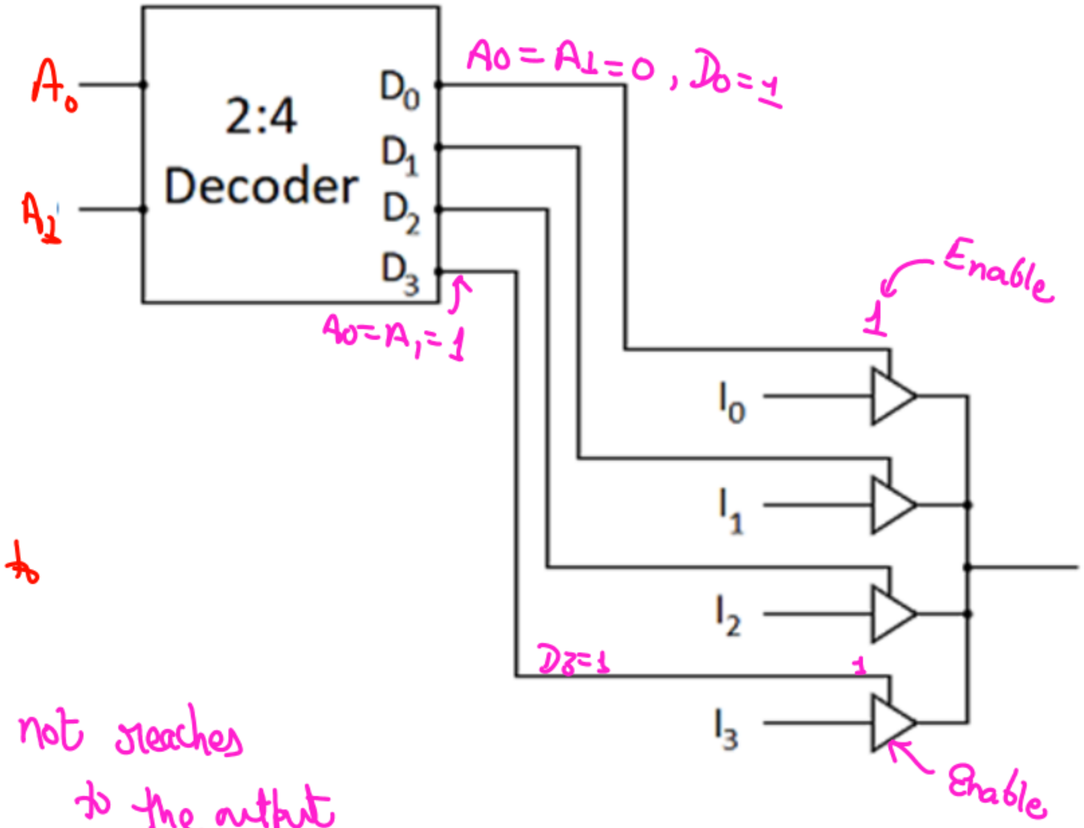
DECODERS AND ENCODERS (PART 02)

Implementation of MUX using Decoder

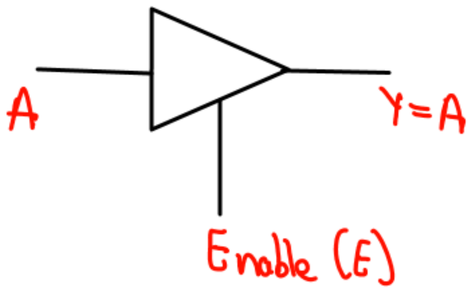
To implement Multiplexer using Decoder,

- The Inputs of decoder act like 'Select Line' of MUX
- The output of decoder will enable the Tri-State Buffer or AND gate to pass input to the output

INPUTS		OUTPUTS			
A1	A0	D0	D1	D2	D3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



tri State Buffer



E=1 (Activate) Y=A

E=0 (disabled)

E	A	Y
1	0	0
1	1	1
0	X	X

} Input reach to output

Input will not reaches to the output

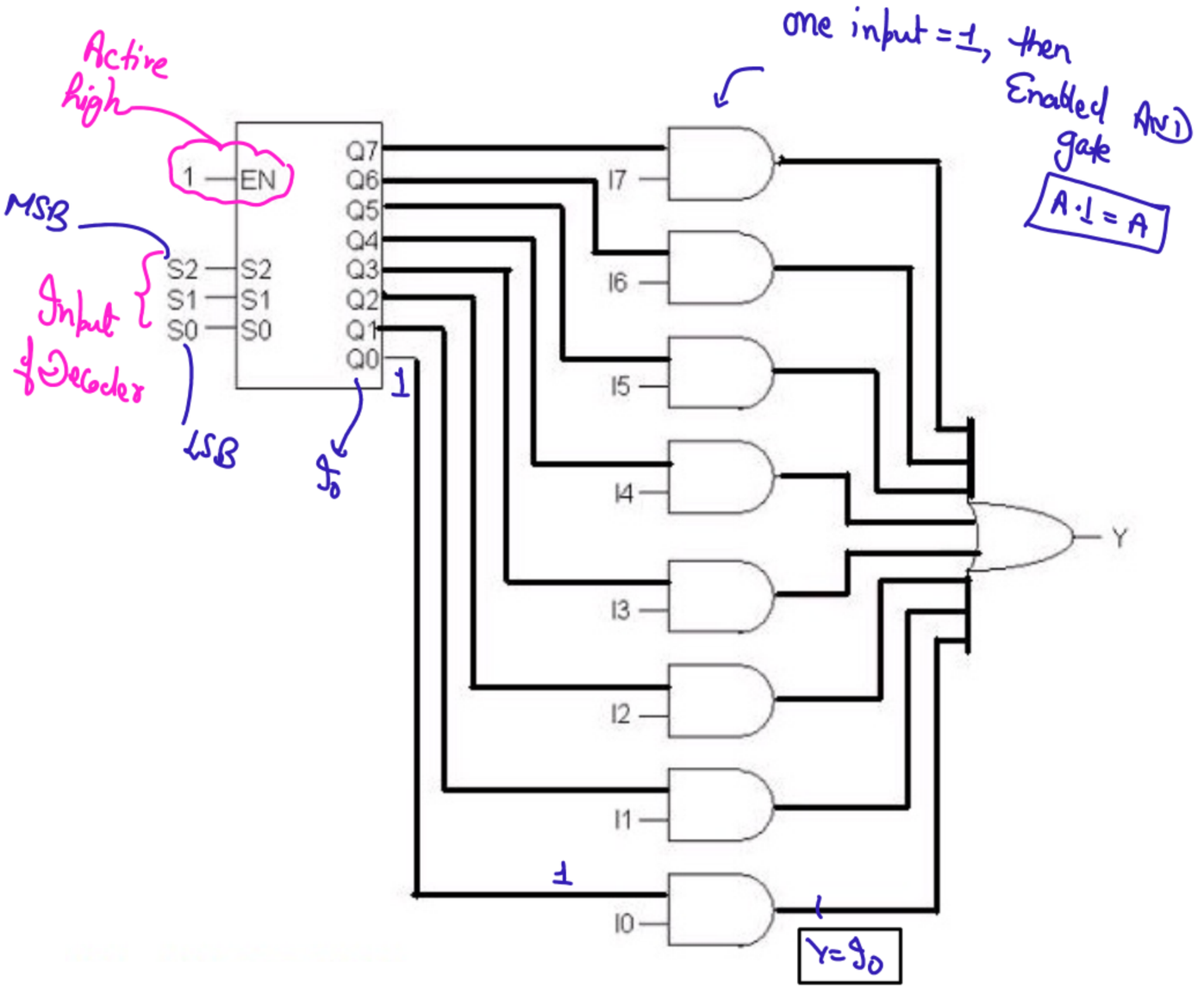
Input will not reach to the output

(Undefined State) [or High-z State] } Output is NOT zero

Implementation 8:1 MUX using 3:8 Decoder

Truth table for 8:1 MUX

Enable	Select Inputs			Output
E	S2	S1	S0	Y
0	X	X	X	0
1	0	0	0	I_0
1	0	0	1	I_1
1	0	1	0	I_2
1	0	1	1	I_3
1	1	0	0	I_4
1	1	0	1	I_5
1	1	1	0	I_6
1	1	1	1	I_7



Implementation of high order Decoder to Low order Decoder

High order decoder to low order decoder

If we have ~~x: 2^x~~ decoder that has to be designed by ~~2^y~~ decoder, then

$\frac{2^y}{2^x} = K1$ → n^{th} Stage → last stage

$\frac{K1}{2^x} = K2$ → $n\text{-}1^{th}$ Stage

.

.

Till we get 1 or Fraction ← Point (MSB at first Stage)

For Fraction we use NOT gate at first stage to Enable the corresponding decoder

Implement 3:8 Decoder using 2:4 Decoder

$y:2^y$ $\quad x:2^x$

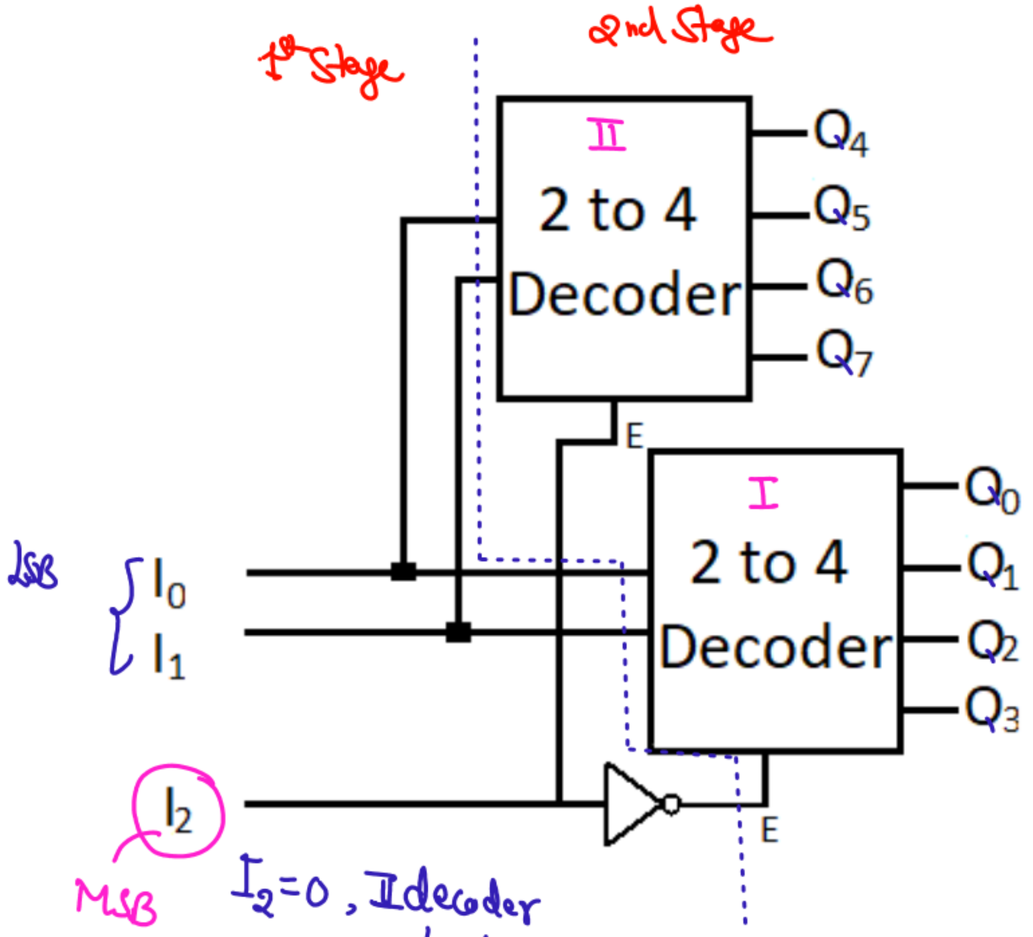
$\frac{2^y}{2^x} = \frac{8}{4} = 2 \leftarrow$ last stage

$2/4 = 0.5 \leftarrow$ fraction (Not gate at first stage

I_2	I_1	I_0		Y
0	0	0	m_0	Q_0
0	0	1	m_1	Q_1
0	1	0	:	Q_2
0	1	1	:	Q_3
1	0	0	:	Q_4
1	0	1		Q_5
1	1	0		Q_6
1	1	1	m_7	Q_7

first 4 output

$I_2=1$



$I_2=0$, II decoder ↳ disable

$I_2=1$, I decoder → disable

Implement 4:16 Decoder using 2:4 Decoder

High order — Low order

$16/4 = 4$ ← last stage (LSB)

$4/4 = 1$ (1st Stage) → 1 decoder → 2 inputs → MSB

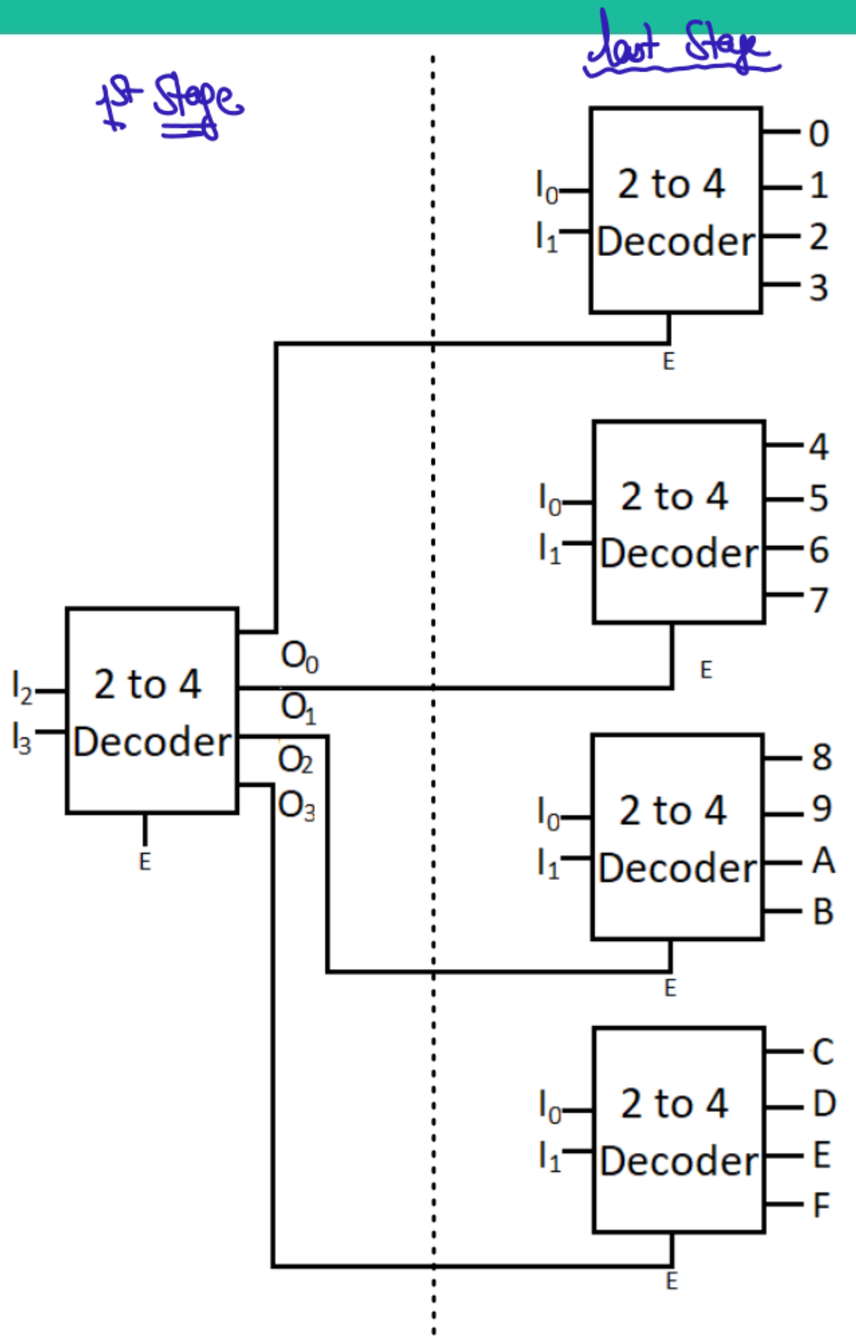
For input $I_3\ I_2\ I_1\ I_0$

I_3 I_2 → MSB → 1st Stage

I_1 I_0 → LSB → last Stage

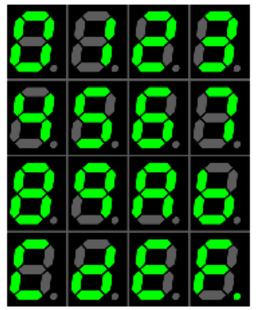
Make Truth Table

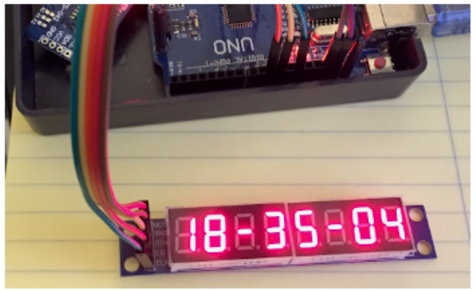
I_3	I_2	I_1	I_0
0	0	0	0
0	1		
1	0		
1	1		

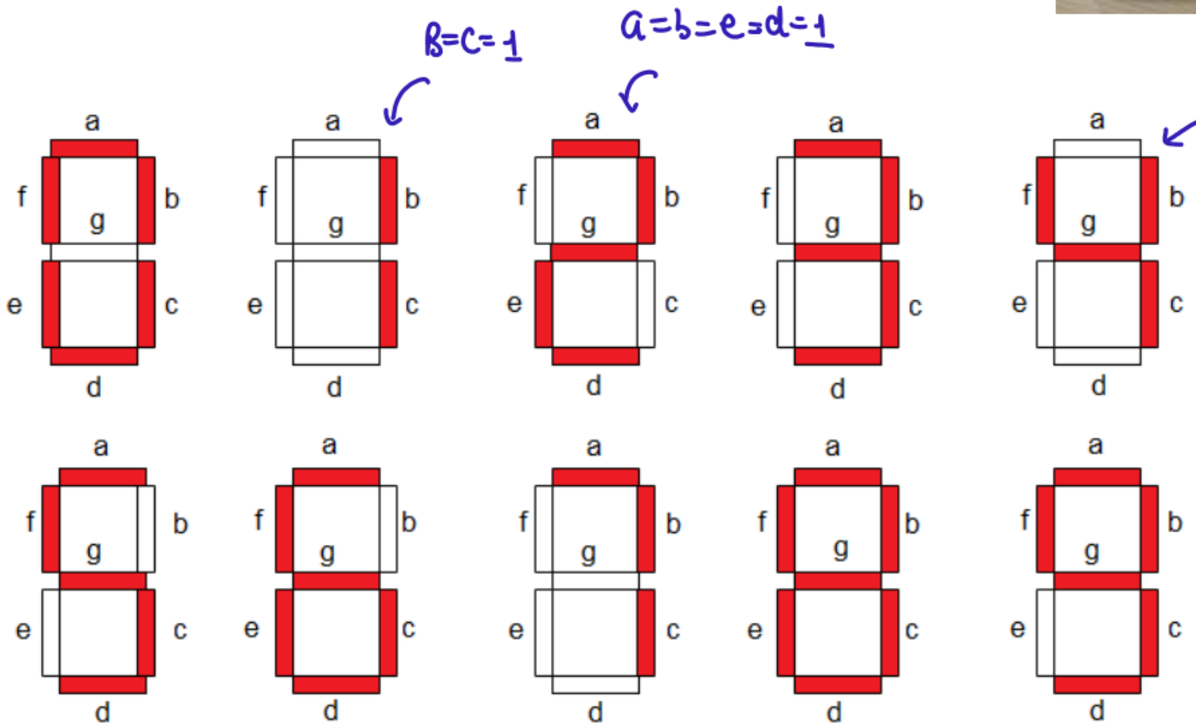


7 Segment Decoder

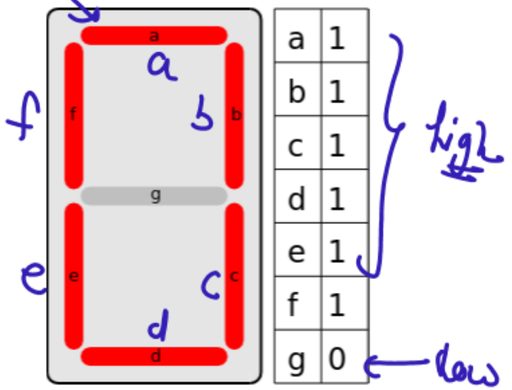
A 7-segment display is an electronic display device used to display decimal numerals and some alphabets. It consists of seven LEDs (segments) arranged in a way to display numbers from 0 to 9 and a few characters like A, b, C, d, E, F, etc.





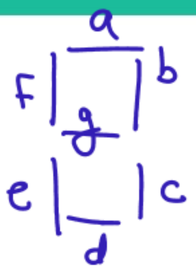


Example:



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

7 Segment Decoder



For each output, we have to design K-Map to get simplified expression.

* We need to K-Map for each LED to glow with a certain combinations of input

(Common for all)

$a = \sum m(0,2,3,5,6,7,8,9) + d(10,11,12,13,14,15)$

$b = \sum m(0,1,2,3,4,7,8,9)$

$c = \sum m(0,1,3,4,7,8,9)$

$d = \sum m(0,2,3,5,6,8,9)$

$e = \sum m(0,2,6,8)$

$f = \sum m(0,4,5,6,8,9)$

$g = \sum m(2,3,4,5,6,8,9)$

Decimal Digit	Input lines				Output lines							Display pattern
	A	B	C	D	a	b	c	d	e	f	g	
0	0	0	0	0	1	1	1	1	1	1	0	0
1	0	0	0	1	0	1	1	0	0	0	0	1
2	0	0	1	0	1	1	0	1	1	0	1	2
3	0	0	1	1	1	1	1	1	0	0	1	3
4	0	1	0	0	0	1	1	0	0	1	1	4
5	0	1	0	1	1	0	1	1	0	1	1	5
6	0	1	1	0	1	0	1	1	1	1	1	6
7	0	1	1	1	1	1	1	0	0	0	0	7
8	1	0	0	0	1	1	1	1	1	1	1	8
9	1	0	0	1	1	1	1	1	0	1	1	9

7 Segment Decoder

a

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

$a = A + C + BD + \overline{B}\,\overline{D}$

b

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

$b = \overline{B} + \overline{C}\,\overline{D} + CD$

c

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

$c = B + \overline{C} + D$

d

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

$d = A + \overline{B}\,\overline{D} + \overline{B}C + C\overline{D} + B\overline{C}D$

e

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

$e = \overline{B}\,\overline{D} + C\overline{D}$

f

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

$f = A + B\overline{C} + B\overline{D} + \overline{C}\,\overline{D}$

g

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

$g = A + B\overline{C} + \overline{B}C + C\overline{D}$

7 Segment Decoder - Circuit Diagram