EEL 6764 Principles of Computer Architecture

Practice #2

(**1**) Consider the following code. Assume maximum vector length of 64.

```
for (i=0;i<300;i++) {
   A[i] = A[i] + B[i];
}
```

(a) Rewrite the code using strip mining.

(b) Assume that ADDER is pipelined with 6 stages. Also assume that memory contains a single bank. It takes 5 cycles to load the first data element from memory, and 1 cycle for each following data element. How many cycles are required to finish the above code? Ignore write time.

(c) Now assume there is an additional ADDER same as the above. Now find the total cycles to finish the above code. Also consider the case where there are 4 ADDERs.

(d) In the previous questin, do more ADDERs help reduce runtime? explain your answer. If yes, explain if the total runtime can be reduced further if more ADDERs are added? If not, propose a solution.

(**2**) Consider the following code. Assume maximum vector length of 64, ADDER is pipelined with 6 stages, and the multiplier is pipelined with 8 stages.

```
for (i=0;i<300;i++) {
   A[i] = A[i] + B[i];
   C[i] = 2 * A[i];
}
```

(a) Assume that memory has sufficient bandwidth to supply data to keep FUs busy. Find the total number of cycles to finish the above code without chaining.

(b) Find the total number of cycles to finish the above code with chaining.

(**3**) Explain how the conditional execution of the following code is supported for a vector architecture? Be specific.

```
for (i = 0; i < 64; i=i+1)
   if (X[i] != 0)
      X[i] = X[i] - Y[i];
```

For a more general case shown below, how would it be processed by a vector machine? Propose necessary architecture features to support the following code.

```
for (i = 0; i < 64; i=i+1)
   if (X[i] > Y[i])
      X[i] = X[i] - Y[i];
   else
      X[i] = Y[i] - X[i];
```

(**4**) Memory addressing with strides is necessary to access non-continuous memory locations. Explain what problems it may cause? By specific.