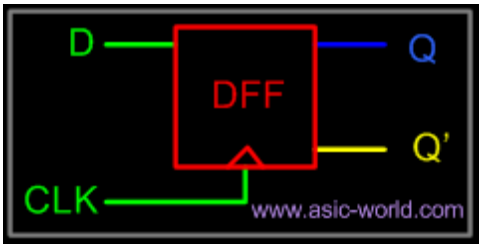
Introduction

Feb-9-2014

ASIC WORLD Directory
Verilog
Tutorial
Examples
Questions
Tools
Books
Links
FAQ
Sponsor
Home
Disclaimer
FAQ
Hot Jobs

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 (digital) hardware at any level.




```
1  // D flip-flop Code
2  module d_ff ( d, clk, q, q_bar);
3  input d ,clk;
4  output q, q_bar;
5  wire d ,clk;
6  reg q, q_bar;
7
8  always @ (posedge clk)
9  begin
10   q <= d;
11   q_bar <= !d;
12 end
13
14 endmodule
```

You could download file d_ff.v here

One can describe a simple Flip flop as that in the above figure, as well as a complicated design having 1 million gates. Verilog is one of the HDL languages available in the industry for hardware designing. It 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 in the final design.

Many engineers who want to learn this language, very often ask this question, how much time will it take to learn Verilog? Well

my answer to them is **"It may take no more than one week, if you happen to know at least one programming language"**.

Design Styles

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

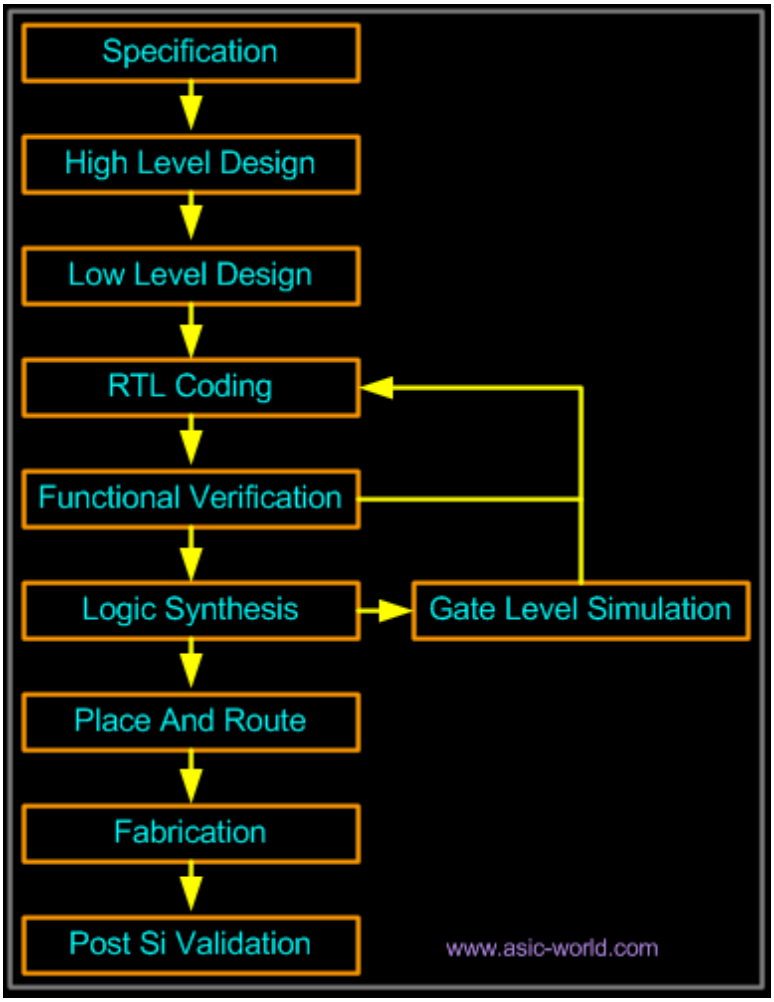
Bottom-Up Design

The traditional method of electronic design is bottom-up. Each design is performed at the gate-level using the standard gates (refer to the Digital Section for more details). With the increasing complexity of new designs this approach is nearly impossible to maintain. New systems consist of ASIC or microprocessors with a complexity of thousands of transistors. These traditional bottom-up designs have to give way to new structural, hierarchical design methods. Without these new practices it would be impossible to handle the new complexity.

Top-Down Design

The desired design-style of all designers is the top-down one. A real top-down design allows early testing, easy change of different technologies, a structured system design and offers many other advantages. But it is very difficult to follow a pure top-down design. Due to this fact most designs are a mix of both methods, implementing some key elements of both design styles.

Figure shows a Top-Down design approach.



Verilog Abstraction Levels

Verilog supports designing at many different levels of abstraction. Three of them are very important:

- Behavioral level
- Register-Transfer Level
- Gate Level

Behavioral level

This level describes a system by concurrent algorithms (Behavioral). Each algorithm itself is sequential, that means it consists of a set of instructions that are executed one after the other. Functions, Tasks and Always blocks are the main elements. There is no regard to the structural realization of the design.

Register-Transfer Level

Designs using the Register-Transfer Level specify the characteristics of a circuit by operations and the transfer of data between the registers. An explicit clock is used. RTL design contains exact timing bounds: operations are scheduled to occur at certain times. Modern RTL code definition is "Any code that is synthesizable is called RTL code".

Gate Level

Within the logic level the characteristics of a system are described by logical links and their timing properties. All signals are discrete signals. They can only have definite logical values (`0', `1', `X', `Z`). The usable operations are predefined logic primitives (AND, OR, NOT etc gates). *Using gate level modeling might not be a good idea for any level of logic design. Gate level code is generated by tools like synthesis tools and this netlist is used for gate level simulation and for backend.*

Do you have any Comment? mail me at:deepak@asic-world.com