Lecture 5: Analog-to-Digital Conversion

TNE097 Microcomputer Systems

Naveen Venkategowda

2025-11-20

LINKÖPING UNIVERSITY

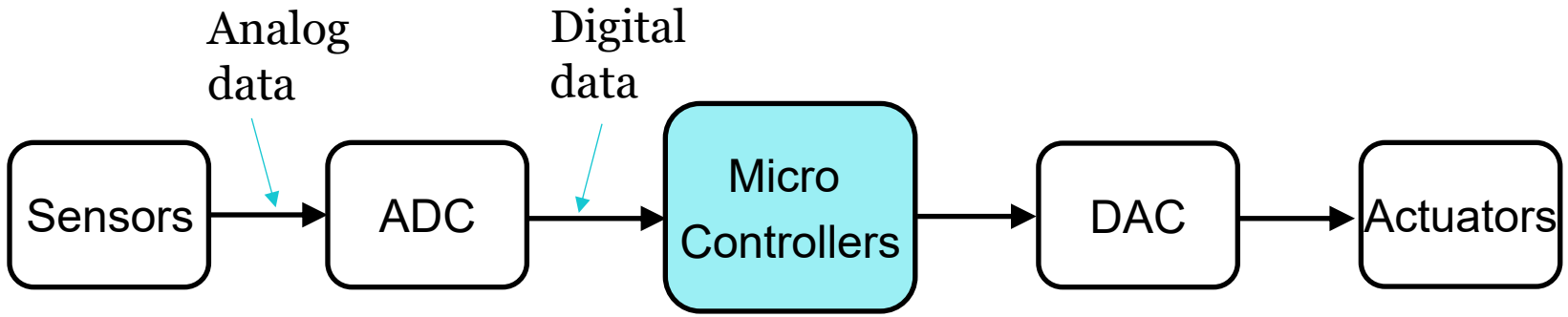
Information

- Reference: Chapter 22, ADC10, *MSP 430 User Guide*, Manul/Document number SLAU144J
- Key points:
  - What are the different settings/ features of ADC?
  - How to configure ADC?
  - How to access ADC output data?
  - How does interrupt operate?
  - Registers associated with ADC

Principles of Analog-to-Digital Conversion

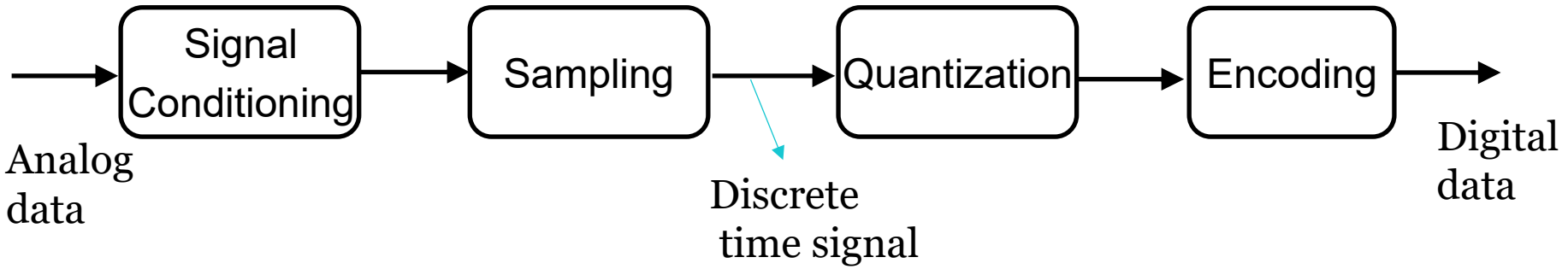
Why Analog to Digital Conversion?

- Microcontrollers process digital data (bit 0 and 1)
- Analog signals: continuous time and real number values
- Many real-world signals are analog
  - Temperature
  - Audio signals

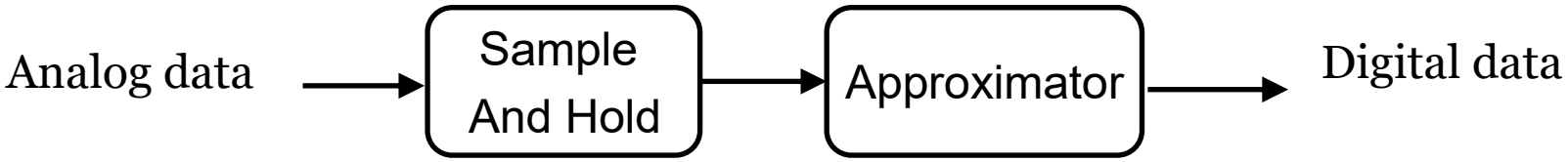


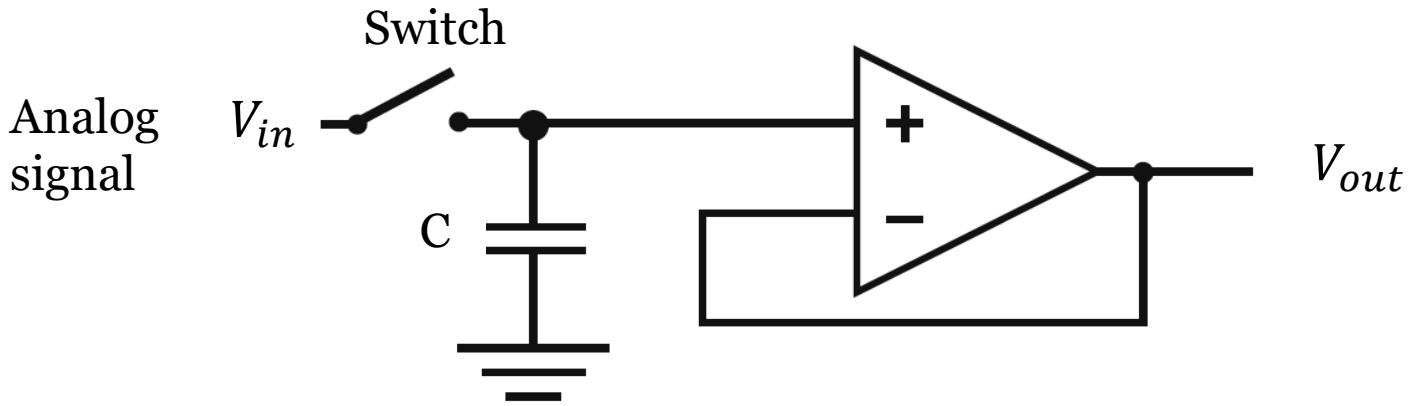
Basic Blocks of ADC

- Signal Conditioning
  - Noise filtering
  - Bandwidth reduction
  - Scaling



Sample and Hold Circuit





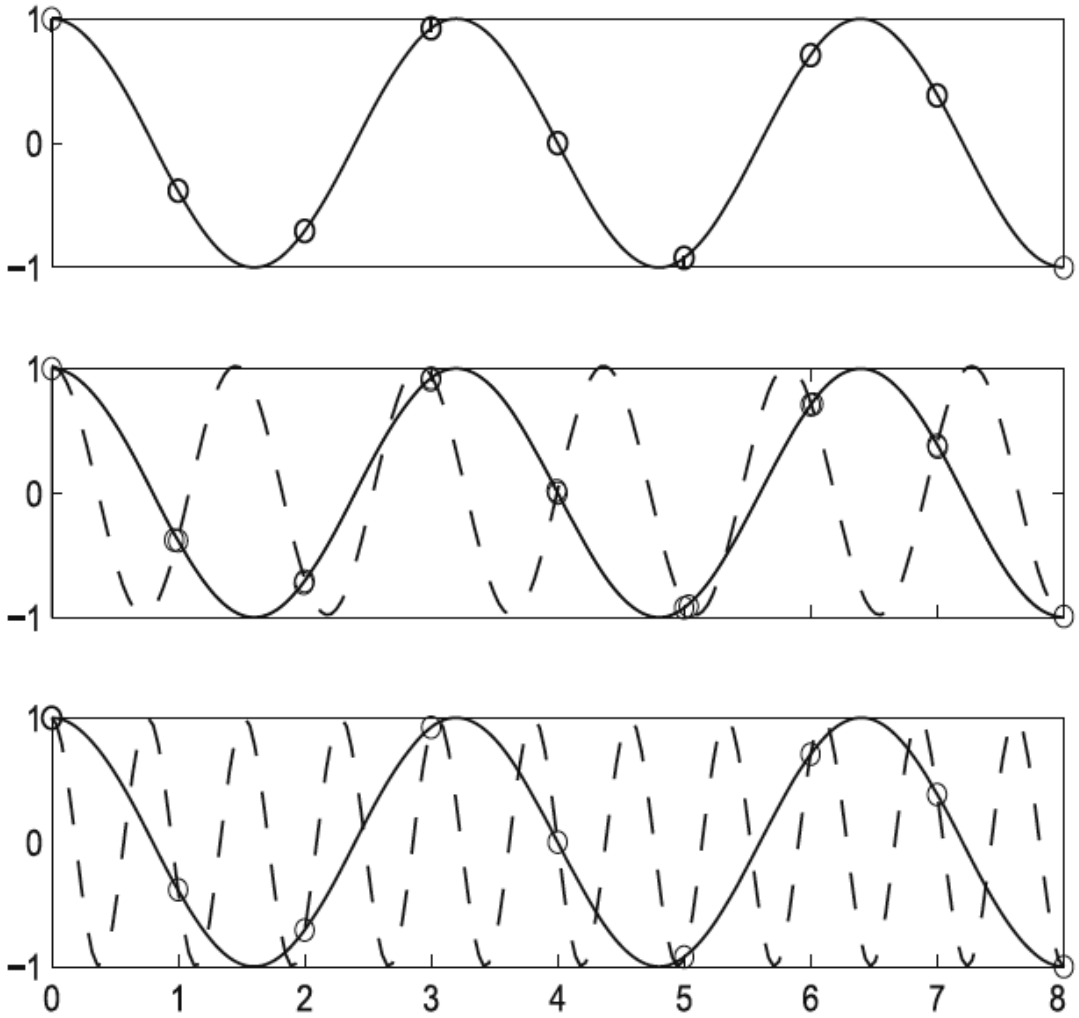
Simple sample and hold circuit

Sampling

- Sampling: Obtaining periodic snapshots of continuous time signal
- Nyquist Sampling Theorem:
  - *A bandlimited continuous-time signal can be sampled and perfectly reconstructed from its samples if the signal is sampled at least twice the highest frequency component.*
  - Sampling rate: f_s Hz or samples per second
  - Signal highest frequency: f Hz
  - Nyquist sampling rate: $f_s \geq 2f$
- Example:
  - Sound signals: 0-20 kHz, CD sampling rate: 44.1kHz

Aliasing

- Aliasing occurs when sampling frequency is less than Nyquist rate
- Inability to distinguish samples from different signals



Low frequency signal $f_s \geq 2f$

Higher frequency signal $f_s \leq 2f$

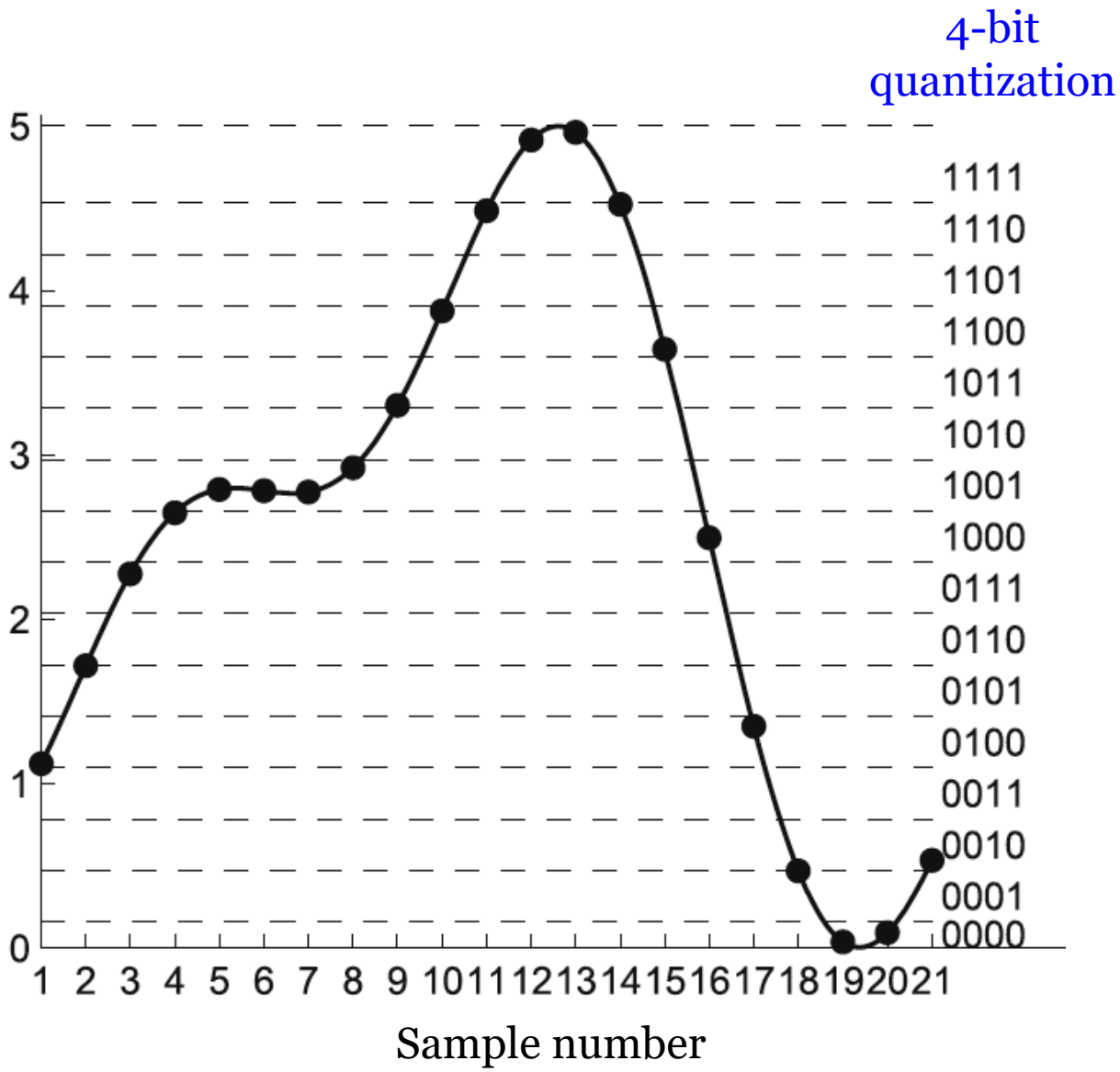
Aliasing effect

Quantization

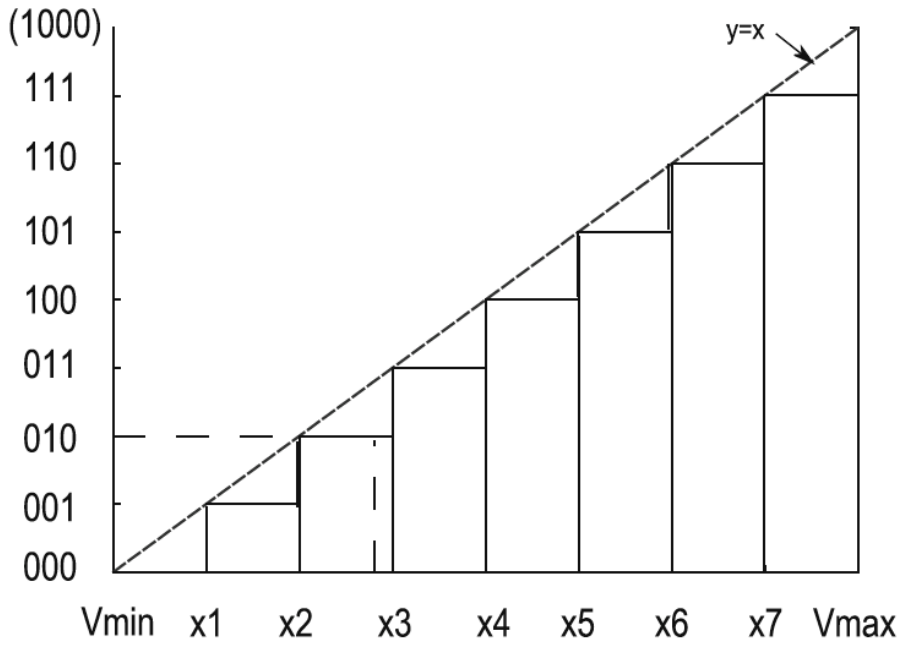
- Convert discrete-time signals to signals with finite representation of amplitude
- N-bit quantization: 2^N levels of signal amplitude
- Difference between each level
  - $\Delta = \frac{V_{R+} - V_{R-}}{2^N}$
- Encoding: assigning binary values to each level
  - Straightforward binary
  - 2´s complement binary

Quantization

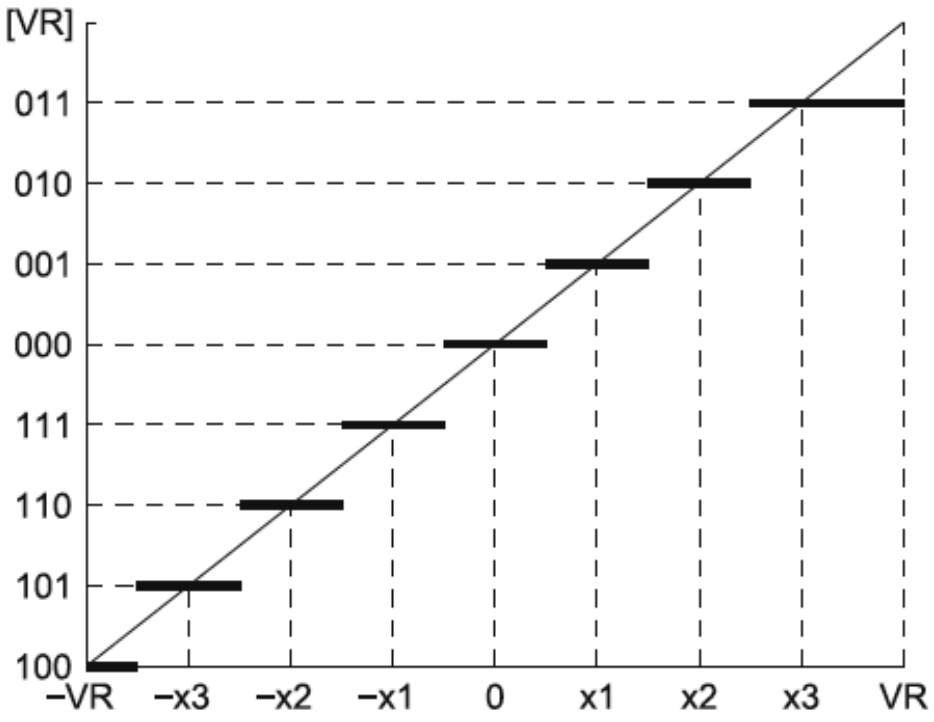
- Converting discrete-time signals to signals with finite representation of amplitude
- N-bit quantization: 2^N levels of signal amplitude
- Difference between each level
  - $\Delta = \frac{V_{R+} - V_{R-}}{2^N}$
- Encoding: assigning binary values to each level
  - Straightforward binary



Digital Number Formats



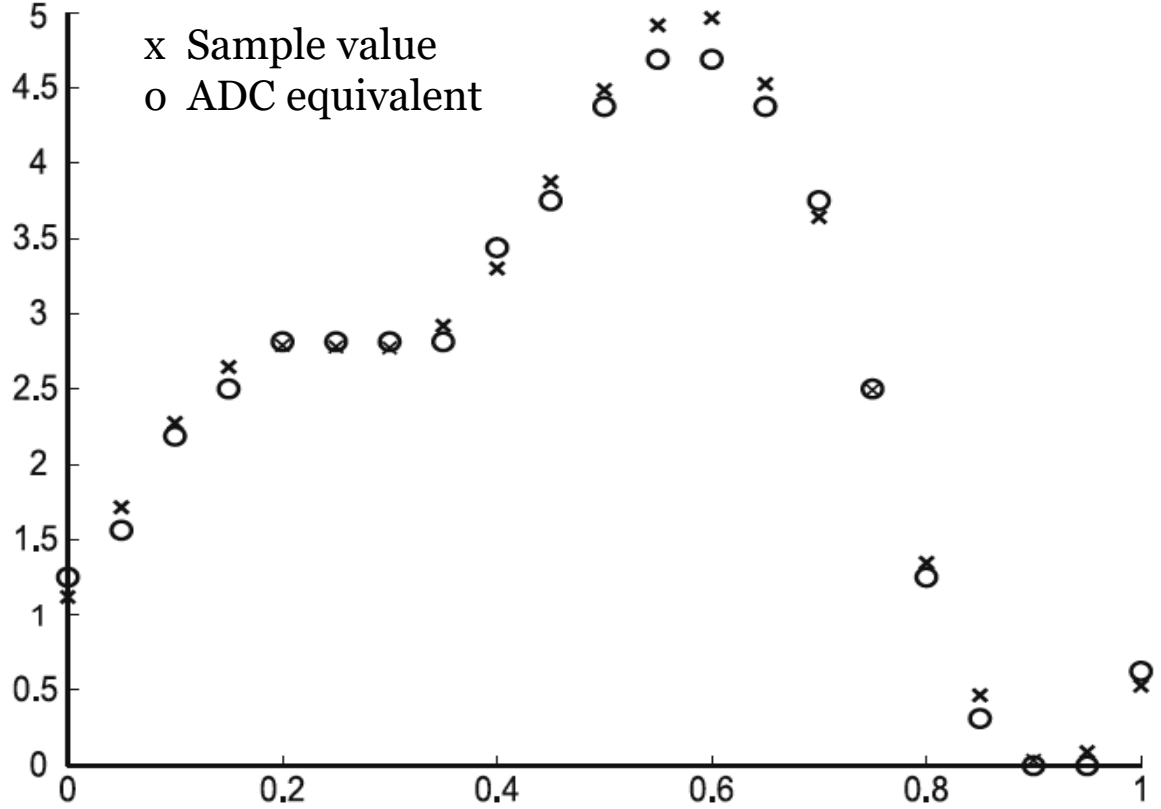
Straightforward binary



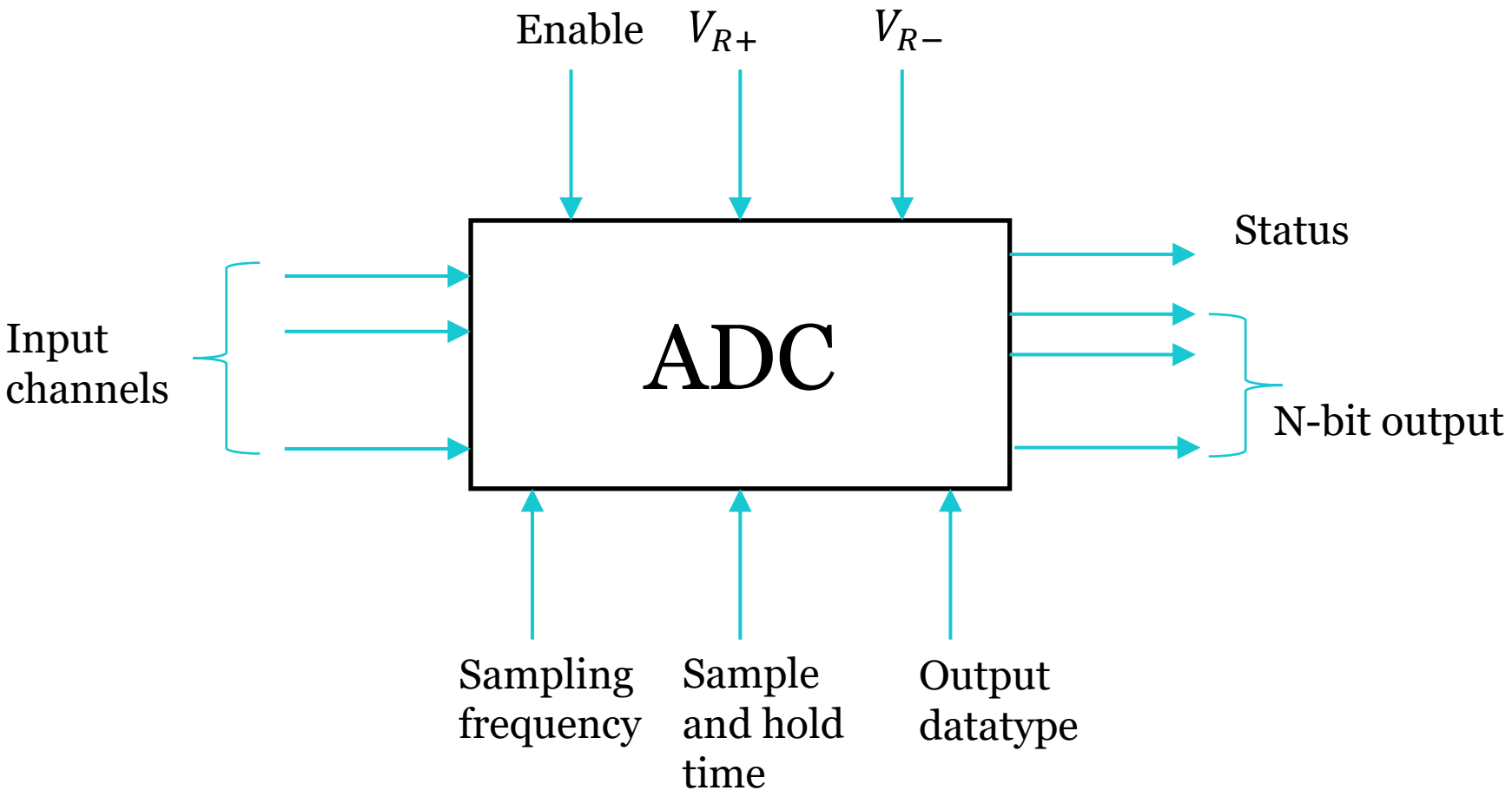
2´s complement binary

ADC Performance Metrics

- Number of samples: sampling rate
  - More samples better reconstruction of signal
- Resolution or Accuracy
  - Quantization error: more bits per sample lesser approximation error
  - More bits implies higher resolution

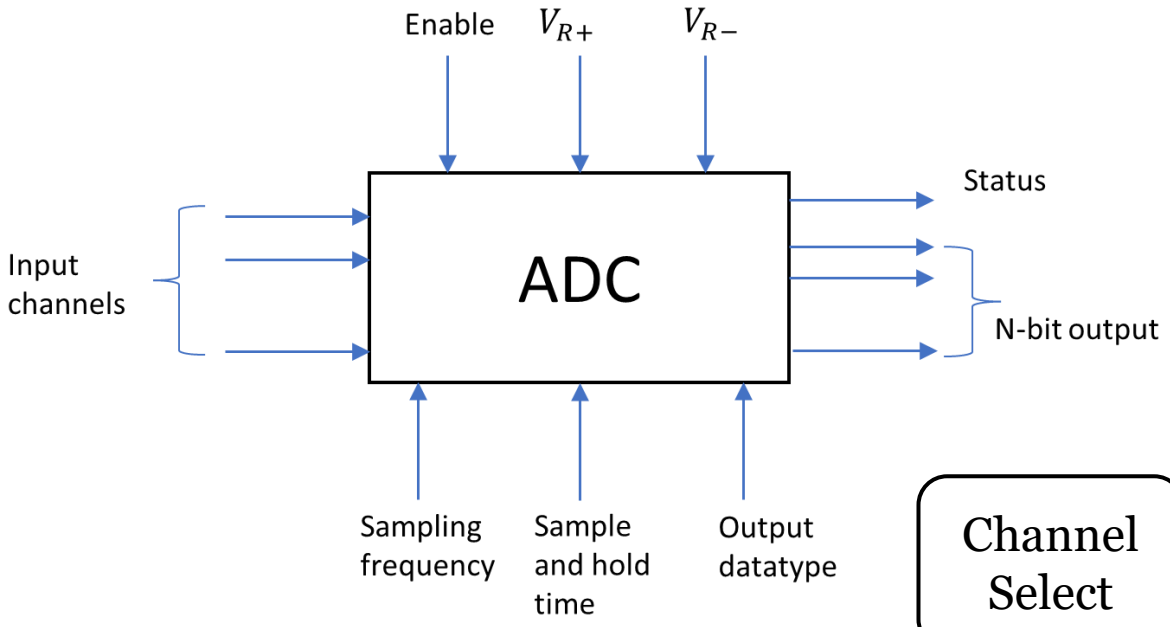


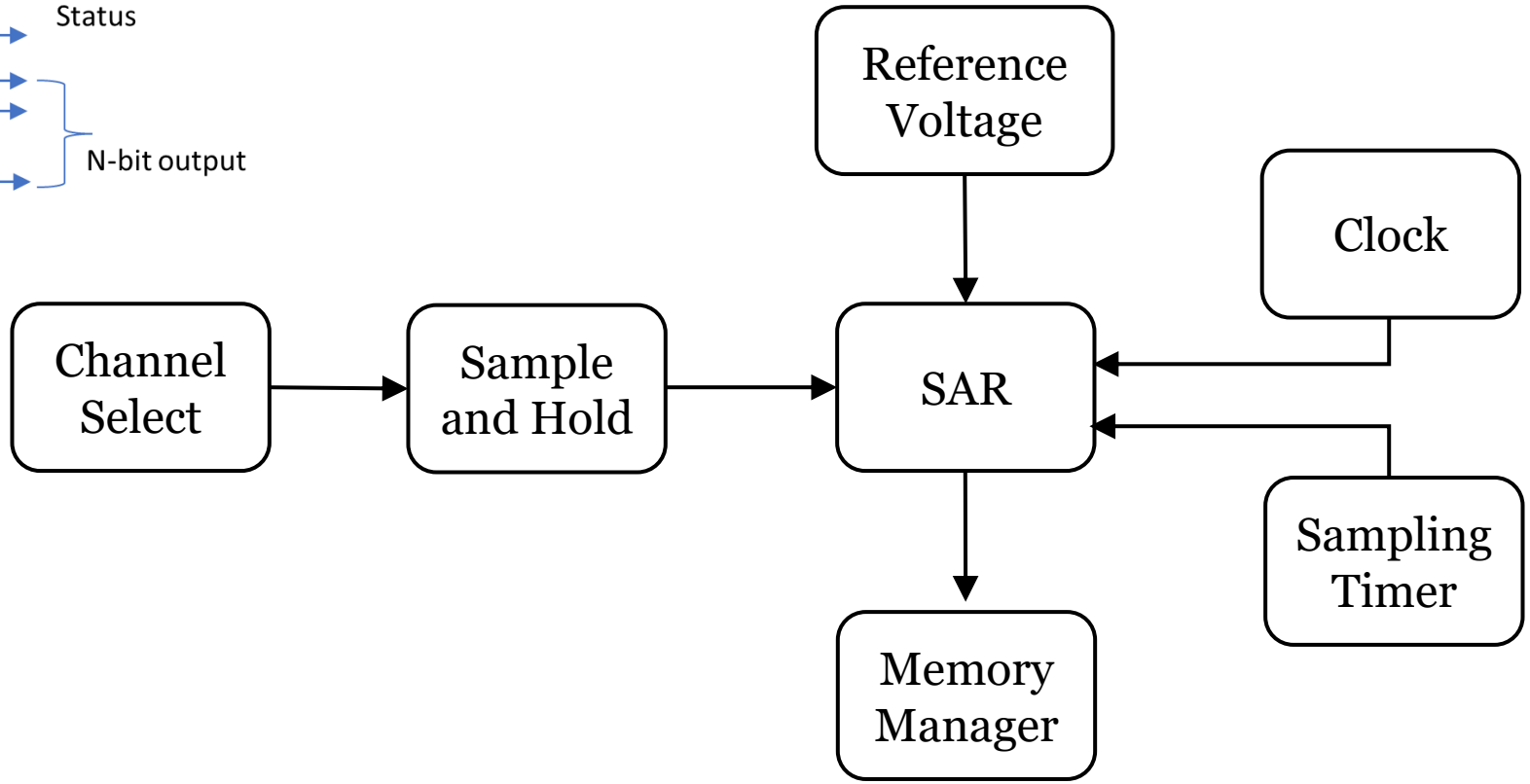
ADC Ports


Enable
V_{R+}
V_{R-}
Input channels
ADC
Status
N-bit output
Sampling frequency
Sample and hold time
Output datatype

