

## Lista Arq. De Comp. E Máquinas CISC / RISC

Henrique B G

### 1 - Discuta as principais características que compõem a definição da Arquitetura de um computador.

É definida pelo seu conjunto de instruções ISA (Instruction Set Architecture) definindo o limite suportado entre Hardware e Software, estudar Arquitetura de uma máquina é como estudar uma Linguagem de Programação de alto nível, só que em um nível mais baixo. Tem como descrição:

- **Operações (conjunto de instruções):**
  - Movimentação de Dados
  - Operações Aritméticas, Op. Lógicas e Deslocamentos
  - Saltos/Desvios e Chamada Procedimento
    - Teste, Repetições e Chamar Funções
  - Entrada e Saída
    - Comunicação entre CPU e mundo externo (periféricos) (out, in)
- **Tipo e Tamanho dos Operandos:**
  - Numéricos:
    - Inteiro (+ -), binário como decimal e Float
  - Não Numérico:
    - Texto, Booleano e Ponteiro
- **Armazenamento de Operandos:**
  - Acumulador:
    - 1 Registrador (memória para 1 palavra)
  - Pilha:
    - Abstração de Memória
  - Registradores de Propósito Geral:
    - Memórias dentro da CPU, operandos mencionados (registradores ou pos. na memória)
    - Reduz tráfego CPU-Memória e facilita a compilação
- **Endereçamento:**
  - Imediato: Acesso a Constante
  - Direto: // Variável na memória
  - Indireto: // Ponteiro em memória
  - Indexado: // Vetor (índice)
  - Registrador: // REG
  - REG Indireto: // Ponteiro dentro REG
  - Pilha: // Pilha
- **Formato de Instruções:**
  - Tamanho Instrução vs. Tamanho Palavra de Memória
    - Fixo – tamanho da palavra
    - Fixo – metade da palavra
    - Variável
  - Algumas precisam informar endereços: Jump – Load – Add

**2 - É muito comum encontrar processadores com a mesma arquitetura, mas que possuem organizações diferentes. Cite ao menos um exemplo e explique qual o objetivo principal desta estratégia.**

### **Arquitetura e Organização dos Processadores**

- ISA (Instruction Set Architecture): Define o conjunto de instruções que o processador pode executar, servindo como a interface entre hardware e software.
- Organização Interna: Mesmo processadores com a mesma ISA podem ter diferentes organizações para atender a diferentes objetivos como eficiência e custo.

### **Exemplo de Processadores com a Mesma Arquitetura, mas Organizações Diferentes**

- Intel e AMD: Ambas as empresas usam a arquitetura x86, mas organizam seus processadores de maneiras diferentes para otimizar desempenho e custo.

### **Objetivo 1: Aumentar a Eficiência e Desempenho**

- Intel: Atualiza a organização interna de seus processadores, mantendo a mesma ISA, para melhorar operações aritméticas, movimentação de dados e comunicação com periféricos, resultando em processadores mais rápidos.
- AMD: Organiza seus processadores de forma a otimizar o desempenho, criando alternativas competitivas em termos de performance, ainda que compatíveis com a mesma ISA da Intel.

### **Objetivo 2: Reduzir Custos**

- Linha Core da Intel (i3, i5, i7): Dentro da mesma família, diferentes modelos têm menos registradores e recursos limitados para reduzir custos. Isso resulta em variações de desempenho, mas mantém a compatibilidade com o mesmo conjunto de instruções.

## **3 – Descreva os Tipos de Dados/Operandos geralmente utilizados no nível de arquitetura.**

Arquiteturas de processadores suportam diversos tipos de operandos para atender às necessidades específicas das operações. Por exemplo, operações aritméticas usam operandos inteiros, enquanto operações lógicas lidam com valores binários (verdadeiro ou falso). Todos os dados são representados em binário, e

embora existam variações, há um conjunto básico de tipos de dados comum à maioria das arquiteturas.

### Tipos de Dados Numéricos

- **Inteiros:** Podem ser sem sinal (positivos apenas) ou com sinal (positivos e negativos). Utilizados em operações aritméticas.
- **BCD (Binário Codificado em Decimal):** Armazena cada dígito decimal em um byte separado, facilitando conversões entre decimal e binário.
- **Ponto-Flutuante:** Representa números reais com frações, permitindo a manipulação de uma ampla gama de valores.

### Tipos de Dados Não Numéricos

- **Texto:** Codificado em ASCII (7 bits) para textos básicos ou Unicode (até 32 bits) para suportar uma ampla gama de caracteres.
- **Valor Booleano:** Representa verdadeiro (true) ou falso (false), essencial para decisões lógicas.
- **Ponteiro:** Endereços de memória que ajudam a acessar e manipular dados armazenados na memória.

## 4 - Compare conceitualmente as abordagens de arquitetura de computadores RISC e CISC. Apresente exemplos de equipamentos que adotam cada abordagem.

Na definição da arquitetura de uma máquina, existem três principais caminhos. Primeiro, pode-se optar por utilizar uma arquitetura já existente, como a AMD fez ao adotar a arquitetura x86 da Intel. Outra abordagem é projetar uma arquitetura totalmente nova, que pode seguir um dos dois modelos principais: CISC (Complex Instruction Set Computer) ou RISC (Reduced Instruction Set Computer). A seguir, vamos comparar essas duas abordagens.

### Abordagem CISC (Complex Instruction Set Computer)

- **Definição:** CISC destaca-se pelo "C" de Complexo.
- **Características:**
  - **Conjunto Extenso de Instruções:** Oferece muitas instruções, cada uma podendo realizar operações complexas.
  - **Instruções Complexas:** Instruções podem realizar múltiplas operações em um único comando.
  - **Poucos Registradores:** Menor número de registradores, com uma ênfase maior em operações que utilizam diretamente a memória.
- **Exemplos de Uso:** A arquitetura CISC é amplamente utilizada em PCs, com exemplos notáveis sendo as arquiteturas x86 e x486 da Intel.

## Abordagem RISC (Reduced Instruction Set Computer)

- **Definição:** RISC é conhecido pelo "R" de Reduzido.
- **Características:**
  - **Conjunto Pequeno de Instruções:** Mantém um número reduzido de instruções, cada uma projetada para ser executada rapidamente.
  - **Instruções Simples:** Cada instrução realiza uma única operação simples.
  - **Muitos Registradores:** Utiliza um maior número de registradores para armazenar dados temporários e minimizar o acesso à memória.
- **Exemplos de Uso:** A arquitetura RISC é comum em sistemas embarcados, como videogames e celulares, onde a eficiência e a simplicidade são cruciais.

## 5 - Compare as abordagens CISC e RISC com relação ao Controle. Apresente as vantagens e desvantagens de cada abordagem.

As arquiteturas CISC (Complex Instruction Set Computer) e RISC (Reduced Instruction Set Computer) oferecem diferentes complexidades em seus conjuntos de instruções, exigindo abordagens distintas tanto no desenvolvimento do compilador quanto na implementação do controle. O compilador deve traduzir linguagens de programação (LP) para linguagem de máquina (LM), enquanto o decodificador/controlador converte LM em sinais de hardware. A seguir, comparamos as abordagens de controle para CISC e RISC.

### Controle em Arquiteturas CISC

- **Controle Microprogramado:**
  - **Descrição:** As instruções CISC são complexas e não podem ser executadas diretamente em hardware. Em vez disso, há uma memória na CPU que contém microcódigo, um conjunto de microoperações necessárias para executar cada instrução.
  - **Funcionamento:** Cada instrução é traduzida em várias microoperações durante a execução. Essas microoperações realizam operações simples diretamente em hardware.
  - **Exemplo:** É como se houvesse uma sub-rotina para cada instrução dentro da CPU.
  - **Desvantagens:**
    - **Velocidade:** A execução é mais lenta porque é necessário ler a memória de microinstruções durante a execução.
    - **Complexidade:** A microprogramação pode aumentar a complexidade do controle e o custo de implementação do hardware.

## Controle em Arquiteturas RISC

- **Controle Direto em Hardware:**
  - **Descrição:** Em arquiteturas RISC, as instruções são simples o suficiente para serem executadas diretamente em hardware. Isso elimina a necessidade de microprogramação.
  - **Funcionamento:** As instruções são decodificadas e executadas diretamente, permitindo uma execução mais rápida.
  - **Vantagens:**
    - **Velocidade:** A execução é mais rápida devido à simplicidade das instruções e ao controle direto em hardware.
    - **Eficiência:** A abordagem remove a complexidade de tradução do hardware, delegando a complexidade ao compilador, que pode otimizar a tradução em tempo de compilação.
  - **Desvantagens:**
    - **Desenvolvimento de Compiladores:** A complexidade do compilador aumenta, pois ele deve otimizar a tradução de operações complexas para uma série de instruções simples.

## 6 – Descreva o conceito de Arquitetura Híbrida, incluindo por que utilizar tal conceito e qual o princípio do seu funcionamento.

CISC e RISC têm abordagens distintas: CISC com um vasto conjunto de instruções e RISC com instruções simples e rápidas. Arquiteturas híbridas combinam os pontos fortes de ambas, como exemplificado pela arquitetura x64 da Intel.

### Princípio de Funcionamento das Arquiteturas Híbridas

**Modelo de Operação:** Processadores híbridos utilizam ISA CISC para compatibilidade com software legado, mas processam instruções simples como em RISC. Um Circuito Decodificador interno traduz instruções complexas em várias simples para uma execução eficiente.

### Benefícios:

- **Compatibilidade e Desempenho:** Mantém compatibilidade com software antigo e melhora o desempenho com instruções simplificadas.
- **Eficiência do Hardware:** Reduz a complexidade do hardware ao evitar microprogramação, acelerando a execução.

### Compiladores e Tradução:

- **Otimização em Tempo de Compilação:** Compiladores modernos convertem instruções complexas em simples antes da execução, maximizando a eficiência da arquitetura híbrida.

## Arquiteturas Híbridas

### Exemplos:

- **Intel Pentium Pro e Família x86:** Usam ISA CISC, mas internamente processam instruções complexas como simples.
- **MIPS R10000 e HP PA-8000:** Misturam características de CISC e RISC, oferecendo flexibilidade e eficiência.