

ADC10_A

NOTE: This chapter is an excerpt from the *MSP430x5xx and MSP430x6xx Family User's Guide*. The most recent version of the full user's guide is available at <http://www.ti.com/lit/pdf/slau208>.

The ADC10_A module is a high-performance 10-bit analog-to-digital converter (ADC). This chapter describes the operation of the ADC10_A module.

Topic	Page
1.1 ADC10_A Introduction	2
1.2 ADC10_A Operation	4
1.3 ADC10_A Registers	16

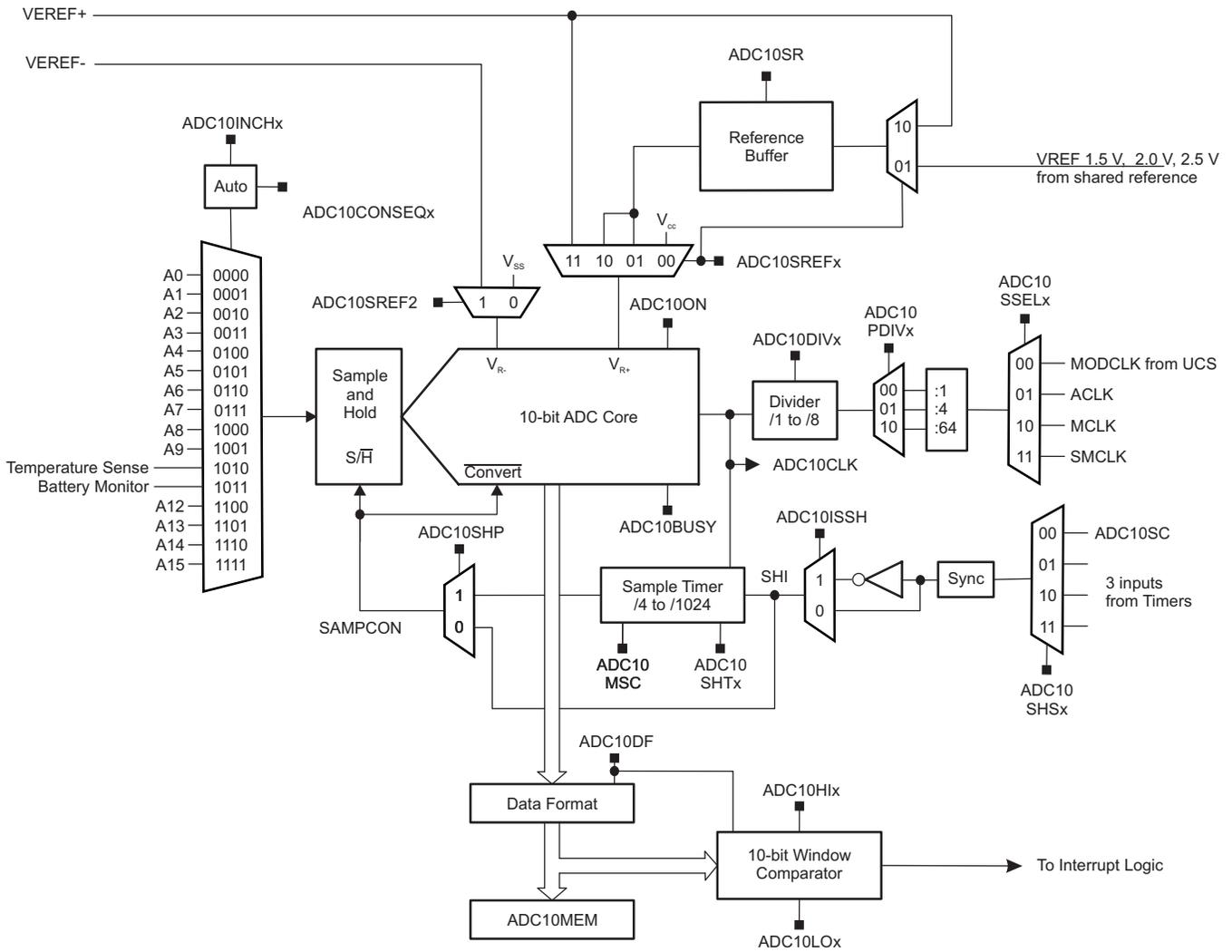
1.1 ADC10_A Introduction

The ADC10_A module supports fast 10-bit analog-to-digital conversions. The module implements a 10-bit SAR core with sample select control and a window comparator.

ADC10_A features include:

- Greater than 200-ksps maximum conversion rate
- Monotonic 10-bit converter with no missing codes
- Sample-and-hold with programmable sampling periods controlled by software or timers.
- Conversion initiation by software or different Timers
- Software-selectable on-chip reference using the REF module or external reference
- 12 individually configurable external input channels
- Conversion channel for temperature sensor of the REF module
- Selectable conversion clock source
- Single-channel, repeat-single-channel, sequence (autoscan), and repeat-sequence (repeated autoscan) conversion modes
- Window comparator for low-power monitoring of input signals
- Interrupt vector register for fast decoding of six ADC interrupts (ADC10IFG0, ADC10TOVIFG, ADC10OVIFG, ADC10LOIFG, ADC10INIFG, ADC10HIIFG)

[Figure 1-1](#) shows the block diagram of ADC10_A. The on-chip reference voltage generation is located in the reference module (see the device-specific data sheet).



Copyright © 2017, Texas Instruments Incorporated

- A MODCLK is generated by the MODOSC, which is part of the UCS. See the [UCS chapter](#) for more information.
- B When using ADC10SHP = 0 no synchronisation of the trigger input is done.

Figure 1-1. ADC10_A Block Diagram

1.2 ADC10_A Operation

The ADC10_A module is configured with user software. The setup and operation of the ADC10_A is discussed in the following sections.

1.2.1 10-Bit ADC Core

The ADC core converts an analog input to its 10-bit digital representation and stores the result in the conversion register ADC10MEM0. The core uses two programmable/selectable voltage levels (V_{R+} and V_{R-}) to define the upper and lower limits of the conversion. The digital output (N_{ADC}) is full scale (03FFh) when the input signal is equal to or higher than V_{R+} , and zero when the input signal is equal to or lower than V_{R-} . The input channel and the reference voltage levels (V_{R+} and V_{R-}) are defined in the conversion-control memory. Equation 1 shows the conversion formula for the ADC result, N_{ADC} .

$$N_{ADC} = 1023 \times \frac{V_{in} - V_{R-}}{V_{R+} - V_{R-}} \quad (1)$$

The control registers ADC10CTL0, ADC10CTL1 and ADC10CTL2 configure the ADC10_A core. The ADC10ON bit enables or disables the core. The ADC10_A can be turned off when not in use to save power. With few exceptions, the ADC10_A control bits can be modified only when ADC10ENC = 0. ADC10ENC must be set to 1 before any conversion can take place.

1.2.1.1 Conversion Clock Selection

The ADC10CLK is used both as the conversion clock and to generate the sampling period when the pulse sampling mode is selected. The ADC10_A source clock is selected using the ADC10SSELx bits. Possible ADC10CLK sources are SMCLK, MCLK, ACLK, and the MODCLK. The input clock can be divided by a value of 1 through 512 using both the ADC10DIVx bits and the ADC10PDIVx bits.

MODCLK, generated internally in the UCS, is in the 5-MHz range but varies with individual devices, supply voltage, and temperature. See the device-specific data sheet for the MODOSC specification.

The user must ensure that the clock chosen for ADC10CLK remains active until the end of a conversion. If the clock is removed during a conversion, the operation does not complete and any result is invalid.

1.2.2 ADC10_A Inputs and Multiplexer

The 14 external and 2 internal analog signals are selected as the channel for conversion by the analog input multiplexer. The input multiplexer is a break-before-make type to reduce input-to-input noise injection resulting from channel switching (see Figure 1-2). The input multiplexer is also a T-switch to minimize the coupling between channels. Channels that are not selected are isolated from the ADC and the intermediate node is connected to analog ground (AV_{SS}) so that the stray capacitance is grounded to eliminate crosstalk.

The ADC10_A uses the charge redistribution method. When the inputs are internally switched, the switching action may cause transients on the input signal. These transients decay and settle before causing an errant conversion.

