<table>
<thead>
<tr>
<th>Version No.</th>
<th>Revision Date</th>
<th>Revised By</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.50</td>
<td>3/31/2013</td>
<td>JC</td>
<td>Starting doc</td>
</tr>
<tr>
<td>0.80</td>
<td>4/7/2013</td>
<td>JC</td>
<td>Improved</td>
</tr>
</tbody>
</table>
The ADC10 module supports fast, 10-bit analog-to-digital conversions. The module implements a 10-bit SAR (Successive approximation register) core, sample select control, reference generator, and data transfer controller (DTC).

ADC10 features include:

- Greater than 200 ksps maximum conversion rate
- Monotonic 10-bit converter with no missing codes
- Sample-and-hold with programmable sample periods
- Conversion initiation by software or Timer_A
- Software selectable on-chip reference voltage generation (1.5 V or 2.5 V)
- Software selectable internal or external reference
- Eight external input channels (twelve on MSP430x22xx devices)
- Conversion channels for internal temperature sensor, V CC, and external references
- Selectable conversion clock source
- Single-channel, repeated single-channel, sequence, and repeated sequence conversion modes
- ADC core and reference voltage can be powered down separately
- Data transfer controller for automatic storage of conversion results
Figure 22-1. ADC10 Block Diagram
ADC10 Block Diagram – Grouped by Function

Reference Setup

ADC Clock

S/H Setup

Input Select

Output Handling

Figure 22-1. ADC10 Block Diagram
ADC10 Setup

Overview

The ADC core converts an analog input to its 10-bit digital representation. As shown in the figure below, the core uses two programmable/selectable voltage levels (Vref+ and Vref-) to define the upper and lower limits of the conversion. The digital output (N ADC) is full scale (10 bits, 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. Result of conversion goes into ADC10MEM. The transfer equation is as shown below:

\[ N_{ADC} = 1023 \times \frac{V_{IN} - V_{R-}}{V_{R+} - V_{R-}} \]

The ADC10 core is configured by two control registers, ADC10CTL0 and ADC10CTL1. The core is enabled with the ADC10ON bit. With few exceptions the ADC10 control bits can only be modified when ENC = 0. ENC must be set to 1 before any conversion can take place.

```c
#define ENC (0x002)  /* ADC10 Enable Conversion */
FR_16BIT(ADC10MEM);  /* ADC10 Memory */
```

To configure ADC, there are five different part you need to consider: ADC clock, Input channel, References, Sample and Conversion Interval and (option) output data transfer control.

---

**Diagram:**

- ADC Result N<sub>ADC</sub> (Digital)
- V<sub>IN</sub>
- V<sub>R+</sub>
- V<sub>R-</sub>
- Digital: 1023, 1022, 
- Analog: 0, 1

---
Clock – ADC10CLK

Related registers and bits: ADC10CTL1 (ADC10SSELx, ADC10DIVx)

ADC10SSELx (ADC10CTL1) selects source of ADC10CLK. Possible sources are SMCLK, MCLK, ACLK and internal oscillator ADC10OSC. ADC10OSC is around 5MHz. ADC10DIVx (ADC10CTL1) can be programmed to further divide the clock to up to 8th of the source frequency.

ADC10SSELx   Bits 4-3   ADC10 clock source select
00   ADC10OSC
01   ACLK
10   MCLK
11   SMCLK

ADC10DIVx   Bits 7-5   ADC10 clock divider
000   /1
001   /2
010   /3
011   /4
100   /5
101   /6
110   /7
111   /8

#define ADC10SSEL0   (0x0008)   /* ADC10 Clock Source Select Bit: 0 */
#define ADC10SSEL1   (0x0010)   /* ADC10 Clock Source Select Bit: 1 */
#define ADC10DIV0   (0x0020)   /* ADC10 Clock Divider Select Bit: 0 */
#define ADC10DIV1   (0x0040)   /* ADC10 Clock Divider Select Bit: 1 */
#define ADC10DIV2   (0x0080)   /* ADC10 Clock Divider Select Bit: 2 */
#define ADC10SSEL_0   (0*8u)   /* ADC100SC */
#define ADC10SSEL_1   (1*8u)   /* ACLK */
#define ADC10SSEL_2   (2*8u)   /* MCLK */
#define ADC10SSEL_3   (3*8u)   /* SMCLK */
#define ADC10DIV_0   (0*0x20u)   /* ADC10 Clock Divider Select 0 */
#define ADC10DIV_1   (1*0x20u)   /* ADC10 Clock Divider Select 1 */
#define ADC10DIV_2   (2*0x20u)   /* ADC10 Clock Divider Select 2 */
#define ADC10DIV_3   (3*0x20u)   /* ADC10 Clock Divider Select 3 */
#define ADC10DIV_4   (4*0x20u)   /* ADC10 Clock Divider Select 4 */
#define ADC10DIV_5   (5*0x20u)   /* ADC10 Clock Divider Select 5 */
#define ADC10DIV_6   (6*0x20u)   /* ADC10 Clock Divider Select 6 */
#define ADC10DIV_7   (7*0x20u)   /* ADC10 Clock Divider Select 7 */

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 will not complete, and any result will be invalid.
Input Channels

Related registers and bits: ADC10CTL1 (INCHx)

There are 8 external can be used for conversion by an analog input multiplexer. There are also four internal analog signal channels that are only available for MSP X22X.

INCHx  Bits 15-12
Input channel select. These bits select the channel for a single-conversion or the highest channel for a sequence of conversions.

0000  A0
0001  A1
0010  A2
0011  A3
0100  A4
0101  A5
0110  A6
0111  A7

1000  V eREF+
1001  V REF- /V eREF-
1010  Temperature sensor
1011  (V CC - V SS ) / 2
1100  (V CC - V SS ) / 2, A12 on MSP430x22xx devices
1101  (V CC - V SS ) / 2, A13 on MSP430x22xx devices
1110  (V CC - V SS ) / 2, A14 on MSP430x22xx devices
1111  (V CC - V SS ) / 2, A15 on MSP430x22xx devices

```
#define INCH0  (0x1000)   /* ADC10 Input Channel Select Bit: 0 */
#define INCH1  (0x2000)   /* ADC10 Input Channel Select Bit: 1 */
#define INCH2  (0x4000)   /* ADC10 Input Channel Select Bit: 2 */
#define INCH3  (0x8000)   /* ADC10 Input Channel Select Bit: 3 */

#define INCH_0 (0*0x1000u) /* Selects Channel 0 */
#define INCH_1 (1*0x1000u) /* Selects Channel 1 */
#define INCH_2 (2*0x1000u) /* Selects Channel 2 */
#define INCH_3 (3*0x1000u) /* Selects Channel 3 */
#define INCH_4 (4*0x1000u) /* Selects Channel 4 */
#define INCH_5 (5*0x1000u) /* Selects Channel 5 */
#define INCH_6 (6*0x1000u) /* Selects Channel 6 */
#define INCH_7 (7*0x1000u) /* Selects Channel 7 */
#define INCH_8 (8*0x1000u) /* Selects Channel 8 */
#define INCH_9 (9*0x1000u) /* Selects Channel 9 */
#define INCH_10 (10*0x1000u) /* Selects Channel 10 */
#define INCH_11 (11*0x1000u) /* Selects Channel 11 */
#define INCH_12 (12*0x1000u) /* Selects Channel 12 */
#define INCH_13 (13*0x1000u) /* Selects Channel 13 */
#define INCH_14 (14*0x1000u) /* Selects Channel 14 */
#define INCH_15 (15*0x1000u) /* Selects Channel 15 */
```
When analog signals are applied to digital CMOS gates, parasitic current can flow from VCC to GND. This parasitic current occurs if the input voltage is near the transition level of the gate. Disabling the port pin buffer eliminates the parasitic current flow and therefore reduces overall current consumption. The ADC10AEx bits provide the ability to disable the port pin input and output buffers.

Voltage Reference

Related Registers and bits: ADC10CTL0 (REFON, REOUT, REF2_5V, SREFx, REFBURST)

The ADC10 module contains a built-in voltage reference with two selectable voltage levels. REFON controls the internal reference. When REF2_5V = 1, the internal reference is 2.5 V. When REF2_5V = 0, the reference is 1.5 V. The internal reference voltage may be used internally and, when REOUT = 0, externally on pin V REF+

External references may be supplied for V R+ and V R- through pins A4 and A3 respectively. When external references are used, or when V CC is used as the reference, the internal reference may be turned off to save power. An external positive reference V eREF+ can be buffered by setting SREF0 = 1 and SREF1 = 1. This allows using an external reference with a large internal resistance at the cost of the buffer current. When REFBURST = 1 the increased current consumption is limited to the sample and conversion period. External storage capacitance is not required for the ADC10 reference source as on the ADC12.

**REFOUT** Bit 9 Reference output
0 Reference output off
1 Reference output on

**REFON** Bit 5 Reference generator on
0 Internal Reference off
1 Internal Reference on

**REF2_5V** Bit 6 Reference-generator voltage. REFON must also be set.
0 1.5 V
1 2.5 V

**SREFx** Bits 15-13 Select reference
000 V R+ = V CC and V R- = V SS
001 V R+ = V REF+ and V R- = V SS
010 V R+ = Ve REF+ and V R- = V SS
011 V R+ = Buffered Ve REF+ and V R- = V SS
100 V R+ = V CC and V R- = V REF- / V eREF-
101 V R+ = V REF+ and V R- = V REF- / V eREF-
110 V R+ = Ve REF+ and V R- = V REF- / V eREF-
111 V R+ = Buffered Ve REF+ and V R- = V REF- / V eREF-
REFBURST Bit 8 Reference burst.
0 Reference buffer on continuously
1 Reference buffer on only during sample-and-conversion

#define REFON (0x020)    /* ADC10 Reference on */
#define REF2_5V (0x040)  /* ADC10 Ref 0:1.5V / 1:2.5V */
#define REFBURST (0x100) /* ADC10 Reference Burst Mode */
#define REFOUT (0x200)   /* ADC10 Enalbe output of Ref. */

#define SREF0 (0x2000)  /* ADC10 Reference Select Bit: 0 */
#define SREF1 (0x4000)  /* ADC10 Reference Select Bit: 1 */
#define SREF2 (0x8000)  /* ADC10 Reference Select Bit: 2 */

Or
#define SREF_0 (0*0x2000u)    /* VR+ = AVCC and VR- = AVSS */
#define SREF_1 (1*0x2000u)    /* VR+ = VREF+ and VR- = AVSS */
#define SREF_2 (2*0x2000u)    /* VR+ = VREF+ and VR- = AVSS */
#define SREF_3 (3*0x2000u)    /* VR+ = VREF+ and VR- = AVSS */
#define SREF_4 (4*0x2000u)    /* VR+ = AVCC and VR- = VREF-/VREF- */
#define SREF_5 (5*0x2000u)    /* VR+ = VREF+ and VR- = VREF-/VREF- */
#define SREF_6 (6*0x2000u)    /* VR+ = VREF+ and VR- = VREF-/VREF- */
#define SREF_7 (7*0x2000u)    /* VR+ = VREF+ and VR- = VREF-/VREF- */

Reference Select and Power Saving Features

The ADC10 internal reference generator is designed for low power applications. The reference generator includes a band-gap voltage source and a separate buffer. The current consumption of each is specified separately in the device-specific data sheet. When REFON = 1, both are enabled and when REFON = 0 both are disabled. The total settling time when REFON becomes set is ? 30 µs.

