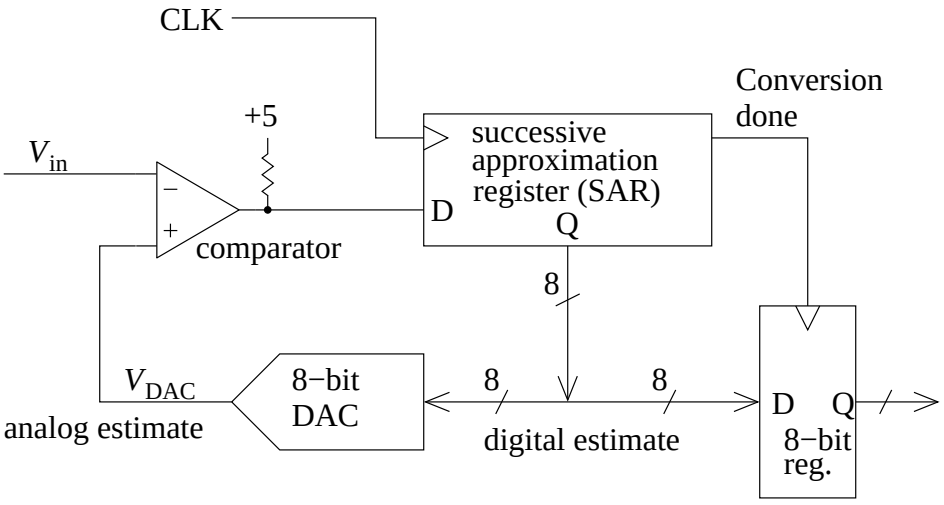
Physics 335
Lab 4 – Analog / Digital Conversion

The goal of this lab is to construct a successive approximation analog-to-digital converter (ADC). The block diagram of such a converter is shown below.



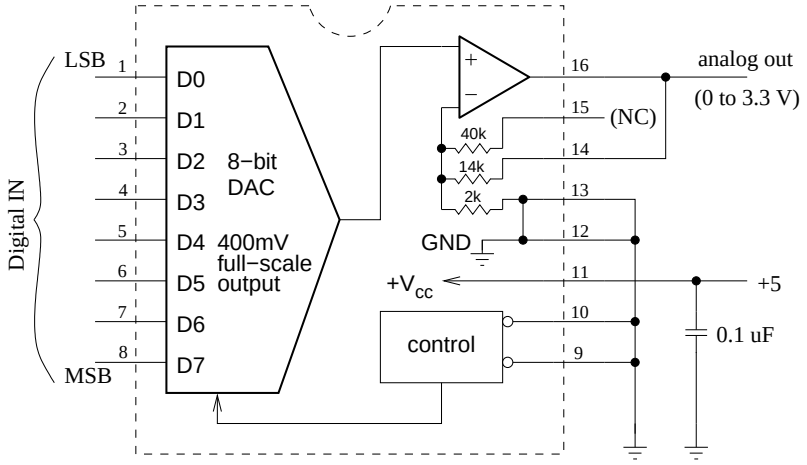
Note the idea: $V_{\rm in}$ on the analog input is compared to the output $V_{\rm DAC}$ of a digital-to-analog converter (DAC). If $V_{\rm in}$ is greater than $V_{\rm DAC}$, then the D input to the successive approximation register is set HIGH. On the first cycle, this sets the MSB to 1. Then the DAC reads this value, and outputs a new value of $V_{\rm DAC}$. On the next cycle, the comparator compares this new value with $V_{\rm in}$, and sets (or not) the next significant bit. The cycle repeats until all 8 bits are determined. This is a "binary search tree," and represents a relatively rapid way to converge on the closest digital number to $V_{\rm in}$ since at each clock cycle, it simply decides whether the input is greater than or less than half of the previous set of possible values.

In this experiment, you will first wire up the DAC and study its properties. Then you will connect the SAR and the comparator so that the DAC is in the feedback loop and watch the conversion process. Finally, if you have time, you can capture the output on a latch and display it

Use only the +5 Volt fixed supplies for this lab. You should tie the negative side of the supply to the ground terminal on the supply if it is not already so. This will ensure that you have +5 Volts above ground throughout the lab. As usual, a 0.1 μ F capacitor between +5 V and ground on the breadboard will help ensure you have a nice clean supply voltage

4-1 D/A Converter

The D/A converter (DAC) is an Analog Devices AD558. It has, on one chip, an 8 bit DAC stage (that uses an R/2R ladder and switches to create the analog signal) an output amplifier with user-selectable gain, and an input latch. We will set the latch into "transparent" mode, so that the analog output immediately follows a change on the digital input (following the delay determined by the DAC "settling time" of about 1000 ns).



Wire up the AD558 as shown in the diagram above. The bypass capacitor (0.1 μF) should be placed close to the chip, right across the "non-standard location" power supply pins, 11 and 12.

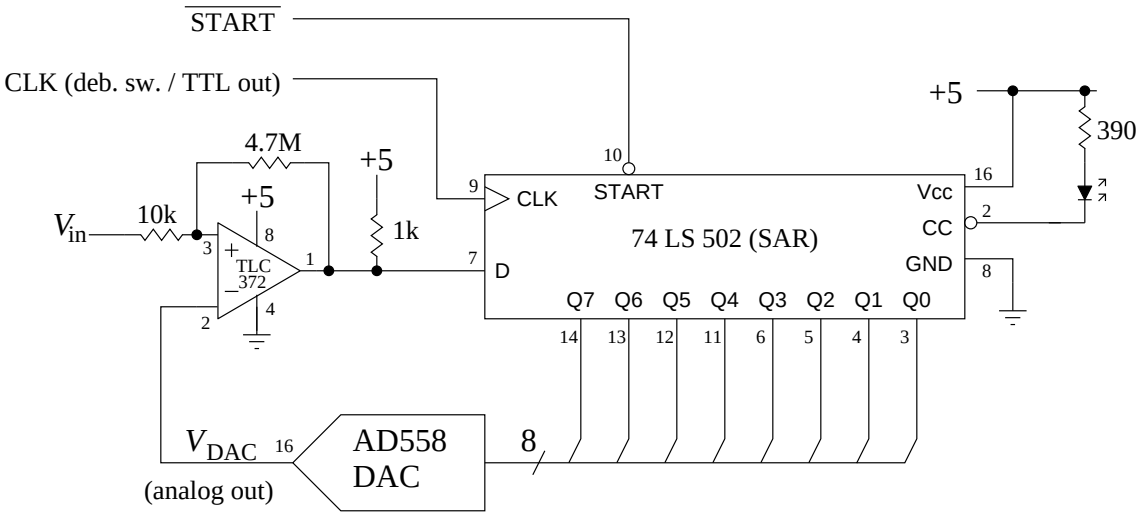
To check out the DACs operation, first connect all data lines high (+5), and measure the output voltage on pin 16 using your DMM. The hex value for this input is FF_{16}.

Based on this input and output, ***predict*** what the output voltage should be for an input of 80_{16} and 01_{16}. Then connect the appropriate lines HIGH or LOW and check your predictions. You should record both your predictions and measurements in your report.

4-2 Comparator and SAR: Completing the A/D Loop

You may have a 74LS503 or a 74LS502 in your parts cabinet. You can use either, but if you have a '503, it is necessary to ground pin 1 of the it to operate correctly in this exercise. (If you have a '502, leave pin 1 unconnected.)

Add the comparator (one-half of a TLC372 CMOS comparator) and the LS502 (or LS503; see note above) successive approximation register (SAR) to your breadboard containing the AD558.



Note that the comparator has a 1k pullup resistor on its output (pin 1) as well as the +5V and ground to pins 8 and 4, respectively.

Wire the clock input on the SAR (pin 9) to the output of one debounced switch and the START* input (pin10) to the output of another debounced switch.

Wire each of the data outputs, Q_0–Q_7 on the SAR to the corresponding data inputs D_0–D_7 on the AD558. Remember that the "8 /" on a line in a diagram means that 8 parallel wires are represented by that single line in the diagram.

