INSTITUTE OF ENGINEERING AND TECHNOLOGY

DAVV ,INDORE

CIRCUIT DESIGN USING HDL (EIR7C4)

(Lab Assignment)

Session (2020-21)

Submitted to:-

Dr. Vaibhav Nema

Submitted by :-

Manjeet Narwariya

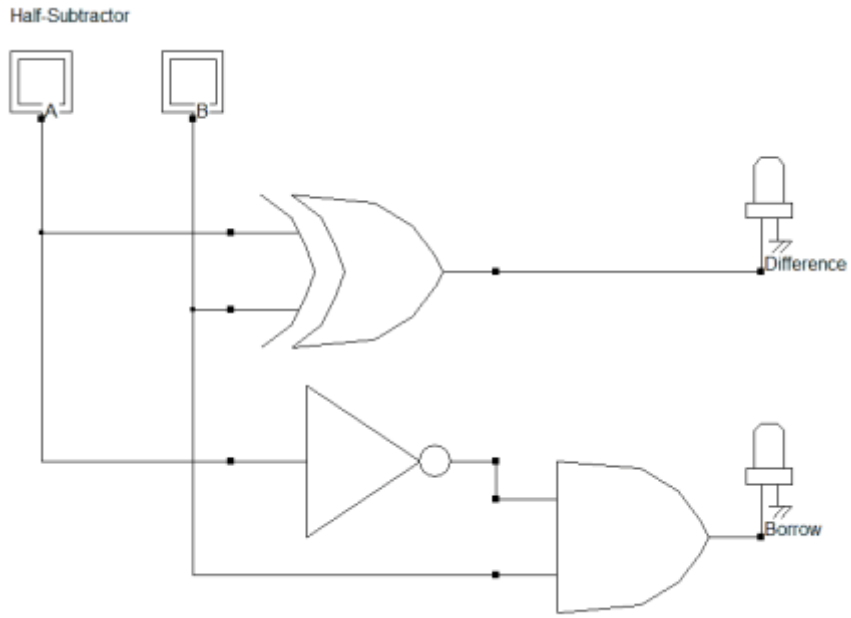
E&I (7th Sem)

Roll no – 17E7024

Enroll no. – DE17216

URL to see my Practical Code and wave - https://www.edaplayground.com/x/b6cL

Q- Design of 2 bit Subtractor using Behavioral Method?



Truth table of the half Subtractor-

A	B	DIFFERENCE	BORROW
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	0

TEST BENCH-

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity half_sub_tb is
end entity;
architecture tb of half_sub_tb is
component HS is
Port ( A : in STD_LOGIC;
B : in STD_LOGIC;
Diff : out STD_LOGIC;
Borrow : out STD_LOGIC);
end component;
signal A,B: STD_LOGIC;
signal Diff : STD_LOGIC;
signal Borrow : STD_LOGIC;
begin
uut: HS port map(
A => A, Diff => Diff,B => B,Borrow => Borrow);
stim: process
begin
    A <= '0'; B <= '0';
```

```
      wait for 20 ns;
    A <= '1'; B <= '0';
      wait for 20 ns;
    A <= '0'; B <= '1';
       wait for 20 ns;
    A <= '1'; B <= '1';
        wait for 20 ns;
end process;
end tb;
```

DESIGN CODE-

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity HS is
Port (
A : in STD_LOGIC;
B : in STD_LOGIC;
Diff : out STD_LOGIC;
Borrow : out STD_LOGIC
);
end HS;
architecture Behavioral of HS is
begin
process(A,B)
```

```
begin
if (A ='0' and B='0') THEN
Diff <= '0';
Borrow <= '0';
elsif (A = '0' and B='1') THEN
Diff <= '1';
Borrow <= '1';
elsif (A ='1' and B='0') THEN
Diff <= '1';
Borrow <= '0';
elsif (A = '1' and B='1') THEN
Diff <= '0';
Borrow <= '0';
else
Diff <= 'Z';
Borrow <='Z';
end if;
end process;
end Behavioral;
```

EDA Playground Test bench-

testbench.sv

SV/Verilog Testbench

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity half_sub_tb is
end entity;
architecture tb of half_sub_tb is
component HS is
Port ( A : in STD_LOGIC;
B : in STD_LOGIC;
Diff : out STD_LOGIC;
Borrow : out STD_LOGIC);
end component;
signal A,B: STD_LOGIC;
signal Diff : STD_LOGIC;
signal Borrow : STD_LOGIC;
begin
uut: HS port map(
A => A, Diff => Diff,B => B,Borrow => Borrow);
stim: process
begin
    A <= '0'; B <= '0';
      wait for 20 ns;
    A <= '1'; B <= '0';
      wait for 20 ns;
    A <= '0'; B <= '1';
        wait for 20 ns;
    A <= '1'; B <= '1';
         wait for 20 ns;
end process;
end tb;
```

Design code -

design.sv

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity HS is
Port (
A : in STD_LOGIC;
B : in STD_LOGIC;
Diff : out STD_LOGIC;
Borrow : out STD_LOGIC
);
end HS;
architecture Behavioral of HS is
begin
process(A,B)
begin
if (A ='0' and B='0') THEN
Diff <= '0';
Borrow <= '0';
elsif (A = '0' and B='1') THEN
Diff <= '1';
Borrow <= '1';
elsif (A ='1' and B='0') THEN
Diff <= '1';
Borrow <= '0';
elsif (A = '1' and B='1') THEN
Diff <= '0';
Borrow <= '0';
else
Diff <= 'Z';
Borrow <='Z';
end if;
end process;
end Behavioral;
```

EP Waveform-