Digital Data Error Techniques

طرق معالجة الأخطاء للبيانات الرقمية

- Data Error Techniques
- Error Types

1. Random Errors
2. Burst Errors

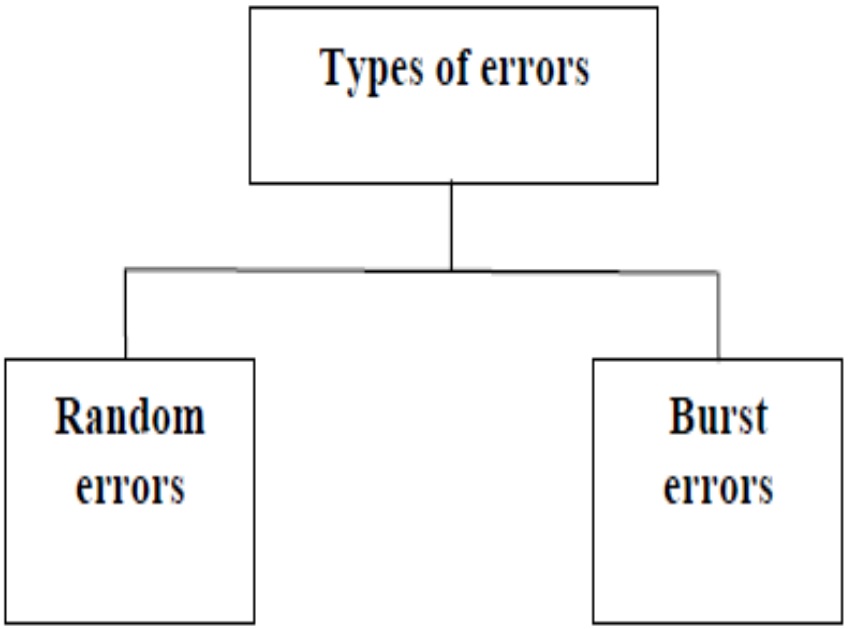
- **Error Detection**
- **VRC**
- **LRC**
- **CRC**
- **Checksum**

Error Control

- Is a set of procedures to provide reliable delivery of data between two physically connected devices.
- The reasons
  - Data can be corrupted during transmission.
  - For reliable communication, errors must be detected and corrupted.
- Error control can include:
  - Error detection:
    - Allows the receiver to detect the presence of errors
  - Error correction:
    - Allows the receiver to correct the errors.

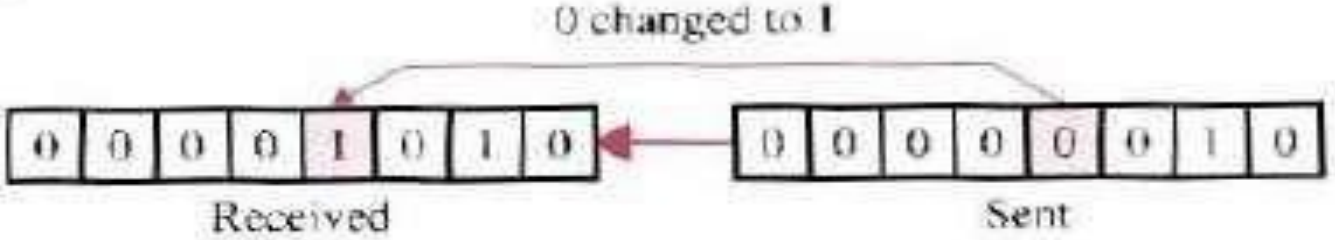
٥ -٤ -١ أنواع الأخطاء (Types of Errors)

عند تراسل الإشارة من نقطة إلي أخري فإنها تتعرض للإشارات العشوائية أو التداخلات غير المرغوب فيها نتيجة مؤثرات حرارية أو مغناطيسية أو كهربية وبالتالي تتعرض الإشارة المرسلة لأنواع مختلفة من الأخطأ التي يبينها الشكل(٥ -١٢).

