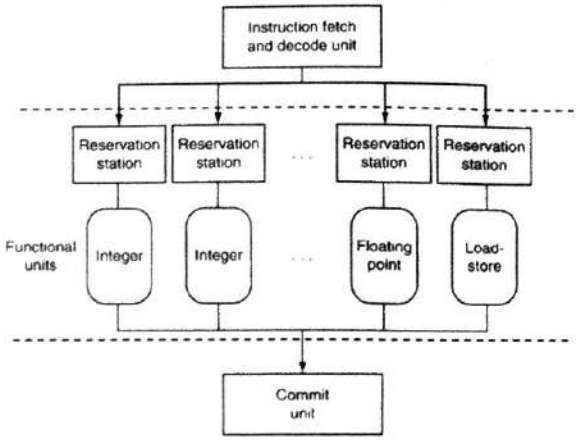
國立交通大學 資訊工程系 計算機組織 第二次考試 (2020/6/18 13:10PM~15:20PM)

請注意：1. 不可看書及任何參考資料(Close book)。 2. 請詳述設計或計算過程(detail your design or computation process)。 3. 請依題號作答，否則高題號答案出現後，之後的低題號答案將不予計分(answer each problem in problem sequence. The answers of small ID problems that appear behind the answers of large ID problems will be ignored)。

1. SuperScalar CPU and hazard (14%)

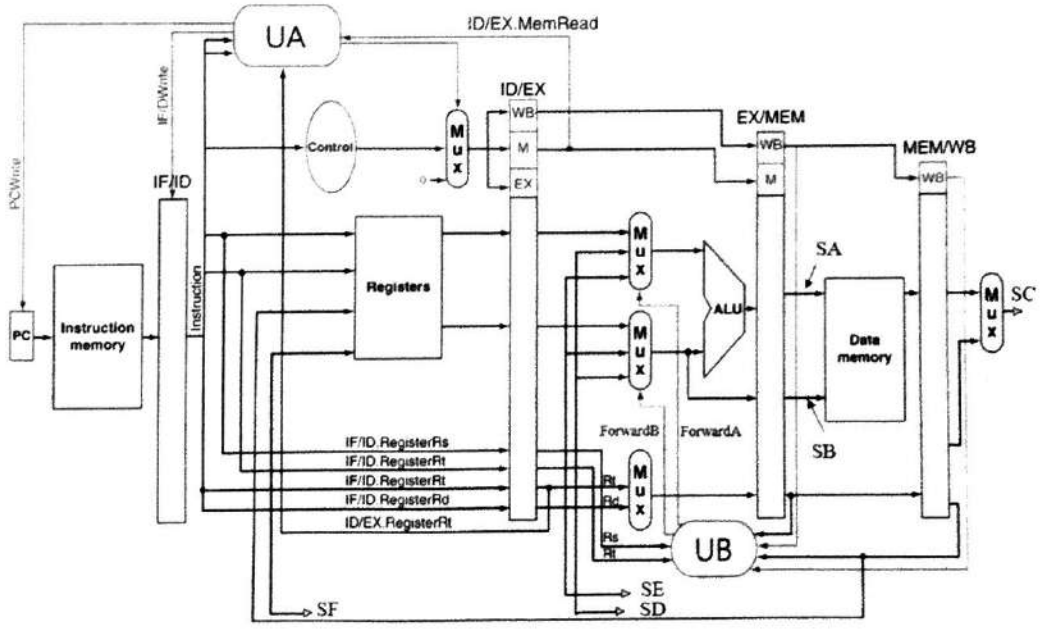


Left figure shows a superscalar CPU with two integer units, one floating-point unit and one load/store unit.

(a). (8 pts) Describe the operations of superscalar CPU. Suggest to categorize the operations into three stages: issue (2 pts), execution (4 pts), and commit (2 pts). In each stage, you need to detail the operations, including the instruction ordering topic at each stage (fetch, issue, execution, and commit).

(b). (6 pts) Below is a sequence of codes that are fetched from memory and going to be issued into the superscalar CPU for execution. Assume currently all functional units of CPU are free. List one example of structure, data, and control hazards for the execution of this code sequence. (*add* \$3, \$1, \$2; *add* \$5, \$4, \$1; *add* \$1, \$2, \$4; *lw* \$6, 10(\$1); *bne* \$6, \$zero, *Loop*1). You need to list the instructions that form some sort of hazard and illustrate the reason to have the hazard.

2. Hazard-Aware Pipelined CPU (23%)



The above figure shows an incomplete hazard-aware pipelined CPU, i.e., it can detect hazard and then determine if a forwarding scheme must be applied to resolve the hazard or a stall must be taken to avoid incorrect execution.

(a). (3 pts) Complete the connection of the design. You need to connect signals *SE*, *SD*, and *SF* to *SA*, *SB*, or *SC*.

(b). (5 pts) Consider the code sequence (*lw* \$1, 4(\$2); *add* \$3, \$1, \$2; *add* \$3, \$3, \$4; *add* \$3, \$5, \$3; *add* \$3, \$6, \$3;). These five instructions are numbered from 1 to 5. Which box (UA or UB) can resolve the hazard between instructions 1 and 2 (2 pts)? List the condition to detect this sort of hazards (3 pts).

(c). (5 pts) Which box can resolve the hazard between instructions 4 and 5 (2 pts)? List the condition to detect this sort of hazards (3 pts). The following signals that may be used in problems (b) and (c) are as follows. Notably, they may not include all signals you need.! MEM/WB.RegWrite, ID/EX.RegisterRt, MEM/WB.RegisterRd, ID/EX.RegisterRt, EX/MEM.RegWrite, IF/ID.RegisterRt, EX/MEM.RegisterRd, EX/MEM.RegisterRd, ID/EX.RegisterRs, ID/EX.MemRead, MEM/WB.RegisterRd, ID/EX.RegisterRs, IF/ID.RegisterRs.

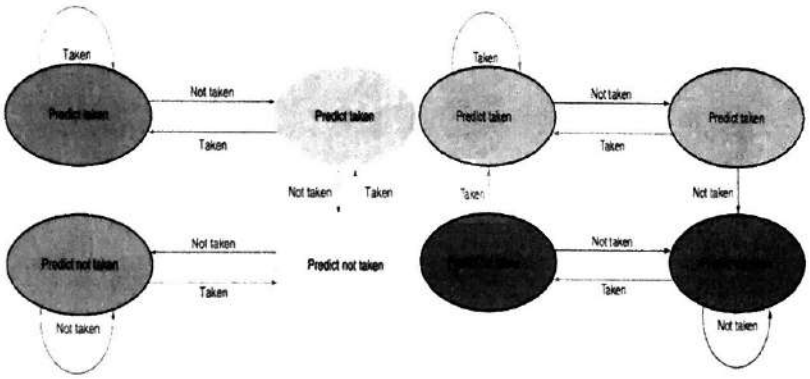
(d). (6 pts) Which three actions are required to stall the "*use*" instruction of load-use hazard by one cycle and why?

(e). (4 pts) Assume the first instruction of the above code sequence enters the CPU at clock cycle 1. Which clock cycles does the third instruction stay at IF stage?

3. Prediction and Speculation (13%)

(a). (3 pts) Consider the code sequence (*sw* \$4, 0(\$5); *lw* \$2, 0(\$8);). Explain how to use speculation to improve the performance of multiple-issue CPU.

(b). (4 pts) The situation becomes complex if an exception occurs on speculated instructions, implying that the exception might not occur without speculation scheme. Use the code sequence of (a) to illustrate this sort of situation (2 pts). Illustrate the way to resolve this problem (2 pts).




```
Codes for computing m;
for (k = 0; k < m; k++) {
   Codes for computing n and n >=0;
   for (j = 0; j <= n; j++) {
      inner loop;
}}
```

(c). (2 pts) The left two of the above figures show two different schemes of 2-bit predictors for branch instructions. Illustrate the differences between these two schemes.

(d). (4 pts) Consider the codes at the right of the above figure with a two-level loop. The iteration number of inner loop is controlled by a variable *n* that is computed by the codes in the outer loop. Variable *n* must be non-negative integers. The behavior of inner loop is that *n* is most likely larger than 0 and once *n* is set as 0, *n* will not be 0 again in next iteration. Which 2-bit predictor can produce better accurate prediction rate on inner loop and why (no credit without reason)?

