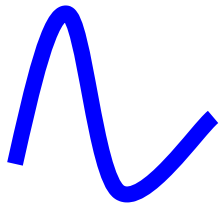
Computer interfacing (CI)

4. The physical interface. Analog I/O

4.1 Analog interface

The world provides our systems a lot of analog data: temperature, barometric pressure, accelerometers, audio (microphone)… and several devices can be driven by analog values: lights, motors, audio (speakers)…



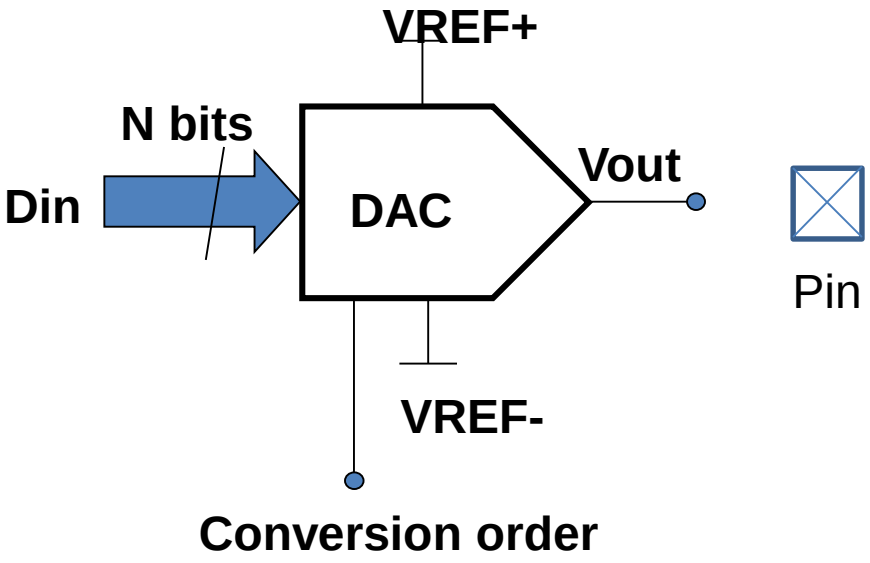
03, 07, 10, 14, 09,
02, 00, 04, …

PIC 18F45K22 has some pins with analog capabilities, using two internal peripherals:

- Digital to Analog Converter (DAC)
- Analog to Digital Converter (ADC)

4.2 Digital to Analog Converter: DAC.

This device aims to provide an analog output to a pin (not only a logical 0 (GND<V<V_{OL}) or logical 1 (V_{OH} >V>Vcc)) that can take a range of values between GND and Vcc (depending on the number of bits: 64, 128, 256 values).



Pin

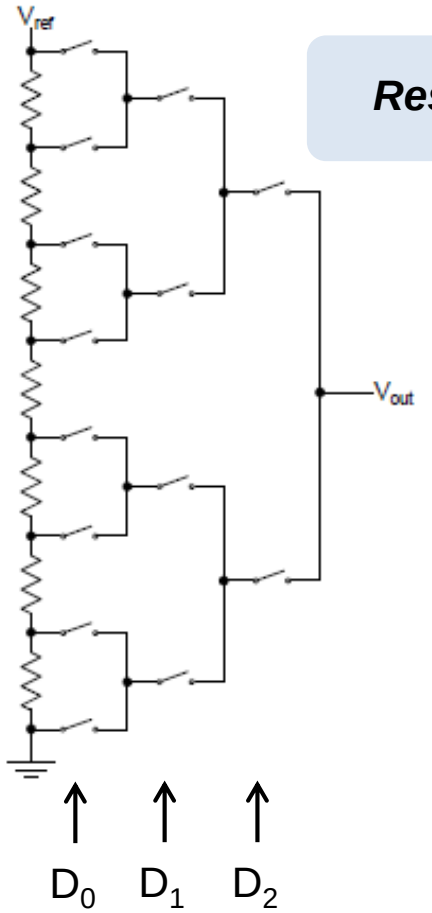
Conversion formula:

$$Vout = V_{REF-} + (V_{REF+} - V_{REF-}) \cdot \frac{D_{in}}{(2^N - 1)}$$

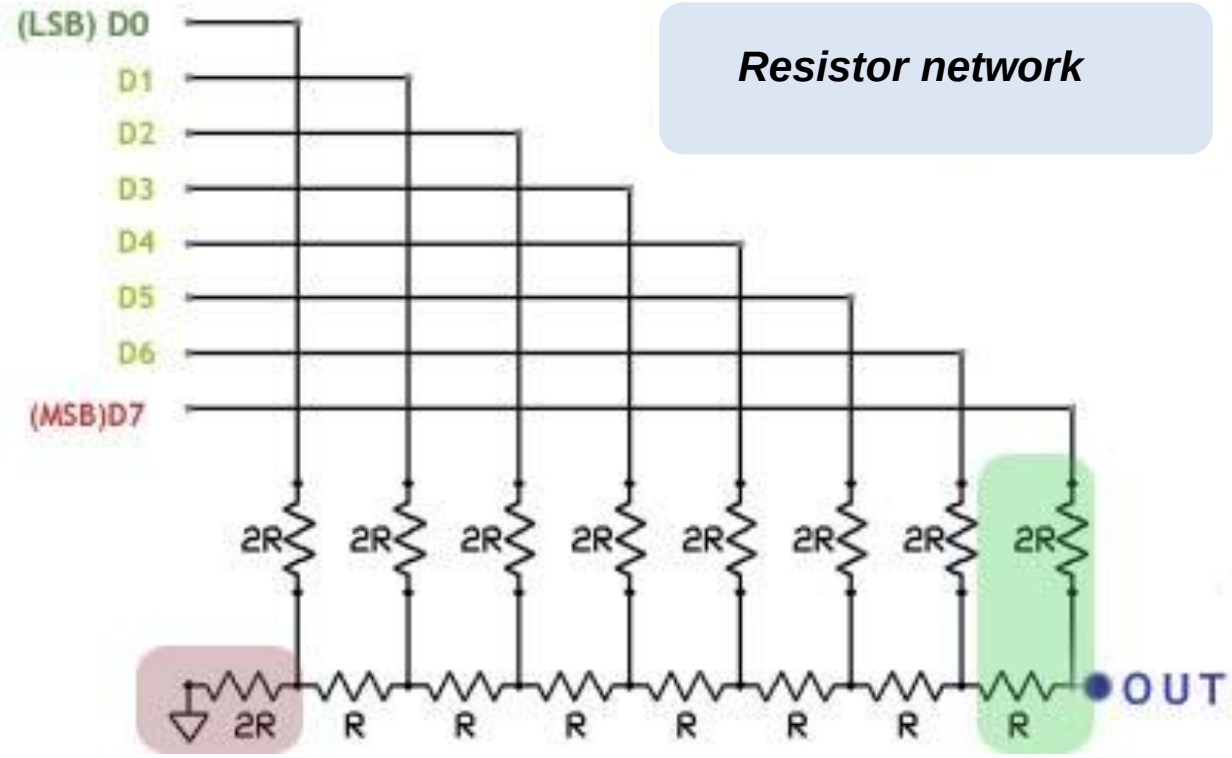
4.2 Digital to Analog Converter: DAC.

Example of implementations: resistor ladder, resistor network.

Resistor ladder



Resistor network

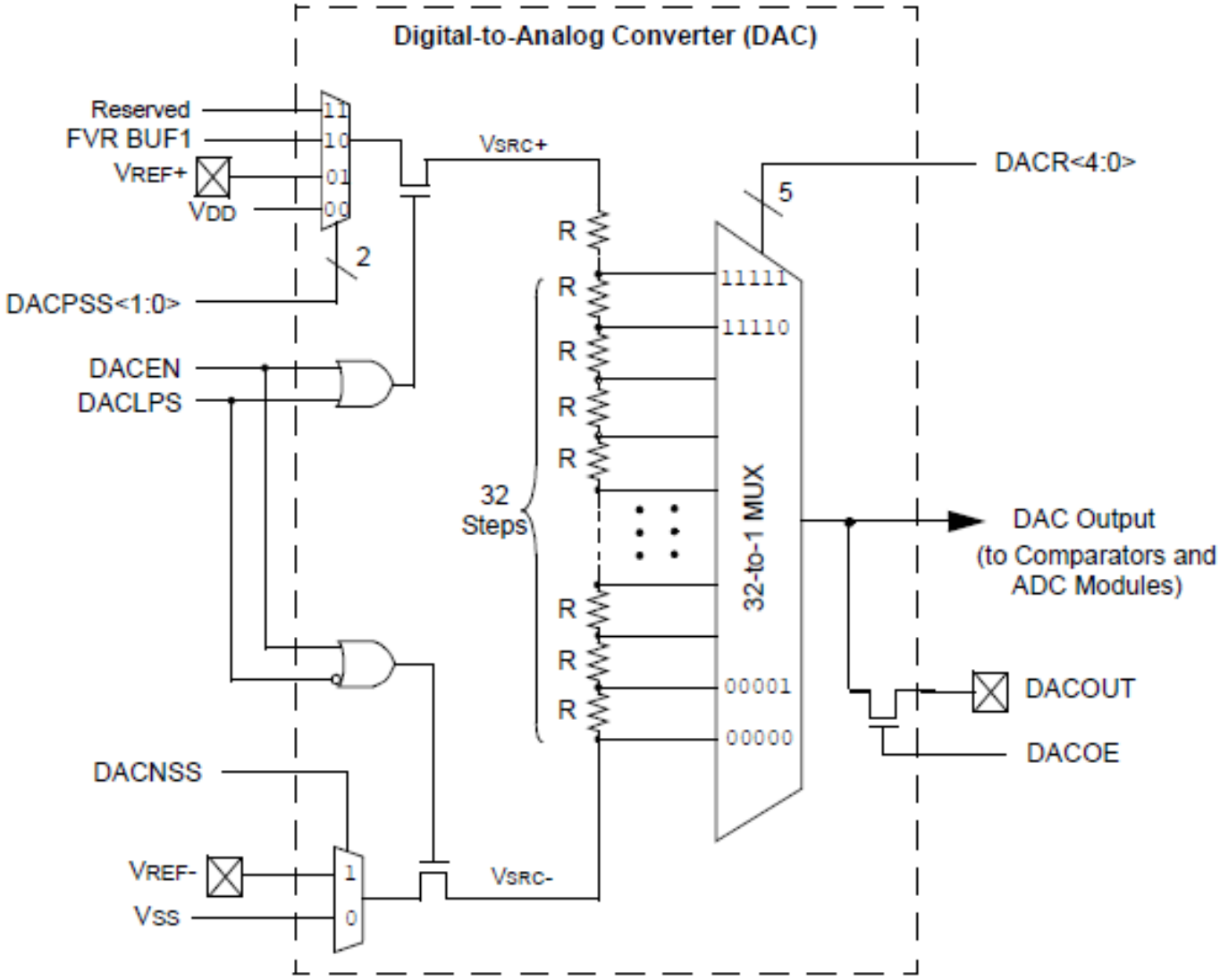


4.2 Digital to Analog Converter: DAC.

The PIC1845K22 has a **5-bit** D/A converter, resistor-ladder type. The output (DACOUT) is mapped to RA2, pin 4.

- DACEN starts/stops the device.
- DACOE enables the output.

There is a set of registers to control the DAC device; VREFCON1 and VREFCON2.



