Digital Logic Design

G V V Sharma*

Contents

***Abstract*—This manual provides a simple introduction to Digital Design.**

1 Seven Segment Display

1.1. Fig. 1.1.1 shows a seven segment display with pins a, b, c, d, e, f, g. Each of these pins is connected to an LED (light emitting device).

1.2. Fig. 1.2.1 shows how to generate the numbers on the display using Table 1.2.1. Complete Table 1.2.1 by drawing the figures for all numbers from 0-9.

a	b	c	d	e	f	g	decimal
1	0	0	1	1	1	1	1
0	0	1	0	0	1	0	2

TABLE 1.2.1

2 Incrementing Decoder

2.1. The incrementing decoder takes the numbers $0, 1, \ldots, 9$ in binary as inputs and generates the consecutive number as output. The corresponding *truth table* is available in Table. 2.1.1.

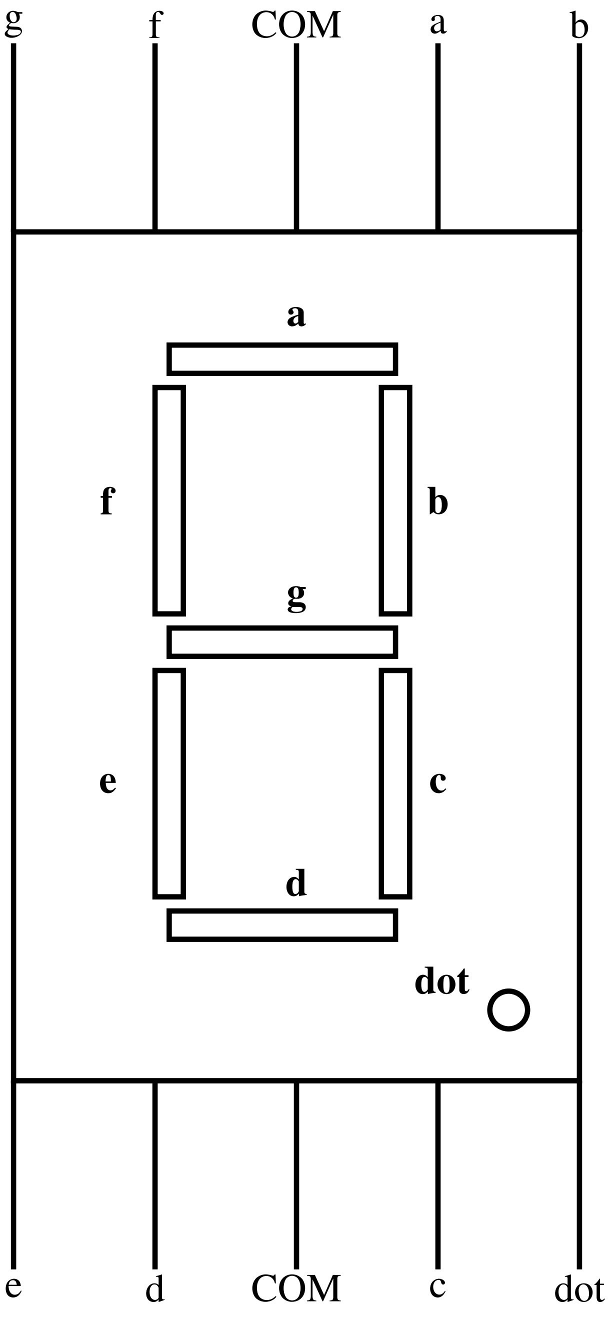


Fig. 1.1.1

2.2. Using Boolean logic, outputs A, B, C and D in Table 2.1.1 can be expressed in terms of the

[*] The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in. All content in this manual is released under GNU GPL. Free and open source.

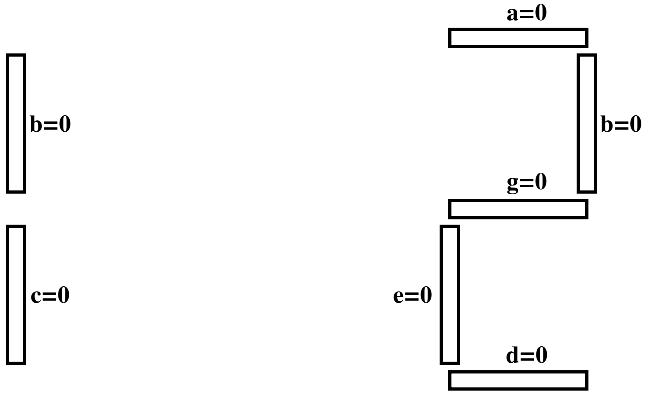


Fig. 1.2.1

Z	Y	X	W	**D**	**C**	**B**	**A**
0	0	0	0	**0**	**0**	**0**	**1**
0	0	0	1	**0**	**0**	**1**	**0**
0	0	1	0	**0**	**0**	**1**	**1**
0	0	1	1	**0**	**1**	**0**	**0**
0	1	0	0	**0**	**1**	**0**	**1**
0	1	0	1	**0**	**1**	**1**	**0**
0	1	1	0	**0**	**1**	**1**	**1**
0	1	1	1	**1**	**0**	**0**	**0**
1	0	0	0	**1**	**0**	**0**	**1**
1	0	0	1	**0**	**0**	**0**	**0**

TABLE 2.1.1: Truth table for the incrementing decoder

inputs W, X, Y, Z as

$$\begin{aligned} A &= W'X'Y'Z' + W'XY'Z' + W'X'YZ' \\ &\quad + W'XYZ' + W'X'Y'Z \qquad (2.2.1) \\ B &= WX'Y'Z' + W'XY'Z' \\ &\quad + WX'YZ' + W'XYZ' \qquad (2.2.2) \\ C &= WXY'Z' + W'X'YZ' \\ &\quad + WX'YZ' + W'XYZ' \qquad (2.2.3) \\ D &= WXYZ' + W'X'Y'Z \qquad (2.2.4) \end{aligned}$$

2.3. Execute the following code for different input values to verify (2.2.4).

```
codes/inc_decode.c
```

2.4. Modify the above C code to verify (2.2.1), (2.2.2) and (2.2.3).

2.5. Repeat the exercise for the truth table in 2.5.1.

D	C	B	A	**a**	**b**	**c**	**d**	**e**	**f**	**g**	**Decimal**
0	0	0	0	**0**	**0**	**0**	**0**	**0**	**0**	**1**	0
0	0	0	1	**1**	**0**	**0**	**1**	**1**	**1**	**1**	1
0	0	1	0	**0**	**0**	**1**	**0**	**0**	**1**	**0**	2
0	0	1	1	**0**	**0**	**0**	**0**	**1**	**1**	**0**	3
0	1	0	0	**1**	**0**	**0**	**1**	**1**	**0**	**0**	4
0	1	0	1	**0**	**1**	**0**	**0**	**1**	**0**	**0**	5
0	1	1	0	**0**	**1**	**0**	**0**	**0**	**0**	**0**	6
0	1	1	1	**0**	**0**	**0**	**1**	**1**	**1**	**1**	7
1	0	0	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	8
1	0	0	1	**0**	**0**	**0**	**1**	**1**	**0**	**0**	9

TABLE 2.5.1: Truth table for display decoder.

3 Karnaugh Map

3.1 Incrementing Decoder

3.1.1. K-Map for A: The expression in (2.2.1) can be minimized using the K-map in Fig. 3.1.1.1. In Fig. 3.1.1.1, the *implicants* in boxes 0,2,4,6 result in $W'Z'$. The implicants in boxes 0,8 result in $W'X'Y'$. Thus, after minimization using Fig. 3.1.1.1, (2.2.1) can be expressed as

$$A = W'Z' + W'X'Y' \qquad (3.1.1.1)$$

Using the fact that

$$\begin{aligned} X + X' &= 1 \\ XX' &= 0, \end{aligned} \qquad (3.1.1.2)$$

derive (3.1.1.1) from (2.2.1) algebraically.

3.1.2. K-Map for B: From Table 2.1.1, using boolean logic, Show that (2.2.2) can be reduced to

$$B = WX'Z' + W'XZ' \qquad (3.1.2.1)$$

using Fig. 3.1.2.1.

3.1.3. Derive (3.1.2.1) from (2.2.2) algebraically using (3.1.1.2).

3.1.4. K-Map for C: From Table 2.1.1, using boolean logic, Show that (2.2.3) can be reduced to

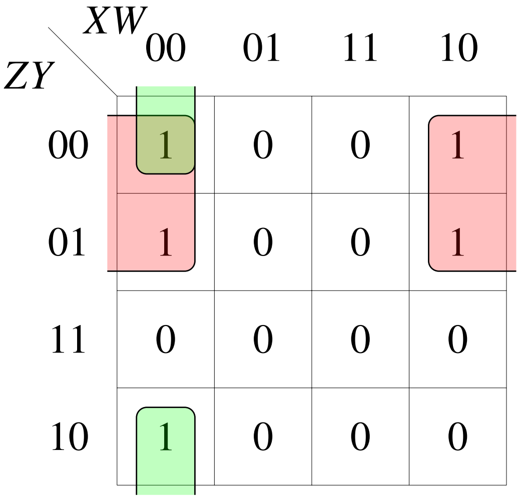
$$C = WXY'Z' + X'YZ' + W'YZ' \qquad (3.1.4.1)$$

using Fig. 3.1.4.1.

3.1.5. Derive (3.1.4.1) from (2.2.3) algebraically using (3.1.1.2).

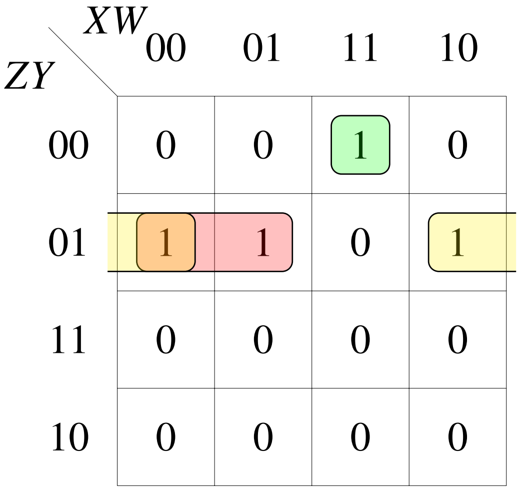
3.1.6. K-Map for D: From Table 2.1.1, using boolean logic,

$$D = WXYZ' + W'X'Y'Z \qquad (3.1.6.1)$$



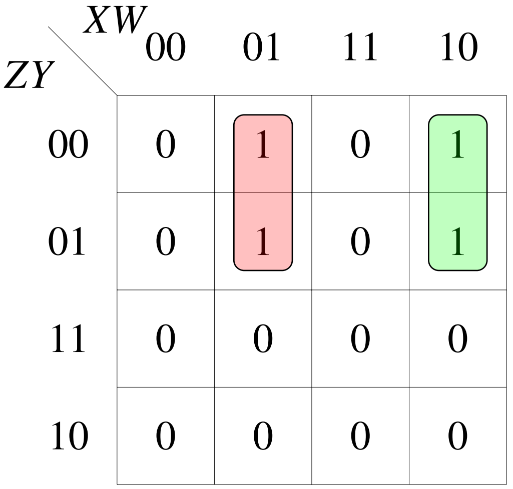
$ZY \backslash XW$	00	01	11	10
00	1	0	0	1
01	1	0	0	1
11	0	0	0	0
10	1	0	0	0

Fig. 3.1.1.1: K-map for A.



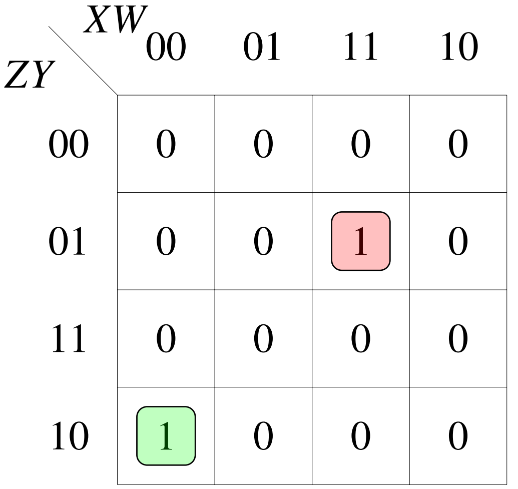
$ZY \backslash XW$	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	0	0	0	0
10	0	0	0	0

Fig. 3.1.4.1: K-map for C.



$ZY \backslash XW$	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	0	0	0	0
10	0	0	0	0

Fig. 3.1.2.1: K-map for B.



$ZY \backslash XW$	00	01	11	10
00	0	0	0	0
01	0	0	1	0
11	0	0	0	0
10	1	0	0	0

Fig. 3.1.6.1: K-map for D.

3.1.7. Minimize (3.1.6.1) using Fig. 3.1.6.1.

3.1.8. Modify your C program to verify the the K-Map equations for A,B,C and D in (3.1.1.1), (3.1.1.1), (3.1.1.1) and (3.1.1.1) respectively.

3.1.9. Revise by using don't care conditions and verify through a C code.

3.2 Display Decoder

Use K-maps to obtain the minimized expressions for a, b, c, d, e, f, g in terms of A, B, C, D in Table 2.5.1 without don't care conditions.

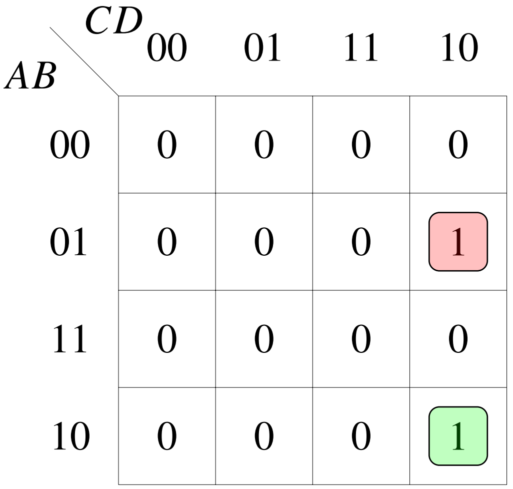
3.2.1. Obtain the expression for b using Fig. 3.2.1.1
Solution:

$$b = AB'CD' + A'BCD' \tag{3.2.1.1}$$

$$= CD'\left(AB' \oplus A'B\right) \tag{3.2.1.2}$$

where $\oplus$ denotes the XOR operation.

3.2.2. Obtain the expression for d using Fig. 3.2.2.1
Solution:

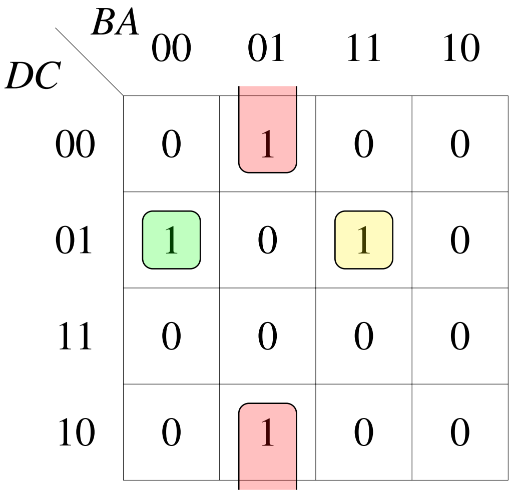


AB \ CD	00	01	11	10
00	0	0	0	0
01	0	0	0	1
11	0	0	0	0
10	0	0	0	1

Fig. 3.2.1.1: K-map for *b*.

$$d = AB'C' + A'B'CD' + ABCD' \tag{3.2.2.1}$$

where ⊕ denotes the XOR operation.



DC \ BA	00	01	11	10
00	0	1	0	0
01	1	0	1	0
11	0	0	0	0
10	0	1	0	0

Fig. 3.2.2.1: K-map for *d*.

4 Don't Care

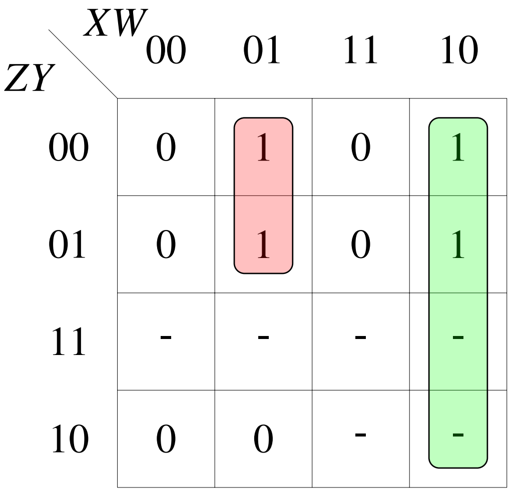
4.1 Incrementing Decoder

4.1.1. Obtain the expression for B using Fig. 4.1.1.1

Solution:

$$B = W'X + WX'Z' \tag{4.1.1.1}$$

where ⊕ denotes the XOR operation.

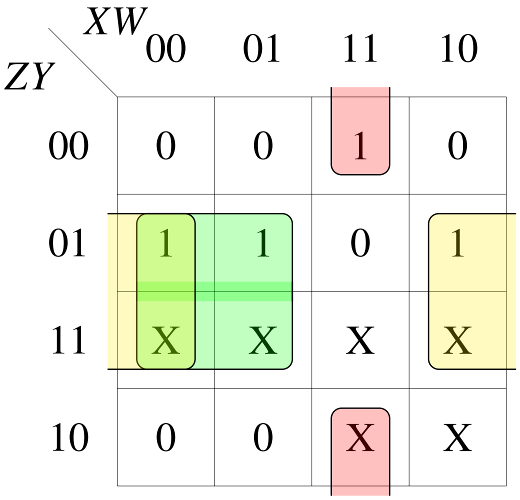


ZY \ XW	00	01	11	10
00	0	1	0	1
01	0	1	0	1
11	-	-	-	-
10	0	0	-	-

Fig. 4.1.1.1: K-map for *B*.

4.1.2. Obtain the expression for C using Fig. 4.1.2.1

$$C = Y'X + W'XZ' + YX' \tag{4.1.2.1}$$



ZY \ XW	00	01	11	10
00	0	0	1	0
01	1	1	0	1
11	X	X	X	X
10	0	0	X	X

Fig. 4.1.2.1: K-map for *C*.

5 Programming

6 Logic Gates