

# Sistemas Lógicos

(parte 2)

Dep.Armas e Electrónica- Escola Naval  
V.2.3 V.Lobo 2003

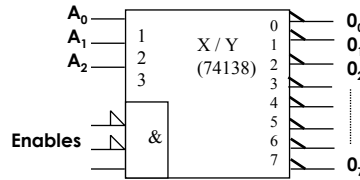
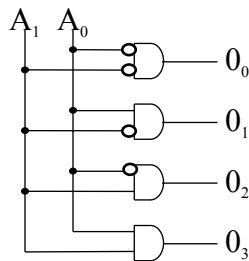
## Descodificadores

Lógicos (2)

- Objectivo: Explicitar um valor codificado
- Exemplo: A partir de um valor codificado em 2 bits, actuar 1, e 1 só linha de saída

- Implementação:
  - Em lógica discreta
  - Com integrados dedicados

Entradas		Saídas			
A <sub>1</sub>	A <sub>0</sub>	O <sub>0</sub>	O <sub>1</sub>	O <sub>2</sub>	O <sub>3</sub>
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1



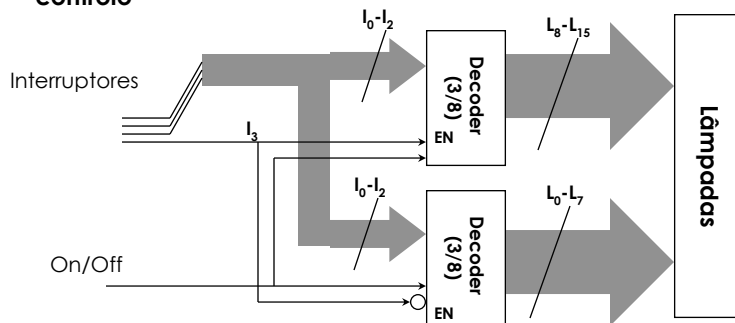
1

## Descodificadores

Lógicos (2)

- Linhas de ENABLE
  - Quando estão desligadas, o circuito tem ZERO nas saídas, ou está em TRI-STATE
  - Podemos usá-las para associar descodificadores

- Exercício
  - Pretende-se acender uma de 16 lâmpadas, com 4 linhas de controlo



2

# Sistemas Lógicos

(parte 2)

Dep.Armas e Electrónica- Escola Naval  
V.2.3 V.Lobo 2003

## Codificadores

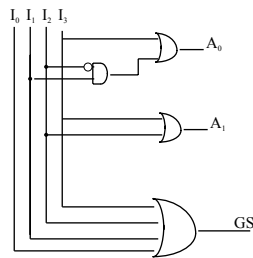
Lógicos (2)

- **Objectivo: Inverso ao do descodificador**

- A partir de várias variáveis (das quais só 1 está activa) gerar um sinal codificado
- Exemplo: a partir de 4 sinais, vindos de sensores, gerar um código de 2 bits indicando qual deles está activo

- **Implementação:**

- Com portas lógicas
- Com integrados dedicados



Entradas				Saídas		
I3	I2	I1	I0	A1	A0	GS
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	x	0	1	1
0	1	x	x	1	0	1
1	x	x	x	1	1	1

3

## Codificadores

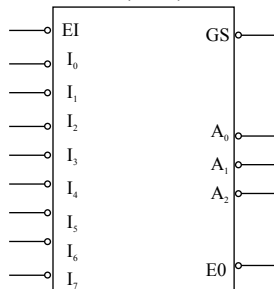
Lógicos (2)

- **Group Signal (GS)**

- Indica que pelo menos uma entrada está activa
- Serve para distinguir a situação em que a entrada 0 está activa (e portanto o código é 0) e a situação em que não nenhuma entrada activa (ficando a saída com o código 0, mas o sinal GS=0)

- **Enable**

- Quando não está activa, as saídas estão todas a 0



Nota: No 74148 todas as entradas e saídas são activas a 0

4

# Sistemas Lógicos

(parte 2)

Dep.Armas e Electrónica- Escola Naval  
V.2.3 V.Lobo 2003

## Conversores de código

Lógicos (2)

