Electronic Software Tool (BEC-255)

INTRODUCTION TO VHDL & VERILOG

PRESENTED BY: SAURABH KUMAR (RCTF)

ECED

MADAN MOHAN MALAVIYA UNIVERSITY OF TECHNOLOGY

CONTENTS

- What is HDL
- History
- Verilog, Development of Verilog & Design entities in Verilog
- Program structure
- Simulation and synthesis
- Limitations and conclusion
- About VHDL
- References

MADAN MOHAN MALAVIYA UNIVERSITY OF TECHNOLOGY

HDL

- A **hardware description language** (**HDL**) is used to describe the structure and behaviour of electronic circuits, and most commonly, digital logic circuits, which can then be placed and routed to produce the set of masks used to create an integrated circuit.

- HDLs form an integral part of electronic design automation (eda) systems, especially for complex circuits, such as application-specific integrated circuits, microprocessors, and programmable logic devices.

MADAN MOHAN MALAVIYA UNIVERSITY OF TECHNOLOGY

MOST COMMONLY USED HDLs

VERILOG-

The Verilog Hardware Description Language (Verilog HDL) is a language that describes the behavior of electronic circuits, most commonly digital circuits. Verilog HDL is defined by IEEE standards. Some common variants are: Verilog 1995, Verilog 2001, and System Verilog 2005.

VHDL-

VHDL stands for very high-speed integrated circuit hardware description language. It is a programming language used to model a digital system by dataflow, behavioral and structural style of modeling. This language was first introduced in 1981 for the department of Defense (DoD) under the VHSIC program.

HISTORY

- Due to the exploding complexity of digital electronic circuits since the 1970s (see Moore's law), 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.[1]

- There are two major hardware description languages: VHDL and Verilog. There are different types of description in them: "dataflow, behavioral and structural"

VERILOG

- It is a language used for describing a digital system like a network switch or a microprocessor or a memory or a flip−flop.

- Verilog is intended to be used for verification through simulation, for timing analysis, for test analysis (testability analysis and fault grading) and for logic synthesis.

MADAN MOHAN MALAVIYA UNIVERSITY OF TECHNOLOGY

WHY WE NEED VERILOG?

- Verilog is a crucial aspect of digital design for several reasons-
- First that it is a high-level language that provides abstraction from low level details of circuit design. This makes easier for designer to specify the functionality of circuit without getting bogged down.
- Secondly, Verilog is a platform independent language ,means it can we synthesized on any hardware platform providing high degree of flexibility and versatility.
- Thirdly, Verilog is an open standard language with a wide range of support from EDA tools vendors , making it easy and affordable for designers to access and use language.

MADAN MOHAN MALAVIYA UNIVERSITY OF TECHNOLOGY

DEVELOPMENT OF VERILOG

- The history of the Verilog HDL goes back to the 1980s, when a company called Gateway Design Automation developed a logic simulator, Verilog-XL, and with it a hardware description language.
- Cadence Design Systems acquired Gateway in 1989, Cadence put the language (but not the simulator) into the public domain.
- The Verilog HDL is now maintained by a non profit making organisation, Accellera, which was formed from the merger of Open Verilog International (OVI) and VHDL International. OVI had the task of taking the language through the IEEE standardisation procedure.
- In December 1995 Verilog HDL became IEEE Std. 1364-1995. A significantly revised version was published in 2001: IEEE Std. 1364-2001. There was a further revision in 2005 but this only added a few minor changes.

DESIGN ENTITIES IN VERILOG

- Design entities in Verilog includes modules , functions and tasks. Module define the input output and behaviour of a circuit, while functions are smaller units of code that can be called from within modules. Tasks are similar to function but can have multiple entry points and allow for more complex control flow
- Verilog supports a range of operators including arithmetic, relational and logical operators which can be used to perform various operations on digital signals.

Different styles of modelling-

- Structural – instantiation of primitives and modules.
- Data flow- continuous assignment.
- Behavioural- procedural assignment.

PROGRAM STRUCTURE

- Structure

```
module <module name> (port list>);
        <declares>
        <module items>
         endmodule
```

Module name

an identifier that uniquely name the module.

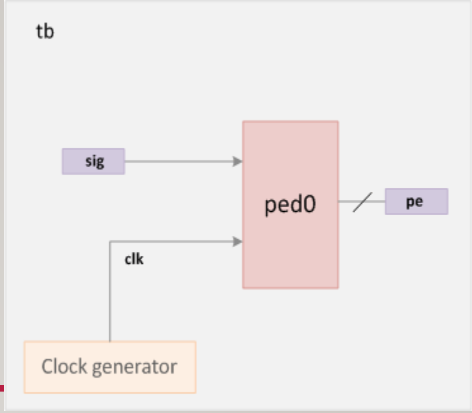
Port list

a list of input, inout and output ports which are used to other modules.

