

# DAILY ASSESSMENT FORMAT

Date:	05/06/2020	Name:	Lavanya B
Course:	Verilog HDL	USN:	4a117ec043
Topic:	Verilog Building project using FPGA	Semester & Section:	6th A
Github Repository:	Lavanya-B		

## FORENOON SESSION DETAILS

### Image of session



#### ❖ Fall Delay

The fall delay is associated with a gate output transition to 0 from a



#### ❖ Turn-off Delay

The Turn-off delay is associated with a gate output transition to z if

#### ❖ Min Value

The min value is the minimum delay value that the gate is expected

#### ❖ Typ Value

The typ value is the typical delay value that the gate is expected to

#### ❖ Max Value

The max value is the maximum delay value that the gate is expected

#### ❖ Examples

## FPGA Projects

This page presents FPGA projects on [fpga4student.com](http://fpga4student.com). The first FPGA project helps students understand the basics of FPGAs and how Verilog/ VHDL works on [FPGA](#).



Some of the FPGA projects can be FPGA tutorials such as [What is FPGA Programming](#), [image processing on FPGA](#), [matrix multiplication on FPGA](#) Xilinx using Core Generator, [Verilog vs VHDL: Explain by Examples](#) and [how to load text files or images into FPGA](#). Many others FPGA projects provide students with full Verilog/ VHDL source code to practice and run on FPGA boards. Some of them can be used for another bigger FPGA projects.

Following are the [FPGA](#) projects on [fpga4student.com](http://fpga4student.com).

Report

## Verilog

It is a hardware description language

Design styles

- bottom-up design
- top-down design

Abstraction levels of Verilog

- Behavioral level
- Register level
- Gate level

Various stages of ASIC/FPGA

- Specification
- High level Design
- Macro Design / low level Design
- RTL coding
- Simulation
- Synthesis
- Place & Route
- Post SI validation

Data types

NETS

Registers

Explicitly declared

Implicitly declared - wire

Gate level modelling

module gate1

input out;

input out1;

input out2;

input in1, in2, in3, in4;

not out(out1, in1);

and out2(out1, in1, in2, in3, in4);

xor out3(out1, in1, in2, in3);

initial begin

inout1 = 0; in2 = 1; in3 = 0; in4 = 1; out1 = 0; out2 = 1; out3 = 0;

end

inout1 = 0;

inout2 = 0;

inout3 = 0;

inout4 = 0;

inout5 = 0;

inout6 = 0;

inout7 = 0;

inout8 = 0;

inout9 = 0;

inout10 = 0;

inout11 = 0;

inout12 = 0;

inout13 = 0;

inout14 = 0;

inout15 = 0;

inout16 = 0;

inout17 = 0;

inout18 = 0;

inout19 = 0;

inout20 = 0;

inout21 = 0;

inout22 = 0;

inout23 = 0;

inout24 = 0;

inout25 = 0;

inout26 = 0;

inout27 = 0;

inout28 = 0;

inout29 = 0;

inout30 = 0;

inout31 = 0;

inout32 = 0;

inout33 = 0;

inout34 = 0;

inout35 = 0;

inout36 = 0;

inout37 = 0;

inout38 = 0;

inout39 = 0;

inout40 = 0;

inout41 = 0;

inout42 = 0;

inout43 = 0;

inout44 = 0;

inout45 = 0;

inout46 = 0;

inout47 = 0;

inout48 = 0;

inout49 = 0;

inout50 = 0;

inout51 = 0;

inout52 = 0;

inout53 = 0;

inout54 = 0;

inout55 = 0;

inout56 = 0;

inout57 = 0;

inout58 = 0;

inout59 = 0;

inout60 = 0;

inout61 = 0;

inout62 = 0;

inout63 = 0;

inout64 = 0;

inout65 = 0;

inout66 = 0;

inout67 = 0;

inout68 = 0;

inout69 = 0;

inout70 = 0;

inout71 = 0;

inout72 = 0;

inout73 = 0;

inout74 = 0;

inout75 = 0;

inout76 = 0;

inout77 = 0;

inout78 = 0;

inout79 = 0;

inout80 = 0;

inout81 = 0;

inout82 = 0;

inout83 = 0;

inout84 = 0;

inout85 = 0;

inout86 = 0;

inout87 = 0;

inout88 = 0;

inout89 = 0;

inout90 = 0;

inout91 = 0;

inout92 = 0;

inout93 = 0;

inout94 = 0;

inout95 = 0;

inout96 = 0;

inout97 = 0;

inout98 = 0;

inout99 = 0;

inout100 = 0;

inout101 = 0;

inout102 = 0;

inout103 = 0;

inout104 = 0;

inout105 = 0;

inout106 = 0;

inout107 = 0;

inout108 = 0;

inout109 = 0;

