Logic Design 2DI4
Lab 4

Instructor: Dr. Bauman/Doyle
Lab TAs: Kamal Vaghasiya (vagha1@mcmaster.ca)
Severin Hidajat (hidajats@mcmaster.ca)
FengYang Sun (sunf10@mcmaster.ca)
Orges Ali (alio5@mcmaster.ca)
Paniz Eilkhani (eilkhan p@mcmaster.ca)

Mustafa Shahid– L08 – shahim45 - 400440384
Abaan Khan – L08 – khana454 - 400428399

As a future member of the engineering profession, the student is responsible for performing the required work in an honest manner, without plagiarism and cheating. Submitting this work with my name and student number is a statement and understanding that this work is my own and adheres to the Academic Integrity Policy of McMaster University and the Code of Conduct of the Professional Engineers of Ontario. Submitted by **[Abaan Khan, khana454, 400428399]**

REGISTERS

4.5.2 Registers

A register is a group of n flip-flops operating together to represent an n-bit binary number. Each flip-flop represents a single bit. The simplest form of a register is used to capture its input(s) and hold the value (e.g.,the D flip-flop) in response to the write command (a clock pulse). A common application in using registers is to shift their content by one bit left or right. A left shift, moving all bits one position to the left, has the effect of multiplying the content value by 2. Conversely, a right shift has the effect of dividing the content value by 2. For the implementation of these registers, please design the circuit using schematic capture in the Quartus Prime Lite IDE. Load each circuit onto the MAX10 chip and test with toggle switches and LEDs.

Milestone 1: TA to check Circular Shift register operation

1. There are a number of ways to connect flip-flops in series to create shift registers. Figure 4.2 illustrates a simple left-to-right shift register. Note it is constructed using JK's in such a way they operate like D flip-flops. Implement the circuit as shown using schematic capture design entry and load onto the FPGA. Tie your preset values HI and experiment with this circuit using a very low clock frequency supplied by the physical bench top function generator (Alternatively you could use a toggle switch to simulate the clock). Observe the register bit output patterns (Q3 Q2 Q1 Q0) on LEDs. Remember to also use the toggle switches to input the clear and serial data signals. If the HDL model of the flip-flop does not provide a Q' output then implement one using an inverter off of Q.

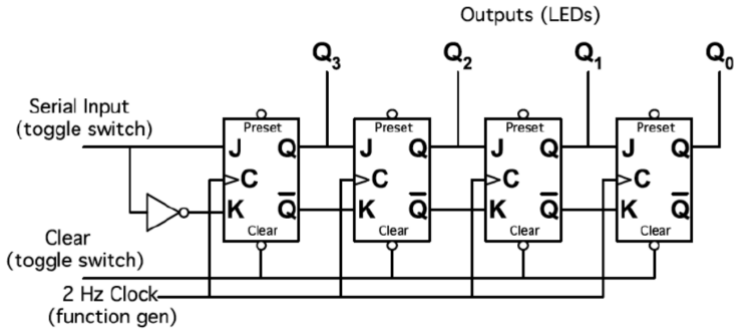
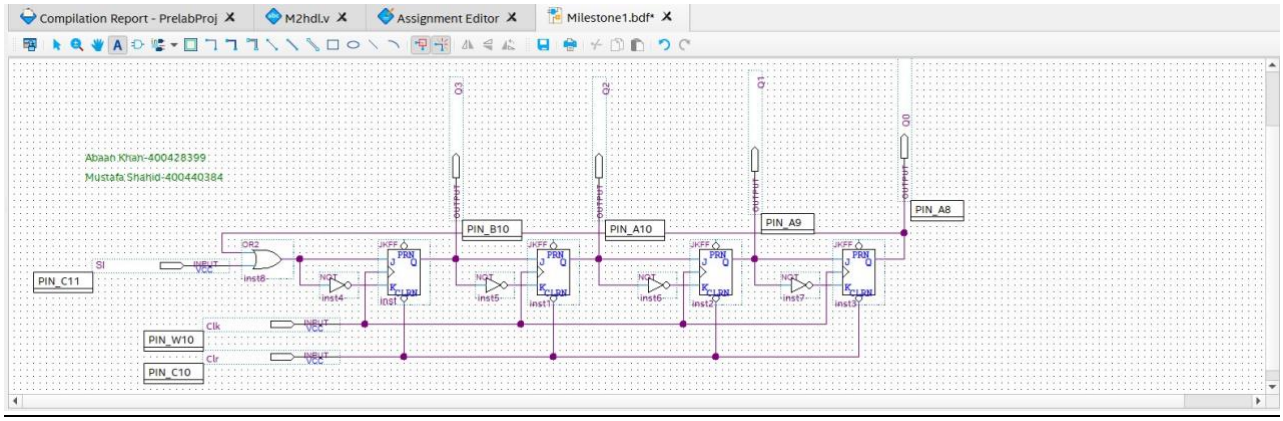
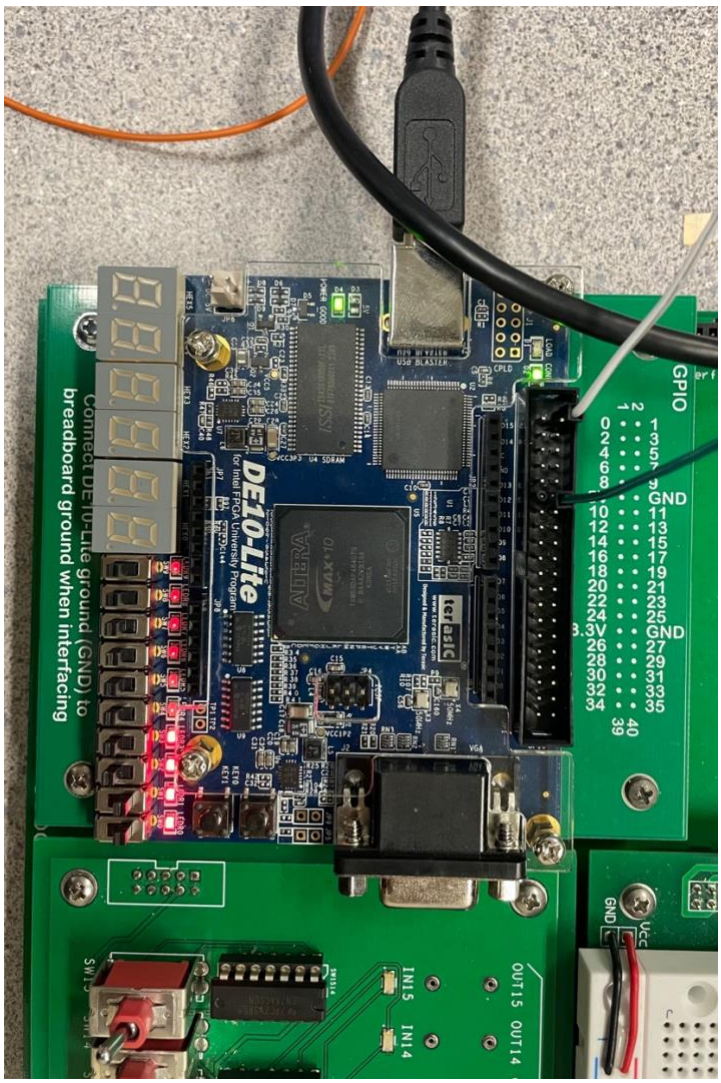


Figure 4.2: A Shift Register Implementation

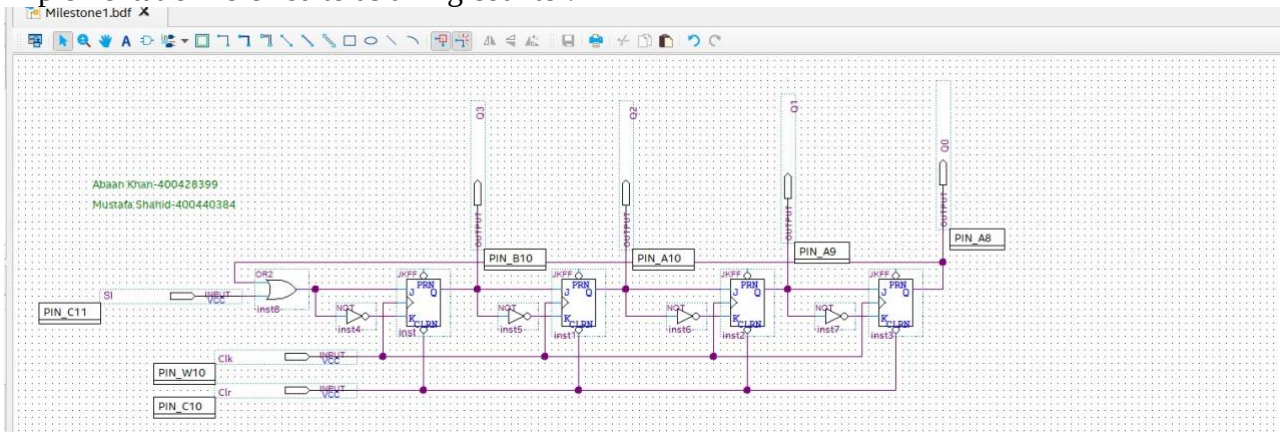
Truth Table

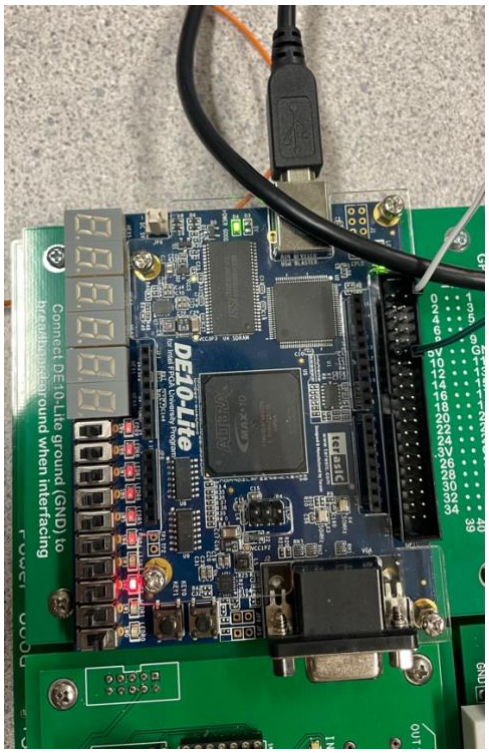
J	K	CLK	Q
0	0	↑	Q_0 (no change)
1	0	↑	1
0	1	↑	0
1	1	↑	$\overline{Q}_0$ (toggles)


Compilation Report - PrelabProj
M2hdl.v
Assignment Editor
Milestone1.bdf*
Abaan Khan-400428399
Mustafa Shahid-400440384
PIN_C11
PIN_W10
PIN_C10
PIN_B10
PIN_A10
PIN_A9
PIN_A8
Q3
Q2
Q1
Q0
SI
Clk
Clr
OR2
NOT
JKFF
INPUT
VCC


DE10-Lite
for Intel FPGA University Program
GPIO
GND
3.3V
Connect DE10-Lite ground (GND) to
breadboard ground when interfacing
IN15
IN14
OUT15
OUT14

2. Modify your Shift Register circuit as shown in Figure 4.4 to implement a circular shift capability by connecting the Q0 output back to the input through an OR gate. You will also find this implementation referred to as a ring counter.

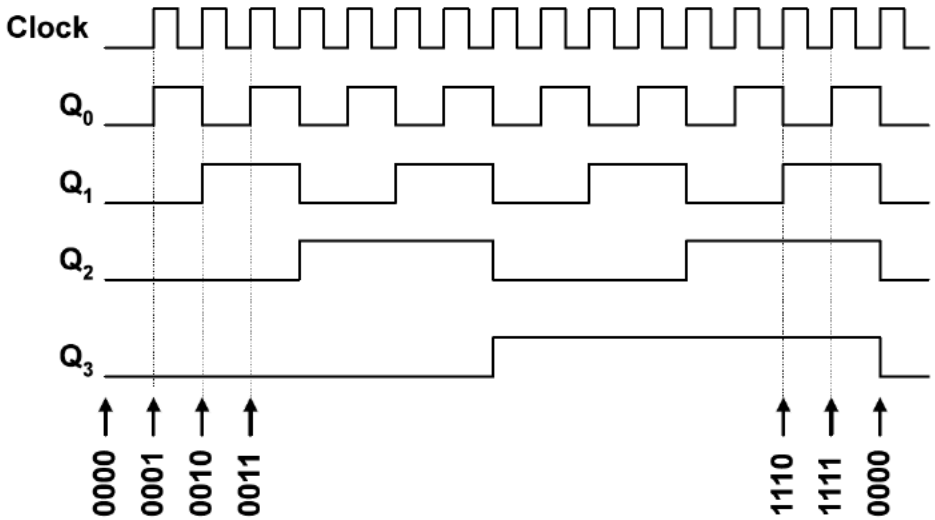




4.5.3 Counters
As we have a focus on synchronous sequential logic, you will only be required to build a synchronous counter.
However, the student should be aware that there exist many asynchronous counter designs which can cause difficulties due to intermediate false outputs caused by time delays. Using HDL implement the synchronous counter from Figure 4.6. The counter from Figure 4.6 is a better design because all flip-flops receive a common
clock pulse. Verify the timing and operation using the timing diagram provided, Note: count enable can freeze
the clock count, and clock transition occurs on rising edge.

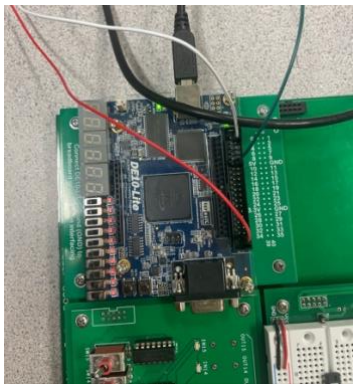
Milestone 2: TA to check Synchronous Counter operation

The design of this counter uses *serial gating* which means that, for example, in going from state **0111** to **1000**, there are three delays through the AND gates required to generate the T-input to the final bit. This restricts the maximum count frequency. *Parallel gating* is analogous to look-ahead carry logic and allows for the design of faster counters.



M2hdl.v | Assignment Editor

```
1   //Abaan Khan - 400428399
2   //Mustafa Shahid - 400440384
3
4
5   module M2hdl(clok,clr,enable,q0,q1,q2,q3,carry);
6      input clok, clr, enable;
7      output q0,q1,q2,q3,carry;
8
9      wire w1,w2,w3;
10
11
12     JKFF one (
13              .j(enable),
14              .k(enable),
15              .clk(clok),
16              .clrn(clr),
17              .prn(1),
18              .q(q0)
19             );
20     assign w1 = enable & q0;
21
22     JKFF two (
23              .j(w1),
24              .k(w1),
25              .clk(clok),
26              .clrn(clr),
27              .prn(1),
28              .q(q1)
29             );
30     assign w2 = w1 & q1;
31
32        JKFF three (
33              .j(w2),
34              .k(w2),
35              .clk(clok),
36              .clrn(clr),
37              .prn(1),
38              .q(q2)
39             );
40
41     assign w3 = w2 & q2;
42
43     JKFF four (
44              .j(w3),
45              .k(w3),
46              .clk(clok),
47              .clrn(clr),
48              .prn(1),
49              .q(q3)
50             );
51     assign carry = q3 & w3;
52  endmodule
53
```



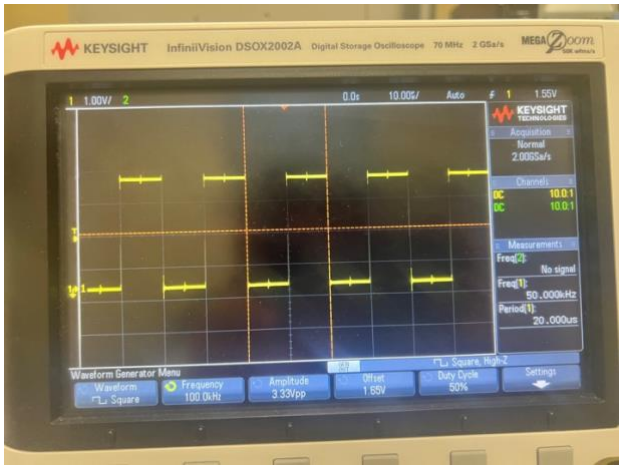
M2hdl.v | Assignment Editor

<<new>> ☑ Filter on node names: *

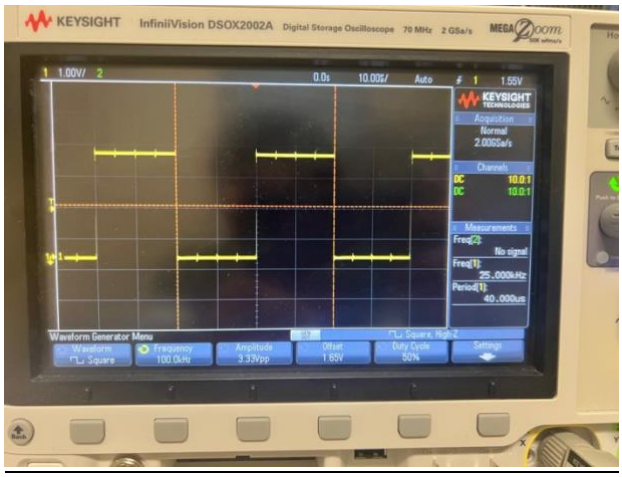
	tatu	From	To	Assignment Name	Value	Enabled	Entity	Comment	Tag
1	✔		in clr	Location	PIN_C10	Yes			
2	✔		in enable	Location	PIN_C11	Yes			
3	✔		out carry	Location	PIN_Y6	Yes			
4	✔		out q0	Location	PIN_AA2	Yes			
5	✔		out q1	Location	PIN_Y3	Yes			
6	✔		out q2	Location	PIN_Y4	Yes			
7	✔		out q3	Location	PIN_Y5	Yes			
8	✔		in clok	Location	PIN_W10	Yes			
9		<<new>>	<<new>>	<<new>>					

Frequency measured at :

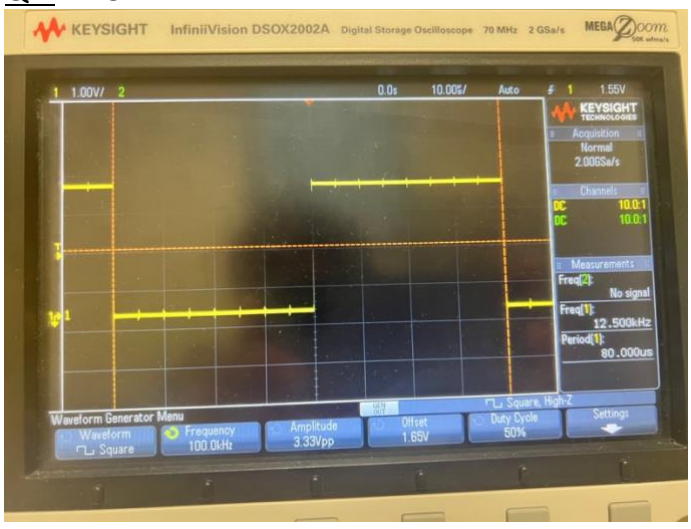
Q0: 50 KHz



Q1: 25 KHz

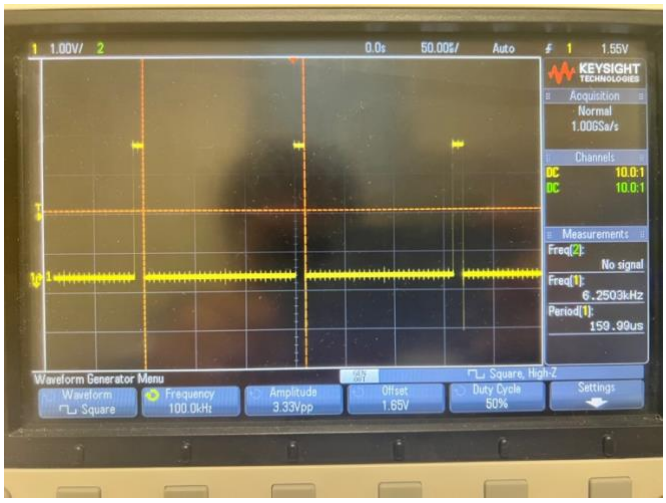


Q2: 12.5 KHz



Q3: 6.25 KHz