4.2 Digital to Analog Converter: DAC.

REGISTER 22-1: VREFCON1: VOLTAGE REFERENCE CONTROL REGISTER 0

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	U-0	R/W-0
DACEN	DACLPS	DACOE	—	DACPSS<1:0>		—	DACNSS
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **DACEN:** DAC Enable bit
1 = DAC is enabled
0 = DAC is disabled

bit 6 **DACLPS:** DAC Low-Power Voltage Source Select bit
1 = DAC Positive reference source selected
0 = DAC Negative reference source selected

bit 5 **DACOE:** DAC Voltage Output Enable bit
1 = DAC voltage level is also an output on the DACOUT pin
0 = DAC voltage level is disconnected from the DACOUT pin

bit 4 **Unimplemented:** Read as '0'

bit 3-2 **DACPSS<1:0>:** DAC Positive Source Select bits
00 = V_{DD}
01 = V_{REF+}
10 = FVR BUF1 output
11 = Reserved, do not use

bit 1 **Unimplemented:** Read as '0'

bit 0 **DACNSS:** DAC Negative Source Select bits
1 = V_{REF-}
0 = V_{SS}

4.2 Digital to Analog Converter: DAC.

REGISTER 22-2: VREFCON2: VOLTAGE REFERENCE CONTROL REGISTER 1

U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	DACR<4:0>				
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7-5 **Unimplemented:** Read as '0'

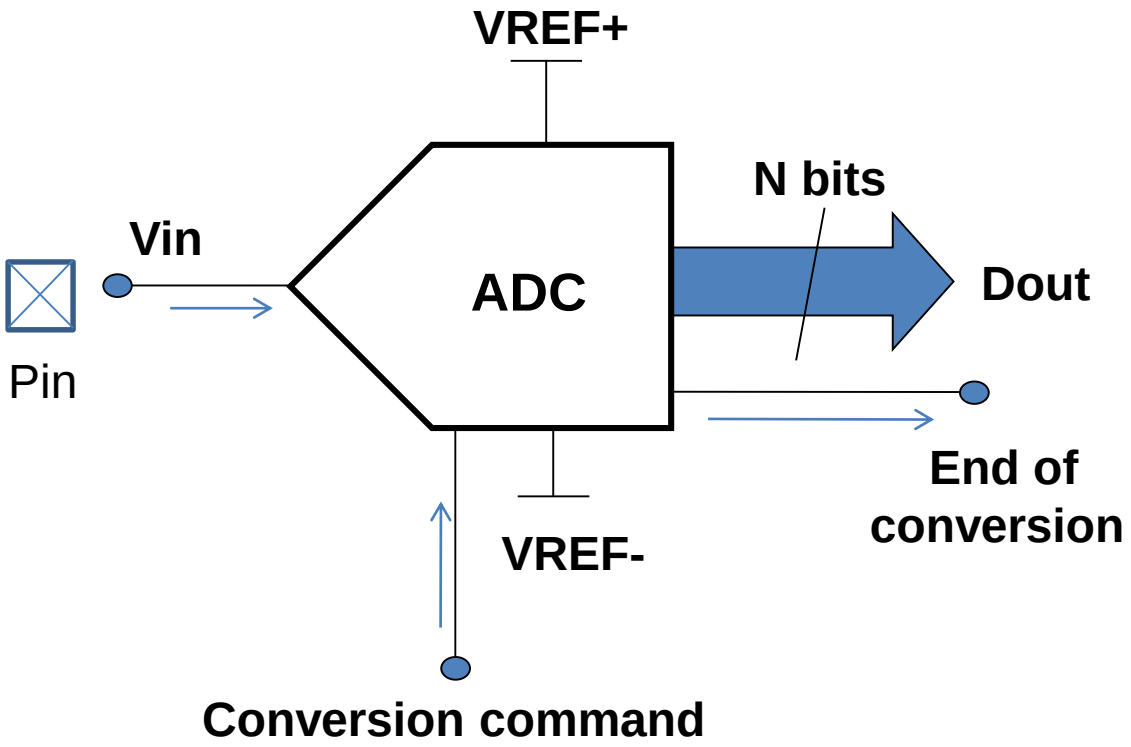
bit 4-0 **DACR<4:0>:** DAC Voltage Output Select bits

$V_{OUT} = ((V_{SRC+}) - (V_{SRC-}))*(DACR<4:0>/(2^5)) + V_{SRC-}$

Proposed exercise: configure the DAC device to output 1.406V in pin DACOUT.

4.3 Analog to Digital Converter: ADC

This device aims to map an analog input from a pin to a range of discrete digital values.



Conversion formula:

$$Dout = round\left[(2^N - 1) \cdot \frac{(V_{IN} - V_{REF-})}{(V_{REF+} - V_{REF-})}\right]$$

4.3 Analog to Digital Converter: ADC

Typically, the data acquisition process of a physical magnitude (i.e. temperature), requires a sensing stage (transducer), signal conditioning (filtering, amplification), and finally the ADC.

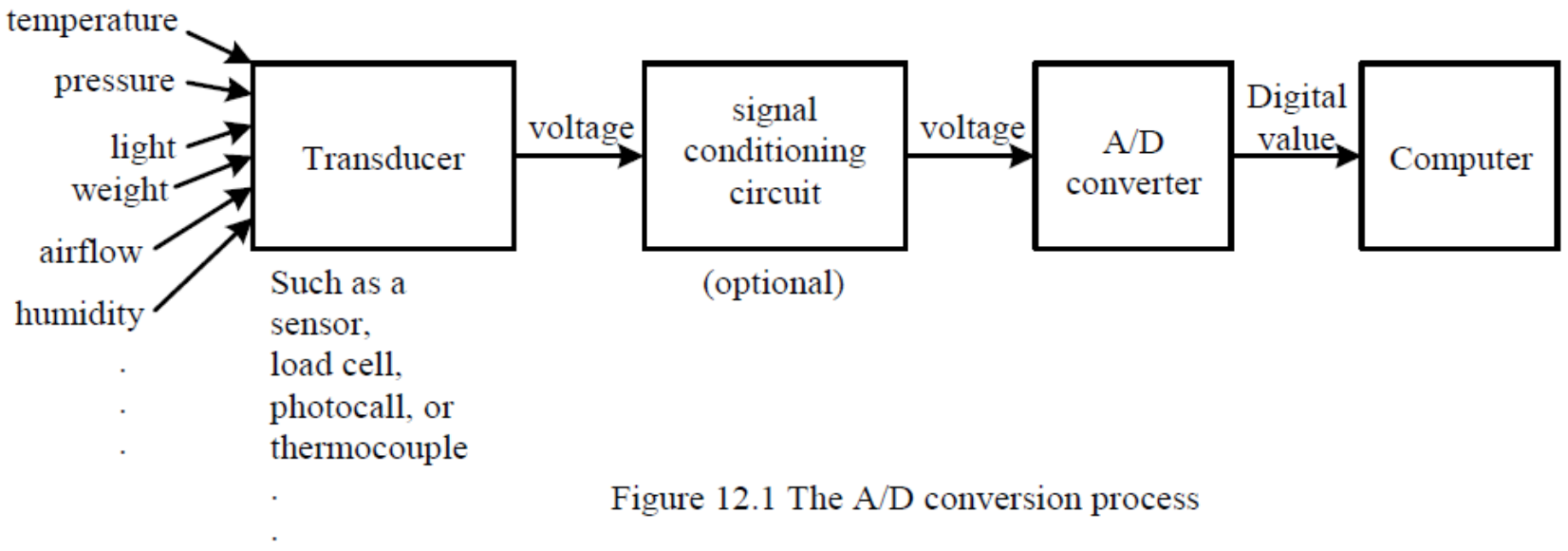


Figure 12.1 The A/D conversion process

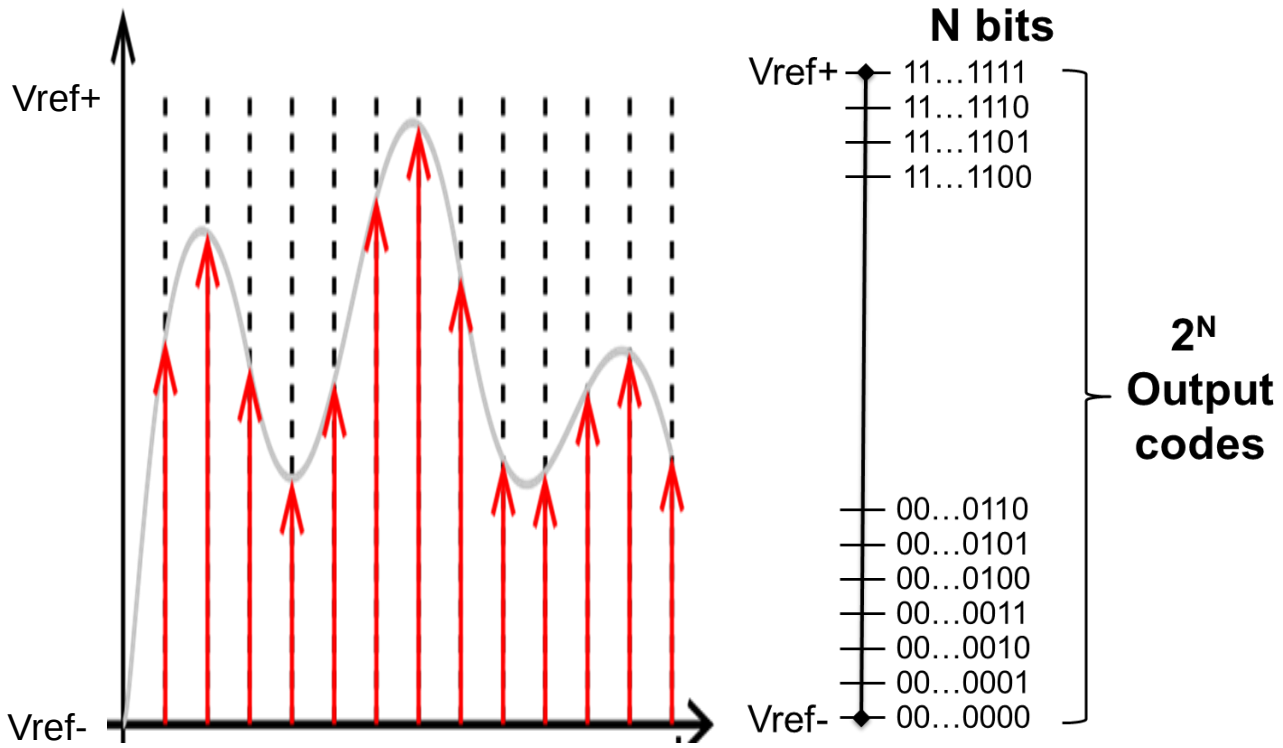
4.3 Analog to Digital Converter: ADC

The ADC discretization process has its limitations in resolution an precision:

Resolution of ADC = $(V_{ref+}-V_{ref-})/(2^N-1)$

Maximum conversion error = $ADC_{resolution}$ / 2

Average conversion error = 0

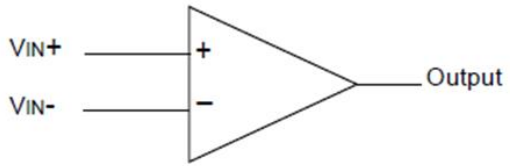


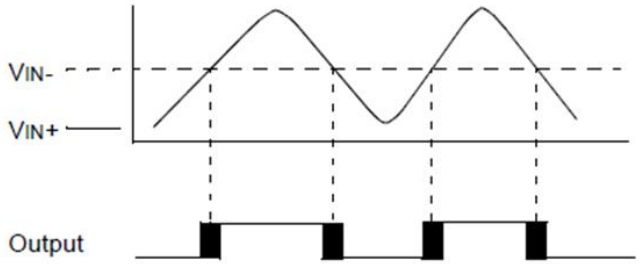
4.3 Analog to Digital Converter: ADC

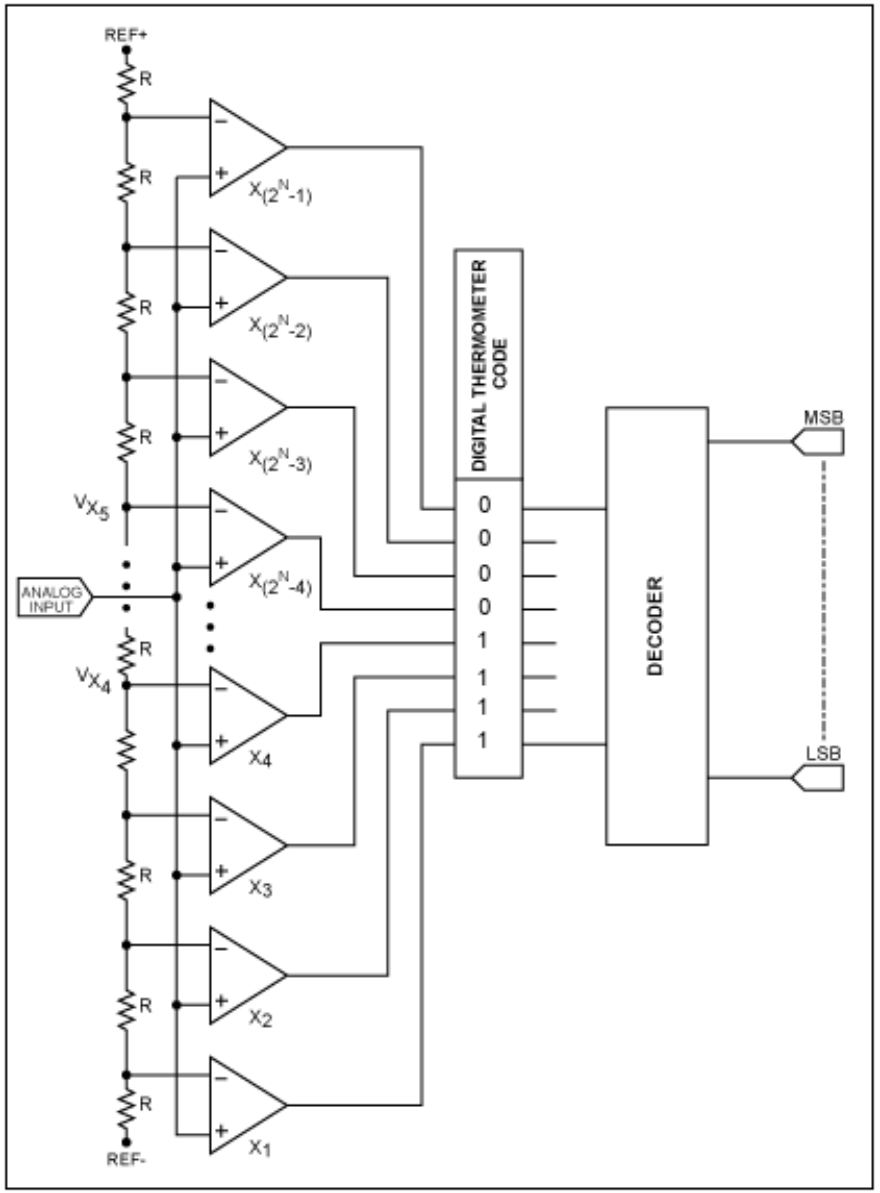
Implementations of ADC: Flash Converter

- Very fast (conversion time near 0)
- Expensive
- Limited to 8-10 bits

Based on comparators:



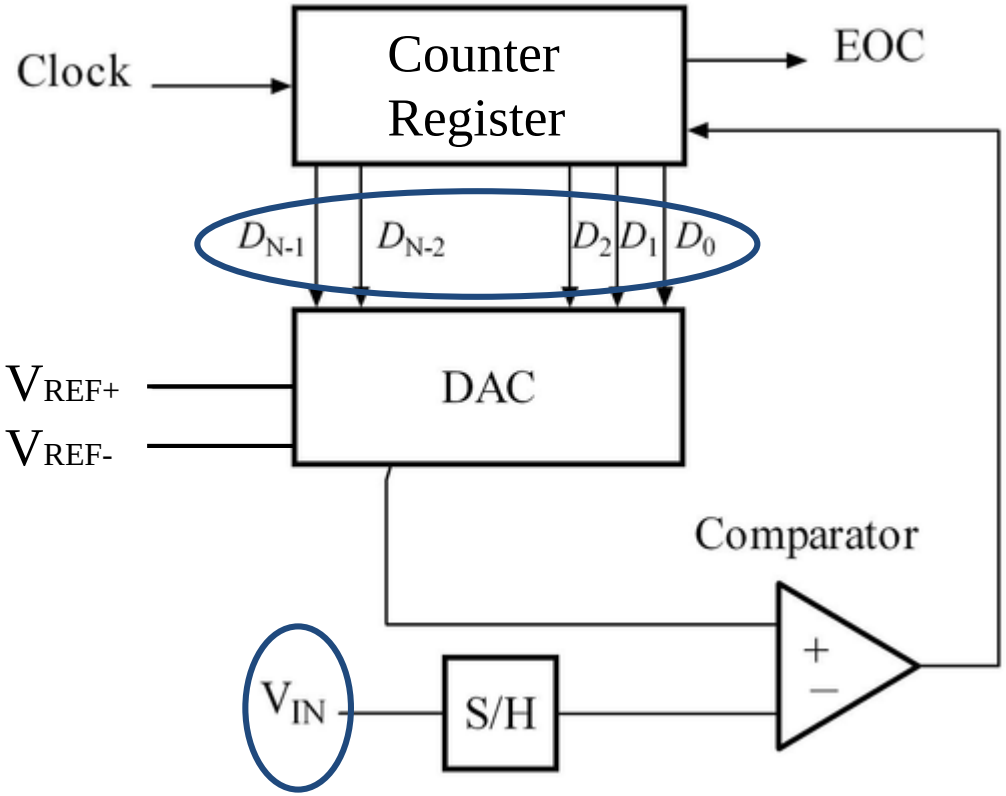




4.3 Analog to Digital Converter: ADC

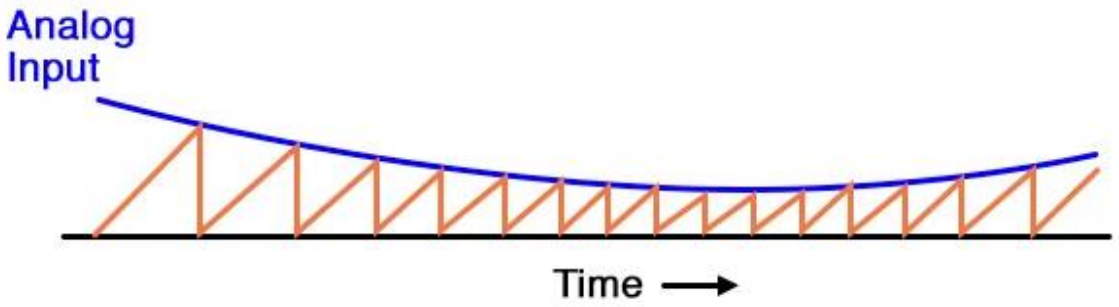
Implementations of ADC: Slope Converter

- Slow (conversion time may vary!)
- Cheap
- Can manage up to 16 bits



Based on a counter and one comparator.

Sampling example over time (not constant):

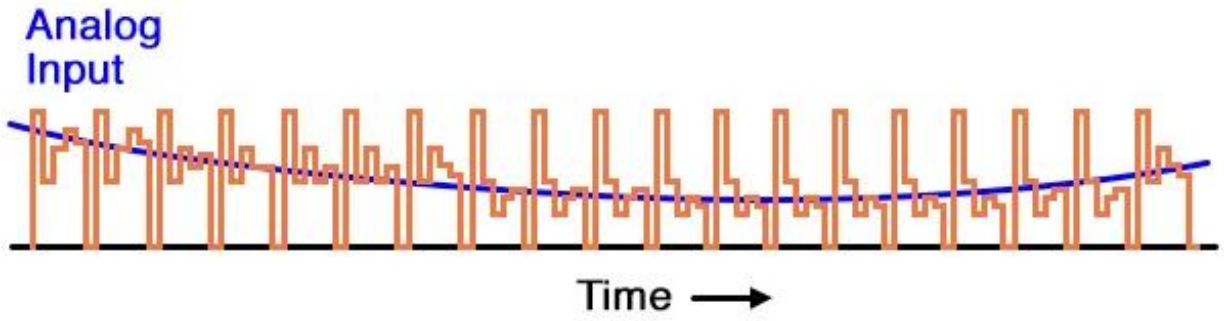


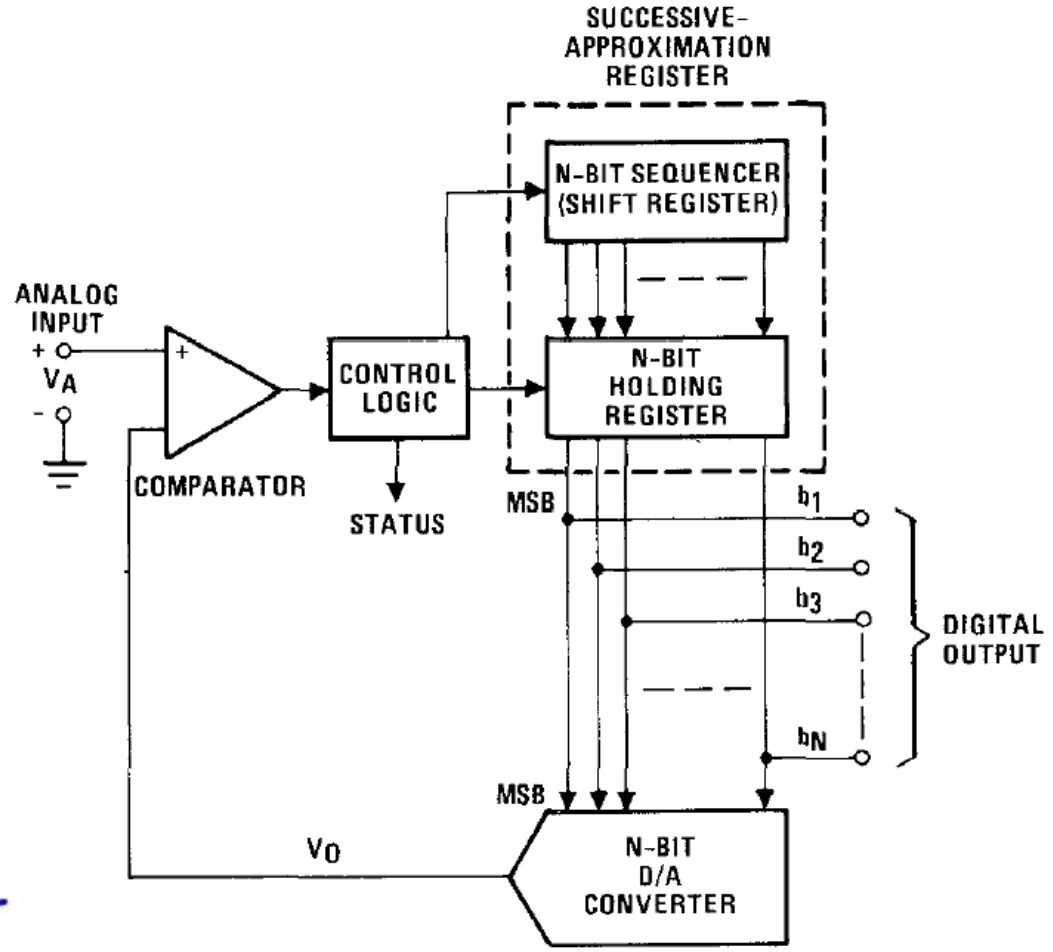
4.3 Analog to Digital Converter: ADC

Implementations of ADC: Successive Approximation Register (SAR)

- Faster (log) than slope
- Cheap
- Can manage up to 16 bits
- Predictable conversion time

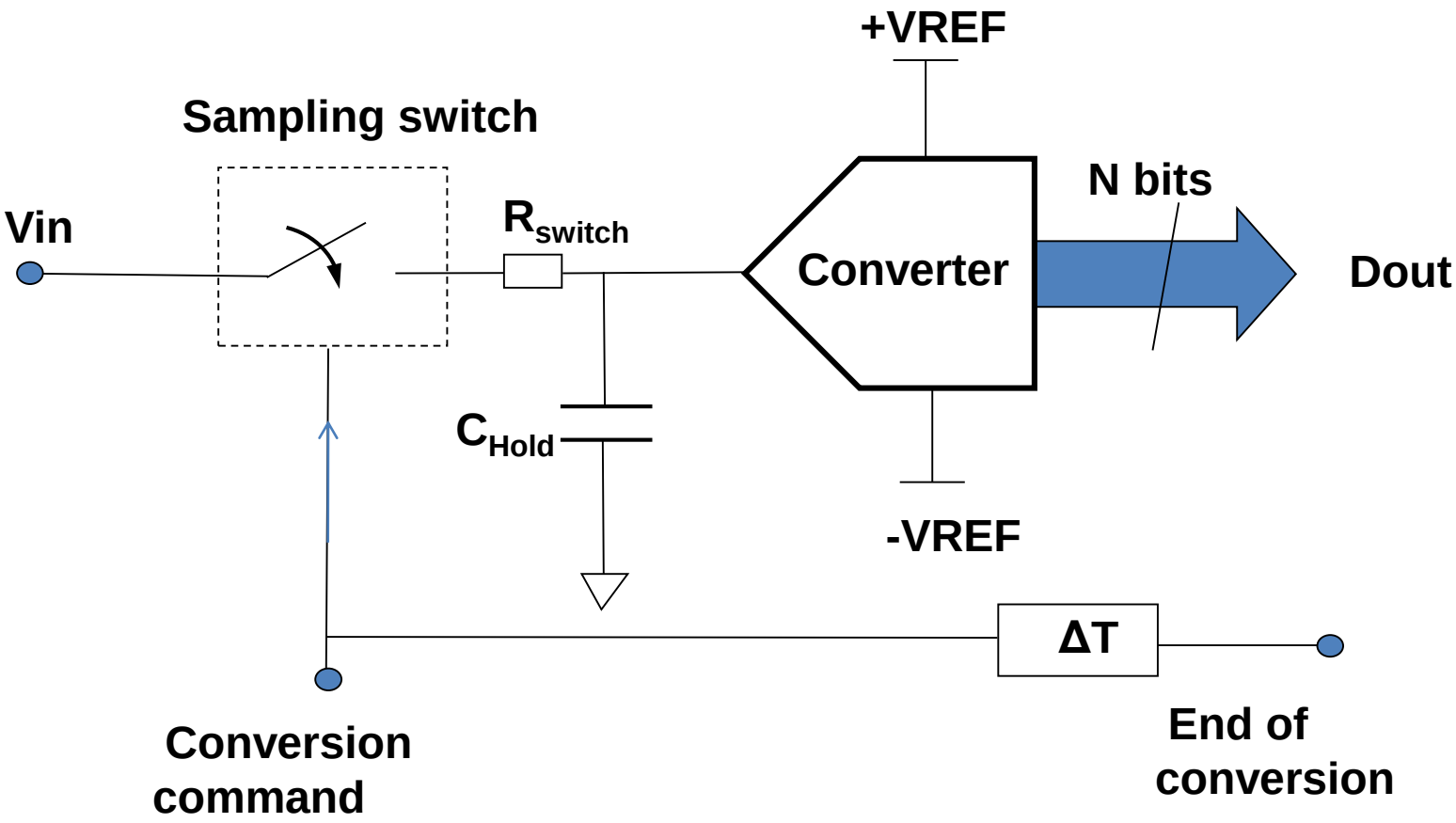
Tests every bit (0/1) performing a dichotomic search.





