

Organização e Arquitetura de Computadores

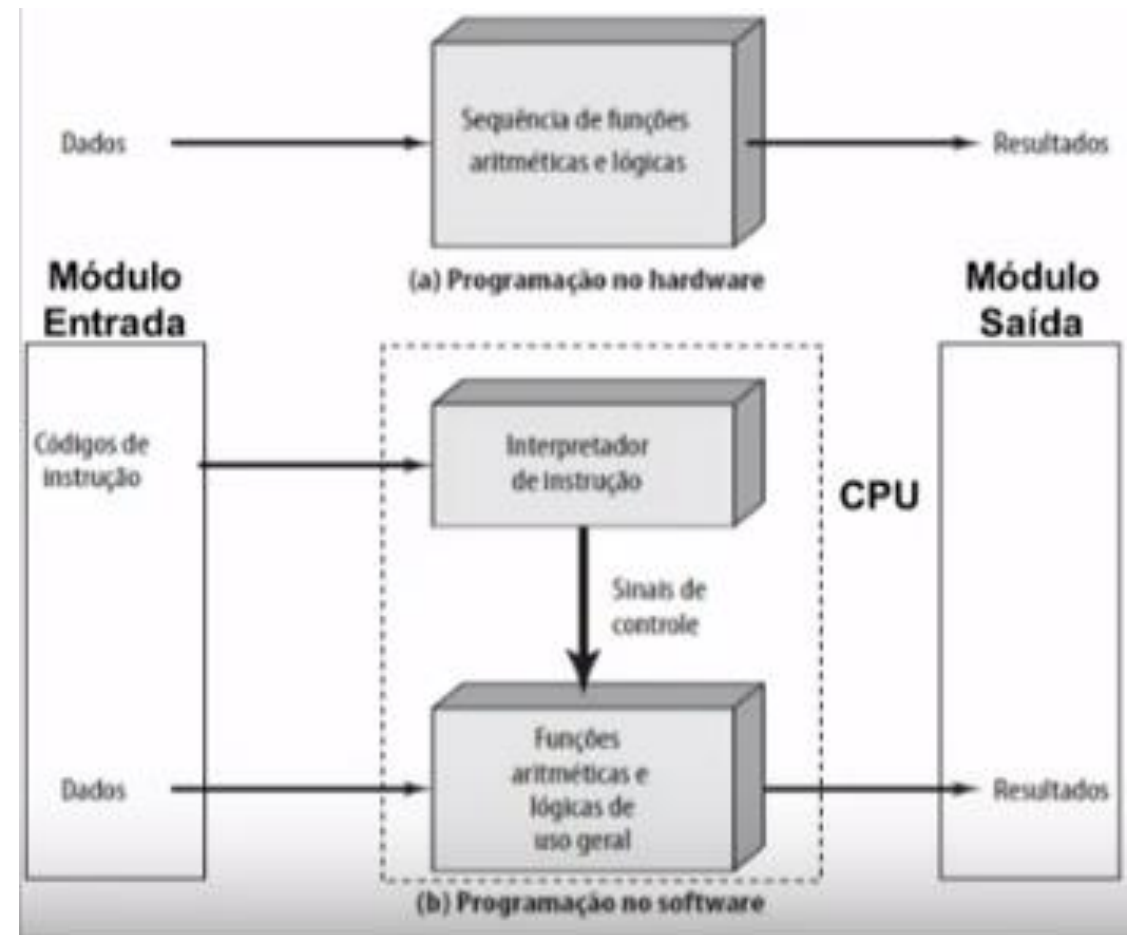
Conjunto de Instruções.

Visão de alto nível

- Abordagem de programação em

- Hardware
(programa hardwired)

