



Departamento de Engenharia Elétrica e de Computação - EESC-USP

SEL-0415 Introdução à Organização de Computadores

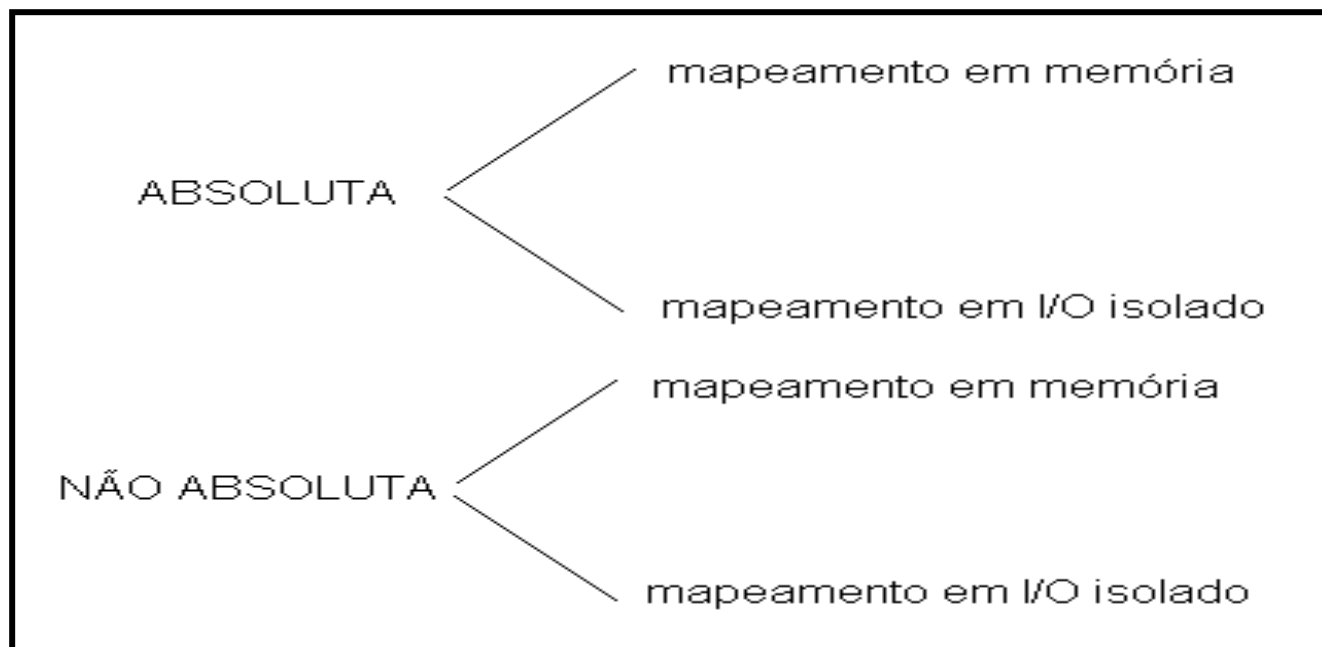
Aula 5 Parte 2 :

Tipos de Lógica de Seleção

Profa. Luiza Maria Romeiro Codá

Lógica de Seleção de Memórias e Dispositivos de I/O

Tipos de Lógica de Seleção



Tipos de Mapeamento de um microprocessador

Mapeamento em Memória:

- os sinais de controle de leitura ou gravação que o microprocessador gera para selecionar memórias e I/O são os **mesmos** (ex: $\overline{\text{RD}}$ e $\overline{\text{WR}}$ ou $\overline{\text{MEMR}}$ e $\overline{\text{MEMW}}$);
- Os dispositivos de memória e I/O são mapeados no mesmo espaço;
- Mesma lógica de seleção com decodificadores

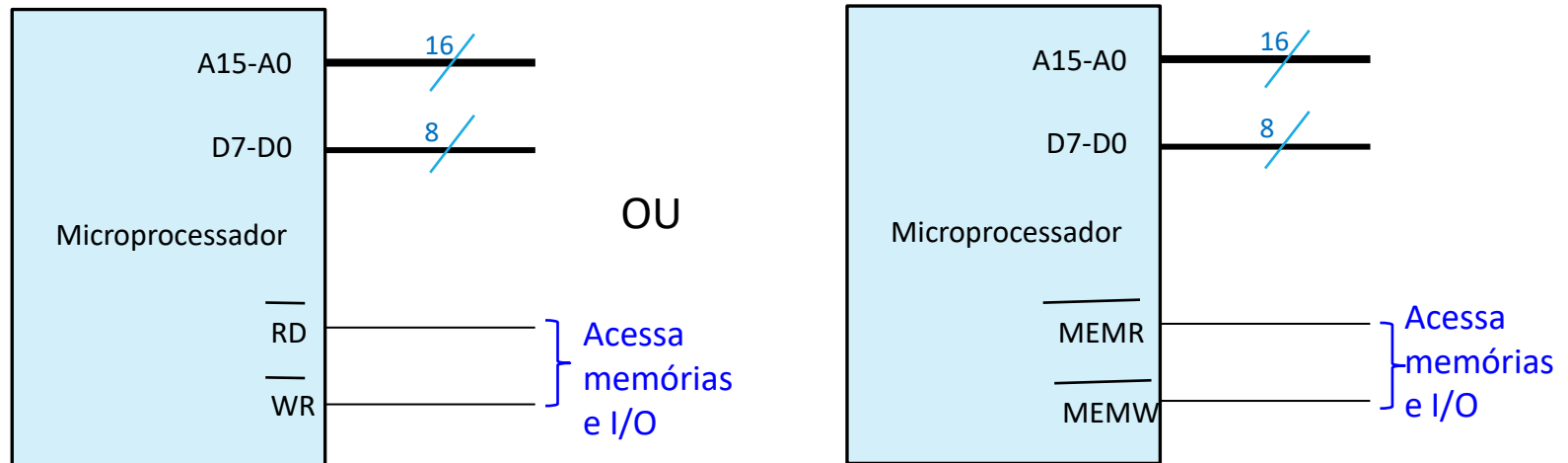
Mapeamento em I/O isolado:

- os sinais de controle de leitura e gravação que o microprocessador gera para selecionar memórias e I/O são **diferentes** (ex: $\overline{\text{MEMR}}$ e $\overline{\text{MEMW}}$ para selecionar memórias e $\overline{\text{IOR}}$ e $\overline{\text{IOW}}$ para selecionar dispositivo de I/O;.
- Os dispositivos de I/O são mapeados num espaço de I/O separado do espaço de memória.
- É feita uma lógica de seleção separada para cada espaço.

Microprocessador com Mapeamento em Memória

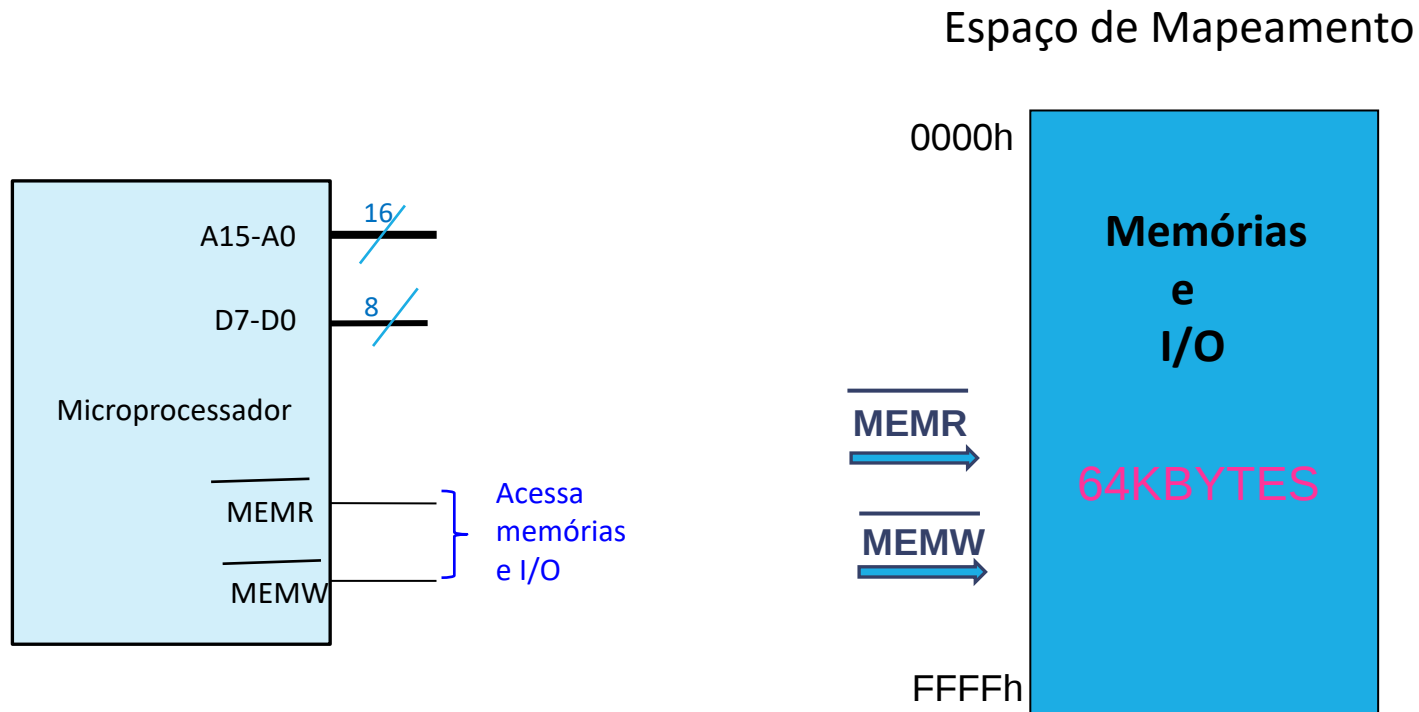
exemplo em um microprocessador com 16 linhas no duto de endereços e 8 linhas de duto de dados

- no espaço de 64K bytes são mapeadas as memórias e os dispositivos de I/O ;
- os sinais de controle de leitura e gravação são *os mesmos* para memórias e para I/O ($\overline{RD}$ e $\overline{WR}$ ou $\overline{MEMR}$ e $\overline{MEMW}$)



Microprocessador com mapeamento em Memória

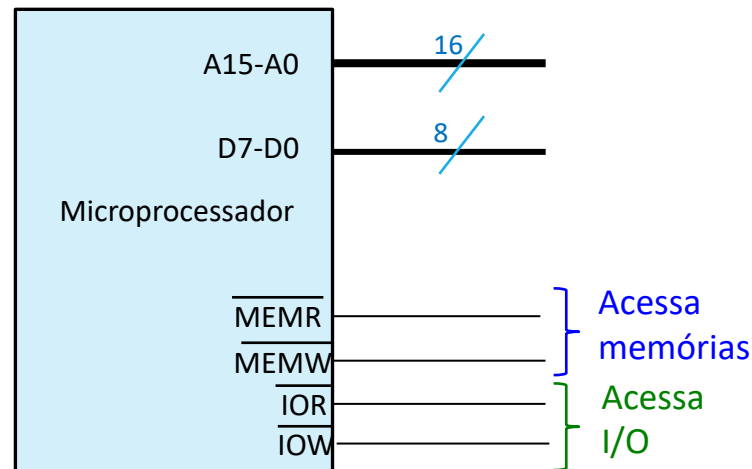
microprocessador com 16 linhas no duto de endereços e 8 linhas de duto de dados



Microprocessador com Mapeamento em I/O Isolado

exemplo em um microprocessador com 16 linhas no duto de endereços e 8 linhas de duto de dados e espaço de I/O de 256 bytes

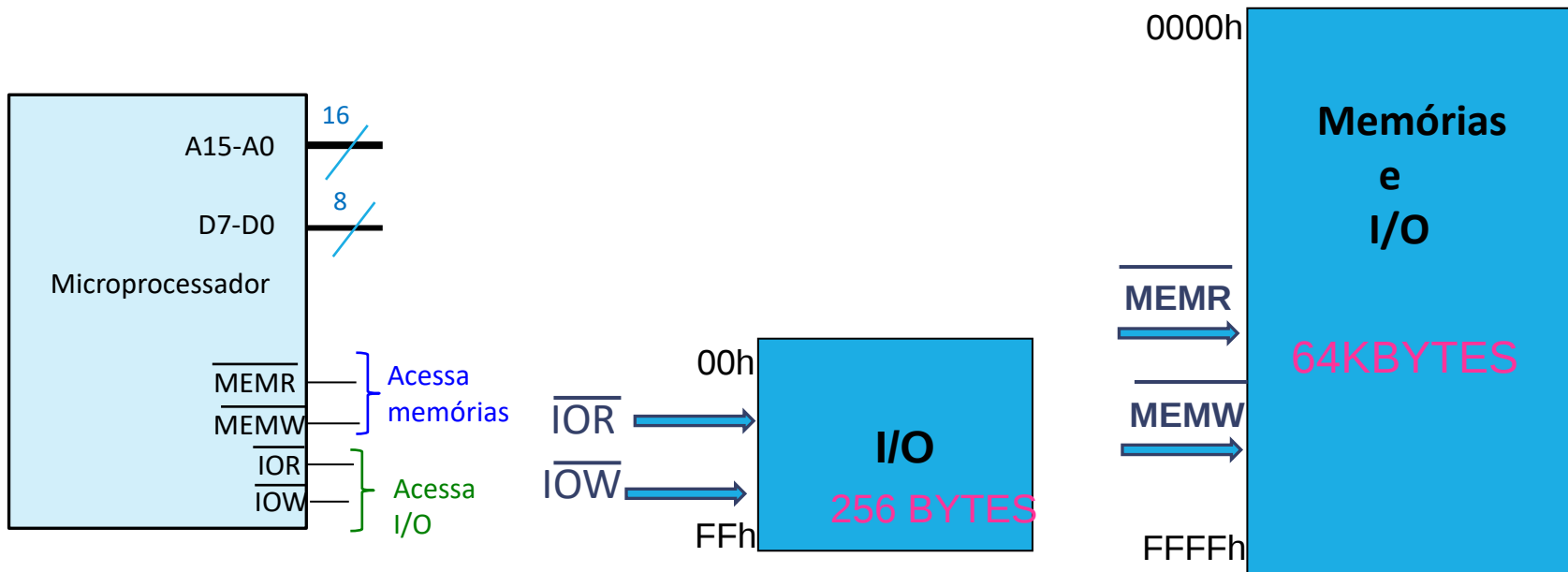
- no espaço de 64K bytes são mapeadas as memórias (sinais de controle de leitura e gravação MEMR e MEMW);
no espaço de 256 x 8 são mapeados os dispositivos de I/O (sinais de controle de leitura e gravação IOR e IOW).



