Electronics Club Handout # 2

Logic simplification with Karnaugh maps

Rules of Boolean Algebra

1. $A + 0 = A$	**7.** $A \cdot A = A$
2. $A + 1 = 1$	**8.** $A \cdot \overline{A} = 0$
3. $A \cdot 0 = 0$	**9.** $\overline{\overline{A}} = A$
4. $A \cdot 1 = A$	**10.** $A + AB = A$
5. $A + A = A$	**11.** $A + \overline{A}B = A + B$
6. $A + \overline{A} = 1$	**12.** $(A + B)(A + C) = A + BC$

A, *B*, or *C* can represent a single variable or a combination of variables.

Karnaugh maps

- Graphical method for simplification of ***boolean*** expression.
- It is a graphical chart which contain boxes.
- K-maps can be written for 2,3,4…. Upto 6 variables .
  Beyond that the technique becomes very cumbersome.

Karnaugh Map

$AB \backslash C$	0	1
00		
01		
11		
10		

(a)

$AB \backslash C$	0	1
00	$\overline{A}\,\overline{B}\,\overline{C}$	$\overline{A}\,\overline{B}C$
01	$\overline{A}B\overline{C}$	$\overline{A}BC$
11	$AB\overline{C}$	ABC
10	$A\overline{B}\,\overline{C}$	$A\overline{B}C$

(b)

How does simplification takes place ?

- Once we plot the logic function or truth table on a k-map , we have to use the grouping technique for simplifying the logic function.
- Grouping means combining the terms in the adjacent boxes.

Way of grouping

- The grouping follows the binary rule i.e. we can group 1,2,,4,8,16,32… number of ***1's*** or ***0's.*** We cannot group 3,5,7,9…. Number of ***1's*** or ***0's***.
- **Pairs** : A group of adjacent **1's** or **0's** is called a pair.
- **Quad** : A group of four adjacent **1's** or **0's** is called a quad.

Grouping of two adjacent I's (pair)

SIMPLIFICATION :-

Y = ABC' + AB'C' = AC'(B + B')

= AC' …. Since (B+B') = 1

ABC' AB'C'

C \ A B	A'B'	A'B	AB	AB'
C'	0	0	1	1
C	0	0	0	0

Conclusion :- By pairing two adjacent 1's we can eliminate one variable.

Simplification:-

Y = A'BC' + A'BC = A'B(C' + C)

= A'B …. SINCE C'+C = 1

A'B

C \ A B	A'B'	A'B	AB	AB'
C'	0	1	0	0
C	0	1	0	0

Simplification:-

Y = A'B'C + AB'C = B'C(A' + A)

= B'C …. SINCE A'+A = 1

C \ A B	A'B'	A'B	AB	AB'
C'	0	0	0	0
C	1	0	0	1

B'C

4 Variable K MAP

For eg:-

CD \ AB	A'B'	A'B	AB	AB'
C'D'	0	0	0	0
C'D	0	0	0	0
CD	0	0	0	0
CD'	1	1	1	1

Y = CD'

- The two variables which are same in all minterms are C and D' .
- The variables which are not same in all variables are A and B .
- A and B will be eliminated and the output will be :

  Y = C D'

Simplification :-

Y = A'B'CD' + A'BCD' + ABCD' + AB'CD' = CD' (A'B' + A'B + AB + AB')
= CD' [A'(B'+B) + A(B+B')]
= CD'[A+A'] = CD' ......Proved.

Conclusion:-
Two variables are eliminated

Top and bottom 1's forming a quad

CD \ AB	A'B'	A'B	AB	AB'
C'D'	0	1	1	0
C'D	0	0	0	0
CD	0	0	0	0
CD'	0	1	1	0

Y = BD'

A and C are changing so they are eliminated

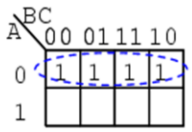
Four adjacent 1's forming a square.

