Error Detection and Correction - 03.03.2006

Hari Krishna Vemuri[1] Debapriya Chatterjee[2]

[1]03CS1016 {hvemuri@iitkgp.ac.in}
[2]03CS1017 { dpcathell@gmail.com }

I. CRC continued..

A. Mathematics behind constructing a CRC checksum generating/checking circuit...Digital logic

The CRC process can be represented by and indeed implemented as a dividing circuit consisting of XOR gates and a shift register for $P(X) = \sum_{i=0}^{n-k} A_i X^i$ with $A_0 = A_{n-k} = 1$ The circuit is implemented as follows...

1. The register contains n-k bits equal to the length of the FCS
2. There are upto n-k XOR gates
3. The presence or absence of a gate corresponds to the presence or absence of a term in the divisor polynomial ,$P(X)$,excluding the terms 1 and X^{n-k}

The architecture of a CRC circuit is best explained by first considering an example which is illustrated in figure 2 In this example we use

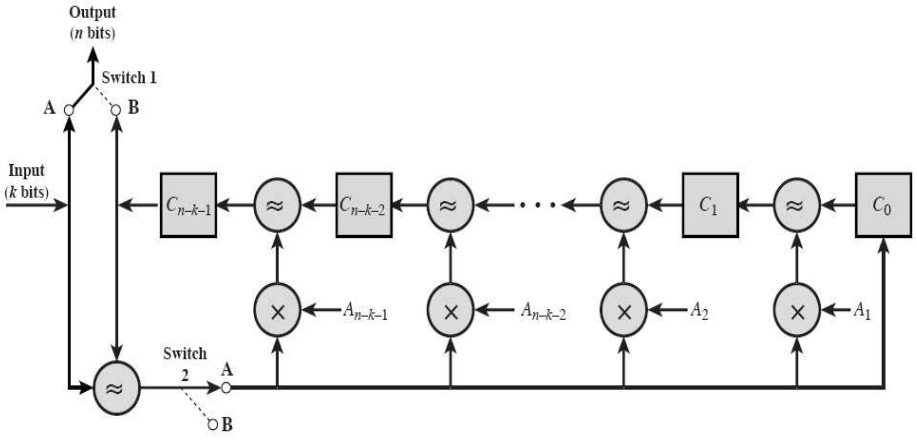
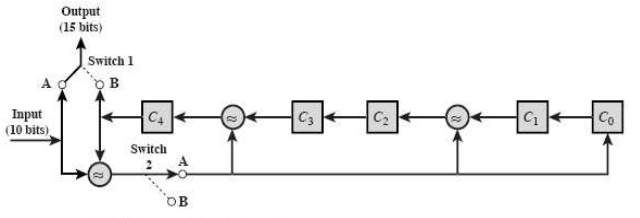


Fig. 1. General circuit for CRC checksum generation or checking: $\oplus$ means XOR $\otimes$ means AND



(a) Shift-register implementation

	C_4	C_3	C_2	C_1	C_0	$C_4 = C_3 = I$	$C_4 = C_1 = I$	$C_4 = I$	I = input
Initial	0	0	0	0	0	1	1	1	1
Step 1	1	0	1	0	1	1	1	1	0
Step 2	1	1	1	1	1	1	1	0	1
Step 3	1	1	1	1	0	0	0	1	0
Step 4	0	1	0	0	1	1	0	0	0
Step 5	1	0	0	1	0	1	0	1	0
Step 6	1	0	0	0	1	0	0	0	1
Step 7	0	0	0	1	0	1	0	1	1
Step 8	1	0	0	0	1	1	1	1	0
Step 9	1	0	1	1	1	0	1	0	1
Step 10	0	1	1	1	0				

Message to be sent

(b) Example with input of 1010001101

Fig. 2. Case study circuit for CRC checksum generation or checking: $\oplus$ means XOR

Data D=1010001101	$D(X) = X^9 + X^7 + X^3 + X^2 + 1$
Divisor P=110101	$P(X) = X^5 + X^4 + X^2 + 1$

