

1. (10) **True or False:**

- (1) It is possible to have a WAW hazard in a 5-stage MIPS pipeline.
- (1) Amdahl's law does not apply to parallel computers.
- (1) You can predict the cache performance of Program A by analyzing Program B.
- (1) The MIPS architecture uses condition codes.
- (1) It is not necessary to simulate very many instructions in order to get an accurate performance measure of the memory hierarchy.
- (1) A program's locality behavior is constant over the run of an entire program.
- (1) Communication is not a significant problem for parallel processor systems.
- (1) Throughput and response time are the same thing.
- (1) The design of an instruction set has little impact on how pipelinable it is.
- (1) Benchmark programs remain valid indefinitely.

2. (12) What do the following acronyms stand for?

SMP	DSM	COMA	CMP
SMT	TLB	UMA	MPP
MISD	MIMD	SIMD	SISD

For the next 5 problems, circle the correct answer.

3. (1) Which type of cache miss can be reduced by using longer lines?

Capacity Coherence Compulsory Conflict

4. (1) Which type of cache miss can be reduced by using shorter lines?

Capacity Coherence Compulsory Conflict

5. (1) Which type of operation is necessary in order to support synchronization?

Nuclear Atomic Radioactive

6. (1) Relaxing the requirement that Writes complete before Reads yields a model known as

Total Store Ordering Partial Store Ordering Weak Ordering

7. (1) To achieve a speedup of 90 using 100 processors, what percentage of the program must be parallelizable?

99.99% .9975% 2.5% .25% 5% .1%

8. (5) Circle the following techniques that would reduce the Miss Penalty.

Small and simple caches

Larger block size

Bigger caches

Way prediction

Trace caches

Higher associativity

Multilevel caches

Pipelined caches

Non-blocking caches

Multibanked caches

Compiler optimizations

Victim cache

Priority to Read Misses

Critical Word First/Early Restart

Merging Write Buffer

Avoiding Address Translation when Indexing Cache

Hardware Prefetching

Software prefetching

Very Short Answer:

9. (2) There are two types of performance - what are they?
10. (2) What concept lies at the heart of "RISC" processing?
11. (2) Why is there a "shift left 2" block feeding the PC incrementing adder?
12. (2) Where is a Dispatch ROM used?
13. (2) What is Amdahl's law (in words)?
14. (3) What are the 3 types of hazards?
15. (1) Which one can be eliminated by "throwing more money at the problem"?

Short Answers:

16. (3) What is the goal of the memory hierarchy? What two principles make it work?

17. (3) What is a victim cache, and how does it work?

18. (3) Branch predictors have trouble with a certain kind of control flow change that occurs very frequently in well-written, modular programs. What is the control flow change we are talking about, and what is one popular way to predict it?

19. (3) Why are there multiple dies per silicon wafer? Why not just fabricate one huge die per wafer?

20. (3) Write down the 3-term CPU performance equation developed in class.

21. (3) What is Strong Scaling?

22. (3) What is a "balanced" pipeline?

23. (4) What are the two main ways to insert a bubble/stall into a pipeline?

24. (2) What is the difference between static and dynamic scheduling?

25. (3 pts) What is the advantage of using a fixed-size instruction?

26. (4 pts) What are the 4 benchmark types we discussed in class?

Slightly Longer Answers:

27. (4) What is Cache Coherence, and why is it necessary?

28. (4) What pair of instructions are used to implement a lock in RISC systems (as described in the text)? Describe how this pair works together in order to accomplish the goal.

39. (8) Describe the difference between shared memory and message passing machines. Include the impact on design, cost, and programming model.

30. (5) What is a Victim Cache, and how does it work?

31. (5) Explain pipelining - why is it so popular? What are the advantages? What are the disadvantages?

32. (5) Supporting precise interrupts in machines that allow out of order completion is a challenge. Briefly explain why.

33. (5) Assume a relatively large fully associative write-back cache that contains no valid data. Given the following sequence of 5 memory operations (the address of the operation is in the square brackets):

WriteMem[100]

ReadMem[100]

WriteMem[100]

WriteMem[200]

WriteMem[100]

What are the number of hits and misses when using write allocate versus no-write allocate?

34. (4) Given 2 Megabytes of physical memory, a 22 bit Virtual address, and a page size of 4K bytes, write down the number of entries in the Page Table, and the width of each entry.

35. (4). What is one technique for reducing the amount of page table information that must be resident in memory at all times? (Drawing a picture is recommended.)

36. (8) In class, we talked about the cycle by cycle steps that occur on different interrupts. For example, here is what happens if there is an illegal operand interrupt generated by instruction i+1:

	1	2	3	4	5	6	7	8	9
i	IF	ID	EX	MEM	WB				
i+1		IF	ID	EX	MEM	WB	<- Interrupt detected		
i+2			IF	ID	EX	MEM	WB	<- Instruction Squashed	
i+3				IF	ID	EX	MEM	WB	<- Trap Handler fetched
i+4					IF	ID	EX	MEM	WB

Fill out the following table if instruction i experiences a fault in the Mem stage (page fault, for example):

	1	2	3	4	5	6	7	8	9	10
i	IF	ID	EX	MEM	WB					
i+1		IF	ID	EX	MEM	WB				
i+2			IF	ID	EX	MEM	WB			
i+3				IF	ID	EX	MEM	WB		
i+4					IF	ID	EX	MEM	WB	
i+5						IF	ID	EX	MEM	WB

What happens in this case?

	1	2	3	4	5	6	7	8	9	10
i	IF	ID	EX	MEM	WB	<- Data write causes Page Fault				
i+1		IF	ID	EX	MEM	WB	<- Inst Read causes Page Fault			
i+2			IF	ID	EX	MEM	WB			
i+3				IF	ID	EX	MEM	WB		
i+4					IF	ID	EX	MEM	WB	
i+5						IF	ID	EX	MEM	WB

37. (20 pts) In this question, we are going to wire up a 16-bit machine and write down the boolean equation for one of the signals.

The machine has 2 different instruction formats: R and I.

R-type:

Opcode	rs	rt	rd	funct
15-12	11-9	8-6	5-3	2-0

I-type:

Opcode	rs	rt	Immediate
15-12	11-9	8-6	5-0

The machine is byte-addressable.

Immediates are sign-extended

Branches are PC-relative.

The ALU can perform 4 functions:

	ALU1	ALU0
Add	0	0
Sub	0	1
And	1	0
Not	1	1

There are 16 instructions:

NOP	0000	Add	0010	Sub	0100	And	0110
Not	0001	Add Imm	0011	Sub Imm	0101	And Imm	0111
lw	1000	sw	1001	beqz	1010	j	1011

There are 9 control signals. Here are 4 of them:

ALU0

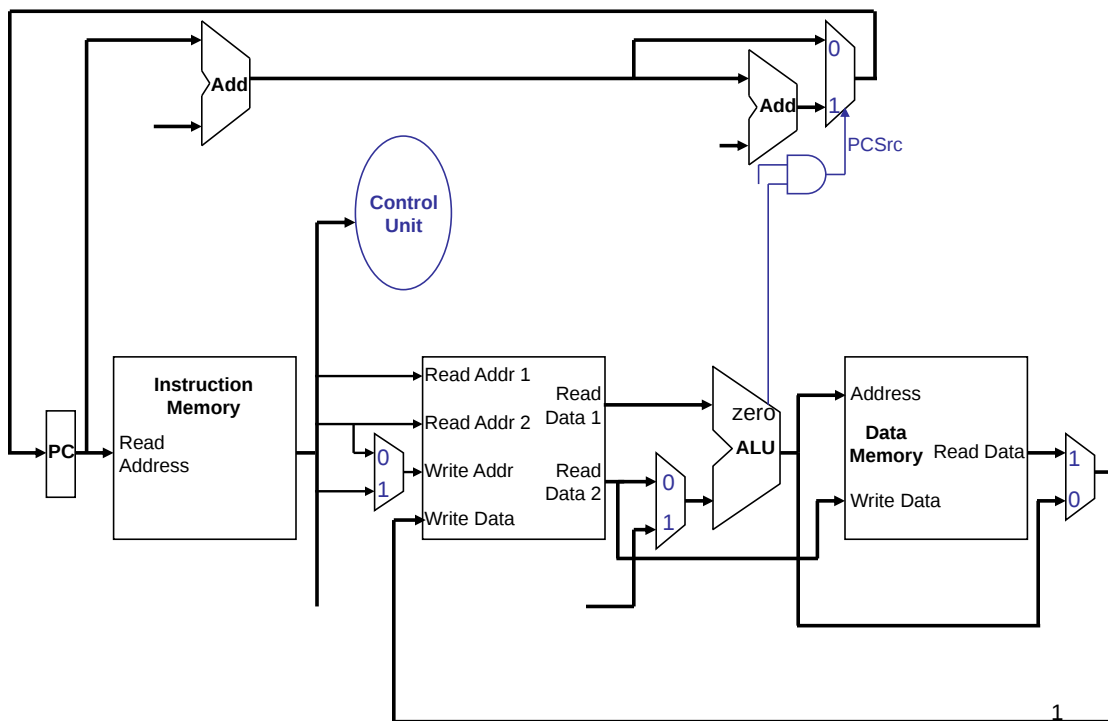
ALU1

MemRead

RegWrite

Your job: list the other 5 control signals, then wire up the diagram below.

Add all the parts and the various signal names so that someone like me could implement the circuit.



How many architecturally visible registers does this machine have?

Now, write down the exact boolean equation for the ALU1 signal.

38. (8 pts)

There are a number of changes that must be made to the previous design in order to make it a multicycle CPU. What are the 5 new registers that must be added, and where do they go? (Sketch them in on the diagram below). Also, there are several new control signals - list 3 of them, and explain what they do.

