國立交通大學試題紙

九十七學年度第二次
博士班資格考

科目：計算機結構(A)　　日期： 98 年 7 月 22 日　第 1 頁 共 3 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. (5%) Prior to the early 1980s, machines were built with more and more complex instruction set. Why has there been a move to RISC machines away from complex instruction machines?

2. (10 %)
   a) (7 %) What is a multiple issue processor? Define the two types of multiple issue processors, and give two advantages of each.
   b) (3 %) What is dynamic and static scheduling? Give an example that shows when dynamic scheduling is useful.

3. (5 %) Consider the following assembly language code:

```
I0: ADD R4 ← R1 + R0;
I1: SUB R9 ← R3 - R4;
I2: ADD R4 ← R5 + R6;
I3: LDW R2 ← MEM[R3 + 100];
I4: LDW R2 ← MEM[R2 + 0];
I5: STW MEM[R4 + 100] ← R2;
I6: AND R2 ← R2 & R1;
I7: BEQ R9 — R1, Target;
I8: AND R9 ← R9 & R1;
```

   Consider a pipeline with forwarding, hazard detection, and 1 delay slot for branches. The pipeline is the typical 5-stage IF, ID, EX, MEM, WB MIPS design. For the above code, complete the pipeline diagram (instructions on the left, cycles on top) for the code. Insert the characters IF, ID, EX, MEM, WB for each instruction in the boxes. Assume that there are two levels of bypassing, that the second half of the decode stage performs a read of source registers, and that the first half of the write-back stage writes to the register file.
   Label all data stalls (Draw an X in the box). Label all data forwards that the forwarding unit detects (arrow between the stages handing off the data and the stages receiving the data). What is the final execution time of the code?

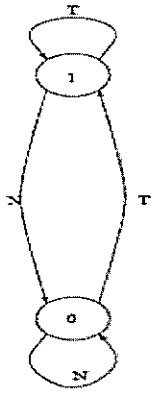
4. (5 %) Structural, data and control hazards typically require a processor pipeline to stall. Listed below are a series of optimization techniques implemented in a compiler or a processor pipeline designed to reduce or eliminate stalls due to these hazards. For each of the following optimization techniques, state which pipeline hazards it addresses and how it addresses it. Some optimization techniques may address more than one hazard, so be sure to include explanations for all addressed
   (a) Branch Prediction
   (b) Instruction Scheduling
   (c) Branch Delay slots
   (d) Increasing number of functional units (ALUs, adders etc)

5. (12 %) This problem tests on your knowledge of branch prediction.

The figures below illustrate two possible predictors.

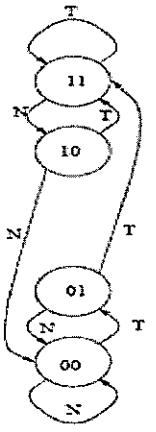
- **Last taken** predicts taken when 1
- **Automata A3** predicts taken when 11 and 10

Fill out the tables below in your answer sheet for each branch predictor. The execution pattern for the branch is NTNNTTTN.



LAST-TAKEN

Execution Time	Branch Execution	State Before	Prediction	Correct or Incorrect	State After
0	N	0			
1	T				
2	N				
3	N				
4	T				
5	T				
6	T				
7	N				



AUTOMATON-A3

Execution Time	Branch Execution	State Before	Prediction	Correct or Incorrect	State After
0	N	01			
1	T				
2	N				
3	N				
4	T				
5	T				
6	T				
7	N				

6. (13 %) For the following problem, assume an in-order DLX-style pipelined architecture that has functional units that take the following number of execution cycles:

   1. Floating-point multiplier: **5 cycles**
   2. Floating-point adder: **2 cycles**
   3. Integer operations: **1 cycle**

   Assume as well that there is **one branch delay slot**, that there is no delay between integer operations and dependent branch instructions, and that the load-use latency is **2 cycles** (i.e. **2 cycles** needed to use the data after load a data). Assume that all functional units are fully pipelined and bypassed. The following code computes a dot product. Assume that r1 and r2 contain addresses of arrays of floating-point numbers, and r3 contains the length of the arrays (in elements). Assume that r4 is initialized to zero. Then, the dot product can be computed as follows:

```
loop:  ld     f5, 0(r1)     ; load element from first array
       ld     f6, 0(r2)     ; load element from second array
       multd  f7, f5, f6    ; multiply elements
       addd   f4, f4, f7    ; add elements to accumulator in f4
       addi   r1, r1, #8    ; increment pointers
       addi   r2, r2, #8
       subi   r3, r3, #1    ; decrement element count
       bnez   r3, loop      ; continue until all elements done
        nop                 ; (branch delay slot)
```

   (a) (2 %) How many cycles does this loop take per iteration? Indicate stalls in the above code by labeling each of them with a number of cycles of stall.
   (b) (3 %) Reschedule this code to run with as few cycles per iteration as possible. Do not unroll it. How many cycles do you get per iteration of the loop now?
   (c) (4 %) Unroll the loop once and schedule it to run with as few cycles as possible per iteration of the original loop. How many cycles do you get per iteration now?
   (d) (4 %) If you were to unroll the loop 10 times, how many cycles per iteration would this achieve? *(hint: you do not need to actually perform the unrolling to answer this question)*

國立交通大學試題紙

九十七學年度第二次
博士班資格考

科目：計算機結構(B)　　日期：98 年 7 月 22 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. (12%) In studying growth in processor performance in past 30 years,
   [I] During the period of 1978~1985,
   (a) What is the approximate rate of improvement per year?
   (b) What are the major driving force(s)?
   (c) Give at least one example for each driving force, with proper explanation.
   [II] For the period of 1985~2002, repeat questions (a) to (c) in (d) to (f). In your answers to (d) to (f), specifically indicate if the performance growth rate had increased or decreased compared with the 1978~1985 period, and what is/are the reasons.
   [III] For the period of 2002~2005, repeat questions (a) to (c) in (g) to (i). In addition, in your answers to (g) to (i), specifically indicate if the performance growth rate had increased or decreased compared with the 1985~2002 period, and what is/are the reasons causing this increase or decrease.

2. (9%) In examining trends in technologies, we have examined four closely related technologies: microprocessor, memory (DRAM), disk, and network, in terms of "relative bandwidth improvement vs. relative latency improvement."
   (a) (3%) Which of the two, bandwidth and latency, improves more, and roughly by how much compared with the other?
   (b) (6%) The class lecture notes have listed six reasons for these unequal improvements in bandwidth and latency. Give any three, out of the six, with proper explanations if necessary.

3. (6%) Two cooling fans are installed in a desktop computer, operated together, and only one fan is sufficient to cool the computer. What is the $MTTF_{two}$ of this redundant fan pair, given the $MTTF_{one}$ per fan, and
   (a) If you sit and wait until both fans fail? Or
   (b) If you fix the fan once it fails?
   (Make any necessary assumptions of yours in solving this problem; for example, assume $MTTR_{one}$.)

4. (8%) In MIMD architectures,
   (a) Clearly define multiprocessor cache coherence problem; and then use a simple but comprehend example to illustrate this problem.
   (b) Clearly define multiprocessor memory consistency problem; and then use a simple but comprehend example to illustrate this problem.

5. (7%) In implementing cache coherence protocol, when a local cache performs a write, what will it do to other caches or to the home directory? Assume the protocol you use (snoopy or directory), and state the procedure, each and every signal (those internal to cache itself, or those to be sent out for broadcast of to directory) used, clearly in your answer. Note that the written cache block may be in *invalid*, or *shared*, or *exclusive* state, and I suggest that you organize your answers accordingly. Note also that use of a state transition diagram may not be necessary here.

6. (4%) Calculate the page table size in bits for the following process: 32-bit byte-addressable virtual address space, 1 kB page size, and every page table entry has 10 bits for valid, replacement, protection, etc.

7. (4%) A computer system uses three caches: L1$, L2$, and L3$. Each of them has a $hitrate_i$ and $hittime_i$, $i=1\sim3$. Assume that the main memory never misses, and has an access time $hittime_m$. What is the computer's average memory access time?