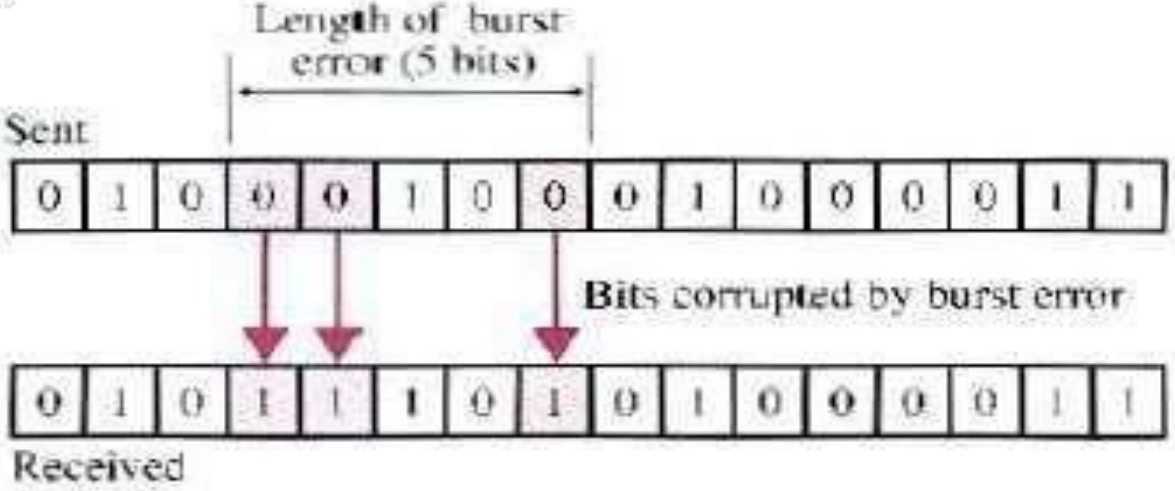


شكل ٥ -١٢

- Errors:
  - Can be caused by signal attenuation or noise.
  - Types of errors:
    - Single-bit error: only one bit of a given data unit is changed from 1 to 0 or from 0 to 1.



    - Burst-error: two or more bits in the data unit have changed from 1 to 0 or from 0 to 1.



أ - الأخطاء العشوائية (Random Errors)

حيث تظهر الأخطاء بطريقة عشوائية وفي أماكن متفرقة خلال البيانات المرسلة سواء بالطريقة المتوالية أو الطريقة المتوازية والخطاء الفردي single error يمثل أحد الأمثلة للأخطاء العشوائية كما هو مبين بالشكل.

البيانات المرسلة	0	1	1	0	0	0	1	0
البيانات المستقبلة	0	1	1	0	1	0	1	0

error

ب - الأخطاء الحزمية (Burst Errors)

حيث تظهر الأخطاء في صورة حزمة تكون فيها الأخطاء متجاورة أو متباعدة مثال ذلك حدوث خطأين أو أكثر خلال البيانات المرسلة بالطريقة المتوالية. طول حزمة الأخطاء يقاس من أول خطأ إلي آخر خطأ خلال البيانات المرسلة مع الأخذ في الاعتبار أن بعض البت الموجودة بين أول خطأ وآخر خطأ تكون صحيحة وبدون أخطاء كما هو مبين بالشكل الذي نجد فيه طول حزمة الأخطاء يساوي خمسة بت.

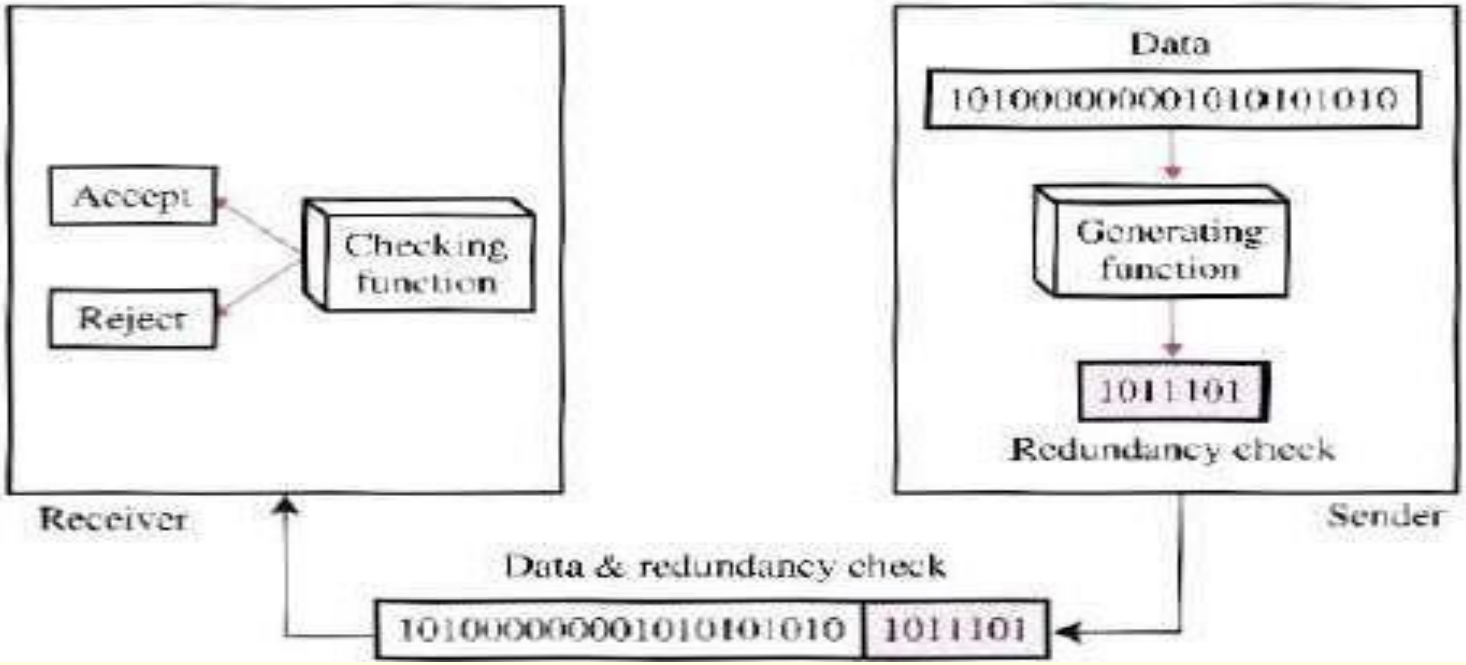
بيانات مرسلة	1	1	0	0	0	1	0	0	0	1	0	0	0	0	1	0
بيانات مستقبلة	1	1	0	0	0	1	0	1	0	1	0	0	0	1	1	0
							error						error			

لتحسين أداء قنوات التراسل يتم إتباع عدد من الأساليب لاكتشاف الأخطاء error detection أو تصحيح الأخطاء error correction خلال البيانات المرسلة وتعتمد هذه الأساليب علي إضافة عدد من البت الإضافية إلي بت البيانات الأصلية المطلوب إرسالها ويطلق علي هذه البت الإضافية (البت الزائدة redundant bits) أو (البت المكافئة parity bits) وعند نقطة الاستقبال يقوم جهاز الاستقبال باستخدام هذه البت المكافئة لاكتشاف أو تصحيح الأخطاء خلال البيانات إن وجدت كما هو مبين بالشكل(٥ -١٣). تستخدم دوائر الـ XOR عند المرسل للحصول علي هذه البت المكافئة أو عند المستقبل لفحص البيانات المستقبلة وذلك باستخدام العمليات الرياضية المنطقية التالية:

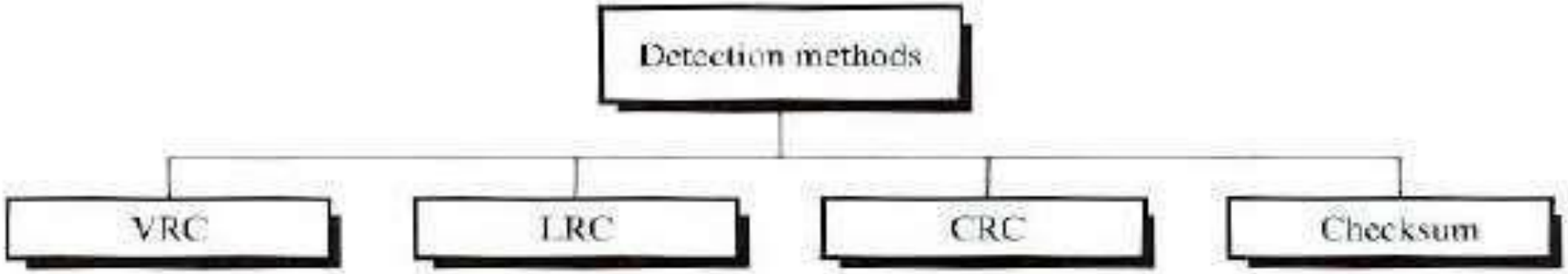
$1 + 0 = 1$	$0 + 1 = 1$	$1 + 1 = 0$	$0 + 0 = 0$

Error Detection

- Error detection uses the concept of redundancy.
  - Adds extra bits for detecting errors at destination.
    - For efficiency, extra bits $k << n$ (information bits).
    - Generating function is used to generate these extra bits.



- Four types of redundancy checks:

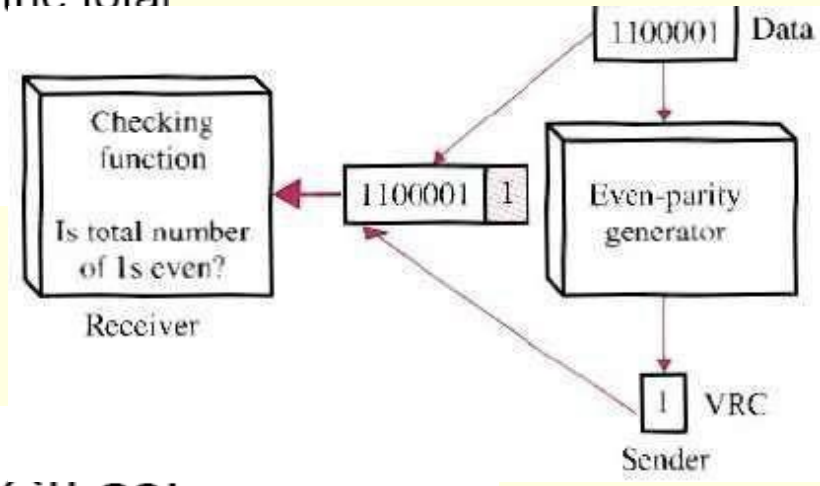


  - VRC: vertical redundancy check or parity check
  - LRC: longitudinal redundancy check
  - CRC: cyclic redundancy check
  - VRC, LRC, CRC used by data link layers.
  - Checksum used by higher-layer protocols.

Vertical redundancy check (VRC):

- Adds a parity bit to every data unit so that the total number of 1s becomes
  - even – even parity checking
  - odd – odd-parity checking

Even number of ones –add 0
Odd number of ones – add 1



Example:

- The word "cute" is coded in ASCII as:

  1100011 1110101 1110100 1100101
  c u t e

- Using even-parity checking, the sender will send:

  11000110 11101011 11101000 11001010

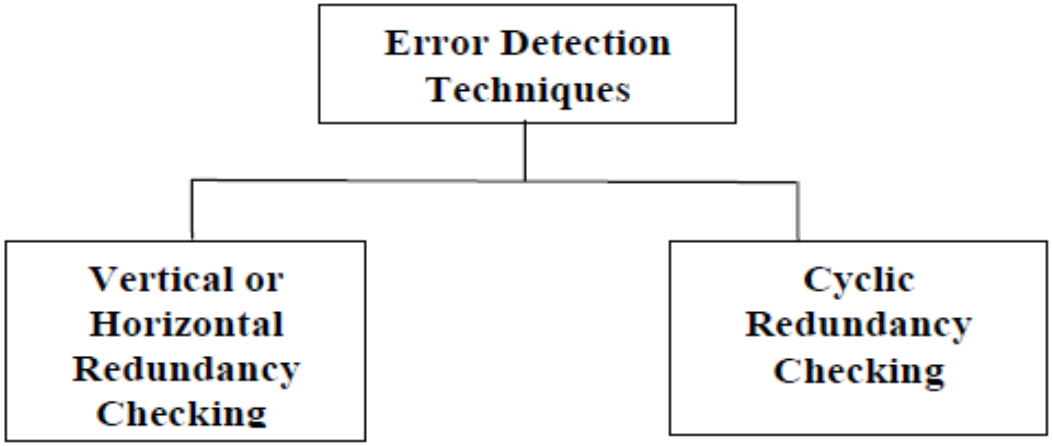
- If the word is not corrupted during transmission:
  - The receiver counts the 1s in each character and comes up with (4, 6, 4, 4) – all even numbers.
- If the word is corrupted during transmission, say:

  11010110 11101011 11101000 11000010
  - The receiver counts the 1s in each character and comes up with (5, 6, 4, 3).

Can detect all single-bit errors. Can detect burst errors only if the total number of errors in each data unit is odd.

٥ -٤ -٢ أساليب اكتشاف الأخطاء (Error Detection)

في جميع الأساليب التي تستخدم لاكتشاف الأخطأ لا يتم تحديد أماكن البت التي حدثت بها الأخطاء ولكن فقط تحديد ما إذا كانت البيانات المستقبلة بها خطاء أم لا. يمكن تصنيف معظم التقنيات المستخدمة في اكتشاف الأخطاء بنظم اتصالات البيانات والشبكات كما هو مبين بالشكل(٥ -١٤).



شكل ٥-١٤

أ - أسلوب البت المكافئة (Parity check Bit)

في هذه التقنية فإن كل حرف أو رقم ممثل بـ ٧ بت أو ٨ بت يضاف إليها بت واحدة تسمى المكافئ الفردي odd parity أو المكافئ الزوجي even parity اعتمادا علي عدد الـ 1,s التي يحتويها

الحرف أو الرقم الواحد (فردي أو زوجي) وفي كلتا الحالتين يقوم جهاز الاستقبال بإحصاء عدد الـ 1,s مرة أخري بالحرف أو الرقم المستقبل فإذا ما ظهر اختلاف بين البت المكافئة المرسلة والجديدة فهذا معناه وجود خطاء في الحرف المستقبل كما هو مبين بالجدول التالي ولا يمكن تصحيح هذا الخطاء. في هذا الأسلوب يتم اكتشاف الأخطاء الفردية فقط بينما لا يمكن اكتشاف الأخطاء الزوجية.

بت الحرف الأساسي	مضاف إليه البت المكافئ الفردي	مضاف إليه البت المكافئ الزوجي
1010111	1010111 0«	1010111 1«

ب - أسلوب البت الزائدة الطولي((Longitudinal Redundancy Check-LRC

في هذه التقنية يتم وضع مجموعة بت الحروف في مصفوفة عدد صفوفها يساوي عدد الحروف ثم بعد ذلك يتم إيجاد البت الزائدة أو المكافئة لكل عمود من أعمدة المصفوفة سواء بطريقة البت المكافئ الفردي أو البت المكافئ الزوجي فنحصل في النهاية علي صف جديد يمثل البت الزائدة المكافئة الطولية الذي يضاف إلي صفوف بت الحروف الأصلية كما هو مبين بالمثال الذي سنورده بعد ذلك. يتم إرسال البيانات صفوفاً متتالية وفي النهاية يرسل الصف الأخير الذي يمثل البت الزائدة المكافئة التي تسمى LRC.

عند نقطة الاستقبال يقوم جهاز الاستقبال بإيجاد LRC جديدة ويقارنها بالمستقبلة فإذا كان هناك اختلاف بينهما فهذا معناه وجود خطاء بالرسالة المستقبلة. هذه التقنية تتيح اكتشاف الأخطاء الفردية والزوجية ما لم تحدث هذه الأخطاء في مواضع متناظرة من المصفوفة كما إن هذه التقنية تتيح اكتشاف الأخطاء الحزمية بطول حزمة يصل إلي طول بت الـ LRC.

