

Curso: Engenharia de Computação

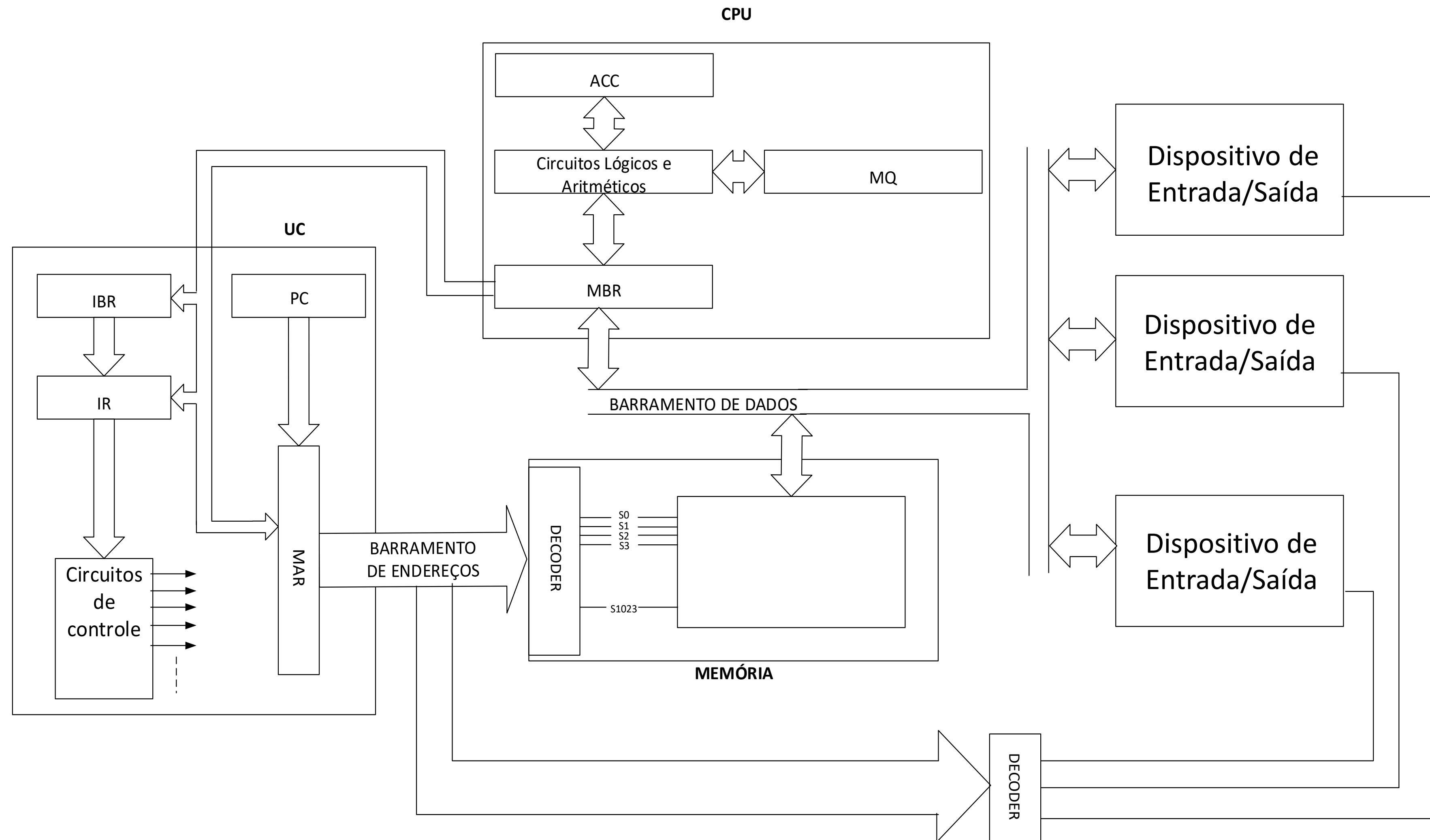
Arquitetura de Computadores

Prof. Clayton J A Silva, MSc
clayton.silva@professores.ibmec.edu.br



IAS - a máquina de von Neuman

Referência para as arquiteturas modernas



Quais são os
elementos
das
arquiteturas
modernas?



Elementos do sistema de computação

- Memória
- Barramento
- Processadores
- Dispositivos de Entrada/Saída

Elementos do sistema de computação

- **Memória**
- Barramento
- Processadores
- Dispositivos de Entrada/Saída

Sistema de memória

Parâmetros de avaliação

- Capacidade de armazenamento de bits
- Velocidade
- Custo

Sistema de memória

ibmec.br

Capacidade de armazenamento

- Quantidade total de bits ou bytes (B) armazenados em cada elemento, medidos normalmente em múltiplos binários: K, M, G ou T, por exemplo.
- $1 \text{ KB} = 2^{10} \text{ B} = 1024 \times 8 \text{ bits}$
- $1 \text{ MB} = 2^{20} \text{ B} = 1024 \times 1024 \times 8 \text{ bits}$
- ...

Sistema de memória

ibmec.br

Velocidade

- **Tempo de acesso** - representa o tempo que a memória gasta para disponibilizar um dado após o endereço ser disponibilizado e ser requisitado pelo processador.
- **Ciclo de memória (latência)** - representa o tempo decorrido entre duas operações sucessivas de acesso à memória, seja para leitura ou escrita.
- O ciclo de memória compreende o tempo de acesso mais o tempo necessário para a realização de outras operações do sistema, logo $tC = tA + tS$.

Características tecnológicas

ibmec.br

1. Acesso **aleatório** aos dados x **sequencial** aos dados armazenados

Acesso Randômico (aleatório): Tempo de acesso **igual para todas as células**. Não há necessidade de ler outras posições antes de acessar uma célula.

Acesso sequencial: Existem tecnologias em que o acesso a uma célula requer uma **sucessão preliminar de posições** antes de acessar a **posição de interesse**.

Características tecnológicas

ibmec.br

2. **Volatilidade** x **não volatilidade** do armazenamento sem alimentação

Memórias **voláteis** mantêm os dados armazenados somente se houver alimentação elétrica.

Memórias **não voláteis** mantêm os dados armazenados **mesmo na ausência de alimentação** elétrica.

Características tecnológicas

ibmec.br

3. Necessidade de recarga (*refresh*) dos dados armazenados

Memórias **estáticas** - o dado permanece armazenado uma vez carregado.

Memórias **dinâmicas** - para o dado se manter armazenado é necessário periodicamente uma recarga (*refresh*).

Características tecnológicas

ibmec.br

4. Controle sincronizado ou não

Memórias **assíncronas** - o dado é disponibilizado para leitura mediante um **signal de controle** do processador

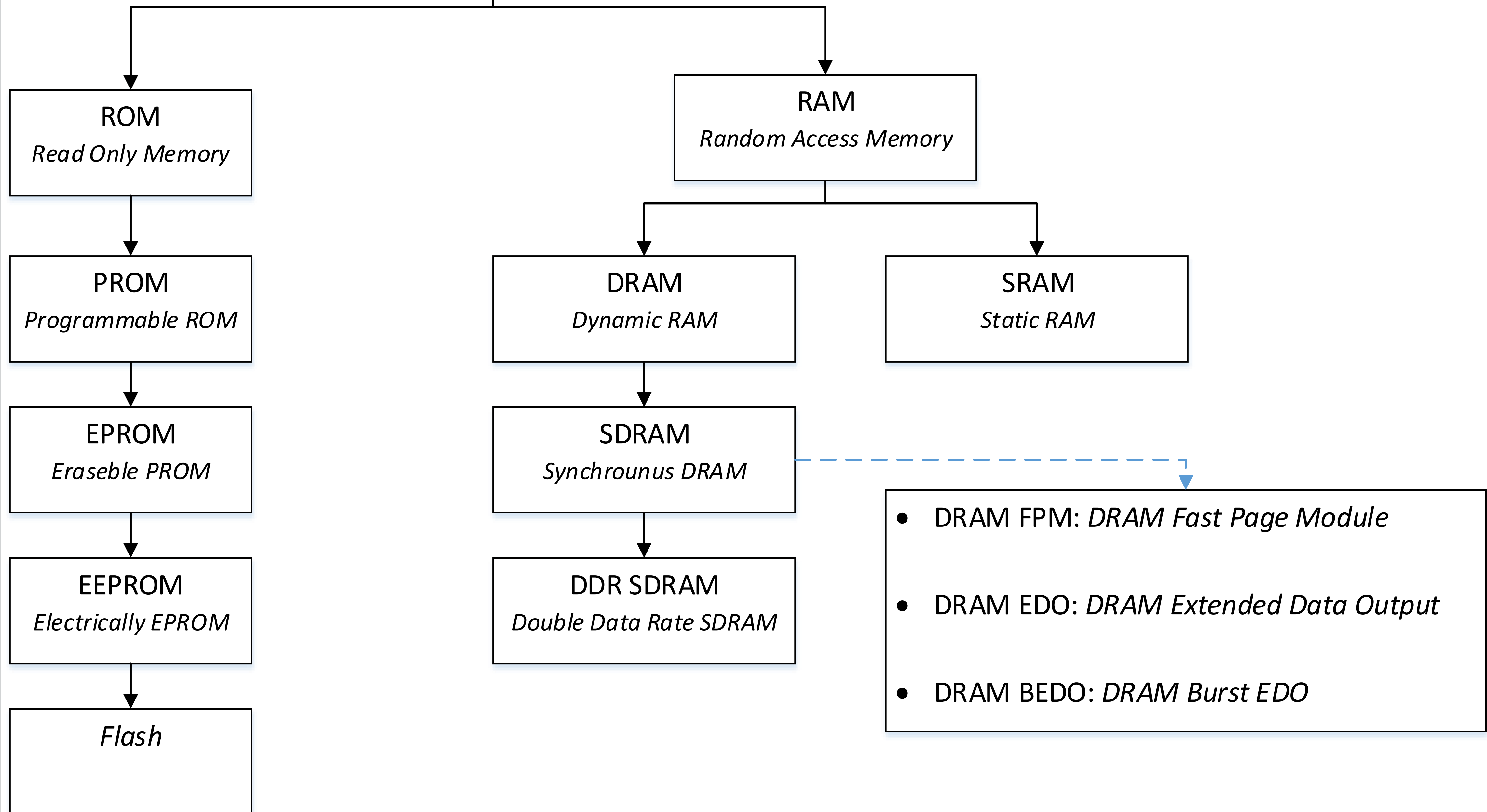
Memórias **síncronas** - a operação acompanha a velocidade do *clock* do sistema, sem a necessidade de sinal de controle.

Características tecnológicas

5. Material de construção

- A construção dos dispositivos de memória pode ser feita com **semicondutores**.
- São duas tecnologias muito utilizadas: **MOS**, semicondutor de óxido metálico e **CMOS**, semicondutor de óxido complementar.
- Além disso, podem ser construídas com meios **magnéticos** e meios **ópticos**.

TECNOLOGIAS



Tipos

Memórias RAM

- As RAMs são tipicamente:
 - voláteis
 - estáticas (SRAM) ou dinâmicas (DRAM)
 - assíncronas ou síncronas
 - Evolução:

SDRAM => SDRAM DDR, ...

Tipos

Memórias RAM

- Aplicações
 - Armazenamento temporário dos dados processados

Tipos

Memórias ROM

- As ROMs são tipicamente:

- Não voláteis
- Estáticas
- Assíncronas

Evolução:

ROM => (PROM) ROM programável => (EPROM) PROM apagável => (EEPROM) EPROM apagável eletricamente => *flash* - EEPROM reprogramada no próprio circuito.

Tipos

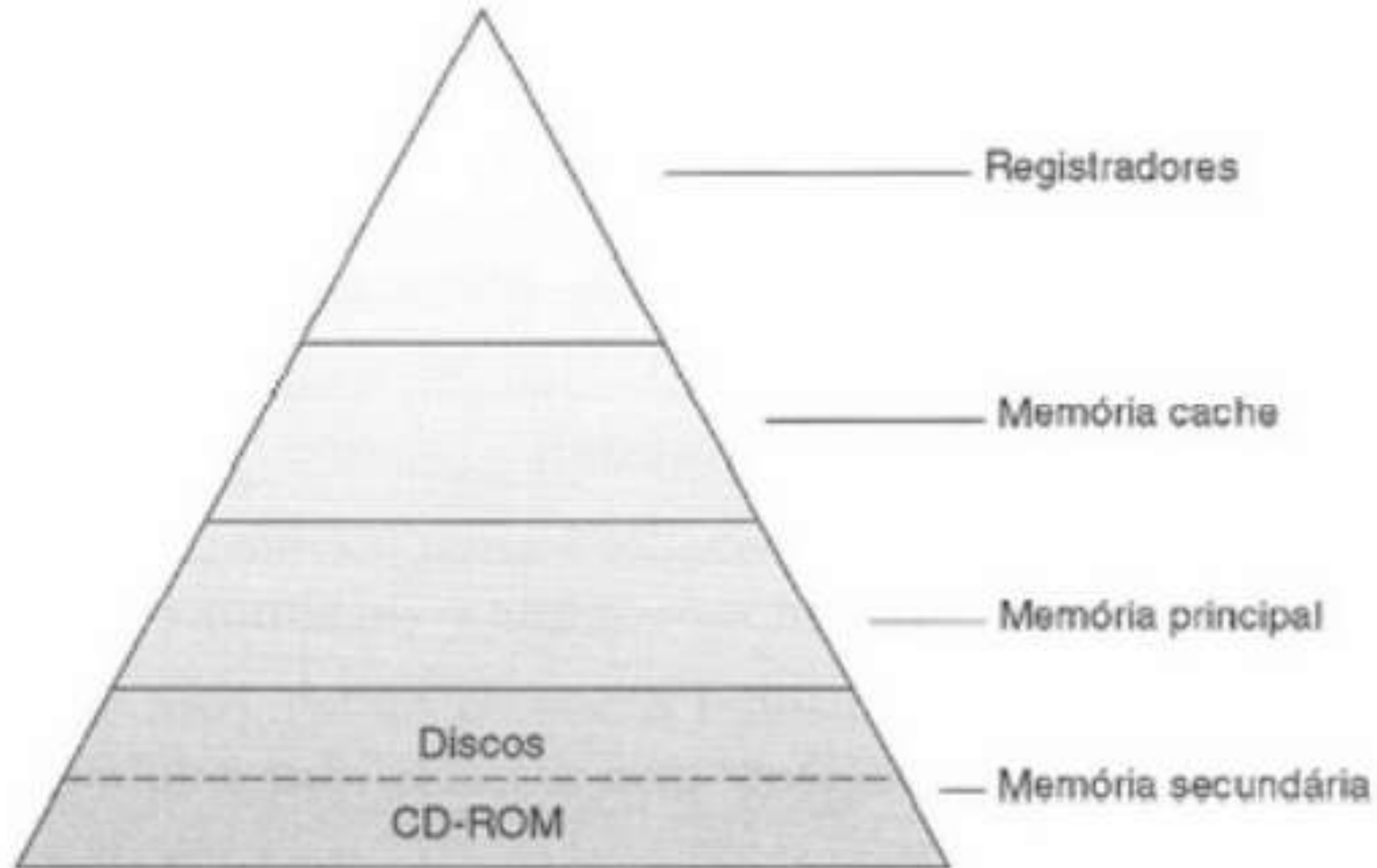
Memórias ROM

- Aplicações

- *Firmware*: inicialização de máquinas, hardware+software
- *Bootloader*: inicialização do sistema operacional
- Tabelas de dados
- Conversores de códigos
- Geradores de funções
- Armazenamento auxiliar

Hierarquia do sistema de memória

ibmec.br



Hierarquia do sistema de memória

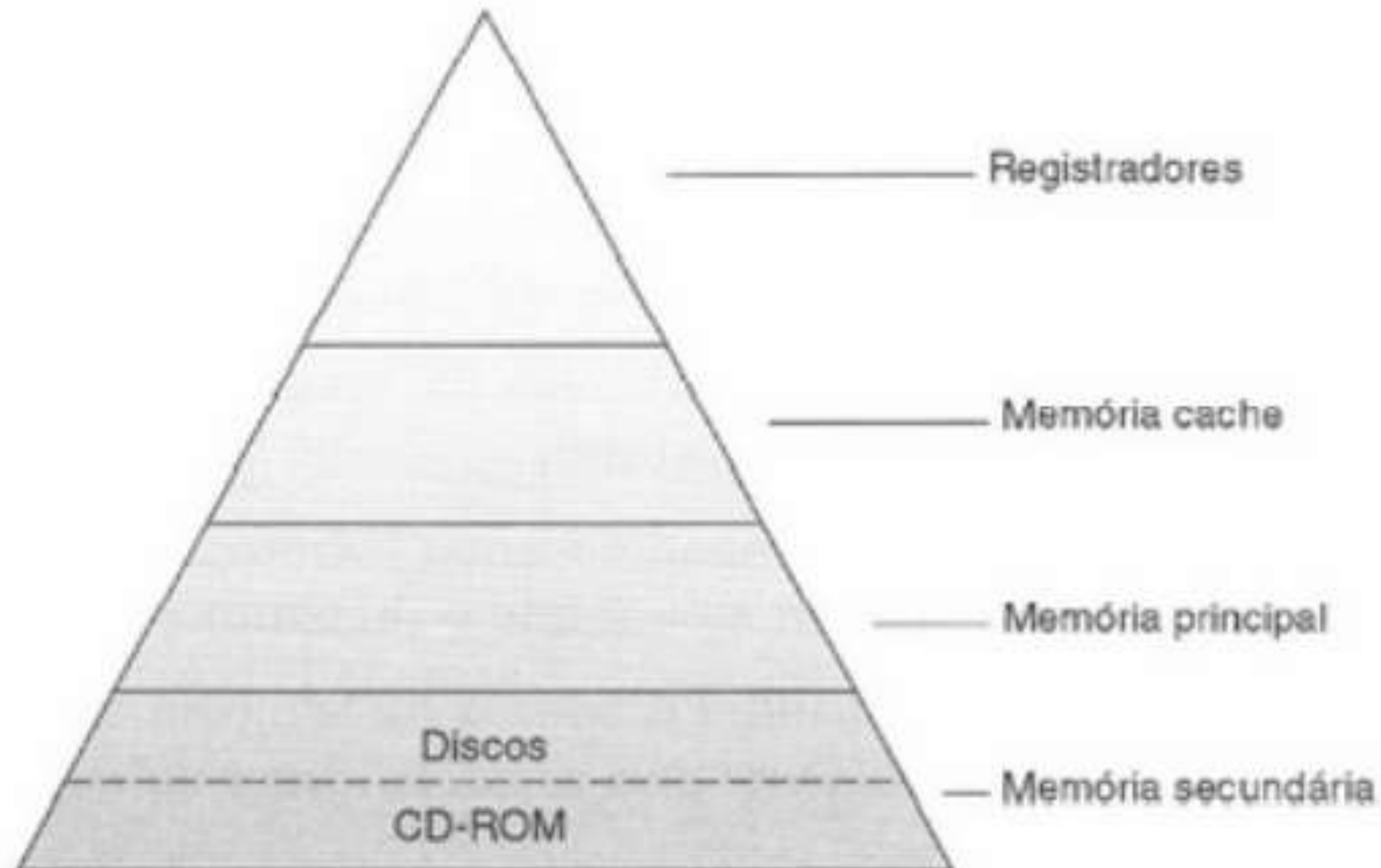
ibmec.br

Desempenho

Custo alto
Velocidade alta
Baixa capacidade



Custo baixo
Velocidade baixa
Capacidade elevada



Sistema de memória

ibmec.br

Desempenho

Tipo	Tempo de Acesso	Tamanho	Custo (por MB)
Registradores	Ciclos CPU	32-64 bits	---
L1	Ciclos CPU	32-64 Kbytes	---
L2	8-35 ns	512Kbytes -2 Mbytes	50 Us\$
principal	40-120 ns	64 Mbytes – 1 Gbyte	1 Us\$
secundária	5 ms	6 Gbytes-128 Gbytes	0.02 Us\$

 **ibmec**

Hierarquia do sistema de memória

Tecnologias

- As **memórias secundárias** são construídas em meios ópticos e magnéticos (fitas, discos, ...) e sua função é **armazenar grandes quantidades de dados**, com a desvantagem de possuir **tempo de acesso e de ciclo lentos**, porém a um **custo baixo**
- O acesso aos dados é sequencial e mecânico
- **Memórias SSD** são eletrônicas, utilizam tecnologia mais moderna, mas desempenham na hierarquia do sistema de memória o mesmo papel dos HD (*hard disk*)

Hierarquia do sistema de memória

Tecnologias

	HD	SSD	SSD NVMe
Velocidade média de leitura e escrita	30 a 150MB/s	500MB/s	3.000MB/s
Conexões	SATA III	SATA III ou M.2	M.2
Custo médio por GB em um modelo de 500GB*	R\$ 0,31	R\$ 1,56	R\$ 2,33

Fonte: [O que é SSD e como ele funciona? Saiba tudo sobre esse componente \(buscape.com.br\)](http://buscape.com.br)

Hierarquia do sistema de memória

Tecnologias

- A **memória principal** é construída em **semicondutor**, com tecnologia de **RAMs dinâmicas**, possui **média capacidade de armazenamento e custo**.
- Sua principal função é armazenar os dados do programa de máquina e os dados armazenados a serem acessados pelo processador.
- Veja alguns tipos no mercado em <[Memória RAM: 7 melhores do mercado em 2021 - Canaltech](#)>

Hierarquia do sistema de memória

Tecnologias

- A **memória cache** é construída em **semicondutor**, internamente ao próprio processador ou não, com **tecnologia estática, média/baixa capacidade de armazenamento, custo elevado**.
- Sua principal função é armazenar os dados do programa de máquina e os dados armazenados **local e temporalmente mais prováveis de serem acessados** pelo processador.
- Utiliza o chamado **princípio de localidade**.

Hierarquia do sistema de memória

Tecnologias

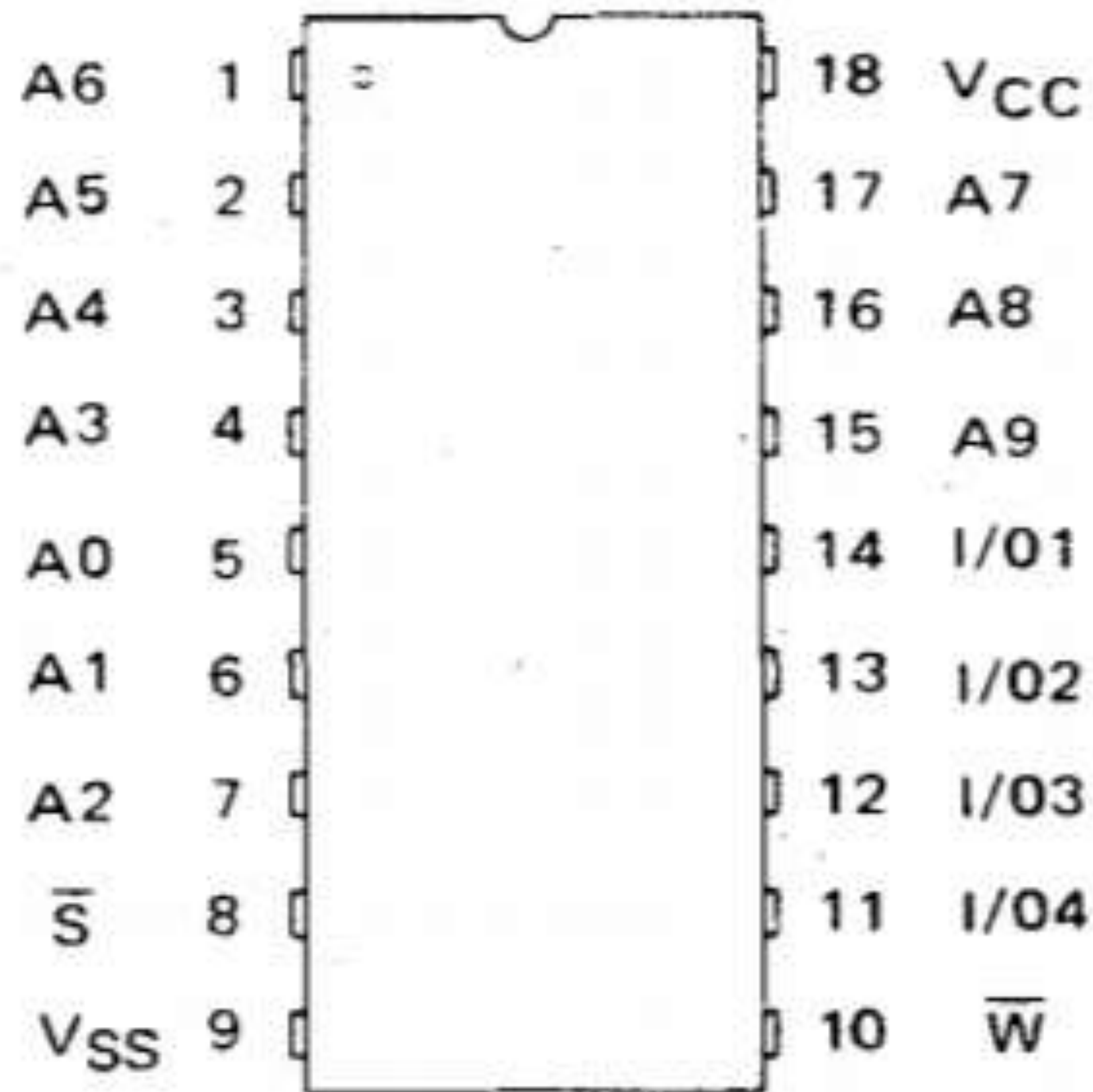
- Os registradores possuem capacidade de armazenamento de **dezenas de unidade de bits**
- São construídos em dispositivos **semicondutores** e
- localizam-se internamente ao processador.

Memória Primária

Memória Principal

- Armazena programas (sequência de instruções) e dados.
- Os programas executados pela CPU contemplam um conjunto de instruções no nível do *set* de instruções do processador.
- Assim como as instruções do programa, os dados operados pela CPU também ocupam um endereço da MP.

