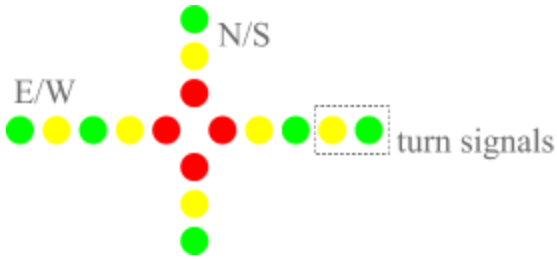
Major Project 2: Traffic Lights

Jonathan Lam
Professor Risbud
DLD Summer 2018

Abstract

The aim of the project is to control a group of LEDs featuring north/south (N/S) and east/west (E/W) directions, as well as green and yellow turn lights for the E/W directions. The layout of the LEDs and the timing of each combination of lights ("stage") is shown below (the value "X" represents a lit LED, and a blank cell represents an unlit LED). The stages are represented in BCD 0-7.



Stage (CBA)	Duration (s)	N/S			E/W				
		Red	Yellow	Green	Red	Yellow	Green	Yellow turn	Green turn
001	4	X			X				
010	21			X	X				
011	3		X		X				
100	2	X			X				
101	5	X			X				X
110	2	X			X			X	
111	18	X					X		
000	5	X				X			

The extra credit consists of two parts: a pair of two seven-segment displays, one for each direction; and an "accelerate red light" button that pedestrians can press to immediately make the green light in the other direction immediately go down to five seconds, but only if one direction (N/S or E/W) currently has a green light and the green light countdown is over five seconds. The seven segment displays should show display "FF" when their light is red or yellow; a countdown timer beginning when the green light has fifteen seconds remaining; and "EE" when the green light has more than fifteen seconds remaining. The countdown should also blink at a frequency between two and five hertz when the countdown is between seven and zero seconds. The specifications for this project dictate that the LEDs must light on a logical zero, no more than three breadboards be used (for full credit or in order to receive extra credit), and that wires must not overlap or be significantly bent.

The regular credit option was constructed for this project, but the design for the extra credit options are included below. The extra credit was not constructed because I could not find a solution to handle all of the requirements for both extra credit additions while following all the specifications.

Regular Credit Design Process and Final Design

Here is an overview of all of the major possibilities for the regular-credit part of the project: RYG-lights in two directions, and green and yellow turn signals in the E/W direction.

All of the following solutions include a 555 chip that produces a 1Hz astable clock signal. They also all use a "stage counter" that loops through the stages 000-111 (0-7). In every case, the three bits of the stage counter (S_C, S_B, and S_A) are used as address pins to a negative-logic demultiplexer (1:8 DEMUX with the COM pin tied to ground and all outputs pulled-up). Because the demultiplexer uses negative logic and most of the lights (except for the red lights) are only lit on one stage, those LEDs can be connected directly to the demultiplexer outputs. The difference between all of these designs is in the way the stage counter is triggered; i.e., what its clock signal input is.

This base design comprises four chips: a timer (555), stage counter (4024 or 4040), 1:8 demultiplexer (4051), and a two-input NAND chip (4011).

---

1. Counters with durations stored in RAM or shift registers.

The first idea for a design was to store all of the durations onto a RAM chip or shift registers. Depending on whether pure-binary or base-ten encoded numbers are used, each number would require five bits (binary 21 is 10101) or six bits (decade-mode 21 is 10 0001). The stored duration bits would set the duration for a presettable down-counter (e.g., a pair of 4029 four-bit presettable down counters in binary/decade mode linked together) whose clock was the 1Hz signal from the 555 timer. Once the pair of counters counts down to zero, the counters would send a high CLK signal to the stage counter, advance the shift register five or six bits (or advance the RAM one position), and then preset the countdown timer with the new value. It would also take additional chips (another timer and counter) to make the shift registers shift exactly five or six bits to get to the next stage's duration.

The benefit of this design is that if the specifications are changed, the stored values can easily be changed because they are hardcoded in. However, the problem with using RAM is that, while space-efficient, the duration data must be inputted serially, which would either require additional logic or be inputted manually. The opposite is true for using shift registers: while it is simple to hardcode in duration values, storing data for eight stage durations will be between 40 and 48 bits in length, which would require many shift registers and take up much space.

---

2. Counter with MUXes linked together.

The second idea was to use an up-counter whose clock would be the 1Hz signal from the 555 timer. The counter would reset when it reached 60, for the total duration of the traffic light loop. Then, some number of MUXes would use the counter's bits to select whether to increase the stage (1) or not (0).

