

CS 5513 Fall 2009 Midterm Exam

Mark your answer for each question on the SCAN-TRON form provided. Mark only one answer per question. If you believe the right answer is not among the options given, choose the closest answer. Write your name on the SCAN-TRON. You must turn in your question paper along with your SCAN-TRON form.

True (A) or False (B)

1. Increasing the associativity of a cache tends to decrease the number of misses due to conflicts.
2. Virtually tagged caches are infeasible because the same virtual address in two different processes would refer to two different physical addresses.
3. The RISC-like instruction set from the homework has a special instruction to read a character from the keyboard.
4. Write-through caches must keep dirty bits.
5. A cache miss is an example of an exception.
6. Page table entries contain dirty bits.
7. A branch history table (BHT) predictor with n two-bit entries is likely to be more accurate than one with n one-bit entries.
8. Pipelining decreases the amount of time it takes to execute an instruction.
9. RAW hazards can be eliminated by register renaming.
10. It is possible that a cache with a higher miss rate can still deliver improved performance over a cache with a lower miss rate.

Multiple Choice

11. **One consequence of Amdahl's law is:**
 - (a) Performance is inversely proportional to execution time.
 - (b) The maximum speedup for an optimization that can be applied to 50% of a program's cycles is 2.0.
 - (c) The CPI for a processor with one stall cycle for all branches is the ideal CPI plus the fraction of instructions that are branches.
 - (d) Store buffers can improve performance by up to 20% for a write-through cache.
12. **Machine A has a clock rate of 1 GHz and a cache miss rate of 5%. Machine B has a clock rate of 2 GHz and a cache miss rate of 10%. Loads are 25% of all instructions. Only loads can cause cache misses. All instructions execute in one cycle except for loads that miss in the cache. Cache miss penalty is 100 nanoseconds. Which machine is faster, and by how much (rounding to the nearest percent)?**
 - (a) Machine B is faster by 29%.
 - (b) Machine A is faster by 56%.
 - (c) Machine A is faster by 33%.
 - (d) Machine B is faster by 19%.

13. Consider a virtual memory system with 48-bit virtual addresses and 32-bit physical addresses. Pages are 64KB. The cache has a block size of 64 bytes, is 4-way set associative, and uses physical tags. What is the maximum capacity of the cache?

- (a) 256KB
- (b) 64KB
- (c) 4KB
- (d) There is not enough information given to answer the question.

14. Consider a pipeline with the following six stages: IF (fetch), ID (decode), RR (read registers), EX (execute), MM (memory), WB (write-back registers). The decode stage does not read registers; rather, registers are written in the first half of the WB stage and read in the second half of the RR stage. The other stages behave as they normally do in the classic 5-stage pipeline. How many cycles does the following code take, assuming no forwarding?

```
DADD  R1, R7, R8
DSUB  R2, R1, R9
DADD  R3, R1, R2
```

- (a) 12 cycles.
- (b) 9 cycles.
- (c) 7 cycles.
- (d) 14 cycles.

15. What would be the answer to the previous problem if we allow forwarding?

- (a) 8 cycles.
- (b) 7 cycles.
- (c) 6 cycles.
- (d) 11 cycles.

16. In the classic 5-stage pipeline with the ID stage reading registers in the second half of the cycle and the EX stage writing them in the first half, and with no forwarding, how many cycles will the following take?

```
DADDI  R1, R1, #4
LD      R2, 0(R1)
DADD   R4, R2, R6
ST      R4, 0(R1)
```

- (a) 12 cycles.
- (b) 9 cycles.
- (c) 7 cycles.
- (d) 14 cycles.

17. What would be the answer to the previous problem if we allow forwarding?

- (a) 7 cycles.

```

int add (double A[64][64], double B[64][64]) {
    int    i, j;

    for (i=0; i<64; i++)
        for (j=0; j<64; j++)
            A[i][j] = A[i][j] + B[i][j];
}

```

Figure 1: Code for Problems 18, 19, and 20.

```

1      ADD    R1, R0, R0      ;; i = 0
2 L1:   SUBI   R2, R1, #100   ;; R2 = i - 100
3      BEQ   R2, R0, L2      ;; if (R2 == 0) goto L2
4      ADDI  R3, R3, #1      ;; A = A + 1
5      J     L3              ;; goto L3
6 L2:   ADDI  R4, R4, #1      ;; B = B + 1
7 L3:   ADDI  R1, R1, #1      ;; i = i + 1
8      SUBI  R2, R1, #200    ;; R2 = R1 - 200
9      BNE  R2, R0, L1      ;; if (R2 != 0) goto L1

```

Figure 2: Code for problems 25 and 26.

