

# EA075

## Decodificação de Endereços



Faculdade de Engenharia Elétrica e de Computação (FEEC)  
Universidade Estadual de Campinas (UNICAMP)

Prof. Rafael Ferrari

# Introdução

- Geralmente, um sistema computacional possui mais do que um dispositivo de memória conectado ao processador.
- Ao acessar uma palavra no espaço de endereçamento, apenas um dos dispositivos deve ser ativado.
- Veremos quais são as estratégias para decodificação de endereços e acionamento dos módulos de memória do sistema.

# Decodificação Completa

- A decodificação completa usa todos os bits do barramento de endereços do sistema que não correspondem a *offset* (i.e., à parte do endereço que corresponde a uma palavra em um determinado dispositivo) para selecionar cada um dos dispositivos conectados ao barramento.
- Exemplo:
  - Sistema com barramento de endereços de 16 bits (A0, A1, ..., A15).
  - Três memórias conectadas ao barramento:
    - EEPROM (ROM) de 8K x 8 ( $2^{13} \times 8$ ): 13 bits para identificar uma palavra (*offset*)
    - RAM estática (RAM1) de 2K x 8: 11 bits de *offset*
    - RAM estática (RAM2) de 4K x 8: 12 bits de *offset*

# Decodificação Completa

- A ROM deve ocupar as primeiras 8192 palavras do espaço de endereçamento. A RAM1 e a RAM2 podem ocupar qualquer posição no espaço de endereçamento fora da faixa ocupada pela ROM.
- O primeiro passo é construir uma tabela de endereços em que as colunas correspondem aos sinais de endereço (A15 a A0) e as linhas representam os dispositivos a serem decodificados. Os bits correspondentes ao *offset* do dispositivo (menos significativos) são preenchidos com um X pois não são usados na sua seleção. Os bits restantes são preenchidos com um código binário que selecione unicamente o dispositivo. NO exemplo, a única restrição que devemos respeitar é que a memória ROM ocupe a faixa 0 a 8191.

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
ROM	0	0	0	X	X	X	X	X	X	X	X	X	X	X	X	X
RAM1	0	0	1	0	0	X	X	X	X	X	X	X	X	X	X	X
RAM2	0	0	1	1	X	X	X	X	X	X	X	X	X	X	X	X

# Decodificação Completa

- A partir da tabela, o sinal de seleção de cada uma das memórias é obtido a partir bits de endereço mais significativos (que não correspondem a *offset*). Considerando que cada memória é acionada a partir de um sinal CS' ativo baixo, temos:
  - $CS' \text{ ROM} = (A15' A14' A13)'$
  - $CS' \text{ RAM1} = (A15' A14' A13 A12' A11)'$
  - $CS' \text{ RAM2} = (A15' A14' A13 A12)'$
- O mapeamento das memórias no espaço de endereçamento usando essa decodificação é:

Dispositivo	Faixa de endereços
ROM	0 a 1FFF
RAM1	2000 a 27FF
RAM2	3000 a 3FFF

# Decodificação Completa

- **Vantagem:** É uma técnica de decodificação que permite a inclusão de outros dispositivos de memória no sistema em faixas não utilizadas a posteriori. Ou seja, é uma técnica que permite a expansão da memória do sistema.
- **Desvantagem:** O circuito de decodificação é relativamente complexo pois pode envolver um grande número de portas lógicas.

# Decodificação Parcial

- A decodificação parcial usa o menor número de bits possível que possibilite distinguir univocamente as faixas alocadas a cada um dos dispositivos.
- No exemplo anterior, podemos usar os dois bits mais significativos do barramento de endereços para realizar a decodificação. Dessa forma, uma faixa de 16384 endereços é alocada para cada dispositivo.

