Lecture 8
SAR ADC

Texas Instruments Incorporated
University of Beira Interior (PT)

Pedro Dinis Gaspar, António Espírito Santo, Bruno Ribeiro, Humberto Santos
University of Beira Interior, Electromechanical Engineering Department
www.msp430.ubi.pt

>> Contents

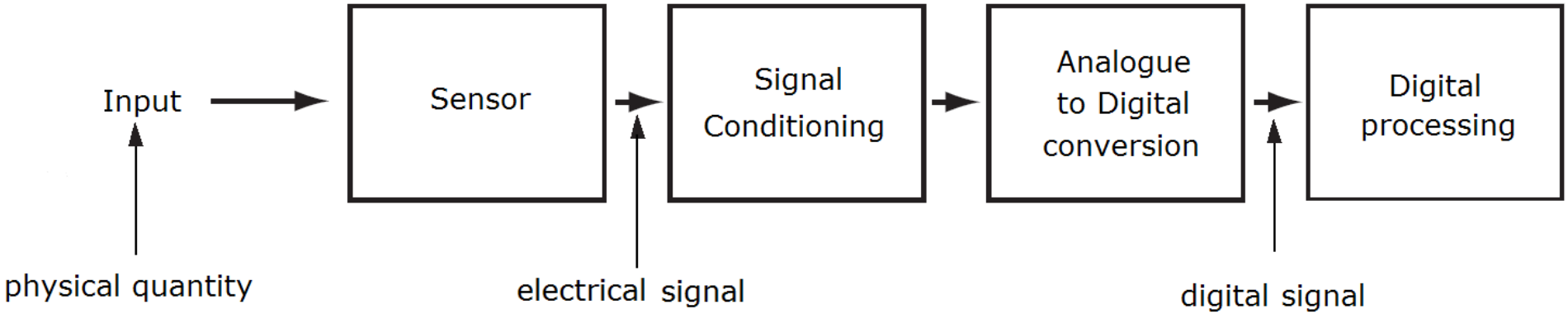
Contents (1/3)

Introduction (1/4)

- **Most engineering applications require some form of data processing: measurement, control, calculation, communication or data recording;**
- **These operations, either grouped or isolated, are built into the measuring instruments;**
- **The measuring equipment must maintain:**
  - Compatibility and communication between measuring devices;
  - Acceptable error margin;
  - Noise and interference immunity;
  - Predictable measurement uncertainty;
  - Suitable type of control (analogue/digital);
  - Mathematical processing capacity;
  - …

Introduction (2/4)

❑ Data acquisition system components:



- **Sensors:**
  - Convert analogue measurements of physical quantities (e.g. temperature, pressure, humidity, velocity, flow-rate, linear motion, position) into electrical signals (voltage or current).

Introduction (3/4)

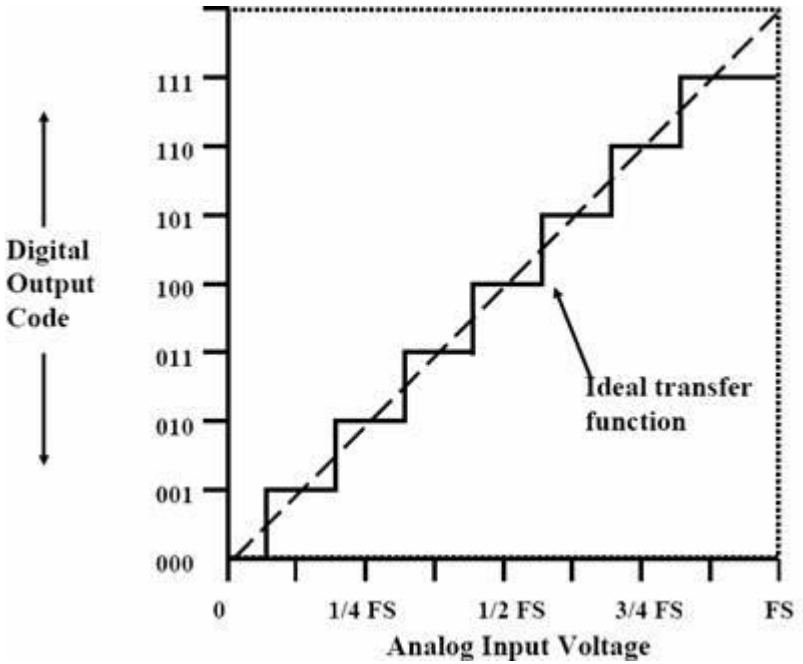
- **Data acquisition system components:**
  - **Signal conditioning (filtering and amplification):**
    - The operations required to convert the measured analogue signal to the electrical signal range of the analogue-to-digital converter (ADC) may involve filtering, amplification, attenuation or impedance transformation.
  - **Analogue-to-Digital Converter (ADC):**
    - Input: Signal to be measured;
    - Output: A digital code compatible with the digital processing system;
    - Requires:
      - Sample-and-hold: Used to take a snapshot of the continuously changing input signal and maintain the value over the sample interval set by a clock system;
      - A sampling frequency based on the Nyquist theorem.

Introduction (4/4)

- **Data acquisition system components:**
  - **Analogue-to-Digital Converter (ADC) (continued):**
    - Sample-and-Hold: Not necessary for Sigma-Delta (SD) converters, nor for slope converters, nor for all flash converters and is automatically implemented as part of the structure of capacitive successive approximation Register (SAR) converters on the MSP430.

Analogue-to-Digital Conversion (1/2)

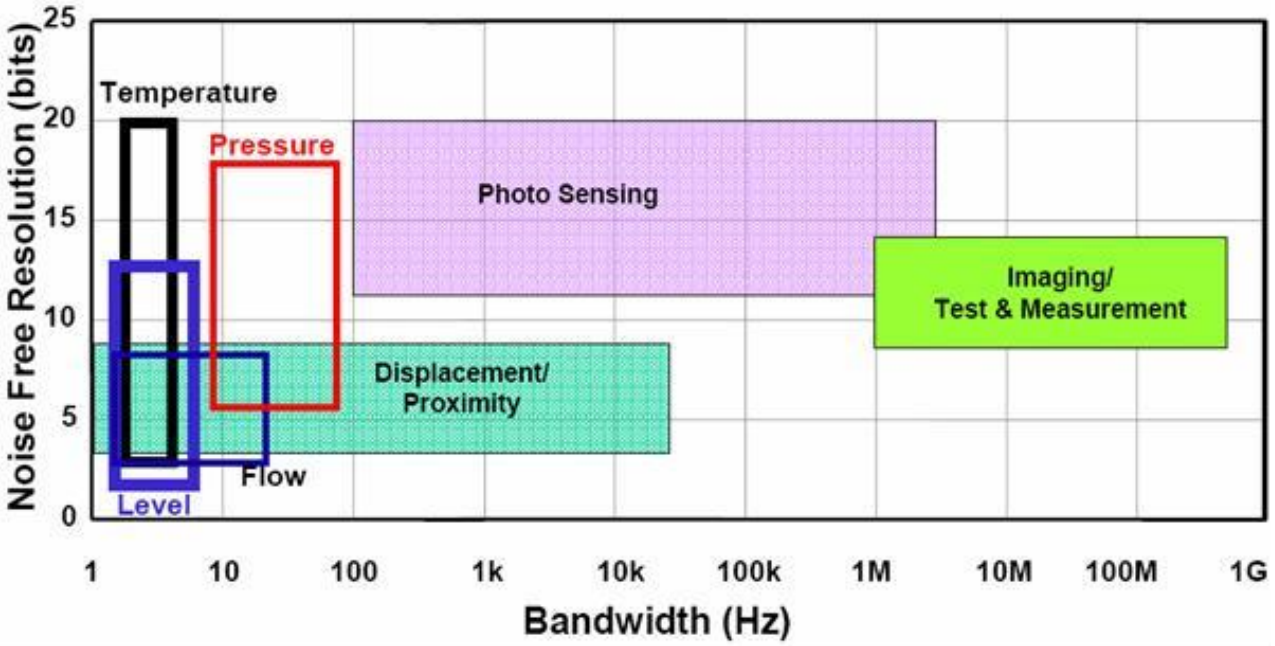
- The analogue world (the real one) interfaces with digital systems through ADCs;
- The ADC takes the voltage from the acquisition system (after signal conditioning) and converts it to an equivalent digital code;
- The ADC ideal transfer function for a 3 bit ADC is given by:



- The digital code can be displayed, processed, stored or transmitted.

Analogue-to-Digital Conversion (2/2)

- **There are sufficient analogue peripherals in a number of MSP430 family devices to realize a complete signal chain;**
- **Analogue class of applications:**
  - Is more or less defined by bandwidth range;
  - Require an established resolution range.



