CS305
Computer Architecture

Extending the Multi-Cycle MIPS Implementation

Original subset:
add, sub, and, or, slt
lw, sw
beq

Bhaskaran Raman
Room 406, KR Building
Department of CSE, IIT Bombay

http://www.cse.iitb.ac.in/~br

Adding Support for `j`: Execution Plan

Stage-1:

(a) Instruction fetch
IR = Mem[PC]

(b) PC = PC+4

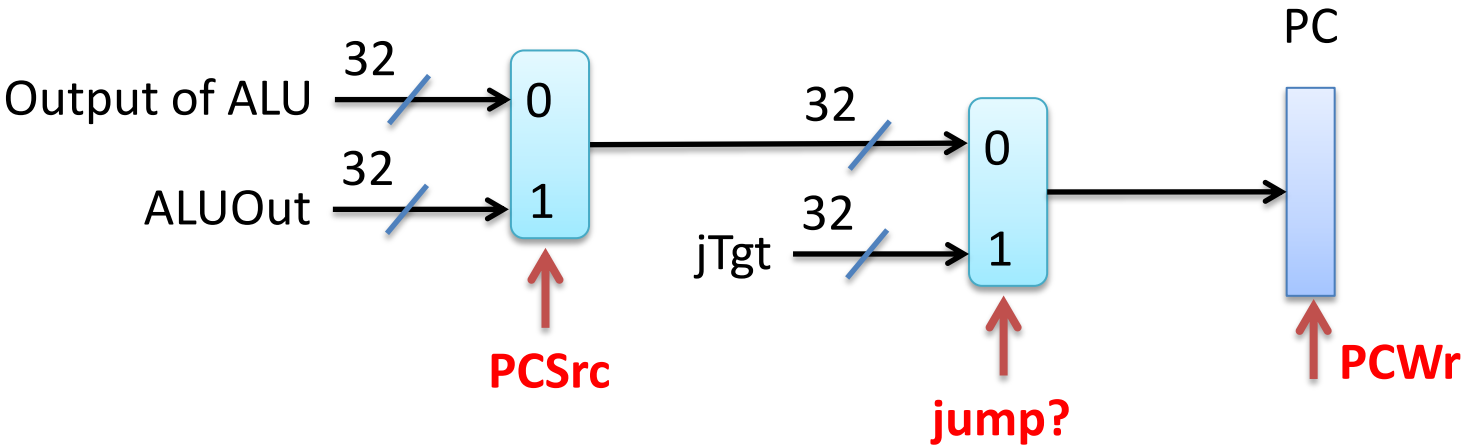
Stage-2:

(a) Register read
Reg1T = Reg[Rs]
Reg2T = Reg[Rt]

(b) Compute BrTgt
ALUOut =
(PC+4) + SgnExtImmShift2

(c) If jump instrn, PC = jTgt

Adding Support for `j`: Data Path Changes



Adding Support for `j`: Control Logic Changes

Stage-2:

(a) Register read
    Reg1T = Reg[Rs]
    Reg2T = Reg[Rt]

