# Introduction to Computer Architecture

## Assignment 3

**Due December 21, 2021**

**Grading Policy:**
**The ultimate goal for all presentation and assignments is review and understand.**
**The grading process may concentrate more on completeness than correctness.**

*****CHAPTER 3 - ILP*****

**1. [32 = 8 + 8 + 8 + 8]**

Assume the following code and latencies.

| Latencies beyond single cycle | |
|---|---|
| Memory LD | +3 |
| Memory SD | +1 |
| Integer ADD, SUB | +0 |
| Branches | +1 |
| fadd.d | +2 |
| fmul.d | +4 |
| fdiv.d | +10 |

```
Loop:   fld     f2,0(Rx)      4
I0:     fmul.d  f2,f0,f2      5
I1:     fdiv.d  f8,f2,f0      11
I2:     fld     f4,0(Ry)      4
I3:     fadd.d  f4,f0,f4      3
I4:     fadd.d  f10,f8,f2
I5:     fsd     f4,0(Ry)
I6:     addi    Rx,Rx,8
I7:     addi    Ry,Ry,8
I8:     sub     x20,x4,Rx
I9:     bnz     x20,Loop
```

11+3+4+10+3+2+2+1+1

a. What is the baseline performance (in cycles, per loop iteration) of the code sequence if no new instruction's execution could be initiated until the previous instruction's execution had completed? Ignore front-end fetch and decode. Assume for now that execution does not stall for lack of the next instruction, but only one instruction/cycle can be issued. Assume the branch is taken, and that there is a one-cycle branch delay slot.

b. Consider a multiple-issue design. Suppose you have two execution pipelines, each capable of beginning execution of one instruction per cycle, and enough fetch/decode

bandwidth in the front end so that it will not stall your execution. Assume results can be immediately forwarded from one execution unit to another, or to itself. Further assume that the only reason an execution pipeline would stall is to observe a true data dependency. Now how many cycles does the loop require?
c. Reorder the instructions to improve performance of the code. How many cycles does your reordered code take?
d. Use loop unrolling to unroll two iterations of the loop in the reordered code in question c.

**2. [10 = 5 + 5]**
Please explain the principles of register renaming and provide an example code snippet to showcase how it can be accelerated by register renaming.

**3. [10 = 5 + 5]**
Consider a branch-target buffer that has penalties of zero, two, and two clock cycles for correct conditional branch prediction, incorrect prediction, and a buffer miss, respectively. Consider a branch-target buffer design that distin- guishes conditional and unconditional branches, storing the target address for a conditional branch and the target instruction for an unconditional branch.

a. What is the penalty in clock cycles when an unconditional branch is found in the buffer?
b. Determine the improvement from branch folding for unconditional branches. Assume a 90% hit rate, an unconditional branch frequency of 5%, and a two-cycle penalty for a buffer miss. How much improvement is gained by this enhancement? How high must the hit rate be for this enhancement to provide a performance gain?

*****CHAPTER 4 - DLP*****
**4. [24 = 8 + 8 + 8]**
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. What is the arithmetic intensity of this kernel?
b. Assuming chaining and a single memory pipeline, how many chimes are required? How many clock cycles are required per complex result value, including start-up overhead?
c. Now assume that the processor has three memory pipelines and chaining. If there

are no bank conflicts in the loop's accesses, how many clock cycles are required per result?

**5. [24 = 8 + 8 + 8]**

Assume a GPU architecture that contains 10 SIMD processors. Each SIMD instruction has a width of 32 and each SIMD processor contains 8 lanes for single-precision arithmetic and load/store instructions, meaning that each nondiverged SIMD instruction can produce 32 results every 4 cycles. Assume a kernel that has divergent branches that causes, on average, 80% of threads to be active. Assume that 70% of all SIMD instructions executed are single-precision arithmetic and 20% are load/store. Because not all memory latencies are covered, assume an average SIMD instruction issue rate of 0.85. Assume that the GPU has a clock speed of 1.5 GHz.

Assume that you have the following choices:
a. Increasing the number of single-precision lanes to 16
b. Increasing the number of SIMD processors to 15 (assume this change doesn't affect any other performance metrics and that the code scales to the additional processors)
c. Addingacachethatwilleffectivelyreducememorylatencyby40%,which will increase instruction issue rate to 0.95

What is speedup in throughput for each of these improvements?