Departamento de Engenharia Elétrica e de Computação - EESC-USP

SEL-0415 Introdução à Organização de Computadores

Aula 5 Parte 1 :

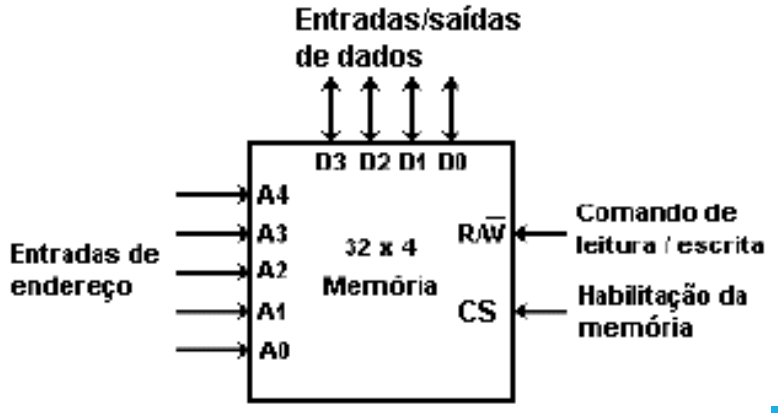
Princípio de Operação de memórias
Expansão de Memórias
Lógica de Seleção

Profa. Luiza Maria Romeiro Codá

Memórias Semicondutoras

PRINCÍPIOS DE OPERAÇÃO DAS MEMÓRIAS:

- Selecionar o endereço a ser acessado (leitura ou escrita);
- Se a operação for escrita, fornecer os dados de entrada;
- Se a operação for leitura, os dados estarão disponíveis na saída;
- Habilitar a memória ($\overline{CS}$) para que as portas de I/O sejam liberadas para a operação desejada;
- Selecionar o tipo de operação: leitura ou escrita (*R/W*);

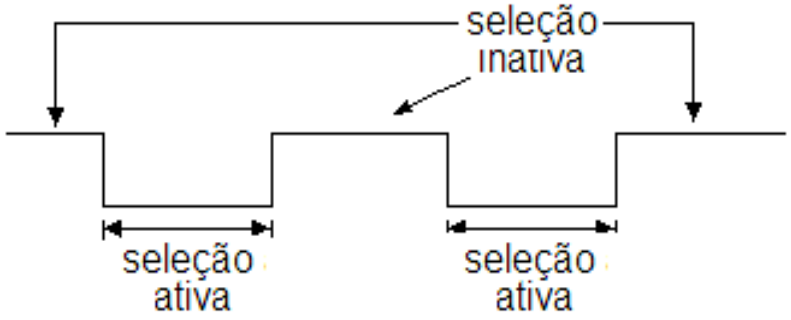


Sinais nos pinos de controle

Sinal de Habilitação:

$\overline{ME}$: *Memory Enable*
$\overline{E}$: *Enable*
$\overline{CS}$: *Chip Select*

É um sinal de seleção, ativo em "0" → seleciona o dispositivo.
Se colocado em nível "1" → desabilita o dispositvo , geralmente colocando em estado de alta-impedância (tristate).

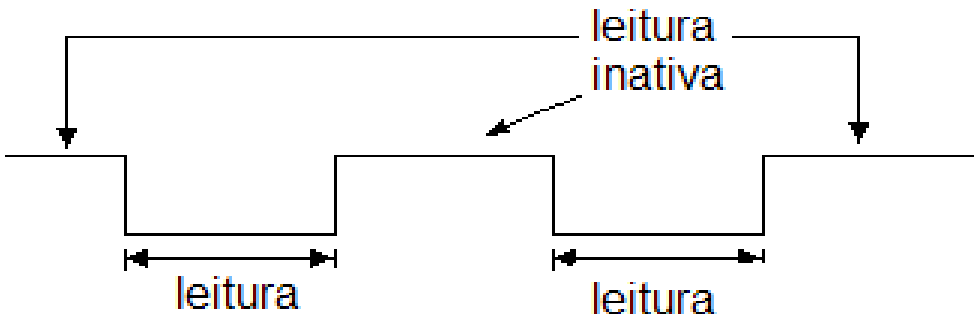


Sinais nos pinos de controle

Sinal de Leitura:

$\overline{R}$: *Read*

É um sinal de leitura, ativo em nível lógico “0” → Coloca o dado armazenado na memória, **na posição definida no duto de endereços**, no duto de dados.

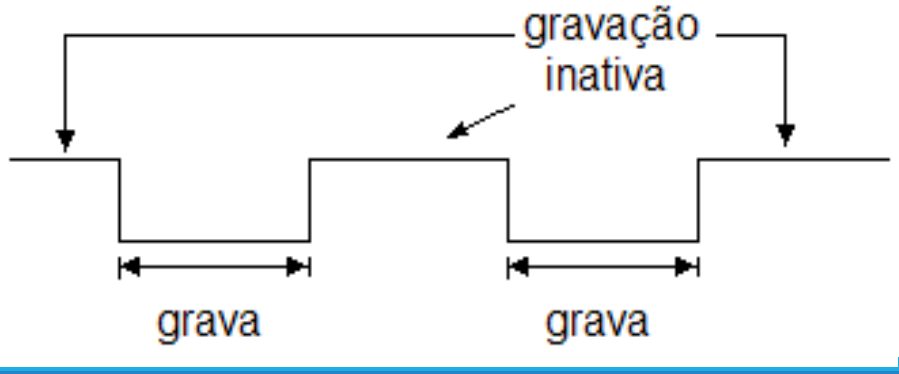


Sinais nos pinos de controle

Sinal de Escrita:

$\overline{W}$:*Write*

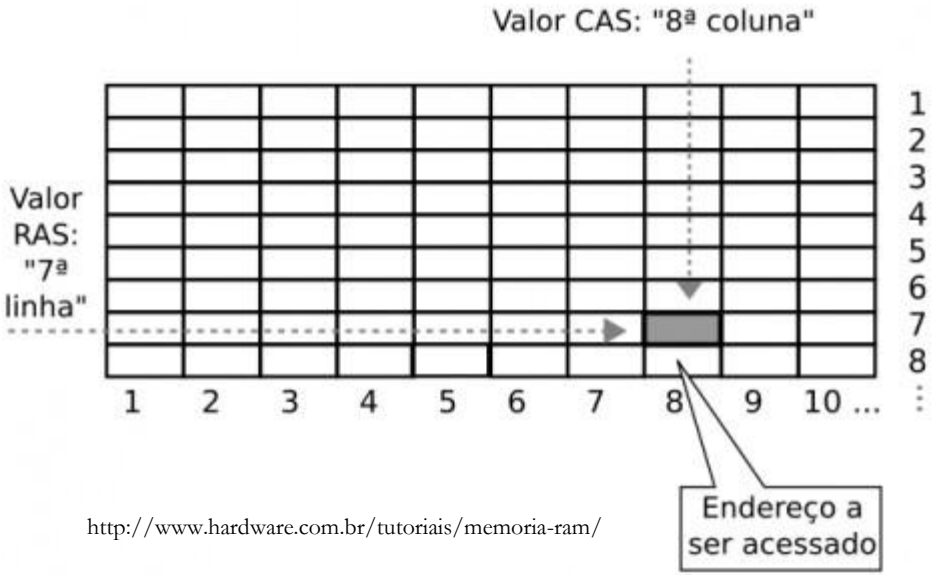
É um sinal de escrita (gravação), ativo em nível lógico “0” → Armazena o dado presente no duto de dados na posição de memória definida no duto de endereços.



Acesso à Memória

As células de memória são organizadas em uma espécie de matriz, ou seja, são orientadas em um esquema que lembra linhas(*wordline)* e colunas(*bitline*).

controlador de memória: acessa a memória gerando primeiro o valor **RAS** (*Row Address Strobe*) (nºda linha de qual o endereço faz parte), e depois do valor **CAS** (*Column Address Strobe)* da coluna **.**

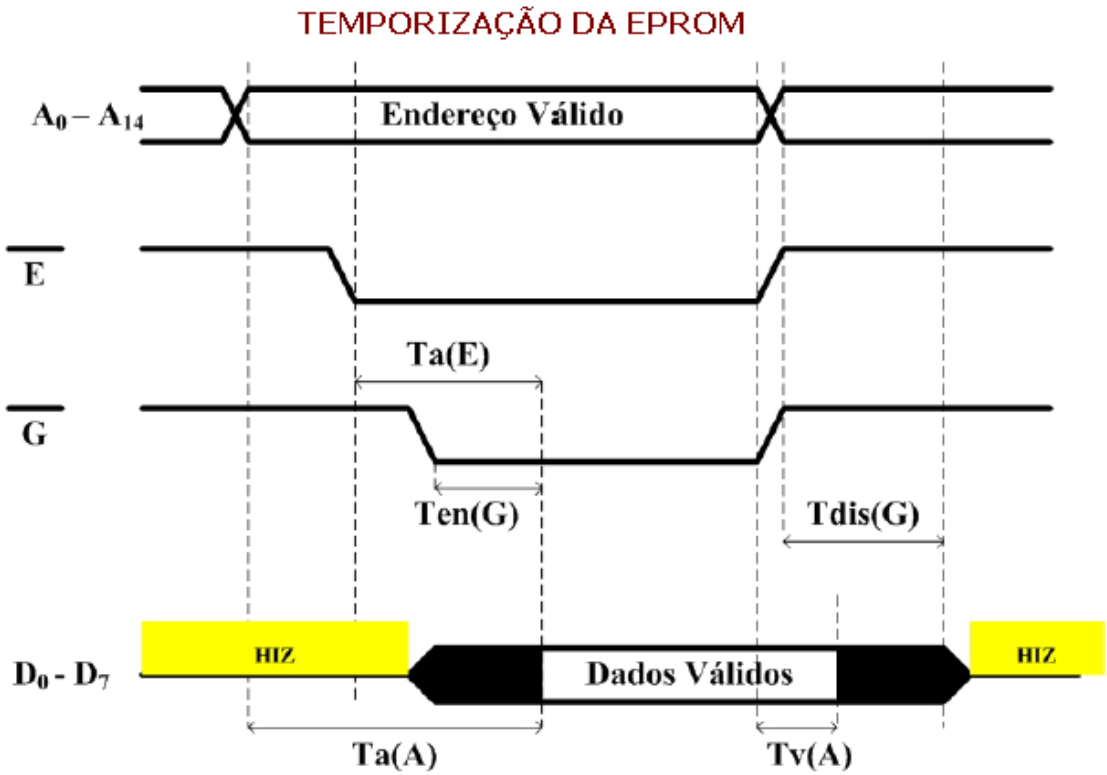


http://www.hardware.com.br/tutoriais/memoria-ram/

Tempos de Chaveamento das Memórias Semicondutoras

Ciclo de Leitura

Ta(A): Tempo de acesso após endereço válido.
Ta(E): Tempo de acesso após habilitação do chip.
Ten(G): Tempo de acesso após habilitação da saída (tristate).
Tv(A): Tempo em que os dados estão válidos após a mudança de endereço, de E ou de G.
Tdis(G): Tempo para desabilitar a saída após a mudança de endereço.



Obs: Existe um atraso de propagação entre a aplicação das entradas (endereços e controle e seleção) de uma ROM e a aparição das saídas de dados durante a operação de leitura.

Tempos de Chaveamento das Memórias Semicondutoras (continuação)

Ciclo de Leitura

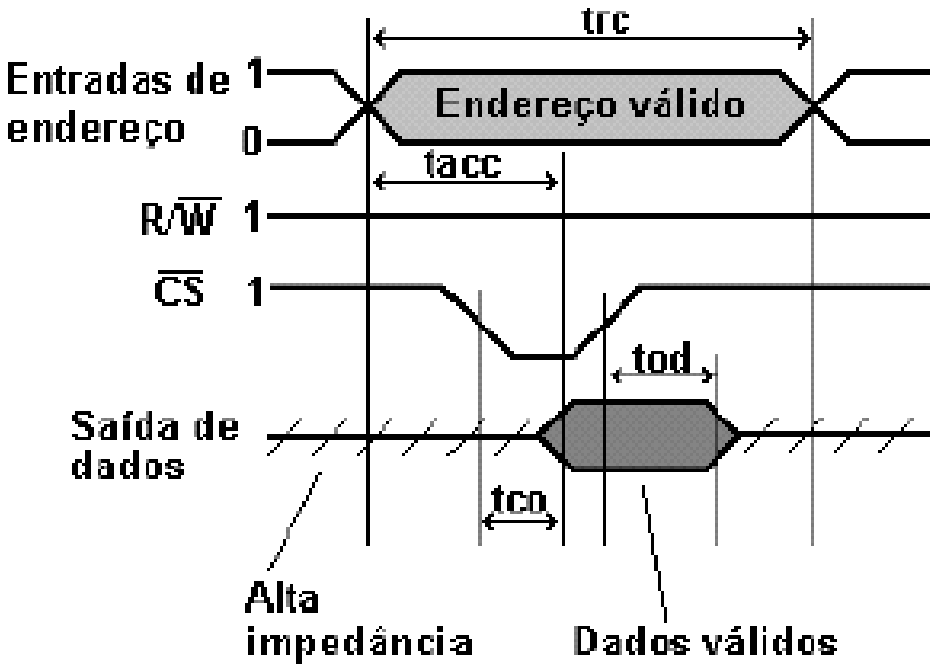
trc : intervalo de duração do ciclo de leitura;

tacc : tempo de acesso à RAM;

tco : tempo que a saída da RAM leva para sair de alta impedância e ter um dado válido;

tod : tempo decorrido entre a desabilitação da RAM e o instante que as saídas da RAM vão para alta impedância.

Temporização memória RAM



Tempos de Chaveamento das Memórias Semicondutoras (continuação)

Ciclo de Escrita (ou Gravação)

twc = intervalo de duração do ciclo de escrita;

tas = tempo para estabilização do duto de endereços, antes de habilitar a RAM;

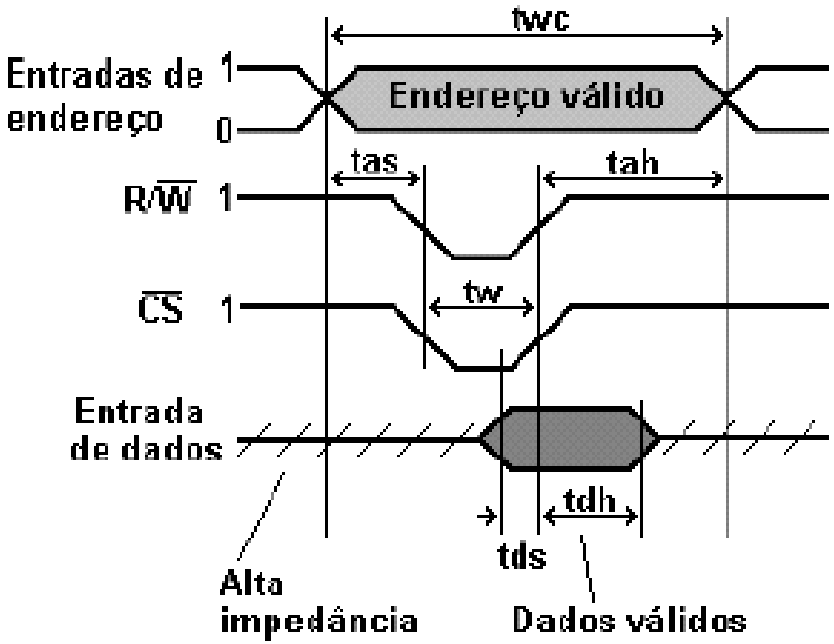
tah = intervalo necessário para que o duto de endereços permaneça estável;

tw = tempo de escrita, onde /CS e R/W ficam em “0”;

tds = tempo em que os dados devem ser mantidos na entrada, antes da desabilitação de CS e W / R ;

tdh = tempo em que os dados devem ser mantidos na entrada depois da desabilitação de CS e R /W .

Temporização memória RAM



Expansão de Memórias Semicondutoras

Expansão de Memórias

a) Aumentar o número de bits da palavra:

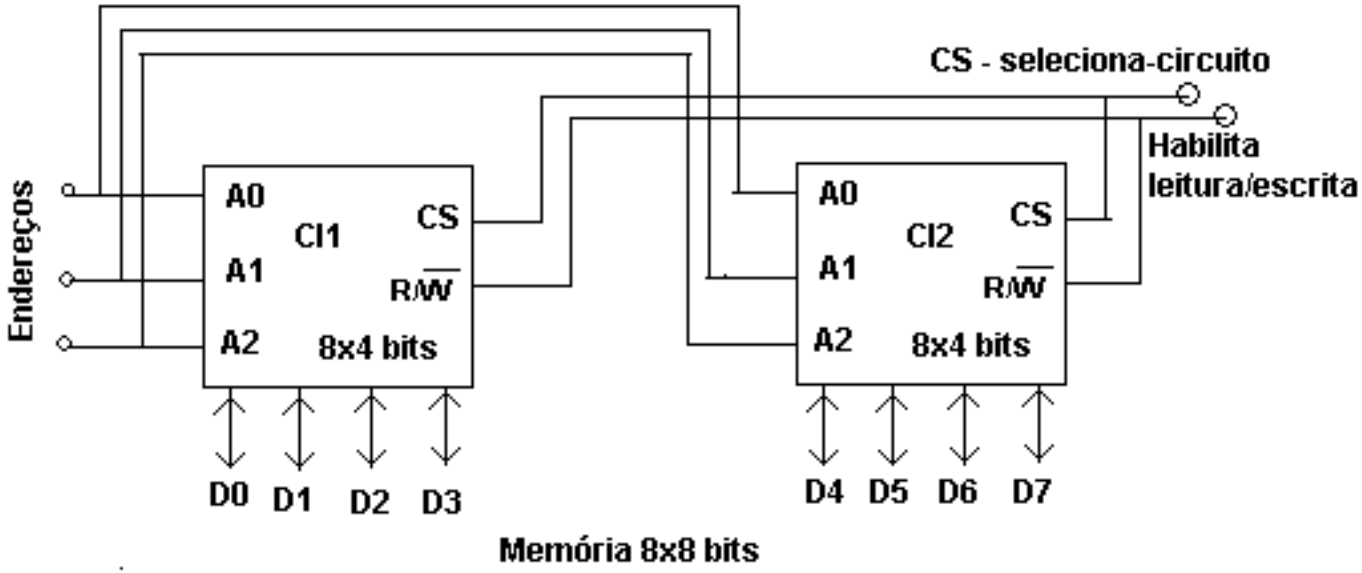
* Exemplo:
  - Organização desejada: **2K x 8** (EPROM ou RAM)
  - Memória disponível: **2K x 4**

b) Aumentar o número de palavras (endereços):

* Exemplo:
  - Organização desejada: **4K x 8** (EPROM ou RAM)
  - Memória disponível: **2K x 8**

Aumentar o Tamanho (nº de bits) da Palavra

Dois CI's com 3 linhas de endereços → 2^3 = 8 endereços. Cada endereço aponta para uma palavras de 4 bits (8x4). Os dois Cis são ligados de modo a formar uma memória de 8 palavras de 8 bits (8 x8)

