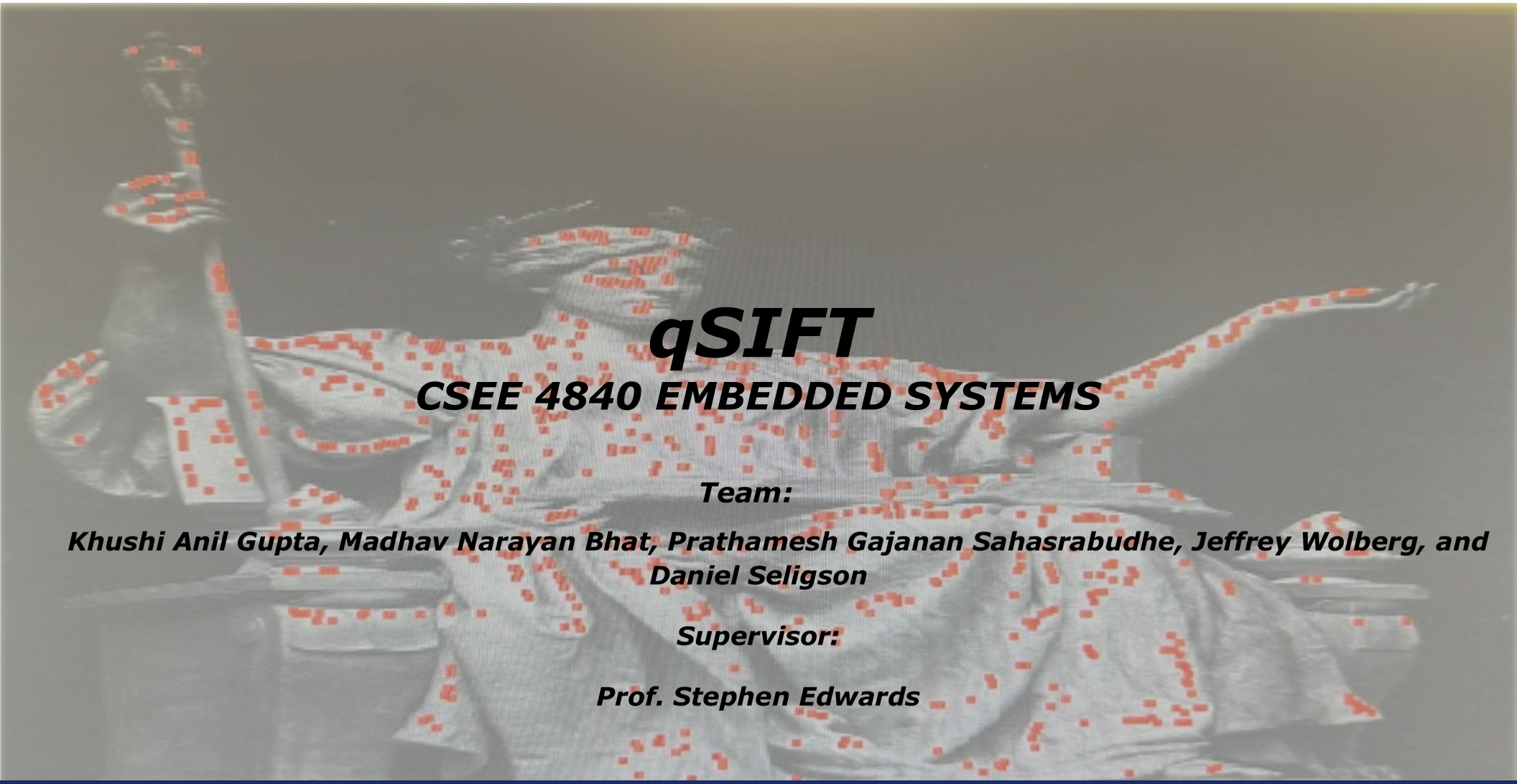


qSIFT

CSEE 4840 EMBEDDED SYSTEMS

Team:

Khushi Anil Gupta, Madhav Narayan Bhat, Prathamesh Gajanan Sahasrabudhe, Jeffrey Wolberg, and Daniel Seligson

