

SEL-0415 Introdução à Organização de Computadores

Lógica de Seleção de Memórias

Aula 8

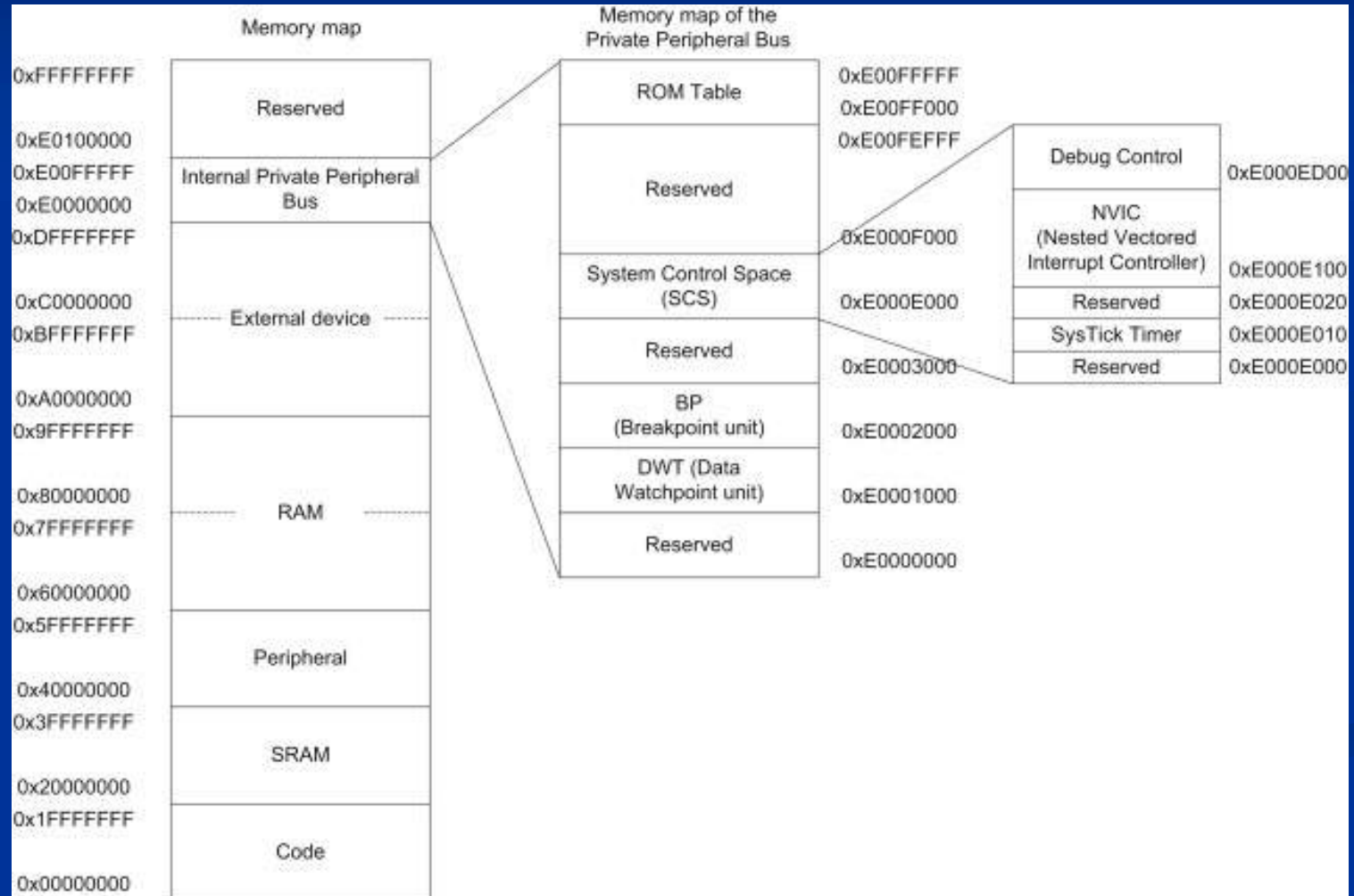
Prof. Dr. Marcelo Andrade da Costa Vieira

LÓGICA DE SELEÇÃO E MAPEAMENTO

- Um microprocessador que tem duto de endereços de N bits e duto de dados de M bits, consegue endereçar 2^N posições de M bits.
- As 2^N posições que o microprocessador consegue endereçar podem ser representadas graficamente por um retângulo dividido em 2^N posições, que é denominado espaço de endereços do microprocessador.
- Dentro do espaço de endereços de 2^N palavras que o microprocessador consegue endereçar, são mapeadas as memórias, os dispositivos de I/O e os periféricos.

LÓGICA DE SELEÇÃO E MAPEAMENTO

Exemplo: Microcontrolador ARM CORTEX-M3



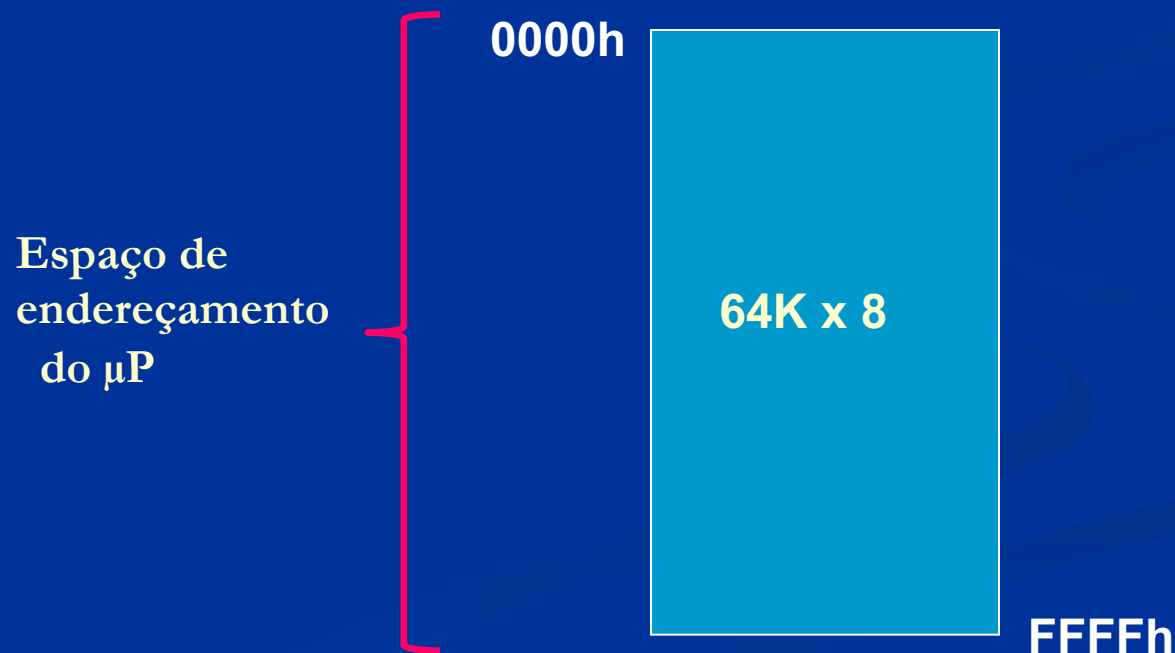
LÓGICA DE SELEÇÃO E MAPEAMENTO

- A “Lógica de Seleção”, construída pelo projetista, define as faixas de endereços do μP que irão selecionar cada uma das memórias e dispositivos.
- O acesso aos endereços dos dispositivos depende de:
 - como eles estão interligados ao barramento de endereços;
 - como foi projetada a lógica de seleção.

LÓGICA DE SELEÇÃO

Exemplo: Um microprocessador que pode endereçar 64Kbytes de memória (16 linhas de endereços e 8 bits de dados). Os endereços de 16 bits, gerados pelo microprocessador pode ser visto como sendo constituído por duas partes:

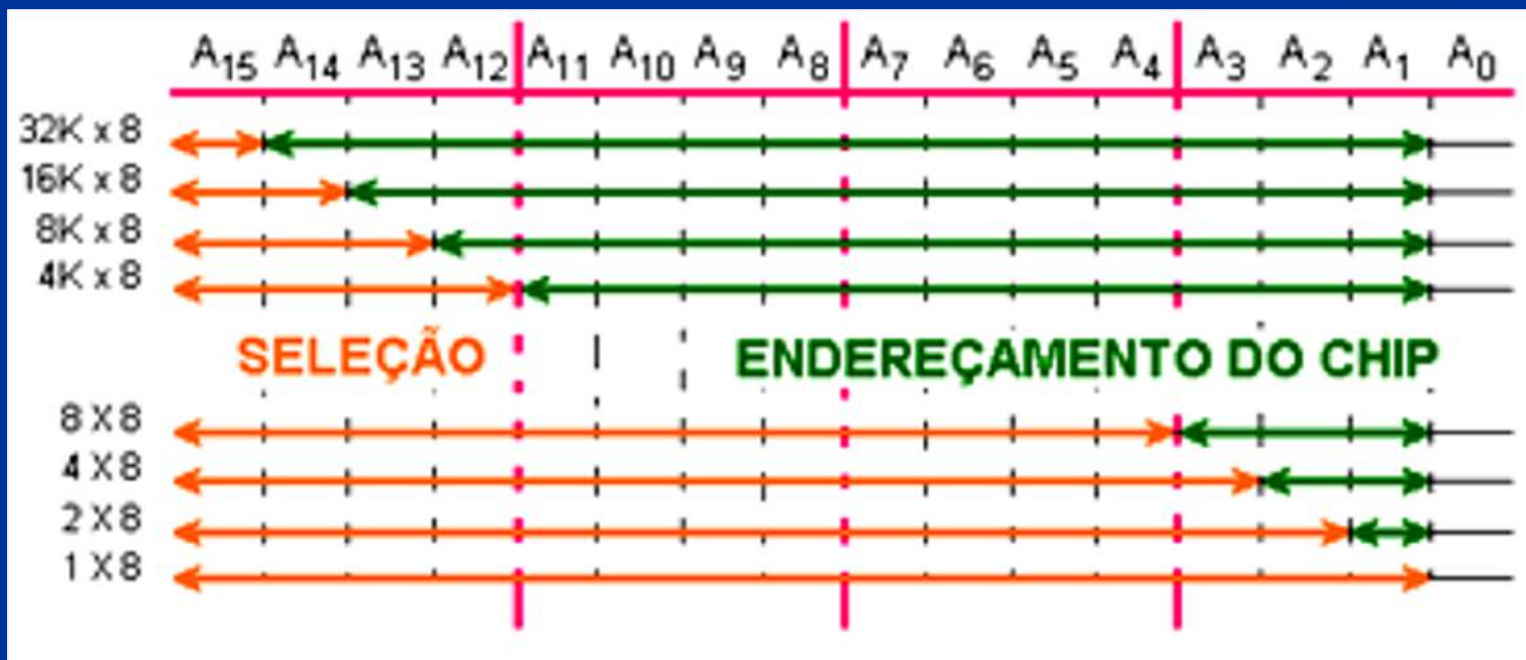
- Bits de endereçamento (tamanho da memória)
- Bits de seleção (número de blocos que podem ser selecionados)



LÓGICA DE SELEÇÃO

Exemplo: Um microprocessador que pode endereçar 64Kbytes de memória (16 linhas de endereços e 8 bits de dados). Os endereços de 16 bits, gerados pelo microprocessador pode ser visto como sendo constituído por duas partes:

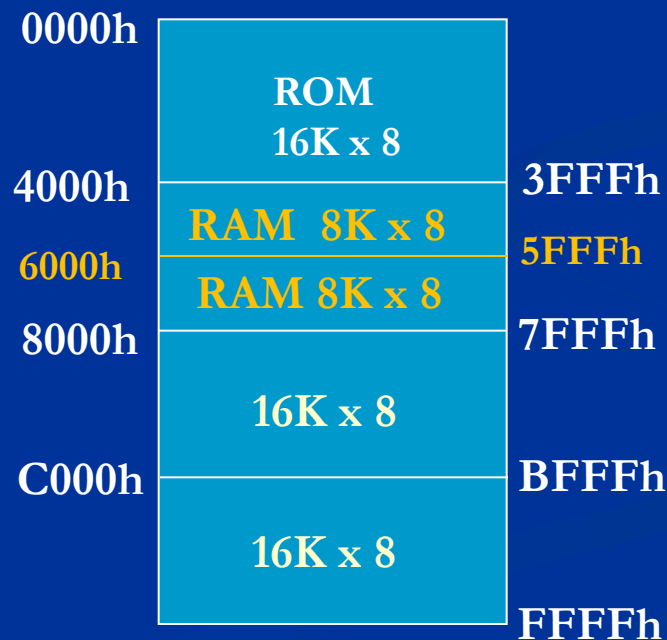
- Bits de endereçamento (tamanho da memória)
- Bits de seleção (número de blocos que podem ser selecionados)



