Problem #1 15 Points



The input to your circuit should be a 1Mhz Square Wave.

The first thing to notice is that the period of the repeating waveform is 12μ S. This will require a modulo 12 counter. While there are many ways to achieve this, probably the simplest is to use the synchronous load capabilities of the 161 (or the synchronous reset of the 163). This will require defining logic to recognize the state of 11, since the counter should count 0-11, then repeat. Eleven is a state of 1011 on Q₀-Q₃. Having done that we need to define logic that will produce a high output corresponding to the desired waveform. Recognizing any of the many combinations of counts that would give the desired relationship could do this. The easiest to use is simply 0,1 and 6,7,8. To find the simplest logic to recognize these, make a Karnaugh map.



From this, we can read off the expressions for the output: $\overline{(Q0 * Q1 * Q2)} + (\overline{Q3} * \overline{Q2} * \overline{Q1}) + (\overline{Q3} * Q2 * Q1)$. We can implement this in the obvious way, as shown below.



However, this solution has a subtle problem (one I did not catch myself until I looked closely (I don't expect you to have caught it)). As the count moves from 7 to 8, the true output moves between the min-terms A ($\overline{Q3} * Q2 * Q1$) and B ($\overline{Q3} * \overline{Q2} * \overline{Q1} * \overline{Q1}$).

The subtlety has to do with the fact that the first min-term has True versions of Q2 & Q1, while the second minterm has inverted versions of all of its inputs. 1 'LS11 propagation delay after the count changes the minterm from gate A will go false. The minterm from gate B will not go true until 1 'LS04 propagation delay later. As a result, the output will have a glitch in it every time the count transitions from 7 to 8. The problem is that we are moving between minterms with different propagation delays. We could fix the problem by modifying one of the minterms so that its propagation delay matched that of the other. Here is a good place to apply DeMorgans Theorem. We can take $(\overline{Q3} * \overline{Q2} * \overline{Q1})$ and transform it into $(\overline{Q3} + Q2 + Q1)$. This is a NOR gate with True inputs so now as the A gate output disappears, the B gate output will appear, and we eliminate the glitch.



Comments on your solutions:

Many people tried to make this much more complicated than it needed to be. I had thought that this was a straight-forward extension of the waveform generation problem that we worked in class. All that was needed was to recognize two sets of states to generate the output, rather than the one of the example.

The most common problems were:

1) not using the synchronous pre-load

2) Karnaugh map problems with not recognizing wrap-around pairs. This didn't cause wrong answers, just expressions that could have been simplified further.

Other than that, most of the real problems arose in the course of executing those one-of-a-kind solutions, and as such were not "common" to any other person's solution.

The mean score on this problem was 12.2 out of 15, with 38% of the class getting full credit for the problem

Problem #2 10 Points

Given the output of a 74C04 as a control, design the necessary circuitry to power a load with a resistance of 14.5Ω and requiring at least 1A to actuate when the output of the 74C04 is high. You have a 15V supply available.

The 74C04 has so little current drive capability, especially in the high state, that we are going to need a **LOT** of current gain $\left(\frac{1A}{1}\right) = 2.778$ requiring multiple bi-polar transistors) or we could use a voltage controlled device (i.e. MOSEET). This is

 $\left(\frac{1A}{360m} = 2,778\right)$, requiring multiple bi-polar transistors) or, we could use a voltage controlled device (i.e. MOSFET). This is

clearly the better solution.



With its very low ON resistance, the MOSFET will represent less than 50mV drop at 1A, making most of the 15V available for the load. This will be more than enough to meet the minimum 1A current through the load.

Comments on your solutions:

Far and away the most common mistake was to use the short circuit output current (1.75mA) for the 'C04. In this context, this is a useless spec. If we are trying to use a bi-polar transistor, we need both a certain amount of voltage as well as current. The output short circuit current is spec'd. with the output at 0 Volts.

The mean score on this problem was 6.3 out of 10, 39% of the class got full credit for this problem.

Problem #3 15 Points

You have been asked to size the pull-down resistor that will be placed on the input of a 74HC14. The pull-down resistor should guarantee a logic low input to the 74HC14 when there is nothing else connected to the 'HC14 input. From time to time, there will an 'HC14 output connected to this input, under those conditions, your pull-down resistor should not add more than 50μ A to the input current required by the 'HC14.

The situation described can be modeled as

Our task is to size R1. When the switch is open R1 must be small enough to allow the input current from the right hand 74HC14 to pass to ground without creating so much drop that it won't be interpreted as a low. For the 74HC14, that means that we are interested in the minimum negative going threshold. The voltage drop must be less than V_{T} .so

 $R_{\text{max}} = \frac{V_{T-}}{I_{in}} = \frac{0.9V}{1\text{ mA}} = 900k\Omega$ (I used the maximum input current over temperature). When the switch is closed, the resistor

should not add more than 50μ A to the amount that the right hand 'HC14 has to provide. For this we need to know what the output voltage will be from the 'HC14 when it is driving 50μ A + the Iih (1 μ A) of the 74HC14. The data sheet provides two sets of output voltage spec's., one at -20μ A ouput current and one at -4mA output current. If we are going to limit the load to 51μ A, we can't use the spec's at -20μ A, so we choose the voltage spec. at 4mA. The minimum value for the resistor will then be

$$R_{\min} = \frac{V_{oh}}{50 \text{ mA}} = \frac{3.84 V}{50 \text{ mA}} = 76800 \Omega$$
. Technically, any value in between these extremes will satisfy the requirements.

However, the maximum value is outside our guidelines values for typical resistor values (a few hundred Ω to a few hundred $k\Omega$) so I would choose the next largest standard value above the minimum, $82k\Omega$.

Comments on your solutions:

Many people used 5 V rather than V_{oh} for the driving device. Some of you quoted a V_{il} spec. rather than V_{t} which leads me to believe that you were not reading the 74HC14 data sheet.

The mean score on this problem was 7.6 out of 10, 30% of the class got full credit for this problem.

Problem #4 15 Points

You have a signal of interest that is a sine wave at 16kHz of 1V (pk-pk) amplitude, centered at 0V. However, it is corrupted by 50Hz noise (superimposed) with an amplitude of 100mV. Design the simplest possible filter that will reduce the noise level to no more than 5mV, while introducing no more than 3% loss in the 16kHz signal. The output should also be shifted to be centered around 2.5V.

The first thing to recognize here is that what we need is a high pass filter that will pass the 16kHz, while attenuating the 50Hz noise. There are at least three possible approaches to designing the filter:

- 1) Start with the desired attenuation of the50Hz and then check for the signal loss
- 2) Start with the loss limit and then check the noise signal for meeting the attenuation spec.
- 3) Calculate the Min & Max RC products to meet the signal loss & attenuation specs., check to be sure that there is a range between.

I chose approach #3 which encompasses #1 & #2.

We can use the standard expression relating V_{in} to V_{out} for a high-pass filter to calculate the required time constant for the

filter.
$$V_{out} = \frac{2pfRC}{[1 + (2pfRC)^2]^{\frac{1}{2}}} V_{in}$$

at 50Hz, $V_{in} = 100mV$ & V_{out} must be $\leq 5mV$; at 16kHz, $V_{in} = 1V$ & V_{out} must be $\geq 0.97V$

For this to be useful, you need to rearrange to be able to solve for RC. Start by expressing things in terms of Vout/Vin:

$$\frac{V_{out}}{V_{in}} = \frac{2\mathbf{p}fRC}{\left[1 + (2\mathbf{p}fRC)^2\right]^{\frac{1}{2}}}$$

to simplify the algebra, substitute X for the Vout/Vin ratio, A for 2pf, B for RC, then get rid of the square root in the denominator:

$$X^{2} = \frac{(AB)^{2}}{1 + (AB)^{2}}$$

re-arranging to isolate B:

$$B^{2} = \frac{X^{2}}{A^{2}(1 - X^{2})}$$

re-placing the original components:

$$RC^{2} = \frac{\left(\frac{V_{out}}{V_{in}}\right)^{2}}{\left(2pf\right)^{2}\left(1 - \left(\frac{V_{out}}{V_{in}}\right)^{2}\right)}$$
$$RC = \sqrt{\frac{\left(\frac{V_{out}}{V_{in}}\right)^{2}}{\left(2pf\right)^{2}\left(1 - \left(\frac{V_{out}}{V_{in}}\right)^{2}\right)}}$$

Now we have an expression into which we can plug the desired performance and determine the required RC. At 50 Hz we can calculate the maximum RC that will give us the required attenuation:

$$RC = \sqrt{\frac{(\frac{5mV}{100mV})^2}{(2\mathbf{p}\,50Hz)^2(1-(\frac{5mV}{100mV})^2)}} = 1.6x10^{-4}\,S$$

At 16kHz we can calculate the minimum RC that will give us no more the 3% loss:

$$RC = \sqrt{\frac{\left(\frac{0.97V}{1V}\right)^2}{\left(2\mathbf{p}16kHz\right)^2\left(1 - \left(\frac{0.97V}{1V}\right)^2\right)}} = 4.0x10^{-5}S$$

Any RC that falls between these values will meet the frequency response specifications.

The offset from a 0 reference to a 2.5V reference is easily accomplished using a configuration that you saw in the Lab 0 as well as in lecture:



In order to use standard values, I chose to use an RC of $1.5x10^{-4} S$. Note: the R in this is expression is the R_{eq} of the voltage divider resistors that give the DC offset.

Comments on your solutions:

Many of you tacked on voltage dividers to your high pass filters, without considering the effects on the corner frequency. Another common failure mode was to start with the complex impedances (not necessarily bad), but then got lost in the math.

Many of you tried things with diodes, like these:



this gives an output centered about 5V



The mean score on this problem was 10.5 out of 15, 35% of the class got full credit for this problem.

Problem #5 10 Points



Given an LTR-3208E photo transistor, design a circuit the will provide a clean logic level 0 to logic level 1 transition as the light level changes from 0 to 1mW/cm² respectively. Show calculations and call out relevant specifications.

Here we need to use the phototransistor to change the input voltage to a Schmitt trigger input. We need a Schmitt trigger, because of the spec for a clean logic level transition and the slow speed of rise & fall of the phototransistor. Since the simplest Schmitt trigger device is an inverter, we need to pull the input low with increasing light in order to make the output go high.



What remains is to size the pull-up resistor. The minimum value for the pull-up is determined by the maximum current that we can expect the phototransistor to sink. From the data sheet we can see that the phototransistor will be saturated at 0.5mA I_{ce} with 0.5mW/cm² illumination. Under these conditions, the V_{ce} will be 0.4V, a legal low for all logic device that we have studied. Since we have more than 0.5mW/cm² illumination, if we limit the current through the phototransistor to 0.5mA, it should be saturated. That 0.5mA will be coming from the input low I_{il} of the Schmitt trigger and the pull-up resistor. Using the data from a 74HC14 (others may work as well) we see that the worst case input low current is 1µA, leaving essentially the entire 0.5mA to

come from the resistor. The resistor may be as small as $\frac{5-0.4}{.5mA-1mA} = 9200\Omega$. In the light off state, the pull-up will need to

supply both the input high current, I_{ih} , to the Schmitt trigger as well as the dark leakage current through the phototransistor. Under these conditions, the voltage drop across the pull-up must be low enough to guarantee a legal high to the Schmitt trigger. The worst case input high voltage required by the 74HC14 is 3.15V, so with 5-3.15 = 1.85V across the resistor Iih $(1.\mu A) + I_{CEO}$

(0.1µA) should flow. The maximum value for the resistor is $\frac{5-3.15}{1\mathbf{m}A+0.1\mathbf{m}A} = 1681818\Omega$. This is larger than the values that

we would like to be working with, so I would once again choose the next larger standard size above the minimum (1k Ω), though any value between the two calculated extremes will work.

Comments on your solutions:

Some people used 1 mA as the saturated current instead of 0.5mA, they were probably looking at the spec for current as a function of light, failing to notice that those specs apply at a Vce of 5V.

-Other folks failed to calculate the bounds of the pull-up resistor (instead, lots of people picked a resistor and then showed that it worked on one of the two bounds)

Forgetting leakage currents or not using the correct leakage currents.

Not using a Schmitt trigger input device for the clean edge.

The mean score on this problem was5 out of 10, only 3 people got full credit for this problem.

Problem #6 10 Points



Design a circuit that will produce a 0.5S high pulse for every input pulse like that shown below:



These pluses will occur at random intervals, but never separated by less than 0.5S.

This is the perfect place for the mono-stable configuration of a 555. The delay is calculated as 1.1RC, so RC=0.45, if C=2.2 μ F, then R= 204545 Ω , choosing the closest standard value of 200k Ω , yields a pulse width of $1.1(200k\Omega*2.2\mu$ F)=0.484S. I chose to go down to 200k rather than up to 210k because, for proper operation, the next trigger to the 555 should not come before the prior pulse has completed. With a 210k Ω resistor, the pulse would have been slightly longer than 0.5S.