4. Memory Addressing for Hierarchical memory (35%)

Consider a 32-bit computer with a hierarchical memory system consisting of a disk, a 2^{32}-byte main memory using byte-addressing on PCB, and a two-level cache with primary cache of 2^{17}-byte data inside

CPU. The OS adopts a 32-bit virtual address with a page size of 8KB.

(a). (6 pts) How many pages are there in this virtual memory (2 pts)? How many bits are used for the fields of page offset and virtual page number, respectively (4 pts)?

(b). (6 pts) The primary cache is designed as follows. It is a 2-way set associated cache. Each block is 16-word long and the replacement policy is LRU. How many bits are used for the fields of tag, set, and offset (including byte and word offset)?

(c). (7 pts) What is the disadvantage of using virtually address to access cache (2 pts)? What is the advantage of the physically tagged and virtually indexed cache (2 pts)? Is the physically tagged and virtually indexed cache suited to this hierarchical memory system and why (3 pts)?

(d). (10 pts) Assume the disadvantage of virtually addressed cache has been resolved, we apply virtually addressed cache in this computer. What are the results of using the following virtually address accesses to access cache if the cache is empty initially? 0x4A08 CA10, 0x2731 CA08, 0x2731 CA10, 0x4A10 CA08, 0x4A08 CA38. You need to write the values of tag, set, and offset to show why an access is hit or miss to get full credit. No credit is given if you do not list these detailed information.

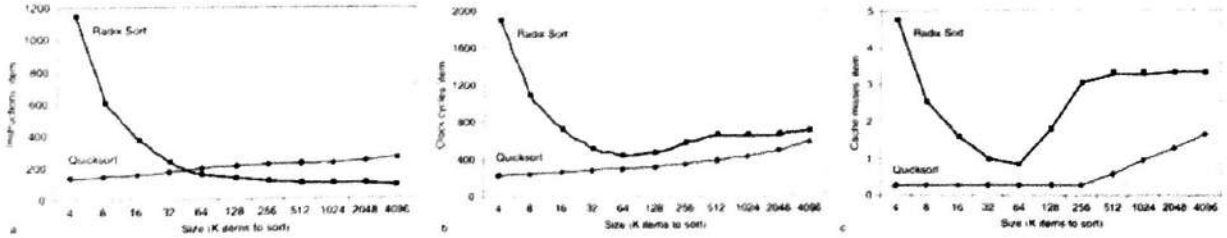
(e). (6 pts) If the CPU base CPI = 1 and the CPU clock is running at 5GHz. The miss rate per instruction is 2% and the miss penalty is 200*ns*. If the cache has only primary cache without the second-level cache, what is the effective CPI (3 pts)? For the 2-level cache, the access time to the second-level cache is 4*ns* and the global miss rate to main memory is 0.4%. What is the performance ratio in terms of effective CPI (3 pts)?

5. **Data Coherence for Multi-Core Computer (7%)**

(a). (3 pts) Consider the data coherence issue for a multi-core computer with local cache inside each core. If core $C1$ writes $d1$ to Address X, core $C2$ writes $d2$ to Address X and then core $C3$ writes $d3$ to Address X, please describe the system behavior we need to maintain, as all other cores access Address X, to make data coherent among cores.

(b). (4 pts) Please describe how the snooping protocol operates to maintain data coherence.

6. **Algorithm, Cache, and Performance (4 pts)**



The above three tables compare the instruction number per item, required clock cycle per item, and cache missed per item between radix sort and quick sort. Explain the reasons to get this comparison result.

7. **Three Types of Caches Misses (6%)**

How to decrease compulsory misses, capacity misses, and conflict misses (3 pts)? What are the side-effects of your methods (3 pts)?

8. **SEC/DED Hamming Codes (6%)**

(a). (2 pts) Consider the Hamming SEC $p_1p_2d_1p_3d_2d_3d_4$, one received data is 0001001, List the

computation procedure to show it has no error or where the error is.

(b). (4 pts) Extend previously correct data of SEC to SEC/DED by adding p_4 in the end of data and send it to us. If we receive a data of 10110010, list the computation procedure to show it is correct or what kind of error it has.