Lab 6 Report
ECE1212 Electronic Circuits Design Lab
Lab Due Date: April 11, 2019

Nate Carnovale and Seth So
Lab Station: 19
TA: Yanhao Du

August 15, 2020

Introduction

A shift register is a digital memory device in which data are loaded serially, one bit at a time. The simplest and most common implementation of a shift register is constructed by connecting a series of D flip-flops, in which the Q output of each stage feeds the D input of the next. The figure below shows a four-bit shift register.

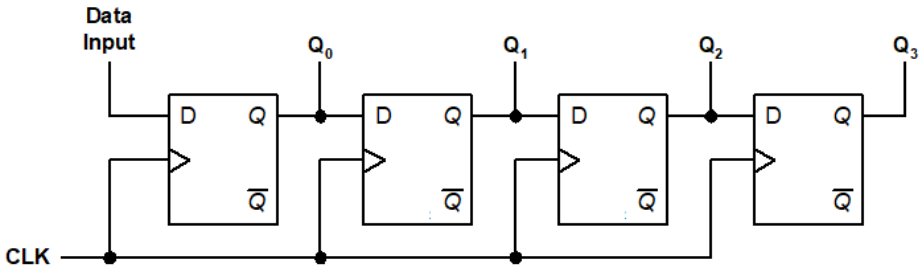

Figure 1: Four-Bit Shift Register

Shift registers are found in a variety of digital and analog applications. Perhaps the most widely used of these is in digital filtering, where bits representing samples of an analog signal are shifted through the register. By connecting appropriate logic gates or resistor networks to the shift register outputs, the desired filtering operation can be achieved.

In this experiment, students will use a shift register as a simple memory device. Students will then proceed to implement a Johnson counter and a digital sine wave generator, each of which uses a shift register as the primary element.

Procedure

Part 1: Shift Register Construction

(1) Using a 1kΩ resistor and SPST switch, students constructed two of the logic sources shown in the figure below.

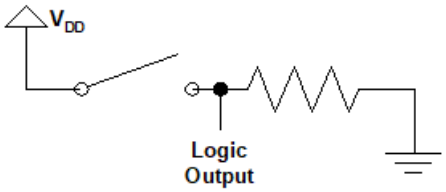

Figure 2: Logic Source

(2) Students next built the 8-bit shift register using the 74HC4015, by connecting the two clock inputs together, and connecting the last output from one shift register to the data input of the other. Students connected the logic sources to the data and clock inputs of the 8-bit shift register.

(3) To each of the Q outputs of the shift register, students connected a LED and a series resistor between the output and ground, to visualize the contents of the shift register. The completed shift register circuit schematic is shown below.

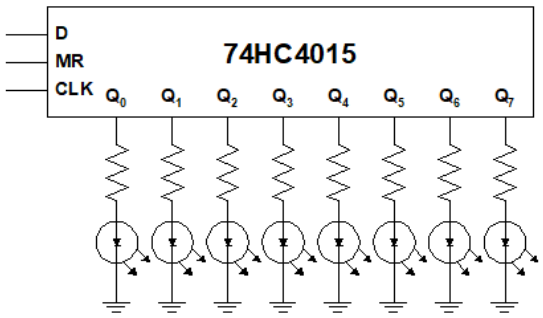

Figure 3: Completed Shift Register

(4) Students then experimented with loading the shift register with data, including loading it with an alternating pattern of zeros and ones, and then shifting these bits out. Students also explored the effect of switch bounce in the circuit.

(5) Finally, the register was loaded with the 8-bit ASCII code for the letter S of Seth's name (01110011) to show proficiency in using the register circuit.

Part 2: Johnson Counter

(1) Students constructed a CMOS inverter using a 4007 MOSFET package. The last output of the CMOS inverter was connected to register data input as shown below.

