

Aula 4

Professores:

Lúcia M. A. Drummond
Simone de Lima Martins

Conteúdo:

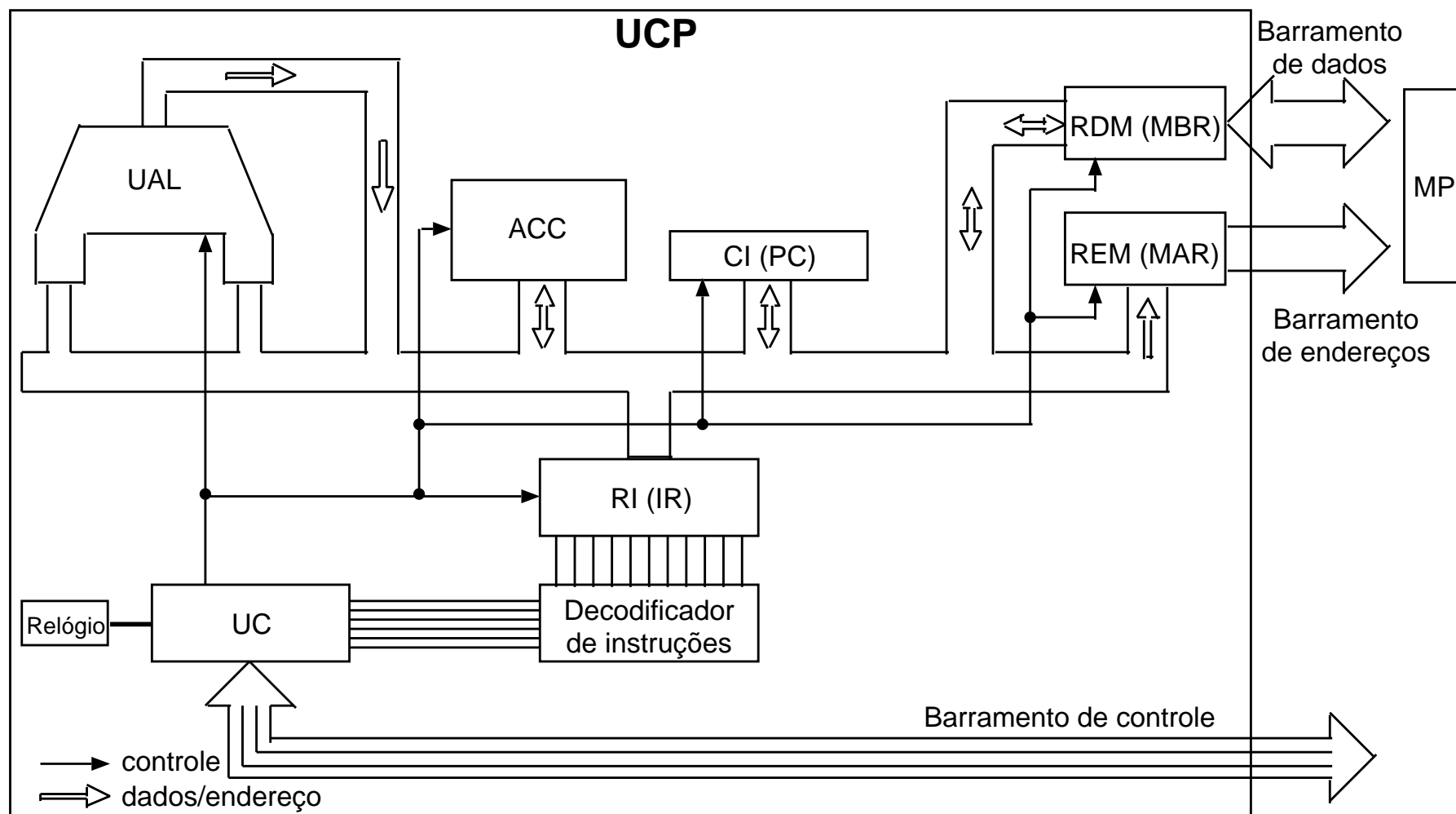
Unidade Central de Processamento 2

- Funcionamento da UCP: O ciclo de instrução
- Linguagem de montagem
- Unidade Aritmética e Lógica

Funcionamento da UCP: O ciclo de instrução

- A base do projeto de uma UCP é a escolha do conjunto de instruções
- Os componentes da arquitetura e sua organização são definidos para interpretar e executar as instruções deste conjunto
- Um subsistema UCP/MP hipotético será utilizado para mostrar as etapas requeridas para a execução de instruções

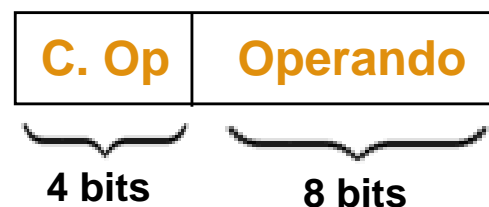
Funcionamento da UCP: O ciclo de instrução



(Fig. 6.3 do livro texto)

Funcionamento da UCP: O ciclo de instrução

- Processador hipotético
 - Palavra: 12 bits
 - Endereços: 8 bits ($2^8 = 256$ células de memória)
 - Células de 12 bits
 - A UCP possui um registrador de dados (ACC) com 12 bits de tamanho, o CI e REM com 8 bits cada um e o RDM com 12 bits
 - Formato das instruções



- Operando indica endereço na memória, exceto em instrução de desvio

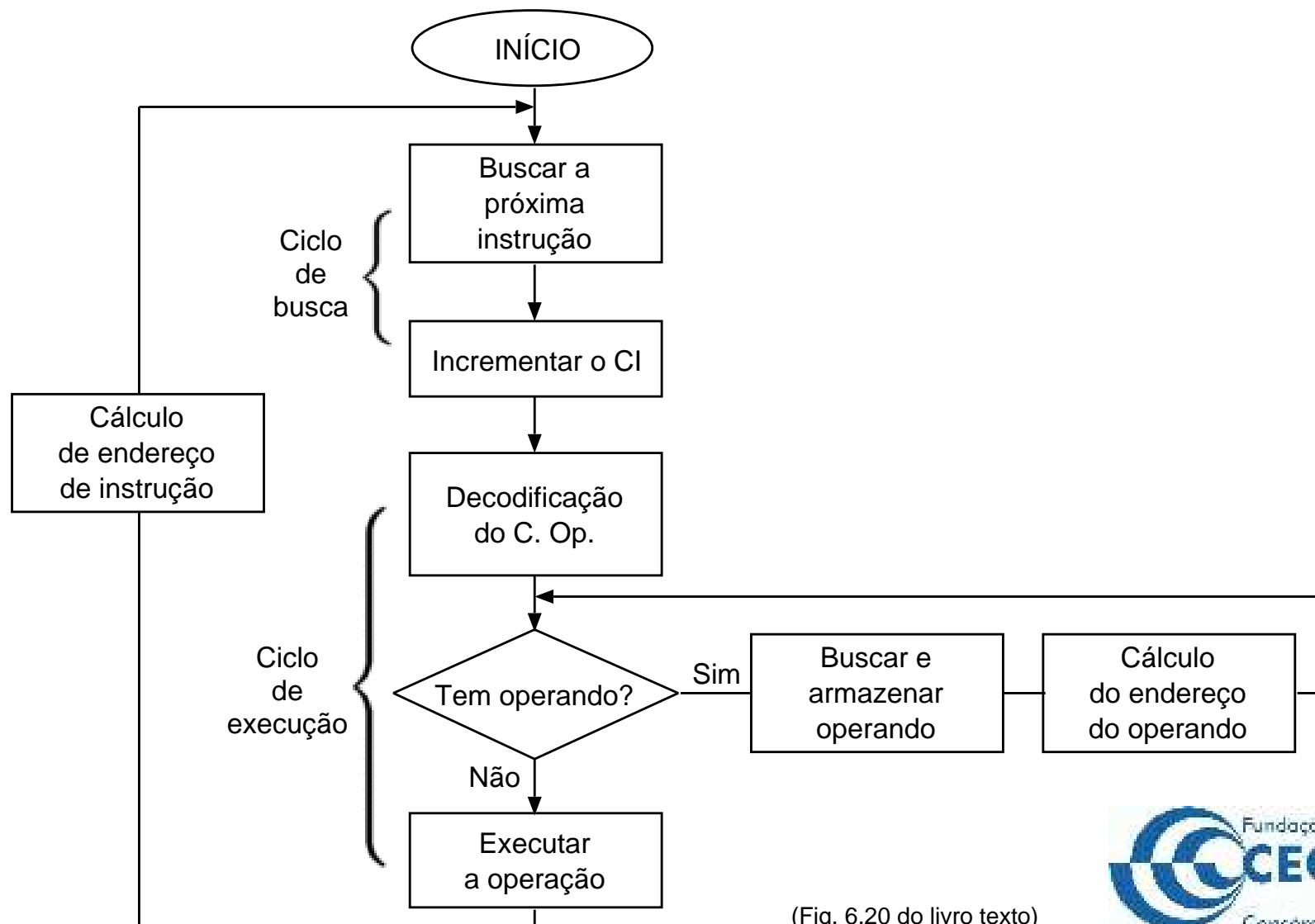
Funcionamento da UCP: O ciclo de instrução

- Instruções disponíveis

C. Op.	Sigla	Descrição
0	HLT	Parar a execução do programa
1	LDA Op.	$ACC \leftarrow (Op.)$, Load
2	STR Op.	$(Op.) \leftarrow ACC$, Store
3	ADD Op.	$ACC \leftarrow ACC + (Op.)$
4	SUB Op.	$ACC \leftarrow ACC - (Op.)$
5	JZ Op.	Se $ACC = 0$, então $CI \leftarrow Op.$
6	JP Op.	Se $ACC > 0$, então $CI \leftarrow Op.$
7	JN Op.	Se $ACC < 0$, então $CI \leftarrow Op.$
8	JMP Op.	$CI \leftarrow Op.$
9	GET Op.	Ler dado da porta de entrada e armazená-lo em (Op.)
A	PRT Op.	Colocar dado armazenado em (Op.) na porta refer. à impressora

