

ADC Review

Last updated 7/20/23

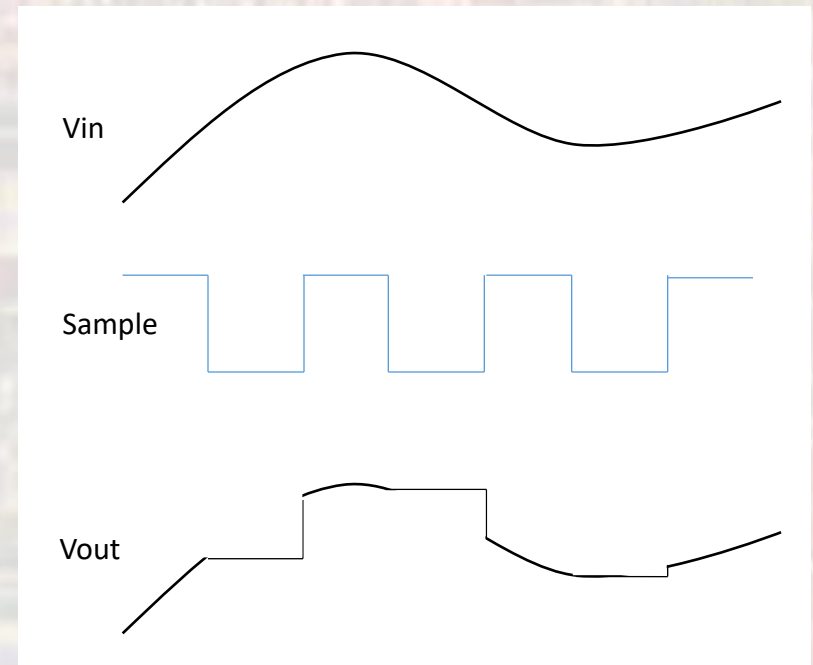
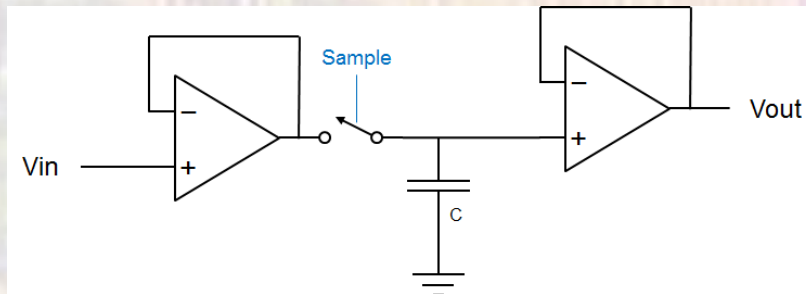
A/D

- 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

A/D

- 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

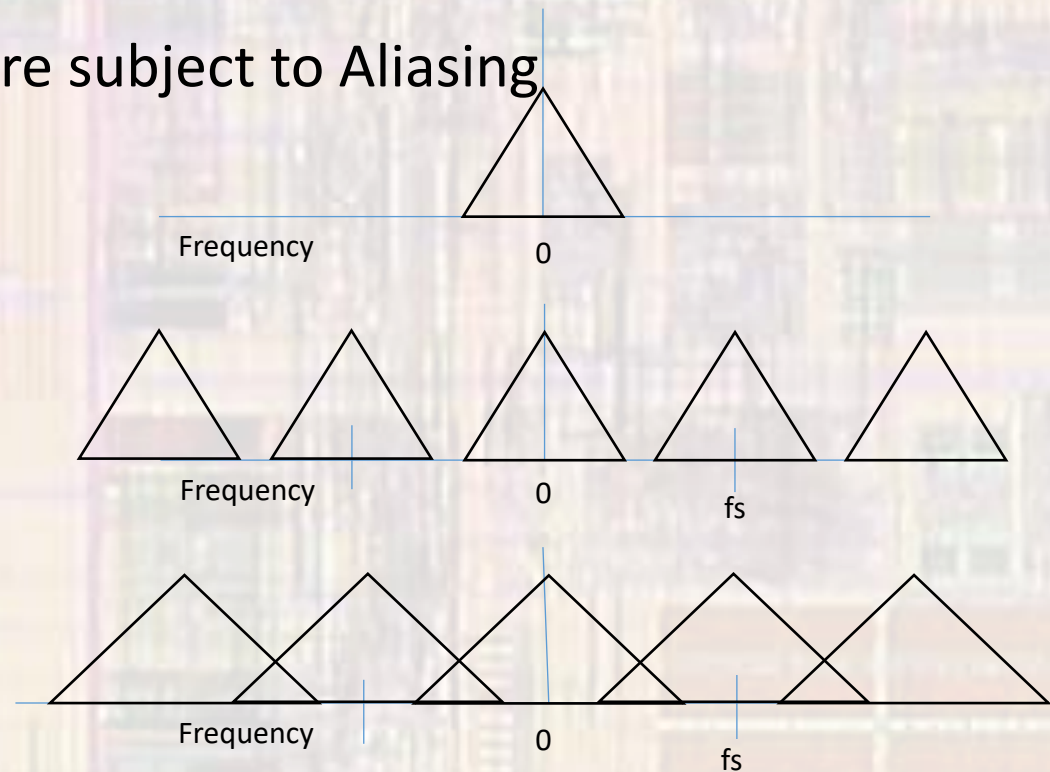


A/D

- Sampling
 - Sampling is a kind of MODULATION
 - Modulation systems are subject to Aliasing
- $f_{in} < f_s/2$

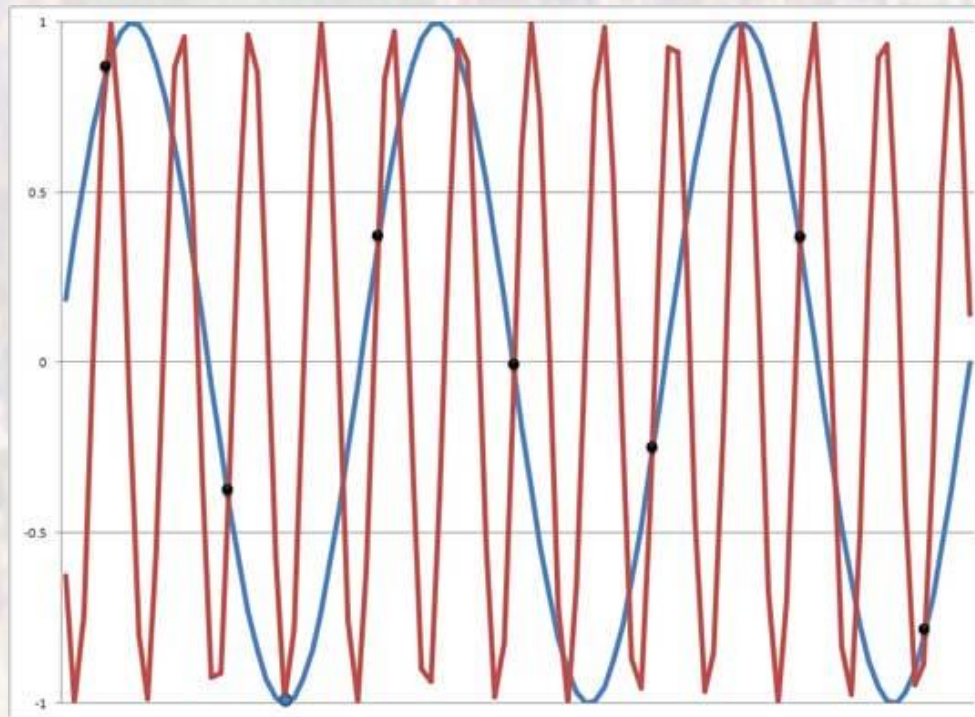
- f_s : Nyquist rate

→ LPF the input
(anti-aliasing filter)



