Sistemas de Computação

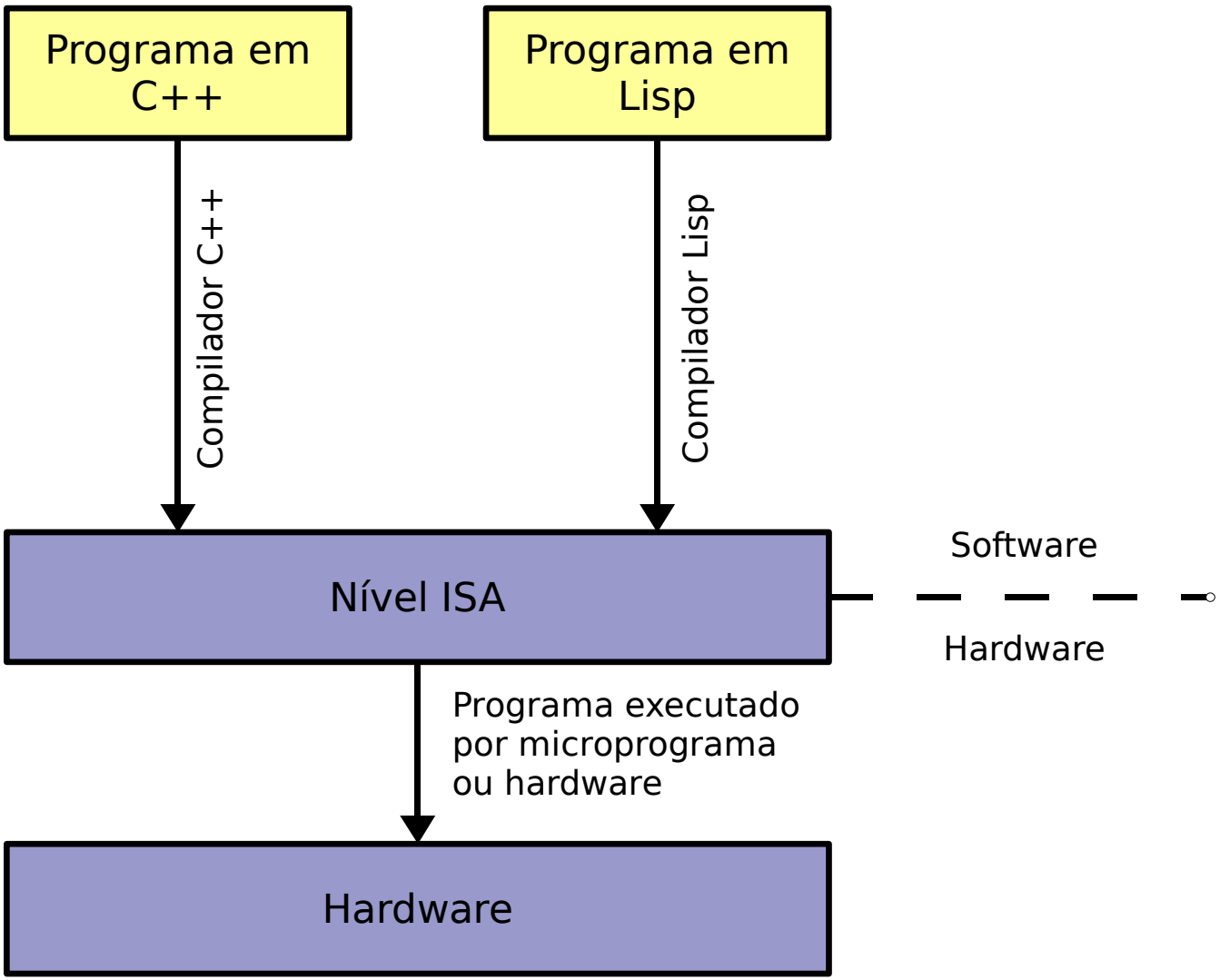
Sétima Aula

Haroldo Gambini Santos

Universidade Federal de Ouro Preto - UFOP

2 de setembro de 2009

De Compiladores Até o Hardware



Representação de Instruções

- Instruções são números (grupamento de 0´s e 1´s)
- Instruções são compostas por 2 tipos de campos a saber:

  OP CODE código da instrução. Indica o que a ALU irá fazer (soma, subtração, etc.)

  Operando(s) indica a CPU com que dados a CPU realizará a operação (podem ser endereços ou mesmo valores)
- 1 OP CODE por instrução
- 0 ou mais operandos por instrução

Mnemônicos

- codificação de instruções para facilitar o programador na tarefa de escrever código de baixo nível
- ao invés de 01010001110101000101000111010100
- `ADD, A, B, D`

Endereços mais curtos

Hexadecimal

- também chamado base-16, hexa ou hex

Endereços mais curtos

Hexadecimal

- também chamado base-16, hexa ou hex
- símbolos: `{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}`
- representação mais amigável de valores binários
  - cada letra representa um *nibble* (4 bits)

Endereços mais curtos

Hexadecimal

- também chamado base-16, hexa ou hex
- símbolos: `{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F}`
- representação mais amigável de valores binários
  - cada letra representa um *nibble* (4 bits)
- exemplo:
  - `01101011000011011111000011001010`
  - em hex: `6B0DF18A`

Representação de Instruções

Instruções sem operandos

- `RESET`
- `NOP`
- ...

Representação de Instruções

Instruções sem operandos

- `RESET`
- `NOP`
- ...

Instruções com 1 operando

- `ADD A` (ACC = ACC + A)
- ...

Representação de Instruções

Instruções com 2 operandos

- `ADD A, B` (A = A + B)
- `MOVE A, B` (A = B)
- ...

Representação de Instruções

Instruções com 2 operandos

- `ADD A, B` ($A = A + B$)
- `MOVE A, B` ($A = B$)
- ...

Instruções com 3 operandos

- `ADD A, B, C` ($A = B + C$)
- `MUL A, B, C` ($A = B \times C$)

Representação de Instruções

Instruções com 4 operandos

- `ADD A, B, C, 04h` (A = B + C, próx. instr. em 04h)
- ...

Arquitetura de um computador

- CPU: caminho de dados + controle
- Organização da memória
- Conjunto de instruções básicas
  - pontos de vista: **Projetista** $\times$ **Programador**
  - instruções, registradores e endereços de memória são números

ISA

Instruction Set Architecture

- descreve os aspectos da arquitetura do computador, "visíveis" ao programador
- Aspectos considerados:
  - tipos primitivos de dados
  - instruções
  - número de operandos
  - registradores (tipos e forma de acesso)
  - forma de endereçamento (acesso) à memória
  - ...

ISA

Instruction Set Architecture

- inclui todos os comandos capazes de serem reconhecidos pelo projeto de uma CPU
- esses comandos são instruções
- cada instrução possui um OP CODE próprio
- o conjunto dessas instruções (OP CODES) de cada CPU forma o que chamamos de linguagem de máquina da CPU. (assembly)

ISA : Decisões importantes

- operações
  - quantas
- operandos
  - quantos
  - localização
  - tipos
  - como especificar
- formato de instruções
  - tamanho
  - quantos formatos

ISA

Localização de Operadores

- Acumulador
- Pilha
- Registradores
- Memória
- Tipos comuns de máquinas:
  - acumulador
  - pilha
  - registrador-memória
  - load-store

ISA

Quantos operandos

- Acumulador
  - 1 endereço: `ADD A` (acc ← acc + mem[A])
- Pilha
  - 0 endereços: `ADD` (tos ← tos + next)
- *General Purpose Register*
  - 2 endereços: `ADD A B` (A ← A + B)
  - 3 endereços: `ADD A B` (A ← B + C)

ISA


Acumulador Simples(EDSAC 1950)
Accumulator +Index Registers
(Manchester Mark I, IBM 700 series 1953)
Separação do Modelo de Programação
da Implementação
Baseado em Linguagens
de Alto Nível
(B5000 1963)
Conceito de Família
(IBM 360 1964)
General Purpose Register Machines
Complex Instruction Sets
(Vax, Intel 432 1977-80)
Load/Store Architecture
(CDC 6600, Cray 1 1963-76)
RISC
(Mips,Sparc,88000,IBM RS6000, . . .1987)

ISA - Classes

Acumulador

Vantagens minimiza os estados na máquina; instruções curtas

Desvantagens acumulador é o único local de armazenamento temporário - icorre em uma comunicação maior com a memória

ISA - Classes

Acumulador

Vantagens minimiza os estados na máquina; instruções curtas

Desvantagens acumulador é o único local de armazenamento temporário - icorre em uma comunicação maior com a memória

Pilha

Vantagens instruções curtas; avaliação fácil de expressões

Desvantagens pilha não pode ser acessada aleatóriamente - difícil de se gerar código eficiente; implementação eficiente também difícil

ISA - Classes

Registradores

Vantagem modelo mais geral para geração de código

Desvantagem todos os operandos precisam ser nomeados, o que gera instruções mais longas

- registradores são mais rápidos do que a memória
- registradores são mais fáceis para o compilador utilizar
- registradores podem ser utilizados mais eficientemente do que outras formas de armazenamento interno

ISA

Operações comuns

lógicas e aritméticas soma, E, diferença, OU

transferência de dados loads/stores

controle branch, jump, ...

sistema gerenciamento de cache, memória virtual, ...

ponto flutuante operações de PF: soma, multiplicação,...

string operações em cadeias de caracteres: mover, comparar, buscar,...

ISA

posição	instrução	% perc. execuções
1	load	22,00%
2	conditional branch	20,00%
3	compare	16,00%
4	store	12,00%
5	add	8,00%
6	and	6,00%
7	sub	5,00%
8	move register-register	4,00%
9	call	1,00%
10	return	1,00%
		total: 96%