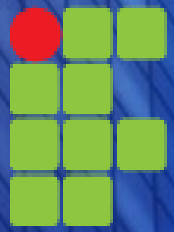


INSTITUTO FEDERAL
SANTA CATARINA

ACESSO A MEMÓRIAS

MICROPROCESSADORES

.....MEMÓRIAS

A memória é o componente de um sistema de computação cuja função é armazenar as Informações que são, foram ou serão manipuladas pelo sistema.

Para tanto temos ações:

- armazenamento(escrita / gravação / *write*);
- recuperação(leitura / gravação / *read*);

.....MEMÓRIAS

Na prática, a memória de um computador possui muitos parâmetros, destaca-se:

- » tempo de acesso (velocidade);
- » capacidade de armazenamento(tamanho);
- » Tecnologia (DRAM / SRAM);
- » Custo (relativo aos itens anteriores);

que a memória é considerada um subsistema estruturado e hierarquizado.

.....MEMÓRIAS

O elemento a ser manipulado é o bit, onde ***n bits*** compreendem a unidade de informação a ser armazenada [alocação de memória].

Geralmente, utiliza-se *n=8b* ←→ 1B [byte].

n deve ser múltiplo de potência de base 2.

MEMÓRIAS

A gravação é destrutiva (sobrescrever), ou seja, os dados que estavam gravados anteriormente são substituídos pelos que estão sendo gravados naquele momento.

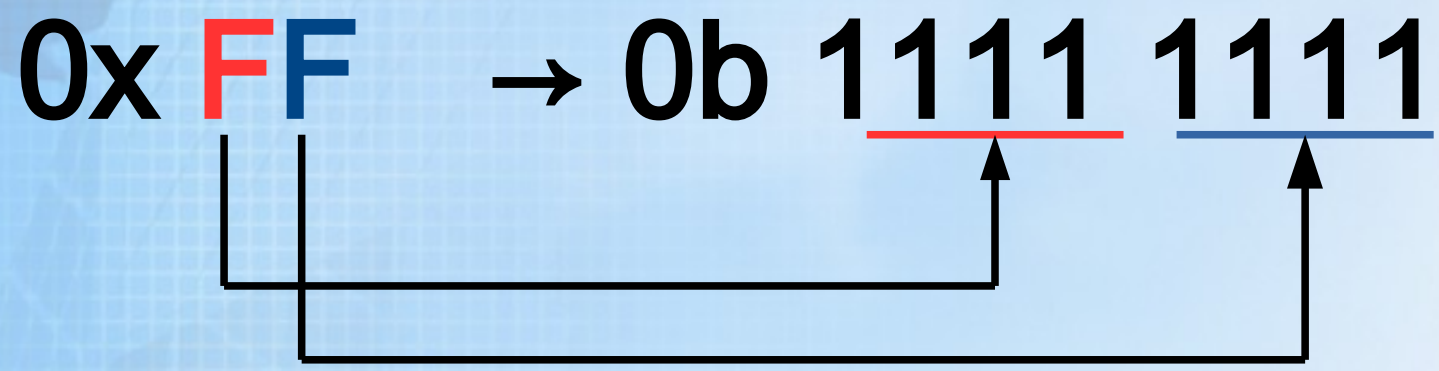
Por outro lado, a recuperação apenas copia o valor armazenado para outro local. **O valor original continua sem alteração**.

MEMÓRIAS

Para que a informação possa ser armazenada em uma memória (gravação/escrita/*write*) é necessário que seja definido um local disponível identificado de alguma forma precisa e única (um código, por exemplo). O código associado ao local é o endereço ("*address*") e irá permitir que a informação possa ser localizada.