A/D

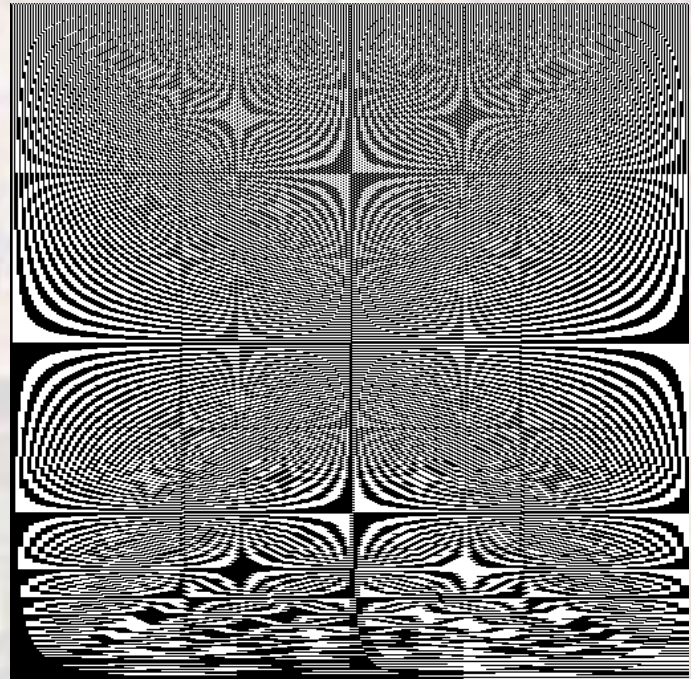
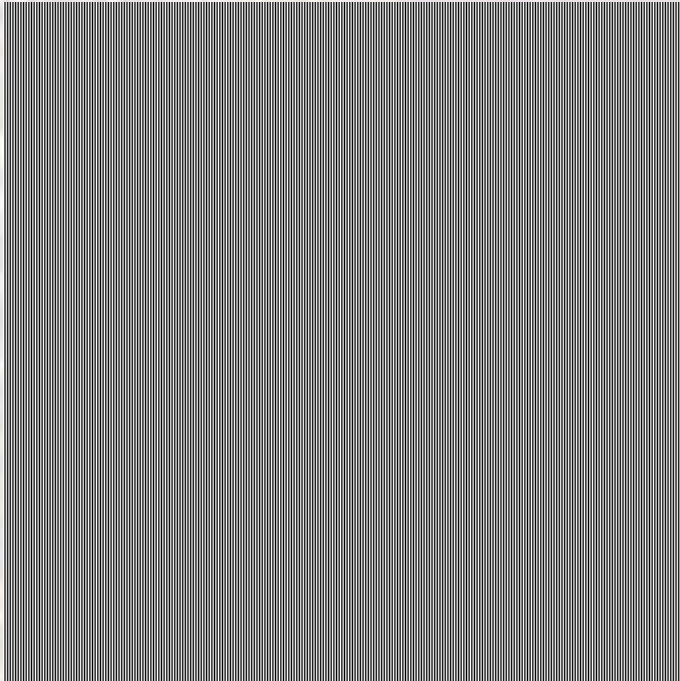
- Sampling
- Example of analog aliasing



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

A/D

- Sampling
 - Example of digital aliasing



<http://www.cs.unm.edu/~brayer/vision/perception.html>

A/D

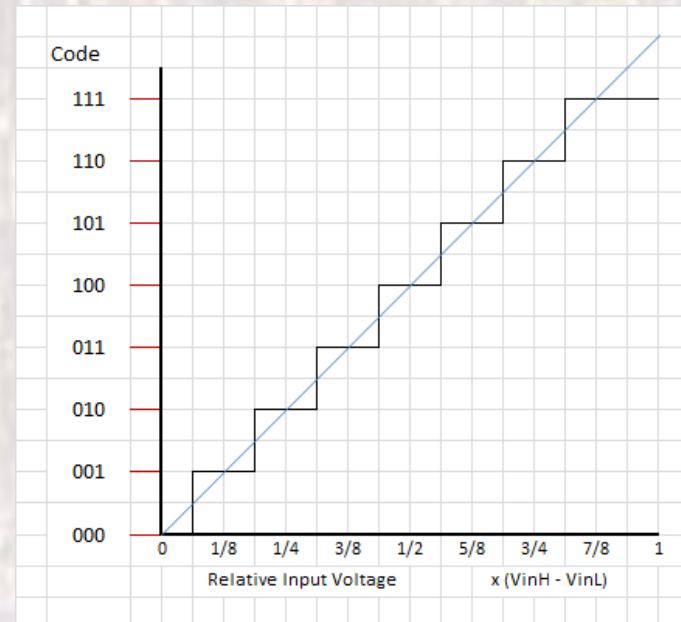
- Quantizing

- 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 - $V_{\text{ref-H}}$, $V_{\text{ref-L}}$

- Each binary step represents

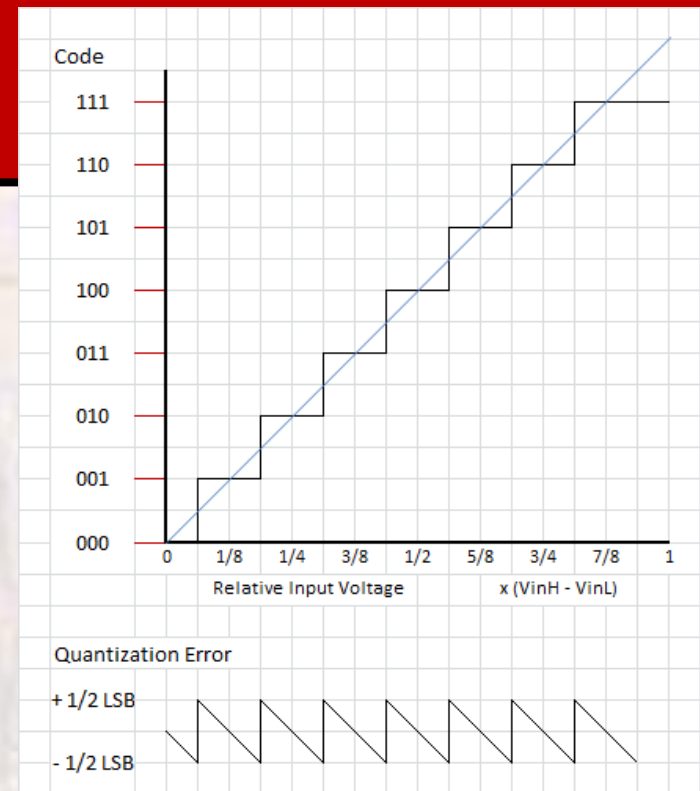
$$(V_{\text{ref-H}} - V_{\text{ref-L}}) / 2^n \text{ for an } n \text{ bit conversion}$$

- 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



A/D

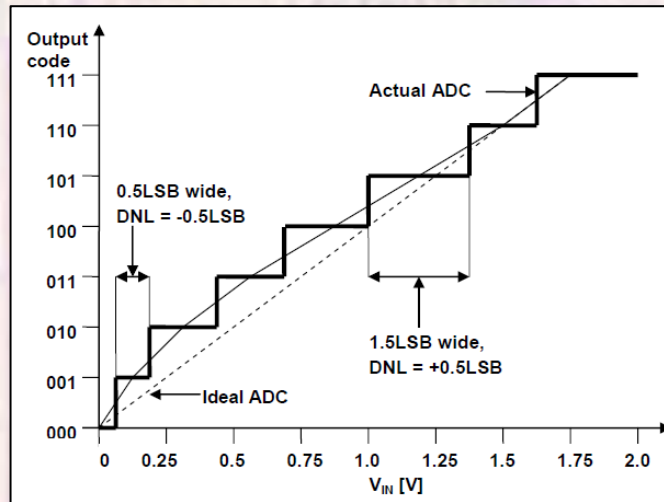
- Quantizing
 - Quantization error looks like noise on the signal (Quantization Noise)
 - Dynamic Range is a measure of signal to noise ratio. (SNR in dB)
 - For an AtoD the Dynamic Range is the measure of signal to Quantizing Noise ratio (SQNR)
 - $SQNR = 20 \log_{10}(2^n / (1/2 - (-1/2)))$
 $= 20 \log_{10} 2^n$
 - 8bit \rightarrow 48dB
 - 10bit \rightarrow 60dB



