



ADC12_A

The ADC12_A module is a high-performance 12-bit analog-to-digital converter (ADC). This chapter describes the operation of the ADC12_A module.

| Topic | Page |
|---------------------------------|------|
| 28.1 ADC12_A Introduction | 723 |
| 28.2 ADC12_A Operation | 726 |
| 28.3 ADC12_A Registers | 740 |

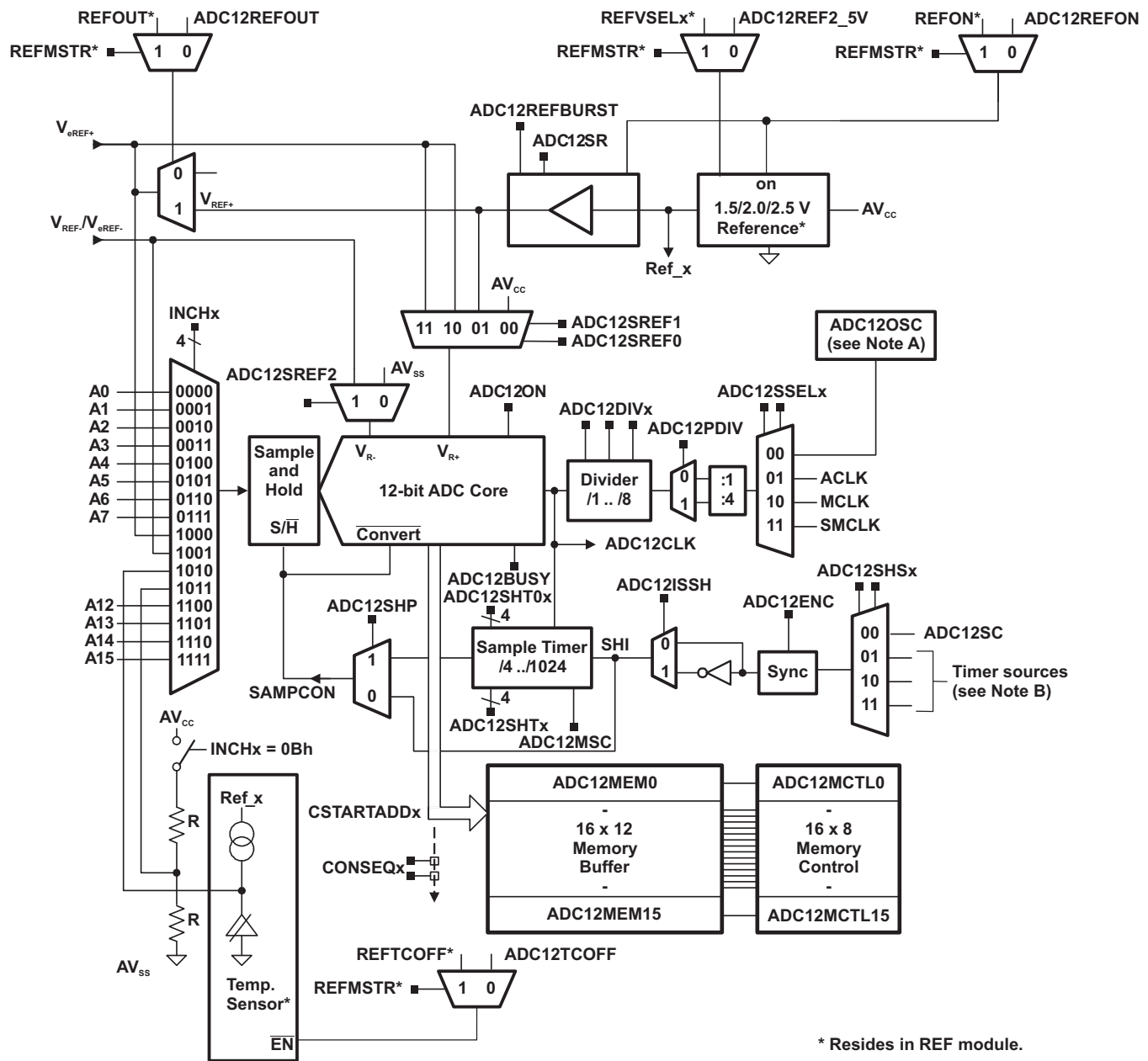
28.1 ADC12_A Introduction

The ADC12_A module supports fast 12-bit analog-to-digital conversions. The module implements a 12-bit SAR core, sample select control, reference generator (MSP430F54xx (non-A only) – in other devices, separate REF module), and a 16-word conversion-and-control buffer. The conversion-and-control buffer allows up to 16 independent analog-to-digital converter (ADC) samples to be converted and stored without any CPU intervention.

ADC12_A features include:

- Greater than 200-ksps maximum conversion rate
- Monotonic 12-bit converter with no missing codes
- Sample-and-hold with programmable sampling periods controlled by software or timers
- Conversion initiation by software or timers
- Software-selectable on-chip reference voltage generation (MSP430F54xx (non-A only): 1.5 V or 2.5 V, all other devices: 1.5 V, 2.0 V, or 2.5 V)
- Software-selectable internal or external reference
- Up to 12 individually configurable external input channels
- Conversion channels for internal temperature sensor, AV_{CC} , and external references
- Independent channel-selectable reference sources for both positive and negative references
- Selectable conversion clock source
- Single-channel, repeat-single-channel, sequence (autoscan), and repeat-sequence (repeated autoscan) conversion modes
- ADC core and reference voltage can be powered down separately
- Interrupt vector register for fast decoding of 18 ADC interrupts
- 16 conversion-result storage registers

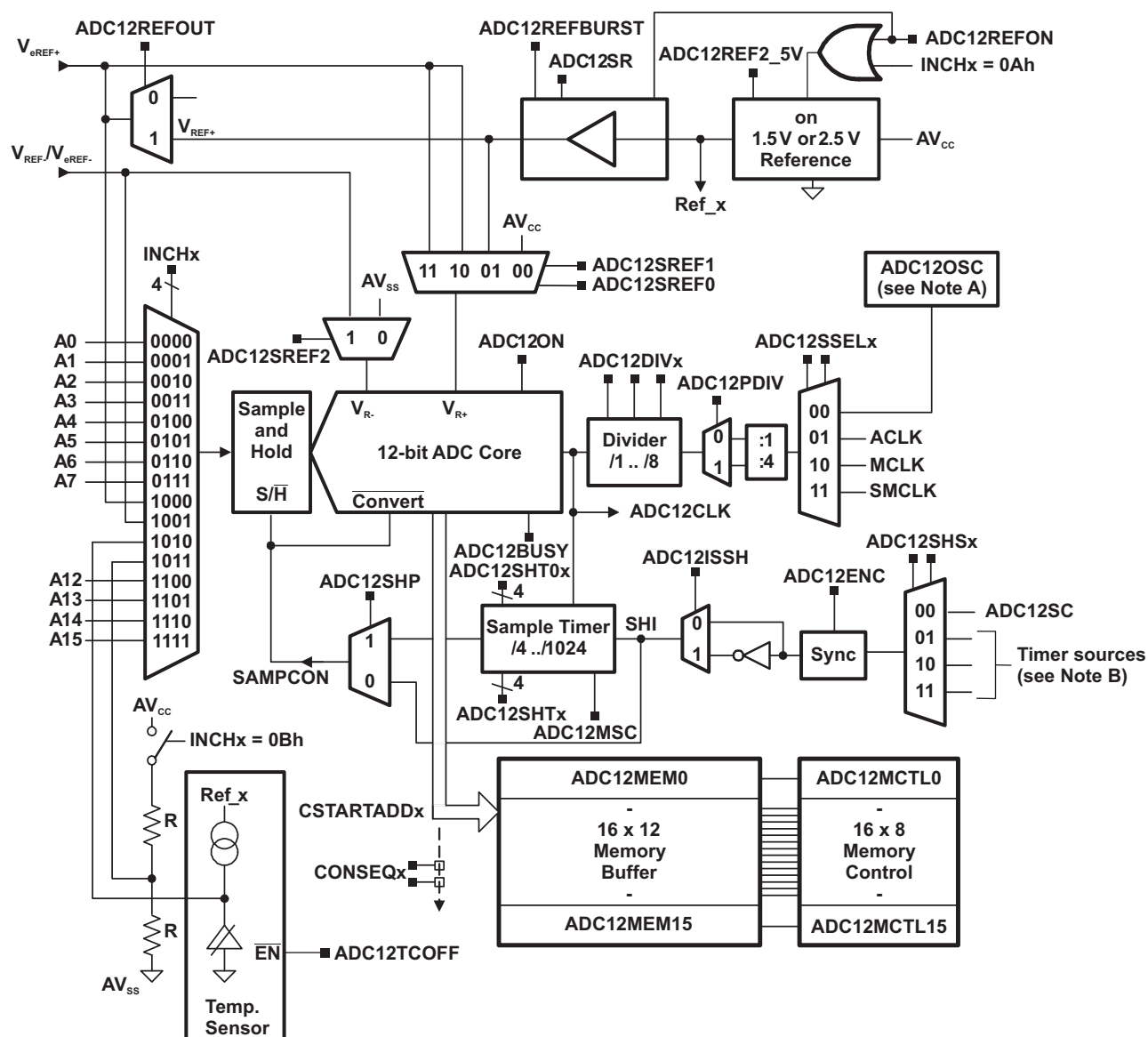
The block diagram of ADC12_A is shown in [Figure 28-1](#). In MSP430F54xx (non-A only), the reference generator is located in the ADC12_A module itself. In other devices, the reference generator is located in the reference module, REF. See the REF module chapter and the device-specific data sheet for further details. [Figure 28-1](#) shows the block diagram for devices that have the REF module available. [Figure 28-2](#) shows the block diagram for the MSP430F54xx (non-A only) which does not incorporate the REF module.



A The MODOSC is part of the UCS. See the [UCS chapter](#) for more information.

B See the device-specific data sheet for timer sources available.

