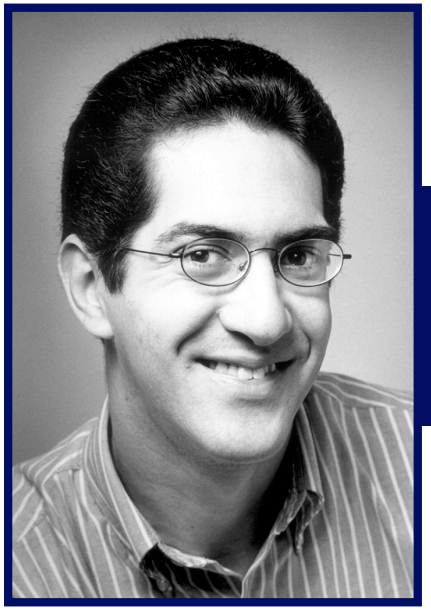
inst.eecs.berkeley.edu/~cs61c

CS61C : Machine Structures

Lecture 34 – Single Cycle CPU Control I



2004-04-16

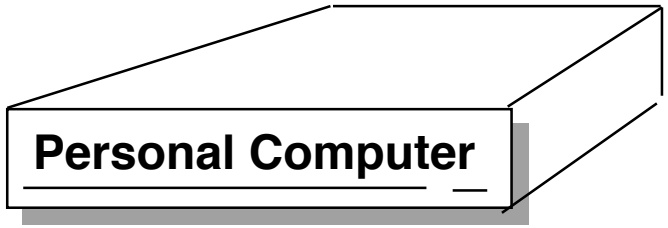
Lecturer PSOE Dan Garcia

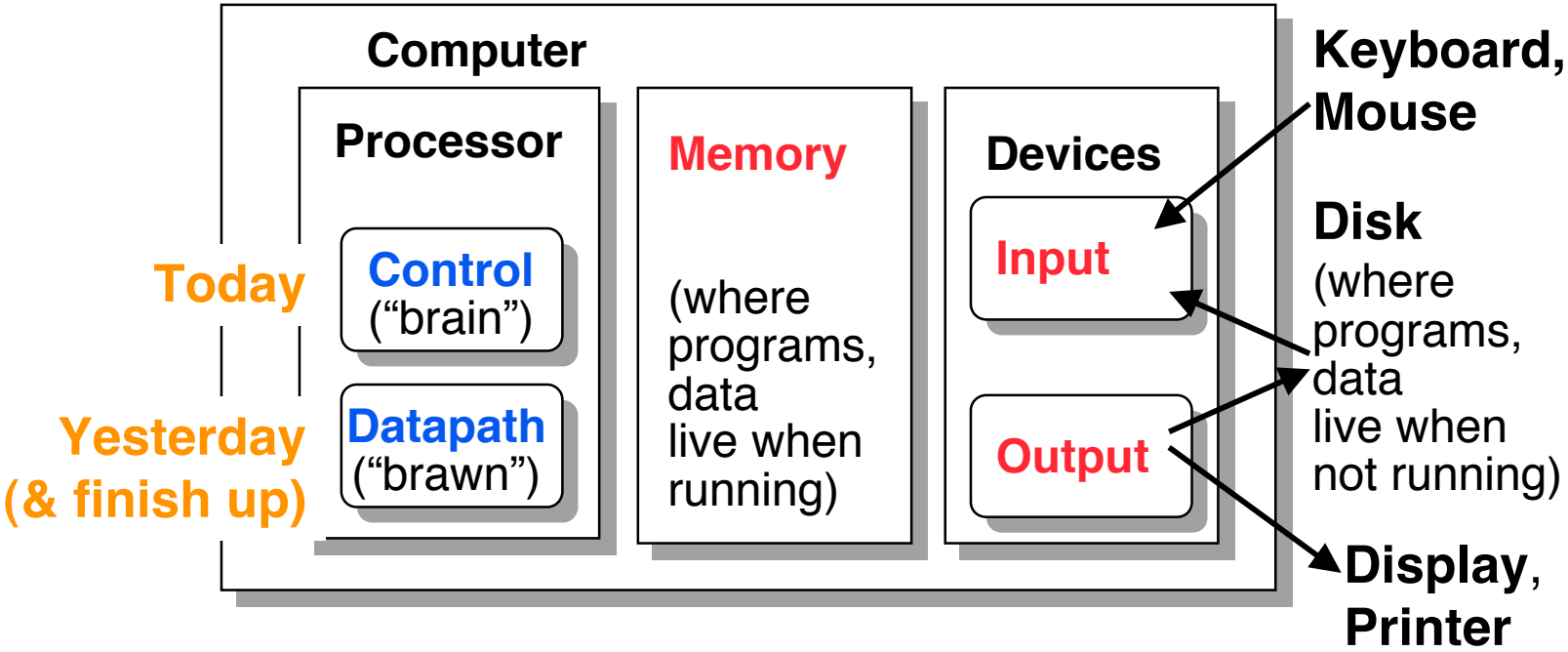
www.cs.berkeley.edu/~ddgarcia

10.5 Quake?! NBC ⇒ movie on May 3rd.

Garcia, Spring 2004 © UCB

Anatomy Review: 5 components of any Computer





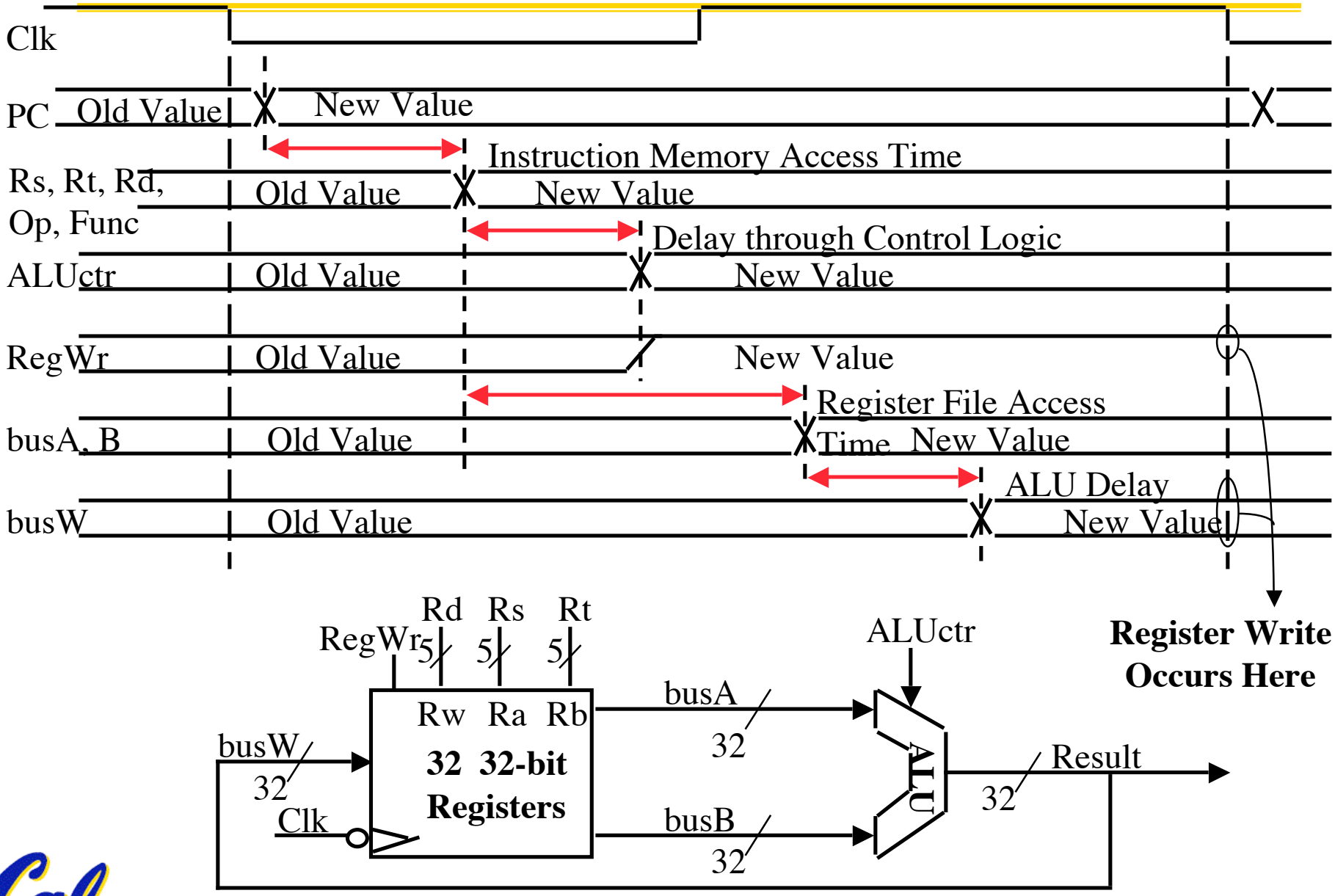
Garcia, Spring 2004 © UCB

Review: How to Design a Processor: step-by-step

- 1. Analyze instruction set architecture (ISA) => datapath requirements
  - meaning of each instruction is given by the *register transfers*
  - datapath must include storage element for ISA registers
  - datapath must support each register transfer
- 2. Select set of datapath components and establish clocking methodology
- 3. Assemble datapath meeting requirements
- **4. Analyze implementation of each instruction to determine setting of control points that effects the register transfer.**
- **5. Assemble the control logic**