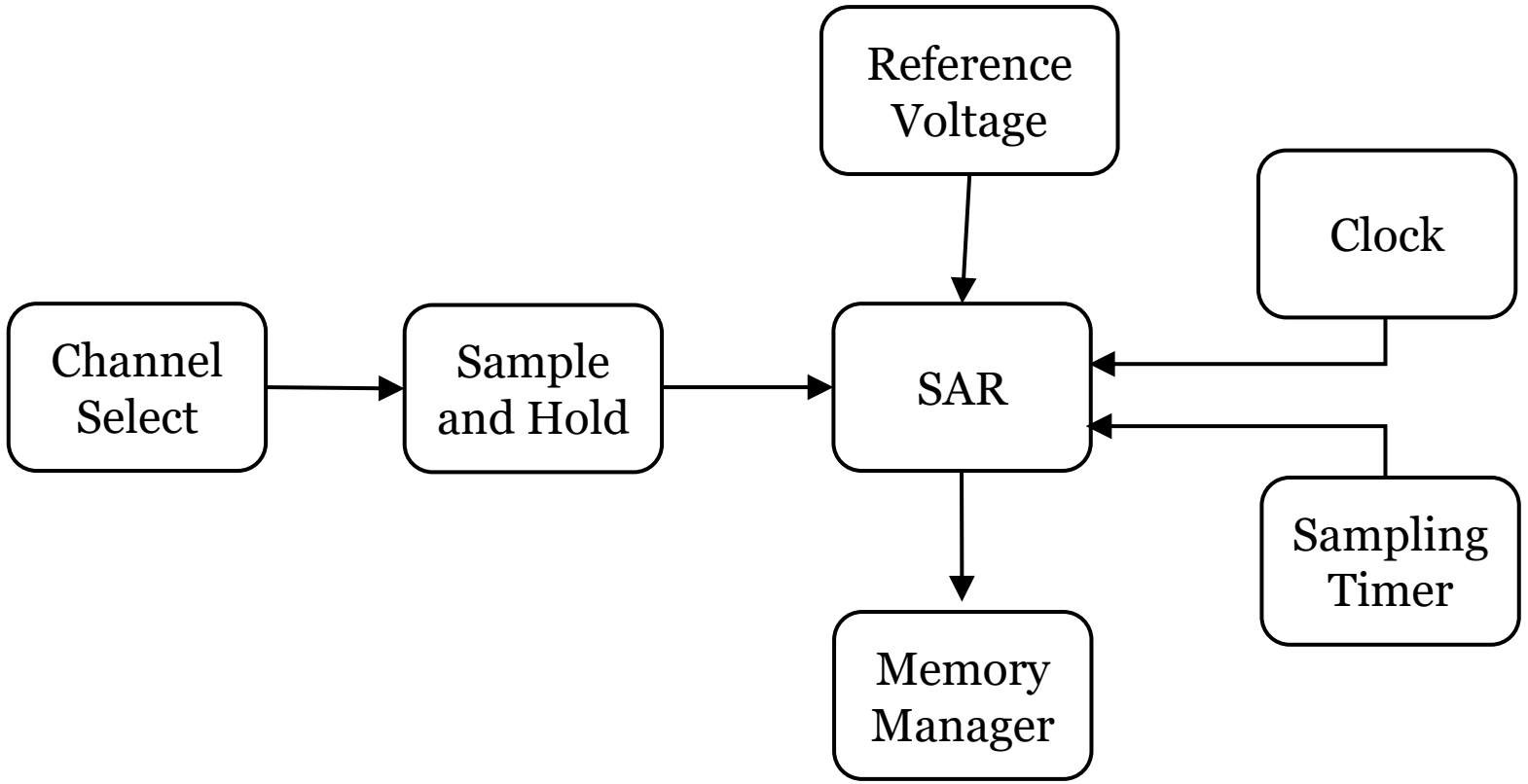
MSP 430 ADC Block

Basic Blocks of ADC in MSP 430



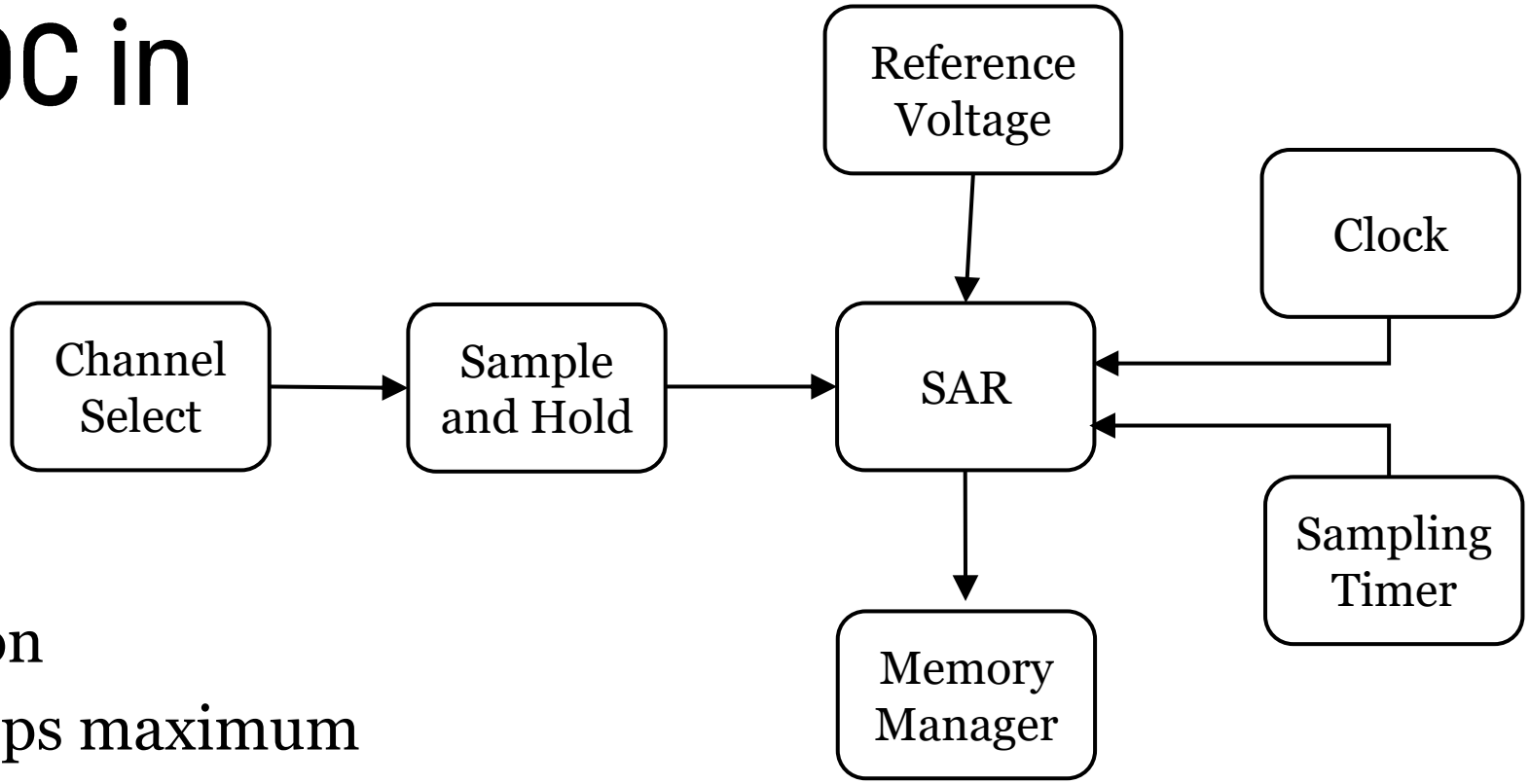


Basic Blocks of ADC in MSP 430



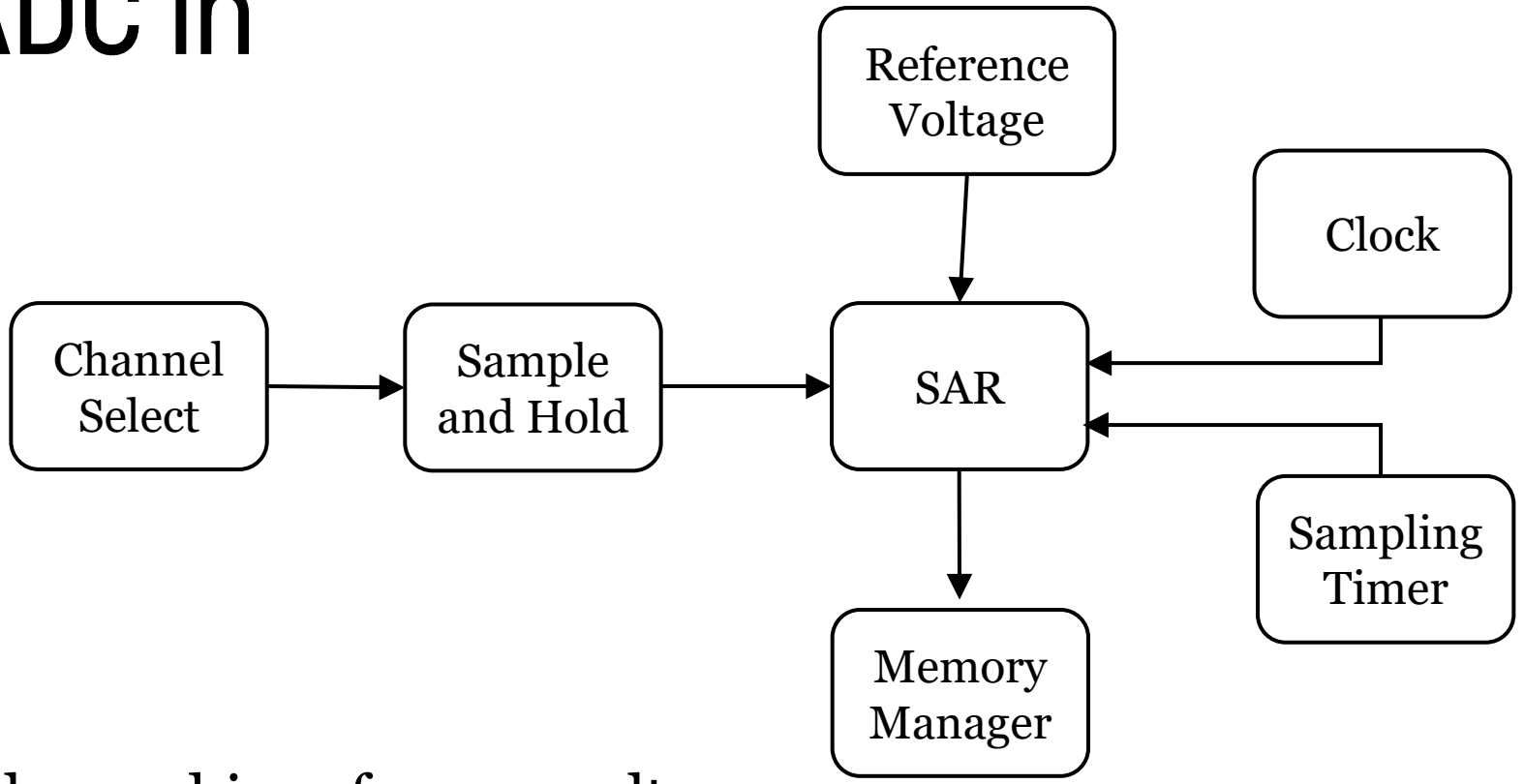
- ADC10 module
  - 10-bit SAR core
  - Reference generator
  - Data transfer controller (DTC)

Features of ADC in MSP 430



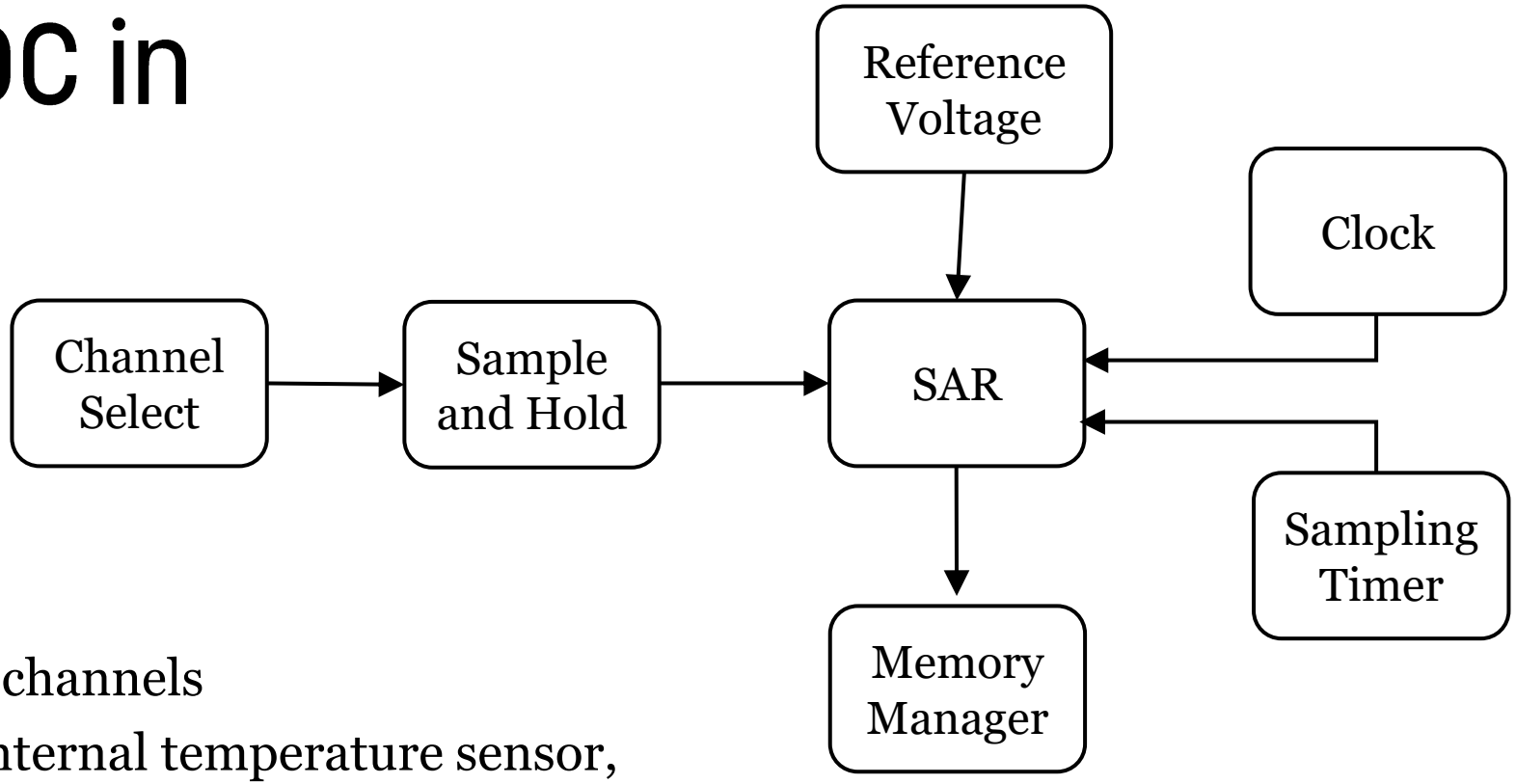
- Sampling and conversion
  - Greater than 200-ksps maximum conversion rate
  - Sample-and-hold with programmable sample periods
  - Conversion initiation by software or Timer_A

Features of ADC in MSP 430



- Reference volatges
  - Software selectable on-chip reference voltage generation (1.5 V or 2.5 V)
  - Software selectable internal or external reference

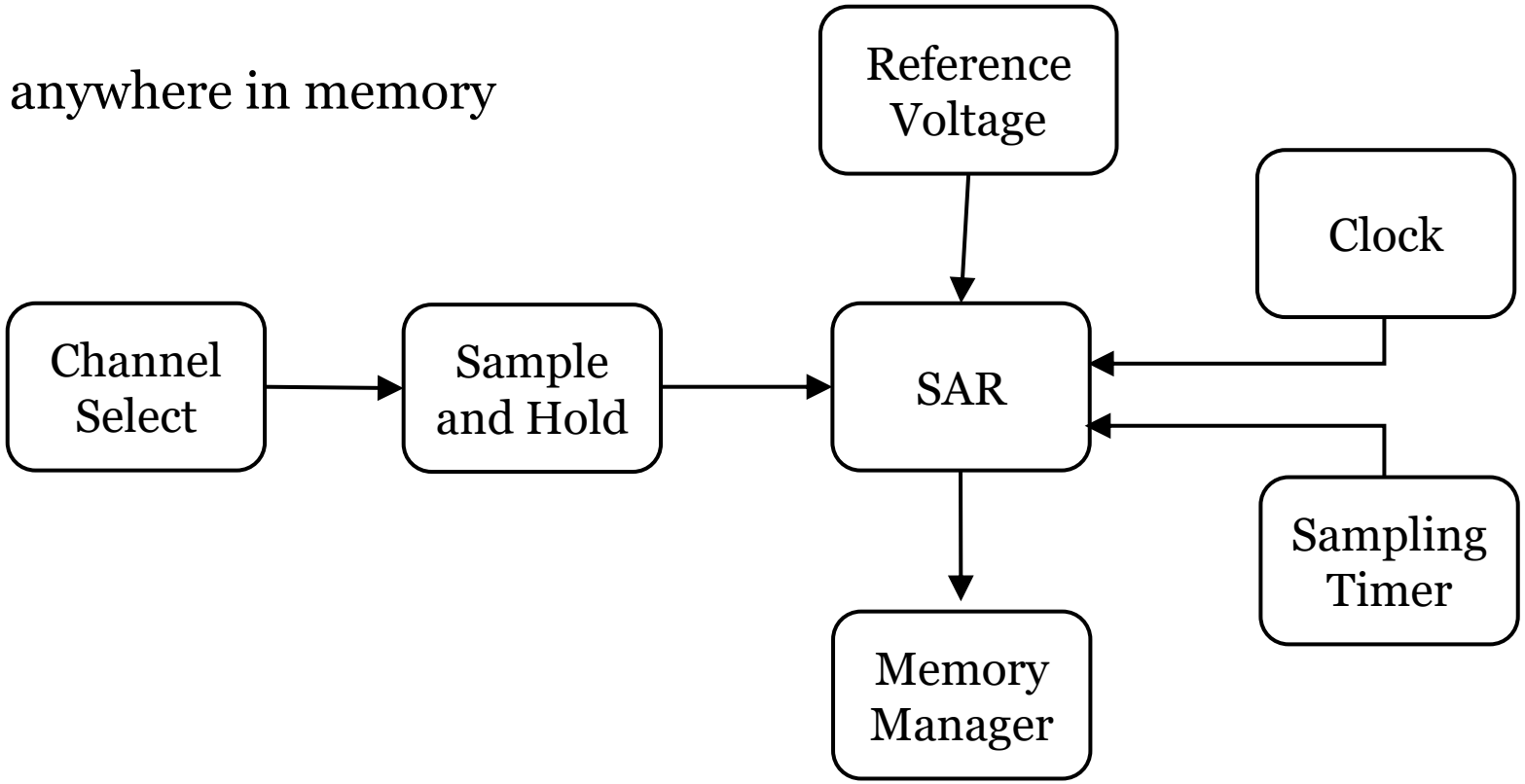
Features of ADC in MSP 430



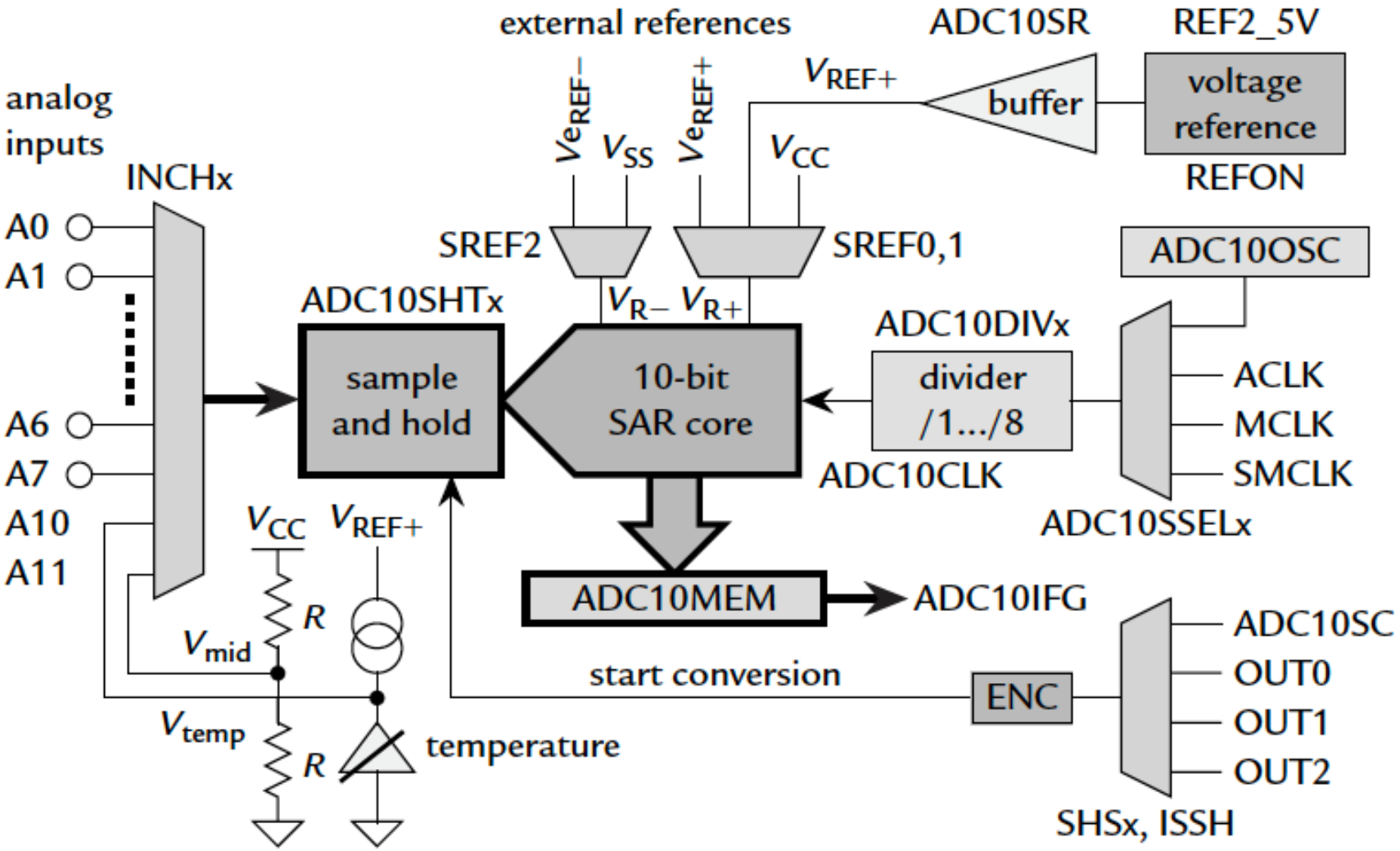
- Input channels
  - Up to eight external input channels
  - Conversion channels for internal temperature sensor, VCC, and external references
  - Selectable conversion clock source
- Conversion modes
  - Single-channel, repeated single-channel, sequence, and repeated sequence conversion modes

Features of ADC in MSP 430

- Data transfer controller for automatic storage of conversion results
- Data samples can be stored anywhere in memory without CPU intervention



MSP 430 ADC Block Diagram



ADC Core

- Converts analog input to 10-bit digital representation
- Stores result in ADC10MEM register
- Conversion formula for ADC result in straight binary format

$$N_{ADC} = 1023 \frac{V_{in} - V_{R-}}{V_{R+} - V_{R-}}$$

  - Digital output is full scale 0 to 3FFh
  - Conversion results may be in straight binary or 2s-complement format
- Uses two programmable/selectable voltage levels
  - V_{R+} Upper limit of conversion
  - V_{R-} Lower limit of conversion

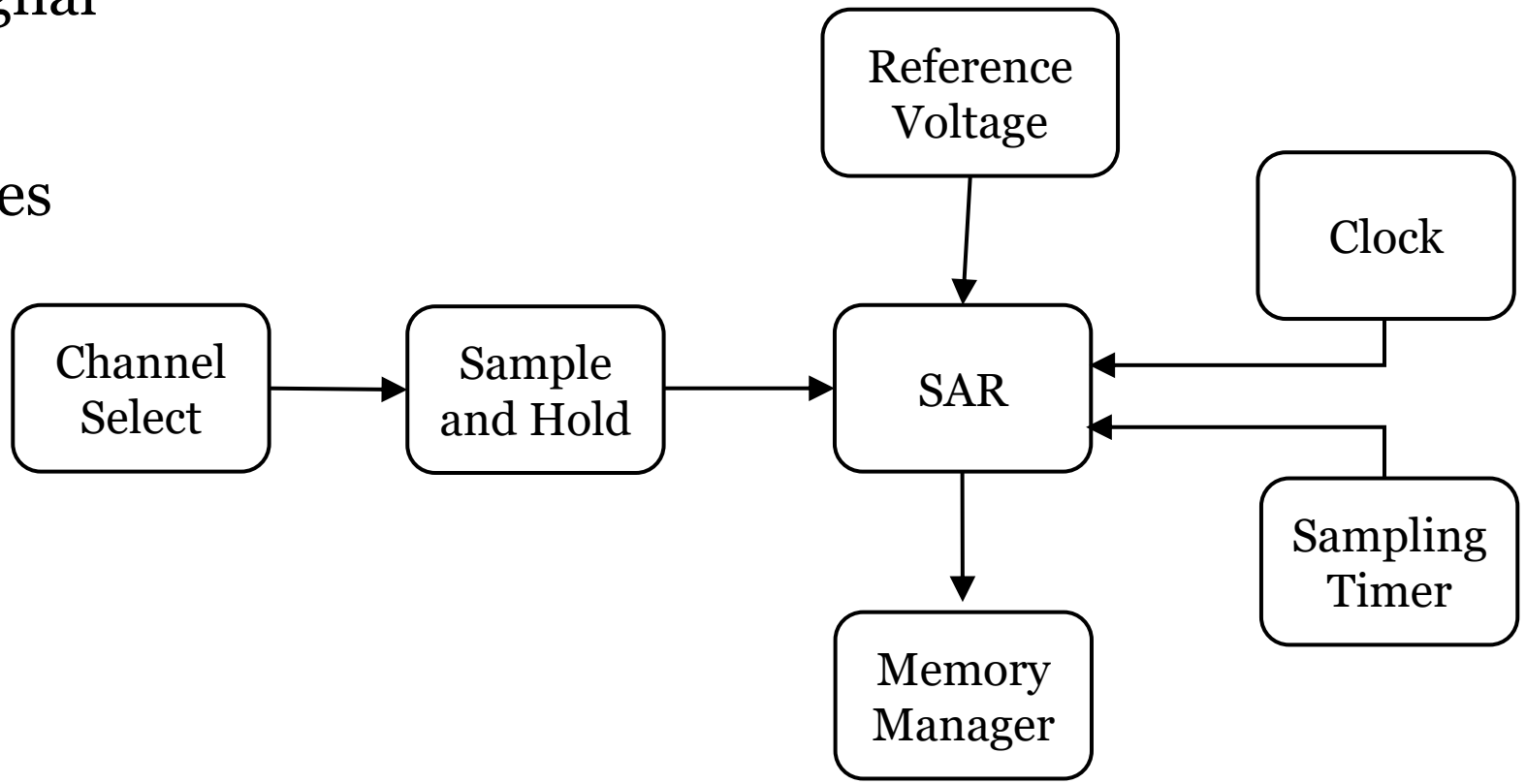
ADC10 Core Configuration

ADC10 Core Configuration

- ADC10 core is configured by two control registers
  - ADC10CTL0
  - ADC10CTL1
- ADC10ON bit and enable conversion ENC bits
  - Core is enabled with the ADC10ON bit
  - ADC10 control bits can be modified only when enable conversion bit ENC = 0
- ENC must be set to 1 before any conversion can take place

ADC Configuration: Control Register 0: ADC10CTL0

- Input port for analog signal
- Reference voltages
- Reference voltage sources
- Number format
- Sampling time
- Sampling clock source
- Sample and hold time
- Interrupts



ADC Control Register 0: ADC10CTL0

15	14	13	12	11	10	9	8
SREFx			ADC10SHTx		ADC10SR	REFOUT	REFBURST
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC10ON	ADC10IE	ADC10IFG	ENC	ADC10SC
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Can be modified only when ENC = 0

SREFx	Bits 15-13	Select reference
	000	VR+ = VCC and VR- = VSS
	001	VR+ = VREF+ and VR- = VSS
	010	VR+ = VeREF+ and VR- = VSS.
	011	VR+ = Buffered VeREF+ and VR- = VSS.

ADC Control Register 0: ADC10CTL0

15	14	13	12	11	10	9	8
SREFx			ADC10SHTx		ADC10SR	REFOUT	REFBURST
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC10ON	ADC10IE	ADC10IFG	ENC	ADC10SC
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Can be modified only when ENC = 0

SREFx	Bits 15-13	Select reference
	100	VR+ = VCC and VR- = VREF-/ VeREF-
	101	VR+ = VREF+ and VR- = VREF-/ VeREF-
	110	VR+ = VeREF+ and VR- = VREF-/ VeREF-
	111	VR+ = Buffered VeREF+ and VR- = VREF-/ VeREF-

ADC Control Register 0: ADC10CTL0

15	14	13	12	11	10	9	8
SREFx			ADC10SHTx		ADC10SR	REFOUT	REFBURST
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC10ON	ADC10IE	ADC10IFG	ENC	ADC10SC
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Can be modified only when ENC = 0

ADC10SHTx	Bits 12-11	ADC10 sample-and-hold time
	00	4 × ADC10CLKs
	01	8 × ADC10CLKs
	10	16 × ADC10CLKs
	11	64 × ADC10CLKs

ADC Control Register 0: ADC10CTL0

15	14	13	12	11	10	9	8
SREFx			ADC10SHTx		ADC10SR	REFOUT	REFBURST
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC10ON	ADC10IE	ADC10IFG	ENC	ADC10SC
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Can be modified only when ENC = 0

ADC10SRs	Bits 10	ADC10 sampling rate
	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

ADC Control Register 0: ADC10CTL0

15	14	13	12	11	10	9	8
SREFx			ADC10SHTx		ADC10SR	REFOUT	REFBURST
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC10ON	ADC10IE	ADC10IFG	ENC	ADC10SC
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Can be modified only when ENC = 0

REFBURST	Bits 8	Reference burst
	0	Reference buffer on continuously
	1	Reference buffer on only during sample-and-conversion

ADC Control Register 0: ADC10CTL0

15	14	13	12	11	10	9	8
SREFx			ADC10SHTx		ADC10SR	REFOUT	REFBURST
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC10ON	ADC10IE	ADC10IFG	ENC	ADC10SC
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Can be modified only when ENC = 0

MSC	Bit 7	Multiple sample and conversion. Valid only for sequence or repeated modes.
	0	Sampling requires a rising edge of the SHI signal to trigger each sample-and-conversion.
	1	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

ADC Control Register 0: ADC10CTL0

15	14	13	12	11	10	9	8
SREFx			ADC10SHTx		ADC10SR	REFOUT	REFBURST
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC10ON	ADC10IE	ADC10IFG	ENC	ADC10SC
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Can be modified only when ENC = 0

REF2_5V	Bits 6	Reference-generator voltage. REFON must also be set.
	0	1.5 V
	1	2.5 V
REFON	**Bit 5**	**Reference generator on**
	0	Reference off
	1	Reference on

ADC Control Register 0: ADC10CTL0

15	14	13	12	11	10	9	8
SREFx			ADC10SHTx		ADC10SR	REFOUT	REFBURST
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC10ON	ADC10IE	ADC10IFG	ENC	ADC10SC
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Can be modified only when ENC = 0

ADC10ON	Bits 4	ADC10 on
	0	ADC10 off
	1	ADC10 on
ADC10IE	**Bit 3**	**ADC10 interrupt enable**
	0	Interrupt disabled
	1	Interrupt enabled

ADC Control Register 0: ADC10CTL0

15	14	13	12	11	10	9	8
SREFx			ADC10SHTx		ADC10SR	REFOUT	REFBURST
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC10ON	ADC10IE	ADC10IFG	ENC	ADC10SC
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Can be modified only when ENC = 0

ADC10IFG	Bit 2	ADC10 interrupt flag.
	0	No interrupt pending
	1	Interrupt pending

- **ADC10IFG** bit is set if ADC10MEM is loaded with a conversion result.
- **ADC10IFG** 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.

ADC Control Register 0: ADC10CTL0

15	14	13	12	11	10	9	8
SREFx			ADC10SHTx		ADC10SR	REFOUT	REFBURST
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC10ON	ADC10IE	ADC10IFG	ENC	ADC10SC
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Can be modified only when ENC = 0

ENC	Bit 1	Enable conversion
	0	ADC10 disabled
	1	ADC10 enabled
ADC10SC	**Bit 0**	**Start conversion. Software-controlled sample-and-conversion start.**
	0	No sample-and-conversion start
	1	Start sample-and-conversion

ADC10SC and ENC may be set together with one instruction. ADC10SC is reset automatically.

ADC Control Register 0: ADC10CTL0

15	14	13	12	11	10	9	8
SREFx			ADC10SHTx		ADC10SR	REFOUT	REFBURST
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC10ON	ADC10IE	ADC10IFG	ENC	ADC10SC
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

Can be modified only when ENC = 0

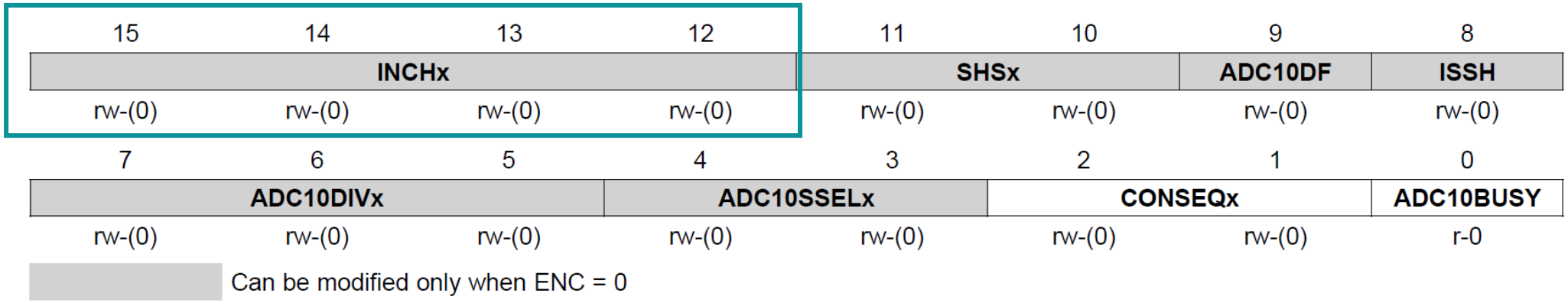
ENC	Bit 1	Enable conversion
	0	ADC10 disabled
	1	ADC10 enabled
ADC10SC	**Bit 0**	**Start conversion. Software-controlled sample-and-conversion start.**
	0	No sample-and-conversion start
	1	Start sample-and-conversion

ADC10SC and ENC may be set together with one instruction. ADC10SC is reset automatically.

ADC10 Core Configuration

- ADC10ON bit and enable conversion ENC bits
  - Core is enabled with ADC10ON bit
  - ADC10 control bits can be modified only when enable conversion bit ENC = 0
- ENC must be set to 1 before any conversion can take place

ADC Control Register 1: ADC10CTL1

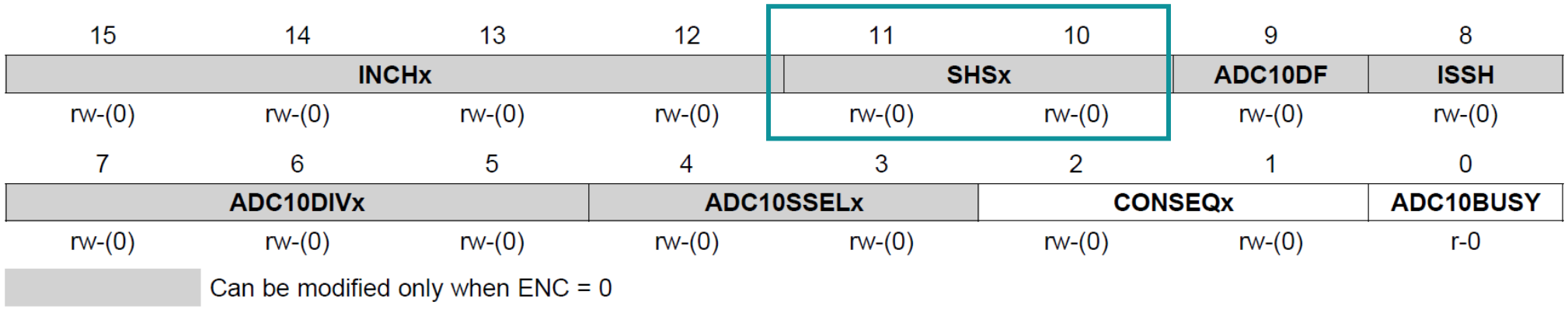


INCHx	Bits 15-12	Input channel select
	0000	A0
	0001	A1
	0010	A2
	.	.
	0111	A7

Bits 15-12	Input channel select
1000	VeREF+
1001	VREF-/VeREF-
1010	Temperature sensor
1011	(VCC - VSS) / 2

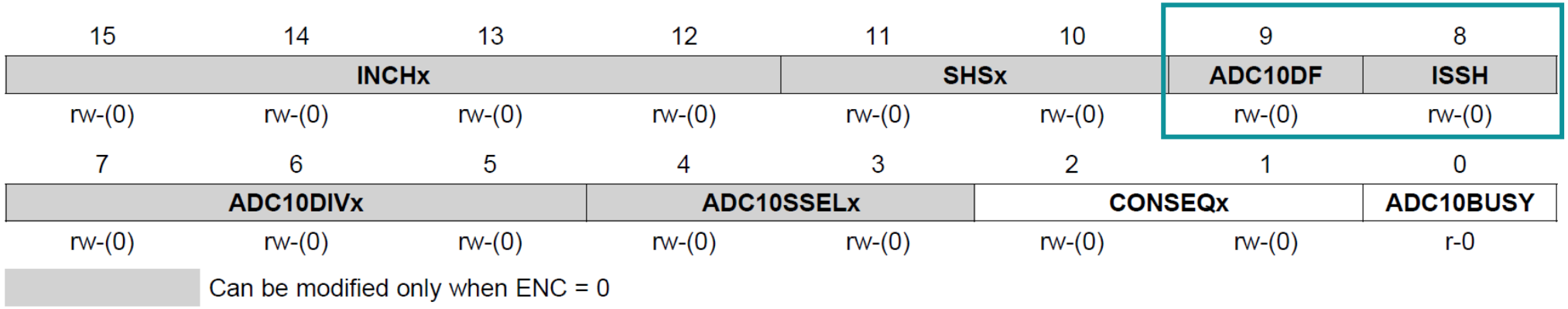
Bits 15-12	Input channel select
1100	(VCC - VSS) / 2, A12 on MSP430F22xx
1101	(VCC - VSS) / 2, A13 on MSP430F22xx
1110	(VCC - VSS) / 2, A14 on MSP430F22xx
1111	(VCC - VSS) / 2, A15 on MSP430F22xx

ADC Control Register 1: ADC10CTL1



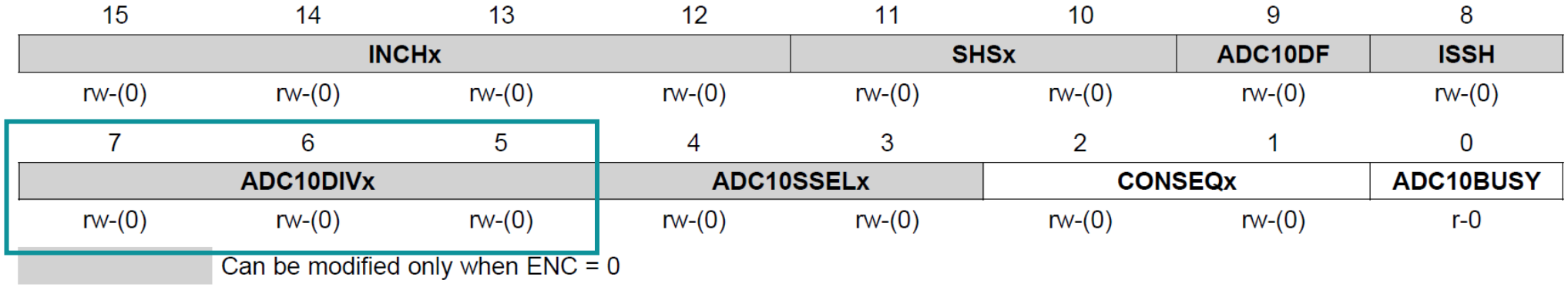
SHSx	Bits 11-10	Sample-and-hold source select
	00	ADC10SC bit
	01	Timer_A.OUT1
	10	Timer_A.OUT0
	11	Timer_A.OUT2

ADC Control Register 1: ADC10CTL1



ADC10DF	Bit 9	ADC10 data format
	0	Straight binary
	1	2s complement
ISSH	**Bit 8**	**Invert signal sample-and-hold**
	0	sample-input signal is not inverted
	1	sample-input signal is inverted

ADC Control Register 1: ADC10CTL1



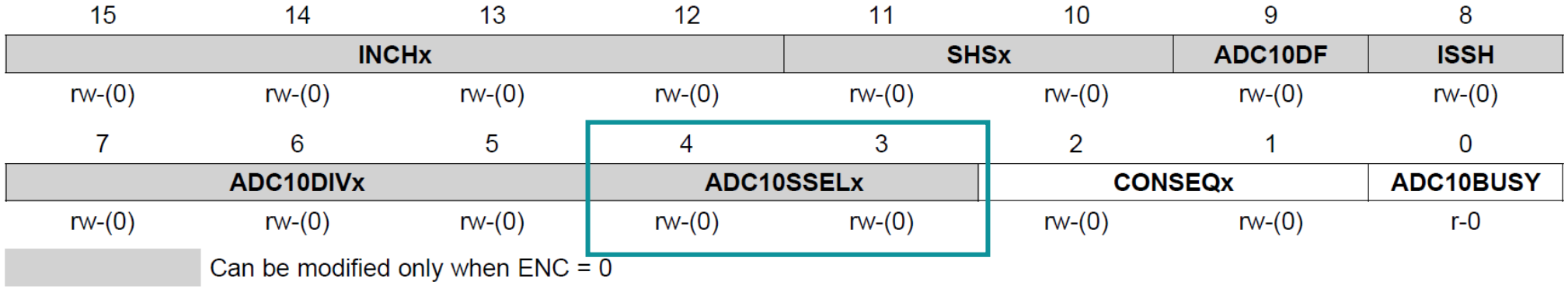
15	14	13	12	11	10	9	8
INCHx				SHSx		ADC10DF	ISSH
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

7	6	5	4	3	2	1	0
ADC10DIVx			ADC10SSELx		CONSEQx		ADC10BUSY
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r-0

Can be modified only when ENC = 0

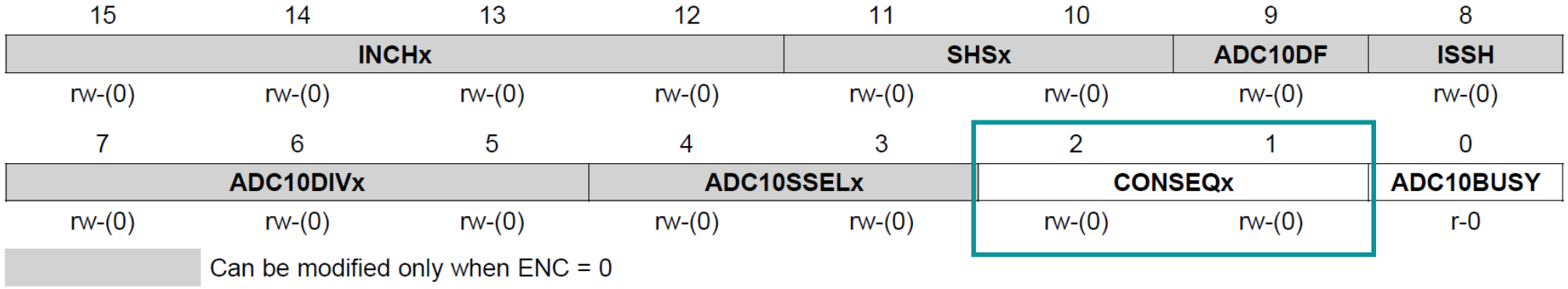
ADC10DIVx	Bits 7-5	ADC10 clock divider
	000	/1
	001	/2
	010	/3
	.	.
	111	/8

ADC Control Register 1: ADC10CTL1



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

ADC Control Register 1: ADC10CTL1



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

ADC Control Register 1: ADC10CTL1

15	14	13	12	11	10	9	8
INCHx				SHSx		ADC10DF	ISSH
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
ADC10DIVx			ADC10SSELx		CONSEQx		ADC10BUSY
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r-0

Can be modified only when ENC = 0

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

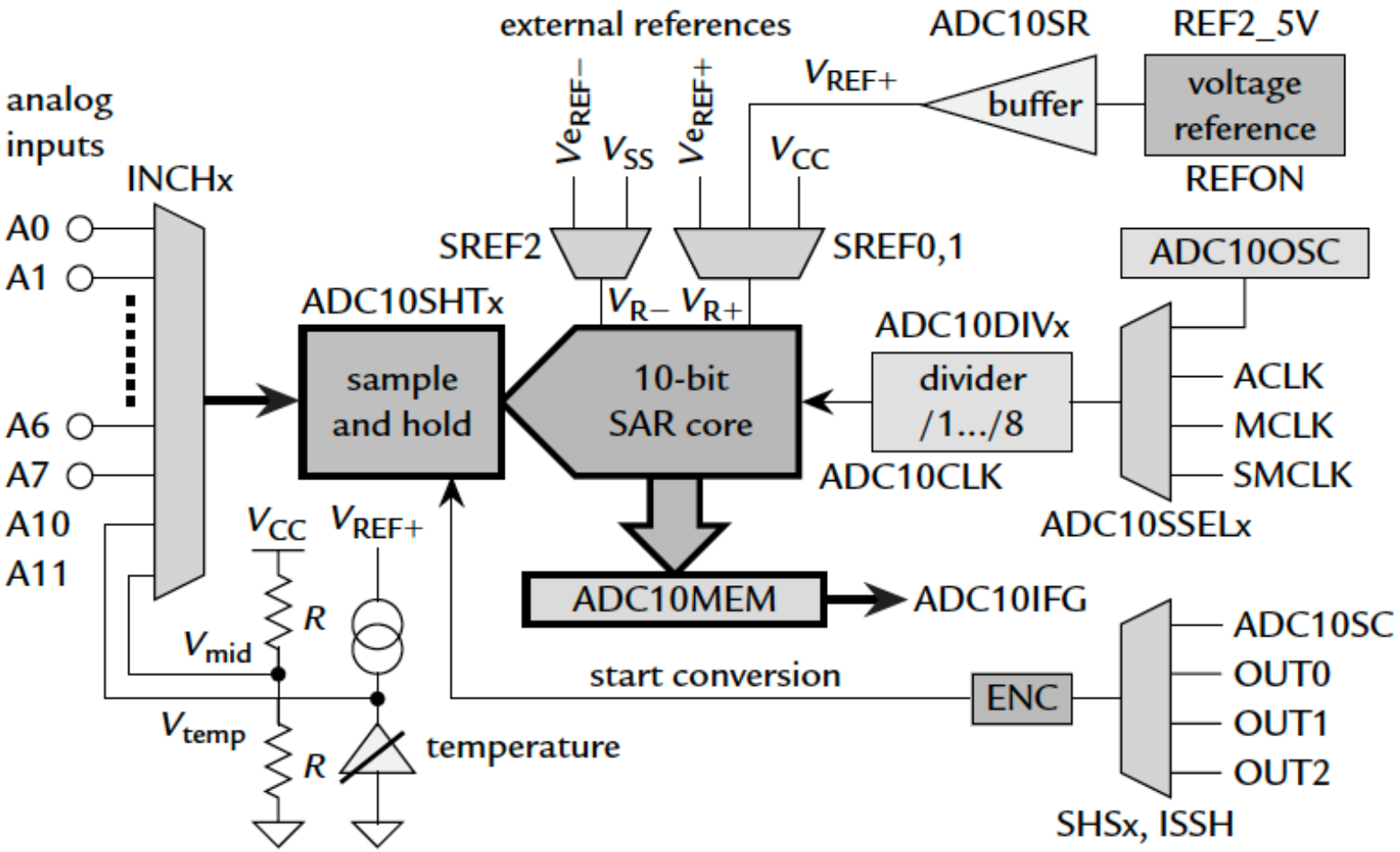
Conversion Clock Selection

- ADC10CLK is used both as conversion clock and to generate sampling period.
- ADC10 source clock is selected using ADC10SSELx bits
  - Can be divided from 1 to 8 using ADC10DIVx bits
- Possible ADC10CLK sources
  - SMCLK, MCLK, ACLK
  - Internal oscillator ADC10OSC
- Must ensure that clock for ADC10CLK remains active until end of conversion
  - If clock is removed during a conversion, ADC operation is incomplete and result is invalid

ADC10 Inputs and Multiplexer

- Eight external and four internal analog signals are selected as channel for conversion by analog input multiplexer
- ADC10 external inputs Ax, VeREF+,and VREF- share terminals with general purpose I/O ports
- Parasitic current can flow from VCC to GND when analog signals are applied to digital CMOS gates
  - Disabling port pin buffer can eliminate parasitic current flow
  - ADC10AEx bits can be used disable the port pin input and output buffers

MSP 430 ADC Block Diagram



Port Selection

- ADC10AEx bits can be used disable the port pin input and output buffers

22.3.3 ADC10AE0, Analog (Input) Enable Control Register 0

7	6	5	4	3	2	1	0
ADC10AE0x							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

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. The analog enable bit of not implemented channels should not be programmed to 1. 0 Analog input disabled 1 Analog input enabled

```
ADC10CTL1 = INCH_1;    // input A1, constant INCH_1 = 0x1000
ADC10AE0 = 0x02;       // P1.1 ADC function selected
```

Port Schematics

Table 16. Port P1 (P1.0 to P1.2) Pin Functions

PIN NAME (P1.x)	x	FUNCTION	CONTROL BITS AND SIGNALS[1]				
			P1DIR.x	P1SEL.x	P1SEL2.x	ADC10AE.x INCH.x=1[2]	CAPD.y
P1.0/	0	P1.x (I/O)	I: 0; O: 1	0	0	0	0
TA0CLK/		TA0.TACLK	0	1	0	0	0
ACLK/		ACLK	1	1	0	0	0
A0[2]/		A0	X	X	X	1 (y = 0)	0
CA0/		CA0	X	X	X	0	1 (y = 0)
Pin Osc		Capacitive sensing	X	0	1	0	0
P1.1/	1	P1.x (I/O)	I: 0; O: 1	0	0	0	0
TA0.0/		TA0.0	1	1	0	0	0
		TA0.CCI0A	0	1	0	0	0
UCA0RXD/		UCA0RXD	from USCI	1	1	0	0
UCA0SOMI/		UCA0SOMI	from USCI	1	1	0	0
A1[2]/		A1	X	X	X	1 (y = 1)	0
CA1/		CA1	X	X	X	0	1 (y = 1)
Pin Osc		Capacitive sensing	X	0	1	0	0
P1.2/	2	P1.x (I/O)	I: 0; O: 1	0	0	0	0
TA0.1/		TA0.1	1	1	0	0	0
		TA0.CCI1A	0	1	0	0	0
UCA0TXD/		UCA0TXD	from USCI	1	1	0	0
UCA0SIMO/		UCA0SIMO	from USCI	1	1	0	0
A2[2]/		A2	X	X	X	1 (y = 2)	0
CA2/		CA2	X	X	X	0	1 (y = 2)
Pin Osc		Capacitive sensing	X	0	1	0	0

(1) X = don't care
(2) MSP430G2x53 devices only

Port Schematics

Port Schematics in msp430g2553.pdf

Table 16. Port P1 (P1.0 to P1.2) Pin Functions

PIN NAME (P1.x)	x	FUNCTION	CONTROL BITS AND SIGNALS[1]				
			P1DIR.x	P1SEL.x	P1SEL2.x	ADC10AE.x INCH.x=1[2]	CAPD.y
P1.0/	0	P1.x (I/O)	I: 0; O: 1	0	0	0	0
TA0CLK/		TA0.TACLK	0	1	0	0	0
ACLK/		ACLK	1	1	0	0	0
A0[2]/		A0	X	X	X	1 (y = 0)	0
CA0/		CA0	X	X	X	0	1 (y = 0)
Pin Osc		Capacitive sensing	X	0	1	0	0
P1.1/	1	P1.x (I/O)	I: 0; O: 1	0	0	0	0
TA0.0/		TA0.0	1	1	0	0	0
		TA0.CCI0A	0	1	0	0	0
UCA0RXD/		UCA0RXD	from USCI	1	1	0	0
UCA0SOMI/		UCA0SOMI	from USCI	1	1	0	0
A1[2]/		A1	X	X	X	1 (y = 1)	0
CA1/		CA1	X	X	X	0	1 (y = 1)
Pin Osc		Capacitive sensing	X	0	1	0	0
P1.2/	2	P1.x (I/O)	I: 0; O: 1	0	0	0	0
TA0.1/		TA0.1	1	1	0	0	0
		TA0.CCI1A	0	1	0	0	0
UCA0TXD/		UCA0TXD	from USCI	1	1	0	0
UCA0SIMO/		UCA0SIMO	from USCI	1	1	0	0
A2[2]/		A2	X	X	X	1 (y = 2)	0
CA2/		CA2	X	X	X	0	1 (y = 2)
Pin Osc		Capacitive sensing	X	0	1	0	0

(1) X = don't care
(2) MSP430G2x53 devices only

Reference Voltages

- ADC10 contains built-in voltage ref with two selectable voltage levels.
- Select between internal and external reference
- Setting REFON = 1 enables internal reference
  - When REF2_5V = 1, internal reference is 2.5 V
  - When REF2_5V = 0, reference is 1.5 V
- Internal reference voltage may be used internally (REFOUT = 0)
  - Externally on pin VREF+ when REFOUT = 1
  - REFOUT = 1 should only be used if the pins VREF+ and VREF- are available as device pins.

Reference Voltages

- External references
  - Supplied via pin A4 for VR+
  - Supplied via pin A3 for VR-
- When external source or VCC are used as reference, turn off internal reference to save power.

ADC Conversion Initiation

Sample and Conversion Timing

- ADC conversion is initiated with rising edge of sample input signal SHI
  - ENC must be set
- Source for SHI is selected with SHSx bits and sources are
  - ADC10SC bit
  - Timer_A Output Unit 1 TA0CCR1
  - Timer_A Output Unit 0 TA0CCR0
  - Timer_A Output Unit 2 TA0CCR2

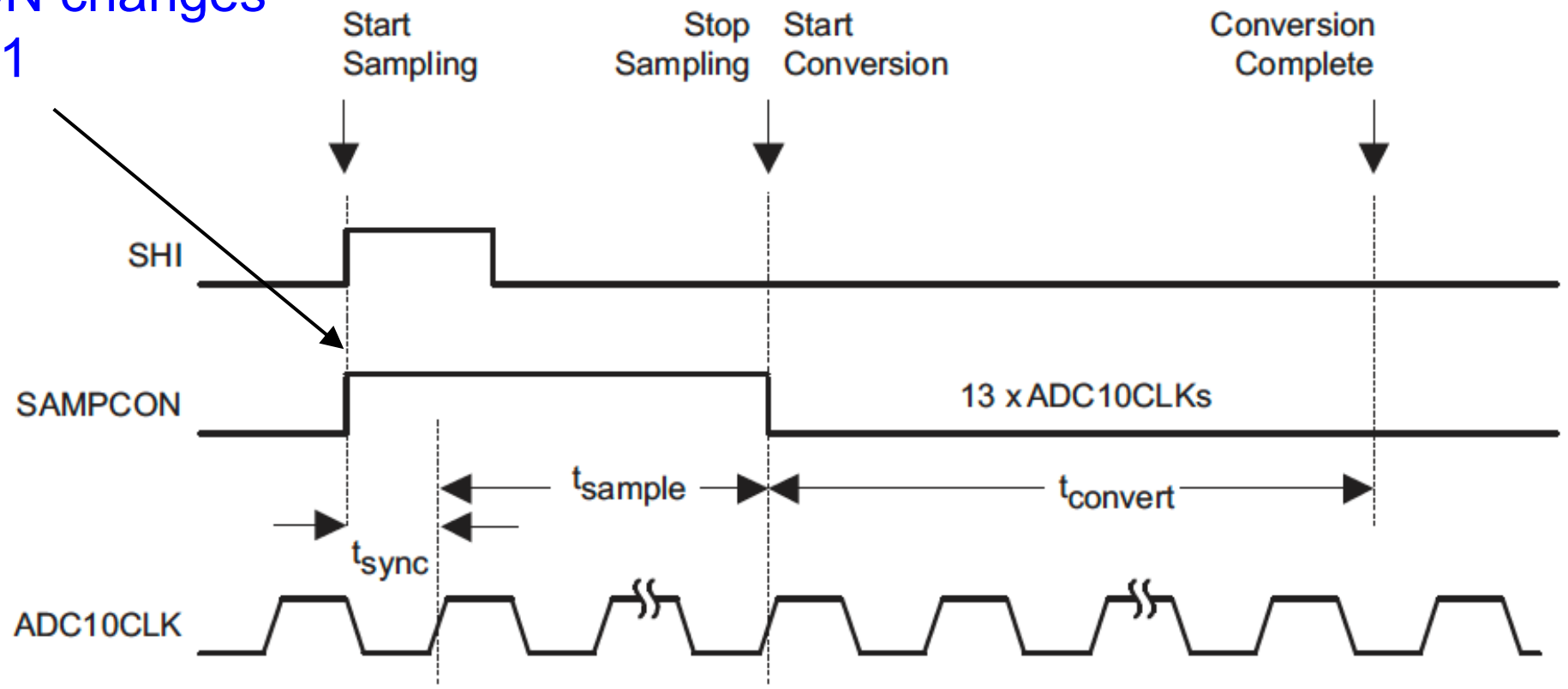
Timer triggers are from Timer0_A (e.g. TA0CCR0, TA0CCR1, TA0CCR2). Timer1_A cannot be used to trigger ADC.

- SHTx bits select sample period t_sample to be 4, 8, 16, or 64 ADC10CLK cycles.

Sample Timing

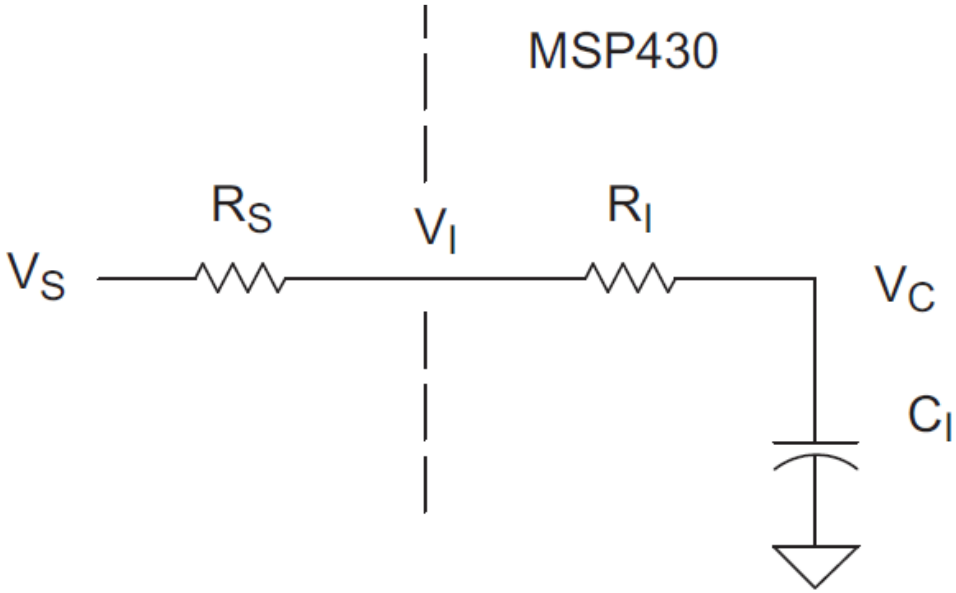
ENC = 1

SAMPCON changes from 0 to 1



Sample Timing Constraints

- Resistance of the source R_S and R_I affect t_{sample}
- Internal MUX-on input resistance R_I (2 kΩ max.) in series with capacitor C_I (27 pF max.) is seen by source
- Capacitor must be charged to within ½ LSB of source voltage for accurate conversion
- Minimum sampling time for a 10-bit conversion
- $t_{sample} > (R_S + R_I)C_I \ln(2^{11}) = (R_S + 2\ \text{k}\Omega) \times 7.625 \times 27\text{pF}$



VI = Input voltage at pin Ax
VS = External source voltage
RS = External source resistance
RI = Internal MUX-on input resistance
CI = Input capacitance
VC = Capacitance-charging voltage

Conversion Memory Register: ADC10MEM

- Data formats: Straight binary format for unsigned data
  - 10-bit conversion results are right justified
  - Bit 9 is the MSB. Bits 15-10 are always 0
- Data formats: 2's complement format for signed data
  - 10-bit conversion results are left-justified
  - Bit 15 is the MSB. Bits 5-0 are always 0
- Change data format:
  - ADC10DF in ADC10CTL1
  - straight binary: ADC10DF = 0
  - 2's complement: ADC10DF = 1

MSP 430 Conversion Modes

Conversion Modes

The ADC10 has four operating modes selected by the CONSEQx bits as discussed in Table 22-1.

Table 22-1. Conversion Mode Summary

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.

