

Date:	02-06-2020	Name:	Rajeshwari Gadagi
Course:	HDL	USN:	4AL17EC076
Topic:	Fpga basics.verilog HDL basics and testbench code	Semester and section:	6 th sem and B sec

Verilog History

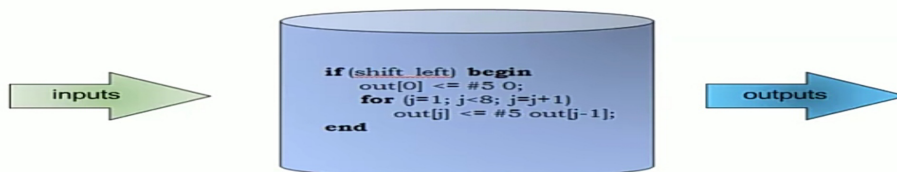
- Introduced in 1984 by Gateway Design Automation
- 1989 Cadence purchased Gateway and subsequently released Verilog to the public
- Open Verilog International (OVI) was formed to control the language specifications
- 1995 IEEE accepted OVI Verilog as a standard
- 2001 and 2005 IEEE revised standard
- 2009 Merged with SystemVerilog becoming IEEE Standard 1800-2009

© 2013 Altera Corporation - Public

ALTERA
MEASURABLE ADVANTAGE™

Behavior Modeling

- Only the functionality of the circuit, no structure
- Synthesis tool creates correct logic

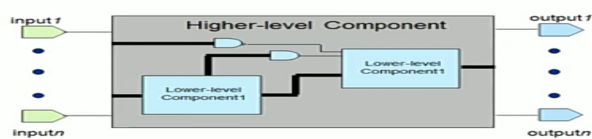


© 2013 Altera Corporation - Public

ALTERA
MEASURABLE ADVANTAGE™

Structural Modeling

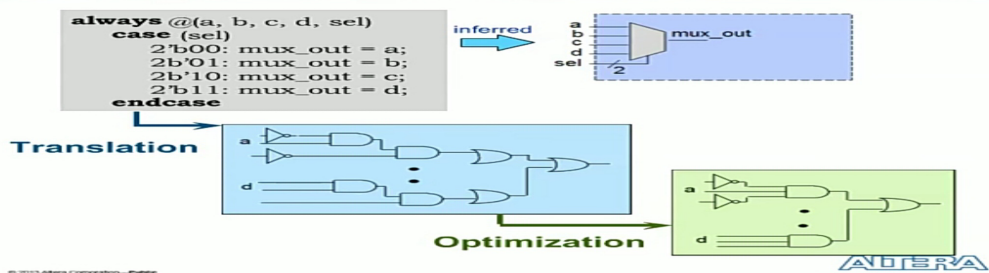
- Functionality and structure of the circuit
- Call out the specific hardware



© 2013 Altera Corporation - Public

ALTERA
MEASURABLE ADVANTAGE™

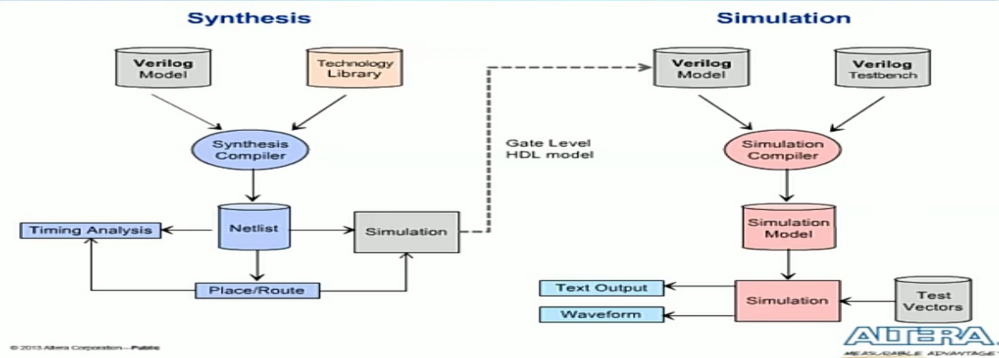
RTL Synthesis



© 2013 Altera Corporation - Public

ALTERA
MEASURABLE ADVANTAGE™

Typical RTL Synthesis & RTL Simulation Flows



FPGA Basics: Architecture, Application and Uses

- FPGA is an integrated circuit that consists of internal hardware blocks with user-programmable interconnects to customize operation for specific application.
- The basic FPGA architecture consists of thousands of fundamental elements called configurable logic blocks (CLBs) surrounded by a system of programmable interconnects, called a fabric, I/O blocks and external devices.
- Applications - Image processing, Artificial intelligence, data centre hardware accelerators, enterprise networking.

Verilog HDL Basics

- Modeling Structure -
 - module name (port list);
 - port declarations;
 - data type declarations;
 - circuit functionality;
 - timing specifications;

Verify Design Under Test

```

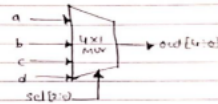
module testbench;
    reg a, b, c; wire sum, cout;
    full_adder FA (sum, cout, a, b, c);
    initial
    begin
        $monitor ($time, "a=%b, b=%b, c=%b, sum=%b, cout=%b",
            a, b, c, sum, cout);
        #5 a=0; b=0; c=1;
        #5 b=1;
        #5 a=1;
    end
endmodule
    
```

```

#s a=0; b=0; c=0;
#5 $finish;
end
endmodule
    
```

Task for Day 2:

- 4:1 MUX:



Testbench code:

```

module tb_4to1_mux;
    reg [3:0] a;
    reg [3:0] b;
    reg [3:0] c;
    reg [3:0] d;
    wire [3:0] out;
    reg [1:0] sel;
    integer i;
    mux_4to1_mux m1 (a, b, c, d, out, sel);
    initial begin
        $monitor ("Time: %t, sel=%b, a=%b, b=%b, c=%b, d=%b, out=%b",
            $time, sel, a, b, c, d, out);
        #10 sel <= 0;
        a <= $random;
        b <= $random;
        c <= $random;
        d <= $random;
    end
endmodule
    
```

```

2)
3)
for (i=1; i<4; i=i+1) begin
    #5 sel <= i;
end
#5 $finish;
end
endmodule
    
```

Date:	02-06-2020	Name:	Rajeshwari Gadagi
Course:	Python programming	USN:	4AL17EC076
Topic:	Bokeh, webscraping with python beautiful soup.	Semester and section:	6 th sem and B sec

Interactive Data Visualization with Bokeh -

- Bokeh is the library
- We can grab data from the value using Bokeh
- Installing Bokeh -
 pip install bokeh
 or pip3 install bokeh
- Snippet producing the triangle based plot :-

```
from bokeh.plotting import figure
from bokeh.io import output_file, show

x = [3, 7, 5, 10]
y = [3, 6, 9]
output_file("line.html")
f = figure()
f.triangle(x, y) [for circle based, f.circle(x, y)]
show(f)
```
- Plotting educational data.

```
from bokeh.plotting import figure
from bokeh.io import output_file, show
import pandas
df = pandas.read_csv("http://pythonhow.com/data/bachelors.csv")
x = df["Year"]
y = df["Engineering"]
output_file("line_from_bachelors.html")
f = figure()
f.line(x, y)
show(f)
```
- Loading Excel files -
 pip install xlrd or pip3 install xlrd

Webscraping with Python Beautiful Soup :-

- Request headers

```
r = requests.get("http://www.pythonhow.com/example.html")
```
- import requests
 from bs4 import BeautifulSoup

```
r = requests.get("http://pythonhow.com/examples.html")
c = r.content
c
soup = BeautifulSoup(c, "html.parser")
all = soup.find_all("div", {"class": "cities"})
all[0].find_all("h2")[0].text
for item in all:
    print(item.find_all("h2")[0].text)
```