Karnaugh Map

G V V Sharma*

Contents

***Abstract*—This manual explains Karnaugh maps (K-map) and state machines by deconstructing a decade counter.**

1 The Decade Counter

The block diagram of a decade counter (repeatedly counts up from 0 to 9) is available in Fig. 0. The *incrementing* decoder and *display* decoder are part of *combinational* logic, while the *delay* is part of *sequential* logic.

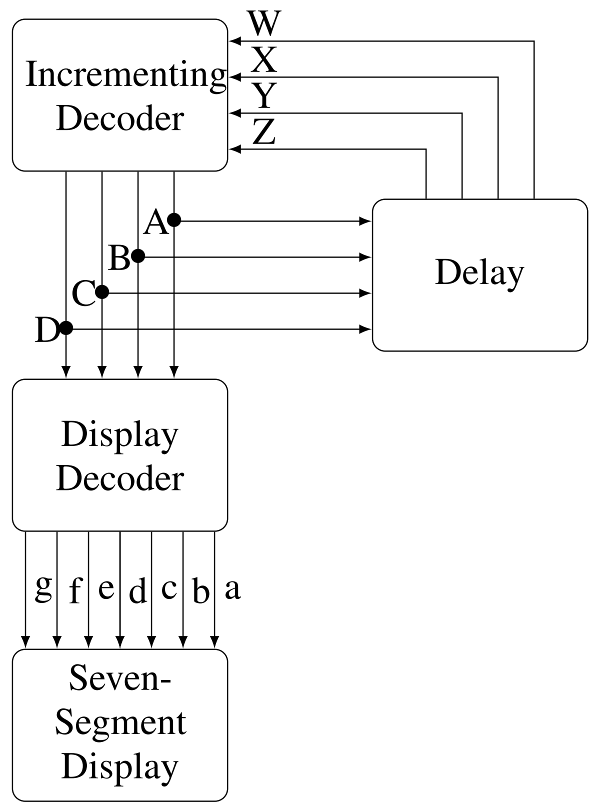

Fig. 0: The decade counter

2 Incrementing Decoder

The incrementing decoder in Fig. 0 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 Fig. 0.

3 Karnaugh Map

Using Boolean logic, output A in Table 0 can be expressed in terms of the inputs W, X, Y, Z as

$$A = W'X'Y'Z' + W'XY'Z' + W'X'YZ' + W'XYZ' + W'X'Y'Z \quad (3.1)$$

*The author is with the Department of Electrical Engineering, Indian Institute of Technology, Hyderabad 502285 India e-mail: gadepall@iith.ac.in.

1. K-Map for A: The expression in (3.1) can be minimized using the K-map in Fig. 1. In Fig. 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. 3.2, (3.1) can be expressed as

$$A = W'Z' + W'X'Y' \quad (3.2)$$

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 0

Using the fact that

$$\begin{aligned} X + X' &= 1 \\ XX' &= 0, \end{aligned} \tag{3.3}$$

derive (3.2) from (3.1) algebraically.

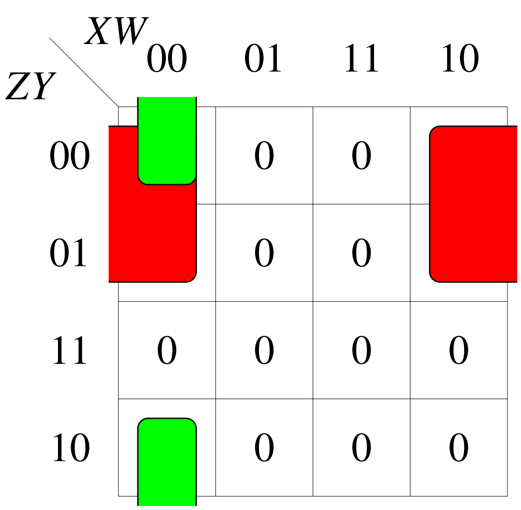

Fig. 1: K-map for A.

2. K-Map for B: From Table 0, using boolean logic,

$$B = WX'Y'Z' + W'XY'Z' + WX'YZ' + W'XYZ' \tag{3.4}$$

Show that (3.4) can be reduced to

$$B = WX'Z' + W'XZ' \tag{3.5}$$

using Fig. 2.

3. Derive (3.5) from (3.4) algebraically using (3.3).

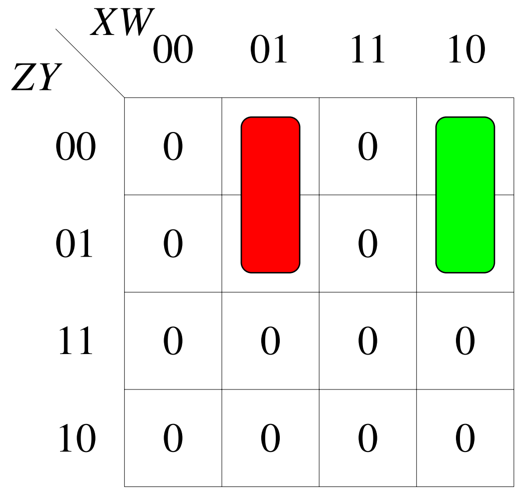

Fig. 2: K-map for B.

4. K-Map for C: From Table 0, using boolean logic,

$$C = WXY'Z' + W'X'YZ' + WX'YZ' + W'XYZ' \tag{3.6}$$

Show that (3.6) can be reduced to

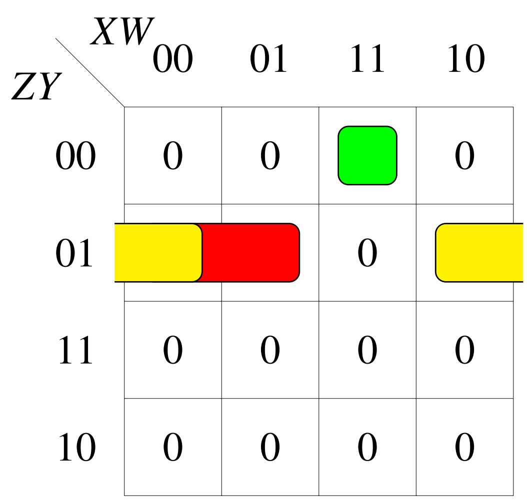

Fig. 4: K-map for C.

$$C = WXY'Z' + X'YZ' + W'YZ' \tag{3.7}$$

using Fig. 4.

5. Derive (3.7) from (3.6) algebraically using (3.3).
6. K-Map for D: From Table 0, using boolean

logic,

$$D = WXYZ' + W'X'Y'Z \tag{3.8}$$

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

Fig. 6: K-map for D.

7. Minimize (3.8) using Fig. 6.
8. Download the code in

```
wget https://raw.githubusercontent.com/gadepall
    /arduino/master/7447/codes/inc_dec/
    inc_dec.ino
```

and modify it using the K-Map equations for A,B,C and D. Execute and verify.

9. Display Decoder: Table 8 is the truth table for the display decoder in Fig. 0. Use K-maps to obtain the minimized expressions for a, b, c, d, e, f, g in terms of A, B, C, D with and without don't care conditions.

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 9: Truth table for display decoder.