Fall ’17 CIS 314 Assignment 7 – 100/100 points – Due Monday, 11/20, 11:59 PM

Please submit individual source files for coding exercises (see naming conventions below) and a single solution document for non-coding exercises (.txt or .pdf only). Your code and answers need to be documented to the point that the graders can understand your thought process. Full credit will not be awarded if sufficient work is not shown.

Suppose we’ve got a procedure that computes the inner product of two vectors u and v. Consider the following C code:

```
void inner(float *u, float *v, int length, float *dest) {
    int i;
    float sum = 0.0f;
    for (i = 0; i < length; ++i) {
        sum += u[i] * v[i];
    }
    *dest = sum;
}
```

The x86-64 assembly code for the inner loop is as follows:

```
# u in %rbx, v in %rax, length in %rcx, i in %rdx, sum in %xmm1
.L87:
    movss (%rbx, %rdx, 4), %xmm0    # Get u[i]
    mulss (%rax, %rdx, 4), %xmm0    # Multiply by v[i]
    adds %xmm0, %xmm1               # Add to sum
    addq $1, %rdx                   # Increment i
    cmpq %rcx, %rdx                 # Compare i to length
    jl .L87                         # If <, keep looping
```

1. [20] Diagram how this instruction sequence would be decoded into operations and show the data dependencies between them. Use Figure 5.14 as a guide. Include your diagram in your solutions document.

2. [20] Which operation(s) in the loop can NOT be pipelined? Why? What are the latencies of these operations? Based on this, what is the lower latency bound (in terms of CPE) of the procedure? Assume that *float* addition has a CPE of 3, *float* multiplication has a CPE of 5, and all integer operations have a CPI of 1. Write your answers in your solutions document.

3. [40] Implement a procedure *inner2* that is functionally equivalent to *inner* but uses four-way loop unrolling with four parallel accumulators. Also implement a *main* function to test your procedure. Name your source file 7-1.c.

4. [20] Using your code from part 3, collect data on the execution times of *inner* and *inner2* with varying vector lengths. Summarize your findings and argue whether *inner* or *inner2* is more

efficient than the other (or not). Create a graph using appropriate data points to support your argument. Include your summary and graph in your solutions document.

Zip the source files and solution document (if applicable), name the .zip file <Your Full Name>Assignment7.zip (e.g., EricWillsAssignment7.zip), and upload the .zip file to Canvas (see Assignments section for submission link).