國立中山大學資訊工程學系

100 學年度第 2 學期博士班資格考試　計算機結構

1. (1) (6%) Consider the execution of a program which results in the execution of 2 million instructions on a 400-MHz processor. The program consists of four major types of instructions. The instruction mix and the CPI for each instruction type are given below based on the result of a program trace experiment:

Instruction Type	CPI	Instruction Mix
Arithmetic and logic	1	60%
Load/store with cache hit	2	18%
Branch	4	12%
Memory reference with cache miss	8	10%

What are the average CPI and the corresponding MIPS rate when the program is executed on a uniprocessor with the above trace results?

(2) (6%) Suppose that a task makes extensive use of floating-point operations, with 40% of the time is consumed by floating-point operations. With a new hardware design, the floating-point module is speeded up by a factor of 8. What is the overall speedup?

2. Consider a system in which bus cycles takes 500 ns. Transfer of bus control in either direction, from processor to I/O device or vice versa, takes 250 ns. One of the I/O devices has a data transfer rate of 50 KB/s and employs DMA. Data are transferred one byte at a time.

(1) (8%) Suppose we employ DMA in a burst mode. That is, the DMA interface gains bus mastership prior to the start of a block transfer and maintains control of the bus until the whole block is transferred. For how long would the device tie up the bus when transferring a block of 128 bytes?

(2) (6%) Repeat the calculation for cycle-stealing mode.

3. Suppose the page table for the process currently executing on the processor looks like the following. All numbers are decimal, everything is numbered starting from zero, and all addresses are memory byte addresses. The page size is 1024 bytes.

Virtual page number	Valid bit	Reference bit	Modify bit	Page frame number
0	1	1	0	4
1	1	1	1	7
2	0	0	0	–
3	1	0	0	2
4	0	0	0	–
5	1	0	1	0

(1) (6%) Describe exactly how, in general, a virtual address generated by the CPU is translated into a physical main memory address.

(2) (9%) What physical address, if any, would each of the following virtual addresses correspond to? (Do not try to handle any page faults, if any.)　(i) 1052　(ii) 2221　(iii) 5499

4. Show how the following floating-point calculations are performed (where significands are truncated to 4 decimal digits). Show the results in normalized form.

(1) (4%) $3.344 \times 10^{1} + 8.877 \times 10^{-2}$

(2) (4%) $8.844 \times 10^{-3} - 2.233 \times 10^{-1}$

(3) (4%) $(3.344 \times 10^{2}) \times (1.234 \times 10^{0})$

5. Figure 1 shows an example of a superscalar processor organization. The processor can issue two instructions per cycle if there is no resource conflict and no data dependence problem. There are essentially two pipelines, with four processing stages (fetch, decode, execute, and store). Each pipeline has its own fetch decode and store unit. Four functional units (multiplier, adder, logic unit, and load unit) are available for use in the execute stage and are shared by the two pipelines on a dynamic basis. The two store units can be dynamically used by the two pipelines, depending on availability at a particular cycle. There is a lookahead window with its own fetch and decoding logic. This window is used for instruction lookahead for out-of-order instruction issue. Consider the following program to be executed on this processor:

```
I1:  Load R1, A        /R1 ← Memory (A)/
I2:  Add R2, R1        /R2 ← (R2) + R(1)/
I3:  Add R3, R4        /R3 ← (R3) + R(4)/
I4:  Mul R4, R5        /R4 ← (R4) * R(5)/
I5:  Comp R6           /R6 ← (R6)/
I6:  Mul R6, R7        /R6 ← (R6) * R(7)/
```

(1) (4%) What dependencies exist in the program?

(2) (6%) Show the pipeline activity for this program on the processor of Figure 1 using in-order issue with in-order completion policies and using a presentation similar to Figure 2.

(3) (5%) Repeat for in-order issue with out-of-order completion.

(4) (5%) Repeat for out-of-order issue with out-of-order completion.

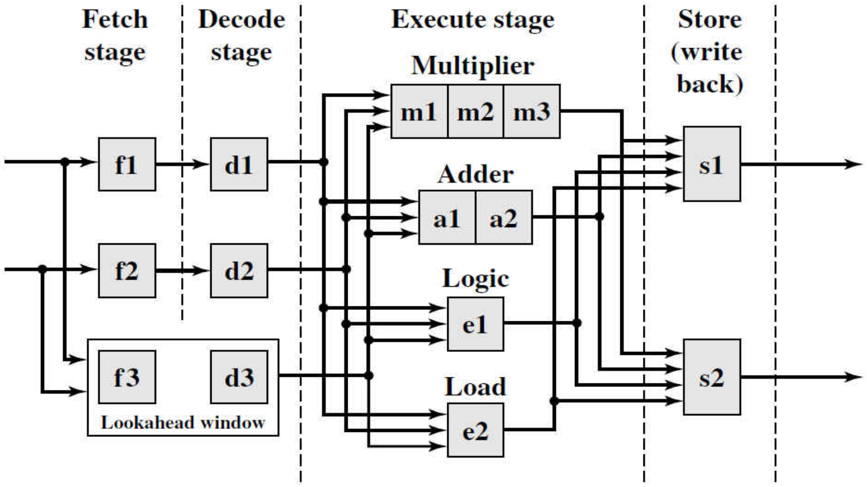

Figure 1

I1	f1	d1	e2	s1				
I2	f2	d2	▨	a1	a2	s2		
	I3	f1	d1		•••			
	I4	f2	d2		•••			•••
			⋮					

Figure 2

6. (15%) An earlier version of the IBM mainframe, the S/390 G4, used three levels of cache. As with the z990, only the first level was on the processor chip [called the processor unit (PU)]. The L2 cache was also similar to the z990. An L3 cache was on a separate chip that acted as a memory controller, and was interposed between the L2 caches and the memory cards. The following table shows the performance of a three-level cache arrangement for the IBM S/390.The purpose of this problem is to determine whether the inclusion of the third level of cache seems worthwhile. Determine the access penalty (average number of PU cycles) for a system with only an L1 cache, and normalize that value to 1.0. Then determine the normalized access penalty when both an L1 and L2 cache are used, and the access penalty when all three caches are used. Note the amount of improvement in each case and state your opinion on the value of the L3 cache.

Memory Subsystem	**Access Penalty (PU cycles)**	**Cache Size**	**Hit Rate (%)**
L1 cache	1	32 KB	89
L2 cache	5	256 KB	5
L3 cache	14	2 MB	3
Memory	32	8 GB	3

7. Please answer the following questions.

(1) (6%) What are the differences among direct mapping, associative mapping, and set associative mapping for a cache?

(2) (6%) What are the advantages of floating-point representation compared with fixed-point representation? What are the disadvantages of floating-point operations compared with fixed-point operations?