Garcia, Spring 2004 © UCB

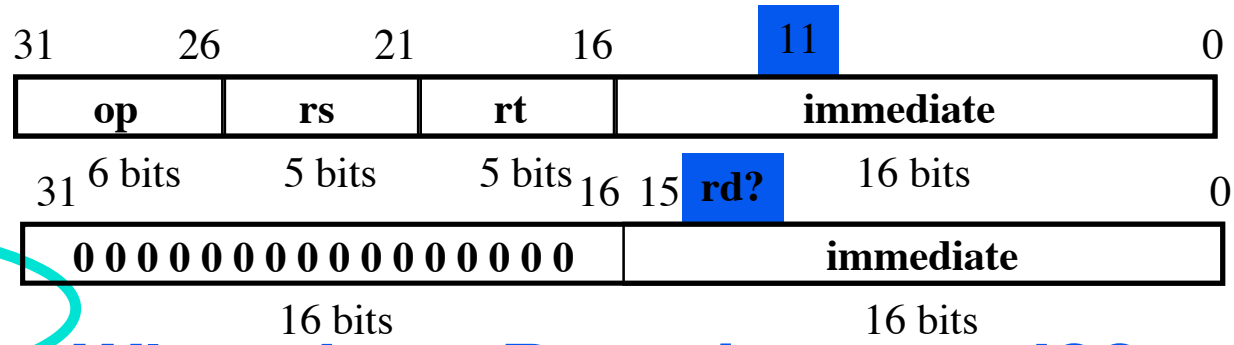
Register-Register Timing: One complete cycle

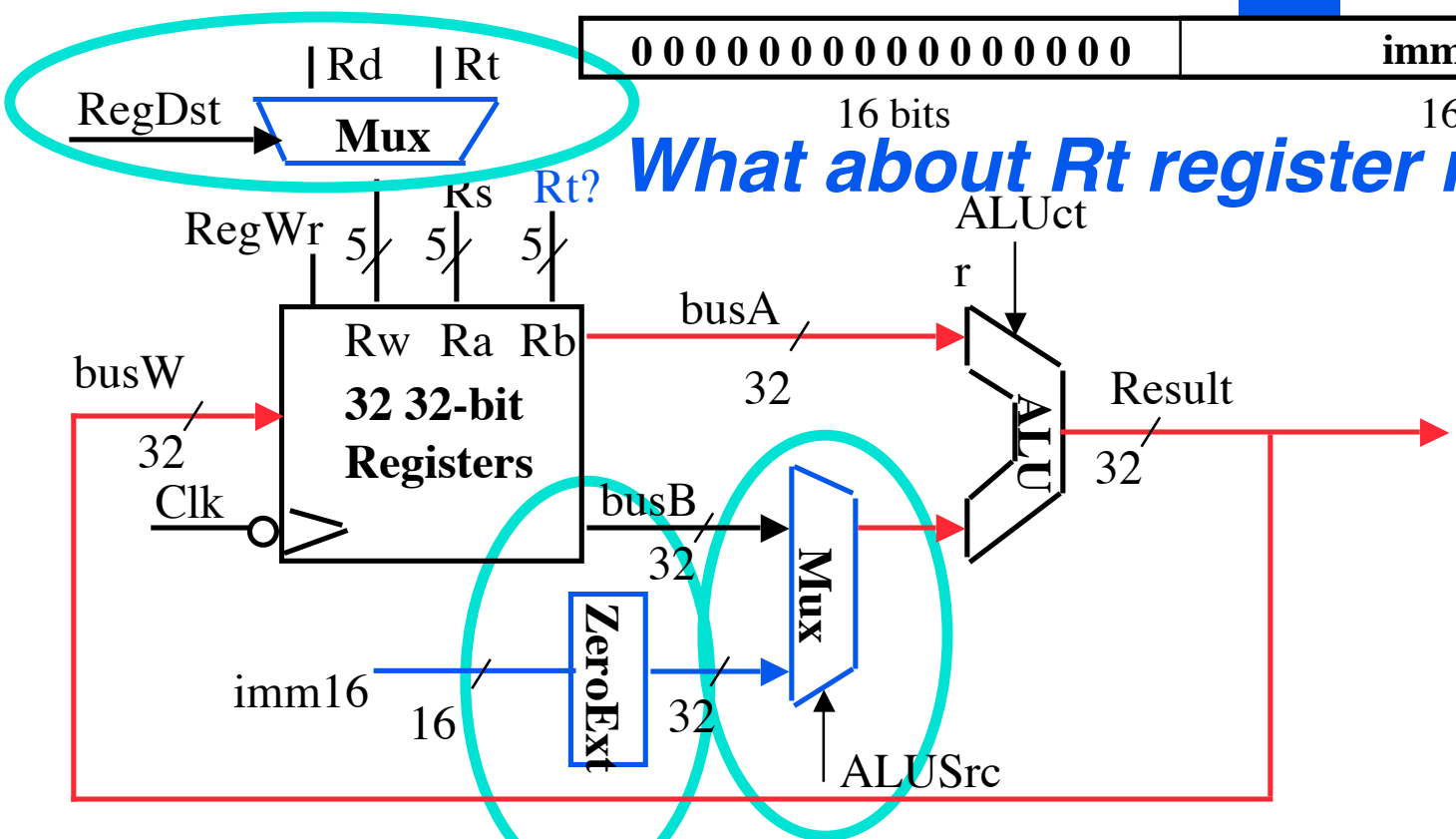


Garcia, Spring 2004 © UCB

3c: Logical Operations with Immediate

- R[rt] = R[rs] op ZeroExt[imm16]]





What about Rt register read??

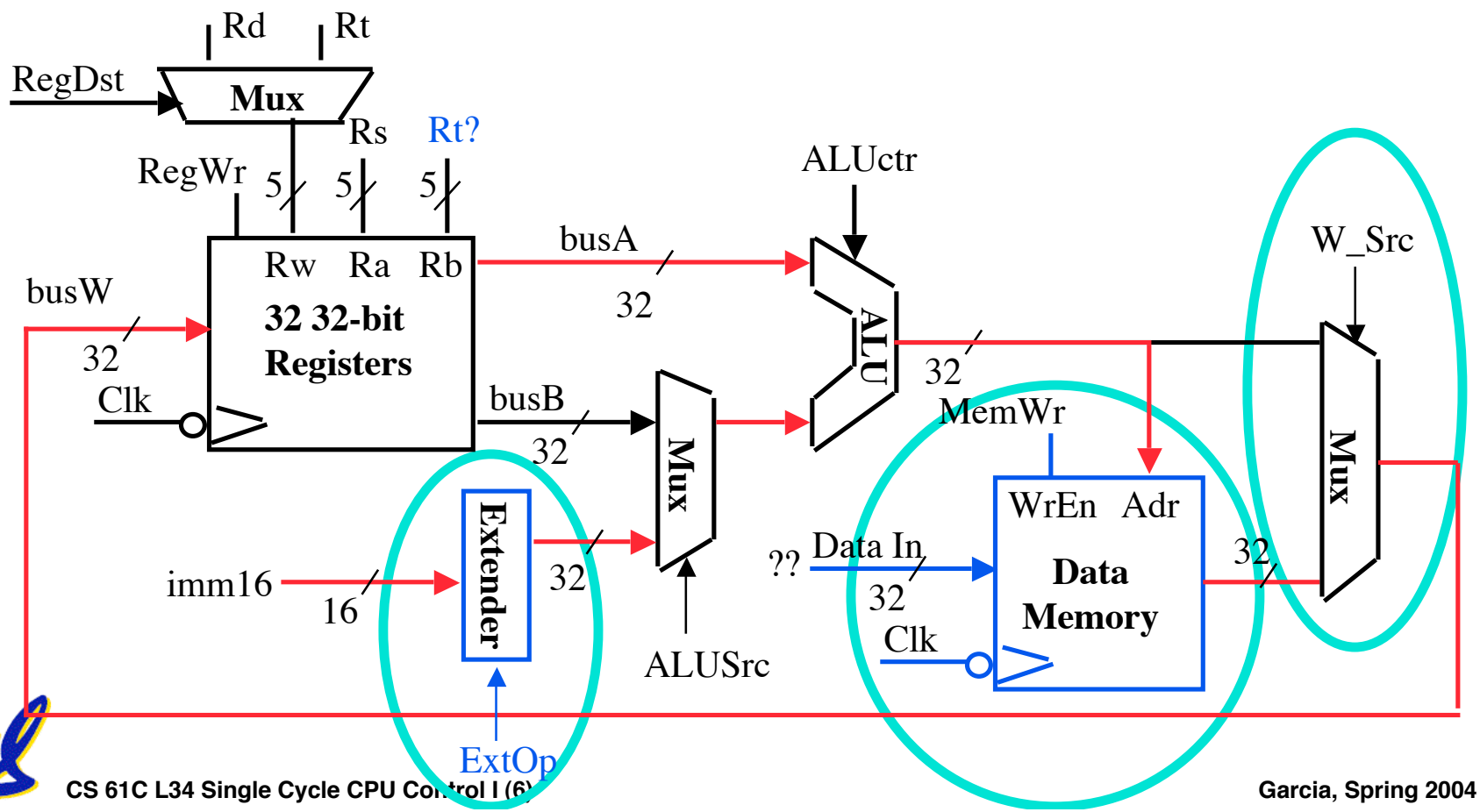
- Already defined 32-bit MUX; Zero Ext?

