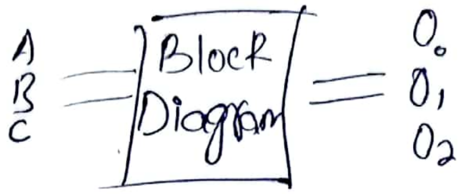
Combinational * transforming binary information from a given input data to a required O/P data.

(i) specify # of I/P & # of outputs



(ii) Truth Table 2^n, where n is number of inputs

BCD – Excess-3, Adder, Subtractor

(iii) K-map

(iv) Simplified Expression or Boolean Algebra

Decoder digital circuit that detects presence of code on its input and indicates the presence of that codes by specified level of inputs.

n to 2^n

1 →	2^1
2 →	2^2
3 →	2^3
4 →	2^4
5 →	2^5
6 →	2^6 ...

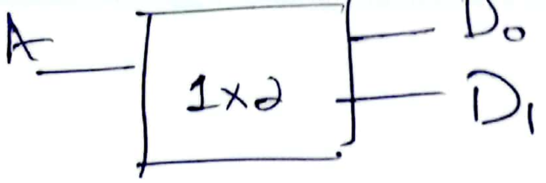
Full Adder

$S = \Sigma(1,2,4,7)$

$C = \Sigma(3,5,6,7)$

1 x 2

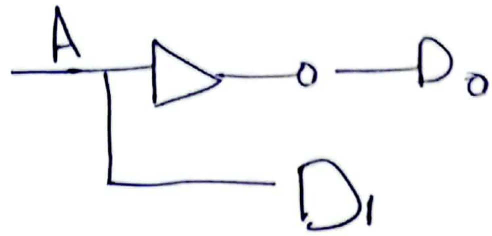
inputs — outputs



A	D_0	D_1
0	1	0
1	0	1

$D_0 = \bar{A}$

$D_1 = A$



D_0 shows presence of 0 at input

D_1 shows " " 1 " "

2 x 4

input — outputs

dec	A	B	D_3	D_2	D_1	D_0
0	0	0	0	0	0	1
1	0	1	0	0	1	0
2	1	0	0	1	0	0
3	1	1	1	0	0	0

$D_0 = \bar{A}\bar{B}$

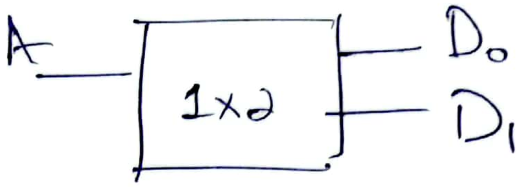
$D_1 = \bar{A}B$

$D_2 = A\bar{B}$

$D_3 = AB$

1×2

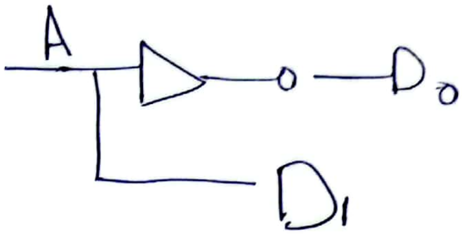
inputs → 1, outputs → 2



A	D_0	D_1
0	1	0
1	0	1

$D_0 = \overline{A}$

$D_1 = A$



D_0 shows presence of 0 at input

D_1 shows " " 1 " "

