CS 250 Lab 3 – Sequential circuit for a traffic light

**Demonstration due in lab today:**

1. [25 points] Demonstration to your TA of your correctly functioning microcontroller circuit built from the 555, 74163, and 74138 chips.

**Due at the beginning of your next lab session:**

Type in your answers to the following questions. For the schematic diagrams, you may either hand draw or use schematic drawing software of your choice. Upload your answers to Blackboard in the form of a PDF file.

Questions

1. [10 points] Does the timing of the 555 chip output match the predictions of the equations for frequency, t\_high, and t\_low that are given below the schematic? Compute the predicted values and comment.

Equations for frequency are

F = 1 / (ln(2) \* C2\* (R1 + 2\*R2))

T\_high = ln(2)\*(R1+R2)\*C2

T\_low = ln(2)\*R2\*C2

Values of C1,C2,R1,R2

C1 = 100μF

C2 = 1000μF

R1 = 470Ω

R2 = 470Ω

0.000001 sec = 1μF\*Ω

1 sec = 1000000μF\*Ω

F = 1/(ln(2)\*1000μF\*(470Ω+2\*470Ω))

= 1/(ln(2)\*1000\*(470+940Ω)) = 1/(0.6931\*1410000μF\*Ω)

= 1/0.977271 = 1.023257623 Hz

T\_high = ln(2) \* (470Ω + 470Ω) \* 1000μF

= 0.6931 \* (940Ω) \* 1000μF = 651514μF \* Ω

= 0.651514

1. [10 points] The 10,000 ohm and 470 ohm resistors in the lab kit are rated to dissipate up to 1/4 watt. How much more than enough to use with the 555 chip is ¼ watt?