This solution works by tying each input that corresponds to a second in time that should advance the state to 1, and the rest to 0. The only difference between the different solutions is the number of MUXes, and therefore the amount of pre-processing necessary. The seconds in time that should be connected to 1 are shown below, and all of the rest should be tied to 0. Finally, the base-design demultiplexer output from 000 should be connected to the RESET pin of the counter, causing the countdown to reset after a full cycle (60 seconds).

Stage (CBA)	Duration	Elapsed Time (time begins at 0)	Binary representation (FE DCBA)
001	4	3	00 0011
010	21	24	01 1000
011	3	27	01 1011
100	2	29	01 1101
101	5	34	10 0010
110	2	36	10 0100
111	18	54	11 0110
000	5	59	11 1011

Using eight MUXes, no logic would be needed, and each of the sixty input pins will be tied to GND or Vcc. Using four MUXes, "D-logic" (inverters only) would be needed for pre-processing. Using two MUXes, "DE-logic" is necessary for pre-processing, but this may take multiple logic chips (AND, OR, NAND, NOR, XOR, and/or NOT ICs). It is also possible to use one MUX and 8:1 pre-processing or zero MUXes and similar logic with only logic, but this is impractical because of the amount of logic necessary to map the six-bit Karnaugh map. Zhihao Wang helped worked through solutions with different numbers of MUXes.

While this MUX logic is simple and space-efficient using either two or four MUXes, it makes the extras difficult to integrate into the project because there is no downcounter that can easily be used to determine how much time is left in the stage, and therefore was not selected as the final design.

---

3. Map current stage bits to next stage's duration.

Like the first idea, this design involves pre-setting a count-down timer with the five- or six-bit duration of the next stage. However, unlike the first idea, it involves using logic with the current stage's bits to preset the duration of the next stage, instead of having hardcoded values. This generally uses fewer chips than the other two solutions, and can easily fit onto two breadboards. This idea was first suggested by Dan Kim.

This table shows a mapping using the six-bit base-ten representation (FE represent the tens-place digit; DCBA represent the ones-place digit).

Stage (CBA)	Duration (s)						
	Decimal	F	E	D	C	B	A
001	4	0	0	0	1	0	0
010	21	1	0	0	0	0	1

011	3	0	0	0	0	1	1
100	2	0	0	0	0	1	0
101	5	0	0	0	1	0	1
110	2	0	0	0	0	1	0
111	18	0	1	1	0	0	0
000	5	0	0	0	1	0	1

3a. Using basic logic gates to map bits to next stage's duration.

This is the more straightforward solution. The data inputs to the preset JAM inputs on the counters (T_F through T_A) can be written as functions of S_C, S_B, and S_A. This can be done the same was as creating any other three-input map: by creating Karnaugh maps for each output and creating reduced mathematical functions to implement. Because a demultiplexer will be used with only the correct stage corrected, it can be used to simplify functions that are only 1 for one stage (F, E, and D). D_{XXX} refers to an output of the demultiplexer. Here is one example set of expressions only requiring seven logic gates.

T_F: $\overline{D_{010}}$
T_E: $\overline{D_{111}}$
T_D: $\overline{D_{111}}$
T_C: $\overline{S_A + S_B} + (S_A \oplus S_B \oplus S_C).S_C$
T_B: $(S_B \oplus S_C).(S_A + S_B)$
T_A: $S_A \oplus S_B \oplus S_C$

3b. Using the DEMUX to map bits to next stage's duration.

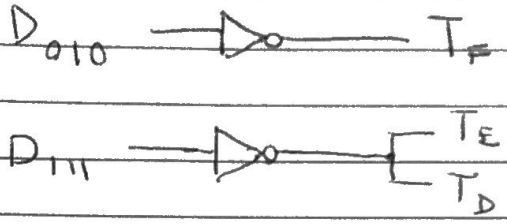
This solution uses the demultiplexer already included in the base solution to simplify logic without using additional logic gates, and is used in the final design. It only requires NOT and NAND gates for the logic. Using a decoder/demultiplexer to simplify the logic was also suggested by Dan Kim.

