資訊工程學系 計算機結構 期末考(98 下)

姓名：　　　　　　　　　　　　學號：

[**Instruction Level Parallelism its Exploitation**] (20%)

12. (2%) Please illustrate an example code for a WAR hazard existing between a load and a store.

```
ld F0, 0(R1)
sd F2, 0(R1)
```

13. (5%) Please explain what *hardware speculation* is.

Overcome control dependence by hardware speculating on outcome of branches and executing program as if guesses were correct

14. (6%) When does an instruction go to the commit step? What behavior is done in the commit step of an instruction?

When an instruction is at head of reorder buffer and the result present, then the register is updated with the result (or store to memory) and removes the instruction from the reorder buffer.

15. (7%) How does Tomasulo with speculation maintain a precise exception model? You can use an example to explain your concept.

For example, if the MUL.D instruction caused an interrupt, 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.

In the example using Tomasulo' s algorithm, the SUB.D and ADD.D instructions could both complete before the MUL.D raised the exception.

```
L.D  F6, 32(R2)
L.D  F2, 44(R3)
MUL.D   F0, F2, F4
SUB.D   F8, F6, F2
DIV.D   F10, F0, F6
ADD.D   F6, F8, F2
```

[Limits to ILP] (20%)

16. (10%) Please describe Fine-Grained Multithreading and its corresponding advantages and disadvantages.

It switches between threads on each instruction, causing the execution of multiples threads to be interleaved. It is usually done in a round-robin fashion, skipping any stalled threads. Besides, CPU must be able to switch threads every clock.

- **Advantage is it can hide throughput losses that arise from both short stalls, since instructions from other threads executed when one thread stalls**
- **Disadvantage is it slows down execution of individual threads, since a thread ready to execute without stalls will be delayed by instructions from other threads**

17. (10%) Please describe Coarse-Grained Multithreading and its corresponding advantages and disadvantages.

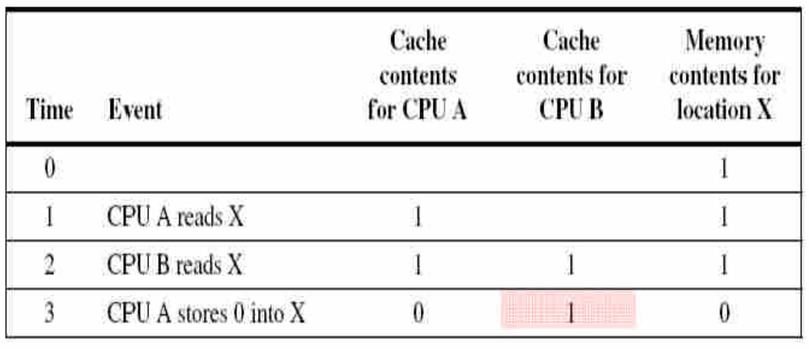
Switches threads only on costly stalls, such as L2 cache misses

- **Advantages**
  - **Not need to have very fast thread-switching**
  - **Doesn't slow down thread, since instructions from other threads issued only when the thread encounters a costly stall**
- **Disadvantage is hard to overcome throughput losses from shorter stalls, due to pipeline start-up costs**
  - **Since CPU issues instructions from 1 thread, when a stall occurs, the pipeline must be emptied or frozen**
  - **New thread must fill pipeline before instructions can complete**

[Multiprocessors and Thread-Level Parallelism] (42%)

7. (6%) What is the cache coherence problem of a multiprocessor systems?

When the memory data item is held in the individual cache of two processor, two different values are saw.



Time	Event	Cache contents for CPU A	Cache contents for CPU B	Memory contents for location X
0				1
1	CPU A reads X	1		1
2	CPU B reads X	1	1	1
3	CPU A stores 0 into X	0	1	0

8. (6%) In the directory protocol, there are three kinds of nodes, i.e., Local node, Home node, and Remote node. What roles do the nodes play in the protocol?

- **Local node where a request originates**
- **Home node where the memory location and the directory entry of an address reside**
- **Remote node has a copy of a cache block, whether exclusive or shared**

9. (7%) Conference miss can be divided into two categories, i.e., true sharing misses and false sharing misses. What is "false sharing miss"? Please provide a method to eliminate all false sharing misses.

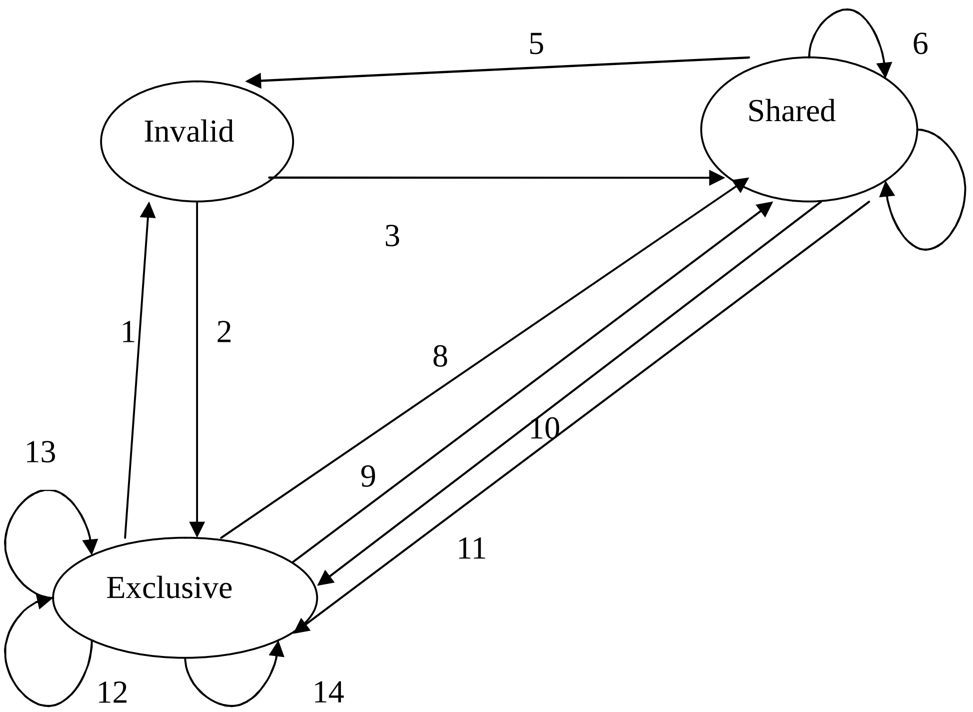
False sharing misses when a block is invalidated because some word in the block, other than the one being read, is written into