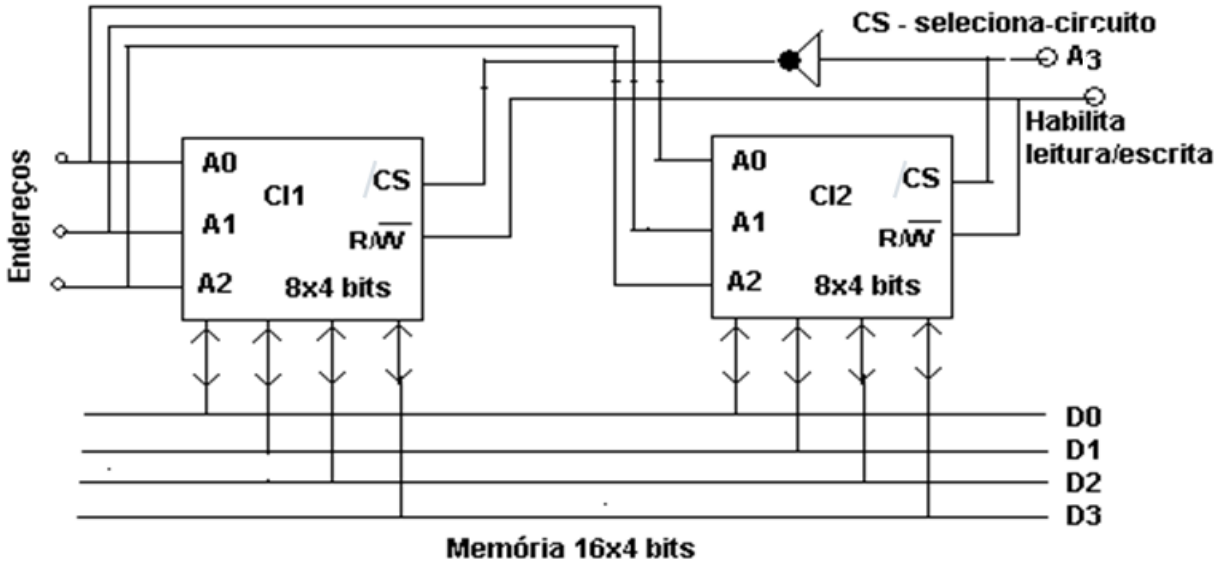


- O duto de endereços e os pinos de controle dos CIs são interligados;
- O duto de dados fica dividido entre os CIs , de forma que cada CI contribui com uma parcela do dado : 4 MSB no CI2 - pino D7
  4 LSB no CI1 - pino D0

Aumentar o Nº de Palavras(células)

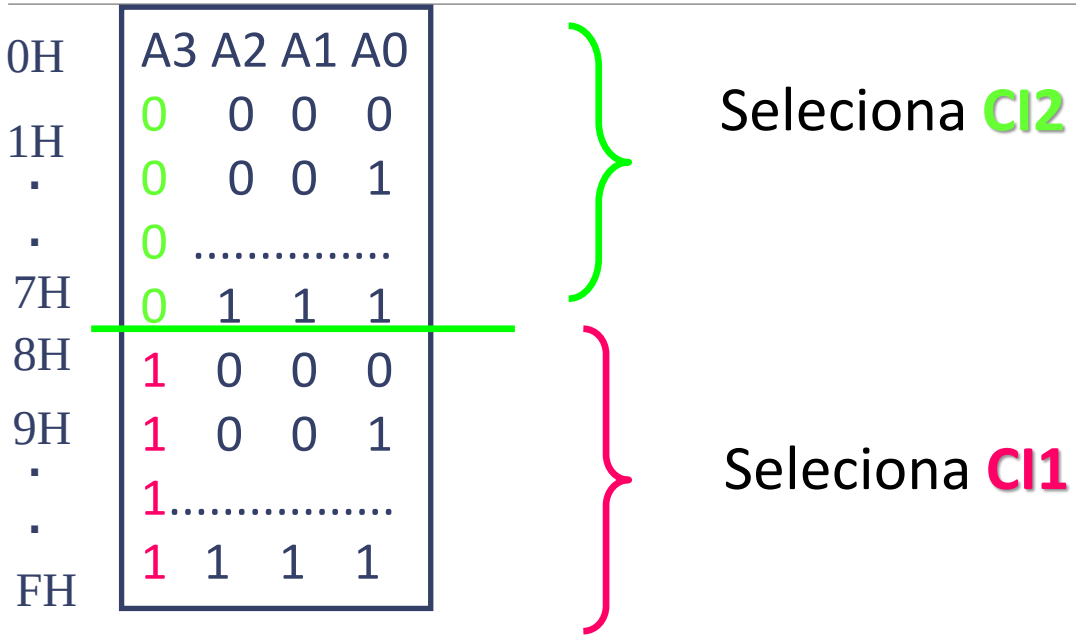
Dois CI's com 8 palavras de 4 bits cada (8x4), ligados de modo a formar uma memória de 16 palavras(células) de 4 bits (16x4)

A3



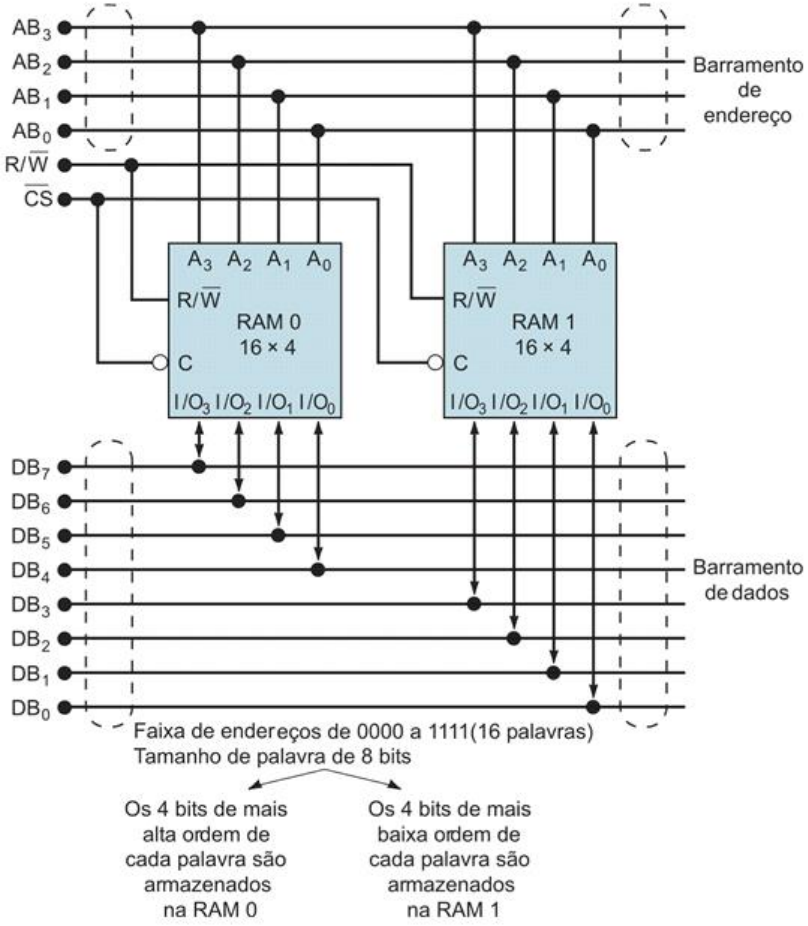
- O duto de dados, endereços e o pino de R/W dos CIs são interligados;
- O pino de controle (/CS) dos CIs **não** recebem o mesmo sinal;
- a seleção de CI1 e CI2 é feita através da linha de endereço A3, como segue:
  - Pino de endereço A3 = 1 - Seleciona o CI1 (8 end. mais signif.)
  - Pino de endereço A3 = 0 - Seleciona o CI2 (8 end. menos signif.)

Aumentar o Nº de Palavras (continuação)

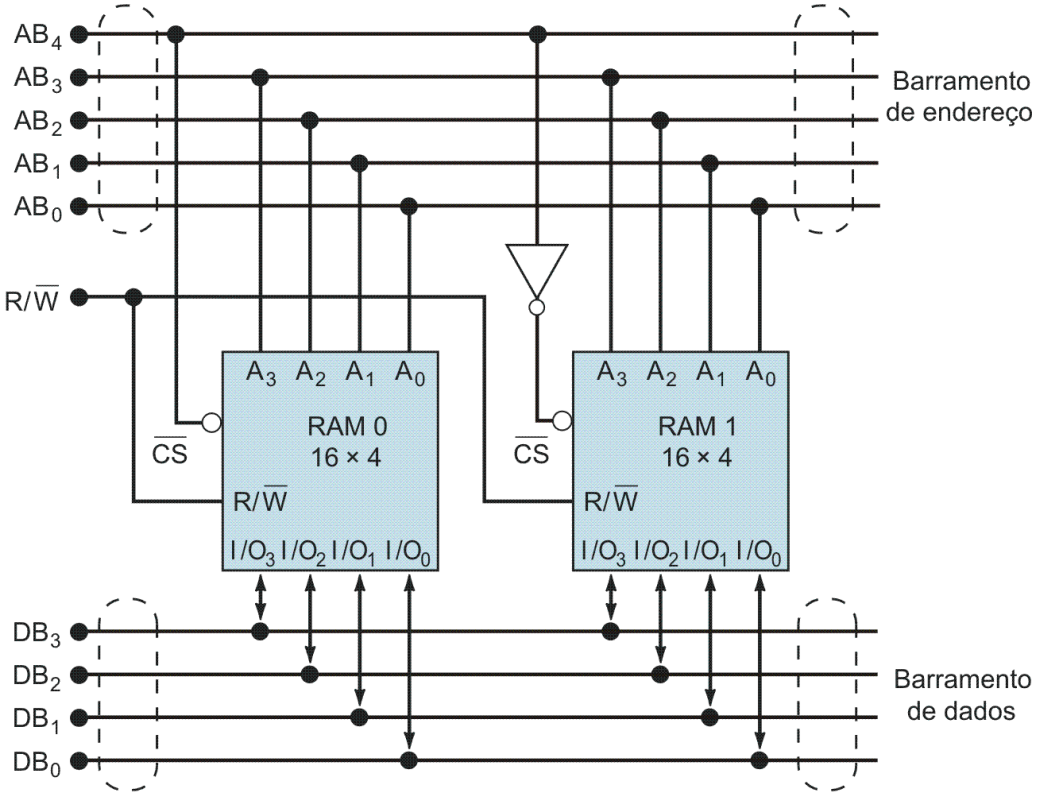


- O duto de **dados, endereços** e o pino de R/W dos CIs são interligados;
- O pino de controle (/CS) dos CIs não recebem o mesmo sinal;
- a seleção de CI1 e CI2 é feita através da linha de endereço A3, como segue:
  - Pino de endereço **A3** = 1 - Seleciona o CI1 (8 end. mais signif.)
  - Pino de endereço A3 = 0 - Seleciona o CI2 (8 end. menos signif.)

Duas RAMs de 16 X 4 em um módulo de 16 X 8



Duas RAMs de 16 X 4 em um módulo de 32 X 4



Faixas de endereço:	00000 a 01111 – RAM 0 10000 a 11111 –RAM 1
Total	00000 a 11111 – (32 palavras)

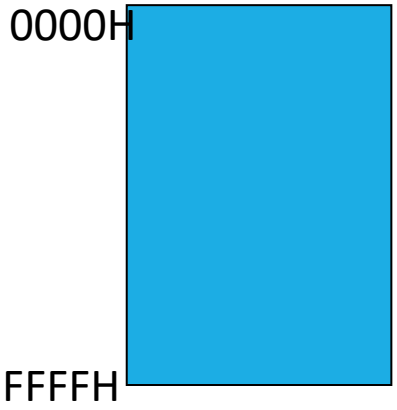
Abreviaturas usadas em Computação

Nome da unidade	Símbolo	Valor em potência de 2	Valor em Unidades
Quilo	K	2^{10}	1024
Mega	M	1024 K = 2^{20}	1 048 576
Giga	G	1024M = 2^{30}	1 073 741 824
Tera	T	2^{40}	1 099 511 627 776
Peta	P	2^{50}	1 125 899 906 843 624
Exa	Ex	2^{60}	1 152 921 504 607 870 976
Zeta	Z	2^{70}	1 180 591 620 718 458 879 424
Yotta	Y	2^{80}	1 208 925 819 615 701 892 530 176

Lógica de Seleção de Memória e Dispositivos de I/O (mapeamento)

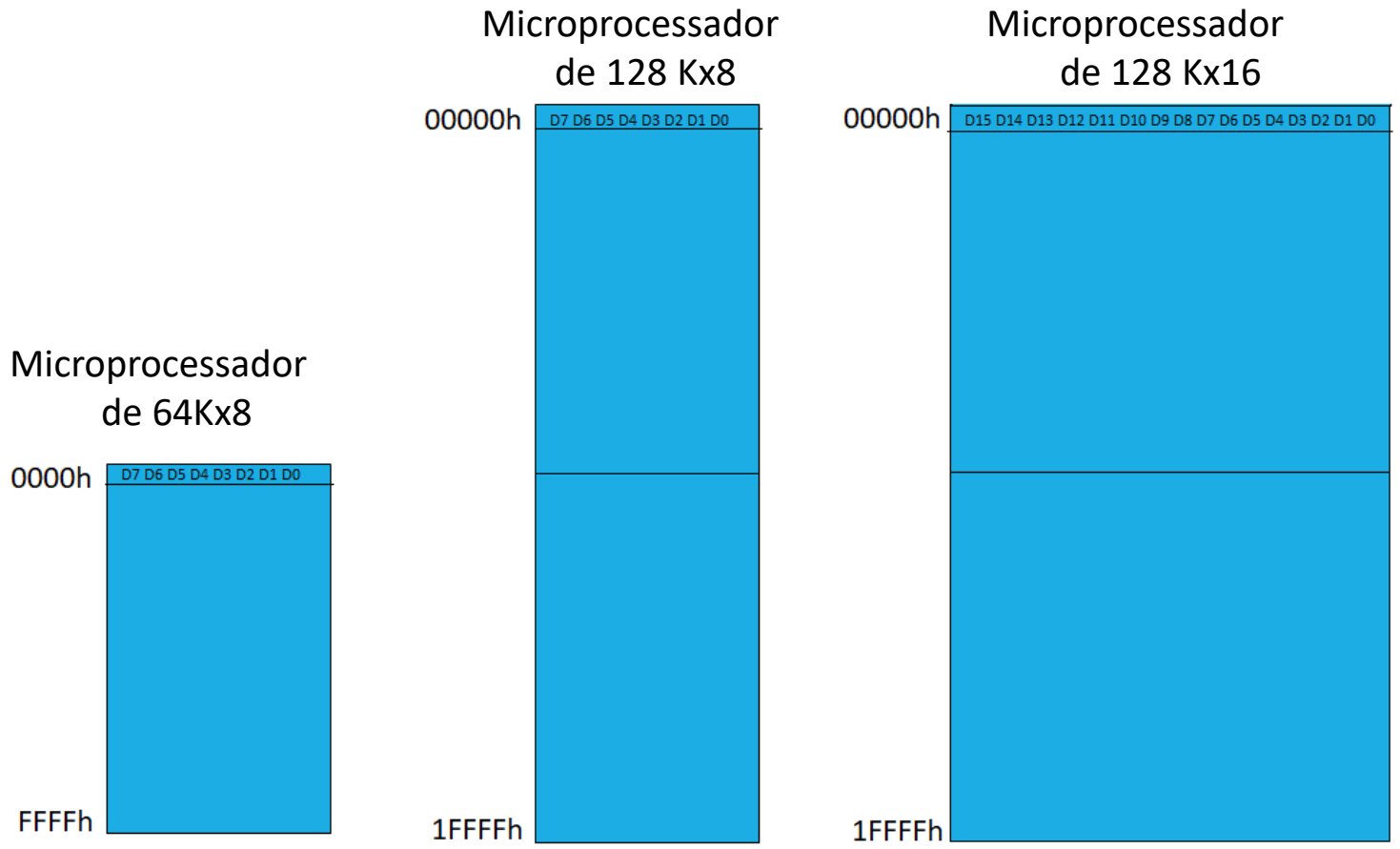
Seleção de Memórias e Dispositivos de I/O

- Um microprocessador que tem duto de endereços de 16 bits e duto de dados de 8 bits, consegue endereçar 2^{16} = 65536 (ou 64K) bytes
- As 64K posições que o microprocessador consegue endereçar podem ser representadas graficamente por um retângulo dividido em 64K posições, que é denominado **espaço de endereços do microprocessador.**



Seleção de Memórias e Dispositivos de I/O

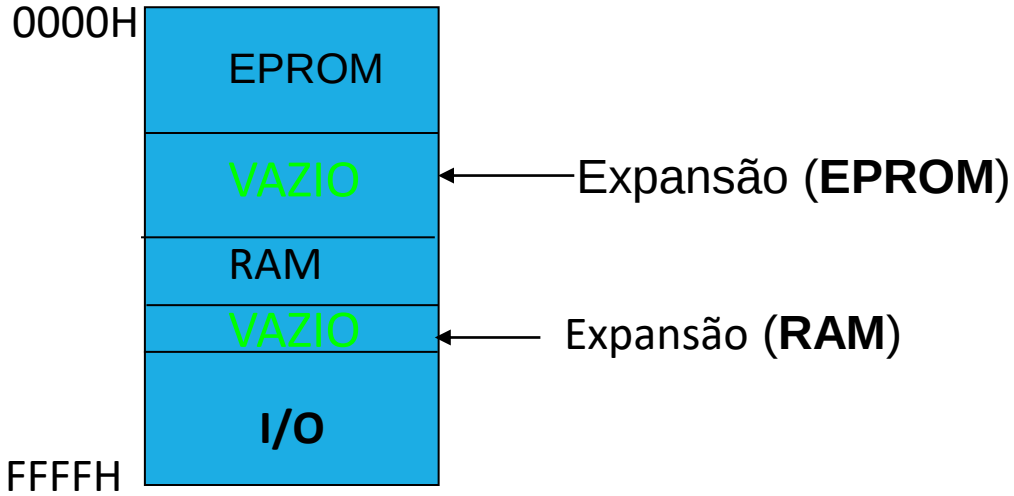
Representação do espaço de endereçamento do microprocessador:



Seleção de Memórias e Dispositivos de I/O

Dentro do **espaço de endereços** de 64K bytes que o microprocessador consegue endereçar, são mapeadas (alocadas) as memórias e os dispositivos de I/O

A "Lógica de Seleção", construída pelo projetista, define as faixas de endereços do microprocessador que irão selecionar cada uma das memórias e dispositivos