- Software

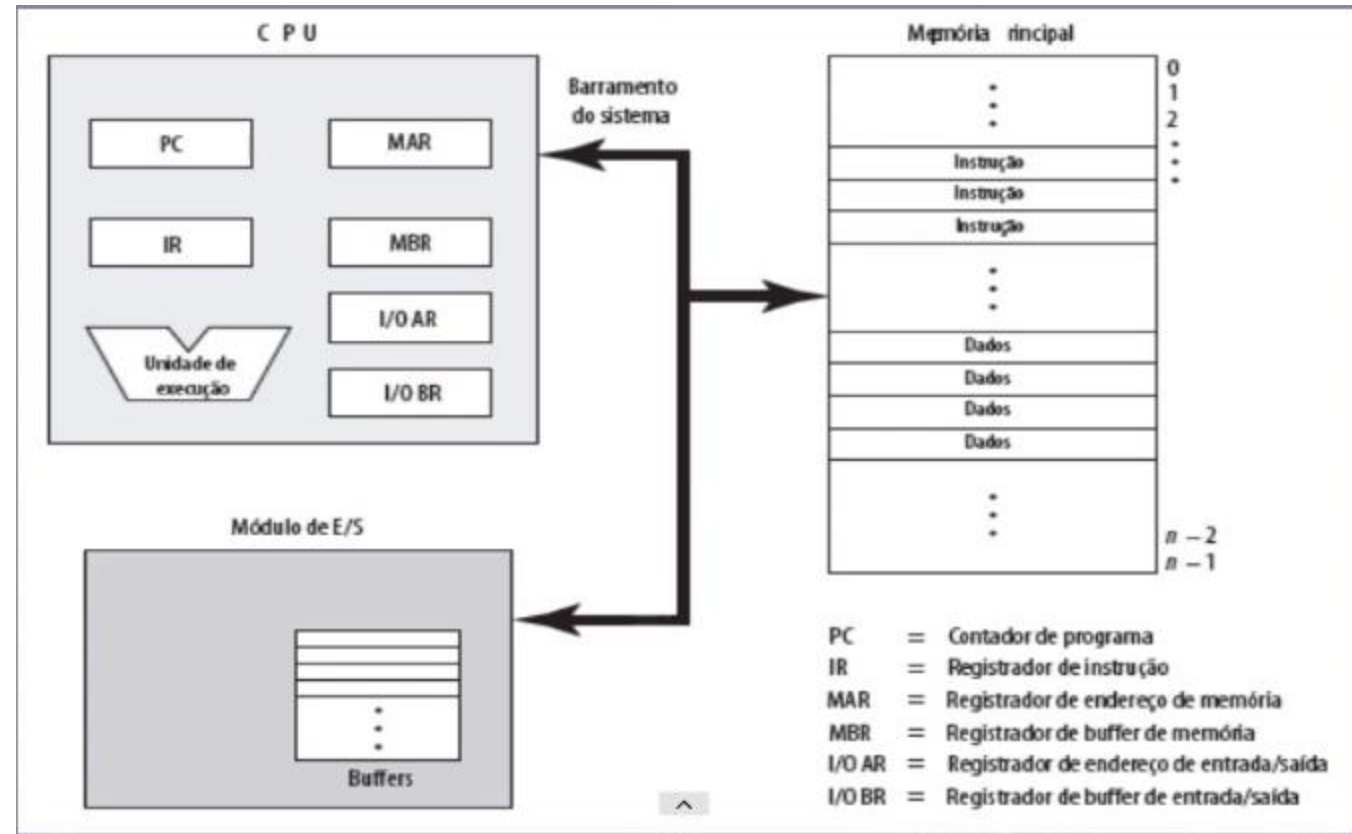


Visão de alto nível

- Meios externos

- Memória principal para armazenamento de dados e instruções.

- Módulos de E/S



Arquitetura do Conjunto de Instrução

- ISA = Instruction Set Architecture
- É o repertório de instruções de um computador.
- Diferentes computadores possuem ISA diferentes mas com muitos aspectos em comum
- Diferentes implementações para um mesmo ISA.

Ex. x86

ISA: Aspectos Principais

- Repertório de Operações
 - Quantas, quais e complexidade
 - Ex.: ULA, transferência de dados, E/S, controle,...
- Tipos e Tamanhos dos Operandos (dados)
- Endereçamento dos operandos
- Formato de instrução
 - N° de operandos (explícitos), tamanho em bits, ...
- Armazenamento dos operandos
 - Registrador, memória, ...; e suas características

CISC (Complex Instruction Set Computer)

- Complexo: grande quantidade de instruções com múltiplos modos de endereçamento
- Contexto:
 - memória era cara e pequena,
 - Códigos compactos e eficientes na execução.
 - Microprogramação.

CISC – exemplos de instruções

- CAS – compare and swap operands
- RTR – return and restore codes
- SWAP – swap register words

CISC – Características de Projeto

- Formato de dois operandos mais comum
- Modos registrador-registrador; registrador-memória e memória-registrador
- Múltiplos modos de endereçamento
- Instrução com largura variável
- Instruções requerem múltiplos ciclos de relógio para sua complexa execução
- Hardware possuem poucos registradores.

RISC (Reduced Instruction Set Computer)

- Pequeno conjunto de instruções
- Todas as instruções tem tamanho fixo
- Execução otimizada de chamada de funções (Call / Return)
- Poucos modos de endereçamento
- Controle por hardware
- Execução rápida de cada instrução (1 por ciclo de relógio)

Visão de Alto Nível

- Características de uma máquina hipotética



(a) Formato de instrução



(b) Formato de inteiro

Contador de programa (PC) = Endereço da instrução
Registrador de instrução (IR) = Instrução sendo executada
Acumulador (AC) = Armazenamento temporário

(c) Registradores internos da CPU

0001 = Carrega AC da memória
0010 = Armazena AC na memória
0101 = Adiciona da memória ao AC

(d) Lista parcial de opcodes

Visão de Alto Nível

- Características de uma máquina hipotética



(a) Formato de instrução



(b) Formato de inteiro

Contador de programa (PC) = Endereço da instrução
Registrador de instrução (IR) = Instrução sendo executada
Acumulador (AC) = Armazenamento temporário

(c) Registradores internos da CPU

0001 = Carrega AC da memória
0010 = Armazena AC na memória
0101 = Adiciona da memória ao AC

(d) Lista parcial de opcodes

Visão de Alto Nível

- Ações realizadas pelo opcode da instrução
 - Processador interpreta os bits do opcode e realiza uma das seguintes ações:
 - Processador – memória: transfere dados entre processador e memória
 - Processador – E/S: transfere dados entre processador e periféricos (módulos de E/S)
 - Processamento de Dados: realiza operações aritméticas ou lógicas
 - Controle: muda a sequência de operação de execução atual (desvio)