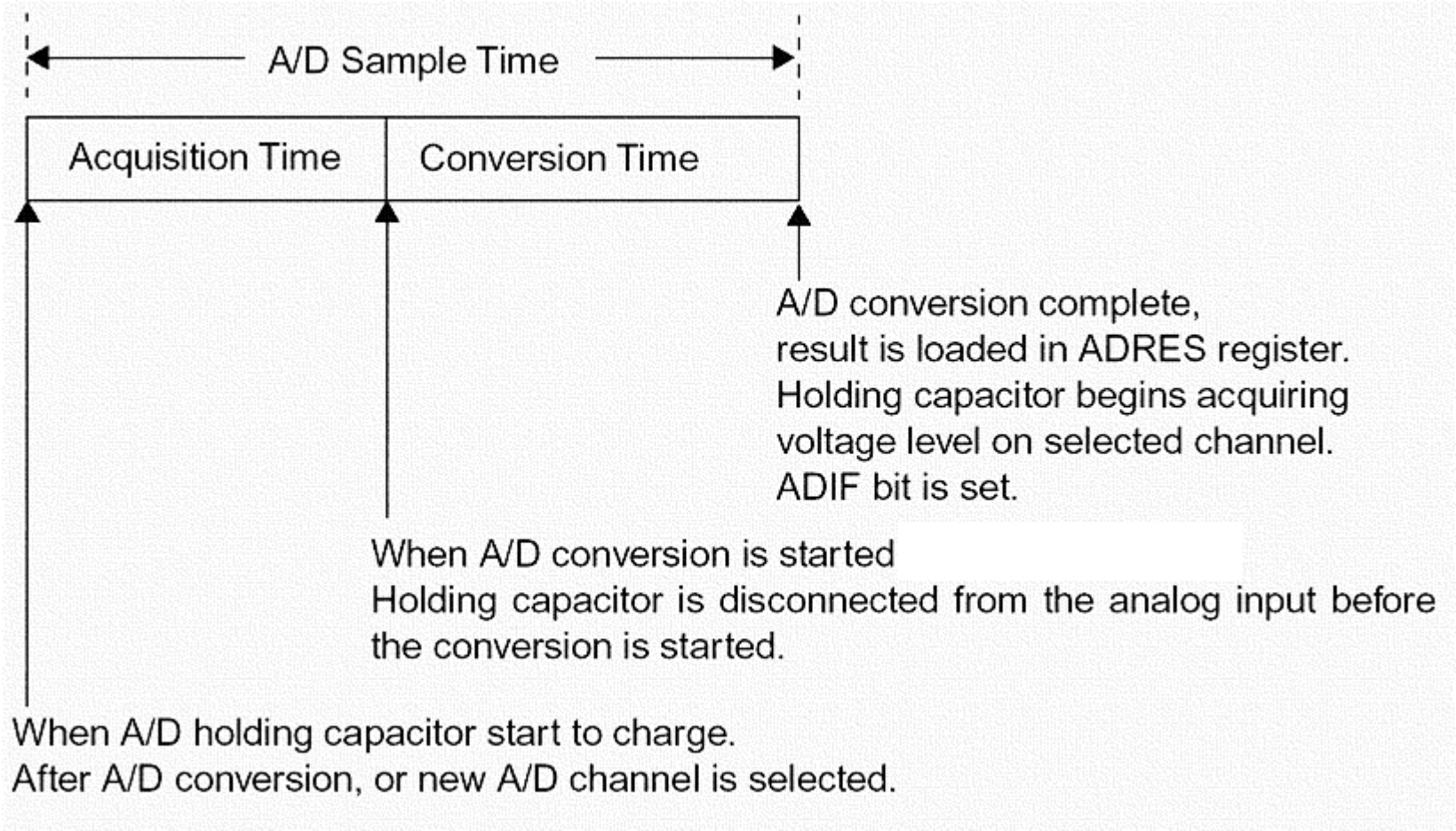
4.3 Analog to Digital Converter: ADC

ADC has a previous stage “Sample&Hold” to take a sample of the analog data before starting the conversion. The Switch – Resistor – Capacitor circuit takes a time to operate, it’s called, acquisition time.



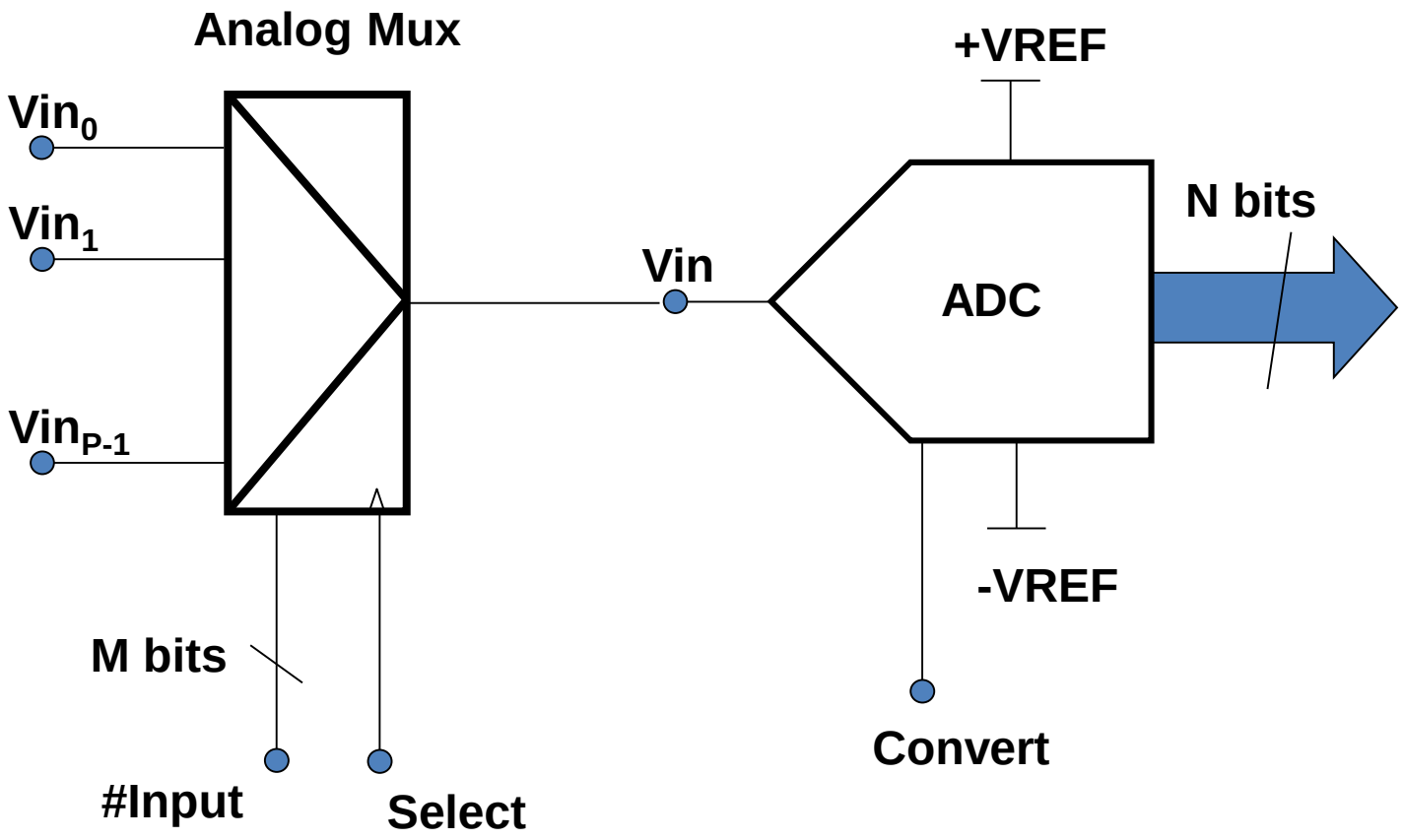
4.3 Analog to Digital Converter: ADC

So, the total time taken is the addition of the acquisition time and the conversion time.



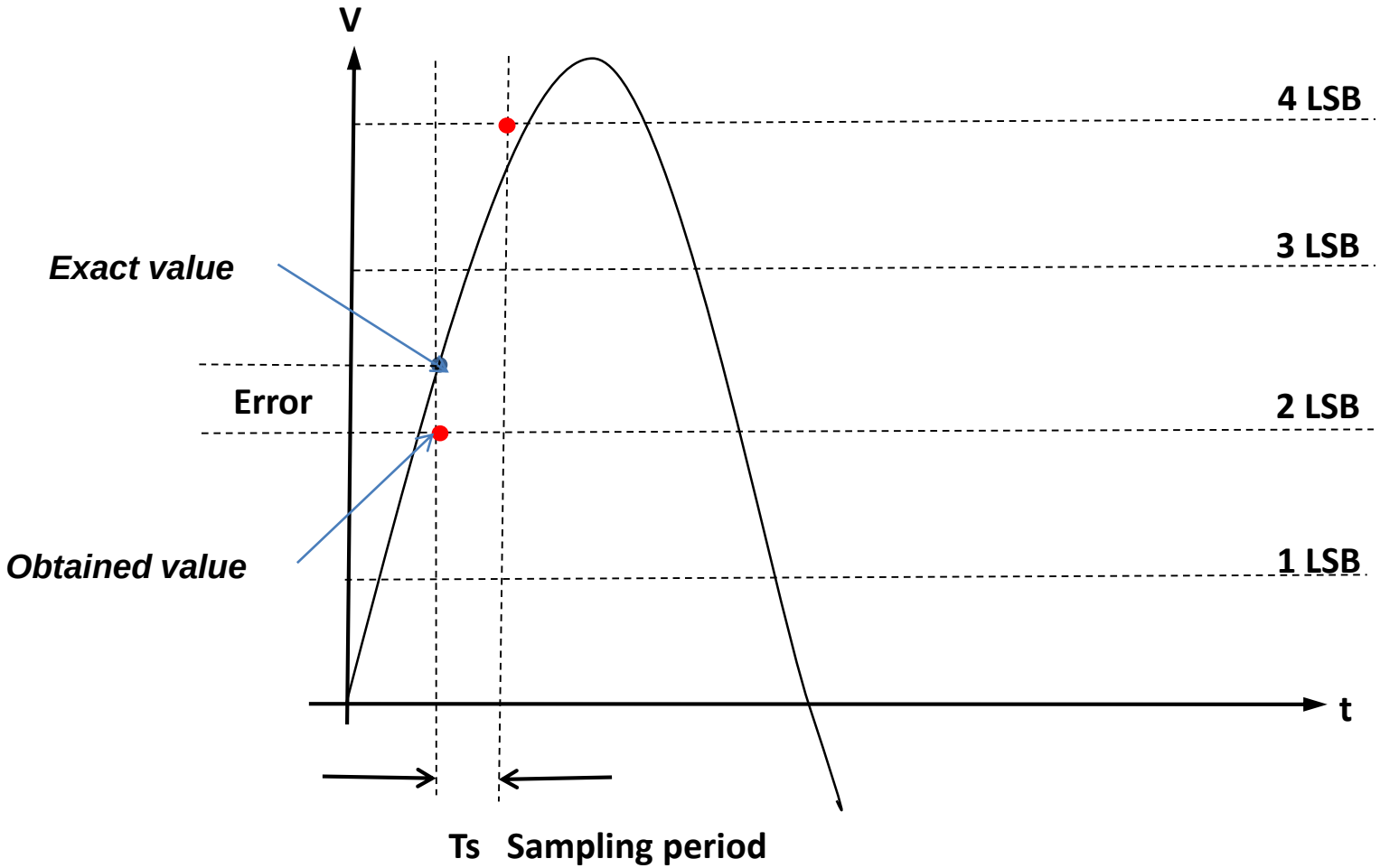
4.3 Analog to Digital Converter: ADC

If we have to acquire data from multiple channels, and provided that we have only one ADC in our microcontroller, we had to consider an extra time for multiplexing the inputs.



