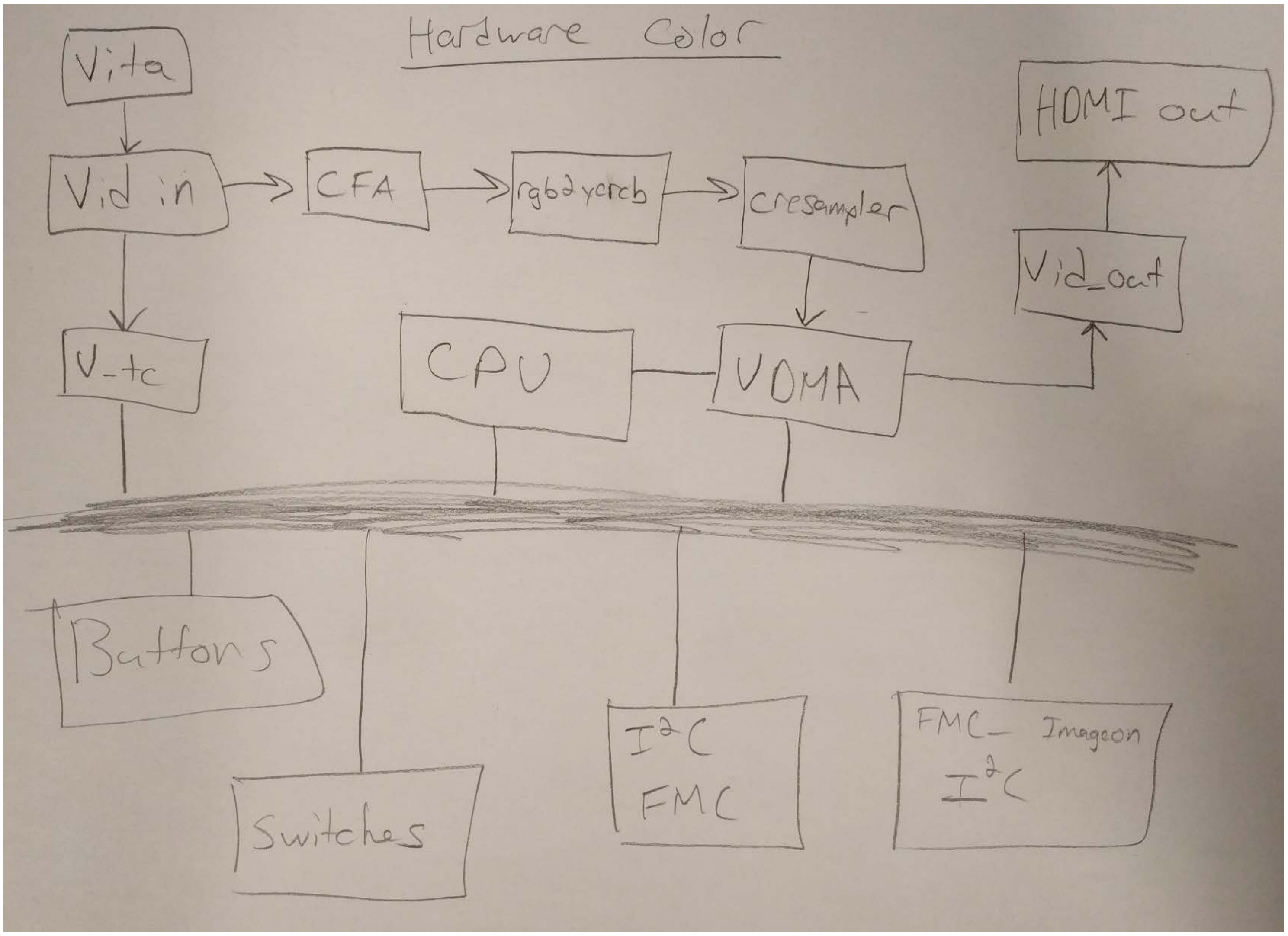
Kris Burney
Kyle Fischer
Khoi Cao
Group B-2
2/16/17

