EXPERIMENT No-9

Aim: To Design an Encoder and Decoder using VHDL

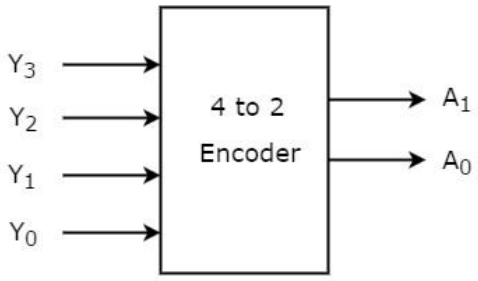
Encoder:

An encoder is a combinational circuit that converts binary information in the form of a 2^N input lines into N output lines, which represent N bit code for the input. For simple encoders, it is assumed that only one input line is active at a time.

4:2 Encoder

Let 4 to 2 Encoder has four inputs Y_3, Y_2, Y_1 & Y_0 and two outputs A_1 & A_0. At any time, only one of these 4 inputs can be '1' in order to get the respective binary code at the output.

Block diagram

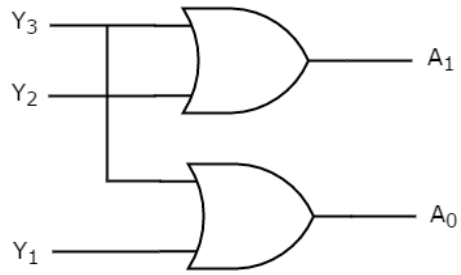


Truth Table

Inputs				**Outputs**	
Y_3	**Y_2**	**Y_1**	**Y_0**	**A_1**	**A_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

Boolean functions for each output as: A1=Y3+Y2, A0=Y3+Y1

Circuit diagram



VHDL program for 4 to 2 encoders:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity encod1 is
Port (a: in STD_LOGIC_VECTOR (3 downto 0);
q:out STD_LOGIC_VECTOR (1 downto 0));
end encod1;
architecture Behavioral of encod1 is
begin
q<="00" when a="0001" else
"01" when a="0010" else
"10" when a="0100" else
"11";
end Behavioral;
```

VHDL program for 8 to 3 encoder:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity encoder is
Port ( i : in STD_LOGIC_VECTOR (7 downto 0); y : out
STD_LOGIC_VECTOR (2 downto 0));
end encoder;
```

```
architectureBehavioralof encoder is begin
with i select
y<="000" when "00000001",
"001" when "00000010",
"010" when "00000100",
"011" when "00001000",
"100" when "00010000",
"101" when "00100000",
"110" when "01000000",
"111" when others;
end Behavioral;
```

Output:

Decoder:

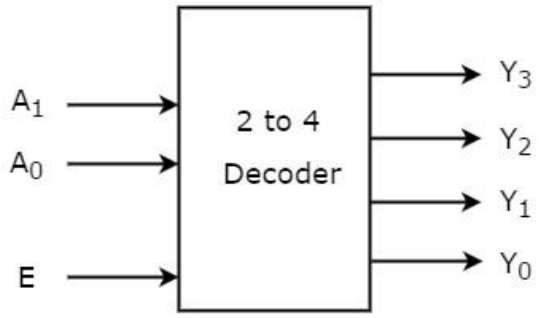
A decoder is a device which does the reverse of an encoder, undoing the encoding so that the original information can be retrieved. The same method used to encode is usually just reversed in order to decode.

Decoder is a combinational circuit that has 'n' input lines and maximum of 2^n output lines.

2 to 4 Decoder

Let 2 to 4 Decoder has two inputs A_1 & A_0 and four outputs Y_3, Y_2, Y_1 & Y_0.

B**lock diagram**



One of these four outputs will be '1' for each combination of inputs when enable, E is '1'.

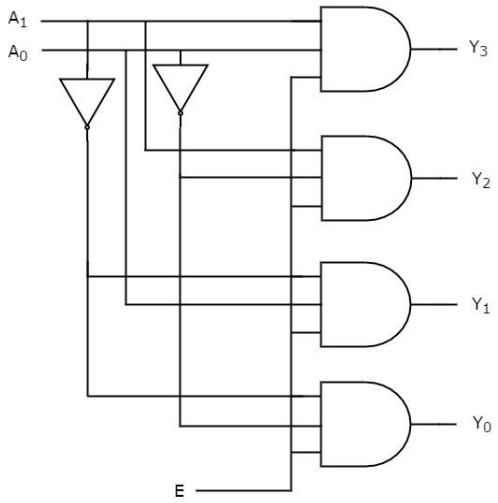
Truth table

Enable	**Inputs**		**Outputs**			
E	**A_1**	**A_0**	**Y_3**	**Y_2**	**Y_1**	**Y_0**
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0

1	1	1	1	0	0	0

Boolean functions for each output as Y3=E.A1.A0, Y2=E.A1.A0′, Y1=E.A1′.A0, Y0=E.A1′.A0′

Circuit diagram



VHDL Program for 2:4 decoder:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity decoder is
Port ( en : in STD_LOGIC;
s : in STD_LOGIC_VECTOR (1 downto 0);
z : out STD_LOGIC_VECTOR (3 downto 0));
end decoder;
architecture arch of decoder is
signal p,q : STD_Logic;
begin
p <= not s(1);
q <= not s(0);
z(0) <= p and q;
z(1) <= p and s(0);
z(2) <= q and s(1);
z(3) <= s(0) and s(1);
```

```
end arch;
```

VHDL Program for 3:8 decoder:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity dec1 is
Port (s: in STD_LOGIC_VECTOR (2 downto 0);
y:out STD_LOGIC_VECTOR (7 downto 0));
end dec1;
architecture Behavioral of dec1 is begin
with sel select y<="00000001" when "000",
"00000010" when "001",
"00000100" when "010",
"00001000" when "011",
"00010000" when "100",
"00100000" when "101",
"01000000" when "110",
"10000000" when "111",
"00000000" when others;
end Behavioral;
```

Output: