

**QUESTION-1 (25pts)**

- A. (5 pts) Give the performance equation (ignoring memory system effects) discussed in the class. Explain what each component means.

Execution time = #Instructions \* CPI \* CCT.  
 #Instructions: number of instructions,  
 CPI: cycles per instruction  
 CCT: clock cycle time.

- B. (5 pts) The following measurements have been made using a simulator for a design that is projected to have a clock rate of 1GHz. What is the design's CPI?

Instruction Class	CPI	Frequency
R-type	1	45%
Load	4	25%
Branch	2	20%
Store	3	10%

$$1 \times 45\% + 4 \times 25\% + 2 \times 20\% + 3 \times 10\% \\ 0.45 \quad 1 \quad 0.4 \quad 0.3 \\ = 2.15$$

- C. (5 pts) How does this compare to using branch prediction to shave 1 cycle off the branch latency, but increasing the number of cycles a store operation takes by 1 cycle?

$$1 \times 45\% + 4 \times 25\% + 1 \times 20\% + 4 \times 10\%$$

$$= 2.05$$

- D. (10 pts) Assume that an optimized version of the design has been implemented which doubles the projected clock rate to 2 GHz but also doubles the CPI of each instruction class. Which design is faster, the 1GHz version or the 2 GHz version, and by how much (the 1 GHz version is the one in part-B, not in part-C)?

Same.

## QUESTION-2 (25 pts)

A. (5 pts) Why are pipeline state registers required in a pipelined datapath?

stores the result from all states, avoid of be flushed by next instruction.

B. (10 pts) Consider the following loop instruction sequence:

Loop: add \$3, \$3, \$2

lw \$4, -100(\$3)

beq \$3, \$4, Loop

Suppose this loop executes exactly 3 times (iterations). Further assume that we have 5 execution stages, namely, instruction fetch, reading register file, performing an ALU computation, reading or writing memory, and storing data back to the register file (writeback), and that the clock cycle times for these stages are 4ns, 1ns, 2ns, 4ns, 1ns, in that order. What is the CPI and CCT of the 3 iteration loop in a single-cycle machine?

CPI : 1.

$$\text{CCT} = 4 + 1 + 2 + 4 + 1 \\ = 12 \text{ ns.}$$

C. (10 pts) In part-B, what is the CPI and CCT in 3 iterations in a multi-cycle machine?

CPI = 4.

(4) add \$3 \$3 \$2.

(CCT : 4ns

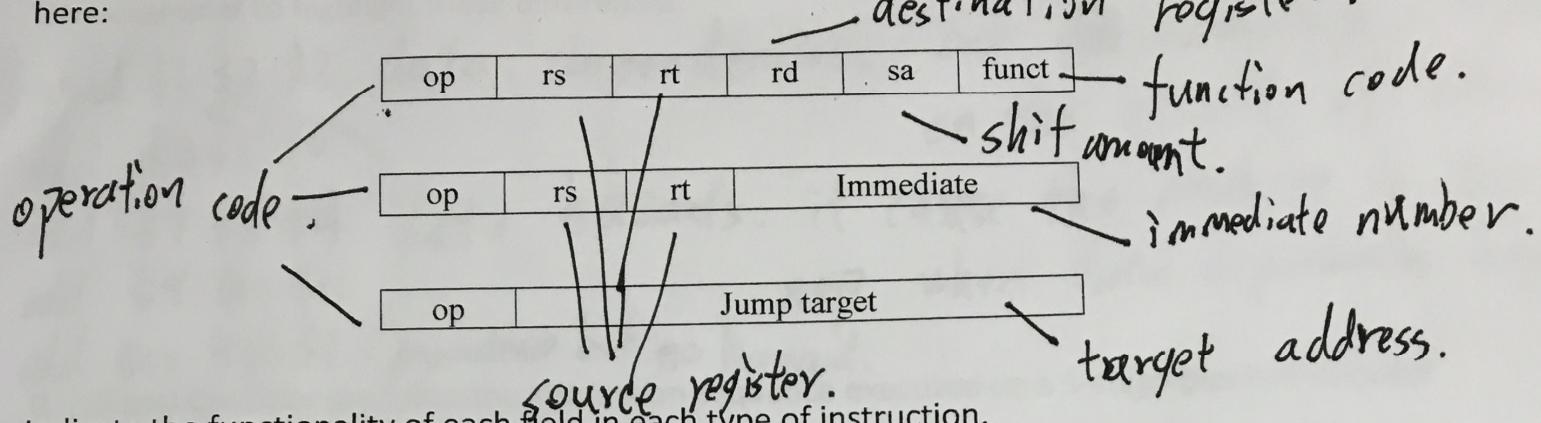
(5) lw \$4, -100(\$3).

(3) beq \$3, \$4, Loop.

$(4+5+3)/3 = 4 \text{ cycles per instruction}$

### QUESTION-3 (25 pts)

A. (10 pts) Recall the MIPS instructions type (R, I, J) we discussed. The 32-bit formats are shown here:



Indicate the functionality of each field in each type of instruction.

B. (5 pts) The IBM PowerPC supports an addressing mode called indexed addressing which adds the values stored in two registers (whose register file addresses are contained in the instruction) together to form the memory address of operand. If this were an addressing mode supported by our MIPS machine, what instructions format (R, I or J) would it be? Explain.

R, has one rd & ~~rs~~ rs, rt.  
one destination register  
& two source register.

C. (10 pts) The individual stages of a pipelined datapath have the latencies and the instruction percentage has the mix as shown in the tables below.

IF	ID	EX	MEM	WB
300ps	300ps	400ps	500ps	200ps

ALU	BEQ	LW	ST
45%	20%	25%	10%

What is the fastest possible clock time (in ps) for a non-pipelined, single-cycle MIPS datapath and for a 5-stage pipelined MIPS datapath, respectively?

non-pipeline :  $300 + 300 + 400 + 500 + 200 = 1700 \text{ ps}$

pipeline :  $\underline{\underline{500 \text{ ps}}}$ .

#### QUESTION-4 (25 pts)

A. (5 pts) What are the fundamental differences between data dependencies and data hazards?

Give examples to highlight those differences.

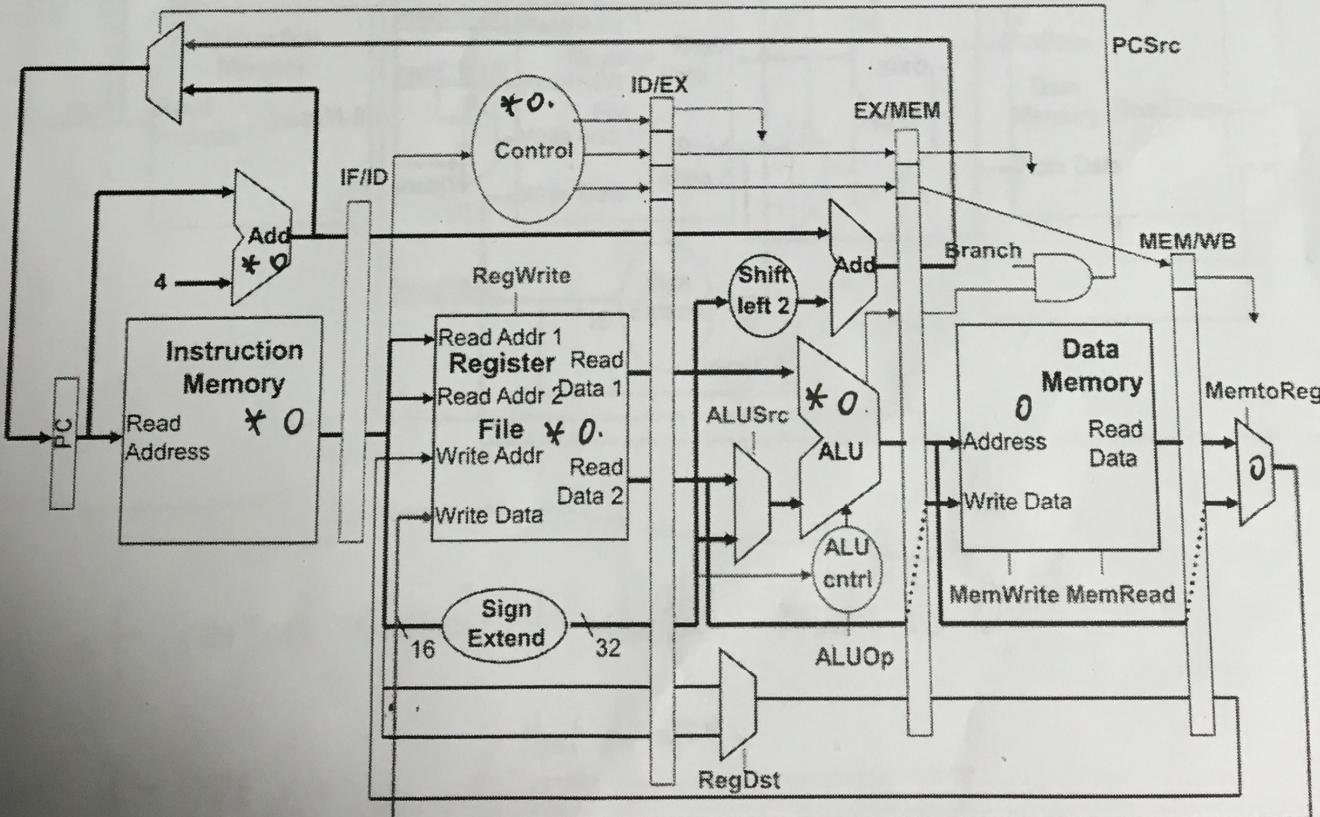
~~# add \$1 \$2 \$3.~~ data dependencies: one ~~instruction~~ instruction is dependent on the result of others.  
~~add. \$4 \$5 \$6~~  
~~add \$7 \$8 \$9~~  
~~add \$9 \$10 \$11~~  
~~add \$12 \$1 \$1~~ dependence but no hazard.

