

Departamento de Engenharia Elétrica e de Computação
EESC-USP

SEL-415 Introdução à Organização de Computadores

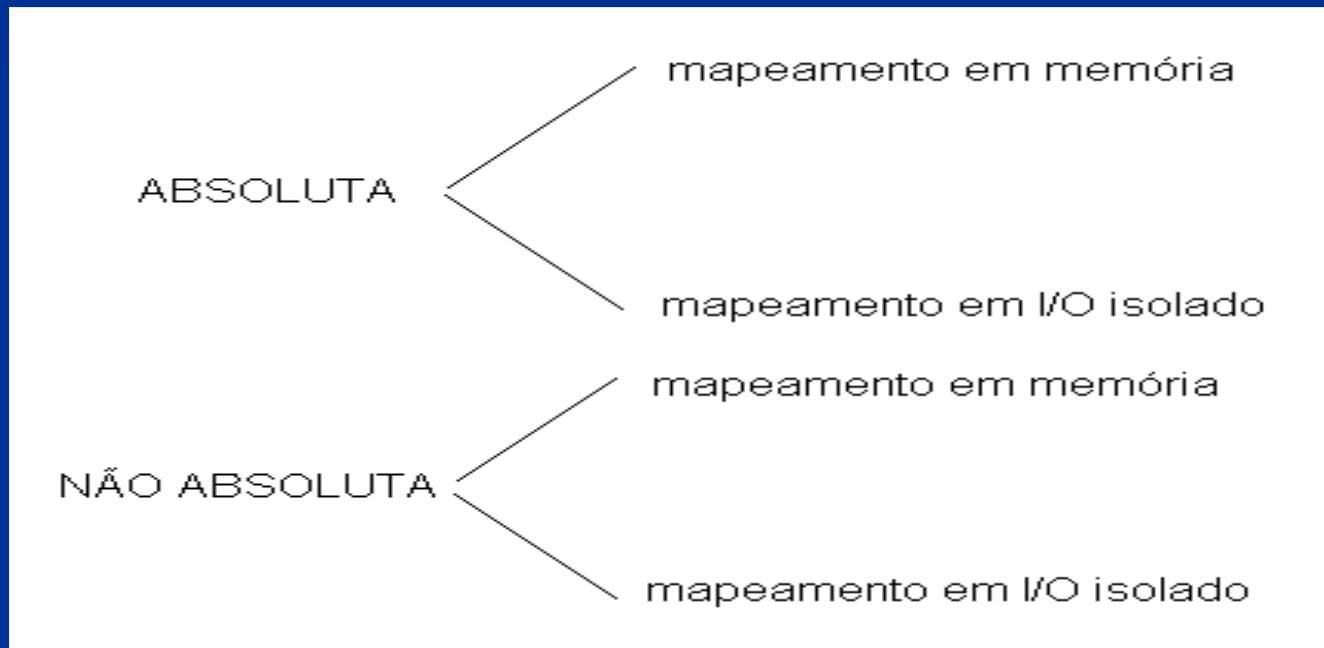
Tipos de Lógica de Seleção

Aula 5- *parte2*

Profa. Luiza Maria Romeiro Codá

LÓGICA DE SELEÇÃO DE MEMÓRIA E DE DISPOSITIVOS DE I/O

3. Tipos de Lógica de Seleção



LÓGICA DE SELEÇÃO DE MEMÓRIA E DE DISPOSITIVOS DE I/O

3. Tipos de Lógica de Seleção

Mapeamento em Memória: 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

Mapeamento em I/O isolado: os dispositivos de I/O são mapeados num espaço de I/O separado do espaço de memória.

➤ os sinais de controle de leitura e gravação são diferentes para os dois espaços

➤ É feita uma lógica de seleção separada para cada espaço

LÓGICA DE SELEÇÃO DE MEMÓRIA E DE DISPOSITIVOS DE I/O

3.1 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.
- 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 chip** tem valor **zero** para o endereço inicial.

LÓGICA DE SELEÇÃO DE MEMÓRIA E DE DISPOSITIVOS DE I/O

3.1 Lógica de Seleção Absoluta

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

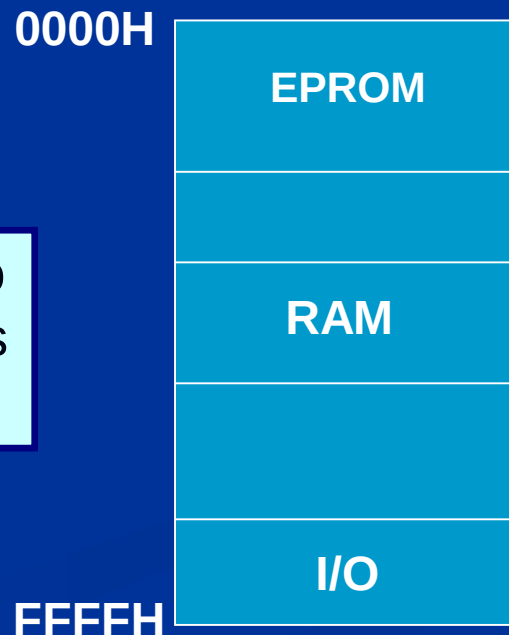
Desvantagem: o hardware é mais complexo que o da lógica não absoluta, principalmente para interfaces que ocupam poucas posições de memória

LÓGICA DE SELEÇÃO DE MEMÓRIA E DE DISPOSITIVOS DE I/O

3.1.a Exemplos de lógica de seleção absoluta, Mapeamento em Memória

Para o circuito de seleção da figura 2 é determinada a faixa de endereço associada a cada saída de seleção CSi (tabela 2 e 3)