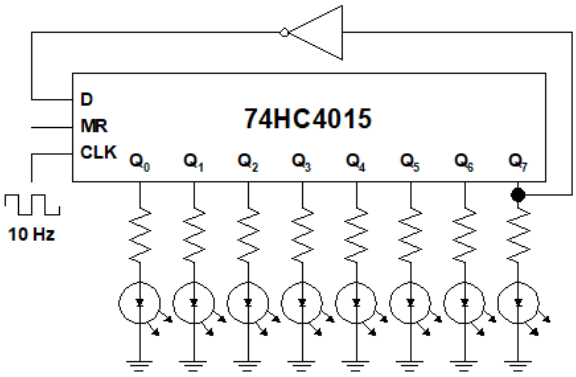

Figure 4: Schematic for Johnson and Johnson - a family company - counter schematic

(2) Students set the function generator to produce a 10 Hz square wave and set the amplitude and DC offset so that the square wave oscillates between 0V and +5V. Then they connected this signal to the CLK input of your shift register. The resulting circuit is called a Johnson counter. Using the scope to display CLK and Q0 simultaneously, students captured an image of the scope display. This was done for the relationships among the four outputs (Qo – Q3).

Part 3: Digital Sine Wave Generator

(1) Students disconnected the LED's from the outputs of the Johnson Counter and attached resistors to outputs Q0 – Q6, as shown below, and tied the opposite ends together to form the output signal.

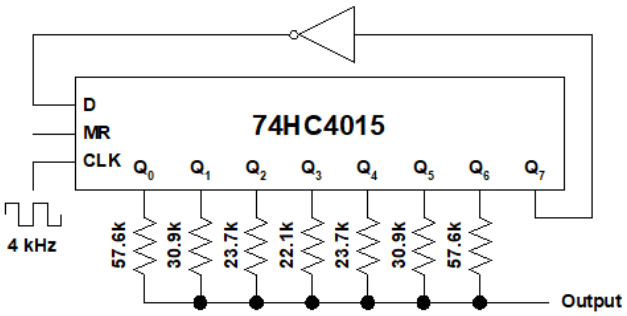

Figure 5: Schematic for sine-wave generator utilizing the Johnson counter with the 74HC4015 integrated circuit

(2) The frequency of the square wave CLK signal was set to 4 kHz. The clock and output signals were displayed on the scope and the images were obtained. Also, the frequency and amplitude were determined by superimposing a sine wave onto the "staircase" output.

(3) Students determined that the output could be improved and smoothed by using a capacitor in the circuit. Adding a capacitor across the output, students also captured a scope image of the smoothed output.

Results

Part 1: Shift Register Construction

(1, 2, and 3) No results to report from these steps.

(4) The shift register functioned as intended. The task of loading the register with an alternating pattern of zeros and ones was completed successfully. Switch bouncing occurred mostly from the clock input since the register triggers on the clock edge. The result of the switch bounce was multiple shifts occurring for one intended click.

(5) The following following picture shows the binary ASC-ii code for "S" loaded into the shift register:

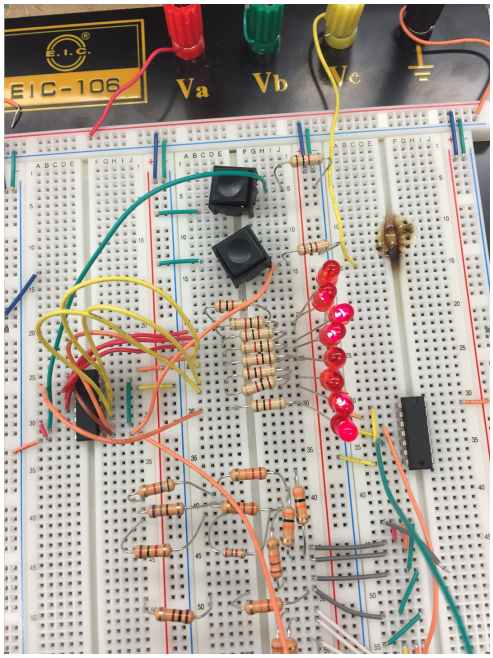

Figure 6: Loaded shift register

q_0 was the top, followed by the other outputs. The lights display the sequence "01110011" (0 being off and 1 being on), which correctly represents ASC-ii "S."

Part 2: Johnson Counter

(1) The circuit was correctly made after several mis-wiring issues were resolved. The resulting circuit was the same as the show in Part 1 with the addition of an inverter, but the output of the LEDs was no longer static. Instead it was appeared as a wave of the lights turning on and off, as described in the would happen in the Lab instructions.

(2) Below are comparisons of the different outputs of the Johnson counter:

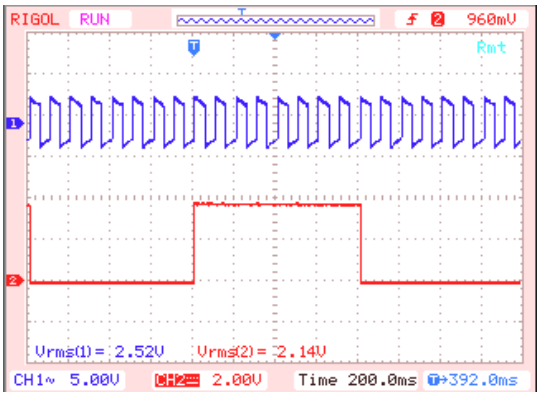

Figure 7: Loaded shift register

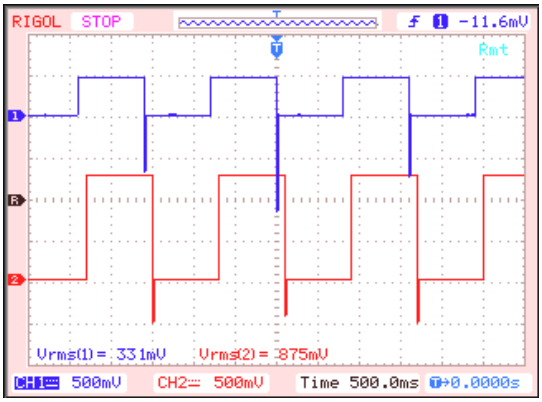

Figure 8: Oscilloscope image of q0 and q1

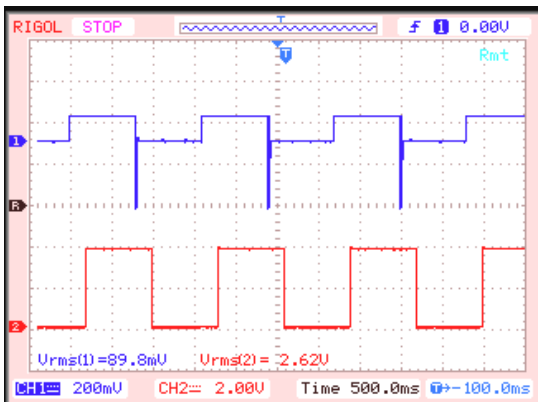

Figure 9: Oscilloscope image of q0 and q2

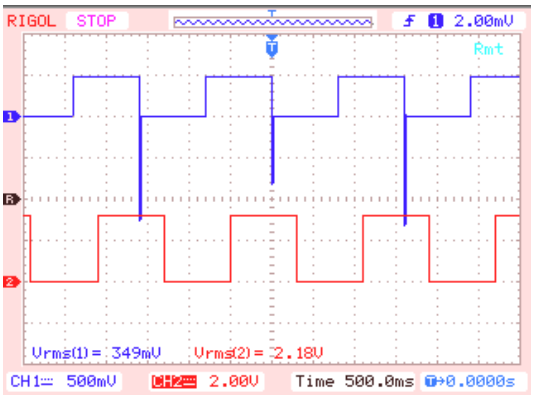

Figure 10: Oscilloscope image of q0 and q3

First is a comparison of clock and q0, followed by q0 with q1, q2, and q3.

Part 3: Digital Sine Wave Generator

(1) Construction was successfully completed with no problems, with no further results to report. Output images will follow.

(2) Below is the output of the sinewave generator when tested with a 4kHz [0-5]Vpp squarewave at 50% duty-cycle:

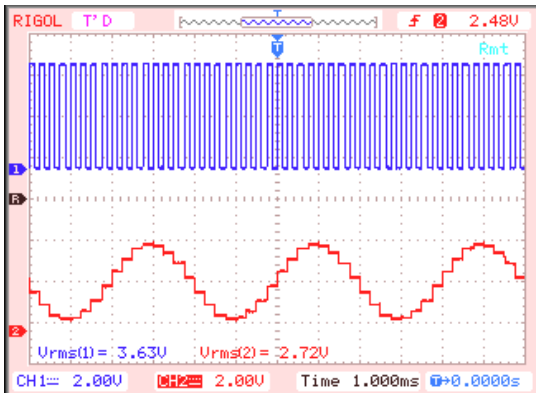

Figure 11: Oscilloscope capture of output of sine-wave generator

(3) Below is the output of the sinewave generator with the same input, after 1uF capacitors were added from output to ground:

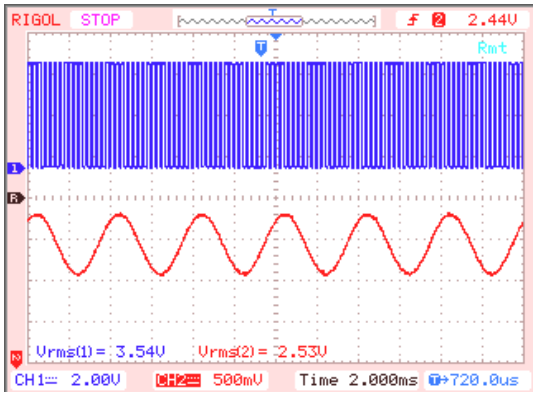

Figure 12: Oscilloscope capture of output of sine-wave generator after addition of 1uF capacitors

Discussion

Part 1: Shift Register Construction

(1) The logic source proved to be an effective way to control the shift register. By using one SPST switch for the data input and one for the clock signal, we were able to easily manipulate the input to get the desired output. The D input button could be held pressed to supply a "1" or left unpressed for a "0". The clock signal button also worked well for this part of the circuit because we could send singular clock pulses to manipulate a specific input for the register rather than using a clock signal which would continually send clock pulses.

(2 and 3) We found that connecting 1kΩ resistors in series with each output LED for the shift register circuit gave the best results. We originally tried using the resistors that were necessary for part 3 of this lab, but found that the resistances were to large and that the illumination of the LEDs could not be seen. Using 1kΩ's seemed to fix this issue.

(4) The circuit worked as intended and was fairly easy to control. However, switch bouncing became a challenge when it came to setting specific combinations of 1's and 0's on the shift register output. The data input D was not very sensitive to switch bounce. This is because the D input was held at either a 1 or 0 for

a fairly long period of time, and therefore any bouncing went away by the type the data was shifted into the register. However, the clock input was more sensitive to switch bouncing. Because the register was triggered on the clock edge, each bounce had the potential of unintentionally triggering the circuit. We observed this issue when we tried to shift in a single bit and the result was actually shifting in two bits for one click.

(5) As seen in the results section, we were successful in loading the ASCII letter S (01110011) into the shift register. As discussed in the previous paragraph, the one challenge we faced was switch bouncing. Therefore, it took more than one try to load the specific bit sequence into the register because the switch sometimes triggered the clock edge more than once although there was only one click.

Part 2: Johnson Counter

(1) We designed the CMOS inverting using a 4007 MOSFET package in the prelab. We put this inverter between the last bit, Q7, to the D input. An 8-bit Johnson counter goes through 16 states beginning with the state of all 0's, 1's begin marching in from the left until the register contains all 1's. At that point 0's march in again and the process is repeated.

There are a total of 16 states and the sequence is as follows:

State
00000000
10000000
11000000
11100000
11110000
11111000
11111100
11111110
11111111
01111111
00111111
00011111
00001111
00000111
00000011
00000001

The Johnson counter works due to the combination of the shift register and inverter between Q7 and the D input. The shift register takes the input D and shifts it into the Q0 place, shifting everything else rightward on each clock edge. The value of Q7 is inverted and the inverted value becomes the next bit shifted into the Q0 place of the register. This can be observed in the table of states above. This is why the register begins with all zeros and then ones march rightward until the register is full. Then zeros march rightward until full. This pattern continues.

(2) Figure 7 shows the clock signal and the Q0 bit of the 8-bit Johnson counter. As seen in the scope image, Q0 is a 1 for 8 clock edges (states) and a 0 for 8 clock edges (states). This totals the 16 states in the Johnson counter sequence. This verifies that the counter was working correctly. Figures 8, 9, and 10 show the relation of Q0 to Q1, Q2, and Q3 independently. These three scope images verify the shifting action of the Johnson counter. As seen by the figures, the time of each HIGH pulse is the same for each Q0, Q1, Q2, and Q3. This shows that each bit is HIGH for the same amount of time. Furthermore, the HIGH pulse starts first in Q0. Compared to the other three bits, Q1 follows Q0 first, then Q2, then Q3. This shows that the HIGH pulses (1's) march in from Q0 towards Q3. They enter Q0 first and leave Q3 last. This again verifies that our Johnson counter function correctly.

Part 3: Digital Sine Wave Generator

(1 and 2) As seen in Figure 11, the result of our circuit did correctly generate a staircase-type output with 16 distinct states. As seen in the figure, there are 16 clock edges per one period of the staircase wave, meaning that the frequency of the staircase was 1/16 that of the clock signal and 16 times the period. This was verified by superimposing a sine wave to measure frequency and amplitude. As expected, the frequency was approximately 250Hz equivalent to a period of 4msec. This frequency is 1/16 of the 4kHz input and 16 times the period of 0.25msec of the clock signal. Our measured peak to peak voltage was about 4V which is 1 V of the predicted 5V. However this could be due to theoretical calculations with nominal resistor values and reduced voltage due to the inverter circuit.

(3) This circuit works by using voltage division with specific resistor values. The output voltage is the result of the voltage division equation:

$$V_{out} = 5V * (\frac{\text{Parallel Equiv. R of all 0 Bits Resistors}}{\text{Parallel Equiv. R of all 0 Bits Resistors + Parallel Equiv. R of all 1 Bits Resistors}})$$

As the number of 1 bits in the shift register increases, the voltage drop across the 0 bits resistors increases because the parallel equivalent resistance for the 1 bits decreases. The maximum voltage is 5V when all of the bits Q0-Q6 are 1.

One of the issues seen in this circuit is that the output is similar to a staircase and is not a smooth sine wave. As seen in Figure 12, we used a 1μF capacitor to smooth out the staircase curve and had fairly good results.

Conclusion

The purpose of lab 6 was to experimentally test an 8-bit shift register and utilize different circuit configurations to obtain different sequences. The experiments of the lab included designing and testing an 8-bit shift register circuit using two logic sources as the data and clock input signals. Once the shift register circuit was verified to be functioning correctly, a CMOS inverter was constructed and used to create a Johnson counter in which 1's and 0's alternate marching through the shift register from left to right until they fill the register. Finally, the Johnson counter circuit was then adapted using specific resistances in a circuit configuration to create a digital sine wave approximation. Using a capacitor, the output was smoothed to better approximate an actual sine wave.

Results accurately reflected what was expected and taught in lecture. Key takeaways were learning the structure and usage of nMOSFET IC CD4007 as well as the 74HC4015 8-bit shift register. This experiment also exposed students to one of the applications of the variety of digital applications for shift register. This experiment also demonstrated the importance of accuracy in design to obtain desired results and the differences that can arise between theoretical and practical responses.

References

[1] Experiment 6 – Shift Registers and Applications Lecture Notes

[2] Rigol DM3058 User's Guide. https://www.csulb.edu/sites/default/files/groups/college-of-engineering/About/rigol-dm3058-digital-multimeter-user-guide.pdf

[3] CD4007UB Data Sheet from course website

[4] 74HC4015 Data Sheet. http://www.ti.com/lit/ds/symlink/cd74hc4015.pdf

[5] Professor Li