Project Specification ECE 484W

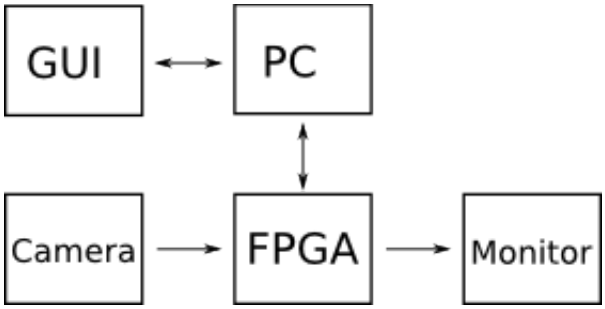

Figure 1. Data flow in the real-time video editing system.

Milestone 1:

1. Develop a command-line program that loads an 8-bit bitmap grayscale image file (original image) and an 8-bit image grayscale overlay file (overlay image, must have same size) and creates two 8-bit images as output:
    1. One image (out1) that contains the original image overlaid (IO) with the overlay image. While there are several useful ways of overlaying, the one that you need to implement is the following: if a pixel is black in the overlay image, the corresponding pixel in out1 is white. If a pixel is not black in the overlay image, the corresponding pixel in out1 equals the corresponding pixel in the original image. We will call this way of processing image overlay (IO).
    2. One image (out2) that contains a histogram-equalized (HE) version of the original image. To generate out2, first compute the histogram of the brightnesses of the original image. From the histogram, compute the cumulative histogram. Once you have the cumulative histogram, you can compute the pixel values of out2 pixel by pixel from the original image.
2. The program has to be written in C++.
3. The program needs to work for original/overlay images of any reasonable size (up to 1024x768 pixels).
4. The program needs to be tested on both test images provided on BB (modules section, the names of the images are test1.bmp and test2.bmp).

Milestone 2:

1. Create a graphical user interface that allows the user to take the following actions:
    1. Load/display an 8-bit bitmap original image.
    2. Load/display an 8-bit bitmap overlay image.
    3. Generate and display an output image by overlaying the overlay image over the original image (as described in MS1).
    4. Generate and display and output image from the original image by histogram-equalizing it (see MS1).
    5. Adjust image brightness with a slider (this slider sets a factor by which the brightness is multiplied)

6. Adjust image contrast with a slider (this slider configures a transfer function that either compresses the brightness histogram for lower contrast or stretches it for higher contrast).
7. Save the output image as an 8-bit bitmap file.

2. The GUI has to be written in Qt.

Milestone 3:

Program Altera's DE2-115 board (which includes a “Cyclone IV” FPGA) via the Quartus software to perform the following tasks (the items listed are separate tasks):

1. Use two slide switches as inputs for an AND gate, display the output with one of the LEDs.
2. Use a push button as input for a 8-bit counter and display the counter value on 8 of the LEDs.
3. Use a push button as input for a two-digit, binary-coded decimal counter (BCD) and display the value of the counter using two of the board's seven-segment displays.

   An important resource for this milestone is the DE2-115 user manual, available for download on BB (modules section). You are highly encouraged to use VHDL (VHSIC Hardware Description Language) to program the DE2-115. VHDL is a high-level programming language that the Quartus software can automatically convert into a FPGA design, many sample program are available online.

Milestone 4:

Establish communication between your Qt GUI from MS2 and Altera’s DE2(-115) via the USB port. The following functionality is required:

1. When you move the brightness slider in your GUI, the new value should be transmitted the Altera board and displayed on the 7-segment display.

This is not as simple as it sounds. On your Altera board, you need a way to receive data, the simplest way is the use the RS232 interface. This involves programming a soft microprocessor (called NIOS) onto the FPGA, I recommend the “Basic Computer” for the DE2-board (see https://www.altera.com/support/training/university/materials-computer-systems.html) and then programming it to receive data from the RS232 interface and displaying it on a 7-segment display. On the Qt side, you need to send data through the serial port; for this, you should use a library such as http://www.teuniz.net/RS-232/.

Milestone 5:

In the first part of this milestone, you use the “Media Computer” by Altera (an extension of the “Basic Computer” that you used in the last MS) to display video data on an external monitor. In more detail:

1. In your Qt GUI, add buttons for “transfer test image” and “transfer overlay image”.

   If the “transfer test image” button is pressed, the test image should be sent through theRS232 interface to the FPGA board, transferred into video memory, and displayed on an external VGA screen. Note that the “Media Computer” only supports a resolution of 320x240; it is fine if you restrict your test images to this resolution.

   If the “transfer overlay image” button is pressed, transfer the overlay image (which can also be restricted to 320x200 resolution) via the RS232 interface to the FPGA board and store it. Extend your NIOS program to perform the overlay and display the resulting picture on . You need to implement a mechanism in your Qt interface to turn the overlay at the FPGA on and off.

2. Enhance the functionality of your brightness and contrast sliders by having them communicate their values through the RS-232 interface to the FPGA board, and enhance your NIOS program so that it uses brightness and contrast values to adjust the picture in the same way that your Qt program does.

In the second part of this milestone, you need to write short essays (approx. 1 page) on the following subjects:

1. Professional and ethical responsibilities in hardware and software development. Use examples that are related to our class material as much as possible.
2. How have hardware and software development impacted societies (in America and the rest of the world)? Please include positive and negative impacts. Each of you should be involved in at least one of the essays (so they can be written by more than one person).

Complete project

Add a TRDB-D5M CMOS camera to your project.

1. Test the camera with the reference FPGA design that the manufacturer (Terasic) supplies with the camera.
2. Make an FPGA design that includes both the VGA camera controller from the reference design and the Altera Media Computer that you have used in MS5. This design will allow you to pass camera data through the NIOS processor to the VGA output. I recommend using Qsys for making the design.
3. Use your GUI from MS2 and the RS232 communication from MS5 to allow a user to
    1. Add an overlay image to every frame of the video stream
    2. Adjust brightness and contrast of the video stream.

The final report is not only on the functionality added after the fifth milestone, but should be a complete description of the final design. It cannot reference milestone reports, but you are welcome to reuse text from these reports.

It is expected that each student spends 10 to 15 hours a week on this project in order to achieve the project goals.