Kairi Kozuma

GTID: 903050898

ECE 3056

03/01/2016

Assignment 3 Design and Verification Report

**Design: Modifications to Datapath**

****

**Figure 1.**  Multicycle datapath with modifications annotated in red to implement the jal instruction.

Changes made to datapath:

* RegDst control signal increased to two bits to accommodate 31 as a destination register for the jal instruction
  + 00 for Rt
  + 01 for Rd
  + 10 for $31 (Return address register)
* Register $31 added as third input to RegDst multiplexor
* MemtoReg control signal increased to two bits to allow PC + 4 to be written to $31
  + 00 for ALUOut
  + 01 for MemoryData
  + 10 for PC + 4
* PC + 4 signal routed as third input to MemtoReg multiplexor
* Dispatch table 1 expanded by two to include the jump and jal instructions

**Verification: Jump and Link Instruction Test Results**

X"00222020", -- add $4, $1, $2

X"0c000003", -- jal 0x0000000C [label1]

X"00832820", -- add $5, $4, $3

X"00a43020", -- add $6, $5, $4 ;[label1]

X"0c000006", -- jal 0x00000018 [label2]

X"00c33820", -- add $7, $6, $3

X"00c34020", -- add $8, $6, $3 ;[label2]

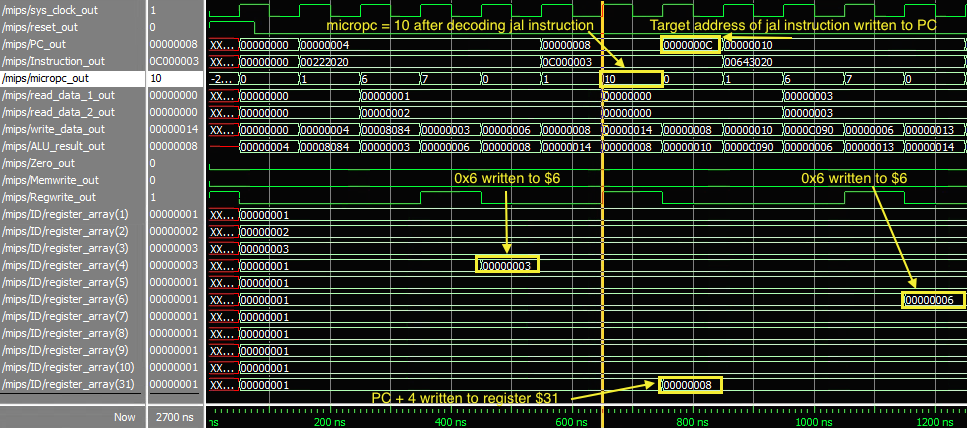
X"0c000009", -- jal 0x00000024 [endlabel]

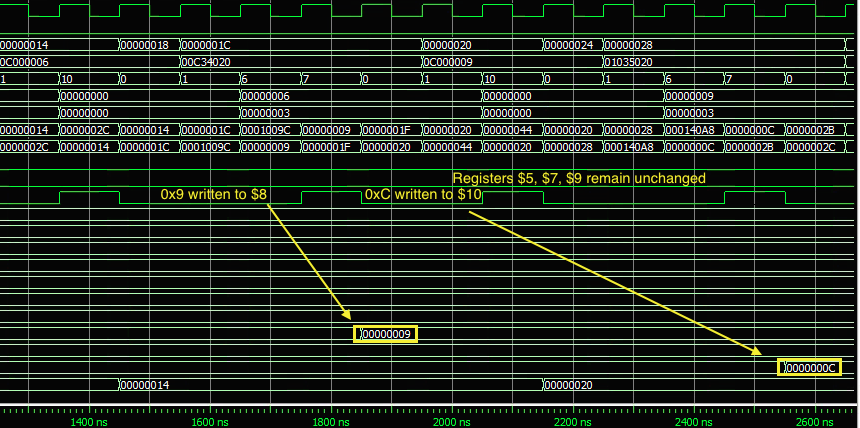
X"01034820", -- add $9, $8, $3

X"01035020", -- add $10, $8, $3 ;[endlabel]

**Figure 2.** Translated MIPS instructions to verify jal instruction.

In Figure 2, the hexadecimal binary encoded versions are shown with the original MIPS instructions in same-line comments. The target label for the hexadecimal address is noted in brackets. If an instruction has an associated label, it is noted in brackets after the semicolon.





**Figure 3.** Simulation trace of the test program. The cursor is located at the start of the jal stage of the first jal instruction, indicated by micropc = 10. The PC + 4 ALU result is written back to register $31, and the target address for the first jal (0x0000000C) is written to the PC. The jump is implemented correctly, evidenced by unchanged values in registers $5, $7, $9.