One of the most important interfaces between the microcontroller and the real word is the Analog-to-DigitalConverter (ADC).

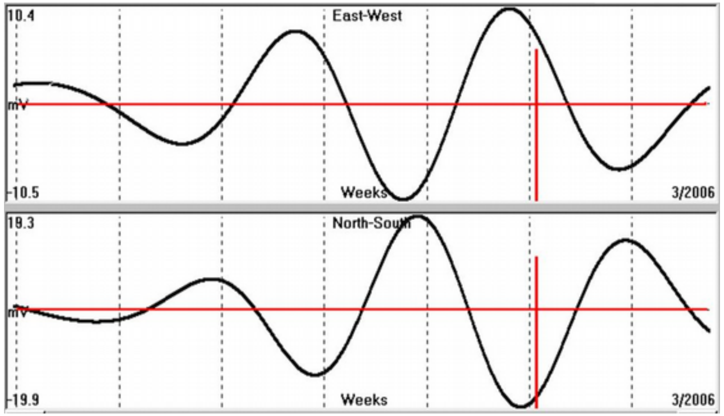
This allows a digital representation of a physical signal to be measured, usually an electrical signal and measured in volts.

Typically, the low amplitude of most analogue signals representing physical quantities, such as temperature, humidity, pressure, velocity among others, require some form of signal conditioning.

The first stage in this process is often amplification of the analogue signal.

REAL WORLD SENSING

RAW SIGNAL - SENSOR



AMPLIFIER - FILTER

VOLTMETER - ADC

Analog-to-Digital Converter

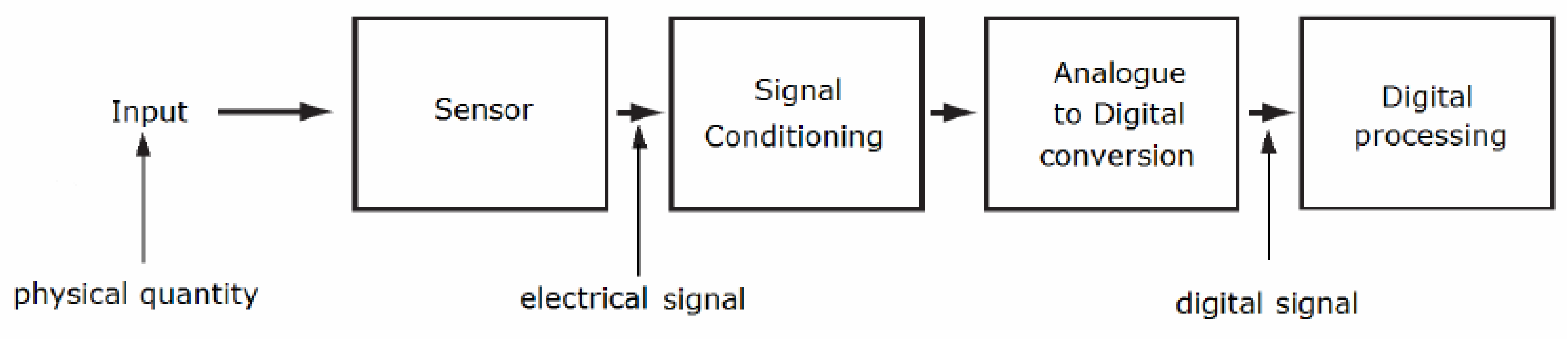
‘Voltmeter’ for Computer

To convert an analog signal to a digital value, it is necessary to use an ADC.

The Successive Approximation Register (SAR) converter uses a technique which determines the digital value (bits) by approximating the input signal using an iterative process.

Computer analog data acquisition block diagram

Figure 9-1. Data acquisition block diagram.



The analog world (the real one), interfaces with the digital systems through an ADC. (a voltmeter for the computer)

