6-Bit Successive Approximation ADC Datasheet SAR6 V 1.5

Copyright © 2002-2012 Cypress Semiconductor Corporation. All Rights Reserved.

Resources	PSoC® Blocks			API Memory (Bytes)		Pins (per External I/O and Clock)
	Digital	Analog CT	Analog SC	Flash	RAM	
CY8C29/27/26/25/24/22x13, CY8CLED04/08/16, CY8CLED0xD, CY8CLED0xG, CY8CTST120, CY8CTMG120, CY8CTMA120, CY8C28x45, CY8CPLC20, CY8CLED16P01, CY8C28x43, CY8C28x52						
	0	0	1	58	0	1

See AN2239, ADC Selection Guide for other converters.

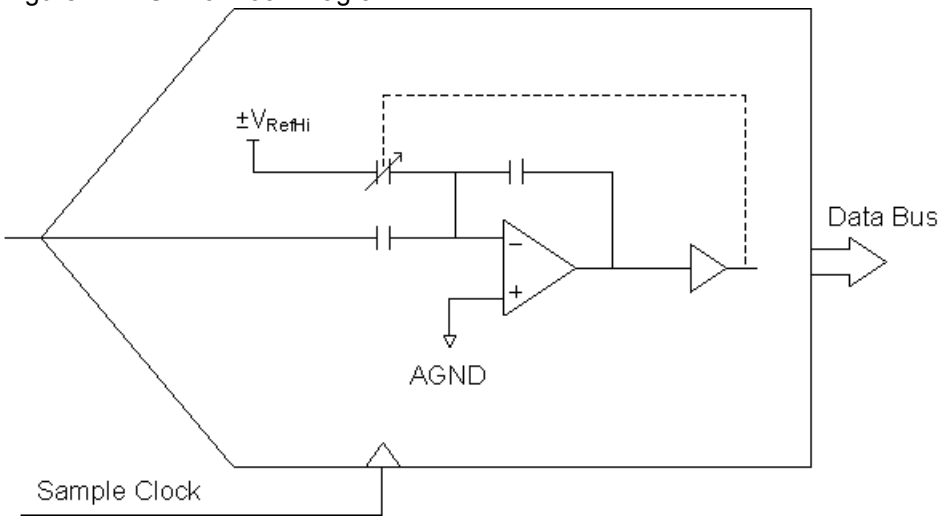
For one or more fully configured, functional example projects that use this user module go to www.cypress.com/psocexampleprojects.

Features and Overview

- 6-bit resolution
- Single PSoC block
- Conversion time of 25 µs, typical
- API optimized to help minimize aperture jitter

The SAR6 User Module converts an input voltage to a digital code, using a single switched-capacitor analog PSoC block. It features typical conversion times of 25 µs, producing a 2's complement value in the closed interval of [-32..+31] for each sample. The SAR6 Application Programming Interface (API) provides a time-equalized function, so that synchronous sampling can be managed by a timing loop for minimum aperture jitter.

