GPU Architectures and Programming
Assignment- Week 5
TYPE OF QUESTION: MCQ/MSQ

Number of questions: 10 **Total mark: 10 X 1 = 10**

MCQ Question

Question 1:

Consider the following grid and block dimensions for a kernel.

gridDim =< 4, 2 > and blockDim =< 4, 4 >

For a hypothetical GPU architecture, where number of SMs is 4 and number of SPs per SM is 32, how many thread blocks of the kernel are mapped to a single SM?

A. 1
B. 2
C. 4
D. 8

Answer: B

Solution: The kernel is launched with a total of 16 threads per block, but the target GPU architecture has support for 32 SPs. A total of 32/16 = 2 thread blocks can thus be scheduled to each SM.

Question 2:

Consider a GPU architecture with the following constraints.

The total number of threads in a thread block is 512. The maximum number of threads in the x-dimension is 256, in the y-dimension is 256 and in the z dimension is 8. Considering a three dimensional CUDA kernel, which of the following thread block configurations is feasible ?

A. blockDim.x = 256, blockDim.y = 256 blockDim.z = 8
B. blockDim.x = 32, blockDim.y = 32 blockDim.z = 8
C. blockDim.x = 32, blockDim.y = 16 blockDim.z = 8

D. blockDim.x = 32, blockDim.y = 16 blockDim.z = 1

Answer: D

Solution:

32*16*1 = 512, which is equal to the maximum limit

Question 3:

What policy does TBS use to place a thread block on an SM:

A. Round Robin
B. Most-Room
C. Least Recently Fetched
D. Fair

Answer: B

Question 4:

Which scheduling policy ensures equal opportunity for all warps in terms of the number of instructions fetched?

A. Round Robin
B. Least Recently Fetched
C. Fair
D. Criticality Aware Warp Scheduling

Answer: C

Question 5:

When calling `cudaGetDeviceCount()`, what type of variable should you pass to the function:

A. A pointer to an integer to store the number of devices
B. A pointer to a float to store the device count
C. A string buffer to store the device names
D. A boolean variable to check the presence of devices

Answer: A

Question 6:

Consider a GPU architecture where the warp size is 16. What are the total number of warps launched during the lifetime of a kernel with the following kernel launch configuration parameters - <<< (1,1,1), (32,32,1)>>>

A. 16
B. 32
C. 64
D. 128

Correct Answer: C

Solution:

Total number of threads launched is 1*1*1*32*32*1=1024. Total number of warps launched is 1024/16=64

Question 7:

Analyze the following code snippet for divergence:

```
int idx = threadIdx.x;
if (idx % 2 == 0) {
    arr[idx] = idx * 2;
} else {
    arr[idx] = idx * 3;
}
```

Which statement best explains the behavior of this code regarding thread divergence?

A. Divergence occurs because threads within a wrap follow different paths based on their index
B. No divergence occurs because idx % 2 is a uniform conditional
C. Wrap size ensures even and odd threads are seperated
D. Wrap execution is independent of branching

Answer: A

Solution:

Divergence occurs because threads within a warp follow different paths

based on their index

Question 8:

Consider a hypothetical GPU architecture where the warp size is 8 and a kernel program which is launched with a configuration where the total number of threads in a thread block is 32. The total number of warps launched per thread block is thus 4. Consider the following conditional statements in the kernel.

i. if(threadIdx.x <16)
ii. if(threadIdx.x %2)
iii. if(threadIdx.x %32)
iv. if(threadIdx.x < 8)

Which of the following options is correct?

A. All conditional branches (i)-(iv) are divergent for all warps
B. Conditional branch (iv) is not divergent for warp 0
C. Conditional branch (ii) is divergent only for warps 0 and 4
D. Conditional branch (i) is divergent only for warps 0 and 1

Answer: B

Solution:

thread ids in warp 0: 0-7
thread ids in warp 1: 8-15
thread ids in warp 2: 16-23
thread ids in warp 3: 24-31

One can observe that all threads in warp 0 can satisfy threadIdx.x < 8.

Question 9:

Which of the following statements is false?

A. Multiple CUDA thread blocks can execute in a single SM of an NVIDIA GPU.
B. The computation of one CUDA thread block can be distributed across multiple SMs.

C. Threads across multiple CUDA thread blocks cannot be synchronized using __syncthreads().
D. Threads in a single CUDA thread block mapped to a single SM can communicate using shared memory .

Answer: B

Question 10:

Consider a kernel processing a 1D array of 8192 elements where each thread is assigned to perform an operation on a single element of the array. The kernel is launched with the following grid and block configurations: <16,a,2> blocks of <b,2,4>

A. a=16,b=2
B. a= 2 b= 16
C. a=8 b=4
D. All of the above

Correct Answer: D

The number of threads for all configurations is equal to 8192.

**********END**********