CRC-8

António Mota nº72622
João Pedro Fonseca nº73779

CRC or Cyclic Redundancy Check is a method of detecting accidental changes/errors in a message passed through a communication channel.

CRC uses a **Polynomial Generator** $b(x)$ that represents a key of ***p*** bits which is available on both sender and receiver side:

$$b(x) = \sum_{k=0}^{p-1} b_k x^k$$

$$b(x) = x^8+x^7+x^6+x^4+x^2+1$$ represents the binary key: **111010101**

This generator is used to encode a message $m(x)$ with ***n*** bits of data to be sent:

$$m(x) = \sum_{k=0}^{n-1} m_k x^k$$

Sender mechanism

Generation of Encoded Data from Data and Polynomial Generator

1. The binary data M is first augmented by adding **p-1** zeros in the end of the data
2. Use ***modulo-2 binary division*** to divide binary data by the **key** and store remainder of division.
3. Append the remainder at the end of the data to form the **encoded** data and send it

Calculation of CRC: $crc(x) = x^p m(x) \bmod b(x)$

Modulo 2 division

The process of modulo-2 binary division is the same as the familiar division process we use for decimal numbers. Just that instead of subtraction, we use XOR gates.

- In each step, a copy of the divisor $m(x)$ is **XOR**'ed with the ***p*** bits of the dividend $b(x)$
- The result of the **XOR** operation $q(x)$ is **n-1** bits, which is used for the next step after 1 extra bit is pulled down to make it **n** bits long.
- When there are no bits left to pull down, we have a result. The **n-1** bit remainder which is appended at the sender side.

Receiver mechanism

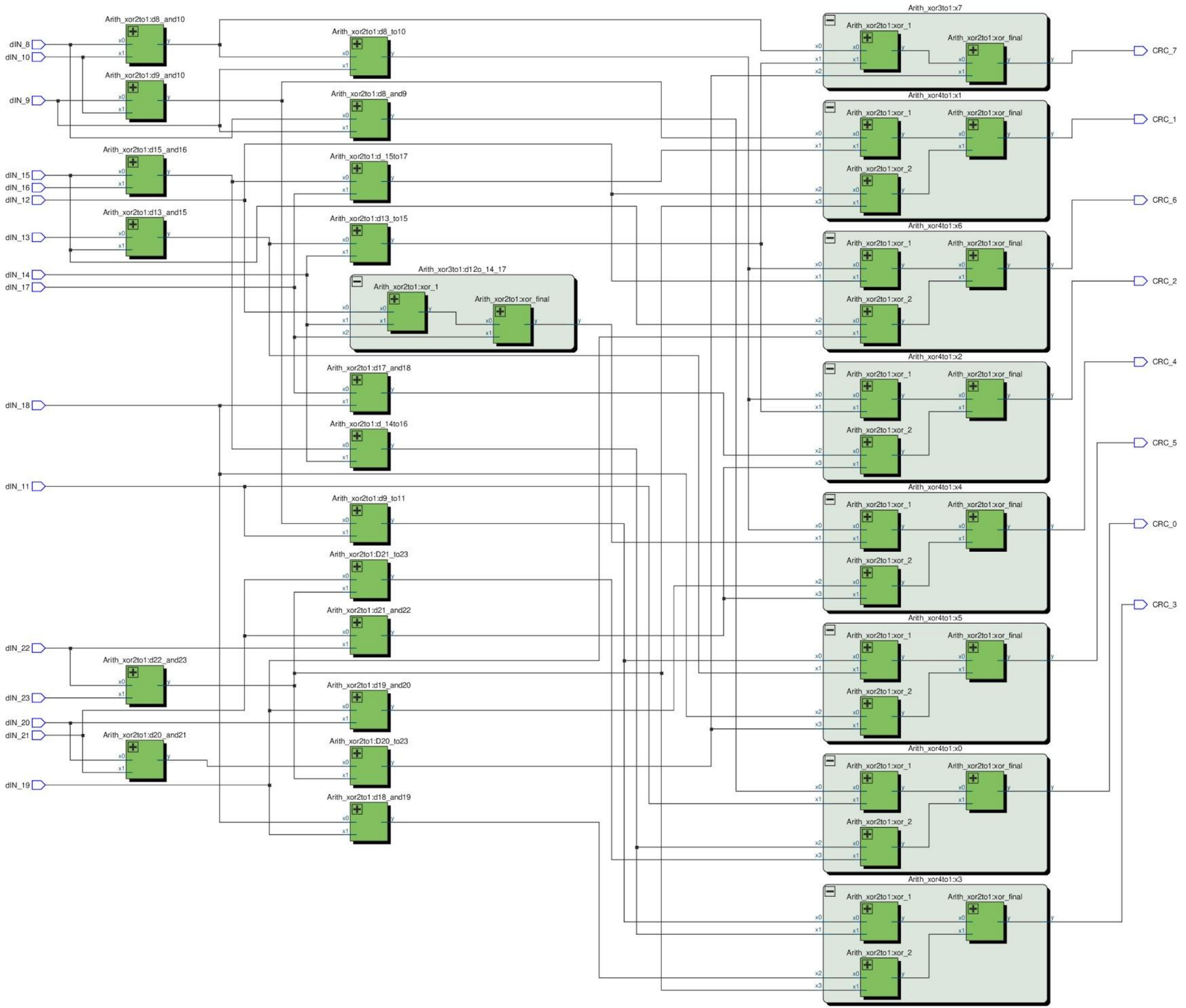
Check if there are errors introduced in transmission

1. Perform modulo-2 division again and if remainder is 0, then there are no errors.

Verification of CRC: $x^p m(x) - crc(x) = q(x)b(x)$

Must be zero

Encoder diagram



Checker diagram