INSTITUTE OF ENGINEERING AND TECHNOLOGY

DAVV , INDORE

CIRCUIT DESIGN USING HDL (EIR7C4)

(Lab Assignment)

Session (2020-21)

Submitted to:-

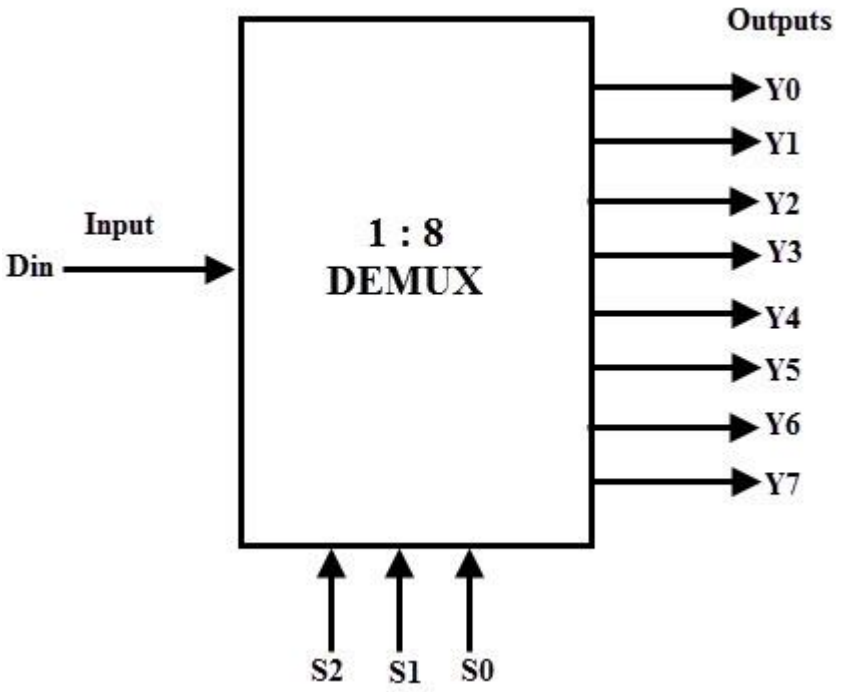
Dr. Vaibhav Nema

Submitted by :-

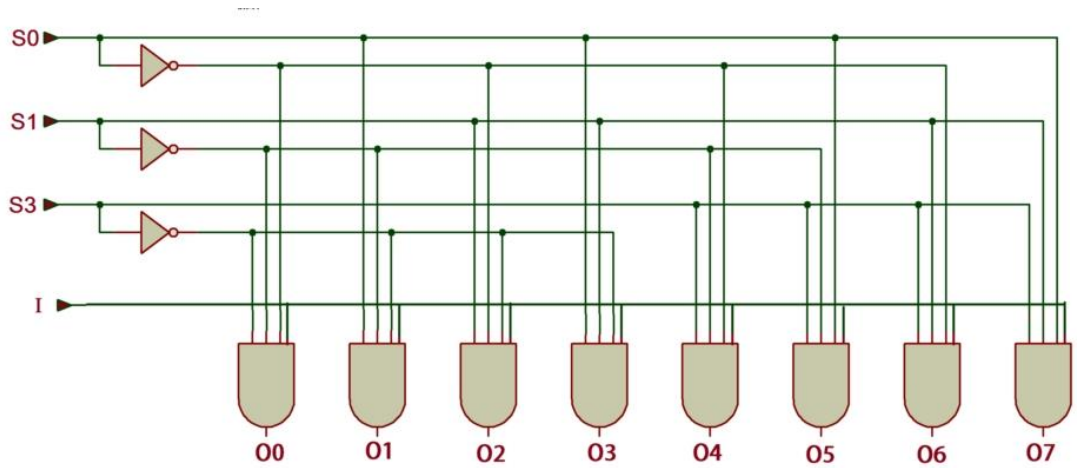
Name:- Rahul Patel

Roll. No:-17E7033

1:8 demux using structural modeling :-



Circuit diagram:-



Truth table:-

S2	S1	S0	O0	O1	O2	O3	O4	O5	O6	O7
0	0	0	I	0	0	0	0	0	0	0
0	0	1	0	I	0	0	0	0	0	0
0	1	0	0	0	I	0	0	0	0	0
0	1	1	0	0	0	I	0	0	0	0
1	0	0	0	0	0	0	I	0	0	0
1	0	1	0	0	0	0	0	I	0	0
1	1	0	0	0	0	0	0	0	I	0
1	1	1	0	0	0	0	0	0	0	I

Design code:-

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity demuxstu is
        Port ( din : in STD_LOGIC;
               s : in STD_LOGIC_VECTOR (2 downto 0);
               y : out STD_LOGIC_VECTOR (7 downto 0) );
end demuxstu;
architecture structural  of demuxstu is
component notgate
        Port ( a : in STD_LOGIC;
                 b : out STD_LOGIC) ;
end component;
component andgate is
```

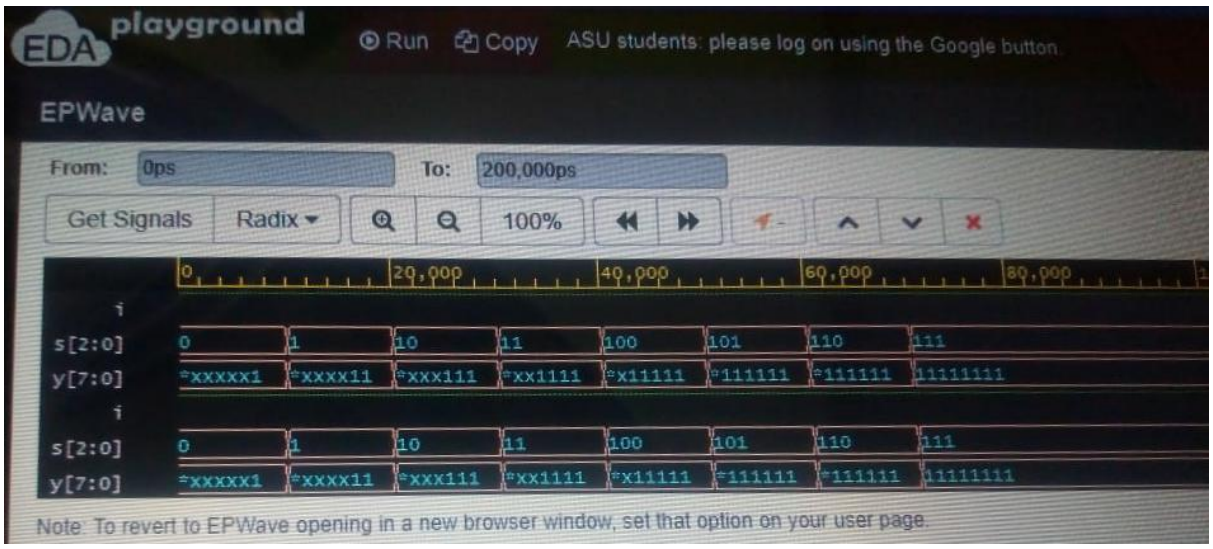
```
Port ( a,b,c,d : in STD_LOGIC;
       e : out STD_LOGIC);
end component;
signal s2b, s1b, s0b:std_logic;
begin
z1:notgate port map (s(2), s2b);
z2:notgate port map (s(1), s1b) ;
z3:notgate port map (s(0), s0b) ;
z4:andgate port map (s2b, s1b, s0b, din, y (0));
z5:andgate port map (s2b, s1b, s(0), din, y(1) );
z6: andgate port map (s2b, s(1), s0b, din, y (2));
z7: andgate port map (s2b, s(1),s (0),din, y (3));
z8: andgate port map (s(2), s1b,s0b,din, y(4));
z9: andgate port map (s(2), s1b,s (0), din,y(5));
z10: andgate port map (s(2),s(1), s0b, din, y(6));
z11:andgate port map (s(2),s (1),s(0), din,y(7));
end structural ;
```

Test bench:-

```
library IEEE;
use IEEE.std_logic_1164.all;
entity demux_tb is
end entity;
architecture demux1_8 of demux_tb is
component dmux1_8 is
```

```
port( din : in STD_LOGIC;
        s : in STD_LOGIC_VECTOR (2 downto 0);
        y : out STD_LOGIC_VECTOR (7 downto 0));
end component;
signal din : STD_LOGIC;
signal s: STD_LOGIC_VECTOR (2 downto 0);
signal y: STD_LOGIC_VECTOR (7 downto 0);
begin
uut: dmux1_8 port map(      din => din,
s => s,
y => y);
stim: process
begin
din <= '1';
s <= "000";   wait for 20 ns;
s <= "001";   wait for 20 ns;
s <= "010";   wait for 20 ns;
s <= "011";   wait for 20 ns;
s <= "100";   wait for 20 ns;
s <= "101";   wait for 20 ns;
s <= "110";   wait for 20 ns;
s <= "111";   wait for 20 ns;
wait;
end process;
end demux1_8;
```

Output wave:-


EDA playground
Run
Copy
ASU students: please log on using the Google button.
EPWave
From: 0ps
To: 200,000ps
Get Signals
Radix
100%
0
20,000
40,000
60,000
80,000
i
s[2:0]
y[7:0]
0
1
10
11
100
101
110
111
*xxxxx1
*xxxx11
*xxx111
*xx1111
*x11111
*111111
*111111
11111111
Note: To revert to EPWave opening in a new browser window, set that option on your user page.