

Digital-to-Analog Converter (DAC)

Analog-to-Digital Converter (ADC)

Part I

0. In this part of the lab you will investigate a 1408 (a.k.a. DAC0808) which is an 8-bit, monolithic, current-switching, multiplying DAC with a settling time of less than $1 \mu\text{s}$ and a price of about \$2. As this is a CMOS chip, so be careful of static.
1. Locate the 1408 1" above the bottom of the middle breadboard. This is an analog chip: resistors, capacitors and a -15 V power supply are needed. The values given in the pin-out figure are only approximate. Note that the current flowing into pin 14 sets the scale of (multiplies) the digitally controlled current that flows into pin 4. This in-going current at pin 4 is converted to a voltage by having it flow through a resistor connected between pin 4 and ground. Thus, the (negative) voltage on pin 4 is the analog output of this DAC. Place a capacitor between pins 16 and 3 (100 pF or more should be fine). Ground pins 1, 2, and 15. Power with -15 V at pin 3 and $+5 \text{ V}$ at pin 13. Route the data inputs to the data switches. Since we have not yet used the -15 V power supply in the course, connect a voltmeter up to it before using it and adjust the pot to get -15 V . (Note: pin 5=MSB, pin 12=LSB)
2. Select the following input data on the data switches and record the (negative) voltage produced. Use inputs $255=\text{FF}_{16}$, $224=\text{E0}_{16}$, $192=\text{C0}_{16}$, $160=\text{A0}_{16}$, $128=80_{16}$, $96=60_{16}$, $64=40_{16}$, $32=20_{16}$, $16=10_{16}$, 8, 4, 2, 1, 0. Remember to switch voltage scales on the DMM to get the as many significant figures as possible.
3. Plot your data "backwards" with the measured voltage as x and the data-switch data as y . Report 1 bit as y -error. Fit ($WAPP^+$) a line to your data. Tape the hardcopy plot and fit report into your notebook. Circle the largest deviation shown on the fit report. You may now disconnect data switch inputs, but otherwise *save this circuit!*