Figure 28-1. ADC12_A Block Diagram (Devices With REF Module)



- A The MODOSC is part of the UCS. See the [UCS chapter](#) for more information.
B See the device-specific data sheet for timer sources available.

Figure 28-2. ADC12_A MSP430F54xx (non-A) Block Diagram

28.2 ADC12_A Operation

The ADC12_A module is configured with user software. The setup and operation of the ADC12_A is discussed in the following sections.

28.2.1 12-Bit ADC Core

The ADC core converts an analog input to its 12-bit digital representation and stores the result in conversion memory. The core uses two programmable and 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 (0FFFh) when the input signal is equal to or higher than V_{R+} . The digital output (N_{ADC}) is 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. The conversion formula for the ADC result N_{ADC} is:

$$N_{ADC} = 4095 \times \frac{V_{in} - V_{R-}}{V_{R+} - V_{R-}}$$

The ADC12_A core is configured by two control registers, ADC12CTL0 and ADC12CTL1. The core is enabled with the ADC12ON bit. The ADC12_A can be turned off when it is not in use to save power. With few exceptions, the ADC12_A control bits can be modified only when ADC12ENC = 0. ADC12ENC must be set to 1 before any conversion can take place.

28.2.1.1 Conversion Clock Selection

The ADC12CLK is used both as the conversion clock and to generate the sampling period when the pulse sampling mode is selected. The ADC12_A source clock is selected using the predivider controlled by the ADC12PDIV bit and the divider using the ADC12SSELx bits. The input clock can be divided from 1 to 32 using both the ADC12DIVx bits and the ADC12PDIV bit. Possible ADC12CLK sources are SMCLK, MCLK, ACLK, and the ADC12OSC.

The ADC12OSC in the block diagram (see [Figure 28-1](#)) refers to the MODOSC 5-MHz oscillator from the UCS (see the UCS module for more information) which can vary with individual devices, supply voltage, and temperature. See the device-specific data sheet for the ADC12OSC specification.

The user must ensure that the clock chosen for ADC12CLK remains active until the end of a conversion. If the clock is removed during a conversion, the operation does not complete and the results are invalid.

28.2.2 ADC12_A Inputs and Multiplexer

The 12 external and 4 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 28-3](#)). The input multiplexer is also a T-switch to minimize the coupling between channels. Channels that are not selected are isolated from the A/D and the intermediate node is connected to analog ground (AV_{SS}), so that the stray capacitance is grounded to eliminate crosstalk.

The ADC12_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 errant conversion.

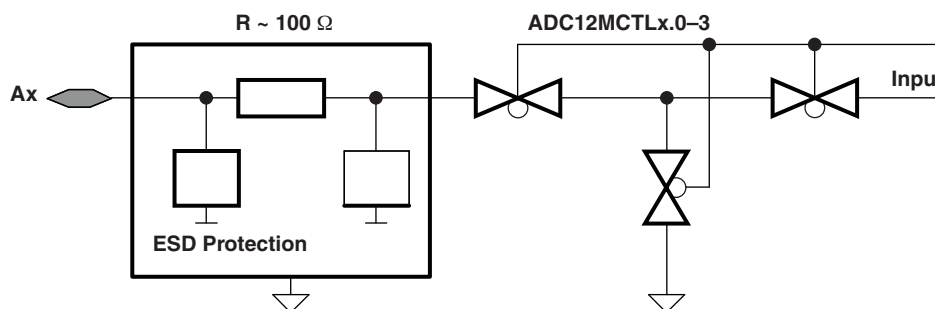


Figure 28-3. Analog Multiplexer

28.2.2.1 Analog Port Selection

The ADC12_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 PySELx bits provide the ability to disable the port pin input and output buffers.

```
; Py.0 and Py.1 configured for analog input
BIS.B #3h,&PySEL ; Py.1 and Py.0 ADC12_A function
```

28.2.3 Voltage Reference Generator

The ADC12_A modules have a separate reference module (REF) that supplies three selectable voltage levels, 1.5 V, 2.0 V, and 2.5 V to the ADC12_A. Any of these voltages may be used internally and externally on pin VREF+. The internal AV_{CC} can also be used as the reference.

The ADC12_A module of the MSP430F54xx devices (non-A only) does not use the REF module and only has two selectable voltage levels, 1.5 V and 2.5 V. The internal AV_{CC} can also be used as the reference.

On devices with the REF module, the voltage reference settings can be controlled either by the REF module or by the ADC12_A module. This is to allow for backward compatibility with older families. This is handled by the REFMSTR bit in the REF module. If REFMSTR = 1 (default), the REF module registers control the reference settings. If REFMSTR = 0, the ADC12_A reference setting define the reference voltage of the ADC12_A module. Four control settings that reside in the ADC12_A can be controlled also by four corresponding settings in the REF module: ADC12REF2_5V (REFVSEL), ADC12REFON (REFON), ADC12REFOUT (REFOUT), and ADC12TCOFF (REFTCOFF), respectively. When REFMSTR = 1, ADC12REF2_5V, ADC12REFON, ADC12REFOUT, and ADC12TCOFF are do not care. Similarly, when REFMSTR = 0, REFVSEL, REFON, REFOUT, and REFTCOFF are do not care. See the REF module chapter for further details.

On devices with the REF module, to use the ADC12_A reference control bits, set REFMSTR = 0. In this case, setting ADC12REFON = 1 enables the reference voltage of the ADC12_A module. When ADC12REF2_5V = 1, the internal reference is 2.5 V; when ADC12REF2_5V = 0, the reference is 1.5 V. Similarly, on devices with the REF module, to use the REF module reference control bits, set REFMSTR = 1. In this case, setting REFON = 1 of the REF module enables the reference voltage. The REFVSEL bits of the REF module can be used to select either 1.5 V, 2.0 V, or 2.5 V. The reference can be turned off to save power when not in use. On the MSP430F54xx devices (non-A only), as stated previously, the REF module is not present and behaves the same as devices the REF module with REFMSTR = 0.

External references may be supplied for V_{R+} and V_{R-} through pins VREF+/VeREF+ and VREF-/VeREF-, respectively.

External storage capacitors are required only if ADC12REFOUT = 1 (REFOUT = 1 when using REF module) and the reference voltage is made available at the pins.

28.2.3.1 Internal Reference Low-Power Features

The ADC12_A internal reference generator is designed for low-power applications. The reference generator includes a bandgap voltage source and a separate buffer. The current consumption and settling time of each is specified separately in the device-specific data sheet. When ADC12REFON = 1 (REFON = 1 when using REF module), both are enabled; when ADC12REFON = 0 (REFON = 0 when using REF module), both are disabled.

When ADC12REFON = 1 (REFON = 1 when using REF module) and ADC12REFBURST = 1 but no conversion is active, the buffer is automatically disabled and automatically reenabled when needed. When the buffer is disabled, it consumes no current. In this case, the bandgap voltage source remains enabled.

The ADC12REFBURST bit controls the operation of the reference buffer. When ADC12REFBURST = 1, the buffer is automatically disabled when the ADC12_A is not actively converting, and is automatically reenabled when needed. When ADC12REFBURST = 0, the buffer is on continuously. This allows the reference voltage to be present outside the device continuously if ADC12REFOUT = 1 (REFOUT = 1 when using REF module).

The internal reference buffer also has selectable speed versus power settings. When the maximum conversion rate is below 50 kps, setting ADC12SR = 1 reduces the current consumption of the buffer by approximately 50%.

28.2.4 Auto Power Down

The ADC12_A is designed for low-power applications. When the ADC12_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.

28.2.5 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 SHSx bits and includes the following:

- ADC12SC bit
- Up to three timer outputs (see the device-specific data sheet for available timer sources)

The ADC12_A supports 8-bit, 10-bit, and 12-bit resolution modes selectable by the ADC12RES bits. The analog-to-digital conversion requires 9, 11, and 13 ADC12CLK cycles, respectively. The polarity of the SHI signal source can be inverted with the ADC12ISSH 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. Two different sample-timing methods are defined by control bit ADC12SHP, extended sample mode, and pulse mode. See the device-specific data sheet for available timers for SHI sources.

28.2.5.1 Extended Sample Mode

The extended sample mode is selected when ADC12SHP = 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 ADC12CLK (see Figure 28-4).

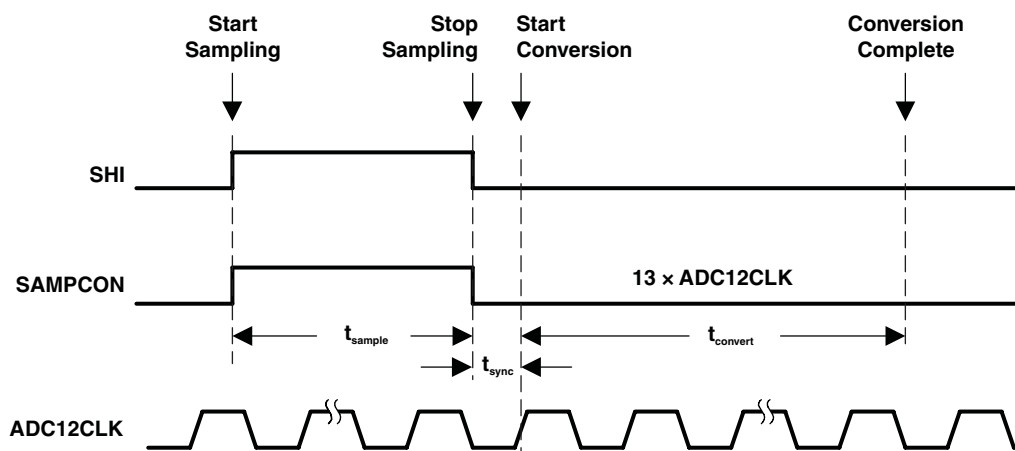
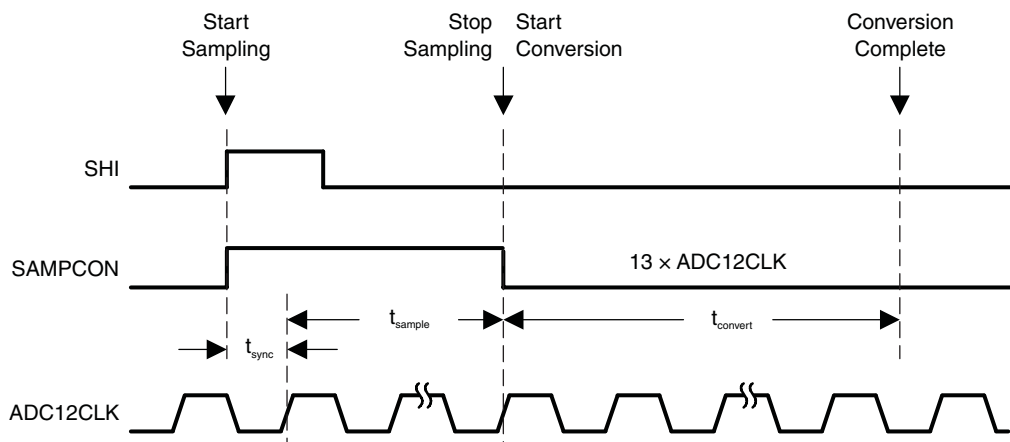


