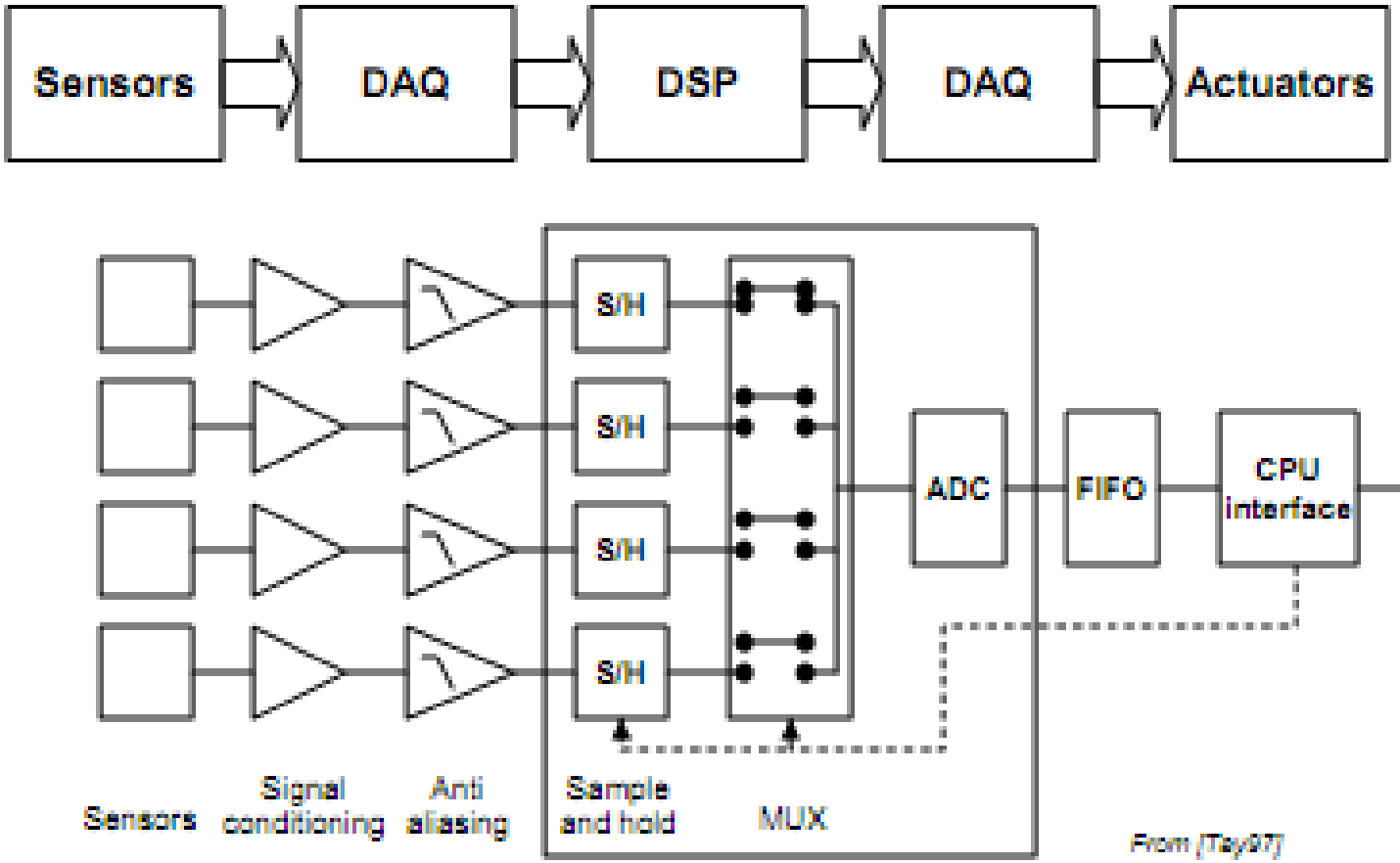
Data Acquisition II

- Sample and hold
- Multiplexing
- Analog to digital conversion
- Digital to analog conversion

Architecture of data acquisition systems



ANALOG ⇔ DIGITAL CONVERSION

Physical world is analog (mostly)

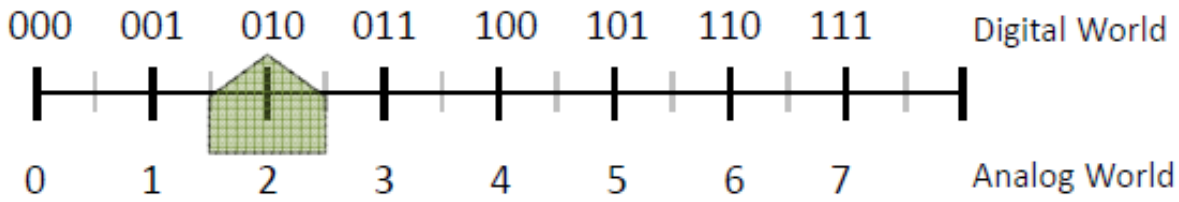
Analog is infinitely variable, while digital is discrete in both time and value

Analog/digital conversion can be part of the instrument or part of the actuator, e.g.

- Incremental encoder – analog position gets encoded into digital pulse
- Stepper motor – digital step pulse gets transformed to analog position

ANALOG ⇔ DIGITAL CONVERSION

To be processed by computers, all information must be converted to a binary representation possessing a finite number of distinct values that combine logical low (0) and logical high (1):



Digital coding is arbitrary, but natural binary order is most often used

ANALOG ⇔ DIGITAL CONVERSION

For notational compactness and ease of reading, binary (base 2) representations are often expressed in octal (base 8) or hexadecimal (base 16):

$$10100110_2 = 0246_8 = 166_{10} = A6_{16}$$

Arduino code:

Binary:	`B10100110`	(leading '`B`', only 8-bit values)
Octal:	`0246`	(leading '`0`')
Decimal:	`166`	(no formatting needed)
Hexadecimal:	`0xA6`	(leading "`0x`", chars `0-9`, `A-F`, `a-f`)

DIGITAL REPRESENTATION

When using natural binary order, the digital code consists of an *N*-bit binary word:

$$b_{N-1}\ b_{N-2}\ b_{N-3} \ldots b_1\ b_0$$

where b_{N-1} is the most-significant bit (MSB) and b_0 is the least-significant bit (LSB)

8-bit digital word: ***byte*** (or octet).

4-bit digital word: ***nibble*** (or quartet).

DIGITAL REPRESENTATION

The fractional value of the digital word is:

$$2^{-1}\, b_{N-1} + 2^{-2}\, b_{N-2} + 2^{-3}\, b_{N-3} + \ldots + 2^{-N}\, b_o$$

with a range of $0 \longrightarrow (1 - 2^{-N})$ and a precision of 2^{-N}

Digital fractions sometimes notated with leading point notation:

$$.1011_2 = .6875_{10}$$

Example:

- 4-bits
- Range $= 0 \longrightarrow (1 - 2^{-4}) = 0 \longrightarrow \frac{15}{16}$ (nearly 1)
- Precision (absolute and relative) $= 2^{-4} = \frac{1}{16}$
- Code of 1011 $= \frac{1}{2} + \frac{1}{8} + \frac{1}{16} = \frac{11}{16} = .6875$

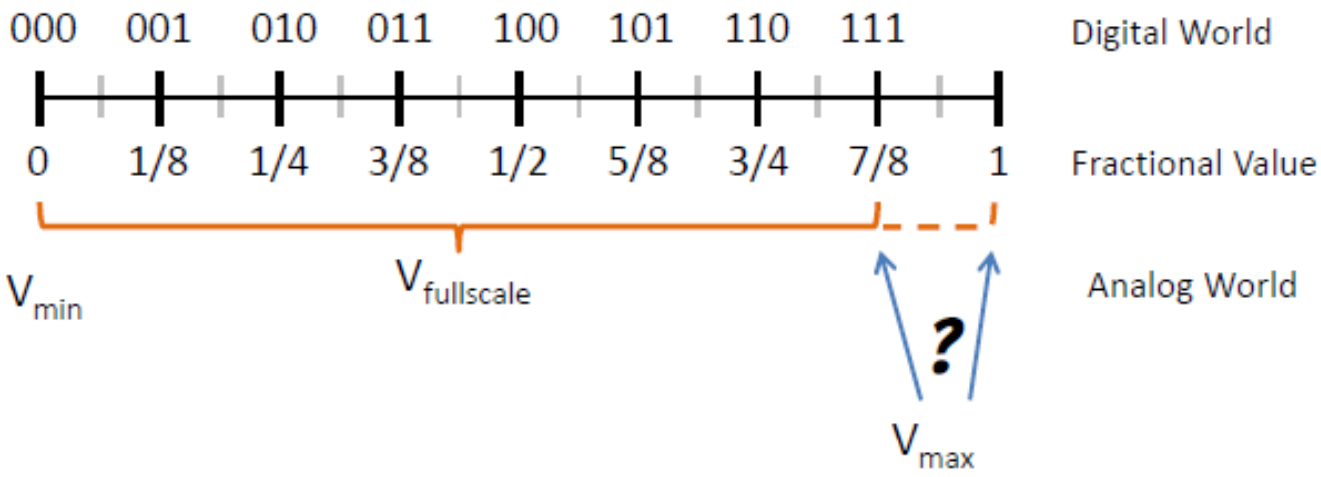
DIGITAL REPRESENTATION

- Digital codes have no inherent units, so scaling must be defined for code to have real-world meaning.

- How to assign relationship between binary code and analog values? It depends on the application…

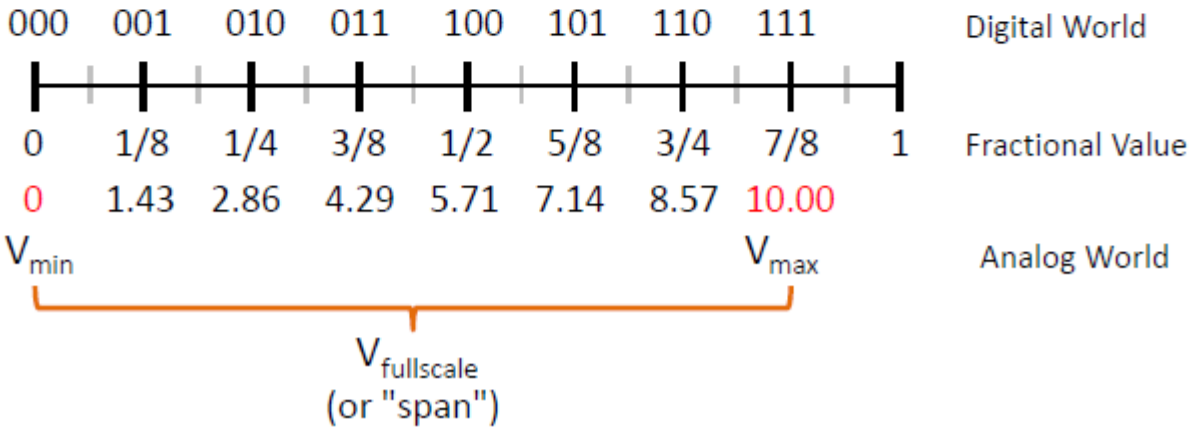
DIGITAL REPRESENTATION

For a unipolar signal, it seems obvious to associate V_{min} with binary zero. Do we associate V_{max} with binary 2^N (fractional value of 1), or with binary 2N-1 (and its fraction value of $1-2^N$)?



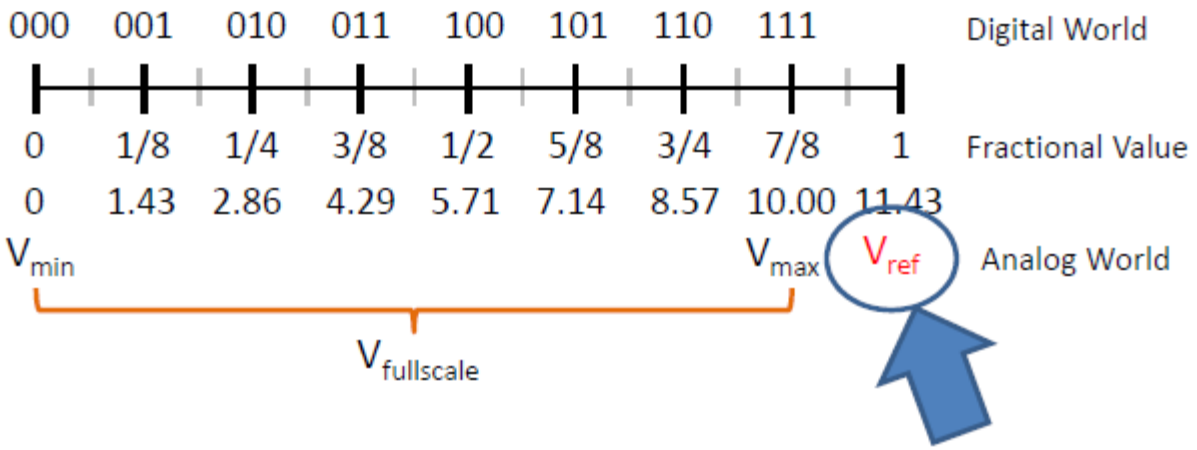
DIGITAL REPRESENTATION

To maximize the conversion range for N-bit coding of a *unipolar* value, we often associate V_{min} with binary zero and V_{max} with binary 2^N-1. For example, for a 0-10V signal, we may assign values such that:



DIGITAL REPRESENTATION

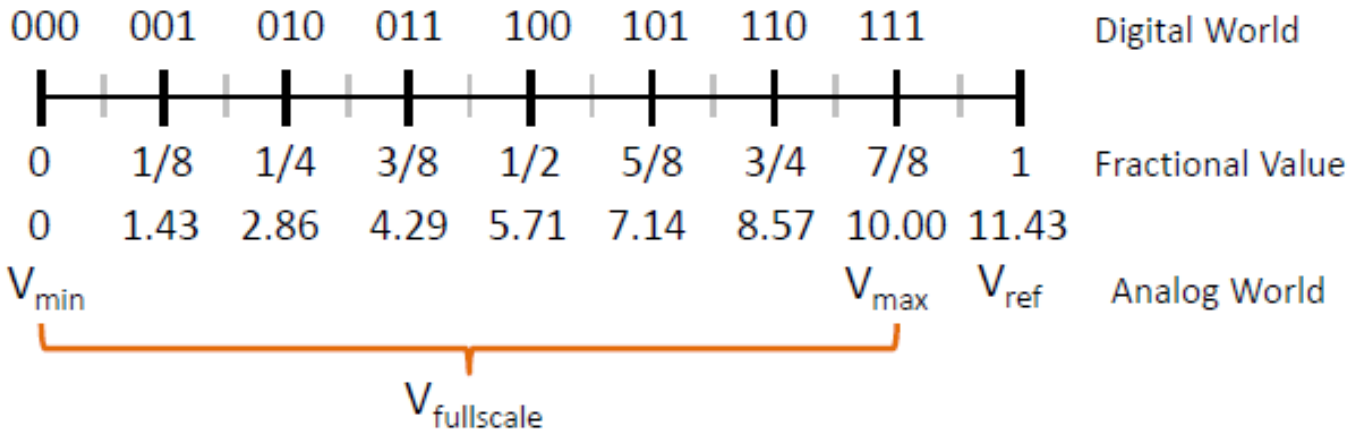
However, it is common to associate a reference value of V_{ref} with the digital code used to denote a fraction of 1