T_F: $\overline{D_{010}}$
T_E: $\overline{D_{111}}$
T_D: $\overline{D_{111}}$
T_C: $\overline{D_{000}.D_{100}.D_{111}}$
T_B: $\overline{D_{010}.D_{011}.D_{101}}$
T_A: $\overline{D_{001}.D_{010}.D_{100}.D_{111}}$

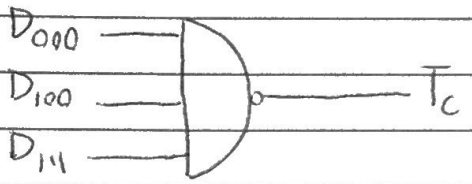
TRAFFIC LIGHT LOGIC DIAGRAMS

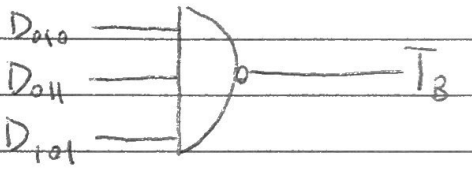
JONATHAN LAM

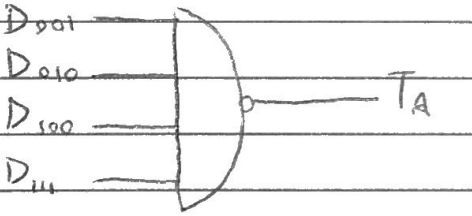
countdown timer JAM inputs ($T_F - T_A$)



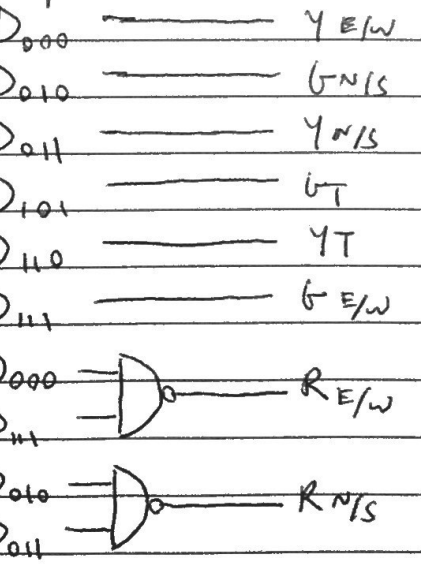
(D_{xxx} are demultiplexer outputs with negative logic)



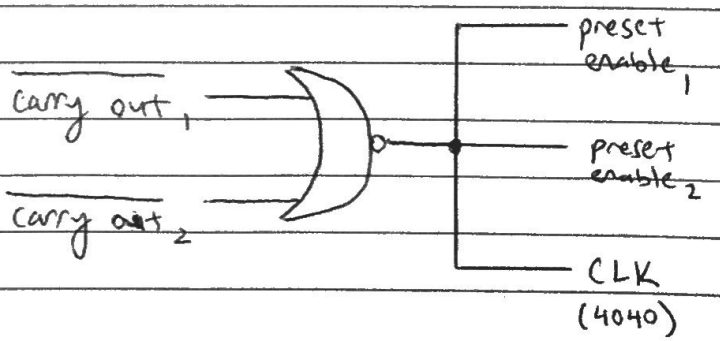




outputs (LEDs)