Microprocessador com mapeamento em I/O Isolado

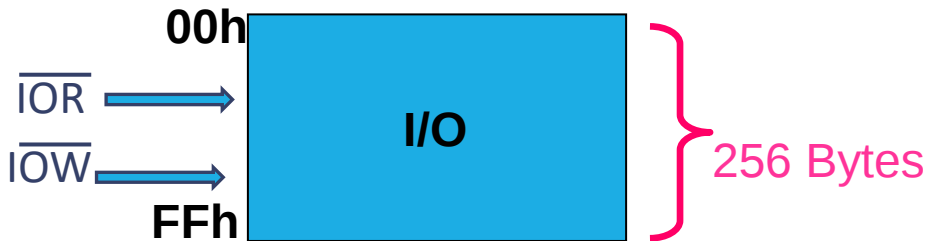
microprocessador com 16 linhas no duto de endereços e 8 linhas de duto de dados

Espaços de Mapeamento



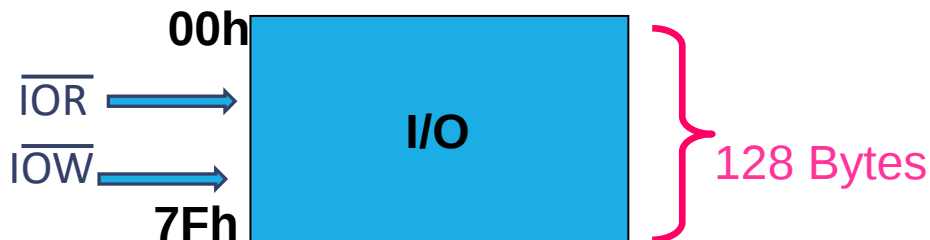
Mapeamento em I/O Isolado

- Se o microprocessador reserva um espaço de 256 bytes para endereçamento de I/O, o endereçamento desse espaço tem linhas de endereços de A7 a A0



$256 = 2^8$
8 linhas: A7 A6 A5 A4 A3 A2 A1 A0

- Se o microprocessador reserva um espaço de 128 bytes para endereçamento de I/O, o endereçamento desse espaço tem linhas de endereços de A6 a A0



$128 = 2^7$
8 linhas: A6 A5 A4 A3 A2 A1 A0

Lógica de Seleção de Memórias e Dispositivos de I/O

1. Lógica de Seleção Absoluta

- São usados todos os bits de seleção do CHIP
- É selecionado um espaço que tem exatamente as dimensões do chip. Exemplo: para memória de 1k x 8 é selecionado um bloco de 1k x 8.

Lógica de Seleção de Memórias e Dispositivos de I/O

1. Lógica de Seleção Absoluta

Vantagem: não há possibilidade de conflito de espaços de endereço.

Desvantagem: o hardware é mais complexo que o da lógica não absoluta, principalmente para interfaces que ocupam poucas posições de memória

1.a) Lógica de seleção Absoluta, Mapeamento em Memória

Para o circuito de seleção abaixo(Figura 1) são determinadas as faixas de endereço associadas a cada saída de seleção CSi (tabela 1 e 2) considerando que o microprocessador tem espaço para mapeamento em memória de 64kx8

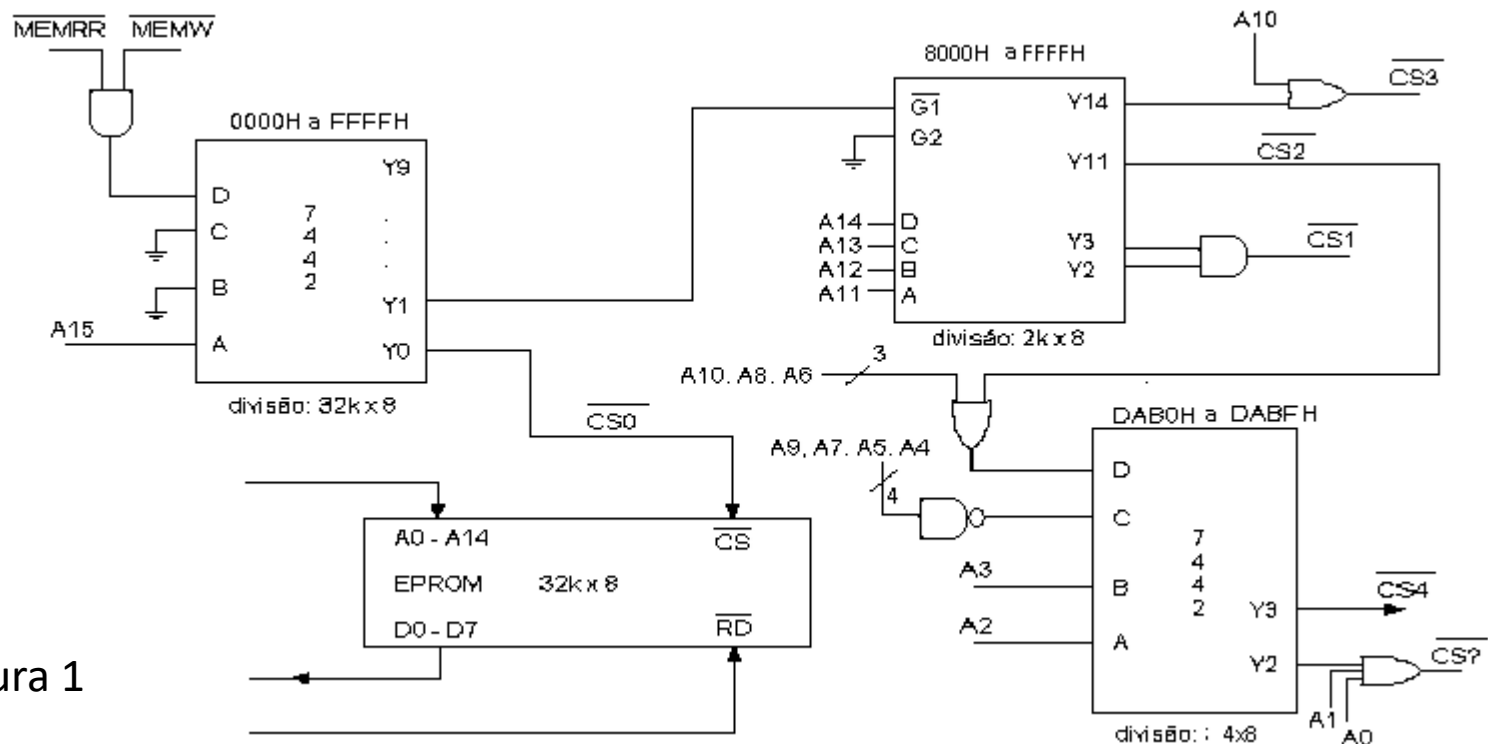


Figura 1

1a) Lógica de seleção Absoluta, Mapeamento em Memória

Tabela 1 - Faixa de endereços para cada saída de seleção da Figura 1

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
$\overline{\text{CS0}}$ (32k x 8)	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\overline{\text{CS1}}$	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1
$\overline{\text{CS2}}$	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0
	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1
$\overline{\text{CS3}}$	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	1	1	1	1	0	0	1	1	1	1	1	1	1	1	1	1
$\overline{\text{CS4}}$	1	1	0	1	1	0	1	0	1	0	1	1	1	1	0	0
	1	1	0	1	1	0	1	0	1	0	1	1	1	1	1	1
$\overline{\text{CS}}$	1	1	0	1	1	0	1	0	1	0	1	1	1	0	0	0

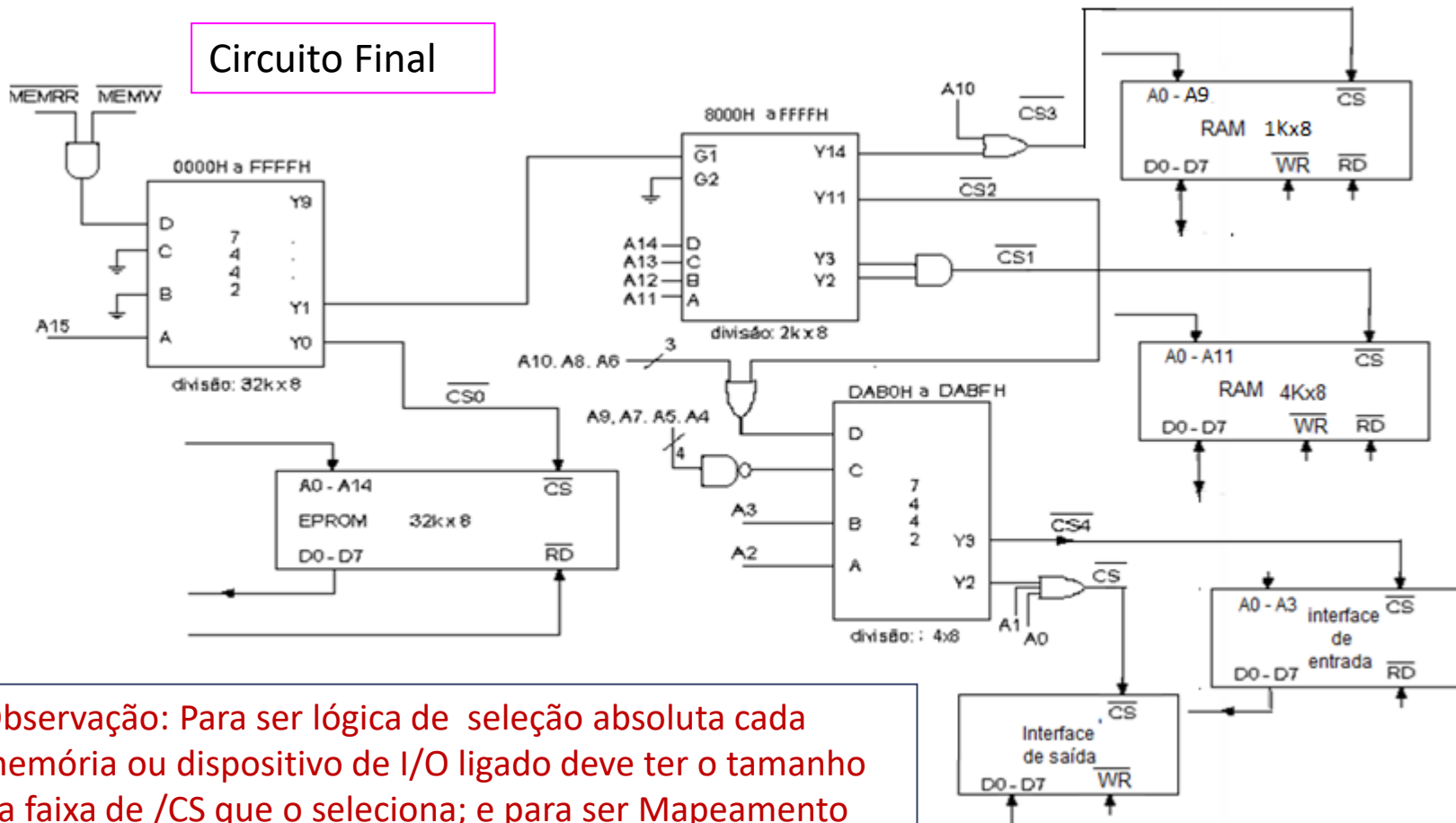
1.a) Lógica de seleção Absoluta, Mapeamento em Memória

Sinal de Seleção	Faixa em hexadecimal	Tamanho da organização	Memória ou Dispositivo I/O
$\overline{CS0}$	0000 a 7FFF	32 K x 8	EEPROM
$\overline{CS1}$	9000 a 9FFF	4K x 8	RAM
$\overline{CS2}$	D800 a DFFF	2K x 8	vazio
$\overline{CS3}$	F000 a F3FF	1K x 8	vazio
$\overline{CS4}$	DABC a DABF	4 x 8	Dispositivo de entrada
$\overline{CS}$	DAB8	1 x 8	Dispositivo de saída

Tabela 2 - Faixa de endereços em hexadecimal, referentes à tabela 1 e Figura 1

1.a) Lógica de seleção Absoluta, Mapeamento em Memória

Circuito Final



Observação: Para ser lógica de seleção absoluta cada memória ou dispositivo de I/O ligado deve ter o tamanho da faixa de /CS que o seleciona; e para ser Mapeamento em Memória os mesmos sinais de controle (/MEMR e /MEMW) selecionam memórias e dispositivos de I/O

Lógica de Seleção de Memórias e Dispositivos de I/O

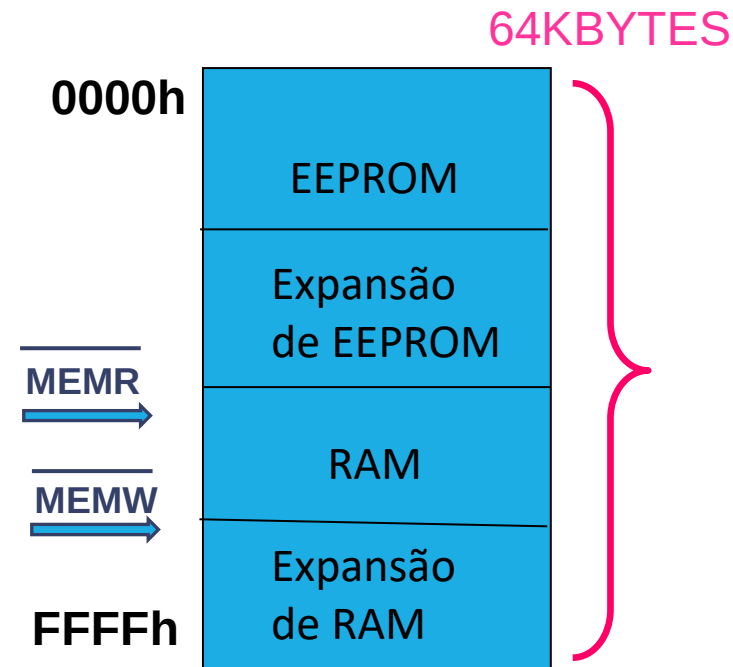
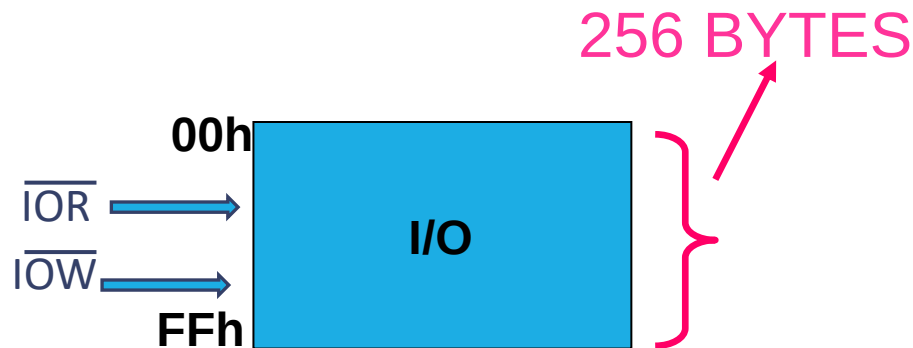
1.b) Lógica de seleção Absoluta, Mapeamento em I/O Isolado

Um microprocessador com 16 linhas de duto de endereços e 8 linhas no duto de dados, apresenta **Mapeamento em I/O isolado** quando disponibiliza um espaço de mapeamento(no caso do exemplo de 64K bytes), para mapear (alocar) memórias e outro espaço (no caso do exemplo de 256 Bytes) para mapear dispositivos de I/O. Portanto, vai ter uma lógica de seleção onde os sinais de leitura ($\overline{\text{MEMR}}$) e escrita ($\overline{\text{MEMW}}$) acessam o espaço de mapeamento de memória (64K) e sinais de leitura ($\overline{\text{IOR}}$) e de escrita ($\overline{\text{IOW}}$) para acessar o espaço de mapeamento de dispositivo de entrada e saída.

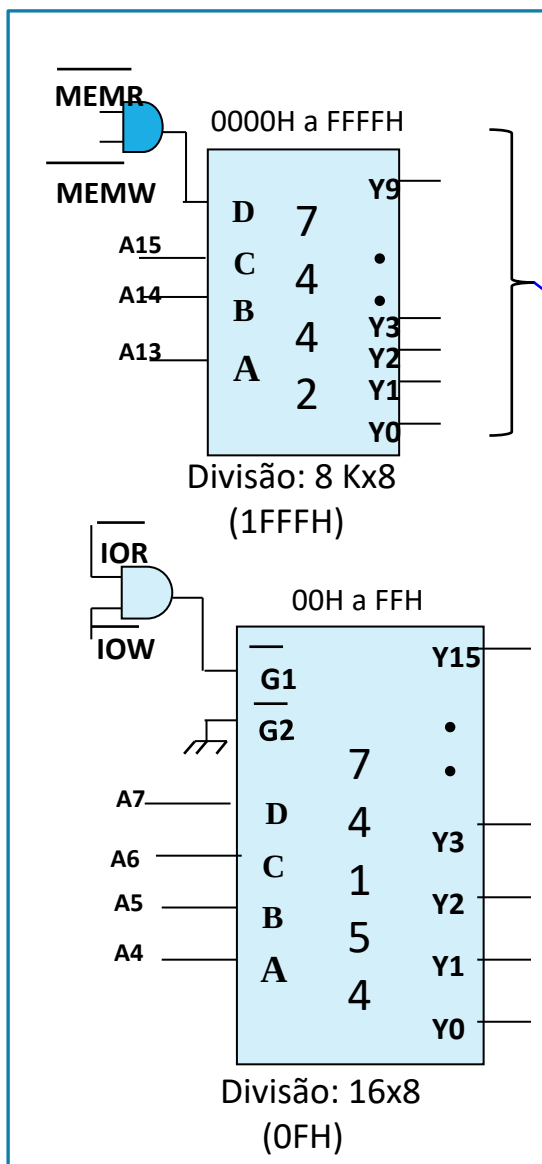
1.b) Lógica de seleção Absoluta, Mapeamento em I/O Isolado

Para esse exemplo, os dispositivos de I/O são mapeados num espaço de I/O de 256 bytes, **separado** do espaço de memória. O espaço só para mapeamento de memória é de 64k x8

Os sinais de controle são diferentes para os dois espaços



1.b) Lógica de seleção Absoluta, Mapeamento em I/O Isolado



Exemplo de Circuito Completo de Lógica de Seleção para microprocessador que apresenta espaço de 64K x 8 (linhas de A0 a A15) para mapear memória e espaço de 256 x 8 (linhas de A0 a A7) para mapear dispositivos de I/O

Saídas válida como $\overline{CS}$: Y0 a Y7

Espaço de 64K x 8

Utilizadas para ligar apenas memórias

Saídas válida como $\overline{CS}$: Y0 a Y15

Espaço de 256 x 8

Utilizadas para ligar apenas interfaces para dispositivos de I/O

1.b) Lógica de seleção Absoluta, Mapeamento em I/O Isolado

Para ser lógica de seleção absoluta só podem ser ligados memórias ou interfaces de I/O do tamanho da faixa da saída do decodificador

Nesse exemplo o microprocessador reserva espaço de 64K x 8 (linhas de A0 a A15) para mapear memória e espaço de 256 x 8 (linhas de A0 a A7) para mapear dispositivos de I/O

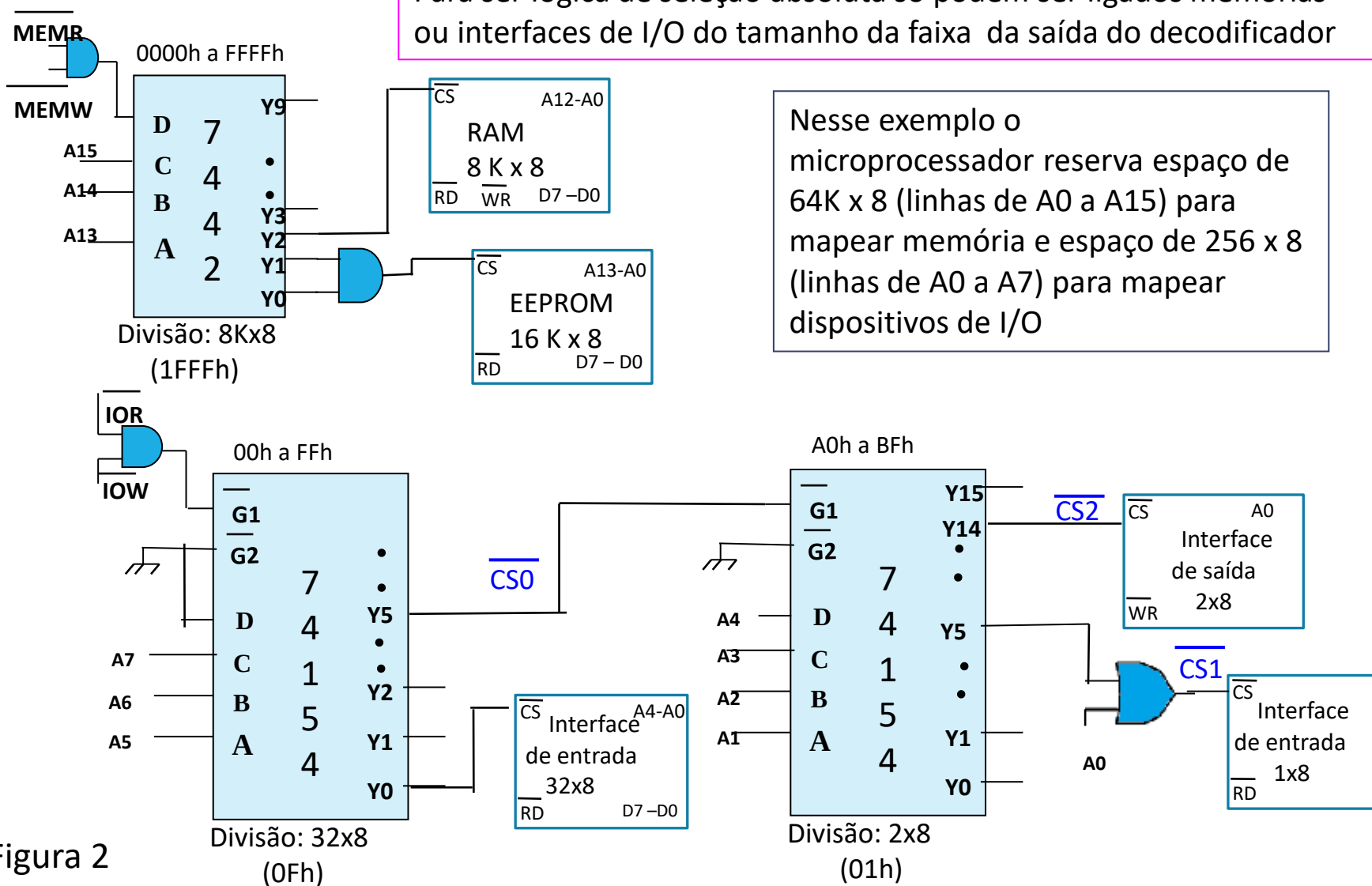


Figura 2

1.b) Lógica de seleção Absoluta, Mapeamento em I/O Isolado

Tabela 3 Mapa de endereços para o circuito de seleção de I/O da Figura 2 (slide anterior)

	A7	A6	A5	A4	A3	A2	A1	A0
$\overline{CS0}$	1 1	0 0	1 1	0 1	0 1	0 1	0 1	0 1
$\overline{CS1}$	1 1	0 0	1 1	0 0	1 1	0 0	1 1	0 0
$\overline{CS2}$	1 1	0 0	1 1	1 1	1 1	1 1	0 0	0 1

$\overline{CS0}$: A0h a BFh -> espaço: 32 x 8
 $\overline{CS1}$: AAh -> espaço: 1 x 8
 $\overline{CS2}$: BCh a BDh -> espaço: 2 x 8

Lógica de Seleção de Memórias e Dispositivos de I/O

2 Lógica de Seleção NÃO Absoluta

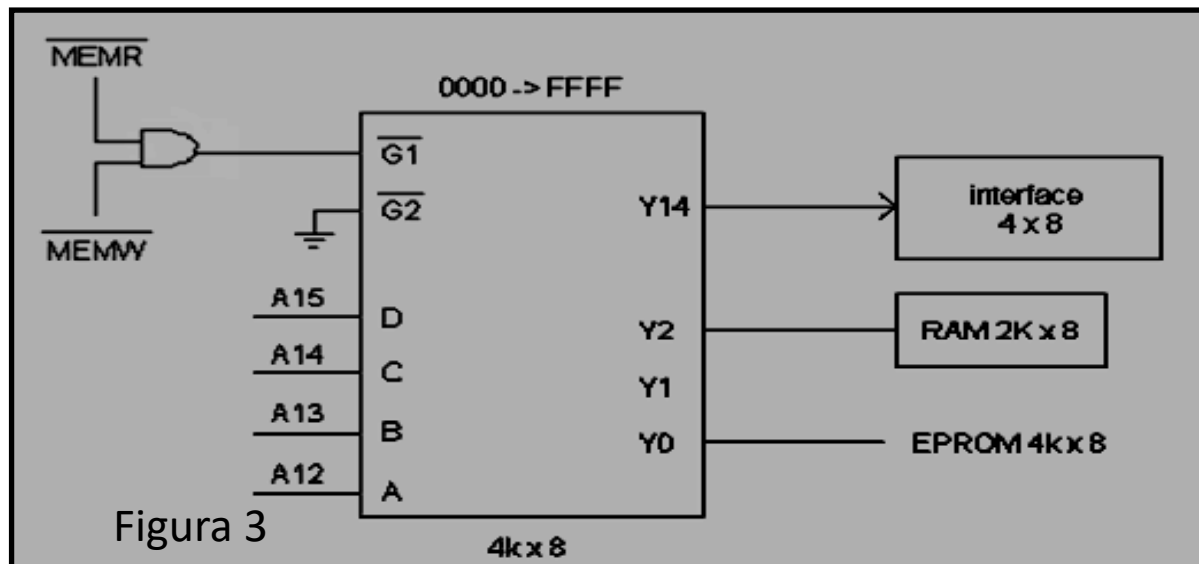
- Usa-se parte dos bits de seleção, na lógica de seleção
- Os bits de seleção não usados geram espaços de endereço extra, associados ao dispositivo, denominados **espaços de endereço fantasma (espelhos)**;

esse hardware de seleção é inadequado para área de dados sequenciais;

os endereços fantasmas não podem ser usados por outros chips, pois não são espaços livres.

