

Digital Voltmeter

I. Introduction

Our goal here is to build a voltmeter. We will do this using almost all digital circuitry rather than analog. The circuit works like many other digital measurement circuits in that it converts the quantity to be measured into a time interval, then measures that time.

Call the voltage we wish to measure V_{in} . Say we charge a capacitor linearly with time (that is, at a constant charging current I), starting at zero volts. It will take a time $t = V_{in} / (dV_{in} / dt) = V_{in} C / (dQ / dt) = V_{in} C / I$ for the capacitor to charge up to V_{in} . So if we can measure this time interval, we can determine V_{in} . To measure time in digital electronics, we produce a high frequency pulse train and count the pulses occurring during the time interval to be measured. The number of pulses counted is a measure of the time interval, and hence of V_{in} . With some suitable conversion and a display, this makes a digital voltmeter (DVM).

To realize this idea with actual circuit elements (mostly integrated circuits or *IC's*) we require a number of separate sections. The complete circuit diagram for the eventual project is shown on the last page of this writeup (don't let it frighten you!). On the second-to-last page is the same circuit with its functional sections outlined and labeled. The sections needed for this project are:

- A) A pulse train generator. In electronics, this is called a *clock*. We have already learned how to build this using the 555 timer chip. This is the *clock generation* section.
- B) A way to monitor the voltage across the capacitor and generate signals when the charging starts at zero volts and when the voltage becomes equal to the voltage to be measured, V_{in} . I call this the section *control and gating*.
- C) A way to “count” the number of clock pulses that occur between the start and stop charging signals. This is the *counting* section.
- D) A constant current capacitor charging circuit that can be turned on and off and reset to zero volts by the start and stop charging signals. *analog input/transducer*
- E) A human-readable way to display the number of pulses counted during this interval. This is the *latching and display* section.

We will learn how to build each of these sections, then connect them together into the functioning DVM.

II. Section-by-Section Build-Up

So, let us get started.

A. Clock Generation

Part A we already have learned to do in the previous lab. Digital things, though, prefer to work at 0 or 5V. Therefore, either power the 555 from 5V instead of 10 or 15V (see spec sheet), or put pin three on a voltage divider to bring the output down to 5V. This is the method chosen in the final schematic at the end

B. Basic Control and Gating

To carry out the function of Part B we use a solution based on an integrated circuit called a *comparator*. The entire purpose of this IC is to signal which of two voltages is larger. One of the voltages will be V_{in} and the other will be the voltage across the capacitor. The comparator has to do its job without drawing any appreciable current, otherwise it will interfere with the constant-current charging. To achieve near-zero input current we will use a LF355 FET-input

op-amp as our comparator.

We will discuss op-amps in detail in class. For now, we can just learn that the op-amp is a chip with two inputs labeled + and – (with the voltages at these points called V_+ and V_-) and called the *non-inverting* and *inverting inputs*. It has just one output. Of course like any IC it requires power and ground connections, in this case two power supplies ($V_{cc} = +10V$ and $V_{ee} = -10V$ —these are just conventional names for the power voltages).

If the voltage at the + input is greater (more positive) than the voltage at the – input ($V_+ > V_-$), the 355 will drive its output high, near to V_{cc} . If the opposite becomes true, ($V_- > V_+$), the output will go low, near to V_{ee} .

Just for this part of the lab, we will use a couple of LED's to monitor the state of the op-amp output. The LED is symbolized by:



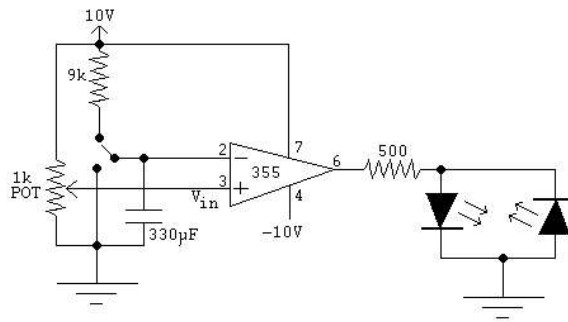
and is a Light Emitting Diode.

We discussed diodes generally in class. They act as one way valves for current. The arrow points in the direction conventional (positive) current is allowed to flow. The shorter lead wire on an LED corresponds to the tip of the arrow. (For historical reasons this side is also called the *cathode*; the other side of the device is called the *anode*.) When the diode is “biased” (i.e. voltage is applied) to conduct in this direction, it is said to be *forward biased*.

If you forward bias these LED's with less than about 1.6 Volt, little current flows and no light is produced. On the other hand, if you try to cause a larger forward voltage drop across a diode, you will find that you won't be able to do it. Any diode will always “try” to maintain a certain voltage drop of about 1 Volt across itself (see below for details on the exact value). The diode will keep its forward bias voltage drop constant by drawing more and more current from the power supply until either the supply sags or the diode burns up. The exact value of the forward bias voltage drop depends on the material the diode is made of and nothing else. Red LED's are made of gallium arsenide (GaAs), a man-made material that does not exist in nature. They have a forward bias voltage of about 1.6 volts and they only emit light when the forward bias voltage is at least this big. Signal diodes are usually made of silicon with a forward bias voltage of 0.6 Volt. An LED will not light when it is reverse biased (voltage polarity is reversed).

