|  |
| --- |
| Serial No: |
| **1st Mid Term Exam** |
| **Total Time:1 Hour** |
| **Total Marks: 40** |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  Signature of Invigilator |

|  |
| --- |
| **CS326 Parallel and Distributed Computing** |
| Monday, March 14, 2022 |
| **Course Instructor** |
| Ch. Usman Ghous |

|  |
| --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ . .  Roll No Section Signature |

## DO NOT OPEN THE QUESTION BOOK OR START UNTIL INSTRUCTED.

**Instructions:**

1. Verify at the start of the exam that you have a total of five (5) questions printed on five (5) pages including this title page.
2. Attempt all questions on the question-book and in the given order.
3. The exam is closed books, closed notes. Please see that the area in your threshold is free of any material classified as ‘useful in the paper’ or else there may a charge of cheating.
4. Read the questions carefully for clarity of context and understanding of meaning and make assumptions wherever required, for neither the invigilator will address your queries, nor the teacher/examiner will come to the examination hall for any assistance.
5. Write Daira 2022 on the top left corner of title page.
6. Fit in all your answers in the provided space. You may use extra space on the last page if required. If you do so, clearly mark question/part number on that page to avoid confusion.
7. Use only your own stationery and calculator. If you do not have your own calculator, use manual calculations.
8. Use only permanent ink-pens. Only the questions attempted with permanent ink-pens will be considered. Any part of paper done in lead pencil cannot be claimed for checking/rechecking.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Q-1 | Q-2 | Q-3 | Q-4 | Q-5 | Total |
| **Total**  **Marks** | 6 | 13 | 5 | 10 | 6 | **40** |
| **Marks Obtained** |  |  |  |  |  |  |

**Vetted By: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Vetter Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **University Answer Sheet Required:** | **No** |  |  |  |  | **Yes** |  |

|  |  |
| --- | --- |
| Question 1 | Marks:3+3 |

State Moore’s Law and its revised version. Explain with the help of an example.

*The complexity for minimum component costs has increased at a rate of roughly a factor of two per year. Certainly, over the short term this rate can be expected to continue, if not to increase. Over the longer term, the rate of increase is a bit more uncertain, although there is no reason to believe it will not remain nearly constant for at least 10 years. That means by 1975, the number of components per integrated circuit for minimum cost will be 65,000.*

*There is no room left to squeeze anything out by being clever. Going forward from here we have to depend on the two size factors - bigger dies and \_ner dimensions.*

|  |  |
| --- | --- |
| Question 2 | Marks:3+5 +5 |

Define Pipelining. State it benefits and limitations. Also include an example that demonstrates how pipelining is done in modern architecture.

Pipelining overlaps various stages of instruction execution to achieve performance. At a high level of abstraction, an instruction can be executed while the next one is being decoded and the next one is being fetched.

Limitations:

The speed of a pipeline is eventually limited by the slowest stage.

For this reason, conventional processors rely on very deep pipelines (20 stage pipelines in state-of-the-art Pentium processors).

However, in typical program traces, every 5-6th instruction is a conditional jump! This requires very accurate branch prediction.

The penalty of a misprediction grows with the depth of the pipeline, since a larger number of instructions will have to be flushed.

|  |  |
| --- | --- |
| Question 3 | Marks:5 |

Define vertical and horizontal waste.

Not all functional units can be kept busy at all times.

If during a cycle, no functional units are utilized, this is referred to as vertical waste.

If during a cycle, only some of the functional units are utilized, this is referred to as horizontal waste.

|  |  |
| --- | --- |
| Question 4 | Marks:3+7 |

Consider the following code fragment for matrix sum:

for (i = 0; i < 1000; i++)

column\_sum[i] = 0.0;

for (j = 0; j < 1000; j++)

column\_sum[i] += b[j][i];

Make changes to the code such that performance of sum operation is improved. Describe how the performance is improved after the changes made (the answer must be elaborative. Add diagrams if necessary).

for (i = 0; i < 1000; i++)

column\_sum[i] = 0.0;

for (j = 0; j < 1000; j++)

for (i = 0; i < 1000; i++)

column\_sum[i] += b[j][i];

In this case, the matrix is traversed in a row-order and performance can be expected to be significantly better.

|  |  |
| --- | --- |
| Question 5 | Marks:6 |

Fill the missing values in the following table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Network | Diameter | Bisection Width | Arc Connectivity | Cost  (No. of links) |
| Completely-connected | 1 | P2/4 | p-1 | p(p-1)/2 |
| Star | 2 | 1 | 1 | p-1 |
| Complete binary tree | 2log((p+1)/2) | 1 | 1 | p-1 |
| Linear array | p-1 | 1 | 1 | p-1 |