Digital Electronics

Experiment 10

Name: Ayush Jain

SAP ID: 60004200132

Div: B1

Branch: Computer Engineering

Aim: Case Study on VHDL

Theory: VHDL was designed by IBM, Texas Instruments, and Intermetrics as part of the DoD funded VHSIC program. It was standardized by the IEEE in 1987: IEEE 1076-1987 and enhanced version of the language defined in 1993: IEEE 1076-1993.

Introduction:

- The long form of VHDL is Very High Speed Integrated Circuit(VHSIC) hardware description language.
- It defines the syntax as well as simulation semantics for each language. It is strong typed language which frequently contains too many words to write.
- It is a hardware description language used to form a digital system at many levels of ideas ranging from algorithmic to the gate level.

Features of VHDL:

- Concurrency: VHDL is a concurrent language which executes statements simultaneous in parallel.
- Supports sequential statement: It can execute one statement at a time in sequence only.
- It is a strongly typed language.
- It supports hierarchies. For e.g.: full adder is composed of half adder and OR gate.
- It supports synchronous and asynchronous models.

Structure of VHDL Module:

Design units of VHDL code are independent components which are separately compiled and stored in library.

VHDL program is composed of following design units:

1. Package (optional): A VHDL package is a method to store and share some declarations which are common across many design units. It is represented by using package declaration and package body.

2. Entity: The basic unit of VHDL hardware design is entity. The entity describes the interface between design and external environment. It states the parameters such as name and port of the entity. ○ Entity declaration: The entity declaration defines the entity name and lists the input and output ports.
    - Port modes: In VHDL program, there are four modes available for ports: IN,OUT,INOUT and BUFFER.

    - SYNTAX:
      entity entity_name is
        port(port_name : mode port_type;
              port_name : mode port_type);
      end[entity_name];
3. Architecture Body: An architecture describes the internal organization or entity operation and it includes the statements which are used to model the entity behaviour. Architecture body is used to explain internal details of a design entity using any one of the following modelling styles:
    - Structural style: As a set of interconnected components. ○ Dataflow style: As a set of concurrent assignment statements. ○ Behavioral style: As a set of sequential assignment statements.
    - Mixed style: Any combination of the above three.

4. Configuration Declaration (optional): The configuration statement denotes the binding between the entity and architecture.

Types of VHDL operators in precedence order:

1. Miscellaneous operators.
2. Multiplying operators.
3. Adding operators.
4. Relational operators.
5. Logical operators.

VHDL Data Objects: In VHDL, code information is represented as data objects. Data objects are classified into four types:

1. Signals: Signal holds a list of values which includes the current value and set of possible future values.
2. Variables: For a given type, to hold a signal value a variable is used.
3. Constant: The use of constant is to improve the code readability, which is achieved by using constant name in place of number or value.
4. File: This type of data object includes value of specific type sequentially. It provides an interface between VHDL programs and Host environment.

VHDL Data Types: In VHDL, each data object has a data type. By using type conversion functions or casting, objects of different base types can be assigned or compared to one another directly. Following data types are available:

- Scalar: Values belonging to scalar type are ordered. Sub-types of Scalar data type are:
    1. Enumeration type: includes character literals and identifiers.
    2. Integer type: range of integer values within a specified integer range.
    3. Physical type: represents measurement of physical quantity.
    4. Real (Floating point) type: set of positive and negative numbers which contains a decimal point.

- Composite Types: A collection of values are represented by composite data type. Two classes of composite types are:
    1. Array Types: An array is a collection of objects that have the same subtype. By specifying the index values into the array, elements of an array can be accessed.

    2. Record Types: Record is heterogeneous composite type i.e. same or different types of elements are included in the record type. Elements in the record type are accessed through field name.

- Access Types: The values which belongs to an access type are pointers to dynamically allocated object of some other types.
- File Types: File types are used for object representation of file in the host environment. The value of file object is sequential values included in the host file.

Program Examples:

1. **Implementation of full adder using half adder:**

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
entity fa is
Port(x,y,z: in std_logic;
Sum,carry: out std_logic);  end
fa;
architecture structural of fa is
component HA is
```

```
port(A,B: in std_logic; S,C: out std_logic);
end component;
signal P,Q,R: std_logic;
begin  u1: HA port
map(x,y,P,R);  u2: HA port
map(P,z,sum,Q);  carry<=Q
or R;
end structural;
```

2. Implementation of 4:2 Priority Encoders using If-Then-Else Statement:

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

entity priorityencoder is     port(c,d,e,f:
in std_logic_1164.all;  s: in
std_logic_vector(1 downto 0);  pout:
out std_logic);
end priorityencoder;

architecture arch of priorityencoder is
begin  myif_pro: process (s,c,d,e,f)

begin  if s="00"
      then  pout<=c;
      elseif s="01" then
                pout<=d;
      elseif s="10" then
                pout<=e;
      else pout<=f;
      end if;
  end process myif_pro;  end
arch;
```