ADC Review

Last updated 6/3/20

ADC Review

These slides review the operation of an Analog to Digital converter

Upon completion: You should be able to describe and operation of an ADC and complete simple calculations

© tj

ADC Review

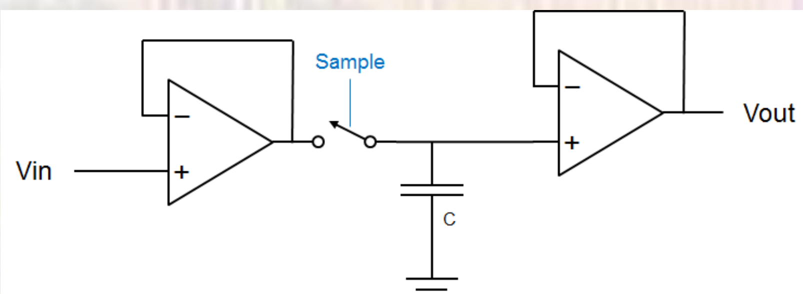
- Analog to Digital Conversion
  - Most of the real world is analog
    - temperature, pressure, voltage, current, …
  - To work with these values in a computer we must convert them into digital representations
  - Three steps to this conversion
    - Sampling
    - Quantizing
    - Encoding

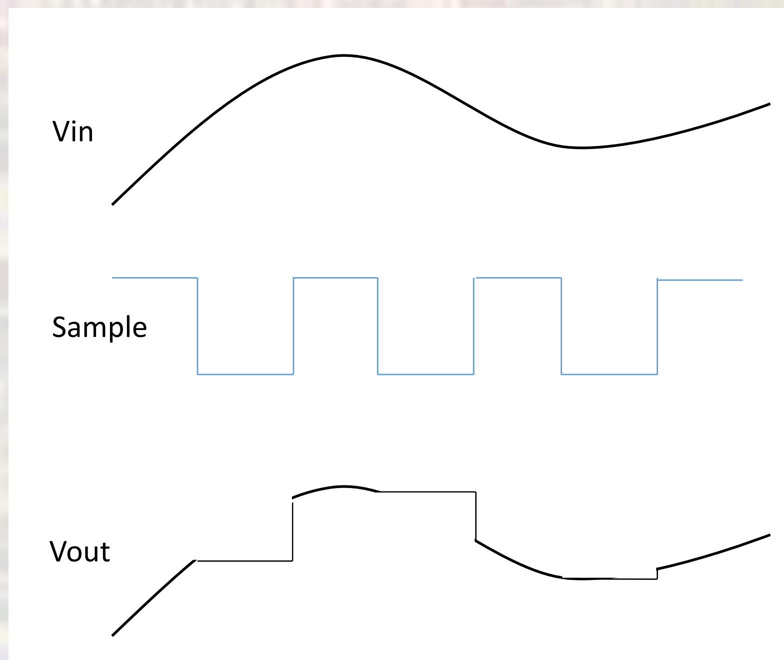
© tj

ADC Review

- Sampling

  - A to D Conversion takes a finite amount of time
  - What if the input changes during this time?
  - We must take a snapshot of the input → Sample and Hold



