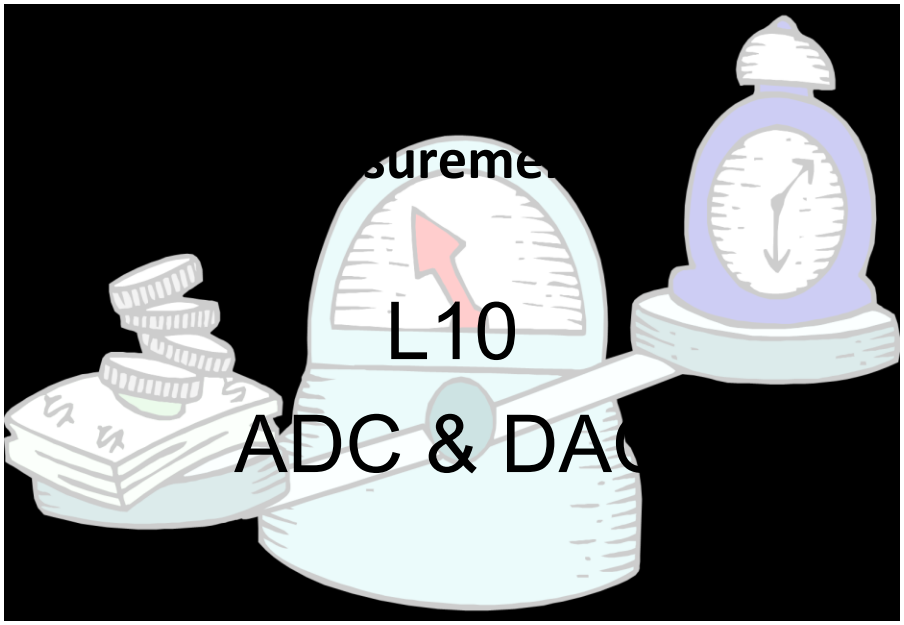


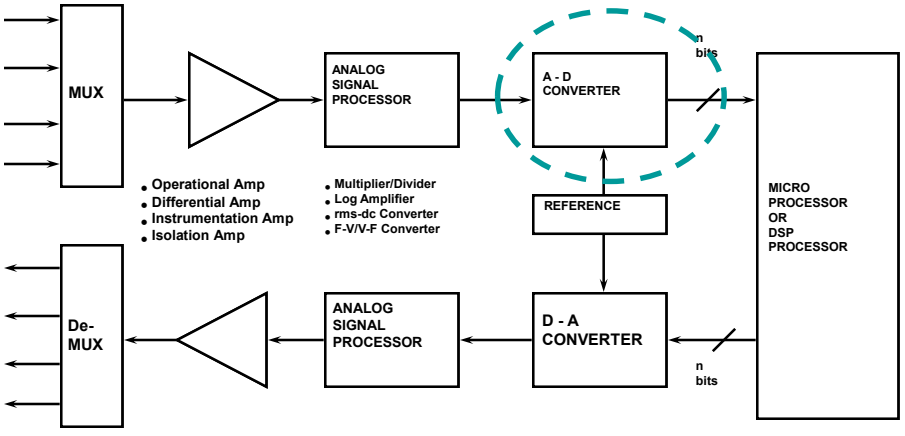
Outline

- Digital Signal Processing:
- Sampling;
- Sample and Hold;
- Analog to Digital Conversion;
- Digital to Analog Conversion.

Introduction

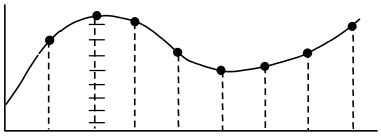
- Most Instrumentation systems include a digital processing device (PC, Microprocessor, DSP, Microcontroller….)
- These devices require information in digital format (binary, grey code, signed 2's complement , BCD…..etc)
- This requires some form of digitizer, or *analog-to-digital converter (ADC)*
- An ADC converts real-world signals (usually voltages) into digital numbers so that a computer or digital processor can (1) acquire signals automatically, (2) store and retrieve information about the signals, (3) process and analyze the information, and (4) display measurement results.

The Measurement & Control Loop



- The two main functions of an ADC are ***sampling*** and ***quantization***
- These two processes convert analog signals into digital numbers having discrete amplitudes, at discrete times.
- To represent changing signals at every instant in time or at every possible voltage would take an infinite amount of storage.

ADC SAMPLED AND QUANTIZED WAVEFORM



Sampling & Quantization

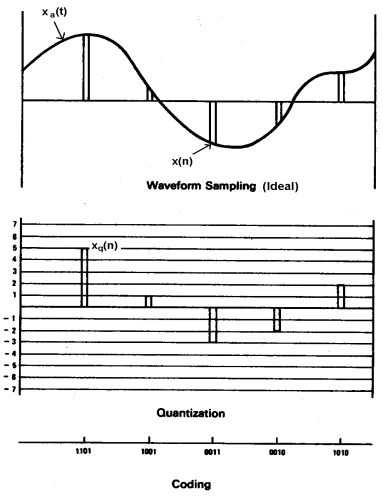
- For every system there is an appropriate ***sampling rate*** and ***degree of quantization (resolution)*** so that the system retains as much information as it needs about the input signals
- Ultimately, the purpose of sampling and quantization is to reduce as much as possible the amount of information about a signal that a system must store in order to reconstruct or analyze it meaningfully.

What is an Analog-Digital Converter?

- A devise that produces a Digital Output Corresponding to the Value of the Signal Applied to Its Input Relative to a Reference Voltage
- Finite Number of Discrete Values : 2^N Resulting in Quantization Uncertainty (N defines the resolution)
- Sampling and Quantization Impose Fundamental yet Predictable Limitations

Sampling

- **Sampling is the process of picking one value of a signal to represent the signal for some interval of time.**
- Sampling can be done at regular (constant) or variable rate.



- Sampling is done by a circuit called a *sample-and-hold (S/H)* , which, at a sampling instant, transfers the input signal to the output and holds it steady, even though the input signal may still be changing.
- Most modern ADC chip has a built-in S/H or T/H, and virtually all data acquisition systems include them.
- Of course, sampling necessarily throws away some information, so the art of sampling is in choosing the right sample rate so that enough of the input signal is preserved.

Minimum sampling rate (Nyquest Rate)

- **Nyquest criterion states that the sampling rate must be at least twice the highest frequency of the signal of interest:**

$$f_{sampling} > 2f_{signal}$$

- **Nyquest criterion guarantees the preservation of the frequency content of the signal, but not the time dependency**
- **In order to be able to reconstruct the signal in time domain, as a rule of thump , we use : fsampling > 10*fsignal**
- Aliasing occurs when the sampling is done at a rate less than Nyquest rate

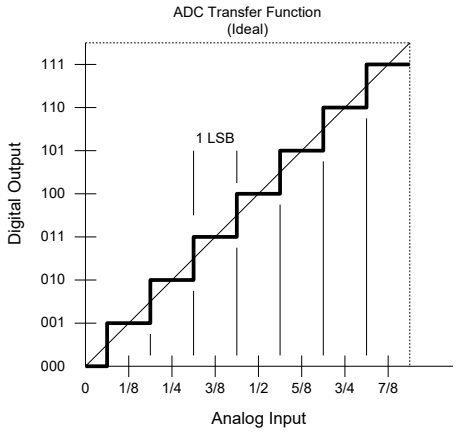
Quantization

- **Quantization is the second step in ADC in which the samples are converted into equivalent digital value**
- What sampling accomplishes in the time domain, quantization does in the amplitude domain
- Conversion takes finite time , and next sample cannot be taken before the last sample conversion is completed.
- The smaller the intervals between the samples, the higher the sampling rate (sampling frequency)

Quantization

- An ADC quantizes a sampled signal by picking one integer value from a predetermined, finite list of integer values to represent each analog sample.
- Each integer value in the list represents a fraction of the total analog input range.
- Normally, an ADC chooses the value closest to the actual sample from a list of uniformly spaced values.
- **This rule gives the *transfer function* of analog input to- digital output a uniform "staircase" characteristic.**

- Figure represents a three-bit quantizer, which maps a continuum of analog input values to only eight (2^3) possible output values.
- Each step in the staircase has (ideally) the same width along the *x* -axis, which we call *code width* and define as 1*LSB (least significant bit)*
- In this case 1 LSB is equal to 1/8 V. Each digital code corresponds to one of eight 1-LSB intervals making up the analog input range, which is 8 LSB (and also 1 V in this case).



