

Lista Tipos e Formatos de Instruções

Henrique B G

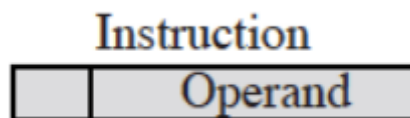
1 – Fale sobre os Modos de Endereçamento empregados em arquiteturas de computadores, e explique ao menos três exemplos apresentados em aula.

Modos de Endereçamento

- **Importância:** Determinam como dados são acessados, podendo estar em registradores, memória ou pilha.
- **Variedade:** Máquinas CISC oferecem vários modos de endereçamento.

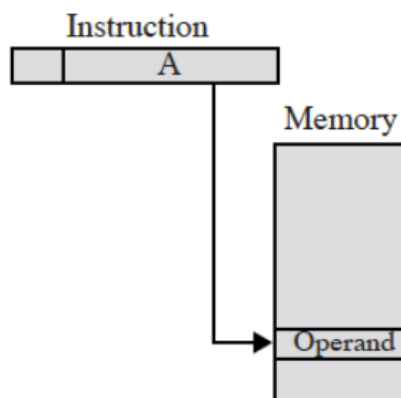
Modo de Endereçamento Imediato

- **Operando:** Valor próprio incluído na instrução.
- **Exemplo:** LOAD R1 10 (carrega o valor 10 em R1).
- **Vantagem:** Acesso rápido, pois o valor está na instrução.
- **Desvantagem:** Valor fixo, sem possibilidade de alteração sem recompilar o código.



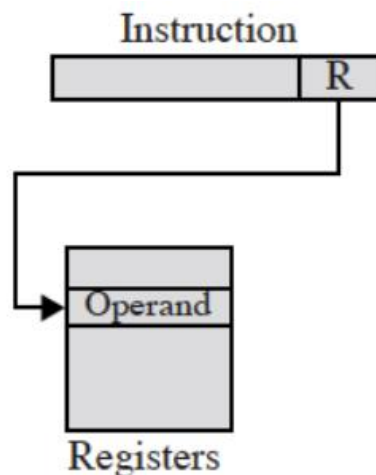
Modo de Endereçamento Direto

- **Operando:** Endereço fixo na memória.
- **Exemplo:** LOAD R1 A (acessa o valor no endereço A e carrega em R1).
- **Vantagem:** Valor pode ser alterado ao modificar a memória.
- **Desvantagem:** Sempre acessa a mesma posição de memória; alteração do endereço requer edição do código.



Modo de Endereçamento por Registrador

- **Operando:** Registrador com valor armazenado.
- **Exemplo:** LOAD R1 R (carrega o valor de R para R1).
- **Vantagem:** Acesso rápido, pois usa o banco de registradores.
- **Desvantagem:** Espaço de endereçamento limitado pela quantidade de registradores.



2 – Descreva a relação entre o Tamanho das Instruções e o Tamanho da Palavra. Apresente os prós e contras de cada possibilidade apresentada em aula.

O tamanho da palavra de memória em um sistema nem sempre coincide com o tamanho das instruções da arquitetura. Essa diferença requer estratégias para o mapeamento eficiente das instruções na memória. A seguir, apresentamos três alternativas para lidar com essa questão.

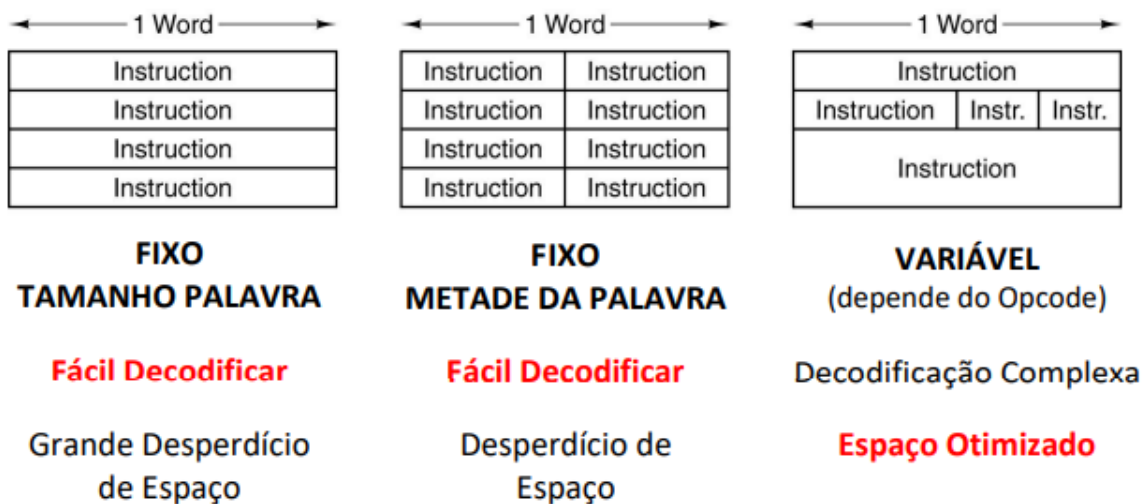
Instruções de Tamanho Fixo

- **Definição:** As instruções têm um tamanho constante, como é comum em arquiteturas RISC.
- **Alternativas:**
 - **Tamanho Fixo = Tamanho da Palavra:** Cada instrução ocupa toda a largura da palavra de memória.
 - **Tamanho Fixo = Metade do Tamanho da Palavra:** Cada instrução ocupa metade da largura da palavra de memória.
- **Vantagens:**
 - **Facilidade de Decodificação:** O padrão de decodificação é simples e uniforme, pois cada campo da instrução sempre começa e termina no mesmo ponto.
 - **Simplicidade de Implementação:** O design do processador é mais simples, pois o tamanho das instruções é fixo.
- **Desvantagens:**

- **Desperdício de Memória:** Mesmo que uma instrução não precise de toda a largura alocada, ela ocupa o espaço total reservado, resultando em desperdício.
- **Exceções:** Instruções maiores podem exigir múltiplas palavras de memória.

Instruções de Tamanho Variável

- **Definição:** As instruções variam em tamanho, comum em arquiteturas CISC.
- **Funcionamento:** O número de bits para a instrução é variável, dependendo do opcode e da complexidade da operação.
- **Vantagens:**
 - **Eficiência de Memória:** As instruções ocupam apenas o número de bits necessário, evitando desperdício de memória.
 - **Flexibilidade:** Permite instruções mais complexas sem restringir a arquitetura a um tamanho fixo de instrução.
- **Desvantagens:**
 - **Complexidade de Decodificação:** Decodificar instruções é mais difícil, pois o início e o fim da instrução podem variar, complicando a identificação dos campos.
 - **Desempenho:** A complexidade adicional na decodificação pode impactar a performance do processador.



3 – Compare as abordagens CISC e RISC com relação ao Conjunto de Instruções disponibilizadas. Apresente as vantagens e desvantagens de cada abordagem (aprendizado x programação).

O tamanho do conjunto de instruções afeta o aprendizado e a programação. Arquiteturas CISC e RISC oferecem abordagens distintas para o conjunto de instruções, impactando a curva de aprendizado e a flexibilidade na programação.

Conjunto de Instruções CISC

- **Características:**
 - Conjunto extenso de instruções.
 - Instruções podem realizar múltiplas operações.
- **Vantagens:**
 - Maior flexibilidade na programação.
 - Código potencialmente mais compacto.
- **Desvantagens:**
 - Aprendizado mais difícil devido ao grande número de instruções.
 - Hardware mais complexo e caro.

Conjunto de Instruções RISC

- **Características:**
 - Conjunto reduzido de instruções.
 - Instruções simples, geralmente um ciclo de clock.
- **Vantagens:**
 - Aprendizado mais fácil com menos instruções.
 - Hardware mais simples e eficiente.
- **Desvantagens:**
 - Menos flexibilidade, pode exigir mais código.
 - Código pode ser maior devido à simplicidade das instruções.

4 – Compare as abordagens CISC e RISC com relação a Complexidade das Instruções disponibilizadas. Apresente as vantagens e desvantagens de cada abordagem (compilador x controle).

A complexidade das instruções afeta a complexidade da tradução feita pelo compilador e a decodificação durante a execução.

- **CISC:** Amiga do compilador (tradução 1-para-1, fácil para o compilador, mas complexa para o controle).
- **RISC:** Amiga do decodificador (tradução 1-para-N, fácil para o controle, mas exige um compilador mais inteligente).
- Arquiteturas CISC simplificam a tradução de alto nível para código de máquina, enquanto arquiteturas RISC facilitam a execução com um design de hardware mais simples.
- **Compilação:** Tradução de código de alto nível (LP) para código de máquina (LM) antes da execução.
- **Decodificação:** Tradução de código de máquina (LM) para sinais de hardware (HW) durante a execução.

