CS305
Computer Architecture

Multi Cycle Implementation of MIPS ISA (Subset)

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

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

Multi-Cycle Implementation: Overall Idea

- High level fixes from single-cycle implementation:
  - Reuse hardware components
  - Each instruction executes only as long as necessary
- Main changes to incorporate these fixes:
  - I and D memory the same
  - PC+4, BrTgt also computed using main ALU
  - Latches between main hardware components: remember values across cycles
  - Control will become much more complex!

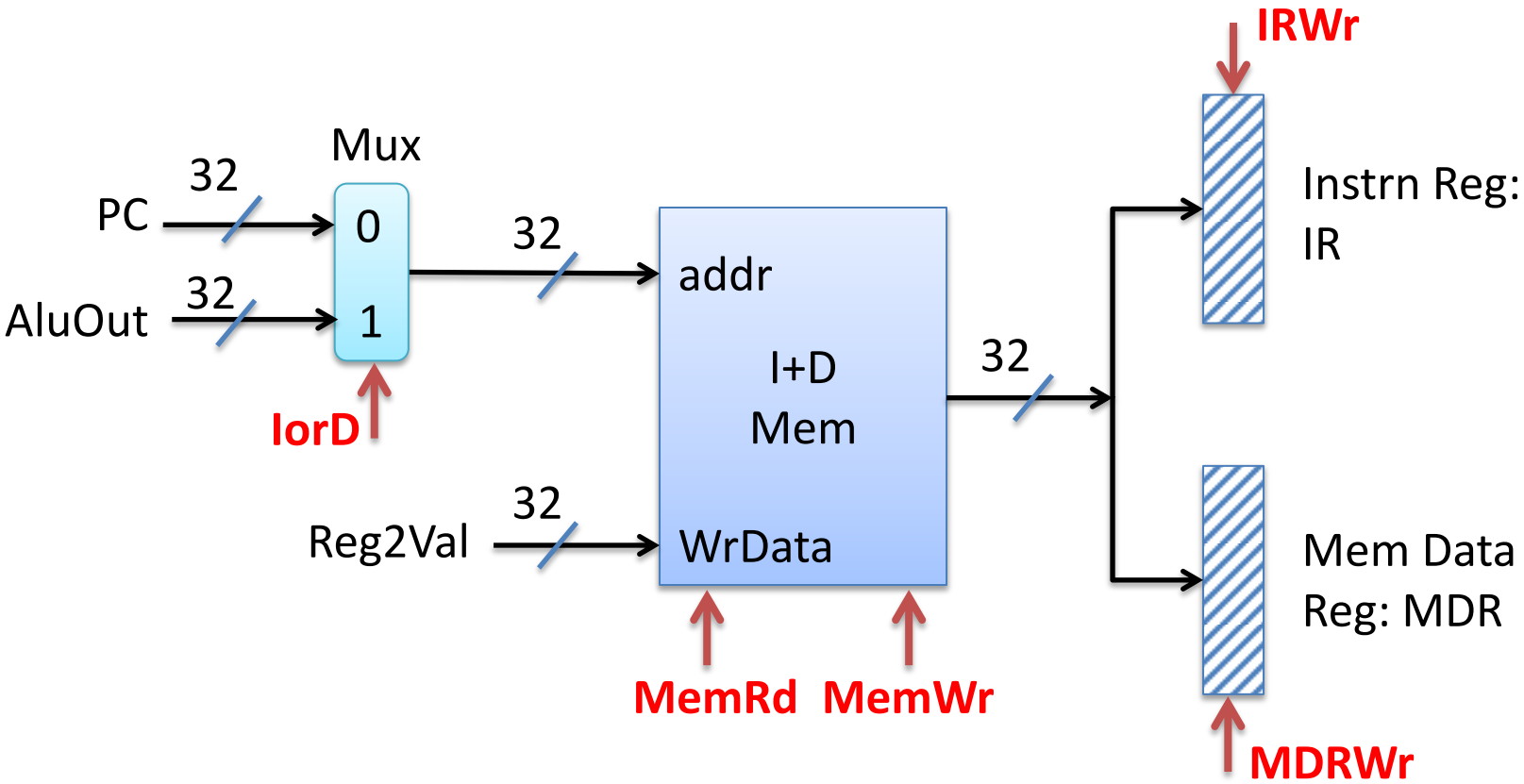
MIPS ISA Subset

- Reg-reg: **`add, sub, and, or, slt`**
- Memory: **`lw, sw`**
- Branch: **`beq`** (and **`j`** later)
- Subset ➔ keep it simple, understand techniques

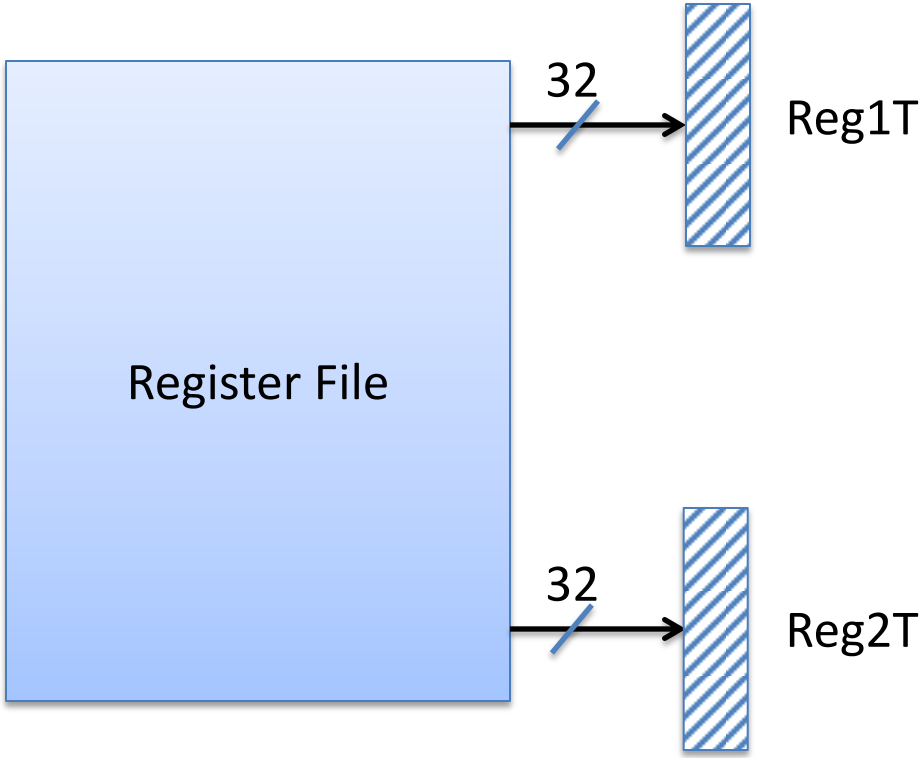
The Three Modules in Multi-Cycle Implementation