Figure 28-4. Extended Sample Mode

28.2.5.2 Pulse Sample Mode

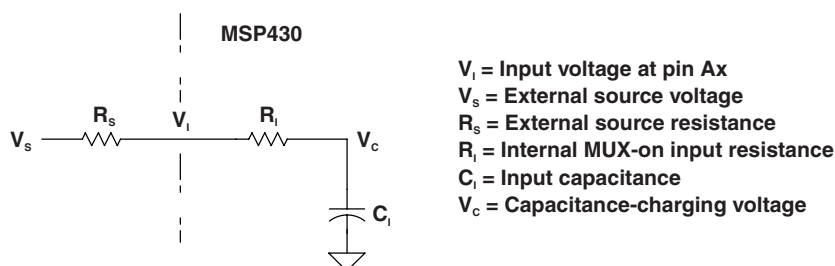
Set ADC12SHP = 1 to select the pulse sample mode. The SHI signal is used to trigger the sampling timer. The ADC12SHT0x and ADC12SHT1x bits in ADC12CTL0 control the interval of the sampling timer that defines the SAMPCON sample period t_{sample} . The sampling timer keeps SAMPCON high after synchronization with ADC12CLK for a programmed interval t_{sample} . The total sampling time is t_{sample} plus t_{sync} (see Figure 28-5).

The ADC12SHTx bits select the sampling time in 4x multiples of ADC12CLK. ADC12SHT0x selects the sampling time for ADC12MCTL0 to ADC12MCTL7. ADC12SHT1x selects the sampling time for ADC12MCTL8 to ADC12MCTL15.


Figure 28-5. Pulse Sample Mode

28.2.5.3 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 28-6). An internal MUX-on input resistance R_i (maximum 1.8 k Ω) in series with capacitor C_i (25 pF maximum) 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 n-bit conversion, where n is the bits of resolution required.


Figure 28-6. Analog Input Equivalent Circuit

The resistance of the source R_s and R_i affect t_{sample} . The following equation can be used to calculate the minimum sampling time t_{sample} for a n-bit conversion, where n equals the bits of resolution:

$$t_{\text{sample}} > (R_s + R_i) \times \ln(2^{n+1}) \times C_i + 800 \text{ ns}$$

Substituting the values for R_i and C_i given above, the equation becomes:

$$t_{\text{sample}} > (R_s + 1.8 \text{ k}\Omega) \times \ln(2^{n+1}) \times 25 \text{ pF} + 800 \text{ ns}$$

For example, for 12-bit resolution, if R_s is 10 k Ω , t_{sample} must be greater than 3.46 μs .

28.2.6 Conversion Memory

There are 16 ADC12MEMx conversion memory registers to store conversion results. Each ADC12MEMx is configured with an associated ADC12MCTLx control register. The SREFx bits define the voltage reference and the INCHx bits select the input channel. The ADC12EOS bit defines the end of sequence when a sequential conversion mode is used. A sequence rolls over from ADC12MEM15 to ADC12MEM0 when the ADC12EOS bit in ADC12MCTL15 is not set.

The CSTARTADDx bits define the first ADC12MCTLx used for any conversion. If the conversion mode is single-channel or repeat-single-channel, the CSTARTADDx points to the single ADC12MCTLx to be used.

If the conversion mode selected is either sequence-of-channels or repeat-sequence-of-channels, CSTARTADDx points to the first ADC12MCTLx location to be used in a sequence. A pointer, not visible to software, is incremented automatically to the next ADC12MCTLx in a sequence when each conversion completes. The sequence continues until an ADC12EOS bit in ADC12MCTLx is processed; this is the last control byte processed.

When conversion results are written to a selected ADC12MEMx, the corresponding flag in the ADC12IFGx register is set.

There are two formats available to store the conversion result, ADC12MEMx. When ADC12DF = 0, the conversion is right justified, unsigned. For 8-bit, 10-bit, and 12-bit resolutions, the upper 8, 6, and 4 bits of ADC12MEMx are always zeros, respectively. When ADC12DF = 1, the conversion result is left justified, two's complement. For 8-bit, 10-bit, and 12-bit resolutions, the lower 8, 6, and 4 bits of ADC12MEMx are always zeros, respectively. This is summarized in [Table 28-1](#).

Table 28-1. ADC12_A Conversion Result Formats

| Analog Input Voltage | ADC12DF | ADC12RES | Ideal Conversion Results | ADC12MEMx |
|--------------------------|---------|----------|--------------------------|---------------|
| $-V_{REF}$ to $+V_{REF}$ | 0 | 00 | 0 to 255 | 0000h - 00FFh |
| | 0 | 01 | 0 to 1023 | 0000h - 03FFh |
| | 0 | 10 | 0 to 4095 | 0000h - 0FFFh |
| | 1 | 00 | -128 to 127 | 8000h - 7F00h |
| | 1 | 01 | -512 to 511 | 8000h - 7FC0h |
| | 1 | 10 | -2048 to 2047 | 8000h - 7FF0h |

28.2.7 ADC12_A Conversion Modes

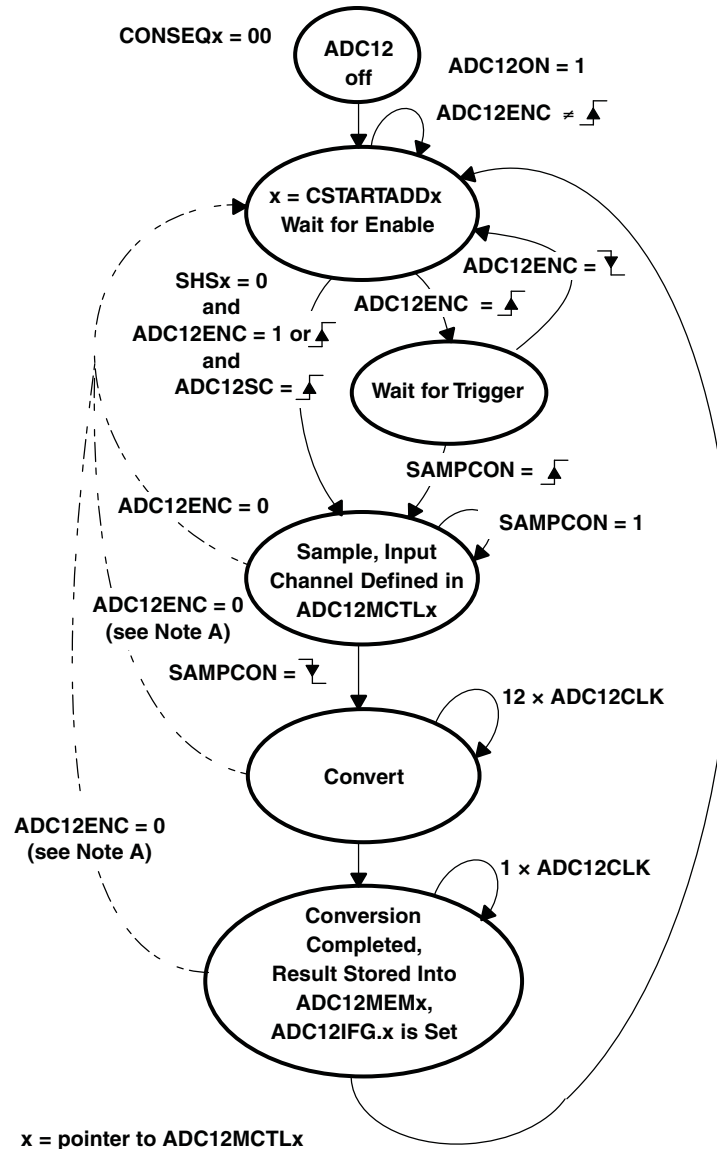
The ADC12_A has four operating modes selected by the CONSEQx bits as listed in [Table 28-2](#). All state diagrams assume a 12-bit resolution setting.

Table 28-2. Conversion Mode Summary

| ADC12CONSEQx | 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. |

28.2.7.1 Single-Channel Single-Conversion Mode

A single channel is sampled and converted once. The ADC result is written to the ADC12MEMx defined by the CSTARTADDx bits. Figure 28-7 shows the flow of the single-channel single-conversion mode. When ADC12SC triggers a conversion, successive conversions can be triggered by the ADC12SC bit. When any other trigger source is used, ADC12ENC must be toggled between each conversion.



A Conversion result is unpredictable.

Figure 28-7. Single-Channel Single-Conversion Mode

28.2.7.2 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 ADC results are written to the conversion memories starting with the ADCMEM_x defined by the CSTARTADD_x bits. The sequence stops after the measurement of the channel with a set ADC12EOS bit. Figure 28-8 shows the sequence-of-channels mode. When ADC12SC triggers a sequence, successive sequences can be triggered by the ADC12SC bit. The ADC12SC must be cleared by software after each sequence to trigger another sequence. When any other trigger source is used, ADC12ENC must be toggled between each sequence.

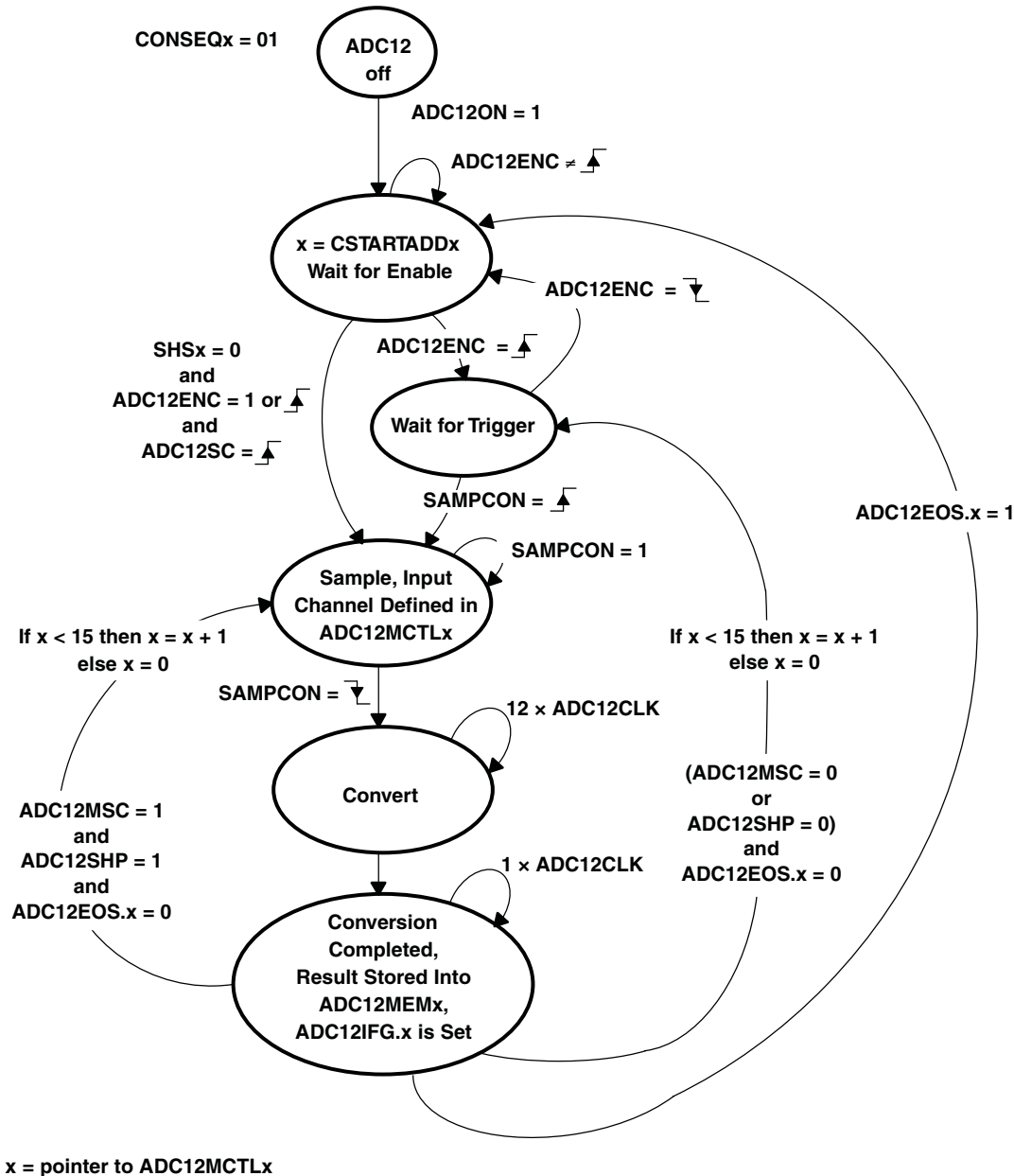


Figure 28-8. Sequence-of-Channels Mode

28.2.7.3 Repeat-Single-Channel Mode

A single channel is sampled and converted continuously. The ADC results are written to the ADC12MEMx defined by the CSTARTADDx bits. It is necessary to read the result after the completed conversion because only one ADC12MEMx memory is used and is overwritten by the next conversion. [Figure 28-9](#) shows the repeat-single-channel mode.

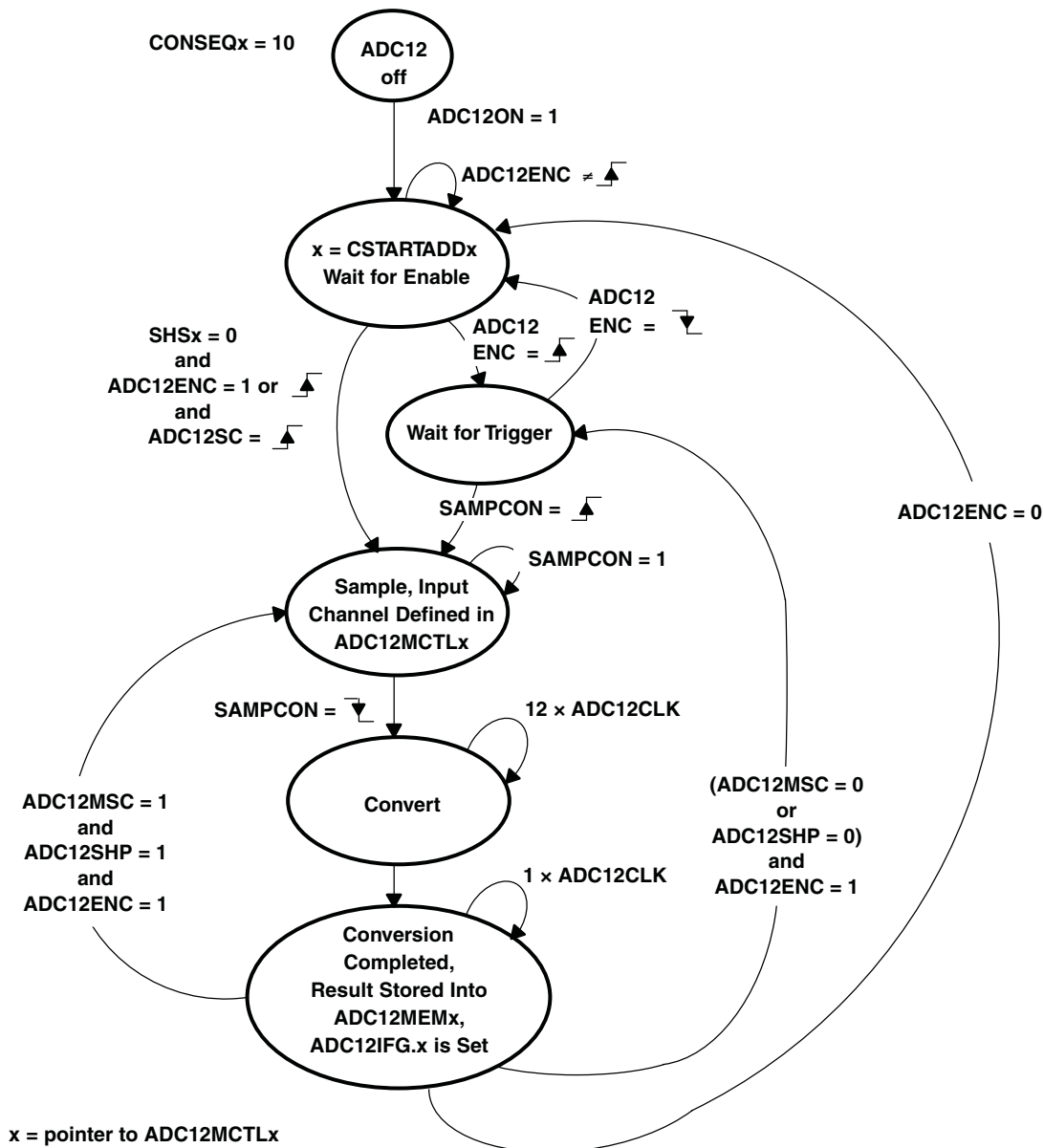


Figure 28-9. Repeat-Single-Channel Mode

In this mode, a sequence of channels is sampled and converted repeatedly. This mode is also referred to as repeated autoscan mode. The ADC results are written to the conversion memories starting with the ADC12MEMx defined by the CSTARTADDx bits. The sequence ends after the measurement of the channel with a set ADC12EOS bit and the next trigger signal restarts the sequence. [Figure 28-10](#) shows the repeat-sequence-of-channels mode.



28.2.7.5 Using the Multiple Sample and Convert (ADC12MSC) Bit

To configure the converter to perform successive conversions automatically and as quickly as possible, a multiple sample and convert function is available. When $ADC12MSC = 1$, $CONSEQx > 0$, 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 ADC12ENC bit is toggled in repeat-single-channel or repeated-sequence modes. The function of the ADC12ENC bit is unchanged when using the ADC12MSC bit.

28.2.7.6 Stopping Conversions

Stopping ADC12_A activity depends on the mode of operation. The recommended ways to stop an active conversion or conversion sequence are:

- Resetting ADC12ENC 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 ADC12ENC.
- Resetting ADC12ENC during repeat-single-channel operation stops the converter at the end of the current conversion.
- Resetting ADC12ENC 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 ADC12ENC bit. Conversion data are unreliable.

NOTE: No ADC12EOS bit set for sequence

If no ADC12EOS bit is set and a sequence mode is selected, resetting the ADC12ENC bit does not stop the sequence. To stop the sequence, first select a single-channel mode and then reset ADC12ENC.

28.2.8 Using the Integrated Temperature Sensor

To use the on-chip temperature sensor, the user selects the analog input channel $\text{INCHx} = 1010$. Any other configuration is done as if an external channel were selected, including reference selection, conversion-memory selection, etc. The temperature sensor is part of the reference. Therefore, for devices with the REF module, in addition to the input channels selection $\text{INCHx} = 1010$, configuring $\text{ADC12REFON} = 1$ (for $\text{REFMSTR} = 0$) or $\text{REFON} = 1$ (for $\text{REFMSTR} = 1$) is required to enable the temperature sensor.

