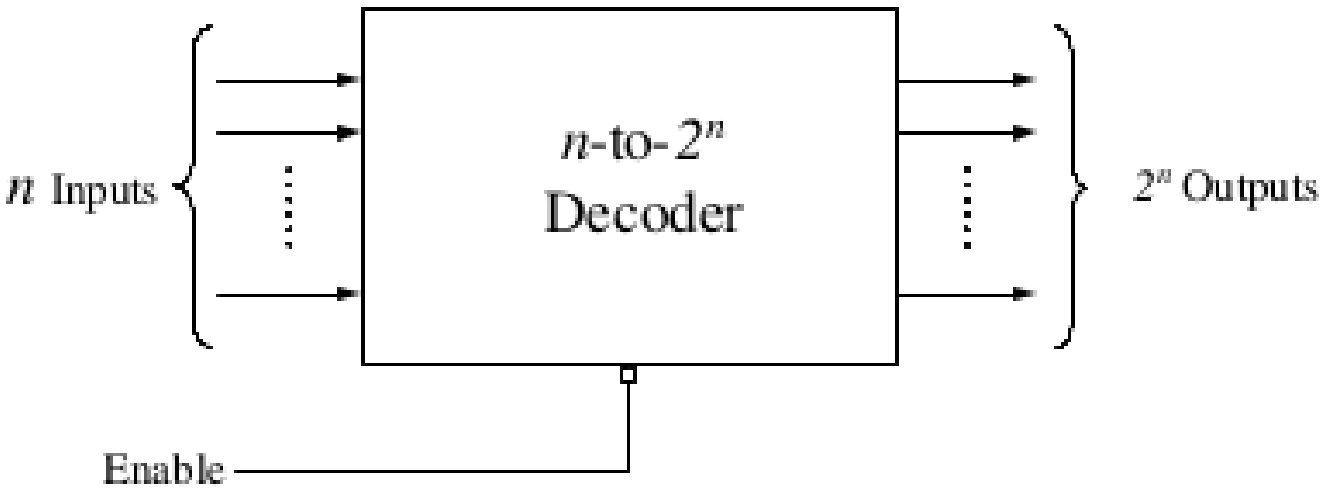
Decoder and Encoder

Theory

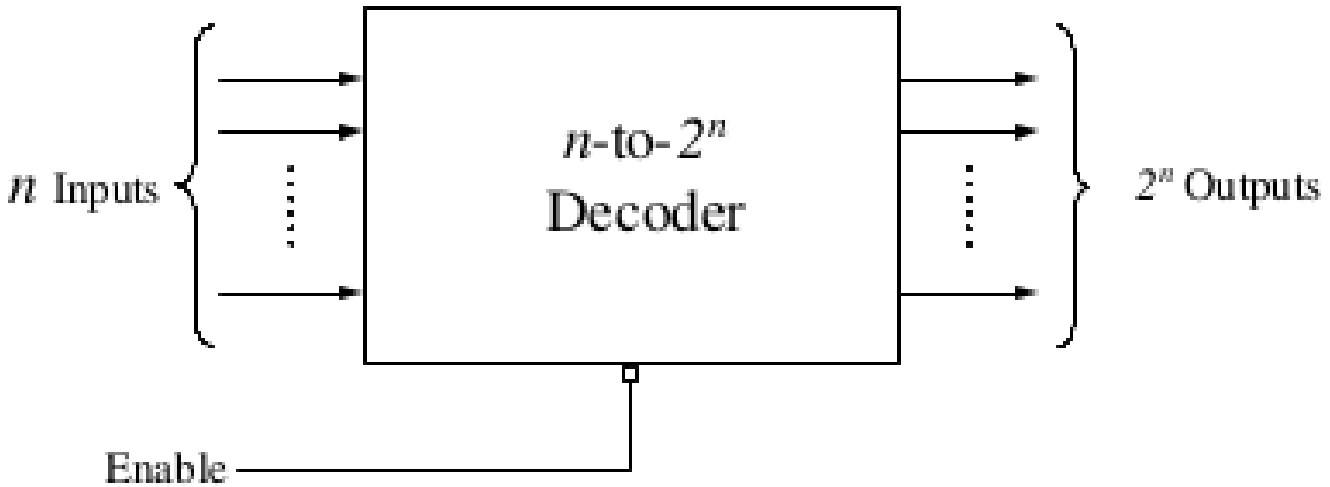
- Combinational Circuit
- *Decode*s the given input
- Input: n bits
- Output: 2^n bits

General Block Diagram – n to m decoder



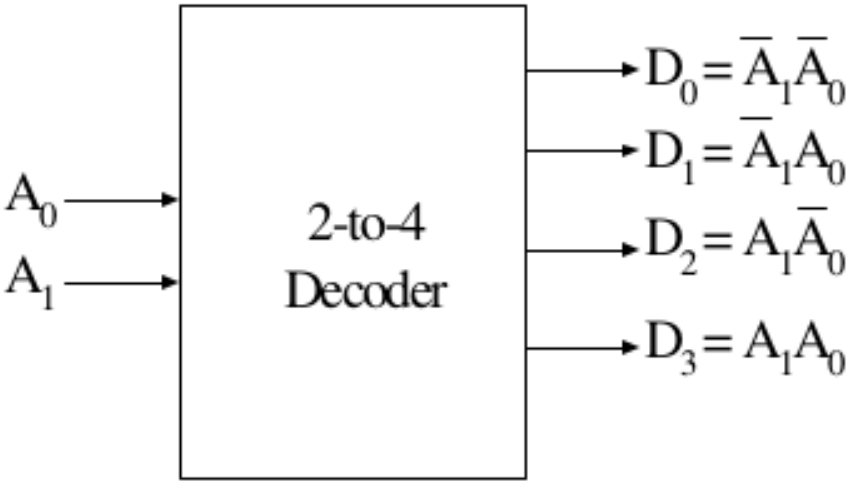
Each output corresponds to one of the 2^n possible combinations, $m=2^n$

General Block Diagram



- The ENABLE line performs no logical operation
- Used to toggle decoder between ACTIVE and INACTIVE states
  - When enable input is 0, all outputs are 0

2-to-4 line decoder



- 2 inputs A1(MSB) and A0(LSB),
- 2^2=4 outputs

  D0, D1, D2 and D3

Truth Table – without ENABLE

Decimal #	Input		Output			
	A_1	A_0	D_0	D_1	D_2	D_3
0	0	0	1	0	0	0
1	0	1	0	1	0	0
2	1	0	0	0	1	0
3	1	1	0	0	0	1

- For each input combination, one single output line is activated(=1)
- Each output is actually a MINTERM by taking a certain combination of the inputs

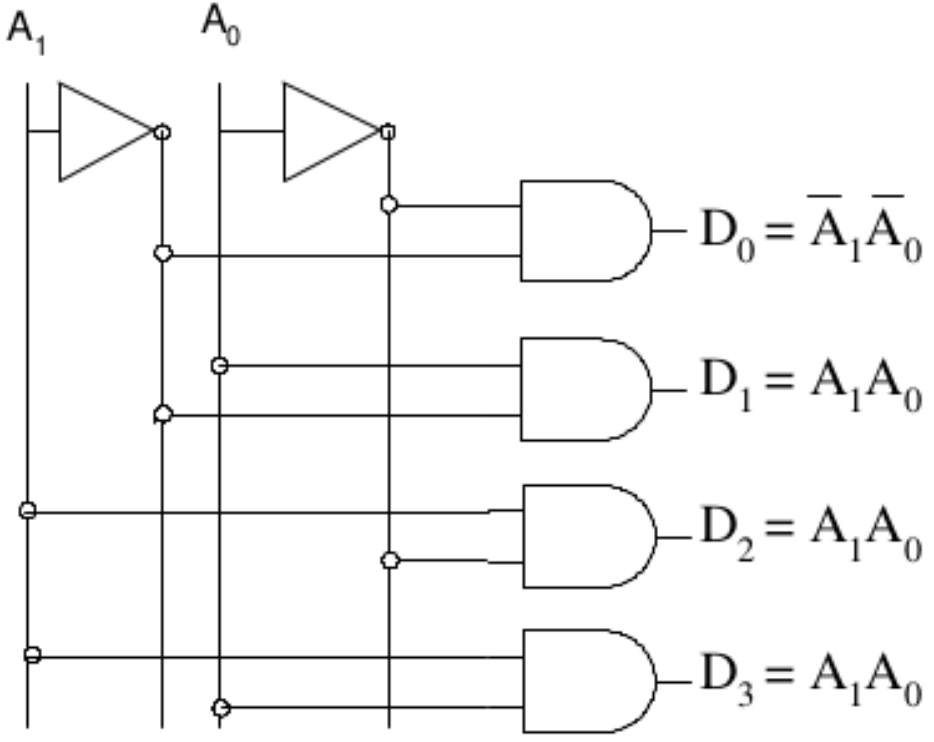
- D 0 =A 1’ A 0’ , (minterm m 0) which corresponds to input 00
- D 1 =A 1’ A 0 , (minterm m 1) which corresponds to input 01
- D 2 =A 1 A 0’ , (minterm m 2) which corresponds to input 10
- D 3 =A 1 A 0 , (minterm m 3) which corresponds to input 11