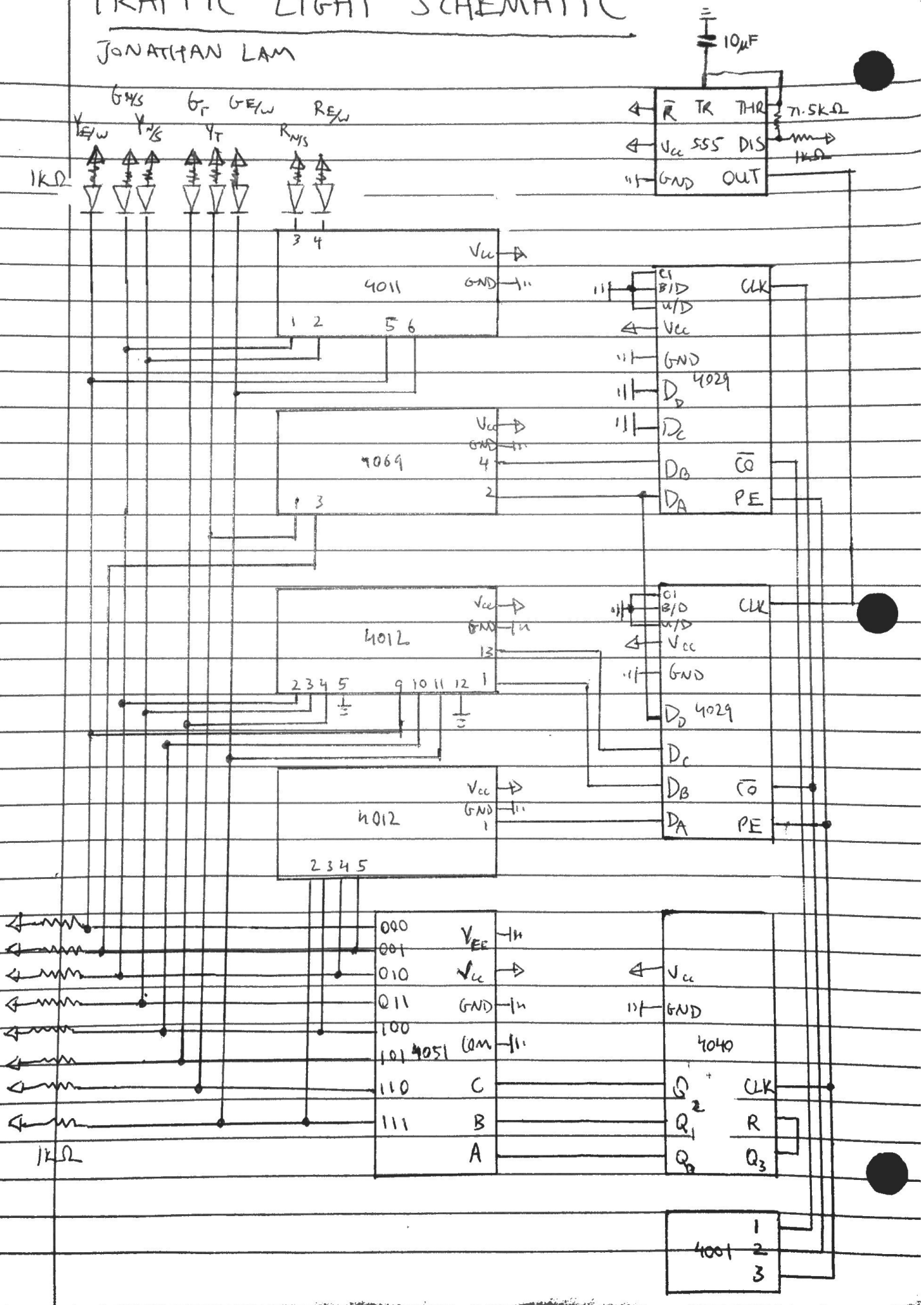


advance state clock



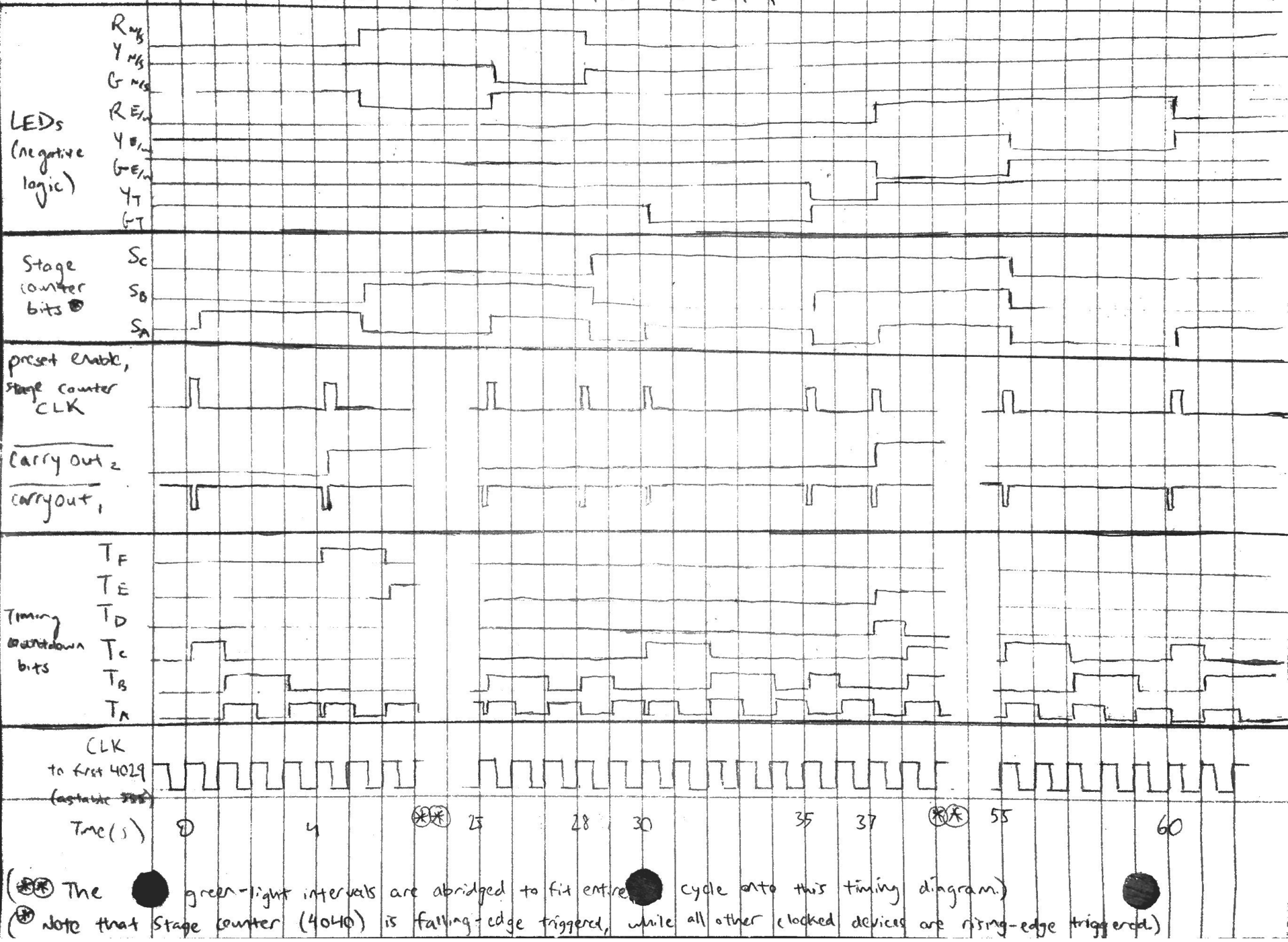
TRAFFIC LIGHT SCHEMATIC

JONATHAN LAM



