DLD PROJECT REPORT (CS-B)

GROUP MEMBERS:

1. SHEHERYAR RAMZAN (20I-0441)
2. FAHAD RAMZAN (20I-0443)

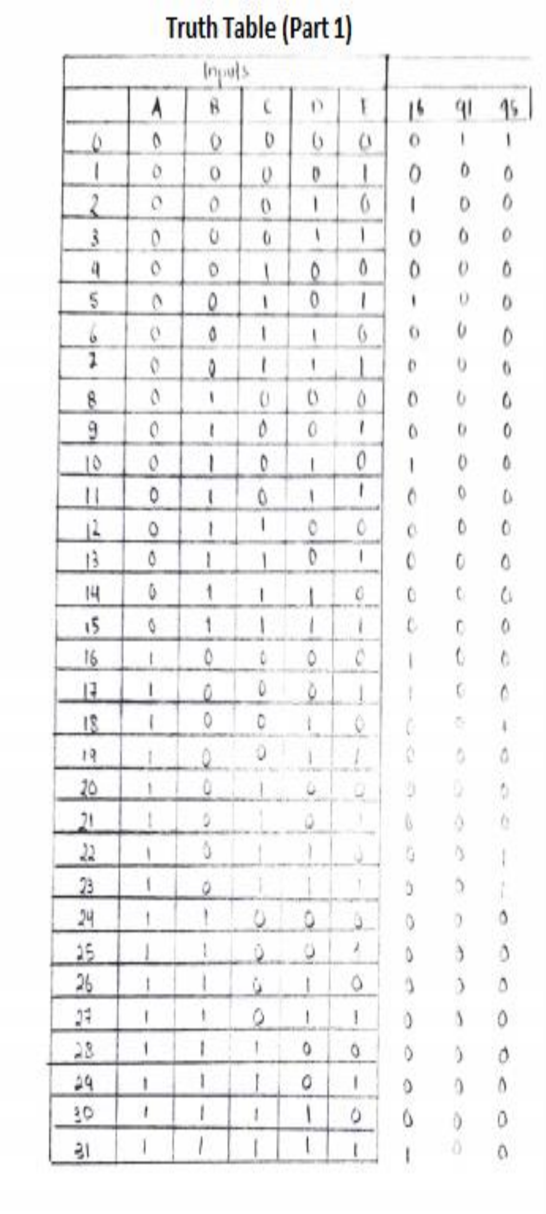
Components Used

- **Component** (2D Graphics Box Mode)
- **Logic Toggle** (Component Mode)
- **Red LED** (Component Mode)
- **Default** (Device Pin Mode)
- **Ground** (Terminal Mode)
- **NOT** (Component Mode)
- **AND** (Component Mode)
- **AND_5** (Component Mode)
- **OR_4** (Component Mode)
- **OR_3** (Component Mode)
- **OR_5** (Component Mode)
- **Input** (Terminals Mode)
- **Output** (Terminals Mode)
- **Logic Probe** (Component Mode)

Implementation Details (Part 1)

First of all, we started off by designing some kind of logic related to the problem given as we needed to convert a 7-segment display to a 100-segment display. So, eyeing the given document named as 32-Patterns, we devised a logic and saw that our given LED's; 16, 19, and 95 who were present to add up a part to make up a number. For example, to make 0 on a 100-segment display Led 16 was not in use whereas LED 91 and 95 were in use. Similarly, if we see which of our LED's make up number 10; we see that only one of our 3 LED's (16) is present int the given pattern. In the similar fashion we filled up the truth table. Secondly, we built K-maps from the truth table and derived a simplified expression. Thirdly, using proteus we made our own component and device. We attached a hierarchy model and dived into the child sheet. In the child sheet, after designing our circuit and after completing the circuit we exited to the parent sheet. Lastly, we attached logic toggles and logic probe to double checked our result using our truth table.

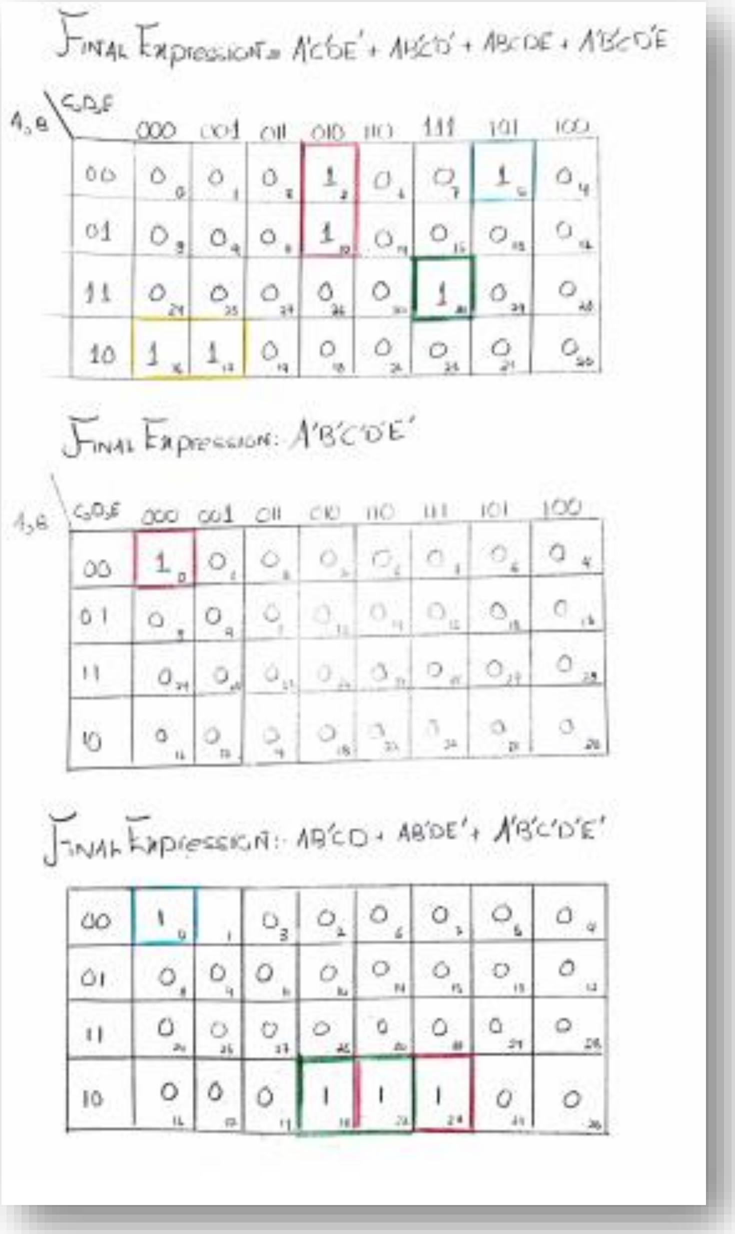
Truth Table (Part I)



Truth Table (Part 1)

Inputs								
	A	B	C	D	E	16	91	95
0	0	0	0	0	0	0	1	1
1	0	0	0	0	1	0	0	0
2	0	0	0	1	0	1	0	0
3	0	0	0	1	1	0	0	0
4	0	0	1	0	0	0	0	0
5	0	0	1	0	1	1	0	0
6	0	0	1	1	0	0	0	0
7	0	0	1	1	1	0	0	0
8	0	1	0	0	0	0	0	0
9	0	1	0	0	1	0	0	0
10	0	1	0	1	0	1	0	0
11	0	1	0	1	1	0	0	0
12	0	1	1	0	0	0	0	0
13	0	1	1	0	1	0	0	0
14	0	1	1	1	0	0	0	0
15	0	1	1	1	1	0	0	0
16	1	0	0	0	0	1	0	0
17	1	0	0	0	1	1	0	0
18	1	0	0	1	0	0	0	1
19	1	0	0	1	1	0	0	0
20	1	0	1	0	0	0	0	0
21	1	0	1	0	1	0	0	0
22	1	0	1	1	0	0	0	1
23	1	0	1	1	1	0	0	1
24	1	1	0	0	0	0	0	0
25	1	1	0	0	1	0	0	0
26	1	1	0	1	0	0	0	0
27	1	1	0	1	1	0	0	0
28	1	1	1	0	0	0	0	0
29	1	1	1	0	1	0	0	0
30	1	1	1	1	0	0	0	0
31	1	1	1	1	1	1	0	0

Handwritten K-Map (Part 1)



Final Expression= A'C'DE' + AB'C'D' + ABCDE + A'B'CD'E

A,B \ C,D,E	000	001	011	010	110	111	101	100
00	0	0	0	1	0	0	1	0
01	0	0	0	1	0	0	0	0
11	0	0	0	0	0	1	0	0
10	1	1	0	0	0	0	0	0

Final Expression: A'B'C'D'E'

A,B \ C,D,E	000	001	011	010	110	111	101	100
00	1	0	0	0	0	0	0	0
01	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0

Final Expression: AB'CD + AB'DE' + A'B'C'D'E'

00	1		0	0	0	0	0	0
01	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0
10	0	0	0	1	1	1	0	0

Figure 1: K-Map (Part 1)

Handmade Circuit Diagram for LED (16)

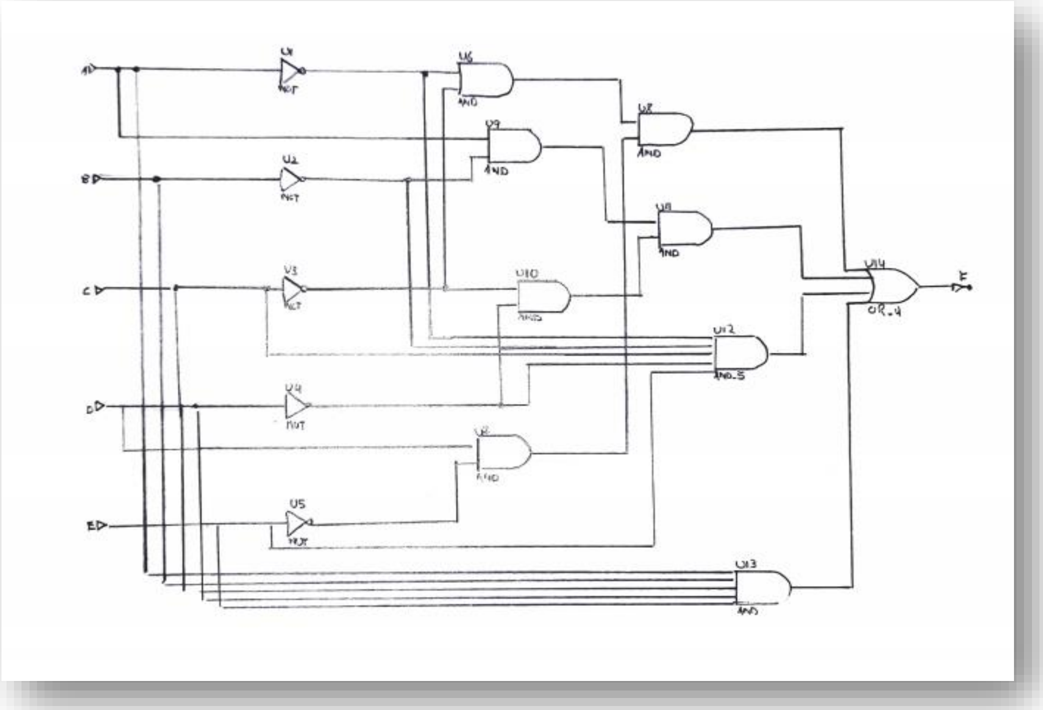


Figure 2: Handmade Circuit Diagram for LED 16

Proteus Diagram for (LED 16)

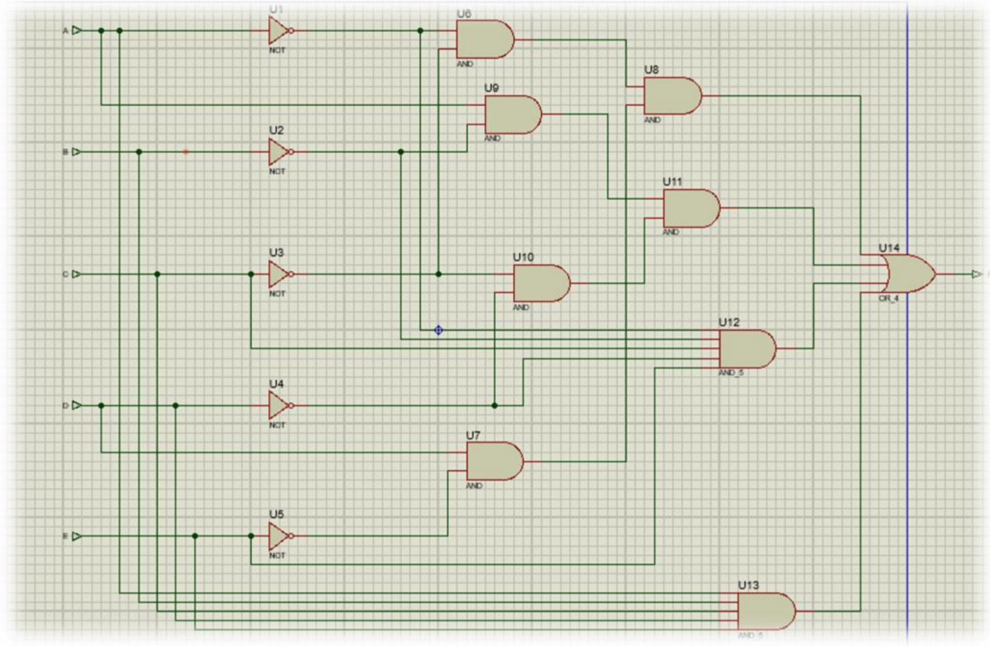


Figure 3: Proteus Diagram for LED (16)

Handmade Circuit Diagram for (LED 91)

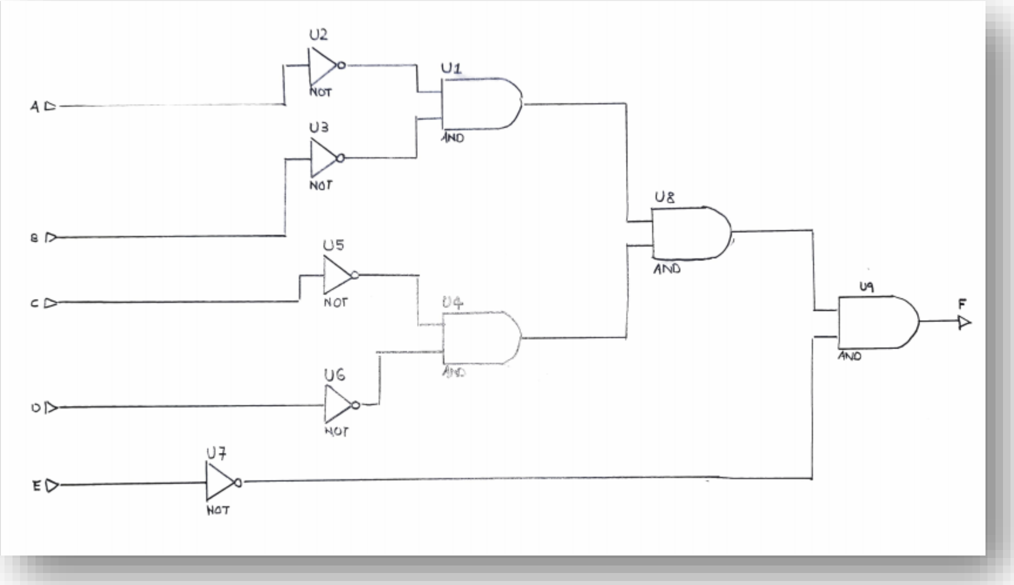


Figure 4: Handmade Circuit Diagram for LED 91

Proteus Diagram for (LED 91)

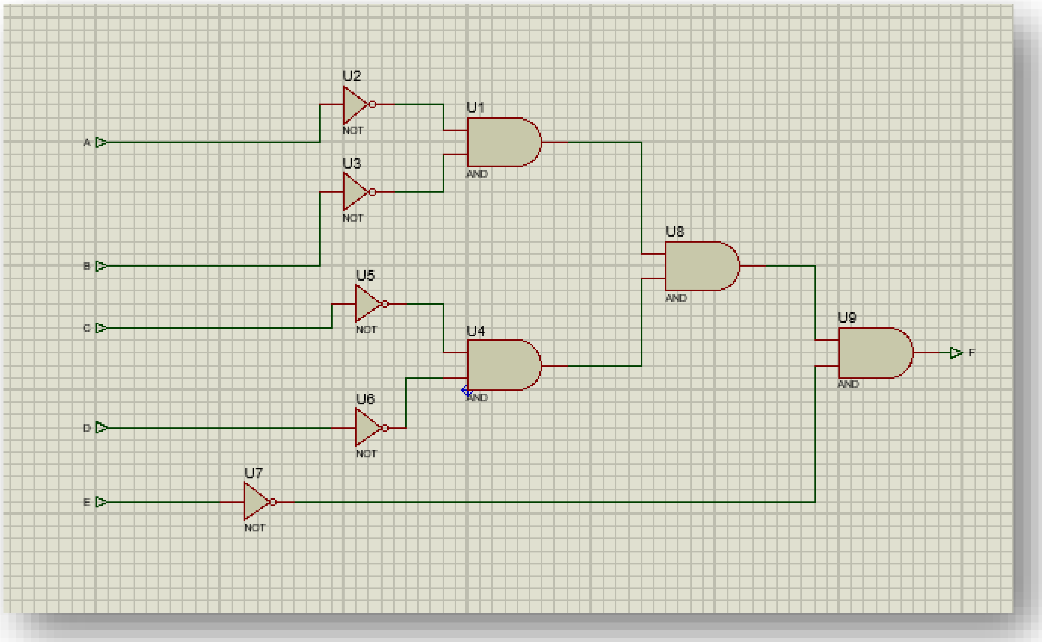


Figure 5: Proteus Diagram for LED 91

Handmade Circuit Diagram for (LED 95)

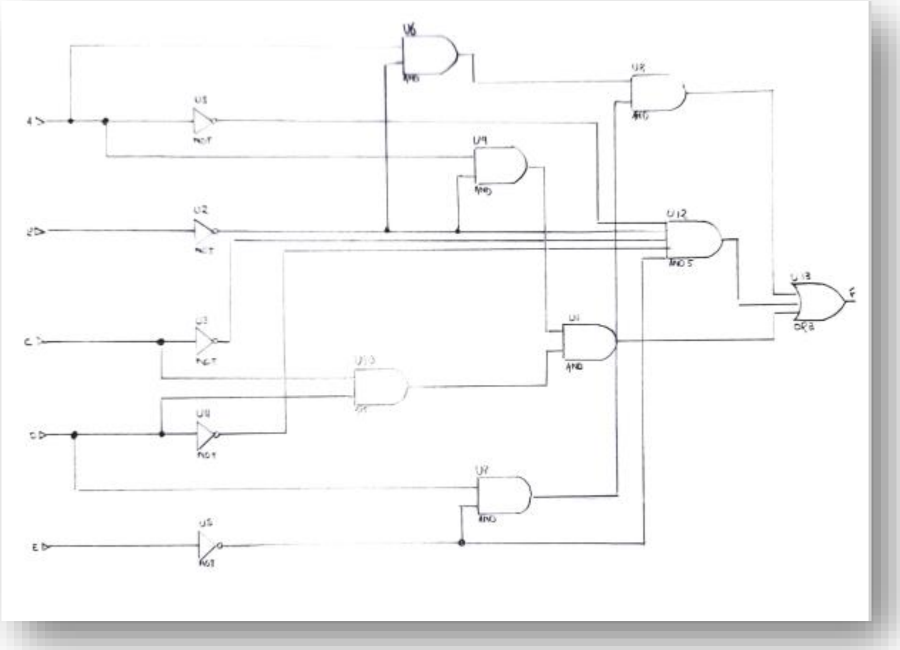


Figure 6: Handmade Circuit Diagram for LED 95

Proteus Diagram for (LED 95)

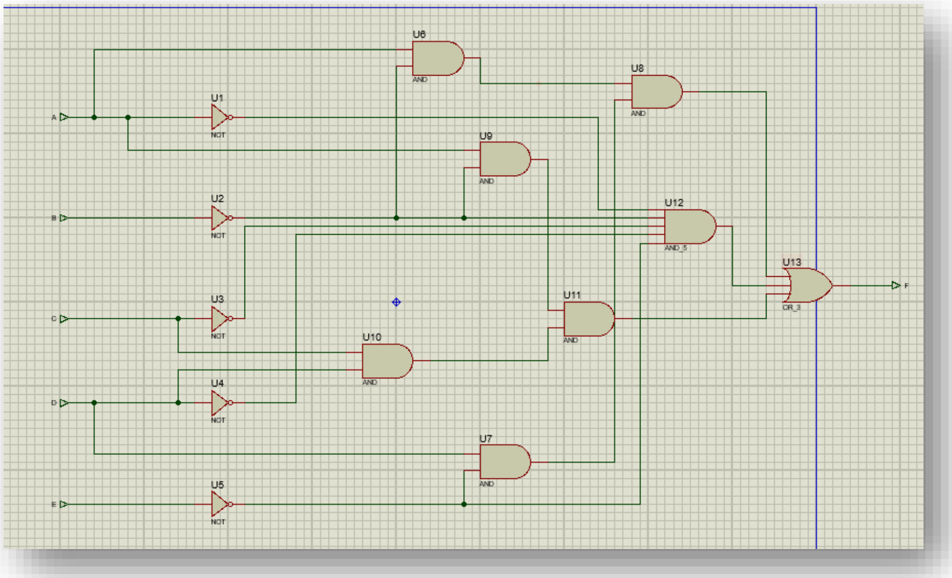
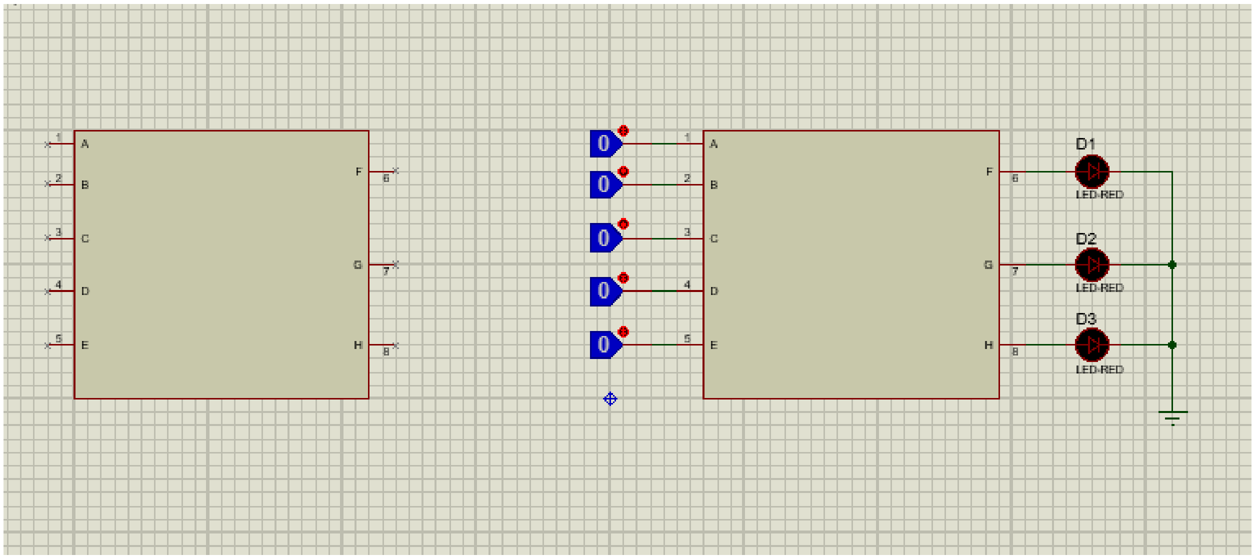
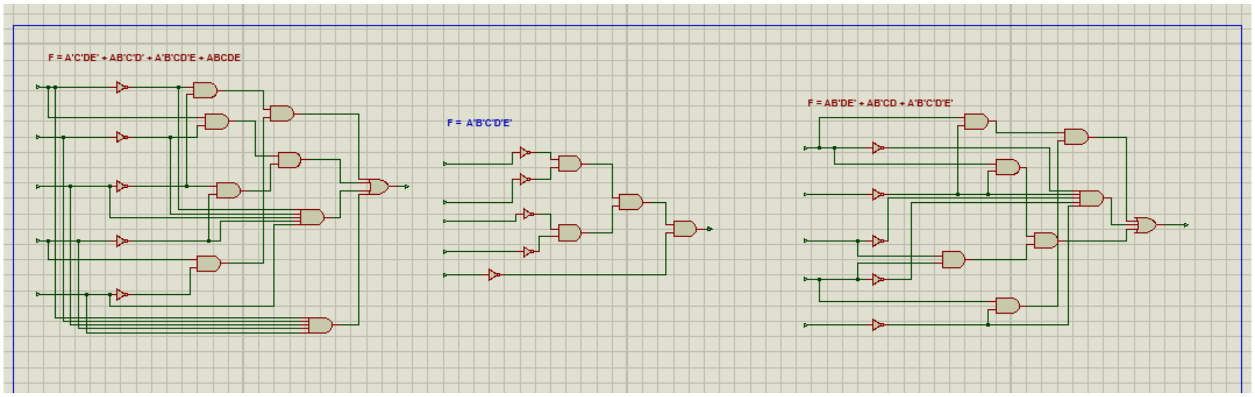


Figure 7: Proteus Diagram for LED 95

Parent Sheet for Part I



Child Sheet for Part I:


F = A'C'DE' + AB'C'D' + A'B'CD'E + ABCDE
F = A'B'C'D'E'
F = AB'DE' + AB'CD + A'B'C'D'E'

