**CDA-5106 Assignment 3**

**4.9**

1. Arithmetic intensity = number of operations / memory accesses

Since every iteration has 6 operations (4 reads and 2 writes) and (6 \* 4) bytes are accessed,

Intensity = 6/24 = 0.25

li r1, 0 # initialize index

loop:

vld v1, a\_re + r1 # load a\_re

vld v2, b\_re + r1 # load b\_re

vmul v5, v1, v2

vld v3, a\_im + r1 # load a\_im

vld v4, b\_im + r1 # load b\_im

vmul v6, v3, v4

vsub v5, v5, v6

vst v5, c\_re + r1 # store c\_re

vmul v5, v1, v4

vmul v6, v3, v2

vadd v5, v5, v6 # a+re\*b\_im + a+im\*b\_re

vst v5, c\_im+r1 # store c\_im

bne x1, 0, else # check if first iteration

addi x1, x1, 44 # first iteration, increment by 44

jump loop

else: addi x1, x1, 256

skip: blt x1, 1200, loop

1. Chime 1: vld, vmul

Chime 2: vld, vmul

Chime 3: vsub, vst

Chime 4: vld, vmul

Chime 5: vmul, vld

Chime 6: vadd, vst

1. 6 chimes, 64 elements = 384 cycles

5 load/store (for 2 operands and 1 output, 6 cycles each) = 3\*30 = 90

8 multiples (4 cycles each) = 32

5 arithmetic (2 cycles) = 10

Total cycles per iteration = 516

Number of cycles per result = 4

1. There is no change in performance due to additional units

**4.13**

a) Throughput = Clock rate \* Number of processors \* Number of lanes \* Active threads \* Issue rate \* Percentage of single precision operations

= 1.5 \* 10 \* 8 \* 0.8 \* 0.85 \* 0.7 = **57.12 GFLOPS**

b)

1. It will be doubled to 57.12\*2 = 114.24 GLOPS
2. It will be increased by a factor of (15/10 = 1.5) = 87.725 GLOPS
3. New throughput = 1.5 \* 10 \* 8 \* 0.8 \* 0.95 \* 0.7 = 63.84 FLOPS, speed up = 1.12 times

**4.15**

1. Off chip memory reference: While accessing off-chip memory we should access consecutive locations to exploit spatial locality
2. On-chip memory reference: Exploit the on-chip memory available to reduce bank conflicts
3. Instruction issue rate: Increase instruction issue rate to increase throughput and reduce wait times
4. Memory accesses: Increase the number of active threads to be able to access more parts of the memory concurrently

**4.16**

Max attainable throughput = Clock rate \* Number of cores \* Number of lanes

= 1.5 \* 16 \* 16 = **384 GFLOPS**

Each single precision operand is of 4 bytes and each operation requires two operands

Therefore, for each operation, we have two operands and one output each of 4 bytes

Total memory required = 12 bytes

Memory bandwidth required to sustain maximum throughput = 384 \* 12 ~ **4000 GBps**

Since the maximum available bandwidth is 100 GBps, the maximum throughput is not sustainable