

universidade de aveiro



deti

departamento de electrónica,
telecomunicações e informática

Mestrado Integrado em Engenharia de Computadores e Telemática

Arquitetura de Computadores Avançada

Assignment 1 – Cyclic Redundancy Check

Pedro Santos, nº 76532
José Pedro Duarte, nº 64194

Prof. António Rui Borges

Cyclic Redundancy Check (CRC-8)

Método de detecção de erros na transmissão de cadeias de dados baseado no resto da divisão polinomial.

Com base nisto, associando cada bit ao correspondente grau polinomial, calculamos o resultado da divisão polinomial aplicado a um campo de Galois de 2 elementos para cada um dos bits individualmente. Os resultados obtidos podem ser representados de acordo com a tabela seguinte:

	x8	x9	x10	x11	x12	x13	x14	x15	x16	s17	x18	x19	X20	x21	x22	x23
x7	1	0	1	0	0	1	1	1	0	0	0	0	1	1	1	1
x6	1	1	1	0	1	0	0	1	0	0	0	1	0	0	0	0
x5	0	1	1	1	0	1	0	1	0	0	1	0	1	1	1	1
x4	1	1	1	0	1	0	1	0	0	1	0	1	1	1	1	0
x3	0	1	1	1	0	0	1	1	1	0	1	1	0	0	1	1
x2	1	1	1	0	0	1	1	1	0	1	1	0	0	1	1	0
x1	0	1	1	0	1	0	0	1	1	1	0	0	0	0	1	1
x0	1	1	0	1	0	0	1	1	1	0	0	0	0	1	1	1

Aos valores obtidos e representados na tabela anterior aplicámos um programa que construímos para detetar recursivamente pares de valores repetidos em diferentes equações e ao encontrar, agrupá-los (respeitando algumas restrições).

Os resultados obtidos permitiram reduzir a complexidade da solução reduzindo o número de gates a usar. O resultado foi o seguinte considerando que as somas entre parêntesis foram agrupadas para serem reusadas posteriormente:

$$r^7 = x^{14} + x^{22} + x^{23} + (x^{10} + x^{15}) + (x^8 + x^{21}) + (x^{13} + x^{20})$$

$$r^6 = x^8 + x^9 + (x^{10} + x^{15}) + (x^{12} + x^{19})$$

$$r^5 = x^{21} + x^{23} + (x^{11} + x^{18}) + (x^{13} + x^{20}) + ((x^{10} + x^{15}) + (x^9 + x^{22}))$$

$$r^4 = x^{10} + x^{20} + (x^{14} + x^{17}) + (x^{12} + x^{19}) + ((x^8 + x^{21}) + (x^9 + x^{22}))$$

$$r^3 = x^{14} + x^{19} + (x^{11} + x^{18}) + ((x^{16} + x^{23}) + ((x^{10} + x^{15}) + (x^9 + x^{22})))$$

$$r^2 = x^{13} + x^{18} + (x^8 + x^{21}) + (x^{14} + x^{17}) + ((x^{10} + x^{15}) + (x^9 + x^{22}))$$

$$r^1 = x^{12} + x^{17} + ((x^{16} + x^{23}) + ((x^{10} + x^{15}) + (x^9 + x^{22})))$$

$$r^0 = x^{11} + x^{14} + x^{15} + (x^{16} + x^{23}) + (x^8 + x^{21}) + (x^9 + x^{22})$$

As equações acima referenciadas foram usadas para o caso do Encoder, mas a lógica é semelhante para o Checker. Apenas se tem que considerar em cada uma das equações o correspondente bit dos 8 bits do resto previamente obtidos através do encoder.

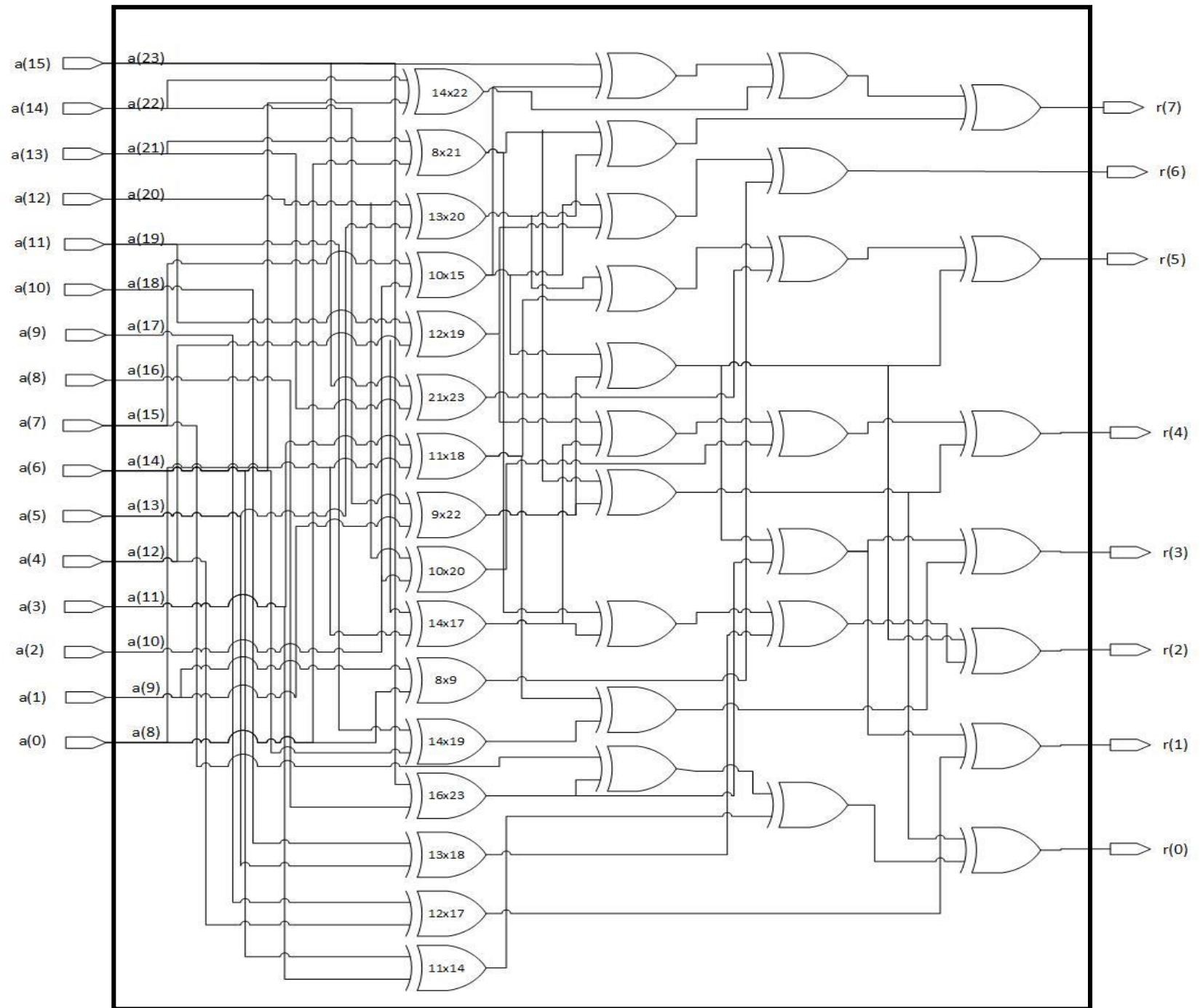
No caso do Checker, para verificar se existiu erro na transmissão da mensagem basta verificar se algum dos valores obtidos se encontra a um, ou seja, se o resto da divisão polinomial da soma do resto com os dados pelo polinómio geral (CRC-8) é diferente de zero.

Encoder

Considerando que cada gate XOR tem um atraso de 4 ns:

Complexidade: 40 gates XOR

Atraso total: 4 níveis de XORs correspondem a um atraso total de (4×4) 16 ns.



Checker

Considerando que cada gate XOR tem um atraso de 4 ns e cada gate OR um atraso de 2 ns:

Complexidade: 48 gates XOR + 7 gates OR

Atraso total: 4 níveis de XORs correspondem a um atraso de (4×4) 16 ns. 3 níveis de ORs correspondem a um atraso de (3×2) 6 ns. Atraso total de $(16 + 6)$ 22 ns.