MADAN MOHAN MALAVIYA UNIVERSITY OF TECHNOLOGY

MADAN MOHAN MALAVIYA UNIVERSITY OF TECHNOLOGY

SIMULATION AND SYNTHESIS



- Simulation is a technique of applying different input stimulate to the design at different times to check if the RTL code behaves the intended way. It is also similar to how fabricated chip will be used in the real world.

- For example- the above design represents a edge detector with input clock and signal , simulation allows us to view the timing diagram of related signals to understand how the design description in Verilog actually behaves.

SIMULATION AND SYNTHESIS

MADAN MOHAN MALAVIYA UNIVERSITY OF TECHNOLOGY

- Verilog design can be simulated by using a range of tools. These tools allow designers to test their designs and identify potential errors before moving into synthesis.

- Synthesis is the process of converting a Verilog design into a gate level netlist, which can be used to program a physical device such as an FPGA. The synthesis process involves mapping Verilog constructs to specific hardware components, such as logic gates and flip- flops.

LIMITATION AND CONCLUSION

- Verilog have several limitations, including limitations on the types of constructs that can be synthesized and limited support for data types such as, array and strings. Verilog design may also be subject to timing constraints, which can affect the performance of the circuit.

- If we conclude all the given information we can say that Verilog is a powerful HDL language that is widely used in industry for designing digital circuits. It provides a range of design constructs and features that allow designers to create complex digital system. However Verilog also has several limitations that designers must be aware of when creating their designs.

VHDL

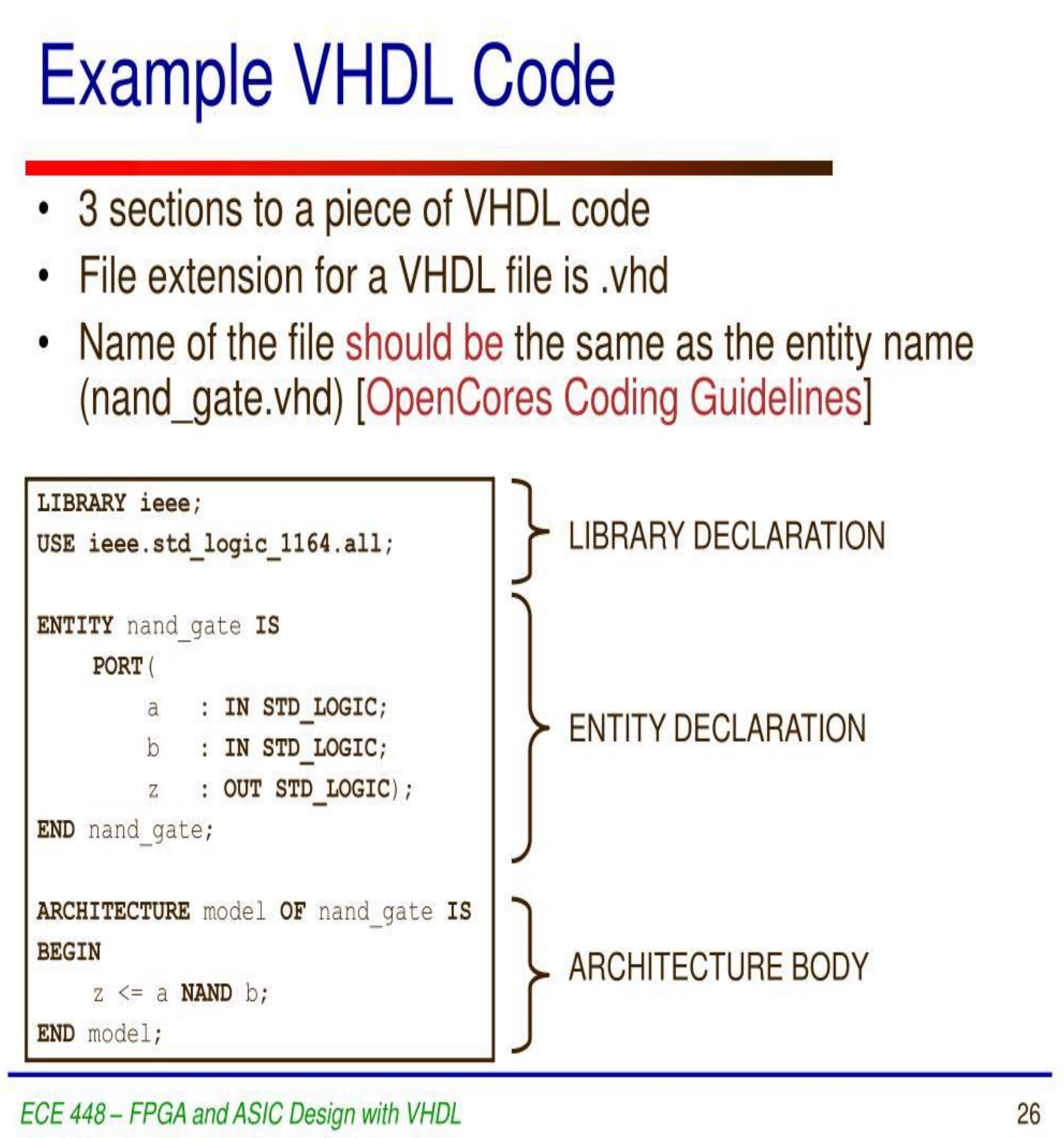
- VHDL stands for very high-speed integrated circuit hardware description language

