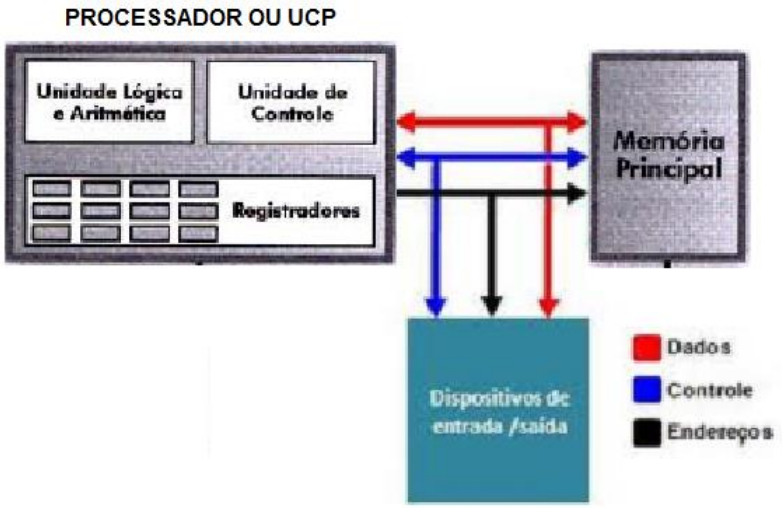
CONJUNTO DE INSTRUÇÕES

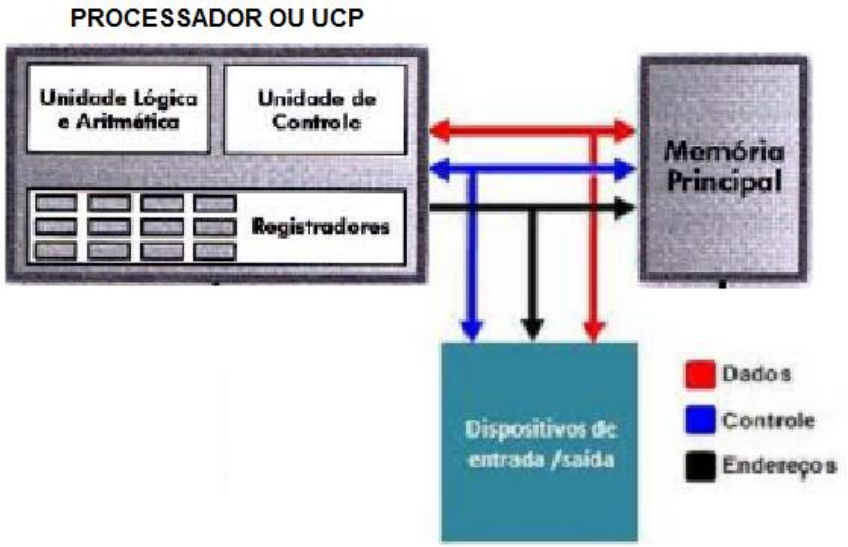
CARACTERÍSTICAS DE INSTRUÇÕES DE MÁQUINA

- Quando um programador usa uma linguagem de alto-nível, como C, muito pouco da arquitetura da máquina é visível.
- O usuário que deseja programar em linguagem de máquina (na verdade linguagem de montagem ou linguagem assembly) deve conhecer:
  - O conjunto de registradores da UCP
  - A estrutura da memória principal (MP)
  - Tipos de dados (endereços, números, caracteres, etc.) disponíveis diretamente na máquina
  - Funcionamento da ULA (unidade lógica aritmética)



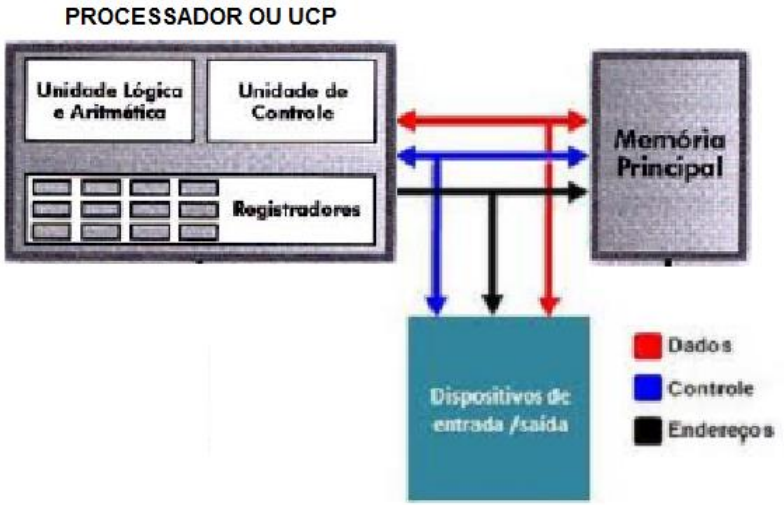
CARACTERÍSTICAS DE INSTRUÇÕES DE MÁQUINA

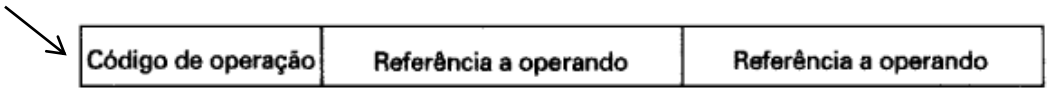
- Implementar uma UCP envolve implementar o conjunto de instruções de máquina.
- As **instruções de máquina** podem ser classificadas nas seguintes categorias gerais:
  - Operações lógicas e aritméticas
  - Operações de movimentações de dados entre **dois registradores**
  - Operações de movimentações de dados entre **registradores e memória**
  - Operações de movimentações de dados entre **duas posições de memória**
  - **Operações de Entrada/Saída**
  - **Operações de controle.**



CARACTERÍSTICAS DE INSTRUÇÕES DE MÁQUINA

- ELEMENTOS DE UMA INSTRUÇÃO DE MÁQUINA
  - Código da operação
    - especifica a operação a ser executada.
  - Referência a operando fonte
    - Operandos que constituem dados de entrada da operação
    - Podem existir mais de um operando fonte
  - Referência a operando de destino
    - A operação pode produzir um resultado
  - Endereço da próxima instrução
    - Indica onde a UCP deve procurar a próxima instrução na MEMÓRIA PRINCIPAL a ser executada. Na maioria dos casos é a que segue imediatamente

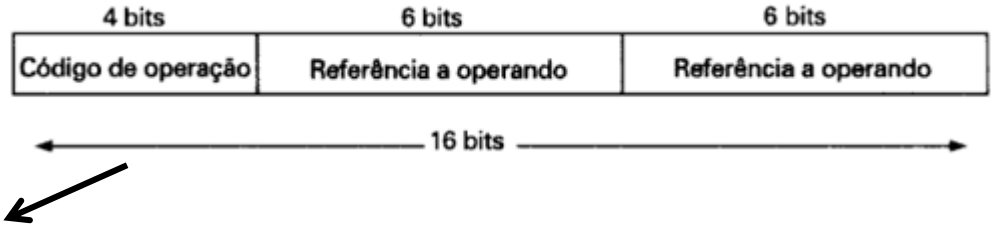


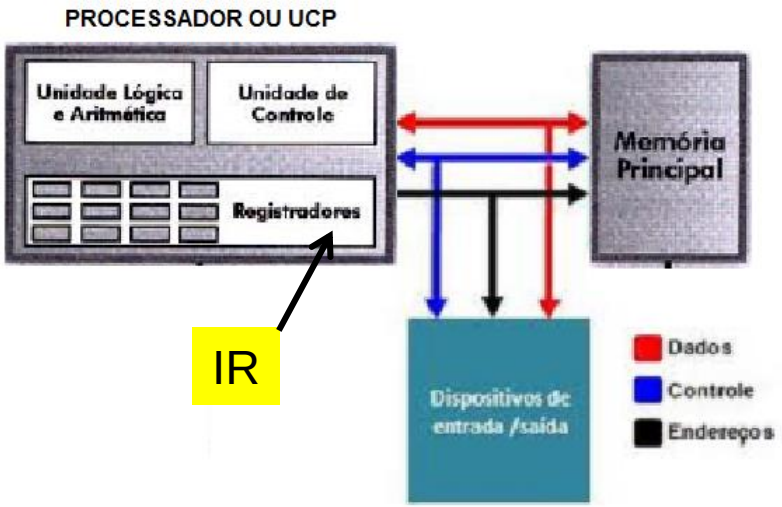


Formato de instrução simples.

CARACTERÍSTICAS DE INSTRUÇÕES DE MÁQUINA

- ELEMENTOS DE UMA INSTRUÇÃO DE MÁQUINA
  - Internamente, cada instrução é representada como uma sequência de bits.
  - Durante a execução de uma instrução a mesma deve ser lida em um registrador de instrução (RI) da UCP.
  - A UCP deve ser capaz de extrair os dados dos vários campos da instrução armazenada no RI e efetuar a operação requerida.





CARACTERÍSTICAS DE INSTRUÇÕES DE MÁQUINA

- LINGUAGEM ASSEMBLY
  - Tornou-se prática comum usar uma representação simbólica (de mnemônicos) conhecida como **instrução em Assembly** para cada **instrução de máquina**.
  - Alguns exemplos comuns

ADD	Adição
SUB	Subtração
MPY	Multiplicação
DIV	Divisão
LOAD	Carregar dados da memória
STOR	Armazenar dados na memória

  - Os **operandos da instrução** são também representados de maneira simbólica. **Por exemplo**: adicionar o valor contido na posição Y com o conteúdo do registrador R

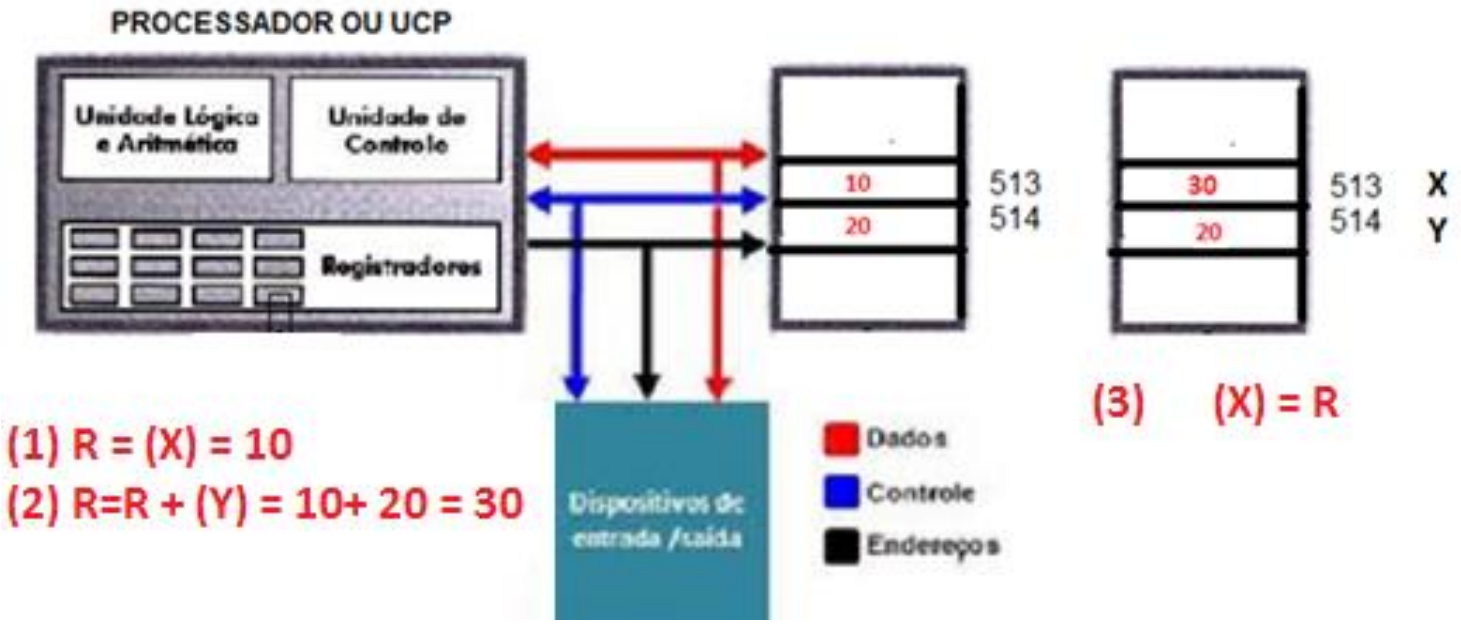
```
ADD R, Y
```

CARACTERÍSTICAS DE INSTRUÇÕES DE MÁQUINA

- LINGUAGEM ASSEMBLY
  - A linguagem Assembly permanece como ferramenta útil para descrever instruções de máquina.
  - Cada código em Assembly tem sua representação binária correspondente
  - Uma instrução em uma linguagem de alto nível, tal como C, pode requer várias instruções em linguagem Assembly :
  - Por exemplo
    - X = X + Y (instrução em linguagem de alto-nível)
    - Essa instrução (em linguagem de alto nível) instrui ao computador adicionar o valor armazenado em Y ao valor armazenado em X e colocar o resultado em X.

CARACTERÍSTICAS DE INSTRUÇÕES DE MÁQUINA

- LINGUAGEM ASSEMBLY
  - X = X + Y (instrução em linguagem de alto-nível)
  - Supondo que as variáveis X em Y correspondam, respectivamente, as posições de memória 513 e 514, em instrução de máquina o comando acima pode ser implementado com três instruções:
  - (1) Carregar um registrador com o conteúdo de posição de memória 513.
  - (2)Adicionar o conteúdo da posição de memória 514 ao conteúdo do registrador.
  - (3). Armazenar o conteúdo do registrador na posição de memória 513