0xFF → 0b 1111 1111

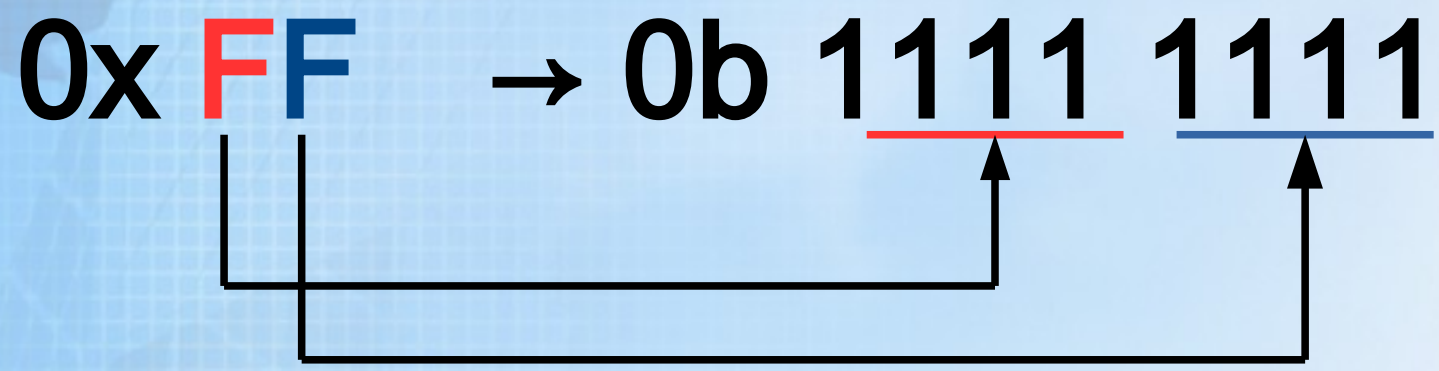
MEMÓRIAS



Quantos fios há nesse barramento?

Quantos são os endereços possíveis?

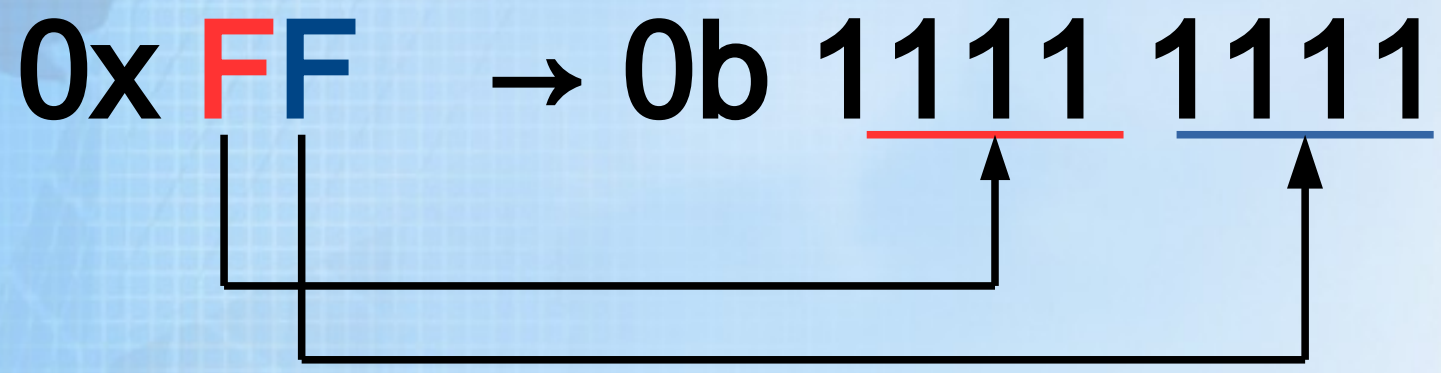
Qual a capacidade dessa memória?



Quantos fios há nesse barramento?

Tanto quantos forem os números de bits a serem transferidos... (8 bits)

MEMÓRIAS



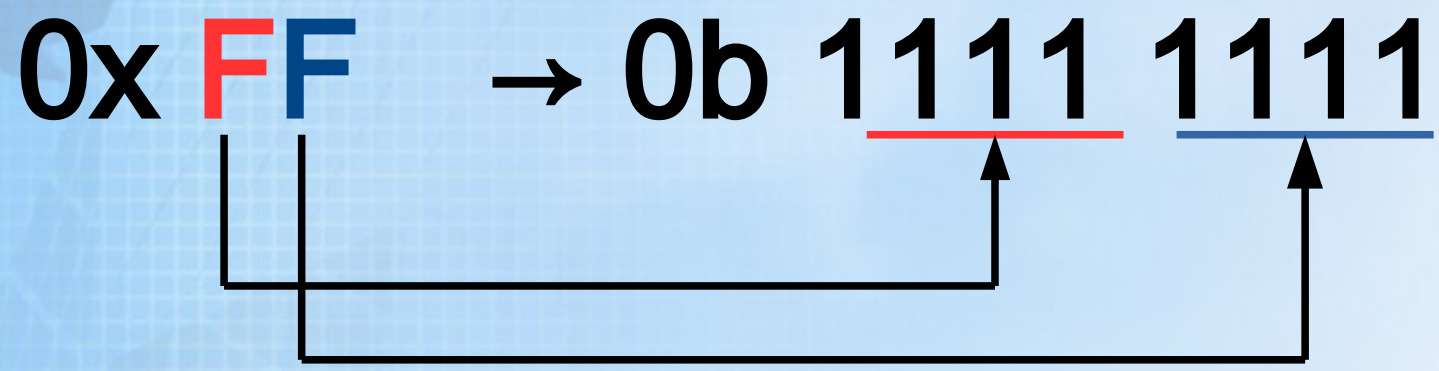
Quantos são os endereços possíveis?

2^8 = 256 bits (possibilidades)

Primeiro:	0000 0000	0x00
	... ...	...
Último:	1111 1111	0xFF

00, 01, 02...0F, 10, 11, 12...1F, 20, 21, 22...2F, 30, 31...3F, 40, 41, 42...4F, 50, 51...5F, 60, 61...6F, 70, 71, 72...7F, 80, 81...8F, 90.....FF

MEMÓRIAS



Qual a capacidade dessa memória?

$$C = E \times A$$

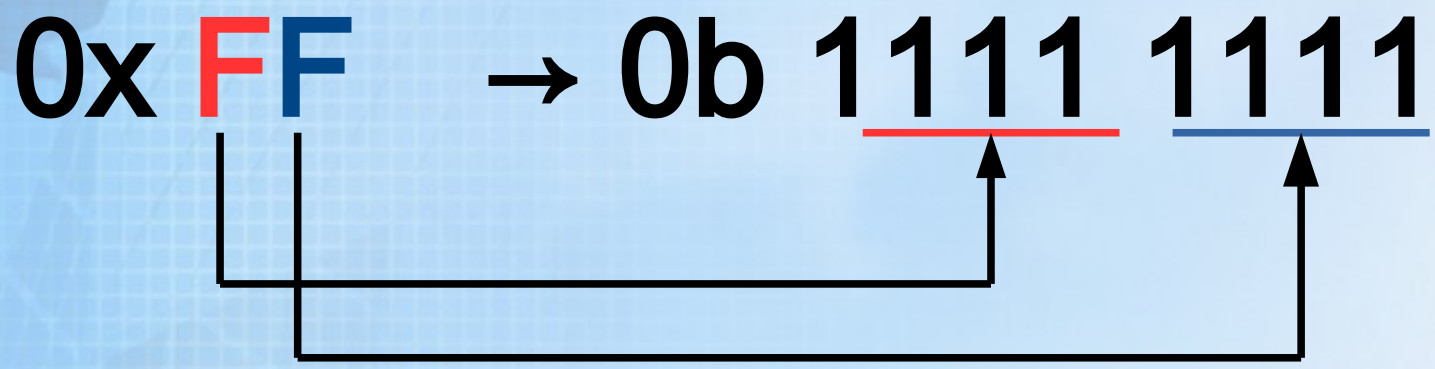
Endereços

Alocação

MEMÓRIAS

Dados

Endereço	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
00	0	1	1	1	0	1	0	1
.								
.	1	1	1	1	0	0	0	0
.								
0F	0	0	0	1	1	0	1	1
10	1	0	0	0	1	1	1	0



Qual a capacidade dessa memória?

$$C = 256 \times 8bits$$

$$C = 2048\ bits$$

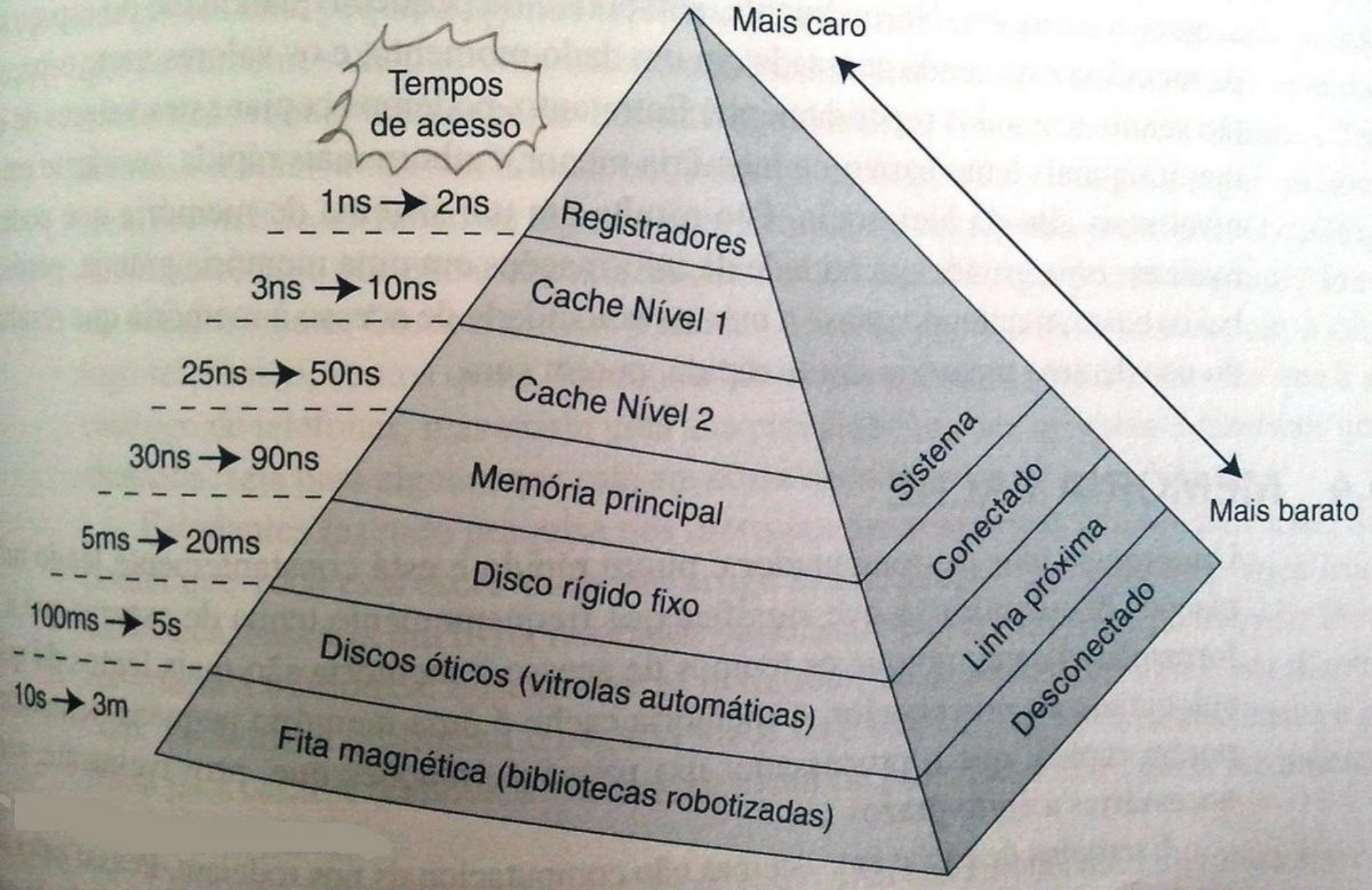
MEMÓRIAS

- Acesso Sequencial
- Acesso Direto
- Acesso Randômico
- Acesso Associativo

MÉTODOS DE ACESSO

Vamos verificar cada um dos tipos...

MEMÓRIAS


Mais caro
Tempos de acesso
1ns → 2ns
3ns → 10ns
25ns → 50ns
30ns → 90ns
5ms → 20ms
100ms → 5s
10s → 3m
Registradores
Cache Nível 1
Cache Nível 2
Memória principal
Disco rígido fixo
Discos óticos (vitrolas automáticas)
Fita magnética (bibliotecas robotizadas)
Sistema
Conectado
Linha próxima
Desconectado
Mais barato

ACESSO SEQUENCIAL

Os dados são organizados na memória em unidades chamadas de registros.

O acesso é feito segundo uma sequência específica. O tempo de acesso depende da posição relativa do registro, variando significativamente.

Exemplo: unidade de fitas magnética e a concepção de memória.

ACESSO DIRETO

Por meio de uma pesquisa sequencial em uma vizinhança do registro é obtido o seu endereço físico, sendo então é possível a leitura ou gravação. O tempo de acesso também é variável.

Exemplo: Disco magnético (HD).

ACESSO RANDÔMICO

Cada posição de memória possui mecanismo de endereçamento fisicamente conectado a ela. O tempo de acesso é o mesmo para todos os endereços.

Exemplo: gravação de diversos arquivos na memória RAM e acesso em momentos distintos.

ACESSO ASSOCIATIVO

Um dado é buscado na memória com base em uma parte de seu conteúdo, e não de acordo com seu endereço.

