

# SEL0415

## Introdução à Organização de Computadores

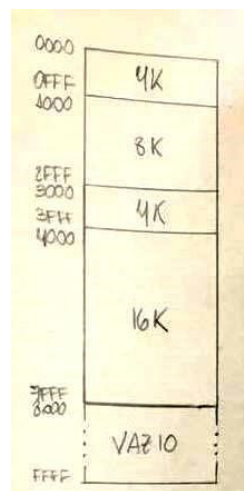
Prof. Dr. Marcelo Andrade da Costa Vieira

Lista 06 - Lógica de Seleção

Em todos as questões, considere um microprocessador de 16 bits e dutos de dados de 8 bits.

[01] Considere memórias de 4k, 8k, 4k e 16k, nessa ordem:

(a) Faça o mapeamento das faixas de endereço do  $\mu P$ .



(b) Determine os valores dos bits finais de endereçamento do  $\mu P$  ( $A_{15}$ ,  $A_{14}$ ,  $A_{13}$  e  $A_{12}$ ) para os endereços iniciais e finais das faixas correspondentes a cada memória.

|      | $A_{15} A_{14} A_{13} A_{12}$ | - | $A_{15} A_{14} A_{13} A_{12}$ |
|------|-------------------------------|---|-------------------------------|
| 4K:  | 0000                          | - | 0000                          |
| 8K:  | 0001                          | - | 0010                          |
| 4K:  | 0011                          | - | 0011                          |
| 16K: | 0100                          | - | 0111                          |

(c) Determine quantos bits de seleção serão usados para a lógica correspondente a cada memória.

Com relação à primeira memória de 4k, não podemos usar apenas um bit, pois  $A_{15} = 0$  também seleciona faixas das demais memórias. Da mesma forma, não podemos usar dois bits pois  $A_{15}A_{14} = 00$  também seleciona a de 8k e a outra de 4k e não podemos usar três bits pois  $A_{15}A_{14}A_{13} = 000$  também seleciona endereços da de 8k. Assim, devemos usar 4 bits de seleção para essa memória.

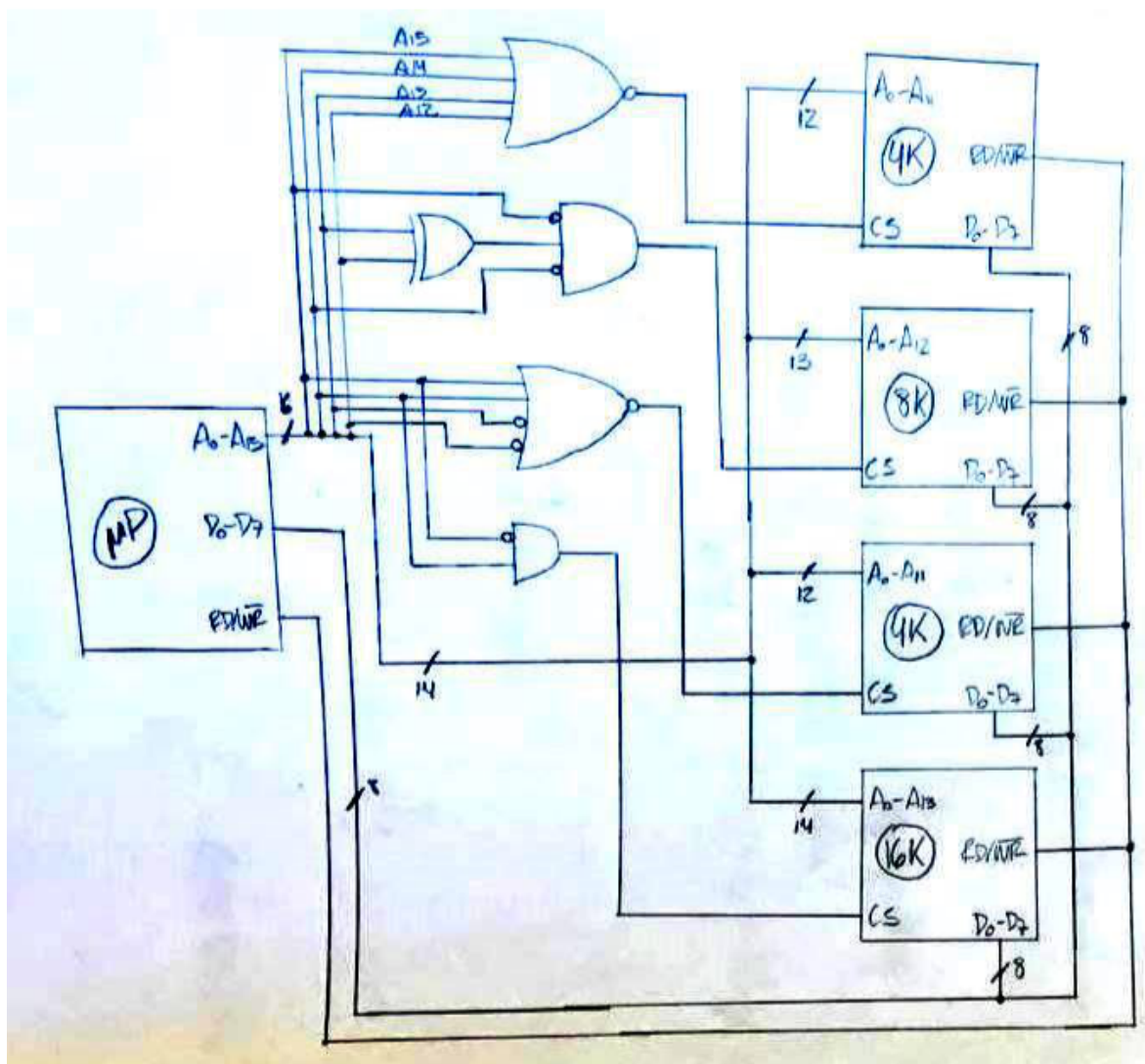
Da mesma forma, também devemos usar 4 bits de seleção para a outra de 4k e para a de 8k.

Entretanto, no caso da memória de 16k podemos utilizar apenas 2 bits, uma vez que  $A_{15}A_{14} = 01$  seleciona apenas essa memória.

(d) Considerando que as memórias são selecionadas quando o pino de habilitação se encontra em nível alto, determine as expressões lógicas de seleção de cada uma das memórias.

- Primeira memória de 4k: como devemos utilizar os 4 últimos bits de endereço para a seleção, podemos selecioná-la com o sinal  $S = \overline{A_{15}} + \overline{A_{14}} + A_{13} + A_{12}$ , de forma que essa memória será selecionada, ou seja, S será igual a 1, apenas quando  $A_{15}A_{14}A_{13}A_{12} = 0000$ .
- Memória de 8k: utilizando uma porta AND podemos selecioná-la com a expressão  $S = \overline{A_{15}} + \overline{A_{14}} \cdot f(A_{13}, A_{12})$ , de forma que  $f(A_{13}, A_{12})$  corresponde a uma lógica que resulte em 1 apenas quando  $A_{13}A_{12} = 01$  ou  $A_{13}A_{12} = 10$ . Para isso, podemos utilizar uma porta XOR, de modo que  $S = \overline{A_{15}} \cdot A_{14} \cdot \overline{A_{13}} \cdot \overline{A_{12}}$ .
- Segunda memória de 4k: de modo análogo ao da primeira memória, temos  $S = A_{15} + A_{14} + A_{13} + A_{12}$ .
- Memória de 16k: nesse caso utilizamos apenas os dois últimos bits de forma que a seleção ocorra apenas em  $A_{15}A_{14} = 01$ , o que pode ser feito com  $S = A_{15} \cdot A_{14}$ .

(e) Desenhe o esquema de ligações necessárias para essa lógica de seleção, considerando que todos os dispositivos possuem um pino de RD =  $\overline{WR}$ .



[02] Considere memórias de 8k, 32k, 4k e 16k, nessa ordem:

(a) Faça o mapeamento das faixas de endereço do  $\mu P$ .

|                      |       |
|----------------------|-------|
| 0000<br>1FFF<br>2000 | 8K    |
| 9FFF<br>A000         | 32K   |
| AFFF<br>B000         | 4K    |
| FFFF<br>F000         | 16K   |
| FFF                  | VAZIO |

(b) Determine o decodificador necessário para o projeto da lógica de seleção dessas memórias.