ADC Specifications

- 1. Range: The input range of ADC is the span of voltage over which the ADC can make conversion
- The end points at the bottom and the top of the range are called –*full-scale* and + *full-scale* , respectively.
- When –full-scale = 0 V the range is called *unipolar*
- when –full-scale is a negative voltage of the same magnitude as +full-scale the range is said to be *bipolar*
- **When the input voltage exceeds the input range, the conversion data are certain to be wrong**, and most ADCs report the code at the end point of the range closest to the input voltage.
- This condition is called an *over-range*

- $Range = V_{FSR} = +Full\ Scale\ -(-Full\ Scale)$
- For Example : Unipolar 0 to 15 V
  Range=15-0=15V
- Bipolar : -5V to +5V
  Range =5-(-5)=10 V

- **2)The resolution** of an ADC is the smallest change in voltage the ADC can detect, which is inherently **1 LSB**.
- It is customary to refer to the resolution of an ADC by the number of binary bits or decimal digits it produces; for example, “12 bits” means that the ADC can resolve one part in 2 ^12 (= 4096).
- In the case of a digital voltmeter that reads decimal digits, we refer to the number of digits that it resolves.
- A “6-digit” voltmeter on a 1 V scale measures from –0.999999 V to +0.999999 V in 0.000001 V steps; it resolves one part in 2 000 000.
- It is also common to refer to a voltmeter that measures from –1.999999 to +1.999999 as a “6 ½digit” voltmeter.

- Resolution of ADC is the smallest detectable change in voltage ; however some refer to number of bits as resolution
- $\Delta = LSB = Q = \frac{V_{FSR}}{2^n}$
- Example : 8 bit ADC with a range 0-2.5 V input

$\Delta = LSB = Q = \frac{2.5}{2^8} = \frac{2500mV}{256}$=9.765 mV

Or as % $\Delta = \frac{9.765}{2500} * 100\% = 0.3906\%$

Or 0.003906 ➔ 3906 ppm (parts per million)

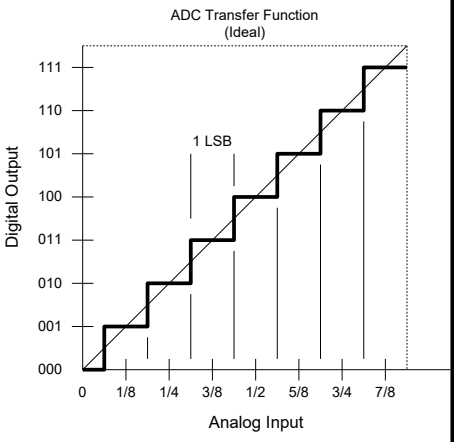
ADC Resolution vs. Quantization Parameters

Resolution, Bits (n)	2^n	LSB, mV (2.5V FS)	% Full Scale	ppm Full Scale	dB Full Scale
8	256	9.77	0.391	3906	-48.0
10	1024	2.44	0.098	977	-60.0
12	4096	0.610	0.024	244	-72.0
14	16,384	0.153	0.006	61	-84.0
16	65,536	0.038	0.0015	15	-96.0
18	262,164	0.0095	0.00038	3.8	-108.0

- Higher n gives better resolution

DC Specifications (Ideal-midtread linear type)

- 3 bit, 0-1 V range
- There are 2^3 codes.
- For this 3-bit ADC, 1 LSB = ($1V/2^3$ = 1/8th)
- Each "step" is centered on an eighth of full scale
- $ADC_{CODE} = Round\left(\frac{2^n}{V_{FSR}}.Vin\right)$



Example

- A 5 bit ADC with a 0-5V input range have a 3.127V input signal at its input, what is the output **binary code** at the output?
- Solution:
- $ADC_{CODE} = Round\left(\frac{2^5}{5}.3.127\right)$=$(20)_{10}$
- $(20)_{10}$ =$(10100)_2$

Example

- Given a sensor with 0.02 V / degree C sensitivity, choose a suitable size ADC with V_{FSR}=2V such that to be able to measure 0-100 deg C with :
- a) 1 deg C resolution
- b) 0.1 $\deg C$ resolution
- **Solution:**
- 0 deg C => 0.0 V
- 100 deg C => 100x 0.02=2V
- We have an ADC with V_{FSR}= 2 V range
- For 1 deg C resolution , ADC resolution Δ = 0.02 V
- $\Delta = 0.02V = \frac{V_{FSR}}{2^n} = \frac{2}{2^n} \Rightarrow 2^n$=100==> n =$\log_2$(100)=6.644
- N must be an integer, so we choose n=7

Example

sensor output matched to ADC input

- For 0.1 deg C resolution , ADC resolution Δ = 0.002 V
- $\Delta = 0.002V = \frac{V_{FSR}}{2^n} = \frac{2}{2^n} \Rightarrow 2^n$=1000==> n =$\log_2$(1000)=9.97
- N must be an integer, so we choose n=10

Example: ADC range 5 V

(not matched by sensor output "2V")

- **Solution:**
- 0 deg C => 0 V ; 1 deg C ➔ 0.02V
- 100 deg C => 100x 0.02=2V
- Suppose we have an ADC with V_{FSR}= 5 V range
- For 1 deg C resolution , ADC resolution Δ = 0.02 V
- $\Delta = 0.02V = \frac{V_{FSR}}{2^n} = \frac{5}{2^n} \Longrightarrow 2^n$=250==> n =$\log_2$(250)=7.96
- N must be an integer, so we choose n=8
- For 0.1 deg C resolution , ADC resolution Δ = 0.02 V
- $\Delta = 0.002V = \frac{V_{FSR}}{2^n} = \frac{5}{2^n} \Longrightarrow 2^n$=2500==> n =$\log_2$(2500)=11.3
- N must be an integer, so we choose n=12
-

Example: ADC range 5 V

(matched by sensor output through S/C block)

- **Solution:** **S/C**
- 1 deg C => 0.02V ===> **multiplied by 2.5 ==> 0.05 V**
- 100 deg C => 100x 0.02=2V **===> X 2.5 = 5V**
- Suppose we have an ADC with V_{FSR}= 5 V range
- For 1 deg C resolution , ADC resolution Δ = 0.02 V
- $\Delta = 0.05V = \frac{V_{FSR}}{2^n} = \frac{5}{2^n} \Longrightarrow 2^n$=100==> n =$\log_2$(250)=6.64
- N must be an integer, so we choose n=7
- For 0.1 deg C resolution , ADC resolution Δ = 0.05 V
- $\Delta = 0.005V = \frac{V_{FSR}}{2^n} = \frac{5}{2^n} \Longrightarrow 2^n$=1000==> n =$\log_2$(1000)=9.96
- N must be an integer, so we choose n=10

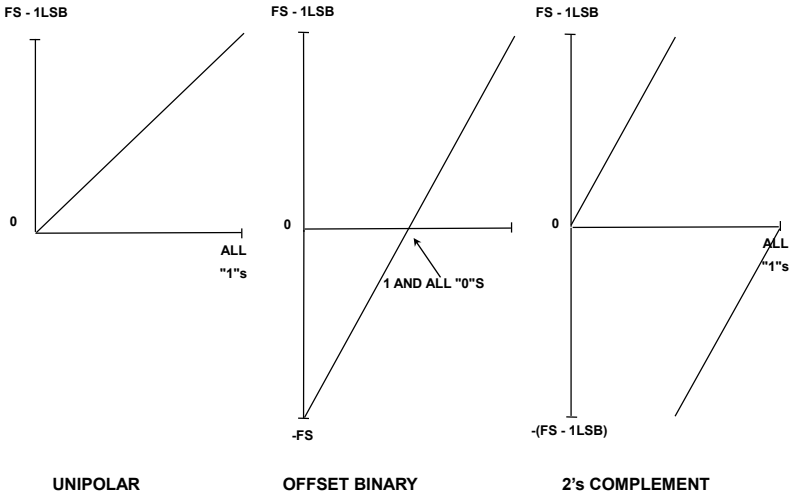
Conclusion about range matching

- When the ranges are not matched, higher number of bits “n” is required to achieve higher measurement resolution
- When we match the output range of the sensor to the input range of ADC, we can get better resolution of a given ADC with given number of bits

Coding Conventions

- There are several different formats for ADC output data:
  - Unipolar
  - Bipolar
- An ADC using *binary* coding produces all 0s (e.g., 000 for the three-bit converter) at –full-scale and all 1s (e.g., 111) at +full-scale.
- If the range is bipolar, so that –full-scale is a negative voltage, binary coding is sometimes called *offset binary* since the code 0 does not refer to 0 V.
- To make digital 0 correspond to 0 V, bipolar ADCs use *two’s complement* coding, which is identical to offset binary coding except that the *most significant bit (MSB*) is inverted, so that 100 … 00 corresponds to –full-scale, 000 … 00 corresponds to 0 V (*midscale*), and 011 … 11 corresponds to +full-scale.