No espaço de 64K bytes são mapeadas as memórias e os dispositivos de I/O



LÓGICA DE SELEÇÃO DE MEMÓRIA E DE DISPOSITIVOS DE I/O

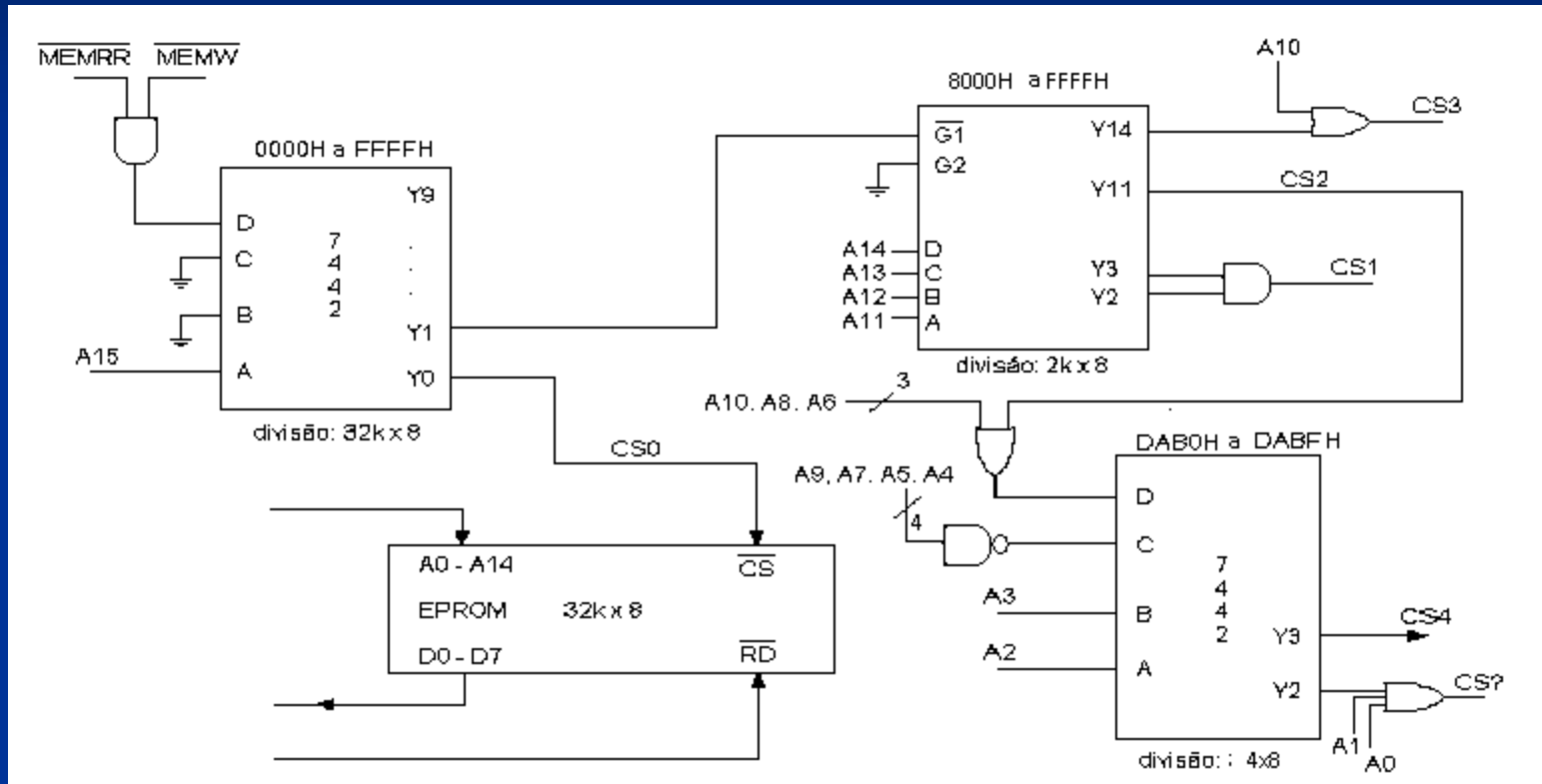


Figura 2 - Exemplo de lógica de seleção absoluta, mapeamento em memória

LÓGICA DE SELEÇÃO DE MEMÓRIA E DE DISPOSITIVOS DE I/O

Tabela 2 - Faixa de endereços para cada saída de seleção

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

LÓGICA DE SELEÇÃO DE MEMÓRIA E DE DISPOSITIVOS DE I/O

- CS0 : 0000 -> 7FFFFH ; 32k x 8
- CS1 : 9000H -> 9FFFFH ; 4k x 8
- CS2 : D800H -> DFFFFH ; 2k x 8
- CS3 : F000H -> F3FFFH ; 1k x 8
- CS4 : DABCH -> DABFH ; 4 x 8

Tabela 3 - Faixa de endereços em hexadecimal, referentes à tabela 2

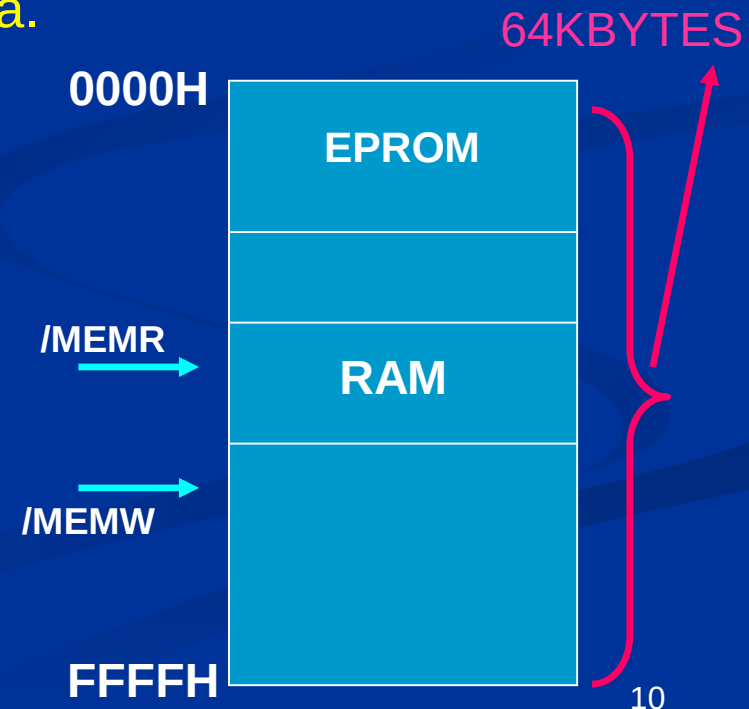
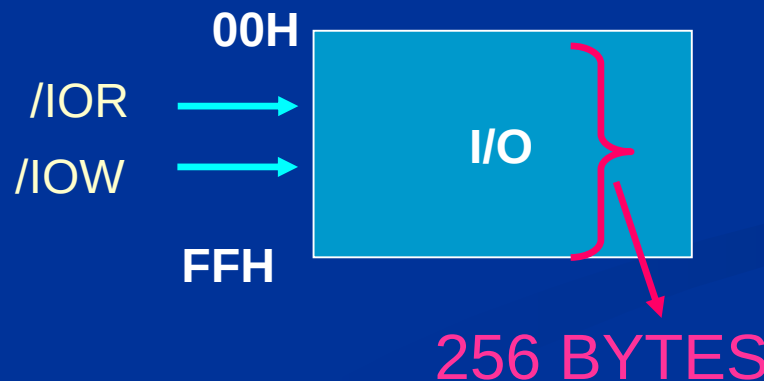
LÓGICA DE SELEÇÃO DE MEMÓRIA E DE DISPOSITIVOS DE I/O

3.1.b Exemplos de lógica de seleção absoluta, Mapeamento em I/O Isolado

Para o circuito de seleção da figura 3 é determinada a faixa de endereço associada a cada saída de seleção CSi (tabela 4) .

Os dispositivos de I/O são mapeados num espaço de I/O de 256 bytes, separado do espaço de memória.

Os sinais de controle são diferentes para os dois espaços



LÓGICA DE SELEÇÃO DE MEMÓRIA E DE DISPOSITIVOS DE I/O

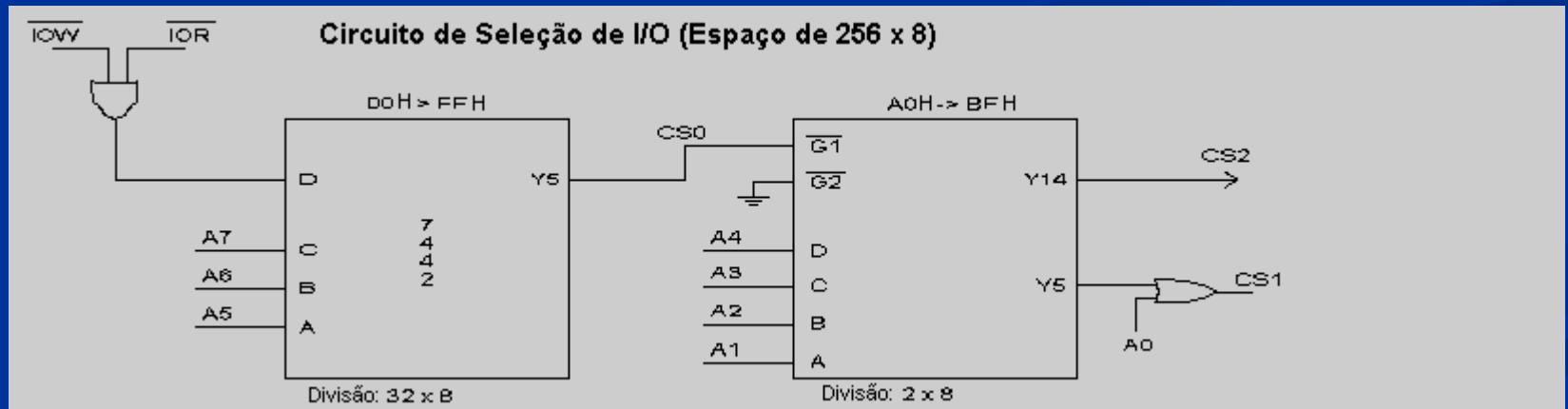


Figura 3 - exemplo de lógica de seleção absoluta, mapeamento em I/O isolado 11

LÓGICA DE SELEÇÃO DE MEMÓRIA E DE DISPOSITIVOS DE I/O

	A7	A6	A5	A4	A3	A2	A1	A0
CS0	1 1	0 0	1 1	0 1	0 1	0 1	0 1	0 1
CS1	1 1	0 0	1 1	0 0	1 1	0 0	1 1	0 0
CS2	1 1	0 0	1 1	1 1	1 1	1 1	0 0	0 1

CS0 : **A0H -> BFH** **->** **espaço: 32 x 8**
CS1 : **AAH** **->** **espaço: 1 x 8**
CS2 : **BCH -> BDH** **->** **espaço: 2 x 8**

Tabela 4 : mapa de endereços para o circuito de seleção de I/O da figura 3

LÓGICA DE SELEÇÃO DE MEMÓRIA E DE DISPOSITIVOS DE I/O

3.2 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**;
 - esse hardware de seleção é inadequado para área de dados seqüenciais;
 - os endereços fantasmas não podem ser usados por outros chips, pois não são espaços livres.

LÓGICA DE SELEÇÃO DE MEMÓRIA E DE DISPOSITIVOS DE I/O

3.2.a Lógica de Seleção NÃO Absoluta, mapeamento em memória

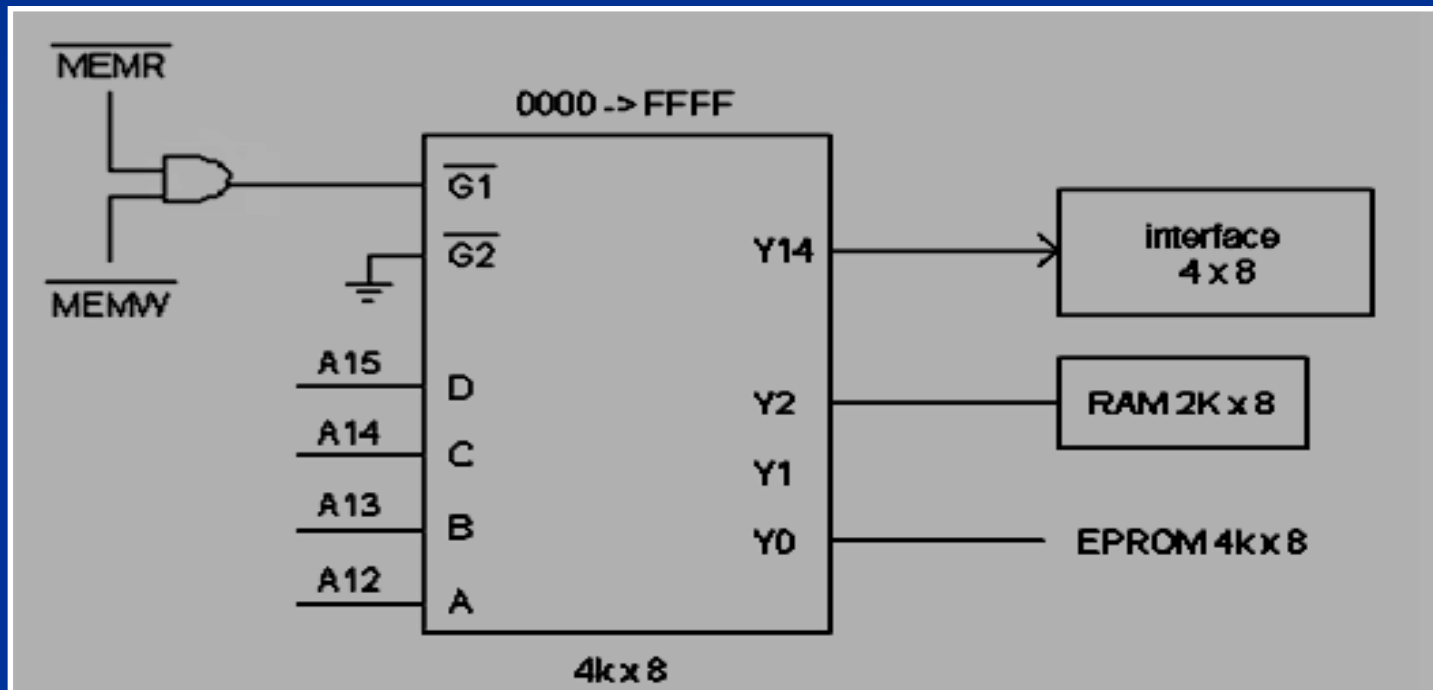


Figura 4 - Exemplo de lógica de seleção não absoluta, mapeamento em memória

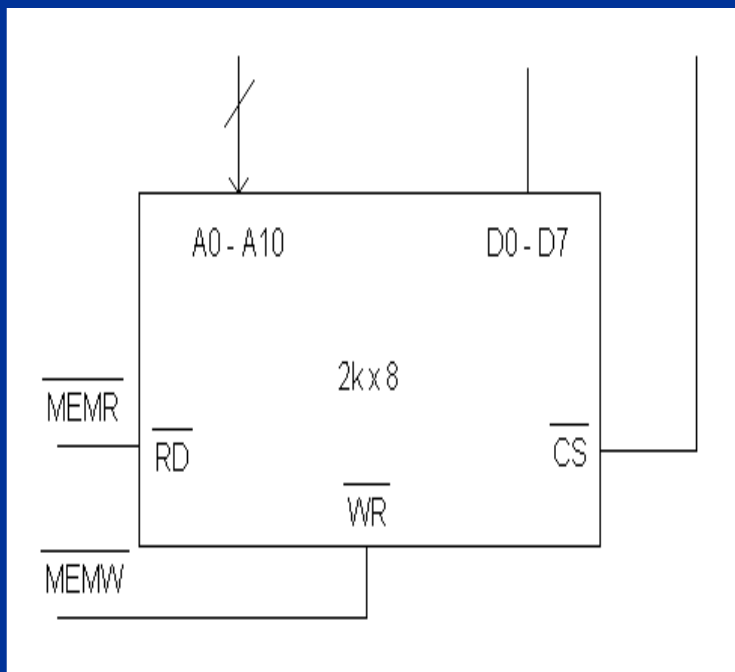
LÓGICA DE SELEÇÃO DE MEMÓRIA E DE DISPOSITIVOS DE I/O

Faixas de endereço para a figura 4

a) Y0 : 0000H -> 0FFF H (4k x 8)

b) Y2 : 2000H -> 2FFFH (4k x 8)

A memória conectada a Y2 é de 2k x 8 , organização menor do que o espaço gerado pela lógica de seleção.



O bit de endereço **A11** não está presente na lógica de seleção para a memória de **2k x 8**, o que leva essa memória a ter dois espaços de endereço associados a ela

LÓGICA DE SELEÇÃO DE MEMÓRIA E DE DISPOSITIVOS DE I/O

Faixa de endereço para a memória RAM de 2Kx8

	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**

- Um dos espaços de 2k x 8 é denominado de **espaço fantasma (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
- (linhas de endereçamento do chip são iguais a zero para esses dois endereços).

LÓGICA DE SELEÇÃO DE MEMÓRIA E DE DISPOSITIVOS DE I/O

c. Faixa de endereço para interface de 4 x 8

Y14: E000H -> EFFFH (4Kx8)

1ª. Faixa de endereço da interface: E000 -> E003H

2ª. Faixa de endereço da interface: E004 -> E007H

.

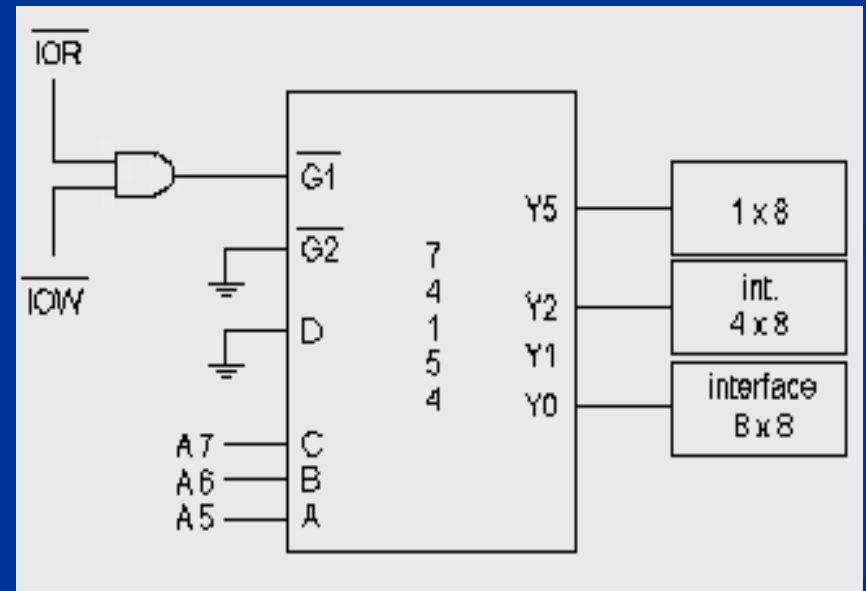
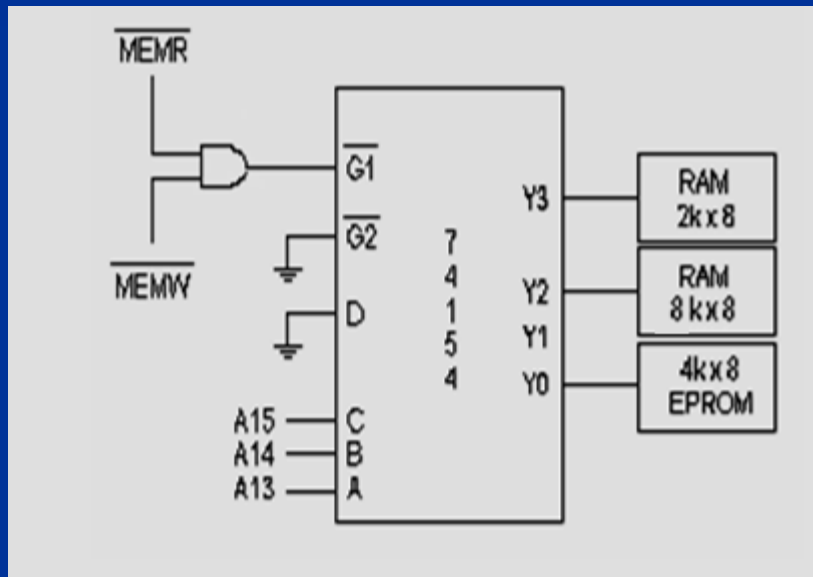
.

.

Quantas faixas de endereço estarão associadas a essa interface?

LÓGICA DE SELEÇÃO DE MEMÓRIA E DE DISPOSITIVOS DE I/O

3.2.b. - Exemplo de lógica de seleção não absoluta, mapeamento I/O isolado



Determine a faixa de endereço para cada saída de seleção, e os endereços fantasmas

Exemplos

Exercício 1 da Lista nº9

Faça o projeto da lógica de seleção para dividir o espaço de endereço do microprocessador, de 16 linhas de endereços e 8 linhas de dados, em blocos de 4Kbytes, especificando endereço inicial e final de cada bloco.

Divida o bloco que inicia no endereço 4000H, em blocos de 1Kbytes e o bloco que inicia no endereço A000H em blocos de 512 bytes.

Usando lógica de seleção absoluta, ligue uma memória de 2kbytes, a partir do endereço 1000H, outra a partir do endereço 4000H e outra a partir do endereço A000H, determinando o endereço final de cada memória.

Repita o mesmo para lógica de seleção não absoluta, determinando os endereços fantasmas. Justifique quando a lógica de seleção absoluta não se aplicar.

Ex. 3: Endereçamento de um bloco de memórias utilizando

Decodificação Absoluta

3 Memórias na seqüência

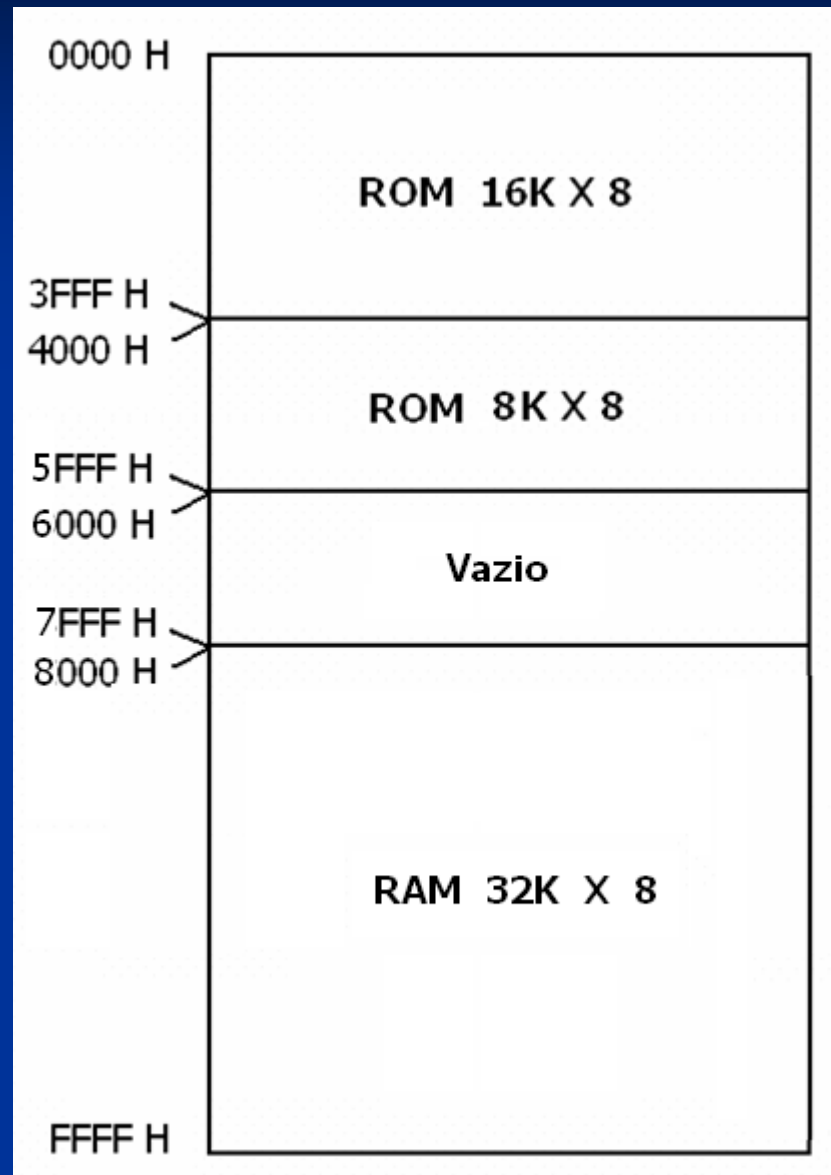
Com alinhamento dos CIs

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

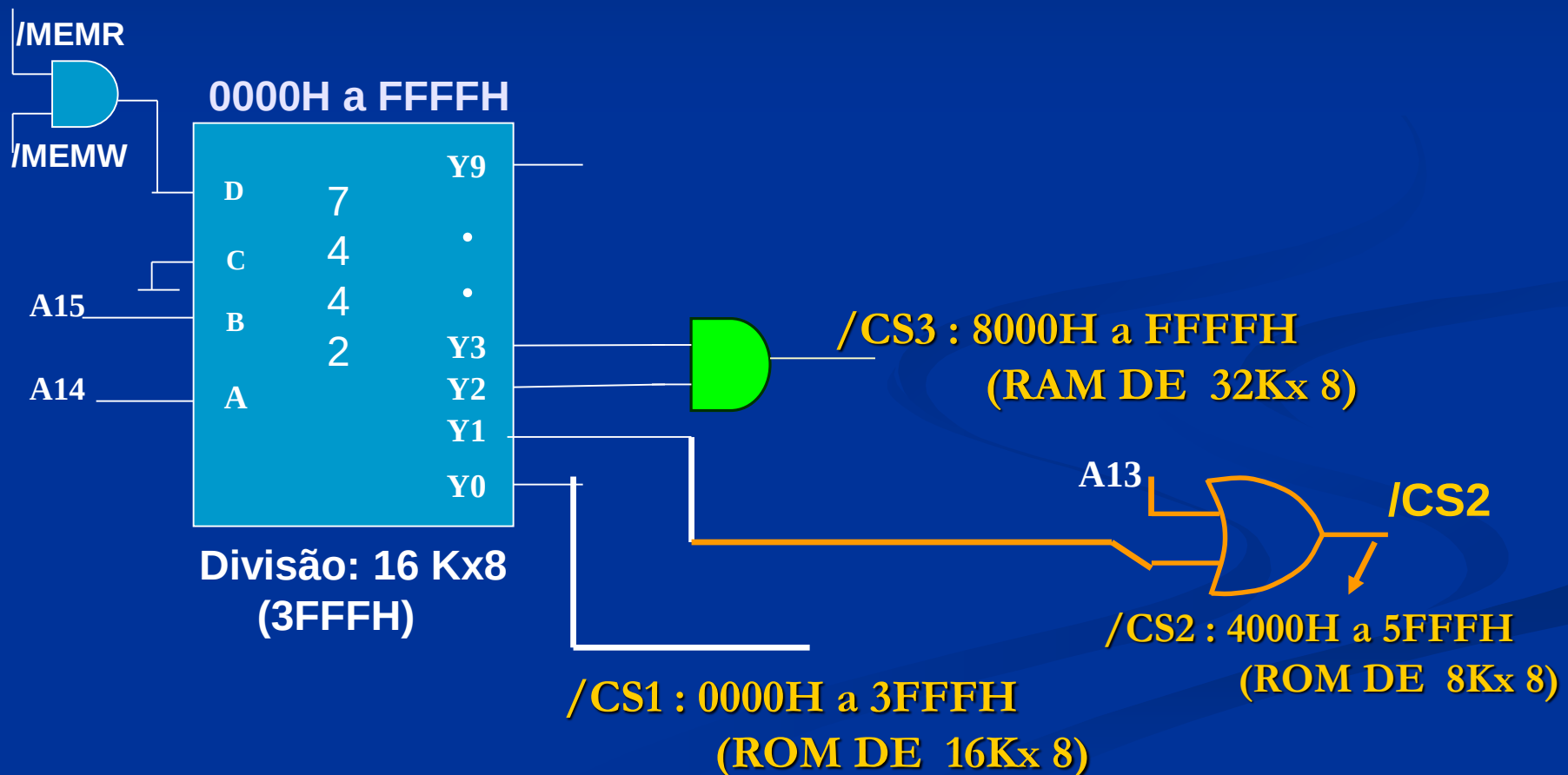
Lógica de Seleção – Decodif. Absoluta com alinhamento das memórias

