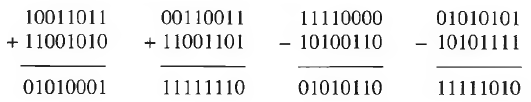
**CRC (controllo ciclico di ridondanza)**

la codifica polinomiale, nota anche come CRC (controllo ciclico di ridondanza). Le codifiche polinomiali sono basate sul fatto di trattare le sequenze di bit come dei polinomi a coefficienti che possono assumere solo i valori 0 oppure 1. Un frame di k bit è visto come una lista di coefficienti per un polinomio con k termini che variano da x^k-1 a x^0. Tale polinomio è detto di grado k-1. Il coefficiente di termine più alto (quello più a sinistra) è il coefficiente per x^k-1; il successivo è per x^k-2 e così via. Per esempio 110001 ha 6 bit e quindi rappresenta un polinomio di 5° grado con coefficienti 1, 1, 0, 0, 0 e 1 : x^5 + x^4 + x^0.

L’aritmetica dei polinomi si gestisce in modulo 2 secondo le regole della teoria dei campi algebrici. Non ci sono riporti per le addizioni o prestiti per le sottrazioni. Sia l’addizione sia la sottrazione sono identici all’OR esclusivo. Per esempio:



Le divisioni lunghe vengono eseguite come in binario, salvo che le sottrazioni sono in modulo 2 come spiegato sopra. Si dice che un divisore “sta” nel dividendo se il dividendo ha tanti bit quanti il divisore.

Quando si utilizza una codifica polinomiale, la sorgente e la destinazione devono mettersi d’accordo in anticipo su un polinomio generatore, G(x). Il generatore deve avere i bit di ordine più alto e più basso uguali a 1. Per poter calcolare il checksum di un frame di m bit che corrisponde al polinomio M(x), il frame deve essere più lungo del polinomio generatore. L’idea è quella di aggiungere un checksum alla fine del frame in modo che il polinomio rappresentato dal frame con checksum sia divisibile per G(x). Quando la destinazione riceve il frame con checksum prova a dividerlo per G(x). Se c’è un resto vuol dire che c’è stato un errore di trasmissione. L’algoritmo per calcolare il checksum è il seguente:

1. posto r il grado di G(x), aggiungere r bit con valore zero dopo la parte di ordine più basso del frame, così che adesso contenga m + r bit e corrisponda al polinomio x^r M(x)

2. dividere la sequenza di bit corrispondente a G(x) per la sequenza corrispondente a x^r M(x) usando la divisione modulo 2. [NdR - il dividendo è x^r M(x) mentre il divisore è G(x)]

3. sottrarre il resto (che contiene sempre al massimo r bit) dalla sequenza corrispondente a x^r M(x) usando la sottrazione in modulo 2. Il risultato è il frame con checksum pronto per la trasmissione. Chiamiamolo polinomio T(x).

La Figura 3.8 illustra il calcolo per il frame 1101011011 usando il generatore G(x) = x^4 + x+ 1.

Dovrebbe essere chiaro che T(x) è divisibile (modulo 2) per G(x). In ogni divisione, se si sottrae il resto dal dividendo, quello che resta è divisibile per il divisore. Vediamo un esempio: in base 10, se dividiamo 210.278 per 10.941 il resto è 2.399. Sottraendo 2.399 da 210.278, quello che rimane (207.879) è divisibile per 10.941.

Analizziamo la potenza di questo metodo. Quali tipologie di errori potrà rilevare? Immaginiamo che accada un errore in modo che al posto della sequenza di bit T(x) arrivi una sequenza T(x) + E(x). Ogni bit a 1 in E(x) corrisponde a un bit che è stato invertito. Se ci sono k bit a 1 in E(x), vuol dire che ci sono stati k errori su singoli bit. Un singolo burst di errori è caratterizzato da un 1 iniziale, una miscela di 0 e 1 e un 1 finale, mentre tutti gli altri bit sono a 0.