- **Descodificadores de 7 segmentos**

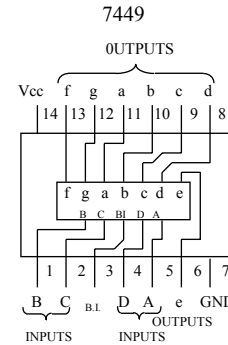
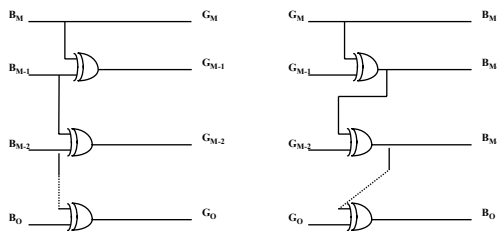
- Convertem 4 bits (contendo um número em binário natural ou BCD) nos sinais necessários para actuar um display de 7 segmentos

- **Codificadores/Descod.BCD**

- Exemplos: 74184,74185

- **Codificadores/Descod.Gray**

- Feitos à custa de XOR



5

## Aplicações

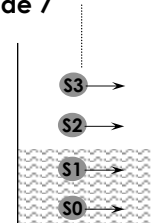
Lógicos (2)

- **Problema do nível de combustível**

- Pretende-se saber o nível de combustível num tanque que não é facilmente acessível
- Dispomos de 16 sensores, que indicam se há ou não combustível nesse ponto. NOTA: Para que a visualização seja mais fácil, pode depois voltar a resolver o problema para com apenas 10 sensores.
- Dispomos de um display de 7 segmentos para visualizar o nível dos tanques.
- Dispomos de codificadores/descodificadores de 16 para 4 bits, e vice-versa.
- Pretende-se ter a indicação do nível de combustível na ponte  
→Faça o logigrama completo incluindo os sensores, o sistema de codificação, a transmissão, dos descodificadores de 7 segmentos, e os displays

- **Resolva o mesmo problema usando apenas os codificadores/descodificadores dados nas aulas**

- (codificadores de 8 para 3 bits e vice-versa)



6

# Sistemas Lógicos

(parte 2)

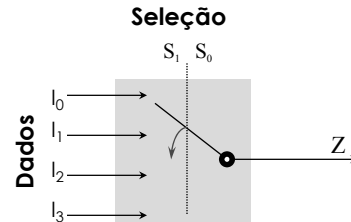
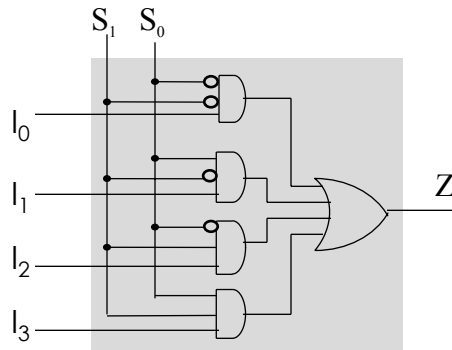
Dep.Armas e Electrónica- Escola Naval  
V.2.3 V.Lobo 2003

## Multiplexers (Mux)

Lógicos (2)

- **Objectivo: Seleccionar uma de várias entradas**
  - Também chamados DATA-SELECTORS ou simplesmente MUX

- **Implementação:**
  - Com portas lógicas
  - Com integrados dedicados



Entradas		Saída
S <sub>1</sub>	S <sub>0</sub>	Z
0	0	I <sub>0</sub>
0	1	I <sub>1</sub>
1	0	I <sub>2</sub>
1	1	I <sub>3</sub>

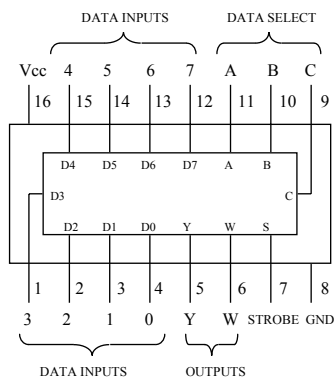
7

## Multiplexers (Mux)

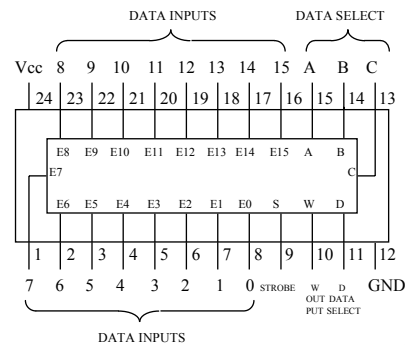
Lógicos (2)

- **Multiplexers da família 74xx**
  - Dispõe de um sinal de ENABLE, também chamado STROBE
    - O sinal STROBE é normalmente gerado pelo circuito que pretende ler a informação
  - 74151 - 8 entradas, 3 bits de selecção, e strobe
  - 74150 - 16 entradas, 4 bits de selecção, e strobe