In the circuit below, only one LED should light at a time- the one on the left will light if the op-amp output is positive, or the one on the right will light if the op amp output is negative. LEDs should always have a resistor of $\sim 500\Omega$ in series with them to guarantee that the voltage they see will sag sufficiently at high current to avoid burning out the diode.

1) Build the voltage monitor circuit shown above. The SPDT switch shown below the 9K resistor controls the capacitor charging. The 1k potentiometer is wired to form a voltage divider, allowing us to set V_{in} to anything between 0 and 10V. The time constant for charging the capacitor is fixed by the capacitor value and the 9 K resistor, and is slow enough that you should be able to notice the longer charging times as V_{in} is increased. (Note: V_{in} is what will eventually be the voltage we are trying to measure. The capacitor is always charged—in this schematic—from 0 to some V_{in} value less than 10V. But it will take it longer to charge to a higher voltage, so the 355 comparator will measure a longer time for V_- to become greater than V_+ as V_{in} is increased.) Make sure the electrolytic capacitors are placed with the correct



polarity!! This means the lead marked positive should go towards the +10V.

2) In this configuration, the op-amp will put out -10V or 10V at pin 6 (actually slightly less. Measure the real values). Since our two LEDs are wired with opposite polarity, in either case only one of them will be forward biased and hence emit light. Choose one of the LED's red and the other one green and orient them so that the green one lights when the capacitor has a larger voltage than V_{in} . When the switch is flipped to connect the 9 K resistor to the capacitor, the charging starts. The red LED should initially light, then after some delay, it should go out and the green one should come on. If V_{in} is increased by adjusting the 1K pot, the delay should get longer. After the switch is flipped in the other direction (disconnecting from the 9K resistor and shorting across the capacitor), the red LED should come on immediately, since the capacitor is discharged directly to ground without a resistor.

3) Making certain the capacitor has the right polarity, turn the potentiometer to apply its maximum V_{in} . Flip the switch to start charging. You may not see the green light go on. This is because the op-amp cannot work with input signals very near its power supply voltages. Back off a bit on the potentiometer to get V_{in} a volt or so lower than $V+$ and try again.

C. Counting Clock Pulses to Measure Elapsed Time

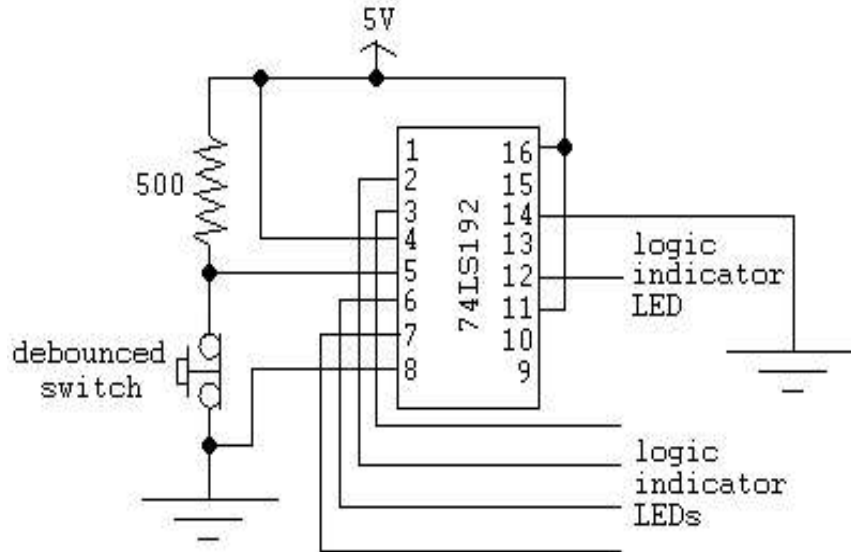
1. Counting Pulses with the 74LS192 Chip

Digital electronic chips of the type known as "TTL" use 0V for an off signal (0) and 5V for an on signal (1). This is the defining characteristic of digital electronics: any input or output can only be in one of two possible states. The TTL output of the breadboard's function generator generates a pulse train for use with the TTL "family". So does an appropriate 555 circuit. The TTL chips used in this class are all designated by the letters "LS" in its JEDEC number (Example: 74LS00 is a quad NAND gate).

To "count" the pulses generated while the capacitor is charging, we will use a 74LS192 integrated circuit, which is a "divide by ten counting chip". It is a very capable little guy, and costs only about a buck. This chip accepts a logic pulse train as input on pin 5 ("UP COUNT"). It counts pulses from 0 (0000 binary) to 9 (1001 binary), continuously putting out the binary bits corresponding to the number of pulses counted on pins 3, 2, 6, 7 (LSB to MSB). When it gets past 9 it puts out a "carry" signal on pin 12, the binary bit outputs roll over to 0000 again, and then it repeats. It can also count down as well as up but we won't use that here. Secure the datasheet for this part by asking your instructor or entering "74LS192 data sheet" into Google and picking the appropriate reference.

The innards of the chip (the "magic" in the sense Arthur C. Clarke used the term) will be explained in class. To familiarize ourselves with this chip, we will send signals to it one-at-a-

time using a pull-up resistor and a *debounced switch*. Read the section *Switch Debouncing* on pages 506 ff. of Horowitz and Hill (H & H). This section explains why we must use a “debounced switch” to signal the counter - so that each switch press sends one and only one pulse into the counter. Fortunately the breadboards already provide debounced switches, located on the left side.



1) Wire the switch-and-resistor part of the counting circuit above, using a *normally closed*, debounced, pushbutton switch. A 500 Ω *pull-up resistor* should run from 5V to the switch. The resistor will then be tied to ground whenever the switch is closed, presenting a logic 0 to the COUNT-UP input (pin 5). When the switch pushbutton is pressed, the switch is opened and therefore breaks the resistor connection to ground. A 5V input is then presented to the COUNT-UP input (minus of course, the voltage drop across the resistor—but remember that inputs to logic chips draw little current so this drop is negligible.). This transition from 0 to 5 V is counted as one pulse.

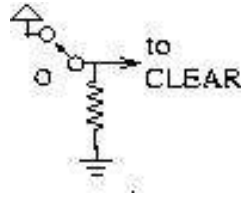
2) Wire the rest of the circuit as shown. The 74LS192 is a sophisticated device with a number of inputs we are not using. These unused input pins must be tied to either ground or +5V, because pins left “floating” (unconnected) can drift to random voltages and cause spurious inputs to the chip. The COUNT DOWN (4) and LOAD (11) pins must be tied high (5V). The CLEAR input (pin 14) must be tied to 0V. Wire the 74192 binary bit outputs 3, 2, 6, and 7 to the first four logic indicator LEDs on the right side of the board. (These logic indicators already have series resistors built-in to protect the LED's from overcurrent.) Make sure the least significant bit (LSB, pin 3) is wired to the right-most LED, and the most significant bit (MSB, pin 7) is wired to the left most LED so that the binary display will read normally.

3) Now push and release the switch repeatedly. You should see the logic indicators count up one unit for each button press, from 0000 to 1001 (in binary for 0 to 9), then go back to zero and repeat. This should repeat as long as you continue actuating the switch.

4) Disconnect pin 5 from the pull-up resistor and connect it instead to a TTL signal from the function generator running at a low frequency. Observe that the binary outputs now count up

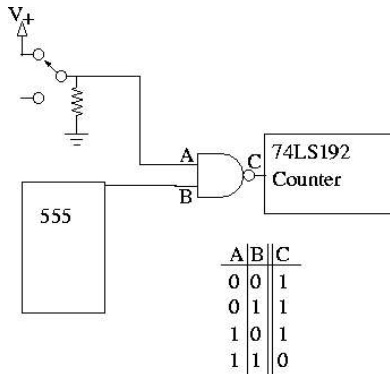
without your button-pushing. Next, tie the CLEAR (pin 14) to a pull-up resistor and a DIP switch (see note below) and note its function. Lastly, tie the CARRY (\overline{CO} , pin 12) to a fifth logic indicator LED and note its behavior. Keep this circuit together.

Note on DIP switch pull-up setup:



Some people would call this a *pull-down* resistor since it connects to ground rather than 5V as in the debounced switch setup above. To set this up use the row of DIP switches on the lower left of the prototyping board. The slide-switch on the far right internally connects one side of all the DIPs to either 0 or +5 V. Make sure this switch is set to 5V. To avoid shorting the 5 V supply directly to ground, a $\sim 500\Omega$ resistor to ground is needed. From the high side (5V) of this resistor, run your input to the CLEAR pin (or where ever else in the circuit a manually-operated logic input is needed).

2. Switching the input to 74LS192 On and Off with a NAND Gate



The voltmeter will work by counting the pulses from the 555 between the time the capacitor charging starts, and the time the 355 comparator detects that the capacitor voltage has surpassed V_{in} and therefore changes its output state. To accomplish this we will use a simple NAND gate in the circuit shown above.

The NAND gate was discussed in class. A NAND is a NOT AND boolean circuit element. The truth table is given in the diagram above. The pull-up resistor will provide a logic 1 to input A of the NAND when the switch is closed (this imitates the situation we will have while the capacitor is still charging), and 0 if the switch is open. Input B of the NAND is connected to the pulse train output of the 555. The output of the NAND will therefore follow the 555 pulse train (inverted) if the switch is closed, and the NAND output will just sit at logic 0 otherwise. The signal from the pull-up resistor and switch is said to *gate* the counting of the 555 pulse train.

1) First explore the operation of the NAND. Find a 74LS00 IC and also its datasheet. The 7400 actually has four independent NAND gates in one package. Just pick one of them for this exercise. To drive the NAND, assemble the 555 timer circuit from page 287 of H & H. You could use the TTL generator, but the 555 will be needed later anyway. Select resistors and

capacitors to give a pulse repetition rate of a few Hz so that you can see the output change visually. Then connect the 555 output pulse train to one input of the NAND, and the second input to the DIP switch with a pull-up resistor.

2) Verify with a scope that the output of the NAND only oscillates when the DIP switch is closed and the input is therefore pulled up to +5 volts (logic 1)

3) Now wire the NAND output to the COUNT UP pin of the 74LS192 counter from the preceding section. Verify that the counter only counts when the DIP switch is pulled up. If the 555 pulse repetition rate is too high, you won't be able to see the four indicator LEDs switching as the count progresses. They will look as if they are all on at the same time, with a reduction in brightness. When the DIP switch is off though, the LED's should freeze in a steady pattern.

