|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | |  | | | | | |  | |  | | **Reg. No.:** | |  | |
|  | | |  | |  | | **Name :** | |  | |
|  | |  | | | See the source image | | | | | | | | | |  | |
|  | |  | | | **Continous Assessment Test 1(CAT 1) – August 2022** | | | | | | | | | |  | |
| Programme | | | | | | : | **B.Tech. CSE** | Semester | | : | | **Fall Semester 2022-23** | | | |
| Course Code | | | | | | : | **CSE4001** | Class Nbr(s) | | : | | **CH2022231000229**  **CH2022231000223**  **CH2022231000226**  **CH2022231000220**  **CH2022231000217** | | | |
| Course Title | | | | | | : | **Parallel & Distributed Computing** |
| Faculty(s) | | | | | | : | **M.Sivagami, Harini.S, Kumar.R**  **Ayeesha S.K, VenkatRaman.S** | Slot | | : | | **D2** | | | |
| Time | | | | | | : | **90 Minutes** | Max. Marks | | : | | **50** | | | |
|  | | | **Answer all the Questions** | | | | | | | | | | | | |
| Q. No. | | Sub- division | | | Question Text | | | | | | | | | | Marks | |
| 1. | |  | | | In a shared memory multiprocessor system with a distinct cache memory for each processor has been chosen for parallel processing implementation. The local copy of the data is kept in shared memory while the other copy is kept in each core's own cache (d-cache). As a result, every time a local copy of the matching data is changed in the cache memory of any processor in the system, it is necessary to update the data in shared memory. You have been tasked as a computer architect with a suitable system to preserve consistent data throughout processor cache memory with a neat sketch.  Solution:  Given architecture is multicore with private cache (2 marks)  By using cahce coherence protocol, the above problem can be solved (2 marks)  MSI/Invalidate protocol explanation (6 marks) | | | | | | | | | | 10 | |
| 2. | | a)  b) | | | Identify the suitable memory architecture for a tightly coupled application and justify your answer with a proper explanation.  An application, **SumApp**, performs the sum of marks of CSE4001 course for a class of 1024 students. Compare the performance of **SumApp** in a sequential processor and a vector processor with a vector size of 64. Highlight the speedup you will get in vector processing. Justify your answer with a neat sketch.  Solution:  UMA could be better. Explanation (2 + 2 marks)  b)  SUMAPP 1024 students takes 1024 clocks in sequential processor with explanation (2 marks)  In Vector processor of size 64, it will be 1024/64 with explanation (3 marks)  Speedup🡪 16 | | | | | | | | | | 4  6 | |
| 3. | |  | | | A NEXTGEN company developing processors where you are working has just bought a new server based on an octa-core Intel Core i7 processor, and you have been asked to optimize your software applications for this processor. Assuming an application consists of 50% of non-parallelizable code  I. Compute the speed up using Amdahl’s Law [2 Marks]  II. Compute the speedup using Gustafson’s Law [2 Marks]  III. Justify whether the speedups achieved with Amdahl’s and Gustafson’s Law are equal. [1 Mark]  Solution:   1. 1/(0.5)+((0.5)/8) = 1.77 ( 2marks) 2. 8-0.5(8-1) = 8-3.5 =4.5 (2 marks) 3. No they are not ( explanation) (1 mark) | | | | | | | | | | 5 | |
| 4. | |  | | | Assume 3 stage pipeline: Instruct Fetch, Decode, and Execute. Each stage takes 1 clock cycle. How many clock cycle improvements can be achieved in an in-order pipeline processor compared to a non-pipelined processor core for the below pseudo-code? Justify your answer by showing the step-by-step execution in both the processors.  **Instruction Format:**  *Operation Destination, Source 1, Source 2*  **Pseudo-Code**  *Add R0, R1, R2*  *Sub R3, R4, R0*  *LogicalOR R13, R5, R6*  *Mul R7, R8, R9*  *Div R10, R11, R12*  *Solution:*  Non-pipelined : 15 clocks ( 5 marks – with explanation)  Pipelined : 8 clocks (5 marks – with explanation) | | | | | | | | | | 10 | |
| 5. | | a)  b) | | | The XYZ company maintains its 500 employee ids in an array. In order to increase security, HR would like to encrypt the employee ids and store them in another array as below.  Assume that the company maintains 6 digits employee id – Last two digits should be swapped with the first two digits (6th digit with the first digit and 5th digit with the second digit), 3rd and 4th digits should be left as such. You are requested to help the manager to write the open MP code for this scenario to store the all-encrypted ids to the resultant array and display the count of the employee ids which does not have the significance of encryption.  Example:  Scenario 1: 123456 is an employee id. Then it should be encrypted as 653421  Scenario 2: some employee ids may be like this. Ex: 123421. After encryption 123421. Hence encryption on this employee id does not have significance.  Write the output of the following code snippets with proper justification. (1 mark each)   |  |  | | --- | --- | | **Code Snippet 1** | **Code Snippet 2** | | *int x= 34;*  *#pragma omp critical*  *x=x+34;*  *printf(“%d”, x);*  *Solution:*  *68 ( printed once)* | *int x= 34;*  *#pragma omp parallel*  *{#pragma omp single*  *x=x+34;}*  *printf(“%d”, x);*  *68 printed once* | | **Code Snippet 3** | **Code Snippet 4** | | *int x= 34;*  *omp\_set\_num\_threads(4);*  *#pragma omp parallel*  *{x=x+34;}*  *printf(“%d”, x);*  *Incoherent answer since x is shared* | *int x= 34;*  *omp\_set\_num\_threads(4);*  *#pragma omp parallel private(x)*  *{x=x+34;*  *printf(“%d”, x);}*  *x=68 printed four times (if firstprivate given, else junk value)* | | **Code Snippet 5**  *int x= 34;*  *#pragma omp parallel*  *{#pragma omp critical*  *x=x+34;}*  printf(“%d”, x);  x=68, x= 102; x= 136; x=170… depending on number of threads | | | | | | | | | | | | 10  5 | |
| **Total** | | | | | | | | | | | | | | **50** | |
|  | |  | | | ⇔⇔⇔ | | | | | | | | | | |