

### Problem 5 (15 points)

A programmer writes the following two C code segments. She wants to run them concurrently on a multicore processor, called SC, using two different threads, each of which will run on a different core.

Thread T1

```
a = X[0];
```

```
b = a + Y[0];
```

```
while(*flag == 0);
```

```
Y[0] += 1;
```

Infinite loop until flag becomes 1,  
at which point Y[0] must be 1. So  
in the end Y[0] could only be 2.

Thread T2

```
Y[0] = 1;
```

```
*flag = 1;
```

```
X[1] *= 2;
```

```
a = 0;
```

X, Y, and flag have been allocated in main memory, while a and b are contained in the processor registers. A read or write to any of these variables generates a single memory request. The initial values of all memory locations and variables are 0. Assume each line of the C code segment of each thread translates to a single machine instruction.

**Part a (5 points):** Both threads have a variable a. Are they referring to the same variable?

No

**Part b (5 points):** What are the possible final value(s) of Y[0] after both threads finish execution? Consider all the possible thread interleavings.

2

**Part c (5 points):** What are the possible final value(s) of b after both threads finish execution? Consider all the possible thread interleavings.

0 and 1