CD \ AB	A'B'	A'B	AB	AB'
C'D'	0	0	0	0
C'D	1	1	0	0
CD	1	1	0	0
CD'	0	0	0	0

Y = A'D

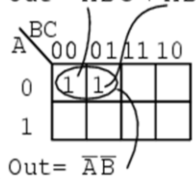
B and C are changing so they are eliminated

Out = $\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + \bar{A}B\bar{C}$



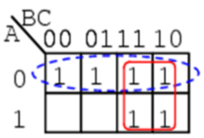
Out = $\bar{A}$

Out = $\bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C$



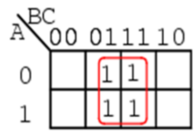
Out = $\bar{A}\bar{B}$

Out = $\bar{A}\bar{B}\bar{C}+\bar{A}\bar{B}C+\bar{A}BC+\bar{A}B\bar{C}+ABC+AB\bar{C}$



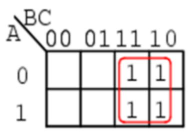
Out = $\bar{A} + B$

Out = $\bar{A}\bar{B}C + \bar{A}BC + A\bar{B}C + ABC$



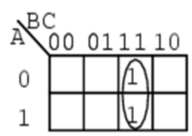
Out = C

Out = $\bar{A}BC + \bar{A}B\bar{C} + ABC + AB\bar{C}$



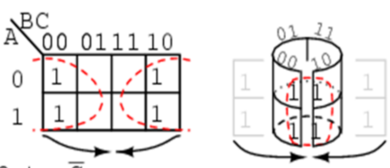
Out = B

Out = $\bar{A}BC + ABC$



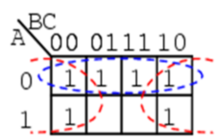
Out = BC

Out = $\bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + \bar{A}B\bar{C} + AB\bar{C}$



Out = $\bar{C}$

Out = $\bar{A}\bar{B}\bar{C}+\bar{A}\bar{B}C+\bar{A}BC+\bar{A}B\bar{C}+A\bar{B}\bar{C}+AB\bar{C}$



Out = $\bar{A} + \bar{C}$

Determining Standard Expressions from Truth Table

INPUTS			OUTPUT
A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

There are four 1s in the output column and the corresponding binary values are 011, 100, 110, and 111. These binary values are converted to product terms as follows:

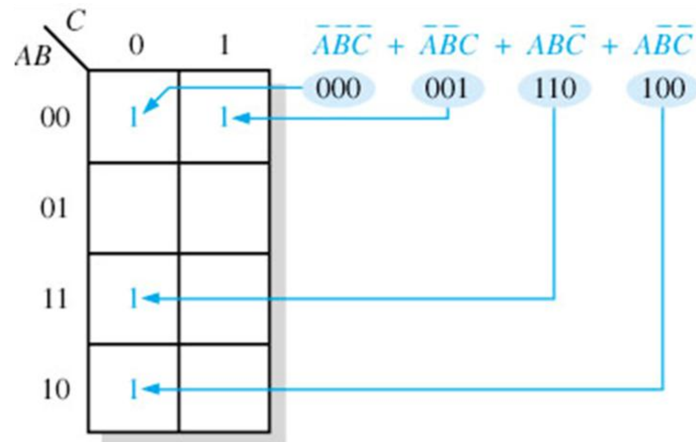
$$011 \longrightarrow \overline{A}BC$$
$$100 \longrightarrow A\overline{B}\,\overline{C}$$
$$110 \longrightarrow AB\overline{C}$$
$$111 \longrightarrow ABC$$

SOP =Sum of Product

The resulting standard SOP expression for the output X is

$$X = \overline{A}BC + A\overline{B}\,\overline{C} + AB\overline{C} + ABC$$

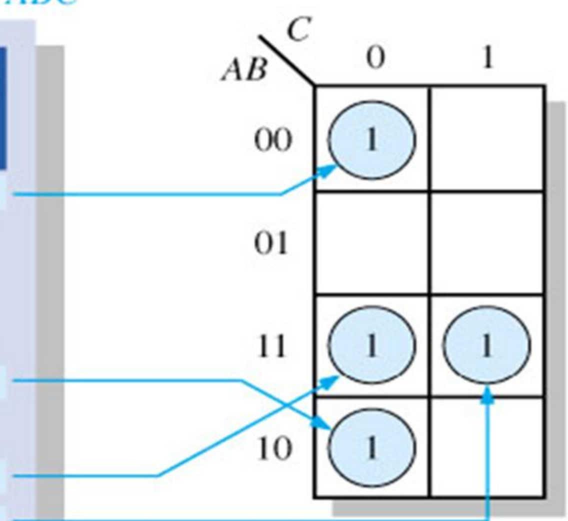
Karnaugh Map SOP Minimization



Mapping Directly from Truth Table

$$X = \overline{A}\,\overline{B}\,\overline{C} + A\overline{B}\,\overline{C} + AB\overline{C} + ABC$$

