4. Computadores Microprogramados

(capítulo 3 do livro texto) – (utilize o simulador)

Conceito proposto por Wilkes – lembrar EDSAC / Cambridge.

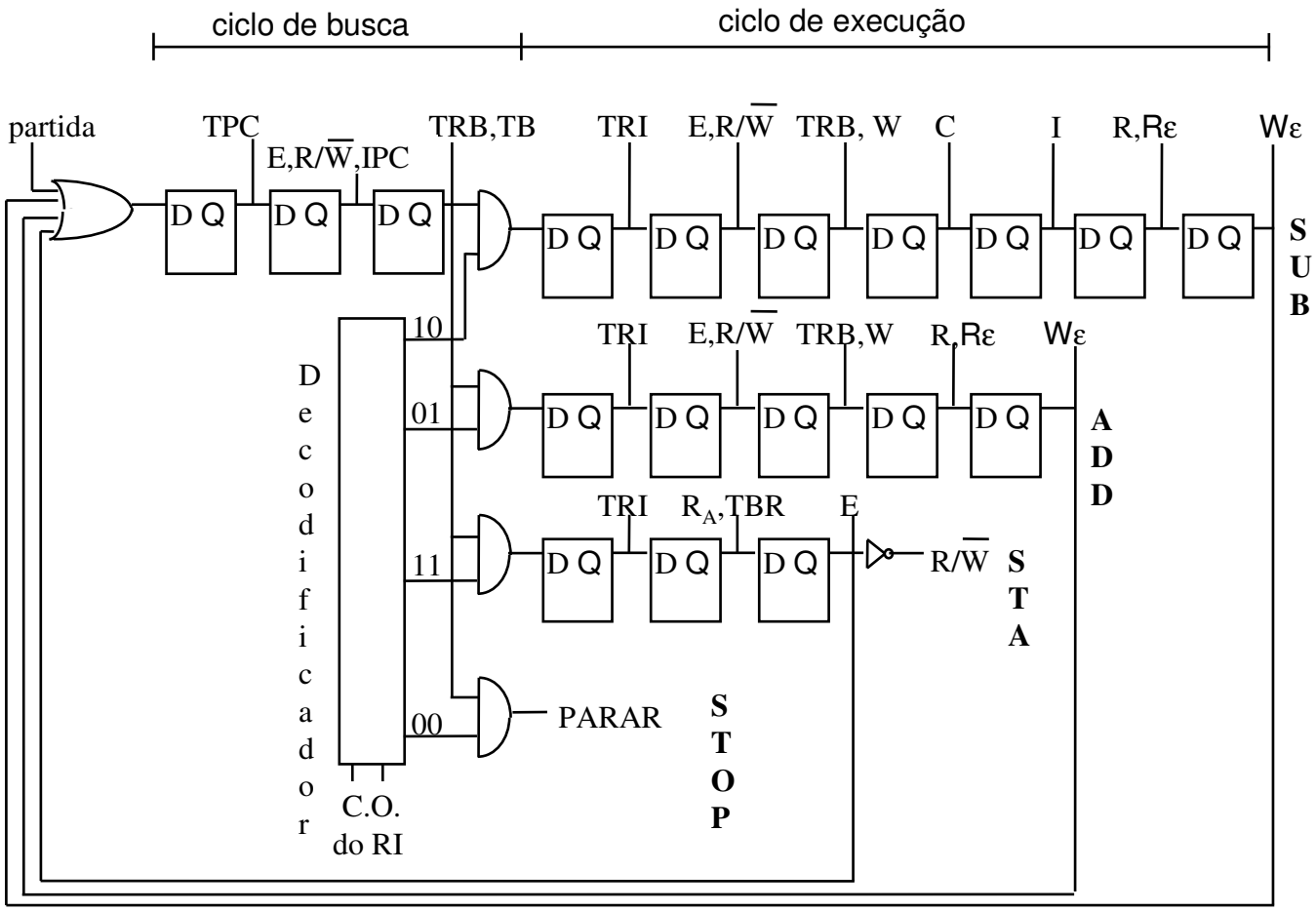
É uma abordagem para a construção de Unidades de Controle de CPUs.

A UC – Unidade de Controle – é a seqüenciadora e controladora da execução das instruções de máquina.

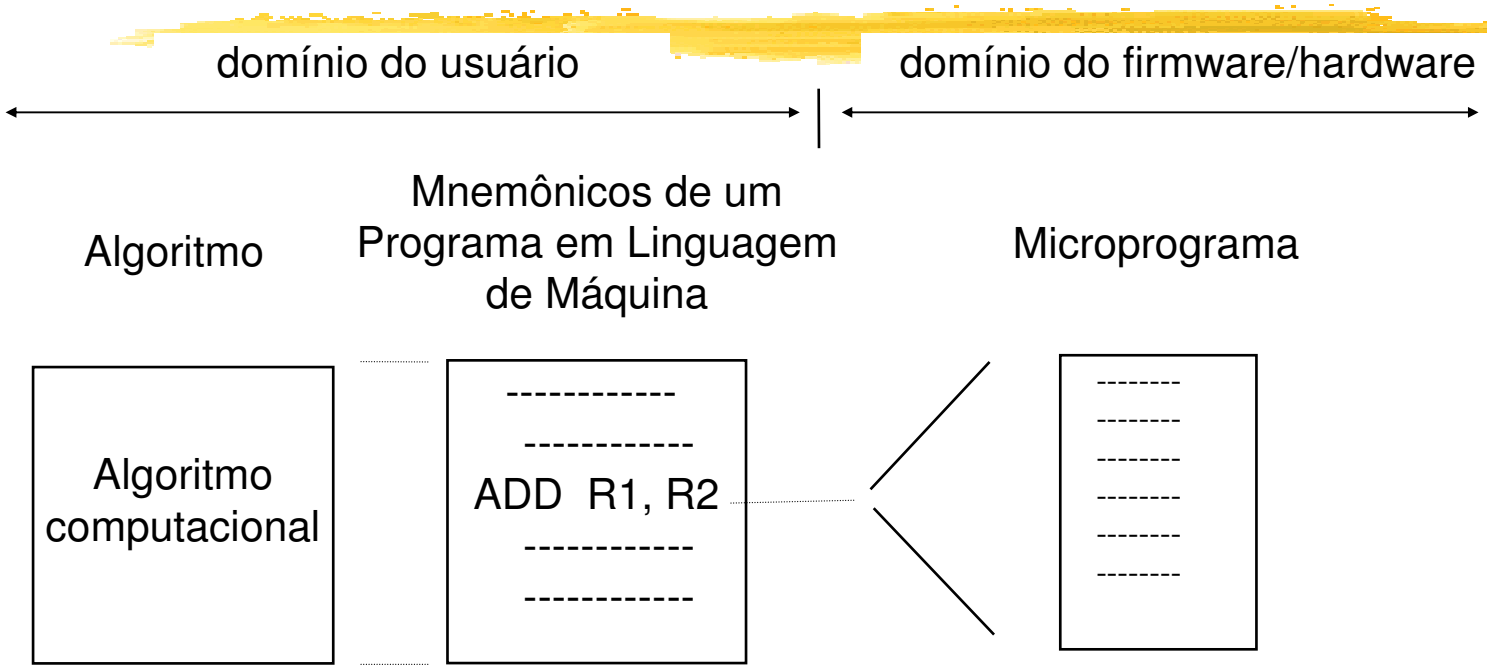
via um conjunto de microoperações que atuam sobre regs., memória, etc.

4.1 Controlador Convencional

Convencional: C.O. → decodificação → microcomandos (sinais de controle)



4.2 Microprograma e microinstrução



Microprograma x Instrução em Assembly

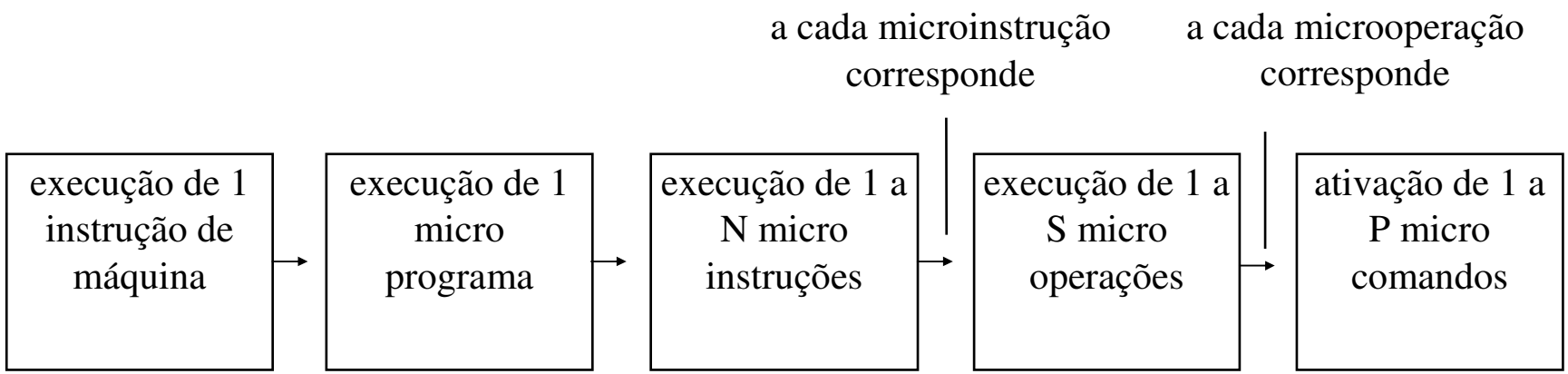
Microprograma e microinstrução (cont.)

Instrução em Linguagem de Máquina	Microprograma	Microinstrução	Microoperação	Microcomando
uma instrução de máquina	um micro programa	1	REM ← (PC)	*TPC*
		2	RDM ← (m)	*E, R/$\overline{W}$,TBR*
			PC ← (PC) + 1	*IPC*
		3	:	:
		:	:	:

Um ou um conjunto de sinais de controle definem uma **microinstrução**; uma ou mais microinstruções definem um **microprograma**.

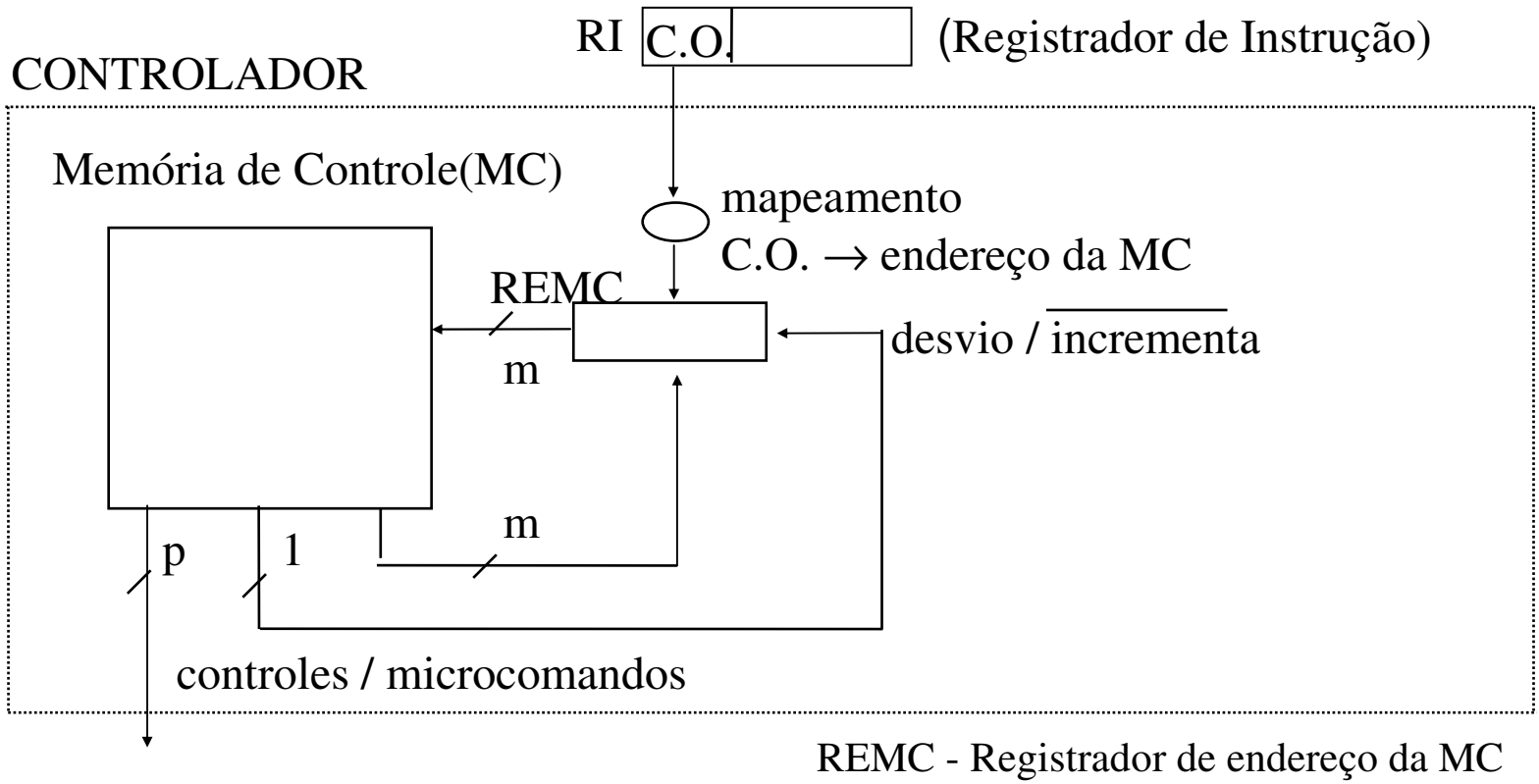
Microprograma e microinstrução (cont.)

Instrução em Linguagem de Máquina	Microprograma	Microinstrução	Microoperação	Microcomando
uma instrução de máquina	um micro programa	1	REM ← (PC)	*TPC*
		2	RDM ← (m)	*E, R/$\overline{W}$,TBR*
			PC ← (PC) + 1	*IPC*
		3	:	:
		:	:	:

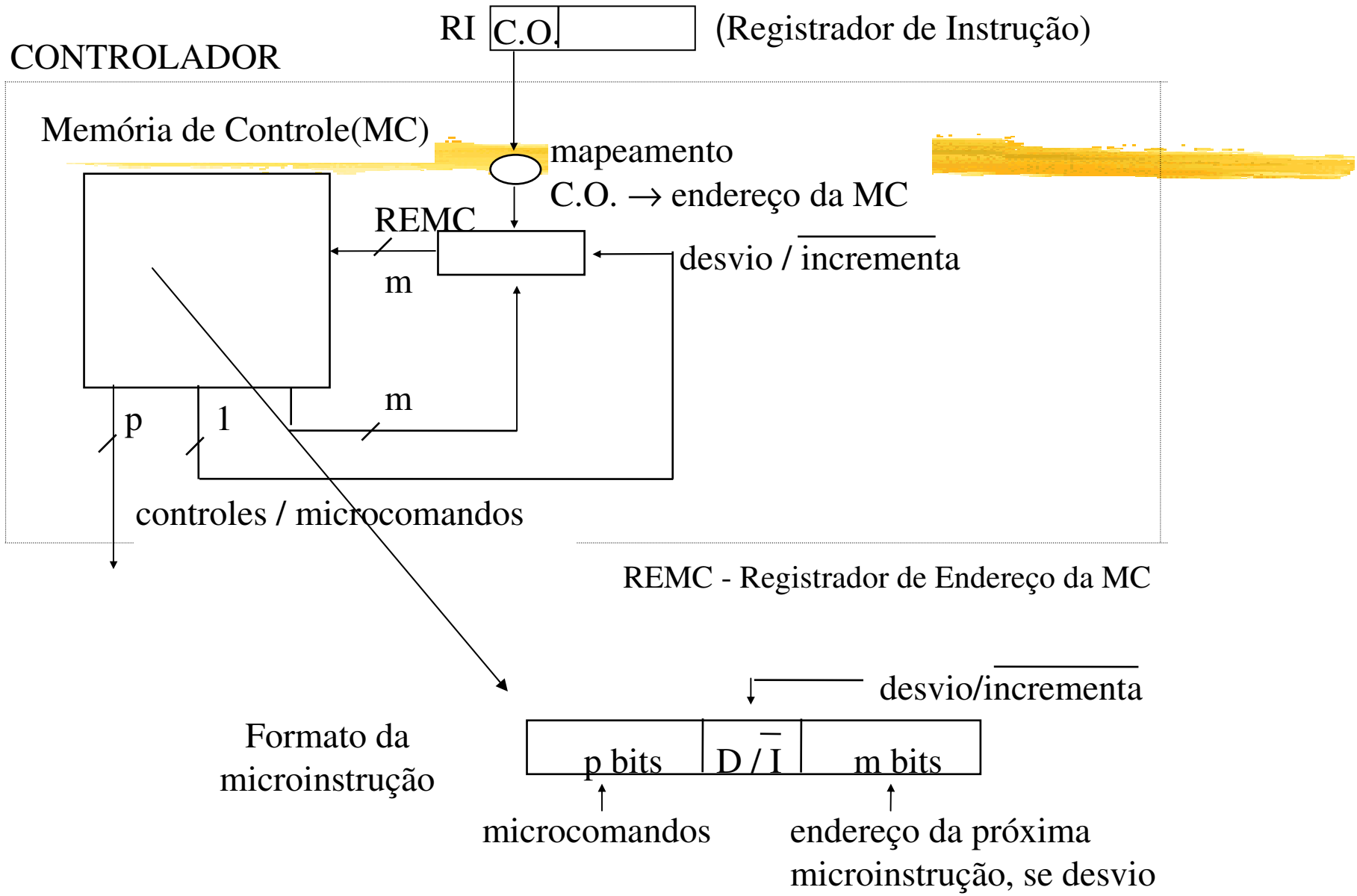


Controlador Microprogramado

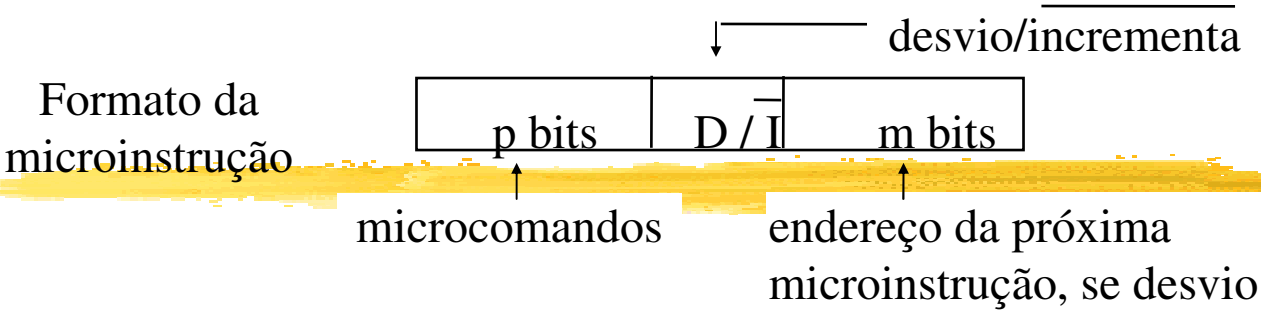
Microprogramado: C.O. → mapeamento para endereço da memória de controle → execução do microprograma gerando microcomandos (sinais de controle)