Funcionamento da UCP: O ciclo de instrução

- Fluxograma de um ciclo de instrução



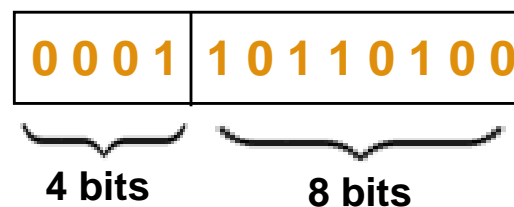
(Fig. 6.20 do livro texto)

Funcionamento da UCP: O ciclo de instrução

- Execução de duas instruções armazenadas na memória
 - LDA Op armazenada no endereço 2
 - ADD Op armazenada no endereço 3

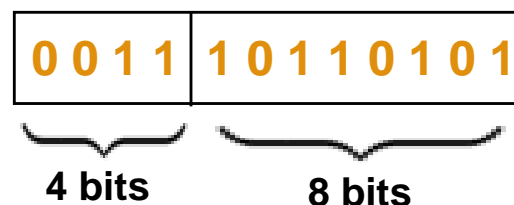
Funcionamento da UCP: O ciclo de instrução

- **LDA 180**
 - Coloca o conteúdo do endereço de memória 180 no acumulador
 - Código de operação = $1_{16} = 0001$
 - Endereço de memória = $180_{10} = 10110100$
 - Instrução = $1B4_{16}$



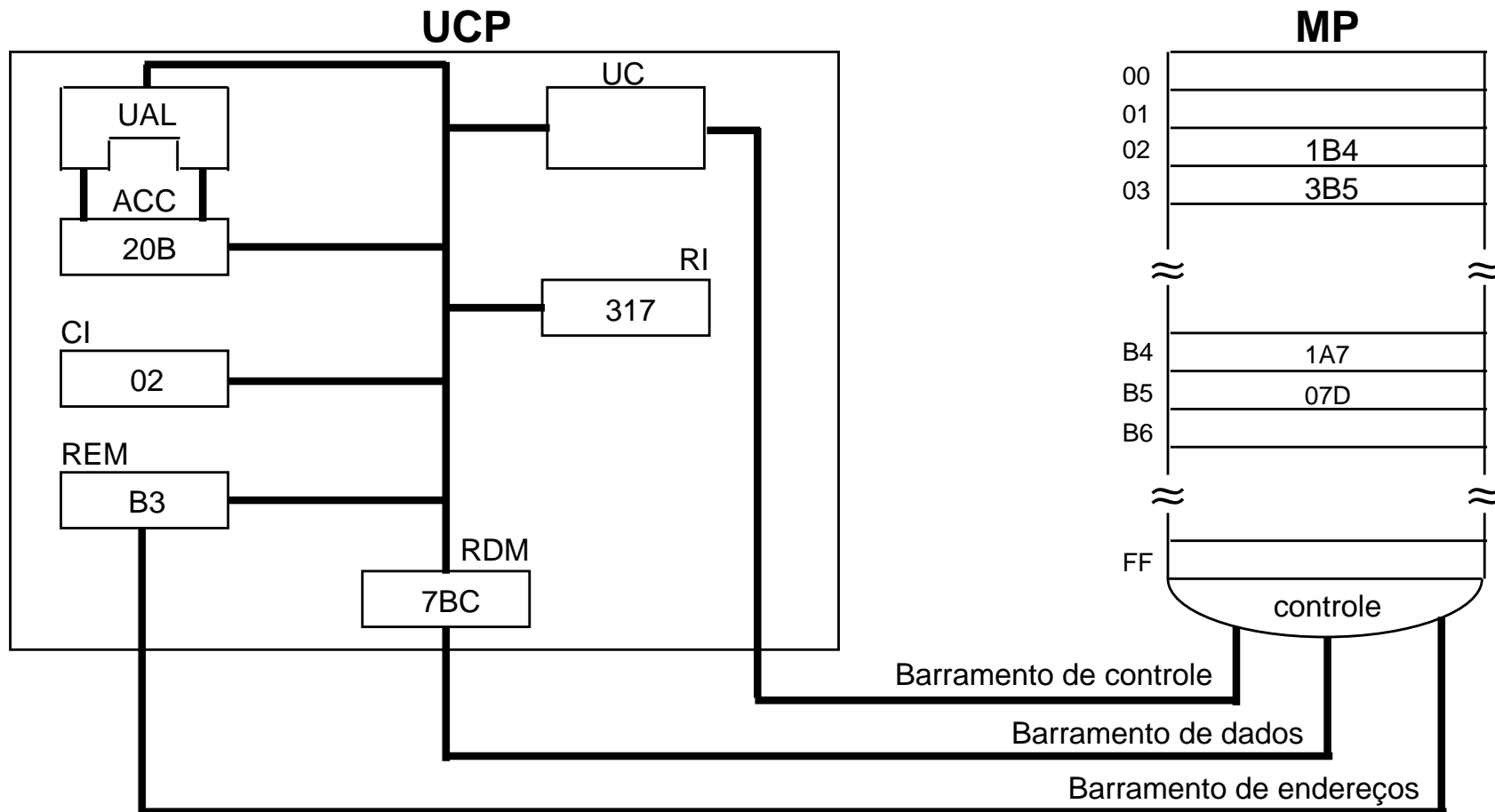
Funcionamento da UCP: O ciclo de instrução

- **ADD 181**
 - Soma o conteúdo do endereço de memória 181 com o conteúdo do acumulador e armazena resultado no acumulador
 - Código de operação = $3_{16} = 0011$
 - Endereço de memória = $181_{10} = 10110101$
 - Instrução = $3B5_{16}$



Funcionamento da UCP: O ciclo de instrução

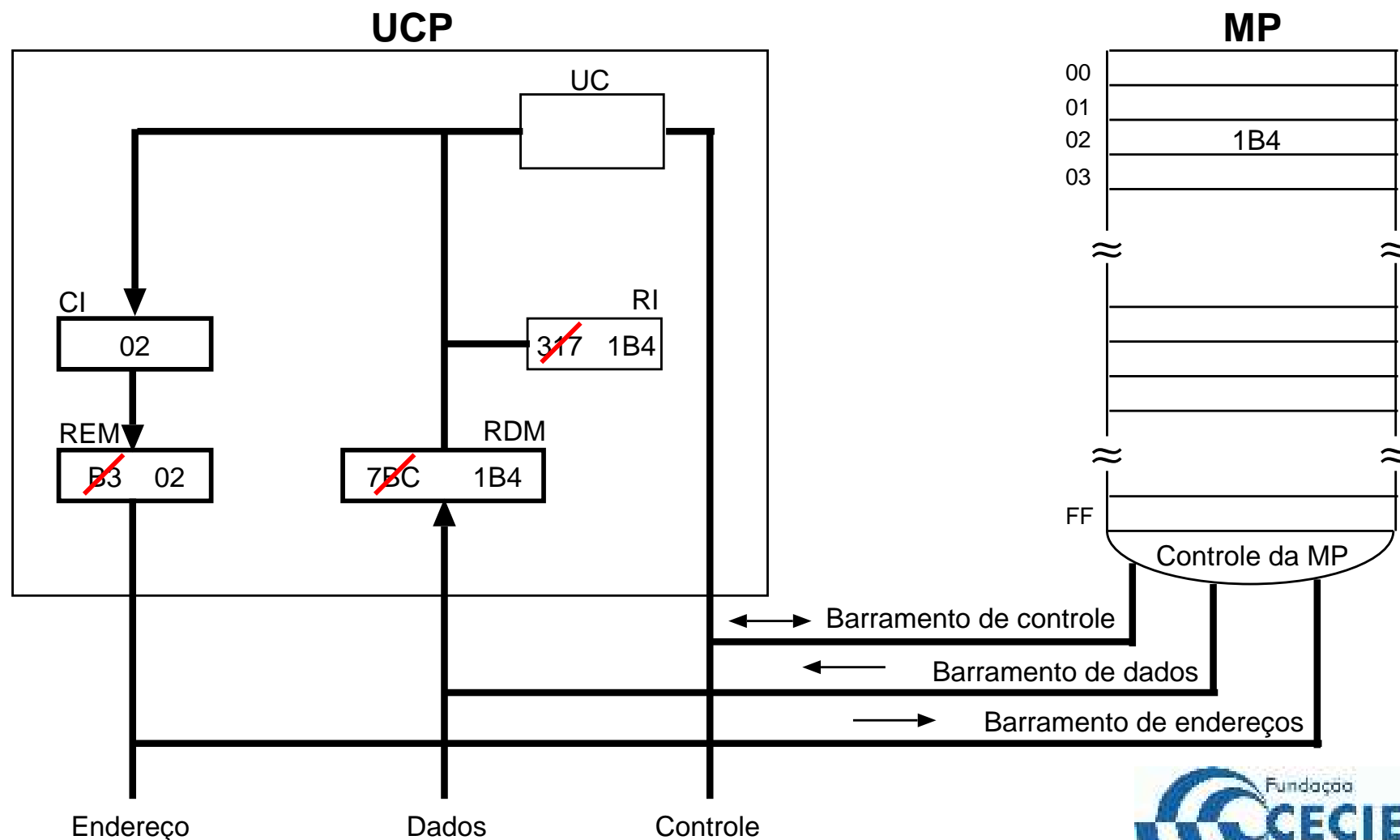
- Cenário de execução das instruções



(Fig. 6.21 do livro texto)