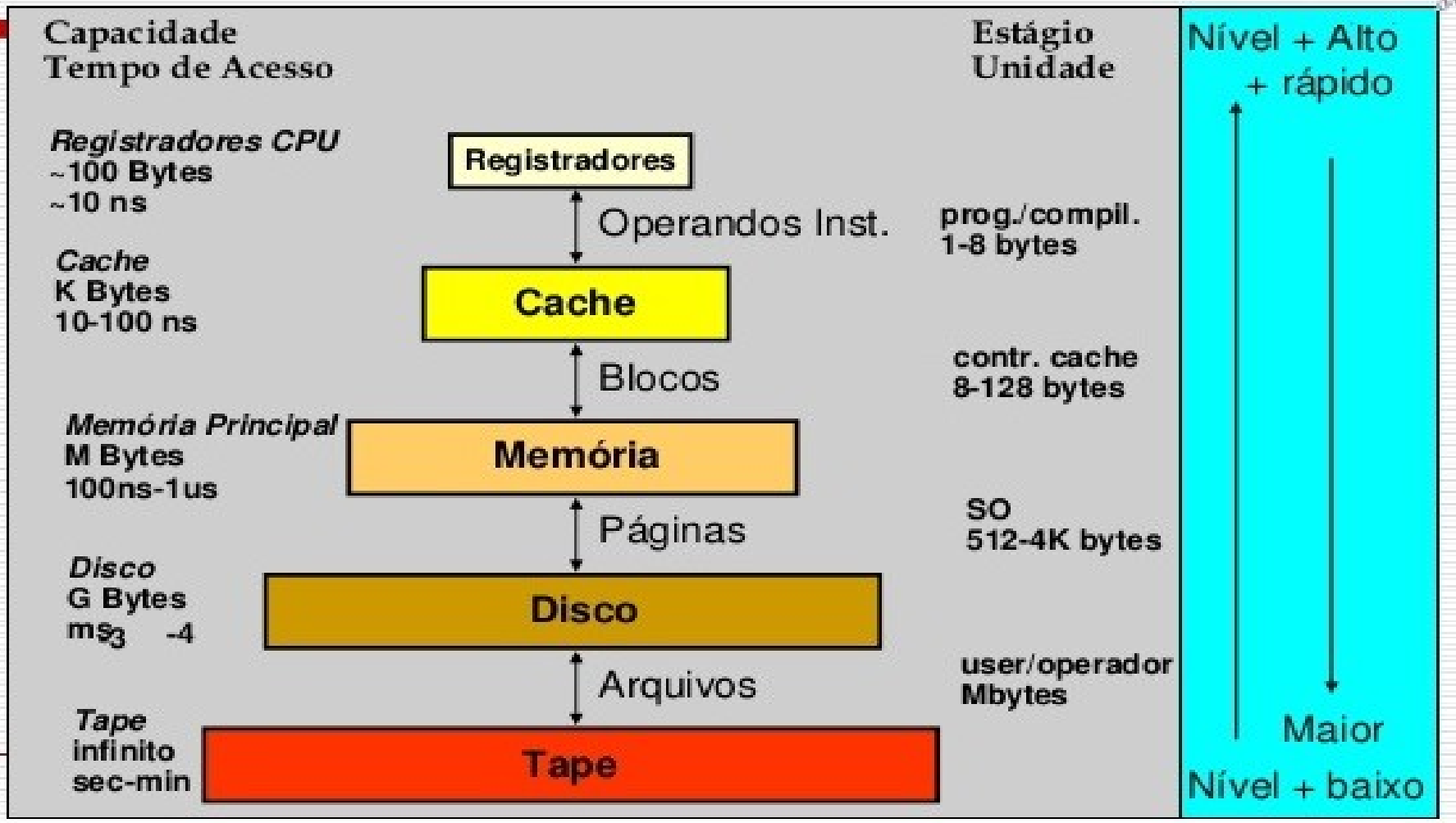
Pode-se ainda guardar a informação do "atalho" daquele conteúdo, buscando-se apenas seu apontamento (histórico).

Exemplo: Memória CACHE.

FORMAS DE ACESSO

Hierarquia de Memória atualmente

Fonte: Slides Patterson



ACESSO À CACHE

Existem vários **registradores** na UCP que favorecem os acessos da memória (tempo), destaca-se:

MBR (Memory Buffer Register)
Armazena temporariamente a informação que está sendo transferida da MP para a UCP (leitura) ou da UCP para MP (escrita). Possui a mesma quantidade de bits do barramento de dados.