Garcia, Spring 2004 © UCB

3d: Load Operations

- R[rt] = Mem[R[rs] + SignExt[imm16]]
  Example: lw rt, rs, imm16

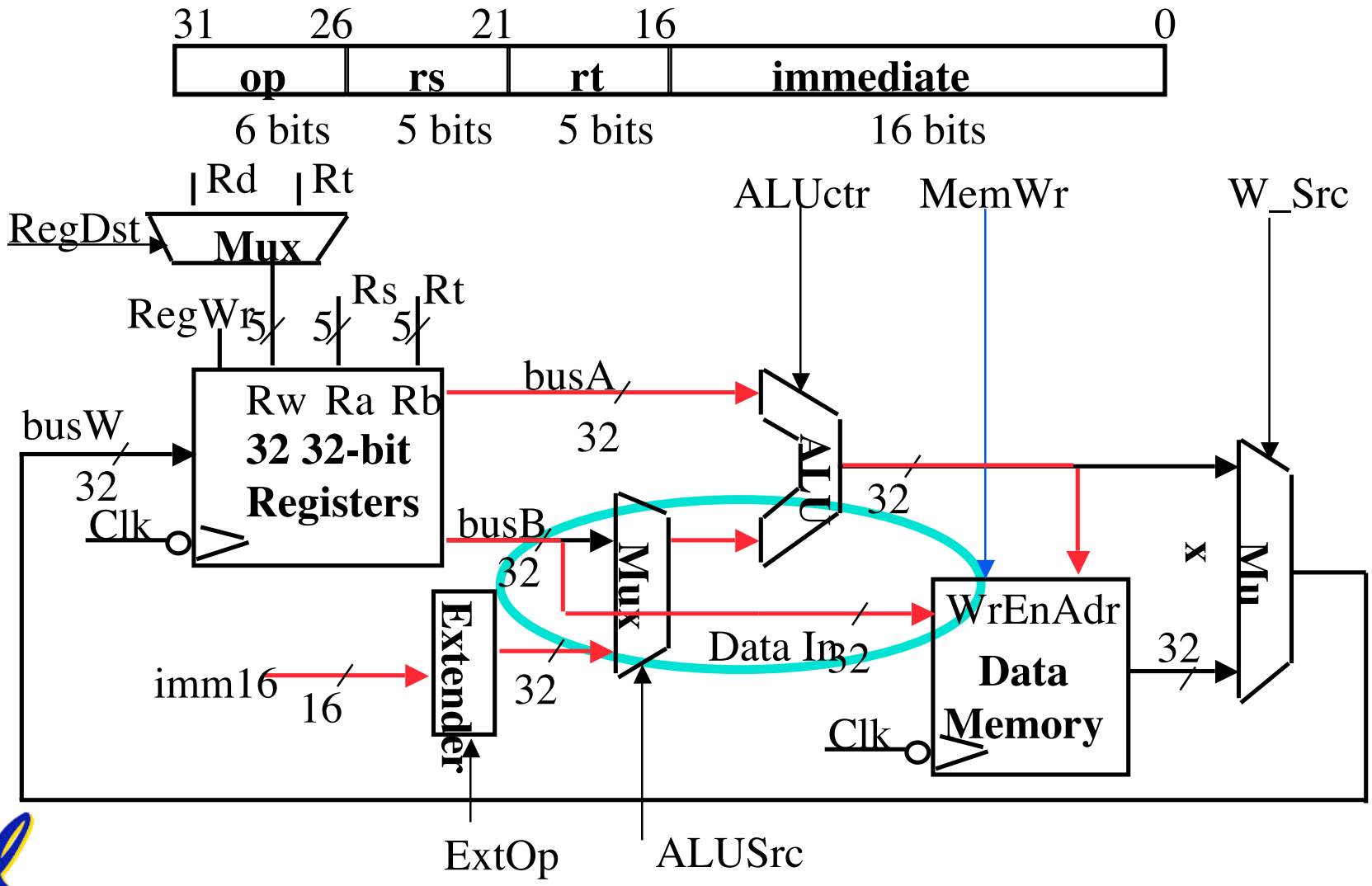
31–26	25–21	20–16	15–0
op	rs	rt	immediate
6 bits	5 bits	5 bits	16 bits



Garcia, Spring 2004 © UCB

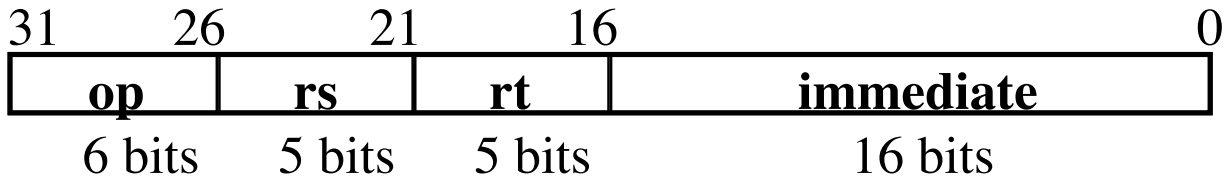
3e: Store Operations

- **Mem[R[rs] + SignExt[imm16]] = R[rt]**
  Ex.: sw rt, rs, imm16



Garcia, Spring 2004 © UCB

3f: The Branch Instruction



- **beq rs, rt, imm16**
  - **mem[PC] Fetch the instruction from memory**
  - **Equal = R[rs] == R[rt] Calculate the branch condition**
  - **if (Equal) Calculate the next instruction's address**
    - **PC = PC + 4 + (SignExt(imm16) x 4)**

