CS232 Exam 2
March 14, 2008

Name: ______________________________

Section: 10am noon 2pm 3pm 4pm (circle one)

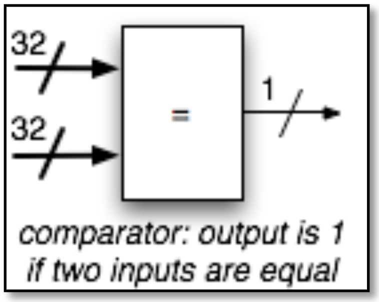
- This exam has 5 pages, including this cover. Page 5 is a reference on pipelining, performance and IEEE 754 single-precision floating point.
- There are three questions, worth a total of 100 points.
- No written references or calculators are allowed.
- Write clearly and show your work. State any assumptions you make.
- You have 50 minutes. Budget your time, and good luck!

Question	Maximum	Your Score
1	25	
2	50	
3	25	
Total	100	

Question 1: Single-cycle CPU implementation (25 points)

Part (a) Implement the "branch if memory and register are equal" (**bmre**) instruction, which uses the i-type format:

Field	op = 0	rs	rt	offset
Bits	31-26	25-21	20-16	15-0



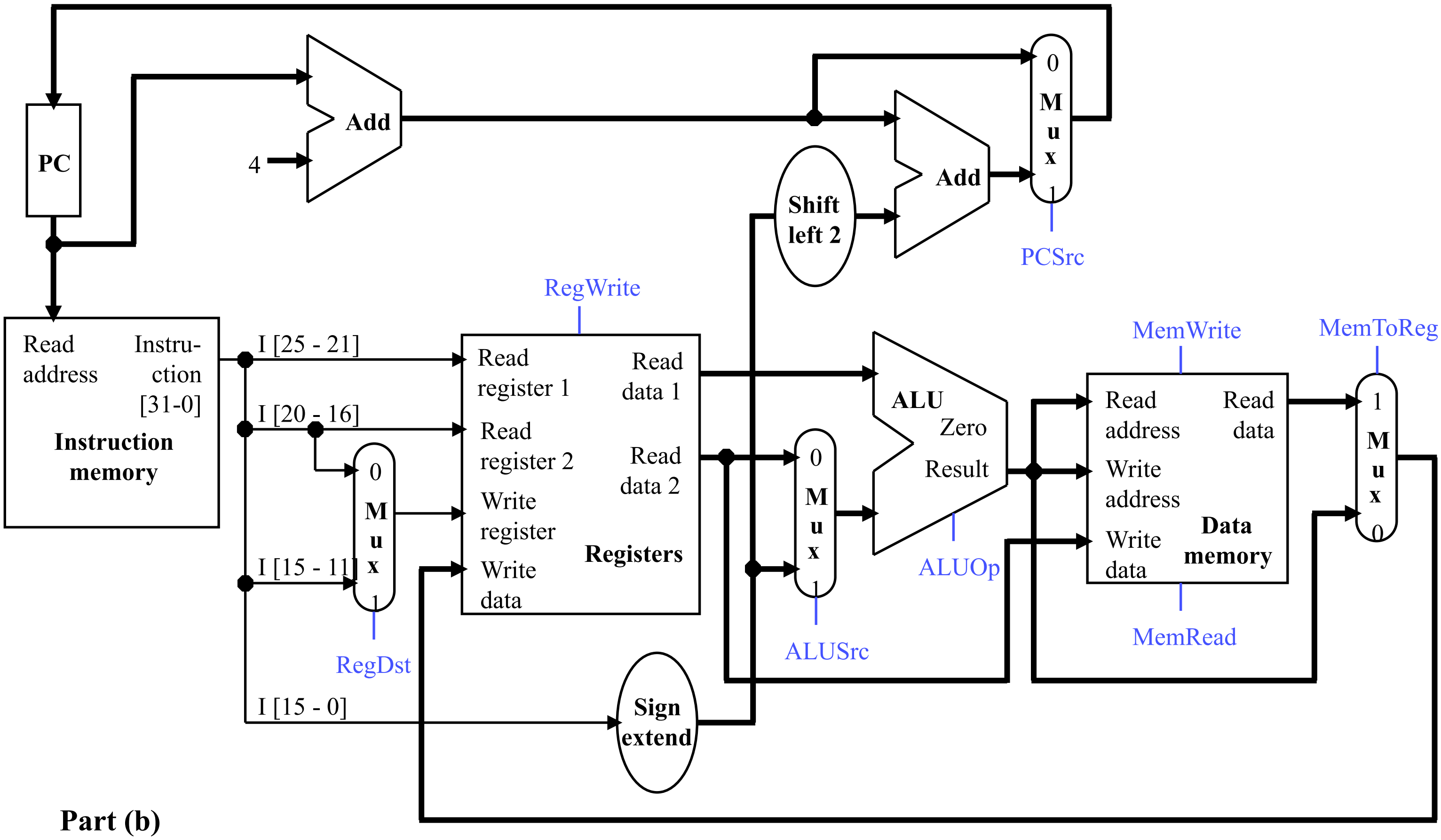
comparator: output is 1 if two inputs are equal