Conjuntos de Instruções

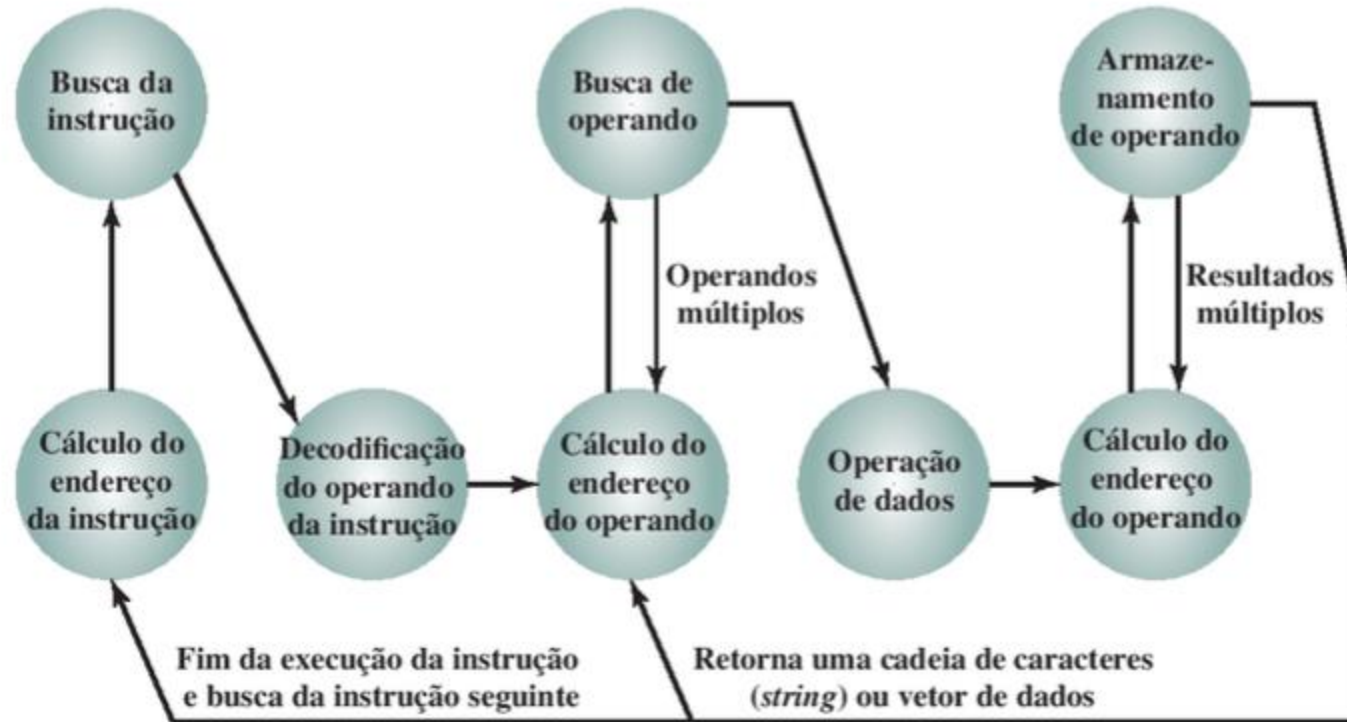
- Definição
 - A coleção das diferentes instruções (códigos de máquina ou operação – opcode) que um processador (CPU) pode executar.
 - Representada através de códigos mnemônicos (linguagem de programação assembly).

Conjuntos de Instruções

- Elementos de uma instrução de máquina
 - Código de operação: especifica a operação a ser realizada (por exemplo, ADD, E/S).
 - Especificada por um código de operação ou opcode (operation Code).
 - Referencia à operando fonte: a operação pode envolver um ou mais operandos fontes.
 - Referencia a operando destino: a operação pode produzir um resultado.
 - Referencia a próxima instrução: diz ao processador onde buscar a próxima instrução após o termino da execução da instrução corrente.

Conjuntos de Instruções

- Diagrama de estado do ciclo da instrução



Conjunto de Instruções

- Tipos de operações
 - Transferência de dados
 - Aritmética
 - Lógica
 - Conversão
 - E/S
 - Controle do sistema
 - Transferência de controle

Conjunto de Instruções

- Transferência de dados
 - Definem
 - Origem
 - Destino
 - Quantidade de dados transferidos
 - Qual instrução para o tipo de transferência
 - Modo de endereçamento para cada operando

Conjunto de Instruções

- Conjunto básico de instruções

Tipo	Nome da operação	Descrição
Transferência de dados	Move (transferência)	Transfere palavra ou bloco da origem ao destino
	Store (armazenamento)	Transfere palavra do processador para a memória
	Load (busca)	Transfere palavra da memória para o processador
	Exchange (troca)	Troca o conteúdo da origem e do destino
	Clear (reset)	Transfere palavra de 0s para o destino
	Set	Transfere palavra de 1s para o destino
	Push	Transfere palavra da origem para o topo da pilha
	Pop	Transfere palavra do topo da pilha para o destino

Conjunto de Instruções

- Aritmética
 - Adição, subtração, multiplicação e divisão
 - Inteiro com ou sem sinal (valos absoluto $|a|$)
 - Ponto Flutuante
 - Incremento ($a++$)
 - Decremento ($a--$)
 - Negação ($-a$)

Sendo a um operando

Conjunto de Instruções

- Conjunto básico de instruções

Tipo	Nome da operação	Descrição
<u>Aritmética</u>	Add	Calcula a soma de dois operandos
	Subtract	Calcula a diferença de dois operandos
	Multiply	Calcula o produto de dois operando
	Divide	Calcula o quociente de dois operandos
	Absolute	Substitui o operando pelo seu valor absoluto
	Negate	Troca o sinal do operando
	Increment	Soma 1 ao operando
	Decrement	Subtrai 1 do operando