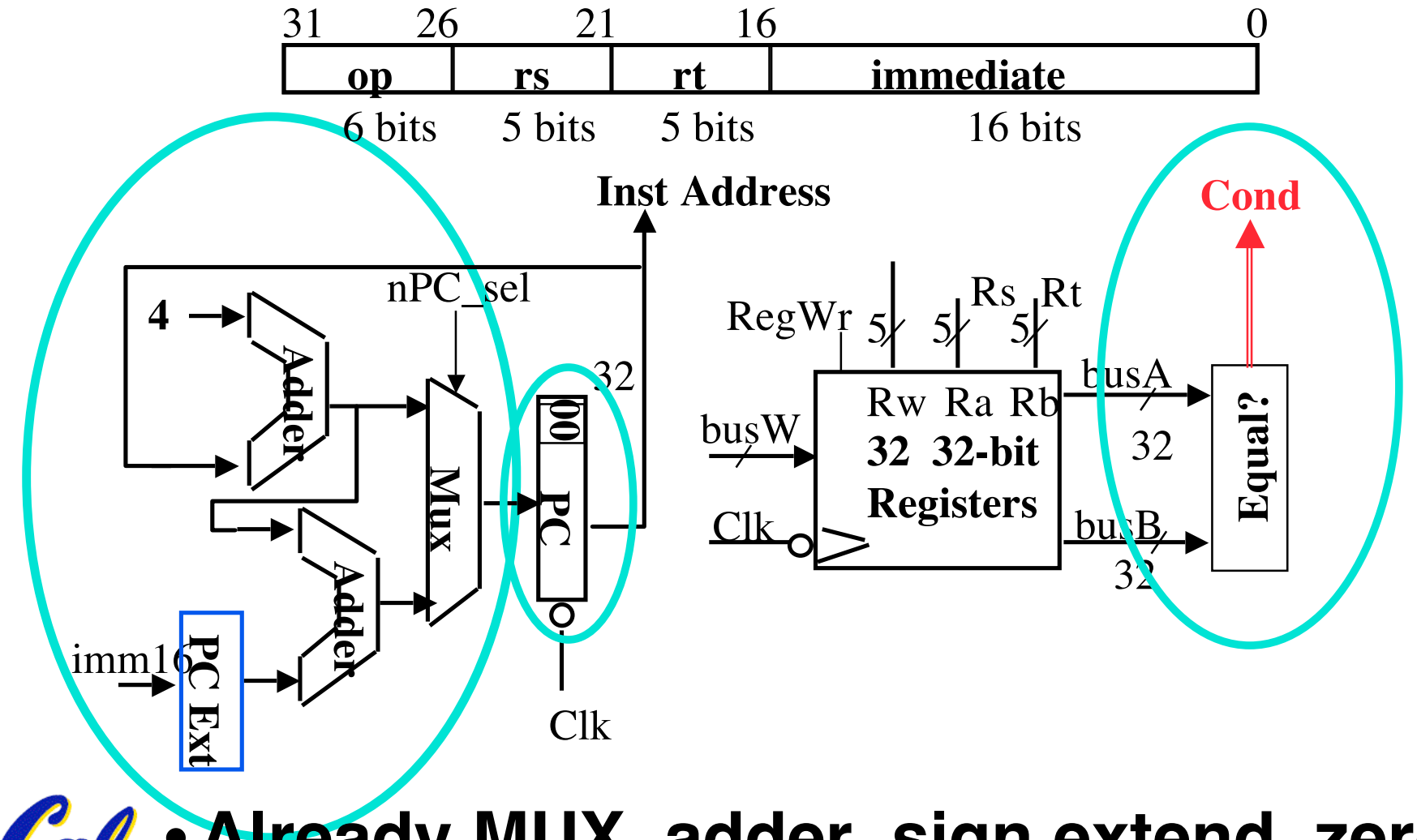
    else
    - **PC = PC + 4**

Garcia, Spring 2004 © UCB

Datapath for Branch Operations

- **beq rs, rt, imm16**

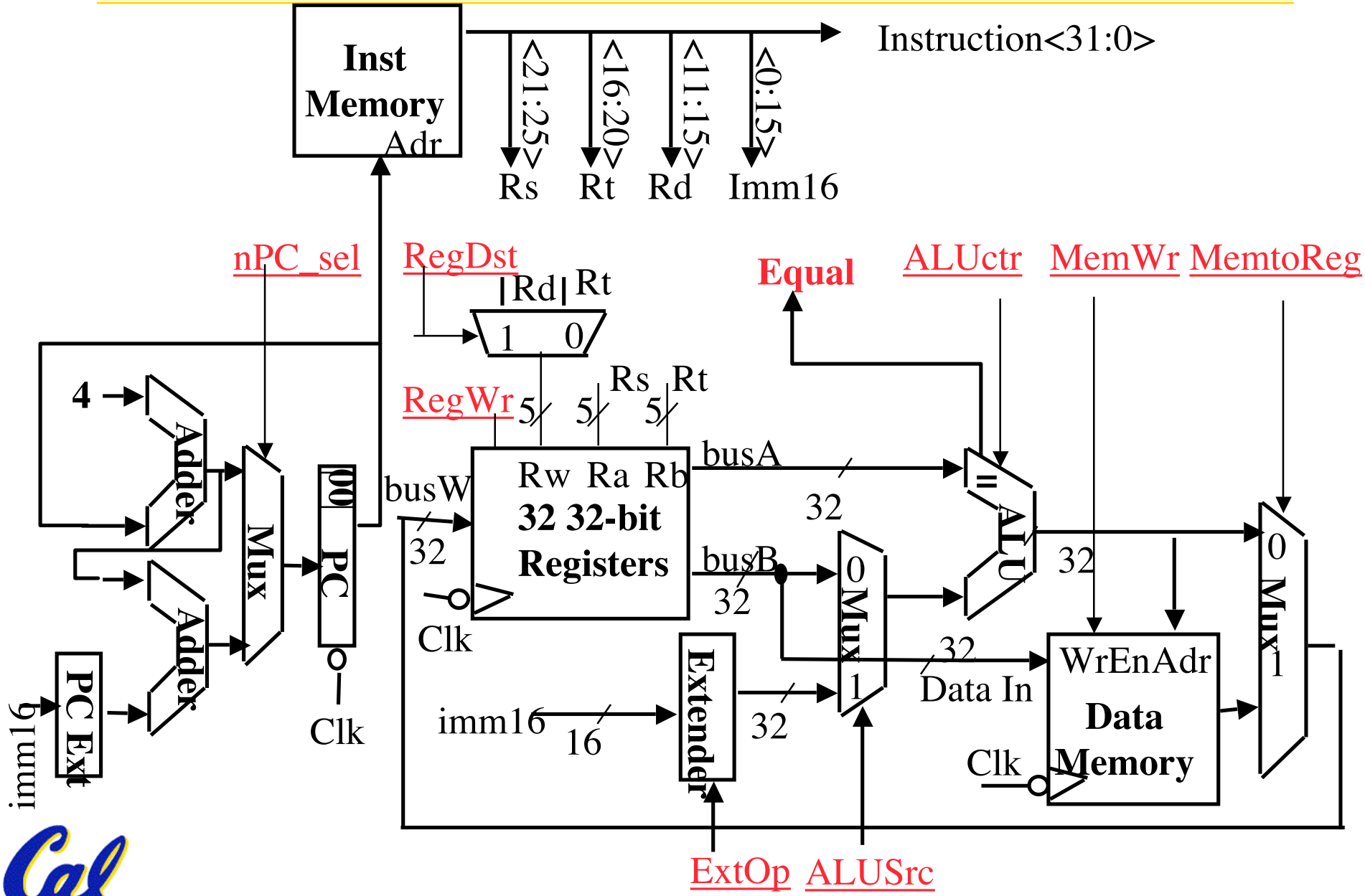
Datapath generates condition (equal)



