CS250 Summer 2017 Lab 03
Traffic Signal Light

Build a fixed logic processor (see Comer section 4.5). Three key chips will be used: a clock, binary counter, and decoder/demultiplexer.

1. The Clock → Counter → Decoder/Demultiplexer Circuit

In lab, you will build a circuit that combines three key building blocks of a computer: Clock, Counter, and Decoder/Demultiplexer. A clock is the key to incorporating behavior that changes with time. A counter generates a sequence of numeric values modulo its base. Most typically the base is a power of 2, such as 2^4 = 16 or 2^8 = 256. However, counters modulo 10 are useful often enough to be available pre-made in a chip rather than requiring us to design our own circuit. Counter output will be sent to a combination decoder/demultiplexer circuit, decoder/dmux for short, 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.

With a steady clock signal driving the counter, the counter will increment its output steadily, which in turn will drive the decoder/demux to assert each of its 2^k outputs in sequence. A common application of this circuit is to trigger the refresh of the information stored in every row of a Dynamic, Random-Access Memory (DRAM) chip. DRAM forms the main memory of most computers, but the information stored in every DRAM bit must be periodically read out and then re-written (refreshed) or it will fade away because the bunch of electrons stuffed into a DRAM storage cell to indicate storage of a 1 leak away from that storage cell quite quickly. At room temperatures, refresh must happen roughly every millisecond for every DRAM memory location to ensure sufficiently low probability of error due to stored 1 values leaking away enough to be mistaken for stored 0 values.

The lab kit contains three integrated circuits packaging each of these key functions: clock (555 timer), binary counter (74HC163), and decoder/demultiplexer (74HC138).

1.1 The 555 Timer

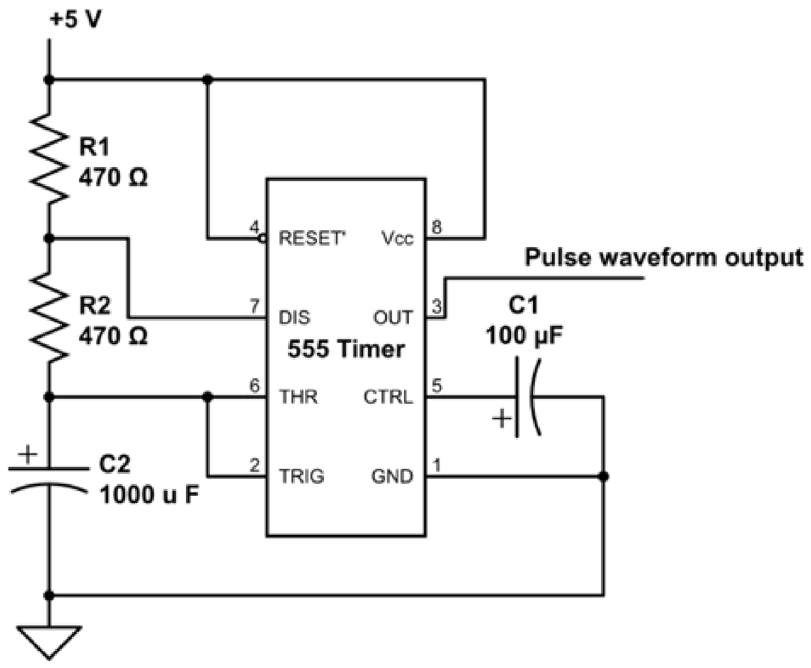
Many digital circuits need a centralized pulse source, called a clock, to synchronize signals and events. The 555 timer is a widely used, general purpose circuit for timer, pulse generation, and oscillator applications. In particular, the astable mode (oscillator) of the 555 produces a steady pulse waveform suitable for clocking digital circuits. Read more at

http://www.ehow.com/facts_5977976_555-timer-used-for_.html
http://en.wikipedia.org/wiki/555_timer_IC

See an animation of 555 operation at http://www.falstad.com/circuit/e-555int.html

Place the 555 timer on your breadboard and wire it as shown in the schematic below. Use caution in connecting capacitors C1 and C2. They are polarized. Be sure to match the polarity orientation shown in the schematic. Once assembly of the 555 circuit is done, use one of your LEDs to probe the output of the 555 to ensure correct operation.

When configured for astable operation the schematic for the 555 looks like below. Note carefully that the pin positions on the 555 box icon in the schematic are not in the same configuration as they are on your 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 at the upper left.