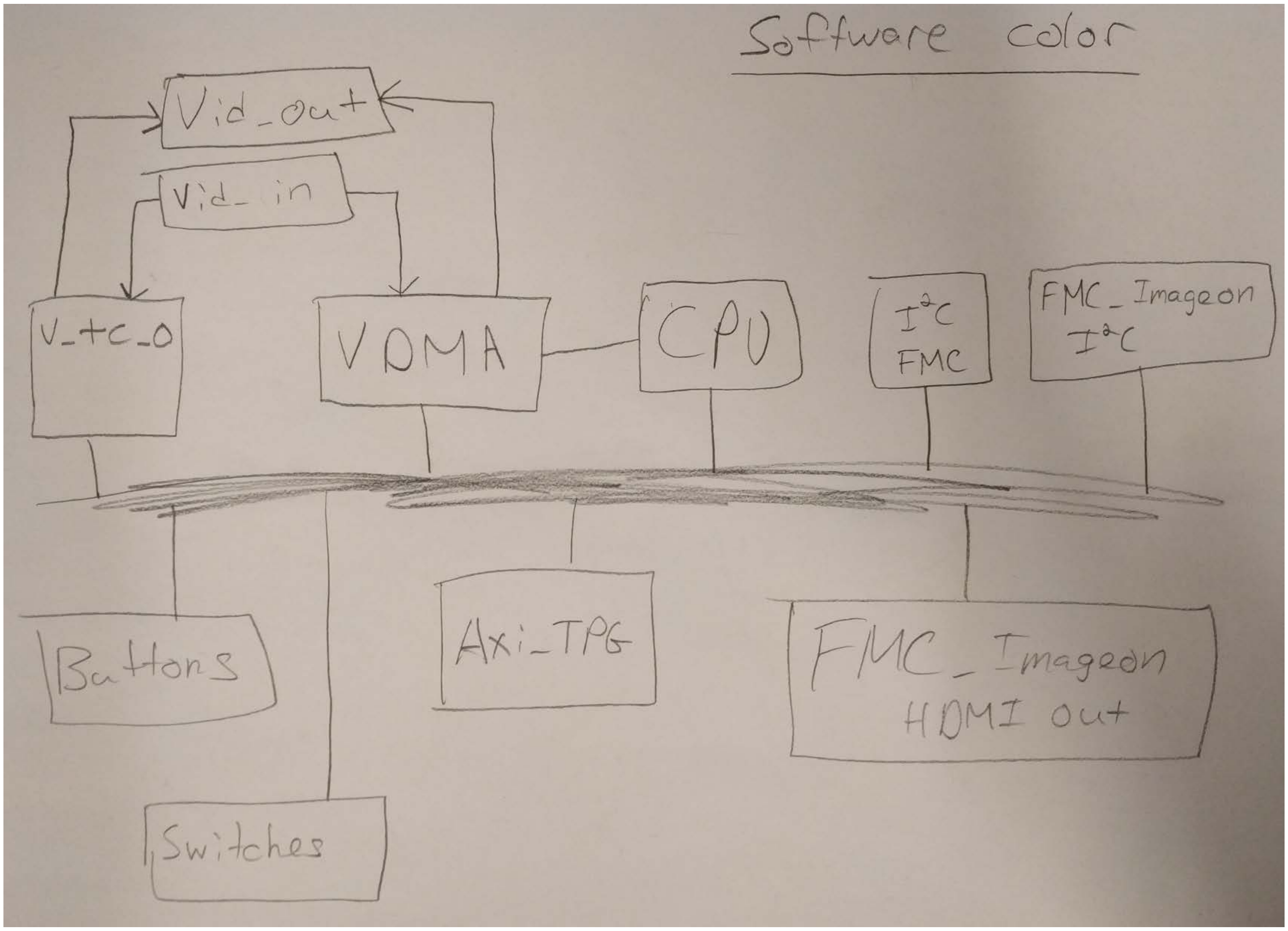
# MP-2

Digital Camera Design

## Design Test Analysis

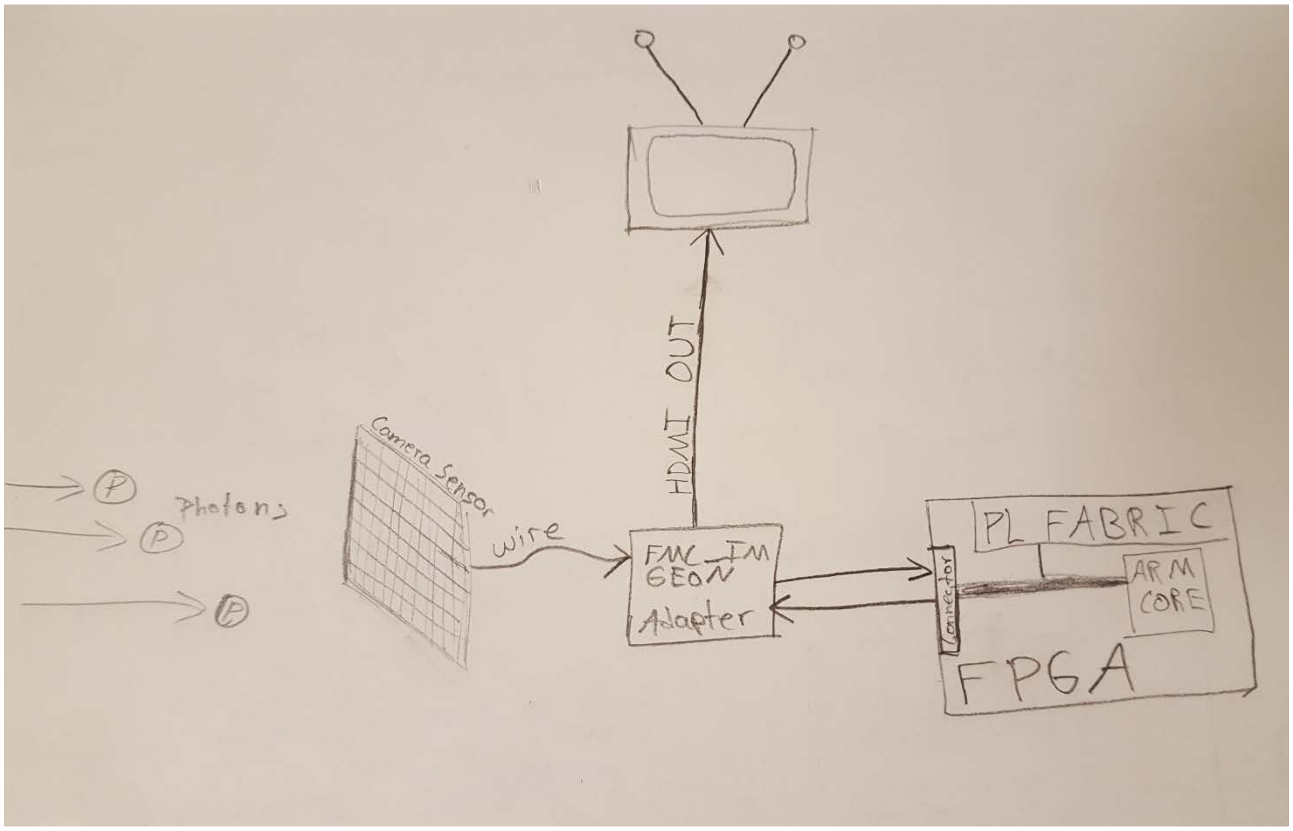


Kris Burney
Kyle Fischer
Khoi Cao
Group B-2
2/16/17



In the starting hardware for MP-2, vid_in sent the image data to the vdma. The vdma would store the data in a frambuffer in memory. Our software would then copy data from the input framebuffer to an output framebuffer. The vdma then send the output framebuffer data to vid_out. Our software could either directly copy the data from out input to output framebuffers or it could manipulate it by applying a bayer filter.

Kris Burney
Kyle Fischer
Khoi Cao
Group B-2
2/16/17



For part 3, we changed camera_app.c but not fmc_imageon_utils.c. In camera_app.c we used the modulo operator on our data to get different colored images. For example, moding by 255 made the image very green.

## Grayscale Camera

These wires are paired as a positive and negative wire. This wiring is used in low voltage differential signaling.

The output enable for the FMC is enables on lines 138-145 of the fmc_imageon_utils.c file. Depending on which output you wish to use.

The signal that the camera sensor sends out to the FMC adapter is not actually grey scaled. It is simply in a different color space than what we are outputting. It comes down to the location of the color bits. The relative locations of R, G and B data are not in the correct places for the way

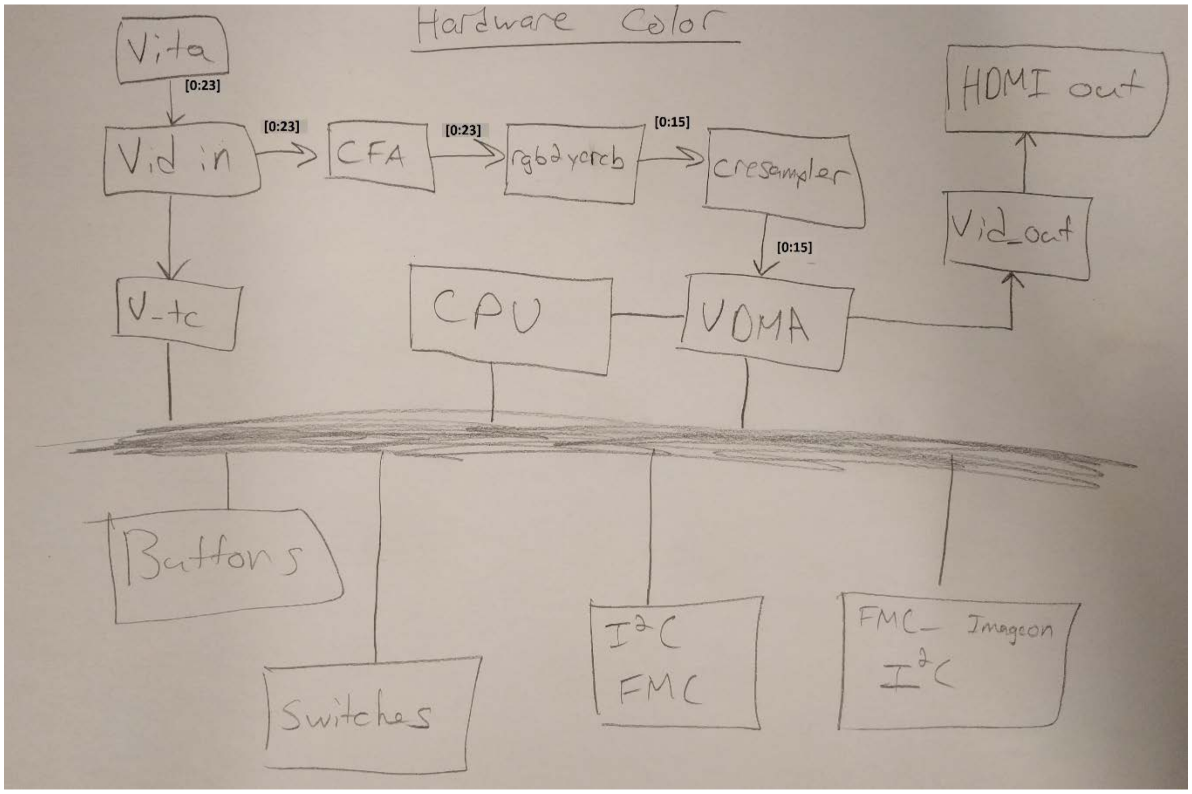
Kris Burney
Kyle Fischer
Khoi Cao
Group B-2
2/16/17

we are interpreting them. If we were to convert color spaces and interpret them correctly (given our system works) we would then see color.

## Color Conversion Software

1. We modified the function camera_loop() inside of the camera_app.c file.
    a. Added a color_lut to tell us if we are on a RED, GREEN or BLUE pixel
    b. Found the other two colors by averaging nearest neighbors of our pixel
    c. Convert to YCBCR format with the matrix multiplication provided in the lab document.
    d. Then finally either a YCR or YCB to be saved into the MM2S pointer.
2. Describe YCbCr in fmc_imageon_enable()
    a. This is a 4:2:2 encoding
    b. The first 4 bytes are the Y value
    c. The next two bytes are Cb and the next two are Cr respectively.
3. Our performance on the first go around was 0.5 fps. We measured this in the most accurate way possible. Simply by using our super intelligent brains to start and stop a timer on a watch. As the fps was so small, this was sufficient.

## Image Processing Pipeline



Kris Burney
Kyle Fischer
Khoi Cao
Group B-2
2/16/17

We couldn't get our hardware frame rate measurement working. Our plan was to implement a status register that was incremented in hardware every time a frame is completed. Our software could then poll the register at two different times and figure out our frames per second. We do know that applying color in hardware is a lot better than doing it in software.

## Participation

Khoi – 33%

Kris – 34%

Kyle – 33%

Kris went the extra percent this lab. He stayed late to fix some git issues.