- I+D memory
- Register File
- ALU

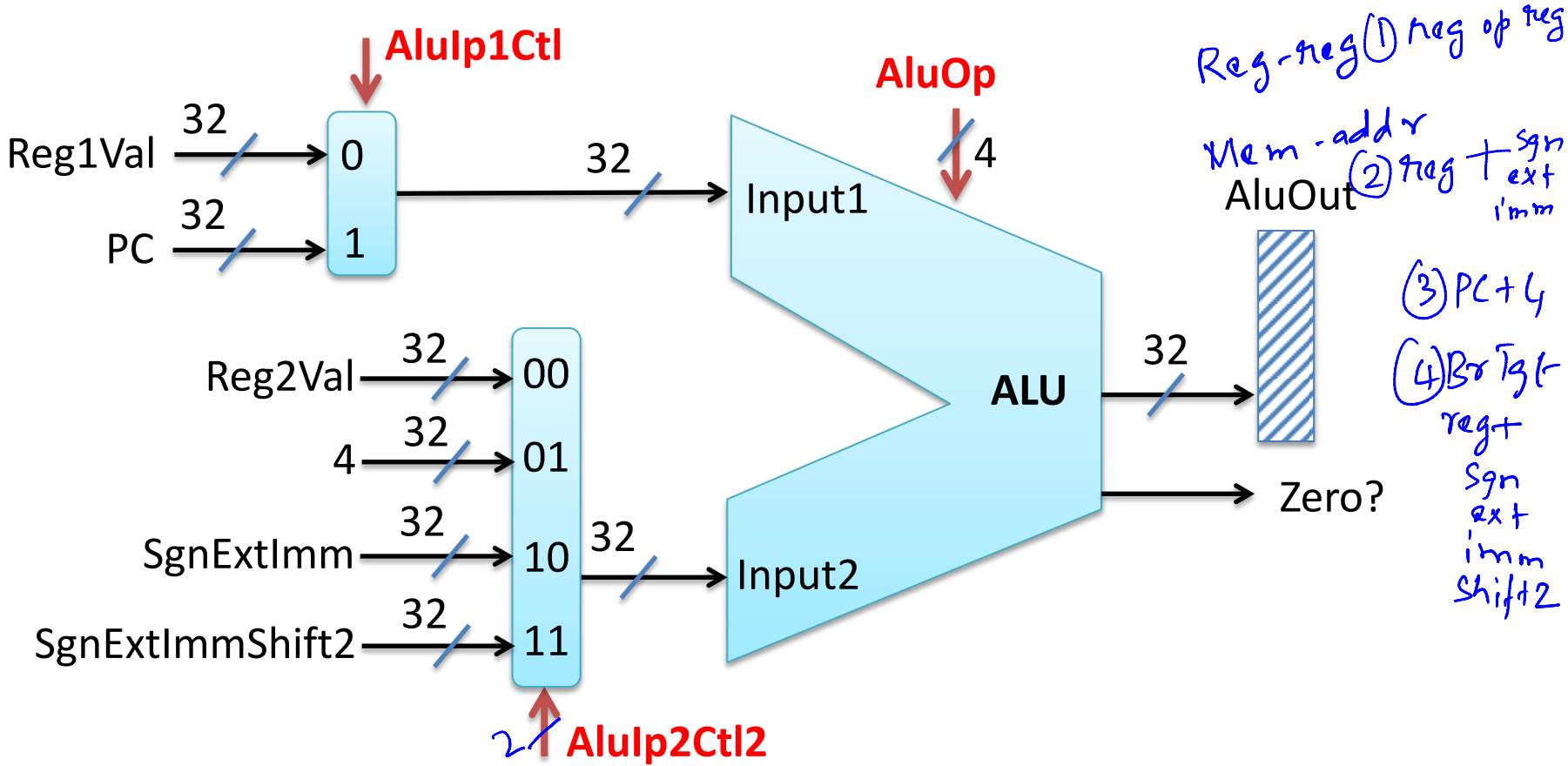
Data-Path, Control Changes: I+D Mem



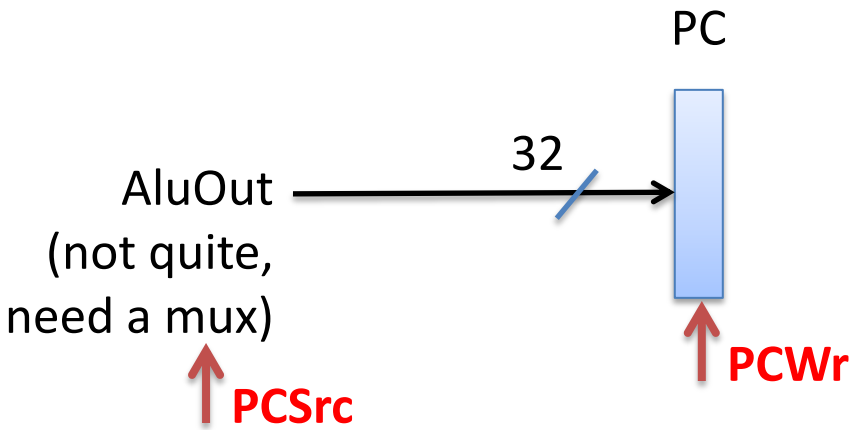
Data-Path, Control Changes: Register File



Data-Path, Control Changes: ALU



Data-Path, Control Path Changes: PC



Summary of Data Path Changes

- 3 modules: I+D mem, RegFile, ALU
- Changes to input of each: for reuse
  - Control lines to select input
- Output of each: to latch, to carry result to next cycle
  - Control lines to enable write

The Execution Plan: Definition

- Multi-cycle also called multiple ***stages***
- Execution Plan: what happens in each ***stage/cycle*** of an instruction
  - In terms of how each module is used, for what purpose

The 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

Stage-3: reg-reg
(a) ALU operation
ALUOut = Reg1T op Reg2T

Stage-3: lw, sw
(a) Mem addr computation
ALUOut = Reg1T + SgnExtImm

Stage-3: beq
(a) Branch condition check
ALUOut = Reg1T – Reg2T
zero? = (ALUOut != 0)
(b) If zero?, PC = BrTgt (ALUOut)

The Execution Plan (Continued)

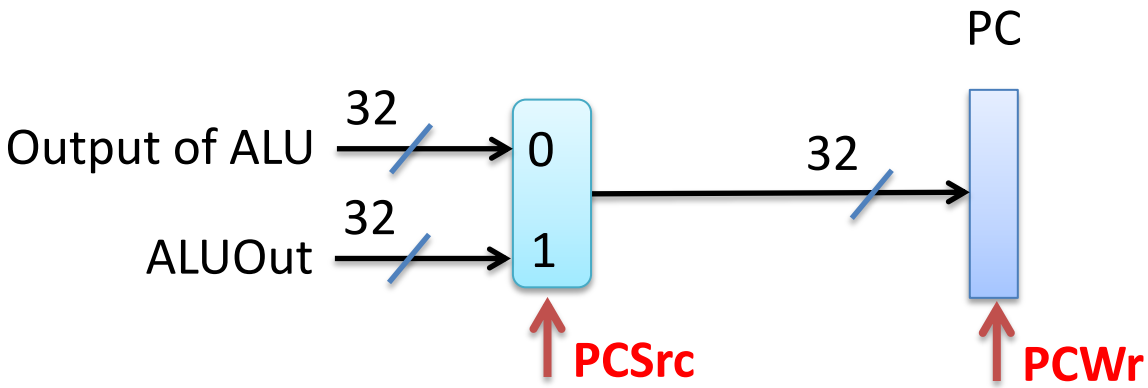
Stage-4: reg-reg
(a) Write back ALUOut to RegFile

Stage-4: lw
(a) Read memory
MDR = Mem[ALUOut]

Stage-4: sw
(a) Write memory
Mem[ALUOut] = Reg2T

Stage-5: lw
(a) Write back to RegFile from MDR

Overwriting PC: The Two Options



Summary

- Changes to multi-cycle implementation
  - Input muxes, Latches, Control lines
- Execution plan
  - Stage-by-stage description of instruction execution
- Next: logic for generation of control lines