Part 1A Paper 3 : Electrical and Information Engineering

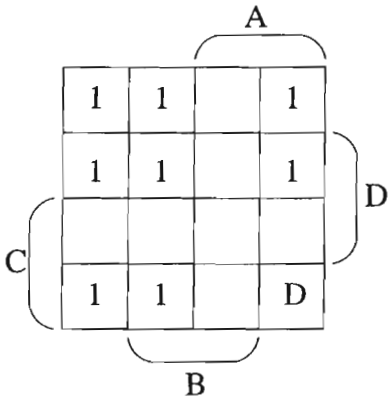
DIGITAL CIRCUITS AND INFORMATION PROCESSING

SOLUTIONS TO EXAMPLES PAPER 2

1. From the map:

$$F = \overline{A}.\overline{C} + \overline{B}.\overline{C} + \overline{A}.\overline{D}$$
$$\overline{F} = A.B + C.D$$

The don't-care-state is taken as a 0 in the expression for F and as a 1 in the expression for $\overline{F}$.



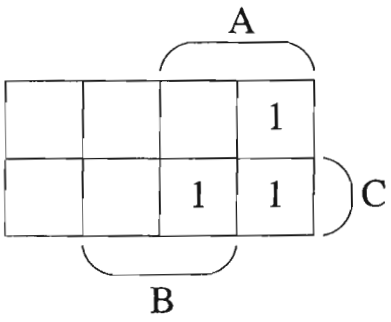
2. From the map:

(a) for NAND gates

$$F = A.\overline{B} + A.C$$
$$= \overline{\overline{(A.\overline{B})}.\overline{(A.C)}}$$

(b) for NOR gates

$$\overline{F} = \overline{A} + B.\overline{C}$$
$$\Rightarrow F = \overline{\overline{A} + \overline{(\overline{B} + C)}}$$

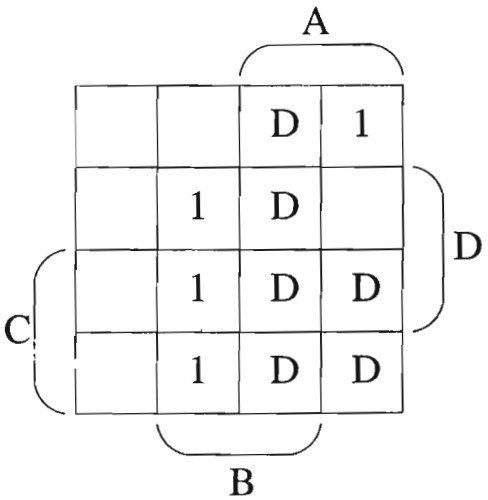


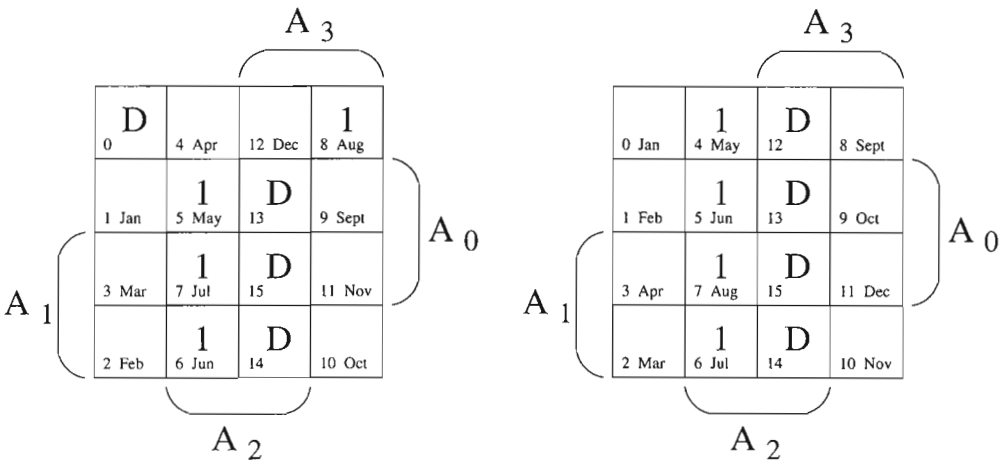
3. Sum of products:

$$G = A.\overline{D} + B.C + B.D$$

Product of sums:

$$\overline{G} = \overline{A}.\overline{B} + \overline{A}.\overline{C}.\overline{D} + A.D$$
$$= \overline{(A+B)} + \overline{(A+C+D)} + \overline{(\overline{A}+\overline{D})}$$
$$\Rightarrow G = \overline{\overline{(A+B)} + \overline{(A+C+D)} + \overline{(\overline{A}+\overline{D})}}$$
$$= (A+B).(A+C+D).(\overline{A}+\overline{D})$$





(a) (b)

Figure 1:

4. Using the first code we obtain the Karnaugh map in Figure 1(a), leading to $F = A_1.A_2 + A_0.A_2 + \overline{A_0}.\overline{A_1}.\overline{A_2}$.

Using the second code we obtain the Karnaugh map in Figure 1(b), leading to $F = A_2$.

5.

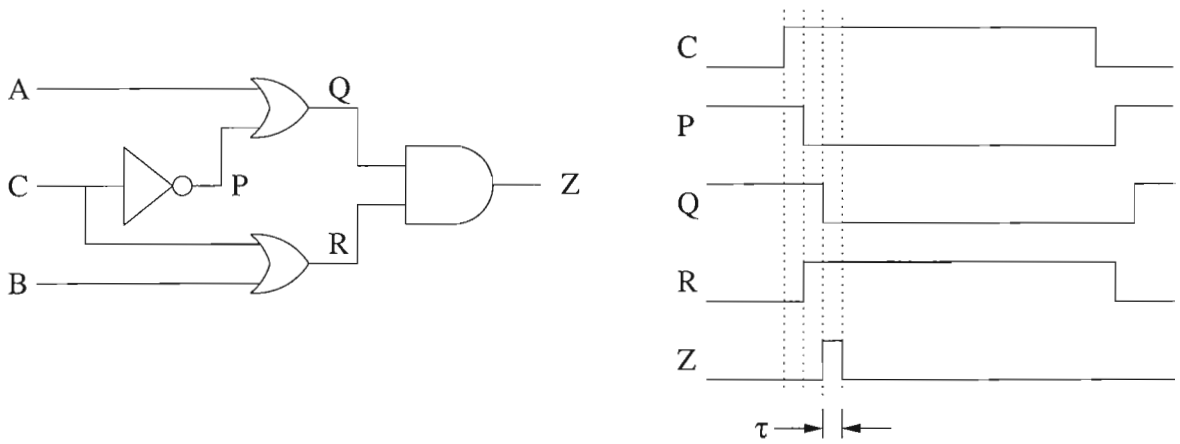


Figure 2:

(a) Figure 2 shows that we have a static 0-hazard.

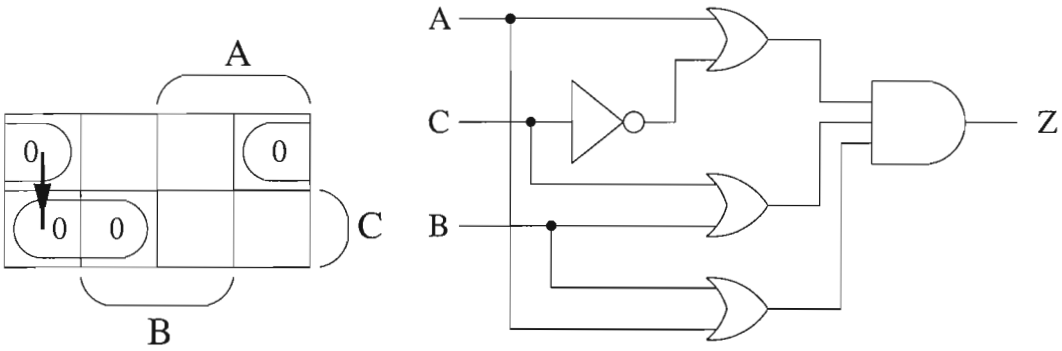
(b)

$$Z = (A + \overline{C}).(B + C)$$

$$= \overline{\overline{(A+\overline{C})} + \overline{(B+C)}}$$

$$= \overline{\overline{A}.C + \overline{B}.\overline{C}}$$

$$\Rightarrow \overline{Z} = \overline{A}.C + \overline{B}.\overline{C}$$



(a) $\overline{Z}$ (b) Solution: hazard-free circuit

Figure 3:

(c) The Karnaugh map of $\overline{Z}$ is shown in Figure 3(a). To fix the hazard we need an additional term of $\overline{A}.\overline{B}$ in this Karnaugh map.

$$\overline{Z} = \overline{A}.C + \overline{B}.\overline{C} + \overline{A}.\overline{B}$$

$$\Rightarrow Z = \overline{\overline{A}.C + \overline{B}.\overline{C} + \overline{A}.\overline{B}}$$

$$= \overline{\overline{(A+\overline{C})} + \overline{(B+C)} + \overline{(A+B)}}$$

$$= (A+\overline{C}).(B+C).(A+B)$$

Thus to remove the hazard we need to add another OR-gate as shown in Figure 3(b).

6.

(a)

$$38_{10} = 00100110_2$$

$$\Rightarrow -38_{10} = 11011010_2 \text{ (using: complement and add 1)}$$

$$100_{10} = 01100100_2$$

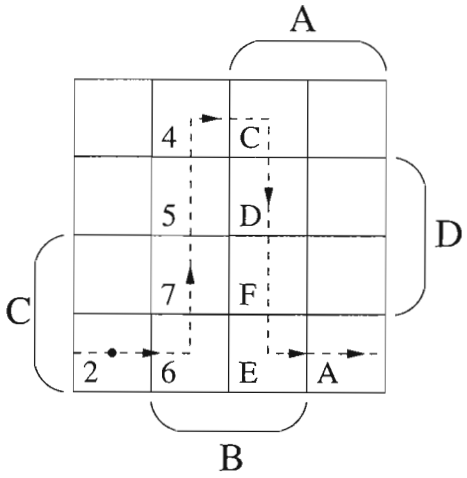
$$\Rightarrow 100_{10} - 38_{10} = (1)00111110_2$$

Discarding the top bit of the sum we have

$$100_{10} - 38_{10} = 00111110_2 = 62_{10} = 3E_{16}$$

(b)

Base 16	Base 2			
	A	B	C	D
2	0	0	1	0
6	0	1	1	0
7	0	1	1	1
5	0	1	0	1
4	0	1	0	0
C	1	1	0	0
D	1	1	0	1
F	1	1	1	1
E	1	1	1	0
A	1	0	1	0



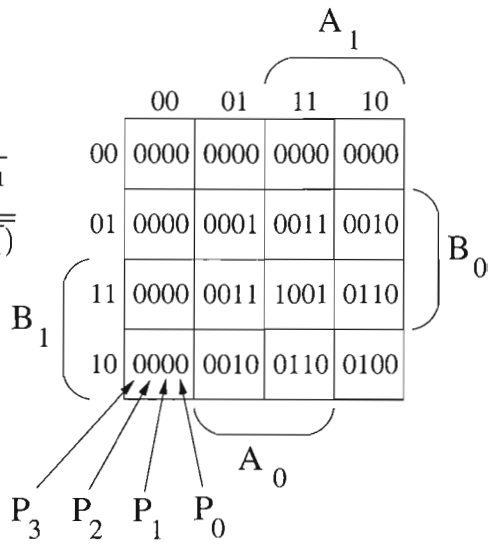
7.

$$P_0 = A_0.B_0$$
$$= \overline{\overline{A_0.B_0}}$$

$$P_1 = \overline{A_0}.A_1.B_0 + A_0.\overline{A_1}.B_1 + A_0.\overline{B_0}.B_1 + A_1.B_0.\overline{B_1}$$
$$= \overline{\overline{(\overline{A_0}.A_1.B_0)}.\overline{(A_0.\overline{A_1}.B_1)}.\overline{(A_0.\overline{B_0}.B_1)}.\overline{(A_1.B_0.\overline{B_1})}}$$

$$P_2 = \overline{A_0}.A_1.B_1 + A_1.\overline{B_0}.B_1$$
$$= \overline{\overline{(\overline{A_0}.A_1.B_1)}.\overline{(A_1.\overline{B_0}.B_1)}}$$

$$P_3 = A_0.A_1.B_0.B_1$$
$$= \overline{\overline{A_0.A_1.B_0.B_1}}$$



8. The timing diagram and output counter states are shown in Figure 4.

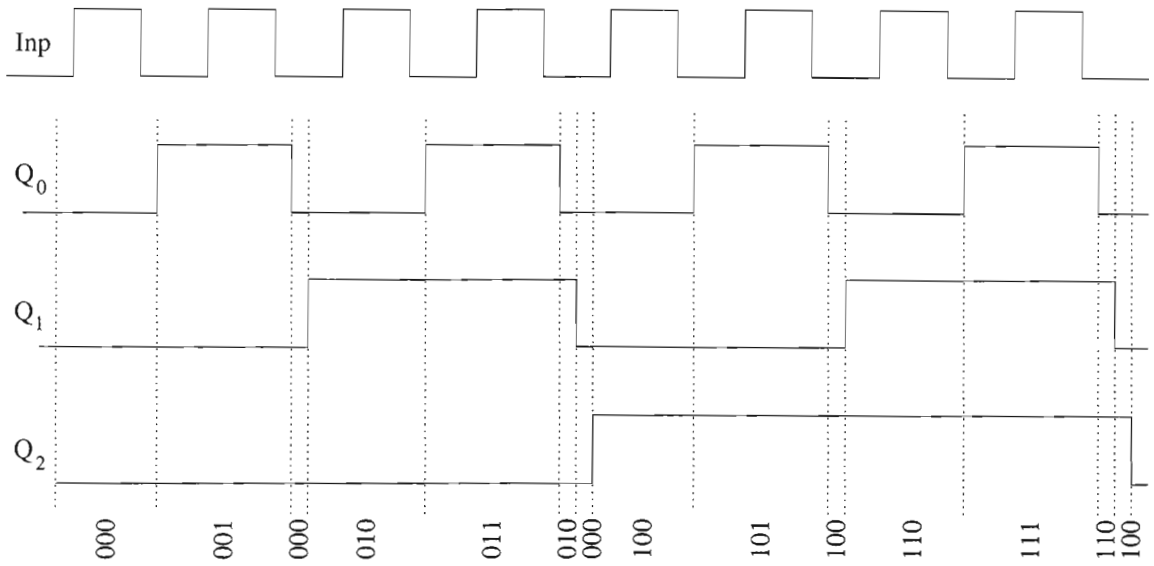


Figure 4:

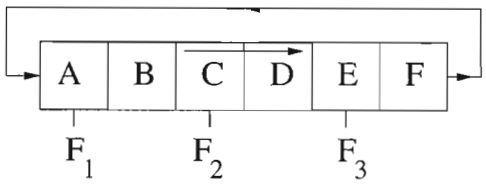
9. The input to stage A is formed from the exclusive-or of stages C and E. The sequence repeats after 31 steps.

A	B	C	D	E	Decimal
1	1	1	1	1	31
0	1	1	1	1	15
0	0	1	1	1	7
0	0	0	1	1	3
1	0	0	0	1	17
1	1	0	0	0	24
0	1	1	0	0	12
1	0	1	1	0	22
1	1	0	1	1	27
1	1	1	0	1	29
0	1	1	1	0	14
1	0	1	1	1	23
0	1	0	1	1	11
1	0	1	0	1	21
0	1	0	1	0	10
0	0	1	0	1	5

A	B	C	D	E	Decimal
0	0	0	1	0	2
0	0	0	0	1	1
1	0	0	0	0	16
0	1	0	0	0	8
0	0	1	0	0	4
1	0	0	1	0	18
0	1	0	0	1	9
1	0	1	0	0	20
1	1	0	1	0	26
0	1	1	0	1	13
0	0	1	1	0	6
1	0	0	1	1	19
1	1	0	0	1	25
1	1	1	0	0	28
1	1	1	1	0	30

If the register starts in the state 00000, it will stay in this state. Thus there are two possible sequences, one of length 31 steps and one of length 1 step. As $31 + 1 = 32 = 2^5$, and there are 5 stages to the shift register, we have discovered all the sequences for this configuration.

10. The pattern has a period of 6 units so a six stage shift register is required. From Figure 5 we see that the initial state needs to be 100011.



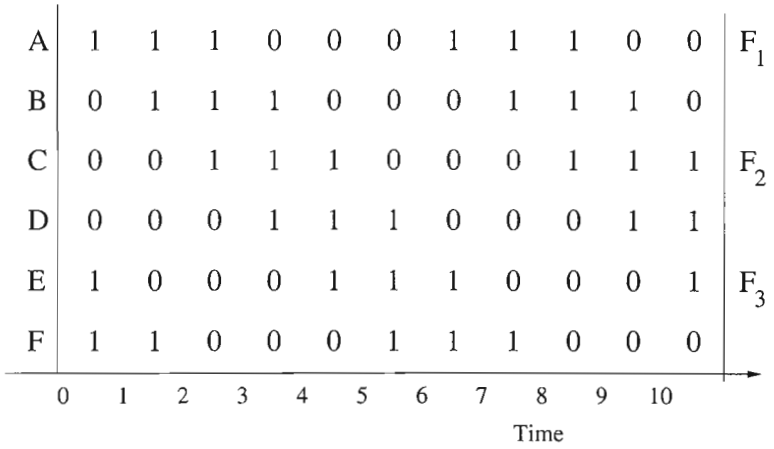


Figure 5:

11. For a 4-stage register, two sequences of length 8 are possible. $2^4 = 16 = 8 + 8$ so we have found all the sequences for this configuration.

0000	0010
1000	1001
1100	0100
1110	1010
1111	1101
0111	0110
0011	1011
0001	0101

For a 5-stage register there are four possible sequences with lengths 10, 10, 10 and 2. $2^5 = 32 = 10 + 10 + 10 + 2$ so we have found all the sequences for this configuration.

00000	00100	00010	10101
10000	10010	10001	01010
11000	11001	01000	
11100	01100	10100	
11110	10110	11010	
11111	11011	11101	
01111	01101	01110	
00111	00110	10111	
00011	10011	01011	
00001	01001	00101	

We are going to use the first of these sequences to generate a counter that runs from 0–9.

A	B	C	D	E	Output
0	0	0	0	0	0
1	0	0	0	0	1
1	1	0	0	0	2
1	1	1	0	0	3
1	1	1	1	0	4

A	B	C	D	E	Output
1	1	1	1	1	5
0	1	1	1	1	6
0	0	1	1	1	7
0	0	0	1	1	8
0	0	0	0	1	9

The 10 states are plotted in a 5-variable Karnaugh map in Figure 6. All the unlabelled squares may be taken as 'don't care' states. The logic function for output 1 is shown as two loops with a solid line: $A.\overline{B}$. The function for 5 is shown as a dotted line: $A.E$, and the function for 9 is shown as a dashed line: $\overline{D}.E$.

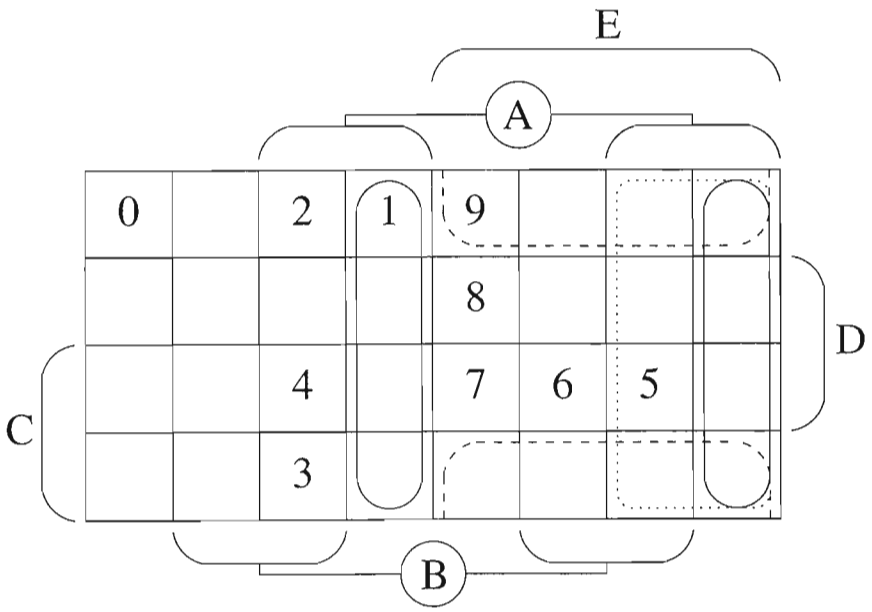


Figure 6: