National University of Computer and Emerging Sciences

Islamabad Campus

## High Performance Computing with GPUs (CS4110)

**Course Instructor(s):**

Dr. Imran Ashraf

**Section(s): A, B, C**

# Final Exam

**Total Time (Min):** 75

**Total Marks:** 60

**Total Questions:** 4

**Date:** Dec 23, 2025

_____ _____ _____

**Roll No** **Course Section** **Student Signature**

**Do not write below this line.**

# Part 01

1. **Part 01 should be attempted first.**
2. **RETURN this part to the invigilator to get Part 02.**
3. **Choose the correct answer.**
4. **Solve each question in this part on the CORRESPONDIG bubble sheet attached at the end of the question paper.**
5. **As a backup, you can encircle the correct choice on the question paper.**
6. **Cutting, overwriting will result in ZERO marks.**
7. **Completely shade the circle in the bubble sheet with ink pen.**

**CLO 1: Describe** the terms related to HPC systems, GPU architecture and GPU programming models. **[Understand, C2]**

**Q1:** **[15 marks]**

**1. In CUDA terminology, which component is responsible for issuing and scheduling warps on a Streaming Multiprocessor (SM)?**

A. CUDA Core
B. Warp Scheduler
C. Thread Block Scheduler
D. Global Scheduler

**2. Which CUDA memory space has the lowest latency but is also private to each thread?**

A. Shared memory
B. Constant memory
C. Registers
D. Local memory

**3. A warp divergence occurs when:**

A. Threads from different blocks access different memory spaces
B. Threads within a warp follow different execution paths

C. Multiple warps access the same memory address
D. A block contains more than 1024 threads

**4. Which CUDA memory space is cached and optimized for broadcast when all threads read the same address?**

A. Global memory
B. Shared memory
C. Constant memory
D. Local memory

**5. In CUDA, occupancy of an SM is defined as:**

A. Percentage of GPU memory in use
B. Ratio of active warps to maximum possible warps per SM
C. Number of active threads per kernel launch
D. Ratio of shared memory to registers used

**6. Which factor does NOT directly limit SM occupancy?**

A. Number of registers per thread
B. Shared memory per block
C. Instruction cache size
D. Maximum threads per SM

**7. CUDA local memory is:**

A. Physically stored in registers
B. Stored in shared memory
C. Mapped to global memory
D. Stored in constant cache

**8. In CUDA's execution hierarchy, which mapping is correct?**

A. Grid → Warp → Block → Thread
B. Thread → Warp → Block → Grid
C. Block → Thread → Warp → Grid
D. Warp → Thread → Grid → Block

**9. Which CUDA API call ensures that all threads within a block have reached the same execution point?**

A. `cudaDeviceSynchronize()`
B. `__syncwarp()`
C. `__syncthreads()`
D. `cudaThreadSynchronize()`

**10. Coalesced global memory access occurs when:**

A. Threads in a block access random addresses
B. Threads in a warp access consecutive memory locations
C. Multiple blocks access the same address
D. All threads access constant memory

**11. In CUDA, SIMT differs from classical SIMD because:**

A. Each thread executes completely different instructions
B. Threads are grouped dynamically at runtime

C. Each thread has its own program counter
D. Instructions are not vectorized

**12. Which CUDA programming model feature enables overlapping data transfer and kernel execution?**

A. Unified memory
B. CUDA streams
C. Constant memory
D. Texture memory

**13. The primary role of the CUDA runtime API (vs driver API) is to:**

A. Provide low-level GPU control
B. Enable kernel scheduling at hardware level
C. Offer a higher-level, easier programming interface
D. Replace the CUDA compiler

**14. Which statement about shared memory is TRUE?**

A. It is visible to all threads on the GPU
B. It is slower than global memory
C. It is shared only among threads in the same block
D. It is read-only

**15. In CUDA, increasing block size beyond a certain point may reduce performance primarily due to:**

A. Increased PCIe latency
B. Higher global memory bandwidth usage
C. Reduced occupancy due to resource limits
D. More kernel launches

1. B
2. C
3. B
4. C
5. B
6. C
7. C
8. B
9. C
10. B
11. C
12. B
13. C
14. C
15. C

**CLO 2: Identify** application hotspots based on the application profile using realistic workload. **[Analyze, C4]**

**Q2:** **[15 marks]**

**1. In `gprof`, a hotspot is identified primarily by:**

A. Functions with the highest number of calls
B. Functions consuming the largest percentage of total execution time
C. Functions with deepest call stack
D. Functions with largest binary size

**2. Which `gprof` output is most useful for identifying where optimization will give maximum benefit?**

A. Flat profile
B. Call graph profile
C. Compilation warnings
D. Symbol table

**3. A function that is called very frequently but consumes little total execution time should:**

A. Always be optimized first
B. Be ignored completely
C. Be optimized only if it limits scalability
D. Be rewritten in assembly

**4. According to Amdahl's Law, the maximum speedup of a program is limited by:**

A. Number of available processors
B. Memory bandwidth
C. Fraction of serial execution
D. Instruction-level parallelism

**5. Arithmetic Intensity (AI) of an application is BEST defined as:**

A. Number of floating-point operations per second
B. Ratio of floating-point operations to memory bytes transferred
C. Ratio of compute time to memory latency
D. Number of instructions executed per cycle

**6. Which statement BEST explains why profiling must use a realistic workload?**

A. Synthetic workloads execute faster
B. Compiler optimizations behave differently
C. Hotspots may shift depending on input size and data patterns
D. Profilers do not work on small inputs

**7. The theoretical peak FLOPS of a processor depends on:**

A. Cache size and latency
B. Clock frequency, number of cores, and operations per cycle

C. Memory bandwidth only
D. Instruction count

**8. If an application achieves performance close to peak FLOPS but low memory bandwidth usage, it is likely:**

A. Memory-bound
B. I/O-bound
C. Compute-bound
D. Latency-bound

**9. Memory bandwidth (GB/s) is BEST defined as:**

A. Amount of memory per core
B. Time taken to access memory
C. Data transferred per second between memory and processor
D. Number of memory instructions executed

**10. An application shows low FLOPS utilization but is close to peak memory bandwidth. This indicates:**

A. Compute-bound behavior
B. Instruction-level parallelism
C. Memory-bound behavior
D. Poor compiler optimization

**11. Which metric is MOST useful to decide whether optimizing arithmetic intensity will improve performance?**

A. Cache miss rate
B. FLOPS per byte (operational intensity)
C. Number of threads
D. Function call count

**12. According to Amdahl's Law, optimizing a function that takes 5% of execution time can at best:**

A. Double overall performance
B. Give unlimited speedup
C. Provide marginal overall speedup
D. Remove the serial bottleneck

**13. In `gprof`, if a function has high self time but low children time, it implies:**

A. Most time is spent in its callees
B. It dominates execution by itself
C. It is I/O-bound
D. It cannot be optimized

**14. Which of the following best combines profiling and hardware limits to identify performance bottlenecks?**

A. Counting number of instructions
B. Comparing measured performance to theoretical peaks
C. Increasing compiler optimization level
D. Reducing function calls

**15. If profiling shows a kernel achieves only 10% of peak FLOPS and 15% of peak bandwidth, the most likely cause is:**

A. Hardware failure
B. Poor parallelism or low occupancy
C. Excessive memory bandwidth
D. Overclocking

1. B
2. A
3. C
4. C
5. B
6. C
7. B
8. C
9. C
10. C
11. B
12. C
13. B
14. B
15. B

**CLO 3: Develop** data-parallel solutions using appropriate programming model.**[Create, C6]**

**Q3:** **[15 marks]**

**1. Which CUDA kernel launch configuration is MOST appropriate for processing an array of size `N`?**

A. `<<<1, N>>>`
B. `<<<N, 1>>>`
C. `<<<ceil(N/256), 256>>>`
D. `<<<256, ceil(N/256)>>>`

**2. What happens if the number of threads launched exceeds the data size?**

A. Program crashes
B. GPU ignores extra threads automatically
C. Threads must explicitly guard against out-of-bounds access
D. CUDA runtime resizes the grid

**3. A CUDA kernel uses `__syncthreads()` inside an `if` statement executed by only some threads. This will most likely:**

A. Improve performance
B. Cause deadlock or undefined behavior
C. Synchronize only active threads
D. Be optimized away

**4. Which CUDA keyword is required to define a GPU kernel?**

A. `__device__`
B. `__host__`
C. `__global__`
D. `__shared__`

**5. In CUDA, which situation requires atomic operations?**

A. Threads read from same memory location
B. Threads write to independent memory locations
C. Multiple threads update the same memory location
D. Threads use shared memory

**6. A kernel launch `<<<grid, block>>>` fails with an "invalid configuration argument" error most likely because:**

A. Kernel uses too many registers
B. Block size exceeds hardware limits
C. Global memory is insufficient
D. Host and device code mismatch

**7. Which CUDA memory access pattern yields the best performance?**

A. Each thread accesses a random address
B. Threads in a warp access consecutive addresses
C. Threads access memory with large stride
D. Threads repeatedly access the same global address

**8. In a reduction kernel, why is it common to reduce data within a block first?**

A. To avoid global memory
B. To exploit fast shared memory and synchronization
C. To reduce register usage
D. To increase kernel launch overhead

**9. Which CUDA construct allows independent kernel executions to overlap?**

A. CUDA events
B. CUDA streams
C. Cooperative groups
D. Warp primitives

**10. When developing a CUDA data-parallel solution, which step should be performed first?**

A. Tune block size
B. Optimize memory coalescing
C. Identify independent data elements
D. Reduce register usage

**11. Which CUDA scenario most clearly violates data-parallel principles?**

A. Each thread computes one array element
B. Threads perform identical operations on different data
C. Threads frequently wait on each other
D. Threads use shared memory

**12. What is the purpose of `__restrict__` in CUDA kernel parameters?**

A. Prevent pointer aliasing for better optimization
B. Restrict kernel launch dimensions
C. Limit shared memory usage
D. Enforce thread synchronization

**13. If a kernel shows low performance despite high occupancy, the MOST likely cause is:**

A. Excessive global memory latency
B. Too many threads per block
C. Too many blocks in grid
D. Low instruction throughput

**14. Why should kernel code minimize branching?**

A. Branching increases register count
B. Branching causes warp divergence
C. Branching increases memory bandwidth
D. Branching prevents kernel launch

**15. Which of the following operations can execute concurrently in different CUDA streams (on capable hardware)?**

A. Two kernels and one memory copy, all in the same stream
B. Two memory copies from host to device in the default stream
C. A kernel in one stream and an asynchronous memory copy in another stream
D. Two kernels launched from the same stream

1. C
2. C
3. B
4. C
5. C
6. B
7. B
8. B
9. B
10. C
11. C
12. A
13. A
14. B
15. C

**CLO 4: Analyze** performance of an application running on an HPC system to improve compute and memory performance. **[Evaluate, C5]**

**Q4:** **[12 marks]**

**1. The primary effect of shared memory bank conflicts is:**

A. Incorrect program output
B. Increased global memory traffic
C. Serialized shared memory accesses
D. Kernel launch failure

**2. Non-coalesced global memory access results in:**

A. Reduced register pressure
B. Increased memory transactions per warp
C. Improved cache reuse
D. Higher arithmetic intensity

**3. Which access pattern is most likely to cause coalescing issues?**

A. `A[idx]`
B. `A[threadIdx.x]`
C. `A[threadIdx.x * stride]` where `stride > warp size`
D. `A[blockIdx.x * blockDim.x + threadIdx.x]`

**4. Adding padding to data structures primarily helps by:**

A. Increasing memory usage
B. Improving arithmetic intensity

C. Avoiding alignment and bank conflict issues
D. Reducing instruction count

**5. Which launch configuration choice MOST directly affects occupancy?**

A. Grid size only
B. Block size only
C. Kernel name
D. Number of streams

**6. Choosing too SMALL a block size often leads to:**

A. Kernel launch failure
B. Excessive register spilling
C. Poor latency hiding and low occupancy
D. Memory bank conflicts

**7. Which situation PREVENTS effective overlap when using CUDA streams?**

A. Using pinned host memory
B. Using `cudaMemcpyAsync`
C. Using the default stream
D. Launching multiple kernels

**8. In Python Numba (CUDA), which practice MOST NEGATIVELY affects performance?**

A. Using `@cuda.jit` decorator
B. Launching kernels inside Python loops repeatedly
C. Using shared memory
D. Using explicit grid-stride loops

**9. Which of the following techniques is most effective at eliminating bank conflicts when accessing a 2D shared memory array with a power-of-two stride?**

A. Increasing the number of threads per block.
B. Using __syncthreads() after every load.
C. Padding the leading dimension of the array (e.g., [row][column + 1]).
D. Moving the data to Constant Memory.

**10. Global memory coalescing is achieved when threads in a warp access:**

A. Random locations across the entire GPU heap.
B. Addresses that fall into a single aligned memory transaction (e.g., 32, 64, or 128 bytes).
C. Different shared memory banks simultaneously.
D. Only the registers of neighboring threads.

**11. You observe that a kernel's performance is limited by "Global Load Throughput." The code uses array[threadIdx.x * stride]. Which value of stride will likely result in the worst coalescing?**

A. 1
B. 2
C. 16
D. 32

**12. When optimizing for memory performance, why is "Structure of Arrays" (SoA) generally preferred over "Array of Structures" (AoS) in CUDA?**

A. SoA uses less total memory on the device.
B. SoA facilitates coalesced memory access patterns for individual fields.
C. AoS allows for faster atomic operations.
D. The GPU L1 cache cannot store data from structures.

**13. A developer increases the number of threads per block from 256 to 1024. If the kernel uses a high amount of registers per thread, what is the most likely negative impact?**

A. Decreased occupancy due to register pressure.
B. Increased shared memory bank conflicts.
C. Improved warp scheduling efficiency.
D. Reduced global memory bandwidth.

**14. Which of the following allows for the overlapping of data transfers (Host-to-Device) with kernel execution?**

A. Using the default stream (Stream 0).
B. Using multiple non-blocking CUDA streams and asynchronous copy functions.
C. Increasing the GPU clock speed.
D. Using __syncthreads() within the kernel.

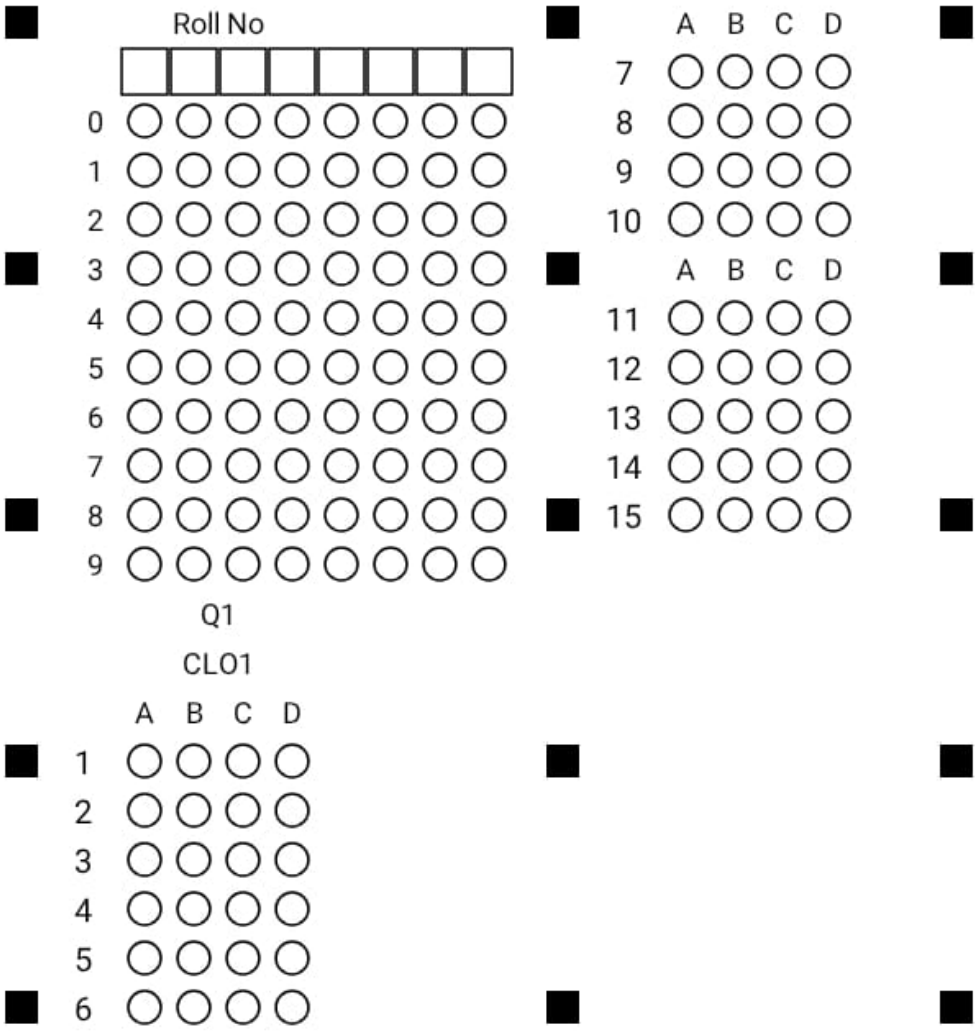
**15. If a kernel is "Compute Bound," which of the following actions is most likely to improve performance?**