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	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000	32 k	
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			FFFF
Vazio	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	6000	8 k	
	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			7FFF

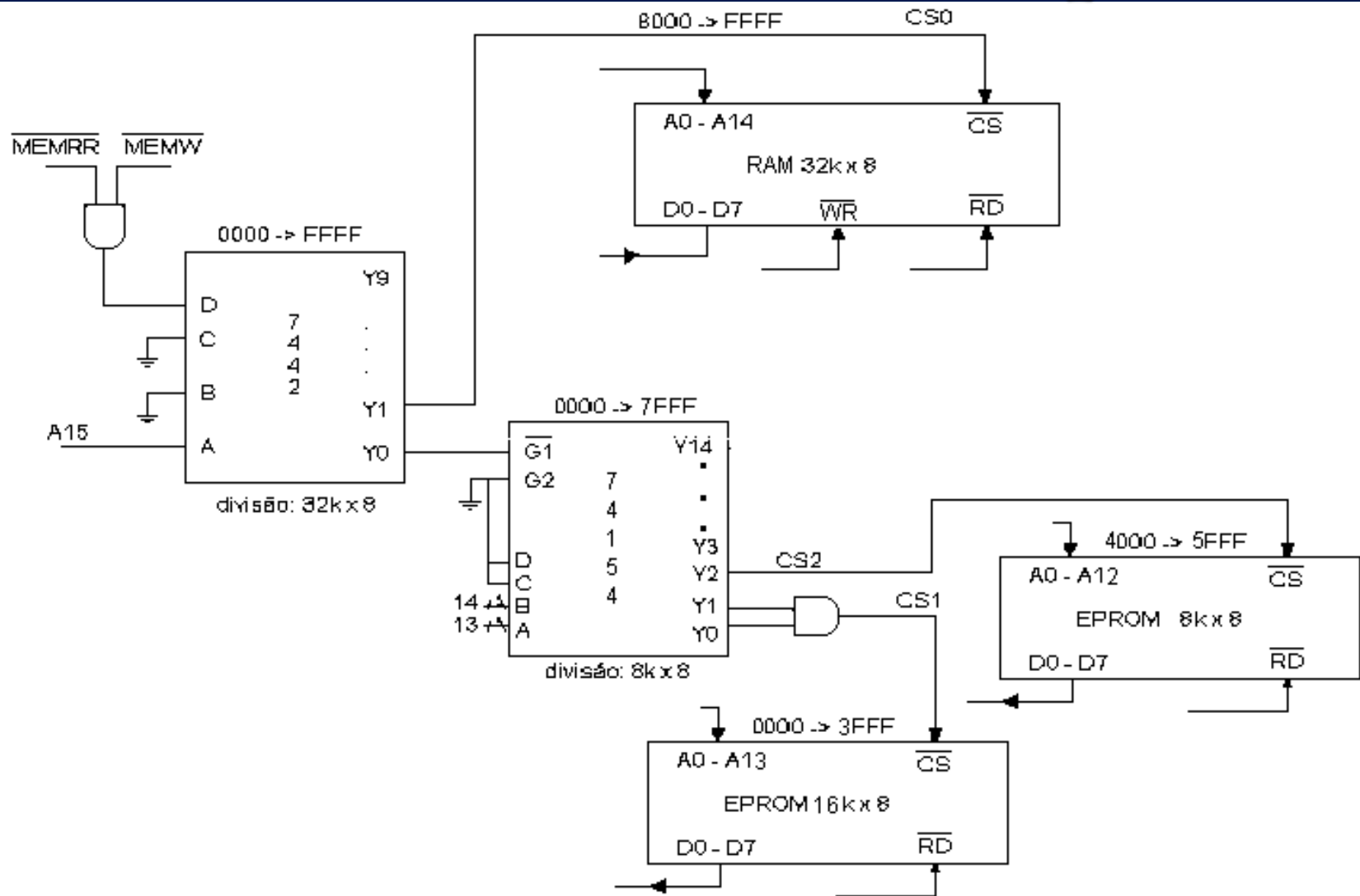
Mapeamento da Memória



Circuito Final - Exemplo 1



Circuito Final - Exemplo 2



Ex. 4: Endereçamento de um bloco de memórias utilizando

Decodificação Absoluta:

5 Memórias na seqüência

Sem o alinhamento dos CIs

- ROM 16 k x 8
- ROM 8 k x 8
- RAM 32 k x 8
- 2 x RAM 4 k x 8

Lógica de Seleção – Decodif. Absoluta

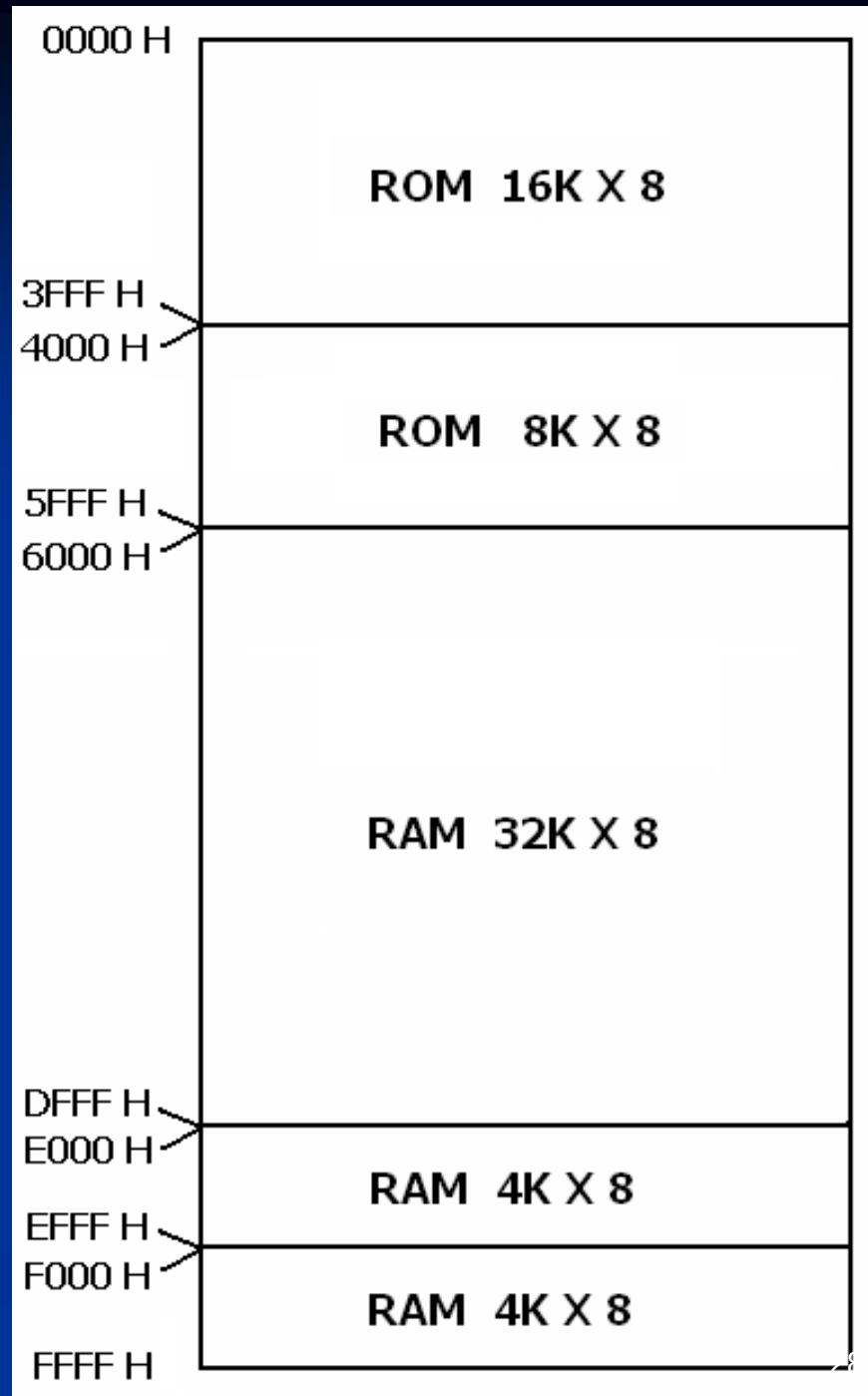
Lógica de Endereçamento do μ P – Endereço de dados																	Memória		
Tipo	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	Início (H)		Fim (H)
ROM	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000	16k	
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1			3FFF
ROM	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000	8 k	
	0	1	0	1	1	1	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			FFFF
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

O 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.

