NEW HORIZON COLLEGE OF ENGINEERING

New Horizon Knowledge Park, Ring Road, Marathalli

Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC

Accredited by NAAC with 'A' Grade, Accredited by NBA

DEPARTMENT OF ELECTRICAL & ELECTRONICS ENGINEERING

Workshop Report

Hands on training session on Verilog HDL

Dr. Kamalakannan V,Advanced Engineer,Altran Technologies Ltd.,Bangalore.

26th October,2019

Semester : 3rd

No. of Students Attended : 40

Course: Digital System Design

Course code:19EEE34

Section : A & B

Venue : Circuit Simulation Lab, EEE Department,NHCE

Faculty Co-ordinators: Ms. Karthika M

Mr. Satishkumar D

Report Prepared by: Md. Sagar Khan (1NH18EE727)

Contents

1. Introduction about Resource Person

- The workshop was conducted by Dr. Kamalakannan V, Advanced Engineer, Altran Technologies Pvt. Ltd.,Bangalore. He has 13 years of teaching at college level and industrial at Design Engineer level + 4 Years of Research Experience. He has Solid experience in VLSI design implementation with strong understanding of digital and analog design, Specs to RTL coding, including VHDL and Verilog, Very efficient in converting specs to RTL and FPGA design flow using Xilinx Tool & Interfacing FPGA board. He has published number of papers in scopus indexed international and national journals. Also he has conducted workshops on FPGA domain.

2. AIM of the Workshop session

1) To make the students to understand about the Verilog Hardware Description Language (which is included as a part of Digital System Design (19EEE34) in the III semester).
2) To provide an interactive session with the industry people.
3) To create an aware of the importance of co-curricular activities in engineering domain.

3. Abstract of the workshop session

Initially session dealt with the basic concepts of digital system design. Then the concepts of Verilog HDL were related with C Programming & complete details on Verilog HDL with examples were discussed.

4. Introduction to Basics of Digital System

The construction of different systems, the designs which are nothing but the specification of requirement, difference between small devices (like flip-flops) & large circuit (like microprocessors).
The hardware decription languages are used to design the large & advanced circuit such as micro-controllers & microprocessors etc.
The usage of the variable, constants & logics etc in Verilog HDL are similar to C Language.

5. Design in Details

Design is the specification of requirement as discussed earlier.
Design can be of different levels such as :

1. Switch Level

2. Register Transfer Level
3. Instruction Set Architecture Level etc.

For the design to check whether it is working or not
A. Simulation & B. Synthesis is done.

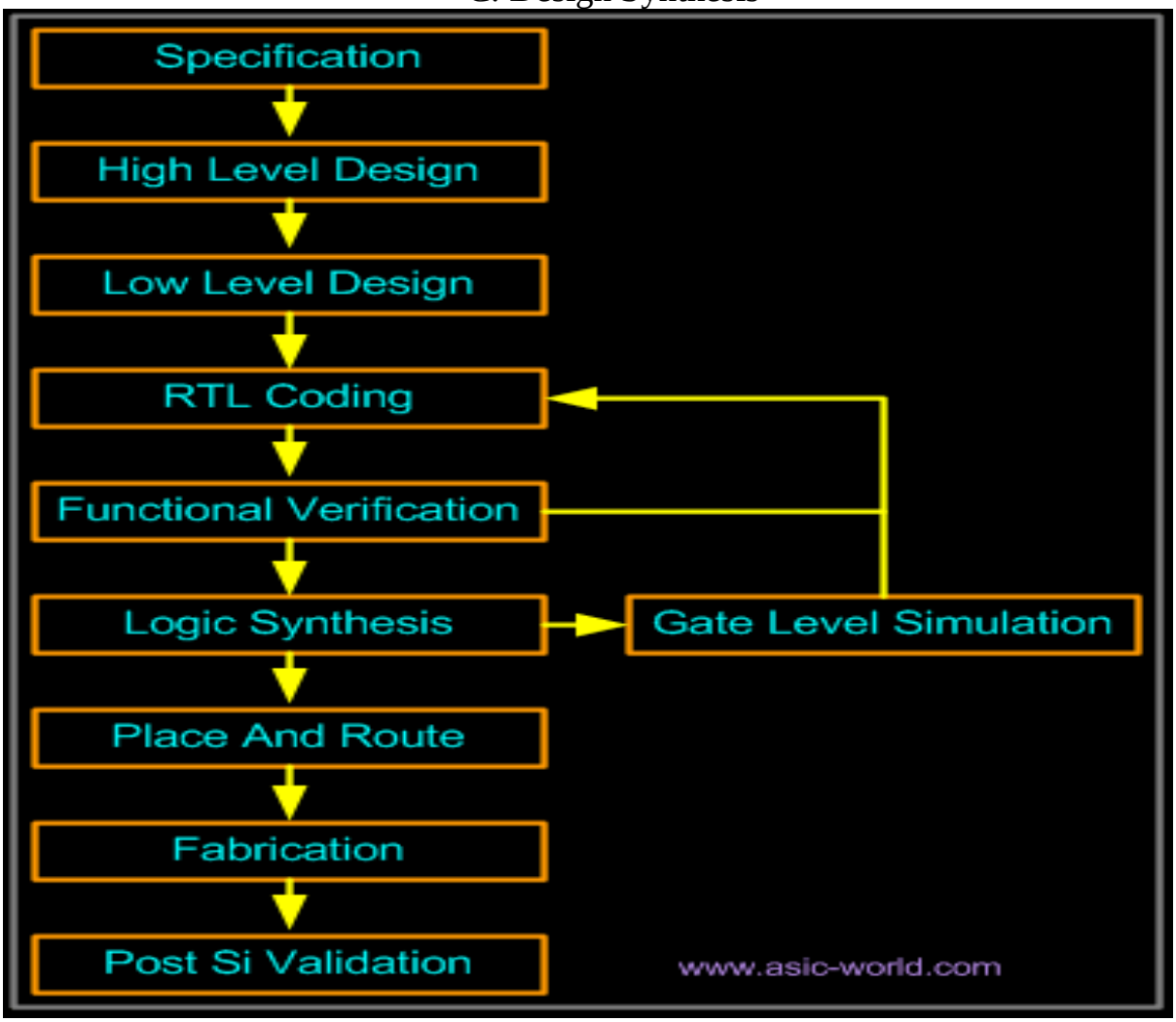
5.1 Description :

If the simluation is successfull then the design is working & then it can be synthesised where the netlist is created. This netlist gives out the list of the components to be used.

5.2 Design of Describe System :

The very 1st step after the description of system is how to design the describe system. It includes :

A. Design Specification
B. Design Simulation
C. Design Synthesis



5.3 Hardware Description Language :

A **hardware description language** (**HDL**) is a specialized computer language used to describe the structure and behavior of electronic circuits, and most commonly, digital logic circuits

A hardware description language looks like a programming language such as C; it is a textual description consisting of expressions, statements and control structures. One important difference between most programming languages and HDLs is that HDLs explicitly include the notion of time.

The functional working of HDL is parrellel way.

5.4 Logic Simulation :

It gives the output for the design made for specification. We can modify any gate by verifying the truth table & waveforms. It includes the fabrication of the devices & also we can make our own devices.

5.5 Things which can & can't be translated :

Transaltion is conversion of input to output. The things which can be translated are :

A. Strucutural Definitions
B. Behavioral Blocks
C. User Defined Function(Blocks) etc.

The things which can't be translated are:

A. Initial Blocks
B. Delays
C. A variety of other obscure language features etc.

5.6 The Role of HDL :

The language helps to describe any digital circuit in the form of structural, behavioral and gate level and it is found to be an excellent programming language for FPGAs and CPLDs.