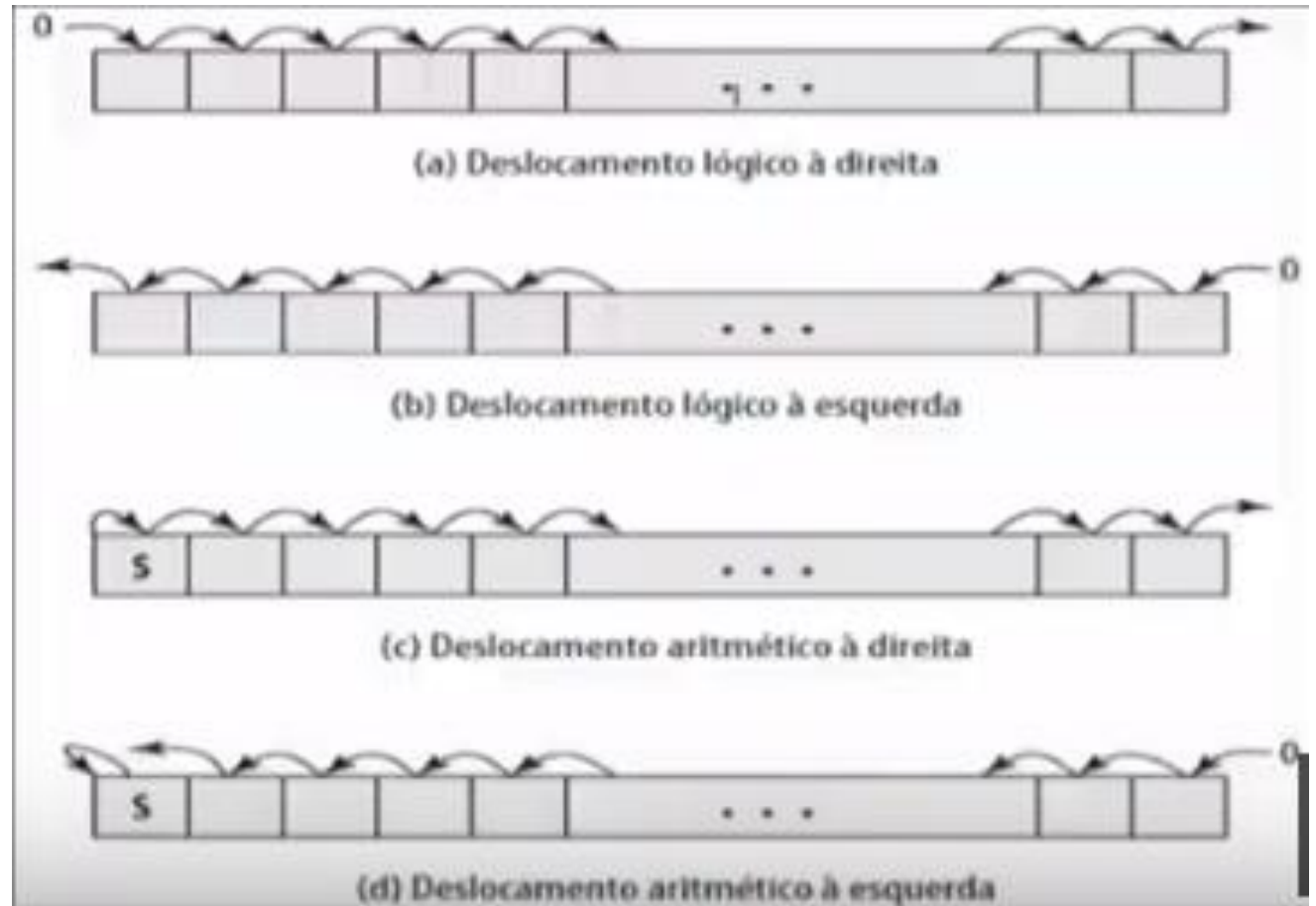
Conjunto de Instruções

- Lógica
 - Operações lógicas sobre dados booleanos e binários.
 - NOT: inverte um bit.
 - AND, OR, exclusive-or (XOR).
 - EQUAL: teste binário
 - Deslocamentos lógicos:
 - Bits deslocados para direita ou esquerda.
 - Bits rotacionados a direita ou a esquerda.

Conjunto de Instruções

- Lógico – operações de deslocamento
 - Úteis para isolar campos dentro de uma palavra.

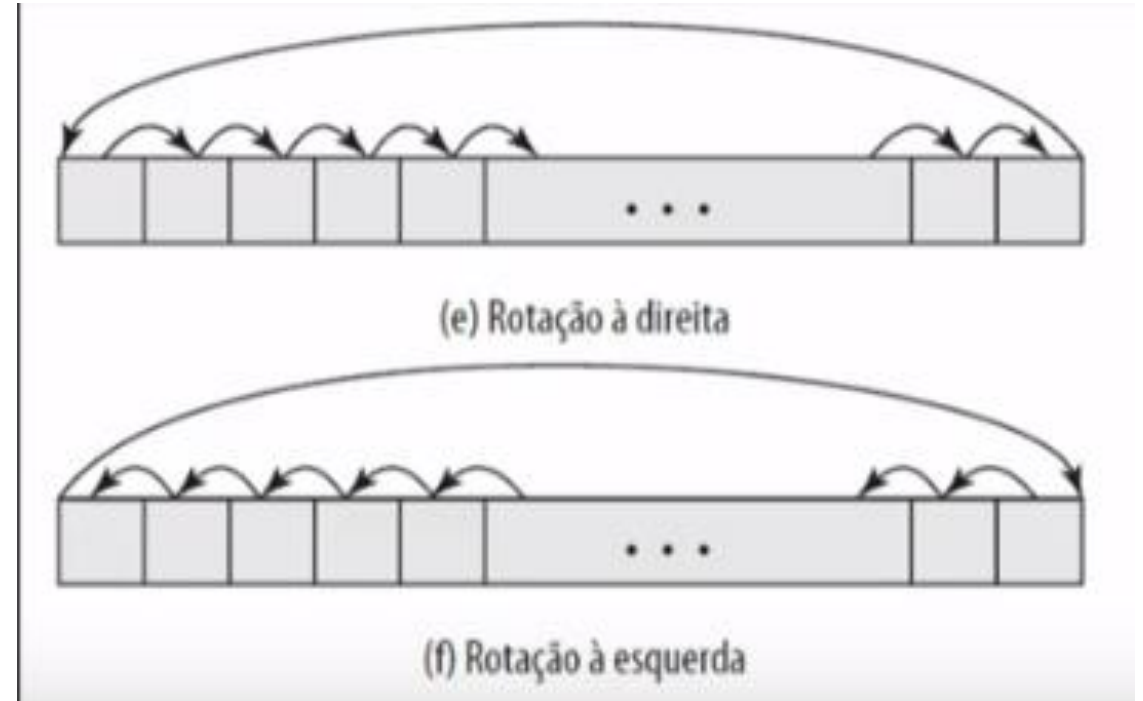
Ex.: aplicação na transmissão de caracteres por uma linha de comunicação.



Conjunto de Instruções

- Lógico – operações de rotação
 - Deslocamento cíclico.
 - Preservam todos os bits como operando.

