

PCA Laboratory 2

Adur Saizar
Josep Oriol Vilarrubí
pca27

April 8, 2014

Contents

1	Inlining	2
2	Loop Unrolling	2
2.1	Inner loop	2
2.2	Medium loop	2
2.3	Outer loop	2
3	Loop fusion and other optimization applied to pi.c	2

1 Inlining

the time obtained without inlining is 3.55 seconds, when applying inlining we obtain an speed-up of 1.23X because it inlines all the routines insides the function calculate, the number of operations of branching and the multiplication get highly reduced, is not always a good option to force the inline of a function as we've seen in the case when we forced the DIVIDE function to be inlined.

2 Loop Unrolling

The original code of matriu4x4 takes 2,017 seconds to complete, we try to optimize that by unrolling some loops

2.1 Inner loop

1. Time: 1,361
2. Speed-up: 1,48x
3. number of instruction: $89 * 16 = 1424$

2.2 Medium loop

1. Time: 1.187
2. Speed-up: 1,7x
3. number of instruction: $339 * 4 = 1356$

2.3 Outer loop

1. Time: 0.099
2. Speed-up: 20,4x
3. number of instruction: 383

The best unrolling degree in this case is to unroll all the loop(4^3) this is because you don't need to calculate any effective address as the preprocessor would do it for you.

The version inlined by hand is better than the one with O3 due to the fact that we pass the array pointers as arguments, the compiler cannot compute the effective address during the compilation process, and it has to add the relative offset to the array initial address.

3 Loop fusion and other optimization applied to pi.c

We applied loop fusion to the pi.c code in the substract function that are nearby, the another fusion we've done is the DIVIDE25,the both of the DIVIDE239 and the LONGDIV next to it, also we've substituted the set(c,1) function for an

static array of ones, after the loop we fused two subtract and two divides, and also we've buffered the progress function and done some conditional branch removal in the SUSBTRACT function.