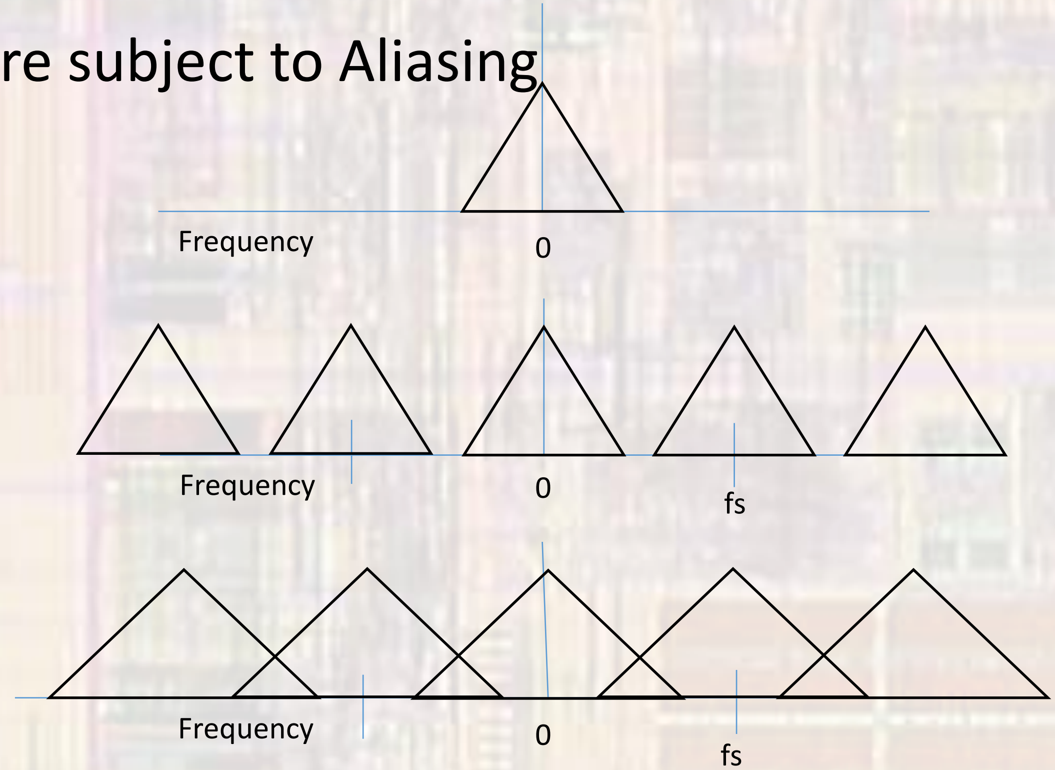


 © tj

ADC Review

- Sampling
  - Sampling is a kind of MODULATION
  - Modulation systems are subject to Aliasing
  - Fin < fs/2
    - Fs: Nyquist rate

→ LPF the input
(anti-aliasing filter)

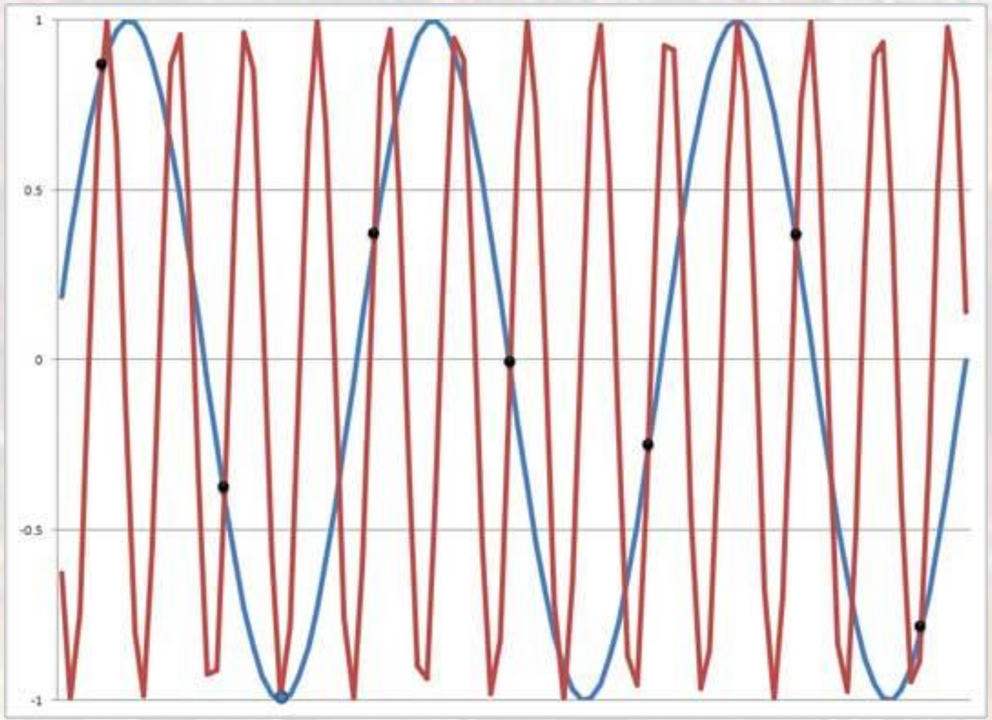


 © tj

ADC Review

- Sampling

    - Example of analog aliasing



http://arstechnica.com/features/2007/11/audiofile-analog-to-digital-conversion/

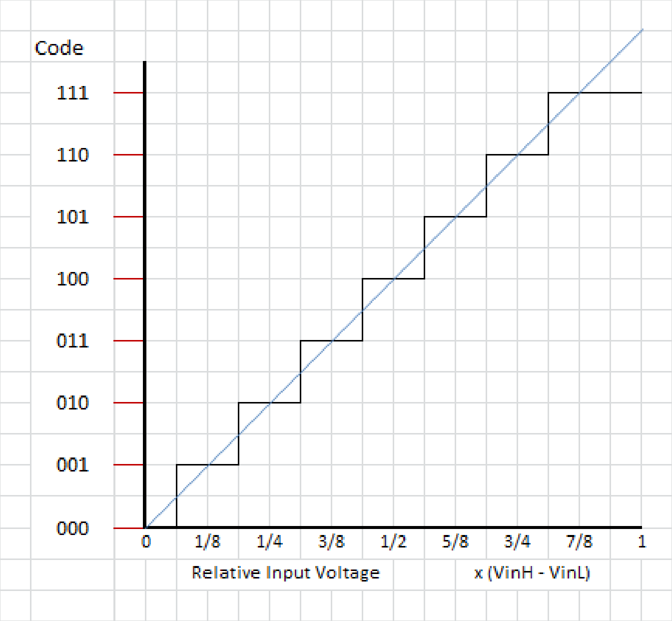
ADC Review

- Quantizing

- In the A to D process we are converting an "infinite" resolution analog signal into a finite number of digital bits
- Converters use reference voltages to set the range of allowed input voltages - Vref-H , Vref-L
- Each binary step represents

  $(V_{ref\text{-}H} – V_{ref\text{-}L}) / 2^n$ for an n bit conversion

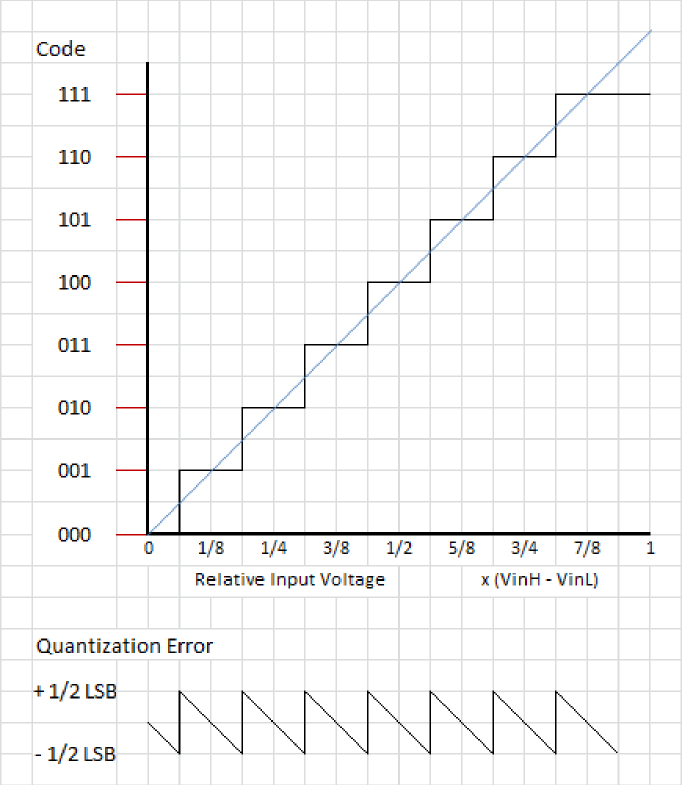
- e.g. 0V – 1V input converted to 3 bit digital value
  - each binary step represents 0.125V
  - since 000 typically represents 0.0V, 111 represents 0.875V



© tj

ADC Review

- Quantizing



- Quantization error looks like noise on the signal (Quantization Noise)
- Dynamic Range is a measure of signal to noise ratio. (SNR in dB)
- For an AtoD the Dynamic Range is the measure of signal to Quantizing Noise ratio (SQNR)
- SQNR = $20 \log_{10}(2^n/(1/2 - (-1/2))$

  $= 20 \log_{10} 2^n$
  - 8bit → 48dB
  - 10bit → 60dB

n	steps	Step Size rel to Vref-H - Vref-L	SQNR (dB)
1	2	0.5	6
2	4	0.25	12
3	8	0.125	18
4	16	0.0625	24
5	32	0.03125	30
6	64	0.015625	36
7	128	0.0078125	42
8	256	0.00390625	48
9	512	0.001953125	54
10	1024	0.000976563	60
11	2048	0.000488281	66
12	4096	0.000244141	72

© tj

ADC Review

- A/D Conversion Example
- 10 bit converter with VrefH=3.0V, VrefL=0.0V
- If the input is 2V, what is the output code