Describing a Design

- In VHDL an entity is used to describe a hardware module. An entity can be described using,
- Entity declaration
- Architecture
- Configuration
- Package declaration
- Package body

MADAN MOHAN MALAVIYA UNIVERSITY OF TECHNOLOGY

VHDL EXAMPLE



Example VHDL Code

- 3 sections to a piece of VHDL code
- File extension for a VHDL file is .vhd
- Name of the file should be the same as the entity name (nand_gate.vhd) [OpenCores Coding Guidelines]

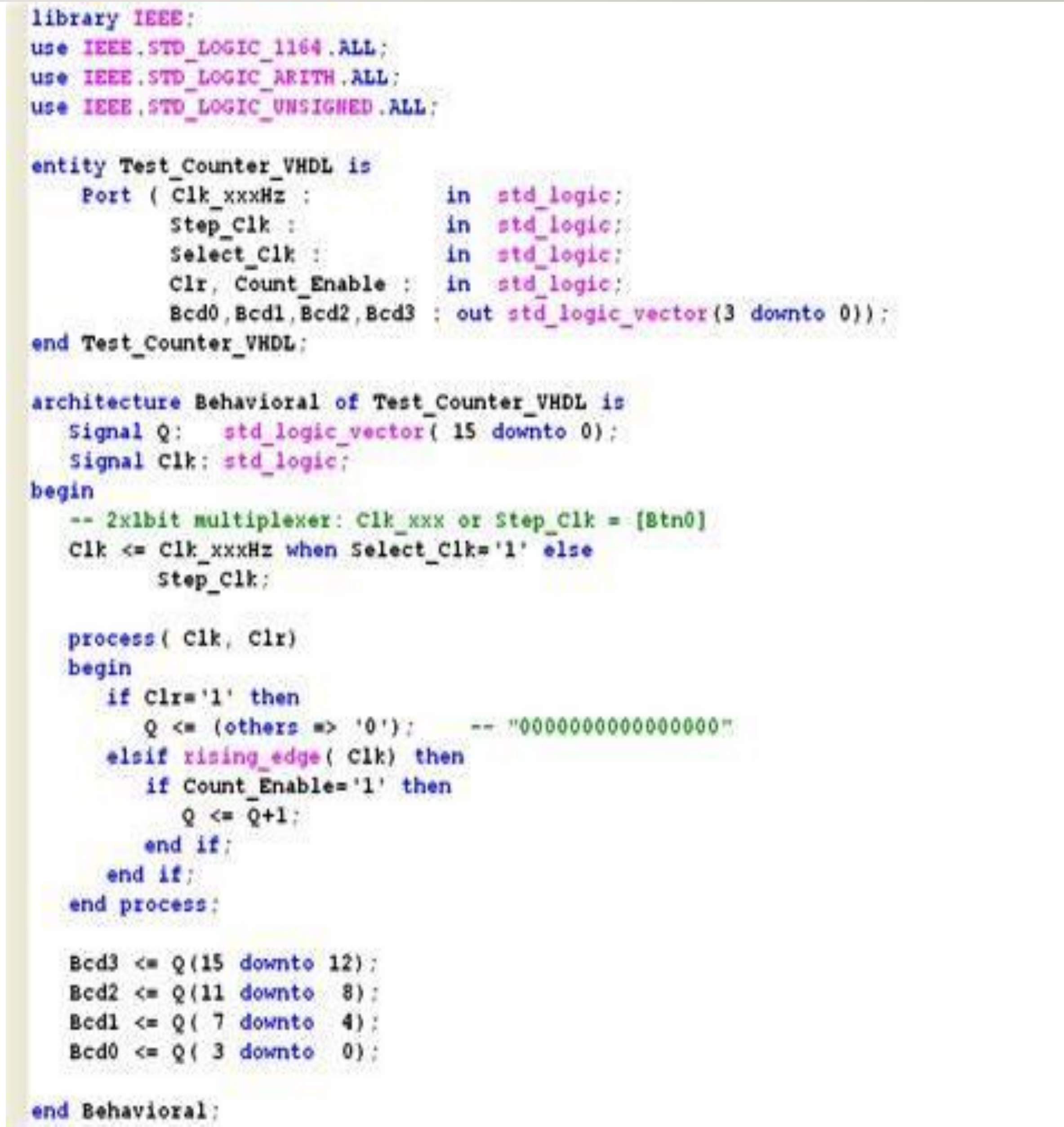
```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY nand_gate IS
     PORT(
          a    : IN STD_LOGIC;
          b    : IN STD_LOGIC;
          z    : OUT STD_LOGIC);
END nand_gate;

ARCHITECTURE model OF nand_gate IS
BEGIN
     z <= a NAND b;
END model;
```

