

Cyclic Redundancy Check - 8

ACA

André Alves 88811
Renato Valente 89077

Departamento de Eletrónica, Telecomunicações e Informática
Universidade de Aveiro

December 2020

Encoder

Para a codificação dos dados usamos as Propriedades do módulo (Properties of the remainder).

$$\left(\sum_{n=0}^{15} a_n \times x^{n+8}\right) \bmod (x^8 + x^5 + x^3 + x^2 + x + 1)$$

Com a resolução da fórmula anterior podemos concluir que são necessários **64 xor** gates e **11 xor** de atraso de propagação no pior caso.

Para reduzir o número de gates recorremos à utilização de operações comuns com execução em paralelo para a redução do atraso de propagação.

Obtivemos uma redução na implementação para **38 XOR**-gates e **4 XOR**-gates de atraso de propagação no pior caso.

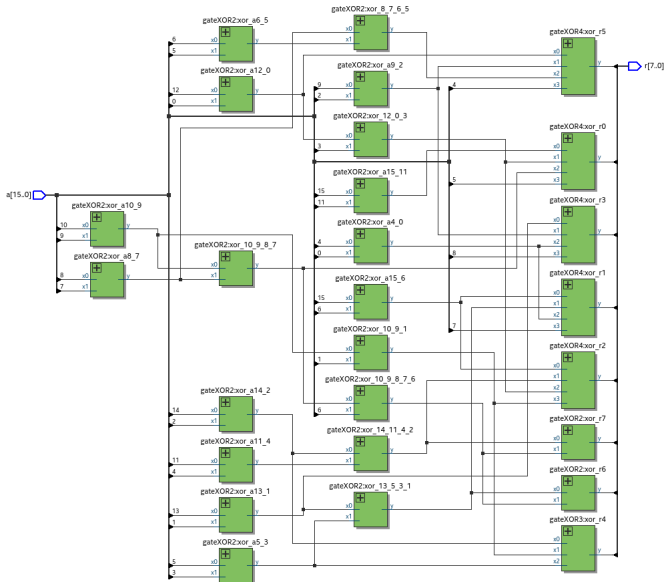
A maneira como reduzimos as XOR-gates e a arquitetura do encoder encontram-se nos slides seguintes.

Tabela de XOR-gates

[illegible]

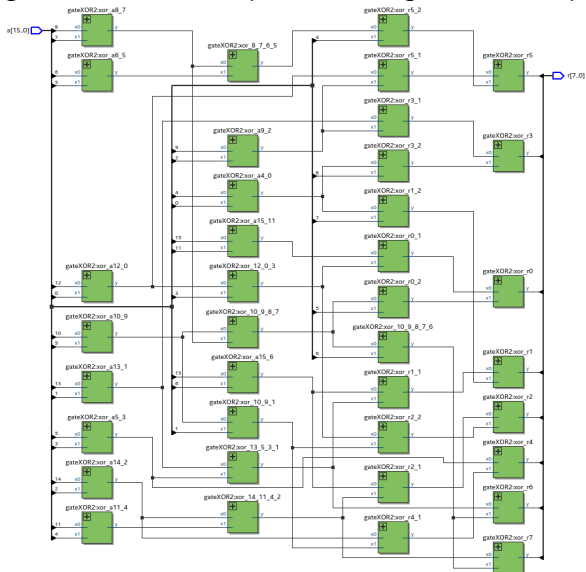
As cores iguais equivalem à junção dessas entradas para poderem ser reaproveitadas ou para diminuir o atraso. Agrupámos 19 vezes as várias entradas, dando por isso, 19 XOR-gates. O Encoder agora com menos entradas precisa apenas de 19 XOR-gates, dando um total de 38 XOR-gates.

Arquitetura



Arquitetura Extenso

Igual à anterior mas passando as gates XOR-4 para gates XOR-2.



Verificar se existem erros introduzidos na transmissão.

Para isso aceitamos a entrada de $\mathbf{a}(\mathbf{x}) \times \mathbf{x}^8 + \mathbf{r}$ ($\mathbf{a_r}$), em que o $\mathbf{a}(\mathbf{x})$ é a nossa word em bits e o \mathbf{r} é o resto da divisão do $\mathbf{a}(\mathbf{x})$ pelo polinómio $\mathbf{b}(\mathbf{x})$. Usamos os 16 primeiros bits da entrada, $\mathbf{a_r}$ na entrada \mathbf{a} do encoder e recebemos o \mathbf{r} . Depois fazemos comparação bit a bit do \mathbf{r} dado pelo encoder com os 8 bits menos significativos da entrada $\mathbf{a_r}$ usando XOR-gates de 2 entradas. Para não dar erro é necessário que todos os bits de comparação sejam 0, por isso usamos um OR-gate de 8 entradas, em que os bits de comparação são as entradas e a saída é a saída do checker. Saíndo 0 significa que a mensagem não contém erros. Esquema da arquitetura no slide seguinte.

Arquitetura Checker

