**Digital Encoder** :

**Digital Encoder** ( **Binary Encoder** )takes its inputs one at a time and then converts them into a single encoded output.

we can say that a **Digital Encoder**, is a multi-input combinational logic circuit that converts the logic level “1” inputs into an equivalent binary code at its output.

A digital encoder produces outputs of 2 bit or 3 bit or 4 bit codes depending on the number of input lines. An n bit digital encoder has 2 to the power of N input lines and n-bit output lines with common types that include 4-to-2, 8-to-3 and 16-to-4 line configurations.

 common types of **Digital Encoder**s include :

(4 to 2) – (8 to 3) – (16 to 4) line configurations

Usage :   1- encode either a decimal or hexadecimal input pattern to typically a binary or “B.C.D”

2-generate the binary equivalent of the input line whose value is equal to “1”

-----------------------------------------------------------------

Why do you need a **Priority Encoder** ?

the main disadvantages of standard digital encoders is that they can generate the wrong output code when there is more than one input present at logic level “1”.

Also, an output code of all logic “0”s can be generated when all of its inputs are at “0” OR when input D0 is equal to one.

The solution is to Prioritize the level of each input pin

. then if there is more than one input at logic level 1 at the same time, the actual output code would only correspond to the input with the highest designated priority

And we do that by using **Priority Encoder**

-------------------------------------------------------------

**4 : 2 Encoder:**

consists of

**four inputs x0 ,x1 ,x2 ,x3**

**two outputs A,B**

**only one of these 4 inputs can be 1 so that we get the respective binary code at the output**

**Truth Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **x3** | **x2** | **x1** | **x0** | **B** | **A** |
| 0 | 0 | 0 | 0 | X | X |
| 0 | 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | X | 0 | 1 |
| 0 | 1 | X | X | 1 | 0 |
| 1 | X | X | X | 1 | 1 |

\*\*the lower priority bits are shown as x. and also when the inputs are (0000) the outputs are not valid so they are X X

***circuit operation :***

every output is gained by OR gate which is connected to the NAND INV outputs of the corresponding input lines.

The NAND gate of each stages receives its input bit, as well as the NAND gate outputs of all higher priority stages.

This structure implies that an active input on stage n effectively disables all lower stages n-1 🡪 0.

-----------------------------------

# **The circuit :**

Output of B = X2  + X3

Output of A = X3 + X1 2