Ex.: troca entre bits mais e menos significativos



Conjunto de Instruções

- Conjunto básico de instruções

Lógica	AND	Realiza o AND lógico
	OR	Realiza o OR lógico
	NOT (complemento)	Realiza o NOT lógico
	Exclusive-OR	Realiza o XOR lógico
	Test	Testa condição especificada; define flag(s) com base no resultado
	Compare	Faz comparação lógica ou aritmética de dois ou mais operandos; define flag(s) com base no resultado
	Definir variáveis de controle	Classe de instruções para definir controles para fins de proteção, tratamento de interrupção, controle de tempo etc.
	Shift	Desloca o operando para a esquerda (direita), introduzindo constantes na extremidade
	Rotate	Desloca ciclicamente o operando para a esquerda (direita), de uma extremidade à outra

Conjunto de Instruções

- Conversão

- Mudam o formato ou operam sobre o formato dos dados

Ex.: conversão de decimal para binário (10D -> 01010B)

Conjunto de Instruções

- Conjunto básico de instruções

Conversão	Translate	Traduz valores em uma seção da memória com base em uma tabela de correspondências
	Convert	Converte o conteúdo de uma palavra de uma forma para outra (por exemplo, decimal empacotado para binário)

Conjunto de Instruções

- E/S
 - E/S mapeada em memória: utiliza instruções de movimentação de dados sem distinção entre memória e E/S
 - E/S mapeada independentemente: adota comandos especiais de E/S
 - Acesso direto à memória (direct memory access – DMA): processador especializado de E/S assume o controle da operação de E/S para movimentar grandes quantidades de dados.

Conjunto de Instruções

- Conjunto básico de instruções

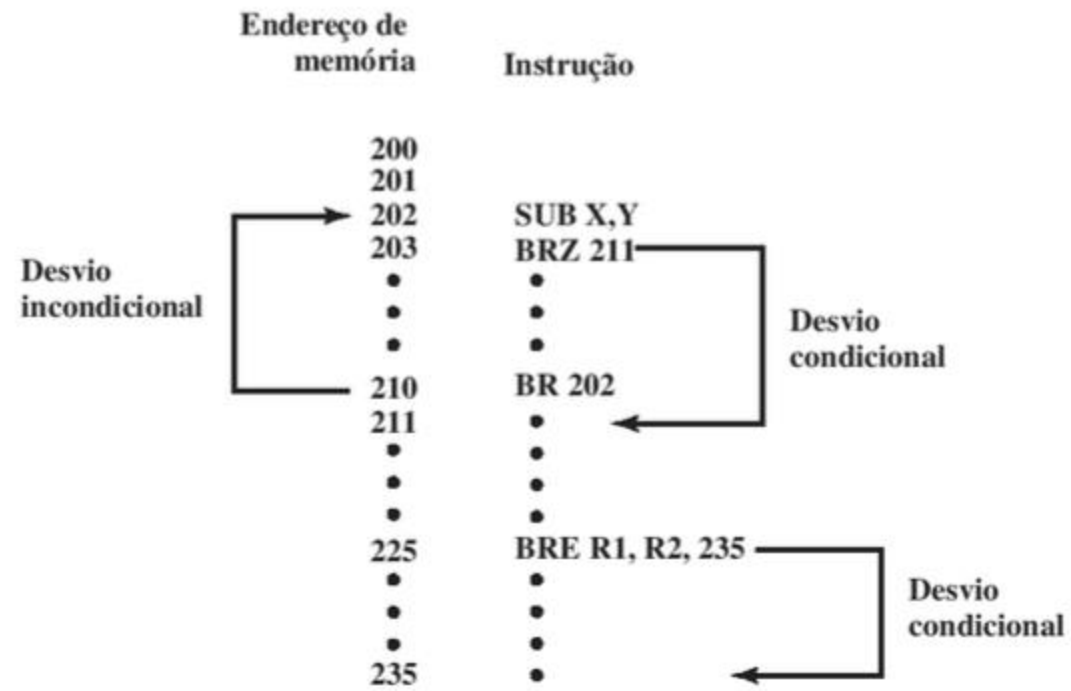
Entrada/ saída	Input (leitura)	Transfere dados da porta de E/S ou dispositivo especificado para o destino (por exemplo, memória principal ou registrador do processador)
	Output (escrita)	Transfere dados da origem especificada para porta de E/S ou dispositivo
	Start I/O	Transfere instruções para o processador de E/S para iniciar operação de E/S
	Test I/O	Transfere informações de estado do sistema de E/S para destino especificado

Conjunto de Instruções

- Transferência de controle
 - Instrução de desvio (endereço explícito)
 - Condicional: desvio ocorre se a condição for satisfeita
Ex. desvio para o endereço xxxx se o resultado de uma operação for zero.
 - Incondicional: desvio sempre ocorre
Ex. desvio para o endereço xxxx
 - Instrução de salto (SKIP, endereço implícito).
Incrementa e salta uma instrução se for zero (ISZ – Increment-and-SLIP-if-zero)
 - Chamada de sub-rotina ou procedimento:
Chamada de interrupção de E/S

