DAILY ASSESSMENT FORMAT

Date:	**5th June 2020**	**Name:**	**Sahana S R**
Course:	**Digital design using HDL**	**USN:**	**4AL17EC083**
Topic:	**Verilog Tutorials and practice programs, Building/ Demo projects using FPGA**	**Semester & Section:**	**6th sem 'B' sec**
Github Repository:	**sahanasr-course**		

FORENOON SESSION DETAILS
Image of session

Introduction

Verilog is a **HARDWARE DESCRIPTION LANGUAGE (HDL)**. A hardware description Language is a language used to describe a digital system, for example, a network switch, a microprocessor or a memory or a simple flip–flop. This just means that, by using a HDL one can describe any hardware (digital) at any level.

One can describe a simple Filp flop as that in above figure as well as one can describe a complicated designs having 1 million gates. Verilog is one of the HDL languages available in the industry for designing the Hardware. Verilog allows us to design a Digital design at Behavior Level, Register Transfer Level (RTL), Gate level and at switch level. Verilog allows hardware designers to express their designs with behavioral constructs, deferring the details of implementation to a later stage of design in the final design.

Many engineers who want to learn Verilog, most often ask this question, how much time it will take to learn Verilog?, Well my answer to them is "**It may not take more then one week, if you happen to know at least one programming language**".

Design Styles

Verilog like any other hardware description language, permits the designers to design a design in either Bottom–up or Top–down methodology.

VERILOG TUTORIALS

Verilog is a Hardware Description Language; a textual format for describing electronic circuits and systems. Applied to elect for verification through simulation, for timing analysis, for test analysis (testability analysis and fault gra The Verilog HDL is an IEEE standard - number 1364. The first version of the IEEE standard for Verilog was published in 1995. A revised version was published in 2001; this is the version used by most Verilog users. The IEEE Verilog standard document is known as the Language Reference Manual, authoritative definition of the Verilog HDL.A further revision of the Verilog standard was published in 2005, though it has little extra compared to the 2001 standard. SystemVerilog is a huge set of extensions to Verilog, and was first published as an IEEE standard in 2005. See the ap SystemVerilog.IEEE Std 1364 also defines the Programming Language Interface, or PLI. This is a collection of software routines which permit between Verilog and other languages (usua Note that VHDL is not an abbreviation for Verilog HDL - Verilog and VHDL are two different HDLs. They have more similarities than differences, however. 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, and with it the rights to the language and the simulator. In 1990, Cadence put the language (but not the simulator) into the public domain, with the intention that it should become a standard, non-proprietary language.The Verilog HDL is now maintained by a non profit making organisation,

FPGA

FPGA Basics – A Look Under the Hood An introductory look inside Field Programmable Gate Arrays. We'll go over:Strengths & Weaknesses of FPGAs How FPGAs work What's inside an FPGA So you keep hearing about FPGAs being utilized in more and more applications, but aren't sure whether it makes sense to switch to a new technology. Or maybe you're just getting into the embedded world and want to figure out if an FPGA-based system makes sense for you or not.This paper provides an overview of some of the key elements of FPGAs for engineers interested in utilizing FPGA-based technologies. It's worth noting that this is a complex topic, and as such, some topics are not covered, some are just introductory, and others will evolve over time. This paper should still give you a lot of helpful information if you're new to the world of FPGAs.What are the most important things you should know right away?Get out of the software mindset – You're not writing software. Let me say that again because this is the single most important point if you're thinking about working with FPGAs.You-are-NOTwritingsoftware.You're designing a digital circuit

T-FLIP FLOP

```
 module tff (t,clk,q,qb);
 input t,clk;
output q,qb;
 reg q, qb;
 initial
 begin
 q=0;
 q=1;
 end
 always@(posedge (clk))
 begin
 if(t==0) q=q;
 else
q=qb;
 qb=~q;
end
 endmodule
```