INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, SRI CITY

TERM-III EXAMINATION – SPRING 2024

Computer Architecture

CSE:UG1(PC) **Date: 23-04-2024**

Duration: 90 Mins (09:00-10:30 AM) **Max. Marks: 30**

Roll No:_____ **Section:** _____

Instructions:

1. All questions are compulsory.
2. Write the answers legibly.
3. All sub-parts of a question should be written together.
4. **Attach the question paper with the answer sheet.**

Section-A (Multiple Objective Questions)

1	A five-stage pipeline has stage delays of 150,120,150,160 and 140 nanoseconds. The registers that are used between the pipeline stages have a delay of 5 nanoseconds each. The total time to execute 100 independent instructions on this pipeline, assuming there are no pipeline stalls, is _____ nanoseconds. a. 17160 b. 16640 c. 17640 d. 15000	**[1 Mark]**
2	Consider a non-pipelined processor operating at 2.5 GHz. It takes 5 clock cycles to complete an instruction. You are going to make a 5- stage pipeline out of this processor. Overheads associated with pipelining force you to operate the pipelined processor at 2 GHz. In a given program, assume that 30% are memory instructions, 60% are ALU instructions and the rest are branch instructions. 5% of the memory instructions cause stalls of 50 clock cycles each due to cache misses and 50% of the branch instructions cause stalls of 2 cycles each. Assume that there are no stalls associated with the execution of ALU instructions. For this program, the speedup achieved by the pipelined processor over the non-pipelined processor (round off to 2 decimal places) is _____ Answer (2.16)	**[1 Mark]**
3	Memory Aliasing is _____. a. Two different memory references specify two location b. Two different memory references specify single location c. Both (a) and (b) d. None of the above	**[1 Mark]**
4	A superscalar processor can issue and execute_____instructions in one cycle. a. Multiple	**[1 Mark]**

	b. Single c. Both (a) and (b) d. None of the above	
5	Code motion can_____. a. Reduce frequency with which computation performed b. Increase frequency with which computation performed c. Both (a) and (b) d. None of the above	**[1 Mark]**
6	The below shows refined disassembler code for procedure <multstore> and <mult2>. What are the contents of the register %rsp and %rip when call to <mult2> is performed. 0000000000400540 <multstore>: • • 400544: callq 400550 <mult2> 400549: mov %rax,(%rbx) • • 0000000000400550 <mult2>: 400550: mov %rdi,%rax • • 400557: retq a. 400544 and 400549 b. 400549 and 400550 c. 400549 and 400544 d. 400550 and 400549	**[1 Mark]**
7	_____ occurs when the set of active cache blocks is larger than the cache. a. Cold miss b. Compulsory miss c. Conflict miss d. **Capacity miss**	**[1 Mark]**
8	Which type of parallelism involves breaking down a program into a large number of small tasks? a. Coarse-grained parallelism **b. Fine-grained parallelism** c. Medium-grained parallelism d. Low-grained parallelism	**[1 Mark]**

9	What factor strongly influences the performance gain achieved by utilising multi-core processors? a. Clock speed of individual cores b. Size of the processor package c. Software algorithms and implementation d. Type of network topology used for interconnecting cores	**[1 Mark]**
10	What is a characteristic of tightly coupled multiprocessor systems in the context of MIMD machines? a. Each processing element (PE) has its own local memory. b. Communication between PEs occurs through a single global memory. c. They are more likely to scale compared to distributed-memory MIMD systems. d. Cache coherence is a significant challenge in this type of system.	**[1 Mark]**

Section-B (Descriptive Questions)

1	a. Which architecture is most suitable for fine-grained parallelism? Justify your answer **[2M].** i. Shared memory architecture most suitable for fine-grained parallelism because it has a low communication overhead. b. Describe Flynn's Taxonomy and its importance in classifying parallel computing systems**[3M]** i. Flynn's Taxonomy is a classification system proposed by Michael J. Flynn in 1966 to categorise parallel computing architectures based on the number of instruction streams (I) and data streams (D) **that can be processed concurrently**. It defines four categories of parallel architectures: 1. **Single Instruction, Single Data (SISD):** a single instruction stream is processed by a single processing unit, and each instruction operates on a single data stream.**[0.5M]** 2. **Single Instruction, Multiple Data (SIMD):**SIMD architectures involve a single instruction stream that operates simultaneously on multiple data streams.**[0.5M]** 3. Multiple Instruction, Single Data (MISD): MISD architectures are less common and involve multiple instruction streams operating on the same data stream. Practical implementations of MISD architectures are rare due to the complexity of coordinating multiple instructions on the same data.**[0.5M]** 4. **Multiple Instruction, Multiple Data (MIMD):**MIMD architectures feature multiple instruction streams and multiple data streams. Each processing unit operates independently, executing its own set of instructions on its own data.**[0.5M]**	**[5 Marks]**

