CSC390 (Computer Org. & Arch.)
Final Exam- Spring 2016

Name:

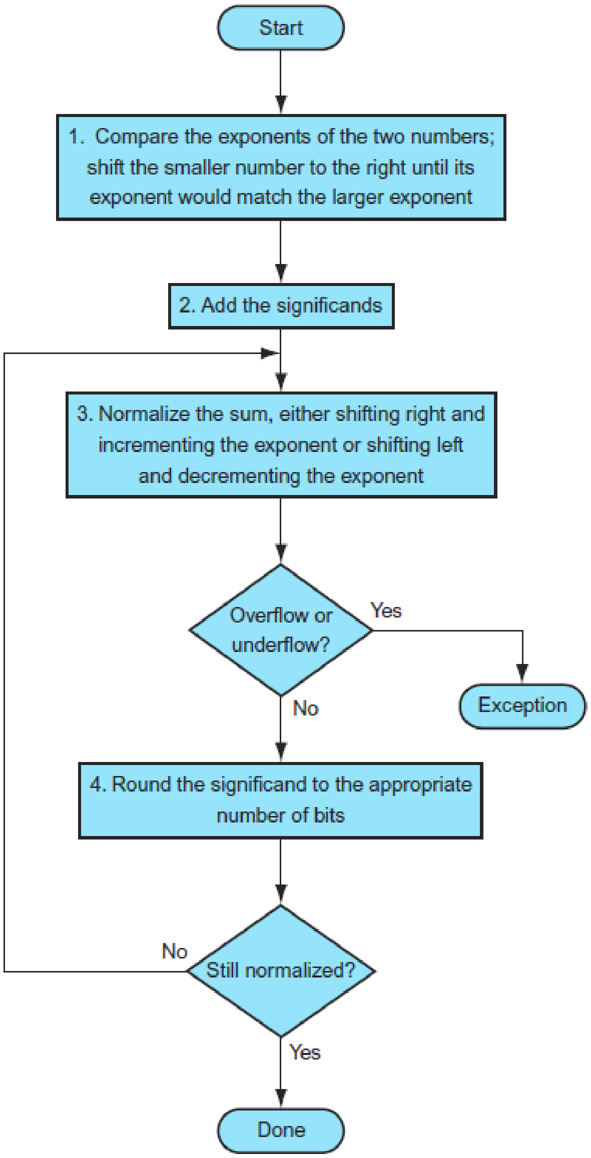
Student ID:

There are seven questions in the Exam.

Q1. What decimal number is represented by the following IEEE754 single precision floating point number? (10 pts)

1	1	0	0	0	0	0	1	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Q2. The following flow chart shows the binary floating-point addition algorithm for adding two floating point numbers in MIPS architecture. (10 pts)


Start
1. Compare the exponents of the two numbers; shift the smaller number to the right until its exponent would match the larger exponent
2. Add the significands
3. Normalize the sum, either shifting right and incrementing the exponent or shifting left and decrementing the exponent
Overflow or underflow?
Yes
Exception
No
4. Round the significand to the appropriate number of bits
Still normalized?
No
Yes
Done