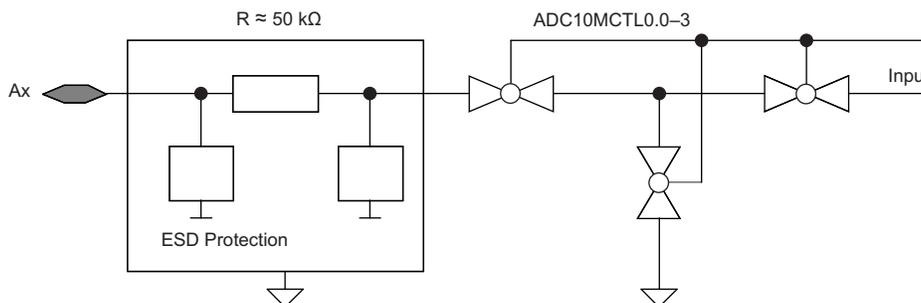


Figure 1-2. Analog Multiplexer

1.2.2.1 Analog Port Selection

The ADC10_A inputs are multiplexed with digital port pins. When analog signals are applied to digital gates, parasitic current can flow from V_{CC} to GND. This parasitic current occurs if the input voltage is near the transition level of the gate. Disabling the digital part of the port pin eliminates the parasitic current flow and, therefore, reduces overall current consumption. The PxSEL.y bits can disable the port pin input and output buffers.

```

; Px.0 and Px.1 configured for analog input
BIS.B #3h,&PxSEL ; Px.1 and Px.0 ADC10_A function
    
```

1.2.3 Voltage Reference Generator

The ADC10_A module can be used either with the on-chip reference supplied by the REF module or an externally reference voltage supplied on external pins.

The on-chip reference supplies 1.5 V, 2.0 V, and 2.5 V. The reference voltages are controlled by the control registers of the REF module (see the [REF chapter](#) for details). The internal V_{CC} can also be used as the voltage reference.

External reference voltages may be supplied for V_{R+} and V_{R-} through pins VEREF+ and VEREF-, respectively.

1.2.3.1 Internal Reference Low-Power Features

The on-chip reference is designed for low-power applications. This reference includes a bandgap voltage source and a separate reference buffer both located in the REF module. The current consumption of each is specified separately in the device-specific data sheet. The ADC10_A also contains an internal buffer for reference voltages. This buffer is automatically enabled when the internal reference is selected for VREF+, but it is also optionally available for VEREF+. The on-chip reference from the REF module must be enabled by software. Its settling time is 25 μ s (typical). See the device-specific data sheet and the [REF chapter](#) for further information on the on-chip reference.

The reference buffer of the ADC10_A also has selectable speed versus power settings. When the maximum conversion rate is below 50 ksps, setting ADC10SR = 1 reduces the current consumption of the buffer by approximately 50%.

1.2.4 Auto Power Down

The ADC10_A is designed for low-power applications. When the ADC10_A is not actively converting, the core is automatically disabled, and it is automatically reenabled when needed. The MODOSC is also automatically enabled when needed and disabled when not needed.

1.2.5 Sample and Conversion Timing

An analog-to-digital conversion is initiated with a rising edge of the sample input signal, SHI. The source for SHI is selected with the ADC10SHSx bits and includes the following:

- ADC10SC bit
- Three timer outputs

The polarity of the SHI signal source can be inverted with the ADC10ISSH bit. The SAMPCON signal controls the sample period and start of conversion. When SAMPCON is high, sampling is active. The high-to-low SAMPCON transition starts the analog-to-digital conversion, which requires 12 ADC10CLK cycles in 10-bit resolution mode. One additional ADC10CLK is used for the window comparator. Two different sample-timing methods are defined by control bit ADC10SHP, extended sample mode and pulse mode.

1.2.5.1 Extended Sample Mode

The extended sample mode is selected when $ADC10SHP = 0$. The SHI signal directly controls $SAMPCON$ and defines the length of the sample period t_{sample} . When $SAMPCON$ is high, sampling is active. The high-to-low $SAMPCON$ transition starts the conversion after synchronization with $ADC10CLK$ (see Figure 1-3).

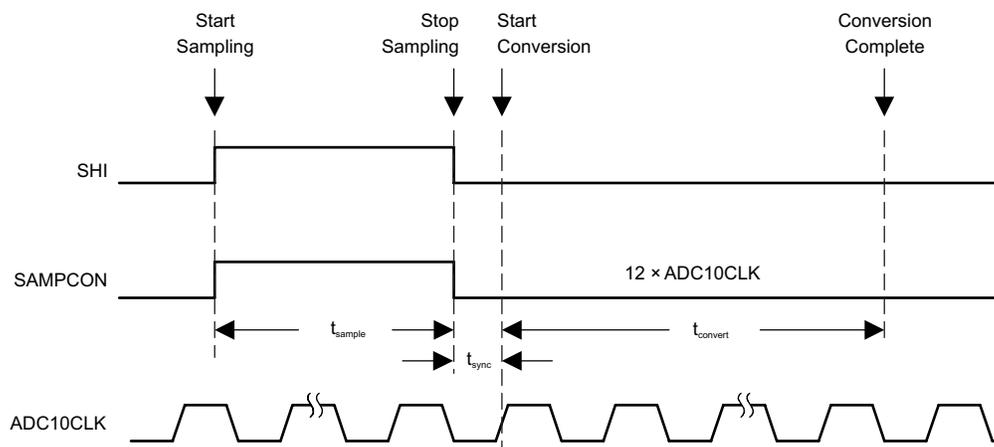


Figure 1-3. Extended Sample Mode

1.2.5.2 Pulse Sample Mode

The pulse sample mode is selected when $ADC10SHP = 1$. The SHI signal triggers the sampling timer. The $ADC10SHTx$ bits in $ADC10CTL0$ control the interval of the sampling timer that defines the $SAMPCON$ sample period, t_{sample} . The sampling timer keeps $SAMPCON$ high after synchronization with $ADC10CLK$ for a programmed interval of t_{sample} . The total sampling time is t_{sample} plus t_{sync} (see Figure 1-4).

The $ADC10SHTx$ bits select the sampling time in 4x multiples of $ADC10CLK$.

NOTE: The $ADC10SC$ bit is automatically cleared. Do not modify this bit while it is set.

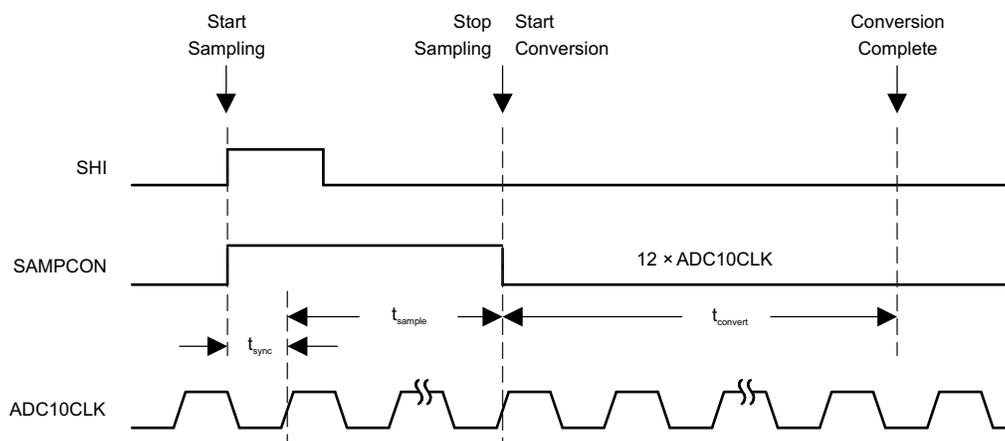


Figure 1-4. Pulse Sample Mode

1.2.5.3 Sample Timing Considerations

When SAMPCON = 0, all Ax inputs are high impedance. When SAMPCON = 1, the selected Ax input can be modeled as an RC low-pass filter during the sampling time t_{sample} (see Figure 1-5). An internal MUX-on input resistance R_i (see the device-specific data sheet) in series with capacitor C_i (see the device-specific data sheet) is seen by the source. The capacitor C_i voltage V_c must be charged to within one-half LSB of the source voltage V_s for an accurate 10-bit conversion.

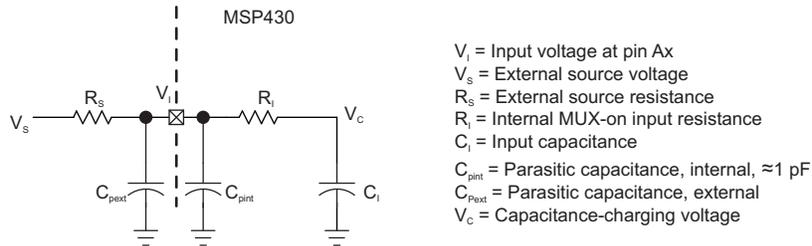


Figure 1-5. Analog Input Equivalent Circuit

The resistance of the source R_s and R_i affect t_{sample} . The minimum sample time must not be violated. Violation of the minimum sample time may cause a conversion not to take place. See the device-specific data sheet for the t_{sample} limits.

1.2.6 Conversion Result

The conversion result is accessible using the ADC10MEM0 register independently of the conversion mode selected by the user. When a conversion result is written to ADC10MEM0, the ADC10IFG0 is set.

1.2.7 ADC10_A Conversion Modes

The ADC10_A has four operating modes selected by the CONSEQx bits (see Table 1-1).

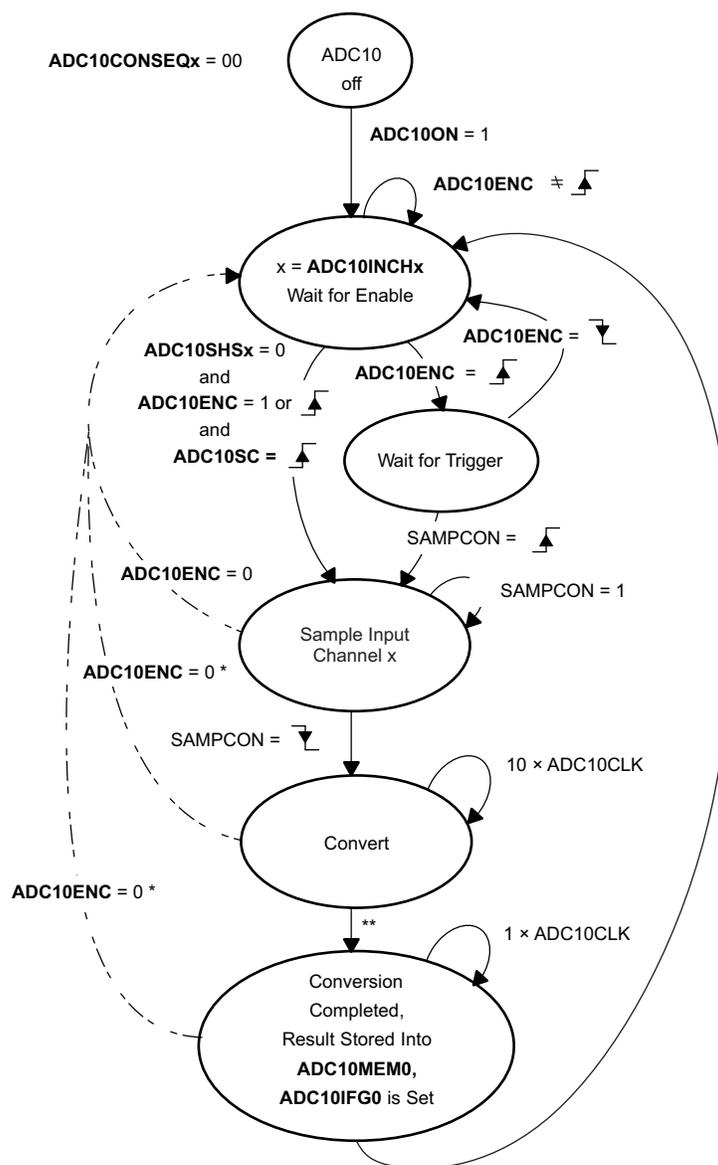
Table 1-1. Conversion Mode Summary

ADC10CONSEQx	Mode	Operation
00	Single-channel single-conversion	A single channel is converted once.
01	Sequence-of-channels (autoscan)	A sequence of channels is converted once.
10	Repeat-single-channel	A single channel is converted repeatedly.
11	Repeat-sequence-of-channels (repeated autoscan)	A sequence of channels is converted repeatedly.

1.2.7.1 Single-Channel Single-Conversion Mode

A single channel selected by ADC10INCHx is sampled and converted once. The ADC result is written to ADC10MEM0. Figure 1-6 shows the flow of the single-channel single-conversion mode. When ADC10SC triggers a conversion, successive conversions can be triggered by the ADC10SC bit. When any other trigger source is used, ADC10ENC must be toggled between each conversion.

Resetting the ADC10ON bit during a conversion causes the ADC10_A to enter the "ADC10 off" state. In this case, the value of the conversion register and the value of the interrupt flags are unpredictable.



* Conversion result is unpredictable

** Two ADC10CLK cycles needed

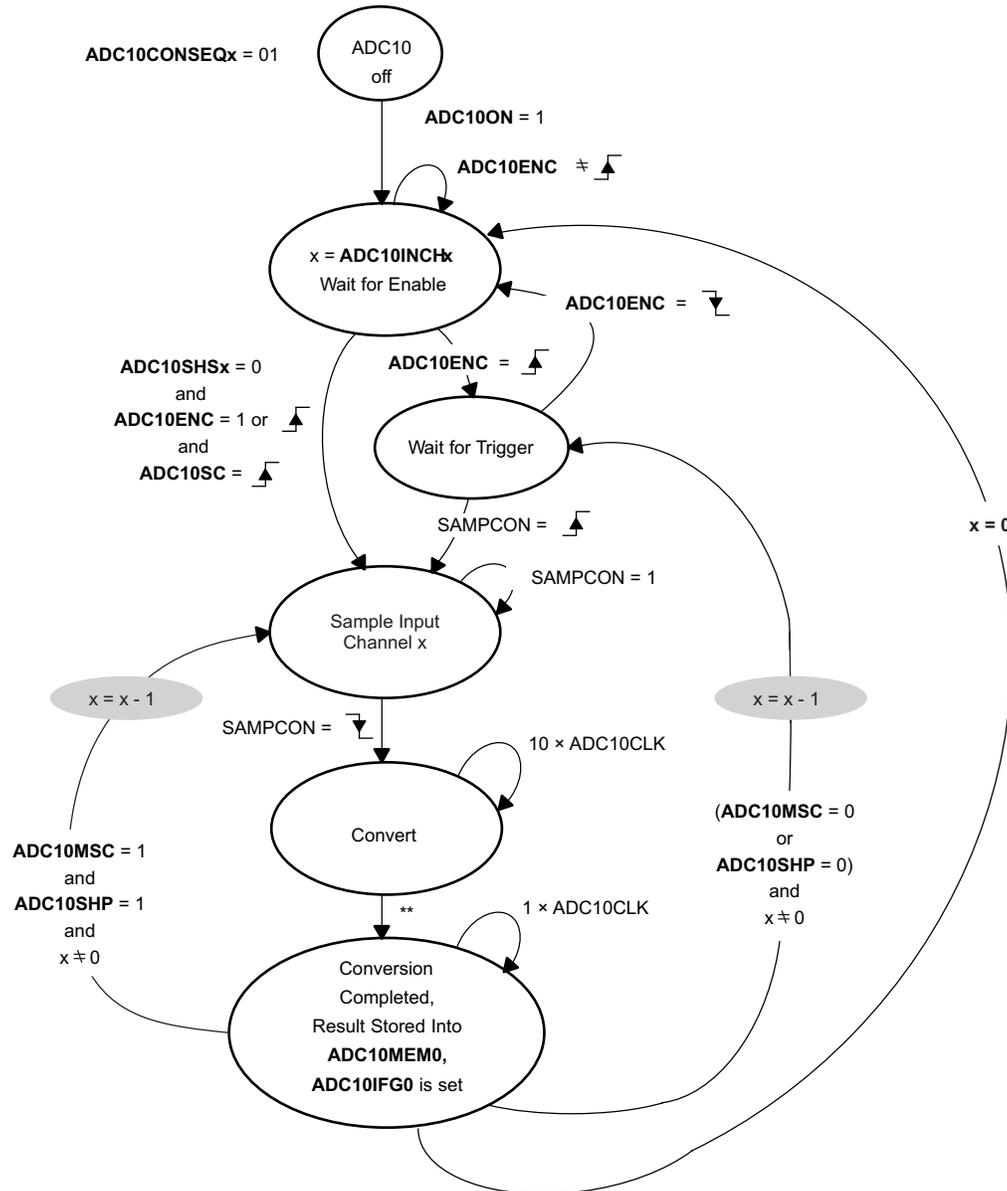
x = Pointer to the selected ADC10_A channel defined by ADC10INCHx

All bit and register names are bold font; signals names are normal font.

Figure 1-6. Single-Channel Single-Conversion Mode

1.2.7.2 Sequence-of-Channels Mode (Autoscan Mode)

In sequence-of-channels mode, also referred to as autoscan mode, a sequence of channels is sampled and converted once. The sequence begins with the channel selected by the ADC10INCHx bits and decrements to channel A0. Each ADC result is written to ADC10MEM0. The sequence stops after conversion of channel A0. Figure 1-7 shows the sequence-of-channels mode. When ADC10SC triggers a sequence, successive sequences can be triggered by the ADC10SC bit. When any other trigger source is used, ADC10ENC must be toggled between each sequence. As in all conversion modes resetting ADC10ON bit within a conversion causes the ADC10_A to go back into "ADC10 off" state.

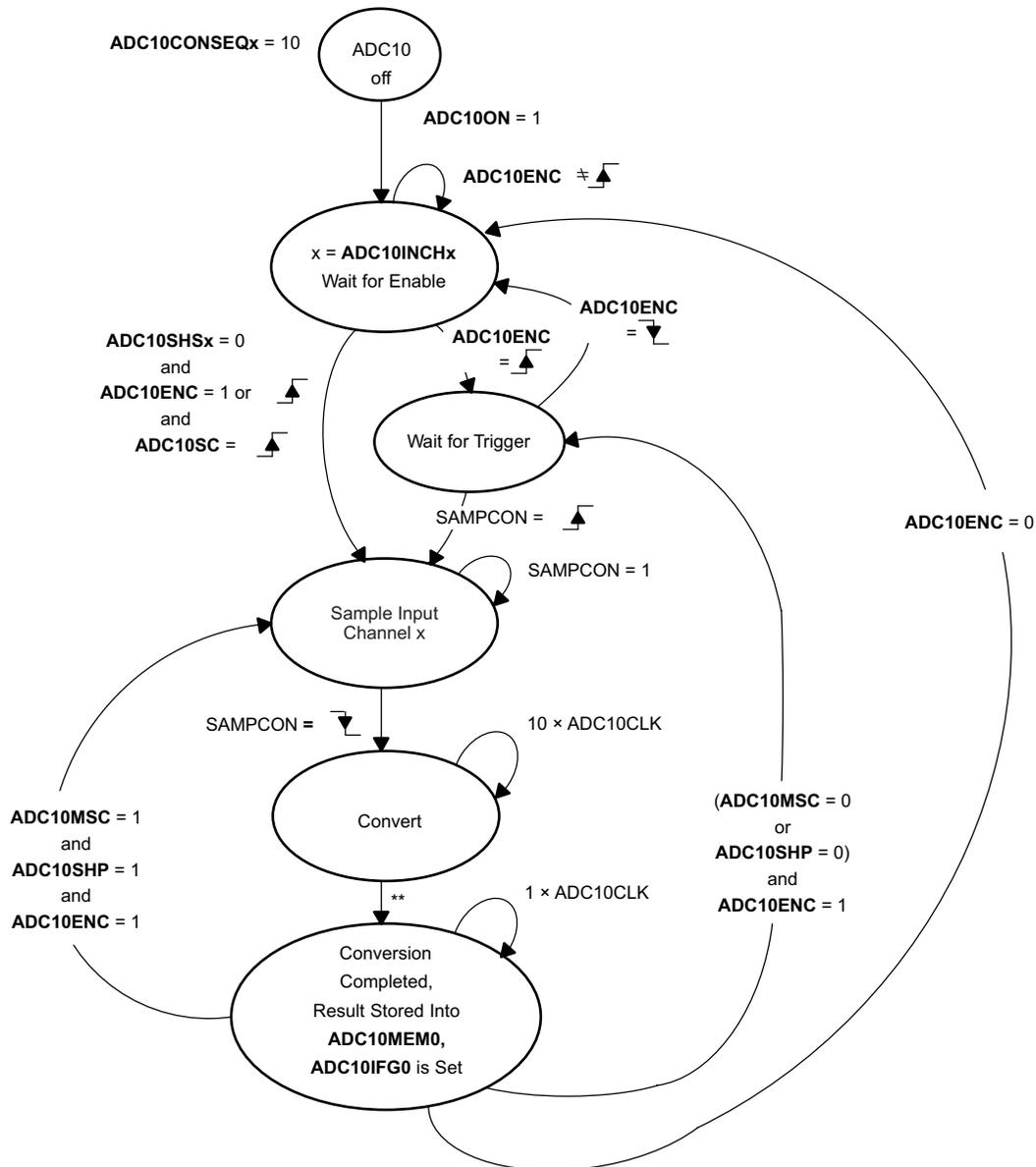


x = Input channel Ax
 ** Two ADC10CLK cycles are needed.
 All bit and register names are bold font; signals names are normal font.

Figure 1-7. Sequence-of-Channels Mode

1.2.7.3 Repeat-Single-Channel Mode

A single channel selected by ADC10INCHx is sampled and converted continuously. Each ADC result is written to ADC10MEM0. Figure 1-8 shows the repeat-single-channel mode.



x = Pointer to the selected ADC10_A channel defined by **ADC10INCHx**.

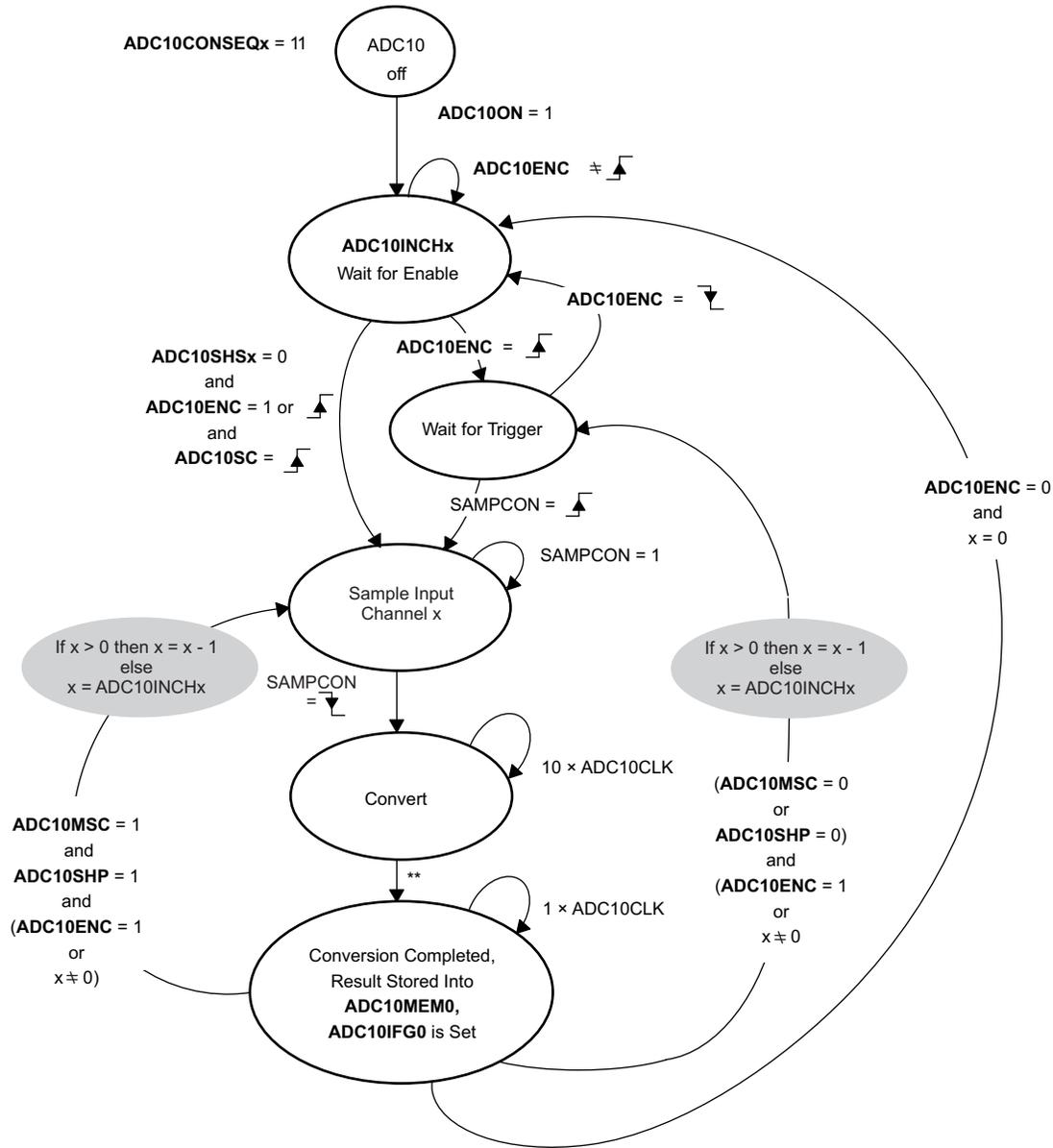
** Two ADC10CLK cycles are needed.

All bit and register names are bold font; signals names are normal font.

Figure 1-8. Repeat-Single-Channel Mode

1.2.7.4 Repeat-Sequence-of-Channels Mode (Repeated Autoscan Mode)

In this mode, a sequence of channels is sampled and converted repeatedly. This mode is also referred to as repeated autoscan mode. The sequence begins with the channel selected by ADC10INCHx and decrements to channel A0. Each ADC result is written to ADC10MEM0. The sequence ends after conversion of channel A0, and the next trigger signal re-starts the sequence. Figure 1-9 shows the repeat-sequence-of-channels mode.



x = Input channel Ax
 ** Two ADC10CLK cycles are needed.
 All bit and register names are bold font; signals names are normal font.

Figure 1-9. Repeat-Sequence-of-Channels Mode

1.2.7.5 Using the Multiple Sample and Convert (ADC10MSC) Bit

To configure the converter to perform successive conversions automatically and as quickly as possible, a multiple sample and convert function is available. When $ADC10MSC = 1$, $CONSEQx > 0$, and the sample timer is used, the first rising edge of the SHI signal triggers the first conversion. Successive conversions are triggered automatically as soon as the prior conversion is completed. Additional rising edges on SHI are ignored until the sequence is completed in the single-sequence mode, or until the ADC10ENC bit is toggled in repeat-single-channel or repeated-sequence modes. The function of the ADC10ENC bit is unchanged when using the ADC10MSC bit.

1.2.7.6 Stopping Conversions

Stopping ADC10_A activity depends on the mode of operation. The recommended ways to stop an active conversion or conversion sequence are:

- Resetting ADC10ENC in single-channel single-conversion mode stops a conversion immediately and the results are unpredictable. For correct results, poll the busy bit until reset before clearing ADC10ENC.
- Resetting ADC10ENC during repeat-single-channel operation stops the converter at the end of the current conversion.
- Resetting ADC10ENC during a sequence or repeat-sequence mode stops the converter at the end of the sequence.
- Any conversion mode may be stopped immediately by setting the $CONSEQx = 0$ and resetting the ADC10ENC bit. Conversion data are unreliable.

1.2.8 Window Comparator

The window comparator monitors analog signals without any CPU interaction. The following list describes the available interrupt flags and the conditions when they are asserted:

- The ADC10LO interrupt flag (ADC10LOIFG) is set if the current result of the ADC10_A conversion is below the low threshold defined in register ADC10LO.
- The ADC10HI interrupt flag (ADC10HIIFG) is set if the current result of the ADC10_A conversion is greater than the high threshold defined in register ADC10HI.
- The ADC10IN interrupt flag (ADC10INIFG) is set if the current result of the ADC10_A conversion is greater than the low threshold defined in register ADC10LO and less than the high threshold defined in ADC10HI.

These interrupts are generated independent of the conversion mode selected by the user. The update of the window comparator interrupt flags happens in parallel to the ADC10IFG0.

Make sure that the values in the ADC10HI and ADC10LO registers are in the correct data format. For example, if the binary data format is selected ($ADC10DF = 0$), then the thresholds in the threshold registers ADC10HI and ADC10LO also must be entered binary coded. Changing ADC10DF or the ADC10RES resets the threshold registers.

The interrupt flags must be reset by the user software. The ADC10_A only updates the flags each time a new value is available in the ADC10MEM0. This update is only a set of the corresponding interrupt flag. When the user uses the window comparator flags, it must be ensured that they are reset by software according to the application needs.

1.2.9 Using the Integrated Temperature Sensor

To use the on-chip temperature sensor, select the analog input channel ADC10INCHx = 1010. Any other configuration is done as if an external channel is selected, including reference selection, conversion-mode selection, and other settings. The temperature sensor is located in the REF module of the device and is configured by the REF module control registers.

Figure 1-10 shows the typical temperature sensor transfer function. When using the temperature sensor, the sample period must be greater than 30 μ s. The temperature sensor offset error can be large and may need to be calibrated for most applications (see the device-specific data sheet for parameters). Some MSP430 devices include calibration data that can be used to compute temperature more accurately. For more information, refer to .

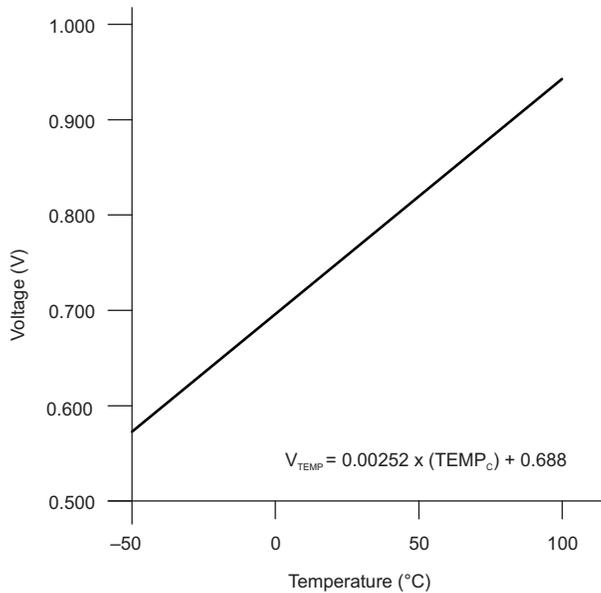


Figure 1-10. Typical Temperature Sensor Transfer Function

1.2.10 ADC10_A Grounding and Noise Considerations

As with any high-resolution ADC, appropriate printed-circuit-board layout and grounding techniques must be followed to eliminate ground loops, unwanted parasitic effects, and noise.

Ground loops are formed when return current from the ADC flows through paths that are common with other analog or digital circuitry. If care is not taken, this current can generate small, unwanted offset voltages that can add to or subtract from the reference or input voltages of the ADC. The connections shown in [Figure 1-11](#) prevent this.

In addition to grounding, ripple and noise spikes on the power-supply lines due to digital switching or switching power supplies can corrupt the conversion result. TI recommends a noise-free design using separate analog and digital ground planes with a single-point connection to achieve high accuracy.

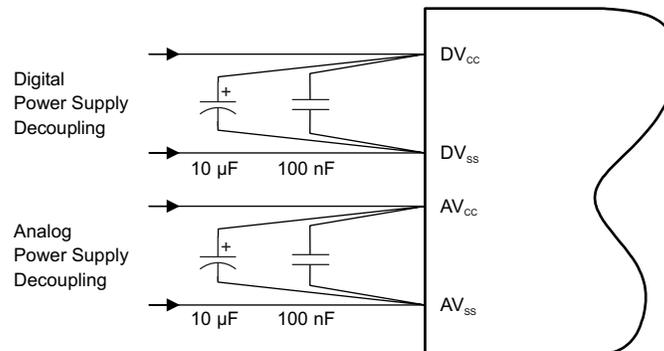


Figure 1-11. ADC10_A Grounding and Noise Considerations

1.2.11 ADC10_A Interrupts

The ADC10_A has six interrupt sources:

- ADC10IFG0 : conversion ready
- ADC10OVIFG : ADC10MEM0 overflow
- ADC10TOVIFG : ADC10_A conversion time overflow
- ADC10LOIFG, ADC10INIFG, ADC10HIIFG : window comparator interrupt flags

The ADC10IFG0 bit is set when the ADC10MEM0 memory register is loaded with the conversion result. An interrupt request is generated if ADC10IE0 bit and the GIE bit are set. The ADC10OV condition occurs when a conversion result is written to the ADC10MEM0 before its previous conversion result was read. The ADC10TOV condition is generated when another sample-and-conversion is requested before the current conversion is completed. The DMA is triggered after each conversion with the ADC10IE0 bit reset.

The window comparator interrupt flags are set corresponding to the description in the Window Comparator section (see [Section 1.2.8](#)).

