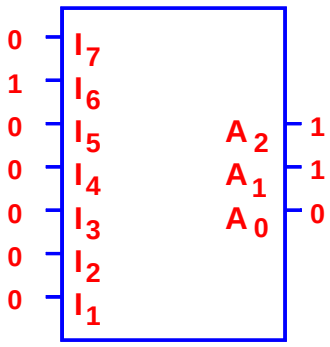
Encoder Introduction

An **Encoder** is the **opposite of a decoder**. It **outputs a binary number for each input applied to it**. There are several different types but we will only study the two major types.

- **Encoder with Mutually Exclusive Inputs and**
- **Priority Encoders**

Encoder's with Mutually Exclusive Inputs

This type of Encoder will have **one and only one active input at a time**. If two or more inputs are active at the same time, all outputs will be inactive (in this case, 000).



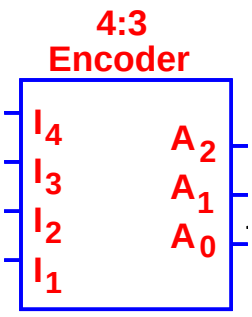
Encoder with MutuallyExclusive Inputs

The term "**Mutually Exclusive**" refers to the **absence of an I_0 input**. When the device is not active, due to a chip enable gate, **all outputs will go inactive (all 0's)**.

If the device had an I_0 input, there wouldn't be a way to discern between **an inactive encoder** and an **active encoder with I_0 as the single active input**. So, in this type of device, the **I_0 input** is left out. **The output of the encoder will represent the subscript of the input which is active.** For example, if I_3 is active, the output will be **$(A_2, A_1, A_0) = (0,1,1)$**.

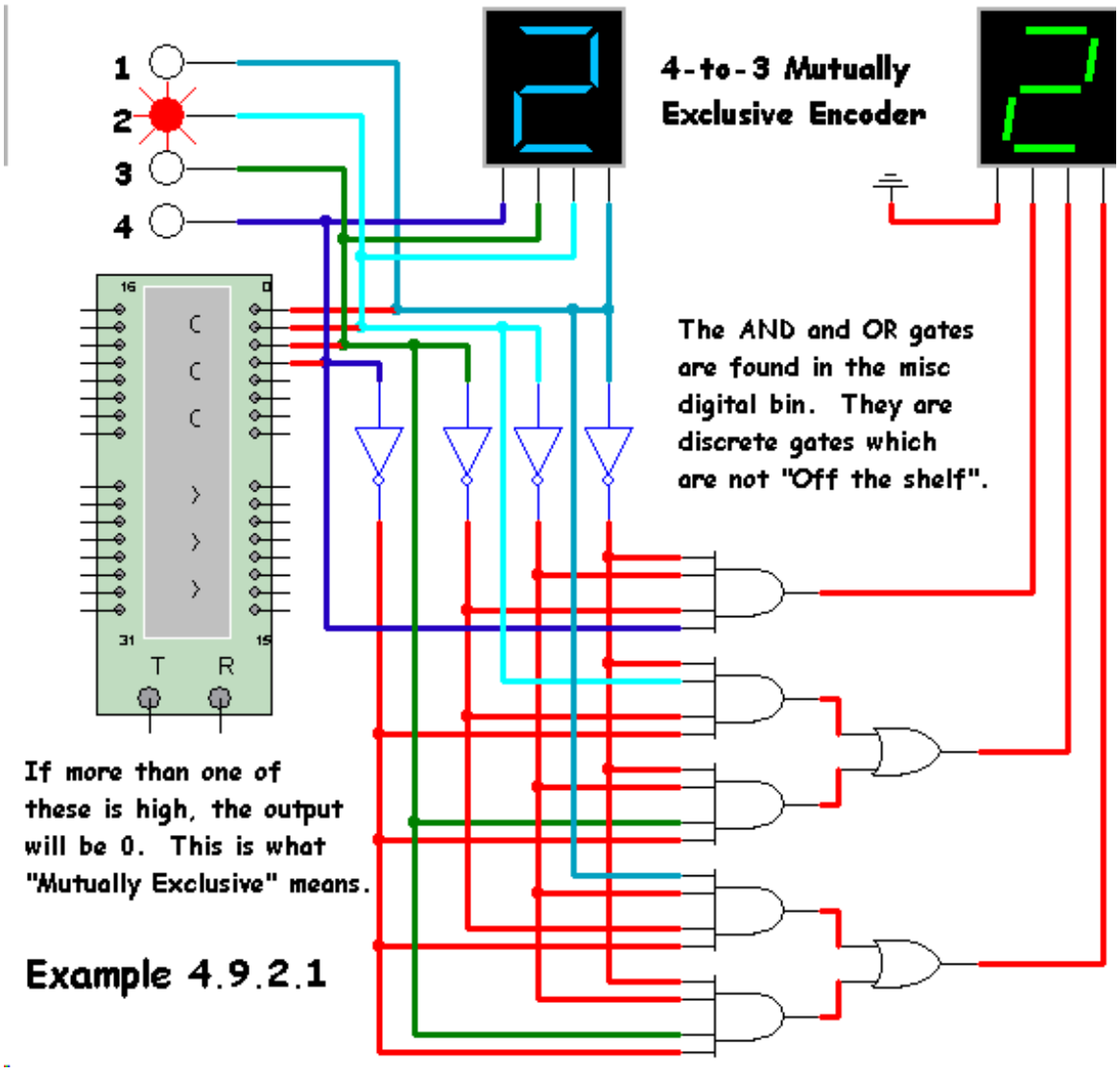
The 4:3 Mutually Exclusive Encoder

Note that the output will be the binary equivalent of the input position subscript of the "1". **If there are more than 1 input active on the input the output is "0".**

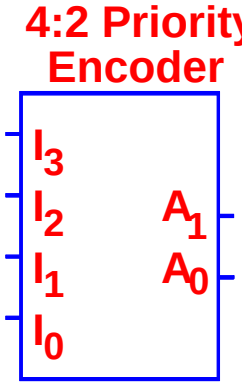


I_4	I_3	I_2	I_1	A_2	A_1	A_0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	0	0
0	1	0	0	0	1	1
0	1	0	1	0	0	0
0	1	1	0	0	0	0
0	1	1	1	0	0	0
1	0	0	0	1	0	0
1	0	0	1	0	0	0
↓		↓	↓		↓	
1	1	1	1	0	0	0

A MultiSim 4:3 Mutually Exclusive Encoder



The Priority Encoder



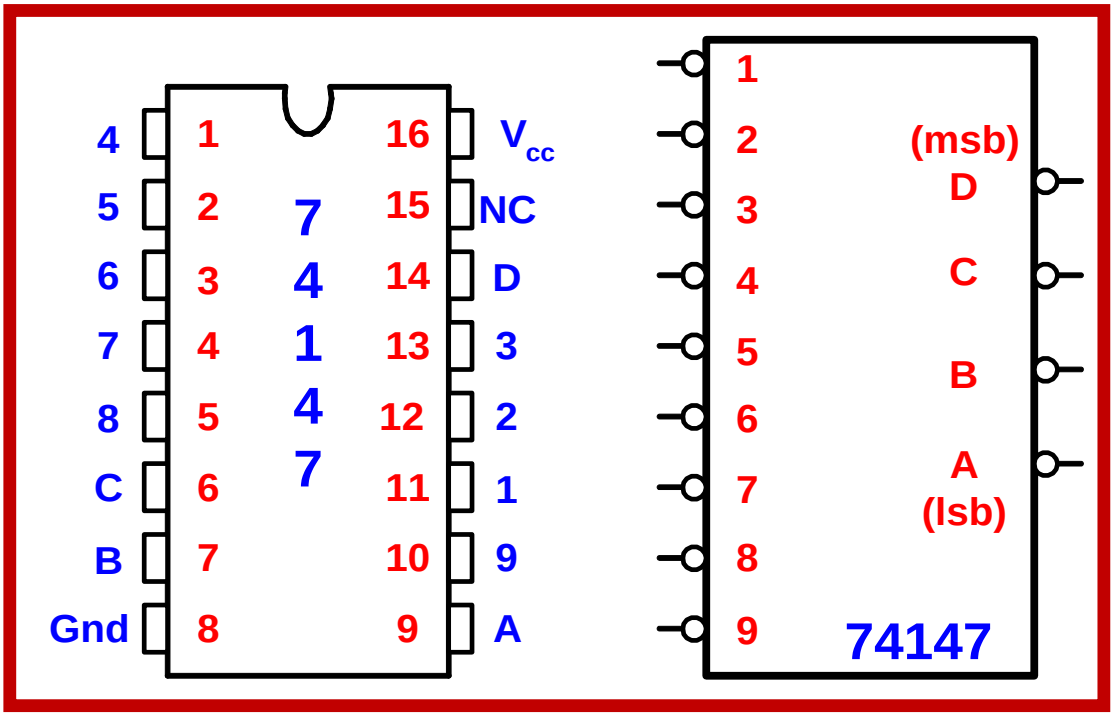
I_3	I_2	I_1	I_0	A_1	A_0
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