Truth Table – with ENABLE

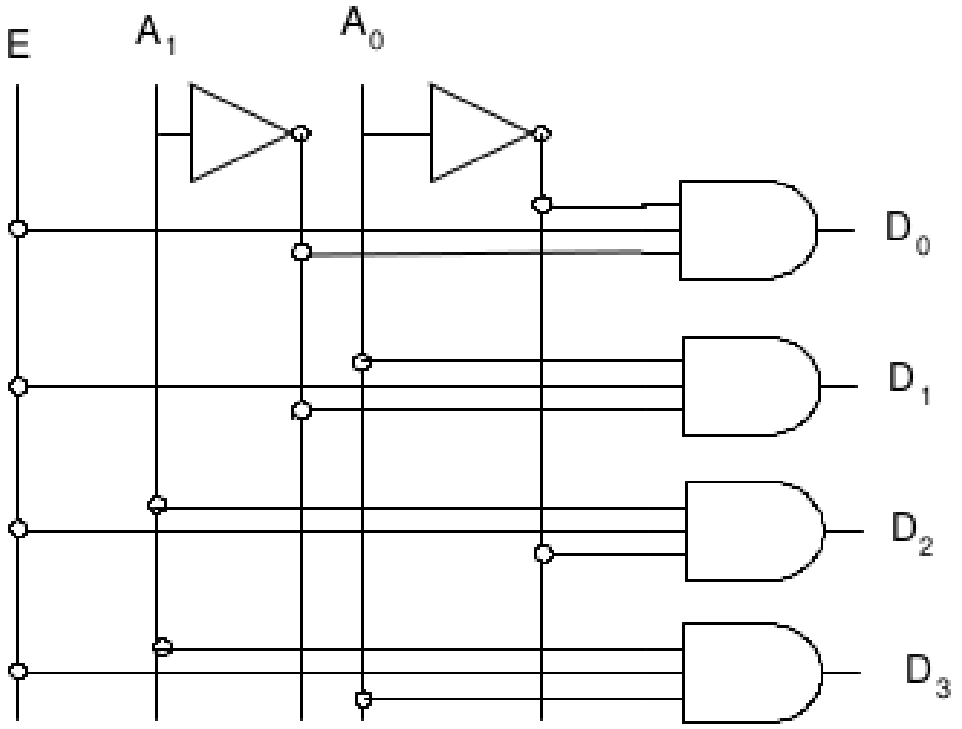
Decimal value	Enable	Inputs		Outputs			
	E	A_1	A_0	D_0	D_1	D_2	D_3
	0	X	X	0	0	0	0
0	1	0	0	1	0	0	0
1	1	0	1	0	1	0	0
2	1	1	0	0	0	1	0
3	1	1	1	0	0	0	1