مثال:

البيانات الأصلية: w1= 11100111,w2= 11011101,w3= 00111001,w4= 10101001

البيانات الأصلية بعد ترتيبها في مصفوفة:

w 1= 11100111

W2= 11011101

W3= 00111001

w4= 10101001

LRC= 10101010

البيانات الأصلية + LRC:

W1	W2	W3	W4	LRC
11100111	11011101	00111001	10101001	10101010

Longitudinal redundancy check (LRC):

- 2-dimensional parity checking.
  - Divides a block of bits into rows of *n* bits.
  - Calculates the even/odd parity for each column.
    - This results in an extra row of parity bits.

Example:

- The word "cute" is coded in ASCII as:

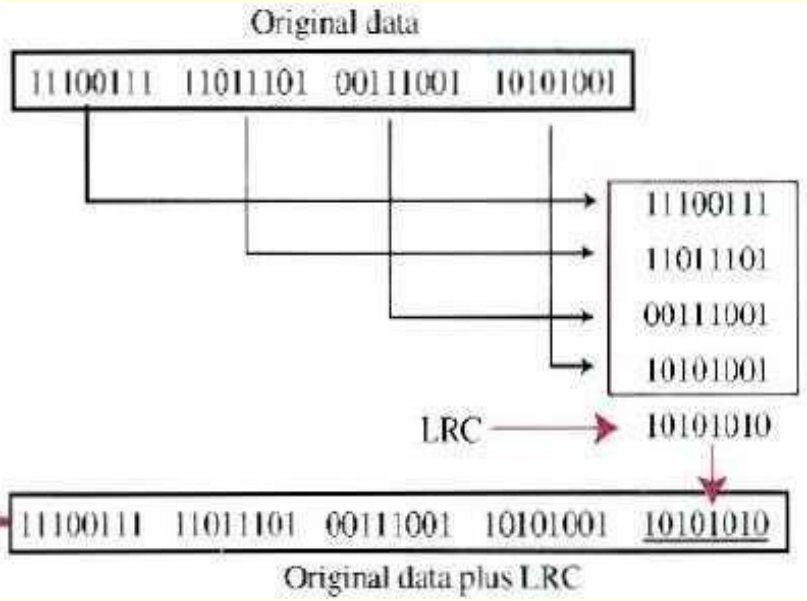
  1100011 1110101 1110100 1100101
  c u t e

- Using LRC, the sender will send:

  1100011 1110101 1110100 1100101 0000111

- If the word is corrupted during transmission, say:

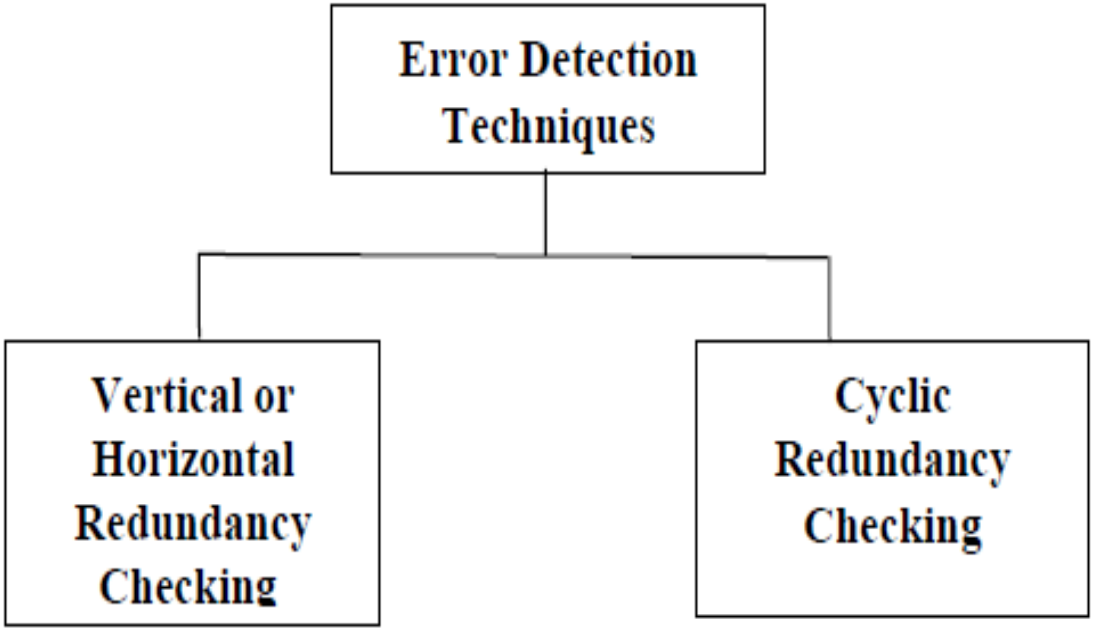
  1101101 0100101 1110100 1100001
  - The receiver calculates the LRC and and comes up with 1011001 ≠ 0000111.
  - 5 bits of LRC have changed – 5 errors have been detected.



- Increases the likelihood of detecting burst errors.
- *n* bits LRC can detect a burst error of *n* bits.
- Errors may be undetected if:
  - Have even number of errors in that position.

٥-٤-١ أساليب اكتشاف الأخطاء (Error Detection)

في جميع الأساليب التي تستخدم لاكتشاف الأخطأ لا يتم تحديد أماكن البت التي حدثت بها الأخطاء ولكن فقط تحديد ما إذا كانت البيانات المستقبلة بها خطاء أم لا. يمكن تصنيف معظم التقنيات المستخدمة في اكتشاف الأخطاء بنظم اتصالات البيانات والشبكات كما هو مبين بالشكل(٥ -١٤).

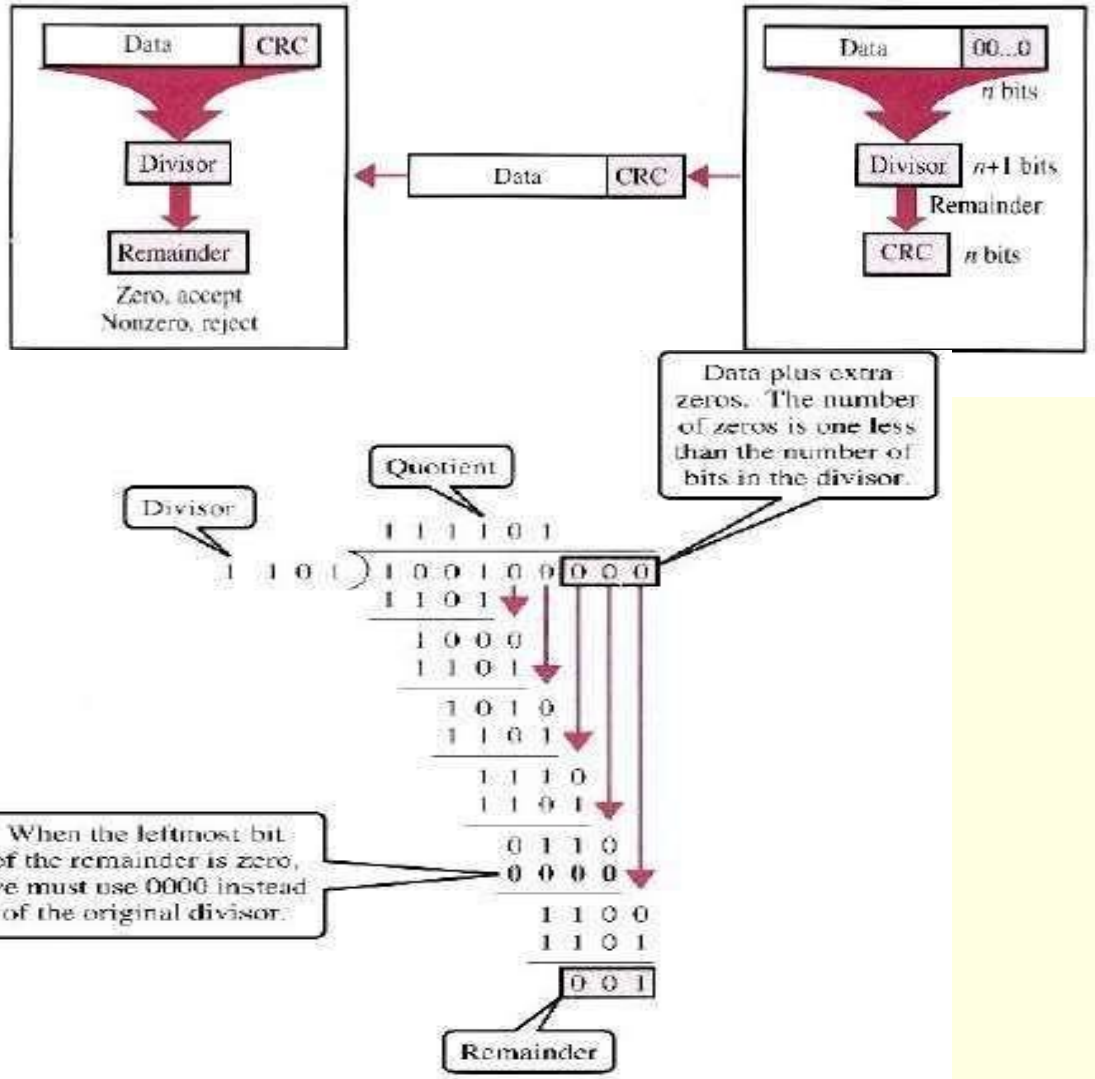


- Cyclic redundancy check (CRC):
  - Based on binary division
    - Cf. VRC and LRC based on addition
  - Divides the data unit by predetermined divisor or generator using modulo-2 division.
  - CRC = remainder.

- CRC generator:
  - For *n*-bits data unit and *m*-bits divisor:
    - Forms dividend: *n*-bits + (*m*-1) bits of zeros.
    - Divides dividend by divisor.
    - Subtracting each bit of divisor without disturbing the next higher bit.