When REFON = 1, but no conversion is active, the buffer is automatically disabled and automatically re-enabled when needed. When the buffer is disabled, it consumes no current. In this case, the band-gap voltage source remains enabled.

When REFOUT = 1, the REFBURST bit controls the operation of the internal reference buffer.

When REFBURST = 0, the buffer will be on continuously, allowing the reference voltage to be present outside the device continuously. When REFBURST = 1, the buffer is automatically disabled when the ADC10 is not actively converting, and automatically re-enabled when needed.

The internal reference buffer also has selectable speed vs. power settings. When the maximum conversion rate is below 50 kSPS, setting ADC10SR = 1 reduces the current consumption of the buffer approximately 50%.
Low power consumption features

The ADC10 is designed for low power applications. When the ADC10 is not actively converting, the core is automatically disabled and automatically re-enabled when needed. The ADC10OSC is also automatically enabled when needed and disabled when not needed. When the core or oscillator is disabled, it consumes no current.

Sampling and Conversion Setup

Timing

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

- The ADC10SC bit
- The Timer_A Output Unit 1
- The Timer_A Output Unit 0
- The Timer_A Output Unit 2

The polarity of the SHI signal source can be inverted with the ISSH bit. The SHTx bits select the sample period t_sample to be 4, 8, 16, or 64 ADC10CLK cycles. The sampling timer sets SAMPCON high for the selected sample period after synchronization with ADC10CLK. Total sampling time is t_sample plus t_sync. The high-to-low SAMPCON transition starts the analog-to-digital conversion, which requires 13 ADC10CLK cycles.

![Sample Timing Diagram](image_url)

**Figure 22-3. Sample Timing**

ADC10SC Bit 0: Start conversion. Software-controlled sample-and-conversion start. ADC10SC and ENC may be set together with one instruction. ADC10SC is reset automatically.

0: No sample-and-conversion start
1: Start sample-and-conversion

#define ADC10SC (0x001) /* ADC10 Start Conversion */
Timing Consideration

The resistance of the source $R_S$ and $R_I$ affect $t_{sample}$. The following equations can be used to calculate the minimum sampling time for a 10-bit conversion.

$$t_{sample} > (R_S + R_I) \times \ln(2^{11}) \times C_I$$

Substituting the values for $R_I$ and $C_I$ given above, the equation becomes:

$$t_{sample} > (R_S + 2\,\mathrm{k}\Omega) \times 7.625 \times 27\,\mathrm{pF}$$

For example, if $R_S$ is $10\,\mathrm{k}\Omega$, $t_{sample}$ must be greater than $2.47\,\mu\mathrm{s}$.

When the reference buffer is used in burst mode, the sampling time must be greater than the sampling time calculated and the settling time of the buffer, $t_{REFBURST}$:

![Figure 22-4. Analog Input Equivalent Circuit](image)

For example, if $Rs$ is $10\,\mathrm{k}\Omega$, $t_{sample}$ must be greater than $2.47\,\mu\mathrm{s}$ when ADC10SR = 0, or $2.5\,\mu\mathrm{s}$ when ADC10SR = 1. See the device-specific data sheet for parameters.

To calculate the buffer settling time when using an external reference, the formula is:

$$t_{REFBURST} = S R \times V_{Ref} - 0.5\,\mu\mathrm{s}$$

Where:
- $S R = \text{Buffer slew rate} \ (\sim 1\,\mu\mathrm{s}/\mathrm{V}\ \text{when}\ ADC10SR = 0\ \text{and}\ \sim 2\,\mu\mathrm{s}/\mathrm{V}\ \text{when}\ ADC10SR = 1)$
- $V_{Ref} = \text{External reference voltage}$
Conversion Mode

The ADC10 has four operating modes selected by the CONSEQx bits

CONSEQx Mode Operation
00 Single channel single-conversion A single channel is converted once.
01 Sequence-of-channels A sequence of channels is converted once.
10 Repeat single channel A single channel is converted repeatedly.
11 Repeat sequence-of-channels A sequence of channels is converted repeatedly.

CONSEQx Bits 2-1 Conversion sequence mode select
00 Single-channel-single-conversion
01 Sequence-of-channels
10 Repeat-single-channel
11 Repeat-sequence-of-channels

#define CONSEQ0 (0x0002) /**< ADC10 Conversion Sequence Select 0 */
#define CONSEQ1 (0x0004) /**< ADC10 Conversion Sequence Select 1 */
Or
#define CONSEQ_0 (0*2u) /**< Single channel single conversion */
#define CONSEQ_1 (1*2u) /**< Sequence of channels */
#define CONSEQ_2 (2*2u) /**< Repeat single channel */
#define CONSEQ_3 (3*2u) /**< Repeat sequence of channels */

More about Conversion

MSC Bit

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

Stopping ADC10 activity depends on the mode of operation. The recommended ways to stop an active conversion or conversion sequence are:

- Resetting ENC in single-channel single-conversion mode stops a conversion immediately and the results are unpredictable. For correct results, poll the ADC10BUSY bit until reset before clearing ENC.
- Resetting ENC during repeat-single-channel operation stops the converter at the end of the current conversion.
- Resetting ENC 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 ENC bit. Conversion data is unreliable.