ACESSO À CACHE

MAR (Memory Address Register)

Armazena temporariamente o endereço de acesso a uma posição de memória, ao se iniciar a operação de leitura ou escrita. Em seguida, o endereço é encaminhado à área de controle da MP para decodificação e localização da célula desejada. Possui a mesma quantidade de bits do barramento de endereços.

ACESSO À CACHE

Os conceitos de **"Localidade"** justificam a existência da CACHE por prever vários acessos à mesma memória num breve espaço de tempo:

Localidade Temporal: Se um programa acessa um dado (endereço), existe uma boa probabilidade que ele venha a acessá-la novamente, em breve.

ACESSO À CACHE

Localidade Espacial: Se um programa acessa um dado (endereço), existe uma boa probabilidade que ele venha a acessar os dados armazenados em sua proximidade, em breve.

Baseado nestes conceitos, a memória CACHE trabalha seguindo o seguinte algoritmo:

ACESSO À CACHE

1. Sempre que a UCP vai buscar uma nova instrução, ela acessa a memória CACHE.

2. Se a instrução ou dado estiver na CACHE (ACERTO ou HIT), ela é transferida em alta velocidade para a UCP.

ACESSO À CACHE

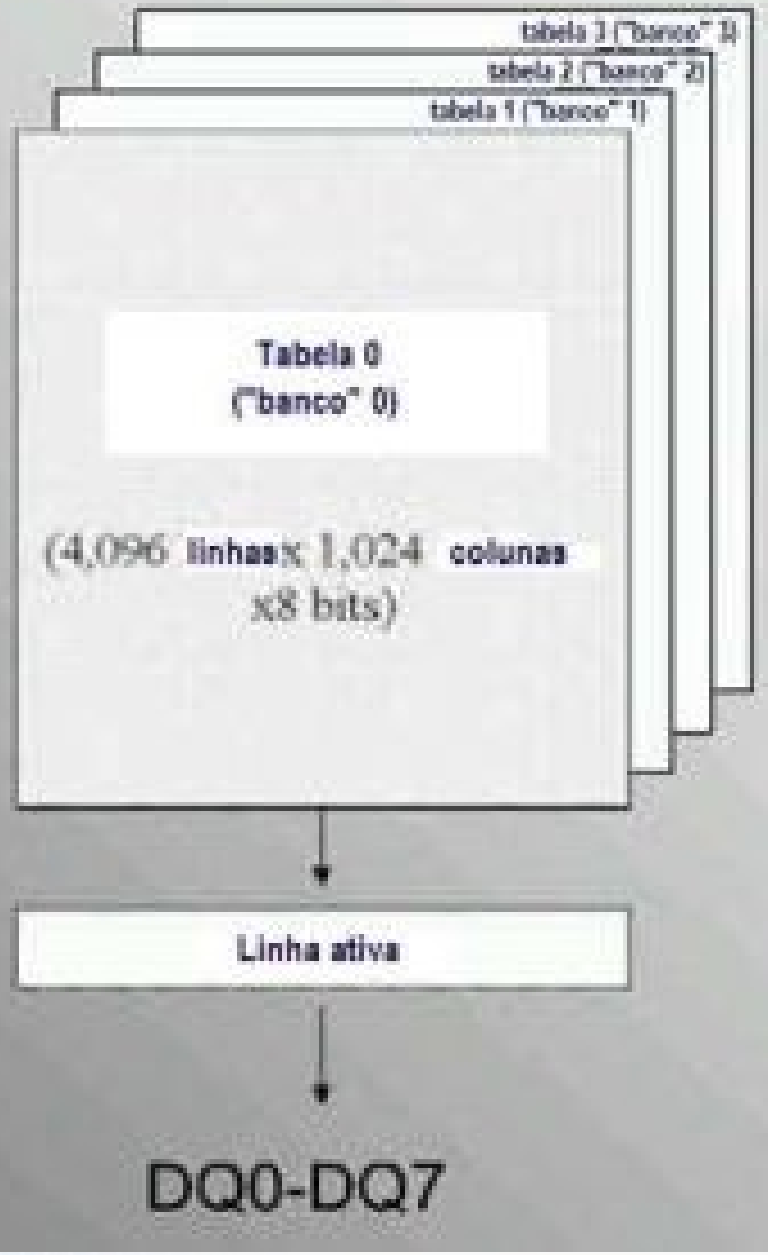
3. Se a instrução ou dado não estiver na CACHE (FALTA ou MISS), então o sistema está programado para interromper a execução do programa e transferir a instrução desejada da MP para a UCP. A UCP entra em “Estado de Espera” (WAIT) enquanto ocorre a demorada transferência do dado vindo da MP.

