

NAME: \_\_\_\_\_

**Problem Q5.4: Where has all the speedup gone?      26 points**

In each of the following problems, we consider a parallel program running on a cache-coherent, shared-memory multiprocessor. Each program is correct as written but suffers poor performance.

**Problem Q5.4.A**

**8 points**

The following code performs vector-vector addition. The integer  $P$  equals the number of processors running the program, and the integer  $id$  contains a given processor's ID number, which lies in the range  $[0, P-1]$ .  $N$  is the vector size; you may assume that  $N$  is large and that  $P$  divides  $N$ .

```
int P, id;
int Z[N], X[N], Y[N];

for(int i = id; i < N; i += P)
    Z[i] = X[i] + Y[i];
```

Unfortunately, false sharing eliminates any hope of parallel speedup. Rewrite the code to avoid false sharing to the extent possible.

False sharing will be essentially eliminated if the processors operate on contiguous chunks (i.e. unit-stride).

```
for(int i = (N/P)*id; i < (N/P)*(id+1); i++)
    Z[i] = X[i] + Y[i];
```