

Arquitetura e Organização de Computadores II

Introdução

Prof. Daniel S. Marcon
danielstefani@unisinos.br

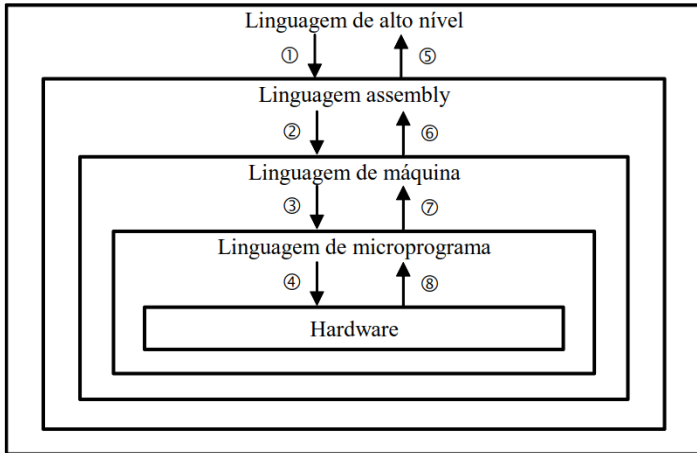
Arquitetura vs. Organização

Qual é a diferença entre a **arquitetura** e a **organização** de um computador?

Arquitetura vs. Organização

- **Arquitetura:** corresponde aos aspectos visíveis a um programador em linguagem de máquina
 - Ex: repertório de instruções, número de bits utilizado para representar vários tipos de dados, mecanismo de E/S e modos de endereçamento
- **Organização:** diz respeito às unidades operacionais (ex: CPU, unidade de memória, barramentos e sinais de controle) necessárias para implementar as especificações de uma arquitetura
 - Geralmente transparente ao programador
- Exemplo: operação de multiplicação
 - Aspecto arquitetônico: saber se o computador provê uma instrução para a operação
 - Aspecto organizacional: como a operação é implementada

Etapas de Codificação



Etapas de Codificação

- 1 Alto nível → assembly: tradutor ou compilador específico para a plataforma alvo
- 5 Assembly → alto nível: ferramentas de análise específicas para a linguagem a ser extraída
 - O código gerado para a linguagem de alto nível será idêntico ao código original?

Etapas de Codificação

- 1 Alto nível → assembly: tradutor ou compilador específico para a plataforma alvo
- 5 Assembly → alto nível: ferramentas de análise específicas para a linguagem a ser extraída
 - O código gerado para a linguagem de alto nível será idêntico ao código original?
 - Não, por causa de questões de otimização dos compiladores e de questões semânticas
- 2 Assembly → linguagem de máquina: montador (em inglês, assembler)
- 6 Linguagem de máquina → assembly: desassemblador
 - O código gerado para assembly será idêntico ao código original?

Etapas de Codificação

- 1 Alto nível → assembly: tradutor ou compilador específico para a plataforma alvo
- 5 Assembly → alto nível: ferramentas de análise específicas para a linguagem a ser extraída
 - O código gerado para a linguagem de alto nível será idêntico ao código original?
 - Não, por causa de questões de otimização dos compiladores e de questões semânticas
- 2 Assembly → linguagem de máquina: montador (em inglês, assembler)
- 6 Linguagem de máquina → assembly: desassemblador
 - O código gerado para assembly será idêntico ao código original?
 - Sim, pois existe uma relação unívoca entre as linguagens

Etapas de Codificação

- 3 Linguagem de máquina → microprograma: ferramentas específicas que conhecem a organização de cada processador (em geral, simuladores de arquiteturas)
 - Essas ferramentas interpretam as instruções mapeadas em linguagem de máquina e conhecem os recursos disponíveis no processador de forma a saber quais as ações que são necessárias para o seu funcionamento
- 7 Microprograma → linguagem de máquina: mesmas ferramentas do item anterior
 - Processo não usual
- 4 Microprograma → hardware: ocorre de forma semelhante ao item 3, porém com maior detalhamento
- 8 Hardware → microprograma: ocorre de forma semelhante ao item 7, porém com maior detalhamento

Relação entre Linguagens de Alto e Baixo Nível

- Velocidade? Acesso a recursos da máquina? Simplicidade de programação? Inteligibilidade? Facilidade de alteração? Facilidade de manutenção? Portabilidade?

Relação entre Linguagens de Alto e Baixo Nível

- Velocidade? Acesso a recursos da máquina? Simplicidade de programação? Inteligibilidade? Facilidade de alteração? Facilidade de manutenção? Portabilidade?

	Alto nível	Baixo nível
Velocidade	programas lentos	programas rápidos
Acesso a recursos da máquina	menor liberdade	maior liberdade
Simplicidade de programação	alta	baixa
Inteligibilidade	alta	baixa
Facilidade de alteração	alta	baixa
Facilidade de manutenção	alta	baixa
Portabilidade	alta	baixa

Complex Instruction Set Computer (CISC)

- No começo da década de 80, a tendência era construir chips com um conjunto cada vez mais complexo de instruções
- Microprogramação: criar novas funções adicionadas diretamente ao hardware, facilitando o trabalho do programador
- Resultado: programas com poucas instruções
 - Menos acesso à memória

Complex Instruction Set Computer (CISC)

- Introdução em hardware de novas operações e modos de endereçamento:
 - Substituir sequências de operações aritméticas primitivas
 - Suportar operações repetitivas
 - Fornecer métodos alternativos de endereçamento
 - Auxiliar invocação de procedimentos
 - Auxiliar execução de funções do sistema operacional
 - Facilitar construção de multiprocessadores

Complex Instruction Set Computer (CISC)

- Motivações:
 - Velocidade da memória vs. velocidade da CPU
 - Densidade do código: programas com poucas instruções representam menos acessos à memória
 - Compatibilidade entre máquinas: manter as instruções de modelos anteriores
 - Suporte para linguagens de alto nível: aproximar linguagem de máquina das novas linguagens de alto nível e reduzir a disparidade semântica

Complex Instruction Set Computer (CISC)

- Características comuns à maior parte dos processadores CISC:
 - Grande número de instruções (i386: 111 instruções e i486: 157 instruções)
 - Codificação de instruções em tamanho variável para otimizar memória
 - Execução em microcódigo
 - Taxas variáveis de execução, dependendo da instrução (dificulta o pipeline)
 - Vários modos de endereçamento (i386: 8 modos e i486: 12 modos)
 - Operações diversas envolvendo a memória principal

Complex Instruction Set Computer (CISC)

- Desvantagem?

Complex Instruction Set Computer (CISC)

- Desvantagem?
- Processador **sobrecarregado, complexo, maior e mais lento**

Reduced Instruction Set Computer (RISC)

- 85% do programa consiste em apenas três tipos de instruções: assinalamentos, comandos if e chamadas de procedimentos
 - Desnecessário a inclusão de microprogramas no processador que quase nunca são utilizados
 - Instruções complexas devem ser incluídas somente quando o benefício no desempenho compensar a degradação de velocidade
- Pode-se otimizar a área disponível do processador para diversos objetivos, incluindo criar novas instruções, aumentar o número de registradores, incluir memória cache ou adicionar unidades de execução

Reduced Instruction Set Computer (RISC)

- O uso de microprogramação deve ser evitado
 - Aumento significativo da velocidade das memórias resultou na utilização de software em substituição aos microprogramas
- O compilador deve substituir eficientemente as operações complexas eliminadas do hardware
 - Otimização é fundamental
- O projeto de compiladores deve ser realizado juntamente com o projeto dos processadores
- Em suma, RISC viável graças ao avanço do software

Reduced Instruction Set Computer (RISC)

- Características comuns à maior parte dos processadores RISC
 - Número de instruções limitado
 - Codificação de instruções em uma palavra de tamanho fixo
 - Execução sem microcódigo
 - Altas taxas de execução (próximas a 1 instrução/ciclo)
 - Uso intenso de pipelines
 - Poucos modos de endereçamento
 - Operações envolvendo a memória principal restritas a transferências (load e store)
 - Operações lógicas e aritméticas entre registradores, tipicamente com instruções de três operandos

Reduced Instruction Set Computer (RISC)

- Vantagens?

Reduced Instruction Set Computer (RISC)

- Vantagens?
 - Processador **mais simples, mais barato** e **mais rápido**
- Menor número de circuitos internos permite clocks mais altos

Reduced Instruction Set Computer (RISC)

- Vantagens?
 - Processador **mais simples, mais barato e mais rápido**
- Menor número de circuitos internos permite clocks mais altos
- Qual é o tamanho do código gerado para uma máquina RISC em relação a uma CISC para um determinado programa?

Reduced Instruction Set Computer (RISC)

- Vantagens?
 - Processador **mais simples, mais barato e mais rápido**
- Menor número de circuitos internos permite clocks mais altos
- Qual é o tamanho do código gerado para uma máquina RISC em relação a uma CISC para um determinado programa?
 - Geralmente possui mais instruções (**por quê?**)
 - Potencial para executar de forma mais eficiente, pois possui instruções mais simples, com operandos inteiros (tempo de busca e execução de cada instrução é muito menor que o de uma instrução complexa)
 - Trechos de instruções mais simples em linguagem de máquina podem ser melhor otimizados pelo compilador (ao invés de instruções complexas que não podem ser decompostas)

Reduced Instruction Set Computer (RISC)

- Desvantagens?

Reduced Instruction Set Computer (RISC)

- Desvantagens?
 - Não são boas para cálculo em ponto flutuante sem a ajuda do hardware

Comparação entre RISC e CISC

RISC	CISC
Instruções levam 1 ciclo	Instruções levam múltiplos ciclos
Apenas load/store na memória	Qualquer instrução pode acessar a memória
Utiliza muito pipeline	Não tem pipeline ou tem pouco
Instruções com formato fixo	Instruções com formato variável
Poucas instruções e modos	Muitas instruções e modos
A complexidade está no compilador	A complexidade está no microprograma
Múltiplos conjuntos de registradores	Conjunto único de registradores

RISC e CISC Atualmente

- O que se encontra hoje em dia?

RISC e CISC Atualmente

- O que se encontra hoje em dia?
 - Processadores que misturam as duas filosofias (o que há de melhor em cada uma)
 - RISC: processadores ARM usados em smartphones, Playstation e Nintendo 64
- Como o Pentium operava?

RISC e CISC Atualmente

- O que se encontra hoje em dia?
 - Processadores que misturam as duas filosofias (o que há de melhor em cada uma)
 - RISC: processadores ARM usados em smartphones, Playstation e Nintendo 64
- Como o Pentium operava?
 - Sobre suas instruções internas, há um circuito decodificador que converte as instruções complexas em várias instruções simples entendidas pelo processador
 - E processadores mais recentes?

RISC e CISC Atualmente

- O que se encontra hoje em dia?
 - Processadores que misturam as duas filosofias (o que há de melhor em cada uma)
 - RISC: processadores ARM usados em smartphones, Playstation e Nintendo 64
- Como o Pentium operava?
 - Sobre suas instruções internas, há um circuito decodificador que converte as instruções complexas em várias instruções simples entendidas pelo processador
 - E processadores mais recentes?
- Por que a filosofia RISC não tomou conta do mercado?

RISC e CISC Atualmente

- O que se encontra hoje em dia?
 - Processadores que misturam as duas filosofias (o que há de melhor em cada uma)
 - RISC: processadores ARM usados em smartphones, Playstation e Nintendo 64
- Como o Pentium operava?
 - Sobre suas instruções internas, há um circuito decodificador que converte as instruções complexas em várias instruções simples entendidas pelo processador
 - E processadores mais recentes?
- Por que a filosofia RISC não tomou conta do mercado?
 - Apesar do RISC poder ser melhor tecnicamente que o CISC, os vastos recursos da Intel e o embalo do DOS e do Windows fizeram com que os x86 continuassem competitivos

- 1 O código executável de um processador CISC:
 - a é interpretado por microprogramas durante sua compilação, gerando microinstruções, que são interpretadas pelo hardware.
 - b é armazenado durante sua interpretação, gerando comandos na linguagem fonte, que são executados pelo hardware.
 - c é interpretado por microprogramas anteriormente à sua execução, gerando instruções compiláveis, que são executadas pelo software de execução.
 - d é interpretado por microprogramas durante sua execução, gerando microinstruções, que são executadas pelo hardware.
 - e gera microprogramas após sua execução, decorrentes de peculiaridades operacionais do hardware.

- 2 Qual técnica os processadores RISC utilizam para melhorar o desempenho? Justifique sua resposta.
- 3 Como você sugere minimizar o tempo gasto nas chamadas a microprogramas? Por quê?

Exercícios

- 4** Indique se as afirmativas são verdadeiras ou falsas, justificando a resposta quando forem falsas:
- a** () São características de arquiteturas RISC: mudança da complexidade do software para o hardware; diminuição do tamanho do código, em troca de um maior número de ciclos por instrução; modos de endereçamento simples que permitem somente que as funções LOAD e STORE acessem a memória, com todas as outras operações do tipo registro-registro.
 - b** () RISC e CISC são tecnologias que apresentam soluções semelhantes para problemas de aproveitamento da memória, um dos recursos de hardware mais escassos.
 - c** () O microprocessador Pentium 4 especificado possui tecnologia exclusivamente RISC, que, comparada à tecnologia CISC, utilizada principalmente em workstations, permite um número de instruções muito superior, o que o torna muito mais rápido e poderoso que os microprocessadores CISC.

- 5** Indique se as afirmativas são verdadeiras ou falsas, justificando a resposta quando forem falsas:
- a** () O projeto de um RISC não procura minimizar o tempo gasto nas chamadas a procedimentos, pois programas escritos para esse tipo de processador têm menos chamadas do que os escritos para um CISC.
 - b** () Uma técnica para simplificar o conjunto de instruções em um RISC é prover instruções com modos de endereçamento variados e complexos.
 - c** () Programas com um menor número de instruções necessariamente executarão mais rapidamente.

Exercícios

- 6** Indique se as afirmativas são verdadeiras ou falsas, justificando a resposta quando forem falsas:
- a** () A principal distinção entre as arquiteturas RISC e CISC reside no fato de que, na arquitetura RISC, as instruções são guardadas no próprio processador, o que facilita a programação, visto que ela já dispõe de todas as instruções necessárias para a execução dos programas, enquanto, na arquitetura CISC, apenas as instruções simples devem ser combinadas pelo programador para a realização de tarefas mais complexas.
 - b** () Atualmente, os processadores Intel contêm um núcleo RISC que executa as instruções mais simples que normalmente são as mais comuns em um único ciclo de caminho de dados, enquanto interpreta as instruções mais complexas no modo CISC.
 - c** () Os processadores de arquitetura CISC (complex instruction set computer) apresentam a vantagem de reduzir o tamanho do código executável, quando comparados a um processador de arquitetura RISC (reduced instruction set computer).

- 7** Indique se as afirmativas são verdadeiras ou falsas, justificando a resposta quando forem falsas:
- a** () A abordagem da arquitetura CISC (complex instruction set computer) procura minimizar o número de ciclos para que uma instrução seja executada, e aumenta, em contrapartida, o número de instruções por programa.
 - b** () A arquitetura RISC (reduced instruction set computer) busca reduzir o número de ciclos necessários para que uma instrução seja executada, sendo amplamente utilizada em processadores que têm por base o conjunto de instruções x86, desde as primeiras versões desses processadores.

Exercícios

- 8** A respeito das arquiteturas de computadores RISC e CISC, assinale a opção correta.
- a** Os computadores que implementam simultaneamente as arquiteturas RISC e CISC dispõem de processadores híbridos: um núcleo RISC executa instruções mais simples, enquanto instruções mais complexas são interpretadas na arquitetura CISC.
 - b** Na arquitetura RISC, é realizada mais de uma instrução em um ciclo de relógio.
 - c** A arquitetura CISC utiliza intensamente microcódigos que interpretam cada micro-operação de uma instrução.
 - d** Para melhoria de desempenho, a arquitetura CISC utiliza o princípio de paralelismo na execução de instrução, de forma a melhor explorar a técnica pipelining.
 - e** A abordagem RISC permite a simplificação de compiladores, uma vez que é mais simples gerar uma sequência de instruções de máquina a partir de instruções semelhantes a comandos de alto nível.

Exercícios

- 9** A respeito das arquiteturas de hardware RISC e CISC, assinale a opção correta.
- a** A unidade de medida mais eficiente e mais utilizada para a comparação dos processadores de ambas as arquiteturas é o MIPS (milhões de instruções por segundo).
 - b** As máquinas RISC executam instruções com maior rapidez do que as máquinas CISC, já que o faz por meio de subprogramas e não por meio da execução direta pelo hardware.
 - c** Diversamente do que ocorre na arquitetura RISC, na CISC as chamadas de funções ocorrem basicamente no processador, empregando-se um número menor de registradores, o que contribui para um aumento no desempenho total do processador.
 - d** Na arquitetura CISC, não se permite a utilização de muitos modos de endereçamento para realizar uma instrução que retorne o resultado da divisão de A por B.
 - e** O pipelining, execução de várias instruções simultaneamente no processador, é utilizado em máquinas RISC.

Exercícios

- 10** Assinale a opção correta relativa a características das arquiteturas RISC e CISC.
- a** CISC: instruções executadas pelo hardware. Instruções em diversos formatos. Instruções com poucos modos de endereçamento. Pouco uso da técnica de pipelining.
 - b** RISC: muitas instruções. Instruções executadas pelo hardware. Instruções com formato fixo. Instruções utilizam múltiplos ciclos. Arquitetura com poucos registradores.
 - c** RISC: poucas instruções. Instruções executadas pelo hardware. Instruções com diversos formatos. Arquitetura com poucos registradores. Pouco uso da técnica de pipelining.
 - d** CISC: instruções executadas por microcódigo. Instruções com formato fixo. Instruções com diversos modos de endereçamento. Arquitetura pipelining.
 - e** RISC: poucas instruções. Instruções executadas pelo hardware. Instruções com formato fixo. Instruções utilizam poucos ciclos de máquina. Arquitetura com muitos registradores.

- 11** Comparativamente com CISC, são características da arquitetura de computadores RISC ocupar
- a** menos espaço na memória e exigir programação mais fácil.
 - b** menos espaço na memória e exigir programação mais difícil.
 - c** mais espaço na memória e exigir programação mais difícil.
 - d** mais espaço na memória e exigir programação mais fácil.
 - e** menos espaço na memória e não influir no tipo de programação.