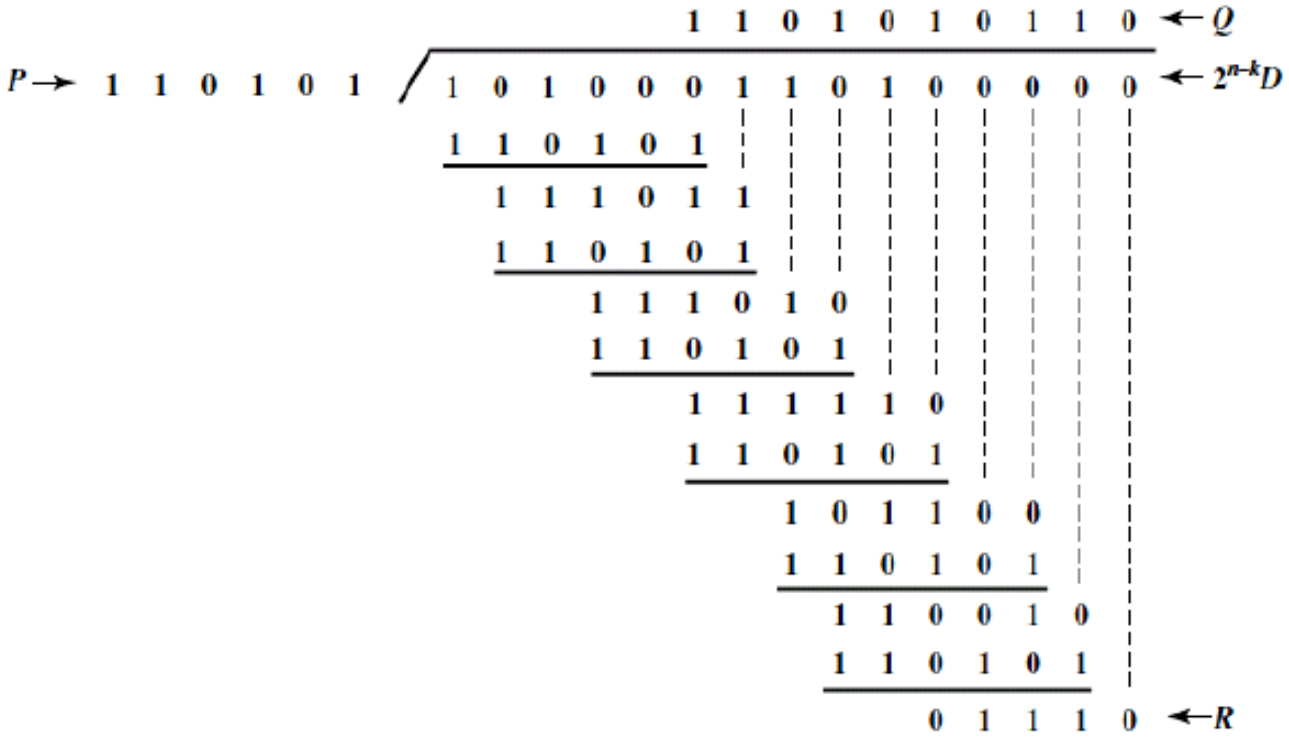
      1 – 1 or 0 – 0 = 0

      1 – 0 or 0 – 1 = 1
  - Sends data + CRC.



Example

- Pattern P = 110101 (6 bits)
- Message D = 1010001101 (10 bits)
- FCS R = to be calculated (5 bits)



ت - أسلوب البت الزائدة الدوري (Cyclic Redundancy Check-CRC)

تعتمد تقنية الـ CRC علي فكرة بسيطة مفادها أن جهاز الإرسال يضيف إلي بيانات المعلومات الأصلية مجموعة من البت الزائدة والتي تنتج بطريقة تكرارية كباق من ناتج قسمة بت بيانات المعلومات الأصلية علي عدد أولي تتم اختياره بعناية من قبل هيئات المواصفات القياسية العالمية للاتصالات. وفي جهاز الاستقبال يتم قسمة البت المستقبلة كلها علي نفس العدد الأولي الذي تم استخدامه عند المرسل فإذا كان هذا الباقي الناتج من عملية القسمة يساوي صفراً فهذا معناه عدم وجود خطاء بالبت المستقبلة أما إذا كان هذا الباقي لا يساوي صفراً فهذا معناه وجود خطاء بالبت المستقبلة وقد دلت التجارب والدراسات أنه باستخدام هذه التقنية فإن حوالي ٩٩,٩٥٪ من كل الأخطاء الناتجة خلال قنوات التراسل يمكن اكتشافها.

نظرا لصعوبة متابعة الأعداد الثنائية وقراءتها فقد جرى الاصطلاح علي تمثيل تلك الأعداد بسلسلة متعددة الحدود polynomial بحيث يمثل الأس لكل حد موقع البت المقابل للعدد الثنائي ويمثل معامل الحد قيمة البت 0 أو 1 فعلى سبيل المثال:

البيانات في صورة أرقام ثنائية - D - 1 0 1 0 0 1 1 1
البيانات في صورة متسلسلة كثيرة الحدود $D(X) - X^7 + X^5 + X^2 + X + 1$

كما إنه يتم استخدام الأعداد الأولية القياسية من قبل هيئات المواصفات العالمية الدولية لنظم الاتصالات فعلى سبيل المثال:

$CRC\text{-}12 = 1100100000011 = X^{12} + X^{11} + X^8 + X + 1$
$CRC\text{-}16 = 1100000000000101 = X^{16} + X^{15} + X^2 + 1$
$CRC\text{-}ITU\text{-}T = 1000100000100001 = X^{16} + X^{12} + X^5 + 1$

المثال التالي يوضح عملية استنتاج البت الزائدة بالطريقة التكرارية.

- بيانات المعلومات: $D = 10110111 \quad , D(X) = X^7 + X^5 + X^4 + X^2 + X + 1$
- العدد الأولي : $g = 110011 \quad , g(x) = X^5 + X^4 + X + 1$

الحل:

البت المكافئة في صورة متسلسلة يرمز لها بـ b(x). حيث b(x) تساوي الباقي من قسمة $X^m \times D(x)/g(x)$ حيث m هي درجة أو أعلي أس في المتسلسلة g(x) وفي نفس الوقت هي تساوي عدد البت المكافئة CRC وفي مثالنا هذا $m = 5$.

$$X^m \times D(x) = X^5 \times (X^7+X^5+X^4+X^2+X+1)$$
$$= X^{12}+X^{10}+X^9+X^7+X^6+X^5$$

ولإيجاد متسلسلة البت المكافئة نستخدم طريقة القسمة المطولة بطريقة دورية آخذين في الاعتبار طريقة الجمع أو الطرح المنطقية حتى نحصل علي باقي القسمة الذي يمثل متسلسلة البت المكافئة كما يلي.

	$X^7+X^6+X^4+X^2+X+1$
X^5+X^4+X+1	$X^{12}+X^{10}+X^9+X^7+X^6+X^5$ $X^{12}+X^{11}+X^8+X^7$
	$X^{11}+X^{10}+X^9+X^8+X^6+X^5$ $X^{11}+X^{10}+X^7+X^6$
	$X^9+X^8+X^7+X^5$ $X^9+X^8+X^5+X^4$
	X^7+X^4 $X^7+X^6+X^3+X^2$
	$X^6+X^4+X^3+X^2$ $X^6+X^5+X^2+X$
	$X^5+X^4+X^3+X$ X^5+X^4+X+1
	X^3+1 = remainder = b(x) , b = 01001 = CRC

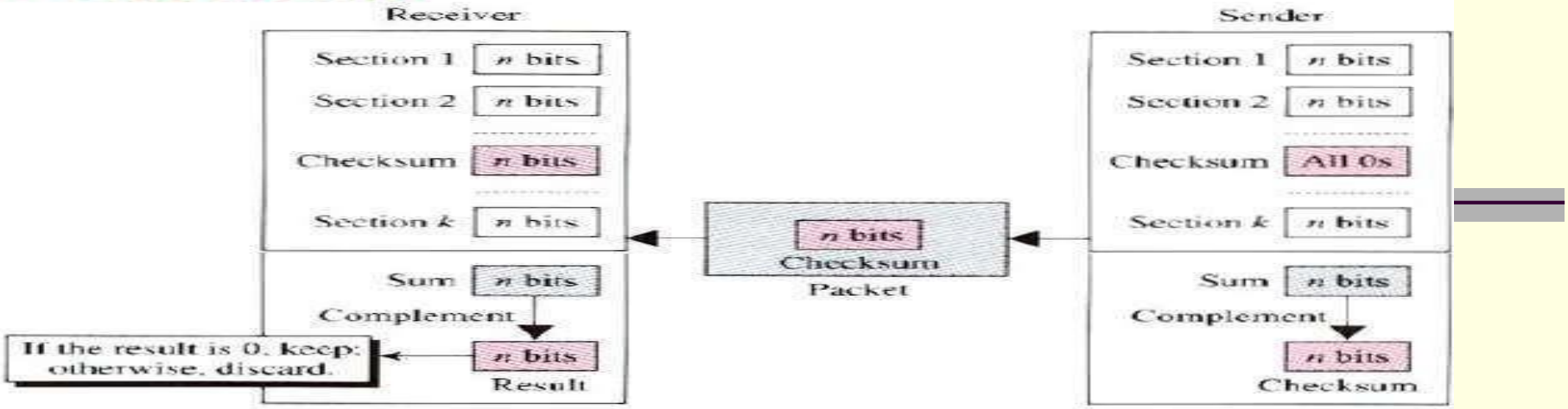
وبالتالي يمكن كتابة البيانات الكلية المرسلة في صورة متسلسلة $T(x)$ كما يلي:

$$T(x) = X^{m} \times D(x) + b(x)$$
$$= X^{12}+X^{10}+X^{9}+X^{7}+X^{6}+X^{5}+X^{3}+X$$

كما يمكن كتابة هذه المتسلسلة في صورة بيانات رقمية ثنائية T كما يلي:

$$T = 1\,0\,1\,1\,0\,1\,1\,1\,0\,1\,0\,0\,1 = \text{Data} + \text{CRC}$$

Checksum:



- Checksum generator (sender):
  - Divide data unit into segments of *n* bits (usually *n* = 16).
  - Add together all segments using one's complement to get the sum
  - Complement the sum to become the checksum.
  - Send the checksum with the data.
- Checksum checker (receiver):
  - Divide data unit into segments of *n* bits.
  - Add together all segments using one's complement to get the sum.
  - Complement the sum.
  - If the result is zero, accept the data, otherwise, reject the data.