

BLOCOS COMBINATÓRIOS FUNDAMENTAIS

Tópicos

- *Multiplexers.*
- Síntese de circuitos combinatórios genéricos com multiplexers e lógica elementar.
- Simulação no Quartus Prime.

Exercícios

1. Utilizando os blocos elementares apresentados na Figura 1, projecte um sistema de multiplexagem que permita seleccionar uma entre 8 palavras de 4 *bits* cada.

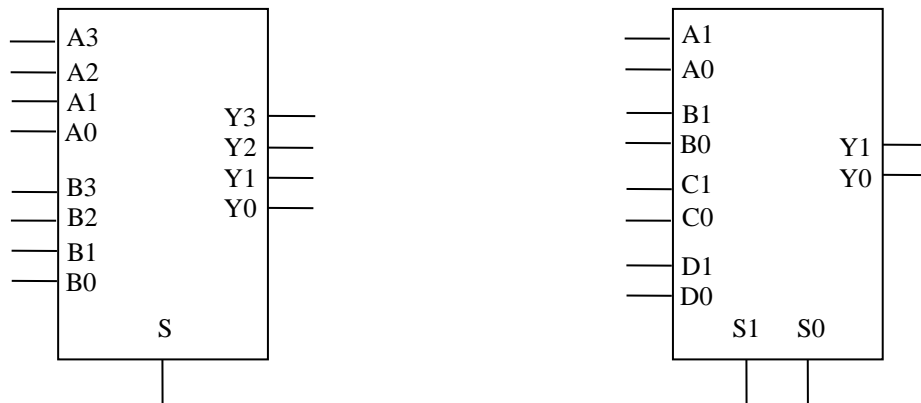


Figura 1 Multiplexers elementares: quad 2:1 (74157) e dual 4:1 (74153).

Justifique as suas opções e simule o circuito recorrendo aos componentes da família “74xxx” disponíveis na biblioteca “maxplus2” do Quartus Prime.

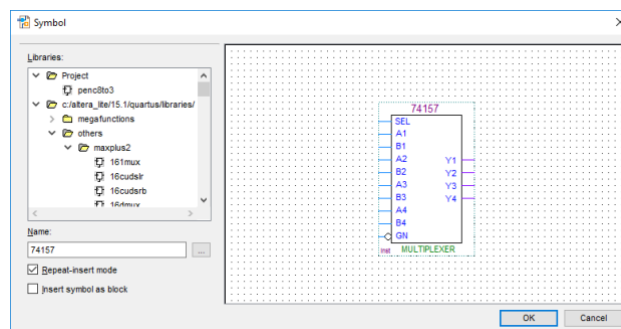


Figura 2 Pesquisa de componentes da família “74xxx” na biblioteca maxplus2 disponível no ambiente Quartus Prime.

2. Implemente o mesmo sistema de multiplexagem do problema anterior, agora com base em *buffers* de 3 estados do tipo 74244 (e lógica adicional que julgue pertinente utilizar). Compare as duas soluções em termos do número de circuitos integrados necessário.
3. Implemente a função $F(A, B, C, D) = A + C' \cdot D + B \cdot D' + B' \cdot D + B' \cdot C$, usando apenas um *multiplexer*, as variáveis independentes (não complementadas) e as constantes “0” e “1”.

Exercícios Complementares

4. Sugira implementações da função $f(A, B, C, D) = \sum m_{A,B,C,D}(0,3,5,7,11,12,13,15)$ baseadas em:

- a) *Multiplexer* 16:1
- b) *Multiplexer* 8:1
- c) *Multiplexer* 4:1 e lógica elementar adicional.

[Nota: nas alíneas b) e c), admita que dispõe das variáveis também na forma complementada]

5. Pretende-se construir um subsistema computacional com 2 entradas de dados, A e B, e 3 entradas de controlo, C2, C1 e C0. A saída do circuito, F, obedece à seguinte a tabela de verdade:

| C2 | C1 | C0 | F |
|----|----|----|--------------------------|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | $\overline{A+B}$ |
| 0 | 1 | 0 | $\overline{A \bullet B}$ |
| 0 | 1 | 1 | $\overline{A \oplus B}$ |
| 1 | 0 | 0 | $\overline{A \oplus B}$ |
| 1 | 0 | 1 | $\overline{A \bullet B}$ |
| 1 | 1 | 0 | $\overline{A+B}$ |
| 1 | 1 | 1 | 0 |

Implemente o circuito com base num *multiplexer* e lógica elementar adicional.