Pin	Name	Purpose
1	GND	Ground connection.
2	TRIG	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	OUT	The output pin. Source of the clock signal.
4	RESET	Active low. Timer is reset by driving this input to GND. Timing begins when RESET rises above approximately 0.7V.
5	CTRL	Provides control access to the internal voltage divider (by default, 2/3 Vcc).
6	THR	Threshold setting. The timing interval ends when the voltage at THR is greater than CTRL.
7	DIS	Open collector output which may discharge a capacitor between intervals. In phase with output.
8	Vcc	Connect to positive voltage supply.

The values of the external components R1 and R2, in ohms, and C2, in farads, control the waveform of the pulses emitted by the 555. The frequency, f, of the pulse stream generated by the 555 depends on the values of R1, R2, and C2:

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

The time 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:

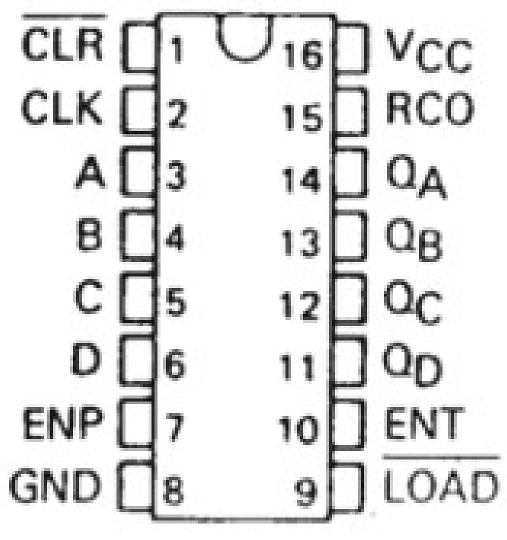
t_low = ln(2)*R2*C2

The power capability (heat dissipation rating) of R1 must be greater than V_cc^2 / R1 Watts. The smaller the resistor, the smaller its heat dissipation capability. The small resistors in the lab kit are rated to dissipate up to 1/4 Watt.

1.2 Binary Counter

Next we add the binary counter to the circuit. A data sheet is available through the link: http://www.futurlec.com/74HC/74HC163.shtml

The pin diagram for the chip is



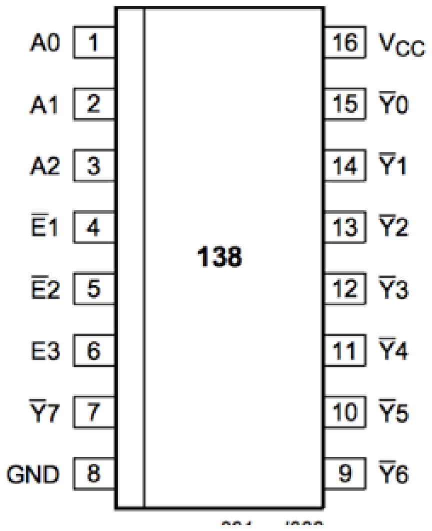
NAME	**PURPOSE**
POWER & GROUND	As usual, Pin 16 is supply voltage (Vcc) and Pin 8 is Ground (zero reference, or GND).
CLOCK INPUT	Pin 2 is the clock input pin.
DATA INPUT	Unused in this lab. Pins 3, 4, 5, and 6 are data input pins corresponding to A, B, C, D where A is the least significant bit and D the most significant. The chip circuit has been designed so that when these pins are disconnected (logic voltage levels undefined) the chip still operates correctly.
COUNT ENABLE (ENP)	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 (asserted when the value is logic 0) 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 QA, QB, QC, and QD, respectively, where QA is the least significant bit and QD the most.
COUNT ENABLE CARRY INPUT (ENT)	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 (LOAD)	Allows loading a count value from the data input pins 3 through 6. Connect Pin 9 to Vcc to disable this function.

Connect the 555 output to the clock input of the 74HC163 and then wire the rest of the 74163 to enable counting. Again, using LEDs to confirm correct operation, probe the three 74163 output signals that you have selected.

