INSTITUTE OF ENGINEERING AND TECHNOLOGY, INDORE

CIRCUIT DESIGN USING HDL (EIR7C4)

(Lab Assignment)

Session (2020-21)

Submitted to:
Dr. Vaibhav Neema

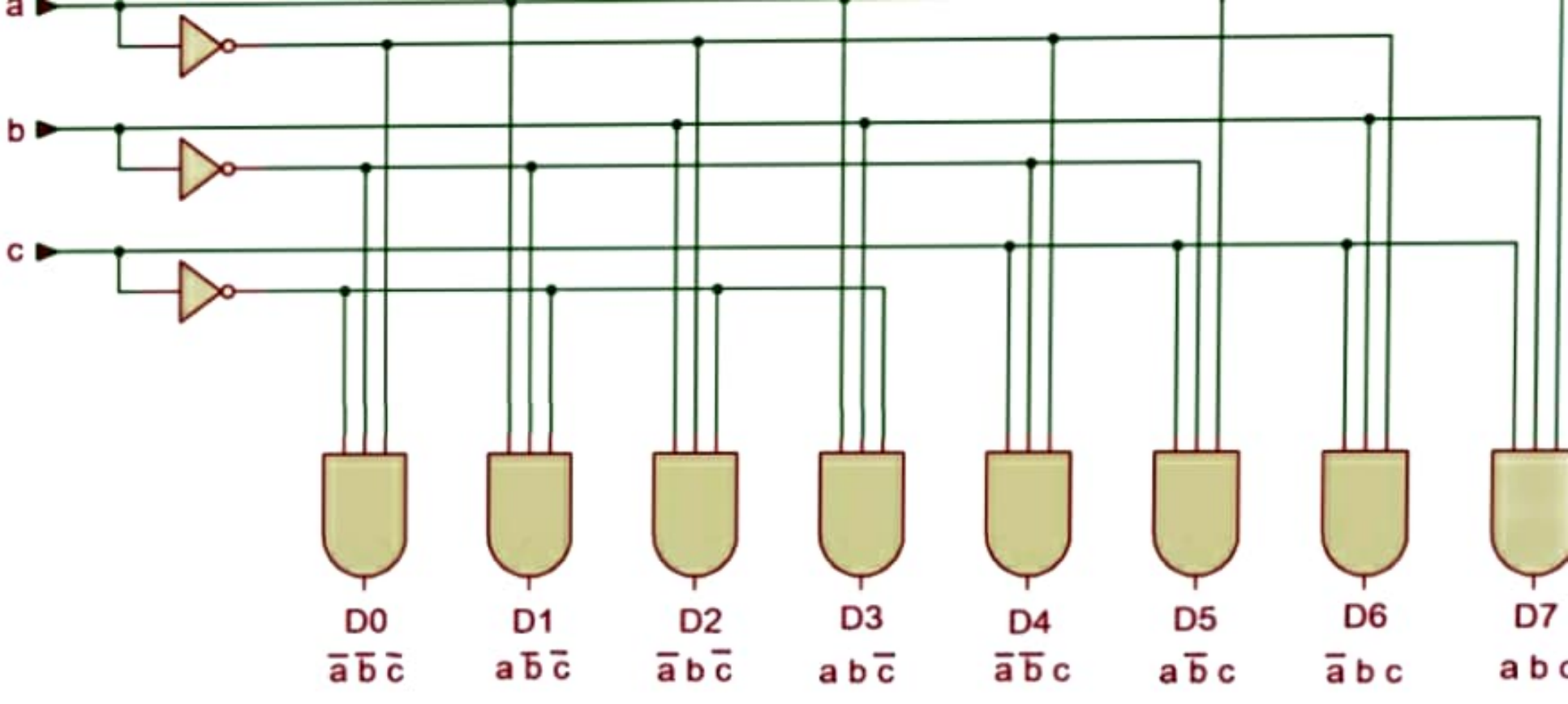
Submitted by
Name:-
E&I (7th Sem)

Roll no. – 17E7039

Enroll no.- DE17230

Truth Table

c	b	a	D0	D1	D2	D3	D4	D5	D6	D7
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


a
b
c
D0
$\bar{a}\,\bar{b}\,\bar{c}$
D1
$a\,\bar{b}\,\bar{c}$
D2
$\bar{a}\,b\,\bar{c}$
D3
$a\,b\,\bar{c}$
D4
$\bar{a}\,\bar{b}\,c$
D5
$a\,\bar{b}\,c$
D6
$\bar{a}\,b\,c$
D7
a b c

testbench.vhd

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity testbench is
end testbench;
architecture behavior of testbench is
--signal declarations.
signal input : std_logic_vector(2 downto 0) :=(others => '0');
signal output :  std_logic_vector(7 downto 0) :=(others => '0');
begin
--entity instantiation
UUT : entity work.decoder port map(input,output);
--definition of simulation process
tb : process
begin
input<="000";  --input = 0.
wait for 2 ns;
input<="001";   --input = 1.
wait for 2 ns;
input<="010";   --input = 2.
wait for 2 ns;
input<="011";   --input = 3.
wait for 2 ns;
input<="100";   --input = 4.
wait for 2 ns;
input<="101";   --input = 5.
wait for 2 ns;
input<="110";   --input = 6.
wait for 2 ns;
input<="111";   --input = 7.
wait;
end process tb;
end;
```

design.vhd

VHDL Design

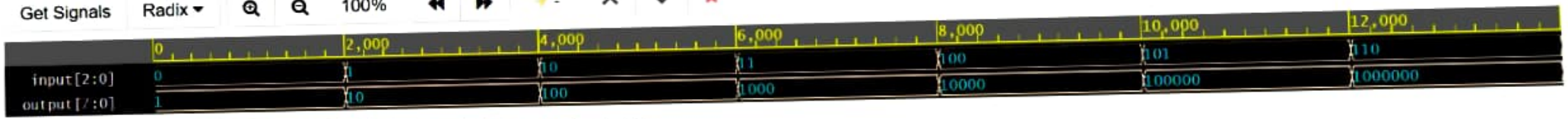
```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

--entity declaration with port definitions
entity decoder is
port(    input :        in std_logic_vector(2 downto 0);  --3 bit input
            output : out std_logic_vector(7 downto 0)  -- 8 bit ouput
   );
end decoder;
--architecture of entity
architecture Behavioral of decoder is

begin
output(0) <= (not input(2)) and (not input(1)) and (not input(0));
output(1) <= (not input(2)) and (not input(1)) and input(0);
output(2) <= (not input(2)) and input(1) and (not input(0));
output(3) <= (not input(2)) and input(1) and input(0);
output(4) <= input(2) and (not input(1)) and (not input(0));
output(5) <= input(2) and (not input(1)) and input(0);
output(6) <= input(2) and input(1) and (not input(0));
output(7) <= input(2) and input(1) and input(0);

end Behavioral;
```

EPWave

EPWave

From: 0ps To: 14,000ps

Get Signals Radix 100%



Note: To revert to EPWave opening in a new browser window, set that option on your user page.