CARACTERÍSTICAS DE INSTRUÇÕES DE MÁQUINA

- LINGUAGEM ASSEMBLY
  - EXEMPLO DE UTILIZAÇÃO

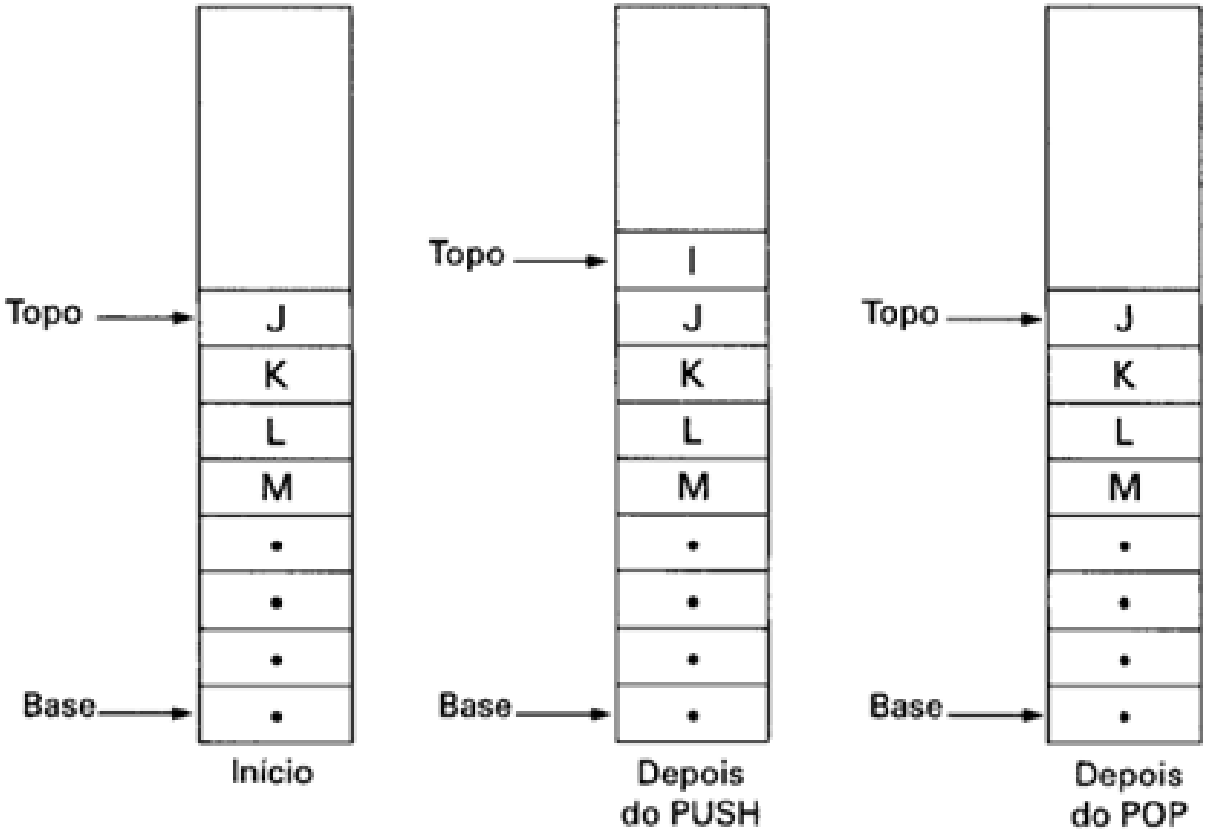
Rótulo	Operação	Operando
FORMUL	LDA	I
	ADD	J
	ADD	K
	STA	N
I	DATA	2
J	DATA	3
K	DATA	4
N	DATA	0

Programa em linguagem de montagem

Computação da fórmula N = I + J + K.

CONJUNTO DE INSTRUÇÕES

- PILHA
  - É um conjunto ordenado de elementos em que apenas um pode ser acessado em um dado instante (o topo da pilha). Duas operações:
  - PUSH (item): adiciona item na PILHA
  - POP: remove o item do topo da PILHA

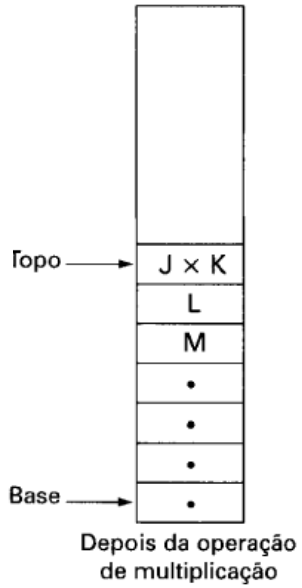


CONJUNTO DE INSTRUÇÕES

- PILHA
  - **As pilhas podem ser usadas para implementar operações.**
  - **Operações binárias**:
  - requerem dois operandos (multiplicação, divisão, soma, subtração) retiram dois operandos do topo da pilha e colocam o resultado de volta na pilha.
  - **A = POP; B= POP; C= A+B; PUSH (C).**
  - **Operações unárias**:
  - requererem um operando (NOT por exemplo), usa o item do topo da pilha.
  - **A = POP; C= NOT(C) ; PUSH(C)**

Operações sobre pilhas

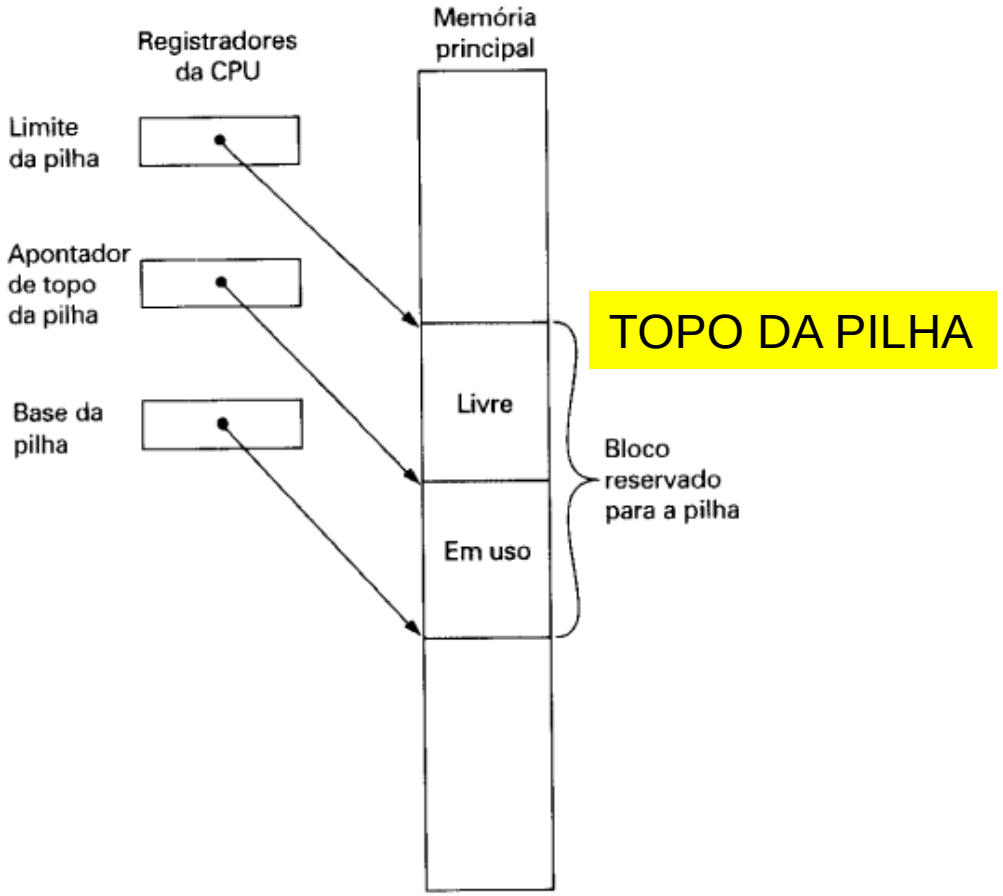
PUSH	Coloca um novo elemento no topo da pilha.
POP	Retira o elemento do topo da pilha.
Operação unária	Efetua a operação sobre o elemento do topo da pilha. Substitui o elemento do topo pelo resultado.
Operação binária	Efetua a operação sobre os dois elementos no topo da pilha. Retira os dois elementos do topo da pilha. Coloca o resultado da operação no topo da pilha.



