CRC : detecting errors

CRC uses a generator polynomial

LAN uses CRC-32

Q1. Transmit the message 1111111 using CRC polynomial $x^4 + x^2 + 1$.

(a) determine the message that should be transmitted by the sender

(b) what is the result of the reciever's CRC calculation?

(c) How does the reciever know if an error occured?

generator polynomial $= x^4 + x^2 + 1$

$$= \begin{array}{ccccc} x^4 & x^3 & x^2 & x^1 & x^0 \\ 1 & 0 & 1 & 0 & 1 \end{array}$$

(a) From the <u>SENDER side</u>,

Since the degree of the generator polynomial = 4

→ add 4 redundant bits of 0's at the end of the message.

∴ message = 11111110000

Now we do polynomial long division of the message by the generator polynomial

```
           1100001
10101 ) 11111110000
      ⊕ 10101
          10101
        ⊕ 10101
        00000 10000
            ⊕ 10101
              00101
```

we perform a XOR i.e ⊕ operation at each step

We get the remainder as 0101

→ The data sender will transmit will be

```
  11111110000   data + redundant bits
⊕        0101   remainder
  -----------
  11111110101
```

The message being transmitted = 11111110101

(b) The <u>RECIEVER side</u>.

There could be two scenario's at the reciever end

successful transmission (no error)

unsuccessful transmission (error induced)

CASE 1 : No error in transmission

Receiver receives the data = 11111110101

Receiver performs polynomial long division on the message.

```
          1100001
       ____________
10101 ) 11111110101
        10101
        -----
         10101
         10101
        -----------
         00000010101
               10101
               -----
               00000
```

Since there was no error in transmission, error = 0

CASE 2: Induced error in transmission

Let us induce an error in the message i.e flip the leftmost bit

∴ the message received at receiver end after transmission (with induced error) = 01111110101

Receiver performs long division on the message.

```
          0110000
       ____________
10101 ) 01111110101
         10101
         -----
          10101
          10101
         ----------
         0000000101
```

Remainder = 0101

Since the remainder ≠ 0, the receiver knows that there was some error in transmission

(c) If the remainder on polynomial long division = 0, then there was no error in transmission

if remainder ≠ 0, then there was errors in transmission

∴ ERROR is DETECTED

Q2. A bit stream 10011101 is transmitted using the standard CRC method. The generator polynomial is $x^3 + 1$.

(a) Show the actual bit string transmitted

(b) Suppose the 3rd bit from the left is inverted during transmission, show how this error is detected at the reciever end.

The generator polynomial $= x^3 + 1$

$$= \begin{array}{cccc} 1 & 0 & 0 & 1 \\ x^3 & x^2 & x^1 & x^0 \end{array}$$

(a) SENDER side

degree of generator polynomial = 3

message after adding redundant bits = 10011101000

Polynomial long division,

```
1001 ) 10011101000 ( 100011
       1001
       ----------
       00001101
           1001
           ------
            1000
            1001
           ------
            000100
```

Remainder = 100

Now

```
  10011101000
      ⊕   100
  -----------
  10011101100
```

∴ the message being transmitted = 10011101100

(b) RECIEVER side

inducing error in the 3rd bit from left

we get message = 10111101100

Reciever recieves this message after transmission

Reciever performs the polynomial long division,

$$
\begin{array}{r}
1001\,\overline{)\,10111101100\,}\,(10101 \\
1001 \\
\hline
1011 \\
1001 \\
\hline
1001 \\
1001 \\
\hline
0000100
\end{array}
$$

The remainder $= 100 \neq 0$

Hence, the receiver detects the error in transmission, since remainder $\neq 0$.

Q3. Let $g(x) = x^3 + x + 1$. Consider the information sequence 1001. Find the code word that will be transmitted corresponding to the given information. Suppose that the code word has a transmission error in the first bit, what does the reciever obtain when it does its error checking.

$g(x) = x^3 + x + 1$
$= 1011$

(a) Transmitted code,

degree of $g(x) = 3$

message with redundant bits $= 1001000$

Polynomial long division of message with the generator polynomial,

$$
\begin{array}{r}
1011\,\overline{)\,1001000\,}\,(101 \\
1011 \\
\hline
1000 \\
1011 \\
\hline
110
\end{array}
$$

Remainder $= 110$

Now,
$$
\begin{array}{r}
1001000 \\
\oplus\ 110 \\
\hline
1001110
\end{array}
$$

The message that will get transmitted $= 1001110$

(b) Receiver side

message with error $= 0001110$

Receiver performs polynomial long division,

$$\begin{array}{r} 1011 \overline{)\,0001110\,}(\,0001 \\ 1011 \\ \hline 101 \end{array}$$

Remainder $= 101 \neq 0$

$\therefore$ receiver identifies that there was some error in transmitting data.

