ECEN 4013

Design of Engineering Systems

Agenda

Project 1

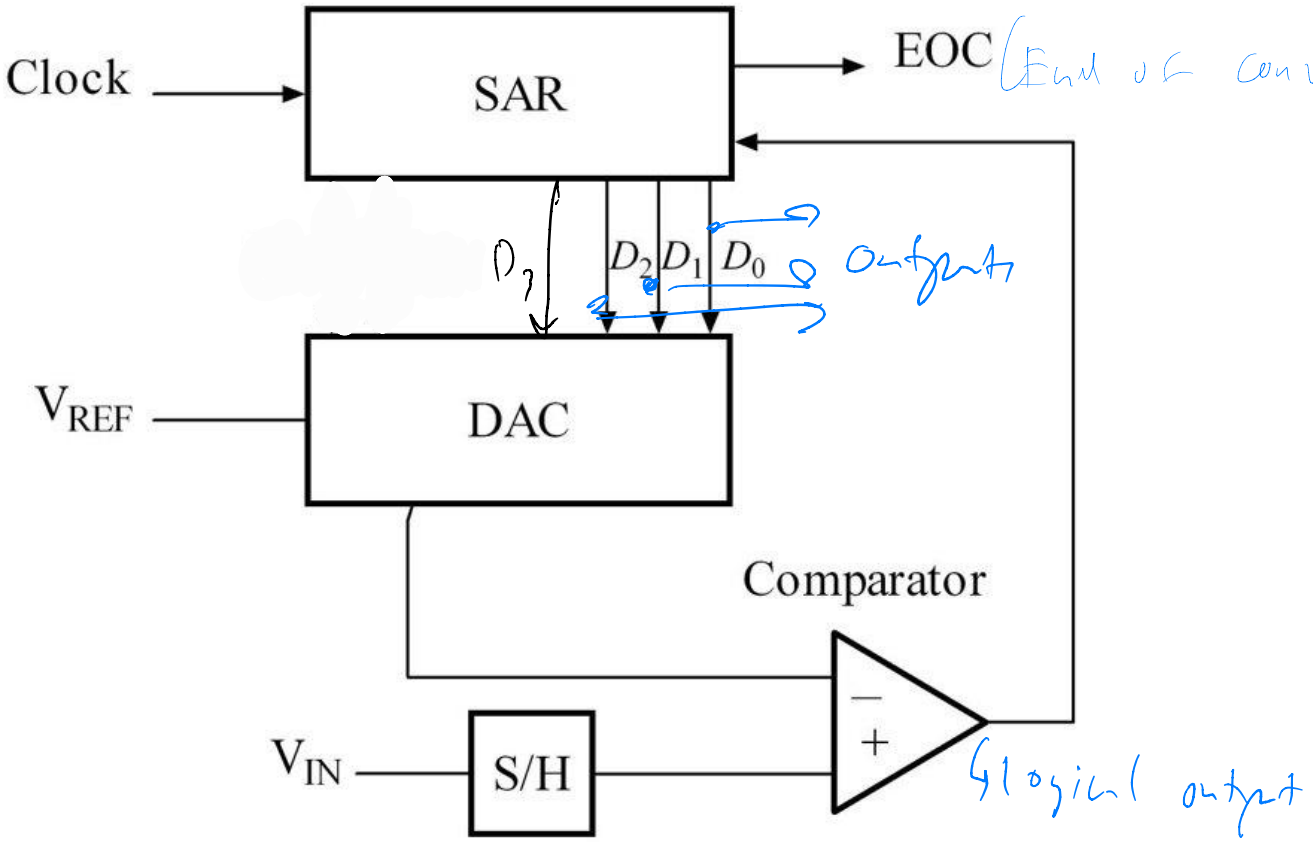
SAR

Sample and Hold

DACs

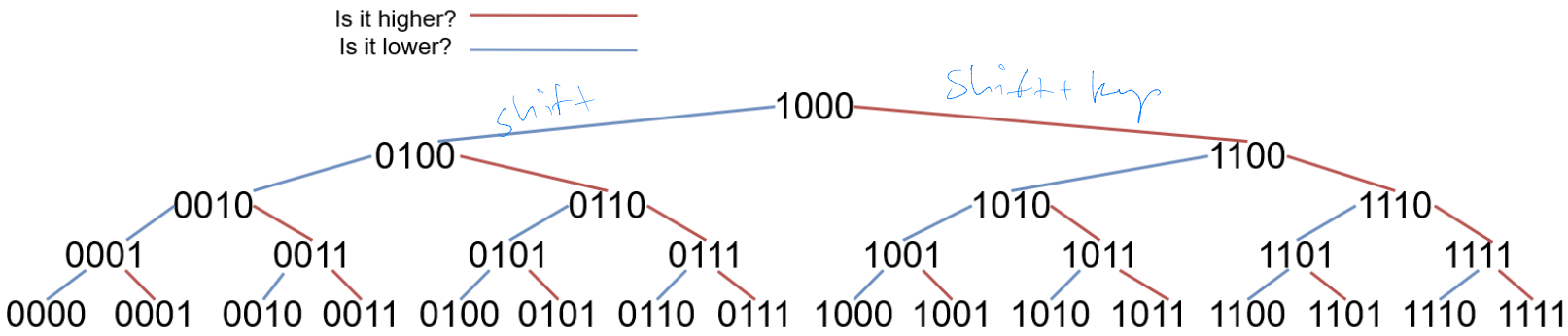
Project 1

Questions?

