

LISTA 6

Questão 1)

a) Na decodificação Não-Absoluta não são todos os bits do microprocessador são usados para a lógica de seleção. Por consequência, a memória pode ser acessada por dois endereços diferentes, embora seja a mesma posição fixa de memória. Cria-se assim uma “memória fantasma”, que é uma cópia da original, acessada por outro endereço. Na Absoluta, pelo contrário, cada memória possui uma única faixa de endereços. O Absoluta não há possibilidade de conflitos no espaço de endereços, por outro lado, o hardware para seleção é mais complexo.

b) Uma memória é dita alinhada quando, em sua primeira palavra, os bits de endereçamento do duto de endereços são preenchido por “0”s.

c) Um espaço vazio não contém informação. Nenhum dado ou instrução pode ser armazenado ou acessado em um espaço vazio. Um espaço fantasma, por outro lado, é a cópia de um outro espaço já preenchido por uma memória, e tudo o que for feito em uma memória ocorrerá na outra.

d) N° de endereços que podem ser acessados pelo μP : $2^{15} = 32K$

$(2 \times 8K) + (3 \times 2K) + (1 \times 16 K) = 38K > 32K$

Não é possível inserir.

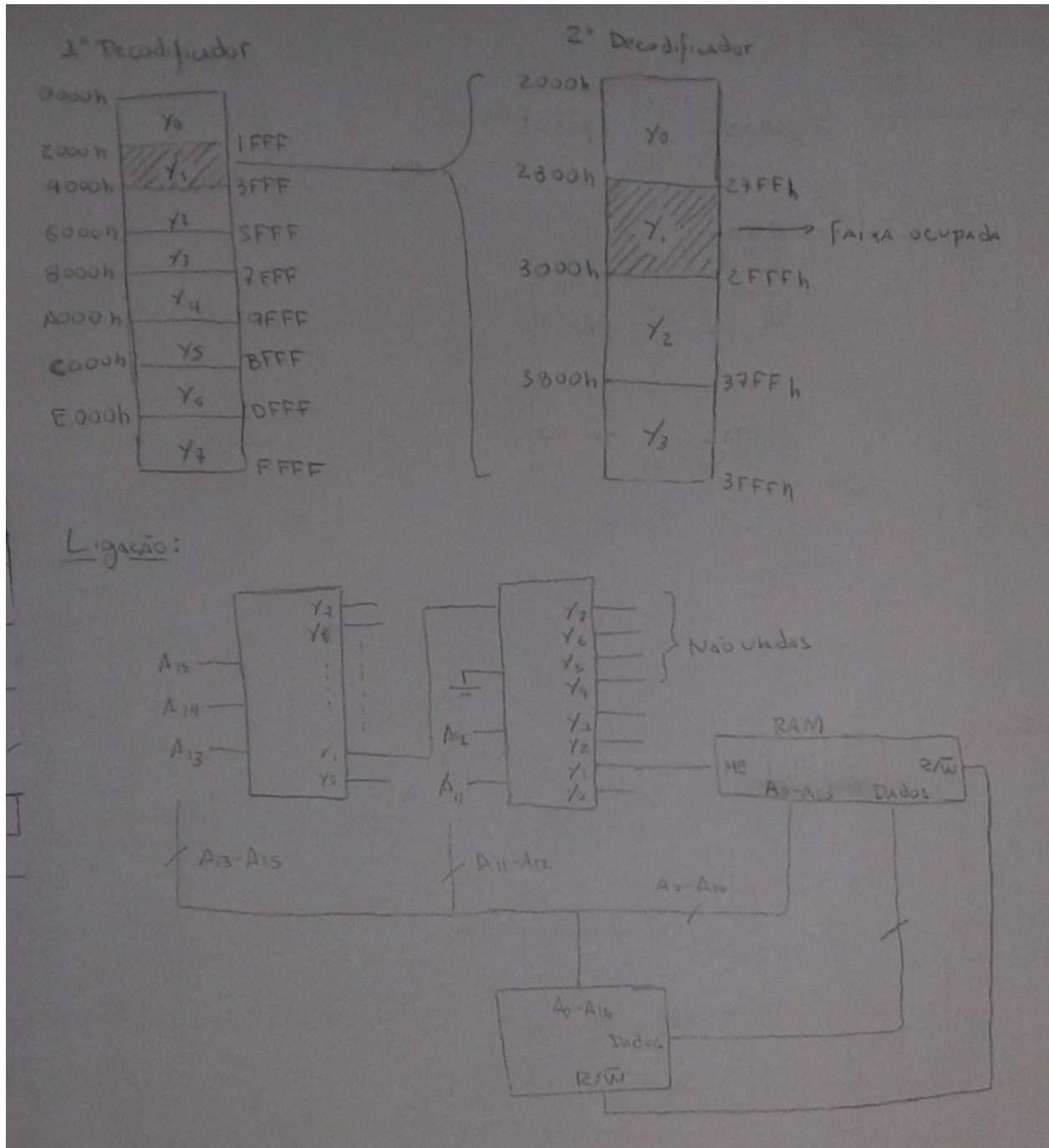
e) $64K = 2^{16} \rightarrow 16$ bits de endereço do microprocessador.