Depois da operação de multiplicação

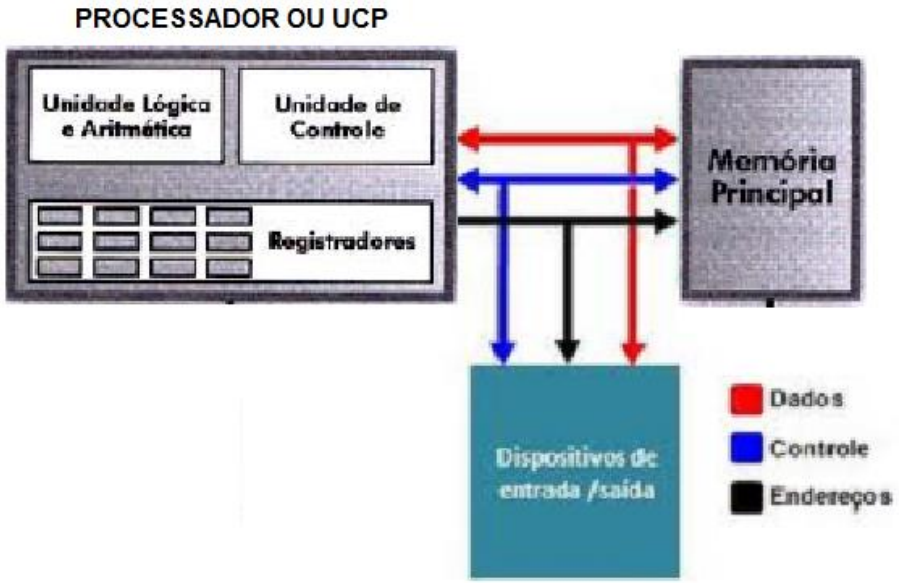
CONJUNTO DE INSTRUÇÕES

- PILHA (IMPLEMENTAÇÃO)
  - Um bloco contíguo de posições de memória principal é reservado para pilha.
  - Base da pilha: endereço da posição inicial do bloco de memória reservado para a pilha.
  - Limite da pilha: endereço da extremidade do bloco de memória reservado para a pilha.
  - Apontador da pilha: endereço do topo da pilha.



CARACTERÍSTICAS DE INSTRUÇÕES DE MÁQUINA

- As **instruções de máquina** podem ser classificadas nas seguintes categorias gerais:
  - **Operações lógicas e aritméticas** (processamento de dados)
  - Operações de movimentações de dados entre **dois registradores**
  - Operações de movimentações de dados entre **registradores e memória**
  - Operações de movimentações de dados entre **duas posições de memória**
  - **Operações de Entrada/Saída**
  - **Operações de controle** (testes e desvios)



CARACTERÍSTICAS DE INSTRUÇÕES DE MÁQUINA

- NÚMERO DE ENDEREÇOS EM UMA INSTRUÇÃO
- INSTRUÇÕES COM ZERO, UM DOIS OU TRÊS ENDEREÇOS

Utilização de endereços em instruções (exceto instruções de desvio)

Número de endereços	Representação simbólica	Interpretação
3	OP A, B, C	A ← B OP C
2	OP A, B	A ← A OP B
1	OP A	AC ← AC OP A
0	OP	T ← (T–1) OP T

AC = acumulador
T = topo da pilha
A, B, C = registrador ou posição de memória

- Instrução com menos endereços resultam em uma UCP menos complexa.
- **Com instruções com 01 endereço**
  - Usa-se apenas um registrador de propósito geral (o acumulador → AC)
- **Com instruções com 02 ou mais endereços**
  - É comum haver múltiplos registradores de propósito geral.

CARACTERÍSTICAS DE INSTRUÇÕES DE MÁQUINA

- NÚMERO DE ENDEREÇOS EM UMA INSTRUÇÃO
- **Instruções com 03 endereços:**
  - **Dois endereços de operandos e um endereço de resultado.**
  - **Não são comuns.**

Instrução		Comentário
SUB	Y, A, B	Y ← A − B
MPY	T, D, E	T ← D × E
ADD	T, T, C	T ← T + C
DIV	Y, Y, T	Y ← Y ÷ T

- **Instruções com 02 endereços:**
  - **Um endereço é de operando e o outro é um resultado.**

Instrução		Comentário
MOVE	Y, A	Y ← A
SUB	Y, B	Y ← Y − B
MOVE	T, D	T ← D
MPY	T, E	T ← T × E
ADD	T, C	T ← T + C
DIV	Y, T	Y ← Y ÷ T

CARACTERÍSTICAS DE INSTRUÇÕES DE MÁQUINA

- NÚMERO DE ENDEREÇOS EM UMA INSTRUÇÃO
  - **Instruções com 01 endereço:**
    - **Um endereço implícito é o do registrador acumulador (AC) da UCP**

Instrução		Comentário
LOAD	D	AC ← D
MPY	E	AC ← AC × E
ADD	C	AC ← AC + C
STOR	Y	Y ← AC
LOAD	A	AC ← A
SUB	B	AC ← AC – B
DIV	Y	AC ← AC ÷ Y
STOR	Y	Y ← AC

  - **Instruções com 0 endereço:**
    - **Esse formato se aplica a uma organização de memória especial, denominada pilha.**

PROJETO DO CONJUNTO DE INSTRUÇÕES

- Algumas questões relativas ao projeto de um conjunto de instruções:

Repertório de operações: quantas e quais são as operações que devem ser fornecidas e quão complexas elas podem ser.

Tipos de dados: quais os tipos de dados sobre os quais as operações são efetuadas.

Formatos de instrução: qual o tamanho das instruções (em bits), o número de endereços por instrução, o tamanho dos vários campos etc.

Registradores: qual o número de registradores da CPU que podem ser usados pelas instruções e qual o propósito de cada um.

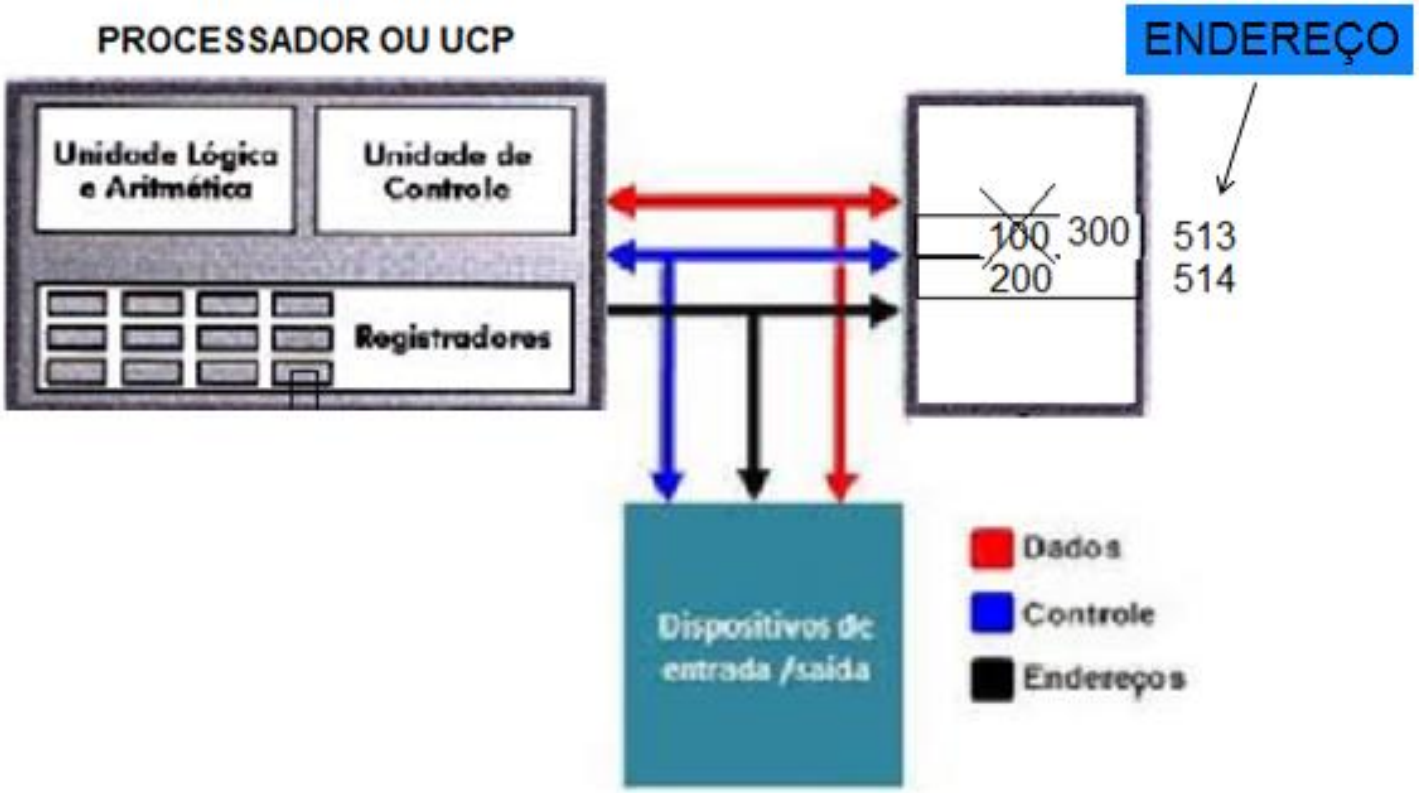
Endereçamento: de que modo (ou modos) o endereço de um operando pode ser especificado.

TIPOS DE OPERANDOS

- Instruções de máquina operam sobre dados (operandos). As CLASSES são as seguintes
  - Endereços
  - Números
  - Caracteres
  - Dados Lógicos

TIPOS DE OPERANDOS

- ENDEREÇO
  - É uma forma de dado



TIPOS DE OPERANDOS

- NÚMERO
  - Inteiro
  - Ponto flutuante
  - Número decimal
    - Às vezes é preferível armazenar e operar dados na forma decimal.
    - Usa-se comumente a representação BCD (decimal codificada em binário).
    - Exemplo: 246 = 001001000110
    - É menos compacto que a representação binária, mas evita o custo da conversão.

Decimal	BCD 8421			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
11	1	0	1	1
12	1	1	0	0
13	1	1	0	1
14	1	1	1	0
15	1	1	1	1

TIPOS DE OPERANDOS

- CARACTERE
  - O código de caracteres mais usado é o alfabeto de referência internacional conhecido como código ASCII.
  - Cada caractere é representado por um conjunto de 7 bits

CODIGO ASCII

Caracteres ASCII extendidos imprimibles :

codigo ascii 128 = Ç (Letra C cedilla mayúscula)

codigo ascii 129 = ü (Letra u minúscula con diéresis)

codigo ascii 130 = é (Letra e minúscula con acento agudo)

codigo ascii 131 = â (Letra a minúscula con acento circunflejo)

codigo ascii 132 = ä (Letra a minúscula con diéresis)

codigo ascii 133 = à (Letra a minúscula con acento grave)

codigo ascii 134 = å (Letra a minúscula con anillo)

codigo ascii 135 = ç (Letra c cedilla minúscula)

codigo ascii 136 = ê (Letra e minúscula con acento circunflejo)

TIPOS DE OPERANDOS

- DADO LÓGICO
  - É quando uma unidade de N bits é considerada como composta de N itens de dados.
  - Exemplo: O dado lógico 11010101 é composto de 8 bits.
  - Vantagem:
    - **manipulação dos bits do item de dados**
    - **11010101**
    - **AND**
    - **10010001**
    - **10010001**

TIPOS DE OPERAÇÕES

- O número de códigos de operação varia de máquina para máquina, mas o mesmo conjunto de classe de operações é encontrada em todas as máquinas:
  - 1. Operações de transferência de dados
  - 2. Operações aritméticas
  - 3. Operações lógicas
  - 4. Operações de conversão
  - 5. Operações de E/S
  - 6. Operações de controle do sistema
  - 7. Operações de transferência de controle.

TIPOS DE OPERAÇÕES

- 1. OPERAÇÕES DE TRANSFERÊNCIA DE DADOS
  - Deve especificar os endereços fonte e destino da operação (posição de memória, registrador ou topo da pilha)
  - Deve indicar o tamanho dos dados a serem transferidos
  - Deve indicar o modo de endereçamento de cada operando (iremos ver).
  - BASEADA EM HAYES, 1988

Tipo	Nome da operação	Descrição
Operações de transferência de dados	Move	Transfere uma palavra ou bloco da fonte para o destino
	Store	Transfere uma palavra do processador para a memória
	Load	Transfere uma palavra da memória para o processador
	Exchange	Troca os conteúdos dos operandos fonte e de destino
	Clear	Transfere uma palavra contendo 0s para o destino
	Set	Transfere um palavra contendo 1s para o destino
	Push	Transfere uma palavra da fonte para o topo da pilha
	Pop	Transfere uma palavra do topo da pilha para o destino

TIPOS DE OPERAÇÕES

- 1. OPERAÇÕES DE TRANSFERÊNCIA DE DADOS
  - IBM S/370

Mnemônico da operação	Nome	Número de bits transferidos	Descrição
L	Load	32	Transfere da memória para o registrador
LH	Load Halfword	16	Transfere da memória para o registrador
LR	Load	32	Transfere do registrador para o registrador
LER	Load (Short)	32	Transfere do registrador de ponto flutuante para o registrador de ponto flutuante
LE	Load (Short)	32	Transfere da memória para o registrador de ponto flutuante
LDR	Load (Long)	64	Transfere do registrador de ponto flutuante para o registrador de ponto flutuante
LD	Load (Long)	64	Transfere da memória para o registrador de ponto flutuante
ST	Store	32	Transfere do registrador para a memória

TIPOS DE OPERAÇÕES

- 2. OPERAÇÕES ARITMÉTICAS
  - A maioria das máquinas oferece operações aritméticas básicas tais como adição, subtração, multiplicação e divisão.
  - Outras possíveis operações incluem instruções com 01 operando.
  - BASEADA EM HAYES, 1988

Tipo	Nome da operação	Descrição
Operações aritméticas	Add	Soma dois operandos
	Subtract	Calcula a diferença entre dois operandos
	Multiply	Calcula o produto de dois operandos
	Divide	Calcula o quociente de dois operandos
	Absolute	Substitui o operando pelo seu valor absoluto
	Negate	Muda o sinal do operando
	Increment	Soma 1 ao operando
	Decrement	Subtrai 1 do operando

TIPOS DE OPERAÇÕES

- 3. OPERAÇÕES LÓGICAS
  - Operações lógicas que podem ser efetuadas sobre dados binários ou booleanos:

Operações lógicas básicas

P	Q	NOT P	P AND Q	P OR Q	P XOR Q	P=Q
0	0	1	0	0	0	1
0	1	1	0	1	1	0
1	0	0	0	1	1	0
1	1	0	1	1	0	1

  - As operações lógicas podem ser aplicadas bit a bit a uma unidade de dados lógicos de n bits. Se dois registradores contem os dados