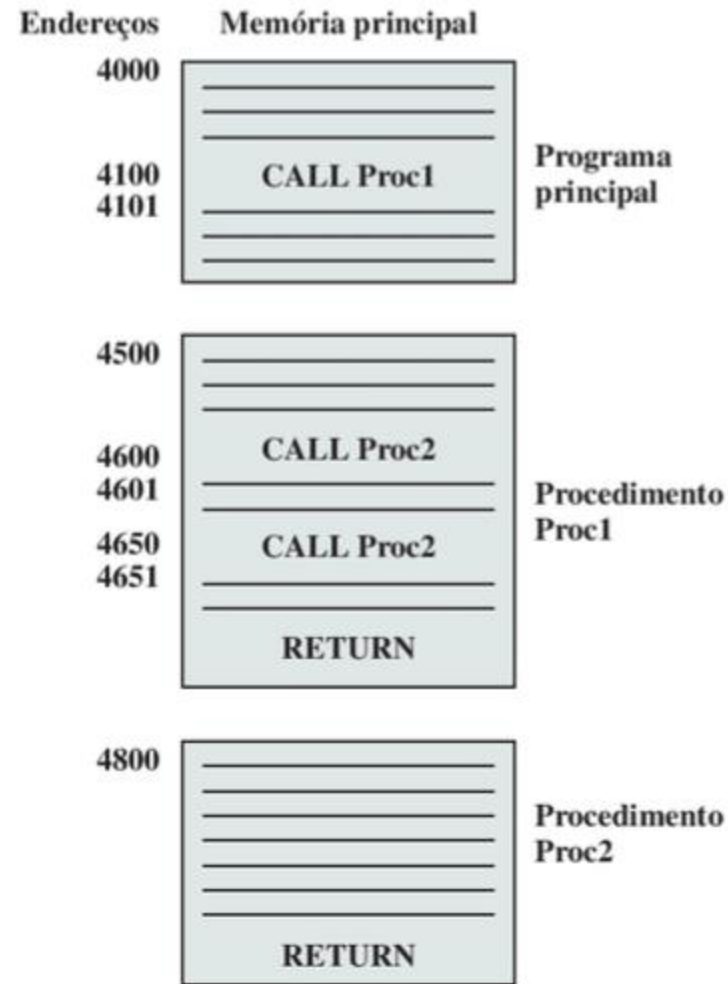
Conjunto de Instruções

- Transferência de controle - instruções de desvio

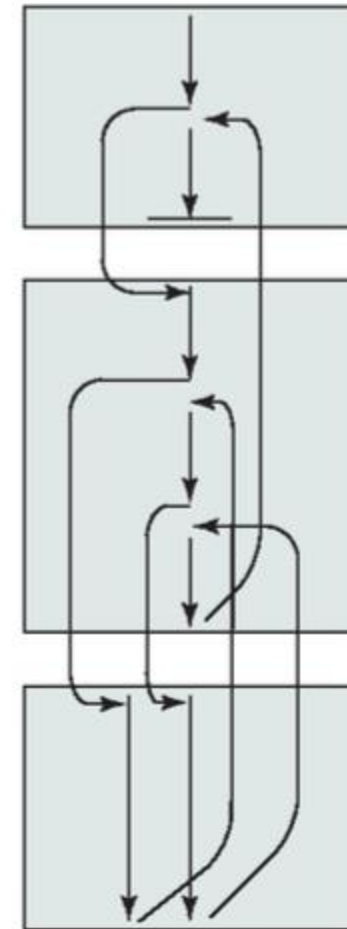


Conjunto de Instruções

- Chamada de sub-rotina



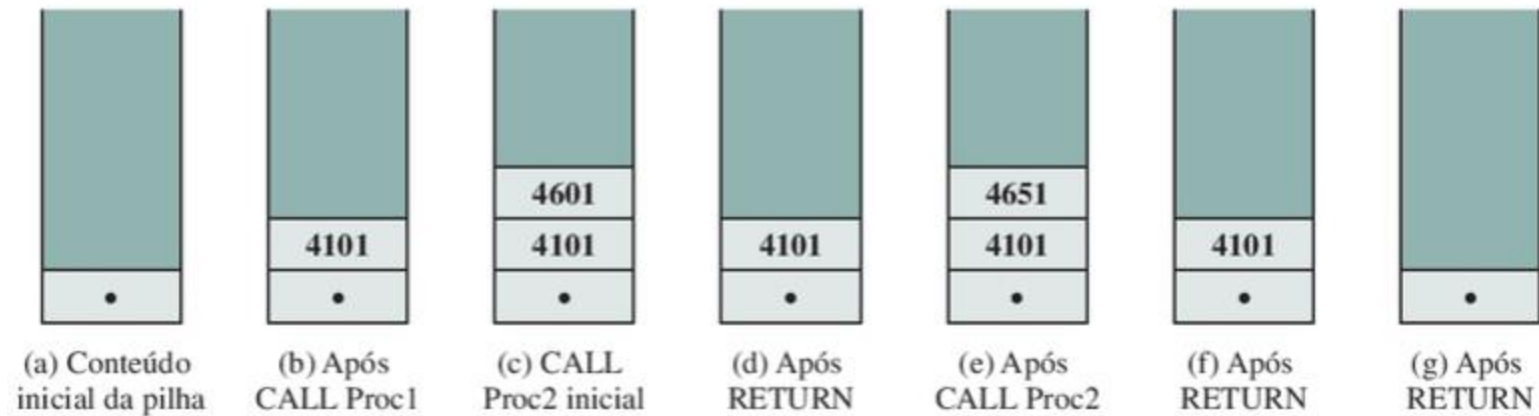
(a) Chamadas e retornos



(b) Sequência de execução

Conjunto de Instruções

- Pilha para implementar sub-rotinas aninhadas



Conjunto de Instruções

- Conjunto básico de instruções

Transferência de controle	Jump (desvio)	Transferência incondicional; carrega PC com endereço especificado
	Jump condicional	Testa condição especificada; ou carrega PC com endereço especificado ou não faz nada, com base na condição
	Jump para sub-rotina	Coloca informação do controle do programa atual em local conhecido; salta para endereço especificado
	Return	Substitui conteúdo do PC por outro registrador de local conhecido
	Execute	Busca operando do local especificado e executa como instrução; não modifica o PC
	Skip	Incrementa o PC para saltar para a próxima instrução
	Skip condicional	Testa condição especificada; ou salta ou não faz nada, com base na condição
	Halt	Termina a execução do programa
	Wait (hold)	Termina a execução do programa; testa condição especificada repetidamente; retoma a execução quando a condição for satisfeita
	No operation	Nenhuma operação é realizada, mas a execução do programa continua

Modos de endereçamento

- Endereçamento
 - Campos de endereços nas instruções relativamente pequenos
 - Referência a grande quantidade de posições de endereço requer uso de técnicas de endereçamento
 - Quase todas as arquiteturas de computadores fornecem um ou mais modos de endereçamento

Modos de endereçamento

- Técnicas de Endereçamento
 - Endereçamento imediato
 - Endereçamento direto
 - Endereçamento indireto
 - Endereçamento de registrador
 - Endereçamento indireto por registrador
 - Endereçamento por deslocamento:
 - Endereçamento relativo
 - Endereçamento por registrador base
 - Indexação (pré e pós indexação).
 - Endereçamento à pilha.

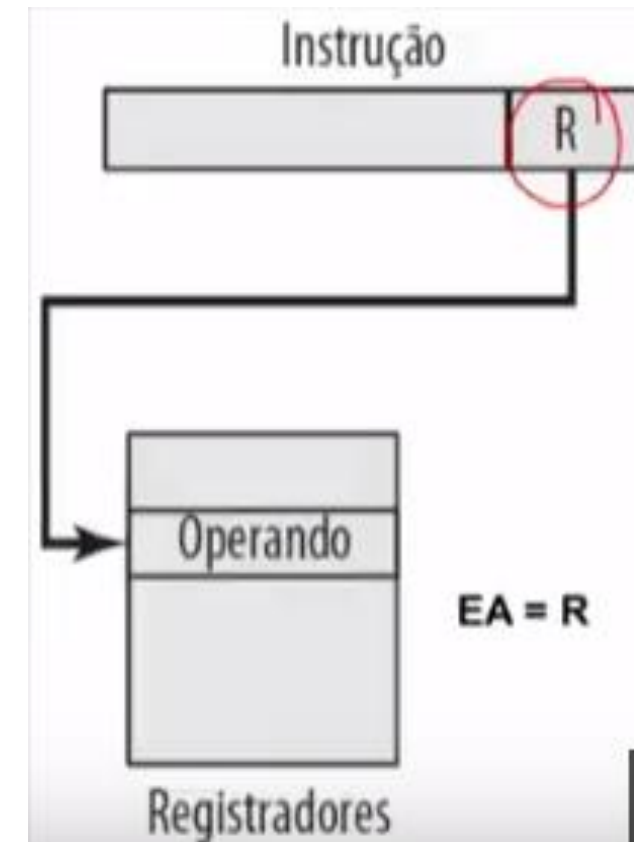
Modos de endereçamento

- Endereçamento de registrador
 - Semelhante ao endereçamento direto.
 - Vantagem:
 - Nenhum acesso a memória
 - Desvantagem:
 - Espaço de endereçamento limitado

ex.,. ADD R

$AC=AC+R$

R contém o registrador cujo conteúdo é o operando



Modos de endereçamento

- Endereçamento imediato
 - Valor do operando é especificado diretamente na instrução
 - Vantagem
 - Nenhum acesso a memória (economia de tempo)
 - Desvantagem
 - Tamanho do operando limitado ao tamanho do campo de endereço da instrução.

Ex. ADD 10H

Adicionando o operando 10H ao acumulador $AC=AC+10H$

Modos de endereçamento

- Endereçamento direto

- O campo de endereço da instrução contém o endereço do operando na memória
- Vantagem
 - Simplicidade
- Desvantagem
 - Espaço de endereçamento limitado

Ex ADD A

Adiciona ao AC o operando apontado pelo endereço contido em A, $AC=AC+EA=A$.

A contém o endereço do operando

EA=endereço Efetivo

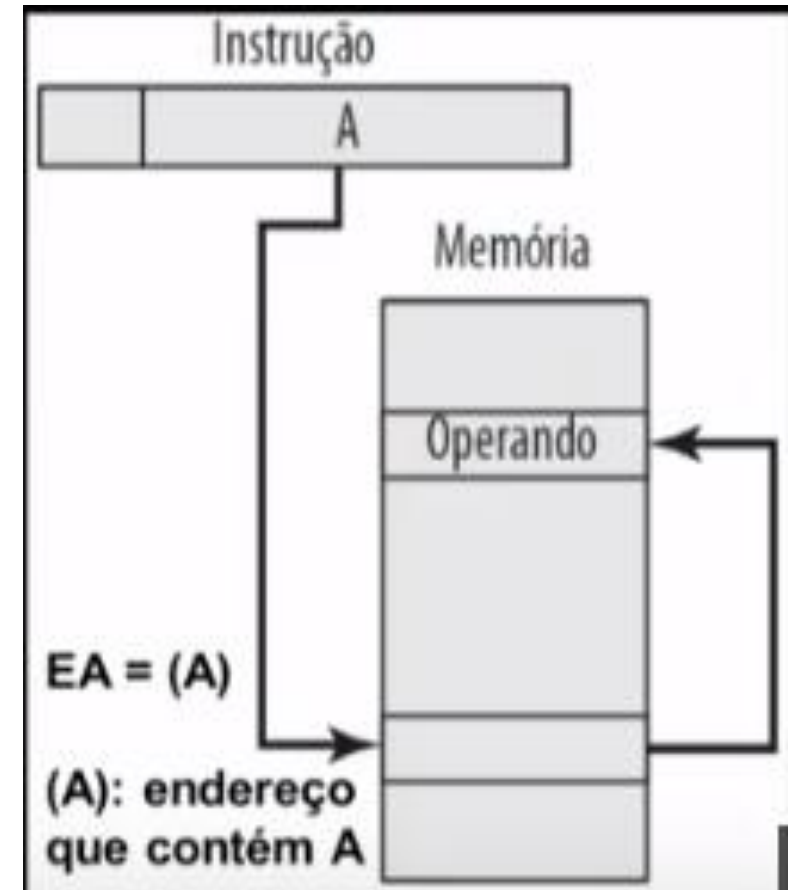


Modos de endereçamento

- Endereçamento indireto
 - Campo de endereço da instrução contém o endereço de memória cujo conteúdo é o endereço da memória onde encontra-se o operando
 - Vantagem:
 - Espaço de endereçamento grande
 - Desvantagem
 - Acesso múltiplos a memória.

Ex. $ADD(A), AC=AC+EA=(A)$

A contém o endereço que aponta para o endereço que contém o operando.



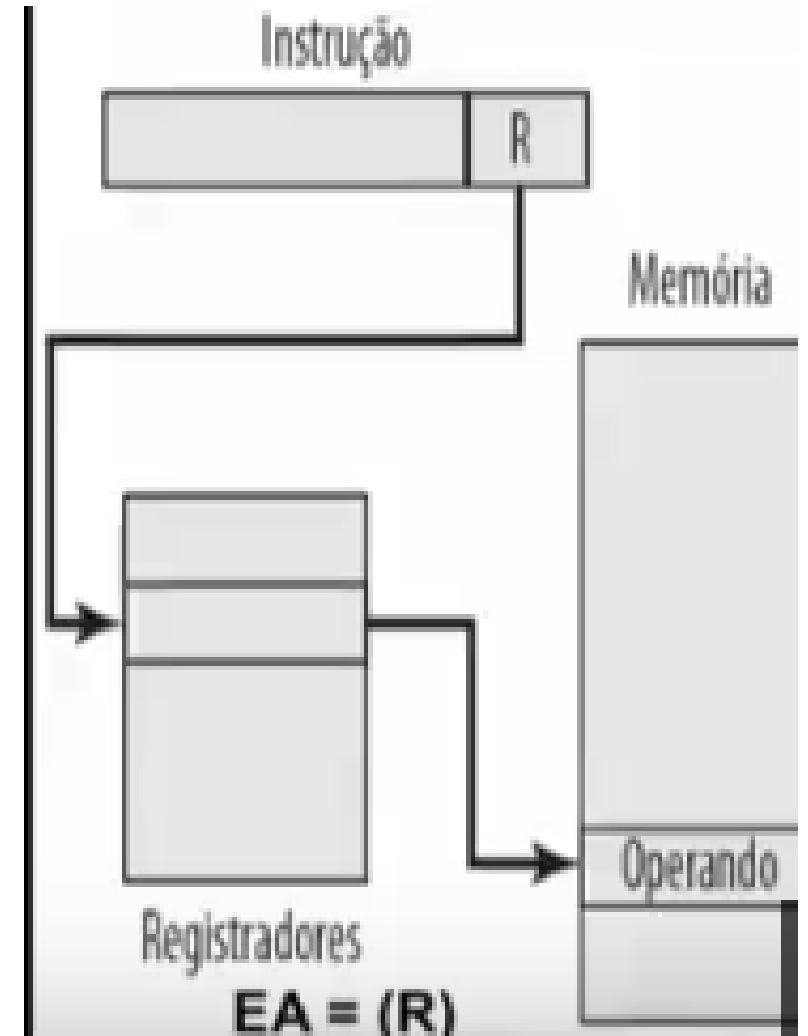
Modos de endereçamento

- Endereçamento indireto por registrador
 - Semelhante ao endereçamento indireto
 - Vantagem:
 - Espaço de endereçamento grande
 - Desvantagem
 - Acesso adicional a memória.

Ex. ADD (R)

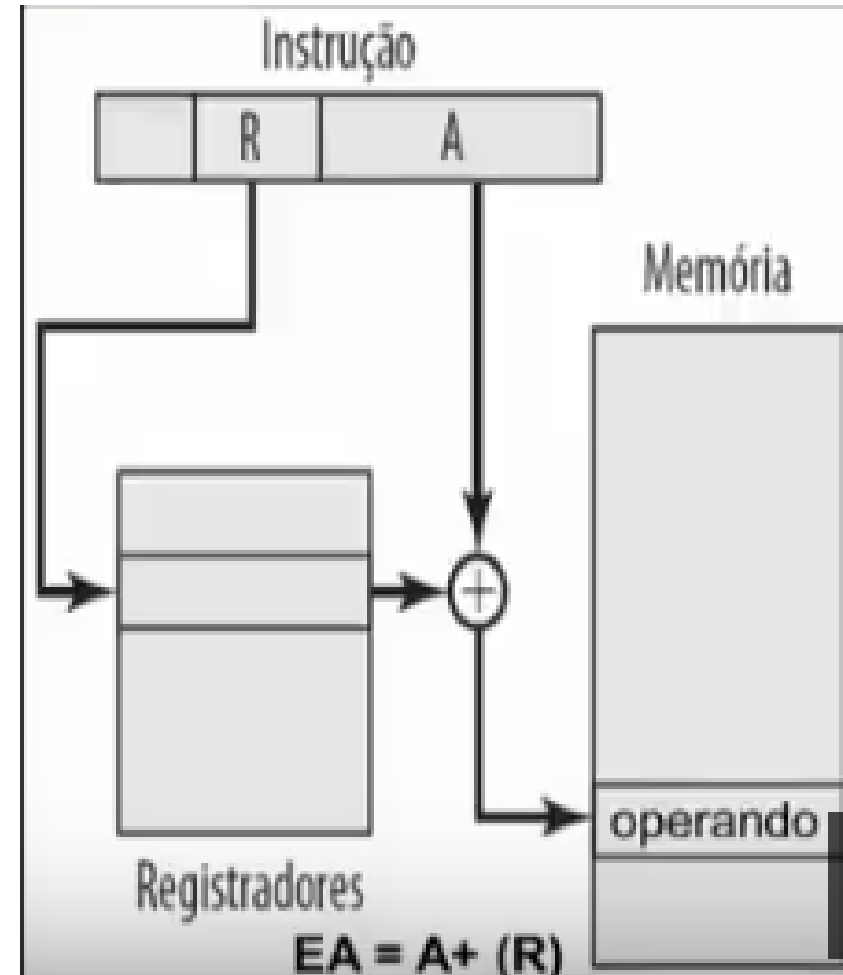
R contém o registrador cujo conteúdo é o endereço de memória que contém o operando.

$AC = AC + EA = (R)$



Modos de endereçamento

- Endereçamento indireto por deslocamento:
 - Combinação dos modos direto e indireto via registrador.
 - Vantagem
 - Flexibilidade de referencia (conceito de localidade, segmentação, operações iterativas).
 - Desvantagem:
 - Complexidade no calculo do EA



Dúvidas

