Project Objective

In this project, you will

1. learn how to implement a video system on FPGA.
2. get introduced to video hardware functions.
3. develop an application that runs on the video system.

Hardware Requirements

a. Remove the 7-segments components, as well as onchip_memory component, bcd_switches and speed_switches. Keep only the system_mode switches.

b. Rename the clock source as *clock_source_0*, nios processor as *processo*r and JTAG-UART as *jtag* components.

c. Make sure that your processor is on the fast mode (niosII/f) and the instruction and data cache is 4KB.

d. Add the following components and setup its parameters as shown below. You may also use the diagram in L.6 as a reference.

1. Clock_source_1

name	Conduit name
clock_source_1	sdram_clk

2. System and sdram clock

name	Setting
sys_sdram_pll	DE-Series Board: DE2-115

3. Dual-Clock FIFO

name	Color bits	Color planes
dual_clock_fifo	10	3

4. RGB Resampler

name	Incoming format	Outgoing format
rgb_resampler	24-bit RGB	30-bit RGB

5. SDRAM controller

board device	name	Conduit name	Bit Width	Architecture	Address width	Generic memory
DRAM	sdram_controller	sdram	32	Chip select: 1, Banks:4	Row: 13, Col: 10	Enable

6. SRAM Controller

Name	conduit	Setting	Enable
sram_controller	sram	DE2-115	Use as pixel buffer for video out

7. Pixel Buffer DMA Controller

name	Addressing Mode	Frame Resolution	Pixel Format
dma_buffer	Consecutive	Width 320, Height 240	24-bit RGB

8. Video Alpha Blender

name	Addressing Mode	Clock	Connection
Video Alpha Blender	Simple	cys_cllk	Connect foreground to video Character buffer. Connect Background to Video Scaler. Connect output to Video FIFO.

9. Video Character Buffer

Transaction	Clock	Connection
Enable	cys_cllk	Connect data/control to processor Avalon data bus. Connect output to foreground Alpha blender

10. Scaler

name	Width factor	Height factor	width	Height	Data bits per symbol	Symbol per beat
video_scaler	2	2	320	240	10	3

11. VGA Controller

name	conduit	DE-series Board	Video out device	VGA Resolution
vga	vga	DE2-115	VGA connector	VGA 640 x 480

12. Video clocks for DE-series Boards

Input settings	Video out setting
50 MHz	Enable vga clock

e. Set reset and exception vectors of NIOS processor to sdram_controller (instead of onchip memory as we had in previous projects).

f. In Blackboard, along with the assignment, you will find video system_connections.pdf attached which can be used as a reference to connect the components.

g. **All resets must be connected to the Reset Output (clk_reset) of clock_source_0

Software Requirements

You are required to write a C/C++ application code to execute the functions shown in the following table. Several things are explained below as a guidance you use to write the code.

mode	Function
0	Display the gamecock image on the screen
1	Resize the image by factor of 0.5 (both width and height)
2	Resize the image by factor of 2 (both width and height)
3	Print message “Video project”

1. A software template (ctemplate.c) is attached which you can use it as your reference.
2. Two attached documents (file.c and file.h) include the source code of gamecock image and the header file respectively.
3. You should add these two files (file.c and file.h) into your application project by doing the following:
    a. Download the two files into any directory (for example download folder).
    b. Then, open the folder containing the files (the download folder)
    c. Drag these two files into your application project.
    d. The instructor/TA will also explain this during the lecture/lab.

4. To access the pixel buffer and dma functions, call the following the header in your application C/C++ code:

```
#include <altera_up_avalon_video_pixel_buffer_dma.h>
```

5. To read the pixel_buffer (or the sram content), use the following code:

```
alt_up_pixel_buffer_dma_dev *my_pixel_buffer;
my_pixel_buffer =
alt_up_pixel_buffer_dma_open_dev("/dev/dma_buffer");
```

6. After that you need to check if the pixel buffer array (my_pixel_buffer) contains the image pixel to do so, add the following code:

```
if(!my_pixel_buffer) printf("Error opening pixel buffer\n");
```

7. Anytime you want to display an image on the screen, try first to clear the screen by calling the following function:

```
alt_up_pixel_buffer_dma_clear_screen(my_pixel_buffer,0);
```

8. To display an image, you need to use the following function and include the following header file:

```
alt_up_pixel_buffer_dma_draw(my_pixel_buffer,
          (myimage[(i*320*3+j*3+2)]) +
          (myimage[(i*320*3+j*3+1)]<<8) +
                    (myimage[(i*320*3+j*3+0)]<<16),j,i);
```

9. To resize the image by any factor f (where f is a float) and display the new image. For example, to double the image size, f=2.0. You may use the following code.

```
alt_up_pixel_buffer_dma_draw(my_pixel_buffer,
          (myimage[(i*320*3+j*3+2)]) +
          (myimage[(i*320*3+j*3+1)]<<8) +
                    (myimage[(i*320*3+j*3+0)]<<16),j*f,i*f);
```

Project Report (50%)

The project report will be graded out of 100, and the points will be distributed as following:

1.0 (*30 points total, each value is 10 points*) Run the application project and take a picture for the screen that shows the execution of each function: display image, resize image by 0.5 and resize image by 2.

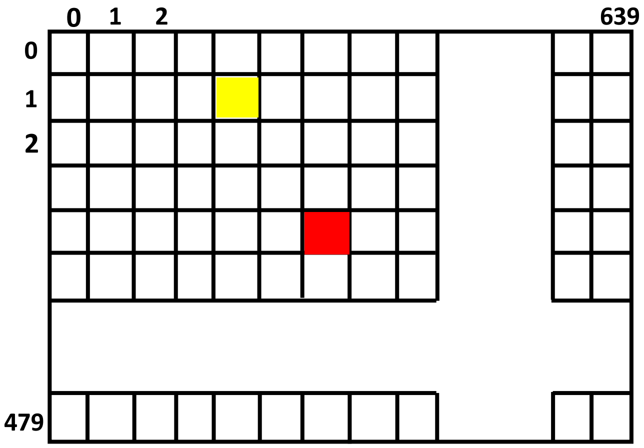
2.0 (*70 points*) Answer the following questions:

a. You are given the following frame.

1. (*15 points*) Find its resolution
2. (*15 points*) The coordinates of the red pixel.

b. Assume you are required to map the given frame as 24-bit RGB into a pixel memory which its base address is 0x08000000.

1. (*20 points*) Find the pixel address of the yellow pixel.
2. (*20 points*) Find the frame size



Project Demo (50%)

- The main purpose of the demo is to test your project functionality and execution.
- Demos will be graded out of 100, but worth 50% of total project grade.
- Both partners must show up in that day. If a member didn't show up, he/she receives 0 unless an excused absence was provided.
- Demos will be conducted during the lab time on the following dates:
  - Wednesday March 15th for group1
  - Friday March 17th for group2
- Below are how the demo points will be distributed.

Tasks	Point
Display image at mode=0	/25
Resize image by 0.5 at mode=1	/25
Resize image by 2 at mode=2	/25
Questions	/25

Project Submission

1. Save the project report as **r4_username1_username2.pdf**, username of both students in the group. Five Points will be deducted if the group didn’t save their files as it is explained above.
2. For this project, you are required to submit only the project report (No project submission is required). Submission date is Tuesday November 1st before the class starts.
3. **Only one attempt** is allowed.
4. **Only one group member** can submit the report.
5. **Remember:** Any grade dispute must be raised within one week of the grade posting.