

THIS SPEC IS OBSOLETE

Spec No: 001-17397

Spec Title: ADJUSTING PSOC (R) TRIMS FOR 3.3V AND 2.7V OPERATION AN2012

Sunset Owner: Dennis Seguine (SEG)

Replaced by: None

Adjusting PSoC® Trims for 3.3V and 2.7V Operation

AN2012

Author: Jeff Dahlin
Associated Project: Yes
Associated Part Family: CY8C21x34, CY8C21x24, CY8C24x33, CY8C24x34A, CY8C24x94, CY8C27x43, CY8C29x66
Software Version: PSoC Designer™ 5 and newer
Associated Application Notes: None

PROFILE

Application Note Abstract

For the PSoC® microcontroller to meet the Internal Main Oscillator frequency accuracy and Internal Voltage Reference accuracy specifications, proper settings must be loaded in the associated trim registers. The trim register settings are different for 5V operation (5.25V ≥ V_{DD} ≥ 4.75V), 3.3V operation (3.6V ≥ V_{DD} ≥ 3.0V), and 2.7V operation (2.4V ≥ V_{DD} ≥ 3.0V). Trim values for 5V operation are automatically loaded during power on reset (POR). This application note demonstrates how to set the trim voltages for 5V, 3.3V, and 2.7V operation.

Introduction

During factory test, trim values for the 2.7V, 3.3V, and 5V ranges are determined and stored in a device parameter table accessed by supervisor ROM. On power up, the Bandgap Trim Register (BDG_TR) and the Internal Main Oscillator Trim Register (IMO_TR) are automatically loaded with the factory settings for the 5V operating range. Designs that operate in the 3.3V and 2.7V range must load these registers with the correct factory settings. This is automatically done in *boot.asm*, based on the "Power Setting" setting in Global Resources. This application note documents the process required to perform this for projects that may require switching between V_{DD} ranges.

Setting Trims for 3.3V/2.7V Operation

The trim settings determined during factory calibration are accessed through the System Supervisor Call (SSC) instruction. See section 4.35 of the *PSoC Designer: Assembly Language User Guide* for a description of the SSC instruction and the "Supervisory ROM (SROM)" section of the *PSoC Technical Reference Manual (TRM)* for device-specific functions of the SSC instruction. Macros have been written to set the bandgap and IMO trim values based on the operating voltage range.

Note Be certain to carefully use the SSC instruction; it is a powerful instruction that can make unrecoverable changes to the PSoC microcontroller if used improperly.

These macros can be found in the *m8ssc.asm* file found in the External Headers folder of a PSoC Designer project. This application note includes a PSoC Designer project with sample code including a header and assembly file to set the trims in runtime from assembly or C.

Trim Table Parameters

The trim values are organized in the supervisory ROM. The values used depend on the IMO speed and voltage range in which the device operates. Each value has a symbolic name, hex value, and table number. Table 1 and Table 2 describe the bandgap and IMO parameters.

Table 1: Bandgap Flash Table References

Symbolic Name	Table	Address	Description
SSCTBL2_TRIM_BGR_2V	2	0xF8	2.7V Bandgap reference voltage trim
SSCTBL1_TRIM_BGR_3V	1	0xF8	3.3V Bandgap reference voltage trim
SSCTBL1_TRIM_BGR_5V	1	0xFC	5.0V Bandgap reference voltage trim

Table 2: IMO Flash Table References

Symbolic Name	Table	Address	Description
SSCTBL2_TRIM_IMO_2V_6MHz	2	0xFD	2.7V IMO trim for 6 MHz
SSCTBL2_TRIM_IMO_2V_12MHZ	2	0xF9	2.7V IMO trim for 12 MHz
SSCTBL2_TRIM_IMO_3V_6MHZ	2	0xFC	3.3V IMO trim for 6 MHz
SSCTBL1_TRIM_IMO_3V_24MHZ	1	0xF9	3.3V IMO trim for 24 MHz
SSCTBL1_TRIM_IMO_5V_24MHZ	1	0xFD	5.0V IMO trim for 24 MHz
SSCTBL2_TRIM_IMO_5V_6MHZ	2	0xFE	5.0V IMO trim for 6 MHz

[+] Feedback

Trim SSC Macros

There are four macros that modify the bandgap and IMO trims. The predefined macros modify the bandgap and IMO trim registers in a specific order. The macros take the following steps to modify the trim registers:

1. Insert the Voltage Trim Table number (01h) in [TABLE_ID].
2. Insert the stack frame key in [KEYSP].
3. Insert the Supervisory key in [KEY1].
4. Place the Table Read supervisor function code (06h) in A.
5. Execute the SSC instruction (opcode = 00h).
6. Update the trim registers from the values in the supervisor's 8-byte buffer.
7. Note [KEY1], [KEYSP], and [TABLE_ID] refer to locations in the supervisor's 8-byte buffer.

Steps 2, 3, 4, and 5 must be done in the order shown. The sequence of opcodes provides a signature for the debugger. If the sequence of opcodes is changed, the debugger does not synchronize correctly after the SSC instruction is executed, resulting in an Invalid Memory Reference error. If this happens, reset the ICE.

The macros handle the seven steps discussed in this section. The macro used depends on the flash parameter table, the operating voltage, and IMO frequency. The `M8SSC_SetableIMOTrim` and `M8SSC_SetTable-VoltageTrim` macros set the IMO and bandgap trims independently. If the bandgap and IMO trim values are stored in the same parameter table the `M8SSC_SetTableTrims` macro can be used. The `M8SSC_Set2TableTrims` macro sets the bandgap and IMO trims when the values are stored in two different parameter tables. Macro descriptions, arguments, and syntax are shown in Table 3 to Table 6.

Table 3: M8SSC_SetTableTrims Macro

Macro	M8SSC_SetTableTrims
Description	Loads the bandgap and IMO trim registers from the specified flash System Parameter table entries. The values come from the same table.
Arguments	
BYTE Table	Parameter table holding the trim values
BYTE Volt_Trim	Address (F8 - FF) of voltage trim value
BYTE IMP_Trim	Address (F8 - FF) of IMO trim value
BYTE Bypass	Value for AGNDBYP in the BDG_TR register (justified).
Syntax	`M8SSC_SetTableTrims(Table, IMO_Trim, Volt_Trim, Bypass)`

Table 4: M8SSC_SetTable2Trims Macro

Macro	M8SSC_Set2TableTrims
Description	Loads the bandgap and IMO trim registers from the specified flash System Parameter table entries. The values come from two different tables.
Arguments	
BYTE TableA	Parameter table holding the voltage value
BYTE Volt_Trim	Address (F8 - FF) of voltage trim value
BYTE TableB	Parameter table holding the IMO value
BYTE IMP_Trim	Parameter table holding the IMO value
BYTE Bypass	Value for AGNDBYP in the BDG_TR register (justified).
Syntax	`M8SSC_Set2TableTrims(TableA, IMO_Trim, TableB, Volt_Trim, Bypass)`

Table 5: M8SSC_SetTableVoltageTrim Macro

Macro	M8SSC_SetTableVoltageTrim
Description	Loads bandgap Voltage Trim register with values from the specified flash System Parameter table entry.
Arguments	
BYTE Table	Parameter Table holding the voltage value
BYTE Volt_Trim	Address (F8 - FF) of voltage trim value
BYTE Bypass	Value for AGNDBYP in the BDG_TR register (justified).
Syntax	`M8SSC_SetTableVoltageTrim( Table, Volt_Trim, Bypass)`

Table 6: M8SSC_SetTableIMOTrims Macro

Macro	M8SSC_SetTableIMOTrim
Description	Loads IMO trim register with values from the specified flash System Parameter table entry.
Arguments	
BYTE Table	The parameter table holding the voltage value
BYTE Volt_Trim	Address (F8 - FF) of IMO trim value
Syntax	`M8SSC_SetTableIMOTrim( Table, IMO_Trim)`

Troubleshooting

If debugging halts with an error in a project that uses the SSC instruction, make sure that the macros are being used. The M8SSC macros present the opcodes in a specific order. The In-Circuit Emulator (ICE) relies on this signature to detect that an SSC instruction is being executed (SSC instructions are handled differently by the ICE). If this order is not maintained, the ICE does not recognize the SSC instruction and emulation halts with an error while debugging in PSoC Designer.