The **bmre** instruction has the following semantics:

```
if (R[rt] == M[R[rs]]) {
        PC = PC + 4 + 4*offset;
} else {
        PC = PC + 4
}
```

Show what changes are needed to support **bmre** instruction and write (next to the signal's name) values of **all** control signals for this instruction. You should only add wires, muxes, and a comparator (as shown above) to the datapath; do not modify the main functional units themselves (the memory, register file and ALU). Try to keep your diagram neat! (20 points)

Note: While we're primarily concerned about correctness, full points will only be rewarded to solutions that do not lengthen the clock cycle. Assume that the ALU, Memory and Register file all take 2ns, and everything else is instantaneous.



Part (b)

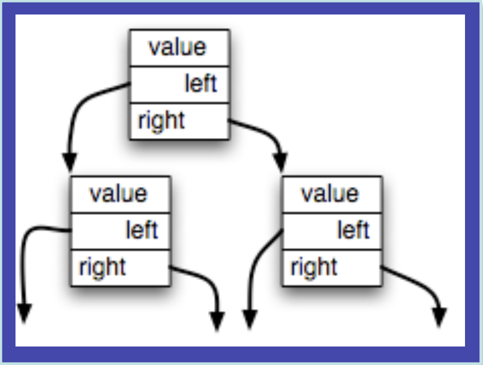
Discuss what impact supporting this instruction would have on a pipelined version of this processor. (5 points)

Question 2: Dependences and Pipelining (50 points)

A loop for searching a tree for a given value can be written as:

```
loop:  lw     $t0,   0($a0)
       beq    $a1,   $t0,   break
       bge    $a1,   $t0,   gt
       lw     $a0,   4($a0)
       j      end
gt:    lw     $a0,   8($a0)
end:   bne    $a0,   $0,    loop
break:
```

```
tree_node *x = tree_root;
while (x != NULL) {
  if (v == x.val)   break;
  if (v < x.val) {
    x = x.left;
  } else {
    x = x.right;
  }
}
```



Part (a)

Label all dependences within one iteration of this code (not just the ones that will require forwarding). *One iteration is defined as the instructions between the first* `lw` *and the* `bne`*, inclusive.* (10 points)

Part (b)

Assuming the 5-stage pipeline (shown on page 5) **with forwarding,** with branches predicted as **not taken** and mispredicted branches resolved in the EX stage, **complete the whole table below** (indicating which pipeline stage each instruction is in during each cycle). Assume the `bge` is taken. (stalls can be marked with an **S** or --) (30 points)

Inst	iter	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2
lw	1	F	D	E	M	W																	
beq	1		F																				
bge	1																						
	1																						

Part (c)

Compute the average CPI for each loop iteration assuming the `bge` is taken and this loop executes many iterations. (5 points)

Part (d)

How would the number of cycles per loop iteration change if the `bge` was not taken. (5 points)

Question 3: Concepts (25 points)

Write a short answer to the following questions. For full credit, answers should not be longer than **two sentences**.

Part a) Convert the decimal number below into an IEEE 754 single-precision floating point number. (10 points)

29.25

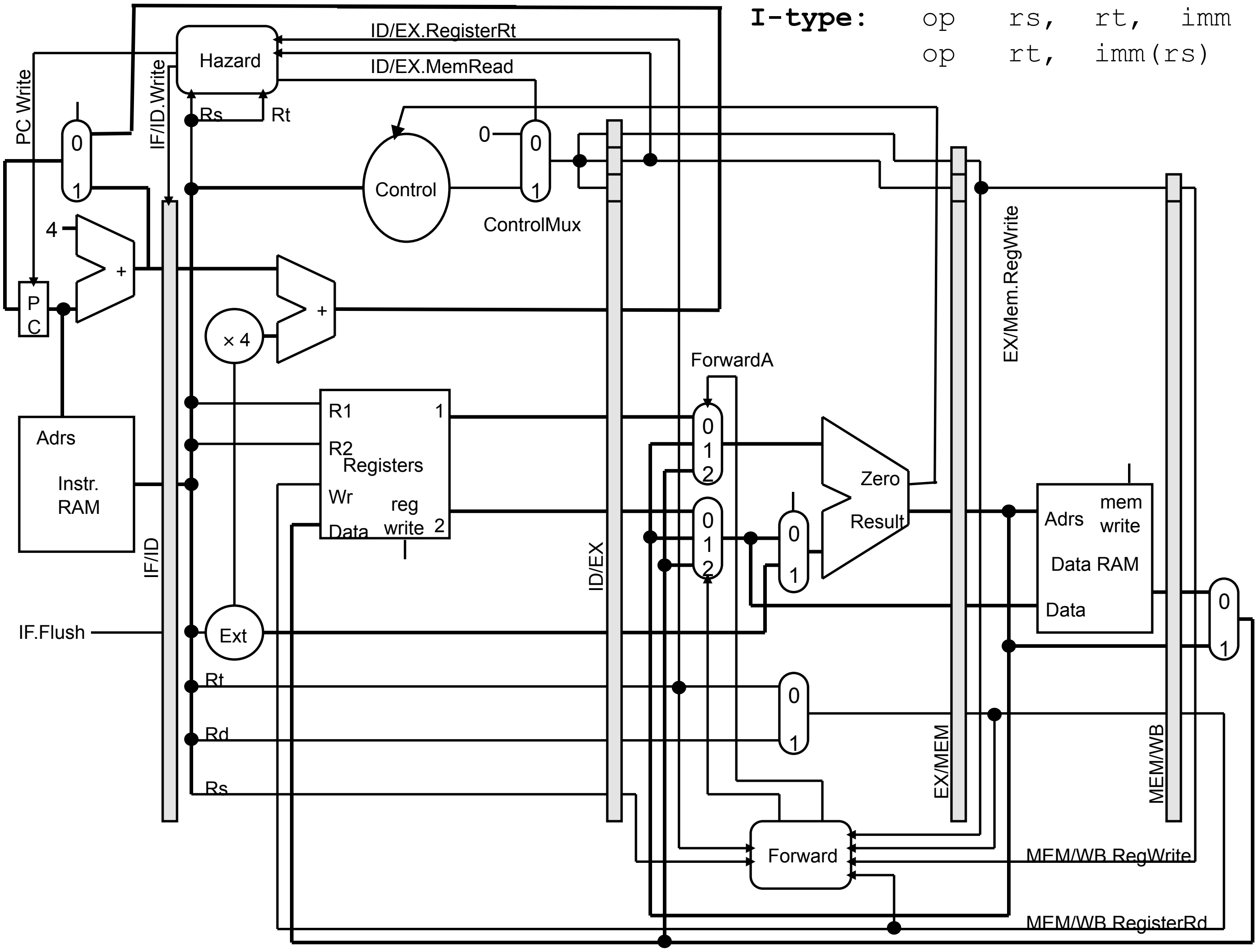
Part b) What is the maximum speedup achievable by pipelining a single cycle machine into 5 stages? Why is this ideal speedup generally not achieved (give at least 2 distinct reasons). (5 points)

Part c) If two processors have different instruction sets, how can you figure out which one is faster? (5 points)

Part d) Provide an expression of how to set the ForwardA control signal in the pipeline on the next page. *If you cannot provide an expression, provide a description for partial credit.* (5 points)

Reference: The 5-stage pipeline discussed in lecture.

R-type:	op	rd,	rs,	rt
I-type:	op	rs,	rt,	imm
	op	rt,	imm(rs)	



Performance:

1. Formula for computing the CPU time of a program P running on a machine X:

$$CPU\ time_{X,P} = Number\ of\ instructions\ executed_{P}\ x\ CPI_{X,P}\ x\ Clock\ cycle\ time_{X}$$

2. CPI is the average number of clock cycles per instruction:

$$CPI = Number\ of\ cycles\ needed\ /\ Number\ of\ instructions\ executed$$

3. Speedup is a metric for relative performance of 2 executions:

$$\begin{aligned} Speedup &= Performance\ after\ improvement\ /\ Performance\ before\ improvement \\ &= Execution\ time\ before\ improvement\ /\ Execution\ time\ after\ improvement \end{aligned}$$

IEEE 754: $(1 - 2s) * (1.f) * 2^{e-bias}$

s	e	f

Single precision numbers:

- include an 8-bit exponent field and a 23-bit fraction, for a total of 32 bits.
- bias = 127
- exponents 00000000 and 11111111 reserved for special numbers
  - +/- zero (f = 0, e = 0), +/- infinity (f = 0, e = 11111111), NaN (f = !0, e = 11111111)