A. Improving memory coalescing.
B. Increasing the complexity of the math (e.g., using sin() instead of __sinf()).
C. Reducing instruction overhead or using faster intrinsic functions.
D. Padding shared memory arrays.

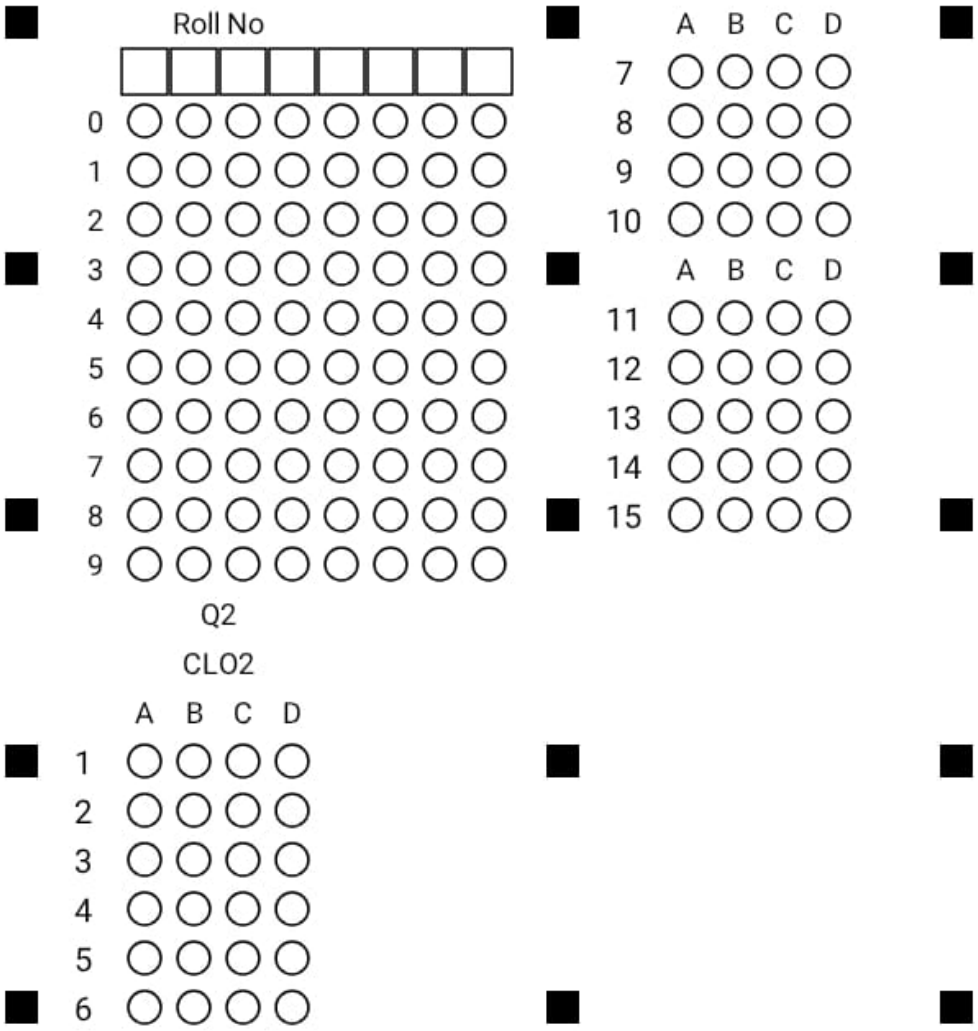
1. C
2. B
3. C
4. C
