1) (17 points) Consider a workload that achieves 10 MPKI (MPKI = misses per 1000 instructions) in the last level cache. Assume that main memory can serve four requests concurrently (because of 4-way banking) and that memory access latency is 300 cycles.
   a. (8 points) What is the maximum achievable IPC (instructions per cycle) for the above system?
   b. (9 points) Because of non-uniform popularity of addresses, it is found that the access distribution among the four memory banks is: 40%, 15%, 10%, and 35%. What is the maximum possible IPC for the above access distribution?

2) (33 points)
   a. (9 points) A given workload interacts with a memory hierarchy in the following way: When the associativity is N, the miss rate (in percent) is \( \frac{1+9}{N} \). In general, the hit time increases with increasing associativity. For our memory hierarchy, the hit time with associativity N is given by \( 1 + \frac{N}{25} \) nanoseconds. Assume that the average miss penalty is 144 ns. For the above workload and system, what is the associativity at which average memory access time is minimized?

   b. Consider a workload whose working set fits entirely in physical memory. However, the pages of the working-set in physical memory are scattered uniformly in the virtual address space. Assume that the virtual address space is significantly larger than the physical address space.
      i. (12 points) For each of the following page tables give an expression for the number of lookups per TLB miss? (a) Traditional flat page table, (b) Inverted page table, (c) Multi-level page table. In your answer, you may define and use any parameters for each of the organizations.
      ii. (12 points) Assuming the above workload, give an expression for the page table size for each of the above page table organizations. In your answer, you may define and use any parameters for each of the organizations.

3) (20 points) Consider a sequentially consistent system with an invalidate-based coherence protocol. Assume that before the following code segment runs X=Y=0.

<table>
<thead>
<tr>
<th>Thread 1</th>
<th>Thread 2</th>
<th>Thread 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>( X \leftarrow 10; )</td>
<td>while ( X != 10 ) { ( // empty loop ) } ; ( Y \leftarrow 20; ) ; Print Y</td>
<td>while ( Y != 20 ) { ( // empty loop ) } ; Print X</td>
</tr>
</tbody>
</table>

Under normal operation, invalidate-based protocols ensure that a write waits for all existing copies to be invalidated before the newly written value is made visible to other processors.

To avoid the wait time associated with all the invalidations and their acknowledgments, an engineer proposes an optimization that serves requests as long as the requester has seen the invalidation (without waiting for ALL invalidations to be acknowledged).
(1) (5 points) Does the optimization maintain correctness?
(2) (15 points) If so, show that all the possible outputs are SC-compliant. If not, show a violation of correctness for the above example. Your answer should (a) provide an output that is impossible under SC but which is possible under the above scheme, and (b) explain the sequence of actions that leads to the incorrect output.

4) (30 points) There is a new transistor technology which offers two interesting features: (1) it is suitable for both logic and main memory, and (2) it is dense enough to hold all the cores and main memory in a single chip. Unfortunately, it suffers from low on-chip communication bandwidth, especially at larger on-chip distances. In general, the technology offers higher bandwidth for shorter distances.

The goal is to design an architecture for a workload that (a) is data intensive (i.e., large input data), (b) is sequential, and (c) has a small code footprint. The workload has no temporal locality but exhibits strong spatial locality in the input data. Further, the “live state” of the workload (which includes register contents and intermediate data) is fairly compact. (In other words, even though the workload accesses a large memory footprint, its intermediate data remains bounded and compact for the entire duration of the run.) Assume that the entire data footprint of the workload fits on the chip (but occupies a large fraction of the chip’s area).

(1) (5 points) What is the basic core and memory architecture you would use for such a workload to maximize performance?
(2) (7 points) Given the sequential nature of the workload and the limits of the technology, what basic strategy would you use to achieve high performance? Specify only the “what” here; the “how” comes next.
(3) (18 points) Explain how the basic strategy can be implemented. Specifically, consider the key actions that are part of your strategy:
   a. (5 points) Qualitatively, when would you initiate those actions?
   b. (8 points) How would you detect that the conditions are appropriate for such actions? Be specific. Mention mechanisms that allow you to monitor conditions under which you can initiate the key actions of your strategy.
   c. (5 points) Describe any conditions the workload must satisfy in addition to those specified in the question above for your strategy to work well.