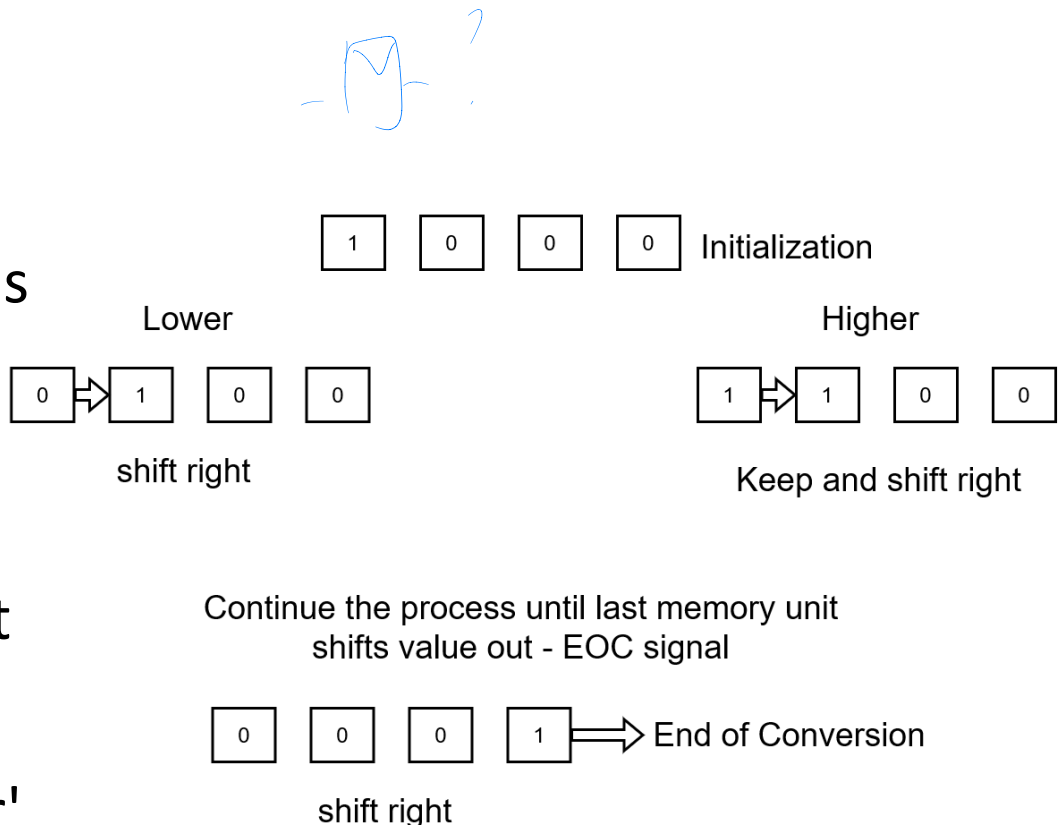

Clock
SAR
EOC
D_3
D_2
D_1
D_0
V_{REF}
DAC
Comparator
V_{IN}
S/H
−
+

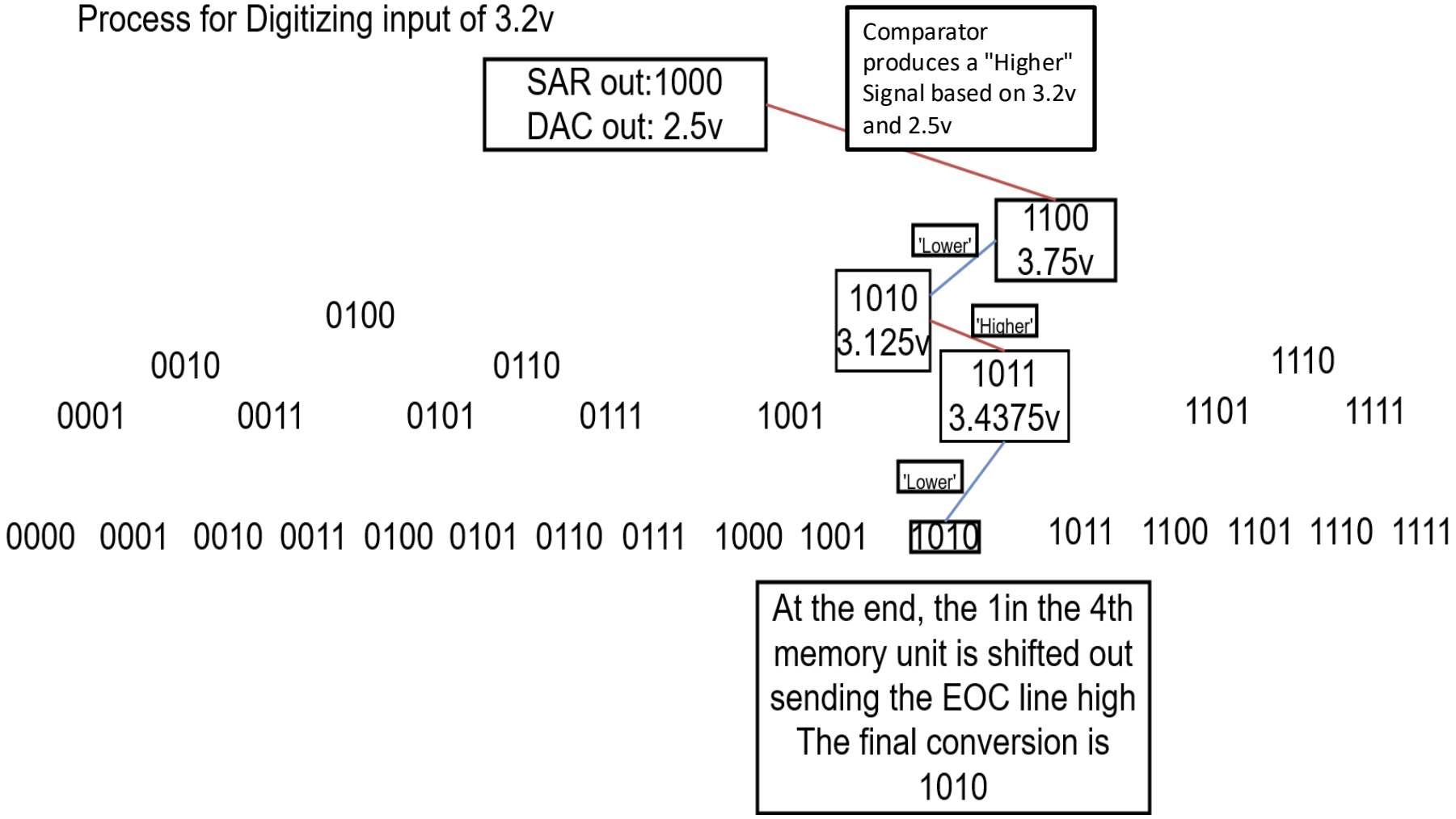
Basics of a Successive Approximation Registers (SAR)

- An SAR implements a binary search algorithm based on the data in from your comparator
- Binary search tree, splits your search space in half through each iteration. In this case it splits your search space in half based on a "higher" or "lower" signal from your comparator



- Think of the SAR as a memory device (flip flops, latches, shift registers, etc.) that holds the current value of your digital approximation.
- This memory device holds 4 logic values for 4 bits and initialized to 1000
- As the memory device progresses through the algorithm, the current bit value is shifted right and either held or reset depending on the 'higher' or 'lower' signal from the comparator.




Process for Digitizing input of 3.2v
SAR out:1000
DAC out: 2.5v
Comparator produces a "Higher" Signal based on 3.2v and 2.5v
1100
3.75v
'Lower'
1010
3.125v
'Higher'
1011
3.4375v
'Lower'
0100
0010
0110
1110
0001
0011
0101
0111
1001
1101
1111
0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
At the end, the 1in the 4th memory unit is shifted out sending the EOC line high The final conversion is 1010

Storage registers

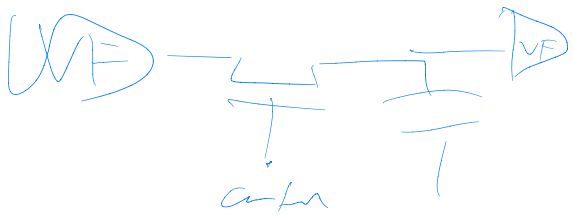
- Option 1 - use a COTS SAR chip:
  - MC14549, MC14559
  - Will need a surface mount to through-hole adapter
  - Try to get the one that can be modified to a 4-bit SAR
- Option 2 - Shift registers:
  - 74LS194
- Option 3 - flipflops or latches:
  - 74HC107, CD4013, CD4043
  - https://www.enggjournals.com/ijet/docs/IJET16-08-02-073.pdf

The idea is that you need to store the current digitized value while making a comparison of the DAC output and voltage to be digitized

Control and sequence logic

Algorithm:

0. Initialize memory device with 1000
   - DAC produces voltage Vdac
1. Sample and hold Vin
2. Compare Vin to Vdac
   - If Vin > Vdac
     - Store 1 at current bit and add 1 to next most significant bit
   - If Vin < Vdac
     - Store 0 at current bit and add 1 to next most significant bit
3. Recurse (go back to step 2)



End condition – last bit if Vin > Vdac store 1 else store 0.

Control and sequence logic

What needs to be controlled

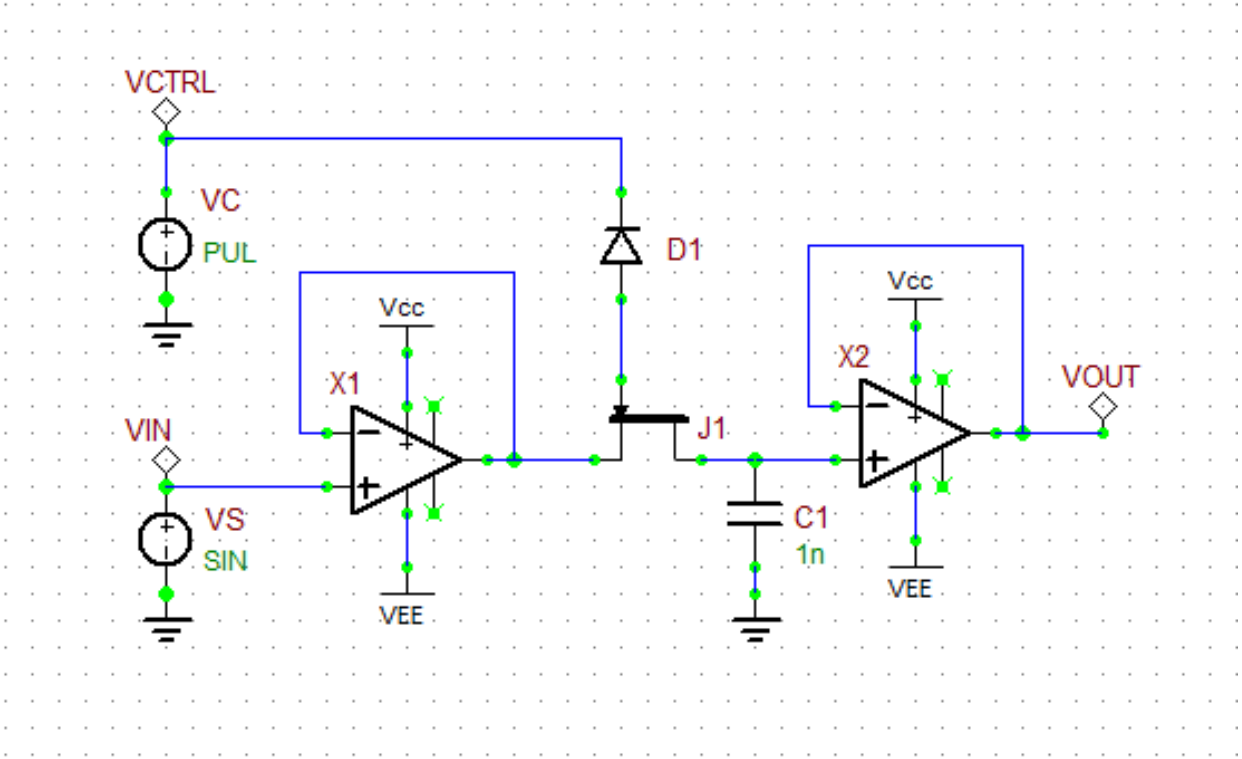
- Sample and Hold
- Memory devices
  - Initialization
  - Storing current digital value and output final value when conversion is complete

Implementation ideas:

- Option 1 - use a COTS SAR chip
- Option 2 - use a small form factor computer
  - Arduino, raspberry pi, beagle bone, nvidia jetson
- Use shift register in combination with output storage device
  - When shifted through all of the bits trigger sample and hold and EOC
- Devise a finite state machine for all states in circuit and use common logic to implement it
- Cascade flipflops together to pass bit values like a shift register and pass to output flipflops

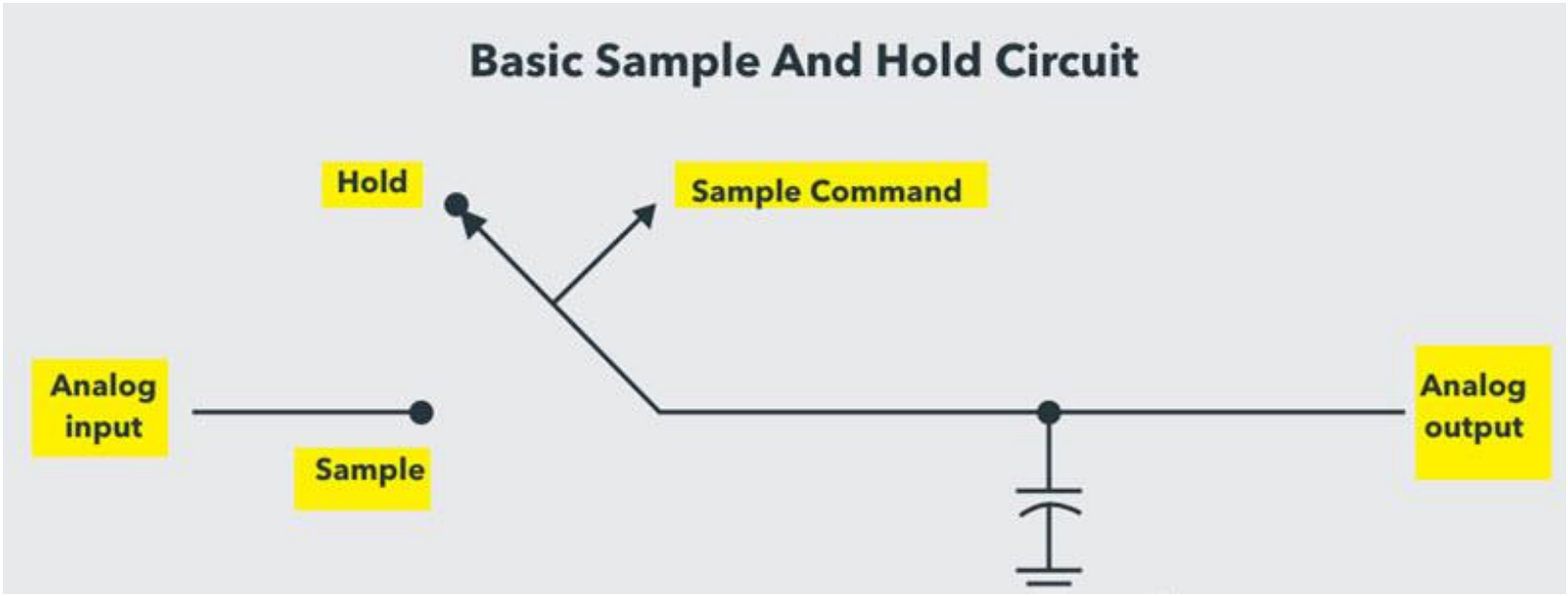
Sample and Hold Circuit

Theoretically our input to the ADC will be steady. In practice this will not be true. If the voltage varies more than ½ LSB you may get code errors. Using a sample and hold circuit can remedy this situation.