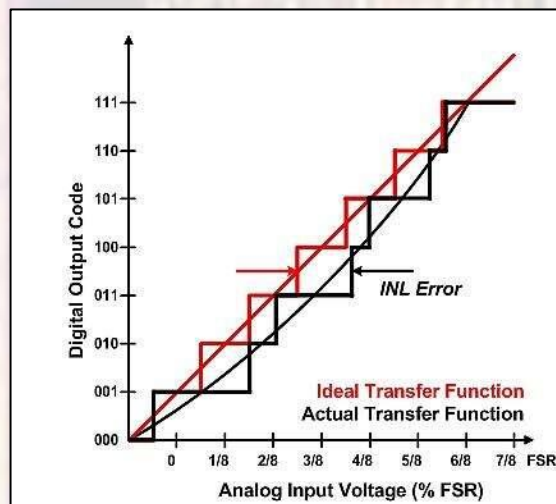
n	steps	Step Size rel to Vref-H - Vref-L	SQNR (dB)
1	2	0.5	6
2	4	0.25	12
3	8	0.125	18
4	16	0.0625	24
5	32	0.03125	30
6	64	0.015625	36
7	128	0.0078125	42
8	256	0.00390625	48
9	512	0.001953125	54
10	1024	0.000976563	60
11	2048	0.000488281	66
12	4096	0.000244141	72

A/D

- Other errors
 - Differential non-linearity



- Integral non-linearity



A/D

- A/D Conversion Example
- 10 bit converter with $V_{refH}=3.0V$, $V_{refL}=0.0V$
- If the input is 2V, what is the output code

$V_{refH}-V_{refL} = 3V$ range

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

$2V / 2.9297mV/step = 682$ steps from V_{refL}

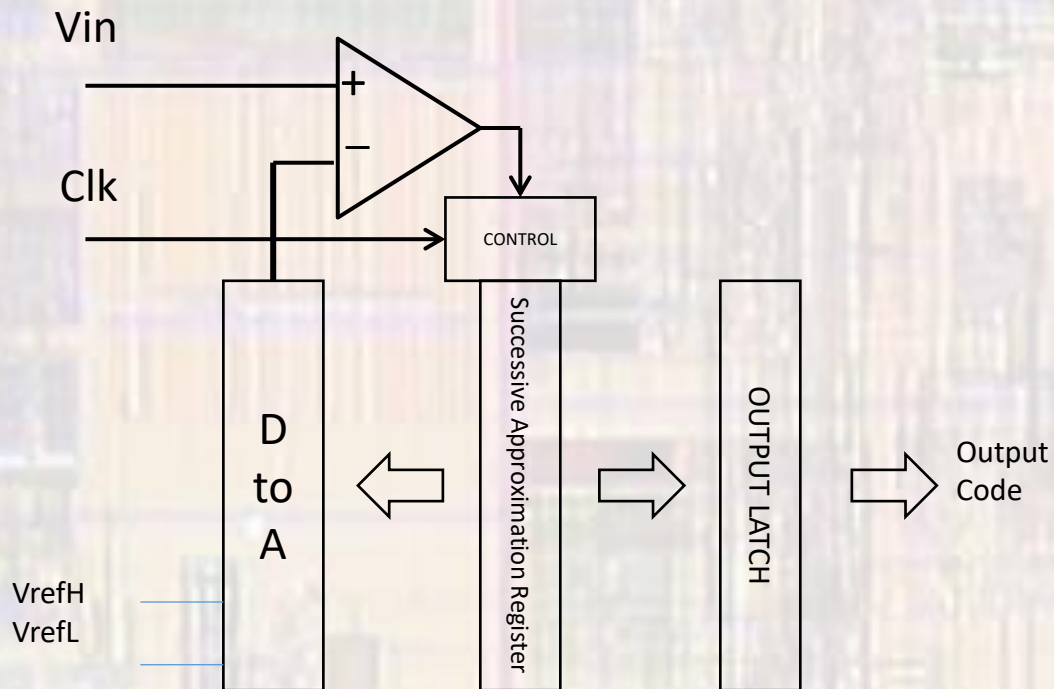
10 1010 1010

A/D

- 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

A/D

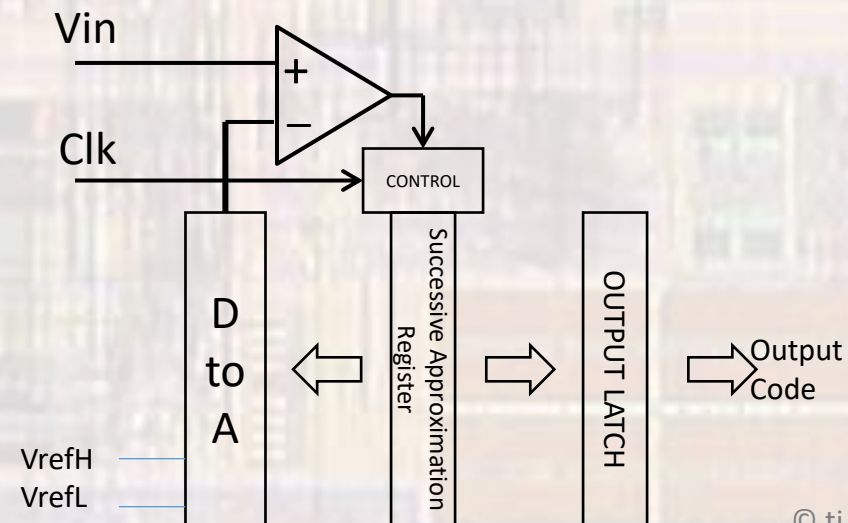
- Successive Approximation A to D



A/D

• 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 $\frac{1}{2}$ 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



