Digital Logic Design

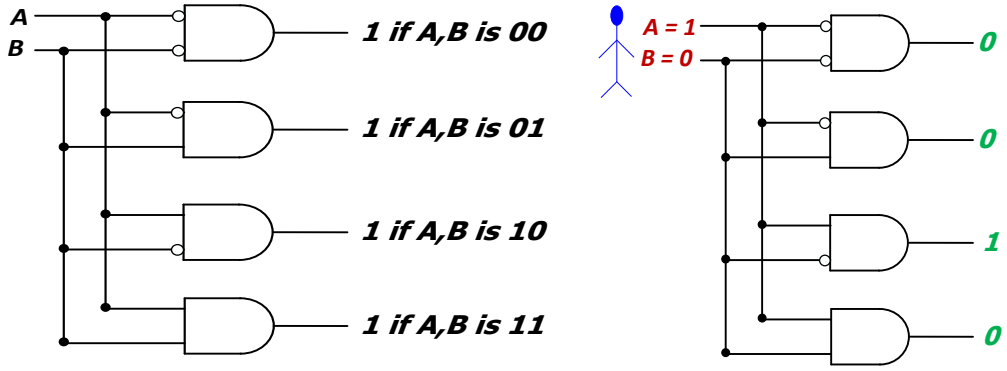
Hajar Falahati

Department of Computer Engineering
IRAN University of Science and Technology

hfalahati@iust.ac.ir

Decoder Implementation

- The one output that is **logically 1** is the output corresponding to the **input pattern** that the **logic circuit is expected to detect**
- Generate 2^n (or fewer) minterms for the *n* input variables



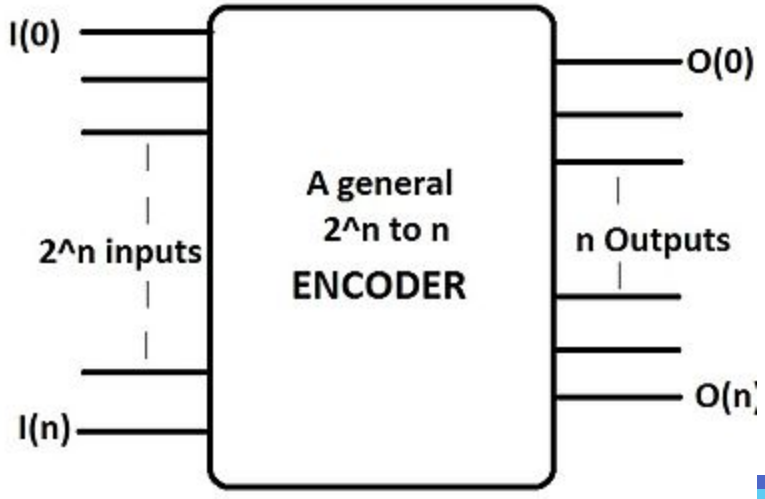
Outline

- Encoder
- Priority Encoder

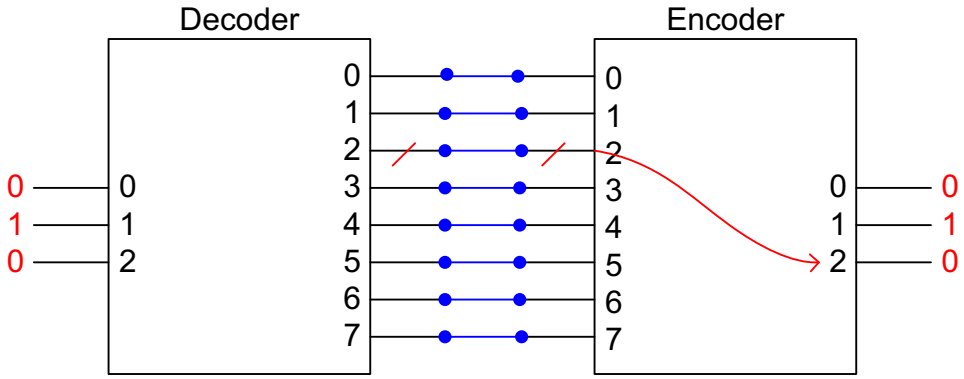
Encoder

Encoder

- Input
  - m-bit code; exactly one bit that is 1
- Output
  - n-bit; $n \leq m \leq 2^n$
- Function
  - Opposite of decoding
  - Generates the binary code corresponding to the input values

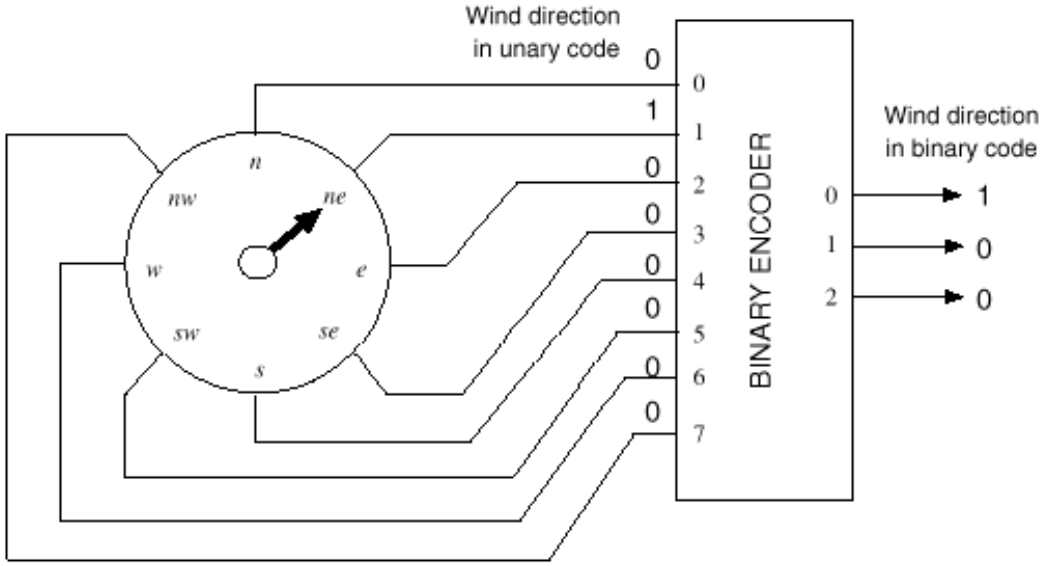


Encoder Vs. Decoder



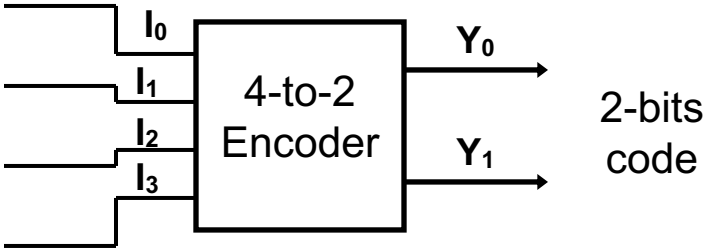
Encoder: Application

- A lot of input lines
- => Fewer input lines
- Example
  - Wind direction encoder



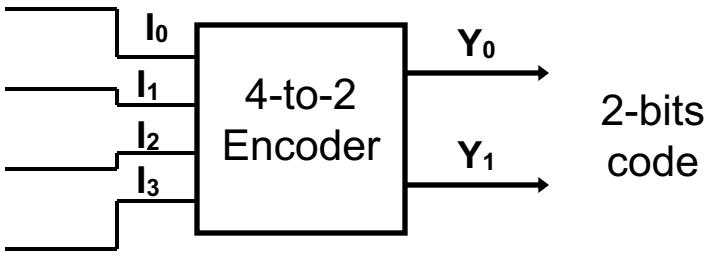
Encoder 4-2

- Input
  - 4 bit
- Output
  - 2 bit



Encoder 4-2: TT

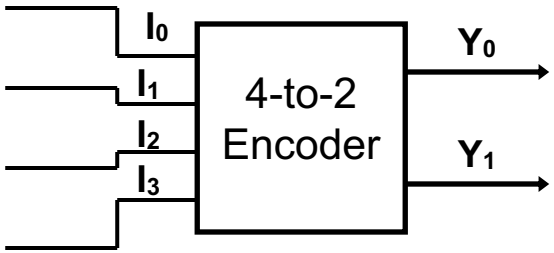
- Input
  - 4 bit
- Output
  - 2 bit



I_0	I_1	I_2	I_3	Y_1	Y_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1
0	0	0	0	X	X
0	0	1	1	X	X
0	1	0	1	X	X
0	1	1	0	X	X
0	1	1	1	X	X
1	0	0	1	X	X
1	0	1	0	X	X
1	0	1	1	X	X
1	1	0	0	X	X
1	1	0	1	X	X
1	1	1	0	X	X
1	1	1	1	X	X

Encoder 4-2: TT

- Input
  - 4 bit
- Output
  - 2 bit

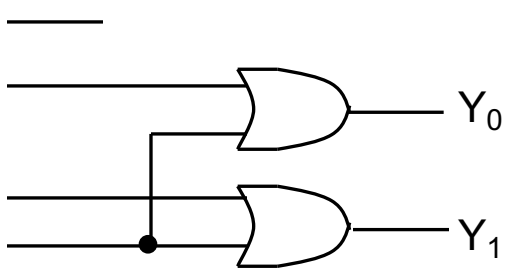


2-bits code

I_0	I_1	I_2	I_3	Y_1	Y_0
1	0	0	0	0	0
0	1	0	0	0	1
0	0	1	0	1	0
0	0	0	1	1	1
0	0	0	0	X	X
0	0	1	1	X	X
0	1	0	1	X	X
0	1	1	0	X	X
0	1	1	1	X	X
1	0	0	1	X	X
1	0	1	0	X	X
1	0	1	1	X	X
1	1	0	0	X	X
1	1	0	1	X	X
1	1	1	0	X	X
1	1	1	1	X	X

$Y_0 = I1 + I3$

$Y_1 = I2 + I3$



Encoder 8-3

- Octal to binary encoder
- Input
  - 8 bit
- Output
  - 3 bit

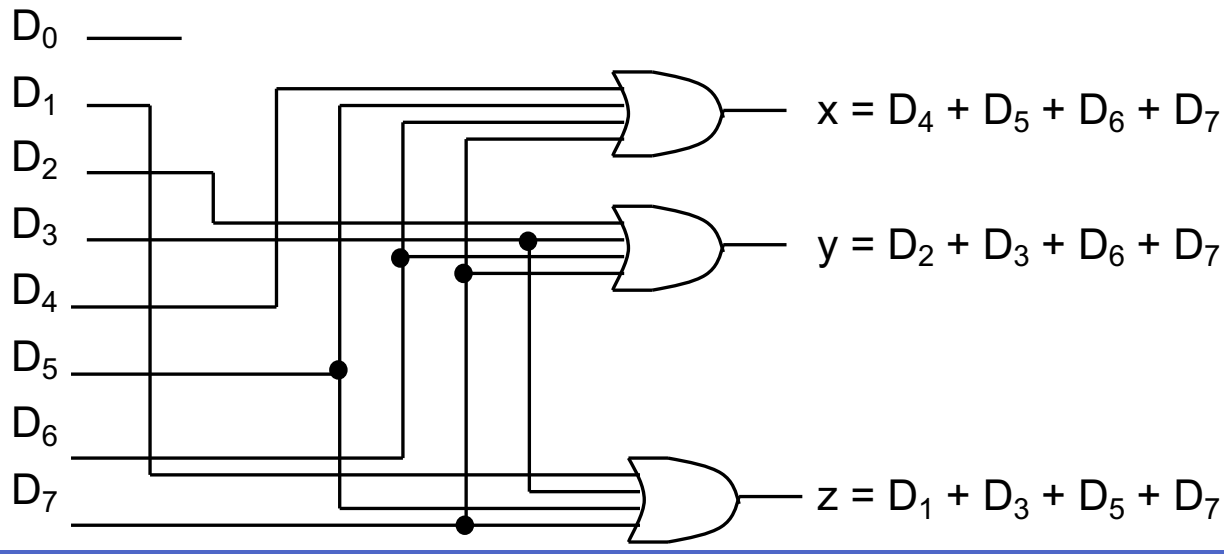
Encoder 8-3: Truth Table

- Octal to binary encoder
- Input
  - 8 bit
- Output
  - 3 bit

Inputs								Outputs		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1

Encoder 8-3: Logic

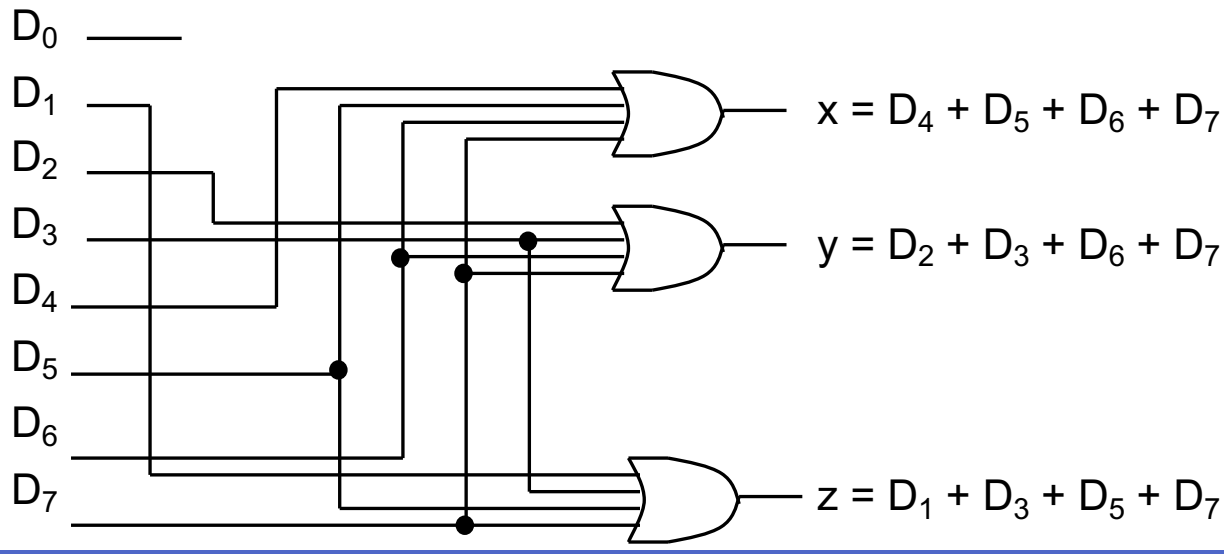
Inputs								Outputs		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1



Encoder: Issue

- At each time only one input can be active
- What happens if more than one input be active at the same time?

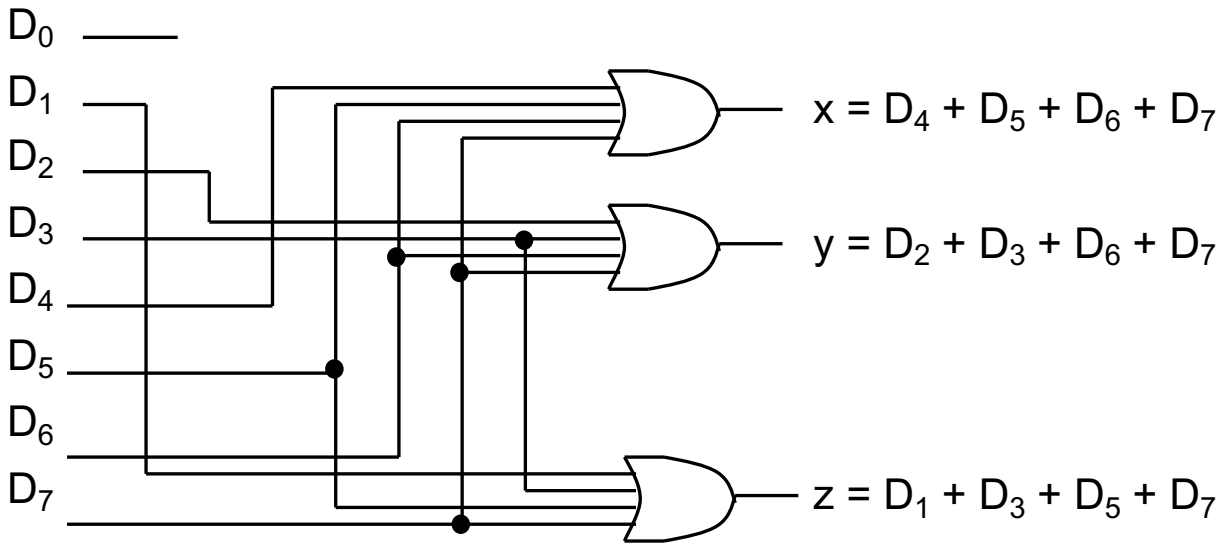
Inputs								Outputs		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1



Encoder: Issue (cont'd)

- At each time only one input can be active
- What happens if more than one input be active at the same time?
  - Output produces an undefined combination
- Example
  - D3 and D6 are 1 at the same time
  - Output: 111

Inputs								Outputs		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	0	0	0	1	0	0
0	0	0	0	0	1	0	0	1	0	1
0	0	0	0	0	0	1	0	1	1	0
0	0	0	0	0	0	0	1	1	1	1



Encoder: Issue: Solutions

1. If more than one input value is 1, encoder does not work.
2. Design an encoder which accepts all possible input combinations

Priority Encoder

- An encoder
- Accepts all possible input combinations
- Produces a meaningful result
- How?

Priority Encoder: Functionality

- Multiple asserted inputs are allowed
- One has priority over all others.
- Selects the most/ least significant input position containing a 1
- Produces the corresponding binary code for that position

Inputs				Outputs		
D_3	D_2	D_1	D_0	A_1	A_0	V
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

Priority Encoder: Sample1

- Design a 4-2 priority encoder

Sample1: Truth Table

- Design a 4-2 priority encoder

Inputs				Outputs		
D_3	D_2	D_1	D_0	A_1	A_0	V
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

Sample1: Simplification

Inputs				Outputs		
D_3	D_2	D_1	D_0	A_1	A_0	V
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

$$A_1 = D_3 + \overline{D_3} D_2$$

$$A_1 = D_3 + D_2$$

$$A_0 = D_3 + \overline{D_3}\,\overline{D_2} D_1$$

$$A_0 = D_3 + \overline{D_2} D_1$$

$$V = D_3 + \overline{D_3} D_2 + \overline{D_3}\,\overline{D_2} D_1 + \overline{D_3}\,\overline{D_2}\,\overline{D_1} D_0$$

$$V = D_3 + \overline{D_3}\,(D_2 + \overline{D_2} D_1 + \overline{D_2}\,\overline{D_1} D_0)$$

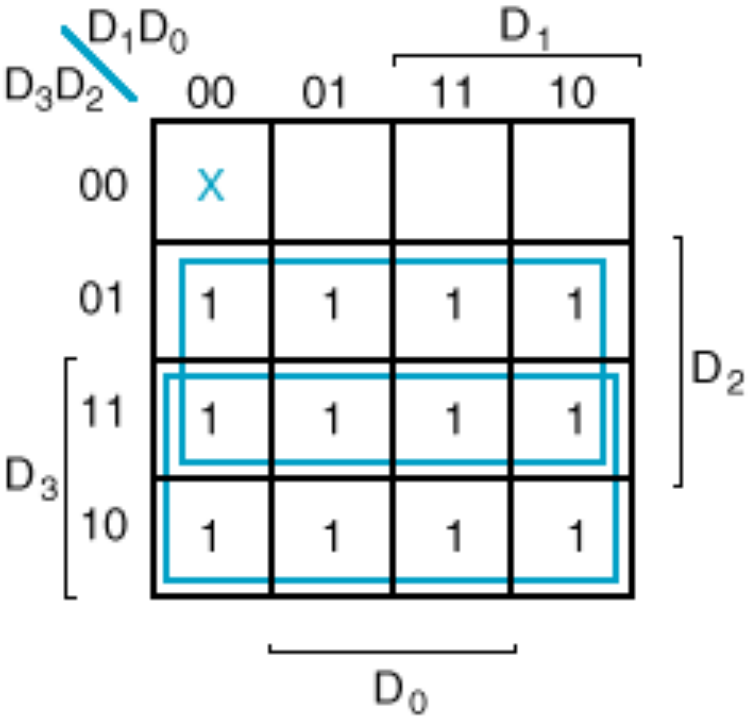
$$V = D_3 + D_2 + \overline{D_2} D_1 + \overline{D_2}\,\overline{D_1} D_0$$

$$V = D_3 + D_2 + \overline{D_2}\,(D_1 + \overline{D_1} D_0)$$

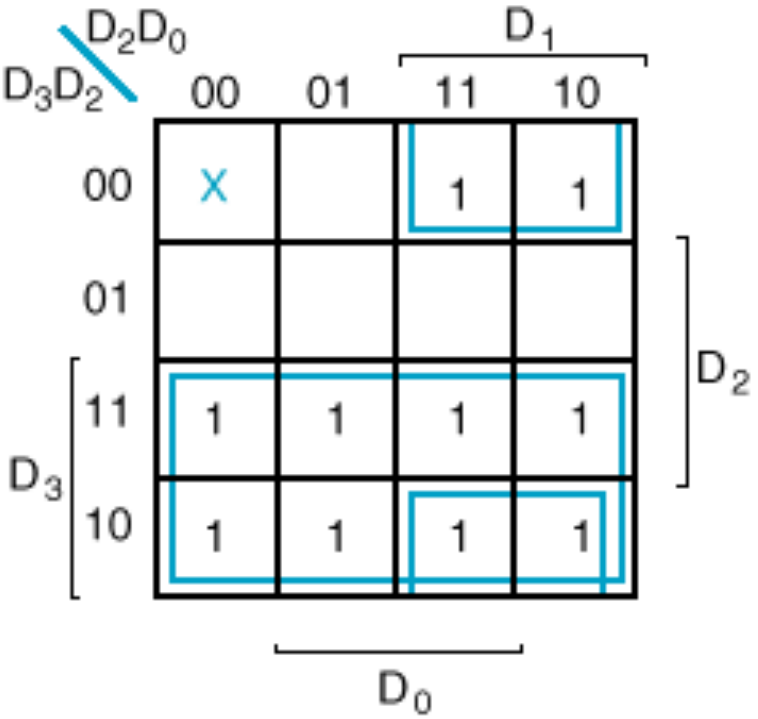
$$V = D_3 + D_2 + D_1 + \overline{D_1} D_0$$

$$V = D_3 + D_2 + D_1 + D_0$$

Priority EncoderK-Map



$A_1 = D_2 + D_3$



$A_0 = D_3 + D_1\overline{D}_2$

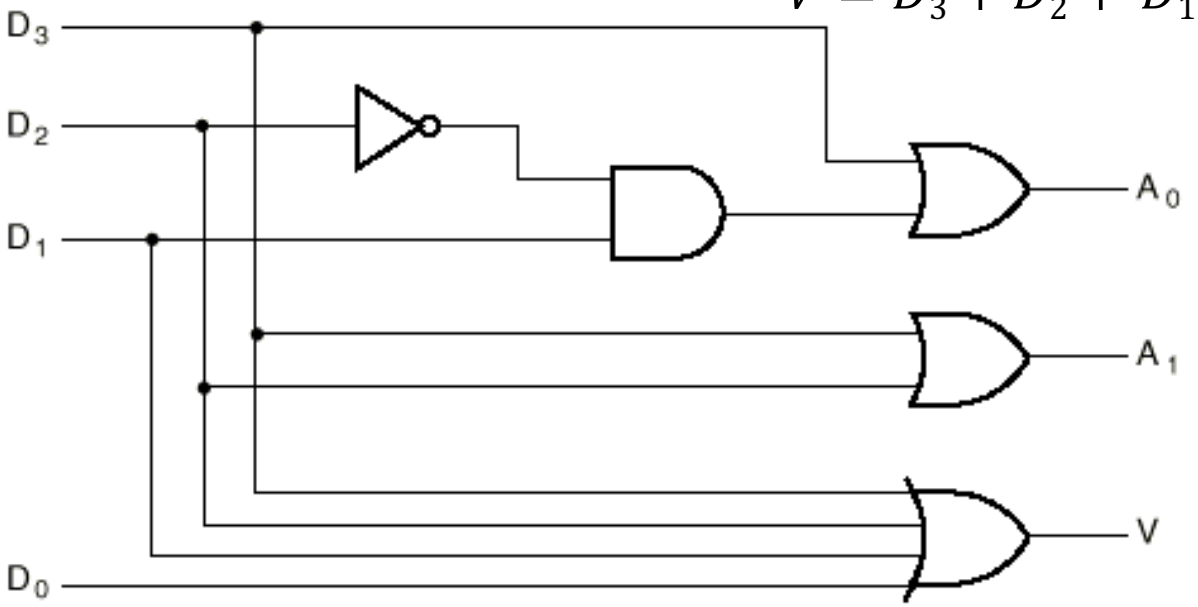
Priority Encoder: Sample1: Logic

Inputs				Outputs		
D_3	D_2	D_1	D_0	A_1	A_0	V
0	0	0	0	X	X	0
0	0	0	1	0	0	1
0	0	1	X	0	1	1
0	1	X	X	1	0	1
1	X	X	X	1	1	1

$$A_1 = D_3 + D_2$$

$$A_0 = D_3 + \overline{D_2} D_1$$

$$V = D_3 + D_2 + D_1 + D_0$$



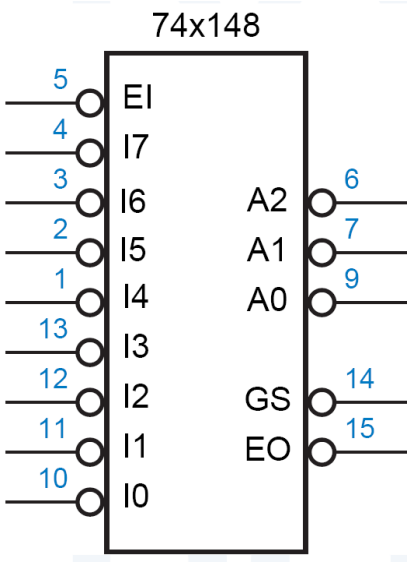
Interrupt logic for a CPU

- Design an interrupt logic for a CPU
  - Has two interrupts at the same time
  - Interrupt 1: Process of a key press event
  - Interrupt 2: Points to a power failure
    - Indicates a very severe problem

- How?
  - Using a priority encoder
  - Gives higher priority to the first interrupt

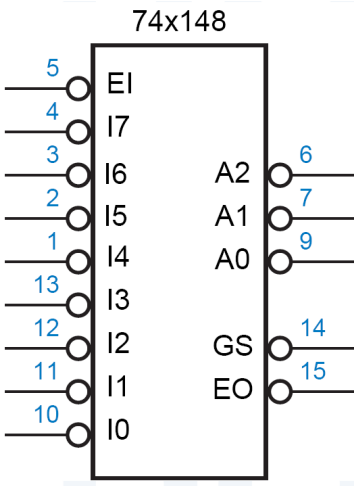
Priority Encoder: 74x148

- Input and outputs are active low
- EI_L: Enable input
  - Must be asserted to activate outputs
- GS_L is asserted
  - Device is enabled and
  - One or more of the request inputs is asserted
  - “Group Select” or “Got Something.”
- EO_L: Enable output
  - To be connected to the EI_L of another chip
  - if EI_L is asserted but no request input is asserted
  - => A lower-priority ’148 may be enabled
  - => EO_L is asserted

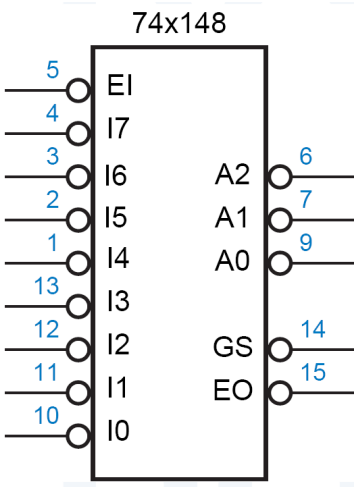


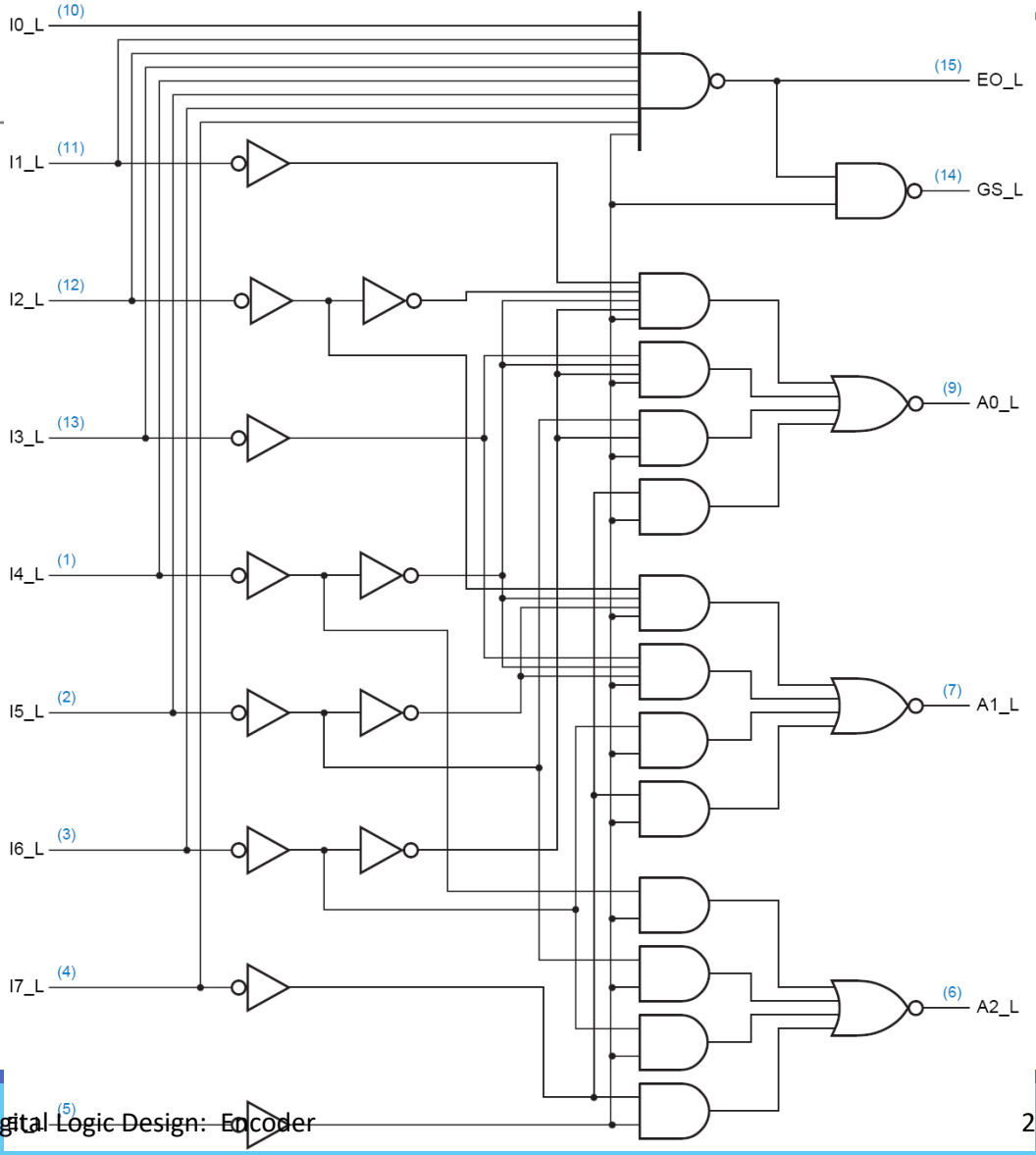
Priority Encoder: 74x148 (cont'd)

Inputs									Outputs				
/EI	/I0	/I1	/I2	/I3	/I4	/I5	/I6	/I7	/A2	/A1	/A0	/GS	/EO
1	x	x	x	x	x	x	x	x	1	1	1	1	1
0	x	x	x	x	x	x	x	0	0	0	0	0	1
0	x	x	x	x	x	x	0	1	0	0	1	0	1
0	x	x	x	x	x	0	1	1	0	1	0	0	1
0	x	x	x	x	0	1	1	1	0	1	1	0	1
0	x	x	x	0	1	1	1	1	1	0	0	0	1
0	x	x	0	1	1	1	1	1	1	0	1	0	1
0	x	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0



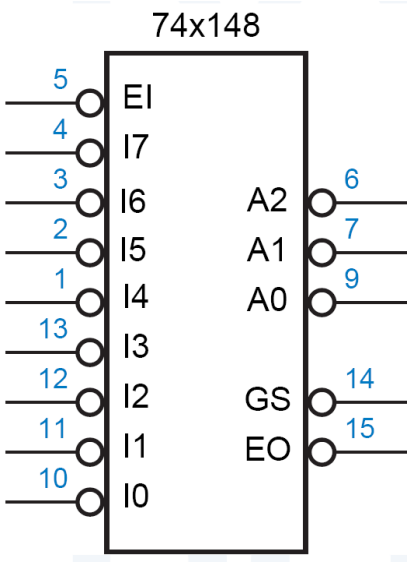
Priority Encoder: 74x148 (cont'd)





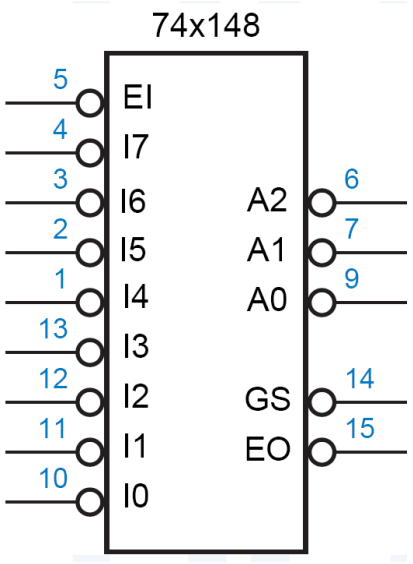
What is Output of 74x148?

- Input:
  - (EI,7,6,5,4,3,2,1,0) = (0,1,0,1,0,1,0,1,1)
- Output?



Output

- Input:
  - (EI,7,6,5,4,3,2,1,0) = (0,1,0,1,0,1,0,1,1)
- Output?
  - (EO, Gs, A2, A1, A0) = (1,0,0,0,1)



Thank You