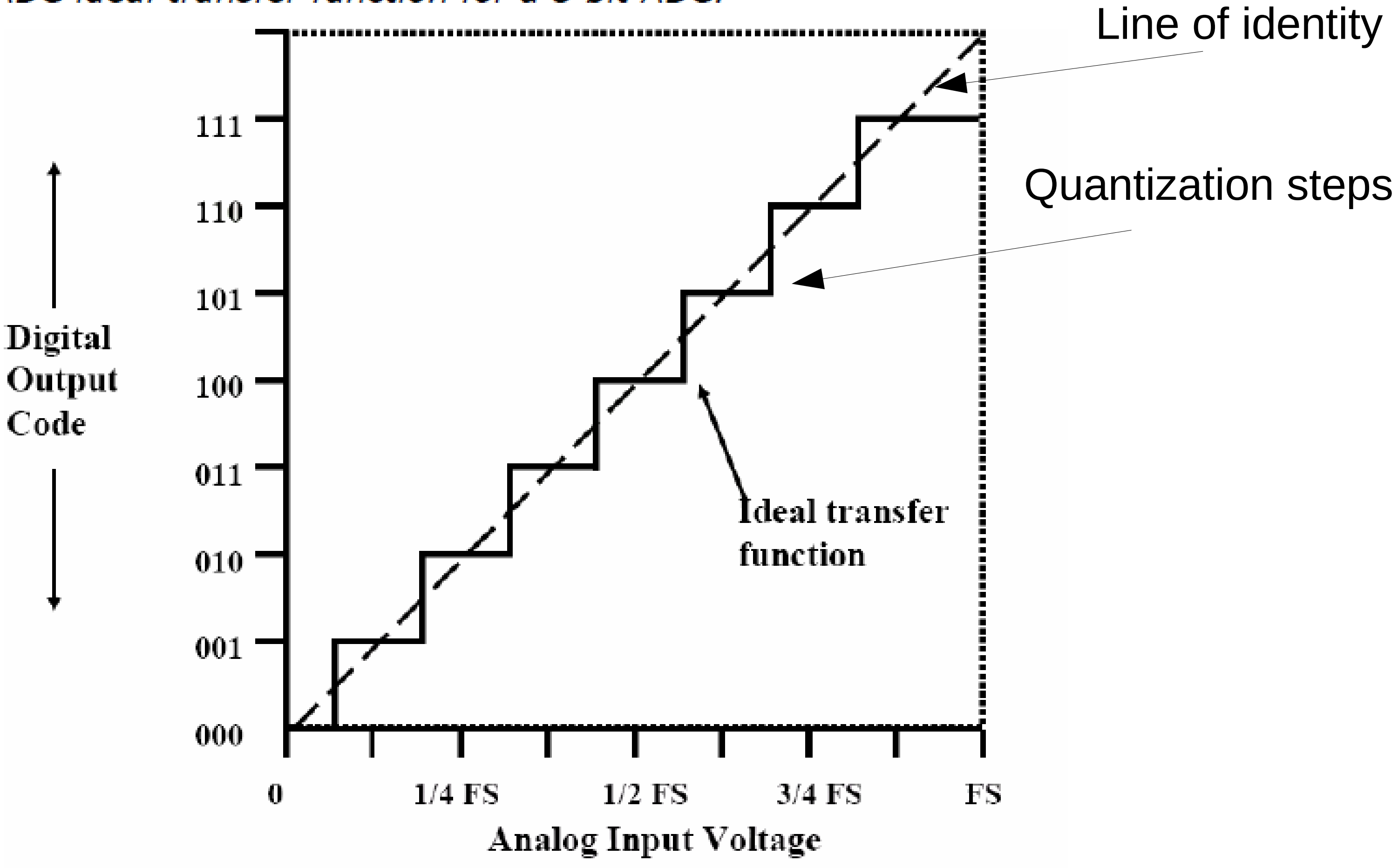
The ADC takes the input voltage from a transducer (after signal conditioning), and converts it to an equivalent digital value.

All analog voltages between zero and full scale of the ADC become quantized, by dividing the range of voltage into sub-ranges. If *FS* is the full-scale analog voltage, the quantization increment is given by *FS* x *LSB*, where *LSB* = 2-*n*, where *n* is the number of bits of the ADC.

The quantization process, which replaces a linear analogue function with a staircase digital representation, results in a quantization uncertainty of +/- 0.5 LSB and a quantization error.

10 bits ADC has 2^10 or 1024 combinations. With a 3 volt reference, each bit value equals: 3.00 V/ 1024 combinations or almost 3mV per bit (2.9296875mV)

Figure 9-20. ADC ideal transfer function for a 3 bit ADC.



Accuracy

1) The resolution, R, of an ADC is the smallest analogue voltage that can be converted into a digital code,

2) The accuracy is the degree of conformity of a digital code to it actual (true) analogue voltage.

3) DNL (non-linearity) reveals how far an output code is from a neighboring output code.

4) Offset error Offset error shifts the transfer function vertically, but does not reduce the number of available codes

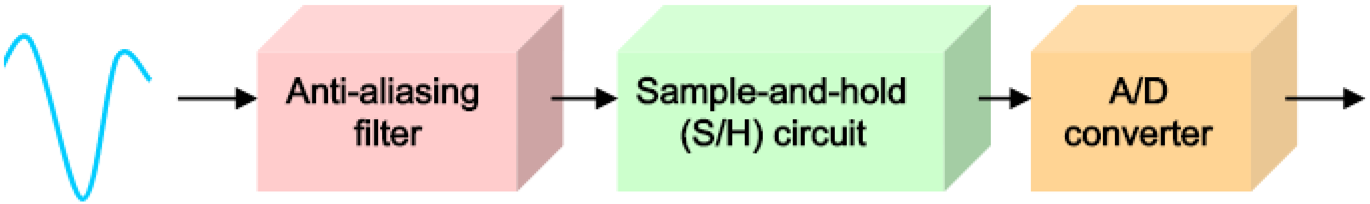
5) The gain error is given by the full-scale error, minus the offset error.

6) SNR is the signal-to-noise ratio without distortion components. SNR reveals where the average noise floor of the converter is, and sets the ADC performance limit for noise

