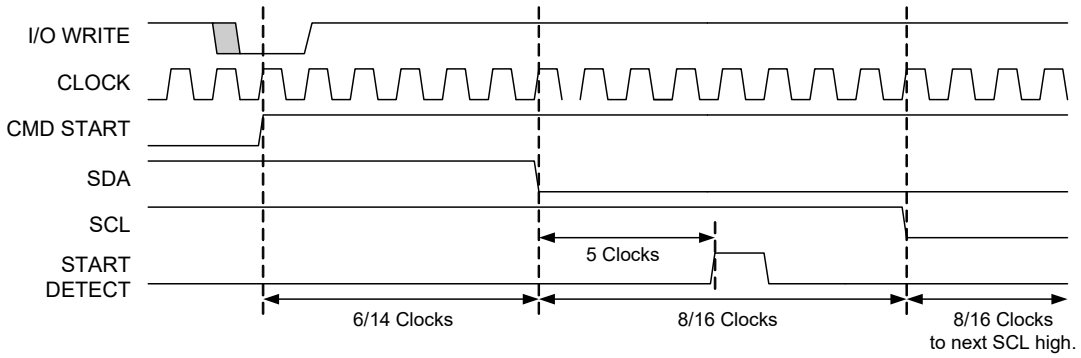


Figure 21-10. Start Timing with a Pending Start

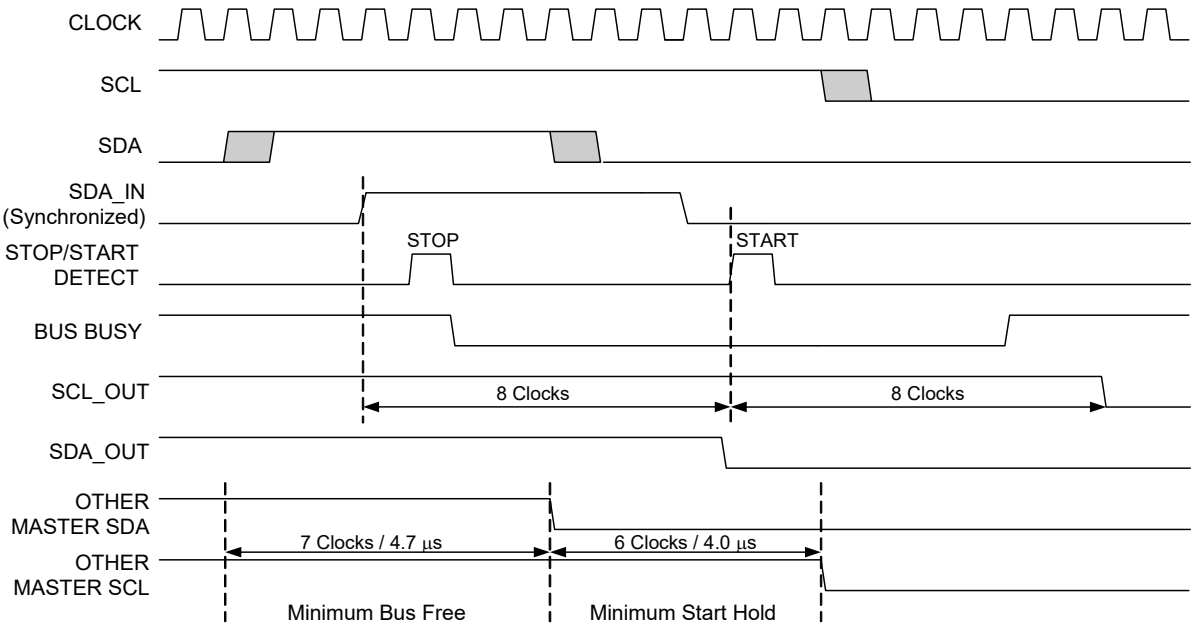
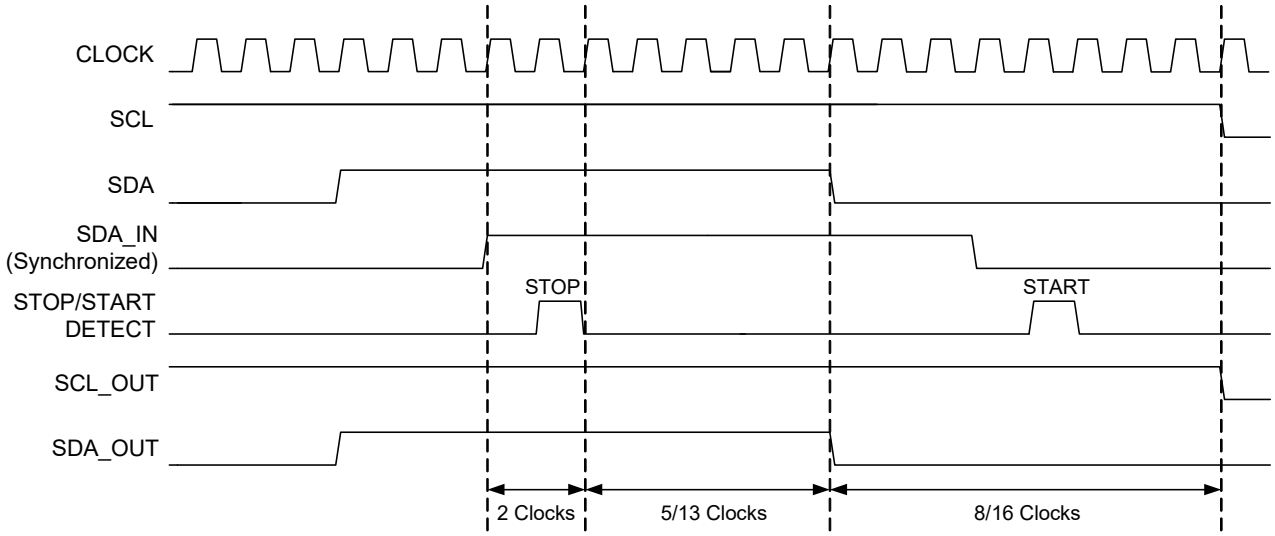


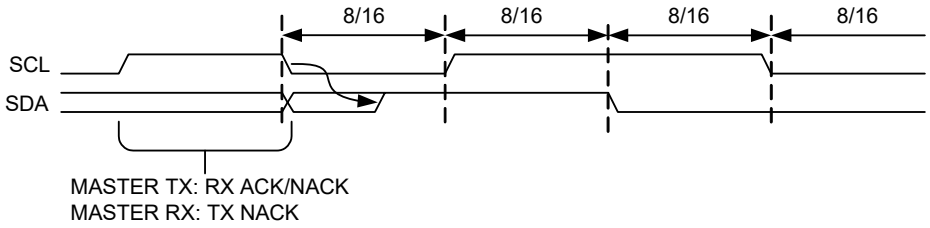
Figure 21-11. Master Stop/Start Chaining



21.4.5 Master Restart Timing

Figure 21-12 shows the Master Restart timing. After the ACK/NACK bit, the clock is held low for a half bit time (8/16 clocks corresponding to the 16 or 32 times sampling rates), during which time the data is allowed to go high, then a valid start is generated in the following 3 half bit times as shown.

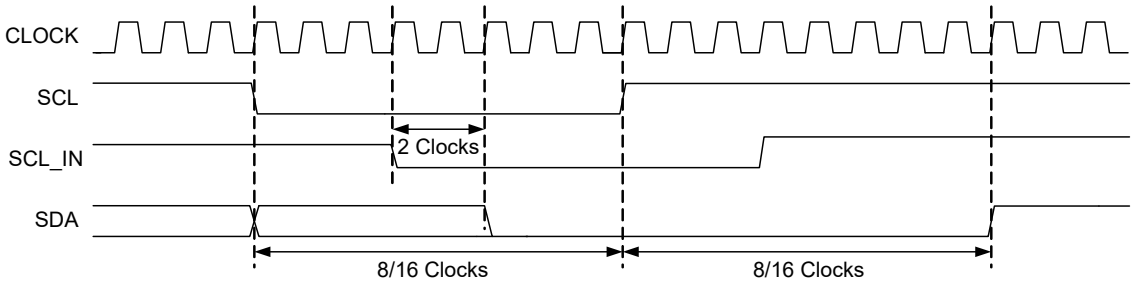
Figure 21-12. Master Restart Timing



21.4.6 Master Stop Timing

Figure 21-13 shows basic Master Stop timing. In order to generate a Stop, the SDA line is first pulled low, in accordance with the basic SDA output timing. Then, after the full low of SCL is completed and the SCL line is pulled high, the SDA line remains low for a full one-half bit time before it is pulled high to signal the Stop.

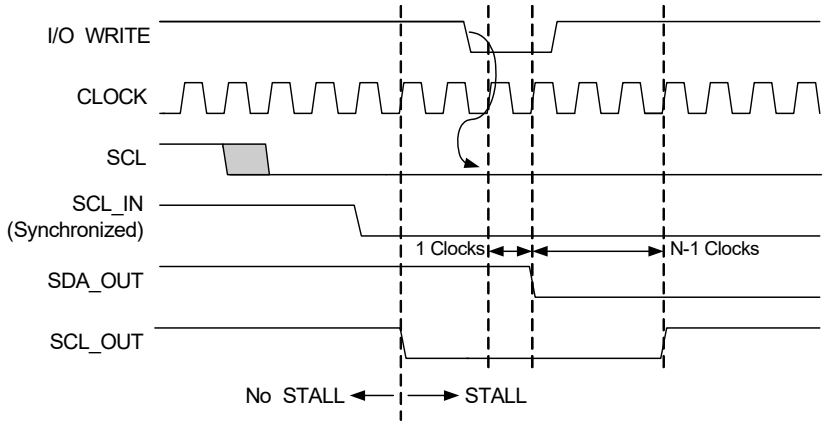
Figure 21-13. Master Stop Timing



21.4.7 Master/Slave Stall Timing

When a Byte Complete interrupt occurs, the PSoC device firmware must respond with a write to the I2C_SCR register to continue the transfer (or terminate the transfer). The interrupt occurs two clocks after the rising edge of SCL_IN (see “Status Timing” on page 405). As illustrated in Figure 21-14, firmware has until one clock after the falling edge of SCL_IN to write to the I2C_SCR register; otherwise, a stall occurs. Once stalled, the I/O write releases the stall. The setup time between data output and the next rising edge of SCL will always be N-1 clocks.

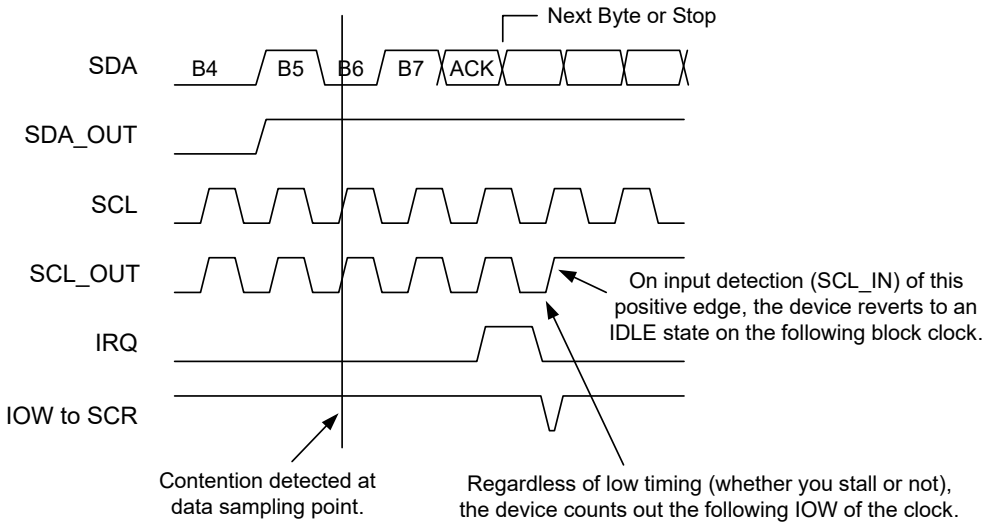
Figure 21-14. Master/Slave Stall Timing



21.4.8 Master Lost Arbitration Timing

Figure 21-15 shows a Lost Arbitration sequence. When contention is detected at the input (SDA_IN) sampling point, the SDA output is immediately released to an IDLE state. However, the master continues clocking until the Byte Complete interrupt, which is processed in the usual way. Any write to the I2C_SCR register results in the master reverting to an IDLE state, one clock after the next positive edge of the SCL_IN clock.

Figure 21-15. Lost Arbitration Timing (Transmitting Address or Data)

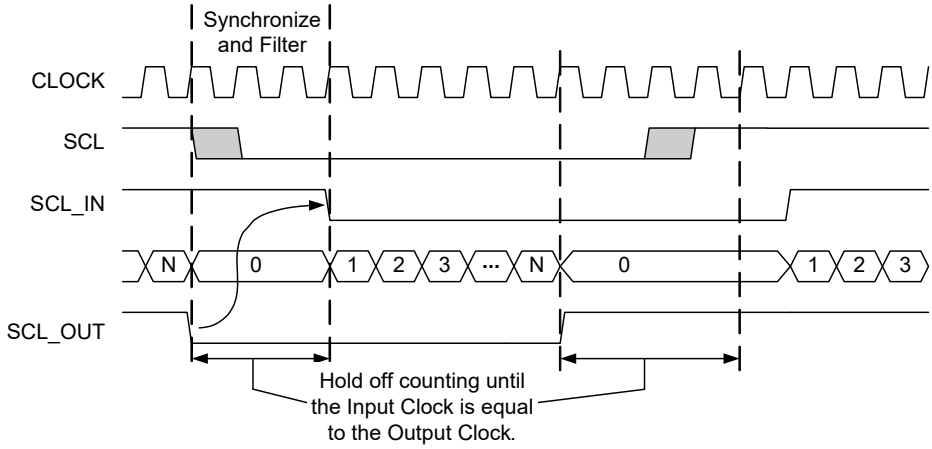


21.4.9 Master Clock Synchronization

Figure 21-16 shows the timing associated with Master Clock Synchronization. Clock synchronization is always operational, even if it is the only master on the bus. In which case, it is synchronizing to its own clock. In the wired AND bus, an SCL output of ‘0’ is seen by all masters. When the hardware asserts a ‘0’ to the output, it is immediately fed back from the PSoC device pin to the input synchronizer for the SCL input. The counter value (depending on the sampling rate) takes into account the worst case latency for input synchronization of three clocks, giving a net period of 8/16 clocks for both high and low time. This results in an overall clocking rate of 16/32 clocks per bit.

In multi-master environments when the hardware outputs a ‘1’ on the SCL output, if any other master is still asserting a ‘0’, the clock counter will hold until the SCL input line matches the ‘1’ on the SCL output line. When matched, the remainder of the high time is counted down. In this way, the master with the fastest frequency determines the high time of the clock and the master with the lowest frequency determines the low time of the clock.

Figure 21-16. Master Clock Synchronization



22. Internal Voltage Reference

This chapter discusses the Internal Voltage Reference and its associated register. The internal voltage reference provides an absolute value of 1.3 V to a variety of subsystems in the PSoC device. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

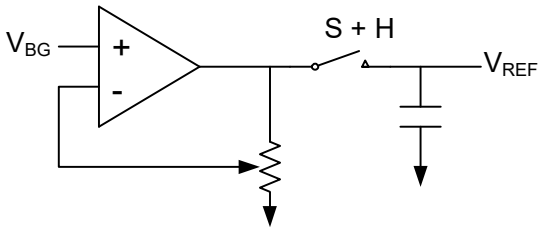
22.1 Architectural Description

The Internal Voltage Reference is made up of two blocks: a bandgap voltage generator and a buffer with sample and hold. The bandgap voltage generator is a typical (V_{BE} + K V_T) design.

The buffer circuit provides gain to the 1.20 V bandgap voltage, to produce a 1.30 V reference. A simplified ***schematic*** is illustrated in Figure 22-1. The connection between amplifier and capacitor is made through a CMOS switch, allowing the reference voltage to be used by the system while the reference circuit is powered down. The voltage reference is trimmed to 1.30 V at room temperature.

A temperature proportional voltage is also produced in this block for use in temperature sensing.

Figure 22-1. Voltage Reference Schematic



22.2 PSoC Device Distinctions

The internal voltage reference register, BDG_TR, is a read and write register with one exception: The CY8C27x43 PSoC device cannot read the BDG_TR register.

22.3 Register Definitions

The following register is associated with the Internal Voltage Reference. The Internal Voltage Reference is trimmed for gain and temperature coefficient using the BDG_TR register. The register description below has an associated register table showing the bit structure. The bits that are grayed out in the table are reserved bits and are not detailed in the register description that follows. Reserved bits should always be written with a value of '0'.

22.3.1 BDG_TR Register

Add.	Name	Cols.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,EAh	BDG_TR	4, 2		AGNDBYP	TC[1:0]		V[3:0]				RW : 00
		1			TC[1:0]		V[3:0]				

NOTE
The CY8C27x43 PSoC device cannot read this register.

The Bandgap Trim Register (BDG_TR) is used to adjust the bandgap and add an RC filter to AGND.

Depending on how many analog columns your PSoC device has (see the Cols. column in the register table above), only

certain bits are accessible to be read or written (refer to the table titled “PSoC Device Characteristics” on page 21).

Bit 6: AGNDBYP. When set, this bit adds an RC filter to AGND. (R is an internal 8.1K resistor and C is external to the PSoC device on P2[4].)

Bits 5 and 4: TC[1:0]. These bits are for setting the temperature coefficient inside the bandgap voltage generator. 10b is the design center for ‘0’ TC.

It is strongly recommended that the user not alter the value of these bits.

Bits 3 to 0: V[3:0]. These bits are for setting the gain in the reference buffer. Sixteen steps of 4 mV are available. 1000b is the design center for 1.30 V.

It is strongly recommended that the user not alter the value of these bits.

For additional information, refer to the BDG_TR register on page 263.

23. System Resets

This chapter discusses the System Resets and their associated registers. PSoC devices support several types of resets. The various resets are designed to provide error-free operation during power up for any voltage ramping profile, to allow for user-supplied external reset and to provide recovery from errant code operation. For a complete table of the System Reset registers, refer to the "Summary Table of the System Resource Registers" on page 374. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

23.1 Architectural Description

When reset is initiated, all registers are restored to their default states. In the Register Details chapter on page 103, this is indicated by the POR column in the register tables and elsewhere it is indicated in the Access column, values on the right side of the colon, in the register tables. Minor exceptions are explained below.

The following types of resets can occur in the PSoC device:

- Power on Reset (POR). This occurs at low supply voltage and is comprised of multiple sources.
- External Reset (XRES). This active high reset is driven into the PSoC device, on parts that contain an XRES pin.
- Watchdog Reset (WDR). This optional reset occurs when the watchdog timer expires, before being cleared by user firmware. Watchdog reset defaults to off.
- Internal Reset (IRES). This occurs during the boot sequence, if the SROM code determines that Flash reads are not valid.

The occurrence of a reset is recorded in the Status and Control registers (CPU_SCR0 for POR, XRES, and WDR) or in the System Status and Control Register 1 (CPU_SCR1 for IRESS). Firmware can interrogate these registers to determine the cause of a reset.