Thus it has following roles :

A. Behavioral
B. Strucural
C. Physical

5.7 Design Methodologies & Types of HDL :

I. There are two types of design methodes:

A. Top Down
B. Down Top

These are mainly based on Hyrerchial Rank.

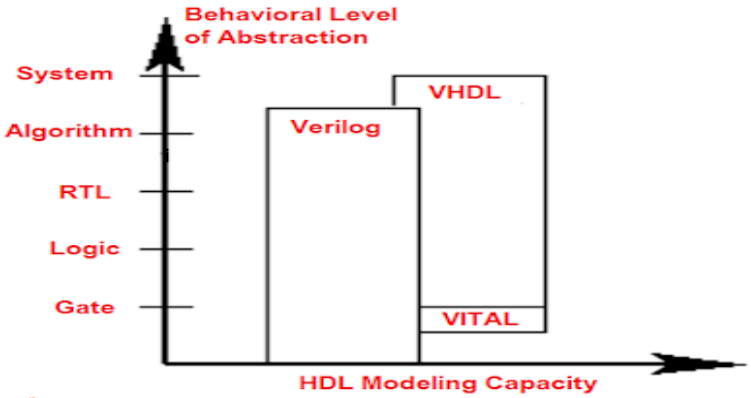
II. There are mainly two types of Hardware Discription Languages which are standarised by IEEE, those are as follows:

A. VHDL/VHSIC HDL (Very High Speed Integrated Circuits Hardware Description Language), discovered by US Department of Defence.
B. Verilog HDL, discovered by Open Verilog International.

5.8 Verilog HDL vs. VHDL :

Why to learn Verilog HDL instead of VHDL?

A. Verilog HDL gives better low end than VHDL
B. It is more modelling language.



VHDL- VERILOG RELATIONAL OPERATORS

Verilog	VHDL	
>	>	Greater Than
>=	>=	Greater than or equal
<	<	Less Than
<=	<=	Less Than or equal
==	=	Logical Equality
!=	/=	Logical Inequality

VHDL has separate input & output blocks wherein Verilog HDL has both input & output in the same block.

6. Verilog HDL & its History

i. History: Verilog HDL was invented by Phil Moorby and Prabhu Goel around 1984. It served as a proprietary hardware modeling language owned by Gateway Design Automation Inc. At that time, the language was not standardized. It modified itself in almost all the revisions that came out between 1984 to 1990.

ii. Definition: Verilog, standardized as **IEEE 1364**, is a hardware description language (HDL) used to model electronic systems.

iii. Use: It is most commonly used in the design and verification of digital circuits at the register-transfer level of abstraction. It is also used in the verification of analog circuits and mixed-signal circuits, as well as in the design of genetic circuits.

iv. Features: It is case sensitive & parellel program compiling function.

6.1 Verilog Data Types in Details :

There are primarily two data types :