Part II

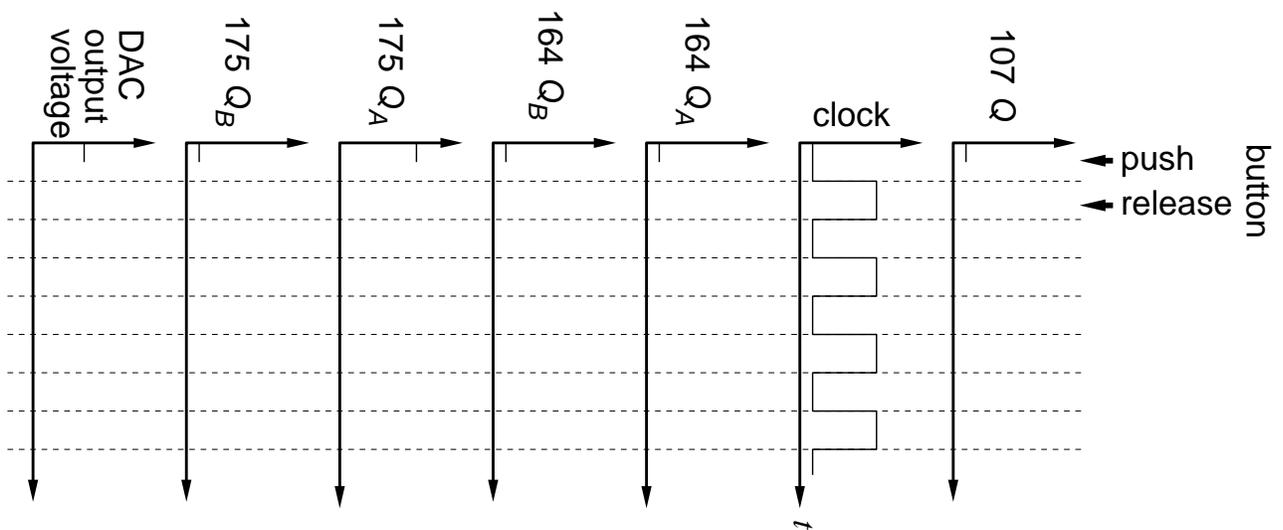
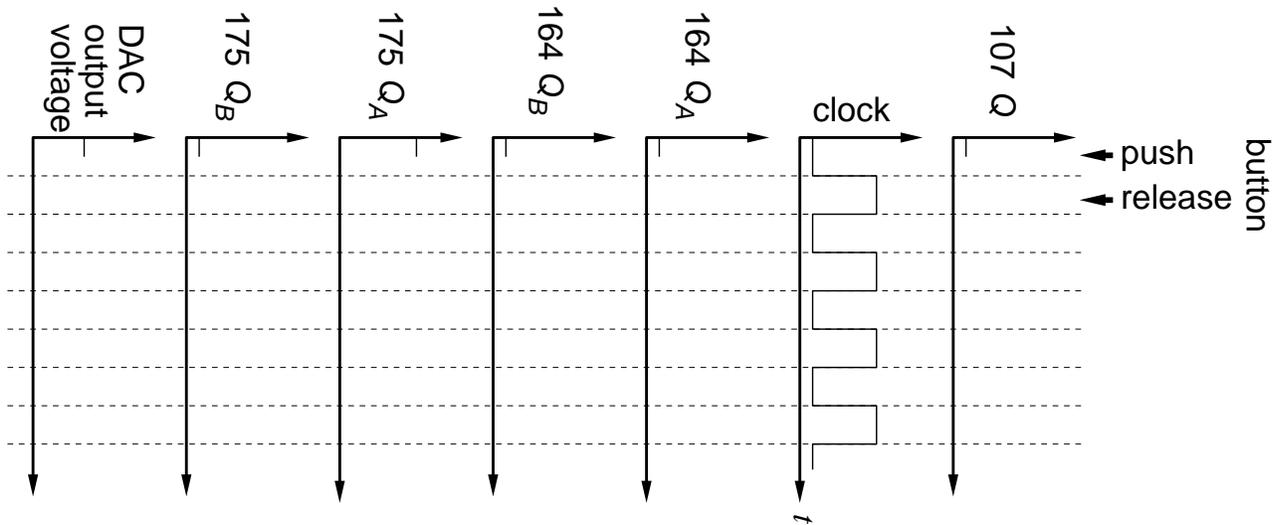
0. In this part you will construct a successive approximation analog-to-digital converter (ADC) from chips. You have seen most of the chips before. The '107 starts the process with a push of a single (debounced) button; the '164 shift register is used to select bits to be tested; two '175 edge-triggered quad DFFs (or one '273 octal DFF) store the digital approximations; INVERTER ('04), AND ('08) and two OR ('32) ICs are needed; the 1408 DAC produces an analog voltage corresponding to the present approximation. The 311 is a new chip to this lab: it's a voltage comparator. The 311 compares the voltage on its two inputs (" $+$ " and " $-$ "). If $V_+ > V_-$ the 311 sends its open collector output high; otherwise the 311 keeps its output low.
1. Starting on the top of the left breadboard, press in the following chips: '32 and another '32, '164, '04, '107. To the right of the '32s and above the 1408, press in two '175s. Between the '175s and the 1408 place a '08. Place the 311 below the 1408. Power-up all the TTL chips "at the corners", power the 311 as shown in the pin-out.