$2K = 2^{11} \rightarrow 11$ bits de endereçamento.

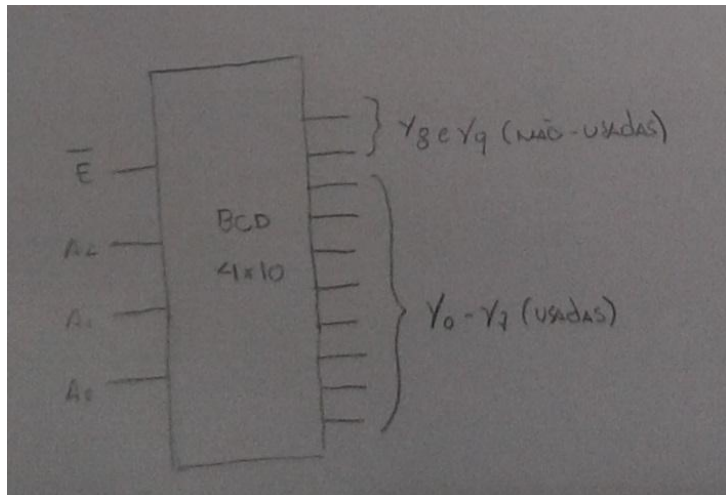
$16 - 11 = 5$ bits de seleção.

A faixa ocupada pela memória é de 2K ou 0800_h endereços. Inserida no início, ocupará a faixa: 0000 0000 0000 0000 até 0000 0111 1111 1111 \rightarrow 0000_h à 07FF_h

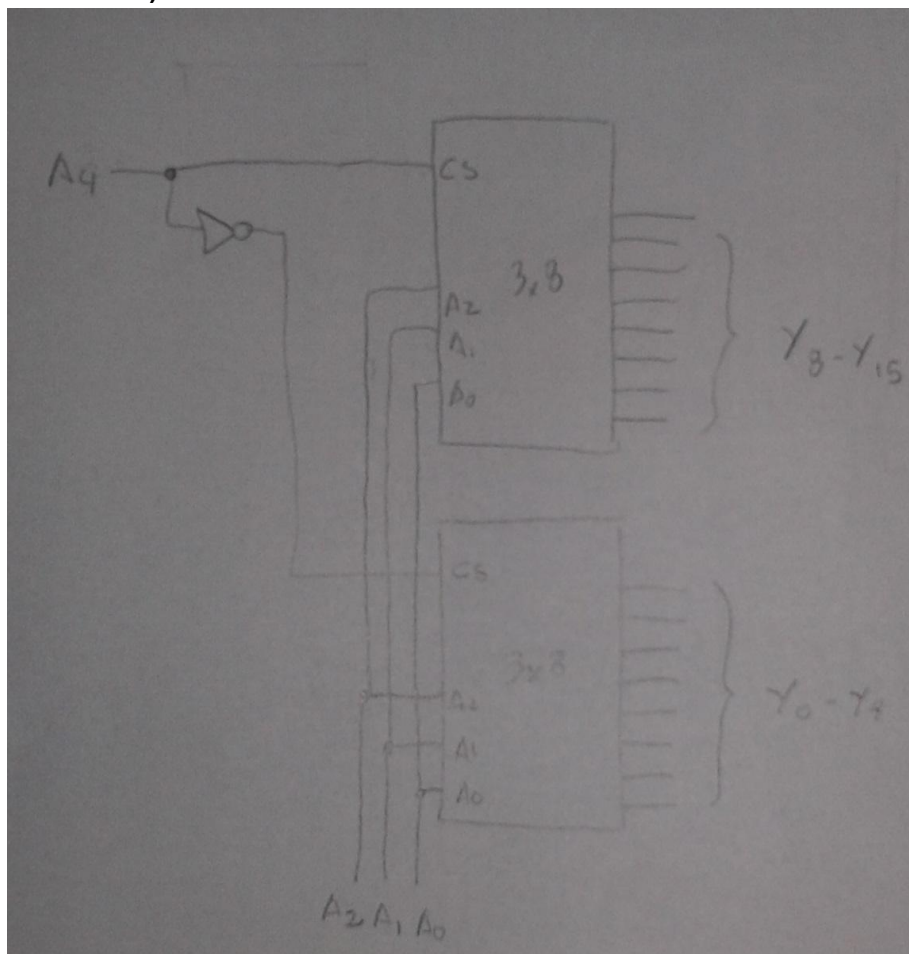
f)



Questão 2) Com Enable em 0 (ativado), as saídas Y_0 à Y_7 podem ser escolhidas normalmente. Com Enable em 1 (desativado), somente as saídas Y_8 e Y_9 poderiam ser selecionadas, mas nenhuma das saídas usadas é ativada.



Questão 3)



Questão 4)

a) Como vai até A15, estamos lidando com um microprocessador de $2^{16} = 64K$. O primeiro decodificador divide o espaço em $2^3 = 8$ partes de 8K cada uma usando os bits A15, A14 e A13. Nesse caso, somente as saídas Y_0 à Y_7 podem ser usadas, e somente nos casos em que a entrada D estiver em nível lógico 0.

A saída usada é a Y_4 , quando os bits de seleção (A15, A14, A13) estão em 100. Logo, a faixa de endereços é: (100)0 0000 0000 0000 à (100)1 1111 1111 1111. Ou seja: 8000_h à $9FFF_h$.

O segundo decodificador toma essa faixa de endereços e a divide novamente em 8 partes de 1K cada. Observa-se que, como a entrada D está em nível lógico alto, devem ser usadas as 8 saídas mais altas do decodificador 4 x 16 para acessar essas partes da memória (Y_8 à Y_{15}).

b) Na saída do primeiro decodificador, é uma organização de 8K palavras. No segundo, de 1K palavras.

c) Primeiro: (010)0 0000 0000 0000 à (010)1 1111 1111 1111, ou: 4000_h à $5FFF_h$.
Segundo: A saída Y_2 não acessa nenhum endereço.

d) Primeiro: Y_0 à Y_7
Segundo: Y_8 à Y_{15}

e) Para acessar algum endereço válido na memória, a entrada D do primeiro decodificador deve possuir nível lógico baixo, caso contrário, seria decodificada em Y_8 , Y_9 , ou nenhuma saída válida. Dessa forma, se nenhum sinal de RD ou WR for enviado (ou seja, ambos forem 1), a entrada D recebe 1 e a memória não é acessada.

Questão 5)

Endereços iniciais:

Memória de 16K: 0000_h

Memória de 8K: $0000_h + 16K = 4000_h$

1º Memória de 2K: $4000_h + 8K = 6000_h$

2º Memória de 2K: $6000_h + 2K = 6800_h$

Memória de 4K: $6800_h + 2K = 7000_h$

Espaço vazio: $7000_h + 4K = 8000_h$

Faixas:

Memória de 16K: 0000_h à $3FFF_h$

Memória de 8K: 4000_h à $5FFF_h$

1º Memória de 2K: 6000_h à $67FF_h$

2º Memória de 2K: 6800_h à $6FFF_h$

Memória de 4K: 7000_h à $7FFF_h$

Questão 6)

a) Semelhante à questão 5, somando 4K ao endereço inicial de cada bloco, temos:

Bloco 1: 0000 _h à 0FFF _h	Bloco 7: 6000 _h à 6FFF _h	Bloco 13: C000 _h à CFFF _h
Bloco 2: 1000 _h à 1FFF _h	Bloco 8: 7000 _h à 7FFF _h	Bloco 14: D000 _h à DFFF _h
Bloco 3: 2000 _h à 2FFF _h	Bloco 9: 8000 _h à 8FFF _h	Bloco 15: E000 _h à EFFF _h
Bloco 4: 3000 _h à 3FFF _h	Bloco 10: 9000 _h à 9FFF _h	Bloco 16: F000 _h à FFFF _h
Bloco 5: 4000 _h à 4FFF _h	Bloco 11: A000 _h à AFFF _h	
Bloco 6: 5000 _h à 5FFF _h	Bloco 12: B000 _h à BFFF _h	

Lógica de seleção usando decodificador 4 x 16. Com A₁₂ à A₁₅ nas entradas.

b) Divisão do Bloco 5:

Bloco 5.1: 4000 _h à 43FF _h	Bloco 5.3: 4800 _h à 4BFF _h
Bloco 5.2: 4400 _h à 47FF _h	Bloco 5.4: 4C00 _h à 4FFF _h

Divisão do Bloco 11:

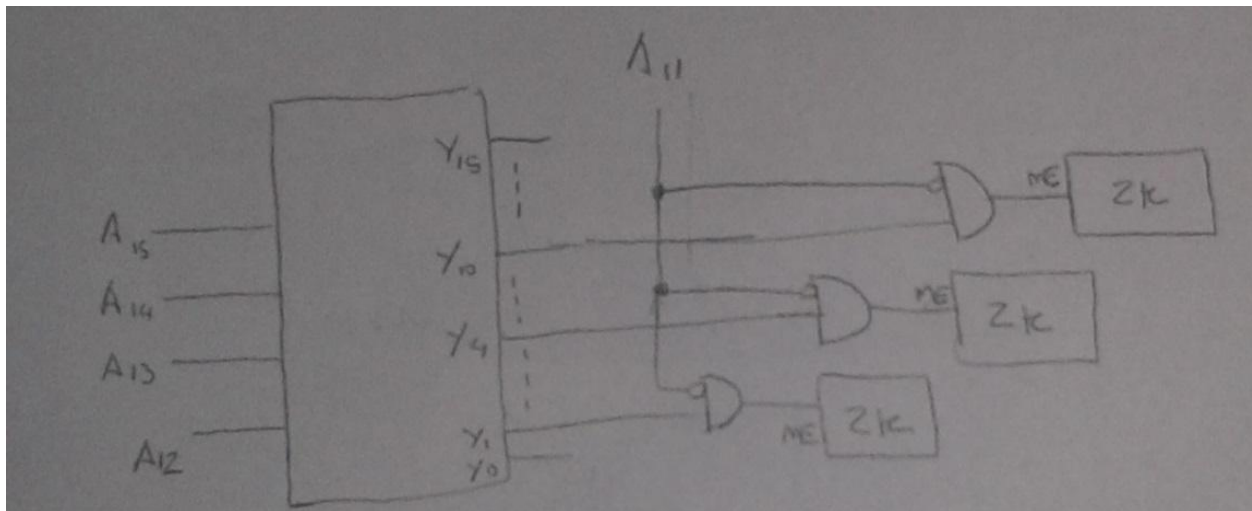
Bloco 11.1: A000 _h à A1FF _h	Bloco 11.5: A800 _h à A9FF _h
Bloco 11.2: A200 _h à A3FF _h	Bloco 11.6: AA00 _h à ABFF _h
Bloco 11.3: A400 _h à A5FF _h	Bloco 11.7: AC00 _h à ADFF _h
Bloco 11.4: A600 _h à A7FF _h	Bloco 11.8: AE00 _h à AFFF _h

c) Endereços:

Memória 1: 1000_h à 17FF_h

Memória 2: 4000_h à 47FF_h

Memória 3: A000_h à A7FF_h



d) Endereços fantasma:

Memória 1: 1800_h à 1FFF_h

Memória 2: 4800_h à 4FFF_h

Memória 3: A800_h à AFFF_h

