Abstract

Caches improve the speed of programs by reducing the number of accesses to the slowly main memory. Unfortunately, most programs which work with arrays suffer from cache conflicts. Cache conflicts slow down the programs and counteract the advantages of the cache. The speed of affected programs can be improved by avoiding cache conflicts. This thesis concentrates on the question *how to avoid cache conflicts with software methods.* I describe

- the state of the art software methods
- how to pad for tiling
- a new padding algorithm (Odd-Padding)
- a new technique to avoid cache conflicts (Tetris)
- experiments which compare these techniques

Furthermore, I prove the odd-padding algorithm correct.