BME2322 – Logic Design

The Instructors:

Dr. Görkem SERBES (C317)

gserbes@yildiz.edu.tr

https://avesis.yildiz.edu.tr/gserbes/

Lab Assistants:

Nihat AKKAN

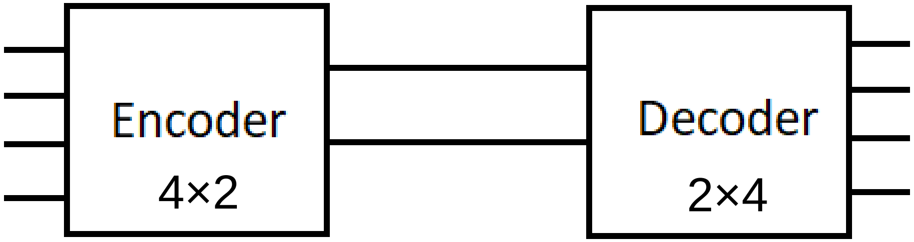
nakkan@yildiz.edu.tr

https://avesis.yildiz.edu.tr/nakkan

LECTURE 7

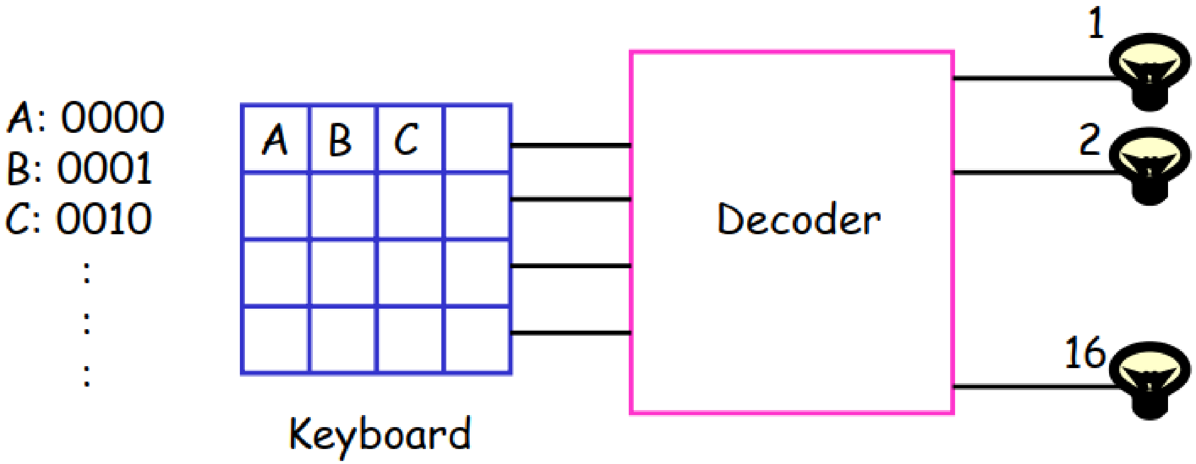
What is a decoder and encoder?

- A decoder is a combinational circuit that converts binary information from ‘n’ input lines to a maximum of 2^n unique output lines.
- They are called as n-to-m line decoders, where $m \leq 2^n$



- An encoder is a combinational circuit that performs the inverse operation of a decoder. An encoder has 2^n (or. fewer) input lines and ‘n’ output lines. The output lines generate a binary code corresponding to the input value.

What is a decoder?

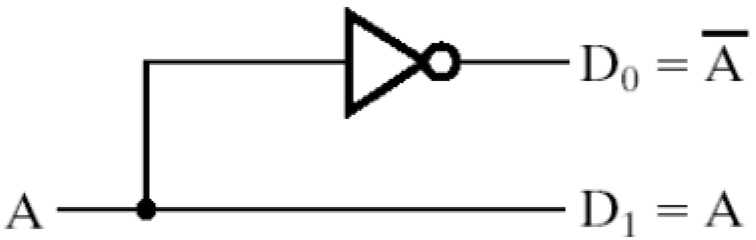


- $n \leq m \leq 2^n$
- Detect which of the 2^n combinations is represented at the inputs
- Produces m number of outputs, in which only one of them is "1"

