Lab 5: Analog to Digital Conversion

5.1 Introduction

In this lab, we will build an 8-bit ADC using the successive approximation technique. The overall scheme is shown in Fig. 1. The input is compared (using the LM311) to the most recent digital estimate, which is converted to analog using the AD557 digital to analog (D/A) converter (DAC). The 74LS503 successive approximation register (SAR) carries out the binary-search algorithm and performs all necessary control functions. At the end of the conversion process, the result is latched by the 574 8-bit register.

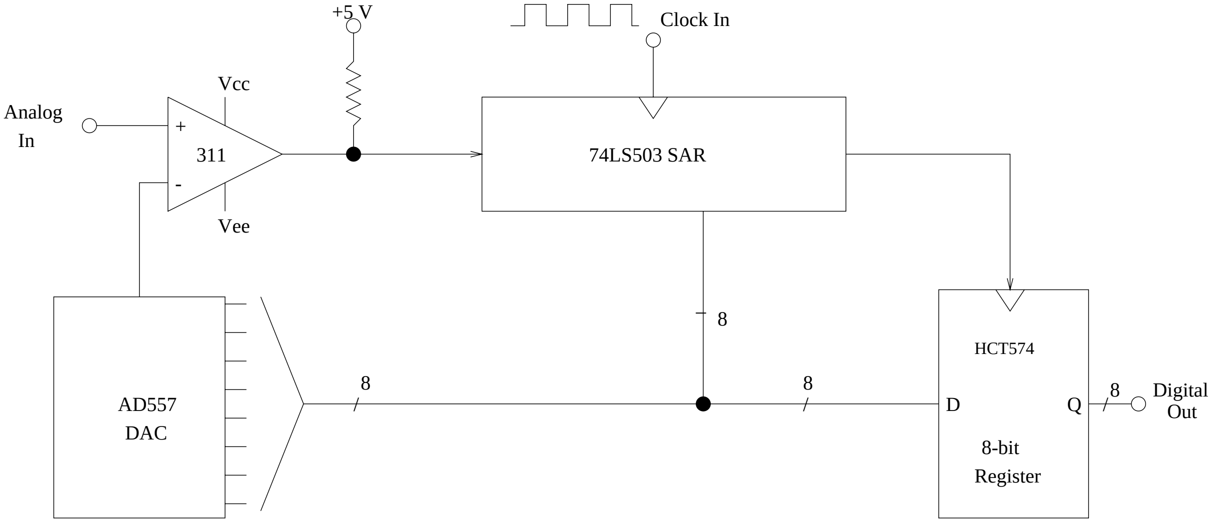

Figure 1: Scheme for successive approximation A/D conversion.

Typically, all components shown in the figure are integrated into a single ADC IC, which does not allow one to examine the internal workings of the device, as we will do in this lab.

5.2 DAC Checkout

First, we need to make sure we understand how the DAC works. The data sheets for the AD557 DAC are appended. We will connect the DAC as shown in Fig. 2. With the output connected in this way the analog output has the range 0 to 2.55 V, corresponding to the 8-bit digital input of 0 to $255 = 2^8 - 1$. The ceramic capacitor may be essential for eliminating runaway oscillation of the output.

5.2.1

Check out the input lines by alternately connecting to ground or 5 V. Note that disconnected inputs will float high, so it is important to hold all inputs at ground until you wish to put an input to `high`. What output does an input of 11111111_2 or 10000000_2 give? How about 00000000_2 and 00000001_2 ? Are these correct ?

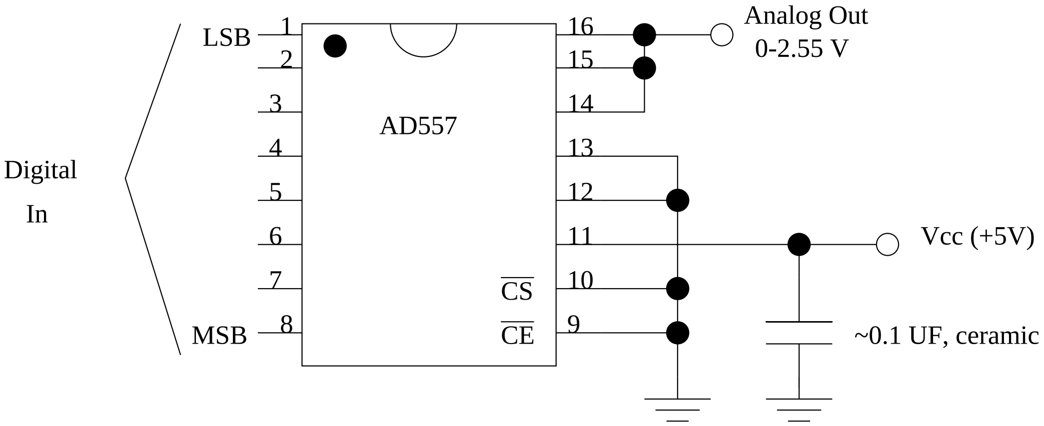

Figure 2: Connection scheme for AD557 DAC. The CS and CE control functions won't be needed for this lab and should be connected to ground, as shown.

5.2.2

Look at V_{out} on the oscilloscope. How does the difference between 00000000_2 and 00000001_2 compare to the level of noise? Qualitatively, what kind of frequency spectrum does the noise have? (That is, which frequencies, if any, seem to be most noticeable?)

5.3 A/D in Slow Motion

Now we can connect the successive approximation register (SAR) and voltage comparator to make our first ADC circuit. The detailed connection scheme is shown in Fig. 3. We will begin by evaluating this ADC in slow motion, clocking it by hand. The pin connections for the LM311 comparator are shown in Fig. 4.

To enable hand clocking of the SAR, use the prototype board debounced switches connected as shown in Fig. 5. Finally, provide an adjustable DC analog input by connecting the analog input (at the 10kΩ resistor) to the variable resistor of the prototype board, as shown in Fig. 6.

5.3.1

Connect the 8 prototype board LEDs to the SAR outputs Q0 – Q7 so that the present digital estimate can be directly viewed. Finally, we need to see when the SAR has completed its conversion cycle: The $\overline{\texttt{CC}}$ pin goes `LOW` when the cycle is complete. Connect a separate red LED with a current-limiting resistor ($\approx 470\ \Omega$), as shown in Fig. 3. The LED should be off during conversion and turn on when conversion is complete. Finally, note that pin 1 of the SAR is "chip enable" ($\overline{\texttt{E}}$); this must be grounded.

5.3.2

The SAR is a "synchronous" device, so it will start conversion only when the $\overline{S}$ signal is accompanied by a `clock` pulse. Give your device some analog input (measure with a DVM or scope). Now start the conversion and watch the analog estimate (output of DAC) with a DVM and the digital estimate (LEDs) while issuing additional `clock` pulses by hand. The binary search pattern should be evident.

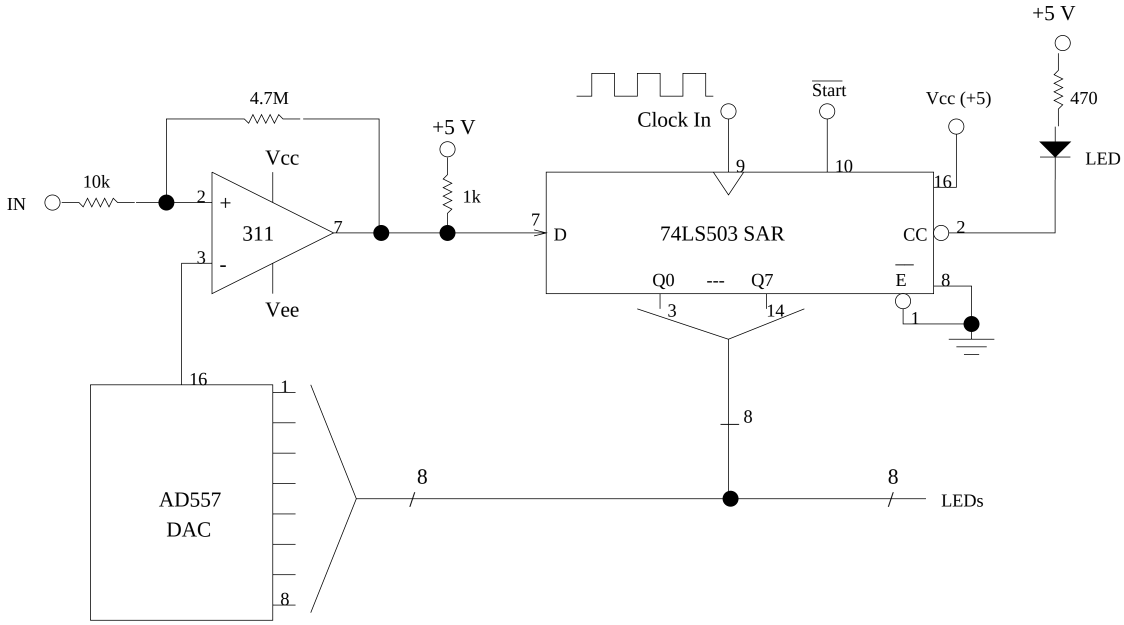

Figure 3: Connections for slow A/D checkout. Consult the data sheet for the pin assignments for the SAR outputs Q_0–Q_7.

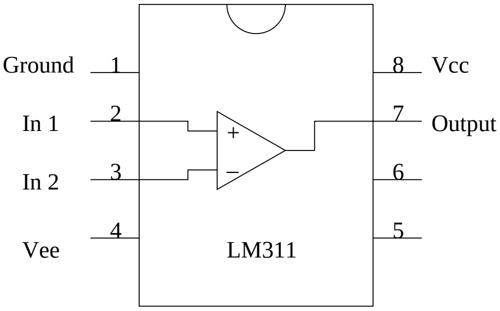

Figure 4: Connections for the LM311 voltage comparator. We should not need to use the balance adjustment (pins 5 and 6) for this lab. For this lab, use $V_{CC} \approx +6$ and $V_{EE} \approx -6$ V.

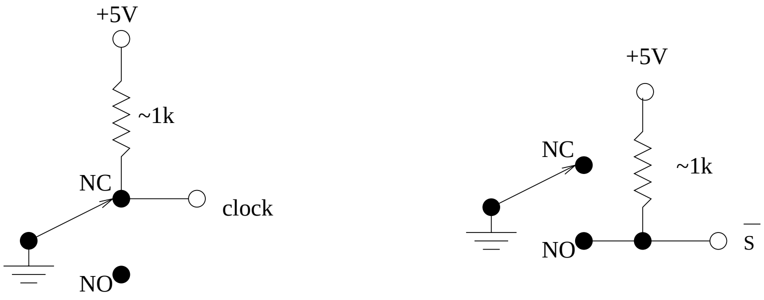

Figure 5: Debounced switch connections for `clock` (left) and $\overline{\texttt{start}} = \overline{S}$ (right).

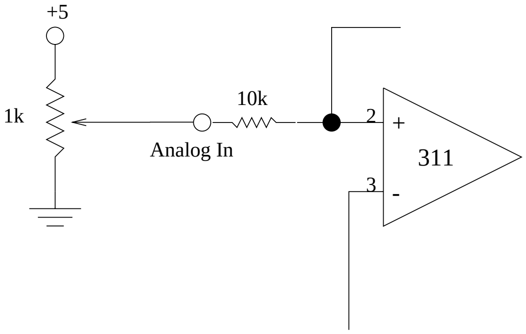

Figure 6: Providing an adjustable analog input.

How many clock pulses occur from start to end of conversion?
Carefully measure 6 input analog values including zero and full scale, and fill out a table with the following 4 headings:

V_{in}	V_{DAC}	Expected Digital Out	Measured Digital Out (LEDs)

Is the response linear? Make a plot of digital output *vs* V_{in}. Estimate the maximum non-linearity over the full input range. Is it consistent with ±LSB/2 ? Is there an offset?

5.4 Normal Speed Checkout

Set up continuous cycling operation by replacing the $\overline{S}$ button with a connection from $\overline{\texttt{CC}}$. That is, directly connect pin 2 to pin 10. Replace the `clock` button input with one from the TTL output of the prototype board function generator. Run the clock frequency somewhere in the range 10 kHz to 100 kHz.

5.4.1

Connect a scope probe to the DAC output. Trigger the scope on $\overline{\texttt{CC}}$. Now watch the conversion process on the scope. Vary the analog input and watch your ADC converge to the input value. Compare what you see to Fig. 9.52A of Horowitz and Hill.