Set CONSEQ to 0 (ref continuous mode) and reset ENC stops any types of conversion.
**ADC10 Data Transfer Controller (DTC)**

The ADC10 includes a data transfer controller (DTC) to automatically transfer conversion results from ADC10MEM to other on-chip memory locations. The DTC is enabled by setting the ADC10DTC1 register to a nonzero value.

When the DTC is enabled, each time the ADC10 completes a conversion and loads the result to ADC10MEM, a data transfer is triggered. No software intervention is required to manage the ADC10 until the predefined amount of conversion data has been transferred. Each DTC transfer requires one CPU MCLK. To avoid any bus contention during the DTC transfer, the CPU is halted, if active, for the one MCLK required for the transfer. A DTC transfer must not be initiated while the ADC10 is busy. Software must ensure that no active conversion or sequence is in progress when the DTC is configured:

```
; ADC10 activity test
BIC.W #ENC,&ADC10CTL0 ;
busy_test BIT.W #BUSY,&ADC10CTL1 ;
  JNZ busy_test ;
  MOV.W #xxx,&ADC10SA ; Safe
  MOV.B #xx,&ADC10DTC1 ;

; continue setup
```
Transfer Mode

One-Block Transfer Mode

The one-block mode is selected if the ADC10TB is reset. The value n in ADC10DTC1 defines the total number of transfers for a block. The block start address is defined anywhere in the MSP430 address range using the 16-bit register ADC10SA. The block ends at ADC10SA+2n-2.

The internal address pointer is initially equal to ADC10SA and the internal transfer counter is initially equal to 'n'. The internal pointer and counter are not visible to software. The DTC transfers the word-value of ADC10MEM to the address pointer ADC10SA. After each DTC transfer, the internal address pointer I incremented by two and the internal transfer counter is decremented by one.

The DTC transfers continue with each loading of ADC10MEM, until the internal transfer counter becomes equal to zero. No additional DTC transfers will occur until a write to ADC10SA. When using the DTC in the one-block mode, the ADC10IFG flag is set only after a complete block has been transferred.

```
#define ADC10TB (0x008) /* ADC10 two-block mode */
```

![Diagram of one-block transfer mode]

- SFR_16BIT(ADC10MEM); /* ADC10 Memory */
- SFR_16BIT(ADC10SA); /* ADC10 Data Transfer Start Address */
- SFR_8BIT(ADC10DTC0); /* ADC10 Data Transfer Control 0 */
- SFR_8BIT(ADC10DTC1); /* ADC10 Data Transfer Control 1 */
**Transfer Mode**

**Two-Block Transfer Mode**

The two-block mode is selected if the ADC10TB bit is set. The value n in ADC10DTC1 defines the number of transfers for one block. The address range of the first block is defined anywhere in the MSP430 address range with the 16-bit register ADC10SA. The first block ends at ADC10SA+2n-2. The address range for the second block is defined as SA+2n to SA+4n-2.

The internal address pointer is initially equal to ADC10SA and the internal transfer counter is initially equal to 'n'. The internal pointer and counter are not visible to software. The DTC transfers the word-value of ADC10MEM to the address pointer ADC10SA. After each DTC transfer the internal address pointer is incremented by two and the internal transfer counter is decremented by one. The DTC transfers continue, with each loading of ADC10MEM, until the internal transfer counter becomes equal to zero. At this point, block one is full and both the ADC10IFG flag the ADC10B1 bit are set. The user can test the ADC10B1 bit to determine that block one is full. The DTC continues with block two. The internal transfer counter is automatically reloaded with 'n'. At the next load of the ADC10MEM, the DTC begins transferring conversion results to block two. After n transfers have completed, block two is full. The ADC10IFG flag is set and the ADC10B1 bit is cleared. User software can test the cleared ADC10B1 bit to determine that block two is full.

```c
SFR_16BIT(ADC10MEM);  // ADC10 Memory */
SFR_16BIT(ADC10SA);    // ADC10 Data Transfer Start Address */
SFR_8BIT(ADC10DTC0);   // ADC10 Data Transfer Control 0 */
SFR_8BIT(ADC10DTC1);   // ADC10 Data Transfer Control 1 */
#define ADC10TB (0x008)   // ADC10 two-block mode */
```
Continuous Transfer

A continuous transfer is selected if ADC10CT bit is set. The DTC will not stop after block one in (one-block mode) or block two (two-block mode) has been transferred. The internal address pointer and transfer counter are set equal to ADC10SA and n respectively. Transfers continue starting in block one. If the ADC10CT bit is reset, DTC transfers cease after the current completion of transfers into block one (in the one-block mode) or block two (in the two-block mode) have been transfer.

```
#define ADC10CT (0x004)   /* ADC10 continuous transfer */
```

DTC Transfer Cycle Time

For each ADC10MEM transfer, the DTC requires one or two MCLK clock cycles to synchronize, one for the actual transfer (while the CPU is halted), and one cycle of wait time. Because the DTC uses MCLK, the DTC cycle time is dependent on the MSP430 operating mode and clock system setup.

