CMPE 200

COMPUTER ARCHITECTURE

Midterm1 Preparation

Bo Yu
Computer Engineering Department
SJSU

Adapted from Computer Organization and Design, 5th Edition, 4th edition, Patterson and Hennessy, MK and Computer Architecture – A Quantitative Approach, 4th edition, Patterson and Hennessy, MK

Midterm 1 Preparation

1.5 Three processors P1, P2, P3 executing same instruction set. P1 has a 3GHz clock and a CPI of 1.5, P2 2.5GHz and CPI1.0, P3 4.0GHz and CPI 2.2.

a. Which processor has the highest performance (instructions/sec)?

Performance of P1 (instructions/sec) = $3 \times 10^9 / 1.5 = 2 \times 10^9$
Performance of P2 (instructions/sec) = $2.5 \times 10^9 / 1.0 = 2.5 \times 10^9$
Performance of P3 (instructions/sec) = $4 \times 10^9 / 2.2 = 1.8 \times 10^9$

b. Find the no. of cycles and instructions for each processor executing a program in 10secs.

No. cycles(P1) = $10 \times 3 \times 10^9 = 30 \times 10^9$, No. insts(P1) = $30 \times 10^9 / 1.5 = 20 \times 10^9$
No. cycles(P2) = $10 \times 2.5 \times 10^9 = 25 \times 10^9$, No. insts(P2) = $25 \times 10^9 / 1 = 25 \times 10^9$
No. cycles(P3) = $10 \times 4 \times 10^9 = 40 \times 10^9$, No. insts(P3) = $40 \times 10^9 / 2.2 = 18.18 \times 10^9$

c. Reducing execution time by 30% leads to an increase of 20% in CPI. What clock rate should we have to get this execution time reduction?

f = No. instr. x CPI / (execution time), then
f(P1) = $3.0 \times 10^9 \times 1.2 / 0.7 = 5.14$ GHz
f(P2) = $2.5 \times 10^9 \times 1.2 / 0.7 = 4.29$ GHz
f(P3) = $4.0 \times 10^9 \times 1.2 / 0.7 = 6.86$ GHz

Midterm 1 Preparation

MIPS Operands and Instructions

- MIPS Instructions are encoded in binary

Instruction	Format	op	rs	rt	rd	shamt	funct	address
add	R	0	reg	reg	reg	0	32_{ten}	n.a.
sub (subtract)	R	0	reg	reg	reg	0	34_{ten}	n.a.
add immediate	I	8_{ten}	reg	reg	n.a.	n.a.	n.a.	constant
lw (load word)	I	35_{ten}	reg	reg	n.a.	n.a.	n.a.	address
sw (store word)	I	43_{ten}	reg	reg	n.a.	n.a.	n.a.	address

- MIPS 32 General Purpose Registers

Register Number	Conventional Name	Usage
$0	$zero	Hard-wired 0
$1	$at	Reserved by assembler to handle large constants
$2-$3	$v0, $v1	Return values from functions
$4 - $7	$a0 - $a3	Arguments to functions, not preserved by subprograms
$8 - $15	$t0 - $t7	Temporary data, not preserved by subprograms
$16 - $23	$s0 - $s7	Saved registers, preserved by subprograms
$24 - $25	$t8 - $t9	More temporary data, not preserved by subprograms
$26 - $27	$k0 - $k1	Reserved by kernel. Do not use.
$28	$gp	Global Area Pointer (base of global data segment)
$29	$sp	Stack Pointer
$30	$fp	Frame Pointer
$31	$ra	Return Address

- R-Format Example

op	rs	rt	rd	shamt	funct
6 bits	5 bits	5 bits	5 bits	5 bits	6 bits

```
add $t0, $s1, $s2
```

special	$s1	$s2	$t0	0	add
0	17	18	8	0	32
000000	10001	10010	01000	00000	100000

$$00000010001100100100000000100000_2 = 02324020_{16}$$

Midterm 1 Preparation

2.8 Translate 0xabcdef12 into binary

0xabcdef12 = 1010 1011 1100 1101 1110 1111 0001 0010

2.9 Translate C code to MIPS. Assume variables f, g, h, i, j are assigned to registers $s0-$s4 respectively. Assume the base address of array A and B are in registers $s6 and $s7 respectively. Assume arrays A and B are 4-byte words.

f = A[g]

```
sll $t0, $s1, 2          # $t0 ← 4*g
add $t0, $t0, $s6        # $t0 ← Addr(A[g])
lw $s0, 0($t0)           # f ← A[g]
```

