ISTANBUL TECHNICAL UNIVERSITY

COMPUTER ENGINEERING DEPARTMENT

BLG 242E

DIGITAL CIRCUITS LABORATORY EXPERIMENT REPORT

EXPERIMENT NO	:	2
EXPERIMENT DATE	:	31.03.2023
LAB SESSION	:	FRIDAY - 14.00
GROUP NO	:	G3

GROUP MEMBERS:

150200054 : ASLI YEL

150190028 : SEVİM EFTAL AKŞEHİRLİ

SPRING 2023

Contents

1 INTRODUCTION

The aim of this report is to present the results of an experiment conducted to find the expression with the lowest cost for combinational logic circuits, and to detail the process of implementing these circuits using Boolean algebra and logic gates. By analyzing and simplifying the expressions, we were able to reduce the number of gates required for implementation, resulting in a more cost-effective design. Through this experiment, we gained a deeper understanding of the importance of minimizing cost in digital circuit design and the techniques used to achieve this goal.

2 MATERIALS AND METHODS

2.1 PRELIMINARY

2.1.1 Finding Prime Implicants

a) Finding Prime Implicants by using Karnaugh Diagram:

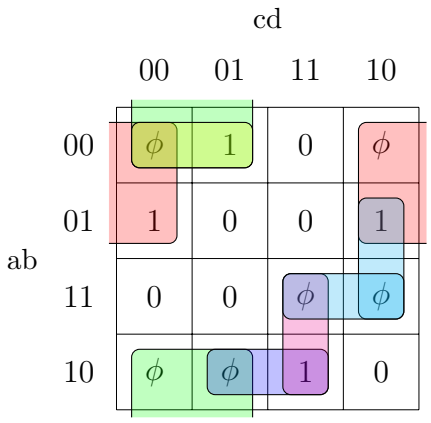

Figure 1: Karnaugh Diagram for the function
F(a, b, c, d) = $\cup_1(1, 4, 6, 11) + \cup_\phi(0, 2, 8, 9, 14, 15)$

All prime implicants of F created on the Karnaugh Diagram: $\bar{a}\bar{d}$, $\bar{b}\bar{c}$, abc, a$\bar{b}$d, acd, bc$\bar{d}$.

b) Finding Prime Implicants by using Quine-McCluskey Method:

Minterm	Binary Representation
m(0)	0000
m(1)	0001
m(2)	0010
m(4)	0100
m(6)	0110
m(8)	1000
m(9)	1001
m(11)	1011
m(14)	1110
m(15)	1111

Table 1: Minterms and their Binary Representations

Num. of 1s	Minterms	a	b	c	d	State
0	m(0)	0	0	0	0	✔
1	m(1)	0	0	0	1	✔
	m(2)	0	0	1	0	✔
	m(4)	0	1	0	0	✔
	m(8)	1	0	0	0	✔
2	m(6)	0	1	1	0	✔
	m(9)	1	0	0	1	✔
3	m(11)	1	0	1	1	✔
	m(14)	1	1	1	0	✔
4	m(15)	1	1	1	1	✔

Table 2: Minterms and Corresponding Input Combinations

Num. of 1s	Size 2 Implicants	a	b	c	d	State
0	m(0), m(1)	0	0	0	-	✔
	m(0), m(2)	0	0	-	0	✔
	m(0), m(4)	0	-	0	0	✔
	m(0), m(8)	-	0	0	0	✔
1	m(1), m(9)	-	0	0	1	✔
	m(2), m(6)	0	-	1	0	✔
	m(4), m(6)	0	1	-	0	✔
	m(8), m(9)	1	0	0	-	✔
2	m(6), m(14)	-	1	1	0	✗
	m(9), m(11)	1	0	-	1	✗
3	m(11), m(15)	1	-	1	1	✗
	m(14), m(15)	1	1	1	-	✗

Table 3: Size 2 Implicants and Corresponding Input Combinations

Num. of 1s	Size 4 Implicants	a	b	c	d	State
0	m(0), m(1), m(8), m(9)	-	0	0	-	✗
	m(0), m(2), m(4), m(6)	0	-	-	0	✗

Table 4: Size 4 Implicants and Corresponding Input Combinations

All prime implicants of F obtained by utilizing the Quine-McCluskey Method: $\bar{a}\bar{d}$, $\bar{b}\bar{c}$, abc, a$\bar{b}$d, acd, bc$\bar{d}$.

2.1.2 Finding the Lowest Cost Expression

The lowest cost expression of F can be derived by using Prime Implicant Chart

	$\bar{a}\bar{d}$	$\bar{b}\bar{c}$	abc	a$\bar{b}$d	acd	bc$\bar{d}$
Symbol:	A	B	C	D	E	F
Cost:	6	6	6	7	6	7
Covered True Points:	4,6	1	X	11	11	6

Table 5: Costs of and True Points Covered by Prime Implicants

	1	4	6	11	*Cost*
A		*X*	*X*		6
B	*X*				6
C					6
D				*X*	7
E				*X*	6
F			*X*		7

Table 6: Prime Implicant Chart - Step1

	11	*Cost*
D	*X*	7
E	*X*	6

Table 7: Prime Implicant Chart - Step2

Selected prime implicants: A($\bar{a}\bar{d}$), B($\bar{b}\bar{c}$), E(acd)

Total Cost: $7 + 6 + 6 = 19$

F_1(a, b, c, d) = a$\bar{b}$d + $\bar{a}\bar{d}$ + $\bar{b}\bar{c}$

2.1.3 Implementing F with AND, OR and NOT Gates

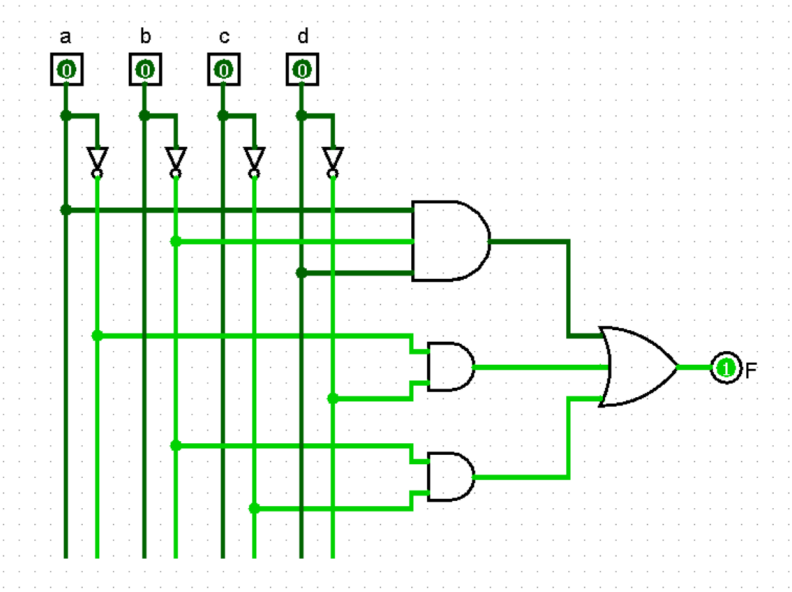

Figure 2: Design of F with AND, OR and NOT Gates

2.1.4 Implementing F with a single 8:1 MUX and NOT Gates

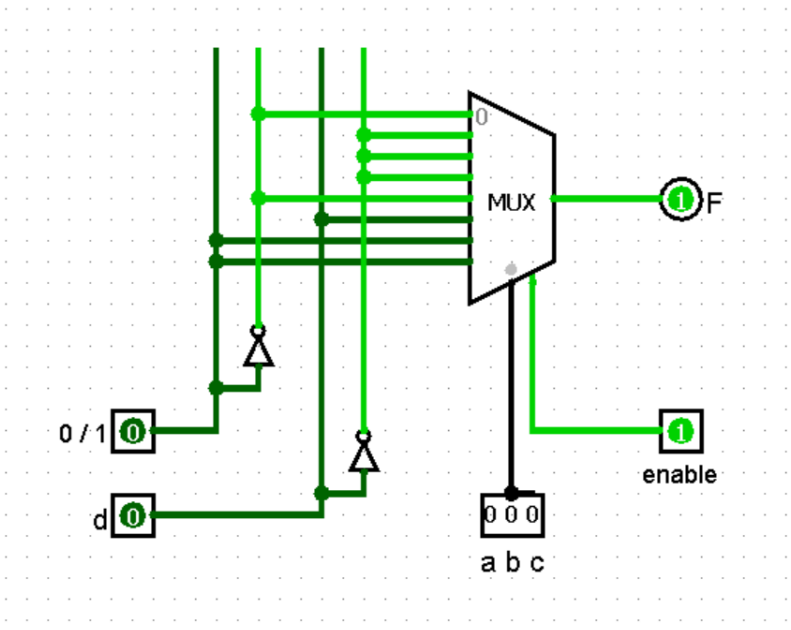

Figure 3: Design of F with a single 8:1 MUX and NOT Gates

2.2 MATERIALS

The materials to be used in this experiment depending on the function F are as follows:

- C.A.D.E.T. (Complete Analogue Digital Electronic Trainer)
- 74000 series ICs
  - 74xx00 - Quadruple 2-input Positive NAND Gates
  - 74xx04 - Hex Inverters
  - 74xx00 - 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

The ones we prioritize to use:

- C.A.D.E.T. (Complete Analogue Digital Electronic Trainer)
- 74000 series ICs
  - 74xx04 - Hex Inverters
  - 74xx00 - Quadruple 2-input Positive AND Gates
  - 74xx11 - Triple 3-input Positive AND Gates
  - 74xx32 - Quadruple 2-input Positive OR Gates
  - 74xx151 - 8:1 Multiplexer

3 RESULTS

In order to evaluate how our implementations work, we created the truth table of the function F as follows:

index	a	b	c	d	a'	b'	c'	d'	ab'd	a'd'	b'c'	(ab'd + a'd' + b'c')	given F
0	0	0	0	0	1	1	1	1	0	1	1	1	ϕ
1	0	0	0	1	1	1	1	0	0	0	1	1	1
2	0	0	1	0	1	1	0	1	0	1	0	1	ϕ
3	0	0	1	1	1	1	0	0	0	0	0	0	0
4	0	1	0	0	1	0	1	1	0	1	0	1	1
5	0	1	0	1	1	0	1	0	0	0	0	0	0
6	0	1	1	0	1	0	0	1	0	1	0	1	1
7	0	1	1	1	1	0	0	0	0	0	0	0	0
8	1	0	0	0	0	1	1	1	0	0	1	1	ϕ
9	1	0	0	1	0	1	1	0	1	0	1	1	ϕ
10	1	0	1	0	0	1	0	1	0	0	0	0	0
11	1	0	1	1	0	1	0	0	1	0	0	1	1
12	1	1	0	0	0	0	1	1	0	0	0	0	0
13	1	1	0	1	0	0	1	0	0	0	0	0	0
14	1	1	1	0	0	0	0	1	0	0	0	0	ϕ
15	1	1	1	1	0	0	0	0	0	0	0	0	ϕ

Table 8: Truth Table For The Function(F)

In both parts of the experiment, we observed the results of the circuits that we implemented according to this truth table.

3.1 Experiment - Part 1

In the first part of the experiment we implemented the lowest cost expression that we obtained previously with AND, OR and NOT gates as seen in the following picture.

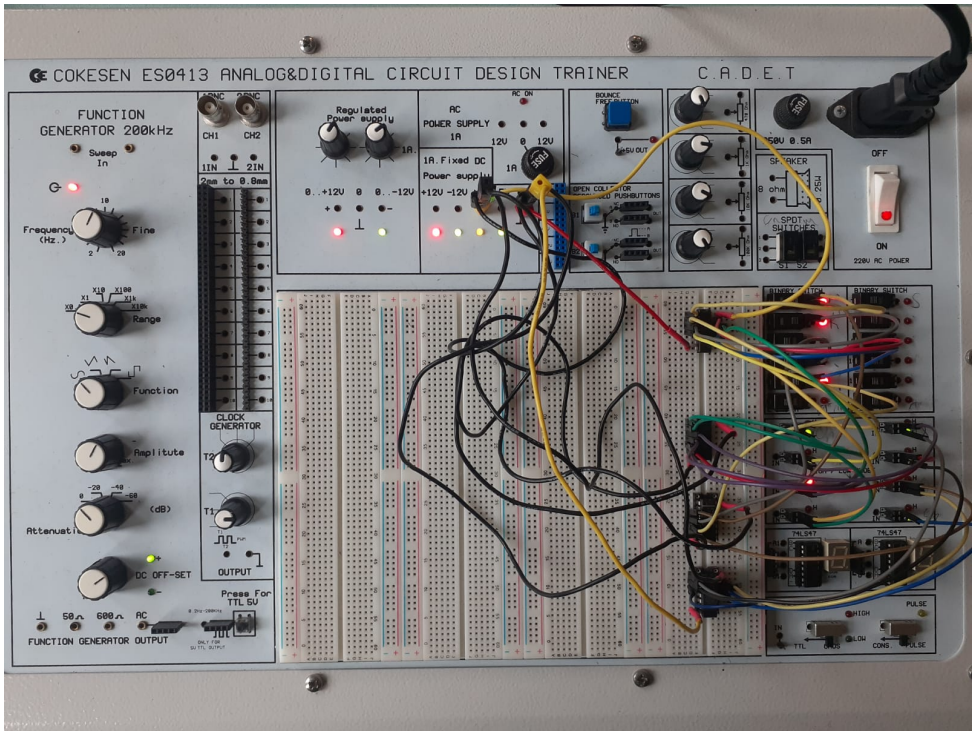

Figure 4: Design of F with AND, OR and NOT Gates

We confirmed that our implementation operates as expected by comparing the results for each input combination with the corresponding value in the truth table that we created earlier.

3.2 Experiment - Part 2

Unfortunately, we have got unsatisfied results for this part of the experiment that we tried to build the same function with a single 8:1 Multiplexer and NOT gates. Because of some troubles we have had during our implementation, which are mentioned in the 'Discussion' section, we were able to get the desired results only after the time of the experiment was up, although we had more than enough time when we started this part. Therefore we do not have a picture that will confirm our work for Part 2 of the experiment.

4 DISCUSSION

While finding the prime implicants with Karnaugh Diagram and Quine-McCluskey Methods, we accepted undetermined values as 1 in order to form larger groupings of minterms that we would not be able to get if treated them as 0. Later, we removed prime implicants that only includes undetermined values while trying to create the lowest cost expression since they are not essential to cover all true generating points and therefore would cause unnecessary increase of cost. But still some prime implicants also include undetermined values and so we may encounter 1 and 0 values, although it is not important for the system.

5 CONCLUSION

We got the lowest cost expression to get exact values of function. We thought of the gates we should use and we planned the process. First we formed our literals of a, b, c, d variables and obtained the literals of complemented variables by using NOT gate. Then by using 2 input and 3 input AND gates we build up our prime implicants in the lowest cost expression. Then with 2 input OR gates we implemented our expression of SOP (Sum of Products) form. We wanted to use a single 3 input OR gate, but we could not since we did not have one. According to the truth table, we got the exact values.

We were supposed to implement our expression also with a 8:1 multiplexer and Not gates. When we try to implement the multiplexer for our circuit, we encountered some difficulties. At first we did not send any signal to the Enable input of the multiplexer, since we were not aware of that we should indeed, without considering the internal structure of an IC. Until fixing this issue we have gotten '0' as output for all input combinations. After that, our circuit still was not working as expected. We thought we had a contact problem with the cables and we tried to fix it. Actually there were some contact problems but this was not the main problem. When we re-examined it with the supervisor assistants in the laboratory afterwards, we realized that we were connecting our selection inputs in reverse order. Meanwhile our multiplexer was removed from the CADET and our course assistant were checking whether it works correctly or not with a device. Since it was working flawlessly and we finally realized our actual mistake, we got back to our work to finish it. And this time we had placed the multiplexer in reverse on the CADET, so we had mismatched VCC and GND signals and all the other inputs. After realizing this mistake and correcting it, our circuit worked just as it should eventually.

With this experiment we remembered the ways to find prime implicant such as Karnaugh Diagram and Quine-McCluskey Method and we try to obtain the lowest cost expression with Prime Implicant Chart to ensure the F function.

We have gained more experience using CADET and certain ICs. We reverse-connected the Multiplexer to CADET and tried to find our mistake by evaluating the wrong results we got in the last part of the experiment. We also noticed that we had a problem with the MSB and LSB rankings.

REFERENCES