POORNIMA

COLLEGE OF ENGINEERING

Session 2017-2018

Branch: Computer Science & Engineering

Name of the Lab: Digital Hardware Design Lab

Year: 3 Yr. /V SEM

Lab Code: - 5CS10

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

LAB RULES

Responsibilities of Users

Users are expected to follow some fairly obvious rules of conduct:

Always:

- Enter the lab on time and leave at proper time.
- Wait for the previous class to leave before the next class enters.
- Keep the bag outside in the respective racks.
- Utilize lab hours in the corresponding.
- Turn off the machine before leaving the lab unless a member of lab staff has specifically told you not to do so.
- Leave the labs at least as nice as you found them.
- If you notice a problem with a piece of equipment (e.g. a computer doesn't respond) or the room in general (e.g. cooling, heating, lighting) please report it to lab staff immediately. Do not attempt to fix the problem yourself.

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

Never:

- Don't abuse the equipment.
- Do not adjust the heat or air conditioners. If you feel the temperature is not properly set, inform lab staff; we will attempt to maintain a balance that is healthy for people and machines.
- Do not attempt to reboot a computer. Report problems to lab staff.
- Do not remove or modify any software or file without permission.
- Do not remove printers and machines from the network without being explicitly told to do so by lab staff.
- Don't monopolize equipment. If you're going to be away from your machine for more than 10 or 15 minutes, log out before leaving. This is both for the security of your account, and to ensure that others are able to use the lab resources while you are not.
- Don’t use internet, internet chat of any kind in your regular lab schedule.
- Do not download or upload of MP3, JPG or MPEG files.
- No games are allowed in the lab sessions.
- No hardware including USB drives can be connected or disconnected in the labs without prior permission of the lab in-charge.
- No food or drink is allowed in the lab or near any of the equipment. Aside from the fact that it leaves a mess and attracts pests, spilling anything on a keyboard or other piece of computer equipment could cause permanent, irreparable, and costly damage. (and in fact *has*) If you need to eat or drink, take a break and do so in the canteen.
- Don’t bring any external material in the lab, except your lab record, copy and books.
- Don’t bring the mobile phones in the lab. If necessary then keep them in silence mode.
- Please be considerate of those around you, especially in terms of noise level. While labs are a natural place for conversations of all types, kindly keep the volume turned down.

- If you are having problems or questions, please go to either the faculty, lab in-charge or the lab supporting staff. They will help you. We need your full support and cooperation for smooth functioning of the lab.

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

INSTRUCTIONS

Before entering in the lab

- All the students are supposed to prepare the theory regarding the next experiment/ Program.
- Students are supposed to bring their lab records as per their lab schedule.
- Previous experiment/program should be written in the lab record.
- If applicable trace paper/graph paper must be pasted in lab record with proper labeling.
- All the students must follow the instructions, failing which he/she may not be allowed in the lab.

While working in the lab

- Adhere to experimental schedule as instructed by the lab in-charge/faculty.
- Get the previously performed experiment/ program signed by the faculty/ lab in charge.
- Get the output of current experiment/program checked by the faculty/ lab in charge in the lab copy.
- Each student should work on his/her assigned computer at each turn of the lab.
- Take responsibility of valuable accessories.

LAB MANUAL

Lab Name : Digital Hardware Design Lab

Lab Code : 5CS10

Branch : Computer Engineering

Year : 3rd Year

INDEX

	• Simulate a digital data path in Hardware Description Language • Understand IC descriptions and select proper GATES in a given circuit based on its timing characteristics 1.2 Objective:- Write VHDL code for AND gate 1.3 Objective:- Write VHDL code for OR gate	
	Sample Viva Question	
Exp:- 2	Objectives :- Write VHDL coding for 4*4 Multiplier	
	Sample Viva Question	
Exp:- 3	Objectives :- Write VHDL coding for Decoder.	
	Sample Viva Question	
Exp:-4	Objectives :- Combinational Multiplexer Perform and draw the circuit diagram and should have knowledge about Concept and Truth Table for the following:-- 4.1 Objective:- Write VHDL code for 4*1 Multiplexer 4.2 Objective:- Write VHDL code for 1*4 De-multiplexer	
	Sample Viva Question	
Exp:- 5	Objective:- Write VHDL code for Design and simulation of a 4-bit Adder	
	Sample Viva Question	
Exp:- 6	Perform and draw the RTL and Technology circuit for the 6.1 Objective:- Write VHDL code for Half Adder 6.2 Objective:- Write VHDL code for Full Adder	
	Sample Viva Question	
	LIST OF EXPERIMENTS (BEYOND SYLLABUS)	
Exp:- 7	OBJECT: Simulation of SR Flip-Flop using VHDL code through Xilinx**.**	
	Sample Viva Question	
Exp:- 8	Objectives :- Simulation of Full sub tractor using VHDL code through Xilinx**.**	
	Sample Viva Question	

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

Experiment 1.1

Objective:- To study about VHDL.

THEORY: HDL stands for very high-speed integrated circuit hardware description language. Which is one of the programming language used to model a digital system by dataflow, behavioral and structural style of modeling. This language was first introduced in 1981 for the department of Defense (DoD) under the VHSIC programe. In 1983 IBM, Texas instruments and Intermetrics started to develop this language. In 1985 VHDL 7.2 version was released. In 1987 IEEE standardized the language.

Describing a design:

In VHDL an entity is used to describe a hardware module.

An entity can be described using,

1. Entity declaration.
2. Architecture.
3. Configuration
4. Package declaration.
5. Package body.

Let's see what are these?

Entity declaration:

It defines the names, input output signals and modes of a hardware module.

Syntax:

```
entity entity_name is
   Port declaration;
end entity_name;
```

An entity declaration should starts with 'entity' and ends with 'end' keywords.

Ports are interfaces through which an entity can communicate with its environment. Each port must have a name, direction and a type. An entity may have no port declaration also. The direction will be input, output or inout.

In	Port can be read
Out	Port can be written
Inout	Port can be read and written
Buffer	Port can be read and written, it can have only one source.

Architecture:

It describes the internal description of design or it tells what is there inside design. Each entity has atleast one architecture and an entity can have many architecture. Architecture can be described using structural, dataflow, behavioral or mixed style. Architecture can be used to describe a design at different levels of abstraction like gate level, register transfer level (RTL) or behavior level.

Syntax:

```
    architecture architecture_name of entity_name
   architecture_declarative_part;
begin
  Statements;
end architecture_name;
```

Here we should specify the entity name for which we are writing the architecture body. The architecture statements should be inside the begin and end keyword. Architecture declarative part may contain variables, constants, or component declaration.

Configuration:

If an entity contains many architectures and any one of the possible architecture binding with its entity is done using configuration. It is used to bind the architecture body to its entity and a component with an entity.

Syntax:

```
configuration configuration_name of entity_name is
  block_configuration;
end  configuration_name.
```

Block_configuration defines the binding of components in a block. This can be written as

```
for block_name
  component_binding;
end for;
```

block_name is the name of the architecture body. Component binding binds the components of the block to entities. This can be written as,

```
for component_labels:component_name
  block_configuration;
end for;
```

Package declaration:

Package declaration is used to declare components, types, constants, functions and so on.

Syntax:

```
    package package_name is
  Declarations;
end package_name;
```

Package body:

A package body is used to declare the definitions and procedures that are declared in corresponding package. Values can be assigned to constants declared in package in package body.

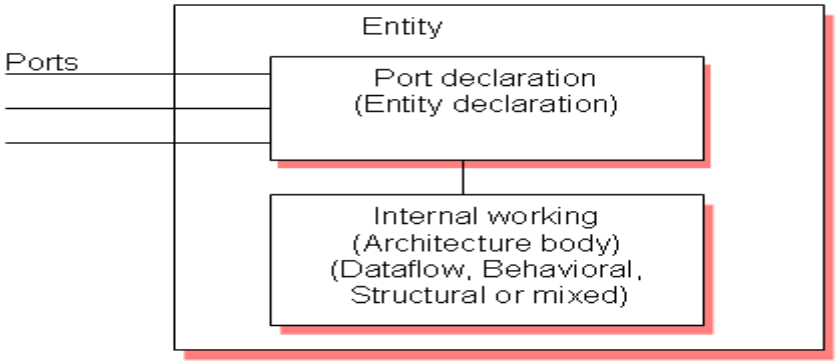
Syntax:

```
package body package_name is
    Function_procedure definitions;
end package_name;
```

The internal working of an entity can be defined using different modeling styles inside architcture body. They are

1. Dataflow modeling.
2. Behavioral modeling.
3. Structural modeling.

Structure of an entity:

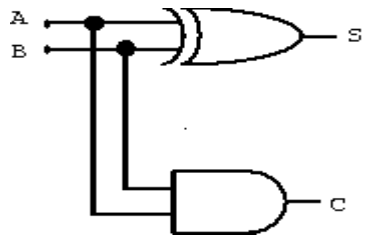


Let's try to understand with the help of one example.

Dataflow modeling:

In this style of modeling, the internal working of an entity can be implemented using concurrent signal assignment.

Let's take half adder example which is having one XOR gate and a AND gate.



```
Library IEEE;
use IEEE.STD_LOGIC_1164.all;

entity ha_en is
        port (A,B:in bit;S,C:out bit);
end ha_en;

architecture ha_ar of ha_en is
begin
        S<=A xor B;
        C<=A and B;

end ha_ar;
```

Here STD_LOGIC_1164 is an IEEE standard which defines a nine-value logic type, called STD_ULOGIC. use is a keyword, which imports all the declarations from this package. The architecture body consists of concurrent signal assignments, which describes the functionality of the design. Whenever there is a change in RHS, the expression is evaluated and the value is assigned to LHS.

Behavioral modeling:

In this style of modeling, the internal working of an entity can be implemented using set of statements.

It contains:

- Process statements
- Sequential statements
- Signal assignment statements
- Wait statements

Process statement is the primary mechanism used to model the behavior of an entity. It contains sequential statements, variable assignment (:=) statements or signal assignment (<=) statements etc. It may or may not contain sensitivity list. If there is an event occurs on any of the signals in the sensitivity list, the statements within the process are executed.

Inside the process the execution of statements will be sequential and if one entity is having two processes the execution of these processes will be concurrent. At the end it waits for another event to occur.

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity ha_beha_en is
    port(
        A : in BIT;
        B : in BIT;
        S : out BIT;
        C : out BIT
        );
end ha_beha_en;
architecture ha_beha_ar of ha_beha_en is
begin
    process_beh:process(A,B)
    begin
       S<= A xor B;
       C<=A and B;
    end process process_beh;
 end ha_beha_ar;
```

Here whenever there is a change in the value of a or b the process statements are executed.

Structural modeling:

The implementation of an entity is done through set of interconnected components.

It contains:

- Signal declaration.

- Component instances
- Port maps.
- Wait statements.

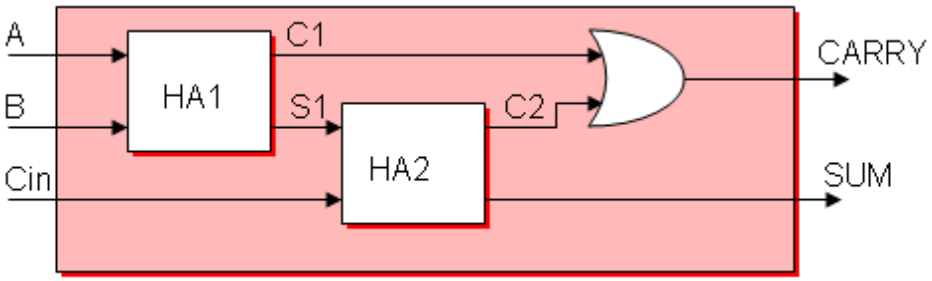
Component declaration:

Syntax:

```
component component_name [is]
  List_of_interface ports;
end component component_name;
```

Before instantiating the component it should be declared using component declaration as shown above. Component declaration declares the name of the entity and interface of a component.

Let's try to understand this by taking the example of full adder using 2 half adder and 1 OR gate.




```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity fa_en is
   port(A,B,Cin:in bit; SUM, CARRY:out bit);
end fa_en;
architecture fa_ar of fa_en is
component ha_en
   port(A,B:in bit;S,C:out bit);
end component;
```

```
signal C1,C2,S1:bit;
begin
   HA1:ha_en port map(A,B,S1,C1);
   HA2:ha_en port map(S1,Cin,SUM,C2);
   CARRY <= C1 or C2;
end fa_ar;
```

The program we have written for half adder in dataflow modeling is instantiated as shown above. ha_en is the name of the entity in dataflow modeling. C1, C2, S1 are the signals used for internal connections of the component which are declared using the keyword signal. Port map is used to connect different components as well as connect components to ports of the entity.

Component instantiation is done as follows.

Component_label: component_name port map (signal_list);

Signal_list is the architecture signals which we are connecting to component ports. This can be done in different ways. What we declared above is positional binding. One more type is the named binding. The above can be written as,

HA1:ha_en port map(A => A,B => B, S => S1 ,C => C1);

HA2:ha_en port map(A => S1,B => Cin, S=> SUM, C => C2);

Test bench:

The correctness of the above program can be checked by writing the test bench.

The test bench is used for generating stimulus for the entity under test. Let's write a simple test bench for full adder.

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity tb_en is
end tb_en;
architecture tb_ar of tb_en is
signal a_i,b_i,c_i,sum_i,carry_i:bit;
begin
  eut: entity work.fa_en(fa_ar)
```

```
        port map(A=>a_i,B=>b_i,Cin=>c_i,SUM=>sum_i,CARRY=>carry_i);
stimulus: process
            begin
               a_i<='1';b_i<='1';c_i<='1';
               wait for 10ns;
               a_i<='0';b_i<='1';c_i<='1';
               wait for 10ns;
               a_i<='1';b_i<='0';c_i<='0';
               wait for 10ns;
               if now=30ns then
                  wait;
               end if;
end process stimulus;
end tb_ar;
```

Here now is a predefined function that returns the current simulation time

What we saw upto this is component instantiation by positional and by name. In this test bench example the entity is directly instantiated. The direct entity instantiation syntax is:

Component_label: entity entity_name (architecture_name)

port map(signal_list);

Experiment No:- 1.2

Objective: Write a VHDL code for AND gate

APPARATUS: Xilinx ISE Design Suite 12.1

TRUTH TABLE

a	b	c=a.b
0	0	0
0	1	0
1	0	0
1	1	1

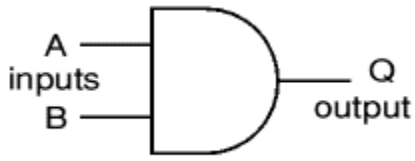
VHDL Module Code:

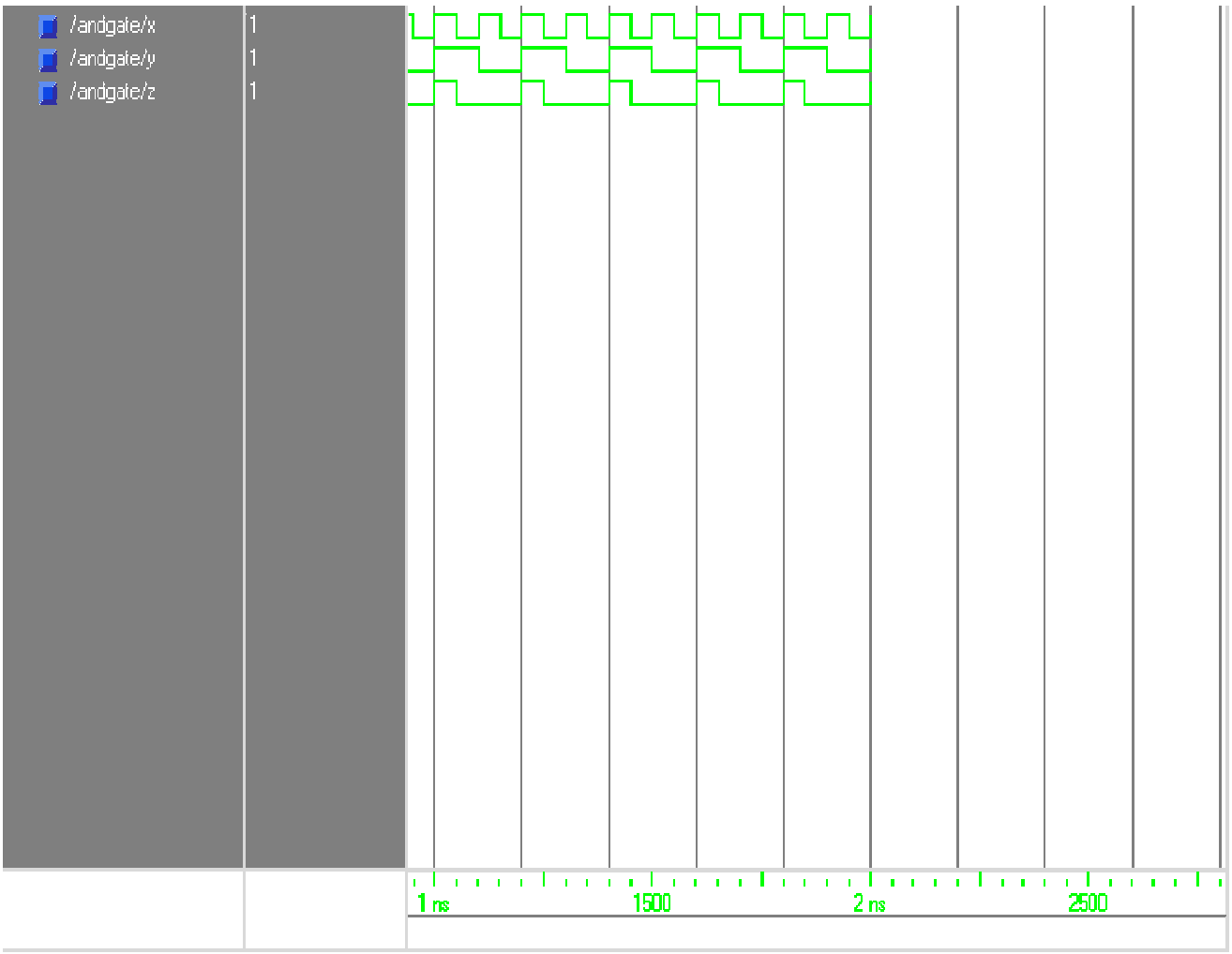
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity vikas is
port  (a: in STD_LOGIC;
    b: in STD_LOGIC;
    c: out STD_LOGIC);
end vikas;
architecture Behavioral of vikas is
begin
process(a,b)
begin
c <= a and b;
```

end process;

end Behavioral;

Circuit Diagram:--





RESULT: We have successfully performed AND gate operation and drawn its characteristics.

Experiment No:- 1.3

OBJECT: Simulation of OR gate using VHDL code through Xilinx**.**

APPARATUS: Xilinx ISE Design Suite 12.1

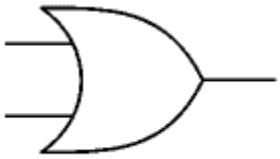
TRUTH TABLE:

a	b	c=a OR b
0	0	0
0	1	1
1	0	1
1	1	1

VHDL Module Code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity vikas is
port  (a: in STD_LOGIC;
      b: in STD_LOGIC;
      c: out STD_LOGIC);
end vikas;
architecture Behavioral of vikas is
begin
process(a,b)
begin
c <= a or b;
end process;
end Behavioral;
```

Circuit Diagram:--



OR GATE

RESULT: We have successfully performed OR gate operation and drawn its characterstics.

Sample Viva Question

S.NO	Viva Questions
1	What is X-OR gate?
2	What are the universal logic gates?
3	How student design NAND and XOR GATE by using AND, OR, NOT Gates.
4	What is application of Boolean algebra?
5	Design all Gates by using Universal Gates.
6	The problem of logic race occurs in which logic?
7	Show that the dual of the EX – OR is equal to its complement?
8	In the expression A+BC, the total number of min terms will be?
9	How does Boolean algebra different from normal algebra?
10	Define the truth table?
11	In the expression A+BC, the total number of min terms will be?
12	Show that the dual of the EX – OR is equal to its complement?
13	. What is the difference between canonical form and standard?
14	The problem of logic race occurs in which logic?
15	How do you convert a hexadecimal no. to octal and vice versa?

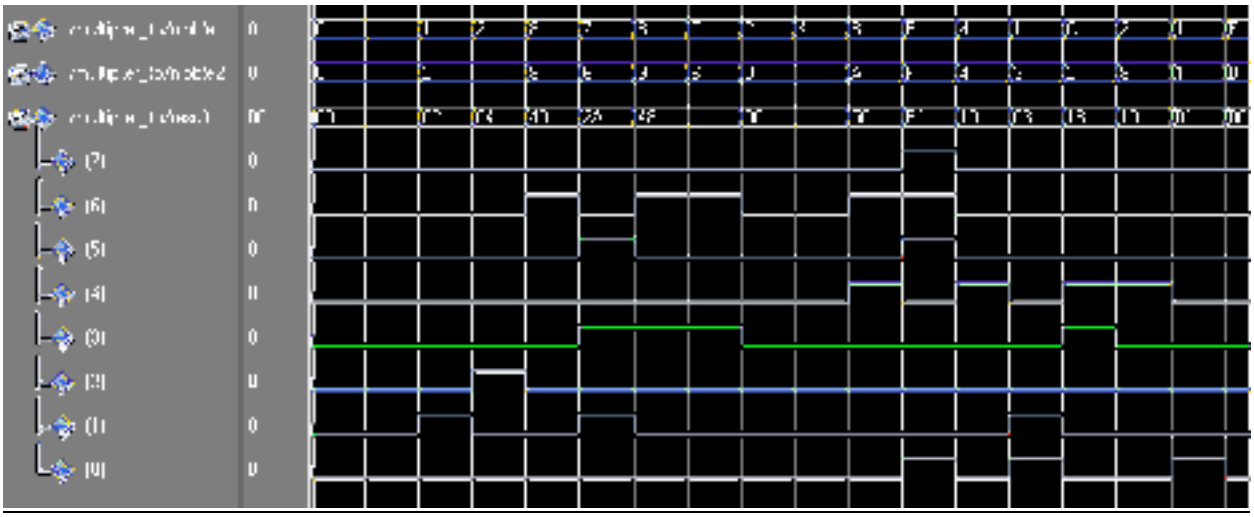
Experiment No. 2

OBJECT- : Simulate 4*4 Multiplier using VHDL code through Xilinx.

APPARATUS: Xilinx ISE Design Suite 12.1

VHDL CODE-

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;
 entity Multiplier_VHDL is
   port
   (
      Nibble1, Nibble2: in std_logic_vector(3 downto 0);
       Result: out std_logic_vector(7 downto 0)
   );
end entity Multiplier_VHDL;

architecture Behavioral of Multiplier_VHDL is
begin
    Result <= std_logic_vector(unsigned(Nibbl1) * unsigned(Nibble2));
 end architecture Behavioral;
```



RESULT: We have successfully performed Multiplier operation and drawn its characteristics

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR

Sample Viva Question

S.NO	Viva Questions
1	What is the multiplier effect?
2	Write syntax for (i) If statement (ii) For loop statement (iii)Wait statement (iv)Case statement
3	What is setup time and hold time?
4	What is identifier and how it's write in program?
5	Define difference between Function and Procedure subtype?
6	In which process student convert concurrent statements into sequential statements?
7	What is mean by structure?
8	Which statements is used, when a student write a concurrent statement in loop or in If statement, and program run without error.
9	Write Syntax for the Package Body.
10	When we declare package body?
11	What is Identifier and define it's rules.
12	What is syntax of IF statement?
13	What do you mean by end around carry?
14	Which of the following logic expressions represents the logic diagram shown?
15	Design a full subtractor with two half sub tractors?

Experiment No. 3

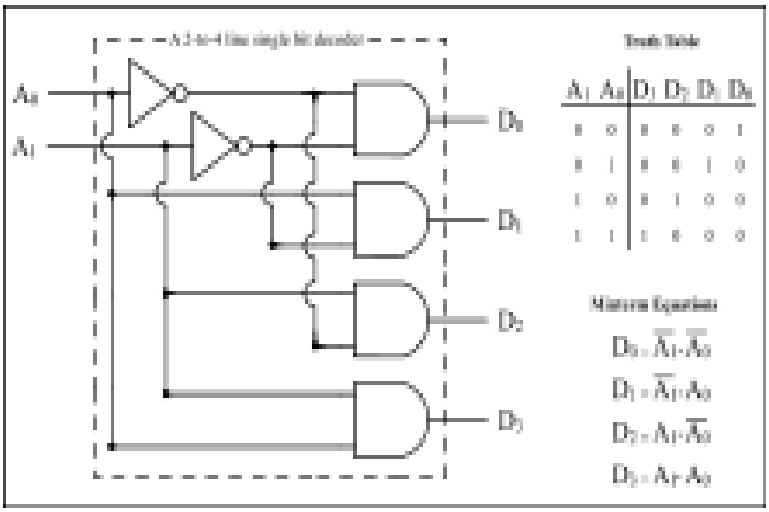
OBJECT: Write VHDL code for Decoder

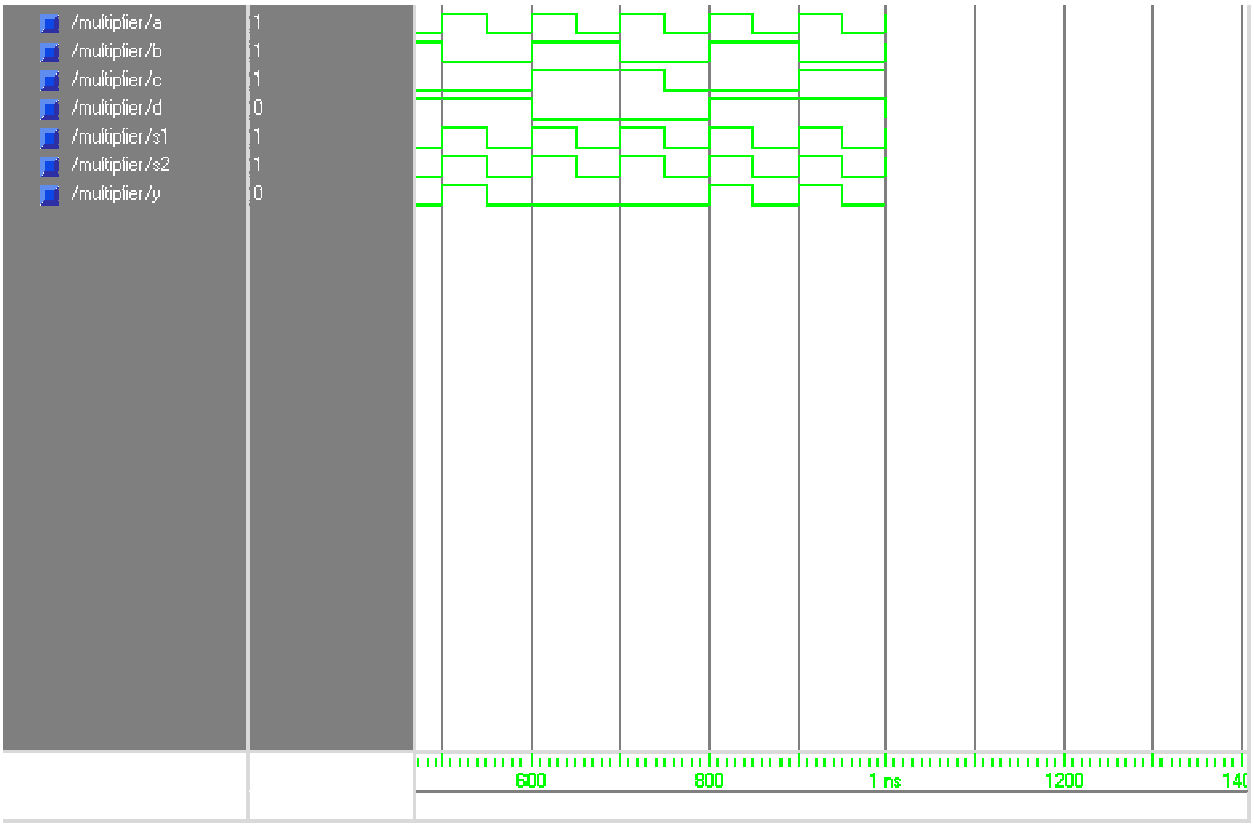
APPARATUS: Xilinx ISE Design Suite 12.1

VHDL CODE-

```
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;
ENTITY dec24d IS
PORT(A,B,EN_L:IN BIT;
        Q0,Q1,Q2,Q3:OUT BIT);
END ENTITY;
ARCHITECTURE dataflow OF dec24d IS
BEGIN
Q0<=(NOT A)AND (NOT B) AND (NOT EN_L);
Q1<=( A)AND (NOT B) AND (NOT EN_L);
Q2<=(NOT A)AND (B) AND (NOT EN_L);
Q3<=(A)AND (B) AND (NOT EN_L);
END dataflow;
```

Circuit Diagram:--





Sample Viva Question

S.NO	Viva Questions (to be answered by the student)
1	What is the advantage and disadvantages. What is the advantage and disadvantage of Decoder and encoder?
2	In which method Decoder is differ from De-multiplexer.
3	What do you mean by (a) serial data and (b) parallel data?
4	What is the basic difference between a counter and a shift register?
5	Draw the symbol of the seven segment display and find the binary no for the display value 4?
6	Draw JK Flip-Flop Circuit by using SR flip-flop.
7	Write Truth Table for all JK, SR, D Flip-Flop.
8	Why we use Sequential Circuit in Memory Cards?
9	Flip flop is a combinational circuit or sequential circuit & why?
10	Describe the main difference between a gated S-R latch and an edge-trigged S-R Flip flop?
11	Define the term FPGA.
12	What are the difference between PAL and PLA ?
13	What are the characteristics of the 2's complement method?
14	How can you perform binary subtraction using one`s and two`s complements ?
15	If a 3-input NOR gate has eight input possibilities, how many of those possibilities will result in a HIGH output?

Experiment 4.1

OBJECT- : Write VHDL code for 4:1 Multiplexer

APPARATUS: Xilinx ISE Design Suite 12.1

VHDL CODE-

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity abcde is
    Port ( i0 : in  STD_LOGIC;
           i1 : in  STD_LOGIC;
           i2 : in  STD_LOGIC;
           i3 : in  STD_LOGIC;
           sel : in  STD_LOGIC_vector (1 downto 0);
           y : out  STD_LOGIC);
end abcde;
architecture Behavioral of abcde is
begin
process(i0,i1,i2,i3,sel)
begin
case sel is
when"00"=> y <=i0;
when"01"=> y <=i1;
```

when"10"=> y <=i2;

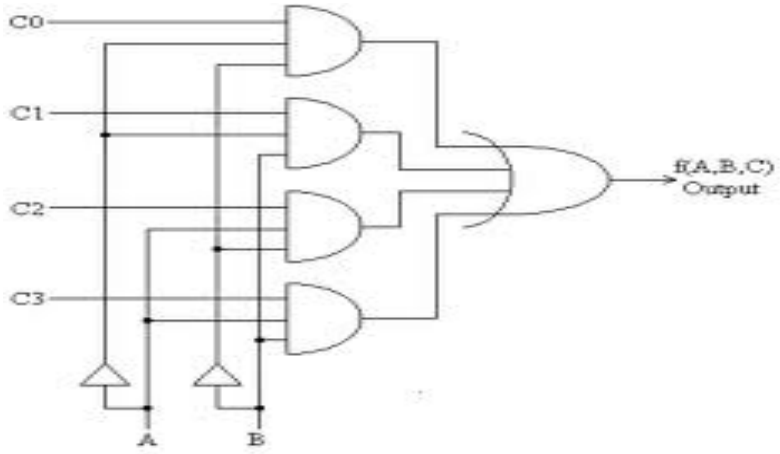
when others=> y <=i3;

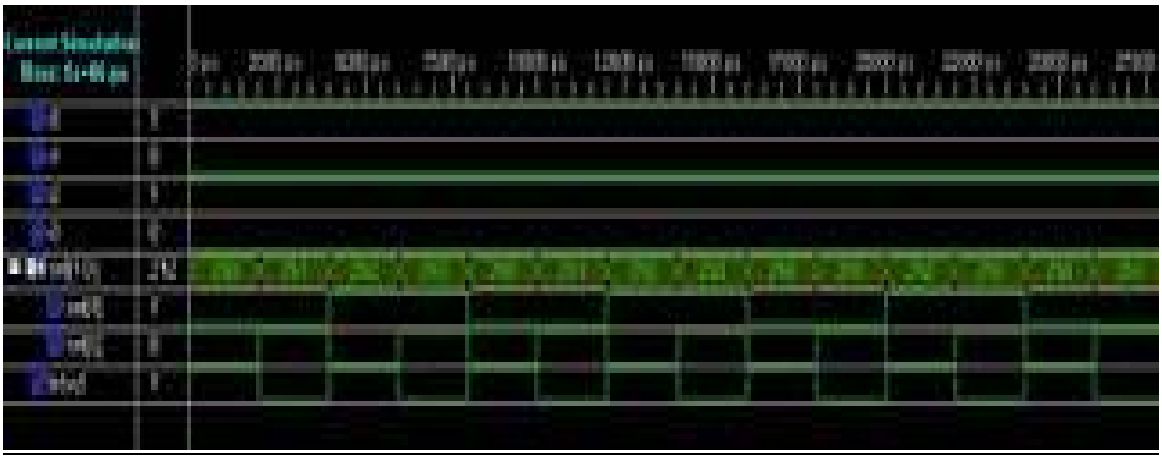
end case;

end process;

end Behavioral;

Circuit Diagram:--





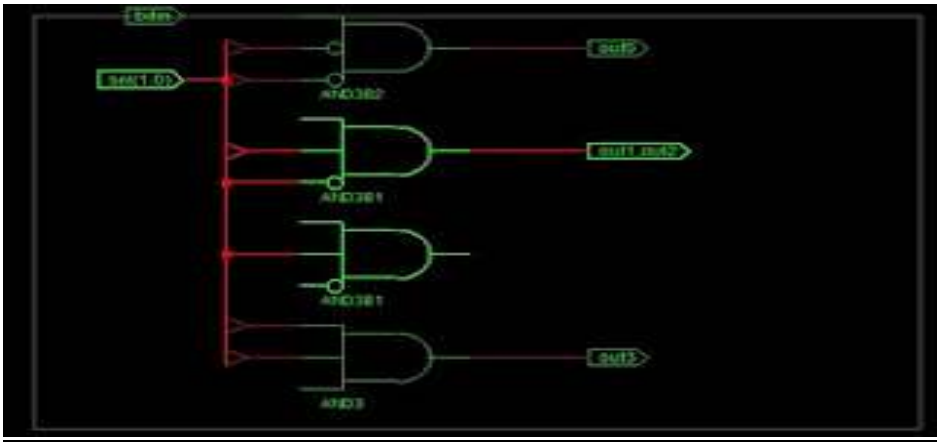
Experiment 4.2

OBJECT- : Write VHDL code for 1:4 De- Multiplexer

VHDL CODE:-

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


entity demux1_4 is
port (
      out0 : out std_logic;   --output bit
      out1 : out std_logic;   --output bit
       out2 : out std_logic;   --output bit
      out3 : out std_logic;   --output bit
       sel : in std_logic_vector(1 downto 0);
       bitin : in std_logic   --input bit
    );
end demux1_4;
architecture Behavioral of demux1_4 is
begin
process(bitin,sel)
begin
case sel is
  when "00" => out0 <= bitin; out1 <= '0'; out2 <= '0'; out3 <='0';
  when "01" => out1 <= bitin; out0 <= '0'; out2 <= '0'; out3 <='0';
  when "10" => out2 <= bitin; out0 <= '0'; out1 <= '0'; out3 <='0';
  when others => out3 <= bitin;  out0 <= '0'; out1 <= '0'; out2 <='0';
end case;
end process;
end Behavioral;
```

RESULT:-



Sample Viva Question

S.NO	Viva Questions **(to be answered by the student)**
1	What are difference between Multiplexer and Encoder?
2	What are difference between De-Multiplexer and Decoder?
3	Write Characteristic Equation for 8*1 Multiplexer. What is the function of an enable input on a multiplexer chip?
4	What is application of a digital multiplexer?
5	Explain difference between Multiplexer and De-multiplexer?
6	What is ripple counter?
7	What is meant by stable state?
8	Distinguish between synchronous and asynchronous Circuit.
9	Design Truth Table for 4*1 multiplexer.
10	What are difference between Multiplexer and Encoder?
11	Design Full adder / Half subtractor using MUX ?
12	Design 16 : 1 MUX using 4 : 1 MUX ?
13	Design 32 : 1 MUX using 8 : 1 MUX?
14	What is the function of an enable input on a multiplexer chip?
15	Design Full subtractor using MUX?

Experiment No. 5

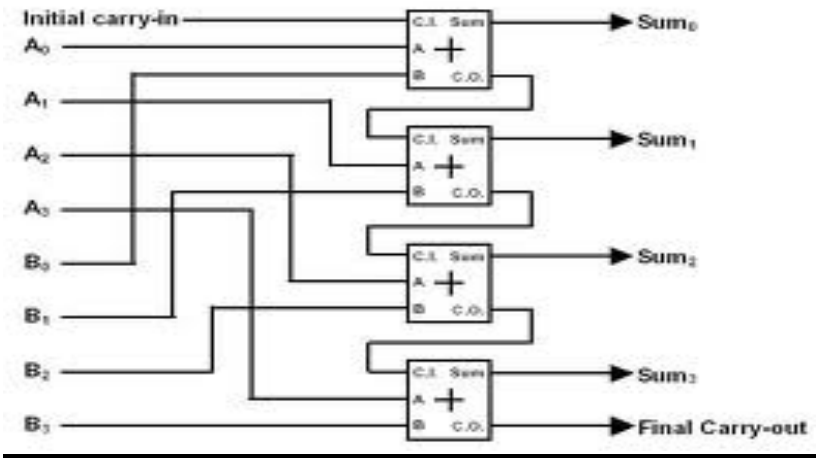
OBJECT: Write VHDL code for 4 bit Adder.

APPARATUS: Xilinx ISE Design Suite 12.1

VHDL CODE:-

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity lab8_final is
Port ( x : in std_logic_vector(3 downto 0);
y : in std_logic_vector(3 downto 0);
s0 : out std_logic_vector(3 downto 0));
end lab8_final;

architecture Behavioral of lab8_final is
signal c : std_logic_vector (3 downto 0):="0000";
component lab8ex3
port(a,b,cin:in std_logic;
s,cout:out std_logic);
end component;

begin

bit1: lab8ex3 port map (a=>x(0), b=>y(0), s=>s0(0), cin=>c(0), cout=>c(1));
bit2: lab8ex3 port map (a=>x(1), b=>y(1), s=>s0(1), cin=>c(1), cout=>c(2));
bit3: lab8ex3 port map (a=>x(2), b=>y(2), s=>s0(2), cin=>c(2), cout=>c(3));
bit4: lab8ex3 port map (a=>x(3), b=>y(3), s=>s0(3), cin=>c(3), cout=>c(0));
end Behavioral;
```

Circuit Diagram:--



Sample Viva Question

S.NO	Viva Questions **(to be answered by the student)**
1	Describe the 4 bit Ripple carry adder?
2	What is the advantage of serial?
3	What is the disadvantage of serial adders? For which applications are they preferred?
4	What do you mean by cascading of parallel adders? Why is it required?
5	What is a parity bit generator?
6	Design the logic diagram for full adder?
7	What is the advantage of parallel adders? For which applications are they preferred?
8	What do you mean by cascading of parallel adders? Why is it required?
9	Which ADDER is mostly used to implement 4-bit adder?
10	Describe the 4 bit Ripple carry sub-tractor.
11	Define scalar and composite Data Type ?
12	What is a parity bit generator?
13	What is the 4 bit binary parallel adder ?
14	What is the truth table of full adder ?
15	What is the truth table of Half adder ?

Experiment No. 6.1

OBJECT: Write VHDL code for Half Adder.

APPARATUS: Xilinx ISE Design Suite 12.1

TRUTH TABLE:

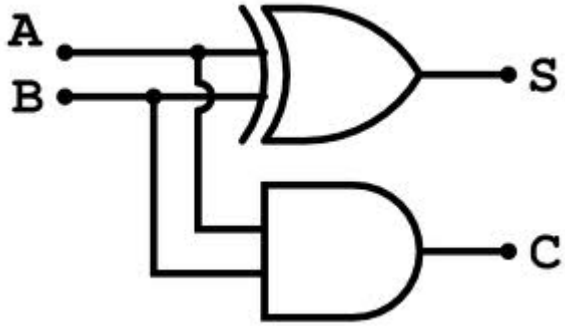
a	b	c= a.b	s
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

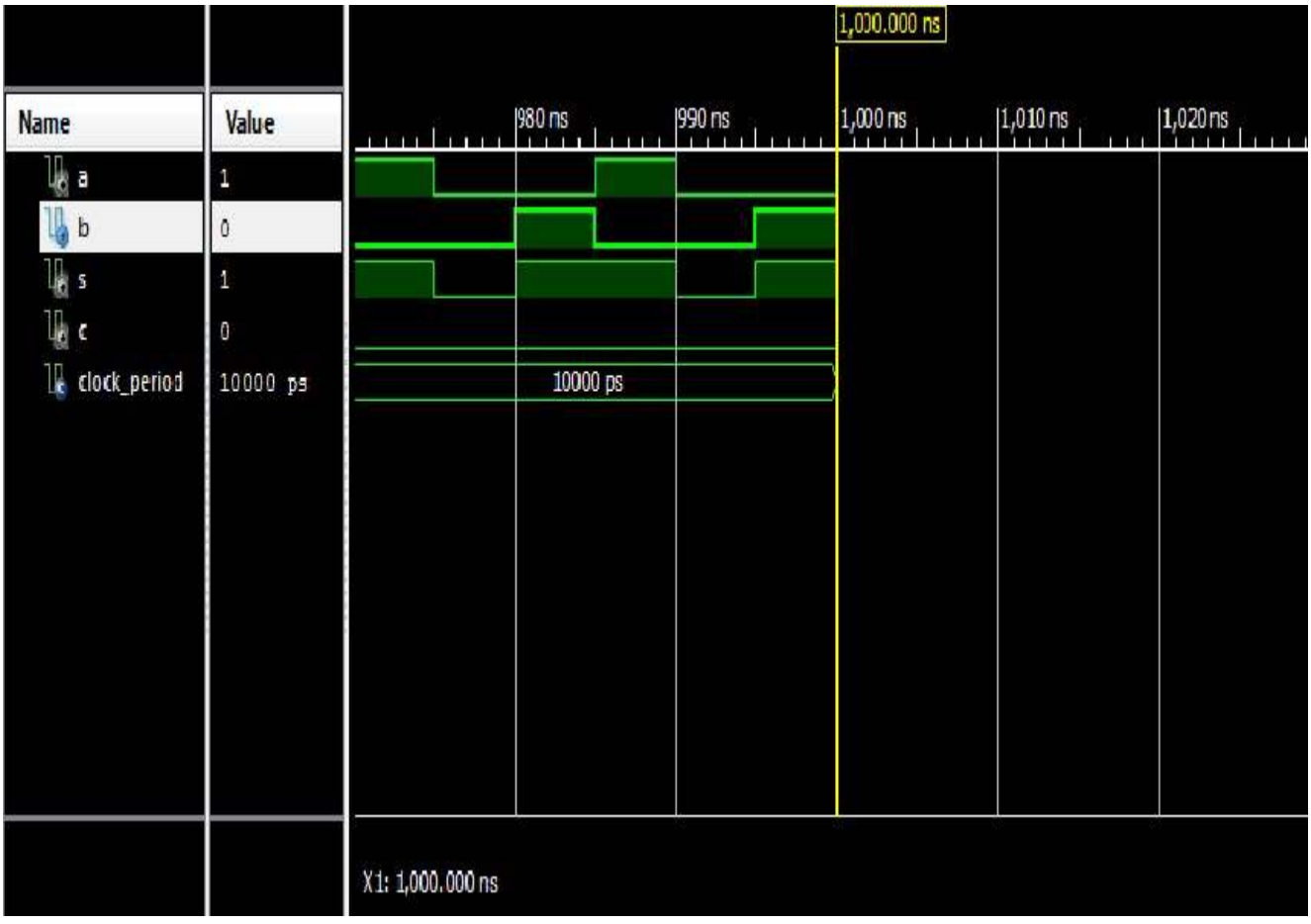
VHDL Module Code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity vikas is
port  (a: in STD_LOGIC;
     b: in STD_LOGIC;
     c: out STD_LOGIC;
          s: out STD_LOGIC);
end vikas;
architecture Behavioral of vikas is
begin
process(a,b)
begin
s <= a xor b;
c <= a and b;

end process;
```

end Behavioral;

Circuit Diagram:--





RESULT: We have successfully performed Half Adder operation and drawn its characterstics.

Experiment No. 6.2

OBJECT: Write VHDL code for Full Adder

APPARATUS: Xilinx ISE Design Suite 12.1

TRUTH TABLE:

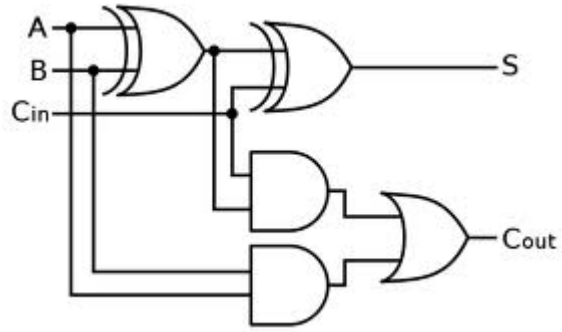
a	b	c	c_{out}	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

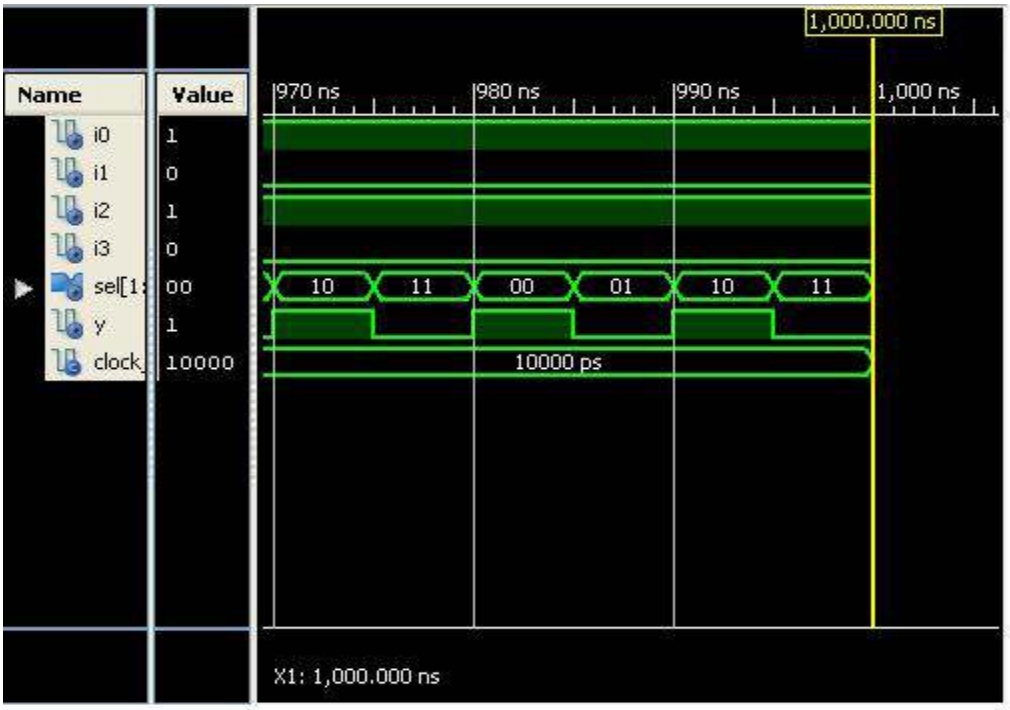
VHDL Module Code:

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity vikas is
port  (a: in STD_LOGIC;
    b: in STD_LOGIC;
    c: in STD_LOGIC;
        cout: out STD_LOGIC;
        s: out STD_LOGIC);
end vikas;
architecture Behavioral of vikas is
begin
```

```
process(a,b,c)
begin
s <= a xor b xor c;
c <= (a and b) or (b and c) or (c and a);
end process;
end Behavioral;
```

Circuit Diagram:--





RESULT: We have successfully performed Full Adder operation and drawn its characterstics.

Sample Viva Question

S.NO	Viva Questions
1	Which statements is used, when a student write a concurrent statement in loop or in If statement, and program run without error.
2	In which process student convert concurrent statements into sequential statements?
3	Write logic coding for Carry adder by using Generate statement?
4	Difference between Generate statement and Sequential Statement.
5	When a programmer use Behavioral style modeling?
6	What is a parity bit generator?
7	Draw circuit of half adder by using AND, NOT and OR GATE.
8	Difference Full Adder and half Adder.
9	What is meant by Ripple Carry Adder?
10	How does the look-ahead carry adder speed up the addition process?
11	Describe the 4 bit Ripple carry adder ?
12	What do you mean by cascading of parallel adder ? Why it is required ?
13	Describe the 4 bit Ripple carry subtractor?
14	Design the logic diagram for full adder?
15	How do you compare serial and parallel adders?

LIST OF EXPERIMENT BEYOND THE SYLLABUS

Experiment No:- 7

OBJECT: Simulation of SR Flip-Flop using VHDL code through xilinx.

APPARATUS: Xilinx ISE Design Suite 12.1

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
entity FDRS is
port(
Q : out std_ulogic;
C : in std_ulogic;
R : in std_ulogic;
S : in std_ulogic;
Clear: in std_ulogic
);
end FDRS;
architecture FDRS_arch of FDRS is
begin
process(C,Clear)
begin
if (rising edge(Clear)) then
Q <= '0';
end if;
if (rising_edge(C)) then
if (Clear = '1')
Q <=0;
elsif (R = '1') then
Q <= '0' ;
elsif (S = '1') then
Q <= '1' ;
```

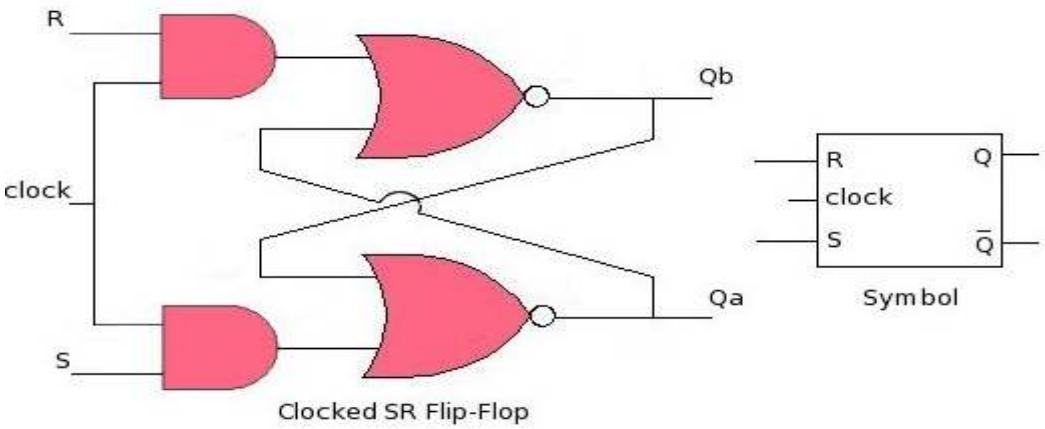
end if;

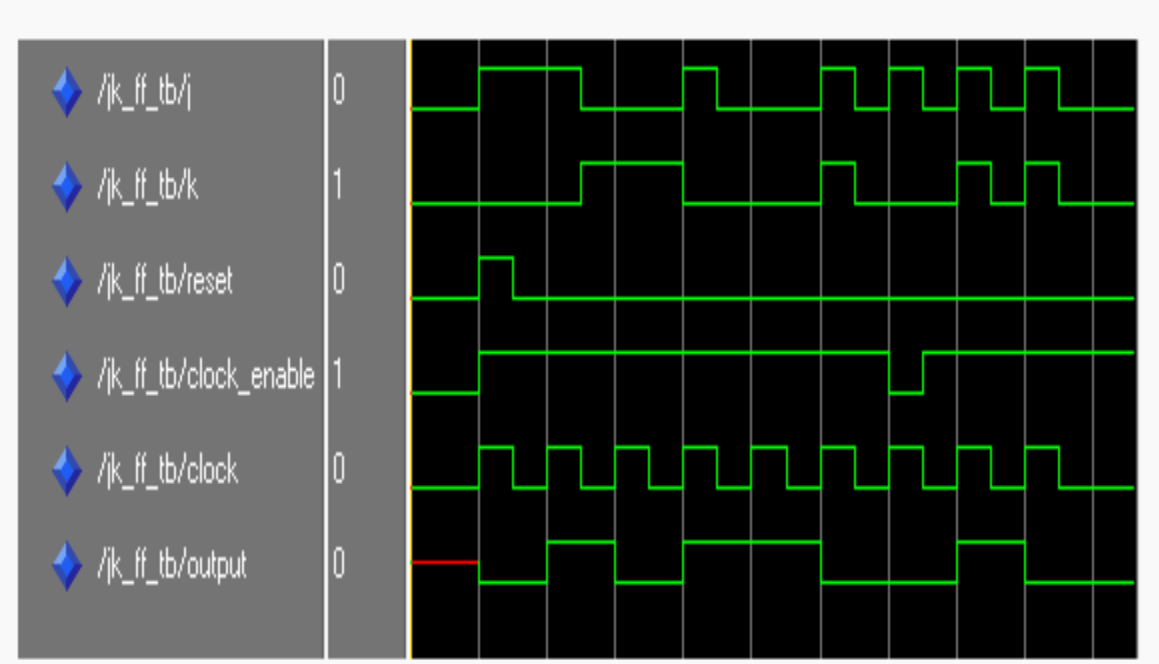
end if;

end process;

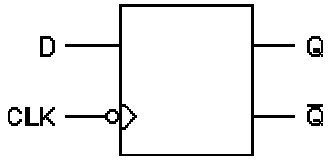
end FDRS_arch;

Circuit Diagram:--





Sample Viva Question

S.NO	Viva Questions
1	On a master-slave flip-flop, when is the master enabled?
2	What is the difference between the 7476 and the 74LS76?
3	when is the flip-flop in a hold condition?
4	What are disadvantage of an S-R flip-flop?
5	Write condition for the Edge-triggered flip-flops.
6	Why are the *S* and *R* inputs of a gated flip-flop said to be synchronous?
7	The symbols on this flip-flop device indicate _____. 
8	What is the significance of the *J* and *K* terminals on the J-K flip-flop?
9	Does the cross-coupled NOR flip-flop have active-HIGH or active-LOW set and reset inputs?
10	What is the hold condition of a flip-flop?
11	How can the cross-coupled NAND flip-flop be made to have active-HIGH *S-R* inputs?
12	When is a flip-flop said to be transparent?
13	What does the triangle on the clock input of a *J-K* flip-flop mean?
14	Which of the following describes the operation of a positive edge-triggered *D* flip-flop?
15	In VHDL, how many inputs will a primitive JK flip-flop have?

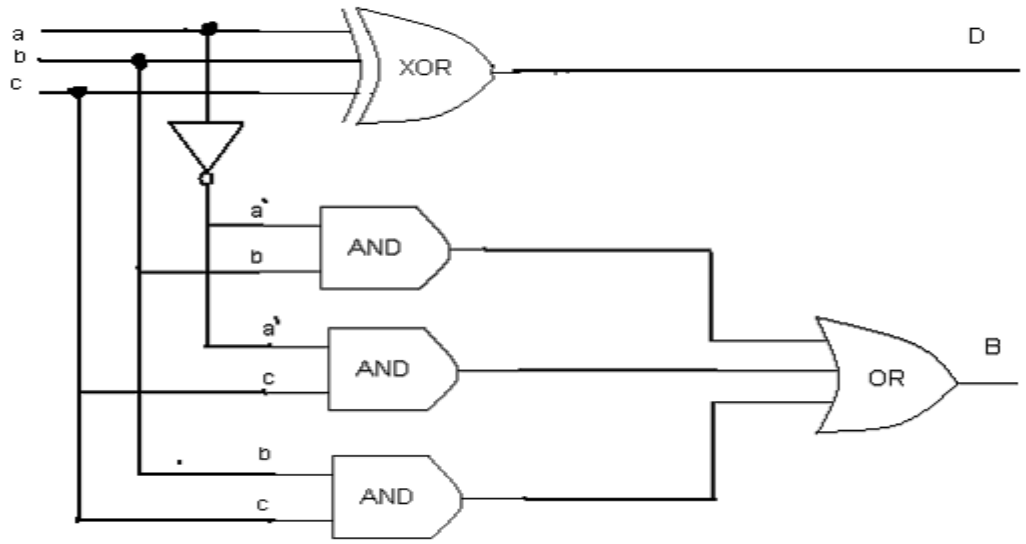
Experiment No:-8

OBJECT: Simulation of Full sub tractor using VHDL code through Xilinx**.**

APPARATUS: Xilinx ISE Design Suite 12.1

```
library ieee;
use ieee.std_logic_1164.all;
entity bejoy_fs is
port(a,b,c: in bit; D,B: out bit;);
end bejoy_fs;
architecture arc of bejoy_fs is
begin
D <=a  xor b xor c ;
b<=x and (not y);
do<=bi xor d;
2<=bi and (not b);
end arc;
```

Circuit Diagram:--



Sample Viva Question

S.NO	Viva Questions (to be answered by the student)
1	Draw circuit diagram of Half Subtractor circuit?
2	Draw circuit diagram of Full Subtractor circuit?
3	Draw Full Subtractor circuit by using Half Subtractor circuit and minimum no. of logic gate?
4	Write Boolean function for half Subtractor?
5	Write Boolean function for Full Subtractor?
6	Design the half Adder & Full Adder using NAND-NAND Logic.
7	What is Truth Table?
8	Explain the working of n-bit full subtractor with the help of neat block diagram.
9	Define half adder and full adder.
10	What is the hold condition of a flip-flop?
11	Define half subtractor and full subtractor.
12	Give the comparison between synchronous and asynchronous circuits.
13	Explain meaning of implies, subsumes, Implicants, prime implicants.
14	What is 1's and 2's complement?
15	What is serial in parallel out shift register. Explain.

Dr. Mahesh Bundele
B.E., M.E., Ph.D.
Director
Poornima College of Engineering
ISI-6, RIICO Institutional Area
Sitapura, JAIPUR