

Aula 7

Professores:

Lúcia M. A. Drummond
Simone de Lima Martins

Conteúdo:

Representação de instruções

- Introdução
- Quantidade de operandos
- Modos de endereçamento

Introdução

- Conjunto de instruções: instruções que o processador pode realizar diretamente
- Exemplo de instrução de máquina:

Instrução de máquina de um operando

Cód. de Operação	Operando
------------------	----------

Cód. de Operação - indica o tipo da operação a ser realizada
Operando - endereço do dado

(Fig. 8.1 do livro texto)

Introdução

- Processadores projetados para realizar instruções muito simples
- Código de operação:
 - código que após ser decodificado durante o ciclo de execução permitirá que a UC emita os sinais necessários para se efetivar a seqüência de passos de realização da operação indicada
 - Quantidade de bits neste campo define o limite máximo de instruções que o processador pode executar
 - **Exemplo:** código de 6 bits, processador pode executar 64 instruções

Introdução

- Operando: contém informação sobre o dado - tipo da informação sobre o dado - seu valor ou o endereço de memória onde localizá-lo. Podem existir um ou mais campos.
- Exemplo:

C. Op.	Op. 1	Op. 2	Op. 3
--------	-------	-------	-------

ADD Op. 1, Op. 2, Op. 3

(a) Instrução de 3 operandos (com memória)

C. Op.	Op. 1	Op. 2
--------	-------	-------

MOVE Op. 1, Op. 2

(b) Instrução de 2 operandos (com memória)

C. Op.	R	Op.
--------	---	-----

ADC R, Op.

(c) Instrução de 2 operandos, sendo um registrador

C. Op.	Op.
--------	-----

JCXZ Op.

(d) Instrução de 1 operando (desvio para memória)

(Fig. 8.2 do livro texto)

Introdução

- Representação de Instruções:
 - Quantidade de operandos
 - Modo de interpretação do valor armazenado no campo operando (modo de endereçamento do dado)

Quantidade de Operandos

- Computadores com conjunto de instruções dos mais variados tipos e formatos
- Exemplo do formato de instrução do sistema SEAC de 1949.

C. Op.	Operando 1	Operando 2	Operando 3	End. próx. instrução
--------	------------	------------	------------	----------------------

Operando n - endereço do dado ou o próprio valor do dado

(Fig. 8.3 do livro texto)

Quantidade de Operandos

- Computador com memória de 2K células e código de operação de 6 bits
- Endereço de 11 bits
- $4(\text{operandos}) \times 11(\text{bits de endereço}) + 6(\text{C. Op}) = 50 \text{ bits}$
- Exemplo: ADD X,Y,Z,P
- $(Z) \leftarrow (X) + (Y)$, onde P é o endereço da próxima instrução
- Essa instrução permitiria a execução de $C = A + B$
- Em linguagem Assembly: ADD A,B,C,P

Quantidade de Operandos

Vantagens de muitos operandos:

- Completeza: instrução possui todos os operandos, dispensando até instruções de desvio incondicional
- Menor quantidade de instruções em um programa

Desvantagens:

- Grande ocupação de espaço de memória
- Nem toda instrução precisa de 3 operandos (Ex: Load, desvio)

Quantidade de Operandos

- Tamanho de instruções:
 - Tamanho da memória
 - Tamanho e organização das células da MP
 - Velocidade de acesso
 - Organização do barramento de dados
- Economia de espaço x conjunto completo de instruções
- RISC
- Introdução do registrador CI (PC - program counter)

Quantidade de Operandos

Instruções com 3 operandos

Formato Básico:

C. Op.	Operando 1	Operando 2	Operando 3
--------	------------	------------	------------

(Fig. 8.4 do livro texto)

As **instruções aritméticas** podem ser do tipo:

ADD A,B,X	$(X) \leftarrow (A) + (B)$
SUB A,B,X	$(X) \leftarrow (A) - (B)$
MPY A,B,X	$(X) \leftarrow (A) \times (B)$
DIV A,B,X	$(X) \leftarrow (A) / (B)$

Quantidade de Operandos

Instruções com 3 operandos

Exemplo:

$$X = A * (B + C * D - E / F)$$

Assembly correspondente:

MPY C,D,T1	; multiplicação de C e D, resultado em T1
DIV E,F,T2	; divisão de E por F, resultado em T2
ADD B,T1,X	; soma de B com T1, resultado em X
SUB X,T2,X	; subtração entre X e T2, resultado em X
MPY A,X,X	; multiplicação de A por X

Quantidade de Operandos

Instruções com 3 operandos

Considerações:

- Operandos com endereços iguais - desperdício de memória
- Número de instruções igual ao número de operações
- Raramente encontradas em conjunto de instruções dos processadores atuais
- Pelo exemplo anterior, a maioria das instruções tem 2 endereços (o outro é repetido)

Quantidade de Operandos

Instruções com 2 operandos

ADD Op1, Op2	$(Op1) \leftarrow (Op1) + (Op2)$
SUB Op1, Op2	$(Op1) \leftarrow (Op1) - (Op2)$
MPY Op1, Op2	$(Op1) \leftarrow (Op1) \times (Op2)$
DIV Op1, Op2	$(Op1) \leftarrow (Op1) / (Op2)$

O conteúdo de Op1 será destruído com o armazenamento do resultado da operação.

Salvamento de variável:

MOVE A,B $(A) \leftarrow (B)$

Quantidade de Operandos

Instruções com 2 operandos

Exemplo:

$$X = A * (B + C * D - E / F)$$

Assembly correspondente:

MPY C,D	; multiplicação de C e D, resultado em C
DIV E,F	; divisão de E por F, resultado em E
ADD B,C	; soma de B com C, resultado em B
SUB B,E	; subtração de B e E, resultado em B
MPY A,B	; multiplicação de A por B, resultado em A
MOVE X,A	; armazenamento do resultado final, A, em X

Quantidade de Operandos

Instruções com 1 operando

ADD Op	$ACC \leftarrow ACC + (Op)$
SUB Op	$ACC \leftarrow ACC - (Op)$
MPY Op	$ACC \leftarrow ACC * (Op)$
DIV Op	$ACC \leftarrow ACC / (Op)$

O acumulador (ACC) é empregado como operando implícito e para guardar o resultado da operação.

Duas novas instruções:

LDA Op	$ACC \leftarrow (Op)$
STA Op	$(Op) \leftarrow ACC$

Quantidade de Operandos

Instruções com 1 operando

Para não destruir valores de variáveis:

$$X = A*(B+C*D-E/F)$$

Assembly correspondente:

```
LDA C
MPY D
STA X
LDA E
DIV F
STA T1
LDA B
ADD X
SUB T1
MPY A
STA X
```


Quantidade de Operandos

Tamanho e Consumo de Tempo de Execução de Instruções de 3, de 2 e de 1 Operando

Instrução de 3 operandos - C. Op. = 8 bits + 3 operandos de 20 bits cada um = 68 bits

Ciclos de memória = 4 (um para buscar a instrução e
3 para os operandos)

Instrução de 2 operandos - C. Op. = 8 bits + 2 operandos de 20 bits cada um = 48 bits

Ciclos de memória = 4 (um para buscar a instrução e
3 para os operandos)

Instrução de 1 operando - C. Op. = 8 bits + 1 operando de 20 bits = 28 bits

Ciclos de memória = 2 (um para buscar a instrução e
1 para o operando)

(Tabela 8.1 do livro texto)

Quantidade de Operandos

Programas Assembly para Solucionar a Equação: $X = A * (B + C * D - E / F)$

Com instruções de 3 operandos	Com instruções de 2 operandos (sem salvamento)	Com instruções de 2 operandos (com salvamento)	Com instruções de 1 operando
MPY C, D, T1 DIV E, F, T2 ADD B, T1, X SUB X, T2, X MPY A, X, X	MPY C, D DIV E, F ADD B, C SUB B, E MPY A, B MOVE X, A	MOVE X, C MPY X, D MOVE T1, E DIV T1, F ADD X, B SUB X, T1 MPY X, A	LDA C MPY D STA X LDA E DIV F STA T1 LDA B ADD X SUB T1 MPY A STA X
Espaço: 340 bits Tempo: 20 acessos	Espaço: 288 bits Tempo: 24 acessos	Espaço: 336 bits Tempo: 28 acessos	Espaço: 308 bits Tempo: 22 acessos

(Tabela 8.2 do livro texto)

Quantidade de Operandos

Considerações:

- Instruções de poucos operandos ocupam menos espaço de memória e tornam o projeto do processador mais simples
- O programa gerado com poucos operandos pode ser maior
- Instruções de 1 operando são simples e baratas. Mas o uso de 1 registrador pode reduzir a flexibilidade e velocidade de processamento
- Instruções com mais de 1 operando podem usar endereços de memória e registradores em seus formatos

Modos de Endereçamento

Considerações:

- O endereçamento de instrução é realizado pelo contador de instruções (CI) e o ciclo de instrução é iniciado pela transferência da instrução para o registrador de instrução (RI)
- Toda instrução consiste em uma ordem codificada para a UCP realizar uma operação sobre um dado (valor numérico, caractere alfabético ou endereço)
- Localização dos dados pode estar explicitamente indicada na própria instrução através de *CAMPO DE OPERANDO*, ou implicitamente (armazenado no ACC)

Modos de Endereçamento

- Existem várias formas de localizar o dado: *modos de endereçamento*
 - Imediato
 - Direto
 - Indireto
 - Por registrador
 - Indexado
 - Base mais deslocamento

Modos de Endereçamento

Modo Imediato

- Método mais simples: indicar o próprio valor no campo operando
- O dado é transferido da memória juntamente com a instrução
- Útil para inicialização de contadores, operação com constantes

Modos de Endereçamento

Modo Imediato

Desvantagens:

- Limitação do campo de operando reduz o valor máximo do dado a ser manipulado
- Programas repetidamente executados com valores diferentes de variáveis em cada execução, acarretariam o trabalho de alteração do campo operando a cada execução

Modos de Endereçamento

Modo Imediato

Exemplo 8.1:

C. Op. / Operando
(4 bits) / (8bits)

JMP Op.

C. Op. = 1010 = hexadecimal A

Instrução: 101000110101 ou A35 (C. Op. = A e Operando = 35)

Armazenar o valor 35 no CI

Modos de Endereçamento

Modo Imediato

Exemplo 8.2:

C. Op. / R / Operando
(4 bits) / (4 bits) / (8 bits)

MOV R, Op.

C. Op. = 0101 = hexadecimal 5

Instrução: 0101001100000111 ou 5307 (C. Op. = 5, R = 3 e Operando = 07)
Armazenar o valor 07 no registrador 3 (R3)

Modos de Endereçamento

Modo Direto

Operando indica o endereço de memória onde se localiza o dado

Considerações:

- Modo simples - requer apenas 1 acesso à memória
- Mais lento que o modo imediato
- Com o aumento do tamanho de memória, são necessários muitos bits para endereçamento direto

Modos de Endereçamento

Modo Direto

Exemplo 8.3:

a) C. Op. / Operando
(4 bits) / (8 bits)

LDA Op.

C. Op. = 7

Instrução: 73B (C. Op. = 7 e Operando = 3B)

Após a execução da instrução, o ACC conterà o valor 05A

	MP
	//
	//
	//
3B	05A
	//
	//
	//
5C	103 15D
	//
	//

Modos de Endereçamento

Modo Direto

Exemplo 8.3:

b) C. Op. / Op.1 / Op.2
(4 bits) / (8 bits) / (8 bits)

ADD Op.1, Op.2

C. Op. = B

Instrução: B5C3B (C. Op. = B, Op.1 = 5C e Op. 2 = 3B)

Após a execução da instrução, o endereço 5C conterà o valor 15D

	MP
	//
	//
	//
3B	05A
	//
	//
	//
5C	103 15D
	//
	//

Modos de Endereçamento

Modo Indireto

- O operando representa o endereço de uma célula, cujo conteúdo é outro endereço de célula
- Requer dois acessos à memória para se obter o dado
- Útil para acessar dados armazenados contiguamente na memória, como vetores.

Modos de Endereçamento

Modo Indireto

Exemplo 8.4:

C. Op. / Operando
(4 bits) / (8 bits)

LDA Op.

C. Op. = 4

Instrução: 474 (C. Op. = 4, Op.= 74)

Após a execução da instrução, o valor 1A4 estará armazenado no ACC

	MP
74	05D
	''
5D	1A4
	''

Modos de Endereçamento

Observações

- 1) Há dois métodos de indicação do modo de endereçamento das instruções:
 - O código de operação estabelece não só o tipo da instrução como também o modo de endereçamento
 - A instrução possui um campo específico para indicar o modo de endereçamento

C. Op.		Modo end.		Operando	
4		2		10	
				00 - Direto (DI) 01 - Imediato (IM) 10 - Indireto (ID) 11 - Não usado	
C. Op. = A	LDA IM	———	A13B	101000100111011	
	LDA DI	———	A43B	101010000111011	
	LDA ID	———	A83B	101100000111011	

(Fig. 8.6 do livro texto)

Modos de Endereçamento

Observações

Modo de endereçamento	Definição	Vantagens	Desvantagens
Imediato	O campo operando contém o dado.	Rapidez na execução da instrução.	Limitação do tamanho do dado. Inadequado para o uso com dados de valor variável.
Direto	O campo operando contém o endereço do dado.	Flexibilidade no acesso a variáveis de valor diferente em cada execução do programa.	Perda de tempo, se o dado é uma constante.
Indireto	O campo operando contém o endereço do endereço do dado.	Manuseio de vetores (quando o modo indexado não está disponível). Uso como "ponteiro".	Muitos acessos a MP para execução.

(Fig. 8.7 do livro texto)

Modos de Endereçamento

Observações

2) Em relação aos 3 modos de endereçamento temos:

- O modo imediato não requer acesso à memória principal para buscar o dado, o modo direto requer um acesso, o modo indireto requer dois acessos
- Em relação ao tempo, modo imediato é mais rápido, seguido de modo direto e por último o modo indireto

Modos de Endereçamento

Endereçamento por Registrador

- Semelhante aos modos direto e indireto, mas o endereço de memória é substituído por um endereço de registrador da UCP
- Menor número de bits na instrução
- Dado armazenado em um meio mais rápido

Modos de Endereçamento

Endereçamento por Registrador

Comparação:

Do I = 1 **to** 100

Read A,B

$X = A + B$

End

Modos de Endereçamento

Endereçamento por Registrador

Modo de endereçamento direto:

GET L	; ler valor do "loop" (no exemplo o valor é igual a 100)
LDI 0	; $ACC \leftarrow 0$
SUBM L	; $ACC \leftarrow ACC - (L)$ (no exemplo valor é 100)
In STA I	; $(I) \leftarrow ACC$ (inicialmente $I = -100$)
JZ Fim	; se $ACC = 0$ vá para Fim
GET A	; ler valor do dado para o endereço A
GET B	; ler valor do dado para o endereço B
LDA A	; $ACC \leftarrow (A)$
ADD B	; $ACC \leftarrow ACC + (B)$
STR X	; $(X) \leftarrow ACC$
LDA I	; $ACC \leftarrow (I)$
INC	; $ACC \leftarrow ACC + 1$ (no exemplo estamos fazendo $I+1$)
JMP In	; vá para In
Fim HLT	; parar

Modos de Endereçamento

Endereçamento por Registrador

Modo de endereçamento direto:

Gastam-se 200 ciclos de memória apenas para carregar o acumulador com A e armazenar o valor do acumulador em X

Modos de Endereçamento

Endereçamento por Registrador

Modo de endereçamento por registrador:

LDI R1, 0	; $R1 \leftarrow 0$
LDI R2, 100	; $R2 \leftarrow 100$
In SUBR R1, R2	; $(R1) \leftarrow (R1) - (R2)$
JZ R, Fim	; se $R1 = 0$ vá para Fim
GET A	; ler valor do dado para o endereço A
GET B	; ler valor do dado para o endereço B
LDA A	; $ACC \leftarrow (A)$
ADD B	; $ACC \leftarrow ACC + (B)$
STR X	; $(X) \leftarrow ACC$
INC R1	; $(R1) \leftarrow (R1) + 1$ (no exemplo estamos fazendo $I+1$)
JMP In	; vá para In
Fim HLT	; parar

Modos de Endereçamento

Endereçamento por Registrador

As instruções:

```
LDI R1,1,  
LDI R2, 100  
SUBR R1, R2  
INC R1
```

Não acessam a memória principal

Não há vantagem em fazer $MP \leftarrow R \leftarrow UAL$ nos casos dos valores A e B (não reduz ciclos de memória)

Modos de Endereçamento

Endereçamento por Registrador

Economia de bits

Instruções com uso de registradores:

C.Op. / R1 / R2

8 bits / 4 bits / 4 bits = 16 bits

Instruções com acesso a 64K células de MP:

C. Op. / Op. 1 / Op.2

8 bits / 16 bits / 16 bits = 40 bits

Modos de Endereçamento

Endereçamento por Registrador

Há duas maneiras de empregar o modo de endereçamento por registrador:

- Modo por **registrador direto**
- Modo por **registrador indireto**

Ocorre dificuldade em se definir quais dados serão armazenados em registradores e quais permanecerão em MP

Modos de Endereçamento

Endereçamento por Registrador

Dois tipos de instruções que usam o modo registrador:

C. Op.	R1	R2
(a)		
(a) ADD	R1, R2	(R1) ← (R1) + (R2)
(b) ADR	R, Op.	(R) ← (R) + (Op.) ou (R) (R) ← (R) + Op.
(a) LDR	R1, R2	(R1) ← ((R2))

(Fig. 8.8 do livro texto)

Modos de Endereçamento

Modo Indexado

- Permite manipular endereços para facilitar acesso a vetores
- O endereço do dado (elemento do array) relaciona-se com o seu índice
- O endereço do dado é a soma do valor do campo operando (fixo) e de um valor armazenado em um dos registradores da UCP (registrador índice)

Modos de Endereçamento

Modo Indexado

Do I = 1 To 100

$C(I) = A(I) + B(I)$

Modos de Endereçamento

Modo Indexado

Usando o Modo Direto sem alterar os bits que descrevem as instruções

```
LDA A(1)
ADD B(1)
STA C(1)
LDA A(2)
ADD B(2)
STA C(2)
-----
-----
LDA A(100)
ADD B(100)
STA C(100)
HLT
```

(Fig. 8.9 do livro texto)

Instruções para cada uma das 100 operações de soma a serem executadas

Modos de Endereçamento

Modo Indexado

Usando o Modo Direto com alteração dinâmica do conteúdo de instruções

Programa Assembly	Programa em linguagem de máquina
T LDA A(1)	11A00
1 ADD B(1)	21A64
2 STA C(1)	31AC8
INC T	8103A
INC 1	8103B
INC 2	8103C
DCR N	919FF
JNZ T	D103A
END	F0000

(Fig. 8.10 do livro texto)

Modos de Endereçamento

Modo Indexado

Usando o Modo Direto com alteração dinâmica do conteúdo de instruções

Formato da instrução			MP
C. Op.	Operando	103A	11A00
		103B	21A64
4 bits	16 bits	Cont. do trecho do progr.	
LDA Op. - C. Op. = 1		(N) 19FF	
ADD Op. - C. Op. = 2		A(1) 1A00	
STA Op. - C. Op. = 3		A(2) 1A01	
-----		A(3) 1A02	
INC Op. - C. Op. = 8		" "	
DCR Op. - C. Op. = 9			

