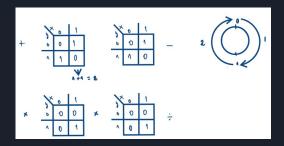
Cyclic Redundancy Check -8

DETI-UA MIECT/MECT Guilherme Claro -98432

Tiago Mostardinha-103944

Algoritmo de Módulos

- O algoritmo do CRC depende do algoritmo do módulo, para obter o resto, para que posteriormente, do lado do receptor, se confirme se os dados foram, devidamente, enviados.
- Os dados, que estão em binário, vão ser operados por um Galois Field F2, onde G(F2) = 2
 - \circ F2 = {0,1}
- Como o nosso F2 é apenas de 2 elementos, a adição ou subtração serão as mesmas, assim como a multiplicação e a divisão.



• O algoritmo do CRC será então o seguinte, onde o b(x) será o polinómio escolhido.

$$[a(x) \times x^8] \mod b(x) = \left[\left(\sum_{n=0}^{15} a_n \times x^n \right) \times x^8 \right] \mod b(x)$$

Tabela de xor-gates

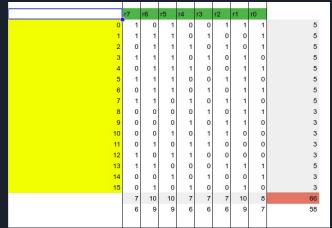




Fig1: Tabela Inicial

Polinómio do CRC: $x^{8} + x^{7} + x^{5} + x^{2} + x + 1$

Para obter a tabela acima usamos a propriedade do módulo que nos deu o custo de implementação inicial.

Custo de implementação inicial - 58 xor gates, 9 xor gates no pior caso

Depois feita a redução do número de xor-gates através de operações comuns com execução em paralelo para reduzir o atraso de propagação (tabela ao lado).

Custo de implementação final - 39 xor gates, 4 xor gates no pior caso



Fig2: Tabela Xor Simplificado

Encoder paralelo- arquitetura

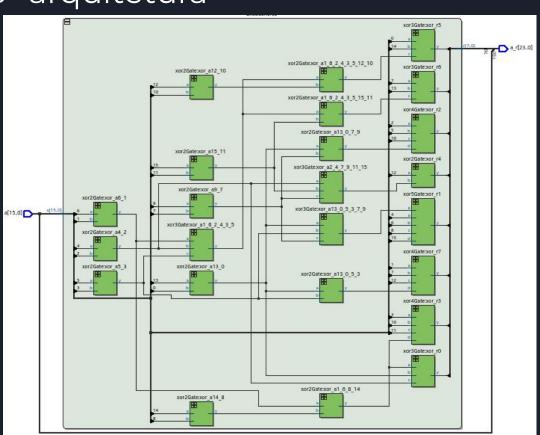
Portos:

o a: 16 bits de dados

o a_r: 24 bits dados + resto

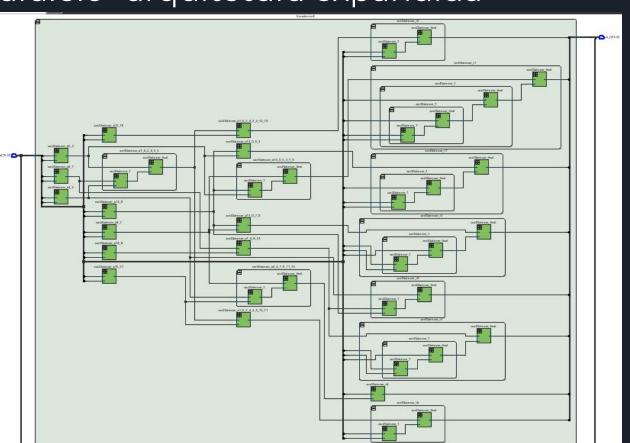
• Comportamento:

- Entram os 16 bits para dividendo do CRC, onde são processados e obtém-se o resto, que são 8 bits
- O resultado do CRC é acrescentado aos 16 bits de dados e são enviados, 23 bits, dados + resto



Encoder paralelo- arquitetura expandida

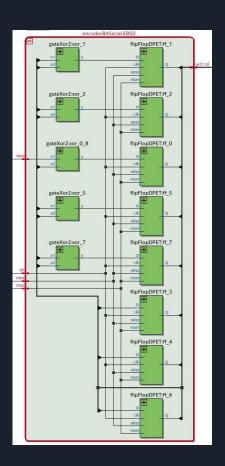
Versão expandida com unicamente xor gates de duas entradas



Encoder Serial

Encoder CRC-8 versão bit serial para a implementação do checker

- Os dados são recebidos bit a bit,em cada ciclo de relógio, no encoder, e são processados por **8 flip flop D e 5 xor gates.**
- 8 Flip Flops D para cada bit de entrada do polinomio
- **5 Xor gates** para cada bit que se encontra a 1 no polinomio
- O flip flop do MSB compara o bit,
 - **Se 0**, é introduzido e os restantes bits são enviados para o próximo flip flop
 - **Se 1**, é calculado o resto e é enviado para o proximo flip flop
- No **24° ciclo** o encoder obtém o resultado válido



Checker Serial- Arquitetura

Blocos implementados para a arquitetura do checker em modo bit serial:

- **16 bit Shift Register (PISO):** 16 Mux + 16 Flip Flops
 - Recebe 16 dados e envia bit a bit por cada ciclo de relogio
- **8 bit Register:** 8 Mux + 8 Flip Flops
 - Recebe 8 bits e guarda nos próximos ciclos
- Control Unit
 - Responsável por orquestrar os componentes do checker
- 32 bit Pop Counter
- **errorBlock_8bits:** 8 Xor + 7 Or + 1 Not
 - Realiza a verificação se os dados são válidos
- **EncoderBitSerial:** 8 Flip Flops + 5 Xor
 - Responsável por processar o CRC