Figure 2(a) shows the shift register implementation. The process begins with the shift register cleared (all zeros). The message,or dividend, is then entered,one bit at a time,starting with the most significant bit. Figure 2(b) shows the step by step operation as the input is applied one bit a time.Each row of the table shows the value that is currently stored in the five shift register elements.In addition the row shows the values that appear at the outputs of the three XOR circuits.Finally the row shows the value of the next input bit,which is available for the operation of the next step. Note that the XOR operation affects C_4 ,C_2 and C_0 on the next shift This is identical to the binary long division shown in figure 3.The process continues for all the bits in the message. To produce the proper output two switches are used. The input data is fed with both switches in A position.As a result,for the first 10 steps the input bits are fed into the shift register and also used as output bits. After the last data bit is processed, the shift register contains the remainder (FCS) (shown shaded). As soon as the last data bit is provided to the shift register both switches go to B position. This has two effects: (1) all the XOR gates become simple pass throughs; no bits are changed (2)as the shifting process continues the 5 CRC bits are output. At the receiver same logic is used. As each bit of M arrives, it is inserted into the shift register. If there have been no errors, the shift register should contain the bit pattern for R at the conclusion of M. The transmitted bits of R now begin to arrive , and the effect is to zero out the register.

B. Advantages of CRC

1. **Designer specification**

Think of the parity bit checking where we did not have any designer specific choice ,but here we have

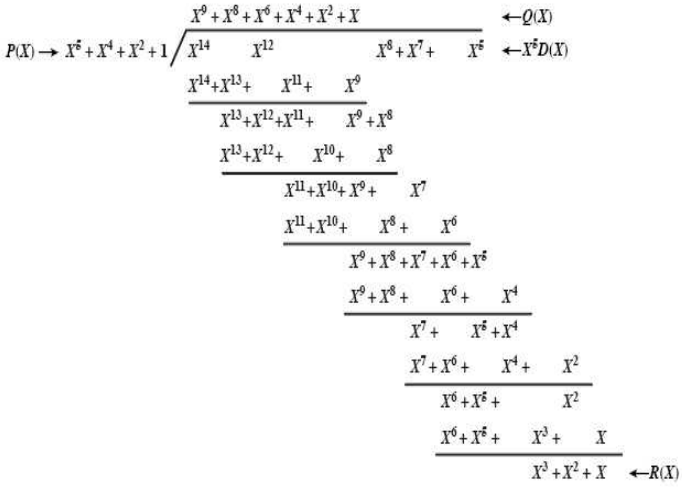


Fig. 3. The circuit actually simulates this division..give it a thought you can understand

the advantage of vast mathematical knowledge of analysis and also Linear algebra based synthesis tools so that given a certain error domain and error detection scheme we can arrive at an appropriate CRC code customized for the purpose in hand.

2. **Already proved features**

CRC has some mathematically established features which makes it ideal for use in cases with moderate error rates, to list a few of them...

(a) All single bit errors are detectable if $P(X)$ has more than one nonzero term

(b) All double bit errors are detectable as long as P(X) has a factor with 3 terms

C. Examples of CRC in use

Three of the very popular CRC codes in use..

CRC-12	$P(X) = X^{12} + X^{11} + X^3 + X^2 + X + 1$
CRC-16	$P(X) = X^{16} + X^{15} + X^2 + 1$
CRC-32	$P(X) = X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$

