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. 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 you will be required to find the minimum execution time and the reduction in 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)**

Submission date: 11th April 2016 to Hardware project's lab before 5pm

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 = 32bits) 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 $rt, imm ($rs), meaning : rt ← Mem[imm + $ rs]

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

branch on equal: beq $rs, $rt, imm, meaning: PC← nPC+imm, if $rs = $rt

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

6–bit	5–bit	5–bit	5–bit	11–bit
opcode	rs	rt	rd	Shamt+function

6–bit	5–bit	5–bit	16–bit
opcode	rs	rt	Immediate

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. Note that IM and DM has an inherent delay of one clock cycle in producing its output as memory is a clocked devise. Control signals are generated in the decode stage and therefore delayed by the number of clock cycles depending on the pipeline stage in which they are used.
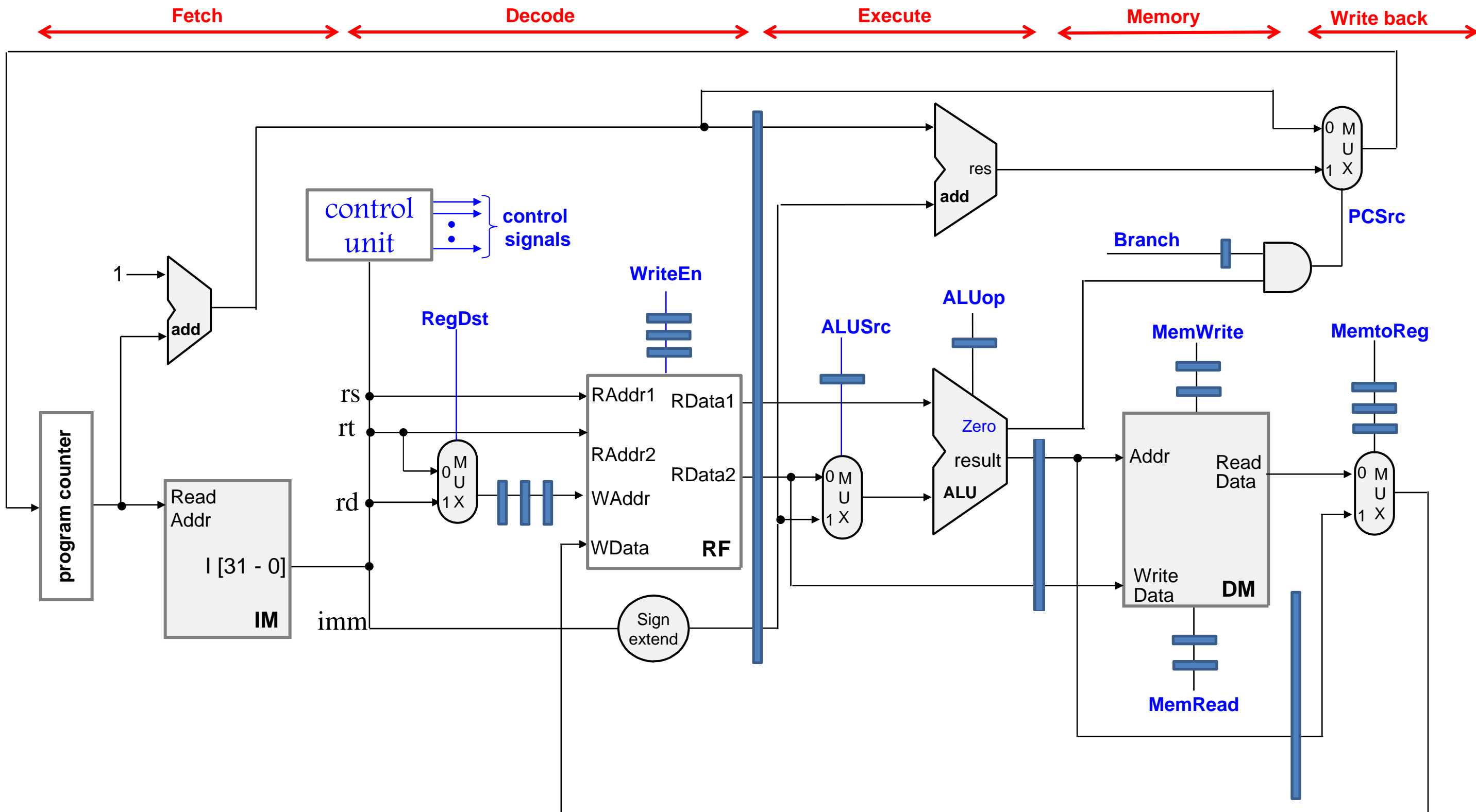
Project Part-2: Modify the processor designed in Part-1 of the project to include jump register (jr), jump (J), and jump and link (jal) instructions. You need to modify the control unit to generate the control signals for the implementation of new instructions. The location of pipeline registers are 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.
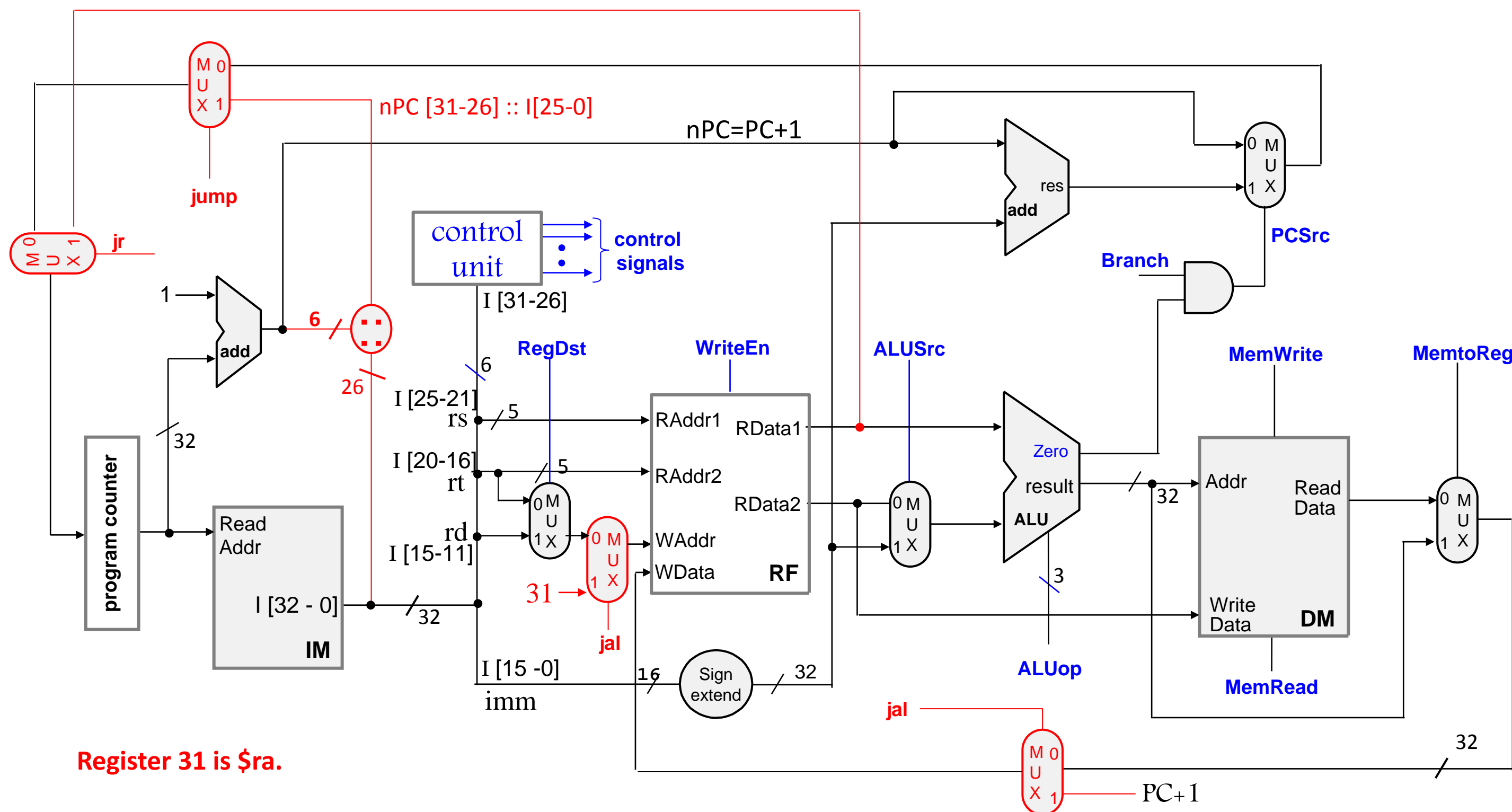
32-bit instruction word: J-type

6 bits	26 bits
opcode	offset

Register 31 is $ra.

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.

Fetch   Decode   Execute   Memory   Write back

M U X 0 1

nPC [31-26] :: I[25-0]

jump

nPC=PC+1

jr

control unit

control signals

WriteEn

res

add

Branch

PCSrc

ALUop

ALUSrc

MemWrite

MemtoReg

1

add

6

26

RegDst

RAddr1 RData1

RAddr2

RData2

0 M U X 1

Zero

result

ALU

Addr

Read Data

0 M U X 1

program counter

Read Addr

0 M U X 1

31

0 M U X 1

WAddr

WData

RF

Write Data

DM

I [31 - 0]

IM

Sign extend

jal

MemRead

jal

M U X 0 1

PC+1