Project Report

Mini Computational Unit (MCU)

Submitted by:

Usman Siddique (18L-1293)

Mahad Naveed Qureshi (18L-1039)

Section:

DLD lab C1

Submitted to:

Ma’am Saba Zahid

Dated:

10th June,2020

Introduction:

A register includes a set of flip-flops. Since each flip-flop is capable of storing one bit of information, an n-bit register, composed of n flip-flops, is capable of storing n bits of binary information. By the broadest definition, a register consists of a set of flip-flops, together with gates that implement their state transitions. More commonly, the term register is applied to a set of flip-flops, possibly with added combinational, gates that perform data-processing tasks. The flip-flops hold data, and the gates determine the new or transformed data to be transferred into the flip-flops.

Objective:

We were assigned to make an MCU (mini-computational unit) that can perform tasks specified:

- **Data Transfer:** Transfer binary data from one register to other and load binary data in a register.
- **Arithmetic:** Performs arithmetic operations on data stored in registers.
- **Logic:** Perform bit manipulation on data in registers.

Procedure (Design):

According to the assigned OPCODE we saw the task to be implemented and implemented it according to the outputs of the 3 flip flops that will be Q3 Q2 Q1 respectively. As we are assigned a 3-bit register. Next states of each state are taken to calculate the input of the registers. Equations were generated for each task. At first, we did **Parallel Loading** into R0 then R1 from the user. Since a data selector was needed to give the output of each specified OPCODE so we used 6 flip flops and for each flip flop 2 MUX so a total of 12 MUX-8 (2 sets of 6 for R0 and R1). We implement the outputs through Karnaugh maps, there were 3 inputs lines hence 8 resulting configurations we made to generate the outputs. We take the next state (Qn+) and look for where it is 1 and optimize it using a 3x3(6 variable) map with Q3 Q2 Q1 as outputs of flip flop and X Y Z as inputs. We can see the output results in a multiplexer equation. We then implemented **Move** which is actually a feedback path to the MUX. As we assign a group of 6 MUX to each register that matches D3 D2 and D1 we have a proper moving operation, that plugs the input back into the desired MUX as the selection line with match the OP code of move. Next, we did the **Addition** by using an adder that incremented the input by 1. Next the **Increment and Up counter** are performed together as they generate the same next state for the same output and see to it that it matches the respective OP that has the MUX enabled. State table for R0 and R1 is made from which maps were constructed and equation was made for D1 D2 and D3. The circuit was implemented. Then **XOR** sub operation was implemented as the desired output lines of each register are placed into their respective XOR gates and then its output is sent to register R0. Lastly, **Circular Shift** which is done by moving 1 bit from right to left most side. We optimize it and take the respective outputs from the register and place it in the input line where the selection line matches the opcode.

S_3 S_2 S_1 S_0 For R_0 :

Q_3	Q_2	Q_1	Q_3^+	Q_2^+	Q_1^+	D_3	D_2	D_1
0	0	0	0	0	1	0	0	1
0	0	1	0	1	0	0	1	0
0	1	0	0	1	1	0	1	1
0	1	1	1	0	0	1	0	0
1	0	0	1	0	1	1	0	1
1	0	1	1	1	0	1	1	0
1	1	0	1	1	1	1	1	1
1	1	ϕ	0	0	0	0	0	0

$Q_1 \backslash Q_3Q_2$	00	01	11	10
0			1	1
1		1		1

$$D3 = \overline{Q_3} Q_2 Q_1 + \overline{Q_1} Q_3 + Q_3 \overline{Q_2}$$

$Q_1 \backslash Q_3Q_2$	00	01	11	10
0		1	1	
1	1			1

$$D_2 = \overline{Q_1} Q_2 + Q_1 \overline{Q}$$
$$= Q_1 \oplus Q_2$$

$Q_1 \backslash Q_3Q_2$	00	01	11	10
	1	1	1	1

$$D_1 = \overline{Q_1}$$

$S_3\ S_2\ S_1\ S_0$ For R_1 :

