|  |  |
| --- | --- |
| essg | Escola Secundária de Sebastião da Gama |
| Sistemas Digitais e Arquitetura de Computadores – Módulo 3 |
| Ano 10º - Turma N |
| Prof. Carlos Pereira |

*Resumo*

***Circuitos lógicos combinatórios***

[**www.ticmania.net**](http://www.ticmania.net)

## Comparadores

Comparadores - circuitos que detetam se dois números de n bits são iguais e, sendo diferentes, qual é o maior.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Entradas** | | **Saída 1** | **Saída 2** | **Saída 3** | **Saída 4** |
| **Posição** | **A** | **B** | **A<B** | **A=B** | **A>B** | **A<>B** |
| 0 | 0 | 0 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 | 0 | 1 |
| 2 | 1 | 0 | 0 | 0 | 1 | 1 |
| 3 | 1 | 1 | 0 | 1 | 0 | 0 |
| Expressão |  |  |  | +AB  XNOR |  | B+A  AB  XOR |

**SAÍDA 2 – COMPARADOR XNOR:**

A porta XNOR (ou coincidência) pode ser considerada como um comparador elementar de 1 bit. Na realidade, na sua saída teremos o valor lógico 1 se as duas entradas forem iguais e o valor lógico 0 se as entradas forem diferentes.



**SAÍDA 4 – COMPARADOR XOR:**

A porta XOR (ou exclusivo) pode ser considerada como um comparador elementar de 1 bit. Na realidade, na sua saída teremos o valor lógico 0 se as duas entradas forem iguais e o valor lógico 1 se as entradas forem diferentes.

**Cria um comparador que verifique se 2 bits são iguais**

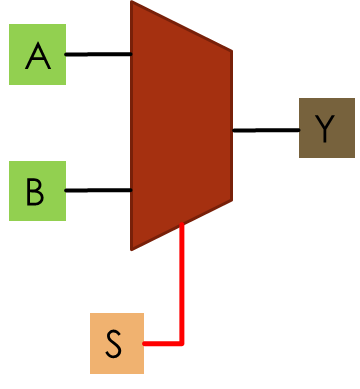
## Multiplexers

## 

Um multiplexador, *multiplexer*, *mux* ou multiplex é um dispositivo que seleciona as informações de duas ou mais fontes de dados num único canal. Normalmente atua em conjunto com um desmultiplexador.

Exemplo:

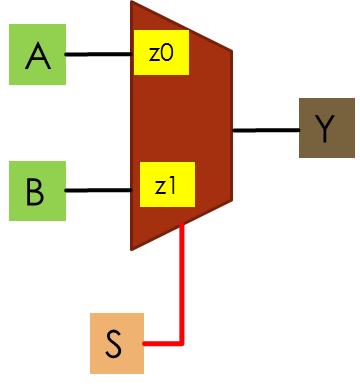
**Multiplexador (2/1) de 2 entradas (A e B) e uma saída Y**



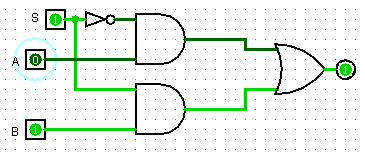
Y=SB+S’A

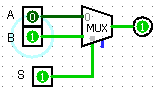
|  |  |  |  |
| --- | --- | --- | --- |
| **Seletor** | **Entradas** | | **Saída** |
| S | B | A | Y |
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

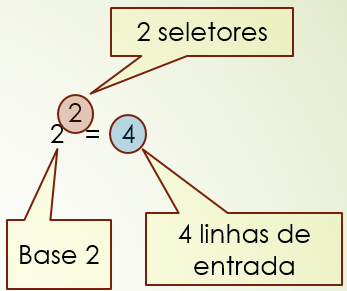
A tabela de verdade pode ser **simplificada** colocando o enfoque em z1 e z2.



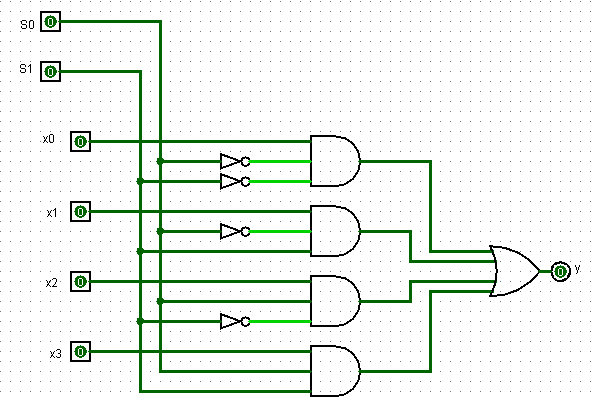
|  |  |
| --- | --- |
| **Seletor** | **Saída** |
| S | Y |
| 0 | Z0 |
| 1 | z1 |