A/D

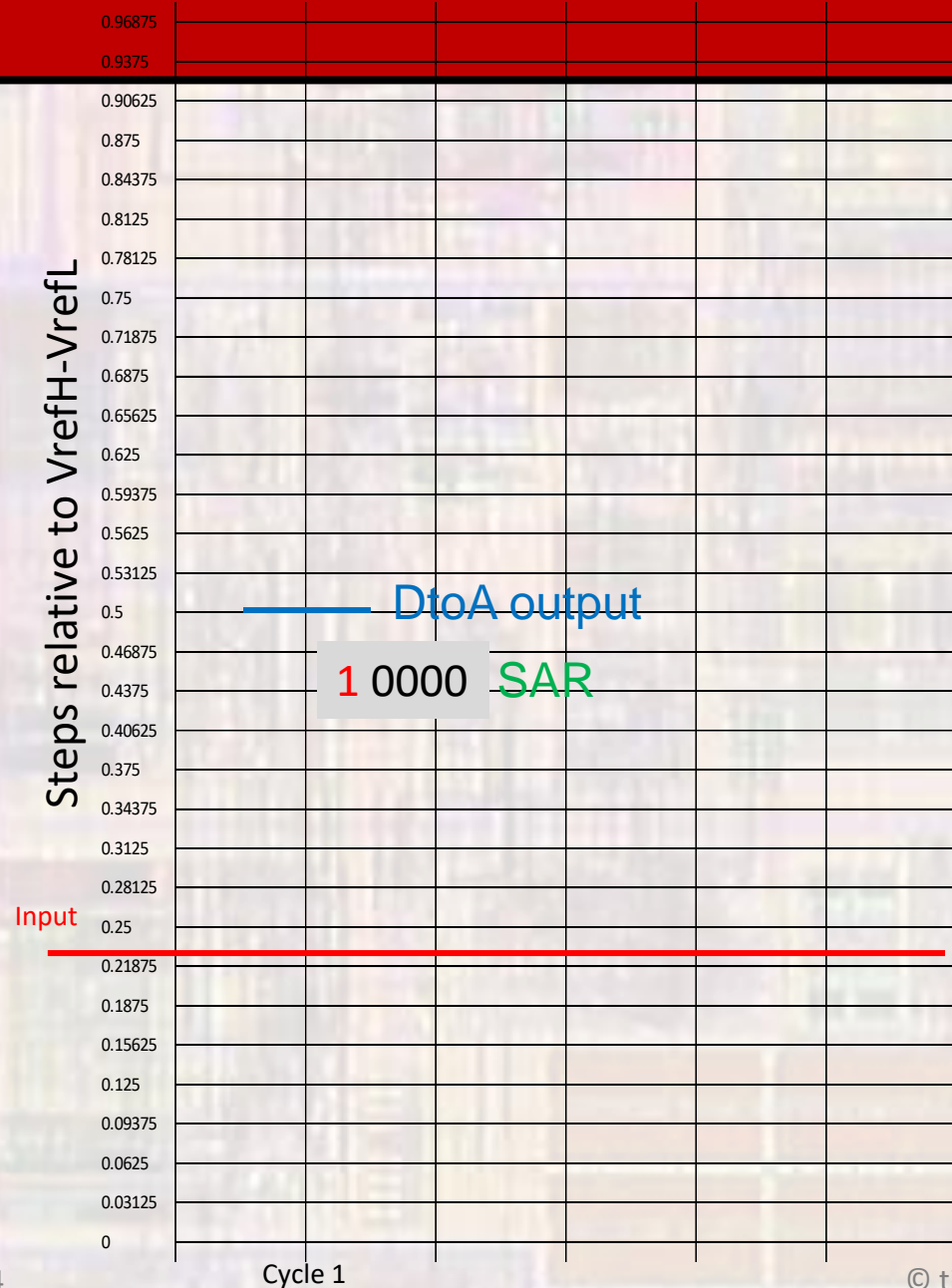
- A to D Converter

- 1V, 5 bit example

- Test to see if input is > or < midpoint

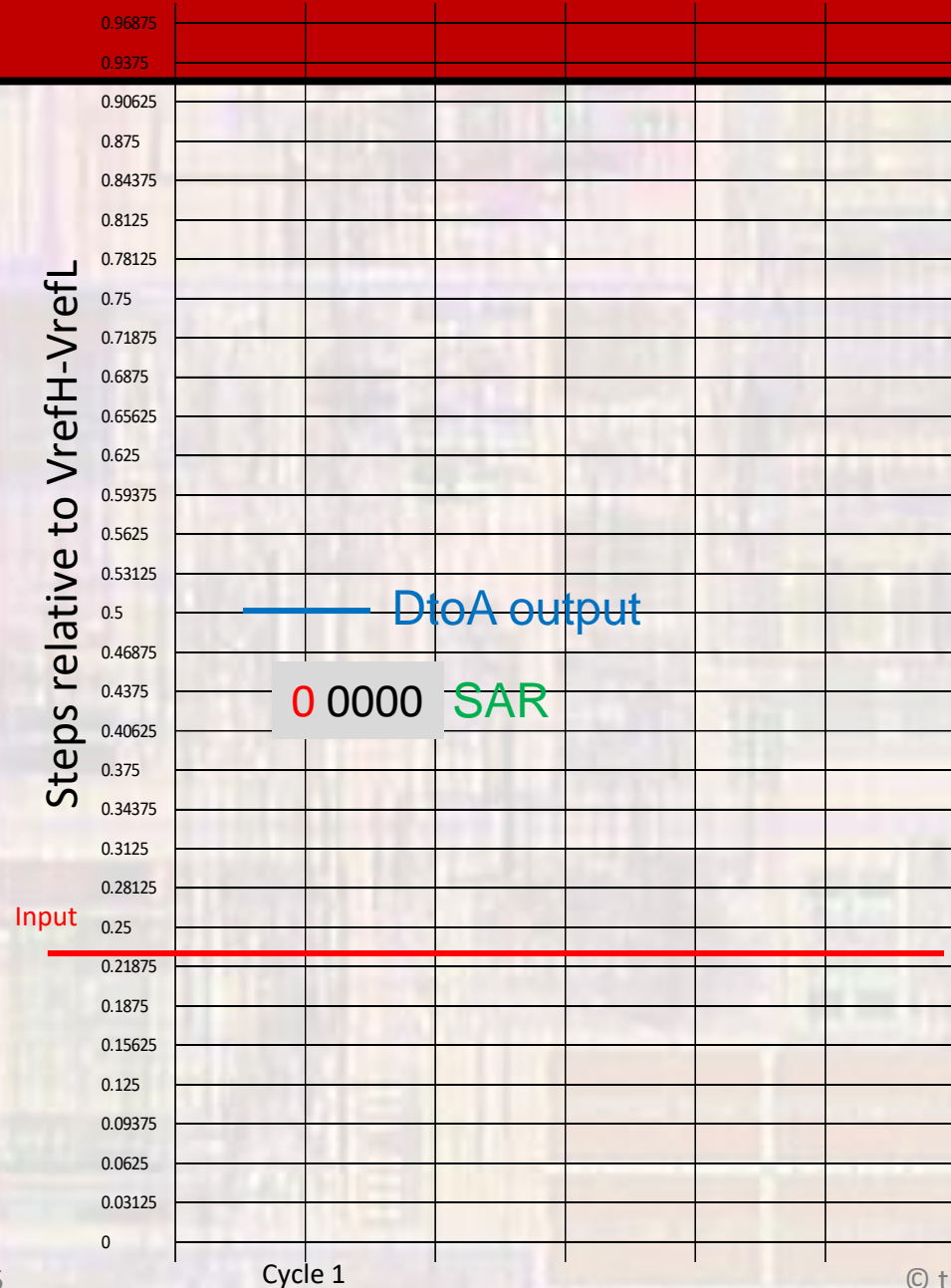
- if < , clear msb

- if > , set msb



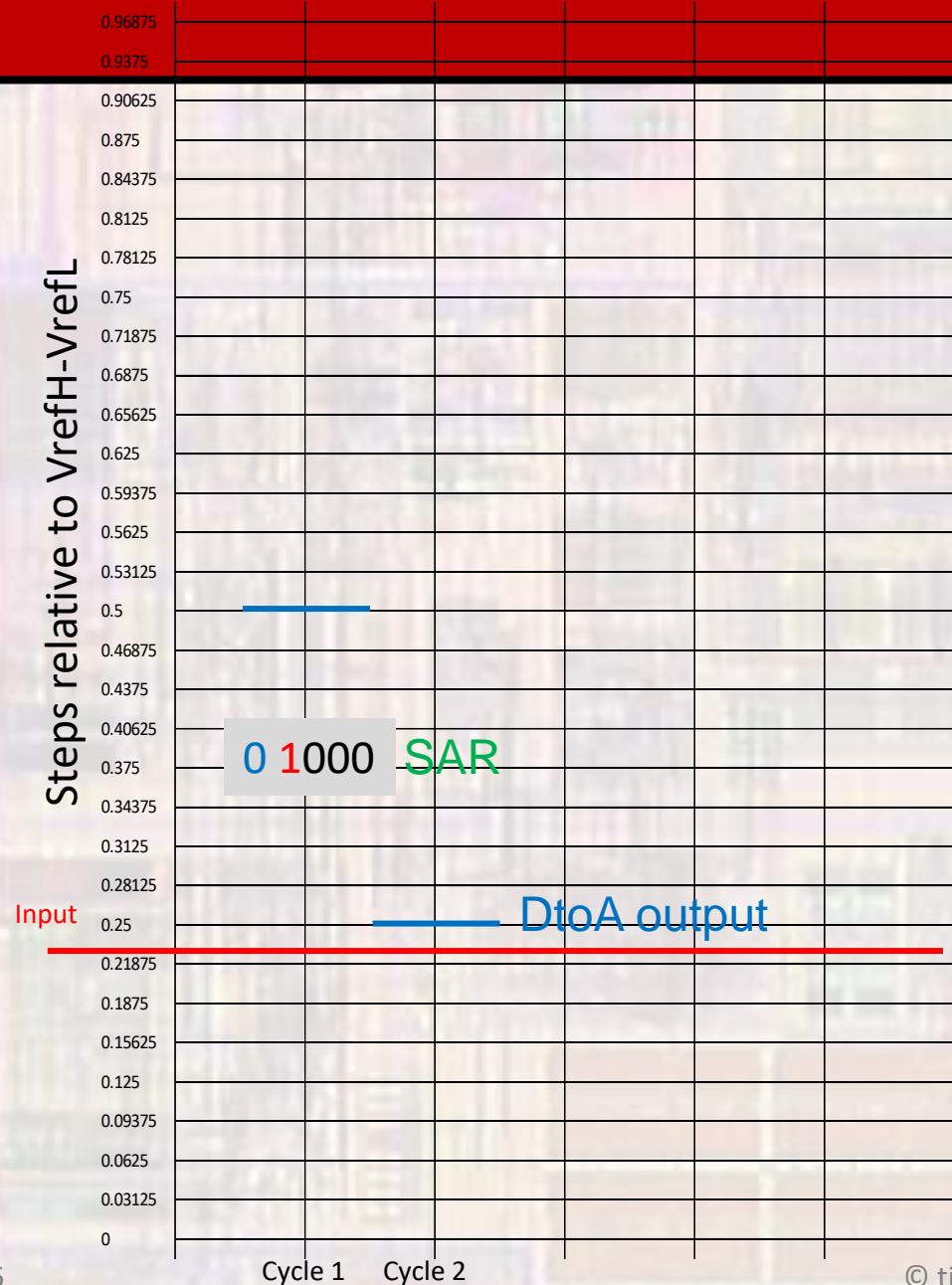
A/D