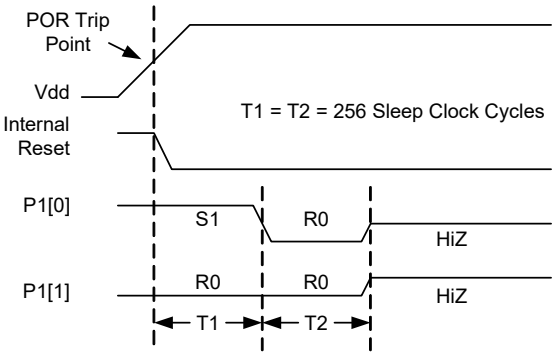
23.2 Pin Behavior During Reset

Power on Reset and External Reset cause toggling on two GPIO pins, P1[0] and P1[1], as described below and illustrated in Figure 23-1 and Figure 23-2. This allows programmers to synchronize with the PSoC device. All other GPIO pins are placed in a high impedance state during and immediately following reset.

23.2.1 GPIO Behavior on Power Up

At power up, the internal POR causes P1[0] to initially drive a strong high (1) while P1[1] drives a resistive low (0). After 256 sleep oscillator cycles, the P1[0] signal transitions to a resistive low state. After additional 256 sleep oscillator clocks, both pins transition to a high impedance state and normal CPU operation begins. This is illustrated in Figure 23-1.

Figure 23-1. P1[1:0] Behavior on Power Up

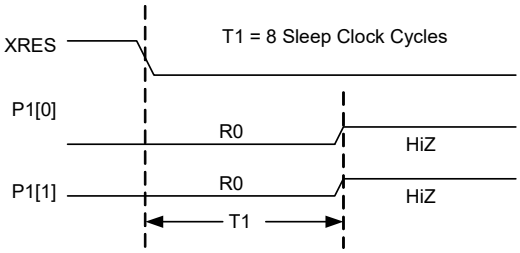


23.2.2 GPIO Behavior on External Reset

During External Reset (XRES=1), both P1[0] and P1[1] drive resistive low (0). After XRES de-asserts, these pins continue to drive resistive low for another 8 sleep clock cycles. After this time, both pins transition to a high impedance state and

normal CPU operation begins. This is illustrated in Figure 23-2.

Figure 23-2. P1[1:0] Behavior on External Reset (XRES)



23.3 Register Definitions

The following registers are associated with the PSoC System Resets and are listed in address order. Each register description has an associated register table showing the bit structure for that register. The bits in the tables that are grayed out are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of '0'. For a complete table of system reset registers, refer to the "Summary Table of the System Resource Registers" on page 374.

23.3.1 CPU_SCR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
x,FEh	CPU_SCR1	IRESS			SLIMO	ECO EXW *	ECO EX *		IRAMDIS	# : 00

LEGEND
x An "x" before the comma in the address field indicates that this register can be read or written to no matter what bank is used.
Access is bit specific. Refer to the Register Details chapter on page 103 for additional information.
* Bits 3 and 2 (ECO EXW and ECO EX, respectively) cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24x23 and CY8C22x13 PSoC devices.

The System Status and Control Register 1 (CPU_SCR1) is used to convey the status and control of events related to internal resets and watchdog reset.

Bit 7: IRESS. The Internal Reset Status bit is a read only bit that may be used to determine if the booting process occurred more than once.

When this bit is set, it indicates that the SROM SWBootReset code was executed more than once. If this bit is not set, the SWBootReset was executed only once. In either case, the SWBootReset code will not allow execution from code stored in Flash until the M8C Core is in a safe operating mode with respect to supply voltage and Flash operation. There is no need for concern when this bit is set. It is provided for systems which may be sensitive to boot time, so that they can determine if the normal one-pass boot time was exceeded. For more information on the SWBootReest code see the Supervisory ROM (SROM) chapter on page 43.

Bit 4: SLIMO. When set, the Slow IMO bit allows the active power dissipation of the PSoC device to be reduced by slowing down the IMO from 24 MHz to 6 MHz. The IMO trim value must also be changed when SLIMO is set (see "Engaging Slow IMO" on page 76). When not in external clocking mode, the IMO is the source for SYSCLK; therefore, when the speed of the IMO changes, so will SYSCLK.

Bit 3: ECO EXW. The ECO Exists Written bit is used as a status bit to indicate that the ECO EX bit has been previously written to. It is read only. When this bit is a '1', this indicates that the CPU_SCR1 register has been written to and is now locked. When this bit is a '0', the register has not been written to since the last reset event. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24x23 and CY8C22x13 PSoC devices.

Bit 2: ECO EX. The ECO Exists bit serves as a flag to the hardware, to indicate that an external crystal ***oscillator*** exists in the system. Just after boot, it may be written *only once* to a value of '1' (crystal exists) or '0' (crystal does not exist). If the bit is '0', a switch-over to the ECO is locked out by hardware. If the bit is '1', hardware allows the firmware to freely switch between the ECO and ILO. It should be written as early as possible after a ***Power On Reset (POR)*** or ***External Reset (XRES)*** event, where it is assumed that program execution integrity is high. Note that this bit cannot be used by the CY8C27x43 for silicon revision A, and by the CY8C24x23 and CY8C22x13 PSoC devices.

Bit 0: IRAMDIS. The Initialize RAM Disable bit is a control bit that is readable and writeable. The ***default value*** for this bit is '0', which indicates that the maximum amount of SRAM should be initialized on watchdog reset to a value of 00h. When the bit is '1', the minimum amount of SRAM is initialized after a watchdog reset. For more information on this bit, see the "SROM Function Descriptions" on page 44.

For additional information, refer to the CPU_SCR1 register on page 190.

23.3.2 CPU_SCR0 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,FFh	CPU_SCR0	GIES		WDRS	PORS	Sleep			STOP	# : XX

LEGEND
Access is bit specific. Refer to register detail for additional information.
XX The reset value is 10h after POR/XRES and 20h after a watchdog reset.

The System Status and Control Register 0 (CPU_SCR0) is used to convey the status and control of events for various functions of a PSoC device.

Bit 7: GIES. The Global Interrupt Enable Status bit is a read only status bit and its use is discouraged. The GIES bit is a legacy bit which was used to provide the ability to read the GIE bit of the CPU_F register. However, the CPU_F register is now readable. When this bit is set, it indicates that the GIE bit in the CPU_F register is also set which, in turn, indicates that the microprocessor will service interrupts.

Bit 5: WDRS. The WatchDog Reset Status bit may not be set. It is normally '0' and automatically set whenever a watchdog reset occurs. The bit is readable and clearable by writing a zero to its bit position in the CPU_SCR0 register.

Bit 4: PORS. The Power On Reset Status (PORS) bit, which is the watchdog enable bit, is set automatically by a POR or External Reset (XRES). If the bit is cleared by user code, the watchdog timer is enabled. Once cleared, the only way to reset the PORS bit is to go through a POR or XRES. Thus, there is no way to disable the watchdog timer, other than to go through a POR or XRES.

Bit 3: Sleep. The Sleep bit is used to enter Low Power Sleep mode when set. To wake up the system, this register bit is cleared asynchronously by any enabled interrupt. There are two special features of this register bit that ensures proper Sleep operation. First, the write to set the register bit is blocked, if an interrupt is about to be taken on that instruction boundary (immediately after the write). Second, there is a hardware interlock to ensure that, once set, the sleep bit may not be cleared by an incoming interrupt until the sleep circuit has finished performing the sleep sequence and the system-wide power down signal has been asserted. This prevents the sleep circuit from being interrupted in the middle of the process of system power down, possibly leaving the system in an indeterminate state.

Bit 0: STOP. The STOP bit is readable and writeable. When set, the PSoC M8C will stop executing code until a reset event occurs. This can be either a POR, WDR, or XRES. If an application wants to stop code execution until a reset, the preferred method would be to use the HALT instruction rather than a register write to this bit.

For additional information, refer to the CPU_SCR0 register on page 191.

23.4 Timing Diagrams

23.4.1 Power On Reset

A Power on Reset (POR) is triggered whenever the supply voltage is below the POR trip point. POR ends once the supply voltage rises above this voltage. Refer to the POR and LVD chapter on page 419 for more information on the operation of the POR block.

POR consists of two pieces: an imprecise POR (IPOR) and a Precision POR (PPOR). “POR” refers to the OR of these two functions. IPOR has coarser accuracy and its trip point is typically lower than PPOR’s trip point. PPOR is derived from a circuit that is calibrated (during boot), for a very accurate location of the POR trip point.

During POR (POR=1), the IMO is powered off for low power during start-up. Once POR de-asserts, the IMO is started (see Figure 23-4).

POR configures register reset status bits as shown in Table 23-1. PPOR does not affect the BandGap Trim register (BDG_TR), but IPOR does reset this register.

23.4.2 External Reset

An External Reset (XRES) is caused by pulling the XRES pin high. The XRES pin has an always-on, pull down resistor, so it does not require an external pull down for operation and can be tied directly to ground or left open. Behavior after XRES is similar to POR.

During XRES (XRES=1), the IMO is powered off for low power during start-up. Once XRES de-asserts, the IMO is started (see Figure 23-4). How the XRES configures register reset status bits is shown in Table 23-1.

23.4.3 Watchdog Timer Reset

The user has the option to enable the Watchdog Timer Reset (WDR), by clearing the PORS bit in the CPU_SCR0 register. Once the PORS bit is cleared, the watchdog timer cannot be disabled. The only exception to this is if a POR/XRES event takes place, which will disable the WDR. Note that a WDR does not clear the Watchdog timer. See “Watchdog Timer” on page 98 for details of the Watchdog operation.

When the watchdog timer expires, a watchdog event occurs resulting in the reset sequence. Some characteristics unique to the WDR are as follows.

- PSoC device reset asserts for one cycle of the CLK32K clock (at its reset state).
- The IMO is not halted during or after WDR (that is, the part does not go through a low power phase).
- CPU operation re-starts one CLK32K cycle after the internal reset de-asserts (see Figure 23-3).

How the WDR configures register reset status bits is shown in Table 23-1.

Figure 23-3. Key Signals During WDR and IRES

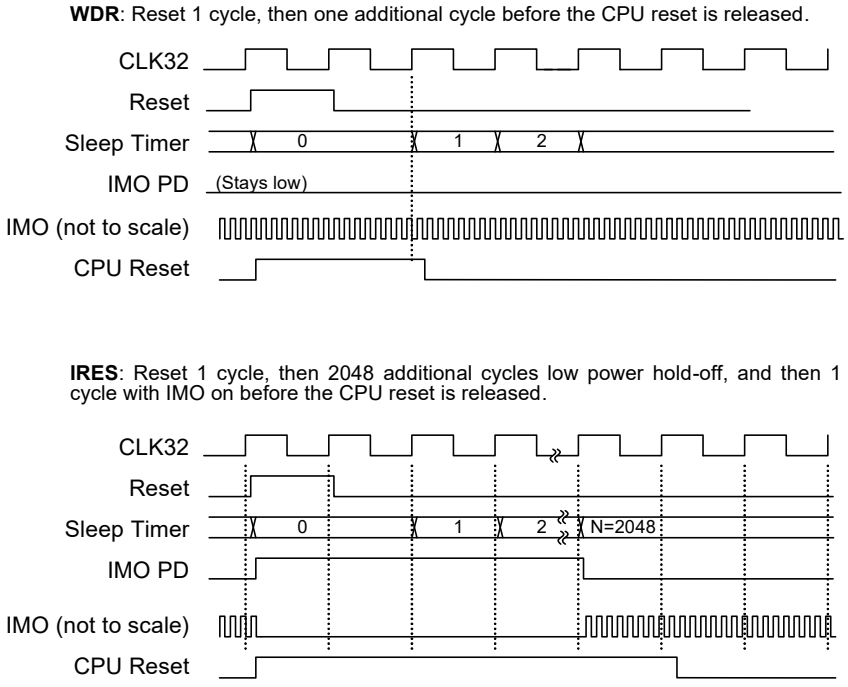
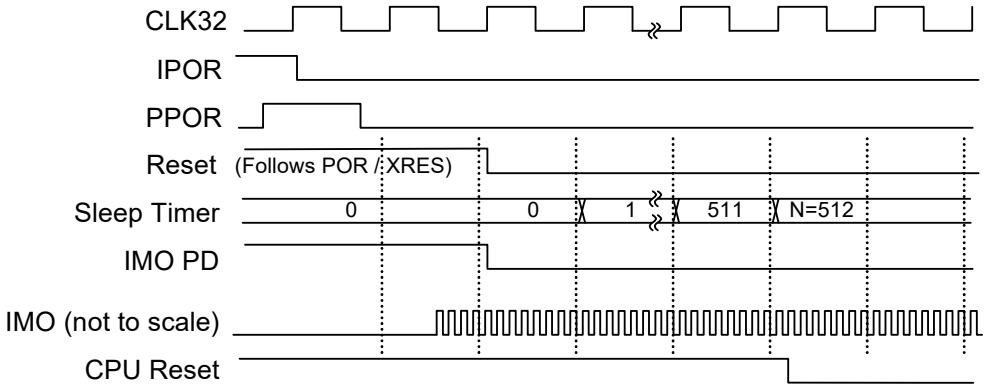
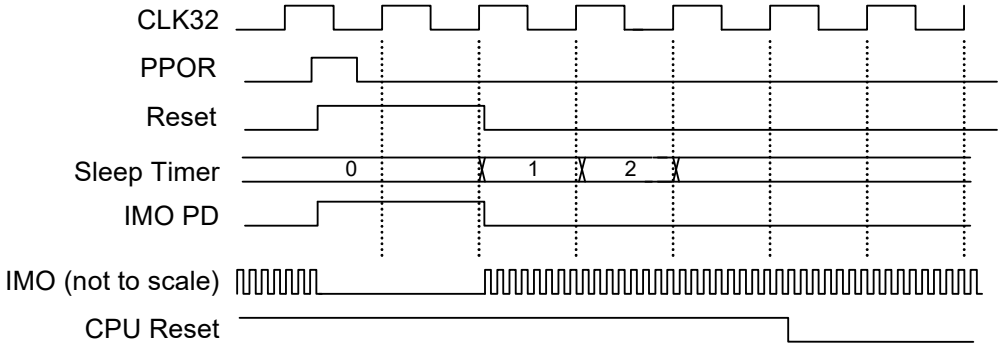


Figure 23-4. Key Signals During POR and XRES

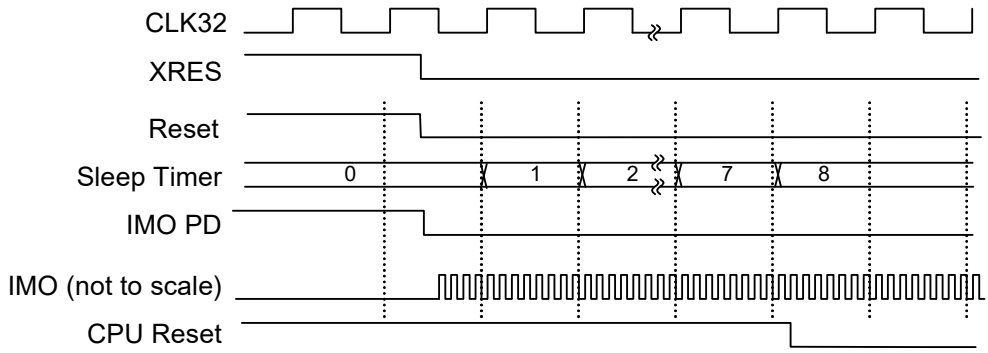
POR (IPOR followed by PPOR): Reset while POR is high (IMO off), then 511(+) cycles (IMO on), and then the CPU reset is released. **XRES** is the same, with N=8.



PPOR (with no IPOR): Reset while PPOR is high and to the end of the next 32K cycle (IMO off); 1 cycle IMO on before the CPU reset is released. Note that at the 5V level, PPOR will tend to be brief, because the reset clears the POR range register (VLT_CR) back to the default 3V setting.



XRES: Reset while XRES is high (IMO off), then 7(+) cycles (IMO on), and then the CPU reset is released.



23.4.4 Reset Details

Timing and functionality details are summarized in Table 23-1. Figure 23-4 shows some of the relevant signals for IPOR, PPOR, and XRES, while Figure 23-3 shows signaling for WDR and IRES.

Table 23-1. Details of Functionality for Various Resets

Item	IPOR (Part of POR)	PPOR (Part of POR)	XRES	WDR
Reset Length	While POR=1	While PPOR=1, plus 30-60 µs (1-2 clocks)	While XRES=1	30 µs (1 clock)
Low Power (IMO Off) During Reset?	Yes	Yes	Yes	No
Low Power Wait Following Reset?	No	No	No	No
CLK32K Cycles from End of Reset to CPU Reset Deasserts[a]	512[b]	1[d]	8[d]	1[d]
Register Reset (See next line for CPU_SCR0, CPU_SCR1)	All	All, except PPOR does not reset Bandgap Trim register	All	All
Reset Status Bits in CPU_SCR0, CPU_SCR1	Set PORS, Clear WDRS, Clear IRAMDIS	Set PORS, Clear WDRS, Clear IRAMDIS	Set PORS, Clear WDRS, Clear IRAMDIS	Clear PORS, Set WDRS, IRAMDIS unchanged
Bandgap Power	On	On	On	On
Boot Time[c]	2.2 ms	2.2 ms	2.2 ms	2.2 ms

a. CPU reset is released after synchronization with the CPU Clock.
b. The CLK32K clock source (Internal Low Speed Oscillator, ILO) is not trimmed at this point in the boot sequence. The F_{32KU} specification from the device data sheet should be used to determine the slowest untrimmed ILO frequency and the maximum reset recovery time.
c. Measured from CPU reset release to execution of the code at Flash address 0x0000.

23.5 Power Consumption

The ILO block drives the CLK32K clock used to time most events during the reset sequence. This clock is powered down by IPOR, but not by any other reset. The sleep timer provides interval timing.

While POR or XRES assert, the IMO is powered off to reduce start-up power consumption.

During and following IRES (for 64 ms nominally), the IMO is powered off for low average power during slow supply ramps.

During and after POR or XRES, the bandgap circuit is powered up.

Following IRES, the bandgap circuit is only powered up occasionally, to refresh the sampled bandgap voltage value. This sampling follows the same process used during sleep mode.

The IMO is always on for at least one CLK32K cycle, before CPU reset is de-asserted.

24. POR and LVD

This chapter briefly discusses the POR and LVD circuits and their associated registers. For a complete table of the POR and LVD registers, refer to the “Summary Table of the System Resource Registers” on page 374. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

24.1 Architectural Description

The Power on Reset (POR) and Low Voltage Detect (LVD) circuits provide protection against low voltage conditions. The POR function senses Vdd and holds the system in reset until the magnitude of Vdd will support operation to specification. The LVD function senses Vdd and provides an interrupt to the system when Vdd falls below a selected ***threshold***. Other outputs and status bits are provided to indicate important voltage trip levels.

Refer to Section 23.2 Pin Behavior During Reset for a description of GPIO pin behavior during power up.

24.2 PSoC Device Distinctions

For the CY8C21345, CY8C22x45, CY8C24x23A, CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices, the lowest POR and LVD trip level is set for 2.4 V operation; the next lowest is set for 3.0 V operation (instead of 3.0 V or 4.5 V operation). Refer to the PSoC data sheets for electrical specification information.

24.3 Register Definitions

The following registers are associated with the POR and LVD, and are listed in address order. The register descriptions below have an associated register table showing the bit structure. Depending on how many analog columns your PSoC device has (see the Cols. column in the register tables below), only certain bits are accessible to be read or written (refer to the table titled “PSoC Device Characteristics” on page 21).