Q4. For the given generator polynomial $g(x) = x^7 + x^4 + x^2$. Consider the information sequence $x^{10} + x^8 + x^6 + x^4 + 1$. Find the code word that will be transmitted corresponding to the given information sequence.

(i) Show the sender side calculation

(ii) if it is an error free transmission, show the calculation at receiver side

(iii) when the receiver receives the leftmost bit as inverted, show the calculations

(iv) how much bits of error is detected in CRC? and identify the CRC technique used in LAN.

$g(x) = x^7 + x^4 + x^2$

$$= \begin{array}{cccccccc} 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ x^7 & x^6 & x^5 & x^4 & x^3 & x^2 & x^1 & x^0 \end{array}$$

Message $= x^{10} + x^8 + x^6 + x^4 + 1$

$$= \begin{array}{ccccccccccc} 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \\ x^{10} & x^9 & x^8 & x^7 & x^6 & x^5 & x^4 & x^3 & x^2 & x^1 & x^0 \end{array}$$

(i) SENDER side,

degree of $g(x) = 7$

adding redundant bits to the message =

101010100010000000

```
                    1011100111
10010100 ) 1010101000010000000
           10010100↓↓
            11111000
            10010100↓
             11011001
             10010100↓
              10011010
              10010100↓↓↓↓
                 11100000
                 10010100
                  11101000
                  10010100↓
                   11111000
                   10010100
                    1101100
```

Remainder = 1101100

```
1010101000010000000
   ⊕        1101100
-------------------
1010101000011101100  ⇒ data transmitted
```

(ii) error free transmission

```
10010100 ) 1010101000011101100 ( 1011100111
           10010100↓↓
            11111000
            10010100↓
             11011001
             10010100↓
              10011011
              10010100↓↓↓↓
                 11111011
                 10010100↓
                  11011110
                  10010100↓
                   10010100
                   10010100
                   00000000
```

Receiver recieves the data = 1010101000111011100 (without error)

On polynomial long division, he gets remainder = 0

∴ no transmission error.

(iii) Leftmost bit inverted

message = 0010101000111011100

Receiver performs polynomial long division,

```
10010100 ) 0010101000111011100 ( 001011011
             10010100
             --------
              11110011
              10010100
              --------
               11001111
               10010100
               --------
                10110110
                10010100
                --------
                 10001011
                 10010100
                 --------
                    11111100
```

Remainder = 1111100

$\neq 0$

∴ Receiver detects that there was some error in transmission.

(iv) CRC can detect any number of bits of error in the data transmitted, using the generator polynomial.

But CRC can only detect errors, not correct them.

LAN uses CRC − 32.

HAMMING CODE : Error detection + correction technique

STANDARD technique

(7, 4) technique

7 bit message $\longrightarrow$ 4 bits of data + 3 bits of parity

Uses Right to Left
Even parity

Q1. Calculate the parity bits for the data = 1011 using standard (7,4) Technique.

I. No. of message bits = 7 bits

1	0	1	P_4	1	P_2	P_1
7	6	5	4	3	2	1

The parity bits would be : 2^2 2^1 2^0
P_4 P_2 P_1

NOTE: since $2^3 = 8 > 7$
we can't consider 2^3 as a parity bit

$\therefore$ bits 1, 2, 4 are parity bits
3, 5, 6, 7 are 4 bits of data.

P_4 2^2	P_2 2^1	P_1 2^0	
0	0	0	0
0	0	*	1
0	*	0	2
0	1	1	3
*	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Now, calculating the parity bits,

$P_1 = D_3 \oplus D_5 \oplus D_7$ (even parity)
$= 1 \oplus 1 \oplus 1 = 1$

$P_2 = D_3 \oplus D_6 \oplus D_7$
$= 1 \oplus 0 \oplus 1 = 0$

$P_4 = D_5 \oplus D_6 \oplus D_7$
$= 1 \oplus 0 \oplus 1 = 0$

Substituting values of P_1, P_2 & P_4 in the message.