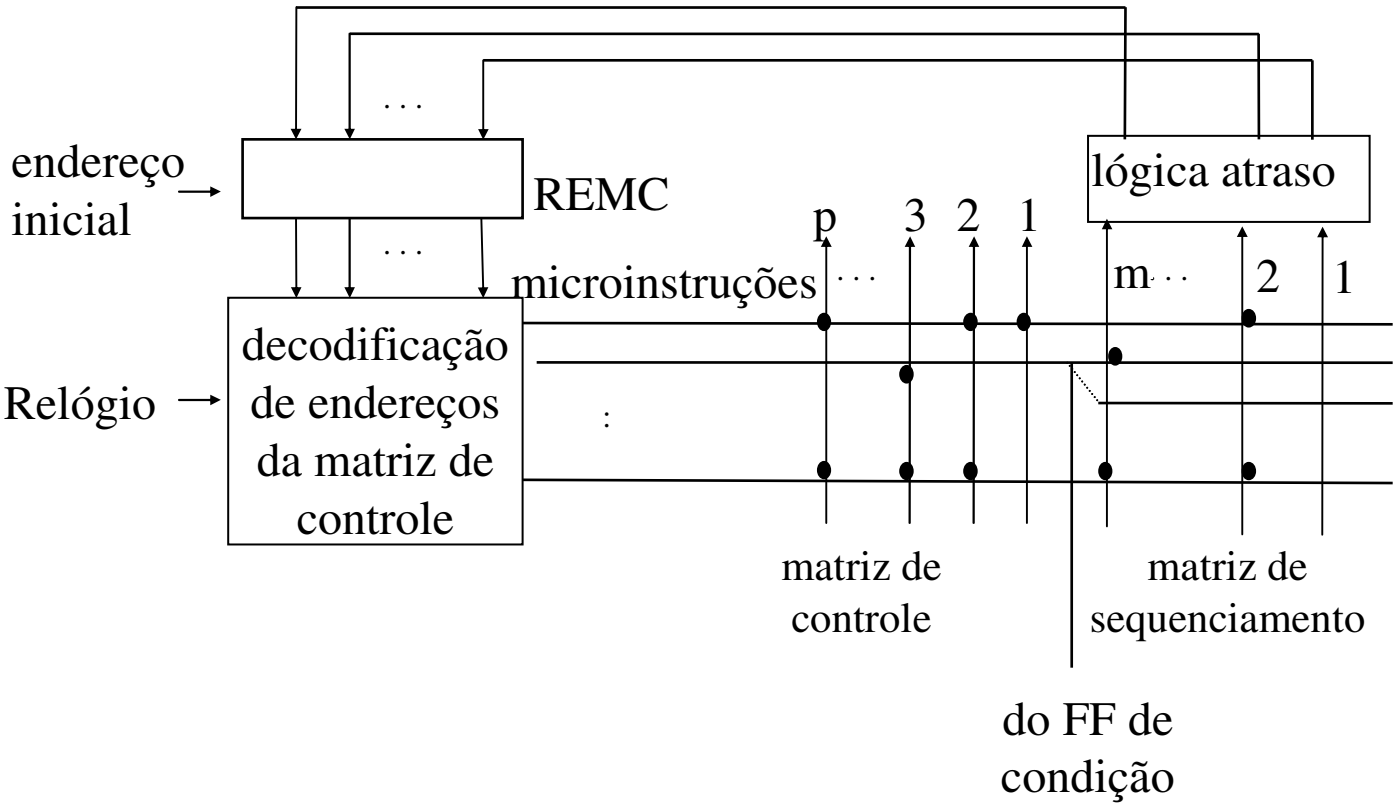


Controlador Microprogramado (cont.)

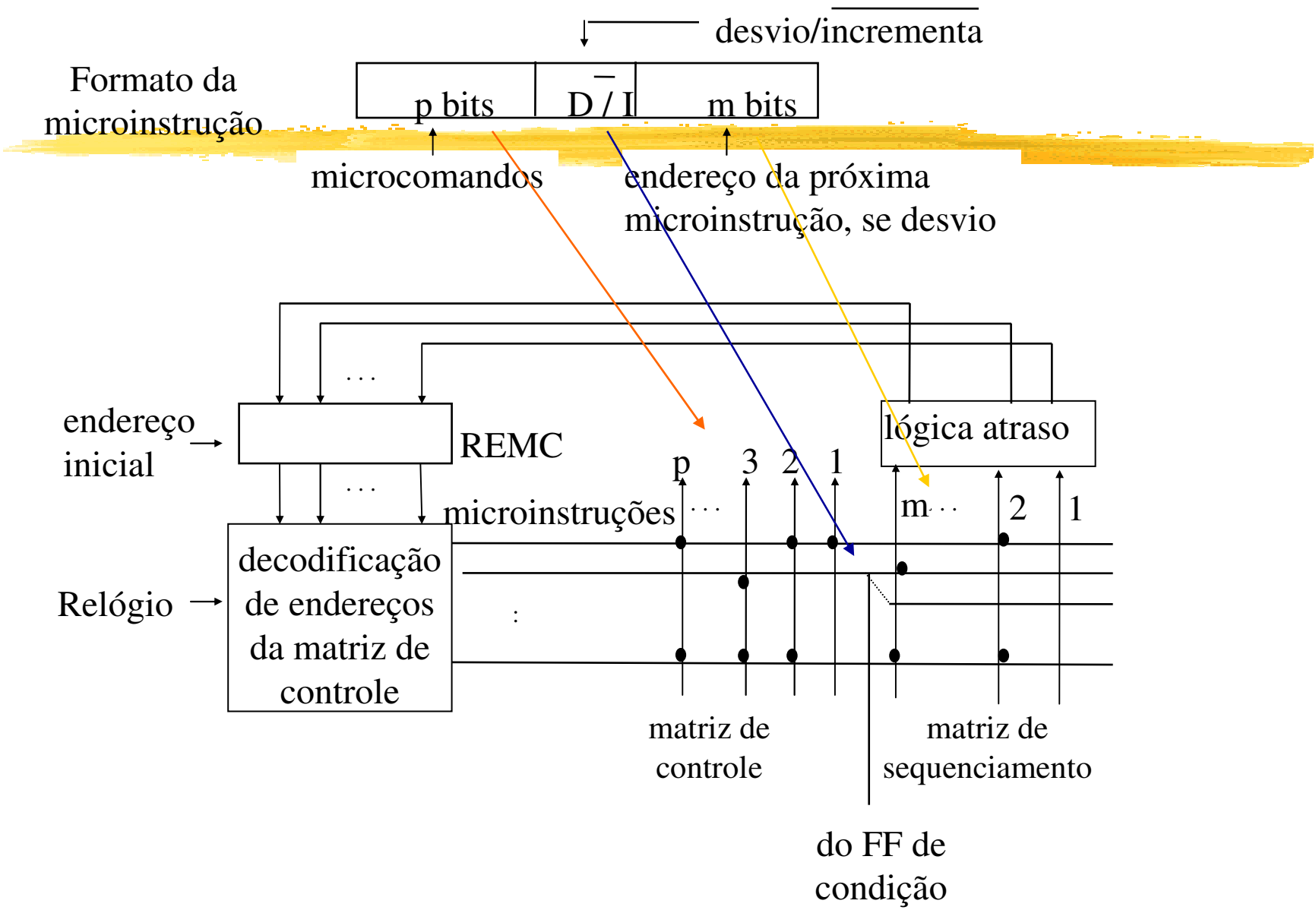


Controlador Microprogramado (cont.)

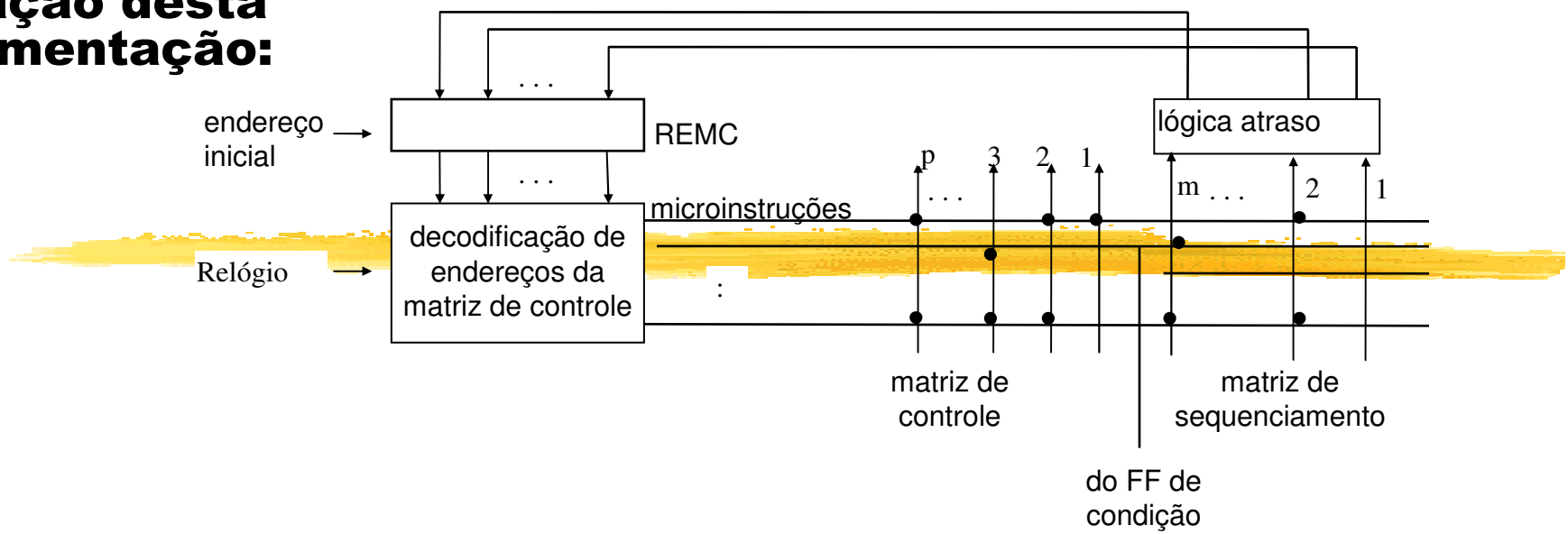




Controlador Microprogramado (cont.)



Operação desta implementação:



- inicialmente o código de operação de uma instrução de máquina é mapeado para um endereço da matriz de controle e carregado no REMC (*endereço inicial* na figura);
- do REMC o endereço é transferido para o decodificador que seleciona uma linha horizontal (microinstrução);
- o sinal de relógio ativa a linha horizontal selecionada que, por sua vez, ativa um subconjunto de linhas verticais da matriz de controle e da matriz de sequenciamento;
- a saída da matriz de controle ativa um subconjunto de microcomandos;
- a saída da matriz de sequenciamento é colocada no REMC através de uma rede de atraso para preparar o endereçamento da próxima microinstrução (o atraso é necessário para haver tempo suficiente para a execução da microinstrução corrente);
- quando o novo endereço é colocado no REMC, repete-se o ciclo inteiro.

4.3 Formato de microinstruções

Microinstruções **horizontais**, **verticais**, **diagonais**

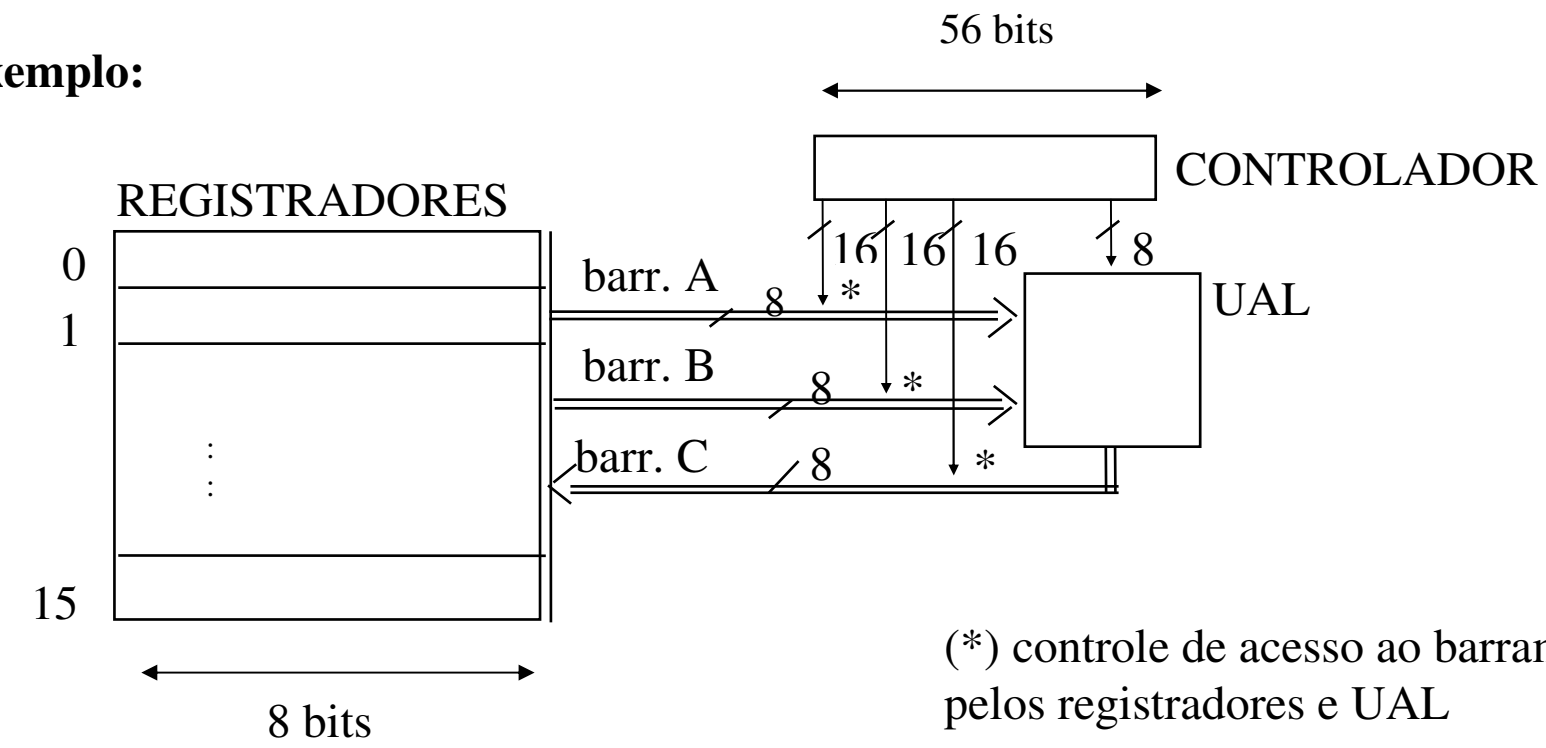
Microinstrução horizontal
permite a especificação na mesma microinstrução de todas as possíveis microoperações da máquina, associando 1 bit da microinstrução a cada microcomando existente.

Formato de microinstruções

Microinstruções **horizontais**, **verticais**, **diagonais**

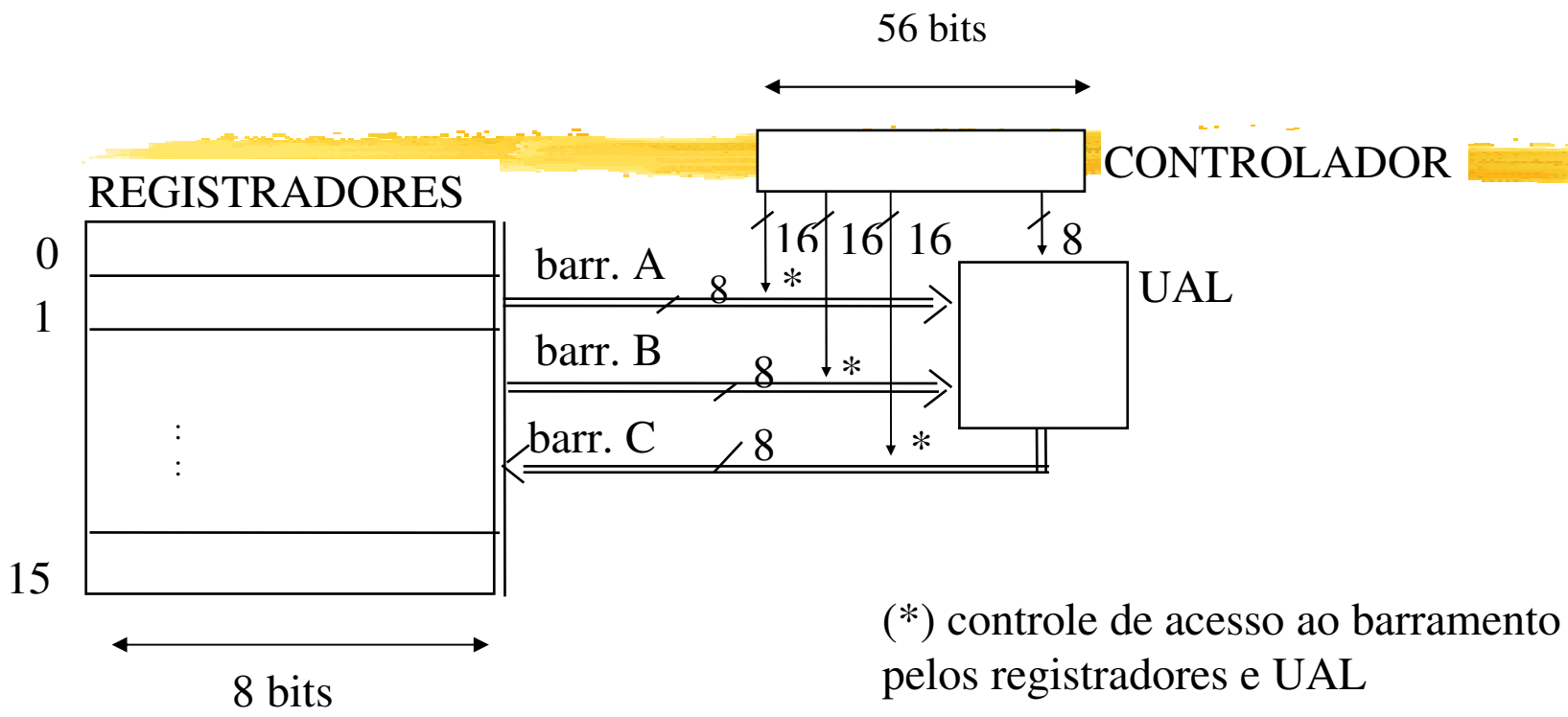
Microinstrução horizontal
permite a especificação na mesma microinstrução de todas as possíveis microoperações da máquina, associando 1 bit da microinstrução a cada microcomando existente.

Exemplo:



(*) controle de acesso ao barramento pelos registradores e UAL

Microinstrução horizontal – cont.

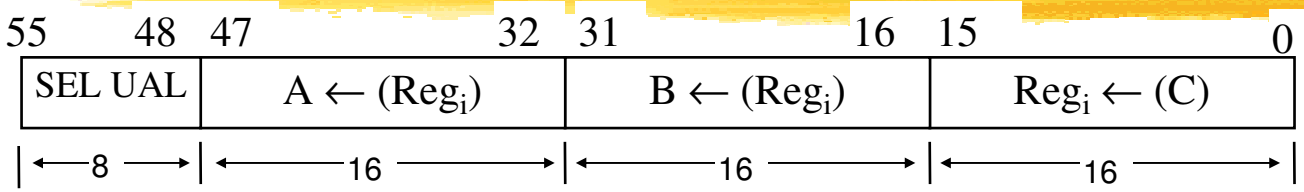


(*) controle de acesso ao barramento pelos registradores e UAL

- uma UAL que implementa 8 operações distintas;
- 16 registradores de 8 bits, numerados de 0 a 15;
- 3 barramentos de 8 bits sendo 2 deles direcionados para transferir os conteúdos de registradores como operandos da UAL, e um terceiro que transfere o resultado da operação para ser armazenado em um registrador.

Microinstrução horizontal – cont.

Formato da microinstrução (neste exemplo):



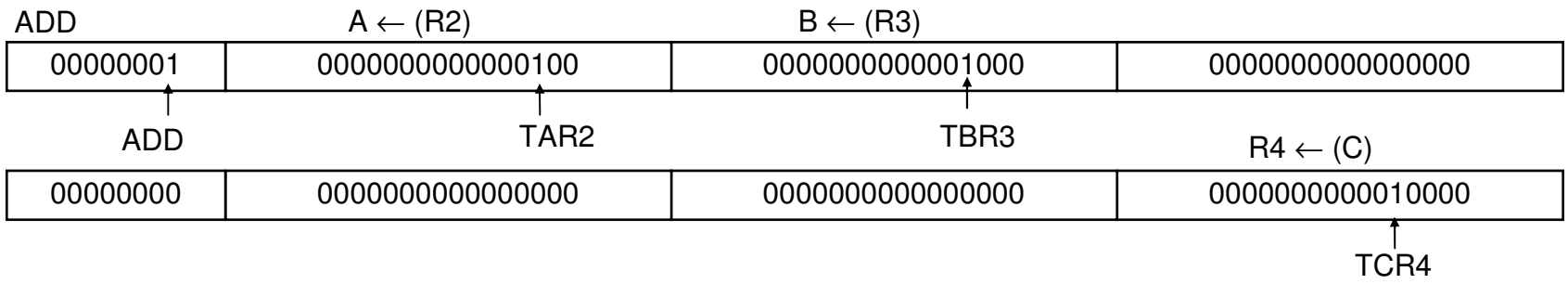
Microoperações implementadas no controlador:

1. Transferência do conteúdo de um registrador para o barramento A: **A ← (Registrador)** com 16 microcomandos, cada um referenciando um dos 16 registradores.
2. Transferência do conteúdo de um registrador para o barramento B: **B ← (Registrador)** com 16 microcomandos, cada um referenciando um dos 16 registradores.
3. Seleciona a operação da UAL: **SEL UAL** com 8 microcomandos, cada um referenciando as 8 operações permitidas para a UAL.
4. Transferência do conteúdo do barramento C para um registrador: **Registrador ← (C)** com 16 microcomandos, cada um referenciando um dos 16 registradores.

Microinstrução horizontal – cont. Exemplo

Seja a instrução **ADD R2, R3, R4 ;** **R4 ← (R2) + (R3)**

na arquitetura definida.



Deverá ser executada em dois ciclos de máquina, pois a transferência

R4 ← (C) só pode ocorrer após a adição.

Microinstrução horizontal – cont. Vantagens e desvantagens

Vantagens:
oferece maior eficiência na utilização dos recursos de hardware da máquina em função de permitir o paralelismo das microoperações que podem ser executadas simultaneamente.

Desvantagens:
desperdício da memória de controle, uma vez que a microinstrução deverá possuir um bit para cada microcomando da máquina e apenas alguns são utilizados em cada microinstrução, em função da possibilidade limitada de paralelismo entre as microoperações;

dificuldade de programação em função da grande quantidade de bits que controlam os microcomandos.

Microinstruções Verticais

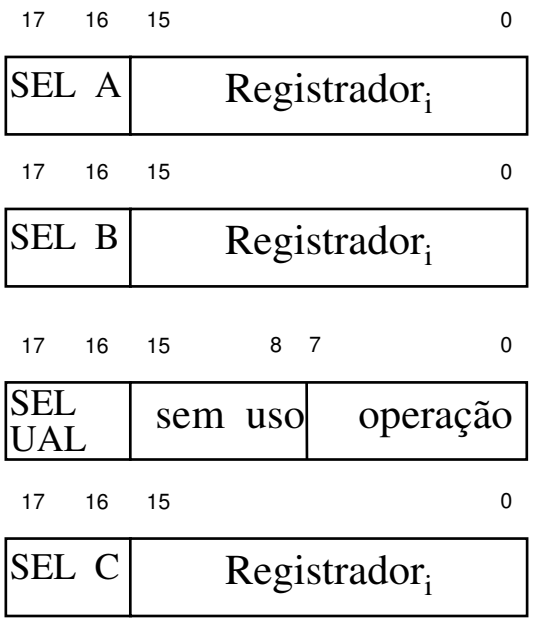
Microinstrução vertical

especifica apenas uma microoperação em cada microinstrução.

Microinstruções verticais

Exemplo: considerando a mesma arquitetura do exemplo anterior.

Microinstruções da arquitetura (note-se que o reg. controlador tem agora 18 bits)



$A \leftarrow (\text{Registrador}_i)$

$B \leftarrow (\text{Registrador}_i)$

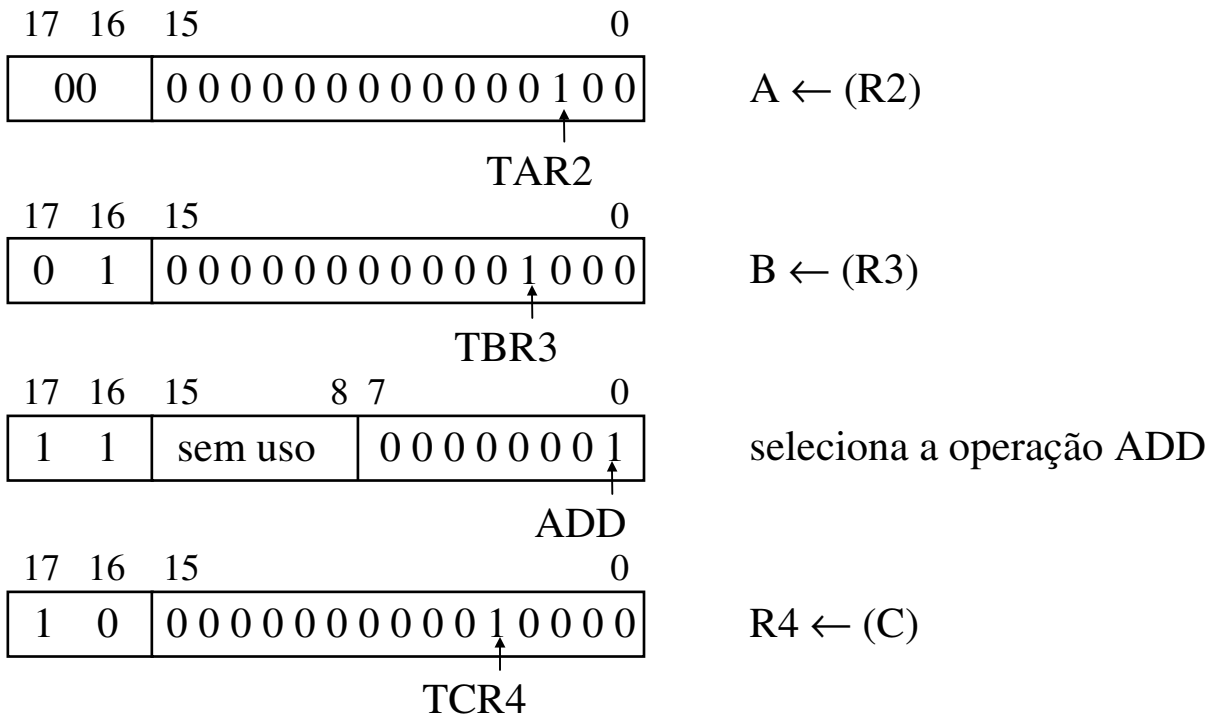
seleciona a operação da UAL

$\text{Registrador}_i \leftarrow (C)$

Microinstrução vertical – cont. Exemplo

Seja a instrução **ADD R2, R3, R4 ; R4 ← (R2) + (R3)**

Na arquitetura definida, agora necessitamos de 4 microinstruções:



Microinstrução vertical – cont. Exemplo

Observemos na figura que:

1. O mesmo bit zero controla os microcomandos TAR0, TBR0, TCR0 e ADD, dependendo de a microoperação ser transferência do registrador R0 para A, transferência de R0 para B, transferência de C para R0, ou operação ADD, respectivamente.
2. A ativação do microcomando correto depende, assim, de um hardware adicional que tem como entrada um dos bits (0 a 15) da microinstrução e o código da microoperação (bits 16 e 17 da microinstrução).

Microinstrução vertical – cont. Vantagens e desvantagens

Vantagens:

- todos os campos da microinstrução são utilizados;
- as microinstruções são mais fáceis de escrever devido ao seu formato simples.

Desvantagens:

- gera microprogramas mais longos, com maior tempo de execução;
- não permite exploração do possível paralelismo de microoperações.

Microinstrução diagonal

A microinstrução diagonal é um compromisso entre a microinstrução horizontal e a microinstrução vertical. Alguns microcomandos são associados a bits de uma microinstrução individualmente (como em configurações horizontais) e outros, a um conjunto de bits (como em configurações verticais).

4.4 Execução de microinstruções

Podemos subdividir a execução de uma microinstrução em vários ciclos de relógio - Fases de Relógio - associando cada Fase de Relógio à execução de microoperações especificadas em campos (partes) determinados da microinstrução.

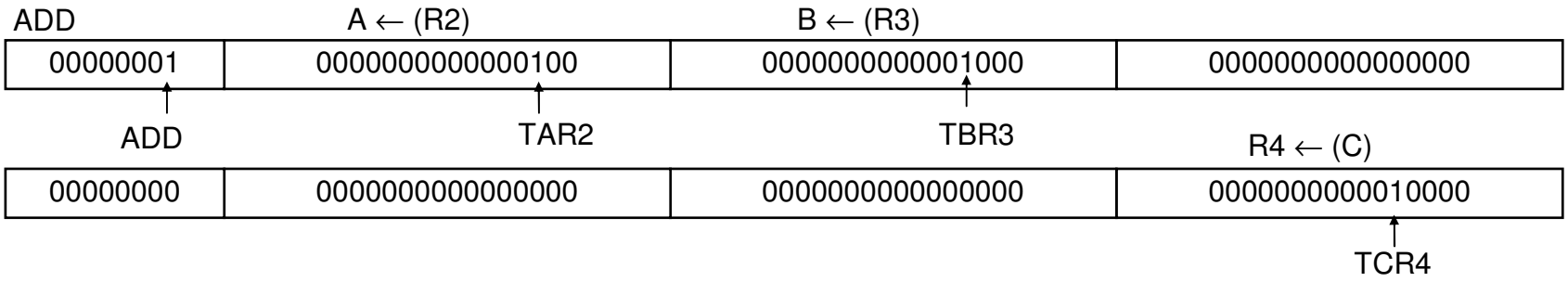
Classificação das microinstruções:

Monofásicas: quando todas as microoperações associadas à microinstrução são executadas na mesma fase do relógio. Todos os exemplos apresentados foram de microinstruções monofásicas.

Polifásicas: quando o ciclo de execução da microinstrução é dividido em várias fases do relógio, permitindo a execução sequencial de microoperações não paralelas da microinstrução. Em cada fase, contudo, só um subconjunto de microoperações pode ser executado concorrentemente. As fases do relógio estarão associadas aos campos da microinstrução.

Monofásicas

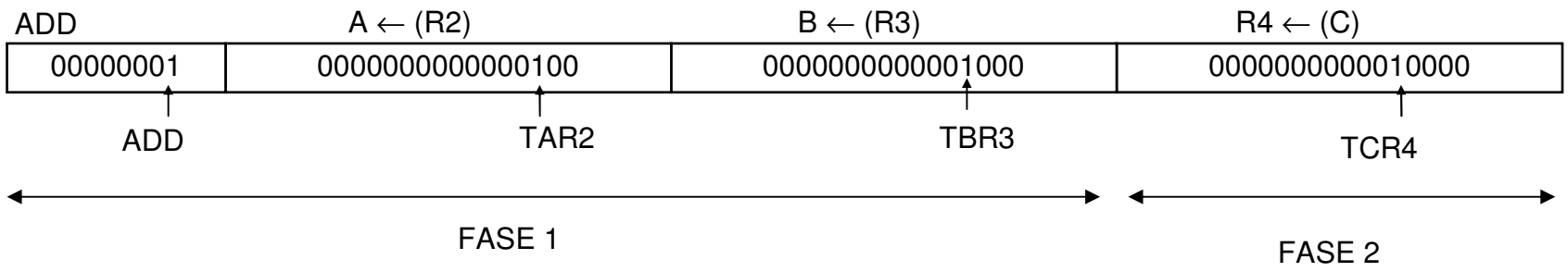
Seja a instrução **ADD R2, R3, R4 ; R4 ← (R2) + (R3)**



Deverá ser executada em dois ciclos de máquina, pois a transferência R4 ← (C) só pode ocorrer após a adição. Na verdade existem **duas** microinstruções.

Polifásicas
Exemplo 1

Seja a instrução **ADD R2, R3, R4 ; R4 ← (R2) + (R3)**



Continuará sendo executada em dois ciclos de máquina, pois a transferência R4 ← (C) só pode ocorrer após a adição.

No entanto só é necessária **uma** microinstrução.

Polifásicas
Exemplo 2

Implementação da busca de uma instrução com microinstruções horizontais polifásicas.

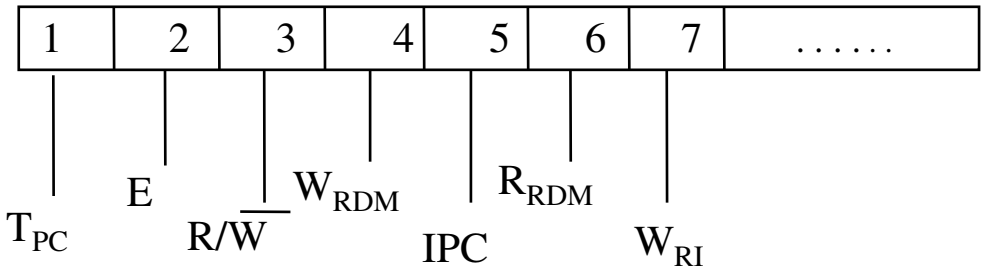
Seja o ciclo de busca de uma instrução definido pela Tabela a seguir :

Tabela: Microoperações e microcomandos para a execução da busca de uma instrução

microoperações	microcomandos
1. REM ← (PC)	T_{PC}
2. RDM ← ((REM))	$E, R/\overline{W}, W_{RDM}$
3. PC ← (PC) + 1	IPC
4. RI ← (RDM)	R_{RDM}, W_{RI}

Polifásicas
Exemplo 2 – cont.

Para uma **microinstrução horizontal e monofásica** teremos a seguinte representação:



A implementação com uma única fase de relógio está mostrada na figura. São necessárias 3 microinstruções pois somente as microoperações 2 e 3 podem ser concorrentes.

Busca:

1	1	0	0	0	0	0	0	. . . . .
2 e 3	0	1	1	1	1	0	0	. . . . .
4	0	0	0	0	0	1	1	. . . . .

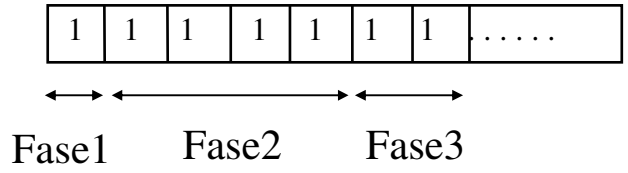
Polifásicas
Exemplo 2 – cont.

Para uma **microinstrução horizontal e polifásica** teremos a seguinte representação:

Definido-se a associação dos campos às fases como : fase 1 - bit 1
fase 2 - bits de 2 a 5
fase 3 - bits de 6 a 7

teremos:

Busca:



Polifásicas
Síncronas e Assíncronas

Microinstruções polifásicas podem ser, por sua vez, classificadas como:

Síncronas: quando o número de fases é fixo para todas as microinstruções. Isto implica no mesmo tempo de execução para qualquer microinstrução.

Assíncronas: quando o número de fases depende da microinstrução. Isto implica que o tempo de execução depende da microinstrução. É de implementação mais complexa.

4.5 Exercícios sobre microprogramação

1. Resolva o exercício (17) do livro texto.