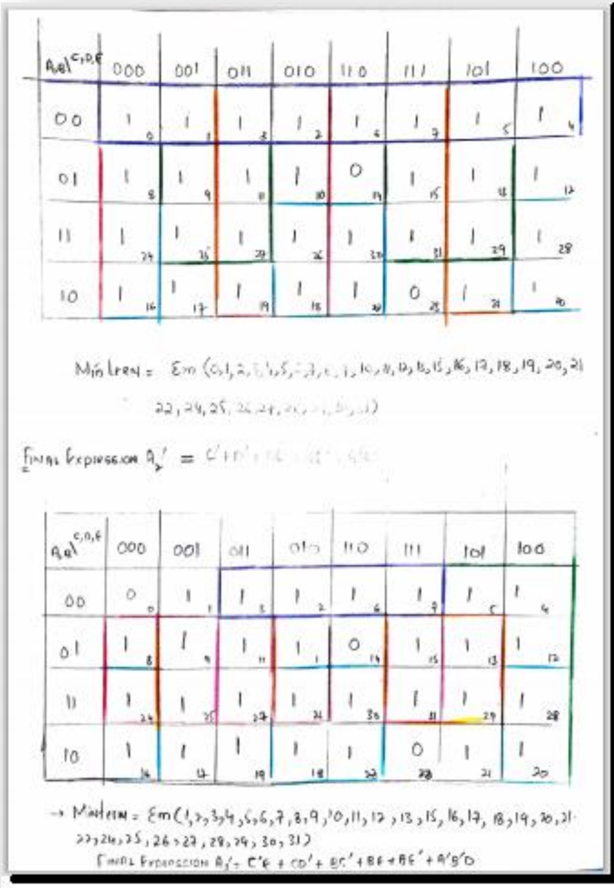
Part II

Assigned Number: **23** **17** **14** **0** **20**

Truth Tables

	A0	A1	A2	A3	A4	A(A+1)	B(B+1)	C(C+1)'	D(D+1)	E(E+1)
0	0	0	0	0	0	1	0	1	0	0
1	0	0	0	0	1	1	0	1	1	1
2	0	0	0	1	0	1	0	1	1	1
3	0	0	0	1	1	1	0	1	1	1
4	0	0	1	0	0	1	0	1	1	1
5	0	0	1	0	1	1	0	1	1	1
6	0	0	1	1	0	1	0	1	1	1
7	0	0	1	1	1	1	0	1	1	1
8	0	1	0	0	0	1	0	1	1	1
9	0	1	0	0	1	1	0	1	1	1
10	0	1	0	1	0	1	0	1	1	1
11	0	1	0	1	1	1	0	1	1	1
12	0	1	1	0	0	1	0	1	1	1
13	0	1	1	0	1	1	0	1	1	1
14	0	1	1	1	0	0	0	0	0	0
15	0	1	1	1	1	1	0	1	1	1
16	1	0	0	0	0	1	0	1	1	1
17	1	0	0	0	1	0	1	1	1	0
18	1	0	0	1	0	1	0	1	1	1
19	1	0	0	1	1	1	0	1	1	1
20	1	0	1	0	0	1	0	1	1	1
21	1	0	1	0	1	1	0	1	1	1
22	1	0	1	1	0	1	0	1	1	1
23	1	0	1	1	1	1	0	0	0	1
24	1	1	0	0	0	1	0	1	1	1
25	1	1	0	0	1	1	0	1	1	1
26	1	1	0	1	0	1	0	1	1	1
27	1	1	0	1	1	1	0	1	1	1
28	1	1	1	0	0	1	0	1	1	1
29	1	1	1	0	1	1	0	1	1	1
30	1	1	1	1	0	1	0	1	1	1
31	1	1	1	1	1	1	0	1	1	1

Handwritten K-Map (Part 2)



A,B\C,D,E	000	001	011	010	110	111	101	100
00	1_{0}	1_{1}	1_{3}	1_{2}	1_{6}	1_{7}	1_{5}	1_{4}
01	1_{8}	1_{9}	1_{11}	1_{10}	0_{14}	1_{15}	1_{13}	1_{12}
11	1_{24}	1_{25}	1_{27}	1_{26}	1_{30}	1_{31}	1_{29}	1_{28}
10	1_{16}	1_{17}	1_{19}	1_{18}	1_{22}	0_{23}	1_{21}	1_{20}

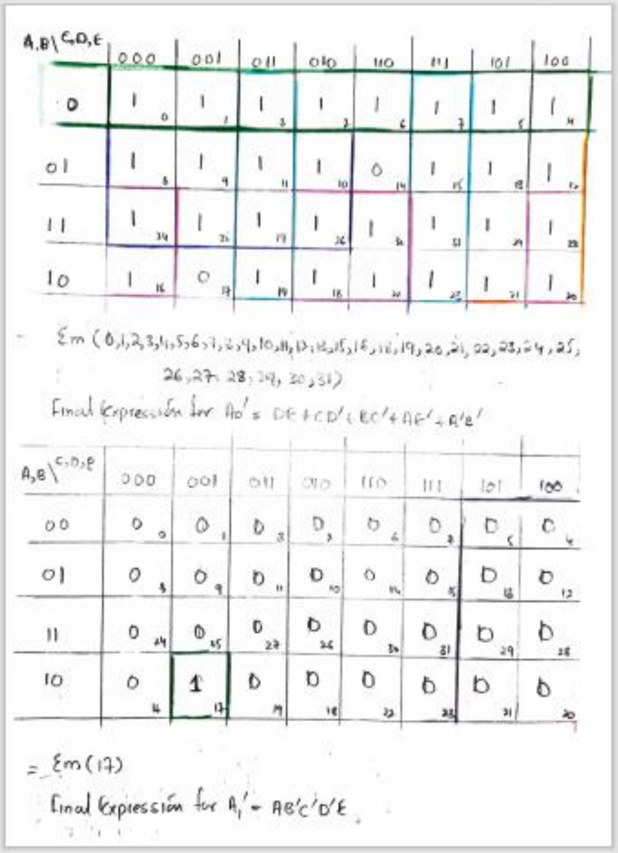
Mín term = Σm (0,1,2,3,4,5,6,7,8,9,10,11,12,13,15,16,17,18,19,20,21
22,24,25,26,27,28,29,30,31)