A. Nets

B. Registers (Integer,Real & Timer)

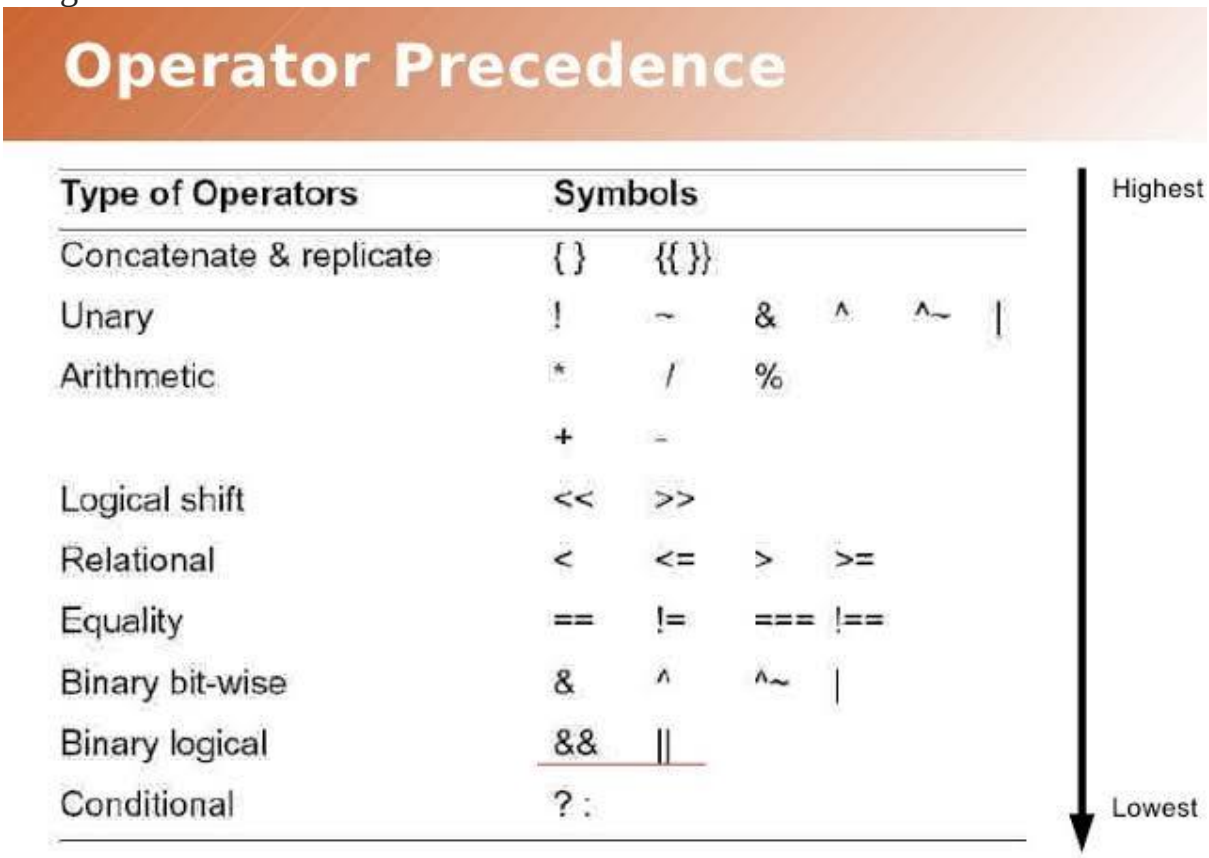
A. Nets : The internal connections of a block diagram except for input & output are called nets.Nets are represented as wires.

B. Registers : The function of registers is storing data & showing the result untill next input is selected,such as integers,real & timer etc.

i. Integers : Integers have default width of 32 bits.These are declared by keyword 'integer'.
ii. Real: Real number,constants & variables are decalred using keyword 'real'.
iii. Time: A special time register data type is used in verilog to store simulation time.
iv. Arrays: These are allowed for reg, integer & time,not allowed for real variables.
v. Strings: Can be stored in reg. Each character in string requires 8 bits of storage.

6.2 Operators, Procedural Constructs & Loop Statements :

i**. Operators:** The operators are same as 'C' Programming operators as shown in figure below.



ii. Procedural Constructs: a. Initial Statement (Evaluate only once)
b. Always Statement (Evaluate in a loop manner)
iii. Loop Statement: Repeat
While
For

7. Verilog Module

7.1 Description & Test Benches :

i. Description : A **"module"** is the basic building block in Verilog. A module can be an element or a collection of lower-level design blocks. A module provides the necessary functionality to the higher-level block through its port interface (inputs and outputs), but hides the internal implementation. In Verilog, a module is declared by the keyword **module**. A corresponding keyword **endmodule** must appear at the end of the module definition. Each module must have a **module_name**, which is the identifier for the module, and a **port list**, which describes the input and output terminals of the module. Design functionality is implemented inside module, after **port declaration**. The design functionality implementation part is represented as "**body**" here.

ii. Test Benches : Implement single model for many input without writing manually.

7.2 Verilog Modelling in Details :

There are mainly three types of modelling in Verilog HDL as follows:

A. Gate level modelling(Logic Gates)
B. Data Flow modelling(Assign Boolean expressions)
C. Behavioral modelling(Result)

7.3 Finite State Machine & Mealy vs. Moore :

i. Finite State Machine : These are the machine used for particular operation with step by step proccess.

ii. Mealy vs. Moore :

Mealy works at present time wherein Moore depends on both present & previous time.

8. Main points to remember & some tips for Verilog HDL Coding

i. Main points to remember:

A. Verilog is concurrent.
B. Think while writing the program-plan for requirement.
C. Blocking & Unblocking Code.

ii. Tips:

Don'ts	Does
1. Don't Write 'C' code 2. Think hardware ,not algorithm Verilog is inherrently parellel, compilers don't map algorithm to circuit.	1.Do describe hardware circuits 2.First draw a dataflow diagram Then start coding

9. Conclusion

The session was concluded with some questions and answers & practical coding example in computer. I asked different questions to Dr. Kamalakannan V sir about Verilog codings and sir cleared all the doubt with satisfactory answers.

9.1 Question & Answer(QNA) Session :

i. Question : What is done for reducing very large circuit transistor?
Answer : The channel length is reduced.

9.1 Question & Answer(QNA) Session :

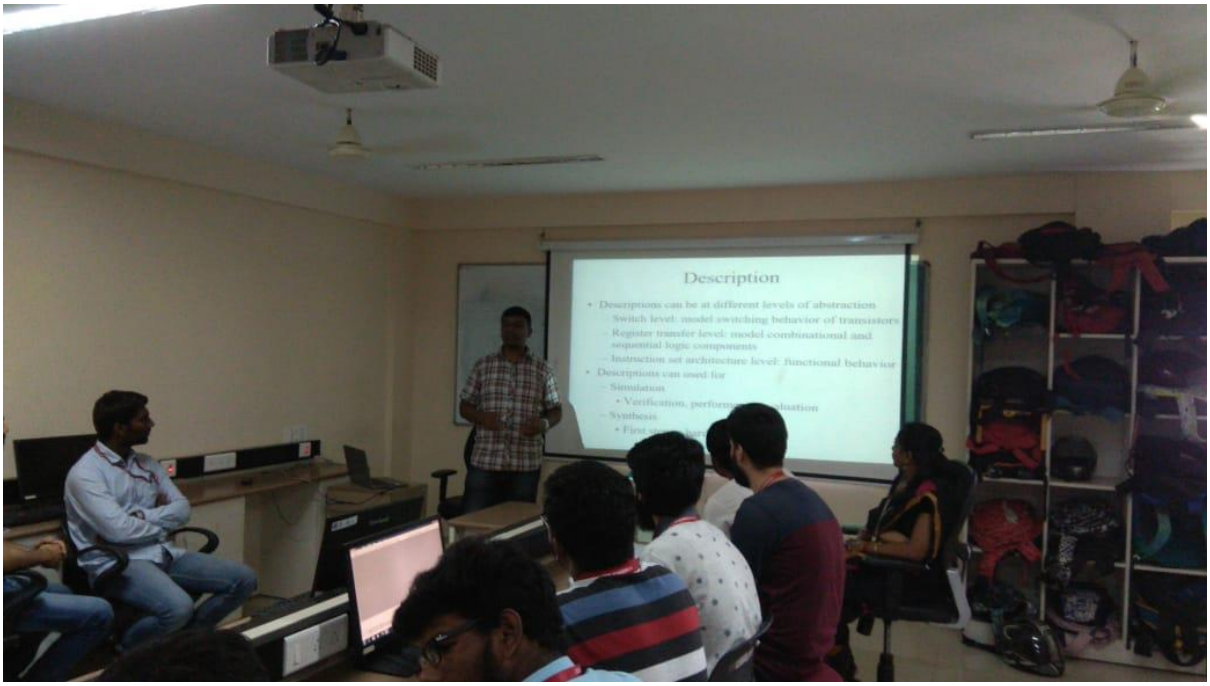
ii Question : Is the 'Always' statement has same working as any other loop?
Answer : Yes, It is similar to 'for', 'while' & 'do while' loop. But the main thing to be keep in mind is conditions.

