CA Assignment 3

1 Answer:

Branch instructions in GPU code can be slow due to the SIMD execution model, which causes divergence within threads, leading to inefficient resource utilization and serialization of execution. Additionally, complex instruction scheduling, memory access patterns, and hardware constraints further contribute to performance degradation. To mitigate this, programmers often minimize branching, optimize memory access, and leverage data parallelism.

2 Answer:
In CPUs, the Single-Instruction Multiple-Data (SIMD) model executes a single instruction simultaneously on multiple data elements packed into vector registers. This model enhances parallelism for tasks like multimedia processing and scientific computing by processing multiple data elements in parallel with a single instruction.

In GPUs, the Single-Instruction Multiple-Thread (SIMT) model organizes threads into groups called warps, where each thread operates on its own data but executes the same instruction simultaneously. This massively parallel architecture allows thousands of threads to execute concurrently, making GPUs highly effective for tasks such as graphics rendering, scientific simulations, and machine learning.

4 Answer:

(a) **Fine-grained Multithreading**:

- **Introduction**: Instructions from multiple threads are interleaved at a fine level, typically at the cycle or sub-cycle level.
- **Execution**: Threads rapidly switch within the processor's pipeline to hide latency, aiming to keep the pipeline busy.

(b) **Coarse-grained Multithreading**:

- **Introduction**: Entire threads are switched when a thread encounters a stall.
- **Execution**: Threads are scheduled onto processor cores, and execution switches between threads at predefined points or events.

(c) **Simultaneous Multithreading (SMT)**:

- **Introduction**: Instructions from multiple threads are introduced simultaneously into the processor.
- **Execution**: Threads share resources and execute concurrently on the same core, maximizing throughput and resource utilization.

3 Answer:

Cycle	**ARITHMETIC LOGIC UNIT 1**	**ARITHMETIC-LOGIC UNIT 2**	**MEMORY1**	**MEMORY2**	**BRANCH**
1	T2: SUB.D F8, F0, F2	T3: DADDUI R3, R1, R2	T1: L.D F0, 0(R1)	T1: L.D F2, 8(R1)	
2	T3: SUB.D F6, F8, F10				
3			T3: L.D F4, 0(R3)		
4	T2: ADD.D F6, F4, F8				
5	T1: ADD.D F4, F0, F8	T1: SUB.D F6, F2, F8	T3: S.D F6, 0(R1)		
6	T1: DADDUI R1, R1, 16	T3: DADDUI R1, R1, 24			
7	T2: DADDUI R1, R1, 16	T3: DADDUI R2, R2, -8	T2: S.D F6, 8(R1)		
8					T1: BNE R1, 1600, L1

9			T2: L.D F2, 0(R1)		T3: BNE R1, R2, LOOP
10					
11					
12					
13	T2: SUB.D F4, F6, F2				

Only one instruction can be carried out by each operational unit in a cycle. The functional unit in question is not actively processing an instruction during that specific cycle if a cell in the table is empty.