Power = (voltage)2/resistance

1. 25/10000 = 0.0025 Watts
2. 25/470 = 0.53191489 Watts

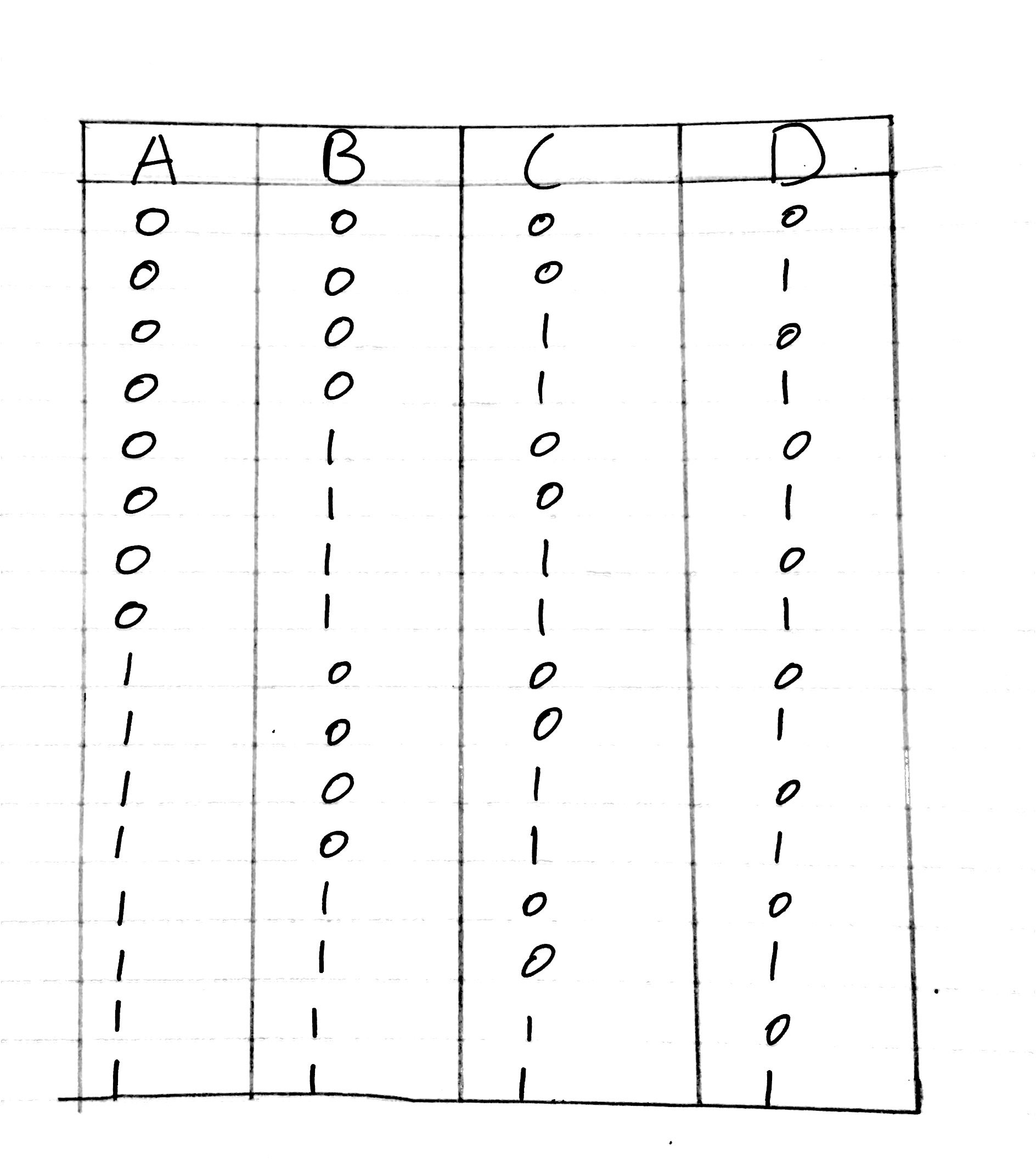
power needed for 555 chip

0.25 - 0.0025 = 0.2475 Watts

0.25 - 0.53191489 = 0.196808511 Watts

Extra resistance is = 0.2475 + 0.196808511 = 0.444308511

1. [10 points] Which three of the 74163 output signals could you select to send to the 74138 decoder so that each output of the decoder is selected in turn, modulo 8? Explain, covering all possibilities.



When we remove A = 0 and use B, C, D. This way the circuit will become modulo 8. Truth table above shows this

1. [10 points] For your traffic light design, write the Boolean expression for each traffic light color, and simplify to 2-input NAND and/or NOR gates.

Truth table for traffic light colors

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| A | B | C | Green color | Red color |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 0 | 1 |

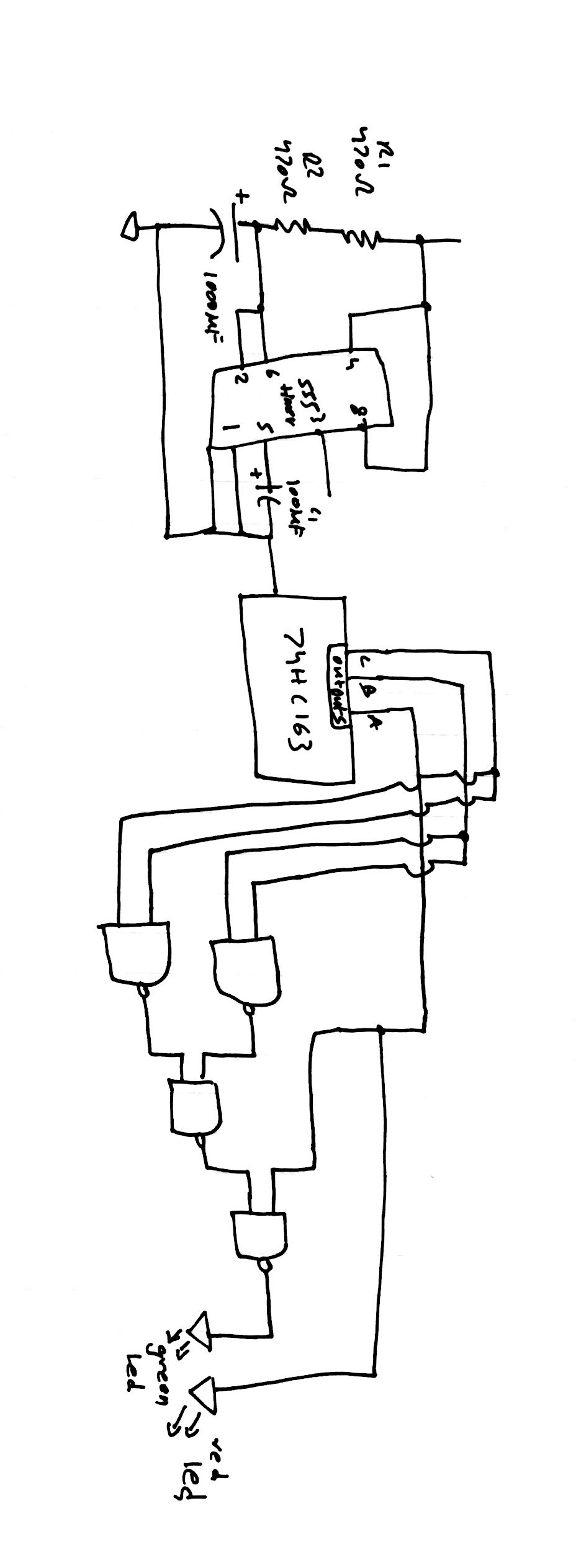
Using the truth table derived this expression

((A’+B’C’)’)’

((A’)’(B’C’)’)’

((A)(B’C’)’)’

1. [10 points] Draw the schematic diagram of your traffic light circuits, following the expression you just derived.



Demonstration due at the start of lab next week:

1. [25 points] Demonstrate your traffic light circuit to your TA at the beginning of lab next week.

**Goal**

Design and build a sequential circuit that uses time as an input, computes, and points.

In this lab three chips will be combined to build a microcontroller function for a traffic light:

a clock, a binary counter, and a demultiplexer (configured as a decoder).

**Part 1 In-Lab Assignment:**

Build the microcontroller and demonstrate correct operation to your TA

For the in-lab part of this lab you will to build a circuit that combines three key functions within a computer: Clock, Counter, and Decoder. A clock is the key to controlling behavior that changes with time. A counter generates a sequence of numeric values modulo some base. Most typically the base is a power of 2, such as 2^4 = 16 or 2^8 = 256, but counters modulo 10 are also useful often enough to be available pre-made in a chip rather than requiring us to design our own circuit. The counter output will be sent to a decoder, which takes k inputs from the counter and then asserts the one of its 2^k outputs that corresponds to the binary value on the k input lines.

The lab kit contains three integrated circuit packaging each of these key functions: Clock (555 Timer), Binary counter (74HC163), and Decoder/Demultiplexor (74HC138).

Step 1. 555 timer

Many digital circuits need a centralized pulse source, called a clock, to synchronize signals and events. The 555 timer is a widely used, general purpose circuit for timer, pulse generation, and oscillator applications. Read more at

<http://www.ehow.com/facts_5977976_555-timer-used-for_.html> and

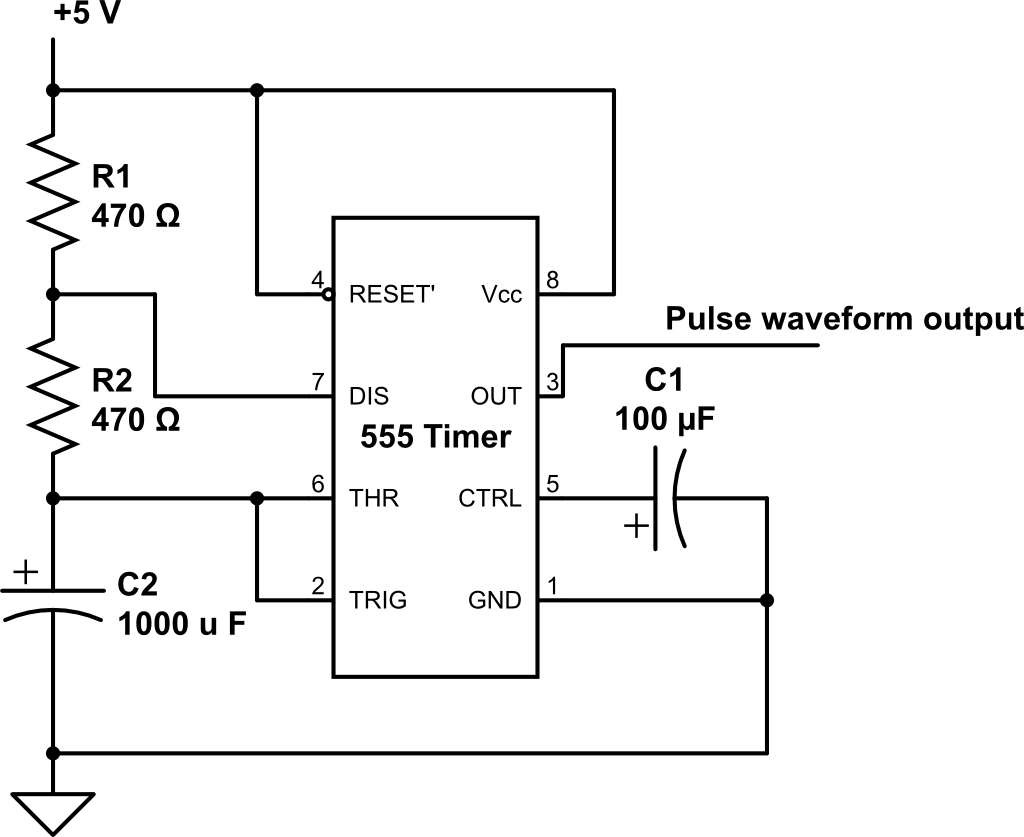
<http://en.wikipedia.org/wiki/555_timer_IC> . See an animation of 555 operation at

<http://www.falstad.com/circuit/e-555int.html> .

|  |  |
| --- | --- |
| Pin | Name and purpose |
| 1 | Ground connection |
| 2 | Trigger. When this input falls below 1/2 of the voltage at CTRL (Pin 5) then OUT (Pin 3) emits a rising edge and a timing interval starts |
| 3 | Output. This pin provides the clock signal output. |
| 4 | Reset. Active low. Timer is reset by driving this input to GND. Timing begins when RESET rises above approximately 0.7V. |
| 5 | Control. Provides control access to the internal voltage divider (by default, 2/3 Vcc). |
| 6 | Threshold setting. The timing interval ends when the voltage at THR is greater than CTRL. |
| 7 | Discharge. Open collector output which may discharge a capacitor between intervals. In phase with output.| |
| 8 | Vcc connection to positive voltage supply |

When configured for astable (clocking) operation the schematic for the 555 looks like this. Note carefully that the pin positions on the 555 symbol in the schematic are not in the same configuration as they are on the physical chip. This is to make the schematic have a simple, untangled arrangement of wires. The pins on the physical 555 chip are numbered 1 to 8 with 1 at the lower left corner with the chip upright, 4 at the lower right corner, 5 at the upper right, and 8 and the upper left.

**Use caution in connecting capacitors C1 and C2. They are polarized. Be sure the match the polarity orientation shown in the schematic.**



The values of the external components R1 and R2 in ohms and C2 in units of Farads control the waveform of the pulses emitted by the 555. The frequency f of the pulse stream generated by the 555 is given by the equation f = 1/( ln(2)\*C2\*(R1 + 2\*R2) ). The time that the signal is at a high voltage t\_high within each pulse is t\_high = ln(2)\*(R1 + R2)\*C2. And the low time from each pulse is t\_low = ln(2)\*R2\*C2.

The power capability (heat dissipation rating) of R1 must be greater than Vcc2 / R1 Watts. The smaller the physical size of a resistor, the smaller its heat dissipation capability.

Once building the 555 circuit is done, use an LED to probe the output of the 555 to check for correct operation

Step 2. Binary Counter

Next add the <http://www.futurlec.com/74HC/74HC163.shtml> 74163 binary counter to the microcontroller circuit. A pin diagram is available through the link.

|  |  |
| --- | --- |
| Name | Purpose |
| Power & Ground | As is common, Pin 16 is supply voltage (Vcc) and Pin 8 is Ground (zero reference, or GND). |
| Clock input | Pin 2 is the clock input pin.| |
| Data input | Pins 3, 4, 5, and 6 are data input pins corresponding to D0 to D3. Unused. Safe to leave unconnected. |
| Count enable | This pin provides the amount to increment the count by. If it is zero the count does not advance. If it is 1, then the count advances by one with each rising edge of the clock signal. This design also allows connecting one counter to the next. Connect Pin 7 to Vcc to enable the counting function. |
| Clear | An active low (active with a 0 value) input. Connect Pin 1 to +5 V (Vcc) to not clear the counter. |
| Flip-flop output | Pins 14, 13, 12, and 11 are the output pins C3, C2, C1, and C0, respectively. |
| Count enable carry input | Connect Pin 10 to Vcc to enable the carry input function. This is used to connect multiple counter chips into one larger counter. |
| Parallel enable input | Allows loading a count value from the data input pins. Connect Pin 9 to Vcc to disable this function. |

Connect the 555 output to the clock input of the 74HC163. Then wire the rest of the 74163 to enable counting. Probe the 74163 output signals using LEDs to confirm correct operation.

Step 3. Decoder/Demultiplexer

Finally, add the <http://pdf1.alldatasheet.com/datasheet-pdf/view/51038/FAIRCHILD/74138.html> 74138 decoder/demultiplexer to the circuit. A pin diagram is available through the link.

|  |  |
| --- | --- |
| Name | Purpose |
| Power pin | As usual, Pin 16 and Pin 8 correspond to the voltage supply (Vcc) and ground (GND). |
| Output pin | Pins 15 through 9 correspond to outputs “Y0” through “y6”. Pin 7 outputs “Y7”. |
| Enable pin | Pins 4, 5, and 6 are the enable pins an allow daisy-chaining this chip with one or three copies to make 4x16 and 5x32 decoders, respectively. To enable only 3x8 decode, set pins 4, 5, and 6 to LOW, LOW, and HIGH respectively. |
| Input pin | Pins 1 through 3 are the address pins to be decoded. Connect them to the counter output. |

The 74HC163 is a 4-bit counter, but the decoder input is a 3-bit address. Which three of the output signals could you select to send to the decoder so that each output of the decoder is selected in turn, modulo 8? Explain.

Use 8 red and green LEDs as probes of the output of the 74HC138 and verify correct circuit operation – each output goes low one at a time in turn. Show your TA.

**Part 2. Take-Home Lab Assignment**

Design and build a traffic light circuit that makes use of the micro-computational capability of the 555/74163/74138 three-chip controller circuit.

Traffic lights continuously cycle the illumination of three different colored lights: green, yellow, and red. Use green, RGB, and red LEDs to create the green, yellow, and red output. Use the RGB LED to display yellow light (roughly) by turning on both the red and green emitters. The output of your traffic signal must show green for 4 seconds, followed by yellow for 1 second, followed by red for 3 seconds, and then repeat in this order.

Design three combinatorial circuits, one to compute the signal for each traffic light color. The controller circuit provides eight outputs from the demultiplexer and three outputs from the counter. Any of these outputs may be helpful in developing the signals for the traffic light. You may also use NAND and NOR gates to implement your computational logic, as you see fit.

Use the steps you have learned in labs 1 and 2 to design your traffic light circuit. Create a truth table for each color of the traffic light. Use outputs from either the counter or decoder as inputs for the truth tables. Think about the operation of a traffic light and think about which of the counter or decoder outputs are similar in timing and logic level to the signal you need for a given traffic light. Using an existing signal that is similar to the LED signal needed for a given light means a simpler computation to perform to create the actual LED signal and, thus, a simpler combinatorial circuit to build.

Reminders:

- Double check that capacitors C1 and C2 are placed into the circuit with correct polarity.

- Double check that LEDs are placed with correct polarity. **Note, for active low outputs, the long LED lead may be connected to power and the short lead used to show a visual inversion of an active low output.**

- Use 470 ohm resistors in series with each color of the RGB LED. This LED will appreciate the reduction in current.

- To display YELLOW light, sort of, send a high voltage signal to both the \*\*RED pin\*\* and the \*\*GREEN pin\*\* of the RGB LED, because mixing red and green light yields yellow light. Because the red and green photons are emitted from separate regions within the RGB LED, this LED will look yellow only when viewing the red and green emitting regions align one behind the other along a line of sight. Placing a translucent cup over the RGB LED will mix the two light sources together before emitting the photons from the cup outer surface, making for a slightly better yellow experience. Your TA has translucent plastic cups; please return it after use for students in later lab sessions to use.