DLD Final PROJECT

Report

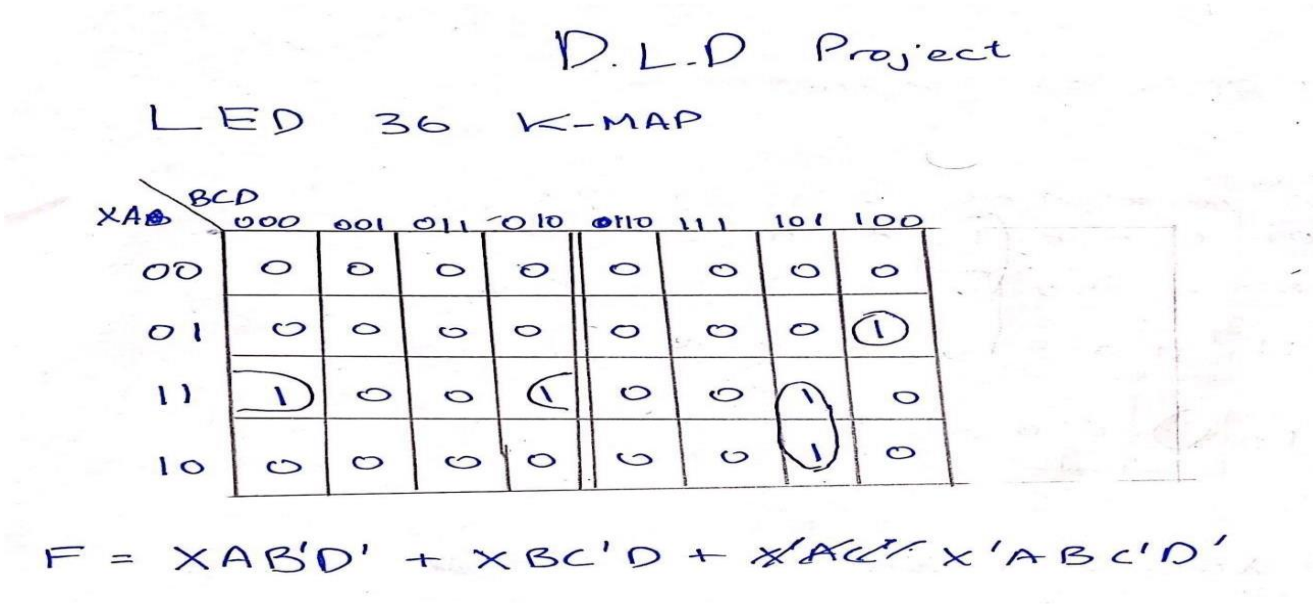
MUHAMMAD AHMED BAIG (20I-1884) || USMAN AFZAL (20I-0786)
BS(CS)-B

Part 1:

Truth Table

	INPUT					OUPUT		
#	**X**	**A**	**B**	**C**	**D**	**36**	**51**	**81**
0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	0	0	0
2	0	0	0	1	0	0	1	1
3	0	0	0	1	1	0	0	1
4	0	0	1	0	0	0	0	0
5	0	0	1	0	1	0	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	0	1	1
11	0	1	0	1	1	0	0	1
12	0	1	1	0	0	0	0	0
13	0	1	1	0	1	1	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	0	1	0
17	1	0	0	0	1	0	0	0

18	1	0	0	1	0	0	0	0
19	1	0	0	1	1	0	0	0
20	1	0	1	0	0	0	0	0
21	1	0	1	0	1	1	0	0
22	1	0	1	1	0	0	0	0
23	1	0	1	1	1	0	0	0
24	1	1	0	0	0	1	0	0
25	1	1	0	0	1	0	0	0
26	1	1	0	1	0	1	1	0
27	1	1	0	1	1	0	0	0
28	1	1	1	0	0	0	1	1
29	1	1	1	0	1	1	0	0
30	1	1	1	1	0	0	0	0
31	1	1	1	1	1	0	0	0