OBSOLETE

[+] Feedback

Summary

This application note explains how to load proper settings in PSoC's trim registers. The process to set up different registers for varying voltages is also explained.

Sample Code

The macros to set the IMO and bandgap trims are executed in the "*m8SSC.inc*" file. Because there is no header file associated with the file, they cannot be called from "*main.c*" or other C programs. The project attached to this application note contains both an assembly and header file with functions to adjust the bandgap and internal main oscillator trims.
The header file provides a mechanism to call these routines from a C program.

Code 1: "AdjustTrim.asm"

```
;-------------------------------------------------------------------------------
;;  FILENAME: AdjustTrim.asm
;;
;;  DESCRIPTION: This file contains functions that call the M8SSC.inc trim
;;            macros.  Three functions are available:
;;               Force5V0Trims
;;               Force3V3Trims
;;               Force2V7Trims
;;            The functions can be called from both .asm and .c files.
;;
;-------------------------------------------------------------------------------

include ".\lib\GlobalParams.inc"  ;File generated by PSoC Designer
include "m8c.inc"                       ;Part specific file
include "m8ssc.inc"               ;Part specific file

;-------------------------------------------------------------------------------
; Export Declarations
;-------------------------------------------------------------------------------
export Force2V7Trims
export _Force2V7Trims
export Force3V3Trims
export _Force3V3Trims
export Force5V0Trims
export _Force5V0Trims

;-------------------------------------------------------------------------------
;  FUNCTION NAME: Force2V7Trims
;
;  DESCRIPTION:
;      Set 2.7V IMO and Bandgap trims based on the System status and control
;      register value.  If Bit 4 of CPI_SCR1 is "1" the device is operating
;      in SLIMO mode.
;-------------------------------------------------------------------------------
Force2V7Trims:
_Force2V7Trims:
       ; Set 2.7V trim with SSC macro
    M8SSC_SetTableVoltageTrim 2, SSCTBL2_TRIM_BGR_2V, AGND_BYPASS_JUST

       ; Set IMO trim based on the IMO speed with the SSC macro
    IF (CPU_SCR1 & CPU_SCR1_SLIMO)
       M8SSC_SetTableIMOTrim 1, SSCTBL2_TRIM_IMO_2V_6MHz
    ELSE
       M8SSC_SetTableIMOTrim 2, SSCTBL2_TRIM_IMO_2V_12MHZ
    ENDIF


;-------------------------------------------------------------------------------
```

OBSOLETE

[+] Feedback

```
;  FUNCTION NAME: Force3V3Trims
;
;  DESCRIPTION:
;      Set 3.3V IMO and Bandgap trims based on the System status and control
;      register value.  If Bit 4 of CPI_SCR1 is "1" the device is operating
;      in SLIMO mode.
;-------------------------------------------------------------------------
Force3V3Trims:
_Force3V3Trims:
       ; Set 3.3V trim with SSC macro
    M8SSC_SetTableVoltageTrim 1, SSCTBL1_TRIM_BGR_3V, AGND_BYPASS_JUST

       ; Set IMO trim based on the IMO speed with the SSC macro
    IF (CPU_SCR1 & CPU_SCR1_SLIMO)
       M8SSC_SetTableIMOTrim 2, SSCTBL2_TRIM_IMO_3V_6MHZ
    ELSE
       M8SSC_SetTableIMOTrim 1, SSCTBL1_TRIM_IMO_3V_24MHZ
    ENDIF

;-------------------------------------------------------------------------
;  FUNCTION NAME: Force5V0Trims
;
;  DESCRIPTION:
;      Set 5.0V IMO and Bandgap trims based on the System status and control
;      register value.  If Bit 4 of CPI_SCR1 is "1" the device is operating
;      in SLIMO mode.
;-------------------------------------------------------------------------
Force5V0Trims:
_Force5V0Trims:
       ; Set 3.3V trim with SSC macro
    M8SSC_SetTableVoltageTrim 1, SSCTBL1_TRIM_BGR_5V, AGND_BYPASS_JUST

       ; Set IMO trim based on the IMO speed with the SSC macro
    IF (CPU_SCR1 & CPU_SCR1_SLIMO)
       M8SSC_SetTableIMOTrim 2, SSCTBL2_TRIM_IMO_5V_6MHZ
    ELSE
       M8SSC_SetTableIMOTrim 1, SSCTBL1_TRIM_IMO_5V_24MHZ
    ENDIF


; end of file AdjustTrim.asm
```

Code 2: "AdjustTrim.h"

```
/* Create pragmas to support proper argument and return value passing */
#pragma fastcall16 Force5V0Trims
#pragma fastcall16 Force3V3Trims
#pragma fastcall16 Force2V7Trims

//-------------------------------------------------
// Prototypes of the AdjustTrim.
//-------------------------------------------------
extern void Force5V0Trims(void);
extern void Force2V7Trims(void);
extern void Force3V3Trims(void);
```

OBSOLETE

[+] Feedback

Document History

Document Title: Adjusting PSoC® Trims for 3.3V and 2.7V Operations

Document Number: 001-17397

Revision	ECN	Orig. of Change	Submission Date	Description of Change
**	1331063	JVY	07/30/2007	New application note
*A	1536344	JVY	10/03/2007	Updated copyright. Add source disclaimer, revision disclaimer, Samples Request Form link, PSoC App. Note Index link. .pdf has been stamped. No technical updates.
*B	2803128	REID	11/10/2009	Updated PSoC Designer 5.0 content. Changed title to Adjusting PSoC® Trims for 3.3V and 2.7V Operations. Added sections on Trim table parameters and Trim SSC Macros.
*C	3082157	MEH	11/09/10	Obsolete document.

OBSOLETE

In March of 2007, Cypress recataloged all of its Application Notes using a new documentation number and revision code. This new documentation number and revision code (001-xxxxx, beginning with rev. **), located in the footer of the document, will be used in all subsequent revisions.

PSoC is a registered trademark of Cypress Semiconductor Corp. PSoC Designer is a trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are the property of their respective owners.

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
Phone: 408-943-2600
Fax: 408-943-4730
http://www.cypress.com/

© Cypress Semiconductor Corporation, 2007-2010. The information contained herein is subject to change without notice. Cypress Semiconductor Corporation assumes no responsibility for the use of any circuitry other than circuitry embodied in a Cypress product. Nor does it convey or imply any license under patent or other rights. Cypress products are not warranted nor intended to be used for medical, life support, life saving, critical control or safety applications, unless pursuant to an express written agreement with Cypress. Furthermore, Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress products in life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

This Source Code (software and/or firmware) is owned by Cypress Semiconductor Corporation (Cypress) and is protected by and subject to worldwide patent protection (United States and foreign), United States copyright laws and international treaty provisions. Cypress hereby grants to licensee a personal, non-exclusive, non-transferable license to copy, use, modify, create derivative works of, and compile the Cypress Source Code and derivative works for the sole purpose of creating custom software and or firmware in support of licensee product to be used only in conjunction with a Cypress integrated circuit as specified in the applicable agreement. Any reproduction, modification, translation, compilation, or representation of this Source Code except as specified above is prohibited without the express written permission of Cypress.

Disclaimer: CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Cypress reserves the right to make changes without further notice to the materials described herein. Cypress does not assume any liability arising out of the application or use of any product or circuit described herein. Cypress does not authorize its products for use as critical components in life-support systems where a malfunction or failure may reasonably be expected to result in significant injury to the user. The inclusion of Cypress' product in a life-support systems application implies that the manufacturer assumes all risk of such use and in doing so indemnifies Cypress against all charges.

Use may be limited by and subject to the applicable Cypress software license agreement.

[+] Feedback