ADC Specifications

Resolution, *R*:

- The smallest change to the analogue voltage that can be converted into a digital code;
- The Least Significant Bit (LSB): $R = \frac{1}{2^n}$
- The resolution only specifies the width of the digital output word, not the performance;
- Most MSP430 devices offer a high-precision ADC:
  - Slope;
  - 10, 12 or 14 Bit SAR;
  - 16 Bit Sigma-Delta.

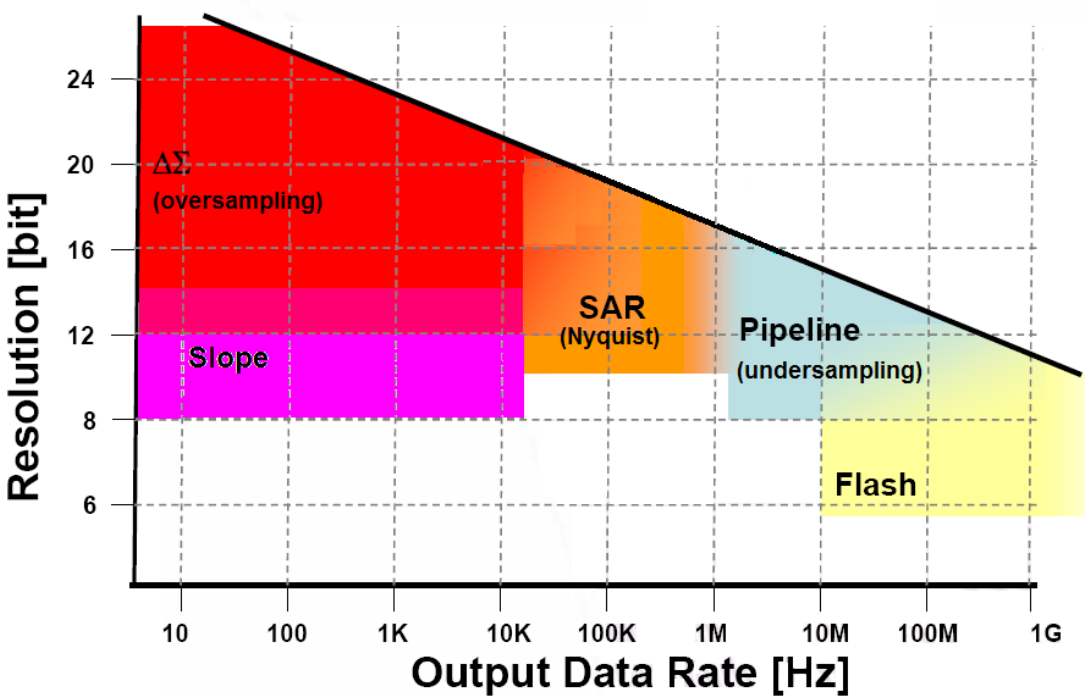
ADC Specifications – DC performance

- **Code-Edge Noise: Amount of noise that appears right at a code transition of the transfer function;**

- **Voltage Reference (internal or external): Besides the settling time, the source of the reference voltage errors is related to the following specifications:**
  - Temperature drift: Affects the performance of an ADC converter based on resolution;
  - Voltage noise: Specified as either an RMS value or a peak-to-peak value;
  - Load regulation: Current drawn by other components will affect the voltage reference;
  - Temperature effects (offset drift and gain drift).

ADC Architectures (1/3)

- **There are many different ADC architectures:**
  - Successive Approximation (SAR);
  - Sigma Delta (SD or ΔΣ);
  - Slope or Dual Slope;
  - Pipeline;
  - Flash...as in quick, not memory.



ADC Architectures (2/3)

- **The selection of an MSP430 ADC will depend on:**
  - Voltage range to be measured;
  - Maximum frequency for A_{IN};
  - Minimum resolution needed *vs.* analogue input variation;
  - The need for differential inputs;
  - Voltage reference range;
  - The need for multiple channels for different analogue inputs.

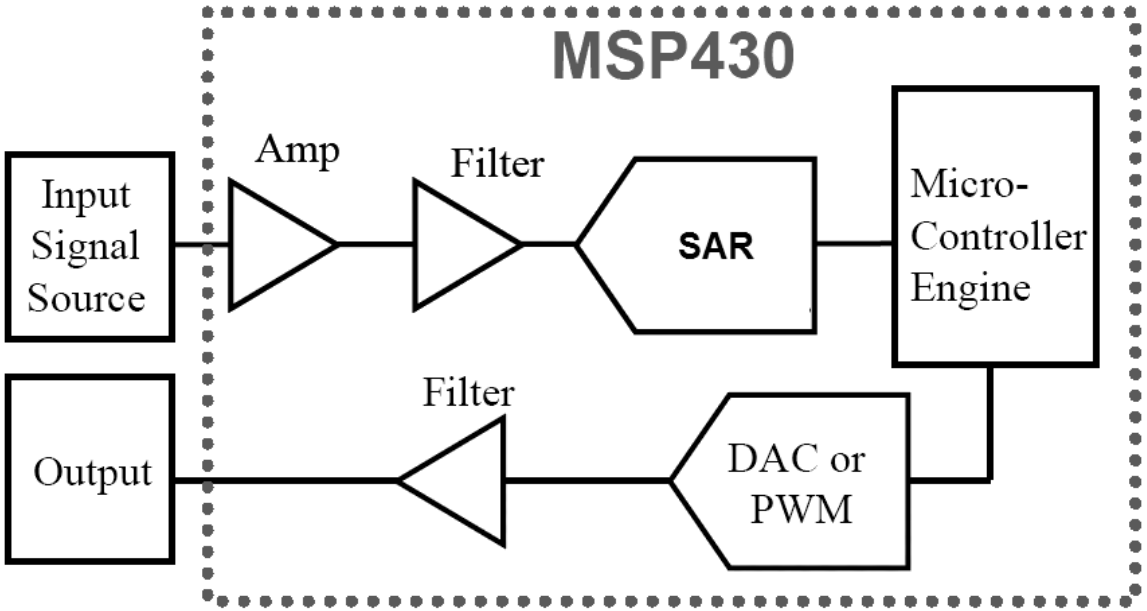
ADC architecture	Resolution	Conversion rate	Advantages	Disadvantages
SAR	≤ 18 bit	< 5 Msps	Zero-cycle latency Low latency-time High accuracy Low power Simple operation	Sample rates 2-5 MHz
SD	≤ 24 bit ≤ 16-18 bit	< 625 ksps < 10 Msps	High resolution High stability Low power Moderate cost	Cycle-latency Low speed
Pipeline	≤ 16 bit	< 500 Msps	Higher speeds Higher bandwidth	Lower resolution Delay/Data latency Power requirements

ADC Architectures (3/3)

- **ADC architectures included in the MSP430 devices populated in the hardware development tools:**
  - 10 Bit SAR: MSP430F2274 ⇒ eZ430-RF2500;
  - 12 Bit SAR: MSP430FG4618 ⇒ Experimenter's board;
  - 16 Bit Sigma-Delta: MSP430F2013 ⇒ eZ430-F2013 and Experimenter's board.

Introduction to SAR ADC (1/4)

