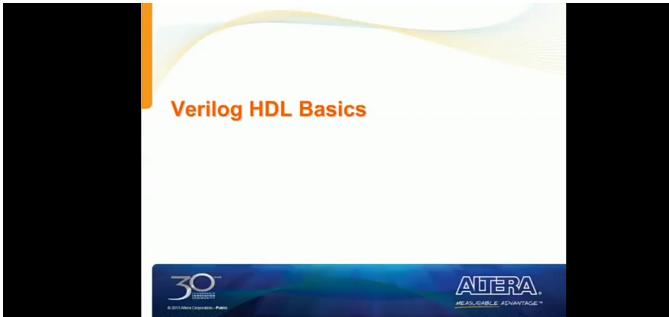
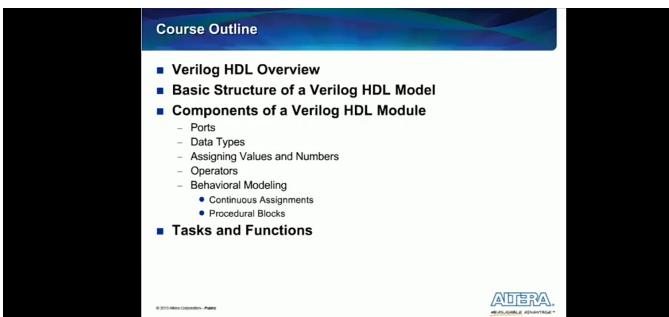


JUNE 2 REPORT

Date:	05/06/20	Name:	ANKITHA C C
Course:	DIGITAL DESIGN USING HDL	USN:	4AL16EC004
Topic:	FPGA & VHDL	Semester & Section:	8TH & A
Github Repository:	ankitha-course		

FORENOON SESSION DETAILS		
<p>Image of session</p>  		

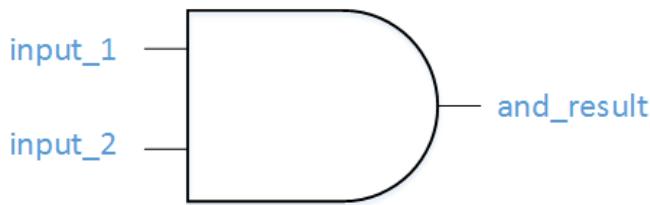
Report - Report can be typed or hand written for up to two pages.

1. Introduction to Verilog

Verilog is a type of Hardware Description Language (HDL). Verilog is one of the two languages used by education and business to design FPGAs and ASICs. If you are unfamiliar with how FPGAs and ASICs work you should read this page for an introduction to FPGAs and ASICs. Verilog and VHDL are the two most popular HDLs used. Compared to traditional software languages such as Java or C, Verilog works very differently. Let's get started by looking at a simple example.

First we will create a Verilog file that describes an And Gate. As a refresher, a simple And Gate has two inputs and one

output. The output is equal to 1 only when both of the inputs are equal to 1. Below is a picture of the And Gate that we will be describing with Verilog.



An And Gate

Let's get to it! One fundamental unit of Verilog is called a wire. For now let's assume that a wire can only be a 0 or a 1. Here is some basic wire logic:

1. `wire and_temp;`
2. `assign and_temp = input_1 & input_2`

We are creating a wire called `and_temp` on the first line of code. On the second line of the code, we are taking the wire that we created and we are assigning the wire. To assign it, we are using the Boolean AND function which in Verilog is the Ampersand (`&`). If you were to describe the code shown above, you might say, "The signal `and_temp` gets `input_1` ANDed with `input_2`."

