

Revisão Prova 1 de Organização de Computadores Luciano L. Caimi

Erickson Giesel Müller

Conteúdos

1. Arquitetura Von Neumann
2. Arquitetura Harvard
3. Arquitetura RISC
4. Arquitetura CISC
5. Pipeline
6. Hierarquia de memória
 - (a) Cache
 - (b) Memória Principal RAM
 - (c) Memória Virtual
 - (d) Armazenamento de Massa

1 Introdução

Ao se descreverem computadores, é comum se fazer uma distinção entre *arquitetura* e *organização* de computadores. A arquitetura se refere aos atributos de um sistema **visíveis ao programador**, em outras palavras, são os atributos que possuem um impacto direto sobre a execução lógica de um programa, representada pela arquitetura de conjunto de instrução (ISA). a organização de computadores refere-se às unidades operacionais e suas interconexões que percebam as especificações de arquitetura, como sinais de controle, interface entre o computador e periféricos e a tecnologia de memória utilizada.

2 Arquitetura Multinível

O Sistema Operacional é a camada mais baixa na parte de softwares de um computador, pois traduz as linguagens para a linguagem de máquina.

3 Arquitetura CISC

A arquitetura CISC apresenta instruções complexas que demandam um **grande número de ciclos** para serem executadas. Dezenas de modos de endereçamento, instruções de tamanhos variados e referência a operandos na memória principal. Nas arquiteturas CISC fica mais difícil implementar o pipeline. A unidade de controle é microprogramada. Menos instruções por ciclo.

4 Arquitetura RISC

Instruções mais simples, demandando um número **fixo** de ciclos de máquina para a sua execução. Uso de poucos e simples modos de endereçamento. Poucos formatos de instruções.

Apenas instruções de load/store referenciam operandos na memória principal. Cada fase de processamento da instrução tem a duração fixa igual a um ciclo de máquina. O formato fixo das instruções facilita o pipeline. As instruções são executadas, na sua maioria, em apenas um ciclo de máquina.

A unidade de controle é em geral hardwired, ou seja, não há microprograma para interpretar as instruções. Arquitetura orientada a registrador, todas as operações aritméticas são realizadas entre registradores.

A desvantagem de usar uma arquitetura com um único ciclo é que o ciclo do clock se torna ineficiente, pois este deve ser do tamanho da instrução mais lenta. Isso é um grande problema quando se considera instruções mais complexas como por exemplo multiplicação de ponto flutuante.

5 Pipeline

6 Memória Cache

6.1 Cache com Mapeamento Direto

Para calcular a linha da memória cache:

$$i = j \mod m$$

onde:

i é o número da linha na memória cache

j é o número de blocos na memória RAM

m é o número de blocos na memória cache

Todas as relações de endereço são em potência de 2. Assim, conseguimos encontrar a linha na memória cache, mas, em uma operação de busca, ainda precisamos saber em qual bloco está a informação, para isso usamos duas informações de controle:

6.2 Bit de Válido

Indica se a entrada do cache está escrita com alguma informação.

6.3 Bit de Tag

Complementa o cálculo de endereço com a sintaxe $[tag]j \bmod m$. Os bits de tag informam em qual parte da memória cache está a informação de acordo com a linha calculada.

6.4 Melhorias na Tecnologia de Semicondutores

6.5 Lei de Moore

A cada nó tecnológico, a largura do canal do transistor diminui, mantendo o mesmo custo de fabricação, tornando o processador mais eficiente por área. Portanto, o tamanho do transistor diminui, a quantidade de transistores aumentam e a frequência do clock acelera.

Devido a isso, surgiu o problema do *powerwall*, os processadores começaram a esquentar mais e precisou surgir métodos de dissipar a frequência do processador.

7 Arquitetura de Von Neumann

Von Neumann apresenta **4 proposições**:

- Unidades Funcionais
 - CPU - Unidade de Controle + ULA + Registradores
 - Memória Principal - Instruções e Dados
 - Dispositivo de Entrada e Saída
- Aritmética Binária
- Ciclo de Instrução Repetitivo
- Programa Armazenado em Memória

7.1 Ciclo de Instrução

1. Buscar a instrução na Memória Principal (FI - *fetch instruction*)
2. Decodificar a Instrução (DI - *decode instruction*)
3. Calcular os Operandos (CO)
4. Buscar Operandos da Instrução (FO - *fetch operands*)
5. Executar Instrução (EI)

6. Armazena o Resultado (WO - *write operands*)

Gargalo da Arquitetura de Von Neumann: Como a arquitetura apresenta uma memória que armazena tanto os dados quanto as instruções. O ciclo de instrução não pode fazer o FI e o FO ao mesmo tempo.

Para solucionar o gargalo, os dados e as instruções saem da memória principal para o barramento de dados e vão para a cache L1 seguindo a seguinte ordem:

$$MP \rightarrow L3 \rightarrow L2 \rightarrow L1$$

8 Arquitetura do Conjunto de Instruções

9 Arquitetura do Tipo Pilha

Diferente das outras arquiteturas, ao retirar um valor dos registradores e encaminhar para a ULA (pop), os valores do topo da pilha são removidos. Em outras arquiteturas, uma operação *mv rd, rs* copia o dado do registrador de origem para o registrador de destino (*addi rd, rs, 0*).

10 Chamadas de Sistema (ecall)

11 Chamadas de Função (call/jal)

12 Manipulação de Vetores e Matrizes