inout110 = 0;

inout111 = 0;

inout112 = 0;

inout113 = 0;

inout114 = 0;

inout115 = 0;

inout116 = 0;

inout117 = 0;

inout118 = 0;

inout119 = 0;

inout120 = 0;

inout121 = 0;

inout122 = 0;

inout123 = 0;

inout124 = 0;

inout125 = 0;

inout126 = 0;

inout127 = 0;

inout128 = 0;

inout129 = 0;

inout130 = 0;

inout131 = 0;

inout132 = 0;

inout133 = 0;

inout134 = 0;

inout135 = 0;

inout136 = 0;

inout137 = 0;

inout138 = 0;

inout139 = 0;

inout140 = 0;

inout141 = 0;

inout142 = 0;

inout143 = 0;

inout144 = 0;

inout145 = 0;

inout146 = 0;

inout147 = 0;

inout148 = 0;

inout149 = 0;

inout150 = 0;

inout151 = 0;

inout152 = 0;

inout153 = 0;

inout154 = 0;

inout155 = 0;

inout156 = 0;

inout157 = 0;

inout158 = 0;

inout159 = 0;

inout160 = 0;

inout161 = 0;

inout162 = 0;

inout163 = 0;

inout164 = 0;

inout165 = 0;

inout166 = 0;

inout167 = 0;

inout168 = 0;

inout169 = 0;

inout170 = 0;

inout171 = 0;

inout172 = 0;

inout173 = 0;

inout174 = 0;

inout175 = 0;

inout176 = 0;

inout177 = 0;

inout178 = 0;

inout179 = 0;

inout180 = 0;

inout181 = 0;

inout182 = 0;

inout183 = 0;

inout184 = 0;

inout185 = 0;

inout186 = 0;

inout187 = 0;

inout188 = 0;

inout189 = 0;

inout190 = 0;

inout191 = 0;

inout192 = 0;

inout193 = 0;

inout194 = 0;

inout195 = 0;

inout196 = 0;

inout197 = 0;

inout198 = 0;

inout199 = 0;

inout200 = 0;

inout201 = 0;

inout202 = 0;

inout203 = 0;

inout204 = 0;

inout205 = 0;

inout206 = 0;

inout207 = 0;

inout208 = 0;

inout209 = 0;

inout210 = 0;

inout211 = 0;

inout212 = 0;

inout213 = 0;

inout214 = 0;

inout215 = 0;

inout216 = 0;

inout217 = 0;

inout218 = 0;

inout219 = 0;

inout220 = 0;

inout221 = 0;

inout222 = 0;

inout223 = 0;

inout224 = 0;

inout225 = 0;

inout226 = 0;

inout227 = 0;

inout228 = 0;

inout229 = 0;

inout230 = 0;

inout231 = 0;

inout232 = 0;

inout233 = 0;

inout234 = 0;

inout235 = 0;

inout236 = 0;

inout237 = 0;

inout238 = 0;

inout239 = 0;

inout240 = 0;

inout241 = 0;

inout242 = 0;

inout243 = 0;

inout244 = 0;

inout245 = 0;

conditional statements (if-else)  
 for loop statement  
 looping statement - while, for  
 conditional assignment statement  
 Propagation Delay  
 Hierarchical Block control  
 Named Block

#### Task & Functions

Sg: module task\_globals;  
 reg [3:0] temp\_out;  
 reg [3:0] temp\_in;  
 task count;  
 begin  
 temp\_out = (9[0] \* temp\_in);  
 end  
endtask  
endmodule

Sg: module simple\_function;  
 function myfunction;  
 input a, b, c, d;  
 begin  
 myfunction = ((a+b) \* (c-d));  
 end  
endfunction  
endmodule

#### System Task & function

\$display, \$strobe, \$monitor  
 \$time, \$stime, \$realtime  
 \$size, \$stop, \$finish  
 \$scope, \$scopelevel  
 \$random

#### Writing testbenches

Sg: counter  
 module counter\_tb;  
 reg clk, reset, enable;  
 wire [3:0] count;  
 counter uut (.clk(clk), .reset(reset), .enable(enable), .count(count));  
 initial begin  
 clk = 0;  
 reset = 0;  
 enable = 0;  
 end  
 always  
 #10 clk = ~clk;  
 initial begin  
 \$dumpfile("counter.vcd");  
 \$dumpvars;  
 end  
 initial begin  
 \$display("Reset time, rate, reset, enable, count");  
 \$monitor("%d, %d, %d, %d, %d", \$time, clk, reset, enable, count);  
 end  
 initial  
 #100 finish;  
endmodule

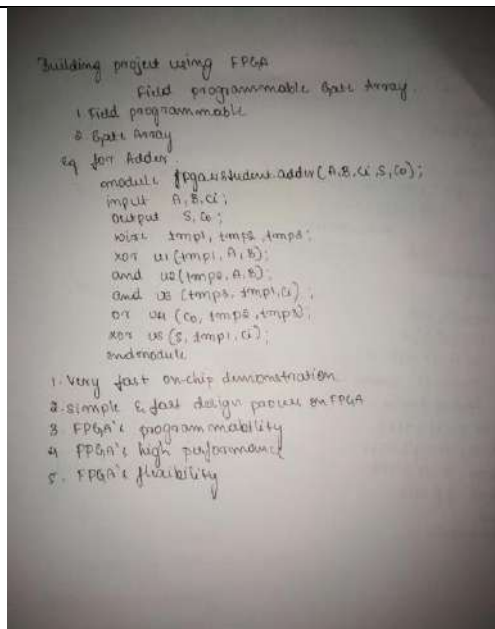
#### Modeling memory and FSM

Syntax: reg[wordsize] array\_name[0:arraysize]  
 Storing values, Reading values, Bit read

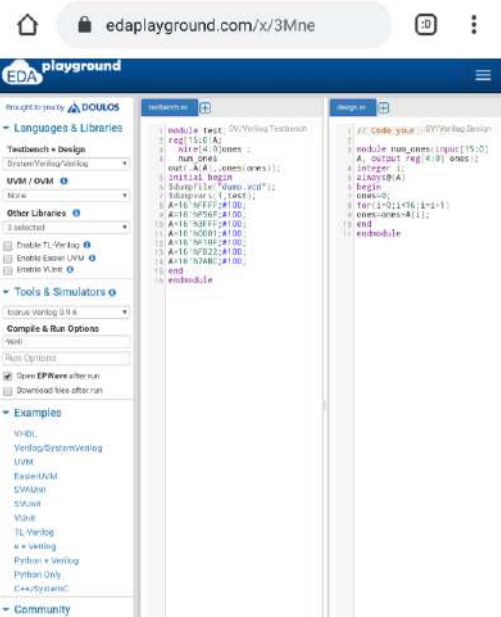
#### Mealy & Moore Model

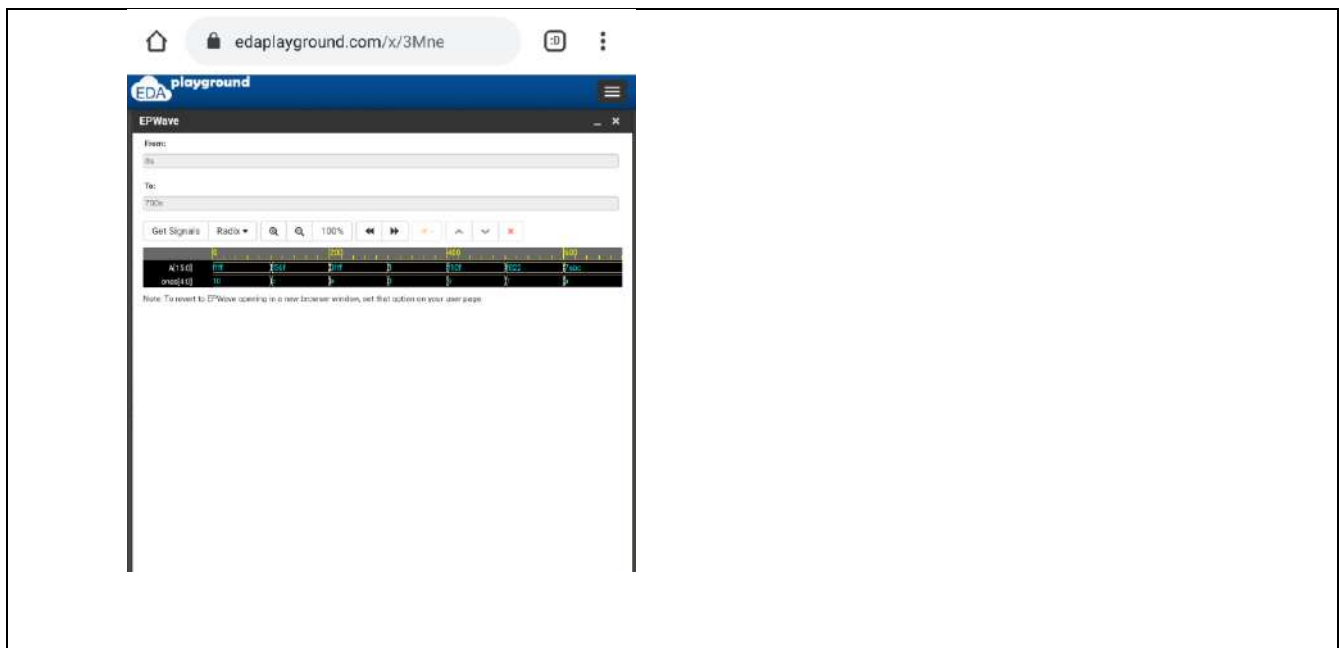
#### Parameterized Modules

Parameter override using dyanam  
 Parameter override during instantiating  
 Passing more than one parameter



## Task



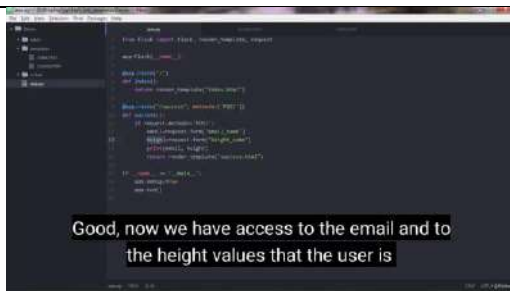


**Date:** 05/06/2020  
**Course:** Python  
**Topic:** Application 10

**Name:** Lavanya B  
**USN:** 4a17ec043  
**Semester** 6th A  
**& Section:**

#### AFTERNOON SESSION DETAILS

**Image of session**



Lectures More

257 Video - 08:31 mins - Resources (1)

Section 32 - Application 10: Build a Data Collector Web App with PostGre...

258 Data Collector Web App - How The Output Wi... Video - 02:59 mins - Resources (1)

259 PostgreSQL Database Web App with Flask: S... Video - 06:06 mins

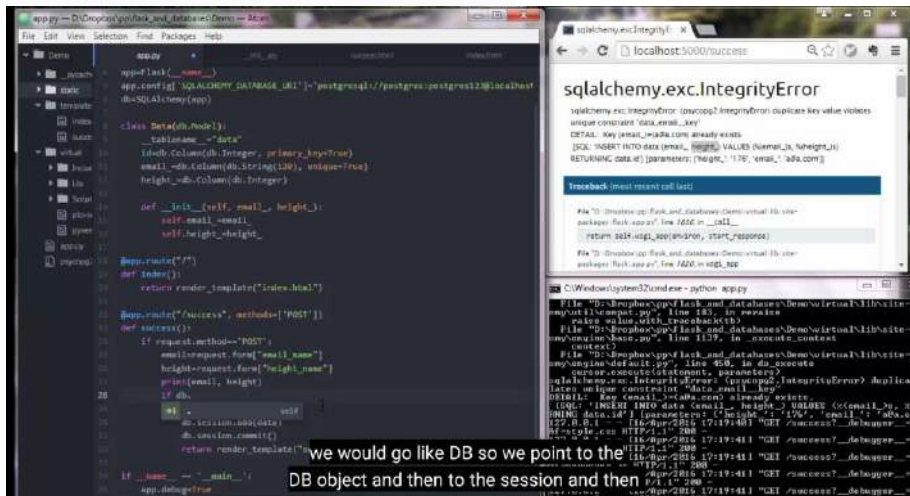
260 Frontend: HTML Part Video - 14:52 mins

261 Frontend: CSS Part Video - 10:11 mins

262 Backend: Getting User Input Video - 17:31 mins

263 Backend: The PostgreSQL Database Model Video - 18:17 mins

264 Backend: Storing User Data to the Database Video - 19:14 mins



## Report

# Application 10: Build a data collector web application with post GRESQL and flask

## Flask start up and configuration

Most widely used Python libraries, the Flask package is installable from the Python Package Index (PPI). First create a directory to work in (something like flask\_ to do is a fine directory name) then install the flask package. You'll also want to install flask- so your Flask application has a simple way to talk to a SQL data base. A good way to get moving is to turn the codebase into an installable Python distribution.

At the project's root, create setup.py and a directory called to do to hold the source code.

The setup.py should look something like this:

```
requires = [ 'flask', 'flask-sqlalchemy', 'psycopg2', ]
```

```
setup( name='flask_todo', version='0.0', description='A )
```

```
To-Do List built with Flask',
author='<Your actual name here>', author_email='<Your actual e-mail address here>',
keywords='web flask',
packages=find_packages(),
include_package_data=True,
install_requires=requires
```

This way, whenever you want to install or deploy your project, you'll have all the necessary packages in the requires list. You'll also have everything you need to set up and install the package in site packages.

For more information on how to write an installable Python distribution, check out the docs on setup.py.

Within the to do directory containing your source code, create an app.py file and a blank \_\_init\_\_.py file.

The \_\_init\_\_.py file allows you to import from to do as if it were an installed package. The app.py file will be the application's root. This is where all the Flask application goodness will go, and you'll create an environment variable that points to that file. If you're using pipenv (like I am), you can locate your virtual environment with pipenv --venv and set up that environment variable in your environment's activate script.