VrefH-VrefL = 3V range

10 bit converter step size = range/2^{10} = 2.9297mV/step

2V / 2.9297mV/step = 682 steps from VrefL

10 1010 1010

 © tj

ADC Review

- Successive Approximation A to D

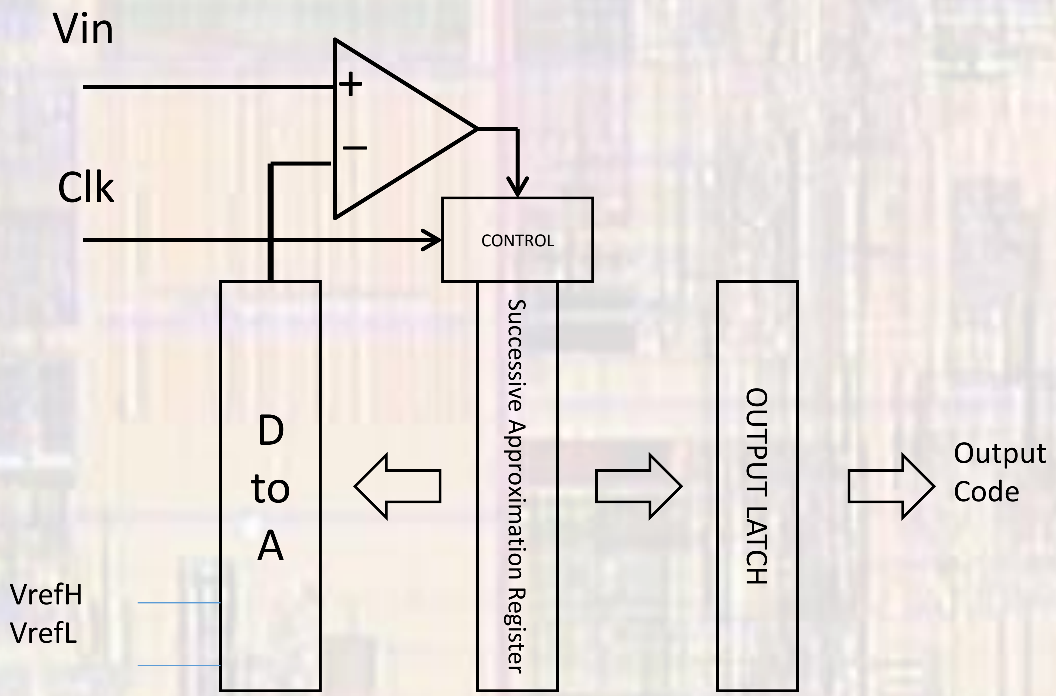
  - Uses an iterative process to determine the correct digital value for the analog input

  - Requires
    - Input (sample and held)
    - A register to hold the current estimate of the digital value
    - D to A converter to convert the digital estimate back to analog
    - A comparator to determine if the estimate is above or below the actual input value
    - Control logic to run the process

  - Uses a binary search to find the nearest code value to the input value

ADC Review

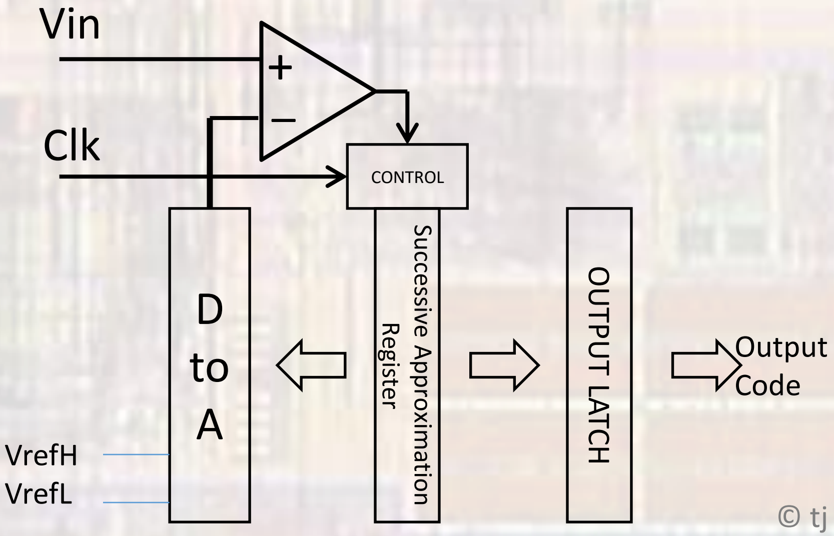
- Successive Approximation A to D



ADC Review

- Successive Approximation A to D

- The control logic resets the SAR before each conversion
- The control logic then sets the msb
  - The DtoA converts this to ½ the reference voltage
  - The comparator tests to see if the input is above or below this value
    - if above, the 1 in the msb stays
    - if below, the msb is reset to zero
- The control logic then sets the msb-1 bit
  - The DtoA converts this to the appropriate voltage level
  - The comparator tests to see if the input is above or below this value
    - if above, the 1 stays
    - if below, the msb-1 bit is reset to 0
- The control logic then sets the msb-n bit
  - The DtoA converts this to voltage
  - The comparator tests to see if the input is above or below this value
    - if above, the 1 stays
    - if below, the msb-n bit is reset to 0



© tj

ADC Review

- Successive Approximation A to D
  - Test to see if input is
    - > or < new "midpoint"
      - if < , clear bit
      - if >, set bit