Add the LED to the CC* ("conversion complete") pin through a pullup resistor (390Ω) to +5 V, as shown. Remember that the flat side of the LED housing is the cathode (more negative) side, so this side will "face" CC*.

To check out the circuit, first tie V_{in} to ground and connect a DMM to the analog output of the AD558 (pin 16). Then do the following:

(a) Assert START* (pull it LOW).

(b) CLOCK once (positive edge, LOW to HIGH).

(c) Look at (and record) the state of the data lines Q_0–Q_7. Use your logic probe. Also record the voltage on the output of the DAC.

(d) Disassert START* (pull it HIGH). If you don't do this, the conversion will stop at the first digit.

(e) CLOCK again, and note and record the data lines and DAC voltage.

(f) Repeat the above step until you see the LED on CC* turn on.

You should see the data lines turn off in succession as the SAR homes in on the correct input voltage (0 V), and at the same time, the output of the DAC should get closer to zero. (There may be a small offset voltage at the end.)

How many clock cycles does one conversion take, from start to finish?

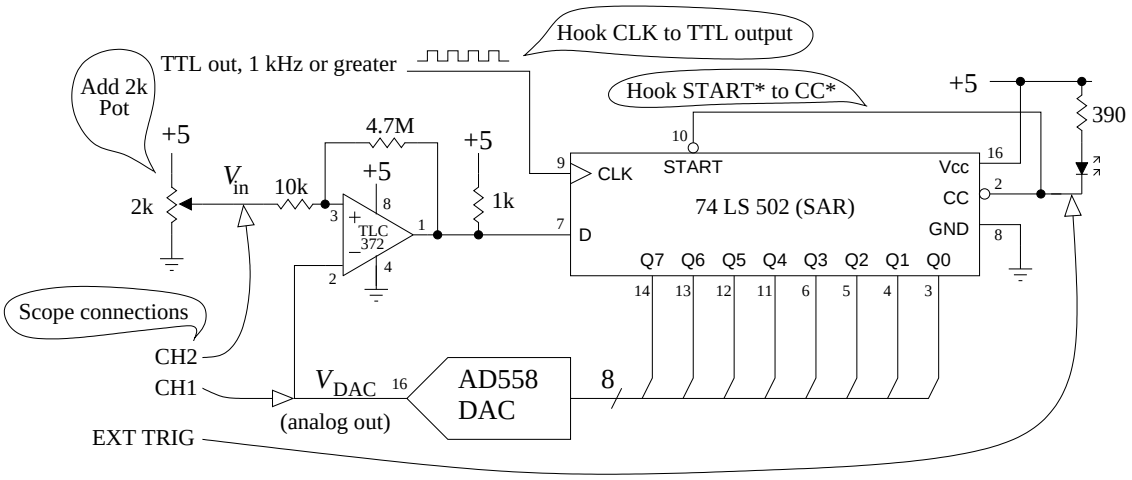
4-3 **Operation at Normal Speed**

Now make three changes to your circuit:

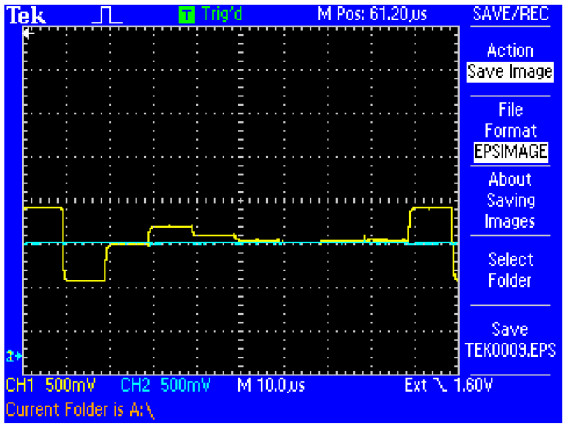
(a) Disconnect START* (pin 10 on the SAR) from your debounced switch and instead connect "Conversion Complete" CC* (pin 2) to START*. You may leave the LED attached (or not).

