

## Quiz Questions for Module 10

1. For the work inefficient scan kernel based on reduction trees, assume that we have 1024 elements, which of the following gives the closest approximation of the number of add operations performed?

(A)  $(1024-1)*2$   
(B)  $(512-1)*2$   
(C)  $1024*1024$   
(D)  $1024*10$

Answer: (D)

Explanation: The number of add operations performed by the work inefficient scan kernel is approximately  $N*\log(N)$ , where N is the number of elements.

2. For the work efficient exclusive scan kernel based on reduction trees and inverse reduction trees, assume that we have 1024 elements, which of the following gives the closest approximation on the total number of add operations performed in both the reduction tree phase and the reverse reduction tree phase?

(A)  $(1024-1)*2$   
(B)  $(512-1)*2$   
(C)  $1024*1024$   
(D)  $1024*10$

Answer: (A)

Explanation: The reduction tree performs approximately N-1 add operations performed in the reduction tree phase and the inverse reduction performs approximately N-1 add operations (technically  $N-1-\log(N) - 1$ , but for large N this is close to N-1)

3. For the work efficient scan kernel based on reduction trees and inverse reduction trees, assume that we have 2048 elements (each block has BLOCK\_SIZE=1024 threads) in each section and warp size is 32, how many warps in each block will have control divergence during the reduction tree phase iteration where stride is 16? For your convenience, the relevant code fragment from the kernel is given below:

```
for (unsigned int stride = 1; stride <= BLOCK_SIZE; stride = stride*2) {  
    int index = (threadIdx.x+1)*stride*2 - 1;  
    if(index < 2*BLOCK_SIZE) {XY[index] += XY[index-stride];}  
    __syncthreads();  
}
```

- (A) 0  
(B) 1  
(C) 16  
(D) 32

Answer: (A)

All active threads are consecutive starting with index 0. There are 64 active threads in the iteration in each block. The warps are either all active or all inactive. None will have control divergence.

4. For the work inefficient scan kernel based on reduction trees, assume that we have 1024 elements in each section and warp size is 32, how many warps in each block will have control divergence during the iteration where stride is 16? For your convenience, the relevant code fragment from the kernel is given below:

```
for (unsigned int stride = 1; stride <= threadIdx.x; stride *= 2) {  
    __syncthreads();  
    float in1 = XY[threadIdx.x-stride];  
    __syncthreads();  
    XY[threadIdx.x] += in1;  
}
```

- (A) 0  
(B) 1  
(C) 16  
(D) 32

Answer: (B)

All active threads are consecutive at the end of the block. When stride is 16, there are 16 inactive threads at the beginning of the block. So, Warp 0 has control divergence. No other warp in the block will have control divergence.

5. In the previous question, how many warps in each block will have control divergence during the iteration where stride is 64?

- (A) 0
- (B) 1
- (C) 16
- (D) 32

Answer: (A)

All 64 inactive threads are at the front of the block. Therefore, all threads in the first two warps are inactive. All threads in the remaining warps are active. There is no control divergence.