

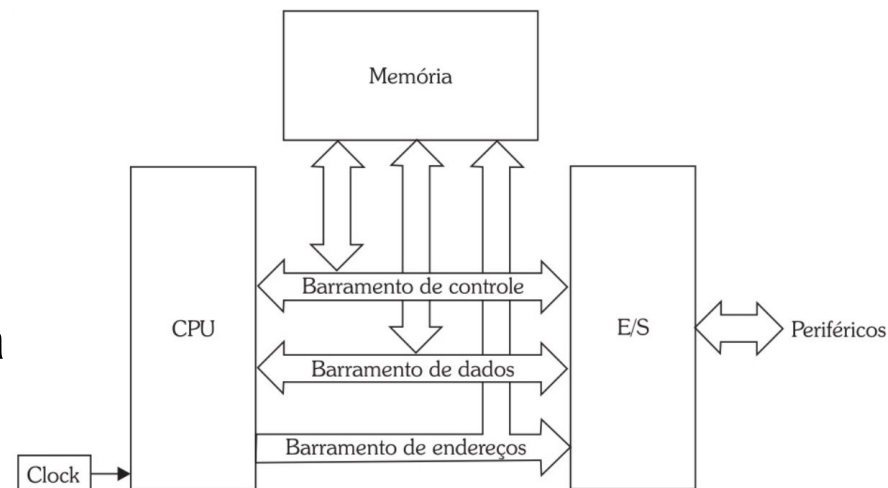
CST Análise e Desenvolvimento de Sistemas

AOC786201 - Fundamentos de Arquitetura e Organização de Computadores

Memórias

Arquitetura de John von Neumann

- Os computadores possuem blocos e sistemas funcionais básicos interligados, permitindo a troca de dados sob o controle da CPU
- Um sinal de clock determina a frequência de operação da CPU que busca uma instrução da memória, a decodifica, executa a operação correspondente, e então passa para a próxima instrução
- Nesta material focaremos na memória



Memórias: Introdução

- As memórias servem como locais de armazenamento de informações essenciais para o processamento de dados.
- Armazenam informações codificadas digitalmente, as quais podem representar uma variedade de dados, incluindo números, letras, comandos, cores e endereços.
- São classificadas por acesso, volatilidade, troca de dados e tipos de armazenamento

2

Números

↓ A
Z

Letras



Comandos



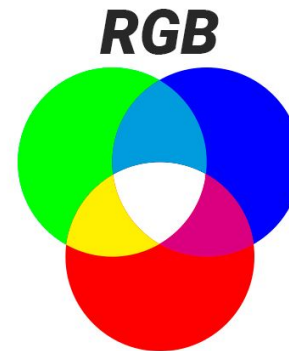
Cores



Endereços

Memórias - Exemplo de Representação: Códigos alfanuméricos, cores e comandos

Dec	Chr	Dec	Chr	Dec	Chr	Dec	Chr	Dec	Chr
0	NUL	26	SUB	52	4	78	N	104	h
1	SOH	27	ESC	53	5	79	O	105	i
2	STX	28	FS	54	6	80	P	106	j
3	ETX	29	GS	55	7	81	Q	107	k
4	EOT	30	RS	56	8	82	R	108	l
5	ENQ	31	US	57	9	83	S	109	m
6	ACK	32		58	:	84	T	110	n
7	BEL	33	!	59	;	85	U	111	o
8	BS	34	"	60	<	86	V	112	p
9	HT	35	#	61	=	87	W	113	q
10	LF	36	\$	62	>	88	X	114	r
11	VT	37	%	63	?	89	Y	115	s
12	FF	38	&	64	@	90	Z	116	t
13	CR	39	'	65	A	91	[117	u
14	SO	40	(66	B	92	\	118	v
15	SI	41)	67	C	93]	119	w
16	DLE	42	*	68	D	94	^	120	x
17	DC1	43	+	69	E	95	_	121	y
18	DC2	44	,	70	F	96	`	122	z
19	DC3	45	-	71	G	97	a	123	{
20	DC4	46	.	72	H	98	b	124	
21	NAK	47	/	73	I	99	c	125	}
22	SYN	48	0	74	J	100	d	126	~
23	ETB	49	1	75	K	101	e	127	DEL
24	CAN	50	2	76	L	102	f		
25	EM	51	3	77	M	103	g		




[CPU 80486]				D 1		[+]
cs:00F8	0100	add	[bx+si],ax	ax	000A	c=0
cs:00FA	EAF6000001	jmp	0100:00F6	bx	F54C	z=0
cs:00FF	00EB	add	bl,ch	cx	021C	s=0
cs:0101	0B900A00	or	dx,[bx+si+000A]	dx	F650	o=0
cs:0105	0A00	or	al,[bx+si]	si	F5D0	p=0
cs:0107	07	pop	es	di	9BBD	a=0
cs:0108	00900090	add	[bx+si-7000],dl	bp	0100	i=1
cs:010C	90	pop		sp	FFFE	d=0
cs:010D	2EA10301	mov	ax,cs:[0103]	ds	4928	
cs:0111	BB0400	mov	bx,0004	es	4928	
cs:0114	F7E3	mul	bx	ss	4928	
cs:0116	2EA30301	mov	cs:[0103],ax	cs	4928	
cs:011A	2EA10501	mov	ax,cs:[0105]	ip	0111	
cs:011E	2E8B1E0501	mov	bx,cs:[0105]			
cs:0123	F7E3	mul	bx			
ds:0000	CD 20 FF 9F 00 EA FF FF	=	č ř	ss:0006	FFFF	
ds:0008	AD DE E0 01 4F 16 AA 01	s0000-00		ss:0004	EA00	
ds:0010	4F 16 89 02 AA 10 1C 02	0-0000-00		ss:0002	9FFF	
ds:0018	01 01 01 00 02 FF FF FF	000 0		ss:0000	20CD	
ds:0020	FF FF FF FF FF FF FF FF			ss:FFFE	0000	

Memórias: Acesso

- No acesso sequencial, para se alcançar um endereço de uma determinada localidade deve-se passar pelas localidades intermediárias, ou seja, o tempo de acesso depende do endereço. Ex.: fitas, os discos rígidos magnéticos e CDs
- No acesso aleatório, é possível alcançar um endereço diretamente, sem que necessitemos passar pelas localidades intermediárias. Ex.: RAM, discos SSD e Pendrives/SdCards.

Tempo de acesso típicos:

- 
- Caches L1-3: 1-5 ns.
 - DDR: 3-10 ns.
 - SSD: 0,1-0,5 ms
 - HDD: 5-15 ms



Acesso Sequencial

Mais lento, dependente da
localização dos dados



Acesso Aleatório

Mais rápido, tempo de
acesso uniforme

Memórias: Volatilidade

- As memórias voláteis perdem a informação ao ser cortada a alimentação. São memórias feitas, geralmente, de Flip-Flops, células biestáveis ou células capacitor-transistor. Ex.: Registradores, Cache, DRAM e SRAM
- As memórias não voláteis continuam com as informações armazenadas mesmo na ausência de energia. Ex.: ROM, EEPROM, Flash e Discos magnéticos.



A palavra latina
"volatilis" refere-se à
dissipação fácil no
ar.



Memória Volátil

Perde dados sem energia



Memória Não Volátil

Retém dados sem energia

Memórias: Troca de dados

- As memórias apenas de leitura, possuem informação fixa, só podendo efetuar-se a leitura. São chamadas ROM (Read-Only Memory).
- As memórias de escrita/leitura permitem armazenagem e recuperação de informações. Ex.: RAM, EEPROM, Flash e Discos. Há, no entanto, as que permitem apenas uma gravação como a PROM e os CDs.



ROMs são pouco utilizadas, mesmo as BIOS atuais são programáveis.



Memória Somente de Leitura

Permite apenas a recuperação de dados



Memória de Leitura/Escrita

Permite tanto o armazenamento quanto a recuperação

Memórias: Tipos de armazenamento

- As memórias estáticas são aquelas que mantêm a informação gravada de forma indefinida, desde que a energia elétrica esteja presente.
- As memórias dinâmicas requerem um "refresh" periódico, pois a informação armazenada se degrada após um certo tempo, necessitando de recargas para manter os dados acessíveis.



Tipicamente as DRAM precisam de refresh a cada 64 ms, mas pode ser mais frequente em temperaturas altas



Memória Estática

Fornece retenção de dados indefinida com energia



Memória Dinâmica

Requer atualização periódica para manter os dados

Comunicação entre memórias

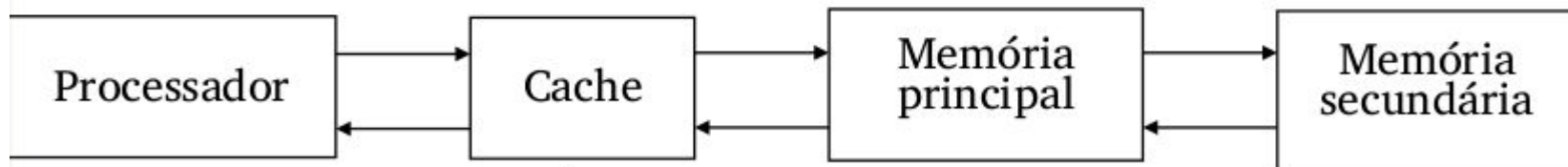
- São armazenados nas memórias de alta velocidade os dados e instruções que o processador vai utilizar com mais frequência.
- Em memórias mais lentas, com grande capacidade de armazenamento podem ser usadas para guardar dados e instruções que não serão necessários naquele momento
- As memórias são organizadas hierarquicamente de forma a obtermos um sistema com desempenho (velocidade) próximo ao da memória mas rápida e custo por bit próximo ao da memória de menor custo

Comunicação entre memórias

- Memória em 2 níveis:



- Memória em 3 níveis:



→ A partir do Intel 80486 (8 Kbytes)

Hierarquia de Memórias

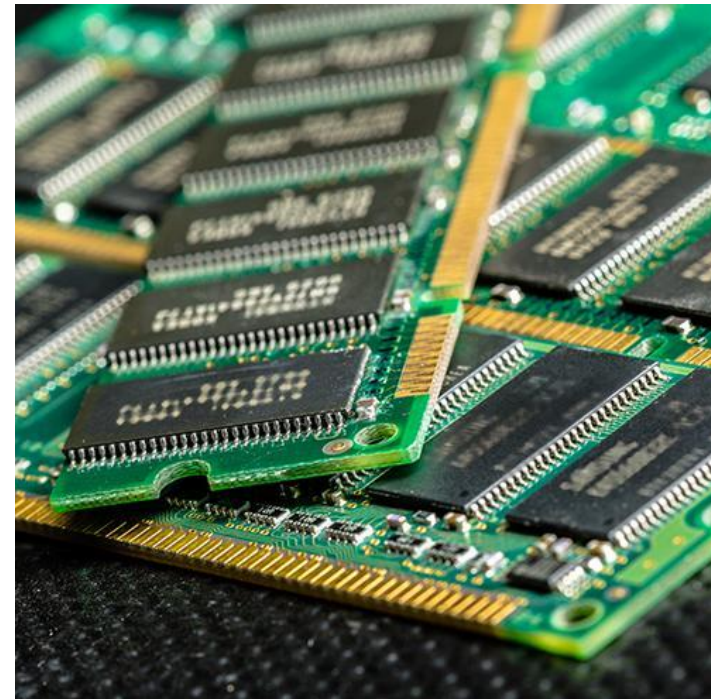


Memórias comerciais

- Memória ROM (Read Only Memory) são estáticas e não voláteis já vindo gravada de fábrica com informações, permitindo apenas leitura
- Memória PROM (Programmable Read Only Memory) são estáticas e não voláteis e permitem uma única gravação que ficará por definitivo
- Memória EEPROM (Electrically Erasable Read Only Memory) são estáticas e não voláteis que permitem leitura e escrita
 - Suportam de 10.000 a 1.000.000 de ciclos de escrita
- Memória Flash são estáticas e não voláteis que permitem leitura e escrita, são mais velozes se comparadas com memórias EEPROM
 - Suportam de 10.000 a 100.000 de ciclos de escrita

Memórias comerciais - DRAM

- Memória RAM (Rando-Access Memory) são voláteis.
- Podem ser estáticas ([SRAM](#)) que são normalmente construídas com Flip-Flops ou dinâmicas ([DRAM](#)) que são construídas com circuitos mais simples (latches).
- Não tem limite prático em termos de ciclos de escrita
- Vêm sofrendo evolução ao longo das últimas décadas, tendo o tempo de acesso que era de cerca de 100 ns chegando atualmente a 5 ns.



Memórias comerciais - DRAM

Tecnologia	Melhoria principal	bits	Latência	Pico Transf.
DRAM tradicionais	Assíncronas, tem controlador p/ leitura, escrita e refresh	8/16	80 a 120 ns	1 a 16 MB/s
FPM DRAM Fast Page Mode	Acelerou acesso a dados consecutivos da página	32	50 a 70 ns	20 a 50 MB/s
EDO DRAM Extended Data Out	Leitura mais rápida mantendo dados anteriores na saída	32	40 a 50 ns	50 a 100 MB/s
SDRAM Synch. Dynamic	são síncronas com o clock do processador	64	10 a 15 ns	133 a 800 MB/s
DDR SDRAM Double Data Rate	Transferência em ambas as bordas de subida e descida	64	7 a 10 ns	2,1 a 3,2 GB/s
DDR2, DDR3, DDR4	Melhor desempenho, menor consumo, maior densidade e frequência de operação	64	3 a 7,5 ns	4 a 6 GB/s 8 a 15 GB/s 17 a 25 GB/s
DDR5	Barramentos independentes por banco e refresh mais efíc.	64 canais diferentes	3 ns	32 a 51 GB/s

Memórias comerciais - DRAM

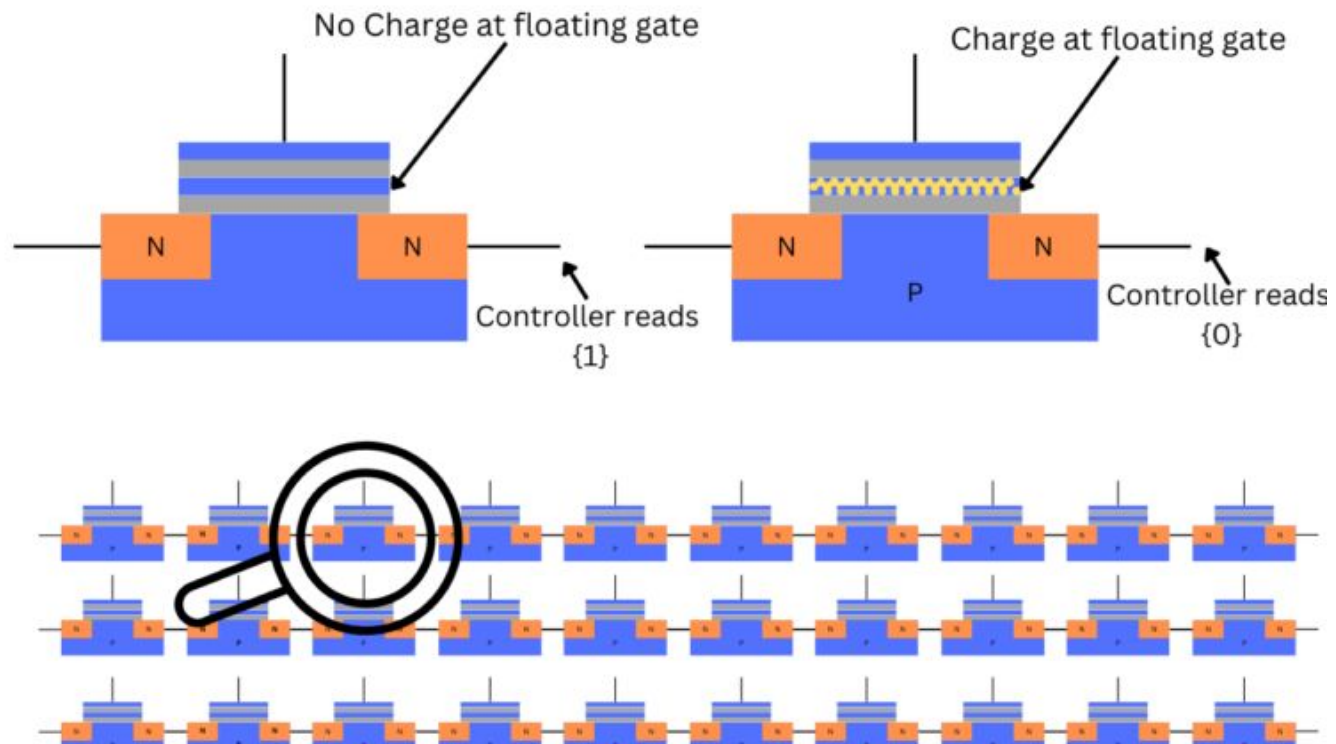
- Algumas tecnologias DRAM:
 - DRAM tradicionais são assíncronas, gerenciadas por um controlador que trata leitura, escrita e refresh (80 ~ 120 ns, 8 ou 16 bits, 1 a 16 MB/s).
 - FPM DRAM (Fast Page Mode DRAM) acelerou acesso dentro da página (50 ~ 70 ns, 20 a 50 MB/s)
 - EDO DRAM (Extended Data Out DRAM) leitura mais rápida (40 ~ 50 ns, 50 a 100 MB/s)
 - SDRAM (Synchronous Dynamic RAM) são síncronas com o clock do processador (10 ~ 15 ns, 133 a 800 MB/s)
 - DDR SDRAM (Double Data Rate SDRAM) operam em ambas as bordas de subida e descida (taxas de transf. de 2,1 a 3,2 GB/s)
 - DDR2, DDR3, DDR4, DDR5: melhor desempenho, menor consumo e maior densidade de dados (taxas de transferência 4,27 a 6,4 / 8,5 a 14,9 / 17 a 25,6 / 32 a 51.2 GB/s)

Outros tipos de RAMs

- NVRAM – RAM não-volátil: Armazenamento de Dados com o Sistema Desligado. Muitos equipamentos armazenam seus dados em memórias RAM alimentadas com baterias de reserva.
- Memória FIFO (First-In, First-Out): também conhecidas como buffers para armazenamento temporário. Por exemplo, os dados são transferidos em altas taxas do PC para a impressora. Como a impressão é mais lenta que a transferência, esses dados devem ser armazenados em FIFOs e daí podem ser impressos.

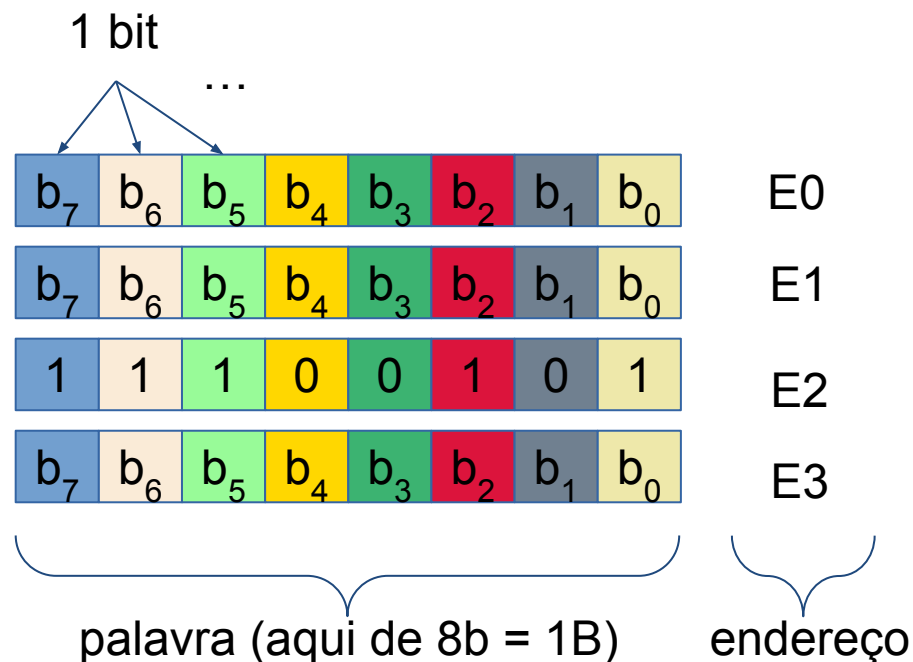
Memórias que utilizam FGTs

- Memórias EEPROM e Flash são formadas por células que contêm Floating Gate Transistors (Transistores de porta flutuante) que permitem armazenamento de dados de forma não volátil

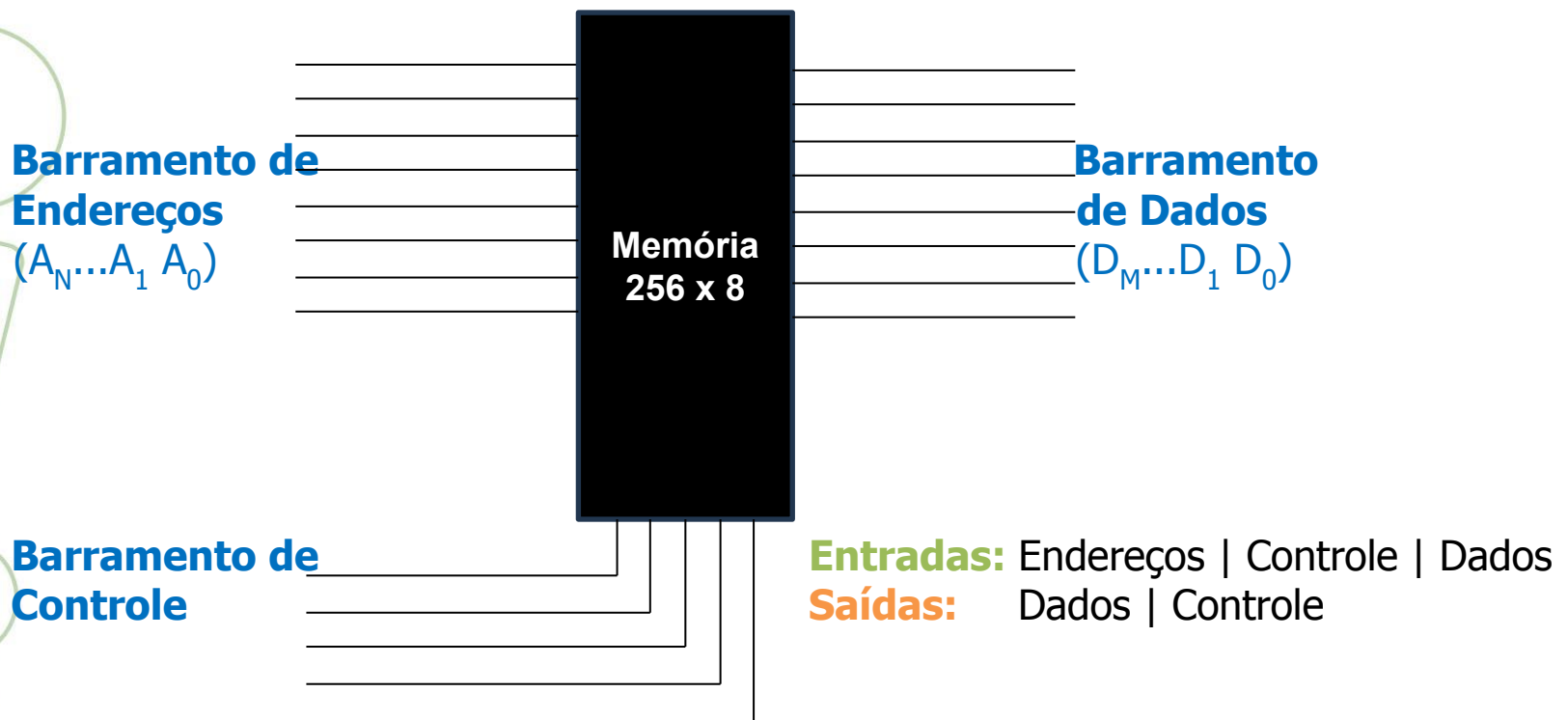


Interação entre Processador e Memória Principal

As células de bit não normalmente não são endereçáveis individualmente, mas em conjuntos (palavras ou bytes)



Memória: pinos de entrada e saída



Barramento de dados

- Conjunto de pinos utilizados para disponibilizar dados para serem gravados na memória ou para que a memória disponibilize dados que serão lidos por um sistema externo
- A quantidade de bits segue normalmente o tamanho da palavra utilizado pela memória
 - 4 bits (Nibble)
 - 8 bits (Byte)
 - 16 bits
 - 32 bits
 - 64 bits

Barramento de endereços

- Conjunto de pinos utilizados para apontar uma posição da memória
- Cada posição identifica um agrupamento de bits (cujo tamanho depende do tamanho da palavra adotado pela memória)
 - Cada palavra deve ter um endereço único
- A quantidade de pinos (bits) do barramento depende da quantidade de palavras endereçáveis (para “n” a quantidade de fios, a quantidade de palavras deve ser $\leq 2^n$)

Comandos de controle comuns

EN,CS,CE: Seleção (ou habilitação) do chip

RD,OE: Habilitar leitura de dados (“read”)

WR: Permitir escrita/gravação de dados (“write”)

RW: Híbrido para leitura ou escrita/gravação

CL: Limpa toda a memória (dados \square 0)

PS: Preseta toda a memória (dados \square 1)

Escrevendo na memória: simplificado

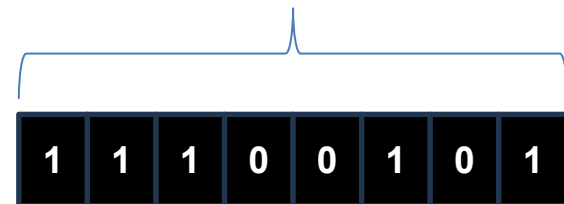
- Habilita a memória
- Informar o endereço de destino
- Disponibilizar o dado
- Dar comando de escrita

Ex.: Escrever o byte **1110 0101** no endereço **E2**

b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀	E0
b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀	E1
1	1	1	0	0	1	0	1	E2
b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀	E3

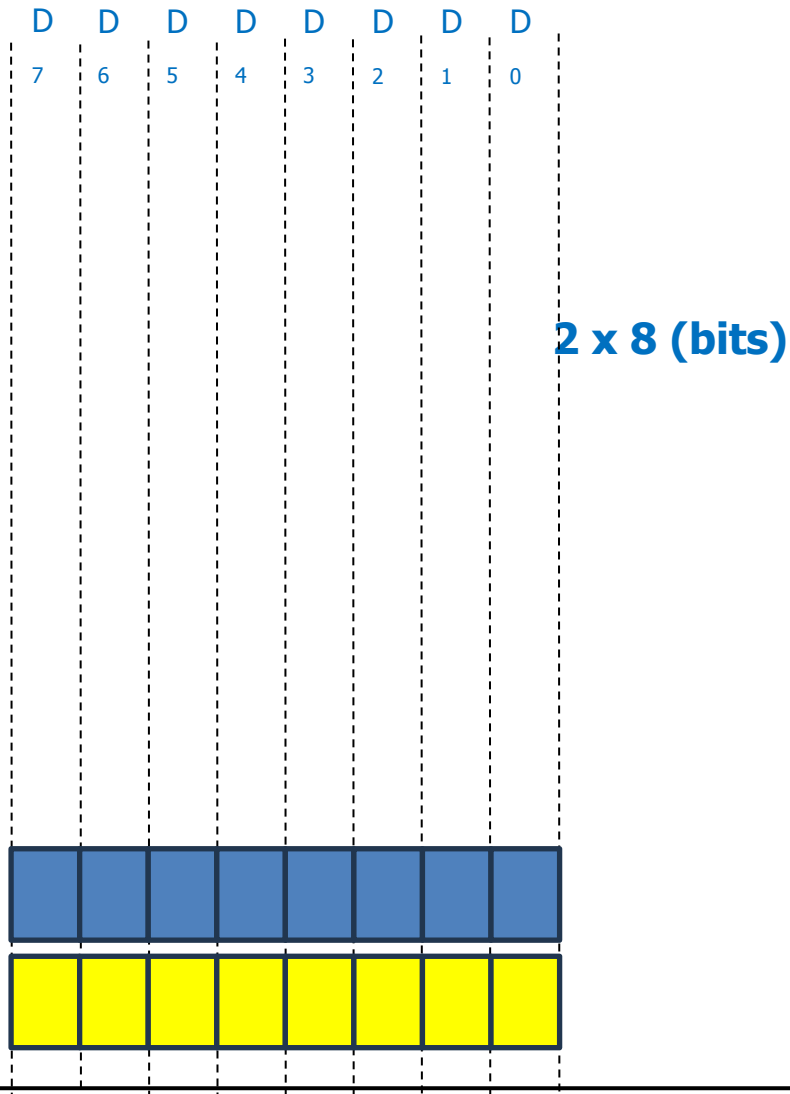


barramento de dados



Escrevendo na memória:

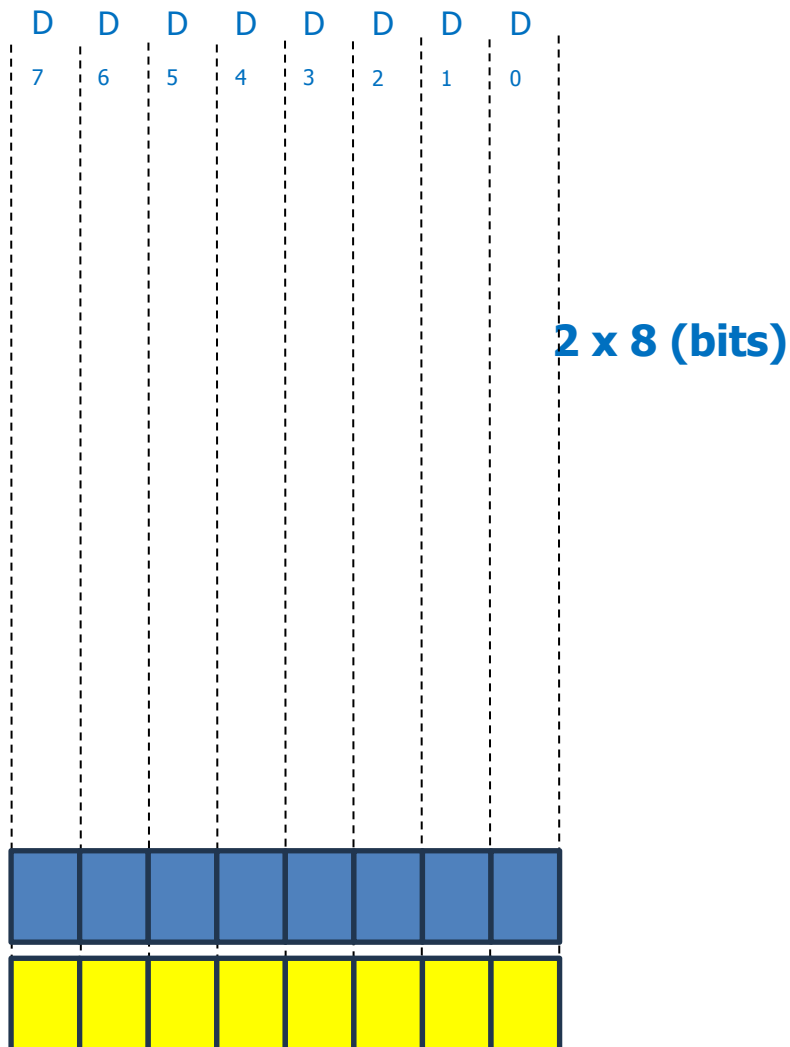
Memória 2 x 8 bits



Objetivo:
Escrever '#'
no endereço 0

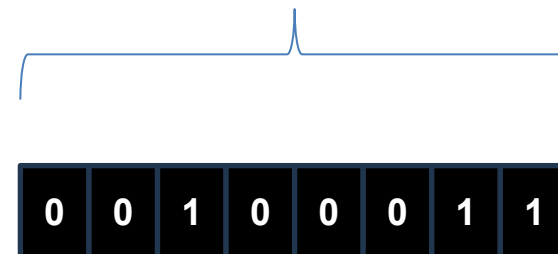
Escrevendo na memória:

Memória 2 x 8 bits



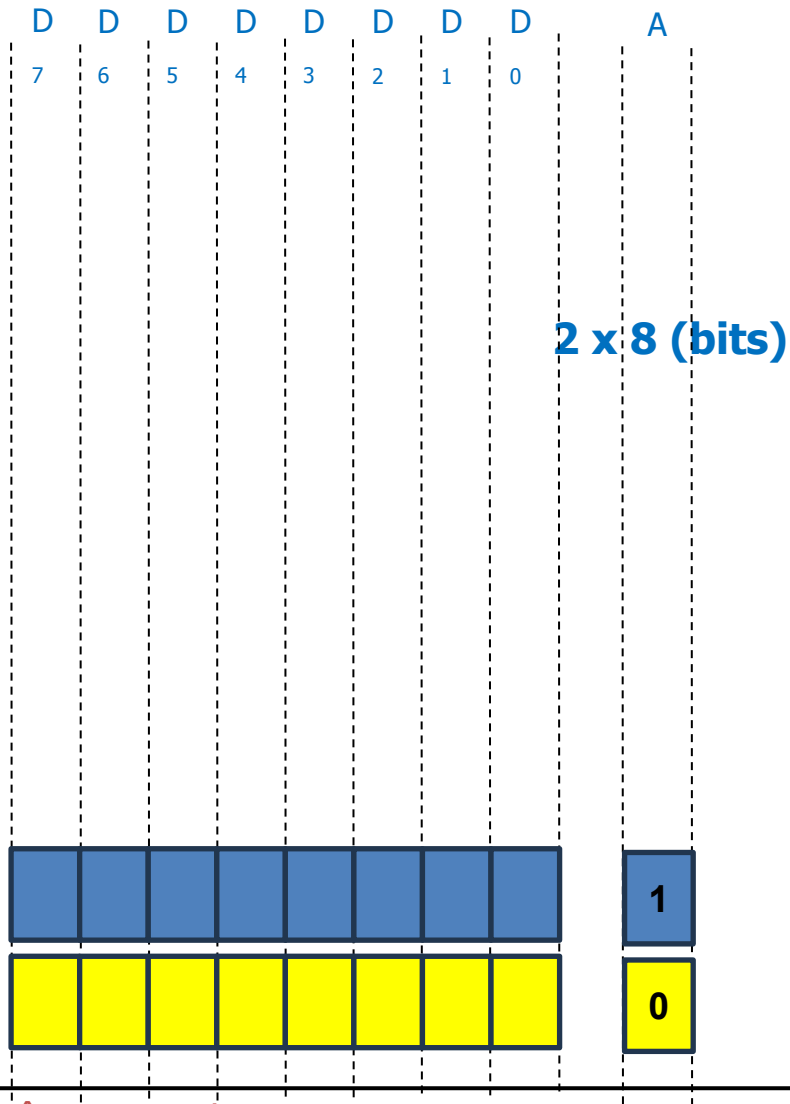
Objetivo:
Escrever '#'
no endereço 0

#



Escrevendo na memória:

Memória 2 x 8 bits



Objetivo:
Escrever '#'
no endereço 0

Escrevendo na memória:

Memória 2 x 8 bits

D D D D D D D D
7 6 5 4 3 2 1 0

2 x 8 (bits)

$\overline{EN} = 0$

Objetivo:
Escrever '#'
no endereço 0



Escrevendo na memória:

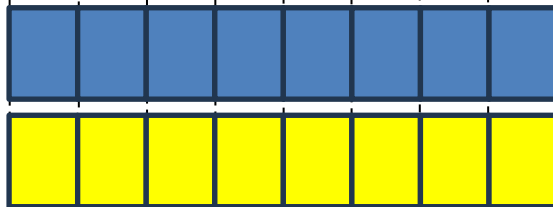
Memória 2 x 8 bits

D D D D D D D D
7 6 5 4 3 2 1 0

2 x 8 (bits)

$\overline{EN} = 0$
 $A = 0$

Objetivo:
Escrever '#'
no endereço 0



Escrevendo na memória:

Memória 2 x 8 bits

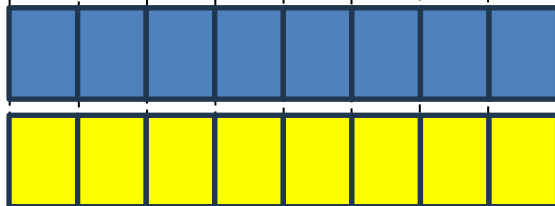
D D D D D D D D
7 6 5 4 3 2 1 0

2 x 8 (bits)

$\overline{EN} = 0$

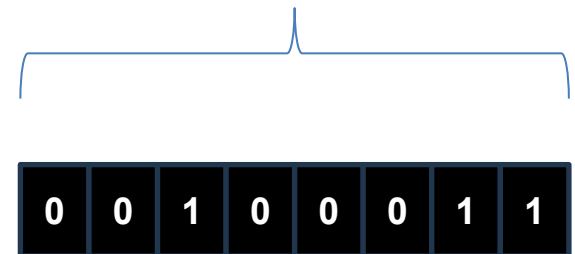
A = 0

Dados



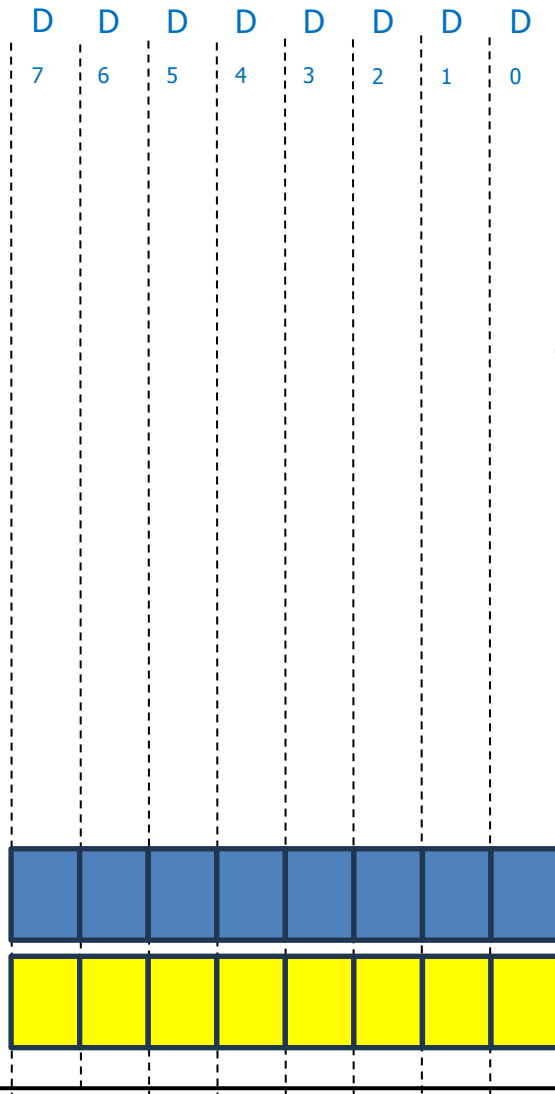
Objetivo:
Escrever '#'
no endereço 0

#



Escrevendo na memória:

Memória 2 x 8 bits



2 x 8 (bits)

$\overline{EN} = 0$

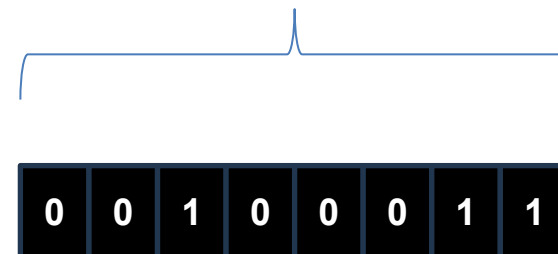
$A = 0$

Dados

$\overline{WR} = 0$

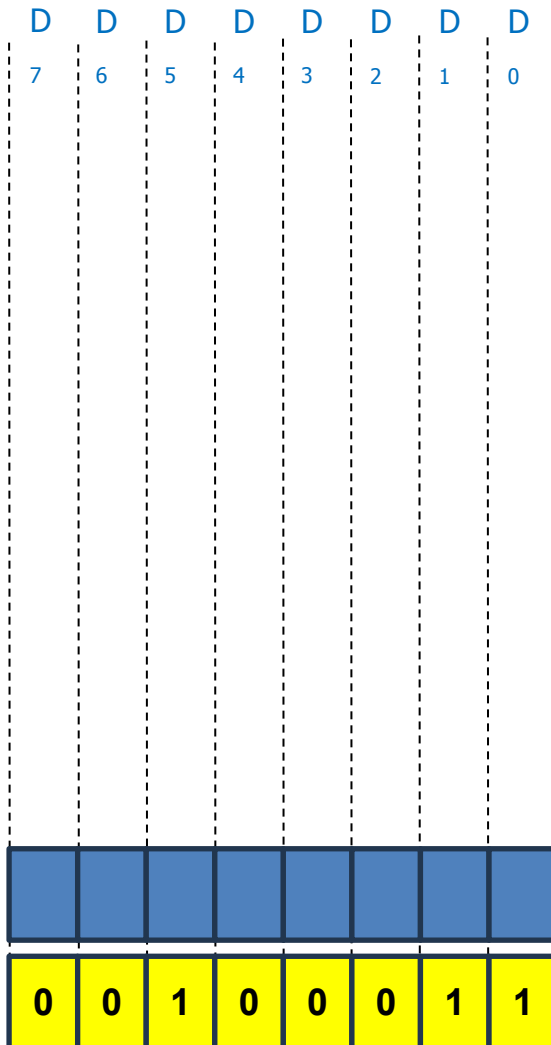
Objetivo:
Escrever '#'
no endereço 0

#



Escrevendo na memória:

Memória 2 x 8 bits



2 x 8 (bits)

$\overline{EN} = 0$

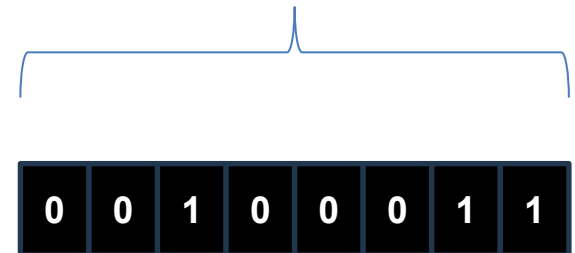
$A = 0$

Dados

$\overline{WR} = 0$

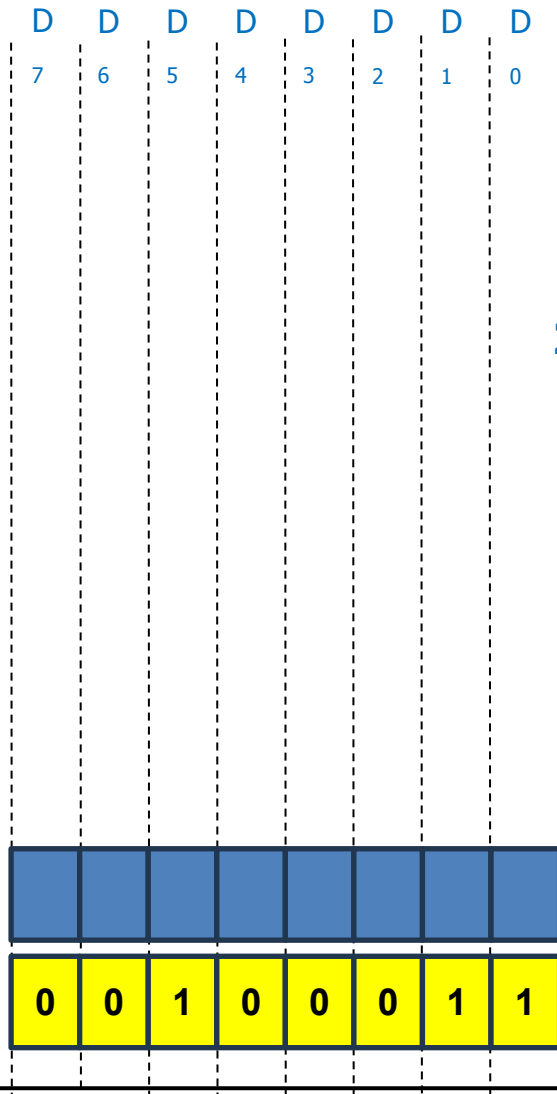
Objetivo:
Escrever '#'
no endereço 0

#



Escrevendo na memória:

Memória 2 x 8 bits



2 x 8 (bits)

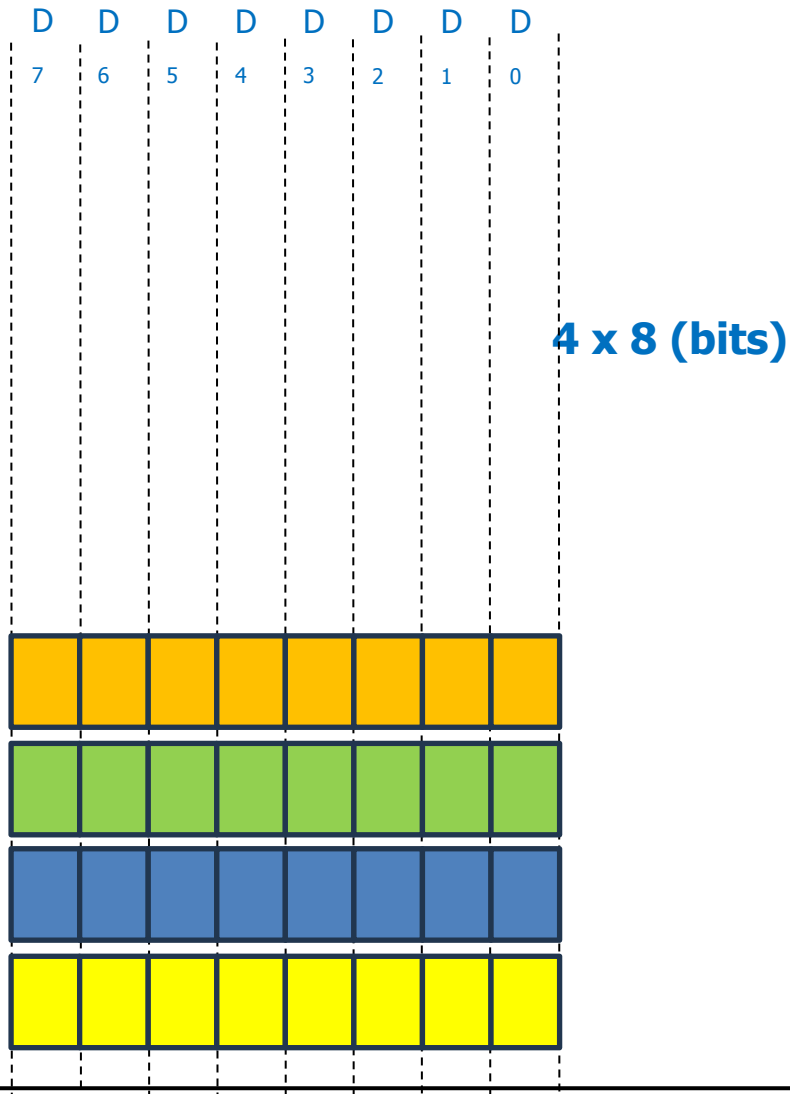
$\overline{EN} = 1$

$\overline{WR} = 1$

Objetivo:
Escrever '#'
no endereço 0

Escrevendo na memória:

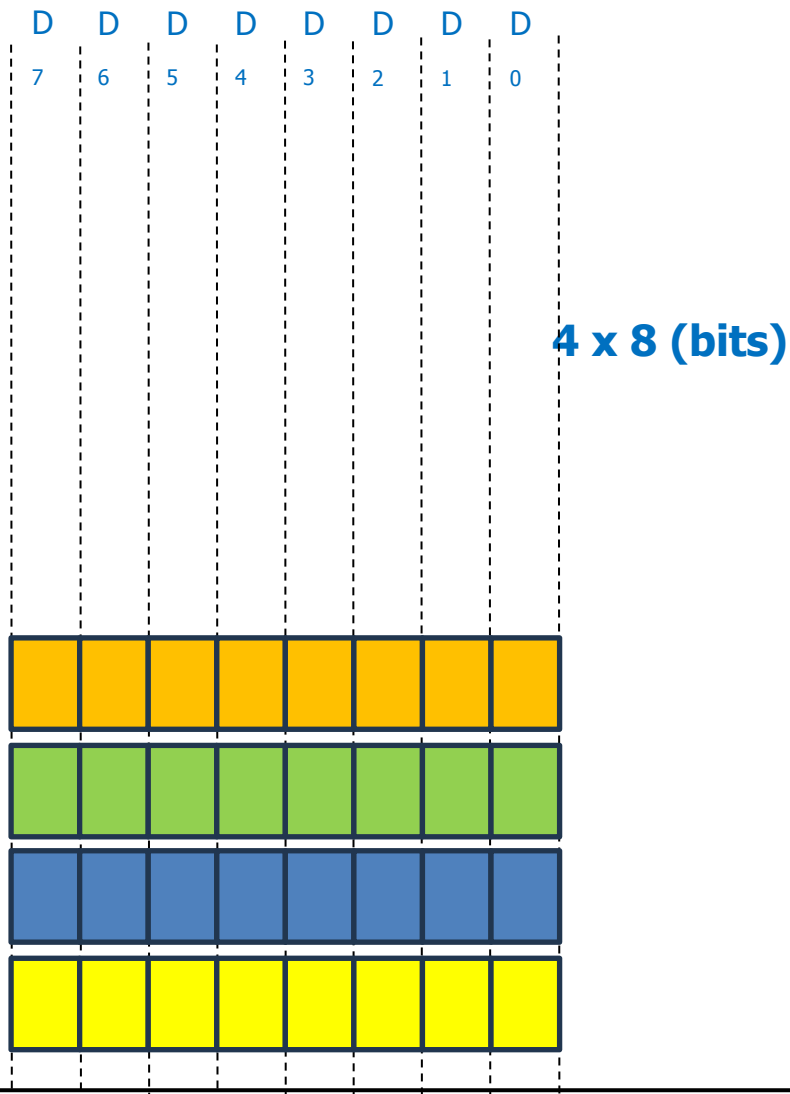
Memória 4 x 8 bits



Objetivo:
Escrever '#'
no endereço
0b10

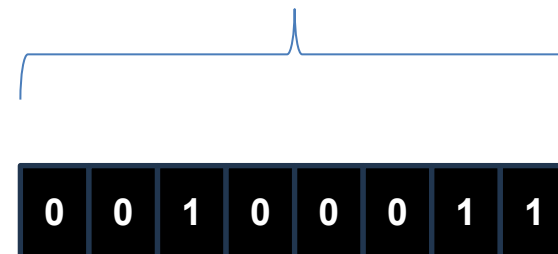
Escrevendo na memória:

Memória 4 x 8 bits



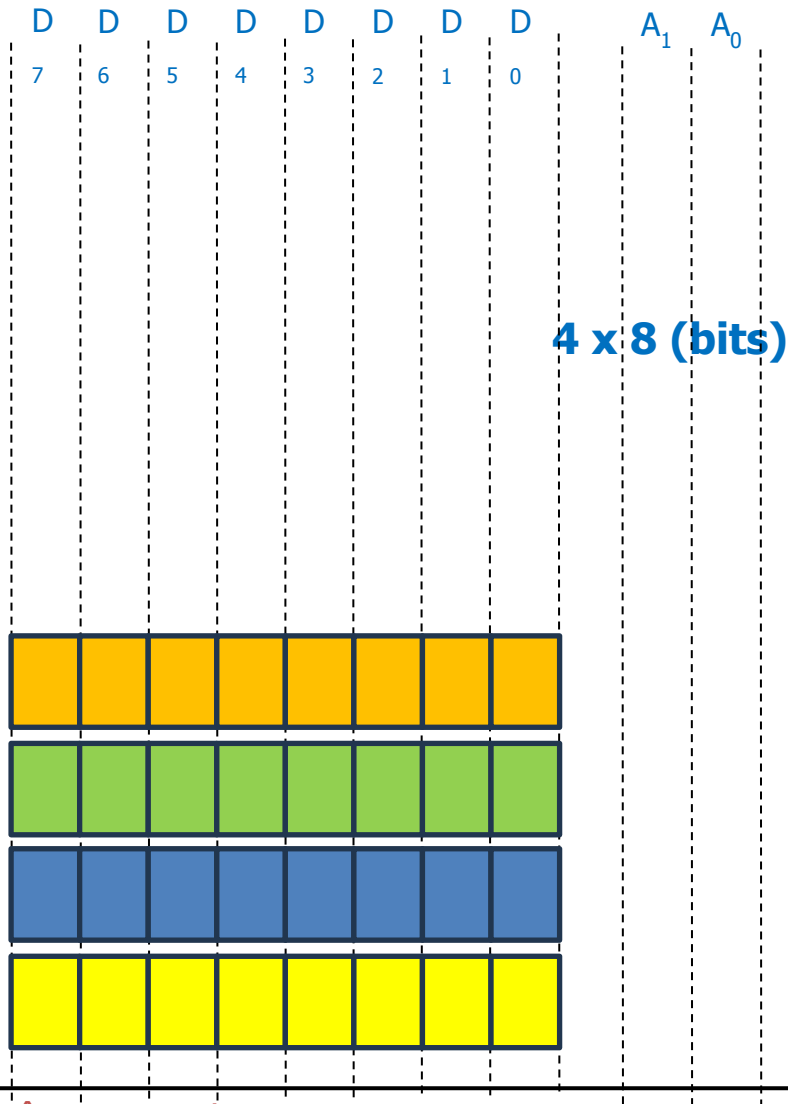
Objetivo:
Escrever '#'
no endereço
0b10

#



Escrevendo na memória:

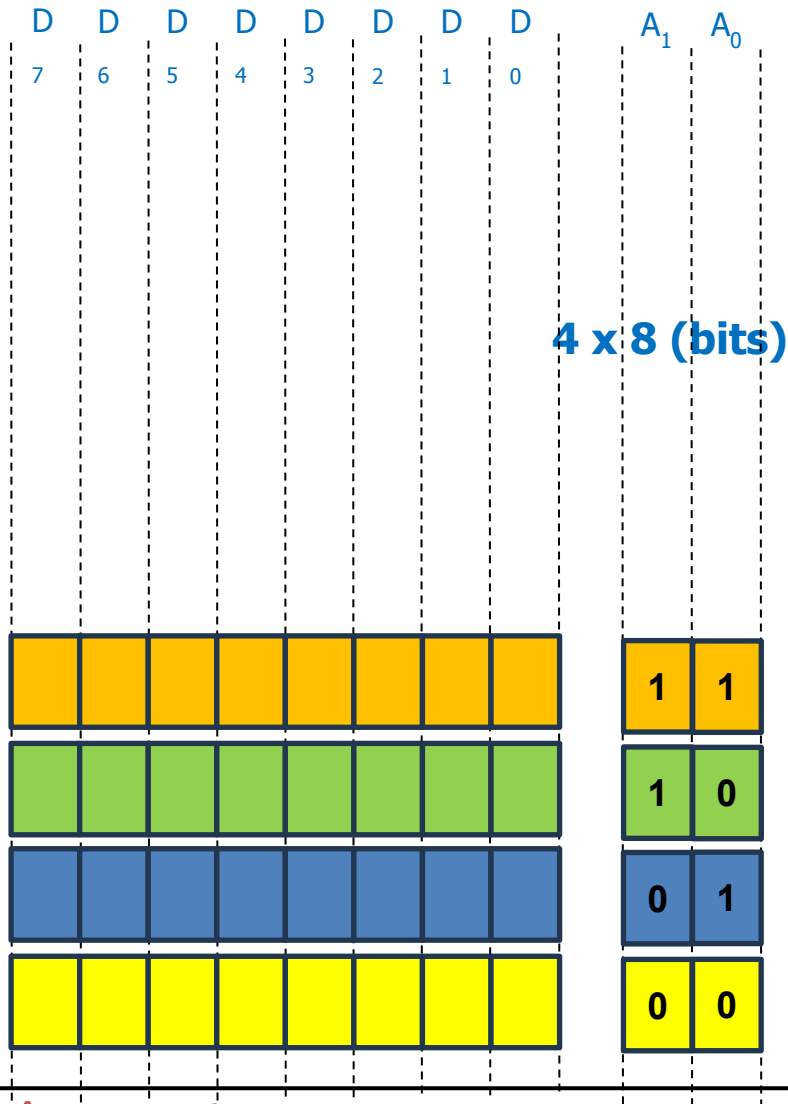
Memória 4 x 8 bits



Objetivo:
Escrever '#'
no endereço
0b10

Escrevendo na memória:

Memória 4 x 8 bits



Objetivo:
Escrever '#'
no endereço
0b10

Escrevendo na memória:

Memória 4 x 8 bits

D D D D D D D D
7 6 5 4 3 2 1 0

4 x 8 (bits)

$\overline{EN} = 0$



Objetivo:
Escrever '#'
no endereço
0b10

Escrevendo na memória:

Memória 4 x 8 bits

D D D D D D D D
7 6 5 4 3 2 1 0

4 x 8 (bits)

$\overline{EN} = 0$
 $A = 10$



Objetivo:
Escrever '#'
no endereço
0b10

Escrevendo na memória:

Memória 4 x 8 bits

D D D D D D D D
7 6 5 4 3 2 1 0

4 x 8 (bits)

$\overline{EN} = 0$

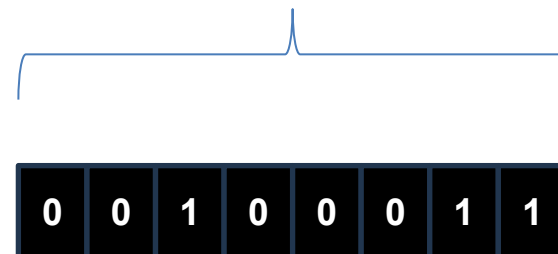
$A = 10$

Dados



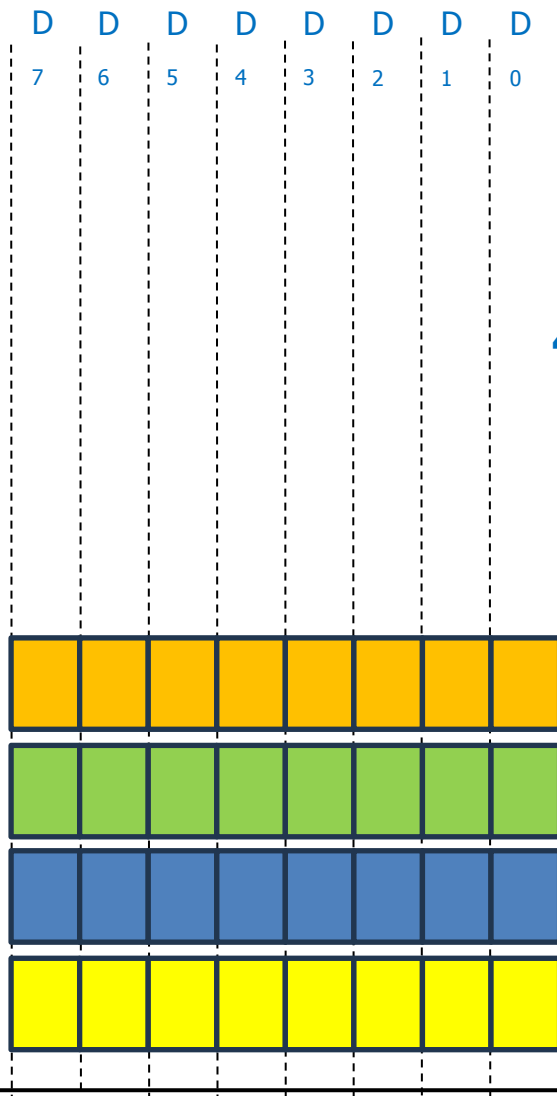
Objetivo:
Escrever '#'
no endereço
0b10

#



Escrevendo na memória:

Memória 4 x 8 bits



4 x 8 (bits)

$\overline{EN} = 0$

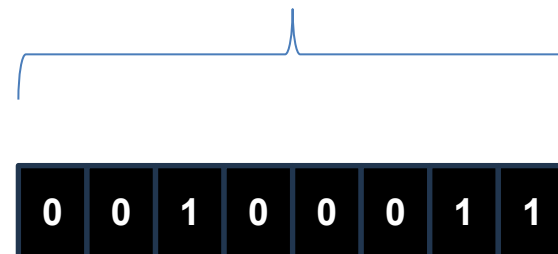
$A = 10$

Dados

$\overline{WR} = 0$

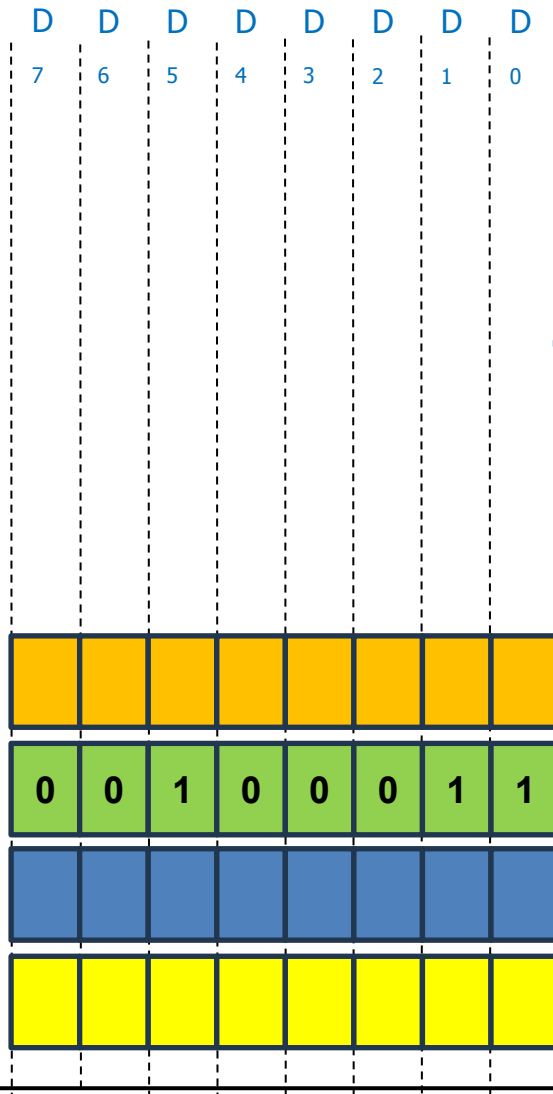
Objetivo:
Escrever '#'
no endereço
0b10

#



Escrevendo na memória:

Memória 4 x 8 bits



4 x 8 (bits)

$\overline{EN} = 0$

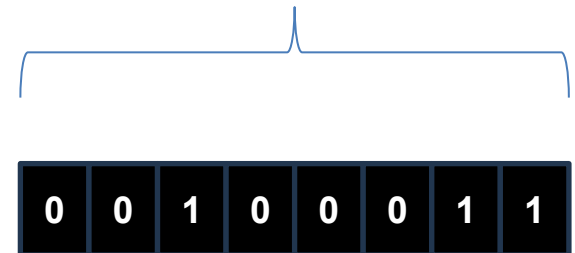
$A = 10$

Dados

$\overline{WR} = 0$

Objetivo:
Escrever '#'
no endereço
0b10

#



Escrevendo na memória:

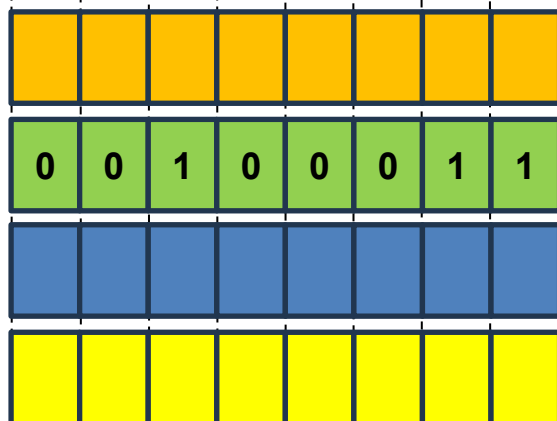
Memória 4 x 8 bits

D D D D D D D D
7 6 5 4 3 2 1 0

4 x 8 (bits)

$\overline{EN} = 1$

$\overline{WR} = 1$



Objetivo:
Escrever '#'
no endereço
0b10

Capacidade da memória

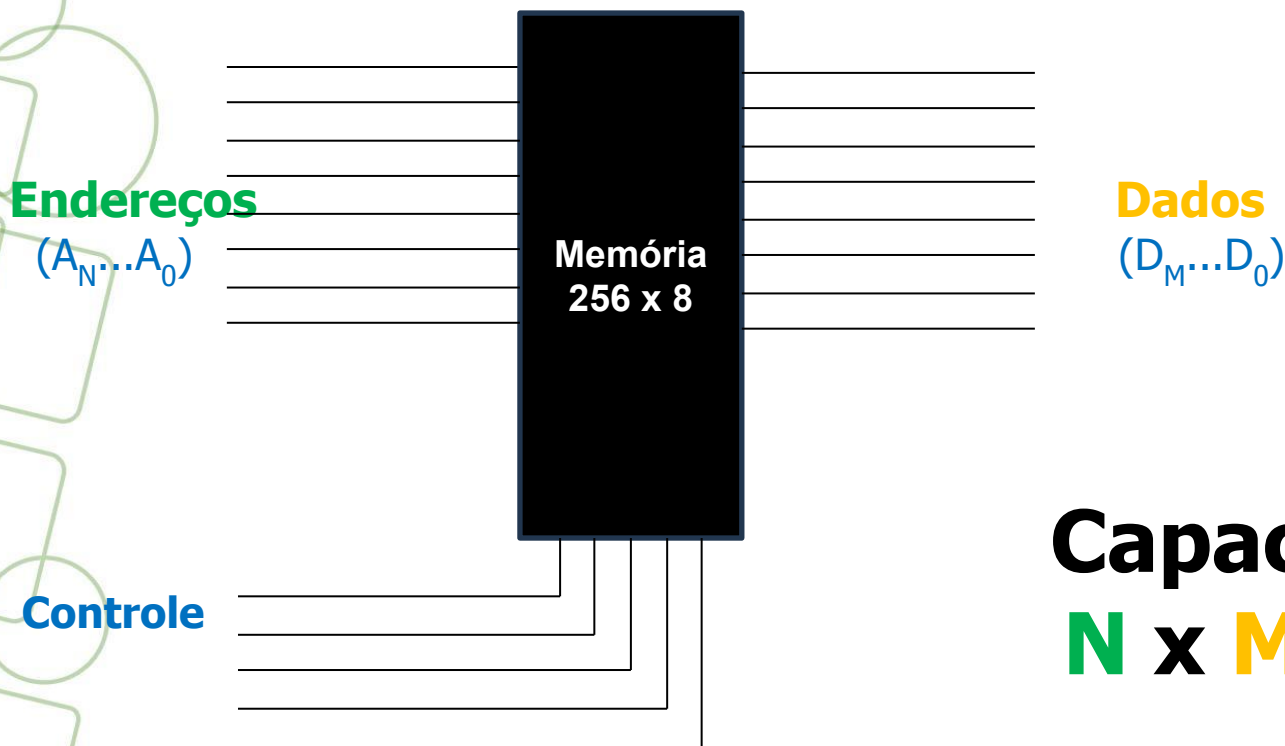
- É a quantidade de informação que pode ser armazenada em uma memória.
- Medida em bits ou Bytes

4 registradores de 8 bits

Tamanho de 32 bits = 4 x 8

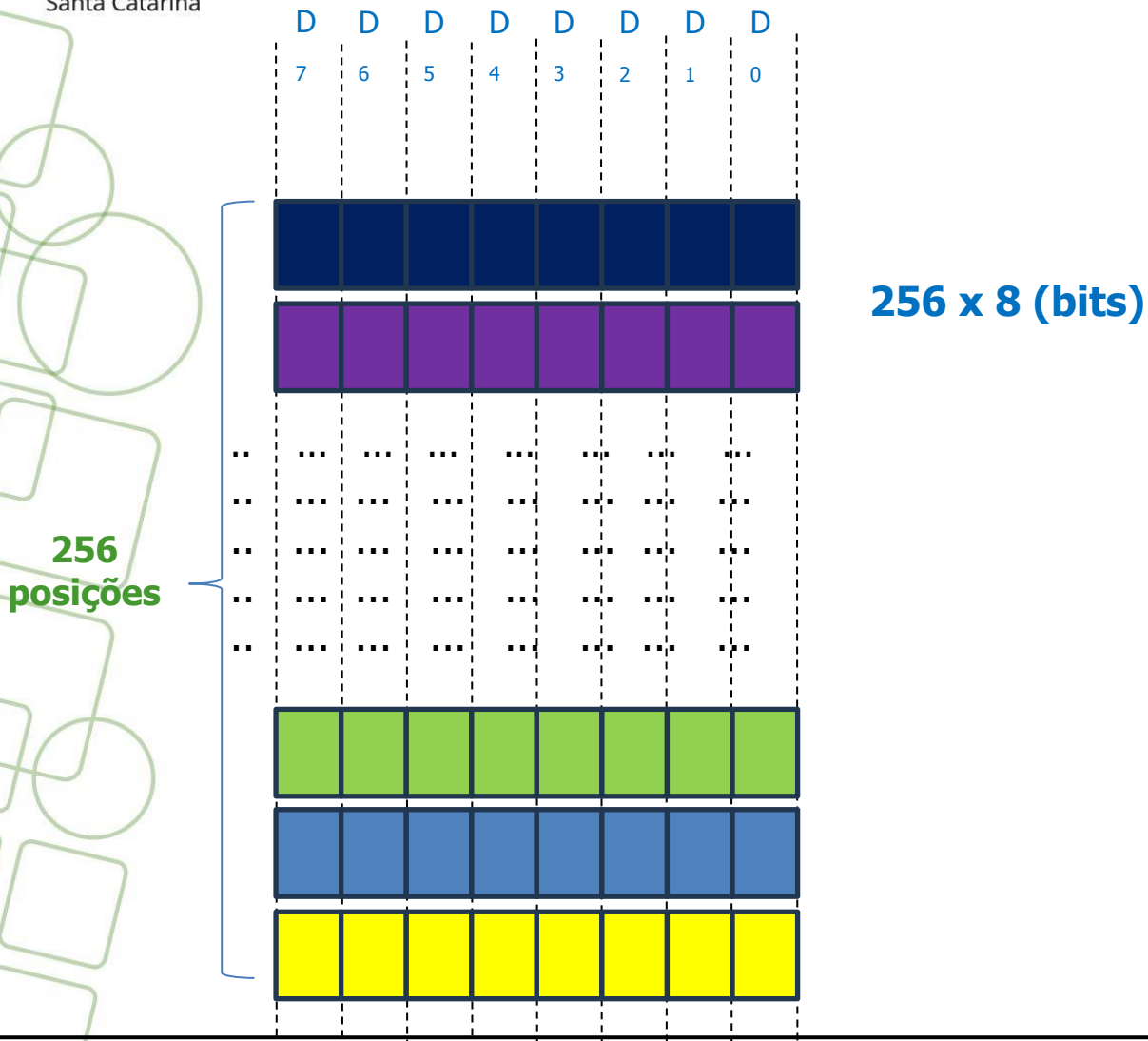


Capacidade da memória



Capacidade
N x M (bits)

Capacidade da memória

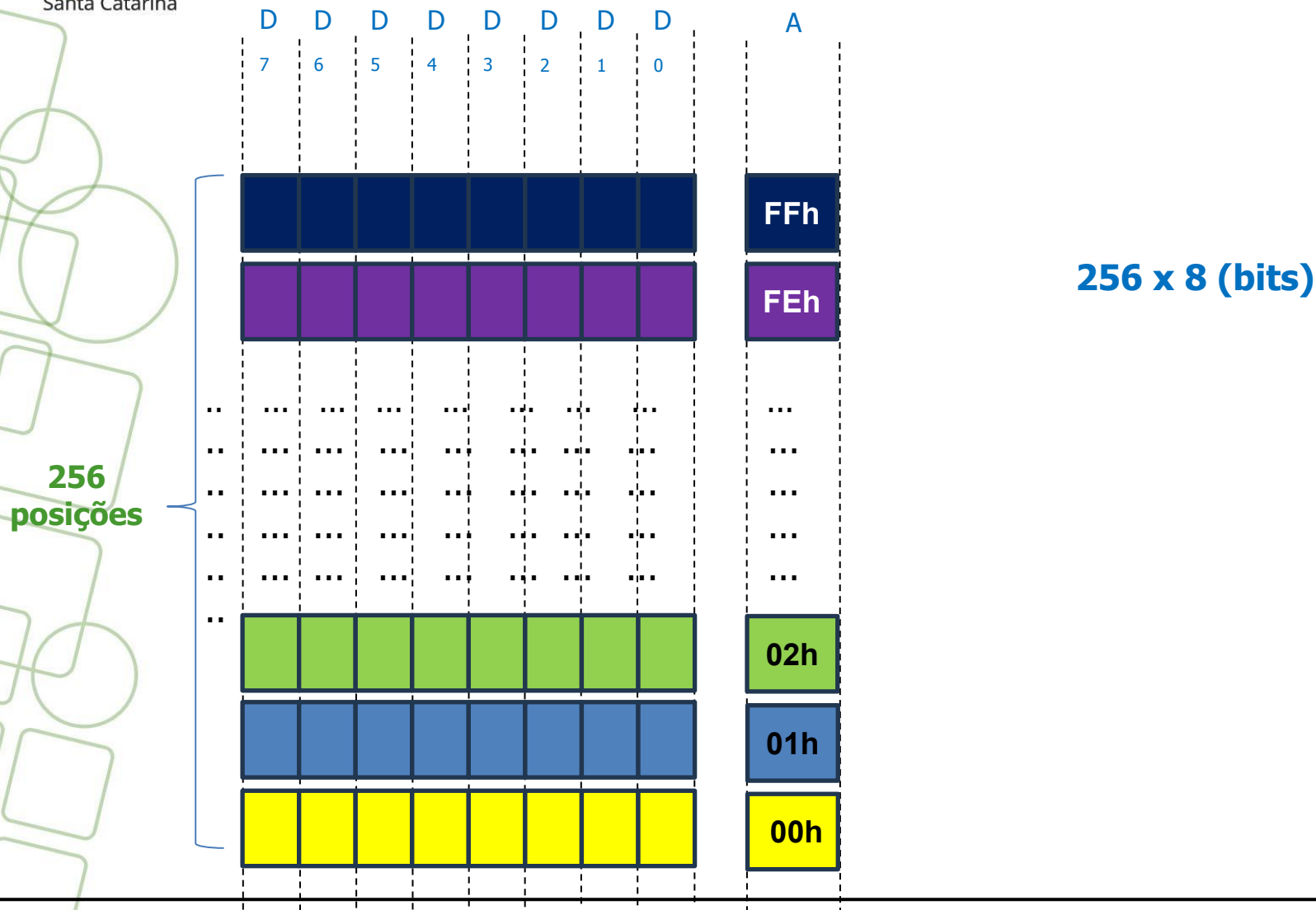


Capacidade da memória

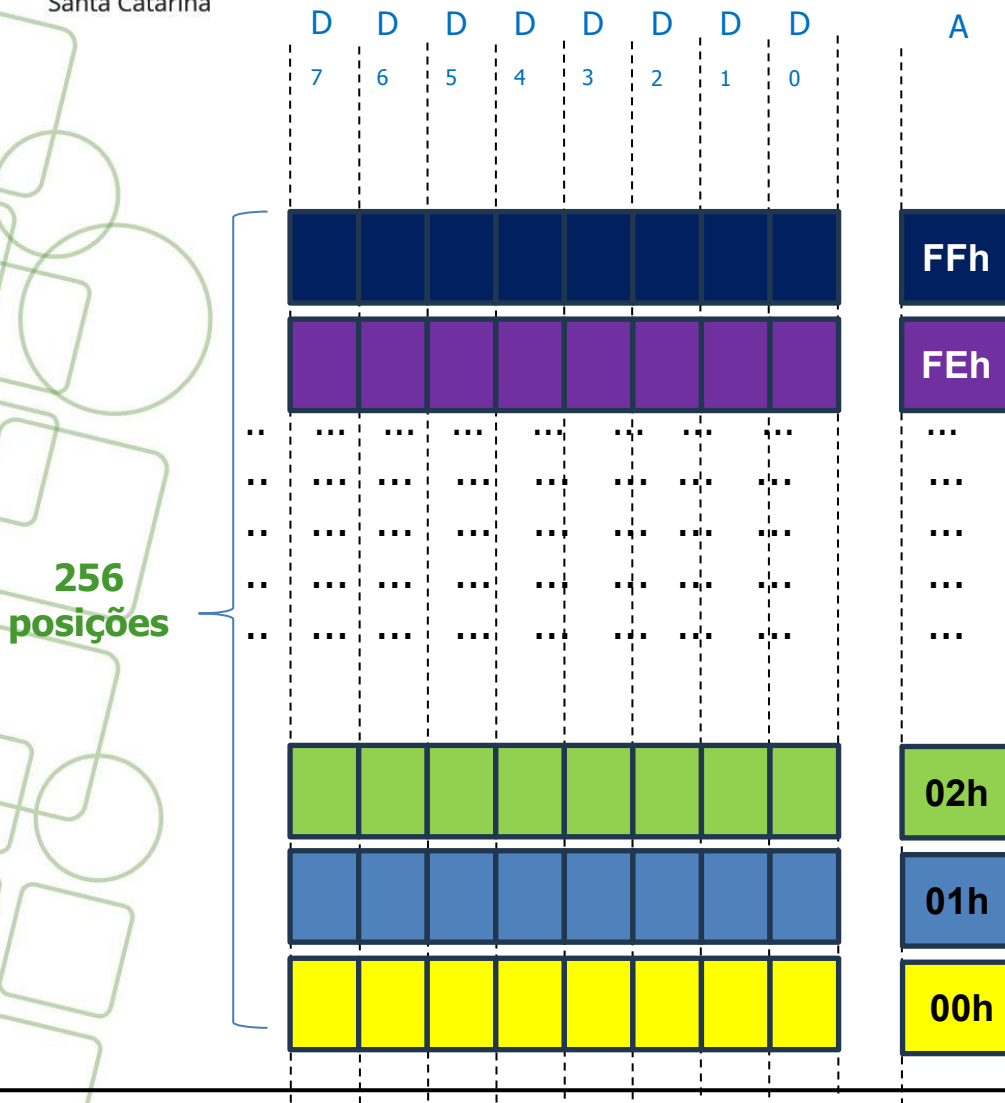
256
posições

D	D	D	D	D	D	D	D	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
7	6	5	4	3	2	1	0								
								1	1	1	1	1	1	1	1
								1	1	1	1	1	1	1	0
...
...
...
...
...
								0	0	0	0	0	0	1	0
								0	0	0	0	0	0	0	1
								0	0	0	0	0	0	0	0