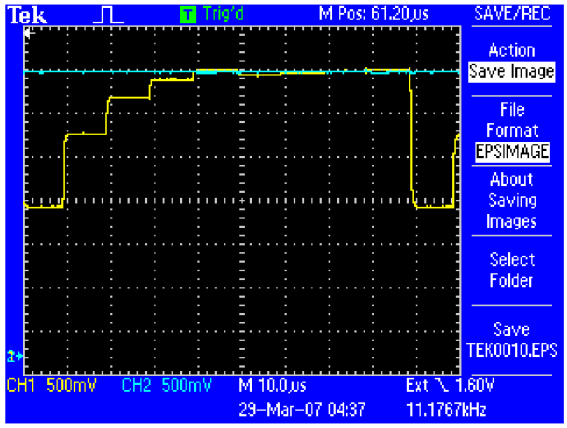
(b) Connect a 2k potentiometer between +5V and ground, and feed the wiper pin into V_{in} as shown below. This will allow you to see the ADC circuit convert a range of input voltages.

(c) Clock the SAR with the TTL output of your function generator. Use a clock frequency of around 1 kHz.



Then hook up your 'scope as follows: Connect the EXT TRIG input to CC*, and set the level to the typical TTL threshold of around 1.4 volts. This will synchronize your 'scope to the beginning of the conversion cycle. Then, using scope probes, connect CH1 to V_{DAC} and CH2 to V_{in}. You should see something like the screenshots shown below.





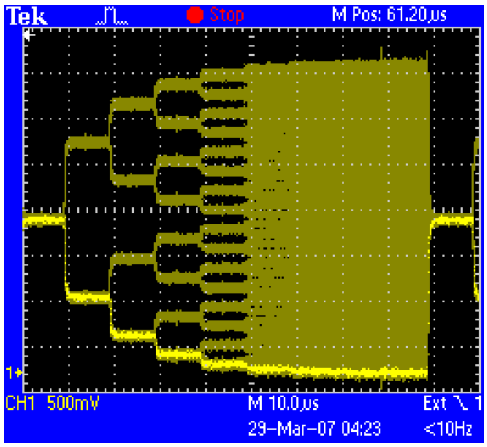
Watch what happens when you vary the setting of the 2k pot. Notice how the search tree finds its way to your value of V_{in}.

To see the digital number being produced, connect CH2 to the D input of the SAR (pin 7) and show it on the 'scope above (or below) the DAC output. Notice how the sequence of high and low parts of the waveform correspond to the up or down branches of the search tree. With a little effort, you can read the binary number corresponding to V_{in}.

4-4 Displaying the Full Search Tree

On your 'scope, press DISPLAY and then press the PERSIST softkey repeatedly until is shows "Infinity". This will store all traces indefinitely.

Then slowly turn the pot, varying V_{in} and see what happens. Pretty neat, huh? Notice how one particular branch is highlighted against a background of the whole search tree. You should see something like the picture below.



4-5 **A/D Speed Limit**

The A/D circuit completes one conversion for every 9 clock cycles. The faster you run the clock, the faster it will convert, up to a point. The "speed limit" of the A/D determines the highest frequency that can be digitized. According to the Nyquist theorem, to accurately digitize a signal of frequency f, one must sample the waveform at a little over $2f$.

To predict the highest frequency available to this A/D, you need to add up all of the delays through each part of the circuit. In looking at the diagram, between two clock edges there must be the following time-related events:

- The propagation of the output state on all Q_n of the SAR following the clock edge: *the SAR propagation delay.*
- The creation of an analog voltage on the output of the AD558 after the data are presented on the inputs: *the DAC settle time.*
- The creation of a HIGH or LOW bit on the output of the comparator following a change in levels on the comparator inputs: *the comparator delay.*
- The establishment of a stable level on the D input of the SAR: *the SAR setup time.*

This adds up to a minimum time of about 1800 ns. What maximum clock frequency does this imply? Given this, what would be the highest frequency signal that could be digitized? (Think about all of the information presented above.)

