# **CprE 381, Computer Organization and**

# **Assembly-Level Programming**

# **Lab 2 Report**

Student Name \_\_\_Yin Quen Choong\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

***Submit a typeset pdf version of this on Canvas by the due date. Refer to the highlighted language in the lab document for the context of the following questions****.*

[Part 1 (a)] Draw the interface description (i.e., the “symbol” or high-level blackbox) for the MIPS register file. Which ports do you think are necessary, and how wide (in bits) do they need to be?

[Part 1 (b)] Create an N-bit register using this flip-flop as your basis.

Done

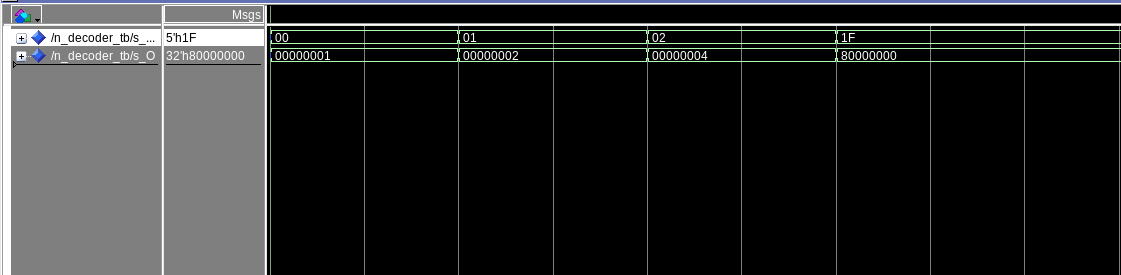
[Part 1 (c)] Waveform.



[Part 1 (d)] What type of decoder would be required by the MIPS register file and why?

5 bit decoder since there are 32 registers.

[Part 1 (e)] Waveform.

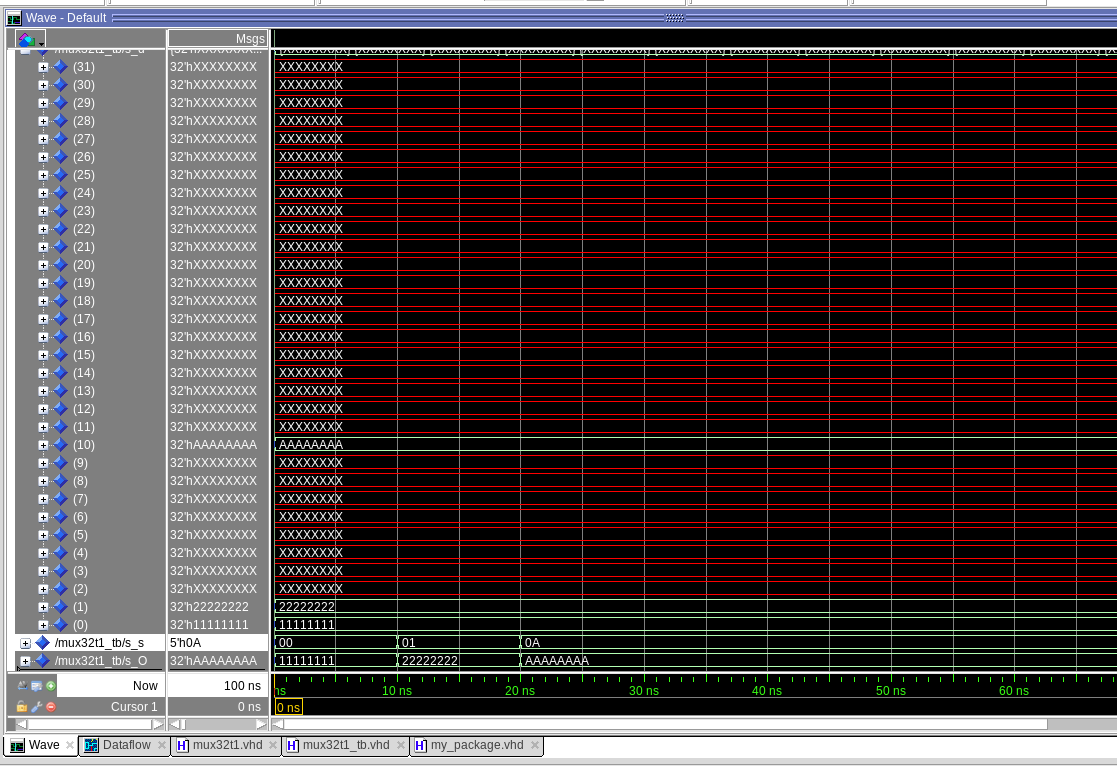


[Part 1 (f)] In your write-up, describe and defend the design you intend on implementing for the next part.

Well to implement this, we will use a data flow structure which will make our life so much easier. Additionally, a 2D array of std\_logic\_vector will be used too. It will be used as as a data input and a 5 bit select line will be use for the multiplexer.

[Part 1 (g)] Waveform.