- A modern A/D converter is a single-chip IC

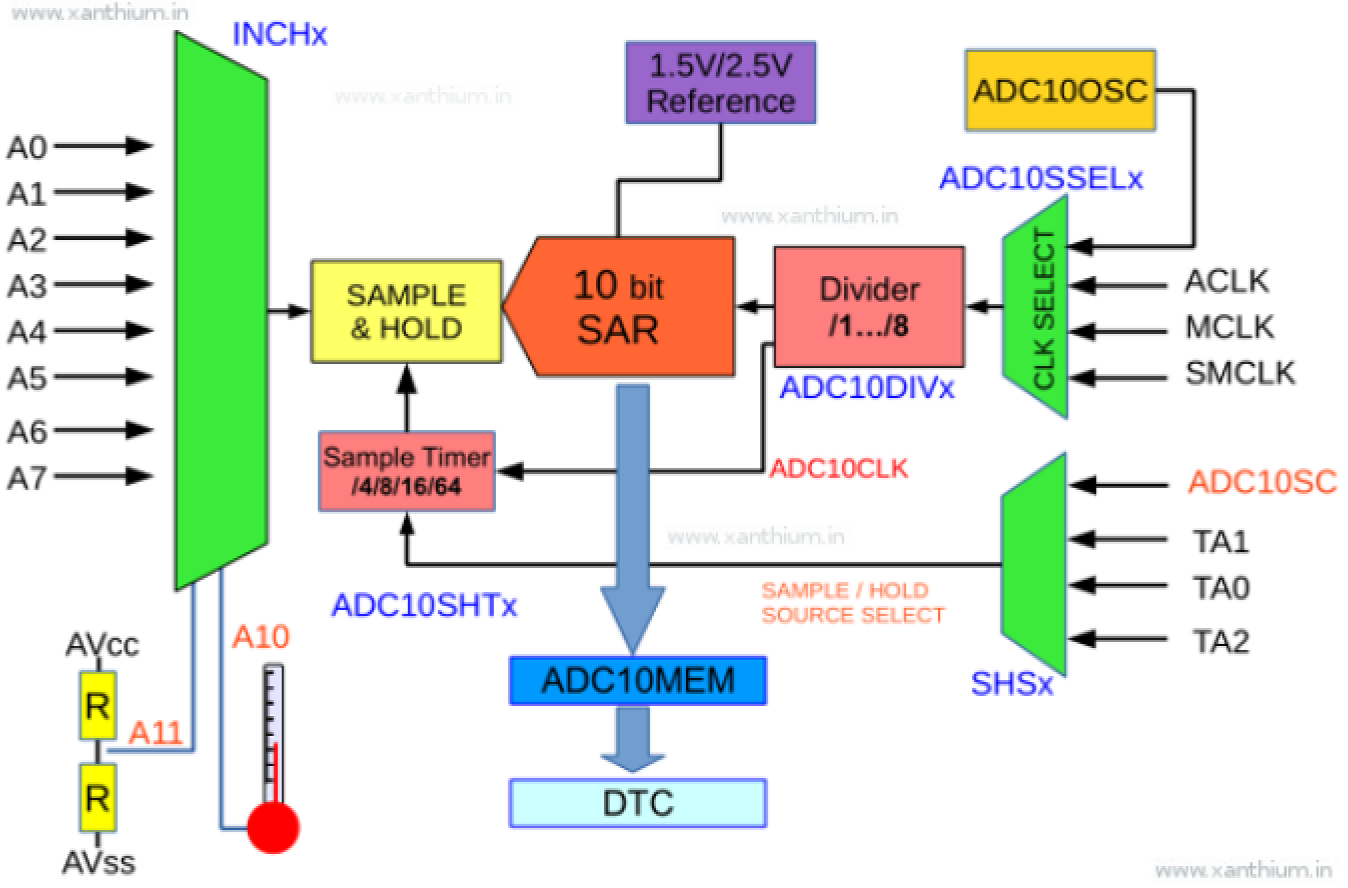
Now inside MicroProcessor

which performs the following processes.

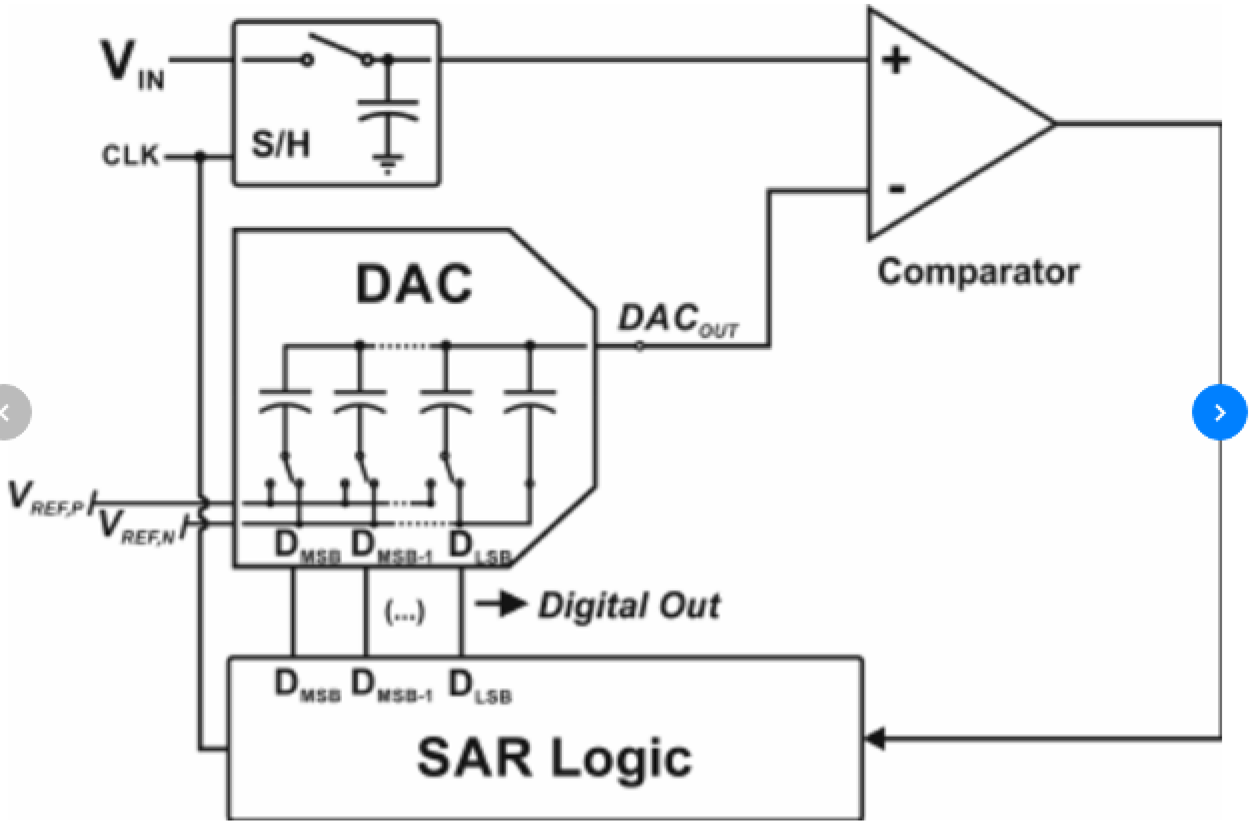


http://slideplayer.com/slide/4597523/

Block Diagram of ADC10



Capacitor logic to produce half voltages for SAR sequence



Generic SAR ADC architecture with a capacitive DAC in the feedback path.

