

TMS320x2802x, 2803x Piccolo Analog-to-Digital Converter (ADC) and Comparator

Reference Guide



Literature Number: SPRUGE5F
December 2008–Revised December 2011

Preface	6
1 Analog-to-Digital Converter (ADC)	8
1.1 Features	8
1.2 Block Diagram	9
1.3 SOC Principle of Operation	9
1.4 ADC Conversion Priority	13
1.5 Simultaneous Sampling Mode	16
1.6 EOC and Interrupt Operation	16
1.7 Power Up Sequence	17
1.8 ADC Calibration	17
1.9 Internal/External Reference Voltage Selection	19
1.10 ADC Registers	20
1.11 ADC Timings	38
1.12 Internal Temperature Sensor	42
2 Comparator Block	43
2.1 Features	43
2.2 Comparator Function	44
2.3 DAC Reference	45
2.4 Initialization	46
2.5 Digital Domain Manipulation	46
2.6 Comparator Registers	48
Appendix A Revision History	53

List of Figures

1	ADC Block Diagram	9
2	SOC Block Diagram.....	10
3	ADCINx Input Model	12
4	ONESHOT Single Conversion	13
5	Round Robin Priority Example.....	14
6	High Priority Example	15
7	Interrupt Structure	17
8	ADC Control Register 1 (ADCCTL1) (Address Offset 00h)	20
9	ADC Control Register 2 (ADCCTL2) (Address Offset 01h)	22
10	ADC Interrupt Flag Register (ADCINTFLG) (Address Offset 04h).....	23
11	ADC Interrupt Flag Clear Register (ADCINTFLGCLR) (Address Offset 05h)	23
12	ADC Interrupt Overflow Register (ADCINTOVF) (Address Offset 06h)	24
13	ADC Interrupt Overflow Clear Register (ADCINTOVFCLR) (Address Offset 07h)	24
14	Interrupt Select 1 And 2 Register (INTSEL1N2) (Address Offset 08h).....	25
15	Interrupt Select 3 And 4 Register (INTSEL3N4) (Address Offset 09h).....	25
16	Interrupt Select 5 And 6 Register (INTSEL5N6) (Address Offset 0Ah)	25
17	Interrupt Select 7 And 8 Register (INTSEL7N8) (Address Offset 0Bh)	25
18	Interrupt Select 9 And 10 Register (INTSEL9N10) (Address Offset 0Ch)	26
19	ADC Start of Conversion Priority Control Register (SOCPRICTL).....	27
20	ADC Sample Mode Register (ADCSAMPLEMODE) (Address Offset 12h).....	29
21	ADC Interrupt Trigger SOC Select 1 Register (ADCINTSOCSEL1) (Address Offset 14h)	30
22	ADC Interrupt Trigger SOC Select 2 Register (ADCINTSOCSEL2) (Address Offset 15h)	31
23	ADC SOC Flag 1 Register (ADCSOCFLG1) (Address Offset 18h).....	31
24	ADC SOC Force 1 Register (ADCSOCFRC1) (Address Offset 1Ah)	31
25	ADC SOC Overflow 1 Register (ADCSOCOVF1) (Address Offset 1Ch).....	32
26	ADC SOC Overflow Clear 1 Register (ADCSOCOVFCLR1) (Address Offset 1Eh)	32
27	ADC SOC0 - SOC15 Control Registers (ADCSOCxCTL) (Address Offset 20h - 2Fh)	33
28	ADC Reference/Gain Trim Register (ADCREFTRIM) (Address Offset 40h)	35
29	ADC Offset Trim Register (ADCOFFTRIM) (Address Offset 41h)	35
30	Comparator Hysteresis Control Register (COMPHYSTCTL) (Address Offset 4Ch).....	36
31	ADC Revision Register (ADCREV) (Address Offset 4Fh)	36
32	ADC RESULT0 - RESULT15 Registers (ADCRESULTx) (PF1 Block Address Offset 00h - 0Fh).....	37
33	Timing Example For Sequential Mode / Late Interrupt Pulse.....	38
34	Timing Example For Sequential Mode / Early Interrupt Pulse.....	39
35	Timing Example For Simultaneous Mode / Late Interrupt Pulse	40
36	Timing Example For Simultaneous Mode / Early Interrupt Pulse	41
37	Timing Example for NONOVERLAP Mode	41
38	Temperature Sensor Transfer Function	42
39	Comparator Block Diagram	44
40	Comparator.....	44
41	Ramp Generator Block Diagram	45
42	Ramp Generator Behavior	46
43	Comparator Control (COMPCTL) Register	48
44	Compare Output Status (COMPSTS) Register.....	49
45	DAC Control (DACCTL) Register	49
46	DAC Value (DACVAL) Register	50
47	Ramp Generator Maximum Reference Active (RAMPMAXREF_ACTIVE) Register	50

48	Ramp Generator Maximum Reference Shadow (RAMPMAXREF_SHDW) Register	51
49	Ramp Generator Decrement Value Active (RAMPDECVAL_ACTIVE) Register.....	51
50	Ramp Generator Decrement Value Shadow (RAMPDECVAL_SHDW) Register.....	51
51	Ramp Generator Status (RAMPSTS) Register.....	51

List of Tables

1	Sample timings with different values of ACQPS	11
2	ADC Configuration & Control Registers (AdcRegs and AdcResult):	20
3	ADC Control Register 1 (ADCCTL1) Field Descriptions	21
4	ADC Control Register 2 (ADCCTL2) Field Descriptions	23
5	ADC Interrupt Flag Register (ADCINTFLG) Field Descriptions	23
6	ADC Interrupt Flag Clear Register (ADCINTFLGCLR) Field Descriptions.....	24
7	ADC Interrupt Overflow Register (ADCINTOVF) Field Descriptions.....	24
8	ADC Interrupt Overflow Clear Register (ADCINTOVFCLR) Field Descriptions	25
9	INTSELxNy Register Field Descriptions.....	26
10	SOCPRCTL Register Field Descriptions	28
11	ADC Sample Mode Register (ADCSAMPLEMODE) Field Descriptions	29
12	ADC Interrupt Trigger SOC Select 1 Register (ADCINTSOCSEL1) Register Field Descriptions.....	30
13	ADC Interrupt Trigger SOC Select 2 Register (ADCINTSOCSEL2) Field Descriptions.....	31
14	ADC SOC Flag 1 Register (ADCSOCFLG1) Field Descriptions	31
15	ADC SOC Force 1 Register (ADCSOCFRC1) Field Descriptions	32
16	ADC SOC Overflow 1 Register (ADCSOCOVF1) Field Descriptions	32
17	ADC SOC Overflow Clear 1 Register (ADCSOCOVFCLR1) Field Descriptions	32
18	ADC SOC0 - SOC15 Control Registers (ADCSOCxCTL) Register Field Descriptions	33
19	ADC Reference/Gain Trim Register (ADCREFTRIM) Field Descriptions	35
20	ADC Offset Trim Register (ADCOFFTRIM) Field Descriptions.....	36
21	Comparator Hysteresis Control Register (COMPHYSTCTL) Field Descriptions.....	36
22	ADC Revision Register (ADCREV) Field Descriptions	36
23	ADC RESULT0 - ADCRESULT15 Registers (ADCRESULTx) Field Descriptions.....	37
24	Comparator Truth Table	44
25	Comparator Module Registers	48
26	COMPCTL Register Field Descriptions.....	49
27	Compare Output Status (COMPSTS) Register Field Descriptions	49
28	DACCTL Register Field Descriptions	50
29	DAC Value (DACVAL) Register Field Descriptions	50
30	Ramp Generator Maximum Reference Active (RAMPMAXREF_ACTIVE) Register Field Descriptions	50
31	Ramp Generator Maximum Reference Shadow (RAMPMAXREF_SHDW) Register Field Descriptions	51
32	Ramp Generator Decrement Value Active (RAMPDECVAL_ACTIVE) Register Field Descriptions	51
33	Ramp Generator Decrement Value Shadow (RAMPDECVAL_SHDW) Register Field Descriptions	51
34	Ramp Generator Status (RAMPSTS) Register Field Descriptions.....	52
35	Changes in this Document	53

Read This First

Notational Conventions

This document uses the following conventions.

- Hexadecimal numbers are shown with the suffix h or with a leading 0x. For example, the following number is 40 hexadecimal (decimal 64): 40h or 0x40.
- Registers in this document are shown in figures and described in tables.
 - Each register figure shows a rectangle divided into fields that represent the fields of the register. Each field is labeled with its bit name, its beginning and ending bit numbers above, and its read/write properties below. A legend explains the notation used for the properties.
 - Reserved bits in a register figure designate a bit that is used for future device expansion.

Related Documents From Texas Instruments

The following documents are available for download from the Texas Instruments website, www.ti.com.

[SPRS523](#) — **TMS320F28020, TMS320F28021, TMS320F28022, TMS320F28023, TMS320F28026, TMS320F28027 Piccolo Microcontrollers Data Manual** contains the pinout, signal descriptions, as well as electrical and timing specifications for the 2802x devices.

[SPRZ292](#) — **TMS320F28020, TMS320F28021, TMS320F28022, TMS320F28023, TMS320F28026, TMS320F28027 Piccolo MCU Silicon Errata** describes known advisories on silicon and provides workarounds.

[SPRS584](#) — **TMS320F28032, TMS320F28033, TMS320F28034, TMS320F28035 Piccolo Microcontrollers Data Manual** contains the pinout, signal descriptions, as well as electrical and timing specifications for the 2803x devices.

[SPRZ295](#) — **TMS320F28032, TMS320F28033, TMS320F28034, TMS320F28035 Piccolo MCU Silicon Errata** describes known advisories on silicon and provides workarounds.

CPU User's Guides—

[SPRU430](#) — **TMS320C28x CPU and Instruction Set Reference Guide** describes the central processing unit (CPU) and the assembly language instructions of the TMS320C28x fixed-point digital signal processors (DSPs). It also describes emulation features available on these DSPs.

Peripheral Guides—

[SPRUFN3](#) — **TMS320x2802x Piccolo System Control and Interrupts Reference Guide** describes the various interrupts and system control features of the 2802x microcontrollers (MCUs).

[SPRUGL8](#) — **TMS320x2803x Piccolo System Control and Interrupts Reference Guide** describes the various interrupts and system control features of the 2803x microcontrollers (MCUs).

[SPRU566](#) — **TMS320x28xx, 28xxx DSP Peripheral Reference Guide** describes the peripheral reference guides of the 28x digital signal processors (DSPs).

[SPRUG00](#) — **TMS320x2803x Piccolo Boot ROM Reference Guide** describes the purpose and features of the bootloader (factory-programmed boot-loading software) and provides examples of code. It also describes other contents of the device on-chip boot ROM and identifies where all of the information is located within that memory.

- [SPRUFN6](#) — **TMS320x2802x Piccolo Boot ROM Reference Guide** describes the purpose and features of the bootloader (factory-programmed boot-loading software) and provides examples of code. It also describes other contents of the device on-chip boot ROM and identifies where all of the information is located within that memory.
- [SPRUGE6](#) — **TMS320x2803x Piccolo Control Law Accelerator (CLA) Reference Guide** describes the operation of the Control Law Accelerator (CLA).
- [SPRUGE2](#) — **TMS320x2803x Piccolo Local Interconnect Network (LIN) Module Reference Guide** describes the operation of the Local Interconnect Network (LIN) Module.
- [SPRUFK8](#) — **TMS320x2803x Piccolo Enhanced Quadrature Encoder Pulse (eQEP) Reference Guide** describes the operation of the Enhanced Quadrature Encoder Pulse (eQEP) module, which is used for interfacing with a linear or rotary incremental encoder to get position, direction, and speed information from a rotating machine in high performance motion and position control systems. It includes the module description on registers.
- [SPRUGL7](#) — **TMS320x2803x Piccolo Enhanced Controller Area Network (eCAN) Reference Guide** describes the operation of the Enhanced Controller Area Network (eCAN) which uses established protocol to communicate serially with other controllers in electrically noisy environments.
- [SPRUGE5](#) — **TMS320x2802x, 2803x Piccolo Analog-to-Digital Converter (ADC) and Comparator Reference Guide** describes how to configure and use the on-chip ADC module, which is a 12-bit pipelined ADC.
- [SPRUGE9](#) — **TMS320x2802x, 2803x Piccolo Enhanced Pulse Width Modulator (ePWM) Module Reference Guide** describes the main areas of the enhanced pulse width modulator that include digital motor control, switch mode power supply control, UPS (uninterruptible power supplies), and other forms of power conversion.
- [SPRUGE8](#) — **TMS320x2802x, 2803x Piccolo High-Resolution Pulse Width Modulator (HRPWM)** describes the operation of the high-resolution extension to the pulse width modulator (HRPWM).
- [SPRUGH1](#) — **TMS320x2802x, 2803x Piccolo Serial Communications Interface (SCI) Reference Guide** describes how to use the SCI.
- [SPRUFZ8](#) — **TMS320x2802x, 2803x Piccolo Enhanced Capture (eCAP) Module Reference Guide** describes the enhanced capture module. It includes the module description and registers.
- [SPRUG71](#) — **TMS320x2802x, 2803x Piccolo Serial Peripheral Interface (SPI) Reference Guide** describes the SPI - a high-speed synchronous serial input/output (I/O) port - that allows a serial bit stream of programmed length (one to sixteen bits) to be shifted into and out of the device at a programmed bit-transfer rate.
- [SPRUFZ9](#) — **TMS320x2802x, 2803x Piccolo Inter-Integrated Circuit (I2C) Reference Guide** describes the features and operation of the inter-integrated circuit (I2C) module.
- Tools Guides—**
- [SPRU513](#) — **TMS320C28x Assembly Language Tools v5.0.0 User's Guide** describes the assembly language tools (assembler and other tools used to develop assembly language code), assembler directives, macros, common object file format, and symbolic debugging directives for the TMS320C28x device.
- [SPRU514](#) — **TMS320C28x Optimizing C/C++ Compiler v5.0.0 User's Guide** describes the TMS320C28x™ C/C++ compiler. This compiler accepts ANSI standard C/C++ source code and produces TMS320 DSP assembly language source code for the TMS320C28x device.
- [SPRU608](#) — **TMS320C28x Instruction Set Simulator Technical Overview** describes the simulator, available within the Code Composer Studio for TMS320C2000 IDE, that simulates the instruction set of the C28x™ core.

Analog-to-Digital Converter and Comparator

The ADC module described in this reference guide is a Type 3 ADC and exists on the Piccolo™ family of devices. The Comparator function described in this reference guide is a Type 0 Comparator. See the *TMS320C28xx, 28xxx DSP Peripheral Reference Guide* ([SPRU566](#)) for a list of all devices with modules of the same type, to determine the differences between the types, and for a list of device-specific differences within a type.

1 Analog-to-Digital Converter (ADC)

The ADC module described in this reference guide is a 12-bit recyclic ADC; part SAR, part pipelined. The analog circuits of this converter, referred to as the "core" in this document, include the front-end analog multiplexers (MUXs), sample-and-hold (S/H) circuits, the conversion core, voltage regulators, and other analog supporting circuits. Digital circuits, referred to as the "wrapper" in this document, include programmable conversions, result registers, interface to analog circuits, interface to device peripheral bus, and interface to other on-chip modules.

1.1 Features

The core of the ADC contains a single 12-bit converter fed by two sample and hold circuits. The sample and hold circuits can be sampled simultaneously or sequentially. These, in turn, are fed by a total of up to 16 analog input channels. See the device datasheet for the specific number of channels available. The converter can be configured to run with an internal bandgap reference to create true-voltage based conversions or with a pair of external voltage references (VREFHI/LO) to create ratiometric based conversions.

Contrary to previous ADC types, this ADC is not sequencer based. It is easy for the user to create a series of conversions from a single trigger. However, the basic principle of operation is centered around the configurations of individual conversions, called SOC's, or Start-Of-Conversions.

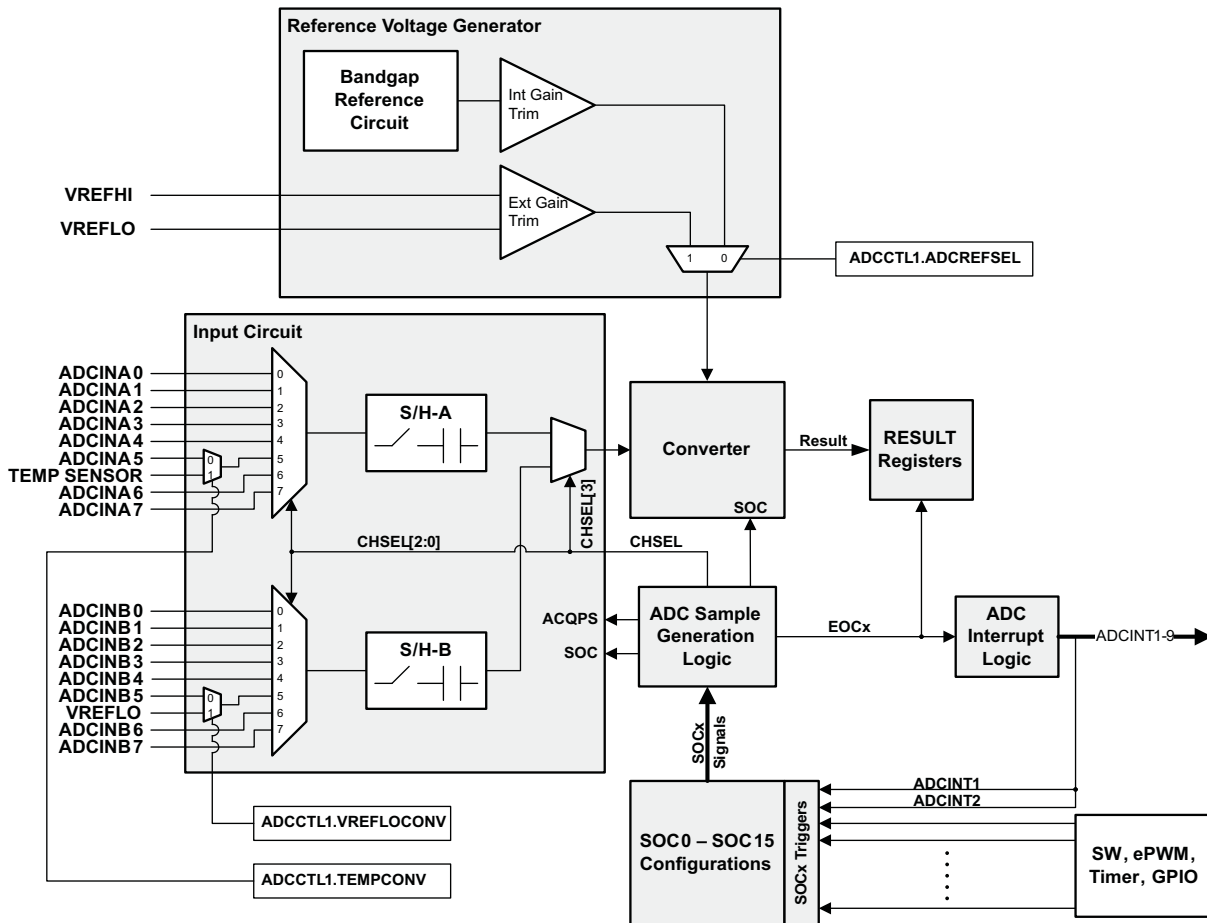
Functions of the ADC module include:

- 12-bit ADC core with built-in dual sample-and-hold (S/H)
- Simultaneous sampling or sequential sampling modes
- Full range analog input: 0 V to 3.3 V fixed, or VREFHI/VREFLO ratiometric
- Runs at full system clock, no prescaling required
- Up to 16-channel, multiplexed inputs
- 16 SOC's, configurable for trigger, sample window, and channel
- 16 result registers (individually addressable) to store conversion values
- Multiple trigger sources
 - S/W - software immediate start
 - ePWM 1-8
 - GPIO XINT2
 - CPU Timers 0/1/2
 - ADCINT1/2
- 9 flexible PIE interrupts, can configure interrupt request after any conversion

1.2 Block Diagram

Figure 1 shows the block diagram of the ADC module.