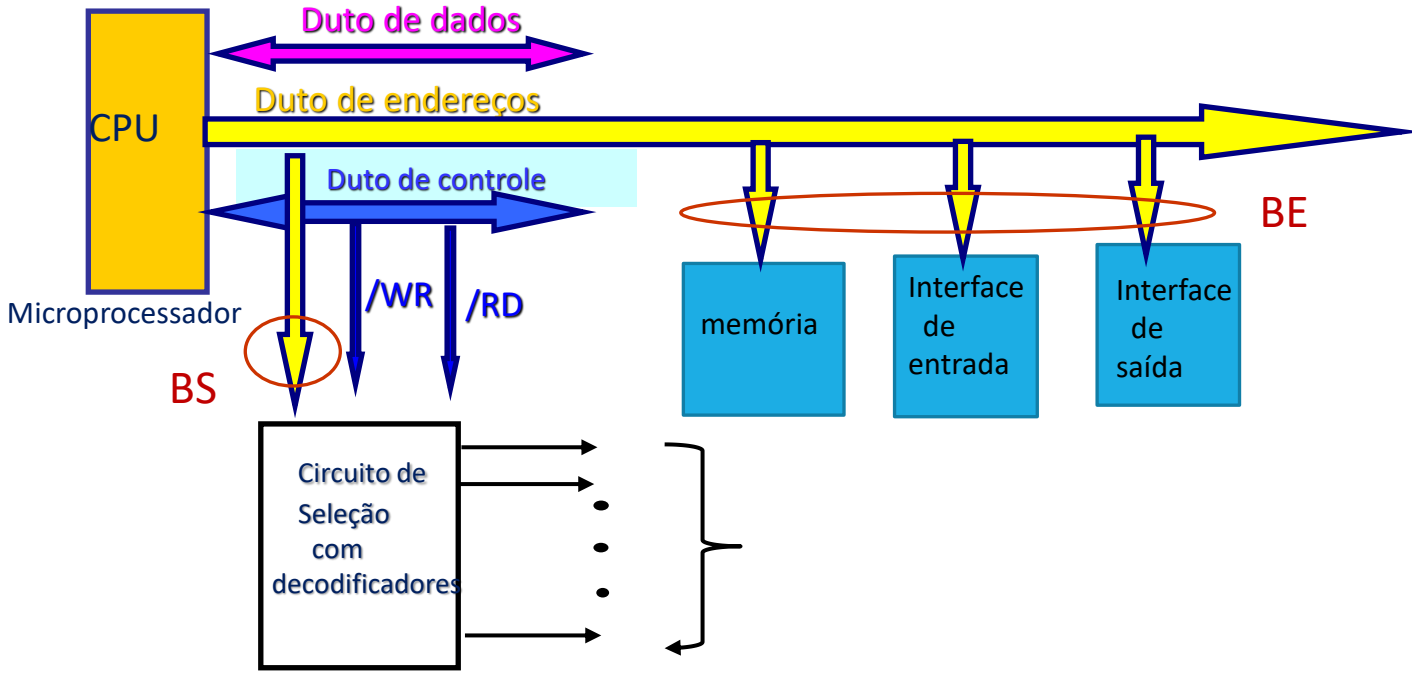


Seleção de Memórias e Dispositivos de I/O

O endereço de **16 bits**, gerado pelo **microprocessador** pode ser visto como sendo constituído por duas partes :

BE: Bits de endereçamento do chip

BS: Bits de seleção

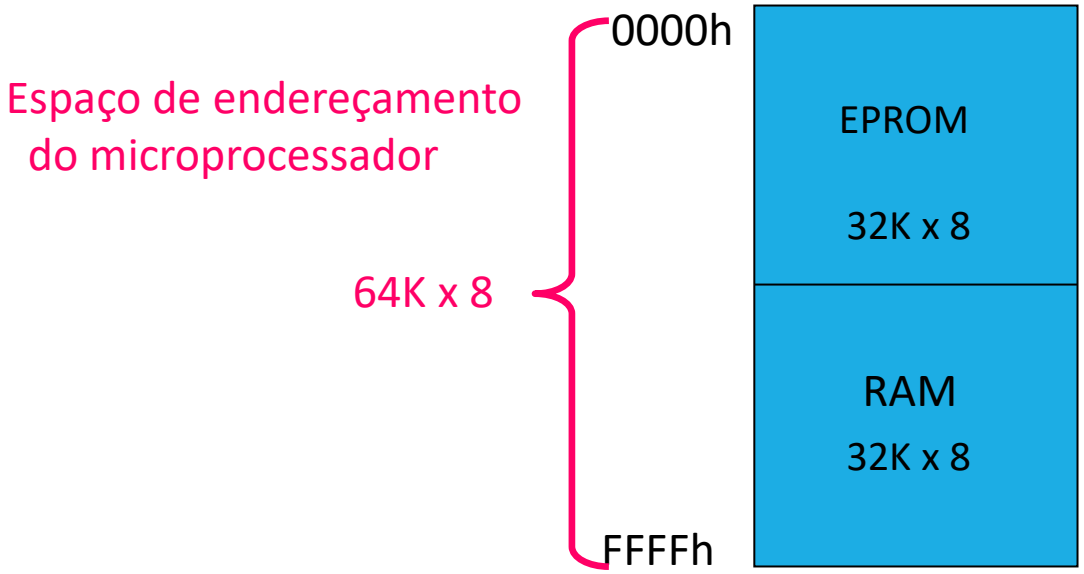


Seleção de Memórias e Dispositivos de I/O

Tabela 1 Bits de seleção (BS) e de endereçamento (BE) para diferentes organizações ligadas a um microprocessador de 16 linhas de endereços e 8 bits de duto de dados

organizações	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
64Kx8	Bits de Endereçamento =BE															
32Kx8	BS	BE														
16Kx8	BS		BE													
8Kx8	BS			BE												
4Kx8	BS				BE											
2Kx8	BS					BE										
1Kx8	BS						BE									
512x8	BS							BE								
8x8	BS															BE
4x8	BS														BE	
2x8	BS													BE		
1x8	Bits de Seleção = BS															

Seleção de Memórias e Dispositivos de I/O

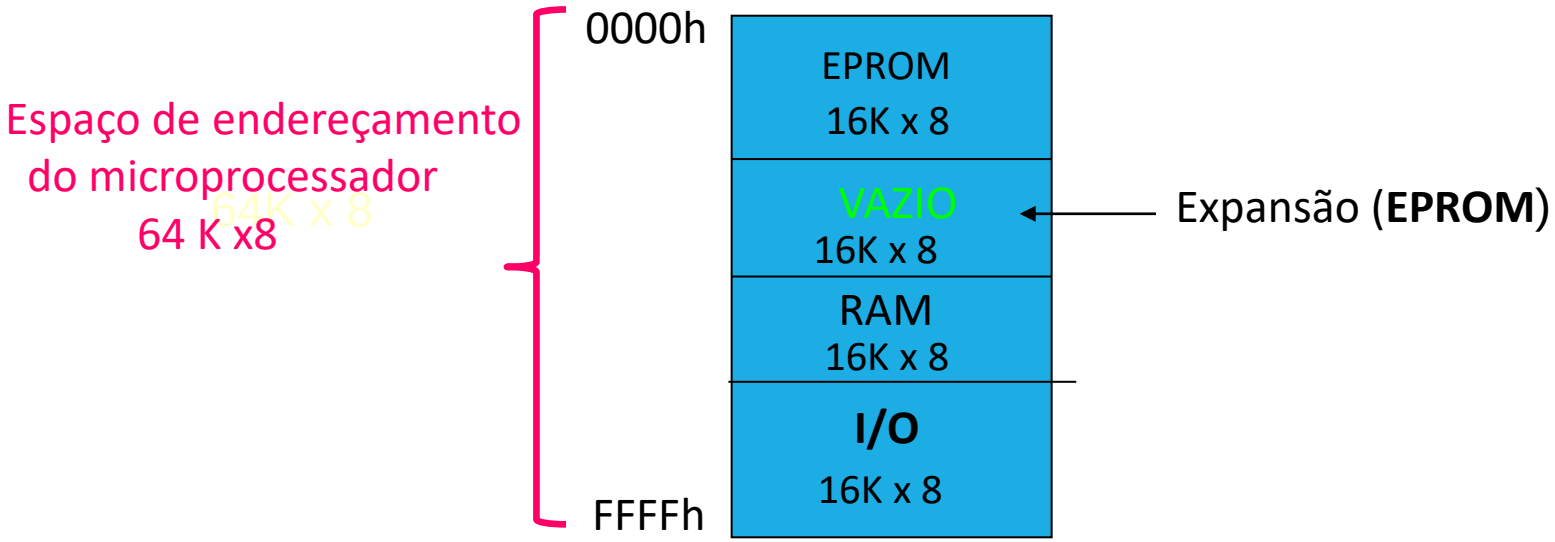


O endereço de **16 bits**, gerado pelo **microprocessador para endereçar** cada memória de 32Kx8, pode ser visto como sendo constituído por duas partes:

BE: Bits de endereçamento dos chips de memória de 16Kx8 de A_{14} a A_0

BS: Bit de seleção dos chips de memória de 32Kx8 é A_{15}

Seleção de Memórias e Dispositivos de I/O



O endereço de **16 bits**, gerado pelo **microprocessador para endereçar** cada memória de 16K x 8 e dispositivos de I/O de 16Kx8, pode ser visto como sendo constituído por duas partes:

BE: Bits de endereçamento dos chip de memória e dispositivos de I/O de 16Kx8 de A_{13} a A_0

BS: Bits de seleção chip de memória e dispositivos de I/O de 16Kx8 são A_{15} e A_{14}

Seleção de Memórias e Dispositivos de I/O

Cálculo do Número de posições de memória (em hexadecimal) para cada organização:

Fazendo-se os bits de seleção (**BS)** iguais a **zero**, e bits de endereçamento (**BE**), iguais a **"1"**, na tabela **1**, é possível determinar o **(número de posições – 1)** ocupadas por cada organização:

32k x 8 : 7FFFh	**2k x 8 : 07FFh**	**8 x 8 : 0007h**
16k x 8 : 3FFFh	**1k x 8 : 03FFh**	**4 x 8: 0003h**
8k x 8 : 1FFFh	**512 x 8 : 01FFh**	**2 x 8: 0001h**
4k x 8 : 0FFFh	**:**	**1 x 8 : 0000h**

Esse valor é somado ao endereço inicial, no espaço de endereçamento, para se obter o endereço final da memória

Relação entre os blocos de memórias e as linhas de endereço de um microprocessador de 16 linhas (bits) de endereço

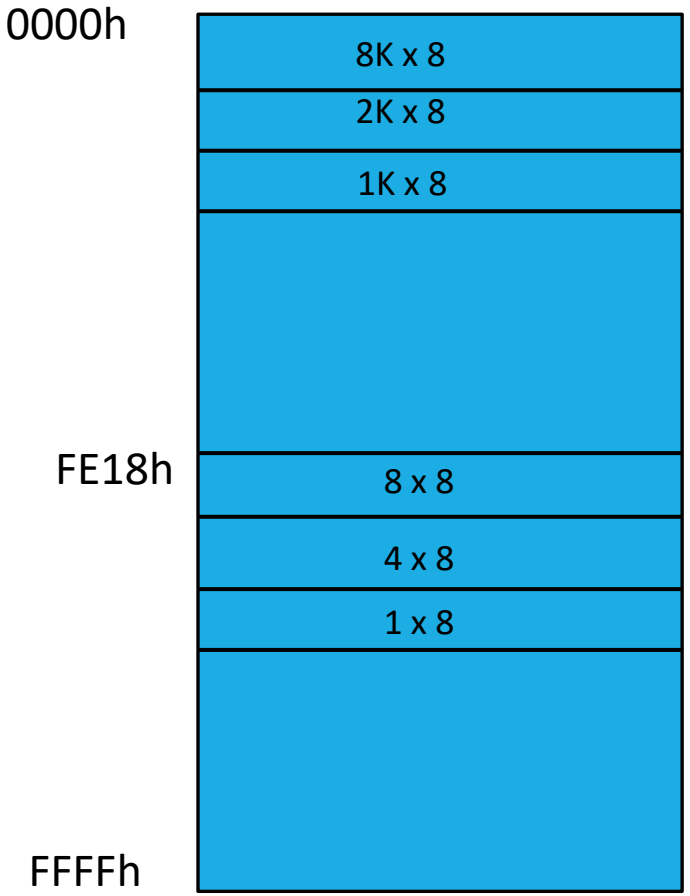
Blocos de memória	Nº linhas de endereço	Bits de seleção (BS)	Bits de endereço (BE)	Tamanho do bloco
32K x 8	15	A15	A0 a A14	7FFFh
16K 8	14	A14 a A15	A0 a A13	3FFFh
8K x 8	13	A13 a A15	A0 a A12	1FFFh
4K x 8	12	A12 a A15	A0 a A11	0FFFh
2K x 8	11	A11 a A15	A0 a A10	07FFh
1K x 8	10	A10 a A15	A0 a A9	03FFh
512 x 8	9	A9 a A15	A0 a A8	01FFh
256 x 8	8	A8 a A15	A0 a A7	00FFh
128 x 8	7	A7 a A15	A0 a A6	007Fh
64 x 8	6	A6 a A15	A0 a A5	003Fh
32 x 8	5	A5 a A15	A0 a A4	001Fh
16 x 8	4	A4 a A15	A0 a A3	000Fh
8 x 8	3	A3 a A15	A0 a A2	0007h
4 x 8	2	A2 a A15	A0 a A1	0003h
2 x 8	1	A1 a A15	A0	0001h
1 x 8	0	A0 a A15	Nenhum	0000h

Cálculo do endereço final a partir do tamanho do bloco

Exemplo: considerando endereço inicial = 8000h		
tamanho do bloco	**endereço Inicial**	**endereço Final**
32Kx8	8000h	FFFFh
8KX8	8000h	9FFFh
1KX8	8000h	83FFh
8X8	8000h	8007h
4X8	8000h	8003h
2X8	8000h	8001h
1X8	8000h	8000h

Lógica de Seleção de Memórias e Dispositivos de I/O

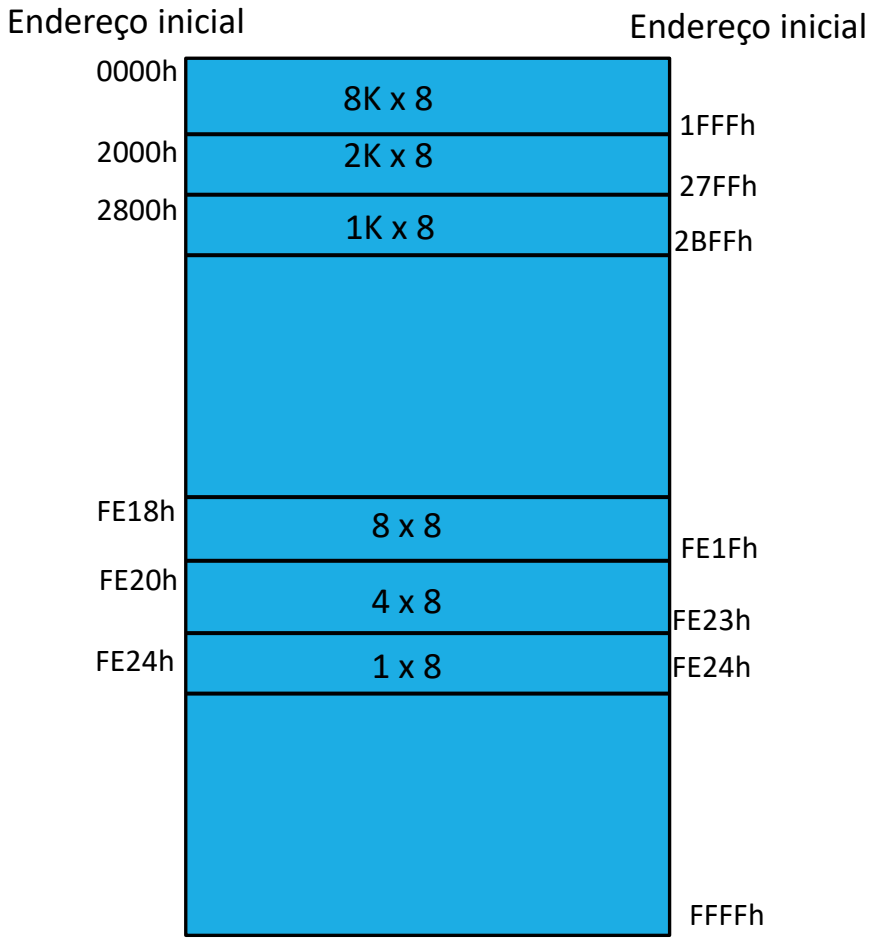
Exercício: Encontrar o endereço inicial e final, para as organizações de memória e de I/O mapeadas na Figura ao lado



Lógica de Seleção de Memórias e Dispositivos de I/O

Resposta:

Exercício: Encontrar o endereço inicial e final, para as organizações de memória e de I/O mapeadas na Figura ao lado



Lógica de Seleção de Memórias e Dispositivos de I/O

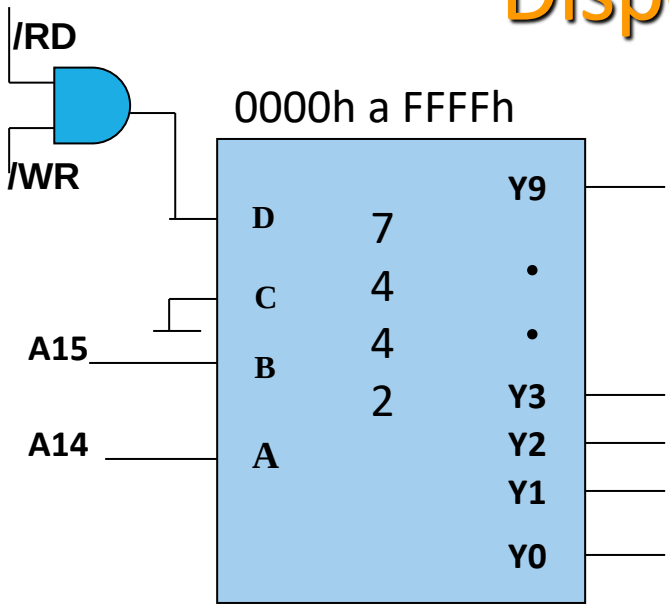
A lógica de seleção implementada com circuitos decodificadores garante a seleção de **uma única memória ou interface**, que se comunicará com o microprocessador.

- Cada decodificador fica "dentro" de um espaço de endereço, e divide esse espaço em blocos menores.

- O tamanho da divisão depende de qual é o **bit de seleção** menos significativo conectado na entrada do decodificador (ver tabela 1) .

- **Exemplo:** Microprocessador com 16 linhas de endereços (A15-A0) e 8 linhas de dados. ⇨ Espaço de mapeamento de 64K x 8.
  Se no decodificador entram **2** linhas de endereços (A15 e A14),
  esse decodificador divide o espaço total por 4 (= 2^2), espaços de 16K x 8,
  ou seja, espaços que são endereçados por A13-A0

Lógica de Seleção de Memórias e Dispositivos de I/O



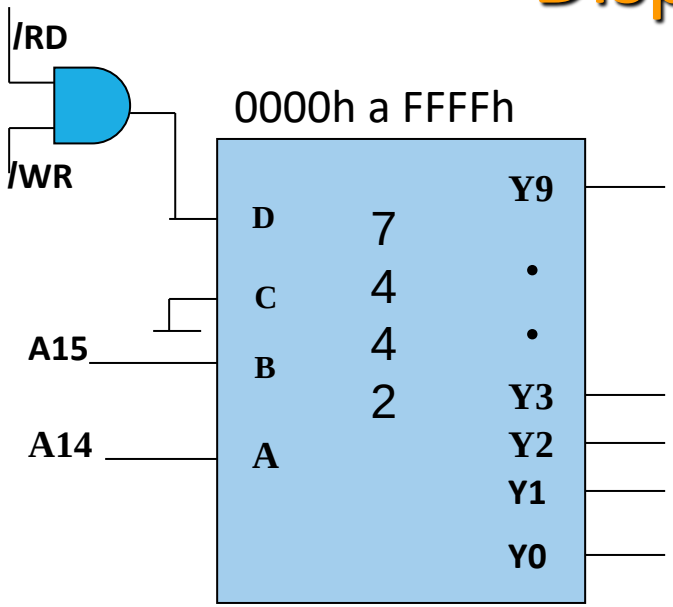
- Espaço de endereço: **0000 a FFFFh**

(o decodificador não é selecionado por nenhum outro decodificador)

- tamanho da divisão: **16 Kx8**, pois o bit menos significativo é o **A14**
- pode-se **conectar direto** nesse decodificador organizações que tem linhas de endereço de A0 - A13

Quais saídas do decodificador que podem ser ativadas ?
Qual a faixa de endereço associada a cada saída?

Lógica de Seleção de Memórias e Dispositivos de I/O



Divisão: 16 Kx8
(3FFFh)

- Espaço de endereço: **0000 a FFFFh**

(o decodificador não é selecionado por nenhum outro decodificador)

- tamanho da divisão: **16 Kx8**, pois o bit menos significativo é o **A14**
- pode-se **conectar direto** nesse decodificador organizações que tem linhas de endereço de A0 - A13

Resposta:

Saídas válidas para serem usadas como /CS:	**faixa de endereço**
Y0 : (A15, A14) = (0,0)	0000H até 3FFFh
Y1 : (A15, A14) = (0,1)	4000H até 7FFFh
Y2 : (A15, A14) = (1,0)	8000H até BFFFh
Y3 : (A15, A14) = (1,1)	C000H até FFFFh

Lógica de Seleção de Memórias e Dispositivos de I/O

Cada saída **válida** do decodificador, que pode ser usada como saída de seleção **(/CS**), tem a ela associada uma **faixa de endereço**, determinada pelos bits das linhas de endereço do microprocessador conectados nas entradas deste decodificador (bits de seleção BS).

Há duas maneiras de se determinar a **faixa de endereço da saída**:

a . Soma do bloco divisor ao endereço inicial de cada saída válida do decodificador

b. determina-se o endereço inicial e final associado à saída do decodificador como segue:

- **endereço inicial :** valor dos bits de seleção conectados no decodificador, que ativa a saída. Os demais bits em "**0**".
- **endereço final:** valor dos bits de seleção conectados no decodificador, que ativa a saída. Os demais bits em "**1**"

Lógica de Seleção de Memórias e Dispositivos de I/O

Utilizando o mesmo exemplo do decodificador anterior:

a . Soma do bloco divisor ao endereço inicial de cada saída

tamanho do bloco divisor = **3FFFh** (16 K x 8)

Y0 = 0000h a 3FFFh Y1 = 4000h a 7FFFH

Y2 = 8000h a BFFFh Y3 = C000h a FFFFh

b. Valor dos bits de seleção (A15 e A14)

		A15 A14 ..... A0
Saída Y0	endereço inicial: 0000h	**00**00 0000 0000 0000
	endereço final : 3FFFh	**00**11 1111 1111 1111
Saída Y1	endereço inicial: 4000h	**01**00 0000 0000 0000
	endereço final: 7FFFh:	**01**11 1111 1111 1111
Saída Y2	endereço inicial: 8000h	**10**00 0000 0000 0000
	endereço final : BFFFh	**10**11 1111 1111 1111
Saída Y3	endereço inicial: C000h	**11**00 0000 0000 0000
	endereço final : DFFFh	**11**11 1111 1111 1111

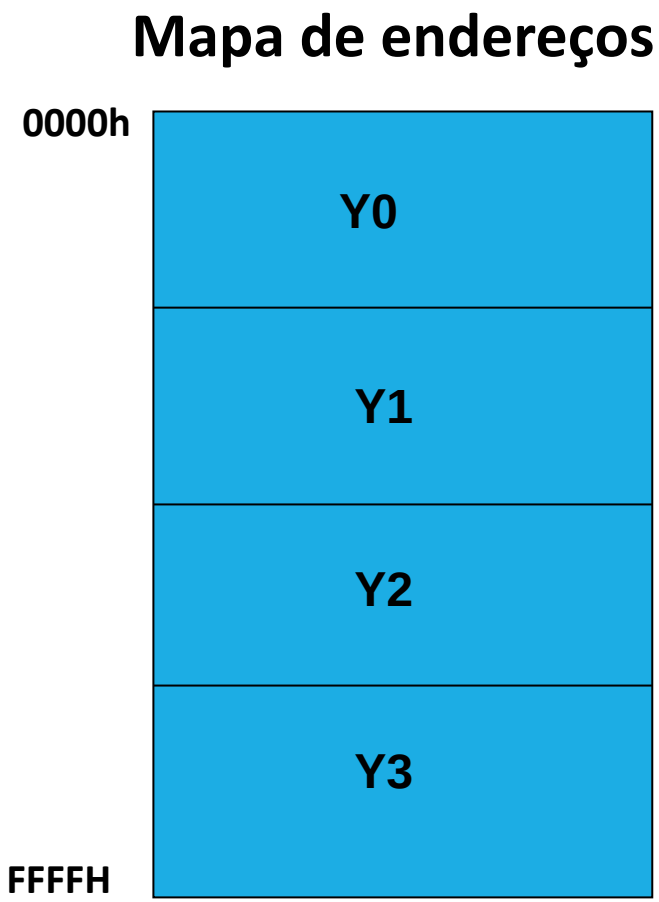
Lógica de Seleção de Memórias e Dispositivos de I/O

Dado o mapa de endereço ao lado:

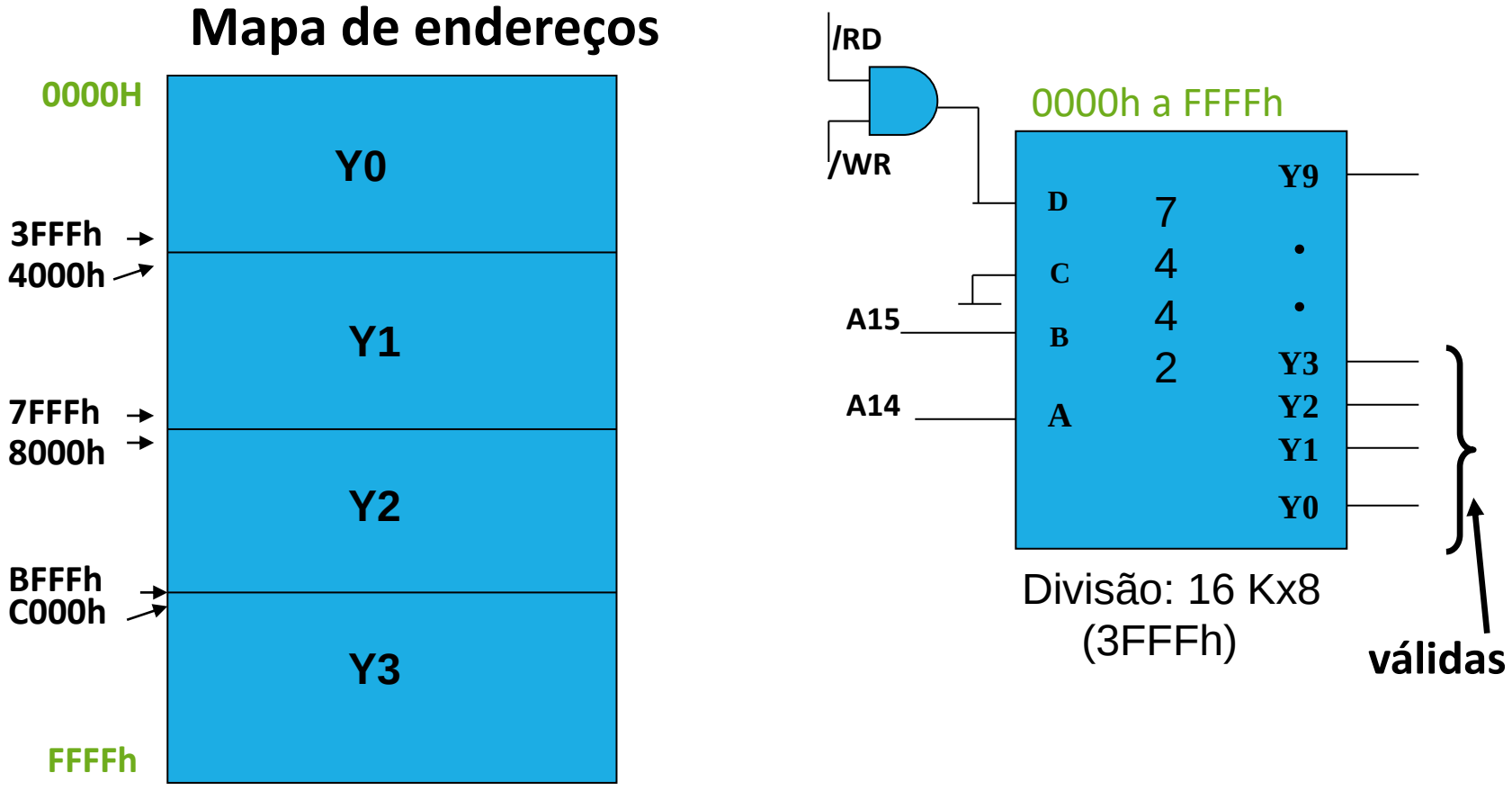
a. como seria a ligação do decodificador(7442) para obter a divisão do espaço de endereços de 64Kbytes em 4 espaços iguais?

b. Qual o tamanho de cada bloco em hexadecimal?

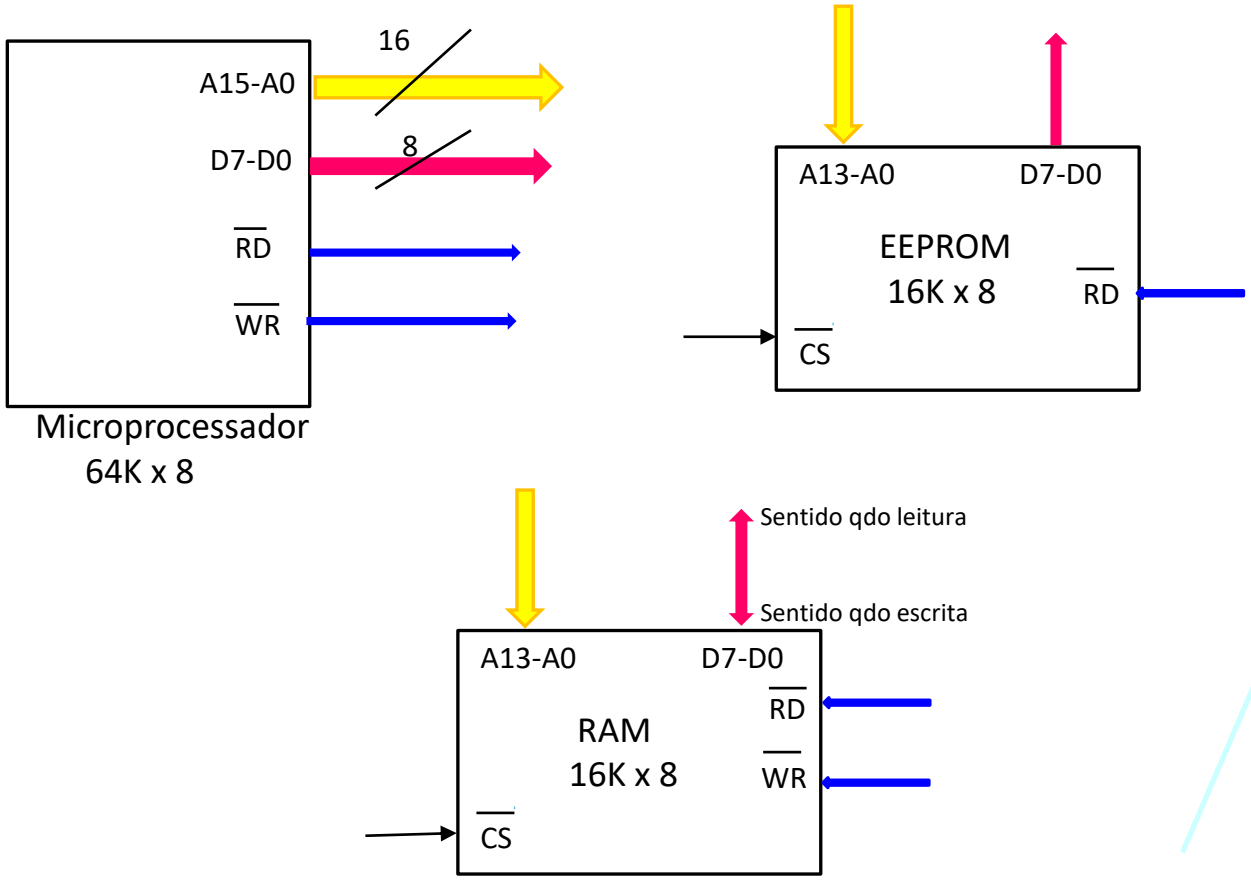
c. Qual o endereço inicial e final em hexadecimal de cada bloco?



Lógica de Seleção de Memórias e Dispositivos de I/O

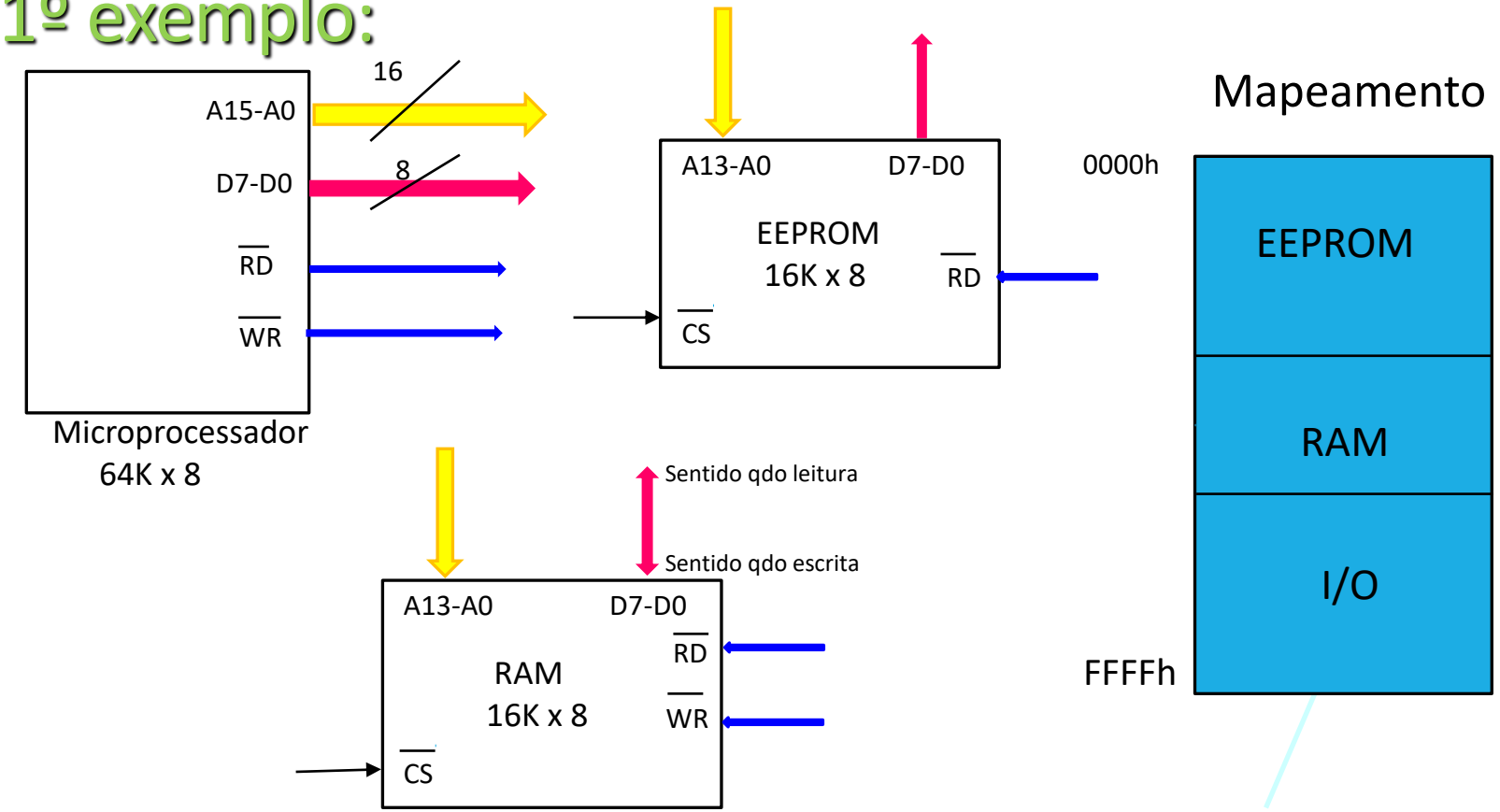


Como ligar o microprocessador às Memórias?



Como ligar o microprocessador às Memórias?

1º exemplo:



Lógica de Seleção de Memórias e Dispositivos de I/O

1º exemplo:

Lógica de Seleção do µP – Linhas de Endereços																	Memória		
Tipo	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Início (h)		Fim (h)
ROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	16k	
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1			3FFF
RAM																			
I/0																			

Lógica de Seleção de Memórias e Dispositivos de I/O

1º exemplo:

Lógica de Seleção do µP – Linhas de Endereços																	Memória		
Tipo	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Início (h)		Fim (h)
ROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	16k	
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1			3FFF
RAM	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000	16k	
	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			7FFF
I/0																			

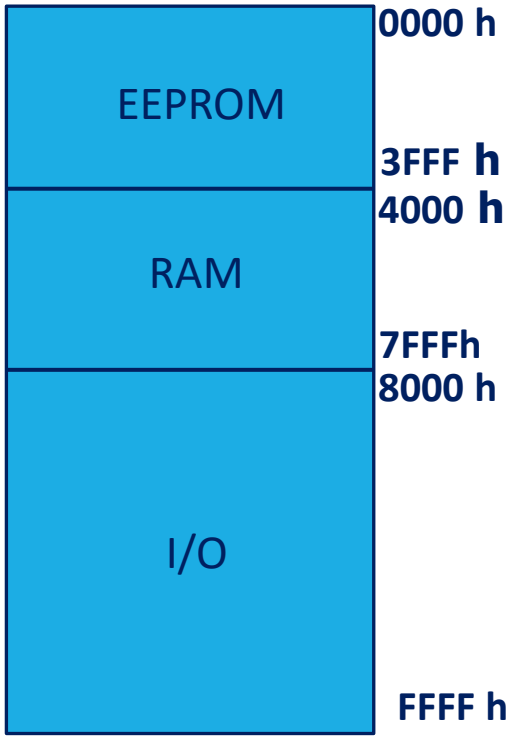
Lógica de Seleção de Memórias e Dispositivos de I/O

1º exemplo:

Lógica de Seleção do µP – Linhas de Endereços																	Memória		
Tipo	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Início (h)		Fim (h)
ROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	16k	
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1			3FFF
RAM	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000	16k	
	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			7FFF
I/0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000	32k	
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			FFFF

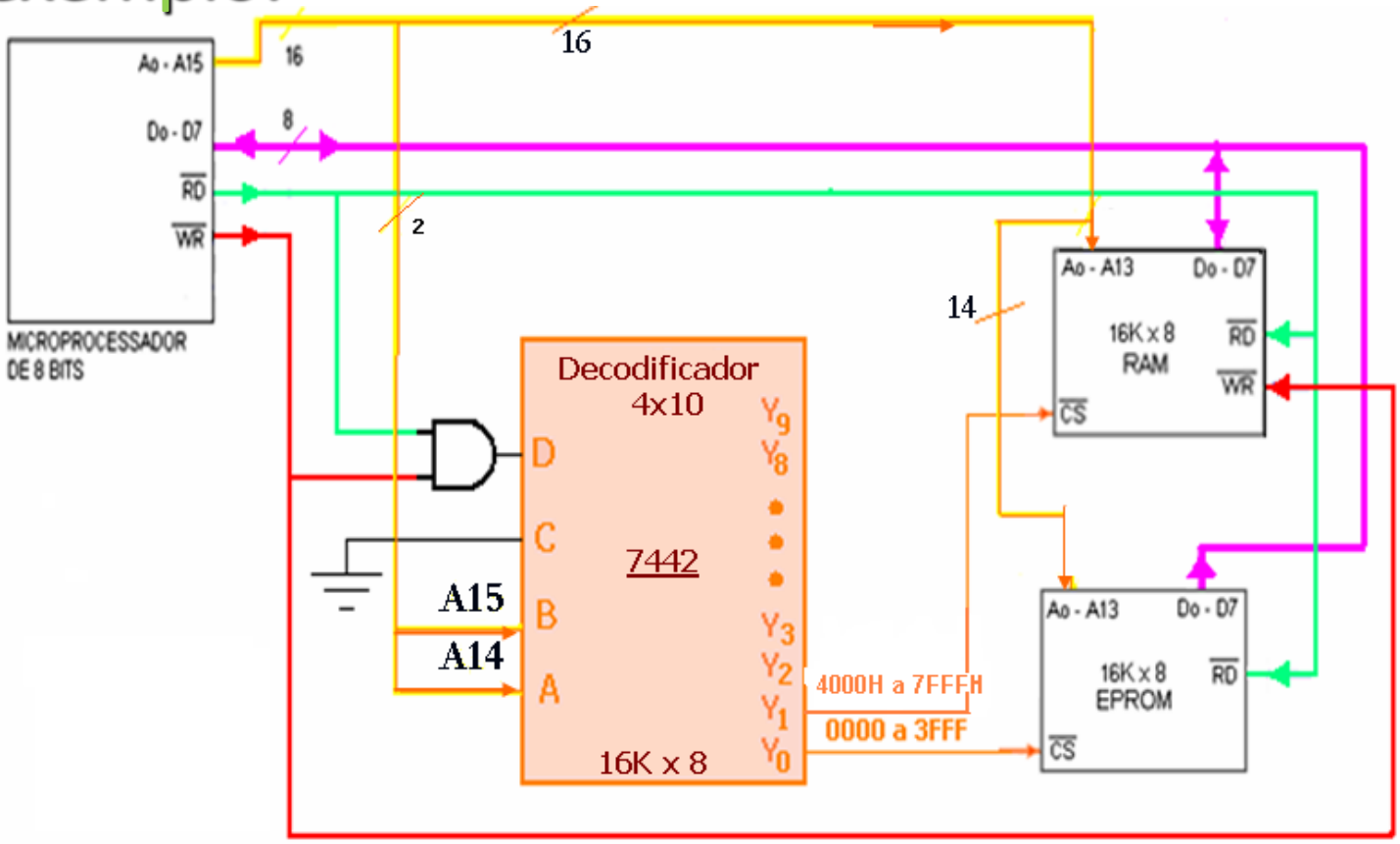
Mapeamento Completo

1.o exemplo:



Circuito completo do microprocessador ligado às Memórias

1.o exemplo:



Mapeamento a ser realizado

2.o exemplo:

Ligar no espaço de mapeamento de memória apenas memórias e reservar espaço para uma possível expansão das mesmas

Lógica de Seleção

2.o exemplo:

Lógica de Seleção do µP – Linhas de Endereços																	Memória		
Tipo	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Início (H)		Fim (H)
ROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	16k	
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1			3FFF
vazio expansão de ROM																			
RAM																			
vazio expansão de RAM ou I/O																			

Lógica de Seleção

2.o exemplo:

Lógica de Seleção do µP – Linhas de Endereços																	Memória		
Tipo	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Início (H)		Fim (H)
ROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	16k	
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1			3FFF
vazio expansão de ROM	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000	16k	
	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			7FFF
RAM																			
vazio expansão de RAM ou I/O																			

Lógica de Seleção

2.o exemplo:

Lógica de Seleção do µP – Linhas de Endereços																	Memória		
Tipo	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Início (H)		Fim (H)
ROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	16k	
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1			3FFF
vazio expansão de ROM	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000	16k	
	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			7FFF
RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000	16k	
	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1			BFFF
vazio expansão de RAM ou I/O																			

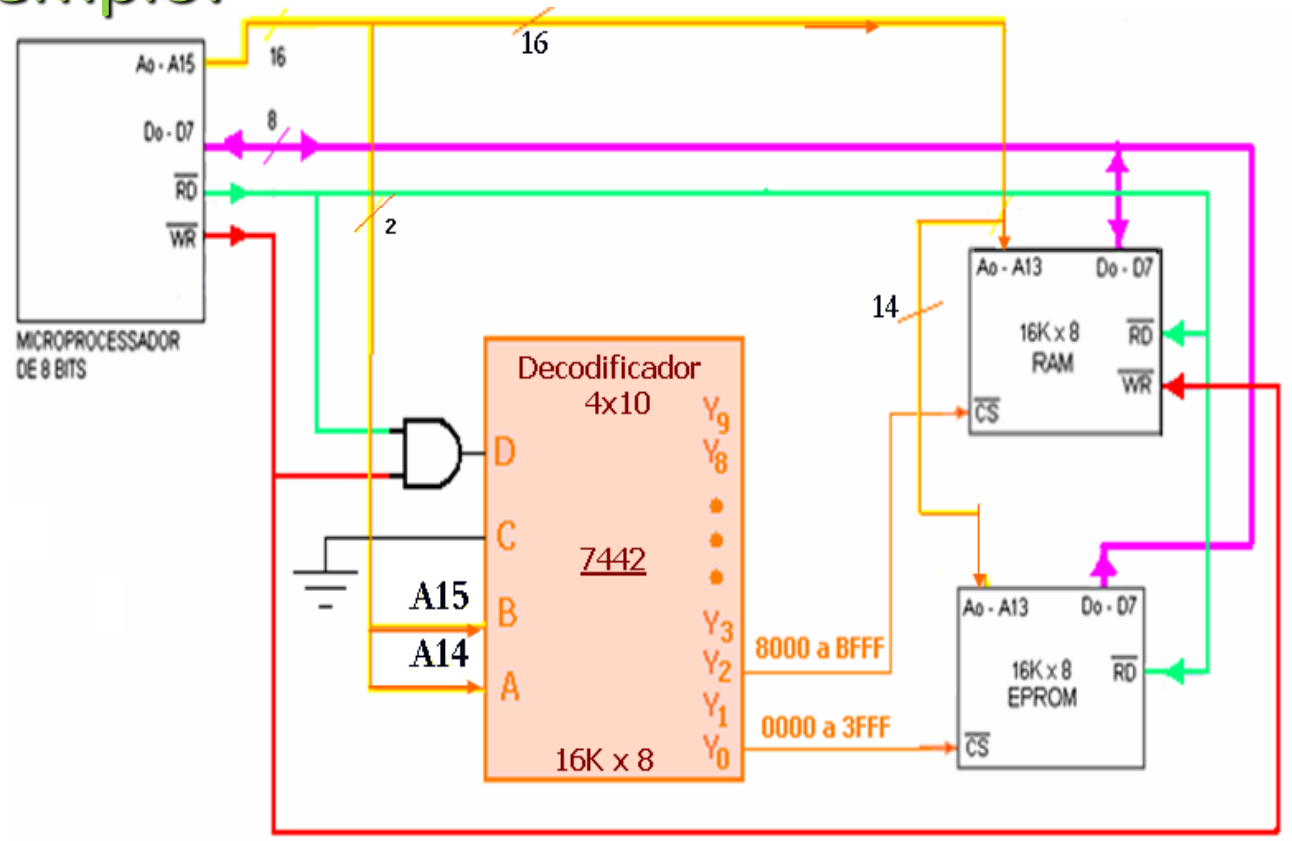
Lógica de Seleção

2.o exemplo:

Lógica de Seleção do µP – Linhas de Endereços																	Memória		
Tipo	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Início (H)		Fim (H)
ROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	16k	
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1			3FFF
vazio expansão de ROM	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000	16k	
	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			7FFF
RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000	16k	
	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1			BFFF
vazio expansão de RAM ou I/O	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C000	16k	
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			FFFF

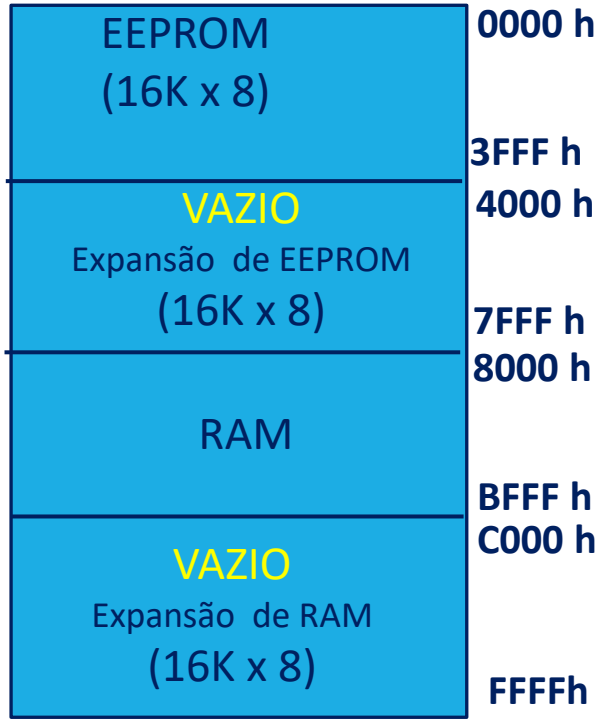
Circuito completo do microprocessador ligado às Memórias

2.o exemplo:



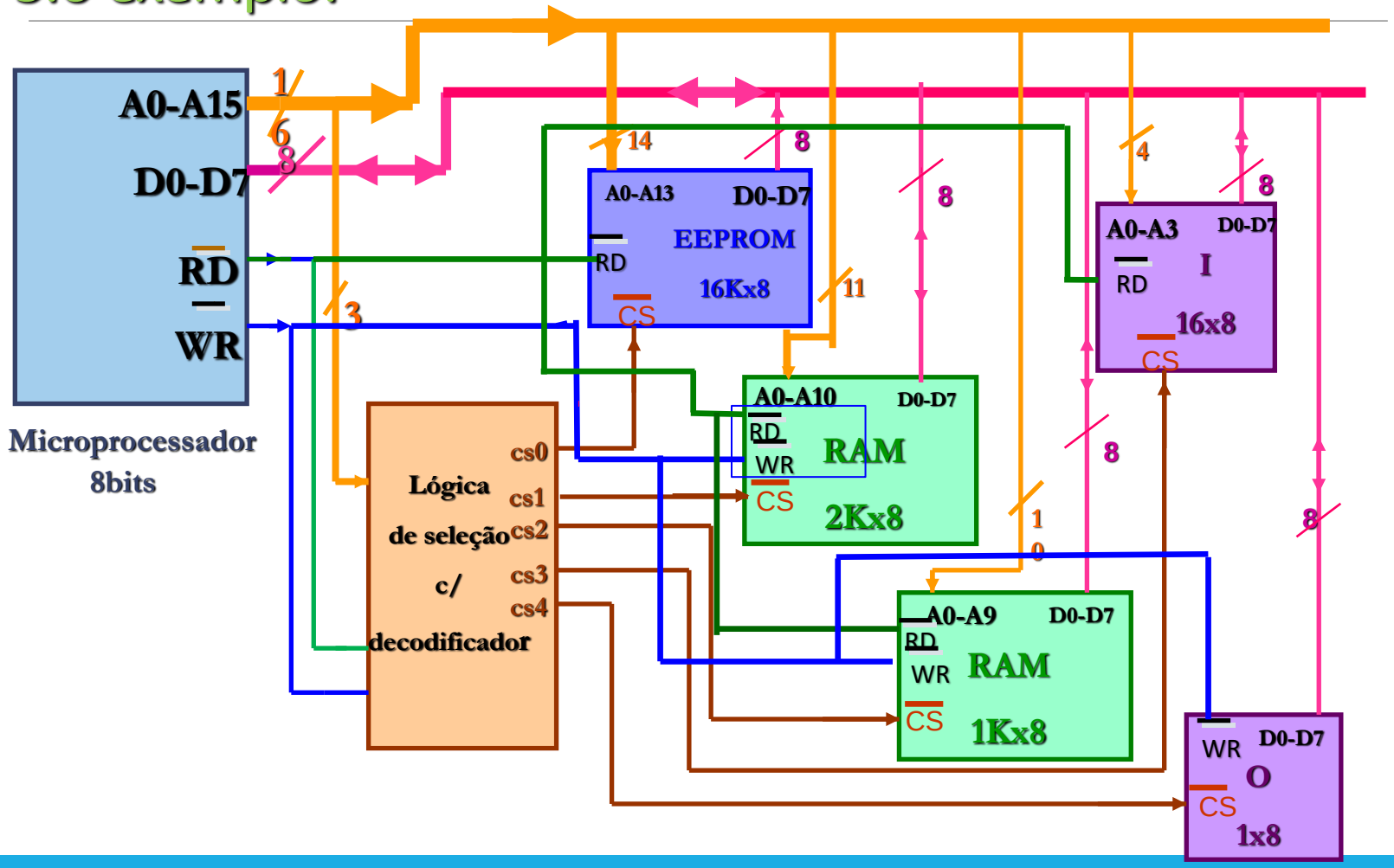
Mapeamento Completo

2.o exemplo:



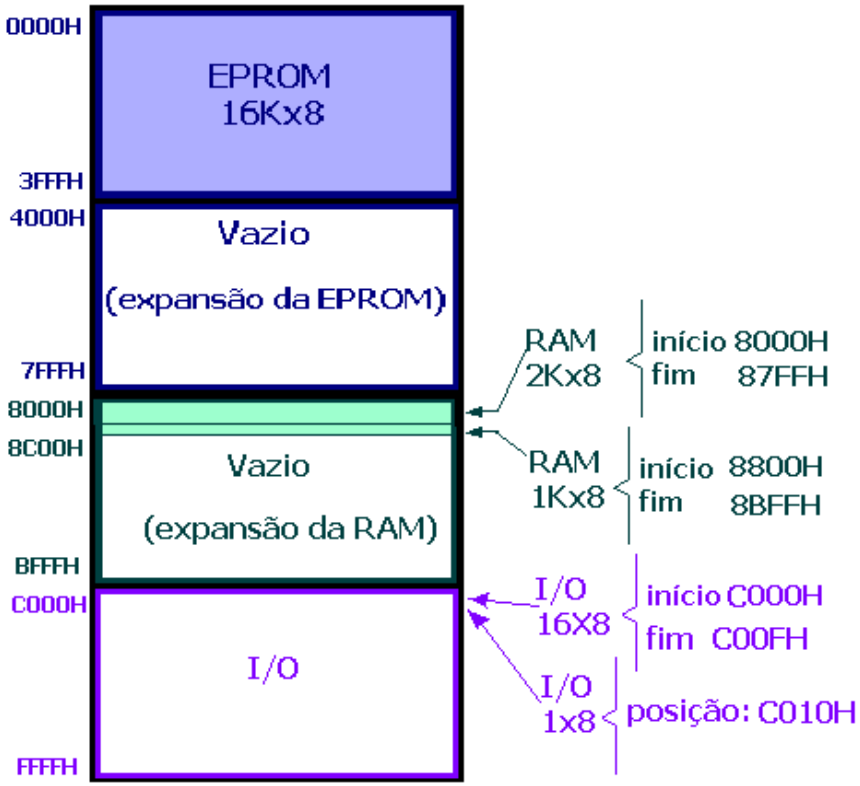
Como ligar o microprocessador às Memórias?

3.o exemplo:



Mapeamento Completo

3.o exemplo:



Lógica de Seleção de Memória e Dispositivos de I/O

Pode-se escolher qualquer endereço inicial ?

Resposta: Não!!! há endereços que simplificam a lógica de seleção para isso deve-se escolher Alinhar a memória

- Se a memória está **"alinhada"** com o endereço inicial, os bits de seleção tem o mesmo valor para qualquer posição da memória, o **que simplifica a lógica de seleção**

- a memória está alinhada com o endereço inicial se os **bits de endereçamento do bloco de memória** tem valor **zero** para o endereço inicial.

Lógica de Seleção de Memória e Dispositivos de I/O

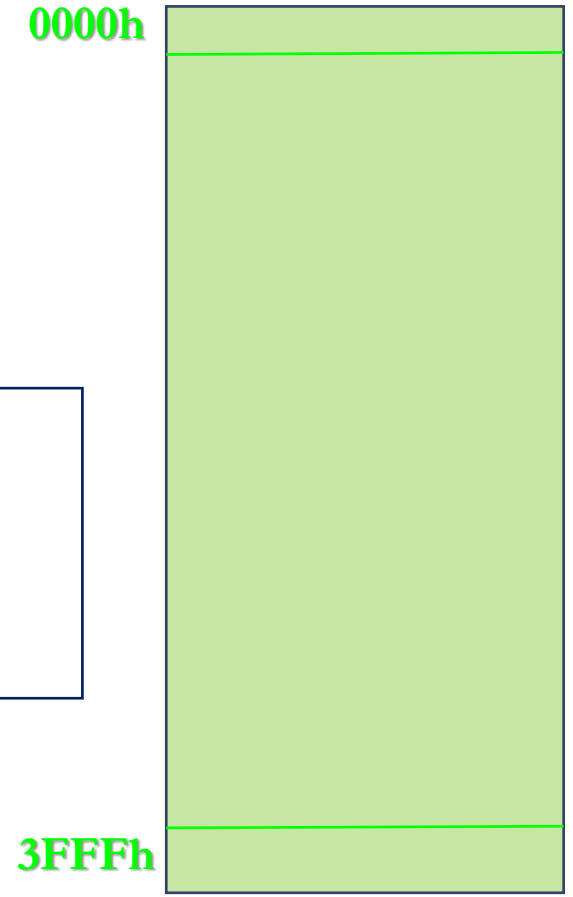
a) Memória não alinhada - RAM de 16Kx8

Memória de 16K x 8

tem $2^{14} \times 2^{10}$ = 14 linhas de endereçamento de A0 a A13

END. INICIAL:	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0000 H :	0	0	0	0	0	0	0	0	0	0	0	0	0	0
END. FINAL :														
3FFFH:	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Representação da RAM 16Kx 8



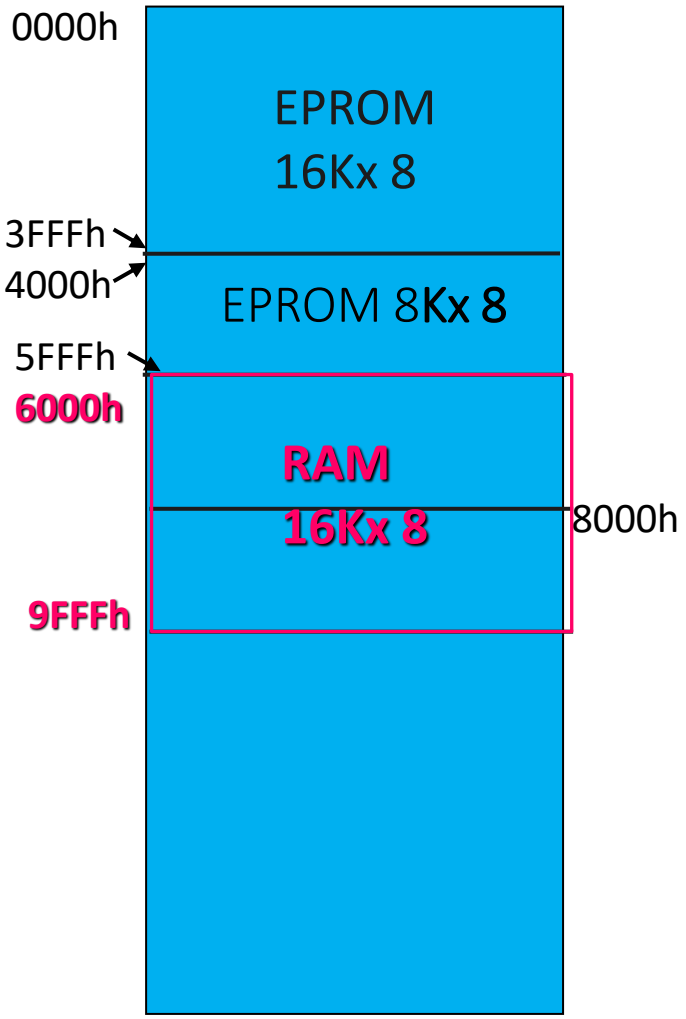
Lógica de Seleção de Memória e Dispositivos de I/O

a) Memória não alinhada - RAM de 16Kx8

A13

END. INICIAL:	**0**	**1**	**1**	**0**	**0000**				**0000**				**0000**		
END. FINAL :	**1**	**0**	**0**	**1**	**1111**				**1111**				**1111**		

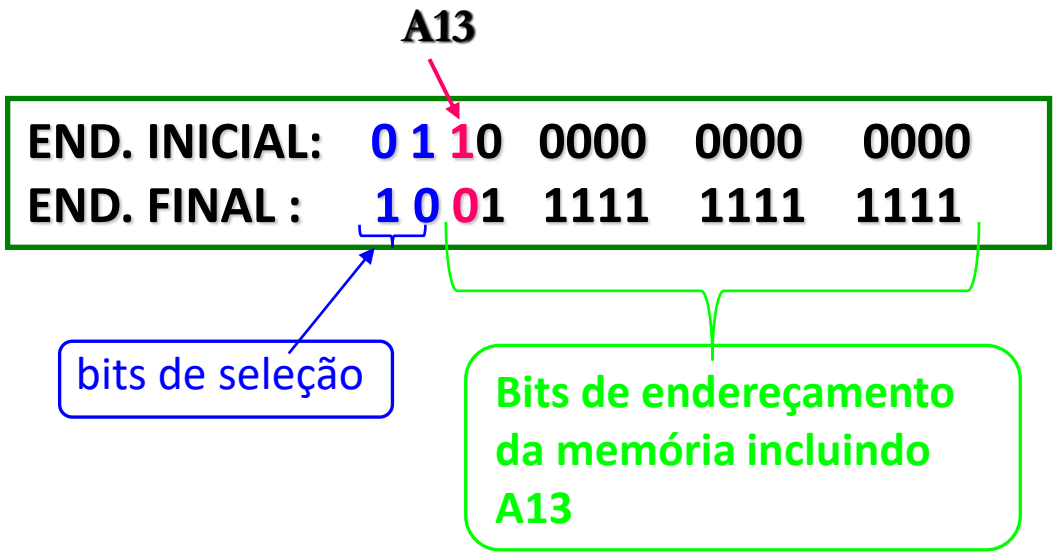
- A13 que é bit de endereçamento da memória, tem valor 1, para o end. Inicial que o microprocessador envia.
- Os bits de seleção para endereço inicial e final **NÃO** são os mesmos
- Para alinhar a memória. a lógica de seleção deve ser feita com divisão de 8Kx8 (ou menor), para que esse bloco tenha os bits de Endereço em 0.

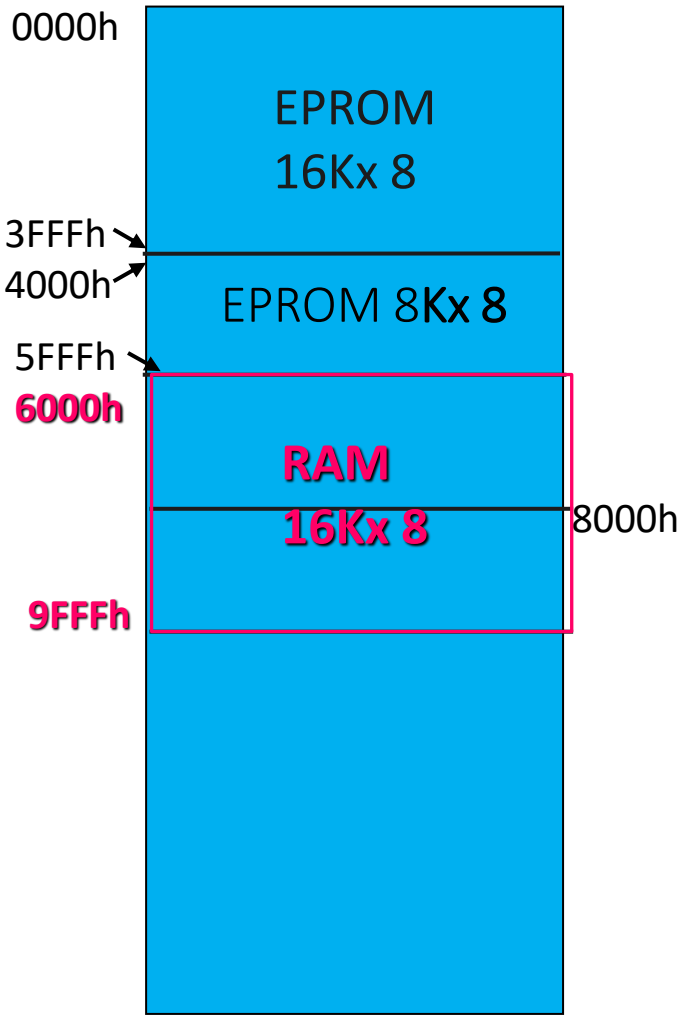


Lógica de Seleção de Memória e Dispositivos de I/O

a) Memória não alinhada - RAM de 16Kx8

A memória ficou alocada da posição 6000h a 9FFFh portanto, quando o microprocessador coloca no duto de endereços o valor 6000h deveria acessar a posição 0000h da memória, mas o bit A13, que pertence ao endereçamento da memória é 1 para 6000h





Lógica de Seleção de Memória e Dispositivos de I/O

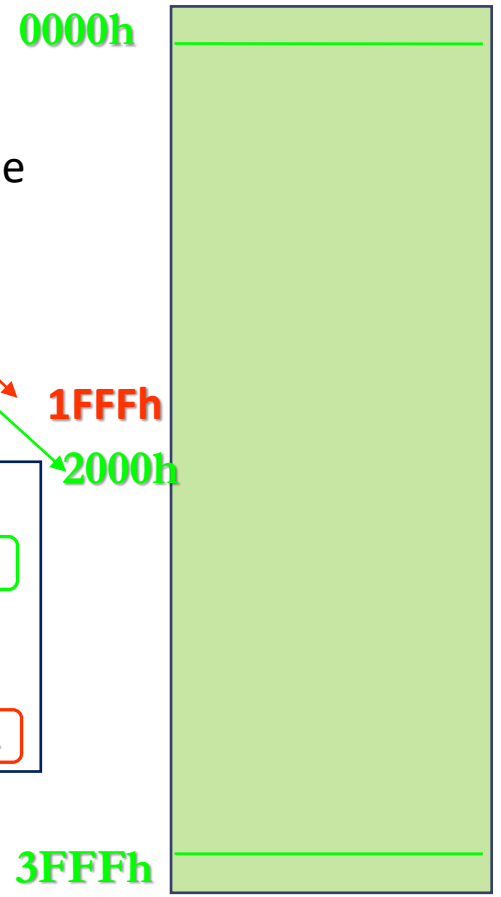
a) Memória não alinhada - RAM de 16Kx8

Um endereço mais baixo 6000h enviado pelo microprocessador acessa posições mais altas da memória (posição 2000h) do que quando o microprocessador envia um endereço mais alto 9FFFh que vai acessar posições mais baixas da memória (posição 1FFFh)

Complicando para o programador que vai ter que armazenar na memória dados da posição mais alta para a mais baixa.

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
END. INICIAL: 6000 H :	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
END. FINAL : 9FFFH:	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

Representação da RAM 16Kx 8

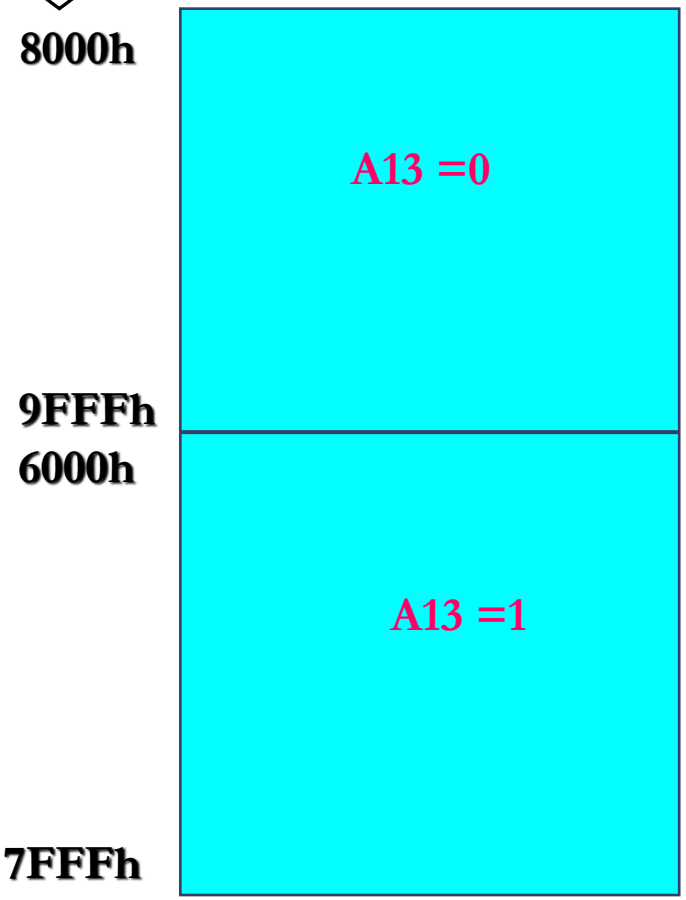


Lógica de Seleção de Memória e Dispositivos de I/O

a) Memória não alinhada - RAM de 16Kx8

Endereços enviados pelo microprocessador que acessam o chip da memória

Chip da RAM 16Kx 8



END. INICIAL:	**0**	**1**	**1**	**0**	**0000**				**0000**				**0000**		
END. FINAL :	**1**	**0**	**0**	**1**	**1111**				**1111**				**1111**		

Obs: O chip da memória é acessado da metade até o fim (qdo A13=0) e depois do início até a metade (qdo A13=1)

Lógica de Seleção de Memória e Dispositivos de I/O

b) Memória alinhada - RAM de 16Kx8

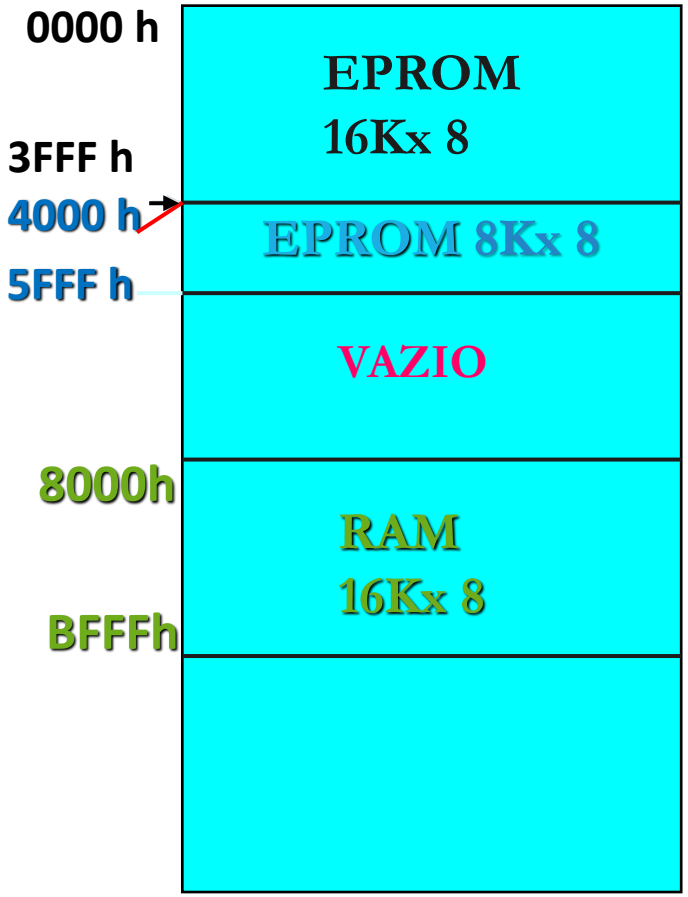
Pode ser criado um “buraco” (vazio) para o alinhamento

A13

END. INICIAL: 1 0 00 0000 0000 0000

END. FINAL : 1 0 11 1111 1111 1111

- 0s bit de endereçamento da memória, A0 até A13, tem valor “ 0”;
- Os bits de seleção, para endereço inicial e final, são os mesmos;
- O lógica de seleção deve ser feita com divisão de 16Kx8 , o mesmo tamanho da memória.



Lógica de Seleção de Memória e Dispositivos de I/O

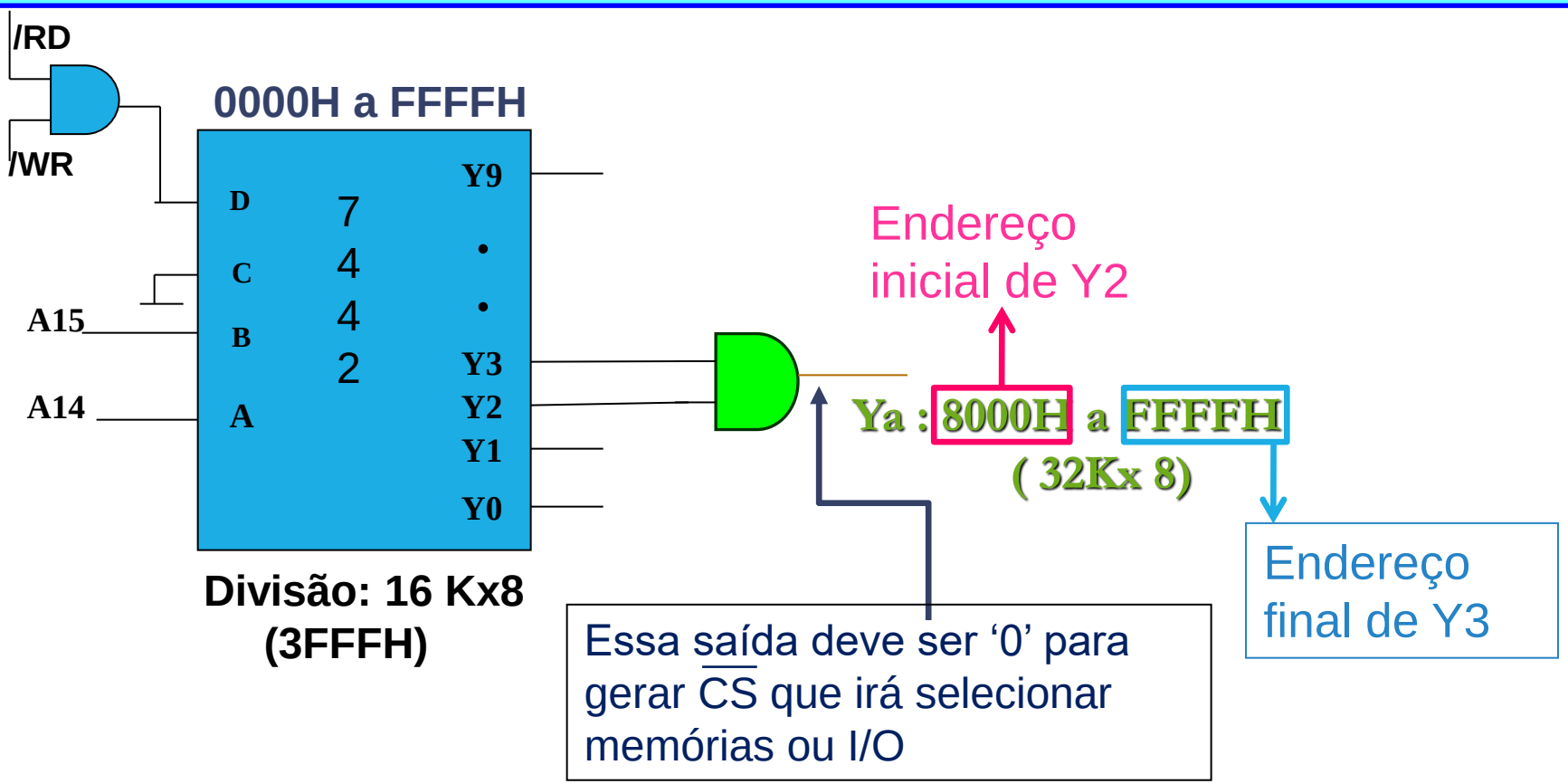
- **No mapeamento das memórias, é interessante que as memórias do tipo EPROM sejam mapeadas em sequência, para se ter continuidade do programa armazenado;**

- **Memórias do tipo RAM também são mapeadas em sequência para se ter continuidade na área de dados.**

- **Para evitar a ocorrência de memória não alinhada, além da escolha do endereço inicial adequado, as memórias devem ser mapeadas de forma que as de maior organização ocupem os blocos iniciais no mapa de endereços**

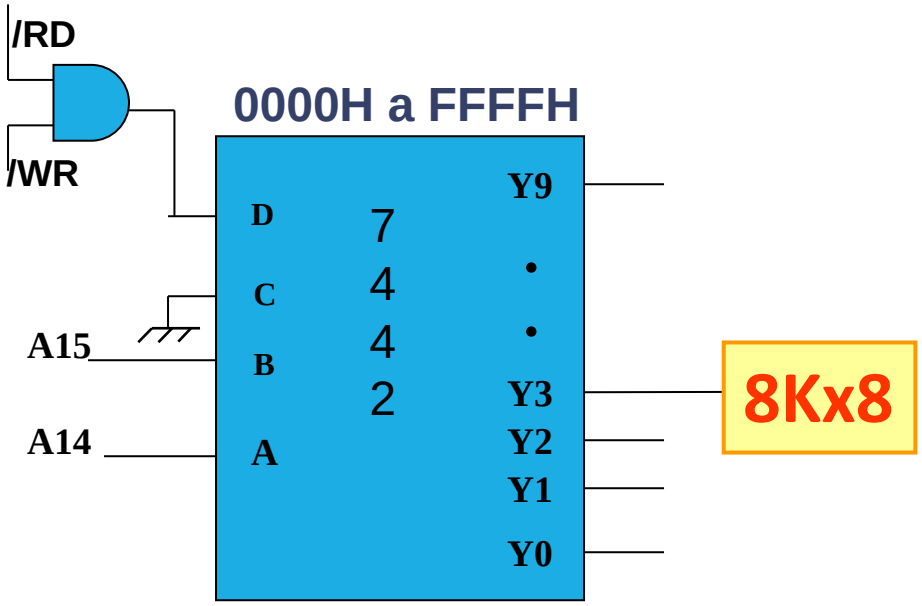
Lógica de Seleção de Memória e Dispositivos de I/O

Pode-se combinar linhas de seleção através da lógica AND, para selecionar organizações de memória MAIORES do que a faixa de endereço das saídas de seleção



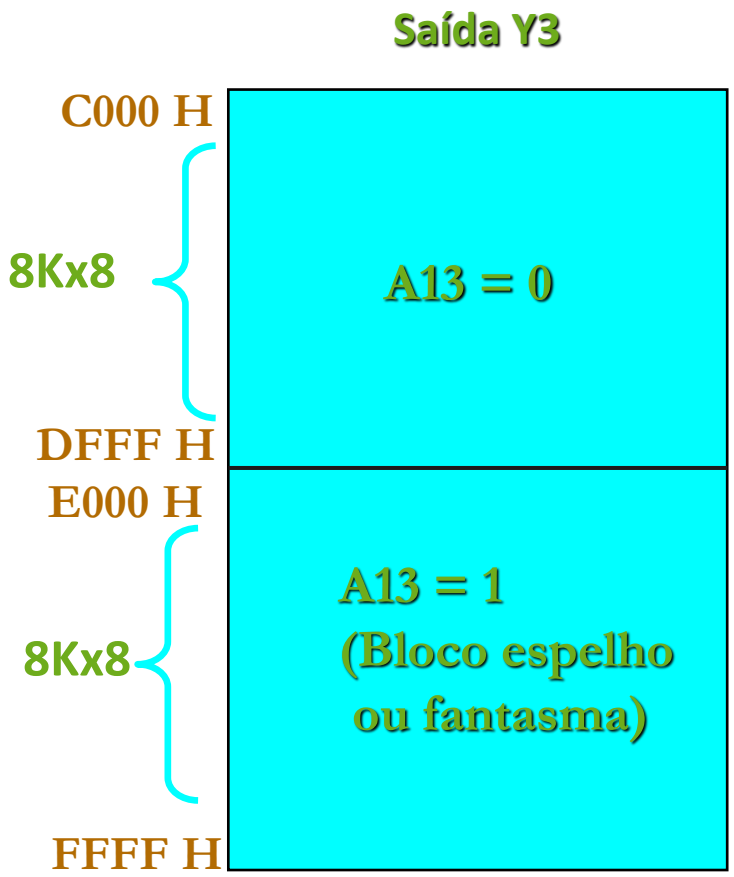
Lógica de Seleção de Memória e Dispositivos de I/O

O que ocorre se uma memória com <u>organização menor</u> que a da saída de seleção, for interligada na saída Y3?



Divisão: 16 Kx8
(3FFFH)

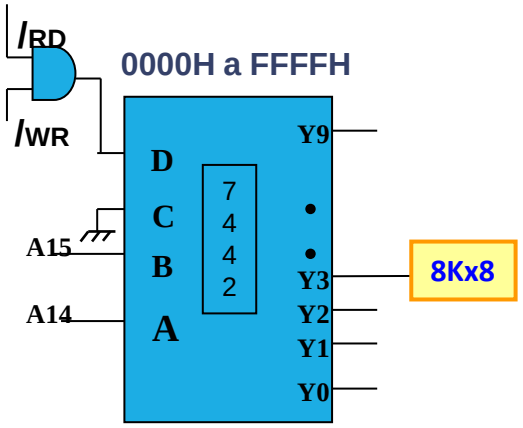
Saída Y3: C000H a FFFFH



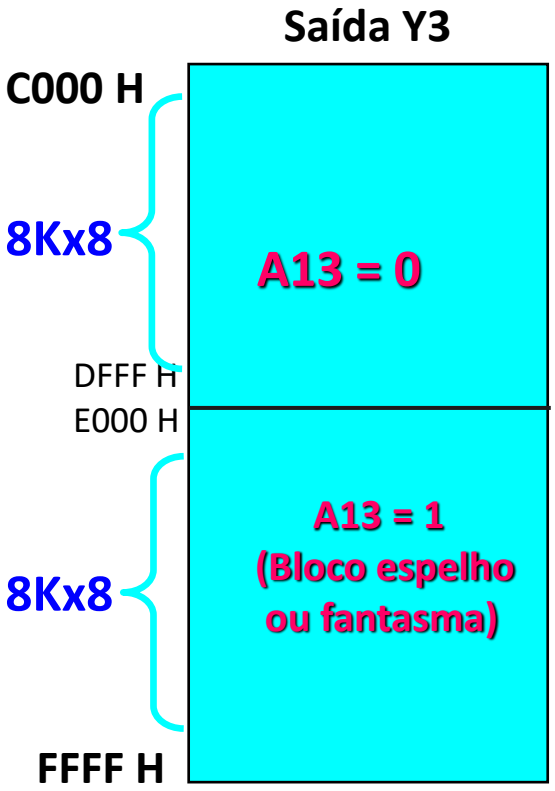
Lógica de Seleção de Memória e Dispositivos de I/O

Exemplo 1: O que ocorre se uma memória com <u>organização menor</u> que a da saída de seleção, for interligada na saída Y3?

A13 não é entrada de endereçamento da organização 8K x 8 nem está sendo utilizada na entrada do decodificador como entrada de seleção.Portanto, gera espaços fantasmas (ou espelhos)



Divisão: 16 Kx8 (3FFFH)



Saída Y3: C000H a FFFFH

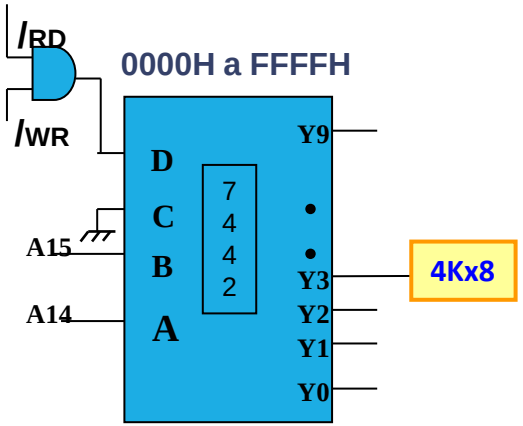
	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	faixa
Y3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C000h
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFFh
				1	1	1	1	1	1	1	1	1	1	1	1	1	8Kx8

Organização de 8Kx8 $\Rightarrow$ $2^3 \times 2^{10}$

Possui 13 linhas de endereçamento de A0 a A12

Lógica de Seleção de Memória e Dispositivos de I/O

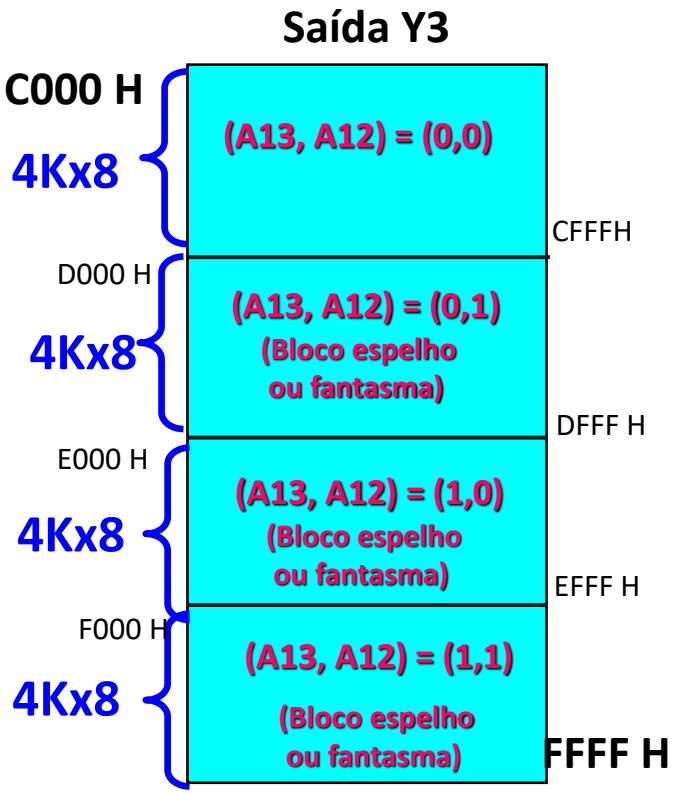
Exemplo 2: O que ocorre se uma memória com <u>organização menor</u> que a da saída de seleção, for interligada na saída Y3?



A13 e A12 não são entradas de endereçamento da organização 4K x 8 nem estão sendo utilizadas na entrada do decodificador como entrada de seleção. Portanto, geram espaços fantasmas (ou espelhos)

Saída Y3: C000H a FFFFH

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	faixa
Y3	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C000h
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	FFFFh
					1	1	1	1	1	1	1	1	1	1	1	1	4Kx8



Organização de 4Kx8 $\Longrightarrow$ $2^2 \times 2^{10}$

Possui 12 linhas de endereçamento de A0 a A11

Lógica de Seleção de Memória e Dispositivos de I/O

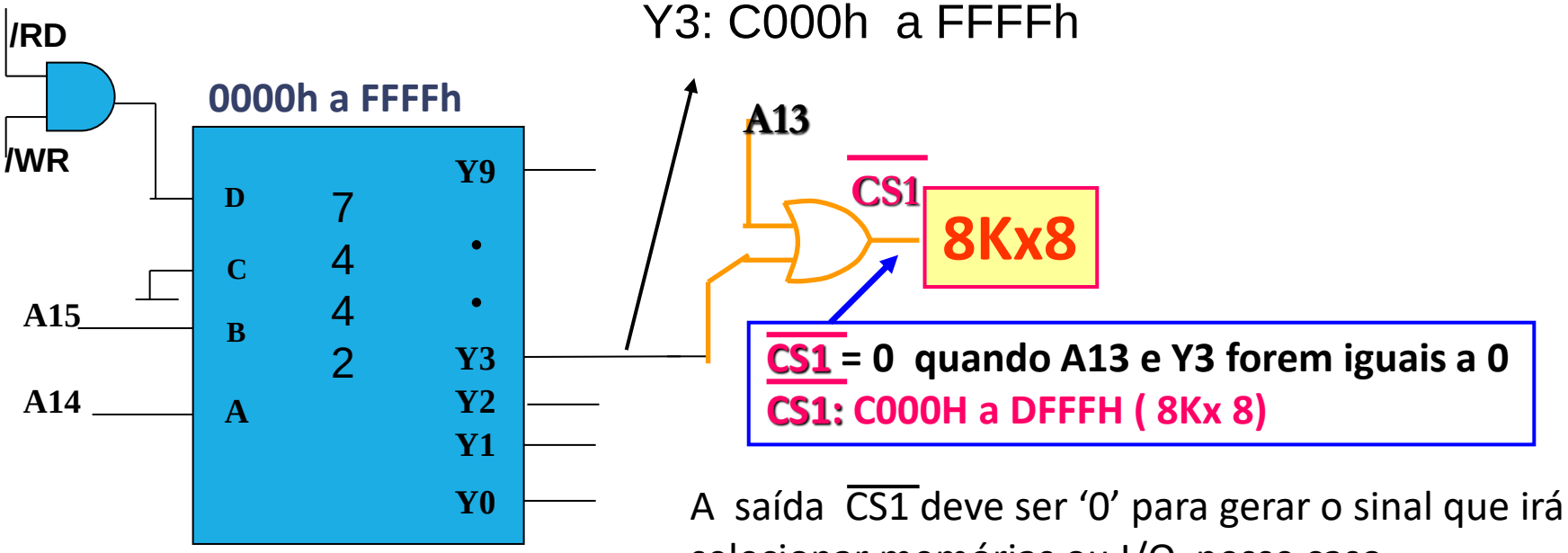
A13 é bit de seleção da memória de 8Kx8, mas NÃO está presente no decodificador, portanto é irrelevante, podendo valer 1 ou 0.

- **Os dois blocos de 8K podem ser usados para selecionar a memória de 8Kx8 : C000H a DFFFh e E000h a FFFFh.**

- **Os dois blocos acessam as mesmas posições físicas das memórias, p. ex., C000h e E000h acessam a mesma posição da memória.**
  Portanto, para este caso (de apenas uma linha de endereço , A13, não constar na seleção) o microprocessador pode enviar 2 endereços para acessar a mesma posição da memória.

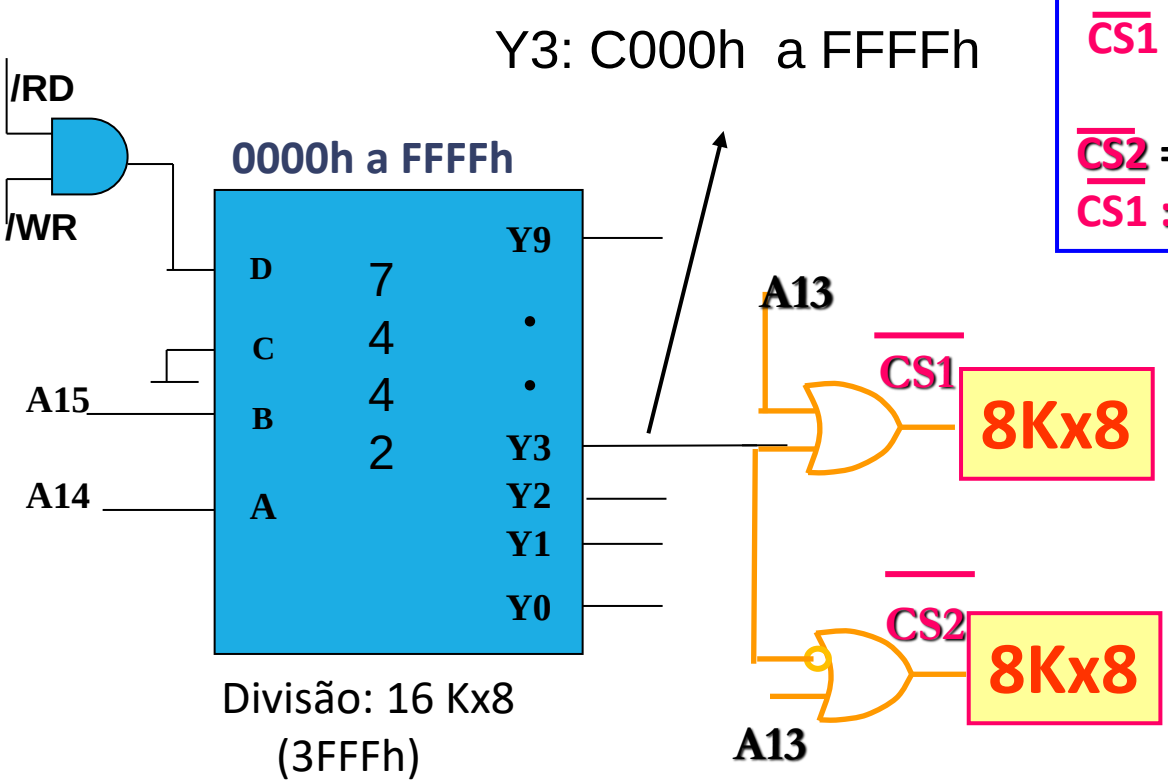
Lógica de Seleção de Memória e Dispositivos de I/O

Pode-se combinar linhas de seleção através da lógica OR, combinada com linhas de endereço adequadas, para selecionar organizações de memória MENORES do que a faixa de endereço das saídas de seleção.



Lógica de Seleção de Memória e Dispositivos de I/O

Selecionando organizações de memória MENORES do que a faixa de endereço das saídas de seleção utilizando portas OR.



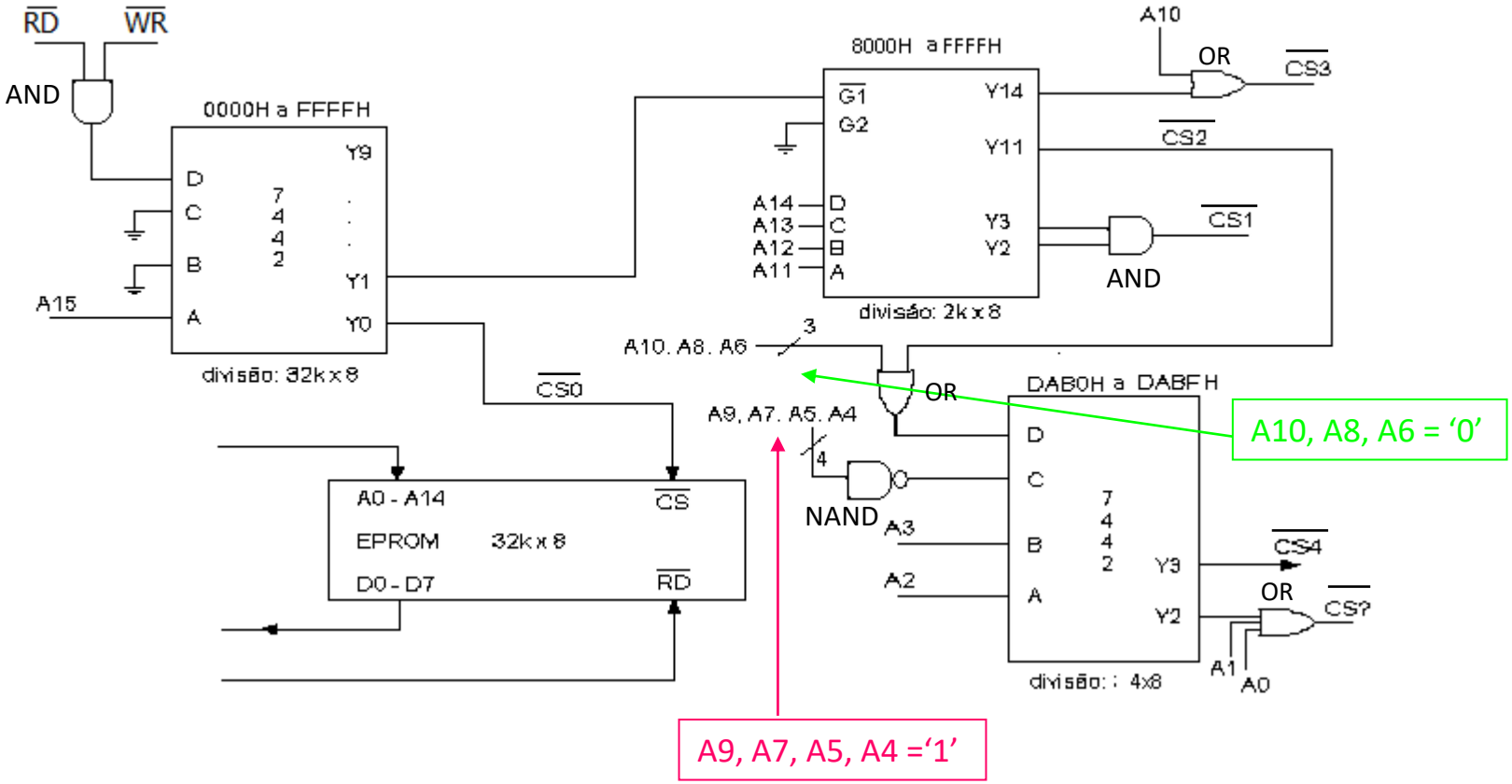
$\overline{CS1}$ = 0 quando A13=0 e Y3=0

$\overline{CS1}$: C000H a DFFFH (8Kx 8)

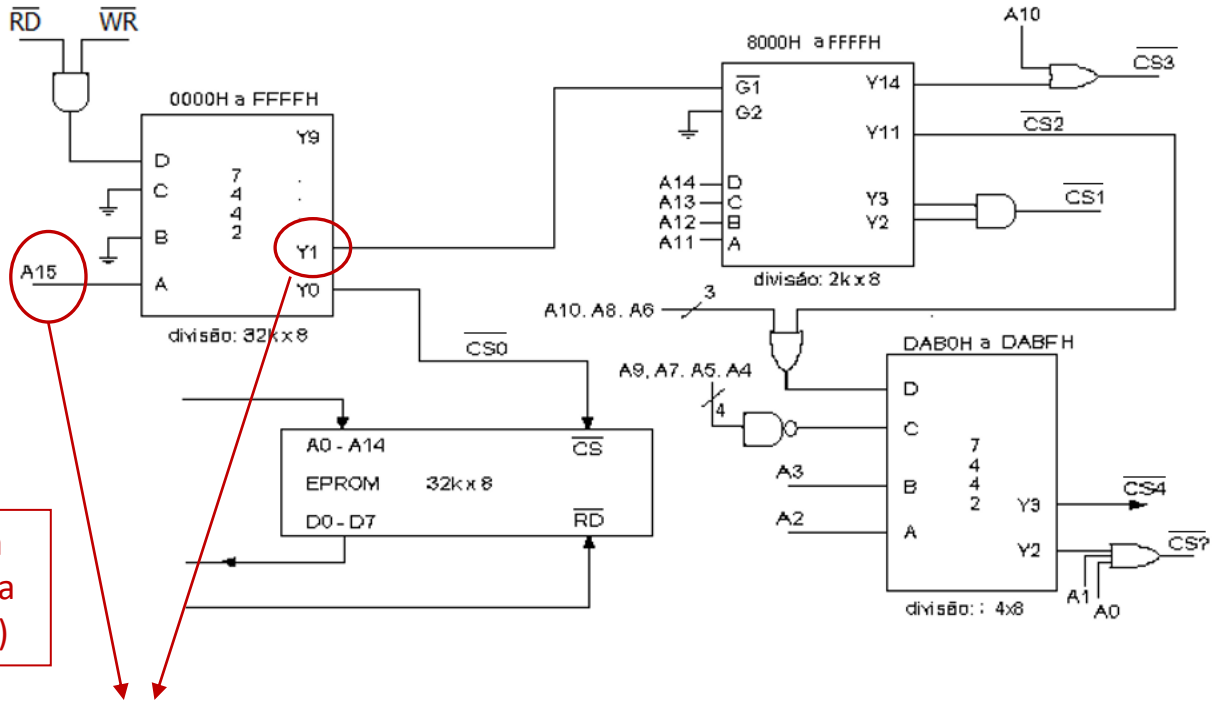
$\overline{CS2}$ = 0 quando A13 =1 e Y3 =0

$\overline{CS2}$: E000H a FFFFH (8Kx 8)

Exercício:
qual a faixa de endereços selecionada por cada saída $\overline{CS}$?



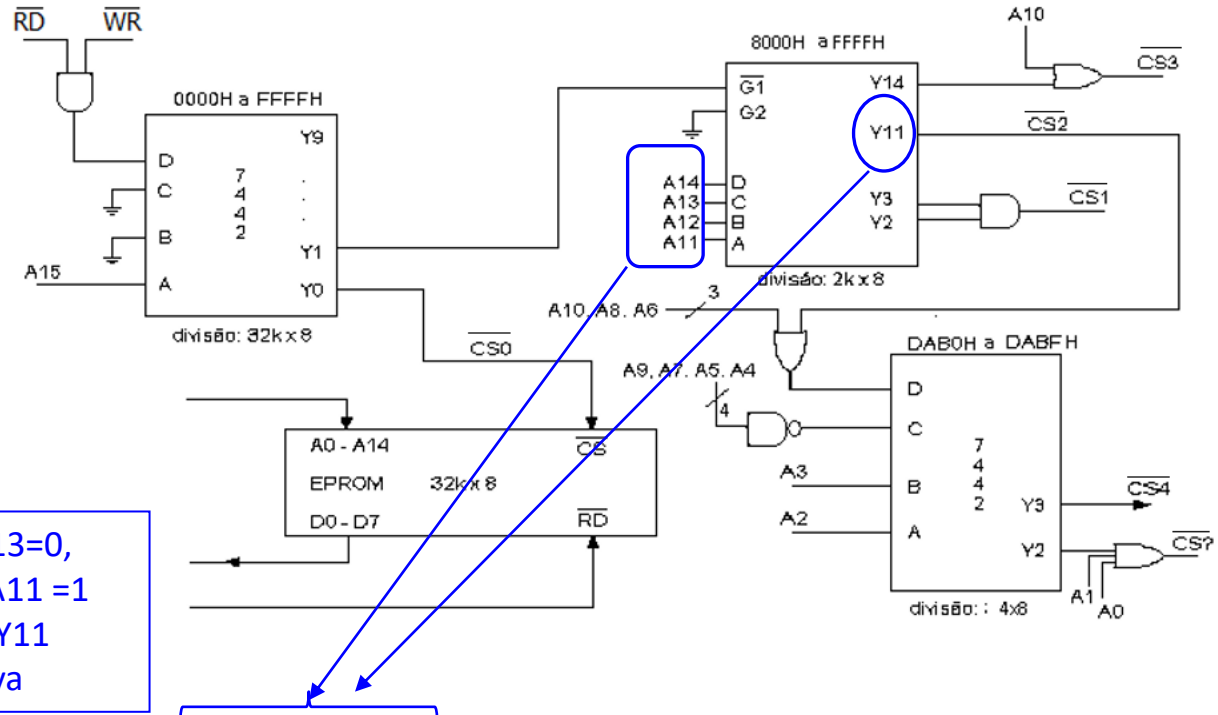
Resolução Exercício:
qual a faixa de endereços selecionada por cada saída $\overline{CS}$?



A15 = 1 para que Y1 esteja ativa (Y1 = 0)

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	faixa
CS	1																
	1																

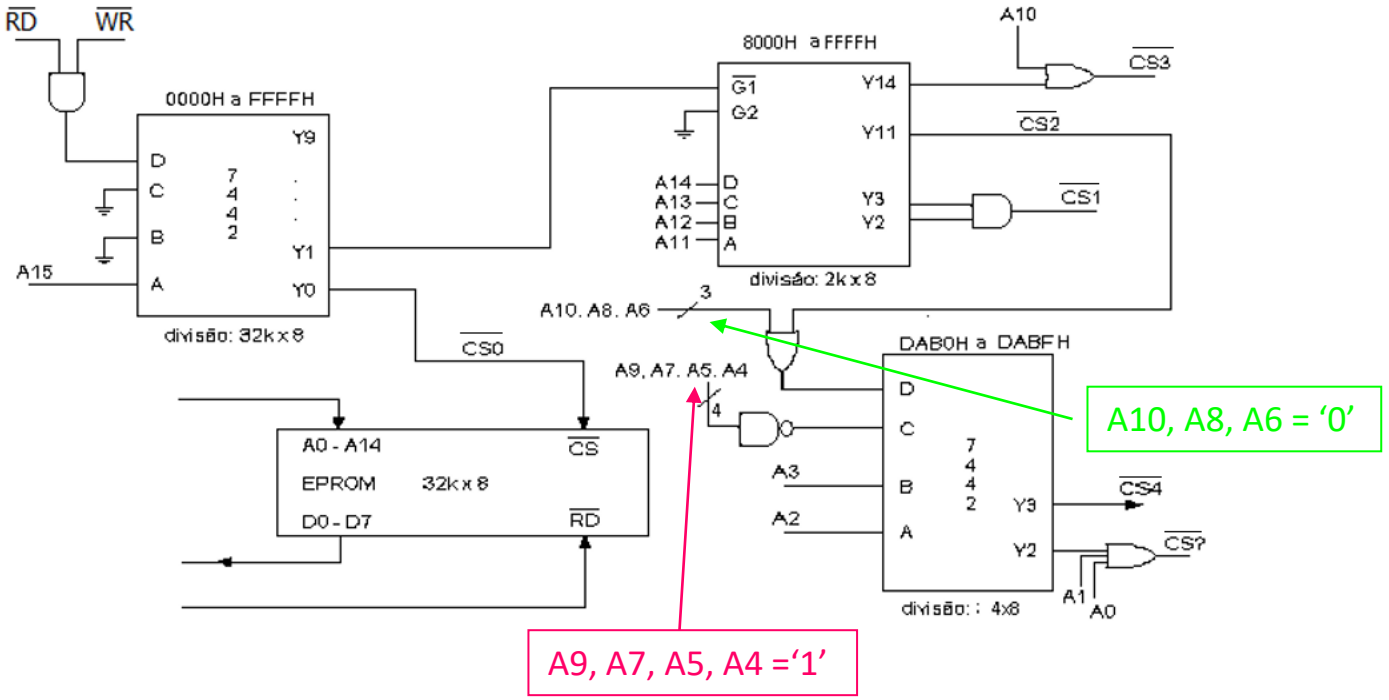
Resolução Exercício:
qual a faixa de endereços selecionada por cada saída $\overline{CS}$?



A14=1, A13=0, A12=1 e A11 =1 para que Y11 esteja ativa

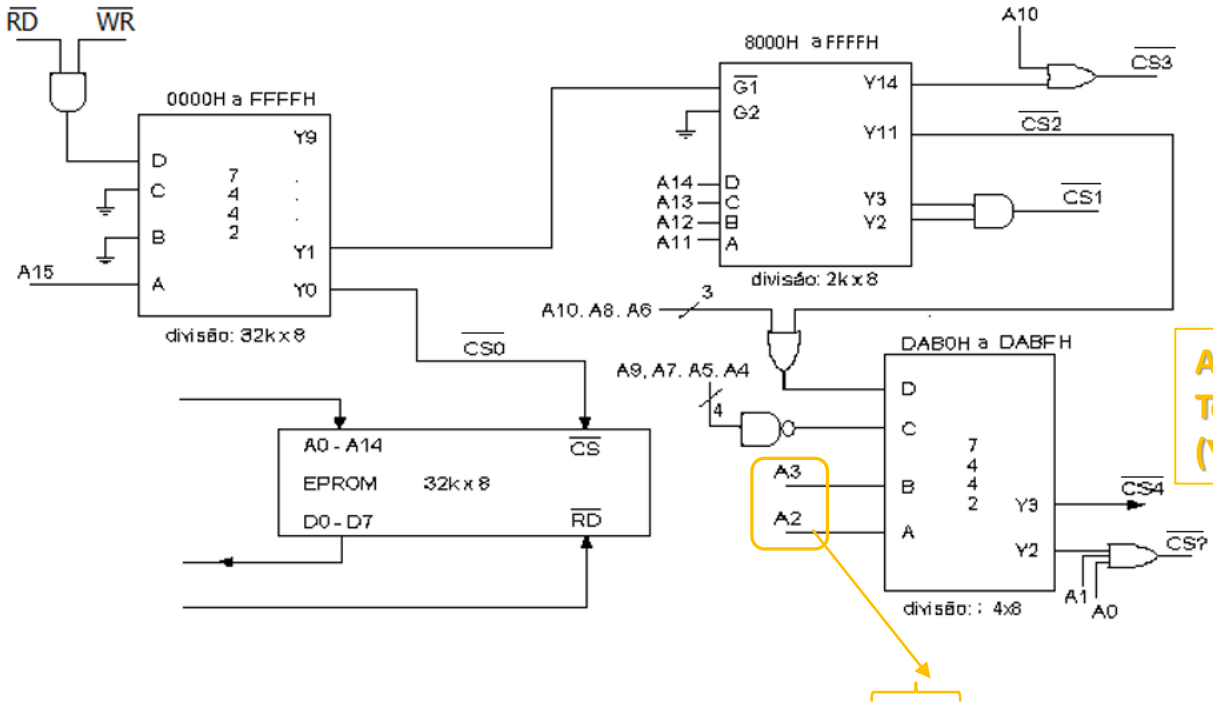
	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	faixa
CS	1	1	0	1	1												
	1	1	0	1	1												

Resolução Exercício:
qual a faixa de endereços selecionada por cada saída $\overline{CS}$?



	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	faixa
CS	1	1	0	1	1	0	1	0	1	0	1	1					
	1	1	0	1	1	0	1	0	1	0	1	1					

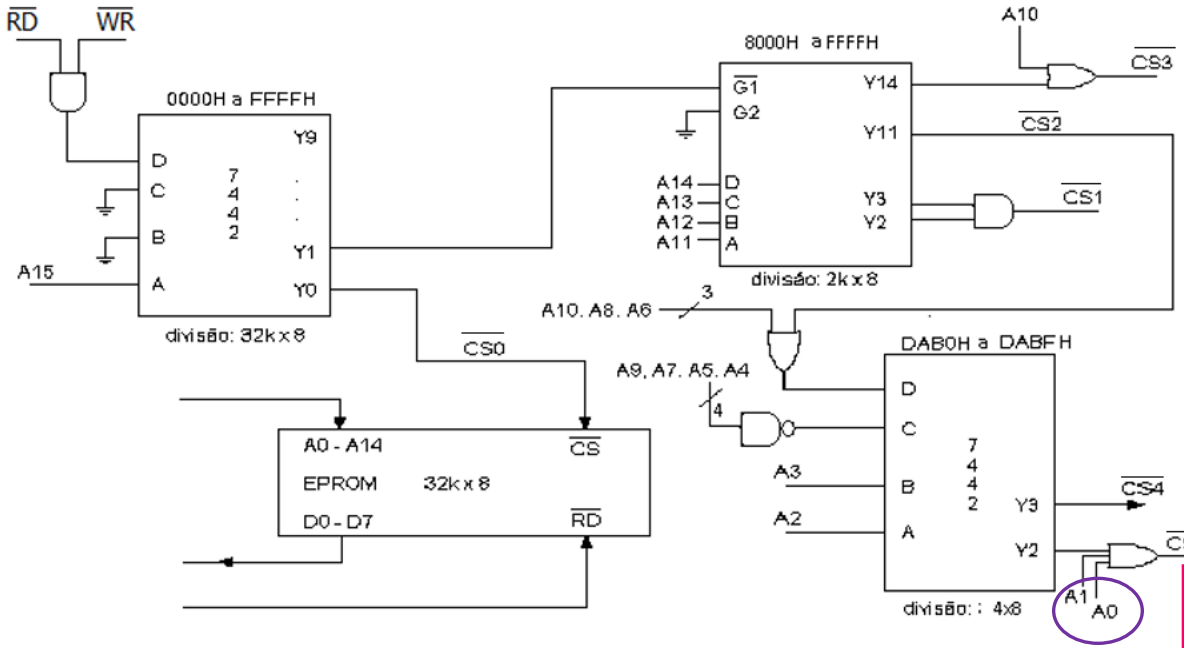
Resolução Exercício:
qual a faixa de endereços selecionada por cada saída $\overline{CS}$?



A3=1 e A2 = 0
Torna a saída Y2 ativa (Y2=0)

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	faixa
CS	1	1	0	1	1	0	1	0	1	0	1	1	1	0			
	1	1	0	1	1	0	1	0	1	0	1	1	1	0			

Resolução Exercício:
qual a faixa de endereços selecionada por cada saída $\overline{CS}$?



Para que CS=0
as linhas de endereços A1 e A0 devem ser 0

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	faixa
CS	1	1	0	1	1	0	1	0	1	0	1	1	1	0	0	0	DAB8h
	1	1	0	1	1	0	1	0	1	0	1	1	1	0	0	0	

Todos os bits de Endereçamento (A0 a A15) entram na seleção portanto, existe apenas 1 endereço que seleciona CS

FIM