74151



74150



8

# Sistemas Lógicos

(parte 2)

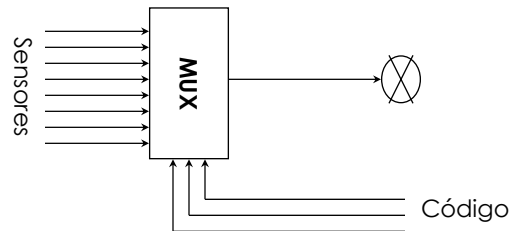
Dep.Armas e Electrónica- Escola Naval  
V.2.3 V.Lobo 2003

## MUX - Aplicações

Lógicos (2)

### • Multiplexagem para fazer "polling"

- Posso "interrogar" um conjunto de sensores, enviando o código do sensor que quero observar.



### • Multiplexagem de "buses"

- Podemos partilhar um conjunto de linhas, passando ora uns dados ora outros pelos mesmos "fios"
- Podemos usar um Clock e um contador para mandar sequencialmente todos os dados

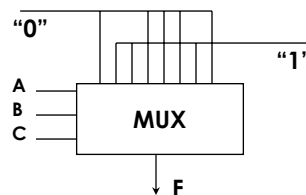
9

## MUX - Aplicações

Lógicos (2)

### • Geração de funções booleanas

- Um MUX é um OR de ANDs e cada AND corresponde a um MINTERMO das entradas de selecção com mais uma das entradas de dados
- Para seleccionar um mintermo basta pôr a entrada de dados desse mintermo a 1.
- Vantagens: com 1 só integrado e quase nenhum trabalho de projecto, implemento uma função.
- Desvantagens: estou a usar mais gates do que as necessárias
- Exemplo:
  - Implementar a função  $F(A,B,C)=\Sigma(1,2,6)$



10

# Sistemas Lógicos

(parte 2)

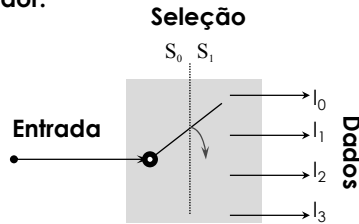
Dep. Armas e Electrónica - Escola Naval  
V.2.3 V.Lobo 2003

## DESMULTIPLEXAGEM

Lógicos (2)

- É a operação inversa da multiplexagem

- As saídas que não estão ligadas à entrada estão ligadas ao estado "inactivo" (normalmente estão ligadas a "0")
- Posso implementar um DEMUX usando um DESCODIFICADOR se ligar as linhas de selecção às entradas de dados do decodificador, e a entrada de dados ao ENABLE do decodificador.



- Aplicações

- Projecte um sistema para transmitir os dados de 16 sensores através de apenas 6 fios.

11

## CIRCUITOS DE ADIÇÃO

Lógicos (2)

- Circuitos COMBINATÓRIOS de adição/subtracção

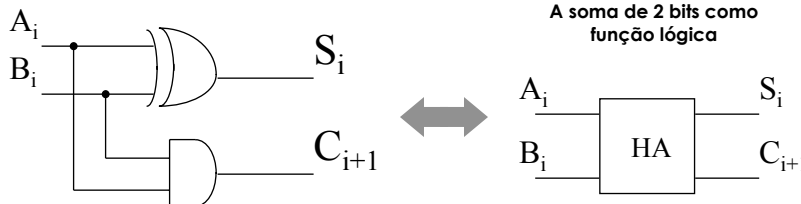
- As somas/subtracções podem também ser feitas com circuitos sequenciais

- Half adder

- soma dois bits (s/carry)
- Só pode somar números com 1 bit

A	B	Soma	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

A soma de 2 bits como função lógica



12

# Sistemas Lógicos

(parte 2)

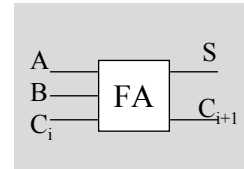
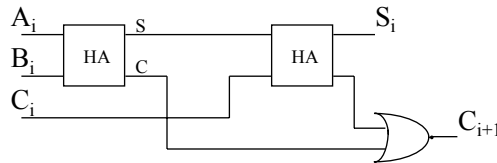
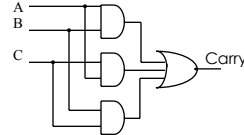
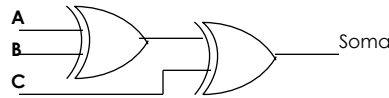
Dep.Armas e Electrónica- Escola Naval  
V.2.3 V.Lobo 2003

## CIRCUITOS DE ADIÇÃO

Lógicos (2)

### • Full adder

- soma dois bits  $c/carry \Rightarrow$  soma três bits
- Pode ser feito directamente como uma função de 3 variáveis
- Pode ser feito com 2 semi-somadores



13

## Somadores de vários bits

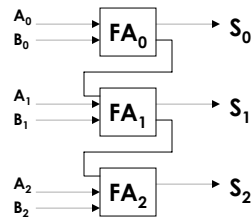
Lógicos (2)

### • Cascatas de full adders

- Problema com tempos de atraso

### • Somador $c/ 2$ andares

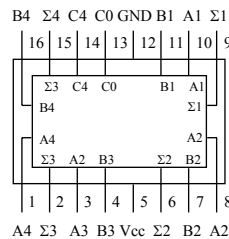
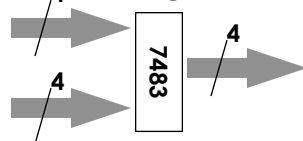
- Implementam a soma como  $n$  funções de  $2n+1$  variáveis
- Muito hardware



### • Solução mista

- Uso do carry generate (G) e carry propagate (P) (look ahead)

### • Circuitos integrado 7483



14

# Sistemas Lógicos

(parte 2)

Dep.Armas e Electrónica- Escola Naval  
V.2.3 V.Lobo 2003

## Circuitos subtratores

Lógicos (2)

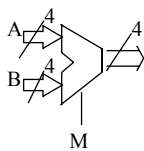
- **Dedicados**
  - Implementam a subtracção como mais uma função Booleana
- **Realizados com somadores**
  - Subtrair é somar o complemento para dois
  - Calcular o complemento para 2 pode ser obtido negando os bits todos e depois somando 1
- **Somadores BCD**
  - Corrigem o resultado para BCD
  - Há “carry” sempre que o resultado ultrapassa 9, sendo necessário reajustar o valor da soma
  - Exercício:
    - Faça um somador BCD usando 2 somadores vulgares, um MUX, e alguma lógica de controlo. Deverá somar 6 sempre há um carry, ou quando o resultado está entre 9 e 15.

15

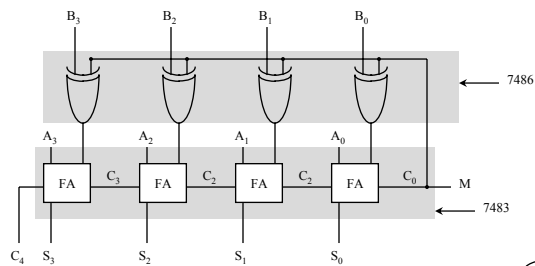
## ALU (UNIDADE DE ARITMÉTICA E LÓGICA)

Lógicos (2)

- **Objectivo**
  - Realizar com um integrado várias operações aritméticas ou lógicas
- **Operação**
  - Tem normalmente 2 entradas de  $n$  bits, que são os dois operandos
  - Tem uma (ou mais) entradas de controlo, que ordenam à ALU que faça uma das suas operações
  - Exemplo:
    - Uma ALU que faz somas e subtracções de 4 bits



M=0 Faz soma  
M=1 Faz subtracção



16



# Sistemas Lógicos

(parte 2)

Dep.Armas e Electrónica- Escola Naval  
V.2.3 V.Lobo 2003

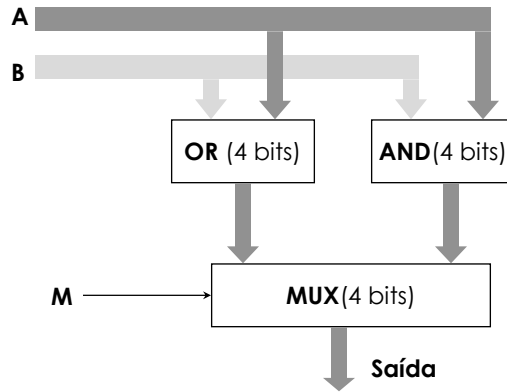
## ALU (UNIDADE DE ARITMÉTICA E LÓGICA)

Lógicos (2)

- **Problema:**

- Fazer uma ALU para calcular  $A \text{ AND } B$  ou  $A \text{ OR } B$  conforme o sinal  $M$  seja 0 ou 1, sendo  $A$  e  $B$  números de 4 bits

- **Solução:**



Problema:  
Faça uma ALU que seja capaz de fazer somas, subtrações, AND, e OR.

17