CS151B/EE M116C
Computer Systems Architecture
Spring 2019 Midterm Exam I
Instructor: Prof. Lei He

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

Problem 1: _____ of 14 points
Problem 2: _____ of 8 points
Problem 3: _____ of 9 points
Problem 4: _____ of 5 points
Problem 5: _____ of 12 points
Problem 6: _____ of 12 points
Total: _____ of 60 points

Problem 1: (total 14 points)

(1) Point out at least two differences between CISC (Complex Instruction Set Computer) and RISC (Reduced Instruction Set Computer). (2 points)

1. **Fixed/flexible instruction size**
2. **large/small # instructions**
3. **Specific/general-purpose instructions**
4. **Any reasonable answers**

(2) Explain the concept of *power wall* in the context of computer architecture? (3 point)

Heat = capacitor density * frequency * voltage^2.
Heat should be all dissipated.
Increasing density and frequency lead to more heat. (voltage cannot be decreasing forever)
Dissipating of heat becomes a barrier for higher performance.

(3) Explain the operations of registers and the stack when calling a subroutine in MIPS? (3 point)

Save saved registers to stack.
Put arguments of the subroutine into argument registers.
Save the next instruction address to ra, and jump.
Subroutine functions.
Save return values to registers v0-v3.
Restore saved registers from the stack.
Fetch ra and jump back.

(For details refer to discussion session slides of week 2)

(4) Show how the Booth's algorithm works with $(00101)_2$*$(00111)_2$. Assume the most significant bit is the sign, and there is no overflow happening. (3 point)

Show correct shift/add/sub operations.

(5) True/False: circle the correct answer, and explain in short. (1 point each)

T F 1. Branch instructions in MIPS can only jump forward 32768 and backward 32767 instructions.
True.
16 bits signed integer for jumping forward and backward.
PC+4+immediate<<2

T F 2. A carry-out at the most significant bit after an addition of two signed numbers always indicates overflow.
False.

T F 3. Program does not effect the average CPI of a CPU.
False.

Problem 2 (10 points):

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

10111001

What does it represent, assuming that it is

a. An unsigned integer?

b. A 2's complement?

a) (2 points)
185

b) (2 points)
-71

We are defining 8-bit floating-point precision, with the following format:

Sign (1 bit)	Exponent (3 bits)	Fraction (4 bits)

Assuming that it follows the same philosophy as single and double precision defined by IEEE 754 standard.

c) (2 points) What should be the bias for this 3-bit exponent? Leave your answer in decimal.
3

d) (2 points) What is the binary representation of the smallest float, which is strictly larger than 1? What are its values in binary and decimal?
00110001
1 + 2^-4 = 1.0625

Problem 3 (9 points):

For the following statements, circle how the proposed change will impact CPI, # of instructions executed, and cycle time (i.e. an answer might be that CPI will increase, # of instructions executed will remain the same, and cycle time MAY decrease). Consider the multi-cycle data path. **Provide a brief justification of each answer** and make sure you address each of the three components by clearly circling ONE option from each group.

Loads can only use the base addressing mode – the base+offset addressing mode is completely removed from the ISA. Any load that requires base+offset addressing will need to use an add instruction before the load to perform the addition.

All reasonable solutions with correct explanations are correct.

a) (3 points) CPI **will** / may increase / **decrease** / stay the same.
Every load will require less cycles.

b) (3 points) # instructions executed **will** / may **increase** / decrease / stay the same
More instructions for loading.

c) (3 points) cycle time **will** / may increase / decrease / **stay the same**
Cycle time is determined by the max(memory access time, ALU time, register file access time).

Problem 4 (5 points):

Assume i and j are assigned to $s0 and $s1. The base address of the array A and B are in registers $s2 and $s3 respectively. Convert the following C code to MIPS code.

C code:

B[7] = A[i] + A[j];

Solution 1 (assume no shift left will be done by lw/sw)

```
sll $t0, $s0, 2          # word to byte i=i*4
add $t0, $t0, $s2        # address of A[i]
lw $t2, 0($t0)           # load A[i]
sll $t1, $s1, 2          # j=j*4
add $t1, $t1, $s2        # address of A[j]
lw $t3, 0($t1)           # load A[j]
add $t4, $t2, $t3        # temp = A[i]+A[j]
sw $t4, 28($s3)          # save temp to B[7]
```

Solution 2 (assume shift left will be done by lw/sw)

```
srl $s2, $s2, 2
add $t0, $s0, $s2
lw $t2, 0($t0)
srl $s3, $s3, 2
add $t1, $s1, $s3
lw $t3, 0($t1)
add $t4, $t2, $t3
sw $t4, 7($s3)
```

Problem 5 (12 points):

Consider the application A and the baseline MIPS processor with the following Instruction cycles. Suppose application A executes three billion instructions. Answer the question below, and explain your work.

