GD

INDIAN INSTITUTE OF TECHNOLOGY, KHARAGPUR

End-Spring Semester 2016-17

Date of Examination : ________________ **Session (FN/AN)** ________ **Duration: 3 hrs**

Subject No. : CS60003 **Subject Name : High Performance Computer Architecture**

Department : Computer Science and Engineering

Specific charts, graph paper, log book etc.: Use of simple non-programmable calculators is permitted.

Special Instructions:

- **Answer all questions.**
- **No clarification to any of the questions shall be provided. In case you feel anything is amiss, you can make suitable assumptions. But, please write down your assumptions clearly.**
- All answers should be brief and concise. Lengthy and irrelevant answers will be penalized.

1. Assume that you have a computer with a 16-core processor. In this computer, the number of cores to be used for executing a program can be set as a compiler parameter. Assume that for a certain program, 1% of the code is purely sequential and the rest of the code is ideally parallel. You need to achieve a speed-up of 10 over the single core performance of this program, while at the same time you should use the minimum number of cores of the computer. How many cores would you configure the compiler to use? Show details of your workout.**[2]**

2. Write one advantage each of a physically tagged, virtually indexed (PTVI) cache over: (a)A virtually tagged, virtually indexed (VTVI) cache, and (b)A Physically tagged, physically indexed (PTPI) cache. Write at most one simple sentence for each. **[2]**

3. Using one simple sentence explain why the **pseudo associative cache technology** that reduces the hit time of an associative cache while maintaining its low miss rate, is rarely used in an L1 cache. **[2]**

4. What is the asymptotic prediction accuracy of a 2 bit predictor which was initialized to strongly not taken before the start of the program for predicting the outcome of a branch whose behavior is the following infinitely repeating pattern .... **TTNTTNTT**... **[2]**

5. Give one example computation (code not necessary --- just describe the operations of the computation) over a shared memory multiprocessor for which the **write-invalidate** based cache coherence protocol would perform better and another example computation for which the **write-update** based solution would perform better. **[2]**

6. Suppose a dual core processor is running a program with two threads T0 and T1. The following pieces of code are executed by the threads in the order shown below, with the side effects listed:
   - T0: x = 12;(T0 writes x, invalidates copy of that block in P1's cache)
   - T1: a = b+2;(T1 experiences cache miss while reading b)

   Assume that the cache miss for T1 is a result of the invalidation due to the cache write by T0. Under what conditions is the cache miss for T1 would be considered be a true sharing miss? Under what conditions would it be considered miss a false sharing miss? **[1+1]**

7. Consider a system with a two-level cache hierarchy. The L2 cache is 128KB, 8-way set-associative with a block size of 128 bytes. To cut down on the number of main memory (DRAM) accesses, the system designers are exploring the following two improvements:

   Optimization-1: Add one way to each set in the L2 cache, resulting in a 144KB 9-way set-associative L2 cache.

Optimitization-2: Add a 128-entry fully-associative victim cache.

Which one of the above two optimizations would be more effective in reducing the number of DRAM accesses? Explain your answer using one or two sentences? **[1+2]**

8. Copy the following table to your exam paper and fill in the blank cells in the table with the most appropriate statement(s) [Just filling any one or more of (a) to (j) per cell should be fine] from the statements given below, so that the table correctly summarizes the advantages and disadvantages of a design change on the cache performance.**[3]**

(a) Decreases capacity misses	(f) Increases capacity misses
(b) Decreases conflict misses	(g) Increases conflict misses
(c) Decreases compulsory misses	(h) Increases compulsory misses
(d) Decreases access time	(i) Increases access time
(e) Decreases miss penalty	(j) Increases miss penalty

Design Change	**Potential advantage**	**Possible Disadvantage**
Increase cache size		
Increase block size		
Increase associativity		

9. The code given below forms the core of an embedded computation. It computes the average of the contents of an array ***A***.

```
total = 0;
for(j=0; j < k; j++) {
        sub_total = 0;
        for(i=0; i < N; i++) {/* Nested loops to avoid overflow */
                sub_total += A[j*N + i];
        }
        total += sub_total/N;
}
average = total/k;
```

When designing a cache to run this program most effectively on the embedded computer, given a constant cache capacity and associativity, should a larger or a smaller block size be preferred? Justify your answer using one short sentence. **[1+2]**

10. Suppose the branch frequencies (as percentage of all instructions) for a certain program are as follows: Conditional branches=15%, Unconditional branches (Jumps and calls) =1%. Of the Conditional branches, 60% are taken. This program is to be run on a four-stage instruction pipeline that does not use any branch or target predictors. In this pipeline, the target computation for unconditional branches is completed at the end of the third stage. For conditional branches, the branch outcome is also determined at the end of the third stage. The target computation for the taken conditional branches is completed at the end of the fourth stage. An instruction is decoded at the second stage. If an instruction is determined to be a branch, the last instruction fetched is flushed and no further instructions are fetched till the branch is resolved. How much faster would the program run if perfect branch and target predictors are used? **[4]**

11. A 2GHz pipelined computer uses a split single level cache. It exhibited an average CPI of 2, when it was simulated with perfect instruction and data caches and perfect branch and target predictors. Later a cache hierarchy as well as branch and target predictors were designed and integrated with the processor. On this computer, while executing a program having 20% branch instructions and 30% memory access instructions, the following were observed: 1% instruction cache miss, 10% data cache miss, and 10% branch miss-predictions. The main memory access time is 20ns, and branch miss-prediction penalty is 1 cycle. What is the actual CPI? **[4]**

12. Restructure the following loop using the software pipelining technique to improve the ILP. It is not required to write the prologue and epilogue. Please don't convert the given program statements into assembly language instructions. **[4]**

```
for(k=0;k<100;k=k+1){
        X[k] = X[k] +1 ;
        Y[k] = Y[k] + X[k] ;
        Z[k] = Y[k] + 2 ;
}
```

13. The ideal CPI for a processor when all memory requests are cache hits is 1 cycle. The average memory access time (AMAT) is 3 clock cycles. The L1 cache miss penalty is 80 clock cycles. Designers are trying to achieve a 50% improvement to AMAT by adding an on-chip L2 cache. The L2 cache hit time is 6 clock cycles and the miss penalty is 80 cycles. To obtain the desired speedup, how often must data be found in the L2 cache? **[5]**

14. A 50 MHz MIPS 5 stage pipelined processor uses split instruction and data caches. The word size of the processor is 32 bits.

- The instruction cache is fully associative, with a total capacity of 4KB (4096 bytes) and a line (block) size of 2 words. The instruction cache access time is 20 ns.
- The data cache is 2-way set associative with a line size of 4 words and a total capacity of 4KB (4096 bytes). The data cache access time is 20 ns.
- Main memory access time is 60 ns/word. In both caches, a missed word is not passed to the processor until the entire line is received from main memory.

The processor is executing a sequential code block containing 400 non-branching instructions and each instruction is 1 word long. The instructions are stored contiguously in memory. The instruction cache is empty at the start of the code execution. Assume the hit rate of the data cache during the execution of this code is 100%. Also, assume no hazards occur while executing the code fragment; stalls only occur due to memory access.

a) When an instruction is not found in the instruction cache, on the average how many stall cycles occur per instruction? **[2]**

b) What is the hit rate of the instruction cache while executing this code block? **[3]**

15. Consider the following memory hierarchy for a processor:

- Level 1 cache: 90% hit rate, 1-cycle hit time.
- Level 2 cache: 98% hit rate, 15-cycle hit time.
- Main memory: 140-cycle hit time.

a) What is the average memory access time for the given memory system? **[2]**

b) Assume that the L2 cache is write-allocate and write-back and that there is no write buffer. If 10% evicted blocks are dirty, how many cycles do L2 write-back operations add to the average memory access time? **[4]**

16. You are building a computer system with an in-order execution processor that runs at 1GHz. The processor achieves a CPI of 1, when the program does not contain any memory access instructions. The memory system has split L1 cache. Both the I-cache and the D-cache are direct mapped and hold 32KB each with block size 64 bytes. The I-cache has a 2% miss rate, and the D-cache is write-through with 5% miss rate. The hit time for both the I-cache and the D-cache is 1ns. The L2 cache is a 512KB unified write-back cache having block size of 64 bytes. The hit time of the L2 cache is 15ns. The local hit rate of the L2 cache is 80%. The 64-bit wide main memory has an access latency of 20ns, after which the words may be transferred at the rate of one bus word (64-bit) per bus cycle on the 64-bit wide 100MHz main memory bus. Compute the CPI considering memory accesses. **[6]**

17. Consider that MSI protocol is used for cache coherency in a bus-based multiprocessor system with four processors: P1, P2, P3, and P4. Each processor has a direct-mapped cache. The size of a cache line is one word. The following sequence of memory operations by different processors access two memory locations (words): A and B, that are mapped to the same cache block. Copy the following table to your examination answer paper and fill-in the blank cells showing for each operation for a processor (first column of the table): the state of the cache line in each processor (use M: modified, S:shared, I:invalid), and the bus requests caused by the MSI protocol(read miss, write miss, write back). Initially A= 1, B= 1. **[10]**

Operation by Processor	P1	P2	P3	P4	Bus request(s)
P4 reads A					
P3 reads A					
P2 writes B=2					
P1 writes A=3					
P2 reads A					
P1 reads A					
P3 reads A					
P4 writes A=5					
P1 writes A=7					
P2 reads A					

---- The End----