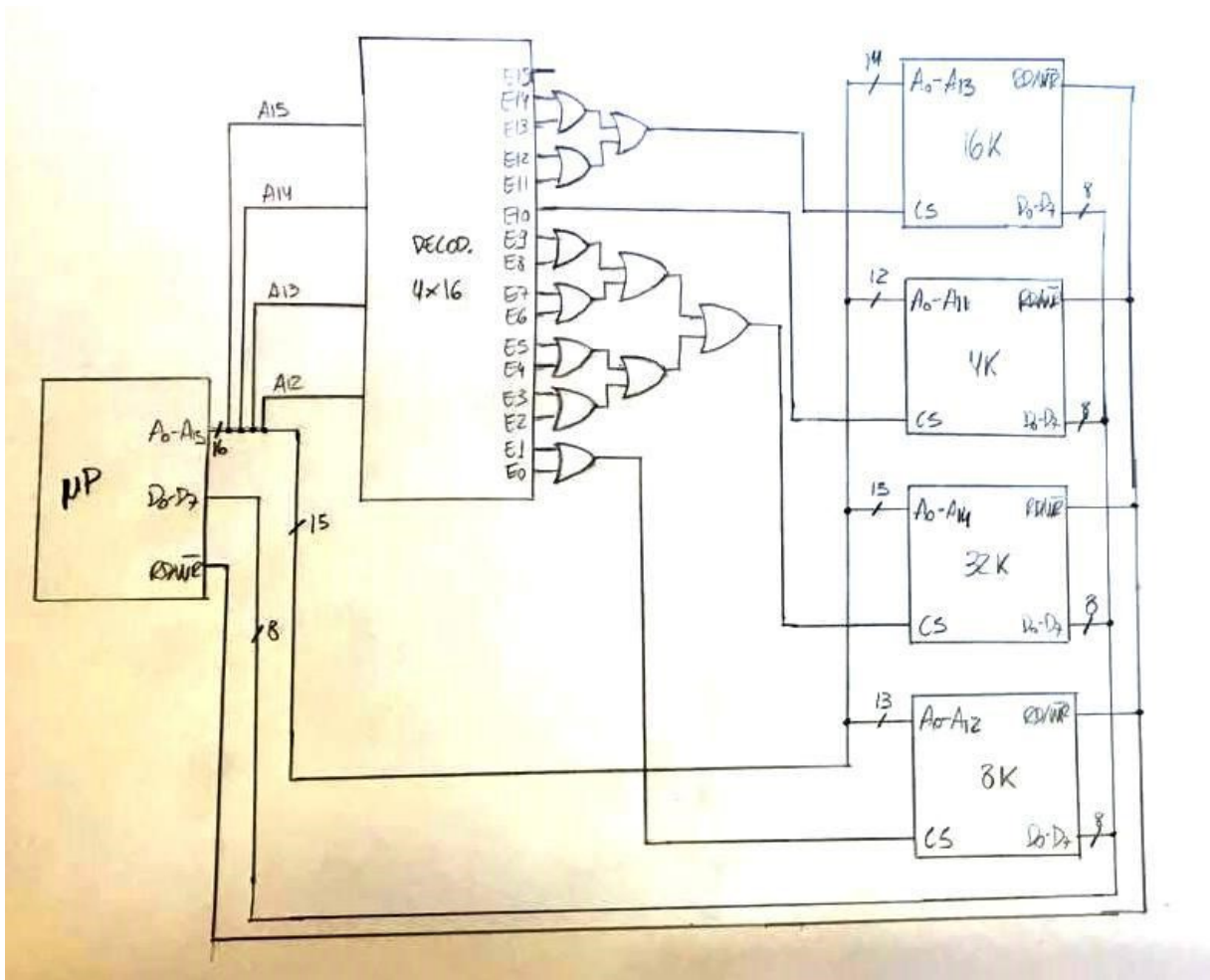
Como a menor memória é de 4k, precisamos de um decodificador que divida os endereços em faixas de 4k. Como temos um  $\mu P$  de 64k endereços, precisamos de um decodificador 4 x 16, que delimitará 16 faixas de 4k a partir dos 4 bits mais significativos.

(c) Considerando que as memórias são selecionadas quando o pino de habilitação se encontra em nível alto, determine as expressões lógicas de seleção de cada uma das memórias a partir das saídas do decodificador.

Sendo E0 E15 as saídas do decodificador, como os endereços são divididos em faixas de 4k, temos que:

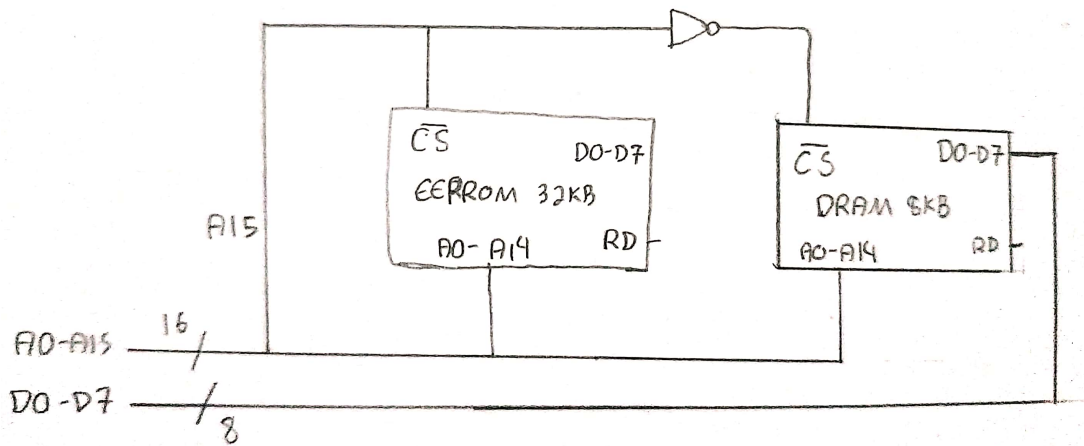
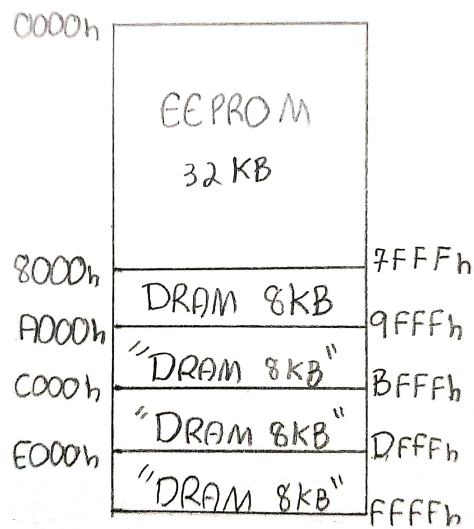
- Para a memória de 8k:  $S = E_0 + E_1$
- Para a memória de 32k:  $S = E_2 + E_3 + E_4 + E_5 + E_6 + E_7 + E_8 + E_9$
- Para a memória de 4k:  $S = E_{10}$
- Para a memória de 16k:  $S = E_{11} + E_{12} + E_{13} + E_{14}$

(d) Desenhe o esquema de ligações necessárias para essa lógica de seleção, considerando que o  $\mu P$  e as memórias possuem, cada um, um pino de  $RD = \overline{WR}$ .



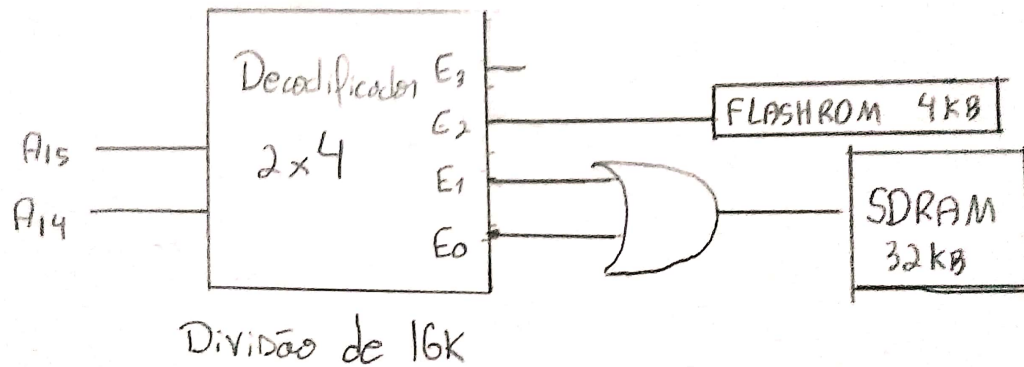
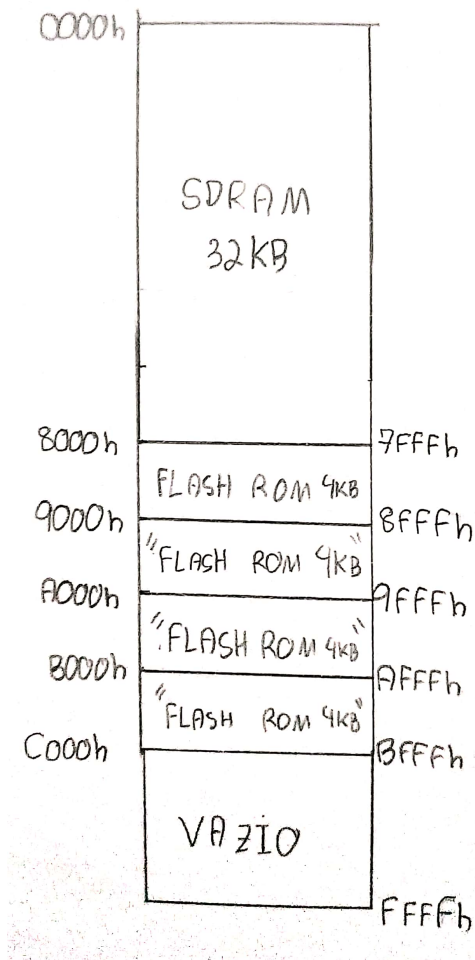
[03] Supondo que você possui uma EEPROM de 32kb e uma DRAM de 8kb, faça o mapeamento de memória para um decodificação não-absoluta utilizando o menor número de bits de seleção possível.

Para utilizar apenas um bit na seleção, consideraremos o bit  $A_{15}$  (MSB). Quando  $A_{15} = 0$ , selecionamos a memória EEPROM de 32KB, e quando  $A_{15} = 1$ , selecionamos a memória DRAM de 9KB. Como a lógica de seleção é não-absoluta, a memória de 8kb aparecerá "espelhada" mais 4 vezes (pois  $A_{14}$  e  $A_{13}$  não foram ligados nela e nem no decodificador, gerando endereços "fantasma"). O mapeamento final fica da seguinte forma:

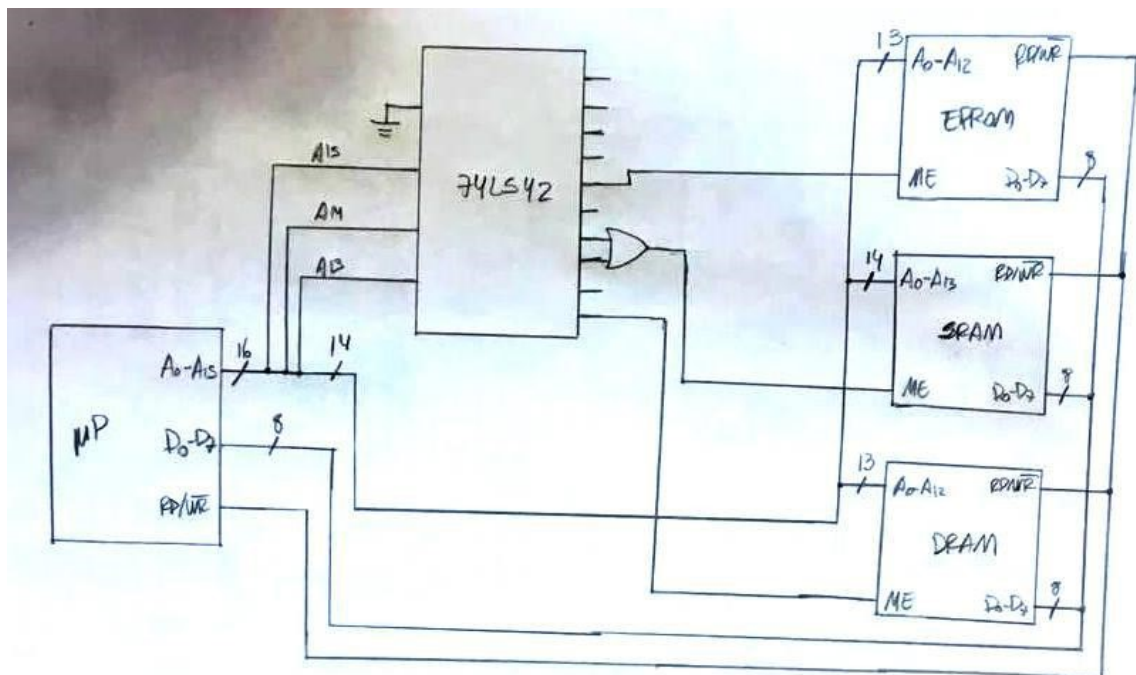
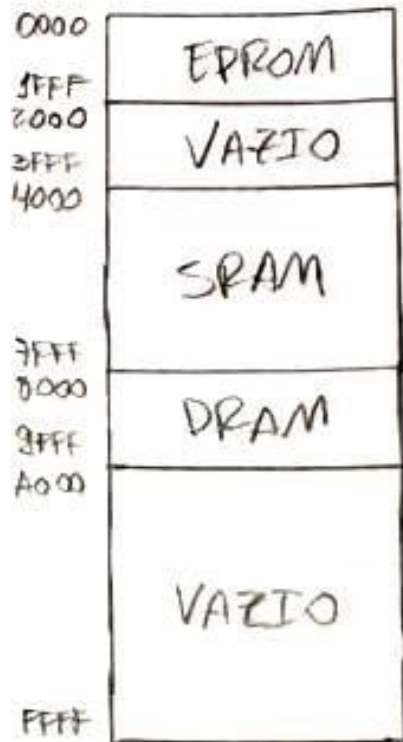


[04] Supondo que você possui uma SDRAM de 32kB, uma Flash ROM de 4kB e um decodificador 2x4, faça o mapeamento de memória utilizando decodificação não-absoluta.

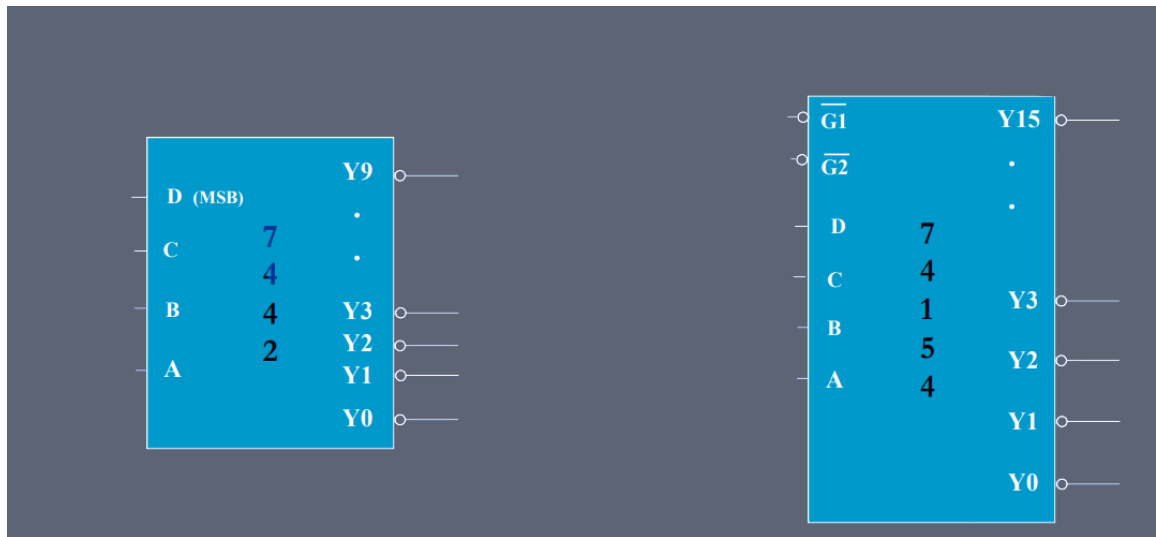
Considerando o decodificador com saídas  $E_0 - E_3$ , ligando a SDRAM a  $E_0 + E_1$  e a Flash ROM a  $E_2$ , temos o seguinte mapeamento:



[05] Supondo que você possui uma EPROM de 8kB, uma SRAM de 16kB, uma DRAM de 8kB e um decodificador 74LS42 (4 entradas, 10 saídas e lógica positiva), faça o projeto da lógica de seleção das memórias nessa ordem utilizando decodificação absoluta.

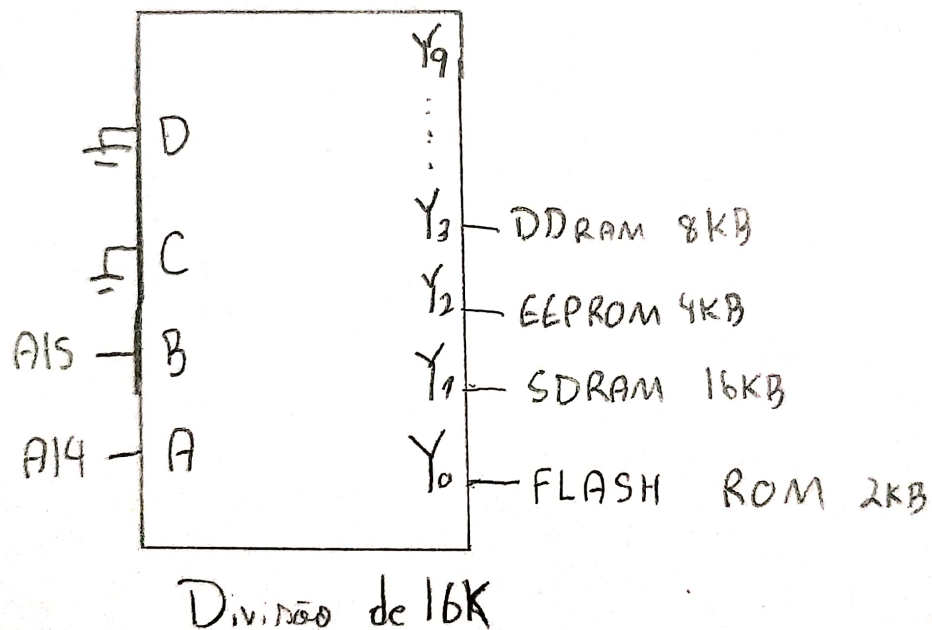


[06] Suponha que você tenha à sua disposição portas lógicas, e dois decodificadores (7442 e 74154). Considere uma Flash ROM de 2kB, uma SDRAM de 16kB, uma EEPROM de 4kB e uma DDRAM de 8kB, nessa ordem:



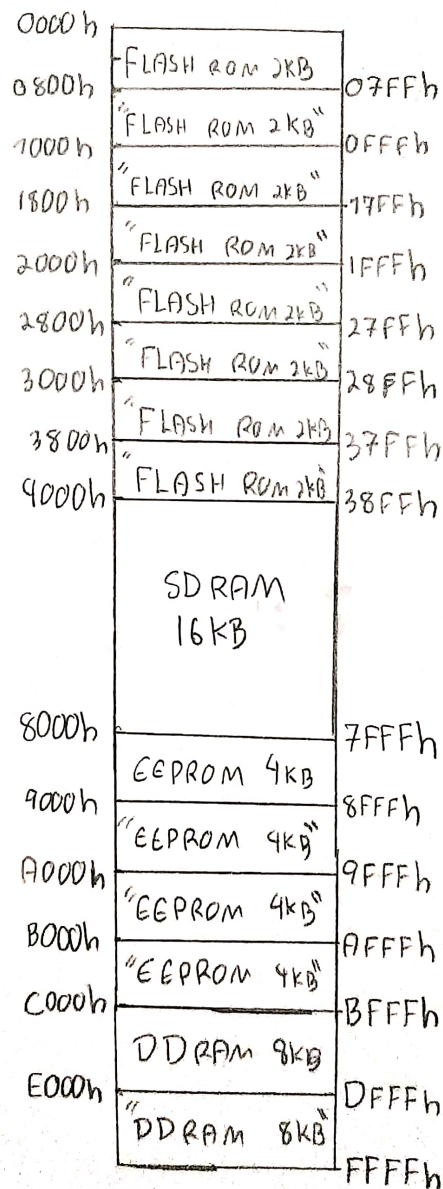
(a) Faça o mapeamento de memória utilizando decodificação não-absoluta, desenhando também o circuito utilizado para tal mapeamento.

O circuito pode ser montado de diferentes formas. No caso, foi utilizado primeiramente o decodificador 7442, que realiza a divisão em 16K:



Tal configuração resulta na criação de espaços de endereço fantasma, conforme pode ser visto no mapeamento abaixo:

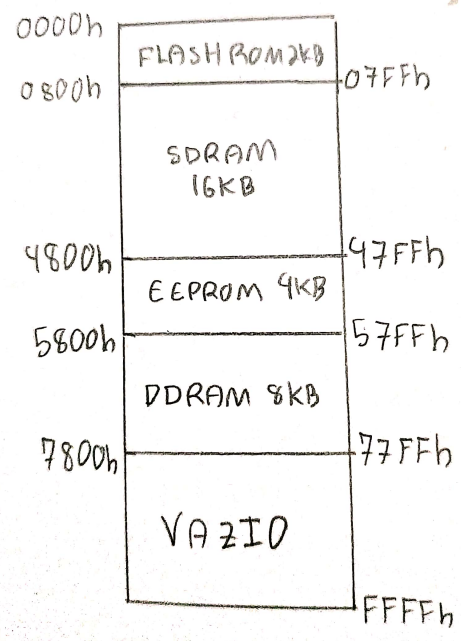
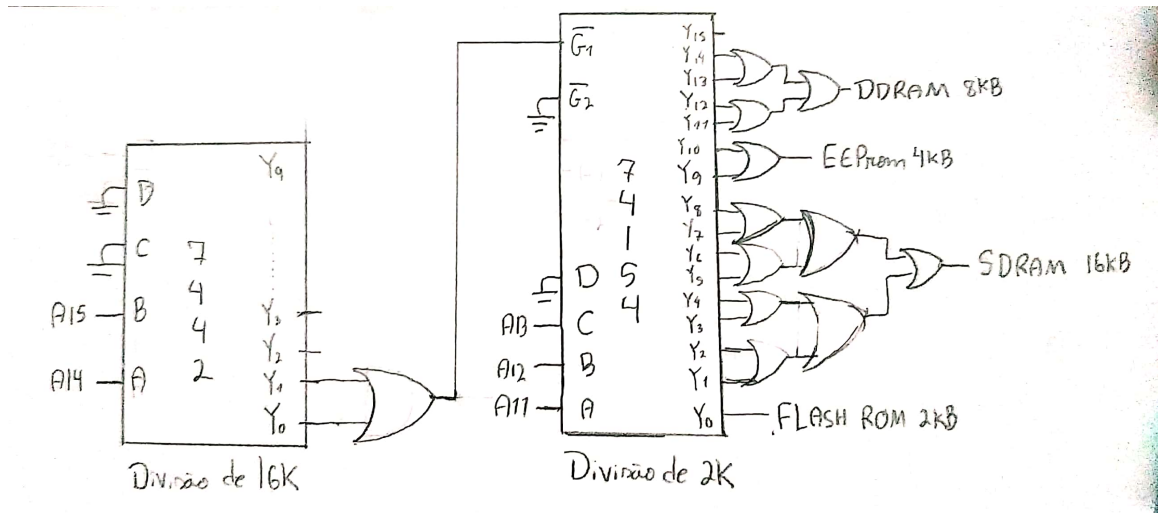




(b) Faça o mapeamento de memória utilizando decodificação absoluta, desenhando também o circuito utilizado para tal mapeamento.

Foi utilizado primeiramente o decodificador 7442 que realiza a divisão em 16K, e uma porta lógica OR para selecionar uma memória de 32k, maior do que a faixa de endereço das saídas de seleção. Nota-se que apenas os pinos Y0 a Y3 estão ativados, devido as entradas D e C estarem conectados ao terra.

A saída Y0 + Y1 é então conectada ao pino  $\overline{G1}$  do decodificador 74154, que é utilizado para a realização da divisão de faixas de endereço de 2K. Por fim, utilizando-se de portas OR, seleciona-se faixas maiores do que de 2K de acordo com a necessidade.



[07] Como ligar 2 decodificadores 3 x 8 (com o pino de Chip Select - CS) para formar um decodificador 4 x 16? Use portas lógicas se necessário.