Lógica de Seleção de Memórias e Dispositivos de I/O

2.a) Lógica de Seleção **NÃO Absoluta**, Mapeamento em Memória



Nesse exemplo, o microprocessador reserva 64Kx8 para mapeamento de memórias e dispositivos de I/O

No Mapeamento em Memória, memórias e Interfaces de I/O são mapeadas no mesmo espaço acessado pelos mesmos sinais de controles (MEMR e MEMW), e se tratando também de Lógica **Não Absoluta**, memórias e dispositivos de I/O **Não** precisam ter o tamanho da organização igual ao tamanho da saída de seleção. Podem ser menores, gerando espaços fantasmas

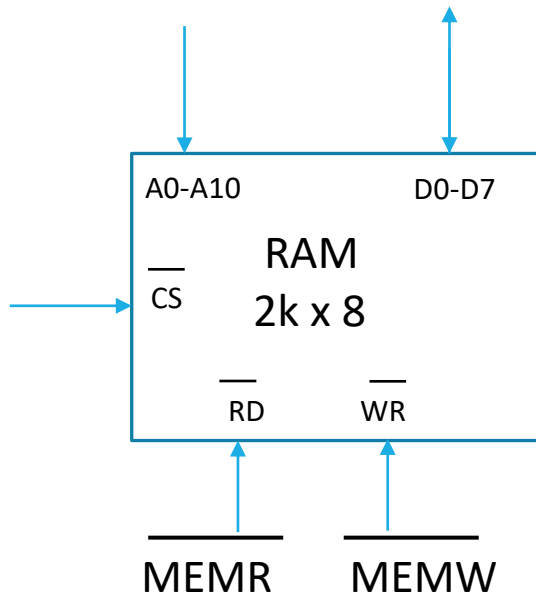
2.a) Lógica de Seleção NÃO Absoluta, Mapeamento em Memória

Tabela 4 Faixas de endereços para a figura 3

saída	Faixa da saída de 4Kx8	Tamanho da organização	Faixa fantasma	Quantidade de blocos do tamanho da organização que são fantasmas
Y0	0000h a 0FFFh	4k x 8	Não tem	0
Y2	2000h a 2FFFh	2k x 8	tem	1
Y14	E000h a EFFFh	4x8	tem	1k-1 =1023

2.a) Lógica de Seleção NÃO Absoluta, Mapeamento em Memória

Na lógica de seleção da Figura 3, a memória conectada a Y2 é de 2k x 8 que é uma organização menor do que o espaço gerado pela lógica de seleção (4kx8).



O bit de endereço **A11** não está presente na lógica de seleção (entrada do decodificador) para a memória de **2k x 8**, o que leva essa memória a ter dois espaços de endereço associados a ela. (apenas **1** bit não está presente, então **A11 = 0** e **A11 = 1** endereçam a memória, ou seja $2^1 = 2$ espaços)

2.a) Lógica de Seleção **NÃO Absoluta**, Mapeamento em Memória

Faixa de endereço que seleciona a memória RAM de 2Kx8 da Figura 3

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Y2	0	0	1	0	X	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	0	X	1	1	1	1	1	1	1	1	1	1	1

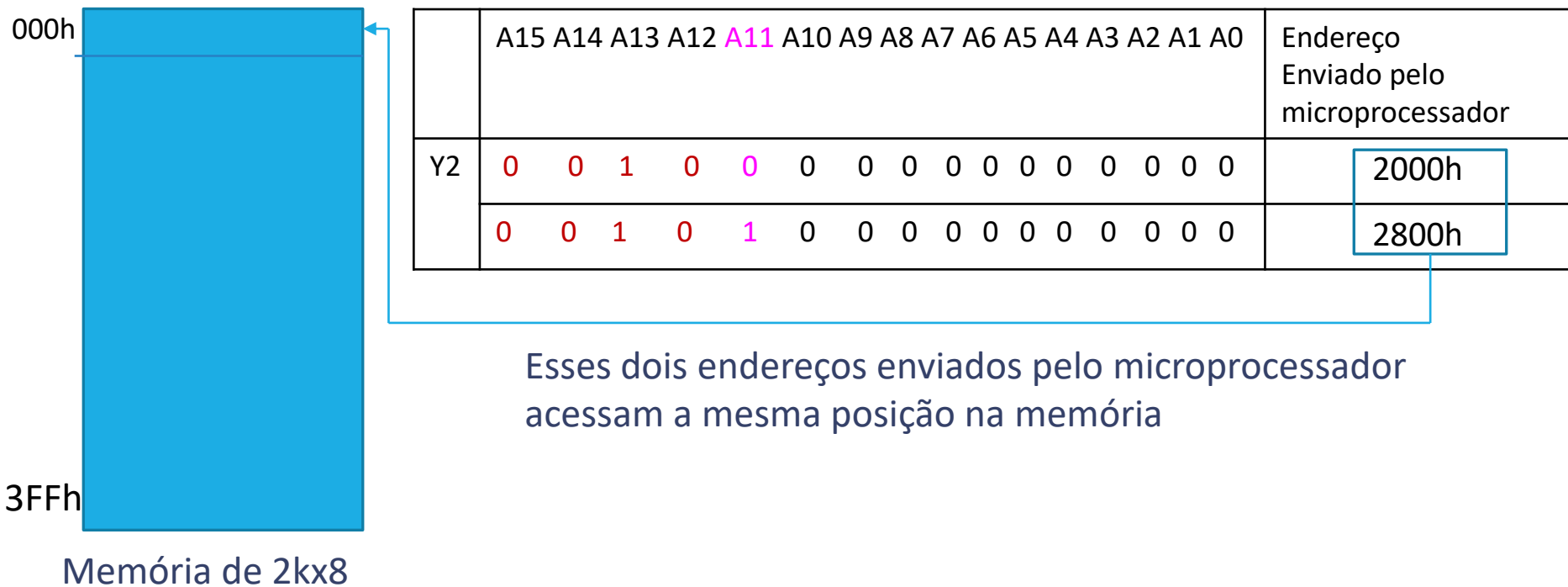
Se $X = 0$ a faixa de endereço = 2000h a 27FFh

Se $X = 1$ a faixa de endereço = 2800h a 2FFFh

- Memória de 2kx8 tem endereços no chip de 000h a 7FFh
- Um dos espaços de 2k x 8 é denominado de **espaço fantasma (ou espelho)**.
- **Qualquer uma** das duas faixas de endereço pode ser escolhida como a fantasma.
- endereço 2000h e 2800h endereçam a mesma posição física do CHIP, a posição 000h

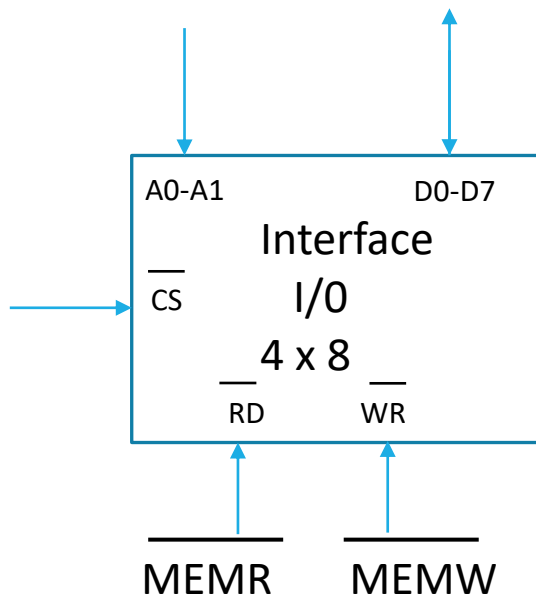
2.a) Lógica de Seleção **NÃO Absoluta**, Mapeamento em Memória

Representação do conteúdo da memória RAM de 2Kx8 da Figura 3



2.a) Lógica de Seleção **NÃO Absoluta**, Mapeamento em Memória

Na lógica de seleção da Figura 3, a memória interface conectada a Y14 é de 4x 8 que é uma organização menor do que o espaço gerado pela lógica de seleção(4kx8).



Os bits de endereço **A11 a A2** (de A2 a A11 são **10** linhas de endereços) não estão presentes na lógica de seleção (entrada do decodificador) para a interface de **4 x 8**, o que leva essa interface a ter **2¹⁰** espaços de endereço associados a ela. (então, 1k endereços endereçam essa interface)

2.a) Lógica de Seleção **NÃO Absoluta**, Mapeamento em Memória

Faixa de endereço para interface de 4 x 8 da Figura 3

Y14: E000H a EFFFH (4Kx8)

1ª. Faixa de endereço da interface: E000h a E003h

2ª. Faixa de endereço da interface: E004h a E007h

. . .

Quantas faixas de endereço estarão associadas a essa interface?

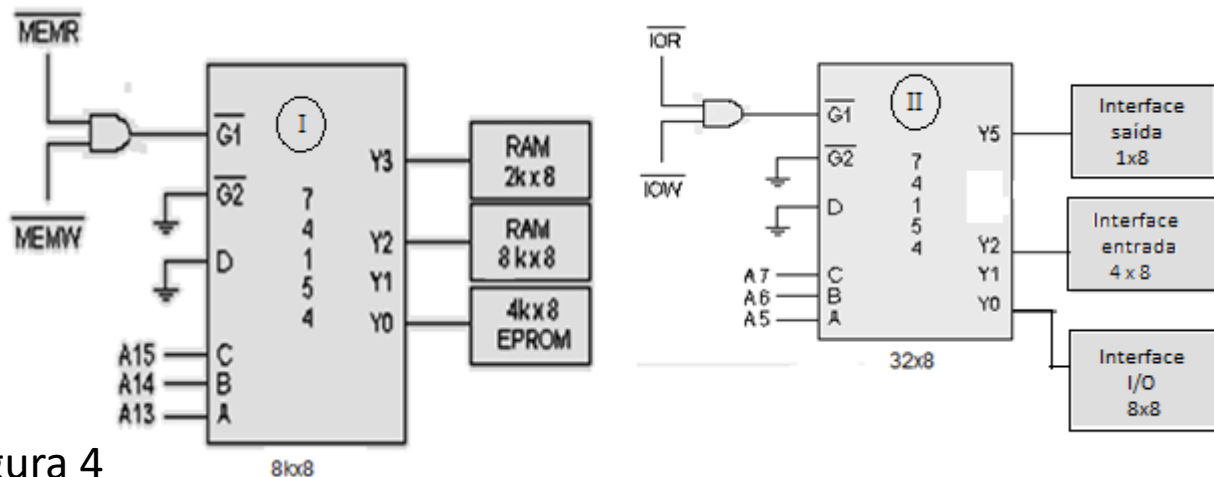
O endereçamento da interface tem linhas de A0 e A1, então de A2 a A11 são 10 linhas que não estão no endereçamento da interface e nem na seleção (entrada do decodificador)

$2^{10} = 1024$ faixas de endereços.

Onde, $1024 - 1$ (1023) são faixas fantasmas (ou espelho)

Lógica de Seleção de Memórias e Dispositivos de I/O

2.b) Lógica de Seleção NÃO Absoluta, Mapeamento em I/O isolado



Nesse exemplo, o microprocessador reserva 64Kx8 para mapeamento de memórias e 256 x 8 para mapeamento de interfaces para dispositivos de I/O

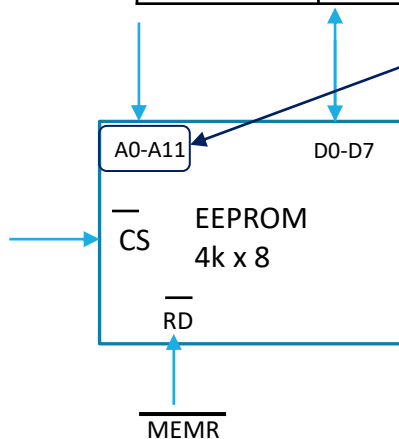
Figura 4

No Mapeamento em I/O Isolado, o microprocessador reserva espaços diferentes para mapear memórias e Interfaces de I/O. Esses espaços são acessados por sinais de controles diferentes, (ex: MEMR e MEMW para o espaço de memórias e IOR e IOW para o espaço para interfaces de dispositivos de I/O). E, se tratando também de Lógica **Não Absoluta**, memórias e dispositivos de I/O **Não** precisam ter o tamanho da organização igual ao tamanho da saída de seleção. Podem ser menores, gerando espaços fantasmas (ou espelhos).

2.b) Lógica de Seleção NÃO Absoluta, Mapeamento em I/O isolado

Faixa de endereços referente ao espaço de 64k x 8 da Figura 4

	A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0	Faixa de seleção	Tamanho do bloco
Y0 do decoder I	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0000h	1FFFh
	0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1	1FFFh	
EEPROM 4kx8	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		0FFFh

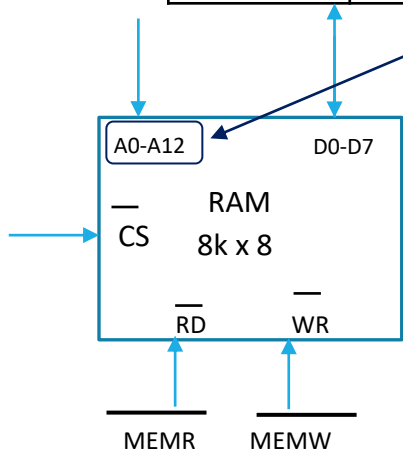


A faixa da saída Y0 do decodificador I que seleciona a EEPROM é de 0000h a 1FFFh . E a EEPROM(4kx8) tem tamanho 0FFFh
 Portanto, a EEPROM pode ocupar a faixa de 0000h a 0FFFh e o restante pode ser faixa fantasma, de 1000h a 1FFFh.
 Como apenas uma linha de endereço (A12) enviado pelo microprocessador não está nem na seleção (entrada do decodificador) nem no endereçamento da EEPROM (de A0 a A11), $2^1 = 2$ é o número de faixas do tamanho da EEPROM (4kx8). Então, $2-1=1$ é a quantidade de faixa fantasma (ou espelho) gerada.

2.b) Lógica de Seleção NÃO Absoluta, Mapeamento em I/O isolado

Faixa de endereços referente ao espaço de 64k x 8 da Figura 4

	A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0	Faixa de seleção	Tamanho do bloco
Y2 do decoder I	0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0	4000h	1FFFh
	0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1	5FFFh	
RAM 8kx8	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		1FFFh



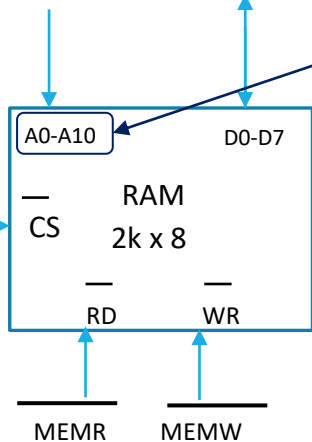
A faixa da saída Y2 do decodificador I que seleciona a RAM é de 4000h a 5FFFh e a RAM(8kx8) tem tamanho 1FFFh

Como a RAM tem o tamanho da faixa que a seleciona, **não são** gerados espaços fantasmas (ou espelhos).

2.b) Lógica de Seleção NÃO Absoluta, Mapeamento em I/O isolado

Faixa de endereços referente ao espaço de 64k x 8 da Figura 4

	A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0	Faixa de seleção	Tamanho do bloco
Y3 do decoder I	0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0	6000h	1FFFh
	0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	7FFFh	
RAM 2kx8	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		07FFh



A faixa da saída Y3 do decodificador I que seleciona a RAM é de 6000h a 7FFFh e a RAM (2kx8) tem tamanho 07FFh

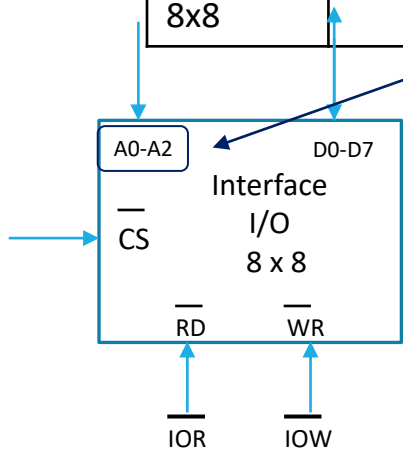
Portanto, a RAM pode ocupar a faixa de 6000h a 67FFh e o restante pode ser faixa fantasma, de 6800h a 7FFFh.

Como duas linhas de endereço (A11 e A12) enviadas pelo microprocessador não estão nem na seleção (entradas do decodificador) nem no endereçamento da RAM (de A0 a A10), $2^2 = 4$ é o número de faixas do tamanho da RAM (2kx8). Então, $4 - 1 = 3$ é a quantidade de faixas fantasmas (ou espelho) geradas.

2.b) Lógica de Seleção NÃO Absoluta, Mapeamento em I/O isolado

Faixa de endereços referente ao espaço de 256 x 8 da Figura 4

	A7	A6	A5	A4	A3	A2	A1	A0	Faixa de seleção	Tamanho do bloco
Y0 do decoder II	0	0	0	0	0	0	0	0	00h	1Fh
	0	0	0	1	1	1	1	1	1Fh	
Interface I/O 8x8							1	1	1	07h



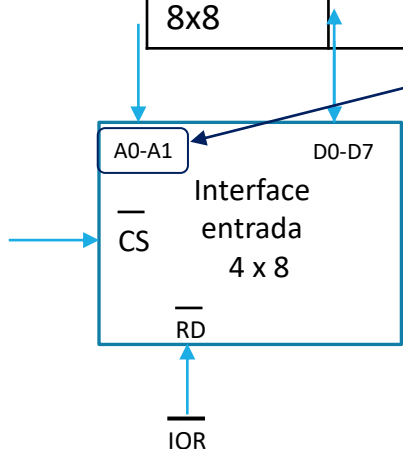
A faixa da saída Y0 do decodificador II que seleciona a Interface de I/O é de 00h a 1Fh . E essa Interface de I/O(8x8) tem tamanho 07h
Portanto, essa Interface de I/O pode ocupar a faixa de 00h a 07h e o restante pode ser faixa fantasma, de 08h a 1Fh.

Como as duas linhas de endereço (A4 e A3) enviadas pelo microprocessador não estão nem na seleção (entradas do decodificador) nem no endereçamento da interface (de A0 a A2), $2^2 = 4$ é o número de faixas do tamanho da interface (8x8). Então, $4-1 = 3$ é a quantidade de faixas fantasma (ou espelho) gerada.

2.b) Lógica de Seleção NÃO Absoluta, Mapeamento em I/O isolado

Faixa de endereços referente ao espaço de 256 x 8 da Figura 4

	A7	A6	A5	A4	A3	A2	A1	A0	Faixa de seleção	Tamanho do bloco
Y2 do decoder II	0	1	0	0	0	0	0	0	40h	1Fh
	0	1	0	1	1	1	1	1	5Fh	
Interface I/O 8x8							1	1		03h



A faixa da saída Y2 do decodificador II que seleciona a Interface de I/O é de 40h a 5Fh . E essa Interface de I/O(4x8) tem tamanho 03h.

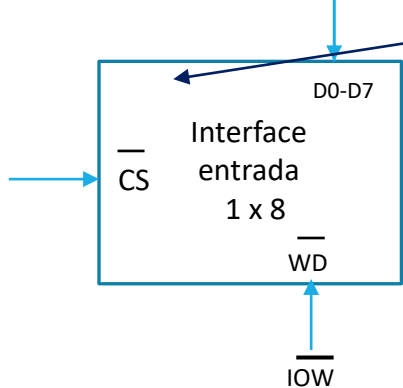
Portanto, essa Interface de I/O pode ocupar a faixa de 40h a 43h e o restante pode ser faixa fantasma, de 44h a 5Fh.

Como as tres linhas de endereço (A4 , A3 e A2) enviadas pelo microprocessador não estão nem na seleção (entradas do decodificador) nem no endereçamento da interface (de A0 a A1), $2^3 = 8$ é o número de faixas do tamanho da interface (4x8). Então, $8-1 = 7$ é a quantidade de faixas fantasma (ou espelho) gerada.

2.b) Lógica de Seleção NÃO Absoluta, Mapeamento em I/O isolado

Faixa de endereços referente ao espaço de 256 x 8 da Figura 4

	A7	A6	A5	A4	A3	A2	A1	A0	Faixa de seleção	Tamanho do bloco
Y5 do decoder II	1	0	1	0	0	0	0	0	A0h	1Fh
	1	0	1	1	1	1	1	1	BFh	
Interface I/O 8x8									Não tem	00h



A faixa da saída Y5 do decodificador II que seleciona a Interface de I/O é de A0h a BFh . E essa Interface de I/O (1x8) ocupa apenas uma posição, podendo ocupar a posição A0H e então de A1h a BFh seriam faixas fantasmas.

Como as cinco linhas de endereço (A4 , A3 , A2 , A1 e A0) enviadas pelo microprocessador não estão nem na seleção (entradas do decodificador) nem no endereçamento da interface $2^5 = 32$ é o número de faixas do tamanho da interface (1x8). Então, $32-1=31$ é a quantidade de faixas fantasma (ou espelho) gerada.

2.b) Lógica de Seleção NÃO Absoluta, Mapeamento em I/O isolado

Tabela 5 Faixas de endereço para cada saída de seleção do espaço de mapeamento apenas de memória da Figura 4

Decoder	saída	Faixa da saída	Tamanho da organização	Faixa fantasma	Quantidade de blocos fantasmas
I	Y0	0000h a 1FFFh	4k x 8	tem	1
	Y2	4000h a 5FFFh	8k x 8	Não tem	0
	Y3	6000h a 7FFFh	2kx8	tem	3
II	Y0	00h a 1Fh	8x8	tem	3
	Y2	40h a 5Fh	4x8	tem	31
	Y5	A0h a BFh	1x8	tem	7

Lógica de Seleção de Memórias e Dispositivos de I/O

Tipos de Lógica de Seleção (Resumo):

Lógica de Seleção Absoluta:

- Mapeamento em Memória com **Alinhamento de Memória**;
- Mapeamento em I/O isolado com **Alinhamento de Memória**;

Lógica de Seleção NÃO Absoluta:

- Mapeamento em Memória com **Alinhamento de Memória**;
- Mapeamento em I/O isolado com **Alinhamento de Memória**;

Lógica de Seleção de Memórias e Dispositivos de I/O

Memória Alinhada

A memória é dita “**alinhada**” com o endereço inicial:

- quando os bits de seleção apresentam o mesmo valor para qualquer posição da memória, o que simplifica a lógica de seleção
- se os **bits de endereçamento do chip** tem valor **zero** para o endereço inicial.

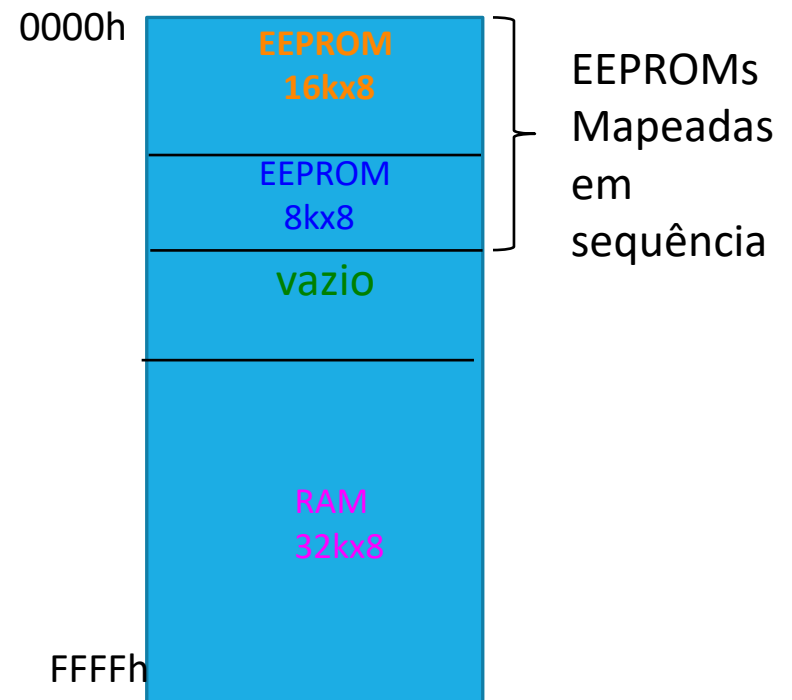
Exemplos

Exemplo 1:

Fazer o mapeamento de um bloco de memórias no espaço de um microprocessador de 16 linhas de endereços e 8 bits de dados:

- Decodificação Absoluta
- Alinhamento das Memórias
- 3 memórias em sequência

- ROM 16 k x 8
- ROM 8 k x 8
- RAM 32 k x 8



Exemplo 1(continuação):

Lógica de Seleção com Decodificação Absoluta com alinhamento das memórias

Lógica de Endereçamento do µP – Endereço de dados																	Memória		
Tipo	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Início		Fim
ROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000h	16k	
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1			3FFFh
ROM	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000h	8 k	
	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1			5FFFh
Vazio	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	6000h	8 k	
	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			7FFFh
RAM	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000h	32 k	
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			FFFFh

Exemplo 1(continuação):

Lógica de Seleção com Decodificação Absoluta com alinhamento das memórias

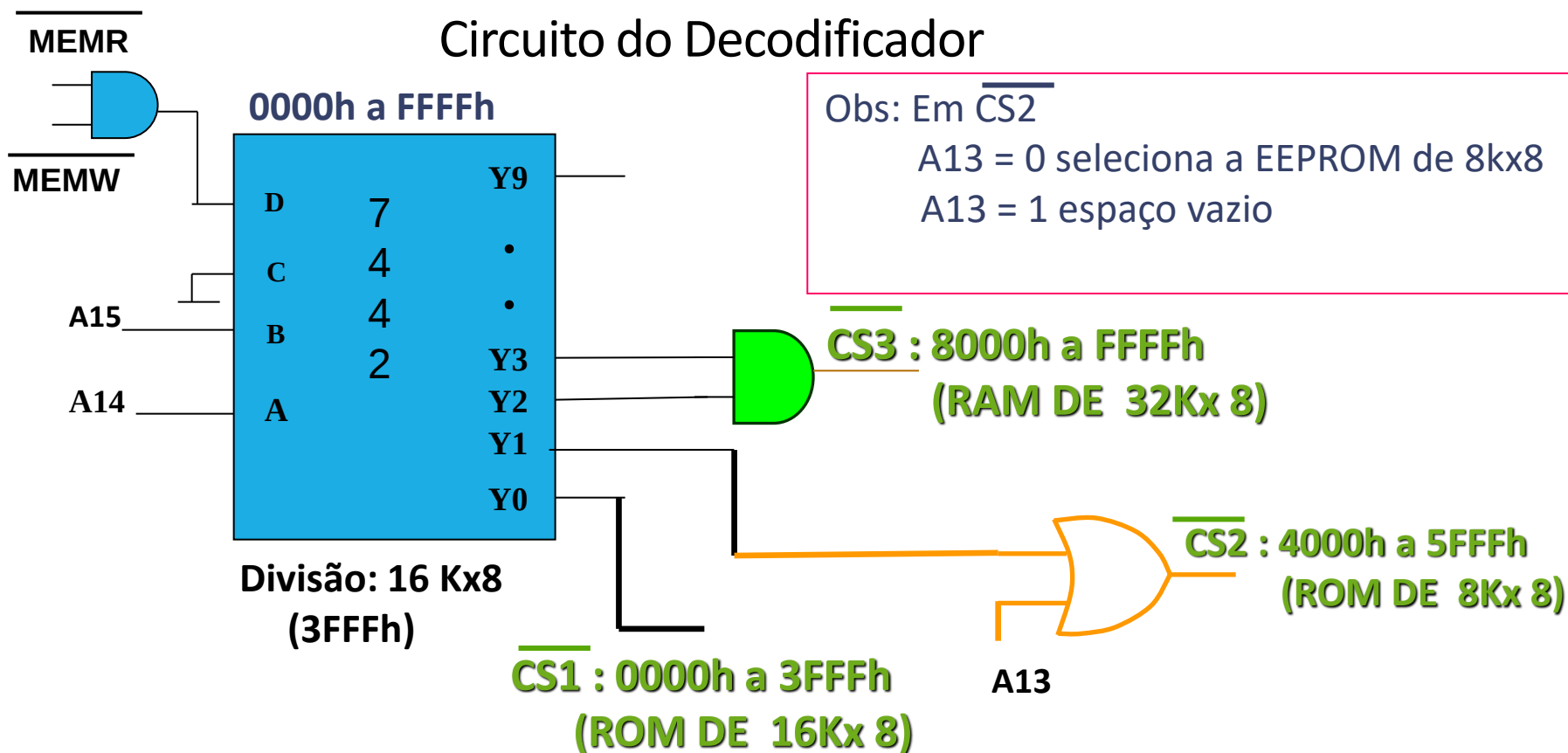
Mapeamento da Memória



Exemplo 1(continuação):

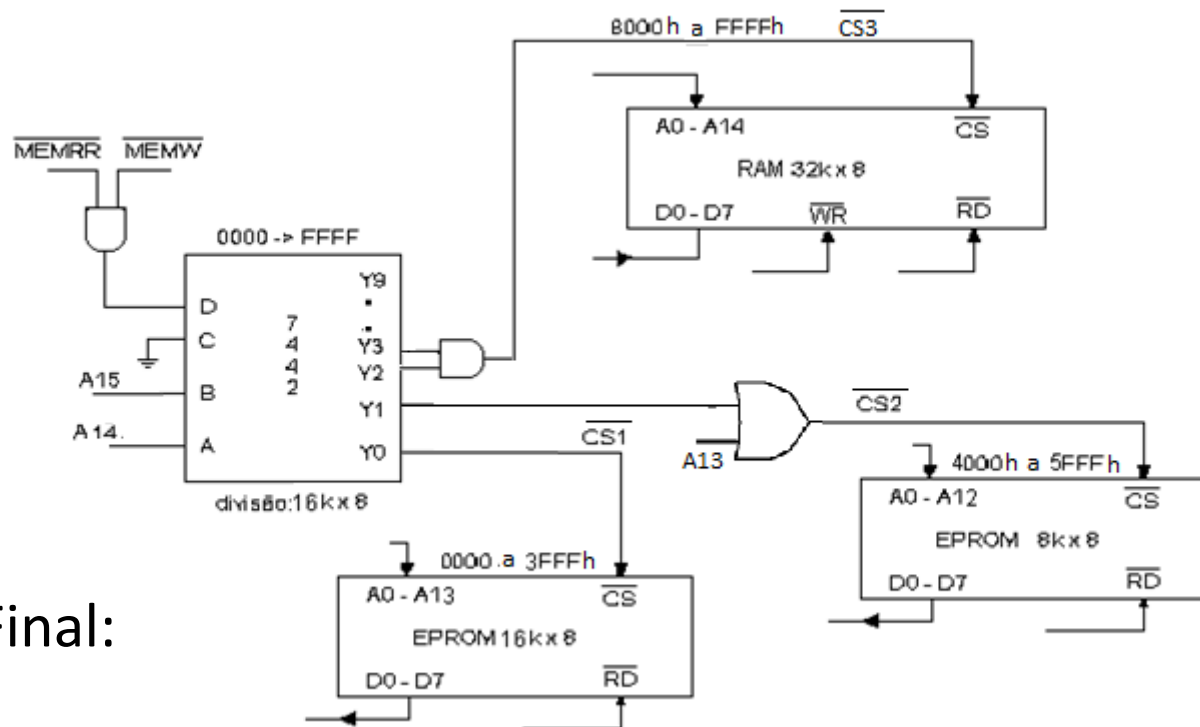
Lógica de Seleção com Decodificação Absoluta com alinhamento das memórias

Circuito do Decodificador



Exemplo 1(continuação):

Lógica de Seleção com Decodificação Absoluta com alinhamento das memórias



Circuito Final:

Exemplo 2:

Fazer o mapeamento de um bloco de memórias no espaço de um microprocessador de 16 linhas de endereços e 8 bits de dados:

- Decodificação Absoluta
- Sem alinhamento das Memórias
- 5 memórias em sequência:
 - ROM 16 k x 8
 - ROM 8 k x 8
 - RAM 32 k x 8
 - 2 RAMs de 4kx8

Exemplo 2 (continuação):

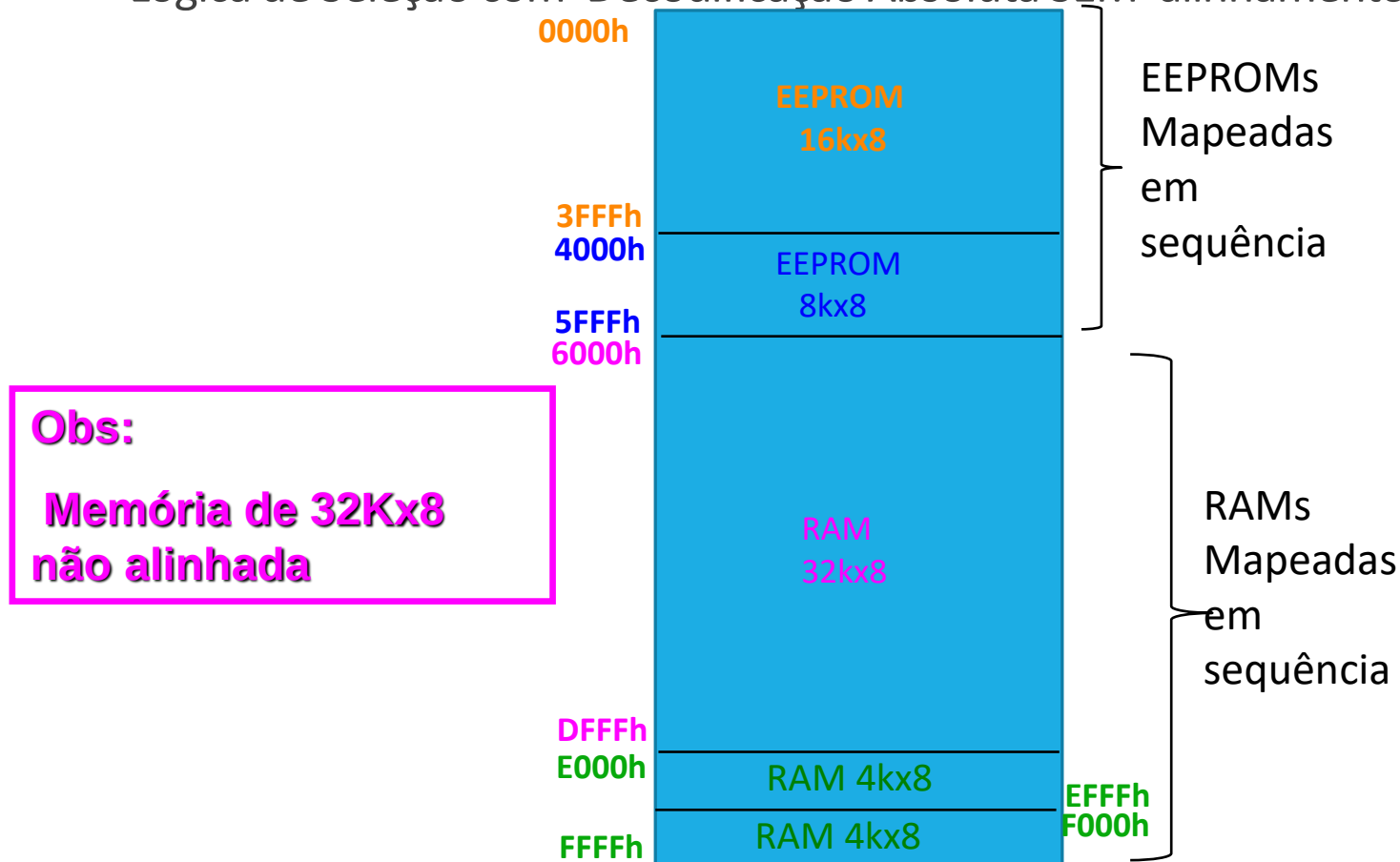
Lógica de Seleção com Decodificação Absoluta SEM alinhamento das memórias

Lógica de Endereçamento do μ P – Endereço de dados																	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
ROM	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000	8 k	
	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1			5FFF
RAM	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	6000	32k	
	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1			DFFF
RAM	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	E000	4 k	
	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1			EFFF
RAM	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	F000	4 k	
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			FFFF

- CI de memória RAM 32K x 8, nesse caso, não será endereçado na sequência normal (0000h – 7FFFh). Mesmo assim, TODOS os seus endereços serão utilizados.

Exemplo 2 (continuação):

Lógica de Seleção com Decodificação Absoluta SEM alinhamento das memórias



Como é endereçada a Memória RAM 32K x 8 não alinhada do Exemplo 2?

Memória de 32K x 8

tem $2^5 \times 2^{10} = 15$ linhas de endereçamento de A0 a A14

END. INICIAL:	A14	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	0
END. FINAL :															
7FFFH:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Representação da RAM
32Kx 8

0000h

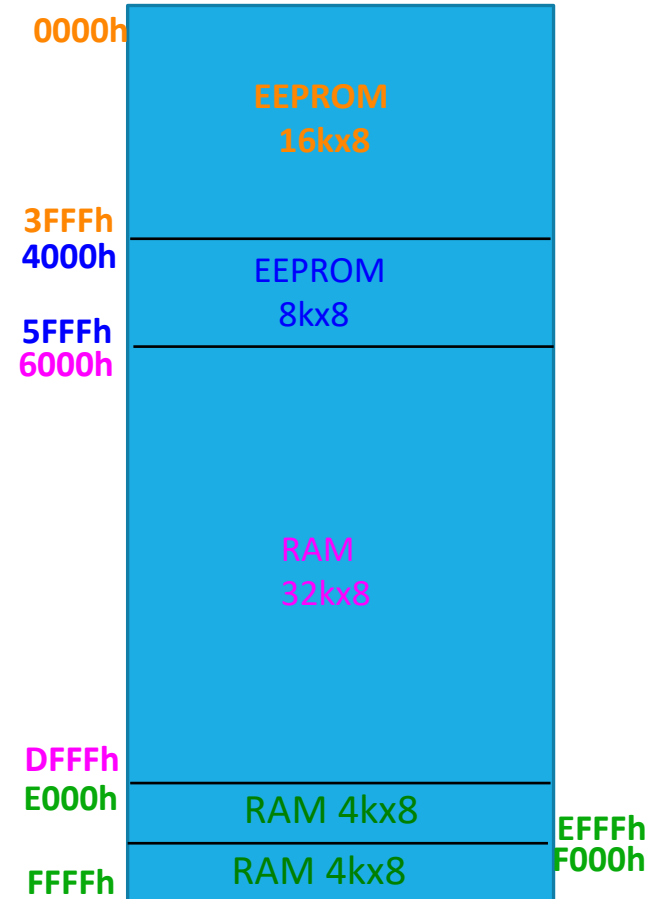
7FFFh

Como é endereçada a Memória RAM 32K x 8 não alinhada do Exemplo 2?

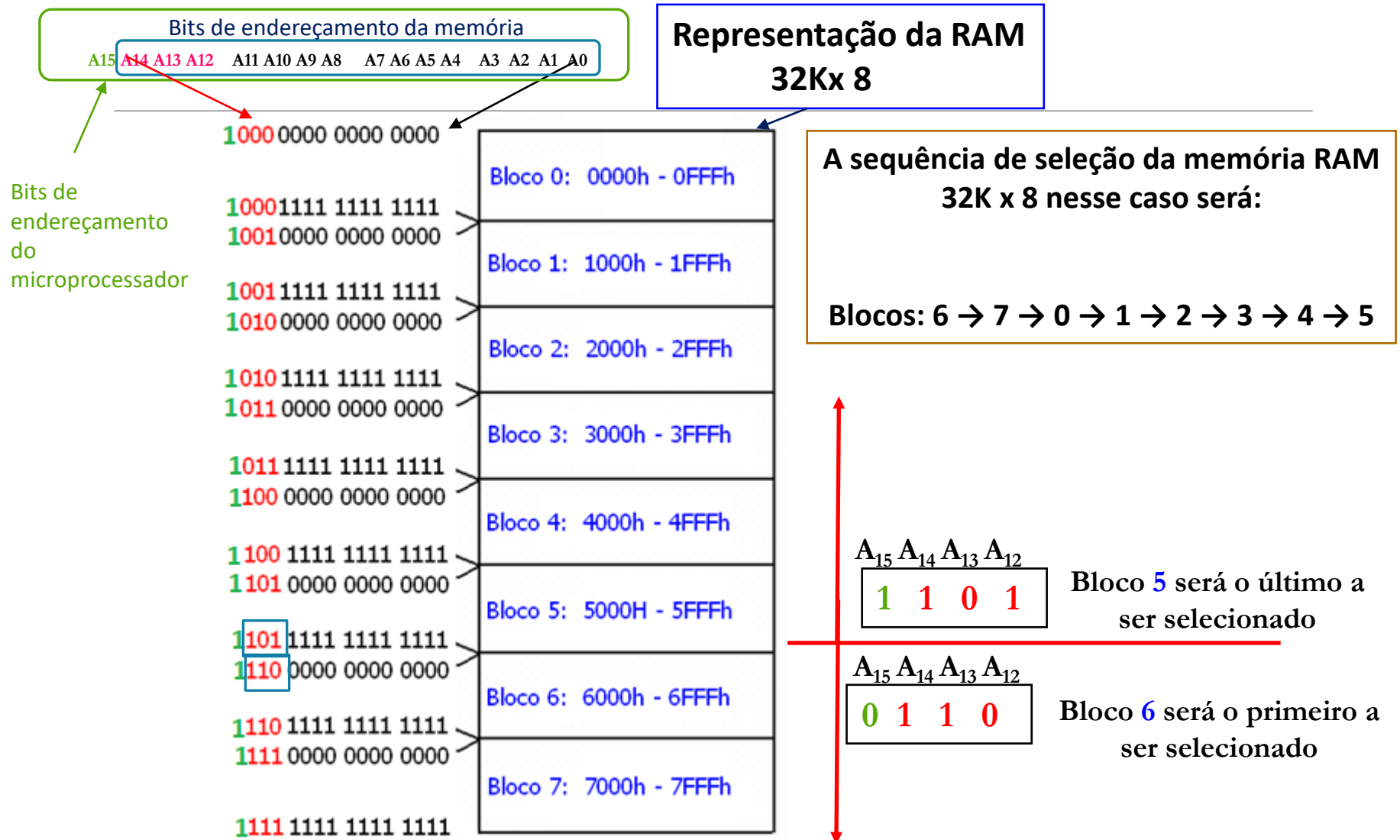
Memória não alinhada - RAM de 32Kx8

	A14	A13				
END. INICIAL: 6000h	0	1	1	0	0000	0000 0000
END. FINAL : DFFFh	1	1	0	1	1111	1111 1111

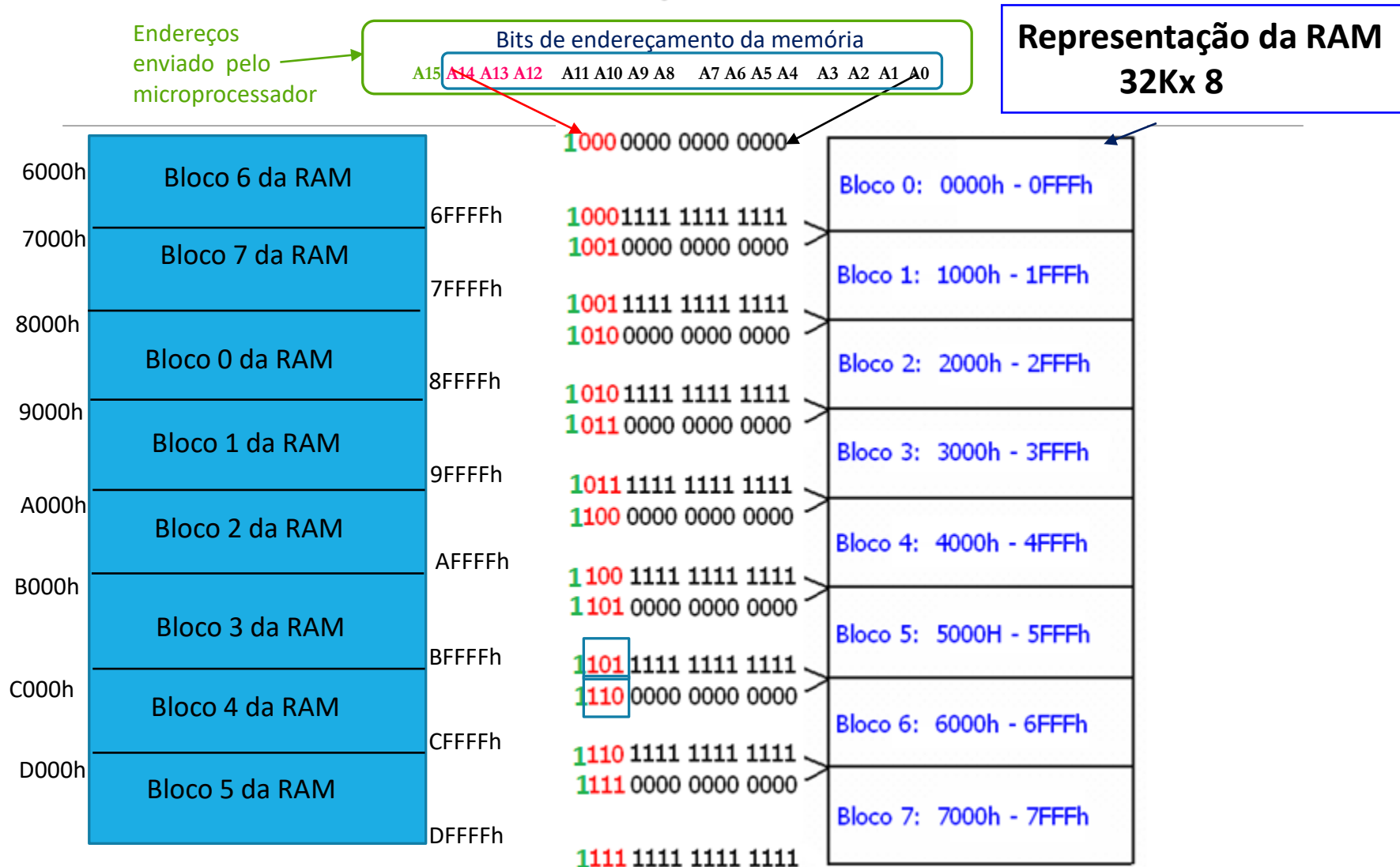
- A13 e A14 são bits de endereçamento da memória, e apresentam valor 1, para o end. Inicial, mas deveriam ser 0 para acessar a primeira posição da memória.
- Os **bits de seleção** para endereço inicial e final **NÃO** são os mesmos.



Como é endereçada a Memória RAM 32K x 8 não alinhada do Exemplo 2?

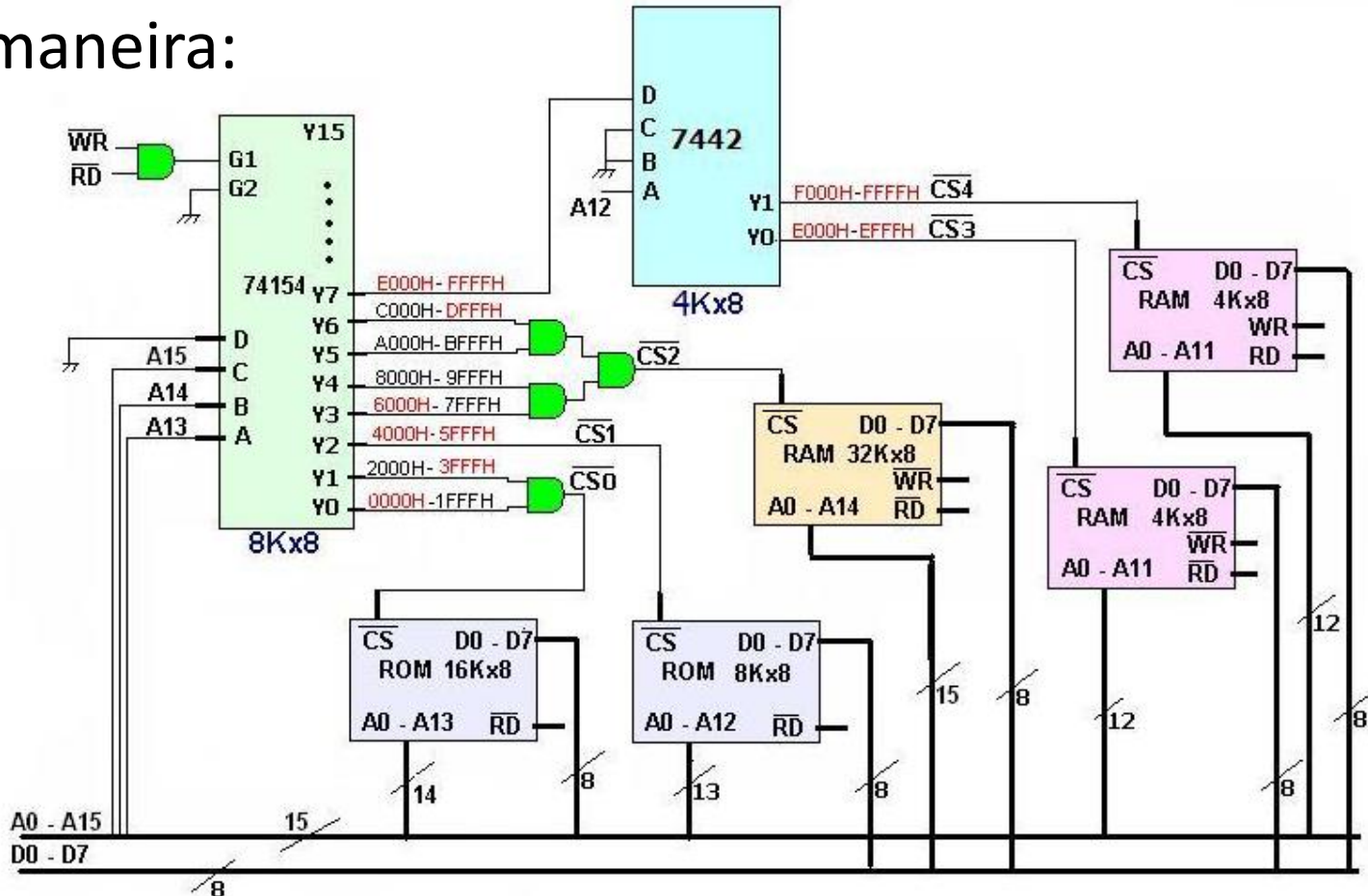


Como é endereçada a Memória RAM 32K x 8 não alinhada do Exemplo 2?



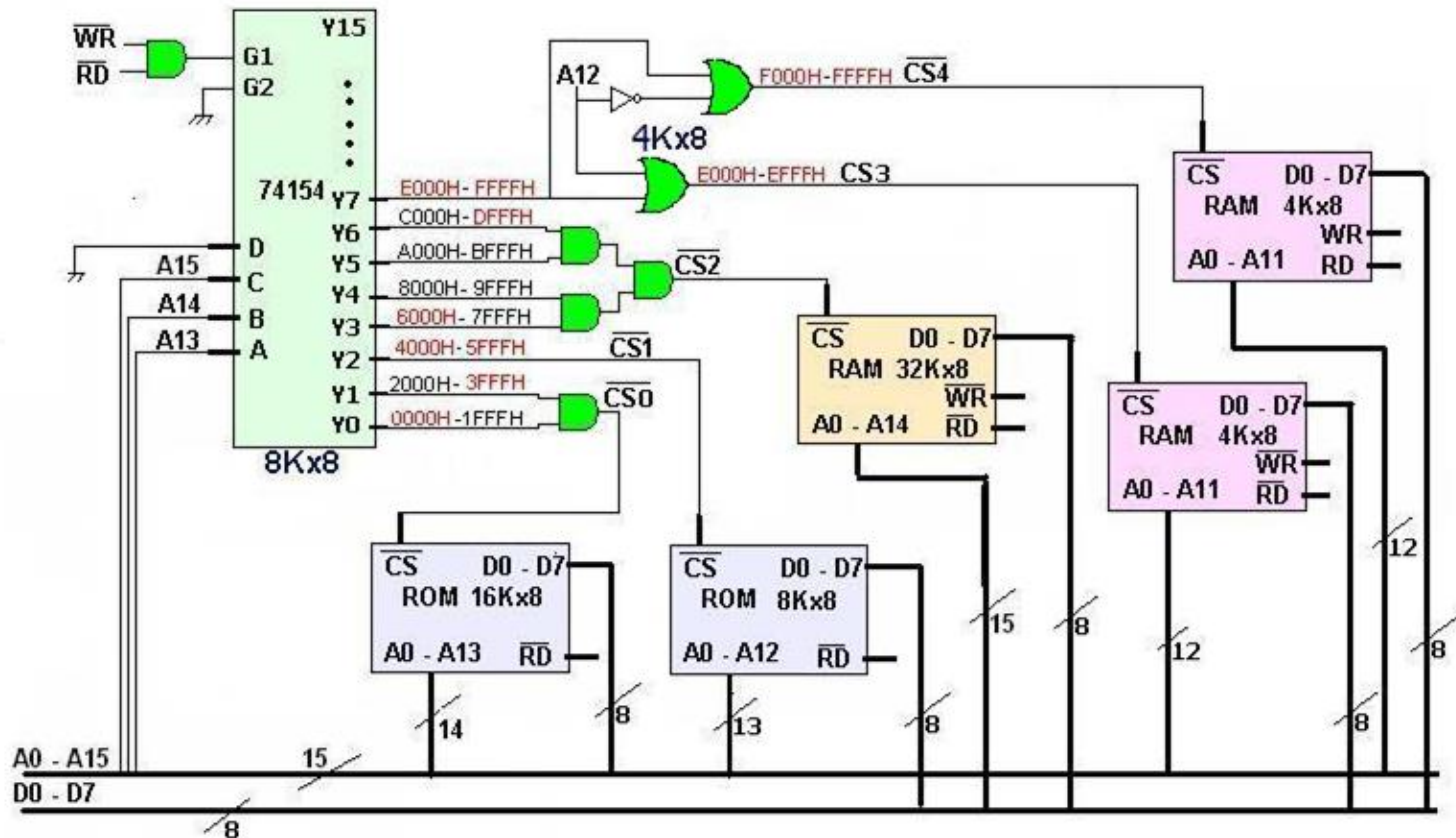
Implementação do Circuito de mapeamento do Exemplo 2:

1ª maneira:



Exemplo de Implementação INCORRETA:

Obs: a lógica de seleção deve ser feita com decodificadores e **NÃO** com uma quantidade grande de portas lógicas OR e AND



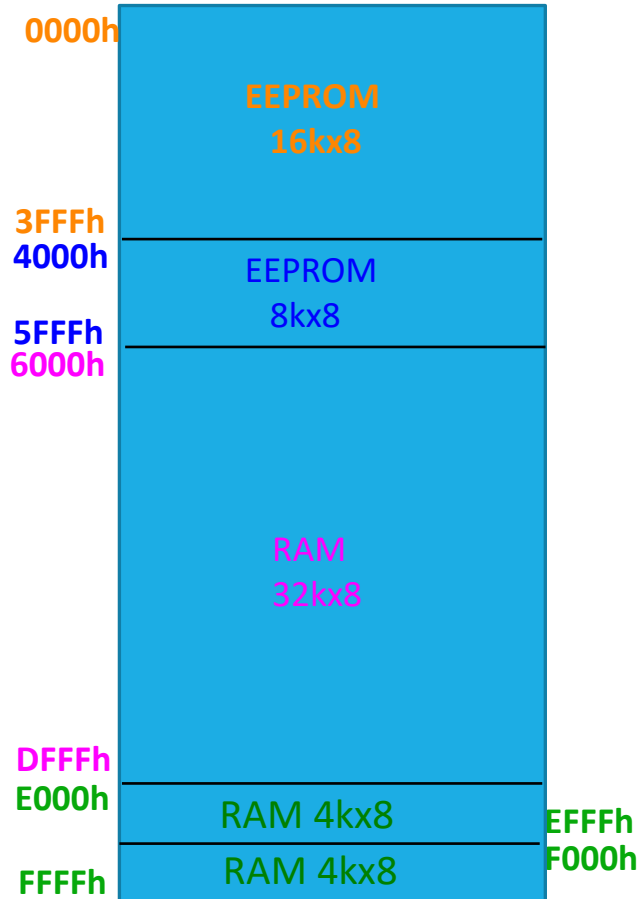
Exemplo 2: Faixas do mapeamento com a Memória RAM 32K x 8 desalinhada (decodificação Absoluta)

- Para alinhar a memória, a lógica de seleção deve ser feita de outra maneira. Uma maneira seria colocar as duas memórias de 4 K x 8 antes da de 32Kx8. Dessa forma, quando o microprocessador acessa o bloco da RAM de 32Kx8, o endereço inicial apresenta valor **zero** para os **bits de endereçamento do chip**.