Organização da MP



PIN NAMES	
A0 - A9	Address Input
\overline{W}	Write Enable
\overline{S}	Chip Select
I/O1 - I/O4	Data Input/Output
V _{CC}	Power (+5 V)
V _{SS}	Ground

Memória Motorola MCM2114 de 4096 bits (a) Pinagem (b) Identificação dos pinos

Organização da MP

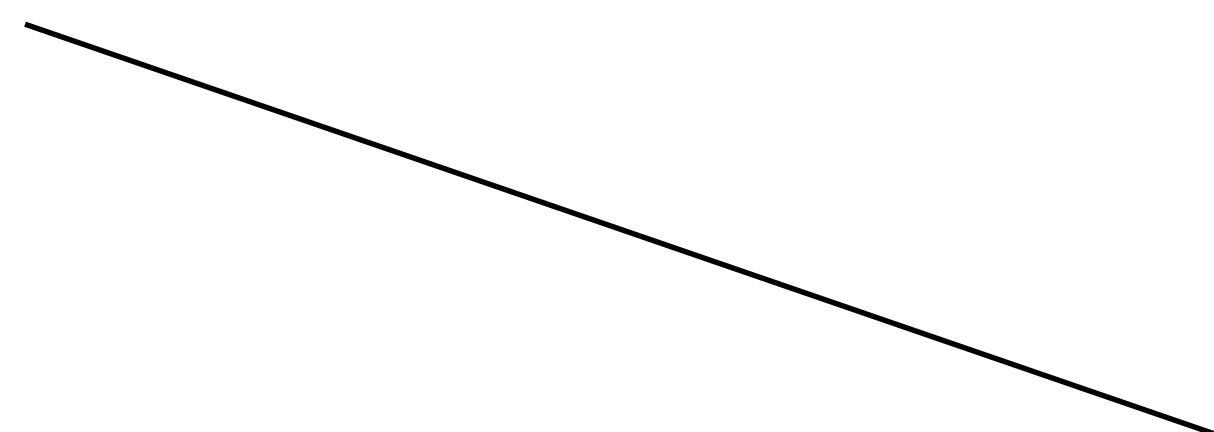
10-bits de **endereço** (*address*)

4-bits de **dados**

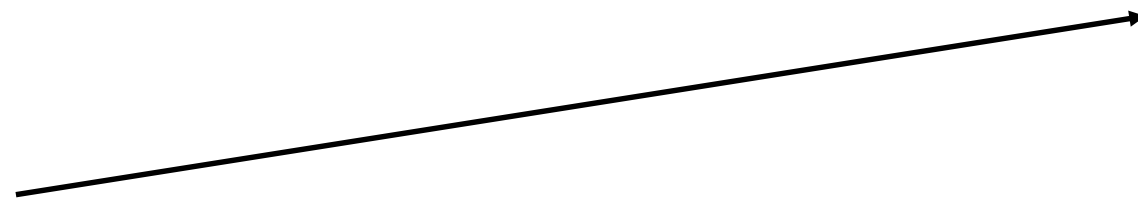
PIN NAMES	
A0 - A9	Address Input
\overline{W}	Write Enable
\overline{S}	Chip Select
I/O1 - I/O4	Data Input/Output
VCC	Power (+5 V)
VSS	Ground

Organização da MP

Tipo de operação



Alimentação elétrica



PIN NAMES	
A0 - A9	Address Input
\overline{W}	Write Enable
\overline{S}	Chip Select
I/O1 - I/O4	Data Input/Output
V _{CC}	Power (+5 V)
V _{SS}	Ground

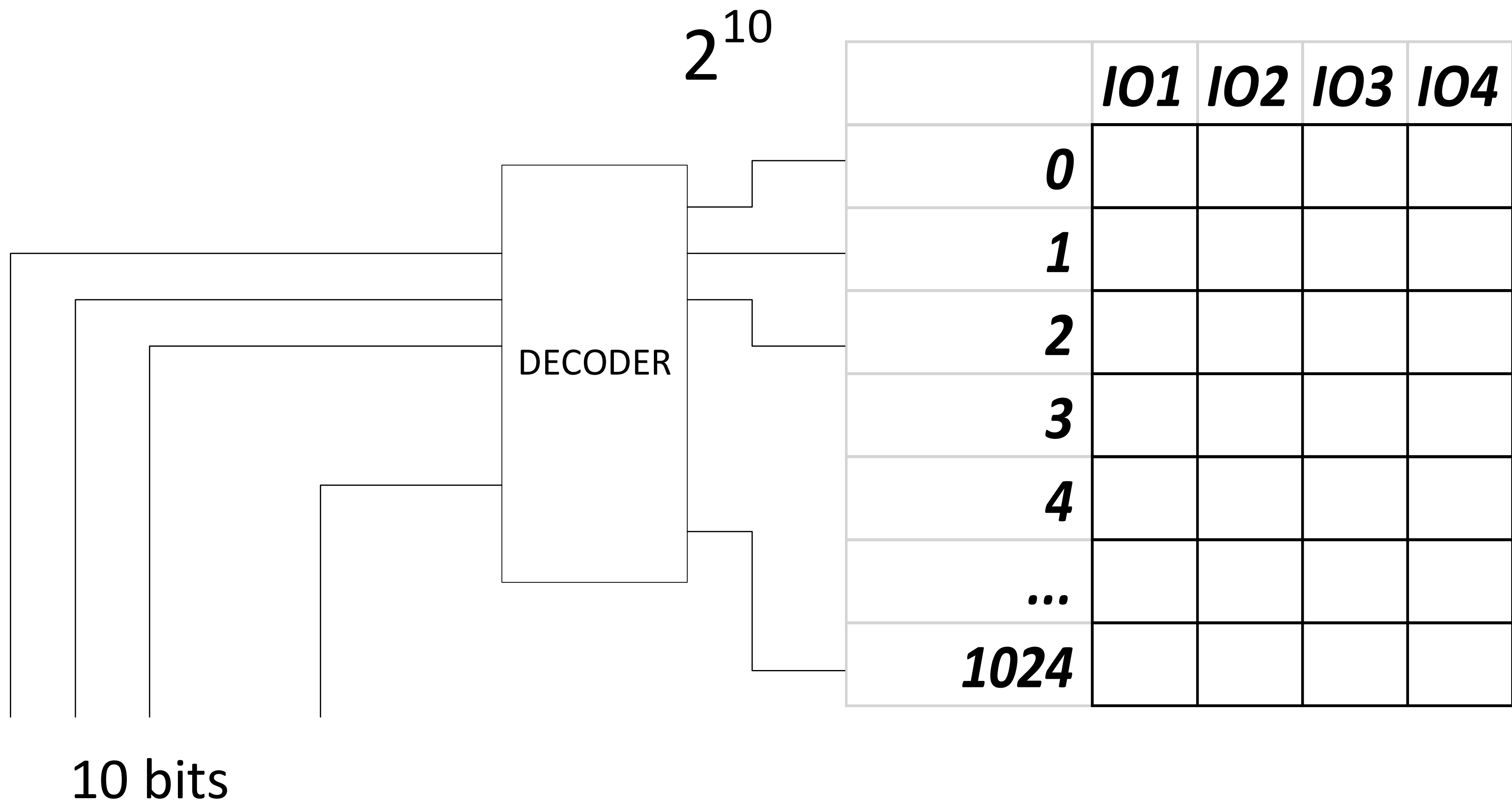
Organização da MP

4-bits de dados

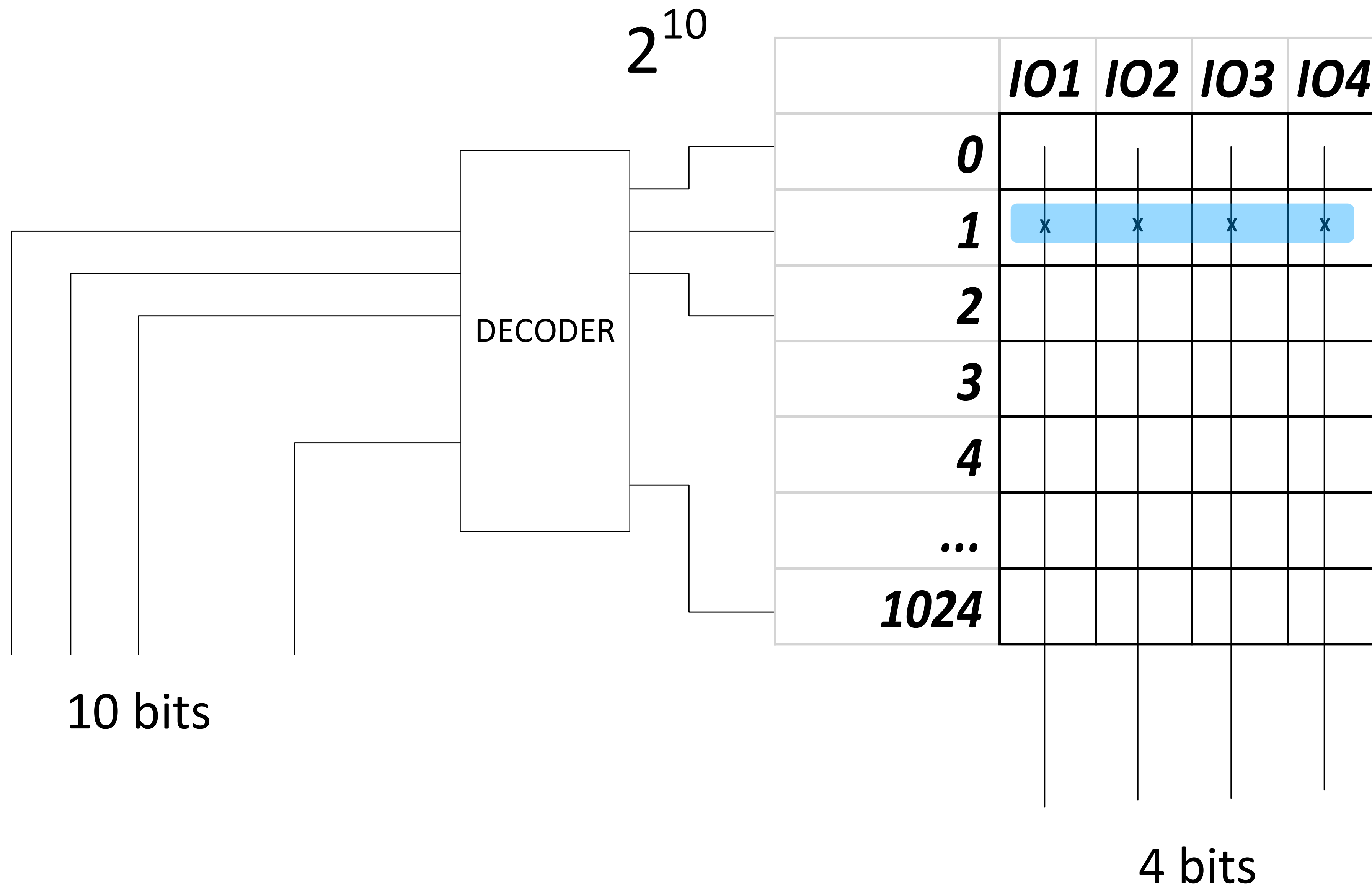
10-bits de **endereçamento** (*adress*)

	<i>IO1</i>	<i>IO2</i>	<i>IO3</i>	<i>IO4</i>
<i>0</i>				
<i>1</i>				
<i>2</i>				
<i>3</i>				
<i>4</i>				
<i>...</i>				
<i>1024</i>				

Organização da MP



Organização da MP



Organização da MP

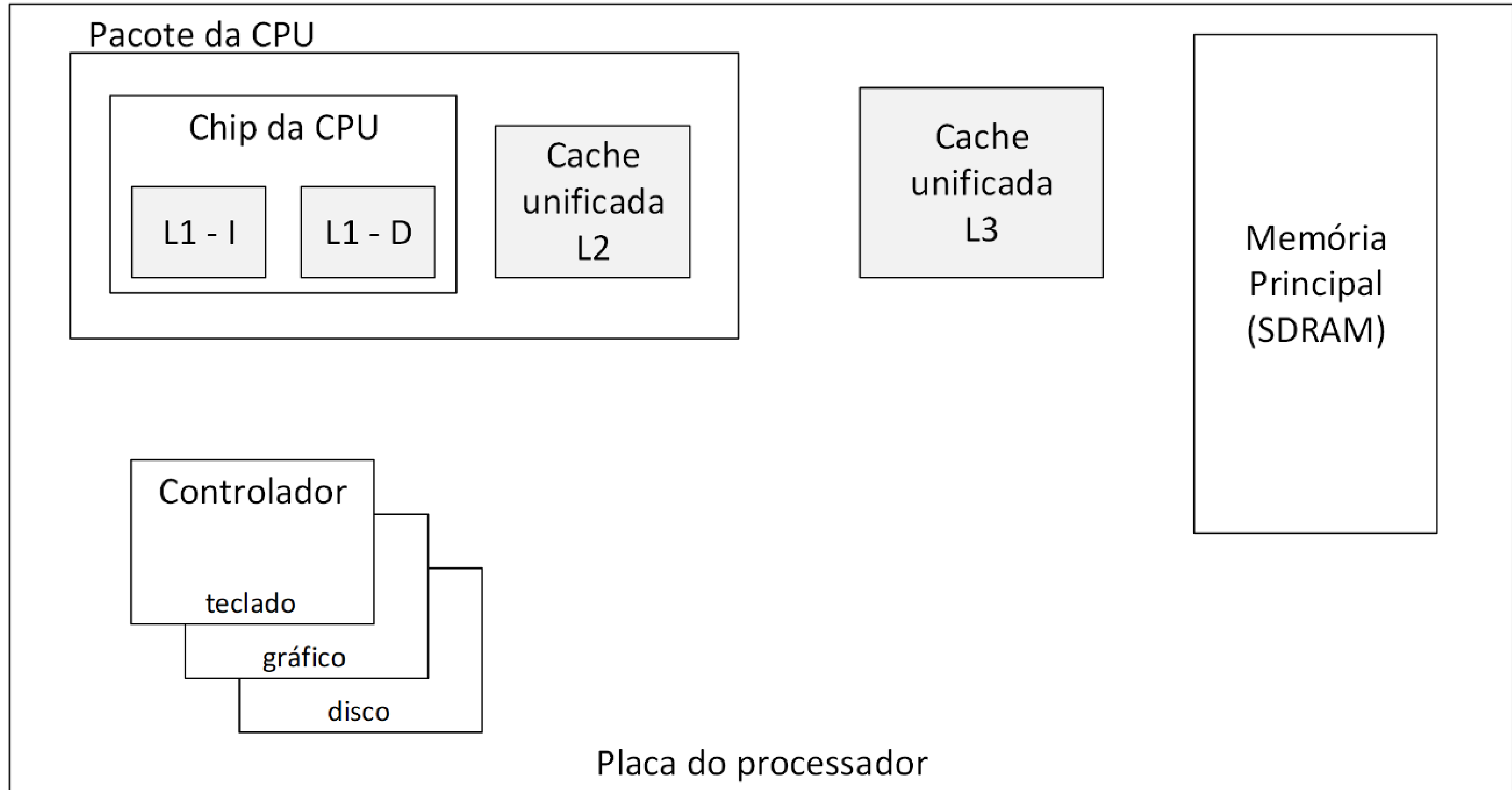
End MP	Dado
0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
10	
11	
12	
13	
14	
15	
16	
17	
18	
19	
20	
21	
22	
23	
24	
25	
26	
27	
28	
29	
30	
31	
32	
33	
34	
35	
36	
37	
38	
39	
40	
41	
42	
43	
44	
45	
46	
47	

Memória cache

Memória cache

- Memória rápida, porém média/baixa capacidade de armazenamento
- Armazena palavras da MP localmente próximas física ou temporalmente
- Princípio da localidade: espacial ou temporal

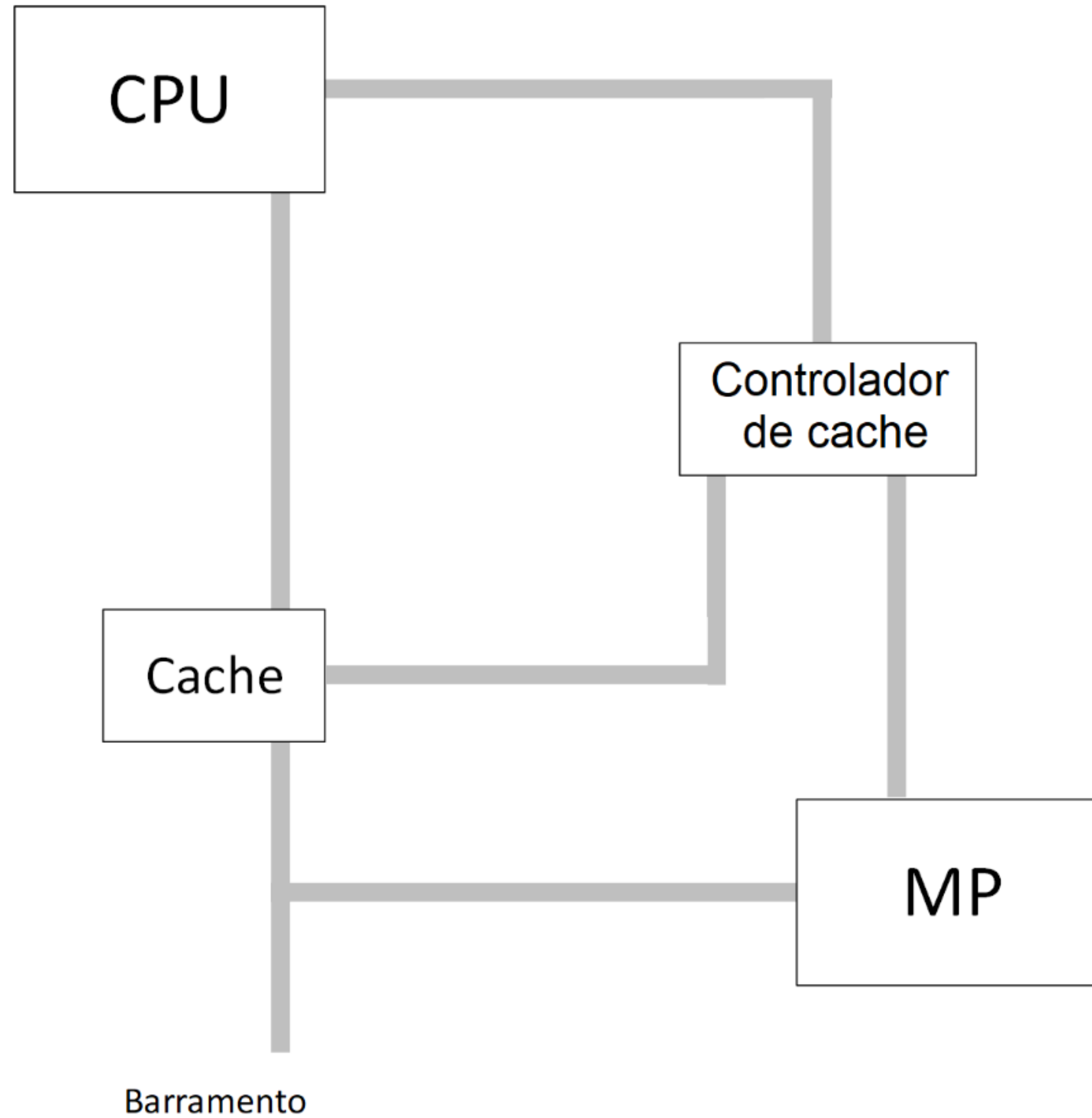
Organização da cache



Organização da cache

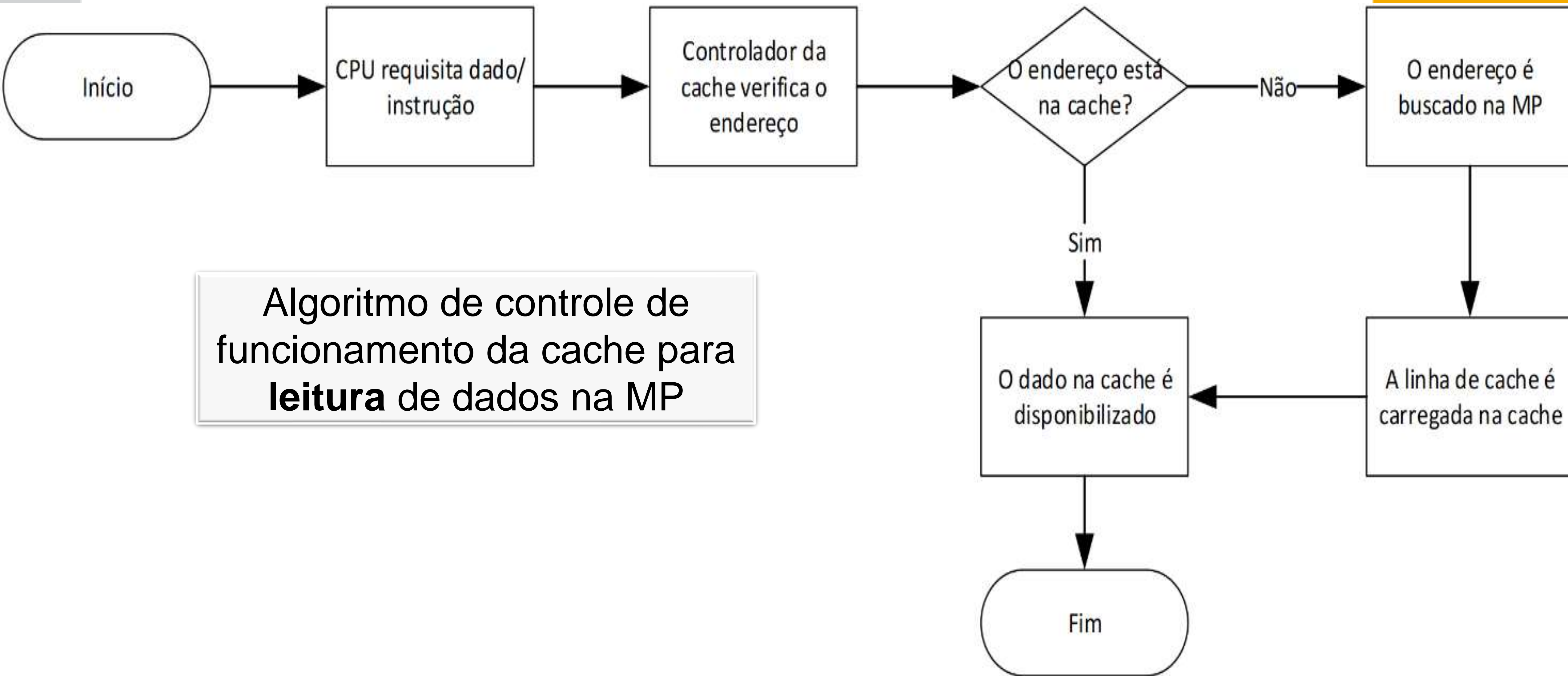
- cache interna ao encapsulamento do processador, chamada **cache de nível 1** (L1) - em L1, instruções e dados do programa de máquina são separados.
- cache no pacote da CPU – nos circuitos diretamente ligados ao processador, **de nível 2** (L2) - instruções e dados de L3 são armazenados de modo a estarem disponíveis via barramento local; e
- cache externa de **nível 3** (L3) - instruções e dados são armazenados reduzindo o tempo de resposta à memória MP.

Organização da cache



controle da cache em leitura de dados

ibmec.br



mapeamento de dados MP-cache

- Existem três técnicas de mapeamento: *mapeamento direto*; mapeamento totalmente associativo; e mapeamento associativo por conjunto.
- No mapeamento direto, o *relacionamento é predefinido* de linhas de cache da MP com os blocos da cache. O controlador de cache possui uma tabela que estabelece a associação.

Organização da MP

- 1. Divisão das ***N*** linhas de endereço de MP em ***M*** linhas de cache
- 2. Cada linha da cache com ***K*** palavras
- 3. Endereçamento de dado:
 - *End(MP)*, posição na MP: 0 a *N-1*
 - *End(LC)*, linha de cache: 0 a *M-1*
 - *End(pos/LC)*, posição na linha de cache: 0 a *K-1*

***N* = 48, *M* = 12, *K* = 4**

End MP	Linha cache	pos/LC	Dado
0	0	0	
1		1	
2		2	
3		3	
4	1	0	
5		1	
6		2	
7		3	
8	2	0	
9		1	
10		2	
11		3	
12	3	0	
13		1	
14		2	
15		3	
16	4	0	
17		1	
18		2	
19		3	
20	5	0	
21		1	
22		2	
23		3	
24	6	0	
25		1	
26		2	
27		3	
28	7	0	
29		1	
30		2	
31		3	
32	8	0	
33		1	
34		2	
35		3	
36	9	0	
37		1	
38		2	
39		3	
40	10	0	
41		1	
42		2	
43		3	
44	11	0	
45		1	
46		2	
47		3	

Organização da cache

1. Divisão das **C** linhas de endereço de cache em **B** blocos
2. Cada bloco da cache com **K** palavras
3. Endereçamento do dado:
 - $End(C)$, posição na cache: 0 a $C-1$
 - $End(B)$, bloco de cache: 0 a $B-1$
 - $End(pos/B)$, posição no bloco de cache: 0 a $K-1$

C = 16, **B** = 4, **K** = 4

End cache	Bloco	pos/B	Dado
0	0	0	
1		1	
2		2	
3		3	
4	1	0	
5		1	
6		2	
7		3	
8	2	0	
9		1	
10		2	
11		3	
12	3	0	
13		1	
14		2	
15		3	

mapeamento de dados
MP-cache

MP

End MP	Linha	Dado
0	0	
1		
2		
3		
4		
5	1	
6		
7		
8	2	
9		
10		
11		
12		
13	3	
14		
15		
16	4	
17		
18		
19	5	
20		
21		
22		
23	6	
24		
25		
26	7	
27		
28		
29	8	
30		
31		
32	9	
33		
34		
35		
36	10	
37		
38		
39		
40	11	
41		
42		
43		
44	12	
45		
46		
47		

cache

End cache	Bloco	Dado
0	0	
1		
2		
3		
4		
5	1	
6		
7		
8		
9	2	
10		
11		
12		
13	3	
14		
15		

mapeamento direto MP-cache

- São mapeadas linhas de cache na MP específicas para blocos na cache
- Um mapeamento básico:
 - Um dado ocupa um $End(MP)$, de 0 a $N-1$ da MP =>
 - Ocupa $End(LC)$, de 0 a $M-1$ =>
 - Ocupa $End(pos/LC)$, 0 a $K-1$ =>
 - Mapeamento: as linhas $M=0, B, 2B, \dots$ são mapeadas no bloco $B=0$ da cache; $1, B+1, 2B+1, \dots$ mapeadas no bloco $B=1$; ...até $B-1, 2B-1, \dots$
 - ...

