

ECE554

Minilab 2 Report

Spring 2026

Group Members: Josh Mlaker, Michael Ferolito, Vedaant Nookala

Implementation:

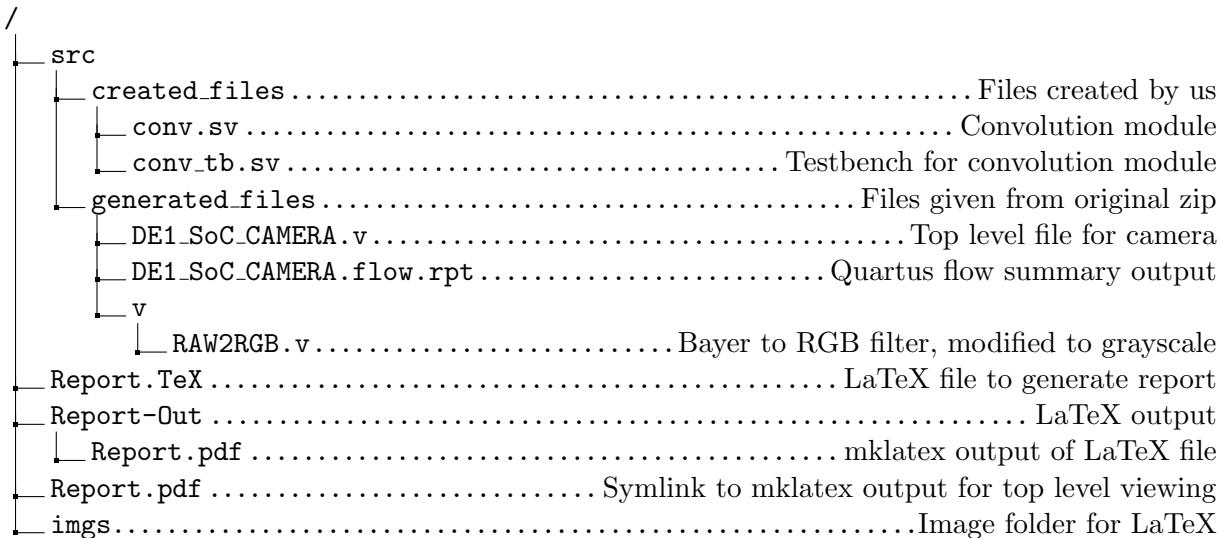
Grayscale: To implement the grayscale, we decided to average the RGB output of the Bayer output, resulting in a very similar, if not identical, result to taking a kernel over the raw camera output.

Convolution: Our convolution modules takes in a shift register of 2 rows of pixels + 3 extra pixels (totaling 1283 12-bit grayscale values). It then calculates an calculates the convolution by using offsets from the center pixel. That is, for a given convolution, it will use indexes 1282, 1281, and 1280 for the top row, 642, 641, and 640 for the middle row, and 2, 1, and 0 for the bottom row.

Testbench:

The only testbench used focuses around the convolution module. To test it, we inserted different 3x3 matrices and looked at the outputs, comparing it to an ideal models calculations.

All other testing was done on the fpga with visual inspection, repeatedly compiling and flashing the board, watching the output on the VGA display.

Project Layout:

In addition to the `created_files`, `DE1_SoC_CAMERA.v` and `RAW2RGB.v` were changed inside the `generated_files` to complete this Minilab.

Quartus Flow Summary:

```
+
; Flow Summary
+
; Flow Status           ; Successful - Sat Feb 14 11:11:13 2026
; Quartus Prime Version ; 25.1std.0 Build 1129 10/21/2025 SC Lite Edition
; Revision Name          ; DE1_SoC_CAMERA
; Top-level Entity Name   ; DE1_SoC_CAMERA
; Family                  ; Cyclone V
; Device                  ; 5CSEMA5F31C6
; Timing Models           ; Final
; Logic utilization (in ALMs) ; 4,560 / 32,070 ( 14 % )
; Total registers         ; 16612
; Total pins              ; 226 / 457 ( 49 % )
; Total virtual pins      ; 0
; Total block memory bits ; 59,344 / 4,065,280 ( 1 % )
; Total DSP Blocks        ; 0 / 87 ( 0 % )
; Total HSSI RX PCSs     ; 0
; Total HSSI PMA RX Deserializers ; 0
; Total HSSI TX PCSs     ; 0
; Total HSSI PMA TX Serializers ; 0
; Total PLLs               ; 1 / 6 ( 17 % )
; Total DLLs               ; 0 / 4 ( 0 % )
+
```

Figure 1: Flow Summary report from Quartus

In the summary, we see a surprisingly low usage of BRAM considering we opted to use a very large shift register inside our design rather than an IP module. The ALM usage is expected considering the amount of values needing to be added between the grayscale calculation and the convolution. It is surprising to see no DSP usage. However, that's likely due to the simpler nature of our math (as we used exact offset for convolution) rather than true matrix multiplication.

Difficulties & Solutions:

Problem 1. Our first problem was getting a constant black display with no signs of motion.

Solution 1. The solution was an to fix an improper instantiation of the shift register.

Problem 2. The second problem was duplicate images on the VGA display.

Solution 2. To solve this, we delved into the specifics of the RAW2RGB and saw it relied on a “valid” signal to determine when to actually send the RGB values. Thus, we only inserted a pixel into the shift register if it was “valid.”