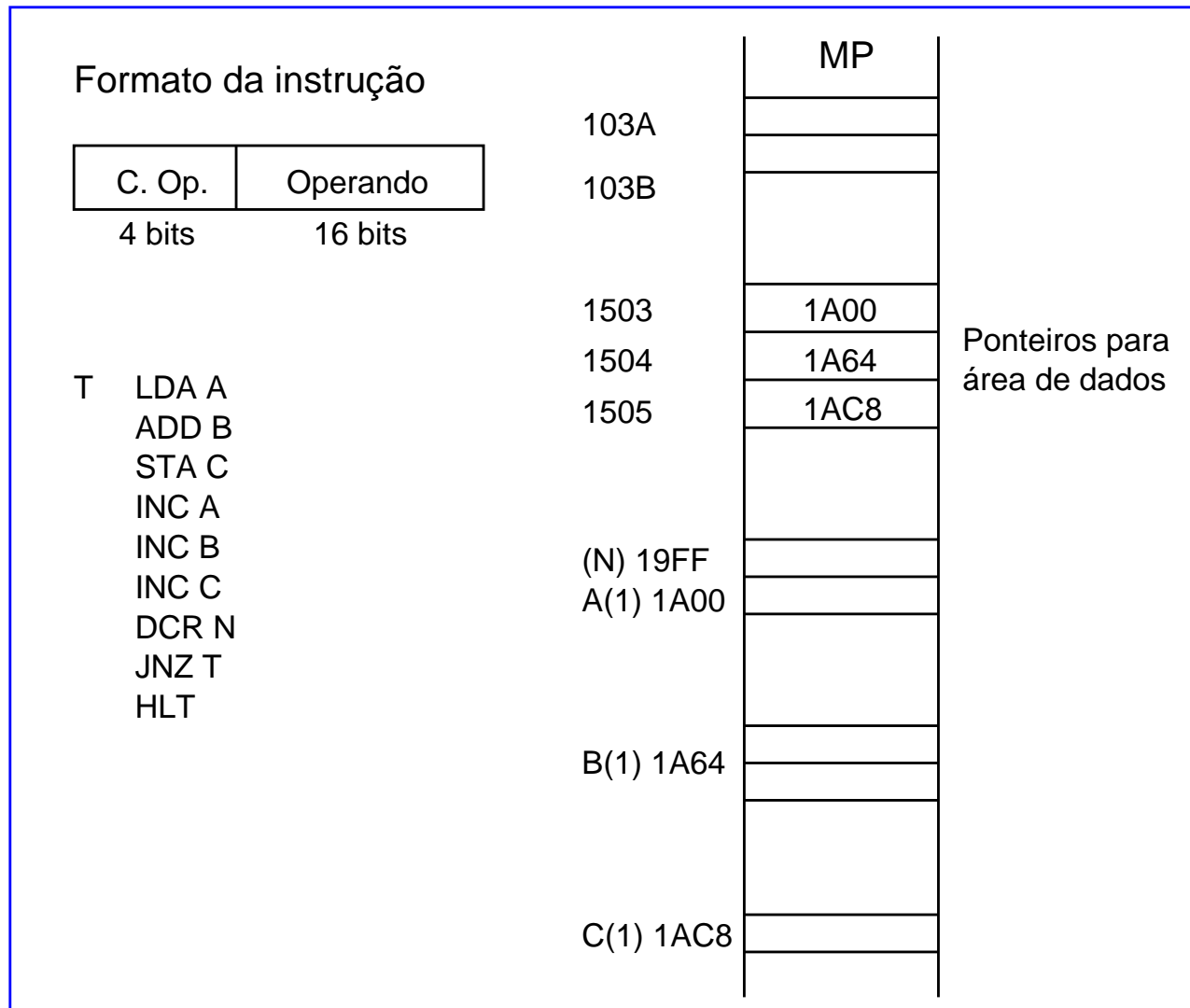
JMP Op. - C. Op. = C		B(1) 1A64	
JNZ Op. - C. Op. = D		B(2) 1A65	
HLT - C. Op. = F		" "	
		C(1) 1AC8	
		C(2) 1AC9	
		" "	

(Fig. 8.11 do livro texto)

Modos de Endereçamento

Modo Indexado

Usando o Modo Indireto



(Fig. 8.12 do livro texto)

Modos de Endereçamento

Modo Indexado

Usando o Modo Indexado

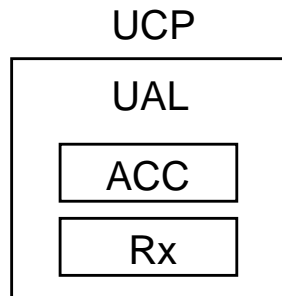
```

MVI (R4), 1
MVI (R2), 100
T LDA (R4), 19FF
ADD (R4), 1A63
STA (R4), 1AC7
INC (R4)
DCR (R2)
JZR (R2), T
HLT
```

(Fig. 8.13 do livro texto)

Modos de Endereçamento

Modo Indexado



Assumimos que um dos 16 reg. da UCP será usado como Rx. Seu endereço consta do campo R da instrução.

C. Op.	R	Operando
4 bits	4	16 bits

0 - MVI Rx, Op.	$(Rx) \leftarrow Op.$
1 - LDA Rx, Op.	$ACC \leftarrow (Op. + (Rx))$
2 - STA Rx, Op.	$(Op. + (Rx)) \leftarrow ACC$
3 - ADD Rx, Op.	$ACC \leftarrow ACC + (Op. + (Rx))$
4 - JMP Op.	$CI \leftarrow Op.$
5 - JZR Rx, Op.	$CI \leftarrow Op., \text{ se } (Rx) = 0$
6 - DCR Rx	$(Rx) \leftarrow (Rx) - 1$
7 - INC Rx	$(Rx) \leftarrow (Rx) + 1$
F - END	Parar

	MP
103A	
	"
	"
	"
A(1) 1A00	
	"
B(1) 1A64	
	"
	"
	"
C(1) 1AC8	

(Fig. 8.14 do livro texto)

Modos de Endereçamento

Modo Base Mais Deslocamento

- Tem características semelhantes ao modo indexado -
 $\text{endereço} = \text{deslocamento} + \text{registrador base}$
- Registrador base fixo e variação do deslocamento
- Redução do tamanho da instrução