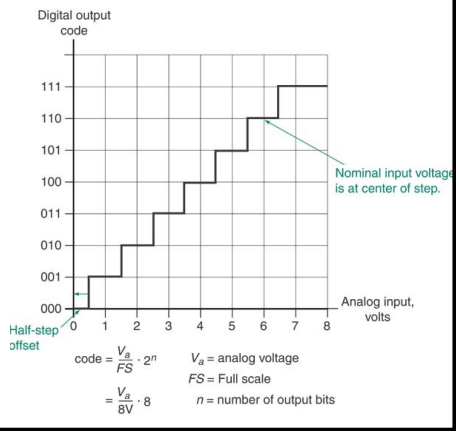
Unipolar and Bipolar Converter Codes



Unipolar ADC Code Equation

$$\text{code} = \frac{V_a}{V_{FS}} \times 2^n$$

- V_a = analog input voltage to be sampled.
- V_{FS} = Full scale range of input voltage.
- n = number of bits in the output code.

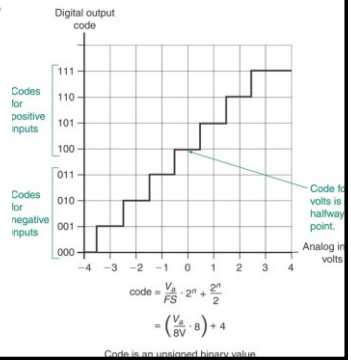


Bipolar ADC (Offset Binary Coding)

- Used to represent positive and negative input voltages.
- Output code an unsigned binary number.
- Numbers below 0 V are negative.
- Numbers above 0 V are positive.

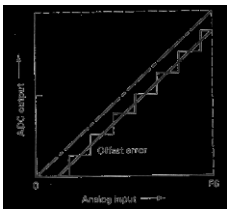
$$\text{code} = \left(\frac{V_a}{V_{FS}} \times 2^n \right) + \text{offset}$$

$$= \left(\frac{V_a}{V_{FS}} \times 2^n \right) + \frac{2^n}{2}$$

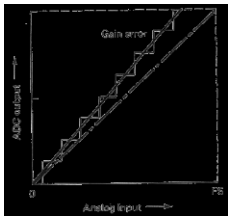


Summary of Other ADC Errors

- Offset Error

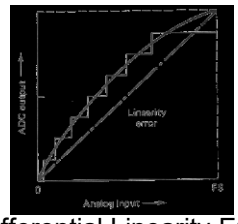


- Gain Error

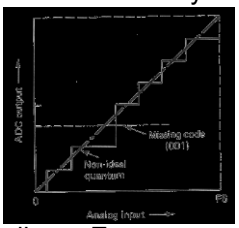


- Can be eliminated by initial adjustments

- Integral Linearity Error



- Differential Linearity Error

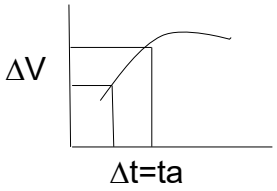


- Nonlinear Error
  - Hard to remove

Chap 0

Aperture Error

- ADC conversion takes time (τ_c) which might be fixed or variable depending of ADC type
- If analog input is changing during conversion(this can also be due to clock jitter), then the converted signal will be in error known as Aperture Error
- To avoid error in digitized output, this change must be small and less than ½ Δ



Actual Sampling time uncertainty due to changing signal or clock jitter

Consider A sine wave

- Vin=Vref Sin $\omega_0 t$
- $\left(\frac{\Delta V}{\Delta t}\right)_{max} = V_{REF}\omega_0 \Longrightarrow \Delta t = \frac{\Delta V}{V_{REF}\omega_0}$

$$let\ \Delta V = \frac{1}{2}LSB = \frac{V_{REF}}{2.2^n}$$

$$\Delta t = \frac{\frac{V_{REF}}{2.2^n}}{V_{REF}\omega_0} = \frac{1}{2.2^n\omega_0}$$

Make sure that τ_c is less than ta

Conversion Time (τ_c)

- The time the ADC takes to convert a a single analog input voltage value to corresponding digital value
- During conversion process the input must remain constant, or if it is allowed to change, this change should be restricted to a value : $\frac{dV_{in}}{dt} < \frac{V_{FSR}}{2^n \tau_c}$

- 10 bit ADC with τ_c=20 μs , what is the maximum allowable rate of change (frequency) of a sinusoidal input voltage to be converted using this ADC
- Solution: $\frac{dV_{in}}{dt} < \frac{V_{FSR}}{2^n \tau_c}$
- $A\omega_o < \frac{V_{FSR}}{2^n \tau_c}$, let A=V_{FSR}

$$\omega_o < \frac{1}{2^n \tau_c}$$

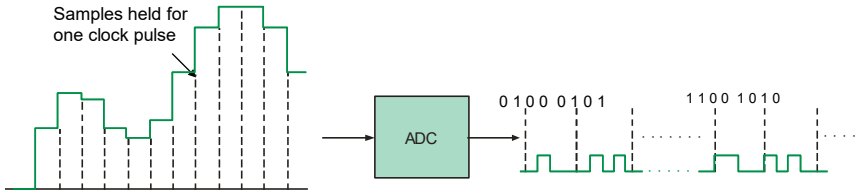
$f_o < \frac{1}{2\pi.2^{10}20\mu s} = 7.75\ Hz$

This is the maximum frequency of input to be used with this ADC>>>>> If higher frequency needed , increase n or reduce τ_c

Sample-and-Hold and ADC

Following the anti-aliasing filter, is the sample-and-hold circuit and the analog-to-digital converter.

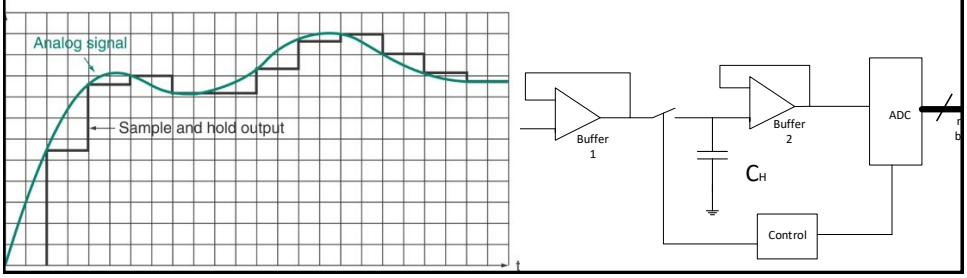
At this point, the original analog signal has been converted to a digital signal.



Many ICs can perform both functions on a single chip and include two or more channels. For audio applications, the AD1871 is an example of a stereo audio ADC.

Sample and Hold Circuit

- Sample and Hold (S& H) reduces Δt
- S&H is placed at the input of ADC, it holds the input signal at a constant level during conversion
- S&H is used to avoid errors if variations of the measurand were allowed to pass to the ADC
- Reduce uncertainty error in the converted output when input changes are fast compared to the conversion time
- Sample mode: output follows input
- Hold: Output is held constant until sample mode is resumed

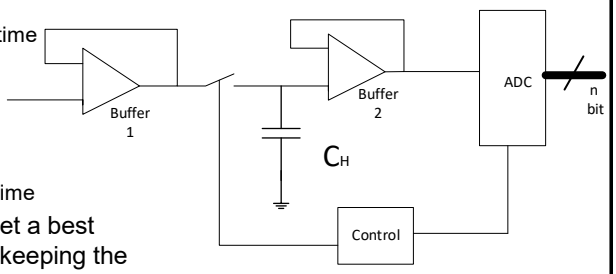


- This is low pass filter with fc as high as possible in order not to disturb the signal

$$f_c = \frac{1}{2\pi(\sum R)C_H}$$

- Care in selecting hold capacitor C_H
  - Low Value
    - Increases BW
    - Reduces acquisition time
    - Increase Droop
  - High Value
    - Reduces BW
    - Minimize Droop
    - Increase acquisition time
  - Choose capacitor to get a best acquisition time while keeping the droop per conversion below 1 LSB

