資訊工程學系 計算機結構 期中考(98 下)

姓名： 學號：

[Fundamentals of Computer Design] (25%)

1. (5%) What is an "Instruction Set Architecture"?

An interface between the software and hardware including

2. (5%) Please explain Moore's Law.

 # on transistors / cost-effective integrated circuit double every N months

3. (5%) Suppose that we want to enhance the processor used for Web service. The new processor is 10 times faster on computation in the Web service application than the original processor. Assume that the original processor is busy with computation 40% of the time and is waiting for I/O 60% of the time, what is the overall speedup gained by incorporating the enhancement?

$$\text{Speedup}_{\text{overall}} = \frac{1}{\left(1 - \text{Fraction}_{\text{enhanced}}\right) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$

$$= \frac{1}{\left(1 - 0.4\right) + \frac{0.4}{10}} = \frac{1}{0.64} = 1.56$$

4. (10%) Please explain Temporal Locality and Spatial Locality.

Temporal Locality (Locality in Time): If an item is referenced, it will tend to be referenced again soon (e.g., loops, reuse)

Spatial Locality (Locality in Space): If an item is referenced, items whose addresses are close by tend to be referenced soon (e.g., straight-line code, array access)

[Instruction Level Parallelism and Its Exploitation] (75%)

5. (10%) Please describe what data dependence exists between which two

instructions due to which registers or memory address.

L.D F0, 0(R1)
ADD.D F0, F4, F6
SUB.D F10, F0, F8
MUL.D F2, F8, F10
DIV.D F8, F2, F6
S.D F2, 0(R1)

true:

L.D and SUB.D F0
ADD.D and SUB.D F0
SUB.D and MUL.D F10
MUL.D and DIV.D F2
MUL.D and S.D F2

output:

L.D and ADD.D F0

anti:

MUL.D and DIV.D F8
SUB.D and DIV.D F8
L.D and S.D Mem[0+R1]

6. (5%) Please explain why Branch Target Buffers need to keep Branch target information.

Branch target calculation is costly and stalls the instruction fetch.

7. (5%) Please explain the difference between Basic Branch Prediction Buffer and Correlating Branch Predictors.

- **The previous schemes use only the recent behavior of a signal branch to predict the future behavior of the branch.**
- **Correlating Branch Predicator**
  - **Branch predictors that use the behavior of other branches to make a predication.**

8. (6%) An (m, n) predictor uses the behavior of the last m branches to chose from (2^m) n-bit predictors. The branch predictor is accessed using the low order p bits of the branch address and the m-bit global history. If we have m = 2, n = 2, p=5, then how many is the number of bits needed to implement an (2, 2) predictor?

$2^2 \times 2^5 \times 2$ = 256 bits

9. (5%) How can a processor with the Reorder Buffer maintain a precise exception model while dynamically execute code?

We could simply wait until it reached the head of the ROB and take the interrupt, flushing any other pending instructions from the ROB. Because instruction commit happens in order, this yields a precise exception.

10. (8%) How can the scheme of Tomasulo's approach can minimize RAW hazard and eliminate the stalls for WAW, and WAR hazards?

A. tracks when operands for instructions are available, to minimize RAW hazards (listen common data bus)
B. introduces register renaming to minimize WAW and WAR hazards with reservation station and register status

11. (5%) Please illustrate the performance bottleneck of Tomasulo Algorithm.
Performance limited by Common Data Bus

12. (6%) Please explain why an exception is **imprecise**.

If the processor state when an exception is raised does not look exactly as if the instructions were executed sequentially in strict program order.

i. The pipeline may have *already completed* instructions that are *later* in program order than the instruction causing the exception.
ii. The pipeline may have *not yet completed* some instructions that are *earlier* in program order than the instruction causing the exception.

13. (5%) What interval does the Reorder Buffer supply operands in the Speculative Tomasulo?

Interval between *completion of instruction execution* and *instruction commit*

14. (20%) Please write the contents of Instruction Status, Reservation Stations, and Register result status at clock cycle 10.

Instruction status:

Instruction		j	k	Issue	Exec Comp	Write Result
LD	F6	34+	R2	1	2	3
LD	F2	45+	R3	2	3	4
MULTD	F0	F2	F4	3	10	
SUBD	F8	F6	F2	4	6	7
DIVD	F10	F0	F6	5		
ADDD	F6	F8	F2	6	9	10

	Busy	Address
Load1	No	
Load2	No	
Load3	No	

Reservation Stations:

Time	Name	Busy	Op	S1 Vj	S2 Vk	RS Qj	RS Qk
	Add1	No					
	Add2	No					
	Add3	No					
0	Mult1	Yes	MULTD	R(F2)	R(F4)		
	Mult2	Yes	DIVD		R(F6)	Mult1	

Register result status:

Clock		F0	F2	F4	F6	F8	F10	F12	...	F3
10	*FU*	Mult1					Mult2			

Note：FP multiply (6 EX cycles)、FP add (2 EX cycles)、and FP divide (12 EX cycles)、load (1 EX cycles)

Note：FP multiply (10 EX cycles)、FP add (2 EX cycles)、and FP divide (40 EX cycles)、load (2 EX cycles)