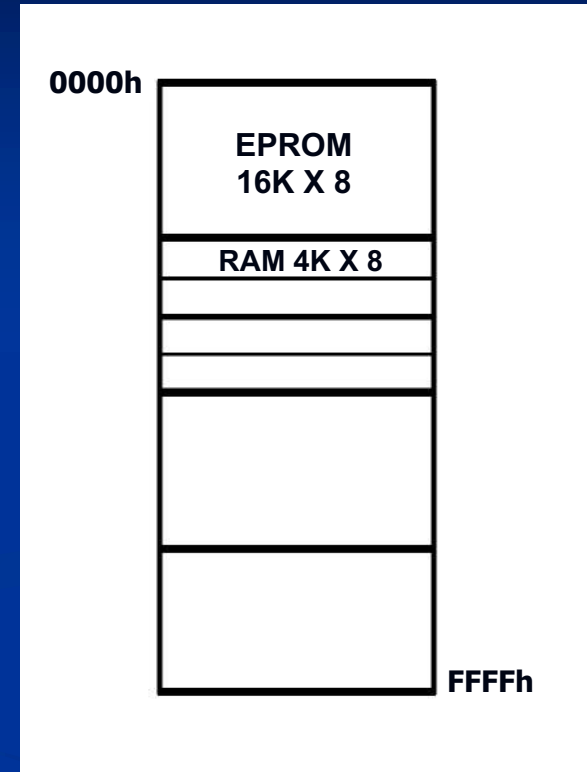
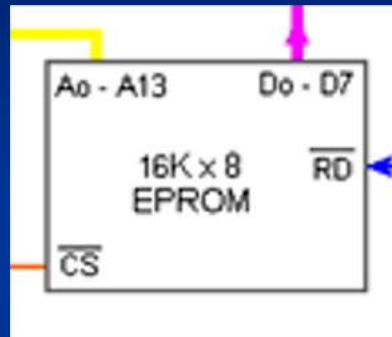
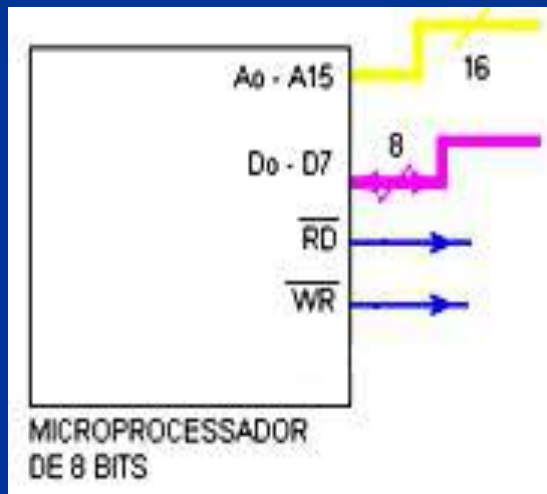
LÓGICA DE SELEÇÃO

■ Exemplo para 64KB:

- ROM 16K x 8 → (A0 – A13) → (A14 – A15: 4 blocos de 16K)
- RAM 8K x 8 → (A0 – A12) → (A13 – A15: 8 blocos de 8K)



Como ligar o μ P às Memórias?



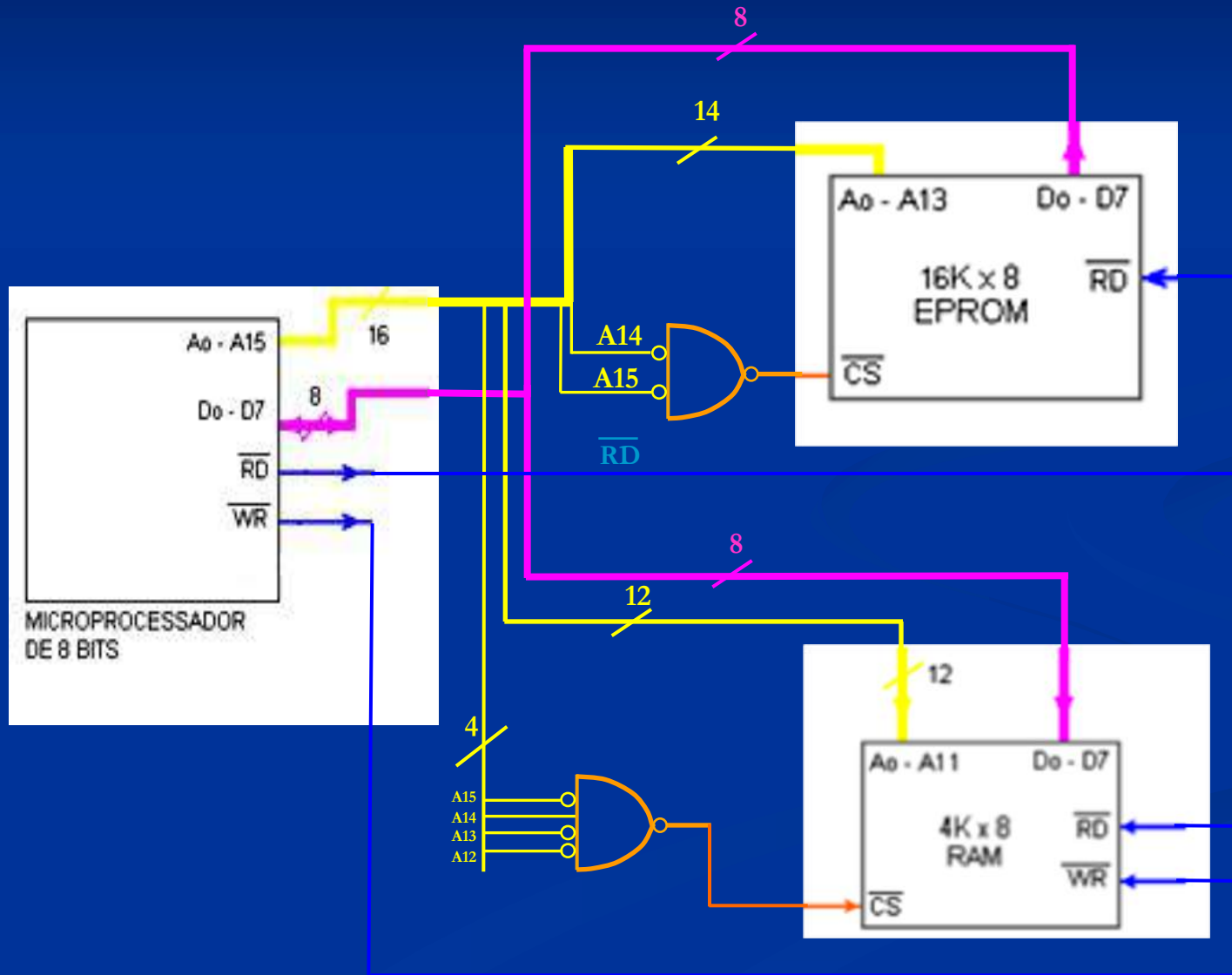
Mapeamento

Lógica de Seleção

Lógica de Seleção do μ P – 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			
RAM																	4000	4k	
Vazio																	5000	44k	

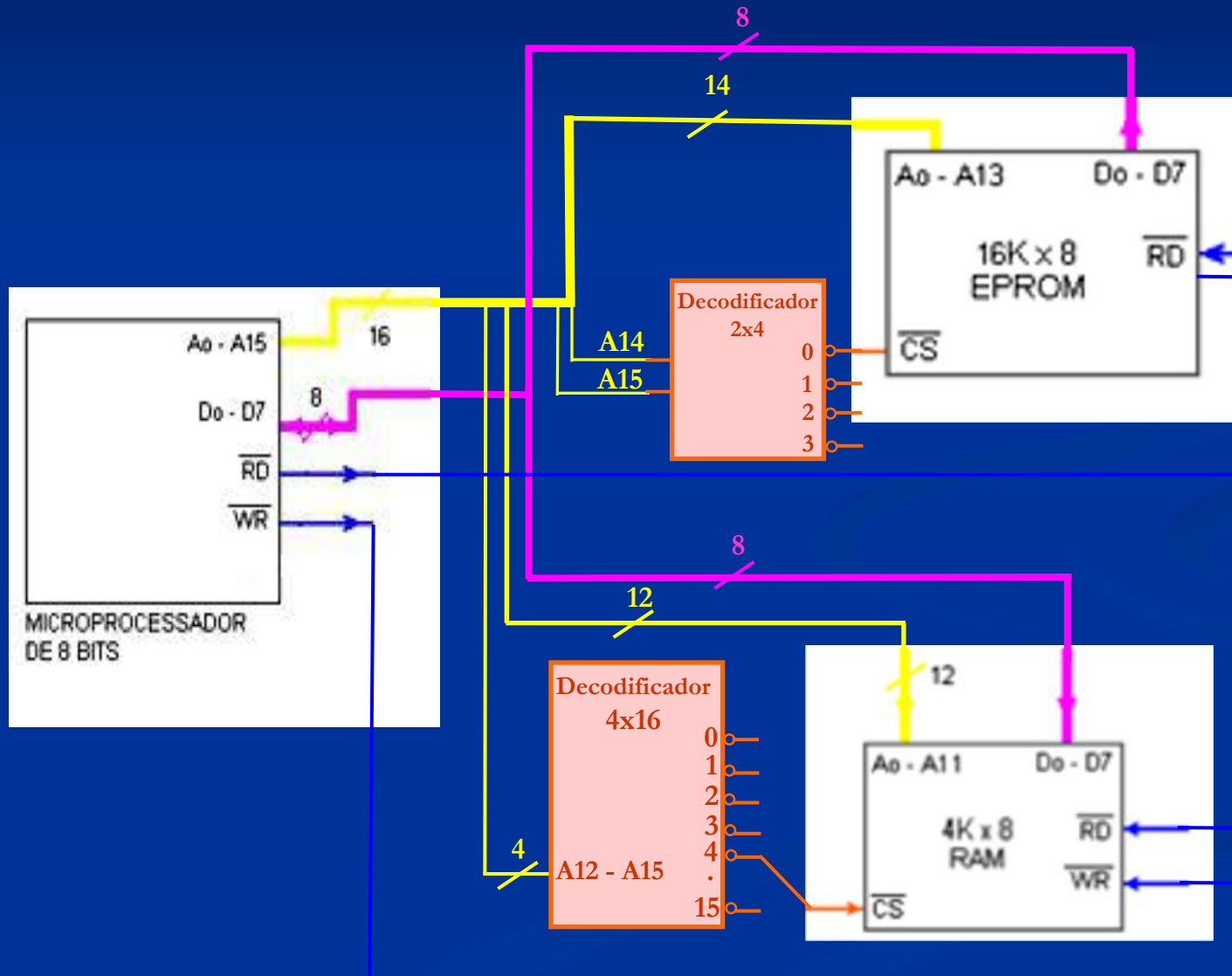
Como ligar o μP às Memórias?

Circuito utilizado para microprocessadores com 16 pinos de endereços e 8 pinos para dutos de dados

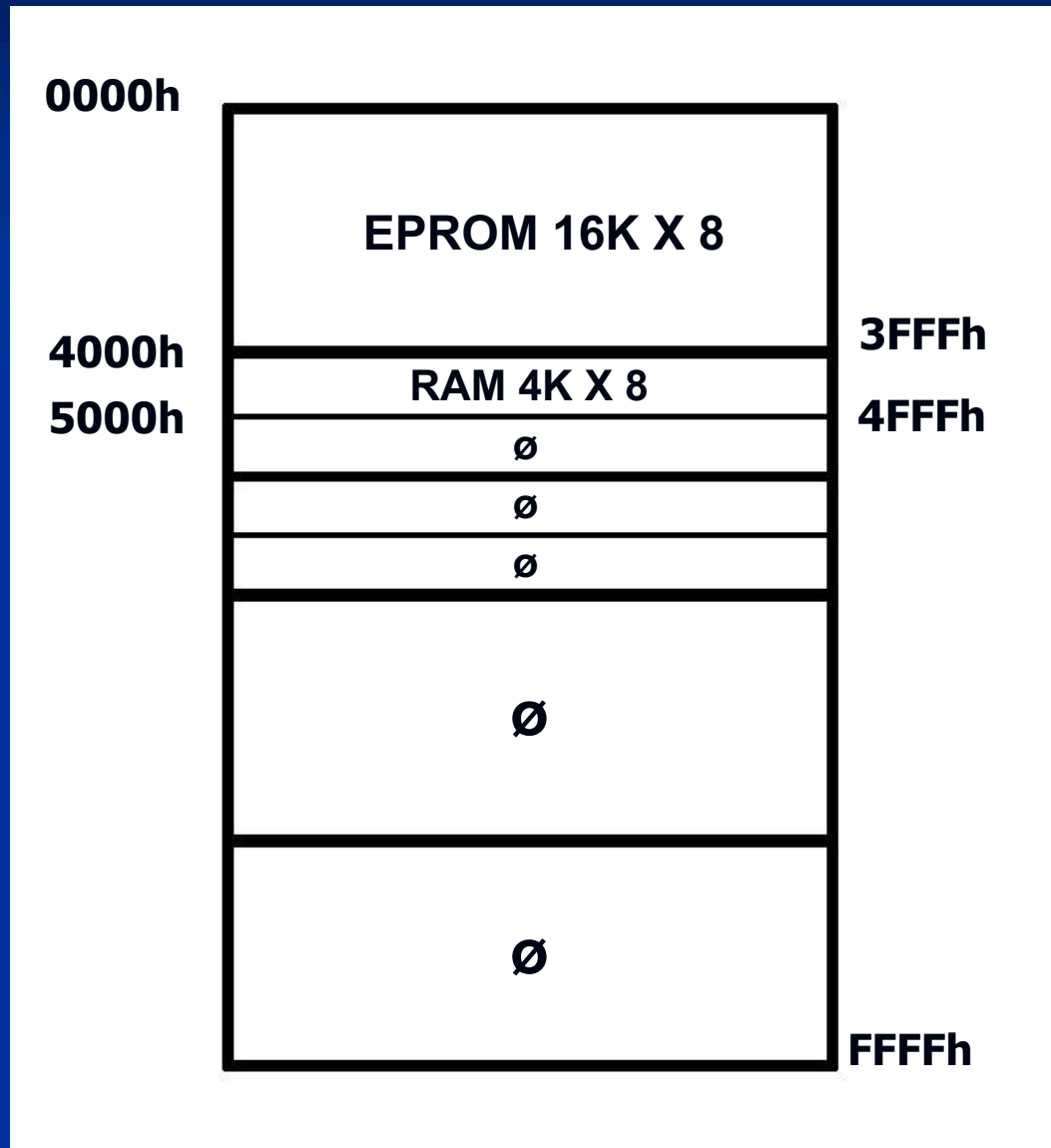


Como ligar o μP às Memórias?