Lógica de Endereçamento do µP – Endereço de dados																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
ROM	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000	8 k	
	0	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1			5FFF
RAM	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	6000	4 k	
	0	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1			6FFF
RAM	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	7000	4 k	
	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	32k	
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			FFFF

Opção de alinhar a Memória RAM 32K x 8 do Exemplo 2

Memória não alinhada
RAM de 32Kx8



Memória alinhada
RAM de 32Kx8

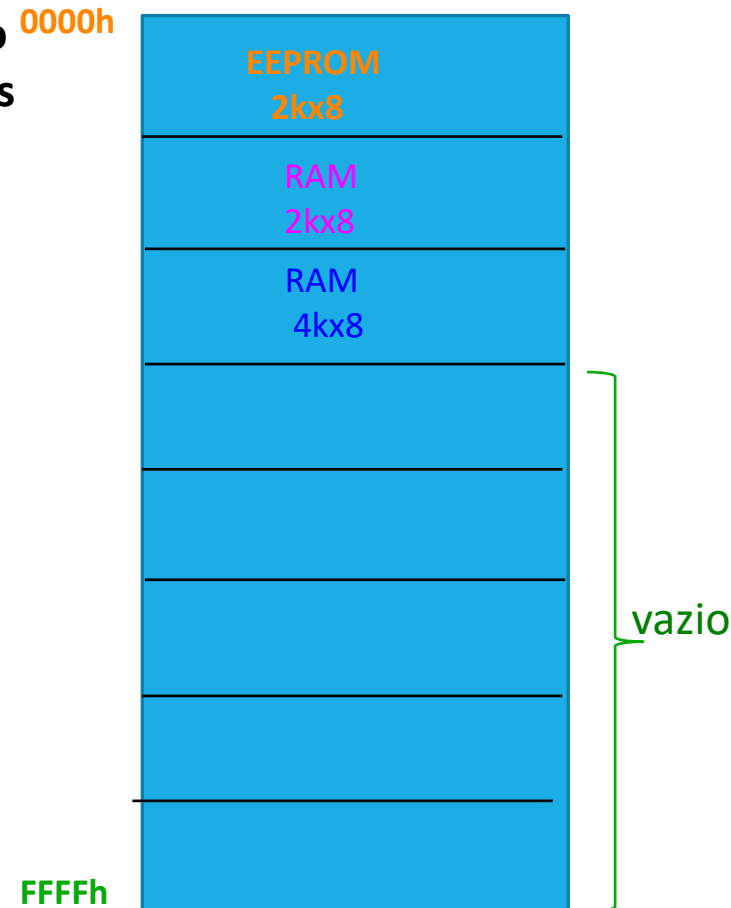


Exemplo 3:

Fazer o mapeamento de um bloco de memórias no espaço de um microprocessador de 16 linhas de endereços e 8 bits de

dados utilizando um decodificador 7442:

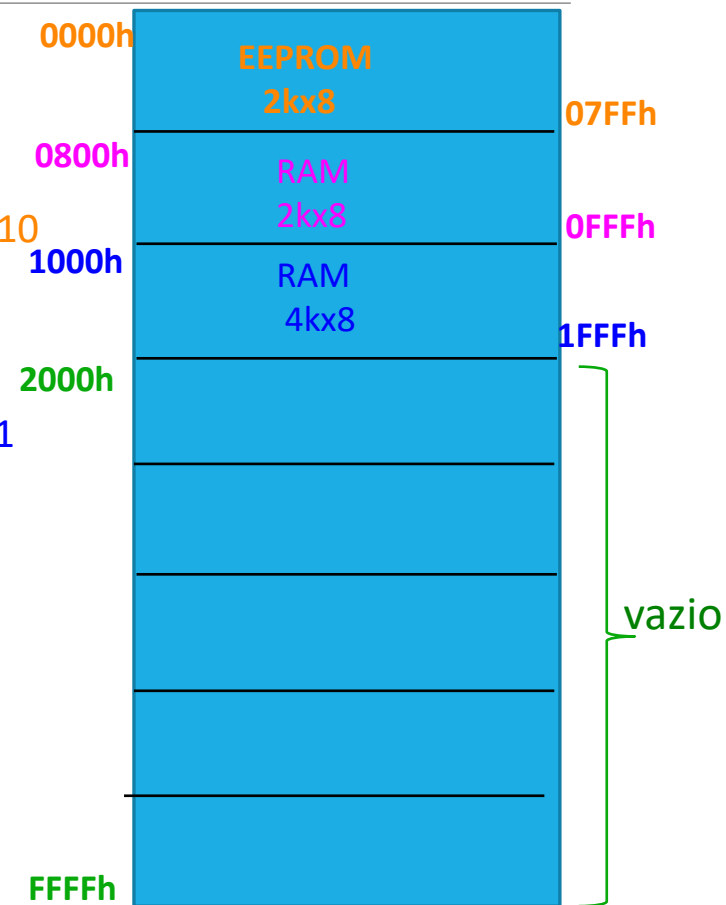
- Decodificação NÃO Absoluta
- Com alinhamento das Memórias
- 3 memórias em sequência:
 - ROM 2 k x 8: bloco =
 - RAM 2 k x 8
 - RAM 4 k x 8



Exemplo 3 (continuação):

Lógica de Seleção com 3 Memórias em sequência:

- ROM 2 k x 8 : $2^1 \times 2^{10} \Rightarrow$ 11 linhas de endereços de A0 a A10
bloco 07FFh
- RAM 2 k x 8 : bloco 07FFh
- RAM 4 k x 8: $2^2 \times 2^{10} \Rightarrow$ 12 linhas de endereços de A0 a A11
bloco de 0FFFh



Exemplo 3 (continuação):

Lógica de Seleção com Decodificação Absoluta e memórias em sequência
Faixa de endereços das memórias

Lógica de Endereçamento do μP – Endereço de dados																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	2 k	
	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1			07FF
RAM	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0800	2 k	
	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1			0FFF
RAM	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000	4 k	
	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1			1FFF

A15 = A14 = A13 = X

A12 = 1 e A11 = X
Seleciona RAM
de 4kx8

A12 = 0 e A11 = 0 seleciona ROM
A12 = 0 e A11 = 1 seleciona RAM de 2kx8

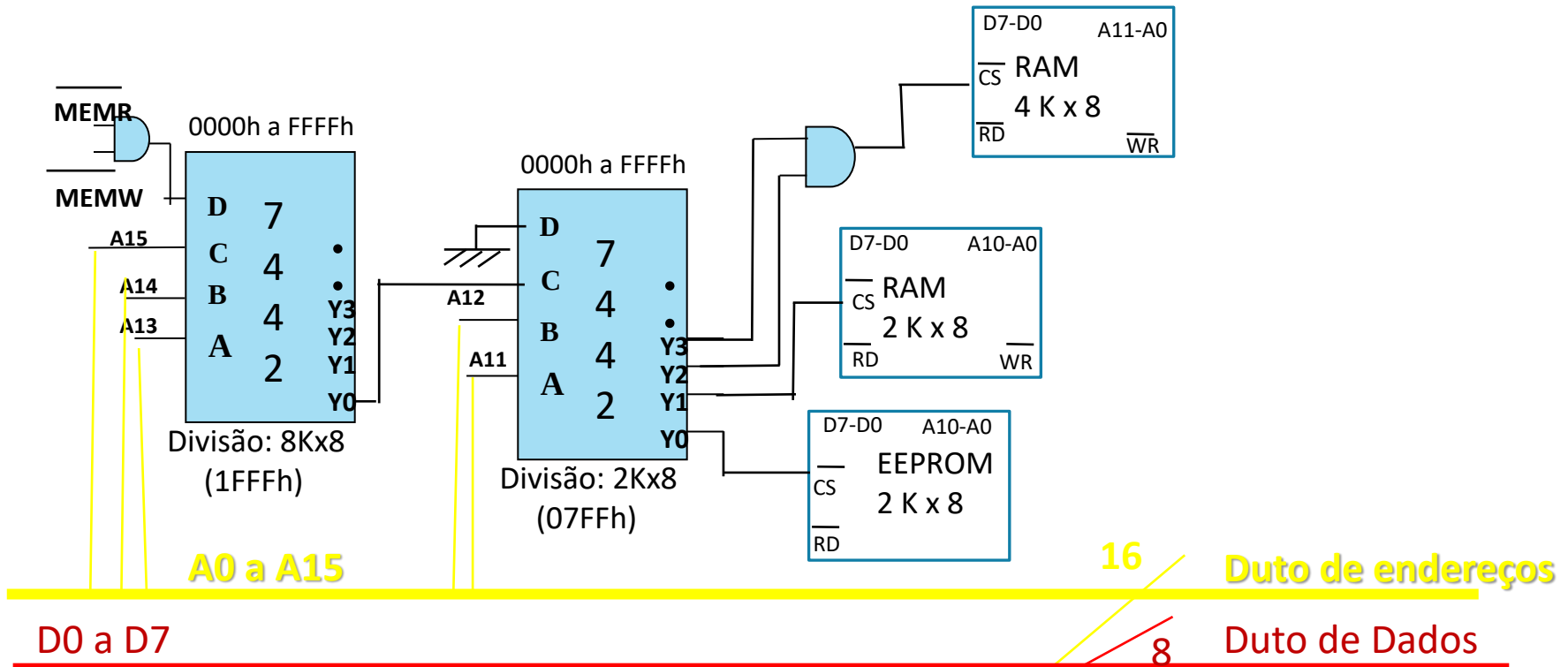
Exemplo 3 (continuação):

Lógica de Seleção com Decodificação Absoluta e memórias em sequência
Utilizando A12 e A11 como seleção:

Lógica de Endereçamento do µP – Endereço de dados																	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	2 k	
	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1			07FF
RAM	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0800	2 k	
	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1			0FFF
RAM	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1000	4 k	
	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1			1FFF

Exemplo 3 (continuação):

Decodificação Absoluta
Circuito Final:



Exemplo 3 (continuação):

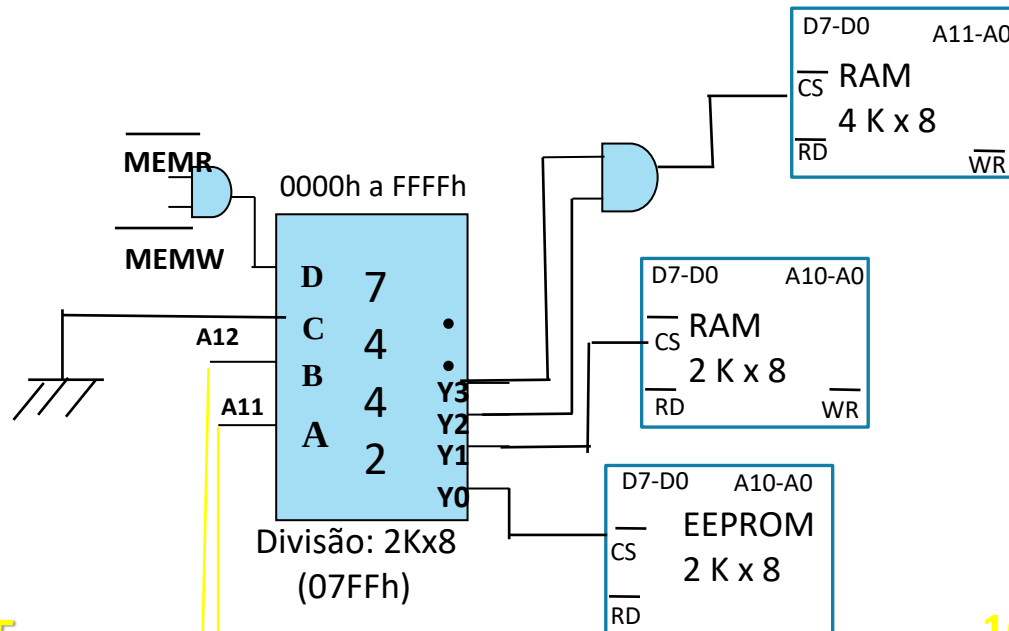
Lógica de Seleção com Decodificação NÃO Absoluta e memórias em sequência

Utilizando A12 e A11 como seleção:

Lógica de Endereçamento do µP – Endereço de dados																	Memória		
Tipo	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Início (H)		Fim (H)
ROM	X	X	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	2 k	
	X	X	X	0	0	1	1	1	1	1	1	1	1	1	1	1			07FF
RAM	X	X	X	0	1	0	0	0	0	0	0	0	0	0	0	0	0800	2 k	
	X	X	X	0	1	1	1	1	1	1	1	1	1	1	1	1			0FFF
RAM	X	X	X	1	0	0	0	0	0	0	0	0	0	0	0	0	1000	4 k	
	X	X	X	1	1	1	1	1	1	1	1	1	1	1	1	1			1FFF

Exemplo 3 (continuação):

Decodificação Não Absoluta
Circuito Final:



A0 a A15

16

Duto de endereços

D0 a D7

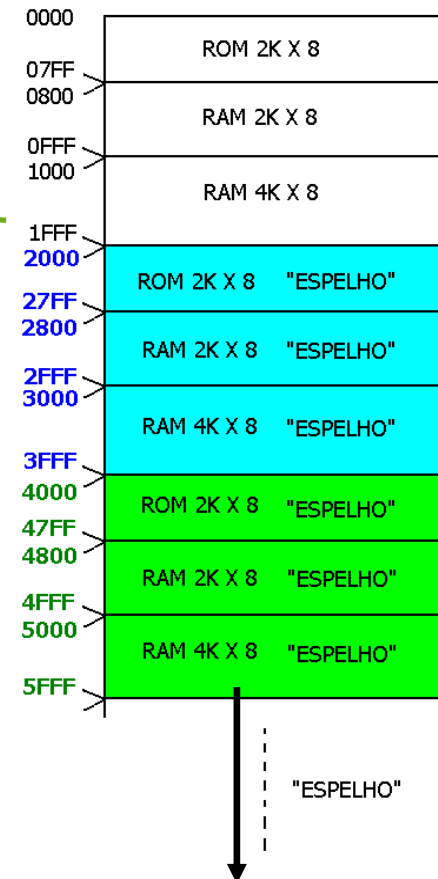
8

Duto de Dados

Exemplo 3 (continuação):

Esse bloco é “espelhado”
mais 8 vezes até o endereço
final (FFFFh)

Decodificação Não Absoluta
Mapeamento da Memória



FIM