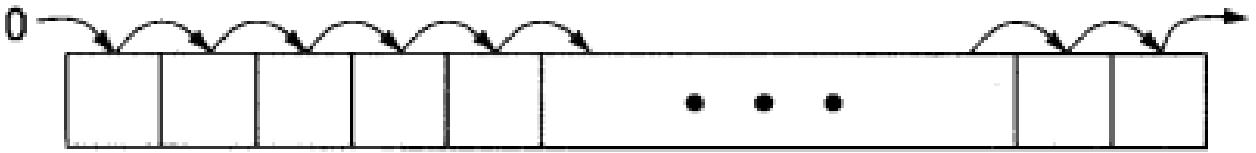
```
(R1) = 10100101
(R2) = 00001111
```

então,

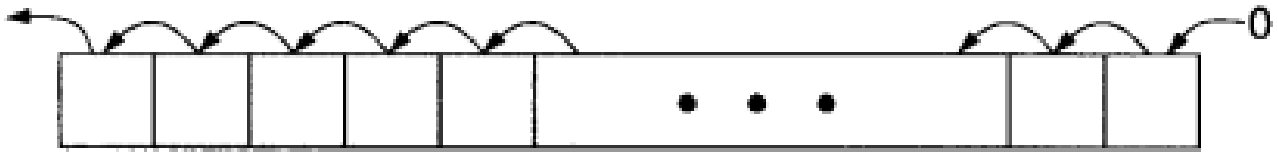
```
(R1) AND (R2) = 00000101
```

TIPOS DE OPERAÇÕES

- 3. OPERAÇÕES LÓGICAS
  - Operações de deslocamento lógico
    - Úteis para isolar campos de bits dentro de uma palavra.
    - EXEMPLO: transmitir caracteres de dados para um dispositivo de E/S (um caracteres de cada vez).



(a) Deslocamento lógico para a direita

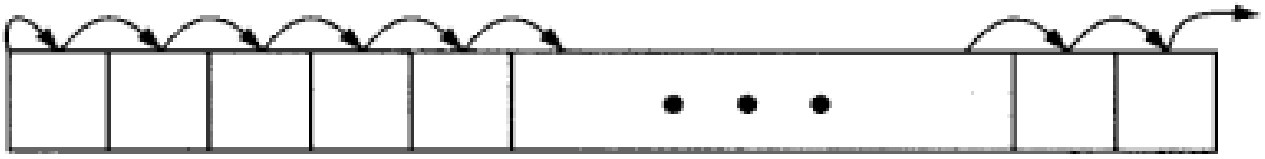


(b) Deslocamento lógico para a esquerda

TIPOS DE OPERAÇÕES

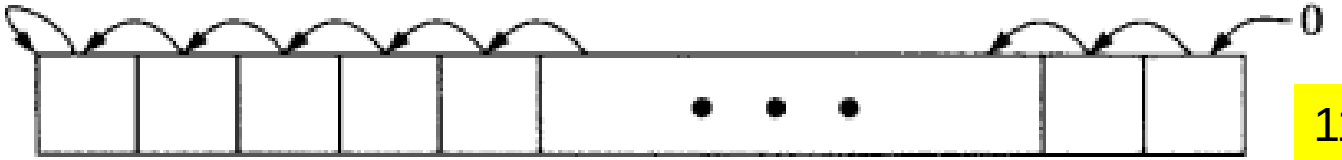
- 3. OPERAÇÕES LÓGICAS
  - Operações de deslocamento aritmético
    - Um deslocamento para esquerda ou para direita corresponde, respectivamente, à multiplicação ou divisão por 2, desde que não ocorra overflow ou underflow.

0110 → 6



0011 → 3

(c) Deslocamento aritmético para a direita

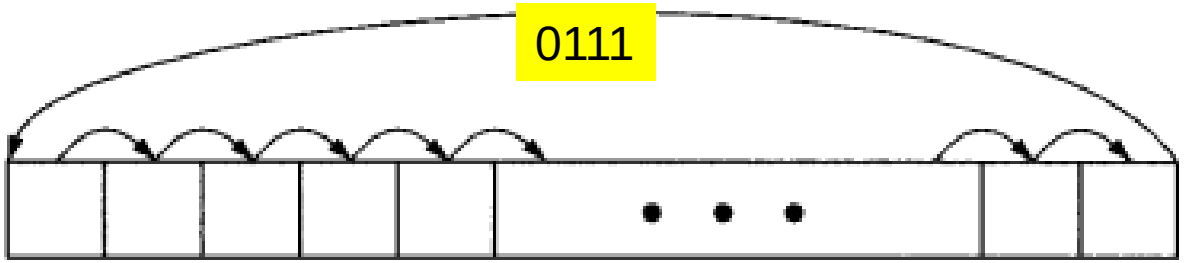


1100 → 12

(d) Deslocamento aritmético para a esquerda

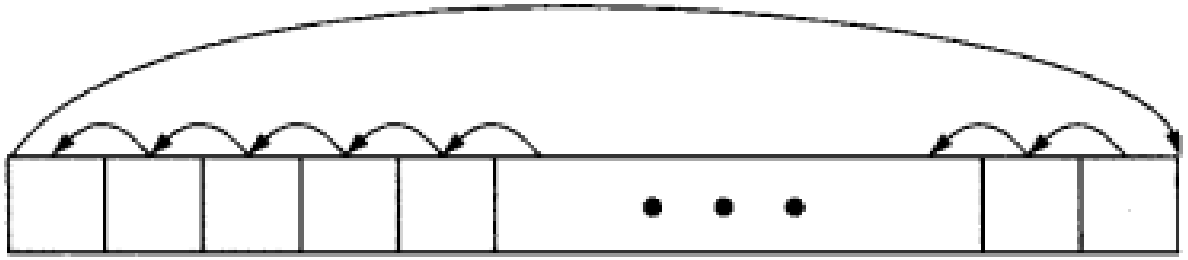
TIPOS DE OPERAÇÕES

- 3. OPERAÇÕES LÓGICAS
  - Operações de rotação
    - Preservam todos os bits sobre as quais uma operação é efetuada



1011

(e) Rotação para a direita



1110

(f) Rotação para a esquerda

TIPOS DE OPERAÇÕES

- 3. OPERAÇÕES LÓGICAS
  - BASEADA EM HAYES, 1988

Tipo	Nome da operação	Descrição
	AND OR NOT (Complemento) Exclusive-OR	Efetua a operação lógica especificada, bit a bit
Operações lógicas	Test	Testa a condição especificada; atualiza códigos de condição (*flags*), de acordo com o resultado
	Compare	Efetua uma comparação lógica ou aritmética de dois ou mais operandos; atualiza códigos de condição (flags), de acordo com o resultado
	Set control variables	Classe de instruções para especificar informação de controle, para fins de proteção, tratamento de interrupção, controle de temporização etc.
	Shift	Deslocamento de operando para a esquerda (direita), introduzindo constantes no final
	Rotate	Rotação circular de operando para a esquerda (direita)

TIPOS DE OPERAÇÕES

- 4. OPERAÇÕES DE CONVERSÃO
  - São aquelas que mudam ou operam sobre o formato de dados. Por exemplo a conversão de um número de decimal para binário.
  - Deve indicar o modo de endereçamento de cada operando.
  - BASEADA EM HAYES, 1988

Tipo	Nome da operação	Descrição
Operações de conversão	Translate	Traduz valores armazenados em uma seção da memória, com base em uma tabela de correspondências
	Convert	Converte o conteúdo de uma palavra de uma representação para outra (por exemplo, decimal empacotado para binário)

TIPOS DE OPERAÇÕES

- 5. OPERAÇÕES DE ENTRADA E SAÍDA
  - Existe uma variedade abordagens:
    - E/S programada
    - E/S mapeada na memória
    - DMA
    - Uso de processadores de E/S

Tipo	Nome da operação	Descrição
Operações de E/S	Read (input)	Transfere dados da porta ou dispositivo de E/S especificado para o destino (por exemplo, memória principal ou registrador de processador)
	Write (output)	Transfere dados da fonte especificada para uma porta ou um dispositivo de E/S
	Start I/O	Transfere instruções para o processador de E/S, para iniciar uma operação de E/S
	Test I/O	Transfere informação de estado do sistema de E/S para o destino especificado

TIPOS DE OPERAÇÕES

- 6. OPERAÇÕES DE CONTROLE DO SISTEMA
  - São as que só podem ser executadas com processador no modo privilegiado (modo KERNEL)
  - São tipicamente são reservadas para uso do sistema operacional

TIPOS DE OPERAÇÕES

- 7. OPERAÇÕES DE TRANSFERÊNCIA DE CONTROLE
  - Alteram a sequência normal de execução das instruções (uma após a outra)
  - Um dos seus operandos contém o endereço da próxima instrução a ser executada
    - Instrução de desvio
    - Instrução de salto
    - Instrução de Chamadas de procedimento.

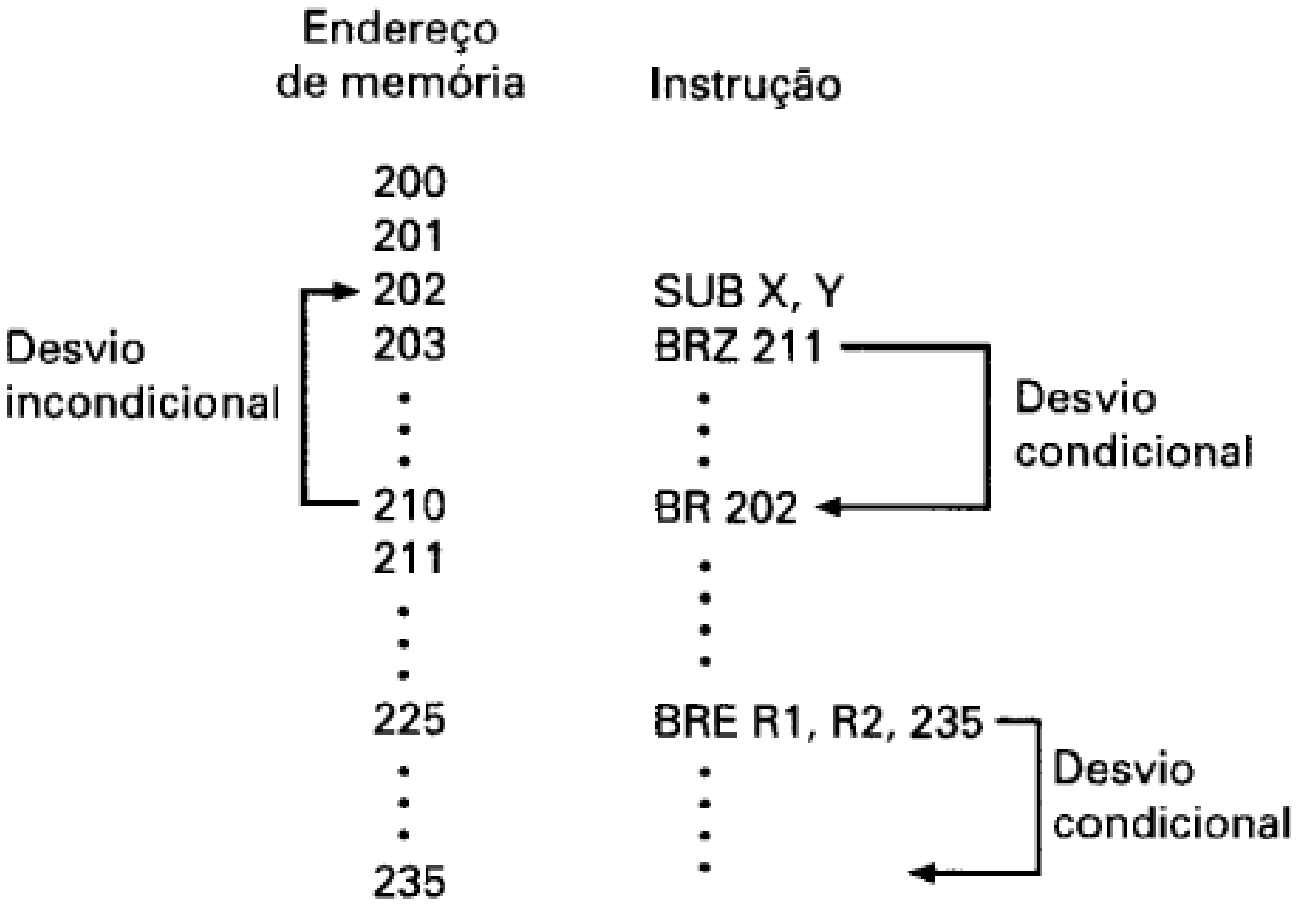
TIPOS DE OPERAÇÕES

- 7. OPERAÇÕES DE TRANSFERÊNCIA DE CONTROLE
- INSTRUÇÃO DE DESVIO
  - INCONDICIONAL
    - BR
  - CONDICIONAL
    - Se uma condição for satisfeita
    - A UCP atualiza o valor do contador de programa (PC) com o endereço especificado no OPERANDO
  - Uma operação pode atualizar um código de condição de 2 bits por exemplo, com os valores 0, POSITIVO, NEGATIVO, OVERFLOW.
    - BRP X = desvia para instrução de endereço X se resultado for POSITIVO
    - BRN X = desvia para instrução de endereço X se resultado for NEGATIVO
    - BRZ X = desvia para instrução de endereço X se resultado for ZERO
    - BRO X = desvia para instrução de endereço X se ocorrer OVERFLOW
  - BRE R1,R2,X
    - Desvia para instrução de endereço X se conteúdo de R1 = conteúdo de R2

PC = CONTADOR DE PROGRAMA.
Contém o endereço da próxima instrução a ser executada

TIPOS DE OPERAÇÕES

- 7. OPERAÇÕES DE TRANSFERÊNCIA DE CONTROLE
- INSTRUÇÃO DE DESVIO



TIPOS DE OPERAÇÕES

- 7. OPERAÇÕES DE TRANSFERÊNCIA DE CONTROLE
- INSTRUÇÃO DE SALTO
  - Incluem um endereço de desvio implícito

```
301
 •
 •
 •
309 ISZ R1
310 BR 301
311
```