If the MCLK source is active, but the CPU is off, the DTC uses the MCLK source for each transfer, without re-enabling the CPU. If the MCLK source is off, the DTC temporarily restarts MCLK, sourced with DCOCLK, only during a transfer. The CPU remains off and after the DTC transfer, MCLK is again turned off. The maximum DTC cycle time for all operating modes is show below:

**Table 22-2. Maximum DTC Cycle Time**

<table>
<thead>
<tr>
<th>CPU Operating Mode</th>
<th>Clock Source</th>
<th>Maximum DTC Cycle Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active mode</td>
<td>MCLK = DCOCLK</td>
<td>3 MCLK cycles</td>
</tr>
<tr>
<td>Active mode</td>
<td>MCLK = LFXT1CLK</td>
<td>3 MCLK cycles</td>
</tr>
<tr>
<td>Low-power mode LPM0/1</td>
<td>MCLK = DCOCLK</td>
<td>4 MCLK cycles</td>
</tr>
<tr>
<td>Low-power mode LPM3/4</td>
<td>MCLK = DCOCLK</td>
<td>4 MCLK cycles + 2 μs(1)</td>
</tr>
<tr>
<td>Low-power mode LPM0/1</td>
<td>MCLK = LFXT1CLK</td>
<td>4 MCLK cycles</td>
</tr>
<tr>
<td>Low-power mode LPM3</td>
<td>MCLK = LFXT1CLK</td>
<td>4 MCLK cycles</td>
</tr>
<tr>
<td>Low-power mode LPM4</td>
<td>MCLK = LFXT1CLK</td>
<td>4 MCLK cycles + 2 μs(1)</td>
</tr>
</tbody>
</table>

(1) The additional 2 μs are needed to start the DCOCCLK. See the device-specific data sheet for parameters.
Use of Internal Temperature Sensor

To use the on-chip temperature sensor, the user selects the analog input channel INCHx = 1010. Any other configuration is done as if an external channel was selected, including reference selection, conversion-memory selection, etc.

When using the temperature sensor, the sample period must be greater than 30 µs. The temperature sensor offset error is large. Deriving absolute temperature values in the application requires calibration. See the device-specific data sheet for the parameters.

Selecting the temperature sensor automatically turns on the on-chip reference generator as a voltage source for the temperature sensor. However, it does not enable the V REF+ output or affect the reference selections for the conversion. The reference choices for converting the temperature sensor are the same as with any other channel.

![Figure 22-13. Typical Temperature Sensor Transfer Function](image-url)
10-Bit ADC, Temperature Sensor and Built-In \( V_{\text{MID}} \) (MSP430G2x53 Only)

over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>( V_{\text{CC}} )</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I_{\text{SENSOR}} )</td>
<td>Temperature sensor supply current(^{(1)})</td>
<td>( \text{REFON} = 0 ), INCHx = 0Ah, ( T_A = 25^\circ\text{C} )</td>
<td>3 V</td>
<td>60</td>
<td></td>
<td>( \mu\text{A} )</td>
</tr>
<tr>
<td>( T_{\text{CSENSOR}} )</td>
<td>Sample time required if channel 10 is selected (^{(2)})</td>
<td>( \text{ADC10ON} = 1 ), INCHx = 0Ah, Error of conversion result ( \leq 1 ) LSB</td>
<td>3 V</td>
<td>3.55</td>
<td></td>
<td>mV/°C</td>
</tr>
<tr>
<td>( t_{\text{Sensor(sample)}} )</td>
<td>Current into divider at channel 11</td>
<td>( \text{ADC10ON} = 1 ), INCHx = 0Ah</td>
<td>3 V</td>
<td></td>
<td></td>
<td>( \mu\text{s} )</td>
</tr>
<tr>
<td>( V_{\text{MID}} )</td>
<td>( V_{\text{CC}} ) divider at channel 11</td>
<td>( \text{ADC10ON} = 1 ), INCHx = 0Ah, ( V_{\text{MID}} = 0.5 \times V_{\text{CC}} )</td>
<td>3 V</td>
<td>1.5</td>
<td></td>
<td>V</td>
</tr>
<tr>
<td>( t_{\text{MID(sample)}} )</td>
<td>Sample time required if channel 11 is selected (^{(5)})</td>
<td>( \text{ADC10ON} = 1 ), INCHx = 0Ah, Error of conversion result ( \leq 1 ) LSB</td>
<td>3 V</td>
<td>1220</td>
<td></td>
<td>ns</td>
</tr>
</tbody>
</table>

\(^{(1)}\) The sensor current \( I_{\text{SENSOR}} \) is consumed if \( \text{ADC10ON} = 1 \) and \( \text{REFON} = 1 \) or \( \text{ADC10ON} = 1 \) and INCH = 0Ah and sample signal is high. When \( \text{REFON} = 1 \), \( I_{\text{SENSOR}} \) is included in \( I_{\text{REF}} \). When \( \text{REFON} = 0 \), \( I_{\text{SENSOR}} \) applies during conversion of the temperature sensor input (INCH = 0Ah).

\(^{(2)}\) The following formula can be used to calculate the temperature sensor output voltage:
\[
V_{\text{Sensor,typ}} = T_{\text{CSensor}} \times (273 + T \text{ [°C]} ) + V_{\text{Offset,Sensor}} \text{ [mV]} \]
or
\[
V_{\text{Sensor,typ}} = T_{\text{CSensor}} \times T \text{ [°C]} + V_{\text{Sensor}}(T_a = 0 \text{°C}) \text{ [mV]}
\]

\(^{(3)}\) The typical equivalent impedance of the sensor is 51 kΩ. The sample time required includes the sensor-on time \( t_{\text{Sensor(on)}} \).

\(^{(4)}\) No additional current is needed. The \( V_{\text{MID}} \) is used during sampling.

\(^{(5)}\) The on-time \( t_{\text{MID(on)}} \) is included in the sampling time \( t_{\text{MID(sample)}} \); no additional on time is needed.

---

10-Bit ADC, Timing Parameters (MSP430G2x53 Only)

over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>( V_{\text{CC}} )</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_{\text{ADC10CLK}} )</td>
<td>ADC10 input clock frequency</td>
<td>( \text{ADC10SR} = 0 ), ( \text{ADC10SR} = 1 )</td>
<td>3 V</td>
<td>0.45</td>
<td>6.3</td>
<td>MHz</td>
</tr>
<tr>
<td>( f_{\text{ADC10SC}} )</td>
<td>ADC10 built-in oscillator frequency</td>
<td>( \text{ADC10DIVX} = 0 ), ( \text{ADC10SSELx} = 0 ), ( f_{\text{ADC10CLK}} = f_{\text{ADC10SC}} )</td>
<td>3 V</td>
<td>3.7</td>
<td>6.3</td>
<td>MHz</td>
</tr>
<tr>
<td>( t_{\text{CONVERT}} )</td>
<td>Conversion time</td>
<td>( \text{ADC10 built-in oscillator, ADC10SSELx} = 0 ), ( f_{\text{ADC10CLK}} = f_{\text{ADC10SC}} )</td>
<td>3 V</td>
<td>2.06</td>
<td>3.51</td>
<td>( \mu\text{s} )</td>
</tr>
<tr>
<td>( t_{\text{ADC10ON}} )</td>
<td>Turn-on settling time of the ADC</td>
<td>( \text{13} \times \frac{\text{ADC10DIV}}{1} \times f_{\text{ADC10CLK}} )</td>
<td>3 V</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

\(^{(1)}\) The condition is that the error in a conversion started after \( t_{\text{ADC10ON}} \) is less than 0.5 LSB. The reference and input signal are already settled.

---

10-Bit ADC, Linearity Parameters (MSP430G2x53 Only)

over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>TEST CONDITIONS</th>
<th>( V_{\text{CC}} )</th>
<th>MIN</th>
<th>TYP</th>
<th>MAX</th>
<th>UNIT</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E_1 )</td>
<td>Integral linearity error</td>
<td>3 V</td>
<td>±1</td>
<td></td>
<td>±2</td>
<td>LSB</td>
</tr>
<tr>
<td>( E_2 )</td>
<td>Differential linearity error</td>
<td>3 V</td>
<td>±1</td>
<td></td>
<td>±2</td>
<td>LSB</td>
</tr>
<tr>
<td>( E_3 )</td>
<td>Offset error</td>
<td>3 V</td>
<td>±1</td>
<td></td>
<td>±2</td>
<td>LSB</td>
</tr>
<tr>
<td>( E_4 )</td>
<td>Gain error</td>
<td>3 V</td>
<td>±1.1</td>
<td>±2</td>
<td></td>
<td>LSB</td>
</tr>
<tr>
<td>( E_5 )</td>
<td>Total unadjusted error</td>
<td>3 V</td>
<td>±2</td>
<td>±5</td>
<td></td>
<td>LSB</td>
</tr>
</tbody>
</table>
**ADC10 Interrupts**

One interrupt and one interrupt vector are associated with the ADC10 as shown in Figure 22-16. When the DTC is not used (ADC10DTC1 = 0) ADC10IFG is set when conversion results are loaded into ADC10MEM. When DTC is used (ADC10DTC1 > 0) ADC10IFG is set when a block transfer completes and the internal transfer counter ‘n’ = 0. If both the ADC10IE and the GIE bits are set, then the ADC10IFG flag generates an interrupt request. The ADC10IFG flag is automatically reset when the interrupt request is serviced or may be reset by software.

![ADC10 Interrupt System](image)

**Figure 22-16. ADC10 Interrupt System**

**ADC10 Registers**

<table>
<thead>
<tr>
<th>Register</th>
<th>Short Form</th>
<th>Register Type</th>
<th>Address</th>
<th>Initial State</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC10 input enable register 0</td>
<td>ADC10AE0</td>
<td>Read/write</td>
<td>04Ah</td>
<td>Reset with POR</td>
</tr>
<tr>
<td>ADC10 input enable register 1</td>
<td>ADC10AE1</td>
<td>Read/write</td>
<td>04Bh</td>
<td>Reset with POR</td>
</tr>
<tr>
<td>ADC10 control register 0</td>
<td>ADC10CTL0</td>
<td>Read/write</td>
<td>01B0h</td>
<td>Reset with POR</td>
</tr>
<tr>
<td>ADC10 control register 1</td>
<td>ADC10CTL1</td>
<td>Read/write</td>
<td>01B2h</td>
<td>Reset with POR</td>
</tr>
<tr>
<td>ADC10 memory</td>
<td>ADC10MEM</td>
<td>Read</td>
<td>01B4h</td>
<td>Unchanged</td>
</tr>
<tr>
<td>ADC10 data transfer control register 0</td>
<td>ADC10DTC0</td>
<td>Read/write</td>
<td>048h</td>
<td>Reset with POR</td>
</tr>
<tr>
<td>ADC10 data transfer control register 1</td>
<td>ADC10DTC1</td>
<td>Read/write</td>
<td>049h</td>
<td>Reset with POR</td>
</tr>
<tr>
<td>ADC10 data transfer start address</td>
<td>ADC10SA</td>
<td>Read/write</td>
<td>01BCh</td>
<td>0200h with POR</td>
</tr>
</tbody>
</table>
### 22.3.1 ADC10CTL0, ADC10 Control Register 0

<table>
<thead>
<tr>
<th>Bit 15-8</th>
<th>Bit 12-11</th>
<th>Bit 10</th>
<th>Bit 9</th>
<th>Bit 8</th>
<th>Bit 7</th>
<th>Bit 6</th>
<th>Bit 5</th>
<th>Bit 4</th>
<th>Bit 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>SREFx</td>
<td>ADC10SHTx</td>
<td>ADC10SR</td>
<td>REFOUT</td>
<td>REFBURST</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rw(0)</td>
<td>rw(0)</td>
<td>rw(0)</td>
<td>rw(0)</td>
<td>rw(0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSC</td>
<td>REF2_5V</td>
<td>REFON</td>
<td>ADC10ON</td>
<td>ADC10IE</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rw(0)</td>
<td>rw(0)</td>
<td>rw(0)</td>
<td>rw(0)</td>
<td>rw(0)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Can be modified only when ENC = 0

