**Worksheet 8**

**Q1: Vectorization**

Vectorize the following code:

for(i = 0; i < M; i++){

C[i] = A[2\*i+1] + A[2\*i]\*B[2\*i]

}

Assume the base address of array A is held in x1, the base address of array B is held in x2, and the base address of array C is held in x3. The maximum vector length is held in x4. M is held in x5. Each array element is 4 bytes. You should not assume that M is an integer multiple of the maximum vector length.

**Q2. Vectorization**

How might the following code be vectorized? Clearly state any assumptions that you used for your answer for what the architecture provides, such as specific instructions, registers, etc.

for (i=0; i < N; i++){

 if(A[i+1])

 A[i] = A[i] + B[C[i]]

}

**Q3. Lane Tradeoff**

Suppose we want to add two vector registers (add v1, v2, v3), followed by another addition to different registers (add v4, v5, v6). The next instruction after that uses a different functional unit. VLR=MAXVL=32. What would you choose between an ALU with 8 lanes and 2 cycles dead time, and an ALU with 16 lanes and 8 cycles dead time?

**Q4. Vector vs Packed-SIMD**

What are distinguishing features between a vector architecture and a packed-SIMD architecture? List advantages of each.

Distinguishing feature:

Vector architecture:

Packed-SIMD:

**Q5. Vector vs GPU**

What are some distinguishing features between a vector and GPU (SIMT) architectures?

Vector:

SIMT: