תרגיל בית רטוב מספר 3 מערכות ספרתיות ומבנה מחשב

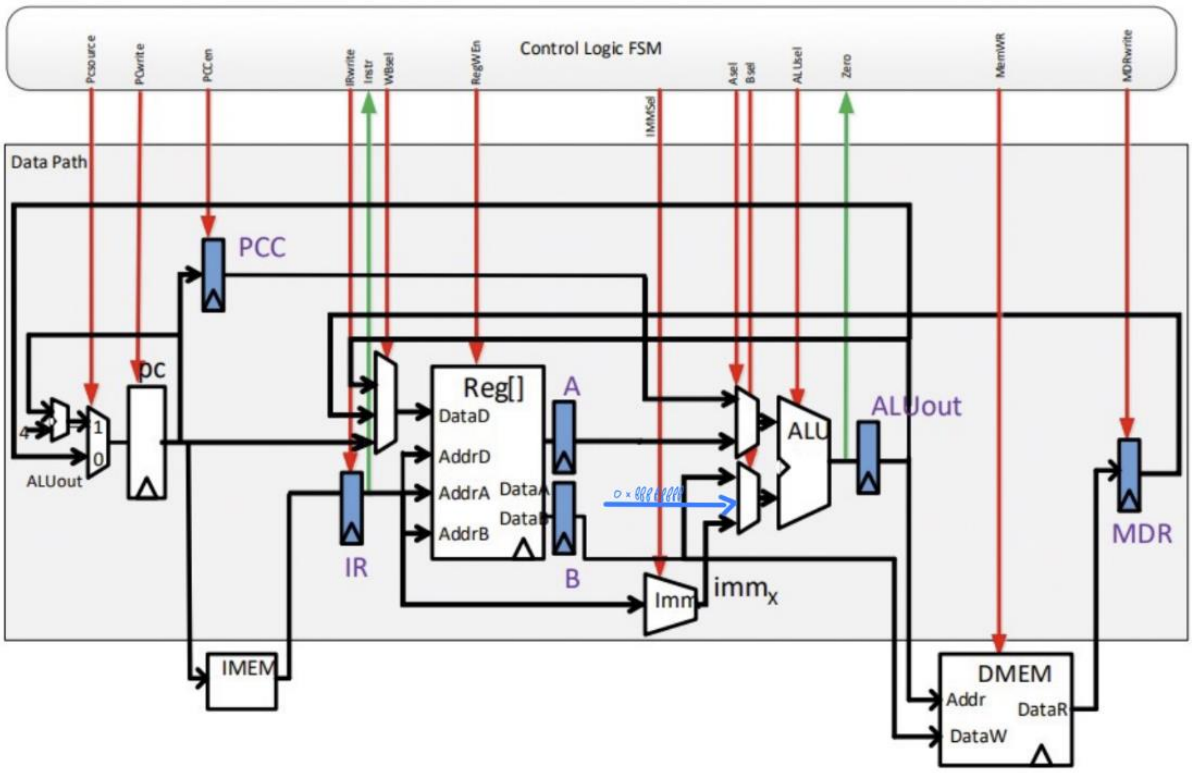
342816097	גיליאן בן שושן
330083858	ראובן טימסיט

Dry part

2.1

We want to add the connection command to the model. In this section we want to add a command to the processor so that the command : ‘addi’ , ‘rs1’ , ‘rd’ , ‘imm’, will perform the action : (R[rs1]+imm) XOR 0xffffffff.
And we will write the resultin the register ‘ rd ‘.

Changes in datapath :

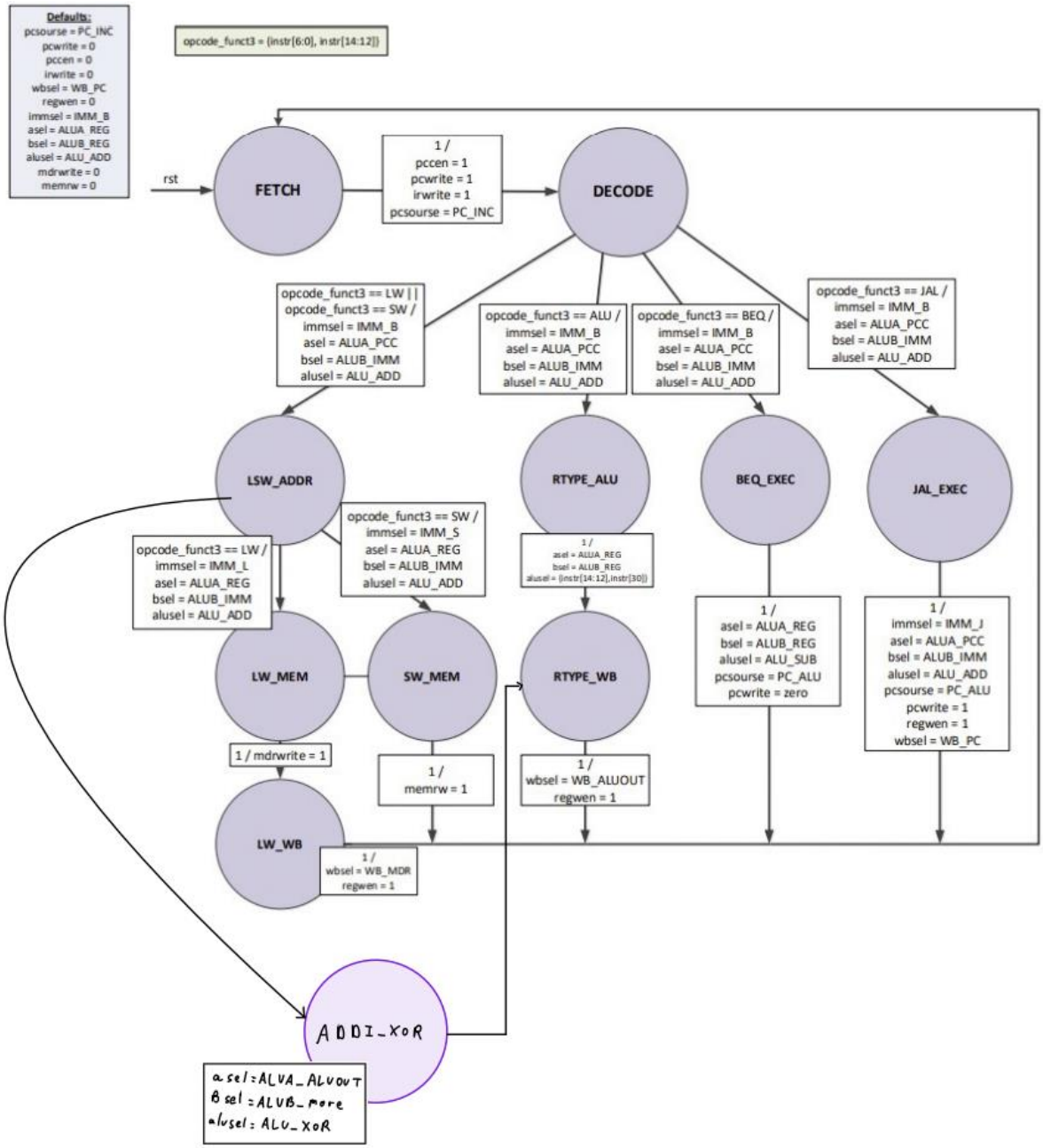


Explanation :

We have to use twice of ALU :

- The first time to do the regular addition between what we received in the first register and the value of ‘ immediate ‘ that we received.
- And we need to do the command XOR between this result and the constant : 0xffffffff.
  So we have to expand the MUX of A_sel and the mux of B_sel to add in the mux of A_sel the result of ALU_OUT of the previous calculation.
  To the end we have to had in the mux of B_sel the constant 0xffffffff.

Changes in the state machine :



Explanation :

We only need to add one state called: ADDI_XOR.

The states DECODE and FETCH are equals for all actions, so stay here.

After that we will pass to the state : LSW_ADDR. Until now we used this mode only for the commands: SW and LW to calculate the destination address to/from which should be written/loaded using a connection of the ‘ immediate’ that we received with the register. The signals remain the same because you connect a register with a constant through a connection operation and this is the signals that are already registered in the state, so you don't need to change nothing, it remains the same.

Now action XOR is required between the result of the connection and the constant 0xffffffff. For this action we have to create a new state called: ADDI_XOR. We are doing XOR between the previous result (so asel = ALUA_ALUOUT) and the constant that we add in the mux (so bsel = ALUB_MORE)
In addition the ALU_SEL will be on the ALU_XOR because we are doing command XOR.