Final Expression $A_2' =$ C'+D' [illegible]

A,B\C,D,E	000	001	011	010	110	111	101	100
00	0_{0}	1_{1}	1_{3}	1_{2}	1_{6}	1_{7}	1_{5}	1_{4}
01	1_{8}	1_{9}	1_{11}	1_{10}	0_{14}	1_{15}	1_{13}	1_{12}
11	1_{24}	1_{25}	1_{27}	1_{26}	1_{30}	1_{31}	1_{29}	1_{28}
10	1_{16}	1_{17}	1_{19}	1_{18}	1_{22}	0_{23}	1_{21}	1_{20}

→ Minterm = Σm(1,2,3,4,5,6,7,8,9,10,11,12,13,15,16,17,18,19,20,21,
22,24,25,26,27,28,29,30,31)

Final Expression $A_3' = C'E + CD' + BC' + BE + BE' + A'B'D$



A,B\C,D,E	000	001	011	010	110	111	101	100
0	1_{0}	1_{1}	1_{3}	1_{2}	1_{6}	1_{7}	1_{5}	1_{4}
01	1_{8}	1_{9}	1_{11}	1_{10}	0_{14}	1_{15}	1_{13}	1_{12}
11	1_{24}	1_{25}	1_{27}	1_{26}	1_{30}	1_{31}	1_{29}	1_{28}
10	1_{16}	0_{17}	1_{19}	1_{18}	1_{22}	1_{23}	1_{21}	1_{20}

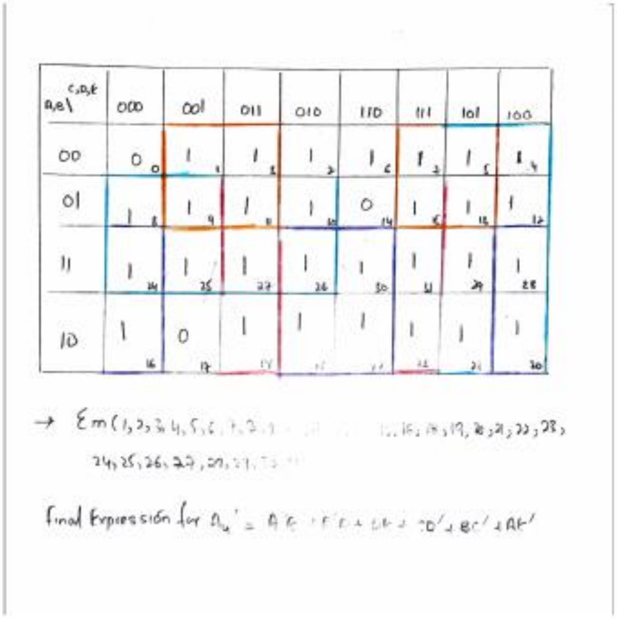
- Σm (0,1,2,3,4,5,6,7,8,9,10,11,12,13,15,16,18,19,20,21,22,23,24,25,
26,27,28,29,30,31)

Final Expresión for $A_0' = DE + CD' + BC' + AE' + A'E'$

A,B\C,D,E	000	001	011	010	110	111	101	100
00	0_{0}	0_{1}	0_{3}	0_{2}	0_{6}	0_{7}	0_{5}	0_{4}
01	0_{8}	0_{9}	0_{11}	0_{10}	0_{14}	0_{15}	0_{13}	0_{12}
11	0_{24}	0_{25}	0_{27}	0_{26}	0_{30}	0_{31}	0_{29}	0_{28}
10	0_{16}	1_{17}	0_{19}	0_{18}	0_{22}	0_{23}	0_{21}	0_{20}

$= \Sigma m(17)$

Final Expresión for $A_1' = AB'C'D'E$



A,B\C,D,E	000	001	011	010	110	111	101	100
00	0_{0}	1_{1}	1_{3}	1_{2}	1_{6}	1_{7}	1_{5}	1_{4}
01	1_{8}	1_{9}	1_{11}	1_{10}	0_{14}	1_{15}	1_{13}	1_{12}
11	1_{24}	1_{25}	1_{27}	1_{26}	1_{30}	1_{31}	1_{29}	1_{28}
10	1_{16}	0_{17}	1_{19}	1_{18}	1_{22}	1_{23}	1_{21}	1_{30}

→ Σm(1,2,3,4,5,6,7,8,9, [illegible] 15,16,18,19,20,21,22,23,
24,25,26,27,28,29,30,31)

Final Expresión for $A_4' = A'E + E'C + BE + CD' + BC' + AE'$

Expression For A(A+1): DE+CD’+BC’+AE’+A’B’

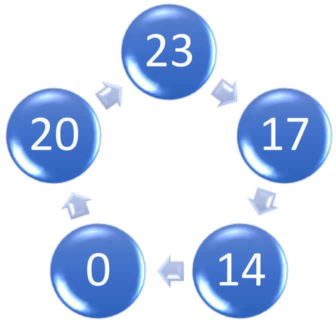
Expression For B(B+1): AB’C’D’E

Expression For C(C+1): C’+D’+BE+AE’+A’B’

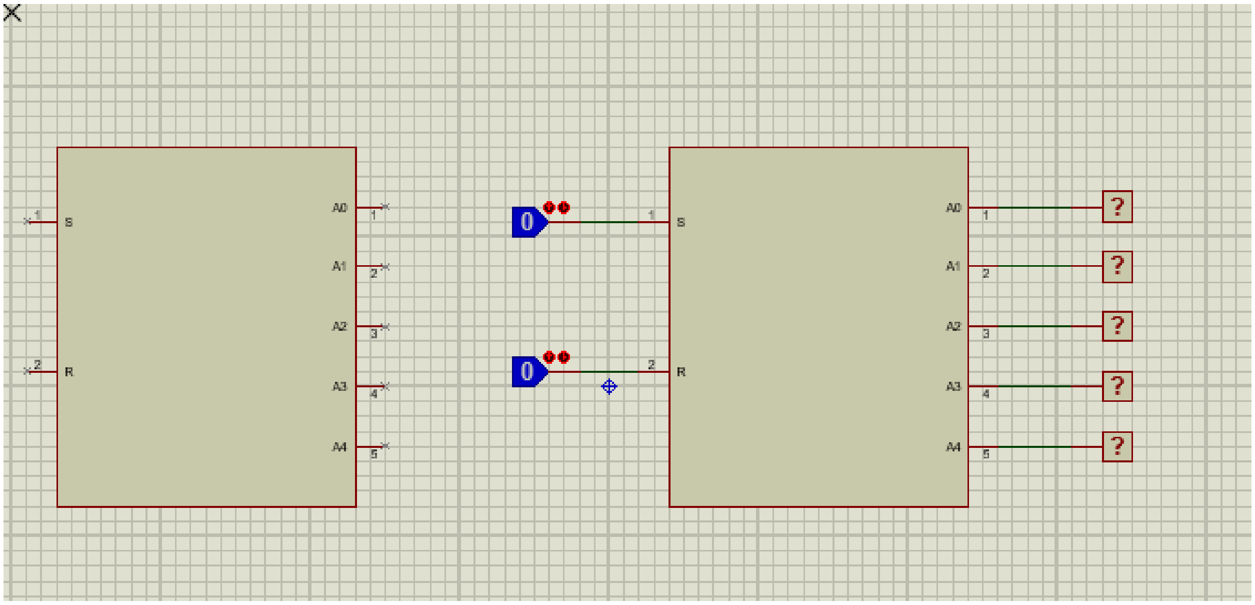
Expression For D(D+1): C’E+CD’+BC’+BE+AE’+A’B’D

Expression For E(E+1): A’E+B’D+DE+CD’+BC’+AE’

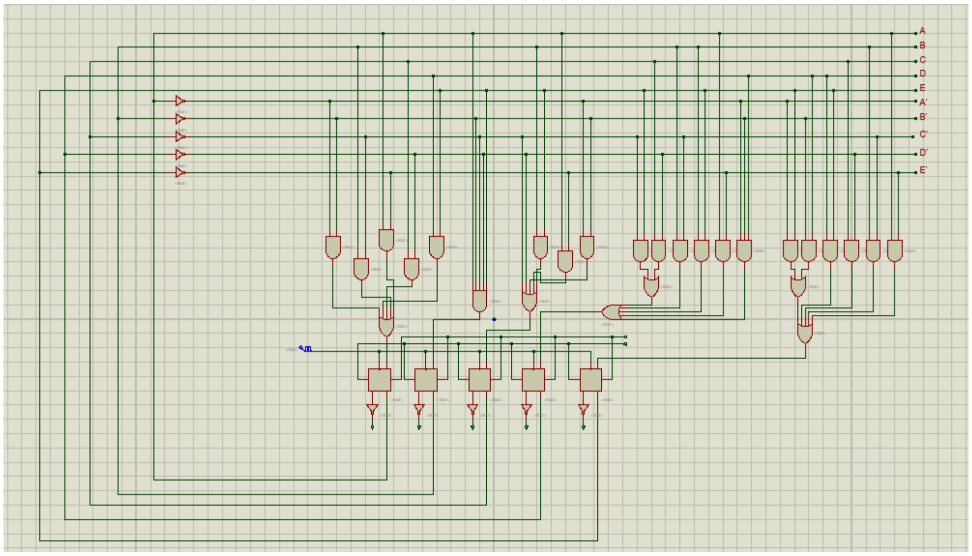
State Diagram Part II



Parent Sheet For Part II



Child Sheet For Part II



Implementation Details (Part 2)

Written:

Using our given sequence of numbers (23-17-14-0-20), we first designed a truth table which was the primary objective followed by a state diagram. According to the mentioned details in project, don’t care terms were set to A0 (23 In our case). 5 different K-maps were constructed for each next state using min terms and a Boolean expression was derived.

Proteus Implementation:

Using 2D Graphics Box Mode, we made a wireless IC same as we did it in part 1. It had 2 input pins for set and reset respectively and 5 output pins. Select 'make device' option for component which officially makes our block diagram a circuit. We edited the properties of our device and enable hierarchy module which gave us access to child sheets. In child sheet, we designed the circuit according to the constructed Boolean expressions using Logic Gate NOT, different AND gates, different OR gates, D clock and D flip flops. All set and reset pins of D flip flop were connected to two inputs, respectively. Validity of circuit were checked for the given sequence.