

Podemos diminuir sensivelmente o tempo de execução de um programa assembler, por um microprocessador, se dispusermos de um sistema computacional que execute otimizações de código durante o tempo de compilação. A base desta pesquisa visa implementar um sistema descrito em software para tal. Desenvolvemos então um software que, após receber um programa, modifica-o até obter uma forma otimizada que garanta ganho de velocidade quando em processamento. Esta otimização consiste em identificar no programa rotinas críticas, instruções onde há utilização de “hardware dedicado”, que possibilita obter melhorias, se submetidas à otimizações. Ao identificar estas rotinas críticas, são efetuadas trocas de posição entre as operações dependentes e as não dependentes do “hardware dedicado”. Quando as trocas não obtêm um resultado satisfatório, ou não é possível a sua realização, pode-se efetuar *loop_unroll*, uma técnica que basicamente consiste em, quando encontra-se uma rotina crítica e instruções insuficientes para uma otimização, efetuar a duplicação do código referente às instruções daquela rotina. Ao duplicar-se o código geram-se dois laços iguais, dividindo pela metade o número final de vezes ao qual cada laço deveria ser executado. Assim, o aproveitamento do paralelismo entre funções torna-se mais eficiente. (CNPq)