The **Priority Encoder** is one of the most popular Encoders. In this encoder, the output will represent the input subscript associated with the highest active input. For instance, take **row three** of the table above. If the Priority Encoder's input were as follows, $(I_3,I_2,I_1,I_0)=(0,1,1,0)$, the output would be $(A_1,A_0)=(1,0)$. As it turns out, all the positions below in priority from the highest active input are "**don't cares**". You really **don't care** if they are 1's or 0's.

Note that **in some Priority Encoders**, the **I_0 input exists (74x148) while in others it doesn't (74x147)**. The Priority Encoder is used for Priority Management in computers. The inputs correspond to **IRQ** numbers.

The 74147 Priority Encoder

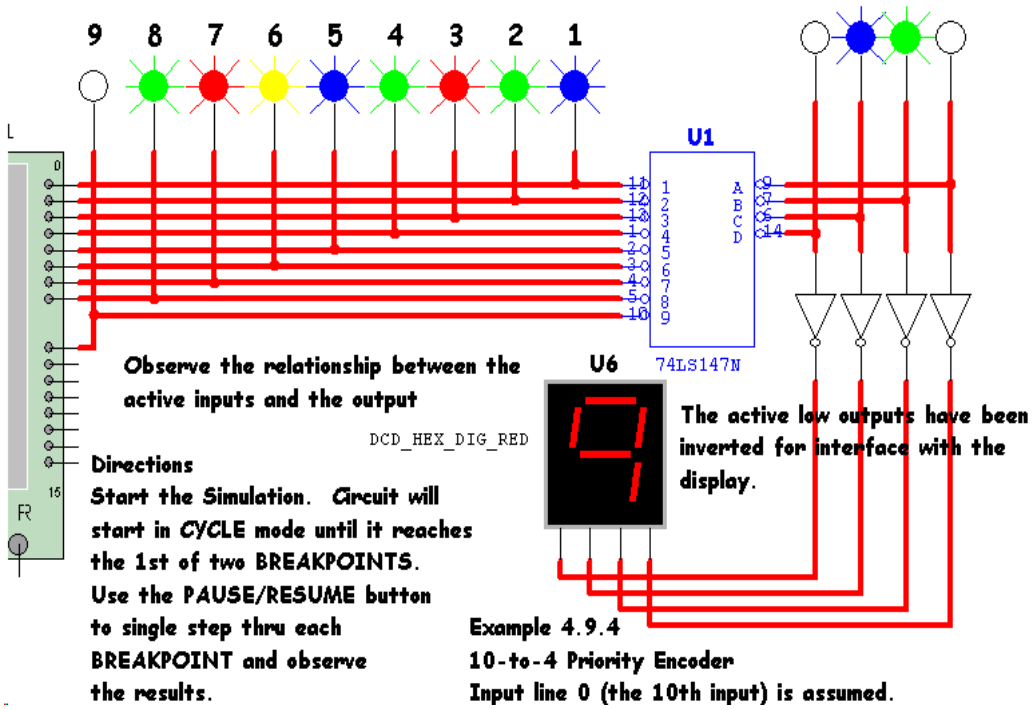
The first Priority Encoder we will discuss is the **74147 (10/4 Active Low input, Active Low output).**



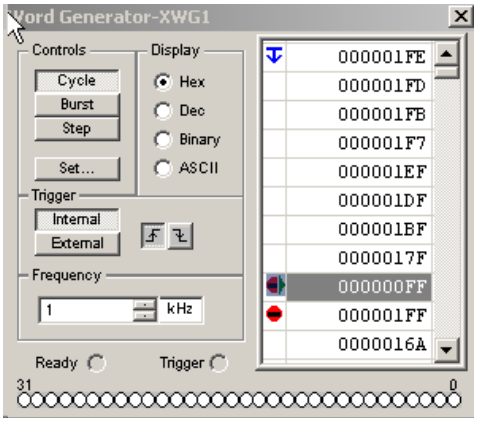
Note that this chip was introduced as being a 10/4 encoder but there are **only 9 input lines**. **The 0 line is implied**. When a zero is desired, **leave all lines inactive and an active low zero will be output**. This chip is essentially a **decimal to BCD encoder**.

MultiSim Example

Hook up the following circuit in MultiSim:



Once you have hooked up the circuit, open the Word Generator and program it in HEX as shown here. Note the two asterisks beside the **0FF** and the **1FF**. These are breakpoints set into to allow the circuit to be placed into ***CYCLE*** mode. Upon reaching the breakpoints, the ***CYCLE*** mode will stop and then the user can single step through the rest of the steps and observe more closely what happens on those particular steps.



If the reader doesn't understand how the HEX program was obtained, the following explanation should help:

WG_{out} ⇒	8	7	6	5	4	3	2	1	0	HEX
$Device_{in}$ ⇒	9	8	7	6	5	4	3	2	1	Equiv.
Add. line 0	1	1	1	1	1	1	1	1	0	1FE
Add. line 1	1	1	1	1	1	1	1	0	1	1FD
Add. line 2	1	1	1	1	1	1	0	1	1	1FB
Add. line 3	1	1	1	1	1	0	1	1	1	1F7
Add. line 4	1	1	1	1	0	1	1	1	1	1EF
Add. line 5	1	1	1	0	1	1	1	1	1	1DF
Add. line 6	1	1	0	1	1	1	1	1	1	1BF
Add. line 7	1	0	1	1	1	1	1	1	1	17F
Add. line 8	0	1	1	1	1	1	1	1	1	0FF
Add. line 9	1	1	1	1	1	1	1	1	1	1FF
Add. line A	1	0	1	1	0	1	0	1	0	16A
Add. line B	0	1	0	0	1	1	0	1	0	09A
Add. line **C**	1	1	1	1	1	0	1	1	0	1F6