1-to-2 (Line) Decoder

A	$\mathbf{D_0}$	$\mathbf{D_1}$
0	1	0
1	0	1

(a)



(b)

2-to-4 Decoder

- A 2-to-4 decoder operates according to the following truth table
  - The 2-bit input is called S1S0, and the four outputs are Q0-Q3
  - If the input is the binary number i, then output Qi is uniquely true

S1	S0	Q0	Q1	Q2	Q3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

- For instance, if the input S1 S0 = 10 (decimal 2), then output Q2 is true, and Q0, Q1, Q3 are all false
- This circuit “decodes” a binary number into a “one-of-four” code

How can you build a 2-to-4 decoder?

We have a truth table, so we can write equations for each of the four outputs (Q0-Q3)

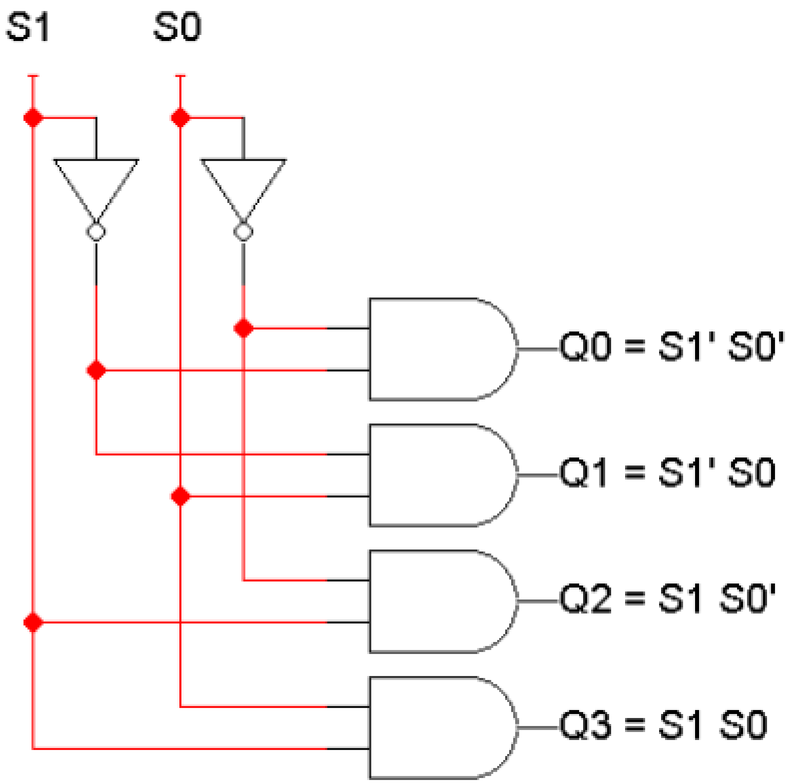
S1	S0	Q0	Q1	Q2	Q3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

Equations are:

$$Q0 = S1' \, S0'$$
$$Q1 = S1' \, S0$$
$$Q2 = S1 \, S0'$$
$$Q3 = S1 \, S0$$

2-to-4 Decoder

S1	S0	Q0	Q1	Q2	Q3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



Enable Inputs

- Many devices have an additional enable input, which is used to “activate” or “deactivate” the device
- For an active-high decoder ,
  – EN=1 activates the decoder, so it behaves as specified earlier. Exactly one of the outputs will be 1
  – EN=0 “deactivates” the decoder. By convention, that means all of the decoder’s outputs are 0
- We can include this additional input in the decoder’s truth table:

EN	S1	S0	Q0	Q1	Q2	Q3
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Abbreviated Truth Tables

- In the right side table, note that whenever EN=0, the outputs are always 0, regardless of inputs S1 and S0.

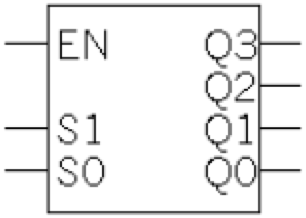
EN	S1	S0	Q0	Q1	Q2	Q3
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	1	1	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

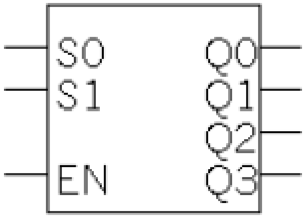
- We can abbreviate the table by writing x's in the input columns for S1 and S0

EN	S1	S0	Q0	Q1	Q2	Q3
0	x	x	0	0	0	0
1	0	0	1	0	0	0
1	0	1	0	1	0	0
1	1	0	0	0	1	0
1	1	1	0	0	0	1

Blocks and Abstraction

- Decoders are commonly used and it is a good idea to encapsulate them and treat them as an individual entity.
- Block diagrams for 2-to-4 decoders are shown below.





$$Q0 = S1' \, S0'$$
$$Q1 = S1' \, S0$$
$$Q2 = S1 \, S0'$$
$$Q3 = S1 \, S0$$

Advantages of a decoder block :

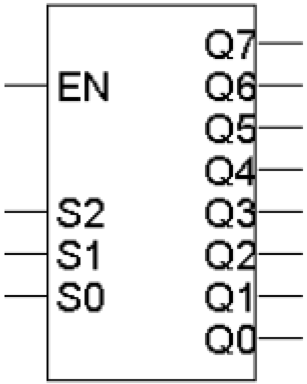
– You can use the decoder as long as you know its truth table or equations, without knowing exactly what's inside.

– It makes diagrams simpler by hiding the internal circuitry.

– It simplifies hardware reuse. You don't have to keep rebuilding the decoder from scratch every time you need it.

3-to-8 Decoder

- Larger decoders are similar. Here is a 3-to-8 decoder
  - The block symbol is on the right
  - A truth table (without EN) is below
  - Output equations are at the bottom right
- Again, only one output is true for any input combination



S2	S1	S0	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

$Q0 = S2' \, S1' \, S0'$

$Q1 = S2' \, S1' \, S0$

$Q2 = S2' \, S1 \, S0'$

$Q3 = S2' \, S1 \, S0$

$Q4 = S2 \, S1' \, S0'$

$Q5 = S2 \, S1' \, S0$

$Q6 = S2 \, S1 \, S0'$

$Q7 = S2 \, S1 \, S0$

When the decoder is used?

S1	S0	Q0	Q1	Q2	Q3
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

$Q0 = S1'\ S0'$
$Q1 = S1'\ S0$
$Q2 = S1\ S0'$
$Q3 = S1\ S0$

- Decoders are sometimes called minterm generators
  – For each of the input combinations, exactly one output is true
  – Each output equation contains all of the input variables
  – These properties hold for all sizes of decoders
- Arbitrary functions can be implemented with decoders. If a sum of minterms equation for a function is given, a decoder (a minterm generator) is used to implement that function

Decoder based Addition operation

- Let's make a circuit that adds three 1-bit inputs X, Y and Z
- We will need two bits to represent the total; let's call them C and S, for "carry" and "sum." Note that C and S are two separate functions of the same inputs X, Y and Z
- Here are a truth table and sum-of-minterms equations for C and S

X	Y	Z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$0 + 1 + 1 = 10 \longrightarrow$ (row 0 1 1)

