DAILY ASSESSMENT FORMAT

Date:	**01-06-2020**	**Name:**	**Yashaswini R**
Course:	**Digital Design using HDL**	**USN:**	**4AL17EC098**
Topic:	**Industry Applications of FPGA, FPGA Business Fundamentals, FPGA vs ASIC Design Flow**	**Semester & Section:**	**6th & B**
Github Repository:	**Yashaswini**		

FORENOON SESSION DETAILS
Image of session

Report

An FPGA is a (mostly) digital, (re-)configurable ASIC. I say mostly because there are analog and mixed-signal aspects to modern FPGAs.

- Parallel processes– if you need to process several input channels of information (e.g. many simultaneous A/D channels) or control several channels at once.
- High data-to-clock-rate-ratio– if you've got lots of calculations that need to be executed over and over and over again, essentially continuously. The advantage is that you're not tying up a centralized processor. Each function can operate on its own.
- Large quantities of deterministic I/O– the amount of determinism that you can achieve with an FPGA will usually far surpass that of a typical sequential processor. If there are too many operations within your required loop rate on a sequential processor, you may not even have enough time to close the loop to update all of the I/O within the allotted time.
- Signal processing– includes algorithms such as digital filtering, demodulation, detection algorithms, frequency domain processing, image processing, or control algorithms.
- Complex calculations infrequently– If the majority of your algorithms only need to make a computation less than 1% of the time, you've generally still allocated those logic resources for a particular function (there are exceptions to this), so they're still sitting there on your FPGA, not doing anything useful for a significant amount of time.
- Sorting/searching– this really falls into the category of a sequential process. There are algorithms that attempt to reduce the number of computations involved, but in general, this is a sequential process that doesn't easily lend itself to efficient use of parallel logical resources.
- Floating point arithmetic– historically, the basic arithmetic elements within an FPGA have been fixed-point binary elements at their core. In some cases, floating point math can be achieved .
- Very low power– Some FPGAs have low power modes (hibernate and/or suspend) to help reduce current consumption, and some may require external mode control ICs to get the most out of this.

Task for Day-1:

Write a verilog code to implement NAND gate in all different styles.

Gate Level modelling:

```
module NAND_2_gate_level(output Y,input A,B);
      wire Yd;
     and(Yd,A,B);
     not(Y,Yd);
endmodule
```

Data flow modelling:

```
module
NAND_2_data_flow(output Y,input A,B);
```

```
    assign Y=~(A&B);
endmodule
```

Behavioral Modeling:

```
module
NAND_2_behavioral(output reg Y,input A,B);
always@(A or B) begin
      if(A==1'b1 & B==1'b1) begin
          Y=1'b0;
       end
       else
           Y=1'b1;
end
endmodule
```

GUVI
tech deserves you

Yashaswini R

is here by awarded the certificate of achievement for
the successful completion of

Step into Robotic Process Automation

during GUVI's RPA SKILL-A-THON 2020

S.P.Balamurugan
Co-founder, CEO

Valid certificate ID 83870w59tr9ke7OI89

Verified certificate issue on June 1 2020

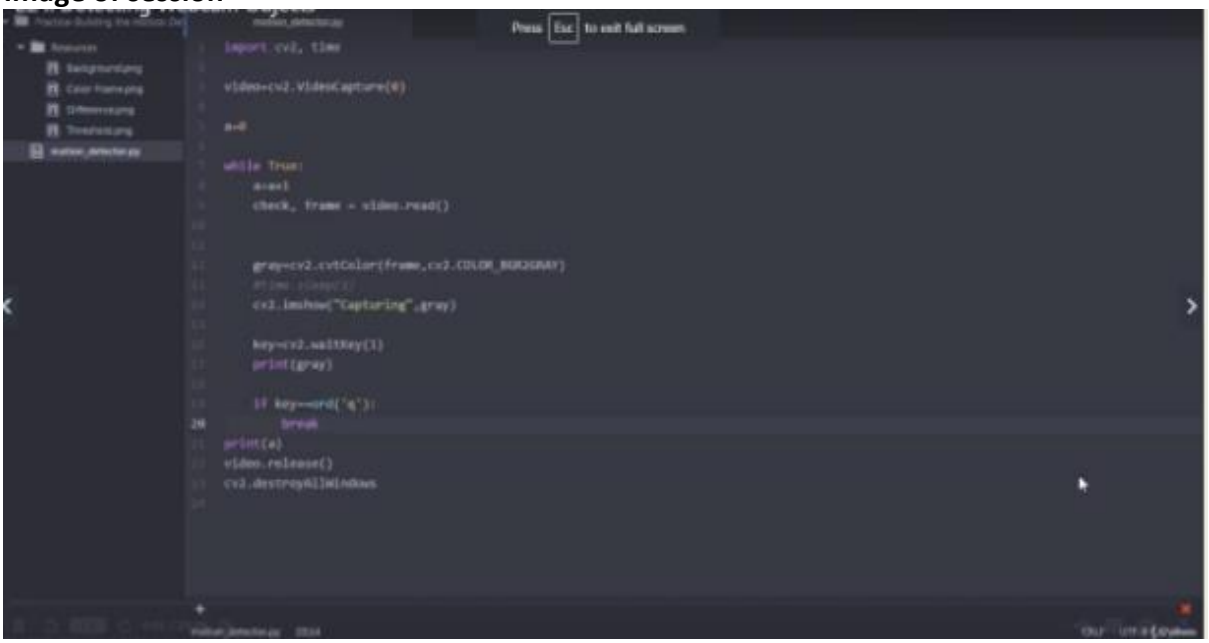
Verify certificate at www.guvi.in/certificate?id=83870w59tr9ke7OI89

In association with
UiPath Academic Alliance

Date:	01-06-2020	Name:	Yashaswini R
Course:	UDEMY PYTHON MEGA_COURSE	USN:	4AL17EC098
Topic:	Build a webcam motion detector	Semester & Section:	6th &B

AFTERNOON SESSION DETAILS

Image of session



Report –

Motion detection is the detection of the change in the position of an object with respect to its surroundings and vice-versa. Buckle up your seat belts to drive through this motion detector application along with me and your lovable Python.You may be able to perform the following tasks using this application, though the list is non-exhaustive:

- Find in front of screen time during working from home.
- Monitor your child's in front of screen time.
- Find trespassing in your backyard.
- Locate unwanted public/animal movements around yoroom/house/alley and what not.....Photo by Williapm Thomas on Unsplash Hardware Requirements: A computer with a webcam or any type of camera installed.

Software Requirements: Python 3 or above.Additional Requirements: 30 mins of your time, Enthusiasm about the topic I will guide you step by step into building the application. Firstly, you will capture the first frame via webcam. This frame will be treated as the baseline frame. Motion will be detected by calculating the phase difference between this baseline frame and the new frame with some object. The new frames will be called Delta frame.Then you will refine your delta frame using pixel intensity. The refined frame will be called the Threshold frame. Then you will apply some intricate image processing techniques like Shadow Removal, Dilation, Contouring, etc. on the Threshold frame to capture substantial objects. Detected Object You will be able to capture the time stamp when an object entered the frame and exited the frame. Thus, you will be able to find the screen-on time.I won't embed my codehere as I would like you to improve the blood circulation on your fingertips.To start with basic installations, please install python 3 or above, pandas, and opencv via pip. Once done, you are ready to begin:

- STEP 1: Import required libraries:
- STEP 2: Initialize variables, lists, data frames:You will get to know when each one of the above will be required in the below code.
- STEP 3: Capture the video frames using webcam:OpenCV has in-built functions to open the camera andcapture video frames. "0" denotes the camera at the hardware port number 0 in your computer. If you have multiple cameras or external cameras or a CCTV setup installed, you may provide the port number accordingly.
- STEP 4: Converting the captured frame to gray-scale and applying Gaussian Blur to remove noise:We convert the color frame to gray frame as an extra layer of color is not required. GaussianBlur is used for image smoothing and it will, in turn, enhance the detection accuracy. In the GaussianBlur function, for the 2nd parameter, we define the width and height of the Gaussian Kernel and for the 3rd parameter, we provide standard deviation value. These are set of higher order differential calculus theorems, so you may use standard values of the kernel size as (21,21) and standard-deviation as 0.