Name: ___________________________

CS 281 — Computer Organization — Fall 2017

Final Exam

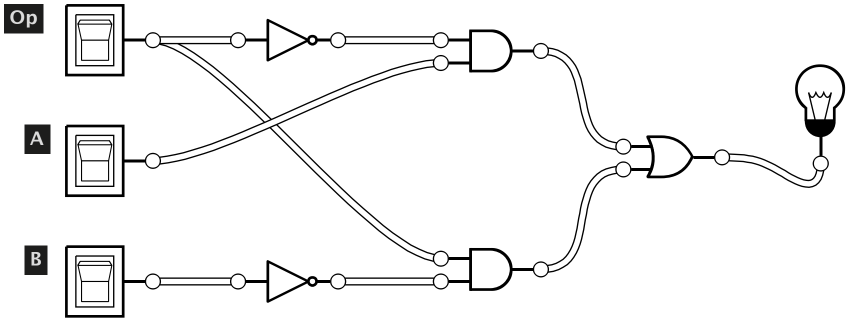
This exam should contain eight pages, including the MIPS instruction reference at the end of the exam. Closed book and notes. Calculators allowed.

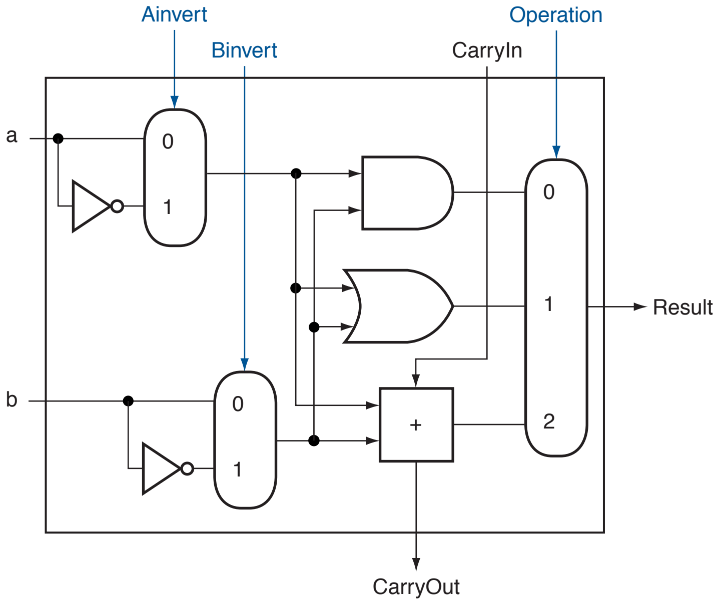
Problem 1: [20 points]

a) Construct a Boolean formula that's equivalent to the `Out` column in the truth table to the right.

Op	**A**	**B**	**Out**
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

b) Is the circuit below equivalent to the truth table above? Justify your answer.





Problem 2: [15 points]

a) What part of the processor is shown in the diagram above?

b) What are a and b on the left of the diagram? What do they represent? Where do those values come from?

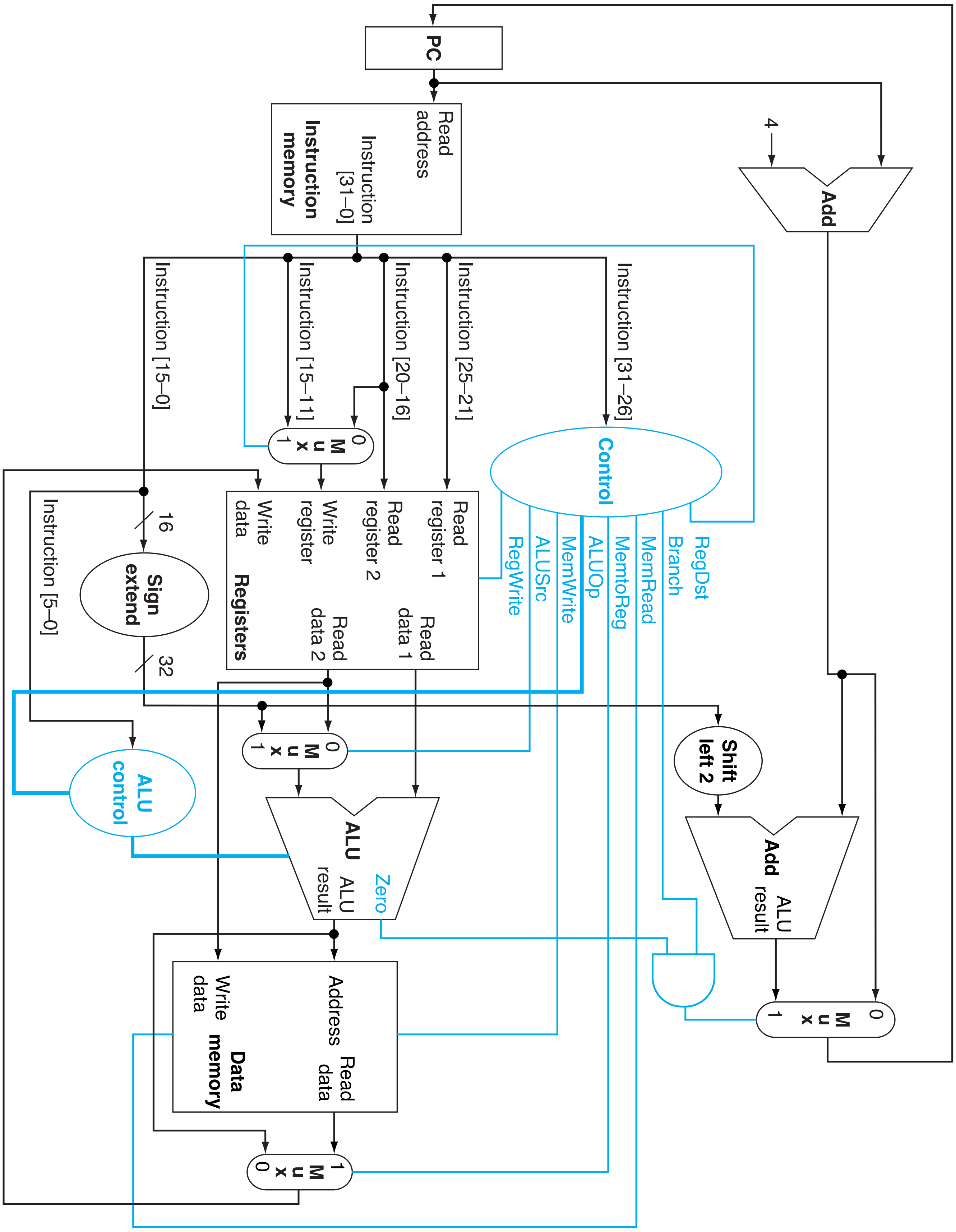
c) What is the role of the Ainvert input? What is it used for?

Problem 3: [20 points]

The diagram on the next page (Figure 4.17 from the book) shows the final development of the single-cycle CPU implementation with all of its control lines. In class, we determined the settings for each of the control lines for `lw`, `sw`, `beq`, and the R-Type instructions, and you did several more as a homework assignment.

a) Extend the hardware diagram as necessary such that it can execute the `bne` instruction as well (draw your modifications and/or extensions on the diagram). Note that `bne` has a different opcode than `beq`, but stores the source register numbers and signed offset in the same bit fields as `beq`.

b) Determine the control line settings required for `bne`. Write the settings next to each control line, using an `X` for "don't care" values. Fill in values for any additional control lines you added, if any, as well.



Problem 4: [20 points]

Below, define the MIPS procedure `print_string`. It should take two arguments, the address of a string to be printed and the number of times to print it, in that order, and print the string the specified number of times. For full credit, your procedure should manage the stack properly and use appropriate register conventions. (As a reminder, `syscall` #4 prints strings.)

```
print_string:
```

Problem 5: [25 points]

You've just been hired by Bradco, an up-and-coming processor manufacturer that needs some advice on how to get the best performance out of a new CPU. The current version of the processor has a clock frequency of 2.5 GHz, a 64 KByte L1 cache, no L2 cache and, at its current clock rate, a 200-cycle penalty for cache misses. A key benchmark contains 100 billion instructions, 35% of which are memory accesses. The average CPI is 5.0 for the benchmark, assuming all memory accesses hit in the L1 cache. There's a 3% miss rate on instructions, and a 6% miss rate on memory accesses.

a) [5 Points] How many seconds does the processor require to execute the benchmark program if we assume a perfect cache (all accesses hit in L1)?

b) [5 Points] How many seconds does the processor require to execute the benchmark if we assume the miss rates and penalties specified above?

c) [5 Points] One group of engineers argues that it would make sense to increase the L1 cache size such that the miss rates fall to 2% for instructions and 5% for data. Unfortunately, the clock speed would have to be slowed to 2.3 GHz to accommodate the larger L1 cache. How long would this processor take to execute the program? (Assume that the miss penalty is still 200 cycles, even though the clock is now a bit slower.)

d) [5 Points] Another group of engineers argues that the best approach is to reorganize the cache so that it uses longer block sizes. Based on simulations, it is projected that the reorganization will drop the instruction miss rate to 2% but leave the data miss rate unchanged. (Longer blocks means the cache holds fewer of them. The increased competition for the remaining slots balances out any benefits from longer block sizes for data accesses.) How long would it take to run the sample program using the new L1 cache design? (Base your work off of the data for the original processor, not the modified design from part c) above.)

MIPS Instruction Set Summary

Category	Instruction	Example
Arithmetic	add	add $1,$2,$3
	subtract	sub $1,$2,$3
	add immediate	addi $1,$2,100
	add unsigned	addu $1,$2,$3
	subtract unsigned	subu $1,$2,$3
	add immediate unsigned	addiu $1,$2,100
	multiply	mul $1,$2,$3
Logical	and	and $1,$2,$3
	or	or $1,$2,$3
	and immediate	andi $1,$2,100
	or immediate	ori $1,$2,100
	shift left logical	sll $1,$2,10
	shift right logical	srl $1,$2,10
Data transfer	load word	lw $1, 100($2)
	store word	sw $1, 100($2)
	move	move $1,$2
	load immediate	li $1,100
	load upper immediate (high 16 bits)	lui $1,100
Conditional branch	branch on equal	beq $1,$2,label
	branch on not equal	bne $1,$2,label
	set on less than	slt $1,$2,$3
	set less than immediate	slti $1,$2,100
	set less than unsigned	sltu $1,$2,$3
	set less than immediate unsigned	sltiu $1,$2,100
Unconditional jump	jump	j label
	jump register	jr $31
	jump and link	jal label