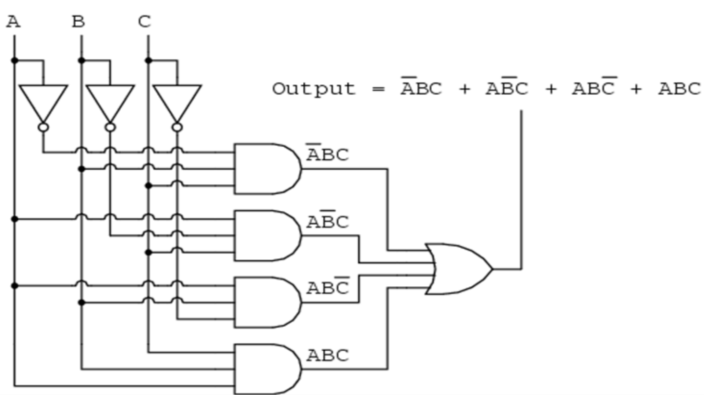
Inputs A	B	C	Output X
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1



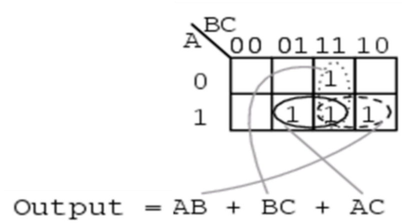
- Consider a situation in which you have 3 user inputs (A, B & C)and you have to detect either 2 or 3 user inputs are ON (X)

A	B	C	X
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Without Simplification



Using K-Map



$\overline{A}BC + A\overline{B}C + AB\overline{C} + ABC$

↓ Factoring BC out of 1st and 4th terms

$BC(\overline{A} + A) + A\overline{B}C + AB\overline{C}$

↓ Applying identity $A + \overline{A} = 1$

$BC(1) + A\overline{B}C + AB\overline{C}$

↓ Applying identity $1A = A$

$BC + A\overline{B}C + AB\overline{C}$

↓ Factoring B out of 1st and 3rd terms

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

↓ Applying rule $A + \overline{A}B = A + B$ to the $C + A\overline{C}$ term

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

↓ Distributing terms

$BC + AB + A\overline{B}C$

↓ Factoring A out of 2nd and 3rd terms

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

↓ Applying rule $A + \overline{A}B = A + B$ to the $B + \overline{B}C$ term

$BC + A(B + C)$

↓ Distributing terms

$BC + AB + AC$

or Simplified result

$AB + BC + AC$

Simplification Boolean Algebra

Observe the Power of K-MAP as number of variables increases

Example-

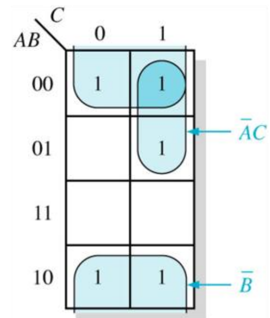
Use a Karnaugh map to minimize the following standard SOP expression:

$$A\bar{B}C + \bar{A}BC + \bar{A}\bar{B}C + \bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C}$$

The binary values of the expression are

101+ 011+ 001+ 000+ 100

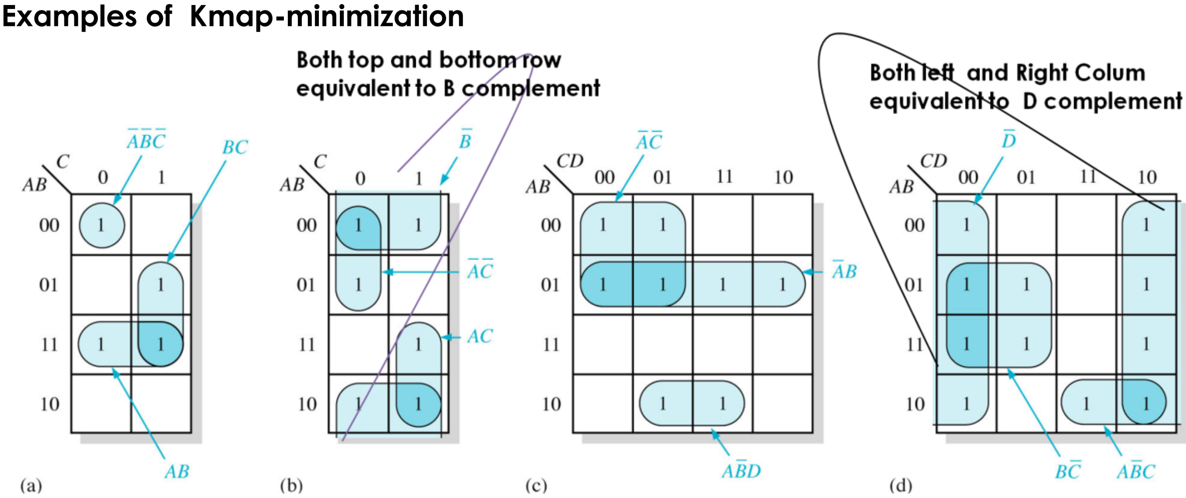
$\bar{A}C + \bar{B}$



Related Problem Use a Karnaugh map to simplify the following standard SOP expression:

$$X\bar{Y}Z + XY\bar{Z} + \bar{X}YZ + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + XYZ$$

Examples of Kmap-minimization

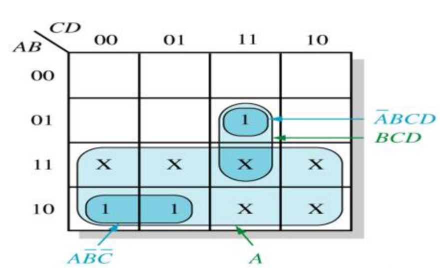


"Don't Care" Conditions

Inputs A B C D	Output Y
0 0 0 0	0
0 0 0 1	0
0 0 1 0	0
0 0 1 1	0
0 1 0 0	0
0 1 0 1	0
0 1 1 0	0
0 1 1 1	1
1 0 0 0	1
1 0 0 1	1
1 0 1 0	X
1 0 1 1	X
1 1 0 0	X
1 1 0 1	X
1 1 1 0	X
1 1 1 1	X

Don't cares

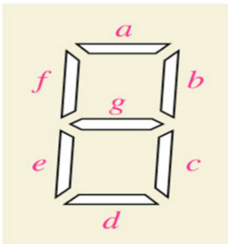
(a) Truth table



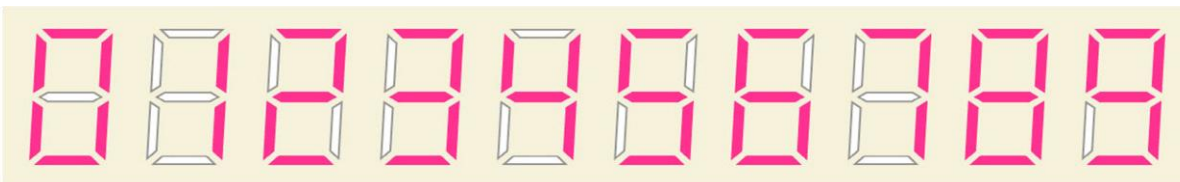
(b) Without "don't cares" $Y = A\bar{B}\bar{C} + \bar{A}BCD$
With "don't cares" $Y = A + BCD$

Example of the use of "don't care" conditions to simplify an expression.

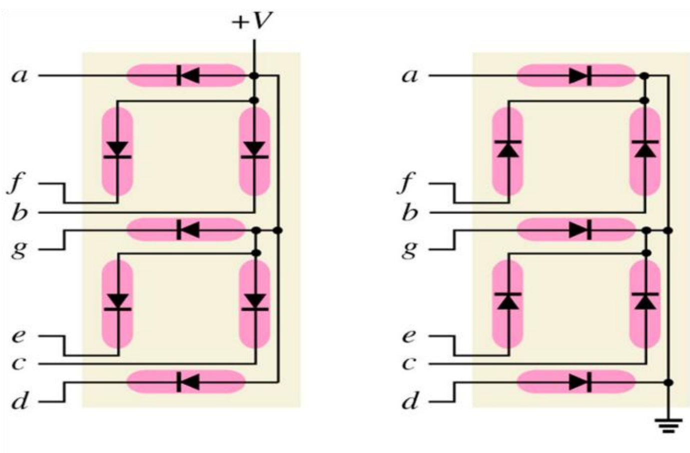
Digital System Application



Seven-segment display format showing arrangement of segments



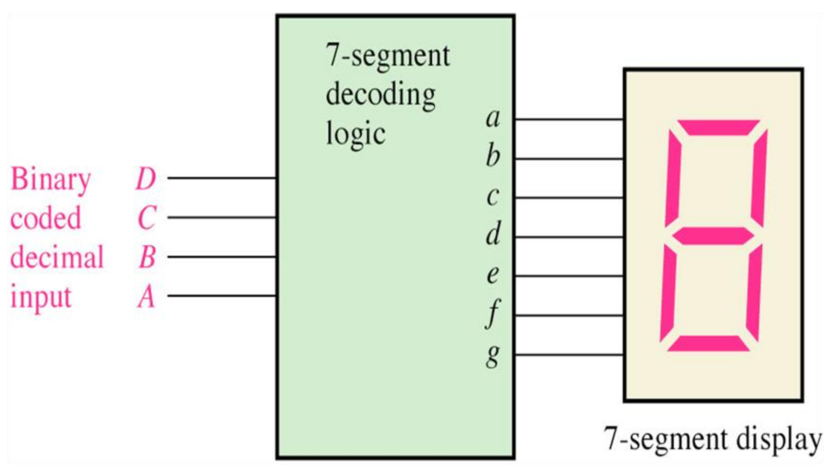
Display of decimal digits with a 7-segment device.



(a) Common-anode (b) Common-cathode

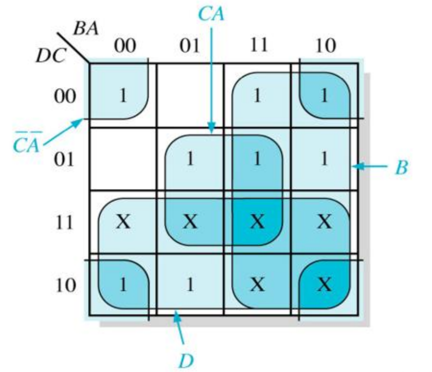
Arrangements of 7-segment LED displays.

Block diagram of 7-segment logic and display



Standard SOP expression:

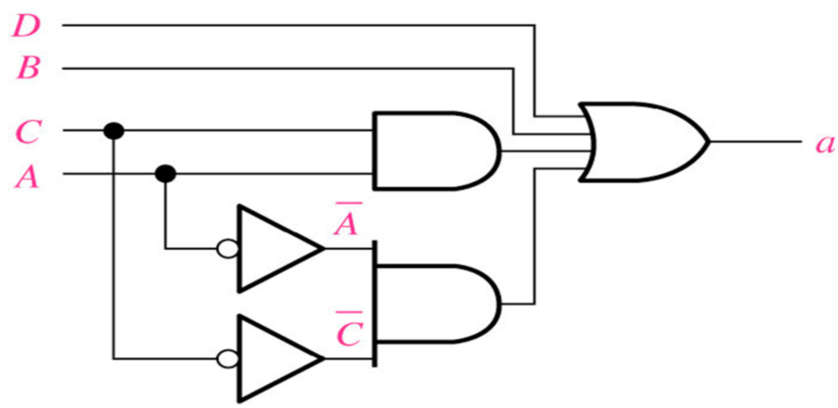
$\overline{D}\,\overline{C}\,\overline{B}\,\overline{A} + \overline{D}\,\overline{C}B\overline{A} + \overline{D}\,\overline{C}BA + \overline{D}C\overline{B}A + \overline{D}CB\overline{A} + \overline{D}CBA + D\overline{C}\,\overline{B}\,\overline{A} + D\overline{C}\,\overline{B}A$



Minimum SOP expression: $D + B + CA + \overline{C}\,\overline{A}$

Karnaugh map minimization of the segment-*a* logic expression

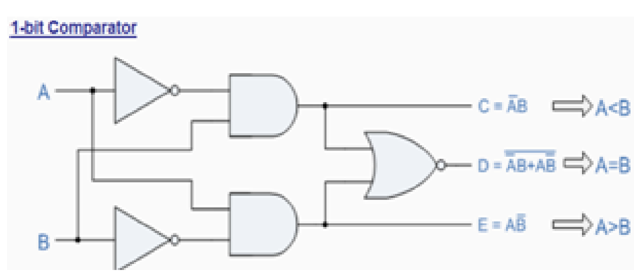
The minimum logic implementation for segment *a* of the 7-segment display.



Now you can think about these problems

- **Can you make a counter which count from 3 to 8 ?**
- **Repeat the 7 segment BCD display example if we don't care about the output values for illegal input from 10 to 15. (Note that in that example we took X=0 for illegal input.**
- **How does MUX work ? Can you make it with using some simple logic ?**
- **Can you make 1 bit comparator ? (solution is given ,so verify it and try for more bit)**
- **<u>Truth table of 1 bit comparator</u>** **<u>Logical Implementation</u>**

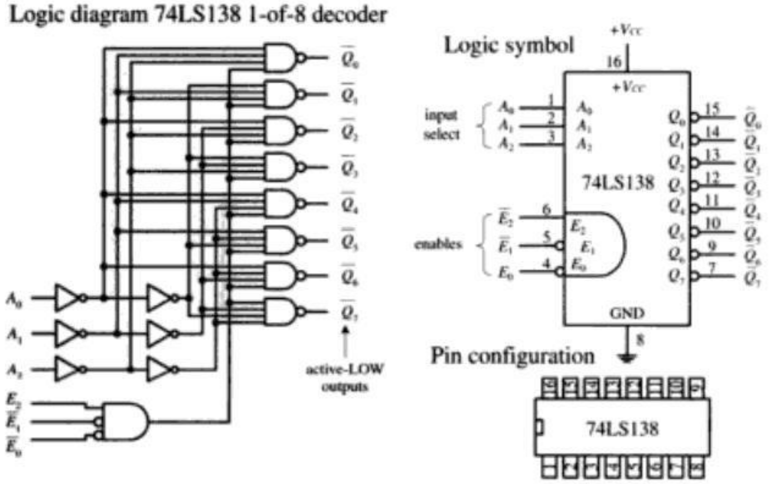
Inputs		Outputs		
B	A	A > B	A = B	A < B
0	0	0	1	0
0	1	1	0	0
1	0	0	0	1
1	1	0	1	0



Reference

MUX

Logic diagram 74LS138 1-of-8 decoder



Truth table for 74LS138

$\overline{E}_0$	$\overline{E}_1$	E_2	A_0	A_1	A_2	$\overline{Q}_0$	$\overline{Q}_1$	$\overline{Q}_2$	$\overline{Q}_3$	$\overline{Q}_4$	$\overline{Q}_5$	$\overline{Q}_6$	$\overline{Q}_7$
H	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
X	X	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

enabled

L = LOW voltage level
H = HIGH voltage level
X = don't care

BCD Decoder Truth table

Seven-segment display decoder truth table

$D_{3:0}$	S_a	S_b	S_c	S_d	S_e	S_f	S_g
0000	1	1	1	1	1	1	0
0001	0	1	1	0	0	0	0
0010	1	1	0	1	1	0	1
0011	1	1	1	1	0	0	1
0100	0	1	1	0	0	1	1
0101	1	0	1	1	0	1	1
0110	1	0	1	1	1	1	1
0111	1	1	1	0	0	0	0
1000	1	1	1	1	1	1	1
1001	1	1	1	0	0	1	1
others	0	0	0	0	0	0	0

Logic and Computer Design Fundamentals by M. Morris Mano, Charle Kime (easily available in reserve section or from your seniors) (Chapter 2 and Chapter 3 in 2^{nd} edition)