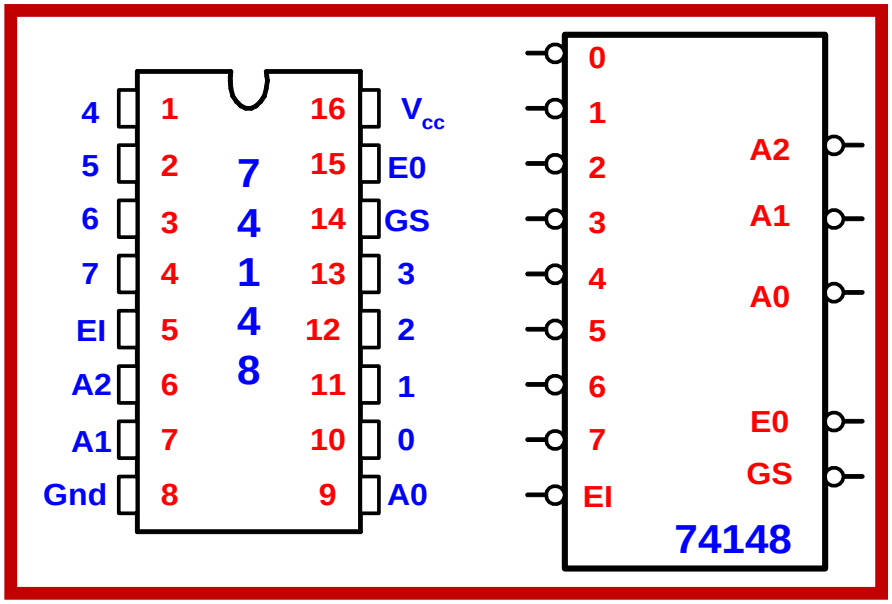
Once you have hooked up the inputs of the circuit to the desired Word Generator outputs, produce a table with the outputs and device inputs listed as column headings. Then program the Word Generator on paper in Binary to make the Generator perform in the desired manner. Then break it up into groups of 4 starting with Generator output 0. Once you do that, convert to HEX as shown.

Once you do this, program the generator by first selecting HEX in the program section. Then, click on address line 0, enter the desired HEX equivalent into the HEX entry block. Each time you hit the enter key, the program will go to the next program line. Continue until you are finished.

Break Points are placed by double clicking in the area to the left of the address line.

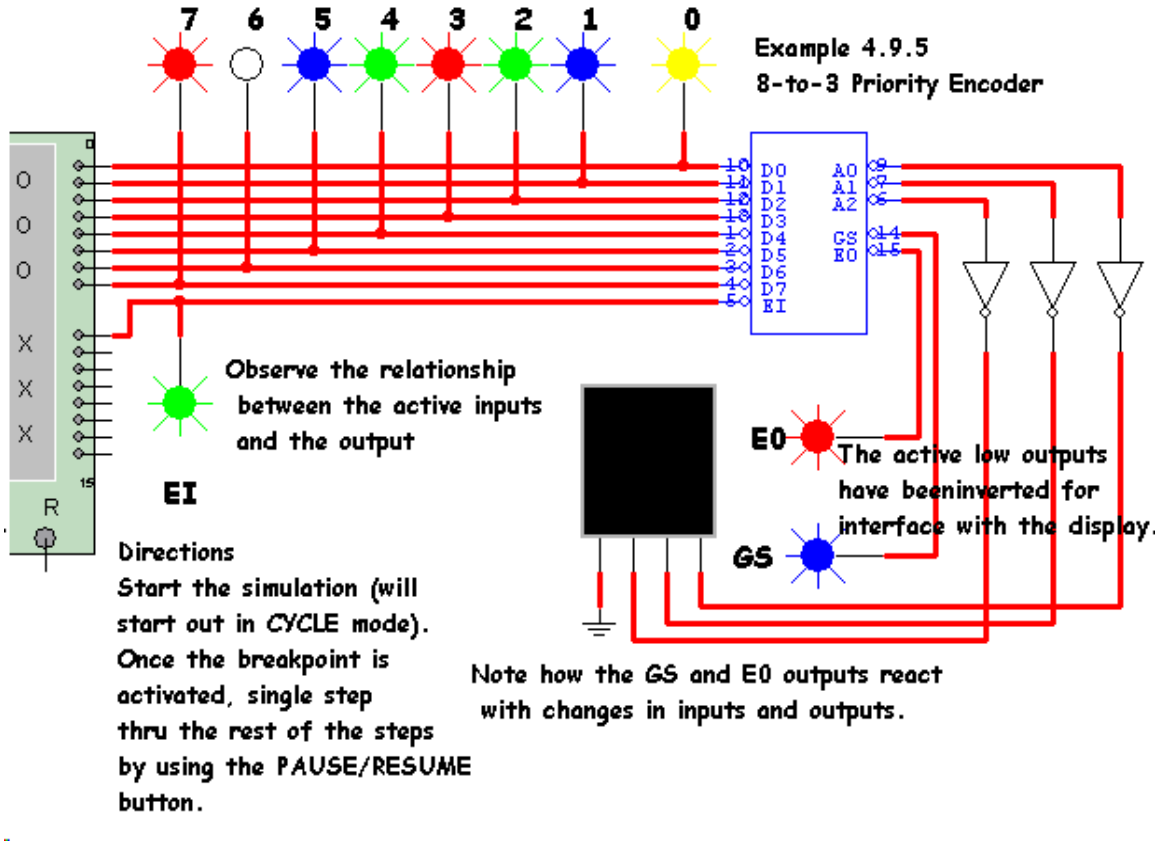
The 74148 Priority Encoder

The 2nd most popular **Priority Encoder** we should discuss is the **74148 (8/3 Active Low input, Active Low Output)**. This chip is useful when encoding decimal to octal.



The additional input is **EI (Enable Input)**. The **EO** and **GS** outputs are provided for cascading chips. The **EO (Enable output)** is low **when none of the input lines are active** and should be connected to the **EI (Enable Input)** of the next stage. The **GS** is **low when one or more input lines are active.**

74148 Priority Encoder MultiSim Example #1

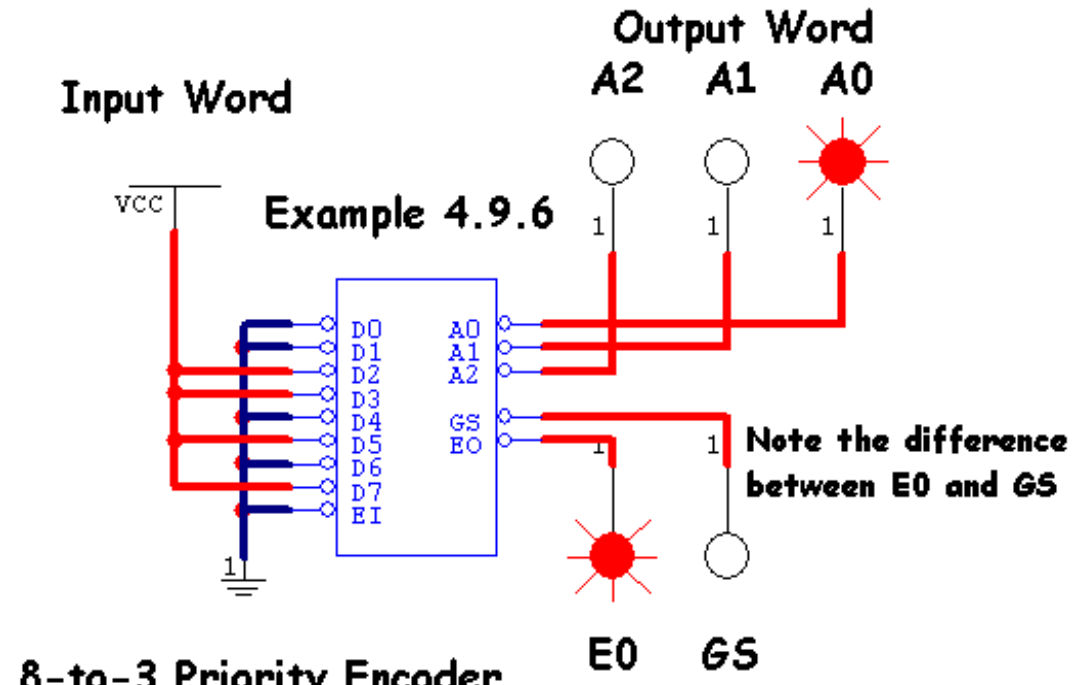


The word generator is programmed as before.

74148 Priority Encoder MultiSim Example #2:

What will the **output word** be **(E0,GS,A2,A1,A0)** when the following **input word** is applied to a **74148?**

Input word = **(EI,7,6,5,4,3,2,1,0) = (0,1,0,1,0,1,1,0,0)**



Answer: Output word **(E0,GS,A2,A1,A0) = (1,0,0,0,1)**
(An Active low 6 preceded by E0 and GS signals)