Figure 9-34. SAR ADC block diagram.

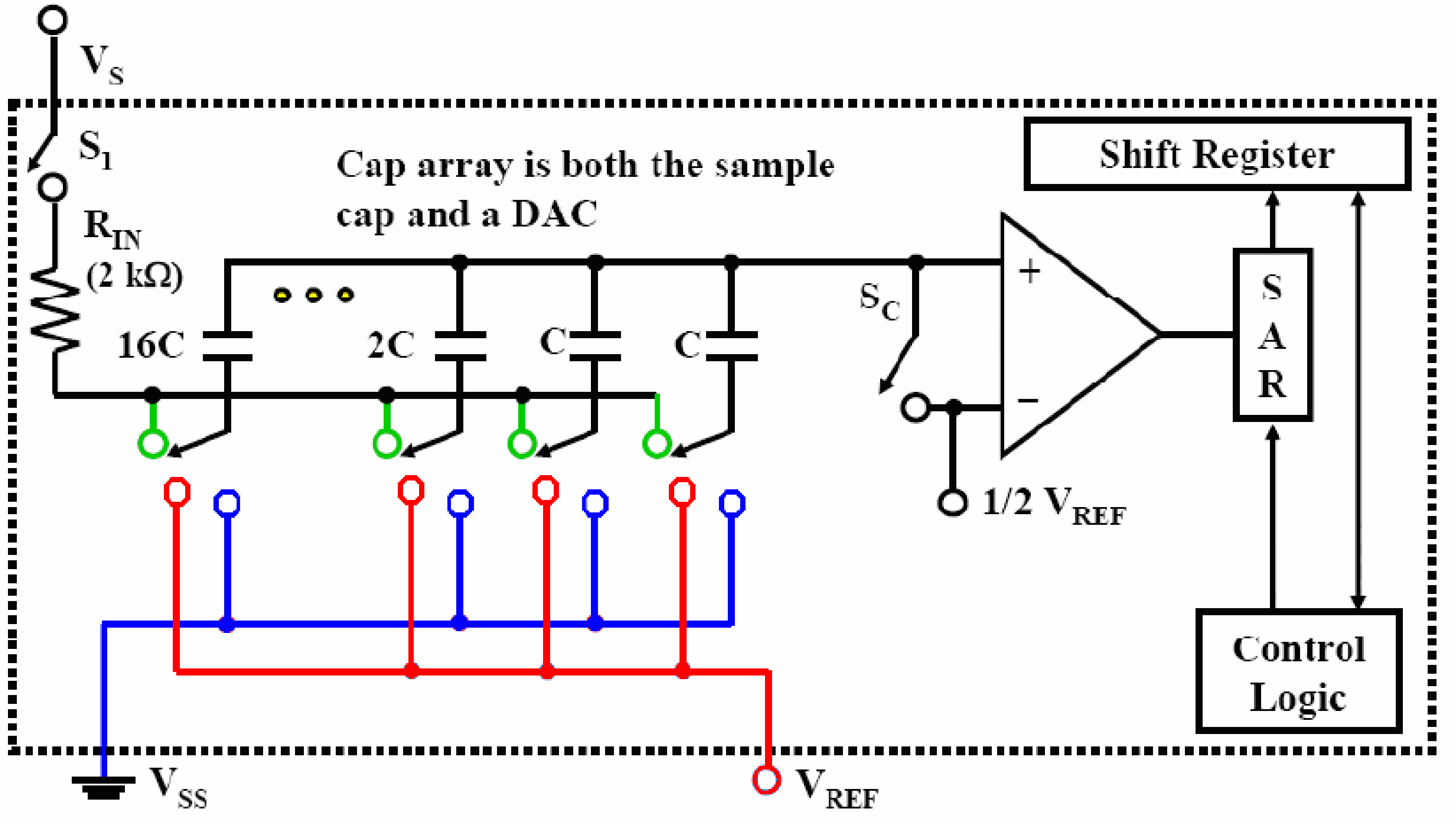
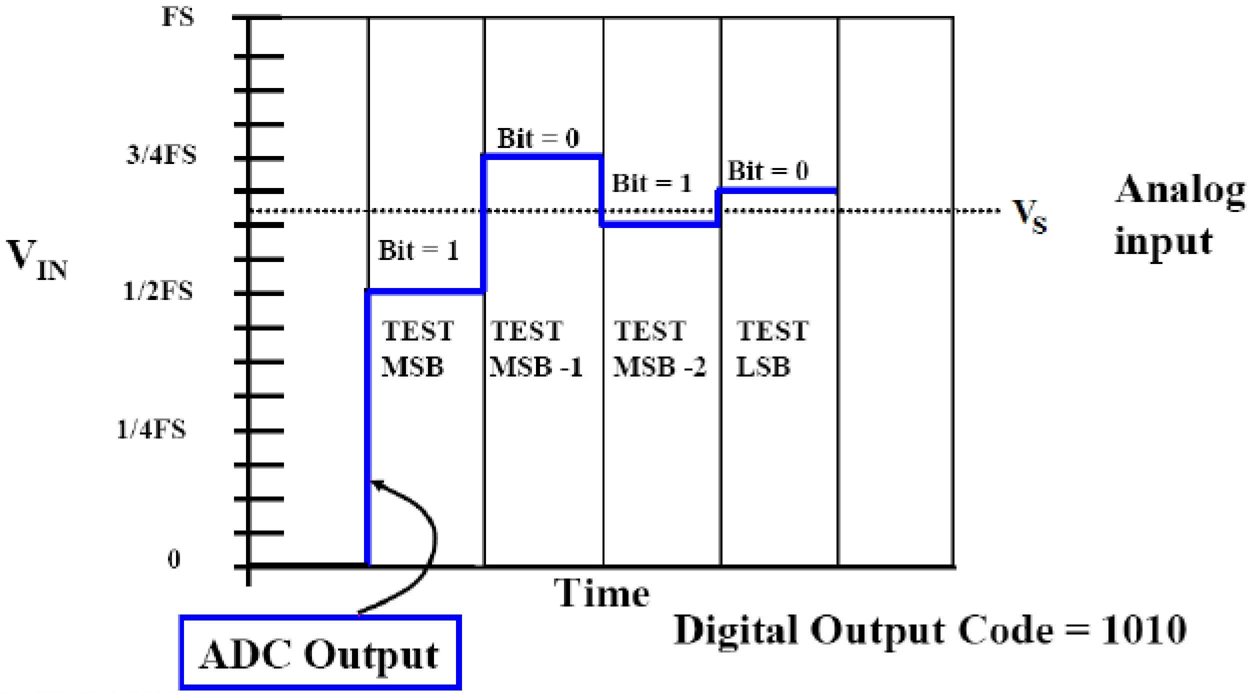