For the MSP430F54xx (non-A) devices, which do not include the REF module, selecting the temperature sensor by configuring $\text{INCHx} = 1010$ automatically enables the reference generator required for the temperature sensor. Any other configuration is done as if an external channel were selected, including reference selection, conversion-memory selection, etc.

A typical temperature sensor transfer function is shown in [Figure 28-11](#). The transfer function shown in [Figure 28-11](#) is only an example—the device-specific data sheet contains the actual parameters for a given device. When using the temperature sensor, the sample period must be greater than 30 μs . The temperature sensor offset error can be large and may need to be calibrated for most applications. Temperature calibration values are available for use in the TLV descriptors (see the device-specific data sheet for locations).

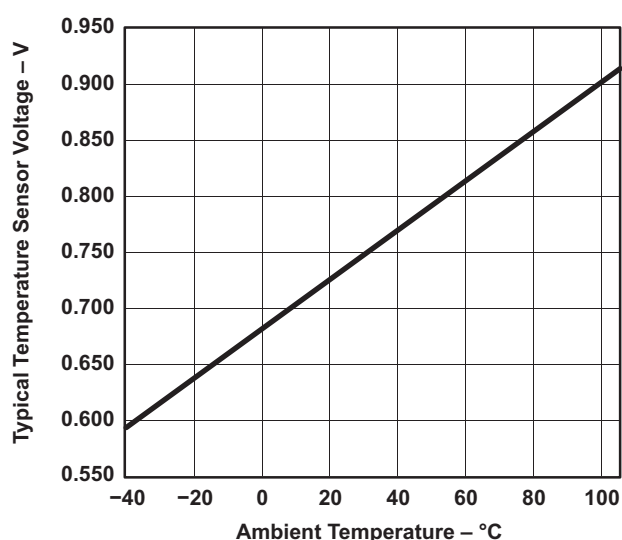


Figure 28-11. Typical Temperature Sensor Transfer Function

28.2.9 ADC12_A Grounding and Noise Considerations

As with any high-resolution ADC, appropriate printed circuit board layout and grounding techniques should 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 28-12](#) 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. A noise-free design using separate analog and digital ground planes with a single-point connection is recommend to achieve high accuracy.

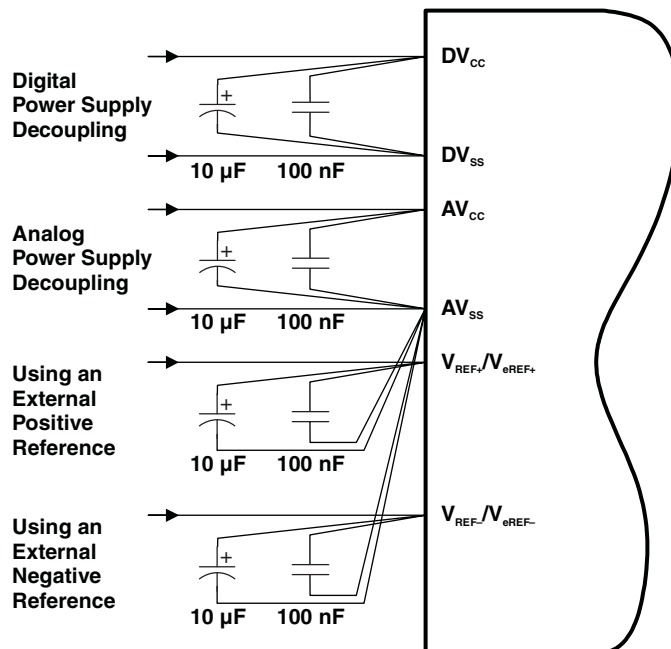


Figure 28-12. ADC12_A Grounding and Noise Considerations

28.2.10 ADC12_A Interrupts

The ADC12_A has 18 interrupt sources:

- ADC12IFG0 to ADC12IFG15
- ADC12OV, ADC12MEMx overflow
- ADC12TOV, ADC12_A conversion time overflow

The ADC12IFGx bits are set when their corresponding ADC12MEMx memory register is loaded with a conversion result. An interrupt request is generated if the corresponding ADC12IEx bit and the GIE bit are set. The ADC12OV condition occurs when a conversion result is written to any ADC12MEMx before its previous conversion result was read. The ADC12TOV condition is generated when another sample-and-conversion is requested before the current conversion is completed. The DMA is triggered after the conversion in single-channel conversion mode or after the completion of a sequence of channel conversions in sequence-of-channels conversion mode.

28.2.10.1 ADC12IV, Interrupt Vector Generator

All ADC12_A interrupt sources are prioritized and combined to source a single interrupt vector. The interrupt vector register ADC12IV is used to determine which enabled ADC12_A interrupt source requested an interrupt.

The highest-priority enabled ADC12_A interrupt generates a number in the ADC12IV register (see register description). This number can be evaluated or added to the program counter (PC) to automatically enter the appropriate software routine. Disabled ADC12_A interrupts do not affect the ADC12IV value.

Any access, read or write, of the ADC12IV register automatically resets the ADC12OV condition or the ADC12TOV condition, if either was the highest-pending interrupt. Neither interrupt condition has an accessible interrupt flag. The ADC12IFGx flags are not reset by an ADC12IV access. ADC12IFGx bits are reset automatically by accessing their associated ADC12MEMx register or may be reset with software.

If another interrupt is pending after servicing of an interrupt, another interrupt is generated. For example, if the ADC12OV and ADC12IFG3 interrupts are pending when the interrupt service routine accesses the ADC12IV register, the ADC12OV interrupt condition is reset automatically. After the RETI instruction of the interrupt service routine is executed, the ADC12IFG3 generates another interrupt.

28.2.10.2 ADC12_A Interrupt Handling Software Example

The following software example shows the recommended use of the ADC12IV and handling overhead. The ADC12IV value is added to the PC to automatically jump to the appropriate routine.

The numbers at the right margin show the necessary CPU cycles for each instruction. The software overhead for different interrupt sources includes interrupt latency and return-from-interrupt cycles, but not the task handling itself. The latencies are:

- ADC12IFG0 to ADC12IFG14, ADC12TOV, and ADC12OV: 16 cycles
- ADC12IFG15: 14 cycles

The interrupt handler for ADC12IFG15 shows a way to check immediately if a higher-prioritized interrupt occurred during the processing of ADC12IFG15. This saves nine cycles if another ADC12_A interrupt is pending.

```
; Interrupt handler for ADC12.
INT_ADC12          ; Enter Interrupt Service Routine
    ADD      &ADC12IV,PC    ; Add offset to PC
    RETI      ; Vector 0: No interrupt
    JMP      ADOV           ; Vector 2: ADC overflow
    JMP      ADTOV          ; Vector 4: ADC timing overflow
    JMP      ADM0           ; Vector 6: ADC12IFG0
    ...          ; Vectors 8-32
    JMP      ADM14          ; Vector 34: ADC12IFG14
;
; Handler for ADC12IFG15 starts here. No JMP required.
;
ADM15      MOV      &ADC12MEM15,xxx    ; Move result, flag is reset
    ...          ; Other instruction needed?
    JMP      INT_ADC12    ; Check other int pending
;
; ADC12IFG14-ADC12IFG1 handlers go here
;
ADM0      MOV      &ADC12MEM0,xxx      ; Move result, flag is reset
    ...          ; Other instruction needed?
    RETI      ; Return
;
ADTOV     ...          ; Handle Conv. time overflow
    RETI      ; Return
;
ADOV      ...          ; Handle ADCMEMx overflow
    RETI      ; Return
```

28.3 ADC12_A Registers

The ADC12_A registers are listed in [Table 28-3](#). The base address of the ADC12_A can be found in the device-specific data sheet. The address offset of each ADC12_A register is given in [Table 28-3](#).

NOTE: All registers have word or byte register access. For a generic register *ANYREG*, the suffix "_L" (*ANYREG_L*) refers to the lower byte of the register (bits 0 through 7). The suffix "_H" (*ANYREG_H*) refers to the upper byte of the register (bits 8 through 15).

