CE/CZ 3001: Lab Project

For the rest of the lab work after Lab-3 you are required to do a project. The project consists of 3 parts and a bonus assignment. You are required to do coding and synthesis, and to demonstrate each part of the project. Write a project report to briefly describe the working of the design of each part. Report should also include the timing report and the waveform generated by simulating the testbench of each part of the project. For Part-3 and bonus assignment you will be required to find the minimum execution time and the reduction CPI which you achieve for the given program.

Project Part-1: Modify the 4-stage pipelined processor of Lab-3 to include BEQ, LW, and SW instructions and convert that to a 5-stage pipelined processor. **(6 marks)**

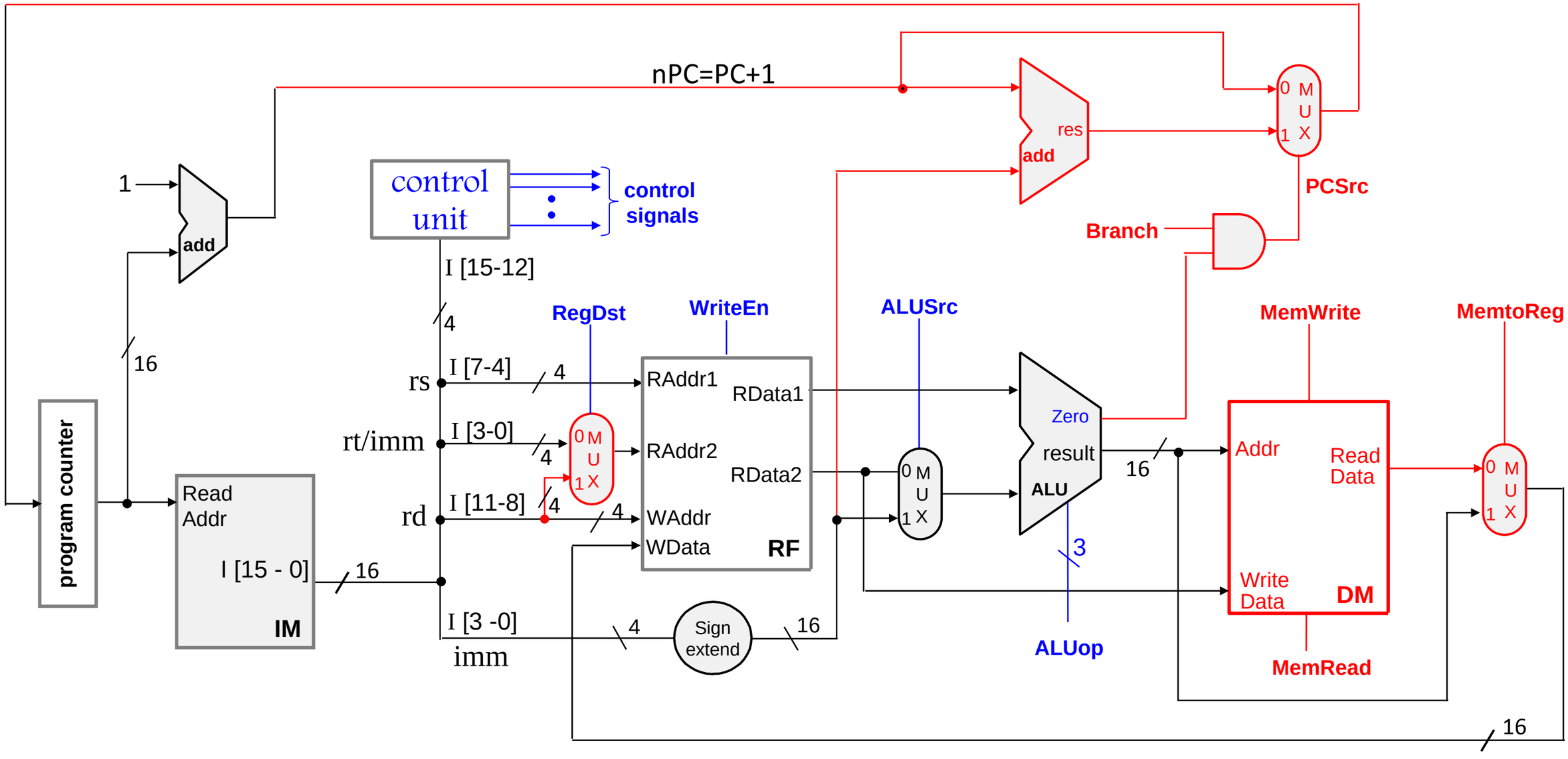
Project Part-2: Modify the processor designed in Part-1 of the project to include jump register (jr), jump (J), and jump & link (jal) instructions**. (5 marks)**

Project Part-3: Each group of students will be given a program which gets slowed due to pipeline stalls. You are required to modify the program to remove the hazards so as to reduce the number of pipeline stalls. Finally, you will estimate the reduction in the CPI and execution time which you achieve. **(4 marks)**

Bonus Assignment: You will be required to incorporate a direct-mapped cache for data memory (DM). *This is an optional assignment. You will not loose any mark if you cannot complete this. On Bonus Part of the project you will be awarded with maximum of 5 marks over and above what you have secured in this lab project such that the total marks secured in course work will not exceed 30 marks for any student.*

Report is to be submitted by 11th November 2015 before 5 pm at Hardware Projects Lab.

Project Part-1: Modify the processor of Lab-3 to **include BEQ, LW, and SW instructions**. You need to include the (i) Data Memory (DM) of size 64 KB (word size = 16 bits) which can be implemented in the same way as Instruction Memory (IM), and (ii) the branch implementation circuit. The components and connections to be used in this part are shown in red. Take note of control signals to include BEQ, LW, and SW. Modify the control unit accordingly. CONTINUED IN THE NEXT PAGE…



load word: lw $rd, imm ($rs), meaning : rd ← Mem[imm + $ rs]

store word: sw $rd, imm ($rs), meaning : Mem[imm + $ rs] ← rd

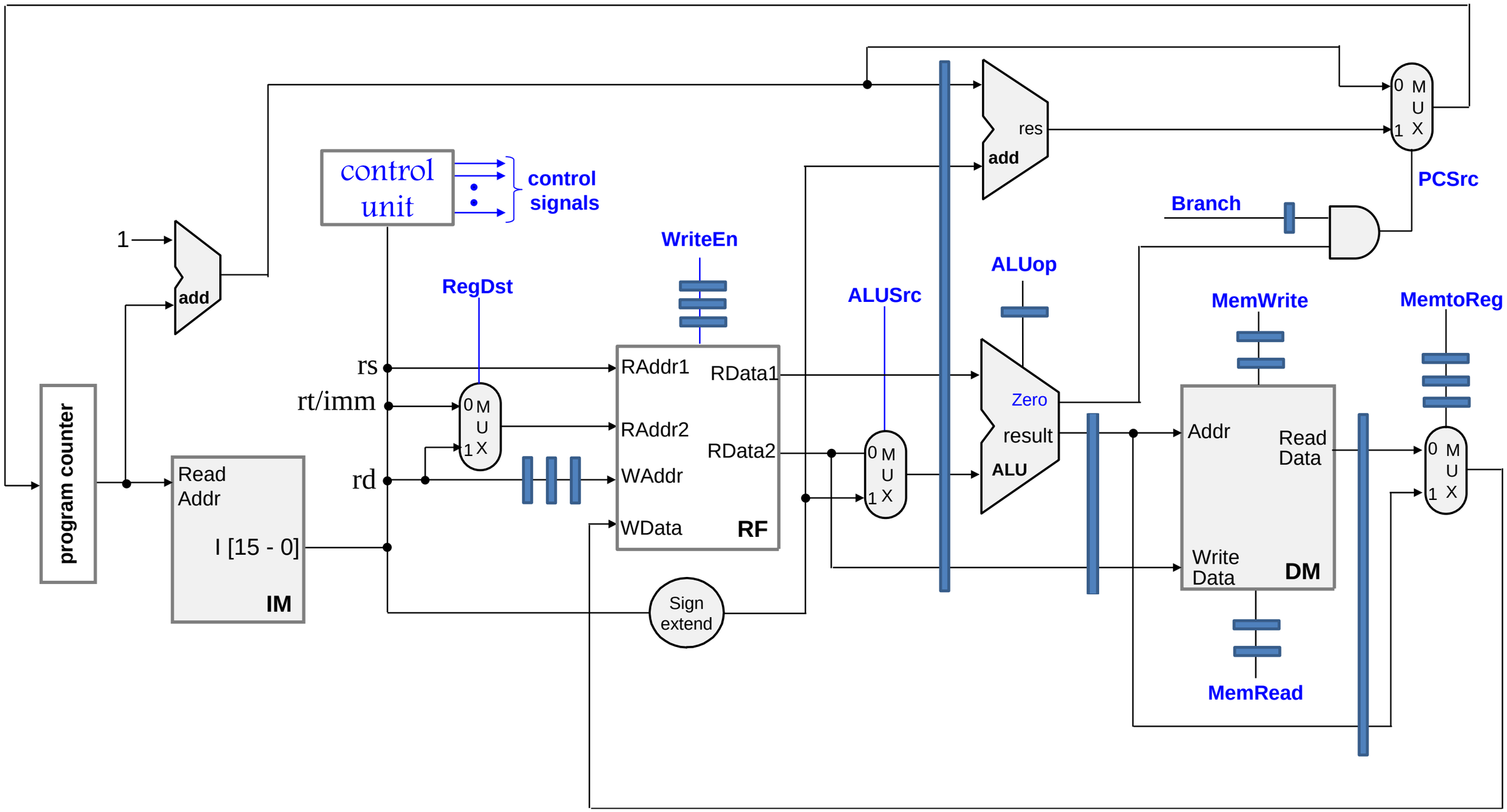
branch on equal: $rs; $rd, imm, meaning: PC← nPC+imm, if $rs = $rd

RegDST= 1 for SW and BEQ, and 0 for ALU and LW.
ALUSrc = 1 for LW and SW and 0 for ALU and BEQ

16-bit instruction word for the R- and I-type instruction

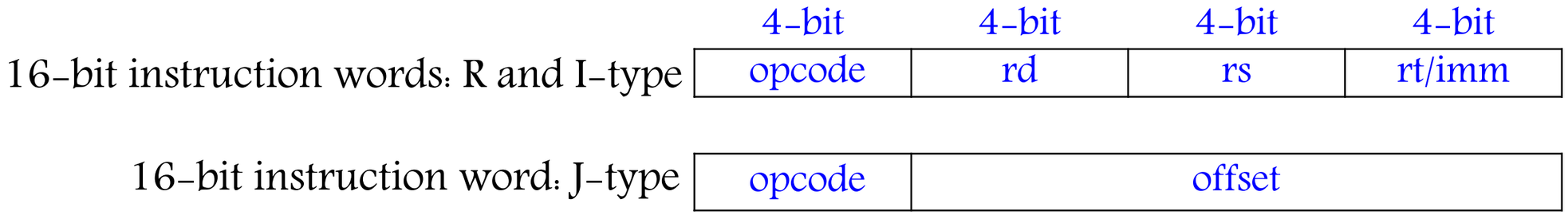
4-bit	4-bit	4-bit	4-bit
opcode	rd	rs	rt/imm