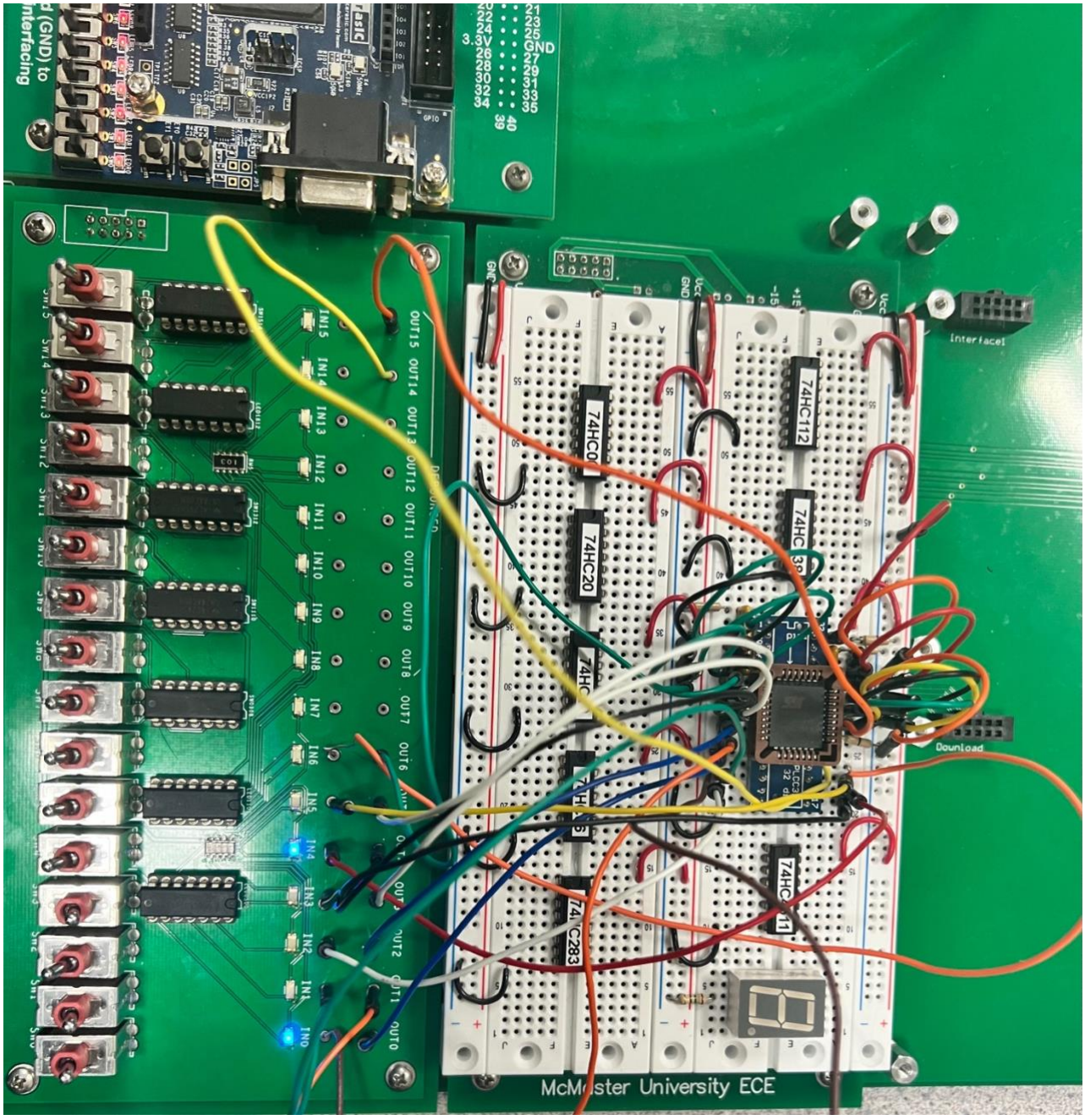
Carry: 6.25 KHz



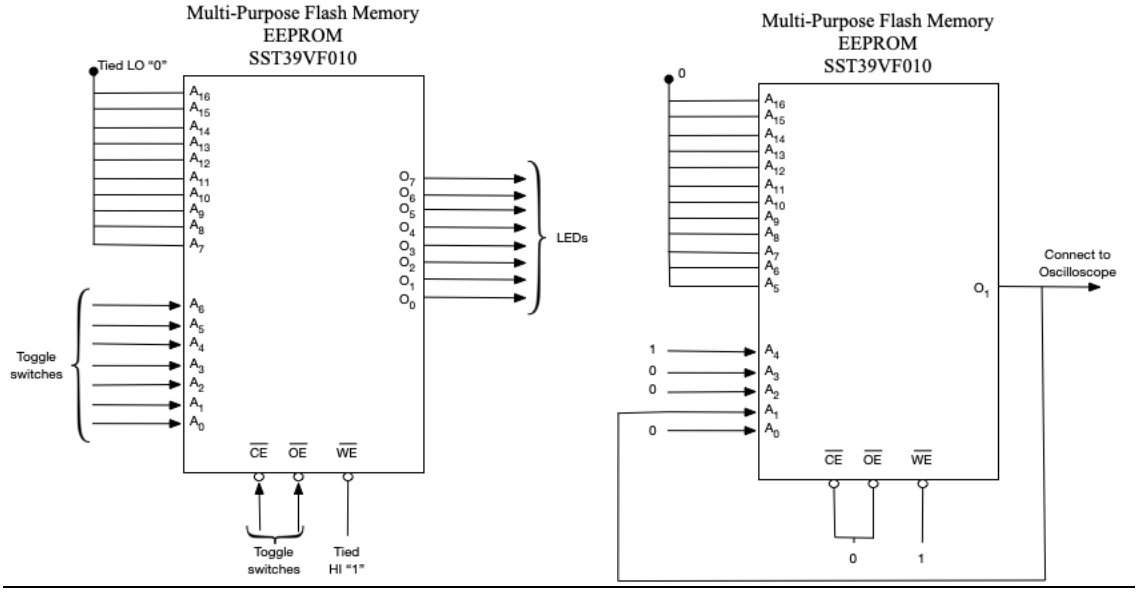
4.5.4 EEPROM: Electrically Erasable Programmable Read Only Memory

Milestone 3: TA to check EEPROM Binary-to-BCD operation

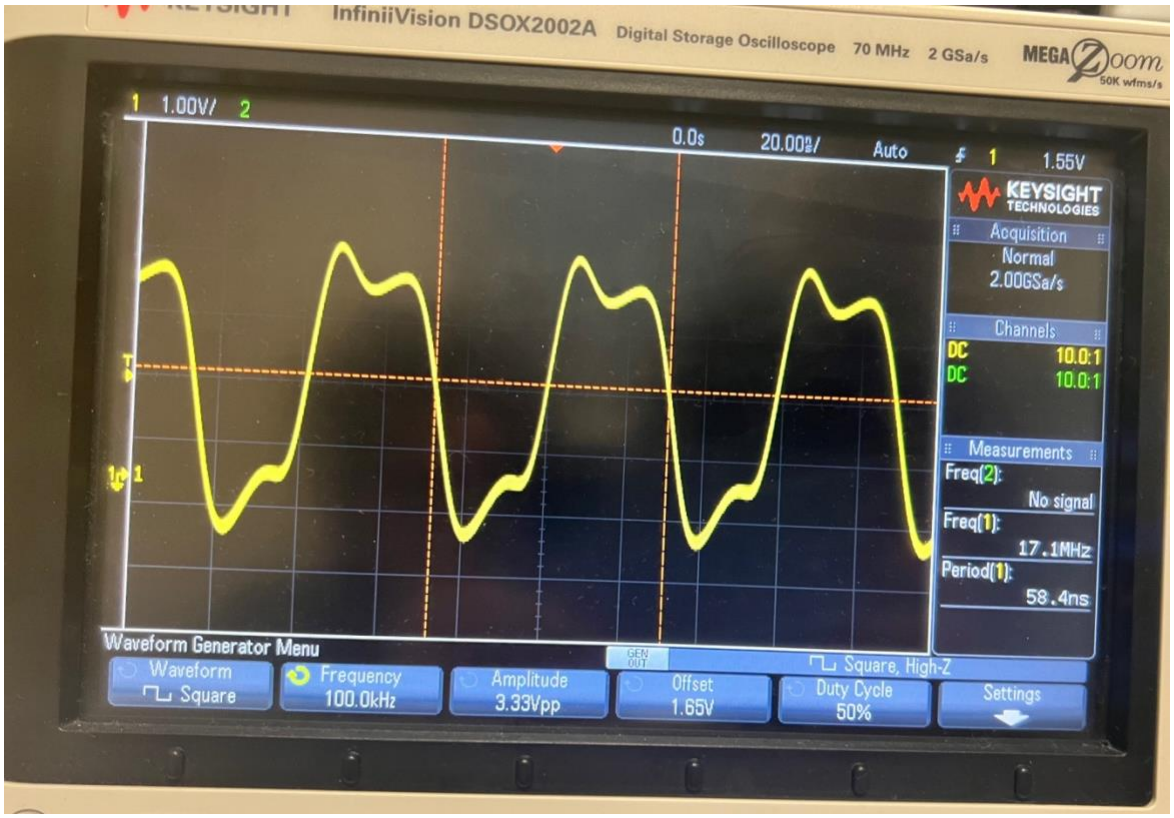
Verify the operation of the device and the conversion

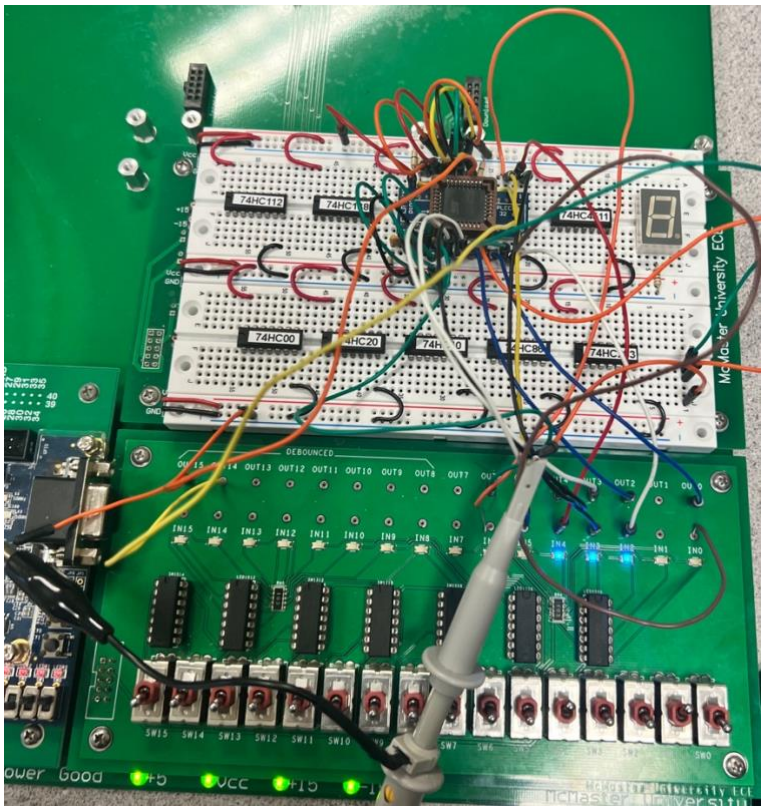


Milestone 4: TA to check EEPROM Access Operation



Access time for peripheral digital devices is always a concern to the digital designer. When the CE' and OE'
lines are tied LO, one way to measure this EEPROM's access time t_{ACC} *is the delay between i) the application*
of an address input, and ii) the appearance of stable data outputs. Configure the EEPROM
circuit as shown in Figure 4.9 and observe the output from pin O1 on the oscilloscope. Using this waveform,
estimate the actual t_{ACC} *for the SST39VF010 EEPROM. Sketch (or take a picture of the scope and annotate)* ***and explain your observations.***




74HC112
74HC00
DEBOUNCED
IN15 IN14 IN13 IN12 IN11 IN10 IN9 IN8 IN7
SW15 SW14 SW13 SW12 SW11
Power Good
+5
VCC
+15

4.6 Bonus Milestone: HDL Simulation of Circular Shift Register

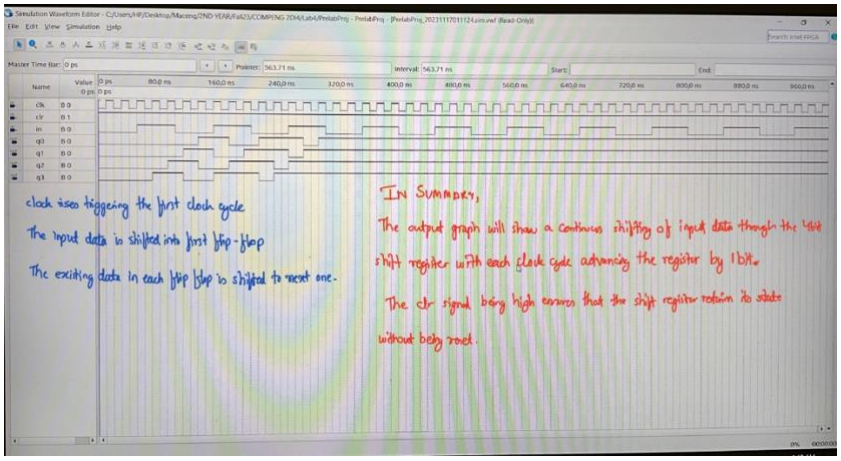
For this bonus include in the report:

a) *Copy of design entry with rationale for approach [4 marks]*

Bonus.v* | Compilation Report - PrelabProj

```
1   //Abaan Khan - 400428399
2   //Mustafa Shahid - 400440384
3
4
5   module Bonus(clk,clr,in,q0,q1,q2,q3);
6      input clk,clr,in;
7      output q0,q1,q2,q3;
8      wire in1;
9
10
11     assign in1 = in || q0;
12     DFF one (
13              .d(in1),
14              .clk(clk),
15              .clrn(clr),
16              .prn(1),
17              .q(q3)
18              );
19
20     DFF two (
21              .d(q3),
22              .clk(clk),
23              .clrn(clr),
24              .prn(1),
25              .q(q2)
26              );
27
28     DFF three (
29              .d(q2),
30              .clk(clk),
31              .clrn(clr),
32              .prn(1),
33              .q(q1)
34              );
35
36     DFF four (
37              .d(q1),
38              .clk(clk),
39              .clrn(clr),
40              .prn(1),
41              .q(q0)
42              );
43  endmodule
44
```

b) *An annotated screenshot of functional simulation results with all combinations of bit inputs. The annotation should illustrate your understanding of the connection between the inputs and outputs. [2 marks],*



c) *A short description your testing and verification of the circuit [4 marks].*

The output graph will show a continuous shiting of input data through the 4 bit shift register with each clock cycle advancing the register by 1 bit. The clr signal being high ensures that the shift register retains its state without being reset.