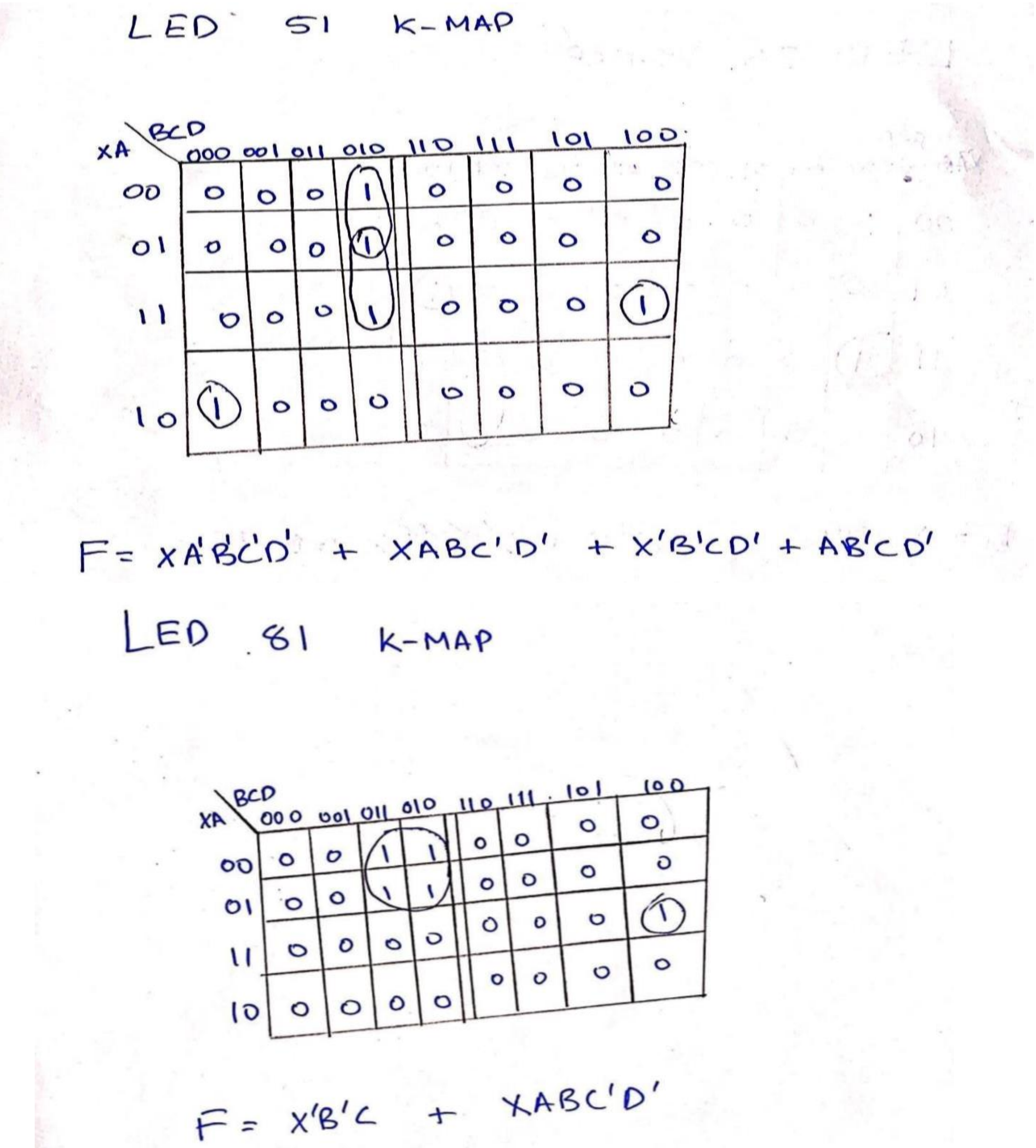


D.L.D Project

LED 36 K-MAP

XA \ BCD	000	001	011	010	110	111	101	100
00	0	0	0	0	0	0	0	0
01	0	0	0	0	0	0	0	1
11	1	0	0	1	0	0	1	0
10	0	0	0	0	0	0	1	0

F = XAB'D' + XBC'D + ~~X'AC'~~ X'ABC'D'



LED 51 K-MAP

XA \ BCD	000	001	011	010	110	111	101	100
00	0	0	0	1	0	0	0	0
01	0	0	0	1	0	0	0	0
11	0	0	0	1	0	0	0	1
10	1	0	0	0	0	0	0	0

F= XA'B'C'D' + XABC'D' + X'B'CD' + AB'CD'

LED 81 K-MAP

XA \ BCD	000	001	011	010	110	111	101	100
00	0	0	1	1	0	0	0	0
01	0	0	1	1	0	0	0	1
11	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0

F= X'B'C + XABC'D'

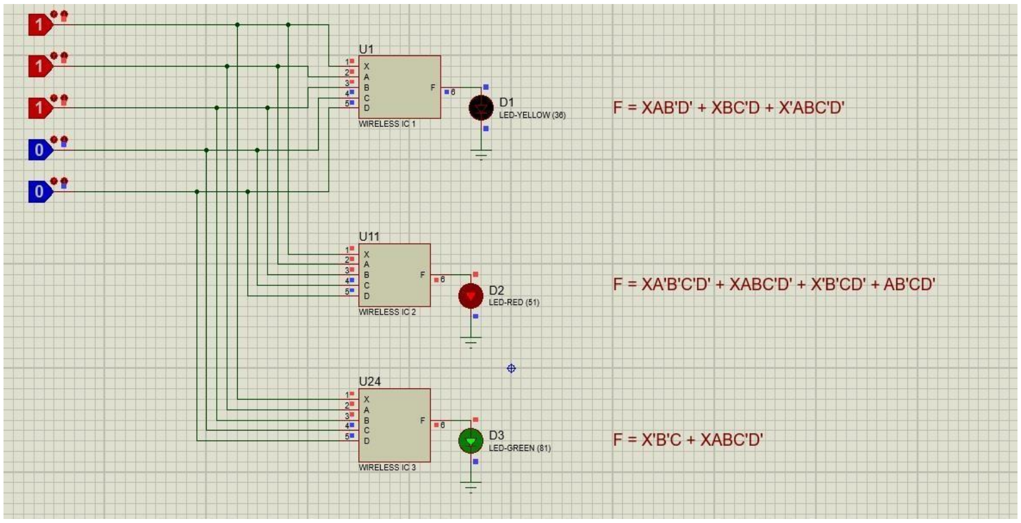
Implementation details for part1:

The output which we were provided were (51-81-36) by using this a truth table of 5 variables was made (X, A, B, C, D). Firstly, the truth table was filled using 32 patterns. K-Map was made for each LED No. output i.e. (51-81-36). After making the K-MAP it was filled using the truth table. Then group of multiplies of 2 were made to make the Boolean Expression for each of the LED. Then we designed the circuit in Proteus. Firstly, we made a wireless IC (each for a specific LED). We made it by drawing a component box through 2D Graphic Box Mode. Then we added default pins from the Device Pin Mode for 5 inputs and 1 output (for each LED). After we clicked the Make Device option. Now the device was made and it was shown in the Component mode. I repeated this process thrice to make IC for each Boolean expression and LED. Now I attached the IC's input with logic state to get the input and attached the output of the IC with the specific LED (i.e. for the Boolean Expression). Now by clicking right click on every IC I choose the option Child Sheet and designed the circuit for each IC in it. Each Boolean Expression were implemented using NOTGATE, AND-GATE and OR-GATE. I attached 5 inputs and 1 output in each child sheet and named them same as the input and output of the IC. In the end I attached the LED with ground and tested my circuit by using the inputs and outputs from truth table.

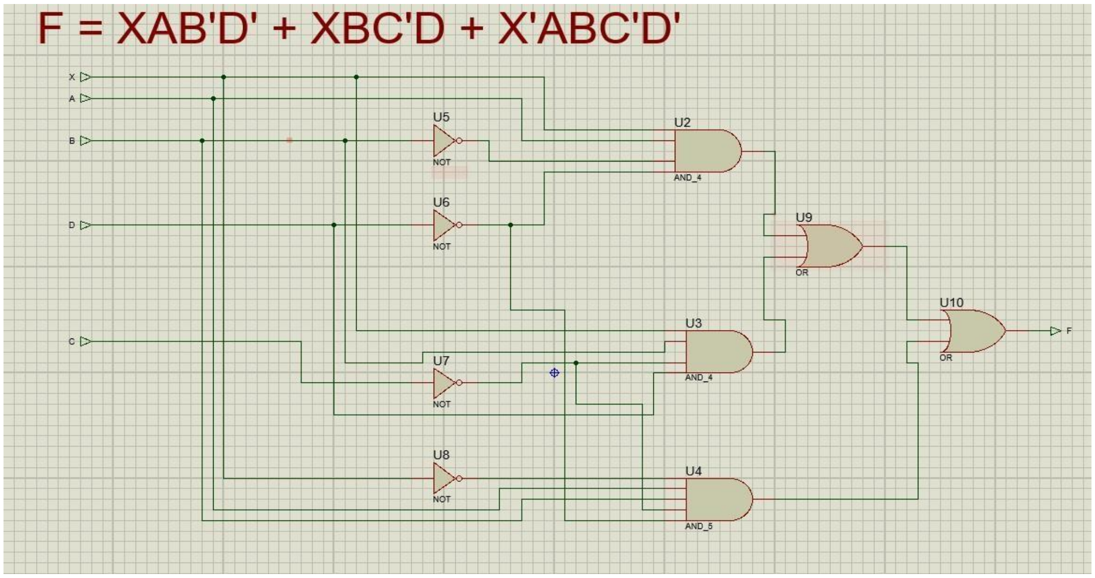
Components used in part1:

And gates, OR gates, input terminals, output terminals, LED's, logic toggle, not gate.

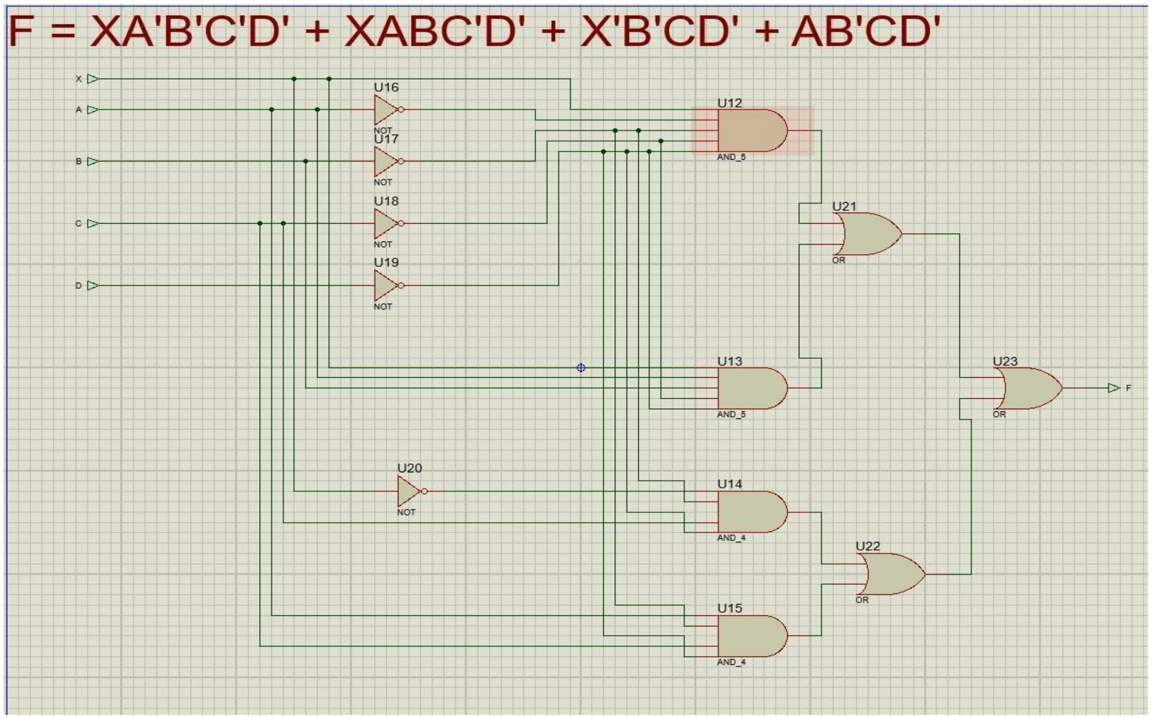
Parent Sheet



Wireless IC 1 (FOR LED 36)

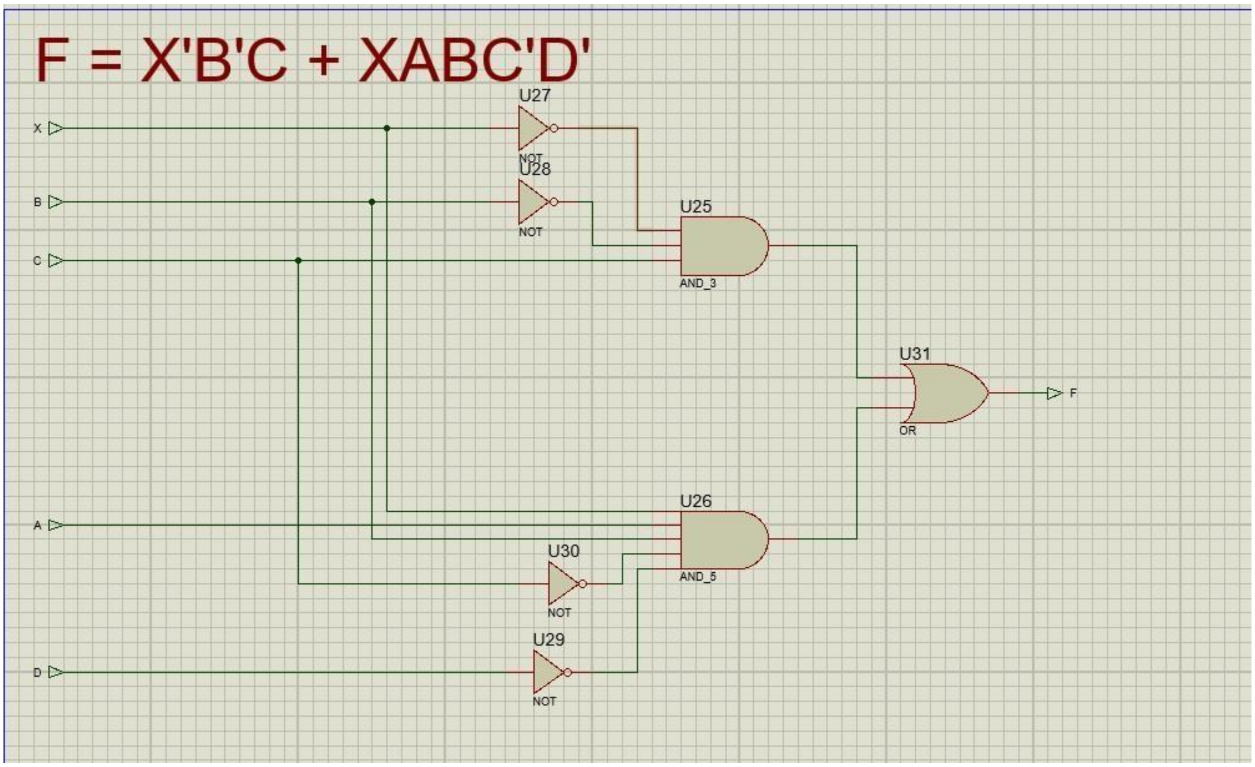


Wireless IC 2 (FOR LE



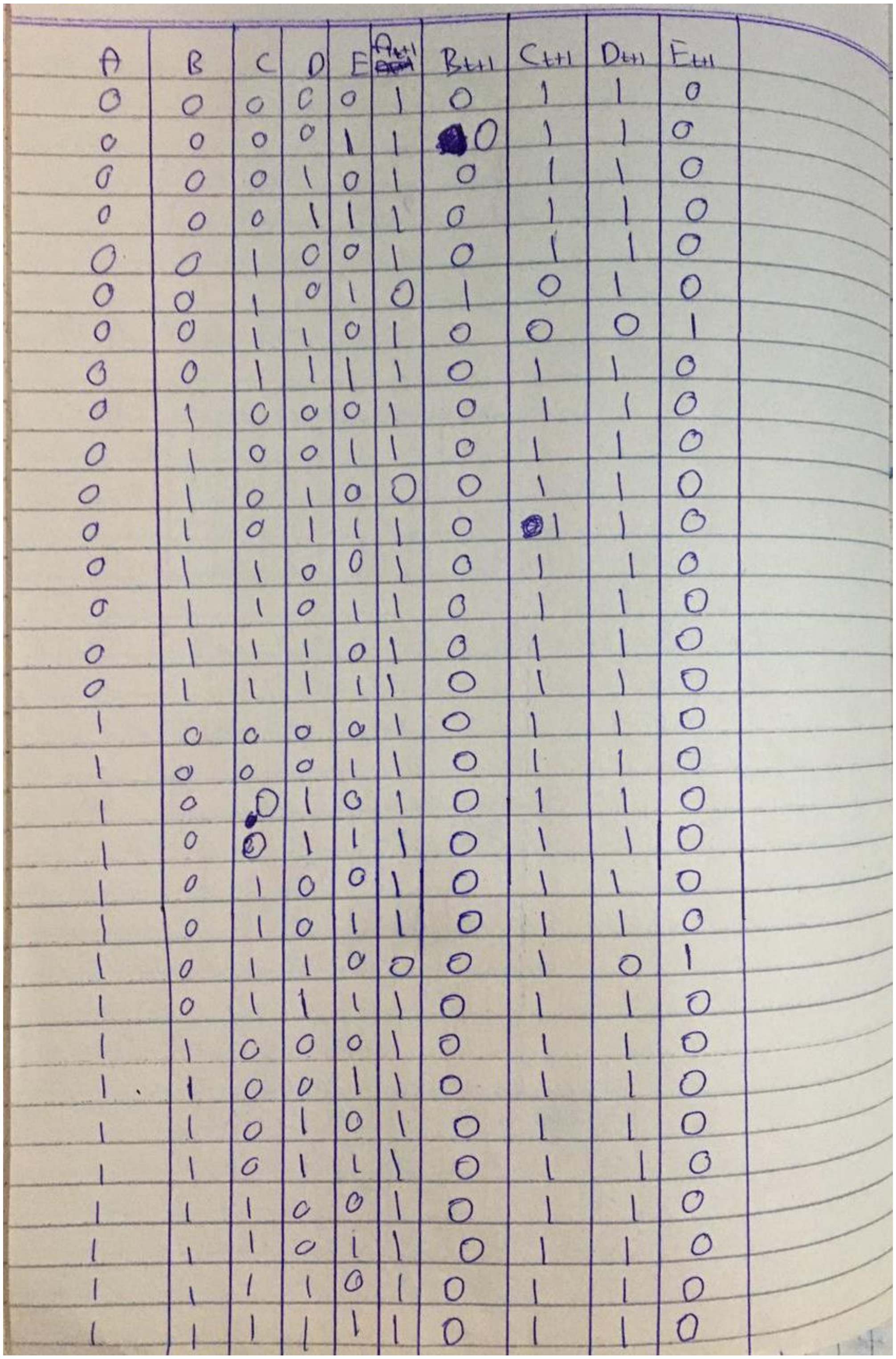
D 51)

Wireless IC 3 (FOR LED 81)



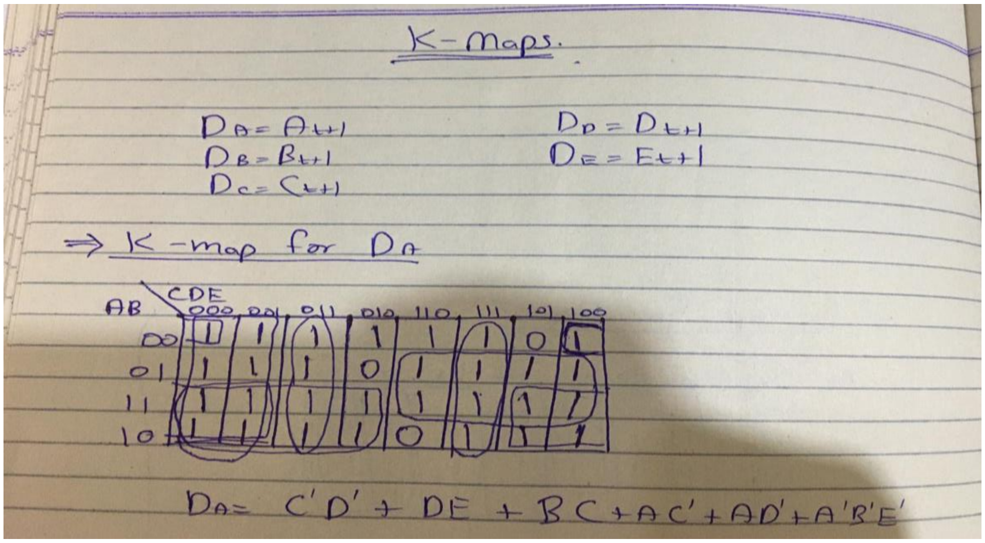
Part 2:

Truth table



A	B	C	D	E	A_{t+1}	B_{t+1}	C_{t+1}	D_{t+1}	E_{t+1}
0	0	0	0	0	1	0	1	1	0
0	0	0	0	1	1	0	1	1	0
0	0	0	1	0	1	0	1	1	0
0	0	0	1	1	1	0	1	1	0
0	0	1	0	0	1	0	1	1	0
0	0	1	0	1	0	1	0	1	0
0	0	1	1	0	1	0	0	0	1
0	0	1	1	1	1	0	1	1	0
0	1	0	0	0	1	0	1	1	0
0	1	0	0	1	1	0	1	1	0
0	1	0	1	0	0	0	1	1	0
0	1	0	1	1	1	0	1	1	0
0	1	1	0	0	1	0	1	1	0
0	1	1	0	1	1	0	1	1	0
0	1	1	1	0	1	0	1	1	0
0	1	1	1	1	1	0	1	1	0
1	0	0	0	0	1	0	1	1	0
1	0	0	0	1	1	0	1	1	0
1	0	0	1	0	1	0	1	1	0
1	0	0	1	1	1	0	1	1	0
1	0	1	0	0	1	0	1	1	0
1	0	1	0	1	1	0	1	1	0
1	0	1	1	0	0	0	1	0	1
1	0	1	1	1	1	0	1	1	0
1	1	0	0	0	1	0	1	1	0
1	1	0	0	1	1	0	1	1	0
1	1	0	1	0	1	0	1	1	0
1	1	0	1	1	1	0	1	1	0
1	1	1	0	0	1	0	1	1	0
1	1	1	0	1	1	0	1	1	0
1	1	1	1	0	1	0	1	1	0
1	1	1	1	1	1	0	1	1	0

K-maps for part2:



K-Maps.

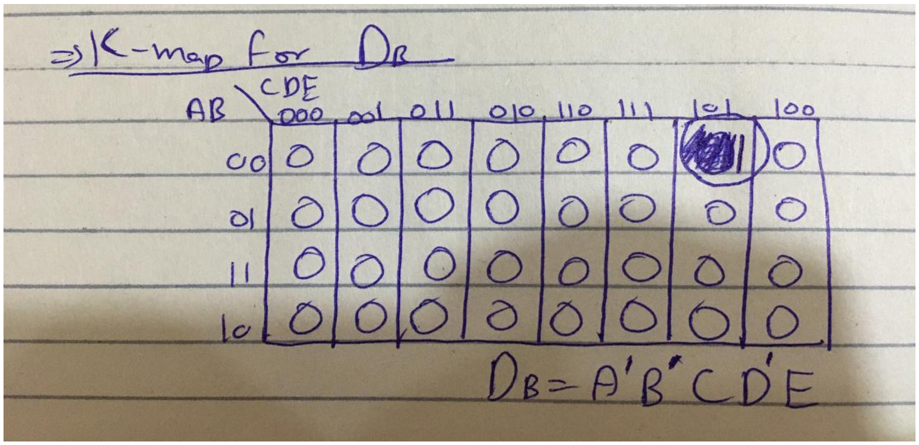
$D_A = A_{t+1}$
$D_B = B_{t+1}$
$D_C = C_{t+1}$

$D_D = D_{t+1}$
$D_E = E_{t+1}$

⇒ K-map for D_A

AB \ CDE	000	001	011	010	110	111	101	100
00	1	1	1	1	1	1	0	1
01	1	1	1	0	1	1	1	1
11	1	1	1	1	1	1	1	1
10	1	1	1	1	0	1	1	1

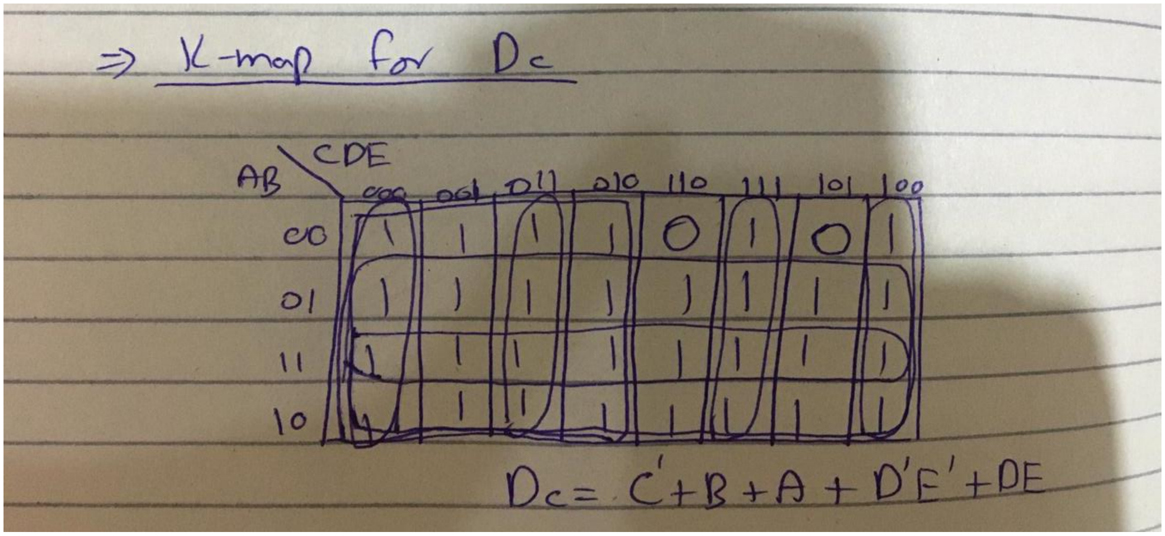
$D_A = C'D' + DE + BC + AC' + AD' + A'B'E'$