2.10 Translate MIPS code to C. Assume variables f, g, h, I, j are assigned to registers $s0-$s4 respectively. Assume the base address of array A and B are in registers $s6 and $s7 respectively. Assume arrays A and B are 4-byte words.

```
sll $t0, $s0, 2
add $t0, $t0, $s6
lw $s1, 4($t0)
```

g = A[1+f]

Midterm 1 Preparation

2.15 Provide the type and hexadecimal representation of following instruction: sw $t1, 32($t2)

I-type
0xAD490020

2.16 Provide the type, assembly language instruction, and binary representation of instruction described by the following MIPS field:

Op=0, rs=3, rt=2, rd=3, shamt=0, funct=34

R-type
sub $v1, $v1, $v0
0x00621822

2.17 Provide the type, assembly language instruction, and binary representation of instruction described by the following MIPS field:

Op=0x23, rs=2, rt=1, const=0x4

I-type
lw $v0, 4($at)
0x8C220004

Midterm 1 Preparation

2.23 Assume $t0 holds the value 0x00101000. What is the value of $t2 after the following instructions?

```
        slt $t2, $0, $t0
        bne $t2, $0, ELSE
        j DONE
ELSE:   addi $t2, $t2, 2
DONE:
```

$t2=1+2=3

Midterm 1 Preparation

2.47 Assume that for a given program 70% of the executed instructions are arithmetic, 10% are load/store, and 20% are branch.

(1) Given this instruction mix and the assumption that an arithmetic instruction requires 2 cycles, a load/store instruction takes 6 cycles, and a branch instruction takes 3 cycles, find the average CPI.

Average CPI = (0.7 I x2 + 0.1 I x6 + 0.2 Ix3) / (0.7 I + 0.1 I + 0.2 I
= 1.4 + 0.6 + 0.6 = 2.6

(2) For a 25% improvement in performance, how many cycles, on average, may an arithmetic instruction take if load/store and branch instructions are not improved at all?

Execution Time (old) / Execution Time (new) = 1.25
Execution Time = Clock Cycles / Clock Rate = IC x CPI / Clock Rate
2.6 = 1.25 x (aI_Cycle + 0.1 x6 + 0.2 x3), aI_Cycle = 0.88

(3) For a 50% improvement in performance, how many cycles, on average, may an arithmetic instruction take if load/store and branch instructions are not

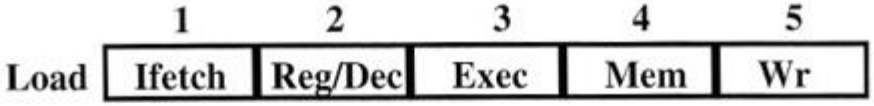
2.6 = 1.5 x (aI_Cycle + 0.1 x6 + 0.2 x3), aI_Cycle = 0.5333

Pipelining Definitions

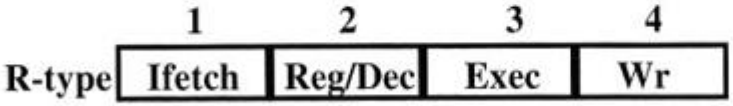
- Pipeline: an implementation technique by which multiple instructions are overlapped in execution. It is not visible to the programmer.
  - ✓ Each stage is called a pipe stage
- Pipeline machine cycle: time required to move an instruction one step down the pipe
- Throughput of a pipeline: number of instructions that can leave the pipeline each cycle
- Latency: the time needed for an instruction to pass through all pipeline stages

Pipelining Important Observation

- Each functional unit can only be used once per instruction
- Each functional unit must be used at the same stage for all instructions:
  - Load uses Register File's Write Port during its 5th stage