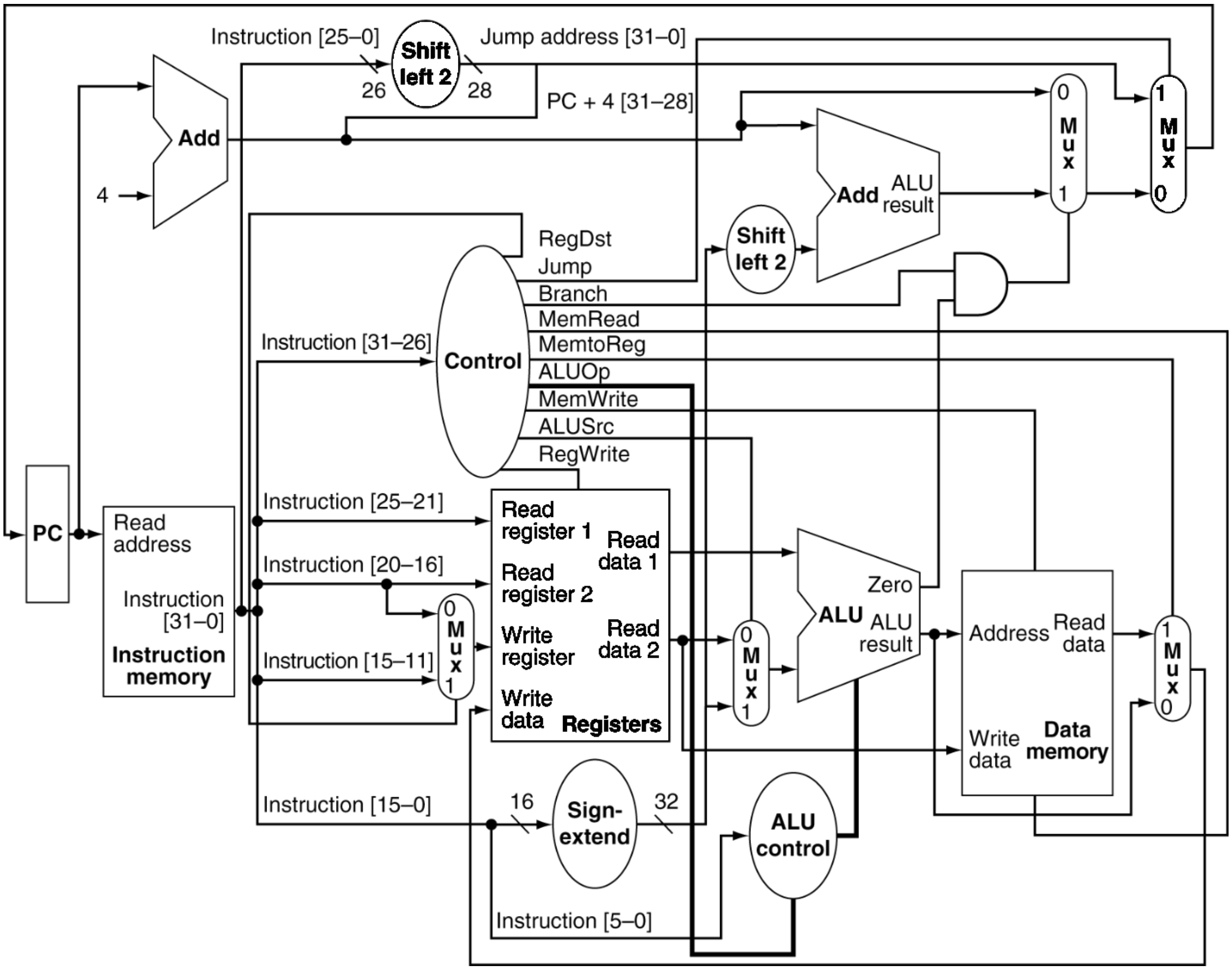
Use the above algorithm to perform addition of two binary floating-point numbers, 1001.000 and 0.0101. Clearly show the results produced in every step of the operation

Q3. Consider the following load-type instruction, **lw $s0, 32($t2)**, and its corresponding machine code:

10001101010100000000000000100000

This machine code consists of Opcode, Source registers (rs), destination registers (rt), and 16 bits offset.

Now suppose the instruction is executed in the MIPS processor as shown in the following figure. Highlight the different control signals and multiplexer inputs that will be activated during the execution process of the above **lw** instruction. Also indicate the buses where different portions of the machine code and the content of **32($t2)** can be found. (10 pts)



Q4. Show what happens when the branch is taken in the following instruction sequence, assuming the pipeline is optimized for branches that are not taken and the branch execution is moved to the ID stage:

10 add $14, $4, $2

14 sub $10, $4, $6

18 beq $5, $2, 4

22 or $13, $2, $6

24 and $12, $2, $6

……..

38 lw $8, 32($7)

Use figure 3 (a & b) to show what happen when a branch is taken. Specifically, **indicate the input/output values of the PC, adders, pipeline register data and control signals.**

Explain how the **beq** instruction is tested in the ID stage and what is advantage of moving the branch execution in the ID section?

(20 pts)