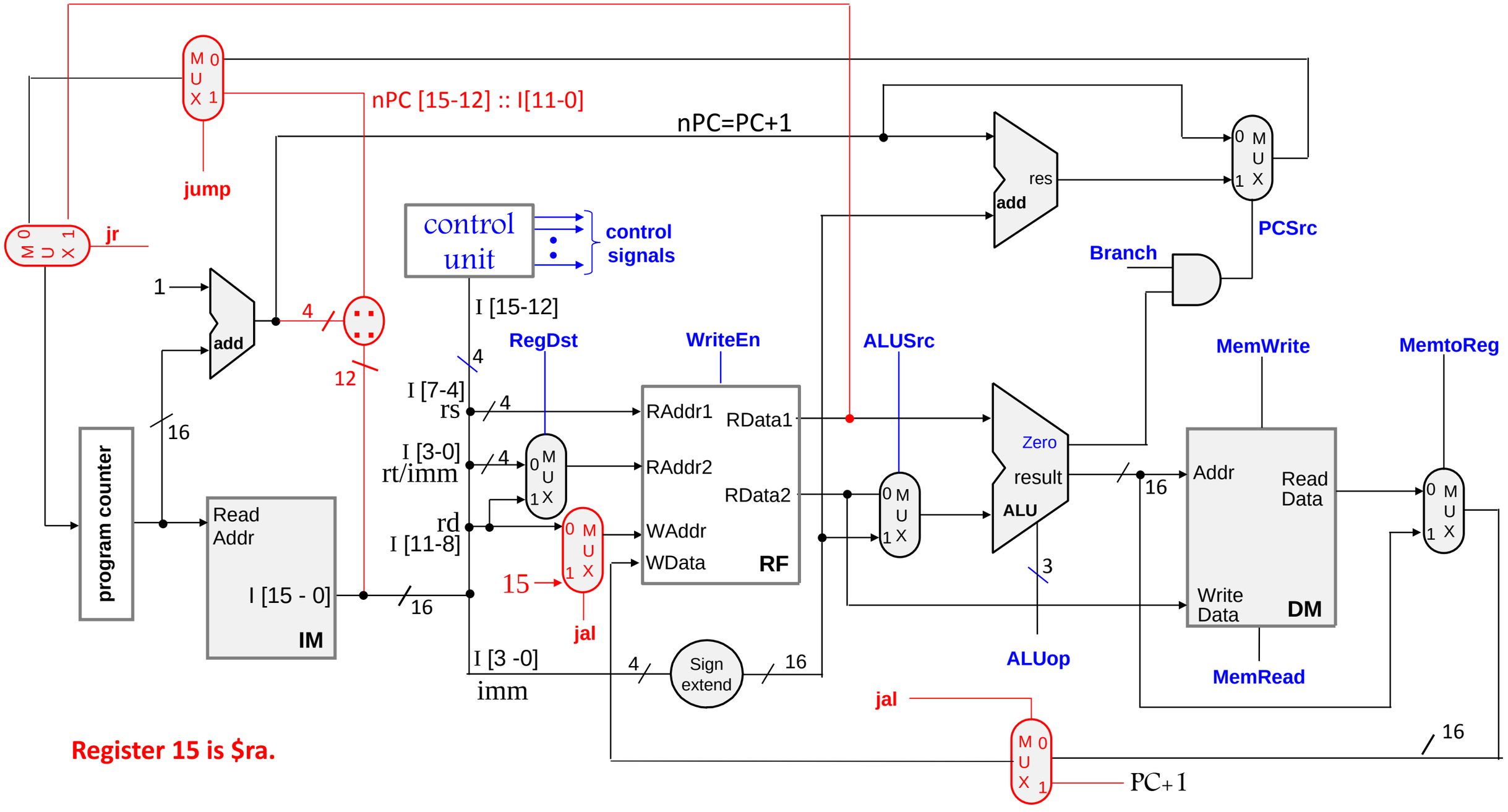
Convert the 4-stage pipelined processor to 5-stage pipelined processor. The pipeline registers are shown in the figure below. Note the 3 pipeline registers in the datapath at the ID/EXE, EXE/MEM, and MEM/WB interfaces. Control signals are generated in the first stage and therefore delayed by the number of clock cycles depending on the pipeline stage in which they are used.