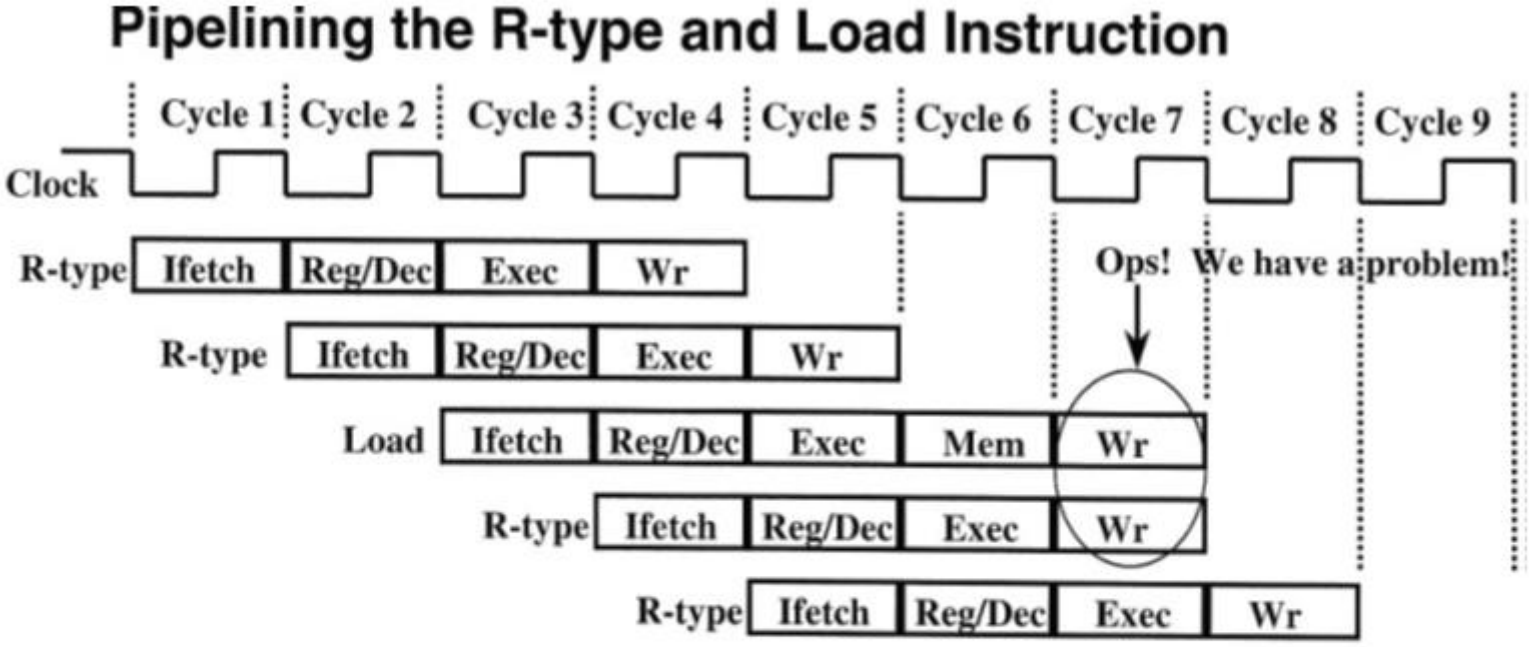


  - R-type uses Register File's Write Port during its 4th stage



- 2 ways to solve this pipeline hazard.

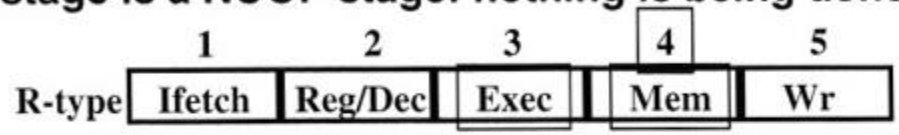
Pipelining Important Observation (resource contention)