- (b) 8 cycles.
- (c) 9 cycles.
- (d) 10 cycles.

18. Consider a fully-associative cache with 4096 blocks, a block size of 32 bytes, and an LRU replacement policy. Suppose every cache block is initially invalid, and then a call is made to the C code from Figure 1. How many compulsory misses will occur? (double is 8 bytes.)

- (a) None.
- (b) 1024.
- (c) 2048.
- (d) 4096.

19. For the previous problem, how many capacity misses will occur?

- (a) None.
- (b) 1024.
- (c) 2048.
- (d) 4096.

20. For the previous problem, how many conflict misses will occur?

- (a) None.
- (b) 1024.
- (c) 2048.
- (d) 4096.

21. Which of the following is NOT a reason that a

5-stage pipeline fails to achieve a speedup of 5?

- (a) Pipeline registers contribute extra latency not present in a non-pipelined machine.
- (b) Pipeline hazards can introduce stall cycles.
- (c) Programs have data-dependent branches that cannot be predicted with 100% accuracy.
- (d) Executing branches in the ID stage causes a structural hazard with the branch predictor.

22. What is Moore's Law?

- (a) The number of devices on an integrated circuit tends to grow exponentially with time.
- (b) The number of transistors per unit area tends to grow exponentially with time.
- (c) The clock rate of microprocessors tends to grow exponentially with time.
- (d) The performance of computers (i.e. the inverse of execution time) tends to grow exponentially with time.

23. How much instruction-level parallelism is there in average programs?

- (a) About 10 to 20.
- (b) About 1 to 2.
- (c) About 5 to 7.
- (d) About 30 to 40.

24. What is a basic block?

- (a) It is a sequence of instructions beginning with a label and ending with a branch.
- (b) It is a sequence of instructions with one entry, multiple

exits, but no internal control flow.

(c) It is a sequence of instructions with one entry and one exit.

(d) It is a sequence of instructions with at most four floating point operations and one branch.

25. Consider the code in Figure 2 in the machine language of homework #2:

Suppose that we use a static branch prediction policy of prediction all branches as taken. Approximately what is the branch misprediction rate for this code?

(a) at most 1%

(b) 25%

(c) 50%

(d) 100%

26. For the previous question, suppose we use a dynamic branch predictor with a branch history table with one bit entries, with each bit initialized to 1. What is the approximate misprediction rate?

(a) at most 1%

(b) 25%

(c) 50%

(d) 100%

27. Suppose we move from 4KB pages to 8KB pages. Which of the following is NOT a likely result?

(a) TLB misses will decrease.

(b) Cache misses will increase.

(c) Memory fragmentation will increase.

(d) The number of sets in the cache may increase.

28. Which of the following is a benefit of write-back caches?

(a) Increased design complexity.

(b) Reduced traffic on the memory bus.

(c) Simpler debugging of the cache hardware.

(d) Fewer tag bits in the cache.

29. What is a disadvantage of the LRU replacement policy?

(a) It has the lower miss rate of all replacement policies.

(b) It is difficult to implement for high associativities.

(c) It relies on a pseudo-random number generator.

(d) It is non-deterministic.

30. In a floating point pipeline with multi-cycle latencies, can the execution stages of some instructions be overlapped? If so, how? If not, why not?

(a) Yes, as long as the related functional units are pipelined or independent and the instructions do not overlap during the initiation period.

(b) Yes, as long as the related instructions do not have data dependences with previous instructions and the functional units are pipelined.

(c) No; even though the functional units are independent and pipelined, the instructions must complete in order so only one instruction may be in the execute stage at any given time.

(d) No; pipelined functional units suffer the same branch prediction problems as traditional pipelines, but there is no recovery mechanism in case of a misprediction due to asynchronous delivery of data over the bypass network.