B. (10 pts) Consider the following instruction sequence executed on a 5-stage pipeline in-order machine.

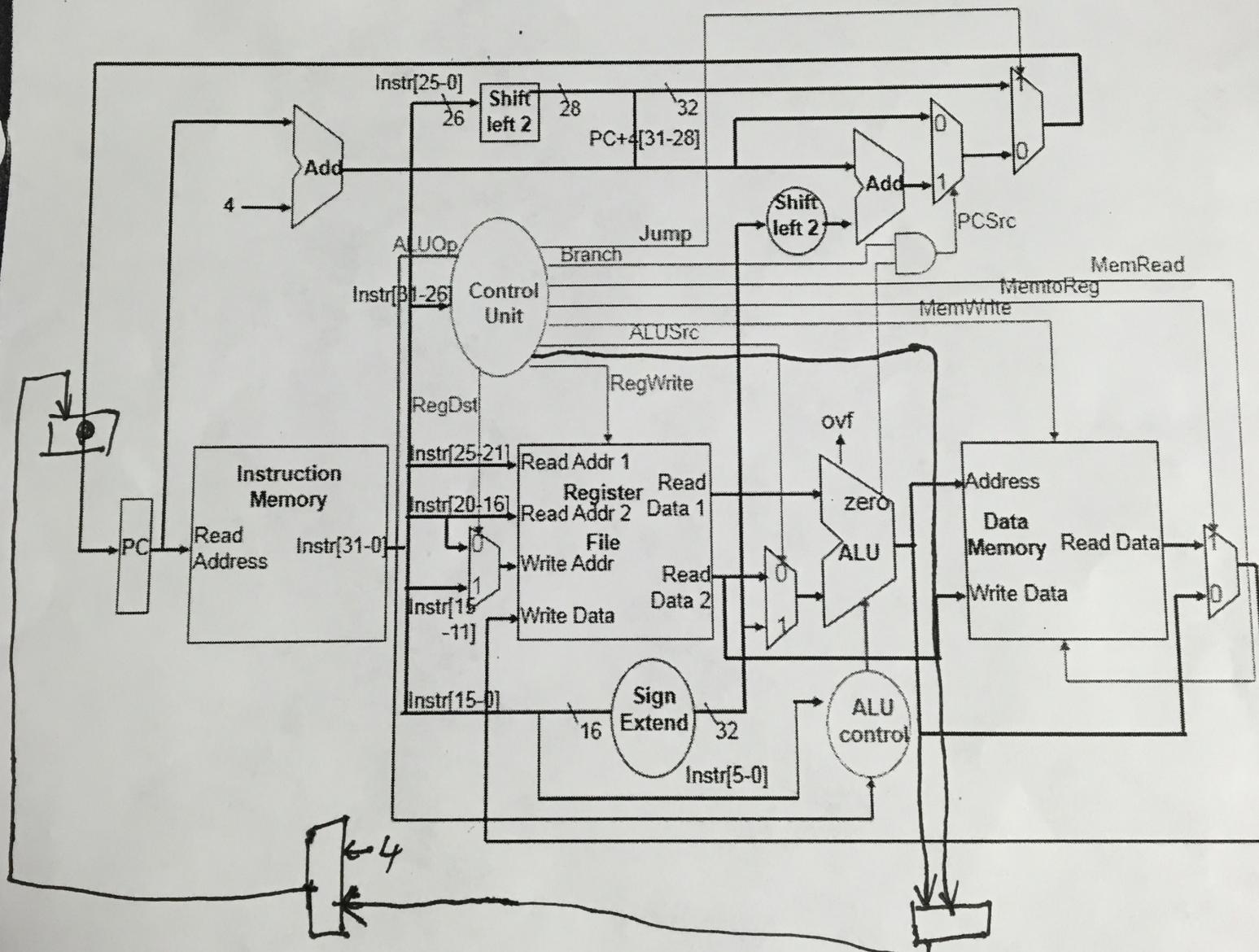
sw \$2 100(\$3)

add \$5, \$6, \$4

Indicate, on the figure below, the components that will be activated to execute lw and add instructions above, using circle (o) and star (\*) to denote lw and add, respectively.



C. (10 pts) Modify the MIPS single cycle datapath shown below to accommodate a new instruction, isz (increment and skip on zero), which increments (by 1) the contents of a register, stores the incremented value back in the register, and skips the next instruction if the result of the incrementing is zero. All other instructions remain unchanged. What new or expanded (in terms of the number of bits) control signals need to be added, if any? Be sure to indicate what instruction format you are using and how the instruction fields are used.



(bit of control signal of whether result is 0.)

use R-type.  
↓ that register.