⇒ K-map for D_B

AB \ CDE	000	001	011	010	110	111	101	100
00	0	0	0	0	0	0	1	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

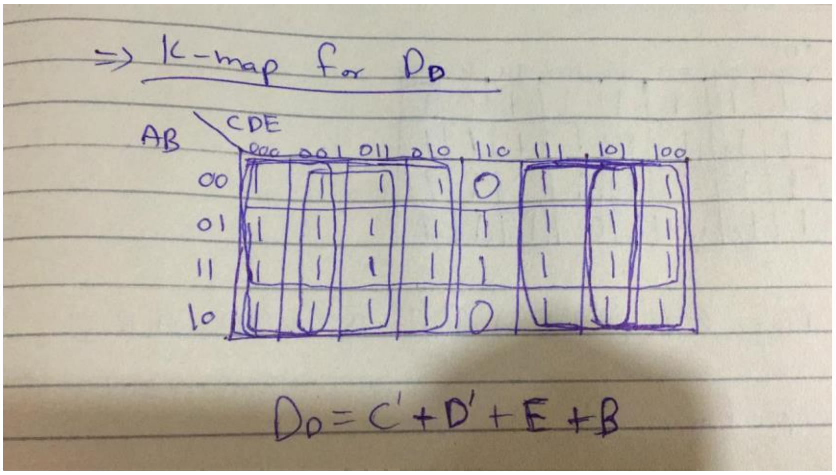
$D_B = A'B'CD'E$



⇒ K-map for D_C

AB \ CDE	000	001	011	010	110	111	101	100
00	1	1	1	1	0	1	0	1
01	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1
10	1	1	1	1	1	1	1	1

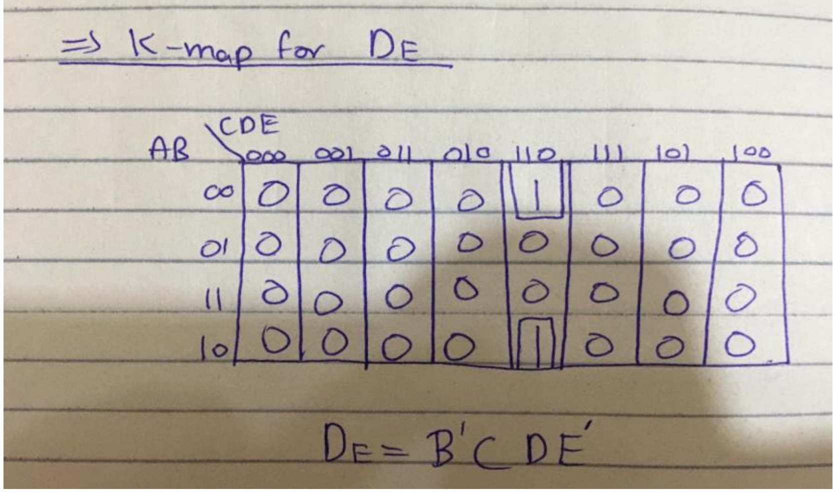
$D_C = C' + B + A + D'E' + DE$



⇒ K-map for D_D

AB \ CDE	000	001	011	010	110	111	101	100
00	1	1	1	1	0	1	1	1
01	1	1	1	1	1	1	1	1
11	1	1	1	1	1	1	1	1
10	1	1	1	1	0	1	1	1

$D_D = C' + D' + E + B$



⇒ K-map for D_E

AB \ CDE	000	001	011	010	110	111	101	100
00	0	0	0	0	1	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	1	0	0	0

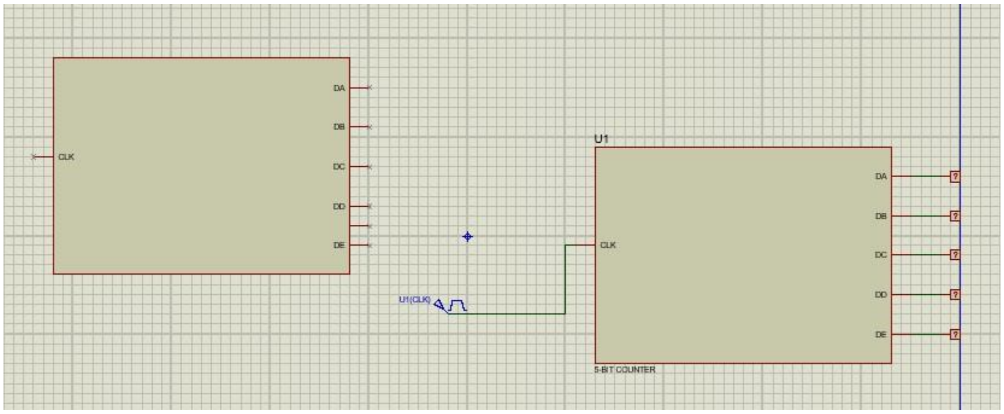
$D_E = B'CDE'$

Implementation details for part2:

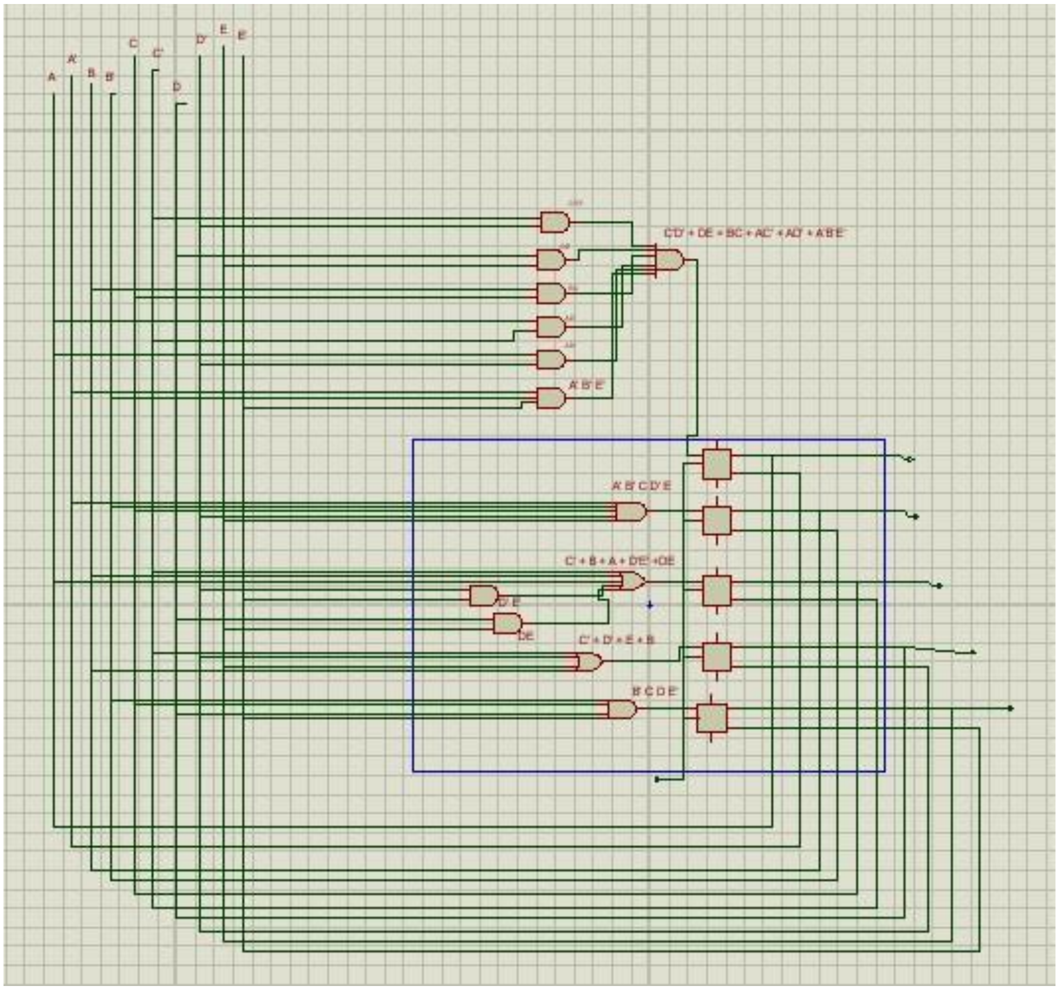
The numbers we were assigned for part 2 were (22, 5, 10, 6, 17), so we filled the 5-bit truth table according to the values assigned. Then we make separate 5-Bit k-maps for DA, DB, DC, DD, DE and derived the Boolean expression for each of them. Then we made the circuit in the proteus. For this, we first made an IC with one input and five outputs in the parent sheet. A DC pulse was given as the input for the clock and toggle probes were used to display the output. Then we made the child sheet in which we used 5-D flip-flops and wired them according to the Boolean expressions formed. The ‘Q’ of the D flip-flops was used as output and the clock was given the input.

Components used for part 2:
D-flip-flops, Dc pulse, toggle probe, And gates, Or gates, input terminals, output terminals, default pins.

Parent sheet:



Child sheet:


A
A'
B
B'
C
C'
D
D'
E
E'
C'D' + DE + BC + AC' + AD' + A'B'E'
A'B'E'
A'B'C D'E
C' + B + A + DE' + D'E
D'E
DE'
C' + D' + E + B
B'C D E'