

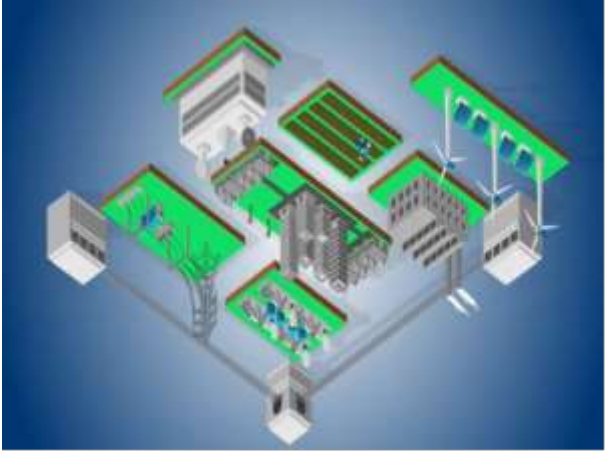
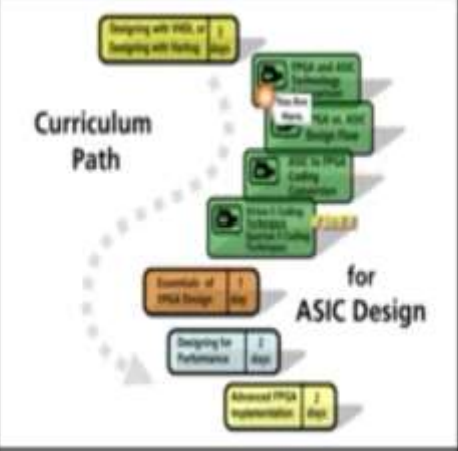


DAILY ASSESSMENT FORMAT

Date:	01-06-2020	Name:	POOJA K S
Course:	DIGITAL DESIGN USING HDL	USN:	4AL17EC070
Topic:	INDUSTRY APPLICATION,BUSINESS FUNDAMENTALS OF HDL,FPGA VS ASIC.	Semester & Section:	6 TH SEM 'B' SECTION
Github Repository :	pooja-shivanna		

	FOREN D	
<p>of session</p> 		
		



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

FPGA Basics – A Look Under the Hood An introductory look inside Field Programmable Gate Arrays. We'll go over: Strengths & Weaknesses of FPGAs How FPGAs work What's inside an FPGA So you keep hearing about FPGAs being utilized in more and more applications, but aren't sure whether it makes sense to switch to a new technology. Or maybe you're



just getting into the embedded world and want to figure out if an FPGA-based system makes sense for you or not. This paper provides an overview of some of the key elements of FPGAs for engineers interested in utilizing FPGA-based technologies. It's worth noting that this is a complex topic, and as such, some topics are not covered, some are just introductory, and others will evolve over time. This paper should still give you a lot of helpful information if you're new to the world of FPGAs. What are the most important things you should know right away? Get out of the software mindset – You're not writing software. Let me say that again because this is the single most important point if you're thinking about working with FPGAs. You-are-NOT-writing- software. You're designing a digital circuit. You're using code to tell the chip how to configure itself. Plan for lots of bugs – yes, plan for them. They are going to happen. Way more than you expected. If you're a newbie developer, you need to pull in someone that has experience with FPGA development to help with this estimate. Application-specific realities – you ought to concern yourself with realities revolving around cyber security and safety, as FPGAs are a different animal than what you're likely used to. What is an FPGA? An FPGA is a (mostly) digital, (re-)configurable ASIC. I say mostly because there are analog and mixed-signal aspects to modern FPGAs. For example, some have A/D converters and PLLs. here for some more info. 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 (see Xilinx FP Operator and Altera FP White Paper), but it will chew up a lot of logical resources. Be mindful of single-precision vs double-precision, as well as deviations from standards. However, this FPGA weakness appears to be starting to fade, as hardened floating-point DSP blocks are starting to be embedded within some FPGAs (see Altera Arria 10 Hard Floating Point DSP Block). 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. Check out an example low power mode FPGA here. There are both static and dynamic aspects to power consumption



Date: 01-06-2020
Course: PYTHON COURSE UDEMY

Name: POOJA K S
USN: 4AL17EC070

Topic: BUILD A WEBCAM

MOTION DETECTOR

The screenshot shows a Udemy video player interface. The video content displays a motion detector application running on a computer. The application window shows a webcam feed with a green bounding box around a person's face. The video player has a progress bar at the bottom, showing the video is at 19:23 / 20:38. The course content sidebar on the right lists sections 26 through 30, with section 27 'Application 6: Build a Webcam Motion Detector' highlighted. The Windows taskbar at the bottom shows the search bar and various application icons.

Udemy | The Python Mega Course: Build 10 Real World Applications

Course content

- Section 26: Python for Image and Video Processing with OpenCV
8 / 8 | 1hr 2min
- Section 27: Application 6: Build a Webcam Motion Detector
3 / 3 | 53min
 - 223. Webcam Motion Detector - How The Output Will Look Like
2min
 - 224. Detecting Webcam Objects
30min
 - 225. Capturing Motion Time
21min
- Section 28: Interactive Data Visualization with Bokeh
17 / 17 | 58min
- Section 29: Webscraping with Python Beautiful Soup
4 / 4 | 23min
- Section 30: Application 7: Scrape Real Estate Property Data from the Web
8 / 8 | 1hr 14min

About this course

A complete Python course for both beginners and intermediates! Master Python 3 by making desktop, web, and mobile apps.

Windows taskbar: Type here to search, 05:01 PM, 01-06-2020

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

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: 1) Find in front of screen time during working from home. 2) Monitor your child's in front of screen time. 3) Find trespassing in your backyard. 4) Locate unwanted public/animal movements around your room/house/alley and what not..... Photo by William 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. Here is the screenshot of what you are going to achieve:

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 code here 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 and capture 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.



GUVI CERTIFICATE:





Edit with WPS Office