

9 GPUs and SIMD [45 points]

We define the *SIMD utilization* of a program that runs on a GPU as the fraction of SIMD lanes that are kept busy with *active threads* during the run of the program. As we saw in lecture and practice exercises, the SIMD utilization of a program is computed across the *complete run* of the program.

The following code segment is run on a GPU. Each thread executes **a single iteration** of the shown loop. Assume that the data values of the arrays A and B are already in vector registers, so there are no loads and stores in this program. (Hint: Notice that there are 3 instructions in each iteration.) A warp in the GPU consists of 32 threads, and there are 32 SIMD lanes in the GPU.

```
for (i = 0; i < 1025; i++) {
    if (A[i] < 33) {          // Instruction 1
        B[i] = A[i] << 1;    // Instruction 2
    }
    else {
        B[i] = A[i] >> 1;    // Instruction 3
    }
}
```

Please answer the following six questions.

- (a) [2 points] How many warps does it take to execute this program?

33 warps.

Explanation:

The number of warps is calculated as:

$$\#Warp s = \lceil \frac{\#Total_threads}{\#Warp_size} \rceil,$$

where

$$\#Total_threads = 1025 = 2^{10} + 1 \text{ (i.e., one thread per loop iteration),}$$

and

$$\#Warp_size = 32 = 2^5 \text{ (given).}$$

Thus, the number of warps needed to run this program is:

$$\#Warp s = \lceil \frac{2^{10}+1}{2^5} \rceil = 2^5 + 1 = 33.$$

- (b) [10 points] What is the *maximum* possible SIMD utilization of this program? (Hint: The warp scheduler does *not* issue instructions when *no* threads are active).

$$\frac{1025}{1056}.$$

Explanation:

Even though all active threads in a warp follow the same execution path, the last warp will only have one active thread.

- (c) [5 points] Please describe what needs to be true about array A to reach the maximum possible SIMD utilization asked in part (b). (Please cover all cases in your answer.)

For every 32 consecutive elements of A, every element should be lower than 33 (if), or greater than or equal to 33 (else). (NOTE: The solution is correct if both cases are given.)

- (d) [13 points] What is the *minimum* possible SIMD utilization of this program?

$$\frac{1025}{1568}.$$

Explanation:

Instruction 1 is executed by every active thread ($\frac{1025}{1056}$ utilization). Then, part of the threads in each warp executes Instruction 2 and the other part executes Instruction 3. We consider that Instruction 2 is executed by α threads in each warp (except the last warp), where $0 < \alpha \leq 32$, and Instruction 3 is executed by the remaining $32 - \alpha$ threads. The only active thread in the last warp executes either Instruction 2 or Instruction 3. The other instruction is not issued for this warp.

The minimum SIMD utilization sums to $\frac{1025 + \alpha \times 32 + (32 - \alpha) \times 32 + 1}{1056 + 1024 + 1024 + 32} = \frac{1025}{1568}.$

- (e) [5 points] Please describe what needs to be true about array A to reach the minimum possible SIMD utilization asked in part (d). (Please cover all cases in your answer.)

For every 32 consecutive elements of A, part of the elements should be lower than 33 (if), and the other part should be greater than or equal to 33 (else).

- (f) [10 points] What is the SIMD utilization of this program if $A[i] = i$? Show your work.

$$\frac{1025}{1072}.$$

Explanation:

Instruction 1 is executed by every active thread ($\frac{1025}{1056}$ utilization). Instruction 2 is executed by the first 33 threads, i.e., all threads in the first warp and one thread in the second warp. Instruction 3 is executed by the remaining active threads.

The SIMD utilization sums to $\frac{1025 + 32 + 1 + 31 + 960 + 1}{1056 + 32 + 32 + 32 + 960 + 32} = \frac{2050}{2144} = \frac{1025}{1072}.$