DUE: Wednesday, 11/12 at 11:59 PM
Absolutely no late assignments accepted

Introduction:

You must work in pairs on this assignment and submit only one electronic copy per team. Make sure that each file contains the names of all team members. Since team work is important throughout the semester, you are expected to spend time reflecting on your team's strategies to deal with problems and work effectively. Eventually, you will have to assess in writing your own and your partner's performance. It is thus to your advantage, for this and every subsequent assignment, to take a deliberate approach to all aspects of your team's work and the skills that each one of you is bringing to the table, including responsibility and commitment, quality of work, timeliness of work, communication skills, helpfulness, and overall attitude.

For this assignment, you have to implement a slightly-modified SINGLEcycle implementation of the Larc computer (very similar to the one that we discussed in class). For full credit, each chip must work properly in all cases and the corresponding HDL file must be clearly organized.

Preparation:

1. Study your notes on the SINGLEcycle Larc implementation.
2. Download a4.zip from d2l.
3. Extract the files in this compressed file.
4. Change to the newly created a4 directory, in which you will do all of your work, namely complete the two files Larc.hdl and CU.hdl. In addition to the files needed by the simulator, the a4 .zip file contains four test files that you should load into the simulator and trace step by step. I will test your submission on these and additional test programs.
5. Copy your file RegisterFile.hdl from assignment 3 into the a4 directory and change the names of the registers in it to RegisterR0, RegisterR1, . . ., RegisterR15 so that you take advantage of these built-in registers and their associated widgets in the simulator window.
6. Do NOT copy your other memory chips from the previous assignment since you want to use the built-in ones and their associated widgets in the simulator window.
7. Copy your ALU.hdl and ALU1bit.hdl files from assignment 2 into the a4 directory.
8. Finally, you will also need to copy a small number of multiplexers and other 16-way chips that you implemented in assignment 2. Most of the chips you need are built into the simulator code and you should NOT copy them into the a4 directory, even if you re-implemented them in a previous assignment. In fact, the best way to proceed is to use the chips you need and only copy them into the a4 directory if the simulator displays an error message about not finding the chip.

Make sure to test each new chip fully. It is your responsibility to make sure that each chip works in all cases according to its intended function, as described in class and subject to the modifications specified below, even if the provided test files do not cover all possible cases. **For full credit, your Larc chip must follow exactly the schematic included at the end of this document and your other chips must use the smallest number of gates and chips needed to meet all of the requirements of the chip's API.** For example, if one of the components in the implementation of a chip behaves like a 4-way 16-bit-wide multiplexer, you must use a single chip (namely, Mux4Way16) for this component, instead of re-implementing it with basic gates.

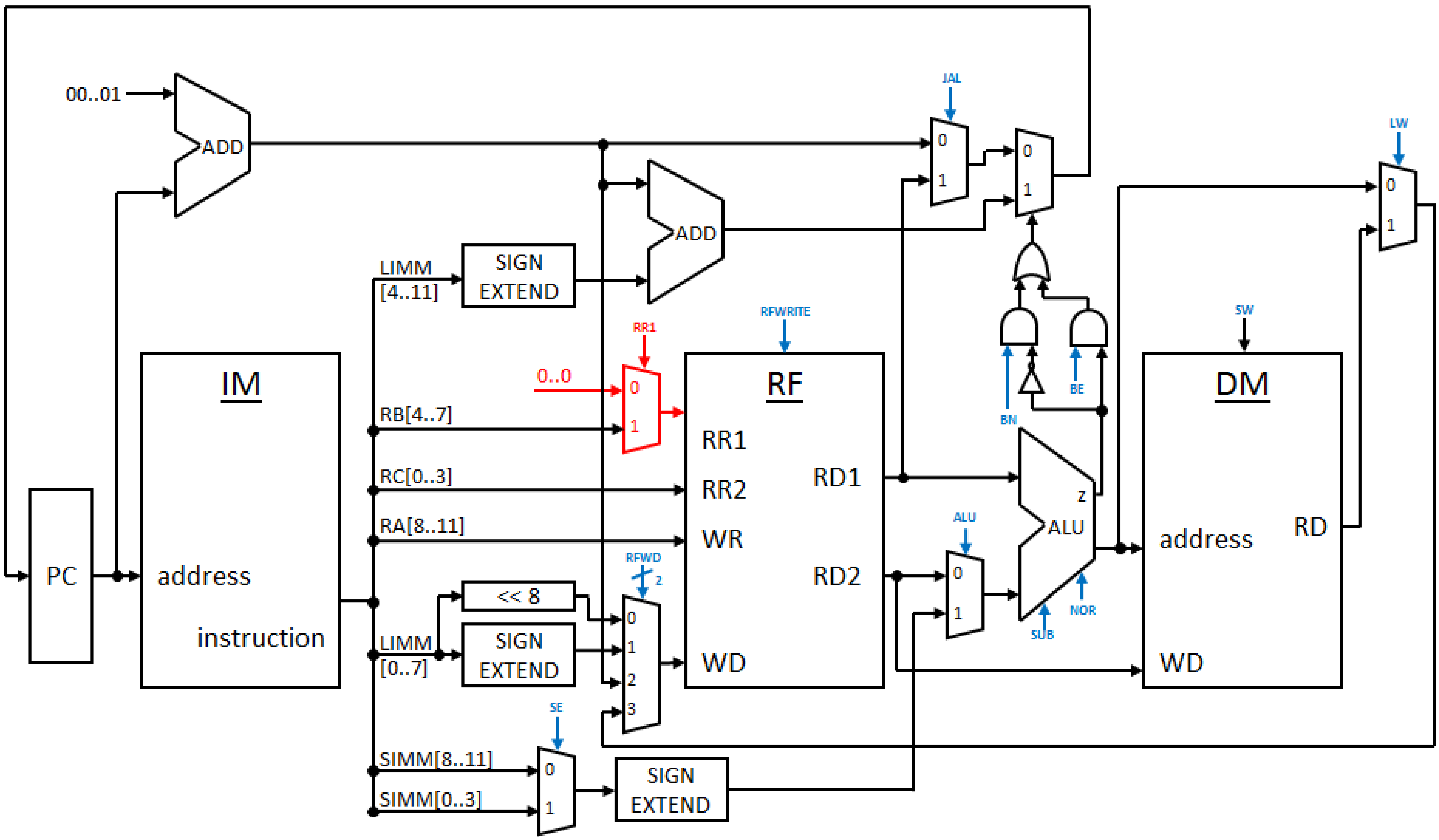
Here is a list of the chips that you must implement, some built-in chips that you must use and the other digital circuits you must implement for this assignment:

1. Larc.hdl: Main file to be completed to meet all of the requirements described in the notes from class. Furthermore:
    a. This chip must include a digital circuit with the smallest number of chips that enforces the constraint that **the first register (i.e., register 0) always contain the value 0, even if the user program attempts to modify this value**. For full credit, your implementation file must contain a short but precise comment that explains how you designed this circuit.
    b. This chip, in combination with the following one, must contain a digital circuit that implements the HALT instruction with opcode 1111.
2. CU.hdl: The other chip to be implemented. A table of all the control signals that must be sent out by this chip is specified at the end of this document. This is slightly different than what we discussed in class.
3. You also need to implement one additional multiplexer chip that is needed in a couple of places in our Larc design.
4. RegisterFile.hdl: Your chip from assignment 3, with the register chips renamed (NO other changes are allowed in the register file).
5. PC: A built-in chip you must use for the Program Counter register. This chip is associated with a GUI widget that displays the value stored in each memory location.
6. RAM16K: A built-in chip you must use for the data memory (DM). This chip is associated with a GUI widget that displays the value stored in each memory location.

Submission

When you have implemented and fully tested all of the above chips, follow this submission procedure:

1. Make sure both of your names appear in the comments of every file you submit.
2. Create a new folder called a4.
3. Copy all of your .hdl files needed to complete this assignment into the directory “a4”. This directory should contain the two .hdl files I provided, and one.hdl file for each non-built-in chip needed by at least one submitted chip.
4. Zip up your “a4” directory. Submit a SINGLE copy of your work. Upload your .ZIP file to the a4 dropbox on d2l.
5. For full credit, the “a4” directory submitted as a .ZIP file must contain only the files specified in step 3 above. Make sure to delete all extra files before submitting your a4 .zip file. Furthermore, each .hdl file must load AS IS in the simulator and pass each and every one of my tests. If I have to copy extra files that you did not submit, you will lose points.



PC
00..01
ADD
IM
address
instruction
LIMM [4..11]
SIGN EXTEND
ADD
RB[4..7]
RC[0..3]
RA[8..11]
0..0
RR1
RFWRITE
RF
RR1
RR2
WR
WD
RD1
RD2
<< 8
LIMM [0..7]
SIGN EXTEND
RFWD
SIMM[8..11]
SIMM[0..3]
SE
SIGN EXTEND
JAL
ALU
ALU
Z
SUB
NOR
BN
BE
SW
DM
address
WD
RD
LW

Here is a list of all the signals that should be sent by the control unit.

HALT	RR1	SE	RFWD		RFWRITE	ALU	JAL	BE	BN	SUB	NOR	SW	LW
0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1	0 / 1