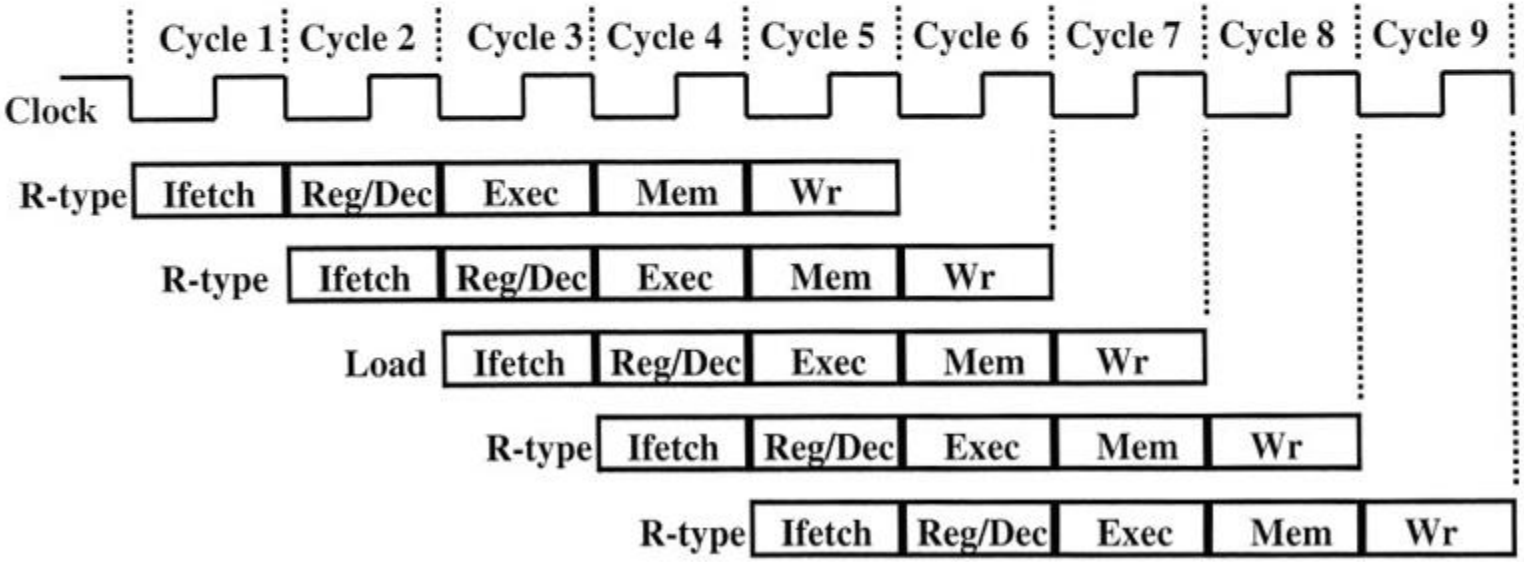


- We have pipeline conflict or structural hazard:
  - Two instructions try to write to the register file at the same time!
  - Only one write port

Solution: Delay R-type's Write by One Cycle

- Delay R-type's register write by one cycle:
  - Now R-type instructions also use Reg File's write port at Stage 5
  - Mem stage is a NOOP stage: nothing is being done.





Midterm 1 Preparation

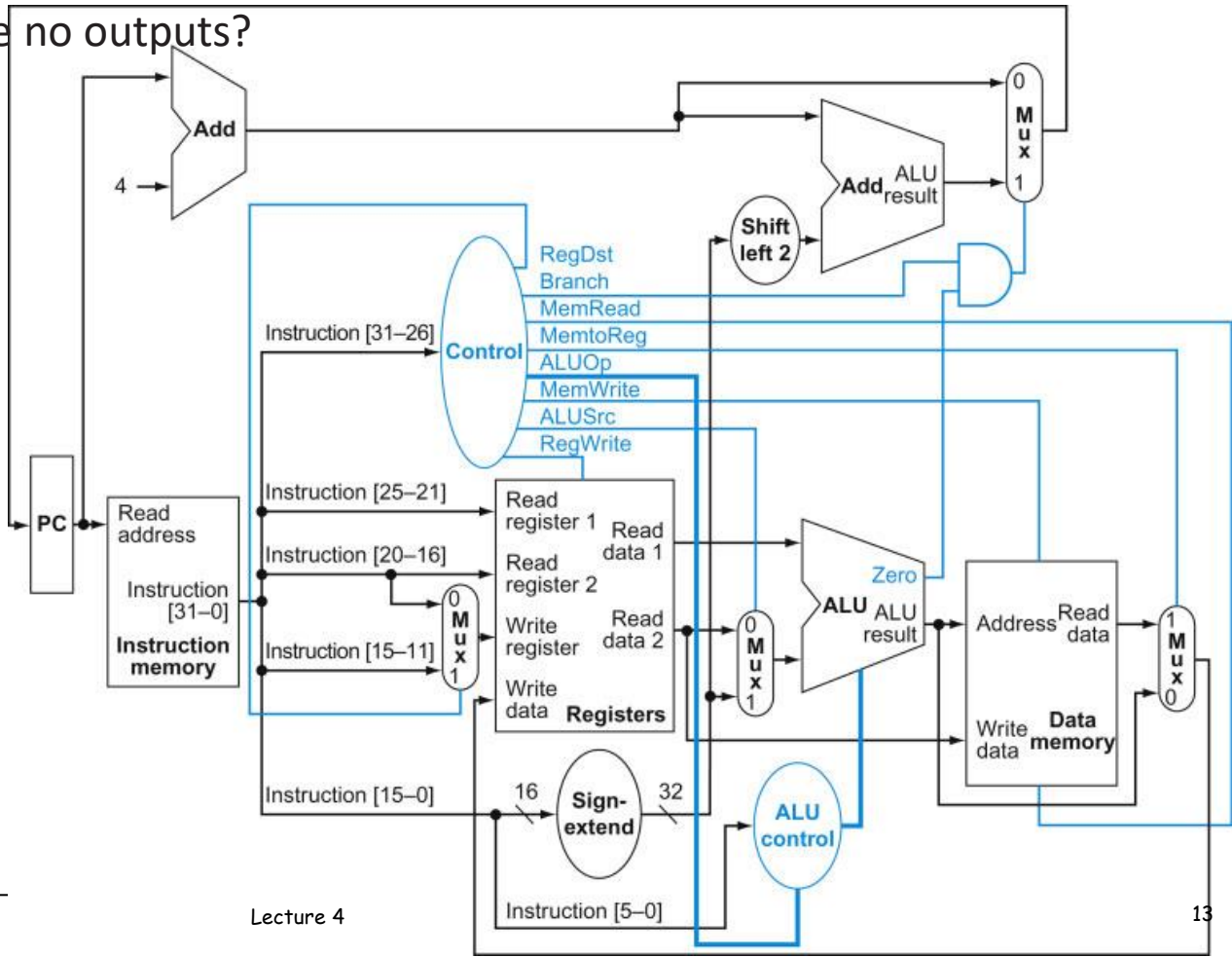
4.1 Consider the following instruction: AND Rd, Rs, Rt

(1) What are the values of control signals generated by the control?

(2) Which resources perform a useful function?

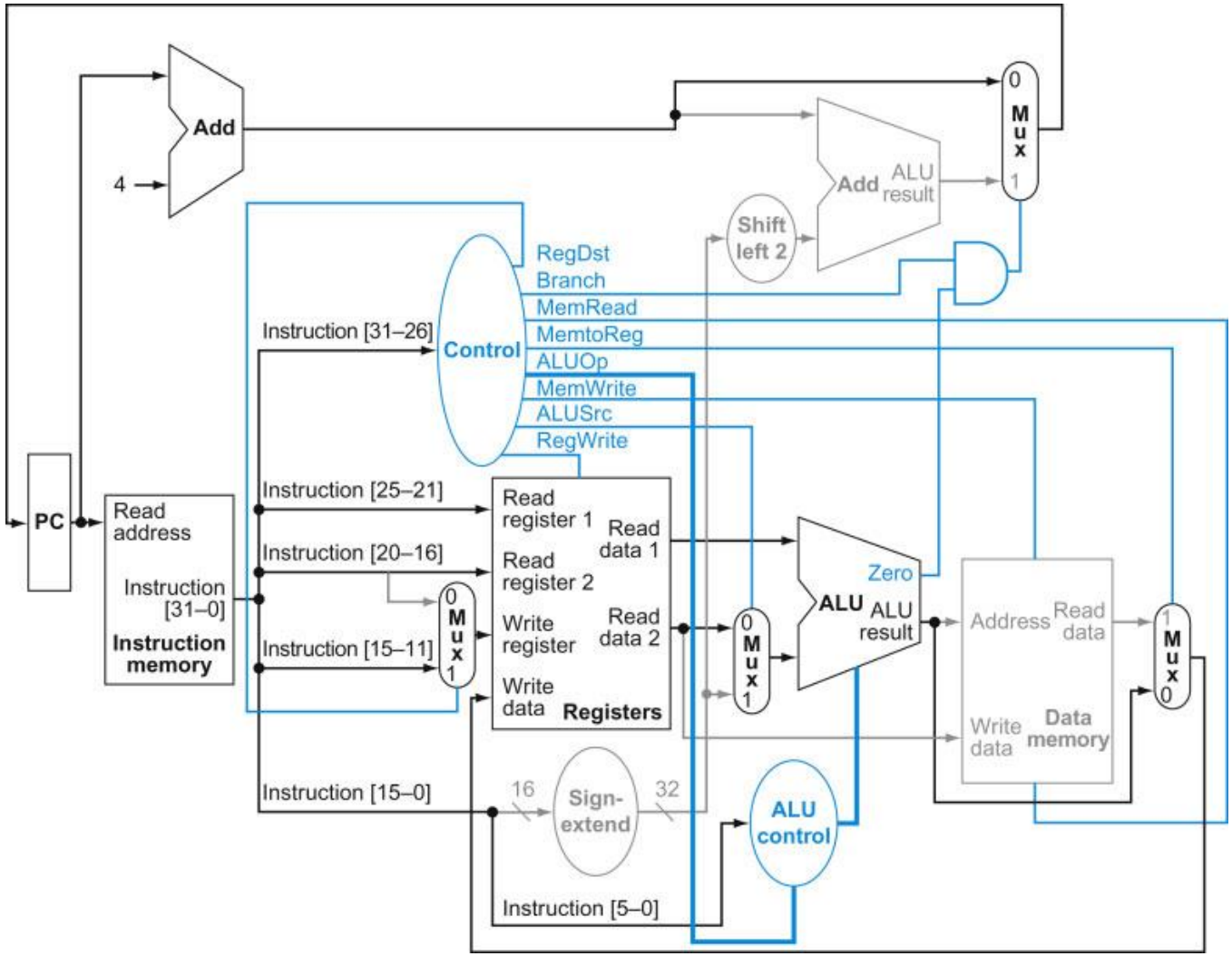
(3) Which resources produce outputs, but their outputs are not used?

(4) Which resources produce no outputs?



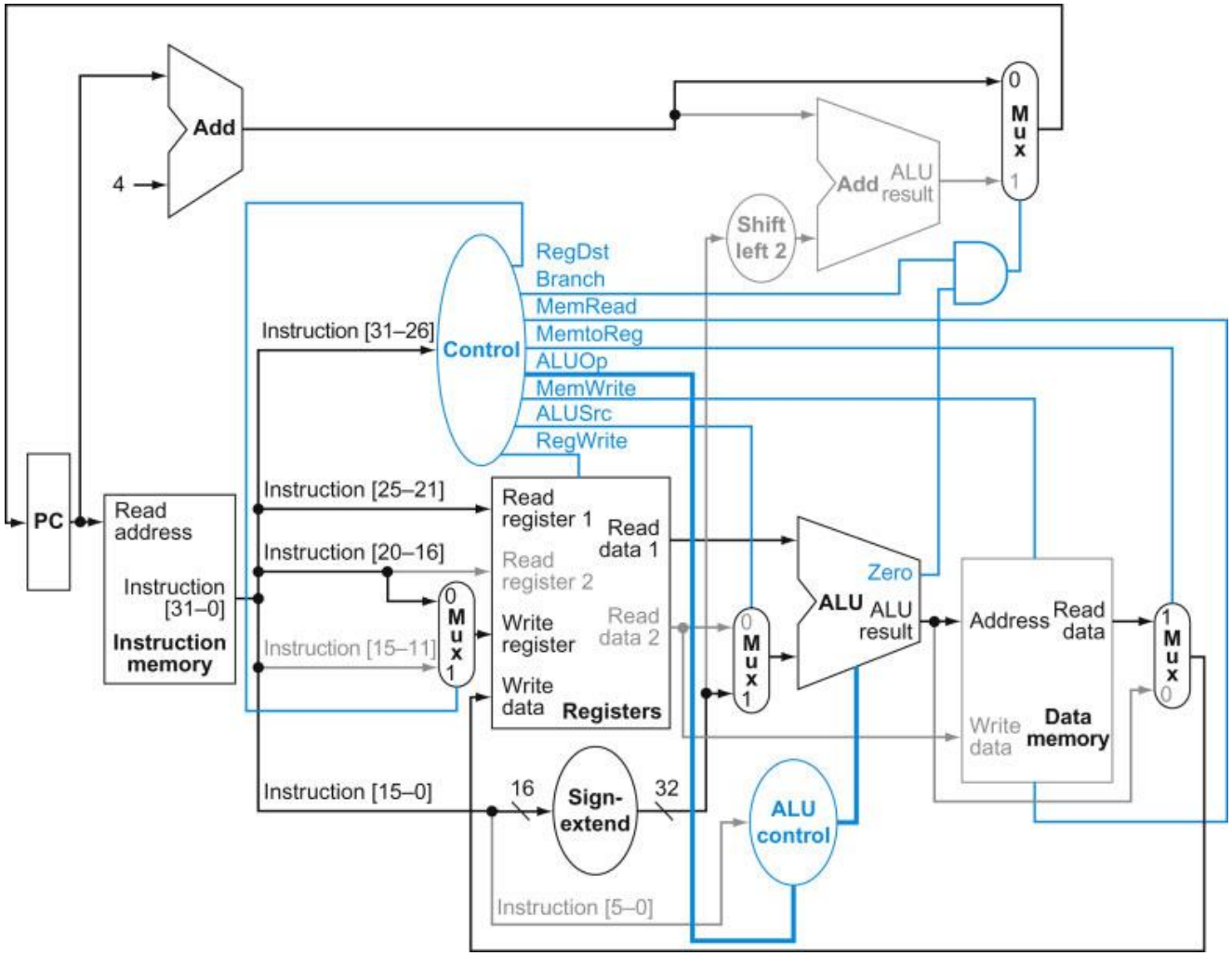
Midterm 1 Preparation

The datapath with control in operation for R-type instruction is shown on the left.
(1) RegWrite=1, ALUSrc=0, MemWrite=0, ALUOp=10, MemtoReg=0, MemRead=0, Branch=0 RegDst=1
(2) All except the branch Add, Sign-extend and Data Memory
(3) None
(4) No outputs: branch Add, Sign-extend and Data Memory



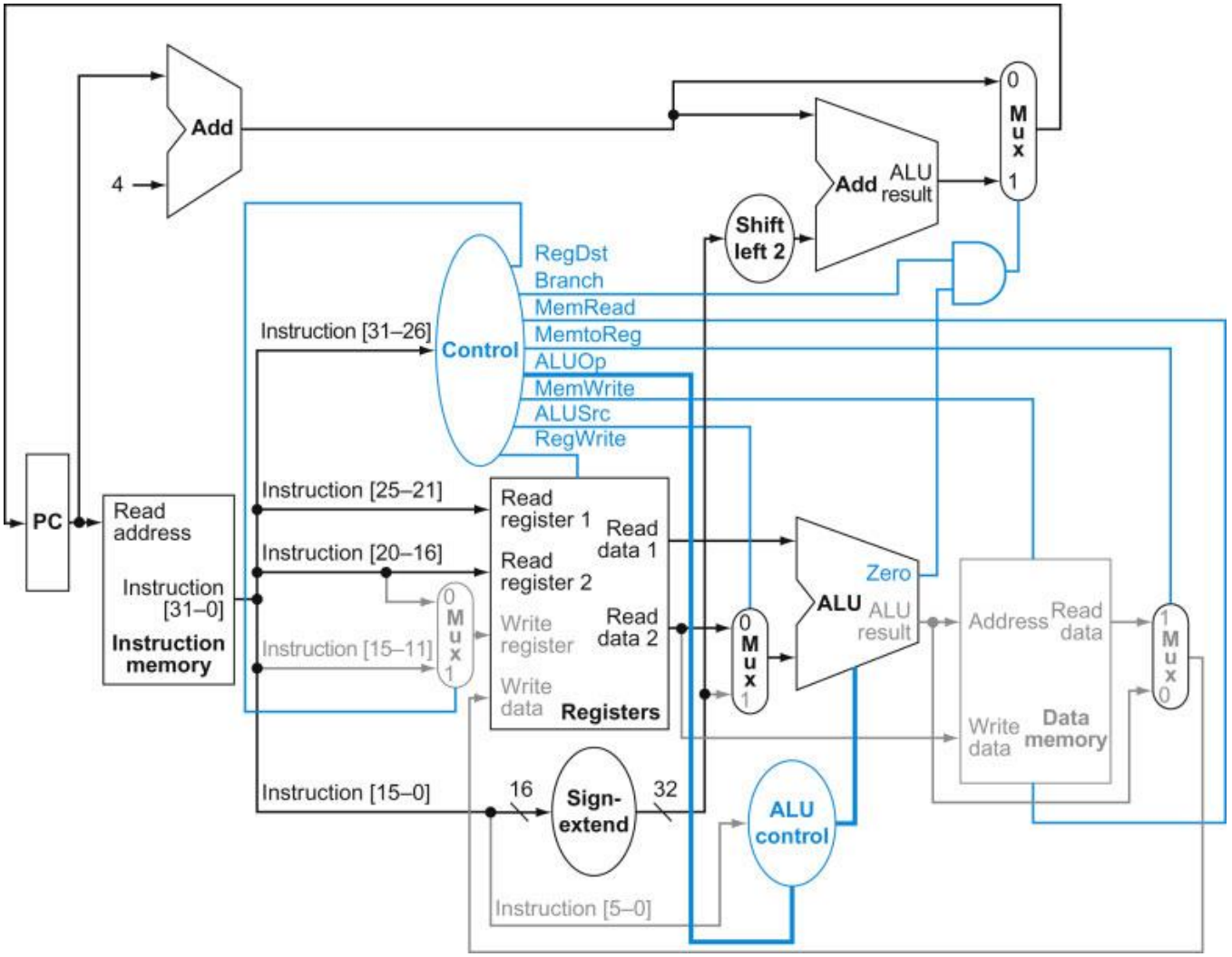
Midterm 1 Preparation

How about: LW $t1, offset($t2)



