

CSCE 513: Computer Architecture, Fall 2018

Assignment #4, Due 11/19/2018, Monday 11:55PM

Covered topics: Data Level Parallelism

Total Points: 100 points for both undergraduate and graduate students

Submission:

- 1. Only electronic submissions on dropbox are accepted.**
- 2. You should submit two files, the conv1d.c source code file, and a SINGLE PDF file that includes everything else.**
- 3. Number your solutions in the same way and in the same order as the questions are numbered in this document and do NOT include the questions as part of your submission.**
- 4. Include your full name in the PDF file.**
- 5. Scanned copy of handwritten answers will NOT be graded.**

Data Level Parallelism:

- 4.9, b of CAQA 6th Edition (since it is single-precision, MVL = 64) (40 points)**
- 4.14 of CAQA 6th Edition (30 points)**
- 4.20 (30 points, Not a textbook question)**

4.9 [10/20/20/15/15] <4.2> Consider the following code, which multiplies two vectors that contain single-precision complex values:

```
for (i=0; i < 300; i++) {  
    c_re[i] = a_re[i] * b_re[i] - a_im[i] * b_im[i];  
    c_im[i] = a_re[i] * b_im[i] + a_im[i] * b_re[i];  
}
```

Assume that the processor runs at 700 MHz and has a maximum vector length of 64. The load/store unit has a start-up overhead of 15 cycles; the multiply unit, 8 cycles; and the add/subtract unit, 5 cycles.

- ~~a. [10] <4.3> What is the arithmetic intensity of this kernel? Justify your answer.~~
- b. [20] <4.2> Convert this loop into RV64V assembly code using strip mining.

4.14 [10/15/15] <4.5> In this exercise, we will examine several loops and analyze their potential for parallelization.

a. [10] <4.5> Does the following loop have a loop-carried dependency?

```
for (i=0; i < 100; i++) {  
    A[i] = B[2*i+4];  
    B[4*i+5] = A[i];  
}
```

b. [15] <4.5> In the following loop, find all the true dependences, output dependences, and antidependences. Eliminate the output dependences and antidependences by renaming.

```
for (i=0; i < 100; i++) {  
    A[i] = A[i] * B[i]; /* S1 */  
  
    B[i] = A[i] + c; /* S2 */  
    A[i] = C[i] * c; /* S3 */  
    C[i] = D[i] * A[i]; /* S4 */  
}
```

c. [15] <4.5> Consider the following loop:

```
for (i=0; i < 100; i++) {  
    A[i] = A[i] + B[i]; /* S1 */  
    B[i+1] = C[i] + D[i]; /* S2 */  
}
```

Are there dependences between S1 and S2? Is this loop parallel? If not, show how to make it parallel.

4.20: In this question, you will create GPU kernels of a simple 1D convolution kernel. Please see `conv1d.c` (<https://passlab.github.io/CSCE513/Assignment 4/conv1d.c>) file and there are three TODOs for you with each count for 10 points. Please follow the code comments of each TODO to implement and your submission only needs to include the `conv1d.c` file.