- A to D Converter
- Test to see if input is > or < midpoint
- if < , clear msb
- if > , set msb



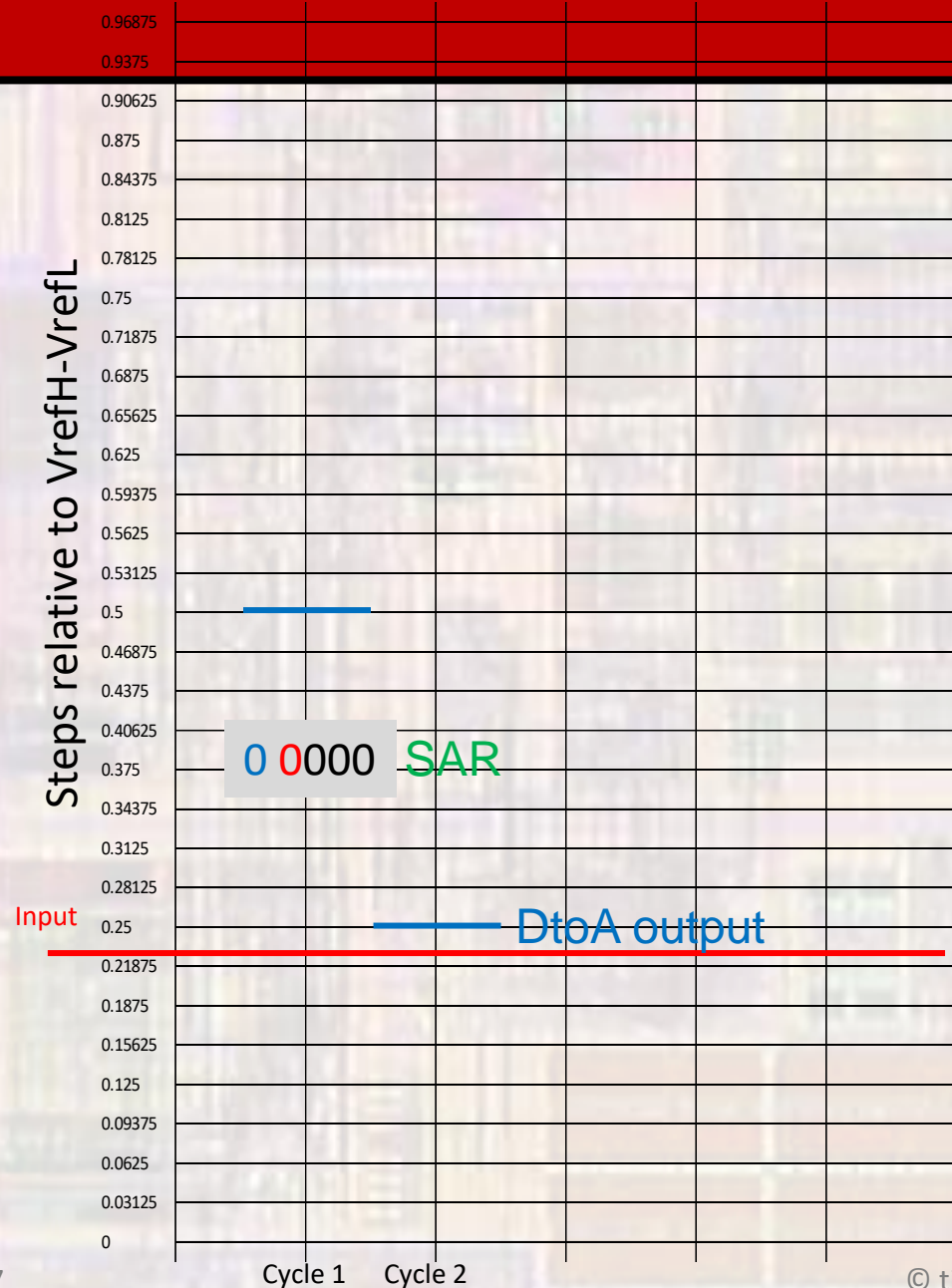
A/D

- A to D Converter
 - Test to see if input is > or < new “midpoint”
 - if < , clear bit
 - if > , set bit



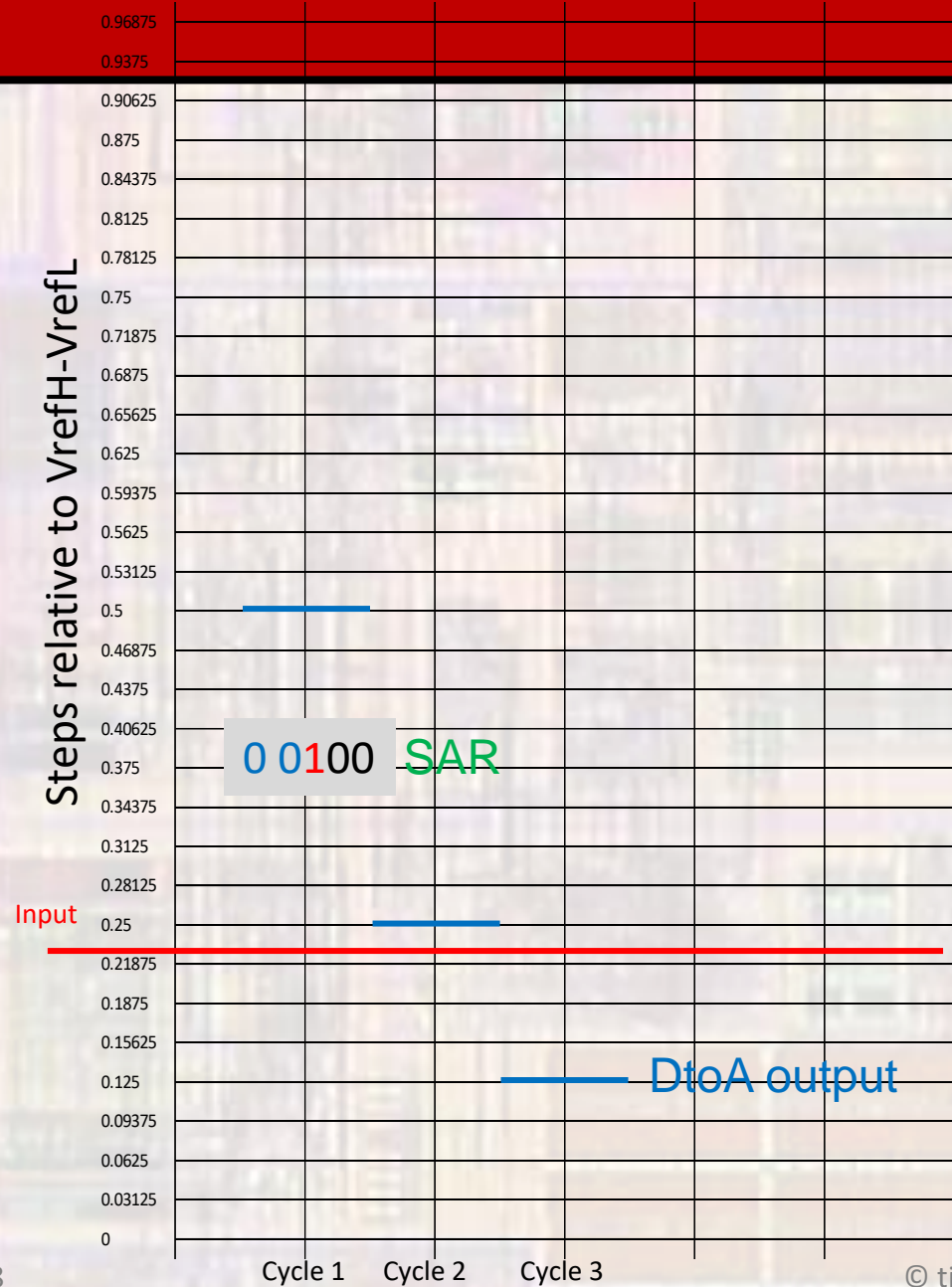
A/D

- A to D Converter
- Test to see if input is > or < new “midpoint”
- if < , clear bit
- if > , set bit



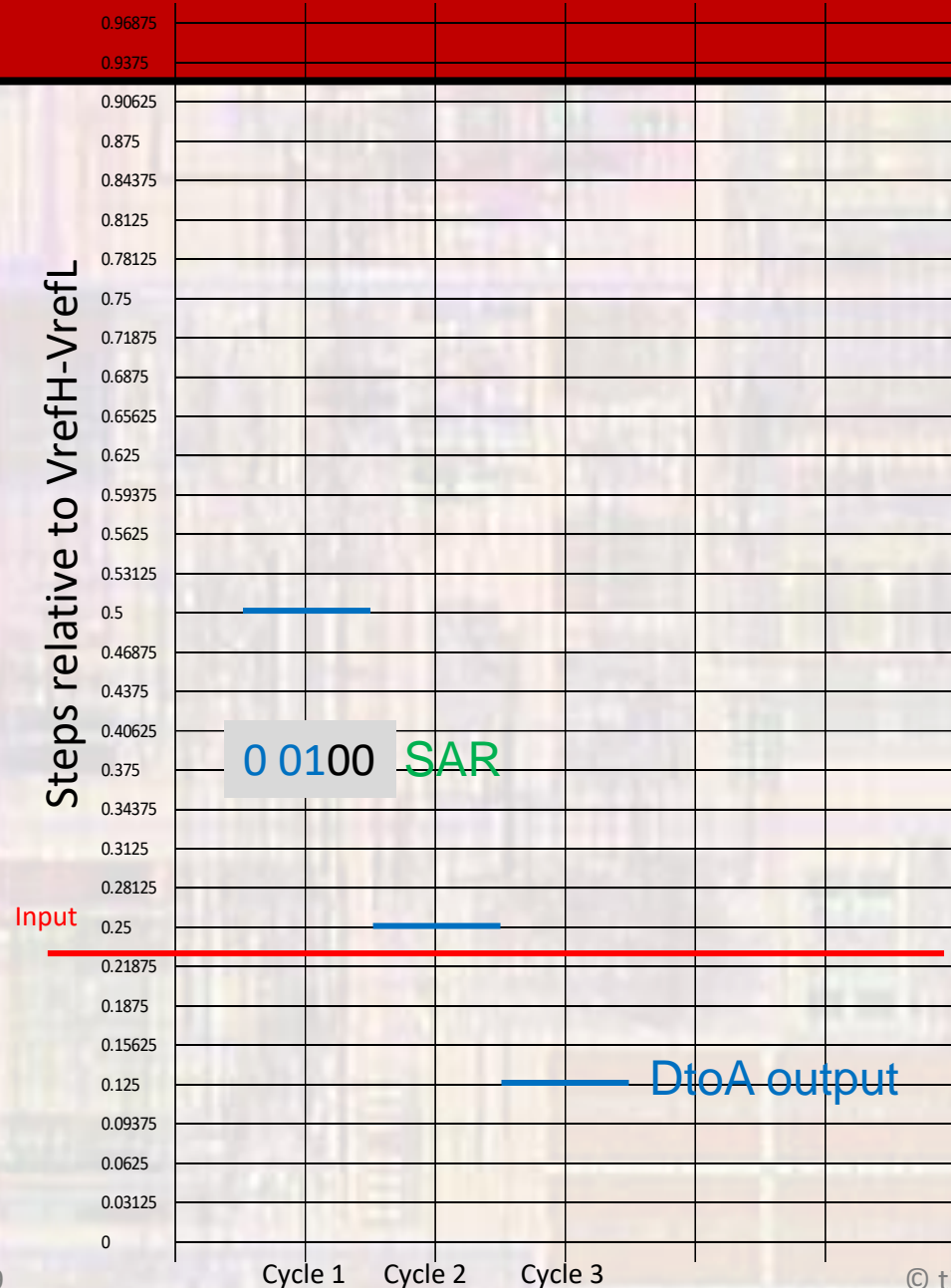
A/D

- A to D Converter
- Test to see if input is > or < new “midpoint”
 - if < , clear bit
 - if > , set bit



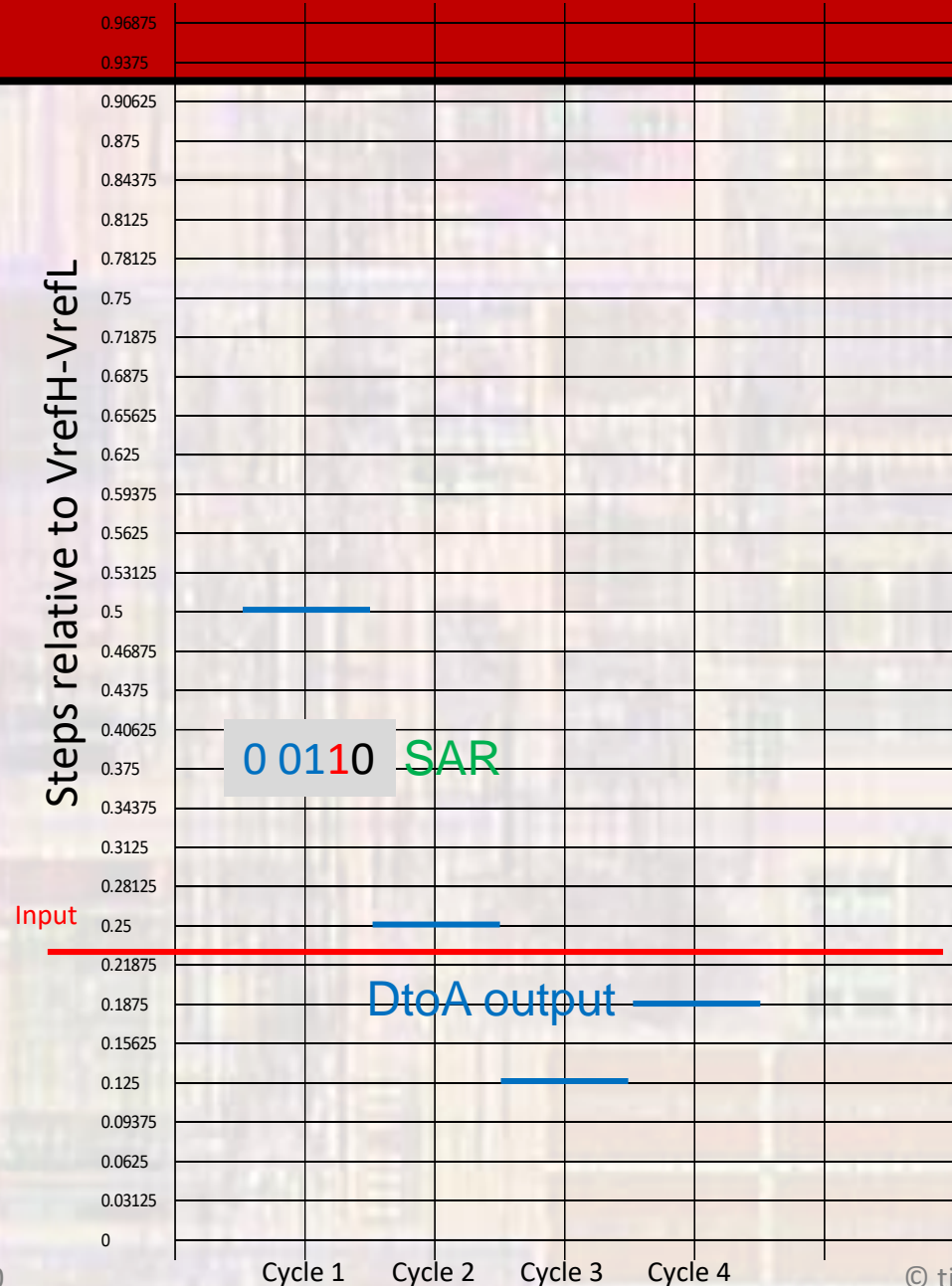
A/D

- A to D Converter
- Test to see if input is > or < new “midpoint”
 - if < , clear bit
 - if > , set bit



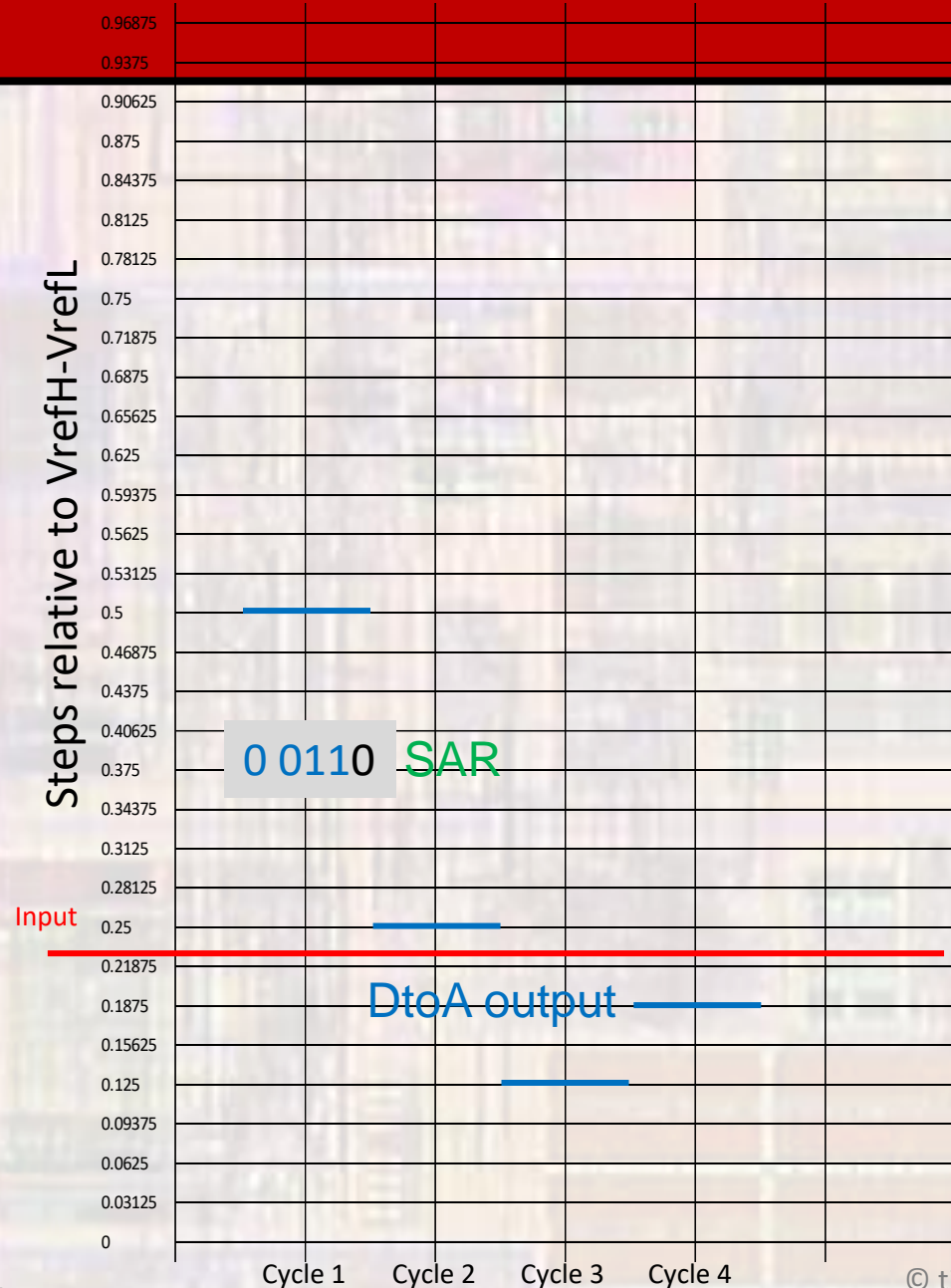
A/D

- A to D Converter
 - Test to see if input is > or < new “midpoint”
 - if < , clear bit
 - if > , set bit



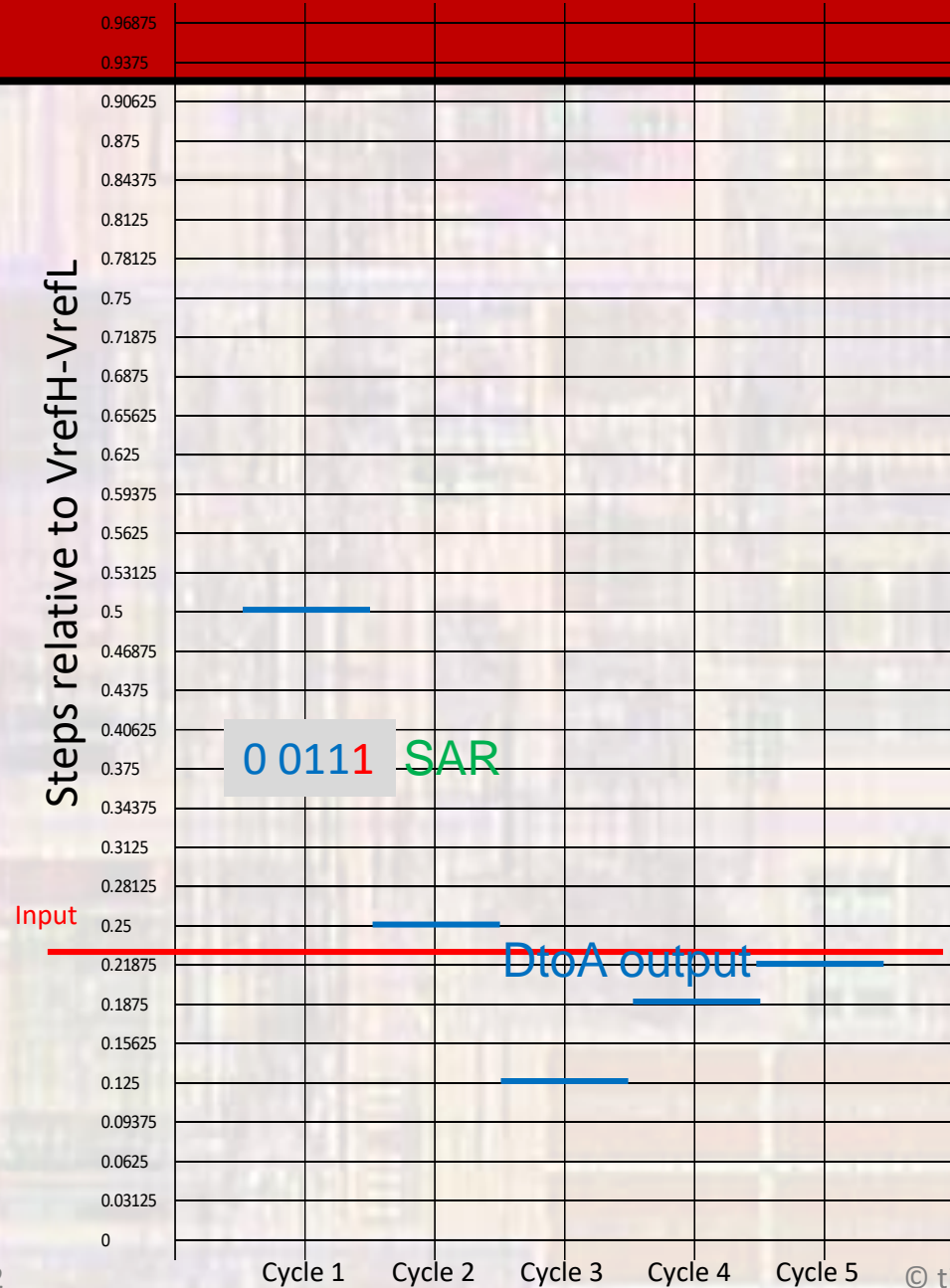
A/D

- A to D Converter
- Test to see if input is > or < new “midpoint”
 - if < , clear bit
 - if > , set bit



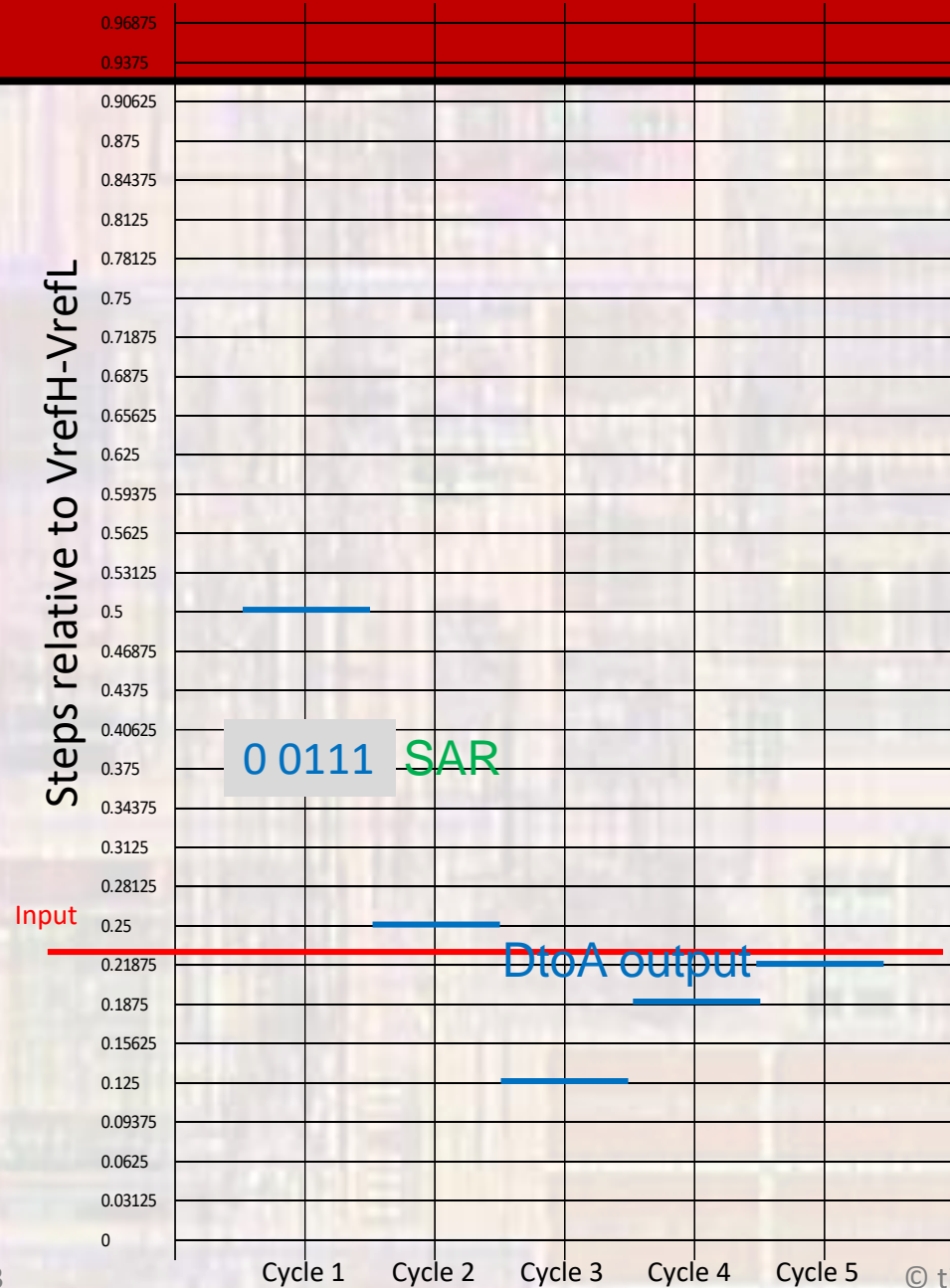
A/D

- A to D Converter
- Test to see if input is > or < new “midpoint”
 - if < , clear bit
 - if > , set bit



A/D

- A to D Converter
- Test to see if input is > or < new “midpoint”
 - if < , clear bit
 - if > , set bit



A/D

- Example 1

Nbits: 4
VrefH: 3v
VrefL: 0v

D/A Code	D/A Output Voltage
HEX	
0	0.000
1	0.188
4	0.750
6	1.125
8	1.500
C	2.250
E	2.625
F	2.813

A/D

- Example2

Nbits: 4
VrefH: 3v
VrefL: 0v

Input Voltage	Output Code
	Hex
0	0
0.3	1
0.9	4
1.47	7
1.53	8
2.4	C
2.85	F
3.3	F

A/D

- Example3

Nbits: 4
VrefH: 3v
VrefL: 0v

Input Voltage	Output Code	D/A Output Voltage
	Hex	
0	0	0.000
0.3	1	0.188
0.9	4	0.750
1.47	7	1.313
1.53	8	1.500
2.4	C	2.250
2.85	F	2.813
3.3	F	2.813

Error Voltage
0.000
0.113
0.150
0.158
0.030
0.150
0.038
0.488

Error Bits
0.000
0.600
0.800
0.840
0.160
0.800
0.200
2.600

A/D

- Add other types

Nbits: 4
VrefH: 3v
VrefL: 0v