Supervisor:

Prof. Stephen Edwards

May 12, 2023

COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

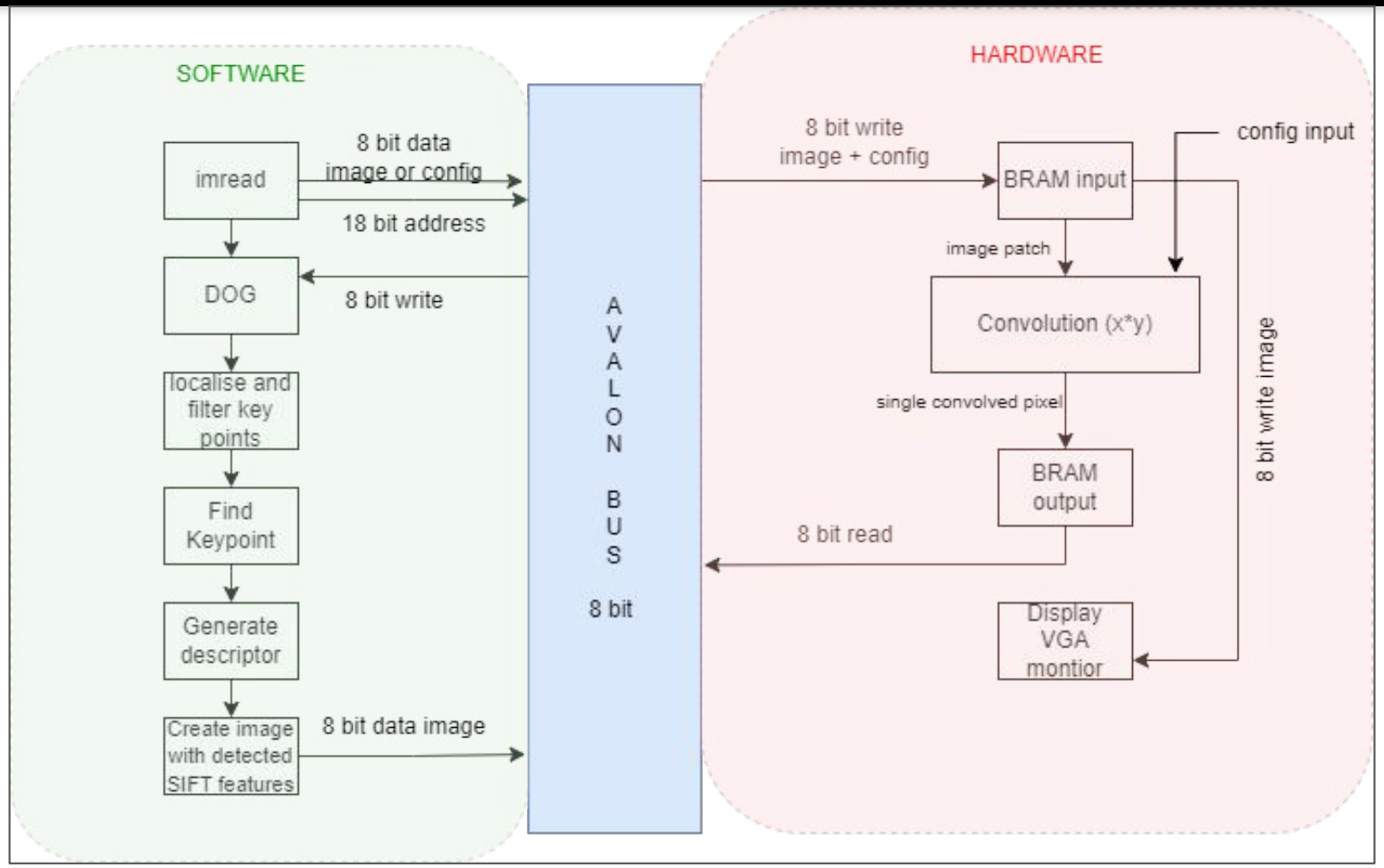
About SIFT

- Computer Vision algorithm for image detection and feature extraction. More useful because of it rotation and invariant properties.

