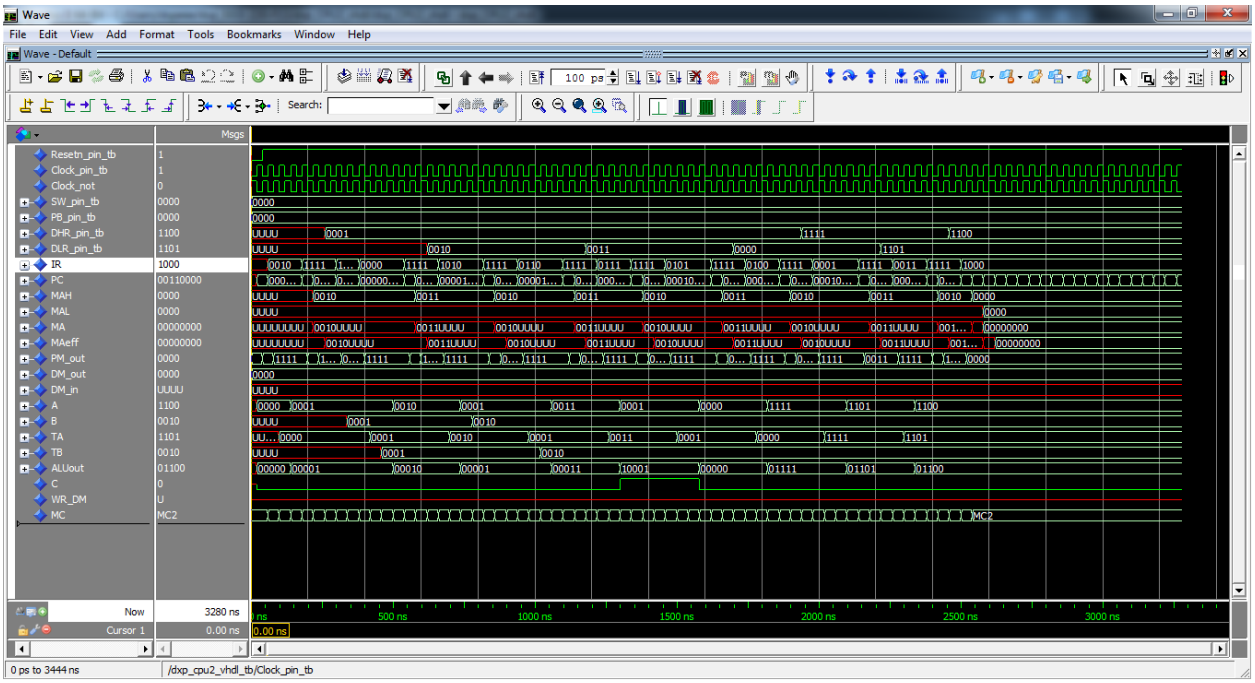
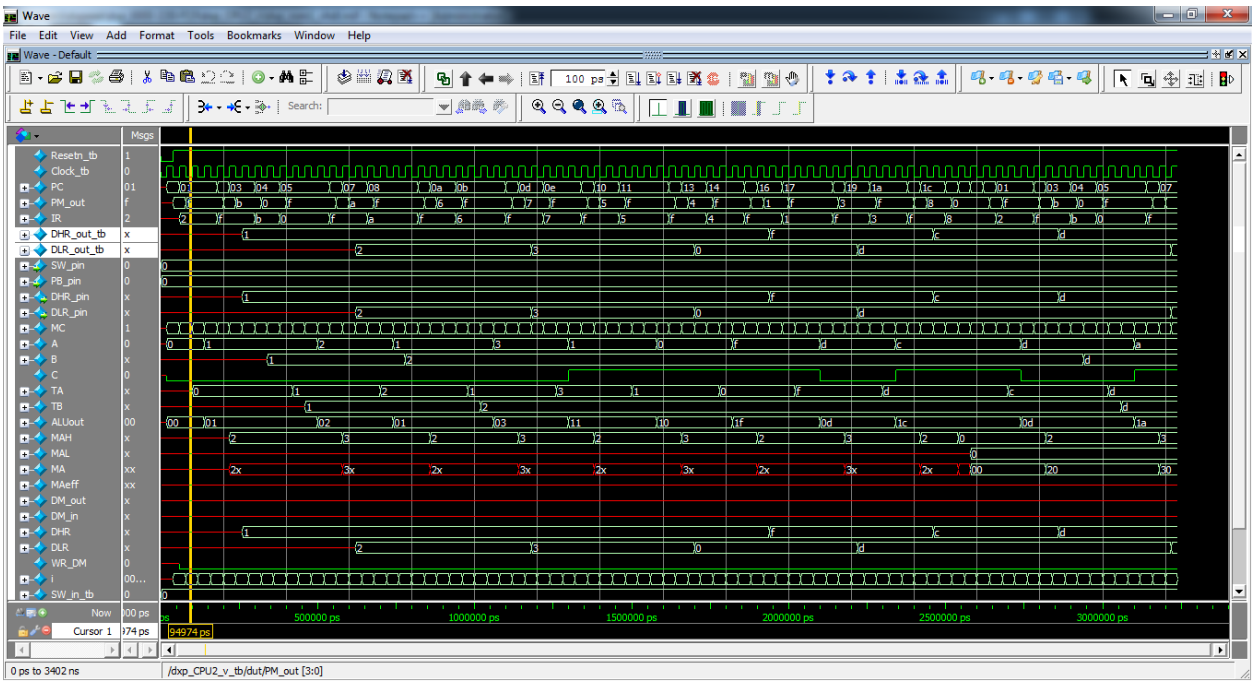
EEEE-220-Lab12

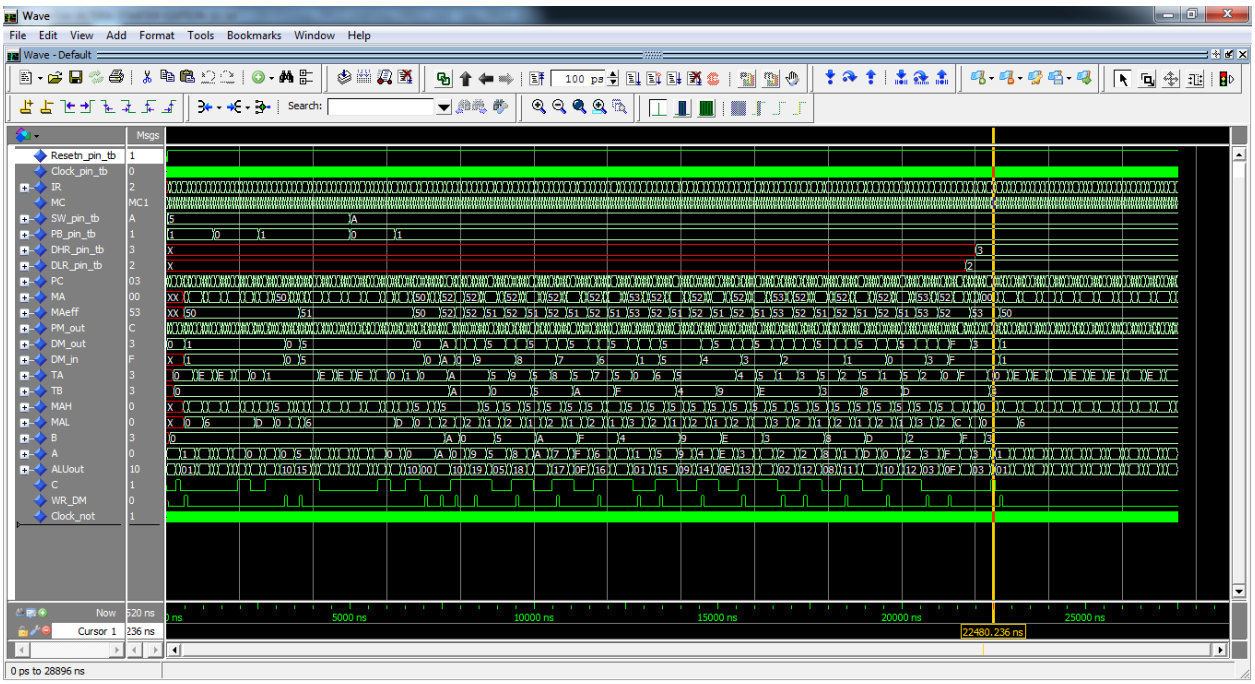
1) From mycourses download on your desktop this pdf file. This lab consists of four parts, which are due as described at the end of this document.
2) **Objective**: The objective of this lab exercise is to capture and verify the functionality of your fmlRISC Central Processing Unit 2 (CPU2) design. You can use either VHDL or Verilog.
3) Your fmlRISC CPU2 is described fully behaviorally as a large FSM, except for the access to memories.
4) The behavioral descriptions use a combination of ***case*** and ***if-then-else*** statements, as can be seen in the files dxp_CPU2_vhdl.vhd and dxp_CPU2_v.v.
5) You need to modify the code so that it matches your design specifications.
6) Comments about individual design specifications:
    a. Bit6 – valid.
    b. Bit5 – valid.
    c. Bit4 – disregard; it is not possible to model at this abstraction model.
    d. Bit3 – cannot be modeled directly; **however, you are required to follow and implement your fmlRISC1 asm chart cycle accurate, i.e. machine cycle by machine cycle!**
    e. Bit2 – disregard; it is not possible to model at this abstraction model.
    f. Bit1 – valid.
    g. Bit0 – valid.
7) You can or should be able to re-use your fmlRISC CPU1 user program files (*.mif) and testbenches without modifications.
8) If you don't have working testbenches for CPU1, use the ones provided in the project directories as templates.
9) **Part 1: fmlRISC – CPU2 – capture and verification with test program 1 (TP1).**
10) Make sure you have all DP and CU related lecture handouts and notes.
11) Download and un-archive the project corresponding to your language choice: dxp_CPU2_vhdl or dxp_CPU2_v.
12) Compile and simulate TP1 using the dxp_CPU2_*_tb.* testbench.
13) Now, modify the CPU2 and testbench code to fit your specifications.
14) As you can see, CPU2 has only the absolute necessary input and output ports:
    a. Inputs: Resetn, Clock, SW_in, PB_in,
    b. Outputs: DHR_out, DLR_out.
15) To see any of the internal signals, once ModelSim is running, select your uut or dut. In the Objects tab you'll see listed internal signals and constants (parameters). Select all former, right-click, and select to ADD WAVE. This will add these signals to the saved waveform files during the next simulation.

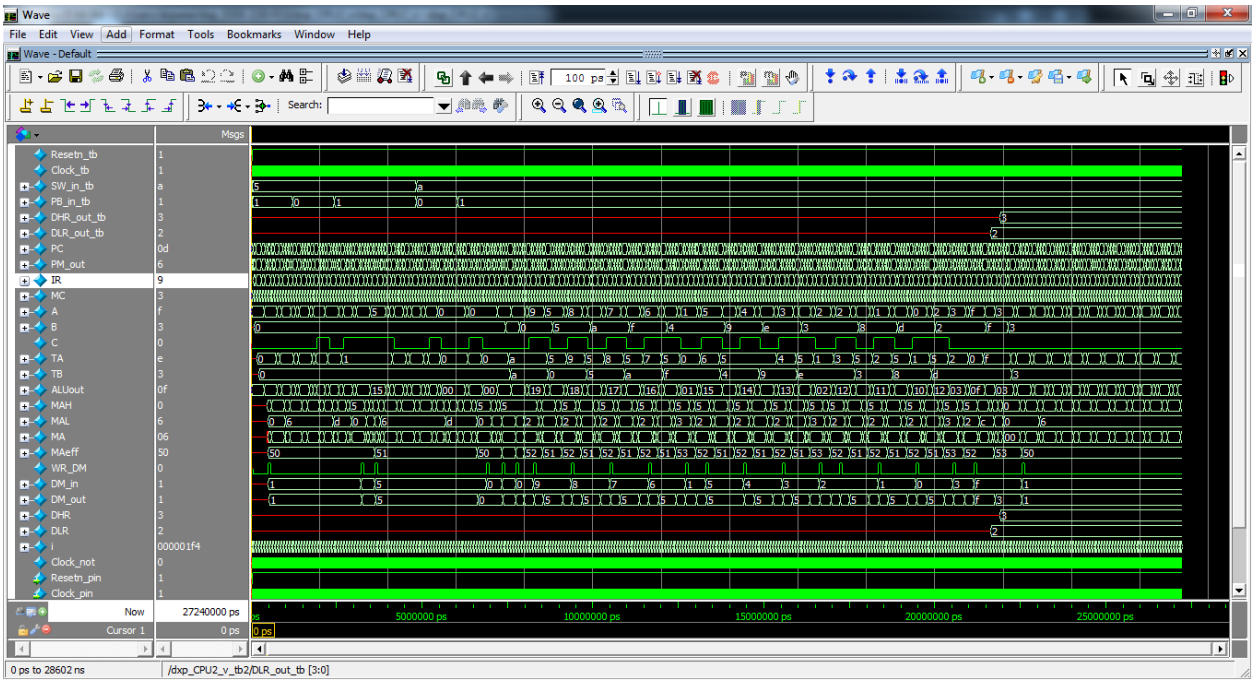
16) To repeat the simulation, go to Simulate > Restart, and then Simulate > Run All. Now you'll see all input and output ports, as well as all internal signals you have selected. The latter will help you in the debugging process.
17) If you make changes to your code, you'll need to re-compile. To do this without the need to close ModelSim, and implicitly loose the waveform settings you have made thus far, click on the Library tab.
18) Expand the work library, right-click and update. Next, right-click and recompile.
19) Now re-start and re-simulate as described in step 16.
20) Hint: it is very useful to group associated signals together, as you can see in the captures below.
21) Once the simulation/verification is successful, you should prepare and emulate it on the board, as described in Lab11.
22) Below are simulation waveform windows for the VHDL and Verilog implementations. Once it simulates correctly, your timing diagram will look as shown below.