4) Use two probes and a scope to verify that the 555 signal is being inverted by the NAND. You will need to display both scope channels at once.

D. Improving the Analog Input/Transduction Section

1. Constant Current Charging

Why is it unsatisfactory to charge the capacitor through a fixed resistor using a fixed voltage? (Hint: $V = V_0(1 - \exp(-t/RC))$). We now need a constant current capacitor charging circuit that can be turned on and off and reset to zero volts by the start and stop charging signals. This is a complex task, and thus part D will have several sub sections.

As explained at the very start of this write-up, the capacitor must be charged linearly with time, so there is a fixed voltage rise per unit of time. How do we charge a capacitor linearly? Recall that $V = q / C$ and that 1 Amp = 1 Coulomb/ Sec. So, only if we charge the capacitor with a constant current do we increment the charge by the same amount in each second, hence also incrementing the voltage by a constant amount in each second. This gives a linear $V(t)$. For the constant current drive we will use the LM334, a three terminal current source that just requires one external resistor to set the current.

An *ideal* current source would provide a fixed current no matter what load it was connected to. But, as with the voltage divider, any *real* current source will sag under certain conditions. Notice $+V_{IN}$ and V_- on the diagram below. The $+V_{IN}$ terminal is the power supply voltage input to the chip. The V_- terminal is the output pin from which the current (I_{SET}) will be supplied to our load (the capacitor). As the signs indicate, V_- must be less than $+V_{IN}$. If the loading is such that V_- gets too close to the supply voltage, the output current will sag or cease altogether. For correct operation we must satisfy the condition $+V_{IN} - V_- > 0.5V$. Naturally this means we cannot charge our capacitor above $(+V_{IN} - .5V)$. As with most IC's, there are many possible applications and circuits for the 334. The figure below shows the simplest configuration. V_{in} can be anywhere between 1 and 40V. R_{SET} determines the current by the following formula: $I_{SET} = (227 \text{ } V / K) * T_{ambient} / R_{SET}$. For this project, assume $T_{ambient} = 300K$. In a real instrument, we would have to build in temperature compensation to make the charging current independent of the ambient temperature.

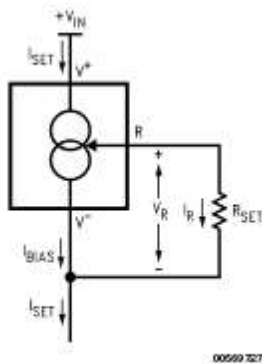


FIGURE 1. Basic Current Source

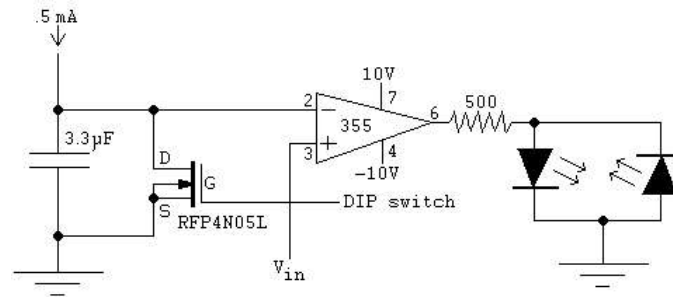
- 1) Using the above formula, calculate R_{SET} to produce a current of 0.5mA.
- 2) Secure a copy of the datasheet for the LM334 so you can identify the leads. Assemble the little fragment shown above, powered by $V_{IN}=5\text{ V}$, and measure the output current to verify your calculations. The 334 is temperature dependent, so warm it a little with your finger and see how the current changes. It should be a small change.
- 3) Put the above constant-current source into the capacitor charging circuit of section B1, in place of the 9K resistor to the positive supply as shown in the full circuit diagram on the last page of this writeup.
Verify the function of the circuit by placing an ammeter in series with the current source and capacitor, and watching the voltage on the capacitor with the oscilloscope. You should observe on the scope that the capacitor never charges above $(+V_{IN} - .5\text{ V}) = 4.5\text{V}$ as mentioned previously. The ammeter should show a .5mA reading until the capacitor voltage levels out near 4.5 V, then it should read zero. Replace the 5V supply voltage with 10V. Notice that the current is still the same. The 334 output current is independent of supply voltage.
- 4). Next, measure the slope of the capacitor charging curve $V(t)$ on the oscilloscope and verify that $V(t)$ increases linearly with time. Compare the measured slope with the value you expect from calculation.

2. Automated Control of the Charging Circuit

Given the measured current of your 334, calculate a capacitor value to give a charging rate in Volts / second between 100 and 200 V / s. This choice will allow the final meter circuit to re-read the input voltage at a fairly high repetition rate, allowing it to follow if the input voltage varies with time. For example, if $I = 0.5\text{ mA}$ and $C = 3.3 \times 10^{-6}\text{ F}$, $V(t) = .5\text{mC} / \text{s} / 3.3 \times 10^{-6}\text{ F} = 150\text{ V} / \text{s}$. This implies .006 seconds per volt.

To start and stop the capacitor charging circuit automatically, we will use a device called a *MOS-FET transistor* as an electronic switch. The MOS-FET to be used here is called an RFP4N05L. MOS-FETs are extremely sensitive devices - meaning they are easy to destroy by static electricity. Ground yourself to the shell of a BNC connector on the breadboard before handling the MOS-FET. The terminals of this device are named *gate*, *source*, and *drain*. The gate is the control terminal which controls how much current passes from drain to source. Much of the magic of MOS-FETs is related to the fact that the gate draws an exceedingly small current, picoamps or less. With that tiny current (and correspondingly tiny control power $I_{gate}V_{gate}$) we can control a much larger current I_{drain} of hundreds of mA (and correspondingly much larger

power $I_{\text{drain}} \cdot (V_{\text{drain}} - V_{\text{source}})$ going from drain to source. These devices come in a variety of



types with different characteristics, but we do not need to concern ourselves with that now.

We will build up the automated control of the charging circuit in steps.

1) First let's see how to use the MOS-FET to start and stop the charging. The device will act essentially like a switch across the capacitor that we can open and close- but it will open and close in response to a logic signal presented to its gate. Rearrange your capacitor circuit from part B once again, this time to look like the figure above (the 0.5mA comes from the 334 current source as before). Supply a variable V_{in} (as before) using a 1k potentiometer between ground and 10 V. Measure V_{in} with a known-working voltmeter. Notice we are tying the gate of the MOS-FET to a DIP switch and pull-up resistor. We are still going to operate a manual switch to start and stop the charging, but now the manual switch is actually just presenting a logic signal to the MOS-FET gate. That signal causes the MOS-FET “switch” to open and close. Later on, the logic signal will be supplied by yet another chip (the 74LS74 flip-flop) instead of a manual switch.

2) Verify that when the DIP switch at the MOS-FET gate is opened and closed, the LEDs toggle as before. When the gate is taken low, the MOS-FET acts like an open circuit from drain to source, allowing the capacitor to charge. When the gate is taken high, the MOS-FET acts like a closed switch shorting across the capacitor and bringing the charge to zero. Observe that the MOS-FET gets warm to the touch. It is passing current ($\sim 0.5\text{mA}$) whenever the gate is high.

3) Though charging linearly, the capacitor charges too fast to visually see the delay between red and green LED's caused by the charging time. To verify the delay, attach the probe of an oscilloscope across the capacitor and watch it charge and discharge for several different values of V_{in} . It may be difficult to get the scope to trigger properly (you should be on *normal* triggering), but persevere! Verify that the slope, etc. are still the same as they were in D1 step 5) above.

B. Controlling the MOS-FET with a 74LS74 Flip-Flop

The 74LS74 is a *clocked D-type flip-flop*. We will use it to mediate between the comparator and the capacitor charging circuit. This device has power and ground connections like any logic IC. It also has inputs called D, CLOCK, and \bar{R} R-bar and outputs called Q and \bar{Q} . The presence of a CLOCK pulse determines whether or not the device “listens” to the D input. Without a logic 1 signal at CLOCK, the outputs stay constant no matter what happens at D. If both CLOCK and D are driven with logic 1 at the same time, then Q will go to logic 1 and \bar{Q} will go to logic 0. If CLOCK is driven with logic 1 and D is driven with logic 0 at the same time, then Q will go to logic 0 and \bar{Q} will go to logic 1. If \bar{R} is driven with logic 0 it will force Q to logic 0 and \bar{Q} to logic 1 regardless of the logic level presented to CLOCK.

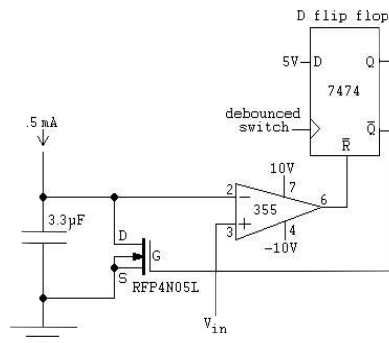
Wire up a 74LS74 and connect the \bar{R} input to 5 V. Using pull-up resistors and DIP switches connected to D and CLOCK, and logic indicator LED's connected to Q and \bar{Q} , verify that the above description of its operation is correct.

Next explore the effect of \bar{R} . Connect D to +5 V so that it is always high (logic 1), and R-bar to a switch and pull up resistor. What should happen when R-bar input is grounded? (Hint- remember the Trigger-bar input on a 555?). Verify that the effect of R-bar is to “jam” Q to zero regardless of the signal presented to CLOCK.

In the final DVM circuit, the comparator will drive \bar{R} and CLOCK will be driven by a pushbutton, or by the TTL pulse train from the function generator. The D input will be tied to +5 V. How these connections perform the necessary measurement cycle will be explained in the next section.

Part D.3

Now we will put together the parts discussed so far and get the automated charging operation cycle to work. We will then add one refinement, and then move on to the latching and display section of the DVM circuit.



1) Once again amend your capacitor charging circuit, making it look like that above. The debounced switch connects to the 74LS74 CLOCK input and is wired as in part C.1 step one. The switch is there to allow us to manually generate a logic signal to start one charging cycle just for testing. This logic signal could alternatively be generated automatically.

2) Run Q to an indicator LED. Also, run \bar{Q} to an indicator, as well as to the MOS-FET gate (determine which MOS-FET pin is which from a data sheet). Tie \bar{R} to a third indicator LED, as well as to the op-amp (comparator) output. As ever, the 355 op-amp takes as its inputs the voltage we wish to measure and the voltage of the charging capacitor. Whenever $V_{in} > V_{Cap}$ the comparator output is +10V. In the other case, $V_{Cap} > V_{in}$, the output is -10V (the power supply voltages +/- 10 V are often referred to as *the rails*). The op-amp output controls the operation of the D flip-flop (through \bar{R}).

So, lets go through a cycle of operation. Assume that initially (at power-up) the capacitor is uncharged. Therefore $V_{in} > V_{Cap}$, $\bar{R} = 1$, $Q = 0$, and $\bar{Q} = 1$. The $\bar{Q} = 1$ signal pulls the RFP4N05L gate high, so the MOS-FET “switch” is closed, shorting across the capacitor.

Now we press the debounced switch and send a signal to the CLOCK input of the 74LS74. That chip then reads the signal at its D input, which is always logic 1 (D is tied to +5V). Therefore the 74LS74 output state changes to $Q = 1$ and $\bar{Q} = 0$. $\bar{Q} = 0$ immediately

opens the MOS-FET switch, allowing the capacitor to start charging. Later we will use the Q output to start the counting and display sections (not shown in the diagram above).

The capacitor continues to charge until $V_{\text{Cap}} > V_{\text{in}}$. As soon as V_{Cap} exceeds V_{in} , the output of the op-amp comparator goes low, driving the \bar{R} input to logic 0. This resets the flip-flop to the state $Q = 0$ and $\bar{Q} = 1$, which closes the MOS-FET “switch” again and instantly drains the capacitor. The state of the circuit is now exactly the same as it was at power-up. Nothing more will happen unless another logic 1 is presented to CLOCK.

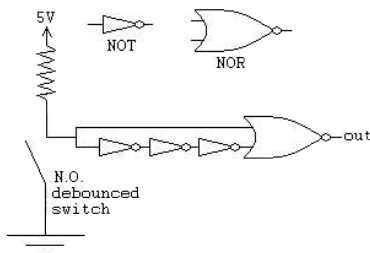
The outcome of all this is that every time the debounced switch is pressed, the capacitor goes through one cycle of charging at .5 mA up to $V_{\text{Cap}} = V_{\text{in}}$, then gets instantly discharged. The Q signal of the 74LS74 is logic 1 during the charging time and logic 0 otherwise. This switching of Q and \bar{Q} in time with the capacitor charging is the heart of the DVM we are building. We will use this to control the counting and display circuitry in the following sections.

- 3) Power the above circuit and use indicator LED's or an oscilloscope to check that it actually goes through the cycle explained in detail above. Basically, you should see $Q = 0$ and $\bar{Q} = 1$ at power-up. Then pressing the switch should reverse the states of Q and \bar{Q} for a short time, after which they should return to their initial states. It will be very difficult to see \bar{R} switch from its power-up value of 1, to 0, and back to 1 again, since it is only in the logic 0 state for that incredibly brief period of time it takes the capacitor to *start* discharging.
- 4) At the beginning of this section, the need for a couple of refinements was mentioned. Here is the symptom of a need for one of them: Notice that if you hold down the switch, the capacitor will charge up, but it will not discharge until you let go.

Part D.4: A Refinement

Why does the capacitor not discharge if the switch is held down? The poor 74LS74 is getting contradictory messages from its inputs, that's why. As long as the switch is pressed, the 74LS74 is sensitive to its D input, which is held high. So this should make the output state go to $Q = 1$ and $\bar{Q} = 0$. However once the capacitor gets fully charged, the op-amp comparator output is signaling $\bar{R} = 0$, which is supposed to drive the outputs to $Q = 0$ and $\bar{Q} = 1$ instead. According to the 74LS74 specs, the D input “wins” this conflict and the capacitor never gets discharged.

There are two ways to fix this. 1) Use an *edge triggered* 74LS74, which does exist and is called a 74LS74A. We discussed edge triggered logic devices in class. Or, 2) send only a very short pulse to the CLOCK input no matter how long the switch is held down. The circuit shown below accomplishes fix number 2) in a “crude but effective” way.



You should have learned about NOT, NOR, and NAND in lecture. In Boolean logic they perform their functions instantaneously, but in the realm of electronics there is some propagation

delay. For example, the inverting of a signal (NOT) may take around 20ns. So here is the deal: with the debounced switch open, the upper input to the NOR is 1. The lower input, produced by the three NOTs, is 0. Recall that a NOR only outputs a 1 when both inputs are 0, so here it outputs 0. Now we close the switch. The upper input to the NOR immediately goes to 0. Due to the propagation delay through the three NOTs, the lower input stays at 0 also, for about 60ns of propagation delay. Then it turns to 1. The effect of this is that the NOR output gives a 60ns pulse of 1, then returns to 0. When we release the switch (long after the 60 ns pulse is finished), the upper input immediately goes back to 1, and 60ns later the lower input goes to 0. But none of this changes the output, since at any time one of the inputs is 1. A circuit that generates a short pulse like this is called a “one shot”. Single IC’s that perform this function are actually available and are discussed in H & H p. 517 ff.

1) Add in the NOR and three NOTs between the switch and flip-flop. The NOR is a 74LS04 and the NOT is a 74LS02. You, of course, need the data sheet for these parts. Do not forget to give them 5V and ground! Observe on the scope that the capacitor will now discharge even if you hold down the switch.

Part E: Latch and Display Section

Here we need to create a human readable display of our pulse counting. This will involve a *seven segment display*, a *display driver*, and a *latch*.

1. Display and Driver

The MAN71A is a seven-segment LED display used for displaying a single human-readable number or other character. It contains seven bar-shaped LED's arranged in the shape of a squared-off “8”. The “anode” (+ terminal in forward bias) connections of all the LED's are massed together into a single “common anode” terminal. The cathode of each segment has its own wire coming out of the DIP package. The segments are designated with letters *a – g*. To light any one or more of the segments simply ground the corresponding pin(s) on the DIP.

To display a given digit one has to take the desired digit and “decode” it into a list of which pins are to be grounded. The IC chip 7447 contains all the circuitry needed to do this decoding. As its input it receives the (four) binary digits of a number between 0 and 9, and “decodes” this information into which LED segments should light, grounding the corresponding pins of its output.

Look up the datasheets for the 7447 and MAN71A. Note that the MAN71A LED's are not internally overcurrent-protected, so a 500Ω resistor is required on each segment's line to ground, meaning between the MAN71A and the 7447. [Why wouldn't it be a good idea to put one such resistor in the common anode line?]

2. Latch

The latch allows us to statically display the final result of one capacitor charging cycle operation, even while the next cycle is in progress.

The latch we use here, the 74LS175, has four inputs and four outputs. It just passes along the logic states on its inputs through to its outputs- but only when an ENABLE signal is driven to logic 1. Each time an ENABLE=1 signal is received, the 74LS175 transfers its input data to its outputs, whereupon it freezes the outputs until another ENABLE signal arrives.

Obtain the datasheet for the 74LS175. You will see that fortunately for us, the ENABLE (also sometimes called CLOCK) input is positive edge triggered, meaning the chip only passes the input bits to the outputs on the CLOCK transition from 0 to 5V. No worries about multiple pass-alongs from one overlong CLOCK pulse, and no need for a homemade one-shot like with

the 74LS74. The freeze-the-outputs-till-the-next-CLOCK-pulse action is just what we need for our static display.

- 1) In the circuit from part B, connect the four output bits from the counter to the inputs of the latch. (You can leave the indicator LEDs connected too if you wish).
- 2) Run the output of the 74LS175 to the 74LS47 decoder for the MAN71A. Make sure to connect the LSB output of the counter to the LSB input of the decoder, and not LSB to MSB or something else.
- 3) Connect the COUNT UP pin of the counter to the TTL output of the function generator and set the repetition rate for about 1 Hz using your oscilloscope. Set up a debounced switch with pull-up resistor so that actuating the switch presents a logic 1 signal to the CLOCK input of the 74LS175 latch.
- 4) Verify operation of the circuit: No number should appear on the display until you hit the switch, “enabling” the latch. No matter that the function generator is running, no change on the display should then occur until you actuate the switch again. Press the switch twice in succession 5 or 10 seconds apart and verify that the change in the number displayed agrees with what you expect from the TTL pulse rate and the time interval between button-presses. To avoid having to subtract the starting and ending values, you could present a logic 1 signal to the CLEAR input of the 74LS192 counter after each count, using yet another switch-and-pull-up. The display will of course have to be cleared like this in the final circuit after each charging cycle is counted and displayed.

Part F: Completing the DVM

To complete the DVM we finally connect the analog input/transduction section, the clock generator, the counting section, and the latch-and-display section. See the marked-up drawing at the end of this writeup.