Instruction	% of Instructions in *A*	Instruction Latency (cycles)
Load	20%	5
Store	10%	4
Simple R-type (i.e. adds, ands, shifts)	45%	4
Multiply	10%	5
BEQ/BNE	10%	3
Jump	5%	3

Suppose processor has different clock cycles for different instructions.

(1) What is the CPI for application A?

0.2 * 5 + .01 * 4 + 0.45 * 4 + 0.1 * 5 + 0.1 * 3 + 0.05 * 3 = 4.15

(2) The hardware running the application has a cycle time of 300ps. What is the execution time for that hardware to run the application.

3 * 10^9 * 4.15 * 300 * 10^-12 = 3.735 (s)

(3) Now suppose we add an instruction into ISA, which does a multiply+accumulate. One multiply+accumulate operation can replace one multiply instruction and one add instruction. It will cost 7 cycles. With this optimization, the compiler replaces 50% of the multiplications (and the corresponding number of additions) in the original program, with the multiply+accumulate operations. What is the new CPI?

Assume there were 100 instructions before the modification. Now there are 100-5-5+5=95 after the modification.
20 / 95 * 5 + 10 / 95 * 4 + 40 / 95 * 4 + 5/ 95 * 5 + 10 / 95 * 3 + 5 / 95 * 3 + 5 / 95 * 7= 4.26
(4) What is the execution time for the same hardware from part (2) to run the new application from part (c)?

0.95 * 3 * 10^9 * 4.26 * 300 * 10^-12 = 3.64 (s)

Problem 6 (12 points):

Carry Look-Ahead adder:
Given the following truth table for a full adder:

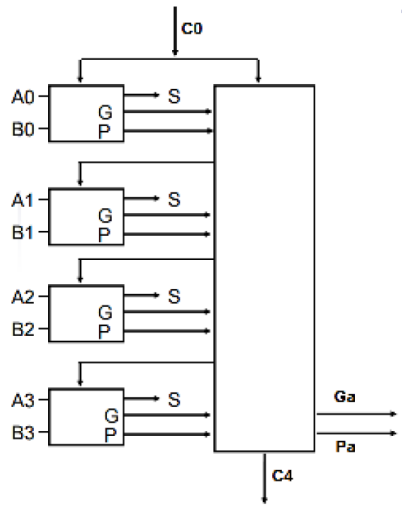
Inputs			Outputs	
A	B	C_{in}	S	C_{out}
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

The G (generate) can be defined as $A*B$ and P (propagate) can be defined as $A+B$.

(1) Write out the expression for S and Cout given A, B, and Cin for a 1-bit adder with truth table as shown above.

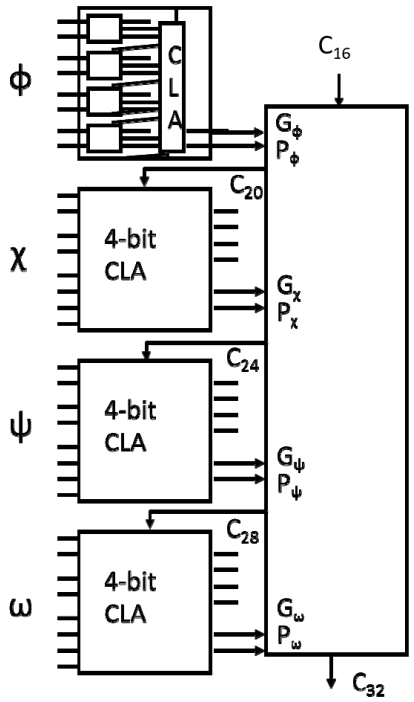
S= Cin xor (A xor B)
Cout = (AB) + (ACin) + (BCin)

(2) Write out the expression for C4, Ga, and Pa with given C0, Ai, and Bi as shown in figure below, which is a 4-bit Carry Lookahead Adder (CLA).



Pi = Ai+Bi
Gi = AiBi
Ga = G0*P1*P2*P3 + G1*P2*P3 + G2*P3 + G3
Pa = P0*P1*P2*P3
C4 = Ga + C0 * Pa

(3) A 16-bit Hierarchical CLA can be built by four 4-bit CLA's with the way as shown in figure. Write out the expression for C20, C24, C28 and C32, when given $G\phi, G\chi, G\psi, G\omega, P\phi, P\chi, P\psi, P\omega$ and C16.



16-bit HCLA

$C20 = G\phi + P\phi \cdot C16$

$C24 = Gx + Px \cdot G\phi + Px \cdot P\phi \cdot C16$

$C28 = G\psi + P\psi \cdot Gx + P\psi \cdot Px \cdot G\phi + P\psi \cdot Px \cdot P\phi \cdot C16$

$C32 = Gw + Pw \cdot G\psi + Px \cdot P\psi \cdot Gx + Pw \cdot P\psi \cdot Px \cdot G\phi + Pw \cdot P\phi \cdot Px \cdot P\phi \cdot C16$