



universidade de aveiro

Hamming Codes

Arquiteturas de Alto Desempenho

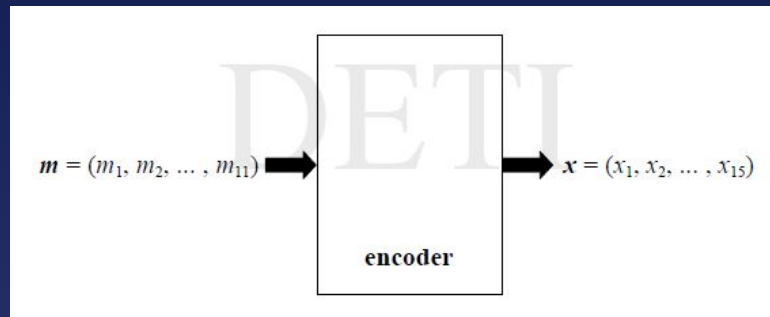
António Domingues (89007) & Tiago Dias (88896)

TP4 - Grupo 05

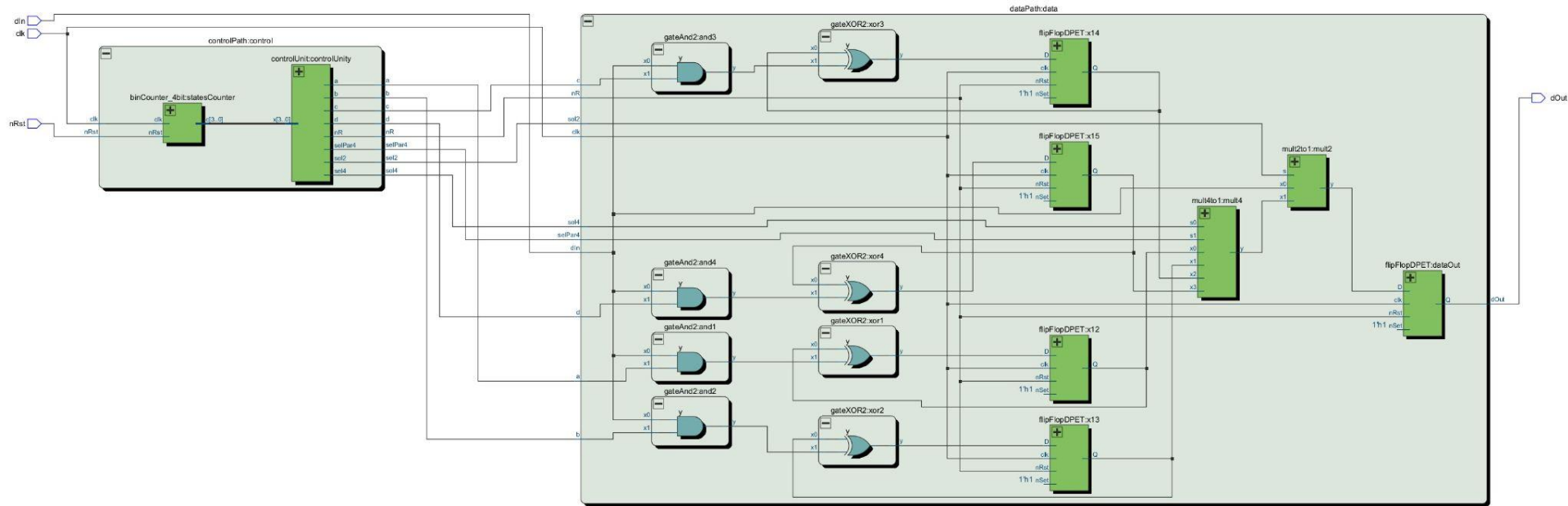


Encoder

- Versão *bit serial*;
- Composto por dois módulos principais:
 - *Control path*;
 - *Data path*;
- Necessários apenas 4 *x-ors*;
- Atrasos de propagação:
 - 5 AND
 - 1 XOR
 - 2 OR
 - 2 NOT



Encoder

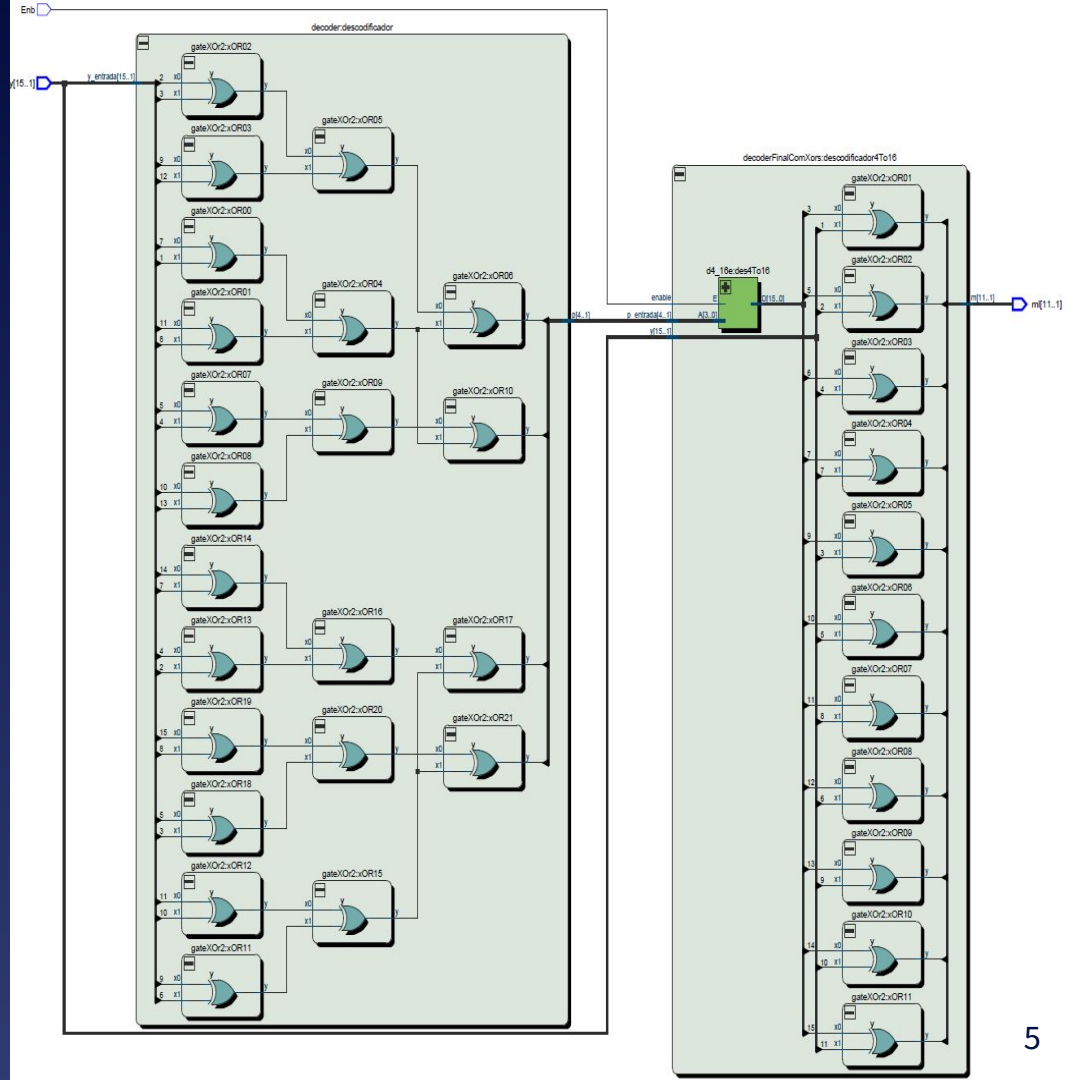


Encoder

1. Pegando na expressão $x_{12} = m_1 \oplus m_2 \oplus m_3 \oplus m_7 \oplus m_8 \oplus m_9 \oplus m_{11}$ foi necessário uniformizar as variáveis da mesma para tornar o processo de x_{12} , x_{13} , x_{14} e x_{15} uniforme. Segundo este pressuposto foi concluído o seguinte:
 - $x_{12} = x_{12} \oplus (a_i \cdot m_i)$, for $i = 1, 2, \dots, 11$
2. O procedimento anteriormente referido foi feito de igual forma para x_{13} , x_{14} e x_{15} .
3. Foram necessários *flip flops* para armazenarem os valores de x_{12} , x_{13} , x_{14} e x_{15} .
4. Um *multiplexer* de 4:1 foi utilizado para selecionar qual das variáveis x_{12} , x_{13} , x_{14} ou x_{15} seria selecionada. Este *multiplexer* está ligado a um outro *multiplexer* mas desta vez de 2:1 para alternar entre as variáveis referidas previamente ou simplesmente o *bit* de entrada apelidado de Din.
5. Para emitir o *output*, foi colocado outro *flip flop* á saída do *multiplexer* 2:1 com o valor de saída do mesmo.
6. De forma a controlar este processo todo foi criada uma unidade de controlo juntamente com um contador de estados de 4 *bits*.

Decoder

- Versão paralela;
- Redução de 28 x -ors para 22;
- Atrasos de propagação de 3 x -ors no pior caso.



Decoder

- **Análise prévia das operações que se repetem ao longo das expressões**
 - Visto que apenas existem propriedades associativas, surge a possibilidade de reutilização de portas, para uso posterior, a fim de poupar portas lógicas assim como atrasos de propagação;
 - Em $p1$, é possível agrupar-se $y1 \oplus y7 \oplus y8 \oplus y11$, utilizando o resultado para em $p2$ economizando portas lógicas;
 - O mesmo acontece em $p3$, reutilizando o resultado de $y6 \oplus y9 \oplus y10 \oplus y11$ em $p4$
 - Implementação cuidadosa de modo a reduzir o atraso de propagação de 7 portas *x-or*, para 3;
 - Solução desenhada implementando 3 níveis de portas *x-or* 2:1 em cascata
- Acréscimo de ou exclusivos nas saídas do decodificador para os vários valores de p correspondentes às colunas da matriz *parity check*, a fim de fazer a correção da palavra se a mesma estiver incorreta ($p \neq 0$)