Otimizações Específicas de Máquina

Marco Túlio 202100560105
Vinícius Schiavon 202100560434

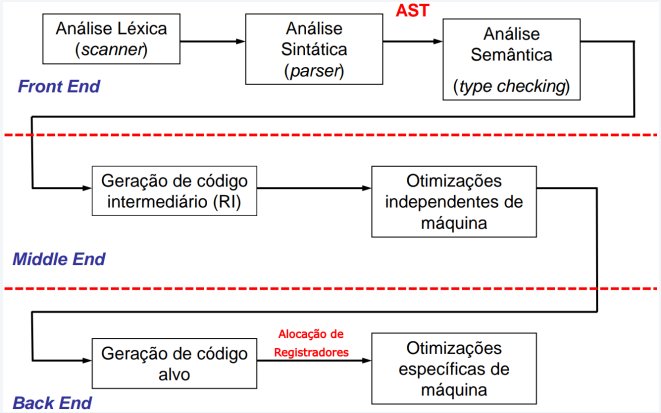
Departamento de Computação
Centro de Ciências Exatas
Universidade Estadual de Londrina

03 de maio de 2024

Índice

Relembrando o Fluxo do Compilador

Fluxo do Compilador



Objetivo das Otimizações

- Transformar o código para que ele consuma menos recursos, como tempo, memória e/ou energia
- Deve manter o significado do algoritmo
- Aumentar a velocidade e performance do programa

Divisão das Otimizações

Independentes e Dependentes de Máquina

Independentes

- Não considera as especificidades do hardware
- Utiliza o código intermediário
- Tenta produzir um melhor código alvo
- Envolve registradores virtuais

Dependentes (ou Específicas)

- Considera as especificidades do hardware
- Utiliza o código alvo
- Tenta obter vantagem das características do hardware
- Envolve registradores físicos

Diferenças entre Máquinas

Relembrando ISA

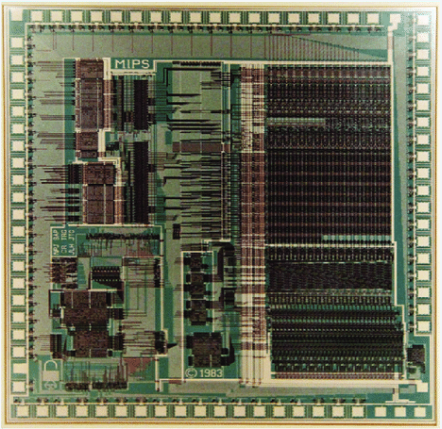
Instruction Set Architecture (ISA):

- Abstração do modelo do computador que define como a CPU é controlada pelo software
- Interface entre o hardware e o software de baixo nível
- Padroniza instruções, padrões de bits da linguagem de máquina, estágios de pipeline, etc.

Exemplos: MIPS, ARM, x86-64

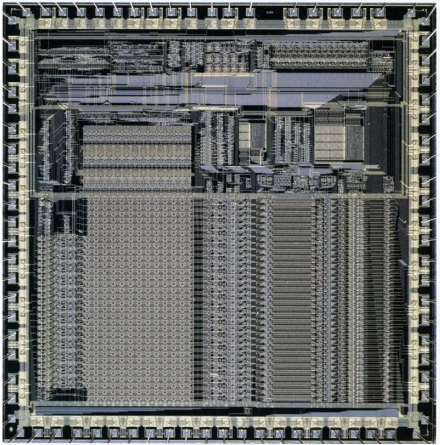
MIPS

- Número de registradores: 32
- Categoria: RISC
- Processadores famosos: R5900 (PlayStation 2), R4300i (Nintendo 64), Mobileye EyeQ3 (Tesla Model S)



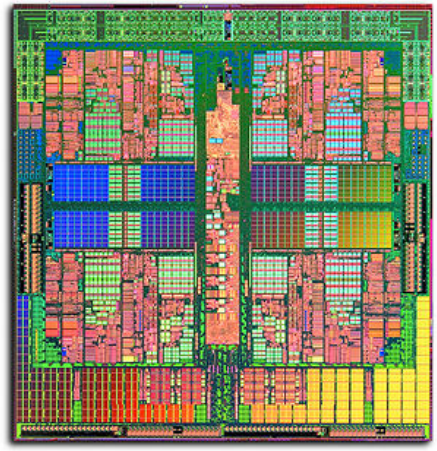
ARM

- Número de registradores: 37
- Categoria: RISC
- Processadores famosos: Snapdragon (Qualcomm), Exynos (Samsung), A-Series (Apple)



x86-64

- Número de registradores: 16
- Categoria: CISC
- Processadores famosos: Athlon 64 (AMD), Xeon (Intel), Core Series (Intel)



Usos das Otimizações Dependentes

Possíveis aplicações práticas

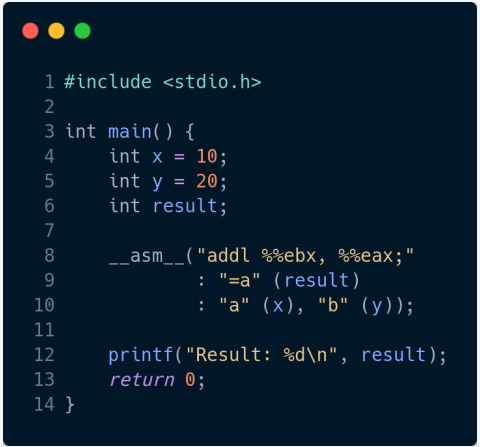
- Computação Científica: processar ou simular dados volumosos pode ser melhorado com SIDM (Single Instruction, Multiple Data) ou outras funcionalidades específicas
- APIs: CUDA ou OpenCL nas GPUs podem melhorar processamento em paralelo, e até influenciar no desempenho de videogames
- Sistemas de tempo real: aviões, por exemplo, exigem restrições estritas de tempo e tirar proveito da máquina pode ser importante

Otimização ”manual”

Incorporando instruções assembly no código fonte:

- Utiliza instruções específicas da máquina no código fonte
- Permite uma manipulação mais próxima do hardware
- Pode melhorar performance
- Não é propriamente uma técnica de compilação, mas sim uma otimização específica para a máquina alvo

Exemplo de incorporação em C



```c
#include <stdio.h>

int main() {
    int x = 10;
    int y = 20;
    int result;

    __asm__("addl %%ebx, %%eax;"
            : "=a" (result)
            : "a" (x), "b" (y));

    printf("Result: %d\n", result);
    return 0;
}
```

Tipos de Otimizações Dependentes e Exemplos

Peephole

- Analisa pequenas seções
- Evita acessos repetidos de memória
- Exemplo: Redundant Load Elimination

Exemplo para Redundant Load Elimination

Before:	After:
`MOVQ %R8, x` `MOVQ x, %R8`	`MOVQ %R8, x`

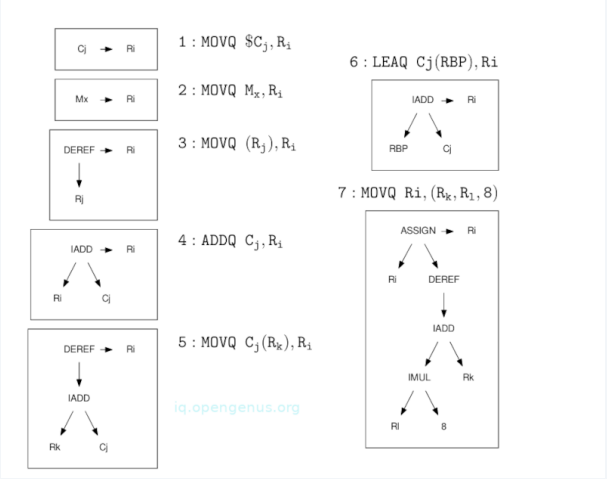
Before:	After:
`MOVQ %R8, x` `MOVQ x, %R9`	`MOVQ %R8, x` `MOVQ %R8, %R9`

O último exemplo economiza loads desnecessários e evita atraso de pipeline

Instruction Selection

- Existem instruções que combinam múltiplas operações (ponteiros, acessos de memória, aritméticas)
- Explorar essas instruções poderosas durante leitura da árvore (tree coverage após geração do código alvo)
- Colocar endereços de memória, constantes ou registradores nas folhas das árvores de template

Subset de instruções x86



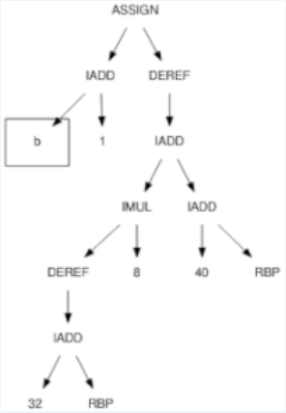
Exemplo para Instruction Selection

Otimizar um código x86 que realiza: a[i] = b + 1
Onde:

- b: variável global
- a: variável local na posição 40
- i: variável local na posição 32

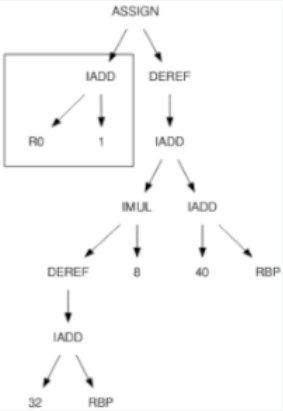
Exemplo para Instruction Selection

1° Carregar b em registrador



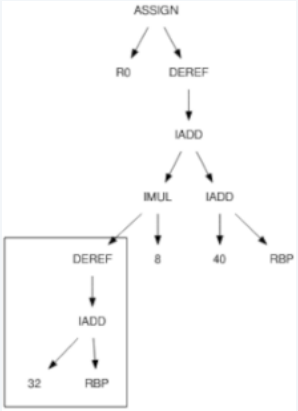
Exemplo para Instruction Selection

2° Computar b + 1



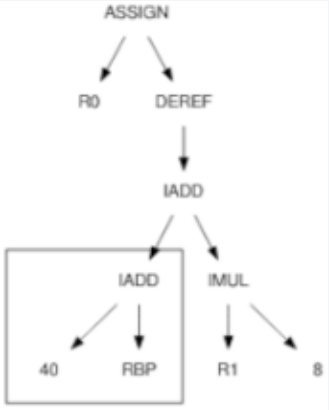
Exemplo para Instruction Selection

3° Carregar index i em registrador



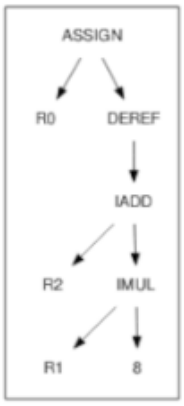
Exemplo para Instruction Selection

4° Computar index aplicado no vetor



Exemplo para Instruction Selection

5° Armazenar resultador de b + 1 na posição i do vetor a



Exemplo para Instruction Selection

Resultado

```
MOVQ b,%R0
ADDQ $1,%R0
MOVQ 32(%RBP),%R1
LEAQ 40(%RBP),%R2
MOVQ %R0,(%R2,%R1,8)
```

Finalização

Compiler Explorer

- Ferramenta interessante para explorar opções de diferentes arquiteturas
- Flag -O0 para remover otimizações
- Flag -O3 para aplicar 3 camadas de otimização
- https://godbolt.org/

Áreas em aberto

- Uso de Redes Neurais para o desenvolvimento de uma extensão do conjunto de instruções de uma arquitetura
- Estudo de compiladores otimizantes para computadores quânticos
- Criação, ou adaptação, de compiladores para novas arquiteturas, por exemplo AMD Zen

Referências

GeeksforGeeks

Code Optimization in Compiler Design
Machine Dependent and Machine Independent Code Optimization
Microarchitecture and Instruction Set Architecture

IEEEXplore

Designing RISC-V Instruction Set Extensions for Artificial Neural Networks: An LLVM Compiler-Driven Perspective
Compiler Optimization for Quantum Computing Using Reinforcement Learning

Referências

ARM
Instruction Set Architecture (ISA)

Naukri360
Machine Dependent Code Optimization

OpenGenius IQ
Machine Dependent Optimizations

Obrigado pela atenção

Marco Túlio 202100560105
Vinícius Schiavon 202100560434

Departamento de Computação
Centro de Ciências Exatas
Universidade Estadual de Londrina

03 de maio de 2024


Senhores, teriam algumas migalhas de
votos para estes cidadãos necessitados?