- **Already MUX, adder, sign extend, zero**

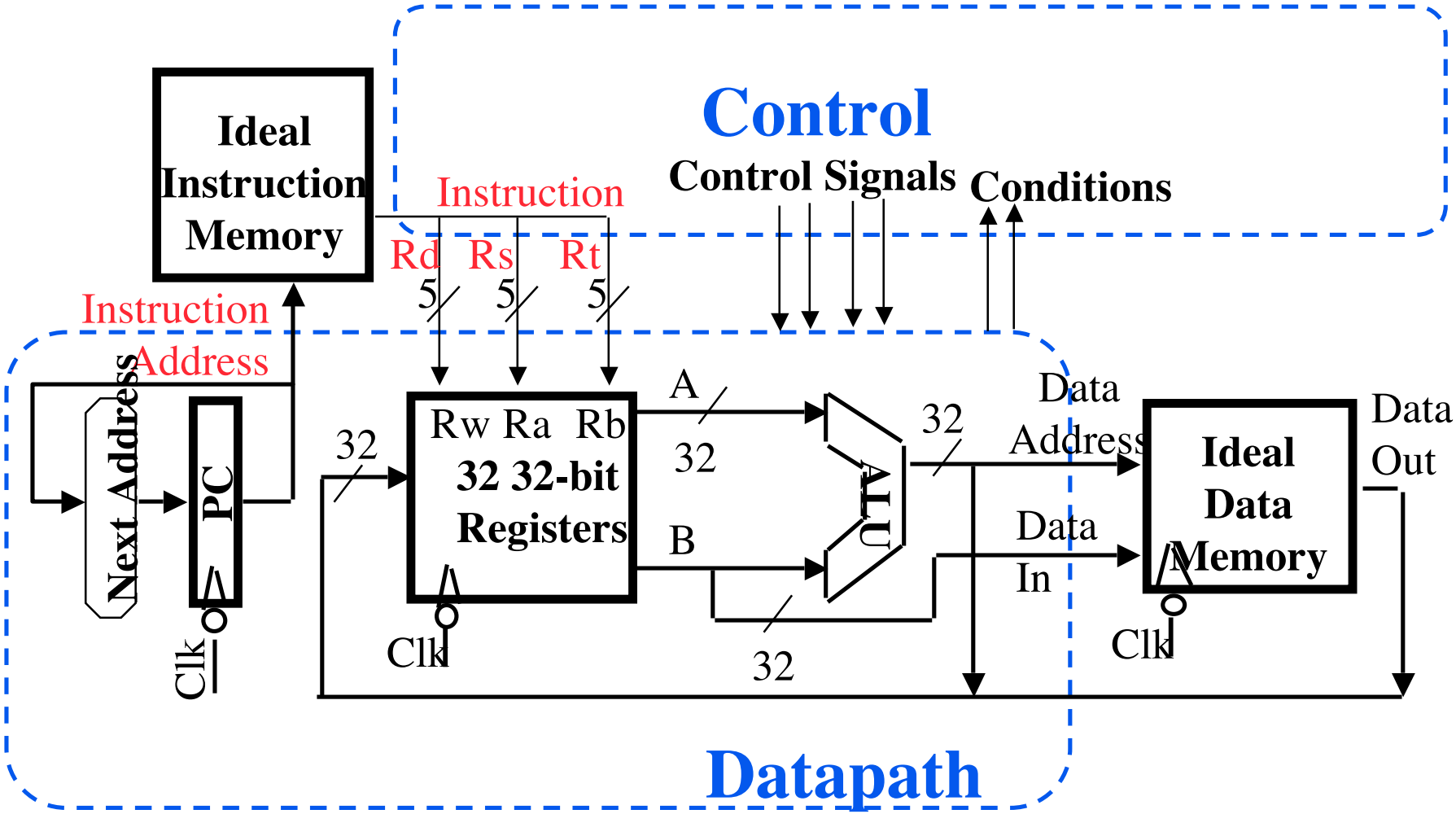
Garcia, Spring 2004 © UCB

Putting it All Together:A Single Cycle Datapath



Garcia, Spring 2004 © UCB

An Abstract View of the Implementation

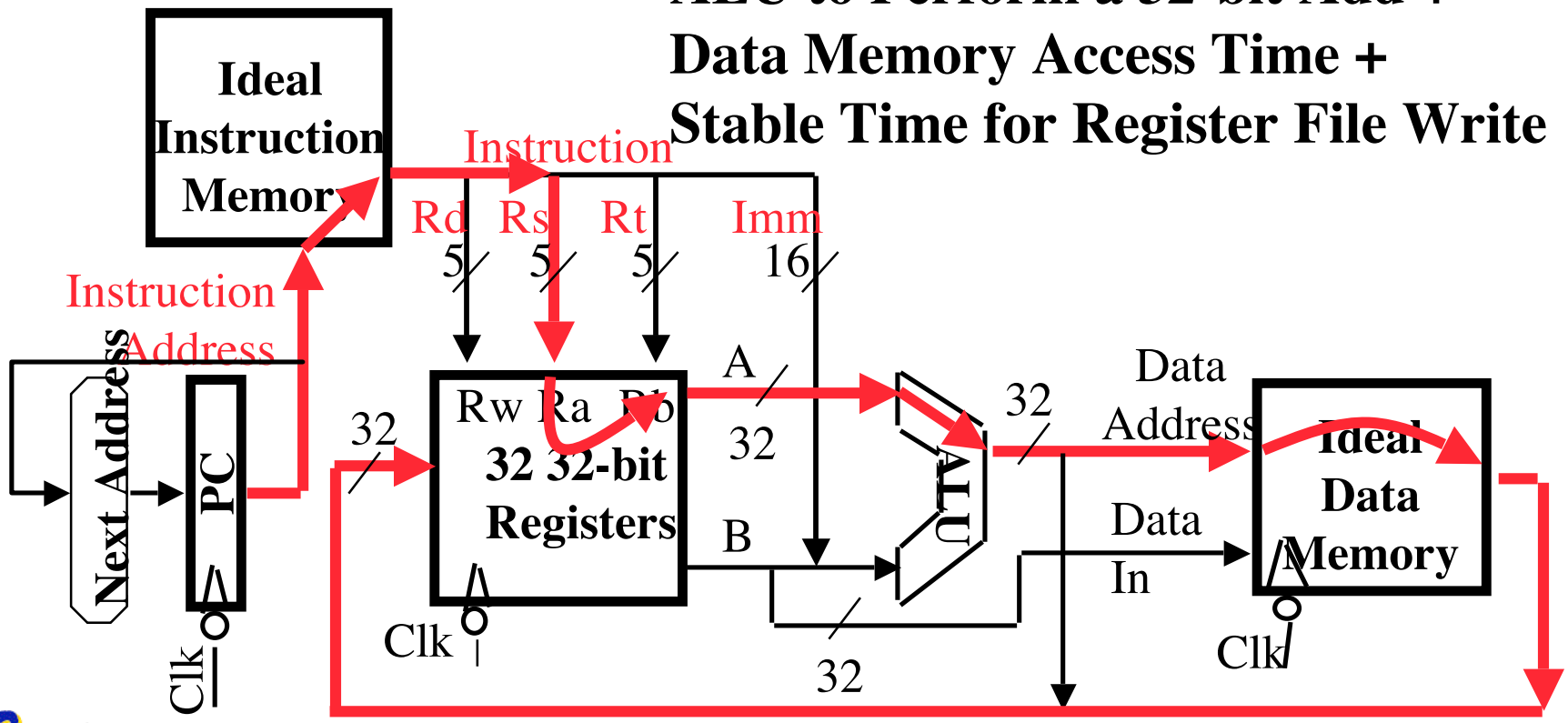


Garcia, Spring 2004 © UCB

An Abstract View of the Critical Path

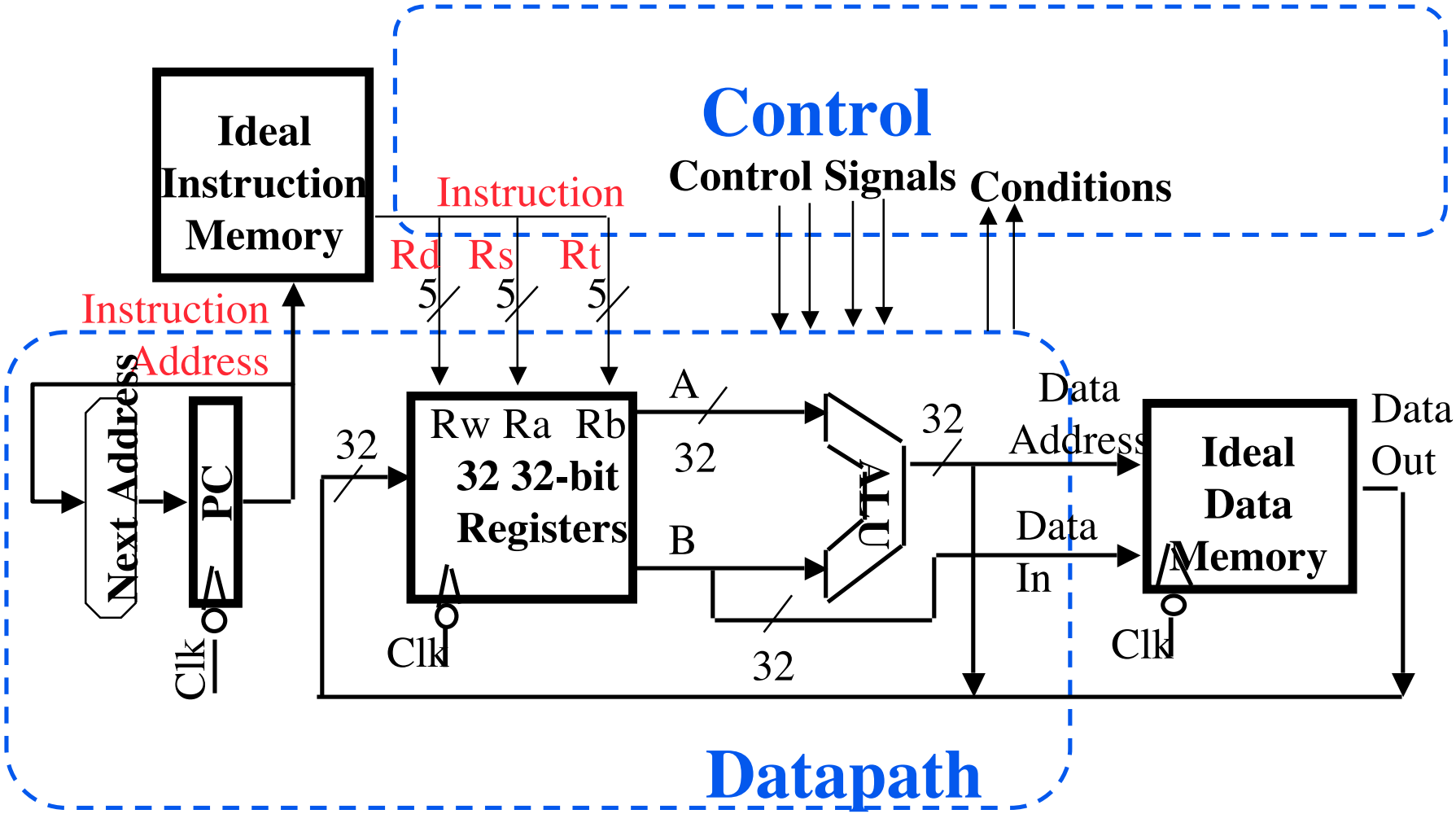
- This affects how much you can overclock your PC!

Critical Path (Load Operation) =
Delay clock through PC (FFs) +
Instruction Memory's Access Time +
Register File's Access Time +
ALU to Perform a 32-bit Add +
Data Memory Access Time +
Stable Time for Register File Write



Garcia, Spring 2004 © UCB

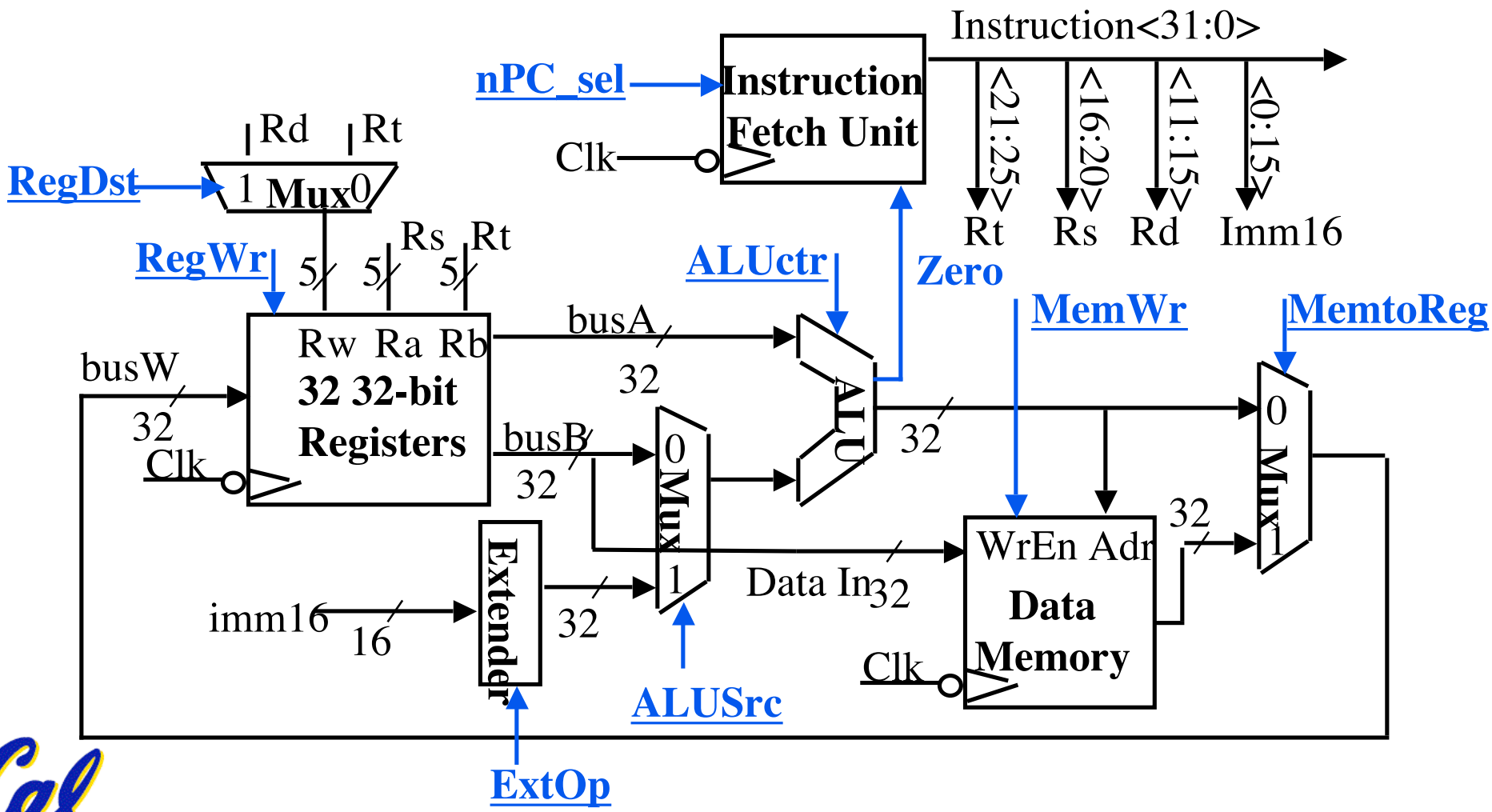
An Abstract View of the Implementation



Garcia, Spring 2004 © UCB

Recap: A Single Cycle Datapath

- Rs, Rt, Rd, Imed16 connected to datapath
- We have everything except control signals

