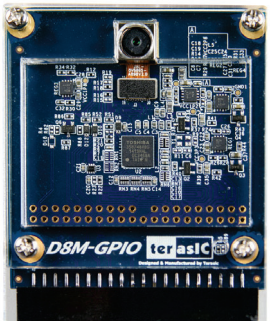
UNIVERSITY OF CALIFORNIA, DAVIS
Department of Electrical and Computer Engineering

EEC 181A **DIGITAL SYSTEMS DESIGN PROJECT** **WINTER 2025**

LAB 5: DE1-SOC CAMERA, VIDEO PROCESSING, AND VGA DISPLAY

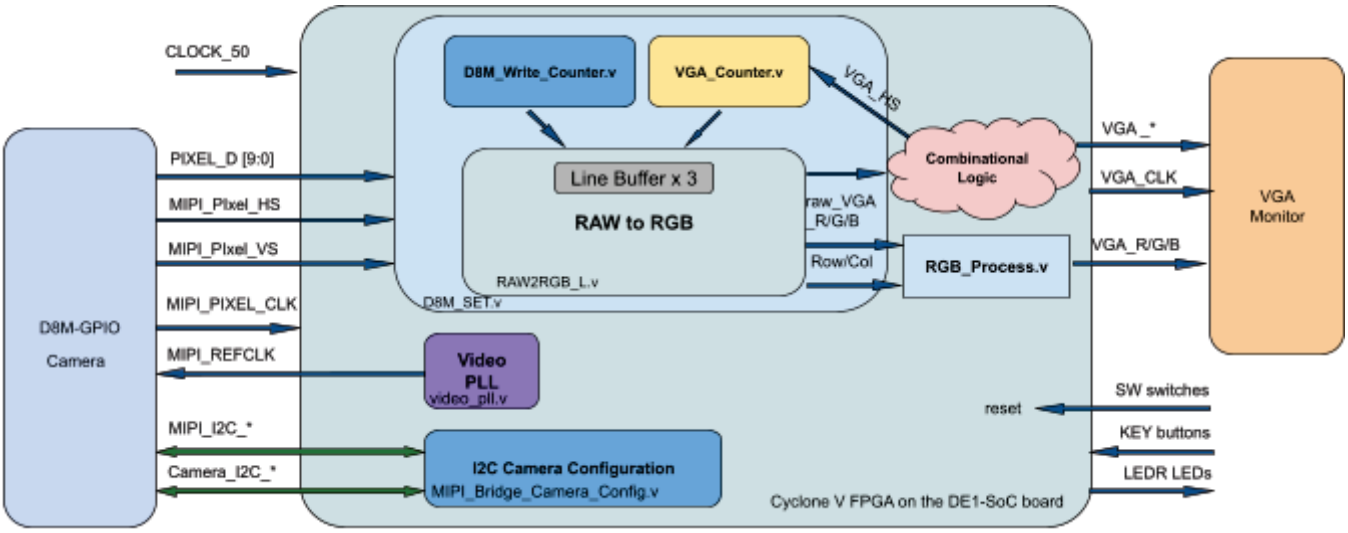
I. Introduction



This lab includes capturing pixel data from the digital camera, converting the raw image into an RGB format, performing some simple video processing in real time, and displaying the image onto a VGA display.

See the *Using the D8M-GPIO 8-Megapixel Camera* tutorial on the EEC 181A/B website for more information.

II. The Top-Level Module



D8M_SET.v: This module is used to process D8M camera 640×480 60 Hz raw data. It includes three line buffers that are used to store raw data. The line buffers are also used to adjust the frequency difference between the D8M camera and the VGA display.

Port Name (Verilog)	Direction	Width	Description
Clock_50	Input	1	Clock @50 MHz
RESET_SYS_N	Input	1	System reset
CCD_DATA	Input	10	Camera input RAW data
READ_EN	Input	1	Enable the Camera
sCCD_R/G/B	Output	8/8/8	Converted RGB data
col	Output	13	The col range of camera output (0 - 616)
row	Output	13	The row range of camera output (0 - 477)

Table 1. Ports of Verilog Module D8M_SET.v

MIPI_BRIDGE_CAMERA_Config.v: The D8MI2C setting controller, such as set D8M to output 640×480 @60Hz timing. It mainly writes I2C corresponding parameters to D8M MIPI decoder IC register and Camera Sensor IC register respectively through their own I2C buses. MIPI_I2C bus is used to write MIPI decoder IC (I2C Slave Address = 0x1c.), CAMERA_I2C bus is used to write Camera Sensor (IC Slave Address = 0x6c).

IV. Real-Time Image Processing

Implement the following video processing functions, or filters, on the stream of pixels coming from the camera and displayed the VGA monitor in real-time. SW switches switch the functions between the various filters.

Many details of the designs are not specified (e.g., which SW switches to use and how to encode their settings to select filters, and how to switch between various filter sub-functions such as the different levels of RGB color reduction); these details must be chosen by you and described fully in your submission.

Limited by the number of SW switches, try to implement your hardware so that multiple filters can operate at the same time where it makes sense.

Use the SW0 switch for your design's *reset* signal.

1. [30 pts] RGB color filters

    a) Using SW switches, modify the red intensity in the following steps: normal, ½, ¼, off

    b) Using SW switches, modify the green intensity in the following steps: normal, ½, ¼, off

    c) Using SW switches, modify the blue intensity in the following steps: normal, ½, ¼, off

2. [40 pts] Cursor

    Draw the following two cursors on the display in blue+green pixels. Use the four KEY buttons to move the cursor around the screen up/down left/right. Design some kind of method and circuit that allows the user to move one cursor at a time. The cursor should never leave the screen and stop when it comes to an edge. Use a counter like the one given in the handout *Variable Frequency Clocking Hardware* to slow the cursor's movement so a user can control its movement easily and precisely.

    a) 3×3 pixel block

    b) 10×10 pixel box with a 1-pixel border

3. [30 pts] Grayscale conversion

    Convert the color pixels to grayscale using the equation below. After calculating the luminance, set all three VGA color signals to the same luminance value. Use 8-bit by 8-bit multipliers for the multiplications. The coefficients should be encoded in unsigned 8-bit fractional format (0.XXXXXXXX) which has a range of [0, +255/256].

    Luminance = 0.2126 red + 0.7152 green + 0.0722 blue

2025/03/03 Posted.