Test the speed limit. Set the knob on the 2k pot to feed a DC level somehere in the middle of the search-tree's range into V_{in} and watch the scope display as you increase the clock frequency. A some point you will see a breakdown: the final estimate will change because the clock will no longer give enough time for all levels to settle.

4-6 **Latching and Displaying the A/D signal**

This section is optional, but should be attempted if there is time.

Through most of this lab, you have been looking at the analog output of the AD558 to study the conversion process. However the goal of an A/D is to get the "D"—the number that corresponds to the voltage in.

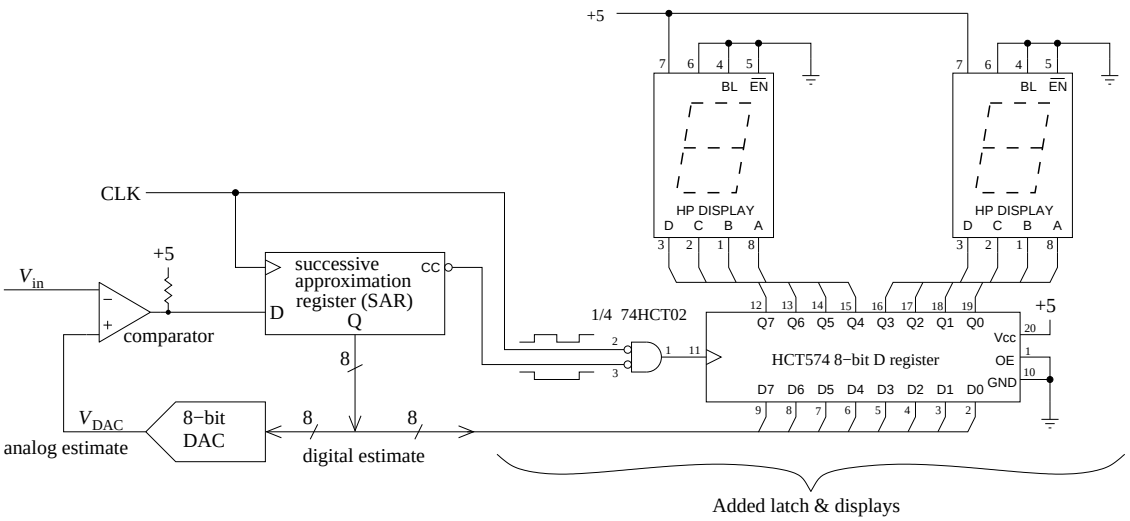
To capture this 1 byte binary number, you can use an 8-bit register, the HCT574 (TTL compatible CMOS). To display it, you may use the pair of HP hex displays.

The latching circuit requires a positive edge happening just after the final conversion. Unfortunately you cannot use CC* alone to create this edge because it comes too late–when CC* is disasserted, the next conversion cycle has already started and the number on the lines is the first "guess": $7F_{16}$.

An edge that happens at the proper time can be made by combining the clock signal with CC* through a NOR gate (one of four on an 74HC02 chip).

Wire up the HCT574 to the data lines D_0–D_7 as shown below. Note that +5 V will go to pin 20 and ground to pin 10. Also ground the register's enable line, pin 1.

Connect up the displays as you did in the counter lab: +5 to pin 7, ground to pins 4, 5 and 6, and the data lines to pins 8 (A), 1 (B), 2 (C), and 3 (D). Connect up the lower 4 bits Q_0–Q_3 to one display and the upper four bits Q_4–Q_7 to the other display.



Set the clock frequency on the SAR to about 10 Hz. Slowly vary the knob on the 2k pot. You should see the displays change at a rate of about 1 Hz as you vary $V_{\rm in}$.

Please disassemble your circuit, gently prying out any chips from underneath with a pair of angled forceps, and return all parts to the SAR LAB parts bin at your lab stations so they're there for the next people!

Prepared by D. B. Pengra and J. Alferness
Parts adapted from *The Student Manual for the Art of Electronics*
by Thomas C, Hayes and Paul Horowitz (Cambridge University, 1989)

`SARLab2.tex -- Updated 25 April 2014`