mapeamento direto MP-cache

- Um mapeamento básico:
 - Um dado ocupa um $End(B)$, de 0 a $B-1$ da cache =>
 - Ocupa $End(pos/BC)$, de 0 a $K-1$ =>
 - Ocupa $End(C)$, 0 a $C-1$

mapeamento direto MP-cache

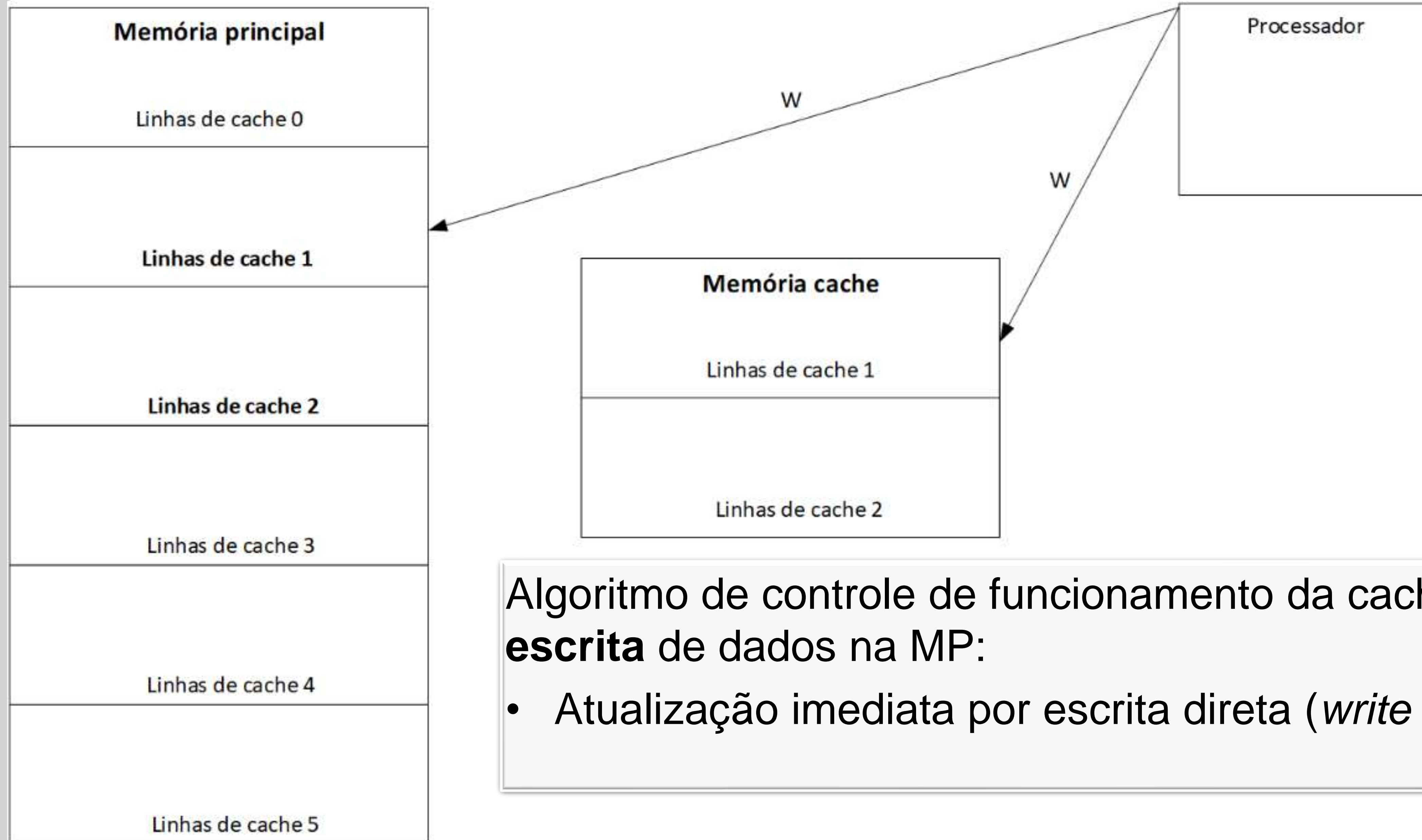
- Matematicamente:
 - $End(LC) = End(MP) \text{ div } K$
 - $End(pos/LC) = End(MP) \text{ Mod } K$
 - $End(B) = LC \text{ Mod } B$
 - $End(pos/B) = End(pos/LC)$
 - $End(C) = End(B).K + End(pos/B)$

mapeamento associativo MP-cache

- Uma vez que tenha ocorrido uma falta, é necessário buscar o dado na memória e **substituir os dados** de algum dos blocos da cache.
- Técnicas de substituição de dados da cache:
 - FIFO – *First-In-First-Out*, em que o bloco cujos dados foram os primeiros a ser inseridos são aqueles a ser substituídos;
 - LRU – *Least Recently Used*, em que o bloco que há mais tempo não é acessado deve ser substituído.

controle da cache na escrita de dados

ibmec.br

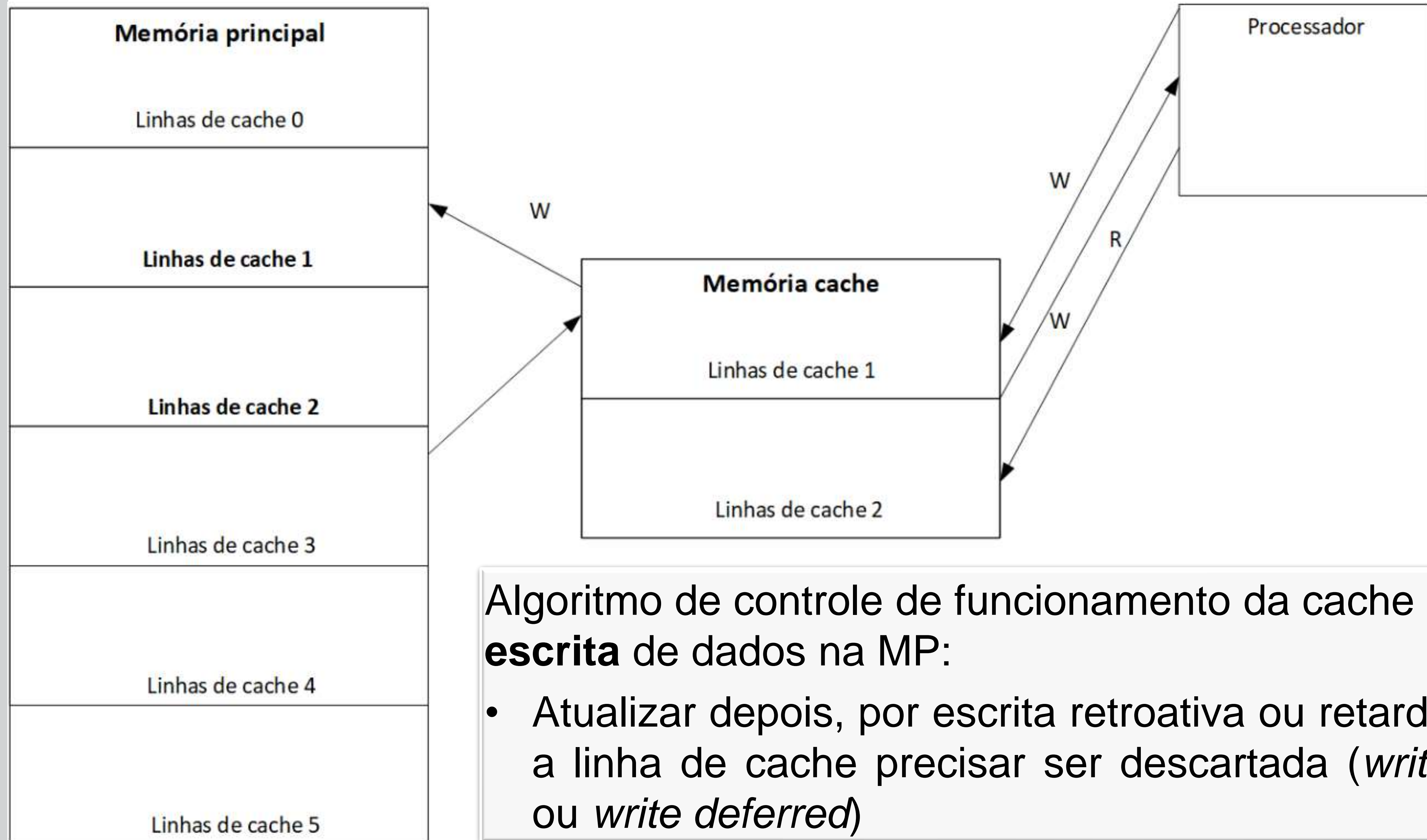


Algoritmo de controle de funcionamento da cache para **escrita** de dados na MP:

- Atualização imediata por escrita direta (*write through*)

controle da cache na escrita de dados

ibmec.br



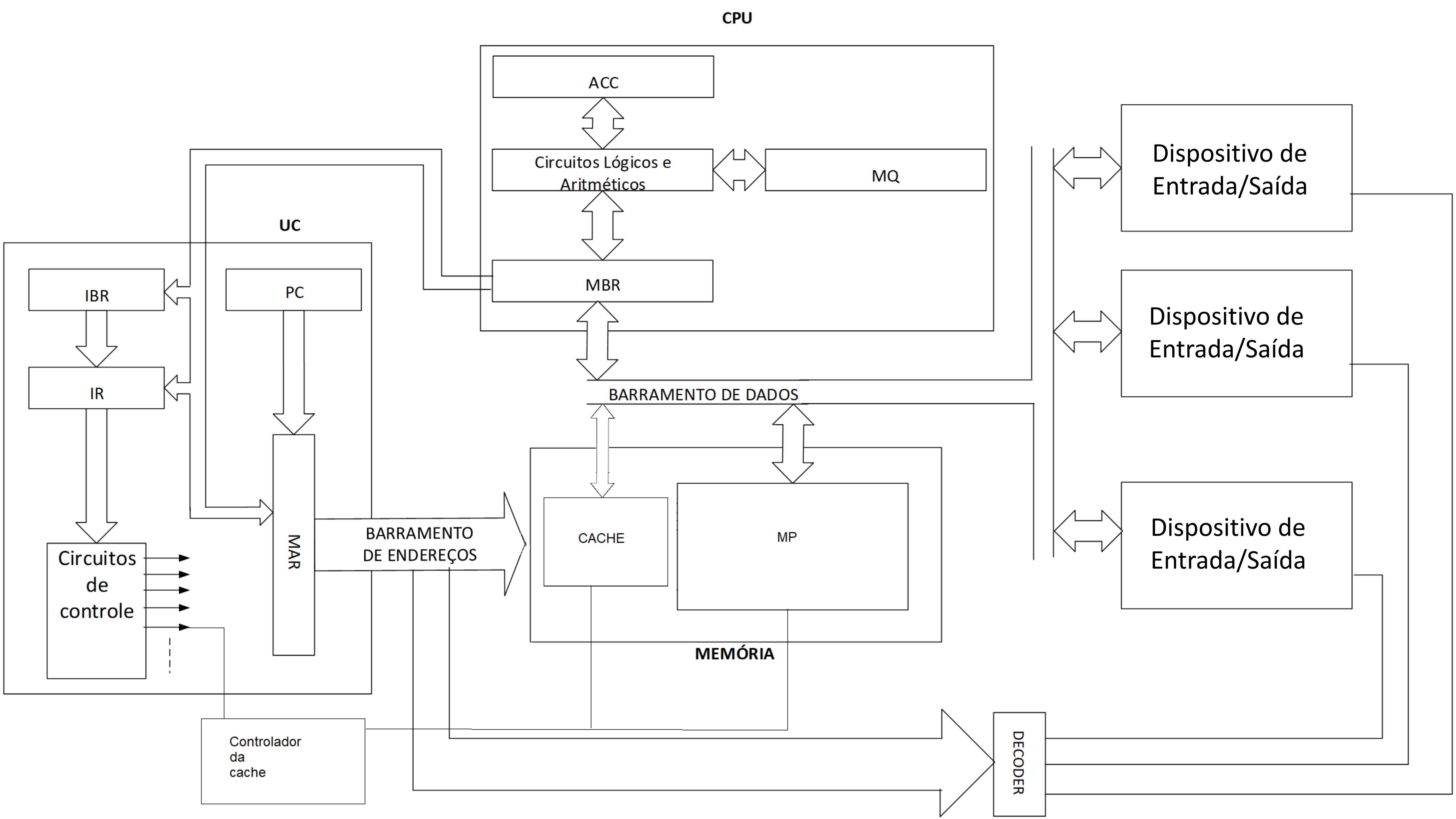
Algoritmo de controle de funcionamento da cache para **escrita** de dados na MP:

- Atualizar depois, por escrita retroativa ou retardada se a linha de cache precisar ser descartada (*write back* ou *write deferred*)

Desempenho do sistema de memória com cache

- CPU busca dados primeiro na memória cache
- k acessos a uma palavra em um intervalo de tempo com a referência 1 vez à MP
- Como avaliar:
 - $c = t_A$ da cache
 - $m = t_A$ da MP
 - taxa de acerto (h), $h = (k-1)/k = 1 - 1/k$
 - tempo médio de acesso (t_M),

$$t_M = \frac{c \cdot (k - 1) + m}{k} = c \cdot h + \frac{1}{k} \cdot m = c \cdot h + (1 - h) \cdot m$$



Elementos do sistema de computação

- Memória
- **Barramento**
- Processadores
- Dispositivos de Entrada/Saída

Barramento

Caminho elétrico que liga os elementos da arquitetura – trilhas e cabos.