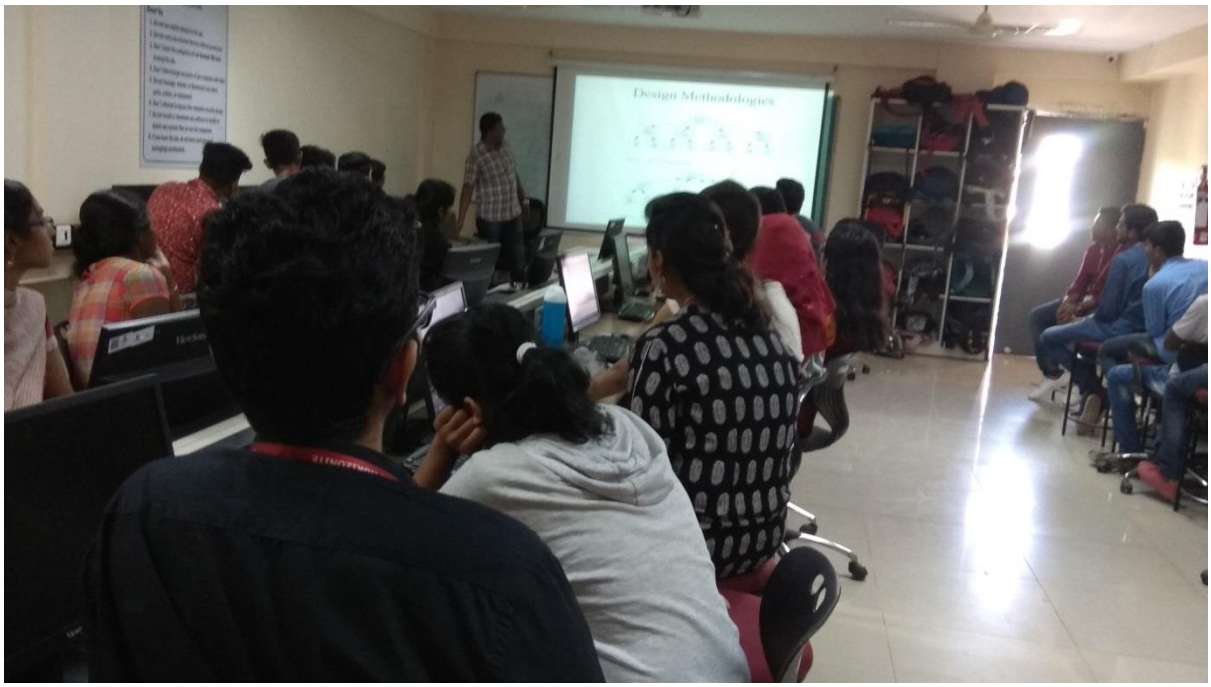
iii. Question : Which style of Verilog coding is being used the most?
Answer : There is nothing a particular style like structural,dataflow or behavioral style is used, for any specification the mixed of any of these three are used.

9.2 Practicle example on computer :

A half adder module coding was shown in three different coding styles practically.The codes are as follows:

```
Moudle Half_adder(a,b,s,c)
//Data Flow Style
input a,b;
outputs,c;
assign s=a^b;
assign c=a&b;
//Structural Style
xor(s,a,b);
and(c,a,b);
//Behavioral Style
if a='0';
   b='0';
  then s=0;
        c=0;
else s=1;
       c=0;
```





10. Experience from the Workshop Session

It was an informative, interesting and a successful workshop session. As a student of Electrical & Electronics Engineering,we understood the importance of Verilog HDL with in a short span of time. I express my thanks to Dr. Kamalakannan V who spent his valuable time for us. We also thank to our respected Principal,HOD & our faculties for arranging such an informative program.