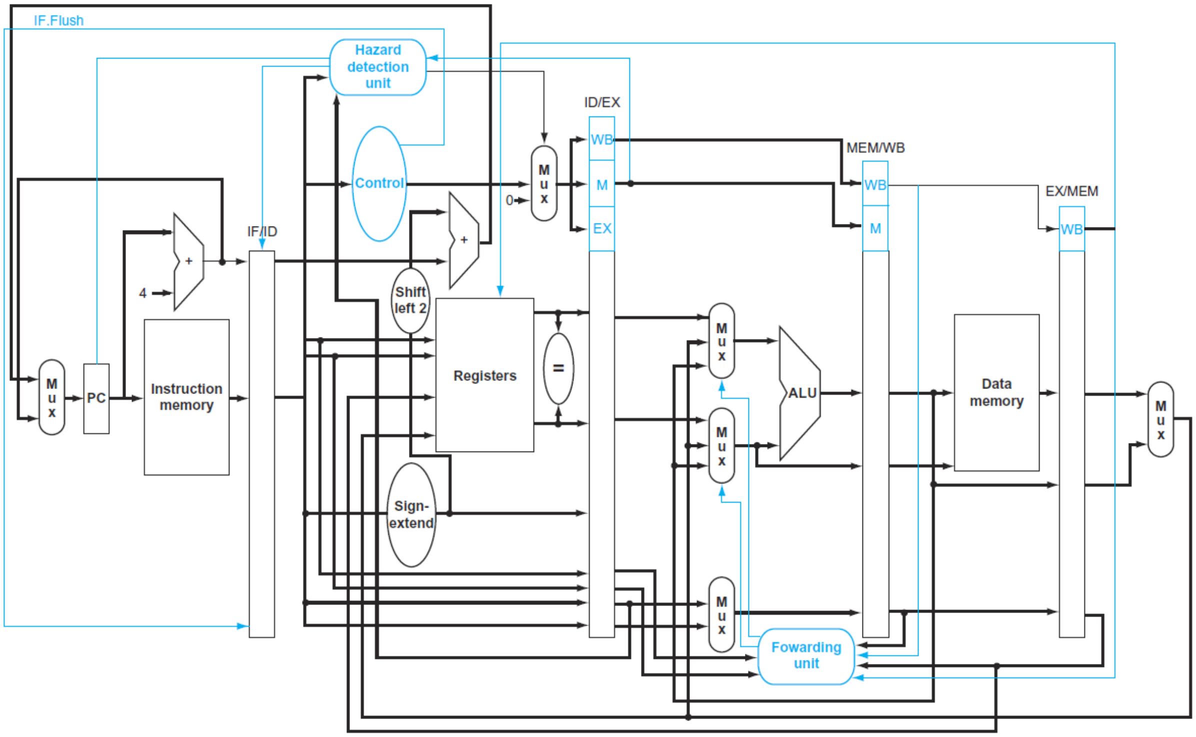


Figure 3(a)

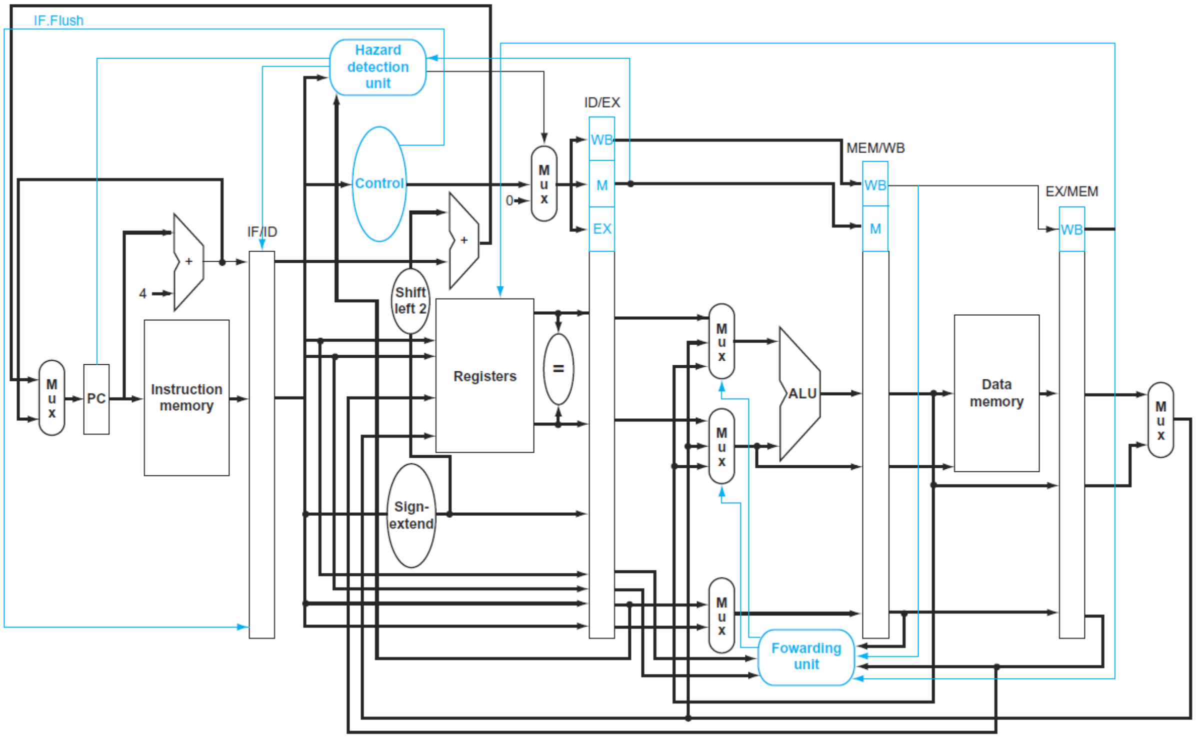

IF.Flush
Hazard detection unit
ID/EX
WB
M
EX
MEM/WB
WB
M
EX/MEM
WB
Control
M u x
0
IF/ID
+
4
+
Shift left 2
M u x
PC
Instruction memory
Registers
=
M u x
ALU
M u x
Data memory
M u x
Sign-extend
M u x
Fowarding unit

Figure 3(b)

Q5. Control is the most challenging aspect of processor design: it is both the hardest part to get right and the hardest part to make fast. One of the hardest part of control is implementing exceptions and interrupts- event other than branches and jumps that change the normal flow of instruction execution. A pipeline implementation treats exceptions as another form of control hazard. For example, suppose there is an arithmetic overflow in an **add** instruction. We must flush the instructions that follow the **add** instruction from the pipeline and began fetching instruction from new address.

Given the following sequence of instruction,

14_{hex} slt \$11, \$2, \$4

18_{hex} or \$12, \$2, \$5

$2C_{hex}$ and \$13, \$2, \$5

30_{hex} add \$7, \$3, \$7

34_{hex} lw \$16, 50(\$7)

38_{hex} slt \$15, \$6, \$7

$4C_{hex}$ sw \$1, 32(\$7)

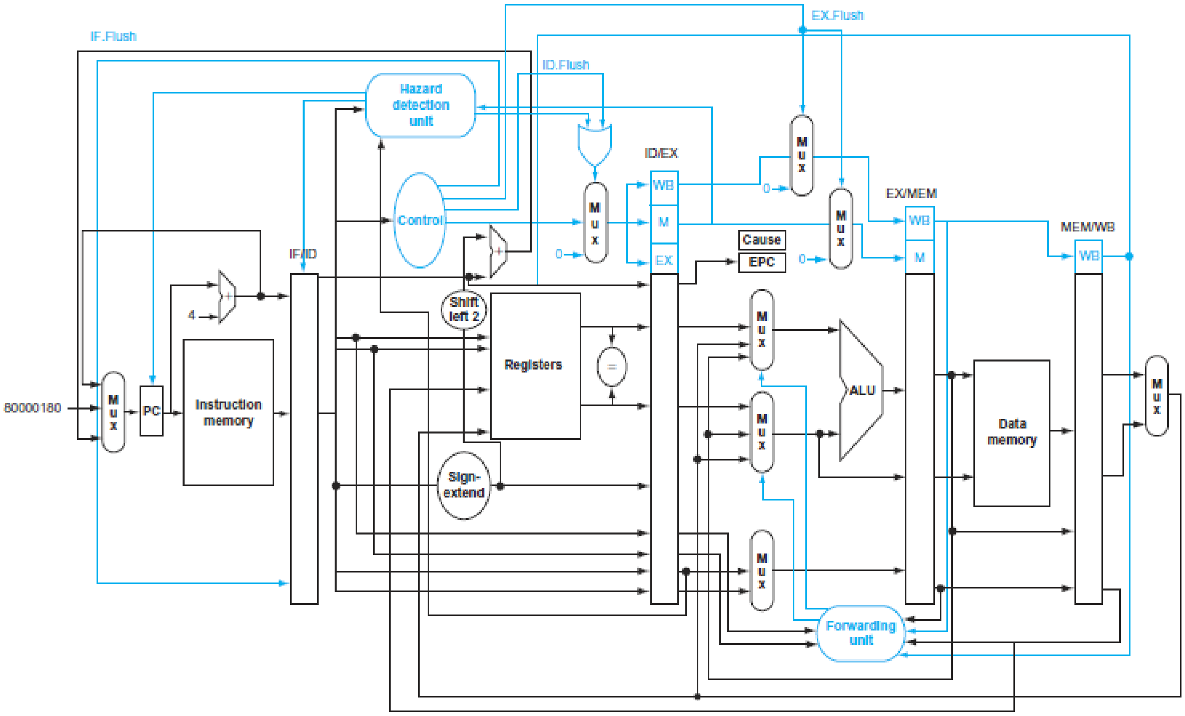
……………

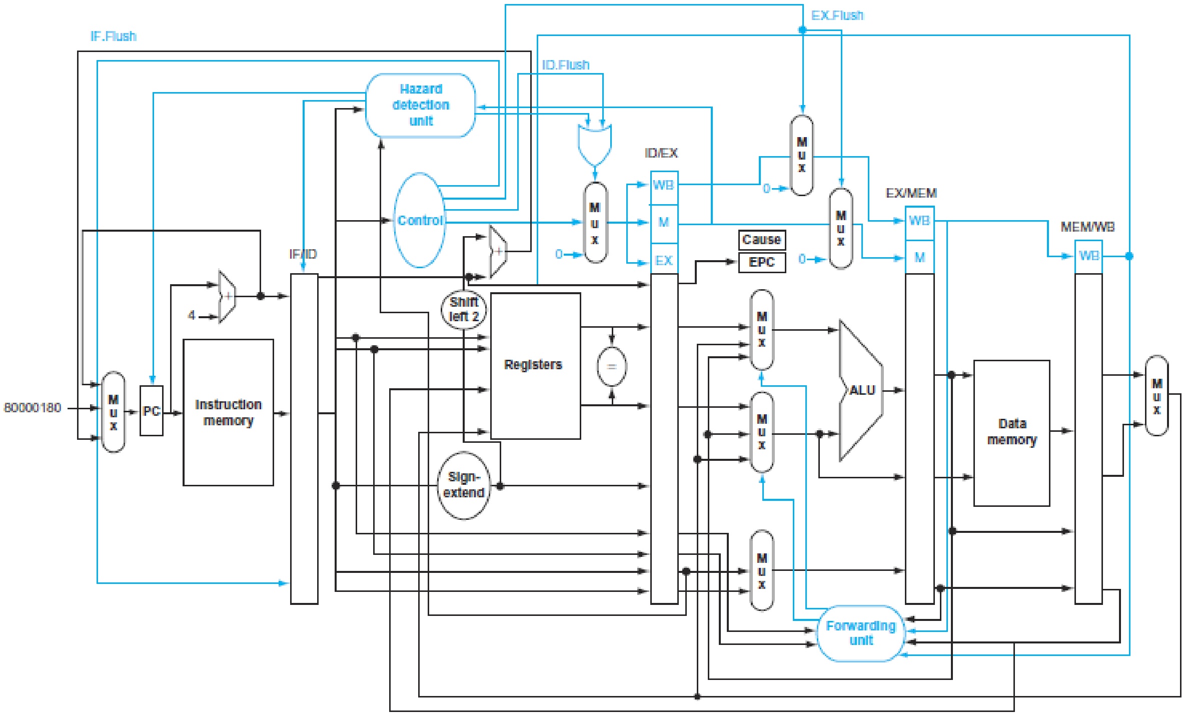
Assume the instruction to be invoked on an exception begin like this:

80000180_{hex} lw \$20, 1000(\$0)

80000184_{hex} sw \$21, 1004(\$0)

Show what happens in the pipeline if an overflow exception occurs in the **add** instruction. Specifically, **indicate the input/output values of the PC, adders, pipeline register data, bubble (no operation) and control signals.** On Figure 4 (a & b) clearly indicate your answer. (15 pts)


IF.Flush
EX.Flush
ID.Flush
Hazard detection unit
Control
M u x
ID/EX
EX/MEM
MEM/WB
WB
M
EX
Cause
EPC
0
IF/ID
Shift left 2
4
80000180
PC
Instruction memory
Registers
=
Sign-extend
ALU
Data memory
Forwarding unit


IF.Flush
EX.Flush
ID.Flush
Hazard detection unit
Control
M u x
ID/EX
EX/MEM
MEM/WB
WB
M
EX
Cause
EPC
0
IF/ID
Shift left 2
4
80000180
PC
Instruction memory
Registers
=
Sign-extend
ALU
Data memory
Forwarding unit

Q6. Given, a magnetic hard disk has 512B sector, 75,00rpm, 2ms average seek time, 200MB/s transfer rate, 0.2ms controller overhead, idle disk. Calculate the average data read time of that disk drive. (7 pts)

Q7. Consider a direct-mapped cache with 16 KiB of data and a block size of 16 bytes. Assume the 32-bit physical address is used to map the cache memory. Identify the memory bits that are used as index bits, tag bits and the offset bits of this direct-mapped cache memory. Also, draw a block diagram of the above cache memory clearly showing the data field, tag field, valid bit and how the hit/miss is detected using the valid bit and the tag bits. How many total bits are required for this direct-mapped cache memory? (18 pts)

Now, consider a memory address of 20500_{dec}. To what block number does this byte address (20500_{dec}) map? (10 pts)