Table 28-3. ADC12_A Registers

| Offset | Acronym | Register Name | Type | Access | Reset | Section |
|--------|-------------|--------------------------|------------|--------|-----------|--------------------------------|
| 00h | ADC12CTL0 | ADC12_A Control 0 | Read/write | Word | 0000h | Section 28.3.1 |
| 00h | ADC12CTL0_L | | Read/write | Byte | 00h | |
| 01h | ADC12CTL0_H | | Read/write | Byte | 00h | |
| 02h | ADC12CTL1 | ADC12_A Control 1 | Read/write | Word | 0000h | Section 28.3.2 |
| 02h | ADC12CTL1_L | | Read/write | Byte | 00h | |
| 03h | ADC12CTL1_H | | Read/write | Byte | 00h | |
| 04h | ADC12CTL2 | ADC12_A Control 2 | Read/write | Word | 0020h | Section 28.3.3 |
| 04h | ADC12CTL2_L | | Read/write | Byte | 20h | |
| 05h | ADC12CTL2_H | | Read/write | Byte | 00h | |
| 0Ah | ADC12IFG | ADC12_A Interrupt Flag | Read/write | Word | 0000h | Section 28.3.7 |
| 0Ah | ADC12IFG_L | | Read/write | Byte | 00h | |
| 0Bh | ADC12IFG_H | | Read/write | Byte | 00h | |
| 0Ch | ADC12IE | ADC12_A Interrupt Enable | Read/write | Word | 0000h | Section 28.3.6 |
| 0Ch | ADC12IE_L | | Read/write | Byte | 00h | |
| 0Dh | ADC12IE_H | | Read/write | Byte | 00h | |
| 0Eh | ADC12IV | ADC12_A Interrupt Vector | Read | Word | 0000h | Section 28.3.8 |
| 0Eh | ADC12IV_L | | Read | Byte | 00h | |
| 0Fh | ADC12IV_H | | Read | Byte | 00h | |
| 20h | ADC12MEM0 | ADC12_A Memory 0 | Read/write | Word | undefined | Section 28.3.4 |
| 20h | ADC12MEM0_L | | Read/write | Byte | undefined | |
| 21h | ADC12MEM0_H | | Read/write | Byte | undefined | |
| 22h | ADC12MEM1 | ADC12_A Memory 1 | Read/write | Word | undefined | Section 28.3.4 |
| 22h | ADC12MEM1_L | | Read/write | Byte | undefined | |
| 23h | ADC12MEM1_H | | Read/write | Byte | undefined | |
| 24h | ADC12MEM2 | ADC12_A Memory 2 | Read/write | Word | undefined | Section 28.3.4 |
| 24h | ADC12MEM2_L | | Read/write | Byte | undefined | |
| 25h | ADC12MEM2_H | | Read/write | Byte | undefined | |
| 26h | ADC12MEM3 | ADC12_A Memory 3 | Read/write | Word | undefined | Section 28.3.4 |
| 26h | ADC12MEM3_L | | Read/write | Byte | undefined | |
| 27h | ADC12MEM3_H | | Read/write | Byte | undefined | |
| 28h | ADC12MEM4 | ADC12_A Memory 4 | Read/write | Word | undefined | Section 28.3.4 |
| 28h | ADC12MEM4_L | | Read/write | Byte | undefined | |
| 29h | ADC12MEM4_H | | Read/write | Byte | undefined | |
| 2Ah | ADC12MEM5 | ADC12_A Memory 5 | Read/write | Word | undefined | Section 28.3.4 |
| 2Ah | ADC12MEM5_L | | Read/write | Byte | undefined | |
| 2Bh | ADC12MEM5_H | | Read/write | Byte | undefined | |
| 2Ch | ADC12MEM6 | ADC12_A Memory 6 | Read/write | Word | undefined | Section 28.3.4 |
| 2Ch | ADC12MEM6_L | | Read/write | Byte | undefined | |
| 2Dh | ADC12MEM6_H | | Read/write | Byte | undefined | |

Table 28-3. ADC12_A Registers (continued)

| Offset | Acronym | Register Name | Type | Access | Reset | Section |
|--------|--------------|---------------------------|------------|--------|-----------|--------------------------------|
| 2Eh | ADC12MEM7 | ADC12_A Memory 7 | Read/write | Word | undefined | Section 28.3.4 |
| 2Eh | ADC12MEM7_L | | Read/write | Byte | undefined | |
| 2Fh | ADC12MEM7_H | | Read/write | Byte | undefined | |
| 30h | ADC12MEM8 | ADC12_A Memory 8 | Read/write | Word | undefined | Section 28.3.4 |
| 30h | ADC12MEM8_L | | Read/write | Byte | undefined | |
| 31h | ADC12MEM8_H | | Read/write | Byte | undefined | |
| 32h | ADC12MEM9 | ADC12_A Memory 9 | Read/write | Word | undefined | Section 28.3.4 |
| 32h | ADC12MEM9_L | | Read/write | Byte | undefined | |
| 33h | ADC12MEM9_H | | Read/write | Byte | undefined | |
| 34h | ADC12MEM10 | ADC12_A Memory 10 | Read/write | Word | undefined | Section 28.3.4 |
| 34h | ADC12MEM10_L | | Read/write | Byte | undefined | |
| 35h | ADC12MEM10_H | | Read/write | Byte | undefined | |
| 36h | ADC12MEM11 | ADC12_A Memory 11 | Read/write | Word | undefined | Section 28.3.4 |
| 36h | ADC12MEM11_L | | Read/write | Byte | undefined | |
| 37h | ADC12MEM11_H | | Read/write | Byte | undefined | |
| 38h | ADC12MEM12 | ADC12_A Memory 12 | Read/write | Word | undefined | Section 28.3.4 |
| 38h | ADC12MEM12_L | | Read/write | Byte | undefined | |
| 39h | ADC12MEM12_H | | Read/write | Byte | undefined | |
| 3Ah | ADC12MEM13 | ADC12_A Memory 13 | Read/write | Word | undefined | Section 28.3.4 |
| 3Ah | ADC12MEM13_L | | Read/write | Byte | undefined | |
| 3Bh | ADC12MEM13_H | | Read/write | Byte | undefined | |
| 3Ch | ADC12MEM14 | ADC12_A Memory 14 | Read/write | Word | undefined | Section 28.3.4 |
| 3Ch | ADC12MEM14_L | | Read/write | Byte | undefined | |
| 3Dh | ADC12MEM14_H | | Read/write | Byte | undefined | |
| 3Dh | ADC12MEM15 | ADC12_A Memory 15 | Read/write | Word | undefined | Section 28.3.4 |
| 3Dh | ADC12MEM15_L | | Read/write | Byte | undefined | |
| 3Eh | ADC12MEM15_H | | Read/write | Byte | undefined | |
| 10h | ADC12MCTL0 | ADC12_A Memory Control 0 | Read/write | Byte | undefined | Section 28.3.5 |
| 11h | ADC12MCTL1 | ADC12_A Memory Control 1 | Read/write | Byte | undefined | Section 28.3.5 |
| 12h | ADC12MCTL2 | ADC12_A Memory Control 2 | Read/write | Byte | undefined | Section 28.3.5 |
| 13h | ADC12MCTL3 | ADC12_A Memory Control 3 | Read/write | Byte | undefined | Section 28.3.5 |
| 14h | ADC12MCTL4 | ADC12_A Memory Control 4 | Read/write | Byte | undefined | Section 28.3.5 |
| 15h | ADC12MCTL5 | ADC12_A Memory Control 5 | Read/write | Byte | undefined | Section 28.3.5 |
| 16h | ADC12MCTL6 | ADC12_A Memory Control 6 | Read/write | Byte | undefined | Section 28.3.5 |
| 17h | ADC12MCTL7 | ADC12_A Memory Control 7 | Read/write | Byte | undefined | Section 28.3.5 |
| 18h | ADC12MCTL8 | ADC12_A Memory Control 8 | Read/write | Byte | undefined | Section 28.3.5 |
| 19h | ADC12MCTL9 | ADC12_A Memory Control 9 | Read/write | Byte | undefined | Section 28.3.5 |
| 1Ah | ADC12MCTL10 | ADC12_A Memory Control 10 | Read/write | Byte | undefined | Section 28.3.5 |
| 1Bh | ADC12MCTL11 | ADC12_A Memory Control 11 | Read/write | Byte | undefined | Section 28.3.5 |
| 1Ch | ADC12MCTL12 | ADC12_A Memory Control 12 | Read/write | Byte | undefined | Section 28.3.5 |
| 1Dh | ADC12MCTL13 | ADC12_A Memory Control 13 | Read/write | Byte | undefined | Section 28.3.5 |
| 1Eh | ADC12MCTL14 | ADC12_A Memory Control 14 | Read/write | Byte | undefined | Section 28.3.5 |
| 1Fh | ADC12MCTL15 | ADC12_A Memory Control 15 | Read/write | Byte | undefined | Section 28.3.5 |

28.3.1 ADC12CTL0 Register

ADC12_A Control Register 0

Figure 28-13. ADC12CTL0 Register

| | | | | | | | |
|------------|------------------|------------|---------|------------|------------|----------|---------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ADC12SHT1x | | | | ADC12SHT0x | | | |
| rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADC12MSC | ADC12REF2_5 V | ADC12REFON | ADC12ON | ADC12OVIE | ADC12TOVIE | ADC12ENC | ADC12SC |
| rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) |

Can be modified only when ADC12ENC = 0

Table 28-4. ADC12CTL0 Register Description

| Bit | Field | Type | Reset | Description |
|-------|--------------|------|-------|--|
| 15-12 | ADC12SHT1x | RW | 0h | ADC12_A sample-and-hold time. These bits define the number of ADC12CLK cycles in the sampling period for registers ADC12MEM8 to ADC12MEM15. |
| 11-8 | ADC12SHT0x | RW | 0h | ADC12_A sample-and-hold time. These bits define the number of ADC12CLK cycles in the sampling period for registers ADC12MEM0 to ADC12MEM7. 0000b = 4 ADC12CLK cycles 0001b = 8 ADC12CLK cycles 0010b = 16 ADC12CLK cycles 0011b = 32 ADC12CLK cycles 0100b = 64 ADC12CLK cycles 0101b = 96 ADC12CLK cycles 0110b = 128 ADC12CLK cycles 0111b = 192 ADC12CLK cycles 1000b = 256 ADC12CLK cycles 1001b = 384 ADC12CLK cycles 1010b = 512 ADC12CLK cycles 1011b = 768 ADC12CLK cycles 1100b = 1024 ADC12CLK cycles 1101b = 1024 ADC12CLK cycles 1110b = 1024 ADC12CLK cycles 1111b = 1024 ADC12CLK cycles |
| 7 | ADC12MSC | RW | 0h | ADC12_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 | ADC12REF2_5V | RW | 0h | ADC12_A reference generator voltage. ADC12REFON must also be set. In devices with the REF module, this bit is only valid if the REFMSTR bit of the REF module is set to 0. In the F54xx devices (non-A), the REF module is not available. 0b = 1.5 V 1b = 2.5 V |
| 5 | ADC12REFON | RW | 0h | ADC12_A reference generator on. In devices with the REF module, this bit is only valid if the REFMSTR bit of the REF module is set to 0. In the F54xx devices (non-A), the REF module is not available. 0b = Reference off 1b = Reference on |

Table 28-4. ADC12CTL0 Register Description (continued)

| Bit | Field | Type | Reset | Description |
|-----|------------|------|-------|--|
| 4 | ADC12ON | RW | 0h | ADC12_A on 0b = ADC12_A off 1b = ADC12_A on |
| 3 | ADC12OVIE | RW | 0h | ADC12MEMx overflow-interrupt enable. The GIE bit must also be set to enable the interrupt. 0b = Overflow interrupt disabled 1b = Overflow interrupt enabled |
| 2 | ADC12TOVIE | RW | 0h | ADC12_A conversion-time-overflow interrupt enable. The GIE bit must also be set to enable the interrupt. 0b = Conversion time overflow interrupt disabled 1b = Conversion time overflow interrupt enabled |
| 1 | ADC12ENC | RW | 0h | ADC12_A enable conversion 0b = ADC12_A disabled 1b = ADC12_A enabled |
| 0 | ADC12SC | RW | 0h | ADC12_A start conversion. Software-controlled sample-and-conversion start. ADC12SC and ADC12ENC may be set together with one instruction. ADC12SC is reset automatically. 0b = No sample-and-conversion-start 1b = Start sample-and-conversion |

28.3.2 ADC12CTL1 Register

ADC12_A Control Register 1

Figure 28-14. ADC12CTL1 Register

| | | | | | | | |
|-----------------|--------|--------|------------|-----------|--------------|----------|-----------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ADC12CSTARTADDx | | | | ADC12SHSx | | ADC12SHP | ADC12ISSH |
| rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADC12DIVx | | | ADC12SSELx | | ADC12CONSEQx | | ADC12BUSY |
| rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | r-(0) |

Can be modified only when ADC12ENC = 0

Table 28-5. ADC12CTL1 Register Description

| Bit | Field | Type | Reset | Description |
|-------|-----------------|------|-------|---|
| 15-12 | ADC12CSTARTADDx | RW | 0h | ADC12_A conversion start address. These bits select which ADC12_A conversion-memory register is used for a single conversion or for the first conversion in a sequence. The value of CSTARTADDx is 0 to 0Fh, corresponding to ADC12MEM0 to ADC12MEM15. |
| 11-10 | ADC12SHSx | RW | 0h | ADC12_A sample-and-hold source select 00b = ADC12SC bit 01b = Timer source (see device-specific data sheet for exact timer and locations) 10b = Timer source (see device-specific data sheet for exact timer and locations) 11b = Timer source (see device-specific data sheet for exact timer and locations) |
| 9 | ADC12SHP | RW | 0h | ADC12_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. 0b = SAMPCON signal is sourced from the sample-input signal. 1b = SAMPCON signal is sourced from the sampling timer. |
| 8 | ADC12ISSH | RW | 0h | ADC12_A invert signal sample-and-hold 0b = The sample-input signal is not inverted. 1b = The sample-input signal is inverted. |
| 7-5 | ADC12DIVx | RW | 0h | ADC12_A clock divider 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 |
| 4-3 | ADC12SSELx | RW | 0h | ADC12_A clock source select 00b = ADC12OSC (MODOSC) 01b = ACLK 10b = MCLK 11b = SMCLK |
| 2-1 | ADC12CONSEQx | RW | 0h | ADC12_A conversion sequence mode select 00b = Single-channel, single-conversion 01b = Sequence-of-channels 10b = Repeat-single-channel 11b = Repeat-sequence-of-channels |
| 0 | ADC12BUSY | R | 0h | ADC12_A busy. This bit indicates an active sample or conversion operation. 0b = No operation is active. 1b = A sequence, sample, or conversion is active. |

28.3.3 ADC12CTL2 Register

ADC12_A Control Register 2

Figure 28-15. ADC12CTL2 Register

| | | | | | | | |
|------------|----------|----------|-----|---------|---------|-------------|---------------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| Reserved | | | | | | | ADC12PDIV |
| r-0 | r-0 | r-0 | r-0 | r-0 | r-0 | r-0 | rw-0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADC12TCOFF | Reserved | ADC12RES | | ADC12DF | ADC12SR | ADC12REFOUT | ADC12REFBURST |
| rw-(0) | r-0 | rw-(1) | | rw-(0) | rw-(0) | rw-(0) | rw-(0) |

Can be modified only when ADC12ENC = 0

Table 28-6. ADC12CTL2 Register Description

| Bit | Field | Type | Reset | Description |
|------|---------------|------|-------|---|
| 15-9 | Reserved | R | 0h | Reserved. Always reads as 0. |
| 8 | ADC12PDIV | RW | 0h | ADC12_A predivider. This bit predivides the selected ADC12_A clock source. 0b = Predivide by 1 1b = Predivide by 4 |
| 7 | ADC12TCOFF | RW | 0h | ADC12_A temperature sensor off. If the bit is set, the temperature sensor turned off. This is used to save power. In devices with the REF module, this bit is only valid if the REFMSTR bit of the REF module is set to 0. In the F54xx devices (non-A), the REF module is not available. 0b = Temperature sensor on 1b = Temperature sensor off |
| 6 | Reserved | R | 0h | Reserved. Always reads as 0. |
| 5-4 | ADC12RES | RW | 2h | ADC12_A resolution. This bit defines the conversion result resolution. 00b = 8 bit (9 clock cycle conversion time) 01b = 10 bit (11 clock cycle conversion time) 10b = 12 bit (13 clock cycle conversion time) 11b = Reserved |
| 3 | ADC12DF | RW | 0h | ADC12_A data read-back format. Data is always stored in the binary unsigned format. 0b = Binary unsigned. Theoretically, the analog input voltage -VREF results in 0000h, the analog input voltage +VREF results in 0FFFh. 1b = Signed binary (2s complement), left aligned. Theoretically, the analog input voltage -VREF results in 8000h, the analog input voltage +VREF results in 7FFFh. |
| 2 | ADC12SR | RW | 0h | ADC12_A sampling rate. This bit selects the reference buffer drive capability for the maximum sampling rate. Setting ADC12SR reduces the current consumption of the reference buffer. 0b = Reference buffer supports up to approximately 200 ksp/s. 1b = Reference buffer supports up to approximately 50 ksp/s. |
| 1 | ADC12REFOUT | RW | 0h | Reference output. In devices with the REF module, this bit is only valid if the REFMSTR bit of the REF module is set to 0. In the F54xx devices (non-A), the REF module is not available. 0b = Reference output off 1b = Reference output on |
| 0 | ADC12REFBURST | RW | 0h | Reference burst 0b = Reference buffer on continuously 1b = Reference buffer on only during sample-and-conversion |

28.3.4 ADC12MEMx Register

ADC12_A Conversion Memory Register

Figure 28-16. ADC12MEMx 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 | | | | | | | |
| rw | rw | rw | rw | rw | rw | rw | rw |

Table 28-7. ADC12MEMx Register Description

| Bit | Field | Type | Reset | Description |
|------|--------------------|------|-----------|---|
| 15-0 | Conversion Results | RW | undefined | <p>Binary unsigned format: This data format is used if ADC12DF = 0. The 12-bit conversion results are right justified. Bit 11 is the MSB. Bits 15–12 are 0 in 12-bit mode, bits 15–10 are 0 in 10-bit mode, and bits 15–8 are 0 in 8-bit mode. Writing to the conversion memory registers corrupts the results.</p> <p>2s-complement format: This data format is used if ADC12DF = 1. The 12-bit conversion results are left justified, 2s-complement format. Bit 15 is the MSB. Bits 3–0 are 0 in 12-bit mode, bits 5–0 are 0 in 10-bit mode, and bits 7–0 are 0 in 8-bit mode. The data is stored in the right-justified format and is converted to the left-justified 2s-complement format during read back.</p> |

28.3.5 ADC12MCTLx Register

ADC12_A Conversion Memory Control Register

Figure 28-17. ADC12MCTLx Register

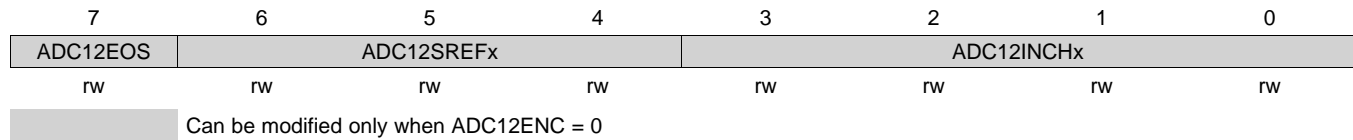


Table 28-8. ADC12MCTLx Register Description

| Bit | Field | Type | Reset | Description |
|-----|------------|------|-------|--|
| 7 | ADC12EOS | RW | 0h | End of sequence. Indicates the last conversion in a sequence. 0b = Not end of sequence 1b = End of sequence |
| 6-4 | ADC12SREFx | RW | 0h | Select reference 000b = V(R+) = AVCC and V(R-) = AVSS 001b = V(R+) = VREF+ and V(R-) = AVSS 010b = V(R+) = VeREF+ and V(R-) = AVSS 011b = V(R+) = VeREF+ and V(R-) = AVSS 100b = V(R+) = AVCC and V(R-) = VREF-/VeREF- 101b = V(R+) = VREF+ and V(R-) = VREF-/VeREF- 110b = V(R+) = VeREF+ and V(R-) = VREF-/VeREF- 111b = V(R+) = VeREF+ and V(R-) = VREF-/VeREF- |
| 3-0 | ADC12INCHx | RW | 0h | Input channel select 0000b = A0 0001b = A1 0010b = A2 0011b = A3 0100b = A4 0101b = A5 0110b = A6 0111b = A7 1000b = VeREF+ 1001b = VREF-/VeREF- 1010b = Temperature diode 1011b = (AVCC – AVSS) / 2 1100b = A12. On devices with the Battery Backup System, VBAT can be measured internally by the ADC. 1101b = A13 1110b = A14 1111b = A15 |

28.3.6 ADC12IE Register

ADC12_A Interrupt Enable Register

Figure 28-18. ADC12IE Register

| | | | | | | | |
|-----------|-----------|-----------|-----------|-----------|-----------|----------|----------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ADC12IE15 | ADC12IE14 | ADC12IE13 | ADC12IE12 | ADC12IE11 | ADC12IE10 | ADC12IE9 | ADC12IE8 |
| rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADC12IE7 | ADC12IE6 | ADC12IE5 | ADC12IE4 | ADC12IE3 | ADC12IE2 | ADC12IE1 | ADC12IE0 |
| rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) |