5. B
6. C

7. C
8. B
9. C
10. B
11. D
12. B
13. A
14. B
15. C

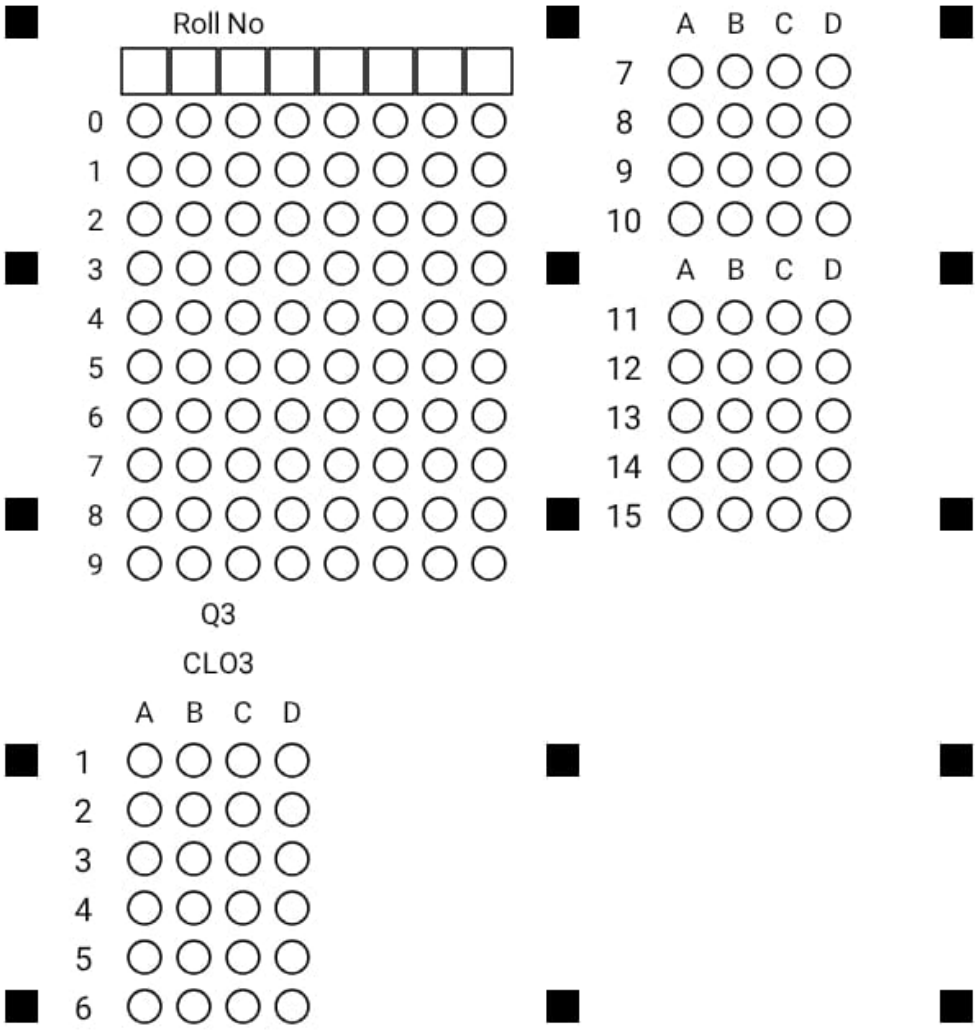
| NAME : |
| --- |
| EXAM : |
| DATE : |



| NAME : |
| --- |
| EXAM : |
| DATE : |



| NAME : |
| --- |
| EXAM : |
| DATE : |



| NAME : |
| --- |
| EXAM : |
| DATE : |