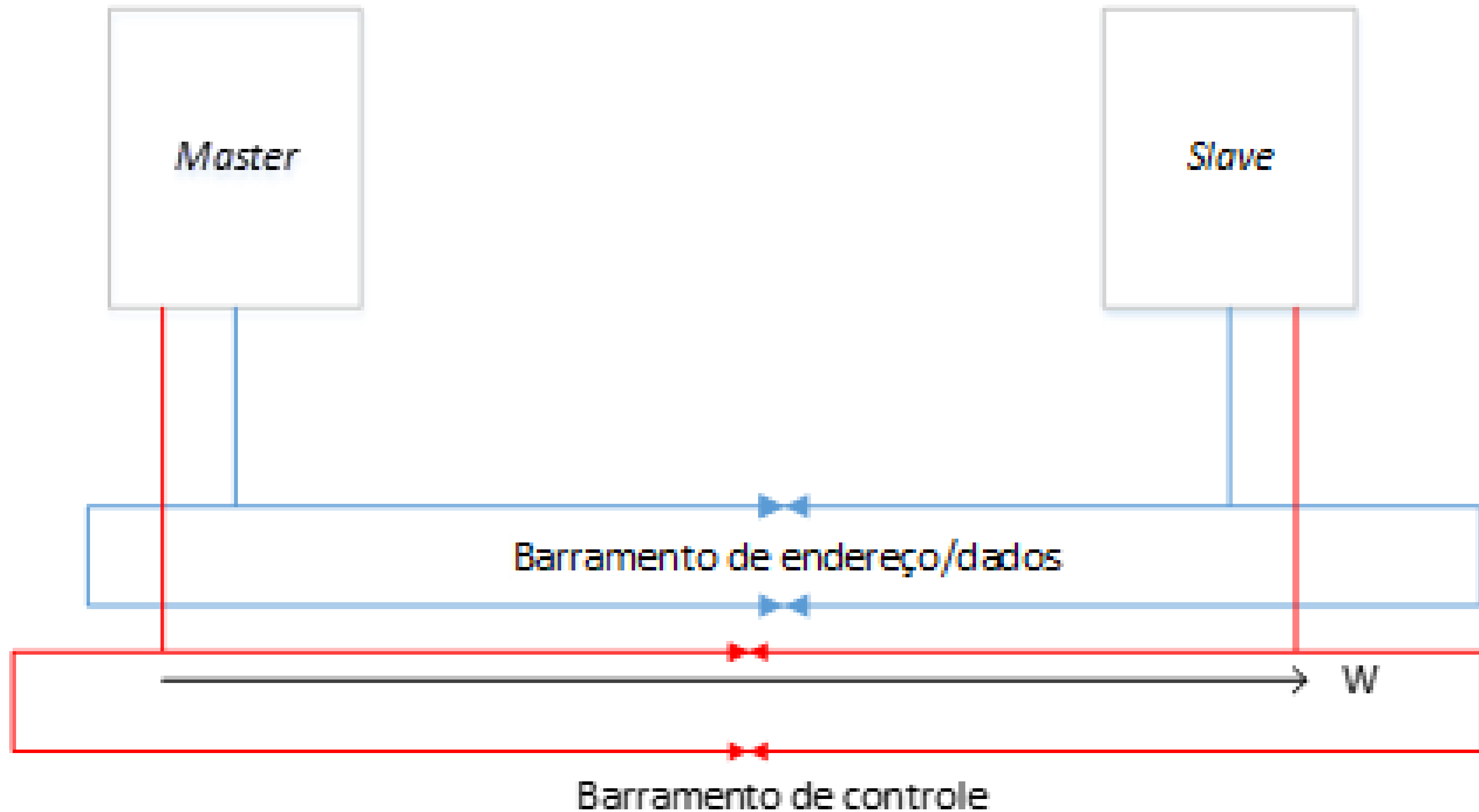
Classificação de barramento

- Quanto à **funcionalidade**: transportar **dados, endereços** ou **sinais de controle**
- Quanto à **localização**: **internos x externos** - proporcionando as ligações entre os componentes da placa-mãe ou dos componentes dela com os dispositivos de I/O

Protocolos de barramento

- Regras que definem a comunicação entre os elementos do sistema computacional
- Controlador de barramento – Mediador da comunicação, responsável pela implementação do protocolo

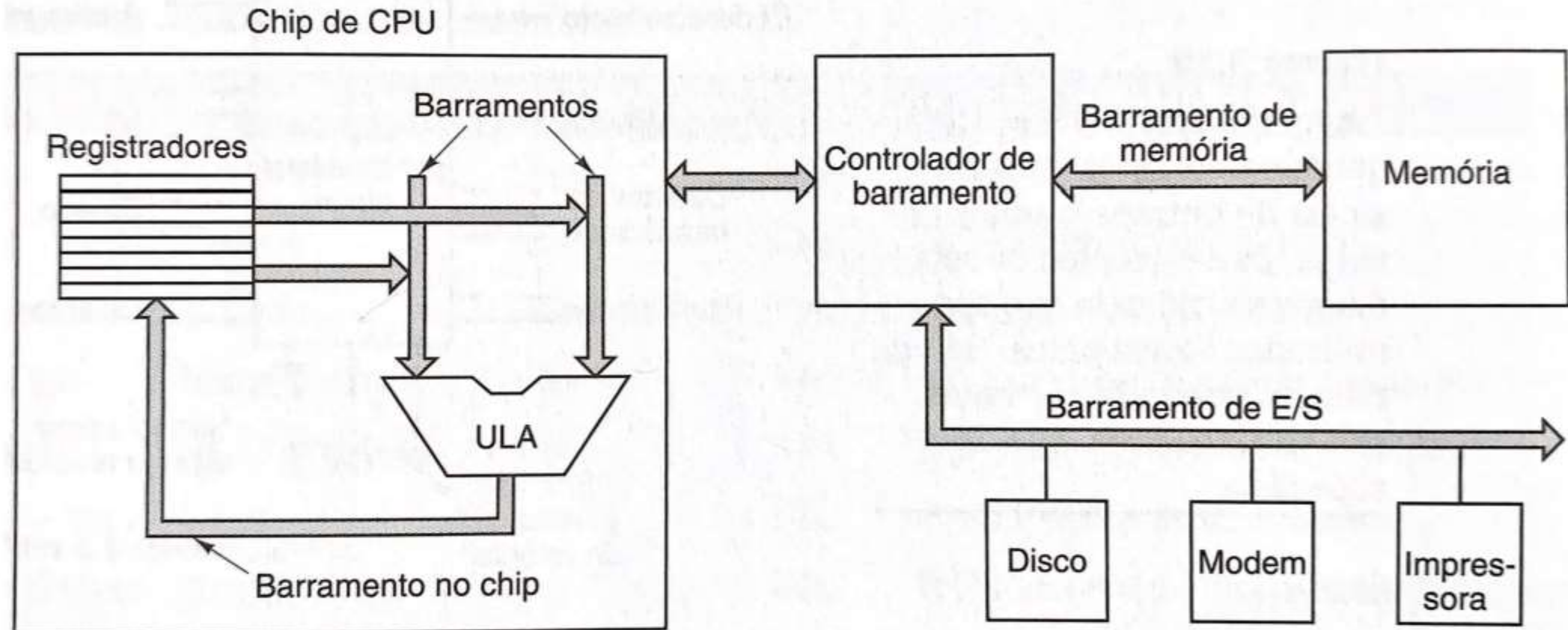
Master x slave



Master x slave

- ***Master*** (mestre) - Elementos **ativos**, que podem **iniciar** a transferência de dados
- ***Slave*** (escravo) – Elementos **passivos**, que esperam as **requisições** para transferência de dados
- Na transferência de dados entre CPU e memória, a CPU é *master* e a memória é *slave*.

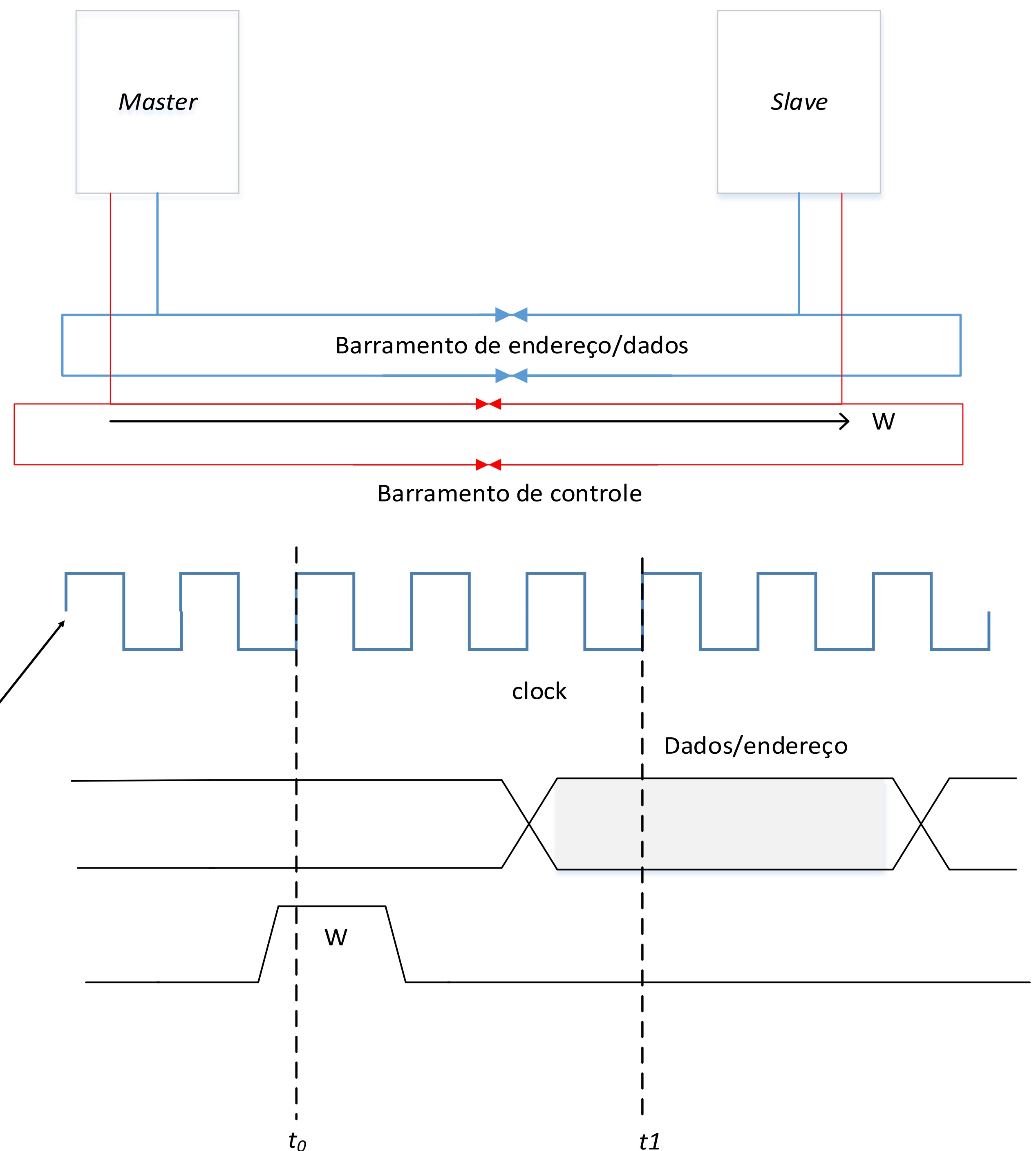
Controlador de barramento



Protocolo de barramento síncrono

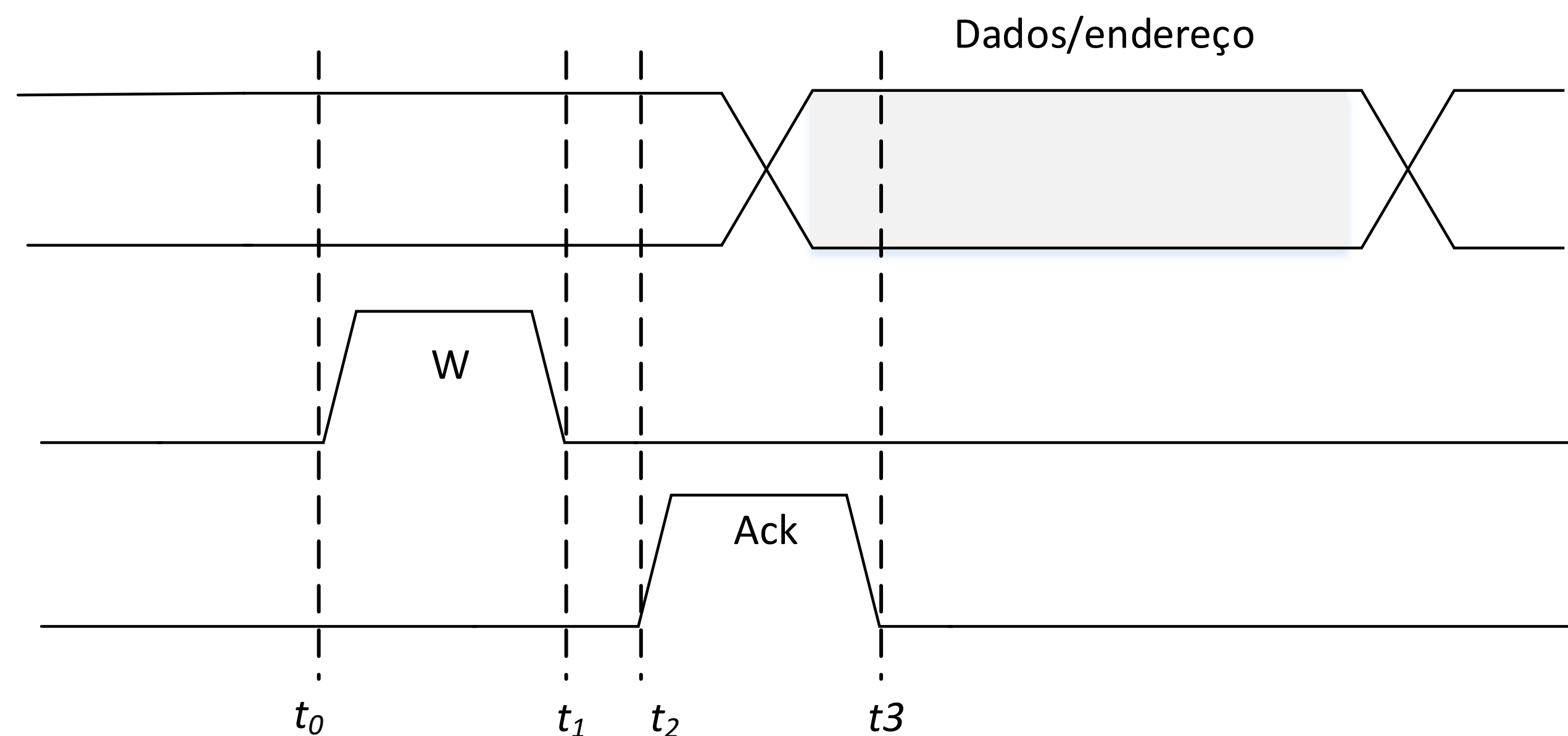
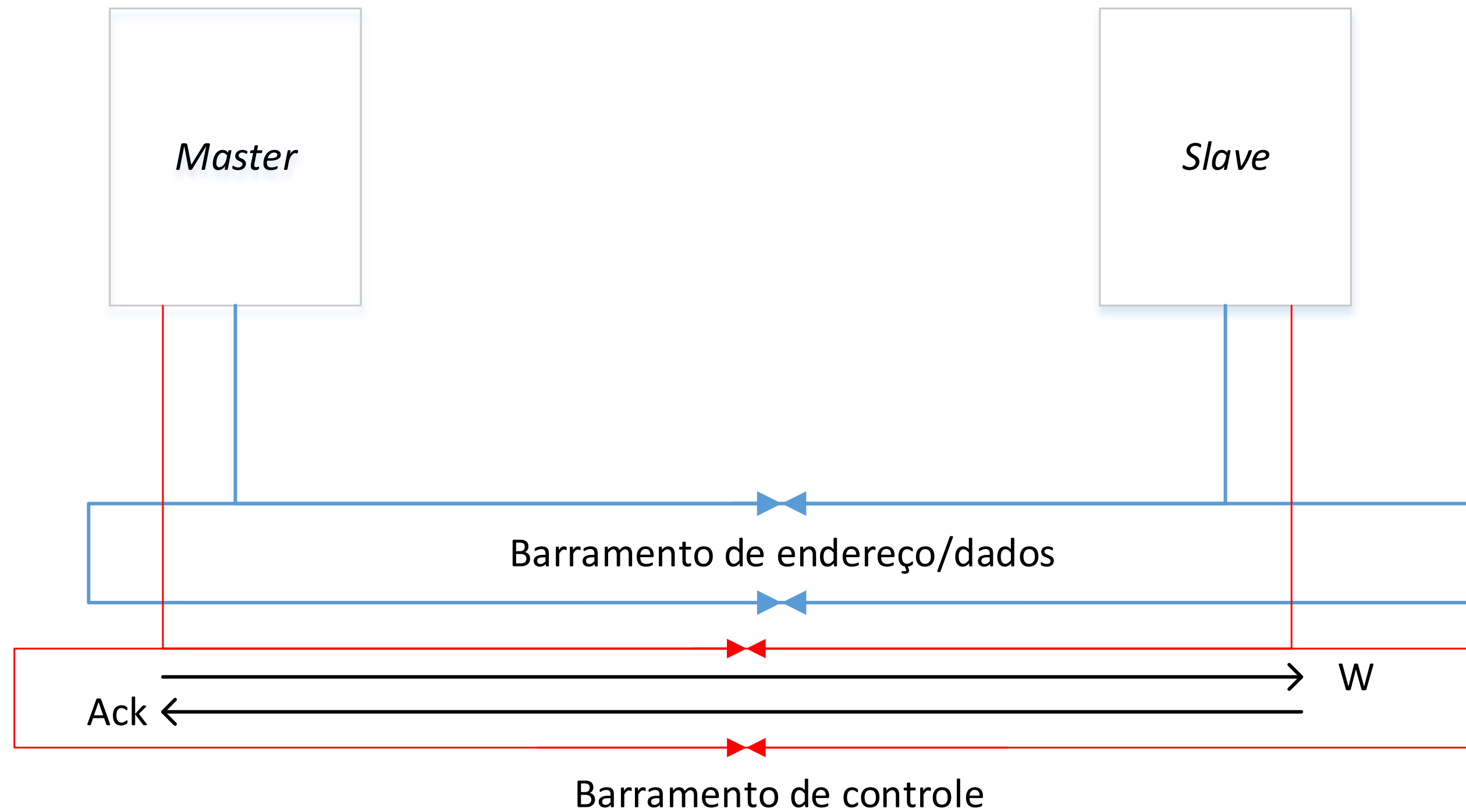
- Possui uma linha de *clock* que estabelece a referência temporal nas transferências de dados no barramento
- A CPU pode precisar ficar em espera até que a leitura/escrita possa ser realizada

Relógio do barramento



Protocolo de barramento assíncrono

- Não possui um relógio mestre
- Não existe temporização – há uma série de eventos sequenciados
- Mais complexa implementação eletrônica

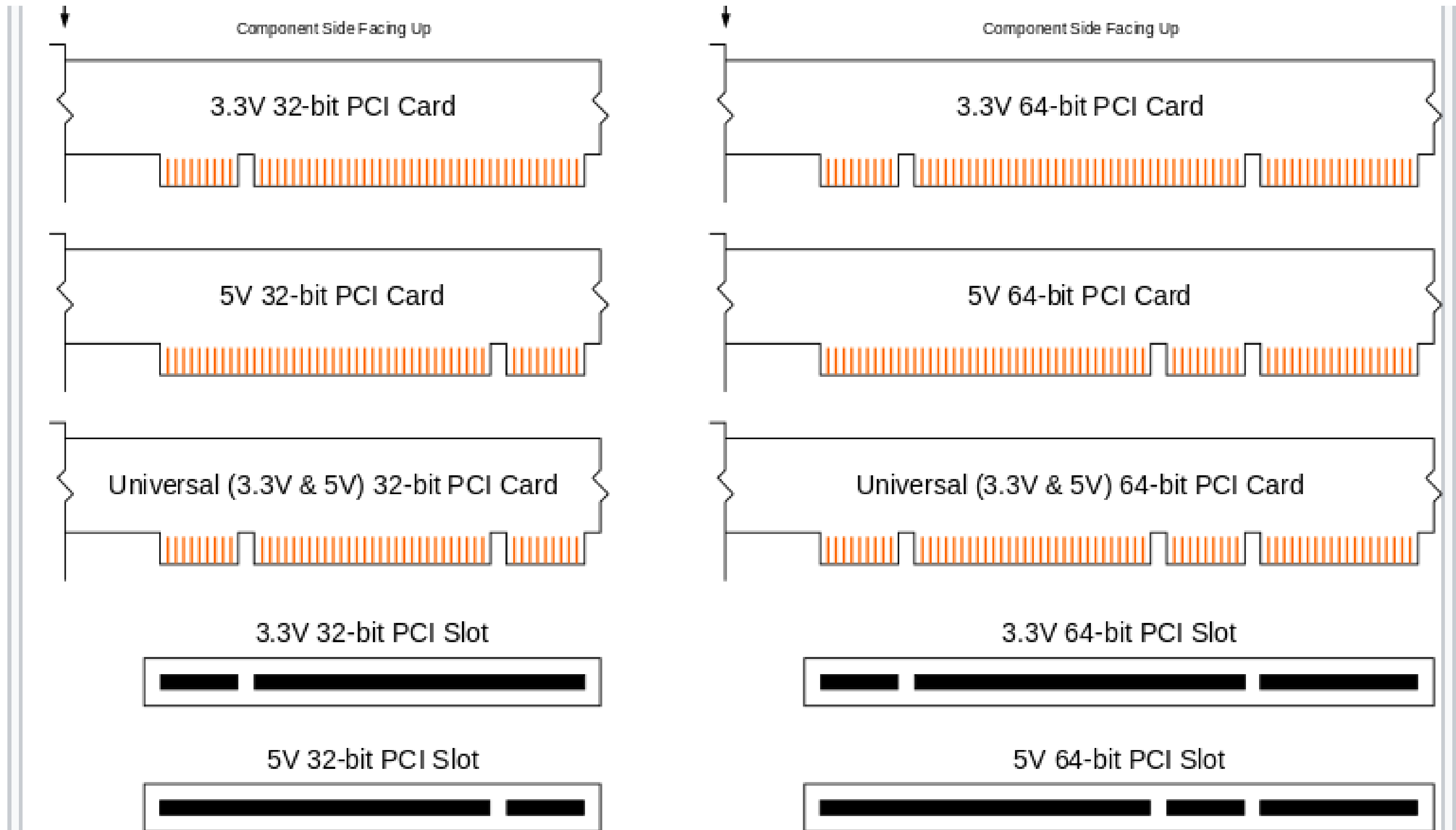


Padrões de barramento

- Barramento ISA (*Industry Standard Architecture*): 8 bits (16 bits), *clock* de 8,33 MHz
- Barramento PCI (*Peripheral Component Interconnect*): 32 bits e *clock* de 33 MHz
- Barramento AGP (*Accelerated Graphics Port*):
 - AGP 1.0: 32 bits, 66 MHz; AGP 2.0 ...
- Barramento PCI-X (*Peripheral Component Interconnect Extended*):
 - PCI-X 1.0: 64 bits, 66 MHz, 100 MHz e 133 MHz
 - PCI-X 2.0: 64 bits, 266 MHz e 533 MHz

PCI-X

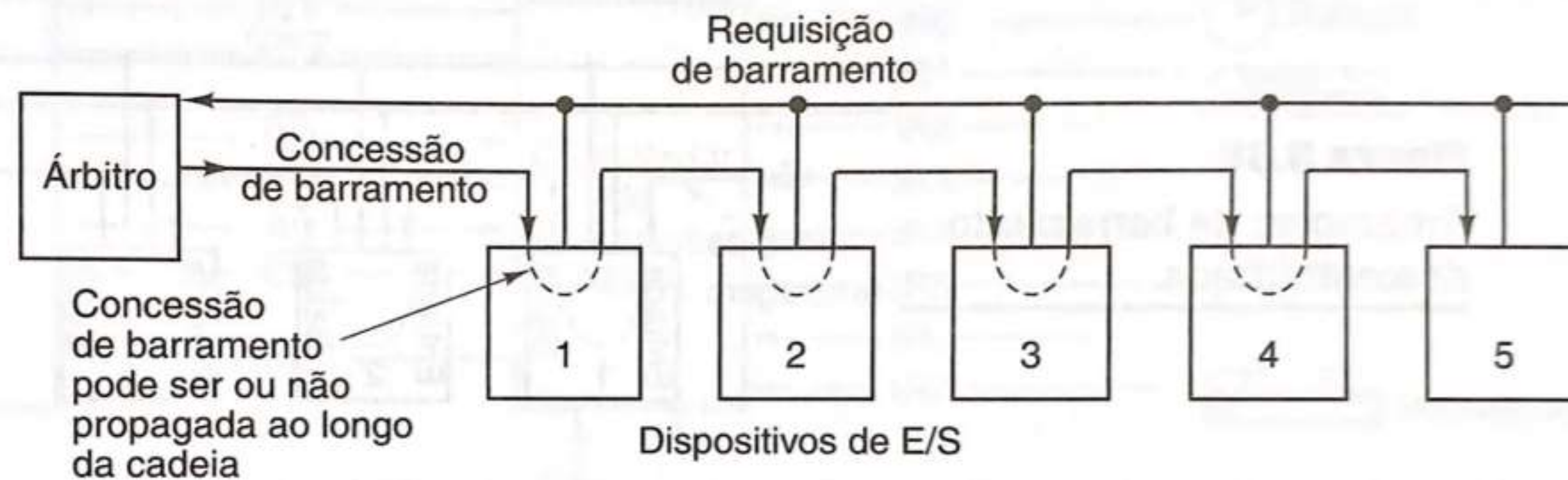
- Cartões
 - 66 MHz
 - 100 MHz
 - 133 MHz
 - 266 MHz
 - 533 MHz
- Slots
 - 66 MHz
 - 133 MHz
 - 266 MHz
 - 533 MHz



3.3 V and 5 V keying of 64-bit PCI cards (both PCI and PCI-X). While most 64-bit PCI-X cards are universal and are backward compatible with common 32-bit 5 V PCI slots, PCI-X slots are 3.3 V and will not accept 5 V-only PCI cards.

Arbitragem de barramento

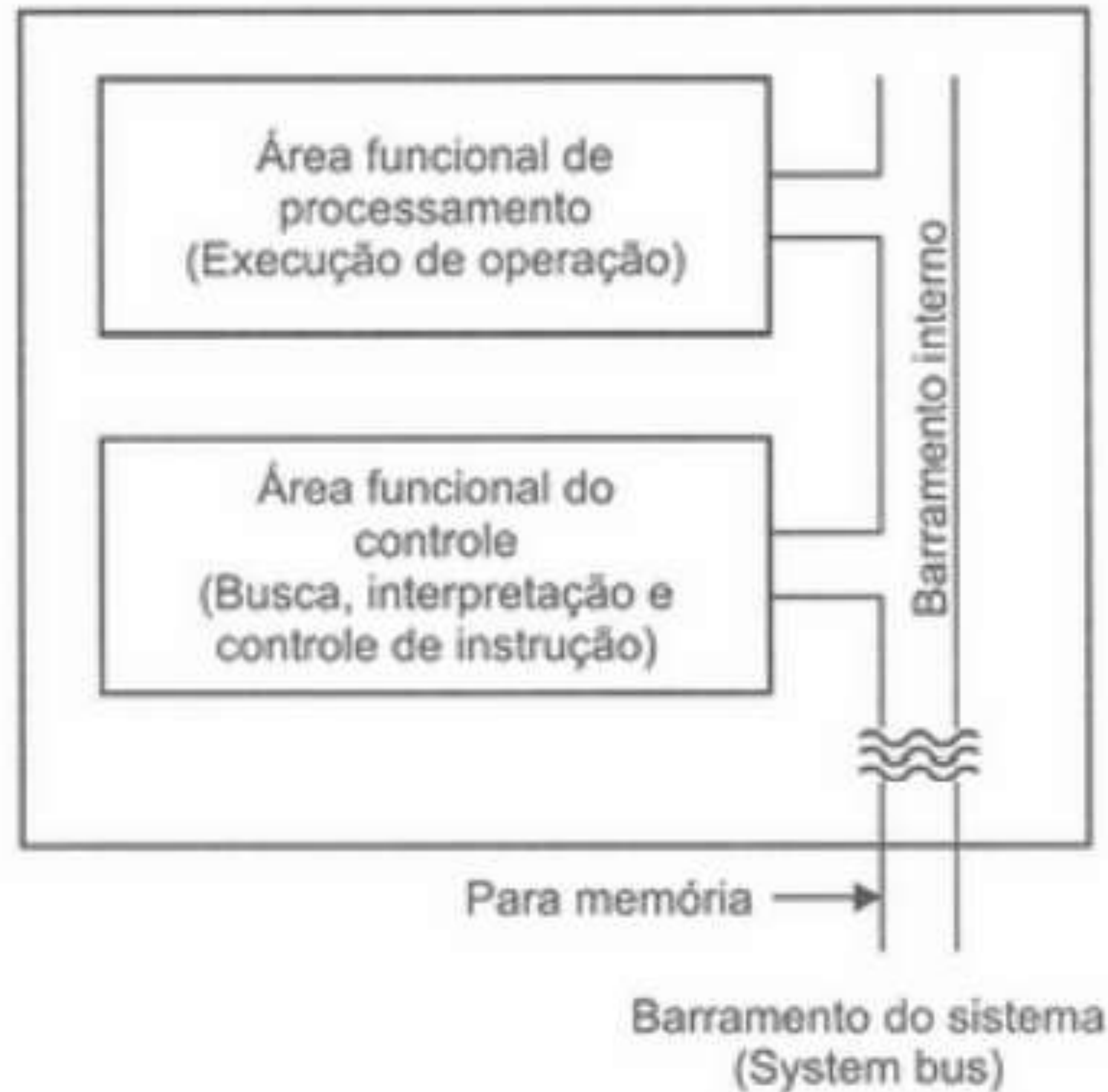
- Nos sistemas computacionais pode haver mais de um *master* na transferência de dados
- Como resolver o conflito se 2 ou mais masters solicitarem transferência de dados ao mesmo tempo

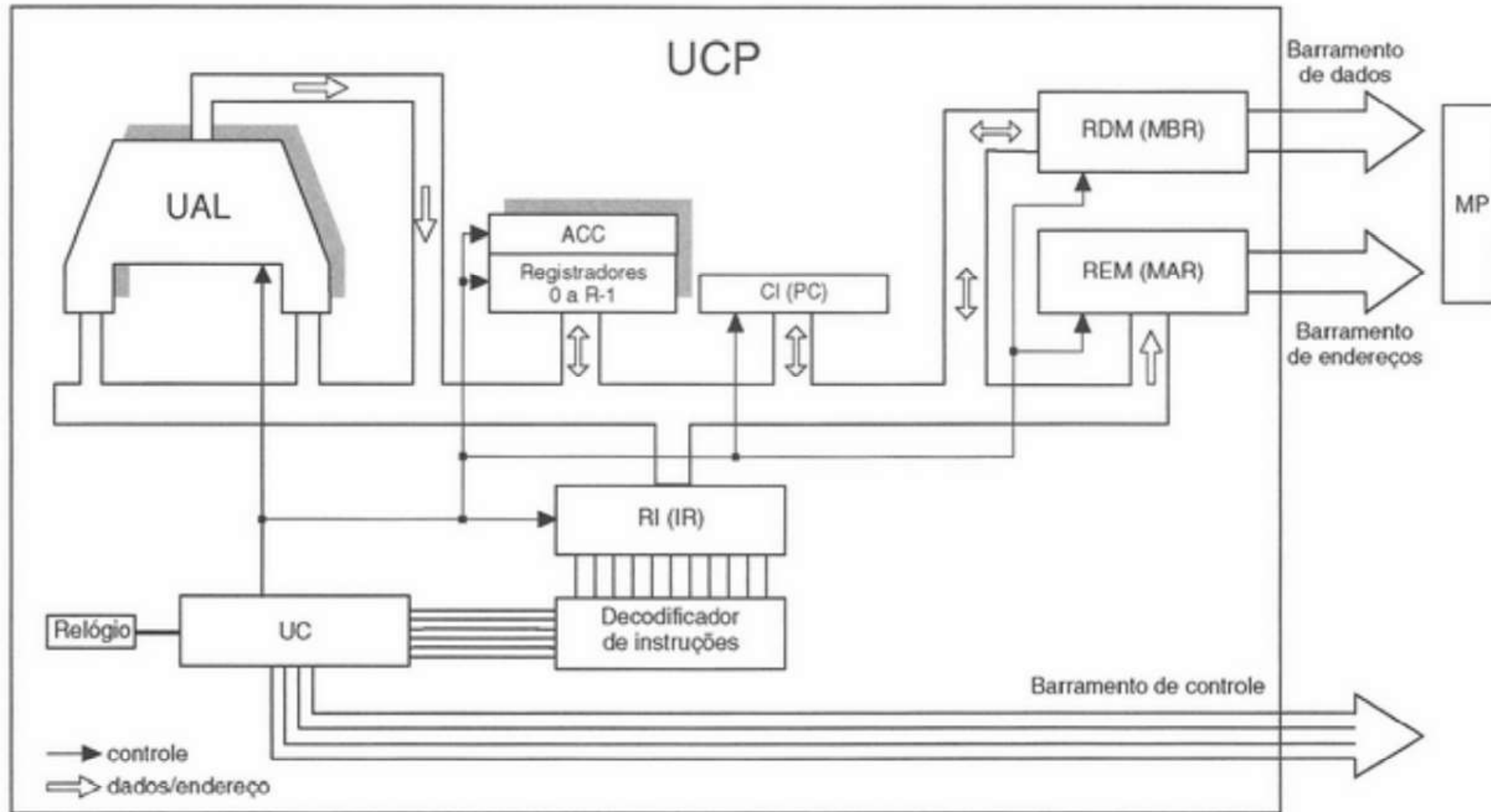


Elementos do sistema de computação

- Memória
- Barramento
- **Processadores**
- Dispositivos de Entrada/Saída

Áreas funcionais





Set de instruções

- Instruções que podem ser **interpretadas** pelo processador, **decodificadas** pela UC
- Cada instrução corresponde um comportamento específico
- Armazenadas em posições lineares de endereço
- Normalmente possuem **visibilidade sobre alguns registradores**
- A instrução pode ser dividida em dois campos, ***opcode***, ou código da instrução; e ***operando***.

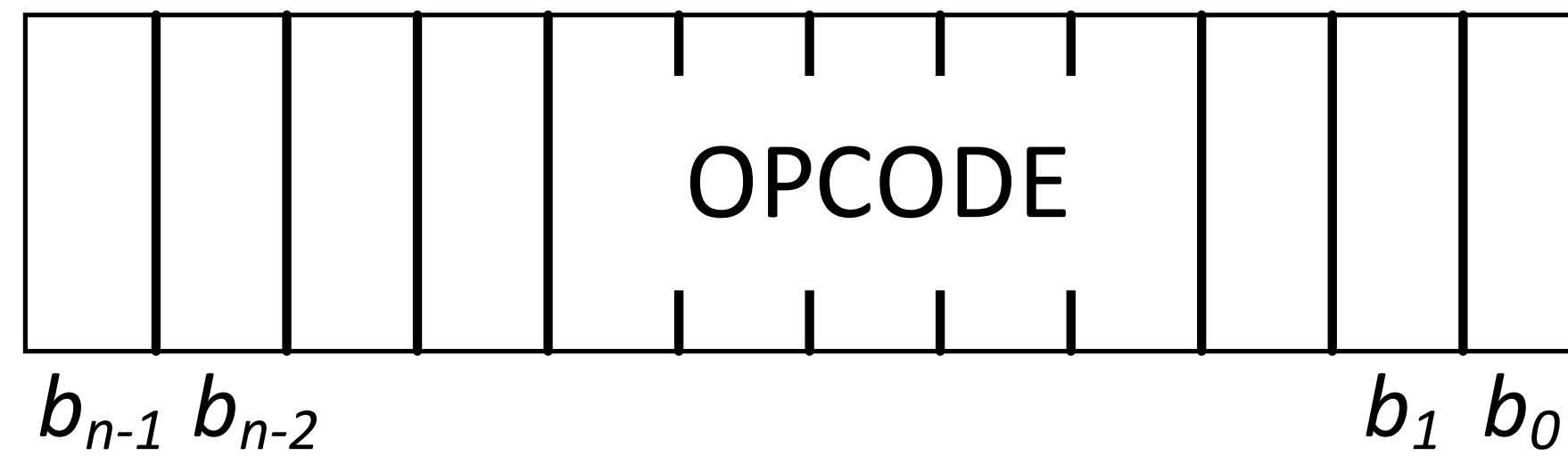
Tipos de arquiteturas

- *RISC – Reduced Instruction Set Computer*: Conjunto reduzido de instruções
- *CISC – Complex Instruction Set Computer*: Conjunto complexo de instruções

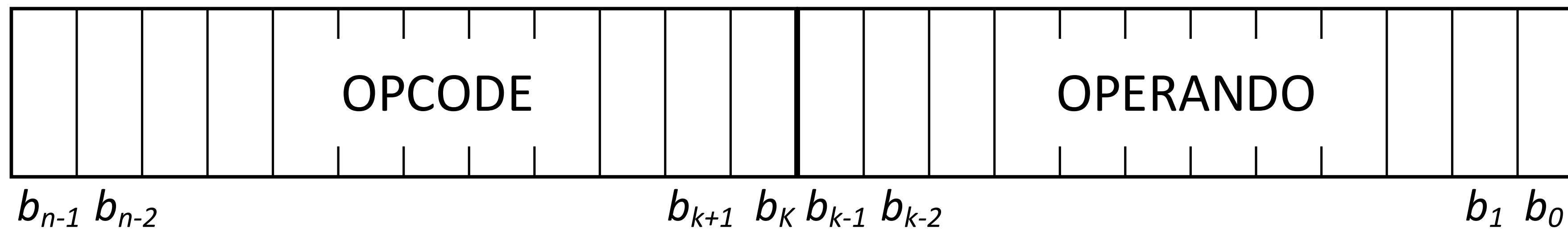
Princípios RISC

- Instruções executadas por hardware
- Iniciar o maior número possível de instruções por segundo, usando, por exemplo, paralelismo na execução
- Instruções fáceis de decodificar
- Usar somente poucas instruções para referenciar a memória (LOAD e STORE, por exemplo)
- Utilizar muitos registradores

Elementos das instruções = opcode + operando



(a)



(b)

Formatos típicos

Instrução sem operando: possui somente *opcode*

Exemplo:

IJMP

operação

- $PC \leftarrow Z$, onde Z é um registrador e PC é o Contador de Programa;
- Executa desvio incondicional para o endereço armazenado em Z

Formatos típicos

Instrução sem operando: possui *opcode* e operando dividido em um ou mais campos

Exemplo:

ADD R_1 , R_2

operação

- $R_1 \leftarrow R_1 + R_2$, onde R_1 e R_2 são registradores de uso geral
- Executa adição binária dos valores armazenados R_1 e R_2 armazenando o resultado em R_1

Modos de endereçamento

como a instrução indica a localização do dado na arquitetura

- imediato
- direto
- registrador
- indireto do registrador
- indexado
- de base indexado
- de pilha
- para instruções de desvio

Endereçamento **imediato**

- operando contém o próprio dado
- dado é buscado juntamente na busca da instrução
- a execução da instrução tipicamente mais rápida

Exemplo

CALL k

operação

- $PC \leftarrow k$, carrega no Contador de Programa o valor do operando

Endereçamento **direto**

- operando contém o endereço do dado em memória
- execução da instrução requer, em consequência, uma busca da instrução, seguida da busca ao dado armazenado no endereço

Exemplo

LDS Rd, k

operação

- $Rd \leftarrow [k]$, carrega no registrador de uso geral Rd o valor armazenado no endereço k da memória

Endereçamento **indireto**

- operando carrega um dado ou lê um dado em memória
- endereço que aponta ao dado está contido em um registrador.
- endereço nessa forma é chamado de **ponteiro**.

Exemplo

LD Rd, X+

operação

- $Rd \leftarrow [X+]$, carrega no registrador de uso geral Rd o valor armazenado no endereço da memória cujo valor está armazenado no registrador X+



Tipos de instrução

- instruções para movimento de dados
- operações diádicas
- operações monádicas
- comparações e desvios condicionais
- instruções de chamada de procedimento
- instruções de controle de laço
- instruções de entrada/saída



Tipos de instrução

- instruções para movimento de dados transferem uma cópia do dado armazenado em um elemento para outro elemento
 - operações diádicas
 - operações monádicas
 - comparações e desvios condicionais
 - instruções de chamada de procedimento
 - instruções de controle de laço
 - instruções de entrada/saída



Tipos de instrução

- instruções para movimento de dados

- operações diádicas

combinam dois campos do operando para produzir um resultado

- operações monádicas

- comparações e desvios condicionais

- instruções de chamada de procedimento

- instruções de controle de laço

- instruções de entrada/saída



Tipos de instrução

- instruções para movimento de dados
- operações diádicas
- operações monádicas

possuem somente um campo de operando e produzem o resultado

- comparações e desvios condicionais
- instruções de chamada de procedimento
- instruções de controle de laço
- instruções de entrada/saída



Tipos de instrução

- instruções para movimento de dados
- operações diádicas
- operações monádicas
- comparações e desvios condicionais
permitem estabelecer fluxos alternativos de execução do programa
- instruções de chamada de procedimento
- instruções de controle de laço
- instruções de entrada/saída



IBMEC.BR

 /IBMEC

 IBMEC

 @IBMEC_OFICIAL

 @IBMEC

 **ibmec**