Problem 40. (9 points):

The following problem concerns optimizing a procedure for maximum performance on an Intel Pentium III. Recall the following performance characteristics of the functional units for this machine:

Operation	Latency	Issue Time
Integer Add	1	1
Integer Multiply	4	1
Integer Divide	36	36
Floating Point Add	3	1
Floating Point Multiply	5	2
Floating Point Divide	38	38
Load or Store (Cache Hit)	1	1

Consider the following two procedures:

Loop 1

```
int loop1(int *a, int x, int n)
{
  int y = x*x;
  int i;
  for (i = 0; i < n; i++)
    x = y * a[i];
  return x*y;
}
```

Loop 2

```
int loop2(int *a, int x, int n)
{
  int y = x*x;
  int i;
  for (i = 0; i < n; i++)
    x = x * a[i];
  return x*y;
}
```

When compiled with GCC, we obtain the following assembly code for the inner loop:

Loop 1

```
.L21:
   movl %ecx,%eax
   imull (%esi,%edx,4),%eax
   incl %edx
   cmpl %ebx,%edx
   jl .L21
```

Loop 2

```
.L27:
   imull (%esi,%edx,4),%eax
   incl %edx
   cmpl %ebx,%edx
   jl .L27
```

Running on one of the Fish machines, we find that Loop 1 requires 3.0 clock cycles per iteration, while Loop 2 requires 4.0.

A. Explain how it is that Loop 1 is faster than Loop 2, even though it has one more instruction

B. By using the compiler flag `-funroll-loops`, we can compile the code to use 4-way loop unrolling. This speeds up Loop 1. Explain why.

C. Even with loop unrolling, we find the performance of Loop 2 remains the same. Explain why.