Table 28-9. ADC12IE Register Description

| Bit | Field | Type | Reset | Description |
|-----|-----------|------|-------|---|
| 15 | ADC12IE15 | RW | 0h | Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG15 bit. 0b = Interrupt disabled 1b = Interrupt enabled |
| 14 | ADC12IE14 | RW | 0h | Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG14 bit. 0b = Interrupt disabled 1b = Interrupt enabled |
| 13 | ADC12IE13 | RW | 0h | Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG13 bit. 0b = Interrupt disabled 1b = Interrupt enabled |
| 12 | ADC12IE12 | RW | 0h | Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG12 bit. 0b = Interrupt disabled 1b = Interrupt enabled |
| 11 | ADC12IE11 | RW | 0h | Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG11 bit. 0b = Interrupt disabled 1b = Interrupt enabled |
| 10 | ADC12IE10 | RW | 0h | Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG10 bit. 0b = Interrupt disabled 1b = Interrupt enabled |
| 9 | ADC12IE9 | RW | 0h | Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG9 bit. 0b = Interrupt disabled 1b = Interrupt enabled |
| 8 | ADC12IE8 | RW | 0h | Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG8 bit. 0b = Interrupt disabled 1b = Interrupt enabled |
| 7 | ADC12IE7 | RW | 0h | Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG7 bit. 0b = Interrupt disabled 1b = Interrupt enabled |
| 6 | ADC12IE6 | RW | 0h | Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG6 bit. 0b = Interrupt disabled 1b = Interrupt enabled |

Table 28-9. ADC12IE Register Description (continued)

| Bit | Field | Type | Reset | Description |
|-----|----------|------|-------|--|
| 5 | ADC12IE5 | RW | 0h | Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG5 bit. 0b = Interrupt disabled 1b = Interrupt enabled |
| 4 | ADC12IE4 | RW | 0h | Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG4 bit. 0b = Interrupt disabled 1b = Interrupt enabled |
| 3 | ADC12IE3 | RW | 0h | Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG3 bit. 0b = Interrupt disabled 1b = Interrupt enabled |
| 2 | ADC12IE2 | RW | 0h | Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG2 bit. 0b = Interrupt disabled 1b = Interrupt enabled |
| 1 | ADC12IE1 | RW | 0h | Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG1 bit. 0b = Interrupt disabled 1b = Interrupt enabled |
| 0 | ADC12IE0 | RW | 0h | Interrupt enable. This bit enables or disables the interrupt request for the ADC12IFG0 bit. 0b = Interrupt disabled 1b = Interrupt enabled |

28.3.7 ADC12IFG Register

ADC12_A Interrupt Flag Register

Figure 28-19. ADC12IFG Register

| | | | | | | | |
|------------|------------|------------|------------|------------|------------|-----------|-----------|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ADC12IFG15 | ADC12IFG14 | ADC12IFG13 | ADC12IFG12 | ADC12IFG11 | ADC12IFG10 | ADC12IFG9 | ADC12IFG8 |
| rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADC12IFG7 | ADC12IFG6 | ADC12IFG5 | ADC12IFG4 | ADC12IFG3 | ADC12IFG2 | ADC12IFG1 | ADC12IFG0 |
| rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) | rw-(0) |

Table 28-10. ADC12IFG Register Description

| Bit | Field | Type | Reset | Description |
|-----|------------|------|-------|--|
| 15 | ADC12IFG15 | RW | 0h | ADC12MEM15 interrupt flag. This bit is set when ADC12MEM15 is loaded with a conversion result. This bit is reset if the ADC12MEM15 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending |
| 14 | ADC12IFG14 | RW | 0h | ADC12MEM14 interrupt flag. This bit is set when ADC12MEM14 is loaded with a conversion result. This bit is reset if the ADC12MEM14 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending |
| 13 | ADC12IFG13 | RW | 0h | ADC12MEM13 interrupt flag. This bit is set when ADC12MEM13 is loaded with a conversion result. This bit is reset if the ADC12MEM13 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending |
| 12 | ADC12IFG12 | RW | 0h | ADC12MEM12 interrupt flag. This bit is set when ADC12MEM12 is loaded with a conversion result. This bit is reset if the ADC12MEM12 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending |
| 11 | ADC12IFG11 | RW | 0h | ADC12MEM11 interrupt flag. This bit is set when ADC12MEM11 is loaded with a conversion result. This bit is reset if the ADC12MEM11 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending |
| 10 | ADC12IFG10 | RW | 0h | ADC12MEM10 interrupt flag. This bit is set when ADC12MEM10 is loaded with a conversion result. This bit is reset if the ADC12MEM10 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending |
| 9 | ADC12IFG9 | RW | 0h | ADC12MEM9 interrupt flag. This bit is set when ADC12MEM9 is loaded with a conversion result. This bit is reset if the ADC12MEM9 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending |
| 8 | ADC12IFG8 | RW | 0h | ADC12MEM8 interrupt flag. This bit is set when ADC12MEM8 is loaded with a conversion result. This bit is reset if the ADC12MEM8 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending |

Table 28-10. ADC12IFG Register Description (continued)

| Bit | Field | Type | Reset | Description |
|-----|-----------|------|-------|---|
| 7 | ADC12IFG7 | RW | 0h | ADC12MEM7 interrupt flag. This bit is set when ADC12MEM7 is loaded with a conversion result. This bit is reset if the ADC12MEM7 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending |
| 6 | ADC12IFG6 | RW | 0h | ADC12MEM6 interrupt flag. This bit is set when ADC12MEM6 is loaded with a conversion result. This bit is reset if the ADC12MEM6 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending |
| 5 | ADC12IFG5 | RW | 0h | ADC12MEM5 interrupt flag. This bit is set when ADC12MEM5 is loaded with a conversion result. This bit is reset if the ADC12MEM5 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending |
| 4 | ADC12IFG4 | RW | 0h | ADC12MEM4 interrupt flag. This bit is set when ADC12MEM4 is loaded with a conversion result. This bit is reset if the ADC12MEM4 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending |
| 3 | ADC12IFG3 | RW | 0h | ADC12MEM3 interrupt flag. This bit is set when ADC12MEM3 is loaded with a conversion result. This bit is reset if the ADC12MEM3 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending |
| 2 | ADC12IFG2 | RW | 0h | ADC12MEM2 interrupt flag. This bit is set when ADC12MEM2 is loaded with a conversion result. This bit is reset if the ADC12MEM2 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending |
| 1 | ADC12IFG1 | RW | 0h | ADC12MEM1 interrupt flag. This bit is set when ADC12MEM1 is loaded with a conversion result. This bit is reset if the ADC12MEM1 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending |
| 0 | ADC12IFG0 | RW | 0h | ADC12MEM0 interrupt flag. This bit is set when ADC12MEM0 is loaded with a conversion result. This bit is reset if the ADC12MEM0 is accessed, or it may be reset with software. 0b = No interrupt pending 1b = Interrupt pending |

28.3.8 ADC12IV Register

ADC12_A Interrupt Vector Register

Figure 28-20. ADC12IV Register

| | | | | | | | |
|----------|----|-------|-------|-------|-------|-------|----|
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
| ADC12IVx | | | | | | | |
| r0 | r0 | r0 | r0 | r0 | r0 | r0 | r0 |
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| ADC12IVx | | | | | | | |
| r0 | r0 | r-(0) | r-(0) | r-(0) | r-(0) | r-(0) | r0 |

Table 28-11. ADC12IV Register Description

| Bit | Field | Type | Reset | Description |
|------|----------|------|-------|--|
| 15-0 | ADC12IVx | R | 0h | <p>ADC12_A interrupt vector value</p> <p>00h = No interrupt pending</p> <p>02h = Interrupt Source: ADC12MEMx overflow; Interrupt Flag: –; Interrupt Priority: Highest</p> <p>04h = Interrupt Source: Conversion time overflow; Interrupt Flag: –</p> <p>06h = Interrupt Source: ADC12MEM0 interrupt flag; Interrupt Flag: ADC12IFG0</p> <p>08h = Interrupt Source: ADC12MEM1 interrupt flag; Interrupt Flag: ADC12IFG1</p> <p>0Ah = Interrupt Source: ADC12MEM2 interrupt flag; Interrupt Flag: ADC12IFG2</p> <p>0Ch = Interrupt Source: ADC12MEM3 interrupt flag; Interrupt Flag: ADC12IFG3</p> <p>0Eh = Interrupt Source: ADC12MEM4 interrupt flag; Interrupt Flag: ADC12IFG4</p> <p>10h = Interrupt Source: ADC12MEM5 interrupt flag; Interrupt Flag: ADC12IFG5</p> <p>12h = Interrupt Source: ADC12MEM6 interrupt flag; Interrupt Flag: ADC12IFG6</p> <p>14h = Interrupt Source: ADC12MEM7 interrupt flag; Interrupt Flag: ADC12IFG7</p> <p>16h = Interrupt Source: ADC12MEM8 interrupt flag; Interrupt Flag: ADC12IFG8</p> <p>18h = Interrupt Source: ADC12MEM9 interrupt flag; Interrupt Flag: ADC12IFG9</p> <p>1Ah = Interrupt Source: ADC12MEM10 interrupt flag; Interrupt Flag: ADC12IFG10</p> <p>1Ch = Interrupt Source: ADC12MEM11 interrupt flag; Interrupt Flag: ADC12IFG11</p> <p>1Eh = Interrupt Source: ADC12MEM12 interrupt flag; Interrupt Flag: ADC12IFG12</p> <p>20h = Interrupt Source: ADC12MEM13 interrupt flag; Interrupt Flag: ADC12IFG13</p> <p>22h = Interrupt Source: ADC12MEM14 interrupt flag; Interrupt Flag: ADC12IFG14</p> <p>24h = Interrupt Source: ADC12MEM15 interrupt flag; Interrupt Flag: ADC12IFG15; Interrupt Priority: Lowest</p> |