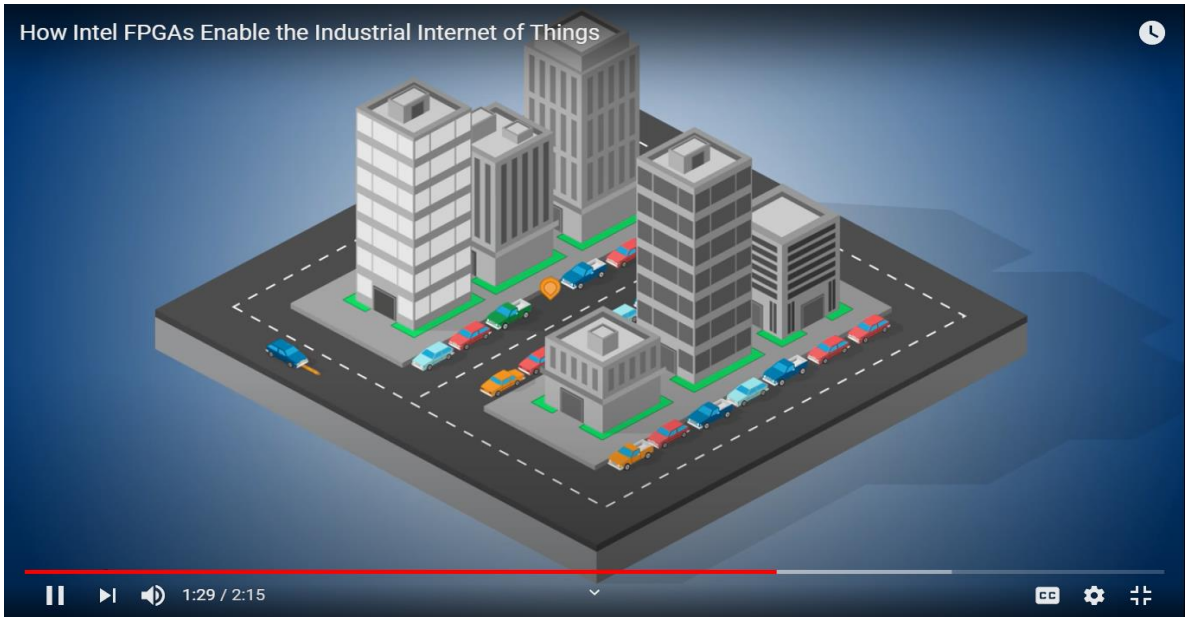
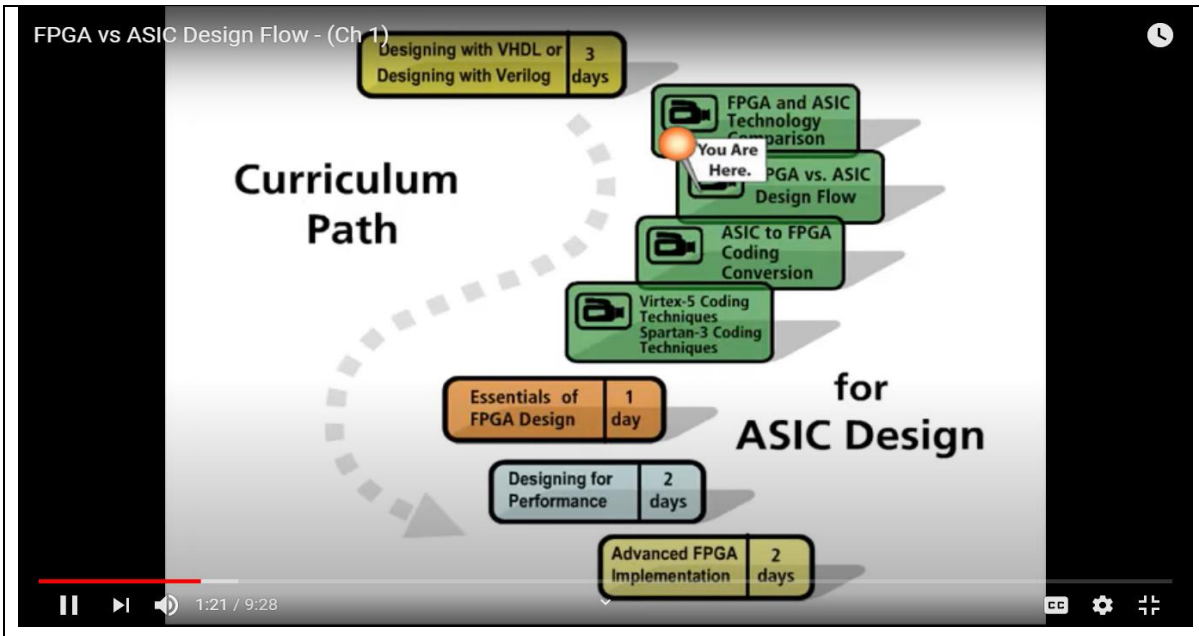
DAILY ASSESSMENT FORMAT

Date:	**01-06-2020**	**Name:**	**K B KUSHI**
Course:	**Logic Design**	**USN:**	**4AL17EC107**
Topic:	**1.Industry applications of FPGA 2.FPGA business fundamentals 3.FPGA vs ASIC design flow 4.FPGA basics- A look under the hood**	**Semester & Section:**	**6th & B**
GitHub Repository:	**https://www.github.com/alvas-education-foundation/KUSHI-COURSES**		

FORENOON SESSION DETAILS
Image of session 



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.**
- **Automation-Intel FPGA and SoC industrial automation solutions enable industrial system designers to reduce costs and time to market significantly for factory automation system designs.**
- **Embedded Vision-Intel FPGA and SoC solutions provide a rapid development path with the flexibility to adapt to evolving challenges and solutions for a wide range of video and intelligent vision applications.**
- **Smart energy-With a single Intel FPGA or SoC, you can better meet evolving standards for your design while increasing performance and scalability demands for mission-critical system functions like the control loop, grid communications, network redundancy, and security.**
- **Industrial IOT-Intel FPGA technology will be fundamental to delivering the value demanded of industrial systems within this software-defined automation environment.**
- **Accelerating innovation- In three minutes, see how Intel FPGAs enable Industry 4.0 and Internet of Things (IoT) across a wide variety of industrial applications like industrial automation, smart energy, and intelligent vision.**

- **ASIC stands for Application Specific Integrated Circuit. As the name implies, ASICs are application specific. They are designed for one sole purpose and they function the same their whole operating life. For example, the CPU inside your phone is an ASIC.**

- **FPGA is Not suited for very high-volume mass production while ASIC is Suited for very high-volume mass production.**

- **FPGA is Less energy efficient, requires more power for same function which ASIC can achieve at lower power but ASIC is Much more power efficient than FPGAs and Power consumption of ASICs can be very minutely controlled and optimized.**

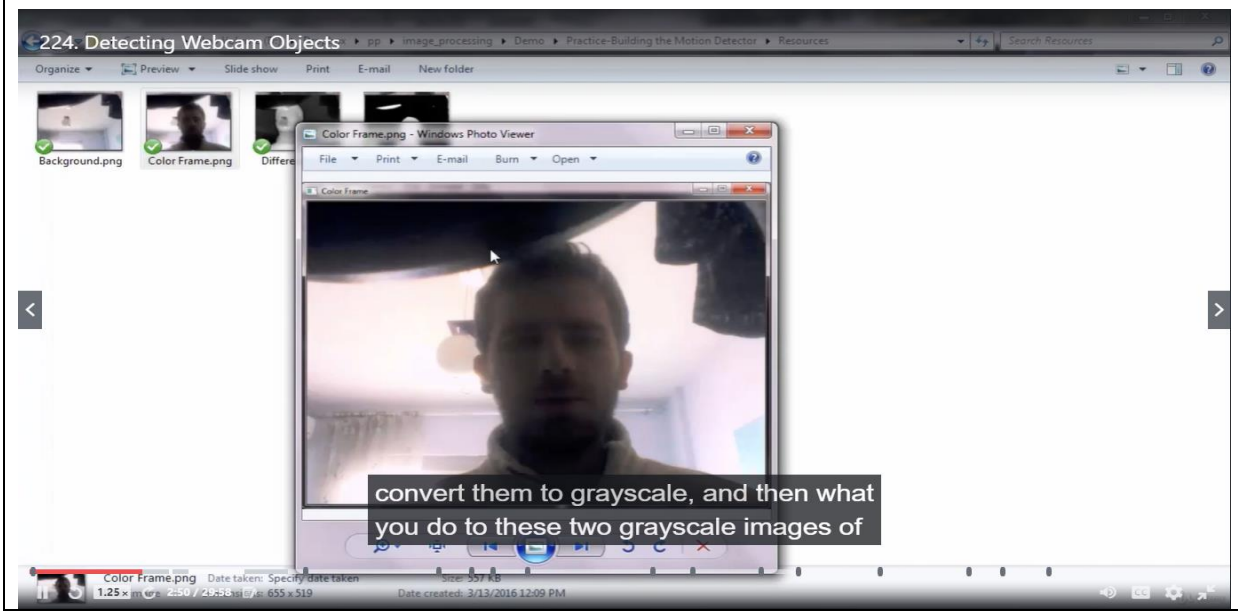
- **Analog designs are not possible with FPGAs but ASICs can have complete analog circuitry, for example WiFi transceiver.**