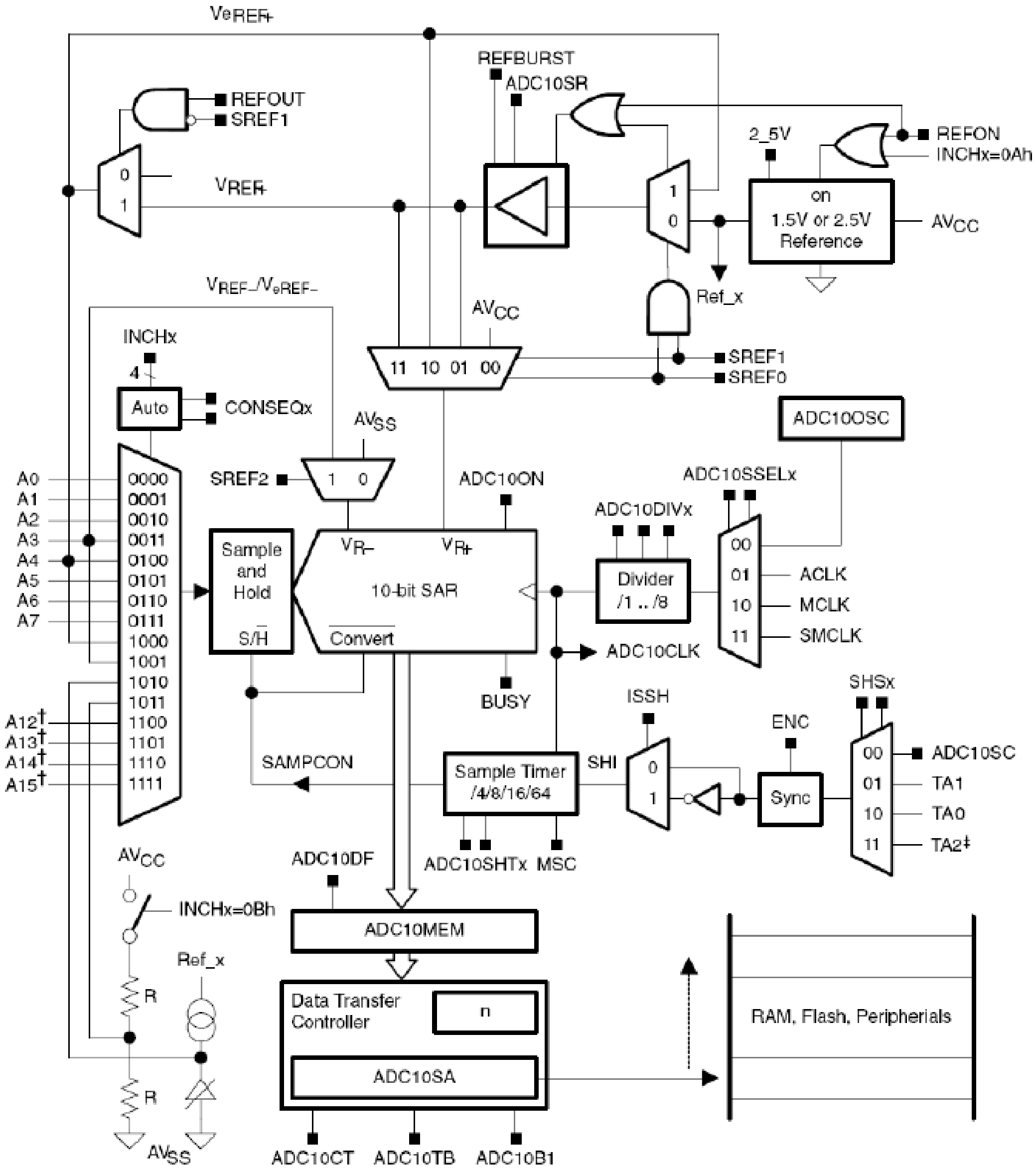
Figure 9-35. SAR analogue-to-digital conversion concept.



