4. GPUs and SIMD [60 points]

We define the *SIMD utilization* of a program running on a GPU as the fraction of SIMD lanes that are kept busy with *active threads*.

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 the vector registers so there are no loads and stores in this program. Hint: Notice that there are 2 instructions in each thread. A warp in this GPU consists of 32 threads, and there are 32 SIMD lanes in the GPU. Assume that each instruction takes the same amount of time to execute.

```
for (i = 0; i < N; i++) {
    if (A[i] % 3 == 0) {     // Instruction 1
       A[i] = A[i] * B[i];   // Instruction 2
    }
}
```

(a) What's the minimum number of bits required to encode the warp ID in order to execute this program? Please leave the answer in terms of N.

$\lceil log_2(\frac{N}{32}) \rceil$

(b) Assume integer array A has a repetitive pattern which consists of 24 ones followed by 8 zeros, and integer array B has a different repetitive pattern which consists of 48 zeros followed by 64 ones. What is the SIMD utilization of this program?

((24+8*2)/(32*2))*100% = 40/64*100 = 62.5%

(c) Is it possible for this program to yield a SIMD utilization of 100% (circle one)?

YES NO

If YES, what should be true about arrays A for the SIMD utilization to be 100%?

Yes. If, for every 32 elements of A, all of them are divisible by 3, or if all are not divisible by 3.

If NO, explain why not.

What should be true about array B?

B can be any array of integers.

(d) Is it possible for this program to yield a SIMD utilization of 56.25% (circle one)? *Hint: 56.25% = 36/64.*

YES NO

If YES, what should be true about arrays A for the SIMD utilization to be 56.25%?

Yes, if 4 out of every 32 elements of A are divisible by 3.

What should be true about arrays B?

B can be any array of integers.

If NO, explain why not.

(e) Is it possible for this program to yield a SIMD utilization of 50% (circle one)?

YES NO

If YES, what should be true about arrays A for the SIMD utilization to be 50%?

What should be true about arrays B?

If NO, explain why not.

No. The minimum is where 1/32 elements in array A are even. This yields a 51.5625% usage.

In lecture, we learned a technique called dynamic warp formation, which tries to improve SIMD utilization, by merging threads executing the same instruction together. The key idea of dynamic warp formation is to move an "active" thread from one warp to another warp such that SIMD utilization is improved and all threads can access their registers.

Consider the following three warps X, Y, and Z that are executing the same code segment specified in the beginning of this question. Assume that the vectors we provide below specify the "active mask", i.e., whether or not the instruction should be executed by each thread in the warp: 1 means the instruction should be executed, 0 means it should not be executed. Assume each warp is at the same Program Counter.

```
Warp X = {10000000000000000000000000000010}
Warp Y = {10000000000000000000000000000001}
Warp Z = {01000000000000000000000000000000}
```

(f) Suppose that you perform dynamic warp formation on these three warps. What are the resulting active masks for each of the newly formed warps X', Y', and Z'? Explain your reasoning as necessary.

There are several answers for this question but the key is that the taken branch in Z can be combined with either X or Y. However, the taken branch in the first thread of X and Y cannot be merged because they are on the same GPU lane.
X = 10000000000000000000000000000010
Y = 11000000000000000000000000000001
Z = 00000000000000000000000000000000

(g) Given the original specification for arrays A and B (integer array A has a repetitive pattern which consists of 24 ones followed by 8 zeros, and integer array B has a different repetitive pattern which consists of 48 zeros followed by 64 ones). Is it possible for this program to yield a better SIMD utilization than the baseline when dynamic warp formation is used?

No. Branch divergence happens on the same lane throughout the program.