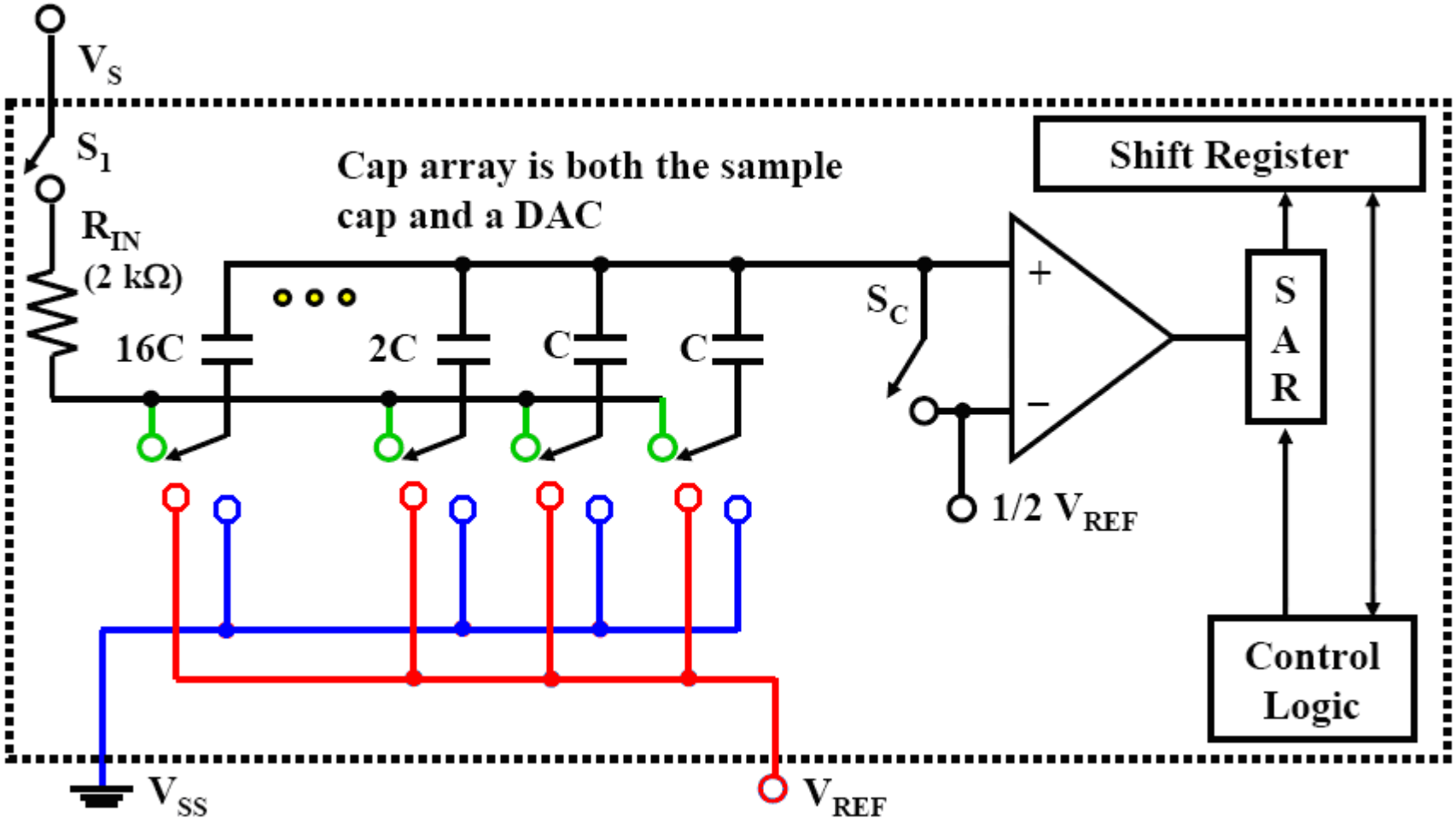
- **Successive Approximation Register (SAR) converters are well-suited to general purpose applications and are used in a wide range signal interfacing applications:**
  - Data loggers;
  - Temperature sensors;
  - Bridge sensors (resistive e.g. strain gauges);
  - General purpose.



>> Contents

Introduction to SAR ADC (2/4)

- **SAR block diagram:**



Introduction to SAR ADC (3/4)

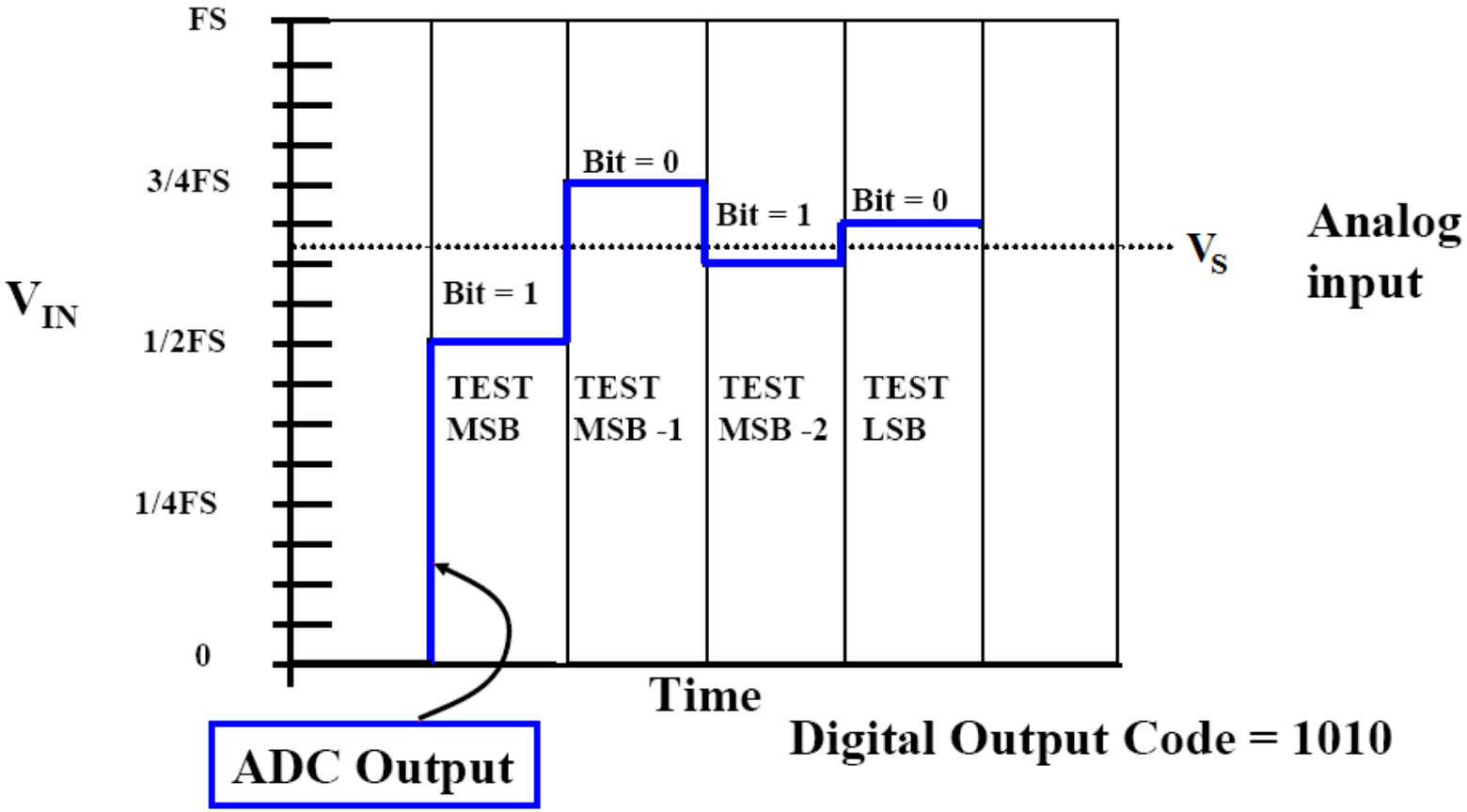
SAR concept:

- Determines the digital word by approximating the analogue input signal using an iterative process, as follows:

  - Discharge the capacitor array to the comparator's V_{offset};
  - Sample the input voltage (V_S) and hold;
  - Switch all of the capacitors in the array to V_S;
  - Switch the capacitors to charge the comparator's input;
  - Initiate a binary search:
    - Switch the MSB capacitor to V_{REF} (ADC's *FS* range):
      - » Divided 1:1 between it and the rest of the array;
      - » Input voltage to the comparator is - V_S + $V_{REF}/2$;
      - » $V_S > V_{REF}/2 \Rightarrow$ Comparator output: MSB = 1;
      - » $V_S < V_{REF}/2 \Rightarrow$ Comparator output: MSB = 0;
    - Switch the other capacitors in a decreasing charge capacity order from 16C to C.

Introduction to SAR ADC (4/4)

- **SAR concept:**



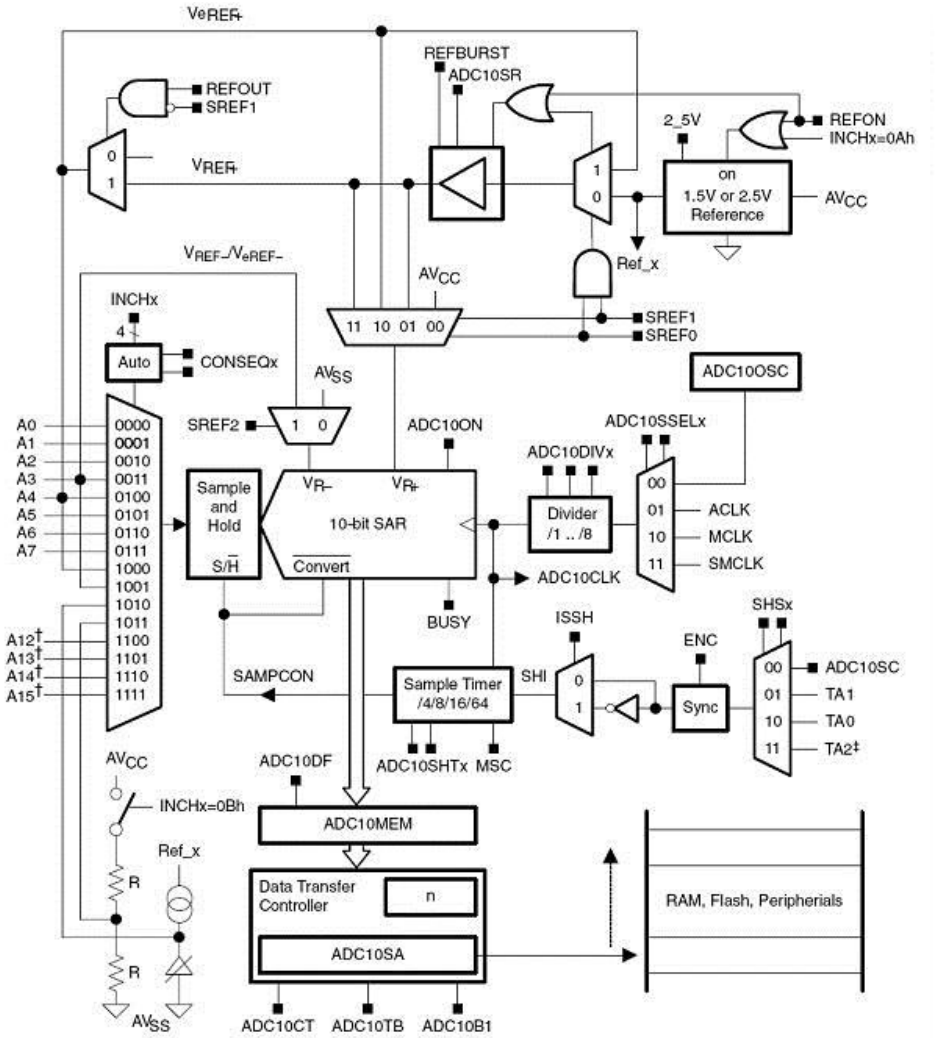
ADC10 (1/2)
Description

- **The ADC10 module of the MSP430F2274 supports fast 10-bit analogue-to-digital conversions;**

- **The module contains:**
  - 10-bit SAR core;
  - Sample select control;
  - Reference generator;
  - Data transfer controller (DTC) for automatic conversion result handling (ADC samples conversion and storage without CPU intervention).

ADC10 (2/2)
Description

- **ADC10 block diagram:**



†MSP430x22xx devices only. Channels A12-A15 tied to channel A11 in other devices
‡TA1 on MSP430x20x2 devices

>> Contents

ADC10
Features

- Greater than 200 ksps maximum conversion rate;
- Monotonic 10-bit converter with no missing codes;
- Sample-and-hold with programmable sample periods;
- Conversion initiated by software or Timer_A;
- Software on-chip reference voltage generation (1.5 V or 2.5 V)
- Software selectable internal or external reference;
- Eight external input channels;
- 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 (powered down separately);
- Data transfer controller (automatic storage of results).

ADC10
10 bit ADC core

- **10 bit ADC core (enable with ADC10ON bit):**
  - Converts an analogue input to its 10-bit digital representation;
  - Stores the result in the ADC10MEM register;
  - The analogue conversion range is limited by the upper and lower limits: V_{R+} ; V_{R-}
  - The digital output (N_{ADC}) is:
    - Full scale: N_{ADC} = 03FFh, when the input signal ≥ V_{R+} -0.5LSB;
    - Zero: N_{ADC} = 0000h, when the input signal ≤ V_{R-} + 0.5 LSB.
  - Conversion results:
    - Binary format: $N_{ADC} = 1023\frac{V_{in} - V_{R-}}{V_{R+} - V_{R-}}$
    - Two's-complement format.

ADC10
Conversion clock

- **The ADC10CLK is used both as the conversion clock and to generate the sampling period;**

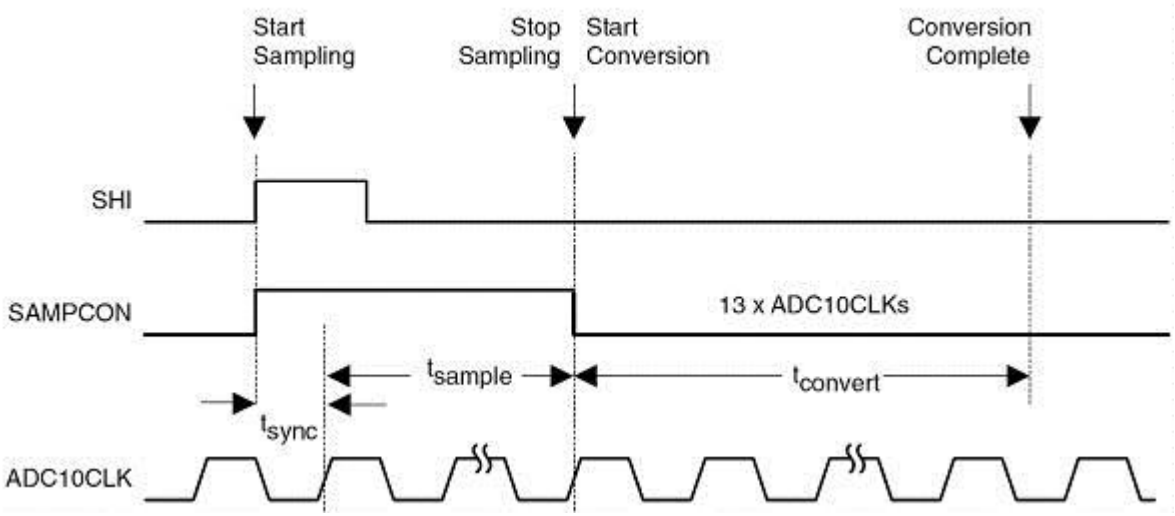
- **Each available ADC10 source clock is selected using the ADC10SSELx bits:**
  - SMCLK;
  - MCLK;
  - ACLK;
  - Internal oscillator ADC10OSC;
  - Each clock source can be divided from 1-8 (ADC10DIVx bits).

- **The ADC10CLK must remain active until the end of a conversion.**

ADC10
Sample and conversion timing

- **An A/D conversion is initiated by the rising edge of SHI. The sources of SHI (SHSx bits selection) can be:**
  - ADC10SC bit;
  - Timer_A Output Unit 1, Output Unit 0, or Output Unit 2.

- **The SHTx bits select the sample period, t_{sample}, to be 4, 8, 16, or 64 ADC10CLK cycles:**



ADC10 (1/2)
Conversion modes

- **Conversion modes (selected by the CONSEQx bits):**
  - ***Single channel, single-conversion***: A single conversion for the channel selected by INCHx bits is performed, with the result being stored in the ADC10MEM register;
  - ***Sequence of channels***: One conversion in multiple channels, beginning with the channel selected by INCHx bits and decrementing to channel A0, looping through the ADC10MEM register and stopping after the conversion of channel A0.

ADC10 (2/2)
Conversion modes

- **Conversion modes (selected by the CONSEQx bits):**
  - ***Repeat single channel***: A single channel selected by INCHx bits is converted repeatedly until stopped and the result is stored in the ADC10MEM register;
  - ***Repeat sequence of-channels***: Repeated conversions for multiple channels, beginning with the channel selected by INCHx bits and decrementing to channel A0. Each ADC result is written to ADC10MEM. The sequence ends after conversion of channel A0, and the next trigger signal re-starts the sequence.

ADC10 (1/2)
Data Transfer Controller (DTC)

- **DTC (ADC10DTC1 ≠ 0):**
  - Automatically transfers the conversion results from ADC10MEM to other on-chip memory locations each time the ADC10 completes a conversion and loads the result to ADC10MEM.

- **Requires one CPU MCLK:**
  - If the CPU is active during this period, it will be halted to ensure the transfer is completed;
  - Ensure that no active conversion or sequence is in progress (ADC10 busy) during DTC transfer initiation.

ADC10 (2/2)
Data Transfer Controller (DTC)

- **The Data Transfer Controller (DTC) can be configured for:**
  - ***One-Block Transfer Mode*** (ADC10TB = 0):
    - The value *n* in ADC10DTC1 defines the total number of transfers for a block;
    - First block address range

      {Start: ADC10SA; End: ADC10SA+2n–2};

  - ***Two-Block Transfer Mode*** (ADC10TB = 1):
    - The value *n* in ADC10DTC1 defines the number of transfers for one block;
    - First block address range

      {Start: ADC10SA ; End: ADC10SA+2n–2};
    - Second block address range:

      {Start: SA+2n ; End: SA+4n–2}.

ADC10
Integrated temperature sensor

- **Input channel selected as INCHx = 1010;**

- **Transfer function relating the input voltage, $V_{Temperature}$ [V] to the temperature, T [ºC], is given by:**

$$V_{Temperature} = 0.00355 \times T + 0.986$$

- **Considerations:**
  - The sampling period must be greater than 30 µs;
  - Large offset error, must be calibrated;
  - Automatically turns on the on-chip reference generator.

ADC10
ADC10 interrupts

- **One interrupt and one interrupt vector are associated with the ADC10 function:**
  - 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 $n = 0$.
- **When ADC10IE = 1 and GIE = 1, the ADC10IFG flag generates an interrupt request.**

ADC10 (1/7)
Registers

❑ ADC10CTL0, ADC10 Control Register 0 (high byte)

15	14	13	12	11	10	9	8
SREFx			ADC10SHTx		ADC10SR	REFOUT	REFBURST

Bit		Description		
15-13	SREFx	Select voltage reference: SREF2 SREF1 SREF0 = 000 ⇒ SREF2 SREF1 SREF0 = 001 ⇒ SREF2 SREF1 SREF0 = 010 ⇒ SREF2 SREF1 SREF0 = 011 ⇒ SREF2 SREF1 SREF0 = 100 ⇒ SREF2 SREF1 SREF0 = 101 ⇒ SREF2 SREF1 SREF0 = 110 ⇒ SREF2 SREF1 SREF0 = 111 ⇒	**V_{R+}** V_{CC} V_{REF+} Ve_{REF+} Buffered Ve_{REF+} V_{CC} V_{REF+} Ve_{REF+} Buffered Ve_{REF+}	**V_{R-}** V_{SS} V_{SS} V_{SS} V_{SS} V_{REF-}/Ve_{REF-} V_{REF-}/Ve_{REF-} V_{REF-}/Ve_{REF-} V_{REF-}/Ve_{REF-}
12-11	ADC10SHTx	ADC10 sample-and-hold time: ADC10SHT1 ADC10SHT0 = 00 ⇒ 4 x ADC10CLKs ADC10SHT1 ADC10SHT0 = 01 ⇒ 8 x ADC10CLKs ADC10SHT1 ADC10SHT0 = 10 ⇒16 x ADC10CLKs ADC10SHT1 ADC10SHT0 = 11 ⇒ 64 x ADC10CLKs		
10	ADC10SR	ADC10 sampling rate: ADC10SR = 0 ⇒ Reference buffer supports up to ~200 ksps ADC10SR = 1 ⇒ Reference buffer supports up to ~50 ksps		
9	REFOUT	Reference voltage output (pin V_{REF+}): REFOUT = 0 ⇒ Disable REFOUT = 1 ⇒ Enable		
8	REFBURST	Controls the operation of the internal reference buffer: REFBURST = 0 ⇒ Reference buffer on continuously allowing the reference voltage to be present outside the device continuously. REFBURST = 1 ⇒ Reference buffer automatically disabled when the ADC10 is not actively converting, and automatically re-enabled when during sample-and-conversion.		

ADC10 (2/7)
Registers

❑ ADC10CTL0, ADC10 Control Register 0 (low byte)

7	6	5	4	3	2	1	0
MSC	REF2_5V	REFON	ADC10ON	ADC10IE	ADC10IFG	ENC	ADC10SC

Bit		Description
7	MSC	Multiple sample and conversion (Valid for sequence or repeated modes): MSC = 0 ⇒ Requires a rising edge of the SHI signal to trigger each sample-and-conversion. MSC = 1 ⇒ After the first rising edge of the SHI signal that triggers the sampling timer the further sample-and-conversions are performed automatically as soon as the prior conversion is completed
6	REF2_5V	Reference-generator voltage select (REFON bit must also be set): REF2_5V = 0 ⇒ Reference voltage = 1.5 V REF2_5V = 1 ⇒ Reference voltage = 2.5 V
5	REFON	Reference generator: REFON = 0 ⇒ Reference generator disable REFON = 1 ⇒ Reference generator enable
4	ADC10ON	ADC10 on: ADC10ON = 0 ⇒ ADC10 off ADC10ON = 1 ⇒ ADC10 on
3	ADC10IE	ADC10 interrupt enable ADC10IE = 0 ⇒ Interrupt disabled ADC10IE = 1 ⇒ Interrupt enabled
2	ADC10IFG	ADC10 interrupt flag: ADC10IFG = 0 ⇒ No interrupt pending (interrupt request is accepted, or it may be reset by software) ADC10IFG = 1 ⇒ Interrupt pending (ADC10MEM is loaded with a conversion result or when a block of DTC transfers is completed)
1	ENC	Enable conversion: ENC = 0 ⇒ ADC10 disabled ENC = 1 ⇒ ADC10 enabled
0	ADC10SC	Start conversion: ADC10SC = 0 ⇒ No sample-and-conversion start ADC10SC = 1 ⇒ Start sample-and-conversion

ADC10 (3/7)
Registers

❑ ADC10CTL1, ADC10 Control Register 1 (high byte)

15	14	13	12	11	10	9	8
INCHx				SHSx		ADC10DF	ISSH

Bit		Description
15–12	INCHx	Input channel select: INCH3 INCH2 INCH1 INCH0 = 0000 ⇒ A0 INCH3 INCH2 INCH1 INCH0 = 0001 ⇒ A1 INCH3 INCH2 INCH1 INCH0 = 0010 ⇒ A2 INCH3 INCH2 INCH1 INCH0 = 0011 ⇒ A3 INCH3 INCH2 INCH1 INCH0 = 0100 ⇒ A4 INCH3 INCH2 INCH1 INCH0 = 0101 ⇒ A5 INCH3 INCH2 INCH1 INCH0 = 0110 ⇒ A6 INCH3 INCH2 INCH1 INCH0 = 0111 ⇒ A7 INCH3 INCH2 INCH1 INCH0 = 1000 ⇒ Ve_{REF+} INCH3 INCH2 INCH1 INCH0 = 1001 ⇒ V_{REF-}/Ve_{REF-} INCH3 INCH2 INCH1 INCH0 = 1010 ⇒ Temperature sensor INCH3 INCH2 INCH1 INCH0 = 1011 ⇒ $(V_{CC} - V_{SS})/2$ INCH3 INCH2 INCH1 INCH0 = 1100 ⇒ $(V_{CC} - V_{SS})/2$ or A12* INCH3 INCH2 INCH1 INCH0 = 1101 ⇒ $(V_{CC} - V_{SS})/2$ or A13 * INCH3 INCH2 INCH1 INCH0 = 1110 ⇒ $(V_{CC} - V_{SS})/2$ or A14 * INCH3 INCH2 INCH1 INCH0 = 1111 ⇒ $(V_{CC} - V_{SS})/2$ or A15 * * on MSP430x22xx devices
11–10	SHSx	Sample-and-hold source: SHS1 SHS0 = 00 ⇒ bit ADC10SC SHS1 SHS0 = 01 ⇒ TIMER_A Output Unit 1 SHS1 SHS0 = 10 ⇒ TIMER_A Output Unit 0 SHS1 SHS0 = 11 ⇒ TIMER_A Output Unit 2
9	ADC10DF	ADC10 data format: ADC10DF = 0 ⇒ Binary ADC10DF = 1 ⇒ Two's complement
8	ISSH	Invert signal sample-and-hold ISSH = 0 ⇒ The sample-input signal is not inverted ISSH = 1 ⇒ The sample-input signal is inverted

ADC10 (4/7)
Registers

❑ ADC10CTL1, ADC10 Control Register 1 (low byte)

7	6	5	4	3	2	1	0
ADC10DIVx			ADC10SSELx		CONSEQx		ADC10BUSY

Bit		Description
7–5	ADC10DIVx	ADC10 clock divider: ADC10DIV2 ADC10DIV1 ADC10DIV0 = 000 ⇒ / 1 ADC10DIV2 ADC10DIV1 ADC10DIV0 = 001 ⇒ / 2 ADC10DIV2 ADC10DIV1 ADC10DIV0 = 010 ⇒ / 3 ADC10DIV2 ADC10DIV1 ADC10DIV0 = 011 ⇒ / 4 ADC10DIV2 ADC10DIV1 ADC10DIV0 = 100 ⇒ / 5 ADC10DIV2 ADC10DIV1 ADC10DIV0 = 101 ⇒ / 6 ADC10DIV2 ADC10DIV1 ADC10DIV0 = 110 ⇒ / 7 ADC10DIV2 ADC10DIV1 ADC10DIV0 = 111 ⇒ / 8
4–3	ADC10SSELx	ADC10 clock source: ADC10SSEL1 ADC10SSEL0 = 00 ⇒ ADC10OSC ADC10SSEL1 ADC10SSEL0 = 01 ⇒ ACLK ADC10SSEL1 ADC10SSEL0 = 10 ⇒ MCLK ADC10SSEL1 ADC10SSEL0 = 11 ⇒ SMCLK
2–1	CONSEQx	Conversion sequence mode: CONSEQ1 CONSEQ0 = 00 ⇒ Single-channel, single-conversion CONSEQ1 CONSEQ0 = 01 ⇒ Sequence-of-channels CONSEQ1 CONSEQ0 = 10 ⇒ Repeat-single-channel CONSEQ1 CONSEQ0 = 11 ⇒ Repeat-sequence-of-channel
0	ADC10BUSY	ADC10 busy: ADC10BUSY = 0 ⇒ No operation is active ADC10BUSY = 1 ⇒ Sequence, sample, or conversion is active

ADC10 (5/7)
Registers

- **ADC10AE0, Analogue (Input) Enable Control Register 0**
  - Enables the analogue input of the ADC10: BIT0 => A0, BIT1 => A1, and so on.

- **ADC10AE1, Analogue (Input) Enable Control Register 1 (‘F2274)**
  - Additional analogue input enable control register. BIT4 => A12, BIT5 => A13, BIT6 => A14, and BIT7 => A15.

- **ADC10MEM, Conversion-Memory Register**
  - Loaded with the conversion results;
  - Numerical result format:
    - Binary: Bits 15-10 = 0. The results in the least significant 10 bits.
    - 2’s complement: The results in the most significant 10 bits. Bits 5-0 = 0.

ADC10 (6/7)
Registers

❑ ADC10DTC0, Data Transfer Control Register 0

7	6	5	4	3	2	1	0
Reserved				ADC10TB	ADC10CT	ADC10B1	ADC10FETCH

Bit		Description
3	ADC10TB	ADC10 block mode: ADC10TB = 0 ⇒ One-block transfer mode ADC10TB = 1 ⇒ Two-block transfer mode
2	ADC10CT	ADC10 continuous transfer ADC10CT = 0 ⇒ Data transfer stops when a block(s) transfer is completed ADC10CT = 1 ⇒ Data is transferred continuously
1	ADC10B1	block filled with ADC10 conversion results (two-block mode): ADC10B1 = 0 ⇒ Block 2 is filled ADC10B1 = 1 ⇒ Block 1 is filled
0	ADC10FETCH	Normally set ADC10FETCH = 0

ADC10 (7/7)
Registers

- **ADC10DTC1, Data Transfer Control Register 1**
  - This 8-bit register defines the number of transfers for each block;
  - ADC10DTC1 = 0 ⇒ DTC is disabled;
  - ADC10DTC1 = 01h–0FFh ⇒ Number of transfers per block.

- **ADC10SA, Start Address Register for Data Transfer**
  - This 16-bit register defines the ADC10 start address for the DTC. It uses only the 15 most significant bits. Bit 0 is always read as 0.

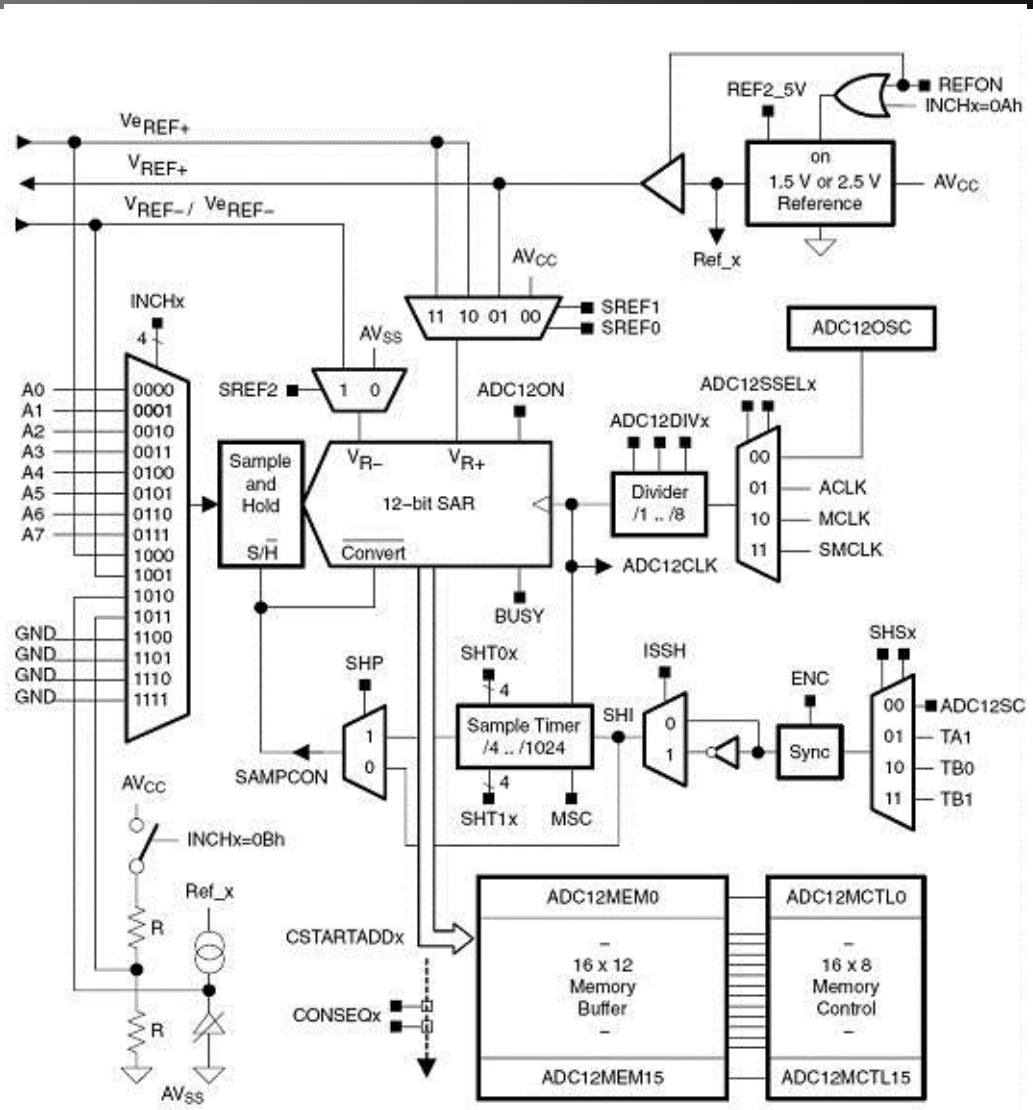
ADC12 (1/2)
Introduction

- **The ADC12 module of the MSP430F2013 supports fast 12-bit analogue-to-digital conversions;**

- **The module contains:**
  - 12-bit SAR core;
  - Sample select control;
  - Reference current generator.

ADC12 (2/2)
Introduction

- **ADC12 block diagram:**



ADC12
ADC12 Features

- **It has same basic features as the ADC10, with the following differences:**
  - Monotonic 12-bit converter with no missing codes;
  - Interrupt vector register for fast decoding of 18 ADC interrupts;
  - Registers for storage of 16 conversion results;
  - No Data Transfer Controller (DTC);
  - 16 control registers ADC12MCTLx for free choice of channels on sequential modes;
  - Can also convert some channels more than once in one loop (e.g. placing two measurements of the same voltage and one measurement of current in between to calculate power).

ADC12
12 bit ADC core

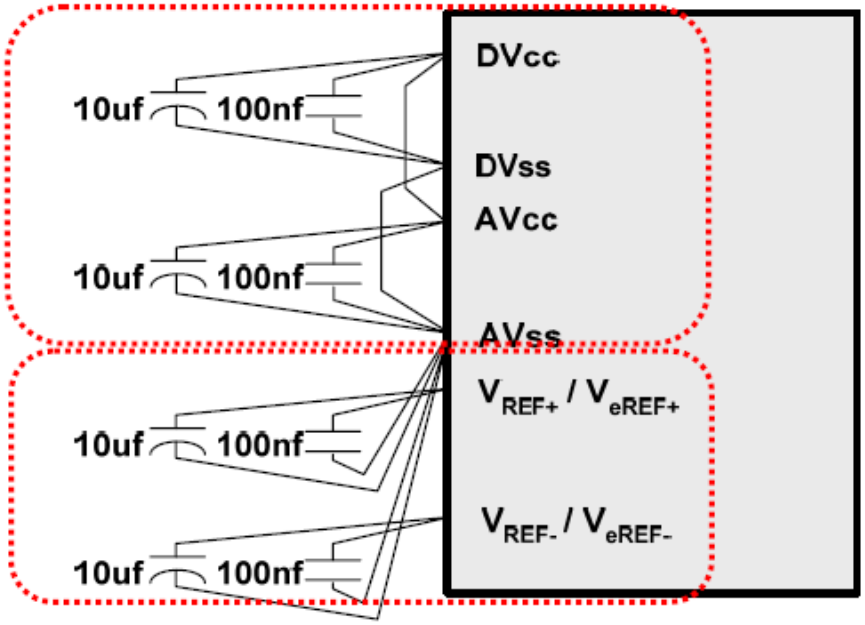
- **12 bit ADC core (enable with ADC12ON bit):**
  - Converts an analogue input to its 12-bit digital representation;
  - Stores the result in a ADC12MEM register.

  - The conversion is limited by the upper and lower limits: V_{R+} ; V_{R-}
  - The digital output (N_{ADC}) is:
    - Full scale: N_{ADC} = 0FFFh, when the input signal ≥ V_{R+};
    - Zero: N_{ADC} = 0000h, when the input signal ≤ V_{R-}.

  - Conversion results:
    - Binary format: $N_{ADC} = 4096 \frac{V_{in} - V_{R-}}{V_{R+} - V_{R-}}$
    - Two's-complement format.