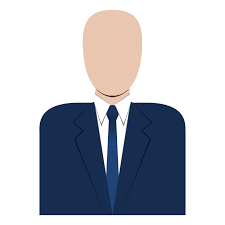
Exercício Mux 4:1

|  |  |  |
| --- | --- | --- |
| **Seletores** | | **Saída** |
| S0 | S1 | f |
| 0 | 0 | Z0 |
| 0 | 1 | Z1 |
| 1 | 0 | Z2 |
| 1 | 1 | z3 |

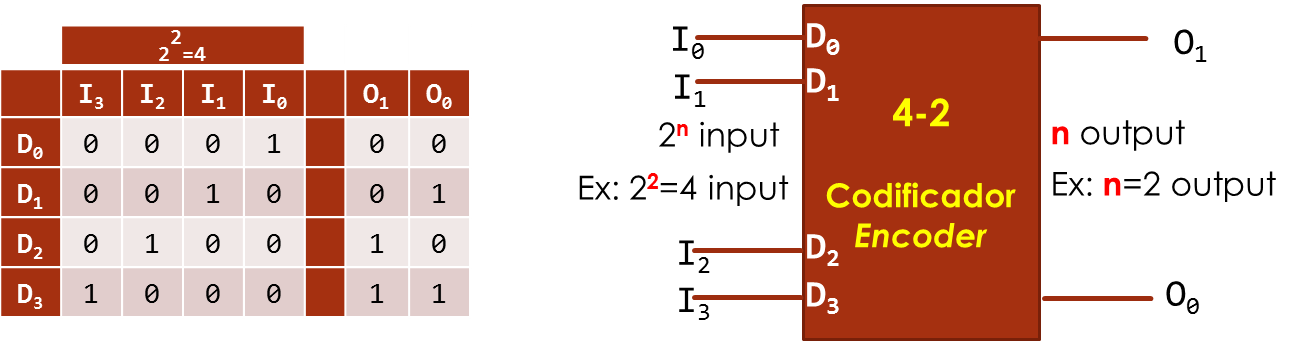


**Cria o circuito lógico**

## Codificadores



**Codificar** - designa o processo de representar caracteres ou grupos de caracteres em código binário.

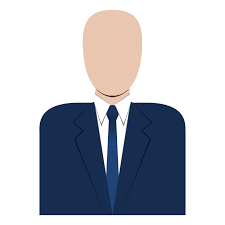
**CODIFICADORES** – É um circuito capaz de converter um sinal de determinado tipo, como por exemplo **decimal**, num sinal **binário ou BCD**.

**Codificador 4:2 (4 entradas e 2 saídas)**

|  |  |
| --- | --- |
| **4 entradas e 2 saídas, ou seja, 4:2**  **Porque base 2 elevado a 2 output, ou seja, 2^2=4** | |
| |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **e3** | **e2** | **e1** | **e0** | **s1** | **s0** | **Posição** | | 0 | 0 | 0 | 1 | 0 | 0 | 0 | | 0 | 0 | 1 | 0 | 0 | 1 | 1 | | 0 | 1 | 0 | 0 | 1 | 0 | 2 | | 1 | 0 | 0 | 0 | 1 | 1 | 3 | | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | **e3** | **e2** | **e1** | **e0** | **s1** | **s0** | **Posição** | | 0 | 0 | 0 | 1 | 0 | 0 | 0 | | 0 | 0 | 1 | 0 | 0 | E1 | 1 | | 0 | 1 | 0 | 0 | E2 | 0 | 2 | | 1 | 0 | 0 | 0 | E3 | E3 | 3 | | S0=E2+E3  S1=E1+E3 | | | | | | | |
|  | |

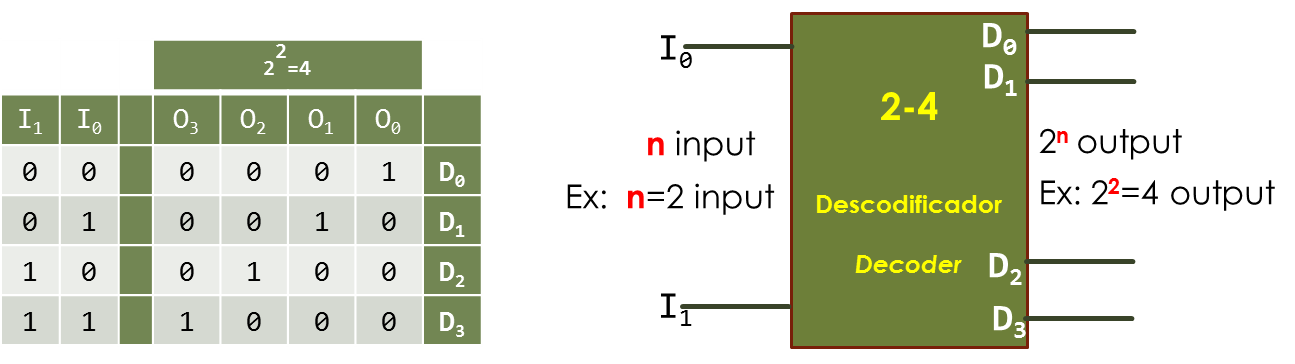
|  |  |
| --- | --- |
| **8 entradas e 3 saídas, ou seja, 8:3**  **Porque base 2 elevado a 3 output, ou seja, 2^3=8** | |
| |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **E7** | **E6** | **E5** | **E4** | **e3** | **e2** | **e1** | **e0** | S2 | **s1** | **s0** | **Posição** | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 1 | | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 2 | | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 1 | 3 | | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 4 | | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 5 | | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 6 | | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 7 | | |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | | **E7** | **E6** | **E5** | **E4** | **e3** | **e2** | **e1** | **e0** | S2 | **s1** | **s0** | **Posição** | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | E1 | 1 | | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | E2 | 0 | 2 | | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | E3 | E3 | 3 | | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | E4 | 0 | 0 | 4 | | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | E5 | 0 | E5 | 5 | | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | E6 | E6 | 0 | 6 | | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | E7 | E7 | E7 | 7 | | S0=e1+e3+e5+e7  S1=e2+e3+e6+e7  S2=e4+e5+e6+e7 | | | | | | | | | | | | |
|  | |

**Descodificadores**



**Descodificar** - converte a informação de binário para outra forma de representação (ex. binário para decimal).

**DESCODIFICADOR** é, genericamente, um circuito com múltiplas entradas e múltiplas saídas, que converte entradas codificadas em saídas igualmente codificadas, em que o código usado na entrada é diferente do usado na saída. (ex: binário para decimal).



**Descodificador 2:4 (2 entradas e 4 saídas)**

|  |  |
| --- | --- |
| **Exercício:**  **O deposito de uma mota tem capacidade para 7 litros de combustível. Desenvolve um descodificador que indique sempre que o deposito atinga os 1, 3 e 5 litros de combustível.** | |
| **3 entradas e 8 saídas, ou seja, 3:8**  **Porque base 2 elevado a 3 input, ou seja, 2^3=8** | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **A** | **B** | **C** | **s0** | **Posição** | | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | 1 | 1 | 1 | | 0 | 1 | 0 | 0 | 2 | | 0 | 1 | 1 | 1 | 3 | | 1 | 0 | 0 | 0 | 4 | | 1 | 0 | 1 | 1 | 5 | | 1 | 1 | 0 | 0 | 6 | | 1 | 1 | 1 | 0 | 7 |   **S0=c.b’.a’+c.b.a’+c.b’.a** | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | BA |  | |  | |  | |  | | | C |  | 00 | | 01 | | 11 | | 10 | | |  | 0 | 0 | 0 | 1 | 1 | 1 | 3 | 0 | 2 | |  |  |  |  | | C | 1 | 0 | 4 | 1 | 5 | 0 | 7 | 0 | 6 | |  |  |  |  | | S0= +A | | | | | | | | | | |
|  | |

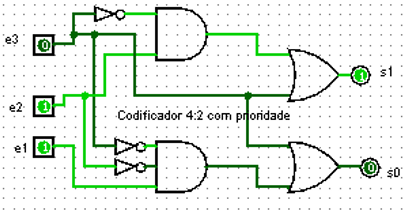
|  |  |
| --- | --- |
| **Exercício:**  **O deposito de uma mota tem capacidade para 7 litros de combustível. Desenvolve um descodificador que indique sempre que o deposito atinga os 2, 3 e 5 litros de combustível.** | |
| **3 entradas e 8 saídas, ou seja, 3:8**  **Porque base 2 elevado a 3 input, ou seja, 2^3=8** | |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | **A** | **B** | **C** | **s0** | **Posição** | | 0 | 0 | 0 |  | 0 | | 0 | 0 | 1 |  | 1 | | 0 | 1 | 0 |  | 2 | | 0 | 1 | 1 |  | 3 | | 1 | 0 | 0 |  | 4 | | 1 | 0 | 1 |  | 5 | | 1 | 1 | 0 |  | 6 | | 1 | 1 | 1 |  | 7 | | |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | BA |  | |  | |  | |  | | | C |  | 00 | | 01 | | 11 | | 10 | | |  | 0 |  | 0 |  | 1 |  | 3 |  | 2 | |  |  |  |  | | C | 1 |  | 4 |  | 5 |  | 7 |  | 6 | |  |  |  |  | |  | | | | | | | | | | |
|  | |

## Codificadores de prioridade

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Entradas** | | | | **Saídas** | |  |  |
| **e3** | **e2** | **e1** | **e0** | **s1** | **s0** | **s1** | **S0** |
| 0 | 0 | 0 | 0 | x | x | x | x |
| 0 | 0 | 0 | 1 | 0 | 0 | x | x |
| 0 | 0 | 1 | x | 0 | 1 | x | e3’e2’e1 |
| 0 | 1 | x | x | 1 | 0 | e3’e2 | x |
| 1 | x | x | x | 1 | 1 | e3 | e3 |

S1=e3+e3’e2

S0=e3’e2’e1+e3



Notas Finais