The bits that are grayed out in the register tables are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of ‘0’. For a complete table of the POR and LVD registers, refer to the “Summary Table of the System Resource Registers” on page 374.

24.3.1 VLT_CR Register

Add.	Name	Cols.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E3h	VLT_CR	4, 2	SMP		PORLEV[1:0]		LVDTBEN	VM[2:0]			RW : 00
		1			PORLEV[1:0]		LVDTBEN	VM[2:0]			

The Voltage Monitor Control Register (VLT_CR) is used to set the trip points for POR, LVD, and the supply pump.

The VLT_CR register is cleared by all resets, which can cause reset cycling during very slow supply ramps to 5 V when the POR range is set for the 5 V range. This is because the reset clears the POR range setting back to 3 V and a new boot/start-up occurs (possibly many times). The user can manage this with Sleep mode and/or reading voltage status bits, if such cycling is an issue.

Bit 7: SMP. This bit controls whether or not the SMP will turn on when the supply (Vdd) voltage has dropped below the trip point set by VM[2:0]. The SMP is enabled when the SMP bit is '0'. Thus, the SMP is on by default. If this bit is set to '1' the SMP will not turn on regardless of the supply voltage level. Refer to the table titled "Availability of System Resources for PSoC Devices" on page 373 to determine if your PSoC device can use this bit.

Bits 5 and 4: PORLEV[1:0]. These bits set the Vdd level at which PPOR switches to one of three valid values. Note that 11b is a reserved value and therefore, should not be used.

The three valid settings for these bits are:

- 00b (3 V or 2.4 V operation)
- 01b (4.5 V or 3.0 V operation)
- 10b (4.75 V operation)

See the "DC POR and LVD Specifications" table in the Electrical Specifications section of the PSoC device data sheet for voltage tolerances for each setting.

Bit 3: LVDTBEN. This bit is AND'ed with LVD to produce a throttle-back signal that reduces CPU clock speed when low voltage conditions are detected. When the Throttle-Back signal is asserted, the CPU speed bits in the OSC_CR0 register are reset, forcing the CPU speed to 3 MHz or EXTCLK / 8.

Bits 2 to 0: VM[2:0]. These bits set the Vdd level at which LVD and the Pump Comparator switches. Refer to the table titled "Availability of System Resources for PSoC Devices" on page 373 to determine if your PSoC device has the Switch Mode Pump (SMP).

See the "DC POR and LVD Specifications" table in the Electrical Specifications section of the PSoC device data sheet for voltage tolerances for each setting.

For additional information, refer to the VLT_CR register on page 257.

24.3.2 VLT_CMP Register

Add.	Name	Cols.	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,E4h	VLT_CMP	4, 2						PUMP	LVD	PPOR	R : 00
		2L *					NoWrite	PUMP	LVD	PPOR	
		1							LVD	PPOR	

* The 2L column row is only applicable to the CY8C21x34, CY8C21x23, CY7C603xx, and CYWUSB6953 PSoC devices, which have two column limited functionality.

The Voltage Monitor Comparators Register (VLT_CMP) is used to read the state of internal supply voltage monitors.

Bit 3: NoWrite. This bit is only used in PSoC devices with a 2.4 V minimum POR. It reads the state of the Flash write voltage monitor.

Bit 2: PUMP. This bit reads the state of the Switch Mode Pump Vdd comparator. The trip points for both LVD and PUMP are set by VM[2:0] in the VLT_CR register. Refer to the table titled “Availability of System Resources for PSoC Devices” on page 373 to determine if your PSoC device can use this bit.

Bit 1: LVD. This bit reads the state of the low voltage detect comparator. The trip points for both LVD and PUMP are set by VM[2:0] in the VLT_CR register. Refer to the table titled “Availability of System Resources for PSoC Devices” on page 373 to determine if your PSoC device can use this bit.

Bit 0: PPOR. This bit reads back the state of the PPOR output. This can only be meaningfully read with PORLEV[1:0] set to disable PPOR. In that case, the PPOR status bit shows the comparator state directly.

For additional information, refer to the VLT_CMP register on page 258.

25. I/O Analog Multiplexer

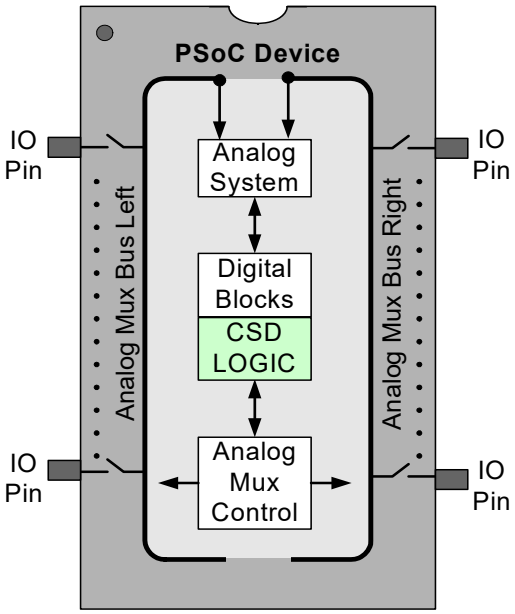
This chapter explains the chip-wide I/O Analog Multiplexer for the CY8C22x45 PSoC device and its associated registers. For a complete table of the I/O Analog Multiplexer registers, refer to the "Summary Table of the System Resource Registers" on page 374. For a quick reference of all PSoC registers in address order, refer to the Register Details chapter on page 103.

25.1 Architectural Description

The PSoC device contains an enhanced analog multiplexer (mux) capability. This function allows many I/O pins to connect to a common internal analog bus. In the CY8C22x45 PSoC device, all I/O pins connect to this bus.

Any number of pins can be connected simultaneously, and dedicated support circuitry allows selected pins to be alternately charged high or connected to the bus. The analog bus can be connected as an input into either the positive or negative inputs of any analog continuous time (CT) block. A block diagram is shown in Figure 25-1.

Figure 25-1. Analog Mux System



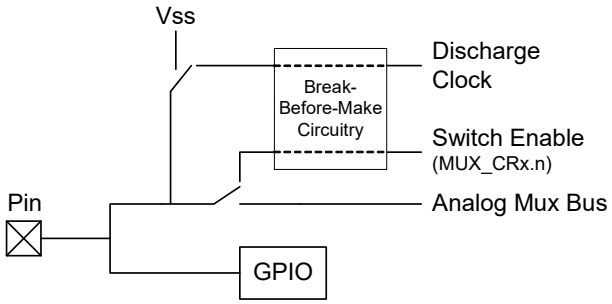
In the CY8C22x45 PSoC device, as in the CY8C24x94 and CY7C64215 PSoC devices, the Analog Mux Bus can be split into two separate nets, as shown in Figure 25-1. The two analog mux nets can be connected to different analog columns for simultaneous signal processing.

25.1.1 IOMUX and GPIO

For each pin, the mux capability exists in parallel with the normal GPIO cell described in the General Purpose I/O (GPIO) chapter on page 68 and shown in Figure 25-2. Normally, the associated GPIO pin is put into a high-impedance state for these applications, although there are cases where the GPIO cell is configured by the user to briefly drive pin initialization states as described below.

Pins are individually connected to the internal bus by setting the corresponding bits in the MUX_CRx registers. Any number of pins can be enabled at the same time. At reset, all of these mux connections are open (disconnected).

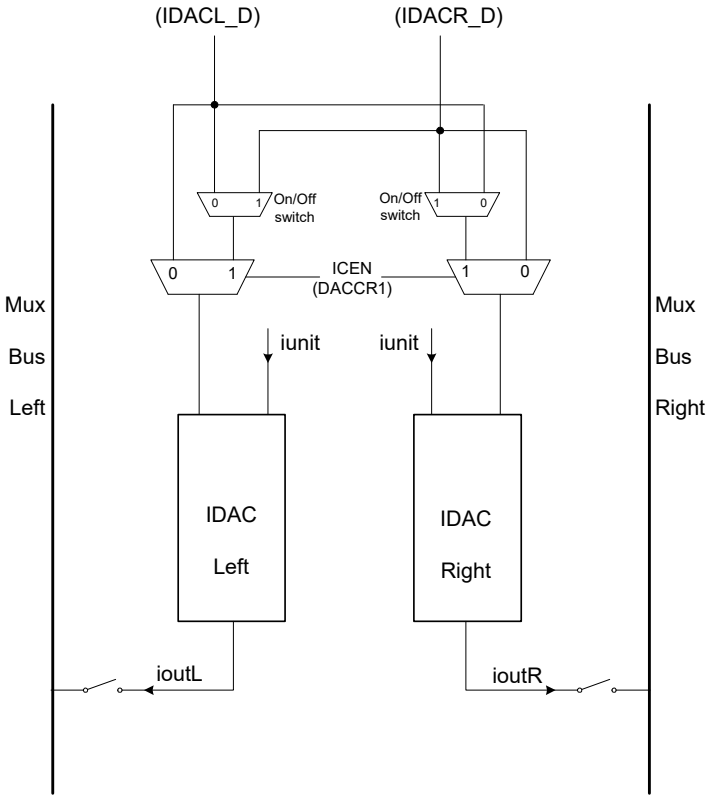
Figure 25-2. I/O Pin configuration



25.1.2 Dual Channel 8-Bit IDAC

The dual channel IDAC can provide two independent current sources based on the input current reference IUNIT. IUNIT is generated from PLL and can be trimmed in register IDAC_CR1. These two current sources can vary from 0 to 637.5 uA with 255 steps (IDACL_D and IDACR_D), and they are connected to analog mux bus left and analog mux bus right independently. These two current sources can act the same or independently due to the value of ICEN in register IDAC_CR1.

Figure 25-3. dual channel IDAC



Therefore, this IDAC block supports 2 different operation modes, depending on 'ICEN' bit setting.

Table 25-1. IDAC Operation Mode

Normal Mode	ICEN = 0	Left channel output current is decided by IDACL_D register. And right channel output current is decided by IDACR_D register. Both of them can be cut-off depend on their switch ON/OFF signal.
Compensation mode	ICEN = 1	The switches won't be cut-off even if he switch control signal is 'OFF'. Both channels will output current based on IDACR_D when the corresponding switch is 'ON'. And both channels will output current based on IDACL_D when the corresponding switch is 'OFF'.

25.2 PSoC Device Distinctions

The CY8C22x45, CY8C21345, CY8C24x94, CY8C21x34, CY7C64215, CY7C603xx, and CYWUSB6953 PSoC devices differ from the other PSoC devices in that GPIO pins can connect to the internal analog bus. The CY8C22x45, CY8C21345, CY8C24x94, and CY7C64215 contain the additional capability to optionally split the analog bus into two separate sections. In the CY8C22x45, CY8C21345, CY8C21x34, CY7C603xx, and CYWUSB6953, all GPIO pins are enabled for this connection. In the CY8C24x94 and CY7C64215, all pins in Ports 0 through 5 are enabled for connection to the analog bus.

25.3 Application Description

The analog mux circuitry enables a variety of unique applications such as those explained in the following sections.

25.3.1 Capacitive Sensing

The analog mux supports capacitive sensing applications through the use of the I/O analog multiplexer and its control circuitry. Two off-chip capacitors are normally connected to the analog mux bus. One is the sense capacitor being measured and the other is an integration capacitor that accumulates charge from the sense capacitor. The integration capacitor is initialized (low) under firmware control, using its pin's GPIO cell. After that, the capacitor is charged through charge-sharing with the sense capacitor.

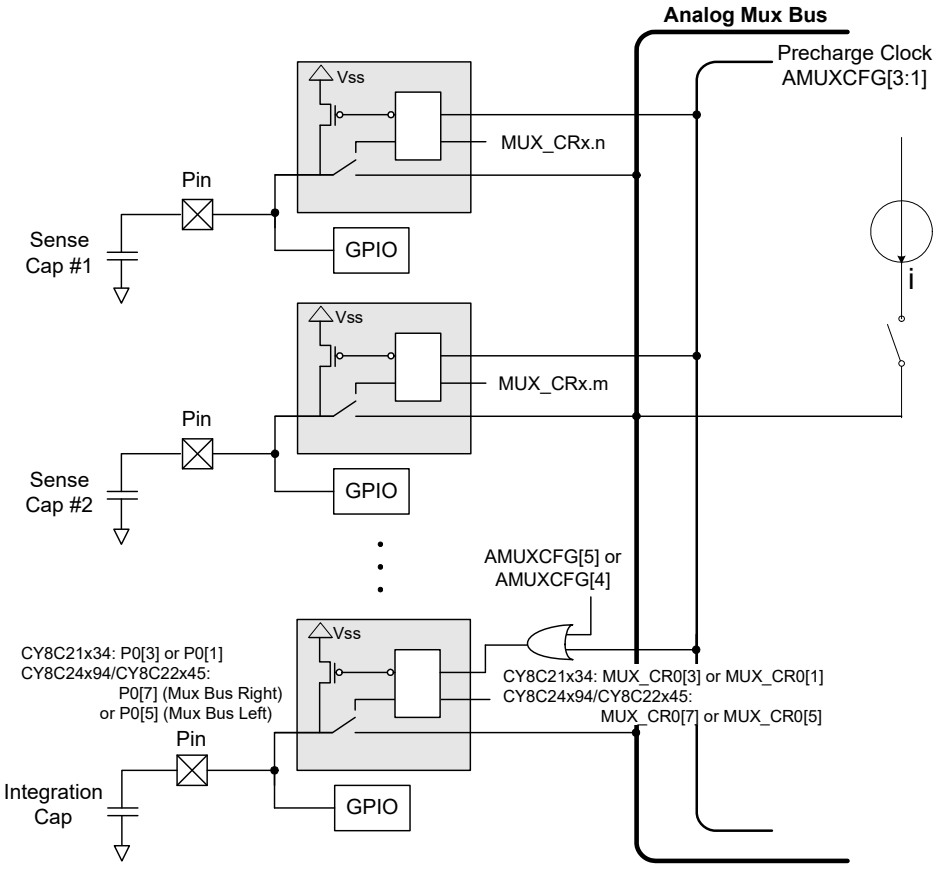
The sense capacitor can be automatically initialized and sensed for a number of cycles, in order to build up sufficient charge on the integration capacitor. Several clocking choices are available for selection in the AMUX_CFG register. The ***break-before-make*** circuitry is contained in each pin's mux so that each cycle's initialization of the sense capacitor does not disturb the internal bus. In the CY8C22x45 PSoC device, the sense capacitor is discharged to Vss and then released and reconnected to the analog mux for charge transfer from the integration capacitor. (For other chips that have IOMUX function, the sense capacitor is charged to Vdd and then released and reconnected to the analog mux for charge transfer to the integration capacitor.)

Charge accumulation on the integration capacitor continues for the time you set. The integration capacitor voltage, seen on the analog mux bus, is typically compared against a reference such as the bandgap. Detecting a capacitance change is often more important than an absolute measurement, and a change in the charging time can indicate such a difference. A system with several sense capacitors can be measured in sequence, using the same integration capacitor.

A pin used as the integration capacitor is not switched during this process, so it remains connected to the analog mux. Two Port 0 pins are available for this function, as shown in Figure 25-4.

In order to activate the charge transfer mode, the precharge clock must be set to any state except the reset state. In the reset state, the mux connections are static, controlled only by the MUX_CRx register settings. The CY8C22x45, CY8C21345, CY8C24x94, and CY7C64215 PSoC devices can be configured to have two Analog Mux Bus nets because they support two simultaneous capacitive sensing operations.

Figure 25-4. Capacitance Sense Example Diagram



25.3.2 Chip-Wide Analog Input

The analog bus forms a multiplexer across many I/O pins. This allows any of these pins to be brought into the analog system for processing, as shown in Figure 25-1. The Port 0 pins are also brought through separate mux paths to the continuous time block, so Port 0 inputs can be routed to the analog system by either path. In the CY8C22x45, CY8C24x94, and CY7C64215, some Port 2 inputs have a dedicated path to switched capacitor blocks.

In the CY8C21x34, CY7C603xx, and CYWUSB6953 PSoC devices, the pins can be connected to a single bus. In the CY8C22x45, CY8C24x94, and CY7C64215 PSoC devices, odd pins are connected to one bus, and even pins to the other bus. The two mux buses can be shorted together using the switch controlled by the SplitMux bit.

25.3.3 Crosspoint Switch

The bi-directional nature of the analog mux switches allows a direct connection between any of the I/O pins, as shown in Figure 25-1. Enabling two (or more) pins at the same time connects these pins together, with approximately 400 ohms of resistance between each pin and the analog mux bus. As long as the clock choice in the AMUX_CFG register is set to the fixed '0' case, the switches will be static, controlled only by the state of the individual switch enable bits in the MUX_CRx registers. The crosspoint can be reconfigured at any time and the user can provide a break-before-make function with firmware if needed.

25.3.4 Charging Current

The analog mux bus can be connected to the dedicated charging current. This enables applications such as capacitor measurement with this current instead of charge sharing. The IDACx_D and IDAC_CR registers control this configurable current. If the CY8C24x94 and CY7C64215 PSoC devices are configured with a split analog mux bus, this current connects only to the right-side bus (even pin numbers). In the CY8C22x45 PSoC device, there is a 2-channel IDAC. If both split analog mux buses are connected to one separate IDAC channel, then it is different from CY8C24x94 and CY7C64215.

25.4 Register Definitions

The following registers are only associated with the Analog Bus Mux in the CY8C22x45, CY8C24x94, CY8C21x34, CY7C64215, CY7C603xx, and CYWUSB6953 PSoC devices and are listed in address order within their system resource configuration. For a complete table of the I/O Analog Multiplexer registers, refer to the "Summary Table of the System Resource Registers" on page 374. Each register description has an associated register table showing the bit structure for that register. Register bits that are grayed out throughout this document are reserved bits and are not detailed in the register descriptions that follow. Reserved bits should always be written with a value of '0'.

25.4.1 AMUX_CFG Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,61h	AMUX_CFG	BCol1Mux	ACol0Mux	INTCAP[1:0]		MUXCLK[2:0]			EN	RW : 00

The Analog Mux Configuration Register (AMUX_CFG) is used to configure the clocked pre-charge mode of the analog multiplexer system. This register is only used by the CY8C22x45, CY8C21345, CY8C24x94, CY8C21x34, CY7C64215, CY7C603xx, and CYWUSB6953 PSoC devices.

Bit 7: BCol1Mux. This bit selects the column 1 port input. It picks between port 0 inputs or the analog mux bus. This bit is only available in the CY8C24x94 and CY7C64215 PSoC devices. See Figure 17-8, "Two Column Limited Analog Pin Block Diagram for the CY8C22x45, CY8C21x34, CY7C603xx, and CYWUSB6953," on page 345.

Bit 6: ACol0Mux. This bit selects the column 0 port input. It picks between port 0 inputs or the analog mux bus. This bit is only available in the CY8C24x94 and CY7C64215 PSoC devices. See Figure 17-8, "Two Column Limited Analog Pin Block Diagram for the CY8C22x45, CY8C21x34, CY7C603xx, and CYWUSB6953," on page 345.

Bits 5 and 4: INTCAP[1:0]. These bits are used to choose static connections to the analog mux bus even if the mux clocking is enabled in the MUXCLK[2:0] setting.

Bits 3 to 1: MUXCLK[2:0]. These bits select the precharge clock that drives the switching on the analog mux. The default choice is to have no clocking and no precharge.

Bit 0: EN. This bit enables the clock output. When the block is disabled, the output is '0'.

For additional information, refer to the AMUX_CFG register on page 128.