DIGITAL REPRESENTATION

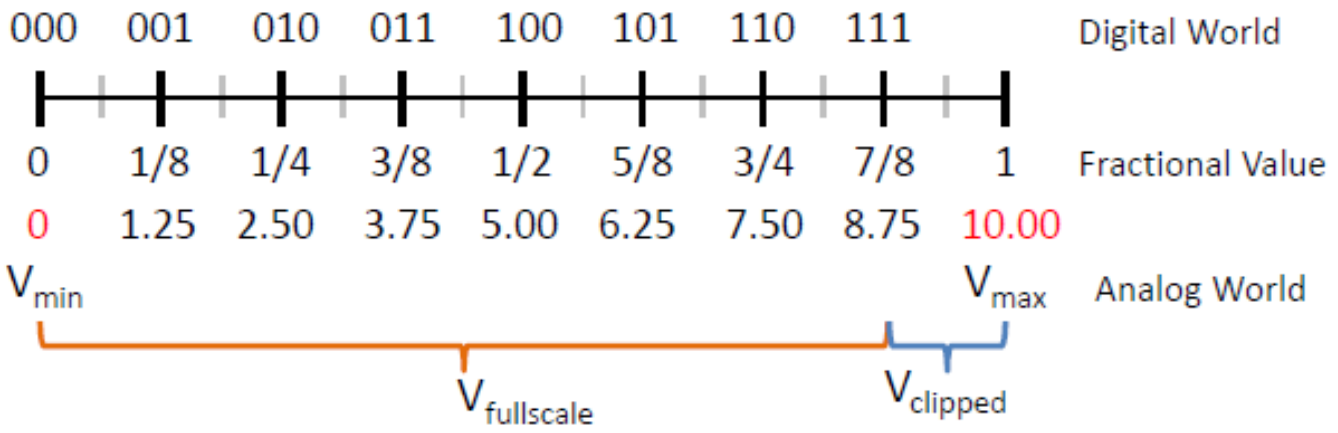
In such an arrangement, with V_{min} = o,

$$V_{\mathrm{fs}} = V_{\mathrm{max}} = V_{\mathrm{ref}}\left(\frac{2^N - 1}{2^N}\right)$$



DIGITAL REPRESENTATION

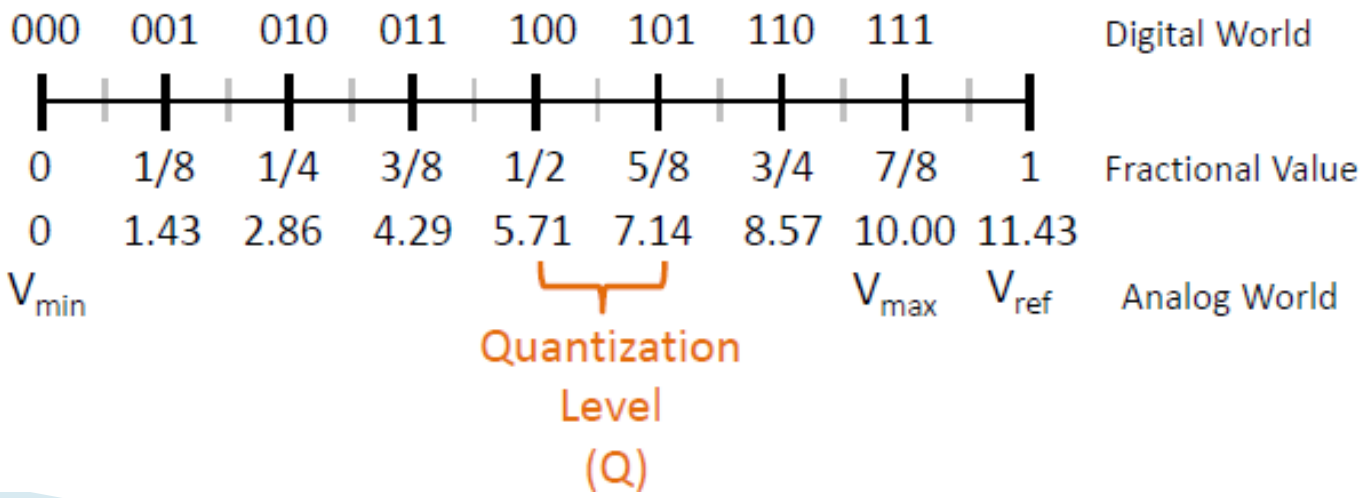
If V_{min} is less than binary zero, or V_{max} is more than binary 2^N-1, then *amplitude clipping* may result.



DIGITAL REPRESENTATION

Quantization Level: change in analog output associated with each discrete step

- We assume this step size to be fixed, although in sophisticated applications, it may vary across the signal range, or adapt to signal characteristics.



DIGITAL REPRESENTATION

Quantization Level (Q)

- Signal change associated with LSB
- For an N-bit converter:
  - $Q = \frac{V_{\mathbf{fullscale}}}{2^N - 1} = \frac{V_{\mathbf{ref}}}{2^N}$
- Quantization ***error*** ranges from $-\frac{Q}{2}$ to $+\frac{Q}{2}$

Example:

- 4-bits, 12 volts full-scale, binary code of 1001
- Q = 12V / $(2^N - 1)$ = .8V
- $|Q_{error}| \leq .4V$
- V_{analog} = Q × (8·1 + 4·0 + 2·0 + 1) = .8V × 9 = 7.2V

DIGITAL REPRESENTATION

Quantization

- An irreversible process
- A source of information loss
- Increasing the number of bits lowers information loss, but usually raises the cost and processing time

Example:

- 10 volts full-scale, with 1% precision required
- $Q = 1\% \cdot 10 = 0.1V$
- $N \geq \log_2 [V_{FS}/Q + 1] = 6.7$ bits $\Rightarrow$ at least 7 bits

INTEGER CODES

Unipolar Voltages

- Can be coded to unsigned integers
- Example – 0-5 volts coded to 3 bit unsigned integer

Voltage	Digital Value	Decimal Equivalent
0	000	0
0.71	001	1
1.43	010	2
2.14	011	3
2.86	100	4
3.57	101	5
4.29	110	6
5	111	7

INTEGER CODES

Bipolar Coding

- Two's Complement

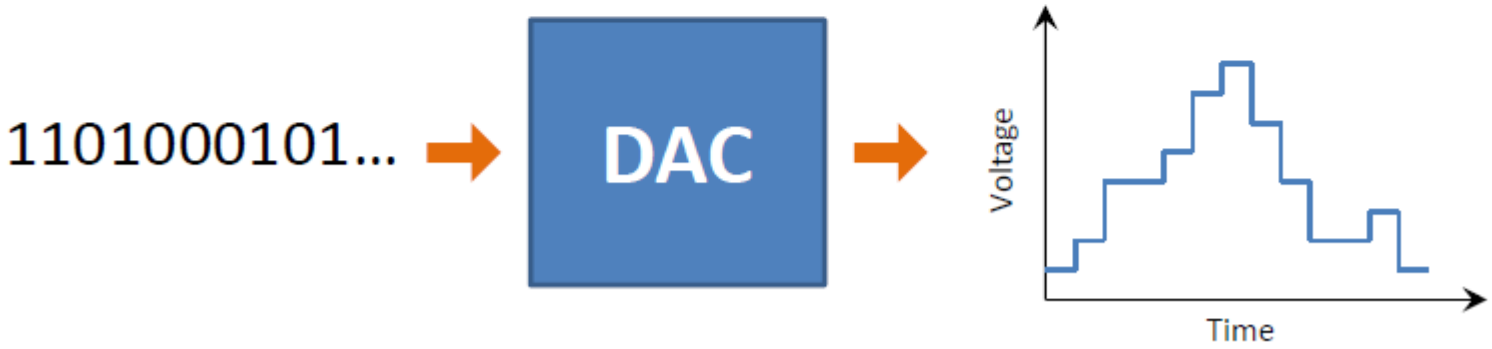
Voltage(1)	Voltage(2)	Digital Value	Decimal Equivalent
+3.75	+5	011	3
+2.50	+3.33	010	2
+1.25	+1.67	001	1
0	0	000	0
-1.25	-1.67	111	-1
-2.50	-3.33	110	-2
-3.75	-5	101	-3
-5	--	100	-4

- MSB is treated with a weighting of -2^{N-1}

$$-3 = -2^{(3-1)} + 1 = -2^2 + 1 = -4 + 1 \rightarrow 101$$

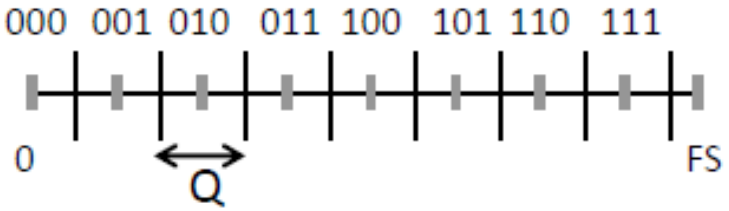
DIGITAL-TO-ANALOG CONVERTER (DAC)

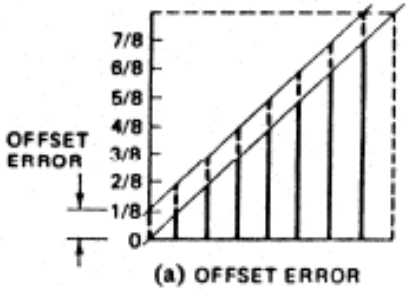
Converts digital values to analog outputs of either voltage or current



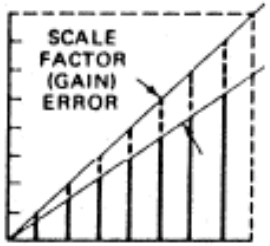
DIGITAL-TO-ANALOG (D/A) CONVERSION

Ideal DA Conversion:

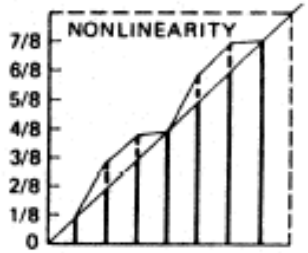




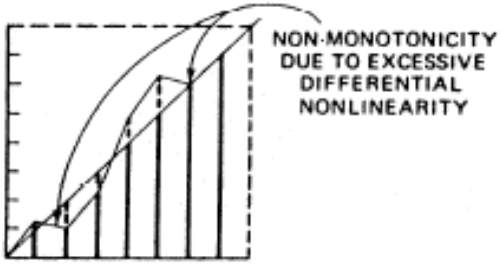
(a) OFFSET ERROR



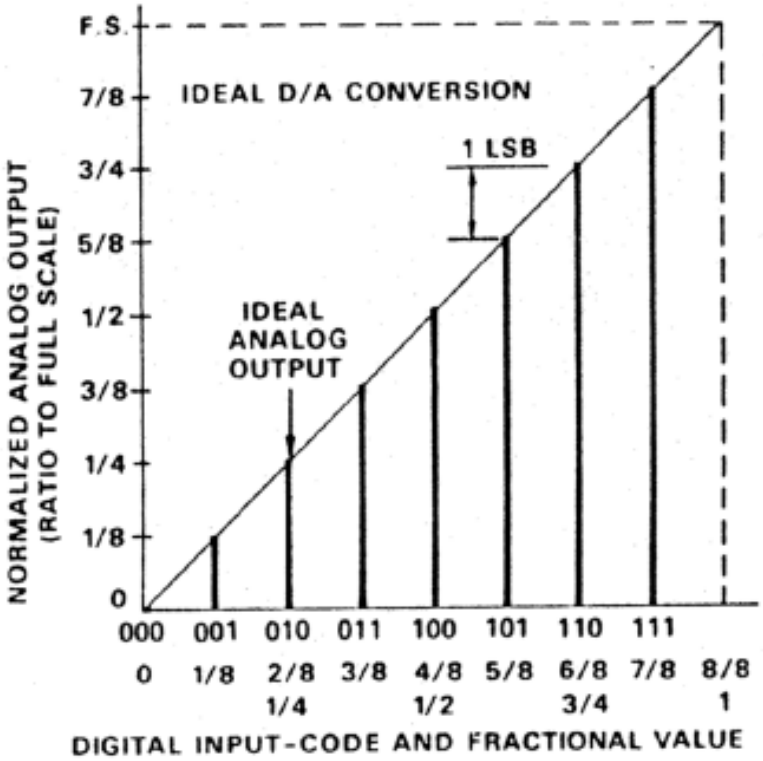
(b) SCALE FACTOR ERROR



(c) LINEARITY ERROR



(d) NON-MONOTONICITY (DUE TO EXCESSIVE DIFFERENTIAL NONLINEARITY)



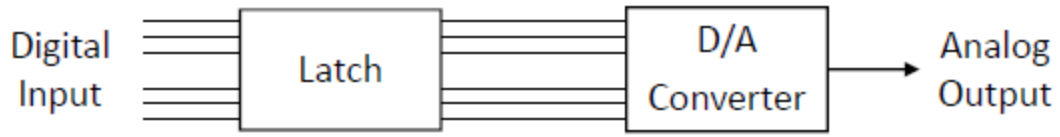
DIGITAL-TO-ANALOG (D/A) CONVERSION

Digital value is stored in a register (latch), then converted. Duration of conversion is called *settling time*.

Output of the DAC remains the same until the next value is sent to the register (latch) – a zero-order hold.

Basic concept:

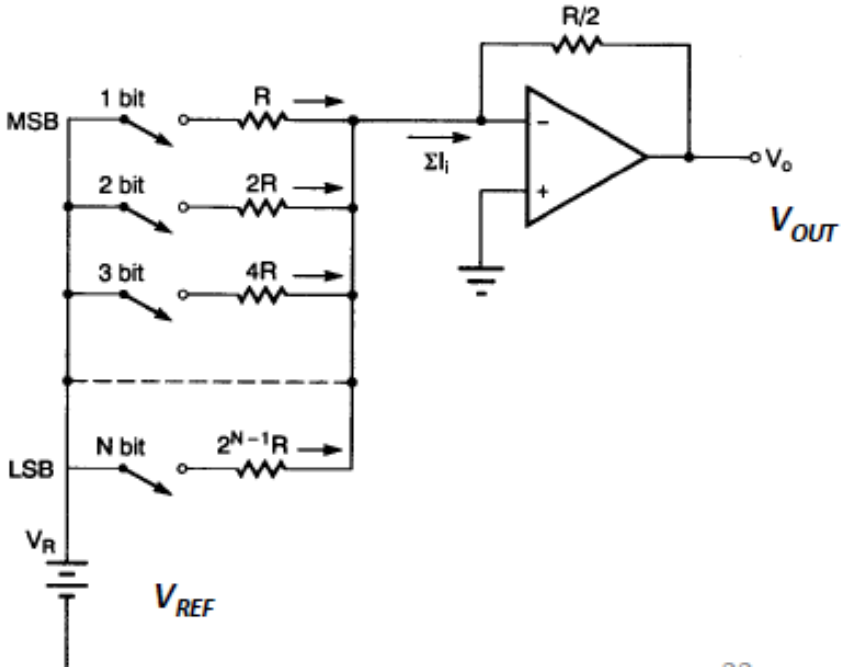
$$V_{OUT} = \underbrace{(b_{N-1} \cdot 2^{N-1} + b_{N-2} \cdot 2^{N-2} + \cdots + b_1 \cdot 2^1 + b_0 \cdot 2^0)}_{\text{Integer equivalent of binary code}} \cdot Q$$



DIGITAL-TO-ANALOG (D/A) CONVERSION

Weighted (Scaled) Resistor DAC

- Fast, but not practical for high bit count due to expense of high precision resistors across a wide magnitude range
- Virtual ground at inverting op-amp input
- Requires
  - Accurate reference voltage.
  - Higher precision resistor.



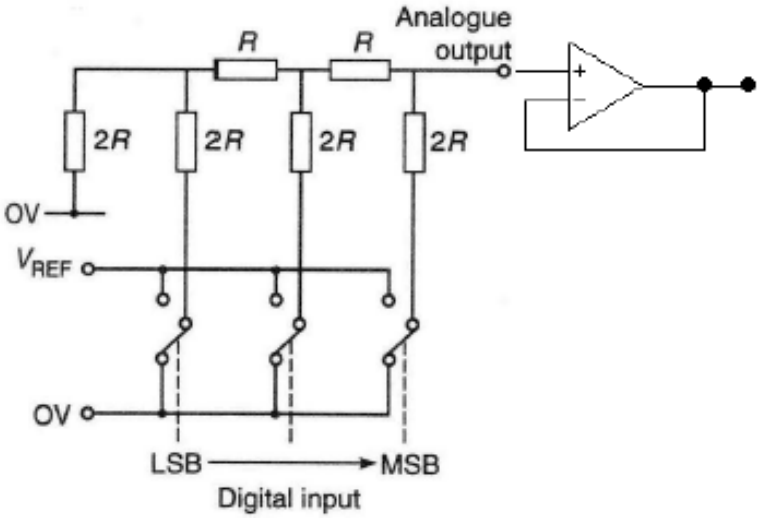
$$V_{OUT} = -\Sigma I_i \frac{R}{2} = -V_{\text{REF}}\left(\frac{b_{N-1}}{2} + \frac{b_{N-2}}{4} + \cdots + \frac{b_0}{2^N}\right)$$

Can $|V_{\text{OUT}}| = |V_{\text{REF}}|$?

DIGITAL-TO-ANALOG (D/A) CONVERSION

R/2R Ladder DAC

- Uses just two resistance values (2R and R, closely matched)
- Input switches define a specific resistor divider network



If only MSB (100) is asserted:

$$V_{OUT} = \frac{4}{8} \cdot V_{REF} = \frac{1}{2} \cdot V_{REF}$$

If all bits are asserted (111):

$$V_{OUT} = \frac{7}{8} \cdot V_{REF}$$

$$V_{OUT} = (b_{N-1} \cdot 2^{N-1} + b_{N-2} \cdot 2^{N-2} + \cdots + b_1 \cdot 2^1 + b_0 \cdot 2^0) \cdot \frac{V_{REF}}{2^N}$$

- Bipolar output can be achieved by substituting the ground with a negative voltage source.

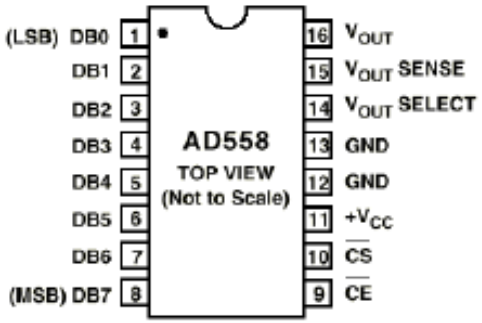
DIGITAL-TO-ANALOG (D/A) CONVERSION

Multiplying DAC

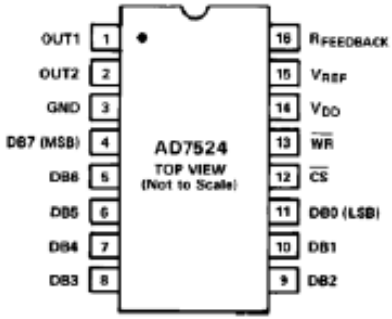
- Conventional DAC has internal reference voltage V_{REF} that is derived from the fixed power supply.
- Multiplying DAC has an externally supplied reference voltage.

$$V_{OUT} = (b_{N-1} \cdot 2^{N-1} + b_{N-2} \cdot 2^{N-2} + \cdots + b_1 \cdot 2^1 + b_0 \cdot 2^0) \cdot \frac{V_{REF}}{2^N}$$

- Advantages:
  - Use a constant frequency sinusoidal reference signal to achieve amplitude modulation, i.e. let $V_{REF} = V_R \sin(\omega t)$.
  - External V_{REF} can be precisely controlled to compensate for drift.



Internally Referenced



Multiplying DAC
(Externally Referenced)

INTERFACING WITH A DAC

- Non-multiplying DAC – use AD558 (8 bit DAC) as example

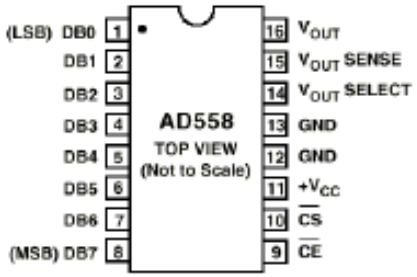
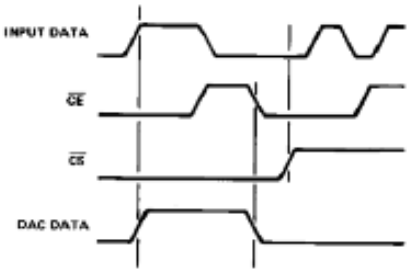
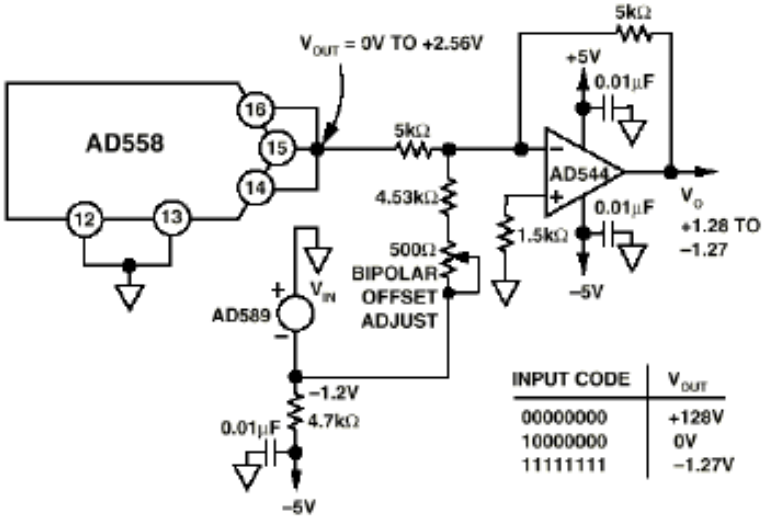


Table I. AD558 Control Logic Truth Table

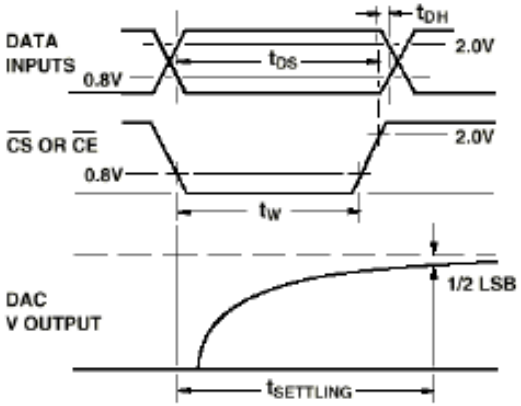
Input Data	$\overline{CE}$	$\overline{CS}$	DAC Data	Latch Condition
0	0	0	0	"Transparent"
1	0	0	1	"Transparent"
0	g	0	0	Latching
1	g	0	1	Latching
0	0	g	0	Latching
1	0	g	1	Latching
X	1	X	Previous Data	Latched
X	X	1	Previous Data	Latched

NOTES
X = Does not matter.
g = Logic Threshold at Positive-Going Transition.





INPUT CODE	V_{OUT}
00000000	+128V
10000000	0V
11111111	–1.27V



t_W = STORAGE PULSE WIDTH = 200ns MIN
t_{DH} = DATA HOLD TIME = 10ns MIN
t_{DS} = DATA SETUP TIME = 200ns MIN

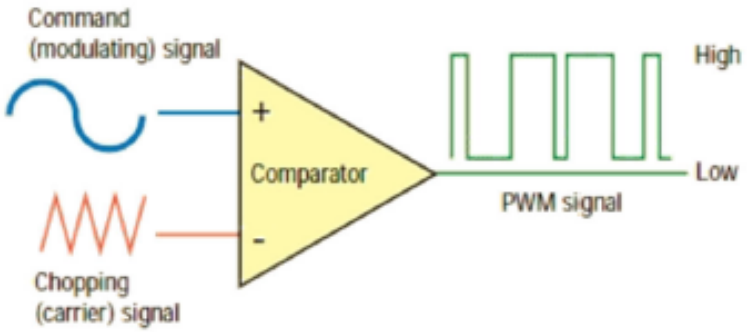
DIGITAL-TO-ANALOG (D/A) CONVERSION

Resistor methods rely on voltage dividers

- Many precision resistors necessary
- Wasted energy dissipated as heat

Pulse Width Modulation (PWM)

- Rectangular pulse wave
- Duty cycle controls average voltage
- Very high frequency
- Need a low-pass filter to remove the sharp transitions at edges of the pulses!
- About 90% efficiency



DIGITAL-TO-ANALOG (D/A) CONVERSION

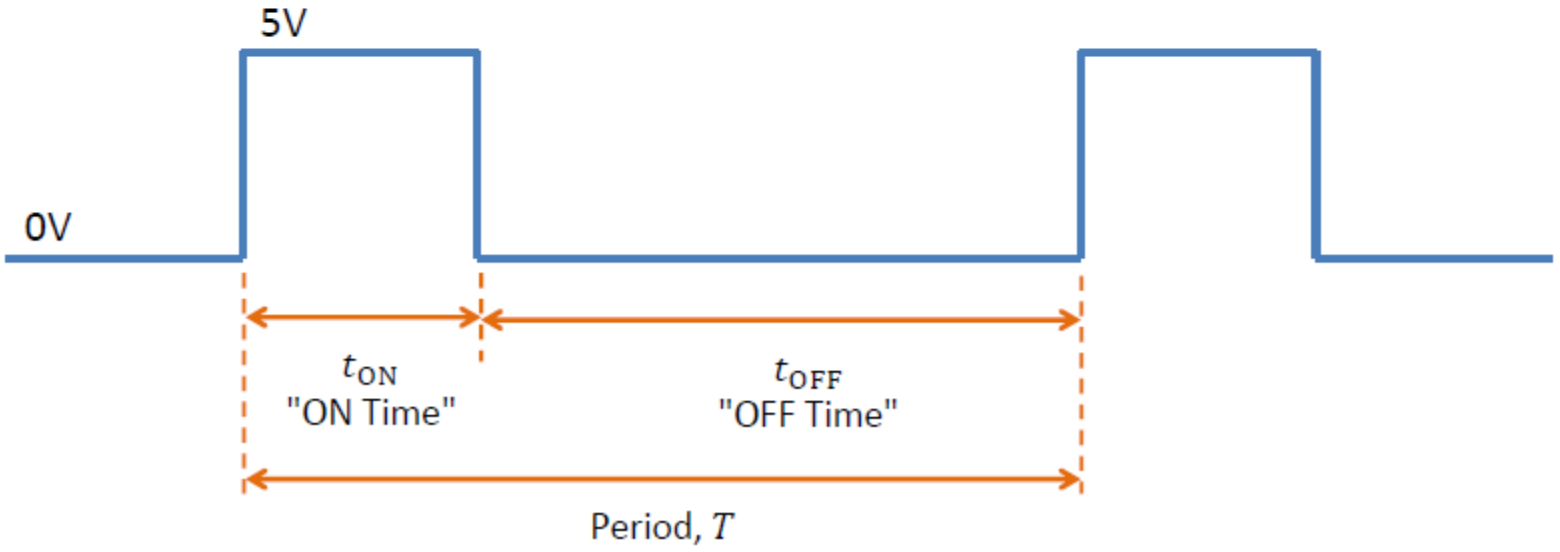
Pulse Width Modulation (PWM)

- Poor man's DAC
- Low pass filtering the PWM signal can produce an analog signal whose magnitude is proportional to the pulse width of the PWM signal
- For motor/motion control, the motor/motion system will act as the low pass filter
- Unipolar output
- Best suited when an analog output is needed but does not require a high resolution DAC

PULSE WIDTH MODULATION (PWM)

$$\text{Duty Cycle} = \frac{t_{ON}}{T} \Rightarrow 0 - 100\%$$

$$\text{Frequency (rad/sec)} = \frac{1}{T}$$



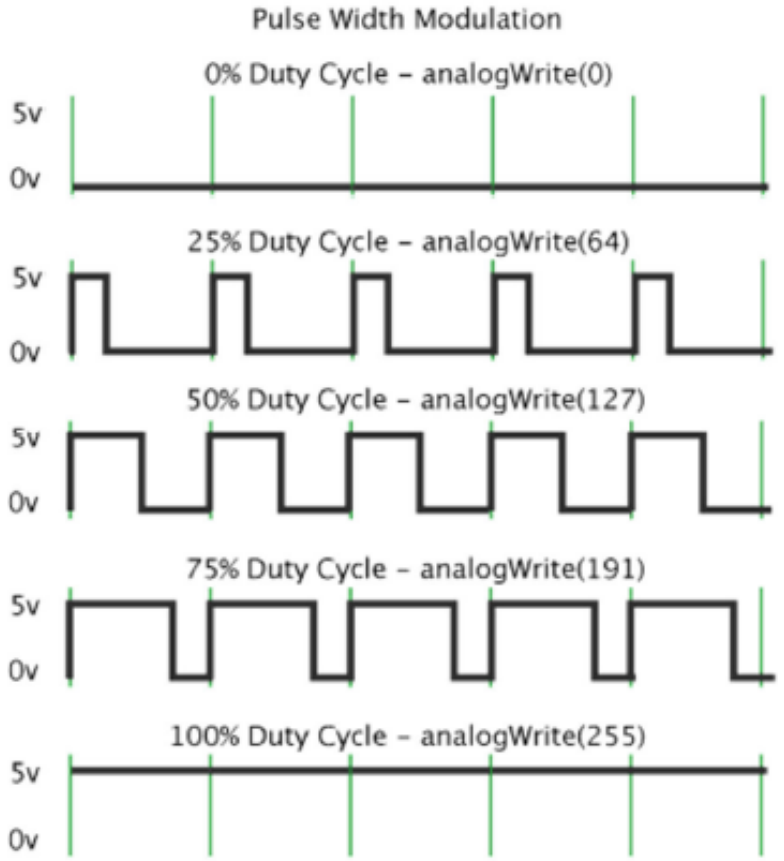
PULSE WIDTH MODULATION (PWM)

8-bit resolution:

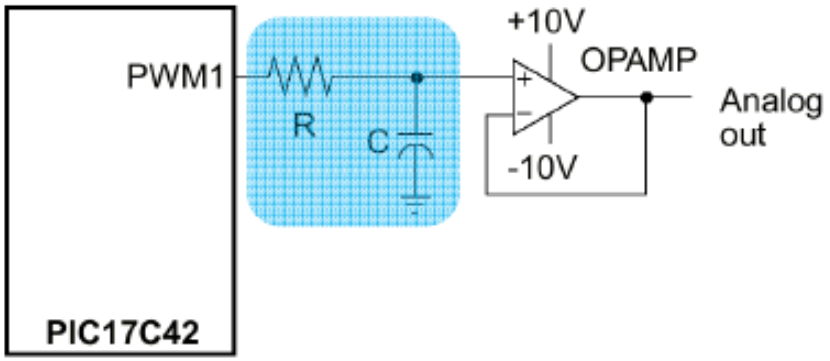
- 100%/255 => 0.39% per step
- 5V/255 => 19.6 mV per step

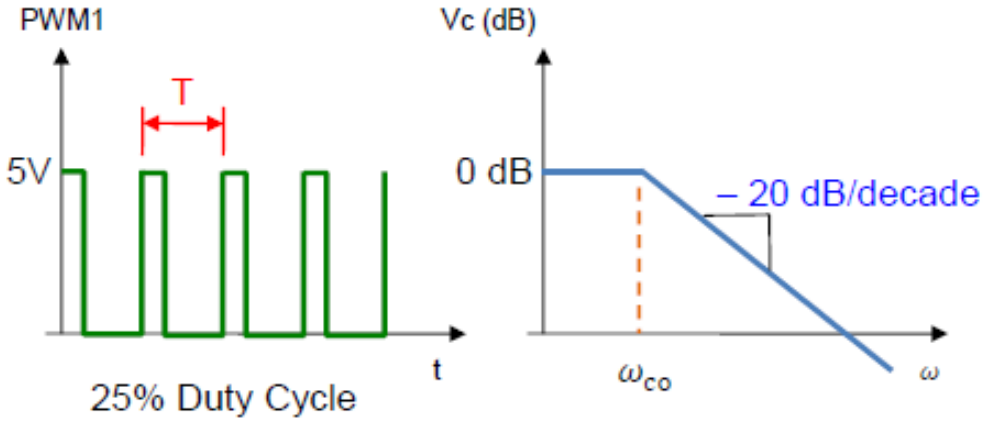
Arduino default PWM frequency:

- Pins 5/6: ~976 Hz
- Pins 3/9/10/11: ~488 Hz
- Frequency can be increased to as much as 62.5 kHz by altering timer control registers
  - Pins 5/6: `TCCR0B`
  - Pins 9/10: `TCCR1B`
  - Pins 3/11: `TCCR2B`

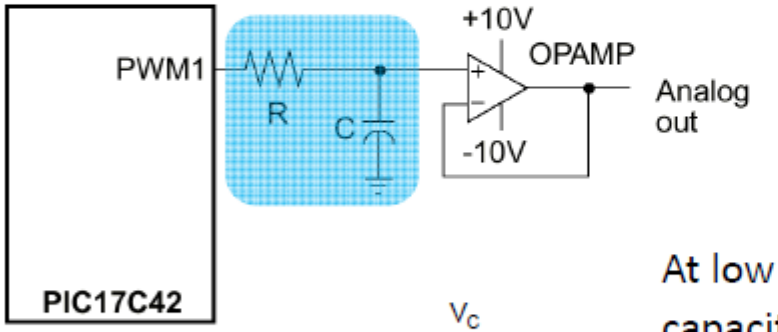


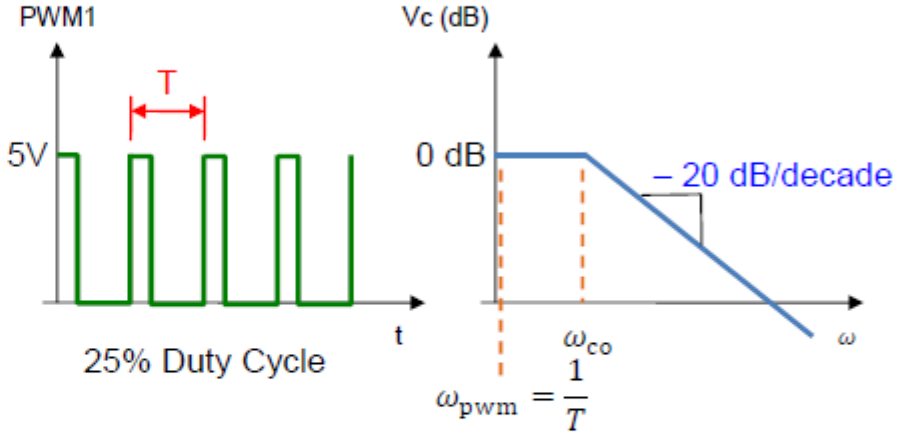
DIGITAL-TO-ANALOG (D/A) CONVERSION



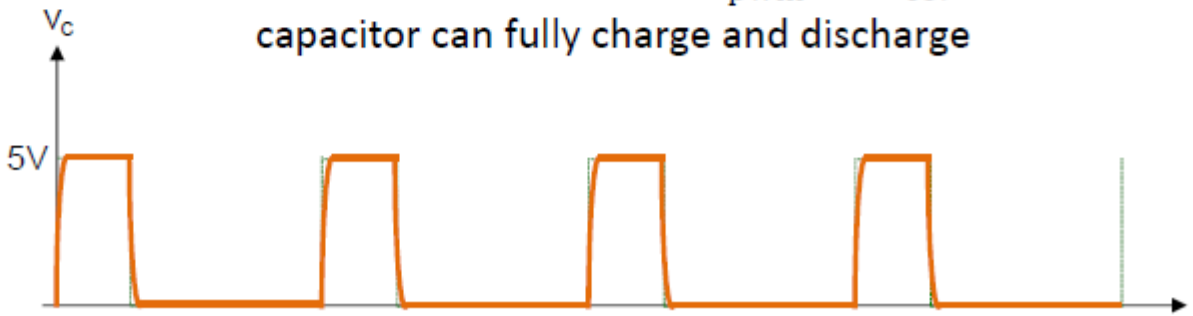


DIGITAL-TO-ANALOG (D/A) CONVERSION

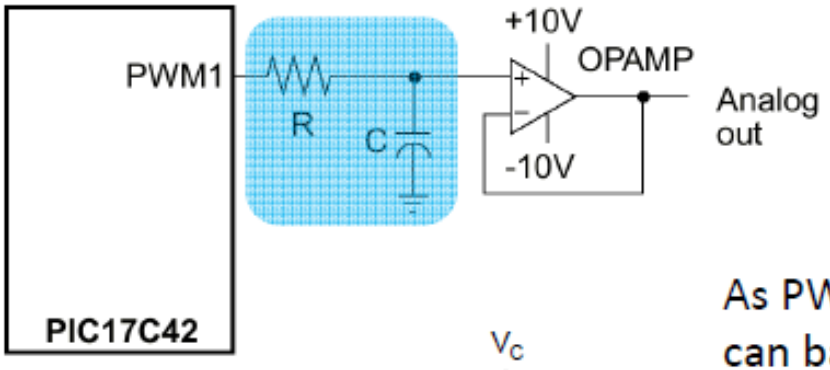


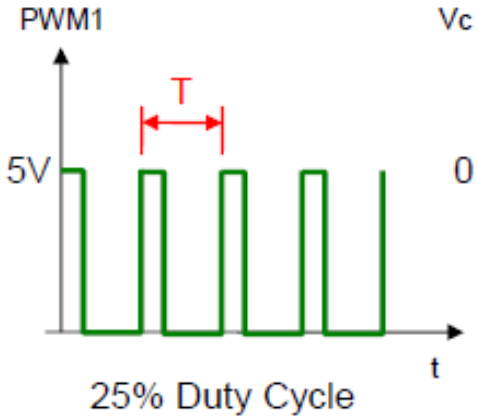


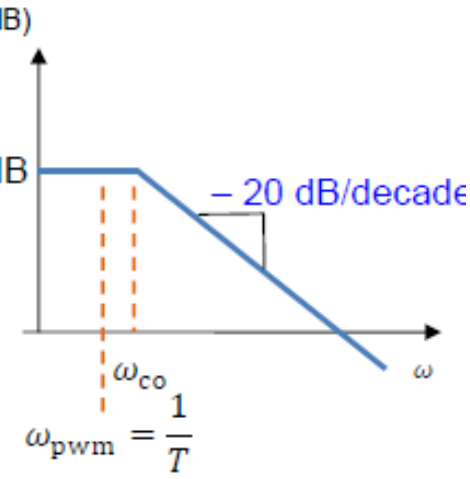
At low PWM frequency ($\omega_{pwm} \ll \omega_{co}$), capacitor can fully charge and discharge



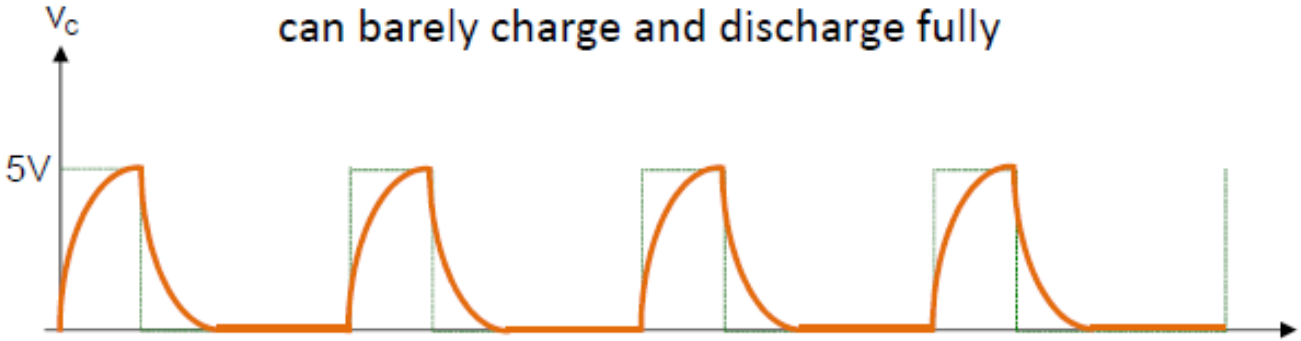
DIGITAL-TO-ANALOG (D/A) CONVERSION



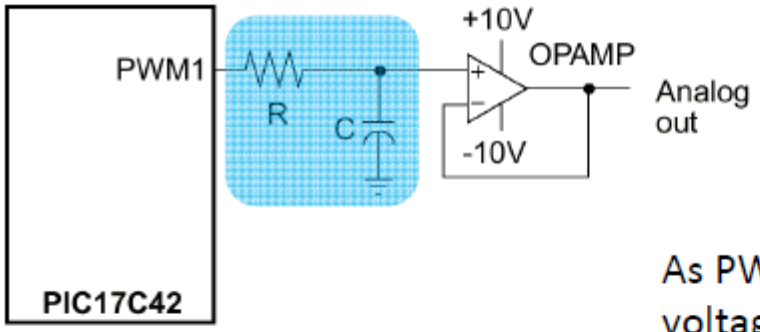


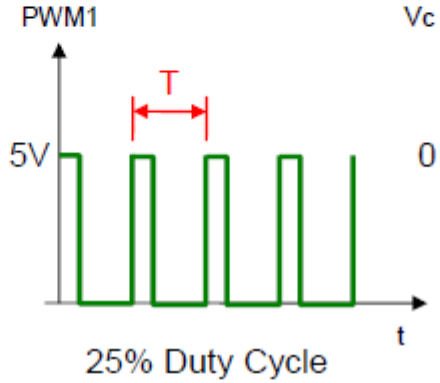


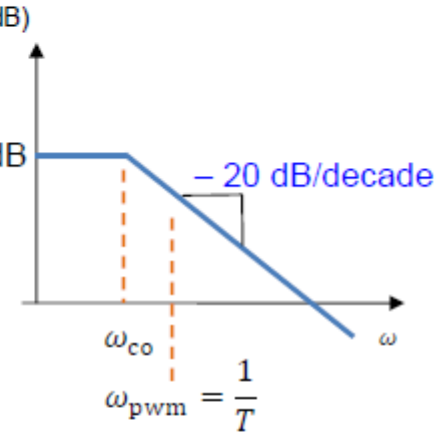
As PWM frequency increases, capacitor can barely charge and discharge fully



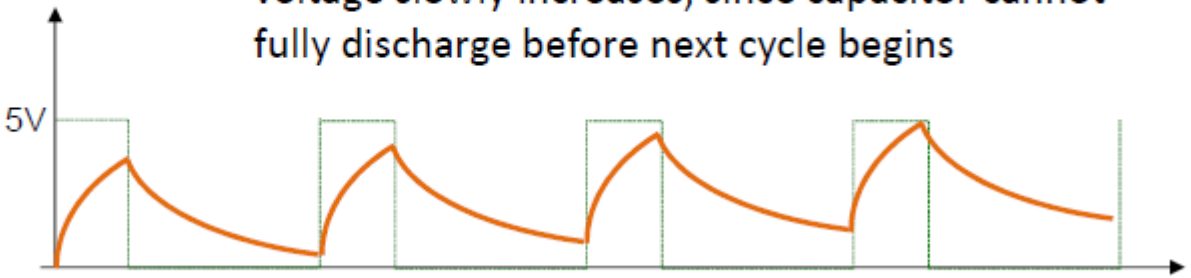
DIGITAL-TO-ANALOG (D/A) CONVERSION



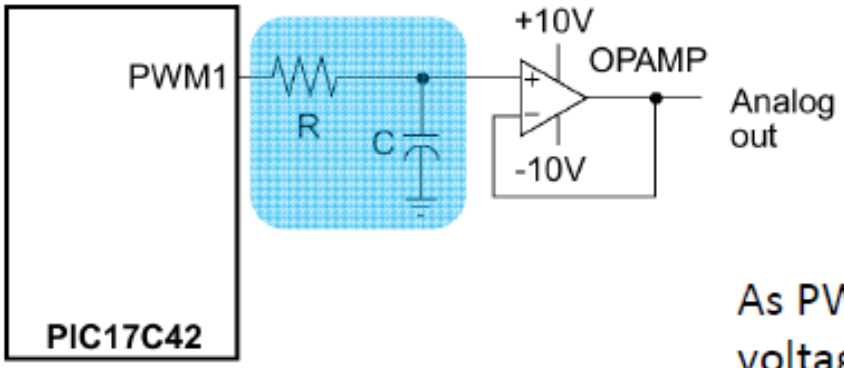


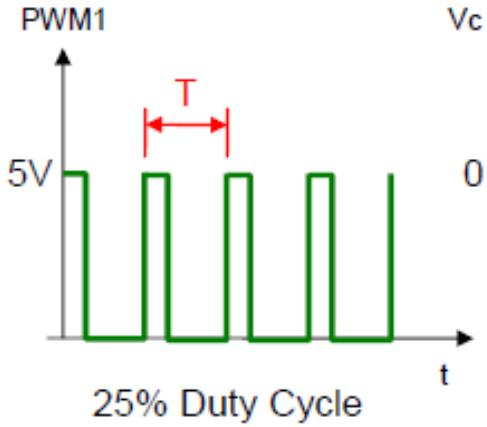


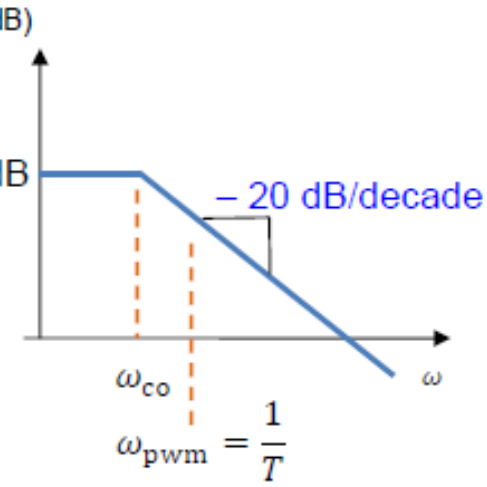
As PWM frequency increases further, capacitor voltage slowly increases, since capacitor cannot fully discharge before next cycle begins



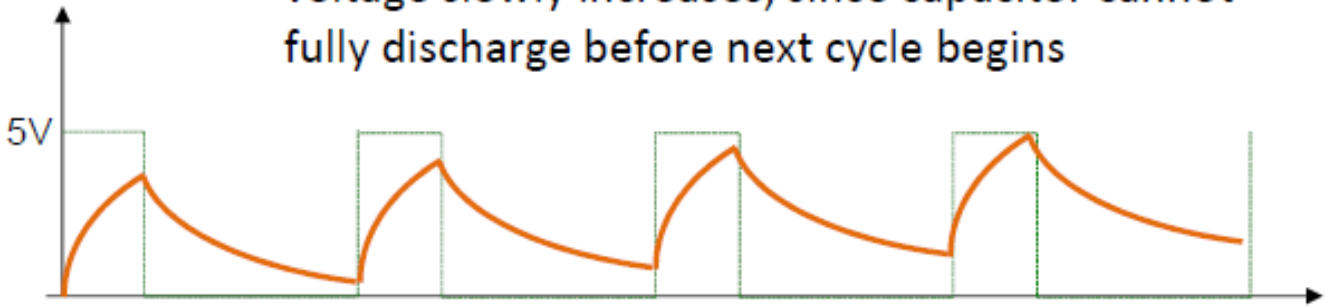
DIGITAL-TO-ANALOG (D/A) CONVERSION



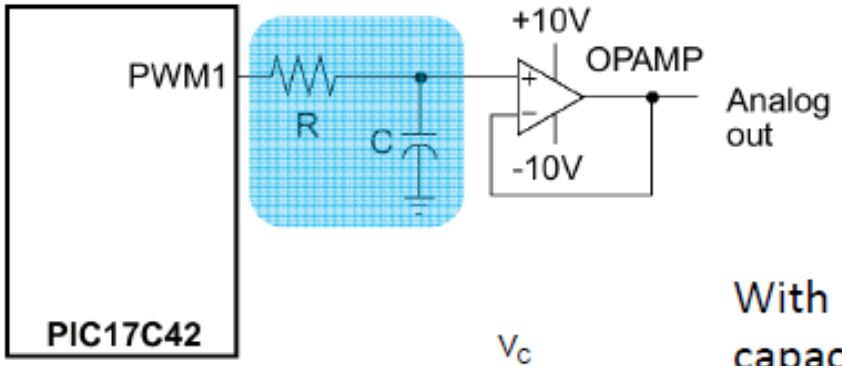


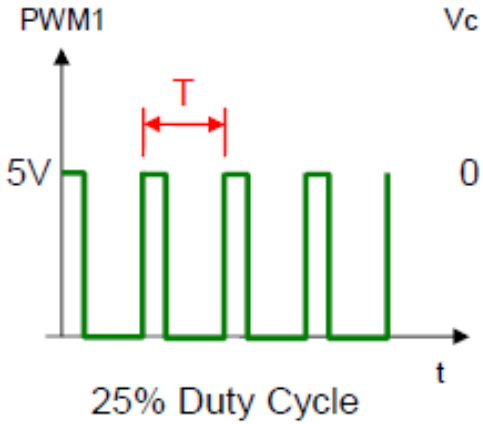


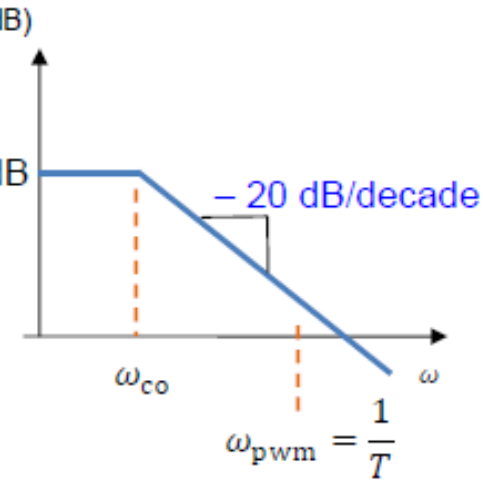
As PWM frequency increases further, capacitor voltage slowly increases, since capacitor cannot fully discharge before next cycle begins



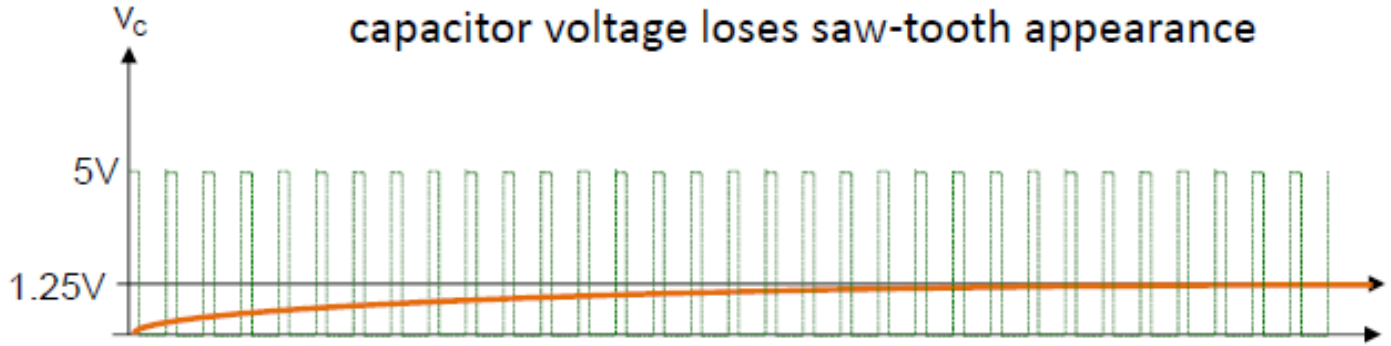
DIGITAL-TO-ANALOG (D/A) CONVERSION





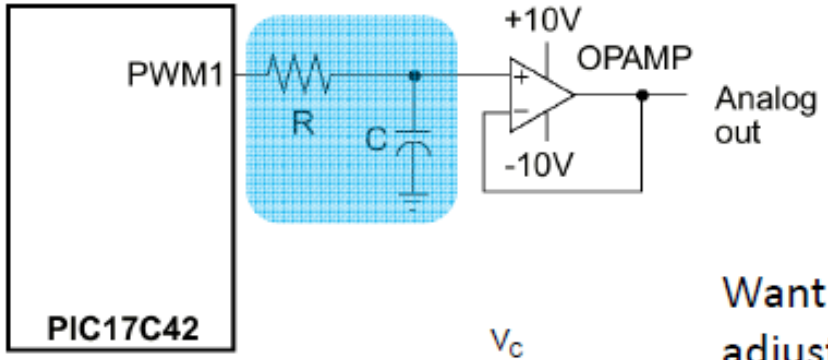


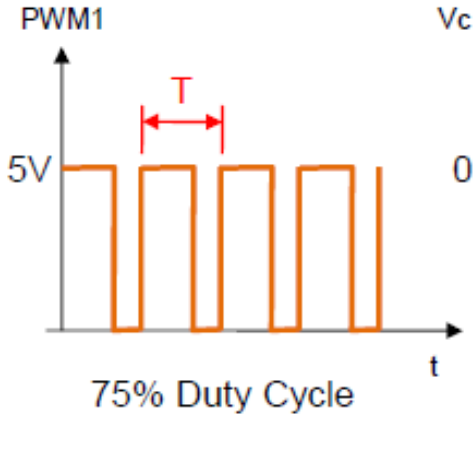
With PWM frequency well above cutoff frequency, capacitor voltage loses saw-tooth appearance

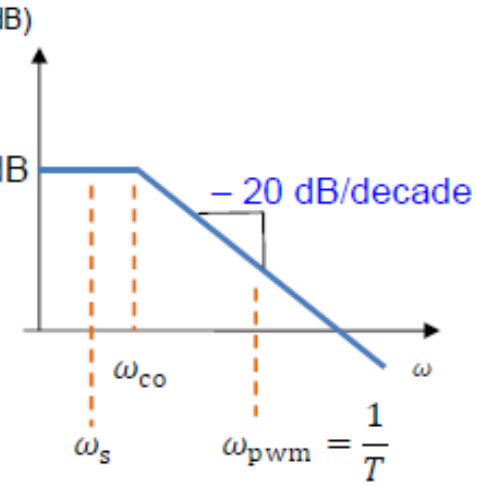


$V_{C,\text{ steady state}} = V_{\text{pwm}} \times \text{Duty Cycle \%}$

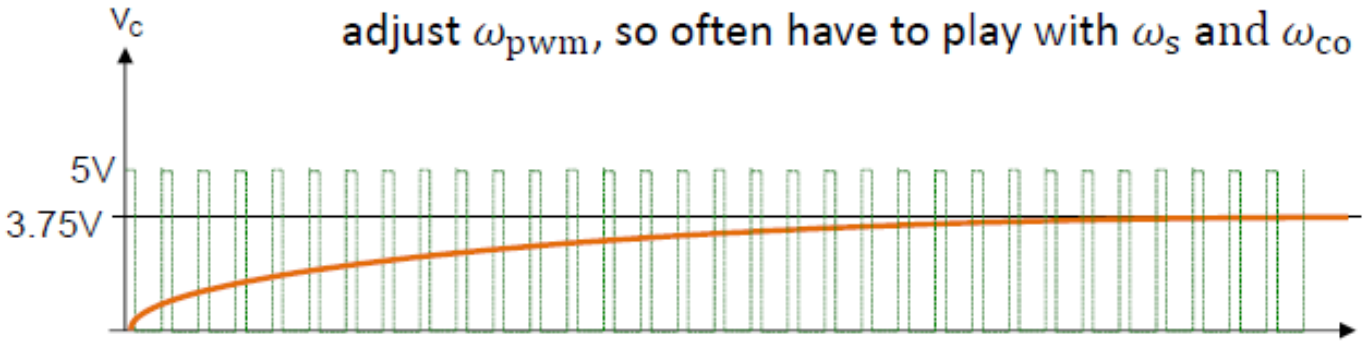
DIGITAL-TO-ANALOG (D/A) CONVERSION





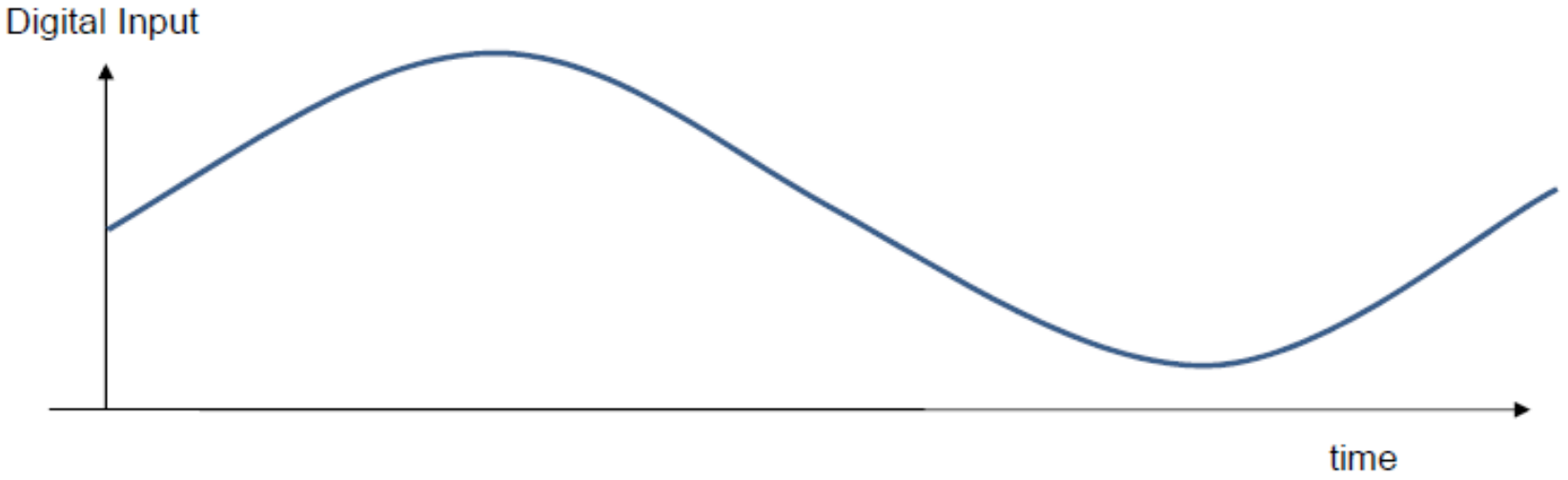


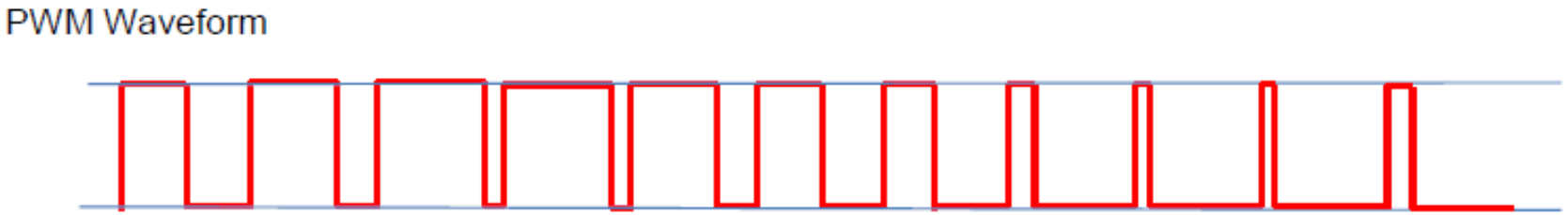
Want signal frequency ω_s less than ω_{co}. Can't always adjust ω_{pwm}, so often have to play with ω_s and ω_{co}



$V_{c,\text{ steady state}} = V_{pwm} \times$ Duty Cycle %

DIGITAL-TO-ANALOG (D/A) CONVERSION



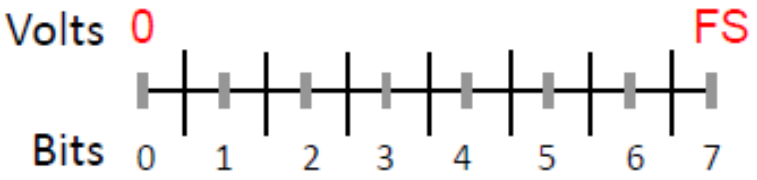


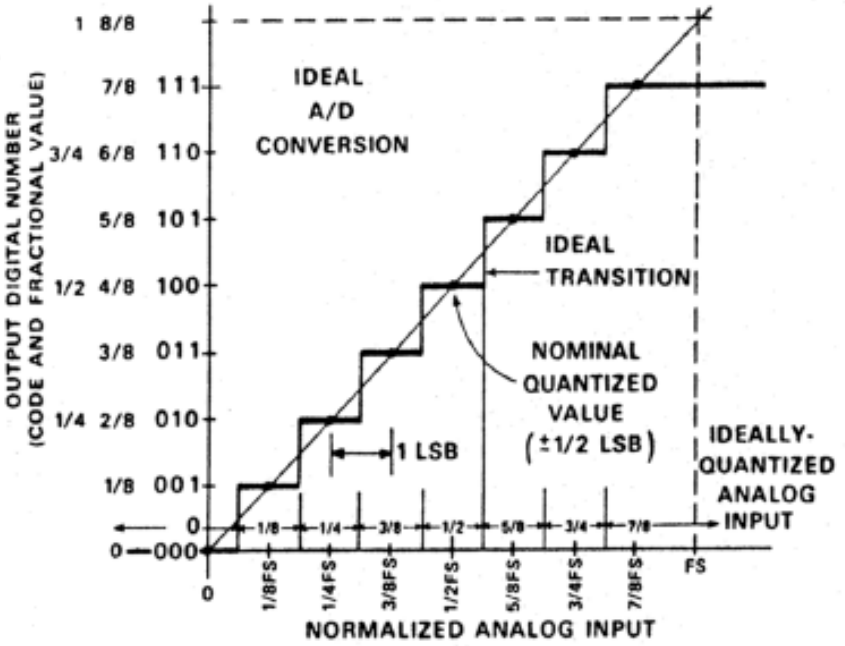
ANALOG TO DIGITAL CONVERTER (ADC)

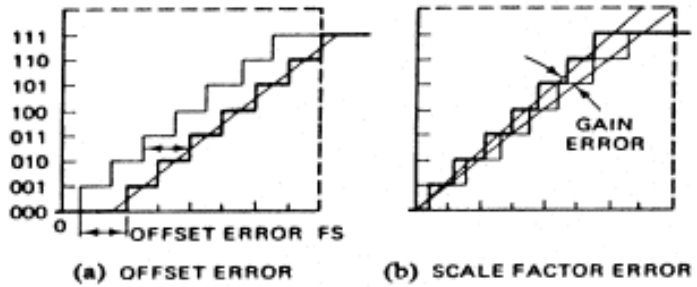
Converts analog inputs to digital values

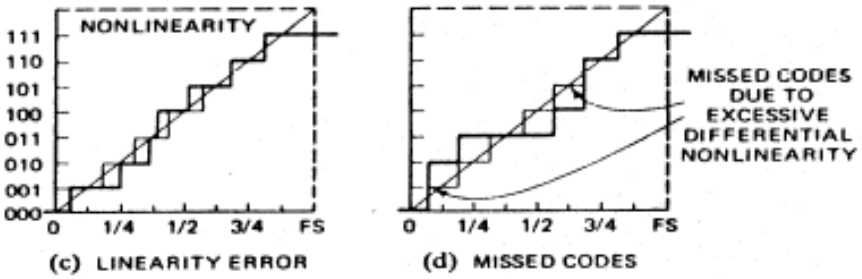
ANALOG-TO-DIGITAL (A/D) CONVERSION

Ideal AD Conversion:









SAMPLING AND ALIASING

The process of converting from analog to digital cannot be instantaneous. Thus, inputs need to be stabilized while conversion is performed.

