universidade de aveiro

theoria poiesis praxis

Arquitecturas de Alto Desempenho

CRC Design

António Rui Borges

Application areas

Two basic application areas are considered

- message transmission
  - bit serial transmission
- data storage
  - parallel access.

Engineering problem to be dealt with

- how confidant can one be that the received message, or the retrieved data, is the same as the one that was transmitted, or stored?

Solution to be pursued

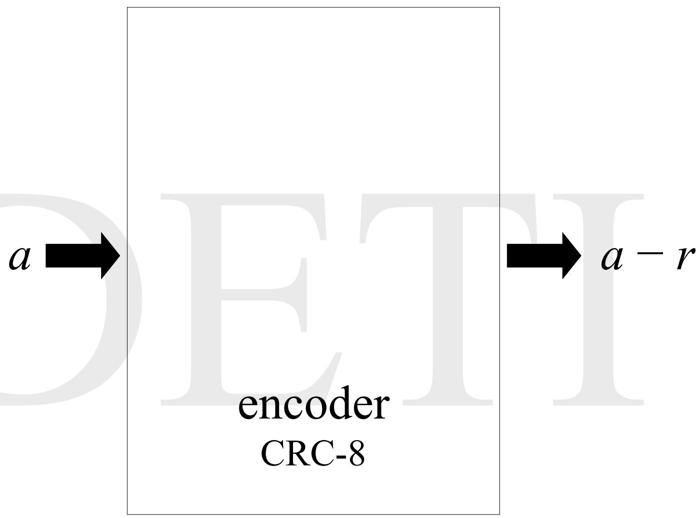
The message, or data, bits will be thought of to represent the coefficients of a polynomial to be operated in the Galois Field F_2.

The remainder $r(x)$, Cyclic Redundancy Checksum (CRC), of the polynomial division of $a(x) \times 10^8$ by $b(x) = x^8 + x^7 + x^5 + x^2 + x + 1$ is to be computed and attached to the message before transmission, or to the data before storage.

Upon message reception, or data retrieval, the polynomial $a(x) \times 10^8 - r(x)$ is to be divided again by $b(x)$ and, if the remainder is not zero, an error should be signaled.

Requirements - 1

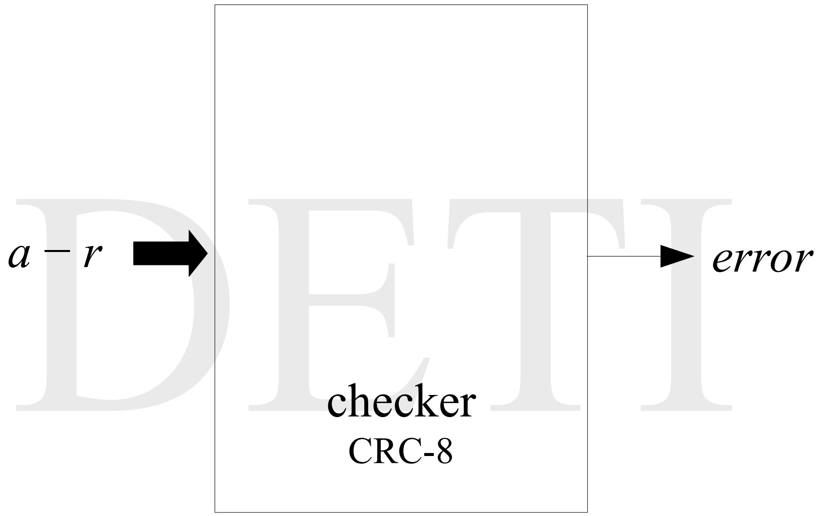
Parallel version



a – 16 bit word
r – 8 bit word

Requirements - 2

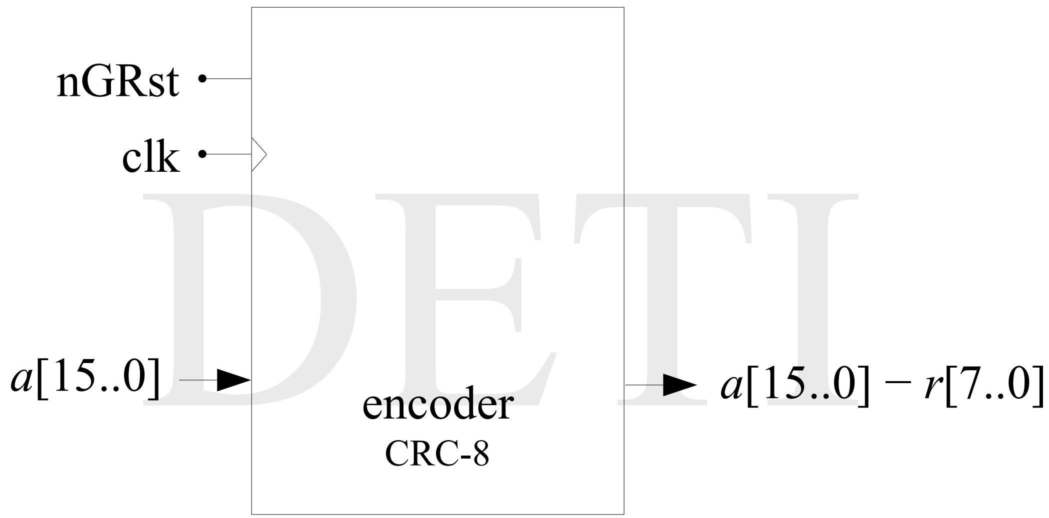
Parallel version



a – 16 bit word
r – 8 bit word
error – 1 bit word

Requirements - 3

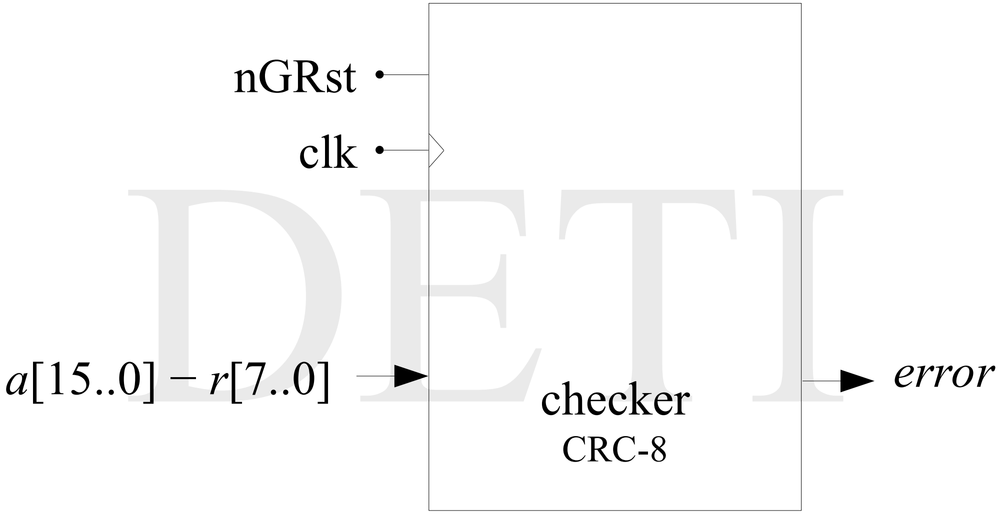
Bit serial version



a – msb is inputted / outputted first
r – msb is outputted first

Requirements - 4

Bit serial version



a – msb is inputted first
r – msb is inputted first

Basic approaches

The design may be approached through different methods, such as

- the division algorithm
- properties of the remainder.

Division Algorithm - 1

$$a(x) \times x^8 = q(x) \times b(x) + r(x)$$

$$\text{where} \quad b(x) = x^8 + x^7 + x^5 + x^2 + x + 1 \quad (\text{CRC}-8\,\text{Bluetooth})$$

$$\overbrace{a_{15}a_{14}a_{13}a_{12}a_{11}a_{10}a_9a_8a_7a_6a_5a_4a_3a_2a_1a_0\,0\,0\,0\,0\,0\,0\,0\,0}^{a(x)\times x^8} \quad \overbrace{1\,1\,0\,1\,0\,0\,1\,1\,1}^{b(x)}$$

$$\underbrace{\#\,\#\,\#\,\#\,\#\,\#\,\#\,\#\,\#\,\#\,\#\,\#\,\#\,\#\,\#\,\#}_{q(x)}$$

$r_{16}(x) \rightarrow \boxed{a_{15}a_{14}a_{13}a_{12}a_{11}a_{10}a_9a_8}\,a_7a_6a_5a_4a_3a_2a_1a_0\,0\,0\,0\,0\,0\,0\,0\,0$

$r_{15}(x) \rightarrow 0\,\boxed{\#\,\#\,\#\,\#\,\#\,\#\,\#\,\#}\,a_6$

$r_{14}(x) \rightarrow 0\,\boxed{\#\,\#\,\#\,\#\,\#\,\#\,\#\,\#}\,a_5$

$\cdots$

$r_9(x) \rightarrow 0\,\boxed{\#\,\#\,\#\,\#\,\#\,\#\,\#\,\#}\,a_0$

$r_8(x) \rightarrow 0\,\boxed{\#\,\#\,\#\,\#\,\#\,\#\,\#\,\#}\,0$

$\cdots$

$r_1(x) \rightarrow 0\,\boxed{\#\,\#\,\#\,\#\,\#\,\#\,\#\,\#}\,0$

$$0\,\underbrace{\boxed{\#\,\#\,\#\,\#\,\#\,\#\,\#\,\#}}_{r_0(x)=r(x)}$$

Division Algorithm - 2

The computation can be simplified if we take into consideration that

- only the polynomial $r(x)$ is required
- the last 8 coefficients of polynomial $a(x) \times x^8$ are known to be zero
- the form of polynomial $b(x)$ is fixed and known.

Division Algorithm - 3

Description of the computation as a recurring process

- there are 16 iteration steps
- initialization

$$r_{16,k} = a_{15+k-7} \quad , \text{ with } k = 0,1, \cdots ,7$$

- iteration step ($15 \geq i \geq 0$)

$$r_{i,8} = r_{i+1,7} \oplus q_i = r_{i+1,7} \oplus r_{i+1,7} = 0$$

$$k = 7, 5, 2, 1 \Rightarrow r_{i,k} = r_{i+1,7} \oplus r_{i+1,k-1}$$

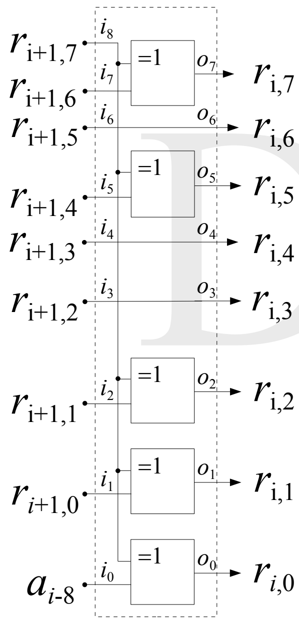
$$k = 6, 4, 3 \Rightarrow r_{i,k} = r_{i+1,k-1}$$

$$k = 0 \wedge i \geq 8 \Rightarrow r_{i,0} = r_{i+1,7} \oplus a_{i-8}$$

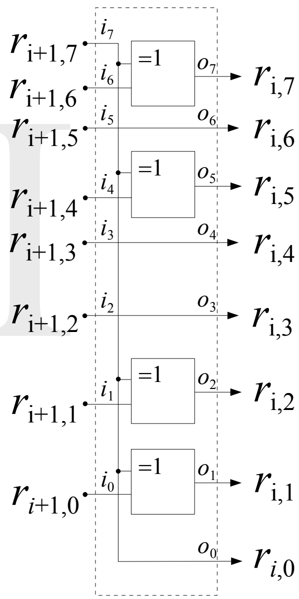
$$k = 0 \wedge i < 8 \Rightarrow r_{i,0} = r_{i+1,7}$$

Division Algorithm - 4

Two basic building blocks are needed.