4.3 Analog to Digital Converter: ADC

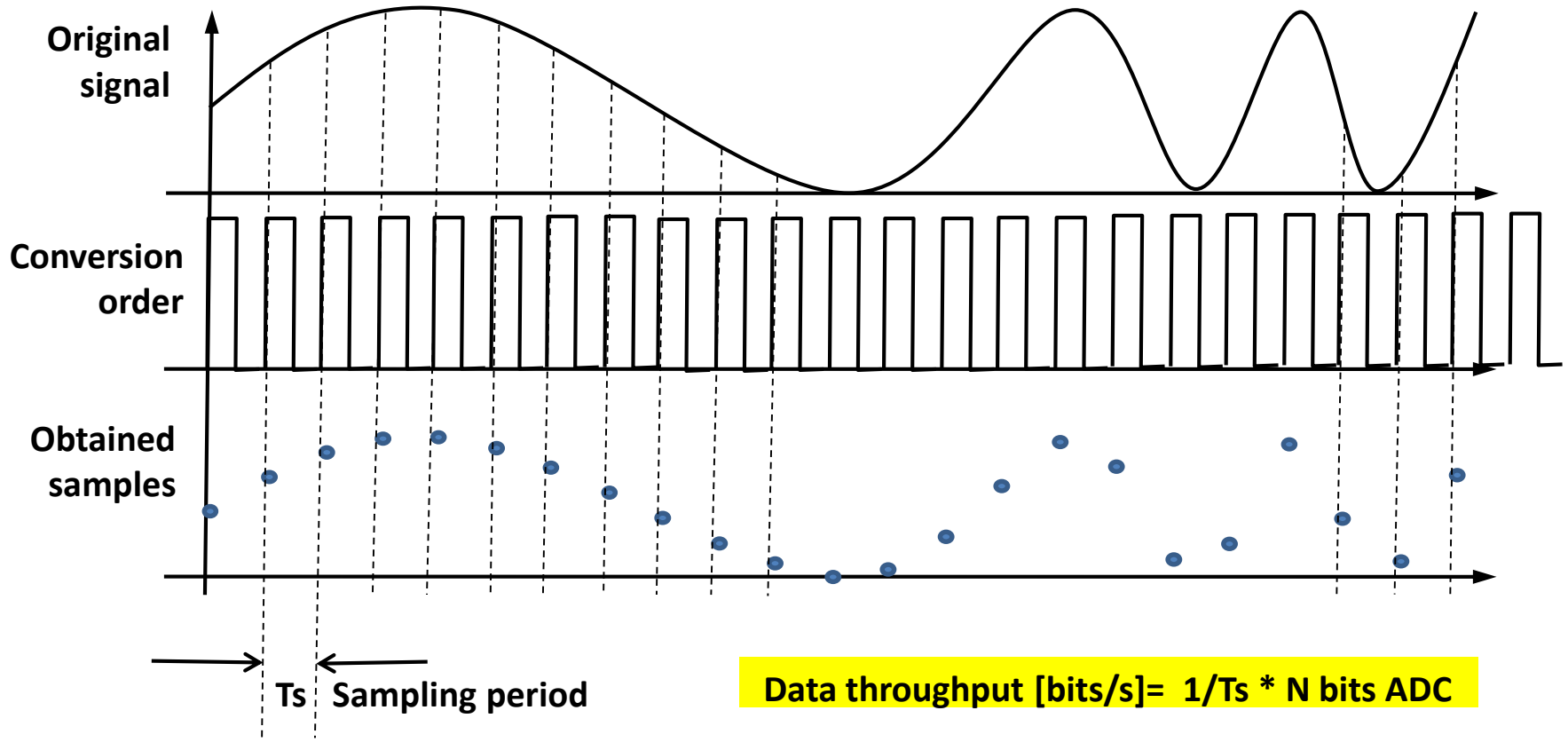
There will be precision errors due to the quantization method (1/2 LSB) and sampling time.



4.4 Data acquisition with ADC

As seen, analogic to digital conversion is not instantaneous.

We have to decide how many **samples per second** we need for a given input.



4.4 Data acquisition with ADC

There are some trade-off in the process:

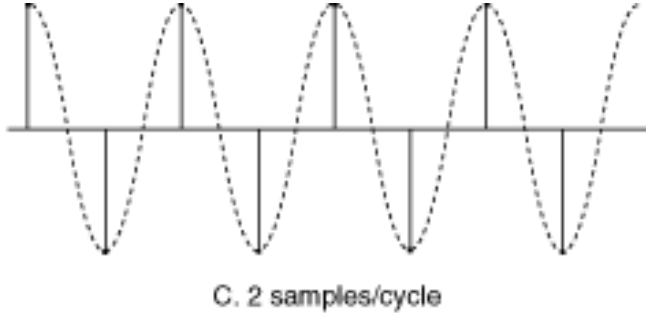
- Sampling period cannot be less than sampling time.
- The more samples we take per second, more data throughput will have (it is necessary?)
- If the original signal has high frequencies (in relation with our sampling), we well be very inaccurate.

So, we can use the Nyquist criterion (mathematics):

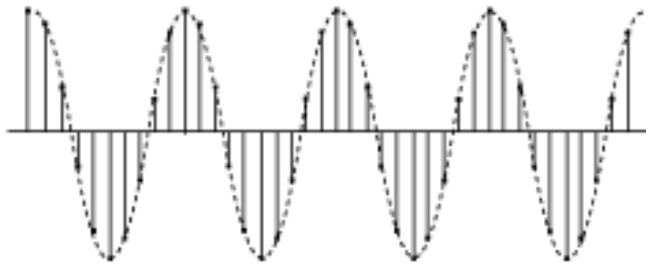
$$f_{sampling} > 2 \cdot f_{signal\ max\ frequency}$$

And, if possible, the engineering standard:

$$f_{sampling} = 10 \cdot f_{signal\ max\ frequency}$$



C. 2 samples/cycle



D. 10 samples/cycle

4.4 Data acquisition with ADC

Some examples:

- Hi-Fi sound (20-20,000 Hz) is generally sampled at about 44 kHz.
- External temperature during flight need only be sampled every few seconds at most.

Audio facts:

Range of audible frequencies: 20 Hz to 20KHz (individual depending)
Frequency range of an analog phone call: 350 Hz to 3500 Hz
Violin frequency range: 96 Hz to 10 kHz (approx.)

8 Hz Lowest organ note (*note = fundamental freq*)
32 Hz Lowest note on a standard 88-key piano
80 Hz Lowest note reproducible by the average female human voice
500 Hz Fundamental frequency of a crying baby
1050 Hz Highest note reproducible by the average female human voice
4186 Hz The highest note on a standard 88-key piano
16K Hz The highest harmonic of a female human voice

4.4 Data acquisition with ADC

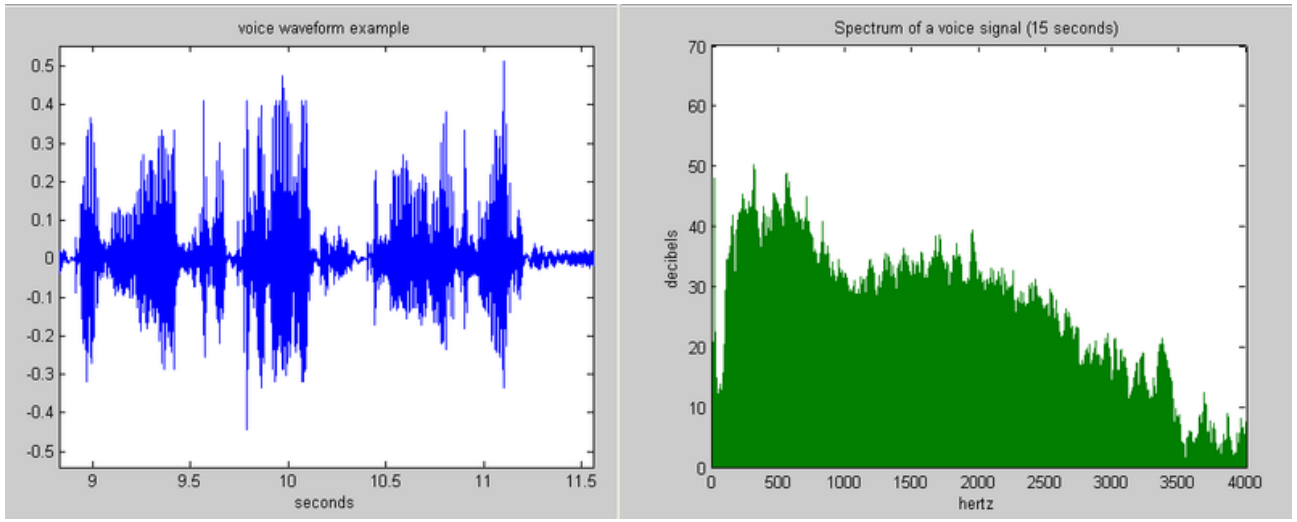
But… how do we know the frequencies of a signal?

Fourier's Theorem states that any periodic signal is composed of a superposition of pure sine waves, with suitably chosen amplitudes and phases, whose frequencies are harmonics of the fundamental frequency of the signal.

Fourier's Transform is a mathematical tool to obtain the frequencies of a given signal:

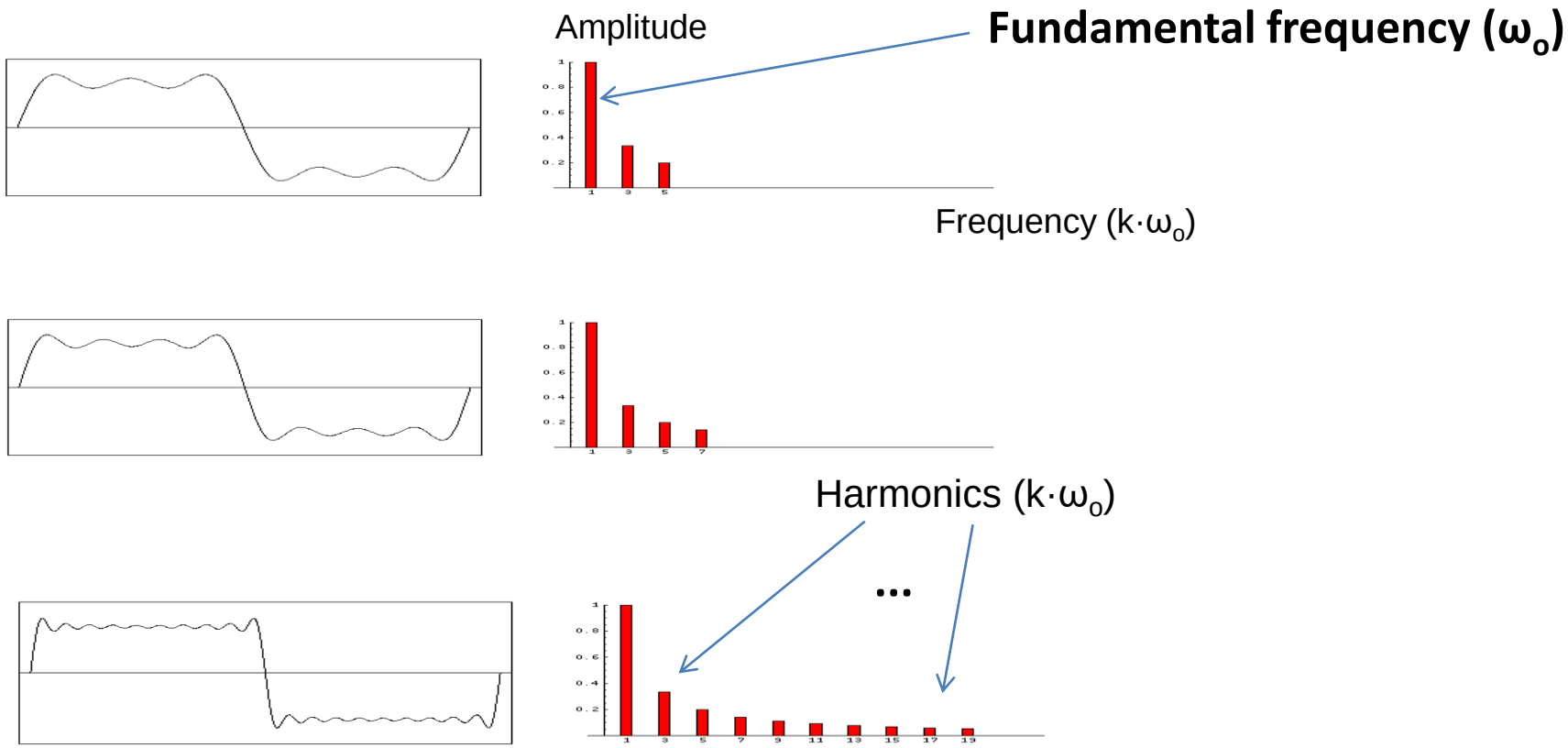
$$\widehat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\, e^{-i2\pi\xi x}\, dx.$$

Or we can use our oscilloscopes or a spectrum analyzer.



4.4 Data acquisition with ADC

Example: reconstruction of a square signal by adding more harmonics (frequencies).

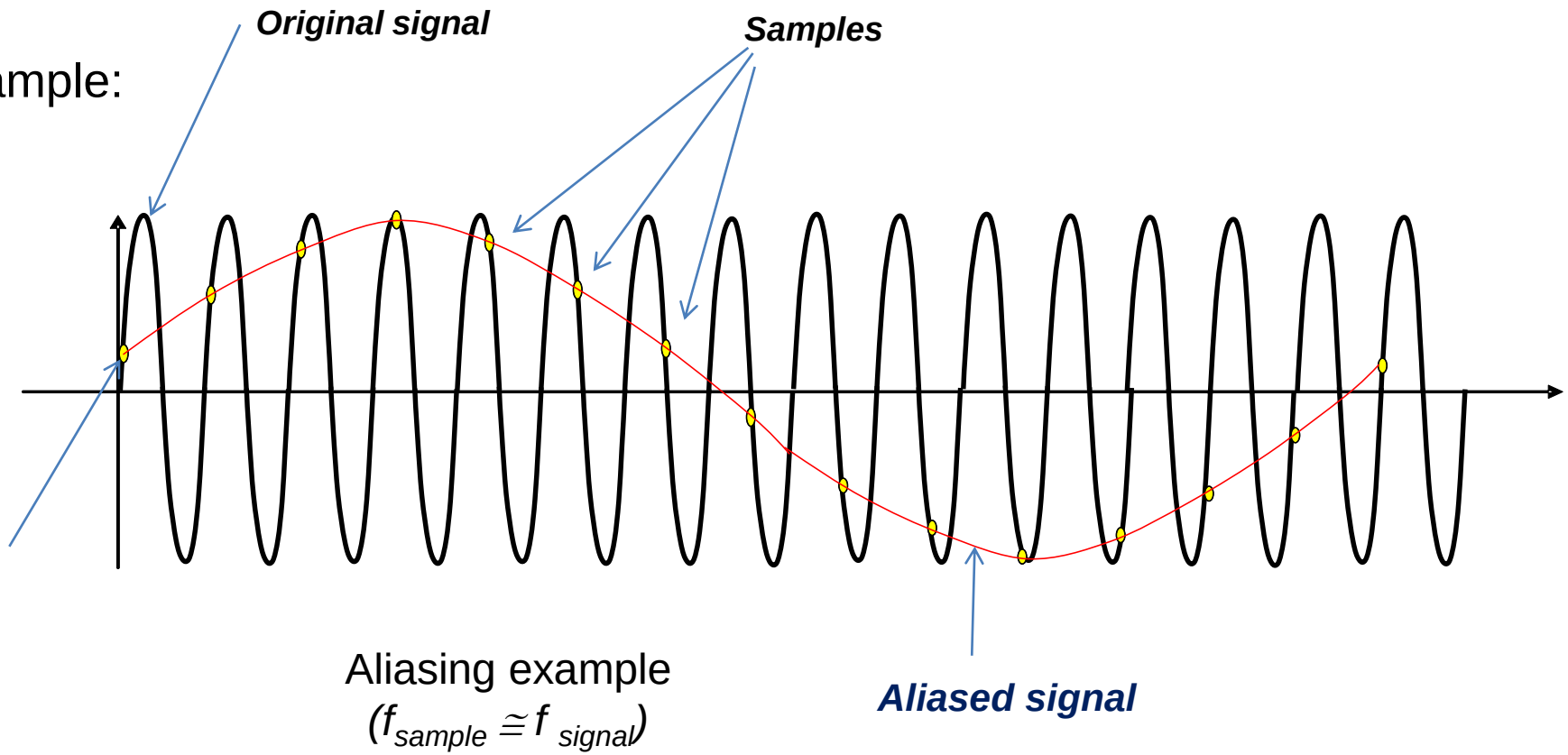


4.4 Data acquisition with ADC

The aliasing phenomena.

Aliasing appears when our input signal contains a frequency very close to the sampling frequency.

1D Aliasing example:



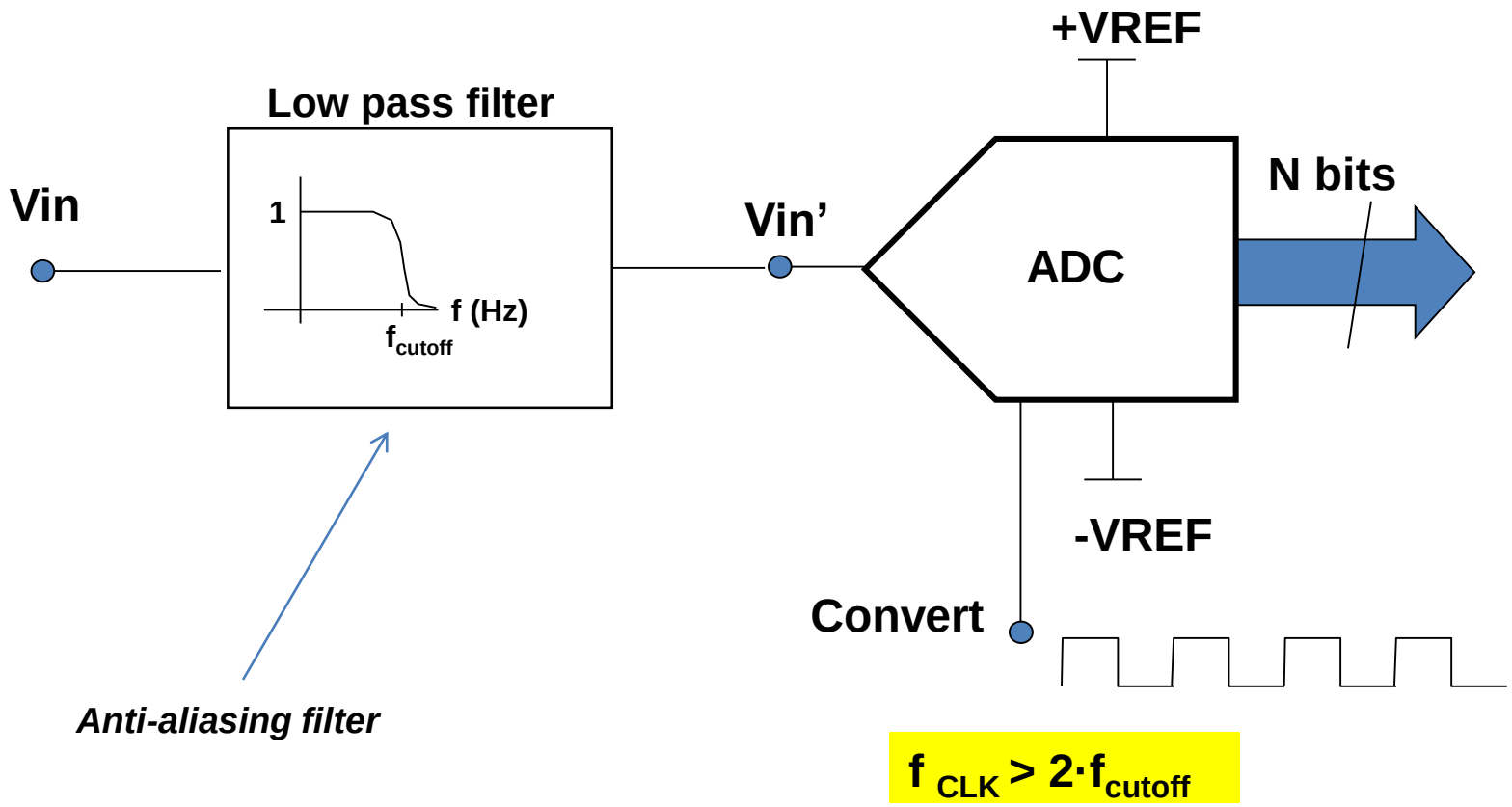
4.4 Data acquisition with ADC

2D Aliasing example:

Frequency of brick's pattern is similar to the pixel resolution.

4.4 Data acquisition with ADC

The only solution is to avoid those sampling-frequency near signals, enter into our system.



4.5 The PIC 18F ADC converter.

The PIC18 has a 10-bit A/D Successive Approximations converter.

- The number of analog inputs varies among different PIC18 devices (multiplexed).

  ANSELx sets the desired pins in analog input mode.

- The A/D converter has the following registers:

  - A/D Result High Register (ADRESH)
  - A/D Result Low Register (ADRESL)
  - A/D Control Register 0 (ADCON0) (source selection)
  - A/D Control Register 1 (ADCON1) (reference selection)
  - A/D Control Register 2 (ADCON2) (timing selections)

- The contents of these registers vary with the PIC18 members.
- Other parameters must be considered: ADIF, ADIE, ADIP (for AD interrupt)…

4.5 The PIC 18F ADC converter.

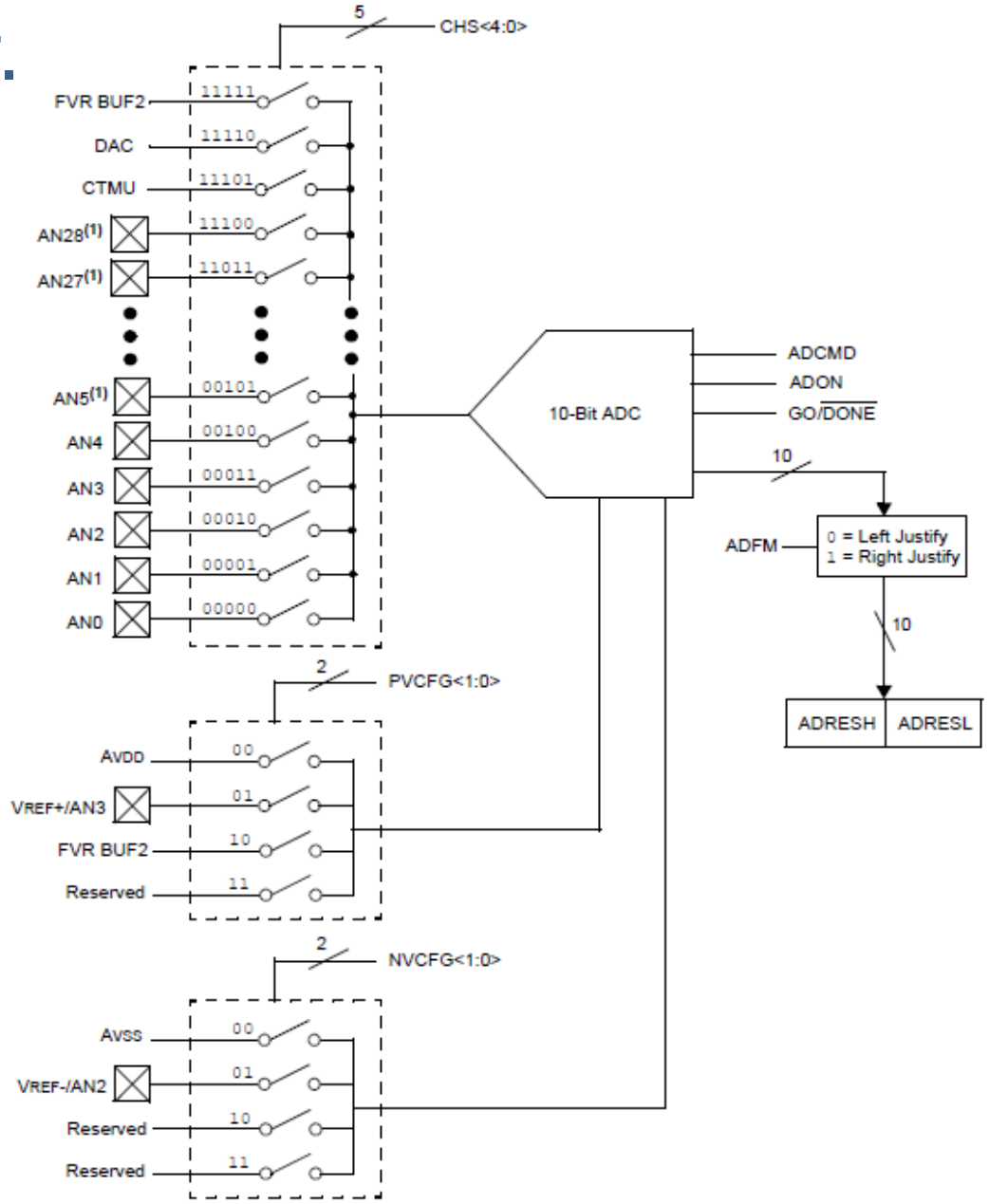
ADC Schematic.