i. Invalidation does not cause a new value to be communicated, but only causes an extra cache miss

Block is shared, but no word in block is actually shared

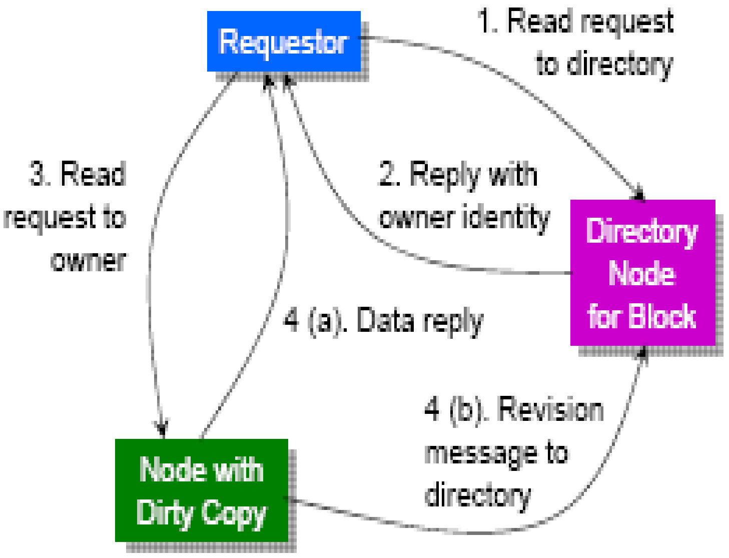
⇒ miss would not occur if block size were 1 word

10. (12%) A write-back cache use the write-invalidate cache coherence protocol. In the **directory protocol**, the cache controller must maintain the state transitions for each cache block. Please complete the state transition diagram for processor requests and bus requests. Please write the request and the corresponding function for each edge in the following diagram. (one point for each row)



Index	Request type (Please note CPU request)	Function/Action
1	Fetch/invalidate	Data write back
2	CPU write miss	Not listed
3	CPU read miss	Not listed
5	Invalidate	Not listed
6	CPU read hit	Not listed
7	CPU read miss	Not listed
8	CPU read miss	Data write back, place read miss on bus
9	Fetch	Data write back
10	CPU write hit	Send invalidate message
11	CPU write miss	Not listed
12	CPU write hit	Not listed
13	CPU read hit	Not listed
14	CPU write miss	Data write back

11. (5%) In the original directory protocol, a home node can be the performance bottleneck. Please propose an idea to improve the performance for the case where a locale node has a read miss for a dirty block in a remote node.



12. (6%) Please compare Write invalidate protocol and Write update protocol.
    A. Write invalidate protocol
        i. To ensure that a processor has exclusive access to a data item before it writes that item.
        ii. It invalidates other copies on a write.
    B. Write update or write broadcast protocol:
        iii. To updates all the cached copies of a data item when that items is written.
        iv. Because a write update protocol must broadcast all writes to shared cache lines, it consumes considerably more bandwidth.

[Memory Hierarchy Design] (29%)

13. (12%) Please discuss the following compiler optimization techniques that can improve the cache performance. You can give an example code and explain it
    - ***Merging Arrays*: improve spatial locality by single array of compound elements vs. 2 arrays**
    - ***Loop Interchange*: change nesting of loops to access data in order stored in memory**
    - ***Loop Fusion*: Combine 2 independent loops that have same looping**

and some variables overlap

- ***Blocking*****: Improve temporal locality by accessing "blocks" of data repeatedly vs. going down whole columns or rows**

14. (8%) There are numerous techniques for improving the performance of caches. Please describe the main idea behind the following techniques, i.e., nonblocking caches and critical word first.

甲、*Non-blocking cache*

1. allows data cache to continue to supply cache hits during a miss

乙、*Critical Word First*—Request the missed word first from memory and send it to the CPU as soon as it arrives; let the CPU continue execution while filling the rest of the words in the block

15. (3%) Please write the equation of the average memory access time.

$$\text{Average Memory Access Time} = \text{Hit Time} + \text{Miss Rate} \times \text{Miss Penalty}$$

16. (6%) There are numerous techniques for improving the performance of caches. Some reduce the frequency of misses, some reduce the miss penalty, some reduce hit times, and so on. Many optimizations involve trade-offs, making one performance factor worse in return for reducing another. In some cases, the so-called optimization can actually hurt performance. For the following optimizations, briefly summarize the trade-offs involved and how it helps or can hurt performance.

甲、 **Higher associativity**

i. Reduce conflict misses

ii. Increase hit time

乙、 **Larger block size**

i. Reduce compulsory misses

ii. Increase miss penalty

丙、 **Bigger caches**

i. Reduce capacity misses

ii. Potentially longer hit time