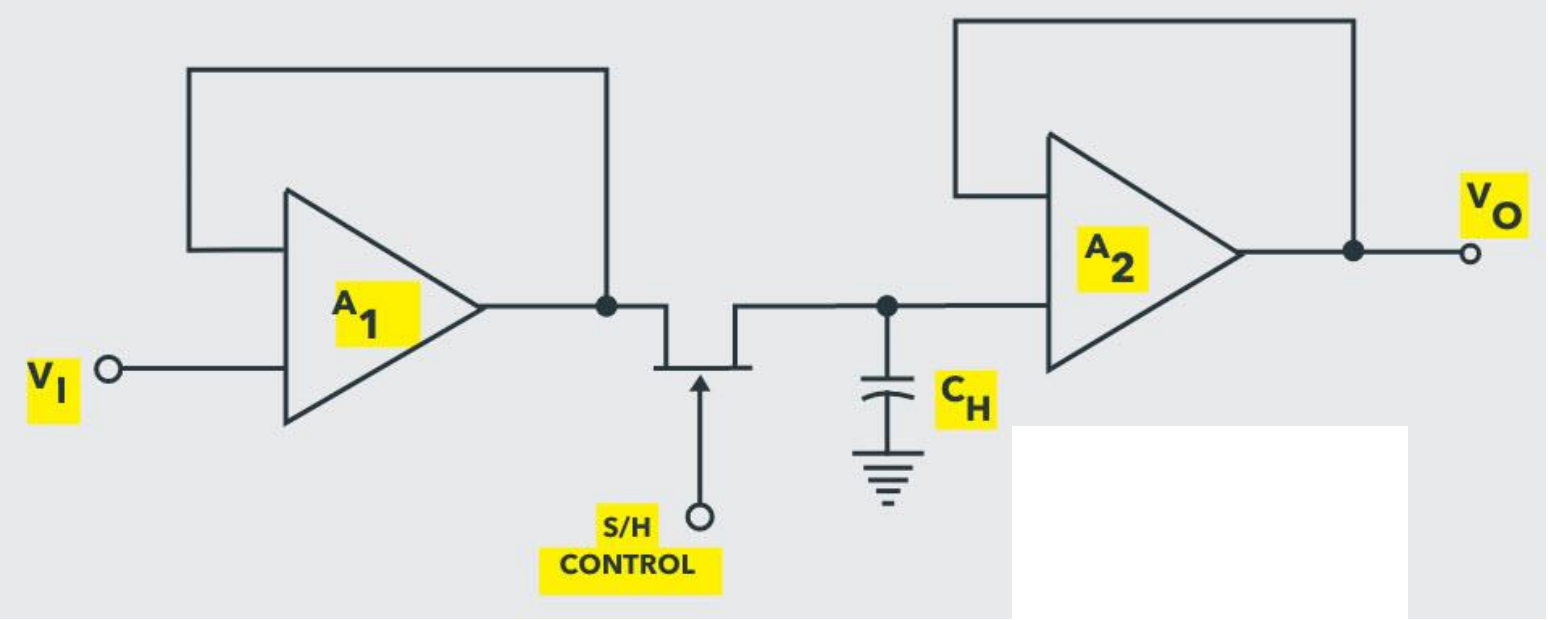
Sample and Hold Circuit

In its most basic form, a sample and hold circuit is a switch with a holding capacitor.



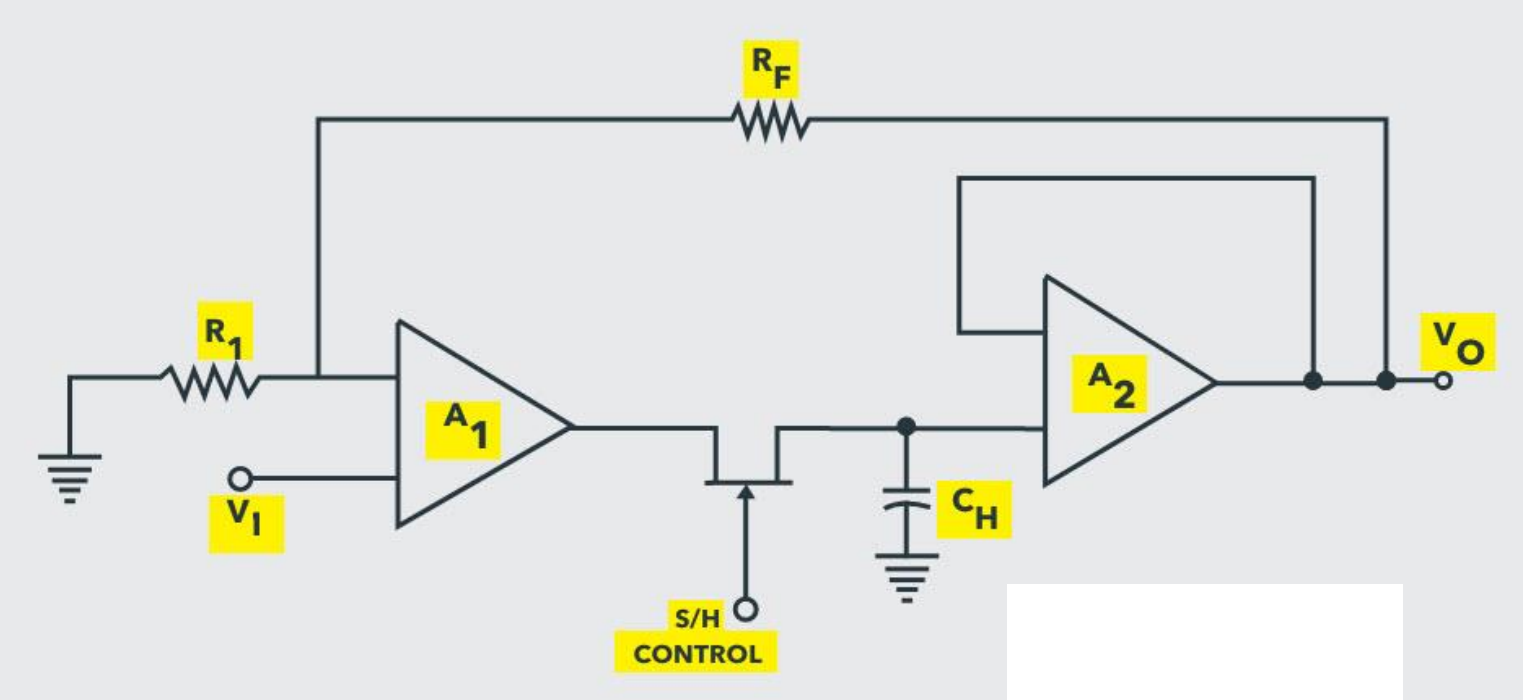
Sample and Hold Circuit

Any FET can act as an analog switch. Generally, two voltage followers are used as buffers to keep voltage stored in capacitor. Circuits with no feedback are fast but suffer in accuracy:



Sample and Hold Circuit

Feedback can be used in many forms to improve the circuit, including adding gain: 1+(Rf/R1)



Sample and Hold Circuit

Issues to consider:

- Acquisition Time (t_{ac})
  - Time for capacitor to rise during sampling, largely affected by RC time constant
- Aperture Time (t_{ap})
  - Time delay caused by propagation delays through the driver and the switch circuits
- Aperture Uncertainty (Δt_{ap})
  - Aperture time will vary from sample to sample
- Hold Mode Settling Time (t_s)
  - The hold mode settling time is the time taken by the output V_o to settle within the specified error band (usually 1%, 0.1% or 0.01%) after the application of hold command.
- Parasitic capacitances
  - Can affect the output voltage in hold mode
- Droop
  - Voltage Droop is a phenomenon where the voltage across the holding capacitor drops down due to leakage currents.

DACs

Types:

- PWM
  - Apply a LPF to remove high frequency components of digital signal to arrive at an equivalent DC value. Applicable for periodic digital signal
- Delta-Sigma Modulation
  - output is a stream of pulses the density of which corresponds to digital number input. This stream is then filtered as in PWM
- Cyclic/Successive Approximation
  - Processes Digital signal serially one bit at a time to arrive at an analog output like SAR AD in reverse

DACs

Types:

- Thermometer coded
  - Uses Unary coding and an equal resistor segment for each possible digital value. 4bits – 16 resistor segments

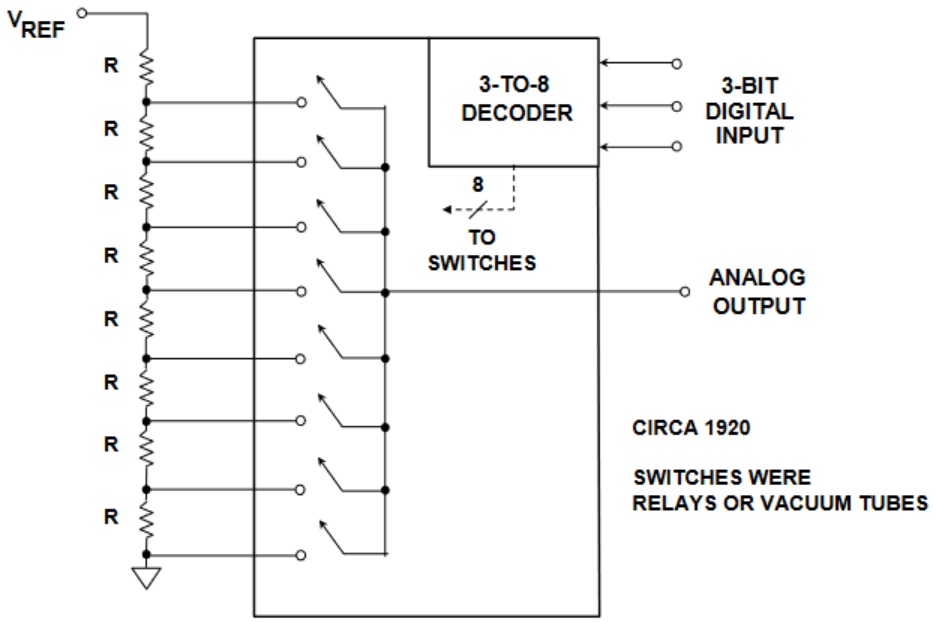


Figure 2: Simplest Voltage-Output Thermometer DAC: The Kelvin Divider ("String DAC")

DACs

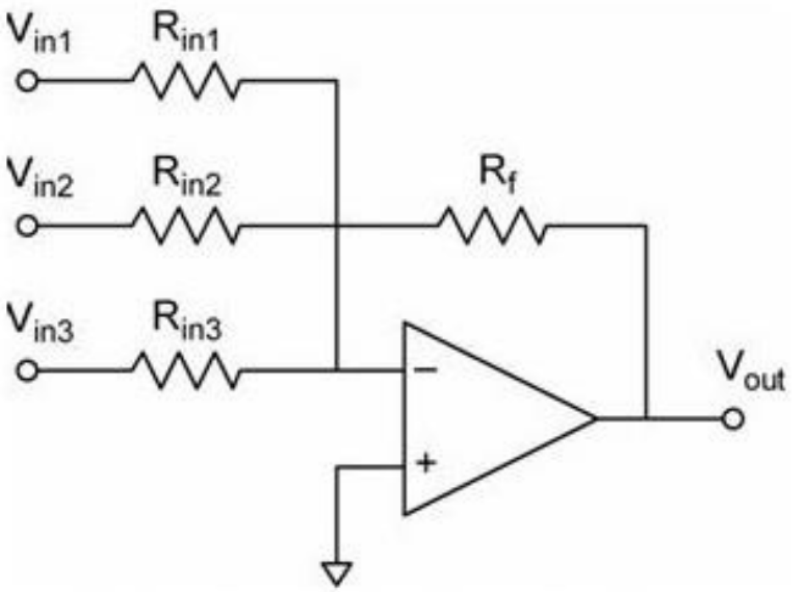
Types:

- Binary Weighted Resistor (can also be done with current sources and capacitors)

Let's consider a 3 bit DAC that has a voltage range of 0 to 1v

- LSB resolution - (1v-0v)/2^3 = 1/8v
- Each digital step represents 1/8v

A binary weighted resistor is essentially an inverting summing amplifier



DACs

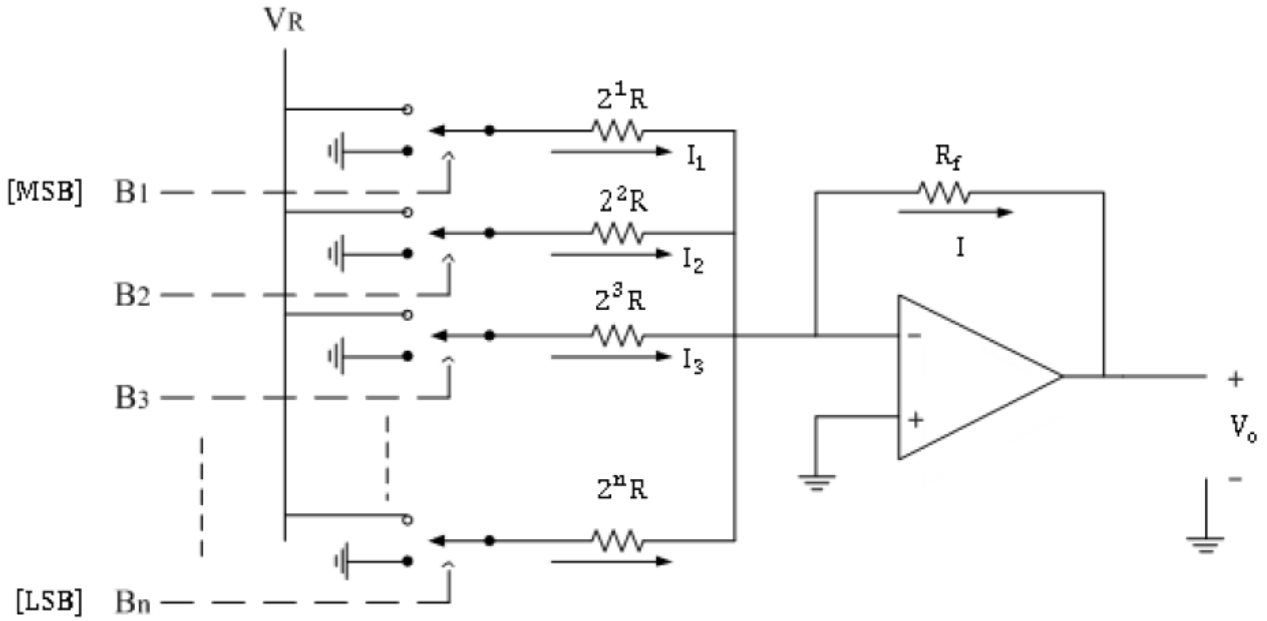
The inverting summing amplifier has a well-known equation for Vout (it is a proportional sum of input voltages):

- Vout = -[(Rf/R1)V1 + (Rf/R2)V2 + (Rf/R3)V3 + …]

If we tie all of the voltages to our reference voltage for our DAC range and make the resistor values scaled by a power of 2 based on the bit they are controlled by we have:

- Vout = -[(Rf/2R)Vr*B1 + (Rf/4R)Vr*B2 + (Rf/8R)Vr*B3 + …] or
- Vout = -Rf*Vr(B1/2R + B2/4R + B3/8R + …)

Where B1 is the most significant bit and each Bn can either be 1 or 0 depending what position the switch is in



DACs

If we let Rf = R we have

- Vout = -R*Vr(B1/2R + B2/4R + B3/8R + …) or
- Vout = -Vr(B1/2 +B2/4 +B3/8 + …)

This makes sense since the MSB is esentially worth ½ the voltage range, B2 is worth ¼ of the range, B3 is worth 1/8 …

Going back to our example where our range is 1v represented by 3 bits we would have the following voltages for each digital representation (because LSB = 1/8v):

000 = 0v	100 = .5v
001 = .125v	101 = .625v
010 = .25v	110 = .75v
011 = .375v	111 = .875v

DACs

If we supply our circuit with –1v for our Vr we can compare what we get from the summing amplifier with what we expect

- Vout = -Vr(B1/2 +B2/4 +B3/8)

000 => Vout = 1(0/2+0/4+0/8) = 0v

001 => Vout = 1(0/2+0/4+1/8) = .125v

010 => Vout = 1(0/2+1/4+0/8) = .25v

011 => Vout = 1(0/2+1/4+1/8) = .375v

100 => Vout = 1(1/2+0/4+0/8) = .5v

101 => Vout = 1(1/2+0/4+1/8) = .625v

110 => Vout = 1(1/2+1/4+0/8) = .75v

111 => Vout = 1(1/2+1/4+1/8) = .875v

DACs

Which is what we expect to get.

You can start with R for the MSB instead of 2R, and then Rref would need to be R/2. This is essentially the same thing, just a matter of how you look at it.

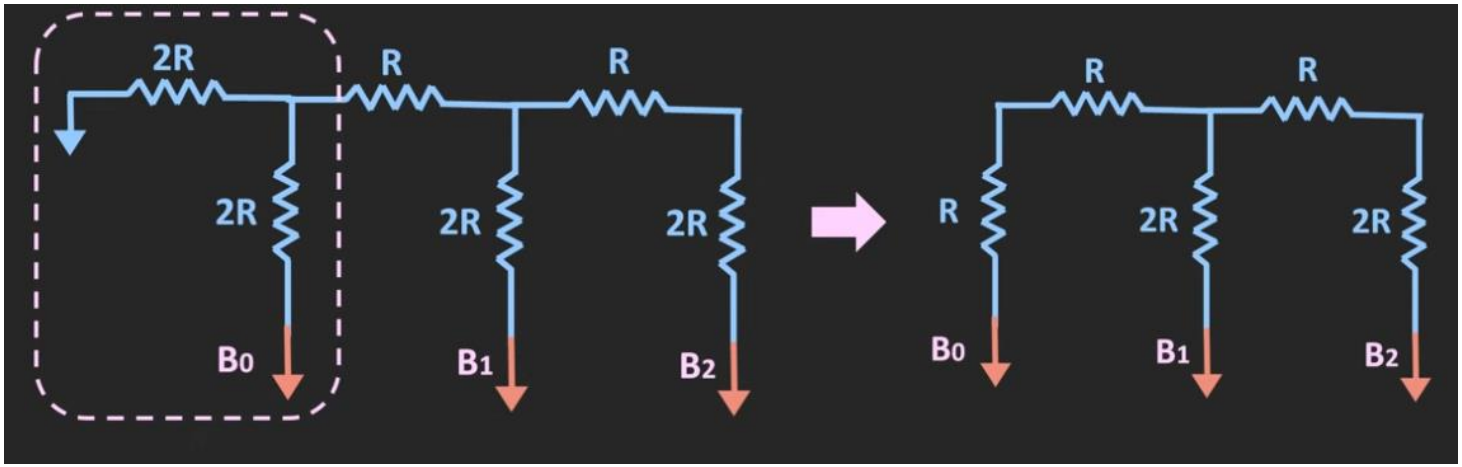
You can also build a non-inverting summing amp and supply the circuit with 1v at the reference voltage instead of –1v. It is quite common for ADCs and DACs to require –Vref to define your voltage range, however.

What happens if we increase our bit resolution to 16 bits?

DACs

Types:

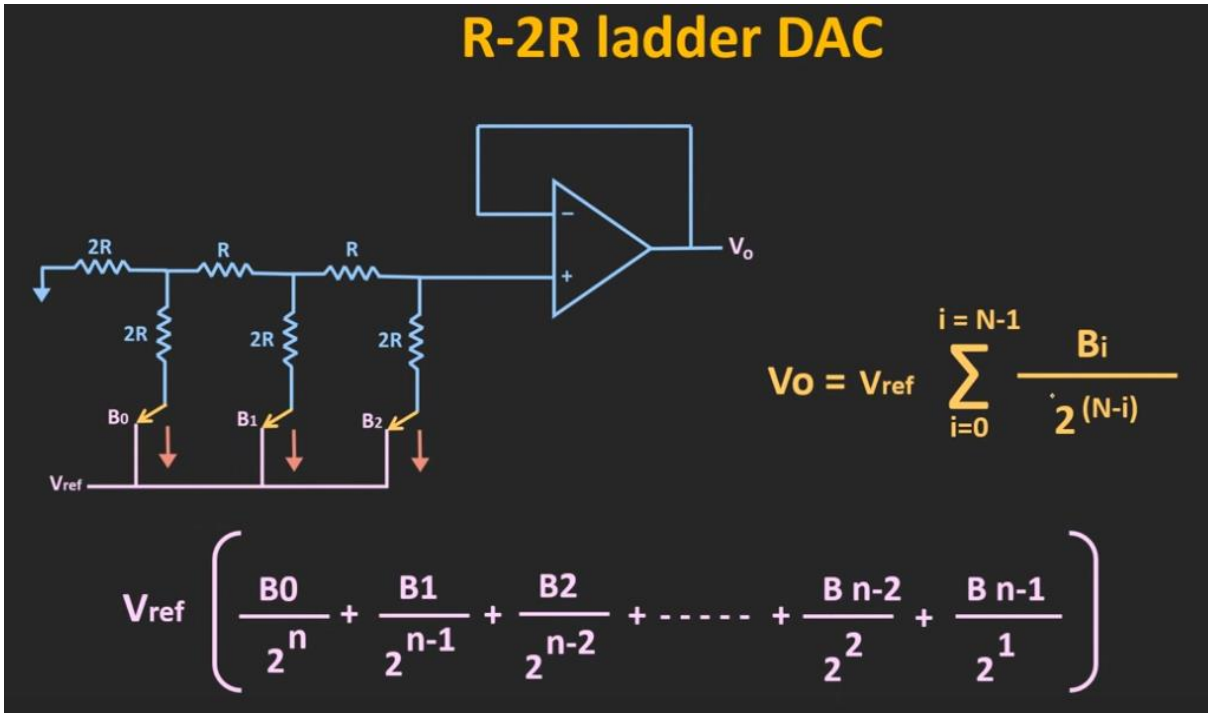
- R-2R Ladder
  - No matter how many bits equivalent resistance is R and only need 2 resistor values



DACs

Types:

- R-2R Ladder
  - Each bit is weighted by 1/2n like in the binary weighted resistor DAC (use Thevanin equivalent circuits to discover this)



DACs

Switching mechanisms:

- Connect resistors directly to the outputs of an SAR chip like the MC14559
- Voltage controlled switches
- Transistor switching circuits

Regardless, the output of your SAR will drive this mechanism