ACESSO À CACHE

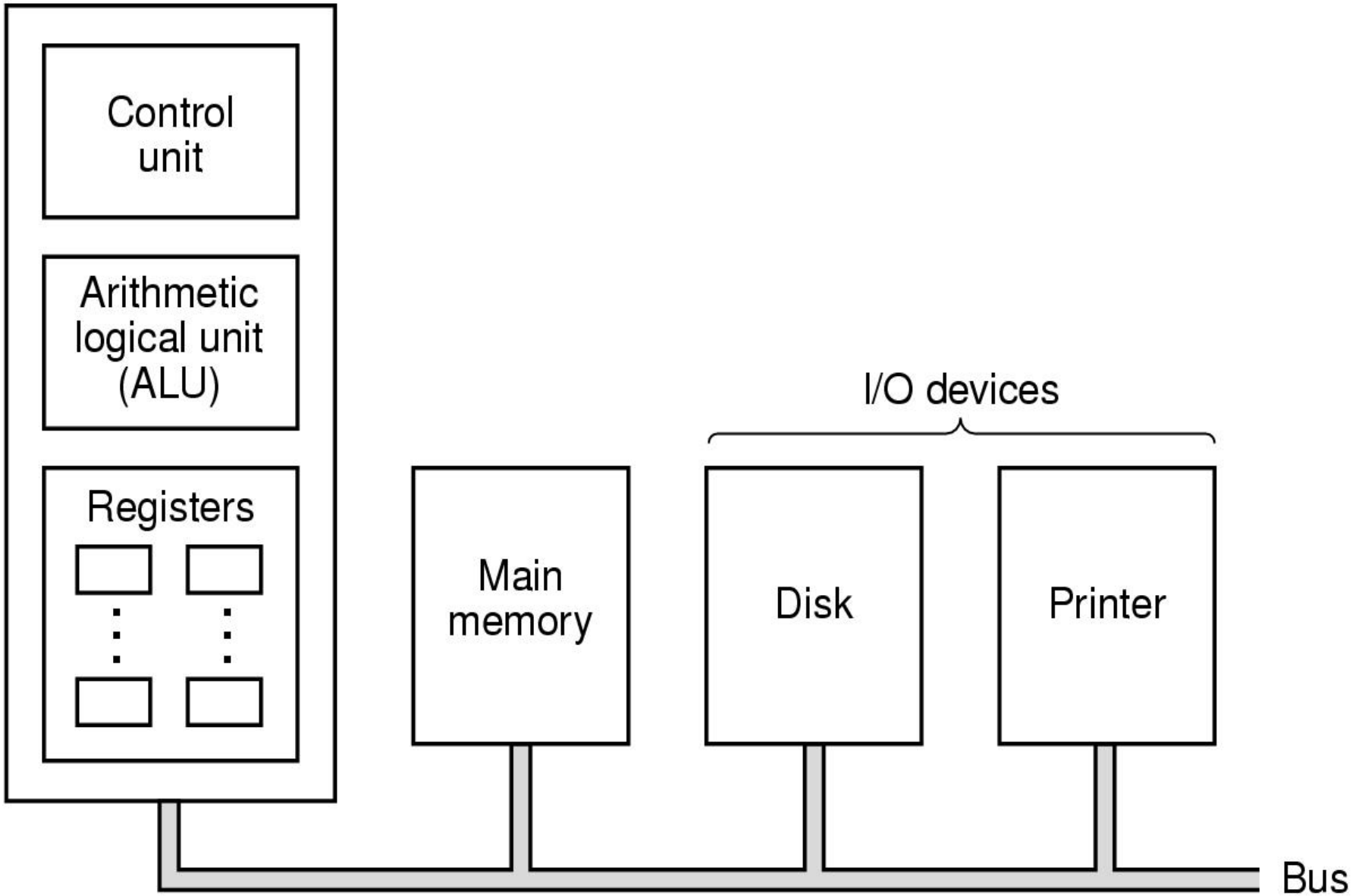
Simultaneamente é transferida uma cópia da instrução desejada mais o conteúdo de alguns endereços de memória subsequentes para a memória CACHE, prevendo novo acesso baseado no princípio da localidade espacial.