Modos de Endereçamento

Modo Base Mais Deslocamento

Sua escolha decorre de:

- Durante a execução de uma grande quantidade de programas, as referências as células de memória, normalmente são seqüenciais
- A maioria dos programas ocupa um pequeno espaço da MP disponível
- Redução do tamanho da instrução

Modos de Endereçamento

Modo Base Mais Deslocamento

Exemplo:

- Campo de registrador base de 4 bits (16 registradores) e campo de deslocamento de 12 bits = 16 bits
- Processador pode endereçar 16 M – 24 bits
- Por registrador base pode-se endereçar 4096 bytes (4 K)
- Economia de 8 bits

Modos de Endereçamento

Modo Base Mais Deslocamento

Diferenças com o modo indexado:

- Indexação é empregada quando se deseja acessar diferentes dados com alteração do endereço por incremento ou decremento do registrador índice
- Quando a modificação do endereço é realizada para relocação do programa, basta uma única alteração no registrador base

Modos de Endereçamento

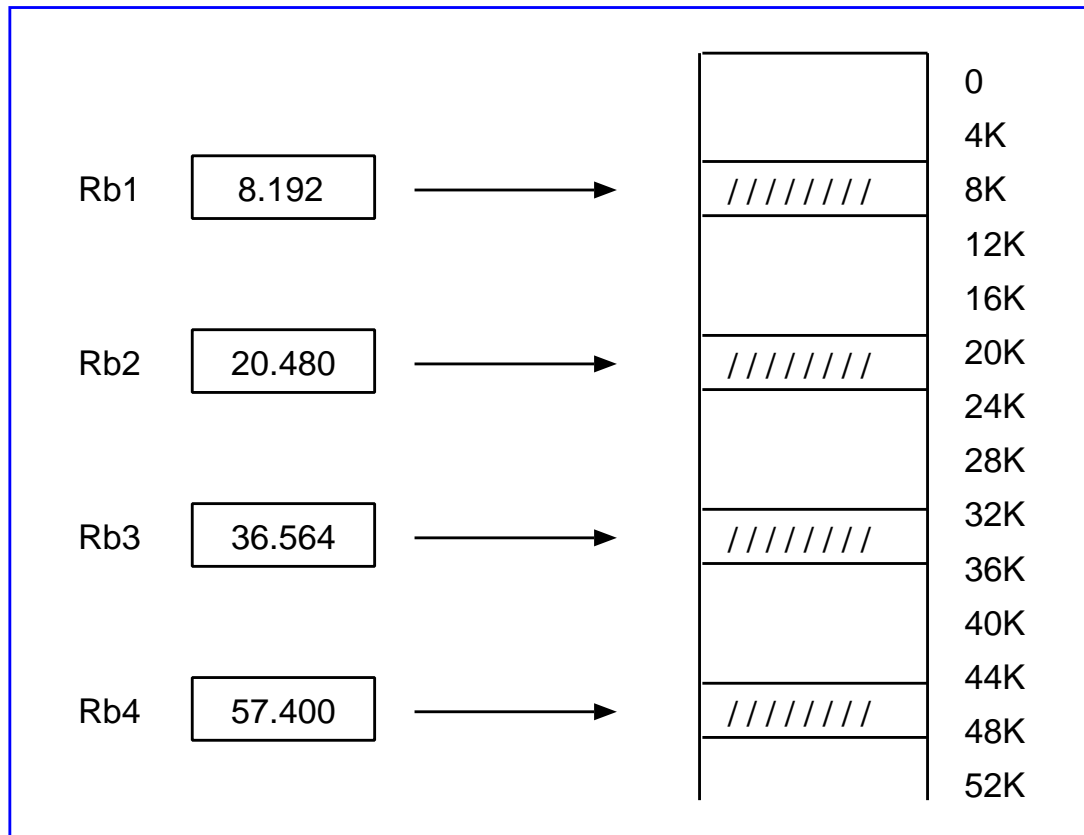
Modo Base Mais Deslocamento

Diferenças com o modo indexado:

- Vários dados são acessados com diversos valores de registrador índice e um único valor de campo operando
- Vários dados são acessados com único valor de registrador base e valores diferentes no campo deslocamento

Modos de Endereçamento

Modo Base Mais Deslocamento



(Fig. 8.15 do livro texto)

Representação de Instruções

Exercícios

Todos os exercícios do Capítulo 8 do Livro Texto