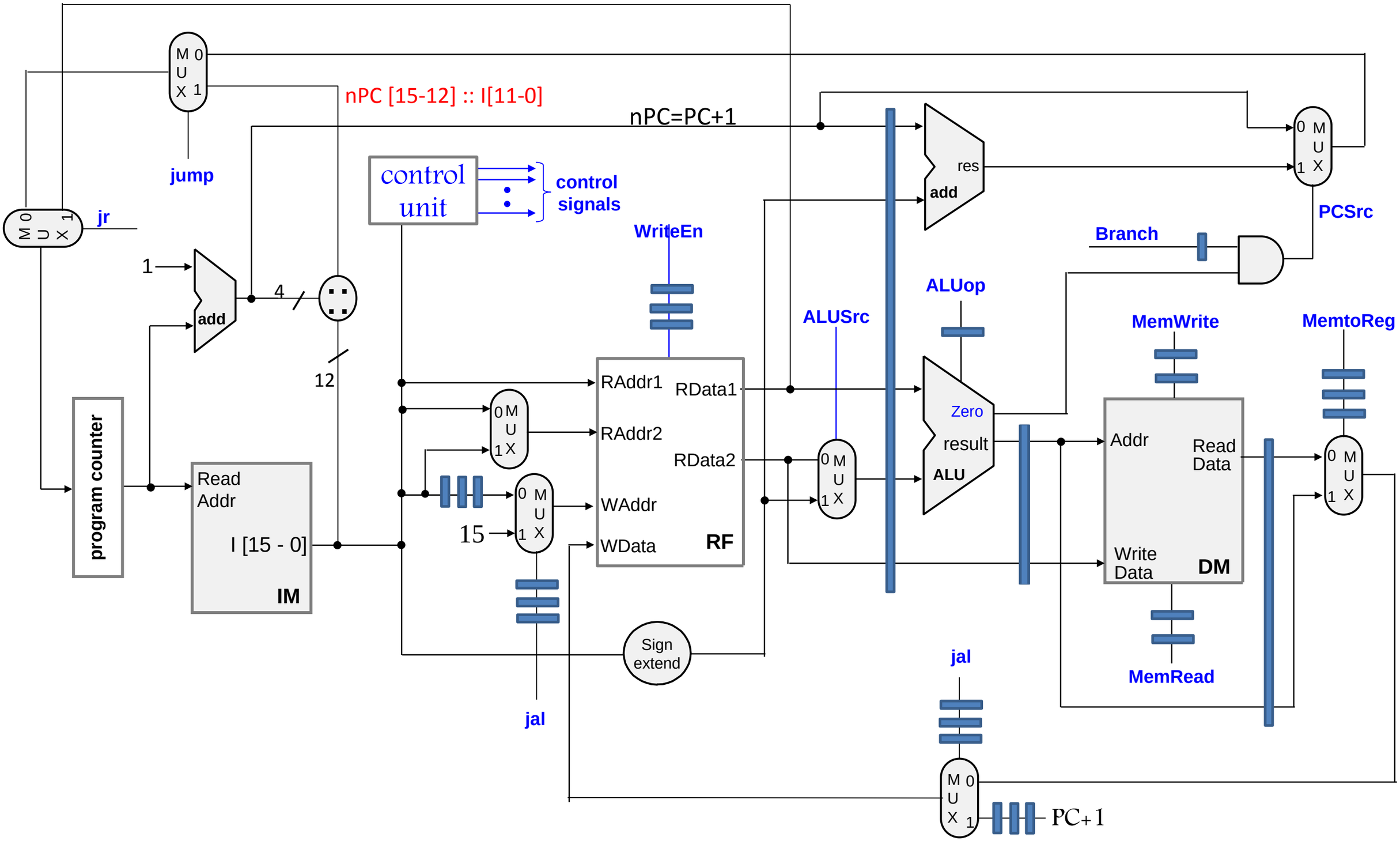
denotes pipeline register

Project Part-2: Modify the processor designed in Part-1 of the project to include jump register (jr), jump (J), and jump and link (jal) instructions. You need to modify the control unit to generate the control signals for the implementation of new instructions. The location of pipeline registers are shown in the figure in the next page. Jump register instruction (jr rs) is an R-type instruction where rt=rd=0. 'Jump' and 'Jump & link' are J-type instructions.



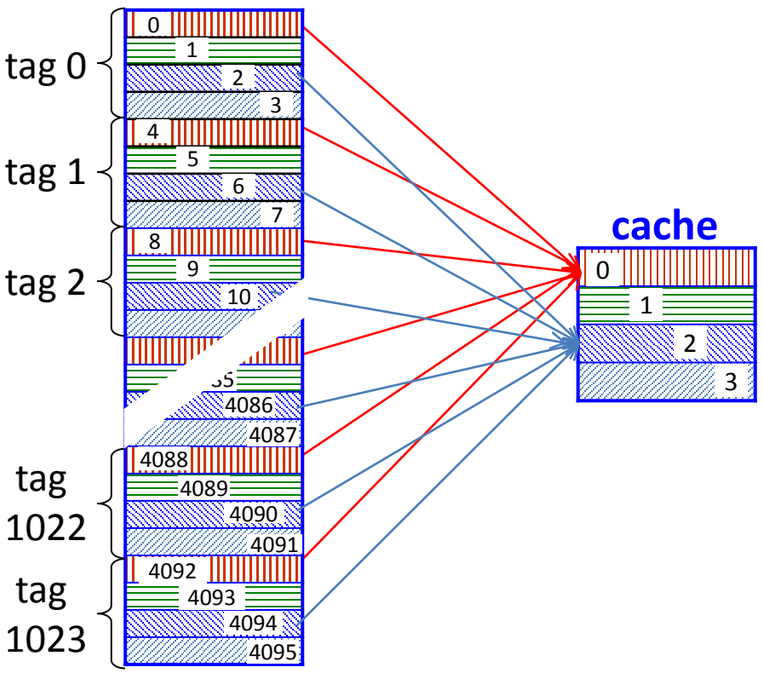


Note that pipelining is not changed due to inclusion of jump register (jr), jump (J), and jump and link (jal) instructions, since all these instructions get executed in the first pipeline stage.



Bonus Assignment: You will be required to incorporate a direct-mapped cache for data memory (DM) of size 64K words. Word size is 16-bit. The size of the cache is 64 words and the size of each cache line is 16 words. the cache can be implemented like a register file. Apart from that you will require a tag-bit register containing 4 registers.

You are required to implement this cache and demonstrate how a given program can be expedited by this cache. You will find the minimum execution time and the reduction of CPI which you achieve for the given program by using this cache.



Cache Design: The data memory (DM) capacity = 64K words: Consists of 4096 blocks. Each block contains 16 words, and word size= 16 bits. Each cache line (block) contains 16 words. 4096 DM blocks are mapped into 4 cache lines.

So the DM consists of 1024 sets where each set consists of 4 blocks.

1024 tags are required to identify 1024 sets of DM blocks.

Tag size of the address = 10 bits, since 2^10 = 1024: Tag-bit register consists of 4 registers , where width of each register is 10 bits. Cache hit occurs when the value read from tag-bit register is the same as tag-bits in the address word.

2-bit cache index is used as address of cache lines. 4-bit block-offset filed in the address is used as address of a word in a cache line.