國立交通大學試題紙

九十九學年度第一次
博士班資格考

科目：計算機結構(A)　　日期： 100 年 1 月 26 日　第 1 頁 共 2 頁

請 "✓" 明　✓不可看書　可看書

* 請將答案依題號順序寫入答案卷

答題時字跡需工整，否則不予計分。Write your answers legibly; otherwise you will get zero score.

1. **(10%)** Describe the seven dimensions of an ISA (instruction set architecture) and give two examples for each dimension.

2. **(8%)** For a 5-GHz processor, the instruction counts and the clock cycle count per instruction for different instruction types of a program and the processor, respectively, are given below:

Instruction type	Instruction count	Clock cycle count
Integer ALU ops	4,000,000	1
Floating-point ops	2,500,000	5
Loads & Stores	1,500,000	3
Branches	2,000,000	2

   (a) (4%) Determine the effective CPI and execution time for this program.
   (b) (4%) How many percentage of the execution time is used for executing floating-point operations in this program? What is the overall speedup if the time for floating-point operations is reduced by 40%?

3. **(6%)** For the following three major methods for evaluating branch conditions, describe how the condition is tested and the advantage and disadvantage of each method: ***condition code***, ***condition register***, and ***compare and branch***.

4. **(8%)** Give the equation to calculate the value of the CPI (clocks per instruction) for a pipelined processor:

   Pipeline CPI = Ideal pipeline CPI + Structural stalls + Data hazard stalls + Control stalls

   For each of the following techniques, which ones of the components of the CPI equation does the technique affect? If a technique affects data hazard stalls, please specify the dependence types (*true*, *anti*, and/or *output* dependences) of the hazards.

   i. Forwarding and bypassing
   ii. Delayed branches
   iii. Dynamic scheduling with renaming
   iv. Branch prediction
   v. Issuing multiple instructions per cycle
   vi. Hardware speculation
   vii. Loop unrolling
   viii. Basic compiler pipeline scheduling

5. **(12%)** Consider the execution of the following code on a single-issue processor.

```
1    SUB.D    F1, F6, F4
2    MULT.D   F8, F1, F2
3    MULT.D   F2, F10, F1
4    ADD.D    F4, F3, F5
5    DIV.D    F8, F2, F5
```

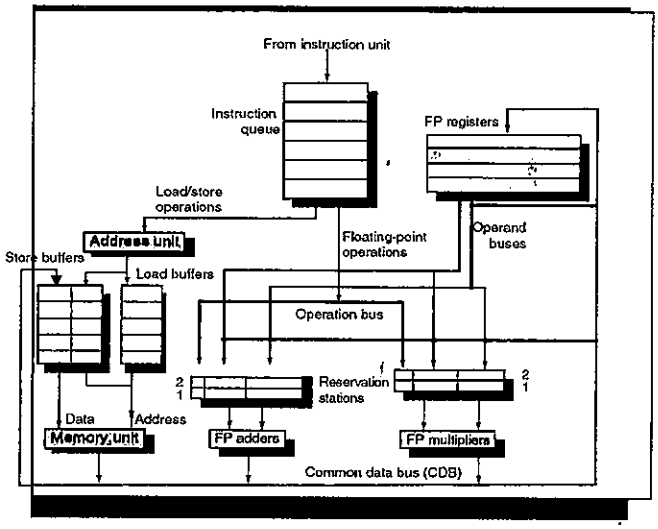
(a) (4%) List the dependencies in the code by specifying the instruction pair and the register that causes a dependence as "(I_i-I_j, F_k)"; for example, (I_1-I_2, F0).

i. For true dependences　ii. For anti dependences　iii. For output dependences

(b) (8%) Use the single-issue Tomasulo MIPS pipeline shown in the following figure. There are two FP adders and two FP multipliers in the processor, and two reservation stations for both FP adders and multipliers. Only one CDB (common data bus) exists in the processor. Assume the following EX (execution) cycle latencies for floating-point functional units:

- Add: 2 cycles
- Subtract: 2 cycles
- Multiply: 3 cycles
- Divide: 4 cycles

Show the time of issue, execution, and writing result of each operation in the pipeline. Express your answer as the following table.



No.	Instruction	Issues at clock number	Execute (starts at clock number)	Execute (Finishes at clock number)	Write CDB at clock number
1	SUB.D F1, F6, F4	**1**	**2**	**3**	**4**
2	MULT.D F8, F1, F2				
3	MULT.D F2, F10, F1				
4	ADD.D F4, F3, F5				
5	DIV.D F8, F2, F5				

6. **(6%)** Compare the following two approaches in use for multiple-issue processors in the aspects of issue structure, hazard detection, scheduling, and distinguishing characteristic of each approach: ***dynamic superscalar*** and ***VLIW***.

4. (5 %) In a shared memory multiprocessor system, two processes have the following code segments:

```
P1:
    A = 1;
    if (A>B) kill P2;
P2:
    B = 1;
    If (B>A) kill P1;
```

Before these code segments are entered, both A and B are 0. Occasionally, both processes are killed Explain how this might happen? You may assume that each processor in this system has a local cache and that a cache coherence protocol has been implemented.

5. (10%) A four-processor shared-memory system implements the snooping protocol for cache coherence. For the following sequence of memory reference, show the state (Exclusive, Share, or Invalid) of the line containing the variable *a* and *b* in each processor's cache after each reference is resolved. Assume *a* and *b* in same cache block. All processors start out with the line containing *a* and *b* invalid in their cache.

Operation	Processor 0	Processor 1	Processor 2	Processor 3
P0 reads a				
P1 writes b =30				
P2 reads a				
P3 writes b =10				
P0 reads a				

6. (10%) Suppose you have a 32-bit processor, with a virtual-memory page-size of 16K. The data cache is 32K in size with 32-byte cache blocks. Finally, your TLB has 4 entries. Assume that you wish to do TLB lookups in parallel with cache lookups.

Draw a block diagram of the data cache and TLB organization, showing a virtual address as input and both a physical address and data as output. Include cache hit and TLB hit output signals. Include as much information about the internals of the TLB and cache organization as possible. Include, among other things, all of the comparators in the system and any muxes as well. You can indicate RAM as with a simple block, but make sure to label address widths and data widths. Make sure to use abstraction in your diagram so that we can understand it. Label the function of various blocks and the width of any buses.

國立交通大學試題紙

九十九學年度第一次
博士班資格考

科目：計算機結構(B)　　日期： 100 年 1 月 26 日　第 1 頁 共 2 頁

請"✓"明　✓不可看書　可看書

＊ 請將答案依題號順序寫入答案卷

答題時字跡需工整，否則不予計分。Write your answers legibly; otherwise you will get zero score.

1. (10%) Branch prediction is an important performance optimization for highly-pipelined and superscalar processors. There are many ways to predict branches, each with its own set of pros and cons.
   (a) Branch prediction schemes are usually classified as either static, dynamic, or a combination of both. Define what is meant by static and dynamic in this context. Discuss the advantages and disadvantages of static vs. dynamic schemes.
   (b) Describe an implementation of the classic two-bit dynamic branch prediction algorithm.
   (c)Two-level predictors use additional information about program behavior to improve branch prediction accuracy. Discuss the kinds of information these predictors use and how they use it.

2. (10 %) The von Neumann execution model dictates sequential semantics: namely that all instructions must appear to execute one at a time and in program order. Control dependences (e.g., branches) are the first challenge. To achieve instruction level parallelism, most high-performance processors predict branches and speculatively execute based on that prediction. But memory dependences (e.g., loads and stores to the same address) also present challenges to out of order instruction execution. All speculative processors must respect memory dependences to ensure sequential semantics. More aggressive processors use additional prediction and speculation mechanisms to expose more parallelism.
   (a) Give a pseudo-assembly-code example that illustrates a case where memory dependences might limit ILP (in the absence of additional prediction and speculation).
   (b) Discuss hardware mechanisms that suffice to detect and enforce memory dependence order in out-of-order processors
   (c) Discuss how prediction and speculation of memory dependences can increase instruction level parallelism.

3. (5 %) Consider a virtual memory system.
   (a) What are the advantages and disadvantages of using a larger page size?
   (b) If you wrote a new operating system, how would you do page replacement and why?