	A15	A14
ROM	0	0
RAM1	0	1
RAM2	1	0

Dispositivo	Faixa de endereços
ROM	0 a 3FFF
RAM1	4000 a 7FFF
RAM2	8000 a BFFF

# Decodificação Parcial

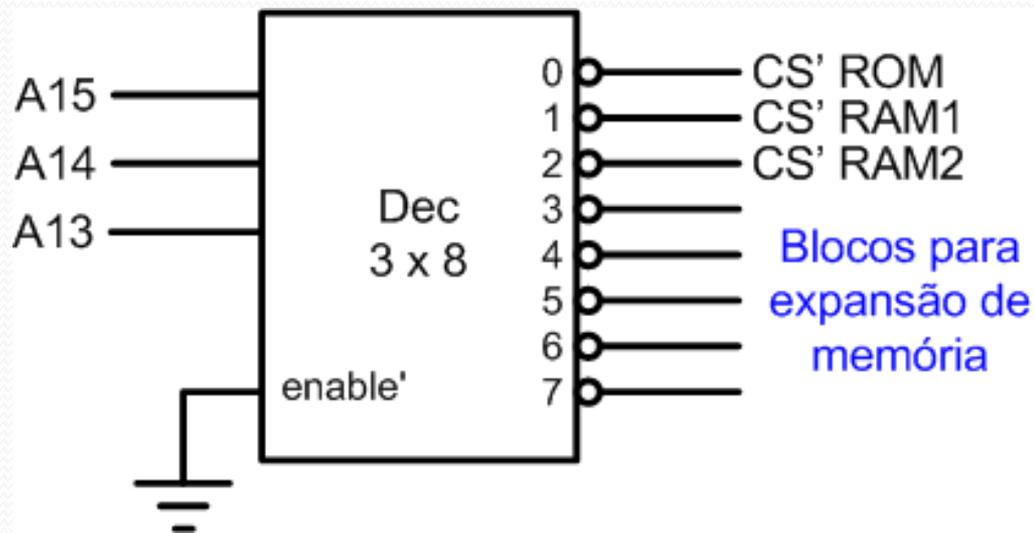
- Um decodificador 2 x 4 pode ser usado para gerar o sinal de ativação (CS') das memórias, usando A14 e A15 como entradas de seleção.
- Note que uma palavra armazenada em um dispositivo pode ser acessada por mais de um endereço distinto: a primeira palavra da memória ROM, por exemplo, é acessada pelo endereço 0 e também pelo endereço 0x2000
- **Vantagem:** circuito de decodificação mais simples
- **Desvantagem:** a faixa de endereços alocada para cada dispositivo é maior que a mínima necessária o que reduz as possibilidades de expansão e faz com que uma palavra possa ser acessada por múltiplos endereços.

# Decodificação por Blocos

- É um meio termo entre a decodificação parcial e a decodificação completa de endereços. Ela evita o uso ineficiente dos endereços que acontece na técnica parcial através da divisão do espaço de endereçamento em blocos de tamanhos iguais empregando a decodificação completa dos bits mais significativos de endereçamento.
- Cada bloco pode ser subdividido usando a mesma estratégia, empregando bits subsequentes de endereçamento.

# Decodificação por Blocos

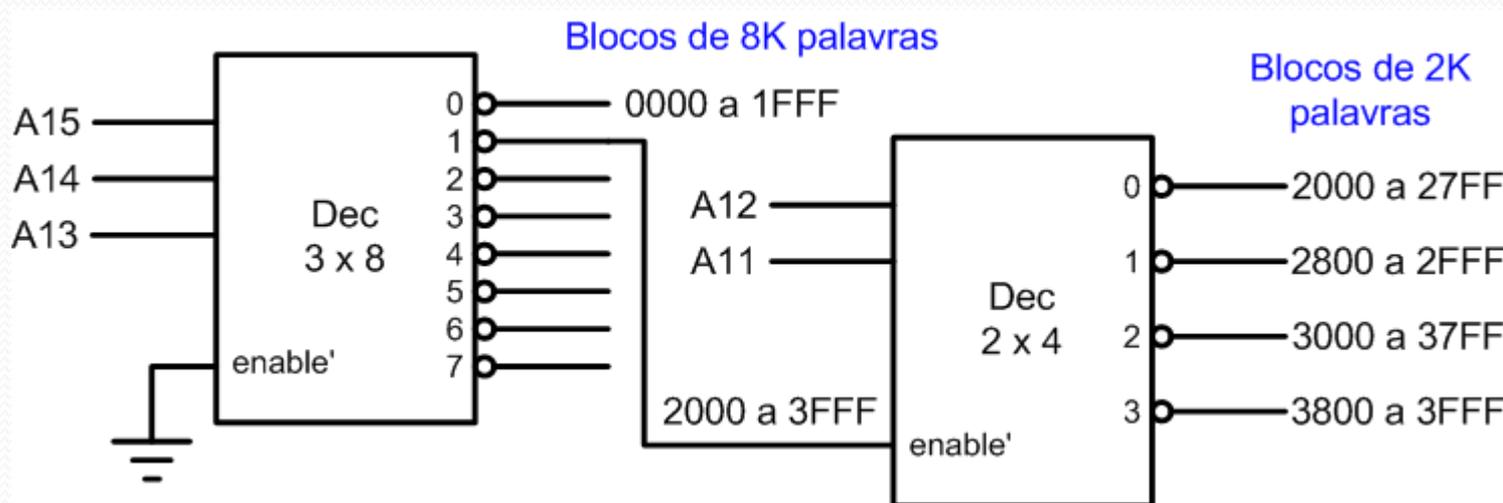
- No nosso exemplo, como a maior memória apresenta capacidade para armazenar 8K palavras, podemos empregar uma decodificação que divide o espaço de endereçamento em 8 blocos de 8K palavras usando os três bits mais significativos, A13, A14 e A15.