1.3 Decoder/Demultiplexer

Finally, we add the decoder/demultiplexer to the circuit. A data sheet is available at
http://www.nxp.com/documents/data_sheet/74HC_HCT138.pdf

The pin diagram is



NAME	PURPOSE
POWER PINS	Pin 16 is for the voltage supply (Vcc) and Pin 8 is for ground (zero volt reference, or GND).
OUTPUT PINS	Pins 15 to Pin 9 are active low outputs Y0 through Y6. Active low output Y7 comes from Pin 7.
ENABLE PINS	Pins 4, 5, 6 facilitate connecting this chip to one or two other 74138 chips to form 4-to-16 line and 5-to-32 line decoder/demultiplexers, respectively. With all three pins asserted (low voltage level applied to pins 4 and 5, high voltage applied to pin 6), the decoder behaves as expected: generating a low voltage at one output, Y0 through Y7, based on the unsigned int presented on inputs A0, A1, and A2.
INPUT PINS	Pins 1 to Pin 3 are connected to binary counter output pins.

1.4 Helpful 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, an LED can be placed from power (long LED lead) to the output pin to show a visual inversion of the active low output.
- Use 470 ohm resistors in series with each color of your RGB LED. This LED will burn out if the current is too large. The 470 ohm resistors limit the current to a tolerable level.
- To display YELLOW light, you should enable both the **RED pin** and the **GREEN pin** of your RGB LED.

1.5 Visualizing the Output

Place 8 of your red and green LEDs as probes of the output of the 74HC138 and verify correct circuit operation. **Show your TA this circuit when you complete it.**

2. Take-Home Assignment

Retaining the clock → counter → decoder/demux circuit already on your breadboard, you will now implement a traffic light fixed logic processor.

Traffic lights continuously cycle the illumination of three physically distinct colored lights: green, yellow, and red. The lights are physically distinct so that even someone who suffers from color blindness can tell which signal message is illuminated. Use a green LED, the RGB LED, and a red LED to create the green, yellow, and red traffic lights, respectively. The RGB LED will display yellow light if you equally turn on its red and green LEDs and observe the LED from a viewpoint where the tiny red and green light emission spots are aligned so that the colors blend together.

Specifications: The output of the traffic signal should be: green for 3 seconds, then yellow for 1 second, then red for 4 seconds. The order of color transition is important: the correct sequence, GYRG... (green, yellow, red, green, …), is not the same as, and is distinguishable from, the reverse sequence, RYGR… .

Use your knowledge gained from previous labs to design a traffic light circuit to meet the above specifications. There are several designs that fully satisfy the specifications and that can be built from the parts in one lab kit. The following steps will help you craft your design.

2.1. Create a truth table for each light in the traffic light.

These truth tables will be a little different from what has come before. The output of each truth table will control the operation of one of the traffic lights. The traffic lights turn on and off with coordinated timing to produce the specified sequence and duration of colors. The outputs of the clock, counter, and decoder/demux chips all transition in synchrony at a rate set by the clock chip output transitions. You can use this synchronized transition behavior to set the time of transitions between colors for the traffic signal.

Choose the inputs for each of the three traffic light truth tables independently and from among the signals provided by the outputs of the clock, the counter, and the decoder in any combination you wish. To help make the choice of inputs for each table, first draw a diagram of the second-by-second timing sequence of operation for each traffic light color. It is likely that you will want to diagram these three timelines so that the operation of any one light relative to the two others is obvious.

Make the same type of diagram for each output of the 555, 74163, and 74138 chips. Compare these timing diagrams with the three traffic light diagrams. Look for similarity between available chip output behavior and behavior of any one of the three traffic light colors. Finding a chip output signal that is an exact match for the timing diagram for a light means that a wire from that signal to the correct LED completes the design for that light. Finding a chip output with behavior in time similar to the signal needed to operate a given traffic light color means that the additional logic circuitry needed to perfect the LED drive signal should be simple. Once the three truth tables are complete, proceed to Step 2.

2.2. Write the Boolean expression for each traffic signal color.

Simplify it to use 2-input NAND and/or NOR gates.

2.3. Draw the schematic diagram for the circuit you have designed.

Answers to 2.1, 2.2, and 2.3 represent the answer to Question 3 on the Lab 03 assignment document.

2.4. Integrate your designed circuit with the existing circuit on your breadboard.

Be sure to test for correct traffic light operation. Demonstrate your circuit to your TA at the beginning of your Tuesday or Thursday lab section.

Here is video of a working traffic light fixed logic processor.

3. Due Dates

The due date for the take home assignment is:
For Monday/Wednesday Lab sessions: 10:58 am, 28th June.
For Tuesday/Thursday Lab session: 10:58 am, 29th June.