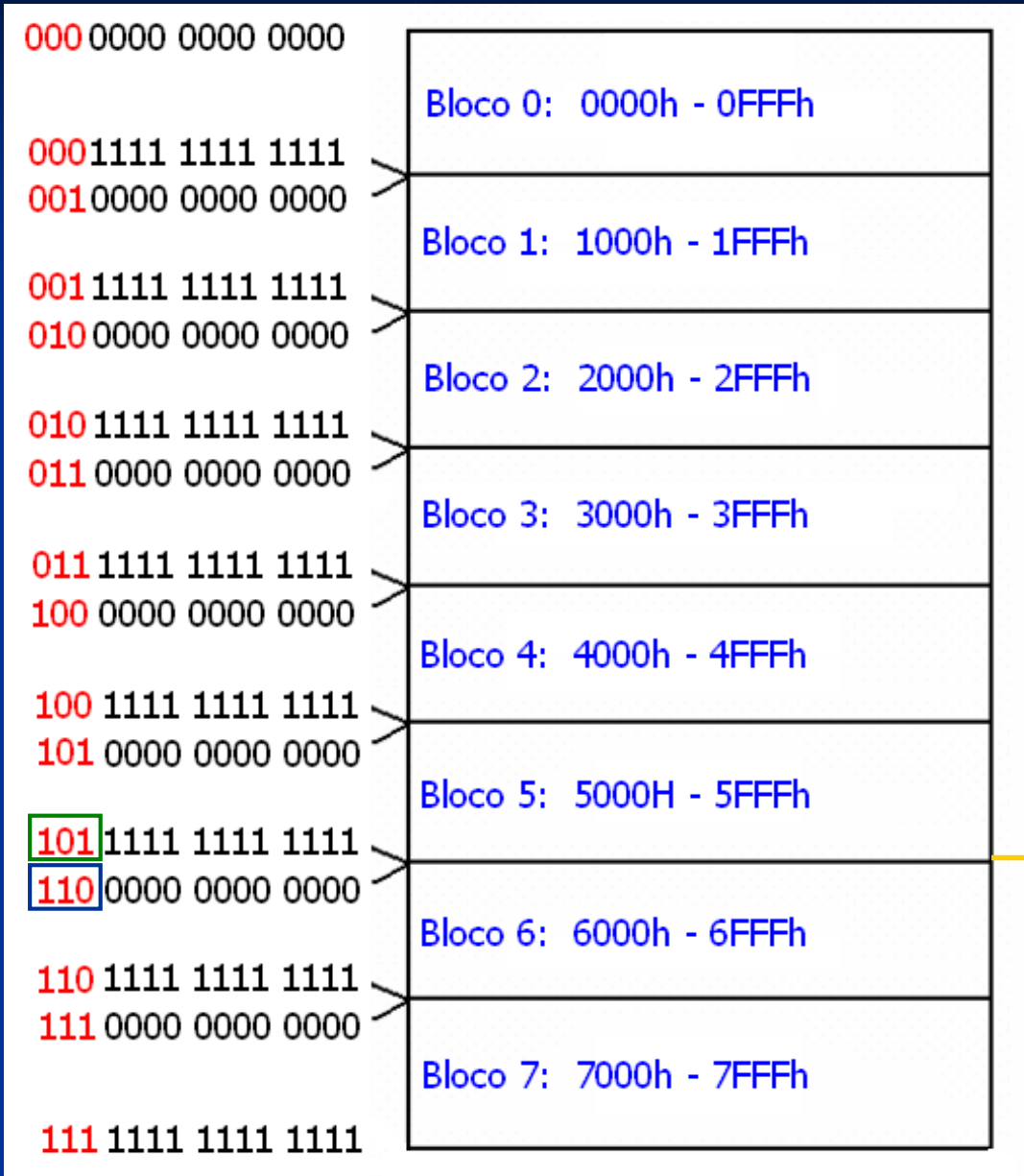
Mapeamento da Memória

Decodificação Absoluta

Obs: Memória de 32Kx8 não alinhada

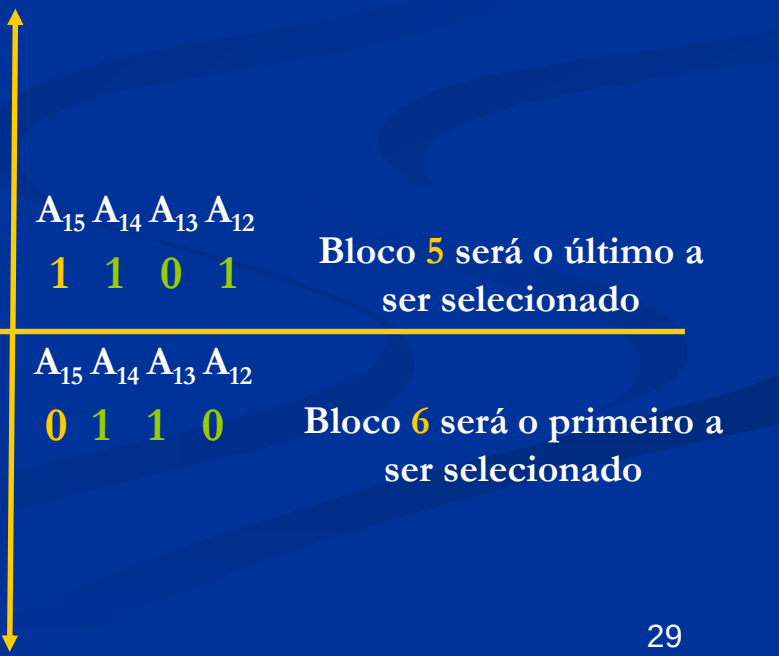


Como é endereçada a Memória RAM 32K x 8 não alinhada?



A sequência de seleção da memória RAM 32K x 8 nesse caso será:

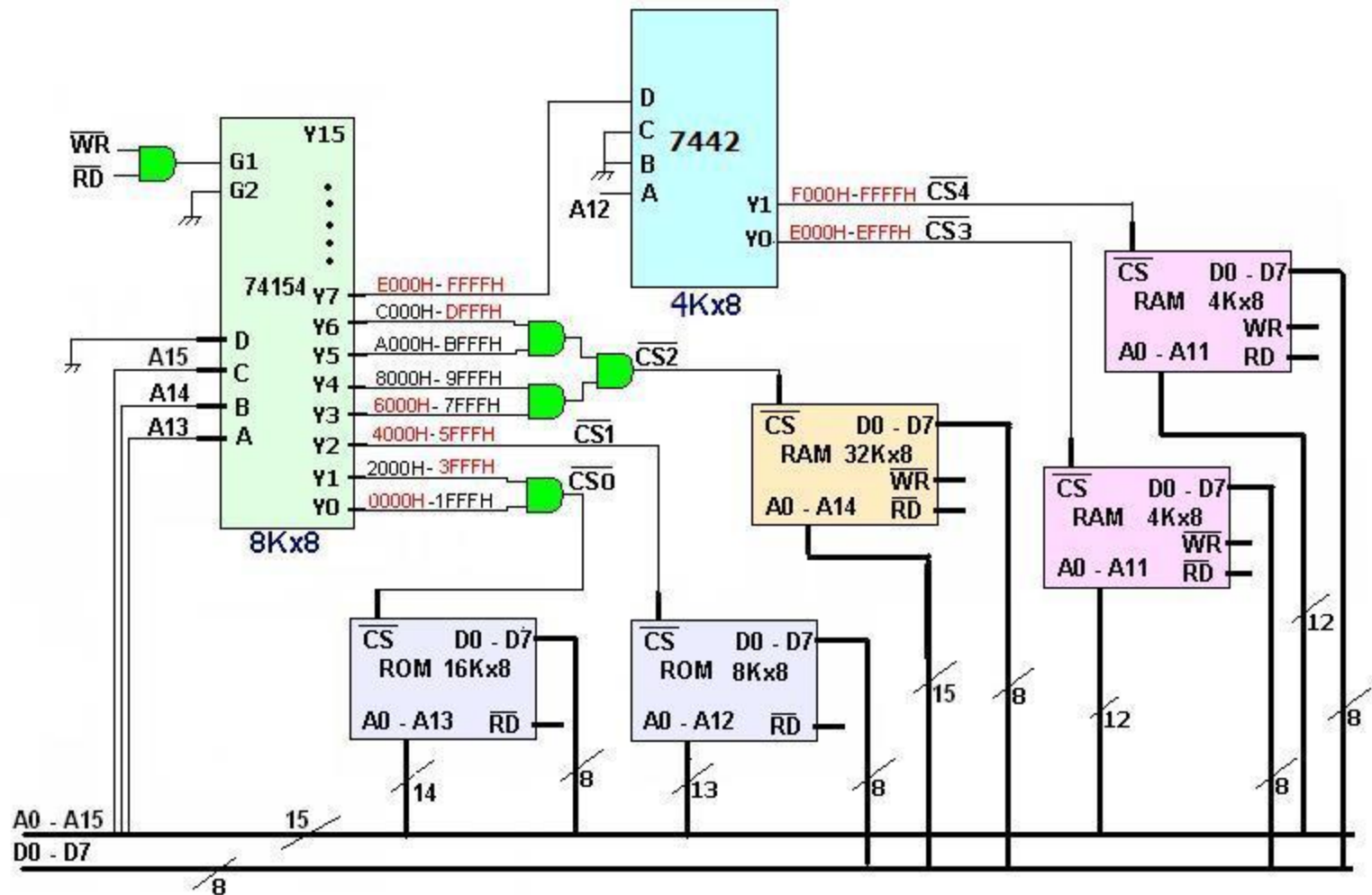
Blocos: 6 → 7 → 0 → 1 → 2 → 3 → 4 → 5



Decodificação Absoluta

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			FFFF
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

1º Exemplo de Implementação:



Como é endereçada a Memória RAM 32K x 8 não alinhada para o exemplo 1?

0110 0000 0000 0000

0111 1111 1111 1111
1000 0000 0000 0000

1001 1111 1111 1111
1010 0000 0000 0000

1011 1111 1111 1111
1100 0000 0000 0000

1101 1111 1111 1111

Bloco 3: 6000 -7FFFF

Bloco 0: 8000 -9FFFF

Bloco 1: A000 -BFFFF

Bloco 2: C000 -DFFFF

A sequência de seleção da memória
RAM 32K x 8 nesse caso será:

Blocos: 1 → 2 → 3 → 0

Memória de 32Kx8

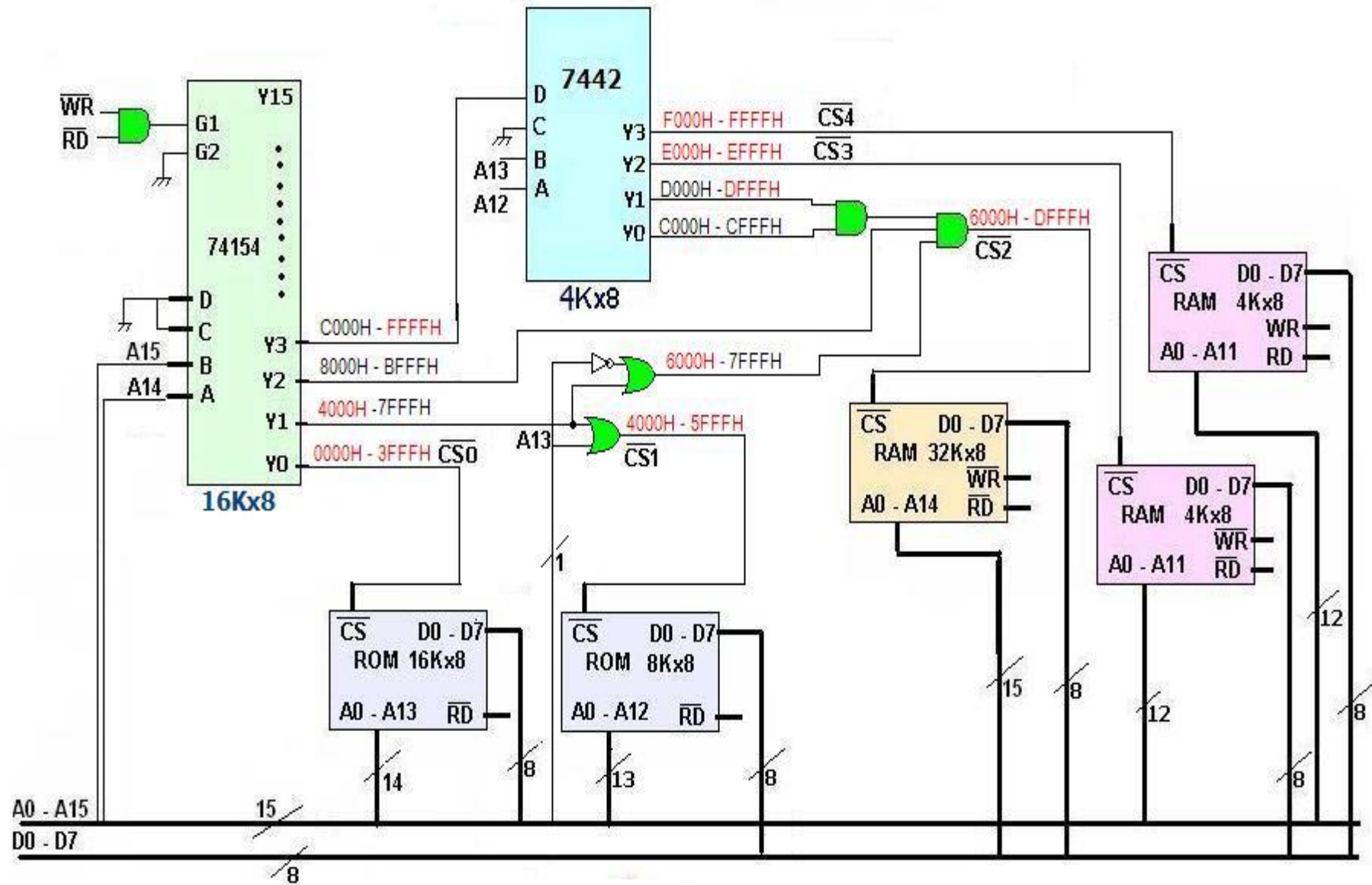
Bloco 0:
(A14,A13) = (0,0)

Bloco 1:
(A14,A13) = (0,1)

Bloco 2:
(A14,A13) = (1,0)

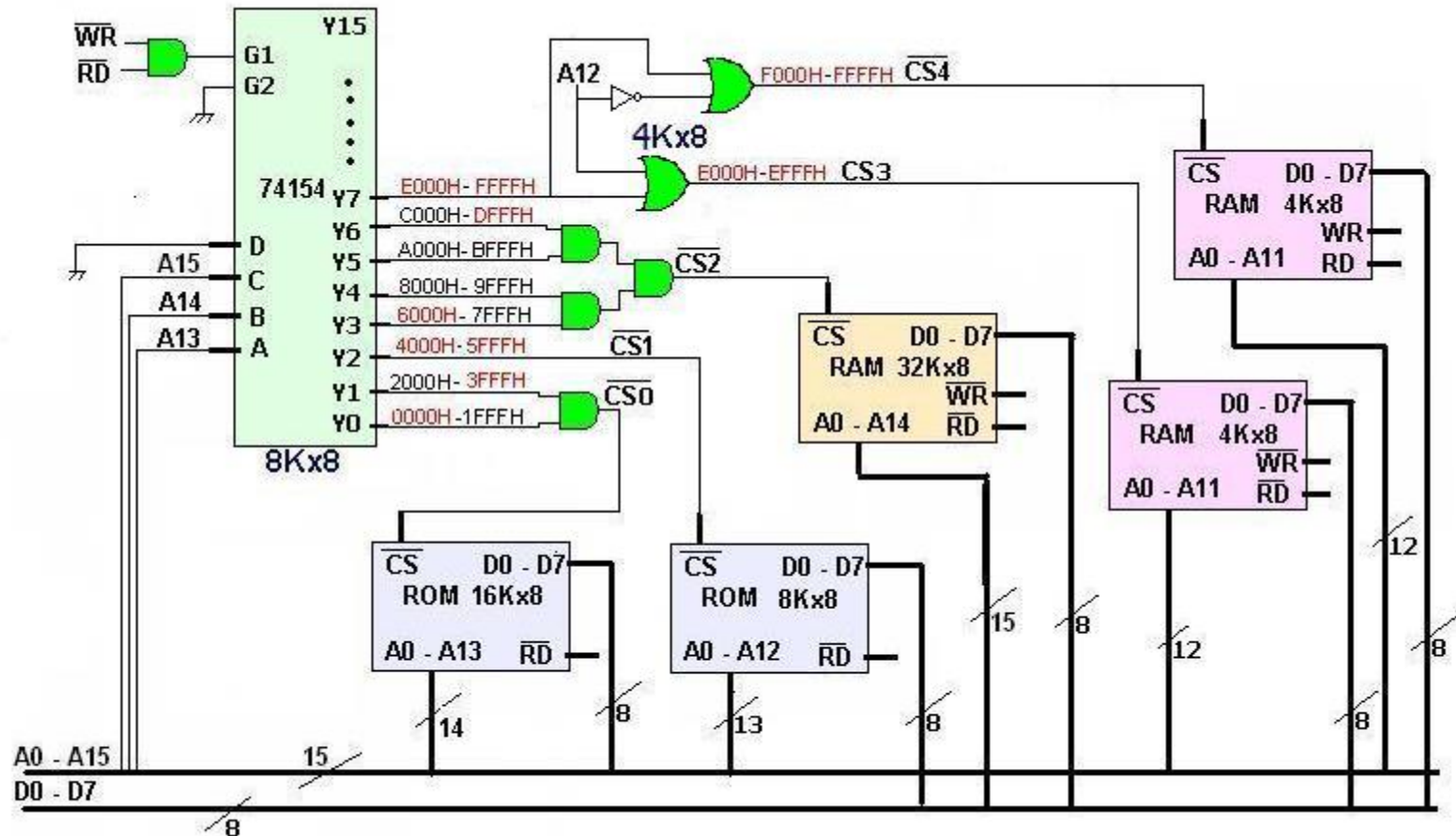
Bloco 3:
(A14,A13) = (1,1)

2º Exemplo de Implementação:



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



Ex. 5: Endereçamento de um bloco de memórias utilizando

Decodificação Não-Absoluta:

3 Memórias na seqüência

Com alinhamento dos CIs

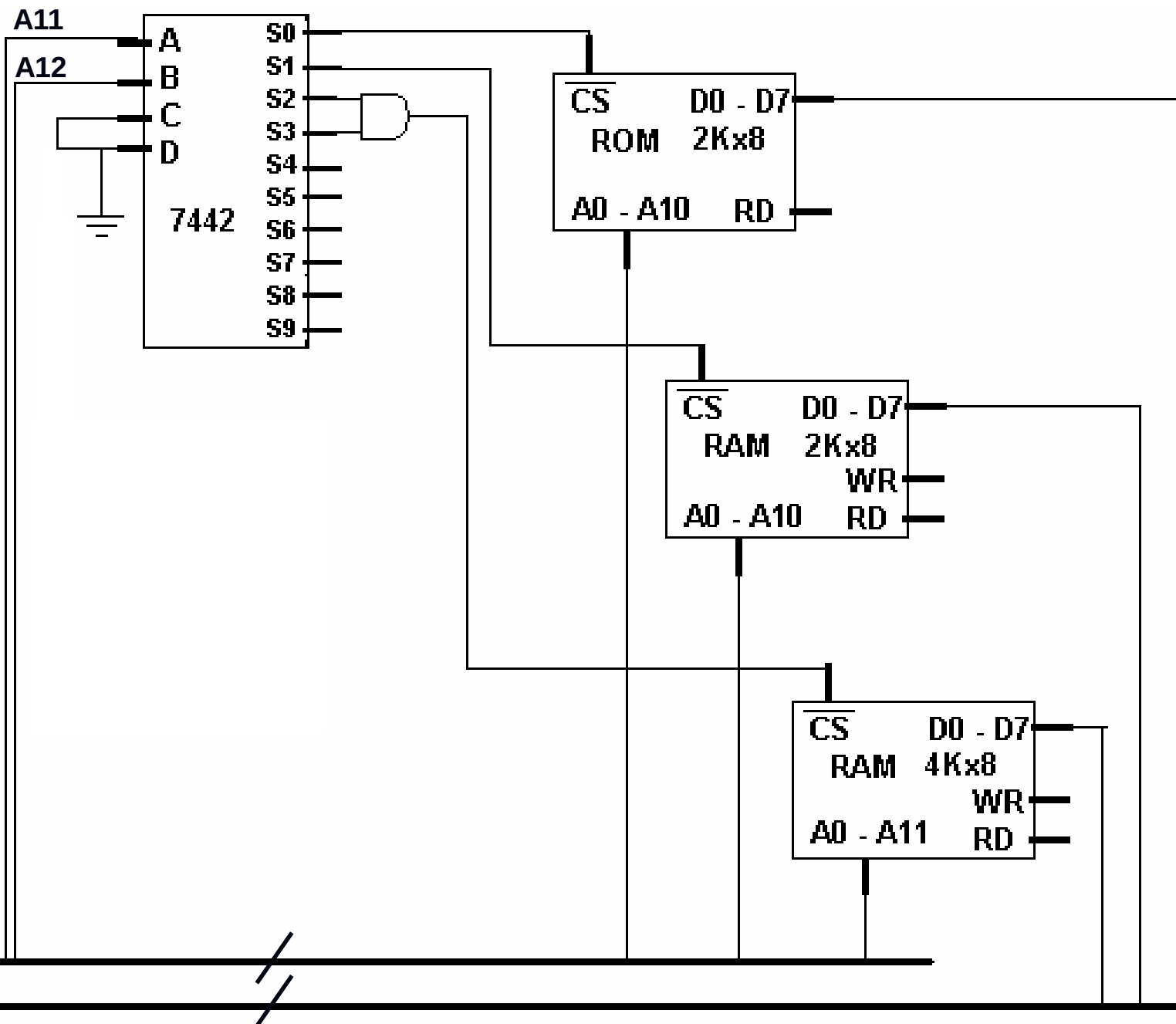
- ROM 2 k x 8
- RAM 2 k x 8
- RAM 4 k x 8

Lógica de Seleção – Decodif. Não-Absoluta

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	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

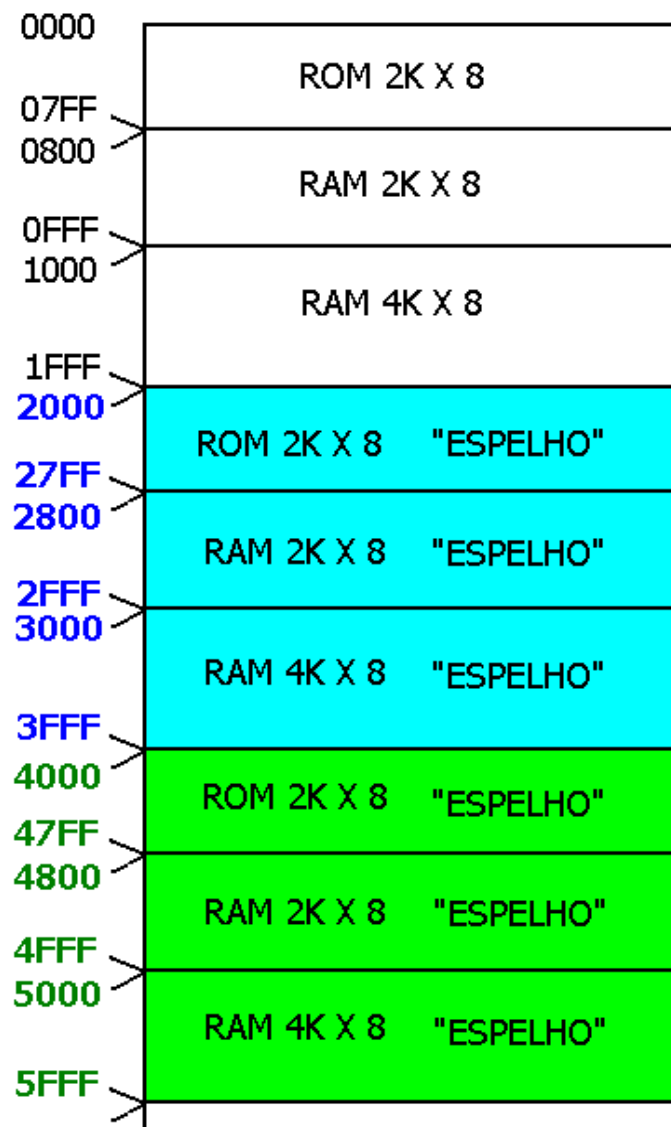
Decodificação Não-Absoluta

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



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

Mapeamento da Memória



FIM