2×4

input → 2, outputs → 4

dec	A	B	D_3	D_2	D_1	D_0
0	0	0	0	0	0	1
1	0	1	0	0	1	0
2	1	0	0	1	0	0
3	1	1	1	0	0	0

$D_0 = \overline{A}\,\overline{B}$

$D_1 = \overline{A}B$

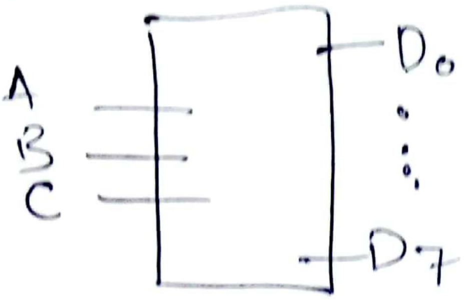
$D_2 = A\overline{B}$

$D_3 = AB$

3 X 8

Octal number

③



dec	A	B	C	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	0	1	0	0	0	0	0	0	0
1	0	0	1	0	1	0	0	0	0	0	0
2	0	1	0	0	0	1	0	0	0	0	0
3	0	1	1	0	0	0	1	0	0	0	0
4	1	0	0	0	0	0	0	1	0	0	0
5	1	0	1	0	0	0	0	0	1	0	0
6	1	1	0	0	0	0	0	0	0	1	0
7	1	1	1	0	0	0	0	0	0	0	1

$D_0 = \bar{A}\bar{B}\bar{C}$

$D_1 = \bar{A}\bar{B}C$

$D_2 = \bar{A}B\bar{C}$

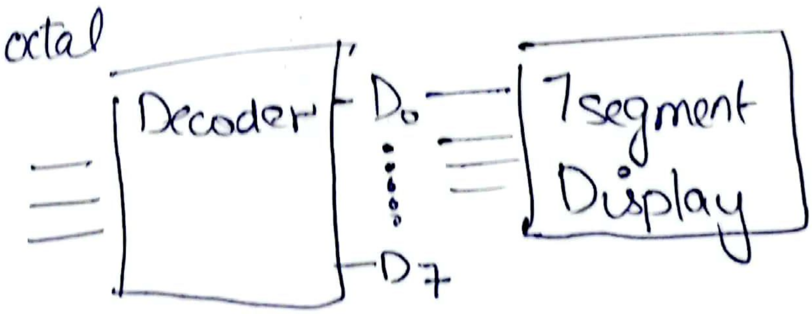
$D_3 = \bar{A}BC$

$D_4 = A\bar{B}\bar{C}$

$D_5 = A\bar{B}C$

$D_6 = AB\bar{C}$

$D_7 = ABC$

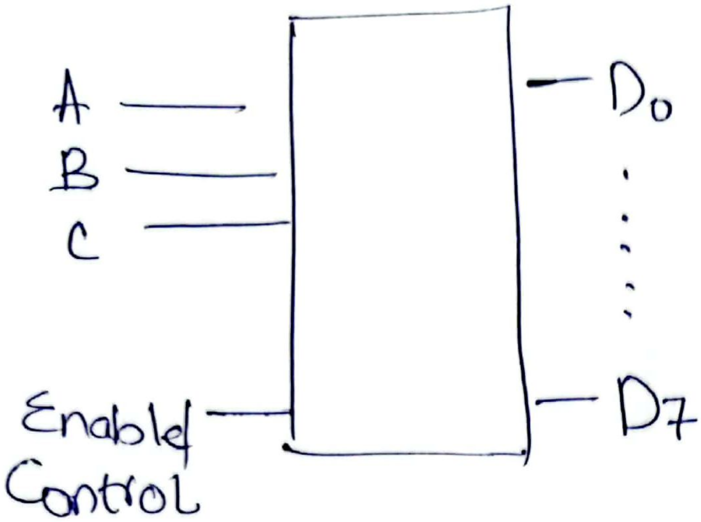


— Accessing a memory.

— Buzzer

— Activating a circuit [selection]

— mouse
— Touchpad
— gamepad



3x8 decoder with enable

E	A	B	C	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	X	X	X	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	0	0	0	0
	⋮										
1	1	1	1	0	0	0	0	0	0	0	1

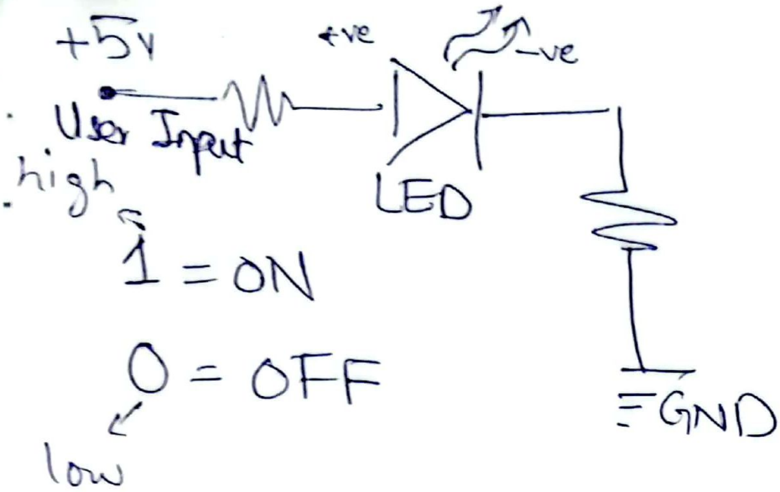
How to implement circuit with enable

EA \ BC	00	01	11	10
00	0	0	0	0
01	0	0	0	0
11	0	0	0	0
10	1	0	0	0

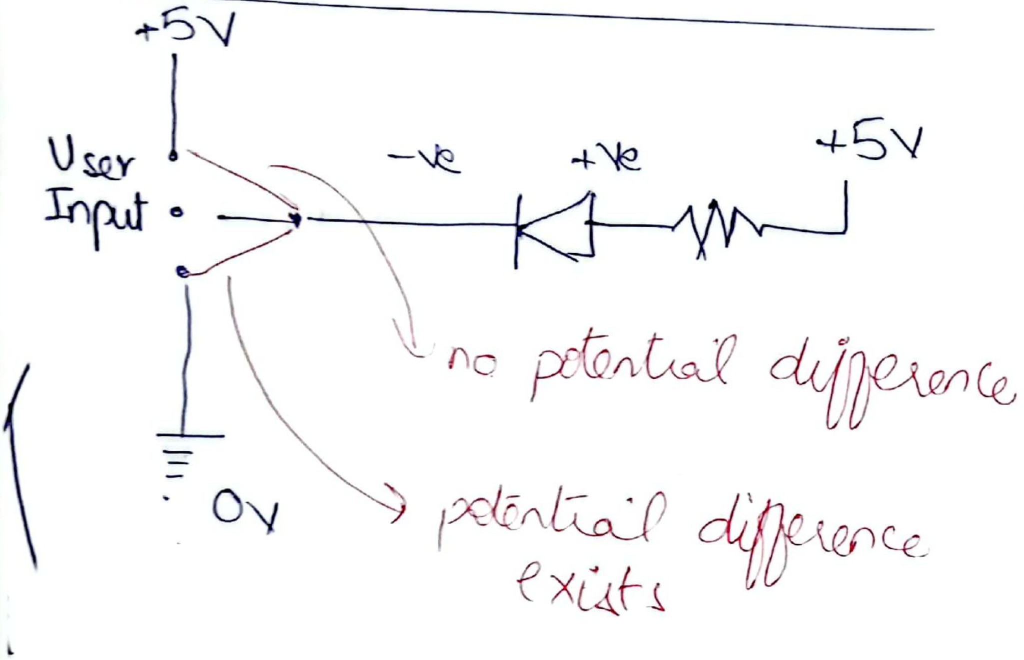
$D_0 = E\bar{A}\bar{B}\bar{C}$

Just add enable

Active high enable

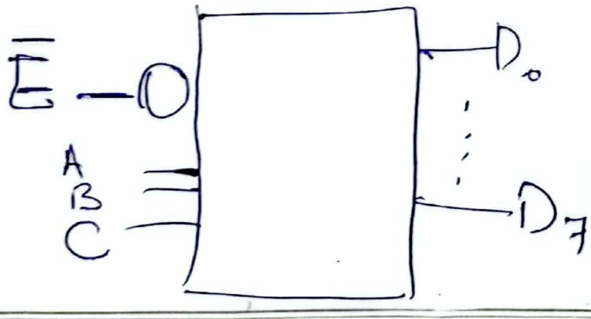


Active High Enable

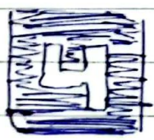


0 = ON
1 = OFF

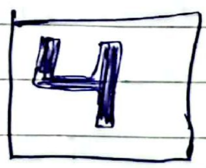
Active low enable



Bubble represents active low input



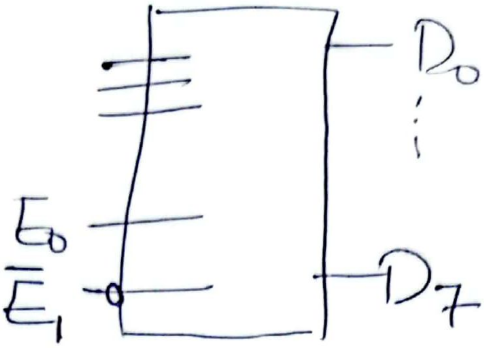
Active High output



Active Low output

- multiple enables
- active low output

E_0 $\overline{E}_1$



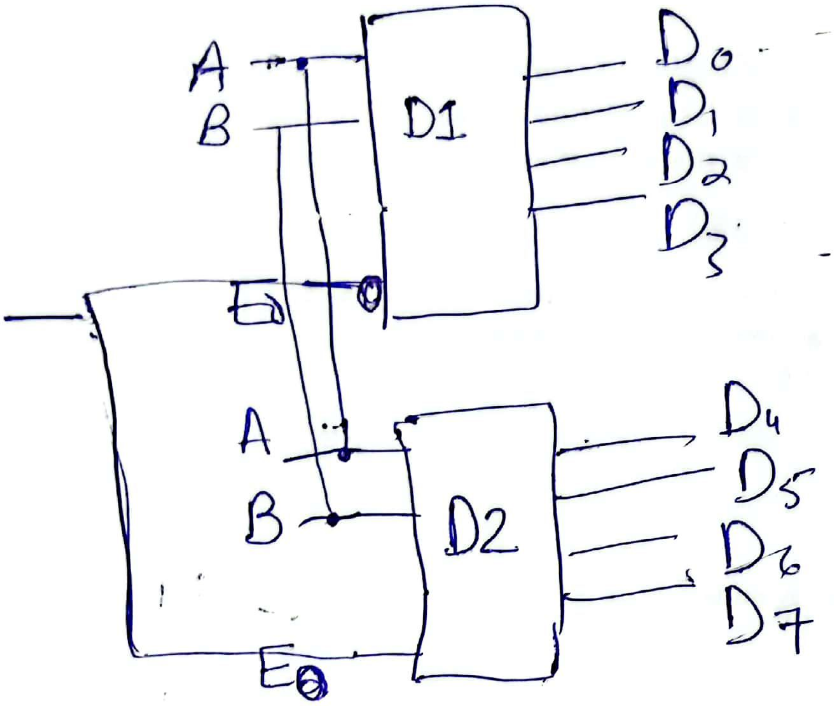
→ add another E_2

→ what if the outputs are also active low

E_2	$\overline{E}_1$	E_0	A	B	C	$\overline{D}_0$	$\overline{D}_1$	$\overline{D}_2$	$\overline{D}_3$	$\overline{D}_4$	$\overline{D}_5$	$\overline{D}_6$	$\overline{D}_7$
0	0	0	X	X	X	all zeros/ones							
0	0	1	X	X	X								
0	1	0	X	X	X								
0	1	1	X	X	X								
1	0	1	...										
1	1	0	X	X	X								
1	1	1	X	X	X								
1	0	1	0	0	0	0	1	1	1	1	1	1	1
1	0	1	0	0	1	1	0						
1	0	1	0	1	0	1		0		all ones			1
1	0	1	0	1	1	1			0				1
1	0	1	1	0	0	1				0			1
1	0	1	1	0	1	1	all				0		1
1	0	1	1	1	0	1		ones				0	1
1	0	1	1	1	1	1	1	1	1	1	1	1	0

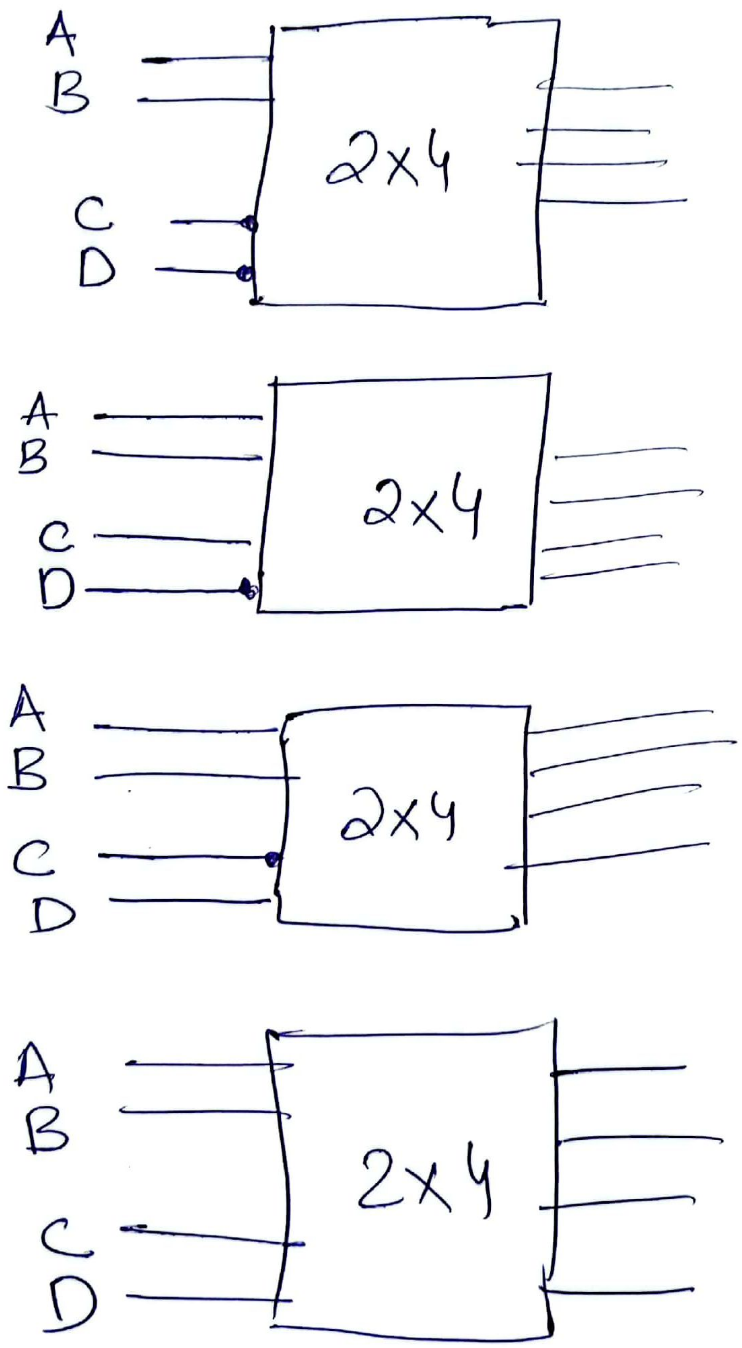
$F(A, B, C) = (0, 2, 3, 4)$

How to use 2x4 decoder to make 3x8 decoder (one enable)



	C	B	A
D1 enable	0	0	0
	0	0	1
	0	1	0
	0	1	1
D2 enable	1	0	0
	1	0	1
	1	1	0
	1	1	1

Design 4x16 decoder using 2x4 decoders (using two enable)

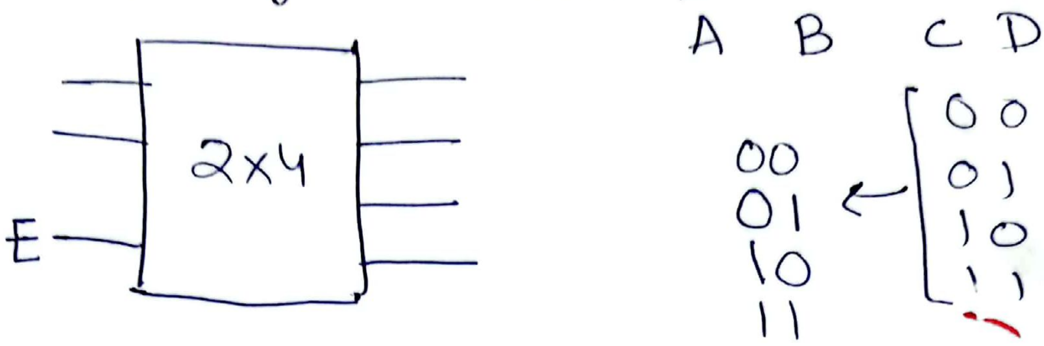


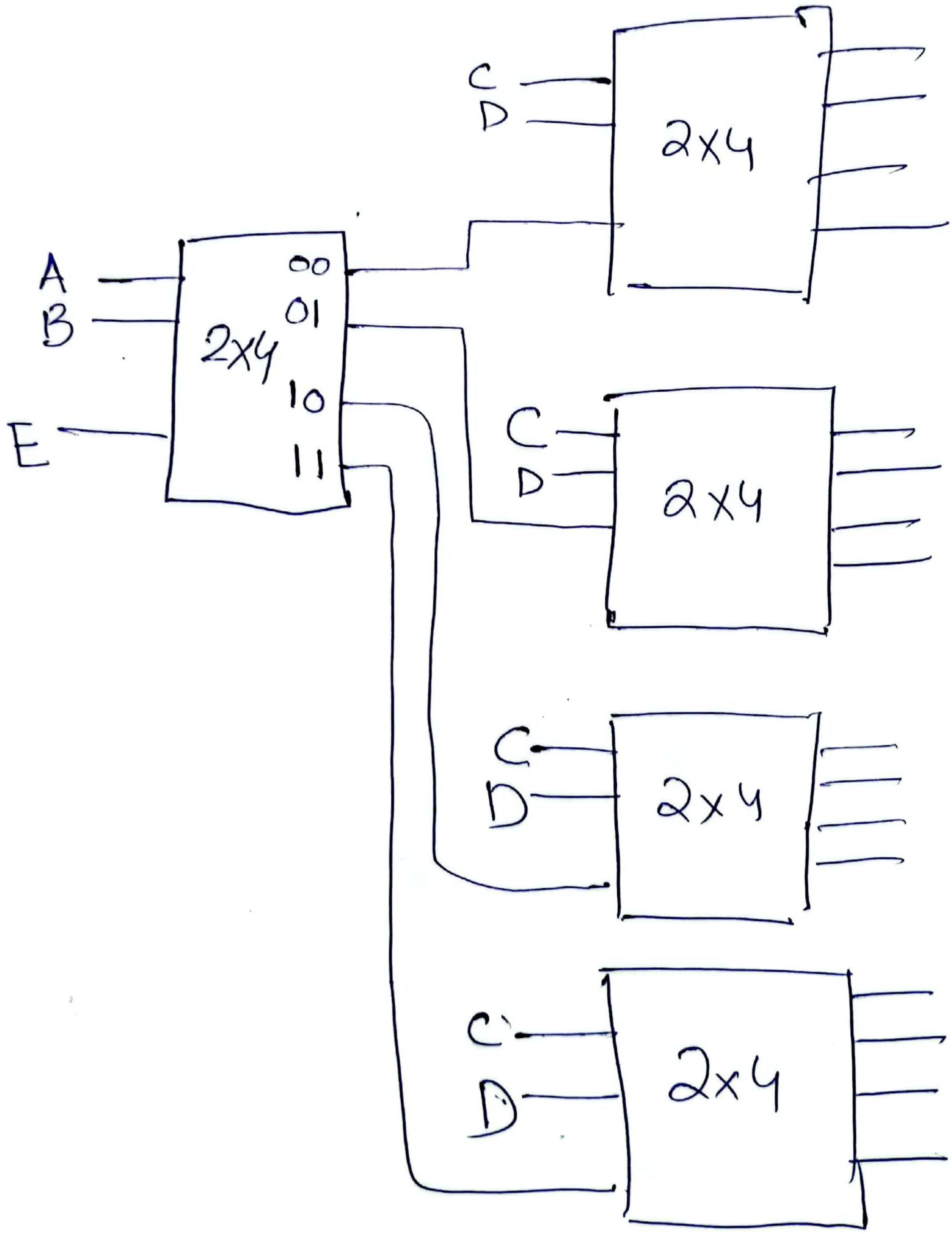
D	C	B	A
0	0		
0	1		
1	0		
1	1		

