

DAILY ASSESSMENT FORMAT

Date:	5-06-2020	Name:	BINDUSHRI
Course:	Digital design using HDL	USN:	4AL17EC011
Topic:	1.verilog tutorial class 2.FPGA projects 3.task		6th A
Github Repository:	Bindushri		

FORENOON SESSION DETAILS

The screenshot shows a Google Drive document titled 'Types of Nets' and 'Register Data Types'. The document is open in a web browser, and the URL is visible at the top: drive.google.com/file/d/15qKyH1Zky_NpOOMR_Dqldy0c8F3MGMaB/view. The document content includes a table of net types and their functionalities, followed by a section on register data types.

Types of Nets

Each net type has functionality that is used to model different types of hardware (such as PMOS, NMOS, CMOS, etc)

Net Data Type	Functionality
wire, tri	Interconnecting wire – no special resolution function
wor, trior	Wired outputs OR together (models ECL)
wand, triand	Wired outputs AND together (models open-collector)
tri0, tri1	Net pulls-down or pulls-up when not driven
supply0, supply1	Net has a constant logic 0 or logic 1 (supply strength)
tri0reg	

Note : Of all the net types, wire is the one which is most widely used

Register Data Types

- Registers store the last value assigned to them until another assignment statement changes their value.
- Registers represent data storage constructs.
- You can create arrays of the regs called memories.
- register data types are used as variables in procedural blocks.
- A register data type is required if a signal is assigned a value within a procedural block

www.asic-world.com VERILOG HDL SYNTAX AND SEMANTICS 42

• Procedural blocks begin with keyword initial and always.

Data Types Page 42 / 227 **Functionality** +

zon I ineinnet variable 20:43

Day 5 Verilog Tutorials & Practice Programs



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. endmodule.

- Bottom-up Design
- Top down Design
- Hierarchy of Verilog
- various stages of ASIC/FPGA.

- Specification
- High level Design
- Logic Design / Low level Design
- RTL coding
- Simulation
- Synthesis
- Place & Route
- Post si validation

I Basic program

```
module hexo_decod;
```

```
    begin
```

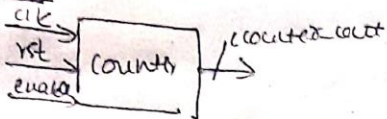
```
        display ("hexo decod by Delbak");
```

```
    end
```

```
end
```

```
endmodule;
```

Counter



- 4-bit synchronous up counter
- active high, synchronous reset
- active high enable

Mux

```
module mux_2to1 (a,b,sel,y);
```

```
    input a,b;
```

```
    output y;
```

```
    input sel;
```

```
    wire y;
```

```
    assign y = (sel) ? b : a;
```

```
endmodule
```

* if-else, case, while same as c

↳ if-else & case statements require all the cases to be covered for combinational logic

↳ for-loop same as c but no ++ & -- operators.

→ FPGA projects

→ Task - day 5

Implement a Verilog module to count number of 0's in a 16-bit number.

```
→ module num_0s_for (
    input [15:0] A,
    output reg [4:0] ones
);
```

integer i;

always @(A) begin

~~reg~~

count = 0;

for (i = 0; i < 16; i = i + 1)

if (A[i] == 1'b0)

count = count + 1;

end

endmodule.

--

Date:5june2020

Course: python

Topic:sec 33-34

Name:Bindushri

USN:4AL17EC011

Sem&Sec:6th A

AFTERNOON SESSION DETAILS

The Python Mega Course: Build 10 Real World Applications

Your progress ▾Share ⓘ

Lists (Practice)

Assign a list of three items to `myList`. The items can be any type of data.

[View Solution](#)

exercise.py

1 myList = [1,2,3]

Line 1, Column 16All changes savedReset code

Well done, your solution is correct!

Check solutionContinue⚙️🔍📺

Overview

Q&A

Notes

Announcements

About this course

Course content

☒ Coding Exercise 55: Lists (Practice)

☐ Coding Exercise 56: Indexing (Practice)

☐ Coding Exercise 57: Slicing (Practice)

☐ Coding Exercise 58: More on Indexing (Practice)

☐ Coding Exercise 59: More on Slicing (Practice)

☐ Coding Exercise 60: List Indexing (Practice)

☐ Coding Exercise 61: Append to List (Practice)

☐ Coding Exercise 62: Remove from List (Practice)

☐ Coding Exercise 63: Append from List to List (Practice)

☐ Coding Exercise 64: Concatenate List Items (Practice)

☐ Coding Exercise 65: Create Dictionary (Practice)

☐ Coding Exercise 66: Create Function (Practice)

The Python Mega Course: Build 10 Real World Applications

Your progress ▾Share ⓘ

End of the Course

Ladies and gentlemen, congratulations on completing the course! I wanted to tell you that this is a huge achievement that not everyone has the willpower to do. I can see that from the course statistics.

I am sure this is a huge step to kickstarting your programming career. I am very happy you were my student and followed everything I had to teach you throughout this long course which I have created with a lot of commitment and passion.

I wish you great success in your future projects and hope to have given you a positive push in your endeavors!

⚙️🔍📺

Overview

Q&A

Notes

Announcements

About this course

A complete Python course for both beginners and intermediates! Master Python 3 by making 10 amazing Python apps.

Course content

Section 32: Application 10: Build a Data Collector Web App with PostGreSQL and ...
11 / 11 | 2hr 47min

Section 33: Application 11: Project Exercise on Building a Geocoder Web Ser...
4 / 4 | 30min

☒ 269. Student Project - How The Output Should Look Like
8minResources ▾

☒ 270. Solution, Part 1
16min

☒ 271. Solution, Part 2
6min

☒ 272. End of the Course
1min

Section 34: Legacy Exercises
0 / 20 | 0min

Section 35: Offers for my Other Python Courses
0 / 1 | 1min

5-06-2020 Friday
Section 33 Building a Groceries web server

creating 4 folders
app1.py, app2.py, app3.py, app4.py

* Modification can be done in previous sec code

* Import datetime module to generate the executor.

* This will make python to generate the CSV file at the o/p.

*

Exercise

1) Create a variable named character and assign john snow

→ character = "Johnsnow"

2) Create a variable named price and assign it value of 10.

In the next line print the value of variable price.

→ price = 10,
print(price)

Certificate :



