

Digital Logic Design

G V V Sharma*

CONTENTS

1	Seven Segment Display	1
2	Incrementing Decoder	1
3	Display Decoder	2
4	Karnaugh Map	2
4.1	Incrementing Decoder	2
4.2	Display Decoder	4
5	Don't Care	4
5.1	Incrementing Decoder	4
5.2	Display Decoder	5
6	Programming	6
7	Logic Gates	6
8	Product of Sums	6

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

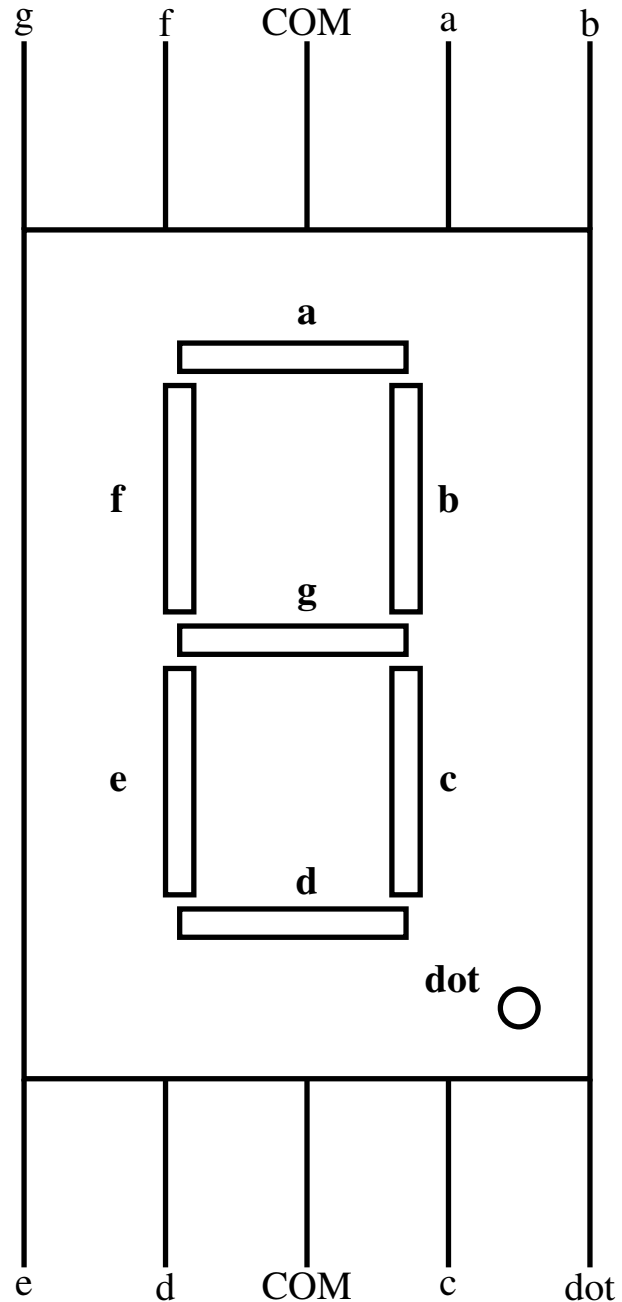


Fig. 1.1.1

2 INCREMENTING DECODER

- 2.1. The incrementing decoder takes the numbers 0, 1, . . . , 9 in binary as inputs and generates 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.

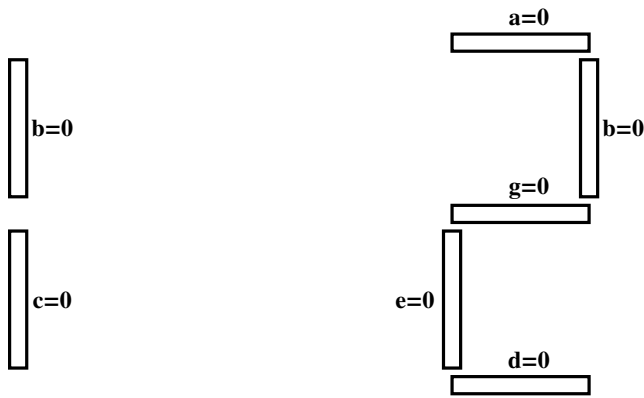


Fig. 1.2.1

consecutive number as output. The corresponding *truth table* is available in Table. 2.1.1.

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

2.2. Using Boolean logic, outputs A, B, C and D in Table 2.1.1 can be expressed in terms of the inputs W, X, Y, Z as

$$A = W'X'Y'Z' + W'XY'Z' + W'X'YZ' + W'XYZ' + W'X'Y'Z \quad (2.2.1)$$

$$B = WX'Y'Z' + W'XY'Z' + WX'YZ' + W'XYZ' \quad (2.2.2)$$

$$C = WXY'Z' + W'X'YZ' + WX'YZ' + W'XYZ' \quad (2.2.3)$$

$$D = WXYZ' + W'X'Y'Z \quad (2.2.4)$$

2.3. Execute the following code for different input values to verify (2.2.1)-(2.2.4).

```
codes/inc_decode.c
```

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 3.1.1: Truth table for display decoder.

3 DISPLAY DECODER

3.1. Using Boolean logic, outputs a, b, c, d, e, f, g in Table 3.1.1 can be expressed in terms of the inputs A, B, C, D as

$$a = \quad (3.1.1)$$

$$b = AB'CD' + A'BCD' \quad (3.1.2)$$

$$c = D'C'BA' \quad (3.1.3)$$

$$d = \quad (3.1.4)$$

$$e = \quad (3.1.5)$$

$$f = \quad (3.1.6)$$

$$g = \quad (3.1.7)$$

3.2. Execute the following code for different input values to verify (3.1.1)-(3.1.7).

```
codes/disg_decode.c
```

4 KARNAUGH MAP

4.1 Incrementing Decoder

4.1.1. K-Map for A: The expression in (2.2.1) can be minimized using the K-map in Fig. 4.1.1.1. In Fig. 4.1.1.1, 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. 4.1.1.1, (2.2.1) can be expressed as

$$A = W'Z' + W'X'Y' \quad (4.1.1.1)$$

Using the fact that

$$X + X' = 1 \quad (4.1.1.2)$$

$$XX' = 0,$$

derive (4.1.1.1) from (2.2.1) algebraically.

ZY \ 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. 4.1.1.1: K-map for A.

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

ZY \ 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. 4.1.2.1: K-map for B.

$$B = WX'Z' + W'XZ' \quad (4.1.2.1)$$

using Fig. 4.1.2.1.

4.1.3. Derive (4.1.2.1) from (2.2.2) algebraically using (4.1.1.2).

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

ZY \ 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. 4.1.4.1: K-map for C.

$$C = WXY'Z' + X'YZ' + W'YZ' \quad (4.1.4.1)$$

using Fig. 4.1.4.1.

4.1.5. Derive (4.1.4.1) from (2.2.3) algebraically using (4.1.1.2).

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

$$D = WXYZ' + W'X'Y'Z \quad (4.1.6.1)$$

ZY \ 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. 4.1.6.1: K-map for D.

4.1.7. Minimize (4.1.6.1) using Fig. 4.1.6.1.

- 4.1.8. Modify your C program to verify the the K-Map equations for A,B,C and D in (4.1.1.1), (4.1.1.1), (4.1.1.1) and (4.1.1.1) respectively.
- 4.1.9. Revise by using don't care conditions and verify through a C code.

4.2 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 3.1.1 without don't care conditions.

- 4.2.1. Obtain the expression for b using Fig. 4.2.1.1

Solution:

$$b = AB'CD' + A'BCD' \quad (4.2.1.1)$$

$$= CD'(AB' \oplus A'B) \quad (4.2.1.2)$$

where \oplus denotes the XOR operation.

AB \ CD	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. 4.2.1.1: K-map for b .

- 4.2.2. Obtain the expression for d using Fig. 4.2.2.1

Solution:

$$d = AB'C' + A'B'CD' + ABCD' \quad (4.2.2.1)$$

where \oplus denotes the XOR operation.

- 4.2.3. Obtain the expression for e using Fig. 4.2.3.1

$$e = AD' + B'CD' + AB'C' \quad (4.2.3.1)$$

- 4.2.4. Obtain the expression for f using Fig. 4.2.4.1

$$f = BC'D' + AB'C'D' + AB'CD \quad (4.2.4.1)$$

DC \ BA	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. 4.2.2.1: K-map for d .

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

Fig. 4.2.3.1: K-map for e .

- 4.2.5. Obtain the expression for g using Fig. 4.2.5.1

$$g = B'C'D' + ABCD' \quad (4.2.5.1)$$

5 DON'T CARE

5.1 Incrementing Decoder

- 5.1.1. Obtain the expression for B using Fig. 5.1.1.1

Solution:

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

Fig. 4.2.4.1: K-map for f .

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

Fig. 5.1.1.1: K-map for B .

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

Fig. 4.2.5.1: K-map for g .

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. 5.1.2.1: K-map for C .

5.2 Display Decoder

$$B = W'X + WX'Z' \quad (5.1.1.1)$$

where \oplus denotes the XOR operation.