Comments on your solutions:

Some people calculated the delay as RC instead of 1.1RC. Some people tried to do it with counters and magical-mystical square waves out of thin air.

The mean score on this problem was 9 out of 10, 73% of the class got full credit for this problem.

Problem #7 25 Points

You have been asked to design the logic for a speed regulator. The input to your circuit will be a pulse train coming from the motor that is being controlled. The goal is to control the speed of the motor such that the pulse train is at approximately 100Hz. At a constant speed, this signal will be a square wave. Your regulator circuit should produce two outputs: Accelerate and Brake. If the speed is less than 96Hz, you should activate (set high) the Accelerate line and not the Brake Line. If the speed is at or above 120Hz, you should activate (set high) the Brake line and not the Accelerate Line. If the speed is within the range, neither signal should be active. Based on the inertia of the driven system, you have been told that the command signals must be updated at least 8 times a second and that the maximum acceleration/deceleration rate is 40Hz/Sec.. From another part of the system, a highly accurate 32.768kHz square wave is available for your use. Complete your design using HC family logic devices. Be sure to label all IC pins in your schematic with their functional label, not just pin numbers. **Note:** Don't get carried away with misplaced accuracy here. Take a look at the width of the acceptable speed range. Consider the impact of the conditions that you are evaluating. For instance, if you are at 96Hz and you fail to remove the accelerate signal: no big deal, you will continue to accelerate for another 1/8sec. taking you to 101Hz, where you should certainly remove the accelerate signal. **Remember:** start with blocks identifying the functions that you need.

The spec. says that we need to evaluate the speed every 1/8sec., so we'll need a block to trigger the evaluation. We'll also need a counter to count the speed pulses during the $1/8^{\text{th}}$ S intervals. Then we'll need some logic to decide to accelerate, brake or do nothing, a latch to capture & hold the brake & accelerate signals until the next evaluation and finally, a way to reset the counter to start counting again for the next $1/8^{\text{th}}$ S.



The over-speed logic logic is easy, since it the count of 15, and TC goes high on a count of 15 The under-speed logic is any count less than 12. Using a Karnaugh map, I reduced this to $\overline{Q_3} \oplus (Q_3 \bullet \overline{Q_2})$





The 74HC404 divides down the 32.768kHz signal to get us an 8Hz signal. The 'HC74 + 'HC08 next to it act as a rising edge detector (just like we did in class). In this case, we can predict the width of the pulse, it will be the clock period minus the propagation delay through the '4040. This signal is used to latch the evaluation logic & an inverted version resets the 'HC161. Between resets, the 'HC161 counts motor pulses. The lower bits of logic detect the <12 condition needed to activate Accelerate. The TC connection to PE, with the Ds tied high acts as an over-speed latch. If the count ever gets to 15 (over-speed) it asserts the PE, which on the next clock loads 15, keeping TC set. MR overrides all other inputs, so TC gets held 'till the next reset pulse. The remaining 'HC74s latch the Brake and Accelerate signals to hold them between reset pulses.

Comments on your solutions:

Many of you made this problem, as with problem #1, much more difficult than it needed to be. The most successful at this decided to update at the motor pulse rate, rather than $1/8^{th}$ sec. and set about counting the 32kHz pulses while the motor was high, or between rising edges of the motor signal. This can work, but it is more complex because you need eight bits of counter to count that number of pulses. More complexity = more chances for error

As an alternative to the '4040, many of you strung together 3 '161s, that worked just fine for generating the 8Hz update clock. Many of you included logic to catch the counter roll-over that resulted from as over-speed condition. I gave 2 bonus points for getting this right, no points taken off if it was omitted or didn't work.

I ignored the fact that a large number of you failed to show pin numbers on your schematics. As a building or debugging tool, schematics without pin numbers are useless.

Common Mistakes

Forgetting to reset the '161 Not latching the outputs Getting the counts wrong Trying to use the sync load to do reset. If you do this, you can't use a short reset pulse, since you can't guarantee overlap between the reset pulse & the motor signal. This would be especially true at low motor speeds. Putting a square wave into the '161 reset only allows it to count ½ the time. 2 outputs tied together. Not using HC logic as specified. Not evaluating every 1/8th sec as specified

The mean score on this problem was 17.7 out of 25, 17% of the class got full credit for this problem.