

MC603 – 2005

Experiência N° 5

1 - Projete, em VHDL, um MUX-4 duplo (similar ao 74153). Usando 4 desses dispositivos implemente um somador de palavras de 4 bits. Programe a FLEX e verifique seu funcionamento, utilizando o Display de 7 segmentos . [RD]¹.

Dica: observar que:

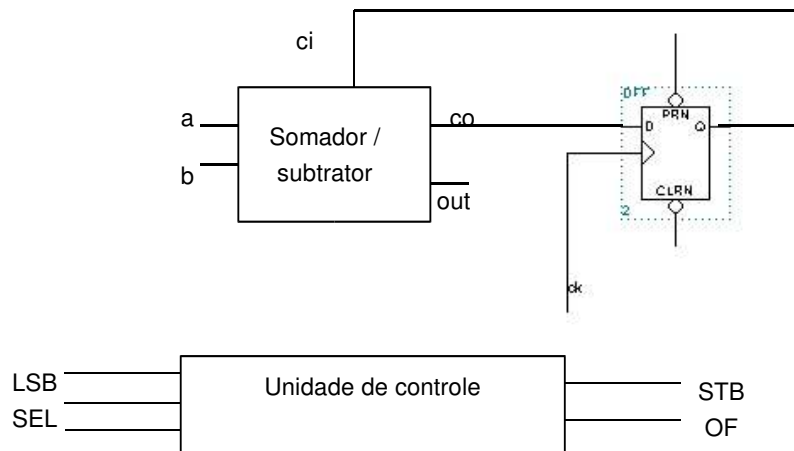
$$c_0 = 0 \cdot (\bar{a} \cdot \bar{b}) + c_i (a \cdot \bar{b} + \bar{a} \cdot b) + 1 \cdot (a \cdot b)$$

2 - Projetar, em VHDL, um somador/subtrator serial, totalmente síncrono, capaz de somar e subtrair palavras seriais de 8 bits (A e B) e entregar o resultado em paralelo na saída (via um registrador de deslocamento), juntamente com um sinal de overflow (OF) e um sinal de strobe (STB) para indicar que os resultados estão prontos. Os sinais de controle na entrada são:

- LSB: indica que o primeiro bit da entrada, o menos significativo, está presente; os bits entram do menos para o mais significativo;
- SEL: se for igual a 1 a operação é soma, caso contrário é subtração.

O diagrama de blocos do circuito é o esboçado abaixo. Escolha os sinais de controle internos e implemente-os convenientemente como saída da unidade de controle. [RSD]

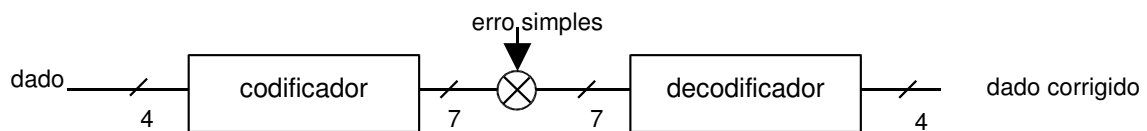
¹ R= relatório; S= simulação; D = demonstração



3 - Projete, usando VHDL, um dado eletrônico que mostre aleatoriamente valores de 1 a 6 no display de 7 segmentos ao se pressionar um push button. Colete dados para demonstrar que este "dado" não é viciado. [DSR]

4 – Usando VHDL, implemente um circuito de correção de erros de entradas codificadas segundo o código de Hamming de distância 3 bits. O circuito deve ser capaz de corrigir erros simples (em um bit). As entradas têm 4 bits de dados e 3 de codificação (segundo especificação distribuída) e podem ser corrompidas em um bit. O circuito decodificador detecta o erro e corrige, apresentando os 4 bits de dados corrigidos. A demonstração deve permitir especificar as entradas (4 bits), inserir erros (em 7 posições) e verificar a saída corrigida e não corrigida . [DSR].

A figura ilustra o funcionamento do circuito:



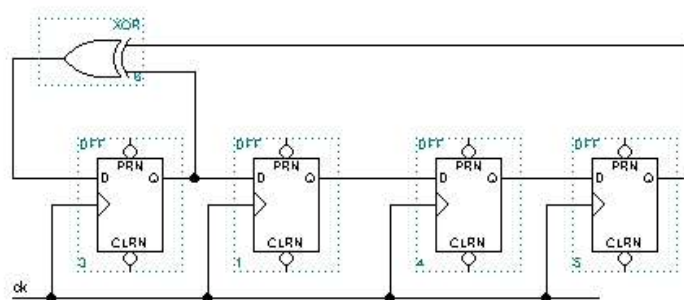
dica: consultar sites:

<http://www.mdstud.chalmers.se/~md7sharo/coding/main/node32.html>

<http://www.ee.unb.ca/tervo/ee4253/hamming.htm>

5 – Implementar (em VHDL) os dois LFSR (Linear Feedback Shift Register) mostrados nas figuras abaixo. Mostrar no relatório a simulação com todos os estados possíveis e a sequência de estados visitados através de diagrama de transição de estados. Usar, para a demonstração, o clock lento e o display de 7 segmentos para mostrar o estado. O circuito deve permitir visitar todos os 16 estados possíveis. Nos circuitos mostrados, as entradas de PRN e CLRN estão desativadas. [RSD]

5-a - LFSR com polinômio primitivo (primo)



5-b - LFSR com polinômio não primitivo (não primo)

