Lógica digital e organização de computadores

Arquiteturas RISC

Campus de Alegre

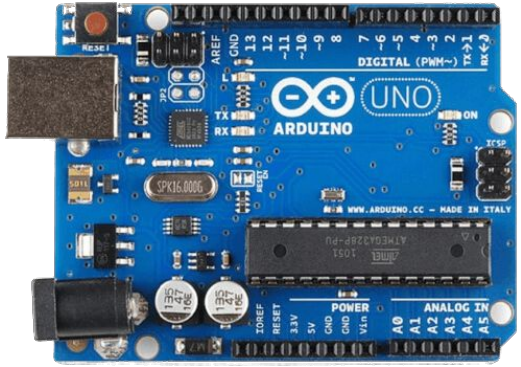
O que é a arquitetura RISC?

- Definição: RISC = Reduced Instruction Set Computer (Computador com Conjunto Reduzido de Instruções).
- Filosofia: Projetar um processador com um conjunto de instruções pequeno, simples e altamente otimizado.
- Objetivo: Aumentar a velocidade de execução, permitindo que a maioria das instruções seja completada em um único ciclo de relógio.

Motivação

- Análises de programas (décadas de 70/80) mostraram que a maioria dos programas usava um pequeno subconjunto das instruções complexas disponíveis.
- Problema CISC: Instruções complexas e de tamanho variável dificultavam a otimização
- Ideia Central RISC: Simplificar o conjunto de instruções para otimizar a execução das operações mais frequentes.

Dispositivos que utilizam



Princípios do RISC

Conjunto Reduzido de Instruções

- Característica: Número significativamente menor de instruções em comparação com abordagens anteriores.
- Foco: Incluir apenas as instruções mais frequentemente utilizadas pelos compiladores.
- Complexidade: Operações complexas são realizadas por combinação de instruções simples (responsabilidade do compilador).
- Benefício: Simplifica a lógica da Unidade de Controle e a decodificação.

Instruções de Tamanho Fixo

- Característica: Todas as instruções têm o mesmo número de bits (ex: 32 bits).
- Benefício:
  - Simplifica a busca (Fetch): A UCP sabe exatamente quantos bytes buscar da memória para obter a próxima instrução.
  - Simplifica a Decodificação: A posição dos campos (C.Op., operandos) é fixa, agilizando a interpretação.
  - Facilita o Pipelining: O processamento das instruções pode ser sobreposto de forma mais eficiente.

Arquitetura Carga/Armazena (Load/Store)

- Característica: As únicas instruções que acessam a memória principal (MP) são as de carga (Load) e armazena (Store).
- Operações Aritméticas/Lógicas: Todas as outras operações (ADD, SUB, AND, OR) ocorrem exclusivamente entre registradores internos da UCP.
- Fluxo: Dados são carregados da MP para registradores -> Operações são feitas entre registradores -> Resultados são armazenados de volta na MP (se necessário).
- Benefício: Reduz o número de acessos lentos à memória.

Grande Conjunto de Registradores

- Característica: Processadores RISC possuem um número elevado de registradores de uso geral (ex: 32, 64 ou mais).
- Objetivo: Minimizar acessos à memória, mantendo o máximo de operandos e variáveis temporárias diretamente na UCP.
- Compilador: O compilador é otimizado para alocar variáveis nos registradores de forma eficiente.
- Benefício: Acesso a registradores é ordens de magnitude mais rápido que acesso à MP.

Pipelining Otimizado

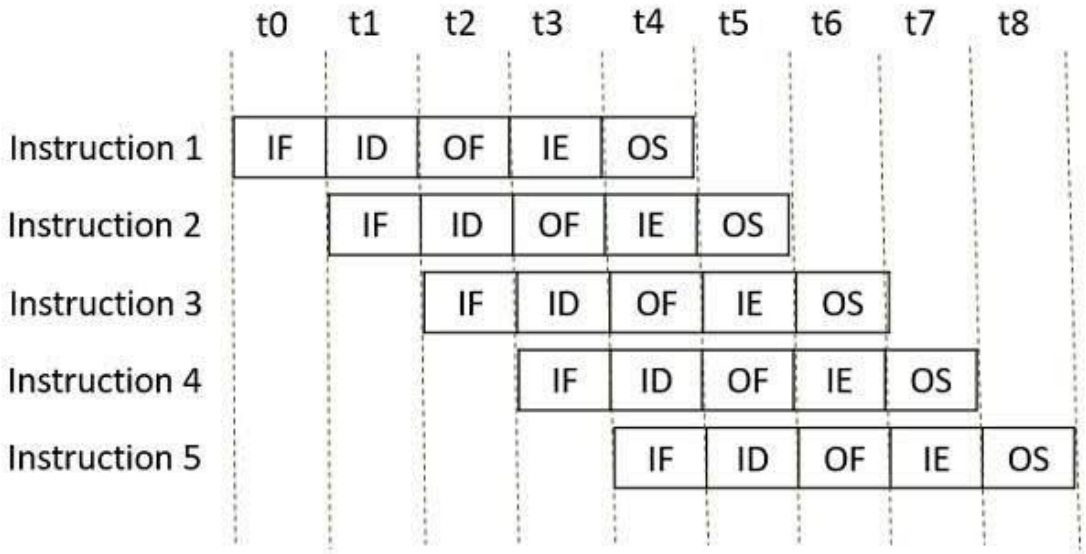
- Conceito: Técnica de execução sobreposta, onde múltiplas instruções estão em diferentes estágios de execução simultaneamente (como uma linha de montagem).
- Favorecido por RISC:
  - Instruções simples e de tamanho fixo facilitam a divisão em estágios uniformes.
  - Arquitetura Load/Store simplifica os estágios de acesso à memória.
- Objetivo: Alcançar a execução de (idealmente) uma instrução por ciclo de relógio.

Pipelining - Estágios Clássicos

- Exemplo de Pipeline de 5 Estágios:

Etapa	Nome da Etapa	Descrição da Função
1. IF	**Instruction Fetch** (Busca da Instrução)	Busca a instrução na Memória Principal ou Cache.
2. ID	**Instruction Decode** (Decodificação)	A Unidade de Controle (UC) interpreta o Código de Operação (C.Op.).
3. OF	**Operand Fetch** (Busca de Operandos)	Localiza e recupera os dados (operandos), geralmente dos Registradores.
4. IE	**Instruction Execute** (Execução)	A UAL ou FPU realiza o cálculo ou a operação lógica.
5. OS	**Output Store** (Armazenamento)	Armazena o resultado no Registrador de Destino ou na Memória Principal.

Pipelining



Pipelining of 5 Instructions

Execução em Um Ciclo de Relógio (Ideal)

- Objetivo: Graças à simplicidade das instruções e ao pipelining, a taxa de transferência (throughput) do processador aproxima-se de uma instrução completada por ciclo de relógio.
- Realidade: Desvios condicionais e dependências de dados podem causar "bolhas" no pipeline, reduzindo a taxa ideal, mas ainda assim é muito mais rápido que execuções não-pipelined de instruções complexas.

Menor Quantidade de Modos de Endereçamento

- Característica: RISC utiliza poucos e simples modos de endereçamento.
- Comuns: Acesso a registrador, imediato (limitado), e deslocamento a partir de um registrador (para Load/Store).
- Eliminados: Modos complexos como indireto com pós-incremento, etc.
- Benefício: Simplifica a lógica de cálculo de endereço e a decodificação da instrução.

O Papel do Compilador em RISC

- Importância Crucial: Grande parte da complexidade é transferida do hardware para o compilador.
- Responsabilidades do Compilador:
  - Otimização: Gerar sequências eficientes de instruções simples para realizar operações complexas.
  - Alocação de Registradores: Gerenciar o grande banco de registradores.
  - Reordenamento de Instruções: Minimizar bolhas no pipeline causadas por dependências.

Vantagens da Abordagem RISC

- Velocidade: Execução mais rápida devido a instruções simples e pipelining eficaz.
- Simplicidade do Hardware: UCP com menos transistores, mais fácil de projetar, fabricar e testar.
- Menor Consumo de Energia: Hardware mais simples geralmente consome menos energia.
- Tempo de Projeto: Ciclos de projeto potencialmente mais curtos.

Desvantagens Potenciais do RISC

- Densidade de Código: Programas em código de máquina RISC podem ser maiores (mais instruções simples para fazer o mesmo que uma complexa).
- Dependência do Compilador: O desempenho depende fortemente da qualidade do compilador otimizador.
- (Nota: Essas desvantagens foram mais pronunciadas nas primeiras gerações e hoje são mitigadas por técnicas avançadas).

RISC Hoje

- Dominância: Os princípios RISC são a base da maioria dos processadores modernos, especialmente em dispositivos móveis e embarcados (ARM).
- Hibridismo: Mesmo processadores tradicionalmente CISC (como x86 da Intel/AMD) utilizam internamente um núcleo tipo RISC (micro-operações) para traduzir e executar as instruções CISC complexas de forma otimizada com pipelining.
- Legado: A filosofia de simplificar para acelerar provou ser extremamente eficaz.