<table>
<tr><td></td><td>

- The importance of Flynn's Taxonomy lies in its ability to **provide a structured framework for understanding and categorising parallel computing architectures** based on their concurrency characteristics. Classifying systems into one of the four categories **allows for better comprehension of different architectures and facilitates the selection of the most appropriate** one for specific application requirements **[1M]**

</td><td></td></tr>
<tr><td rowspan="2">2</td><td>

Consider a computer with an 8-bit address space and a direct-mapped 64-byte data cache with 8-byte cache blocks.

A. The boxes below represent the bit-format of an address. In each box, indicate which field that bit represents (it is possible that a field does not exist) by labelling them as follows: B: Block Offset S: Set Index T: Cache Tag

<table>
<tr><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>T</td><td>T</td><td>S</td><td>S</td><td>S</td><td>B</td><td>B</td><td>B</td></tr>
</table>

</td><td>[1 Marks]</td></tr>
<tr><td>

B. The table below shows a trace of load addresses accessed in the data cache. Assume the cache is initially empty. For each row in the table, please complete the two rightmost columns, indicating (i) the set number (in decimal notation) for that particular load, and (ii) whether that loads hits (H) or misses (M) in the cache

<table>
<tr><th>Load No:</th><th>Hex Address</th><th>Binary Address</th><th>Set Number (In Decimal)</th><th>HIT or MISS</th></tr>
<tr><td>1</td><td>43</td><td>0100 0011</td><td>0</td><td>MISS</td></tr>
<tr><td>2</td><td>b2</td><td>1011 0010</td><td>6</td><td>MISS</td></tr>
<tr><td>3</td><td>40</td><td>0100 0000</td><td>0</td><td>HIT</td></tr>
<tr><td>4</td><td>f9</td><td>1111 1001</td><td>7</td><td>MISS</td></tr>
<tr><td>5</td><td>b2</td><td>1011 0010</td><td>6</td><td>HIT</td></tr>
<tr><td>6</td><td>93</td><td>1001 0011</td><td>2</td><td>MISS</td></tr>
<tr><td>7</td><td>d0</td><td>1101 0000</td><td>2</td><td>MISS</td></tr>
<tr><td>8</td><td>b0</td><td>1011 0000</td><td>6</td><td>HIT</td></tr>
</table>

</td><td>[4 Marks]</td></tr>
<tr><td>3</td><td>

Consider a disk pack with the following specifications-

16 surfaces, 128 tracks per surface, 256 sectors per track and 512 bytes per sector.

</td><td></td></tr>
</table>

	Answer the following questions- 1. What is the capacity of the disk pack? **Answer: Capacity of disk pack** **= Total number of surfaces x Number of tracks per surface x Number of sectors per track x Number of bytes per sector** **= 16 x 128 x 256 x 512 bytes** **= 2^{28} bytes** **= 256 MB** 2. What is the number of bits required to address the sector? **Answer: Total number of sectors** **= Total number of surfaces x Number of tracks per surface x Number of sectors per track** **Thus, Number of bits required to address the sector = 19 bits** 3. If the format overhead is 64 bytes per sector, how much amount of memory is lost due to formatting? **Answer: Amount of memory lost due to formatting** **= Formatting overhead** **= Total number of sectors x Overhead per sector** **= 2^{19} x 64 bytes** **= 2^{19} x 2^{6} bytes** **= 2^{25} bytes** **= 32 MB**	**[1 Mark]** **[2 Marks]** **[2 Marks]**
4	Explain the limitations of optimizing compilers.	**[5 Marks]**

- **Operate under fundamental constraint**
  - Must not cause any change in program behavior
  - Often prevents it from making optimizations that would affect behavior under pathological conditions.
- **Behavior that may be obvious to the programmer can be obfuscated by languages and coding styles**
  - e.g., Data ranges may be more limited than variable types suggest
- **Most analysis is performed only within procedures**
  - Whole-program analysis is too expensive in most cases
  - Newer versions of GCC do interprocedural analysis within individual files
    - But, not between code in different files
- **Most analysis is based only on *static* information**
  - Compiler has difficulty anticipating run-time inputs