Polynomial Codes

- They are also known as CRC codes
  - Check bits are generated in the form of a Cyclic Redundancy Check
  - Implemented using the shift-register circuit

- The k information bits ($i_{k-1}, i_{k-2}, \ldots, i_1, i_0$) are used as binary coefficients to form the information polynomial of degree ($k-1$):

  $i(x) = i_{k-1}x^{(k-1)} + i_{k-2}x^{(k-2)} + \ldots + i_1x + i_0$

- The polynomial code uses binary polynomial arithmetic to calculate the codeword corresponding to the information polynomial

Binary Polynomial Arithmetic

Addition: $(x^7 + x^6 + 1) + (x^6 + x^5) = x^7 + (1 + 1)x^6 + x^5 + 1 = x^7 + x^5 + 1$

Multiplication: $(x + 1)(x^2 + x + 1) = x^3 + x^2 + x + x^2 + x + 1 = x^3 + 1$

Division:

$$
\begin{array}{rl}
 & x^3 + x^2 + x \quad = q(x) \quad \text{quotient} \\
\cline{2-2}
x^3 + x + 1 \;) & x^6 + x^5 \quad \leftarrow \text{dividend} \\
 & x^6 + x^4 + x^3 \\
\cline{2-2}
 & x^5 + x^4 + x^3 \\
 & x^5 + x^3 + x^2 \\
\cline{2-2}
 & x^4 + x^2 \\
 & x^4 + x^2 + x \\
\cline{2-2}
 & x \quad = r(x) \quad \text{remainder}
\end{array}
$$

divisor

Polynomial Encoding

- k information bits define the information polynomial of degree (k – 1)

$$i(x) = i_{k-1}x^{(k-1)} + i_{k-2}x^{(k-2)} + \dots + i_2x^2 + i_1x + i_0$$

- A CRC code is specified by its generator polynomial of degree (n – k) to generate (n – k) check bits

$$g(x) = x^{(n-k)} + g_{n-k-1}x^{(n-k-1)} + \dots + g_2x^2 + g_1x + 1$$

- $x^{(n-k)}\, i(x)$ is the dividend polynomial
- Find the remainder polynomial r(x) of at most degree (n – k – 1)

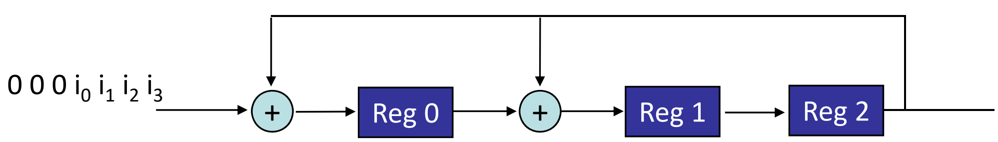
$$x^{(n-k)}\, i(x) = q(x)\, g(x) + r(x)$$

- Get the codeword polynomial of degree (n – 1)

$$\boxed{b(x) = x^{(n-k)}\, i(x) + r(x)}$$

Shift-Register Circuit Implementation

$g(x) = x^3 + x + 1$



Clock	Input	Reg 0	Reg 1	Reg 2
0	-	0	0	0
1	$1 = i_3$	1	0	0
2	$1 = i_2$	1	1	0
3	$0 = i_1$	0	1	1
4	$0 = i_0$	1	1	1
5	0	1	0	1
6	0	1	0	0
7	0	0	1	0

Check bits: $r_0 = 0$ $r_1 = 1$ $r_2 = 0$

$\Longrightarrow r(x) = x$

The Pattern in Polynomial Code

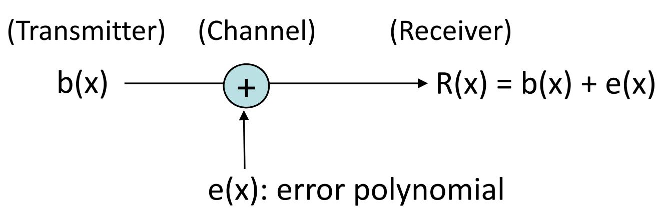
- All codeword polynomials satisfy the following pattern:

  $b(x) = x^{(n-k)}\, i(x) + r(x) = q(x)g(x) + r(x) + r(x) = q(x)g(x)$

  In other words, all codeword polynomials are multiples of g(x)!

- Receiver should
  - Convert the received n-bit block into a degree-(n-1) dividend polynomial
  - Divide the dividend polynomial by g(x)
  - Check whether the remainder polynomial is zero
  - If the remainder polynomial is non-zero, then the received n-bit block is not a valid codeword $\rightarrow$ error detected

Undetectable Errors



- e(x) has “1” coefficients in error locations & “0” coefficients elsewhere

- If e(x) is a multiple of g(x), then:

  $R(x) = b(x) + e(x) = q(x)g(x) + q'(x)g(x) = [q(x) + q'(x)]\, g(x)$

  $\Rightarrow$ If a non-zero error polynomial is divisible by g(x), then the corresponding error is undetectable

Error Detection Capabilities

- For Error Bursts of Length L:
  - For error burst starting at bit location i and ending at bit location (i + L - 1)
    - $e(x) = x^{i+L-1} + \dots\dots + x^i = x^i\, d(x)$ where $d(x) = x^{L-1} + \dots\dots + 1$

L

0000**1**???... ... ... ?????**1**000000

d(x)

  - g(x) has degree (n – k)
    - L < (n – k + 1)
      - g(x) cannot divide d(x) because deg(d(x)) < deg(g(x))
      - Can detect all such error bursts
    - L = (n – k + 1)
      - d(x) is divisible by g(x) if and only if d(x) = g(x)
      - Fraction of such error bursts that are undetectable is $(½)^{(n-k-1)}$
    - L > (n – k + 1)
      - Fraction of such error bursts that are undetectable is $(½)^{(n-k)}$

Standard Generator Polynomials

Name	Polynomial	Used in
CRC-8	$x^8 + x^2 + x + 1$	ATM header
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x + 1$	ATM AAL CRC
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$ $= (x + 1)(x^{11} + x^2 + 1)$	Bisync
CRC-16	$x^{16} + x^{15} + x^2 + 1$ $= (x + 1)(x^{15} + x + 1)$	Bisync
CCITT-16	$x^{16} + x^{12} + x^5 + 1$	HDLC, XMODEM, V.41
CCITT-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	IEEE 802, DoD, V.42, AAL5