1.2.11.1 ADC10IV, Interrupt Vector Generator

All ADC10_A interrupt sources are prioritized and combined to source a single interrupt vector. The interrupt vector register ADC10IV is used to determine which enabled ADC10_A interrupt source requested an interrupt.

The highest-priority enabled ADC10_A interrupt generates a number in the ADC10IV register (see register description). This number can be evaluated or added to the program counter (PC) to automatically enter the appropriate software routine. Disabled ADC10_A interrupts do not affect the ADC10IV value.

Read access of the ADC10IV register automatically resets the highest-pending interrupt condition and flag. Only the ADC10IFG0 is not reset by this ADC10IV read access. ADC10IFG0 is automatically reset by reading the ADC10MEM0 register or may be reset with software.

Write access of the ADC10IV register clears all pending interrupt conditions and flags.

If another interrupt is pending after servicing of an interrupt, another interrupt is generated. For example, if the ADC10OV, ADC10HIIFG and ADC10IFG0 interrupts are pending when the interrupt service routine accesses the ADC10IV register, the highest priority interrupt (ADC10OV interrupt condition) is reset automatically. After the RETI instruction of the interrupt service routine is executed, the ADC10HIIFG generates another interrupt.

1.2.11.2 ADC10_A Interrupt Handling Software Example

The following software example shows the recommended use of the ADC10IV. The ADC10IV value is added to the PC to automatically jump to the appropriate routine.

- ADC10IFG0, ADC10TOV, and ADC10OV: 16 cycles

```

; Interrupt handler for ADC10_A.
INT_ADC10_A                                ; Enter Interrupt Service Routine
ADD    &ADC10IV,PC                          ; Add offset to PC
RETI                                        ; Vector 0: No interrupt
JMP    ADOV                                  ; Vector 2: ADC10_A overflow
JMP    ADTOV                                 ; Vector 4: ADC10_A timing overflow
JMP    ADHI                                  ; Vector 6: ADC10_A window comparator high
interrupt
JMP    ADLO                                  ; Vector 8: ADC10_A window comparator low
interrupt
JMP    ADIN                                  ; Vector 10: ADC10_A window comparator in
interrupt
;
; Handler for ADC10IFG0 starts here. No JMP required.
;
ADMEM  MOV &ADC10MEM0,xxx                    ; Move result, flag is reset
      ...                                     ; Other instruction needed?
      RETI                                    ; Return ;
ADOV   ...                                     ; Handle ADCMEM0 overflow
      RETI                                    ; Return ;
ADTOV  ...                                     ; Handle Conv. time overflow
      RETI                                    ; Return ;
ADHI   ...                                     ; Handle window comparator high interrupt
      RETI                                    ; Return ;
ADLO   ...                                     ; Handle window comparator low interrupt
      RETI                                    ; Return ;
ADIN   ...                                     ; Handle window comparator in window interrupt
      RETI                                    ; Return
    
```

1.3 ADC10_A Registers

The ADC10_A registers are listed in [Table 1-2](#). The base address of the ADC10_A can be found in the device-specific data sheet. The address offset of each ADC10_A register is given in [Table 1-2](#).

Table 1-2. ADC10_A Registers

Offset	Acronym	Register Name	Type	Reset	Section
00h	ADC10CTL0	ADC10_A Control 0 register	Read/write	0000h	Section 1.3.1
02h	ADC10CTL1	ADC10_A Control 1 register	Read/write	0000h	Section 1.3.2
04h	ADC10CTL2	ADC10_A Control 2 register	Read/write	1000h	Section 1.3.3
06h	ADC10LO	ADC10_A Window Comparator Low Threshold register	Read/write	0000h	Section 1.3.9
08h	ADC10HI	ADC10_A Window Comparator High Threshold register	Read/write	FF03h	Section 1.3.7
0Ah	ADC10MCTL0	ADC10_A Memory Control register	Read/write	00h	Section 1.3.6
12h	ADC10MEM0	ADC10_A Conversion Memory register	Read/write	undefined	Section 1.3.4
1Ah	ADC10IE	ADC10_A Interrupt Enable register	Read/write	0000h	Section 1.3.11
1Ch	ADC10IFG	ADC10_A Interrupt Flag register	Read/write	0000h	Section 1.3.12
1Eh	ADC10IV	ADC10_A Interrupt Vector register	Read/write	0000h	Section 1.3.13

1.3.1 ADC10CTL0 Register

ADC10_A Control Register 0

Figure 1-12. ADC10CTL0 Register

15	14	13	12	11	10	9	8
Reserved				ADC10SHTx			
r0	r0	r0	r0	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
ADC10MSC	Reserved		ADC10ON	Reserved		ADC10ENC	ADC10SC
rw-(0)	r0	r0	rw-(0)	r0	r0	rw-(0)	rw-(0)

Can be modified only when ADC10ENC = 0. Resetting ADC10ENC = 0 by software and changing these fields immediately shows effect when a conversion is active.

Table 1-3. ADC10CTL0 Register Description

Bit	Field	Type	Reset	Description
15-12	Reserved	R	0h	Reserved. Always reads as 0.
11-8	ADC10SHTx	RW	0h	ADC10_A sample-and-hold time. These bits define the number of ADC10CLK cycles in the sampling period for the ADC10. 0000b = 4 ADC10CLK cycles 0001b = 8 ADC10CLK cycles 0010b = 16 ADC10CLK cycles 0011b = 32 ADC10CLK cycles 0100b = 64 ADC10CLK cycles 0101b = 96 ADC10CLK cycles 0110b = 128 ADC10CLK cycles 0111b = 192 ADC10CLK cycles 1000b = 256 ADC10CLK cycles 1001b = 384 ADC10CLK cycles 1010b = 512 ADC10CLK cycles 1011b = 768 ADC10CLK cycles 1100b = 1024 ADC10CLK cycles 1101b = 1024 ADC10CLK cycles 1110b = 1024 ADC10CLK cycles 1111b = 1024 ADC10CLK cycles
7	ADC10MSC	RW	0h	ADC10_A multiple sample and conversion. Valid only for sequence or repeated modes. 0b = The sampling timer requires a rising edge of the SHI signal to trigger each sample-and-convert. 1b = The first rising edge of the SHI signal triggers the sampling timer, but further sample-and-conversions are performed automatically as soon as the prior conversion is completed.
6-5	Reserved	R	0h	Reserved. Always reads as 0.
4	ADC10ON	RW	0h	ADC10_A on 0b = ADC10_A off 1b = ADC10_A on
3-2	Reserved	R	0h	Reserved. Always reads as 0.
1	ADC10ENC	RW	0h	ADC10_A enable conversion 0b = ADC10_A disabled 1b = ADC10_A enabled
0	ADC10SC	RW	0h	ADC10_A start conversion. Software-controlled sample-and-conversion start. ADC10SC and ADC10ENC may be set together with one instruction. ADC10SC is reset automatically. 0b = No sample-and-conversion-start 1b = Start sample-and-conversion

1.3.2 ADC10CTL1 Register

ADC10_A Control Register 1

Figure 1-13. ADC10CTL1 Register

15	14	13	12	11	10	9	8
Reserved				ADC10SHSx		ADC10SHP	ADC10ISSH
r0	r0	r0	r0	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
ADC10DIVx			ADC10SSELx		ADC10CONSEQx		ADC10BUSY
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r-(0)

Can be modified only when ADC10ENC = 0. Resetting ADC10ENC = 0 by software and changing these fields immediately shows effect when a conversion is active.

Table 1-4. ADC10CTL1 Register Description

Bit	Field	Type	Reset	Description
15-12	Reserved	R	0h	Reserved. Always reads as 0.
11-10	ADC10SHSx	RW	0h	ADC10_A sample-and-hold source select Can be modified only when ADC10ENC = 0. Resetting ADC10ENC = 0 by software and changing these fields immediately shows effect also when a conversion is active. 00b = ADC10SC bit 01b = Timer trigger 0 - see the device-specific data sheet 10b = Timer trigger 1 - see the device-specific data sheet 11b = Timer trigger 2 - see the device-specific data sheet
9	ADC10SHP	RW	0h	ADC10_A sample-and-hold pulse-mode select. This bit selects the source of the sampling signal (SAMPCON) to be either the output of the sampling timer or the sample-input signal directly. Can be modified only when ADC10ENC = 0. Resetting ADC10ENC = 0 by software and changing these fields immediately shows effect also when a conversion is active. 0b = SAMPCON signal is sourced from the sample-input signal 1b = SAMPCON signal is sourced from the sampling timer
8	ADC10ISSH	RW	0h	ADC10_A invert signal sample-and-hold Can be modified only when ADC10ENC = 0. Resetting ADC10ENC = 0 by software and changing these fields immediately shows effect also when a conversion is active. 0b = The sample-input signal is not inverted 1b = The sample-input signal is inverted
7-5	ADC10DIVx	RW	0h	ADC10_A clock divider Can be modified only when ADC10ENC = 0. Resetting ADC10ENC = 0 by software and changing these fields immediately shows effect also when a conversion is active. 000b = Divide by 1 001b = Divide by 2 010b = Divide by 3 011b = Divide by 4 100b = Divide by 5 101b = Divide by 6 110b = Divide by 7 111b = Divide by 8

Table 1-4. ADC10CTL1 Register Description (continued)

Bit	Field	Type	Reset	Description
4-3	ADC10SSELx	RW	0h	ADC10_A clock source select Can be modified only when ADC10ENC = 0. Resetting ADC10ENC = 0 by software and changing these fields immediately shows effect also when a conversion is active. 00b = MODCLK 01b = ACLK 10b = MCLK 11b = SMCLK
2-1	ADC10CONSEQx	RW	0h	ADC10_A conversion sequence mode select Can be modified only when ADC10ENC = 0. Resetting ADC10ENC = 0 by software and changing these fields immediately shows effect also when a conversion is active. 00b = Single-channel, single-conversion 01b = Sequence-of-channels 10b = Repeat-single-channel 11b = Repeat-sequence-of-channels
0	ADC10BUSY	R	0h	ADC10_A busy. This bit indicates an active sample or conversion operation. 0b = No operation is active 1b = A sequence, sample, or conversion is active

1.3.3 ADC10CTL2 Register

ADC10_A Control Register 2

Figure 1-14. ADC10CTL2 Register

15	14	13	12	11	10	9	8
Reserved						ADC10PDIVx	
r0	r0	r0	r0	r0	r0	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
Reserved			ADC10RES	ADC10DF	ADC10SR	Reserved	
r0	r0	r0	rw-(1)	rw-(0)	rw-(0)	r0	r0

Can be modified only when ADC10ENC = 0. Resetting ADC10ENC = 0 by software and changing these fields immediately shows effect when a conversion is active.

Table 1-5. ADC10CTL2 Register Description

Bit	Field	Type	Reset	Description
15-10	Reserved	R	0h	Reserved. Always reads as 0.
9-8	ADC10PDIVx	RW	0h	ADC10_A predivider. This bit predivides the selected ADC10_A clock source before it gets divided again using ADC10DIVx. 00b = Predivide by 1 01b = Predivide by 4 10b = Predivide by 64 11b = Reserved
7-5	Reserved	R	0h	Reserved. Always reads as 0.
4	ADC10RES	RW	1h	ADC10_A resolution. This bit defines the conversion result resolution. 0b = 8 bit (10 clock cycle conversion time) 1b = 10 bit (12 clock cycle conversion time)
3	ADC10DF	RW	0h	ADC10_A data read-back format. Data is always stored in the binary unsigned format. 0b = Binary unsigned. Theoretically, the analog input voltage -V(REF) results in 0000h, and the analog input voltage +V(REF) results in 03FFh. 1b = Signed binary (twos complement), left aligned. Theoretically, the analog input voltage -V(REF) results in 8000h, and the analog input voltage +V(REF) results in 7FC0h.
2	ADC10SR	RW	0h	ADC10_A sampling rate. This bit selects drive capability of the ADC10_A reference buffer for the maximum sampling rate. Setting ADC10SR reduces the current consumption of this buffer. 0b = ADC10_A buffer supports up to approximately 200 ksps. 1b = ADC10_A buffer supports up to approximately 50 ksps.
1-0	Reserved	R	0h	Reserved. Always reads as 0.

1.3.4 ADC10MEM0 Register

ADC10_A Conversion Memory Register

Figure 1-15. ADC10MEM0 Register

15	14	13	12	11	10	9	8
Reserved						Conversion_Results	
r0	r0	r0	r0	r0	r0	rw	rw
7	6	5	4	3	2	1	0
Conversion_Results							
rw	rw	rw	rw	rw	rw	rw	rw

Table 1-6. ADC10MEM0 Register Description

Bit	Field	Type	Reset	Description
15-10	Reserved	R	0h	Reserved. Always reads as 0.
9-0	Conversion_Results	RW	undefined	The 10-bit conversion results are right justified. Bit 9 is the MSB. Bits 15-10 are 0 in 10-bit mode, and bits 15-8 are 0 in 8-bit mode. Writing to the conversion memory register corrupts the results. This data format is used if ADC10DF = 0.

1.3.5 ADC10MEM0 Register, Twos-Complement Format

ADC10_A Conversion Memory Register, Twos-Complement Format

Figure 1-16. ADC10MEM0 Register

15	14	13	12	11	10	9	8
Conversion_Results							
rw	rw	rw	rw	rw	rw	rw	rw
7	6	5	4	3	2	1	0
Conversion_Results		Reserved					
rw	rw	r0	r0	r0	r0	r0	r0

Table 1-7. ADC10MEM0 Register Description

Bit	Field	Type	Reset	Description
15-6	Conversion_Results	RW	undefined	The 10-bit conversion results are left justified, twos-complement format. Bit 15 is the MSB. Bits 5-0 are 0 in 10-bit mode, and bits 7-0 are 0 in 8-bit mode. This data format is used if ADC10DF = 1. The data is stored in the right-justified format and is converted to the left-justified twos-complement format during read back. Writing to the conversion memory register corrupts the results.
5-0	Reserved	R	0h	Reserved. Always reads as 0.

1.3.6 ADC10MCTL0 Register

ADC10_A Conversion Memory Control Register

Figure 1-17. ADC10MCTL0 Register

7	6	5	4	3	2	1	0
Reserved	ADC10SREFx			ADC10INCHx			
r0	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Can be modified only when ADC10ENC = 0. Resetting ADC10ENC = 0 by software and changing these fields immediately shows effect when a conversion is active.

Table 1-8. ADC10MCTL0 Register Description

Bit	Field	Type	Reset	Description
7	Reserved	R	0h	Reserved. Always reads as 0.
6-4	ADC10SREFx	RW	0h	<p>Select reference. It is not recommended to change this setting while a conversion is ongoing.</p> <p>Can be modified only when ADC10ENC = 0. Resetting ADC10ENC = 0 by software and changing these fields immediately shows effect also when a conversion is active.</p> <p>000b = $V_{R+} = AVCC$ and $V_{R-} = AVSS$ 001b = $V_{R+} = VREF$ and $V_{R-} = AVSS$ 010b = $V_{R+} = VREF+$ buffered and $V_{R-} = AVSS$ 011b = $V_{R+} = VREF+$ and $V_{R-} = AVSS$ 100b = $V_{R+} = AVCC$ and $V_{R-} = VREF-$ 101b = $V_{R+} = VREF$ and $V_{R-} = VREF-$ 110b = $V_{R+} = VREF+$ buffered and $V_{R-} = VREF-$ 111b = $V_{R+} = VREF+$ and $V_{R-} = VREF-$</p>
3-0	ADC10INCHx	RW	0h	<p>Input channel select. Writing these bits select the channel for a single-conversion or the highest channel for a sequence of conversions. Reading these bits in ADC10CONSEQ = 01 or 11 returns the channel currently converted. ADC10INCHx is not synchronized, so a read while the state machine is not in "wait for enable" or "wait for trigger" could lead to a wrong result.</p> <p>Can be modified only when ADC10ENC = 0. Resetting ADC10ENC = 0 by software and changing these fields immediately shows effect also when a conversion is active.</p> <p>0000b = A0 0001b = A1 0010b = A2 0011b = A3 0100b = A4 0101b = A5 0110b = A6 0111b = A7 1000b = A8 1001b = A9 1010b = A10 1011b = A11 1100b = A12 1101b = A13 1110b = A14 1111b = A15</p>

1.3.7 ADC10HI Register

ADC10_A Window Comparator High Threshold Register

Figure 1-18. ADC10HI Register

15	14	13	12	11	10	9	8
Reserved						High_Threshold	
r0	r0	r0	r0	r0	r0	rw-(1)	rw-(1)
7	6	5	4	3	2	1	0
High_Threshold							
rw-(1)	rw-(1)	rw-(1)	rw-(1)	rw-(1)	rw-(1)	rw-(1)	rw-(1)

Table 1-9. ADC10HI Register Description

Bit	Field	Type	Reset	Description
15-10	Reserved	R	0h	Reserved. Always reads as 0.
9-0	High_Threshold	RW	3FFh	The 10-bit threshold value needs to be right justified. Bit 9 is the MSB. Bits 15-10 are 0 in 10-bit mode, and bits 15-8 are 0 in 8-bit mode. This data format is used if ADC10DF = 0.

1.3.8 ADC10HI Register, Twos-Complement Format

ADC10_A Window Comparator High Threshold Register, Twos-Complement Format

Figure 1-19. ADC10HI Register

15	14	13	12	11	10	9	8
High_Threshold							
rw-(0)	rw-(1)	rw-(1)	rw-(1)	rw-(1)	rw-(1)	rw-(1)	rw-(1)
7	6	5	4	3	2	1	0
High_Threshold		Reserved					
rw-(1)	rw-(1)	r0	r0	r0	r0	r0	r0

Table 1-10. ADC10HI Register Description

Bit	Field	Type	Reset	Description
15-6	High_Threshold	RW	1FFh	The 10-bit threshold value needs to be left justified if twos-complement format is chosen. Bit 15 is the MSB. Bits 5-0 are 0 in 10-bit mode, and bits 7-0 are 0 in 8-bit mode. This data format is used if ADC10DF = 1.
5-0	Reserved	R	0h	Reserved. Always reads as 0.

1.3.9 ADC10LO Register

ADC10_A Window Comparator Low Threshold Register

Figure 1-20. ADC10LO Register

15	14	13	12	11	10	9	8
Reserved						Low_Threshold	
r0	r0	r0	r0	r0	r0	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
Low_Threshold							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Table 1-11. ADC10LO Register Description

Bit	Field	Type	Reset	Description
15-10	Reserved	R	0h	Reserved. Always reads as 0.
9-0	Low_Threshold	RW	0h	The 10-bit threshold value needs to be right justified. Bit 9 is the MSB. Bits 15-10 are 0 in 10-bit mode, and bits 15-8 are 0 in 8-bit mode. This data format is used if ADC10DF = 0.

1.3.10 ADC10LO Register, Twos-Complement Format

ADC10_A Window Comparator Low Threshold Register, Twos-Complement Format

Figure 1-21. ADC10LO Register

15	14	13	12	11	10	9	8
Low_Threshold							
rw-(1)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
Low_Threshold		Reserved					
rw-(0)	rw-(0)	r0	r0	r0	r0	r0	r0

Table 1-12. ADC10LO Register Description

Bit	Field	Type	Reset	Description
15-6	Low_Threshold	RW	200h	The 10-bit threshold value needs to be left justified if twos-complement format is chosen. Bit 15 is the MSB. Bits 5-0 are 0 in 10-bit mode, and bits 7-0 are 0 in 8-bit mode. This data format is used if ADC10DF = 1.
5-0	Reserved	R	0h	Reserved. Always reads as 0.

1.3.11 ADC10IE Register

ADC10_A Interrupt Enable Register

Figure 1-22. ADC10IE Register

15	14	13	12	11	10	9	8
Reserved							
r0	r0	r0	r0	r0	r0	r0	r0
7	6	5	4	3	2	1	0
Reserved		ADC10TOVIE	ADC10OVIE	ADC10HIIE	ADC10LOIE	ADC10INIE	ADC10IE0
r0	r0	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Table 1-13. ADC10IE Register Description

Bit	Field	Type	Reset	Description
15-6	Reserved	R	0h	Reserved. Always reads as 0.
5	ADC10TOVIE	RW	0h	ADC10_A conversion-time-overflow interrupt enable. 0b = Conversion time overflow interrupt disabled 1b = Conversion time overflow interrupt enabled
4	ADC10OVIE	RW	0h	ADC10MEM0 overflow interrupt enable. 0b = Overflow interrupt disabled 1b = Overflow interrupt enabled
3	ADC10HIIE	RW	0h	Interrupt enable for the above upper threshold interrupt of the window comparator. 0b = Above upper threshold interrupt disabled 1b = Above upper threshold interrupt enabled
2	ADC10LOIE	RW	0h	Interrupt enable for the below lower threshold interrupt of the window comparator. 0b = Below lower threshold interrupt disabled 1b = Below lower threshold interrupt enabled
1	ADC10INIE	RW	0h	Interrupt enable for the inside of window interrupt of the window comparator. 0b = Inside of window interrupt disabled 1b = Inside of window interrupt enabled
0	ADC10IE0	RW	0h	Interrupt enable. This bits enable or disable the interrupt request for a completed ADC10_A conversion. 0b = Interrupt disabled 1b = Interrupt enabled

1.3.12 ADC10IFG Register

ADC10_A Interrupt Flag Register

Figure 1-23. ADC10IFG Register

15	14	13	12	11	10	9	8
Reserved							
r0	r0	r0	r0	r0	r0	r0	r0
7	6	5	4	3	2	1	0
Reserved		ADC10TOVIFG	ADC10OVIFG	ADC10HIIFG	ADC10LOIFG	ADC10INIFG	ADC10IFG0
r0	r0	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Table 1-14. ADC10IFG Register Description

Bit	Field	Type	Reset	Description
15-6	Reserved	R	0h	Reserved. Always reads as 0.
5	ADC10TOVIFG	RW	0h	The ADC10TOVIFG is set when an ADC10_A conversion is triggered before the actual conversion has completed. 0b = No interrupt pending 1b = Interrupt pending
4	ADC10OVIFG	RW	0h	The ADC10OVIFG is set when the ADC10MEM0 register is written before the last conversion result has been read. 0b = No interrupt pending 1b = Interrupt pending
3	ADC10HIIFG	RW	0h	The ADC10HIIFG is set when the result of the current ADC10_A conversion is greater than the upper threshold defined by the window comparator's upper threshold register. 0b = No interrupt pending 1b = Interrupt pending
2	ADC10LOIFG	RW	0h	The ADC10LOIFG is set when the result of the current ADC10_A conversion is below the lower threshold defined by the window comparator's lower threshold register. 0b = No interrupt pending 1b = Interrupt pending
1	ADC10INIFG	RW	0h	The ADC10INIFG is set when the result of the current ADC10_A conversion is within the thresholds defined by the window comparator's threshold registers. 0b = No interrupt pending 1b = Interrupt pending
0	ADC10IFG0	RW	0h	The ADC10IFG0 is set when an ADC10_A conversion is completed. This bit gets reset, when the ADC10MEM0 get read, or may be reset by software. 0b = No interrupt pending 1b = Interrupt pending

1.3.13 ADC10IV Register

ADC10_A Interrupt Vector Register

Figure 1-24. ADC10IV Register

15	14	13	12	11	10	9	8
ADC10IVx							
r0	r0	r0	r0	r0	r0	r0	r0
7	6	5	4	3	2	1	0
ADC10IVx							
r0	r0	r0	r0	r-(0)	r-(0)	r-(0)	r0

Table 1-15. ADC10IV Register Description

Bit	Field	Type	Reset	Description
15-0	ADC10IVx	R	0h	ADC10_A interrupt vector value. It generates an value that can be used as address offset for fast interrupt service routine handling. Writing to this register clears all pending interrupt flags. 00h = No interrupt pending 02h = Interrupt Source: ADC10MEM0 overflow; Interrupt Flag: ADC10OVIFG; Interrupt Priority: Highest 04h = Interrupt Source: Conversion time overflow; Interrupt Flag: ADC10TOVIFG 06h = Interrupt Source: ADC10HI Interrupt flag; Interrupt Flag: ADC10HIIFG 08h = Interrupt Source: ADC10LO Interrupt flag; Interrupt Flag: ADC10LOIFG 0Ah = Interrupt Source: ADC10IN Interrupt flag; Interrupt Flag: ADC10INIFG 0Ch = Interrupt Source: ADC10_A memory Interrupt flag; Interrupt Flag: ADC10IFG0; Interrupt Priority: Lowest

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ("TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources 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.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products (<http://www.ti.com/sc/docs/stdterms.htm>), [evaluation modules](#), and [samples](http://www.ti.com/sc/docs/sampterm.htm) (<http://www.ti.com/sc/docs/sampterm.htm>).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2018, Texas Instruments Incorporated