- D 0 = E A 1' A 0' , (minterm m 0) which corresponds to input 00
- D 1 = E A 1' A 0 , (minterm m 1) which corresponds to input 01
- D 2 = E A 1 A 0' , (minterm m 2) which corresponds to input 10
- D 3 = E A 1 A 0 , (minterm m 3) which corresponds to input 11

Circuit Diagram – without ENABLE



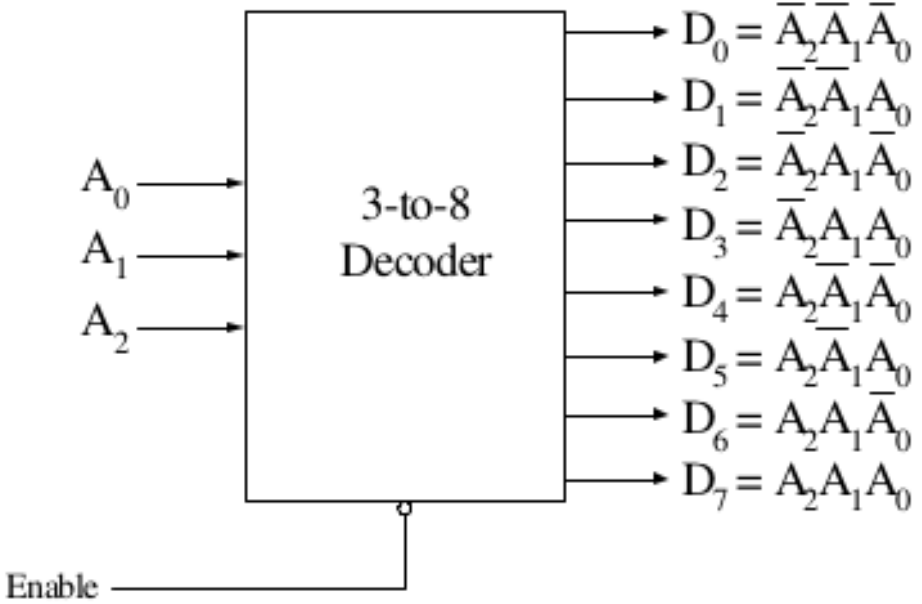
Circuit Diagram – with ENABLE



As long as E is 0,

the outputs remain 0

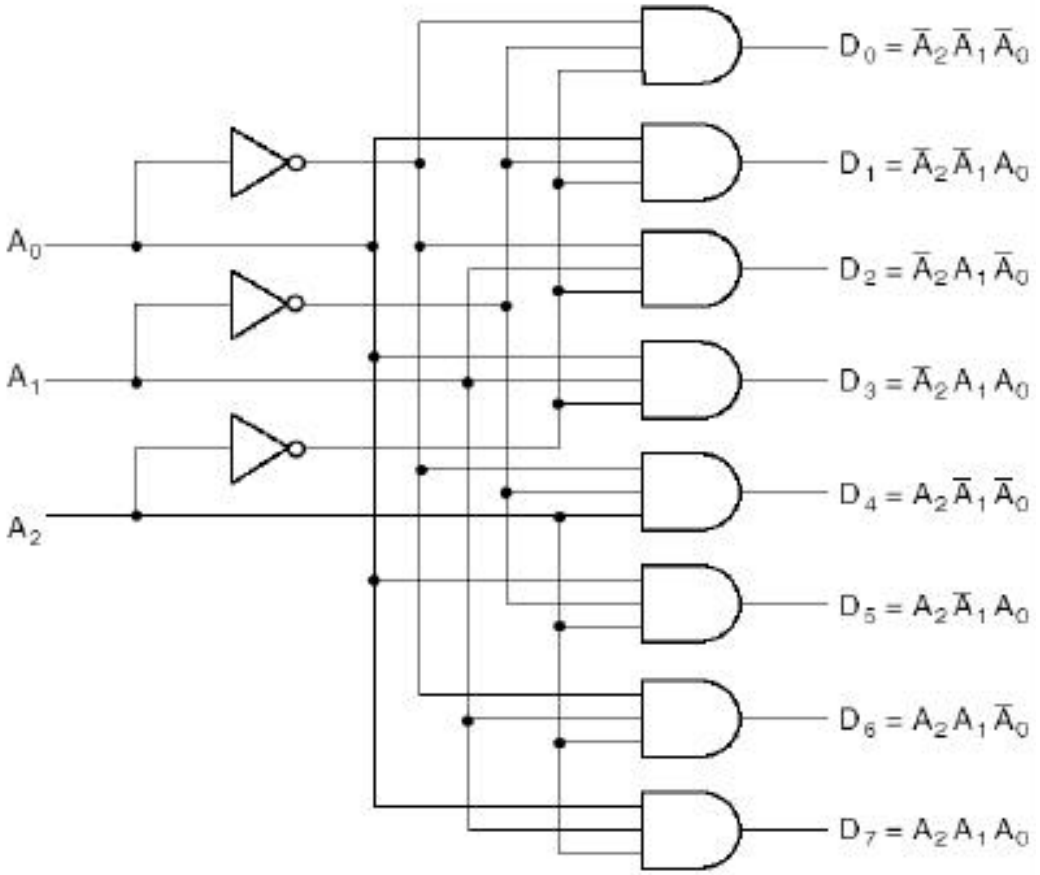
3 to 8 line Decoder



Truth Table

Dec. Code	Inputs			Outputs							
	A_2	A_1	A_0	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

Circuit Diagram



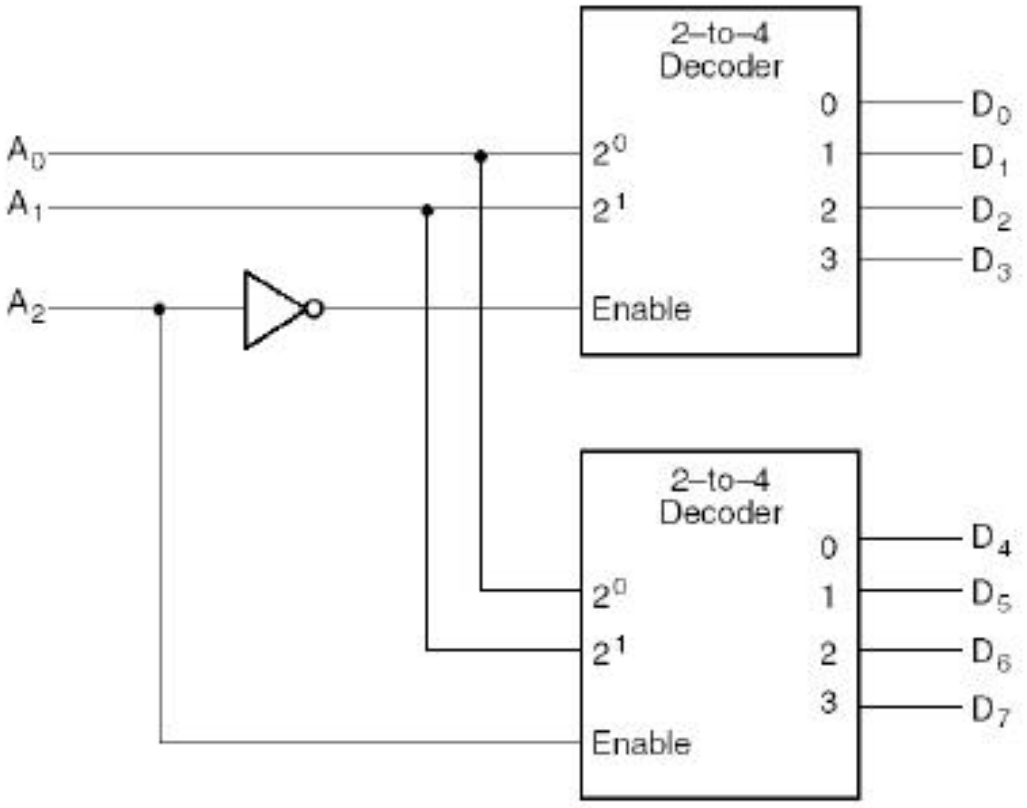
Construct a 3-to-8 decoder using two 2-to-4 decoders with enable inputs.