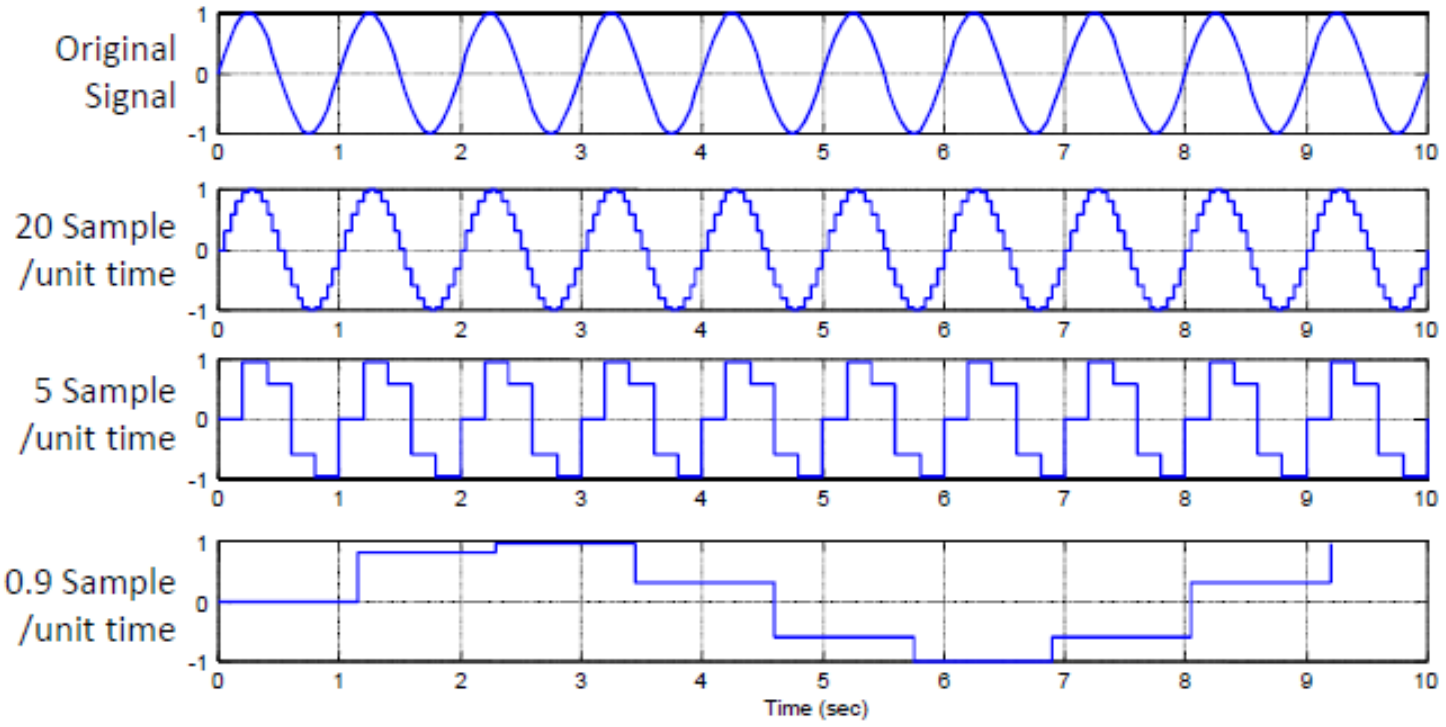
When signal (data) is sampled, temporal information is lost.

Sampling theorem (Shannon, Nyquist) gives limits

- Sampling rate must be faster than twice the highest frequency present (even if the highest frequency is noise) to preserve knowledge of the original signal's frequency content.

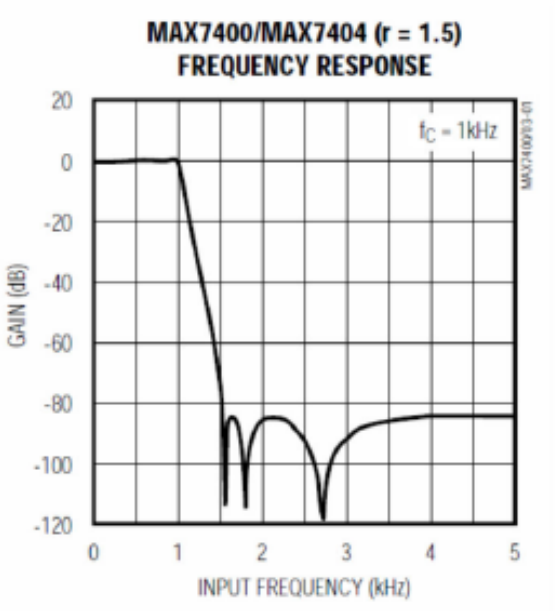
Slower sampling preserves amplitude information, but suggests incorrect frequency content; this is called *aliasing*.

SAMPLING AND ALIASING



ANTI-ALIASING

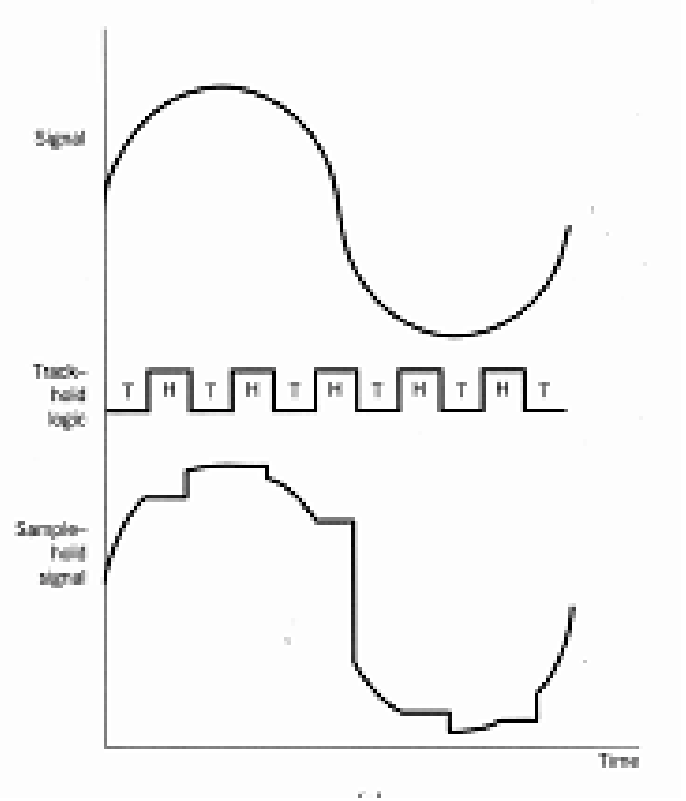
Located upstream of the sampling device, anti-aliasing (low-pass) filters attempt to limit signal bandwidths to no more than half the sampling frequency, thus limiting the extent to which aliasing occurs.



Maxim's 8th-order, low-pass, elliptic, switched-capacitor filters operate from a single +3V or +5V supply

Sample and hold

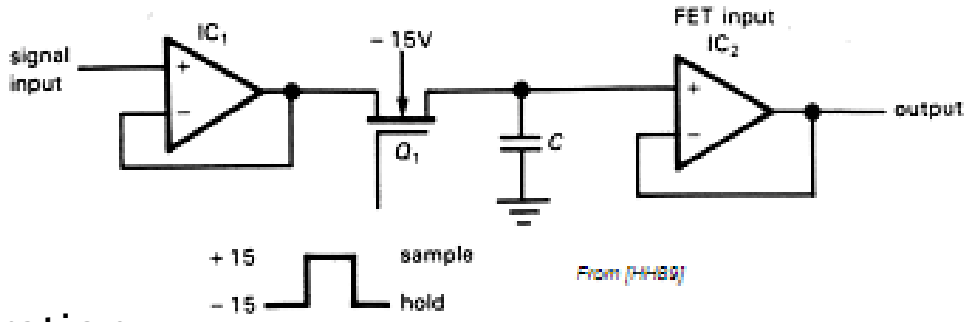
- **A Sample and hold (S/H) circuit has two basic operating modes**
  - **Sample mode**: The output follows the input
  - **Hold mode**: The output is held constant until sample mode is resumed
- **The main application of S/H circuits is to hold the input signal to an ADC constant during conversion**
  - Why? Imagine trying to photograph a moving object!



From [FB00]

Basic S/H circuit

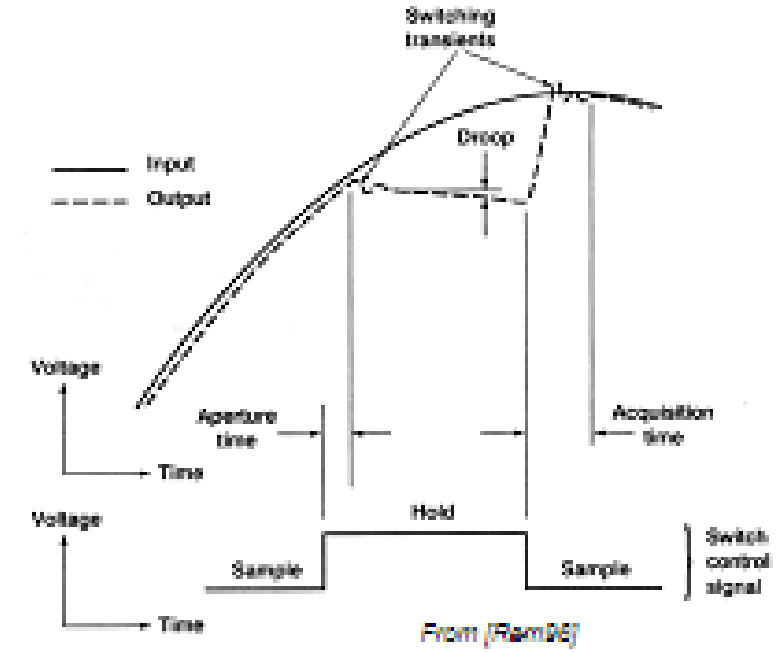
- Basic elements
  - Voltage followers
  - FET switch



- Operation
  - IC1 provides low Zout version of input signal
  - Q1 passes the signal during ‘sample’ and disconnects during ‘hold’
  - C preserves the value during ‘hold’
  - IC2 is a high Zin op-amp to minimize capacitor discharge during ‘hold’

S/H response characteristics

- Response parameters
  - Aperture time: time required for the switch to open (~50ns)
  - Droop: capacitor discharge (~1mV/ms)
  - Acquisition time: switch operation plus capacitor charging time

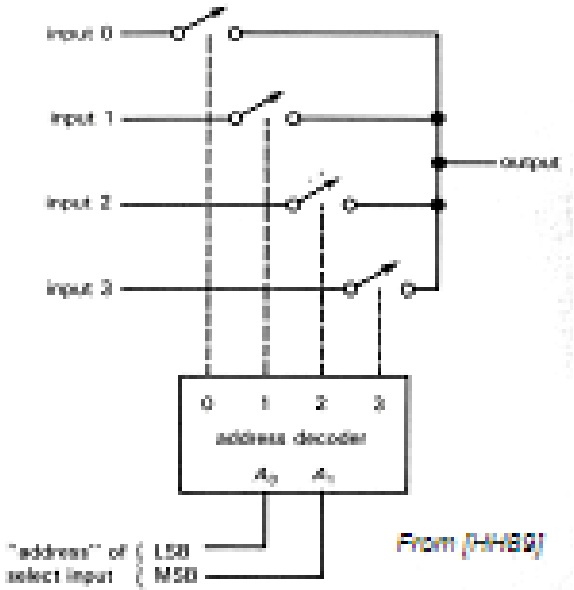


From [Ram96]

- Considerations for choosing C
  - C should be large enough to minimize 'droop' caused by leakage currents in Q1 and IC2
  - C should be small enough to track fast signals since it forms a low-pass filter with Q1's ON resistance!
    - In practice, the slew rate of the entire circuit is determined by IC1's output current and Q1's ON resistance

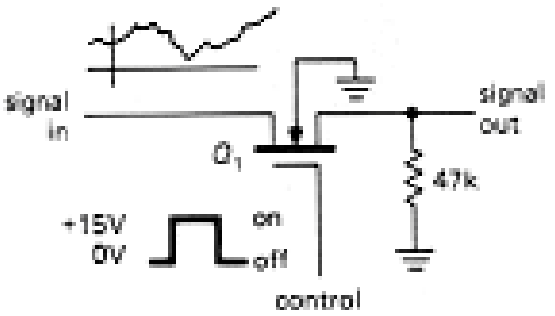
Multiplexers

- **A multiplexer is a circuit that allows you to select any of several inputs, as specified by digital control signals**
  - Since analog switches are bi-directional, this circuit could also be used as a de-multiplexer!
    - It could also be uses as a digital MUX since logic levels are just voltages



From [HH89]

- **FET analog switches**
  - N-channel enhancement-mode MOS-FET
    - When Gate is grounded or negative, the FET is non-conducting
      - Drain-source resistance in the order of 10,000MΩ
    - Bringing the Gate to +15V puts the drain-source channel into conduction
      - Drain-source resistance in the order of 100Ω



From [HH89]

Analog-to-digital converters

- Single slope or ramp
- Successive approximation
- Dual slope
- Parallel or ‘flash’

ADC

$$V_{in} \cong V_{REF}(B_1 * 2^{-1} + B_2 * 2^{-2} + \ldots B_N * 2^{-n}) \cong \frac{N_{10}}{2^n} V_{REF}$$

$$N_{10} = INT(\frac{V_{in}}{V_{REF}} 2^n)$$

Note: In Bipolar ADC Vref/2 sHould be subtraCted

Example:

Temperature is to be measured by a sensor wit an output of 0.02V/C. Determine required ADC (Reference, word size) to measure

0-100C wit0.1C resolution.

Solution

At max. temperature of 100C

.02*100=2V

So 2V reference is taken

A 0.1 C resolution results in .1*0.02=2mV

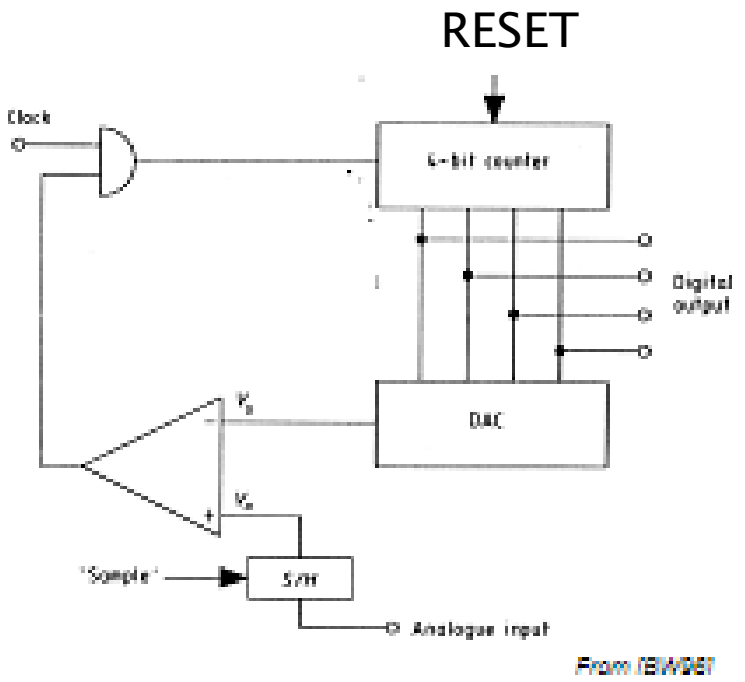
We need a word 2mV=2(2*exp(-y))

y≈9.996

Take n=10.

Single slope or ramp ADC

- Composed of three basic elements
  - A binary counter
  - A digital-to-analog converter
  - An analog comparator



From IEW2B7

Operation

• Counter is reset

•Analog input is sampled

• While VA>VB counter increments

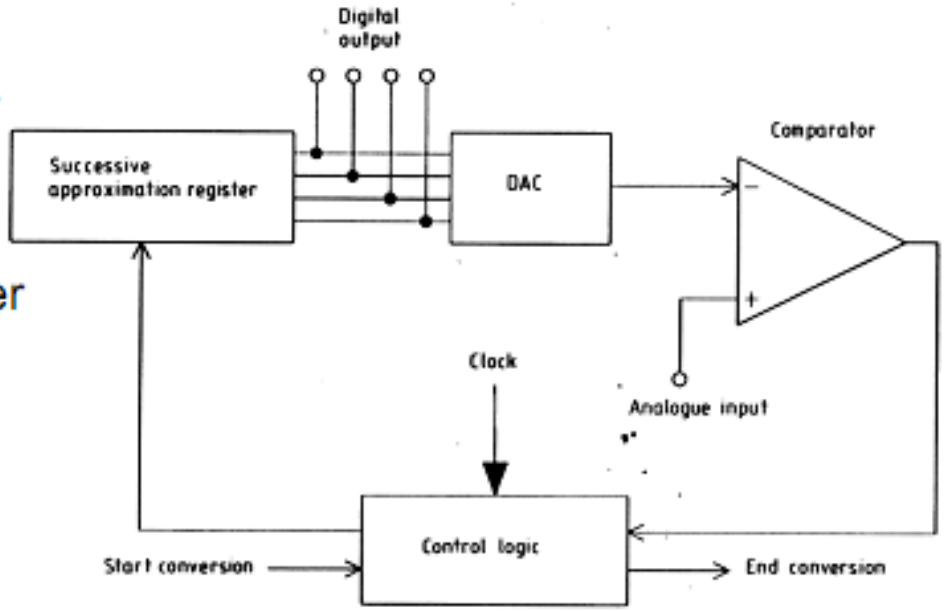
•When VA=VB counter stops and binary code is available at the output

• Characteristics

• Relatively slow since conversion time could be up to 2N, where N is the resolution of the ADC

Successive approximation ADC

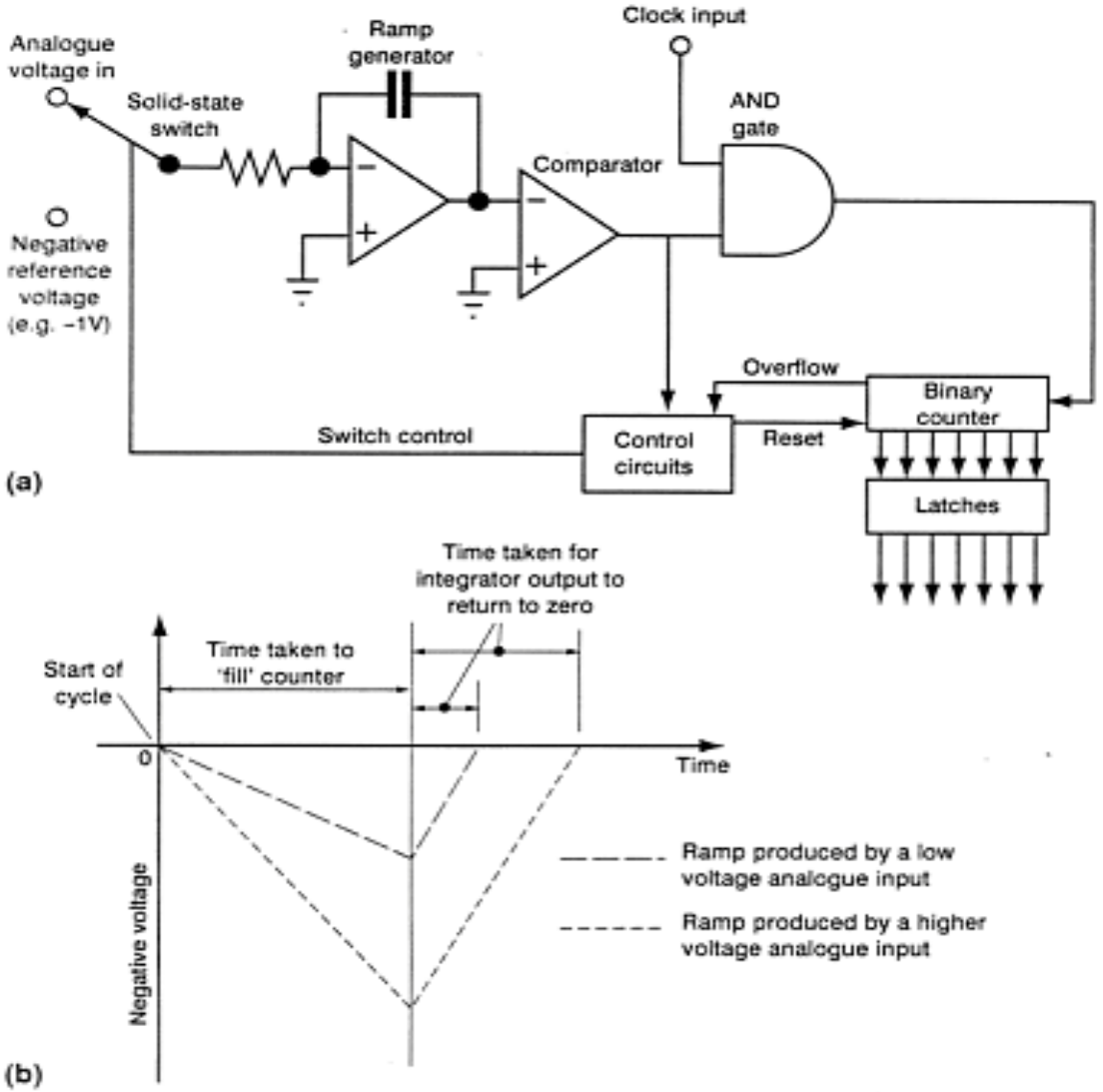
- **Basic elements**
  - A digital-to-analog converter
  - An analog comparator
  - A control logic module
  - A successive approx. register



From [BW96]

- **Operation is based on a binary search**
  - Initially, the register provides an output corresponding to half the range (1000…0)
    - If the analog input is greater, then MSB=1, else MSB=0
  - The register performs the same operation from MSB to LSB
- **Characteristics**
  - Conversion requires only N steps, where N is the resolution of the ADC
    - Conversion times of μs are typical

Dual slope ADC

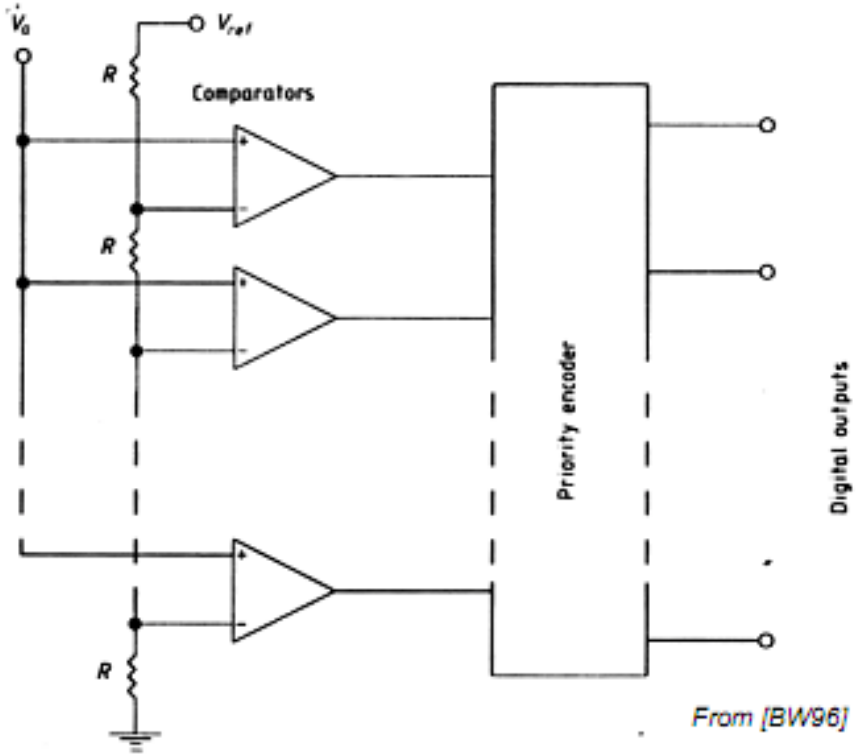


Dual slope ADC

- **Basic elements**
  - An integrator
  - A zero-crossing detector
  - A binary counter
  - Logic gates and switches
- **Operation**
  - Counter is reset and switch is connected to the analog input
    - The integrator generates a negative ramp whose slope is proportional to the analog input
    - The comparator goes HIGH, enabling clock pulses into the counter
  - When counter overflows, it resets to zero and the control circuit switches the switch to a reference negative voltage
    - This causes the integrator to generate a positive slope ramp
    - When this ramp reaches zero, the comparator goes low and stops the counter, whose value represents the analog input
- **Characteristics**
  - Very high resolution, but also slower (30 conversions/sec)
    - Widely used in digital multi-meters
  - Insensitive to clock drift, RC drifts and high-frequency noise

Parallel or flash ADC

- **Basic elements**
  - A multiple voltage divider
  - A set of comparators
  - A priority encoder

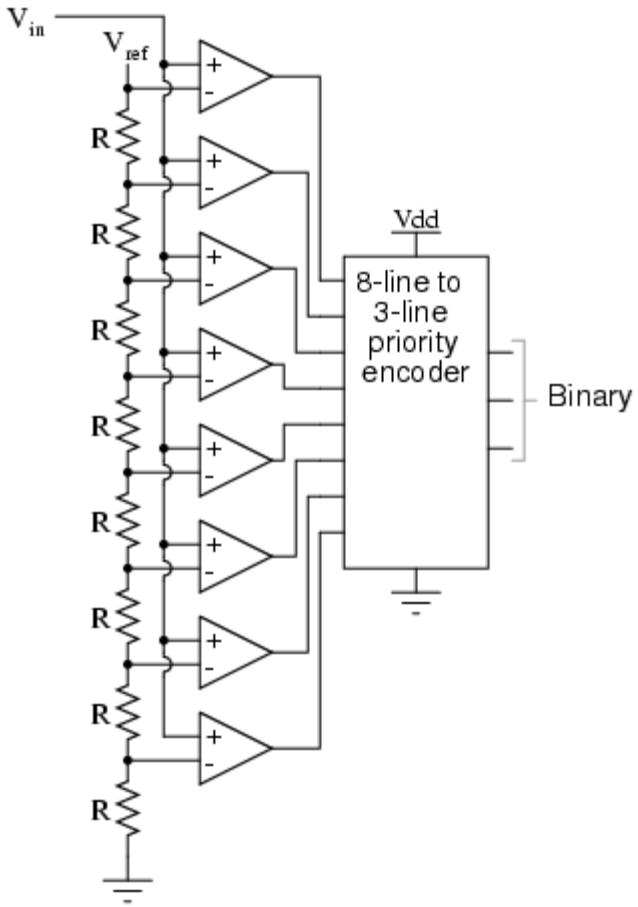


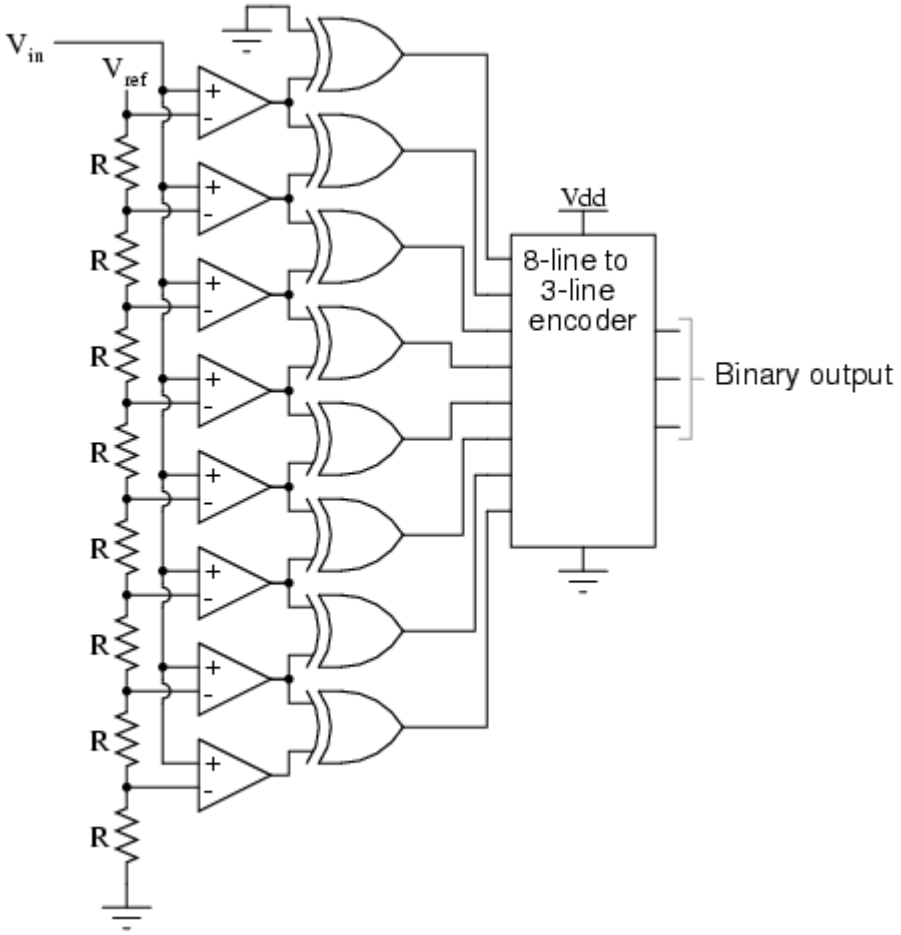
From [BW96]

- **Operation**
  - Analog input applied to all comparators
  - Priority encoder converts comparator pattern into binary
    - E.g.: A 3-bit ADC:
      - For comparator outputs of 0001111, priority encoder generates 100
      - For comparator outputs of 0111111, priority encoder generates 110

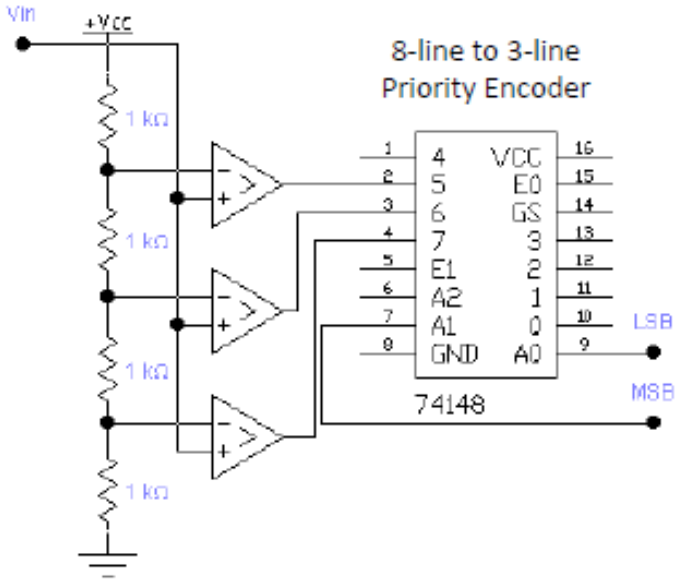
- **Characteristics**
  - Very fast (e.g., 8-bit ADCs capable of 20 million conversions/sec)
  - Very expensive for large N since the number of comparators is 2^N-1

Flash ADC


V_{in}
V_{ref}
R
R
R
R
R
R
R
Vdd
8-line to
3-line
priority
encoder
Binary

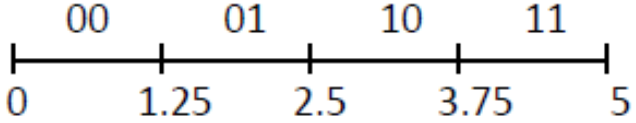

V_{in}
V_{ref}
R
R
R
R
R
R
R
Vdd
8-line to
3-line
encoder
Binary output

ANALOG-TO-DIGITAL (A/D) CONVERSION



Flash ADC

- Example: *2 bit Flash ADC*
  - Code:



  - Needs: 3 comparators

    C1: V > 1.25
    C2: V > 2.5
    C3: V > 3.75

Use truth table to get output values:

C3	C2	C1	MSB	LSB
0	0	0	0	0
0	0	1	0	1
0	1	0	X	X
0	1	1	1	0
1	0	0	X	X
1	0	1	X	X
1	1	0	X	X
1	1	1	1	1

ANALOG-TO-DIGITAL (A/D) CONVERSION

Flash ADC

- Need to provide sample-and-hold at the input -- if input is changing during conversion, erroneous values will be produced.
- Output valid for only a short time after input is held.
- Requires $2^N - 1$ comparators.
- Adjacent comparators must have monotonic range change.