Funcionamento da UCP: Execução de LDA

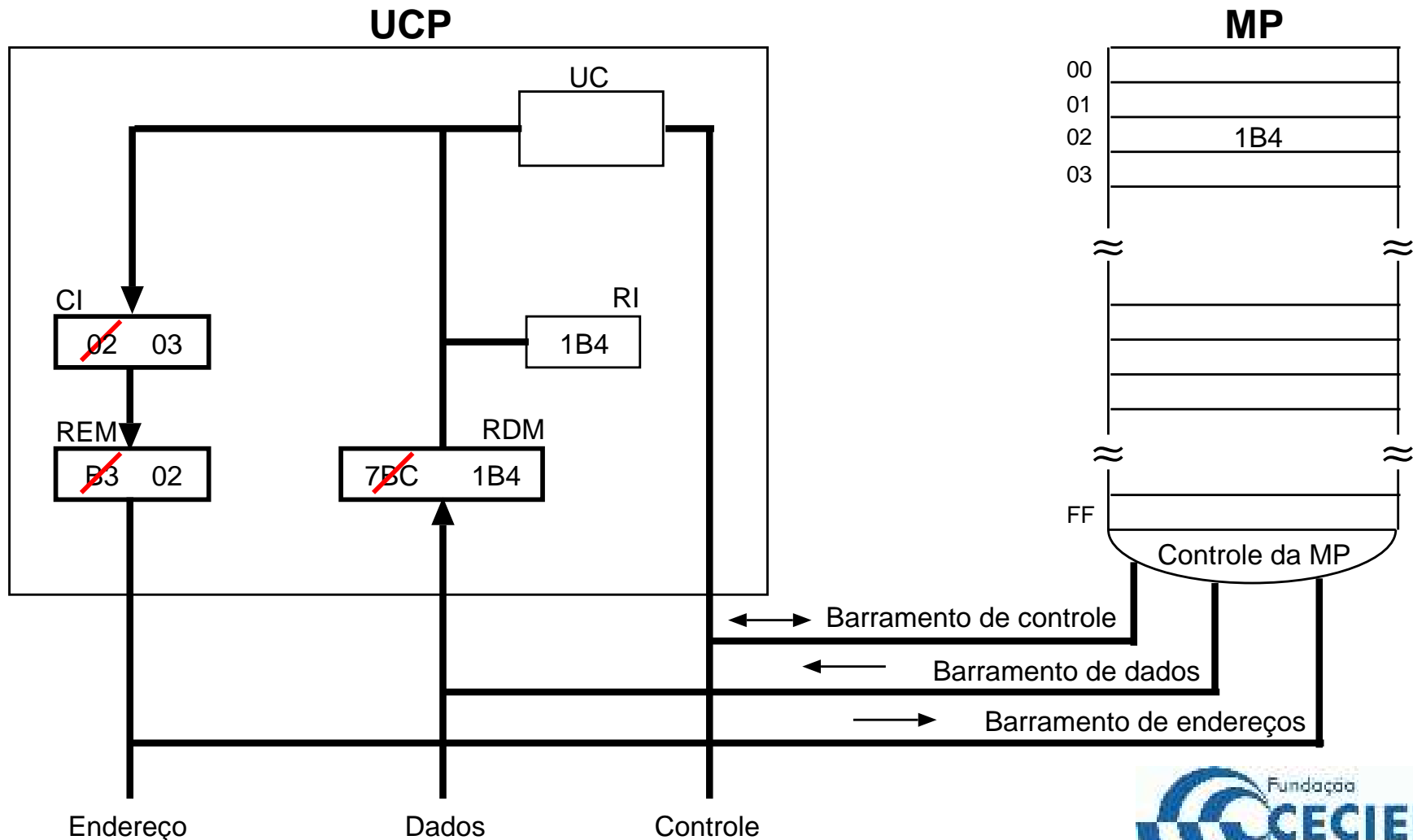
- $RI \leftarrow (CI)$



(Fig. 6.22 do livro texto)

Funcionamento da UCP: Execução de LDA

- $CI \leftarrow CI + 1$



(Fig. 6.22 do livro texto)

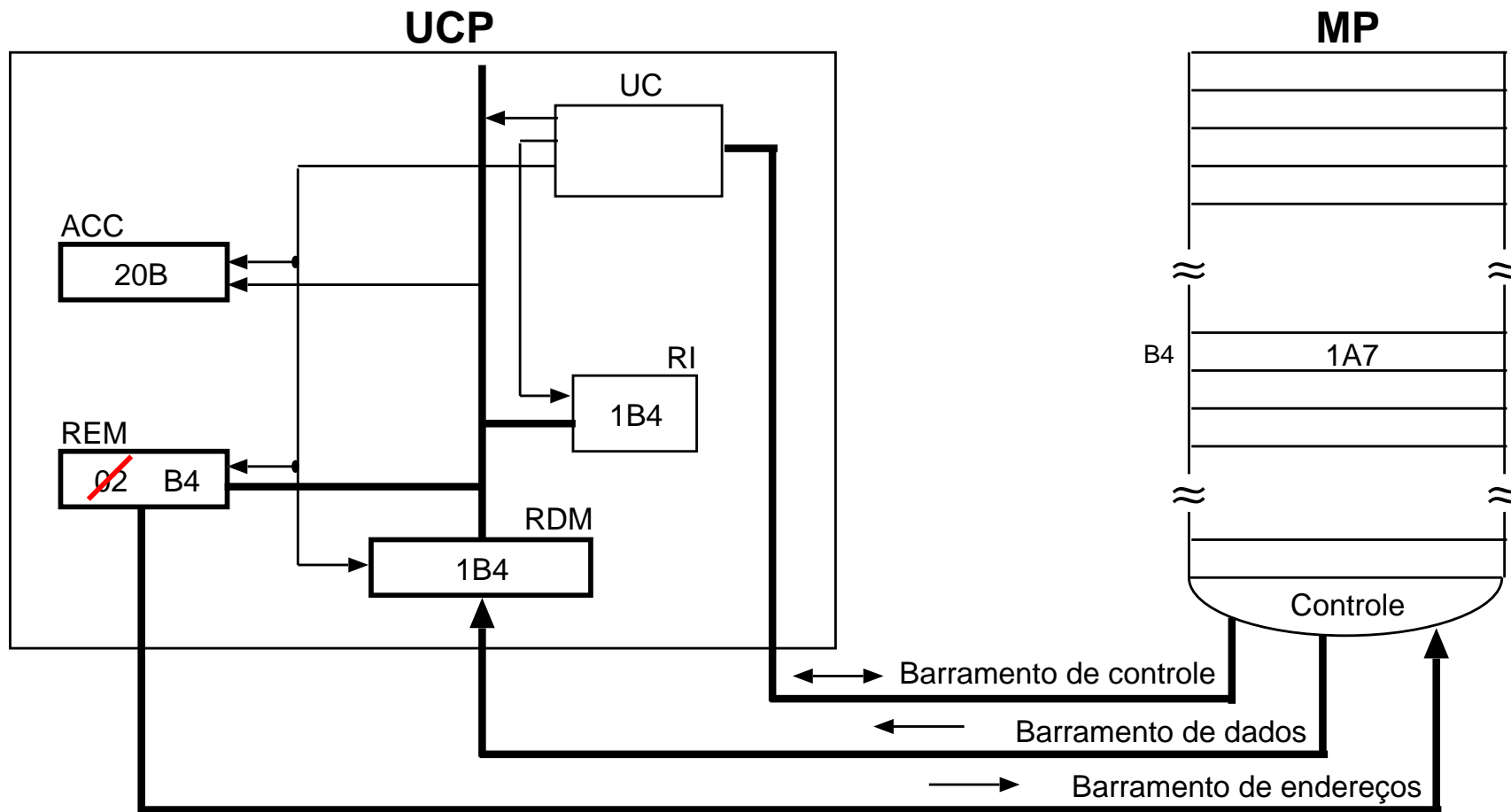
- # UCP



Fundação
CECIEJ
Consórcio cederj

Funcionamento da UCP: Execução de LDA

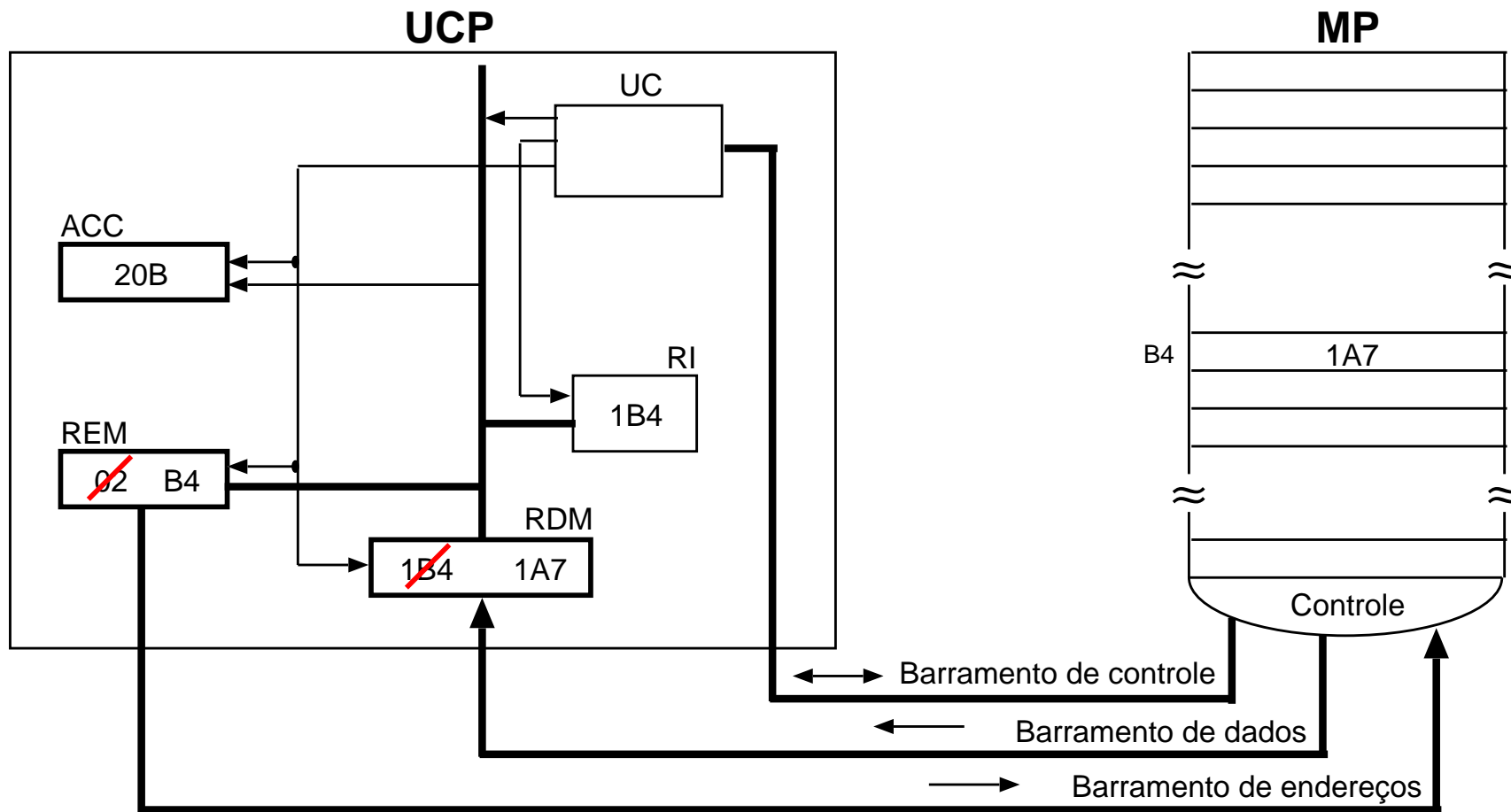
- Execução da operação
 - A UC emite os sinais para que o valor do campo do operando (B4) seja transferido para o REM



(Fig. 6.23 do livro texto)

Funcionamento da UCP: Execução de LDA

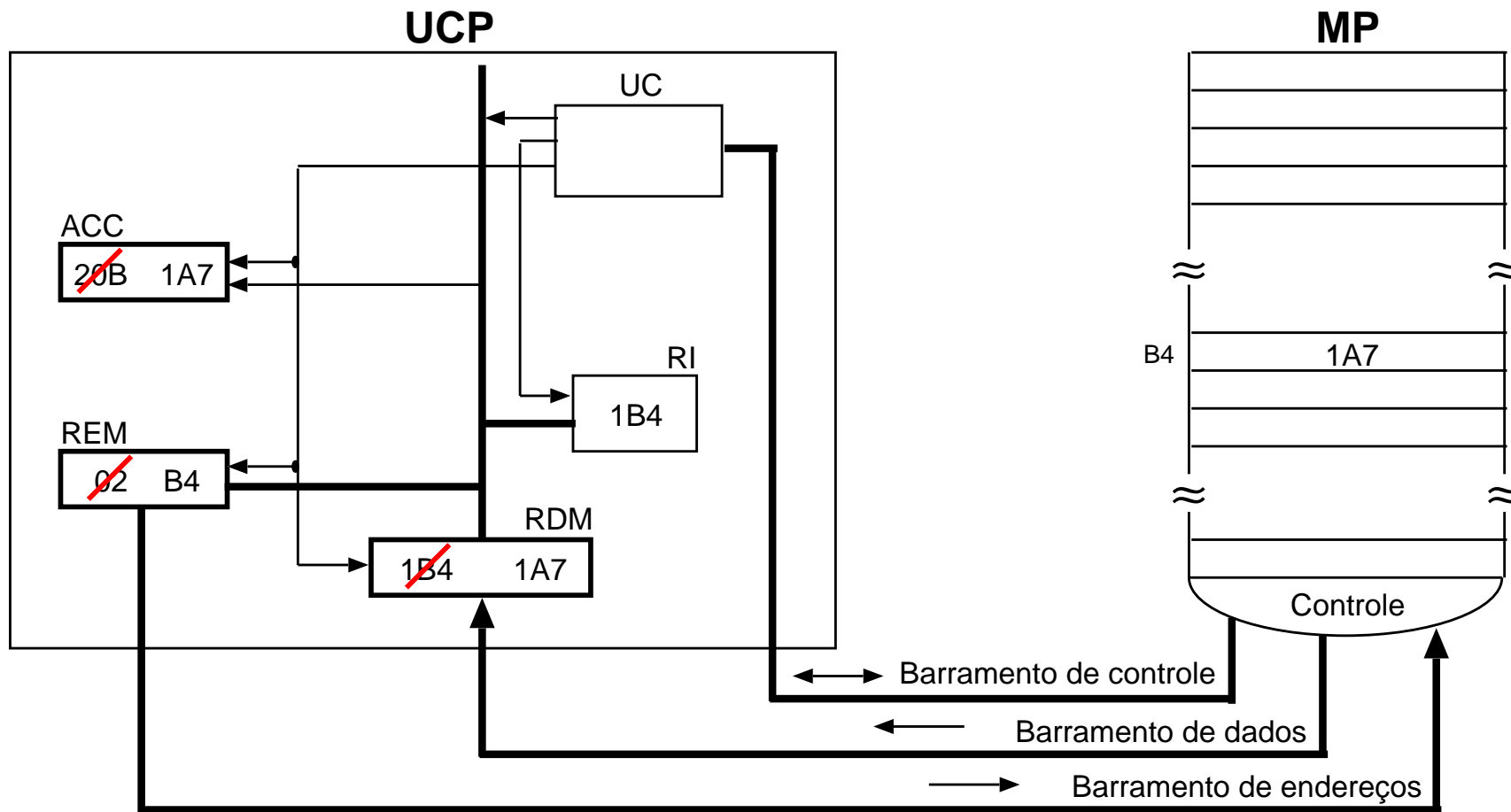
- Execução da operação
 - A UC ativa a linha READ do barramento de controle
 - Conteúdo do endereço de memória B4 é transferido para o RDM



(Fig. 6.23 do livro texto)

Funcionamento da UCP: Execução de LDA

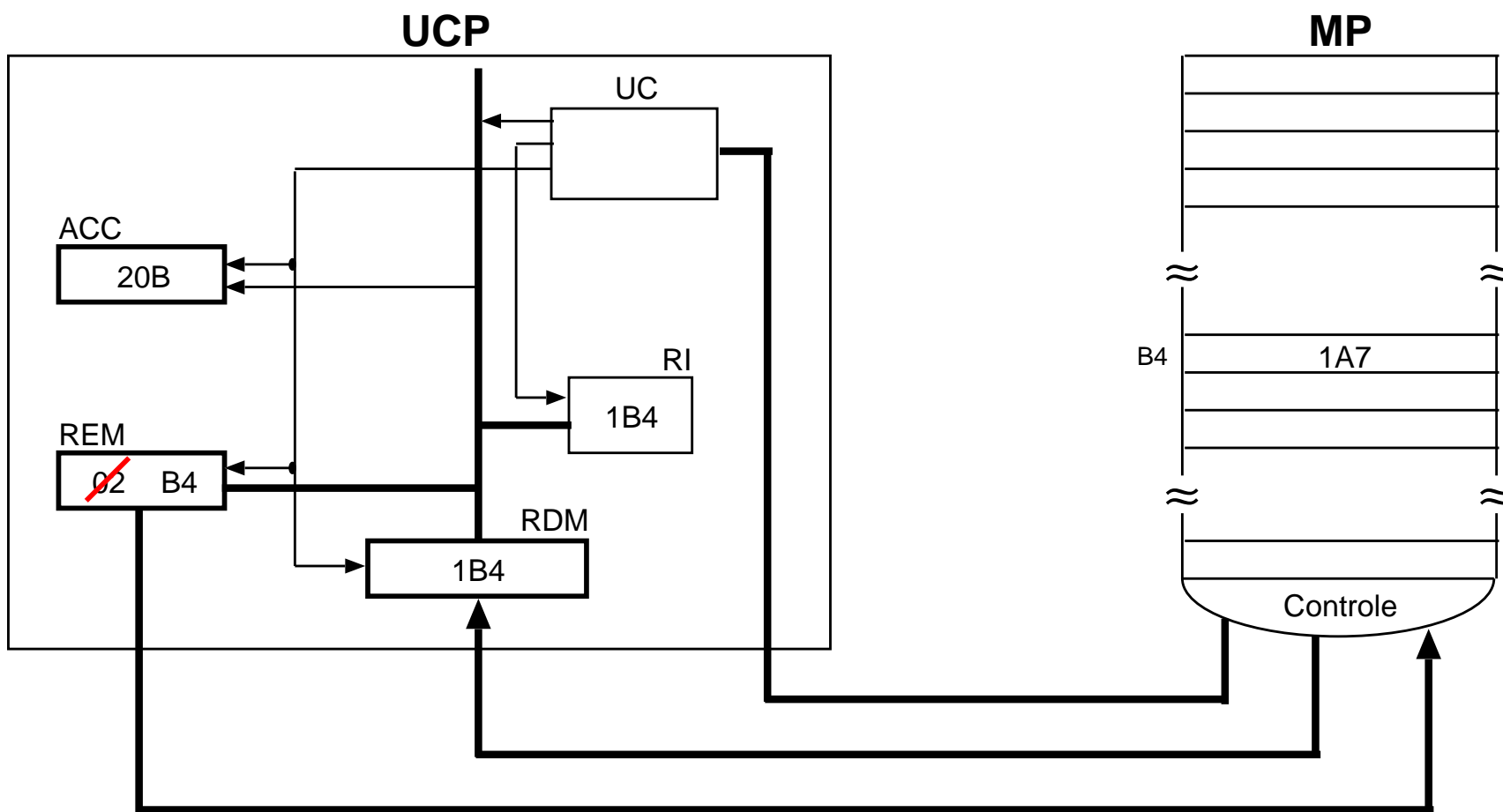
- Execução da operação
 - Conteúdo do RDM é transferido para acumulador



(Fig. 6.23 do livro texto)

Exemplo: Execução da operação

- 1 A UC emite os sinais para que o valor do campo do operando (B4) seja transferido para o REM
- 2 A UC ativa a linha READ do barramento de controle
- 3 Conteúdo do endereço de memória B4 é transferido para o RDM
- 4 Conteúdo do RDM é transferido para acumulador



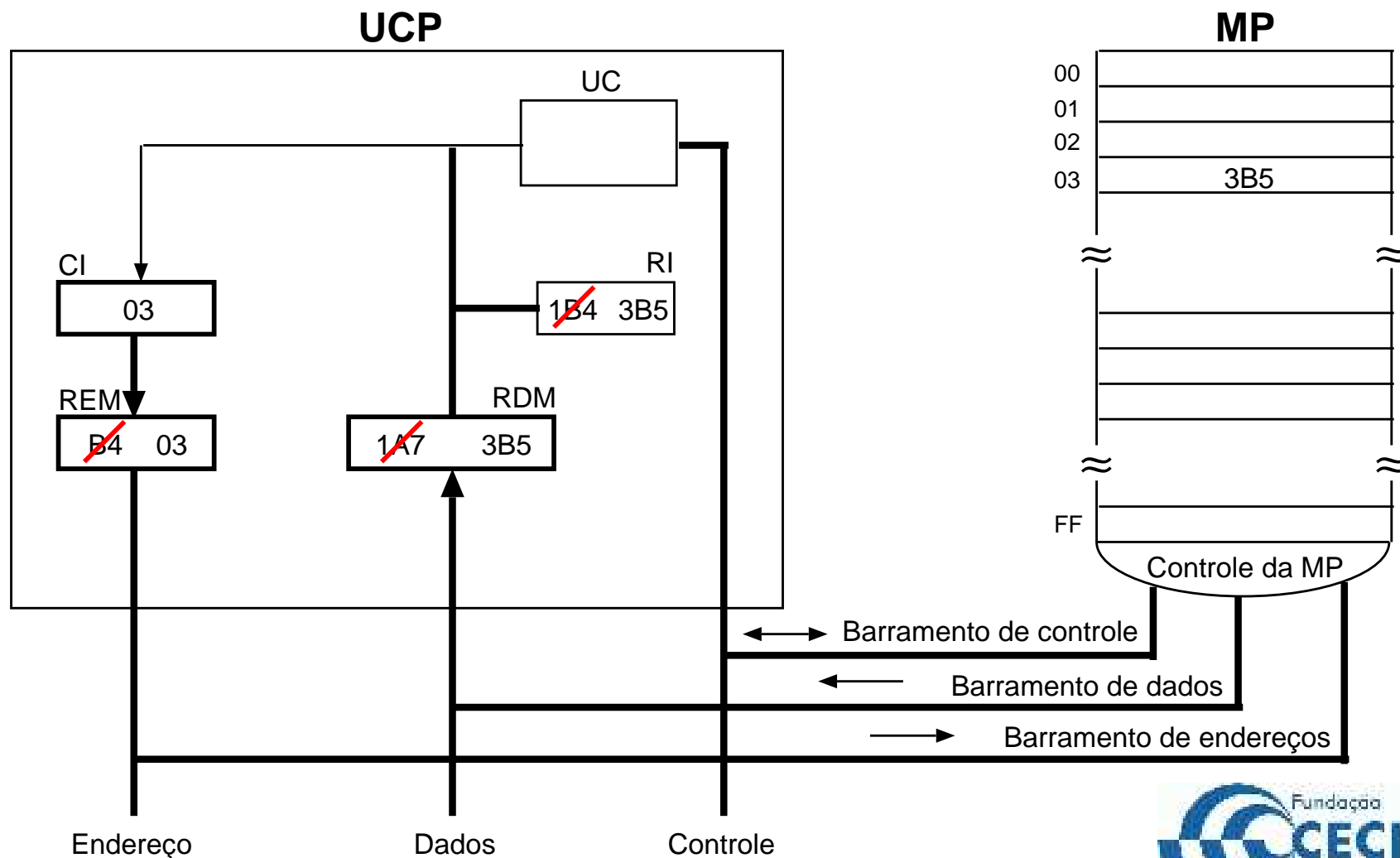
(Fig. 6.23 do livro texto)

Exemplo

Voltar

Funcionamento da UCP: Execução de ADD

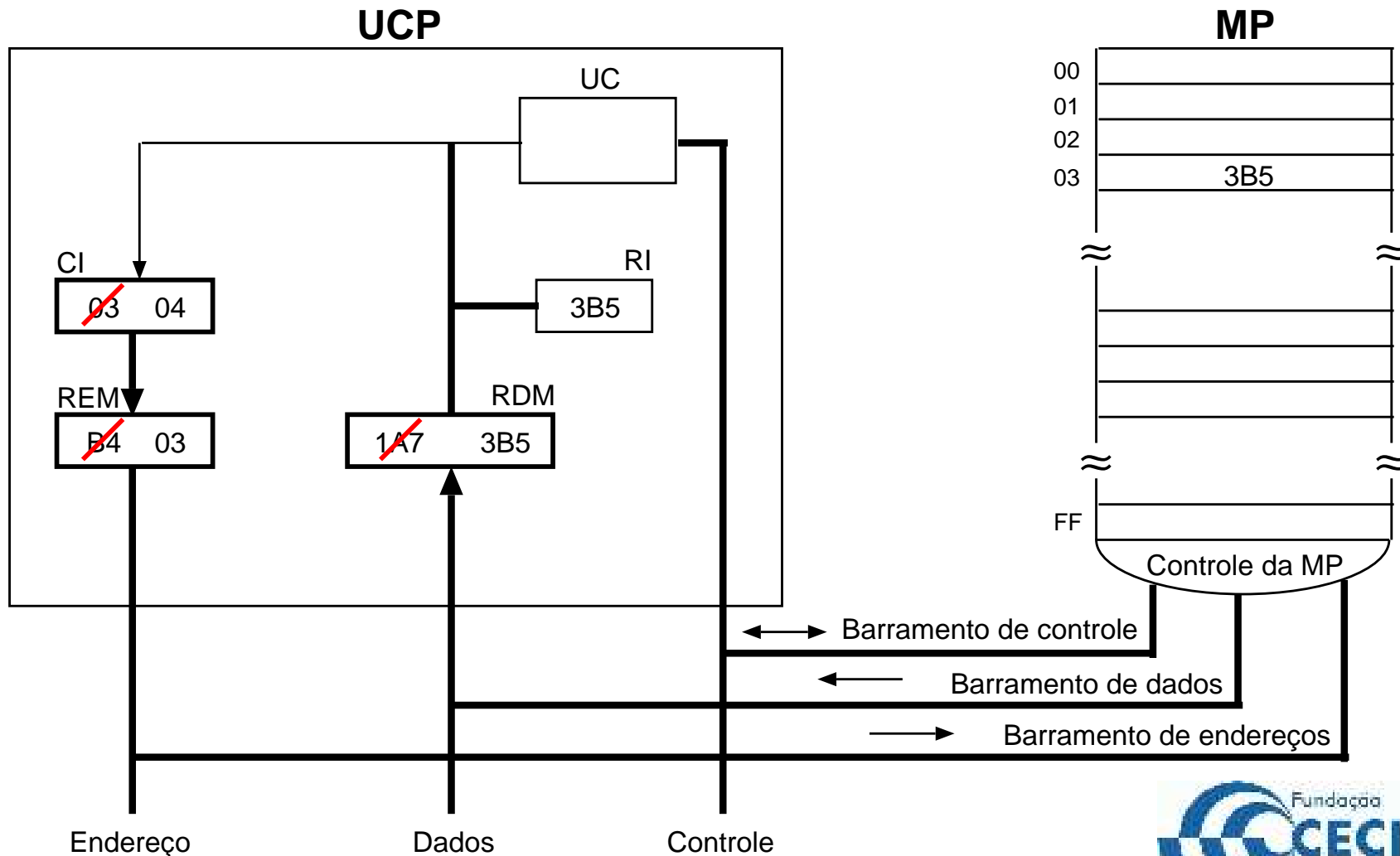
- $RI \leftarrow (CI)$



(Fig. 6.24 do livro texto)

Funcionamento da UCP: Execução de ADD

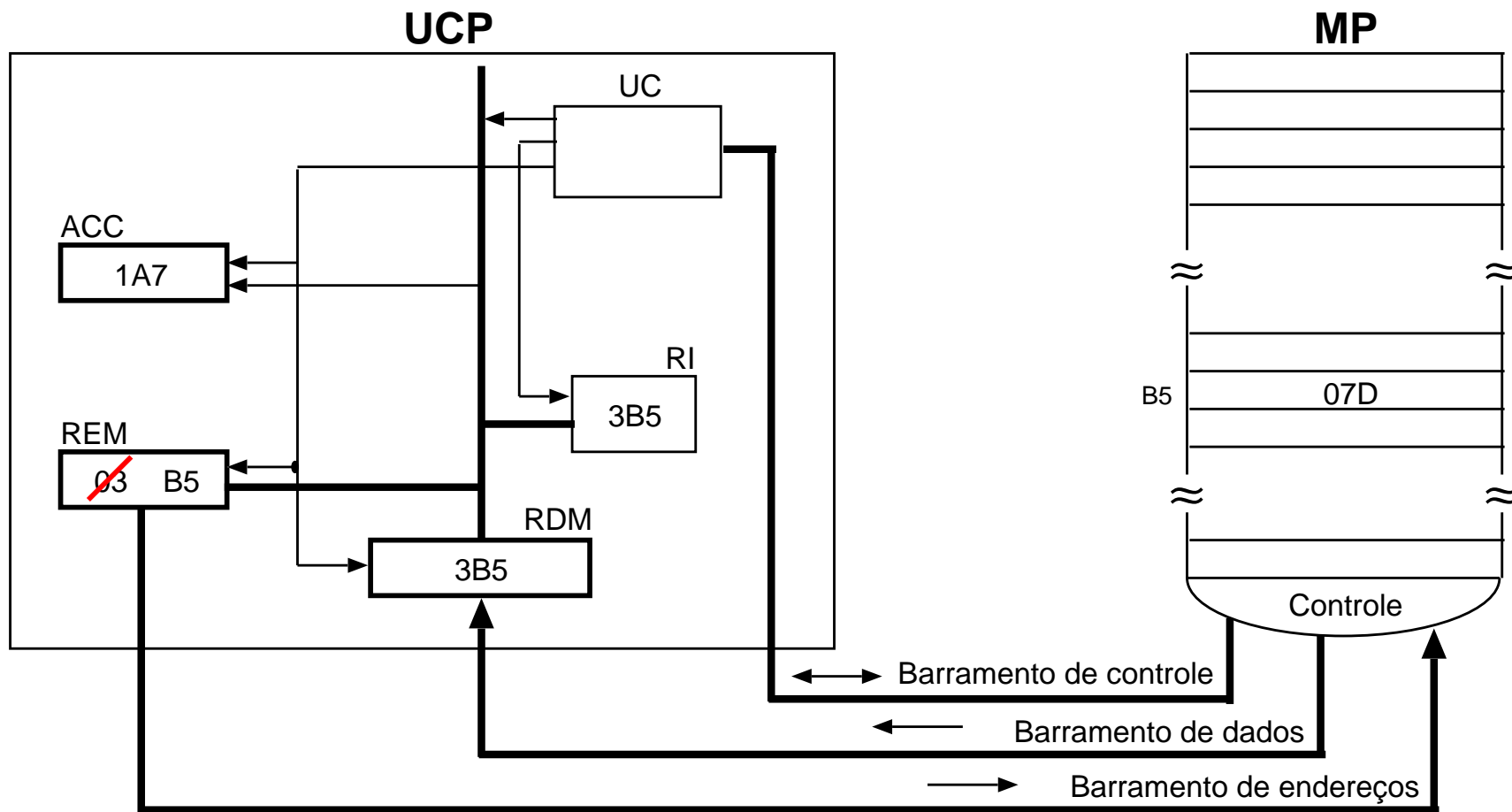
- $CI \leftarrow CI + 1$



(Fig. 6.24 do livro texto)

Funcionamento da UCP: Execução de ADD

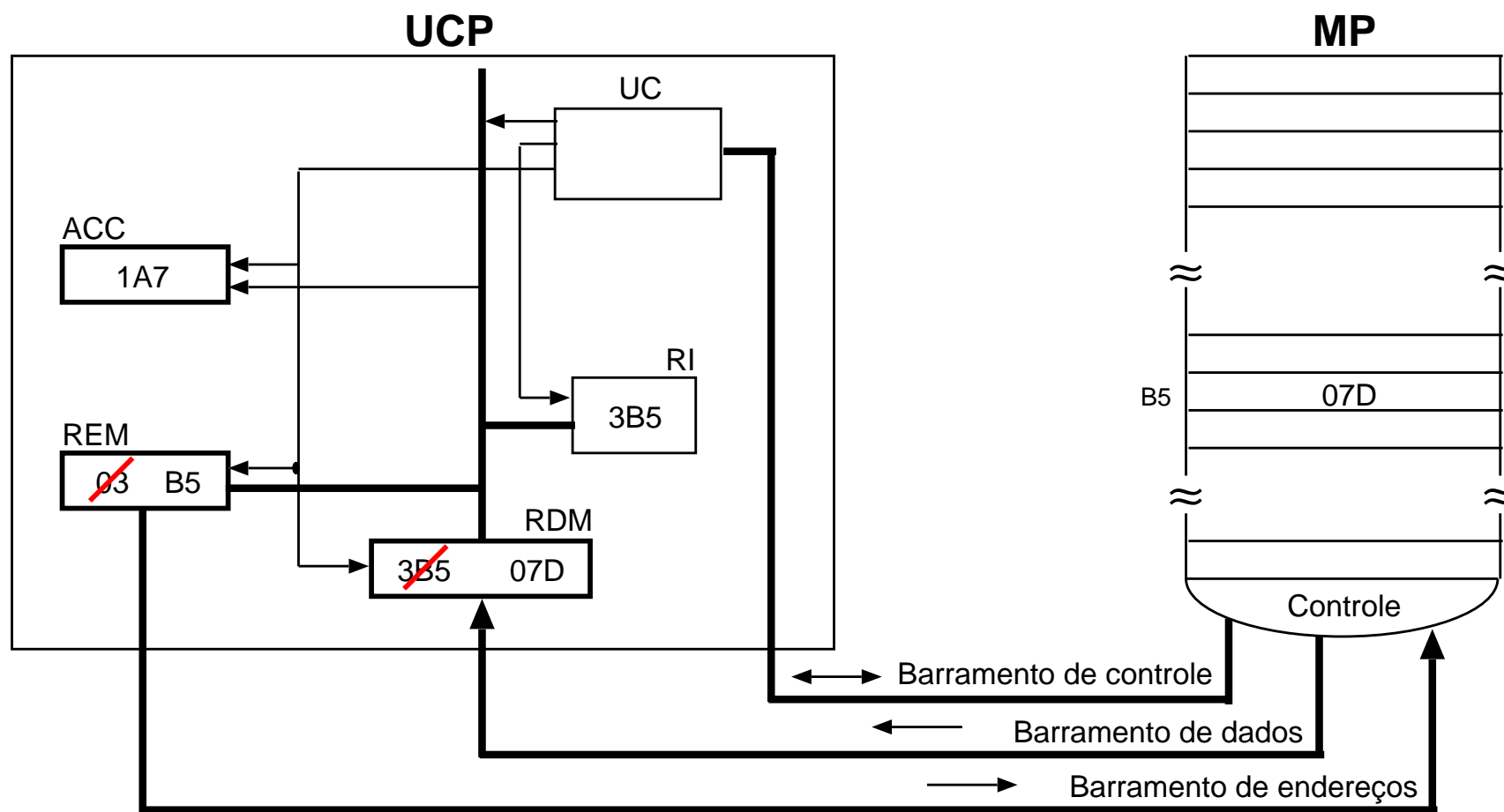
- Busca de operando na memória
 - A UC emite os sinais para que o valor do campo do operando (B5) seja transferido para o REM



(Fig. 6.25 do livro texto)

Funcionamento da UCP: Execução de ADD

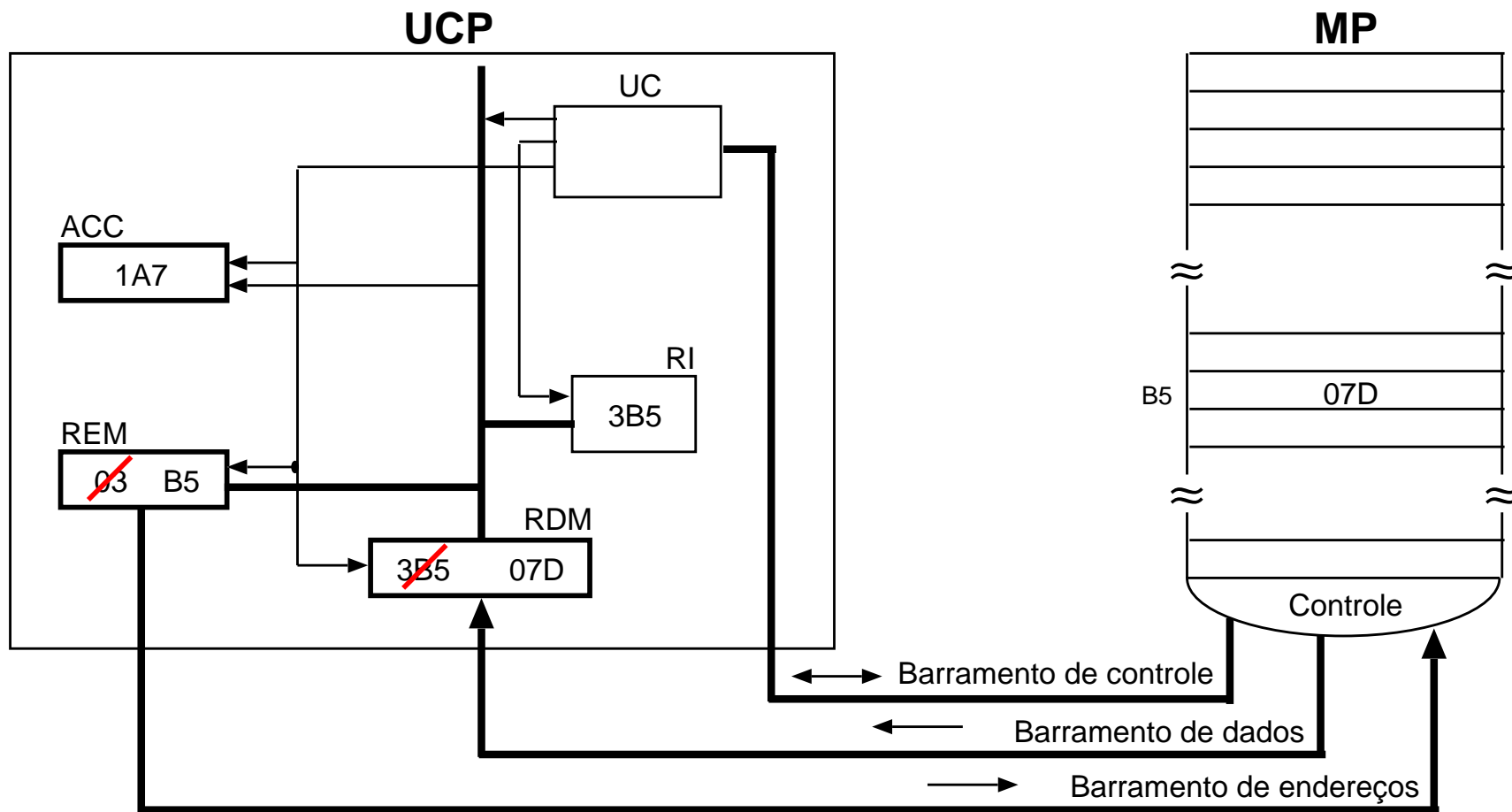
- Busca de operando na memória
 - A UC ativa a linha READ do barramento de controle
 - Conteúdo do endereço de memória B5 é transferido para o RDM



(Fig. 6.25 do livro texto)

Funcionamento da UCP: Execução de ADD

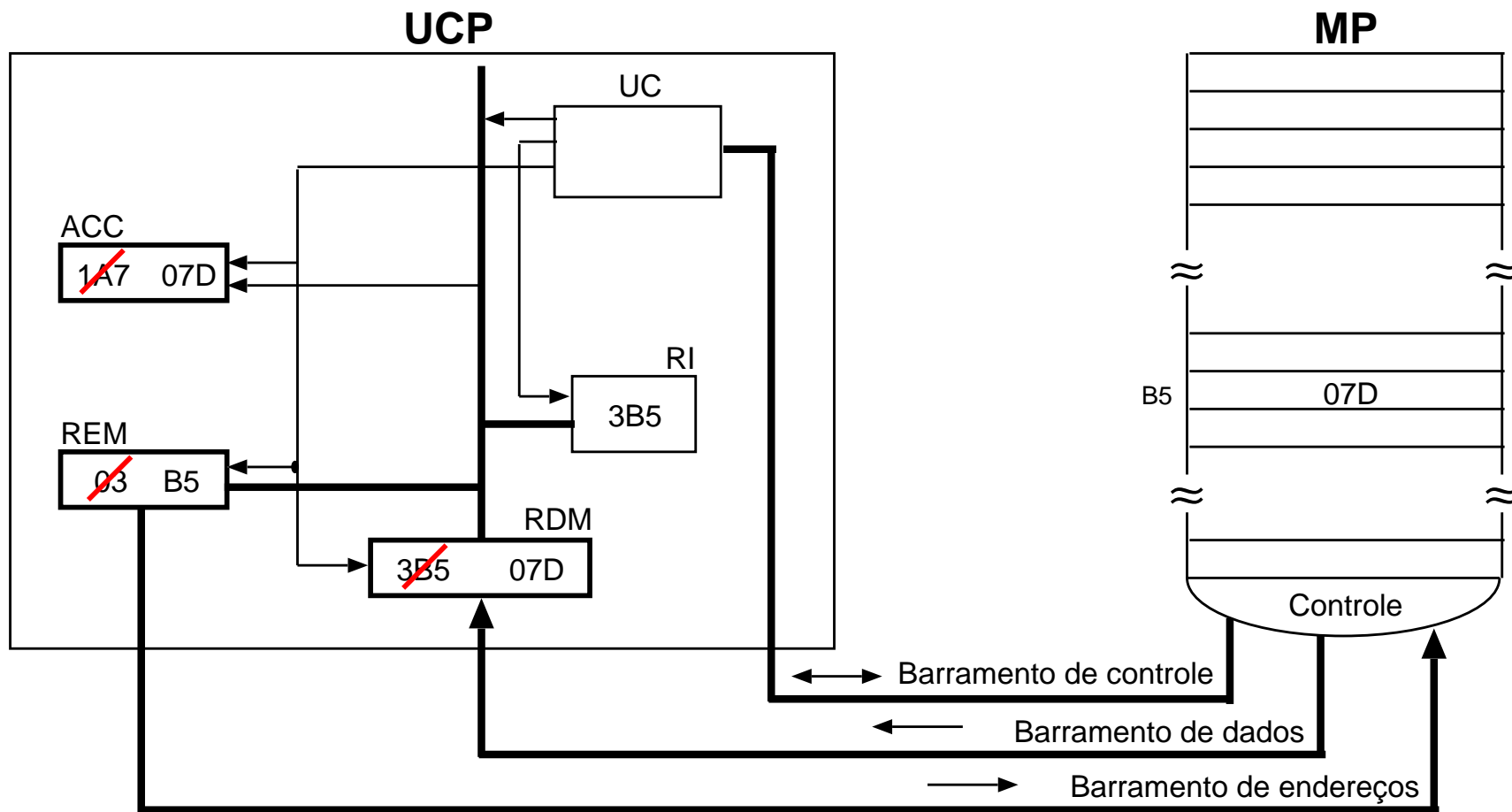
- Busca de operando na memória
 - Dados a serem somados são transferidos para a UAL através do acumulador
 - $UAL \leftarrow ACC$



(Fig. 6.25 do livro texto)

Funcionamento da UCP: Execução de ADD

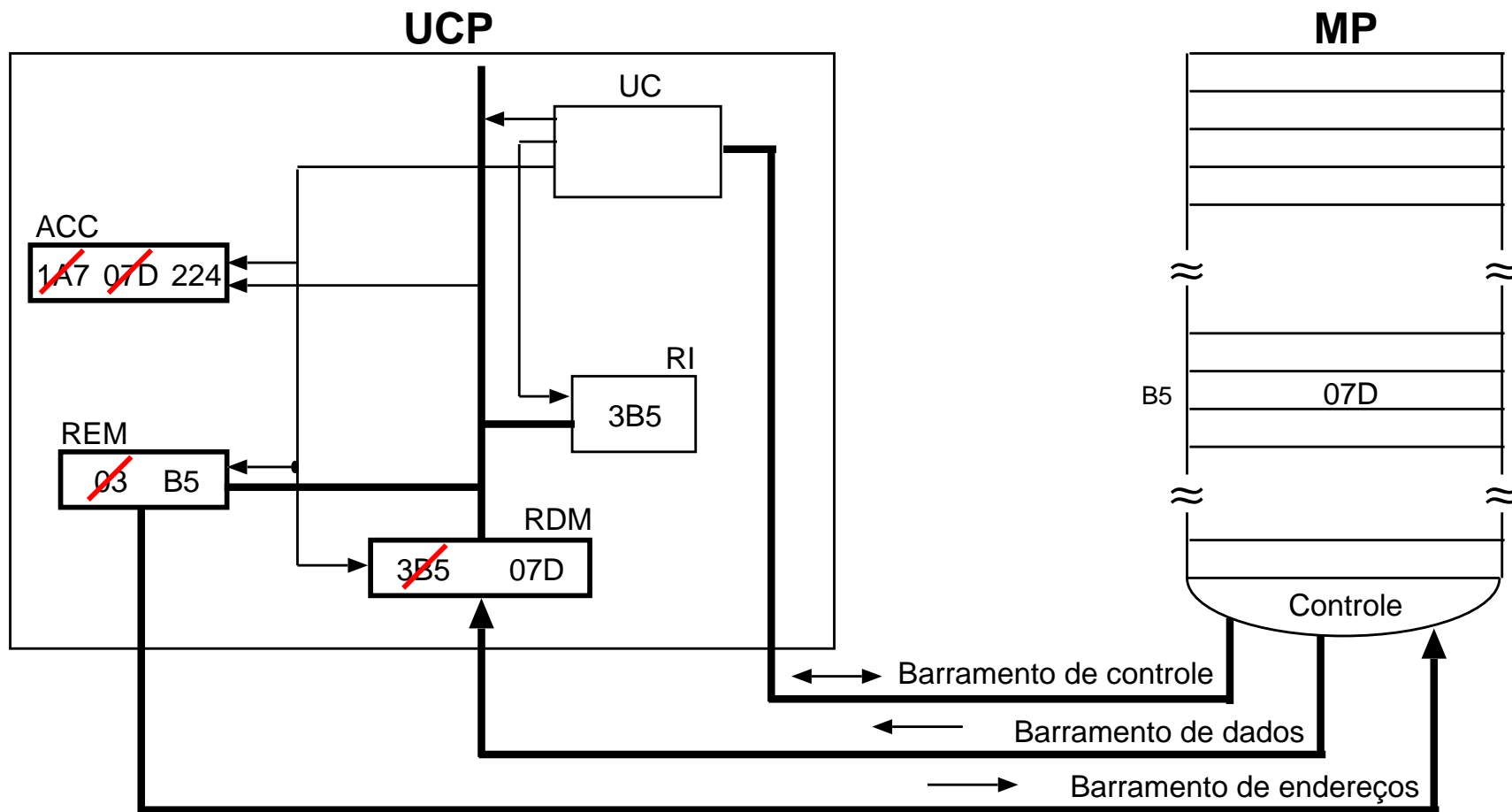
- Busca de operando na memória
 - $ACC \leftarrow RDM$
 - $UAL \leftarrow ACC$



(Fig. 6.25 do livro texto)

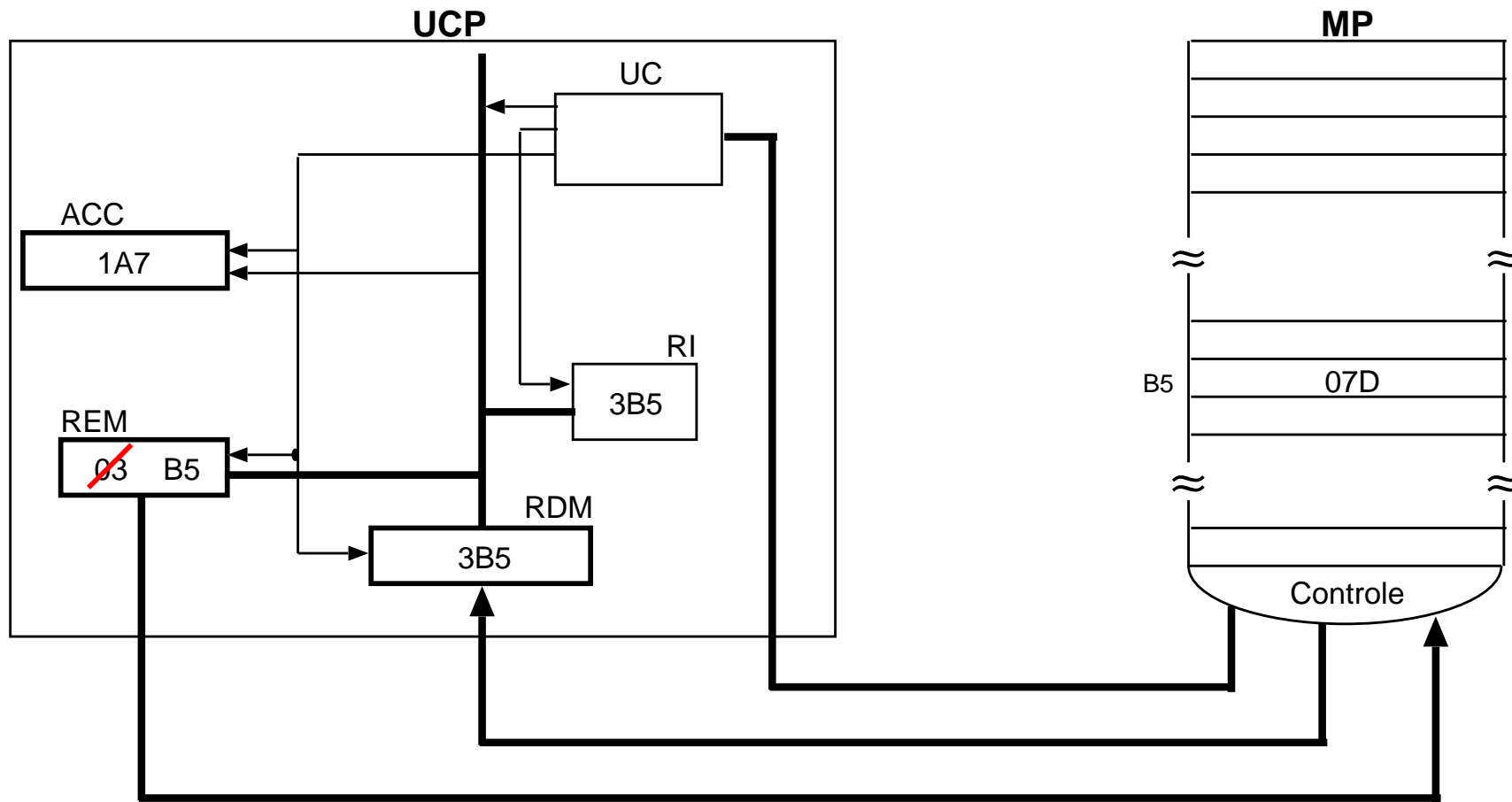
Funcionamento da UCP: Execução de ADD

- Execução da operação
 - Valores são somados na UAL
 - $1A7 + 07D = 224$
 - Resultado armazenado no acumulador



(Fig. 6.25 do livro texto)

- 1 A UC emite os sinais para que o valor do campo do operando (B5) seja transferido para o REM
- 2 A UC ativa a linha READ do barramento de controle
- 3 Conteúdo do endereço de memória B5 é transferido para o RDM
- 4 Dados a serem somados são transferidos para a UAL através do acumulador
- 5 $UAL \leftarrow ACC$ 6 $ACC \leftarrow RDM$ 7 $UAL \leftarrow ACC$
- 8 Valores são somados na UAL - $1A7 + 07D = 224$
- 9 Resultado armazenado no acumulador



Linguagem de montagem

- A maneira mais direta de utilizar o hardware é através da linguagem de máquina (linguagem binária de 0s e 1s)
- Mais complicada e difícil de ser utilizada pelo programador

Linguagem de montagem

- Exemplo de programa em Pascal

$X := A + B - C;$

- Programa convertido em linguagem de máquina

000100100011

001100100100

010000100101

001000100110

Linguagem de montagem

- Valores binários podem ser substituídos por hexadecimais

123

324

425

226

Linguagem de montagem

- Emprego de símbolos alfanuméricos ao invés de números

- $X := A + B - C;$

PROG SOMA

LDA A

ADD B

SUB C

STR X

Linguagem de montagem

- Linguagem de símbolos alfabéticos denominada linguagem de montagem (Assembly)
 - Relação de 1:1 com linguagem de máquina
 - Mais facilidade de compreensão e manuseio
 - Necessita ser traduzida para a linguagem de máquina utilizando-se um montador (Assembler)

Linguagem de montagem

- Linha de instrução da linguagem Assembly é composta geralmente de 4 campos

Rótulo	Operação	Operandos	Comentário
Soma	Proc C Value: Palavra		
	Mov	AX , 0	Inicializar, zerando

Linguagem de montagem

- Linguagem utilizada para desenvolvimento de programas básicos (sistemas operacionais) e de controle
- Atualmente, linguagens de alto nível, como C, permitem manipulação de estruturas primitivas e possibilitam desenvolvimento de programas mais estruturados e menores

Unidade Aritmética e Lógica

- UAL é constituída de circuitos dedicados a realizar:
 - operações aritméticas como soma, subtração, divisão e multiplicação
 - operações lógicas como AND, OR, NOT
 - deslocamentos dos bits de um número



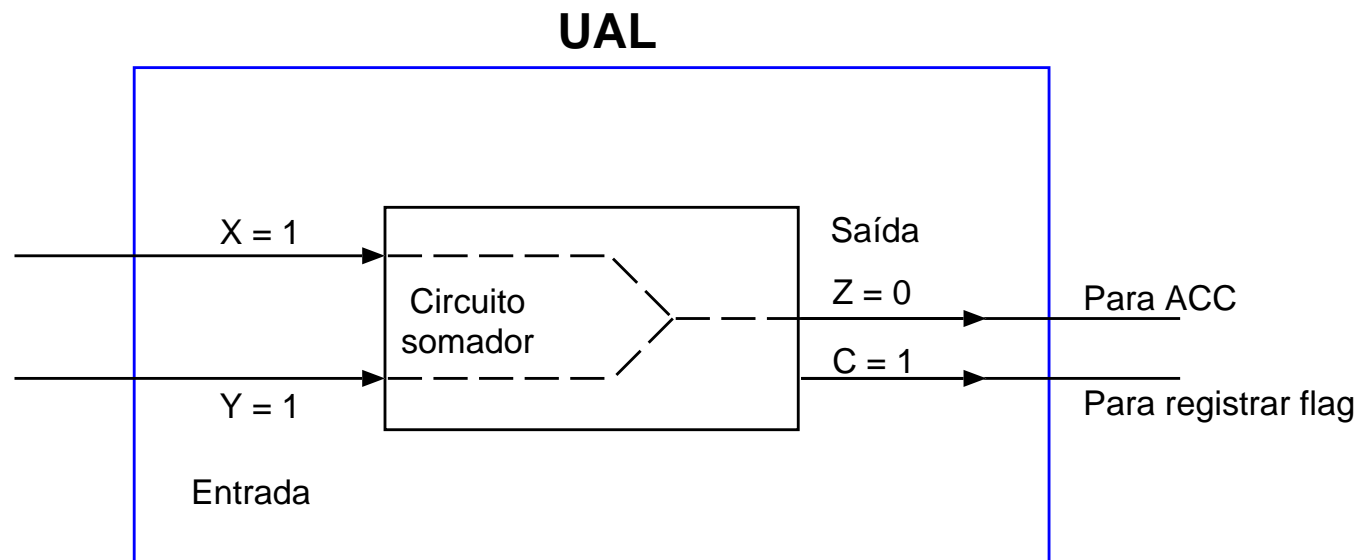
Errata: Deslocar o registrador na base 2 para a esquerda está multiplicando por dois e para a direita está dividindo por dois.

Unidade Aritmética e Lógica

- Primeiros processadores possuíam um co-processador para efetuar operações com valores fracionários
- Co-processador de ponto flutuante
- Atualmente as UALs responsáveis pelas operações com números inteiros e fracionários estão integradas em uma mesma pastilha

Unidade Aritmética e Lógica

- Funcionamento básico da UAL



X e Y - valores que serão somados ($1 + 1 = 0$ e vai 1)
Z - resultado da operação $m = 0$
C - bit "vai 1" = 1

(Fig. 6.31 do livro texto)

Exercício 1

- Considere um computador com um conjunto de 128 instruções. Cada instrução é composta de 2 campos: o primeiro para indicar a operação e o segundo para indicar o endereço do operando. Supondo que sua memória tenha capacidade de armazenar 512 palavras e que cada instrução tem o tamanho de uma palavra e da célula de memória, pergunta-se:
 - Qual é o tamanho em bits do REM, RDM, RI e CI ?
 - Qual a capacidade de memória em bytes ?
 - Se o tamanho das instruções for alterado para 17 bits, mantendo inalterado o tamanho do REM, quantas novas instruções poderiam ser criadas ?

Exercício 1 - Resposta

- Como sua memória tem capacidade de armazenar 512 palavras, o número de bits do endereço é $\log_2 512 = \log_2 2^9 = 9$ bits
- O tamanho em bits do REM e do CI deve ser igual ao número de bits utilizado para o endereço, que é 9 bits
- Como o computador possui um conjunto de 128 instruções, o número de bits do código de operação é $\log_2 128 = \log_2 2^7 = 7$ bits
- A instrução tem o tamanho igual ao número de bits do código de operação (7) somado ao número de bits do endereço (9), igual a 16 bits
- Como cada instrução tem o tamanho de uma palavra e da célula de memória, o tamanho da palavra é igual a 16 bits
- O tamanho em bits do RDM e do RI deve ser igual ao tamanho da palavra que é igual a 16
- A capacidade da memória em bytes é igual ao número de palavras multiplicado pelo tamanho da palavra: $512 \times 16 \text{ bits} = 512 \times 2 \text{ bytes} = 1024 \text{ bytes}$ ou 1 Kbytes
- Se o tamanho das instruções for alterado para 17 bits, se mantendo inalterado o tamanho do REM, teremos um novo número de bits para o código de operação igual a $17 - 9 = 8$ bits. Portanto podemos ter o dobro de instruções.

Exercício 2

- Considere um sistema que possui uma memória com 256 células, sendo que cada célula pode armazenar 12 bits. Na figura abaixo são apresentados os endereços e conteúdos de algumas destas células

Endereço	Conteúdo
00	010
01	AFD
02	000
A4	123
A5	135
A6	B00
FD	440
FE	2F8
FF	0A5

Exercício 2

- Qual a capacidade total de memória ?
- Supondo que, no início de um ciclo de instrução, o conteúdo do CI seja o hexadecimal A5 e que cada instrução ocupa uma única célula, qual instrução será executada, considerando o conjunto de instruções visto nesta aula ?
- Supondo que o conteúdo do REM tenha o valor hexadecimal FD e que um sinal de leitura seja enviado da UCP para a memória, qual deverá ser o conteúdo do RDM ao final do ciclo de leitura ?

Exercício 2 - Resposta

- Como o computador possui 256 células com 12 bits cada uma, a capacidade total da memória é igual a $256 \times 12 = 3072$ bits.
- Como o valor do CI é A5, a instrução a ser executada é aquela armazenada no endereço A5 que é igual a 135. Para podermos decodificar esta instrução convertamos para a base 2, obtendo 000100110101. Os 4 primeiros bits, 0001, indicam que a instrução é LDA. Os oito bits restantes indicam o endereço de memória do operando que é igual a 35 na base hexadecimal ou 53 na base decimal. Logo a instrução é LDA 53.
- Como o REM possui o valor FD, será lido o conteúdo da memória cujo endereço é FD. Este valor será colocado no RDM, logo RDM terá o conteúdo 440.

Exercícios

- Capítulo 6 do livro texto

11, 13, 14 e 17