Lógica de Seleção com Decodificadores



Mapeamento das Memórias



LÓGICA DE SELEÇÃO COM DECODIFICADORES

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 divide o espaço de endereço em blocos menores.
- O tamanho da divisão depende de quais bits de seleção são conectados na entrada do decodificador

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

Circuito utilizado para microprocessadores com 16 pinos de endereços



Divisão: 16K
(+3FFFh)

- **Exemplo:**
 - Espaço de endereço: 0000h a FFFFh (64KBytes)
 - Tamanho da divisão: 16K, pois o bit menos significativo do decodificador é o A14
- Pode-se conectar direto nesse decodificador memórias com 16K ou 14 linhas de endereço (A0 - A13).

Saídas válidas para serem ligadas no pino /CS das mem.	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 MEMÓRIA E DE DISPOSITIVOS DE I/O

Cada saída válida do decodificador pode ser ligada diretamente no pino de habilitação da memória (/CS), e tem a ela associada uma faixa de endereço da lógica de seleção, determinada pelos bits de endereço conectados nas entradas deste decodificador.

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 MEMÓRIA E DE 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 (16K)

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

Y2 = 8000h a BFFFh Y3 = C000h a FFFFh

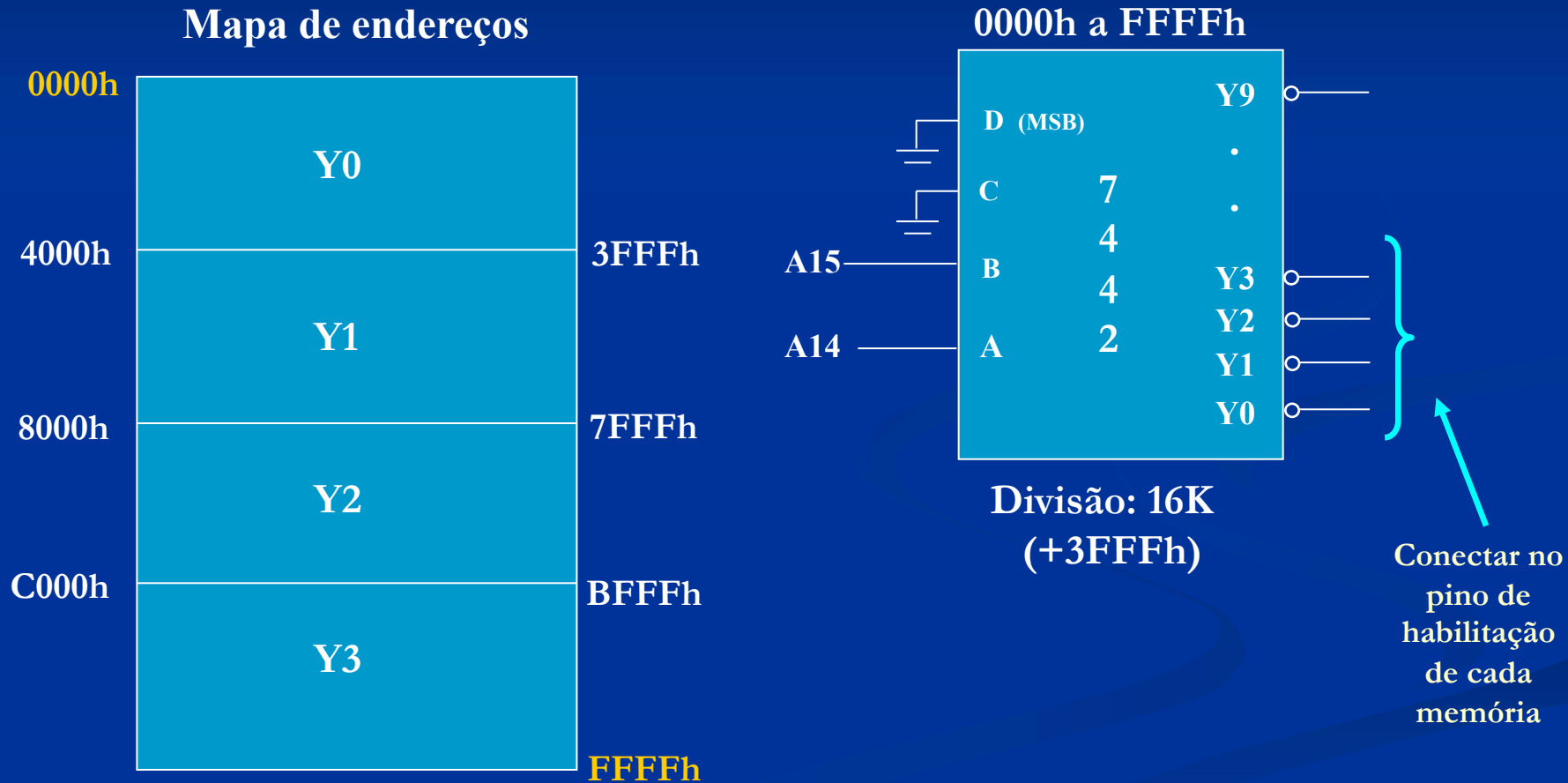
Bits de seleção

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

		A15			A0
Saída Y0	endereço inicial: 0000h	→ 0000	0000	0000	0000	
	endereço final : 3FFFh	→ 0011	1111	1111	1111	
Saída Y1	endereço inicial: 4000h	→ 0100	0000	0000	0000	
	endereço final: 7FFFh	→ 0111	1111	1111	1111	
Saída Y2	endereço inicial: 8000h	→ 1000	0000	0000	0000	
	endereço final : BFFFh	→ 1011	1111	1111	1111	
Saída Y3	endereço inicial: C000h	→ 1100	0000	0000	0000	
	endereço final : DFFFh	→ 1111	1111	1111	1111	

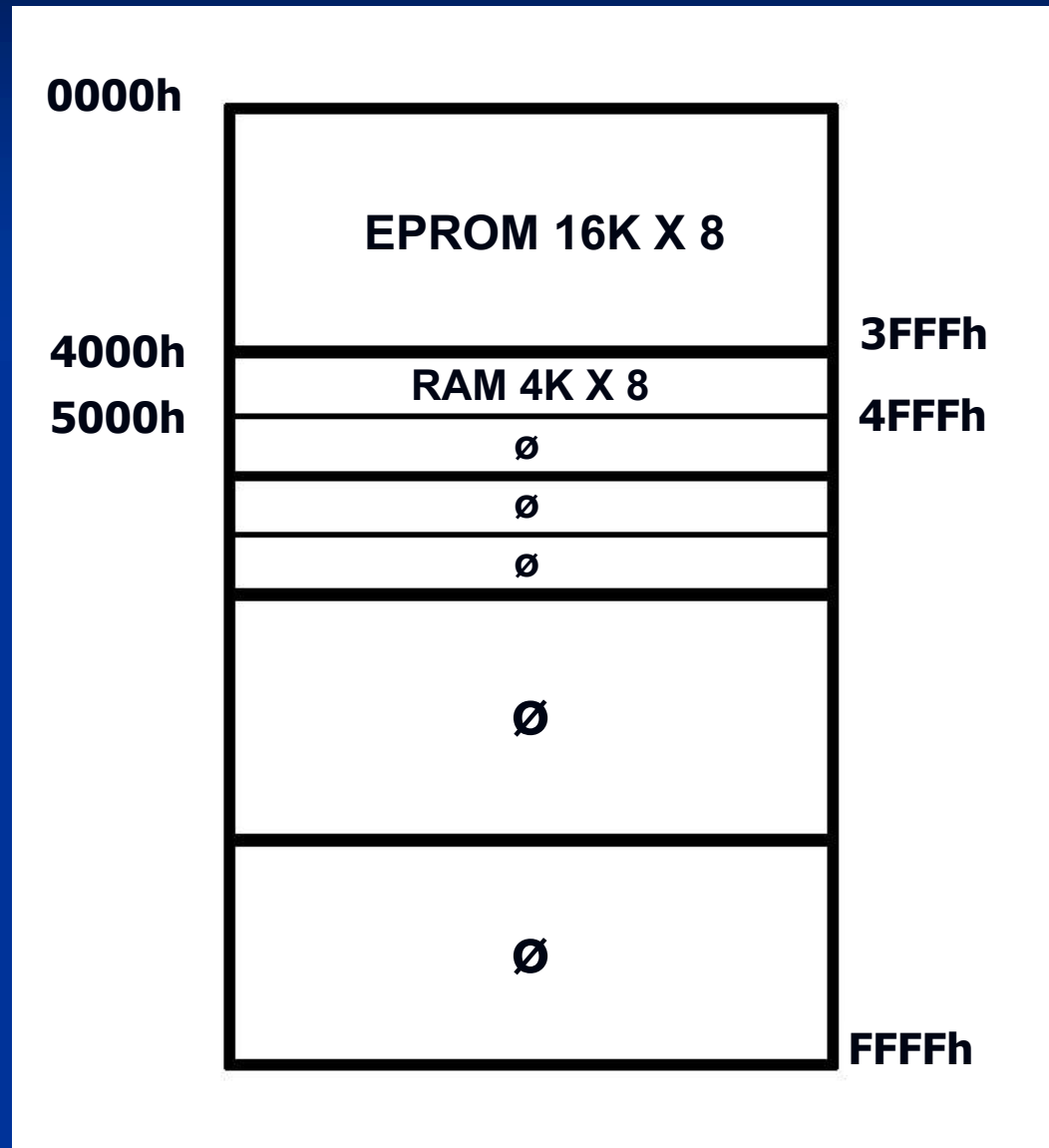
LÓGICA DE SELEÇÃO DE MEMÓRIA E DE DISPOSITIVOS DE I/O

Circuito utilizado para microprocessadores com 16 pinos de endereços

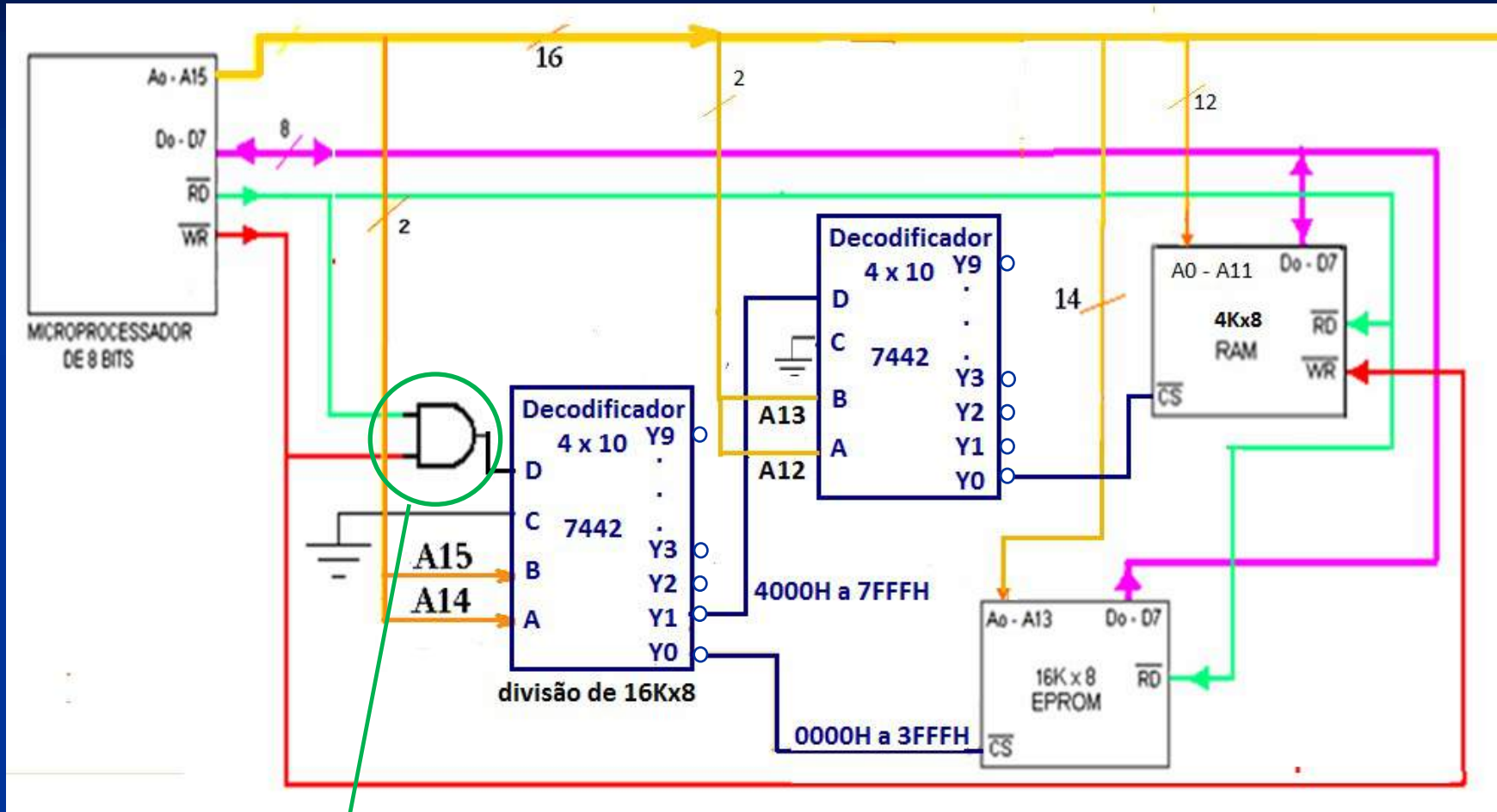


Mapeamento

1º Exemplo:



Lógica de seleção utilizando decodificadores

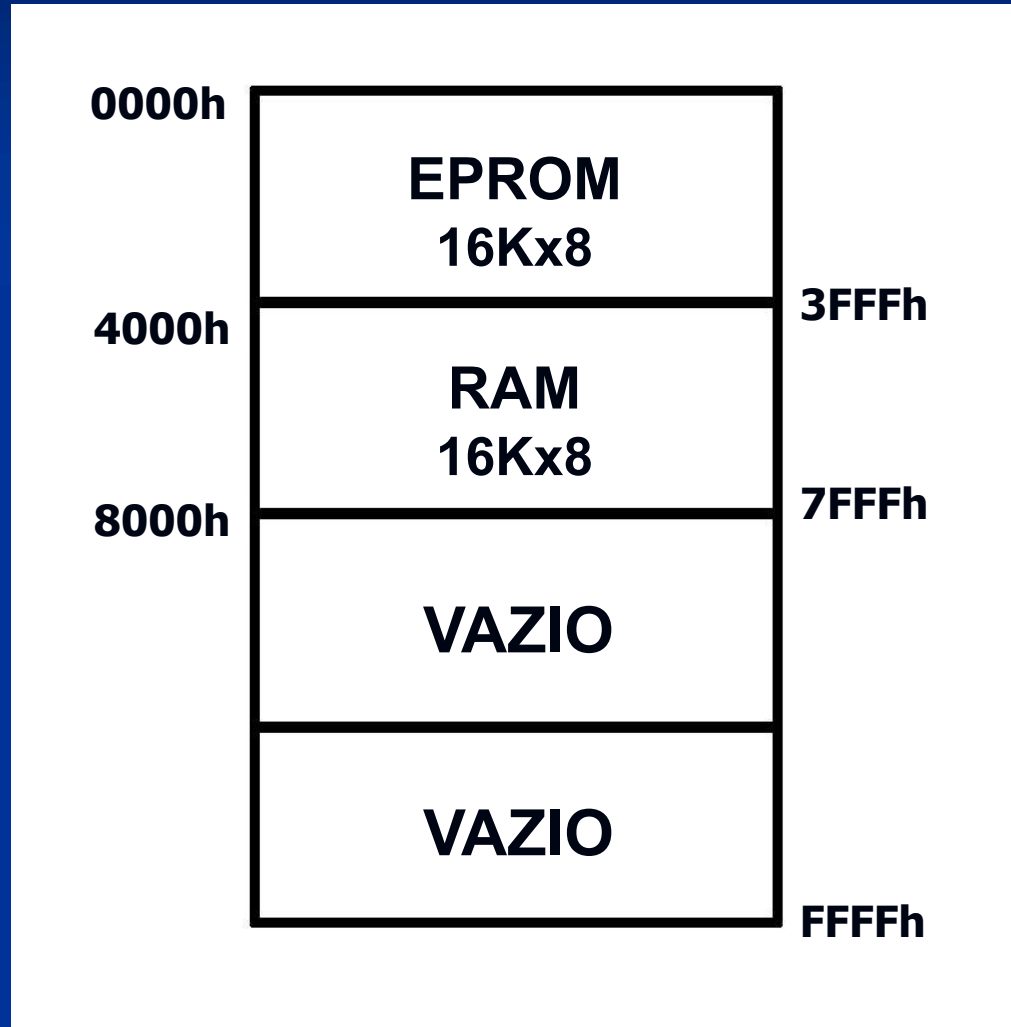


Porta AND de proteção: só seleciona os dispositivos quando o microprocessador estiver fazendo uma leitura (/RD) ou escrita (/WR).

Pode ser usada também para definir em qual decodificador será ligado memórias do tipo ROM e do tipo RAM.

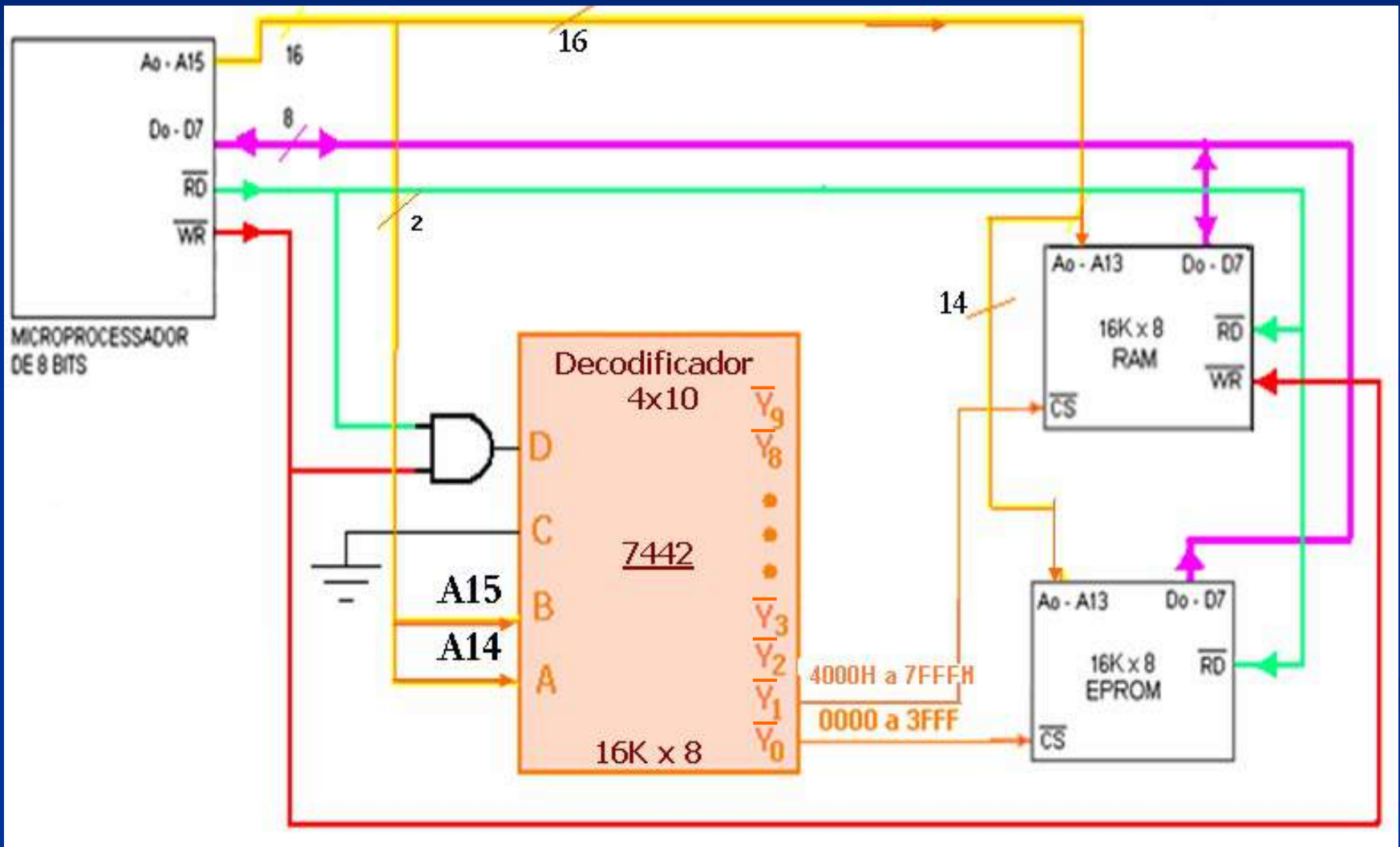
Mapeamento

2º Exemplo:



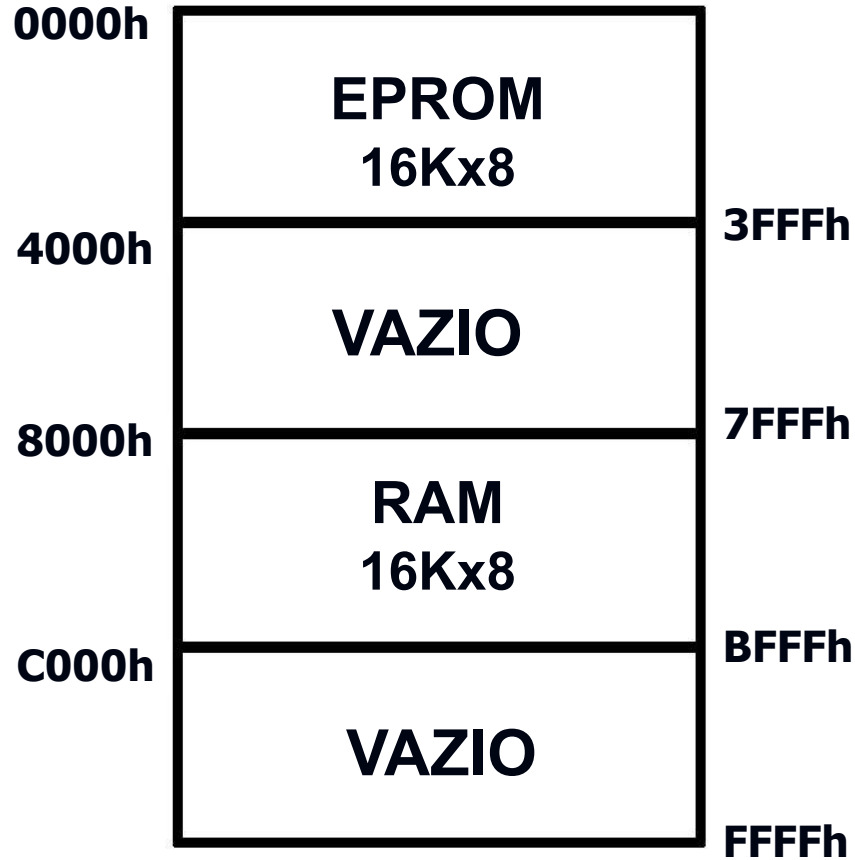
Lógica de Seleção

2º Exemplo:



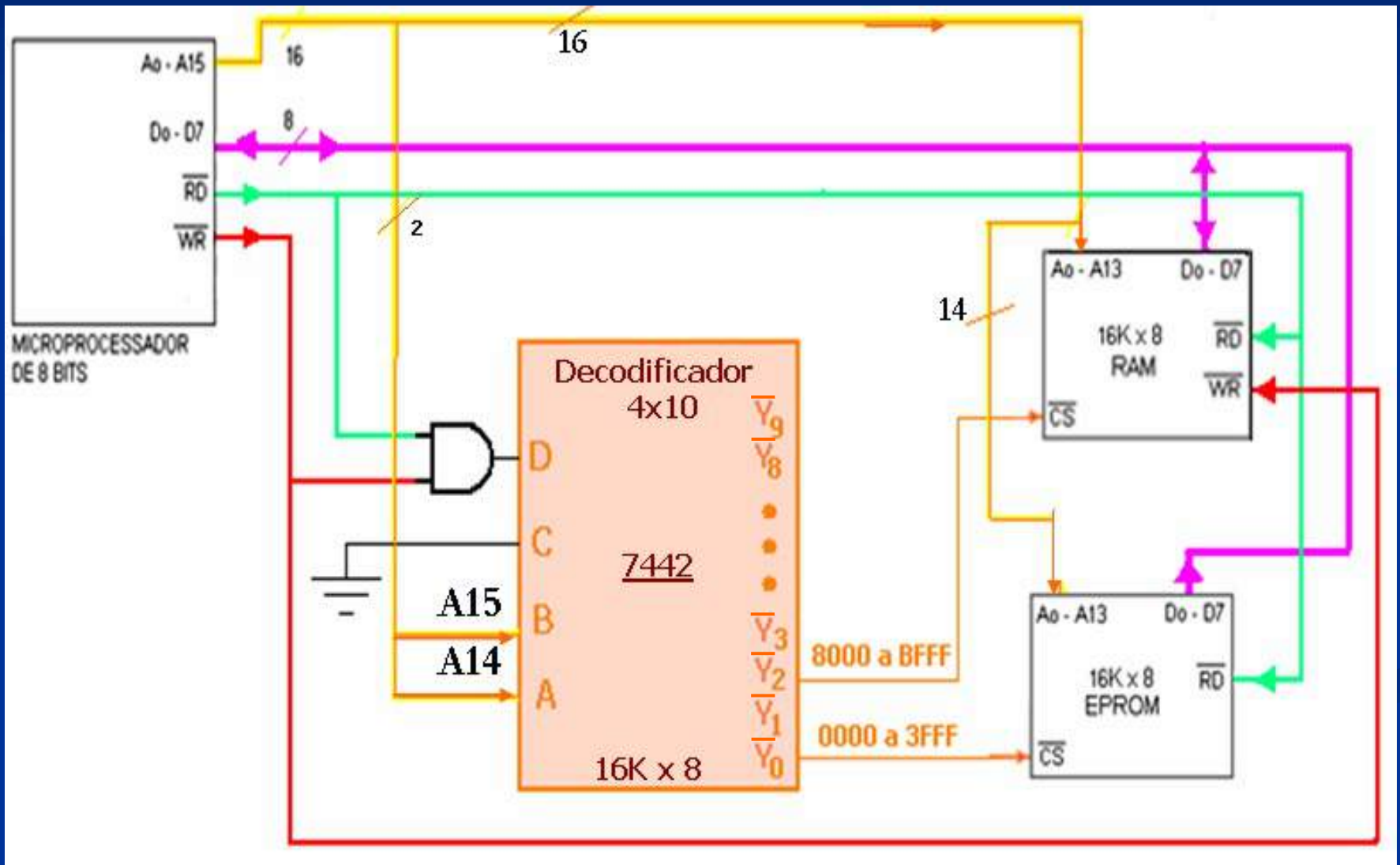
Mapeamento

3º Exemplo:



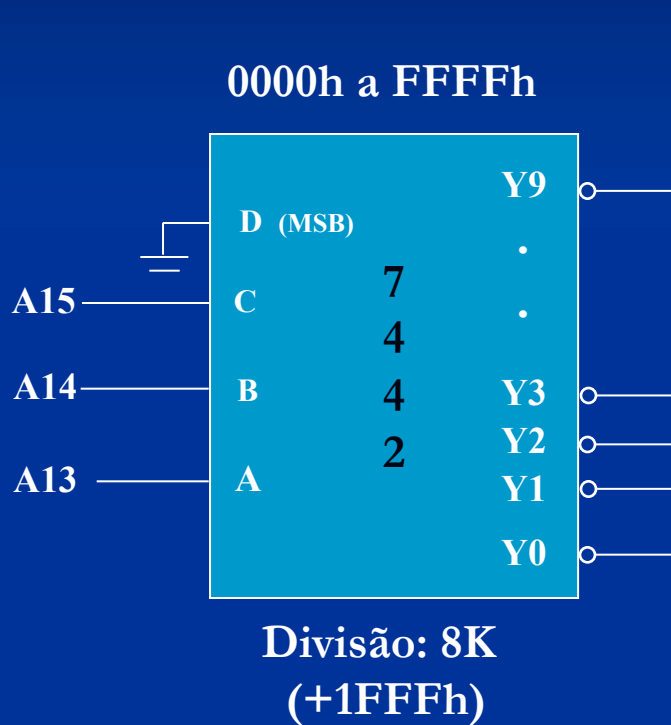
Lógica de Seleção

3.o exemplo:

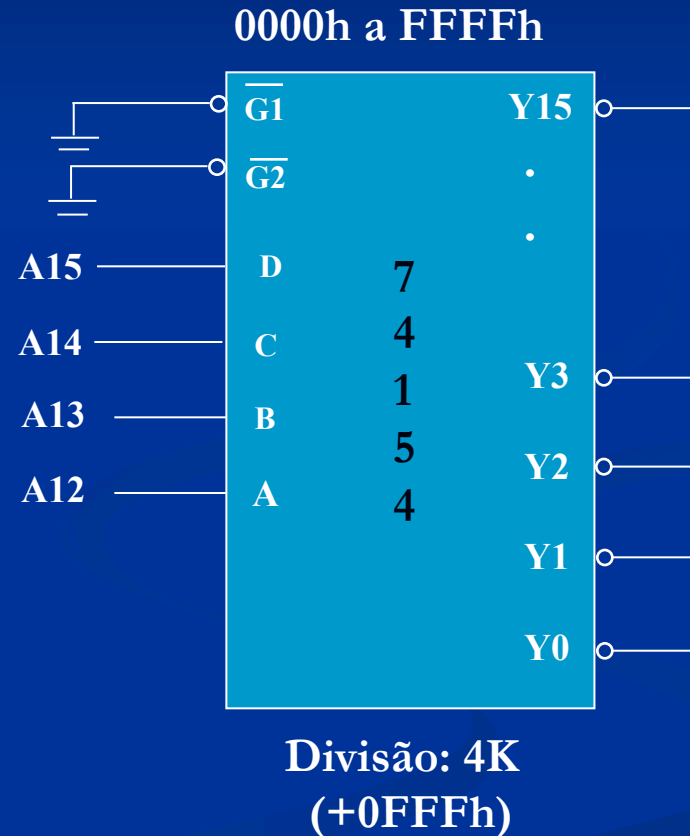


Outros Exemplos

Circuito utilizado para microprocessadores com 16 pinos de endereços.



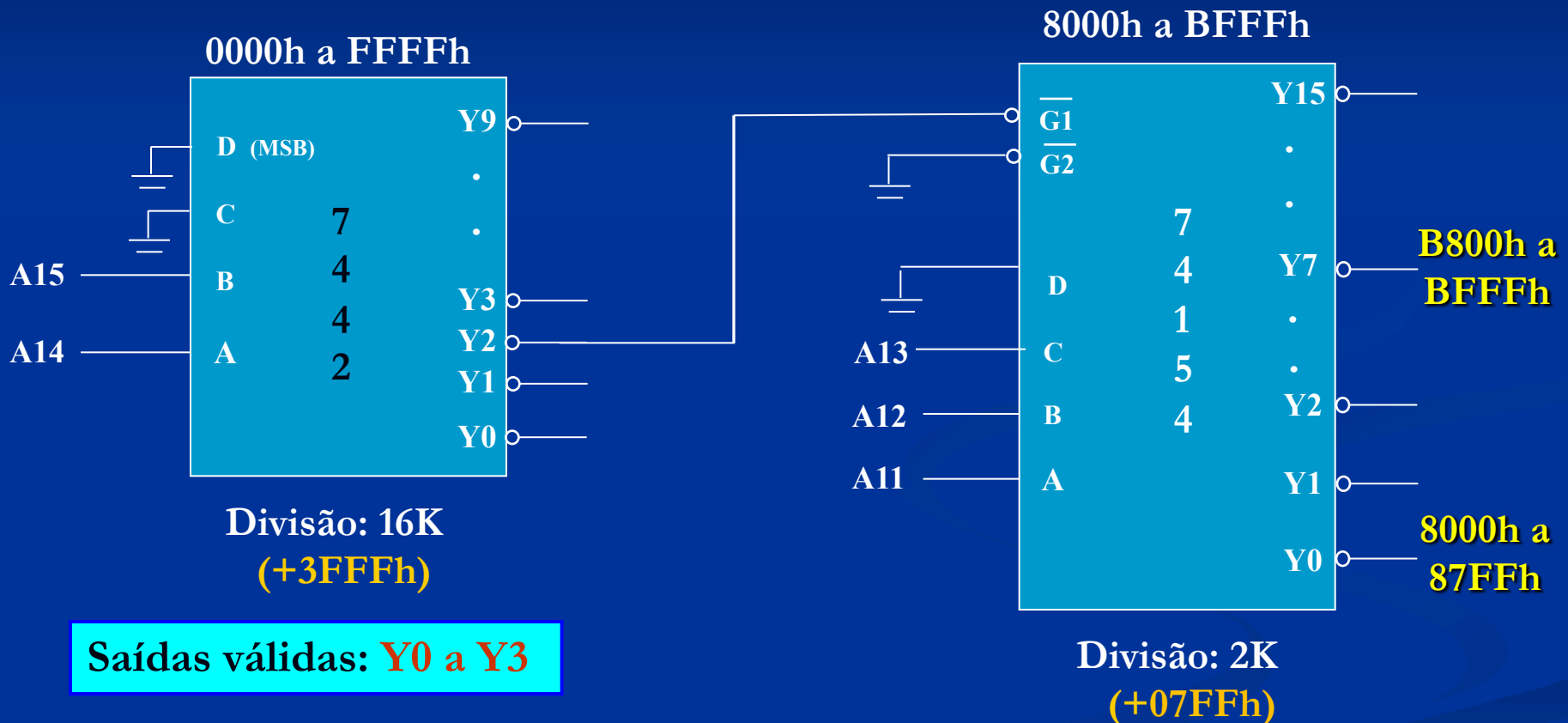
Saídas válidas: **Y0 a Y7**



Saídas válidas: **Y0 a Y15**

Outros Exemplos

Circuito utilizado para microprocessadores com 16 pinos de endereços.



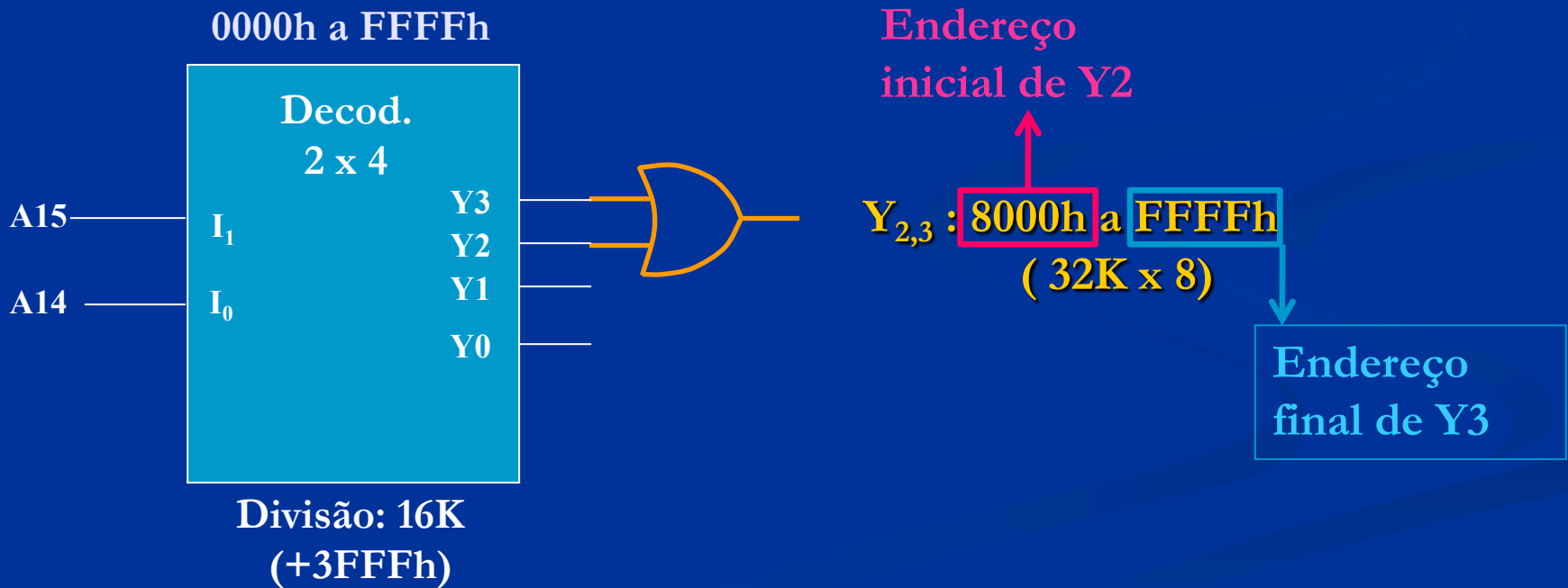
Saídas válidas: Y0 a Y3

Saídas válidas: Y0 a Y7

Outros Exemplos

Circuito utilizado para microprocessadores com 16 pinos de endereços.

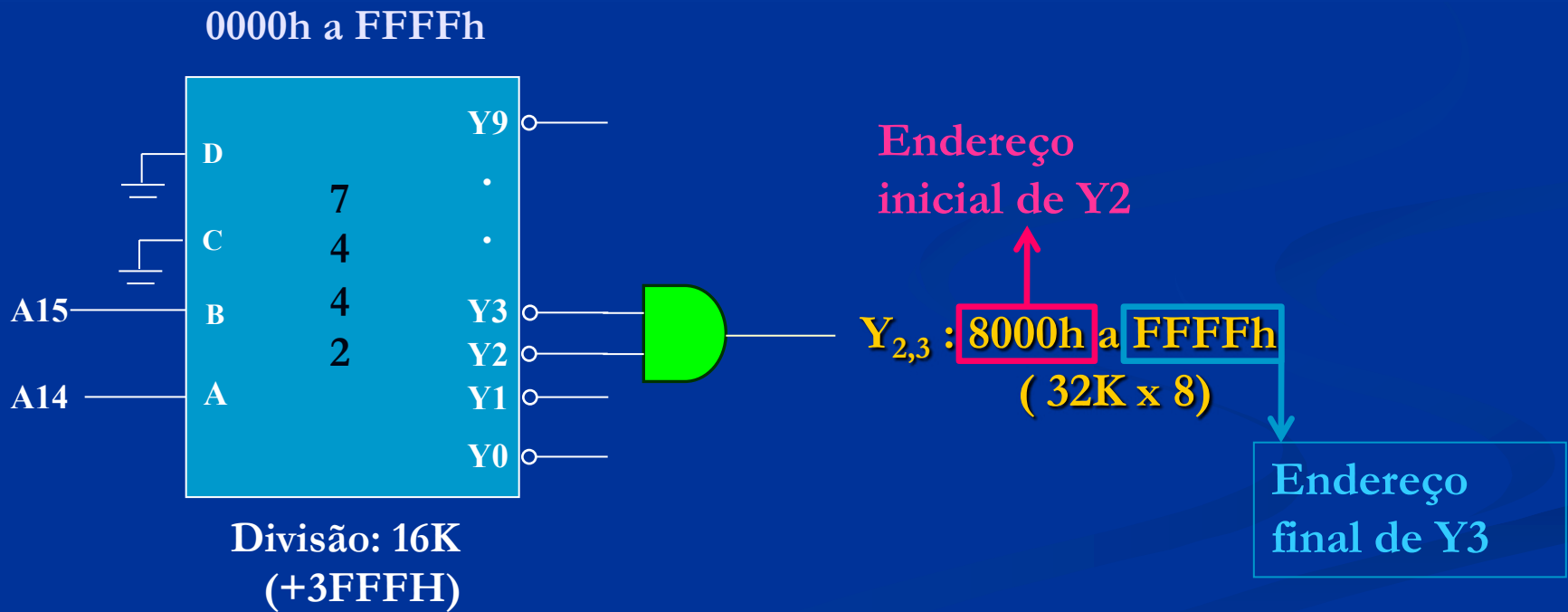
Pode-se combinar linhas de seleção através da lógica OR (para decodificadores com saída normal) ou lógica AND (para decodificadores com saída barrada) para selecionar memória MAIORES do que a faixa de endereço das saídas de seleção.



Outros Exemplos

Circuito utilizado para microprocessadores com 16 pinos de endereços.

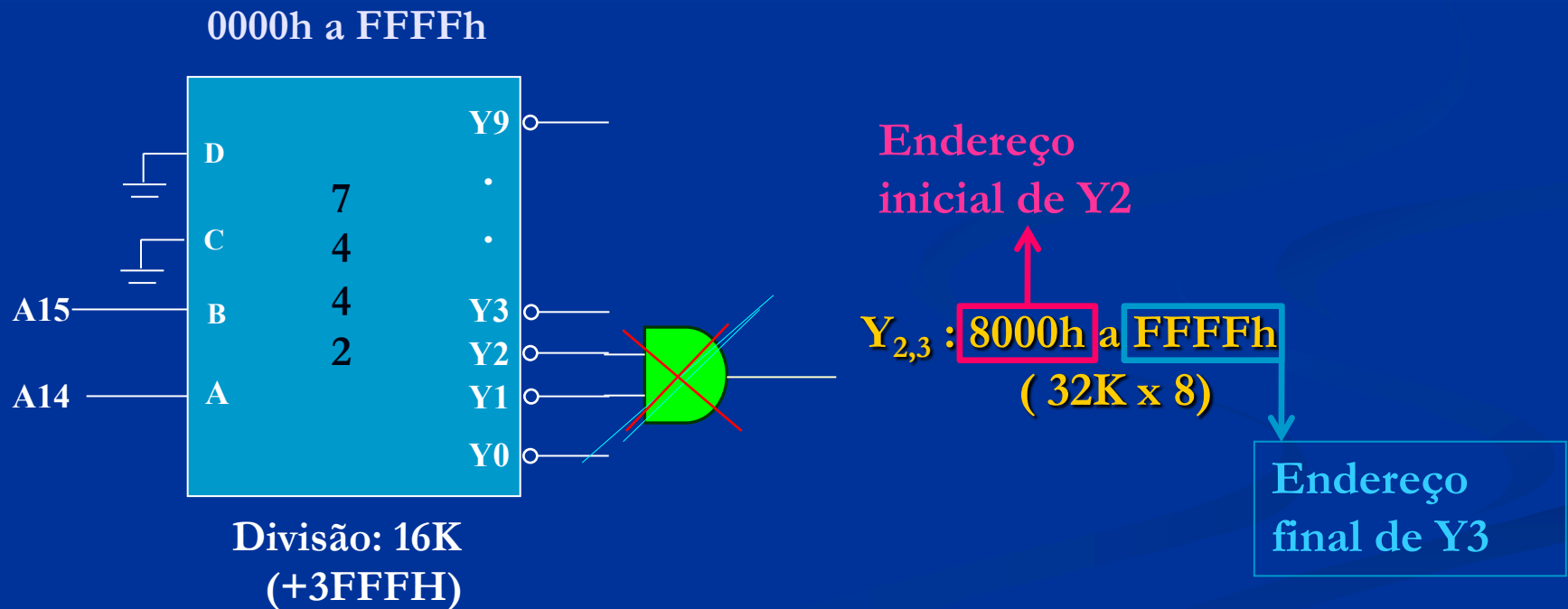
Pode-se combinar linhas de seleção através da lógica OR (para decodificadores com saída normal) ou lógica AND (para decodificadores com saída barrada) para seleccionar memória MAIORES do que a faixa de endereço das saídas de seleção.



CUIDADO!

Circuito utilizado para microprocessadores com 16 pinos de endereços.

Pode-se combinar linhas de seleção através da lógica OR (para decodificadores com saída normal) ou lógica AND (para decodificadores com saída barrada) para seleccionar memória **MAIORES** do que a faixa de endereço das saídas de seleção.



Tipos de Lógica de Seleção

Tipos de Lógica de Seleção

- Modos de seleção:

- **DECODIFICAÇÃO ABSOLUTA**

- **DECODIFICAÇÃO NÃO-ABSOLUTA**

Decodificação Absoluta

- Todos os bits do barramento de endereço do microprocessador são utilizados na lógica de seleção;
- São selecionados espaços que têm exatamente as dimensões do chip
- O chip de memória é selecionado por um único bloco de endereços na lógica de seleção.

Decodificação 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

Decodificação Absoluta

Ex.1: Não preenchendo o espaço de endereçamento do μ P:

ROM 16 k x 8

RAM 16 k x 8

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			
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			



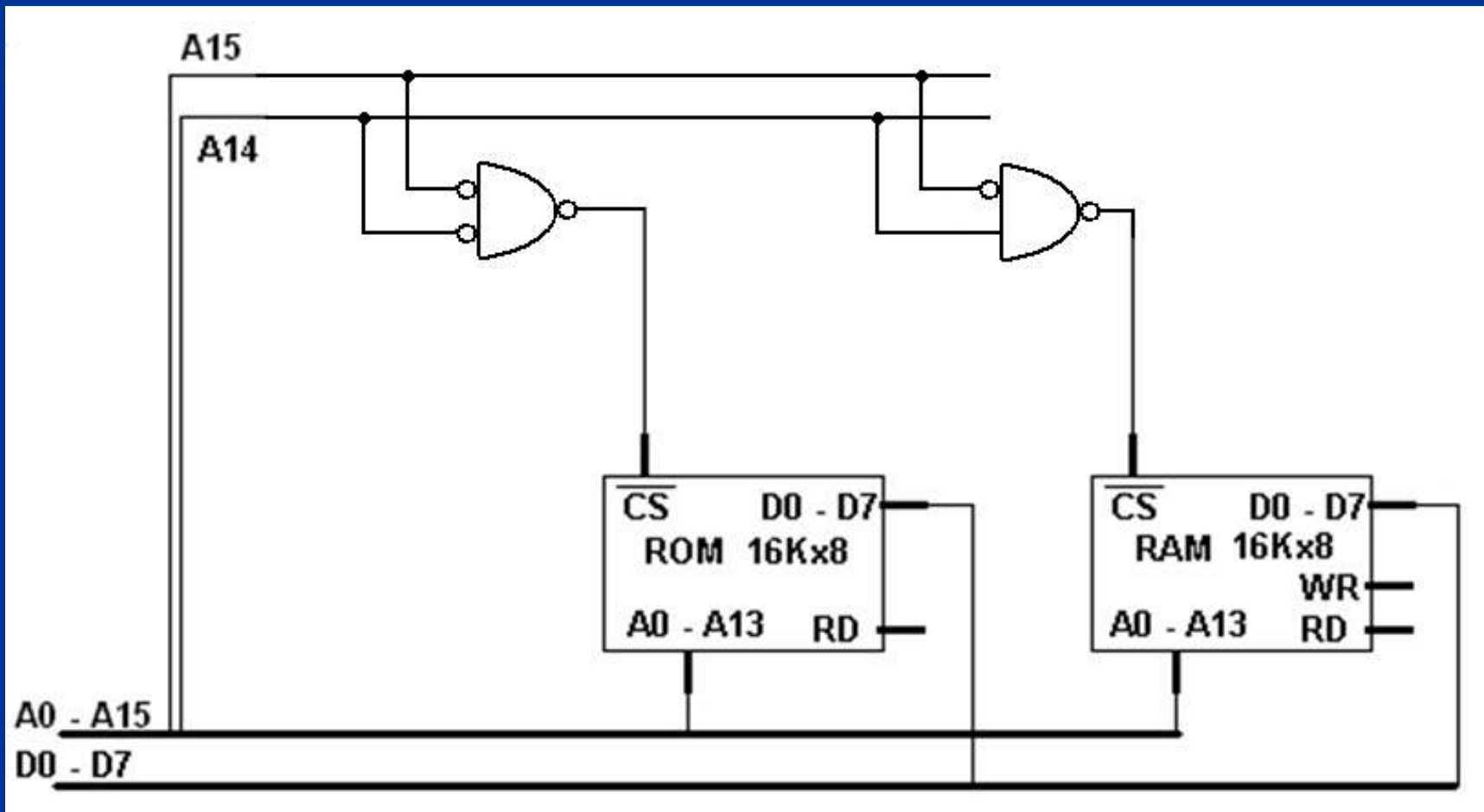
Utiliza-se todos os bits do duto de endereços do μ P para a seleção.

Decodificação Absoluta

Ex.1: Não preenchendo o espaço de endereçamento do μP :
utilizando portas lógicas

ROM 16 k x 8

RAM 16 k x 8

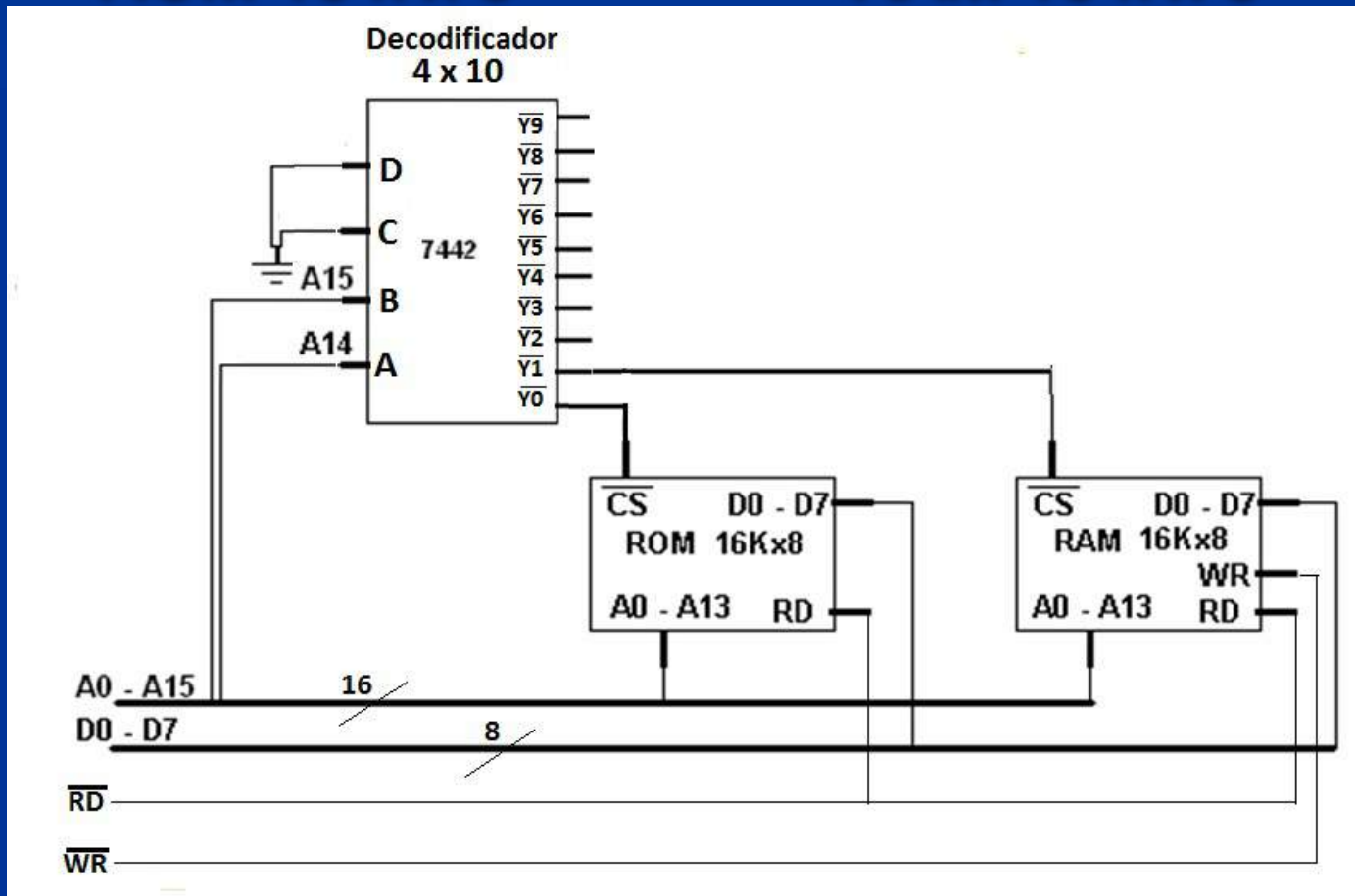


Decodificação Absoluta

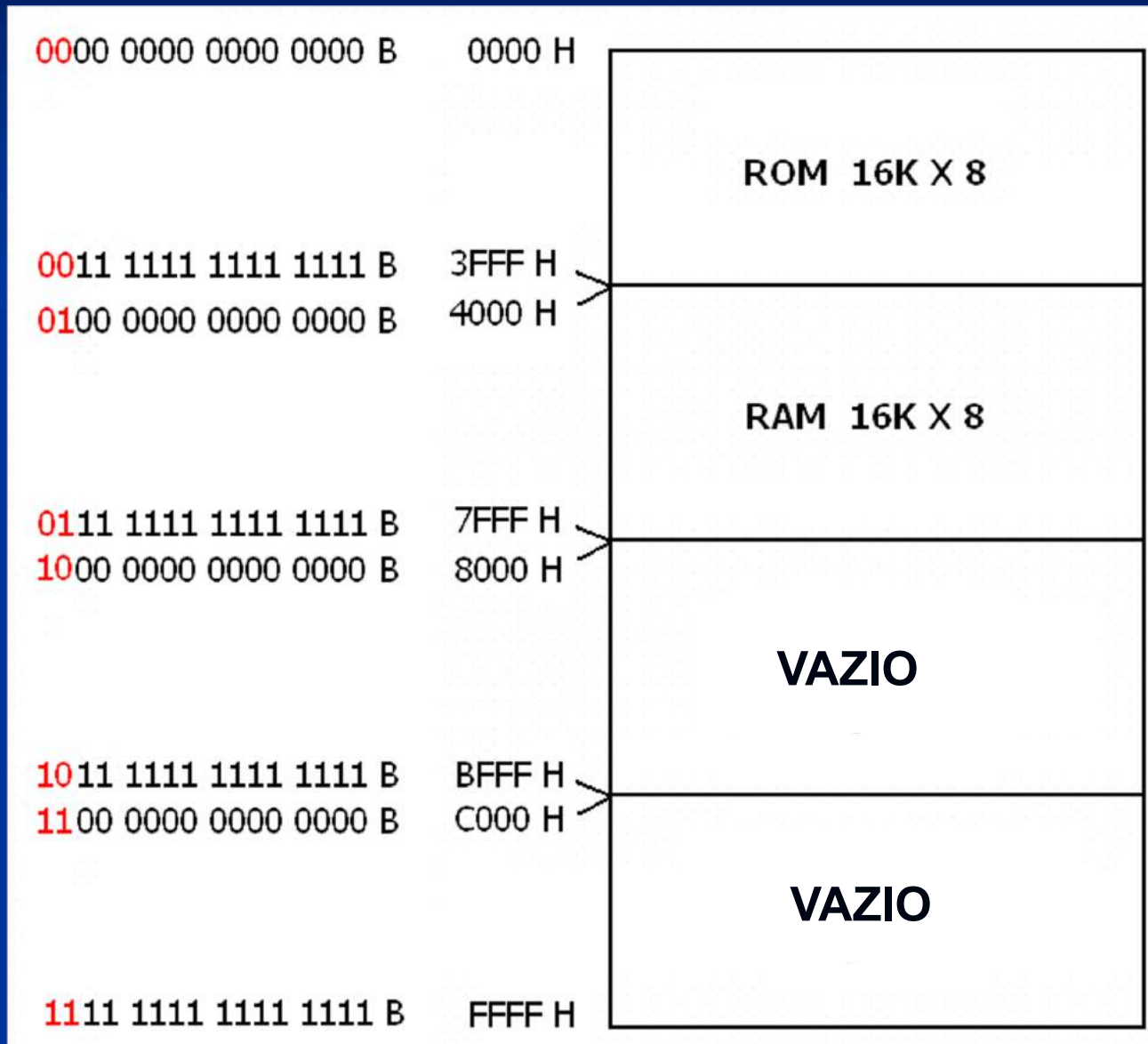
Ex.2: Não preenchendo o espaço de endereçamento do μP :
utilizando decodificador

ROM 16 k x 8

RAM 16 k x 8



Mapeamento da Memória

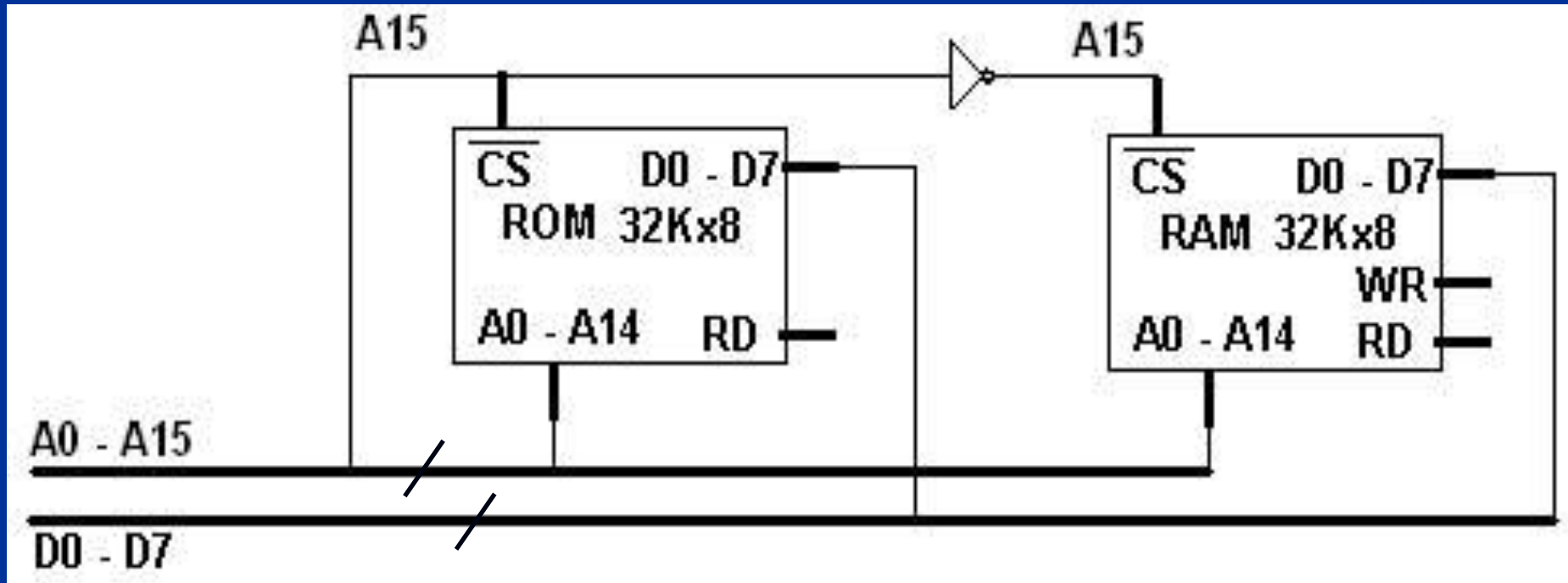


Decodificação Absoluta

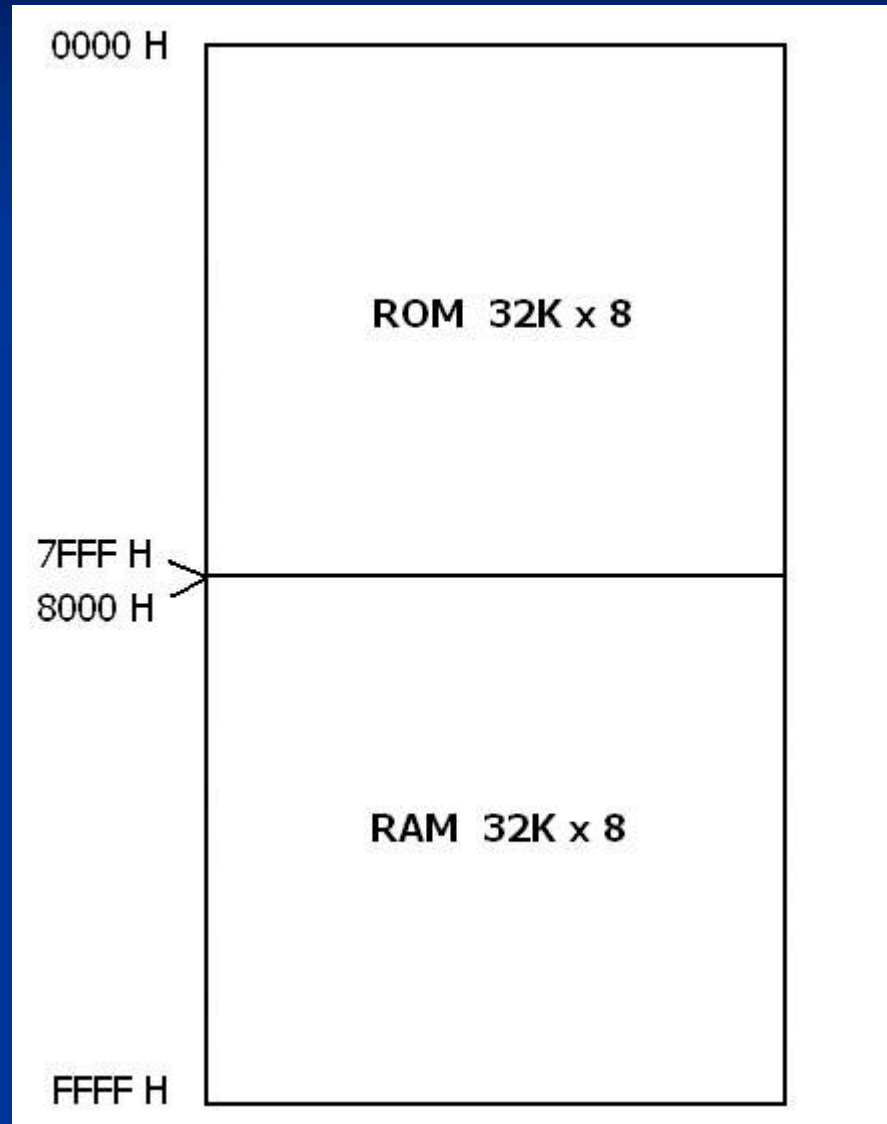
Ex.2: Preenchendo todo o espaço de endereçamento com 2 memórias:

ROM 32 k x 8

RAM 32 k x 8



Mapeamento da Memória



Decodificação Não Absoluta

- Apenas alguns dos bits do barramento de endereço do MP são usados na lógica de seleção do chip de memória
- Consequência → CI pode ser selecionado por mais de um bloco de endereços
- OBS.: cuidado para 2 (ou +) dispositivos não serem selecionados ao mesmo tempo → conflitos no duto de dados

Decodificação Não Absoluta

- 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 sequenciais;
 - os endereços fantasmas não podem ser usados por outros chips, pois não são espaços livres, mas sim endereços diferentes que selecionam a mesma célula na memória;