Q_3	Q_2	Q_1	Q_3^+	Q_2^+	Q_1^+	D_3	D_2	D_1
0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	1	0	0
0	1	0	0	0	1	0	0	1
0	1	1	1	0	1	1	0	1
1	0	0	0	1	0	0	1	0
1	0	1	1	1	0	1	1	0
1	1	0	0	1	1	0	1	1
1	1	1	1	1	1	1	1	1

$Q_1 \backslash Q_3Q_2$	00	01	11	10
0				
1	1	1	1	1

$D_3 = Q_1$

$Q_1 \backslash Q_3Q_2$	00	01	11	10
00			1	1
1			1	1

$D_2 = Q_3$

$Q_1 \backslash Q_3Q_2$	00	01	10	11
0		1	1	
1		1	1	

$D_1 = Q_2$

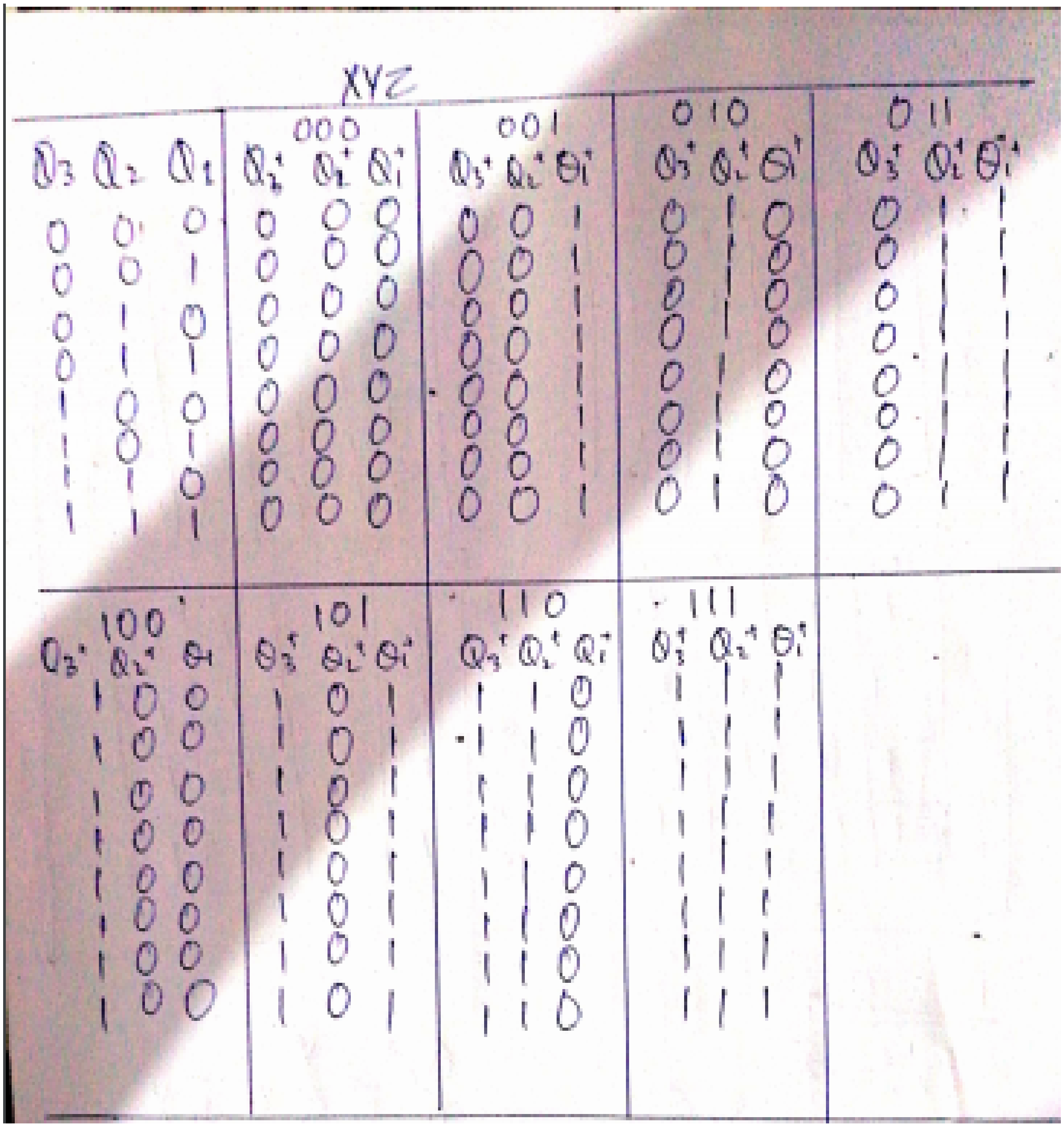
, $S_3\ S_2\ S_1\ S_0$ → feedback path to MUX from out of R_1

$S_3 S_2 \bar{S}_1 \bar{S}_0 \rightarrow Q_3^{R_0} \oplus Q_3^{R_1}$

$Q_2^{R_0} \oplus Q_2^{R_1}$

$Q_1^{R_0} \oplus Q_2^{R_0}$

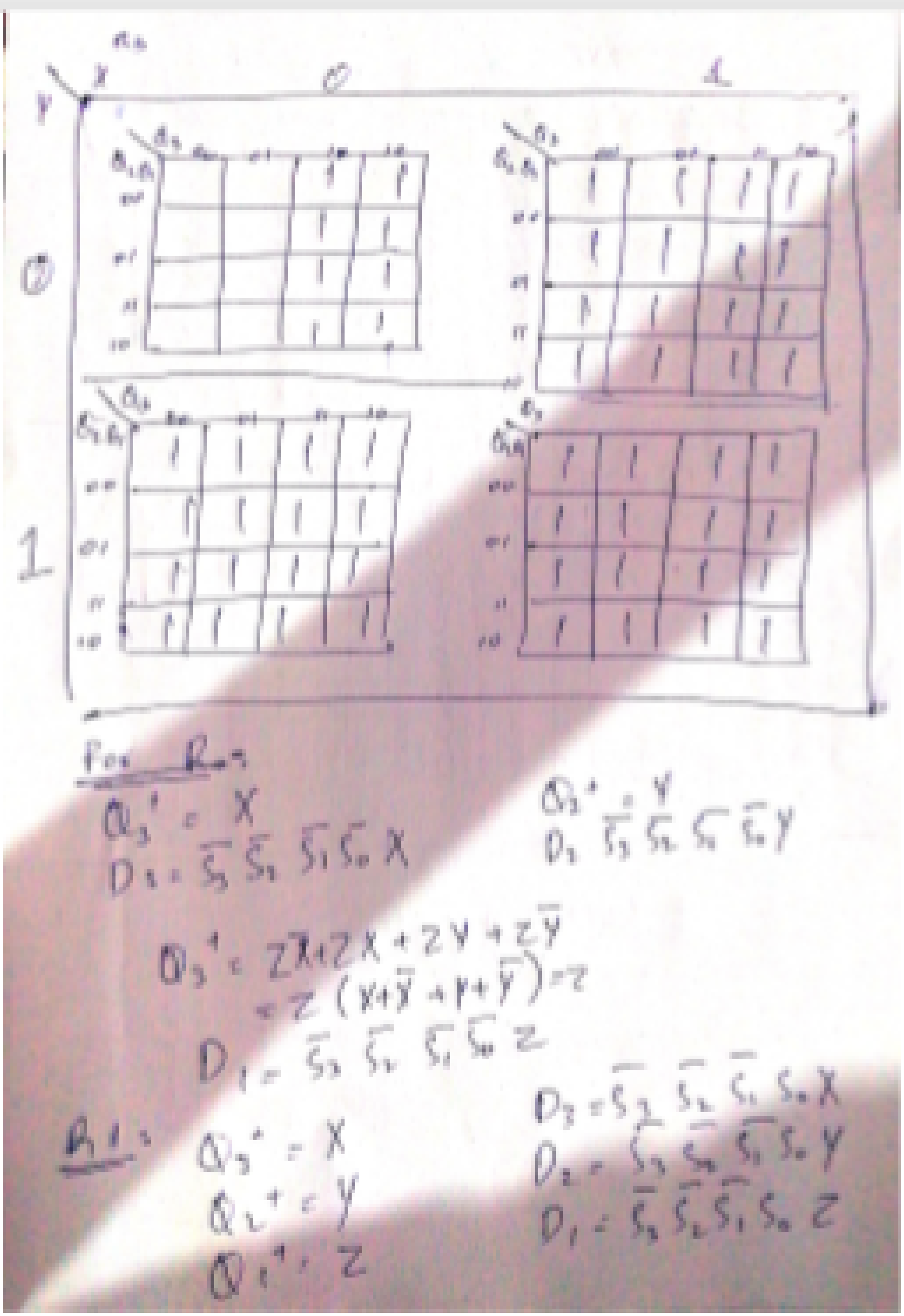
$\bar{S}_3 S_2 \bar{S}_1 \bar{S}_0 \rightarrow$ Adder