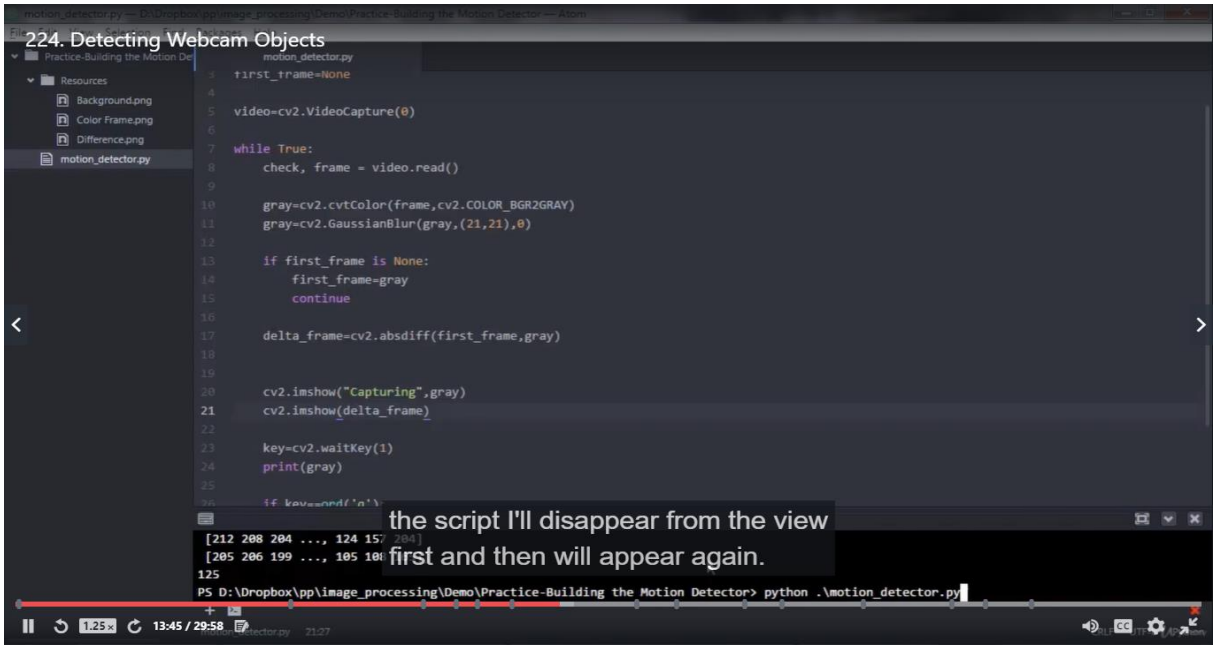
- **FPGAs are highly suited for applications such as Radars, Cell Phone Base Stations etc while ASICs are definitely not suited for application areas where the design might need to be upgraded frequently.**

Date:	**01-06-2020**	**Name:**	**K B KUSHI**
Course:	**Udemy-python**	**USN:**	**4AL17EC107**
Topic:	**1.Build a webcam motion detector**	**Semester &Section:**	**6th & B**
GitHub repository	**https://www.github.com/alvas-education-foundation/KUSHI-COURSES**		

AFTERNOON SESSION DETAILS

Image of session





Report –

- **In this python program will allow you to detect motion and also store the time interval of the motion.**
- **Videos can be treated as stack of pictures called frames.**
- **After running the code 4 new windows will appear on screen.**
- **Gray Frame: In Gray frame the image is a bit blur and in grayscale and they have only one intensity value**
- **Difference Frame: Difference frame shows the difference of intensities of first frame to the current frame.**
- **Threshold Frame: If the intensity difference for a particular pixel is more than 30(in my case) then that pixel will be white and if the difference is less than 30 that pixel will be black**
- **Colour Frame: In this frame the colour images in colour frame along with green colour are around the moving objects**
- **The time of movements will be stored in the folder where your code file is stored and In this file the start time of motion and the end time of motion will be recorded.**

- **Step 1: Importing the required libraries**
- **Step 2: Initialize variables, lists, data frames**
- **Step 3: Capture the video frames using webcam**
- **Step 4: Converting the captured frame to gray-scale**
- **Step 5: Capturing only the first gray frame**
- **Step 6: Creating a Delta Frame and a Threshold Frame**
- **Step 7: Adjusting the Threshold Frame and finding pixel in it**
- **Step 8: Capturing the time-stamp when objects enters and exits the frame**
- **Step 9: Displaying all the different frames**
- **Step 10: Generating the Time Data**