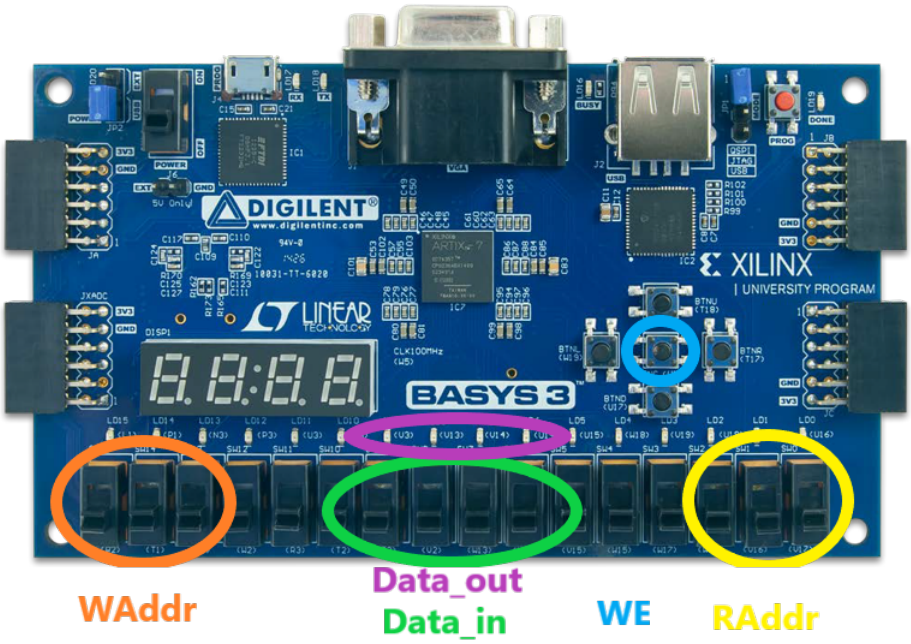
ECEN 240 Lab 9 – Register File

(Vivado and SystemVerilog Instructions)

Your objective is to build the register file circuit using the Basys3. The following switches, LEDs, and button assignments will be made:



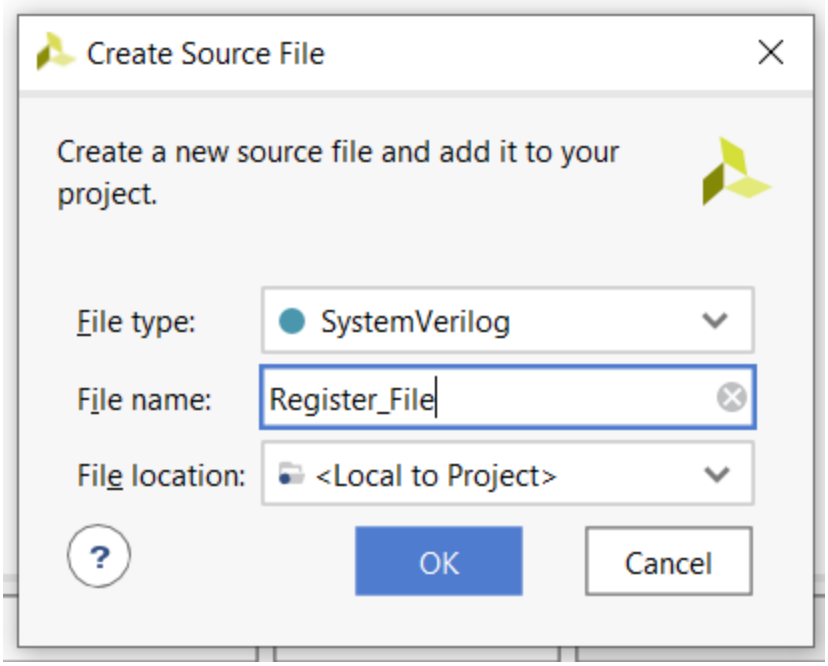
Procedure:

1. Download the “clk_div.v” Verilog module file and the “Register_File.xdc” constraints file from Canvas and place them in a convenient place (like your desk top).