XYZ

			000			001			010			011		
Q_3	Q_2	Q_1	Q_3^+	Q_2^+	Q_1^+	Q_3^+	Q_2^+	Q_1^+	Q_3^+	Q_2^+	Q_1^+	Q_3^+	Q_2^+	Q_1^+
0	0	0	0	0	0	0	0	1	0	1	0	0	1	1
0	0	1	0	0	0	0	0	1	0	1	0	0	1	1
0	1	0	0	0	0	0	0	1	0	1	0	0	1	1
0	1	1	0	0	0	0	0	1	0	1	0	0	1	1
1	0	0	0	0	0	0	0	1	0	1	0	0	1	1
1	0	1	0	0	0	0	0	1	0	1	0	0	1	1
1	1	0	0	0	0	0	0	1	0	1	0	0	1	1
1	1	1	0	0	0	0	0	1	0	1	0	0	1	1

100			101			110			111		
Q_3^+	Q_2^+	Q_1	Q_3^+	Q_2^+	Q_1^+	Q_3^+	Q_2^+	Q_1^+	Q_3^+	Q_2^+	Q_1^+
1	0	0	1	0	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1
1	0	0	1	0	1	1	1	0	1	1	1



R_0

Y \ X: 0, 1

Y = 0, X = 0

Q_2Q_1 \ Q_3	00	01	11	10
00			1	1
01			1	1
11			1	1
10			1	1

Y = 0, X = 1

Q_2Q_1 \ Q_3	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

Y = 1, X = 0

Q_2Q_1 \ Q_3	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

Y = 1, X = 1

Q_2Q_1 \ Q_3				
00	1	1	1	1
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

For R_0:

$Q_3^+ = X$

$D_3 = \bar{S_3}\,\bar{S_2}\,\bar{S_1}\,\bar{S_0}\,X$

$Q_2^+ = Y$

$D_2\ \bar{S_3}\,\bar{S_2}\,\bar{S_1}\,\bar{S_0}\,Y$

$Q_1^+ = Z\bar{X} + ZX + ZY + Z\bar{Y}$

$= Z\,(X+\bar{X}+Y+\bar{Y}) = Z$

$D_1 = \bar{S_3}\,\bar{S_2}\,\bar{S_1}\,\bar{S_0}\,Z$

R1:

$Q_3^+ = X$

$Q_2^+ = Y$

$Q_1^+ = Z$

$D_3 = \bar{S_3}\,\bar{S_2}\,\bar{S_1}\,S_0\,X$

$D_2 = \bar{S_3}\,\bar{S_2}\,\bar{S_1}\,S_0\,Y$

$D_1 = \bar{S_3}\,\bar{S_2}\,\bar{S_1}\,S_0\,Z$

Issues:

No serious issues were faced during the execution of the project. We both were able to complete almost all the tasks in the given project.

Applications:

Microcontrollers are used in automatically controlled products and devices, such as automobile engine control systems, implantable medical devices, remote controls, office machines, appliances, power tools, toys and other embedded systems. The main thing used in this mini controller were registers that are also used to make digital memory chips like ROM Chips. Cache memory in CPU is also made by registers.

Conclusion:

We were able to design an MCU that implement the tasks of Data Transfer, Arithmetic operations by using registers mainly, the experiment was executed successfully.