25.4.2 IDACx_D Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,FCh	IDACR_D	DACDATA[7:0]								RW : 00
0,FDh	IDACL_D	DACDATA[7:0]								RW : 00

The Analog Mux right DAC Data Register (IDACR_D) and the Analog Mux left DAC Data Register (IDACL_D) specify the 8-bit multiplying factor that determines the output DAC current. These registers are used by the CY8C21x45 and CY8C22x45 PSoC devices.

Bits 7 to 0: DACDATA[7:0]. The 8-bit value in this register sets the current driven onto the analog mux bus when the current DAC mode is enabled.

For additional information, refer to the IDACx_D register on page 189.

25.4.3 AMUX_CFG1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,6Ah	AMUX_CFG1			GOINVR	GOINVL	AMUXMODR		AMUXMODL		RW : XX

LEGEND
X The value after power on reset is unknown.

Bit 5/4: GOINVR/GOINVL. 1: AMUXCLK is inverted when it goes to GOO.

Note that the GOO output enable control bit is in IDAC_CRx.

Bits 3 and 2/1 and 0: AMUXMODR/AMUXMODL. Used to select AMUX clock source for left side and right side.

00b Select original AMUX clock decided by 0,61.
01b Select CSD_PO<0> as AMUX clock.
10b Select CSD_PO<1> as AMUX clock.
11b Reserved.

For additional information, refer to the AMUX_CFG1 register on page 229.

25.4.4 MUX_CRx Registers

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,D8h	MUX_CR0	ENABLE[7:0]								RW : 00
1,D9h	MUX_CR1	ENABLE[7:0]								RW : 00
1,DAh	MUX_CR2	ENABLE[7:0]								RW : 00
1,DBh	MUX_CR3	ENABLE[7:0]								RW : 00
1,ECh	MUX_CR4	ENABLE[7:0]								RW : 00

The Analog Mux Port Bit Enables Registers (MUX_CRx) are used to control the connection between the analog mux bus and the corresponding pin.

The CY8C22x45, CY8C21345, CY8C21x34, CY7C603xx, and CYWUSB6953 have a 4-bit wide Port 3. The upper 4 bits of the MUX_CR3 register are reserved in that device and will return zeros when read. Additionally, the CY8C22x45 has a 6-bit wide Port 4 and the highest 2 bits of the MUX_CR4 register are reserved and will return zeros when read. The MUX_CRx registers with addresses 1,ECh and 1,EDh are only used by the CY8C24x94 and CY7C64215 PSoC devices.

Bits 7 to 0: ENABLE[7:0]. The bits in these registers enable connection of individual pins to the analog mux bus. Each I/O port has a corresponding MUX_CRx register.

For additional information, refer to the MUX_CRx register on page 249.

25.4.5 IDAC_CR0 Register

The Analog Mux DAC Control Registers (IDAC_CRx) contain the control bits for the DAC current that drives the analog mux bus and for selecting the split configuraton for the CY8C22x45, CY8C21345, CY8C24x94, and CY7C64215 PSoC devices. This register is only used by the CY8C22x45, CY8C21345, CY8C24x94, CY8C21x34, CY7C64215, CY7C603xx, and CYWUSB6953 PSoC devices.

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,FDh	IDAC_CR0	SplitMux	MuxClkGEL	OSCMDR[1:0]		IRANGE	OSCMDL[1:0]		ENL	RW : 00

Bit 7: SplitMux. This bit allows the analog mux bus to be configured as two separate nets. This bit is only used in the CY8C24x94 and CY7C64215 PSoC devices.

Bit 6: MuxClkGEL. Global enable connection for MUXCLK in the CY8C24x94 and CY7C64215 PSoC devices, or MUXCLK Left in the CY8C22x45 PSoC devices.

Bit 5 and 4: OSCMDR[1:0]. These bits, when set, enable the analog mux bus right to reset to Vss whenever the comparator trip point is reached. These bits are only used in the CY8C22x45 and CY8C21345 PSoC devices.

Bit 3: IRANGE. This bit and Double_Current bit in IDAC_CR1 select the three current ranges that are available for the DAC.

Bits 2 and 1: OSCMDL[1:0]. These bits, when set, enable the analog mux bus left (or analog mux bus right) to reset to Vss whenever the comparator trip point is reached. These bits are only used in the CY8C22x45 and CY8C21345 PSoC devices.

Bit 0: ENL. This bit controls whether or not the left channel IDAC mode is enabled.

For additional information, refer to the IDAC_CR0 register on page 266.

25.4.6 IDAC_CR1 Register

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,DCh	IDAC_CR1	ENR	MuxClkGER	ICEN	IDAC_TRIM				Double_Current	RW : 00

Bit 7: ENR. This bit controls whether or not the right channel IDAC mode is enabled.

Bit 6: MuxClkGER. This bit controls connection of the analog mux bus right clock signal to a global in the CY8C22x45, CY8C21345, CY8C24x94, and CY7C64215 PSoC devices.

Bit 5: ICEN. When ICEN is enabled, both IDAC outputs are enabled.

AGOEL: 0 selects IDACL_D to Left IDAC.
1 selects IDACR_D to Left IDAC

AGOER: 0 selects IDACL_D to Right IDAC.
1 selects IDACR_D to Right IDAC

When ICEN is disabled, both IDAC will be controlled separately by their data registers and output enable signal AGOEL/AGOER.

Bits 4 to 1: IDAC_TRIM. These four bits go to PLL and are used to trim IUNIT32 current output. The default value is 4'b1000 (ideally it is 10uA). Each step will change the current approximately by 3%.

Bit 0: Double_Current. This bit is used for IDAC current range control; it will combine with IDAC_CR0[3].IRANGE to define three different IDAC current ranges.

Double_Current	IDAC_CR0[3].IRANGEL	Current Range
0 (Reserved)	0 (Reserved)	Reserved
1	0	4x
0	1	16x
1	1	32x

For additional information, refer to the IDAC_CR1 register on page 250.

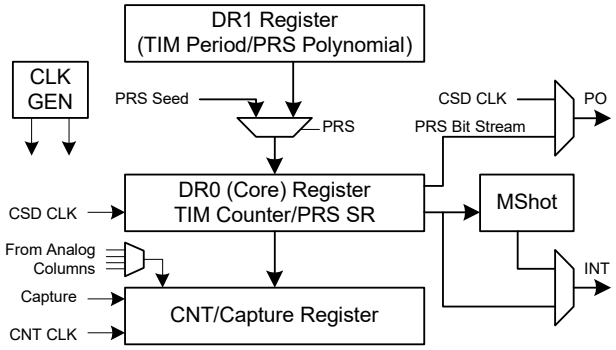
26. CSD Logic System

This chapter covers the configuration and use of the CSD Logic blocks and their associated registers. For a complete table of the CSD Logic registers, refer to Section 26.3. For a quick reference of all CSD Logic registers in address order, refer to the Register Details chapter on page 103.

26.1 Architectural Description

Figure 26-1 shows the main components of CSD Logic blocks are Clock Generator, DR1 register (for timer period register or PRS polynomial register), DR0 core register (for timer's down-counter or PRS shift register), multi-shot control counter, and dual-proposal counter/capture register.

Figure 26-1. CSD Logic Top Level Block Diagram



DR1/DR0/CNT all are 16-bit width to support 16-bit CSD resolution. Multi-shot is 4-bit width, therefore, it is possible to have 1 shot to 15 shots setting.

26.1.1 System Bus Interface

The CSD Logic block Bus Interface is not designed to be connected directly to the M8C system data/address bus interface.

1. Each CSD Logic block is associated with eight bytes registers. Therefore, the highest five address bits are decoded to generate block select (CS_) to enable that block for read/write access. And the lowest three address bits are decoded to select one of eight registers.
2. The drive strength of the CSD Logic block data register tristate drivers is insufficient to drive the system data bus directly. Therefore, a local digital data bus is implemented. This local digital data bus may be interfaced to the system data bus through a transceiver circuit.

26.1.2 Local Data Bus DBDRV signal

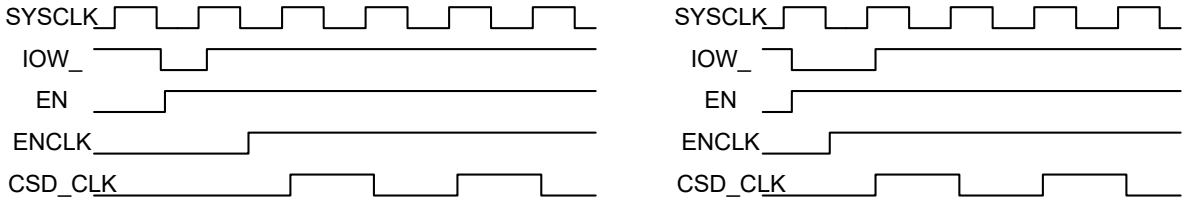
The CPU generates a signal called DBDRV, indicating when the CPU is driving the bus. It is typically asserted high, except during I/O and Memory reads, when it is asserted low. Figure 26-6 shows that the DBDRV brackets the IOR_ signal with setup and hold time of one-half CPU clock.

In the CSD Logic blocks, DBDRV is used as the IOR_ signal, which drives the read data onto the local bus. After one-half CPU clock cycle, the system IOR_ signal is asserted low and enables the connection from the local bus to the system data bus. In this way, the total IOR_ time is extended by one-half CPU cycle.

26.1.3 CSD Clocks Generation

Two clocks are used in CSD Logic block. One is to drive 16-bit timer down-counter/PRS shift register, called CSD_CLK. Another is to drive 16-bit up-counter, called CNT_CLK. CSD_CLK derives from SYSCLK and CNT_CLK derives from CSD_CLK. The dividing coefficient of CSD_CLK could be 1/2/4/8/16/32/128/256. The dividing coefficient of CNT_CLK is 1/2/4/8. To ensure safe timing, a SYSCLK falling edge registered version of 'EN' is used to turn on and/or off CSD_CLK and CNT_CLK. For detailed information, refer to Section 26.3.

Figure 26-2. CSD clock 'EN' timing



26.1.4 16-bit Timer/PRS Function

There are timer and/or PRS function shared by a few of the same registers. Therefore, only one function can be selected at one time. There is compare output from this block. It is 'terminal count' in timer mode, or DR0 equal to seed in PRS mode.

26.1.4.1 Timer Function

In timer mode, DR1 is used to store the period information. When CSD block is in disable mode and you write DR1, it transfers period data into DR0 register. The period data can also be reloaded into DR0 when TC is reached (that is, count to zeros). DR0 counts down at each rising edge of CSD_CLK after you enable CSD.

26.1.4.2 PRS Function

In PRS mode, DR1 is used to store the PRS polynomial instead of the seed to be loaded into DR0. But every time you write to DR1, it transfers a fixed seed (it is fixed when CSD block is instanced) into DR0 register. DR0 runs according to polynomial and LSFR algorithm at each rising edge of CSD_CLK after you enable CSD. The polynomial setting is the same as in CRCPRS function in dpsoc array. Note that there is a limitation in PRS function. PRS polynomial length can only support 8 to 16 bit.

26.1.5 4-bit Multi-shot Function

The multi-shot function is implemented in CSD Logic block. The multi-shot number can be 1 to 15 according to bit[7:4] in CSDxCR0 register. When you set these bits to a non zero value, the timer and/or PRS function run corresponding rounds and then enter into CSD disable mode automatically. But the timer/counter function continuously runs until you write a firmware disable if the multi-shot bits are all 0s. The round 'end' signal is compare signal from timer/PRS function.

26.1.6 16-bit Counter/Capture Function

There are 16-bit dual-proposal register for counter or capture. It is in capture mode if the timer or PRS function is used for non-CSD operation (decided by bit[2] in CSDxCR0). Otherwise it in is counter mode.

26.1.6.1 Counter mode

In counter mode, a 4-to-1 mux is used to select 'GATE' control signal from four analog column outputs. The counter runs on CNT_CLK and only counts up when 'GATE' is high. The counter is reset by either chip reset or if you write 1 to bit[3] of CSDx_CR0.

26.1.6.2 Capture mode

In capture mode, any register reading of CSDx_DR0_L or CSDx_DR0_H always receives 0 on data bus, and transfers all 16 bits data to CSDx_CNT_L and CSDx_CNT_H at the same time. Then the data can be read out from CSDx_CNT_L and CSDx_CNT_H later.

26.1.7 Interrupt and Primary Output

There are two interrupt sources: timer/PRS compare ture and last shot. Bit[7] of CSDx_CR1 is used to select interrupt between them. The primary output is CSD_CLK (in timer mode) or PRS bit stream (in PRS mode). The primary is gated to low if CSD Logic is disabled or CSD Logic is not for CSD operation (that is, for general timer usage).

26.2 Application Description

CSD Logic block is used for CSD operation or as a 16-bit general timer with software capture ability.

26.2.1 CSD Usage

For CSD usage:

1. **SET** ‘CSD’ bit in CSDx_CR0 register
2. Select timer or PRS function
3. Input timer period or PRS polynormial (timer or PRS function **MUST** be chosen first)
4. Select ‘GATE’ signal source from 16-bit counter
5. Select CSD_CLK and CNT_CLK
6. Select interrupt
7. Reset 16-counter: write 1 to ‘CNT_RST’ bit in CSDx-_CR0, and other necessary setting outside
8. **Enable** block with proper multi-shot setting
9. Wait interrupt
10. Read counter data and reset counter and other manipulation
11. Reenable block and go to step 10

Figure 26-3. CSD Configuration with Limited Analog Columns

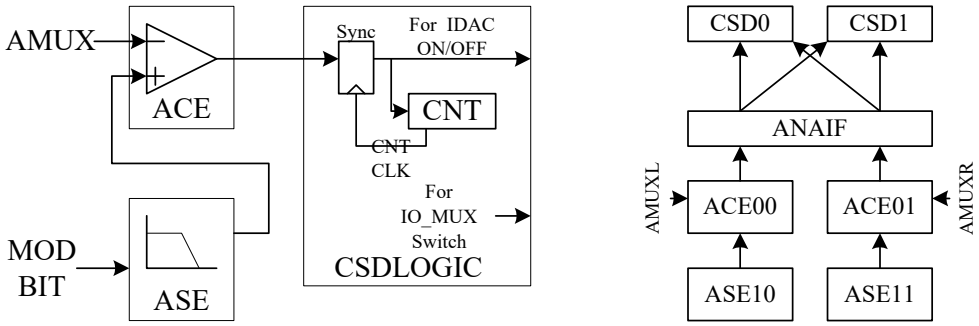


Figure 26-4. CSD Configuration with Compare Analog Columns

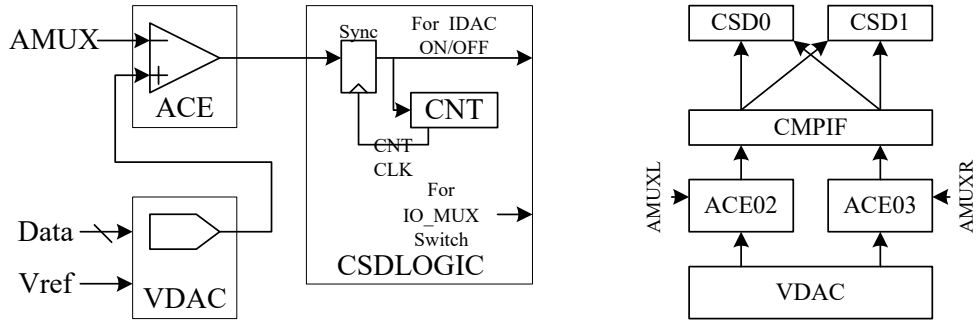
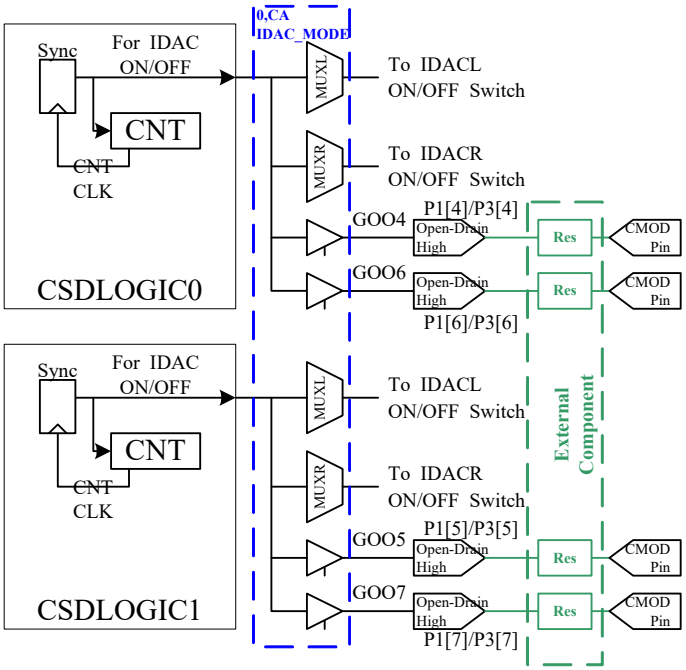


Figure 26-5. The Feedback Component: Internal IDAC or External Resistors



26.2.2 General Timer Usage

For general timer usage:

1. **CLEAR** 'CSD' bit in CSDx_CR0 register
2. Select timer function
3. Input timer period (timer function **MUST** be chosen first)
4. Select CSD_CLK
5. Select interrupt
6. **Enable** block with proper multi-shot setting
7. Wait interrupt

26.3 Register Definitions

26.3.1 CSDx_DR0_L, CSDx_DR0_H

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,50h	CSD0_DR0_L	Data[7:0]								R : 00
0,54h	CSD0_DR0_H	Data[7:0]								R : 00
0,58h	CSD1_DR0_L	Data[7:0]								R : 00
0,5Ch	CSD1_DR0_H	Data[7:0]								R : 00

CSDx_DR0 is 16-bit read-only registers. It is main down-count register in Timer mode, or LFSRs in PRS mode. You read it always as 0s on CPU data bus. If you read any DR0_L or DR0_H, it transfers all 16-bit data to CSDx_CNT registers in non-CSD mode.

26.3.2 CSDx_DR1_L, CSDx_DR1_H

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,51h	CSD0_DR1_L	Data[7:0]								W : 00
0,55h	CSD0_DR1_H	Data[7:0]								W : 00
0,59h	CSD1_DR1_L	Data[7:0]								W : 00
0,5Dh	CSD1_DR1_H	Data[7:0]								W : 00

CSDx_DR1 is 16-bit write-only registers. It is dual purposes registers. It is either polynomial register in PRS mode or period register in Timer mode. If you write to it in Timer mode, it transfers its data to DR0 register if you disable the timer function. In PRS mode, every time polynomial register (DR1) is updated, the DR0 is set by fixed seed value (function is disabled). Always set mode bits first, then set other registers in PRS mode.

26.3.3 CSDx_CNT_L, CSDx_CNT_H

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,52h	CSD0_CNT_L	Data[7:0]								RC : 00
0,56h	CSD0_CNT_H	Data[7:0]								RC : 00
0,5Ah	CSD1_CNT_L	Data[7:0]								RC : 00
0,5Eh	CSD1_CNT_H	Data[7:0]								RC : 00

CSDx_CNT is 16-bit read-only registers for up-count purpose in CSD mode or capture purpose in non-CSD mode. It can be reset.

- The up-counter counts the number of input clock when 'Gate' is open (active).
- The capture register captures the data in DR0 register at the rising edge of 'Load' signal (that is, when you read DR0).