Capacidade da memória



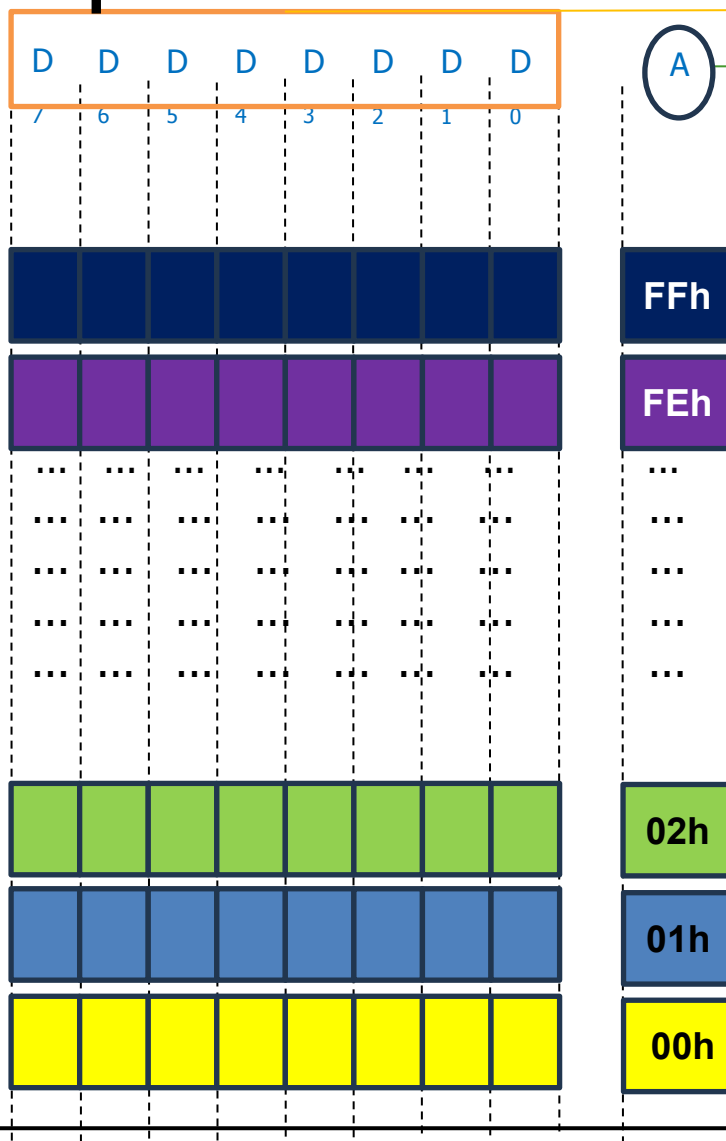
Capacidade da memória



$$2^8 \times 8 \text{ (bits)} = 2 \text{ kbits}$$

$$256 \times 8 \text{ (bits)} = 2048 \text{ bits}$$

Capacidade da memória



256
posições

$$2^8 \times 8 \text{ (bits)} = 2 \text{ kbits}$$

$$256 \times 8 \text{ (bits)} = 2048 \text{ bits}$$

$$256 \times 1 \text{ (byte)} = 256 \text{ B}$$

Leitura de uma memória: simplificado

- Habilita a memória
- Informar o endereço de destino
- Dar comando de leitura
- Ler o dado no barramento de dados

Ex.: Lendo o byte armazenado no endereço **E2**

b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀	E0
b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀	E1
1	1	1	0	0	1	0	1	E2
b ₇	b ₆	b ₅	b ₄	b ₃	b ₂	b ₁	b ₀	E3

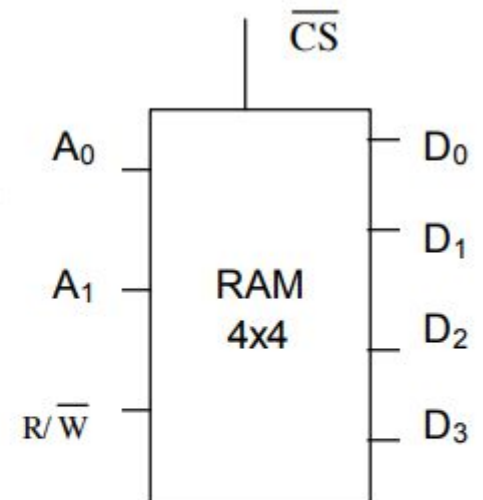
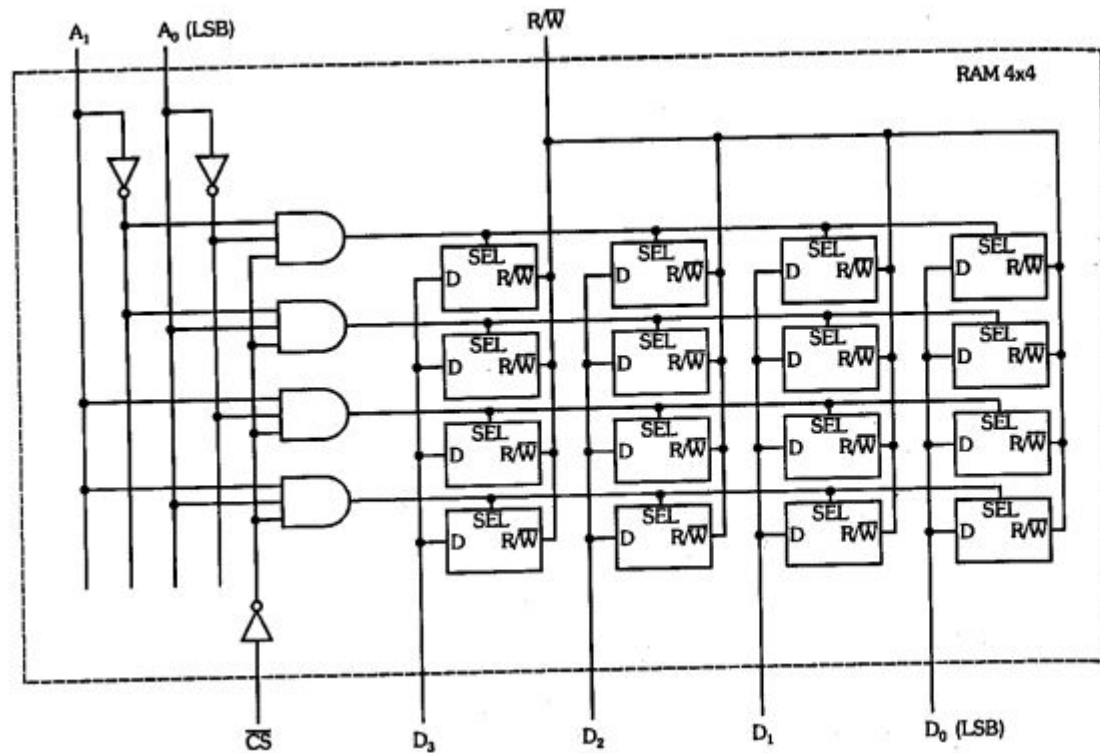


barramento de dados



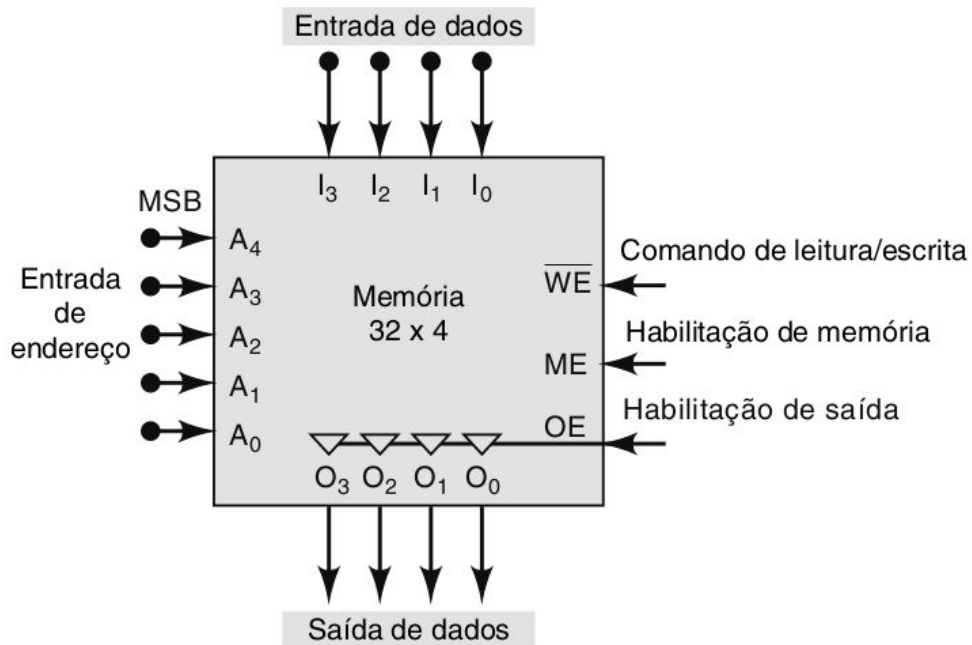
MEMÓRIAS

- Arquitetura interna de uma memória de **4 registradores de 4 bits**



MEMÓRIAS

- Exemplo de diagrama de uma **memória de 32 registradores de 4 bits**



Células de memória

				Endereços
0	1	1	0	0 0 0 0 0
1	0	0	1	0 0 0 0 1
1	1	1	1	0 0 0 1 0
1	0	0	0	0 0 0 1 1
0	0	0	1	0 0 1 0 0
0	0	0	0	0 0 1 0 1
⋮	⋮	⋮	⋮	⋮
1	1	0	1	1 1 1 0 1
1	1	0	1	1 1 1 1 0
0	1	1	1	1 1 1 1 1