Anica Wang

305-532-153

Professor Reinman

CS 33

Homework 3

5.19

In Problem 5.12, we were able to reduce the CPE for the prefix-sum computation to 3.00, limited by the latency of floating-point addition on this machine. Simple loop unrolling does not improve things. Using a combination of loop unrolling and reassociation, write code for a prefix sum that achieves a CPE less than the latency of floating-point addition on your machine. Doing this requires actually increasing the number of additions performed. For example, our version with two-way unrolling requires three additions per iteration, while our version with four-way unrolling requires five. Our best implementation achieves a CPE of 1.67 on our reference machine. Determine how the throughput and latency limits of your machine limit the minimum CPE you can achieve for the prefix-sum operation.

void psum1a(float a[], float p[], long n)

{

long i;

float val, last\_val;

last\_val = p[0] = a[0];

for (i = 1, i < n – 4; i++)

{

p[i] = last\_val + a[i];

p[i + 1] = f1 + a[i + 1];

p[i + 2] = f2 + a[i + 2];

p[i + 3] = f3 + a[i + 3];

last\_val = last\_val + (a[i] + [i + 1] + [i + 2] + [i + 3]);

}

for (i; i < n; i++)

{

last\_val += a[i];

p[i] = last\_val;

}

}