- Anx are the (multiplexed) analog inputs.
- CHS drives the input multiplexor.
- ADON starts/stops the device.

- GO signal starts a conversion.

- ¬ DONE signal, notifies completion of the conversion.

- The result is available in ADRES registers.



4.5 The PIC 18F ADC converter.

REGISTER 17-1: ADCON0: A/D CONTROL REGISTER 0

U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	CHS<4:0>					GO/$\overline{\text{DONE}}$	ADON
bit 7							bit 0

bit 7 **Unimplemented:** Read as '0'

bit 6-2 **CHS<4:0>: Analog Channel Select bits**

00000 = AN0
00001 = AN1
00010 = AN2
00011 = AN3
00100 = AN4
00101 = AN5(1)
00110 = AN6(1)
00111 = AN7(1)
01000 = AN8
01001 = AN9
01010 = AN10
01011 = AN11
01100 = AN12
01101 = AN13
01110 = AN14
01111 = AN15
10000 = AN16
10001 = AN17
10010 = AN18
10011 = AN19
10100 = AN20(1)
10101 = AN21(1)
10110 = AN22(1)
10111 = AN23(1)
11000 = AN24(1)
11001 = AN25(1)
11010 = AN26(1)
11011 = AN27(1)
11100 = Reserved
11101 = CTMU
11110 = DAC
11111 = FVR BUF2 (1.024V/2.048V/2.096V Volt Fixed Voltage Reference)(2)

bit 1 **GO/$\overline{\text{DONE}}$:** A/D Conversion Status bit
1 = A/D conversion cycle in progress. Setting this bit starts an A/D conversion cycle.
This bit is automatically cleared by hardware when the A/D conversion has completed.
0 = A/D conversion completed/not in progress

bit 0 **ADON:** ADC Enable bit
1 = ADC is enabled
0 = ADC is disabled and consumes no operating current

Note 1: Available on PIC18(L)F4XK22 devices only.
2: Allow greater than 15 µs acquisition time when measuring the Fixed Voltage Reference.

4.5 The PIC 18F ADC converter.

REGISTER 17-2: ADCON1: A/D CONTROL REGISTER 1

R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
TRIGSEL	—	—	—	PVCFG<1:0>		NVCFG<1:0>	
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **TRIGSEL**: Special Trigger Select bit
1 = Selects the special trigger from CTMU
0 = Selects the special trigger from CCP5

bit 6-4 **Unimplemented:** Read as '0'

bit 3-2 **PVCFG<1:0>:** Positive Voltage Reference Configuration bits
00 = A/D VREF+ connected to internal signal, AVDD
01 = A/D VREF+ connected to external pin, VREF+
10 = A/D VREF+ connected to internal signal, FVR BUF2
11 = Reserved (by default, A/D VREF+ connected to internal signal, AVDD)

bit 1-0 **NVCFG<1:0>:** Negative Voltage Reference Configuration bits
00 = A/D VREF- connected to internal signal, AVSS
01 = A/D VREF- connected to external pin, VREF-
10 = Reserved (by default, A/D VREF- connected to internal signal, AVSS)
11 = Reserved (by default, A/D VREF- connected to internal signal, AVSS)

4.5 The PIC 18F ADC converter.

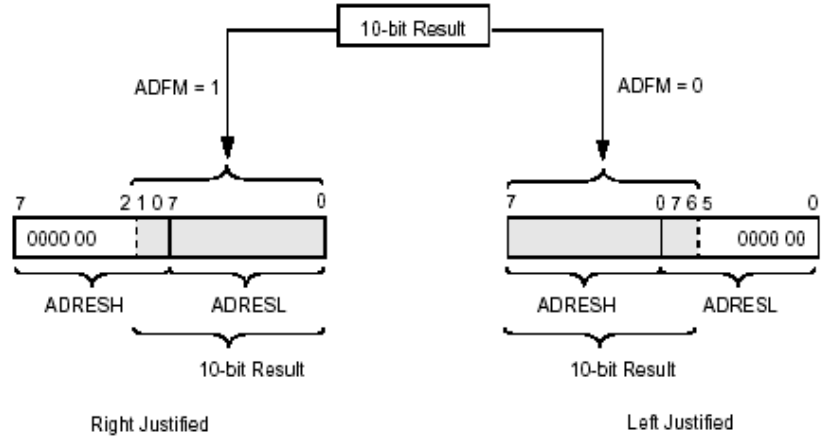
REGISTER 17-3: ADCON2: A/D CONTROL REGISTER 2

R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	ACQT<2:0>			ADCS<2:0>		
bit 7							bit 0

Legend:			
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'	
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared	x = Bit is unknown

bit 7 **ADFM:** A/D Conversion Result Format Select bit
1 = Right justified
0 = Left justified

10 bit data result format



bit 6 **Unimplemented:** Read as '0'

bit 5-3 **ACQT<2:0>:** A/D Acquisition time select bits. Acquisition time is the duration that the A/D charge holding capacitor remains connected to A/D channel from the instant the GO/$\overline{\text{DONE}}$ bit is set until conversions begins.
000 = 0(1)
001 = 2 TAD
010 = 4 TAD
011 = 6 TAD
100 = 8 TAD
101 = 12 TAD
110 = 16 TAD
111 = 20 TAD

Acquisition time based on TAD

bit 2-0 **ADCS<2:0>:** A/D Conversion Clock Select bits
000 = Fosc/2
001 = Fosc/8
010 = Fosc/32
011 = FRC(1) (clock derived from a dedicated internal oscillator = 600 kHz nominal)
100 = Fosc/4
101 = Fosc/16
110 = Fosc/64
111 = FRC(1) (clock derived from a dedicated internal oscillator = 600 kHz nominal)

TAD generated from Fosc

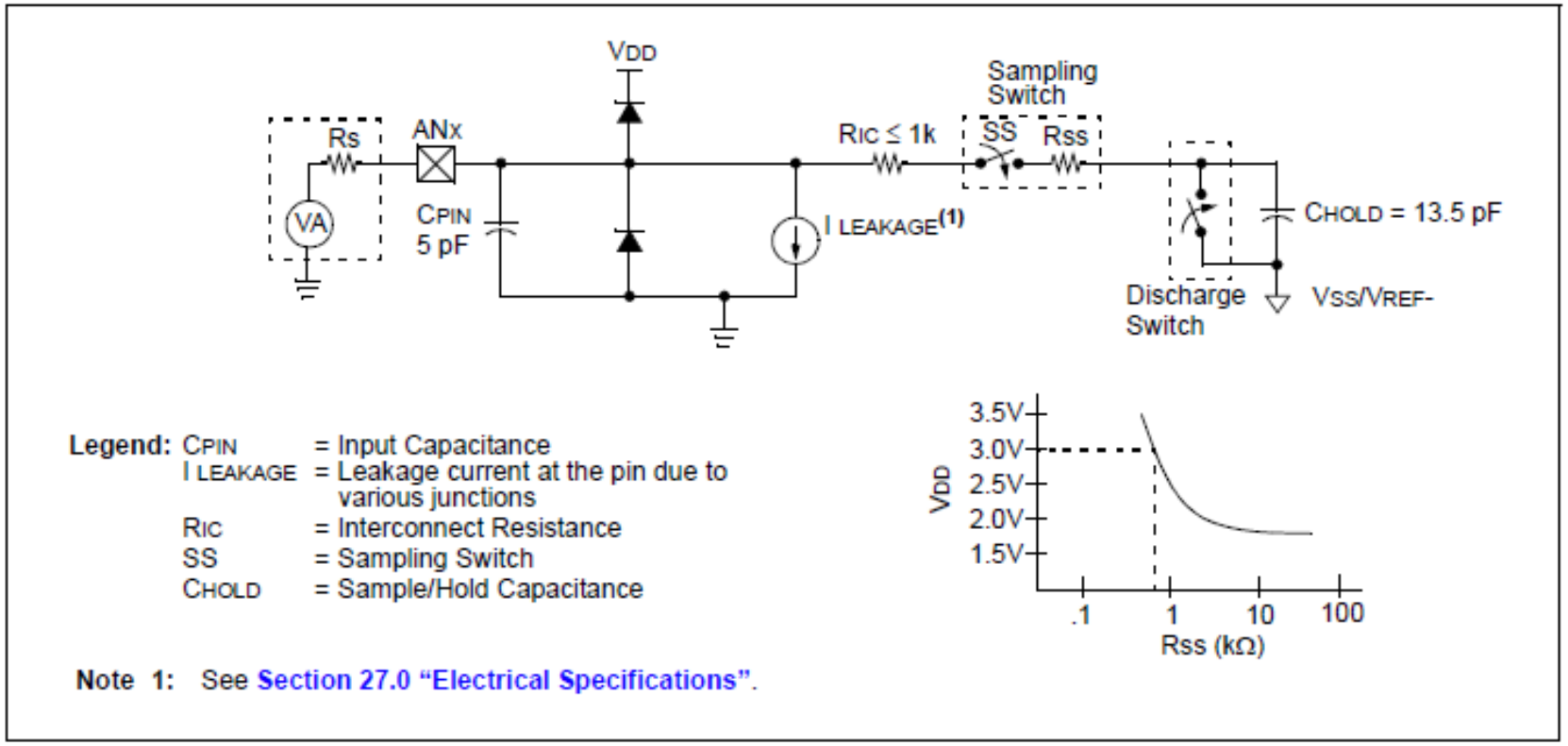
Note 1: When the A/D clock source is selected as FRC then the start of conversion is delayed by one instruction cycle after the GO/$\overline{\text{DONE}}$ bit is set to allow the SLEEP instruction to be executed.

4.5 The PIC 18F ADC converter.

ADC acquisition time requirements.

- The Sample&Hold circuit will keep the voltage stable while it is converted.
- Hold Capacitor needs a time to load the voltage present on the input.