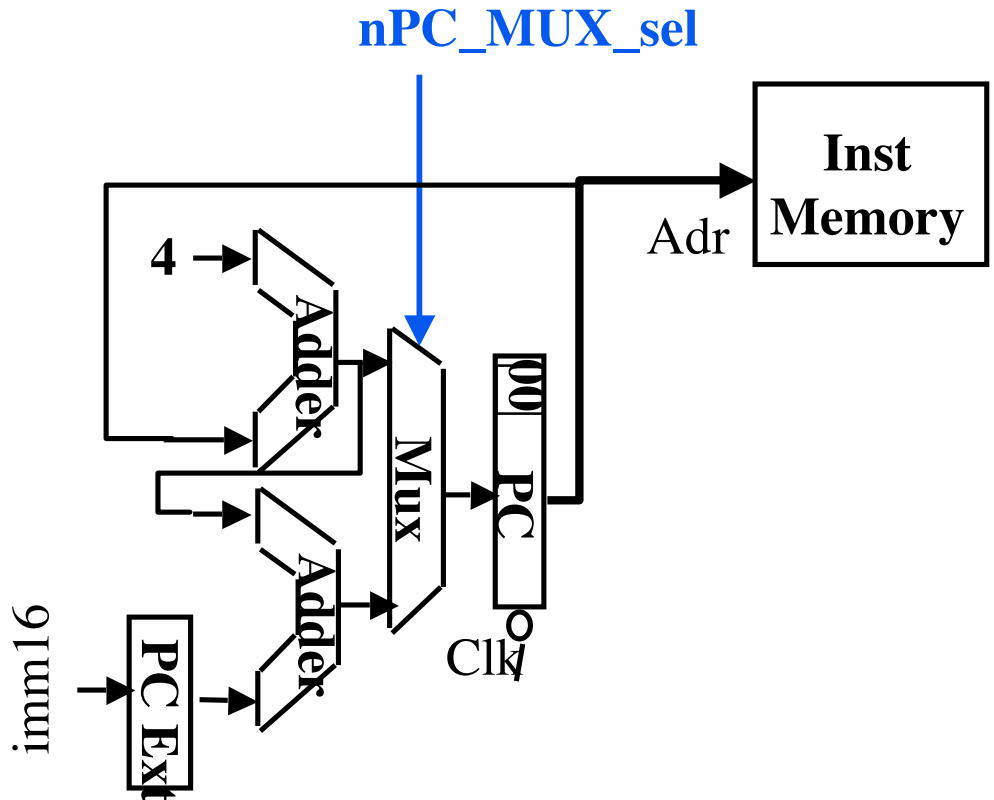


Garcia, Spring 2004 © UCB

Recap: Meaning of the Control Signals

- **nPC_MUX_sel:** 0 ⇒ PC <– PC + 4
  1 ⇒ PC <– PC + 4 + {SignExt(Im16) , 00 }

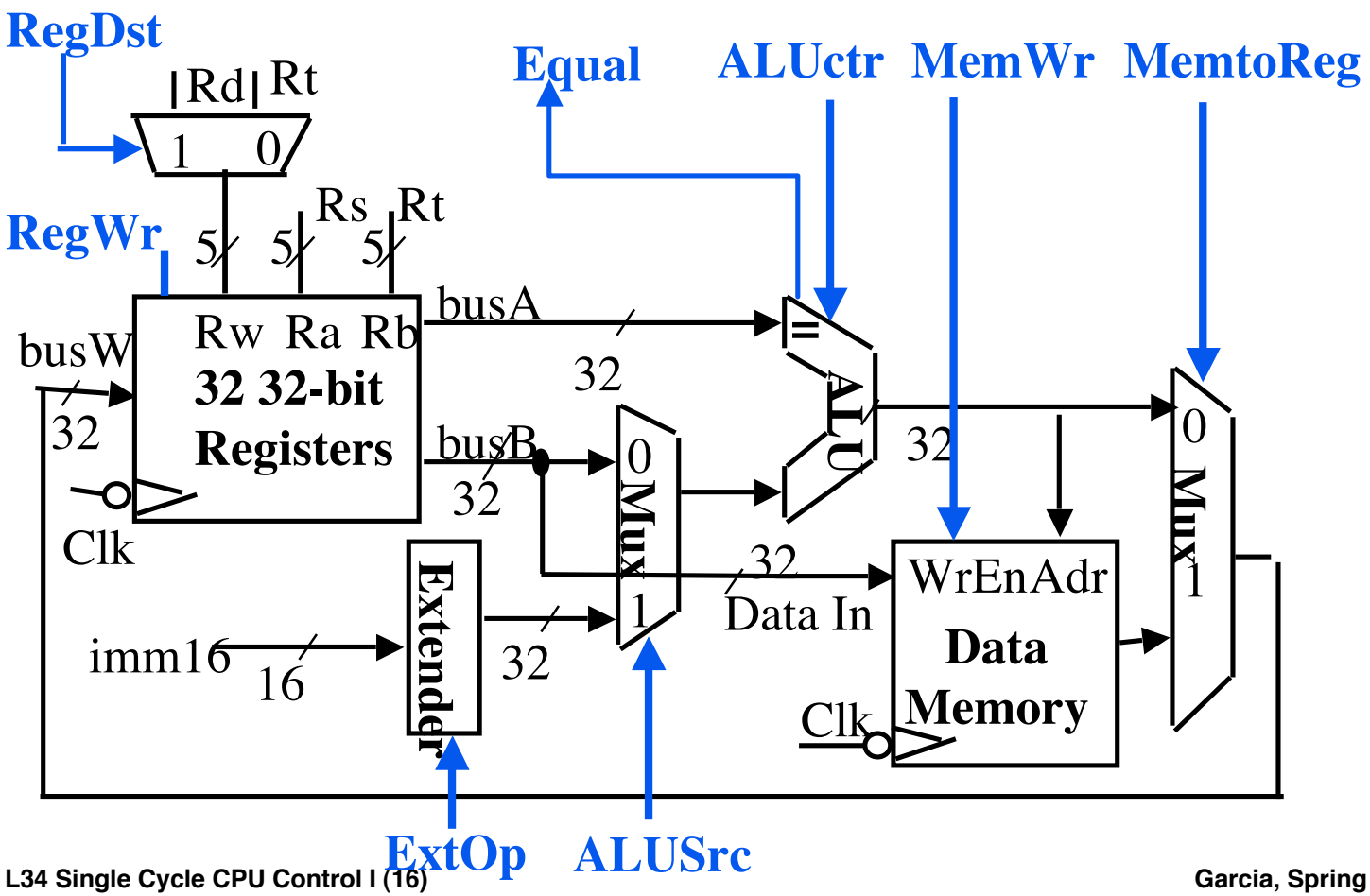
  "n"=next
- Later in lecture: higher-level connection between mux and branch cond



Garcia, Spring 2004 © UCB

Recap: Meaning of the Control Signals

- **ExtOp:** "zero", "sign"
- **ALUsrc:** 0 ⇒ regB; 1 ⇒ immed
- **ALUctr:** "add", "sub", "or"
- **MemWr:** 1 ⇒ write memory
- **MemtoReg:** 0 ⇒ ALU; 1 ⇒ Mem
- **RegDst:** 0 ⇒ "rt"; 1 ⇒ "rd"
- **RegWr:** 1 ⇒ write register



Garcia, Spring 2004 © UCB

Administrivia

- Final Exam will be here! (2050 VLSB)
  - Sat, 2004-05-22, 12:30–3:30pm
- Take EECS Survey this week! (results will be presented to Faculty; you CAN make a difference in your education & undergraduate experience)
  - `http://eecssurvey.berkeley.edu`

Garcia, Spring 2004 © UCB

RTL: The `Add` Instruction

31–26	25–21	20–16	15–11	10–6	5–0
op	rs	rt	rd	shamt	funct
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

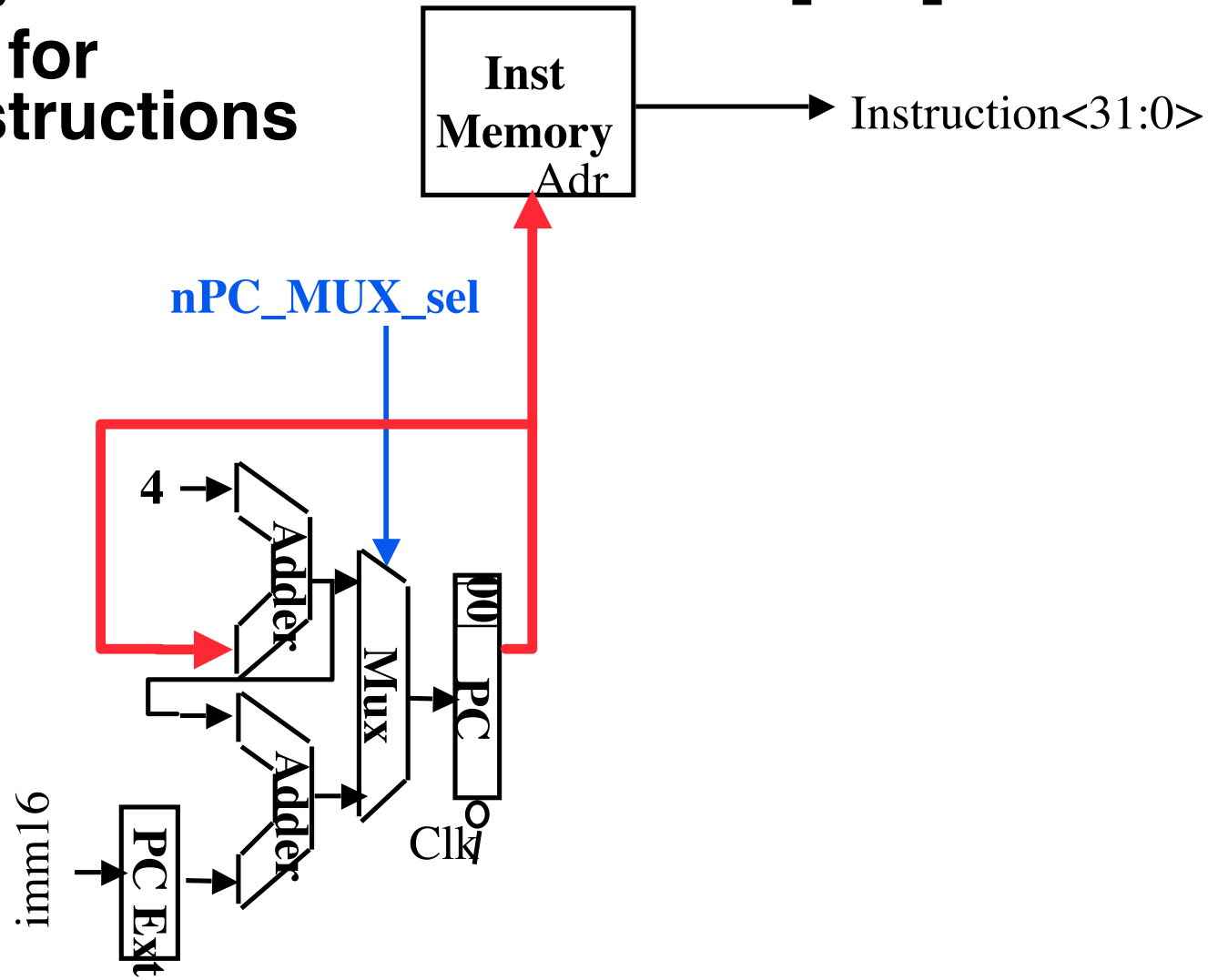
`add rd, rs, rt`

- MEM[PC] — Fetch the instruction from memory
- R[rd] = R[rs] + R[rt] — The actual operation
- PC = PC + 4 — Calculate the next instruction's address

Garcia, Spring 2004 © UCB

Instruction Fetch Unit at the Beginning of `Add`

- **Fetch the instruction from Instruction memory: Instruction = MEM[PC]**
  - **same for all instructions**

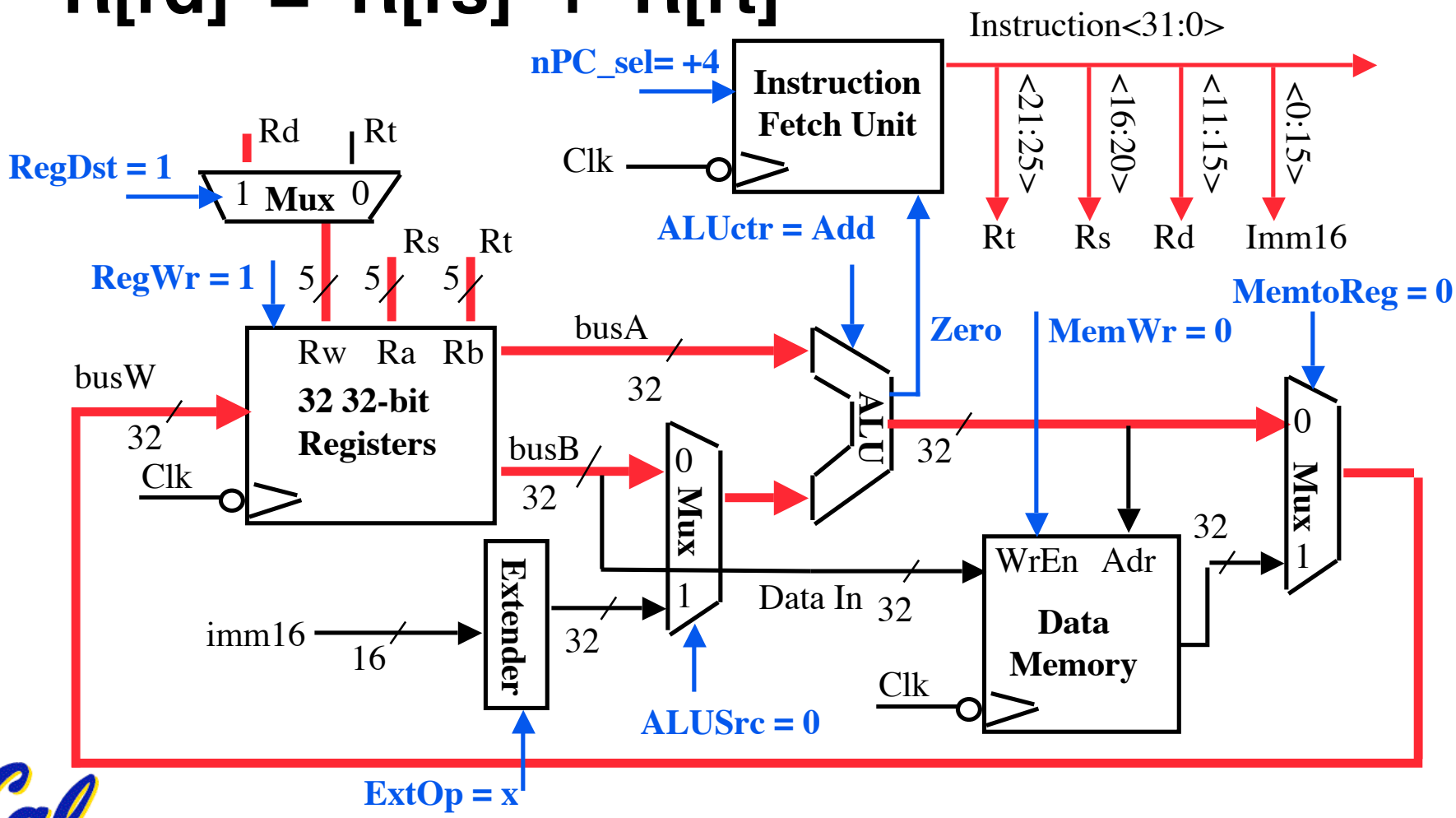


Garcia, Spring 2004 © UCB

The Single Cycle Datapath during Add

31	26	21	16	11	6	0
op	rs	rt	rd	shamt	funct	

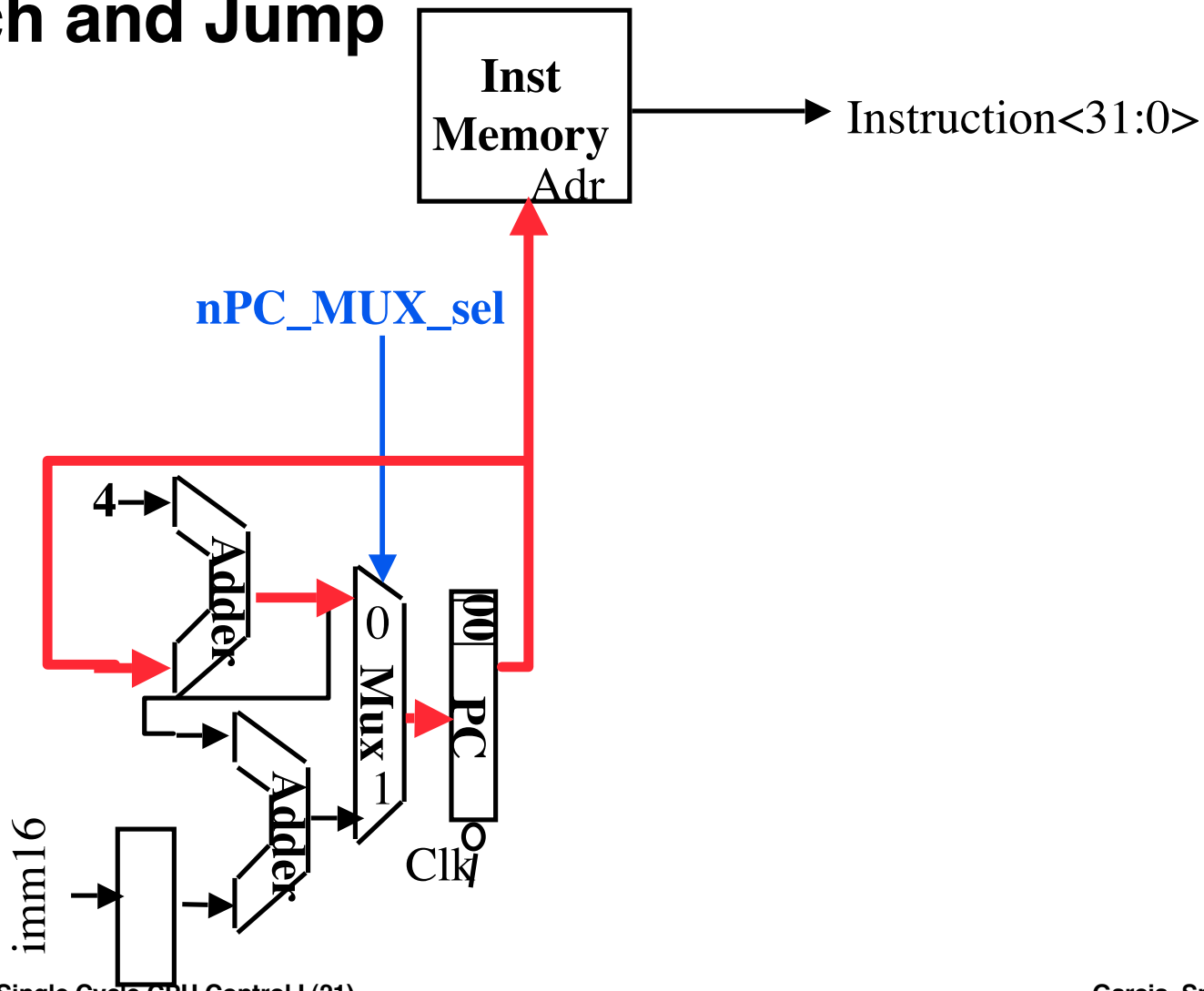
- R[rd] = R[rs] + R[rt]



Garcia, Spring 2004 © UCB

Instruction Fetch Unit at the End of Add

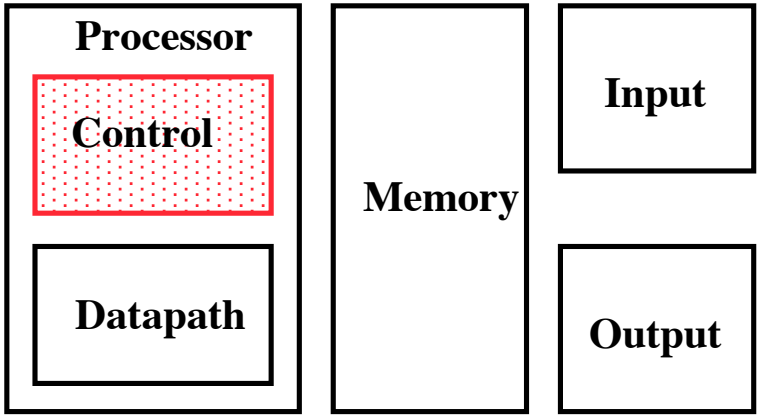
- PC = PC + 4
  - This is the same for all instructions except: Branch and Jump



Garcia, Spring 2004 © UCB

Summary: Single cycle datapath

- 5 steps to design a processor
  - 1. Analyze instruction set => datapath requirements
  - 2. Select set of datapath components & establish clock methodology
  - 3. Assemble datapath meeting the requirements
  - 4. Analyze implementation of each instruction to determine setting of control points that effects the register transfer.
  - 5. Assemble the control logic
- Control is the hard part
- MIPS makes that easier
  - Instructions same size
  - Source registers always in same place
  - Immediates same size, location
  - Operations always on registers/immediates



Garcia, Spring 2004 © UCB