INSTITUTE OF ENGINEERING AND TECHNOLOGY, INDORE

ELECTRONICS & INSTRUMENTATION ENGINEERING

CIRCUIT DESIGN USING HDL (EIR7E1)

LAB VIVA

Session: 2020-21

SUBMITTED TO

VAIBHAV NEEMA

SUBMITTED BY

PRANAY GUPTA

ROLL NO: 16E7036

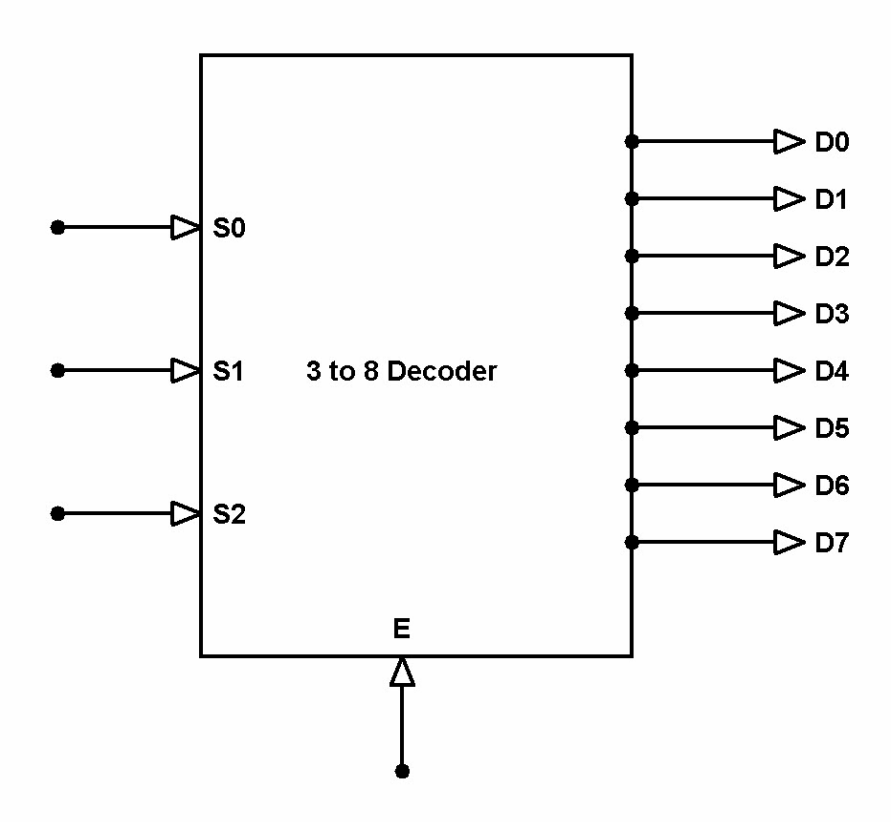
ENROLL: DE16232

MOB: 9993845384

Designing of 3-Line to 8-Line Decoder:-

A decoder is a combinational logic circuit which is used to change the code into a set of signals. It is the reverse process of an encoder. A decoder circuit takes multiple inputs and gives multiple outputs. A decoder circuit takes binary data of 'n' inputs into '2^n' unique output. In addition to input pins, the decoder has a enable pin. This enables the pin when negated, makes the circuit inactive.

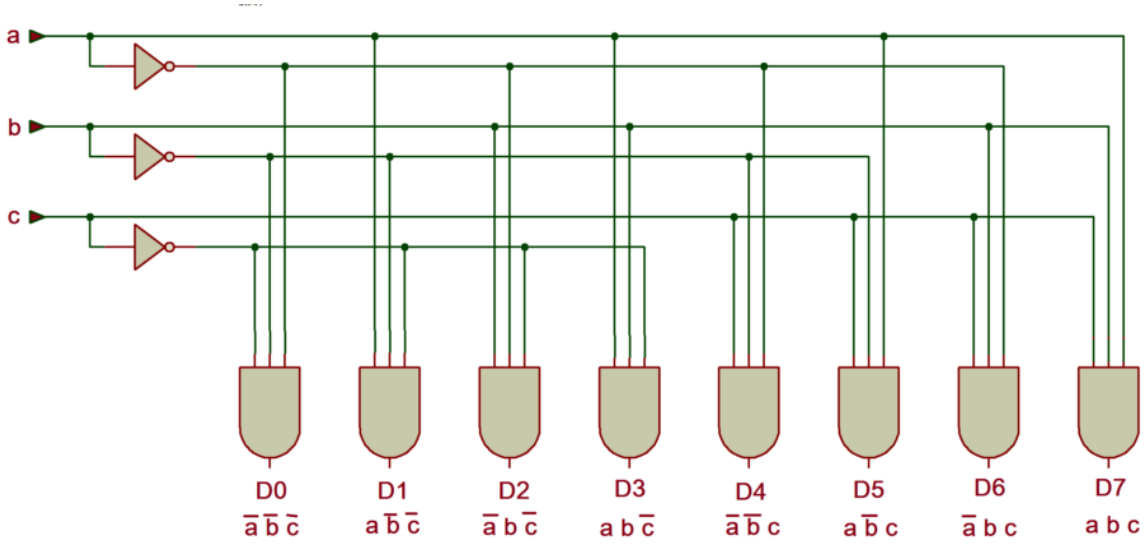
This decoder circuit gives 8 logic outputs for 3 inputs and has a enable pin. The circuit is designed with AND and NAND logic gates. It takes 3 binary inputs and activates one of the eight outputs. 3 to 8 line decoder circuit is also called as binary to an octal decoder.



3 to 8 Line Decoder Block Diagram

The decoder circuit works only when the Enable pin (E) is high. S0, S1 and S2 are three different inputs and D0, D1, D2, D3. D4. D5. D6. D7 are the eight outputs.

3×8 Decoder circuit :-



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

3 TO 8 DECODER CODE :-

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH.all;
use IEEE.STD_LOGIC_UNSIGNED.all;
entity not1 is
port( i1 : in std_logic;
      o1 : out std_logic);
 end not1;
architecture strucnot1 of not1 is
 begin
   o1 <= not i1;
end strucnot1;


library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH.all;
use IEEE.STD_LOGIC_UNSIGNED.all;
entity and3 is
 port(i2 : in std_logic;
      i3 : in std_logic;
      i4 : in std_logic;
      o2 : out std_logic);
 end and3;
```

```
architecture strucand3 of and3 is
 begin
  o2 <= i2 and i3 and i4;
 end strucand3;

 library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH.all;
use IEEE.STD_LOGIC_UNSIGNED.all;
entity decoder_3x8 is
 port(a : in std_logic_vector(2 downto 0);
     d : out std_logic_vector(7 downto 0));
end decoder_3x8;
architecture decoder_arch of decoder_3x8 is
 component not1
       port(i1:in std_logic;
            o1:out std_logic);
 end component;
 component and3
       port(i2,i3,i4:in std_logic;
            o2:out std_logic);
 end component;

signal int1 : std_logic;
signal int2 : std_logic;
```

```
signal int3 : std_logic;


begin
  c1 : not1 port map(a(0),int1);
  c2 : not1 port map(a(1),int2);
  c3 : not1 port map(a(2),int3);
  c4 : and3 port map(int1,int2,int3,d(0));
  c5 : and3 port map(int1,int2,a(2),d(1));
  c6 : and3 port map(int1,a(1),int3,d(2));
  c7 : and3 port map(int1,a(1),a(2),d(3));
  c8 : and3 port map(a(0),int2,int3,d(4));
  c9 : and3 port map(a(0),int2,a(2),d(5));
  c10: and3 port map(a(0),a(1),int3,d(6));
  c11: and3 port map(a(0),a(1),a(2),d(7));
end decoder_arch;
```

TESTBENCH CODE :-

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
ENTITY tb_decoder IS
END tb_decoder;
ARCHITECTURE structure OF tb_decoder IS
-- Component Declaration for the Unit Under Test (UUT)
COMPONENT decoder_3x8 is
PORT(
        a : IN std_logic_vector(2 downto 0);
        d : OUT std_logic_vector(7 downto 0));
END COMPONENT;
       --Inputs
        signal a : std_logic_vector(2 downto 0);
        --Outputs
        signal d : std_logic_vector(7 downto 0);
        --appropriate port name
        BEGIN
        --Instantiate the Unit Under Test (UUT)
        uut: decoder_3x8 PORT MAP (
        a => a,
        d => d);
        --Stimulus process
        stim_proc: process
```

```
        begin
        --hold reset state
        wait for 100 ns;
        a <= "000";
        wait for 100 ns;
        a <= "001";
        wait for 100 ns;
        a <= "010";
        wait for 100 ns;
        a <= "011";
wait for 100 ns;
a <= "100";
        wait for 100 ns;
        a <= "101";
        wait for 100 ns;
        a <= "110";
        wait for 100 ns;
        a <= "111";
        wait for 100 ns;
        end process;
        END structure;
```

EP WAVE IS :-