LIBRARY DECLARATION
ENTITY DECLARATION
ARCHITECTURE BODY

ECE 448 – FPGA and ASIC Design with VHDL 26



```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity Test_Counter_VHDL is
    Port ( Clk_xxxHz :              in  std_logic;
           Step_Clk :               in  std_logic;
           Select_Clk :             in  std_logic;
           Clr, Count_Enable :      in  std_logic;
           Bcd0,Bcd1,Bcd2,Bcd3  : out std_logic_vector(3 downto 0));
end Test_Counter_VHDL;

architecture Behavioral of Test_Counter_VHDL is
   Signal Q:    std_logic_vector( 15 downto 0);
   Signal Clk: std_logic;
begin
   -- 2x1bit multiplexer: Clk_xxx or Step_Clk = [Btn0]
   Clk <= Clk_xxxHz when Select_Clk='1' else
          Step_Clk;

   process( Clk, Clr)
   begin
      if Clr='1' then
         Q <= (others => '0');     -- "0000000000000000"
      elsif rising_edge( Clk) then
         if Count_Enable='1' then
            Q <= Q+1;
         end if;
      end if;
   end process;

   Bcd3 <= Q(15 downto 12);
   Bcd2 <= Q(11 downto  8);
   Bcd1 <= Q( 7 downto  4);
   Bcd0 <= Q( 3 downto  0);

end Behavioral;
```

MADAN MOHAN MALAVIYA UNIVERSITY OF TECHNOLOGY

DIFFERENCE BETWEEN VHDL AND VERILOG

1.	Verilog is based on C language.	1.	VHDL is based on Ada and Pascal languages.
2.	Both Verilog and VHDL are Hardware Description Languages (HDL).	2.	Both Verilog and VHDL are Hardware Description Languages (HDL).
3.	Verilog is the latest language.	3.	VHDL is an older language
4.	Less complex	4.	More complex
5.	Case sensitive	5.	Not case sensitive
6.	Every Verilog program starts with the keyword “module” and ends with the keyword “endmodule”	6.	A hardware module in VHDL is called an entity. The syntax is as follows. The entity starts with “entity” and ends with “end” keyword.

MADAN MOHAN MALAVIYA UNIVERSITY OF TECHNOLOGY

DIFFERENCE BETWEEN VHDL AND VERILOG

7. module <module_name>
   (input,output);

      <program logic>

endmodule

7. entity <entity_name> is

      port declaration;

   end entity_name;

MADAN MOHAN MALAVIYA UNIVERSITY OF TECHNOLOGY

REFERENCES

- https://www.tutorialspoint.com/vlsi_design/vlsi_design_verilog_introduction.htm
- https://en.wikipedia.org/wiki/Hardware_description_language
- https://www.tutorialspoint.com/vlsi_design/vlsi_design_vhdl_introduction.htm
- https://resources.system-analysis.cadence.com/blog/msa2020-vlsi-programming-using-hardware-descriptive-languages

ANY QUERIES??

THANK YOU