ADC DAC Value for successive bit tests

ADC10 Block Diagram

MSPFR2433 Family User Guide
Page 532



ADC10 Registers

The ADC10 module on the MSP430G devices is configured by the user through software. There are various registers used to change the way the ADC10 operates dependent on the application requirements.

Table 20-2. ADC Registers

Offset	Acronym	Register Name	Type	Reset	Section
00h	ADCCTL0	ADC Control 0 register	Read/write	0100h	Section 20.3.1
02h	ADCCTL1	ADC Control 1 register	Read/write	0000h	Section 20.3.2
04h	ADCCTL2	ADC Control 2 register	Read/write	0010h	Section 20.3.3
06h	ADCLO	ADC Window Comparator Low Threshold register	Read/write	0000h	Section 20.3.9
08h	ADCHI	ADC Window Comparator High Threshold register	Read/write	03FFh	Section 20.3.7
0Ah	ADCMCTL0	ADC Memory Control register	Read/write	00h	Section 20.3.6
12h	ADCMEM0	ADC Conversion Memory register	Read/write	undefined	Section 20.3.4
1Ah	ADCIE	ADC Interrupt Enable register	Read/write	0000h	Section 20.3.11
1Ch	ADCIFG	ADC Interrupt Flag register	Read/write	0000h	Section 20.3.12
1Eh	ADCIV	ADC Interrupt Vector register	Read/write	0000h	Section 20.3.13

Clock Sources

Table 20-4. ADCCTL1 Register Description (continued)

Bit	Field	Type	Reset	Description
4-3	ADCSSELx	RW	0h	ADC clock source select. Can be modified only when ADCENC = 0. Resetting ADCENC = 0 by software and changing these fields immediately shows an effect when a conversion is active. 00b = MODCLK 01b = ACLK 10b = SMCLK 11b = SMCLK

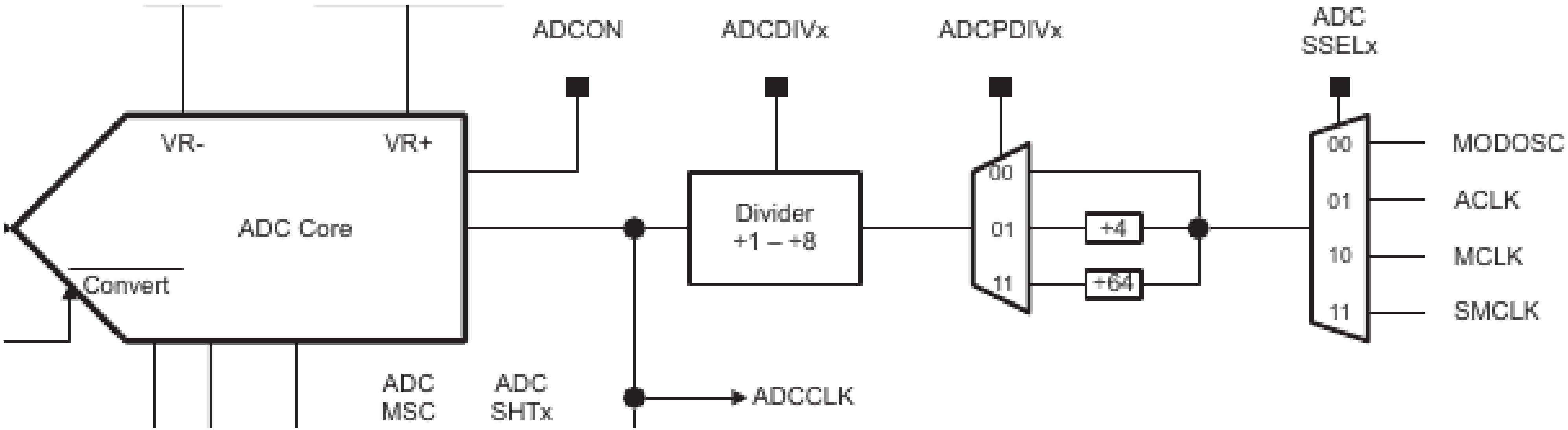


Table 6-15. ADC Channel Connections

ADCSHSx	ADC CHANNELS	EXTERNAL PINOUT
0	A0/Veref+	P1.0
1	A1	P1.1
2	A2/Veref-	P1.2
3	A3	P1.3
4	A4 (1)	P1.4
5	A5	P1.5
6	A6	P1.6
7	A7	P1.7
8	A8	NA
9	A9	NA
10	Not used	N/A
11	Not used	N/A
12	On-chip temperature sensor	N/A
13	Reference voltage (1.5 V)	N/A
14	DVSS	N/A
15	DVCC	N/A

(1) When A4 is used, the PMM 1.2-V reference voltage can be output to this pin by setting the PMM control register. The 1.2-V voltage can be directly measured by A4 channel.