5.4.2

There is a 2 MHz function generator in the lab. Replace your prototype board clock with this one. Watch the conversion process while increasing the clock frequency. You will probably notice a lot of parasitic capacitive coupling at the highest frequencies. But beyond this noise, does the final ADC value remain steady with increasing frequency? Is there a breakdown frequency, where the conversion no longer works properly?

5.4.3

Now we will make a pretty display. As before, connect the `clock` to the prototype board clock. Use a frequency of *few*$\times 10^4$ Hz. Using an external function generator, give your ADC an input consisting of a triangle wave of frequency $\sim$ 100 Hz and amplitude which spans the range of your ADC. Set up the scope as before. You should now see the entire binary search pattern on the scope, as your ADC attempts to converge to all possible input values. By carefully tuning the two frequencies it is possible to get a (nearly) stationary pattern. Compare with Horowitz and Hill Fig. 9.52B.
Use the oscilloscope camera to snap a picture of your trace to include in your report. (Note: for a normally bright scope trace, a shutter speed of 0.25 to 0.5 sec works OK. After pulling the photo from the camera, let it process for 90 sec before removing the chemical backing.)

5.5 Completing the ADC

Any real ADC will have a digital output which is latched and stored at the end of the conversion cycle. This will be done using the 8-bit D-type positive edge-triggered register

(HCT574) shown in Fig. 7. Connect up the HCT574 as shown, connecting the LEDs to its outputs.

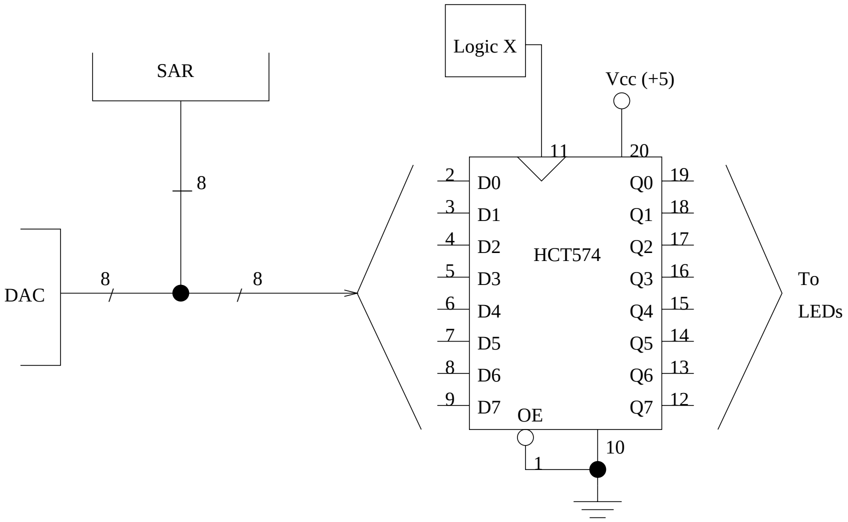

Figure 7: Connections for latching the digital output.

5.5.1

We need to provide a clock signal to latch the output at the end of the conversion cycle. A clear candidate for this signal would be the rising edge of the $\overline{\texttt{CC}}$ signal of the SAR as it returns from `LOW` to `HIGH`. Try using this. Vary the analog input and watch the LEDs. You should find that one of the LEDs never lights up. Which one? The reason for this can be seen from the timing diagram of Fig. 8. The rising edge of $\overline{\texttt{CC}}$ occurs too late, after the next cycle has begun. Similarly, the falling edge of $\overline{\texttt{CC}}$ is too early. In this case, the LSB would never be set.

By examining Fig. 8, can you find a single digital gate which provides the pulse we want, labelled “X” in Fig. 7. Connect this up and draw your “logic X” box. You will probably need to consult one of the TTL books in the lab to get the right IC and connection scheme. With this connection, verify that all digital bits can now be latched.

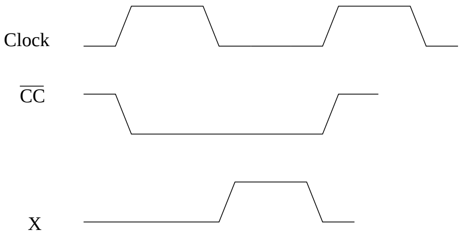

Figure 8: Timing for output register latching.