ANALOG-TO-DIGITAL (A/D) CONVERSION

Successive Approximation ADC:

- Workhorse method.
- Used for wide variety of applications
- Typical conversion time: 1 – 100 µsec. Slower than flash ADC.
- Easily extensible to higher precision.
- Precision is limited by the quality of the components.

SUCCESSIVE APPROXIMATION ADC

Basic idea: Check bits starting from the high order bit (MSB).

- Algorithm:

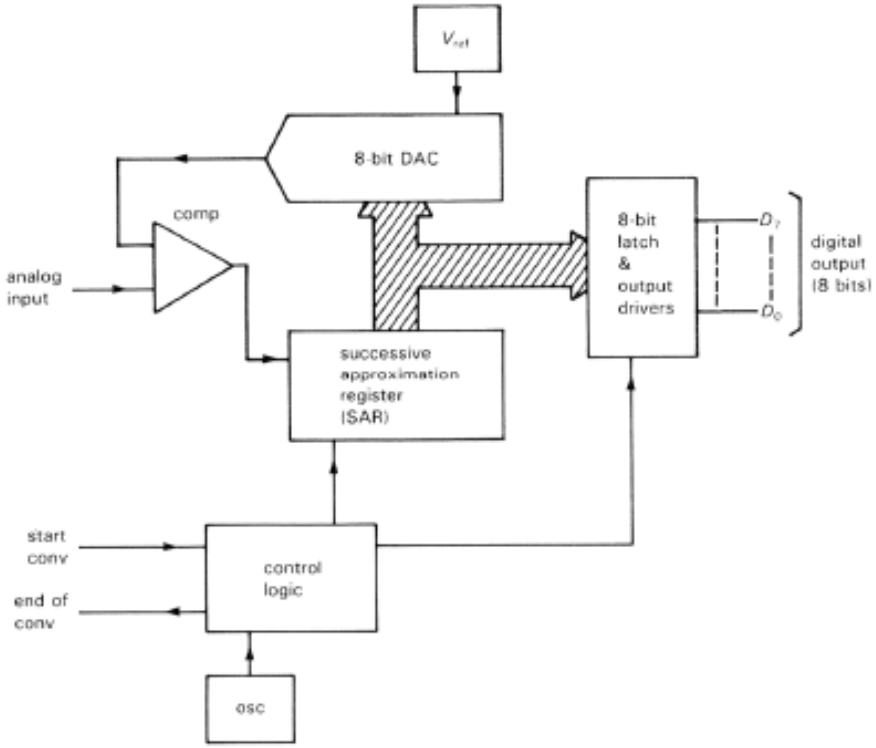
```
START CONVERSION
   SET Result to 0
   FOR i = N-1 TO 0
      SET i-th bit of Result to 1
      IF INPUT VOLTAGE < DtoA(Result)
         SET i-th bit of Result to 0
   END FOR-loop
   OUTPUT Result
END CONVERSION
```

This is a form of *interval halving*.

SUCCESSIVE APPROXIMATION ADC

General structure:

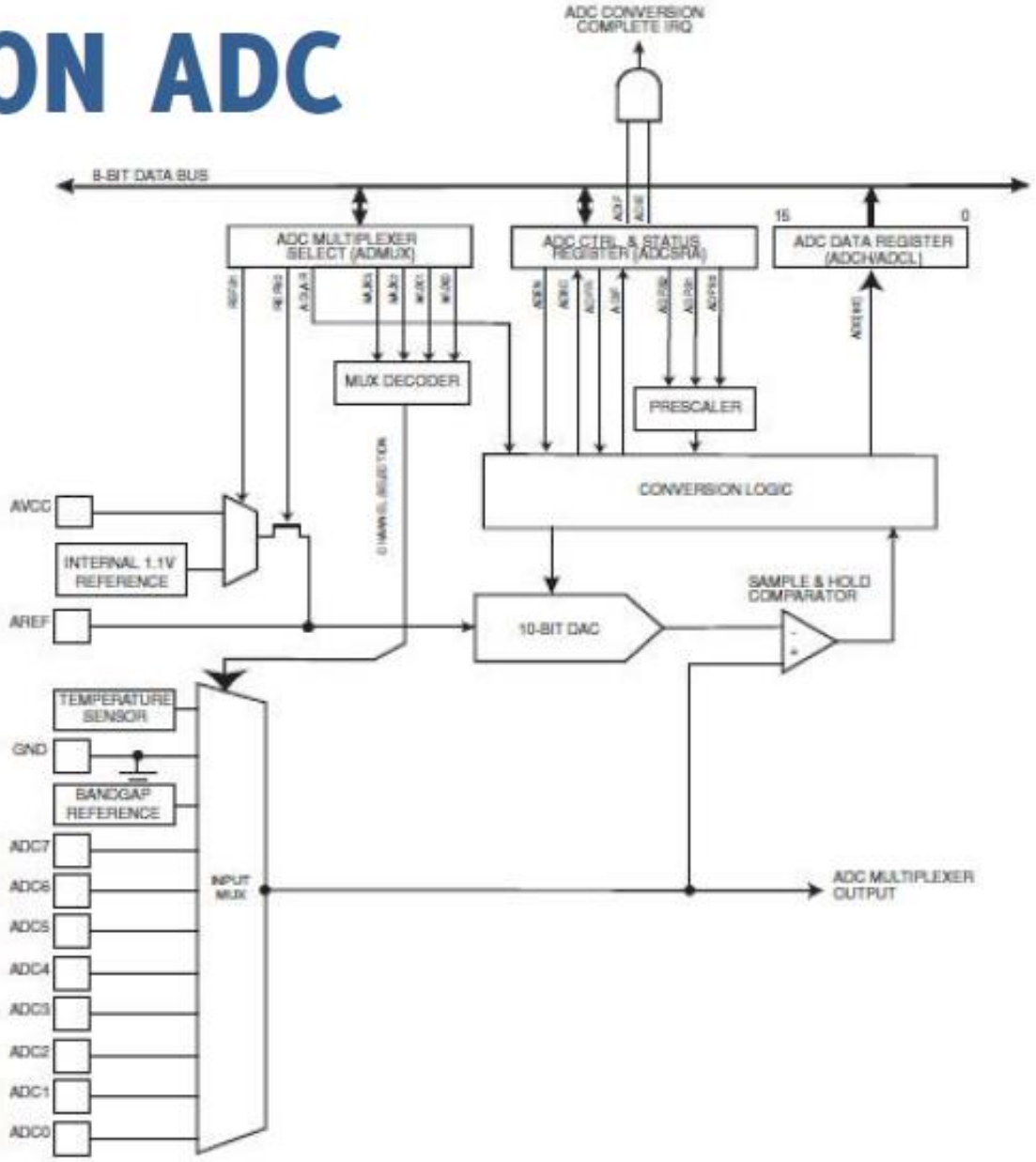
- Successive-approximation converters are quite expensive.
- Usually used with a multiplexer -- many channels feed to a single converter.
- Effective conversion speed for multiplexed ADC depends on number of channels used.
- Sample-and-hold normally precedes the converter.



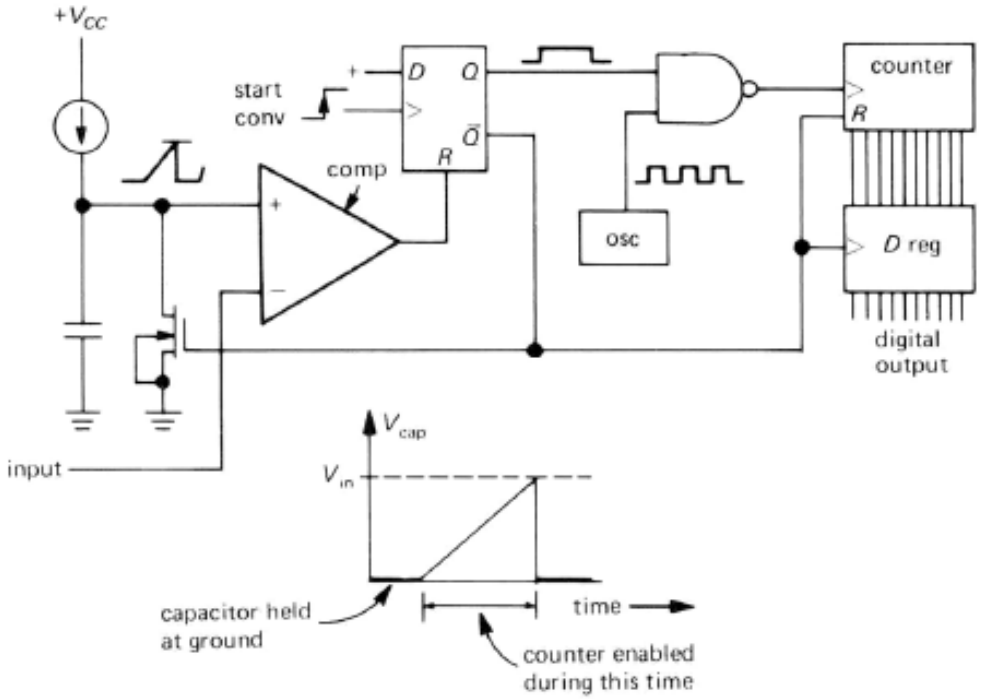
SUCCESSIVE APPROXIMATION ADC

Arduino ADC

6-channel 10-bit ADC



INTEGRATING CONVERTERS



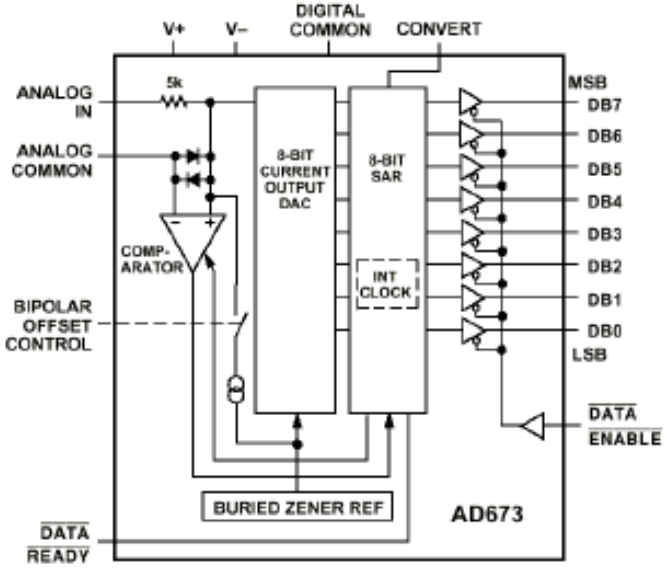
Slowest of the commonly used converters. Typical conversion time is many milliseconds.

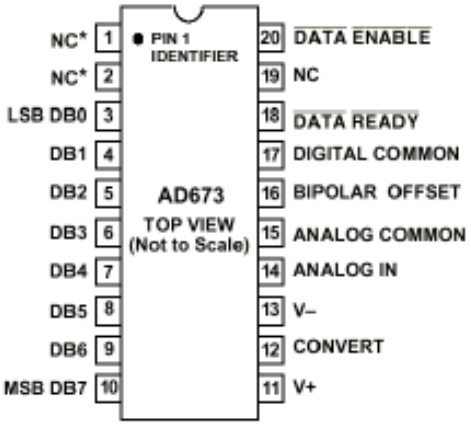
Can be made very accurate and precise – used in DVMs (several conversion per second).

Uses timing to determine digital value of unknown (input) voltage.

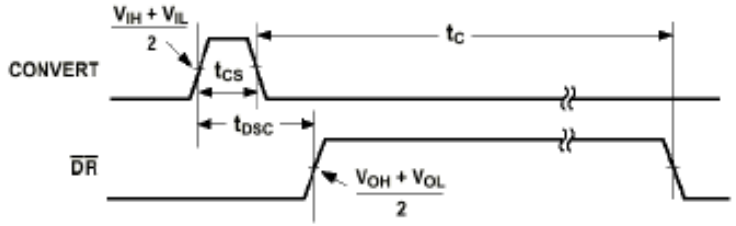
INTERFACING WITH AN ADC

- Successive approximation ADC – use AD673 as example

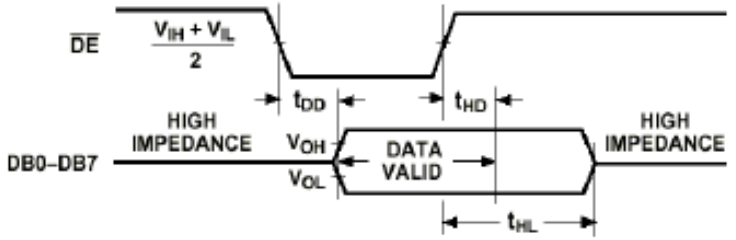


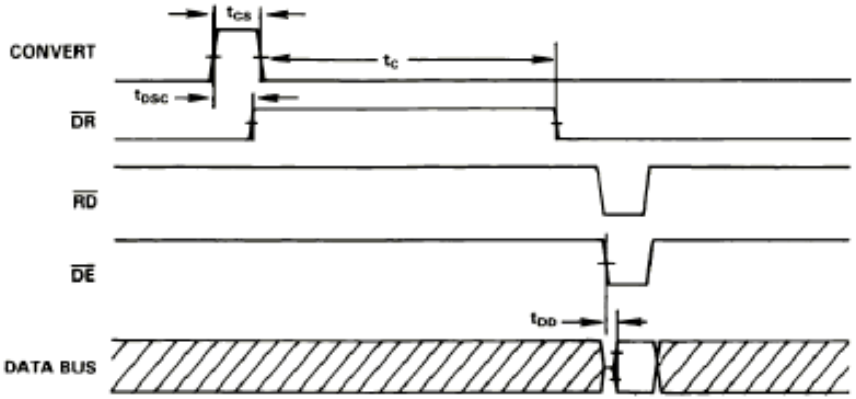


Convert timing

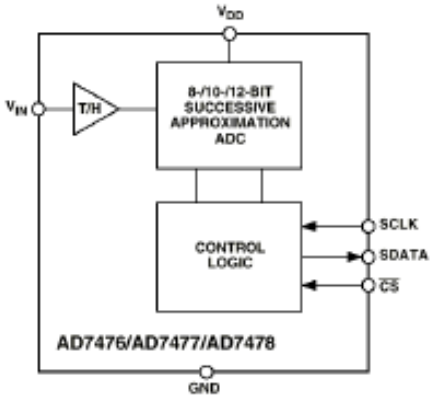


Reading timing





ANALOG-TO-DIGITAL (A/D) CONVERSION



ADC with Serial Output

- Reduce pinouts and package size
- Can be easily interfaced with a microcontroller or a microprocessor with built in serial interface (SCLK, SDATA, T/R)
- Sample rate limited by the maximum SCLK rate.
- Use AD7476 as example – 1 MSPS 6 pin ADC