26.3.4 CSDx_CR0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,53h	CSD0_CR0	MSHOT[3:0]				CNT_RST	CSD	PRS	EN	RW : 00
0,5Bh	CSD1_CR0	MSHOT[3:0]				CNT_RST	CSD	PRS	EN	RW : 00

CSDx_CR0 is the configuration register.

M_SHOT[3:0]: multi-shot control bit.
- 4'b0000: CSD_Logic block continuously works until function is disabled. Interrupt occurs periodically if INT_TYPE in CSDx_CR1 is zero.
- Others: After 'En' bit is set, the function starts running. The interrupt can be generated the numbers specified by these bits (if INT_TYPE in CSDx_CR1 is zero). Next, the 'EN' bit is cleared automatically, then function is disabled.

CNT_RST: not a real register bit and is write-only. If you write 0, it has no effect; if you write 1, it resets CSDx_CNT.

CSD: 1 outputs MSB of CSDx_DR0 (PRS mode) or CSD_CLK (Timer mode) to MUXCLK. It should cooperate with 'EN' bits send MSB/CSD_CLK to MUXCLK. The output is zero if 'EN' is zero.

PRS: CSD block works as PRS generator when it is 1. Otherwise it is Timer function.

EN: to enable function running: 1: function enabled. The interrupt is generated when comparison becomes true. Comparison is CSDx_DR0 equal to seed value in PRS mode, or CSDx_DR0 equal to 0s in timer mode. There is no interrupt when 'EN' is zero.

26.3.5 CSDx_CR1

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,57h	CSD0_CR1	INT_TYPE	CSD_CKSEL[2:0]			CNT_CKSEL[1:0]		ACOL[1:0]		RW : 00
0,5Fh	CSD1_CR1	INT_TYPE	CSD_CKSEL[2:0]			CNT_CKSEL[1:0]		ACOL[1:0]		RW : 00

CSDx_CR1 is the configuration register.

INT_TYPE: the interrupt occurs at CSDx_DR0 compare true (TC in timer or DR0 equal to seed in PRS) when INT_TYPE is 0. Otherwise the interrupt occurs at the end of the last shot.

CSD_CKSEL[2:0]: use to set CSD_CLK frequency (similar divider as CPUCLK generator)
- 000: SYSCLK/8
- 001: SYSCLK/4
- 010: SYSCLK/2
- 011: SYSCLK/1
- 100: SYSCLK/16
- 101: SYSCLK/32
- 110: SYSCLK/128
- 111: SYSCLK/256

CNT_CKSEL[1:0]: use to set CNT_CLK frequency
- 00: CSD_CLK/1
- 01: CSD_CLK/2
- 10: CSD_CLK/4
- 11: CSD_CLK/8

ACOL[1:0]: indicates which analog column is for counter 'Gate' signal
- 00: analog column0
- 01: analog column1
- 10: compare column0
- 11: compare column1

Note The dividers for CSD_CLK and CNT_CLK are reset when you disable the function.

26.4 Clocking

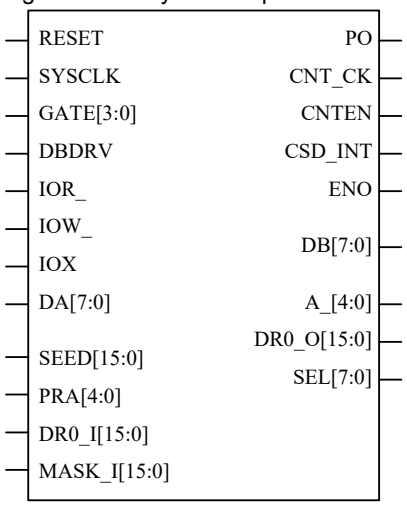
SYSCLK is a primary clock input of CSD Logic Block.

26.5 Reuse Information

Refer to Section 26.8.

26.6 Symbol/Representation

Figure 26-6. Symbol/Representation



26.7 Block Pin Description

Pin Name	Direction	Description
RESET	I	System reset.
SYSCLK	I	System clock (24MHz).
GATE[3:0]	I	The signals come from analog columns.
DBDRV	I	M8C DBDRV.
IOR_	I	M8C IOR_.
IOW_	I	M8C IOW_.
IOX	I	M8C IOX.
DA[7:0]	I	M8C Address Bus.
DB[7:0]	I/O	M8C Data Bus.
PO	O	The primary output to output switch signal for CSD operation.
CNT_CK	O	16-bit counter clock.
CNTEN	O	Count ‘EN’ signal, 16-bit counter counts up when ‘CNTEN’ is 1.
CSD_INT	O	Interrupt.
ENO	O	Buffered version of CSD ‘EN’ signal.
A_[4:0]	O	The inverted version of DA[7:3].
PRA[4:0]	I	Use to select CSD registers location. Refer to Section 26.8.1.
SEED[15:0]	I	The seed in PRS mode. It must have 8 ones and 8 zeros.
DR0_O[15:0]	O	The output of DR0 data.
DR0_I[15:0]	I	Rearranged version of DR0_O[15:0]. Refer to Section 26.8.2.
SEL[7:0]	O	The highest 8 bits mask information according to polynomial.
MASK_I[15:0]	I	Rearranged version of SEL[7:0]. Refer to Section 26.8.2.

26.8 Block Level Interfaces

26.8.1 CSD Logic Register Space Assignment

CSD Logic block occupies eight continuous register spaces in bank 0. The PRA[4:0] input determines the start position. PRA[4:0] are comprised of DA[7:3] and/or A_[4:0] (that is, inverted DA[7:3]). The CSD Logic block register is accessible if PRA[4:0] is all 1s. For example, if the CSD Logic block is at 0,50h, then PRA[4:0] should be {A_[4],DA[6],A_[2],DA[4],A_[0]}.

26.8.2 PRS related Block Level Interface Configuration

26.8.2.1 Seed[15:0]

Seed[15:0] is user-defined, but there must be eight 1s and eight 0s in the value.

26.8.2.2 DR0_O[15:0] and DR0_I[15:0]

DR0_0[15:0] is the output from DR0 core registers. Use the rearranged version DR0_I[15:0] to compare DR0 with seed. The DR0_I[15:8] must be 1s and DR0_I[7:0] must be 0s when DR0 is equal to predefined seed. Therefore, all bits in DR0_O whose corresponding seed bit is 1 must in DR0_I[15:8], and all bits in DR0_O whose corresponding seed bit is 0 must in DR0_I[7:0]. For example, if seed is 16’b1100_0001_1011_0110, then DR0_I[15:8] should be {DR0_O[15:14], DR0_O[8:7], DR0_O[5:4], DR0_O[2:1]}, and DR0_I[7:0] should be {DR0_O[13:9], DR0_O[6], DR0_O[3], DR0_O[0]}.

26.8.2.3 SEL[7:0] and MASK_I[7:0]

This configuration handles seed and DR0 comparison when PRS length is less than 16 bits. If the PRS length is less than 16 bits, the highest redundant (unused) bits are ‘don’t care’ bits in comparison. The output SEL[7:0] represents the ‘don't care’ bit that corresponds to polynomial setting. (It is 1 if the related bit is a ‘don't care’ bit.) However, mask out only the corresponding bits in seed that are 1s, because the corresponding bit in seed are 0s and moves to DRO_I[7:0] and does not have any effect. (This is because the unused bits in DR0 gradually become zeros in PRS running.) Therefore, MASK_I[7:0] should reflect the preceding information. For example, if the polynomial is 12 bit length, then SEL[7:0] outputs 8’b1111_0000. And if the seed is 16’b1100_0001_1011_0110 and you assign {DR0_O[15:14], DR0_O[8:7], DR0_O[5:4], DR0_O[2:1]} to DR0_I[15:8], the MASK_I[7:0] must be {SEL[7:6],SEL[0],5’b0_0000}; that is because you gate (mask) only DR0_O[15:14] and DR0_O[8] by ‘SEL’ signals. Consider rest of the bits in DR0_I[15:8] in comparison.

27. Real Time Clock (RTC)

This chapter covers the configuration and use of the real time clock (RTC) block and its associated registers. For a complete table of the related registers, refer to Section 27.2. For a quick reference of all associated registers in address order, refer to the Register Details chapter on page 103.

27.1 Architectural Description

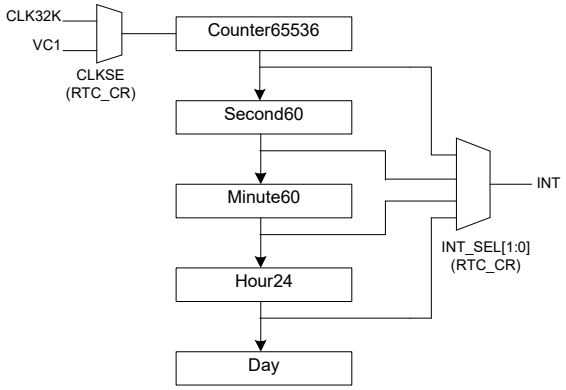
The RTC block supports real time count to count the time with external crystal oscillator. It supports the following features:

- Real time count to count the time with external 32K Crystal oscillator.
- Flexible interrupt sources between second, minute, hour, and day.
- Hour, Minute and Second time read/write in BCD format (easier for external LCD display).
- Normal timer if using VC1 as clock source. For more details about VC1 clock, please refer to Section 27.1.4.
- Sleep mode with internal or external 32K clock source.
- Reset by PPOR, IPOR, and watchdog reset (but cannot be cleared by firmware reset).
- Disable to save power.

27.1.1 BCD Code Counter

There are three counters in the RTC block to count the hour, minute and second in BCD format. The second counter will increase by 1 for every 65536 count clock period. The minute counter will increase by 1 for every 60 seconds. The hour counter will increase by 1 for every 60 minutes, and for every 24 hours there will be an optional day interrupt. As shown in Figure 27-1, for Counter65536, VC1 and CLK32K are two optional clock sources. When VC1 is selected as the clock input, the RTC block can be used as a fixed period timer based on the VC1 period.

Figure 27-1. BCD counters in RTC block



27.1.2 User Data Writing

In RTC block, there are three registers corresponding to the hour, minute and second: RTC_H, RTC_M, and RTC_S. They are all readable and writable. Firmware can write a BCD value for hour, minute and second. Note that writing to any one of these three registers will result in the counter65536 to be reset to all zeros, so that the RTC can start timing function from a good start point. The legal range for writing BCD value for RTC_H is 0 to 23, and for RTC_M and RTC_S is 0 to 59.

27.1.3 RTC Data Reading

To attain the current time of RTC, read registers RTC_H, RTC_M, and RTC_S. There are two methods for reading: sync read and no sync read. Sync read ensures that the HH:MM:SS data be synchronized to the time of RTC_H read operation. When sync read is enabled, the read of RTC_H will latch the current minute and second value into a buffer, therefore, ensure that the MM:SS are synchronized to the time of RTC_H read operation. Also if sync read is disabled, the read value of HH:MM:SS will be aligned to each register's read time independently.

27.1.4 General Timer

As shown in Figure 27-1, there are two clock sources for counter65536. When VC1 is selected as the clock source, the RTC can be used as a general timer based on VC1 clock period. When used as a general timer, the programmed time period can be from *(65536*VC1)* to *(24*60*60)*(65536*VC1)* by setting different start values of RTC_H, RTC_M, and RTC_S.

27.2 Register Definitions

27.2.1 RTC_H

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,A4h	RTCH			HR1[1:0]		HR0[3:0]				R : 00

This register is used to read and write the current hour value in BCD format. Writing to this register will reset count32768 to all zeros. (Will be displayed as “XY”; the legal range is from 00 to 23.)

Bits 5 and 4. X value of hour

Bits 3 to 0. Y value of hour

27.2.2 RTC_M

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,A5h	RTCM		MIN1[2:0]			MIN0[3:0]				R : 00

This register is used to read and write the current minute value in BCD format. Writing to this register will reset count32768 to all zeros. (Will be displayed as “XY”; the legal range is from 00 to 59.)

Bits 6 to 4. X value of minute

Bits 3 to 0. Y value of minute

27.2.3 RTC_S

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,A6h	RTCS		SEC[2:0]			SEC[3:0]				RW : 00

This register is used to read and write the current second value in BCD format. Writing to this register will reset count32768 to all zeros. (Will be displayed as “XY”; the legal range is from 00 to 59.)

Bits 6 to 4. X value of second

Bits 3 to 0. Y value of second

27.2.4 RTC_CR

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,A7h	RTCCR	TREG[1:0]		INT_EN	CLKSE	INT_SEL[1:0]		SYNC_RD	RT_EN	RW : 00

Bits 7 and 6: TREG[1:0]. Only TMOD accessible (for DFT purpose only).

00b: Normal mode

01b: Bypass the Count Registers from 8th to 14th

10b: Bypass the Count Registers from 1st to 7th

11b: Bypass the Count Registers from 1st to 14th, and short the second carry to minute and hour input

Bit 5: INT_EN. RTC interrupt enable. 1 to enable and 0 to disable.

Bit 4: CLKSE. RTC module clock source selection.

0: CLK32K

1: VC1

Bits 3 and 2: INT_SEL[1:0]. RTC interrupt source selection.

00b: Sec

01: Min

10: Hour

11: Day

Bit 1: SYNC_RD. 1 to enable sync read. 0 to disable sync read.

Bit 0: RT_EN. RTC module enable control. 1 to enable and 0 to disable.

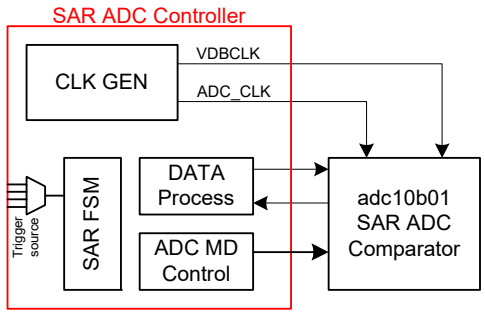
28. 10-Bit SAR ADC Controller

This chapter covers the configuration and use of the 10-bit SAR ADC controller and its associated registers. For a complete table of the 10-bit SAR ADC controller registers, refer to Section 28.3. For a quick reference of all 10-bit SAR ADC controller registers in address order, refer to the Register Details chapter on page 103.

28.1 Architectural Description

Figure 28-1 shows that the main components of SAR ADC controller are: Clock Generator, SAR FSM, DATA Process, and ADC Mode Control. SAR ADC controller must work with IP:adc10b01 (referred to as ADC comparator in the following sections) to complete the whole SAR ADC function.

Figure 28-1. SAR ADC Controller Top Level Block Diagram



28.1.1 System Bus Interface and Local Data Bus DBDRV signal

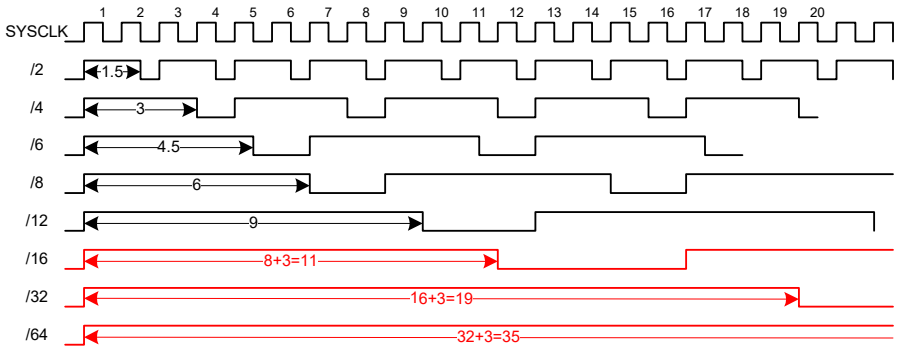
The SAR ADC Controller Bus Interface is not designed to be connected directly to the M8C system data/address bus interface. It uses the similar mechanism as digital blocks.

28.1.2 ADC Clock Generation

ADC clock is the clock to ADC comparator and derivates from SYSCLK. It can be SYSCLK/2, /4, /6, /8, /12, /16, /32 or /64 (total of eight selections). The ADC clock is targeted to provide MAX 3MHz clock to ADC comparator. A special arrangement of 75-25 duty is used because ADC comparator needs more time to settle its internal VDAC output signal at high speed range. But the 75-25 clock duty is not always true to every possible frequency. In low frequency end, it is approximately 52% duty. Figure 28-2 shows the detailed timing diagram. Refer to Section 28.3 for the register definitions.

There is no ADC clock in IDLE state even if ADC is enabled.

Figure 28-2. ADC Clock Dividing



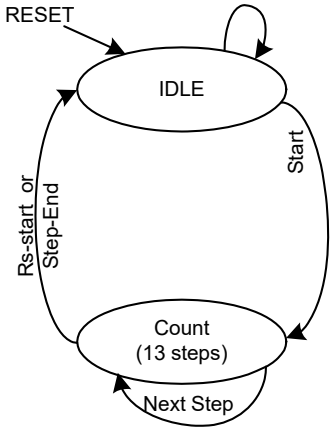
28.1.3 Voltage Doubler Clock Generation

There is a voltage doubler in ADC comparator. Enable it when chip power is less than 3 V. Two options are to directly use SYSCLK, or SYSCLK/4. Refer to Section 28.3 for register definitions.

28.1.4 ADC FSM

SAR ADC needs 12 ADC clocks for one conversion. The first two clocks are for sampling analog input signal, and 10 clocks are for data conversion. But one chosen ADC clock is stretched to two times long compared to other clocks in conversion. (Refer to Section 28.1.6 for details.) Therefore, a conversion has 13 clocks on ADC FSM perspective. From this, at least an extra SYSCLK is needed for IDLE state. Every conversion must start from IDLE state and return to IDLE.

Figure 28-3. ADC FSM



There are three modes to run A-D-C; that is, there are three modes to trigger 'START' signal. The first one is SW trigger mode. Every time you write 1 to 'START' bit in ADC_CR0 register (ADC should be enabled), it triggers a new conversion. The incomplete conversion (if there is one) is interrupted and the new conversion is started immediately after. The state machine returns to IDEL after conversion completes. The second one is free-run mode. The conversions run repeatedly until you disable the ADC controller. But the SW trigger is still available and a new conversion is started if a SW trigger is received. The third mode is HW trigger mode, also called auto-trigger mode or auto-align mode. Select one of four hardware trigger source and use to trigger 'START'. It acts similar to SW trigger mode but trigger source is changed. Refer to Section 28.3 for detailed register definitions.

Table 28-1. ADC Running Mode

Trigger Mode	'FREERUN' Bit	'ALIGN_EN' Bit
SW-Trigger	0	0
FREERUN	1	0
HW-Trigger	*	1

28.1.5 SAR Algorithm and Data Process

In IDLE mode, the ADC data keeps 0. It starts data conversion from MSB to LSB in each ADC clock after sample stage. It uses binary search algorithm to find out the digital data most close to original input. It needs 10 times searching because it is 10-bit ADC. But the format of the data goes to ADC comparator (referred to as D[13:0] in this chapter) and the data stored in ADC controller (referred to as ADC_D[9:0] in this chapter) are different. First, the bits' active state of D[13:0] is 0 and ADC_D[9:0] is 1. Second, ADC_D[9:0] are binary code; D[6:0] are also binary code but D[13:7] are thermal code. A binary-to-thermal code conversion is needed between ADC_D[9:7] and D[13:7]. The thermal code conversion table follows.

Table 28-2. ADC Thermal Code Conversion Table

ADC_D[9:7]	D[13:7]
000	111_1111
001	011_1111
010	001_1111
011	000_1111
100	000_0111
101	000_0011
110	000_0001
111	000_0000
D[6:0] = ~{ADC_D[6:0]}	

28.1.6 A-D-C Operation Mode

ADC comparator requires four A-D-C operation modes to achieve the best performance. Each mode has different control signal timing, as listed in the following figures. Refer to Section 28.3 for detailed register definitions.

Figure 28-4. A-D-C Operation Mode 0 (default Mode)

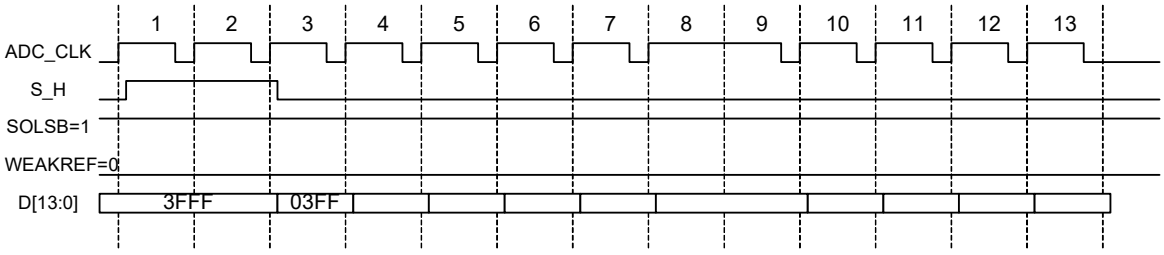


Figure 28-5. A-D-C Operation Mode 1

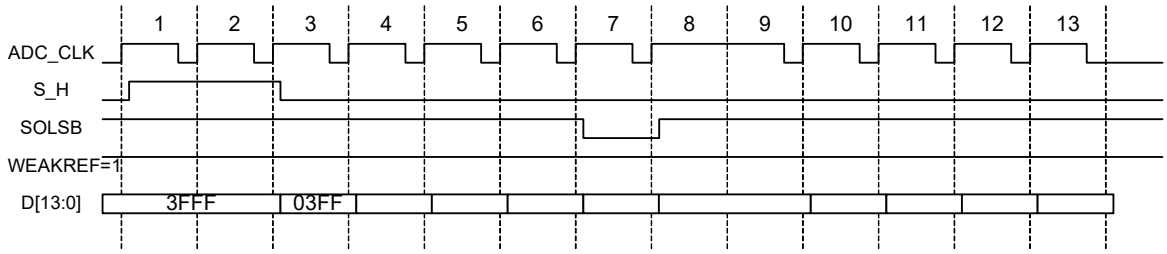


Figure 28-6. A-D-C Operation Mode 2

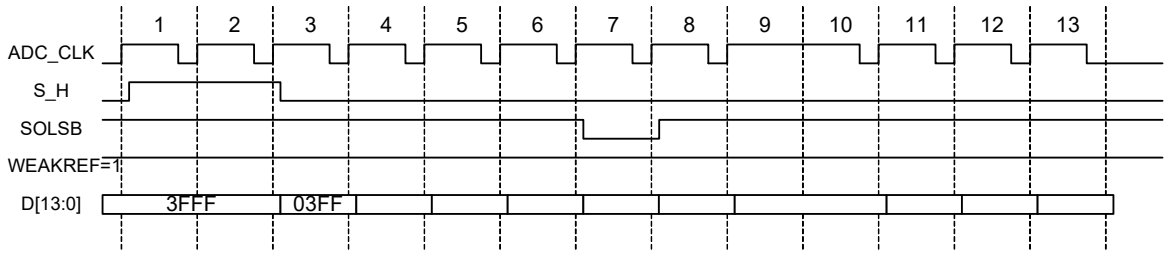
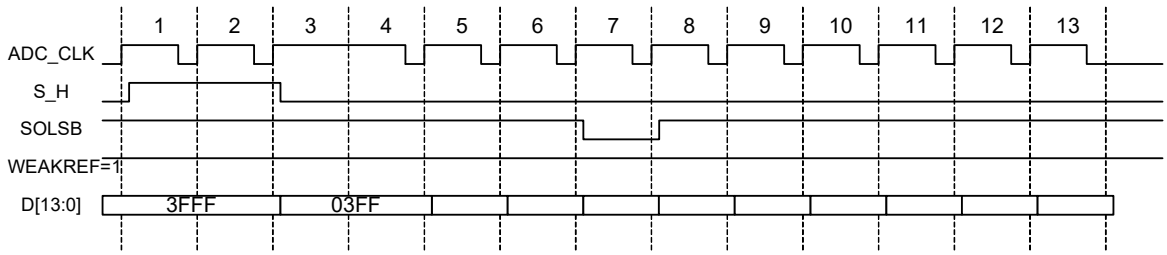


Figure 28-7. A-D-C Operation Mode 3



28.1.7 'Ready' Bit, 'Ongoing' Bit and Interrupt

In ADC_CR0, two status bits are used to reflect ADC status. First is 'READY' bit, which is set when the ADC data register receives new data. It can only be cleared when you read ADC_DH register unless other new data arrives. Second is 'Ongoing' bit which directly comes from the inverted 'IDLE' state.

The ADC interrupt occurs when there is new data.

28.2 Application Description

28.2.1 ADC Sample Rate and Clock Selection

The ADC sample rates are MAX > 150 Ksps and MIN > 20 Ksps, based on the following table.

Table 28-3. ADC Sample Rate and Clock Selection

SYSCLK (IMO)	Fastest		Slowest	
	Clock Setting	Actual SPS	Clock Setting	Actual SPS
24 MHz	SYSCLK/12	152.8KSPS	SYSCLK/64	28.8KSPS

28.2.2 Enable Voltage Doubler

Enable voltage doubler when Vcc is less than 3.0 V. Note that this voltage doubler enable is not gated off by ADC enable. To achieve a low power state, turn it OFF.

28.3 Register Definitions

28.3.1 ADC_DH

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,6Ah	ADC_DH	ADC_DH[7:0]								R : 00

ADC_DH is the highest eight bits of ADC data. In normal mode, it is read-only byte. In ADC test mode, it is a readable/writeable byte. The ADC can be treated as an 8-bit ADC if you only read this byte as ADC data.

Bits 7 to 0: ADC_DH. The highest eight bits of ADC data. In normal mode, it is read-only byte. In ADC test mode, it is readable/writeable byte. The ADC is treated as a 8-bit ADC if only read as ADC data.

28.3.2 ADC_DL

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,6Bh	ADC_DL							ADC_DL[1:0]		R : 00

ADC_DL is the lowest two bits of ADC data. In normal mode, it is read-only byte. In ADC test mode, it is readable/writeable byte.

Bits 1 and 0: ADC_DL. The lowest two bits of ADC data. In normal mode, it is read-only byte. In ADC test mode, it is readable/writeable byte.

28.3.3 ADC_CR0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,A8h	ADC_CR0	ADC_TST1	ADC_CHS[3:0]				READY	START/ONGOING	ADC_EN	RW : 00

Bit 7: ADC_TST1. The ADC data registers are write-only in chip-wide TEST mode when this bit is set. This bit stays in reset state if chip is not in TEST mode. Refer to Section 28.6.1 for usage.

Bits 6 to 3: ADC_CHS[3:0]. ADC input channel selection.

- 0000 ~ 0111: P0[0] ~ P0[7]
- 1000 ~ 1011: reserved
- 1011 ~ 1101: VBG, AMUXL, AMUXR
- 1110 ~ 1111: VDAC0 (VDAC0 voltage is the same as VDAC1 if their VDAC setting is the same), V2bg

Bit 2: READY. 1 shows that there is new data that has not been read.

Bit 1: START/ONGOING. If you read 1, the A-D conversion started and is not finished. Write 1 to it in SW trigger mode and it triggers a new conversion.

Bit 0: ADC_EN. ADC enable bit.

28.3.4 ADC_CR1

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,A9h	ADC_CR1	CVTMD[1:0]		TIGSEL[1:0]		CLKSEL[2:0]			ALIGN_EN	RW : 00

Bits 7 and 6: CVTMD[1:0]. The conversion mode:
- 2'b00: the default mode that only the extra cycle for 6th bit conversion. Refer to Figure 28-4.
- 2’b01: the extra cycle for 6th bit conversion with add-on weak Vref buffer. Refer to Figure 28-5.
- 2’b10: the extra cycle for 7th bit conversion with add-on weak Vref buffer. Refer to Figure 28-6.
- 2’b11: the extra cycle for 1st bit conversion with add-on weak Vref buffer. Refer to Figure 28-7.

Bits 5 and 4: TIGSEL[1:0]. auto-trigger source selection. It must work with ALIGH_EN. Refer to CHIPMISC 0, CD/0, CE/0, CF register definitions for these trigger signals.
- 00B: TG_L
- 01B: TG_H
- 10B: TG_16BIT
- 11B: TG_INCMP

Bits 3 to 1: CLKSEL[3:0]. ADC Clock Selection

0: SYSCLK/2	1: SYSCLK/4	2: SYSCLK/6	3: SYSCLK/8
4: SYSCLK/12	5: SYSCLK/16	6: SYSCLK/32	7: SYSCLK/64

Bit 0: ALIGN_EN. 1 to enable auto-align function. The A-D-C is driven by outside-block trigger signal.

28.3.5 ADC_CR2

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,AAh	ADC_CR2	REFSEL	BUFEN	VDBEN	VDB_CLKSEL	FREERUN	ADC_EXT_HALFVDD	ADC_MODE[1:0]		RW : 00

Bit 7: REFSEL. 1 to select external Vref other than Vdd.

Bit 6: BUFEN. 1 to enable Vref buffer. Otherwise bypasses Vref buffer.

Bit 5: VDBEN. 1 to enable voltage doubler in ADC comparator.

Bit 4: VDB_CLKSEL. 1 to select SYSLCK as VDB clock. Otherwise select SYSCLK/4 as VDB clock.

Bit 3: FREERUN. 1 is ADC in FREERUN mode if ADC is not in auto-align mode.

Bit 2: ADC_EXT_HALFVDD. 1 to select VDD/2 from external pad. Otherwise Vdd/2 comes from internal source. It is only writeable in chip-wide TEST mode. Otherwise it stays in default (0) state.

Bits 1 to 0: ADC_MODE[1:0]. ADC comparator work mode, and only writeable in chip-wide TEST mode. Otherwise it stays in default (00B) mode.
- 00B: normal mode, 'Test_out' is Hi-Z output.
- 01B: test mode, 'Test_out' is comparator output (digital signal).
- 10B: test mode, 'Test_out' is internal Vref (analog signal).
- 11B: test mode, 'Test_out' is internal VDD/2 (analog signal).

28.3.6 SARADC_CR3TRIM

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,ABh	ADC_CR3TRIM						ADC_TRIM0[2:0]			RW : 04

Bits 2 to 0: ADC_TRIM0[2:0]. Sent to ADC comparator block directly. The reset value is 3’b100.

28.3.7 SARADC_CR4

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
1,ACh	ADC_CR4		D[13]	D[11]	D[9]	VDBCLK	ADCCLK	ADC_TST2	ADC_CMP	RW : 02

Bits 6 to 4: D[13]/D[11]/D[9]. Read-only registers. In chip-wide TEST mode, you read them as the value on D[13]/D[11]/D[9] which directly go to ADC comparator for data input. In non TEST mode you read them always as 0s. They are for pure DFT purpose.

Bits 3 and 2: VDBCLK/ADCCLK. Read-only register. In chip-wide TEST mode, you read them as the value on voltage doubler clock and ADC comparator clock. In non TEST mode, you read them always as 0s. These two bits are for pure DFT purpose.

Bit 1: ADC_TST2. Set to 1 only in chip-wide TEST mode.

Bit 0: ADC_CMP. Read-only bit and always 0 when ADC is in normal mode. In ADC_TST2 mode, you read it as the data from ADC comparator output. When ADC_TST2 is set to 1, the data written to it replaces the role of ADC comparator output data; that is, it becomes pseudo compare output.

28.4 PSoC Device Distinctions

10-bit SAR ADC controller only exists in CY8C22345, CY8C22545 and CY8C21345.

28.5 Clocking

SYSCLK is a primary clock input of SAR ADC controller block.

28.6 Test Modes

28.6.1 Test Mode in ADC Comparator

Table 28-4. ADC Comparator Test Mode

No.	Ext Vdd/2	Buffer Vref	Ext Vref	MUX_SEL[2:0]	TEST OUT	ADC_CLK	D[13:0]
1	Yes	No	No	001	CMP	Need	Need
2	Yes	Yes	Yes	001	CMP	Need	Need
3 *	No	No	No	001	CMP	Need	Need
4	No	Yes	Yes	001	CMP	Need	Need
5	No	No	No	010	Vdd/2	Do not need	Do not need
6	No	No	No	100	Vref_int	Do not need	Do not need

* Similar to normal mode except for MUX_SEL bits setting. Use ADC_TST1: to set proper data to ADC comparator.

- MUX_SEL[2:0] is 000 in normal mode, which makes ADC comparator “TEST OUT” tri-stated.
- TEST OUT goes to P2[1] directly. Therefore, you can monitor ADC state through either digital signal or analog signal. (Vref_int and Vdd/2 are analog signals.)
- When D[13:0] are needed, that means the proper data should be presented on ADC comparator data input with proper timing with ADC clock.

28.6.2 Test Mode Bits in ADC Controller

Six register bits are used for test mode. All six register bits can only change their values in chip-wide TEST mode, and is automatically cleared when chip leaves TEST mode. Therefore, you always read them as 0s in normal mode.

ADC_TST1: when you set to 1 in TEST mode, the ADC_DH and ADC_DL are writeable. Also, ADC_DH/ADC_DL can go to D[13:0] when ADC_DH is written. Therefore, write ADC_DL first. In ADC_TST1 mode, D[13:0] is only controlled when you write ADC_DH. Otherwise it keeps its previous data.

ADC_TST2 & ADC_CMP: when you set ADC_TST2 to 1 in TEST mode, the ADC_DH/ADC_DL data register is updated every rising edge of SYSCLK. Also, Section 28.3.7 shows that the ADC comparator out is replaced by ADC_CMP. ADC_CMP is writeable in ADC_TST2 mode. It is also readable. You read it as the output value of ADC comparator.

ADC_EXT_HALFVDD: when you set to 1, this forces external pad input (P2[3]) to ADC comparator to replace its internal VDD/2.

ADC_MD[1:0]: decoded to generate MUX_SEL[2:0] on ADC comparator interface.

28.6.3 Test for ADC Controller

Use ADC_TST2 to test ADC control logic. In this case, control the compare input through ADC_CMP and monitor ADC state in every cycle through a ADC_DH/ADC_DL register read.

28.6.4 ADC Controller Configuration for ADC Comparator Test

Table 28-4 shows there are two types of requirement needed by ADC comparator. It either needs ADC clock and D[13:0] presented, or does not need ADC clock and D[13:0] presented. Retrieve all other settings directly from the register setting.

28.6.4.1 Run ADC clock and set D[13:0]

As the following steps show, first set the D[13:0] and then let ADC clock run one conversion. You can run the ADC comparator test 1 to 4.

1. Go into Test Mode and set ADC_TST1 mode
2. Set related register bits for ADC comparator test mode
3. Set ADC clock speed
4. Enable ADC in SW trigger mode
5. Write data into ADC_DL and ADC_DH register (following the sequence)
6. Trigger ADC once (write ‘START’ bit) and monitor ADC ‘TEST OUT’ from P2[1]
7. Repeat steps 5 through 7

28.6.4.2 Do not run ADC clock and keep D[13:0] stable

The following steps are similar to the steps in the previous section, but do not include the trigger and write data steps. Use it to run the ADC comparator test 5 and 6.

1. Go into Test Mode and set ADC_TST1 mode
2. Set related register bit for ADC comparator testing
3. Enable ADC in SW trigger mode (to also enable ADC comparator)
4. Monitor ADC ‘TEST OUT’ from P2[1]

29. MISC Logic

This chapter covers the configuration and use of the glue and/or hook logic and the associated registers. For a complete table of the related registers, refer to Section 29.2. For a quick reference of all associated registers in address order, refer to the Register Details chapter on page 103.

29.1 Architectural Description

The MISC block provides mixed control on:

- Vref PWM signal generation
- IDAC switches on/off
- Dig-Blocks PWM selection
- Auto-Trig signal options

29.1.1 System Bus Interface and Local Data Bus DBDRV signal

The MISC block Bus Interface is not designed to be connected directly to the M8C system data/address bus interface. It uses a similar mechanism as digital blocks.

29.1.2 Vref PWM Signal Generator

The PWM generator in MISC block runs at a fixed clock rate of SYSCLK/8 (thus 3MHz when SYSCLK is 24MHz) and can generate two independent PWM waveforms with the high signal duty varying from 0 to 15/16. The generated PWM waveform can be fed to switch capacitor block type E for reference voltage generation. Switch capacitor type E, when configured as a RC filter, can generate the wanted voltage based on the input PWM waveform. And the generated voltage reference can be computed as:

Vref **=** *%High***RefHi* **+** *%Low***RefLow*

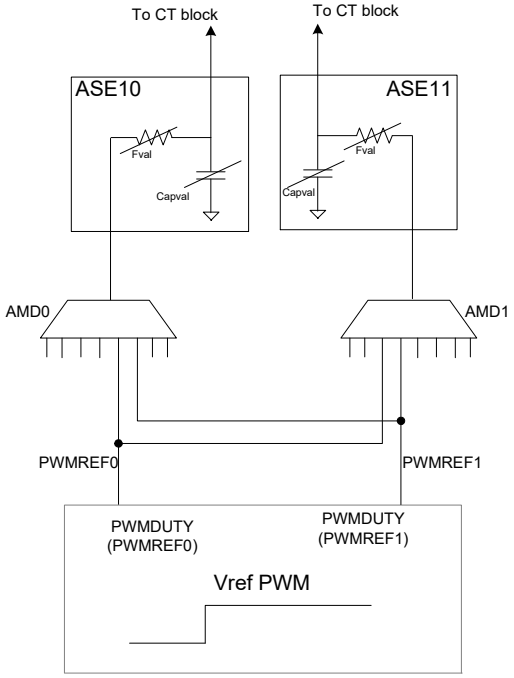
Whereas:

%High is the percentage that the PWM signal is high.

%Low is the percentage that the PWM signal is low.

When you select the generated PWM waveform as the ASE block input, bit AMDx and bits in register AMD_CR0 and AMD_CR1 must be set to select it. Different duty cycle of these two PWM signals can be achieved by configuring bits PWMDUTY in register PWMVREF0 and PWMVREF1.

Figure 29-1. Vref PWM signal generation



29.1.3 IDAC ON/OFF Control

The dual-channel IDAC can produce two configurable currents for capacitive sensing purpose, and these two current sources are connected to analog mux bus left and analog mux bus right independently. As shown in Figure 29-2, these two currents are first connected to analog mux bus via switches, and then connected to capacitance. The switch between the current source and the analog mux bus can be switched on/off in four different modes, and the on/off control option of the switch is determined in register IDAC_MODE. For more details about IDAC, refer to the section.

Figure 29-2. IDAC and analog mux bus connection

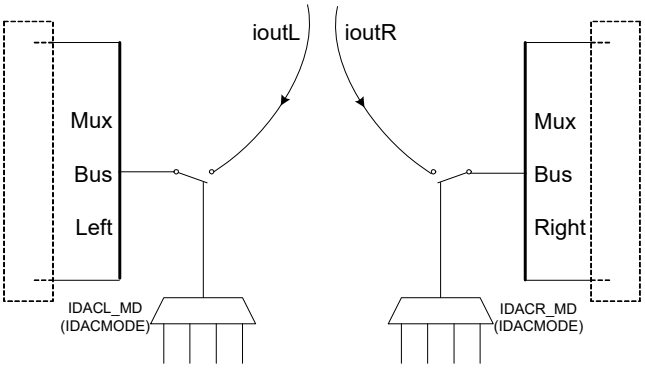


Table 29-1. IDAC ON/OFF Mode

Mode bit	ON/OFF
00	Always ON
01	Follows CSD_CNTEN from CSDLOGIC block
10	Follows CSD_EN from CSDLOGIC block
11	Follows another channel CSD_CNTEN from CSDLOGIC.

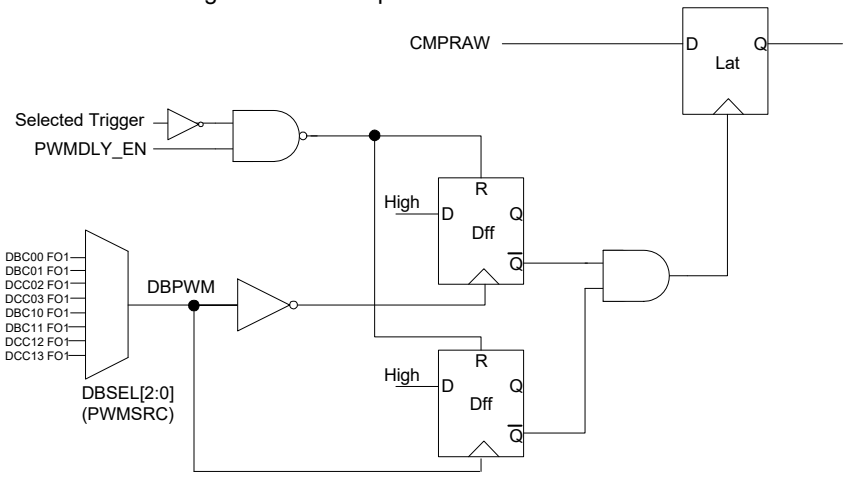
29.1.4 Dig-Blocks PWM Selection

You can select one of the 8 digital blocks' primary output as the PWM input when compare column Timed Gate feature is enabled.

For compare column, when Timed Gate function is enabled, any positive or negative transition on the selected PWM output (DBPWM) will disable the latch for comparator bus output. Thus the comparator bus value used by the following data path will retain the previous value until a new trigger event arrives. And the latch will not be enabled until a trigger event occurs, as shown in. For more details about PWM delay function, refer to the section.

To select a given digital block's primary output as PWM output, set the DBSEL bits in register PWMSRC.

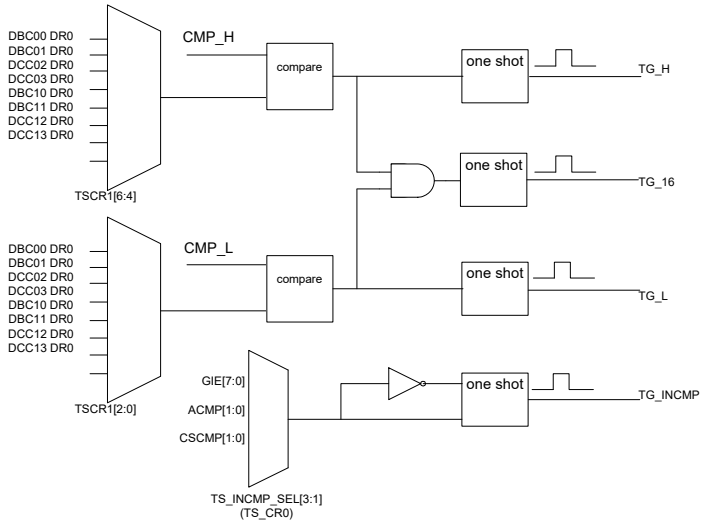
Figure 29-3. Compare Column Timed Gate Function



29.1.5 Auto-Trig Selection

The PSoC device allows a flexible trigger signal selection among a variety of sources for SARADC and Compare Column function. These optional trigger sources include signals directly coming from GIE and analog column output, as well as four compare outputs resulting from configured function digital blocks. See Figure 29-4. For more details about the trigger feature of SARADC and Compare Column Interface, refer to the appropriate sections in the 10-Bit SAR ADC Controller chapter on page 439 and the Two Column Analog Compare System chapter on page 358, respectively.

Figure 29-4. Auto-Trig Signal Generation



For SARADC, when working in Auto-Trigger mode, a new conversion will start when a one-shot pulse occurs on the selected trigger source. The trigger sources that are available for SARADC are: TG_L, TG_H, TG_16, and TG_INCMP. For more details about SARADC trigger signal settings, refer to register ADC_CR1.

For Compare Column, when working in Timed Gate mode, the comparator output latch will be enabled when a transition occurs on the selected trigger source. And the trigger sources that are available for Compare Column are: TG_L, TG_H, and TG_16. For more details about compare column trigger signal settings, refer to register CMPPWMCR.

29.2 Register Definitions

29.2.1 PWMVREF0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,C8h	PWMVREF0	EN	MODBIT0	SADC0	ADC_SOLSB	PWMDUTY0				R : 00

Bit 7: EN. ASE10 PWMVref output enable control. '1' will enable PWMVref output, otherwise it will be 0.

Bit 6: MODBIT0. Read only in test mode. Always returns 0 when read in normal mode. For DFT purpose only.

Bit 5: SADC[0]. Read only in test mode. Always returns 0 when read in normal mode. For DFT purpose only.

Bit 4: ADC_SOLSB. Read only in test mode. Always returns 0 when read in normal mode. For DFT purpose only.

Bits 3 to 0: PWMDUTY0[3:0]. Sets the duty cycle of analog column 0 PWM input waveform, from minimum 0 to maximum 15/16.

29.2.2 PWMVREF1

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,C9h	PWMVREF1	EN	MODBIT1	SADC1	ADC_S_H	PWMDUTY1				R : 00

Bit 7: EN. ASE11 PWMVref output enable control. '1' will enable PWMVref output, otherwise it will be 0.

Bit 6: MODBIT1. Read only in test mode. Always returns 0 when read in normal mode. For DFT purpose only.

Bit 5: SADC[1]. Read only in test mode. Always returns 0 when read in normal mode. For DFT purpose only.

Bit 4: ADC_S_H. Read only in test mode. Always returns 0 when read in normal mode. For DFT purpose only.

Bits 3 to 0: PWMDUTY1[3:0]. Sets the duty cycle of analog column 1 PWM input waveform, from minimum 0 to maximum 15/16.

29.2.3 IDACMODE

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,CAh	IDAC_MODE	CSD1_GO_EN[1:0]		CSD0_GO_EN[1:0]		IDACR_MD		IDACL_MD		RW : 00

Bit 7: CSD1_GO_EN[1]. '1' will enable CSD_CNTEN[1] to GOO[7].

Bit 6: CSD1_GO_EN[0]. '1' will enable CSD_CNTEN[1] to GOO[5].

Bit 5: CSD0_GO_EN[1]. '1' will enable CSD_CNTEN[0] to GOO[6].

Bit 4: CSD0_GO_EN[0]. '1' will enable CSD_CNTEN[0] to GOO[4].

Bits 3 and 2: IDACR_MD[1:0]

00: right channel IDAC on/off switch is always on.

01: right channel IDAC on/off switch will follow CSD_CNTEN in current CSD block.

10: right channel IDAC on/off switch will follow CSD_EN in current CSD block.

11: right channel IDAC on/off switch will follow CSD_CNTEN in another CSD block.

Bits 1 and 0: IDACL_MD[1:0]

00: left channel IDAC on/off switch is always on.

01: left channel IDAC on/off switch will follow CSD_CNTEN in current CSD block.

10: left channel IDAC on/off switch will follow CSD_EN in current CSD block.

11: left channel IDAC on/off switch will follow CSD_CNTEN in another CSD block.

29.2.4 PWMSRC

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,CBh	PWMSRC	TSYNC	SAMP2	ADC_MUX_SEL[2:0]			DBSEL[2:0]			RW : 00

Bit 7: TSYNC. Only TMOD accessible. '1' will use test clock as external clock.

Bit 6: SAMP2. Read only in test mode. Always returns 0 when read in normal mode. For DFT purpose only.

Bits 5 to 3: ADC_MUX_SEL[2:0]. Read only in test mode. Always returns 0 when read in normal mode. For DFT purpose only.

Bits 2 to 0: DBSEL[2:0]. Select which digital block's primary output will be used as PWM input in compare column PWM delay function.

29.2.5 TSCR0

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,CCh	TS_CR0	TS_INCMP_SEL				INCMP_INV	INCMP_EN	CMPH_EN	CMPL_EN	RW : 00

Bits 7 to 4: TS_INCMP_SEL[3:0]. These bits are used to select external or internal source for trigger source.

0h: GIE[0]

1h: GIE[1]

2h: GIE[2]

3h: GIE[3]

4h: GIE[4]

5h: GIE[5]

6h: GIE[6]

7h: GIE[7]

8h: ACMP[0]

9h: ACMP[1]

Ah: CSCMP[0]

Bh: CSCMP[1]

CH to FH: Reserved

Bit 3: INCMP_INV. Inverted version of INCMP will be used when 1.

Bit 2: INCMP_EN. INCMP trigger source enable control; '1' to enable.

Bit 1: CMPH_EN. Enable high channel trigger source when 1.

Bit 0: CMPL_EN. Enable low channel trigger source when 1.

29.2.6 TSCMPH

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,CDh	TS_CMPH	TS_CMPH								RW : 00

This register is used to set the compare value of high channel.

TS_CMPH. The compare value of high channel.

29.2.7 TSCMPL

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,CEh	TS_CMPL	TS_CMPL								RW : 00

This register is used to set the compare value of low channel.

TS_CMPL. The compare value of low channel.

29.2.8 TSCR1

Address	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Access
0,CFh	TS_CR1		TS_CMPH_SEL				TS_CMPL_SEL			RW : 00

This register is used to select the digital block for high channel and low channel comparison.

Bits 6 to 4: TS_CMPH_SEL[2:0]. Used to select which digital block will be used for high channel comparison.

0h: DBC00

1h: DBC01

2h: DCC02

3h: DCC03

4h: DBC10

5h: DBC11

6h: DCC12

7h: DCC13

Bits 2 to 0: TS_CMPL_SEL[2:0]. Used to select which digital block will be used for low channel comparison.

0h: DBC00

1h: DBC01

2h: DCC02

3h: DCC03

4h: DBC10

5h: DBC11

6h: DCC12

7h: DCC13

Section G: Glossary

The Glossary section explains the terminology used in this technical reference manual. Glossary terms are characterized in ***bold, italic font*** throughout the text of this manual.

A

accumulator	In a CPU, a register in which intermediate results are stored. Without an accumulator, it would be necessary to write the result of each calculation (addition, subtraction, shift, and so on.) to main memory and read them back. Access to main memory is slower than access to the accumulator, which usually has direct paths to and from the arithmetic and logic unit (ALU).
active high	1. A logic signal having its asserted state as the logic 1 state. 2. A logic signal having the logic 1 state as the higher voltage of the two states.
active low	1. A logic signal having its asserted state as the logic 0 state. 2. A logic signal having its logic 1 state as the lower voltage of the two states: inverted logic.
address	The label or number identifying the memory location (RAM, ROM, or register) where a unit of information is stored.
algorithm	A procedure for solving a mathematical problem in a finite number of steps that frequently involve repetition of an operation.
ambient temperature	The temperature of the air in a designated area, particularly the area surrounding the PSoC device.
analog	See ***analog signals***.
analog blocks	The basic programmable opamp circuits. These are SC (switched capacitor) and CT (continuous time) blocks. These blocks can be interconnected to provide ADCs, DACs, multi-pole filters, gain stages, and much more.
analog output	An output that is capable of driving any voltage between the supply rails, instead of just a logic 1 or logic 0.
analog signals	A signal represented in a continuous form with respect to continuous times, as contrasted with a digital signal represented in a discrete (discontinuous) form in a sequence of time.
analog-to-digital (ADC)	A device that changes an analog signal to a digital signal of corresponding magnitude. Typically, an ADC converts a voltage to a digital number. The ***digital-to-analog (DAC)*** converter performs the reverse operation.

AND See ***Boolean Algebra***.

API (Application Programming Interface) A series of software routines that comprise an interface between a computer application and lower-level services and functions (for example, user modules and libraries). APIs serve as building blocks for programmers that create software applications.

array An array, also known as a vector or list, is one of the simplest data structures in computer programming. Arrays hold a fixed number of equally-sized data elements, generally of the same data type. Individual elements are accessed by index using a consecutive range of integers, as opposed to an associative array. Most high level programming languages have arrays as a built-in data type. Some arrays are multi-dimensional, meaning they are indexed by a fixed number of integers; for example, by a group of two integers. One- and two-dimensional arrays are the most common. Also, an array can be a group of capacitors or resistors connected in some common form.

assembly A symbolic representation of the machine language of a specific processor. Assembly language is converted to machine code by an assembler. Usually, each line of assembly code produces one machine instruction, though the use of macros is common. Assembly languages are considered low level languages; where as C is considered a high level language.

asynchronous A signal whose data is acknowledged or acted upon immediately, irrespective of any clock signal.

attenuation The decrease in intensity of a signal as a result of absorption of energy and of scattering out of the path to the detector, but not including the reduction due to geometric spreading. Attenuation is usually expressed in dB.

B

bandgap reference A stable voltage reference design that matches the positive temperature coefficient of V_T with the negative temperature coefficient of V_{BE}, to produce a zero temperature coefficient (ideally) reference.

bandwidth
1. The frequency range of a message or information processing system measured in hertz.
2. The width of the spectral region over which an amplifier (or absorber) has substantial gain (or loss); it is sometimes represented more specifically as, for example, full width at half maximum.

bias
1. A systematic deviation of a value from a reference value.
2. The amount by which the average of a set of values departs from a reference value.
3. The electrical, mechanical, magnetic, or other force (field) applied to a device to establish a reference level to operate the device.

bias current The constant low level DC current that is used to produce a stable operation in amplifiers. This current can sometimes be changed to alter the bandwidth of an amplifier.

binary The name for the base 2 numbering system. The most common numbering system is the base 10 numbering system. The base of a numbering system indicates the number of values that may exist for a particular positioning within a number for that system. For example, in base 2, binary, each position may have one of two values (0 or 1). In the base 10, decimal, numbering system, each position may have one of ten values (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9).

bit A single digit of a binary number. Therefore, a bit may only have a value of ‘0’ or ‘1’. A group of 8 bits is called a byte. Because the PSoC's M8C is an 8-bit microcontroller, the PSoC's native data chunk size is a byte.

bit rate (BR) The number of bits occurring per unit of time in a bit stream, usually expressed in bits per second (bps).

block
1. A functional unit that performs a single function, such as an oscillator.
2. A functional unit that may be configured to perform one of several functions, such as a digital PSoC block or an analog PSoC block.

Boolean Algebra In mathematics and computer science, Boolean algebras or Boolean lattices, are algebraic structures which "capture the essence" of the logical operations AND, OR and NOT as well as the set theoretic operations union, intersection, and complement. Boolean algebra also defines a set of theorems that describe how Boolean equations can be manipulated. For example, these theorems are used to simplify Boolean equations, which will reduce the number of logic elements needed to implement the equation.

The operators of Boolean algebra may be represented in various ways. Often they are simply written as AND, OR, and NOT. In describing circuits, NAND (NOT AND), NOR (NOT OR), XNOR (exclusive NOT OR), and XOR (exclusive OR) may also be used. Mathematicians often use + (for example, A+B) for OR and • for AND (for example, A*B) (because in some ways those operations are analogous to addition and multiplication in other algebraic structures) and represent NOT by a line drawn above the expression being negated (for example, ~A, A_, !A).

break-before-make The elements involved go through a disconnected state entering (‘break”) before the new connected state (“make”).

broadcast net A signal that is routed throughout the microcontroller and is accessible by many blocks or systems.

buffer
1. A storage area for data that is used to compensate for a speed difference, when transferring data from one device to another. Usually refers to an area reserved for I/O operations, into which data is read, or from which data is written.
2. A portion of memory set aside to store data, often before it is sent to an external device or as it is received from an external device.
3. An amplifier used to lower the output impedance of a system.

bus
1. A named connection of nets. Bundling nets together in a bus makes it easier to route nets with similar routing patterns.
2. A set of signals performing a common function and carrying similar data. Typically represented using vector notation; for example, address[7:0].
3. One or more conductors that serve as a common connection for a group of related devices.

byte A digital storage unit consisting of 8 bits.

C

C A high level programming language.

capacitance A measure of the ability of two adjacent conductors, separated by an insulator, to hold a charge when a voltage differential is applied between them. Capacitance is measured in units of Farads.

capture	To extract information automatically through the use of software or hardware, as opposed to hand-entering of data into a computer file.
chaining	Connecting two or more 8-bit digital blocks to form 16-, 24-, and even 32-bit functions. Chaining allows certain signals such as Compare, Carry, Enable, Capture, and Gate to be produced from one block to another.
checksum	The checksum of a set of data is generated by adding the value of each data word to a sum. The actual checksum can simply be the result sum or a value that must be added to the sum to generate a pre-determined value.
clear	To force a bit/register to a value of logic ‘0’.
clock	The device that generates a periodic signal with a fixed frequency and duty cycle. A clock is sometimes used to synchronize different logic blocks.
clock generator	A circuit that is used to generate a clock signal.
CMOS	The logic gates constructed using ***MOS*** transistors connected in a complementary manner. CMOS is an acronym for complementary metal-oxide semiconductor.
comparator	An electronic circuit that produces an output voltage or current whenever two input levels simultaneously satisfy predetermined amplitude requirements.
compiler	A program that translates a high level language, such as C, into machine language.
configuration	In a computer system, an arrangement of functional units according to their nature, number, and chief characteristics. Configuration pertains to hardware, software, firmware, and documentation. The configuration will affect system performance.
configuration space	In PSoC devices, the register space accessed when the XIO bit, in the CPU_F register, is set to ‘1’.
crowbar	A type of over-voltage protection that rapidly places a low resistance shunt (typically an SCR) from the signal to one of the power supply rails, when the output voltage exceeds a predetermined value.
crystal oscillator	An oscillator in which the frequency is controlled by a piezoelectric crystal. Typically a piezoelectric crystal is less sensitive to ambient temperature than other circuit components.
cyclic redundancy check (CRC)	A calculation used to detect errors in data communications, typically performed using a linear feedback shift register. Similar calculations may be used for a variety of other purposes such as data compression.

D

data bus	A bi-directional set of signals used by a computer to convey information from a memory location to the central processing unit and vice versa. More generally, a set of signals used to convey data between digital functions.
data stream	A sequence of digitally encoded signals used to represent information in transmission.
data transmission	The sending of data from one place to another by means of signals over a channel.

debugger A hardware and software system that allows the user to analyze the operation of the system under development. A debugger usually allows the developer to step through the firmware one step at a time, set break points, and analyze memory.

dead band A period of time when neither of two or more signals are in their active state or in transition.

decimal A base-10 numbering system, which uses the symbols 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9 (called digits) together with the decimal point and the sign symbols + (plus) and - (minus) to represent numbers.

default value Pertaining to the pre-defined initial, original, or specific setting, condition, value, or action a system will assume, use, or take in the absence of instructions from the user.

device The device referred to in this manual is the PSoC chip, unless otherwise specified.

die An unpackaged integrated circuit (IC), normally cut from a wafer.

digital A signal or function, the amplitude of which is characterized by one of two discrete values: '0' or '1'.

digital blocks The 8-bit logic blocks that can act as a counter, timer, serial receiver, serial transmitter, CRC generator, pseudo-random number generator, or SPI.

digital logic A methodology for dealing with expressions containing two-state variables that describe the behavior of a circuit or system.

digital-to-analog (DAC) A device that changes a digital signal to an analog signal of corresponding magnitude. The ***analog-to-digital (ADC)*** converter performs the reverse operation.

direct access The capability to obtain data from a storage device, or to enter data into a storage device, in a sequence independent of their relative positions by means of addresses that indicate the physical location of the data.

duty cycle The relationship of a clock period ***high time*** to its ***low time***, expressed as a percent.

E

emulator Duplicates (provides an emulation of) the functions of one system with a different system, so that the second system appears to behave like the first system.

External Reset (XRES) An active high signal that is driven into the PSoC device. It causes all operation of the CPU and blocks to stop and return to a pre-defined state.

F

falling edge A transition from a logic 1 to a logic 0. Also known as a negative edge.

feedback The return of a portion of the output, or processed portion of the output, of a (usually active) device to the input.

filter A device or process by which certain frequency components of a signal are attenuated.

firmware The software that is embedded in a hardware device and executed by the CPU. The software may be executed by the end user, but it may not be modified.

flag Any of various types of indicators used for identification of a condition or event (for example, a character that signals the termination of a transmission).

Flash An electrically programmable and erasable, non***volatile*** technology that provides users with the programmability and data storage of EPROMs, plus in-system erasability. Nonvolatile means that the data is retained when power is off.

Flash bank A group of Flash ROM blocks where Flash block numbers always begin with '0' in an individual Flash bank. A Flash bank also has its own block level protection information.

Flash block The smallest amount of Flash ROM space that may be programmed at one time and the smallest amount of Flash space that may be protected. A Flash block holds 64 bytes.

flip-flop A device having two stable states and two input terminals (or types of input signals) each of which corresponds with one of the two states. The circuit remains in either state until it is made to change to the other state by application of the corresponding signal.

frequency The number of cycles or events per unit of time, for a periodic function.

G

gain The ratio of output current, voltage, or power to input current, voltage, or power, respectively. Gain is usually expressed in dB.

gate

1. A device having one output channel and one or more input channels, such that the output channel state is completely determined by the input channel states, except during switching transients.
2. One of many types of combinational logic elements having at least two inputs (for example, AND, OR, NAND, and NOR (also see ***Boolean Algebra***)).

ground

1. The electrical neutral line having the same potential as the surrounding earth.
2. The negative side of DC power supply.
3. The reference point for an electrical system.
4. The conducting paths between an electric circuit or equipment and the earth, or some conducting body serving in place of the earth.

H

hardware A comprehensive term for all of the physical parts of a computer or embedded system, as distinguished from the data it contains or operates on, and the software that provides instructions for the hardware to accomplish tasks.

hardware reset A reset that is caused by a circuit, such as a POR, watchdog reset, or external reset. A hardware reset restores the state of the device as it was when it was first powered up. Therefore, all registers are set to the POR value as indicated in register tables throughout this document.

hexadecimal A base 16 numeral system (often abbreviated and called hex), usually written using the symbols 0-9 and A-F. It is a useful system in computers because there is an easy mapping from four bits to a single hex digit. Thus, one can represent every byte as two consecutive hexadecimal digits. Compare the binary, hex, and decimal representations:

bin	=	hex	=	dec
0000b	=	0x0	=	0
0001b	=	0x1	=	1
0010b	=	0x2	=	2
...				
1001b	=	0x9	=	9
1010b	=	0xA	=	10
1011b	=	0xB	=	11
...				
1111b	=	0xF	=	15

So the decimal numeral 79 whose binary representation is 0100 1111b can be written as 4Fh in hexadecimal (0x4F).

high time The amount of time the signal has a value of '1' in one period, for a periodic digital signal.

I

I2C A two-wire serial computer bus by Phillips Semiconductors. I2C is an Inter-Integrated Circuit. It is used to connect low-speed peripherals in an embedded system. The original system was created in the early 1980s as a battery control interface, but it was later used as a simple internal bus system for building control electronics. I2C uses only two bi-directional pins, clock and data, both running at +5 V and pulled high with resistors. The bus operates at 100 kbps in standard mode and 400 kbps in fast mode. I^2C™ is a trademark of NXP.

ICE The in-circuit emulator that allows users to test the project in a hardware environment, while viewing the debugging device activity in a software environment (PSoC Designer).

idle state A condition that exists whenever user messages are not being transmitted, but the service is immediately available for use.

impedance

1. The resistance to the flow of current caused by resistive, capacitive, or inductive devices in a circuit.
2. The total passive opposition offered to the flow of electric current. Note the impedance is determined by the particular combination of resistance, inductive reactance, and capacitive reactance in a given circuit.

input A point that accepts data, in a device, process, or channel.

input/output (I/O) A device that introduces data into or extracts data from a system.

instruction An expression that specifies one operation and identifies its operands, if any, in a programming language such as C or assembly.

integrated circuit (IC) A device in which components such as resistors, capacitors, diodes, and ***transistors*** are formed on the surface of a single piece of semiconductor.

interface The means by which two systems or devices are connected and interact with each other.

interrupt A suspension of a process, such as the execution of a computer program, caused by an event external to that process, and performed in such a way that the process can be resumed.

interrupt service routine (ISR) A block of code that normal code execution is diverted to when the M8C receives a hardware interrupt. Many interrupt sources may each exist with its own priority and individual ISR code block. Each ISR code block ends with the RETI instruction, returning the device to the point in the program where it left normal program execution.

J

jitter
1. A misplacement of the timing of a transition from its ideal position. A typical form of corruption that occurs on serial data streams.
2. The abrupt and unwanted variations of one or more signal characteristics, such as the interval between successive pulses, the amplitude of successive cycles, or the frequency or phase of successive cycles.

K

keeper A circuit that holds a signal to the last driven value, even when the signal becomes un-driven.

L

latency The time or delay that it takes for a signal to pass through a given circuit or network.

least significant bit (LSb) The binary digit, or bit, in a binary number that represents the least significant value (typically the right-hand bit). The bit versus byte distinction is made by using a lower case “b” for bit in LSb.

least significant byte (LSB) The byte in a multi-byte word that represents the least significant values (typically the right-hand byte). The byte versus bit distinction is made by using an upper case “B” for byte in LSB.

Linear Feedback Shift Register (LFSR) A shift register whose data input is generated as an ***XOR*** of two or more elements in the register chain.

load The electrical demand of a process expressed as power (watts), current (amps), or resistance (ohms).

logic function A mathematical function that performs a digital operation on digital data and returns a digital value.

look-up table (LUT) A logic block that implements several logic functions. The logic function is selected by means of select lines and is applied to the inputs of the block. For example: A 2 input LUT with 4 select lines can be used to perform any one of 16 logic functions on the two inputs resulting in a single logic output. The LUT is a combinational device; therefore, the input/output relationship is continuous, that is, not sampled.

low time The amount of time the signal has a value of ‘0’ in one period, for a periodic digital signal.

low voltage detect (LVD) A circuit that senses Vdd and provides an interrupt to the system when Vdd falls below a selected threshold.

M

M8C An 8-bit Harvard Architecture microprocessor. The microprocessor coordinates all activity inside a PSoC by interfacing to the Flash, SRAM, and register space.

macro A programming language macro is an abstraction, whereby a certain textual pattern is replaced according to a defined set of rules. The interpreter or compiler automatically replaces the macro instance with the macro contents when an instance of the macro is encountered. Therefore, if a macro is used 5 times and the macro definition required 10 bytes of code space, 50 bytes of code space will be needed in total.

mask
1. To obscure, hide, or otherwise prevent information from being derived from a signal. It is usually the result of interaction with another signal, such as noise, static, jamming, or other forms of interference.
2. A pattern of bits that can be used to retain or suppress segments of another pattern of bits, in computing and data processing systems.

master device A device that controls the timing for data exchanges between two devices. Or when devices are cascaded in width, the master device is the one that controls the timing for data exchanges between the cascaded devices and an external interface. The controlled device is called the ***slave device***.

microcontroller An integrated circuit chip that is designed primarily for control systems and products. In addition to a CPU, a microcontroller typically includes memory, timing circuits, and I/O circuitry. The reason for this is to permit the realization of a controller with a minimal quantity of chips, thus achieving maximal possible miniaturization. This in turn, will reduce the volume and the cost of the controller. The microcontroller is normally not used for general-purpose computation as is a microprocessor.

mixed-signal The reference to a circuit containing both analog and digital techniques and components.

mnemonic A tool intended to assist the memory. Mnemonics rely on not only repetition to remember facts, but also on creating associations between easy-to-remember constructs and lists of data. A two to four character string representing a microprocessor instruction.

mode A distinct method of operation for software or hardware. For example, the Digital PSoC block may be in either counter mode or timer mode.

modulation A range of techniques for encoding information on a carrier signal, typically a sine-wave signal. A device that performs modulation is known as a modulator.

Modulator A device that imposes a signal on a carrier.

MOS An acronym for metal-oxide semiconductor.

most significant bit (MSb) The binary digit, or bit, in a binary number that represents the most significant value (typically the left-hand bit). The bit versus byte distinction is made by using a lower case “b” for bit in MSb.

most significant byte (MSB) The byte in a multi-byte word that represents the most significant values (typically the left-hand byte). The byte versus bit distinction is made by using an upper case “B” for byte in MSB.

multiplexer (mux)
1. A logic function that uses a binary value, or address, to select between a number of inputs and conveys the data from the selected input to the output.
2. A technique which allows different input (or output) signals to use the same lines at different times, controlled by an external signal. Multiplexing is used to save on wiring and I/O ports.

N

NAND See ***Boolean Algebra***.

negative edge A transition from a logic 1 to a logic 0. Also known as a falling edge.

net The routing between devices.

nibble A group of four bits, which is one-half of a byte.

noise
1. A disturbance that affects a signal and that may distort the information carried by the signal.
2. The random variations of one or more characteristics of any entity such as voltage, current, or data.

NOR See ***Boolean Algebra***.

NOT See ***Boolean Algebra***.

O

OR See ***Boolean Algebra***.

oscillator A circuit that may be crystal controlled and is used to generate a clock frequency.

output The electrical signal or signals which are produced by an analog or digital block.

P

parallel The means of communication in which digital data is sent multiple bits at a time, with each simultaneous bit being sent over a separate line.

parameter Characteristics for a given block that have either been characterized or may be defined by the designer.

parameter block A location in memory where parameters for the SSC instruction are placed prior to execution.

parity A technique for testing transmitting data. Typically, a binary digit is added to the data to make the sum of all the digits of the binary data either always even (even parity) or always odd (odd parity).

path
1. The logical sequence of instructions executed by a computer.
2. The flow of an electrical signal through a circuit.

pending interrupts An interrupt that has been triggered but has not been serviced, either because the processor is busy servicing another interrupt or global interrupts are disabled.

phase The relationship between two signals, usually the same frequency, that determines the delay between them. This delay between signals is either measured by time or angle (degrees).

Phase-Locked Loop (PLL) An electronic circuit that controls an ***oscillator*** so that it maintains a constant phase angle relative to a reference signal.

pin	A terminal on a hardware component. Also called lead.
pinouts	The pin number assignment: the relation between the logical inputs and outputs of the PSoC device and their physical counterparts in the printed circuit board (PCB) package. Pinouts will involve pin numbers as a link between schematic and PCB design (both being computer generated files) and may also involve pin names.
port	A group of pins, usually eight.
positive edge	A transition from a logic 0 to a logic 1. Also known as a rising edge.
posted interrupts	An interrupt that has been detected by the hardware but may or may not be enabled by its mask bit. Posted interrupts that are not masked become pending interrupts.
Power On Reset (POR)	A circuit that forces the PSoC device to reset when the voltage is below a pre-set level. This is one type of ***hardware reset***.
program counter	The instruction pointer (also called the program counter) is a register in a computer processor that indicates where in memory the CPU is executing instructions. Depending on the details of the particular machine, it holds either the address of the instruction being executed, or the address of the next instruction to be executed.
protocol	A set of rules. Particularly the rules that govern networked communications.
PSoC®	Cypress Semiconductor's Programmable System-on-Chip™ (PSoC).
PSoC blocks	*See* ***analog blocks*** and ***digital blocks***.
PSoC Designer™	The software for Cypress' Programmable System-on-Chip technology.
pulse	A rapid change in some characteristic of a signal (for example, phase or frequency), from a baseline value to a higher or lower value, followed by a rapid return to the baseline value.
pulse width modulator (PWM)	An output in the form of duty cycle which varies as a function of the applied measurand.

R

RAM	An acronym for random access memory. A data-storage device from which data can be read out and new data can be written in.
register	A storage device with a specific capacity, such as a bit or byte.
reset	A means of bringing a system back to a know state. See ***hardware reset*** and ***software reset***.
resistance	The resistance to the flow of electric current measured in ohms for a conductor.
revision ID	A unique identifier of the PSoC device.
ripple divider	An asynchronous ripple counter constructed of flip-flops. The clock is fed to the first stage of the counter. An n-bit binary counter consisting of n flip-flops that can count in binary from 0 to 2^n - 1.
rising edge	*See* ***positive edge***.

ROM An acronym for read only memory. A data-storage device from which data can be read out, but new data cannot be written in.

routine A block of code, called by another block of code, that may have some general or frequent use.

routing Physically connecting objects in a design according to design rules set in the reference library.

runt pulses In digital circuits, narrow pulses that, due to non-zero rise and fall times of the signal, do not reach a valid high or low level. For example, a runt pulse may occur when switching between asynchronous clocks or as the result of a race condition in which a signal takes two separate paths through a circuit. These race conditions may have different delays and are then recombined to form a glitch or when the output of a flip-flop becomes metastable.

S

sampling The process of converting an analog signal into a series of digital values or reversed.

schematic A diagram, drawing, or sketch that details the elements of a system, such as the elements of an electrical circuit or the elements of a logic diagram for a computer.

seed value An initial value loaded into a linear feedback shift register or random number generator.

serial
1. Pertaining to a process in which all events occur one after the other.
2. Pertaining to the sequential or consecutive occurrence of two or more related activities in a single device or channel.

set To force a bit/register to a value of logic 1.

settling time The time it takes for an output signal or value to stabilize after the input has changed from one value to another.

shift The movement of each bit in a word one position to either the left or right. For example, if the hex value 0x24 is shifted one place to the left, it becomes 0x48. If the hex value 0x24 is shifted one place to the right, it becomes 0x12.

shift register A memory storage device that sequentially shifts a word either left or right to output a stream of serial data.

sign bit The most significant binary digit, or bit, of a signed binary number. If set to a logic 1, this bit represents a negative quantity.

signal A detectable transmitted energy that can be used to carry information. As applied to electronics, any transmitted electrical impulse.

silicon ID A unique identifier of the PSoC silicon.

skew The difference in arrival time of bits transmitted at the same time, in parallel transmission.

slave device A device that allows another device to control the timing for data exchanges between two devices. Or when devices are cascaded in width, the slave device is the one that allows another device to control the timing of data exchanges between the cascaded devices and an external interface. The controlling device is called the master device.

software A set of computer programs, procedures, and associated documentation concerned with the operation of a data processing system (for example, compilers, library routines, manuals, and circuit diagrams). Software is often written first as source code, and then converted to a binary format that is specific to the device on which the code will be executed.

software reset A partial reset executed by software to bring part of the system back to a known state. A software reset will restore the M8C to a know state but not PSoC blocks, systems, peripherals, or registers. For a software reset, the CPU registers (CPU_A, CPU_F, CPU_PC, CPU_SP, and CPU_X) are set to 0x00. Therefore, code execution will begin at Flash address 0x0000.

SRAM An acronym for static random access memory. A memory device allowing users to store and retrieve data at a high rate of speed. The term static is used because, once a value has been loaded into an SRAM cell, it will remain unchanged until it is explicitly altered or until power is removed from the device.

SROM An acronym for supervisory read only memory. The SROM holds code that is used to boot the device, calibrate circuitry, and perform Flash operations. The functions of the SROM may be accessed in normal user code, operating from Flash.

stack A stack is a data structure that works on the principle of Last In First Out (LIFO). This means that the last item put on the stack is the first item that can be taken off.

stack pointer A stack may be represented in a computer's inside blocks of memory cells, with the bottom at a fixed location and a variable stack pointer to the current top cell.

state machine The actual implementation (in hardware or software) of a function that can be considered to consist of a set of states through which it sequences.

sticky A bit in a register that maintains its value past the time of the event that caused its transition, has passed.

stop bit A signal following a character or block that prepares the receiving device to receive the next character or block.

switching The controlling or routing of signals in circuits to execute logical or arithmetic operations, or to transmit data between specific points in a network.

Switch phasing The clock that controls a given switch, PHI1 or PHI2, in respect to the switch capacitor (SC) blocks. The PSoC SC blocks have two groups of switches. One group of these switches is normally closed during PHI1 and open during PHI2. The other group is open during PHI1 and closed during PHI2. These switches can be controlled in the normal operation, or in reverse mode if the PHI1 and PHI2 clocks are reversed.

synchronous

1. A signal whose data is not acknowledged or acted upon until the next active edge of a clock signal.
2. A system whose operation is synchronized by a clock signal.

T

tap The connection between two blocks of a device created by connecting several blocks/components in a series, such as a shift register or resistive voltage divider.

terminal count The state at which a counter is counted down to zero.

threshold The minimum value of a signal that can be detected by the system or sensor under consideration.

transistors The transistor is a solid-state semiconductor device used for amplification and switching, and has three terminals: a small current or voltage applied to one terminal controls the current through the other two. It is the key component in all modern electronics. In digital circuits, transistors are used as very fast electrical switches, and arrangements of transistors can function as logic gates, RAM-type memory, and other devices. In analog circuits, transistors are essentially used as amplifiers.

tri-state A function whose output can adopt three states: 0, 1, and Z (high-impedance). The function does not drive any value in the Z state and, in many respects, may be considered to be disconnected from the rest of the circuit, allowing another output to drive the same ***net***.

U

UART A UART or universal asynchronous receiver-transmitter translates between parallel bits of data and serial bits.

user The person using the PSoC device and reading this manual.

user modules Pre-build, pre-tested hardware/firmware peripheral functions that take care of managing and configuring the lower level Analog and Digital PSoC Blocks. User Modules also provide high level ***API (Application Programming Interface)*** for the peripheral function.

user space The bank 0 space of the register map. The registers in this bank are more likely to be modified during normal program execution and not just during initialization. Registers in bank 1 are most likely to be modified only during the initialization phase of the program.

V

Vdd A name for a power net meaning "voltage drain." The most positive power supply signal. Usually 5 or 3.3 volts.

volatile Not guaranteed to stay the same value or level when not in scope.

Vss A name for a power net meaning "voltage source." The most negative power supply signal.

W

watchdog timer	A timer that must be serviced periodically. If it is not serviced, the CPU will reset after a specified period of time.
waveform	The representation of a signal as a plot of amplitude versus time.

X

XOR	See ***Boolean Algebra***.