1	0	1	0	1	0	1
7	6	5	4	3	2	1
D_7	D_6	D_5	P_4	D_3	P_2	P_1

Messages transmitted from sender side (with parity bits)
= 1010101

II. RECEIVER receives the message with NO TRANSMISSION errors:

Message received = 1 0 1 0 1 0 1
7 6 5 4 3 2 1

The parity bits, when message is 7 bits = 2^2 2^1 2^0
P_4 P_2 P_1

∴ from the message,

given parity bits are $P_1 = 1$
$P_2 = 0$
$P_4 = 0$

Now, calculating the parity bits,

$P_1 = D_3 \oplus D_5 \oplus D_7$
$= 1 \oplus 1 \oplus 1 = 1$

$P_2 = D_3 \oplus D_6 \oplus D_7$
$= 1 \oplus 0 \oplus 1 = 0$

$P_3 = D_5 \oplus D_6 \oplus D_7$
$= 1 \oplus 0 \oplus 1 = 0$

	P_4	P_2	P_1	
	0	0	1	← given parity
$\oplus$	0	0	1	← calculate parity
	0	0	0	

Since, remainder = 0
there was no error in transmission

III. RECIEVER receives message with ERROR in DATA BIT

Suppose D_6 has an error

∴ message received = 1 1 1 0 1 0 1
7 6 5 4 3 2 1

The parity bits, when message is 7 bits = 2^2 2^1 2^0
P_4 P_2 P_1

given parity bits from the message: $P_1 = 1$
$P_2 = 0$
$P_3 = 0$

Now calculating the parity bits,

$P_1 = D_3 \oplus D_5 \oplus D_7 = 1 \oplus 1 \oplus 1 = 1$

$P_2 = D_3 \oplus D_6 \oplus D_7 = 1 \oplus 1 \oplus 1 = 1$

$P_3 = D_5 \oplus D_6 \oplus D_7 = 1 \oplus 1 \oplus 1 = 1$

	P_4	P_2	P_1
given →	0	0	1
calculated →	1	1	1
	1	1	0

Since, remainder $\neq 0 = 110 = 6$

the error is in the 110 bit

i.e 6th bit has the error

∴ Correcting the 6th bit,

the message = 1 0 1 0 1 0 1

IV. RECIEVER recieves message with ERROR In PARITY BITS

Suppose P_1 has the error.

Message recieved = 1 0 1 0 1 0 0

7 6 5 4 3 2 1

The parity bits are : $P_1 = 0$

$P_2 = 0$

$P_4 = 0$

Now, calculating the parity bits;

$P_1 = D_3 \oplus D_5 \oplus D_7 = 1 \oplus 1 \oplus 1 = 1$

$P_2 = D_3 \oplus D_6 \oplus D_7 = 1 \oplus 0 \oplus 1 = 0$

$P_4 = D_5 \oplus D_6 \oplus D_7 = 1 \oplus 0 \oplus 1 = 0$

P_4	P_2	P_1	
0	0	0	← given parity bits
0	0	1	← calculated parity bits
0	0	1	

Since remainder $\neq 0 = 001 = 1$

This indicates that the error exits in the 1st bit

∴ After correction of the message,

message = 1 0 1 0 1 0 1

Q2. A 12-bit hamming code whose hexadecimal value is 0x E4F arrives at a receiver. What was the original value in hexadecimal? Assume that not more than 1 bit is in error. Use LEFT to RIGHT & even parity.

NOTE: Hamming code technique can detect & correct error in only 1 bit

Hamming code length = 12 bits

In hexadecimal E → 1110

4 → 0100

F → 1111

∴ message =

1	1	1	0	0	1	0	0	1	1	1	1
1	2	3	4	5	6	7	8	9	10	11	12

The parity bits = 2^3 2^2 2^1 2^0 $(2^4 = 16 > 12)$

P_8 P_4 P_2 P_1

The given parity bits : $P_1 = 1$

$P_2 = 1$

$P_4 = 0$

$P_8 = 0$

P_8	P_4	P_2	P_1	
2^3	2^2	2^1	2^0	
0	0	0	0	0
0	0	0	*	1
0	0	*	0	2
0	0	1	1	3
0	*	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
*	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
only 12 bits 1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

Now, the calculation for the parity bits

$P_1 = D_3 \oplus D_5 \oplus D_7 \oplus D_9 \oplus D_{11}$

$= 1 \oplus 0 \oplus 0 \oplus 1 \oplus 1 = 1$

$P_2 = D_3 \oplus D_6 \oplus D_7 \oplus D_{10} \oplus D_{11}$

$= 1 \oplus 1 \oplus 0 \oplus 1 \oplus 1 = 0$

$P_4 = D_5 \oplus D_6 \oplus D_7 \oplus D_{12}$

$= 0 \oplus 1 \oplus 0 + 1 = 0$

$P_8 = D_9 \oplus D_{10} \oplus D_{11} \oplus D_{12}$

$= 1 \oplus 1 \oplus 1 \oplus 1 = 0$

P_8	P_4	P_2	P_1	
2^3	2^2	2^1	2^0	
0	0	1	1	given
0	0	0	1	calculated
0	0	1	0	⇒ Hence, there is an error in 2^{nd} bit

corrected message = 1 0 1 0 0 1 0 0 1 1 1 1

original value (without parity bits) = 10101111 = 0xAF //

Q① 7 bit hamming code, 3 parity bits . 4 data bits

1) 1001101

ii) 1111111 are they valid hamming codes?

DATE / /
PAGE

default R → L and even parity

since size = 7, parity bits = 2^2 2^1 2^0

P_4 P_2 P_1

(i) 1 0 0 1 1 0 1

7 6 5 4 3 2 1

	2^2	2^1	2^0
0 =	0	0	0
1 =	0	0	1
2 =	0	1	0
3 =	0	1	1
4 =	1	0	0
5 =	1	0	1
6 =	1	1	0
7 =	1	1	1

$P_1 = D_3 \oplus D_5 \oplus D_7 = 101 = 0$

$P_2 = D_3 \oplus D_6 \oplus D_7 = 101 = 0$

$P_4 = D_5 \oplus D_6 \oplus D_7 = 001 = 1$

	P_4	P_2	P_1
given	1	0	1
calculated	0	0	1
	1	0	0

1 0 0 → 4^{th} bit error

corrected = 1000101

(ii) 1 1 1 1 1 1 1

7 6 5 4 3 2 1

$P_1 = D_3 \oplus D_5 \oplus D_7 = 1$

$P_2 = D_3 \oplus D_6 \oplus D_7 = 1$

$P_4 = D_5 \oplus D_6 \oplus D_7 = 1$

P_4	P_2	P_1	
1	1	1	given
1	1	1	calc
0	0	0	⇒ no error

Q4. 16 bit messages are transmitted using hamming code. How many check bits are needed to ensure that receiver can detect and correct all single bit errors. Show ~~that~~ the bit pattern transmitted for the message

1101001100110101

No. of data bits = 16 bits

Parity bits required is = 2^4 2^3 2^2 2^1 2^0

P_{16} P_8 P_4 P_2 P_1

No. of parity bits = 5 bits

∴ Total no. of bits in message = 16 + 5 bits = 21 bits

★ LEFT to RIGHT

$\square$	$\square$	1	$\square$	1	0	1	$\square$	0	0	1	1	0	0	1
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P_1	P_2		P_4				P_8							

$\square$	1	0	1	0	1
16	17	18	19	20	21
P_{16}					

Calculating the parity bits,

	P_{16}	P_8	P_4	P_2	P_1
	2^4	2^3	2^2	2^1	2^0
0	0	0	0	0	0
1	0	0	0	0	X
2	0	0	0	X	0
3	0	0	0	1	1
4	0	0	X	0	0
5	0	0	1	0	1
6	0	0	1	1	0
7	0	0	1	1	1
8	0	X	0	0	0
9	0	1	0	0	1
10	0	1	0	1	0
11	0	1	0	1	1
12	0	1	1	0	0
13	0	1	1	0	1
14	0	1	1	1	0
15	0	1	1	1	1
	X				

$P_1 = D_3, D_5, D_7, D_9, D_{11}, D_{13}, D_{15}, D_{17}, D_{19}, D_{21}$

$= 1, 1, 1, 0, 1, 0, 1, 1, 1, 1 = \boxed{0}$

similarly

$P_2 = D_3, D_5, D_7, D_9, D_{11}, D_{13}, D_{15}, D_{17}, D_{19}, D_{21}$

$= 1\,0\,1\,0\,1\,0\,1\,0\,1 = \boxed{1}$

$P_4 = D_5, D_6, D_7, D_{12}, D_{13}, D_{14}, D_{15}, D_{20}, D_{21}$

$= 1\,0\,1\,1\,0\,0\,1\,0\,1 = \boxed{0}$

$P_8 = D_9, D_{10}, D_{11}, D_{12}, D_{13}, D_{14}, D_{15}$

$= 0\,0\,1\,1\,0\,0\,1 = \boxed{1}$

$P_{16} = D_{17}, D_{18}, D_{19}, D_{20}, D_{21}$

$= 1\,0\,1\,0\,1 = \boxed{1}$

The message = 011 0101100110011101 01