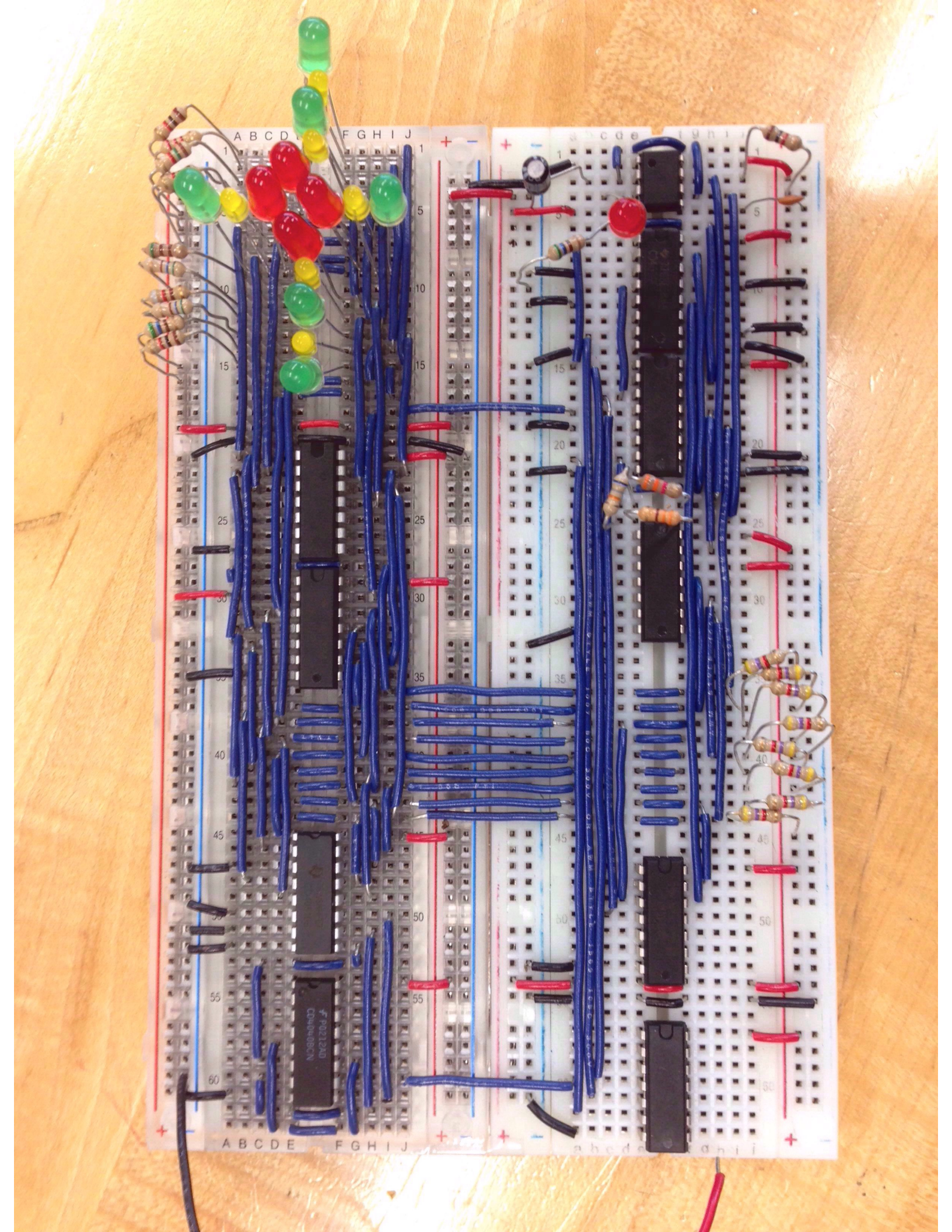
TIMING DIAGRAM FOR TRAFFIC LIGHT

JONATHAN LAM



(** The green-light intervals are abridged to fit entire cycle onto this timing diagram.)

(* Note that Stage counter (4040) is falling-edge triggered, while all other clocked devices are rising-edge triggered.)



Extra Credit Design

Rough drafts of logic diagrams of the extra credit circuits are attached after the descriptions.

Part One: Accelerate Pedestrian Button Circuit

This circuit is centered around a J/K flip flop whose output (Q) indicates whether or not the accelerate button can be pressed. Most of the time, Q is 0, which means that pressing the button will not do anything.

Q is set to 1 by sending a high pulse to J when one of the green-light stages (D_{010} or D_{101}) are entered. Q is set to 0 by sending a high pulse to K when the button is pressed or when the countdown bits EDCBA is 00101 (five seconds remaining).

If the button is pressed when Q is high, then the JAM inputs FEDCBA are preset to the value 00101 (five seconds remaining) using a 2:1 MUX (i.e., 4053), and a high pulse is set to the PRESET ENABLE pins on the downcounters (4029).

This circuit ensures that the remaining time for the current stage is set to five seconds only if the right conditions are met: if one direction has a green light that has over five seconds remaining.

Part Two: Pedestrian Timer Circuit

This circuit uses a J/K flip flop that indicates whether the seven-segment displays are displaying the remaining time or not, similar to the accelerate pedestrian button circuit.

Q is set to 1 by sending a high pulse to J when one of the green-light stages (D_{010} or D_{101}) is active and the countdown bits EDCBA are 10101 (this can re-use some of the gattes from the accelerate pedestrian button circuit, since the lower four bits are equal). Q is set to 0 by sending a high pulse to K when the downcounter reaches zero (both counter carry out pins are low).

A similar process is done with another J/K flip flop, but when EDCBA are 00111 (seven seconds remaining) for the timer to begin blinking at seven seconds. This output is tied to the RESET pin of a second 555 timer, which is set to astable mode between two and five hertz, whose output is connected to the BLANKING pin of the BCD seven-segment display driver (4543).

To make the seven-segment display show “EE”, the phase is inverted and the inputs to the BCD seven-segment display driver are set to 0001. (This is the number 1, but all of the LEDs are inverted; therefore, only the 1 is *not* lit, leaving behind an “E”. This trick was brought up by Nathaniel.) Similarly, the “F” can be displayed by inverting the phase, inputting 0001, and turning off the bottom display LED (segment “d”), which can be switched on and off using a 4:1 pair MUX (4052).

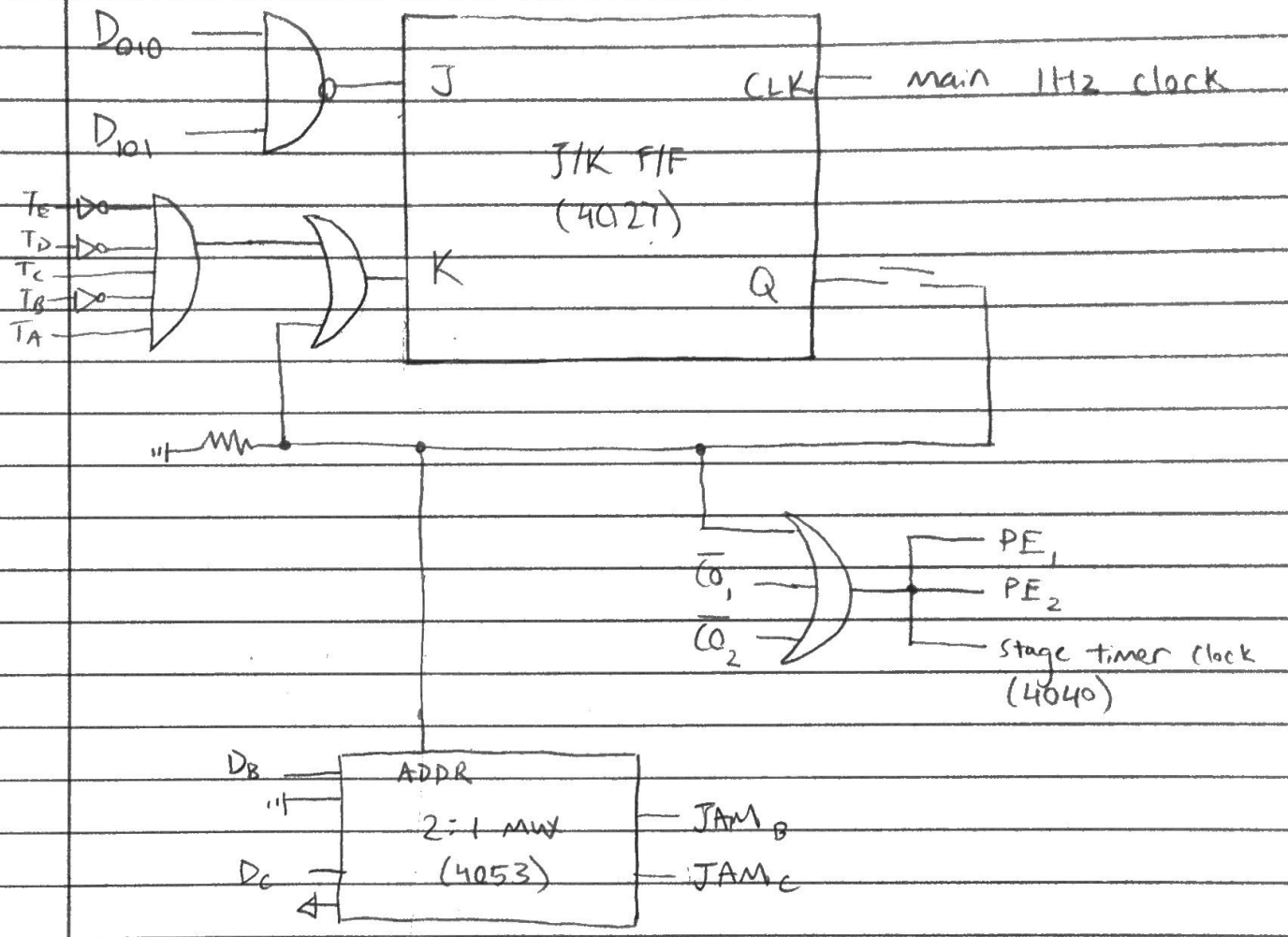
Lastly, to keep the input at 0001 to show “E” or “F”, the LATCH DISABLE pin is set low when the countdown is at one second remaining and remains low until the countdown begins again. This maintains the input to the BCD seven-segment display driver at 0001.

The inputs to the BCD seven-segment display driver are displayed in the following table. These assume that a common-cathode seven-segment LED is in use. If a common-anode seven-segment LED is used, the PHASEs and d segments should be reversed.

Light Color	Time remaining <= 15 seconds	Phase (PH)	Displays input (DCBA DCBA)	Displays showing	Bottom segment (d)
red/yellow	n/a	1	0001 0001	F F	1
green	no	1	0001 0001	E E	from display driver d output
green	yes	0	$00T_FT_E$ $T_DD_CD_BT_A$	time remaining	from display driver d output

TRAFFIC LIGHT EXTRA:
ACCELERATE PEDESTRIAN BUTTON CIRCUIT

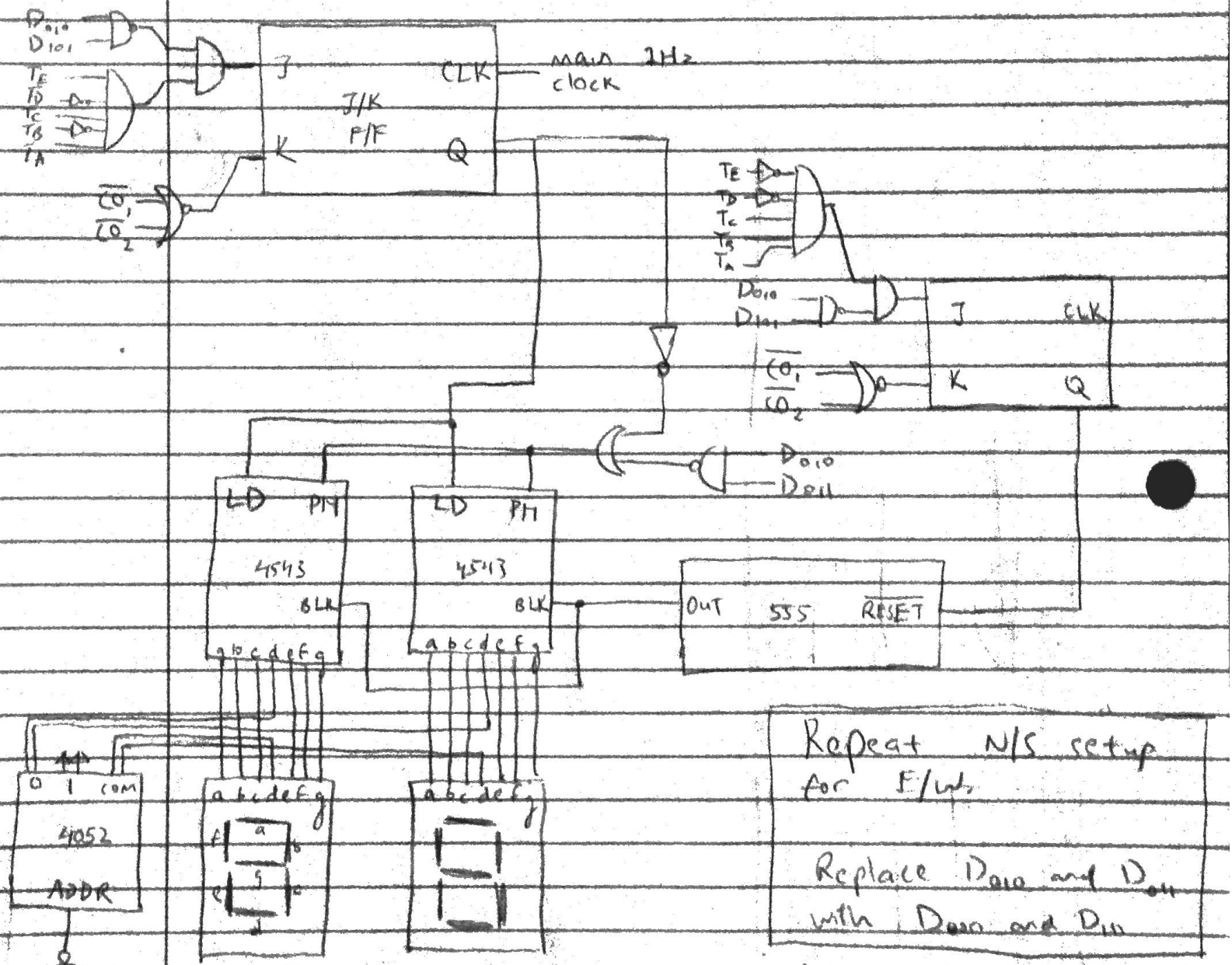
JONATHAN LAM



TRAFFIC LIGHT EXTRA.

PEDESTRIAN TIMER CIRCUIT

JONATHAN LAM



Obstacles and Problem-solving Moments

The first major difficulty was finding a way to make the LEDs light up for the correct amount of time. The idea of using a stage counter quickly arose, but it took the first day to come up with some good idea for ways to advance the stage counter at the right time. I brainstormed with Nathaniel K., Zhihao W., and Dan K. to find multiple possible solutions (outlined in "Regular Credit Design Process and Final Design").

The most difficult design block was working out the details of the two extra credit designs. Most of the time working this project was spent hypothesizing with the aforementioned peers to come up with slimmer, more space-efficient designs for the accelerate green light circuit and the pedestrian timer circuit. Unfortunately, none of us could figure out how to incorporate all of the extra credit portions onto three boards because every solution took far too many chips.

The final problem was meeting the specification in the timing: a tolerance of one second from the total cycle length of 60 seconds. While the values of the resistors was calculated to produce a 1Hz frequency using the 555 timer, there was always some inconsistency caused by the 555 IC, the resistors, and the capacitors. By experimenting a little bit with an oscilloscope and slightly-different valued resistors, I was able to choose a resistor combination that met the specification *most of* the time (staying within the specification at least ¾ of measured cycles), but I was unable to find a solution that always worked.

(Estimated total project time: 50 hours. Actual total project time: probably around 70 hours, including time actively thinking about it but not physically designing or building.)

Attributions

- Zhihao Wang: for working through the MUX-based solutions with 8, 4, 2, 1, and 0 (with similar logic) MUXes with me.
- Dan Kim: for coming up with the idea of mapping current states bits to the duration of the next stage, and for proposing to use a decoder/DEMUX to simplify the logic for the mapping, both of which are included in my final design.
- Nathaniel Kingsbury: for reminding me that the 4029 counters can be used in decade mode to make the transfer to two (base ten) seven-segment displays easy; while seven segment displays were not implemented, the duration mappings and the 4029 counters in the final implementation were left in decade mode. Also for coming up with the idea of inverting the phase to (more) easily produce an “E” and “F” on the seven-segment displays using a BCD encoder chip (4543).