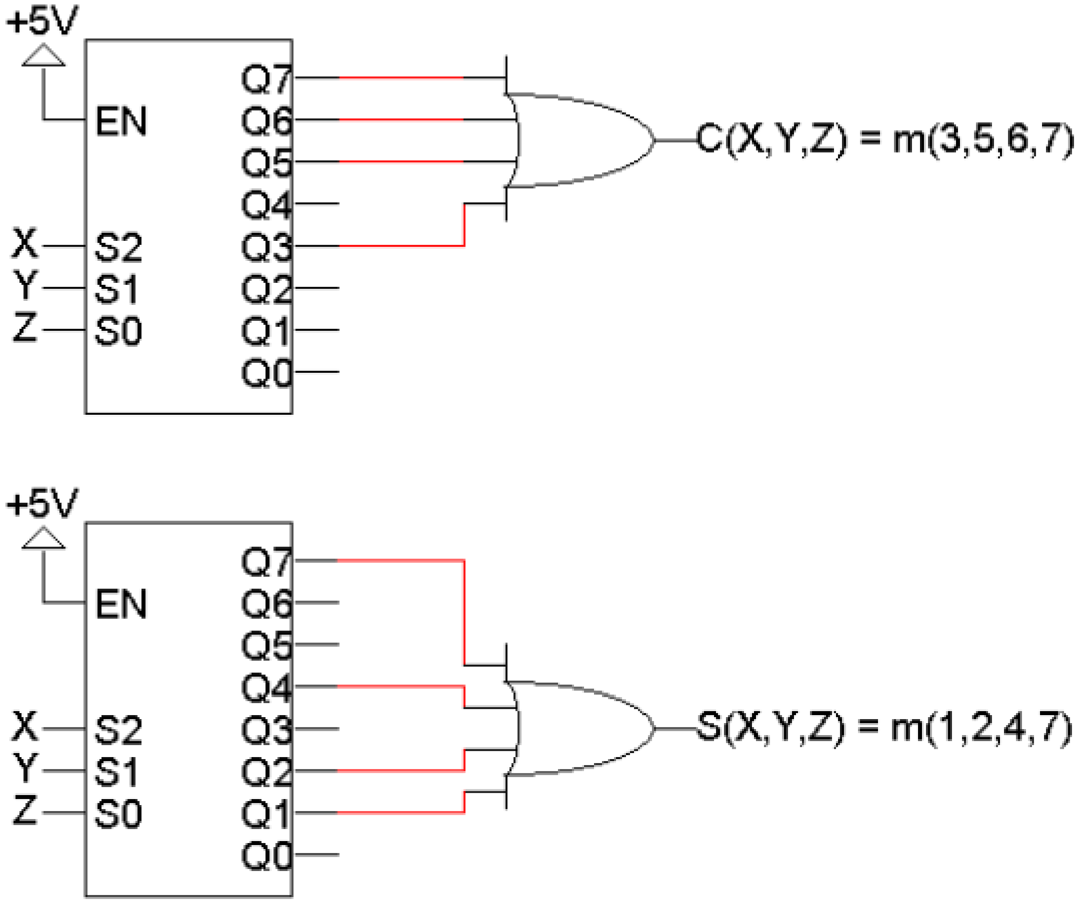
$1 + 1 + 1 = 11 \longrightarrow$ (row 1 1 1)

$C(X,Y,Z) = \Sigma m(3,5,6,7)$

$S(X,Y,Z) = \Sigma m(1,2,4,7)$

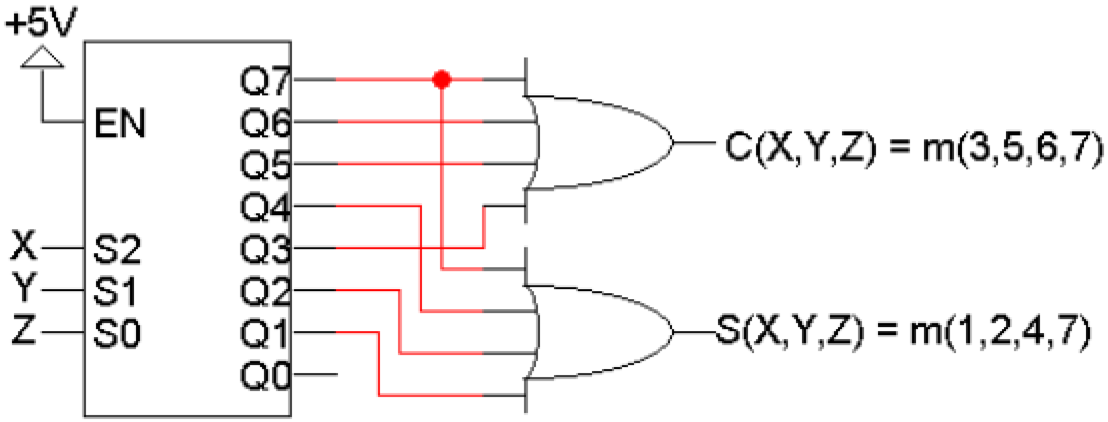
Decoder based Addition operation cont.

- Here, two 3-to-8 decoders implement C and S as sums of minterms.



Decoder based Addition operation cont.

- Since the two functions C and S both have the same inputs, we could use just one decoder instead of two.



Building a 3-to-8 decoder

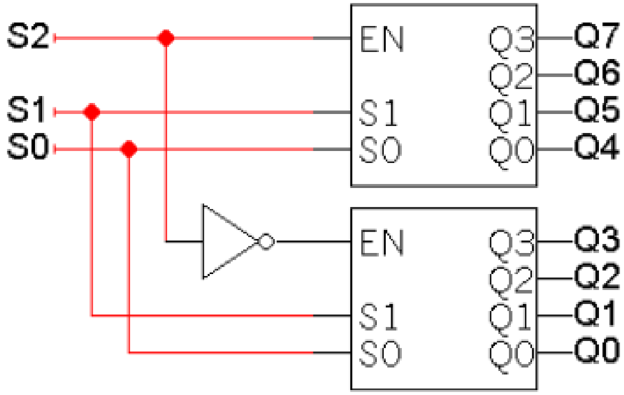
- You could build a 3-to-8 decoder directly from the truth table and equations below, just like how we built the 2-to-4 decoder
- Another way to design a decoder is to break it into smaller pieces
- Notice some patterns in the table below:
  – When S2 = 0, outputs Q0-Q3 are generated as in a 2-to-4 decoder
  – When S2 = 1, outputs Q4-Q7 are generated as in a 2-to-4 decoder

S2	S1	S0	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

$Q0 = S2'\ S1'\ S0' = m_0$
$Q1 = S2'\ S1'\ S0 = m_1$
$Q2 = S2'\ S1\ S0' = m_2$
$Q3 = S2'\ S1\ S0 = m_3$
$Q4 = S2\ S1'\ S0' = m_4$
$Q5 = S2\ S1'\ S0 = m_5$
$Q6 = S2\ S1\ S0' = m_6$
$Q7 = S2\ S1\ S0 = m_7$

Building a 3-to-8 decoder cont.

- You can use enable inputs to string decoders together. Here's a 3-to-8 decoder constructed from two 2-to-4 decoders:

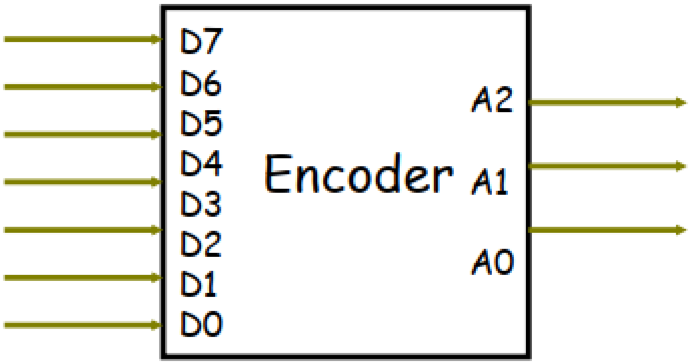


S2	S1	S0	Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

Be careful not to confuse the "inner" inputs and outputs of the 2-to-4 decoders with the "outer" inputs and outputs of the 3-to-8 decoder (which are in boldface)

Encoder

- An encoder is a digital function that performs the inverse operation of a decoder
- Octal-to-Binary Encoder: This encoder has eight inputs, one for each of the octal digits, and three outputs that generate the corresponding binary number
- – Only one input can have the value of 1 at any given time



Octal-to-Binary Encoder

Truth Table for Octal-to-Binary Encoder

Inputs								Outputs		
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0	A_2	A_1	A_0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

A0= D1 + D3 + D5 + D7,

A2= D4 + D5 + D6 + D7

A1= D2 + D3 + D6 + D7

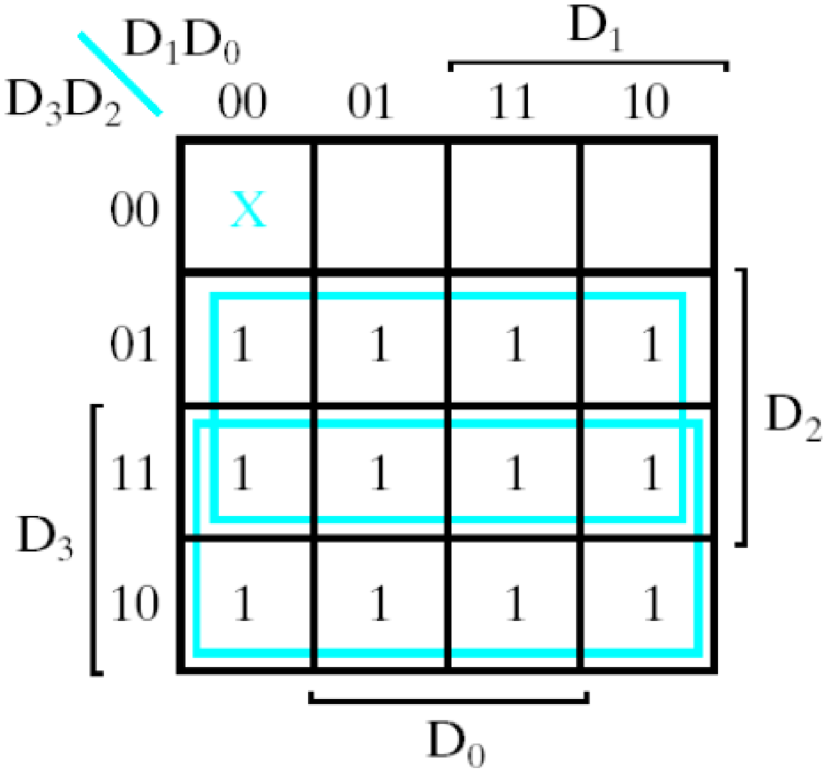
Priority Encoder

- A priority encoder is a combinational circuit that implements a priority function
- Octal-to-Binary: If more than one input is active simultaneously, the output produces an incorrect combination
- Priority Encoder: If two or more inputs are active simultaneously, the input having the highest priority takes precedence

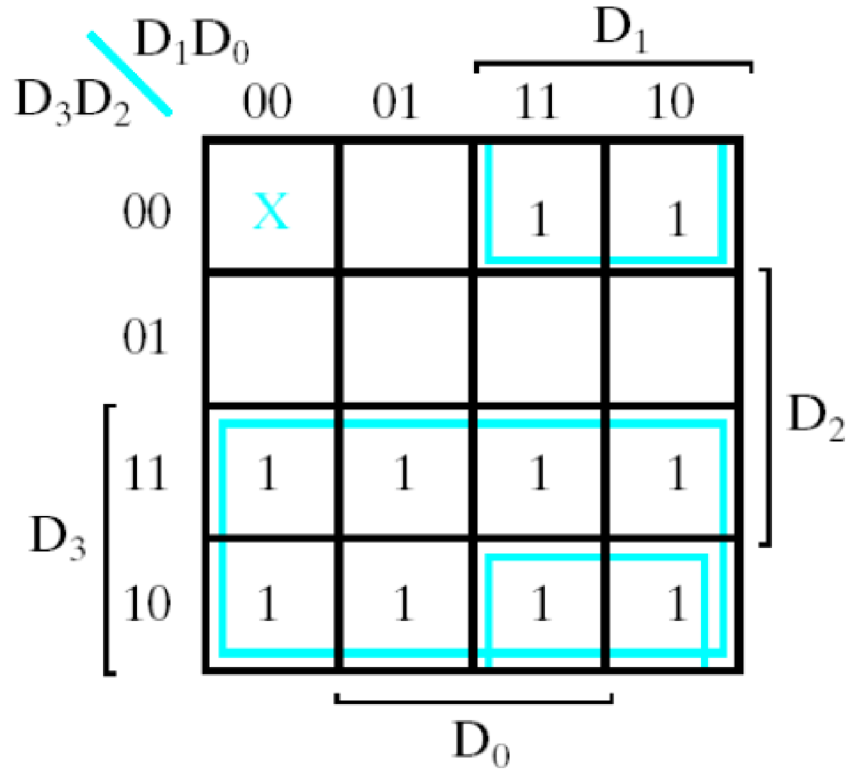
Truth Table of Priority Encoder

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

4 Input Priority Encoder



$A_1 = D_2 + D_3$



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

4 Input Priority Encoder Circuit