`Input_1` and `Input_2` are inputs to this piece of Verilog Code. Let's show the complete list of inputs and outputs. This is done in the module definition. Module is a reserved keyword in Verilog which shows the creation of a block of code with defined inputs and outputs.

```
module example_and_gate
```

```
(  
    input_1,  
    input_2,  
    and_result);
```

```
input input_1;  
input input_2;  
output and_result;
```

This is your basic module. It defines our module called `example_and_gate` and 3 signals, 2 inputs and 1 output. Let's put everything together to finish the file. The only thing we are missing is the assignment of the output `and_result`. One other note, `//` in Verilog is used for a comment

```
//////////
```

```
// File Downloaded from http://www.hardland.com
```

```
//////////  

module example_and_gate  

(  

    input_1,  

    input_2,  

    and_result);  

    input input_1;  

    input input_2;  

    output and_result;  

    wire and_temp;  

    assign and_temp = input_1 & input_2;  

    assign and_result = and_temp;  

endmodule // example_and_gate
```

2. Building/ Demo projects using FPGA

Field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing – hence the term "field-programmable". The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC). Circuit diagrams were previously used to specify the configuration, but this is increasingly rare due to the advent of electronic design automation tools.



- * start system software (SW) development simultaneously with hardware (HW)

- * enable system performance simulations at very early phase of the development

* allow various system partitioning (SW and HW) trials and iterations before final freezing of the system architecture

Easy Design Exploration – Finding the best solutions often requires evaluating multiple solutions. Lattice Diamond allows for easy design exploration.

Easy to Use Powerful Tools – Adapting to a new tool is often difficult. Lattice Diamond employs familiar easy to use tools and methodologies that make common tasks easier.

Optimized for Lattice Devices – Lattice Diamond offers an optimized and tailored design and verification environment for Lattice FPGAs featuring extensive constraints, advanced optimization, accurate analysis, extensive verification, and fast iterations.

Features

Complete GUI based FPGA design and verification environment.

Design exploration with multiple implementations and optimization strategies within a single project.

Graphical environment timing and power analysis.

TASK 5

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

1's Complement of 16 bit number Using 8085 microprocessor

1's COMPLEMENT OF A 16-BIT NUMBER
The 16bit number is stored in C050, C051
The answer is stored in C052, C053

```
LXI H, C050
MOV A, M
CMA
STA C052
INX H
MOV A, M
CMA
STA C053
HLT
```

1's Complement of Binary Number: 0 -> 1, 1 -> 0

1's Complement of 16 bit number Using 8085 microprocessor

1's COMPLEMENT OF A 16-BIT NUMBER
The 16bit number is stored in C050, C051
The answer is stored in C052, C053

```
LXI H, C050 // [H-L] = C050 [H] = C0 [L] = 5E [M] = 15h
MOV A, M // [A] = 15h
CMA // Complement the accumulator [A] = EA
STA C052 // Store accumulator into C052
INX H // [H-L] = C051 [H] = C0 [L] = 51 [M] = 2Ah
MOV A, M // [A] = 2Ah
CMA // Complement the accumulator [A] = D5
STA C053 // Store accumulator into C053
HLT
```

[C050] = 15h [C052] = EA
[C051] = 2Ah
215 - 2 = 13 (D), A, 15-10 = 5

05/06/20

Name:

Ankitha c c

Date:

Course: Python

USN:

4al16ec004

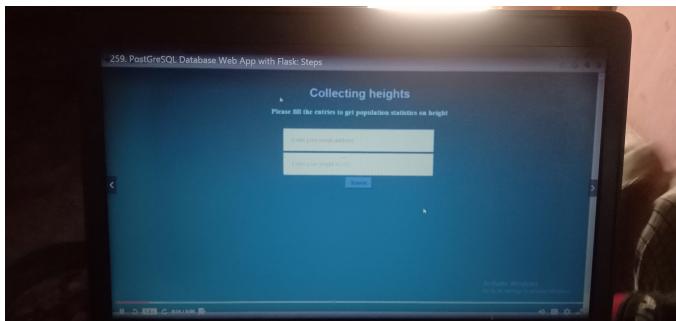
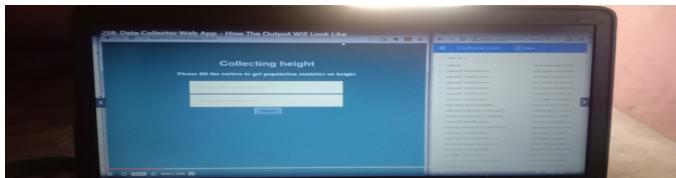
Topic:

Semester &
Section:

8th & a

AFTERNOON SESSION DETAILS

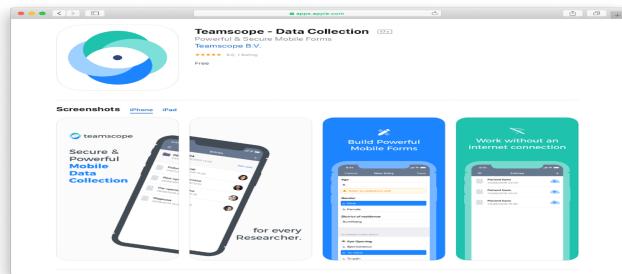
Image of session.



Report - Report can be typed or hand written for up to two pages.

1. Data collector web app

Data-Collector. Data Collector app with python flask framework. This web app collects data and sends an email to the user automatically.



Teamscope is a secure and easy-to-use data collection platform, specially designed for sensitive data and clinical research.

In a field where most tools are web-only and useless without an internet connection, Teamscope offers a unique approach at on-the-go and secure data gathering. With its offline-first Android and iOS app, Teamscope allows researchers to create powerful mobile forms, collect data both qualitative and quantitative data offline and visualize it with a few clicks.

Teamscope sets data security as its highest priority. Data is stored encrypted on mobile devices and users, apart from requiring a username and password to login, must create a four digit passcode to unlock the app. All sessions on its mobile app time out after 30 seconds of inactivity or once the app has been closed, to access the app a user must reenter their Teamscope passcode.

When conducting a longitudinal study, researchers can make use of Teamscope's case management feature. This functionality allows them to create cases for individual subjects, share them with other users in their project, and upload data for their cases in multiple moments

2. Pstgresql database web app with flask

Part One: Set up a local development environment and then deploy both a staging and a production environment on Heroku.

Part Two: Set up a PostgreSQL database along with SQLAlchemy and Alembic to handle migrations. (current)

Part Three: Add in the back-end logic to scrape and then process the word counts from a webpage using the requests, BeautifulSoup, and Natural Language Toolkit (NLTK) libraries.

Part Four: Implement a Redis task queue to handle the text processing.

Part Five: Set up Angular on the front-end to continuously poll the back-end to see if the request is done processing.

Part Six: Push to the staging server on Heroku - setting up Redis and detailing how to run two processes (web and worker) on a single Dyno.

Part Seven: Update the front-end to make it more user-friendly.

Part Eight: Create a custom Angular Directive to display a frequency distribution chart using JavaScript and D3.

3. Front end HTML part

Front-end web development is the practice of converting data to a graphical interface, through the use of HTML, CSS, and JavaScript, so that users can view and interact with that data.

Tools used for front-end development

There are several tools and platforms (wordpress, magento etc..) available that can be used to develop the front end of a website, and understanding which tools are best fit for specific tasks marks the difference between developing a hacked site and a well designed, scalable site.

Hyper Text Markup Language

Hyper Text Markup Language (HTML) is the backbone of any website development process, without which a web page does not exist. Hypertext means that text has links, termed hyperlinks, embedded in it. When a user clicks on a word or a phrase that has a hyperlink, it will bring another web-page. A markup language indicates text can be turned into images, tables, links, and other representations. It is the HTML code that provides an overall framework of how the site will look. HTML was developed by Tim Berners-Lee. The latest version of HTML is called HTML5 and was published on October 28, 2014 by the W3 recommendation. This version contains new and efficient ways of handling elements such as video and audio files.

Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS) controls the presentation aspect of the site and allows your site to have its own unique look. It does this by maintaining style sheets which sit on top of other style rules and are triggered based on other inputs, such as device screen size and resolution.

