**VERILOG -Introduction**

(SivaKumar P R)- Founder and CEO

Objectives-

* Applications of Verilog HDL
* Concepts and constructs
* Abstraction levels

**Verilog HDL**

Features

* Hardware Description language
* IEEE standard 1364-2001-describes syntax,not the style.
* Synthesizable constructs & Constructs for Stimulation also
* Digital and Analog design language
* Verilog AMS- Analog & Mixed Signal Simulation

Synthesizable Constructs Constructs for Simulation

Synthesizer is a tool.it converts your actual code into Netlex.

***Application areas of Verilog***

System Specification

Software

Boards and

Systems

Std Parts

PLD

FPGA

ASIC

Software Spec

Hardware Spec

HW/SW Partition

* Any electronics device locates hardware and software
* System lang – C,C++
* Verilog is used for create Synthesizable model
* Chip – Systhesize the RTL and produce the Netflix .The Netflix goes to backend teams. The backend teams converts Netflix into router Netflix. It send to foundary and foundary fabricated the chip.

**Abstraction levels**

**Circuit Level**

* CMOS circuits

**Gate Level**

* Basic gates are available as “Priitives”

**Data Flow Level**

* Register Transfer level
* Realized through concurrent assignments
* Output-; y= (A&B) |C

**Behavioral level**

* Highest level of design description
* Function only
* Stimulus generation
* No architecture
* Design modelling

always@(a,b,sel)

begin

if (sel== 0)

out = a;

else

out= b;

end

**Design Abstraction levels**

1. System level
2. Register level
3. Gate level
4. Transistor level-companies like IBM and apple (custimize the cmos like transitor)

[notes- Product companies like apple works system level. They buy IC and build the system together.this is called system level of abstraction]

[IT companies acts as RTL level and use HDL and ….

**Language Concepts**

Concurrency,Structure,proccedurall statements,Timing

**Hierarchical Design**

* Design distribution
* Smaller pieces of code
* Design reuse

Top Level chip

Sub-System IP3

Sub-system IP2

Sub-System IP1

Basic

Module

Basic

Module

Basic

Module

Basic

Module

Basic

Module

Stimulation process- debugging

**Verilog Syntax**

* Keywords in lowercase.
* Identifies are case sensitive
* Alphanumeric,underscore and $ characters can be used. Number and character not comes front
* Comments- single line and multiple line.

Eg :

1. // 2. /\* \*/ \*astricts

* Keywords- module and endmodule
* Port declarations

1.Input,output and inout

2.Default : wire

* Functionality -use assign or procedures

EDA TOOLS

1. Quartus prime: Syensizing the design
2. Model sim: Running the stimulation

Installation-

Browser:1. register for Intel FPGA program-

2.Download quartus prime lite edition

**Data types**

Values

* 0 – represents a logic zero, or a false condition
* 1- logic one, true
* x- represents an unknow logic value
* z- a high impedance state

numbers

<size>’<base><number>

Size-numb of bits

Base- binary b or B

Octal- o or O

Decimal – d or D

Hexadecimal- h or H

Number- consecutive charcters 0-f,x,z

In verilog Two type of datatypes –

1. Net –(wire data type)

2.Register-(reg Data type)

Net

* Net are continuously driven by combinational logic
* Use: They represent physical connection between multiple components

Register

* Reg data types do not yield hardwaew registers.
* Values retained until updated
* Assignments-Registers and nets can be mixed

Vectors

* Represents buses
* Slices- range direction must be same as vector

Memories

* A one dimensional array with elements of type reg
* Used to model ROMs,RAM,and reg files

Parameters

* Parameters are not variables,they are constants
* Their value can’t be changed during run time.
* Module parameters

**Operators**

1.logical operators- AND &&

OR ||

NOT !

Use – conditional operations like if-else

2.Bitwise Operators

* AND &
* OR |
* XOR^
* NEGATION ~
* XNOR ~^ or ^~

3.Concatenation operators

* It is the jpining together of bits resulting from two or more expressions
* Operands must be sized.
* Concatenation operator – { }

4.Arithmetic operators

* Multiplication
* Division
* -
* +
* Modulus %
* Power \*\*

**Verilog-Processes**

Types of assignments

Processes

continuous procedural

concurrent procedural Blocking Non-blocking

continuous concurrent assignments

* assigns values to net
* the assignments occurs whenever the value of the hight hand side changes
* excutes in parallel
* continuously active
* order independent

Procedural assignments

* Update the value of variables under the control of the procedural flow constructs that surrended them.
* Each procedure represents a separate activity flow in Verilog,all of which runs in parallel

‘’ initial’’ block

* Execution starts at zero simulation time
* Once execution only
* Non-synthesizable
* Execution stops when the last statement is executed.

Asynchronous clear and synchronous reset

* Asynchronous active low clear
* Synchronous active high reset

Procedural assignments

Blocking assignments

* Which are represented with the sign =
* These are excuted sequentially.i.e one statements is executed then the next statement is executed.
* One statement blocks the execution of other statements until it is executed
* Any delay attached is also got addede to delay in execution of next statements.
* Order is important

Non blocking assign

* A non-blocking assign represents with the sign <=
* Its execution is concurrent with that of the following assign or activity
* For all the non-blocking assign in a block,the right hand sides are evaluated first.subsequently the specified assign are scheduled
* It is illegal to use a non -blocking assign in a continuous assign statement or in a net declaration.

System task functions

System task- $displays or $write

* $displays or $write\* displays the specified variables once when the command is executed in the code flow.
* $display automatically adds a newline character to the end of its output,wheras the $write task does not

Finite state machine (FSM)

FSM architecture

* Moore FSM-AN FSM where the outputs are only a function of the states
* Mealy FSM- An FSM where one or more the outputs are a function of the present state and more of the outputs