C,D are the enable input

3

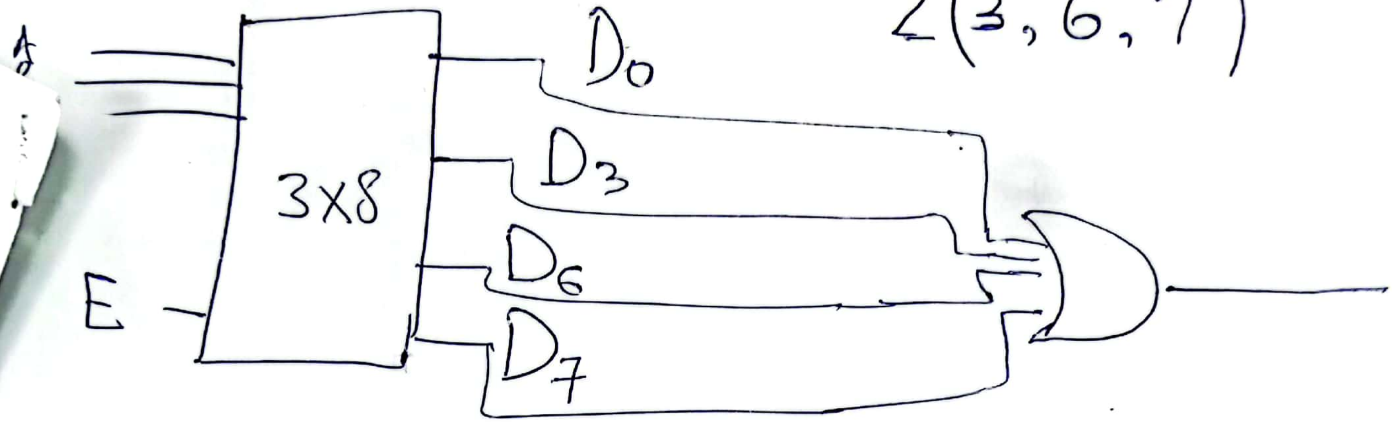
Design 4x16 decoder using 2x4 decoders (using one enable)





Using Decoder to implement SoP

$\Sigma(3, 6, 7)$



Using Decoder to implement PoS

$\pi(2, 3, 5, 7)$

$= \Sigma(1, 4, 6)$