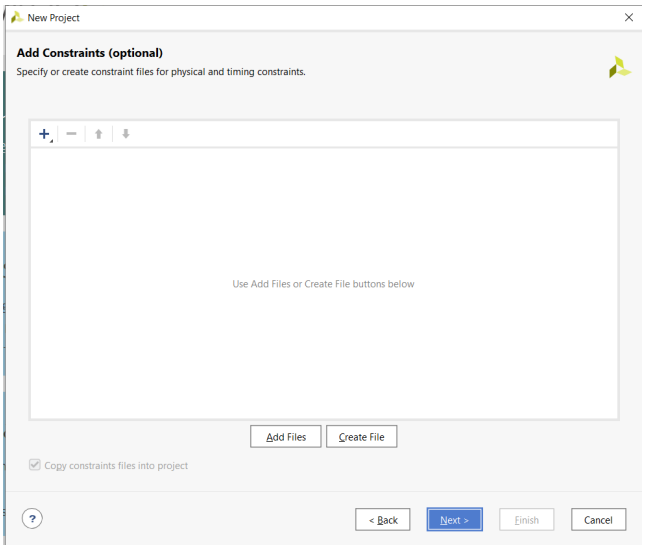
2. Create a Lab9_Register_File project.

3. You will add two sources to your design:

   - Create a new SystemVerilog module file called “Register_File.sv”.
   - Add the “clk_div.v” Verilog module file as a design source.



4. After you have added the two source files, select “Next” and add the “Register_File.xdc” as a constraints file. The constraints file will eliminate the need to create the table with the pin number assignments (you can always do this later, or if you choose, you can add or alter the pin connection information manually as in the past).

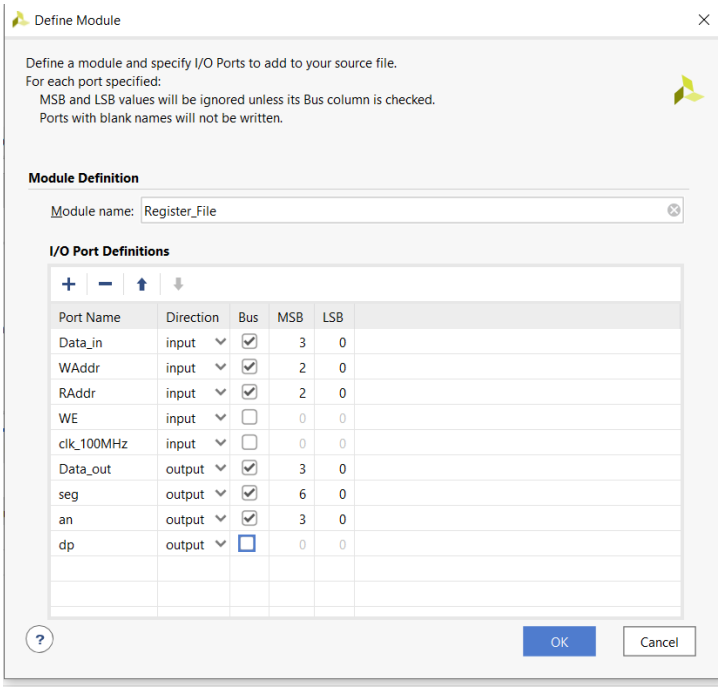


5. Configure the project with the correct FPGA filter Settings:

- Family: Artix-7.
- Package: cpg236
- Speed: -1

6. Select “xc7a35tcpg236-1” and then “Next”.

7. After selecting the “Finish” button you can opt to set up the ports for your “Register_File” module. You can include the following port information, (but remember that you can always change this later with the text editor):

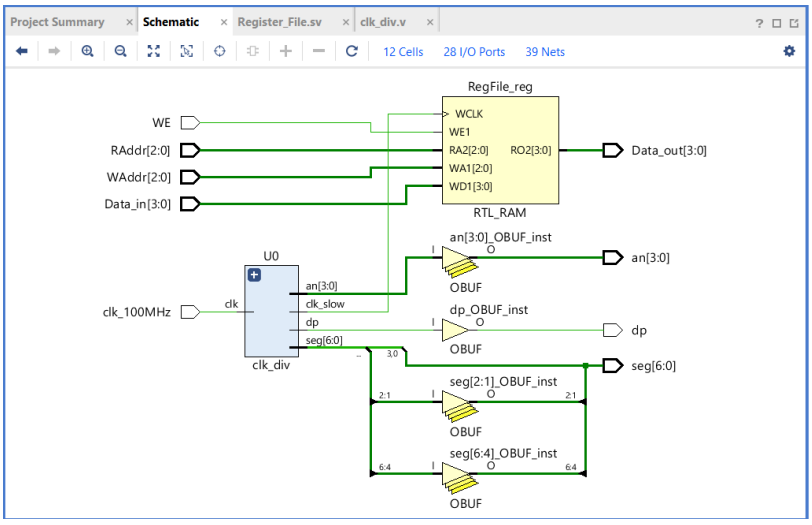


8. Inside of the “Register_File” module, instantiate the “clk_div” module just as you did in the previous lab. You will use the slower clock for your register file.

9. With the guidance of Chapter 19.3 of the textbook, design your register file as prescribed earlier. Just as with the Logisim Evolution schematic, it must have the following specifications:

- Eight memory locations—each with 4 bits (4X8 memory)
- Write enable
- Separate write address and read address
- Synchronous write (needs a clock to write)
- Asynchronous read (does not need a clock to read)

10. Open the “Elaborated” design schematic. It should look something like this:



11. Since you added a constraints file, you do not need to tell Vivado which pins to use on the FPGA chip. You can still click on the blue "28 I/O Ports" at the top of the schematic menu to see if the information is correct. You should see something like this:

Tcl Console | Messages | Log | Reports | Design Runs | **Find Results**

Name	Direction	Interface	Neg Diff Pair	Package Pin	Fixed	Bank	I/O Std	Vcco	Vref
WE	IN			U18	✓	14	LVCMOS33*	3.300	
clk_100MHz	IN			W5	✓	34	LVCMOS33*	3.300	
dp	OUT			V7	✓	34	LVCMOS33*	3.300	
Data_in[3]	IN			T3	✓	34	LVCMOS33*	3.300	
Data_in[2]	IN			V2	✓	34	LVCMOS33*	3.300	
Data_in[1]	IN			W13	✓	14	LVCMOS33*	3.300	
Data_in[0]	IN			W14	✓	14	LVCMOS33*	3.300	
Data_out[3]	OUT			V3	✓	34	LVCMOS33*	3.300	
Data_out[2]	OUT			V13	✓	14	LVCMOS33*	3.300	
Data_out[1]	OUT			V14	✓	14	LVCMOS33*	3.300	
Data_out[0]	OUT			U14	✓	14	LVCMOS33*	3.300	
RAddr[2]	IN			W16	✓	14	LVCMOS33*	3.300	
RAddr[1]	IN			V16	✓	14	LVCMOS33*	3.300	
RAddr[0]	IN			V17	✓	14	LVCMOS33*	3.300	
WAddr[2]	IN			R2	✓	34	LVCMOS33*	3.300	
WAddr[1]	IN			T1	✓	34	LVCMOS33*	3.300	
WAddr[0]	IN			U1	✓	34	LVCMOS33*	3.300	
an[3]	OUT			W4	✓	34	LVCMOS33*	3.300	
an[2]	OUT			V4	✓	34	LVCMOS33*	3.300	
an[1]	OUT			U4	✓	34	LVCMOS33*	3.300	
an[0]	OUT			U2	✓	34	LVCMOS33*	3.300	
seg[6]	OUT			U7	✓	34	LVCMOS33*	3.300	
seg[5]	OUT			V5	✓	34	LVCMOS33*	3.300	
seg[4]	OUT			U5	✓	34	LVCMOS33*	3.300	
seg[3]	OUT			V8	✓	34	LVCMOS33*	3.300	
seg[2]	OUT			U8	✓	34	LVCMOS33*	3.300	
seg[1]	OUT			W6	✓	34	LVCMOS33*	3.300	
seg[0]	OUT			W7	✓	34	LVCMOS33*	3.300	

I/O Ports in 'Schematic' (28)

12. Run the synthesis, implementation and bitstream generation (if you just perform the bitstream generation, you will be prompted to do the other steps required).

13. Connect the Basys3 to the computer using the USB cable, and turn on the Basys3 power switch.

14. Select “Open Hardware Manager” from the bottom of the left menu, and select “Open Target”, then “Auto Connect”. Once the computer has connected to the Basys3, you are ready to dump the configuration data into the FPGA. Select “Program Device”.

You are now running the register file! Verify functionality and pass off your register file when you are sure it is working correctly.