JavaScript

JavaScript is an event-based imperative programming language (as opposed to HTML's declarative language model) that is used to transform a static HTML page into a dynamic interface. JavaScript code can use the [Document Object Model] (DOM), provided by the HTML standard, to manipulate a web page in response to events, like user input.

Using a technique called AJAX, JavaScript code can also actively retrieve content from the web (independent of the original HTML page retrieval), and also react to server-side events as well, adding a truly dynamic nature to the web page experience.

4. Deployment of web application to live server

Something that is overlooked by a lot of web designers and developers is what is actually involved in the deployment of a website; the process when you've finished developing the site, tested to make sure it works, and are ready to push it to a live web server.

In a lot of cases, you will be dealing with clients who are getting their website for the first time, and there is nothing for you to really consider apart from the hosting solution to set them up on. As time goes on, you will start getting larger clients that may have existing websites already, or who have more complicated needs. You may find yourself in a scenario where the outcome could be a very unhappy client with data loss and a whole company's worth of missing emails and site assets. Here are some basic steps that will ensure that you have covered all the bases for a smooth website deployment.

Step 1: Preparation

There are a few things to consider when you are finalizing a website, and they all depend on what type of deployment you will be completing.

The three general scenarios of a website deployment is:

1. The client has nothing (i.e. this is their first website)

2. The client already has hosting and you will be deploying the site on their server

3. The client already has hosting but you will be moving to a new server

The first scenario is the most desired because you are starting with a blank slate. Scenarios 2 and 3 are a bit trickier and involve a more thoughtful deployment process. Once you have worked out what your deployment scenario is, you will be able to better prepare yourself for everything you need to do in order to carry out a smooth transition from the old website to the new one. If you are dealing with scenario 1, then all you need to do is register their domain name and purchase (or provide) web hosting. Simple and fast deployment. Scenarios 2 and 3 require some information gathering. You need domain management credentials for the existing web host so that you can manage the DNS records (more on this in a bit). You will find that, in many cases, the client has no idea what these are or where to get them, so you will need to do as much as you can before you approach your client.

So let's gather information on our own. We can use a tool like whois.domaintools.com to find out some information about the existing domain name.

The screenshot shows the DomainTools.com homepage with the "WHOIS Search at DomainTools.com" section highlighted. It features a search bar with placeholder text "Enter a Domain or IP Address..." and a "lookup" button. Below the search bar is a link to "Whois Examples: [domaintools.com](#), [whois.ac](#), [127.0.0.1](#)".

Type in the domain name and on the results page you will see the whois information. For those not familiar with the term, a whois (pronounced as "who is") is a query of information regarding an Internet resource, such as a domain name. For illustration purposes, here's the whois information for Google.

The screenshot shows the "Whois Record" page for Google.com. At the top, there are tabs for "Whois Record", "Site Profile", "Registration", "Server Stats", and "My Whois". The "Whois Record" tab is selected. The main content area displays the following details:

- Related Domains For Sale or At Auction:** EasyFind.net (Bid), GoogleHouston.com (\$1,000), OurGoogle.com (\$644), RetireFind.com (Bid), LuckyFind.net (Bid), GoogleEdge.com (\$318).
- Reverse Whois:** "Dns Admin" owns about 288,947 other domains.
- Email Search:** contact-admin@google.com is associated with about 6 domains, dns-admin@google.com is associated with about 8,100 domains.
- Registrar History:** IP History shows 25 unique name servers over 6 years.
- Whois History:** 2,683 records have been archived since 2001-05-03.
- Reverse IP:** 587 other sites host on this server.
- Domain Tools for Windows®:** A note stating "Now you can access domain ownership records anytime, anywhere... right from your own desktop! [Download Now!](#)"
- Registrant:** Google Inc., Please contact contact-admin@google.com, 1600 Amphitheatre Parkway, Mountain View CA 94043, dns-admin@google.com, +1.6502530000 Fax: +1.6506188571.
- Domain Name:** google.com, Registrar Name: Markmonitor.com, Registrar Tools: whois.markmonitor.com, Registrar Homepage: http://www.markmonitor.com.
- Administrative Contact:** DNS Admin, Google Inc.