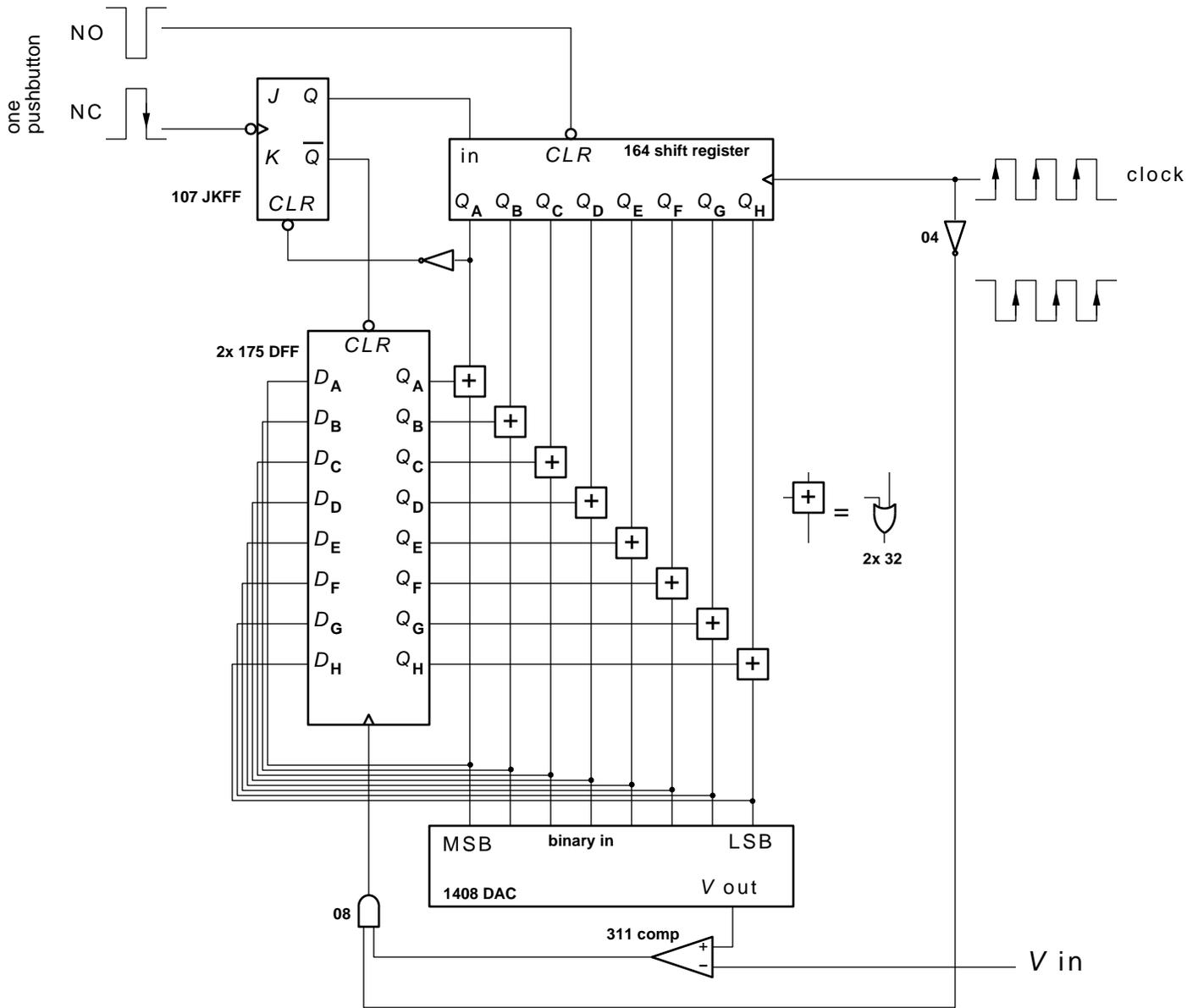
2. Wire up the chips following the functional diagram. Remember to debug as you go, e.g., check that the push button initiated sequencer works by running the '164 outputs to LEDs before you build the messy OR array. Next disconnect the '164→LED wires and reconnect them as DFF D -input→LED wires. (The LEDs are to display the digital output of this ADC.) Build the rest of the circuit, except substitute a side-board data switch for the 311 output. This data switch will then control the clocking of the DFFs, so you should be able to hold or reject a bit by hand. Debug any observed problems.
3. The last thing to add is the 311 (with its pull-up resistor). Look at the output of the 311 on a scope. Watch for high frequency oscillations... extra capacitors may be needed. Watch for 60 Hz “junk” from a poorly designed power supply... you may need to disconnect the ground lead on the power cord.
4. Wire one end of your 10k pot to ground, and—using a $\sim 20k$ resistor—connect the other end to -15 V; with this voltage divider the pot's sweeper can provide a V_{in} between 0 and -5 V. (Warning: don't destroy this pot!) Connect up one channel of the scope to the $+$ input of the 311 and the other channel of the scope to the $-$ input of the 311. Describe the action as the circuit searches for the proper conversion number. 
5. Collect about 14 (V_{in} , digital out) data pairs similar to those collected in Part I #3, i.e., digital output should range from $\sim 200^+$ (but not 255) to near zero. Produce and attach hardcopy plot and fit results. Circle the largest deviation on the fit report.
6. Describe (words!) how the circuit works. Include: chip clearing ('107, '164, & '175), timing considerations: how (and when) the '164 and '175 are clocked, how the 311 controls what is saved and the function of the OR array. What is the shift register doing before you hit the pushbutton? (“Nothing” is not the answer!) Sketch simultaneous scope traces of '107 Q , the clock, '164 Q_A and Q_B outputs, the corresponding '175 outputs, and the DAC output. (Assume that the DAC output corresponds to 10100000_2 at the beginning and end of the conversion, i.e., that the previous conversion was just like the one you are sketching.) Clearly show when each clearing action happens, when the '164 is clocked and when the '175 grabs its data.
7. Show the working ADC and plots to your instructor.

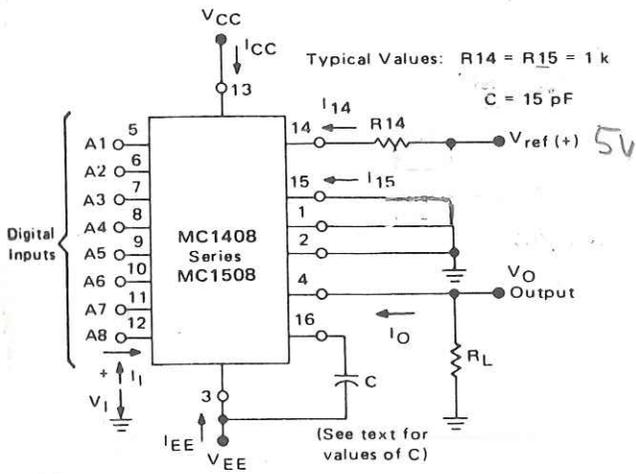
Part III

0. In this part you use a “stand alone”, microprocessor compatible, 8-bit, successive approximation ADC: the ADC0804. This \$3 chip does what the above circuit does, and more (e.g., built in clock, Tri-state outputs). You can scrap everything from Part II.
1. Press in your ADC0804 near the LEDs. Wire it as shown in the pin-out. Ground: \overline{CS} (pin 1, an enable), $V_{in}(-)$ (pin 7, voltage for digital 0 output), and A GND (pin 8, analog ground). Take \overline{RD} (pin 2, Tri-state enable: chip displays the data when low, i.e., microprocessor read) and \overline{WR} (pin 3, starts conversion when low, i.e, microprocessor writes to the chip) to separate NO push buttons. The clock frequency is set by an R and C : Run a ~ 20 k Ω resistor between pins 19 and 4 and a 100 pF capacitor from pin 4 to ground. Use the sweeper from your 10 k Ω pot wired as a voltage divider between $+5$ V and ground as a source of a (positive) variable voltage. Apply this voltage to $V_{in}(+)$ at pin 6. Power up the chip “at the corners.” Wire the digital outputs to the LEDs. Note the location of MSB and LSB in the LED column.

2. Test out your ADC as in Part II #5 (i.e., digital outputs from $\sim 200^+$ to near zero). Circle the largest deviation on the fit report. Note that you hit one push button to start a conversion, and then another to display that conversion. The conversion time depends on the RC set clock speed, typically about $100 \mu\text{s}$.







V_i and I_i apply to inputs A1 thru A8

The resistor tied to pin 15 is to temperature compensate the bias current and may not be necessary for all applications.

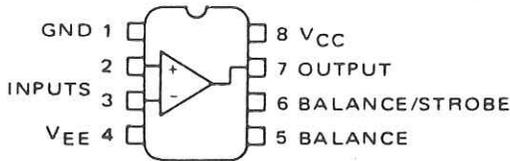
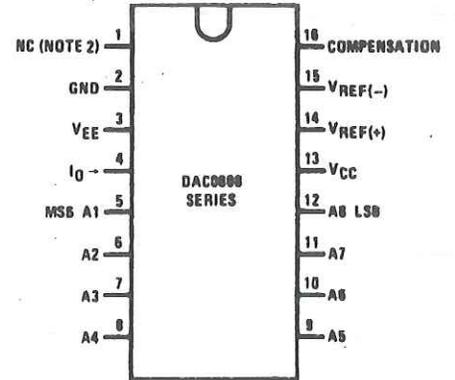
$$I_O = K \left\{ \frac{A_1}{2} + \frac{A_2}{4} + \frac{A_3}{8} + \frac{A_4}{16} + \frac{A_5}{32} + \frac{A_6}{64} + \frac{A_7}{128} + \frac{A_8}{256} \right\}$$

where $K \cong \frac{V_{ref}}{R_{14}}$

and $A_N = "1"$ if A_N is at high level
 $A_N = "0"$ if A_N is at low level

$R_{14} \sim 3k\Omega$
 $R_L \sim 1k\Omega$

$C \sim 100pF$
 $V_{EE} = -15V$
 $V_{CC} = +5V$



311

use $\sim 3k\Omega$ pull up resistor

ADC 0804