ADC12
Similarities to ADC10

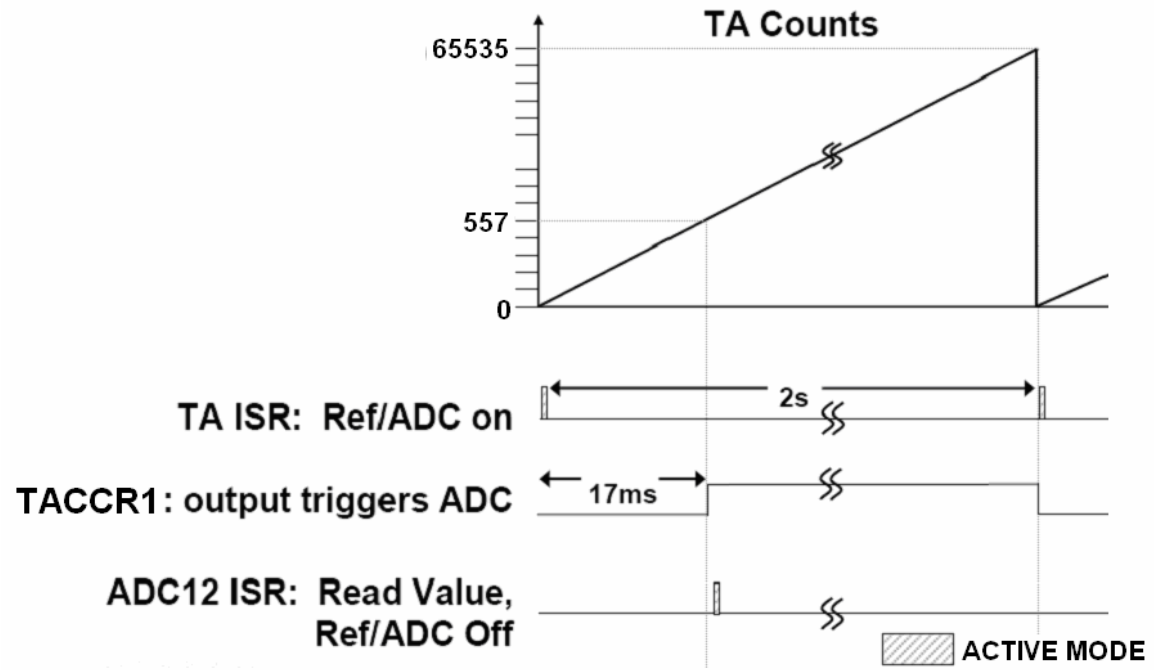
- **Conversion clock selection;**
- **ADC12 inputs and multiplexer;**
- **Analogue port selection (P6);**
  - The ADC12 inputs are multiplexed with the port P6 pins.
- **Voltage reference generator:**
  - For proper operation requires storage capacitors across V_{REF+} and AV_{SS}.
- **Conversion modes;**
- **Integrated temperature sensor**



ADC12 (1/3)
Sample and conversion timing

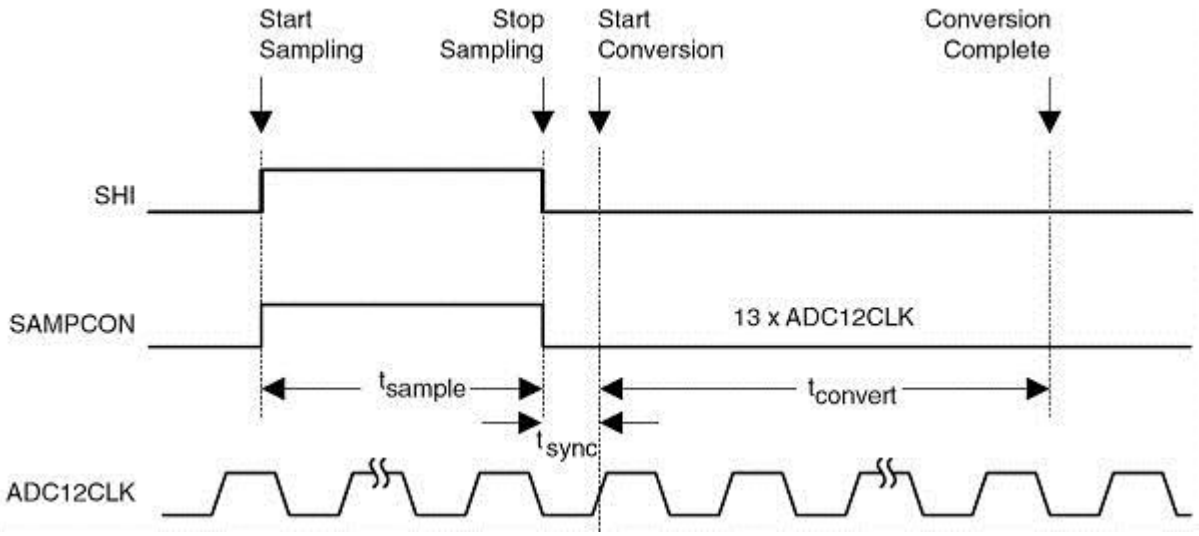
- **An A/D conversion is initiated on the rising edge of SHI. The source for SHI (SHSx bits selection) can be:**
  - ADC12SC bit;
  - Timer_A Output Unit 1; Timer_B Output Unit 0, or ; Output Unit 1.

- **ADC12 timer trigger for reference settling:**



ADC12 (2/3)
Sample and conversion timing

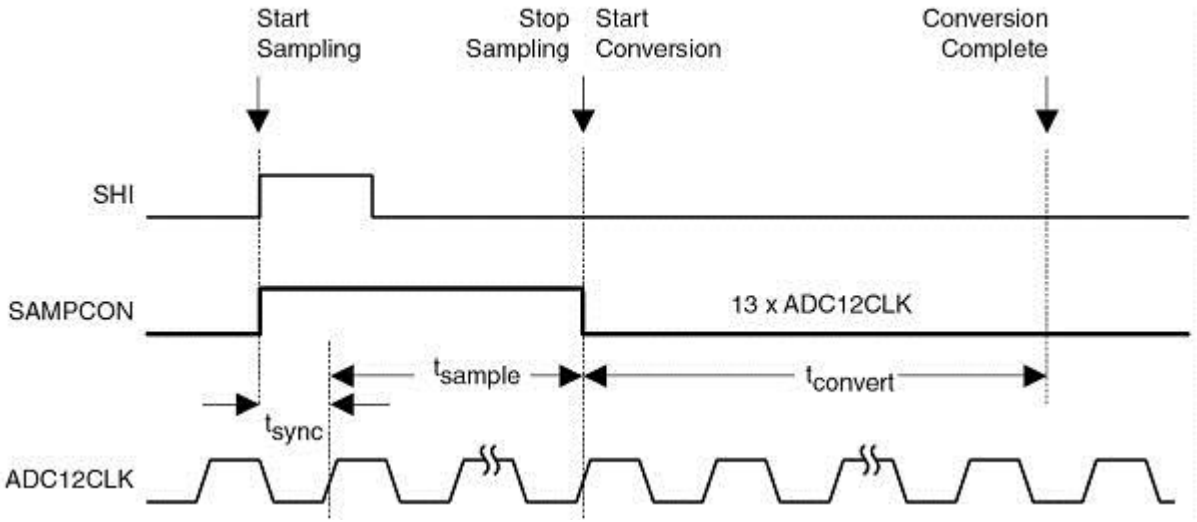
- **Sample-timing methods:**
  - SHP = 0: ***Extended sample mode***:
    - SHI signal directly controls SAMPCON;
    - Defines the length of the sample period *t*sample;
    - SAMPCON = 1 ⇒ sampling is active;
    - High-to-Low SAMPCON transition ⇒ starts the conversion after synchronization with ADC12CLK.



>> Contents

ADC12 (3/3)
Sample and conversion timing

- **Sample-timing methods:**
  - SHP = 1: ***Pulse mode***:
    - SHI signal triggers the sampling timer;
    - SHT0x and SHT1x bits (ADC12CTL0) defines the SAMPCON sample period, t_{sample};
    - The sampling timer keeps SAMPCON = 1 after synchronization with ADC12CLK.



ADC12
Conversion memory

- **16 ADC12MEMx conversion memory registers (configured by the associated ADC12MCTLx control register) to store conversion results.**
- **Non-sequential conversion (single- or repeat-single-channel):**
  - CSTARTADDx define the first and single ADC12MCTLx for conversion.
- **Sequential conversion (sequence-of- or repeat-sequence-of-channels):**
  - A sequence is started by the command found in the ADC12MCTLx register pointed to by CSTARTADDx;
  - The pointer is incremented automatically to the next ADC12MCTLx for the next conversion;
  - After ADC12MCTL15 the next conversion is ADC12MCTL0;
  - The sequence runs until an EOS bit signals that this command is the last conversion of the actual sequence;
  - The 16 ADC12MCTLx registers can contain more than one sequence.

ADC12
ADC12 interrupts

- **The ADC12 has 18 interrupt sources:**
  - ADC12IFG0-ADC12IFG15: ADC12IFGx bits are set when their corresponding ADC12MEMx memory register is loaded with a conversion result;
  - ADC12OV, ADC12MEMx overflow: ADC12OV is set when a conversion result is written to any ADC12MEMx before its previous conversion result was read;
  - ADC12TOV, ADC12 conversion time overflow: ADC12TOV is set when another sample-and-conversion is requested before the current conversion is completed.
- **The DMA is triggered after the conversion in single channel modes or after the completion of sequence−of−channel modes.**

ADC12
ADC12 Interrupt vector generator

- **Interrupt vector register ADC12IV used to determine which enabled ADC12 interrupt source requested an interrupt.**

- **Considerations:**
  - The highest priority enabled interrupt generates a number in the ADC12IV register (evaluated or added to the program counter to automatically call the appropriate routine);
  - Any access, read or write, of the ADC12IV register automatically resets the ADC12OV or the ADC12TOV conditions, if either were the highest pending interrupt;
  - ADC12IFGx bits are reset automatically by accessing their ADC12MEMx register or may be reset by software;
  - If another interrupt is pending after servicing of an interrupt, another interrupt is generated.

ADC12 (1/6)
Registers

ADC12CTL0, ADC12 Control Register 0 (high byte)

15	14	13	12	11	10	9	8
SHT1x				SHT0x			

Bit		Description
15-12	SHT1x	Sample-and-hold time (ADC12CLK cycles in the sampling period for registers ADC12MEM8 to ADC12MEM15): SHT13 SHT12 SHT11 SHT10 = 0000 ⇒ 4 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 0001 ⇒ 8 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 0010 ⇒ 16 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 0011 ⇒ 32 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 0100 ⇒ 64 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 0101 ⇒ 96 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 0110 ⇒ 128 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 0111 ⇒ 192 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 1000 ⇒ 256 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 1001 ⇒ 384 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 1010 ⇒ 512 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 1011 ⇒ 768 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = 1100 ⇒ 1024 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = ⇒ 1101 1024 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = ⇒ 1110 1024 ADC12CLK cycles SHT13 SHT12 SHT11 SHT10 = ⇒ 1111 1024 ADC12CLK cycles
11-8	SHT0x	Sample-and-hold time (ADC12CLK cycles in the sampling period for registers ADC12MEM0 to ADC12MEM7). These bits are configured as the previous ones (SHT1x).

ADC12 (2/6)
Registers

❑ ADC12CTL0, ADC12 Control Register 0 (low byte)

7	6	5	4	3	2	1	0
MSC	**REF2_5V**	**REFON**	**ADC12ON**	ADC12OVIE	ADC12TOVIE	**ENC**	**ADC12SC**

- The **bold** bits have the same function as the ADC10. Refer to the ADC10 to see their description.

Bit		Description
3	ADC12OVIE	ADC12MEMx overflow-interrupt enable (The GIE bit must also be set to enable the interrupt): ADC12OVIE = 0 ⇒ Overflow interrupt disabled ADC12OVIE = 1 ⇒ Overflow interrupt enabled
2	ADC12TOVIE	ADC12 conversion-time-overflow interrupt enable (The GIE bit must also be set to enable the interrupt): ADC12TOVIE = 0 ⇒ Conversion time overflow interrupt disabled ADC12TOVIE = 1 ⇒ Conversion time overflow interrupt enabled

ADC12 (3/6)
Registers

❑ ADC12CTL1, ADC12 Control Register 1

15	14	13	12	11	10	9	8
CSTARTADDx				**SHSx**		SHP	**ISSH**

7	6	5	4	3	2	1	0
ADC12DIVx			**ADC12SSELx**		**CONSEQx**		**ADC12BUSY**

- The **bold** bits have the same funciton as the ADC10. Refer to the ADC10 to see their description.

Bit		Description
15–12	CSTARTADDx	Conversion start address. These bits select which ADC12MEMx 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.
9	SHP	Sample-and-hold mode select: SHP = 0 ⇒ SAMPCON signal is sourced from the sample-input signal SHP = 1 ⇒ SAMPCON signal is sourced from the sampling timer

ADC12 (4/6)
Registers

ADC12MEMx, Conversion-Memory Register

- Loaded with the conversion results. Bits 15-12 are always 0. The results are stored in the least significant 12 bits.

ADC12MCTLx, ADC12 Conversion Memory Control Registers

7	6	5	4	3	2	1	0
EOS	SREFx			INCHx			

Bit		Description		
7	EOS	Indicates the last conversion in a sequence: EOS = 0 ⇒ Not end of sequence EOS = 1 ⇒ End of sequence		
6-4	SREFx	Select voltage reference:	**V_{R+}**	**V_{R-}**
		SREF2 SREF1 SREF0 = 000 ⇒	AV_{CC}	AV_{SS}
		SREF2 SREF1 SREF0 = 001 ⇒	V_{REF+}	AV_{SS}
		SREF2 SREF1 SREF0 = 010 ⇒	Ve_{REF+}	AV_{SS}
		SREF2 SREF1 SREF0 = 011 ⇒	Ve_{REF+}	AV_{SS}
		SREF2 SREF1 SREF0 = 100 ⇒	AV_{CC}	V_{REF-}/Ve_{REF-}
		SREF2 SREF1 SREF0 = 101 ⇒	V_{REF+}	V_{REF-}/Ve_{REF-}
		SREF2 SREF1 SREF0 = 110 ⇒	Ve_{REF+}	V_{REF-}/Ve_{REF-}
		SREF2 SREF1 SREF0 = 111 ⇒	Ve_{REF+}	V_{REF-}/Ve_{REF-}

ADC12 (5/6)
Registers

❑ ADC12MCTLx, ADC12 Conversion Memory Control Registers (INCHx depends on the device)

7	6	5	4	3	2	1	0
EOS	SREFx			INCHx			

Bit		Description
7	EOS	Indicates the last conversion in a sequence: EOS = 0 ⇒ Not end of sequence EOS = 1 ⇒ End of sequence
6-4	SREFx	Select voltage reference: **V_{R+}** **V_{R-}** SREF2 SREF1 SREF0 = 000 ⇒ AV_{CC} AV_{SS} SREF2 SREF1 SREF0 = 001 ⇒ V_{REF+} AV_{SS} SREF2 SREF1 SREF0 = 010 ⇒ Ve_{REF+} AV_{SS} SREF2 SREF1 SREF0 = 011 ⇒ Ve_{REF+} AV_{SS} SREF2 SREF1 SREF0 = 100 ⇒ AV_{CC} V_{REF-}/Ve_{REF-} SREF2 SREF1 SREF0 = 101 ⇒ V_{REF+} V_{REF-}/Ve_{REF-} SREF2 SREF1 SREF0 = 110 ⇒ Ve_{REF+} V_{REF-}/Ve_{REF-} SREF2 SREF1 SREF0 = 111 ⇒ Ve_{REF+} V_{REF-}/Ve_{REF-}
3-0	INCHx	Input channel select: INCH3 INCH2 INCH1 INCH0 = 0000 ⇒ A0 INCH3 INCH2 INCH1 INCH0 = 0001 ⇒ A1 INCH3 INCH2 INCH1 INCH0 = 0010 ⇒ A2 INCH3 INCH2 INCH1 INCH0 = 0011 ⇒ A3 INCH3 INCH2 INCH1 INCH0 = 0100 ⇒ A4 INCH3 INCH2 INCH1 INCH0 = 0101 ⇒ A5 INCH3 INCH2 INCH1 INCH0 = 0110 ⇒ A6 INCH3 INCH2 INCH1 INCH0 = 0111 ⇒ A7 INCH3 INCH2 INCH1 INCH0 = 1000 ⇒ Ve_{REF+} INCH3 INCH2 INCH1 INCH0 = 1001 ⇒ V_{REF-}/Ve_{REF-} INCH3 INCH2 INCH1 INCH0 = 1010 ⇒ Temperature sensor INCH3 INCH2 INCH1 INCH0 = 1011 ⇒ $(AV_{CC} - AV_{SS})/2$ INCH3 INCH2 INCH1 INCH0 = 1100 ⇒ A12 INCH3 INCH2 INCH1 INCH0 = 1101 ⇒ A13 INCH3 INCH2 INCH1 INCH0 = 1110 ⇒ A14 INCH3 INCH2 INCH1 INCH0 = 1111 ⇒ A15

ADC12 (6/6)
Registers

- **ADC12IE, ADC12 Interrupt Enable Register**
  - This 16-bit register enables (ADC12IEx = 1) or disables (ADC12IEx = 0), the interrupt request for the ADC12IFGx bits.

- **ADC12IFG, ADC12 Interrupt Flag Register**
  - Each bit of this 16-bit register is set when the corresponding ADC12MEMx is loaded with a conversion result and reset if the corresponding ADC12MEMx is accessed by software.