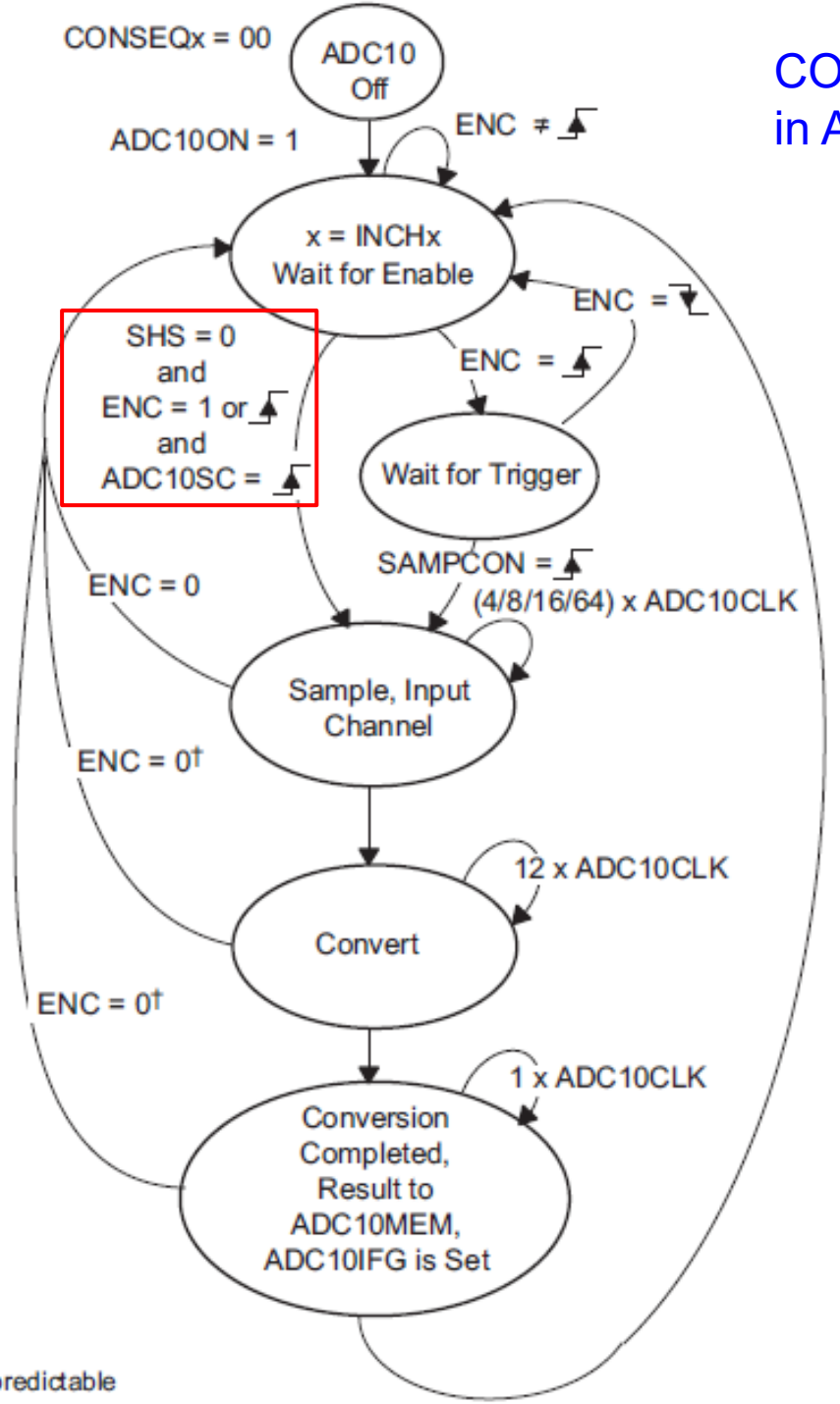
22.3.2 ADC10CTL1, ADC10 Control Register 1

15	14	13	12	11	10	9	8
INCHx				**SHSx**		**ADC10DF**	**ISSH**
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
ADC10DIVx			**ADC10SSELx**		**CONSEQx**		**ADC10BUSY**
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r-0

Can be modified only when ENC = 0

Single-Channel Single-Conversion Mode

- Single channel selected by INCHx is sampled and converted once.
- ADC result is written to ADC10MEM.
- When ADC10SC (Software-controlled conversion start) triggers a conversion, successive conversions can be triggered by ADC10SC bit.
- ADC10SC is reset automatically.
- When any other trigger source is used, ENC must be toggled between each conversion.



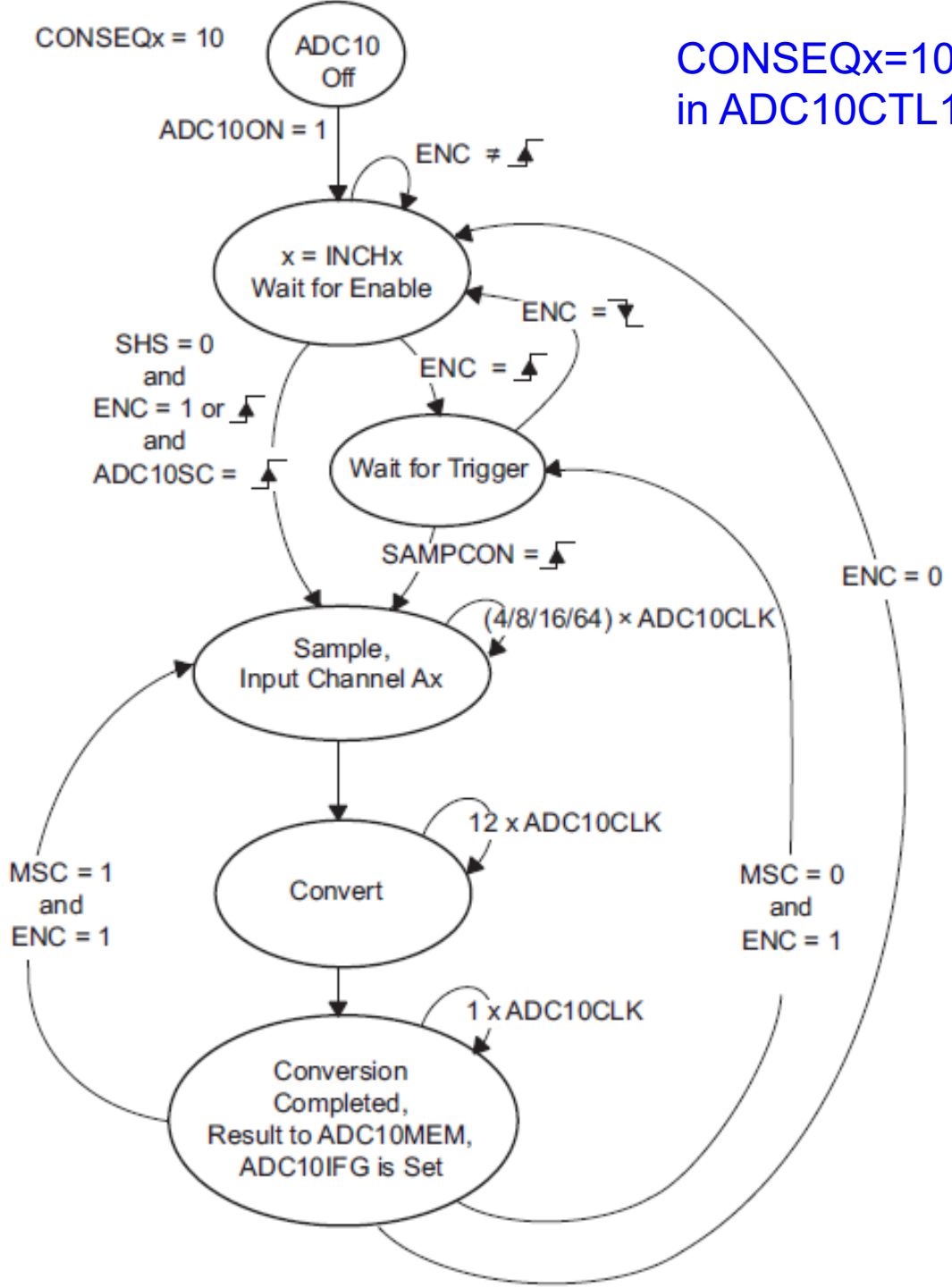
CONSEQx=00 in ADC10CTL1

x = input channel Ax
† Conversion result is unpredictable

Repeat-Single-Channel Mode

- A single channel selected by INCHx is sampled and converted continuously.
- Each ADC result is written to ADC10MEM.
- MSC: Multiple sample and conversion
  - Set in ADC10CTL0

Repeat if MSC=1 and ENC=1

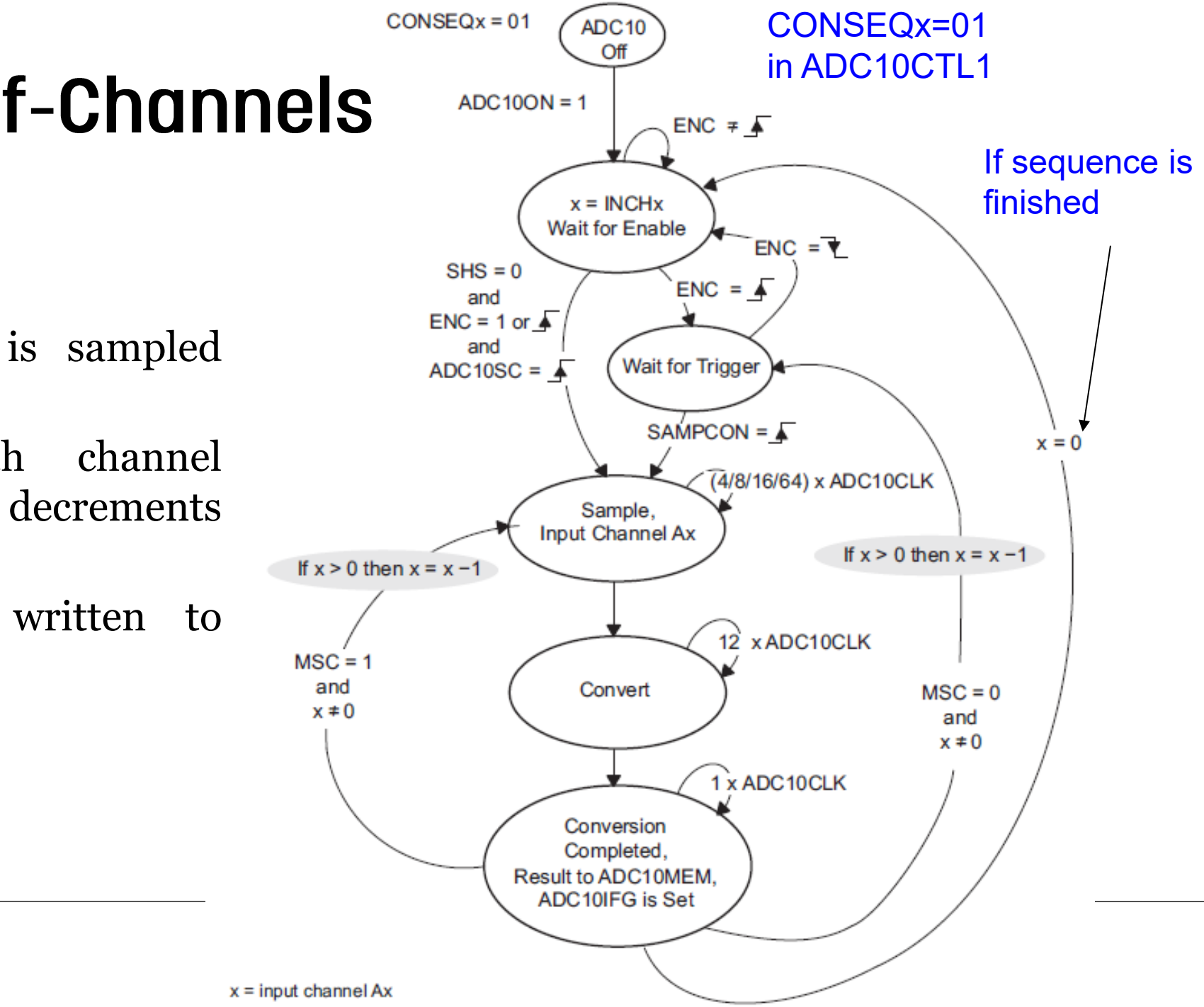


CONSEQx=10 in ADC10CTL1

x = input channel Ax

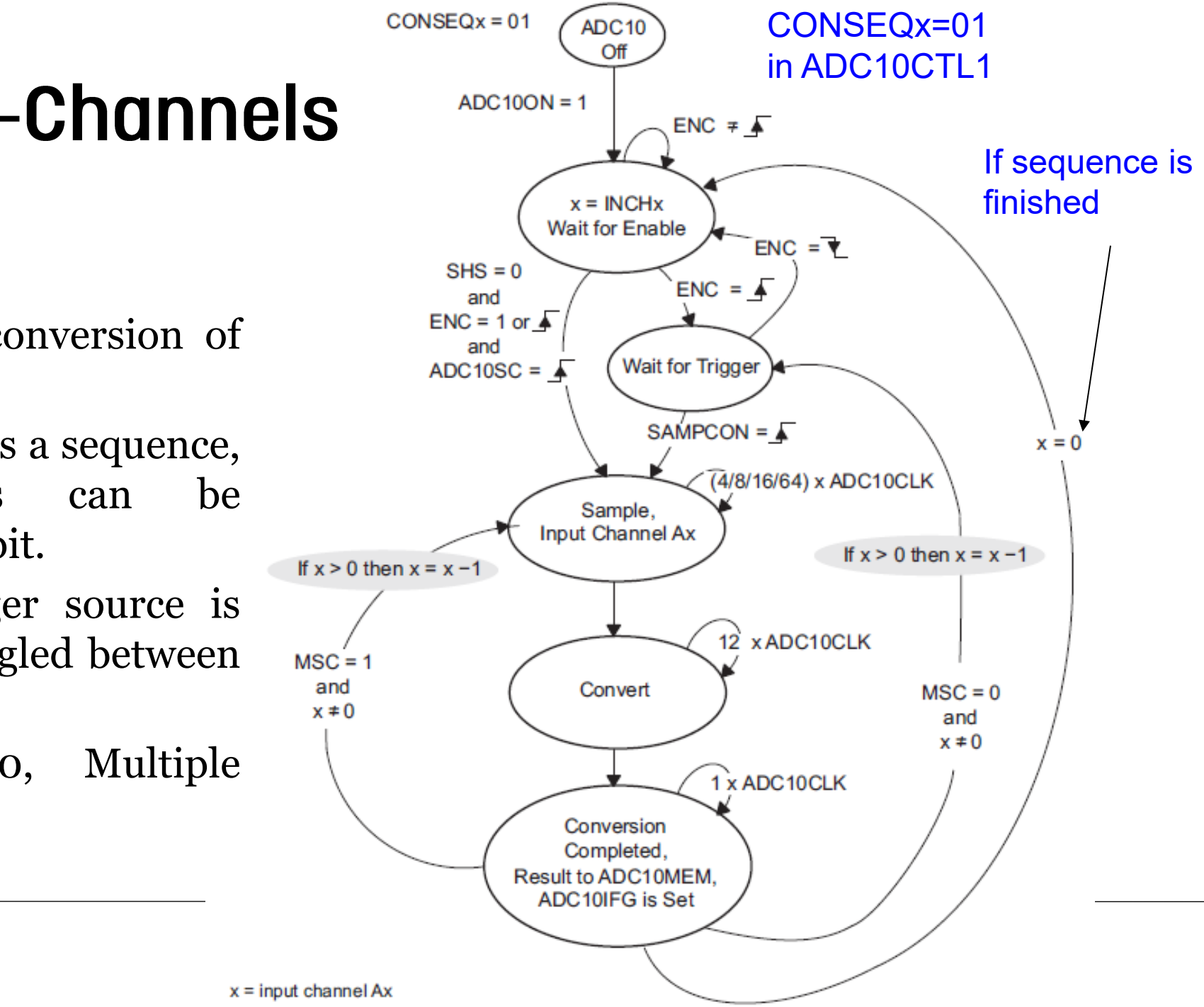
Sequence-of-Channels Mode

- Sequence of channels is sampled and converted once
- Sequence begins with channel selected by INCHx and decrements to channel A0.
- Each ADC result is written to ADC10MEM.



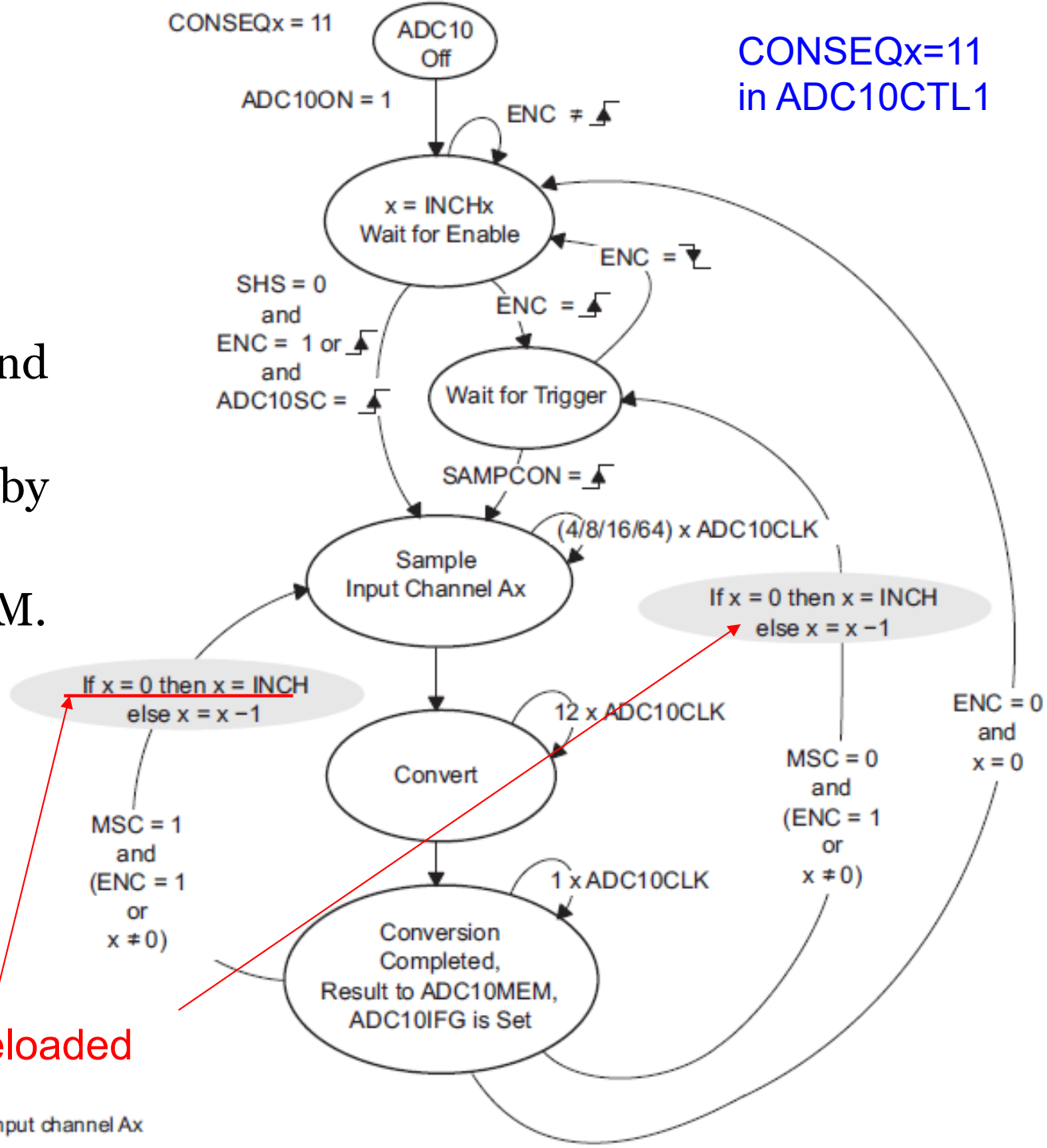
Sequence-of-Channels Mode

- Sequence stops after conversion of channel A0.
- When ADC10SC triggers a sequence, successive sequences can be triggered by ADC10SC bit.
- When any other trigger source is used, ENC must be toggled between each sequence.
- MSC in ADC10CTL0, Multiple sample and conversion



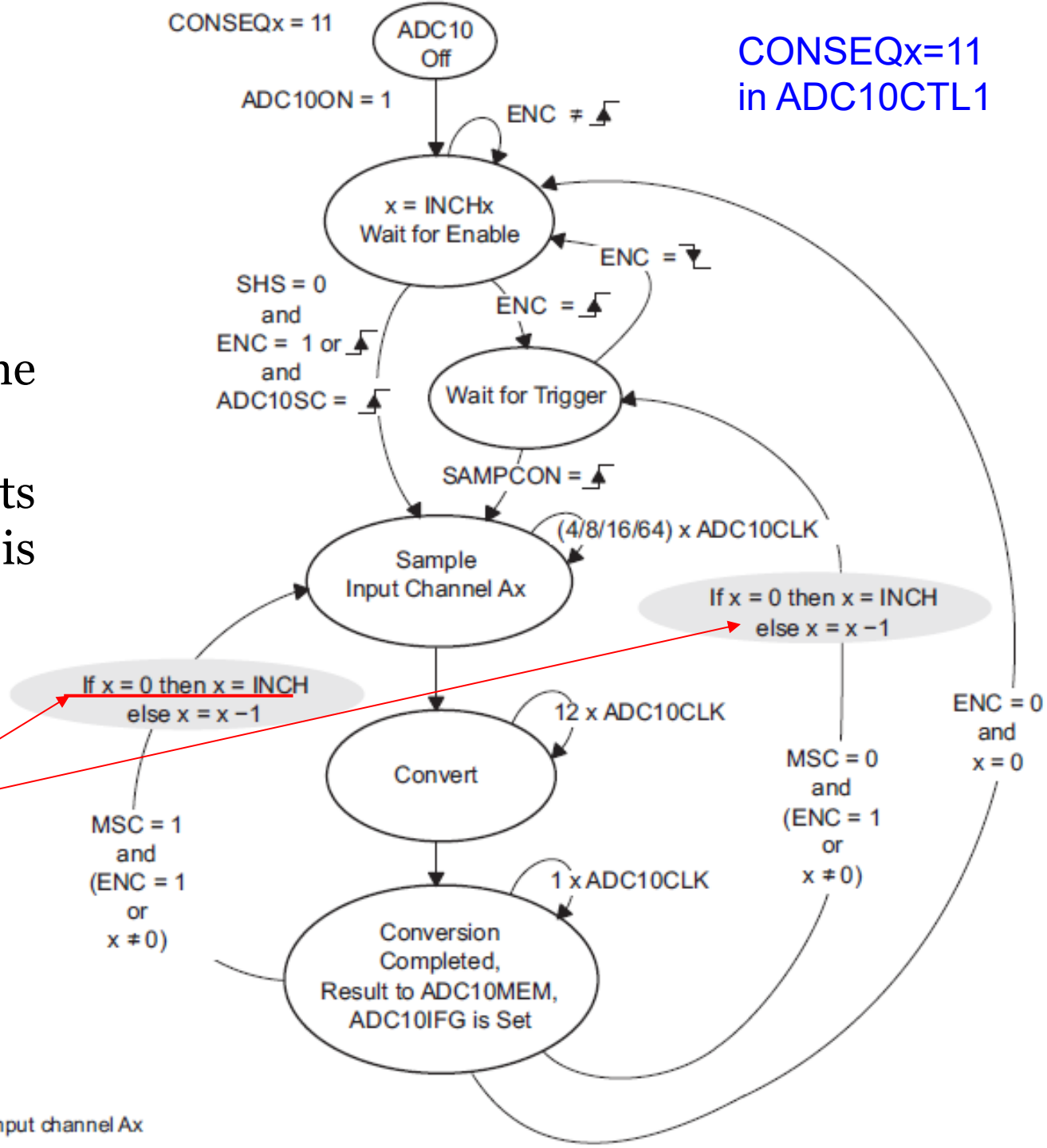
Repeat-Sequence-of-Channels Mode

- Sequence of channels is sampled and converted repeatedly.
- Sequence begins with channel selected by INCHx and decrements to channel A0.
- Each ADC result is written to ADC10MEM.



Repeat-Sequence-of-Channels Mode

- After conversion of channel Ao, the sequence ends.
- If MSC = 0, next trigger signal re-starts the sequence; if MSC = 1, sequence is re-started automatically.



CONSEQx=11 in ADC10CTL1

INCH is reloaded into x.

Stopping Conversions

- Stopping ADC10 depends on mode of operation
- Resetting ENC in single-channel single-conversion mode
  - Stops a conversion immediately and results are unpredictable.
  - For correct results, poll ADC10BUSY bit until reset before clearing ENC
- Resetting ENC during repeat-single-channel operation
  - Stops converter at the end of current conversion
- Resetting ENC during a sequence or repeat sequence mode
  - stops converter at the end of sequence
- Any conversion mode may be stopped immediately by setting CONSEQx = 0 and resetting ENC bit
  - Conversion data is unreliable

ADC Memory Transfer

ADC10 Data Transfer Controller

- DTC automatically transfers conversion results from ADC10MEM to other on-chip memory locations.
- DTC is enabled by setting the ADC10DTC1 register to a nonzero value.
  - When DTC is enabled, each time ADC10 completes a conversion and loads result to ADC10MEM, a data transfer is triggered.
  - No software intervention is required to manage ADC10 until predefined amount of conversion data has been transferred.

DTC provides a simple form of Direct Memory Access (DMA).

ADC10 Data Transfer Controller

- Each DTC transfer requires one CPU MCLK.
  - To avoid any bus contention during DTC transfer, CPU is halted if active
- DTC transfer must not be initiated while the ADC10 is busy.
  - Software must ensure that no active conversion or sequence is in progress when DTC is configured

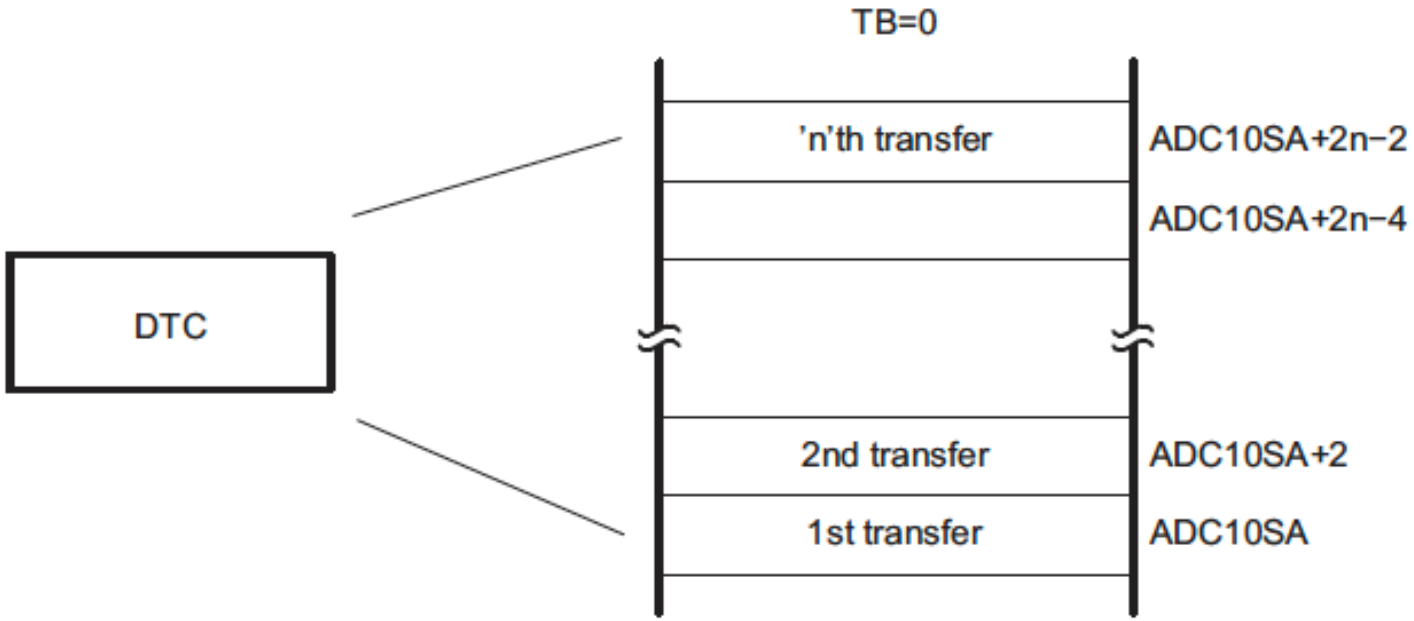
DTC provides a simple form of Direct Memory Access (DMA).

One-Block Transfer Mode

- One-block mode is selected if ADC10TB is reset.
- DTC transfers continue with each loading of ADC10MEM until the internal transfer counter becomes zero.
  - No additional DTC transfers occur until a write to ADC10SA.
- In one-block mode, ADC10IFG flag is set only after a complete block has been transferred.

One-Block Transfer Mode

- Value n in ADC10DTC1 defines total number of transfers for a block
  - Block start address is defined anywhere in MSP430 address range using 16-bit register ADC10SA.
  - Block ends at ADC10SA + 2n – 2.



State Diagram for DTC in One-Block Mode

ADC10CT: ADC10 continuous transfer

ADC10CT=0, Data transfer stops when one block (one-block mode) or two blocks (two-block mode) have completed.

ADC10CT=1, Data is transferred continuously.

DTC operation is stopped only if ADC10CT cleared, or ADC10SA is written to.

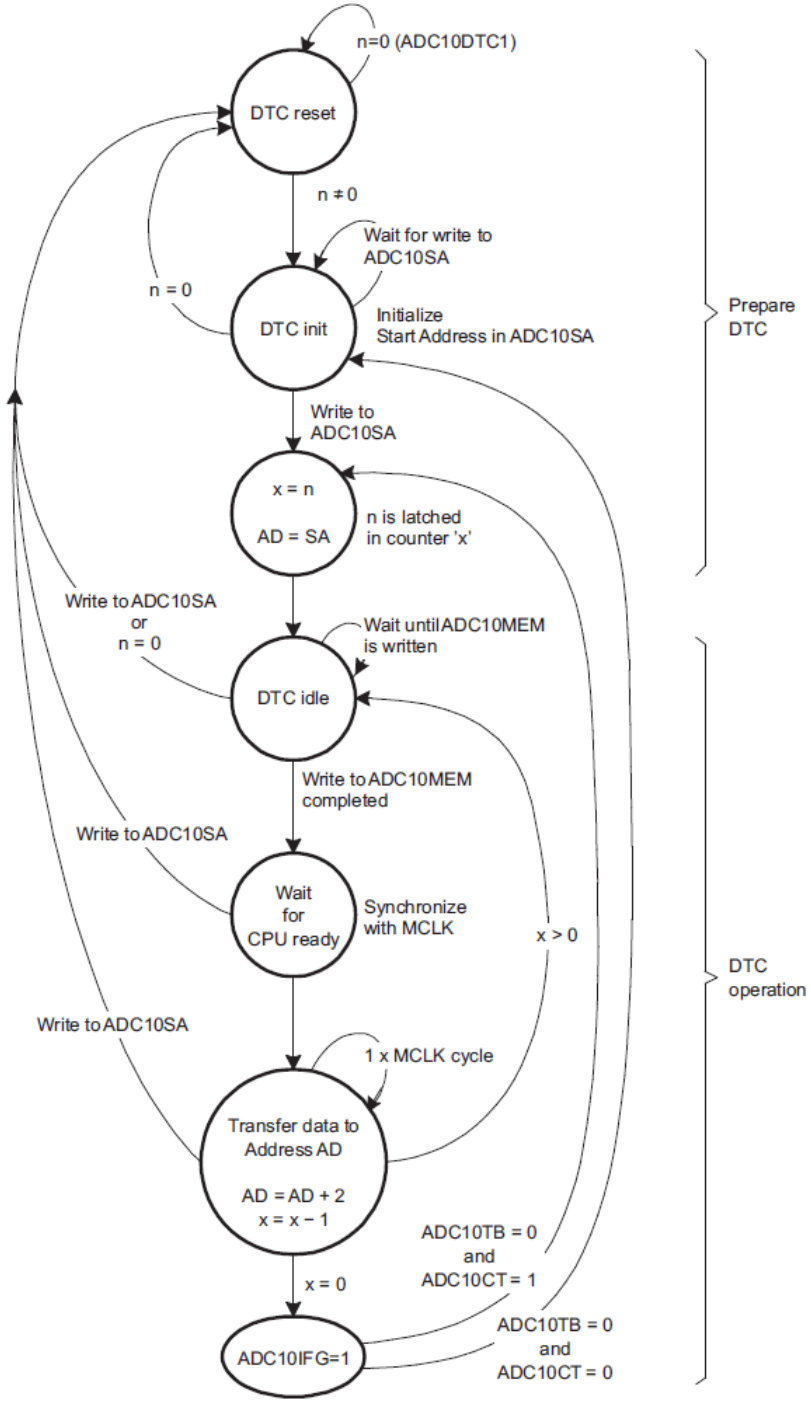
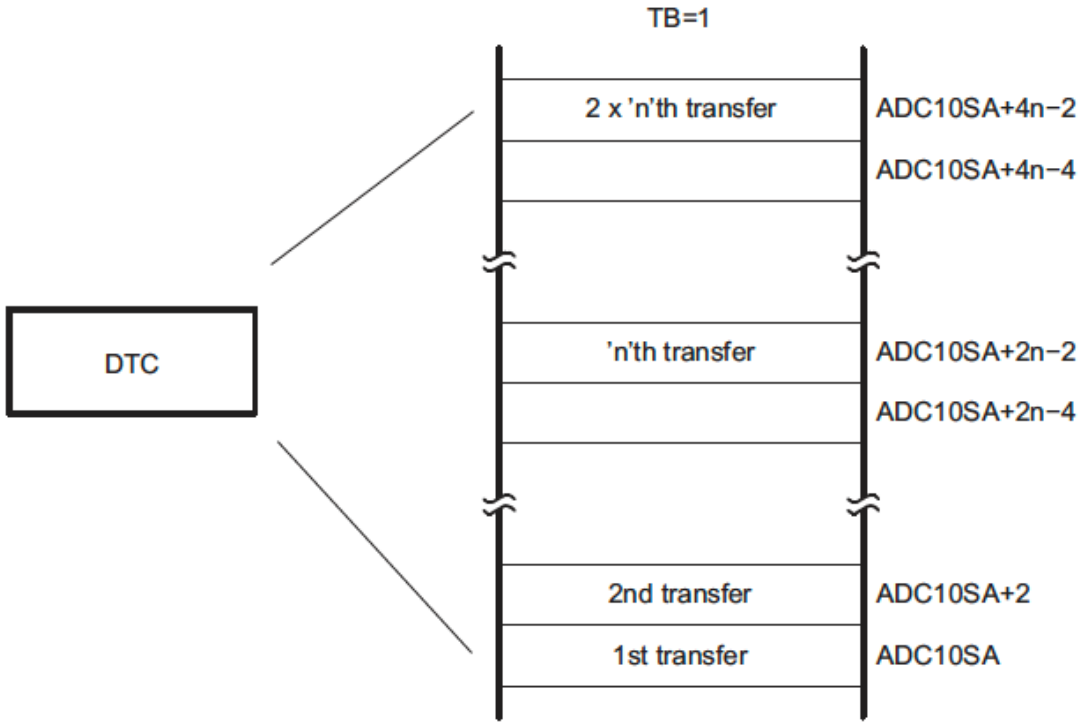


Figure 22-10. State Diagram for Data Transfer Control in One-Block Transfer Mode

Two-Block Transfer Mode

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



Two-Block Transfer Mode

- When block one is full and both ADC10IFG flag and ADC10B1 bit are set.
  - User can test ADC10B1 bit to determine if block one is full.
- DTC continues with block two.
  - Internal transfer counter is automatically reloaded with 'n’.
  - At the next load of ADC10MEM, DTC begins transferring conversion results to block two.
- After n transfers have completed, block two is full.
  - ADC10IFG flag is set and ADC10B1 bit is cleared.

State Diagram for DTC in Two-Block Mode

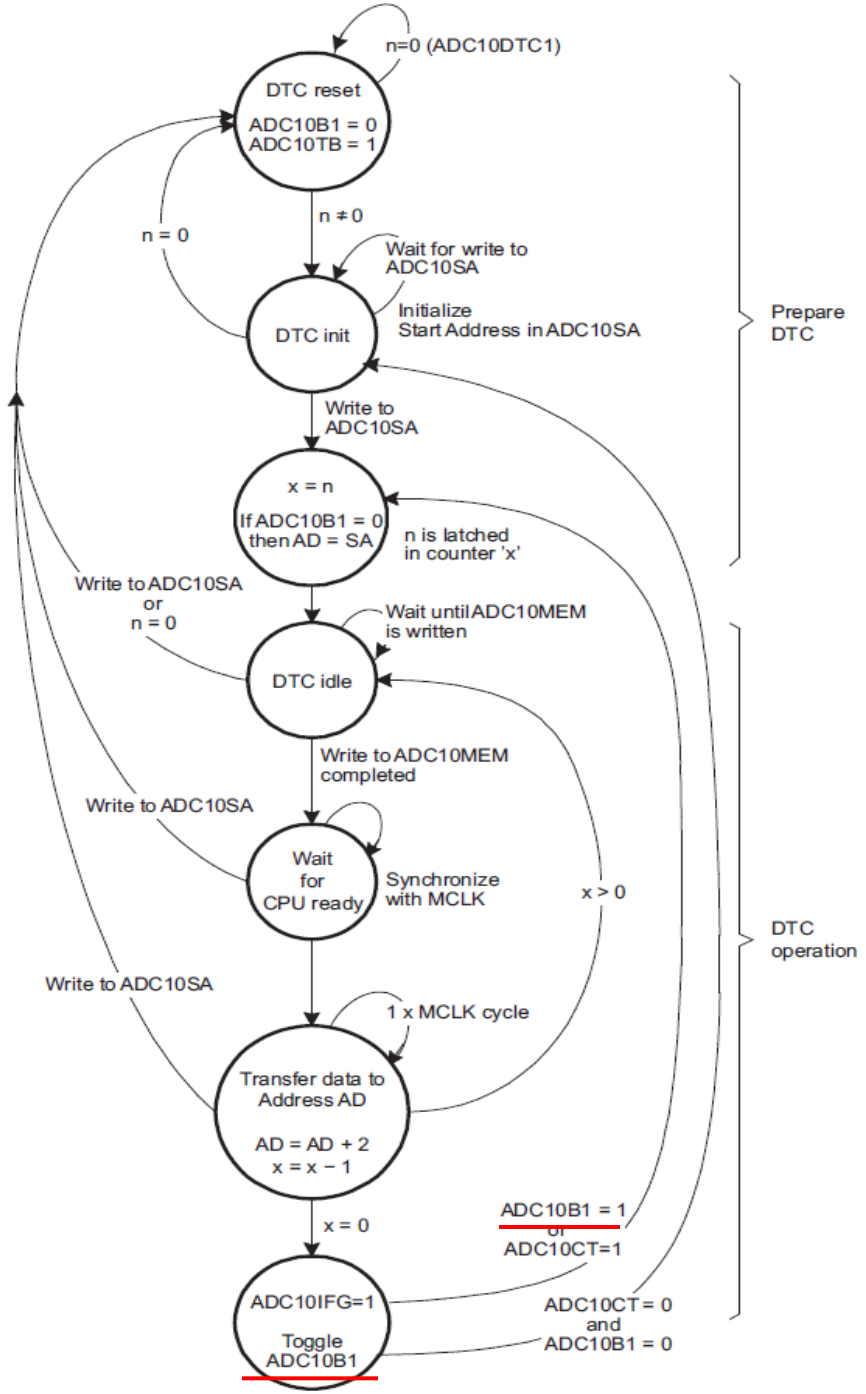


Figure 22-12. State Diagram for Data Transfer Control in Two-Block Transfer Mode

Why should two-block mode DTC be used?

- Three possible methods when ADC should convert continuously and data should be processed while ADC is producing new data

Method 1:

1. DTC is not used:
   - Main program checks the end of conversion (ADC10BUSY or ADC10IFG) in a loop
   - Main reads the data from ADC10MEM and processes the data. (Example 2)

Why should two-block mode DTC be used?

- Three possible methods when ADC should convert continuously and data should be processed while ADC is producing new data

Method 2:

2. DTC is not used. ADC interrupt is used.
   - ISR reads data from ADC10MEM
   - ISR processes the data (or passes the data to the main program). (Examples 4 and 5)

Why should two-block mode DTC be used?

- Three possible methods when ADC should convert continuously and data should be processed while ADC is producing new data

Method 3:

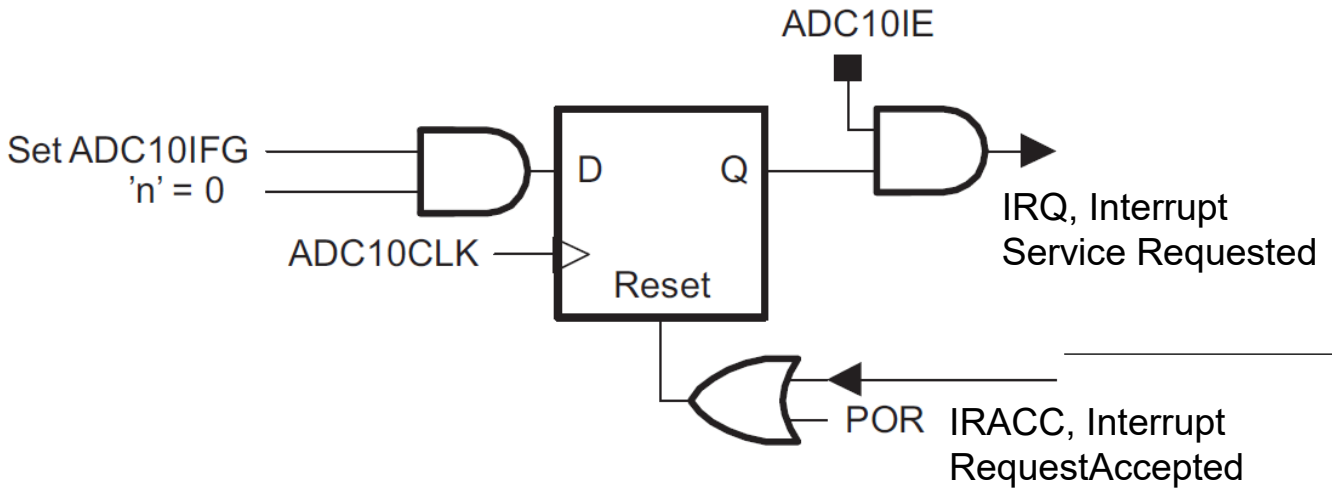
3. DTC in Two-block mode is used.
    - main program or ISR reads the data from Block 1 while DTC is filling Block 2 or vice versa.

ADC10 Interrupts

- One interrupt and one interrupt vector are associated with ADC10
- When DTC is not used (ADC10DTC1 = 0)
  - ADC10IFG is set when conversion results are saved to ADC10MEM.
- When DTC is used (ADC10DTC1 > 0)
  - ADC10IFG is set when a block transfer completes and transfer counter n = 0.
- If both ADC10IE and GIE bits are set, then ADC10IFG flag generates an interrupt request.
  - ADC10IFG flag is automatically reset when interrupt request is serviced,
  - Flag may be reset by software

ADC10 Interrupts

- When DTC is not used (ADC10DTC1 = 0), ADC10IFG is set when conversion results are saved to ADC10MEM.
- When DTC is used (ADC10DTC1 > 0), ADC10IFG is set when a block transfer completes and transfer counter n = 0.
- If both ADC10IE and GIE bits are set, then ADC10IFG flag generates an interrupt request.
  - ADC10IFG flag is automatically reset when interrupt request is serviced,
  - Flag may be reset by software



MSP 430 ADC Registers

ADC Registers summary

Table 22-3. ADC10 Registers

Register	Short Form	Register Type	Address	Initial State
ADC10 input enable register 0	ADC10AE0	Read/write	04Ah	Reset with POR
ADC10 input enable register 1	ADC10AE1	Read/write	04Bh	Reset with POR
ADC10 control register 0	ADC10CTL0	Read/write	01B0h	Reset with POR
ADC10 control register 1	ADC10CTL1	Read/write	01B2h	Reset with POR
ADC10 memory	ADC10MEM	Read	01B4h	Unchanged
ADC10 data transfer control register 0	ADC10DTC0	Read/write	048h	Reset with POR
ADC10 data transfer control register 1	ADC10DTC1	Read/write	049h	Reset with POR
ADC10 data transfer start address	ADC10SA	Read/write	01BCh	0200h with POR

MSP ADC Programming

Function	Parameter	Name	Register
I/O management	Select input channels	INCHx	ADC10CTL1
	Input enable/ disable	ADC10AE0	ADC10AE0
Clock	Clock source	ADC10SSELx	ADC10CTL1
	Clock divider	ADC10DIVx	ADC10CTL1
Reference voltages	Reference source	SREFx	ADC10CTL0
	Internal reference generator on/off	REFON	ADC10CTL0
	Internal reference gen. voltage	REF2_5V	ADC10CTL0
Start/ enable	ADC on/ off	ADC10ON	ADC10CTL0
	Enable/disable conversion	ENC	ADC10CTL0
	Start conversion	ADC10SC	ADC10CTL0
Sampling	Sample-Hold select/trigger	SHSx	ADC10CTL1
	Sample-Hold time	ADC10SHTx	ADC10CTL0
	Sampling rate	ADC10SR	ADC10CTL0

MSP ADC Programming

Function	Parameter	Name	Register
Status	Busy	ADC10BUSY	ADC10CTL1
Interrupts	ADC interrupt enable	ADC10IE	ADC10CTL0
	ADC interrupt flag	ADC10IFG	ADC10CTL0
Data acquisition	Conversion modes	CONSEQx	ADC10CTL1
	Multiple sample and conversion	MSC	ADC10CTL0
	Data format: Binary/ 2's compl.	ADC10DF	ADC10CTL1
Memory management	One/two block mode	ADC10TB	ADC10DTC0
	Continuous transfer: Single/multiple	ADC10CT	ADC10DTC0
	Block order (Block1 or 2)	ADC10B1	ADC10DTC0
	DTC Number of transfers		ADC10DTC1
	DTC start address		ADC10SA

MSP 430 ADC Memory Registers

Analog (Input) Enable Control Register 0: ADC10AE0

7	6	5	4	3	2	1	0
ADC10AE0x							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

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. The analog enable bit of not implemented channels should not be programmed to 1.
		0 Analog input disabled
		1 Analog input enabled

ADC Conversion-Memory Register: ADC10MEM

22.3.5 ADC10MEM, Conversion-Memory Register, Binary Format

15	14	13	12	11	10	9	8
0	0	0	0	0	0	**Conversion Results**	
r0	r0	r0	r0	r0	r0	r	r
7	6	5	4	3	2	1	0
Conversion Results							
r	r	r	r	r	r	r	r

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.

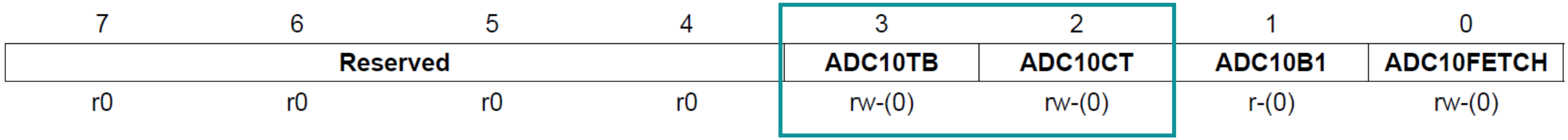
Change data format: ADC10DF in ADC10CTL1

22.3.6 ADC10MEM, Conversion-Memory Register, 2s Complement Format

15	14	13	12	11	10	9	8
Conversion Results							
r	r	r	r	r	r	r	r
7	6	5	4	3	2	1	0
Conversion Results		0	0	0	0	0	0
r	r	r0	r0	r0	r0	r0	r0

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.

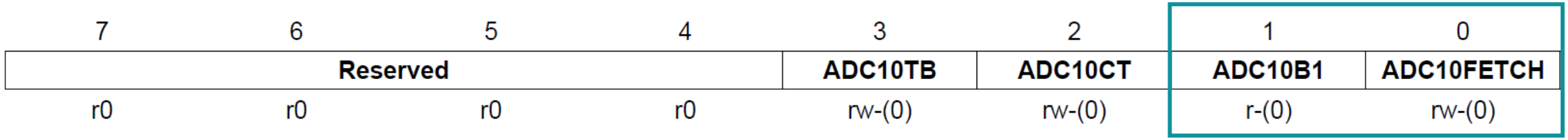
Data Transfer Control Register 0: ADC10DTC0



7	6	5	4	3	2	1	0
Reserved				ADC10TB	ADC10CT	ADC10B1	ADC10FETCH
r0	r0	r0	r0	rw-(0)	rw-(0)	r-(0)	rw-(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.

Data Transfer Control Register 0: ADC10DTC0



7	6	5	4	3	2	1	0
Reserved				ADC10TB	ADC10CT	ADC10B1	ADC10FETCH
r0	r0	r0	r0	rw-(0)	rw-(0)	r-(0)	rw-(0)

ADC10B1	Bit 1	ADC10 block one. This bit indicates for two-block mode which block is filled with ADC10 conversion results.
	0	Block 2 is filled
	1	Block 1 is filled

ADC10FETCH	Bit 0	This bit should normally be reset.

Data Transfer Registers

22.3.8 ADC10DTC1, Data Transfer Control Register 1

7	6	5	4	3	2	1	0
DTC Transfers							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)

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

15	14	13	12	11	10	9	8
ADC10SAx							
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)
7	6	5	4	3	2	1	0
ADC10SAx							0
rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	rw-(0)	r0

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.

MSP430 ADC Operation

- ADC on/off: ADC10ON bit in ADC10CTL0 control register 0
- ENC=0 if we want to modify ADC10CTL registers
- Set ENC=1 to enable conversion
- ADC conversion initiation: rising edge of SHI
  - SHI source: ADC10SC bit or timer output units
- Write into ADC10SA to enable DTC transfer
  - ADC10SA= Start address of destination
- ENC enable and trigger for initiation must be repeated for each conversion

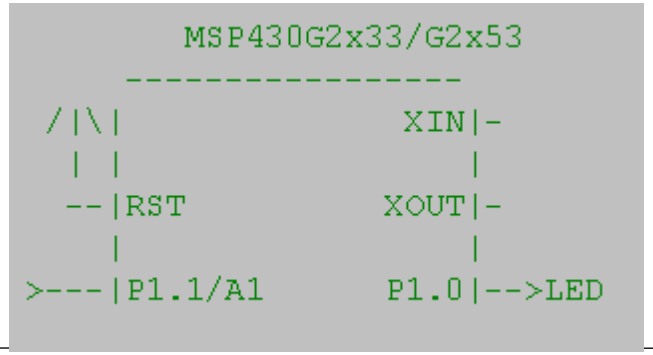
MSP430 ADC Trigger Options

Trigger Source	CONSEQ = 00/01 MSC = x	CONSEQ = 10/11 MSC = 0	CONSEQ = 10/11 MSC = 1
Hardware	Single conversions, individually triggered	Repeated conversions, individually triggered	Multiple conversions, only first triggered
	ENC: toggle ENC every time	ENC: toggle ENC only before start	
Software	Repeated conversions, individually triggered		Multiple conversions, only first triggered
	toggle ENC only before start or with ADC10SC in all cases		

MSP 430 Code Examples

Example 1: Single-Channel Single-Conversion Mode, ADC clock: Internal ADC oscillator, 5 MHz

```
void main(void)
    {  WDTCTL = WDTPW + WDTHOLD;                    // Stop WDT
      ADC10CTL0 = ADC10SHT_2 + ADC10ON;     // Sample and hold time=16 ADC clocks, ADC ON
      ADC10CTL1 = INCH_1;                           // input A1
      ADC10AE0 |= 0x02;                             // P1.1 ADC function selected
      P1DIR |= 0x01;                                // Set P1.0 to output direction
      ADC10CTL0 |= ENC + ADC10SC;                // Enable and start conversion
      while(ADC10CTL1 & ADC10BUSY) {} //Wait until the conversion is ready (Testing ADC10BUSY
bit in ADC10CTL1)
      if (ADC10MEM < 0x1FF)                 //
    {P1OUT &= ~0x01;} // Clear P1.0 LED off
      else
        {P1OUT |= 0x01;} // Set P1.0 LED on
    }
```



Example 2: Single ADC triggered in an infinite for loop, ADC clock: SMCLK 1 MHz

```
void main(void)
    { WDTCTL = WDTPW + WDTHOLD;                    // Stop WDT
      ADC10CTL0 = ADC10SHT_2 + ADC10ON;      // Sample and hold time = 16 ADC clocks, ADC ON
      ADC10CTL1 = INCH_1 + ADC10SSEL_3 ;          // input A1, SMCLK 1 MHz
      ADC10AE0 |= 0x02;                           // P1.1 ADC function selected
      P1DIR |= 0x01;                              // Set P1.0 to output direction
for(;;){
      ADC10CTL0 |= ENC + ADC10SC;                // Enable and start conversion
      while(ADC10CTL1 & ADC10BUSY) {} //wait until the conversion is ready (Testing ADC10BUSY
bit in ADC10CTL1)
      if (ADC10MEM < 0x1FF)                  //Voltage = 3.3V * 0x1FF / 0x3FF
        {P1OUT &= ~0x01;}                         // Clear P1.0 LED off
      else
        {P1OUT |= 0x01;}                          // Set P1.0 LED on
    }
}
```

Example 3: Single ADC triggered in an infinite for loop, Temperature sensor

```
void main(void)
    {  WDTCTL = WDTPW + WDTHOLD;                    // Stop WDT
      ADC10CTL0 = ADC10SHT_3 + ADC10ON; // Sample and hold time = 64 ADC clocks, ADC ON
      ADC10CTL1 = INCH_10 + ADC10SSEL_3 ;          // Temperature sensor, SMCLK 1 MHz
      P1DIR |= 0x01;                              // Set P1.0 to output direction
for(;;){
      ADC10CTL0 |= ENC + ADC10SC;                // Enable and start conversion
      while(ADC10CTL1 & ADC10BUSY) {} //wait until the conversion is ready (Testing
ADC10BUSY bit in ADC10CTL1)
      if (ADC10MEM < 0x1FF)  //Select a proper value to test the temperature sensor
    {P1OUT &= ~0x01;}                            // Clear P1.0 LED off
      else
        {P1OUT |= 0x01;}                         // Set P1.0 LED on
    }
}
```

Temperature sensor requires sample-and-hold time > 30µs.

Example 4: Single-Channel Single-Conversion Mode with Interrupt

```
void main(void)
{  WDTCTL = WDTPW + WDTHOLD;                    // Stop WDT
  ADC10CTL0 = ADC10SHT_2 + ADC10ON + ADC10IE; // ADC10ON, interrupt enabled
  ADC10CTL1 = INCH_1 + ADC10SSEL_3 ;           // input A1, SMCLK
  ADC10AE0 |= 0x02;                           // P1.1 ADC function selected
  P1DIR |= 0x01;                              // Set P1.0 to output direction
  ADC10CTL0 |= ENC + ADC10SC;                // Enable and start conversion
  __bis_SR_register(LPM0_bits + GIE);        // LPM0, general interrupt enable
}
#pragma vector=ADC10_VECTOR
__interrupt void ADC10_ISR(void)
{
    if (ADC10MEM < 0x1FF)                    //Voltage = 3.3V * 0x1FF / 0x3FF
      {P1OUT &= ~0x01; }                        // Clear P1.0 LED off
    else
    {  P1OUT |= 0x01;  }              // Set P1.0 LED on
    ADC10CTL0 |= ENC + ADC10SC;     // Enable and start conversion
}
```

Example 5, Repeat single channel, Multiple sample and conversion with interrupt

```
void main(void)
    { WDTCTL = WDTPW + WDTHOLD;                    // Stop WDT
      // Sample and hold time = 16 ADC clocks, ADC10ON, Enable interrupt, Multiple sample and
conversion
      ADC10CTL0 = ADC10SHT_2 + ADC10ON + ADC10IE + MSC;
      ADC10CTL1 = CONSEQ_2 + INCH_1 + ADC10SSEL_3;  // Repeat single channel, input A1, ADC clock:
SMCLK
      ADC10AE0 |= 0x02;                            // P1.1 ADC function selected
      P1DIR |= 0x01;                               // Set P1.0 to output direction
      ADC10CTL0 |= ENC + ADC10SC;                 // Sampling and conversion start
      __bis_SR_register(LPM0_bits + GIE); //DO NOT forget the global interrupt enable
}
#pragma vector=ADC10_VECTOR
__interrupt void ADC10_ISR(void)
{
    if (ADC10MEM < 0x1FF)
      {P1OUT &= ~0x01; }                     // Clear P1.0 LED off
    else
    {  P1OUT |= 0x01;  }                    // Set P1.0 LED on
}
```

Sample code – DTC (one block)

```
//create a result buffer

/*the number of element is same as
  number of ADC reading using DTC*/

int a[0x10] = {0};

/*enable the DTC by setting the
  ADC10DTC1 with a number other than 0*/

ADC10DTC1 = 0x10;

/*Set the address of the ADC buffer with
  the address of the Array */

ADC10SA = (int)a;

or

ADC10SA = (unsigned short)a;
```

Watch 1

Expression	Value	Location	Type
av	Error (c...		
a	<array>	Memory:0x200	int[16]
[0]	1022	Memory:0x200	int
[1]	1022	Memory:0x202	int
[2]	1022	Memory:0x204	int
[3]	1022	Memory:0x206	int
[4]	1022	Memory:0x208	int
[5]	1022	Memory:0x20A	int
[6]	1021	Memory:0x20C	int
[7]	1023	Memory:0x20E	int
[8]	1022	Memory:0x210	int
[9]	1022	Memory:0x212	int
[10]	1017	Memory:0x214	int
[11]	1022	Memory:0x216	int
[12]	1022	Memory:0x218	int
[13]	1022	Memory:0x21A	int
[14]	1022	Memory:0x21C	int
[15]	1023	Memory:0x21E	int
ADC10M...	1022	Memory:0x1B4	unsigned short
<click to ...			

Next Lecture: Clock and Communication Modules