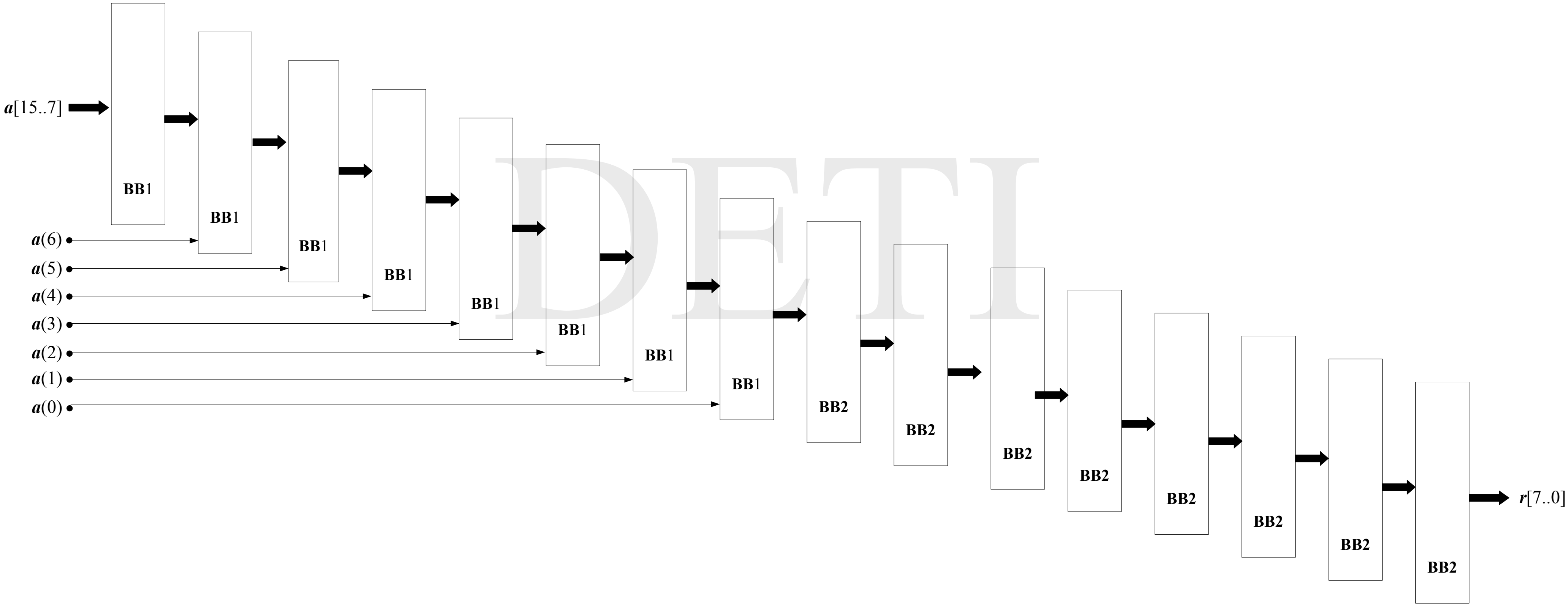


<u>building block of type 1</u>
9 inputs



<u>building block of type 2</u>
8 inputs

Division Algorithm - 5



Division Algorithm - 6

Output to input dependence + cost

iteration number

	r7	r6	r5	r4	r3	r2	r1	r0
16	8000	4000	2000	1000	800	400	200	100
15	C000	2000	9000	800	400	8200	8100	8080
14	E000	9000	C800	400	8200	4100	4080	C040
13	7000	C800	E400	8200	4100	A080	2040	E020
12	B800	E400	F200	4100	A080	5040	9020	7010
11	5C00	F200	F900	A080	5040	2820	C810	B808
10	AE00	F900	FC80	5040	2820	9410	E408	5C04
9	5700	FC80	FE40	2820	9410	4A08	F204	AE02
8	AB80	FE40	7F20	9410	4A08	A504	F902	5701
7	55C0	7F20	3F90	4A08	A504	5282	FC81	AB80
6	2AE0	3F90	1FC8	A504	5282	A941	FE40	55C0
5	1570	1FC8	8FE4	5282	A941	D4A0	7F20	2AE0
4	0AB8	8FE4	47F2	A941	D4A0	6A50	3F90	1570
3	855C	47F2	A3F9	D4A0	6A50	3528	1FC8	0AB8
2	C2AE	A3F9	51FC	6A50	3528	9A94	8FE4	855C
1	6157	51FC	A8FE	3528	9A94	4D4A	47F2	C2AE
0	**30AB**	**A8FE**	**547F**	**9A94**	**4D4A**	**26A5**	**A3F9**	**6157**

$a_{15} \cdots a_0 \rightarrow$ 1, if the variable is present in the expression
0, otherwise

• 72 x-or gates are needed.

Division Algorithm - 7

Propagation delay dependence

iteration number

	pdr7	pdr6	pdr5	pdr4	pdr3	pdr2	pdr1	pdr0
16	0	0	0	0	0	0	0	0
15	1	0	1	0	0	1	1	1
14	2	1	2	0	1	2	2	2
13	3	2	3	1	2	3	3	3
12	4	3	4	2	3	4	4	4
11	5	4	5	3	4	5	5	5
10	6	5	6	4	5	6	6	6
9	7	6	7	5	6	7	7	7
8	8	7	8	6	7	8	8	8
7	9	8	9	7	8	9	9	8
6	10	9	10	8	9	10	10	9
5	11	10	11	9	10	11	11	10
4	12	11	12	10	11	12	12	11
3	13	12	13	11	12	13	13	12
2	14	13	14	12	13	14	14	13
1	15	14	15	13	14	15	15	14
0	**16**	**15**	**16**	**14**	**15**	**16**	**16**	**15**

- 16 x-or propagation time delays in the worst case.