II. Error correction codes

A. Error correction basics

In a general error correcting model the four possiblities are

1. No error
2. Error detected and corrected
3. Error detected but not corrected
4. Error undetected

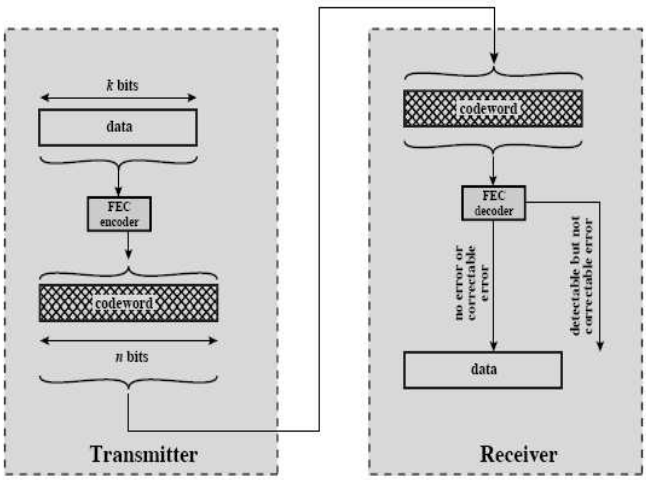


Fig. 4. Error correction process

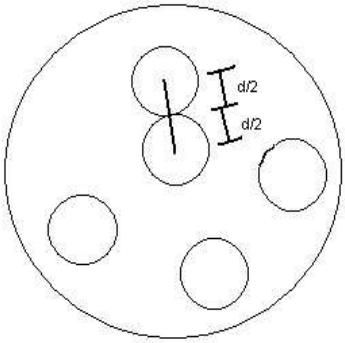


Fig. 5. 2^k valid codes in n bit codeword sphere and their hamming distance relations

Error correction is preferred in cases where retransmission of packets is best avoided.

B. Hamming codes

Block codes are examples of error correction codes, all of them are based on the idea of Hamming distance. To begin, We define Hamming distance $d(v_1, v_2)$ between two n bit binary sequences v_1 and v_2 as the number of bits where v_1 and v_2 disagree. The philosophy behind hamming codes is that given a k bit data vector v_d we use a mapping f to map it to the n bit domain to be specific a n bit code vector v_c with $n > k$, $v_c = f(v_d)$, i.e 2^k valid codewords are possible among all possible 2^n code words. Now for a code consisting of the codewords $w_1, w_2, ..., w_s$,where $s = 2^k$ the minimum hamming distance d_{min} is defined as $d_{min} = min_{i \neq j} d(w_i, w_j)$ which can also be interpreted as the distance between two valid codes in the codeword sphere of n dimensional space, this visualization quickly helps us to understand that...

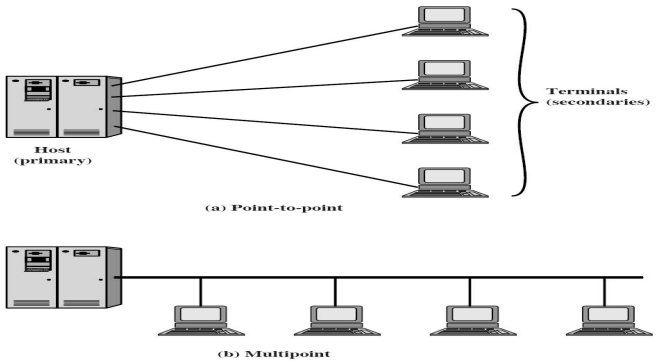


Fig. 6. Point to point topology and Multi point topology

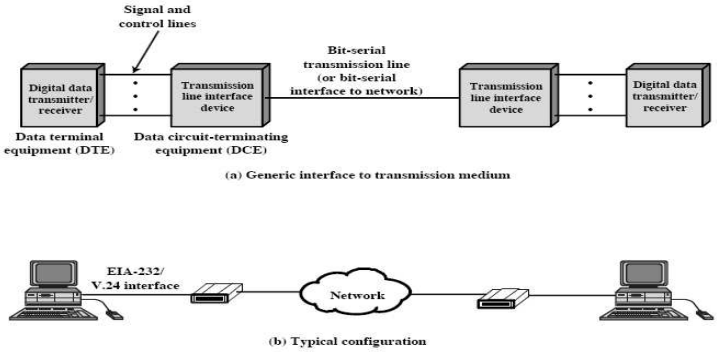


Fig. 7. Data Communications Interfacing

1. upto $t = d_{min} - 1$ bits of error can be detected
2. upto $t = \lfloor \frac{d_{min}-1}{2} \rfloor$ bits of error can be corrected

III. Line configurations

A. Topology

1. Point to point : Impractical after certain level of comlexity
2. Multipoint : Like polling

B. Duplexity

1. Full duplex : Frequency difference in wireless , twisted wire in wired
2. Half duplex : can cause confusion

IV. Interfacing

1. **DTE** : Data Transmission Equipment
2. **DCE** : Data Circuit-terminating Equipment

1. Mechanical : Male-Female connection
2. Electrical : Same code, Same voltage levels,Same duration of signals
3. Functional : Data signal / Control signal

4. Procedural : Asynchronous (Request to send , Request to clear)