CS151B/EE M116C
Computer Systems Architecture
Fall 2017 Sample Midterm Exam I
Instructor: Prof. Lei He

It is a closed-book exam.
There are total NINE pages including this cover page. Check whether you have all pages. If not, let the TA know right now.
Good luck!

Problem 1: _____ of 10 points
Problem 2: _____ of 10points
Problem 3: _____ of 8 points
Problem 4: _____ of 8 points
Problem 5: _____ of 12 points
Problem 6: _____ of 12 points
Total: _____ of 60 points

Problem 1: (total 10 points)

Short problems:

(1) Name five components of a computer, and give an example for each of them (2 points)

Input device: keyboard, mouse, scanner and so on.
Output device: monitor, printer and so on.
Memory: memory, hard disk, CD ROM and so on.
Control: control circuit in CPU
Data-path: ALU in CPU

(2) The most important **interface** between hardware and software in the computer architecture is _____Instruction set architecture (ISA)_____. (1 point)

(3) Target address of conditional branch if it is taken, and target address of jump instruction (3 points. Requirement: specify addresses using bits of instructions)

Conditional branch PC=current PC+4+BranchAddr
BranchAddr is the last 16 bits of the instruction shifted by 2 bits.
Jump : PC(highest 4 bits)=current PC(highest 4 bits)
PC(lowest 28 bits)=JumpAddr
JumpAddr is the last 26 bits of the instruction shifted by 2 bits

(4) Among program, compiler, instruction set, computer organization and technology,

compiler, instruction set, and program can affect CPI. (2 points)

(5) Instruction mix, kernel benchmarks, application benchmarks, and **micro benchmarks** can be used to evaluate the computer performance. **Application benchmark set_** among the above is most reliable. (2 points)

Problem 2 (10 points):

Binary bits have no inherent meaning. Given the bit pattern:

0100 0100 0000 1111 1100 0000 0000 0000

What does it represent, assuming that it is

a. An unsigned integer?

b. What is the smallest and largest floating point numbers that can be represented by a 32-bit word according to IEEE standard 754?

c. A MIPS instruction

a) 230+226+219+218+217+216+215+214=1141882880

b) the range of the positive floating point number is from

$1.00000000000000000000000_{two} \times 2^{-126}$

to

$1.11111111111111111111111_{two} \times 2^{+127}$

c) 0100 01 - 00 000 - 0 1111 - 1100 0 - 000 0000 0000

OP=0x21 → R[rd] = R[rs] + R[rt]
rs = $0, rt = $15 and rd = $24
1001 0100 0000 1111 1100 0100 0000 1000

d) -1810906104

Problem 3 (8 points):

For the following C code,

A = B + C
D = A - C

a) write an equivalent pseudo-assembly language program in architectural styles ***memory-memory*** (format: OP M_1 M_2 M_3, where M_i is address of memory) and ***load-store*****:**

Memory-Memory	Load-Store
ADD A, B, C	Load R1, B
SUB D, A, C	Load R2, C
	ADD R3, R1, R2
	SUB R3, R3, R2
	STORE D R3
(2 points)	(3 points)

b) Now assume that OP Code is 6 bits for both architecture, but memory address takes 24 bits in memory-memory instruction, what are the sizes of the codes? And which architecture is more efficient as measured by code size? (1 point)

Memory-memory is more efficient as measured by code size because it uses fewer instructions. (1 point)

Memory-memory: 2 x (6 + 3x24) / 8 = 19.5 bytes

Load-Store: 5 x 32 / 8 = 20 bytes

c) How many memory accesses are there for each code? Which architecture is more efficient as measured by memory bandwidth requirement? (1 point)

Load-Store architecture is more efficient as measured by memory bandwidth requirement because it can store the middle result in registers not in memory, therefore has fewer memory access. (1 point)

Memory-memory: 3 memory accesses for each line of codes, in total 6 memory access

Load-Store: in total 3 accesses, B C and D.

d) Assume each arithmetic operation uses 4 cycles, and each memory access uses 10 cycles, how many cycles are needed in total for each code. Which architecture is preferred for high performance and why? (1 point)

Load-Store architecture is preferred for high performance because the operations on registers are much faster than those on memory. (1 point)

Memory-memory: 2 x (4 + 3*10) = 78 cycles

Load-store: 4 + 4 + 3*10 = 38 cycles

Problem 4 (8 points):

A computer designed for Java programs can be speed-up by adding hardware support for garbage collection. Garbage collection currently comprises 24% of the cycles of the workload. There are two possible changes.

(1) Automatically handle garbage collection in hardware. This causes an increase in cycle time by a factor of 1.3. How fast is the program after adding automatically handle garbage collection compared to the original one.

For option 1, the total execute time is (1-24%)*1.3 =0.988

(2) Add new instructions for garbage collection to the ISA. This would halve the number of instructions needed for garbage collection but increase the cycle time by 1.1. How fast is the program after adding automatically handle garbage collection compared to the original one.

For option 2, the total execute time is (1-24%/2)*1.1=0.968

(3) Which of the two options, if either, should you choose?

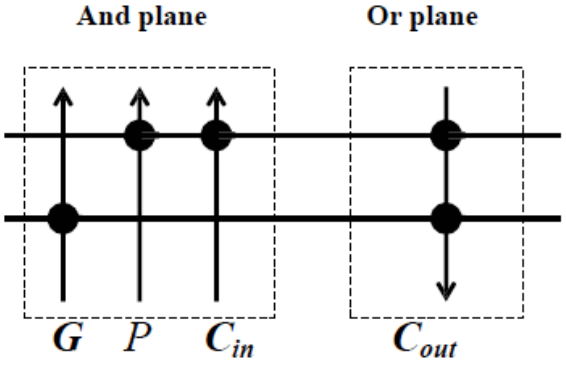
Therefore, option 2 is faster than option 1. Option 2 will be chosen.

Problem 5 (12 points):

1) Truth table

Inputs			Output
G	P	C_{in}	C_{out}
0	0	0	0
0	1	0	0
1	0	0	-
1	1	0	1
0	0	1	0
0	1	1	1
1	0	1	-
1	1	1	1

$C_{out} = G + P \cdot C_{in}$ (2 points for equation and 2points for truth table)



2)

$C_4 = g_{[0,3]} + p_{[0,3]} \cdot C_0$

$C_8 = g_{[4,7]} + p_{[4,7]} \cdot C_4$

$C_{12} = g_{[8,11]} + p_{[8,11]} \cdot C_8$

$G_{[0,15]} = G[12,15] +$

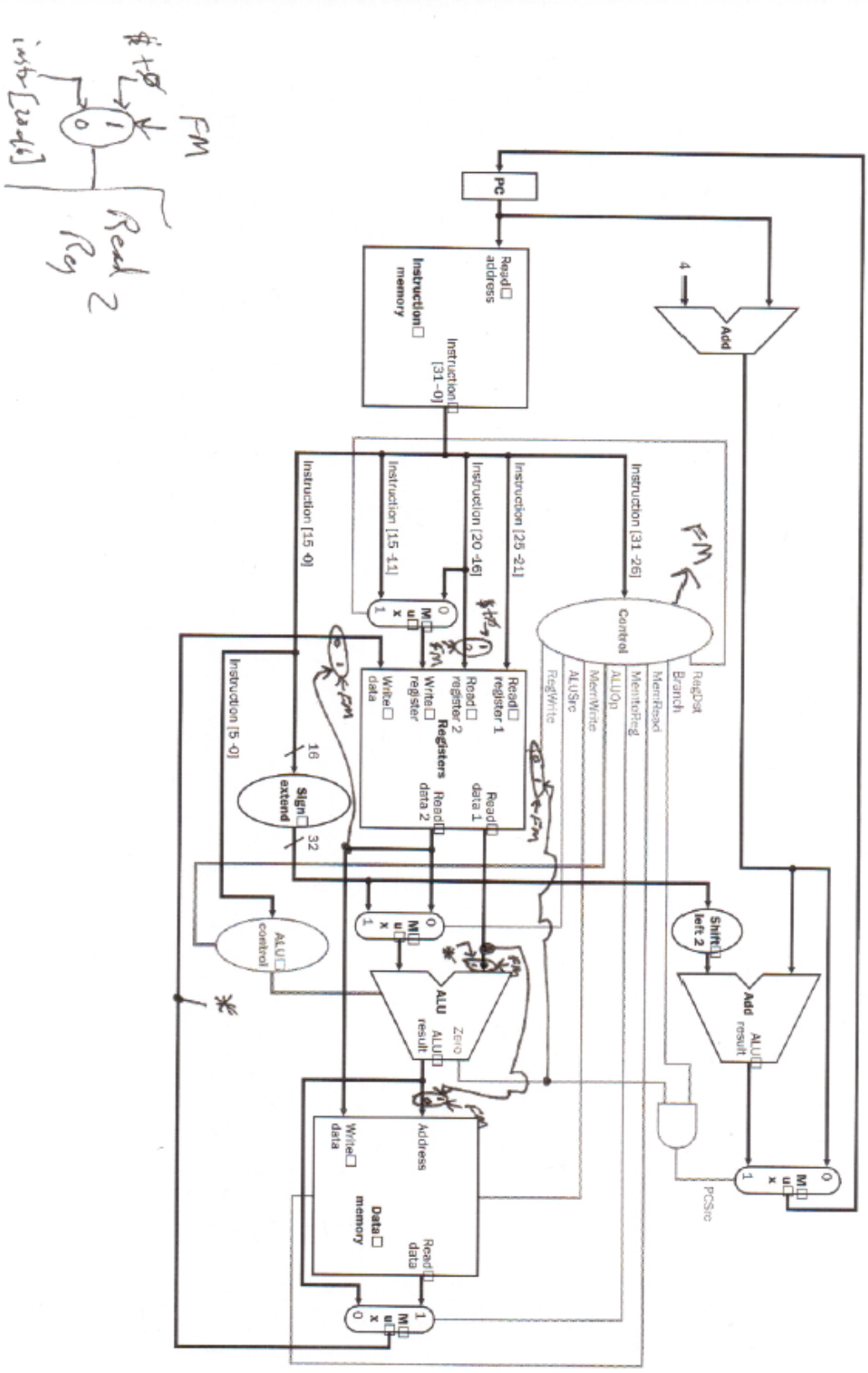
$P[12,15].G[8,11] +$

$P[12,15].P[8,11].G[4,7] +$

$P[12,15]P[8,11]P[4,7]G[0,3]$

$P_{[0,15]} = P[15,12].\ P[8,11].\ P[4,7].\ P[0,3]$

Problem 6 (12 points):



Main Controller

Input or Output	Signal Name	R-format	lw	sw	Beq	Fmov
Inputs	Op5	0	1	1	0	U
	Op4	0	0	0	0	N
	Op3	0	0	1	0	I
	Op2	0	0	0	1	Q
	Op1	0	1	1	0	U
	Op0	0	1	1	0	E
Outputs	RegDst	1	0	X	X	0
	ALUSrc	0	1	1	0	1
	MemtoReg	0	1	X	X	X
	RegWrite	1	1	0	0	X
	MemRead	0	1	0	0	1
	MemWrite	0	0	1	0	0
	Branch	0	0	0	1	0
	ALUOp1	1	0	0	0	0
	ALUOp0	0	0	0	1	1
	FM	0	0	0	0	1

ALU Controller

Opcode	**ALUOp**	**instruction**	**function**	**ALU Action**	**ALUCtrl**
Lw	00	load word	XXXXXX	add	010
Sw	00	store word	XXXXXX	add	010
Beq	01	branch equal	XXXXXX	subtract	110
R-type	10	add	100000	add	010
R-type	10	subtract	100010	subtract	110
R-type	10	AND	100100	AND	000
R-type	10	OR	100101	OR	001
R-type	10	SLT	101010	SLT	111