Midterm 1 Preparation

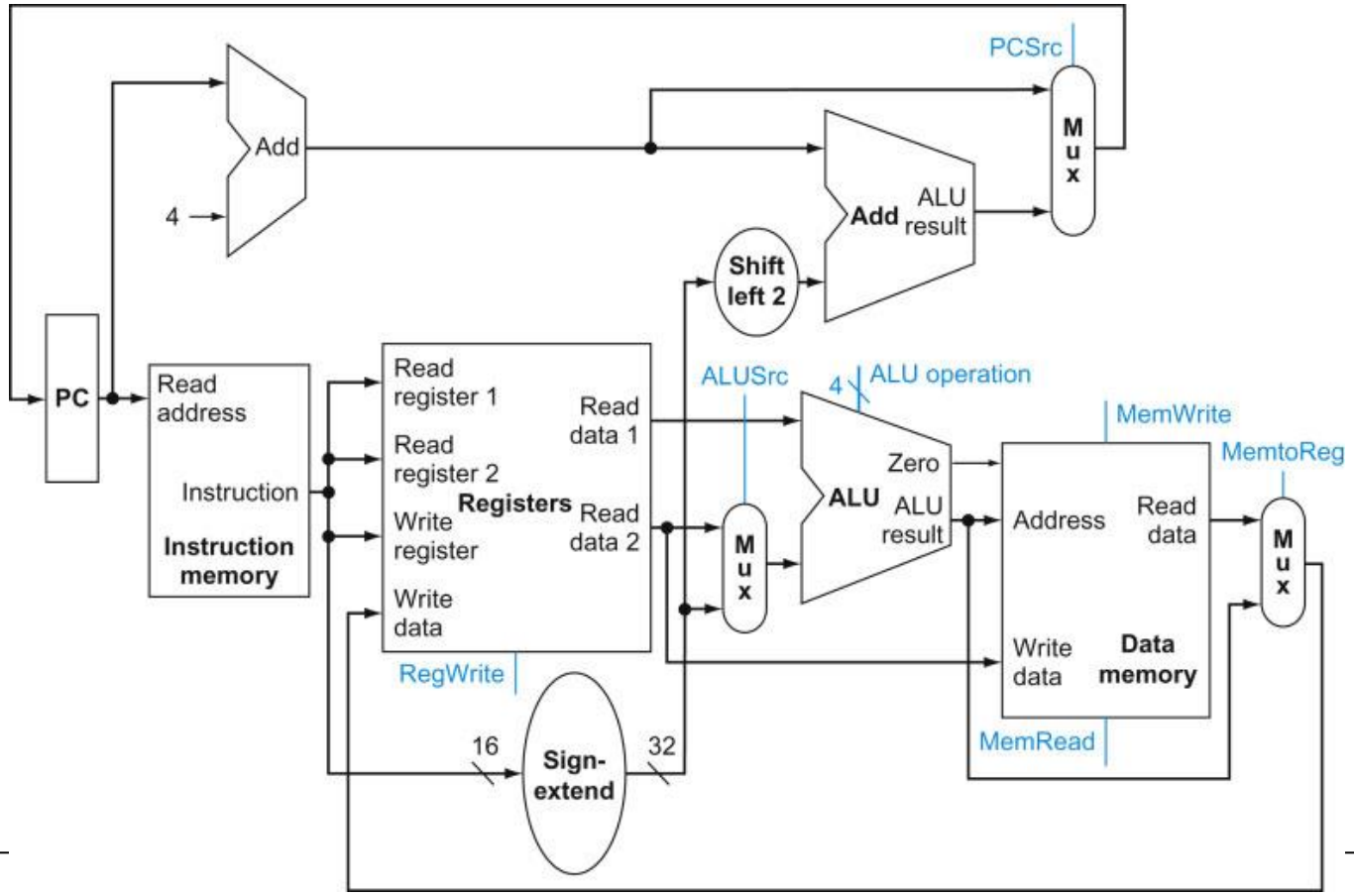
How about: beq $t1, $t2, offset



Midterm 1 Preparation

4.4 Problems in this exercise assume that logic blocks needed to implement a processor's datapath have the following latencies:

I-Mem	Add	Mux	ALU	Regs	D-Mem	Sign-Ext	Shift-left-2
200ps	70ps	20ps	90ps	90ps	250ps	15ps	10ps



Midterm 1 Preparation

I-Mem	Add	Mux	ALU	Regs	D-Mem	Sign-Ext	Shift-left-2
200ps	70ps	20ps	90ps	90ps	250ps	15ps	10ps

(1) What would be the cycle time if processor is only fetching consecutive instructions?

I-Mem takes longer than the Add unit, 200ps.

(2) What would be the cycle time if processor only has unconditional PC-relative branch?

200ps + 15ps + 10ps + 70ps + 20ps = 315ps

(3) What would be the cycle time if processor only needs to support conditional PC-relative branch?

200ps + 90ps + 20ps + 90ps + 20ps = 420ps

(4) Assume we only support bne and add instructions. How does the change in shift-left-2 unit affect the cycle time of processor?

This unit is not on critical path unless it's big enough.
90+90+20=200ps vs. 15+10+70=95ps, Shift-left-2 must be increased by 105ps or more to affect clock cycle time.

Midterm 1 Preparation

4.5 For the problem in this exercise, assume that there are no pipeline stalls and that the reakdown of executed instructions is as follows:

Add	addi	not	beq	lw	sw
20%	20%	0%	25%	25%	10%

(1) In what fraction of all cycles is the data memory used?

The data memory is used by LW and SW instructions, so the answer is: 25% + 10% = 35%

(2) In what fraction of all cycles is the input of the sign-extend circuit needed? What is this circuit doing in cycles in which its input is not needed?

The input of the sign-extend circuit is needed for ADDI (to provide the immediate ALU operand), BEQ (to provide the PC-relative offset), and LW and SW (to provide the offset used in addressing memory) so the answer is: 20%+25%+25%+10%=80%