23) **<u>Part 2</u>: fmlRISC – CPU2 – verification with test program 2 (TP2).**
24) Hereon it is assumed that you have successfully completed part 1.
25) Compile and simulate TP2 using the dxp_CPU2_*_tb2.* testbench. Modify the latter to match your port names.
26) For debugging purposes, follow the recommendation described in steps 15 – 20.
27) Once the simulation/verification is successful, you should prepare and emulate it on the board, as described in Lab11.
28) Below are simulation waveform windows for the VHDL and Verilog implementations. Once it simulates correctly, your timing diagram will look as shown below.





29) **Part 3: fmlRISC – CPU – verification with test program 3 (TP3).**

30) This is the same program as TP2, except that it uses exclusively the indexed/PC_relative addressing mode to store the Read_Count_Variable in RAM.

31) The two templates are provided in the files dxp_rom3_Indexed_vhdl.mif and dxp_rom3_PCrel_vhdl.mif.

32) You can re-use the same testbench as for TP2.

33) Once the simulation/verification is successful, you should prepare and emulate it on the board, as described in Lab11.
34) **Part 4: fmlRISC – CPU1 versus CPU2 comparison.**
35) Using information provided in the Summary of the Analysis & Synthesis (A&S) and Fitter (Fit) steps, create and include in your final report a table with the following header:

	CPU1		CPU2	
	A&S	Fit	A&S	Fit
Total Logic Elements				
Total Combinational Functions				
Dedicated Logic Registers				
Total Memory Bits				

36) Comment in your report on these resource utilization results.
37) **Grading – as per the syllabus and lab policy**:
38) **Due to the delay in posting the lab handout and associated project template files, this lab is due for full points on Wednesday, December 11, 8 – 11 am. Only demos during this time. No help possible. ~4 min. per student. If your 2nd and 3rd programs work, you don't need to show TP1. If you cannot make this time, you can schedule another time on Tuesday, December 10 - TBD.**
    a. 20 points for part1 or TP0 – board demo. If it works in simulation, there's no reason for it not to work on the board. Nonetheless, the simulation alone is only 10 points.
    b. 30 points for part2 or TP2 – board demo. The simulation alone is only 15 points.
    c. 20 points for part3 or TP3 – board demo. The simulation alone is only 10 points.
    d. 10 points for part4.
39) The instructor will grade labs 11 & 12. However, I'm sure that your TAs will enjoy seeing what you have accomplished.
40) The grading for lab 11 remains unchanged.
41) There won't be an overdue period for lab12! No pints of any kind will be given after Wednesday, December 11.
42) These two labs require a single report. You can find the details in the lab policy. This combined report is now due Wednesday, December 11, at 6pm.

43) Even if you don’t complete your entire design before the due date and time, you should still upload whatever you have up to that point on mycourses!
44) This concludes this week and course lab.

Congratulations!

Think back were you have started and how far you have gotten.

From a 2-to-1 multiplexer to a full processor!!!