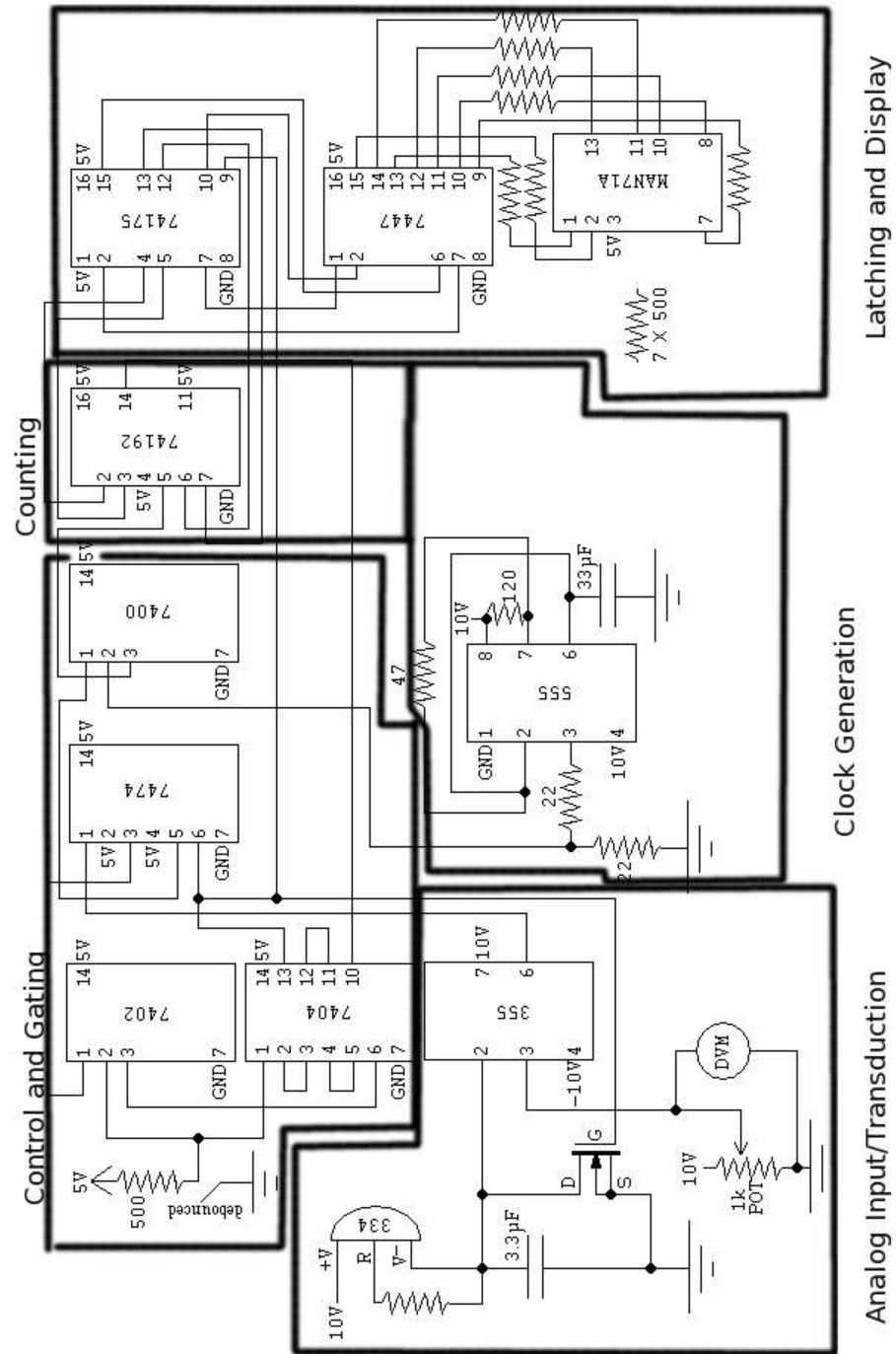
- 1) Reconnect a debounced switch as the input to the one-shot going to the 74LS74 CLOCK input.
- 2) Remove all the other manual switch-and-pull-up setups from the circuits. Connect the Q output from the 74LS74 and also the 555 timer output to the two inputs of the NAND. This will pass along the 555 pulse train only during the capacitor charging phase.
- 3) Connect the output of the NAND to the COUNT UP of the 74LS192 counter. The counter will count the pulses that come during the capacitor charging phase.
- 4) Connect the \bar{Q} output of the 74LS74 to the ENABLE (or CLOCK) pin of the 74LS175 latch. This will cause the result of the count to be latched into the output bits of the 74LS175 when \bar{Q} goes high at the conclusion of each capacitor charging phase.
- 5) In addition to latching the count, it is also necessary to clear the counter each cycle. This could be done simply by connecting the 74LS74 \bar{Q} output to the CLEAR pin of the 74LS192 counter. But this would not quite work, because \bar{Q} is also doing the latching. We couldn't be sure that the data was latched and displayed before the counter was cleared and the information lost. This type of timing condition is called a *logic race*. It is simple to avoid - connect the \bar{Q} to CLEAR after running it through two 74LS02 inverters (NOT gates) to delay it by about 20 ns.
- 6) At last, make a final check with the oscilloscope: connect the capacitor voltage to one channel on the scope, and the pulse train from the NAND to the second channel. Set V_{IN} to say 3 volts with the pot and then start a measuring cycle using the debounced switch on the 74LS74 CLOCK input. Visually count the number of pulses you see during the capacitor charging time. Vary V_{IN} and get an idea of the number of pulses per volt of V_{IN} (if you followed instructions,

it should be around 1).

7) Let her run! Take several readings each for several values of V_{IN} . Make a table of V_{IN} vs. displayed counts. Demonstrate that the relationship is linear (there will be some error due to the counting uncertainty of 1 pulse and the slow pulse rate chosen for a 1-digit display.). Demonstrate that the measurements are repeatable (measure, say 1V, then 8V, then 1V again and see if you get the result same as the first time). If any of these things are false, explain why (maybe a circuit element has heated due to dissipative power and change operating characteristics?).

Part Final+ extended lab (optional)

Only having a one digit display is a bit of a drawback, especially since we see some variation in this digit. You can easily expand it to be a two digit display by sending the CARRY output of the 74LS192 counter to a second identical latching and display section. Also, with two digits you can count 100 pulses during the charging time rather than only 10 with a single digit. Therefore the clock rate generated by the 555 should be increased by a factor of roughly ten. If you do this part, make another graph of V_{IN} vs. counts to check the range and linearity, and verify the reproducibility.



Full circuit with sections outlined and named.