ADC cool: Only 6 lines of code to Set UP ADC for:

Single channel sample

Software triggered conversion

With ISR on conversion completion

Plus one more line to trigger the conversion

Table 1-34. SYSCFG2 Register Description

Bit	Field	Type	Reset	Description
15-8	Reserved	RW	0h	Reserved.
7	ADCPCTL7	RW	0h	ADC input A7 pin select 0b = ADC input A7 disabled 1b = ADC input A7 enabled
6	ADCPCTL6	RW	0h	ADC input A6 pin select 0b = ADC input A6 disabled 1b = ADC input A6 enabled
5	ADCPCTL5	RW	0h	ADC input A5 pin select 0b = ADC input A5 disabled 1b = ADC input A5 enabled
4	ADCPCTL4	RW	0h	ADC input A4 pin select 0b = ADC input A4 disabled 1b = ADC input A4 enabled
3	ADCPCTL3	RW	0h	ADC input A3 pin select 0b = ADC input A3 disabled 1b = ADC input A3 enabled
2	ADCPCTL2	RW	0h	ADC input A2 pin select 0b = ADC input A2 disabled 1b = ADC input A2 enabled
1	ADCPCTL1	RW	0h	ADC input A1 pin select 0b = ADC input A1 disabled 1b = ADC input A1 enabled
0	ADCPCTL0	RW	0h	ADC input A0 pin select 0b = ADC input A0 disabled 1b = ADC input A0 enabled

1. Set input pin:

The ADC pins are controlled by System Configuration Register 2.

SYSCFG2 |= ADCPCTL1

ADC Conversion takes time - Sample Timing

20.2.5.1 Extended Sample Mode

The extended sample mode is selected when ADCSHP = 0. The SHI signal directly controls SAMPCON and defines the length of the sample period t_{sample}. When SAMPCON is high, sampling is active. The higł to-low SAMPCON transition starts the conversion after synchronization with ADCCLK (see 10-bit mode Figure 20-3 or 12-bit mode Figure 20-4). The SHI signal requires at least 4 ADCCLK cycles.

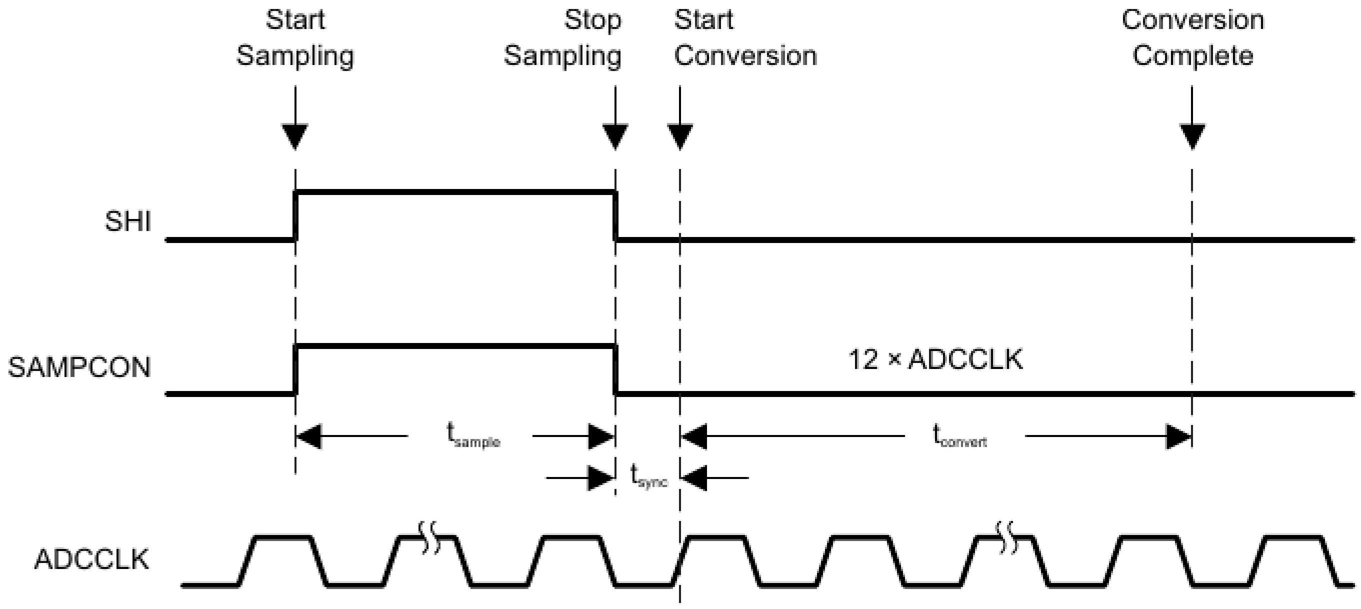


Figure 20-3. Extended Sample Mode in 10-Bit Mode

2. Set ADCCTL0 Register – Sample & Hold time, and turn ADC on

ADCCTL0 |= ADCSHT_2 | ADCON

Table 20-3. ADCCTL0 Register Description

Bit	Field	Type	Reset	Description
15-12	Reserved	R	0h	Reserved. Always reads as 0.
11-8	ADCSHTx	RW	1h	ADC sample-and-hold time. These bits define the number of ADCCLK cycles in the sampling period for the ADC. Can be modified only when ADCENC = 0. Resetting ADCENC = 0 by software and changing these fields immediately shows an effect when a conversion is active. 0000b = 4 ADCCLK cycles 0001b = 8 ADCCLK cycles 0010b = 16 ADCCLK cycles 0011b = 32 ADCCLK cycles 0100b = 64 ADCCLK cycles 0101b = 96 ADCCLK cycles 0110b = 128 ADCCLK cycles
4	ADCON	RW	0h	ADC on. Can be modified only when ADCENC = 0. Resetting ADCENC = 0 by software and changing these fields immediately shows an effect when a conversion is active. 0b = ADC off 1b = ADC on

More to 1024 cycles

3. Set Conversion trigger and timed sampling

ADCCTL1 |= ADCSHP

Two sample-timing methods are selected by control bit ADCSHP: extended sample mode and pulse mode. The pulse sample mode is selected when ADCSHP = 1.

Table 20-4. ADCCTL1 Register Description (continued)

Bit	Field	Type	Reset	Description
4-3	ADCSSELx	RW	0h	ADC clock source select. Can be modified only when ADCENC = 0. Resetting ADCENC = 0 by software and changing these fields immediately shows an effect when a conversion is active. 00b = MODCLK 01b = ACLK 10b = SMCLK
9	ADCSHP	RW	0h	ADC sample-and-hold pulse-mode select. This bit selects the source of the sampling signal (SAMPCON) to be either the output of the sampling timer or the sample-input signal directly. Can be modified only when ADCENC = 0. Resetting ADCENC = 0 by software and changing these fields immediately shows an effect when a conversion is active. 0b = SAMPCON signal is sourced from the sample input signal. 1b = SAMPCON signal is sourced from the sampling timer.

Pulse mode the sample is timed and run by the internal clock source.

Defaule clock source = MODCLK

MODCLK: Internal high-frequency oscillator with 5-MHz typical frequency.

4. set for 10 bit conversion

```
ADCCTL2 |= ADCRES;
```

Table 20-5. ADCCTL2 Register Description

Bit	Field	Type	Reset	Description
5-4	ADCRES	RW	1h	ADC resolution. This bit defines the conversion result resolution.[1] 00b = 8 bit (10 clock cycle conversion time) 01b = 10 bit (12 clock cycle conversion time) 10b = 12 bit (14 clock cycle conversion time) 11b = Reserved

5. Select a Channel (INCH)

ADCMCTL0 |= ADCINCH_1;

20.3.6 ADCMCTL0 Register

ADC Conversion Memory Control Register

Figure 20-23. ADCMCTL0 Register

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

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

Table 20-8. ADCMCTL0 Register Description

Bit	Field	Type	Reset	Description
3-0	ADCINCHx	RW	0h	Input channel select. Writing these bits select the channel for a single-conversion or the highest channel for a sequence of conversions. Reading these bits in ADCCONSEQ = 01,11 returns the channel currently converted.

6-4	ADCSREFx	RW	0h	Select reference. It is not recommended to change this setting while a conversion is ongoing. Can be modified only when ADCENC = 0. Resetting ADCENC = 0 by software and changing these fields immediately shows an effect when a conversion is active. 000b = {V_{R+} = AVCC and V_{R-} = AVSS } 001b = {V_{R+} = VREF and V_{R-} = AVSS} 010b = {V_{R+} = VEREF+ buffered and V_{R-} = AVSS} 011b = {V_{R+} = VEREF+ and V_{R-} = AVSS } 100b = {V_{R+} = AVCC and V_{R-} = VEREF-} 101b = {V_{R+} = VREF and V_{R-} = VEREF-} 110b = {V_{R+} = VEREF+ buffered and V_{R-} = VEREF-} 111b = {V_{R+} = VEREF+ and V_{R-} = VEREF-}

BOR default

Same register: Select Conversion Reference Voltage

ADC10 Interrupts

After the conversion is finished the ADC10 sets the **ADC10IFG** flag and an interrupt is generated. The converted value is available in the **ADC10MEM** register for further processing. The ADC10IFG flag is automatically reset after the interrupt is processed. Please note that for the interrupt to occur the **ADC10IE** flag and **GIE bit** should be set.

6. Enable ADC conversion complete interrupt – ADCIE |= ADCIE0;

Table 20-13. ADCIE Register Description

Bit	Field	Type	Reset	Description
15-6	Reserved	R	0h	Reserved. Always reads as 0.
5	ADCTOVIE	RW	0h	ADC conversion-time-overflow interrupt enable. 0b = Conversion-time overflow interrupt disabled 1b = Conversion-time overflow interrupt enabled
4	ADCOVIE	RW	0h	ADCMEM0 overflow interrupt enable. 0b = Overflow interrupt disabled 1b = Overflow interrupt enabled
3	ADCHIIE	RW	0h	Interrupt enable for the above upper threshold interrupt of the window comparator. 0b = Above upper threshold interrupt disabled 1b = Above upper threshold interrupt enabled
2	ADCLOIE	RW	0h	Interrupt enable for the below lower threshold interrupt of the window comparator. 0b = Below lower threshold interrupt disabled 1b = Below lower threshold interrupt enabled
1	ADCINIE	RW	0h	Interrupt enable for the inside of window interrupt of the window comparator. 0b = Inside of window interrupt disabled 1b = Inside of window interrupt enabled
0	ADCIE0	RW	0h	Interrupt enable. This bits enable or disable the interrupt request for a completed ADC conversion. 0b = Interrupt disabled 1b = Interrupt enabled

Interrupts

```
// ADC interrupt service routine
#pragma vector=ADC_VECTOR
__interrupt void ADC_ISR(void)
{
    switch(ADCIV)
    {
        case ADCIV_NONE:
            break;
        case ADCIV_ADCOVIFG:  // memory overflow - ADCMEM0 overflow
            break;
        case ADCIV_ADCTOVIFG:  // ADC conversion-time-overflow
            break;
        case ADCIV_ADCHIIFG:   // Comparator above upper threshold
            break;
        case ADCIV_ADCLOIFG:   // Comparator below lower threshold
            break;
        case ADCIV_ADCINIFG:   // Comparator inside window
            break;
        case ADCIV_ADCIFG:    // Conversion complete
            ADC_Result = ADCMEM0;
            __bic_SR_register_on_exit(LPM0_bits);            // Clear CPUOFF bit from LPM0
            break;
        default:
            break;
    }
}
```

More Good ADC stuff…….

20.2.7 ADC Conversion Modes

The ADC has four operating modes, selected by the CONSEQx bits (see Table 20-1).

Table 20-1. Conversion Mode Summary

ADCCONSEQx	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.