Complexidade de Instruções CISC

- **Características:**
 - Instruções complexas e multifuncionais.
 - Semelhança com operações de alto nível.
- **Vantagens:**
 - **Facilita o Compilador:** Tradução direta de instruções de alto nível para as instruções CISC (1-para-1).
 - **Código mais compacto:** Menos instruções necessárias para realizar operações complexas.
- **Desvantagens:**
 - **Decodificação Complexa:** Instruções complexas exigem hardware sofisticado para decodificação.
 - **Controle Microprogramado:** Necessidade de microprogramas para interpretar e executar instruções.

Complexidade de Instruções RISC

- **Características:**
 - Instruções simples e uniformes.
 - Executadas em um ciclo de clock.
- **Vantagens:**
 - **Decodificação Simples:** Instruções simples facilitam a implementação direta em hardware.
 - **Controle Eficiente:** O design do hardware é mais direto e eficiente.
- **Desvantagens:**
 - **Compilador Inteligente:** Tradução de instruções complexas de alto nível para várias instruções simples (1-para-N).
 - **Código potencialmente maior:** Mais instruções podem ser necessárias para realizar a mesma tarefa.

5 – Compare as abordagens CISC e RISC com relação ao Número de Registradores disponibilizados. Apresente as vantagens e desvantagens de cada abordagem (armazenamento x modos de endereçamento).

O número de registradores disponíveis em uma arquitetura afeta diretamente o armazenamento interno da CPU e os modos de endereçamento necessários. Arquiteturas com menos registradores tendem a precisar de mais modos de endereçamento e, portanto, instruções mais complexas. Já arquiteturas com mais registradores podem simplificar os modos de endereçamento e reduzir a necessidade de acessos à memória.

- **CISC:** Menos registradores, mais modos de endereçamento, instruções mais complexas e maior dependência de acesso à memória.

- **RISC:** Mais registradores, menos modos de endereçamento, instruções mais simples e menor dependência de acesso à memória.

Número de Registradores nas CISC

- **Características:**
 - Poucos registradores (4 a 16).
- **Impactos:**
 - **Armazenamento Interno:** Menor capacidade de armazenamento interno.
 - **Acesso à Memória:** Maior frequência de acesso à memória, o que pode ser mais lento.
 - **Modos de Endereçamento:** Necessidade de suportar vários modos de endereçamento para compensar a limitação no número de registradores.
 - **Complexidade das Instruções:** Instruções mais complexas devido à necessidade de implementar diferentes versões para cada modo de endereçamento.

Número de Registradores nas RISC

- **Características:**
 - Muitos registradores (32 até centenas).
- **Impactos:**
 - **Armazenamento Interno:** Maior capacidade de armazenamento interno, permitindo guardar mais variáveis e constantes em registradores.
 - **Redução de Acessos à Memória:** Menos necessidade de acessar a memória, resultando em maior velocidade.
 - **Simplicidade dos Modos de Endereçamento:** Menos modos de endereçamento são necessários, pois a arquitetura privilegia o mapeamento por registrador.
 - **Formato das Instruções:** Formato mais simples devido à redução na complexidade dos modos de endereçamento.

6 – Apresente os tipos mais comuns de instruções encontradas em arquiteturas de computadores. Para isso, baseie-se nas instruções da arquitetura MIPS.

As arquiteturas de computadores, como a MIPS, utilizam diferentes tipos básicos de instruções, que incluem transferências de dados, cálculos e desvios. Embora a maioria adote esses tipos básicos, a complexidade e a variedade das instruções podem variar entre arquiteturas. Cada tipo de instrução desempenha um papel específico na manipulação e controle de dados e na execução de programas.

Instruções para Transferência de Dados

- **Função:**
 - Movimentam dados entre a memória principal e os registradores, em ambos os sentidos.
- **Exemplos:**
 - **LOAD:** Copia um dado de um endereço de memória para um registrador. Ex.: LW (Load Word).
 - **STORE:** Copia um dado de um registrador para a memória. Ex.: SW (Store Word).
- **Observação:**
 - Em arquiteturas RISC, os dados devem ser copiados para dentro do banco de registradores antes de serem utilizados.

Instruções que Realizam Cálculos

- **Instruções Aritméticas:**
 - Manipulam valores inteiros ou reais.
 - **Exemplos:** ADD (adição), SUB (subtração), ADDI (adição imediata).
- **Instruções Lógicas:**
 - Executam operações lógicas sobre valores binários.
 - **Exemplos:** AND (e lógico), OR (ou lógico), NOT (negação).
- **Instruções de Deslocamento:**
 - Realizam deslocamento de bits, úteis para multiplicação ou divisão por 2.
 - **Exemplos:** SHIFTL (deslocamento à esquerda), SHIFTR (deslocamento à direita).

Instruções que Realizam Desvios

- **Função:**
 - Alteram o fluxo de execução de um programa modificando o Contador de Programa (PC).
- **Instruções de Desvio Condicional:**
 - Saltos no código baseados no resultado de um teste.
 - **Exemplo:** BEQ (Branch if Equal).
- **Instruções de Desvio Incondicional:**
 - Sempre realizam um salto sem condição.
 - **Exemplo:** JUMP (salto incondicional).

7 – Apresente os 3 formatos de instruções disponíveis na arquitetura MIPS.

Cada arquitetura de computador define seus próprios formatos de instrução. Arquiteturas CISC tendem a ter conjuntos de instruções complexas, com vários

campos e tamanhos variáveis. Em contraste, arquiteturas RISC, como a MIPS, optam por formatos de instruções mais simples e uniformes. Na arquitetura MIPS, as instruções são organizadas em três formatos principais, cada um com uma estrutura específica para facilitar a execução e a decodificação.

Formatos de Instrução na Arquitetura MIPS

Máquinas que seguem o padrão MIPS utilizam três formatos de instrução distintos. Veja a seguir uma breve explicação de cada formato:

- **Descrição:** Utilizado para operações que envolvem apenas registradores.
- **Exemplos:** Instruções aritméticas (ADD, SUB) e lógicas (AND, OR).
- **Estrutura:** Possui campos para códigos de operação, registradores fonte e destino, e código da função a ser executada pela Unidade Lógica e Aritmética (ULA).

1. Formato Tipo I

- **Descrição:** Utilizado para instruções que precisam de um valor imediato, como operações com constantes e movimentação de dados.
- **Exemplos:** Instruções de carregamento (LOAD) e armazenamento (STORE).
- **Estrutura:** Inclui campos para código de operação, registradores, e um campo de valor imediato (constante ou endereço).

2. Formato Tipo J

- **Descrição:** Usado para operações de desvio, como saltos.
- **Exemplos:** Instruções de salto (JUMP).
- **Estrutura:** Inclui campos para código de operação e um campo para o endereço de destino do salto.

OPCODE	rs	rt	rd	sa	funct	R format
OPCODE	rs	rt	immediate			I format
OPCODE	jump target					J format

Principais Campos dos Formatos de Instruções

Os principais campos dos formatos de instrução MIPS são:

- **OP (Código da Operação):** Identifica a operação a ser executada.
- **RS (Registrador Fonte):** Endereço do registrador fonte.
- **RT (Registrador Destino):** Endereço do registrador destino.
- **RD (Registrador de Destino):** Endereço do registrador onde o resultado é armazenado (no formato Tipo R).

- **CONST/END (Valor Constante/Endereço):** Valor constante ou endereço usado na operação (no formato Tipo I).
- **SHAMT (Número de Deslocamento):** Número de bits a serem deslocados (para operações de deslocamento).
- **FUNCT (Código da Função):** Código adicional que especifica a operação exata a ser executada pela ULA (no formato Tipo R).

A decodificação é o processo em que a CPU interpreta cada instrução de acordo com seu formato, identificando o conteúdo dos campos e gerando os sinais de controle apropriados para a execução da instrução.

8 – Explique como o formato e o tamanho das instruções de uma arquitetura podem ser definidos.

Cada instrução em linguagem de máquina é composta por bits divididos em campos, como o código da operação (opcode), operandos e endereços de resultado. O formato e o tamanho das instruções variam conforme a arquitetura do processador e o tamanho da palavra.

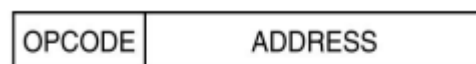
Número de Campos de Operando

Instruções diferentes requerem diferentes números de operandos:

- **JUMP:** 1 operando (endereço de salto).
- **LOAD:** 2 operandos (endereço de origem e destino).
- **ADD:** 3 operandos (2 endereços de operandos e 1 endereço de destino).



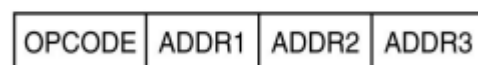
SEM ENDEREÇO



COM 1 ENDEREÇO



COM 2 ENDEREÇOS



COM 3 ENDEREÇOS

Definição do Tamanho dos Campos

- **Opcode:** Determina o número de bits para codificar as instruções (ex.: 4 bits para 16 instruções).
- **Endereçamento:** Baseado no tamanho da memória (ex.: 30 bits para 1 GB de memória).
- **Registradores:** Número de bits para endereçar registradores (ex.: 5 bits para 32 registradores).

O tamanho e formato das instruções são definidos para atender às necessidades específicas de cada arquitetura.