Dispositivo	Faixa de endereços
ROM	0 a 1FFF
RAM1	2000 a 3FFF
RAM2	4000 a 5FFF

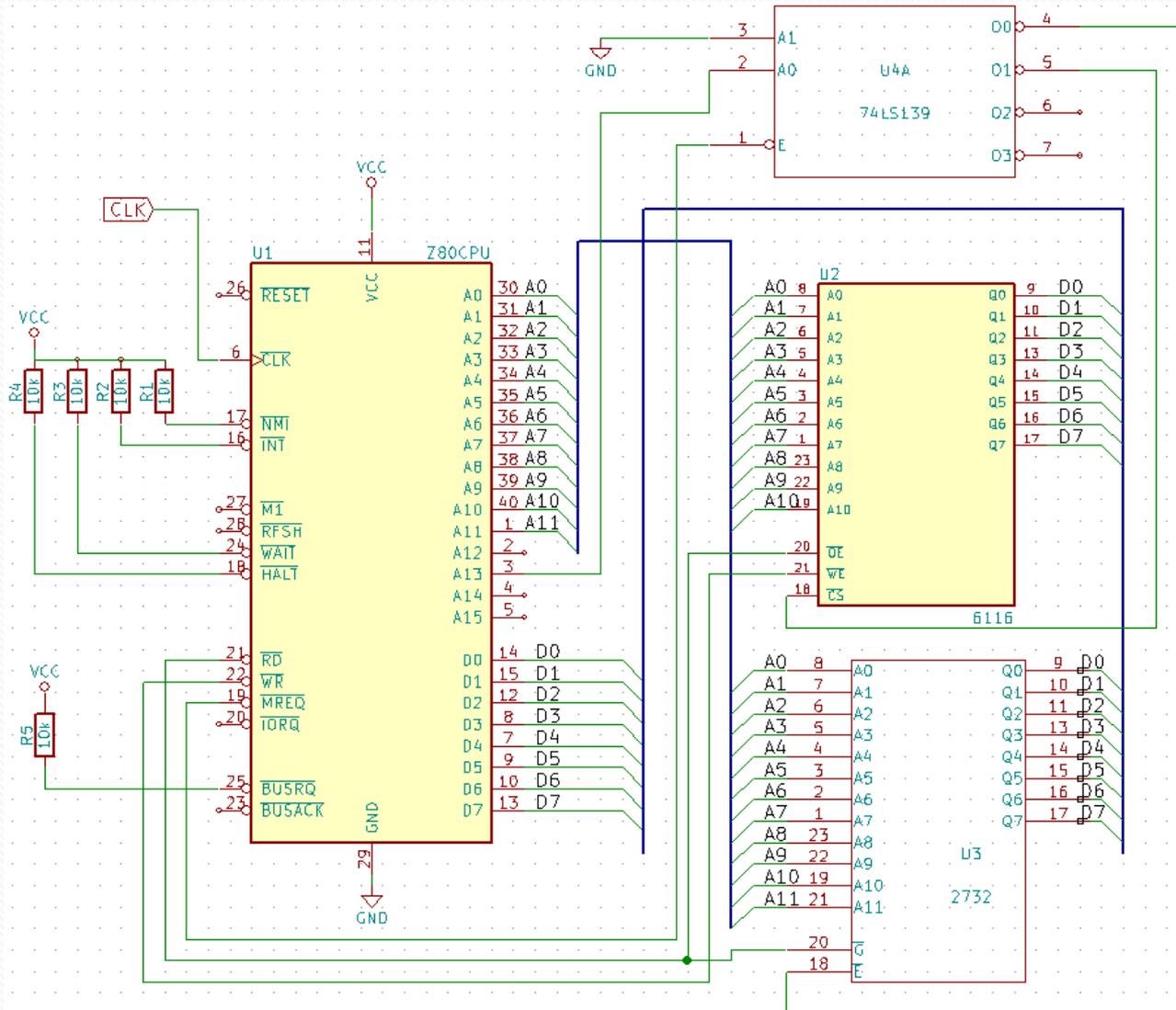
# Decodificação por Blocos

- A eficiência da decodificação pode ser ampliada dividindo-se os blocos em blocos menores.



- O projetista pode combinar as estratégias de decodificação a fim de obter o melhor resultado para seu sistema.

# Exemplo usando o Z80



## RAM estática 6116

Mode	$\overline{CS}$	$\overline{OE}$	$\overline{WE}$	I/O
Standby	H	X	X	High-Z
Read	L	L	H	DATA <sub>out</sub>
Read	L	H	H	High-Z
Write	L	X	L	DATA <sub>in</sub>

## EPROM 2732

Modo	$E'$	$G'$	I/O
Standby	1	X	High-Z
Leitura	0	0	Data Out

# Exemplo usando o Z80

- Qual a decodificação usada nesse sistema?
- Qual a faixa de endereços ocupada por cada uma das memórias?

# Exemplo usando o Z80

- Qual a decodificação usada nesse sistema?

Decodificação parcial

- Qual a faixa de endereços ocupada por cada uma das memórias?

	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
2732	X	X	0	X	X	X	X	X	X	X	X	X	X	X	X	X
6116	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X

- Se fosse usado A15 para discriminar as memórias ao invés de A13, teríamos:

Dispositivo	Faixa de endereços
2732	0 a 7FFF
6116	8000 a FFFF

# Bibliografia

- Clements, Alan. *Microprocessor Systems Design: 68000 Hardware, Software, and Interfacing*. 3<sup>a</sup> ed., PWS Publishing Company, 1997. (Capítulo 5, seção 5.1)