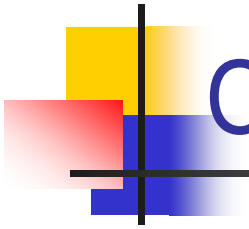
CNT 4104 Intro to Data Comm

Cyclic Redundancy Check

Dr. Sam Hsu
Computer Science & Engineering
Florida Atlantic University



CRC

- Key concepts
  - Polynomials
  - Modulo 2 arithmetic
  - Algorithm
- Hardware implementation
  - Shift registers.
  - Exclusive-OR (XOR) gates

Some Math Concepts (1/2)

- A polynomial of degree n, $n \geq 0$ in x can be represented as:

  $a_n x^n + a_{n-1} x^{n-1} + \ldots + a_0 \quad (a_n \neq 0)$

- Given N(x), a polynomial of degree *n*, we can show that

  $N(x) = Q(x) * D(x) + R(x)$

  where
  - Q(x), D(x) and R(x) are also polynomials of some degrees $\leq n$.
  - R(x) = 0 if N(x) is divisible by D(x).

Some Math Concepts (2/2)

- If B(x) is a polynomial of some degree, then we can show that

$$A(x) = B(x) * x^m \quad (m \geq 0)$$

where

A(x) is also a polynomial.

Modulo 2 Arithmetic

$0 + 0 = 0$

$0 + 1 = 1$

$1 + 0 = 1$

$1 + 1 = 0$

Which are the same as the XOR operations

$0 \oplus 0 = 0$

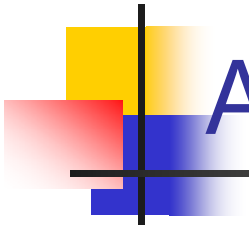
$0 \oplus 1 = 1$

$1 \oplus 0 = 1$

$1 \oplus 1 = 0$

Cyclic Redundancy Check Code

- Is also known as the *polynomial code*.
- Is based upon treating bit patterns as representations of polynomials with binary coefficients of 0 and 1 only.
- An *n*-bit message is regarded as the coefficient list for a polynomial in a dummy variable x, with *n* terms ranging from x^{n-1} to x^0. Such a polynomial is said to be of degree *n-1*.
- A mathematical polynomial known to both the sender and the receiver is used as the divisor for the original and received messages.
  - It is normally called the *generator polynomial*.



Algorithm (1/3)

- First, let's define some terms:

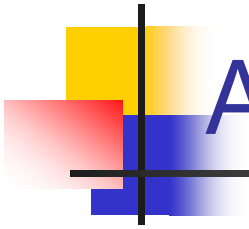
  M: the original message

  D: a predefined divisor

  Q: quotient of M/D

  R: remainder of M/D

  C: actual bit pattern transmitted



Algorithm (2/3)

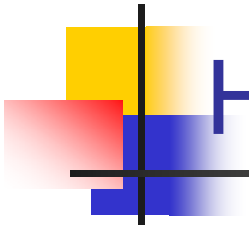
- At the transmitting end
  - Step 1: Multiply M by 2^m; the same as shift M left *m* positions.
  - Step 2: Divide the product by D using modulo 2 arithmetic giving Q and R. Note that R is also called the *frame check sequence* (FCS).
  - Step 3: Add R to the product of M * 2^m giving C, i.e., C = M * 2^m + R.
  - Then C is the form of the message to be transmitted.

Algorithm (3/3)

- At the receiving end:
  - Step 1: Divide C by D using modulo 2 arithmetic and the remainder should be 0; error otherwise.
  - Step 2: Recover the original message by shifting C right *m* positions. It is equivalent to: $M = (C - R) / 2^m$
- To prove that C is divisible by D:

  $M * 2^m = Q * D + R$

  $C = M * 2^m + R = Q * D + R + R = Q * D$



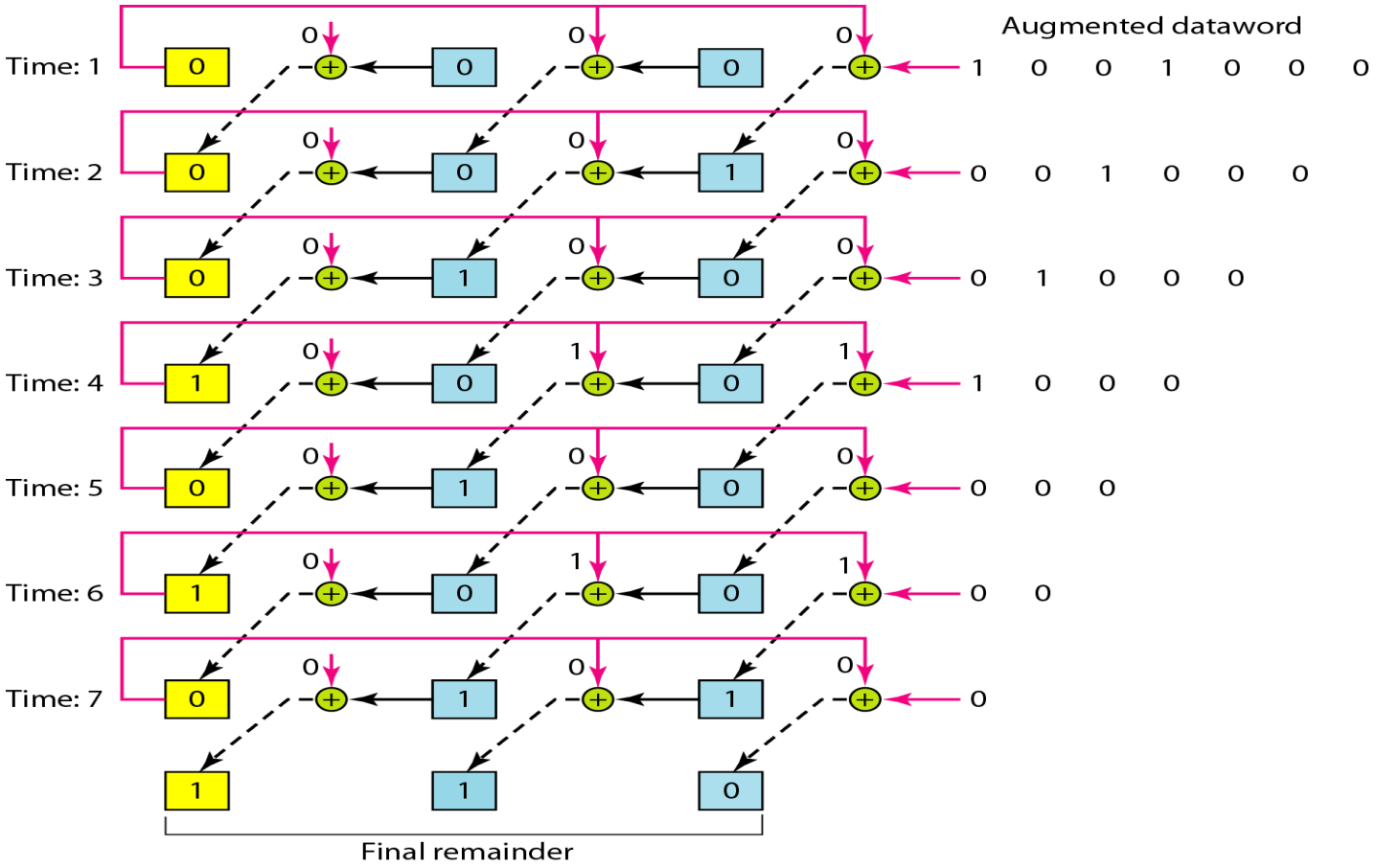
Hardware Implementation

- CRC can be implemented in hardware as a dividing circuit consisting of a shift register and some XOR gates.
  - The number of bits in the shift register corresponds to the degree of the generator polynomial.
  - An XOR gate between two adjacent register bits.
    - An XOR gate may not be needed if the corresponding coefficient value is 0.
    - The last and the first register bits share one XOR gate.

The following slide is from your textbook (by Forouzan) for illustration purposes only.

(M = 1 0 0 1 & D = 1 0 1 1)

Figure 10.18 *Simulation of division in CRC encoder*

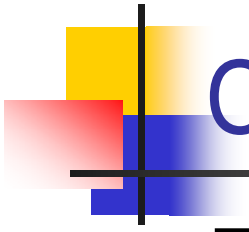


Good Generator Polynomial Characteristics

- A good generator polynomial should have the following characteristics.
  - It should have at least two terms.
  - The coefficient of the term x^0 should be 1.
  - It should not divide $x^t + 1$, for t between n - 1 and 2.
  - It should have the factor x + 1.

Widely Used CRC Polynomials

- There are several widely used polynomials for CRC purposes:
  - CRC-8 = $x^8 + x^2 + x + 1$ (for ATM Header)
  - CRC-10 = $x^{10} + x^9 + x^5 + x^4 + x^2 + 1$ (for ATM AAL)
  - CRC-16 = $x^{16} + x^{12} + x^5 + 1$ (for HDLC)
  - CRC-32 = $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ (for LANs)



CRC Notes

- Today CRC is the most common method of error detection used in data communications.
- It has been shown that CRC is a very robust method. Giving CRC-16 as an example, it catches all single and double errors, all errors with an odd number of bits, all burst errors of length 16 or less, 99.997% of 17-bit error bursts, and 99.998% of 18-bit and longer bursts.
- The computations involved in this method may seem complicated. But in practice, they can be handled by hardware very efficiently using only one simple shift register with some XOR gates. As a matter of fact, this hardware is nearly always used.