R1
CONTÉM O NÚMERO DE INSTRUÇÕES A SEREM EXECUTADAS COM SINAL NEGATIVO

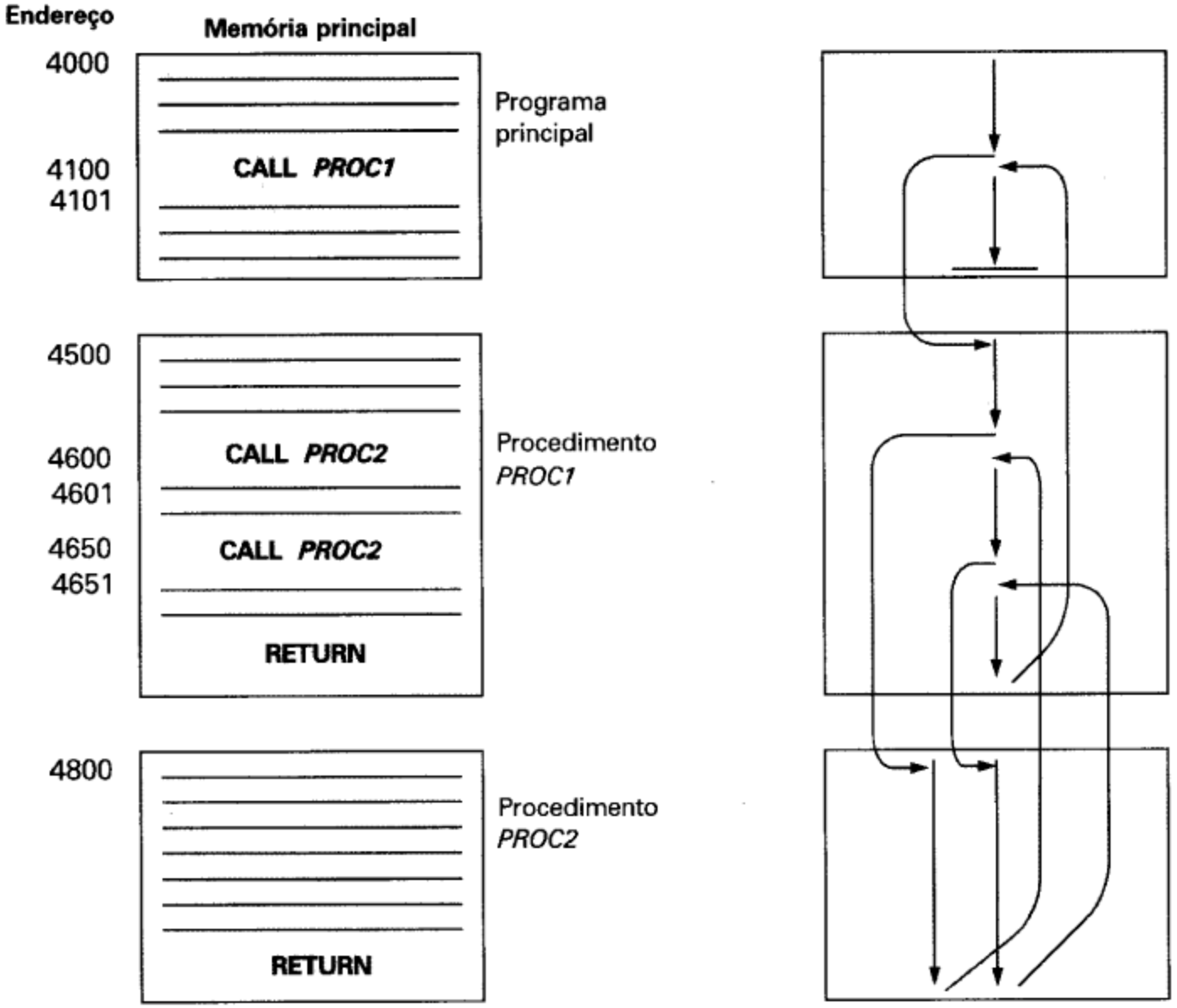
ISZ — *increment-and-skip-if-zero*

ISZ R1
INCREMENTA R1 e EXECUTAR PRÓXIMA INSTRUÇÃO ENQUANTO R1 DIFERENTE DE ZERO
SE R1 = 0 INSTRUÇÃO DE DESVIO OMITIDA E VAI PARA ENDEREÇO 11

OPERAÇÕES DE TRANSFERÊNCIA DE CONTROLE

- 7. OPERAÇÕES DE TRANSFERÊNCIA DE CONTROLE
- INSTRUÇÃO DE CHAMADA DE PROCEDIMENTO
  - PROCEDIMENTO
  - É um subprograma incorporado no programa maior com objetivo de economia e modularidade.
  - Envolve uma instrução de chamada que desvia a instrução corrente para o início do procedimento e uma instrução de retorno que provoca o retorno da execução do procedimento para o endereço em que ocorreu a chamada,
  - Uso de PILHA é uma abordagem para armazenar o endereço de retorno de uma chamada de procedimento.

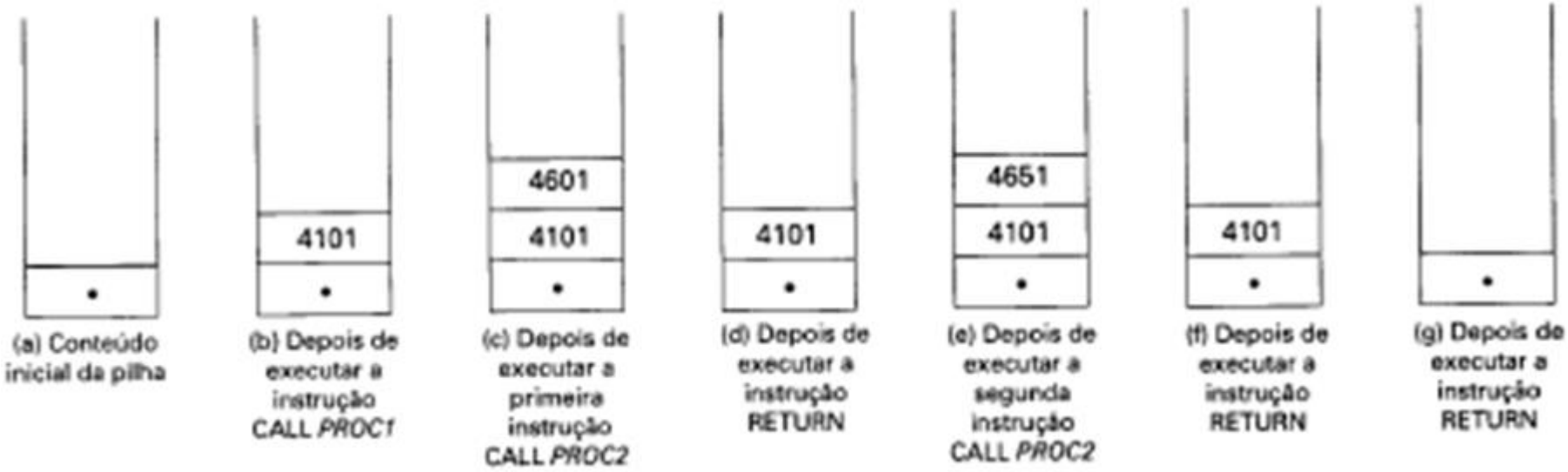
OPERAÇÕES DE TRANSFERÊNCIA DE CONTROLE



(a) Chamadas e retornos de procedimentos

(b) Seqüência de execução

OPERAÇÕES DE TRANSFERÊNCIA DE CONTROLE



TIPOS DE OPERAÇÕES

- 7. OPERAÇÕES DE TRANSFERÊNCIA DE CONTROLE

Tipo	Nome da operação	Descrição
Operações de transferência de controle	Jump (branch)	Desvio incondicional; carrega o PC com o endereço especificado
	Jump conditional	Testa a condição especificada; carrega ou não o PC com o endereço especificado, conforme o resultado do teste
	Jump to subroutine	Armazena informação de controle do programa corrente em uma posição conhecida; desvia para o endereço especificado
	Return	Substitui o conteúdo do PC e de outros registradores com os valores armazenados em uma posição conhecida
	Execute	Busca o operando em uma posição especificada e executa o valor desse operando como uma instrução; não modifica o PC
	Skip	Incrementa o PC (para o endereço da próxima instrução)
	Skip conditional	Testa a condição especificada; desvia ou não com base no resultado do teste
	Halt	Pára a execução do programa
	Wait (hold)	Pára a execução do programa; testa a condição especificada repetidamente; retoma a execução quando a condição é satisfeita
	No operation	Não efetua nenhuma operação e continua a execução do programa