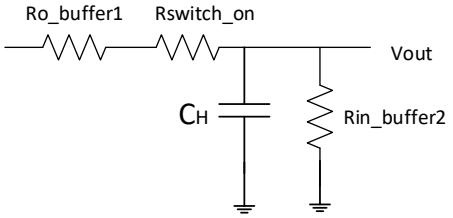
- In Multi-channel system S&H used
  - To hold a sample from one channel while multiplexer proceed to sample next one
  - Simultaneous sampling of two signal



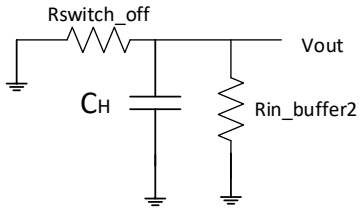
- Care in selecting hold capacitor C_H

  Choose capacitor to get a best acquisition time while keeping the droop per conversion below 1 LSB

Sample/Track Mode
For High fc , C_H must be low



Hold Mode
C_H must be low to minimize discharge of Capacitor (Droop)

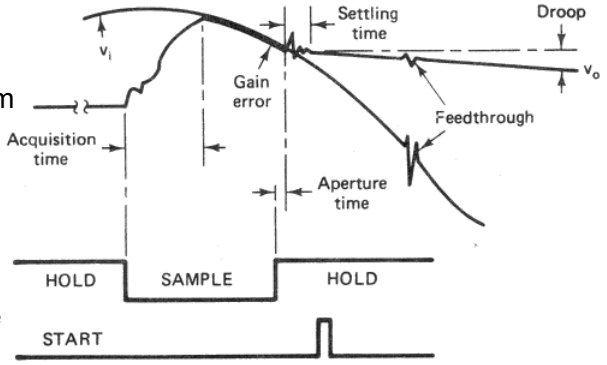


Sample and Hold Circuits

τ_{acq} – time required for the S&H circuit to acquire signal when changing from the hold mode to sample mode

τ_{ap} – aperture time, time required for switch from sample to hold state (time when output stops following the input)

or time between when the hold command is received to actual transition to actual hold mode



Converter Throughput Rate/ Frequency

- It is defined as the number of times the input signal can be sampled maintaining full accuracy
- It is calculated as the inverse of total time required for one successful conversion
- 1) For ADC's without S&H

$$f = throughput = \frac{1}{\tau_c}$$

- 2) For ADC's with S&H: other delays affect the throughput

$$f = throughput = \frac{1}{\sum \tau}$$

where $\sum \tau = \tau_c + \tau_{acq} + \tau_{ap}$

TYPES of ADC

Flash

Integrating

SAR

Sigma-Delta

Pipeline

Why Different Types of ADC?

- Various ADC Types exist as a result of different requirements imposed instrumentation, control, audio and video applications
- Obviously speed / conversion rate is not critical in dc or slowly changing measurands
- However, resolution might be important

Types of ADCs

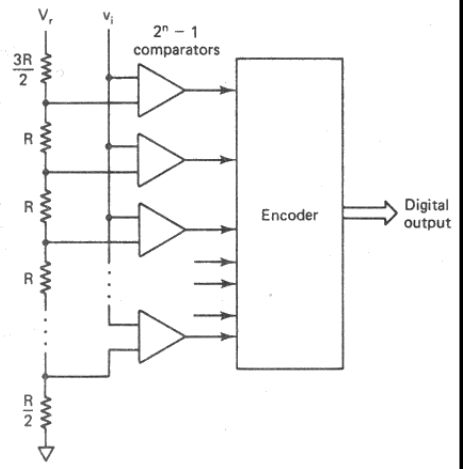
- Most ADC types have the following two blocks in common:
- Comparator ==> Vo= logic "1" if V(+) > V(-)
- Vo= logic "0" if V(+) < V(-)
- i.e. comparator is a 1 bit ADC
- Precise and stable voltage reference

Direct Conversion//Flash//Parallel ADC

- Very High speed of conversion with sampling rate up to 1 GS/s
- For an ADC $\tau c = \frac{1}{1\ GS/s}$=1 ns
- Resolution is limited to 8 bits due to increased number of comparators (2^n -1) and resistors (2^n)

Parallel or Flash ADC

- Very High speed conversion
  - Up to 1GHz and 8 bit resolution
  - Video, Radar, Digital Oscilloscope
- **Single Step Conversion**
- $2^n - 1$ comparator
- 2^n Precision Resistive Network
- Priority Encoder
- Resolution is limited
  - Large number of comparator and resistors in IC ==> bigger die and higher cost



Parallel or Flash ADC

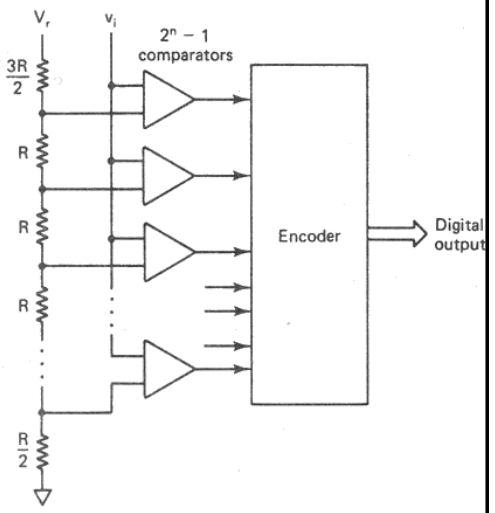
2 bit ADC

Δ=1/4

3 comparators

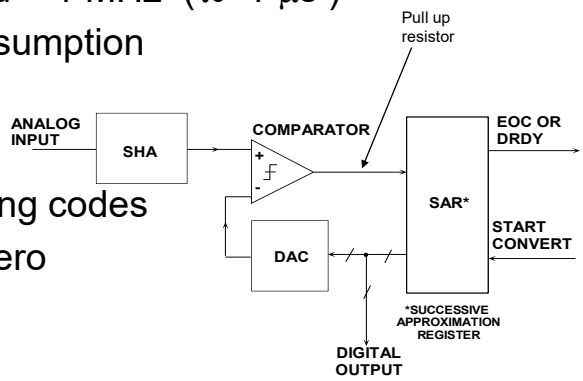
4 Resistors

D3	D2	D1	b2	b1
0	0	0	0	0
0	0	1	0	1
0	1	1	1	0
1	1	1	1	1

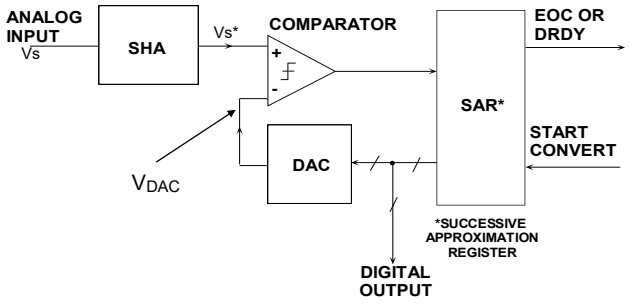


SAR ADC

- SAR – successive approximation register
- SAR ADC are probably the most widely used
- 8-16 bits ;
- Moderate speed ~ 1 MHz (τ_c=1 μs)
- Low power consumption
- Low cost
- Require S&H
- Can have missing codes
- simple to autozero



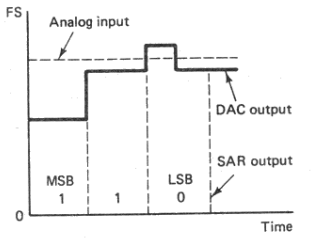
SAR ADC



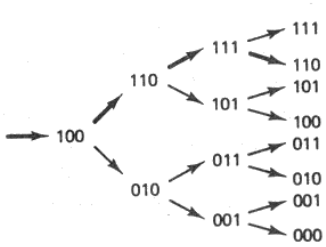
- Based on SAR Register
- DAC output "V_{DAC}" is used as a reference
- Comparator to compare Vs with VDAC
- Final result is reached after n-steps,
- In each step 1 bit conversion is done
- Each step takes one clock cycle

Successive Approximation ADC

- Circuit waveform



- Logic Flow (binary search)

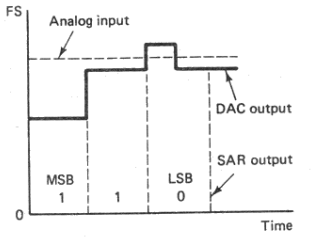


- **Conversion Steps**
- 1. initially SAR provides an output corresponding to have the range (100..0)
- 2. DAC outputs an analog voltage V_{DAC} which if found greater than Vs , then MSB is set to "1", otherwise MSB=0
- 3. IF MSB is reset to 0, next bit is tried until V_{DAC} =Vs
- 4. n -conversion steps, each takes a clock period

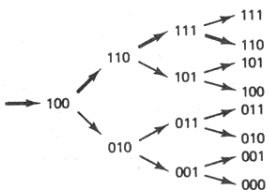
70

Successive Approximation ADC

- Circuit waveform



- Logic Flow (binary search)



- Conversion Time
  - n clock for n-bit ADC
  - Fixed conversion time
- Serial Output is easily generated
  - Bit decision are made in serial order
    - τc= n Tclock
    - Tclock=1/fclock

Typical Applications

- Instrumentation
- Industrial control
- Data acquisition

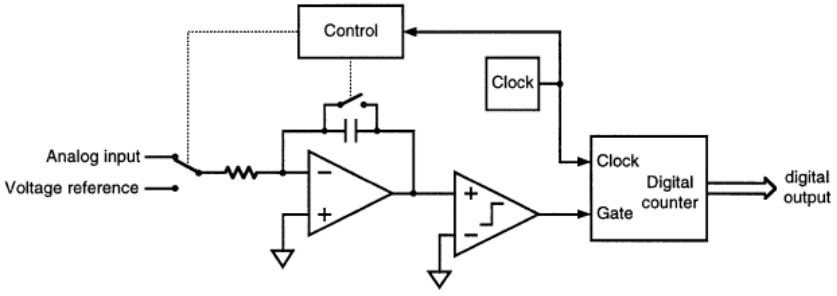
71

Example

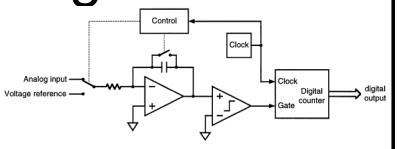
- VFSR=5V, 4 bit, SAR ADC, Vs=Vin=3.127 V, explain how conversion is done?
- Solution:
- Suppose Output is $b_1b_2b_3b_4$
- (1) let b1=1, b2=b3=b4=0
- 1000 ==> VDAC=8/2^4*VREF=2.5V
- Check VDAC > Vs ? (2.5 > 3.127? ==> NO ==> set **b1=1**
- (2) set b2=1, b3=b4=0
- 1100 ==> VDAC=10/2^4*VREF=3.75V
- Check VDAC > Vs ? (3.75 > 3.127? ==> Yes ==> Reset **b2=0**
- (3) set b3=1, b4=0
- 1010 ==> VDAC=10/2^4*VREF=3.1255V
- Check VDAC > Vs ? (3.125 > 3.127? ==> No ==> set **b3=1**
- (4) set b4=1, 1011 ==> VDAC=11/2^4*VREF=3.4375V
- Check VDAC > Vs ? (3.4375 > 3.127? ==> Yes ==> Reset **b4=0**
- **Final Result: 1010**

Dual Slope Integrating ADC

- *Integrating* converters are used for low-speed, high-resolution applications such as voltmeters.
- They are conceptually simple, consisting of an integrating amplifier, a comparator, a digital counter, and a very stable capacitor for accumulating charge
- The most common integrating ADC in use is the dual-slope ADC



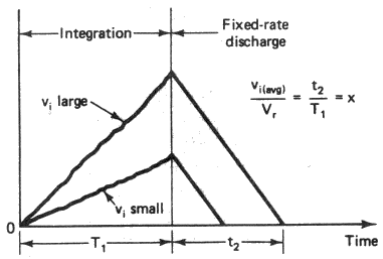
Dual Slope Integrating ADC



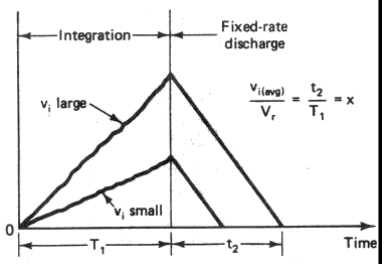
- Initially, the capacitor is discharged and so has no voltage across it.
- At time 0, the input to the integrator is switched to the analog input and the capacitor is allowed to charge for an amount of time, T1, which is fixed.
- Its rate of charging and thus its voltage at T1 are proportional to the input voltage.
- At time T1 the input switch flips over to the voltage reference, which has a negative value so that the capacitor will begin to discharge at a rate proportional to the reference.
- The counter measures how long it takes to discharge the capacitor completely.

- If the capacitor is of high quality, the ratio of the discharge time to the charge time is proportional to the ratio of the input voltage to the voltage reference, and so the counter output represents the analog input voltage.
- t2/T1=Vi(avg)/VREF
- Operation
  - Integrate $\int_0^{T_1} v_i dt$
  - Reset and integrate $\int_0^{t_2} V_r dt$
  - Thus $T_1 v_{i(AVG)} = t_2 V_r$
  - → $v_{i(AVG)} = V_r \frac{t_2}{T_1}$
- Applications
  DMM(Digital Multimeter)

$$t_2 = Count * Tclock$$

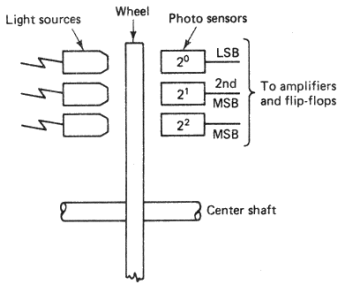


- Integrating converters do not sample the voltage itself; they *average* the voltage over the integration period and *then* they sample the average that is accumulated on the capacitor.
- This tends to reject noise that conventional sampling cannot, especially periodic noises. Most integrating ADCs operate with an integration period that is a multiple of one AC line period (1/60 or 1/50 s) so that any potential interference from stray electric or magnetic fields caused by the power system is canceled.
- Low speed
- High resolution and low cost
- Very stable
- Excellent Noise Rejection
  - High frequency noise cancelled out b
  - Proper T_1 eliminates line noise
  - Easy to obtain good resolution

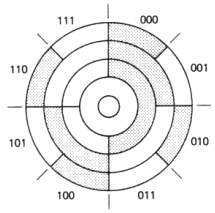


Shaft Encoder

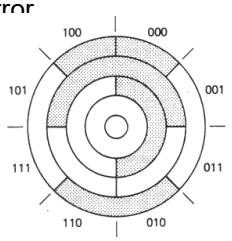
- Electromechanical ADC
  - Convert shaft angle to digital output
- Encoding
  - Optical or Magnetic Sensor
- Applications
  - Machine tools, Industrial robotics, Numerical control



- Binary Encoder
  - Misalignment of mechanism causes large error
    - Ex: 011 → 111 (180deg)



- Gray Encoder
  - Misalignment causes 1 LSB error

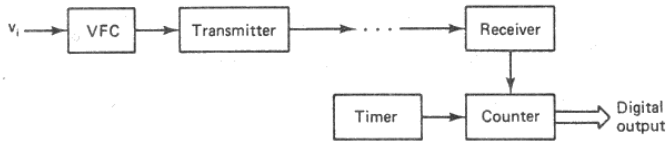


90

Voltage to Frequency ADC

- VFC (Voltage to Frequency Converter)
  - Convert analog input voltage to train of pulses
- Counter
  - Generates Digital output by counting pulses over a fixed interval of time

- Low Speed
- Good Noise Immunity
- High resolution
  - For slow varying signal
  - With long conversion time
- Applicable to remote data sensing in noisy environments
  - Digital transmission over a long distance



91

ADC Comparison

Characteristic	Flash	Pipeline	SAR	Sigma-Delta	Integrating
Throughput Samples/sec	1	2	3	4	5
Resolution	5	3	4	2	1
Latency Sample to output Tc	1	3	2	4	5
Power consumption	Constant High	Constant Low	Variable Low	Constant Medium	Constant Low

END ADC ==>

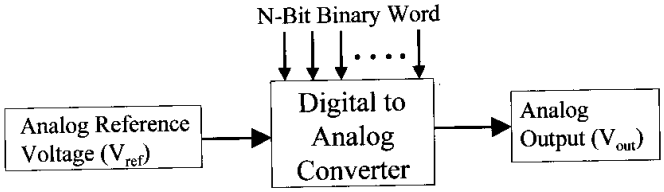
==> START DAC

What is a DAC?
(Digital to Analog Converter)

- A digital to analog converter (DAC) is a device that converts n-bit (parallel) digital input into an analog voltage or current output.
- Primary output is a current which can be easily converted to a voltage

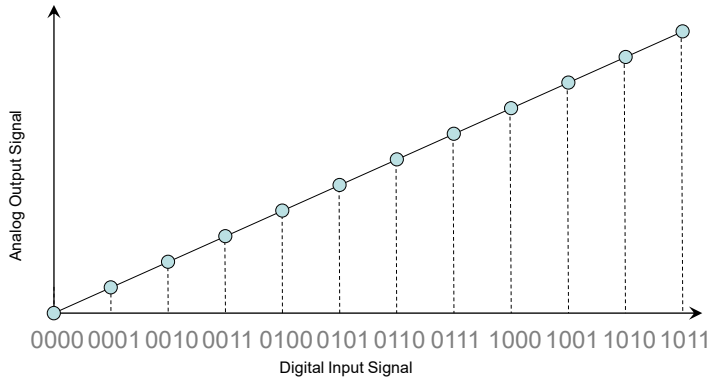
Principal components of DAC

- Consists off:
- Network of analog switches controlled by the input code
- Network of weighted resistors
- The switches control currents or voltages derived from a precise reference voltage
- Output current/voltage represents the ration of the input code to the full scale voltage of the reference source



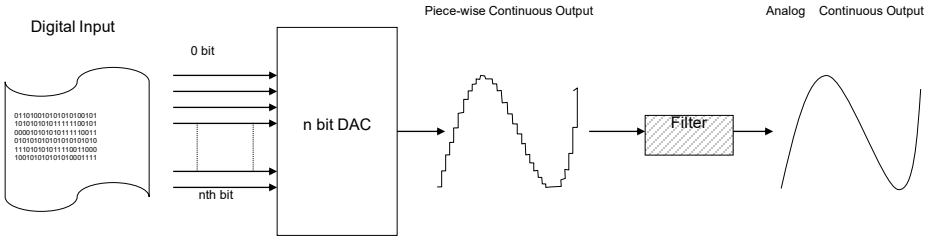
What is a DAC?

- Digital → Analog
- Each binary number sampled by the DAC corresponds to a different output level.



Common Applications

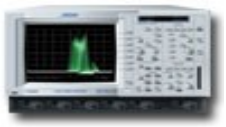
- Used when a continuous analog signal is required.
- Signal from DAC can be smoothed by a Low pass filter



Common Applications:
Function Generators

- Digital Oscilloscopes
  - Digital Input
  - Analog Ouput
- Signal Generators
  - Sine wave generation
  - Square wave generation
  - Triangle wave generation
  - Random noise generation

1



2

Common Applications

Motor Controllers

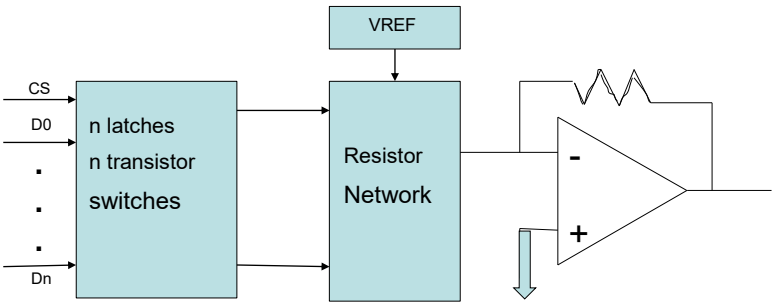
- Cruise Control
- Valve Control
- Motor Control

1 

2 

3 

Typical DAC

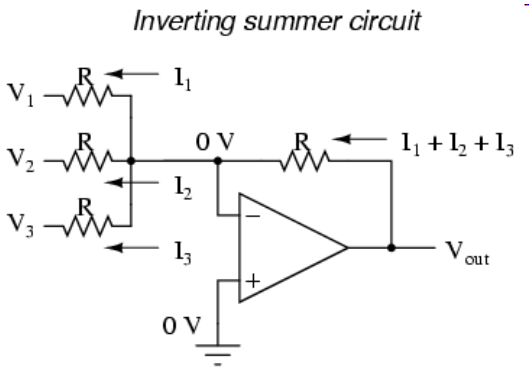


- N latches hold the binary number to be converted
- n transistors controlled by outputs of latches and control a particular resistor each
- VREF precision reference controls output voltage range
- Opamp provides summing function to add together the results of activating multiple switches simultaneously

Types of DAC implementations

- Binary Weighted Resistor
- R-2R Ladder

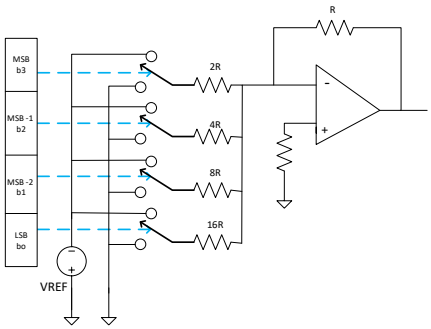
Binary Weighted Resistor



$$V_{out} = -(V_1 + V_2 + V_3)$$

- Start with summing op-amp circuit
- Input voltage either high or ground
- Adjust resistor weighting to achieve desired Vout

Binary Weighted Resistor



- *Details*
  - Use transistors to switch between high and ground
  - Use resistors scaled by two to divide voltage on each branch by a power of two
  - V_1 is MSB, V_4 LSB in this circuit
- *Assumptions:*
  - Ideal Op-Amp
  - No Current into Op-Amp
  - Virtual Ground at Inverting Input

$$I_3 = \frac{V_{REF}}{2R} \rightarrow V_{out3} = \frac{V_{REF}}{2}$$

$$I_2 = \frac{V_{REF}}{4R} \rightarrow V_{out2} = \frac{V_{REF}}{4}$$

$$I_1 = \frac{V_{REF}}{8R} \rightarrow V_{out1} = \frac{V_{REF}}{8}$$

$$I_0 = \frac{V_{REF}}{16R} \rightarrow V_{out0} = \frac{V_{REF}}{16}$$

$$V_{out} = R(I_3 + I_2 + I_1 + I_0)$$

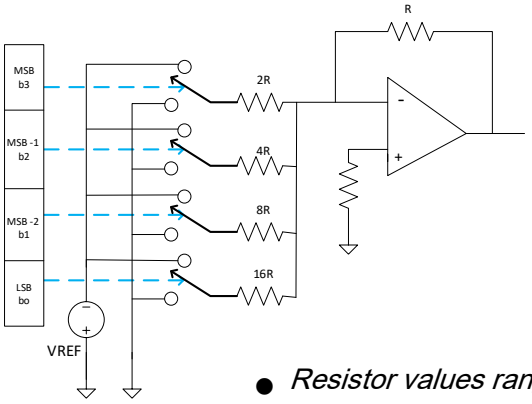
$$= R\left(\frac{V_{REF}}{2R} + \frac{V_{REF}}{4R} + \frac{V_{REF}}{8R} + \frac{V_{REF}}{16R}\right)$$

$$= V_{REF}\left(b_3.\frac{1}{2} + b_2.\frac{1}{4} + b_1.\frac{1}{8} + b_0.\frac{1}{16}\right)$$

$$= V_{REF}\left(b_3.2^{-1} + b_2.2^{-2} + b_1.2^{-3} + b_0.2^{-4}\right)$$

Binary Weighted Resistor

Ken Marek



For a 12 bit DAC

$R = 5k\Omega$

$2^n R = 20.48M\Omega$

- *Resistor values range from R to 2^n*R which must be matched*
- *High value of resistance requires more silicon die area*
- *This results in higher cost*

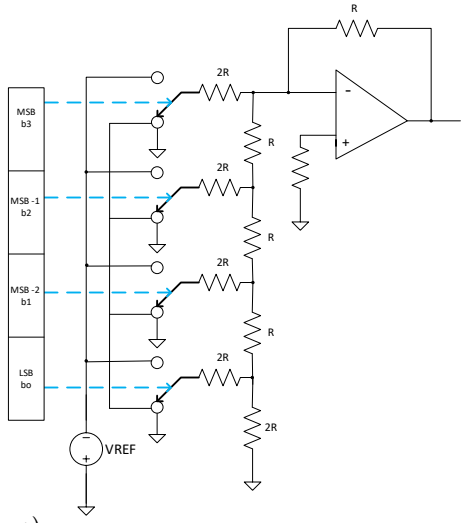
Ken Marek

Binary Weighted Resistor

- Advantages
  - Simple
  - Fast
- Disadvantages
  - Need large range of resistor values (2048:1 for 12-bit) with high precision in low resistor values
  - Need very small switch resistances
  - Op-amp may have trouble producing low currents at the low range of a high precision DAC

R-2R Ladder

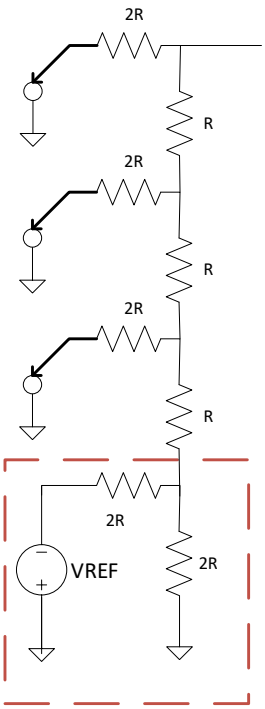
- Each bit corresponds to a switch:
  - If the bit is high “1”, the corresponding switch is connected to the reference voltage
  - If the bit is low “0” , the corresponding switch is connected to ground.

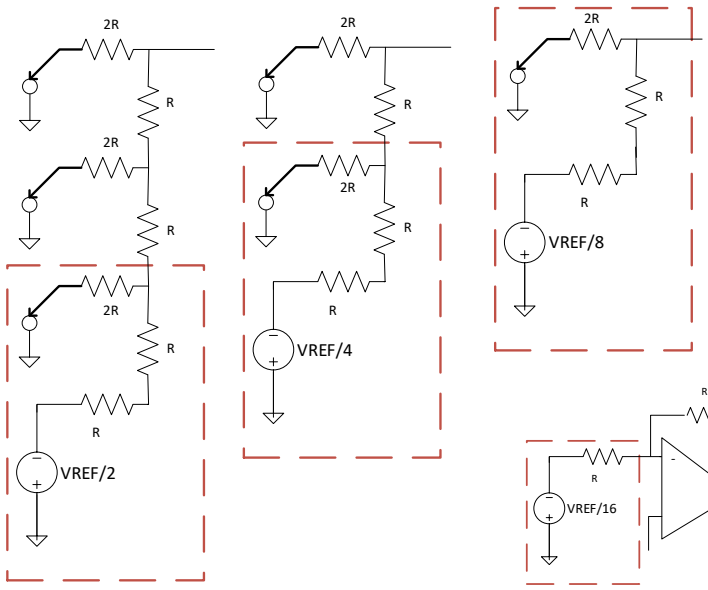


$$V_{out} = \mathrm{V_{REF}}\left(b_3.2^{-1} + b_2.2^{-2} + b_1.2^{-3} + b_0.2^{-4}\right)$$

R-2R Ladder

- Circuit may be analyzed using Thevenin's theorem and superposition by considering one bit high at a time (replace network with equivalent voltage source and resistance)
- Consider input =0001





$$V_{out} = V_{REF}\left(0.2^{-1} + 0.2^{-2} + 0.2^{-3} + 1.2^{-4}\right) = \frac{V_{REF}}{16}$$

- Contribution of each input bit can be found in a similar fashion , by setting its value to 1 while all other bits are set to zero
- R-2R DAC resistor values are limited to two values only R or 2R which is less expensive
- Number of resistors is less: 2n+1 and lower precision is acceptable
- Conversion speed is lower

- Example: an 8 bit DAC with 5V reference has an input 10100111, what is the output?

$$V_{out} = \frac{167}{256} * 5 = 3.2617\,\text{V}$$

- Example: a 10 bit DAC with 10V reference, what input is required to get 6.5V output?

$$V_{out} = \frac{(\text{N})_{10}}{2^{10}} * 10 = 6.5\,\text{V}$$

$$(\text{N})_{10} = \frac{6.5 * 2^{10}}{10} = 665.6$$

$\text{if N} = 665 \Longrightarrow \text{V} = 6.494$

$\text{if N} = 666 \Longrightarrow \text{V} = 6.504$ (closer to required value)

- Highest value of N:
- Unipolar DAC:

$$V_{o\max} = \frac{(\mathrm{Nmax})_{10}}{2^n} * V_{FSR}; \text{where } \mathrm{Nmax} = 2^{\mathrm{n}} - 1$$

$$V_{o\max} = \frac{2^{\mathrm{n}} - 1}{2^n} * V_{FSR}$$

- For Bipolar DAC:

$$V_o = \frac{(\mathrm{N})_{10}}{2^n} * V_{FSR} - \frac{V_{FSR}}{2};$$

$$V_{o\max} = \frac{2^{\mathrm{n}} - 1}{2^n} * V_{FSR} - \frac{V_{FSR}}{2}$$

$$V_{o\min} = -\frac{V_{FSR}}{2}$$

Examples

- A bipolar 10 bit DAC has V_{FSR}=5V and a hexadecimal input 2A4, what is the output? And at what input the output will be zero?
- Solution: 2A4 = 10 1010 0100 ==> $(512+128+32+4)_{10}=676_{10}$
- For Vo=0

$$V_o = \frac{676}{1024} * 5 - \frac{5}{2} = -0.8 \text{ V}$$

$$0 = \frac{(\mathrm{N})_{10}}{1024} * 5 - \frac{5}{2} ==>$$

$$N = 512$$

$$0010\,0000\,0000 ==> (200)_{\mathrm{H}}$$

- Determine how many bits a DAC should have to provide an output voltage increment of 0.04V if V_{FSR}=10V?

$$\Delta = \frac{V_{FSR}}{2^n} = 0.04$$

$$2^n = \frac{10}{0.04} = 250$$

$$n \ln n = \ln 250$$

$$n = \frac{\ln 250}{\ln 2} = 7.966 ==> n = 8$$

Data Acquisition

Need For Data Acquisition

- There are many applications where it is necessary to know, simultaneously, the measured values of several variables associated with a particular process, machine or situation.
- Examples are measurements of temperature measurements at different points in a nuclear reactor core, and components of velocity and acceleration for an aircraft.

Instrumentation

- It would be extremely uneconomic to have several completely independent systems,and a single multi-input/multi-output **data acquisition system** is used.
- Here several elements are ‘time shared’ amongst the different measured variable inputs.

Instrumentation

1.3.4 IEEE-488 (GPIB) remote programmable instruments

- The communications standard now known as GPIB (General Purpose Interface Bus), Was originally developed by Hewlett-Packard in 1965 as a digital interface for interconnecting and controlling their programmable test instruments.
- Originally referred to as the Hewlett Packard Interface Bus (HPIB), its speed, flexibility and usefulness in connecting instruments in a laboratory environment led to its widespread acceptance, and finally to its adoption as a world standard (IEEE-488).
- Since then, it has undergone improvements (IEEE-488.2) and SCPI (Standard Commands for Programmable Instruments), to standardize how instruments and their controllers communicate and operate.
- Evolving from the need to collect data from a number of different stand-alone instruments in a laboratory environment**, the GPIB is a high-speed parallel communications interface that allows the simultaneous connection of up to 15 devices or instruments on a short common parallel data communications bus**.
- **Devices must be placed within 3 meters or so of the host controller/ computer**

Instrumentation

- A device connected to the bus can send data (bytes) to 14 other devices on the bus
- GPIB allows data to be sent at whatever rate the devices on the bus operate.
- Hardware consideration limit the max speed of data transmission to 250 kbytes/s (= 2 Mbits/s).
- GPIB is used to communicate with a set of instruments with the same interface for setting an automatic measurement and control system by a network of instruments

Instrumentation

(for info only)

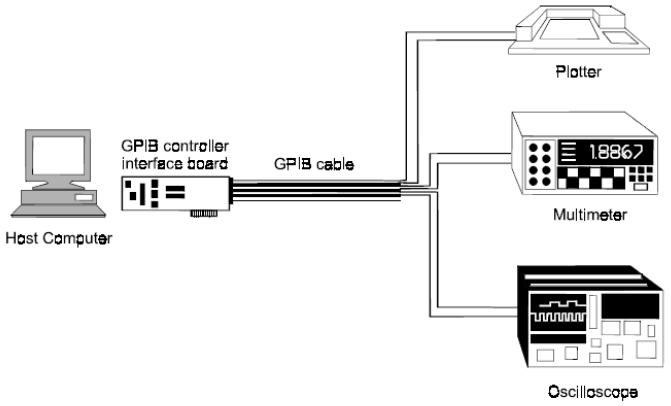
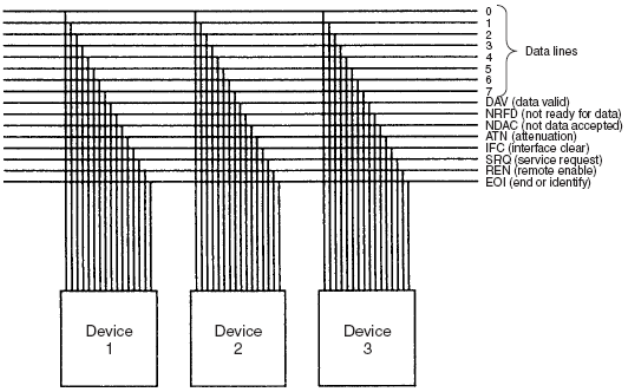


Figure 1.8
A typical GPIB system configuration

GPIB Bus Structure



-16 lines

_8 carry data (or commands)

_8 are bus control lines: 3 provide handshaking to coordinate data transfer and 5 are other bus control functions

Device on GPIB

Are classified into:

- Listener: may receive data over the bus
- Talker: may send data over the bus
- Listener/ Talker:
- Controller: at least one device must act as a controller (usually PC that also act as listener and talker)

*Any one device can perform combination of these functions.

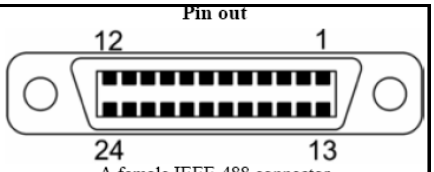
Only 15 devices are allowed on the bus, if more are used, this causes overloading and unreliable operation

- GPIB signal logic:1) TTL voltage levels
  2) negative logic ➔ < 0.8V (logic 1); > 2.5 (logic 0)
- Lines are driven by open collector drivers

IEEE-488 conne

(for info only)

Pin out



A female IEEE-488 connector

Pin 1	DIO1	Data input/output bit.
Pin 2	DIO2	Data input/output bit.
Pin 3	DIO3	Data input/output bit.
Pin 4	DIO4	Data input/output bit.
Pin 5	EOI	End-or-identify.
Pin 6	DAV	Data valid.
Pin 7	NRFD	Not ready for data.
Pin 8	NDAC	Not data accepted.
Pin 9	IFC	Interface clear.
Pin 10	SRQ	Service request.
Pin 11	ATN	Attention.
Pin 12	SHIELD	
Pin 13	DIO5	Data input/output bit.
Pin 15	DIO7	Data input/output bit.
Pin 16	DIO8	Data input/output bit.
Pin 17	REN	Remote enable.
Pin 18	GND	(wire twisted with DAV)
Pin 19	GND	(wire twisted with NRFD)
Pin 20	GND	(wire twisted with NDAC)
Pin 21	GND	(wire twisted with IFC)
Pin 22	GND	(wire twisted with SRQ)
Pin 23	GND	(wire twisted with ATN)
Pin 24	Logic ground	

Other Interfaces

- GPIB used for managing large measurement systems
- Serial Interface is often used when a single instrument is to be connected to a PC over long distance
- RS232 was originally developed in 1960s and it is slow, not flexible and rarely used on instruments, however it is used for specific applications such as reading in data from remote dc sensors and sending data to loggers
- Other more modern, serial asynchronous data transmission protocols include RS422, RS423, RS449, RS485 and USB
- RS stands for recommended standard

RS232 basic characteristics

- 1 driver, 1 receivers
- ~ 50 Feet
- Rate : 20Kb/s
- Single Ended

Instrumentation

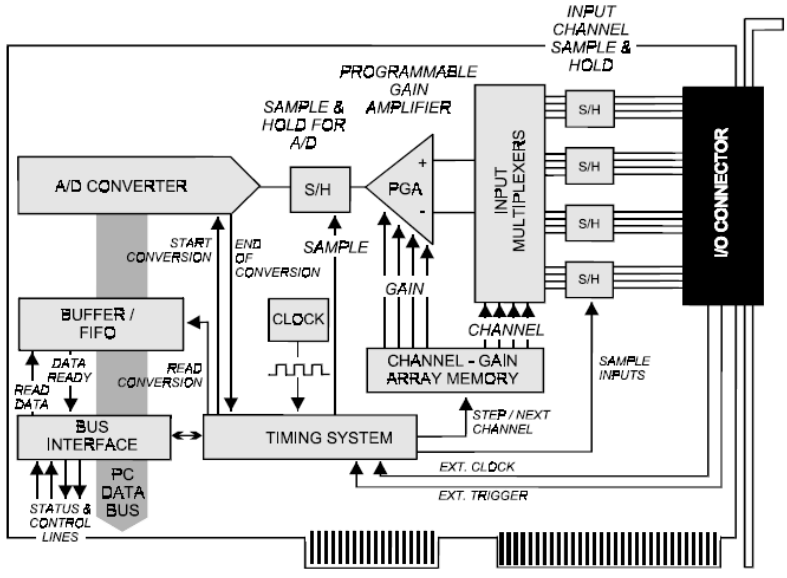
RS-422

- 1 driver, 10 receivers
- 4000 Feet
- Rate : 100Kb/s -10Mb/s
- Differential

RS-485

- 32 driver, 32 receivers
- 4000 Feet
- Rate : 100Kb/s -10Mb/s
- Differential

A/D Board



Instrumentation

- Total throughput, for multiple conversions on different channels, is often increased by overlapping parts of this cycle.
- For example:

  While the A/D converter is busy converting the S/H output, the next channel/gain pair can be output to the multiplexer and PGA, so that their settling and delay times are overlapped with the A/D conversion time.
- The timing circuitry may also include a block-sampling mode, which allows blocks of samples to be collected at regular intervals at the A/D board's maximum sampling rate.

Sampling Techniques

- These techniques are discussed in the following sections:
- **Continuous channel scanning**
- **Simultaneous sampling**
- **Block mode operations**

Continuous channel scanning

- The method of sampling that facilitates the connecting of the required input channel to the A/D converter at a constant rate is known as continuous channel scanning.
- Continuous channel scanning allows channels to be sampled in a pre-determined and arbitrary order (e.g. channel 5, channel 1, channel 11), as well as at different sampling rates.
- An example of this would be the sampling of three channels in the following order (channel 5, channel 1, channel 11, channel 1). Channel 1 is being sampled at twice the rate as channels 5 & 11, which for an A/D board with throughput of 100 kHz represents a sampling rate of 50 kHz.
- Channels 5 & 11 are sampled at 25 kHz.
- There are two methods of continuous channel scanning, either under software control or by on-board hardware control using Channel Gain Array.

- When the input multiplexer switches between channels, a time skew/delay is generated between each channel sampled.
- On an A/D board being sampled at its maximum total throughput of 200 kHz, the minimum channel-to-channel time skew/delay between samples on different channels is 5 μs. Since the skew is additive from channel to channel, the total time skew between the first and last samples, when 16 channels are being sampled, is 80 μs.
- Time skew between signal measurements taken on different channels can lead to an inaccurate portrayal of the events that generated the signals as demonstrated on next slide.

Error due to continuous scanning

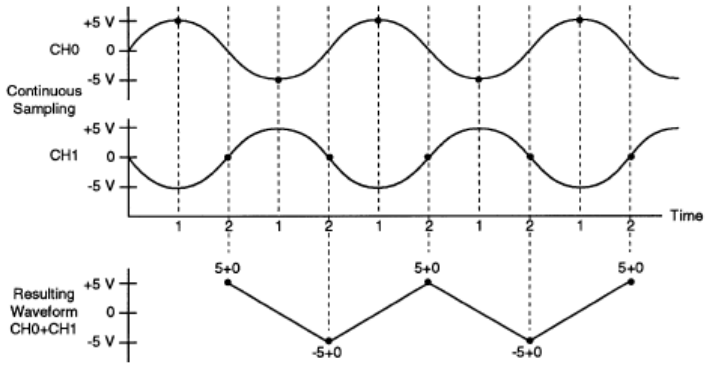


FIGURE 96.9 If the channel skew is large compared with the signal, then erroneous conclusions may result.

Simultaneous Sampling

- **Where the time relationship between each channel sampled is unimportant, or the skew/delay is negligible compared to the speed of the channel scan rate, such delays are not significant**.
- **In many applications, however, such as those dealing with accurate phase measurements or high-speed transient analysis, time skew between channels is unacceptable, since it is crucial to determine the output of several signals on different channels, at precisely the same time.**
- To avoid the timing errors introduced when continuously sampling from one input channel to the next, special applications require A/D boards capable of simultaneous sampling.
- These A/D boards are fitted with **so-called simultaneous sample and hold devices on all input channels**. The sample and hold device on each input channel holds the sampled data until the A/D converter can scan each channel.

Important

- Maximum throughput per channel = Total throughput / # of channels
- For example if you wish to sample 4 channels at 50 kHz each, you need a board with throughput of 200 kHz