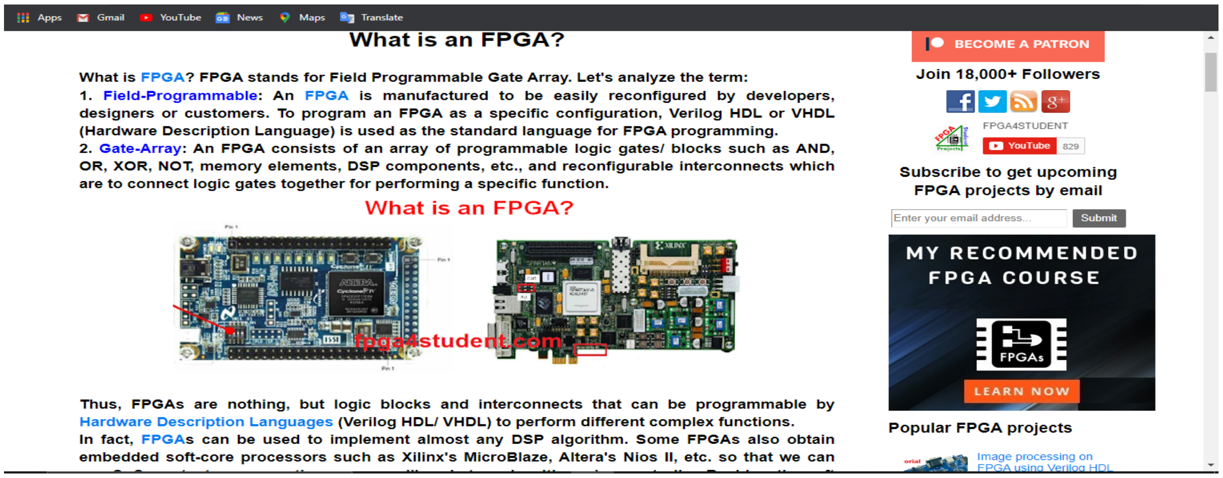
DAILY ASSESSMENT FORMAT

Date:	**05-06-2020**	**Name:**	**Prajwal Kamagethi Chakravarti P L**
Course:	**Digital design using HDL**	**USN:**	**4AL17EC073**
Topic:	• **Verilog Tutorials and practice programs** • **Building/ Demo projects using FPGA**	**Semester & Section:**	**6th & B**
GitHub Repository:	**https://www.github.com/alvas-education-foundation/Prajwal-Kamagethi.git**		

FORENOON SESSION DETAILS

Image of session



What is an FPGA?

What is FPGA? FPGA stands for Field Programmable Gate Array. Let's analyze the term:
1. Field-Programmable: An FPGA is manufactured to be easily reconfigured by developers, designers or customers. To program an FPGA as a specific configuration, Verilog HDL or VHDL (Hardware Description Language) is used as the standard language for FPGA programming.
2. Gate-Array: An FPGA consists of an array of programmable logic gates/ blocks such as AND, OR, XOR, NOT, memory elements, DSP components, etc., and reconfigurable interconnects which are to connect logic gates together for performing a specific function.

What is an FPGA?

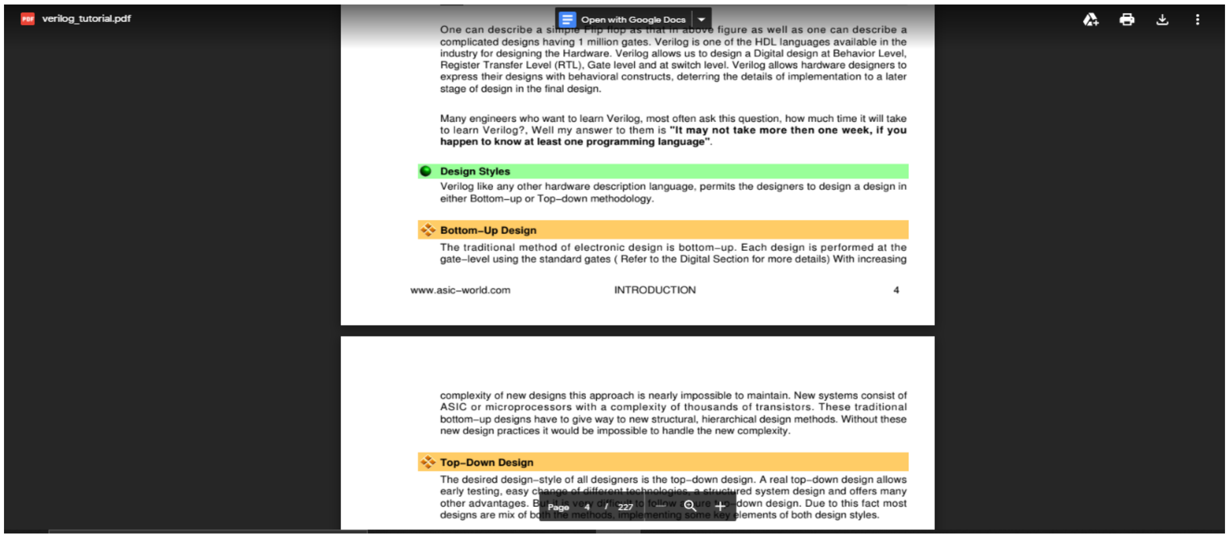
Thus, FPGAs are nothing, but logic blocks and interconnects that can be programmable by Hardware Description Languages (Verilog HDL/ VHDL) to perform different complex functions.
In fact, FPGAs can be used to implement almost any DSP algorithm. Some FPGAs also obtain embedded soft-core processors such as Xilinx's MicroBlaze, Altera's Nios II, etc. so that we can

Join 18,000+ Followers

Subscribe to get upcoming FPGA projects by email

Popular FPGA projects

Image processing on FPGA using Verilog HDL



One can describe a simple Flip flop as that in above figure as well as one can describe a complicated designs having 1 million gates. Verilog is one of the HDL languages available in the industry for designing the Hardware. Verilog 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, deterring the details of implementation to a later stage of design in the final design.

Many engineers who want to learn Verilog, most often ask this question, how much time it will take to learn Verilog?, Well my answer to them is **"It may not take more then one week, if you happen to know at least one programming language"**.

Design Styles

Verilog like any other hardware description language, permits the designers to design 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 increasing

www.asic–world.com INTRODUCTION 4

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 design practices it would be impossible to handle the new complexity.

Top–Down Design

The desired design–style of all designers is the top–down design. 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 mix of both the methods, implementing some key elements of both design styles.

Report –

- **FPGA stands for Field Programmable Gate Array. It is an integrated circuit which can be "field" programmed to work as per the intended design.**

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

- **Verilog simulator was first used beginning in 1985 and was extended substantially through 1987.**

- **Various stages of ASIC/FPGA:**
  Specification
  High Level Design
  Micro Design/Low level design:
  RTL coding
  Simulation
  Synthesis
  Place and route

- **An FPGA designer likes working on this due to these reasons:**
  Very fast on-chip (FPGA) demonstration
  Simple and fast design process on FPGA
  FPGA's programmability
  FPGA's high performance
  FPGA's flexibility

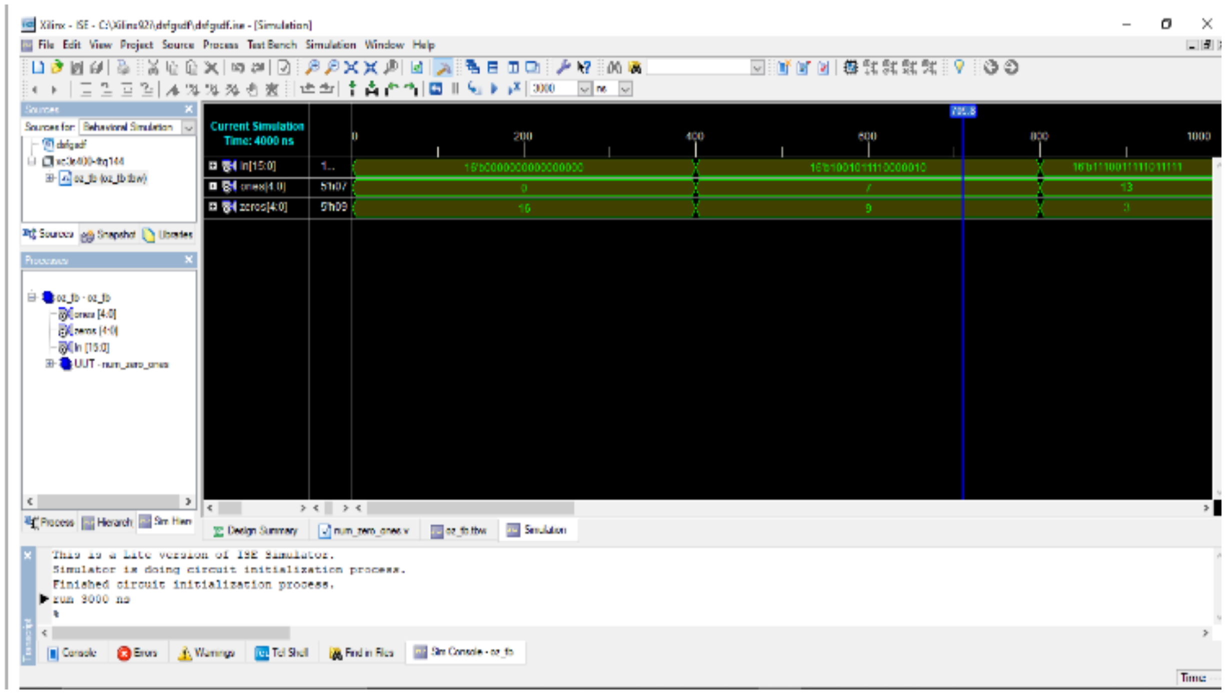
- **We also learnt the different ways and techniques of designing FPGA.**

Implement a verilog module to count number of 0's in a 16 bit number in compiler.

Verilog code:

```
module num_zero_ones
(input [15:0] In,
output reg [4:0] ones,
output reg [4:0] zeros);
integer i, o, z;
always
@(in)
begin
o = 0;
z = 0;
for(i=0;i<16;i=i+1)
```

```
if(In[i] == 1'b1)
o = o + 1;
z = 16-o;
ones = o;
zeros = z;
end
endmodule
```



```
This is a Lite version of ISE Simulator.
Simulator is doing circuit initialization process.
Finished circuit initialization process.
run 3000 ns
%
```

VHDL code for Clock divider on FPGA:

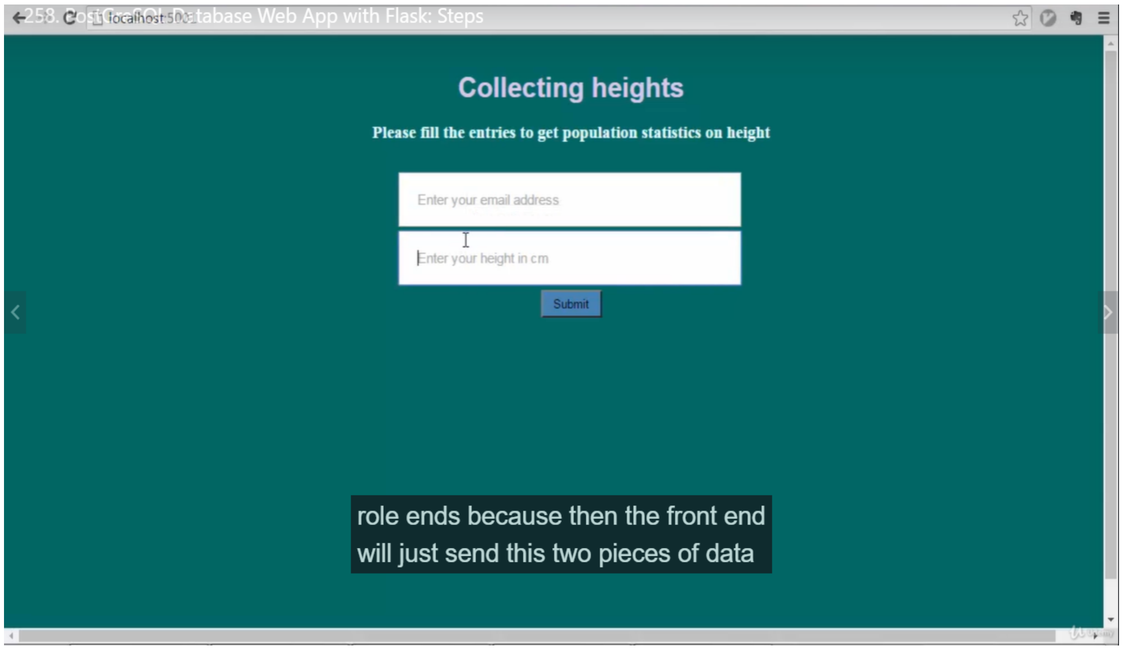
```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
USE IEEE.STD_LOGIC_UNSIGNED.ALL;
entity clock_div is
port (
   clk_in: in std_logic; -- clock input on FPGA
   clk_out: out std_logic -- clock output
  );
end clock_div;

architecture Behavioral of clock_div is
signal divisor: std_logic_vector(27 downto 0):=(others =>'0');
```

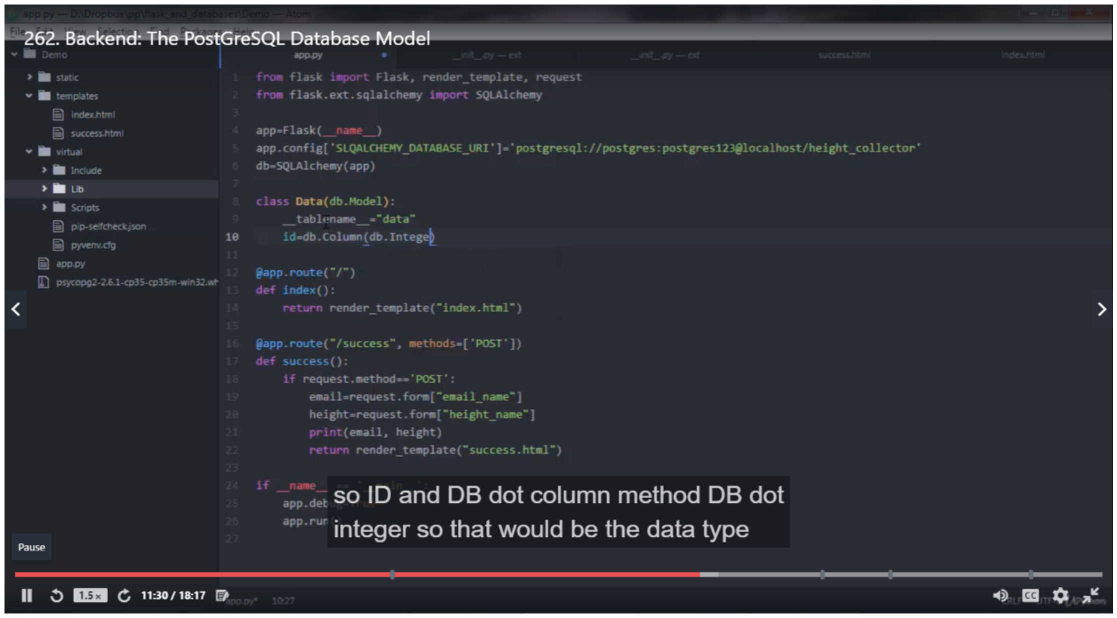
```
begin
 process(clk_in)
 begin
 if(rising_edge(clk_in)) then
 divisor <= divisor + x"0000001";
 -- If(divisor>=x"2FAF07F") then -- for running on FPGA -- comment when running simulation
 -- Modify the divisor (x"2FAF07F"=49999999) above to get the clock frequency you want:
 -- Frequency of clk_out = Frequency of (clk_in) divided by (divisor + 1)
 -- If the frequency of clk_in is 50MHz and the divisor is 49999999=x"2FAF07F",
 -- the frequency of clk_out is 1Hz
 if(divisor>=x"0000001") then -- for running simulation -- comment when running on FPGA
 -- divisor = 1 => divide clock by half for simulation purposes
 divisor <= x"0000000";
 end if;
 end if;
 end process;
 --clk_out <= '0' when divisor < x"17D7840" else '1'; -- half of the clock divisor x"17D7840" = (divisor
+ 1)/2
 clk_out <= '0' when divisor < x"0000001" else '1';-- divide clock by half
end Behavioral;
```

Date:	05-06-2020	Name:	Prajwal Kamagethi Chakravarti P L
Course:	Udemy-python	USN:	4AL17EC073
Topic:	Build a data collector web app with PostGreSQL and Flask	Semester &Section:	6th & B
GitHub repository	https://www.github.com/alvas-education-foundation/Prajwal-Kamagethi.git		

AFTERNOON SESSION DETAILS

Image of session


258. PostGreSQL Database Web App with Flask: Steps
localhost:5000
Collecting heights
Please fill the entries to get population statistics on height
Enter your email address
Enter your height in cm
Submit
role ends because then the front end will just send this two pieces of data


262. Backend: The PostGreSQL Database Model

```
from flask import Flask, render_template, request
from flask.ext.sqlalchemy import SQLAlchemy

app=Flask(__name__)
app.config['SLQALCHEMY_DATABASE_URI']='postgresql://postgres:postgres123@localhost/height_collector'
db=SQLAlchemy(app)

class Data(db.Model):
    __tablename__="data"
    id=db.Column(db.Intege)

@app.route("/")
def index():
    return render_template("index.html")

@app.route("/success", methods=['POST'])
def success():
    if request.method=='POST':
        email=request.form["email_name"]
        height=request.form["height_name"]
        print(email, height)
        return render_template("success.html")

if __name__
    app.deb
    app.run
```

Report –

- **In this section we learnt to build a data collector web application, which collects height data from the user and sends the survey result via e-mail.**
- **To deploy the application to a live web server, Heroku platform is used and to save the data from the users, database is created in Heroku using credentials**
- **We can install Heroku CLI using this shell script.**
- **Then we have to Create python virtual environment for the project.**
- **To activate this environment, use this common inside book server directory.**
- **Then we install flask and write the code and run it.**
- **Then writing the database we need for our application is necessary.**
- **We need to define configurations for deploying environments**
- **Then we need to use flask package for database manipulations.**
- **After finishing the code completely Commit changes using git and push to Heroku.**