

Date:	02-06-2020	Name:	POOJA K S
Course:	HDL	USN:	4AL17EC070
Topic:	FPGA Basics: Architecture, Applications and Uses, Verilog HDL Basics by Intel, Verilog Testbench code to verify the design under test (DUT)	Semester and section	6 th sem 'B' section
Github repository:	pooja-shivanna		

What is FPGA? FPGA Ba

arrow.com/en/research-and-events/articles/fpga-basics-architecture-applications-and-uses

ARROW Products Manufacturers Datasheets Reference Designs Articles and Videos Design Center BOM Tool Hire an Engineer

FPGA Architecture

A basic FPGA architecture (Figure 1) consists of thousands of fundamental elements called configurable logic blocks (CLBs) surrounded by a system of programmable interconnects, called a fabric, that routes signals between CLBs. Input/output (I/O) blocks interface between the FPGA and external devices.

Depending on the manufacturer, the CLB may also be referred to as a logic block (LB), a logic element (LE) or a logic cell (LC).

We've updated our privacy policy. Please take a moment to review these changes. By clicking I Agree to Arrow Electronics Terms Of Use and have read and understand the Privacy Policy and Cookie Policy.

READ MORE ACCEPT TERMS

Activate Windows Go to Settings to activate Windows. Spark Virtual Assistant

SEE SHORT POLICY

Search the web and Windows

11:57 AM 6/2/2020

Verilog HDL Basics - youtube.com/watch?v=PjGvZSIsLKs&list=PL0pUShg9yniZ2ka-XBXROXNR0pAEAEFCB&index=4

* Watch YouTube videos with Chrome. Yes, get Chrome now.

YouTube IN Search

⚠ We'll stop supporting this browser soon. For the best experience please update your browser.

Module and Port Declaration

- Module Declaration:**
 - Begins with keyword **module**
 - Provides module name
 - Includes port list, if any
- Port Types:**
 - input** => input port
 - output** => output port
 - inout** => bidirectional port
- Port Declarations:**

```
<port_type> <port_name>;
```

```
module mult_acc (out,
    ina, inb,
    clk, aclr);
    input [7:0] ina, inb;
    input clk, aclr;
    output [15:0] out;
    ...
endmodule
```

ALTERA MFGABLE ADVANTAGE™

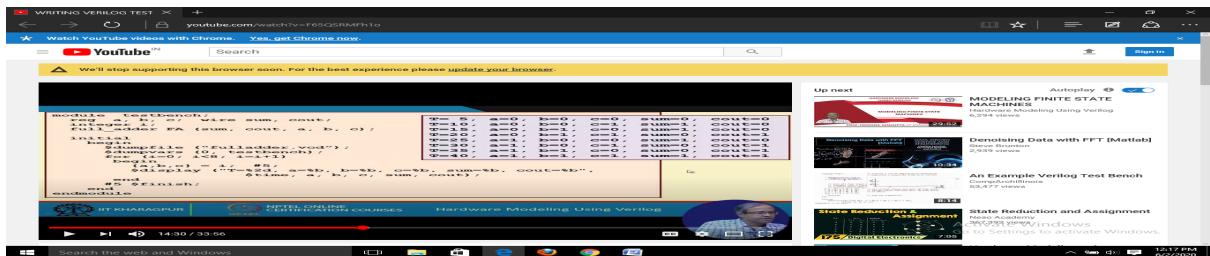
FPGA Design Intel FPGA + 4/185 videos

- Verilog HDL Basics Intel FPGA
- VHDL Basics Intel FPGA
- ModelSim®-Intel FPGA Edition Simulation with intel Quartus Prime Pro Edition Intel FPGA
- Generating PHYLite example design simulation in ModelSim in 16.1 with Arria10 Intel FPGA
- SignalTap II Logic Analyzer: Introduction & Getting Started Intel FPGA
- SignalTap II Logic Analyzer: Basic Trigger Conditions & Configuration Intel FPGA

12:12 PM 6/2/2020



Edit with WPS Office



Edit with WPS Office

02/06/2020

STYL
Date: _____
Page No: _____

Day - 2

HDL

FPGA basics: Architecture applications and uses

The field-programmable gate array (FPGA) is an integrated circuit that consists of internal hardware blocks with user programmable interconnects to customize operation for a specific application.

What is FPGA:

Field-programmable gate array

FPGA architecture

FPGA design

library ieee;

use ieee.std_logic_1164.all;

use ieee.numeric_std.all;

entity signed_adder is

port

(

aclr: in std_logic;

clk: in std_logic;

a: in std_logic_vector;

b: in std_logic_vector;

q: out std_logic_vector);



end signed-adder;

architecture signed-adder-arch of
signed-adder is
signal q, s: signed(a'high & down to 0);

begin

assert (a'length ≤ b'length)

report "Port A must be the longer
vector if different size!"

severity FAILURE;

q <= std-logic-vector(q,s);

adding-proc:

process (aclr, clk)

begin

if (aclr = '1') then

q,s <= (others => '0');

elsif rising-edge (clk) then

q,s <= ('0' & signed(a)) + ('0' & signed
(b));

end if; -- clk'd

end process;

end signed-adder-arch;

* FPGA Use: An Attractive choice
for certain applications

* FPGA Applications



Edit with WPS Office

VHDL HDL Basic by Intel

VHDL

- * IEEE Industry Standard Hardware Description Language (HDL) - used to describe a digital system
- * use in both hardware simulation and synthesis

Behavior modeling

- * only the functionality of the circuit, no structure
- * Synthesis tool creates cost logic

```

→      if (shift & left)
Inputs  begin out[0]
        <= #5'0;
        for (i = 1; i <
                j & j + 1)
                out[i] <= #5
                out[i] <= out[i - 1];
end
→      output
    
```

VHDL

Test bench code to verify the design under the (DUT)

Full adder

```

module full-adder (S, CO, a, b, c);
    input a, b, c;
    output S, CO;
    assign S = a ^ b ^ c;
    
```



```
assign CO = (a&b).i(b&c) | (c&a);  
endmodule
```

4 bit shift register :

```
module shiftreg_4bit (clock, clear, A, E);  
    input clock, clear, A;  
    output reg G;  
    reg B, C, D;  
    always @ (posedge clock or negedge  
    clear)  
    begin  
        if (!clear) begin B <= 0; C <= 0;  
        D <= 0; E <= 0; end  
        else begin  
            E <= D;  
            D <= C;  
            C <= B;  
            B <= A;  
        end  
    end  
endmodule
```

Automatic verification of output

```
module fulladder (a, b, c, S, cout);  
    input a, b, c;  
    output S, cout;  
    assign S = a' b' c;  
    assign cout = (a&b) | (b&c) | (c&a);  
endmodule.
```



Date:	02-06-2020	Name:	POOJA K S
Course:	Python programming	USN:	4AL17EC070
Topic:	Interactive data visualization with bokeh and Webscraping with python beautiful soup	Semester and section:	6 th sem and B sec

The Python Mega Course: Build 10 Real World Applications

Course content

- 226. Introduction to Bokeh
- 227. Installing Bokeh
- 228. Your First Bokeh Plot
- 229. Plotting Triangles and Circle Glyphs (Practice)
- 230. Solution
- 231. Using Bokeh with Pandas
- 232. Plotting Education Data (Practice)
- 233. Solution
- 234. Note on Loading Excel Files

Solution

```

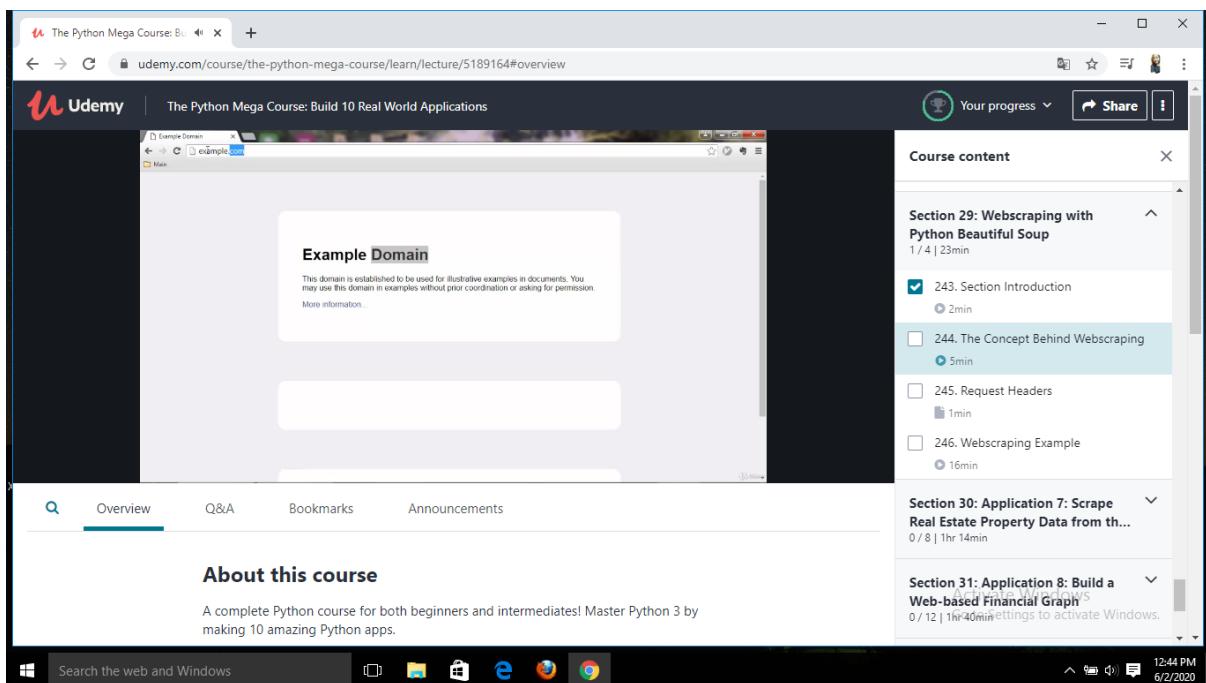
1 import pandas
2
3 from bokeh.plotting import figure, output_file, show
4
5 df=pandas.read_excel("http://pythonhow.com/data/verlegenhuken.xlsx",
6 df["Temperature"]*df["Temperature"]/10
7 df["Pressure"]*df["Pressure"]/10
8
9 p=figure(plot_width=500,plot_height=400,tools='pan')
10
11 p.title.text="Temperature and Air Pressure"
12 p.title.text_color="Gray"
13 p.title.text_font="arial"
14 p.title.text_font_style="bold"
15 p.xaxis.minor_tick_line_color=None
16 p.yaxis.minor_tick_line_color=None
17 p.xaxis.axis_label="Temperature (°C)"
18 p.yaxis.axis_label="Pressure (hPa)"
19
20 p.circle(df["Temperature"],df["Pressure"],size=0.5)
21 output_file("Weather.html")
22 show(p)

```

Activate Windows
Go to Settings to activate Windows.



Edit with WPS Office



Request Headers

Note

When I use this code in the next video:

```
r = requests.get("http://www.pythonhow.com/example.html")
```

please use this instead:

```
r = requests.get("http://www.pyclass.com/example.html",
headers={'User-agent': 'Mozilla/5.0 (X11; Ubuntu; Linux
x86_64; rv:61.0) Gecko/20100101 Firefox/61.0'})
```

The rest of the code stays the same.

So, we're just changing the domain name from `pythonhow` to `pyclass` and we're adding a `header` argument. Some webpages don't like scripts sometimes, so adding a header allows the script to impersonate a web browser.

Activate Windows
Go to Settings to activate Windows.



Edit with WPS Office



Edit with WPS Office

02/06/2020

Day - 14

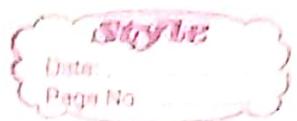
Interactive Data visualization
with Bokeh
and

web scraping with Python
Beautiful Soup

Interactive Data visualization with
Bokeh

- * Introduction to Bokeh
- * Installing Bokeh
- * your first Bokeh plot
- * plotting triangles and circle glyphs
- * Solution
- * using Bokeh with pandas
- * plotting Education Data (practice)
- * Solution
- * Note on Loading excel files
- * plot properties
- * plot properties
- * plotting Weather Data
- * Solution
- * visual attributes
- * Time-series plots
- * more visualization examples with bokeh
- * plotting time intervals of the Motion Detector
- * Hover tool implementation





web scraping with python beautiful soup :

- ★ Section Introduction
- ★ The concept Behind web scraping
- ★ Request Headers
- ★ web scraping example

