INSTITUTE OF ENGINEERING & TECHNOLOGY

DEVI AHILYA VISHWAVIDHYALAYA , INDORE

ELECTRONIC AND INSTRUMENTATION ENGINEERING

SUBMITTED BY : DIVYANSHI KAJLE

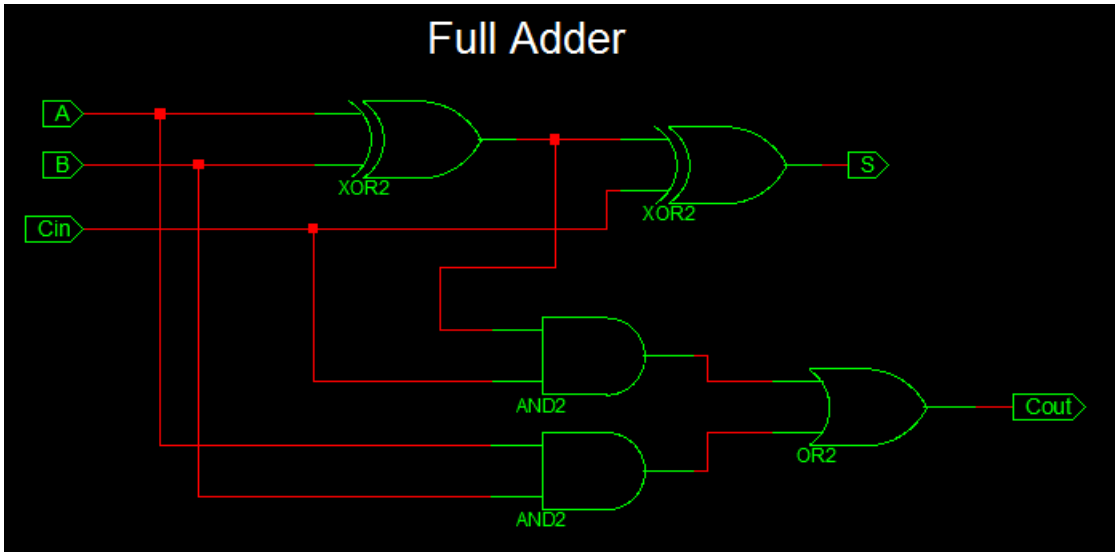
ROLL NUMBER : 17E7016

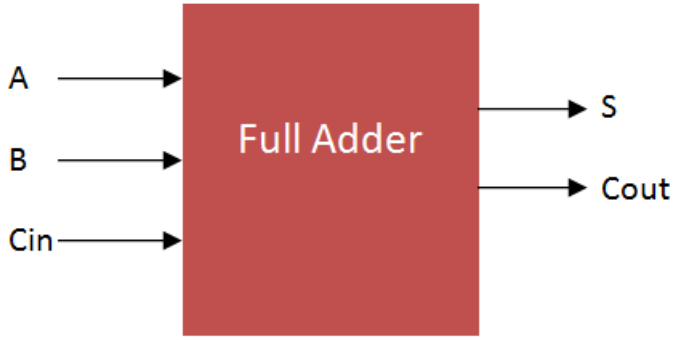
ENROLLEMENT NO : DE17208

CLASS : BE IV YEAR

1 BIT FULL ADDER USING STRUCTURAL MODELLING

The VHDL Code for full-adder circuit adds three one-bit binary numbers (A, B, Cin) and outputs two one-bit binary numbers, a sum (S) and a carry (Cout). Truth Table describes the functionality of full adder. sum(S) output is High when odd number of inputs are High. Cout is High, when two or more inputs are High. VHDL Code for full adder can also be constructed with 2 half adder Port mapping in to full adder.





Cin	B	A	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

VHDL CODE FOR 1 BIT FULL ADDER

```
library IEEE;
use IEEE.std_logic_1164.all;

entity fulladder is
 Port ( A : in STD_LOGIC;
 B : in STD_LOGIC;
 Cin : in STD_LOGIC;
 S : out STD_LOGIC;
 Cout : out STD_LOGIC);
end fulladder;
```

```
architecture structure of fulladder is

signal x1,x2,x3: std_logic;

begin

x1<= A xor B;

x2<= A and B;

x3<= x1 and Cin;

Cout <= x2 or x3;

S<= x1 xor Cin;

end structure;
```

TESTBENCH FOR 1 BIT FULL ADDER

```
library IEEE;

use IEEE.std_logic_1164.all;


ENTITY fulladd IS

END fulladd;


ARCHITECTURE behavior OF fulladd IS


 COMPONENT fulladder
```

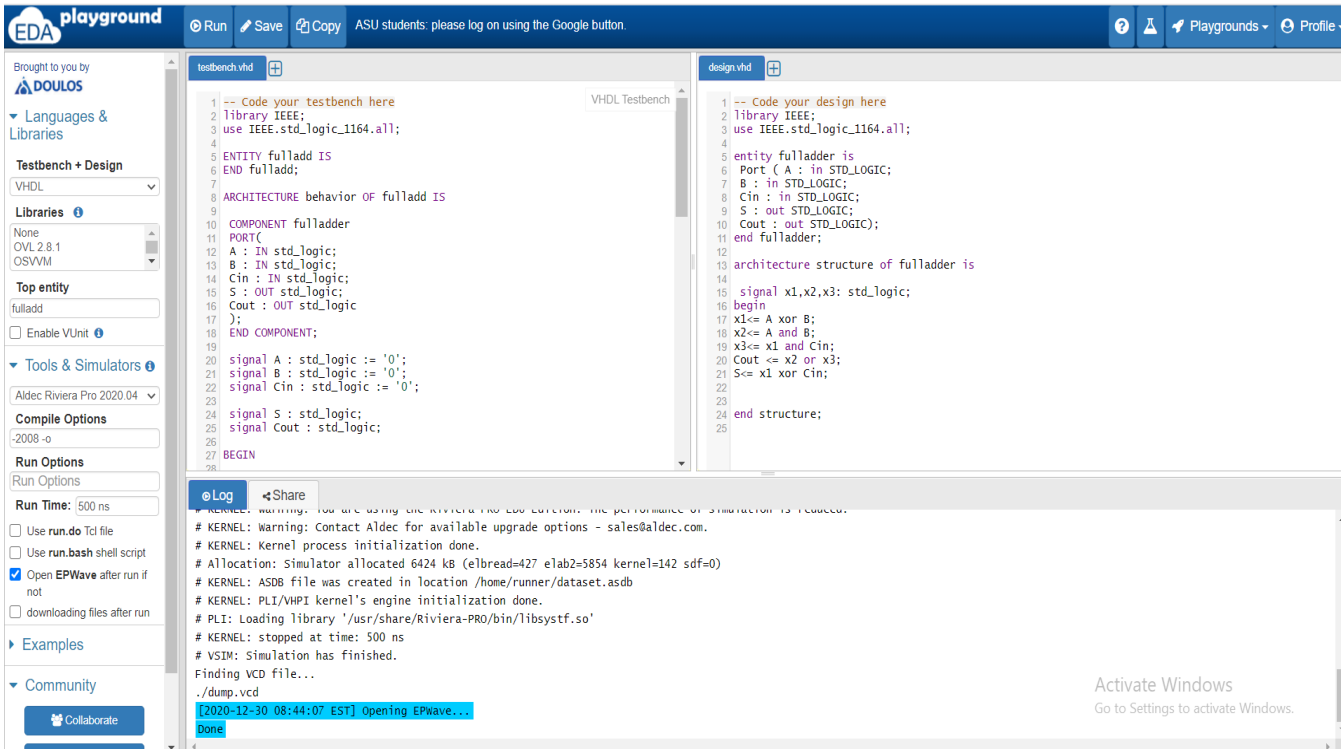
```
PORT(
A : IN std_logic;
B : IN std_logic;
Cin : IN std_logic;
S : OUT std_logic;
Cout : OUT std_logic
);
END COMPONENT;

signal A : std_logic := '0';
signal B : std_logic := '0';
signal Cin : std_logic := '0';

signal S : std_logic;
signal Cout : std_logic;

BEGIN

uut: fulladder PORT MAP (
A => A,
B => B,
Cin => Cin,
```

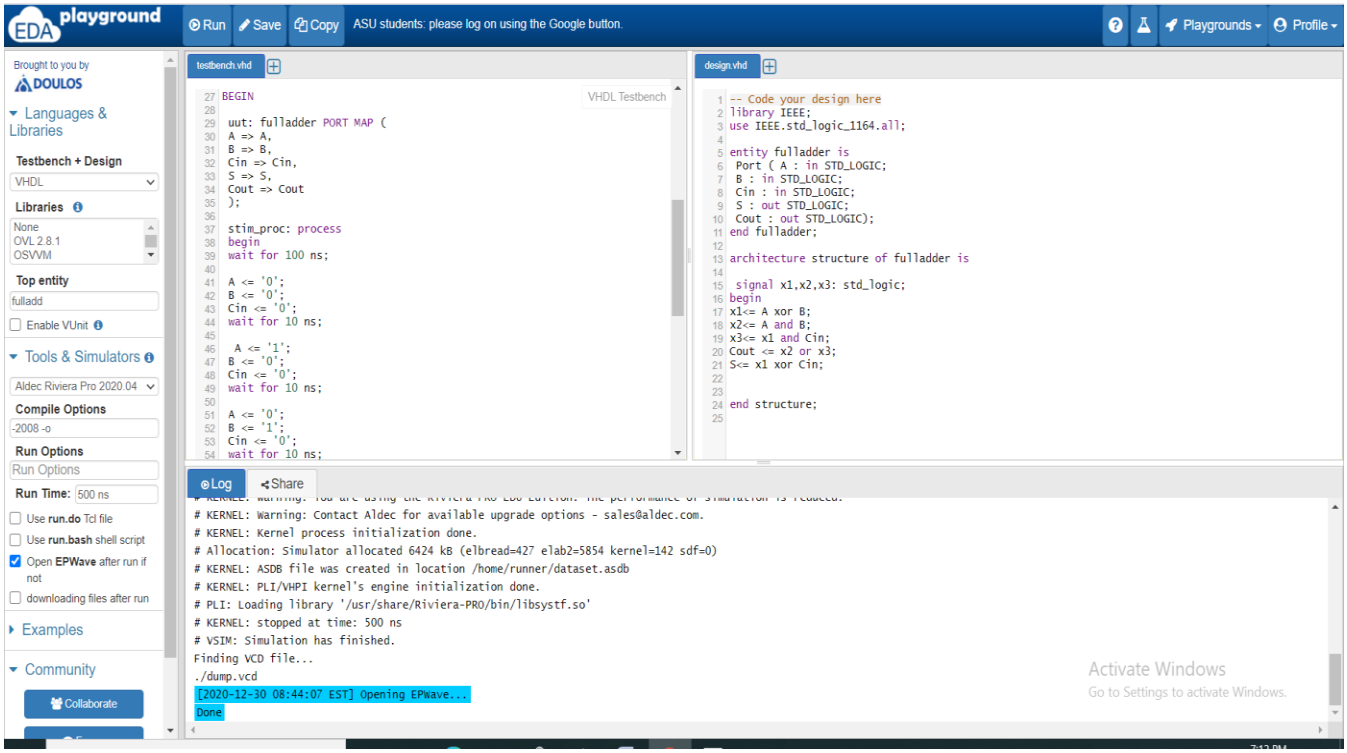
```
S => S,
Cout => Cout
);

stim_proc: process
begin
wait for 100 ns;

A <= '0';
B <= '0';
Cin <= '0';
wait for 10 ns;

A <= '1';
B <= '0';
Cin <= '0';
wait for 10 ns;

A <= '0';
B <= '1';
Cin <= '0';
wait for 10 ns;
```

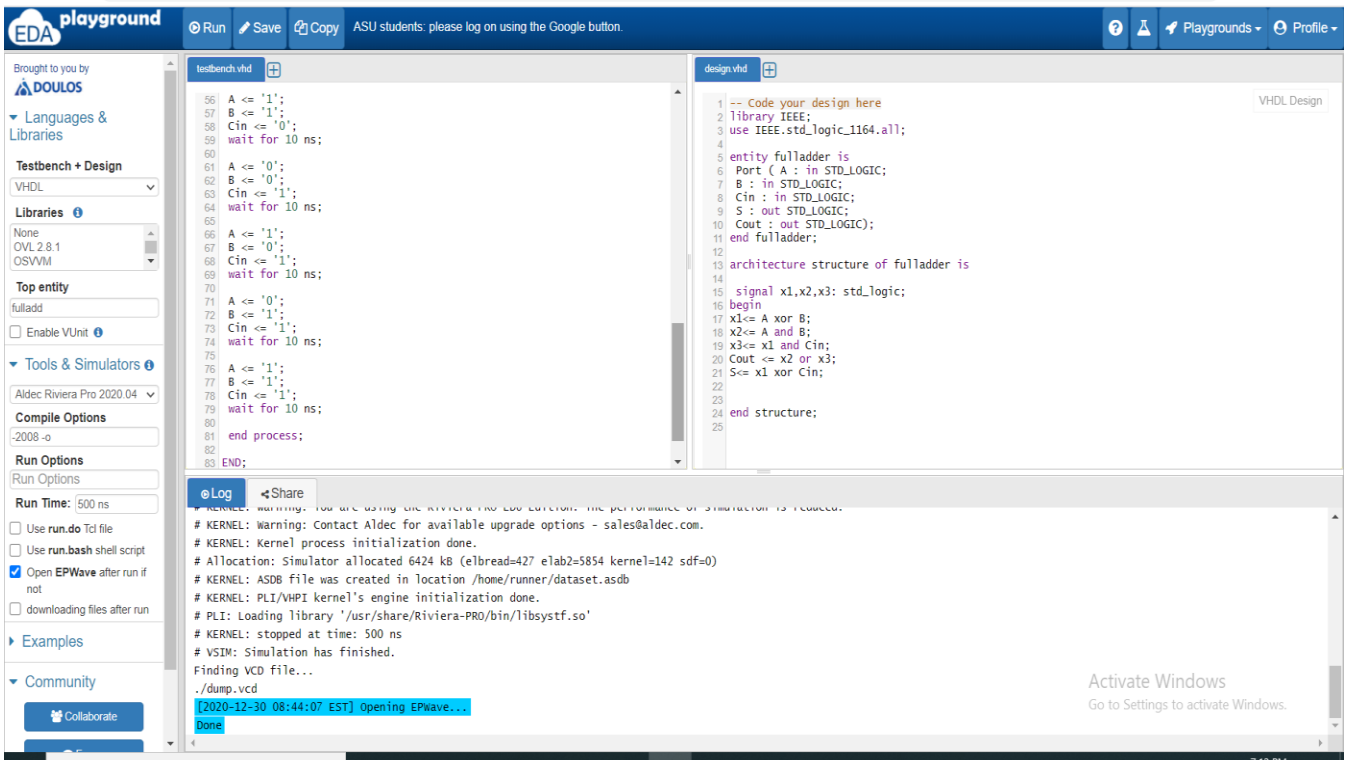
```
A <= '1';
B <= '1';
Cin <= '0';
wait for 10 ns;

A <= '0';
B <= '0';
Cin <= '1';
wait for 10 ns;

A <= '1';
B <= '0';
Cin <= '1';
wait for 10 ns;

A <= '0';
B <= '1';
Cin <= '1';
wait for 10 ns;

A <= '1';
```

```
B <= '1';

Cin <= '1';

wait for 10 ns;


end process;

END;
```

VHDL CODE and TESTBENCH :




