Experiment / Assignment / Tutorial No. 10

Grade: AA / AB / BB / BC / CC / CD /DD

Signature of the Staff In-charge with date

Batch: B1	Roll No.: 1711072	Experiment / assignment / tutorial No.:10

Title: VHDL programming for Half Subtractor

Objective: Implements a simple Half Subtractor in VHDL

Expected Outcome of Experiment:

CO4: Implement digital networks using VHDL.

Books/ Journals/ Websites referred:

- J. Bhasker, "VHDL Primer", Pearson Education
- Douglas L. Perry, "VHDL Programming by Example", Tata McGraw Hill
- http://esd.cs.ucr.edu/labs/tutorial/

Pre Lab/ Prior Concepts:

VHDL is an acronym for VHSlC Hardware Description Language (VHSIC is an acronym for Very High Speed Integrated Circuits). It is a hardware description language that can be used to model a digital system at many levels of abstraction ranging from the algorithmic level to the gate level. The complexity of the digital system being modeled could vary from that of a simple gate to a complete digital electronic system, or anything in between. The digital system can also be described hierarchically. Timing can also be explicitly modeled in the same description.

VHDL Programming Structure

Entity and Architecture are the two main basic programming structures in VHDL.

Entity: Entity can be seen as the black box view of the system. We define the inputs and outputs of the system which we need to interface. It is used to declare the I/O ports of the circuit.
Eg:

```
Entity ANDGATE is
        Port (A: in std_logic;
```

```
                B: in std_logic;
                Y: out std_logic);
End entity ANDGATE;
```

Entity name ANDGATE is given by the programmer, each entity must have a name.

Architecture: Architecture defines what is in our black box that we described using ENTITY. The description code resides within architecture portion. Either behavioral or structural models can be used to describe our system in the architecture. In Architecture we will have interconnections, processes, components, etc.

Eg:

```
Architecture AND1 of ANDGATE is
        --declarations
Begin
        --statements
        Y <= A AND B;
End architecture AND1;
```

Entity name or architecture name is user defined. Identifiers can have uppercase alphabets, lowercase alphabets, and numbers and underscore (_). First letter of identifier must be an alphabet and identifier cannot end with an underscore. In VHDL, keywords and user identifiers are case insensitive.

VHDL is strongly typed language i.e. every object must be declared. Standardized design libraries are typically used and are included prior to the entity declaration. This is accomplished by including the code "library ieee;" and "use ieee.std_logic_1164.all;"

Implementation Details:

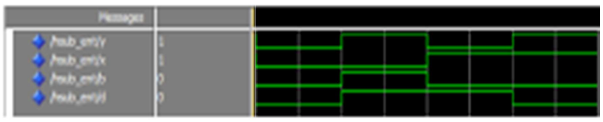
VHDL program code and simulation output

Code:

```
library ieee;
use ieee.std_logic_1164.all;

-------------------------------------

entity HSUB_ent is
port( x: in std_logic;
      y: in std_logic;
      B: out std_logic;
      D: out std_logic
```

```
);
end HSUB_ent;

---------------------------------------

architecture HSUB_arch of HSUB_ent is
begin

    process(x, y)
    begin
        -- compare to truth table
        if (x=y) then
          D <= '0';
      else
          D <= '1';
      end if;
      if (((x and '1')='0') and y='1') then
          B<= '1';
      else
          B<= '0';
      end if;
    end process;

end HSUB_arch;
```

Output:



Conclusion: The program ran successfully as we were able to simulate the half subtractor successfully.

Post Lab Descriptive Questions

1. Write VHDL program for full adder

Ans.

```
library ieee;
use ieee.std_logic_1164.all;

---------------------------------------

entity FADD_ent is
port( x: in std_logic;
      y: in std_logic;
      z: in std_logic;
      S: out std_logic;
      C: out std_logic
);
end FADD_ent;

----------------------------------------

architecture FADD_arch of FADD_ent is
begin

    process(x, y, z)
    begin
        S <= x XOR  y XOR z;
         C <= (x AND  y) OR  (z AND  x) OR  (z AND  y);
    end process;

end FADD_arch;
```