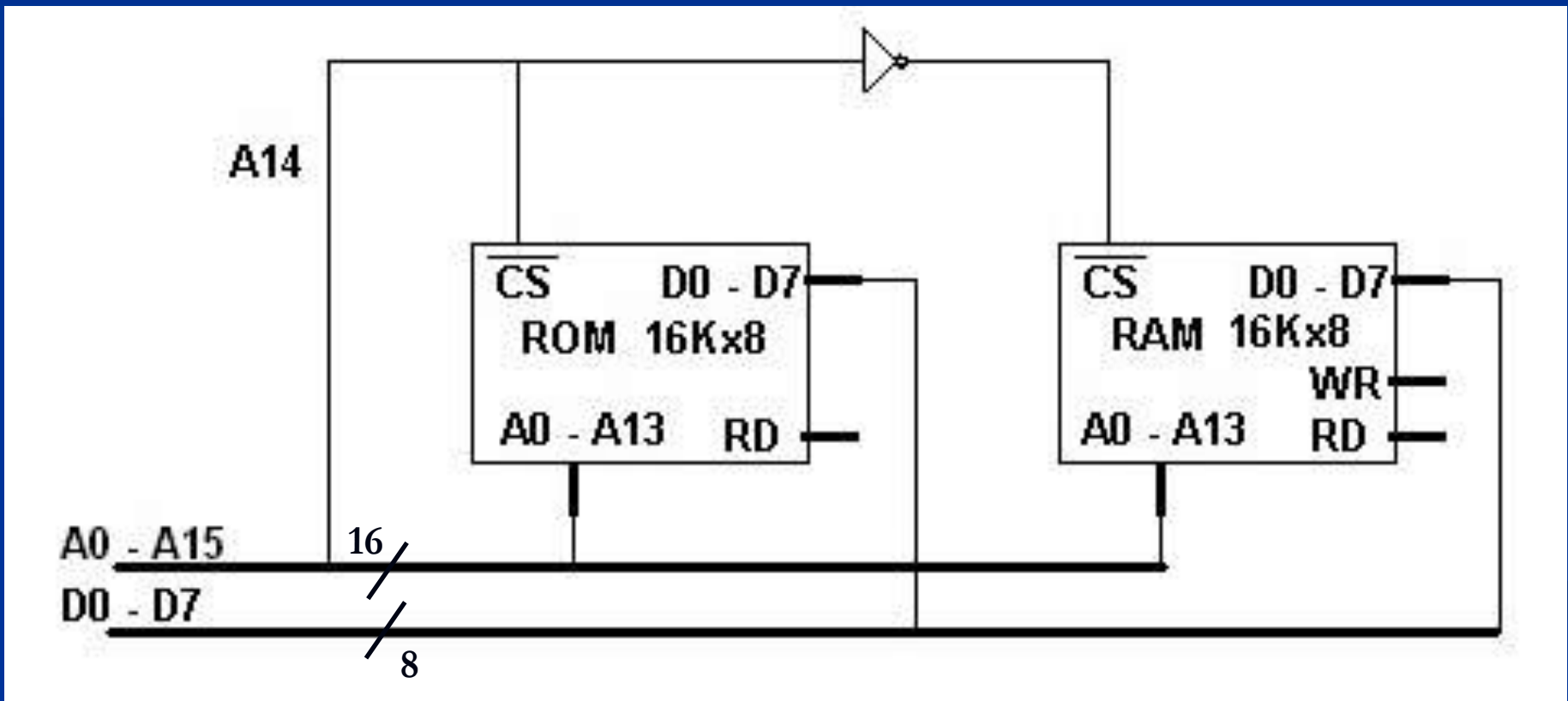
Decodificação Não Absoluta

Ex.1:

Endereçamento de 2 memórias:

ROM 16 k x 8

RAM 16 k x 8



Bit de endereço A15 é irrelevante

Lógica de Seleção – Não Absoluta

Lógica de Seleção do μ P – 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	3FFF
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
RAM	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4000	16k	7FFF
	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
ROM "Espelho"	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8000	16k	BFFF
	1	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1			
RAM "Espelho"	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C000	16k	FFFF
	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			

Bit de endereço A15 é irrelevante pode ser '1' ou '0'

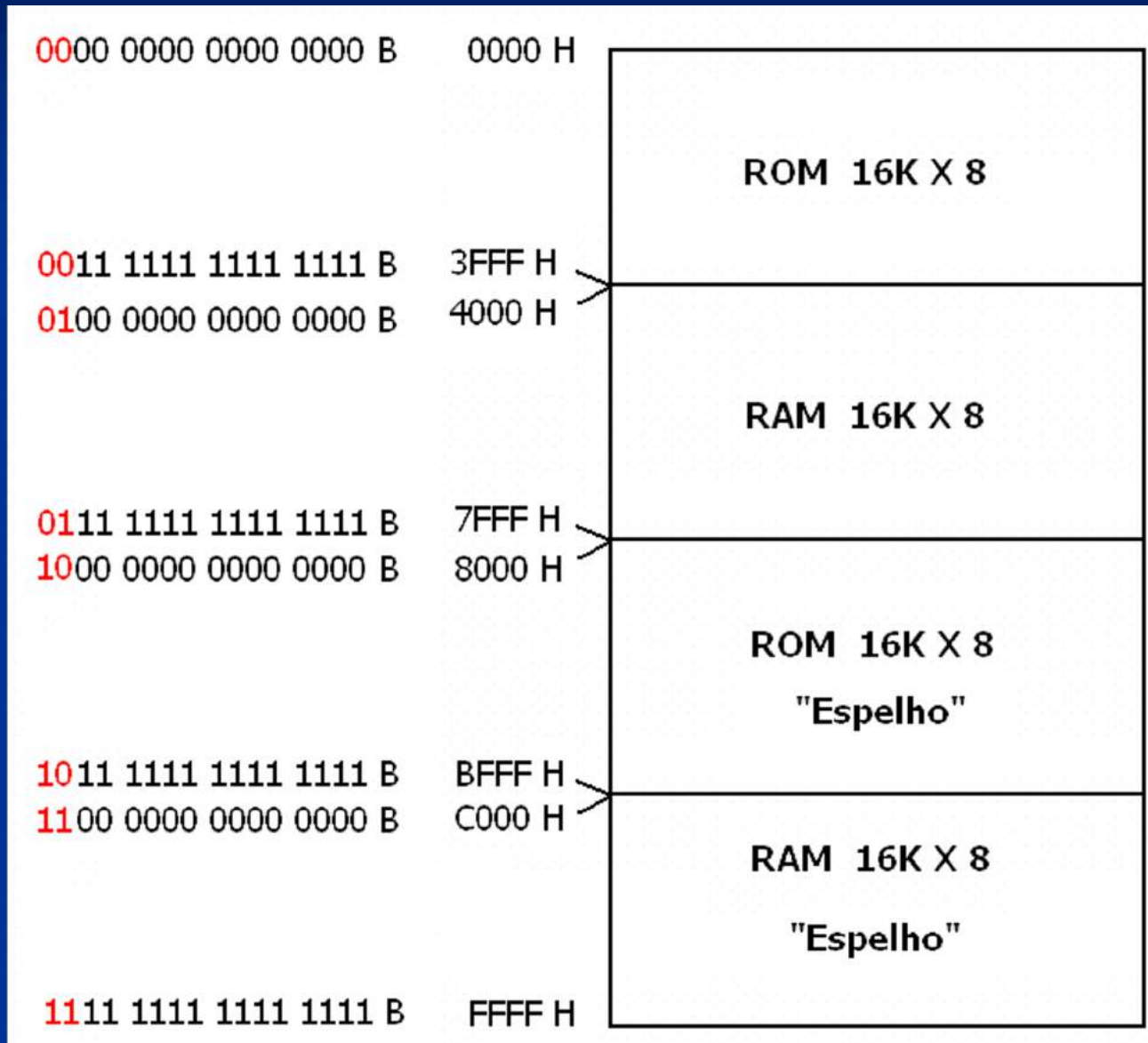
Lógica de Seleção – Não Absoluta

Se $A15 = 0$ a faixa de endereço = 0000H a 7FFFH

Se $A15 = 1$ a faixa de endereço = 8000H a 8FFFH

- Um dos espaços de 16k x 8 é denominado de espaço fantasma.
- Qualquer uma das duas faixas de endereço pode ser escolhida como a fantasma.
- Endereço 0000H e 8000H endereçam a mesma posição física do CHIP (linhas de endereçamento do chip são iguais a zero para esses dois endereços).

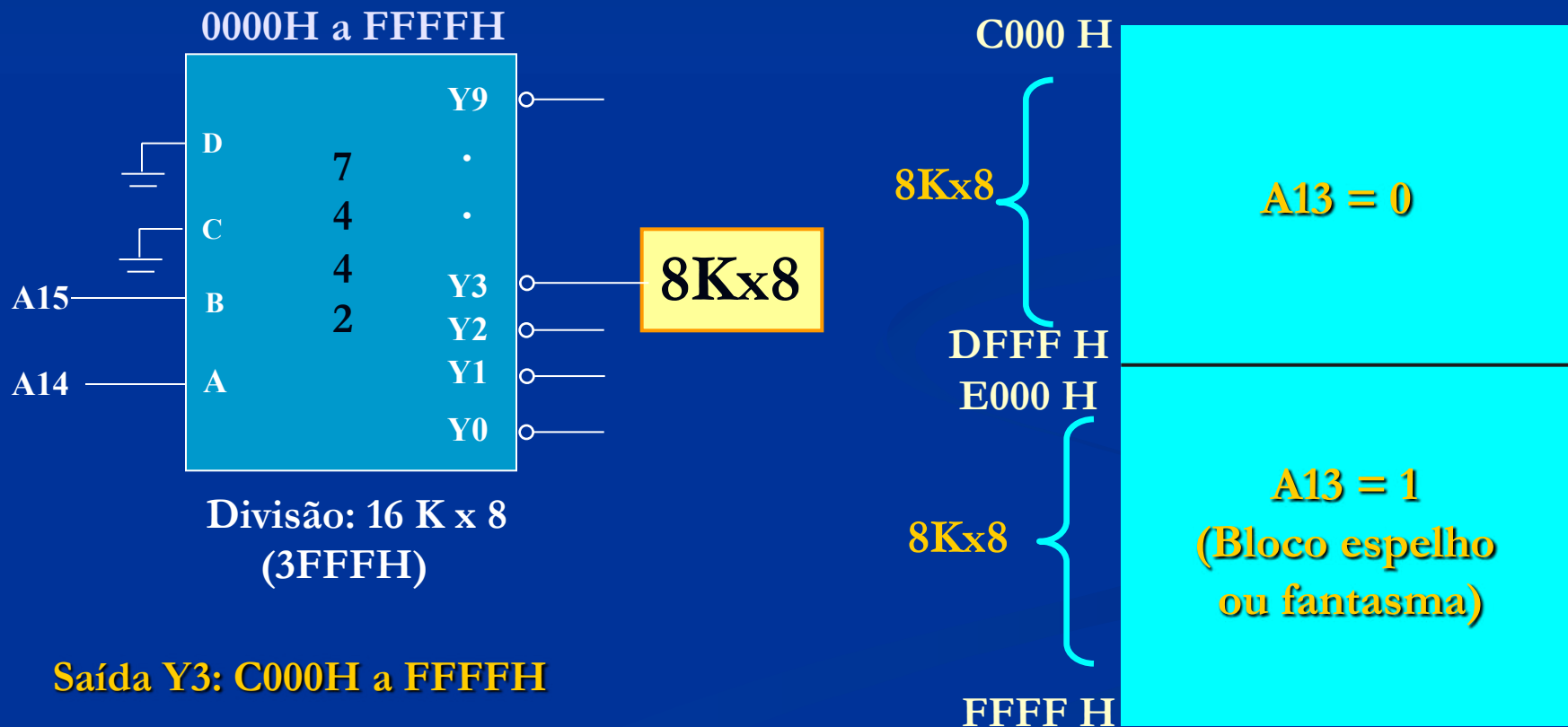
Mapeamento da Memória



Decodificação Não Absoluta

Ligar uma memória com organização menor que a da saída de seleção de um decodificador também gera decodificação não-absoluta

Exemplo 2:



Decodificação Não Absoluta

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.

Decodificação Não Absoluta

Memória de dados (SRAM) do microcontrolador PIC16F877

FIGURE 2-3: PIC16F876A/877A REGISTER FILE MAP

File Address	File Address	File Address	File Address
Indirect addr. ⁽¹⁾ 00h	Indirect addr. ⁽¹⁾ 80h	Indirect addr. ⁽¹⁾ 100h	Indirect addr. ⁽¹⁾ 180h
TMR0 01h	OPTION_REG 81h	TMR0 101h	OPTION_REG 181h
PCL 02h	PCL 82h	PCL 102h	PCL 182h
STATUS 03h	STATUS 83h	STATUS 103h	STATUS 183h
FSR 04h	FSR 84h	FSR 104h	FSR 184h
PORTA 05h	TRISA 85h		
PORTB 06h	TRISB 86h	PORTB 106h	TRISB 186h
PORTC 07h	TRISC 87h		
PORTD ⁽¹⁾ 08h	TRISD ⁽¹⁾ 88h		
PORTE ⁽¹⁾ 09h	TRISE ⁽¹⁾ 89h		
PCLATH 0Ah	PCLATH 8Ah	PCLATH 10Ah	PCLATH 18Ah
INTCON 0Bh	INTCON 8Bh	INTCON 10Bh	INTCON 18Bh
PIR1 0Ch	PIE1 8Ch	EEDATA 10Ch	EECON1 18Ch
PIR2 0Dh	PIE2 8Dh	EEADR 10Dh	EECON2 18Dh
TMR1L 0Eh	PCON 8Eh	EEDATH 10Eh	Reserved ⁽²⁾ 18Eh
TMR1H 0Fh		EEADRH 10Fh	Reserved ⁽²⁾ 18Fh
T1CON 10h			
TMR2 11h	SSPCON2 91h		
T2CON 12h	PR2 92h		
SSPBUF 13h	SSPADD 93h		
SSPCON 14h	SSPSTAT 94h		
CCPR1L 15h			
CCPR1H 16h			
CCP1CON 17h			
RCSTA 18h	TXSTA 98h	General Purpose Register 16 Bytes	General Purpose Register 16 Bytes
TXREG 19h	SPBRG 99h		
RCREG 1Ah			
CCPR2L 1Bh			
CCPR2H 1Ch	CMCON 9Ch		
CCP2CON 1Dh	CVRCON 9Dh		
ADRESH 1Eh	ADRESL 9Eh		
ADCON0 1Fh	ADCON1 9Fh		
General Purpose Register 96 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes	General Purpose Register 80 Bytes
	accesses 70h-7Fh	accesses 70h-7Fh	accesses 70h-7Fh
Bank 0 7Fh	Bank 1 FFh	Bank 2 17Fh	Bank 3 1FFh

Unimplemented data memory locations, read as '0'.
 * Not a physical register.

FIM