CSE 140 HW #9

1. Consider the following piece of C code:

```
for (j = 2; j < 1000; j++)
   D[j] = D[j – 1] + D[j – 2];
```

The MIPS code corresponding to the above fragment is:

```
       addiu $s2, $zero, 8000
       addiu $s1, $zero, 16
loop:  l.d   $f0, -16($s1)
       l.d   $f2, -8($s1)
       add.d $f4, $f0, $f2
       s.d   $f4, 0($s1)
       addiu $s1, $s1, 8
       bne   $s1, $s2, loop
```

   a. When an instruction in a later iteration of a loop depends upon a data value produced in an earlier iteration of the same loop, we say that there is a loop carried dependence between iterations of the loop. Identify the loop-carried dependences in the above code. Identify the dependent program variable and assembly-level registers. You can ignore the loop induction variable j.

   b. Re-write the assembly code above so that the loop is unrolled by 4 iterations.

2. Consider the following three CPU organizations:

   CPU SS: A 2-core superscalar microprocessor that provides out-of-order issue capabilities on 2 function units (FUs). Only a single thread can run on each core at a time.

   CPU MT: A fine-grained multithreaded processor that allows instructions from 2 threads to be run concurrently (i.e., there are two functional units), though only instructions from a single thread can be issued on any cycle.

   CPU SMT: An SMT processor that allows instructions from 2 threads to be run concurrently (i.e., there are two functional units), and instructions from either or both threads can be issued to run on any cycle.

   Assume we have two threads X and Y to run on these CPUs that include the following operations:

Thread X	Thread Y
A1 – takes 3 cycles to execute	B1 – take 2 cycles to execute
A2 – no dependences	B2 – conflicts for a functional unit with B1
A3 – conflicts for a functional unit with A1	B3 – depends on the result of B2
A4 – depends on the result of A3	B4 – no dependences and takes 2 cycles to execute

Assume all instructions take a single cycle to execute unless noted otherwise or they encounter a hazard.

a. List out the operation sequence for each core of the CPU SS that executes these two threads.
b. List out the operation sequence for each functional unit of the CPU MT that executes these two threads.
c. List out the operation sequence for each functional unit of the CPU SMT that executes these two threads.