**SREFx**

Bits 15-13 Select reference

- 000: \( V_{\text{ref}} = V_{\text{CC}} \) and \( V_{L} = V_{\text{SS}} \)
- 001: \( V_{\text{ref}} = V_{\text{REF}}, \) and \( V_{L} = V_{\text{SS}} \)
- 010: \( V_{\text{ref}} = V_{\text{REF}}, \) and \( V_{L} = V_{\text{SS}} \)
- 011: \( V_{\text{ref}} = \text{Buffered } V_{\text{REF}}, \) and \( V_{L} = V_{\text{SS}} \)
- 100: \( V_{\text{ref}} = V_{\text{CC}}, \) and \( V_{L} = V_{\text{REF}} / V_{\text{AREF}} \)
- 101: \( V_{\text{ref}} = V_{\text{REF}}, \) and \( V_{L} = V_{\text{REF}} / V_{\text{AREF}} \)
- 110: \( V_{\text{ref}} = \text{Buffered } V_{\text{REF}}, \) and \( V_{L} = V_{\text{REF}} / V_{\text{AREF}} \)
- 111: \( V_{\text{ref}} = \text{Buffered } V_{\text{REF}}, \) and \( V_{L} = V_{\text{REF}} / V_{\text{AREF}} \)

**ADC10SHTx**

Bits 12-11 ADC10 sample-and-hold time

- 00: \( 4 \times \text{ADC10CLKs} \)
- 01: \( 8 \times \text{ADC10CLKs} \)
- 10: \( 16 \times \text{ADC10CLKs} \)
- 11: \( 64 \times \text{ADC10CLKs} \)

**ADC10SR**

Bit 10 ADC10 sampling rate. This bit selects the reference buffer drive capability for the maximum sampling rate. Setting ADC10SR reduces the current consumption of the reference buffer.

- 0: Reference buffer supports up to ~200 kspS
- 1: Reference buffer supports up to ~50 kspS

**REFOUT**

Bit 9 Reference output

- 0: Reference output off
- 1: Reference output on

**REFBURST**

Bit 8 Reference burst.

- 0: Reference buffer on continuously
- 1: Reference buffer on only during sample-and-conversion

**MSC**

Bit 7 Multiple sample and conversion. Valid only for sequence or repeated modes.

- 0: The sampling requires a rising edge of the SHI signal to trigger each sample-and-conversion.
- 1: 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

**REF2_5V**

Bit 6 Reference-generator voltage. REFO must also be set.

- 0: 1.5 V
- 1: 2.5 V

**REFON**

Bit 5 Reference generator on

- 0: Reference off
- 1: Reference on

**ADC10ON**

Bit 4 ADC10 on

- 0: ADC10 off
- 1: ADC10 on

**ADC10IE**

Bit 3 ADC10 interrupt enable

- 0: Interrupt disabled
- 1: Interrupt enabled
<table>
<thead>
<tr>
<th>Field</th>
<th>Bit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC10FG</td>
<td>2</td>
<td>ADC10 interrupt flag. This bit is set if ADC10MEM is loaded with a conversion result. It is automatically reset when the interrupt request is accepted, or it may be reset by software. When using the DTC this flag is set when a block of transfers is completed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: No interrupt pending&lt;br&gt;1: Interrupt pending</td>
</tr>
<tr>
<td>ENC</td>
<td>1</td>
<td>Enable conversion</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: ADC10 disabled&lt;br&gt;1: ADC10 enabled</td>
</tr>
<tr>
<td>ADC10SC</td>
<td>0</td>
<td>Start conversion. Software-controlled sample-and-conversion start. ADC10SC and ENC may be set together with one instruction. ADC10SC is reset automatically.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0: No sample-and-conversion start&lt;br&gt;1: Start sample-and-conversion</td>
</tr>
</tbody>
</table>
### 22.3.2 ADC10CTL1, ADC10 Control Register 1

<table>
<thead>
<tr>
<th>INCHx</th>
<th>SHSx</th>
<th>ADC10DF</th>
<th>ISSH</th>
</tr>
</thead>
<tbody>
<tr>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ADC10DIVx</th>
<th>ADC10SSELx</th>
<th>CONSEQx</th>
<th>ADC10BUSY</th>
</tr>
</thead>
<tbody>
<tr>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>0</td>
<td>r-0</td>
</tr>
</tbody>
</table>

- **INCHx** Bits 15-12: Input channel select. These bits select the channel for a single-conversion or the highest channel for a sequence of conversions.
  - 0000: A0
  - 0001: A1
  - 0010: A2
  - 0011: A3
  - 0100: A4
  - 0101: A5
  - 0110: A6
  - 0111: A7
  - 1000: $V_{\text{REF+}}$
  - 1001: $V_{\text{REF}}$/$V_{\text{REF-}}$
  - 1010: Temperature sensor
  - 1011: $(V_{\text{CC}} - V_{\text{SS}}) / 2$
  - 1100: $(V_{\text{CC}} - V_{\text{SS}}) / 2$, A12 on MSP430x2xx devices
  - 1101: $(V_{\text{CC}} - V_{\text{SS}}) / 2$, A13 on MSP430x2xx devices
  - 1110: $(V_{\text{CC}} - V_{\text{SS}}) / 2$, A14 on MSP430x2xx devices
  - 1111: $(V_{\text{CC}} - V_{\text{SS}}) / 2$, A15 on MSP430x2xx devices

- **SHSx** Bits 11-10: Sample-and-hold source select
  - 00: ADC10SC bit
  - 01: Timer_A.OUT1
  - 10: Timer_A.OUT0
  - 11: Timer_A.OUT2 (Timer_A.OUT1 on MSP430x20x2 devices)

- **ADC10DF** Bit 9: ADC10 data format
  - 0: Straight binary
  - 1: 2s complement

- **ISSH** Bit 8: Invert signal sample-and-hold
  - 0: The sample-input signal is not inverted.
  - 1: The sample-input signal is inverted.

