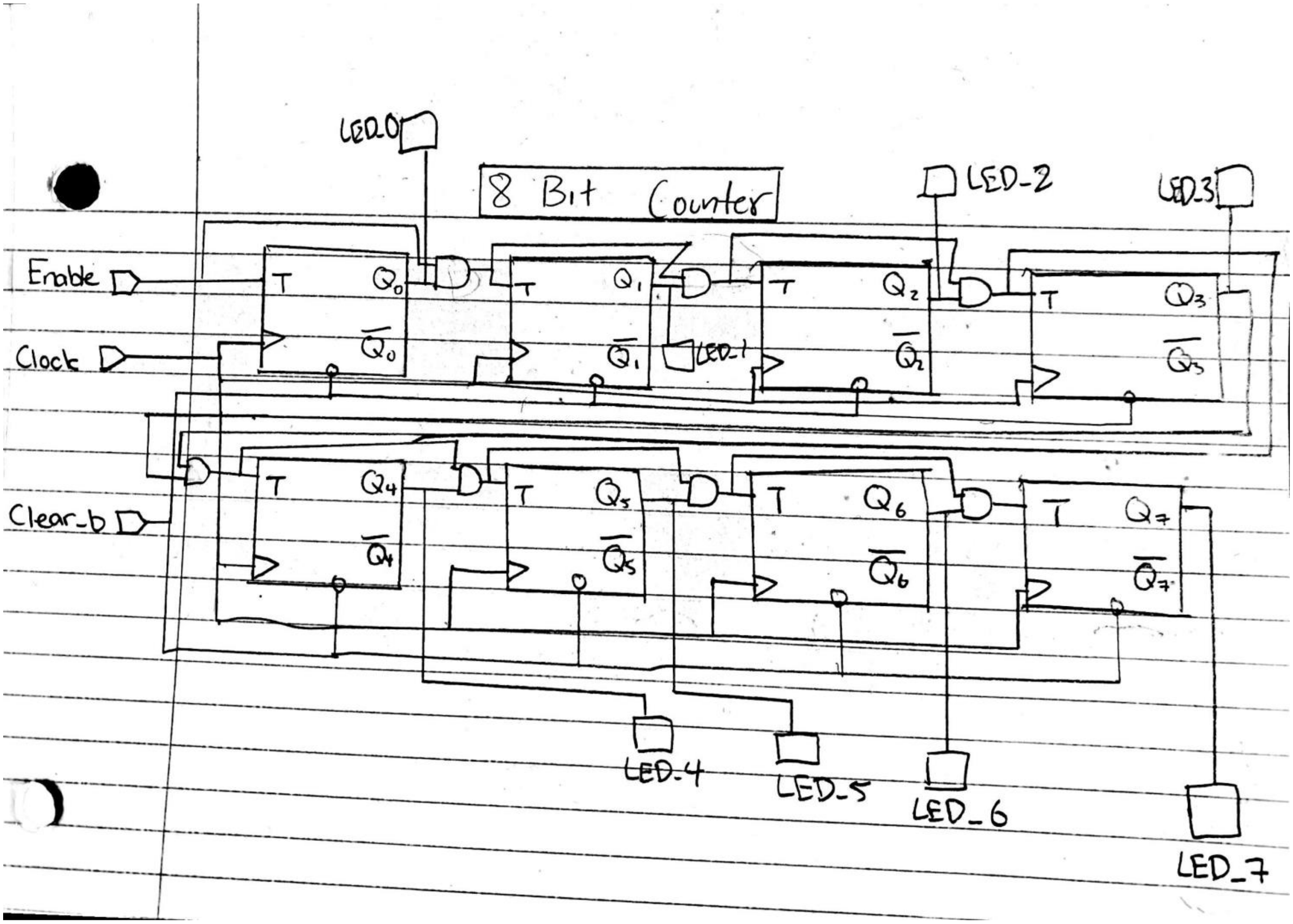
Tony Hong
1005109482
Oct. 21, 2020

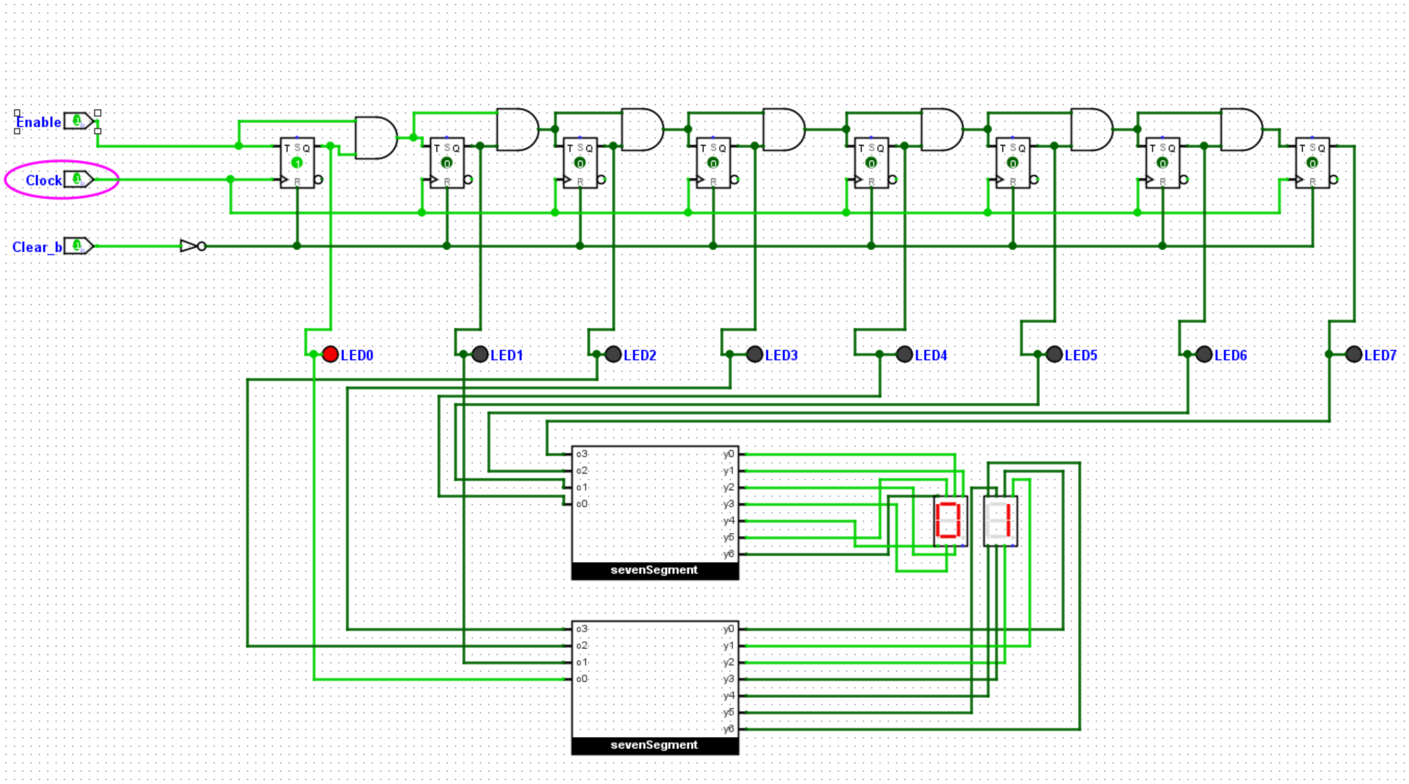
CSC258 Prelab (Lab 5)

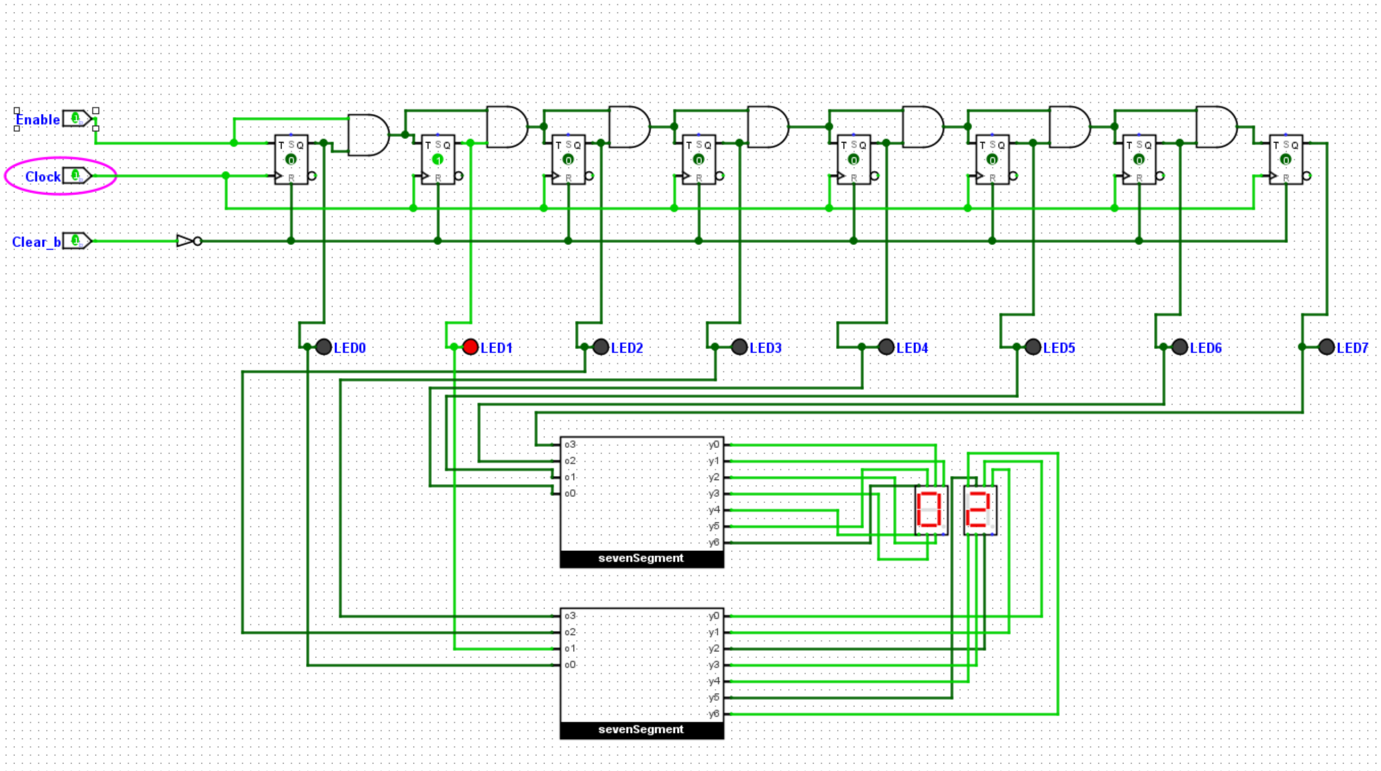
Part 1: 8-bit Counter

1. Below is the schematic for my 8-bit counter.



5. Below are the screenshots of the counter counting from 1 to 2:



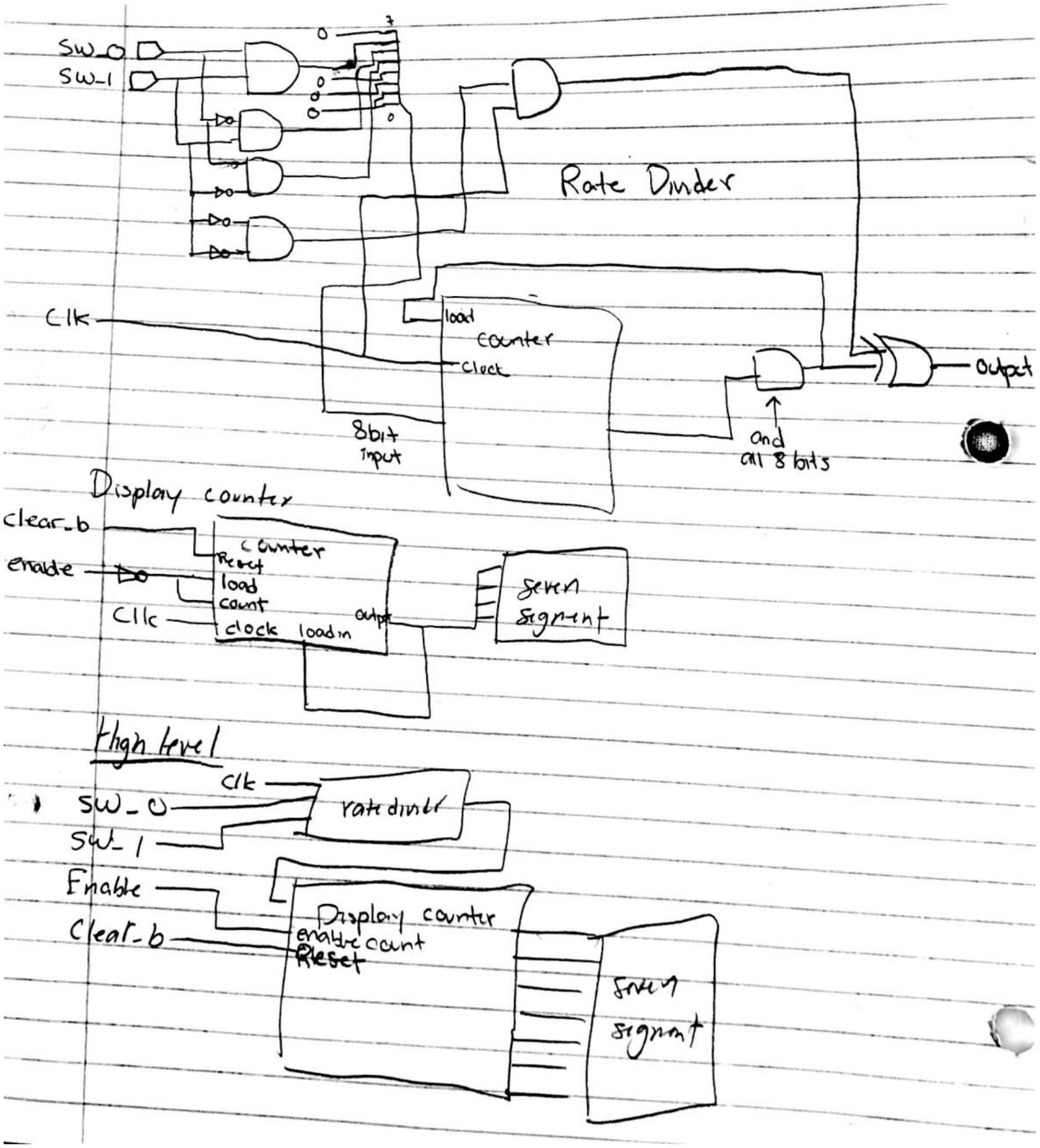


Part 2: Rate Dividing Counter

1. The check for the maximum value is not necessary because once the counter counts to its max value, the next value that is counted will automatically be reset to 0 since we lose the most significant bit.
2. I would add a not gate on output bit 1 and output bit 2 after coming out of the splitter and before the AND gate. This will cause the counter to reset to 0 the clock cycle after we reach value 9. This is because when we count to 9, the load will be enabled instead of count and the default 0 will be loaded in the next clock cycle.
3. - On setting Wrap Around, the counter when reaching maximum displayable value resets back to 0.
   - On setting Stay at Value, the counter stays at the maximum displayable value of all 1's on each bit.
   - On setting Continue Counting, the same behavior is shown compared to Wrap Around.
   - On setting Load in Value, whatever value is connected to the counter is loaded in when the counter reaches maximum displayable value

To count 50 million clock cycles, the ceiling of $\log_2$(50 million) binary bits will be needed to represent such a value. This equates to a **26-bit** counter.

Below is the schematic for my rate dividing circuit.

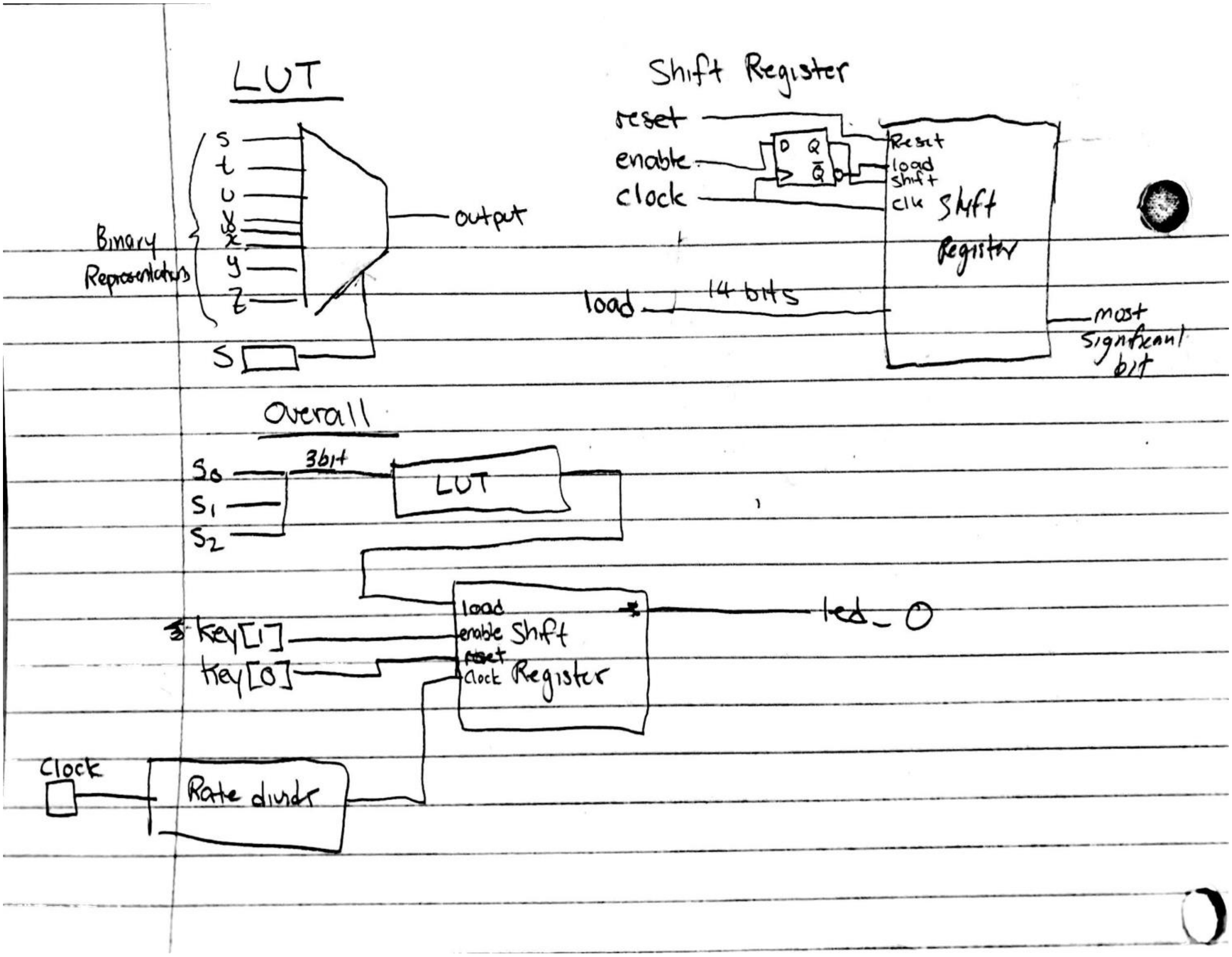


Part 3: Morse Code Translator

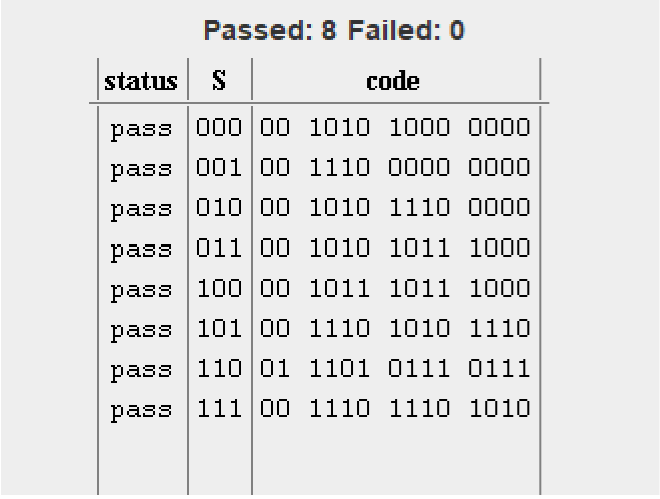
Below is the Look Up Table for my Morse Code implementation.

Letter	Morse Code	Pattern Representation (16 bits)
S	. . .	00101010000000
T	_	00111000000000
U	. . _	00101011100000
V	. . . _	00101010111000
W	. _ _	00101110111000
X	_ . . _	00111010101110
Y	_ . _ _	01110101110111
Z	_ _ . .	00111011101010

2. Below is a schematic of my circuit



4. Below shows a screenshot of test vectors for my LUT.

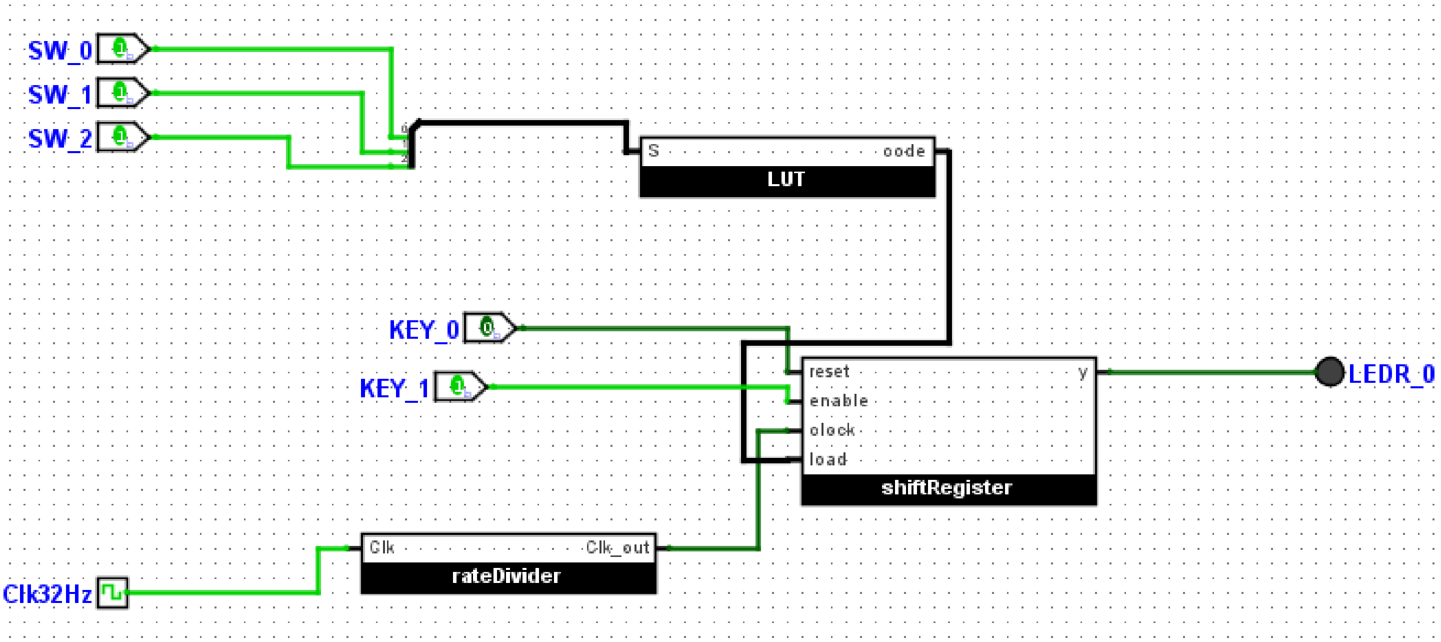


Passed: 8 Failed: 0

status	S	code
pass	000	00 1010 1000 0000
pass	001	00 1110 0000 0000
pass	010	00 1010 1110 0000
pass	011	00 1010 1011 1000
pass	100	00 1011 1011 1000
pass	101	00 1110 1010 1110
pass	110	01 1101 0111 0111
pass	111	00 1110 1110 1010

Tests that I have run for my circuit include testing each individual letter and viewing the behaviour of the LED, matching each sequence of light flashing to the binary representation of the morse code.

Below is a screenshot of my Circuit:


SW_0
SW_1
SW_2
S
code
LUT
KEY_0
KEY_1
reset
enable
clock
load
y
shiftRegister
LEDR_0
Clk
Clk_out
rateDivider
Clk32Hz