Latches and Level Conditioners

Kevin "Yama" Keyser[1, a)]
University of Missouri-Kansas City

In Laboratory 7, we extend our introduction to truth tables and logic gates, and expand these ideas with our first flip-flop. Specifically a Set-Reset, or SR flip-flop. We will also construct a schmitt trigger using a comparator Op-Amp, and finally implement a 555 timer in a circuit.

I. BACKGROUND

Laboratory 7 bring together (in an abstracted manner) everything we've done up until now. The 555 timer when deconstructed, much like the Op-Amp, is a collection of of Op-Amps, resistors, a switch, flip-flop, and buffer (which we haven't used) to create a circuit that gives us specific square wave timing. In this one chip is everything we've used up until now.

II. PROCEDURE

The procedures for Laboratory 7 will be attached as a separate sheet of paper to the back of the laboratory write up.

III. PRESENTATION OF DATA

A. 7-1: Building a "Set-Reset" Latch with NAND gates

NAND SR Truth Table			
S	R	Q	Q'
1	0	0	1
1	1	0	1
0	1	1	0
1	1	1	0

B. 7-2: Building a "Set-Reset" Latch with NOR gates

NOR SR Truth Table			
S	R	Q	Q'
1	0	1	0
0	0	1	0
0	1	0	1
0	0	0	1

a) Electronic mail: kk8r8@mail.umck.edu

C. 7-3: Level Conditioning

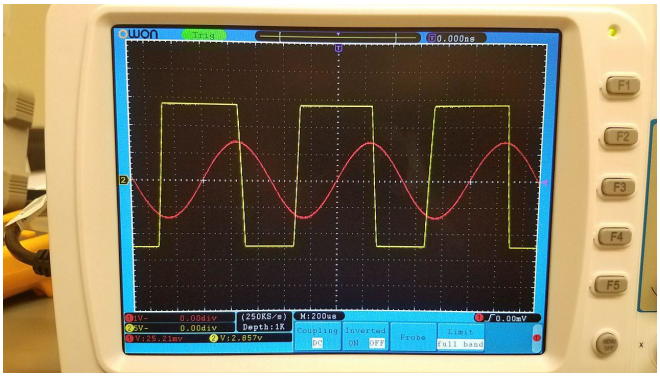

FIG. 1. Input and output for Schmidt trigger at 400Hz.

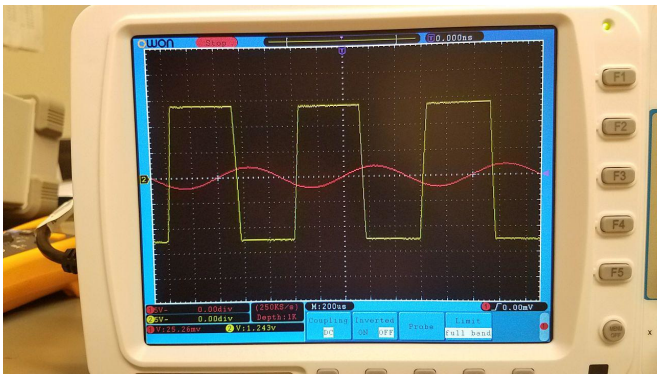

FIG. 2. Input and output for Schmidt trigger at 1kHz.

FIG. 3. Input and output for Schmidt trigger at 11kHz

D. 7-4: 555 Timer Circuit

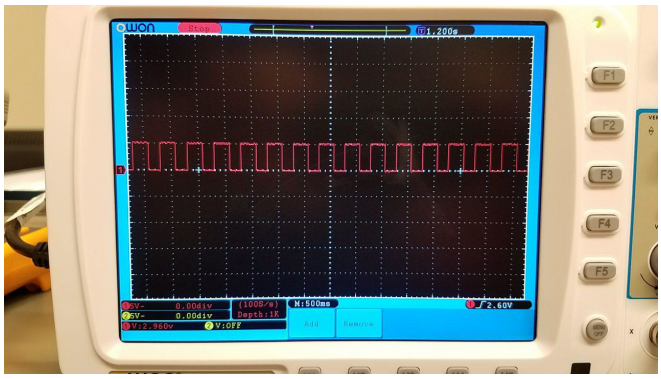

FIG. 4. Cycle for 555 timer

IV. DISCUSSION

A. 7-1: Building a "Set-Reset" Latch with NAND gates

When looking at the results of our truth table, and what the intended application of our Set-Reset flip-flop is, we can figure out which states Sets and Resets our values. We can see that S=1 and R=1 is a two state stable state. It remembers the previous value that came before it, so regardless if S was 0 or R was 0, when they're put back to 1, the SR flip-flop will still keep its previous value. Then we can see that the zero value, or the off value is what actually triggers our desired effect for this circuit. When R is 0, our Q becomes 0, no matter what its previous value was. When S is 0, our Q becomes 1, no matter what our previous value for Q was. Q' will always output the negation of Q in this circuit.

B. 7-2: Building a "Set-Reset" Latch with NOR gates

The NOR implementation of the SR flip-flop works more in line with what we would want for our logic gates, as it requires less power to keep circuit in our stable state. This time, when S=0, R=0, this is our two state stable state. So the previous value of Q is preserved. Now, when S is 1, our Q becomes 1. When R is 1, Q becomes 0. Similarly, Q' is the negation of Q.

C. 7-3: Level Conditioning

With the Schmidt trigger we were able to find different V_{pp} values for different frequencies at which the square wave for our output stabilizes. At 400Hz, we can be at 2.72 V_{pp}, where for 1kHz we go slightly higher to 2.75 V_{pp}. At lower frequencies, this number seems to drop, where at higher values for frequency, we stop seeing inversion happen. There is not enough time for this transition to happen. At 11kHz, we see our square wave look more trapezoidal, and it no longer reaches negative values. Since the default output is +15V in our configuration, it makes sense that it requires a high enough, and long enough positive voltage to give us our negative output (since we are doing an inverted input). When the frequency goes too high, it can't transition fast enough and we get values that no longer can be inverted.

D. 7-4: 555 Timer Circuit

The formula for time to be a high output is given by $T_H = 0.7(R_1 + R_2)C$. The formula for time to be a low output is given by $T_L = 0.7R_2C$. If we have a desired high and low, the best thing to do is to pick a capacitor, plug it into our equation for T_L, since we know what our T_L we want, and we get our R_2. We can do the same by plugging our solved values into the equation for T_H, and solve for R_1. This is how we chose our values for our capacitors, with $C = 1\mu F$, $R_1 = 100k\Omega$, and $R_2 = 300k\Omega$. This gives us an approximate $T_H = 0.3s$, and $T_L = 0.2s$. Since the period is how much time it takes for a function to get back to where it was, we can add both together to get $T = T_H + T_L$, to get a period of 0.5s.

V. CONCLUSION

From 7-1 and 7-2, we see how we can construct a Set-Reset Latch, or Flip-Flop as known in the Computer Science field. The SR flip-flop is used to set a value and reset a value. This can be useful for storing values, like we do in RAM. Both versions show that there are separate ways of implementing an SR flip-flop, and how to read the truth tables (more importantly, how to predict the truth tables) for each. Though from the writer's point of view, the NOR version seems more intuitive than the NAND implementation.

Using a 555 timer in addition to the SR flip-flop, we can set a time for how often we want these values to be update. We can choose how long a clock cycle is with this set up. Though our 0.5s is not very realistic for any practical use, the implementation of and execution of our given formula gives us the basic underlying use of the circuitry.