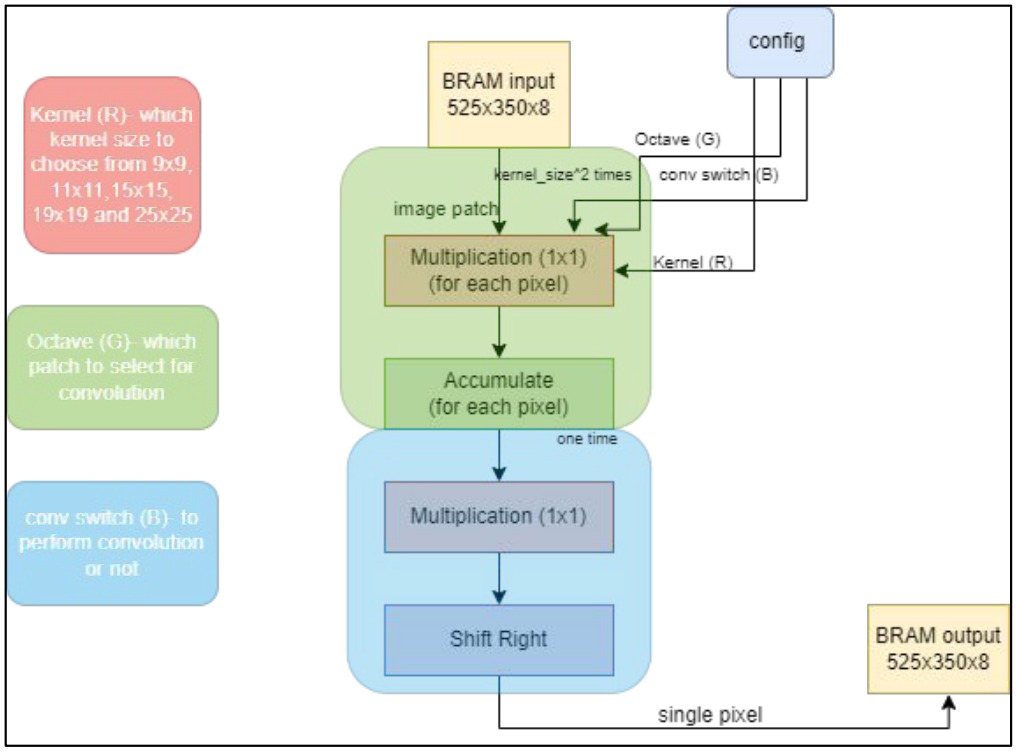
- Steps:
  1. Scale Space Detection
  2. Keypoint Localization
  3. Orientation Assignment
  4. Keypoint Description

- Applications- Computer Vision applications such as image stitching, object detection and modelling and other applications in augmented reality and robotics.

System Overview

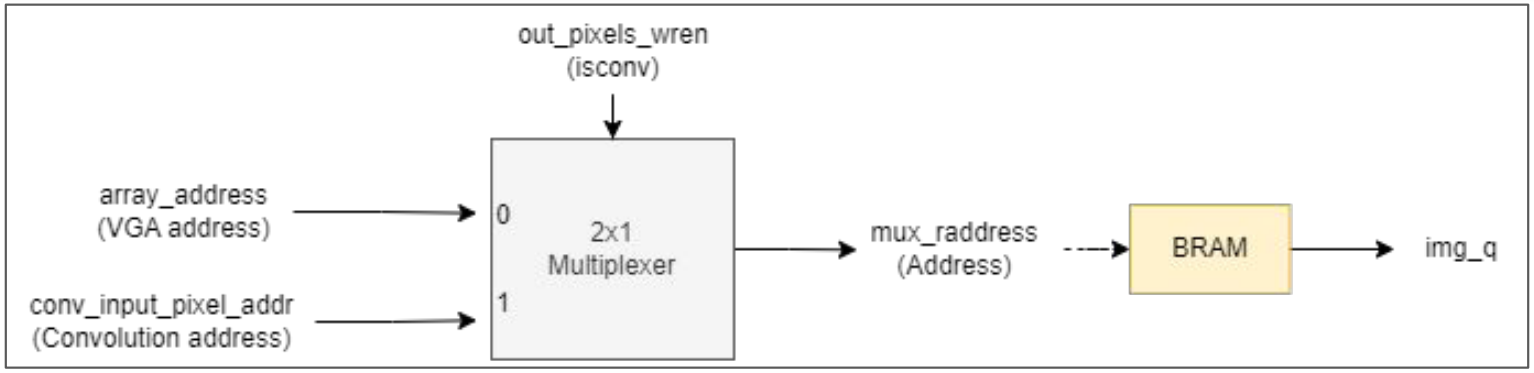


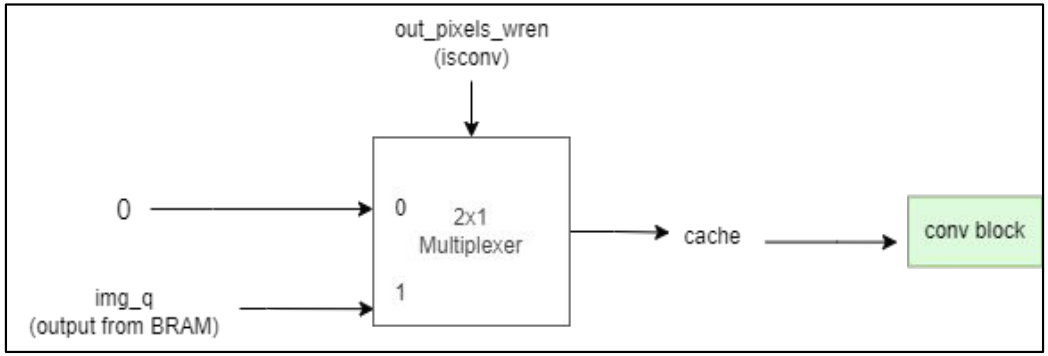
Hardware Design-Convolution Block

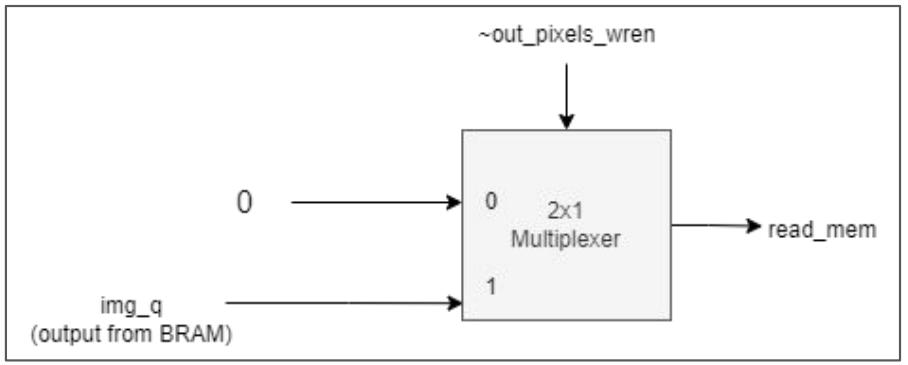


- The inputs to the FPGA are the kernel size, the octave, and an enable.
- As long as the enable is active, for each pixel in the 525 * 350 size image, we perform the convolution using a multiply and accumulate strategy.
- Once done, we need to scale the result to a value between 0 and 255, which we do through clever multiplication and bit shifting (no division)
- The final result is then stored into an output BRAM. Once the entire convolution is done, the data is sent back to the software by deactivating the enable signal.

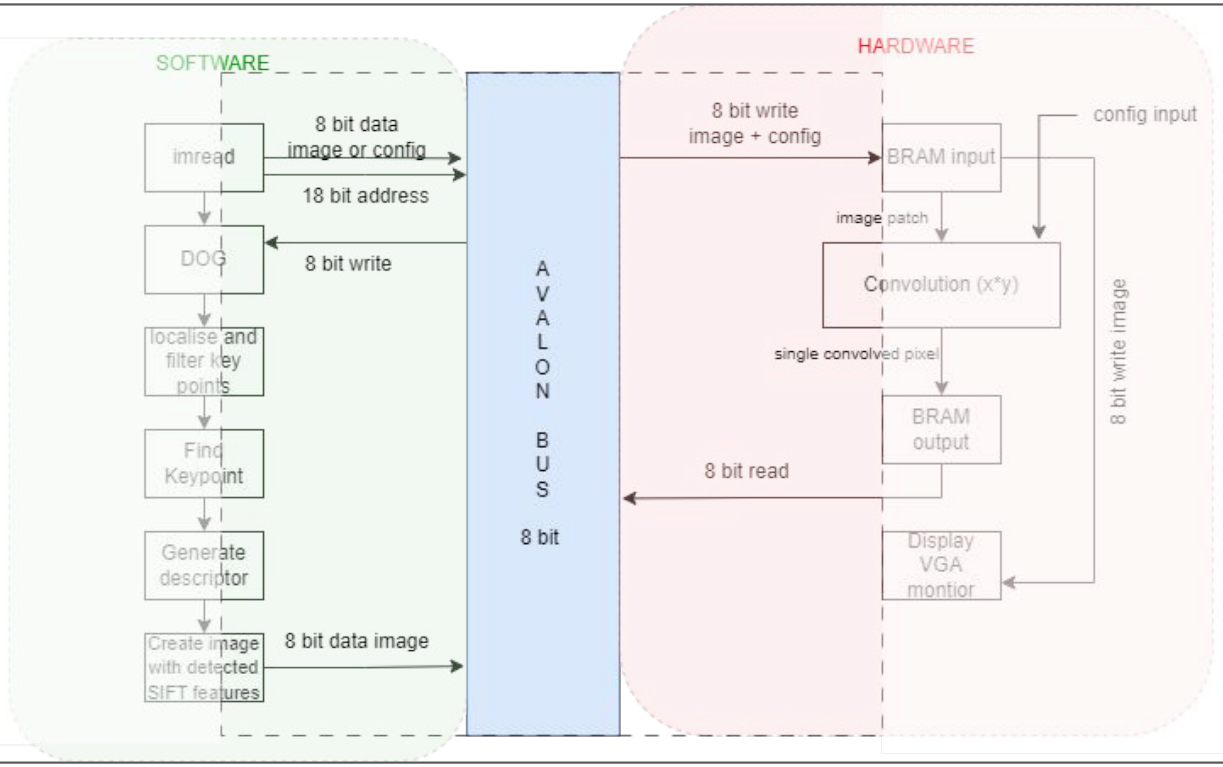
Hardware Design-VGA Block







Interface


SOFTWARE
HARDWARE
imread
8 bit data
image or config
18 bit address
DOG
8 bit write
localise and
filter key
points
Find
Keypoint
Generate
descriptor
Create image
with detected
SIFT features
8 bit data image
AVALON BUS
8 bit
8 bit write
image + config
BRAM input
config input
image patch
Convolution (x*y)
single convolved pixel
BRAM
output
8 bit read
Display
VGA
montior
8 bit write image

Software

Implementing the rest of SIFT:

- Calculating DoGs for each octave
- Finding keypoints if either min or max of neighborhood
- Filtering keypoints
- Marking keypoints on original image and sending back to HW

Project Workflow

Timeline(Feb 27 till May 11)

1. Proposal- Week 1
2. Designing the system - Week 2- Week 5
3. Loading the Image to and from the hardware- Week 6 and 7
4. Convolution- Week 7- Week 10
5. Interfacing the software- Week 10 and Week 11
6. Displaying the image- Week 11
7. Final Report- Week 11

Individual Contribution

1. Khushi- Hardware design and VGA display.
2. Prathamesh- Hardware design and interface.
3. Madhav- Hardware design, interface and VGA display.
4. Jeffrey- Software, SIFT algorithm, python testing
5. Daniel- Interface, software and VGA display.

Lessons Learned

- BRAM is very important.
- LAB 3 LAB 3 LAB 3.
- Learning underlying concepts makes everything easier.
- Start with hardware interfacing first or forget about your project.
- Keep hardware implementation simple unless you are a certified verilog god!
- Testing every part of code is very important.

Lastly be mentally prepared to spend your many nights (almost all) in MUDD 1235
You can chill while the quartus is building :)

DEMO TIME!!