ISTANBUL TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BLG 242E

DIGITAL CIRCUITS LABORATORY EXPERIMENT REPORT

EXPERIMENT NO : 3

EXPERIMENT DATE : 01.03.2019

LAB SESSION : FRIDAY - 14.00

GROUP NO : G13

GROUP MEMBERS:

150180704 : CİHAT AKKİRAZ

150180707 : FATİH ALTINPINAR

150180734 : SİNAN ŞAR

SPRING 2019

ETHIC FORM
for
BLG242E Logic Circuits Laboratory

As a student of

Istanbul Technical University Faculty of Computer and Informatics Engineering;

1. I will not attempt to cheat in quizes and final exam,
2. I will not use disallowed sources or tools (mobile phone, calculator etc.) during the exam,
3. I will not write any information (formula, text, figure etc.) on the table, sheets or books that are allowed to be used during the exam,
4. I will give reference when using printed or online published sources,
5. I will not use the results in a source as they are, or by changing a part of them without giving a reference,
6. I will not show unused sources as used,
7. I will not present someone elses idea as my own idea,
8. I will not make someone do my homework, project or thesis for money or anything else,
9. I will not take an exam or enter a lecture on behalf of others,
10. I will not make excuses for not attending in exams or lessons by taking reports from someone I know (medical doctor parents or relatives),
11. I will refrain from deliberately harming the public materials at our university,
12. I will comply with the safety rules in laboratory work,
13. I will behave in accordance with the rules of respect for the lecturers and teaching assistants

signed by

150180704 : CİHAT AKKİRAZ

150180707 : FATİH ALTINPINAR

150180734 : SİNAN ŞAR

Contents

1 INTRODUCTION

In this experiment we have tried to find prime implicants of some functions and implement them with the lowest cost possible, using combinational logic circuits, NOT, AND, OR, NAND, NOR gates, a decoder and a multiplexer provided to us.

2 REQUIREMENTS

Tools Used[1]

- C.A.D.E.T
- 74000 series ICs
  - 74xx00 - Quadruple 2-input Positive NAND Gates
  - 74xx04 - Hex Inverters
  - 74xx08 - Quadruple 2-input Positive AND Gates
  - 74xx10 - Triple 3-input Positive NAND Gates
  - 74xx11 - Triple 3-input Positive AND Gates
  - 74xx27 - Triple 3-input Positive NOR Gates
  - 74xx32 - Quadruple 2-input Positive OR Gates
  - 74xx138 - 3:8 Decoder
  - 74xx151 - 8:1 Multiplexer

2.1 PART 1

In this part, the circuit is designed for the given function (The function can be seen below). Before building the circuit, its karnaugh diagram is created to find the essential prime implicants of the function. Sum of essential prime implicants gives the expression with minimum cost. The circuit is implemented by using AND, OR and NOT gates. Connections between pins are made for each integrated circuits as needed, outputs ae observed on the logic monitor.

- F_1(a,b,c,d) = U_1(0, 3, 5, 7, 11, 12, 13) + U_Φ(1, 8, 15)

While finding essential prime implicants don't care(Φ) input values are considered as true(1), while creating the prime implicant chart, they are considered as false(0).

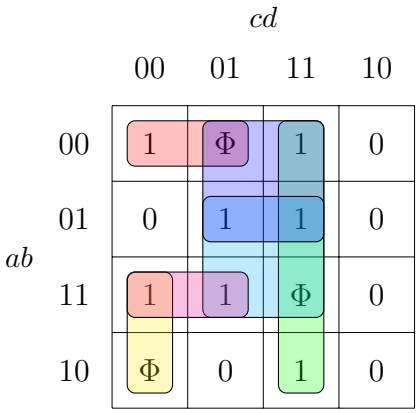

Figure 1: Karnaugh map of F

	$c \cdot d$	$a' \cdot d$	$b \cdot d$	$a' \cdot b' \cdot c'$	$a \cdot b \cdot c'$	$a \cdot c' \cdot d'$
SYMBOL	A	B	C	D	E	F
COST	4	5	4	9	7	8
COVERED POINTS	3, 7, 11, 15	3, 5, 7	5, 7, 13	0	12, 13	12

Table 1: All prime implicants

After labeling, prime implicant chart is formed by putting a prime implicant on a row and marking every point that covered by that prime implicant.

	0	3	5	7	11	12	13	COST
A		x		x	x			4
B		x	x	x				5
C			x	x			x	4
D	x							9
E						x	x	7
F						x		12

Table 2: Prime implicant chart. Distinguished points are shown as red.

As we can see in the prime implicant chart(Table 2) distinguished points are shown as red. Prime implicants that are covering these points must be chosen. Therefore we chose A,C and D then remove them from the chart. We also remove any column that is covered by these prime implicants.

	12	COST
E	x	7
F	x	12

Table 3: Implicant chart after removing A,C and D and colums covered by them.

After removing A,D and C from the prime implicant chart we are left with E and F(Table 3). Since E covers the point that is covered by F and costs less we chose E as the final prime implicant. There is no point left to be covered.

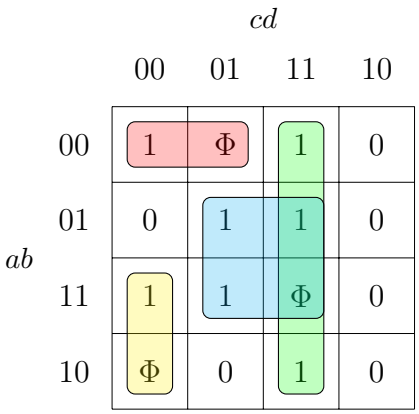

Figure 2: Chosen prime implicants on the Karnaugh map.

Thus we get the F function as:

- F(a,b,c) $= a' \cdot b' \cdot c' + a \cdot b \cdot c' + b \cdot d + c \cdot d$

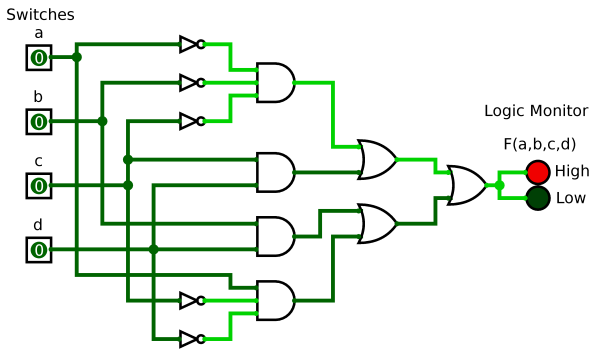

Figure 3: Circuit of $F(a, b, c, d)$

A	B	C	D	F
0	0	0	0	1
0	0	0	1	Φ
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	0
0	1	1	1	1
1	0	0	0	Φ
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	0
1	1	1	1	Φ

Table 4: Truth table of the given expression.

The output values of the circuit are validated with the truth table above. If a don't care points manages to get into a prime implicant (as it can be seen in Figure 2) it gives an output of 1 otherwise 0. But we simply don't care it's value.

2.2 PART 2

In the second part of the experiment, same circuit is implemented using only NAND and NOT gates. Connections between pins are made for each integrated circuits as needed, outputs are observed on the logic monitor. The output values of the circuit are validated with the truth table in the part 1.

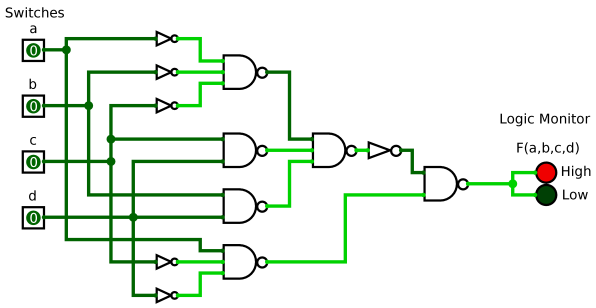

Figure 4: Circuit of F

2.3 PART 3

In the next part of the experiment, the function is designed using a single 8:1 multiplexer and NOT gates. Connections between pins are made for each integrated circuits as needed, outputs are observed on the logic monitor.

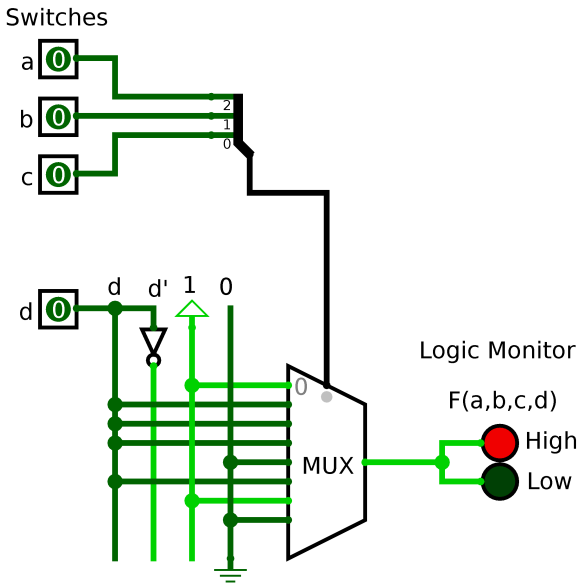

Figure 5: Circuit of F

A	B	C	D	F	
0	0	0	0	1	1
0	0	0	1	Φ	
0	0	1	0	0	d
0	0	1	1	1	
0	1	0	0	0	d
0	1	0	1	1	
0	1	1	0	0	d
0	1	1	1	1	
1	0	0	0	Φ	0
1	0	0	1	0	
1	0	1	0	0	d
1	0	1	1	1	
1	1	0	0	1	1
1	1	0	1	1	
1	1	1	0	0	0
1	1	1	1	Φ	

Table 5: Truth table for multiplexer implementation.

A, B and C input values are connected to the select lines(inputs) of the multiplexer. The remaining D input value is connected according to the truth table above to the data inputs of the multiplexer.

2.4 PART 4

In the final part of the experiment, the circuit is designed for the functions below. The circuit is implemented using a single 3:8 decoder, OR, and NOT gates. Connections between pins are made for each integrated circuits as needed, outputs are observed on the logic monitor.

- F_1(a,b,c) $= a' \cdot c' + b \cdot c$
- F_2(a,b,c) $= a' \cdot b' \cdot c' + a \cdot b$

The first circuit is designed as shown diagram below. The output values in the circuit are validated with the truth table below of the function.

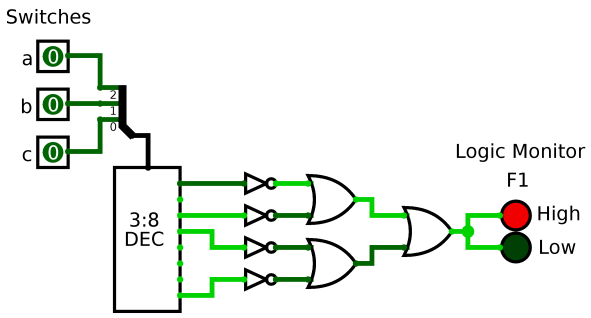

Figure 6: Circuit of F_1

a	b	c	a'	b'	c'	$a' \cdot c'$	$b \cdot c$	$F_1 = a' \cdot c' + b \cdot c$
0	0	0	1	1	1	1	0	1
0	0	1	1	1	0	0	0	0
0	1	0	1	0	1	1	0	1
0	1	1	1	0	0	0	1	1
1	0	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0	0
1	1	0	0	0	1	0	0	0
1	1	1	0	0	0	0	1	1

Table 6: Truth table for F1

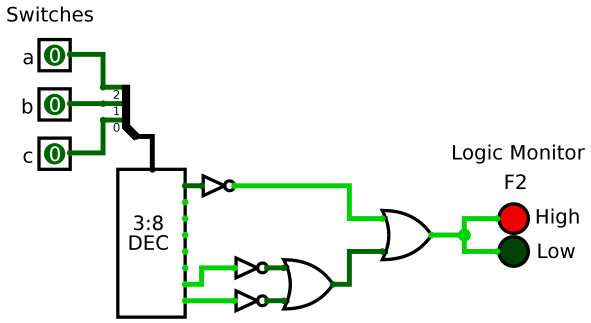

Figure 7: Circuit of F_2

The second circuit is designed as shown in the diagram(Figure 7). The output values in the circuit are validated with the truth table(Table 7).

a	b	c	a'	b'	c'	$a' \cdot b' \cdot c'$	$a \cdot b$	$F_2 = a' \cdot b' \cdot c' + a \cdot b$
0	0	0	1	1	1	1	0	1
0	0	1	1	1	0	0	0	0
0	1	0	1	0	1	0	0	0
0	1	1	1	0	0	0	0	0
1	0	0	0	1	1	0	0	0
1	0	1	0	1	0	0	0	0
1	1	0	0	0	1	0	1	1
1	1	1	0	0	0	0	1	1

Table 7: Truth table for F2

3 INTERPRETATION OF THE RESULTS

We were able to observe the desired-theoretically true results all through the experiment as we were able to figure out the correct designs for the circuits. Results were consistent with the expected result we calculated beforehand. We also faced no equipment failures. Other than the things mentioned above, there wasn't anything worth noting about our results.(Methods used and designs for each circuit have already been explained in detail in part 2 of the report.)

4 CONCLUSION

Even though most of the experiment went as expected, there were 2 points where we had to stop and think once more. First of which was part 2 of the experiment where we designed the circuit correctly but made a mistake while implementing it on the cadet unit. But we were able to resolve the issue with the cost being more time than necessary spent on the step. The second point was part 4, where we had to use a decoder. We had a problem figuring out what the inputs should be. But that confusion also didn't last long and we got back on track to finish the experiment as planned.

REFERENCES

[1] Istanbul Technical University Department of Computer Engineering. Blg 242e logic circuits laboratory experiments booklet version 1.9, Spring 2019.

[2] Overleaf documentation https://tr.overleaf.com/learn.

[3] Detailed info on writing reports https://projects.ncsu.edu/labwrite/res/res-studntintro-labparts.html.