FIGURE 17-5: ANALOG INPUT MODEL



4.5 The PIC 18F ADC converter.

Acquisition time calculation example, provided in the datasheet:

Assumptions: *Temperature = 50°C and external impedance of 10kΩ 3.0V VDD*

$$\begin{aligned} TACQ &= \text{Amplifier Settling Time} + \text{Hold Capacitor Charging Time} + \text{Temperature Coefficient} \\ &= TAMP + TC + TCOFF \\ &= 5\mu s + TC + [(\text{Temperature} - 25°C)(0.05\mu s/°C)] \end{aligned}$$

The value for TC can be approximated with the following equations:

$$V_{APPLIED}\left(1 - \frac{1}{2047}\right) = V_{CHOLD} \quad ;[1]\ V_{CHOLD}\ \text{charged to within 1/2 lsb}$$

$$V_{APPLIED}\left(1 - e^{\frac{-TC}{RC}}\right) = V_{CHOLD} \quad ;[2]\ V_{CHOLD}\ \text{charge response to}\ V_{APPLIED}$$

$$V_{APPLIED}\left(1 - e^{\frac{-Tc}{RC}}\right) = V_{APPLIED}\left(1 - \frac{1}{2047}\right) \quad ;\text{combining [1] and [2]}$$

Solving for TC:

$$\begin{aligned} TC &= -C_{HOLD}(RIC + RSS + RS)\ ln(1/2047) \\ &= -13.5pF(1k\Omega + 700\Omega + 10k\Omega)\ ln(0.0004885) \\ &= 1.20\mu s \end{aligned}$$

Therefore:

$$\begin{aligned} TACQ &= 5\mu s + 1.20\mu s + [(50°C - 25°C)(0.05\mu s/°C)] \\ &= 7.45\mu s \end{aligned}$$

4.5 The PIC 18F ADC converter.

Other ADC time requirements:

TABLE 27-22: A/D CONVERSION REQUIREMENTS PIC18(L)F2X/4XK22

Standard Operating Conditions (unless otherwise stated) Operating temperature Tested at +25°C							
Param. No.	**Symbol**	**Characteristic**	**Min**	**Typ**	**Max**	**Units**	**Conditions**
130	TAD	A/D Clock Period	1	—	25	µs	-40°C to +85°C
			1	—	4	µs	+85°C to +125°C
131	TCNV	Conversion Time (not including acquisition time) **(Note 1)**	11	—	11	TAD	
132	TACQ	Acquisition Time **(Note 2)**	1.4	—	—	µs	VDD = 3V, Rs = 50Ω
135	TSWC	Switching Time from Convert → Sample	—	—	**(Note 3)**		
136	TDIS	Discharge Time	1	—	1	TCY	

Note 1: ADRES register may be read on the following TCY cycle.

2: The time for the holding capacitor to acquire the "New" input voltage when the voltage changes full scale after the conversion (VDD to VSS or VSS to VDD). The source impedance (*Rs*) on the input channels is 50 Ω.

3: On the following cycle of the device clock.

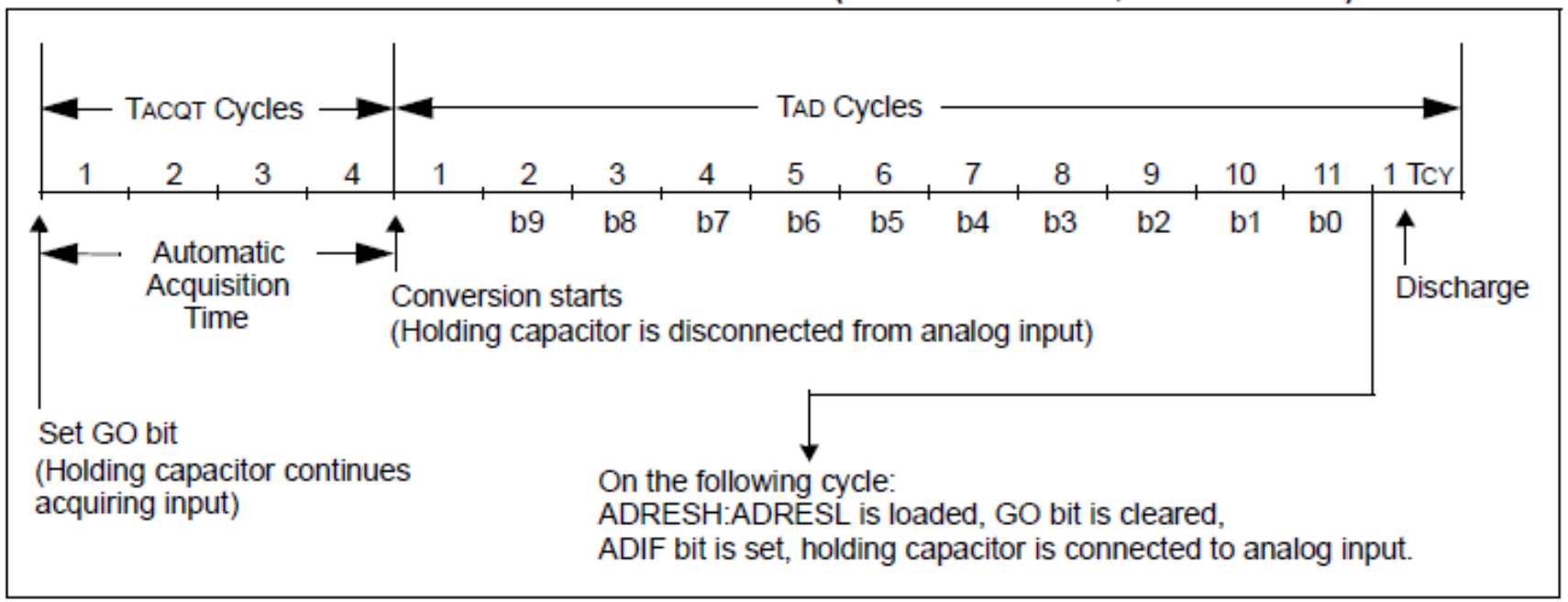
Summary: $T_{AD} \geq 1\ \mu s$ $T_{ACQ} > 1.4\ \mu s$ (but $T_{ACQ} > 7.45\ \mu s$ in the example)

$T_{CNV} = 11$, we need 11 T_{AD} for conversion. $T_{DIS} = 1$ Cycle

4.5 The PIC 18F ADC converter.

ADC Timing example for TACQ = 4 TAD

FIGURE 17-4: A/D CONVERSION TAD CYCLES (ACQT<2:0> = 010, TACQ = 4 TAD)



If we set a TAD of 4us, sampling time will be 64us, so maximum sampling frequency will be 15,6KHz

4.5 The PIC 18F ADC converter.

Procedure for AD conversion: manufacturer's recipe. →

EXAMPLE 17-1: A/D CONVERSION

```
;This code block configures the ADC
;for polling, Vdd and Vss as reference, Frc
clock and AN0 input.
;
;Conversion start & polling for completion
; are included.
;
MOVLW     B'10101111'  ;right justify, Frc,
MOVWF     ADCON2       ; & 12 TAD ACQ time
MOVLW     B'00000000'  ;ADC ref = Vdd,Vss
MOVWF     ADCON1       ;
BSF       TRISA,0      ;Set RA0 to input
BSF       ANSEL,0      ;Set RA0 to analog
MOVLW     B'00000001'  ;AN0, ADC on
MOVWF     ADCON0       ;
BSF       ADCON0,GO    ;Start conversion
ADCPoll:
BTFSC     ADCON0,GO    ;Is conversion done?
BRA       ADCPoll      ;No, test again
; Result is complete - store 2 MSbits in
; RESULTHI and 8 LSbits in RESULTLO
MOVFF     ADRESH,RESULTHI
MOVFF     ADRESL,RESULTLO
```

17.2.10 A/D CONVERSION PROCEDURE

This is an example procedure for using the ADC to perform an Analog-to-Digital conversion:

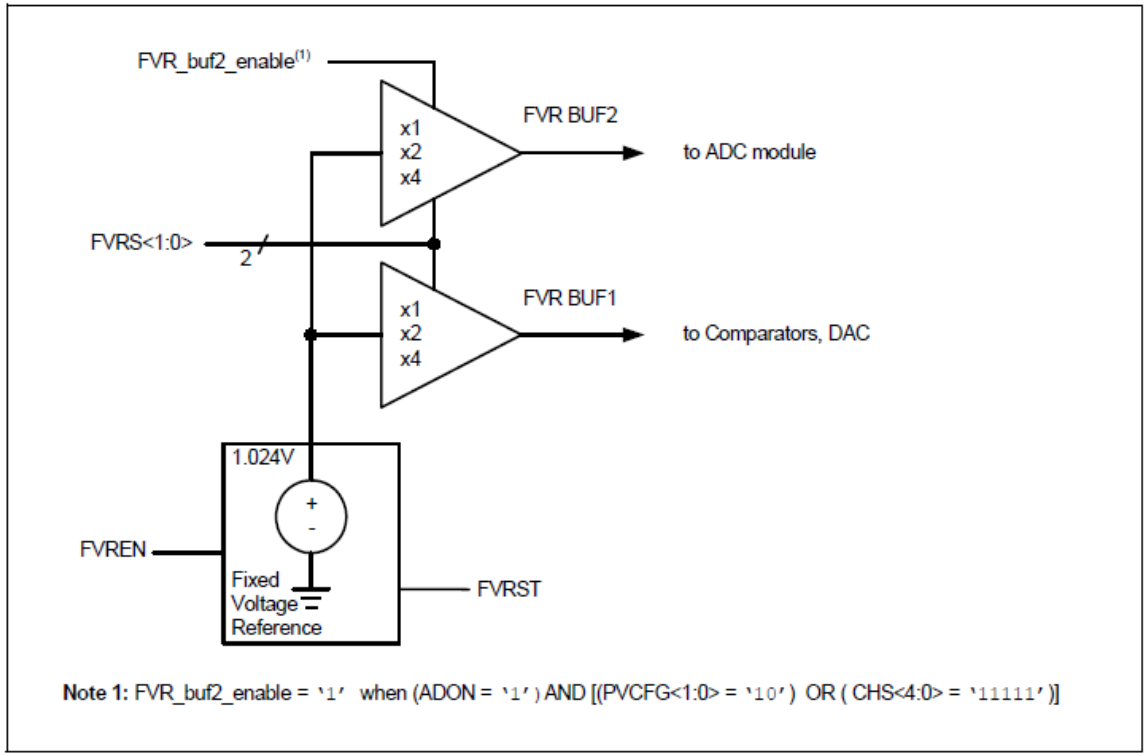
1. Configure Port:
   - Disable pin output driver (See TRIS register)
   - Configure pin as analog
2. Configure the ADC module:
   - Select ADC conversion clock
   - Configure voltage reference
   - Select ADC input channel
   - Select result format
   - Select acquisition delay
   - Turn on ADC module
3. Configure ADC interrupt (optional):
   - Clear ADC interrupt flag
   - Enable ADC interrupt
   - Enable peripheral interrupt
   - Enable global interrupt(1)
4. Wait the required acquisition time(2).
5. Start conversion by setting the GO/$\overline{\text{DONE}}$ bit.
6. Wait for ADC conversion to complete by one of the following:
   - Polling the GO/$\overline{\text{DONE}}$ bit
   - Waiting for the ADC interrupt (interrupts enabled)
7. Read ADC Result
8. Clear the ADC interrupt flag (required if interrupt is enabled).

4.6 The PIC 18F Fixed Voltage Reference (FVR).

FVR units (FVR1, FVR2) can provide a voltage independent of the supply voltage (VDD) that can be used as an absolute reference systems.

ADC and DAC units can require a Fixed Voltage for precise operation.

FIGURE 21-1: VOLTAGE REFERENCE BLOCK DIAGRAM



Note 1: FVR_buf2_enable = '1' when (ADON = '1') AND [(PVCFG<1:0> = '10') OR (CHS<4:0> = '11111')]

FVRCON0 register manages the FVR unit.

- FVREN bit enables the unit.

- FVRST bit is set when the circuitry reaches a stable output.

REGISTER 21-1: VREFCON0: FIXED VOLTAGE REFERENCE CONTROL REGISTER

R/W-0	R/W-0	R/W-0	R/W-1	U-0	U-0	U-0	U-0
FVREN	FVRST	FVRS<1:0>		—	—	—	—
bit 7							bit 0

Legend:		
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
u = Bit is unchanged	x = Bit is unknown	-n/n = Value at POR and BOR/Value at all other Resets
'1' = Bit is set	'0' = Bit is cleared	

bit 7 **FVREN:** Fixed Voltage Reference Enable bit
0 = Fixed Voltage Reference is disabled
1 = Fixed Voltage Reference is enabled

bit 6 **FVRST:** Fixed Voltage Reference Ready Flag bit
0 = Fixed Voltage Reference output is not ready or not enabled
1 = Fixed Voltage Reference output is ready for use

bit 5-4 **FVRS<1:0>:** Fixed Voltage Reference Selection bits
00 = Fixed Voltage Reference Peripheral output is off
01 = Fixed Voltage Reference Peripheral output is 1x (1.024V)
10 = Fixed Voltage Reference Peripheral output is 2x (2.048V)(1)
11 = Fixed Voltage Reference Peripheral output is 4x (4.096V)(1)

bit 3-2 **Reserved:** Read as '0'. Maintain these bits clear.

bit 1-0 **Unimplemented:** Read as '0'.

Note 1: Fixed Voltage Reference output cannot exceed VDD.