(b) Compute BrTgt
    ALUOut =
    (PC+4) + SgnExtImmShift2

(c) If jump instrn, PC = jTgt

RegWr: **0=disable**, 1=enable

RegDst: 0=Rd, 1=Rt

Mem2Reg: 0=ALUOut, 1=MDR

*ALUIp1Ctl: 0=Reg1Val, **1=PC***

*ALUIp2Ctl2: 00=Reg2Val, 01=4, 10=SgnExtImm, **10=SgnExtImmShft2***

ALUOp2: **00=add**, 01=sub, 10=reg-reg

MemRd, MemWr: **0=disable**, 1=enable

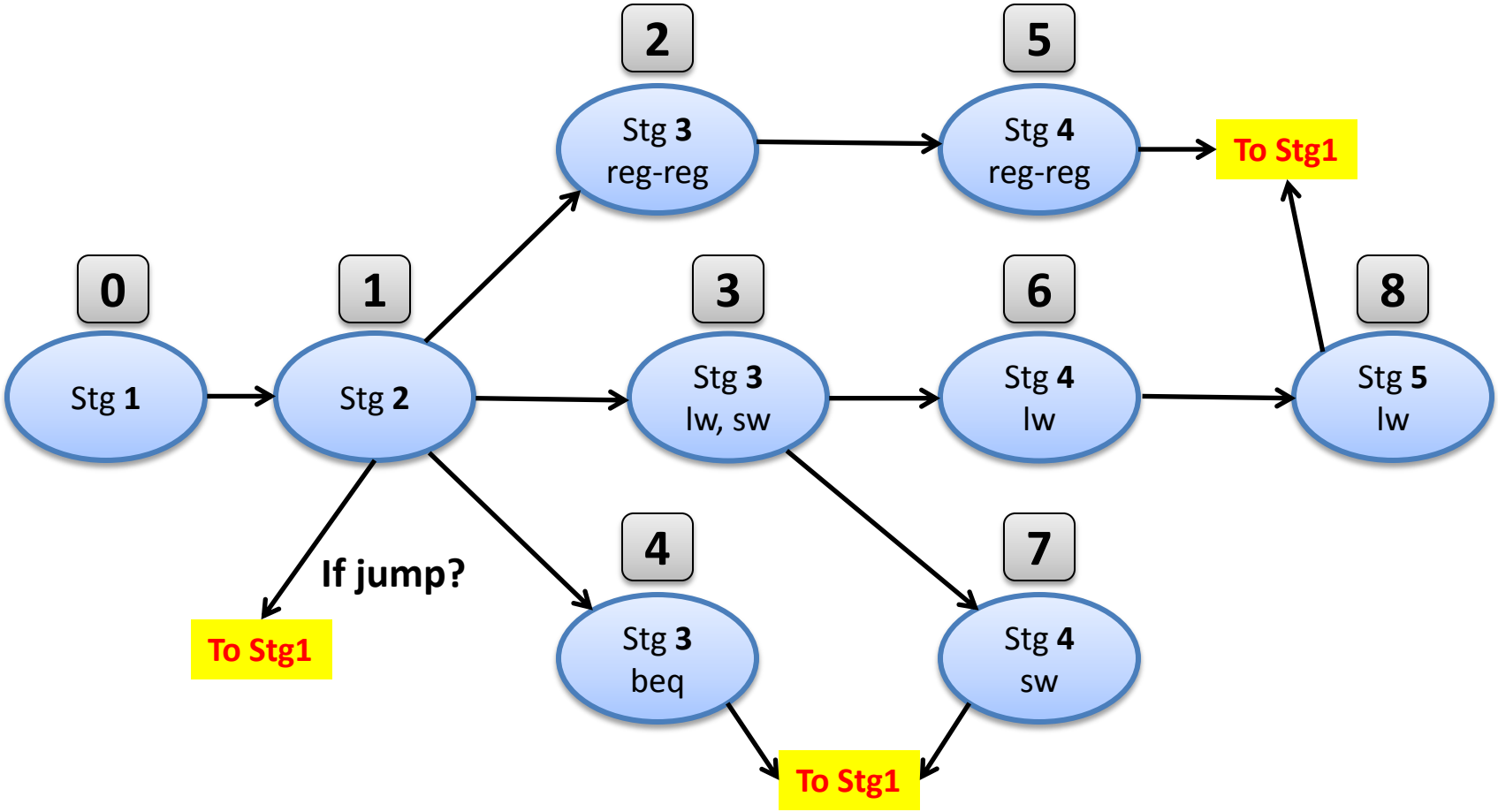
PCSrc: 0=OutputOfALU, 1=ALUOut IorD = *

Jump?: 1 if jump instrn, 0 otherwise

*PCWr: **jump?***

*IRWr: **0=disable**, 1=enable*

Adding Support for `j`: State Machine Changes



Summary

- Changes to be specified in terms of:
  - Execution plan enhancement/changes
  - Data path changes
  - State machine changes
  - Control lines in old/new/modified states