Digital Ramp ADC

Introduction

One of the simplest analog-to-digital conversion (ADC) methods to breadboard is called a digital ramp ADC. A variation of this ADC design is called a tracking ADC. In this lab you will build, measure and compare the important characteristics of these ADCs.

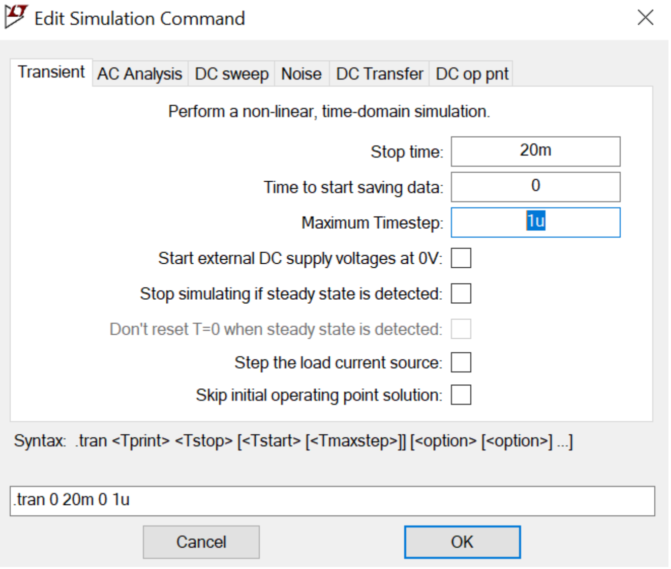
Objectives

- Use a simulated 555 timer as a clock
- Build a Shift Register in LTSpice
- Build two different ADC types in LTSpice
- Measure and compare each ADC type

Note Regarding Simulation Time

When you run a transient analysis in LTSpice you can specify a length of time to run the simulation for T (T="Stop Time" - "Time To Start Saving Data"). This results in a simulation with a number of data points N proportional to the Maximum Timestep setting (N=T/"Maximum Timestep").

If you leave your Maximum Timestep small and increase your sim time, your simulation will take a long time to process. If you make your timestep too big you won't have a fine enough resolution to get the results you want. I recommend using around 20k data points for this lab.



Perform a non-linear, time-domain simulation.

Stop time: 20m

Time to start saving data: 0

Maximum Timestep: 1u

Start external DC supply voltages at 0V: ☐

Stop simulating if steady state is detected: ☐

Don't reset T=0 when steady state is detected: ☐

Step the load current source: ☐

Skip initial operating point solution: ☐

Syntax: .tran <Tprint> <Tstop> [<Tstart> [<Tmaxstep>]] [<option> [<option>] ...]

```
.tran 0 20m 0 1u
```

Figure 1: Example Transient Analysis with 20k Points

Procedure Part 1: 555 Timer

Remove the voltage source you had been using to generate a clock. Instead configure a 555 timer to create a square wave with a frequency between 5kHz and 10kHz. This will be the counter clock input. The LTSpice 555 timer is under misc in the components library. Figure 2 shows a setup for a 1kHz square wave, what values do you need to change to make a faster clk? You can look this up or experiment on your own.

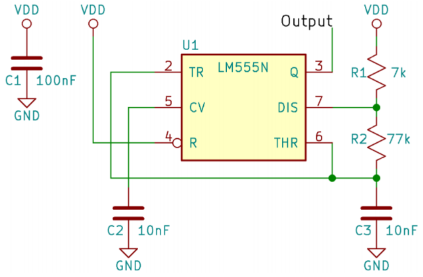


figure 2: 555 Timer Reference Schematic, setup for 1kHz

Include in your report

- How did you change your clk frequency?
- Picture of your 555 timer schematic

Procedure Part 2: 4 bit PIPO Shift Register

You need a 4 bit shift register to latch data in your ADC. The shift register should accept 4 bits in parallel, then load all those bits to the output on a rising edge (Parallel In Parallel Out). Build your shift register using 4 DFLOP components, found under Digital in the LTSpice component library. Figure 3 shows a 2 bit PIPO.

Components from the Digital Library have a default output voltage (1V) and trip point (500mV). You can change these by setting parameters for Vhigh, Vlow and Ref. Example: Vhigh=5 Vlow=0 Ref=1.5. You can assign these values to a component by right clicking on the component and entering them next to “Value” in the window that pops up.

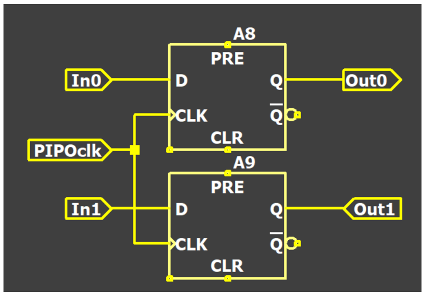


Figure 3: 2 Bit PIPO

Include in your report:

- Values you chose for Vhigh, Vlow and Ref
- Picture of your PIPO

Procedure Part 3: Digital Ramp ADC

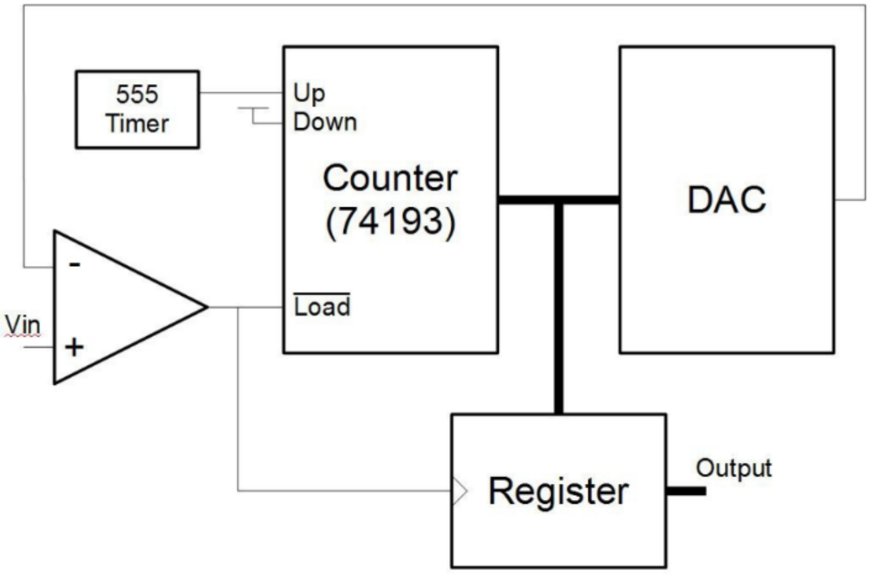


figure4: Block Diagram of Digital Ramp ADC

You’re going to implement the ADC shown in figure 4 in LTSpice. Start with one of the DACs you already built. You need to make the DAC output positive. You can do this by modifying your final gain stage or adding another inverting op amp at the output.

Building from this DAC, add the comparator block (figure 5).Next add the PIPO Shift Register you built. Add a voltage source for Vin.

Note: You may find a race condition where your comparator output does not trigger your PIPO before the input data has changed to all 0’s. In this case add a separate comparator that inverts the pulse as the input for your PIPO clk. You invert the pulse by swapping the inputs to your comparator.

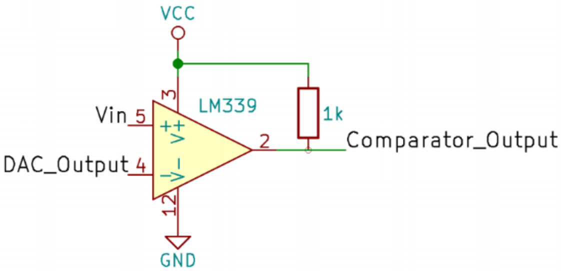


figure 5: Comparator Reference Schematic

A good way to show your digital results is to display them as a math function that gives weight to each bit. Example: ADC0 + 2* ADC1 + 4* ADC2 + 8*ADC3. You can divide this by the value of VDD to get a result as a pure number. On the Y Axis. Figure 6 shows an example of the expected output when you use a sine wave for Vin.

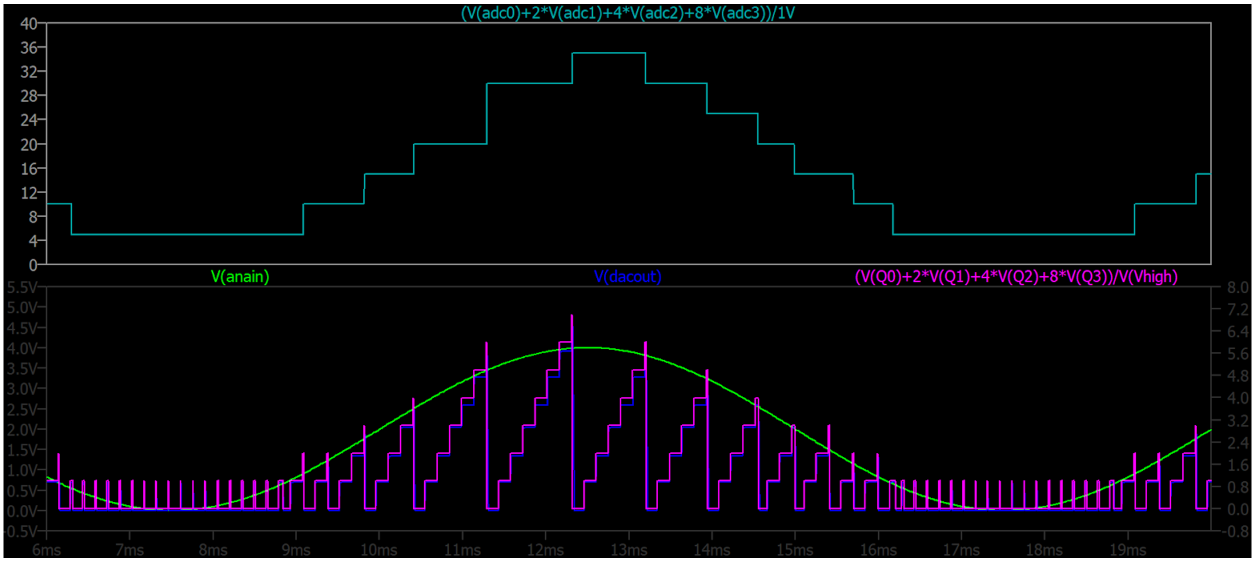


Figure 6: Expected Output for a Sine Wave

Include in your report

- Screen shot of ADC result when Vin is set to a DC value
- Screen shot of ADC result when Vin is set to a sine wave
- What variables determine the step size?
- How difficult would it be to expand the resolution of this DAC?

Procedure Part 4 Tracking ADC

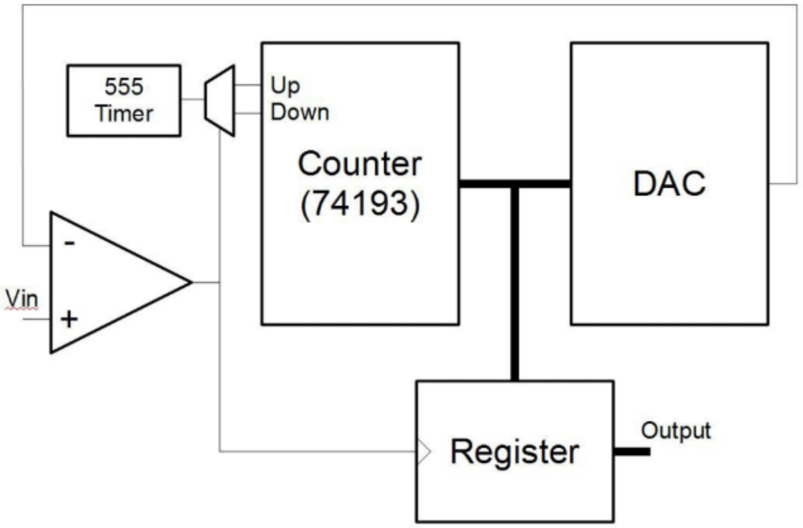


Figure 7: Tracking ADC

Convert your digital ramp ADC into the tracking ADC shown in Figure 7. FIgure 8 shows one way to implement the MUX you need for the counter clocks. Don't forget to change Vhigh, Vlow and Ref same as you did for the PIPO.

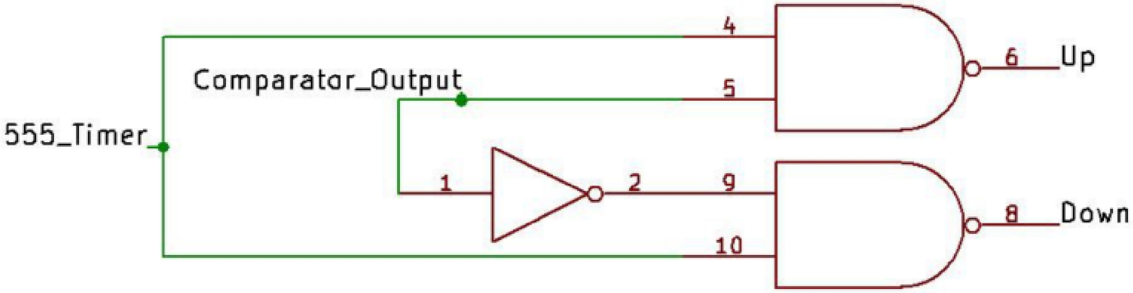


Figure 8: Example of a MUX

Include in your report

- Screen shot of ADC result when Vin is set to a DC value
- Screen shot of ADC result when Vin is set to a sine wave
- What variables determine the step size?
- How difficult would it be to expand the resolution of this DAC?