Properties of the remainder - 1

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

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

where $\quad b(x)=x^8+x^7+x^5+x^2+x+1 \quad (\mathrm{CRC-8\,Bluetooth})$

Properties of the remainder - 2

$$x^8 \; mod \; (x^8 + x^7 + x^5 + x^2 + x + 1) = x^7 + x^5 + x^2 + x + 1$$
$$x^9 \; mod \; (x^8 + x^7 + x^5 + x^2 + x + 1) = x^7 + x^6 + x^5 + x^3 + 1$$
$$x^{10} \; mod \; (x^8 + x^7 + x^5 + x^2 + x + 1) = x^6 + x^5 + x^4 + x^2 + 1$$
$$x^{11} \; mod \; (x^8 + x^7 + x^5 + x^2 + x + 1) = x^7 + x^6 + x^5 + x^3 + x$$
$$x^{12} \; mod \; (x^8 + x^7 + x^5 + x^2 + x + 1) = x^6 + x^5 + x^4 + x + 1$$
$$x^{13} \; mod \; (x^8 + x^7 + x^5 + x^2 + x + 1) = x^7 + x^6 + x^5 + x^2 + x$$
$$x^{14} \; mod \; (x^8 + x^7 + x^5 + x^2 + x + 1) = x^6 + x^5 + x^3 + x + 1$$
$$x^{15} \; mod \; (x^8 + x^7 + x^5 + x^2 + x + 1) = x^7 + x^6 + x^4 + x^2 + x$$
$$x^{16} \; mod \; (x^8 + x^7 + x^5 + x^2 + x + 1) = x^3 + x + 1$$
$$x^{17} \; mod \; (x^8 + x^7 + x^5 + x^2 + x + 1) = x^4 + x^2 + x$$
$$x^{18} \; mod \; (x^8 + x^7 + x^5 + x^2 + x + 1) = x^5 + x^3 + x^2$$
$$x^{19} \; mod \; (x^8 + x^7 + x^5 + x^2 + x + 1) = x^6 + x^4 + x^3$$
$$x^{20} \; mod \; (x^8 + x^7 + x^5 + x^2 + x + 1) = x^7 + x^5 + x^4$$
$$x^{21} \; mod \; (x^8 + x^7 + x^5 + x^2 + x + 1) = x^7 + x^6 + x^2 + x + 1$$
$$x^{22} \; mod \; (x^8 + x^7 + x^5 + x^2 + x + 1) = x^5 + x^3 + 1$$
$$x^{23} \; mod \; (x^8 + x^7 + x^5 + x^2 + x + 1) = x^6 + x^4 + x$$

Properties of the remainder - 3

$$\left(\sum_{n=0}^{15} a_n \times x^{n+8}\right) mod \left(x^8 + x^7 + x^3 + x^2 + x + 1\right) =$$

$$\begin{aligned}
= &\left(a_0 \oplus a_1 \oplus a_3 \oplus a_5 \oplus a_7 \oplus a_{12} \oplus a_{13}\right) \times x^7 + \\
&+ \left(a_1 \oplus a_2 \oplus a_3 \oplus a_4 \oplus a_5 \oplus a_6 \oplus a_7 \oplus a_{11} \oplus a_{13} \oplus a_{15}\right) \times x^6 + \\
&+ \left(a_0 \oplus a_1 \oplus a_2 \oplus a_3 \oplus a_4 \oplus a_5 \oplus a_6 \oplus a_{10} \oplus a_{12} \oplus a_{14}\right) \times x^5 + \\
&+ \left(a_2 \oplus a_4 \oplus a_7 \oplus a_9 \oplus a_{11} \oplus a_{12} \oplus a_{15}\right) \times x^4 + \\
&+ \left(a_1 \oplus a_3 \oplus a_6 \oplus a_8 \oplus a_{10} \oplus a_{11} \oplus a_{14}\right) \times x^3 + \\
&+ \left(a_0 \oplus a_2 \oplus a_5 \oplus a_7 \oplus a_9 \oplus a_{10} \oplus a_{13}\right) \times x^2 + \\
&+ \left(a_0 \oplus a_3 \oplus a_4 \oplus a_5 \oplus a_6 \oplus a_7 \oplus a_8 \oplus a_9 \oplus a_{13} \oplus a_{15}\right) \times x + \\
&+ \left(a_0 \oplus a_1 \oplus a_2 \oplus a_4 \oplus a_6 \oplus a_8 \oplus a_{13} \oplus a_{14}\right)
\end{aligned}$$

- 58 x-or gates are needed
- 9 x-or propagation time delays in the worst case.

Parallel implementation

Following one of the approaches that were described, or some other one that you may devise

- elicit common operations to reduce gate count
- perform them in parallel to reduce time propagation delays.

Bit serial implementation

Following one of the approaches that were described, or some other one that you may devise

- elicit common operations in order to specify the data path
- design the control section so that the bit sequence may proceed smoothly through the data path.