Take note of the various contact email addresses, especially the administrative and technical contact. If you know who they are, then you are all set because you will know who to talk to. If you don't, just write down their contact details and ask your point-of-contact for the project about them.

Next, click on the Registration tab. You will see ICANN Registrar information (the first line), which you should take a note of. Also, note down the Name Servers listed.

Whois Record **Site Profile** **Registration** **Server Stats** **My Whois**

Registration

Related Domains For Sale or At Auction

EasyFindIt.net (Bid)	GoogleHouston.com (\$1,000)	OurGoogle.com (\$644)
RetireFind.com (Bid)	LuckyFind.net (Bid)	GoogleEdge.com (\$51)

ICANN Registrar: MARKMONITOR INC.

Created: 1997-09-15
Expires: 2011-09-14
Updated: 2010-09-15

Registrar Status: **clientDeleteProhibited**
clientTransferProhibited
clientUpdateProhibited
serverDeleteProhibited
serverTransferProhibited
serverUpdateProhibited

Name Server: NS1.GOOGLE.COM (has 7,311 domains)
NS2.GOOGLE.COM (has 7,311 domains)
NS3.GOOGLE.COM (has 7,311 domains)
NS4.GOOGLE.COM (has 7,311 domains)

Whois Server: whois.markmonitor.com

General TLDs: Google.com (registered and active website)
Google.net (registered and active website)
Google.org (registered and active website)
Google.biz (registered and active website)

The ICANN Registrar is the company that registered the domain name. GoDaddy, Network Solutions, and Namecheap.com are examples of ICANN registrars. If you have contact with the person listed as the domain's administration or technical contact, either request the ability to manage the domain name yourself or ask them to modify the DNS records for you when your site is ready to be deployed. If you don't know the contact for the domain, then you will have to get your client to email or phone them for you. At the very least, if you mention the ICANN Registrar's name (e.g. "Hey, you registered your domain name on GoDaddy, does that ring a bell?"), then it might jog their memory and help them recall the information you need.

Step 2: Set Up DNS Records

If you are going to be setting up the website on a new host and you have access to the DNS management administration, then that's great. Create yourself an A record (the address record that maps a domain name to the IP address of the server) or subdomain record for a live development site such as dev.domainname.com. Point this subdomain to the IP address of the new server.

If you don't have DNS access but wish to have full control, I recommend using ZoneEdit.com, which is a free and easy, web-based domain manager. Be warned! Make sure you know what you're doing with this tool; read their DNS basics and FAQ.

The screenshot shows a web-based interface for managing domain zones. On the left, a sidebar titled "Choose zone" lists several domains: hellozone.org, httpcolon.net, line5.com, sslcerts.com, and vagabond.org. The main panel is titled "View Configuration" and contains four sections: "IP Addresses", "Mail Servers", "Aliases", and "WebForwards".

- IP Addresses:** Shows one entry: Domain Name [httpcolon.net](#) and IP Address 209.73.15.231.
- Mail Servers:** Shows one entry: Domain httpcolon.net, Server MailForward™, and Rank 0.
- Aliases:** Shows one entry: Alias [www.httpcolon.net](#) and Actual Domain Name [httpcolon.net](#).
- WebForwards:** Shows two entries: New Domain [shop.httpcolon.net](#) and Destination [http://mom.com](#); and New Domain [stomach.httpcolon.net](#) and Destination [http://amazon.com](#).
- MailForwards:** Shows two entries: New Address [blaster@httpcolon.net](#) and Destination [erik@q32.com](#); and New Address [erik@httpcolon.net](#) and Destination [erik@q32.com](#).

If you don't want to get this far into the technical side, you need access to their account on their domain name registration service, which will usually have GUIs to help you set up DNS records.

Step 3: Set Up a Live Testing Site

It's now time to see if the site works on the live server environment.

A practice I recommend doing is setting up a subdomain URL prior to officially deploying the site. Something like dev.domainname.com which will eventually be on domainname.com.

Don't create a subdomain on the host as this will set up a new directory and make local DNS changes. Set it so that dev.domainname.com acts as a totally separate website. What you want to do is make dev.domainname.com a domain alias (also known as a CNAME record).

So, for example, if you've set up an A record (the record that maps the IP address of the web server to the domain name) like so:

example.com. A 192.0.2.1

You would set an alias for dev.example.com as such:

dev.example.com. CNAME example.com.

By doing this, you can set up the website in the same physical location that it will live. You want to be as accurate as possible here so that you can do your final tests as if the site was truly deployed (which, technically, it is).

You can set all folder permissions and other settings, and then run tests and benchmarks to see how the site performs on the server.

If you're hosting on the same server as the old website, the best you can do is upload to a directory named dev and set up a subdomain DNS record for it while you test. This allows the existing site to function normally, while still allowing you to test the web server environment. You will have to move this when it is time to deploy.

Step 4: Set Up Email Accounts

Developers deploying a website often overlook email, but it will be a priority to the client. Does your client have mail hosted on their old server? Are you moving their email?

If their email is currently in the same hosting account as the old website, then you will probably be moving mail to the new server. If so, collect all email account addresses and set up the exact same accounts on the new server. In most cases, you then won't need to change anything, it will just transition to the new mail server at the same time the website does.

If the client has an internal mail server or third-party mail hosting, then you will need to make sure that the MX records (the DNS records that deal with mail) are all correct.

If your client has no idea, then a quick test is to ping the mail server, and if it has a different IP address to the website, then it's most likely hosted on a different server and you need to double-check the MX records and make sure whoever is managing the DNS is notified of what is happening.

MxToolbox will give you all the information you need about the domain; it will list information about a domain name's MX records.

The screenshot shows the MxToolbox interface with the 'SuperTool Beta' tab selected. A search bar contains 'mx:sixrevisions.com'. Below it, a table displays MX record results:

Pref	Hostname	IP Address	TTL	SMTP Test	Blacklist Check
1	ASPMX.L.GOOGLE.com	74.125.65.27	24 hrs	SMTP Test	Blacklist Check
5	ALT1.ASPMX.L.GOOGLE.com	209.85.227.27	24 hrs	SMTP Test	Blacklist Check
5	ALT2.ASPMX.L.GOOGLE.com	74.125.77.27	24 hrs	SMTP Test	Blacklist Check
10	ASPMX2.GOOGLEMAIL.com	74.125.43.27	24 hrs	SMTP Test	Blacklist Check
10	ASPMX3.GOOGLEMAIL.com	72.14.213.27	24 hrs	SMTP Test	Blacklist Check

The last thing you want to happen is for the client to lose email.

Step 5: Backup and Go Live

Even if you are hosting on a new server, take a full backup including any databases of the old website, as you never know when you might need something.

OK, all set to go live. If you have full control over DNS records then just change the A record for the domain name so that the IP address is set to the new web server and in about 20 minutes the new website will be live.

If anything is not right, just change it back to the old website and do some testing.

If you are changing Name Servers to point to the new host, then this can take anywhere up to 72 hours, so make sure you have the time to monitor and fix any errors as they happen on the new website. Because this is a change in name servers,

you can't just change it back quickly, so be prepared and give yourself enough time.

If you are hosting on the same server and removing the old website to make way for the new one, then do it at a time where you can monitor and fix anything live as it happens.

Give yourself enough time and try to go live in the business hours of the companies that you will need to contact if anything goes wrong.

All done. If you follow these steps, you should have a 100% smooth deployment of your new website and a happy client to spread the word of your business.

Website Deployment Checklist

Have access to DNS record management or know the people to contact

Set up the DNS records and make sure that all the settings are correct

Set up and test the website on the production server (where it will live)

Set up email

Back up the old site (if applicable) and deploy the new one