Now, the last thing to do is to write the result from the XOR operation to the register ‘rd’ and this is basically what we are doing in WB_RTYPE.

And from here we will go as usual to the Fetch mode which indicates that we have finished the command.

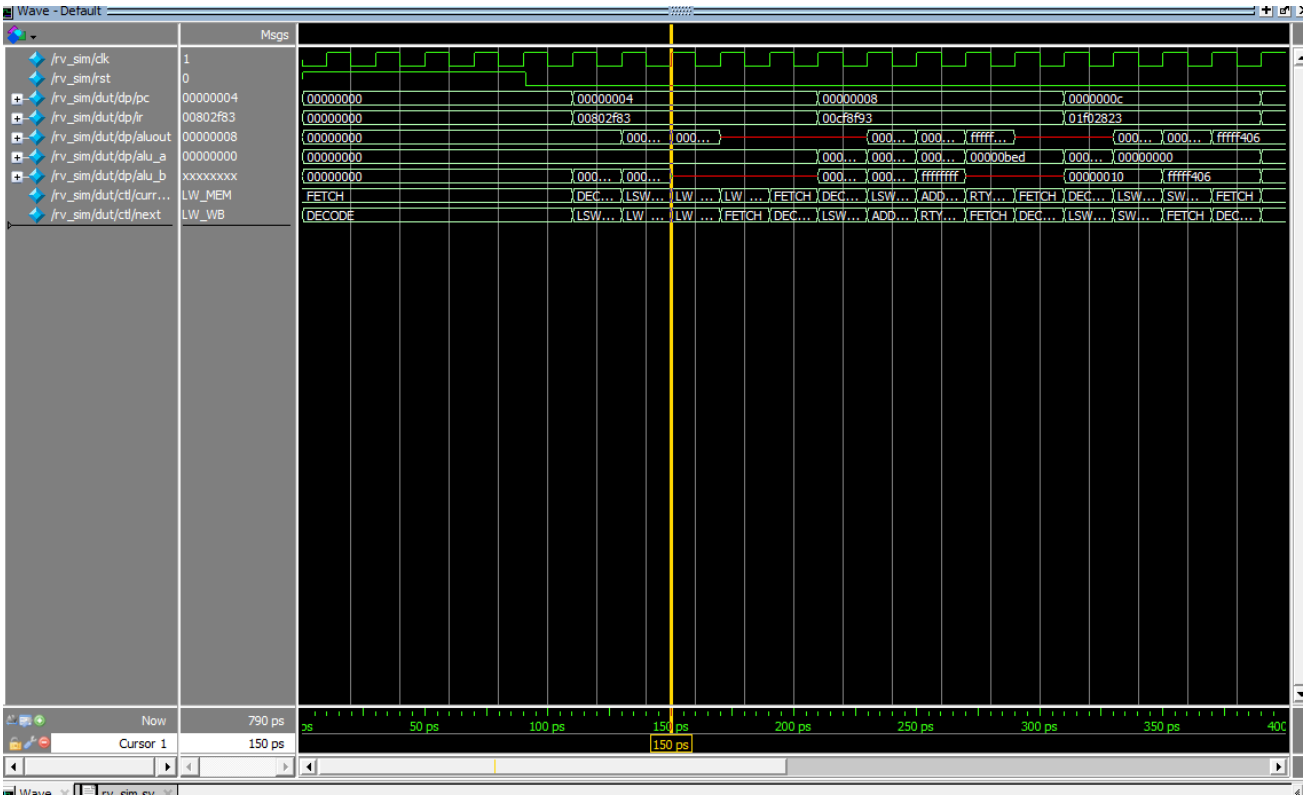
Wet part

2.2

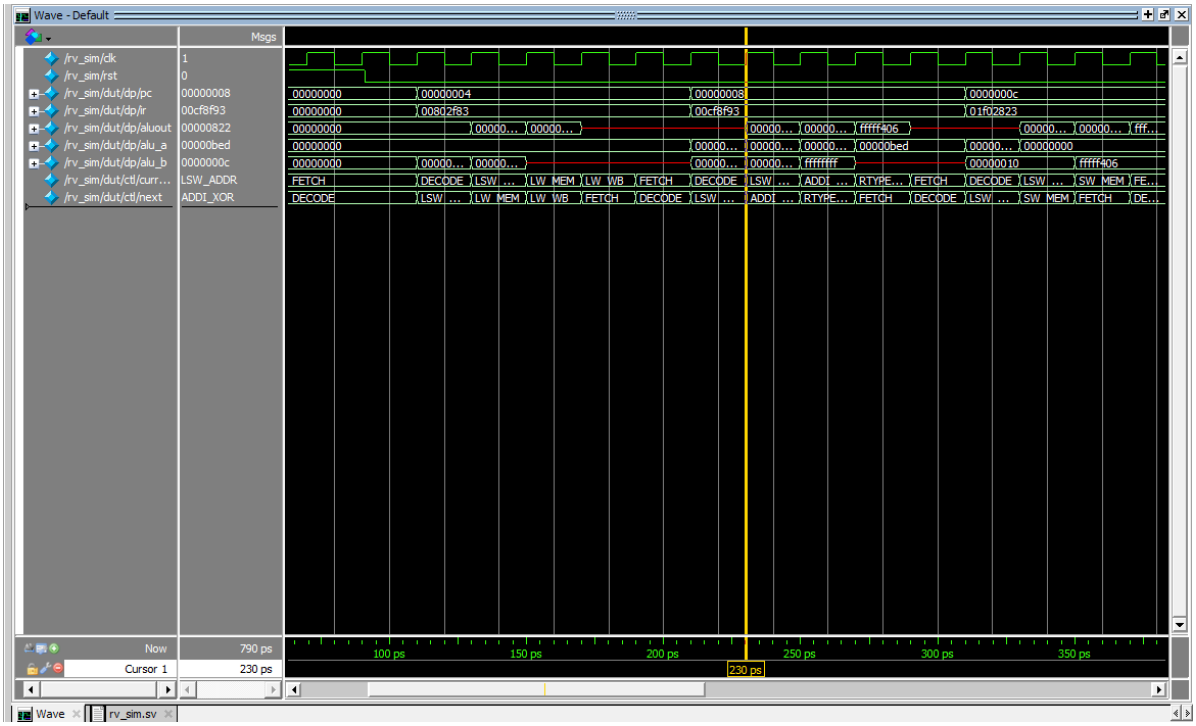
See files params.inc, rv_ctl.sv, rv_dp.sv

2.3

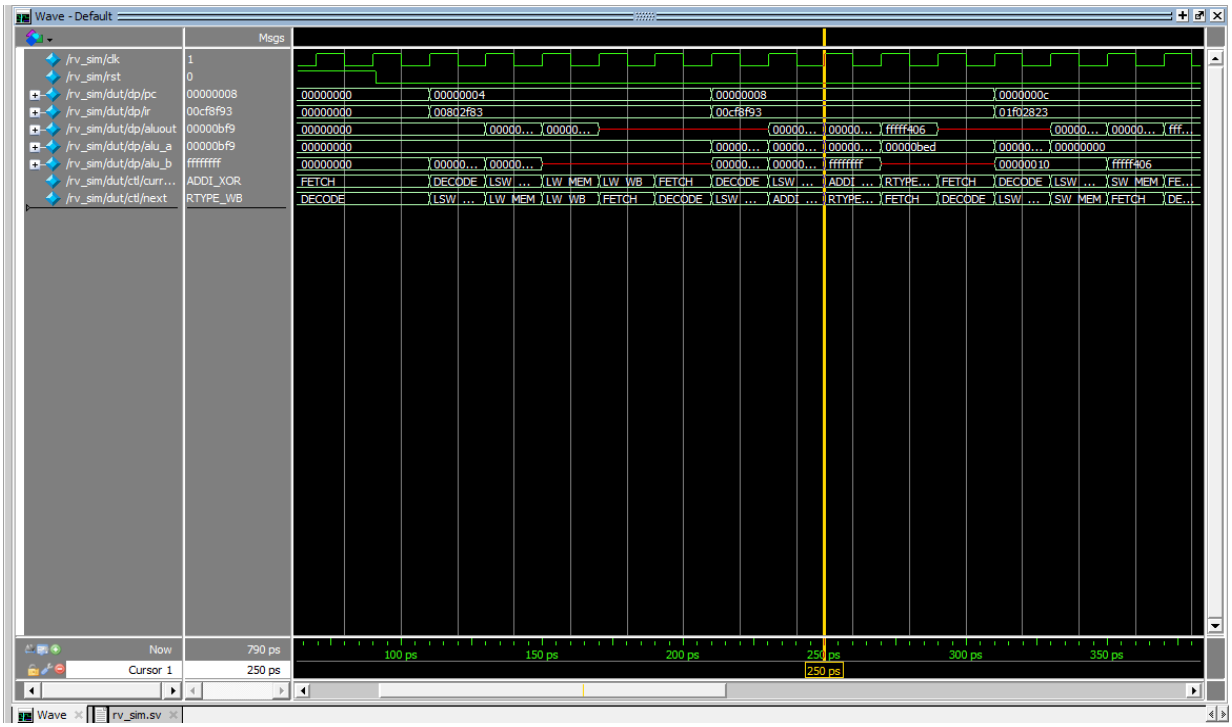
The first instruction in LW it asks from the program to take the value of the address 8 and put it in t6. We ca see that aluout takes the value 8 as expected



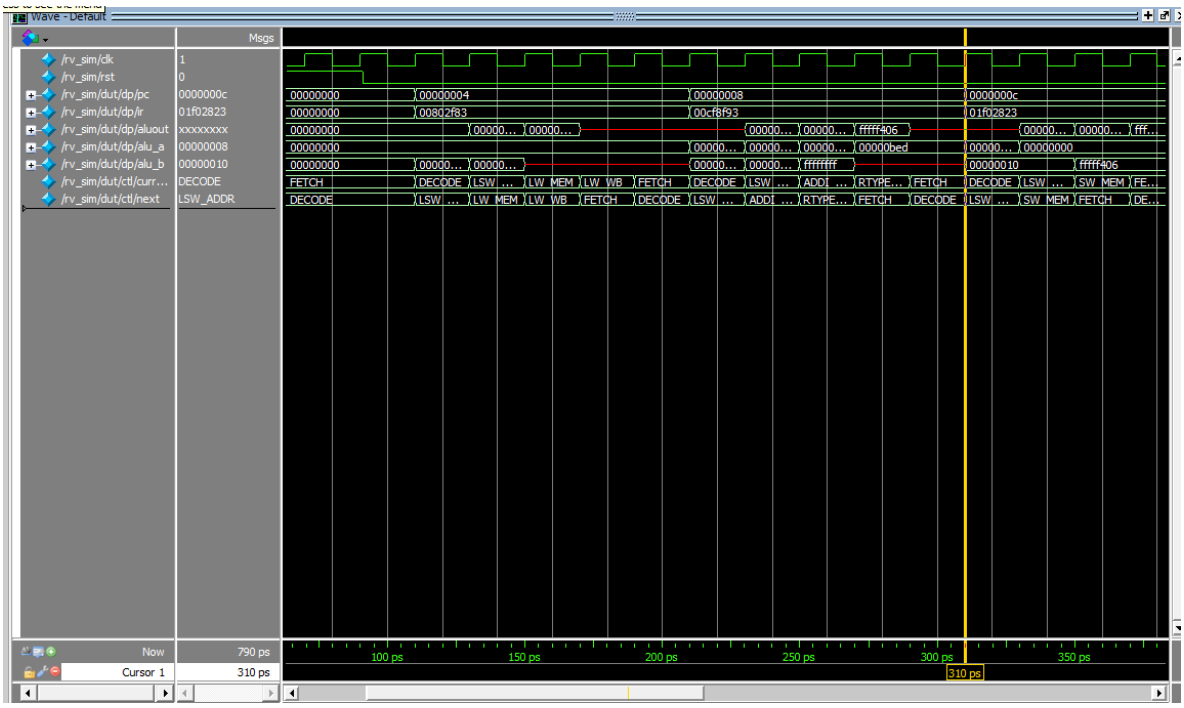
Now we can see that when we want to do the ADDI instruction we add 12 (C) to the number bed and we can see here that alua is bed and alu b is C then we receive the result in aluout that is bf9



Then we're doing the XOR operation between alua (BF9) and alub (FFFFFFFF) and the result Is aluout(FFFFF406)



After that we are doing the instruction SW that take the value of the address 16(10) and we can see that after this step the value is save in the memory



We will now check that the value obtain line 5 in dmem_out is the same as expected

We first take the number in address 8 in dmem_init that is 00000bed

Then we're doing the new operation addi so first we re addind 12(c)

00000bed+ 0000000C=00000bf9

And then we are doing the xor betweenthat result and ffffffff

So $00000bf9 \oplus ffffffff = fffff406$

And we ca see that the 5th line in dmem_out is fffff406