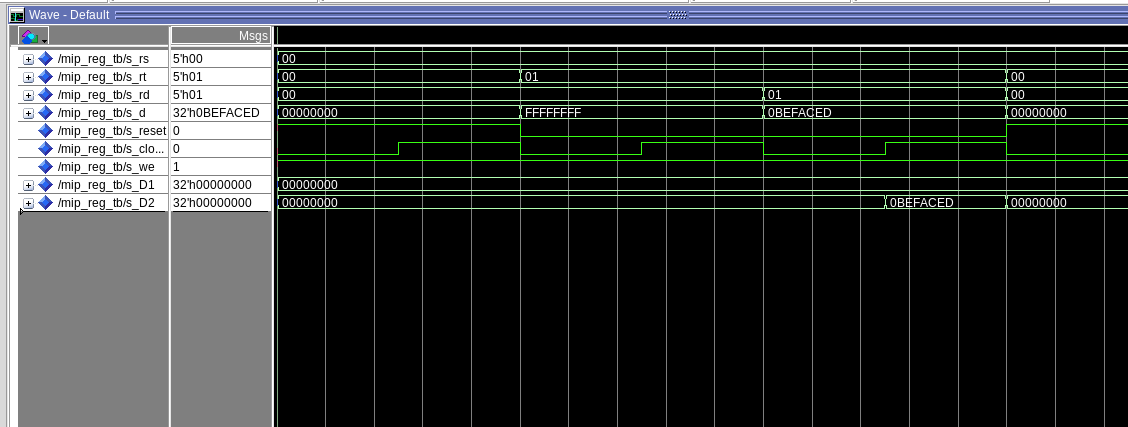
Only mux input 1,2,10 is loaded with data. When select line selected the lines it will produce the data.



[Part 1 (h)] Draw a (simplified) schematic (i.e., components within the high-level blackbox) for the MIPS register file, using the same top-level interface ports as in your solution describe above and using only the register, decoder, and mux VHDL components you have created.

[Part 1 (i)] Waveform.

When reg 0 will ignore all value being store. And reg 1 is store with 0xBEFACED and it shows BEFACE at D2 which means it is stored with the value

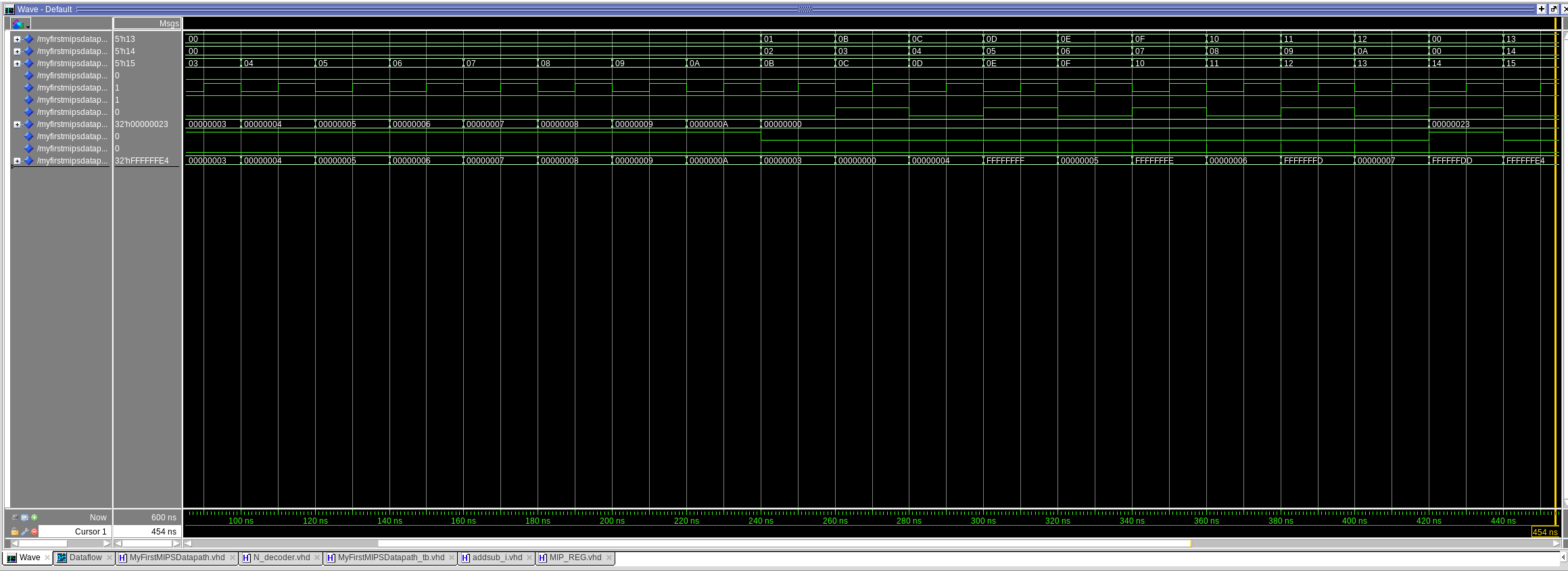


[Part 2 (b)] Draw a symbol for this MIPS-like datapath.

[Part 2 (c)] Draw a schematic of the simplified MIPS processor datapath consisting only of the component described in part (a) and the register file from problem (1).

[Part 2 (d)] Include in your report waveform screenshots that demonstrate your properly functioning design. Annotate what the final register file state should be.

The output are as expected output 3,0,4,-1,5,-2,6,-3,7,-35, -28



[Part 3 (a)] Read through the mem.vhd file, and based on your understanding of the VHDL implementation, provide a 2-3 sentence description of each of the individual ports (both generic and regular).

Data width is the number of bits in the number of bits contain in the memory data

address width is 10 bit which means there is 2^10 number possible address location.

Clk is the input clock and everything happen during the positive edge

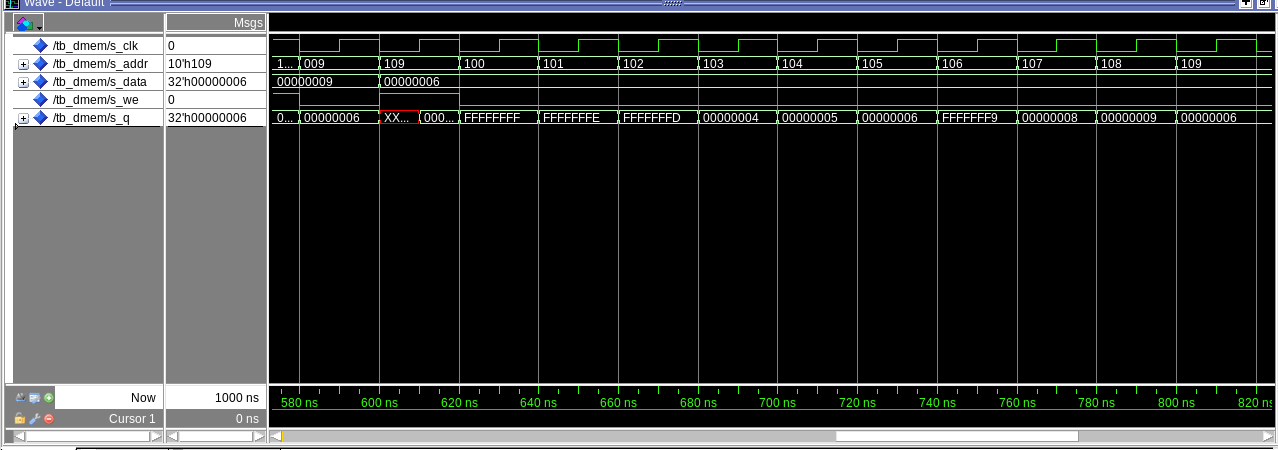
Addr is the address location

data is the input data given to store.

We is write enable to enable writing into the memory

q is to output the data

[Part 3 (c)] Waveforms.



[Part 4 (a)] What are the MIPS instructions that require some value to be sign extended? What are the MIPS instructions that require some value to be zero extended?

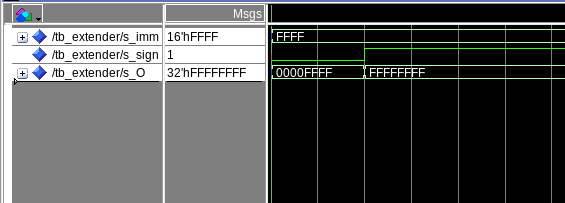
Mips instructions that would require a zero extended is immediate value for addi, ori, lui or any I type operation.

For sign extend will be require for anything that is using immediate value.

[Part 4 (b)] what are the different 16-bit to 32-bit “extender” components that would be required by a MIPS processor implementation?

The 2 main different of the 16 bit to 32 bit extender is padding the 16 bit input with 0 from bit 31 to bit 16 for positive number and padding it with 1 from bit 31 to bit 16 for all negative input

[Part 4 (d)] Waveform.



[Part 5 (a)] what control signals will need to be added to the simple processor from part 2? How do these control signals correspond to the ports on the mem.vhd component analyzed in part 3?

2 extra control signal need to be added which is the MEM2REG\_EN and SW\_EN for storing word to memory

[Part 5 (b)] Draw a schematic of a simplified MIPS processor consisting only of the base components used in part 2, the extender component described in part 4, and the data memory from part 3.

[Part 5 (c)] Waveform.

I loaded the memory with these values

00000001// Loads at address 0

00000002

00000003

00000004

00000005

00000006

00000007

00000008

00000009

0000000A // Loads at address 9

0000000B // Loads at address 10

0000000C // Loads at address 11

0000000D // Loads at address 12

0000000E // Loads at address 13

0000000F // Loads at address 14

00000001 // Loads at address 15

00000002

00000003

00000004

00000005

00000006 // Loads at address 20

00000007

00000008

00000009

0000000A // Loads at address 24

and my expected output for all addition should be 6,F,1b,1d, 23,2d in hex

1 + 5 = 6

6 + 9 = 15 == 0xF

0xF + 0xC = 0x1b

0x1b+ 0x2 = 0x1d

0x1D + 0x6 = 0x23

0x23 + 0xA = 0x2D