Steps

- The two least significant bits (i.e. A 1 and A 0) are connected to both decoders
- Most significant bit (A 2) is connected to the enable input of one decoder.
- The complement of most significant bit (A 2) is connected to the enable of the other decoder.

Steps

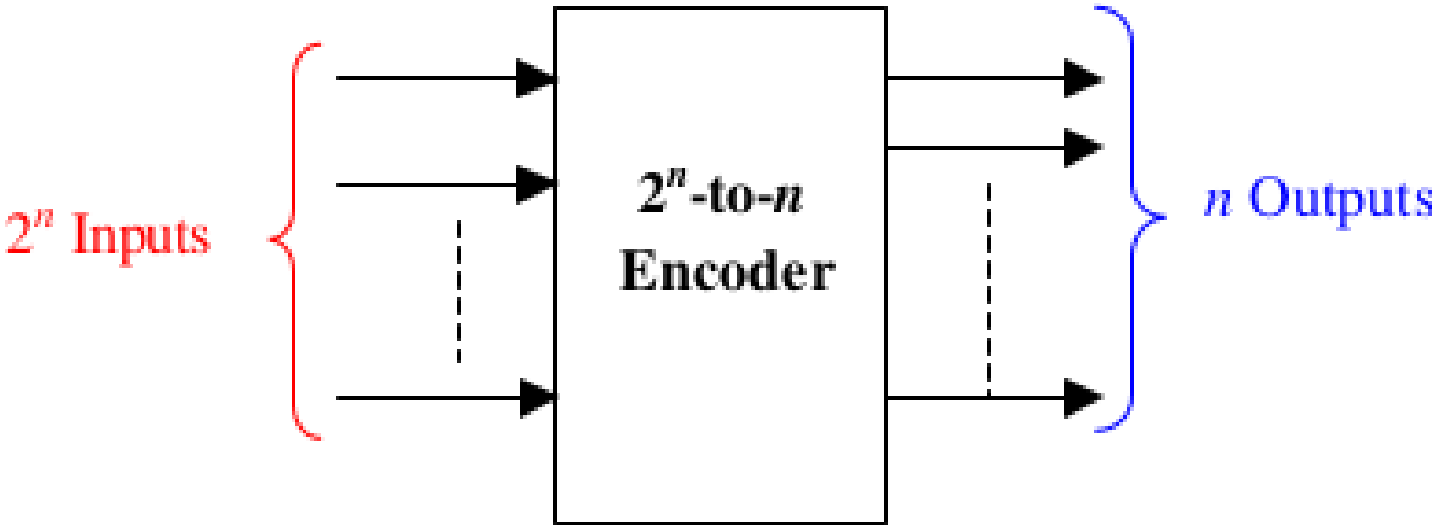
- When A 2 = 0, upper decoder is enabled, while the lower is disabled. Thus, the outputs of the upper decoder correspond to minterms D 0 through D 3.
- When A 2 = 1, upper decoder is disabled, while the lower is enabled. Thus, the outputs of the lower decoder correspond to minterms D 4 through D 7 .


2–to–4
Decoder
0
1
2
3
2^0
2^1
Enable
A_0
A_1
A_2
D_0
D_1
D_2
D_3
2–to–4
Decoder
0
1
2
3
2^0
2^1
Enable
D_4
D_5
D_6
D_7

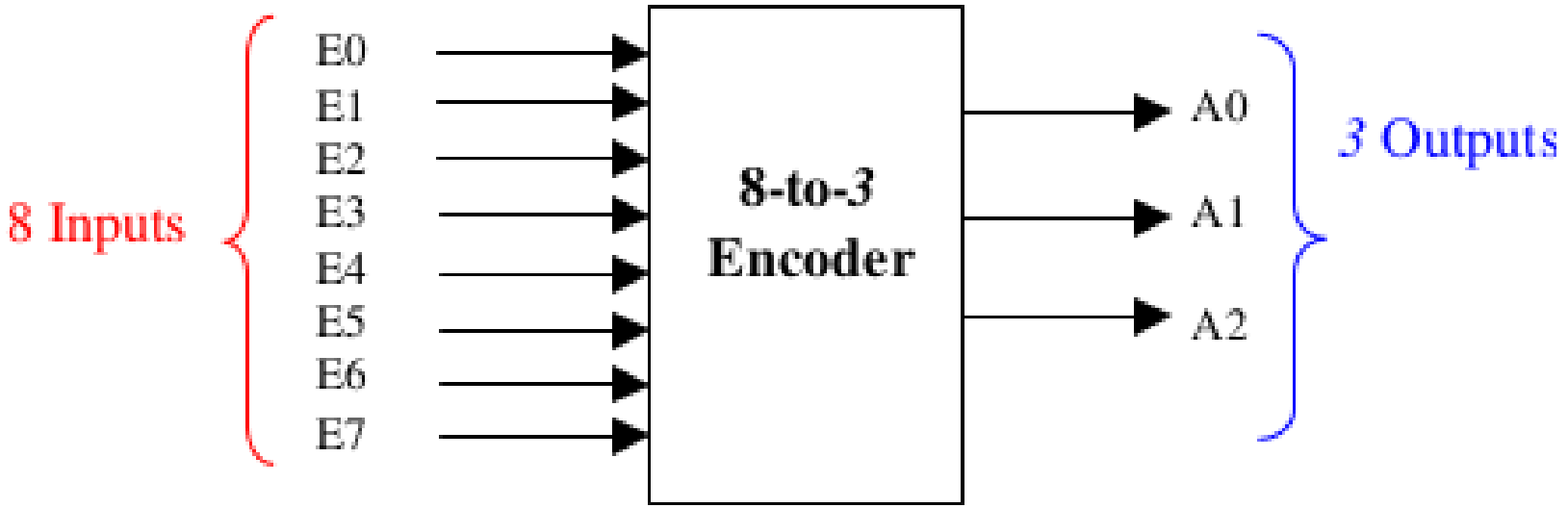
Encoder

- Performs reverse operation of Decoder
- 2^n inputs, n outputs
- At any time instant t, only one input can be high (logic 1)
- Output lines generate the binary code corresponding to the high input

General Block Diagram



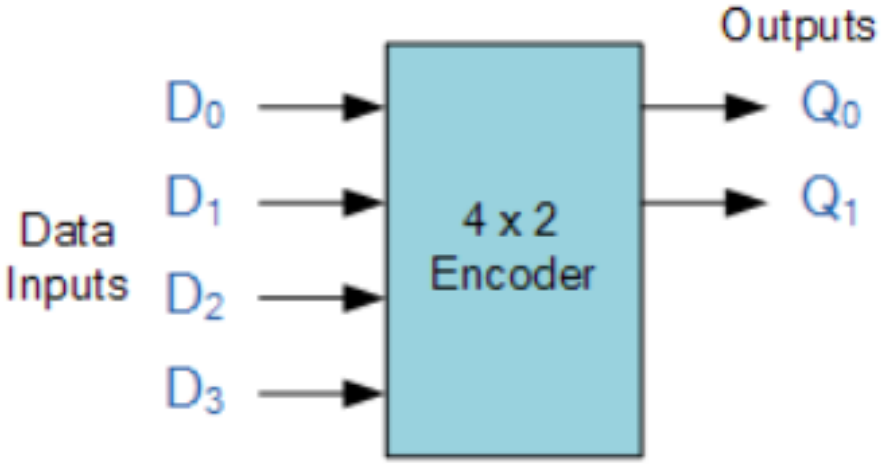
Block Diagram – 8 to 3 Encoder



Truth Table

Inputs								Outputs			Decimal Code
E7	E6	E5	E4	E3	E2	E1	E0	A2	A1	A0	
0	0	0	0	0	0	0	0	0	0	0	
0	0	0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	0	1	0	0	0	1	1
0	0	0	0	0	1	0	0	0	1	0	2
0	0	0	0	1	0	0	0	0	1	1	3
0	0	0	1	0	0	0	0	1	0	0	4
0	0	1	0	0	0	0	0	1	0	1	5
0	1	0	0	0	0	0	0	1	1	0	6
1	0	0	0	0	0	0	0	1	1	1	7

4 to 2 Encoder



Truth Table

Inputs				Outputs	
D_3	D_2	D_1	D_0	Q_1	Q_0
0	0	0	1	0	0
0	0	1	0	0	1
0	1	0	0	1	0
1	0	0	0	1	1
0	0	0	0	x	x

Q0 is high when D1 or D3 is high

Q0 = D1.D0'.D2'.D3' + D3.D2'.D1'.D0'

Q1 = D2.D1'.D3'.D0' + D3. D2'.D1'.D0'

Circuit Diagram

- Do it yourself !

Problems of Encoders

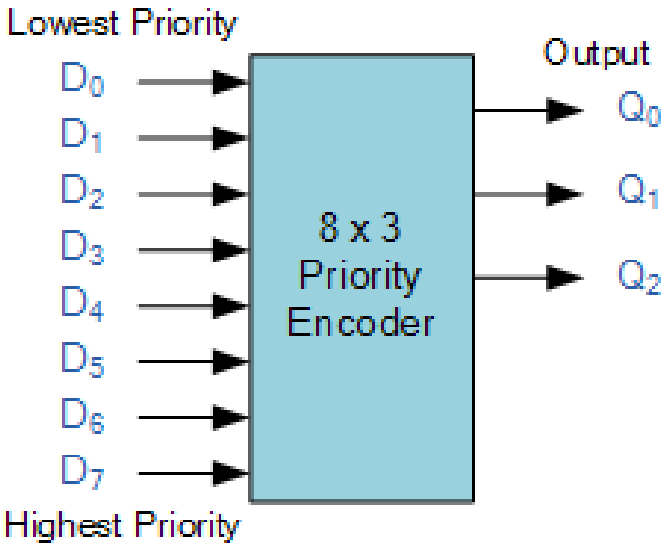
- If two or more inputs are active at the same time, what should the output be?
- An output of all 0's is generated in 2 cases:
  when all inputs are 0
  when E0 is equal to 1.

  How can this ambiguity be resolved?

PRIORITY ENCODER

- Uses pre-defined ‘priority’ value for each input
- If multiple inputs are 1, output will correspond to input pin with highest priority
- Digital Encoders that use this logic, are Priority Encoders
- In the presence of higher priority input, other lower priority inputs are ignored

Block Diagram



Truth Table

Digital Inputs								Binary Output		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Q_2	Q_1	Q_0
0	0	0	0	0	0	0	**1**	0	0	0
0	0	0	0	0	0	**1**	X	0	0	1
0	0	0	0	0	**1**	X	X	0	1	0
0	0	0	0	**1**	X	X	X	0	1	1
0	0	0	**1**	X	X	X	X	1	0	0
0	0	**1**	X	X	X	X	X	1	0	1
0	**1**	X	X	X	X	X	X	1	1	0
1	X	X	X	X	X	X	X	1	1	1

Expressions

Digital Inputs								Binary Output		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	Q_2	Q_1	Q_0
0	0	0	0	0	0	0	**1**	0	0	0
0	0	0	0	0	0	**1**	X	0	0	1
0	0	0	0	0	**1**	X	X	0	1	0
0	0	0	0	**1**	X	X	X	0	1	1
0	0	0	**1**	X	X	X	X	1	0	0
0	0	**1**	X	X	X	X	X	1	0	1
0	**1**	X	X	X	X	X	X	1	1	0
1	X	X	X	X	X	X	X	1	1	1

Q0 = D1 + D3 + D5 + D7

Q1 = D2 + D3 + D6 + D7

Q2 = D4 + D5 + D6 + D7

Priority Encoder