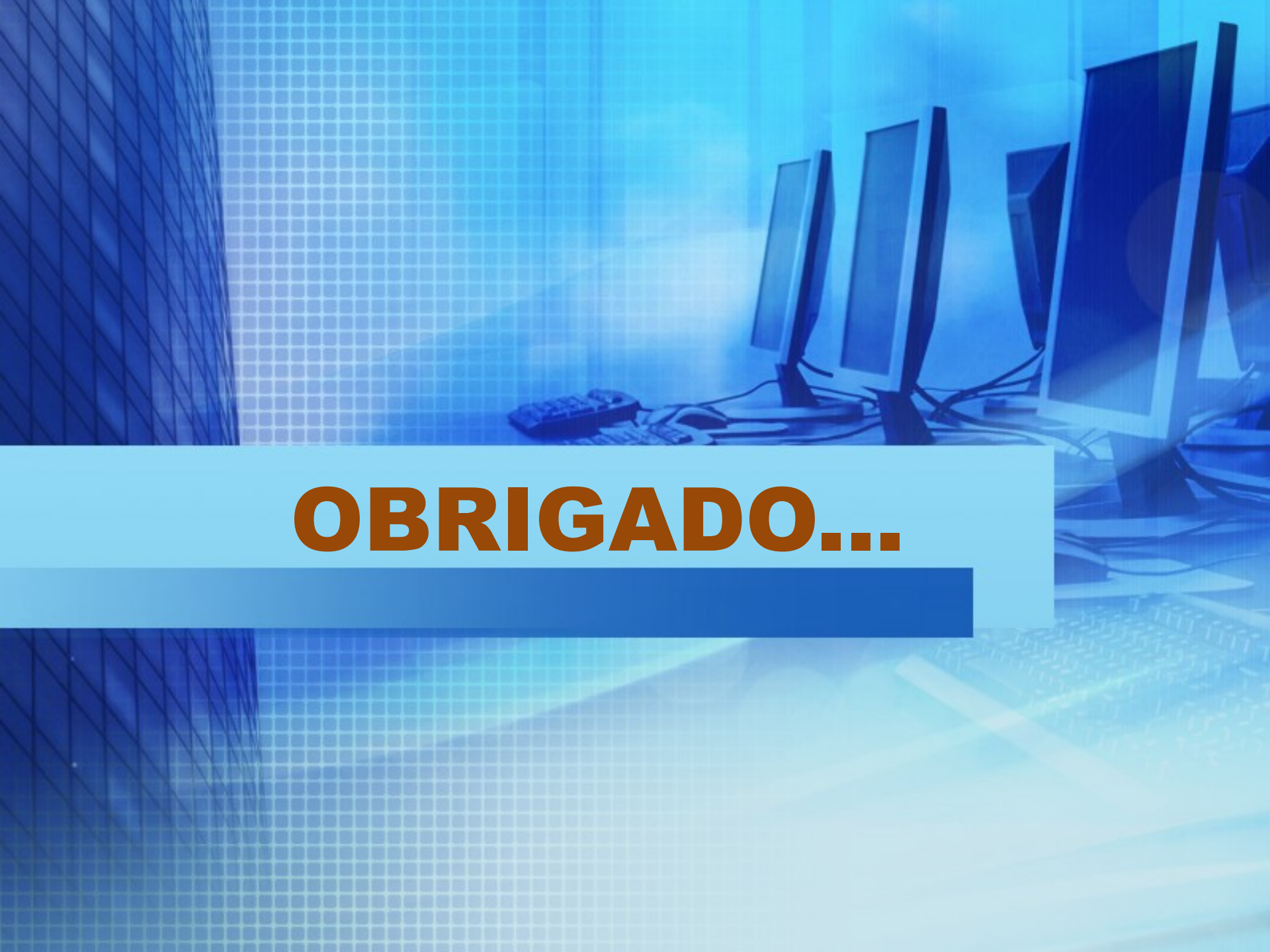
Certamente, todo essa logística conta com a unidade de controle para gerenciamento.

- **RAS:**
  Row Address Strobe
- **CAS:**
  Column Access Strobe
- **WE:**
  Write Enable
- **CS:**
  Chip Select
- **BA0, BA1:**
  Bank Address



Central processing unit (CPU)




OBRIGADO...