Figure 1. SAR6 Block Diagram

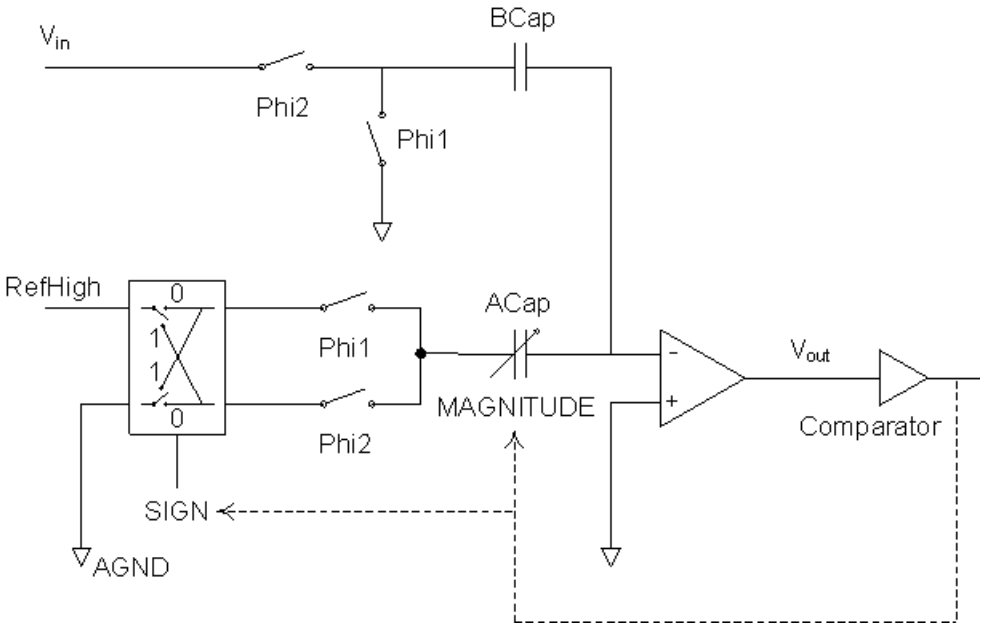


Functional Description

The acronym SAR stands for “successive approximation register." In this case, the register holding the result of the conversion is the PSoC block’s CR0 register. The basic operating principle is the formation of a series of approximations by scaling the reference voltage, RefHigh, and subtracting it from the input voltage. When the scaled reference matches the input voltage, the difference is zero, or equal to analog ground within the limits of 6-bit resolution.

Binary search reduces the number of approximations required and thus, the total conversion time. This procedure first determines the sign of the input voltage by setting the five ACap MAGNITUDE bits in figure below to zero, while holding BCap at its maximum fixed value. A comparator in the switched-cap PSoC block drives the column comparator bus. Because the inverting terminal of the opamp is used when the comparator output is high, the input voltage is negative with respect to AGND. The procedure then sets SIGN, so that the reference voltage impressed across the ACap opposes the input voltage across BCap. The second approximation sets ACap to half its maximum capacitance, by turning the most significant MAGNITUDE bits on. The resulting value on the comparator bus determines whether the next adjustment to ACap will be up or down by a quarter, to either ¼ or ¾ of its maximum value. Down to ¼ entails turning the previous MAGNITUDE bit off and the next most significant bit on. Up to ¾ leaves the previous MAGNITUDE bit "as is" and sets the next most significant bit on. The binary search continues to refine its approximation of the input voltage in this way, until the SIGN and all 5 MAGNITUDE bits are determined. The result is converted into a 1-byte 2’s complement value.

Figure 2. Simplified Schematic of the SAR6



In the ideal case,

Equation 1

$$V_{OUT} = V_{IN}\left(\frac{BCap}{Fcap}\right) \pm V_{REFHI}\left(\frac{ACap}{FCap}\right) = AGND = 0$$

As a result:

Equation 2

$$V_{IN} = \pm V_{REFHI}\left(\frac{ACap}{BCap}\right)$$

BCap is fixed at its maximum value of 31. Thus, the output code represents, in absolute terms, a comparison of the input to the reference voltage scaled by ±(0..31)/31.

Φ_1 and Φ_2 are two clock signals generated from the analog block's input clock. Their frequency is equal to the input clock frequency divided by four. Φ_1 and Φ_2 have the same frequency but are 180º out of phase. For more information on Φ_1 and Φ_2 see the PSoC Technical Reference Manual.

DC and AC Electrical Characteristics

The following values are indicative of expected performance and based on initial characterization data. Unless otherwise specified in the table below, TA = 25ºC, Vdd = 5.0V, Power HIGH, OpAmp bias LOW, output referenced to 2.5V external Analog Ground on P2[4] with 1.25 external Vref on P2[6].

Table 1. 5.0V SAR6 DC and AC Electrical Characteristics

Parameter	Typical	Limit	Units	Conditions and Notes
INPUT				
Input Range	--	Vss to Vdd		Ref Mux = Vdd/2 ± Vdd/2
Input Capacitance	3	--	pF	
Input Impedance	1/(C*clk)[1]	--	Ω	
$t_{convert}$, Conversion Time	--	20	µs	12 MHz CPU, f_{clock} = 250 kHz 24 MHz CPU, f_{clock} = 333 kHz
f_{clock}, Internal Update Rate	--	32 to 333	kHz	Column Clock ÷4
DC Accuracy				
Resolution	--	6	Bits	
DNL	.25	--	LSB	Column Clock 1.33 MHz
INL	.75	--	LSB	
V_{OS}, Offset Voltage[6]	8	--	mV	
Gain Error				
Including Reference Gain Error	1.5	--	% FSR	
Excluding Reference Gain Error[3]	0.4	--	% FSR	
Operating Current				
Low Power	140	--	µA	
Med Power	510	--	µA	
High Power	1890	--	µA	

The following values are indicative of expected performance and based on initial characterization data. Unless otherwise specified in the table below, all limits guaranteed for TA = 25ºC, Vdd = 3.3V, Power HIGH, OpAmp bias LOW, output referenced to 1.64V external Analog Ground on P2[4] with 1.25 external Vref on P2[6].

Table 2. 3.3V SAR6 DC and AC Electrical Characteristics

Parameter	Typical	Limit	Units	Conditions and Notes
INPUT				
Input Range	--	Vss to Vdd		Ref Mux = Vdd/2 ± Vdd/2
Input Capacitance	3	--	pF	
Input Impedance	1/(C*clk)[1]	--	Ω	
$t_{convert}$, Conversion Time	--	20	µs	12 MHz CPU, f_{clock} = 250 kHz
f_{clock}, Internal Update Rate	--	32 to 333	kHz	Column Clock ÷4
DC Accuracy				
Resolution	--	6	Bits	
DNL	.25	--	LSB	Column Clock 1.33 MHz
INL	.75	--	LSB	
V_{OS}, Offset Voltage[6]	8	--	mV	
Gain Error				
Including Reference Gain Error	3.5	--	% FSR	
Excluding Reference Gain Error[3]	2.4	--	% FSR	
Operating Current				
Low Power	140	--	µA	
Med Power	500	--	µA	
High Power	1840	--	µA	

Unless otherwise specified in the table below, all limits guaranteed for T_A = -40ºC to +85ºC, V_{dd} = 4.75V to 5.5V, Power HIGH, OpAmp bias LOW, output referenced to 2.5V external Analog Ground on P2[4] with 1.25 external Vref on P2[6].

Table 3. 5.0V SAR6 DC and AC Electrical Characteristics

Parameter	Typical[4]	Limit[5]	Units	Conditions[2,3] and Notes
INPUT				
Input Range[3]	--	Vss to Vdd		Vdd/2 +/- Vdd/2
Input Capacitance[4]	0.8	--	pF	
Input Impedance[5,6]	1/(C*clk)[1]	--	Ω	
Resolution	--	6	Bits	2s Complement
$t_{convert}$, Conversion Time	--	20	µs	12 MHz CPU, f_{clock} = 250 kHz 24 MHz CPU, f_{clock} = 333 kHz
f_{clock}, Internal Update Rate[7]	--	32 to 333	kHz	
DC ACCURACY				
Resolution	--	6	Bits	
INL	.03	.08	LSB	
DNL	.02	.05	LSB	
Monotonicity	--	½	Bit	
Gain Error	1.0	2.5	%FSR	
V_{OS}, Offset Voltage[6]	8	43	mV	
OPERATING CURRENT[8]				
Low Power	125	--	µA	
Med Power	280	--	µA	
High Power	780	1000	µA	

Unless otherwise specified in the table below, all limits guaranteed for TA = -40ºC to +85ºC, Vdd = 3.0V to 3.6V, Power HIGH, OpAmp bias LOW, output referenced to 1.64V external Analog Ground on P2[4] with 1.25 external Vref on P2[6].

Table 4. 3.3V SAR6 DC and AC Electrical Characteristics

Parameter	Typical	Limit	Units	Conditions[2] and Notes
INPUT				
Input Range[3]	--	Vss to Vdd		Vdd/2 ± Vdd/2
Input Capacitance[4]	0.8	--	pF	
Input Impedance[5,6]	1/(C*clk)[1]	--	Ω	
Resolution	--	6	Bits	2's Complement
$t_{convert}$, Conversion Time	--	20	µs	12 MHz CPU, f_{clock} = 250 kHz
f_{clock}, Internal Update Rate	--	32 to 333	kHz	Column Clock ÷4
DC ACCURACY				
Resolution	--	6	Bits	
INL	.04	.09	LSB	
DNL	.02	.04	LSB	
Monotonicity	--	½	Bit	
Gain Error	1.0	2.5	%FSR	
V_{OS}, Offset Voltage	7	31	mV	
Operating Current				
Low Power	100	--	µA	
Med Power	250	--	µA	
High Power	640	900	µA	

Electrical Characteristics Notes

1. clk = (column clock)/4 = fclock/4; C = Total capacitance of the input capacitor. For this user module, the input capacitance of the block is 31*Csc. Csc is the "Capacitor unit value" and is available in the respective silicon datasheet.
2. f_{clock} = 125 kHz, external AGND 2.50V, external V_{Ref} 1.23V, unless otherwise noted.
3. REFPWR = HIGH, SCPOWER = ON, PSoC block power HIGH, unless otherwise noted.
4. Typical values represent statistical mean plus 1σ.
5. Limits guaranteed by testing or statistical analysis.
6. Two's complement zero scale offset to external AGND. Does not include analog output buffer offset error.
7. Limit for Φ_1, Φ_2 specified for 3 dB increase in broadband noise.
8. PSoC block current requirements exclusive of reference current.

Placement

The SA block can be placed in any of the switched capacitor PSoC blocks. However, it will require the comparator bus for the particular column to which it is connected. Other user modules that require use of the column comparator cannot be placed in the same column.

Parameters and Resources

The SAR6 maps onto any switched-capacitor PSoC block. The only requirement is that the comparator bus in that column is not allocated to another user module in the same configuration. The SAR6 symbolically names its switched-capacitor PSoC™ block SA. After placing SA with the Device Editor, its SignalSource must be set to complete its configuration. Additional tasks include configuring associated pins, multiplexors and/or other PSoC blocks to deliver the input to SA, and setting up a source clock for the column in which SA resides.

SignalSource

The Device Editor restricts input selections to the connections possible for the particular PSoC block onto which SA is mapped. In general, input ports may be directly accessible or that the input must come from another switched capacitor (SC) or continuous time (CT) PSoC block. Although it may be possible to connect an input to an unmapped and unconfigured PSoC block, the selection should be planned for a block that will provide a useful input, such as filters and amplifiers.

A successive approximation register type ADC, requires that the input signal is stable during the conversion. This implementation of a SAR, does not contain an internal sample and hold circuit to stabilize the input during conversion. This means that the input voltage should not change more than 50 percent of the LSB during conversion. For example, if the Reference multiplexor is set at (Vdd/2) +/- BandGap, one LSB is equal to about 1.3V/32 or about 40 mV. With this setting, the input should not deviate more than 20 mV during the conversion. The actual conversion period is six times the period of the sample clock (Φ_1/Φ_2) which is one fourth of the analog column clock. If the analog column clock is set at 1 MHz, the sample clock will be 250 kHz. So the conversion time will be 6 times the sample clock period (6 * 1/250 kHz) or 24 μS.

Analog Column Clock

The analog column clock multiplexors select the source clock used to generate the phase clocks, Φ_1 and Φ_2, that control each successive approximation step. The phase clock generator divides the column clock by four to produce Φ_1 and Φ_2, so the column clock frequency is four times faster than the actual analog step approximation rate. Two levels of multiplexing provide choices for the column clock that include any of the digital blocks and the system clock dividers. The Electrical Characteristics section, above, specifies lower and upper limits for the column clock frequency.

Application Programming Interface

The Application Programming Interface (API) routines are provided as part of the user module to allow the designer to deal with the module at a higher level. This section specifies the interface to each function together with related constants provided by the “include" files.

Each time a user module is placed, it is assigned an instance name. By default, PSoC Designer assigns the SAR6_1 to the first instance of this user module in a given project. It can be changed to any unique value that follows the syntactic rules for identifiers. The assigned instance name becomes the prefix of every global function name, variable and constant symbol. In the following descriptions the instance name has been shortened to SAR6 for simplicity.

Note

In this, as in all user module APIs, the values of the A and X register may be altered by calling an API function. It is the responsibility of the calling function to preserve the values of A and X prior to the call if those values are required after the call. This “registers are volatile" policy was selected for efficiency reasons and has been in force since version 1.0 of PSoC Designer. The C compiler automatically takes care of this requirement. Assembly language programmers must ensure their code observes the policy, too. Though some user module API function may leave A and X unchanged, there is no guarantee they will do so in the future.

For Large Memory Model devices, it is also the caller's responsibility to preserve any value in the CUR_PP, IDX_PP, MVR_PP, and MVW_PP registers. Even though some of these registers may not be modified now, there is no guarantee that will remain the case in future releases.

Entry points are provided to initialize the SAR6 User Module, perform and read conversions, and disable the SAR6 function.

SAR6_Start

Description:

Performs all required initialization for this user module and sets the power level for the switched capacitor PSoC block.

C Prototype:

```
void  SAR6_Start(BYTE bPowerSetting)
```

Assembler:

```
mov   A, bPowerSetting
lcall  SAR6_Start
```

Parameters:

bPowerSetting: One byte that specifies the power level. Following reset and configuration, the PSoC block assigned to the SAR is powered down. Symbolic names, provided in C and assembly, and their associated values, are given in the following table.

Symbolic Name	Value
SAR6_OFF	0
SAR6_LOWPOWER	1
SAR6_MEDPOWER	2
SAR6_HIGHPOWER	3

Returns:

None

Side Effects:

The comparator bus will be driven. The A and X registers may be altered by this function.

SAR6_SetPower

Description:

Sets the power level for the switched capacitor PSoC block. May be used to turn the block off and on.

C Prototype:

```
void  SAR6_SetPower(BYTE bPowerSetting)
```

Assembler:

```
mov   A, bPowerSetting
lcall  SAR6_SetPower
```

Parameters:

bPowerSetting: Same as the PowerSetting parameter used for the Start entry point.

Returns:

None

Side Effects:

The comparator bus will be driven. The A and X registers may be altered by this function.

SAR6_cGetSample

Description:

Performs a conversion, returning a 2’s complement value representing the ratio of the input voltage to the reference voltage, both relative to analog ground.

C Prototypes:

```
CHAR  SAR6_cGetSample(void)
```

Assembler:

```
lcall  SAR6_cGetSample
mov   [abResultBuffer], A
```

Parameters:

None

Returns:

Returns the sample value.

Side Effects:

The A and X registers may be altered by this function.

SAR6_Stop

Description:

Powers the user module off.

C Prototype:

```
void  SAR6_Stop()
```

Assembler Macro:

```
lcall  SAR6_Stop
```

Parameters:

None

Returns:

None

Side Effects:

The A and X registers may be altered by this function.

Sample Firmware Source Code

The sample code given here illustrates the use of a PGA to buffer an input signal that is then converted to a digital value by the SAR6 User Module. The code drives the result onto Port 2 with a “data valid" strobe applied to the high-order bit of Port 1.

For proper functionality, all Port 2 pins and P1[7] must be configured for strong drive mode. Additionally, a PGA User Module must be added to the project that passes an external signal on Port 0 to the input of the SAR6 ADC.

```
;;; SAR6 Example Code
;;;
;;; Send a stream of samples to Port 2 with a data-valid
;;; (rising edge) strobe on P1[7].
;;;
;;;
;;;
include "m8c.inc"             ; part specific constants and macros
include "memory.inc"            ; Constants & macros for SMM/LMM and Compiler
include "PSoCAPI.inc"           ; PSoC API definitions for all user modules

export _main

_main:

    mov A, PGA_HIGHPOWER
    call PGA_Start              ; Start PGA with HIGH power setting
```

```
mov A, SAR6_HIGHPOWER
    call SAR6_Start             ; Start ADC with HIGH power setting
    mov reg[PRT2DR], 00h        ; Set Port 2 initial value
    mov reg[PRT1DR], 80h        ; Initialize data valid strobe on P1[7]


loop:
    xor reg[PRT1DR], 80h        ; De-assert data valid strobe
    call SAR6_cGetSample        ; Get one ADC sample
    mov reg[PRT2DR], A          ; Applied sample value to Port 2
    xor reg[PRT1DR], 80h        ; Assert strobe for data valid
    jmp loop                    ; Repeat the loop
```

The same Program in C is:

```
//-------------------------------------------------------------------
// C Example
//
// Write data from SAR6 to Port 2 and toggle a strobe pin on P1[7].
//-------------------------------------------------------------------
#include <m8c.h>
#include "PSoCAPI.h"


void main(void)
{
    char bResult;                       // This variable holds the ADC sample

PGA_Start(PGA_HIGHPOWER);           // Start PGA in HIGH power mode

SAR6_Start(SAR6_HIGHPOWER);         // Start ADC in HIGH power mode

PRT2DR = 0x00;                      // Set Port 2 initial value
    PRT1DR = 0x80;                      // Init data valid strobe on P1[7]

while(1)                            // Loop forever
    {
  PRT1DR ^= 0x80;                 // De-assert the data valid strobe
        bResult = SAR6_cGetSample();    // Get one ADC sample
        PRT2DR = bResult;               // Write sample data to Port 2
        PRT1DR ^= 0x80;                 // Assert data valid strobe
    }
}
```

Configuration Registers

The following registers are used for the SAR6 switched capacitor SA block.

Table 5. Block SA, mapped to a Switched Capacitor Type SCA or SCC PSoC Block

Bit	7	6	5	4	3	2	1	0
CR0	0	0	Sign and Magnitude[5:0]					
CR1	0	1	0	1	1	1	1	1
CR2	0	1	0	0	0	0	0	0
CR3	0	0	0	0	Input Select		Power	

Table 6. Block SA, mapped to a Switched Capacitor Type SCB or SCD PSoC Block

Bit	7	6	5	4	3	2	1	0
CR0	0	0	Sign and Magnitude[5:0]					
CR1	0	1	0	1	1	1	1	1
CR2	0	1	0	0	0	0	0	0
CR3	0	0	0	0	0	Input Select	Power	

The Sign and Magnitude bit field is manipulated directly by the hardware and the API.

The Input Selector field is initialized by the Device Editor as determined by the InputSource parameter.

The Input Selector field is initialized by the Device Editor as determined by the InputSource parameter. Power: 0 = Off, 1 = Low, 2 = Medium, 3 = Full. Default is off. To set, use the Start call in the API.

Version History

Version	Originator	Description
1.5	DHA	Added Version History

Note PSoC Designer 5.1 introduces a Version History in all user module datasheets. This section documents high level descriptions of the differences between the current and previous user module versions.

Copyright © 2002-2012 Cypress Semiconductor Corporation. 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.

PSoC Designer™ and Programmable System-on-Chip™ are trademarks and PSoC® is a registered trademark of Cypress Semiconductor Corp. All other trademarks or registered trademarks referenced herein are property of the respective corporations.

Any 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.