5.1.2. Obtain the expression for C using Fig. 5.1.2.1

$$C = Y'X + W'XZ' + YX' \quad (5.1.2.1)$$

5.2.1. Obtain the expression for b using Fig. 5.2.1.1

Solution:

$$b = B' + CD + C'D' \quad (5.2.1.1)$$

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

Fig. 5.2.1.1: K-map for b using don't care.

6 PROGRAMMING

7 LOGIC GATES

7.1. The following equation is implemented using gates in Fig. 7.1.1

$$B = W'X + WX'Z' \quad (7.1.1)$$

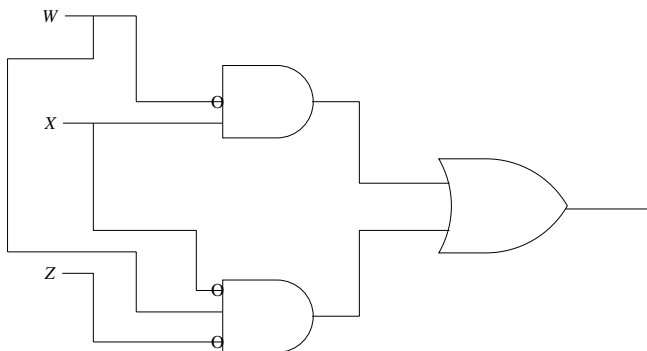


Fig. 7.1.1

7.2. The following equation is implemented using gates in Fig. 7.2.1

$$b = B' + CD + C'D' \quad (7.2.1)$$

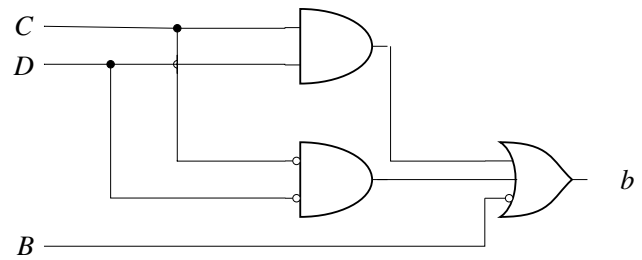


Fig. 7.2.1

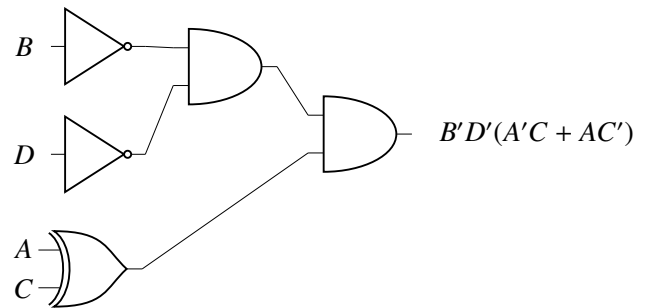


Fig. 7.3.1

7.4. The following equation is implemented using gates in Fig. 7.4.1

$$d = AB'C' + A'B'CD' + ABCD' \quad (7.4.1)$$

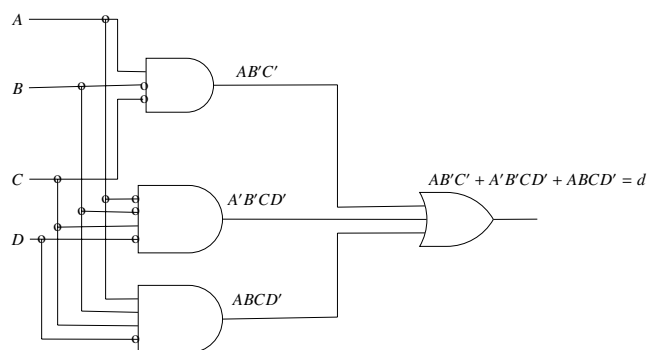


Fig. 7.4.1

7.3. The following equation is implemented using gates in Fig. 7.3.1

$$a = B'D'(A'C + AC') \quad (7.3.1)$$

8 PRODUCT OF SUMS

Using the 0s in Table 2.1.1, the product of sums (POS) expressions are obtained from Figs. 8.2-8.5

as

$$A = (Z' + Y')W'(Z' + X') \quad (8.1)$$

$$B = (X' + W')Z'(X + W) \quad (8.2)$$

$$C = (Z + Y + X)(Y' + X' + W')(X' + Y + W)Z' \quad (8.3)$$

$$D = (Z + Y)(Y' + X)(X + W')(X' + W)(Z' + X') \quad (8.4)$$

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

Fig. 8.2: POS for A

ZY	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. 8.3: POS for B

ZY	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. 8.4: POS for C

ZY	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. 8.5: POS for D