MODULE 4:
WEEK 4

Learning Outcomes:

After completing this course you are expected to demonstrate the following:

1. Describe Hardware Description Language
2. Recognize the basic concepts and uses of HDL,

A. Engage

Trivia

Due to the exploding complexity of digital electronic circuits since the 1970s, circuit designers needed digital logic descriptions to be performed at a high level without being tied to a specific electronic technology, such as ECL, TTL or CMOS. HDLs were created to implement register-transfer level abstraction, a model of the data flow and timing of a circuit.

B. Explore

Video title:Hardware Description Language
YouTube Link:https://www.youtube.com/watch?v=yFRICQtxrj4
Module VideoFilename:**Week 4 – Hardware Description Language**

C. Explain

1. **HARDWARE DESCRIPTION LANGUAGE**

Hardware Description Language is a formal language for the operations of integrated circuits, their design and software based simulations**.** It expresses a temporal behavior and/or a (spatial) Circuit structure in normal text.

STRUCTURE OF HARDWARE DESCRIPTION LANGUAGE

HDLs represent an executable specification of a particular hardware. A simulation program that provides the basic semantics of the language and the passage of time, gives the hardware designer the ability to model a piece of hardware before being physically prepared. This possibility of execution makes it look as if to use this language to program something. There are Hardware Description Languages and simulators for modeling in digital and analog technology.

The following levels of abstraction are used:

1. Behavioral model, partly has no synthesis capability
2. Register Transfer Level (RTL model, synthesis capability)
3. Gate level model (netlist)

VHDL was based on the Ada programming language. Initially, Verilog and VHDL were used to document and simulate circuit designs already captured and described in another form (such as schematic files).

D. Elaborate

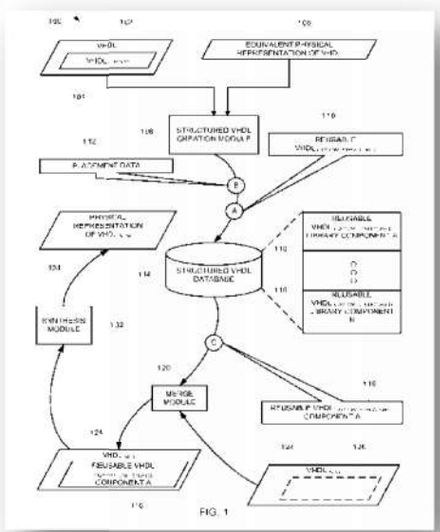
HARDWARE DESCRIPTION LANGUAGE: SIMULATION

A **Synthesizer, or logic synthesis tool** – a program that can infer hardware logic operations from the language statements and produce an equivalent net list of generic hardware primitives to implement the specified behavior.

Synthesizers generally ignore the expression of any timing constructs in the text.

Digital logic synthesizers, for example, generally use clock edges as the way to time the circuit, ignoring any timing constructs. The ability to have a synthesizable subset of the language does not itself make a hardware description language.

Essential to HDL design is the ability to simulate HDL programs. Simulation allows an HDL description of a design (called a m**odel**) to pass design verification, an important milestone that validates the design's intended function **(specification)** against the code implementation in the HDL description. It also permits architectural exploration. The engineer can experiment with design choices by writing multiple variations of a base design, then comparing their behavior in simulation.



Difference VHDL and Veril

VHDL	
It allows the ua types.	to define data types.
It supports thnal array.	ulti-Dimensional array.
It allows concu calls.	nt calls.
A mod operati	sent.
Unary reductic present.	is present.
It is more diffi	

Basic Elements of VHDL

1. **Entity**
   The Entity is used to specify the input and output ports of the circuit. An Entity usually has one or more ports that can be inputs (in), outputs (out), input-outputs (inout), or buffer.
2. **Architecture**
   Architecture is the actual description of the design, which is used to describe how the circuit operates. It can contain both concurrent and sequential statements.
3. **Configuration**
   A configuration defines how the design hierarchy is linked together. It is also used to associate architecture with an entity.

Types of Modelling styles in VHDL

1. **Data flow modeling (Design Equations)**
   Data flow modeling can be described based on the Boolean expression. It shows how the data flows from input to output. It works on concurrent execution.
2. **Behavioral modeling (Explains Behavior)**
   Behavioral modeling is used to execute statements sequentially. It shows that how the system performs according to the current statement.
3. **Structural modeling (Connection of sub modules)**
   Structural modeling is used to specify the functionality and structure of the circuit. It also contain signal declarations, component instances, and port maps in component instance.

E. Evaluate

ASSESSMENT:

Instructions: You may write your answer on the Answer Sheet (AS) provided in this module.

CONTENT FOR ASSESSMENT:

I. Identification (1 point)

1. Modeling used to execute statements sequentially
2. description of a design in Simulation of HDL
3. validates the design's intended function
4. It is a language used for describing a digital system such as a network switch, a microprocessor, a memory, or a flip-flop. traditional method of electronic design
5. traditional method of electronic design
6. Design that specify a circuit's characteristics using operations and the transfer of data between the registers
7. Basic element of VHDL that specify the input and output ports of the circuit
8. Basic element of VHDL that refers to the actual description of the design
9. Defines how the design hierarchy is linked together
10. Shows how the data flows from input to output

References:

1. *Hardware Description Language:* https://www.youtube.com/watch?v=yFRICQtxrj4
2. HDL : Structure & Simulation: https://thecustomizewindows.com/2014/05/hardware-description-language-structure-simulation/
3. https://www.javatpoint.com/vhdl
4. Verilog : https://www.javatpoint.com/verilog

Facilitated By:		
Name	:	
MS Teams Account (email)	:	
Smart Phone Number	:	