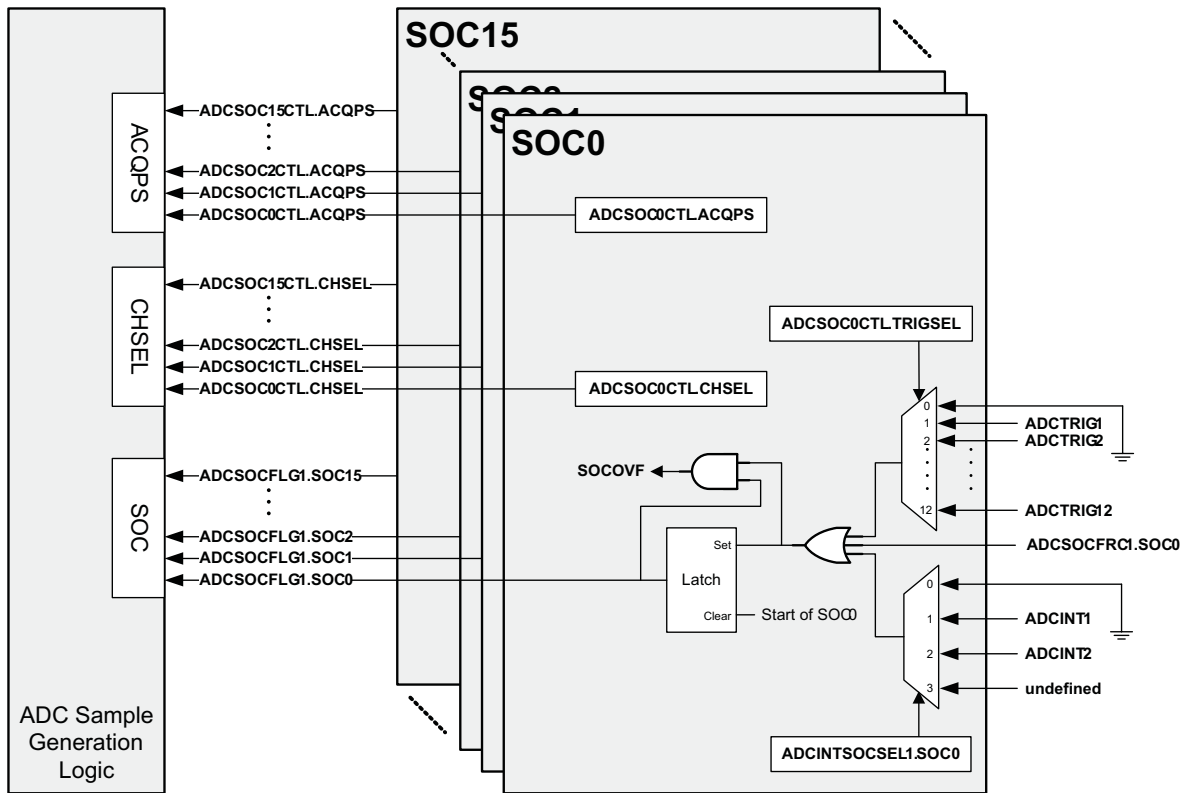
Figure 1. ADC Block Diagram



1.3 SOC Principle of Operation

Contrary to previous ADC types, this ADC is not sequencer based. Instead, it is SOC based. The term SOC is configuration set defining the single conversion of a single channel. In that set there are three configurations: the trigger source that starts the conversion, the channel to convert, and the acquisition (sample) window size. Each SOC is independently configured and can have any combination of the trigger, channel, and sample window size available. Multiple SOC's can be configured for the same trigger, channel, and/or acquisition window as desired. This provides a very flexible means of configuring conversions ranging from individual samples of different channels with different triggers, to oversampling the same channel using a single trigger, to creating your own series of conversions of different channels all from a single trigger.

The trigger source for SOC_x is configured by a combination of the TRIGSEL field in the ADCSOC_xCTL register and the appropriate bits in the ADCINTSOCSEL1 or ADCINTSOCSEL2 register. Software can also force an SOC event with the ADCSOCFRC1 register. The channel and sample window size for SOC_x are configured with the CHSEL and ACQPS fields of the ADCSOC_xCTL register.

Figure 2. SOC Block Diagram


For example, to configure a single conversion on channel ADCINA1 to occur when the ePWM3 timer reaches its period match you must first setup ePWM3 to output an SOCA or SOCB signal on a period match. See the [TMS320x2802x Piccolo Enhanced Pulse Width Modulator Module User's Guide \(SPRUGE9\)](#) on how to do this. In this case, we'll use SOCA. Then, setup one of the SOC's using its ADCSOCxCTL register. It makes no difference which SOC we choose, so we'll use SOC0. The fastest allowable sample window for the ADC is 7 cycles. Choosing the fastest time for the sample window, channel ADCINA1 for the channel to convert, and ePWM3 for the SOC0 trigger, we'll set the ACQPS field to 6, the CHSEL field to 1, and the TRIGSEL field to 9, respectively. The resulting value written into the register will be:

```
ADCSOC0CTL = 4846h;           // (ACQPS=6, CHSEL=1, TRIGSEL=9)
```

When configured as such, a single conversion of ADCINA1 will be started on an ePWM3 SOCA event with the resulting value stored in the ADCRESULT0 register.

If instead ADCINA1 needed to be oversampled by 3X, then SOC1, SOC2, and SOC3 could all be given the same configuration as SOC0.

```
ADCSOC1CTL = 4846h;           // (ACQPS=6, CHSEL=1, TRIGSEL=9)
ADCSOC2CTL = 4846h;           // (ACQPS=6, CHSEL=1, TRIGSEL=9)
ADCSOC3CTL = 4846h;           // (ACQPS=6, CHSEL=1, TRIGSEL=9)
```

When configured as such, four conversions of ADCINA1 will be started in series on an ePWM3 SOCA event with the resulting values stored in the ADCRESULT0 – ADCRESULT3 registers.

Another application may require 3 different signals to be sampled from the same trigger. This can be done by simply changing the CHSEL field for SOC0-SOC2 while leaving the TRIGSEL field unchanged.

```

ADCSOC0CTL = 4846h;           // (ACQPS=6, CHSEL=1, TRIGSEL=9)
ADCSOC1CTL = 4886h;           // (ACQPS=6, CHSEL=2, TRIGSEL=9)
ADCSOC2CTL = 48C6h;           // (ACQPS=6, CHSEL=3, TRIGSEL=9)

```

When configured this way, three conversions will be started in series on an ePWM3 SOCA event. The result of the conversion on channel ADCINA1 will show up in ADCRESULT0. The result of the conversion on channel ADCINA2 will show up in ADCRESULT1. The result of the conversion on channel ADCINA3 will show up in ADCRESULT2. The channel converted and the trigger have no bearing on where the result of the conversion shows up. The RESULT register is associated with the SOC.

NOTE: These examples are incomplete. Clocks must be enabled via the PCLKCR0 register and the ADC must be powered to work correctly. For a description of the PCLKCR0 register see the *TMS320F2802x Piccolo System Control and Interrupts Reference Guide* ([SPRUFN3](#)). For the power up sequence of the ADC, see [Section 1.7](#). The CLKDIV2EN bit in the ADCCTL2 register must also be set to a proper value to obtain correct frequency of operation. For more information on the ADCCTL2 register please refer to [Section 1.3.1](#).

1.3.1 ADC Acquisition (Sample and Hold) Window

External drivers vary in their ability to drive an analog signal quickly and effectively. Some circuits require longer times to properly transfer the charge into the sampling capacitor of an ADC. To address this, the ADC supports control over the sample window length for each individual SOC configuration. Each ADCSOCxCTL register has a 6-bit field, ACQPS, that determines the sample and hold (S/H) window size. The value written to this field is one less than the number of cycles desired for the sampling window for that SOC. Thus, a value of 15 in this field will give 16 clock cycles of sample time. The minimum number of sample cycles allowed is 7 (ACQPS=6). The total sampling time is found by adding the sample window size to the conversion time of the ADC, 13 ADC clocks. Examples of various sample times are shown below in [Table 1](#).

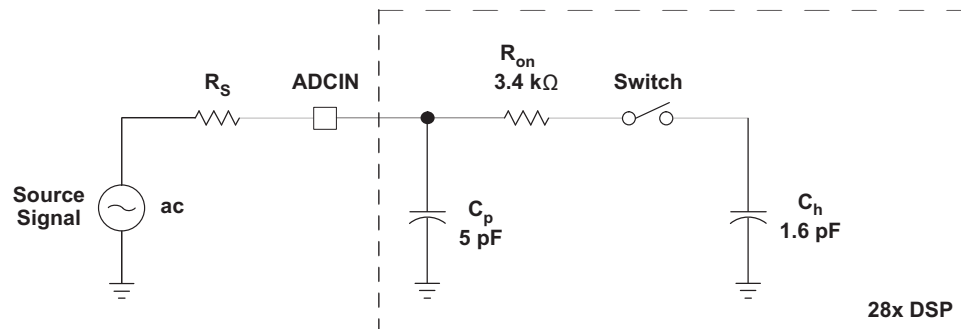
Table 1. Sample timings with different values of ACQPS

ADC Clock	ACQPS	Sample Window	Conversion Time (13 cycles)	Total Time to Process Analog Voltage ⁽¹⁾
40MHz	6	175 ns	325ns	500.00ns
40MHz	25	625 ns	325ns	950.00ns
60MHz	6	116.67ns	216.67ns	333.33ns
60MHz	25	433.67ns	216.67ns	650ns

⁽¹⁾ The total times are for a single conversion and do not include pipelining effects that increase the average speed over time.

As shown in [Figure 3](#), the ADCIN pins can be modeled as an RC circuit. With VREFLO connected to ground, a voltage swing from 0 to 3.3v on ADCIN yields a typical RC time constant of 2ns.

Figure 3. ADCINx Input Model



Typical Values of the Input Circuit Components:

Switch Resistance (R_{on}): 3.4 k Ω
 Sampling Capacitor (C_h): 1.6 pF
 Parasitic Capacitance (C_p): 5 pF
 Source Resistance (R_s): 50 Ω

1.3.2 Trigger Operation

Each SOC can be configured to start on one of many input triggers. Multiple SOC's can be configured for the same channel if desired. Following is a list of the available input triggers:

- Software
- CPU Timers 0/1/2 interrupts
- XINT2 SOC
- ePWM1-8 SOCA and SOCB

See the ADCSOCxCTL Register Bit Definitions for the configuration details of these triggers.

Additionally ADCINT1 and ADCINT2 can be fed back to trigger another conversion. This configuration is controlled in the ADCINTSOCSEL1/2 registers. This mode is useful if a continuous stream of conversions is desired. See section 1.6 for information on the ADC interrupt signals.

1.3.3 Channel Selection

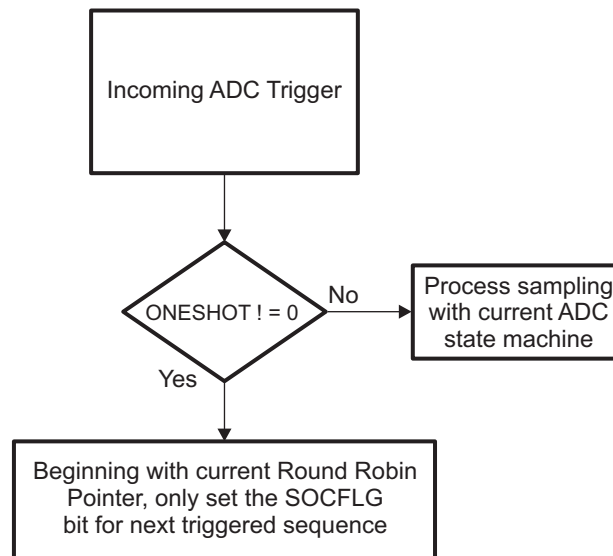
Each SOC can be configured to convert any of the available ADCIN input channels. When an SOC is configured for sequential sampling mode, the four bit CHSEL field of the ADCSOCxCTL register defines which channel to convert. When an SOC is configured for simultaneous sampling mode, the most significant bit of the CHSEL field is dropped and the lower three bits determine which pair of channels are converted.

ADCINA0 is shared with VREFHI, and therefore cannot be used as a variable input source when using external reference voltage mode. See [Section 1.9](#) for details on this mode.

1.3.4 ONESHOT Single Conversion Support

This mode will allow you to perform a single conversion on the next triggered SOC in the round robin scheme. The ONESHOT mode is only valid for channels present in the round robin wheel. Channels which are not configured for triggered SOC in the round robin scheme will get priority based on contents of the SOCPRIORITY field in the ADCSOCPRIORITYCTL register.

Figure 4. ONESHOT Single Conversion



The effect of ONESHOT mode on Sequential Mode and Simultaneous Mode is explained below.

Sequential mode: Only the next active SOC in RR mode (one up from current RR pointer) will be allowed to generate SOC; all other triggers for other SOC slots will be ignored.

Simultaneous mode: If current RR pointer has SOC with simultaneous enabled; active SOC will be incremented by 2 from the current RR pointer. This is because simultaneous mode will create result for SOC_x and SOC_{x+1}, and SOC_{x+1} will never be triggered by the user.

NOTE: ONESHOT = 1 and SOCPRIORITY = 11111 is not a valid combination for above implementation reasons. This should not be a desired mode of operation by the user in any case. The limitation of the above is that the next SOC's must eventually be triggered, or else the ADC will not generate new SOC's for other out-of-order triggers. Any non-orthogonal channels should be placed in the priority mode which is unaffected by ONESHOT mode

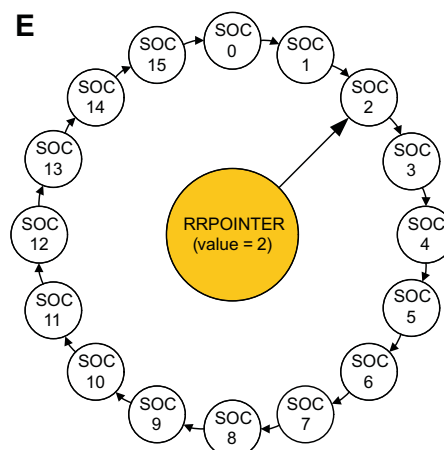
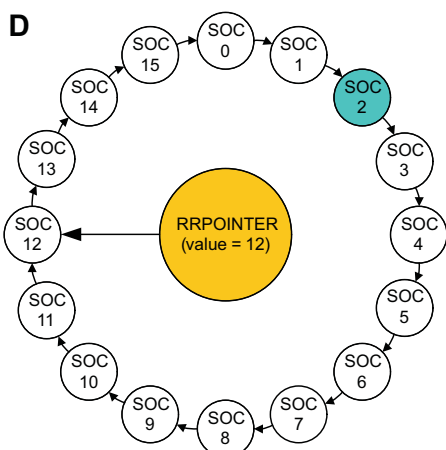
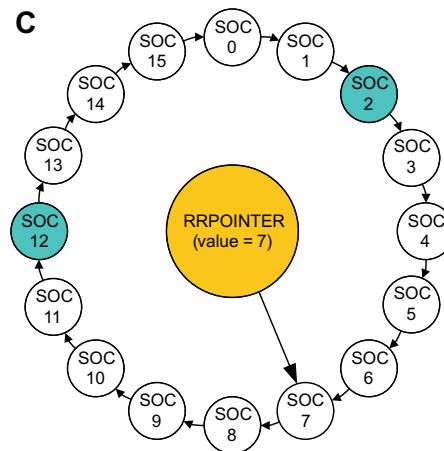
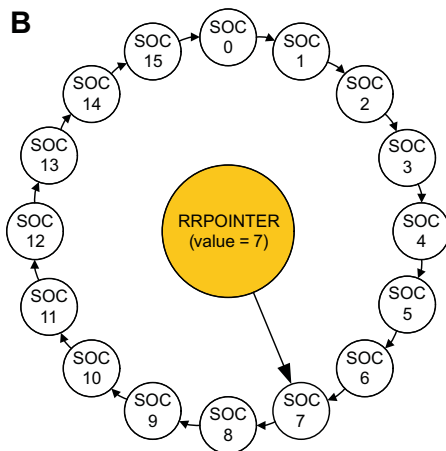
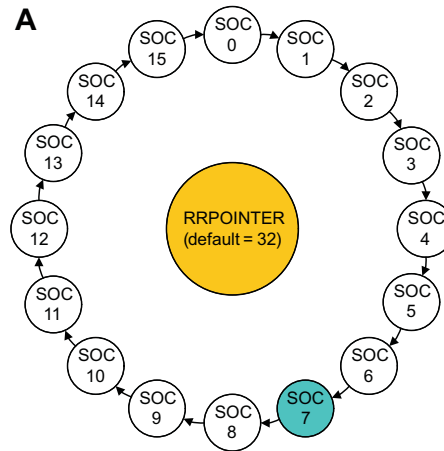
1.4 ADC Conversion Priority

When multiple SOC flags are set at the same time, one of two forms of priority determines the order in which they are converted. The default priority method is round robin. In this scheme, no SOC has an inherent higher priority than another. Priority depends on the round robin pointer (RRPOINTER). The RRPOINTER reflected in the ADCSOCPRIORITYCTL register points to the last SOC converted. The highest priority SOC is given to the next value greater than the RRPOINTER value, wrapping around back to SOC₀ after SOC₁₅. At reset the value is 32 since 0 indicates a conversion has already occurred. When RRPOINTER equals 32 the highest priority is given to SOC₀. The RRPOINTER is reset by a device reset, when the ADCCTL1.RESET bit is set, or when the SOCPRICTL register is written.

An example of the round robin priority method is given in [Figure 5](#).

Figure 5. Round Robin Priority Example

- A** After reset, SOC0 is highest priority SOC ;
SOC7 receives trigger;
SOC7 configured channel is converted immediately .
- B** RRPOINTER changes to point to SOC 7;
SOC8 is now highest priority SOC .
- C** SOC2 & SOC12 triggers rcvd . simultaneously;
SOC12 is first on round robin wheel ;
SOC12 configured channel is converted while
SOC2 stays pending .
- D** RRPOINTER changes to point to SOC 12;
SOC2 configured channel is now converted .
- E** RRPOINTER changes to point to SOC 2;
SOC3 is now highest priority SOC .



The SOC PRIORITY field in the ADCSOC PRIORITY CTL register can be used to assign high priority from a single to all of the SOC's. When configured as high priority, an SOC will interrupt the round robin wheel after any current conversion completes and insert itself in as the next conversion. After its conversion completes, the round robin wheel will continue where it was interrupted. If two high priority SOC's are triggered at the same time, the SOC with the lower number will take precedence.

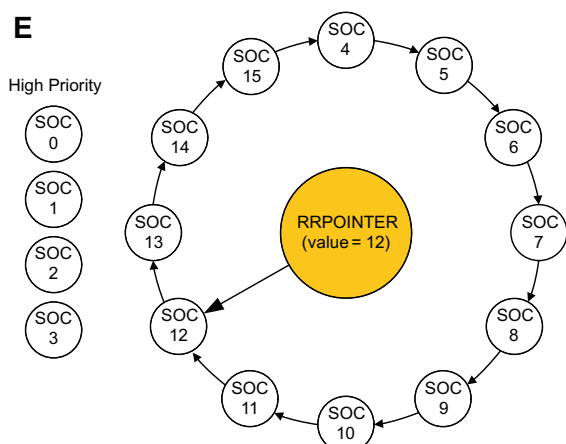
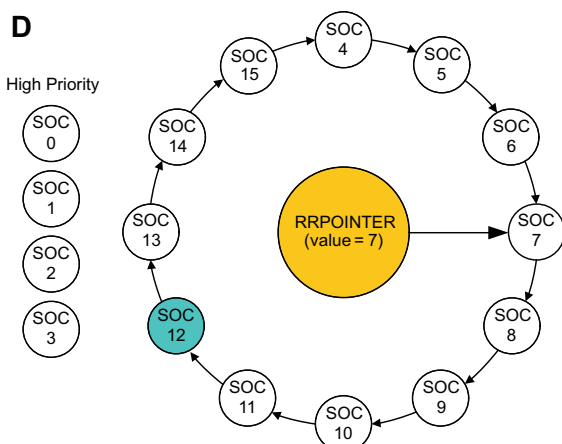
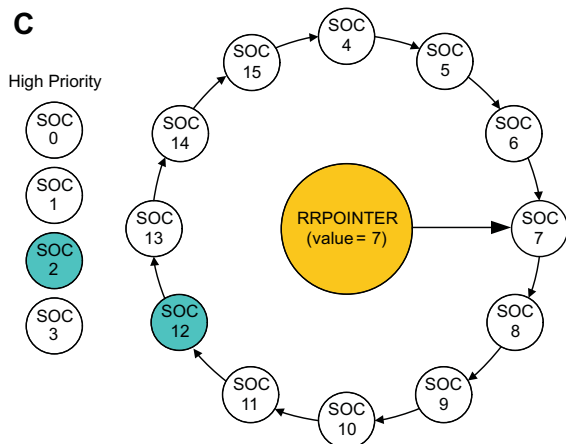
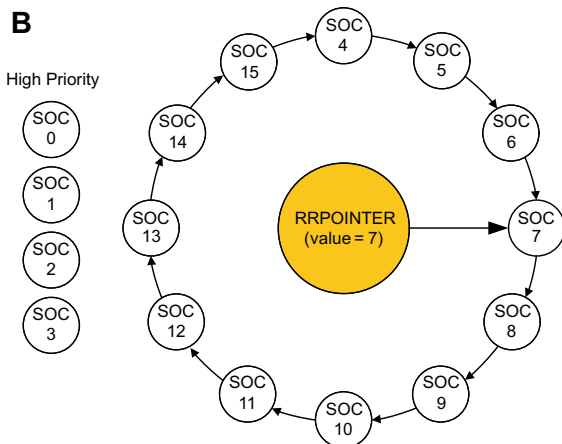
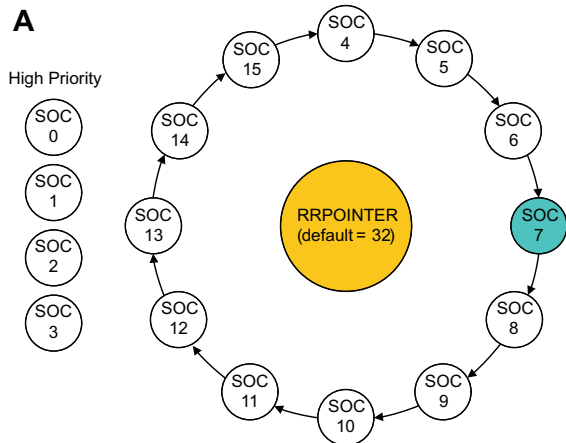
High priority mode is assigned first to SOC0, then in increasing numerical order. The value written in the SOC PRIORITY field defines the first SOC that is not high priority. In other words, if a value of 4 is written into SOC PRIORITY, then SOC0, SOC1, SOC2, and SOC3 are defined as high priority, with SOC0 the highest.

An example using high priority SOC's is given in [Figure 6](#).

Figure 6. High Priority Example

Example when SOC PRIORITY = 4

- A** After reset, SOC4 is 1st on round robin wheel ; SOC7 receives trigger ; SOC7 configured channel is converted immediately .
- B** RRPOINTER changes to point to SOC 7; SOC8 is now 1st on round robin wheel .
- C** SOC2 & SOC12 triggers rcvd . simultaneously ; SOC2 interrupts round robin wheel and SOC 2 configured channel is converted while SOC 12 stays pending .
- D** RRPOINTER stays pointing to 7; SOC12 configured channel is now converted .
- E** RRPOINTER changes to point to SOC 12; SOC13 is now 1st on round robin wheel .



1.5 Simultaneous Sampling Mode

In some applications it is important to keep the delay between the sampling of two signals minimal. The ADC contains dual sample and hold circuits to allow two different channels to be sampled simultaneously. Simultaneous sampling mode is configured for a pair of SOCx's with the ADCSAMPLEMODE register. The even numbered SOCx and the following odd numbered SOCx (i.e., SOC0 and SOC1) are coupled together with one enable bit (SIMULEN0, in this case). The coupling behavior is as follows:

- Either SOCx's trigger will start a pair of conversions.
- The pair of channels converted will consist of the A-channel and the B-channel corresponding to the value of the CHSEL field of the triggered SOCx. The valid values in this mode are 0-7.
- Both channels will be sampled simultaneously.
- The A channel will always convert first.
- The even EOCx pulse will be generated based off of the A-channel conversion, the odd EOCx pulse will be generated off of the B-channel conversion. See Section 1.6 for an explanation of the EOCx signals.
- The result of the A-channel conversion is placed in the even ADCRESULTx register and the result of the B-channel conversion is written to the odd ADCRESULTx register.

For example, if the ADCSAMPLEMODE.SIMULEN0 bit is set, and SOC0 is configured as follows:

CHSEL = 2 (ADCINA2/ADCINB2 pair)

TRIGSEL = 5 (ADCTRIG5 = ePWM1.ADCSOCA)

When the ePWM1 sends out an ADCSOCA trigger, both ADCINA2 and ADCINB2 will be sampled simultaneously (assuming priority). Immediately after, the ADCINA2 channel will be converted and its value will be stored in the ADCRESULT0 register. Depending on the ADCCTL1.INTPULSEPOS setting, the EOC0 pulse will either occur when the conversion of ADCINA2 begins or completes. Then the ADCINB2 channel will be converted and its value will be stored in the ADCRESULT1 register. Depending on the ADCCTL1.INTPULSEPOS setting, the EOC1 pulse will either occur when the conversion of ADCINB2 begins or completes.

Typically in an application it is expected that only the even SOCx of the pair will be used. However, it is possible to use the odd SOCx instead, or even both. In the latter case, both SOCx triggers will start a conversion. Therefore, caution is urged as both SOCx's will store their results to the same ADCRESULTx registers, possibly overwriting each other.

The rules of priority for the SOCx's remain the same as in sequential sampling mode.

[Section 1.11](#) shows the timing of simultaneous sampling mode.

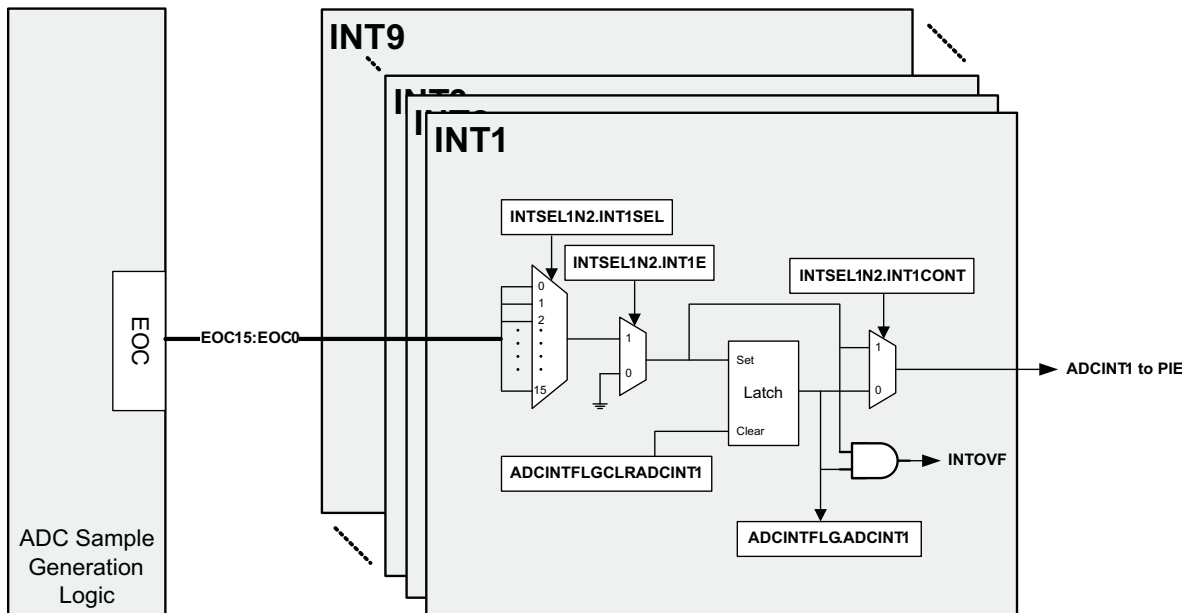
1.6 EOC and Interrupt Operation

Just as there are 16 independent SOCx configuration sets, there are 16 EOCx pulses. In sequential sampling mode, the EOCx is associated directly with the SOCx. In simultaneous sampling mode, the even and the following odd EOCx pair are associated with the even and the following odd SOCx pair, as described in [Section 1.5](#). Depending on the ADCCTL1.INTPULSEPOS setting, the EOCx pulse will occur either at the beginning of a conversion or the end. See section 1.11 for exact timings on the EOCx pulses.

The ADC contains 9 interrupts that can be flagged and/or passed on to the PIE. Each of these interrupts can be configured to accept any of the available EOCx signals as its source. The configuration of which EOCx is the source is done in the INTSELxNy registers. Additionally, the ADCINT1 and ADCINT2 signals can be configured to generate an SOCx trigger. This is beneficial to creating a continuous stream of conversions.

Figure 6 shows a block diagram of the interrupt structure of the ADC.

Figure 7. Interrupt Structure



1.7 Power Up Sequence

The ADC resets to the ADC off state. Before writing to any of the ADC registers the ADCENCLK bit in the PCLKCR0 register must be set. For a description of the PCLKCR0 register see the TMS320F2802x Piccolo System Control Reference Guide (SPRUFN3). When powering up the ADC, use the following sequence:

1. If an external reference is desired, enable this mode using bit 3 (ADCREFSSEL) in the ADCCTL1 register.
2. Power up the reference, bandgap, and analog circuits together by setting bits 7-5 (ADCPWDN, ADCBGPWD, ADCREFPWD) in the ADCCTL1 register.
3. Enable the ADC by setting bit 14 (ADCENABLE) of the ADCCTL1 register.
4. Before performing the first conversion, a delay of 1 millisecond after step 2 is required.

Alternatively, steps 1 through 3 can be performed simultaneously.

When powering down the ADC, all three bits in step 2 can be cleared simultaneously. The ADC power levels must be controlled via software and they are independent of the state of the device power modes.

NOTE: This type ADC requires a 1ms delay after all of the circuits are powered up. This differs from the previous type ADC's.

1.8 ADC Calibration

Inherent in any converter is a zero offset error and a full scale gain error. The ADC is factory calibrated at 25-degrees Celsius to correct both of these while allowing the user to modify the offset correction for any application environmental effects, such as the ambient temperature. Except under certain emulation conditions, or unless a modification from the factory settings is desired, the user is not required to perform any specific action. The ADC will be properly calibrated during the device boot process.

1.8.1 Factory Settings and Calibration Function

During the fabrication and test process Texas Instruments calibrates several ADC settings along with a couple of internal oscillator settings. These settings are embedded into the TI reserved OTP memory as part of a C-callable function named `Device_cal()`. Called during the startup boot procedure in the Boot ROM this function writes the factory settings into their respective active registers. Until this occurs, the ADC and the internal oscillators will not adhere to their specified parameters. If the boot process is skipped during emulation, the user must ensure the trim settings are written to their respective registers to ensure the ADC and the internal oscillators meet the specifications in the datasheet. This can be done either by calling this function manually or in the application itself, or by a direct write via CCS. A gel function is provided as part of the *C2802x C/C++ Header Files and Peripheral Examples* ([SPRC823](#)) to accomplish this.

For more information on the `Device_cal()` function refer to the *TMS320x2802x Boot ROM Reference Guide* ([SPRUFN6](#)).

Texas Instruments cannot guarantee the parameters specified in the datasheet if a value other than the factory settings contained in the TI reserved OTP memory is written into the ADC trim registers.

1.8.2 ADC Zero Offset Calibration

Zero offset error is defined as the resultant digital value that occurs when converting a voltage at VREFLO. This base error affects all conversions of the ADC and together with the full scale gain and linearity specifications, determine the DC accuracy of a converter. The zero offset error can be positive, meaning that a positive digital value is output when VREFLO is presented, or negative, meaning that a voltage higher than a one step above VREFLO still reads as a digital zero value. To correct this error, the two's complement of the error is written into the ADCOFFTRIM register. The value contained in this register will be applied before the results are available in the ADC result registers. This operation is fully contained within the ADC core, so the timing for the results will not be affected and the full dynamic range of the ADC will be maintained for any trim value. Calling the `Device_cal()` function writes the ADCOFFTRIM register with the factory calibrated offset error correction, but the user can modify the ADCOFFTRIM register to compensate for additional offset error induced by the application environment. This can be done without sacrificing an ADC channel by using the VREFLOCONV bit in the ADCCTRL1 register.

Use the following procedure to re-calibrate the ADC offset:

1. **Set ADCOFFTRIM to 80 (50h).** This adds an artificial offset to account for negative offset that may reside in the ADC core.
2. **Set ADCCTL1.VREFLOCONV to 1.** This internally connects VREFLO to input channel B5. See the [ADCCTL1](#) register description for more details.
3. **Perform multiple conversions on B5 (i.e. sample VREFLO) and take an average to account for board noise.** See [Section 1.3](#) on how to setup and initiate the ADC to sample B5.
4. **Set ADCOFFTRIM to 80 (50h) minus the average obtained in step 3.** This removes the artificial offset from step 1 and creates a two's compliment of the offset error.
5. **Set ADCCTL1.VREFLOCONV to 0.** This connects B5 back to the external ADCINB5 input pin.

NOTE: The "AdcOffsetSelfCal()" function located in `DSP2802x(3x)_Adc.c` in the common header files performs these steps.

1.8.3 ADC Full Scale Gain Calibration

Gain error occurs as an incremental error as the voltage input is increased. Full scale gain error occurs at the maximum input voltage. As in offset error, gain error can be positive or negative. A positive full scale gain error means that the full scale digital result is reached before the maximum voltage is input. A negative full scale error implies that the full digital result will never be achieved. The calibration function `Device_cal()` writes a factory trim value to correct the ADC full scale gain error into the ADCREFTRIM register. This register should not be modified after the `Device_cal()` function is called.

1.8.4 ADC Bias Current Calibration

To further increase the accuracy of the ADC, the calibration function Device_cal() also writes a factory trim value to an ADC register for the ADC bias currents. This register should not be modified after the Device_cal() function is called.

1.9 Internal/External Reference Voltage Selection

1.9.1 Internal Reference Voltage

The ADC can operate in two different reference modes, selected by the ADCCTL1.ADCREFSEL bit. By default the internal bandgap is chosen to generate the reference voltage for the ADC. This will convert the voltage presented according to a fixed scale 0 to 3.3v range. The equation governing conversions in this mode is:

Digital Value = 0	when Input ≤ 0v
Digital Value = 4096 [(Input – VREFLO)/3.3v]	when 0v < Input < 3.3v
Digital Value = 4095,	when Input ≥ 3.3v

*All fractional values are truncated

**VREFLO must be tied to ground in this mode. This is done internally on some devices.

1.9.2 External Reference Voltage

To convert the voltage presented as a ratiometric signal, the external VREFHI/VREFLO pins should be chosen to generate the reference voltage. In contrast with the fixed 0 to 3.3v input range of the internal bandgap mode, the ratiometric mode has an input range from VREFLO to VREFHI. Converted values will scale to this range. For instance, if VREFLO is set to 0.5v and VREFHI is 3.0v, a voltage of 1.75v will be converted to the digital result of 2048. See the device datasheet for the allowable ranges of VREFLO and VREFHI. On some devices VREFLO is tied to ground internally, and hence limited to 0v. The equation governing the conversions in this mode is:

Digital Value = 0	when Input ≤ VREFLO
Digital Value = 4096 [(Input – VREFLO)/(VREFHI – VREFLO)]	when VREFLO < Input < VREFHI
Digital Value = 4095,	when Input ≥ VREFHI

*All fractional values are truncated

1.10 ADC Registers

This section contains the ADC registers and bit definitions with the registers grouped by function. All of the ADC registers are located in Peripheral Frame 2 except the ADCRESULTx registers, which are found in Peripheral Frame 0. See the device datasheet for specific addresses.

Table 2. ADC Configuration & Control Registers (AdcRegs and AdcResult):

Register Name	Address Offset	Size (x16)	Description
ADCCTL1	0x00	1	Control 1 Register ⁽¹⁾
ADCCTL2	0x01	1	Control 2 Register ⁽¹⁾
ADCINTFLG	0x04	1	Interrupt Flag Register
ADCINTFLGCLR	0x05	1	Interrupt Flag Clear Register
ADCINTOVF	0x06	1	Interrupt Overflow Register
ADCINTOVFCLR	0x07	1	Interrupt Overflow Clear Register
INTSEL1N2	0x08	1	Interrupt 1 and 2 Selection Register ⁽¹⁾
INTSEL3N4	0x09	1	Interrupt 3 and 4 Selection Register ⁽¹⁾
INTSEL5N6	0x0A	1	Interrupt 5 and 6 Selection Register ⁽¹⁾
INTSEL7N8	0x0B	1	Interrupt 7 and 8 Selection Register ⁽¹⁾
INTSEL9N10	0x0C	1	Interrupt 9 Selection Register (reserved Interrupt 10 Selection) ⁽¹⁾
SOCPRCTL	0x10	1	SOC Priority Control Register ⁽¹⁾
ADCSAMPLEMODE	0x12	1	Sampling Mode Register ⁽¹⁾
ADCINTSOCSEL1	0x14	1	Interrupt SOC Selection 1 Register (for 8 channels) ⁽¹⁾
ADCINTSOCSEL2	0x15	1	Interrupt SOC Selection 2 Register (for 8 channels) ⁽¹⁾
ADCSOCFLG1	0x18	1	SOC Flag 1 Register (for 16 channels)
ADCSOCFRC1	0x1A	1	SOC Force 1 Register (for 16 channels)
ADCSOCOVF1	0x1C	1	SOC Overflow 1 Register (for 16 channels)
ADCSOCOVFCLR1	0x1E	1	SOC Overflow Clear 1 Register (for 16 channels)
ADCSOC0CTL - ADCSOC15CTL	0x20 - 0x2F	1	SOC0 Control Register to SOC15 Control Register ⁽¹⁾
ADCREFRIM	0x40	1	Reference Trim Register ⁽¹⁾
ADCOFFTRIM	0x41	1	Offset Trim Register ⁽¹⁾
COMPHYSTCTL	0x4C	1	Comparator Hysteresis Control Register ⁽¹⁾
ADCREV – reserved	0x4F	1	Revision Register
ADCRESULT0 - ADCRESULT15	0x00 - 0x0F ⁽²⁾	1	ADC Result 0 Register to ADC Result 15 Register

⁽¹⁾ This register is EALLOW protected.

⁽²⁾ The base address of the ADCRESULT registers differs from the base address of the other ADC registers. In the header files, the ADCRESULT registers are found in the AdcResult register file, not AdcRegs.

1.10.1 ADC Control Register 1 (ADCCTL1)

NOTE: The following ADC Control Register is EALLOW protected.

Figure 8. ADC Control Register 1 (ADCCTL1) (Address Offset 00h)

15	14	13	12					8
RESET	ADCENABLE	ADCBSY	ADCBSYCHN					
R-0/W-1	R/W-0	R-0	R-0					
7	6	5	4	3	2	1	0	
ADCPWN	ADCBGPWD	ADCREFPWD	Reserved	ADCREFSEL	INTPULSEPOS	VREFLO CONV	TEMPCONV	
R/W-0	R/W-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; R-0/W-1 = always read as 0, write 1 to set; -n = value after reset

Table 3. ADC Control Register 1 (ADCCTL1) Field Descriptions

Bit	Field	Value	Description
15	RESET	0 1	<p>ADC module software reset. This bit causes a master reset on the entire ADC module. All register bits and state machines are reset to the initial state as occurs when the device reset pin is pulled low (or after a power-on reset). This is a one-time-effect bit, meaning this bit is self-cleared immediately after it is set to 1. Read of this bit always returns a 0. Also, the reset of ADC has a latency of two clock cycles (that is, other ADC control register bits should not be modified until two clock cycles after the instruction that resets the ADC.</p> <p>0 no effect 1 Resets entire ADC module (bit is then set back to 0 by ADC logic)</p> <p>Note: The ADC module is reset during a system reset. If an ADC module reset is desired at any other time, you can do so by writing a 1 to this bit. After two clock cycles, you can then write the appropriate values to the ADCCTL1 register bits. Assembly code: MOV ADCCTL1, #1xxxxxxxxxxxxb ; Resets the ADC (RESET = 1) NOP ; Delay two cycles NOP MOV ADCCTL1, #0xxxxxxxxxxxxb ; Set to user-desired value</p> <p>Note: The second MOV is not required if the default configuration is sufficient.</p>
14	ADCENABLE	0 1	<p>ADC Enable</p> <p>0 ADC disabled (does not power down ADC) 1 ADC Enabled. Musts set before an ADC conversion (recommend that it be set directly after setting ADC power-up bits)</p>
13	ADCBSY	0 1	<p>ADC Busy</p> <p>Set when ADC SOC is generated, cleared per below. Used by the ADC state machine to determine if ADC is available to sample.</p> <p>Sequential Mode: Cleared 4 ADC clocks after negative edge of S/H pulse Simultaneous Mode: Cleared 14 ADC clocks after negative edge of S/H pulse</p> <p>0 ADC is available to sample next channel 1 ADC is busy and cannot sample another channel</p>
12-8	ADCBSYCHN	00h 01h 02h 03h 04h 05h 06h 07h 08h 09h 0Ah 0Bh 0Ch 0Dh 0Eh 0Fh 1xh	<p>Set when ADC SOC for current channel is generated</p> <p>When ADCBSY = 0: holds the value of the last converted channel When ADCBSY = 1: reflects channel currently being processed</p> <p>00h ADCINA0 is currently processing or was last channel converted 01h ADCINA1 is currently processing or was last channel converted 02h ADCINA2 is currently processing or was last channel converted 03h ADCINA3 is currently processing or was last channel converted 04h ADCINA4 is currently processing or was last channel converted 05h ADCINA5 is currently processing or was last channel converted 06h ADCINA6 is currently processing or was last channel converted 07h ADCINA7 is currently processing or was last channel converted 08h ADCINB0 is currently processing or was last channel converted 09h ADCINB1 is currently processing or was last channel converted 0Ah ADCINB2 is currently processing or was last channel converted 0Bh ADCINB3 is currently processing or was last channel converted 0Ch ADCINB4 is currently processing or was last channel converted 0Dh ADCINB5 is currently processing or was last channel converted 0Eh ADCINB6 is currently processing or was last channel converted 0Fh ADCINB7 is currently processing or was last channel converted 1xh Invalid value</p>

Table 3. ADC Control Register 1 (ADCCTL1) Field Descriptions (continued)

Bit	Field	Value	Description
7	ADCPWDN	0 1	ADC power down (active low). This bit controls the power up and power down of all the analog circuitry inside the analog core except the bandgap and reference circuitry All analog circuitry inside the core except the bandgap and reference circuitry is powered down The analog circuitry inside the core is powered up
6	ADCBGPWD	0 1	Bandgap circuit power down (active low) Bandgap circuitry is powered down Bandgap buffer's circuitry inside core is powered up
5	ADCREFPWD	0 1	Reference buffers circuit power down (active low) Reference buffers circuitry is powered down Reference buffers circuitry inside the core is powered up
4	Reserved	0	Reads return a zero; Writes have no effect.
3	ADCREFSEL	0 1	Internal/external reference select Internal Bandgap used for reference generation External VREFHI/VREFLO pins used for reference generation. On some devices the VREFHI pin is shared with ADCINA0. In this case ADCINA0 will not be available for conversions in this mode. On some devices the VREFLO pin is shared with VSSA. In this case the VREFLO voltage cannot be varied.
2	INTPULSEPOS	0 1	INT Pulse Generation control INT pulse generation occurs when ADC begins conversion (neg edge of sample pulse of the sampled signal) INT pulse generation occurs 1 cycle prior to ADC result latching into its result register
1	VREFLOCONV	0 1	VREFLO Convert. When enabled, internally connects VREFLO to the ADC channel B5 and disconnects the ADCINB5 pin from the ADC. Whether the pin ADCINB5 exists on the device does not affect this function. Any external circuitry on the ADCINB5 pin is unaffected by this mode. ADCINB5 is passed to the ADC module as normal, VREFLO connection to ADCINB5 is disabled VREFLO internally connected to the ADC for sampling
0	TEMPCONV	0 1	Temperature sensor convert. When enabled internally connects the internal temperature sensor to ADC channel A5 and disconnects the ADCINA5 pin from the ADC. Whether the pin ADCINA5 exists on the device does not affect this function. Any external circuitry on the ADCINA5 pin is unaffected by this mode ADCINA5 is passed to the ADC module as normal, internal temperature sensor connection to ADCINA5 is disabled. Temperature sensor is internally connected to the ADC for sampling

1.10.2 ADC Control Register 2 (ADCCTL2)

NOTE: The following ADC Control Register is EALLOW protected.

Figure 9. ADC Control Register 2 (ADCCTL2) (Address Offset 01h)

15	2	1	0
Reserved	ADCNONOVERLAP	CLKDIV2EN	
R-0	R-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 4. ADC Control Register 2 (ADCCTL2) Field Descriptions

Bit	Field	Value	Description
15-2	Reserved	0	Reads return a zero; writes have no effect.
1	ADCNONOVERLAP	0 1	ADCNONOVERLAP control bit Overlap of sample and conversion is allowed Overlap of sample is not allowed
0	CLKDIV2EN	0 1	When enabled, divides the ADC input clock by 2. When running /2 ADCCLK, scale the minimum sample duration accordingly to meet 116.6ns for better throughput. ADC clock = CPU clock ADC clock = CPU clock/2

1.10.3 ADC Interrupt Registers

Figure 10. ADC Interrupt Flag Register (ADCINTFLG) (Address Offset 04h)

15							9	8
Reserved							ADCINT9	
R-0							R-0	
7	6	5	4	3	2	1	0	
ADCINT8	ADCINT7	ADCINT6	ADCINT5	ADCINT4	ADCINT3	ADCINT2	ADCINT1	
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 5. ADC Interrupt Flag Register (ADCINTFLG) Field Descriptions

Bit	Field	Value	Description
15-9	Reserved	0	Reads return a zero; Writes have no effect.
8-0	ADCINT _x (x = 9 to 1)	0 1	ADC Interrupt Flag Bits: Reading this bit indicates if an ADCINT pulse was generated No ADC interrupt pulse generated ADC Interrupt pulse generated If the ADC interrupt is placed in continuous mode (INTSEL _x N _y register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register.

Figure 11. ADC Interrupt Flag Clear Register (ADCINTFLGCLR) (Address Offset 05h)

15							9	8
Reserved							ADCINT9	
R-0							W1C-0	
7	6	5	4	3	2	1	0	
ADCINT8	ADCINT7	ADCINT6	ADCINT5	ADCINT4	ADCINT3	ADCINT2	ADCINT1	
W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	

LEGEND: W1C = Write 1 to clear bit, reads return 0; R = Read only; -n = value after reset

Table 6. ADC Interrupt Flag Clear Register (ADCINTFLGCLR) Field Descriptions

Bit	Field	Value	Description
15-9	Reserved	0	Reads return a zero; Writes have no effect.
8-0	ADCINTx (x = 9 to 1)	0 1	ADC interrupt Flag Clear Bit; Reads return 0 No action. If the ADC interrupt is placed in continuous mode (INTSELxNy register) then further interrupt pulses are generated whenever a selected EOC event occurs even if the flag bit is set. If the continuous mode is not enabled, then no further interrupt pulses are generated until the user clears this flag bit using the ADCINTFLGCLR register. Rather, an ADC interrupt overflow event occurs in the ADCINTOVF register. Boundary condition for clearing/setting flag bits: If hardware is trying to set bit while software tries to clear the bit in the same cycle, the following will take place: <ol style="list-style-type: none"> SW has priority, and will clear the flag HW set will be discarded, no signal will propagate to the PIE from the latch Overflow flag/condition will be generated

Figure 12. ADC Interrupt Overflow Register (ADCINTOVF) (Address Offset 06h)

15							9	8
Reserved							ADCINT9	
R-0							R-0	
7	6	5	4	3	2	1	0	
ADCINT8	ADCINT7	ADCINT6	ADCINT5	ADCINT4	ADCINT3	ADCINT2	ADCINT1	
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 7. ADC Interrupt Overflow Register (ADCINTOVF) Field Descriptions

Bit	Field	Value	Description
15-9	Reserved	0	Reserved
8-0	ADCINTx (x = 9 to 1)	0 1	ADC Interrupt Overflow Bits. Indicates if an overflow occurred when generating ADCINT pulses. If the respective ADCINTFLG bit is set and a selected additional EOC trigger is generated, then an overflow condition occurs. No ADC interrupt overflow event detected. ADC Interrupt overflow event detected. The overflow bit does not care about the continuous mode bit state. An overflow condition is generated irrespective of this mode selection.

Figure 13. ADC Interrupt Overflow Clear Register (ADCINTOVFCLR) (Address Offset 07h)

15							9	8
Reserved							ADCINT9	
R-0							W1C-0	
7	6	5	4	3	2	1	0	
ADCINT8	ADCINT7	ADCINT6	ADCINT5	ADCINT4	ADCINT3	ADCINT2	ADCINT1	
W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	

LEGEND: W1C = Write 1 to clear bit, reads return 0; R = Read only; -n = value after reset

Table 8. ADC Interrupt Overflow Clear Register (ADCINTOVFCLR) Field Descriptions

Bit	Field	Value	Description
15-9	Reserved	0	Reads return a zero; Writes have no effect.
8-0	ADCINTx (x = 9 to 1)	0 1	ADC Interrupt Overflow Clear Bits; Reads return 0 No action. Clears the respective overflow bit in the ADCINTOVF register. If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCINTOVF register, then hardware has priority and the ADCINTOVF bit will be set.

NOTE: The following Interrupt Select Registers are EALLOW protected.

Figure 14. Interrupt Select 1 And 2 Register (INTSEL1N2) (Address Offset 08h)

15	14	13	12	8
Reserved	INT2CONT	INT2E	INT2SEL	
R-0	R/W-0	R/W-0	R/W-0	
7	6	5	4	0
Reserved	INT1CONT	INT1E	INT1SEL	
R-0	R/W-0	R/W-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Figure 15. Interrupt Select 3 And 4 Register (INTSEL3N4) (Address Offset 09h)

15	14	13	12	8
Reserved	INT4CONT	INT4E	INT4SEL	
R-0	R/W-0	R/W-0	R/W-0	
7	6	5	4	0
Reserved	INT3CONT	INT3E	INT3SEL	
R-0	R/W-0	R/W-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Figure 16. Interrupt Select 5 And 6 Register (INTSEL5N6) (Address Offset 0Ah)

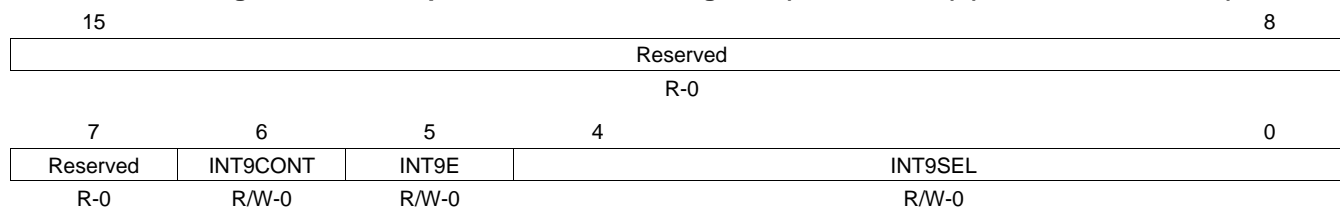
15	14	13	12	8
Reserved	INT6CONT	INT6E	INT6SEL	
R-0	R/W-0	R/W-0	R/W-0	
7	6	5	4	0
Reserved	INT5CONT	INT5E	INT5SEL	
R-0	R/W-0	R/W-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Figure 17. Interrupt Select 7 And 8 Register (INTSEL7N8) (Address Offset 0Bh)

15	14	13	12	8
Reserved	INT8CONT	INT8E	INT8SEL	
R-0	R/W-0	R/W-0	R/W-0	
7	6	5	4	0
Reserved	INT7CONT	INT7E	INT7SEL	
R-0	R/W-0	R/W-0	R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Figure 18. Interrupt Select 9 And 10 Register (INTSEL9N10) (Address Offset 0Ch)


LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 9. INTSELxNy Register Field Descriptions

Bit	Field	Value	Description
15	Reserved	0	Reserved
14	INTyCONT	0	ADCINTy Continuous Mode Enable No further ADCINTy pulses are generated until ADCINTy flag (in ADCINTFLG register) is cleared by user.
		1	ADCINTy pulses are generated whenever an EOC pulse is generated irrespective if the flag bit is cleared or not.
13	INTyE	0	ADCINTy Interrupt Enable ADCINTy is disabled.
		1	ADCINTy is enabled.
12-8	INTySEL	00h	ADCINTy EOC Source Select EOC0 is trigger for ADCINTy
		01h	EOC1 is trigger for ADCINTy
		02h	EOC2 is trigger for ADCINTy
		03h	EOC3 is trigger for ADCINTy
		04h	EOC4 is trigger for ADCINTy
		05h	EOC5 is trigger for ADCINTy
		06h	EOC6 is trigger for ADCINTy
		07h	EOC7 is trigger for ADCINTy
		08h	EOC8 is trigger for ADCINTy
		09h	EOC9 is trigger for ADCINTy
		0Ah	EOC10 is trigger for ADCINTy
		0Bh	EOC11 is trigger for ADCINTy
		0Ch	EOC12 is trigger for ADCINTy
		0Dh	EOC13 is trigger for ADCINTy
		0Eh	EOC14 is trigger for ADCINTy
		0Fh	EOC15 is trigger for ADCINTy
		1xh	Invalid value.
7	Reserved	0	Reads return a zero; Writes have no effect.
6	INTxCONT	0	ADCINTx Continuous Mode Enable. No further ADCINTx pulses are generated until ADCINTx flag (in ADCINTFLG register) is cleared by user.
		1	ADCINTx pulses are generated whenever an EOC pulse is generated irrespective if the flag bit is cleared or not.
5	INTxE	0	ADCINTx Interrupt Enable ADCINTx is disabled.
		1	ADCINTx is enabled .

Table 9. INTSELxNy Register Field Descriptions (continued)

Bit	Field	Value	Description
4-0	INTxSEL		ADCINTx EOC Source Select
		00h	EOC0 is trigger for ADCINTx
		01h	EOC1 is trigger for ADCINTx
		02h	EOC2 is trigger for ADCINTx
		03h	EOC3 is trigger for ADCINTx
		04h	EOC4 is trigger for ADCINTx
		05h	EOC5 is trigger for ADCINTx
		06h	EOC6 is trigger for ADCINTx
		07h	EOC7 is trigger for ADCINTx
		08h	EOC8 is trigger for ADCINTx
		09h	EOC9 is trigger for ADCINTx
		0Ah	EOC10 is trigger for ADCINTx
		0Bh	EOC11 is trigger for ADCINTx
		0Ch	EOC12 is trigger for ADCINTx
		0Dh	EOC13 is trigger for ADCINTx
		0Eh	EOC14 is trigger for ADCINTx
0Fh	EOC15 is trigger for ADCINTx		
	1xh	Invalid value.	

1.10.4 ADC Priority Register

NOTE: The following SOC Priority Control Register is EALLOW protected.

Figure 19. ADC Start of Conversion Priority Control Register (SOCPRCTL)


LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 10. SOCPRICTL Register Field Descriptions

Bit	Field	Value	Description
15-11	Reserved		Reads return a zero; Writes have no effect.
10-5	RRPOINTER	00h SOC0 was last round robin SOC to convert. SOC1 is highest round robin priority. 01h SOC1 was last round robin SOC to convert. SOC2 is highest round robin priority. 02h SOC2 was last round robin SOC to convert. SOC3 is highest round robin priority. 03h SOC3 was last round robin SOC to convert. SOC4 is highest round robin priority. 04h SOC4 was last round robin SOC to convert. SOC5 is highest round robin priority. 05h SOC5 was last round robin SOC to convert. SOC6 is highest round robin priority. 06h SOC6 was last round robin SOC to convert. SOC7 is highest round robin priority. 07h SOC7 was last round robin SOC to convert. SOC8 is highest round robin priority. 08h SOC8 was last round robin SOC to convert. SOC9 is highest round robin priority. 09h SOC9 was last round robin SOC to convert. SOC10 is highest round robin priority. 0Ah SOC10 was last round robin SOC to convert. SOC11 is highest round robin priority. 0Bh SOC11 was last round robin SOC to convert. SOC12 is highest round robin priority. 0Ch SOC12 was last round robin SOC to convert. SOC13 is highest round robin priority. 0Dh SOC13 was last round robin SOC to convert. SOC14 is highest round robin priority. 0Eh SOC14 was last round robin SOC to convert. SOC15 is highest round robin priority. 0Fh SOC15 was last round robin SOC to convert. SOC0 is highest round robin priority. 1xh Invalid value 20h Reset value to indicate no SOC has been converted. SOC0 is highest round robin priority. Set to this value when the device is reset, when the ADCCTL1.RESET bit is set, or when the SOCPRICTL register is written. In the latter case, if a conversion is currently in progress, it will complete and then the new priority will take effect. Others Invalid selection.	Round Robin Pointer. Holds the value of the last converted round robin SOCx to be used by the round robin scheme to determine order of conversions.
4-0	SOCPRIORITY	00h SOC priority is handled in round robin mode for all channels. 01h SOC0 is high priority, rest of channels are in round robin mode. 02h SOC0-SOC1 are high priority, SOC2-SOC15 are in round robin mode. 03h SOC0-SOC2 are high priority, SOC3-SOC15 are in round robin mode. 04h SOC0-SOC3 are high priority, SOC4-SOC15 are in round robin mode. 05h SOC0-SOC4 are high priority, SOC5-SOC15 are in round robin mode. 06h SOC0-SOC5 are high priority, SOC6-SOC15 are in round robin mode. 07h SOC0-SOC6 are high priority, SOC7-SOC15 are in round robin mode. 08h SOC0-SOC7 are high priority, SOC8-SOC15 are in round robin mode. 09h SOC0-SOC8 are high priority, SOC9-SOC15 are in round robin mode. 0Ah SOC0-SOC9 are high priority, SOC10-SOC15 are in round robin mode. 0Bh SOC0-SOC10 are high priority, SOC11-SOC15 are in round robin mode. 0Ch SOC0-SOC11 are high priority, SOC12-SOC15 are in round robin mode. 0Dh SOC0-SOC12 are high priority, SOC13-SOC15 are in round robin mode. 0Eh SOC0-SOC13 are high priority, SOC14-SOC15 are in round robin mode. 0Fh SOC0-SOC14 are high priority, SOC15 is in round robin mode. 10h All SOCs are in high priority mode, arbitrated by SOC number Others Invalid selection.	SOC Priority. Determines the cutoff point for priority mode and round robin arbitration for SOCx

1.10.5 ADC SOC Registers

NOTE: The following ADC Sample Mode Register is EALLOW protected.

Figure 20. ADC Sample Mode Register (ADCSAMPLEMODE) (Address Offset 12h)

Reserved							
R-0							
7	6	5	4	3	2	1	0
SIMULEN14	SIMULEN12	SIMULEN10	SIMULEN8	SIMULEN6	SIMULEN4	SIMULEN2	SIMULEN0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 11. ADC Sample Mode Register (ADCSAMPLEMODE) Field Descriptions

Bit	Field	Value	Description
15:8	Reserved	0	Reserved
7	SIMULEN14	0	Simultaneous sampling enable for SOC14/SOC15. Couples SOC14 and SOC15 in simultaneous sampling mode. See section 1.5 for details. This bit should not be set when the ADC is actively converting SOC14 or SOC15.
		1	Single sample mode set for SOC14 and SOC15. All bits of CHSEL field define channel to be converted. EOC14 associated with SOC14. EOC15 associated with SOC15. SOC14's result placed in ADCRESULT14 register. SOC15's result placed in ADCRESULT15.
		1	Simultaneous sample for SOC14 and SOC15. Lowest three bits of CHSEL field define the pair of channels to be converted. EOC14 and EOC15 associated with SOC14 and SOC15 pair. SOC14's and SOC15's results will be placed in ADCRESULT14 and ADCRESULT15 registers, respectively.
6	SIMULEN12	0	Simultaneous sampling enable for SOC12/SOC13. Couples SOC12 and SOC13 in simultaneous sampling mode. See section 1.5 for details. This bit should not be set when the ADC is actively converting SOC12 or SOC13.
		1	Single sample mode set for SOC12 and SOC13. All bits of CHSEL field define channel to be converted. EOC12 associated with SOC12. EOC13 associated with SOC13. SOC12's result placed in ADCRESULT12 register. SOC13's result placed in ADCRESULT13.
		1	Simultaneous sample for SOC12 and SOC13. Lowest three bits of CHSEL field define the pair of channels to be converted. EOC12 and EOC13 associated with SOC12 and SOC13 pair. SOC12's and SOC13's results will be placed in ADCRESULT12 and ADCRESULT13 registers, respectively.
5	SIMULEN10	0	Simultaneous sampling enable for SOC10/SOC11. Couples SOC10 and SOC11 in simultaneous sampling mode. See section 1.5 for details. This bit should not be set when the ADC is actively converting SOC10 or SOC11.
		1	Single sample mode set for SOC10 and SOC11. All bits of CHSEL field define channel to be converted. EOC10 associated with SOC10. EOC11 associated with SOC11. SOC10's result placed in ADCRESULT10 register. SOC11's result placed in ADCRESULT11.
		1	Simultaneous sample for SOC10 and SOC11. Lowest three bits of CHSEL field define the pair of channels to be converted. EOC10 and EOC11 associated with SOC10 and SOC11 pair. SOC10's and SOC11's results will be placed in ADCRESULT10 and ADCRESULT11 registers, respectively.
4	SIMULEN8	0	Simultaneous sampling enable for SOC8/SOC9. Couples SOC8 and SOC9 in simultaneous sampling mode. See section 1.5 for details. This bit should not be set when the ADC is actively converting SOC8 or SOC9.
		1	Single sample mode set for SOC8 and SOC9. All bits of CHSEL field define channel to be converted. EOC8 associated with SOC8. EOC9 associated with SOC9. SOC8's result placed in ADCRESULT8 register. SOC9's result placed in ADCRESULT9.
		1	Simultaneous sample for SOC8 and SOC9. Lowest three bits of CHSEL field define the pair of channels to be converted. EOC8 and EOC9 associated with SOC8 and SOC9 pair. SOC8's and SOC9's results will be placed in ADCRESULT8 and ADCRESULT9 registers, respectively.

Table 11. ADC Sample Mode Register (ADCSAMPLEMODE) Field Descriptions (continued)

Bit	Field	Value	Description
3	SIMULEN6	0 1	<p>Simultaneous sampling enable for SOC6/SOC7. Couples SOC6 and SOC7 in simultaneous sampling mode. See section 1.5 for details. This bit should not be set when the ADC is actively converting SOC6 or SOC7.</p> <p>0 Single sample mode set for SOC6 and SOC7. All bits of CHSEL field define channel to be converted. EOC6 associated with SOC6. EOC7 associated with SOC7. SOC6's result placed in ADCRESULT6 register. SOC7's result placed in ADCRESULT7.</p> <p>1 Simultaneous sample for SOC6 and SOC7. Lowest three bits of CHSEL field define the pair of channels to be converted. EOC6 and EOC7 associated with SOC6 and SOC7 pair. SOC6's and SOC7's results will be placed in ADCRESULT6 and ADCRESULT7 registers, respectively.</p>
2	SIMULEN4	0 1	<p>Simultaneous sampling enable for SOC4/SOC5. Couples SOC4 and SOC5 in simultaneous sampling mode. See section 1.5 for details. This bit should not be set when the ADC is actively converting SOC4 or SOC5.</p> <p>0 Single sample mode set for SOC4 and SOC5. All bits of CHSEL field define channel to be converted. EOC4 associated with SOC4. EOC5 associated with SOC5. SOC4's result placed in ADCRESULT4 register. SOC5's result placed in ADCRESULT5.</p> <p>1 Simultaneous sample for SOC4 and SOC5. Lowest three bits of CHSEL field define the pair of channels to be converted. EOC4 and EOC5 associated with SOC4 and SOC5 pair. SOC4's and SOC5's results will be placed in ADCRESULT4 and ADCRESULT5 registers, respectively.</p>
1	SIMULEN2	0 1	<p>Simultaneous sampling enable for SOC2/SOC3. Couples SOC2 and SOC3 in simultaneous sampling mode. See section 1.5 for details. This bit should not be set when the ADC is actively converting SOC2 or SOC3.</p> <p>0 Single sample mode set for SOC2 and SOC3. All bits of CHSEL field define channel to be converted. EOC2 associated with SOC2. EOC3 associated with SOC3. SOC2's result placed in ADCRESULT2 register. SOC3's result placed in ADCRESULT3.</p> <p>1 Simultaneous sample for SOC2 and SOC3. Lowest three bits of CHSEL field define the pair of channels to be converted. EOC2 and EOC3 associated with SOC2 and SOC3 pair. SOC2's and SOC3's results will be placed in ADCRESULT2 and ADCRESULT3 registers, respectively.</p>
0	SIMULEN0	0 1	<p>Simultaneous sampling enable for SOC0/SOC1. Couples SOC0 and SOC1 in simultaneous sampling mode. See section 1.5 for details. This bit should not be set when the ADC is actively converting SOC0 or SOC1.</p> <p>0 Single sample mode set for SOC0 and SOC1. All bits of CHSEL field define channel to be converted. EOC0 associated with SOC0. EOC1 associated with SOC1. SOC0's result placed in ADCRESULT0 register. SOC1's result placed in ADCRESULT1.</p> <p>1 Simultaneous sample for SOC0 and SOC1. Lowest three bits of CHSEL field define the pair of channels to be converted. EOC0 and EOC1 associated with SOC0 and SOC1 pair. SOC0's and SOC1's results will be placed in ADCRESULT0 and ADCRESULT1 registers, respectively.</p>

NOTE: The following ADC Interrupt SOC Select Registers are EALLOW protected.

Figure 21. ADC Interrupt Trigger SOC Select 1 Register (ADCINTSOCSEL1) (Address Offset 14h)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOC7		SOC6		SOC5		SOC4		SOC3		SOC2		SOC1		SOC0	
R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 12. ADC Interrupt Trigger SOC Select 1 Register (ADCINTSOCSEL1) Register Field Descriptions

Bit	Field	Value	Description
15--0	SOCx (x = 7 to 0)	00 01 10 11	<p>SOCx ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOCx. This field overrides the TRIGSEL field in the ADCSOCxCTL register.</p> <p>00 No ADCINT will trigger SOCx. TRIGSEL field determines SOCx trigger.</p> <p>01 ADCINT1 will trigger SOCx. TRIGSEL field is ignored.</p> <p>10 ADCINT2 will trigger SOCx. TRIGSEL field is ignored.</p> <p>11 Invalid selection.</p>

Figure 22. ADC Interrupt Trigger SOC Select 2 Register (ADCINTSOCSEL2) (Address Offset 15h)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOC15	SOC14	SOC13	SOC12	SOC11	SOC10	SOC9	SOC8								
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0								

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 13. ADC Interrupt Trigger SOC Select 2 Register (ADCINTSOCSEL2) Field Descriptions

Bit	Field	Value	Description
15-0	SOCx (x = 15 to 8)		SOCx ADC Interrupt Trigger Select. Selects which, if any, ADCINT triggers SOCx. This field overrides the TRIGSEL field in the ADCSOCxCTL register.
		00	No ADCINT will trigger SOCx. TRIGSEL field determines SOCx trigger.
		01	ADCINT1 will trigger SOCx. TRIGSEL field is ignored.
		10	ADCINT2 will trigger SOCx. TRIGSEL field is ignored.
		11	Invalid selection.

Figure 23. ADC SOC Flag 1 Register (ADCSOCFLG1) (Address Offset 18h)

15	14	13	12	11	10	9	8
SOC15	SOC14	SOC13	SOC12	SOC11	SOC10	SOC9	SOC8
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
7	6	5	4	3	2	1	0
SOC7	SOC6	SOC5	SOC4	SOC3	SOC2	SOC1	SOC0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 14. ADC SOC Flag 1 Register (ADCSOCFLG1) Field Descriptions

Bit	Field	Value	Description
15-0	SOCx (x = 15 to 0)		SOCx Start of Conversion Flag. Indicates the state of individual SOC conversions.
		0	No sample pending for SOCx.
		1	Trigger has been received and sample is pending for SOCx.
			The bit will be automatically cleared when the respective SOCx conversion is started. If contention exists where this bit receives both a request to set and a request to clear on the same cycle, regardless of the source of either, this bit will be set and the request to clear will be ignored. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether this bit was previously set or not.

Figure 24. ADC SOC Force 1 Register (ADCSOCFRC1) (Address Offset 1Ah)

15	14	13	12	11	10	9	8
SOC15	SOC14	SOC13	SOC12	SOC11	SOC10	SOC9	SOC8
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
7	6	5	4	3	2	1	0
SOC7	SOC6	SOC5	SOC4	SOC3	SOC2	SOC1	SOC0
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 15. ADC SOC Force 1 Register (ADCSOFC1) Field Descriptions

Bit	Field	Value	Description
15-0	SOCx (x = 15 to 0)	0 1	<p>SOCx Force Start of Conversion Flag. Writing a 1 will force to 1 the respective SOCx flag bit in the ADCSOCFLG1 register. This can be used to initiate a software initiated conversion. Writes of 0 are ignored.</p> <p>No action.</p> <p>Force SOCx flag bit to 1. This will cause a conversion to start once priority is given to SOCx.</p> <p>If software tries to set this bit on the same clock cycle that hardware tries to clear the SOCx bit in the ADCSOCFLG1 register, then software has priority and the ADCSOCFLG1 bit will be set. In this case the overflow bit in the ADCSOCOVF1 register will not be affected regardless of whether the ADCSOCFLG1 bit was previously set or not.</p>

Figure 25. ADC SOC Overflow 1 Register (ADCSOCOVF1) (Address Offset 1Ch)

15	14	13	12	11	10	9	8
SOC15	SOC14	SOC13	SOC12	SOC11	SOC10	SOC9	SOC8
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
7	6	5	4	3	2	1	0
SOC7	SOC6	SOC5	SOC4	SOC3	SOC2	SOC1	SOC0
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 16. ADC SOC Overflow 1 Register (ADCSOCOVF1) Field Descriptions

Bit	Field	Value	Description
15-0	SOCx (x = 15 to 0)	0 1	<p>SOCx Start of Conversion Overflow Flag. Indicates an SOCx event was generated while an existing SOCx event was already pending.</p> <p>No SOCx event overflow</p> <p>SOCx event overflow</p> <p>An overflow condition does not stop SOCx events from being processed. It simply is an indication that a trigger was missed</p>

Figure 26. ADC SOC Overflow Clear 1 Register (ADCSOCOVFCLR1) (Address Offset 1Eh)

15	14	13	12	11	10	9	8
SOC15	SOC14	SOC13	SOC12	SOC11	SOC10	SOC9	SOC8
W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0
7	6	5	4	3	2	1	0
SOC7	SOC6	SOC5	SOC4	SOC3	SOC2	SOC1	SOC0
W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0	W1C-0

LEGEND: W1C = Write 1 to clear bit, reads return 0; -n = value after reset

Table 17. ADC SOC Overflow Clear 1 Register (ADCSOCOVFCLR1) Field Descriptions

Bit	Field	Value	Description
15-0	SOCx (x = 15 to 0)	0 1	<p>SOCx Clear Start of Conversion Overflow Flag. Writing a 1 will clear the respective SOCx overflow flag in the ADCSOCOVF1 register. Writes of 0 are ignored. Reads return 0.</p> <p>No action.</p> <p>Clear SOCx overflow flag.</p> <p>If software tries to set this bit on the same clock cycle that hardware tries to set the overflow bit in the ADCSOCOVF1 register, then hardware has priority and the ADCSOCOVF1 bit will be set.</p>

NOTE: The following ADC SOC0 - SOC15 Control Registers are EALLOW protected.

Figure 27. ADC SOC0 - SOC15 Control Registers (ADCSOCxCTL) (Address Offset 20h - 2Fh)

15	11	10	9	6	5	0
TRIGSEL		Reserved	CHSEL		ACQPS	
R/W-0		R-0	R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 18. ADC SOC0 - SOC15 Control Registers (ADCSOCxCTL) Register Field Descriptions

Bit	Field	Value	Description
15-11	TRIGSEL		SOCx Trigger Source Select. Configures which trigger will set the respective SOCx flag in the ADCSOCFLG1 register to initiate a conversion to start once priority is given to SOCx. This setting can be overridden by the respective SOCx field in the ADCINTSOCSEL1 or ADCINTSOCSEL2 register.
		00h	ADCTRIG0 - Software only.
		01h	ADCTRIG1 - CPU Timer 0, TINT0n
		02h	ADCTRIG2 - CPU Timer 1, TINT1n
		03h	ADCTRIG3 - CPU Timer 2, TINT2n
		04h	ADCTRIG4 - XINT2, XINT2SOC
		05h	ADCTRIG5 - ePWM1, ADCSOCA
		06h	ADCTRIG6 - ePWM1, ADCSOCB
		07h	ADCTRIG7 - ePWM2, ADCSOCA
		08h	ADCTRIG8 - ePWM2, ADCSOCB
		09h	ADCTRIG9 - ePWM3, ADCSOCA
		0Ah	ADCTRIG10 - ePWM3, ADCSOCB
		0Bh	ADCTRIG11 - ePWM4, ADCSOCA
		0Ch	ADCTRIG12 - ePWM4, ADCSOCB
		0Dh	ADCTRIG13 - ePWM5, ADCSOCA
		0Eh	ADCTRIG14 - ePWM5, ADCSOCB
		0Fh	ADCTRIG15 - ePWM6, ADCSOCA
10h	ADCTRIG16 - ePWM6, ADCSOCB		
11h	ADCTRIG17 - ePWM7, ADCSOCA		
12h	ADCTRIG18 - ePWM7, ADCSOCB		
	Others		Invalid selection.
10	Reserved		Reads return a zero; Writes have no effect.

Table 18. ADC SOC0 - SOC15 Control Registers (ADC SOCxCTL) Register Field Descriptions (continued)

Bit	Field	Value	Description
9-6	CHSEL		SOCx Channel Select. Selects the channel to be converted when SOCx is received by the ADC.
			Sequential Sampling Mode (SIMULENx = 0):
		0h	ADCINA0
		1h	ADCINA1
		2h	ADCINA2
		3h	ADCINA3
		4h	ADCINA4
		5h	ADCINA5
		6h	ADCINA6
		7h	ADCINA7
		8h	ADCINB0
		9h	ADCINB1
		Ah	ADCINB2
		Bh	ADCINB3
		Ch	ADCINB4
		Dh	ADCINB5
		Eh	ADCINB6
		Fh	ADCINB7
			Simultaneous Sampling Mode (SIMULENx = 1):
		0h	ADCINA0/ADCINB0 pair
		1h	ADCINA1/ADCINB1 pair
		2h	ADCINA2/ADCINB2 pair
		3h	ADCINA3/ADCINB3 pair
		4h	ADCINA4/ADCINB4 pair
		5h	ADCINA5/ADCINB5 pair
		6h	ADCINA6/ADCINB6 pair
		7h	ADCINA7/ADCINB7 pair
		8h	Invalid selection.
		9h	Invalid selection.
		Ah	Invalid selection.
		Bh	Invalid selection.
		Ch	Invalid selection.
		Dh	Invalid selection.
		Eh	Invalid selection.
		Fh	Invalid selection.

Table 18. ADC SOC0 - SOC15 Control Registers (ADCSOCxCTL) Register Field Descriptions (continued)

Bit	Field	Value	Description
5-0	ACQPS		SOCx Acquisition Prescale. Controls the sample and hold window for SOCx. Minimum value allowed is 6.
		00h	Invalid selection.
		01h	Invalid selection.
		02h	Invalid selection.
		03h	Invalid selection.
		04h	Invalid selection.
		05h	Invalid selection.
		06h	Sample window is 7 cycles long (6 + 1 clock cycles).
		07h	Sample window is 8 cycles long (7 + 1 clock cycles).
		08h	Sample window is 9 cycles long (8 + 1 clock cycles).
		09h	Sample window is 10 cycles long (9 + 1 clock cycles).
	
3Fh	Sample window is 64 cycles long (63 + 1 clock cycles).		
Other invalid selections: 10h, 11h, 12h, 13h, 14h, 1Dh, 1Eh, 1Fh, 20h, 21h, 2Ah, 2Bh, 2Ch, 2Dh, 2Eh, 37h, 38h, 39h, 3Ah, 3Bh			

1.10.6 ADC Calibration Registers

NOTE: The ADC Calibration Register is EALLOW protected.

Figure 28. ADC Reference/Gain Trim Register (ADCREFTTRIM) (Address Offset 40h)

15	14	13	9	8	5	4	0
Reserved		EXTREF_FINE_TRIM		BG_COARSE_TRIM		BG_FINE_TRIM	
R-0		R/W-0		R/W-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 19. ADC Reference/Gain Trim Register (ADCREFTTRIM) Field Descriptions

Bit	Field	Value	Description
15-14	Reserved		Reads return a zero; Writes have no effect.
13-9	EXTREF_FINE_TRIM		ADC External reference Fine Trim. These bits should not be modified after device boot code loads them with the factory trim setting.
8-5	BG_COARSE_TRIM		ADC Internal Bandgap Fine Trim. These bits should not be modified after device boot code loads them with the factory trim setting.
4-0	BG_FINE_TRIM		ADC Internal Bandgap Coarse Trim. A maximum value of 30 is supported. These bits should not be modified after device boot code loads them with the factory trim setting.

Figure 29. ADC Offset Trim Register (ADCOFFTRIM) (Address Offset 41h)

15	9	8	0
Reserved		OFFTRIM	
R-0		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 20. ADC Offset Trim Register (ADCOFFTRIM) Field Descriptions

Bit	Field	Value	Description
15-9	Reserved		Reads return a zero; Writes have no effect.
8-0	OFFTRIM		ADC Offset Trim. 2's complement of ADC offset. Range is -256 to +255. These bits are loaded by device boot code with a factory trim setting. Modification of this default setting can be made to correct any board induced offset.

1.10.7 Comparator Hysteresis Control Register

NOTE: The Comparator Hysteresis Control register is EALLOW protected.

Figure 30. Comparator Hysteresis Control Register (COMPHYSTCTL) (Address Offset 4Ch)

15	12	11	10	7	6	5	2	1	0
Reserved	COMP3_HYST_DISABLE	Reserved	COMP2_HYST_DISABLE	Reserved	COMP1_HYST_DISABLE	Reserved	Reserved	Reserved	Reserved
R-0	R/W-0	R-0	R/W-0	R-0	R/W-0	R-0	R/W-0	R-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 21. Comparator Hysteresis Control Register (COMPHYSTCTL) Field Descriptions

Bit	Field	Value	Description
15-12	Reserved		Reads return a zero; Writes have no effect.
11	COMP3_HYST_DISABLE	0 1	Hysteresis enable Hysteresis disable
10-7	Reserved		Reserved
6	COMP2_HYST_DISABLE	0 1	Hysteresis enable Hysteresis disable
5-2			
1	COMP1_HYST_DISABLE	0 1	Hysteresis enable Hysteresis disable
0	Reserved		Reserved

1.10.8 ADC Revision Register

Figure 31. ADC Revision Register (ADCREV) (Address Offset 4Fh)

15	8
REV	
R-x	
7	0
TYPE	
R-3h	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

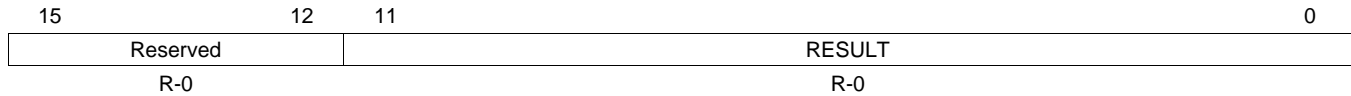
Table 22. ADC Revision Register (ADCREV) Field Descriptions

Bit	Field	Value	Description
15-8	REV		ADC Revision. To allow documentation of differences between revisions. First version is labeled as 00h.
7-0	TYPE	3	ADC Type. Always set to 3 for this type ADC

1.10.9 ADC Result Registers

The ADC Result Registers are found in Peripheral Frame 0 (PF0). In the header files, the ADCRESULTx registers are located in the AdcResult register file, not AdcRegs.

Figure 32. ADC RESULT0 - RESULT15 Registers (ADCRESULTx) (PF1 Block Address Offset 00h - 0Fh)



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 23. ADC RESULT0 - ADCRESULT15 Registers (ADCRESULTx) Field Descriptions

Bit	Field	Value	Description
15-12	Reserved		Reads return a zero; Writes have no effect.
11-0	RESULT		12-bit right-justified ADC result Sequential Sampling Mode (SIMULENx = 0): After the ADC completes a conversion of an SOCx, the digital result is placed in the corresponding ADCRESULTx register. For example, if SOC4 is configured to sample ADCINA1, the completed result of that conversion will be placed in ADCRESULT4. Simultaneous Sampling Mode (SIMULENx = 1): After the ADC completes a conversion of a channel pair, the digital results are found in the corresponding ADCRESULTx and ADCRESULTx+1 registers (assuming x is even). For example, for SOC4, the completed results of those conversions will be placed in ADCRESULT4 and ADCRESULT5. See 1.11 for timings of when this register is written.

1.11 ADC Timings

Figure 33. Timing Example For Sequential Mode / Late Interrupt Pulse

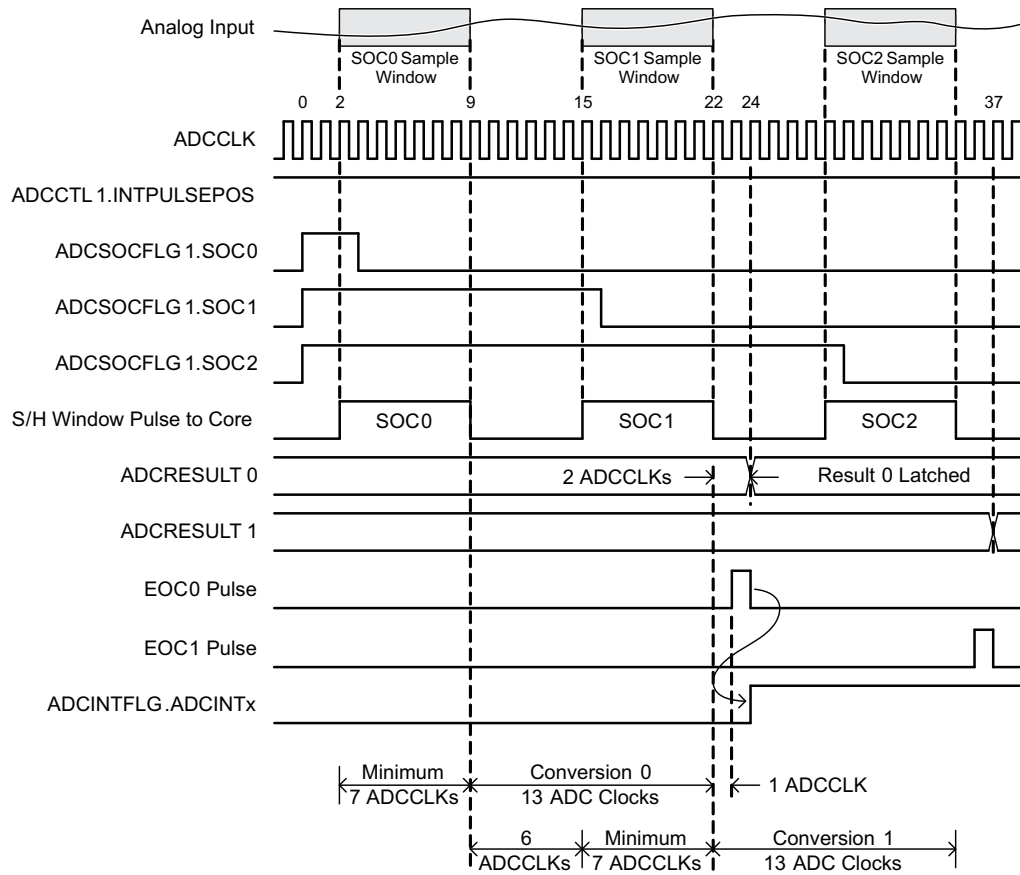


Figure 34. Timing Example For Sequential Mode / Early Interrupt Pulse

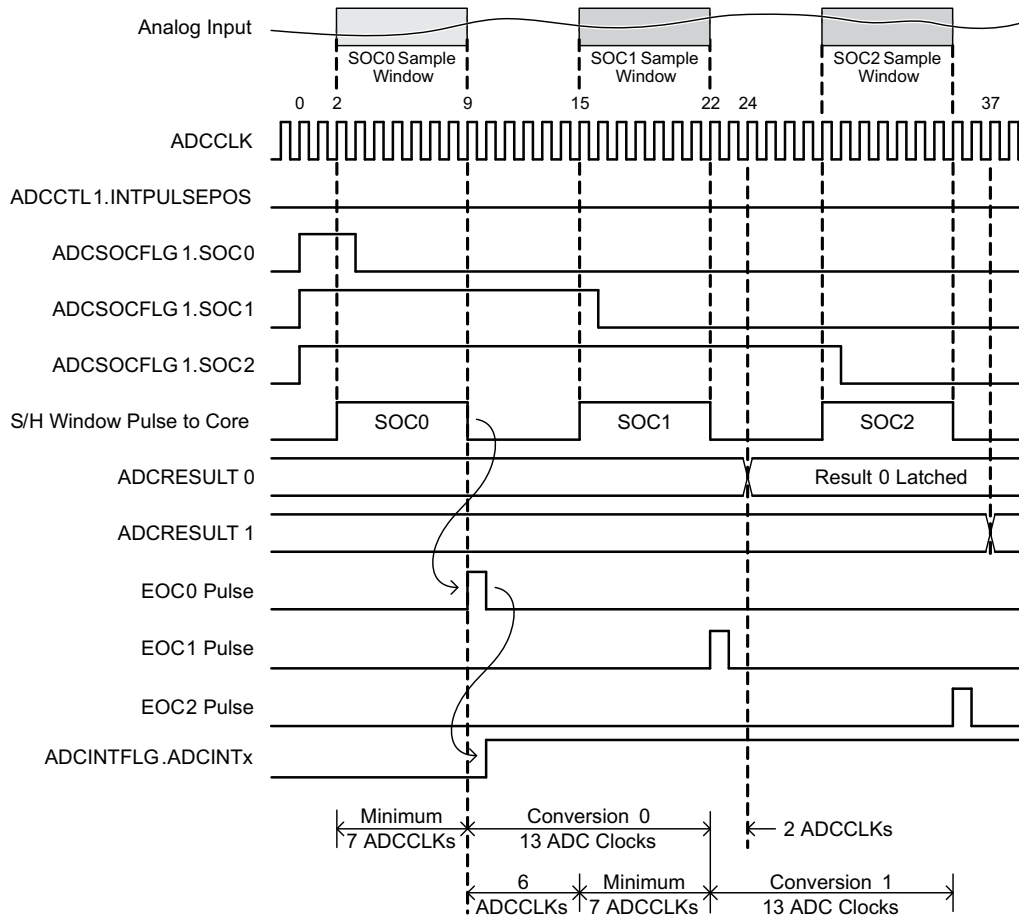


Figure 35. Timing Example For Simultaneous Mode / Late Interrupt Pulse

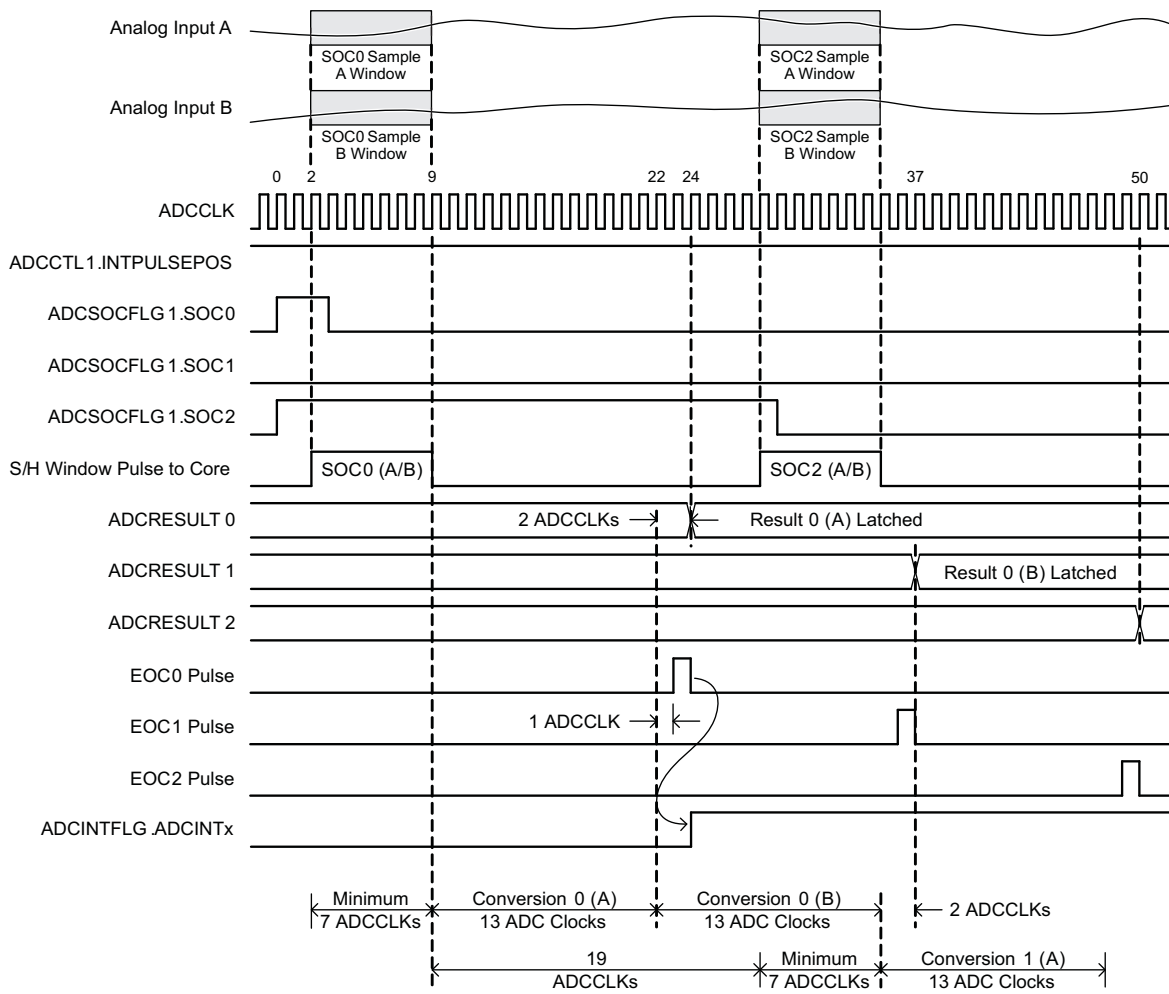


Figure 36. Timing Example For Simultaneous Mode / Early Interrupt Pulse

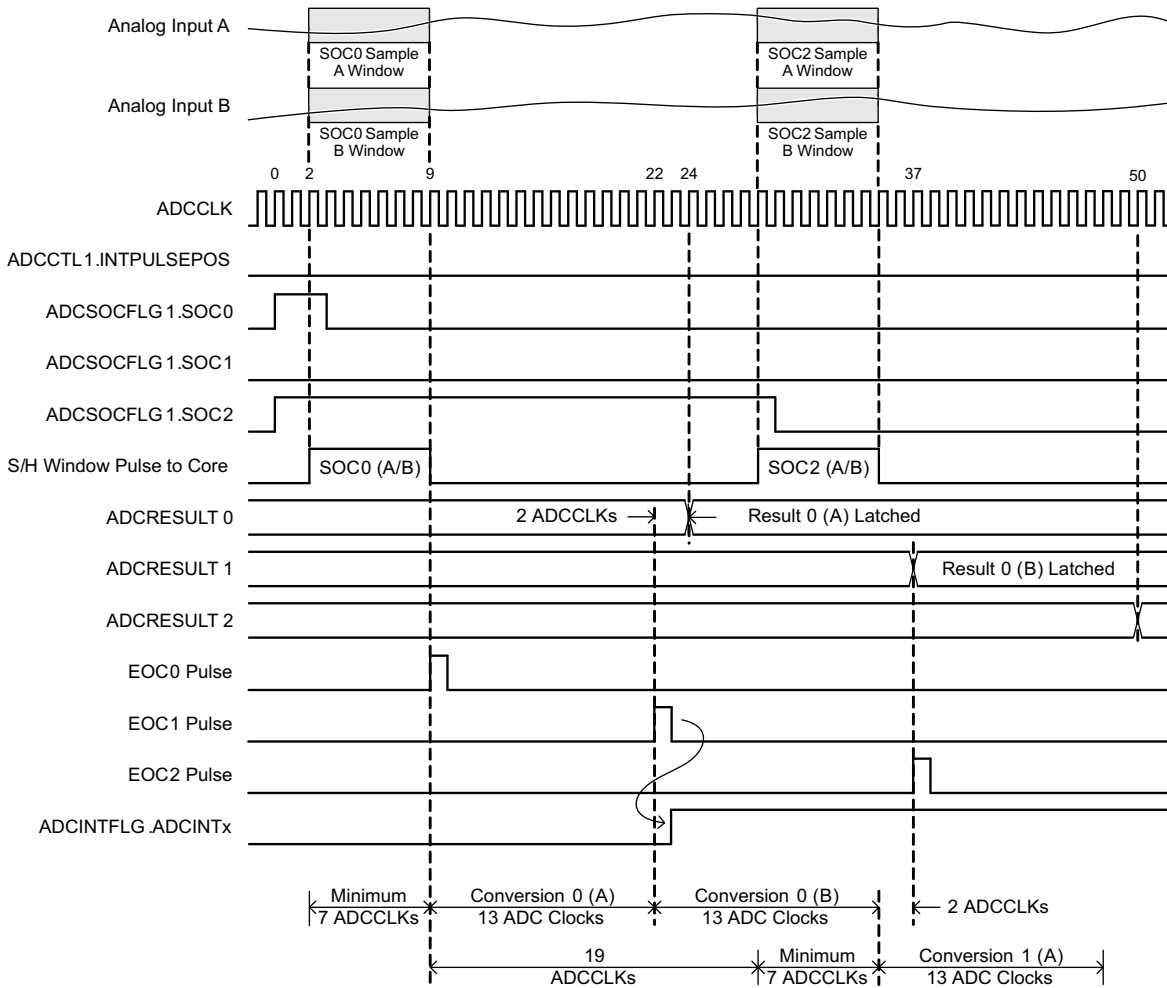
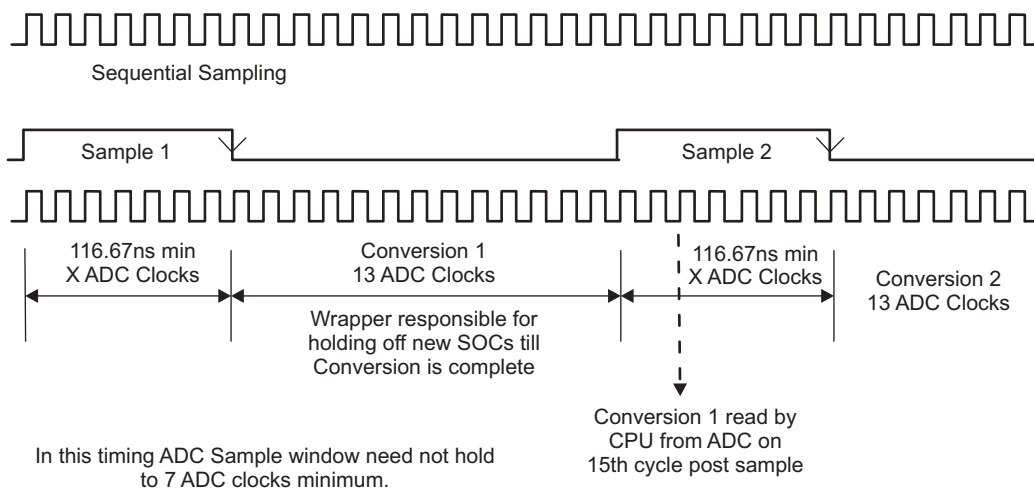


Figure 37. Timing Example for NONOVERLAP Mode



NOTE: The NONOVERLAP bit in the ADCCTL2 register, when enabled, removes the overlap of sampling and conversion stages. This will eliminate 1st sample issue and improve INL/DNL performance.

1.12 Internal Temperature Sensor

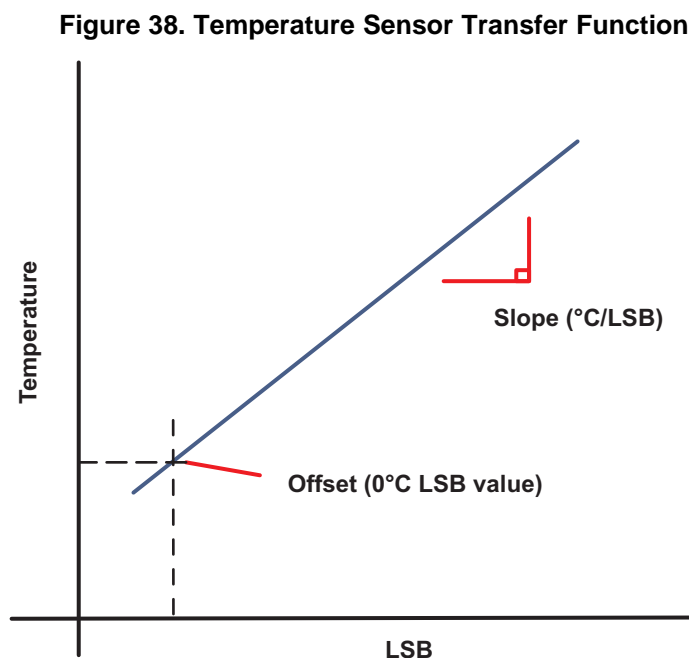
The internal temperature sensor measures the junction temperature of the device. The sensor output can be sampled with the ADC on channel A5 using a switch controlled by the ADCCTL1.TEMP_CONV bit. The switch allows A5 to be used both as an external ADC input pin and the temperature sensor access point. When sampling the temperature sensor, the external circuitry on ADCINA5 has no effect on the sample. Refer to [Section 1.10.1](#) for information about switching between the external ADCINA5 input pin and the internal temperature sensor.

1.12.1 Transfer Function

The temperature sensor output and the resulting ADC values increase with increasing junction temperature. The offset is defined as the 0 °C LSB crossing as illustrated in [Figure 38](#). This information can be used to convert the ADC sensor sample into a temperature unit.

The transfer function to determine a temperature is defined as:

$$\text{Temperature} = (\text{sensor} - \text{Offset}) * \text{Slope}$$



Refer to the electrical characteristics section in *TMS320F28020*, *TMS320F28021*, *TMS320F28022*, *TMS320F28023*, *TMS320F28026*, *TMS320F28027 Piccolo Microcontrollers Data Manual* ([SPRS523](#)) for the slope and offset, or use the stored slope and offset calibrated per device in the factory which can be extracted by a function at the following locations.

For F2802x:

- 0x3D7E80 - Slope (°C / LSB, fixed-point Q15 format)
- 0x3D7E83 - Offset (0 °C LSB value)

For F2803x:

- 0x3D7E82 - Slope (°C / LSB, fixed-point Q15 format)
- 0x3D7E85 - Offset (0 °C LSB value)

The values listed are assuming a 3.3v full scale range. Using the internal reference mode automatically achieves this fixed range, but if using the external mode, the temperature sensor values must be adjusted accordingly to the external reference voltages.

Example

The header files include an example project to easily sample the temperature sensor and convert the result into two different temperature units. There are three steps to using the temperature sensor:

1. Configure the ADC to sample the temperature sensor
2. Sample the temperature sensor
3. Convert the result into a temperature unit, such as °C.

Here is an example of these steps:

```
// Configure the ADC to sample the temperature sensor
EALLOW;
AdcRegs.ADCCTL1.bit.EMPONV = 1; //Connect A5 - temp sensor
AdcRegs.ADCSOC0CTL.bit.CHSEL = 5; //Set SOC0 to sample A5
AdcRegs.ADCSOC1CTL.bit.CHSEL = 5; //Set SOC1 to sample A5
AdcRegs.ADCSOC0CTL.bit.ACQPS = 6; //Set SOC0 ACQPS to 7 ADCCLK
AdcRegs.ADCSOC1CTL.bit.ACQPS = 6; //Set SOC1 ACQPS to 7 ADCCLK
AdcRegs.INTSEL1N2.bit.INT1SEL = 1; //Connect ADCINT1 to EOC1
AdcRegs.INTSEL1N2.bit.INT1E = 1; //Enable ADCINT1
EDIS;

// Sample the temperature sensor
AdcRegs.ADCSOFRC1.all = 0x03; //Sample temp sensor
while(AdcRegs.ADCINTFLG.bit.ADCINT1 == 0){} //Wait for ADCINT1
AdcRegs.ADCINTFLGCLR.bit.ADCINT1 = 1; //Clear ADCINT1
sensorSample = AdcResult.ADCRESULT1; //Get temp sensor sample result

//Convert raw temperature sensor output to a temperature (i.e. degC)
DegreesC = (sensorSample - TempSensorOffset) * TempSensorSlope;
```

For the F2802x, call the below factory stored slope and offset get functions:

```
//Slope of temperature sensor (deg. C / ADC code, fixed pt Q15 format)
#define getTempSlope() (*(int (*)(void))0x3D7E80)()

//ADC code corresponding to temperature sensor output at 0-degreesC
#define getTempOffset() (*(int (*)(void))0x3D7E83)()
```

For the F2803x, call the below factory stored slope and offset get functions:

```
//Slope of temperature sensor (deg. C / ADC code, fixed pt Q15 format)
#define getTempSlope() (*(int (*)(void))0x3D7E82)()

//ADC code corresponding to temperature sensor output at 0-degreesC
#define getTempOffset() (*(int (*)(void))0x3D7E85)()
```

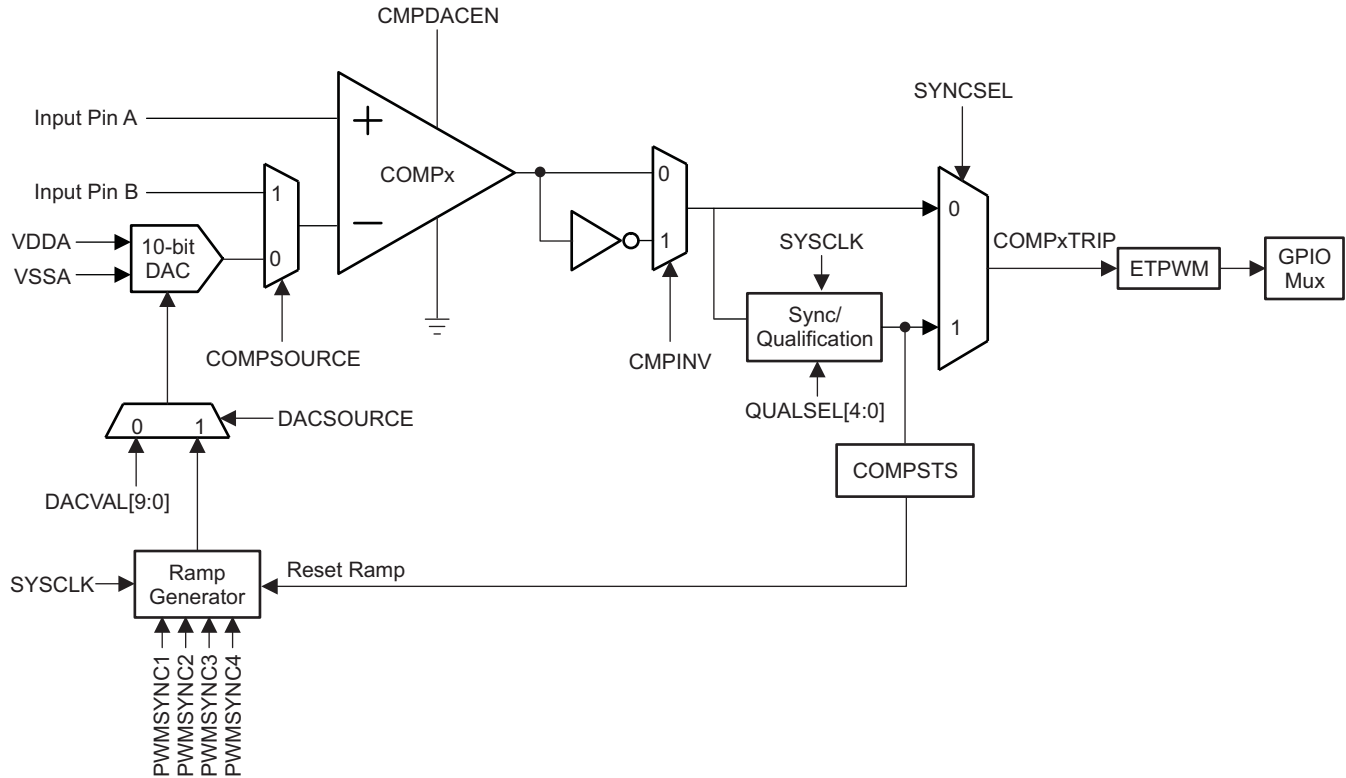
2 Comparator Block

The comparator module described in this reference guide is a true analog voltage comparator in the VDDA domain. The analog portion of the block include the comparator, its inputs and outputs, and the internal DAC reference. The digital circuits, referred to as the wrapper in this document, include the DAC controls, interface to other on-chip logic, output qualification block, and the control signals.

2.1 Features

The comparator block (see [Figure 39](#)) can accommodate two external analog inputs or one external analog input using the internal DAC reference for the other input. The output of the comparator can be passed asynchronously or qualified and synchronized to the system clock period. The comparator output is routed to both the ePWM Trip Zone modules, as well as the GPIO output multiplexer.

Figure 39. Comparator Block Diagram



2.2 Comparator Function

The comparator in each comparator block is an analog comparator module, and as such its output is asynchronous to the system clock. The truth table for the comparator is shown in [Table 24](#).

Figure 40. Comparator

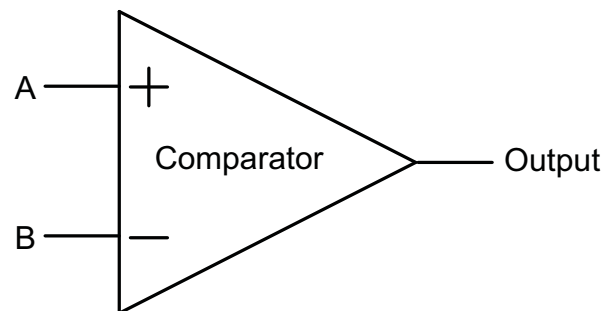


Table 24. Comparator Truth Table

Voltages	Output
Voltage A > Voltage B	1
Voltage B > Voltage A	0

There is no definition for the condition Voltage A = Voltage B since there is hysteresis in the response of the comparator output. Refer to the device datasheet for the value of this hysteresis. This also limits the sensitivity of the comparator output to noise on the input voltages.

The output state of the comparator, after qualification, is reflected by the COMPSTS bit in the COMPSTS register. Since this bit is part of the wrapper, clocks must be enabled to the comparator block for the COMPSTS bit to actively show the comparator state.

2.3 DAC Reference

Each comparator block contains a 10-bit voltage DAC reference that can supply the inverting input (B side input) of the comparator. The voltage output of the DAC is controlled by either the DACVAL register or a ramp down generator.

Since the DAC is also in the analog domain, it does not require a clock to maintain its voltage output. However, a clock is required to modify the digital inputs that control the DAC.

2.3.1 DACVAL Input

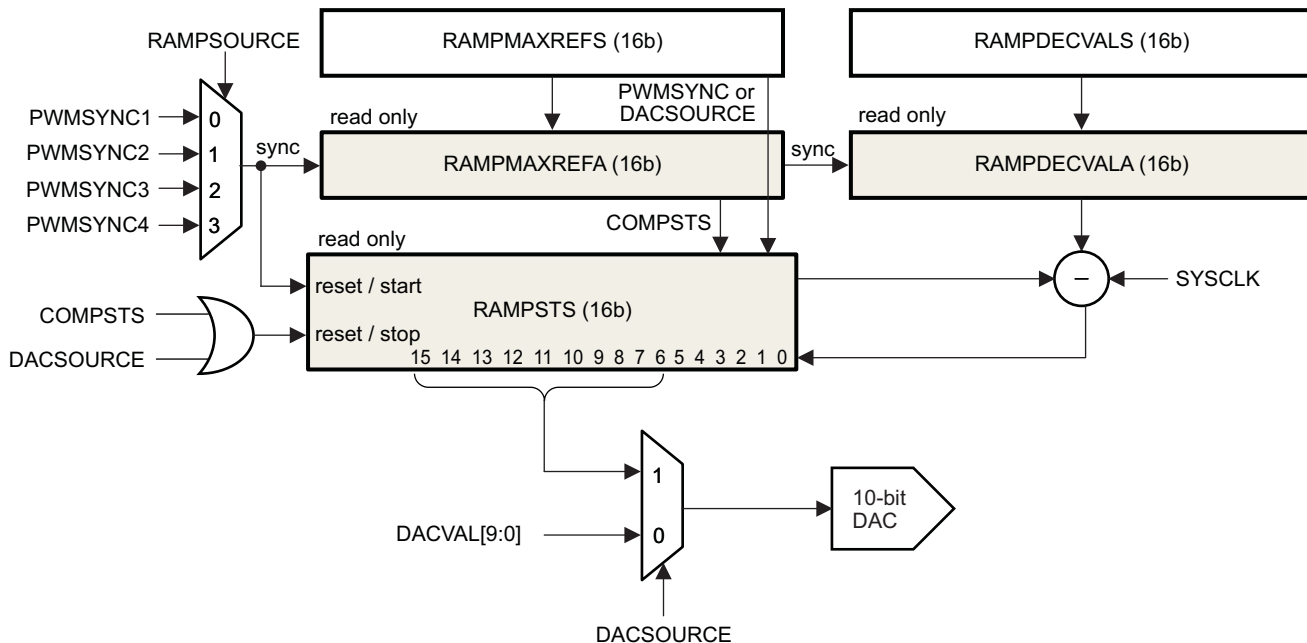
When the DACVAL register is selected to be the DAC input, the output of the DAC is given by the equation:

$$V = \frac{\text{DACVAL} * (\text{VDDA} - \text{VSSA})}{1023}$$

2.3.2 Ramp Generator Input

When selected, the ramp generator (see Figure 41) can produce a falling-ramp DAC output signal. In this mode, the DAC uses the most significant 10-bits of the 16-bit RAMPSTS countdown register as its input.

Figure 41. Ramp Generator Block Diagram



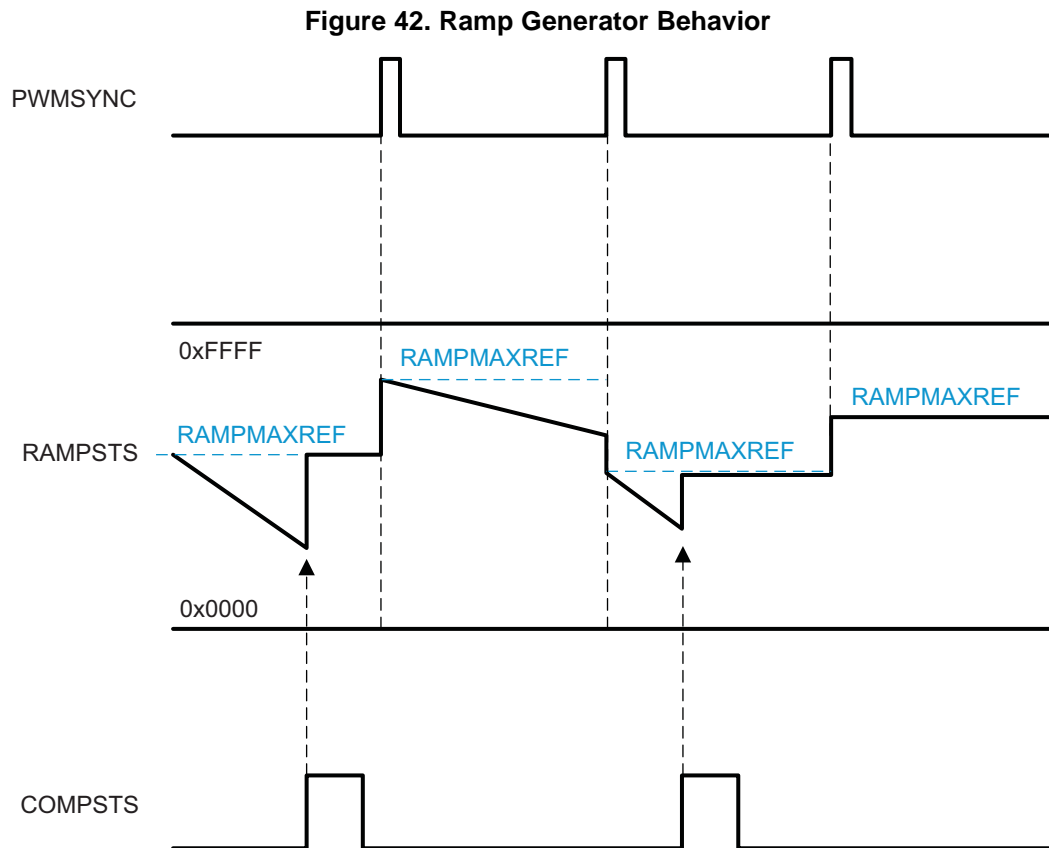
The RAMPSTS register is set to the value of RAMPMAXREF_SHDW when a selected PWMSYNC signal is received, and the value of RAMPDECVAL_ACTIVE is subtracted from RAMPSTS on every SYSCLK cycle thereafter. When the ramp generator is first enabled by setting DACSOURCE = 1, the value of RAMPSTS is loaded from RAMPMAXREF_SHDW, and the register remains static until the first PWMSYNC signal is received.

If the COMPSTS bit is set by the comparator while the ramp generator is active, the RAMPSTS register will reset to the value of RAMPMAXREF_ACTIVE and remain static until the next PWMSYNC signal is received. If the value of RAMPSTS reaches zero, the RAMPSTS register will remain static at zero until the next PWMSYNC signal is received.

To reduce the likelihood of race conditions when updating the ramp generator RAMPMAXREFA and RAMPDECVALA values, only the shadow registers RAMPMAXREF_SHDW and RAMPDECVAL_SHDW have write permissions. The values of the shadow registers are copied to the active registers on the next PWMSYNC signal. User software should take further steps to avoid writing to the shadow registers in the same cycle as a PWMSYNC signal or else the previous shadow register value may be lost.

The PWMSYNC signal width must be greater than SYSCLK to ensure that the ramp generator is able to detect the PWMSYNC signal.

The ramp generator behavior is further illustrated in [Figure 42](#)



2.4 Initialization

Two steps must be performed prior to using the comparator block:

1. Enable the Band Gap inside the ADC by writing a 1 to the ADCBGPWD bit inside ADCTRL1.
2. Enable the comparator block by writing a 1 to the COMPDACEN bit in the COMPCTL register.

2.5 Digital Domain Manipulation

At the output of the comparator there are two more functional blocks that can be used to influence the behavior of the comparator output. They are:

1. Inverter circuit: Controlled by the CMPINV bit in the COMPCTL register; will apply a logical NOT to the output of the comparator. This function is asynchronous, while its control requires a clock present in order to change its value.
2. Qualification block: Controlled by the QUALSEL bit field in the COMPCTL register, and gated by the

SYNCSEL bit in the COMPCTL register. This block can be used as a simple filter to only pass the output of the comparator once it is synchronized to the system clock, and qualified by the number of system clocks defined in QUALSEL bit field.

2.6 Comparator Registers

F280x2x devices have two comparators COMP1 and COMP2. Table 25 lists the registers for these modules.

Name	Address Range	Size(x16)	Description
COMP1	6400h – 641Fh	1	Comparator
COMP2	6420h – 643Fh	1	Comparator
COMP3	6440h – 645Fh	1	Comparator

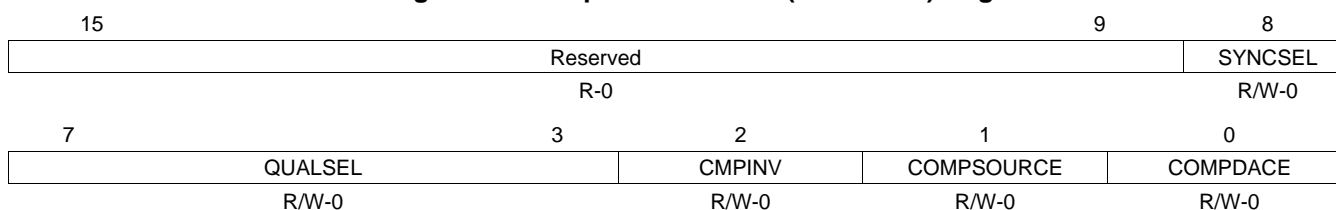
Table 25. Comparator Module Registers

Name	Address Range(base)	Size(x16)	Description
COMPCTL	0x00	1	Comparator Control ⁽¹⁾
Reserved	0x01	1	Reserved
COMPSTS	0x02	1	Compare Output Status
Reserved	0x03	1	Reserved
DACCTL	0x04	1	DAC Control ⁽¹⁾
Reserved	0x05	1	Reserved
DACVAL	0x06	1	10-bit DAC Value
Reserved	0x07	1	Reserved
RAMPMAXREF_ACTIVE	0x08	1	Ramp Generator Maximum Reference (Active)
Reserved	0x09	1	Reserved
RAMPMAXREF_SHDW	0x0A	1	Ramp Generator Maximum Reference (Shadow)
Reserved	0x0B	1	Reserved
RAMPDECVAL_ACTIVE	0x0C	1	Ramp Generator Decrement Value (Active)
Reserved	0x0D	1	Reserved
RAMPDECVAL_SHDW	0x0E	1	Ramp Generator Decrement Value (Shadow)
Reserved	0x0F	1	Reserved
RAMPSTS	0x10	1	Ramp Generator Status
Reserved	0x11 0x1F	15	Reserved

⁽¹⁾ This register is EALLOW protected.

2.6.1 Comparator Control (COMPCTL) Register

Figure 43. Comparator Control (COMPCTL) Register



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 26. COMPCTL Register Field Descriptions

Bit	Field	Value	Description
15-9	Reserved		Reads return a 0; Writes have no effect.
8	SYNCSEL	0 1	Synchronization select for output of the comparator before being passed to ETPWM/GPIO blocks Asynchronous version of Comparator output is passed Synchronous version of comparator output is passed
7-3	QUALSEL	0h 1h 2h ... Fh	Qualification Period for synchronized output of the comparator Synchronized value of comparator is passed through Input to the block must be consistent for 2 consecutive clocks before output of Qual block can change Input to the block must be consistent for 3 consecutive clocks before output of Qual block can change ... Input to the block must be consistent for 16 consecutive clocks before output of Qual block can change
2	CMPINV	0 1	Invert select for Comparator Output of comparator is passed Inverted output of comparator is passed
1	COMPSOURCE	0 1	Source select for comparator inverting input Inverting input of comparator connected to internal DAC Inverting input connected to external pin
0	COMPDACE	0 1	Comparator/DAC Enable Comparator/DAC logic is powered down. Comparator/DAC logic is powered up.

2.6.2 Compare Output Status (COMPSTS) Register

Figure 44. Compare Output Status (COMPSTS) Register

15	1	0
Reserved		COMPSTS
R-0		R-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 27. Compare Output Status (COMPSTS) Register Field Descriptions

Bit	Field	Value	Description
15-1	Reserved		Reads return zero and writes have no effect.
0	COMPSTS		Logical latched value of the comparator

2.6.3 DAC Control (DACCTL) Register

Figure 45. DAC Control (DACCTL) Register

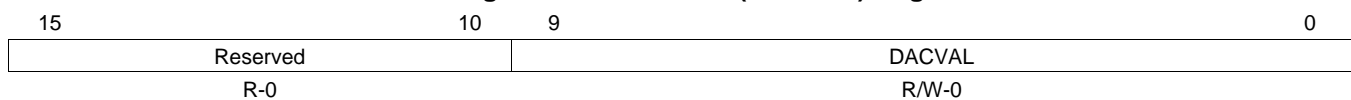
15	14	13	8
FREE:SOFT		Reserved	
R/W-0		R-0	
7	5	4	1
Reserved		RAMPSOURCE	
R-0		R/W-0	
		0	0
		DACSOURCE	
		R/W-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 28. DACCTL Register Field Descriptions

Bit	Field	Value	Description
15-14	FREE:SOFT	0h 1h 2h-3h	Emulation mode behavior. Selects ramp generator behavior during emulation suspend. Stop immediately Complete current ramp, and stop on the next PWMSYNC signal Run free
13-5	Reserved		Reads return a 0; Writes have no effect.
4-1	RAMPSOURCE	0h 1h 2h 3h 4h-Fh	Ramp generator source sync select PWMSYNC1 is the source sync PWMSYNC2 is the source sync PWMSYNC3 is the source sync PWMSYNC4 is the source sync Reserved
0	DACSOURCE	0 1	DAC source control. Select DACVAL or ramp generator to control the DAC. DAC controlled by DACVAL DAC controlled by ramp generator

2.6.4 DAC Value (DACVAL) Register

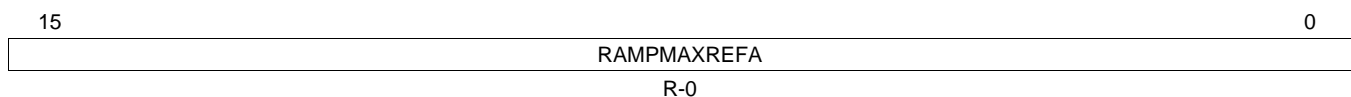
Figure 46. DAC Value (DACVAL) Register


LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

Table 29. DAC Value (DACVAL) Register Field Descriptions

Bit	Field	Value	Description
15-10	Reserved		Reads return zero and writes have no effect.
9-0	DACVAL	0-3FFh	DAC Value bits, scales the output of the DAC from 0 – 1023.

2.6.5 Ramp Generator Maximum Reference Active (RAMPMAXREF_ACTIVE) Register

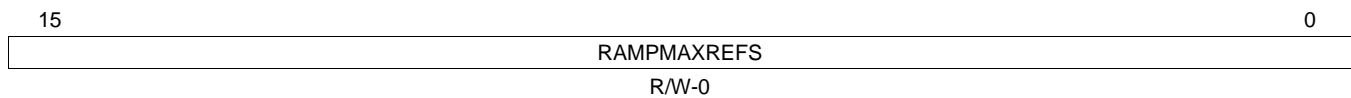
Figure 47. Ramp Generator Maximum Reference Active (RAMPMAXREF_ACTIVE) Register


LEGEND: R = Read only; -n = value after reset

Table 30. Ramp Generator Maximum Reference Active (RAMPMAXREF_ACTIVE) Register Field Descriptions

Bit	Field	Value	Description
15-0	RAMPMAXREFA	0-FFFFh	16-bit maximum reference active value for down ramp generator. This value is loaded from RAMPMAXREF_SHDW when the PWMSYNC signal is received.

2.6.6 Ramp Generator Maximum Reference Shadow (RAMPMAXREF_SHDW) Register

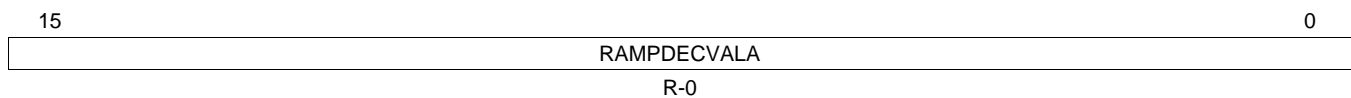
Figure 48. Ramp Generator Maximum Reference Shadow (RAMPMAXREF_SHDW) Register


LEGEND: R/W = Read/Write; -n = value after reset

Table 31. Ramp Generator Maximum Reference Shadow (RAMPMAXREF_SHDW) Register Field Descriptions

Bit	Field	Value	Description
15-0	RAMPMAXREFS	0-FFFFh	16-bit maximum reference shadow value for down ramp generator

2.6.7 Ramp Generator Decrement Value Active (RAMPDECVAL_ACTIVE) Register

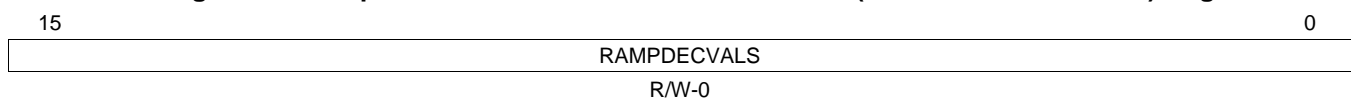
Figure 49. Ramp Generator Decrement Value Active (RAMPDECVAL_ACTIVE) Register


LEGEND: R = Read only; -n = value after reset

Table 32. Ramp Generator Decrement Value Active (RAMPDECVAL_ACTIVE) Register Field Descriptions

Bit	Field	Value	Description
15-0	RAMPDECVALA	0-FFFFh	16-bit decrement active value for down ramp generator. This value is loaded from RAMPDECVAL_SHDW when the PWMSYNC signal is received.

2.6.8 Ramp Generator Decrement Value Shadow (RAMPDECVAL_SHDW) Register

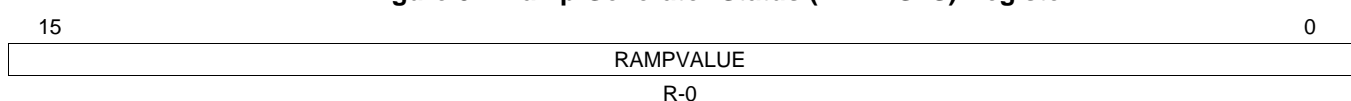
Figure 50. Ramp Generator Decrement Value Shadow (RAMPDECVAL_SHDW) Register


LEGEND: R/W = Read/Write; -n = value after reset

Table 33. Ramp Generator Decrement Value Shadow (RAMPDECVAL_SHDW) Register Field Descriptions

Bit	Field	Value	Description
15-0	RAMPDECVALS	0-FFFFh	16-bit decrement shadow value for down ramp generator

2.6.9 Ramp Generator Status (RAMPSTS) Register

Figure 51. Ramp Generator Status (RAMPSTS) Register


LEGEND: R = Read only; -n = value after reset

Table 34. Ramp Generator Status (RAMPSTS) Register Field Descriptions

Bit	Field	Value	Description
15-0	RAMPVALUE	0-FFFFh	16-bit value of down ramp generator

Appendix A Revision History

This document has been revised to include the following technical change(s).

Table 35. Changes in this Document

Location	Additions/Deletions/Modificatons
Figure 9	Changed CLKDIV4 bit to Reserved
Section 1.10.7	Updated Titles
Table 2	Added COMPHYSTCTL
Section 1.7	Updated Text
Section 2.6	Added COMP3
Figure 42	Updated Figure

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46C and to discontinue any product or service per JESD48B. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products (also referred to herein as "components") are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

TI does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

Reproduction of significant portions of TI information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Resale of TI components or services with statements different from or beyond the parameters stated by TI for that component or service voids all express and any implied warranties for the associated TI component or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards which anticipate dangerous consequences of failures, monitor failures and their consequences, lessen the likelihood of failures that might cause harm and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed a special agreement specifically governing such use.

Only those TI components which TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components which have **not** been so designated is solely at the Buyer's risk, and that Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components which meet ISO/TS16949 requirements, mainly for automotive use. Components which have not been so designated are neither designed nor intended for automotive use; and TI will not be responsible for any failure of such components to meet such requirements.

Products

Audio	www.ti.com/audio
Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DLP® Products	www.dlp.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
OMAP Mobile Processors	www.ti.com/omap
Wireless Connectivity	www.ti.com/wirelessconnectivity

Applications

Automotive and Transportation	www.ti.com/automotive
Communications and Telecom	www.ti.com/communications
Computers and Peripherals	www.ti.com/computers
Consumer Electronics	www.ti.com/consumer-apps
Energy and Lighting	www.ti.com/energy
Industrial	www.ti.com/industrial
Medical	www.ti.com/medical
Security	www.ti.com/security
Space, Avionics and Defense	www.ti.com/space-avionics-defense
Video and Imaging	www.ti.com/video

TI E2E Community e2e.ti.com