Department of Computer Science and Engineering
National Sun Yat-sen University
Second Semester of 2022 PhD Qualifying Exam

Subject： Computer Architecture

Problem 1: (30 points)

Vector processors are used to improve the performance of the vectorized data processing. A *Convoy* in a vector machine is defined that a set of vector instructions can be potentially executed together. Besides, a *Chime* is usually used to measure the length of a convoy, and it is simply the unit of time taken to execute one convoy. Please consider the following code sequence

```
vld       v0, x5       #Load vector X
vmul v1, v0, f0        #Vector-scalar multiply
vld       v2, x6       #Load vector Y
vadd      v3, v1, v2   #Vector-vector add
vst       v3, x6       #Store the sum
```

1. (10%) How the code sequence lays out in convoys, assuming a single copy of each vector function unit.
2. (10%) How many chimes will this vector sequence take?
3. (10%) How many cycles per FLOP (floating-point operation) are needed, ignoring vector instruction issue overhead.

Problem 2: (30 points)

Figure 1 shows a 5-stage pipelined MIPS CPU to support a simple code in Figure 2. We need to spend one clock cycle to complete all tasks in each pipeline stage.

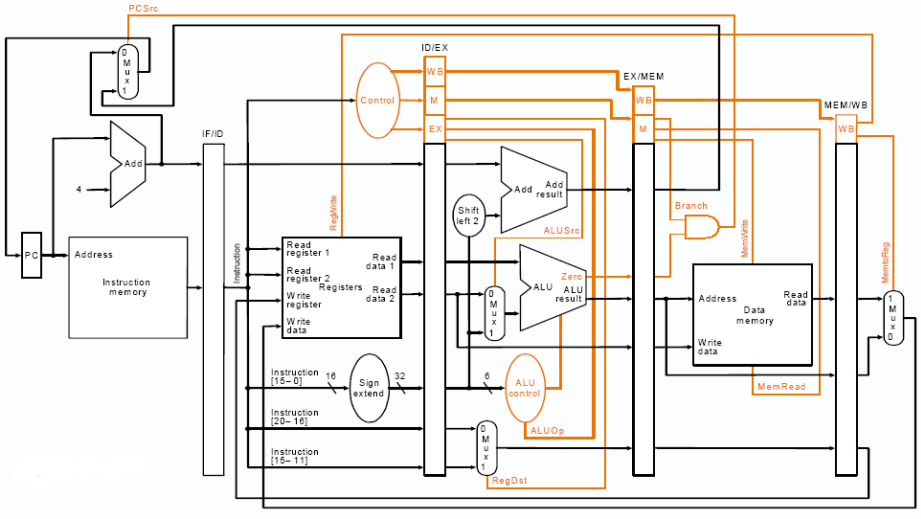

Figure 1

```
sub   $2, $1, $3
and   $12, $2, $5
or    $13, $6, $2
add   $14, $2, $2
sw    $15, 100($2)
```

Figure 2

1. (10%) Do we need to stall this architecture in Figure 1 to run the code in Figure 2? If yes, please determine how many cycles we should stall and explain your reasons. If not, please describe your reason.
2. (10%) To support the code in Figure, please design a minimum viable product (MVP) of the ALU in Figure 1. You need to show the ALU architecture.
3. (10%) Is it possible to move the Adder from the third pipeline stage to the second pipeline stage? If yes, please explain your re-design strategy. If not, please describe your reason.

Problem 3: (20 points)

Suppose we have an application running on a 100-processor multiprocessor, and assume that application can use 1, 50, or 100 processors. If we assume that 95% of the time we can use all 100 processors, how much of the remaining 5% of the execution time must employ 50 processors if we want a speedup of 80%?

Problem 4: (20 points)

Consider an 8-processor multicore where each processor has its own L1 and L2 caches, and snooping is performed on a shared bus among the L2 caches. Assume the average L2 request, whether for a coherence miss or other miss, 15 cycles. Assume a clock rate of 3.0GHz, a CPI of 0.7, and a load/store frequency of 40%. If our goal is that no more than 50% of the L2 bandwidth is consumed by coherence traffic, what is the maximum coherence miss rate per processor?