EDA playground
Run
Save
Copy
ASU students: please log on using the Google button.
Playgrounds
Profile
Brought to you by
DOULOS
Languages & Libraries
Testbench + Design
VHDL
Libraries
None
OVL 2.8.1
OSVVM
Top entity
fulladd
Enable VUnit
Tools & Simulators
Aldec Riviera Pro 2020.04
Compile Options
-2008 -o
Run Options
Run Options
Run Time: 500 ns
Use run.do Tcl file
Use run.bash shell script
Open EPWave after run if not
downloading files after run
Examples
Community
Collaborate
testbench.vhd
VHDL Testbench
27 BEGIN
29 uut: fulladder PORT MAP (
30 A => A,
31 B => B,
32 Cin => Cin,
33 S => S,
34 Cout => Cout
35);
37 stim_proc: process
38 begin
39 wait for 100 ns;
41 A <= '0';
42 B <= '0';
43 Cin <= '0';
44 wait for 10 ns;
46 A <= '1';
47 B <= '0';
48 Cin <= '0';
49 wait for 10 ns;
51 A <= '0';
52 B <= '1';
53 Cin <= '0';
54 wait for 10 ns;
design.vhd
1 -- Code your design here
2 library IEEE;
3 use IEEE.std_logic_1164.all;
5 entity fulladder is
6 Port (A : in STD_LOGIC;
7 B : in STD_LOGIC;
8 Cin : in STD_LOGIC;
9 S : out STD_LOGIC;
10 Cout : out STD_LOGIC);
11 end fulladder;
13 architecture structure of fulladder is
15 signal x1,x2,x3: std_logic;
16 begin
17 x1<= A xor B;
18 x2<= A and B;
19 x3<= x1 and Cin;
20 Cout <= x2 or x3;
21 S<= x1 xor Cin;
24 end structure;
Log
Share
KERNEL: Warning: Contact Aldec for available upgrade options - sales@aldec.com.
KERNEL: Kernel process initialization done.
Allocation: Simulator allocated 6424 kB (elbread=427 elab2=5854 kernel=142 sdf=0)
KERNEL: ASDB file was created in location /home/runner/dataset.asdb
KERNEL: PLI/VHPI kernel's engine initialization done.
PLI: Loading library '/usr/share/Riviera-PRO/bin/libsystf.so'
KERNEL: stopped at time: 500 ns
VSIM: Simulation has finished.
Finding VCD file...
./dump.vcd
[2020-12-30 08:44:07 EST] Opening EPWave...
Done
Activate Windows
Go to Settings to activate Windows.


EDA playground
Run
Save
Copy
ASU students: please log on using the Google button.
Playgrounds
Profile
Brought to you by
DOULOS
Languages & Libraries
Testbench + Design
VHDL
Libraries
None
OVL 2.8.1
OSVVM
Top entity
fulladd
Enable VUnit
Tools & Simulators
Aldec Riviera Pro 2020.04
Compile Options
-2008 -o
Run Options
Run Options
Run Time: 500 ns
Use run.do Tcl file
Use run.bash shell script
Open EPWave after run if not
downloading files after run
Examples
Community
Collaborate
testbench.vhd
56 A <= '1';
57 B <= '1';
58 Cin <= '0';
59 wait for 10 ns;
61 A <= '0';
62 B <= '0';
63 Cin <= '1';
64 wait for 10 ns;
66 A <= '1';
67 B <= '0';
68 Cin <= '1';
69 wait for 10 ns;
71 A <= '0';
72 B <= '1';
73 Cin <= '1';
74 wait for 10 ns;
76 A <= '1';
77 B <= '1';
78 Cin <= '1';
79 wait for 10 ns;
81 end process;
83 END;
design.vhd
VHDL Design
1 -- Code your design here
2 library IEEE;
3 use IEEE.std_logic_1164.all;
5 entity fulladder is
6 Port (A : in STD_LOGIC;
7 B : in STD_LOGIC;
8 Cin : in STD_LOGIC;
9 S : out STD_LOGIC;
10 Cout : out STD_LOGIC);
11 end fulladder;
13 architecture structure of fulladder is
15 signal x1,x2,x3: std_logic;
16 begin
17 x1<= A xor B;
18 x2<= A and B;
19 x3<= x1 and Cin;
20 Cout <= x2 or x3;
21 S<= x1 xor Cin;
24 end structure;
Log
Share
KERNEL: Warning: Contact Aldec for available upgrade options - sales@aldec.com.
KERNEL: Kernel process initialization done.
Allocation: Simulator allocated 6424 kB (elbread=427 elab2=5854 kernel=142 sdf=0)
KERNEL: ASDB file was created in location /home/runner/dataset.asdb
KERNEL: PLI/VHPI kernel's engine initialization done.
PLI: Loading library '/usr/share/Riviera-PRO/bin/libsystf.so'
KERNEL: stopped at time: 500 ns
VSIM: Simulation has finished.
Finding VCD file...
./dump.vcd
[2020-12-30 08:44:07 EST] Opening EPWave...
Done
Activate Windows
Go to Settings to activate Windows.

OUTPUT WAVEFORM: