

Very Short Answer:

- (1) (1) Peak performance does or does not track observed performance.
- (2) (1) Which is more effective, dynamic or static branch prediction?
- (3) (1) Do benchmarks remain valid indefinitely?
- (4) (2) Issuing multiple instructions per cycle puts tremendous pressure on what two parts of the machine?
- (5) (2) In class we mentioned VLIW and Superscalar as two ways to circumvent the Flynn Limit of 1. We also talked about two other approaches - what were they?
- (6) (2) Out of Order completion makes supporting what very difficult?
- (7) (2) Decoupled architectures split a program into two streams. What are they?
- (8) (2) Are wire delays or transistors more likely to be the most significant limit on clock frequency in the future? Why?
- (9) (2) What is Amdahl's law (in words)?
- (10) (2) What is the relationship between speculation and power consumption?

Short Answers:

- (10) (3) What is the primary difference between Scoreboarding and Tomasulo's algorithm? What hardware feature makes Tomasulo's work?
- (11) (3) Why are there multiple dies per silicon wafer? Why not just fabricate one huge die per wafer?
- (12) (3) The book lists several things that limit the amount of achievable ILP. List 3 of them.
- (13) (4) Understanding the hardware can influence how you write programs. Give at least 2 examples of how you might write software differently for a heavily pipelined machine versus a non-pipelined one.

- (14) (4) What is a predicated instruction? What are the advantages to using predicated instructions? When would you *not* want to use one?
- (15) (4) What is the definition of a basic block? Why is there a desire to create larger ones?
- (16) (3) There are at least two types of control flow changes that standard dynamic branch predictors have trouble with. There is a technique that works well for one of these types ... name the two types of branches, and the technique used to successfully deal with one of them.
- (17) (4) Supporting precise interrupts in machines that allow out of order completion is a challenge. Briefly explain why, and give three different techniques that can be used to provide precise interrupts.

- (18) (5) Why is branch prediction important? What performance enhancing techniques have made it so? List 3 examples of existing Branch Prediction strategies in order of (average) increasing effectiveness.
- (19) (5) What does SMT stand for? What is SMT trying to accomplish? What is the difference between Superscalar, coarse MT, fine MT, and SMT?
- (20) (6) Compare and contrast Superscalar and VLIW. Describe each, and list the advantages and disadvantages of each approach.

- (21) (10) Draw a basic high-level picture of what tomasulo's hardware looks like, when the ROB is included. (In other words, sketch out all the hardware involved, and how things are connected.) The emphasis is on conveying knowledge - do not worry about how pretty it is, but do make sure I can read it and understand what you have done.

(22) (10) You are given the following code sequence:

```
ADDF F1,F2,F3
SUBF F1,F4,F5
MULTF      F2,F6,F7
DIVF F1,F8,F9
```

Assume there are 8 logical and 16 physical registers. On the left below is the register mapping upon entering the code sequence. Your job is to fill in the mappings after the execution of the DIVF instruction, including what is on the free list. (Assume that during the execution of this code, no registers are released - in other words, the free list will be shorter at the end than at the beginning.)

BEFORE	
Logical	Physical
0	2
1	4
2	6
3	8
4	10
5	12
6	14
7	0

AFTER	
Logical	Physical
0	
1	
2	
3	
4	
5	
6	
7	

Free Pool: 0,2,4,9,10,13,14,15

Free Pool:

Now, rewrite the code sequence below using the actual physical register names instead of the logical ones.

```
ADDF P__,P__,P__
SUBF P__,P__,P__
MULTF P__,P__,P__
DIVF P__,P__,P__
```

(23) (15) Given the following loop:

```
LOOP: LoadF0,0($1)
      AddF4,F0,F2
      StoreF4,0(F1)
      SubR1,R1,#4
      BneR1,R2,Loop
```

There is a 1 cycle Load Delay Slot, a 1 cycle Branch Delay Slot, and a 2 cycle Add Delay Slot.
Your machine has 16 registers.

a) Calculate how many cycles this loop requires in order to execute 9 times.

b) Now unroll the loop 3 times, schedule the code, and calculate how many cycles your unrolled, scheduled loop requires to execute.

- (24) (4) 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+1 experiences a fault in the EX stage:

	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	<- Divide by Zero			
i+2			IF	ID	EX	MEM	WB	<- Illegal Opcode		
i+3				IF	ID	EX	MEM	WB		
i+4					IF	ID	EX	MEM	WB	
i+5						IF	ID	EX	MEM	WB