- **ADC10DIVx** Bits 7-5: ADC10 clock divider
  - 000: /1
  - 001: /2
  - 010: /3
  - 011: /4
  - 100: /5
  - 101: /6
  - 110: /7
  - 111: /8

- **ADC10SSELx** Bits 4-3: ADC10 clock source select
  - 00: ADC100SC
  - 01: ACLK
  - 10: MCLK
  - 11: SMCLK

*Can be modified only when ENC = 0*
CONSEQx  Bits 2-1  Conversion sequence mode select
  00  Single-channel-single-conversion
  01  Sequence-of-channels
  10  Repeat-single-channel
  11  Repeat-sequence-of-channels

ADC10BUSY  Bit 0  ADC10 busy. This bit indicates an active sample or conversion operation
  0  No operation is active.
  1  A sequence, sample, or conversion is active.

22.3.3  ADC10AE0, Analog (Input) Enable Control Register 0

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ADC10AE0x</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
</tr>
</tbody>
</table>

ADC10AE0x  Bits 7-0  ADC10 analog enable. These bits enable the corresponding pin for analog input. BIT0 corresponds to A0, BIT1 corresponds to A1, etc.
  0  Analog input disabled
  1  Analog input enabled

22.3.4  ADC10AE1, Analog (Input) Enable Control Register 1 (MSP430x22xx only)

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ADC10AE1x</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
</tr>
</tbody>
</table>

ADC10AE1x  Bits 7-4  ADC10 analog enable. These bits enable the corresponding pin for analog input. BIT4 corresponds to A12, BIT5 corresponds to A13, BIT6 corresponds to A14, and BIT7 corresponds to A15.
  0  Analog input disabled
  1  Analog input enabled

Reserved  Bits 3-0  Reserved

22.3.5  ADC10MEM, Conversion-Memory Register, Binary Format

<table>
<thead>
<tr>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>r0</td>
<td>r0</td>
<td>r0</td>
<td>r0</td>
<td>r0</td>
<td>r0</td>
<td>r</td>
<td>r</td>
</tr>
</tbody>
</table>

7 6 5 4 3 2 1 0  Conversion Results

Conversion Results  Bits 15-0  The 10-bit conversion results are right justified, straight-binary format. Bit 9 is the MSB. Bits 15-10 are always 0.
### 22.3.6 ADC10MEM, Conversion-Memory Register, 2s Complement Format

<table>
<thead>
<tr>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td>r</td>
<td>r</td>
</tr>
</tbody>
</table>

**Conversion Results**

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>r</td>
<td>r</td>
<td>r0</td>
<td>r0</td>
<td>r0</td>
<td>r0</td>
<td>r0</td>
<td>r0</td>
</tr>
</tbody>
</table>

**Conversion Results**

Bits 15-0 The 10-bit conversion results are left-justified, 2s complement format. Bit 15 is the MSB. Bits 5-0 are always 0.

### 22.3.7 ADC10DTC0, Data Transfer Control Register 0

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>r0</td>
<td>r0</td>
<td>r0</td>
<td>r0</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>r-(0)</td>
<td>rw-(0)</td>
</tr>
</tbody>
</table>

**Reserved**

Bits 7-4 Reserved. Always read as 0.

**ADC10TB**

Bit 3 ADC10 two-block mode

0 One-block transfer mode

1 Two-block transfer mode

**ADC10CT**

Bit 2 ADC10 continuous transfer

0 Data transfer stops when one block (one-block mode) or two blocks (two-block mode) have completed.

1 Data is transferred continuously. DTC operation is stopped only if ADC10CT cleared, or ADC10SA is written to.

**ADC10B1**

Bit 1 ADC10 block one. This bit indicates for two-block mode which block is filled with ADC10 conversion results. ADC10B1 is valid only after ADC10IFG has been set the first time during DTC operation. ADC10TB must also be set.

0 Block 2 is filled

1 Block 1 is filled

**ADC10FETCH**

Bit 0 This bit should normally be reset.

### 22.3.8 ADC10DTC1, Data Transfer Control Register 1

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
</tr>
</tbody>
</table>

**DTC Transfers**

Bits 7-0 DTC transfers. These bits define the number of transfers in each block.

0 DTC is disabled

01h-0FFh Number of transfers per block

### 22.3.9 ADC10SA, Start Address Register for Data Transfer

<table>
<thead>
<tr>
<th>15</th>
<th>14</th>
<th>13</th>
<th>12</th>
<th>11</th>
<th>10</th>
<th>9</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
</tr>
</tbody>
</table>

**ADC10SAx**

<table>
<thead>
<tr>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>rw-(0)</td>
<td>r0</td>
</tr>
</tbody>
</table>

**ADC10SAx**

Bits 15-1 ADC10 start address. These bits are the start address for the DTC. A write to register ADC10SA is required to initiate DTC transfers.

**Unused**

Bit 0 Unused, Read only. Always read as 0.