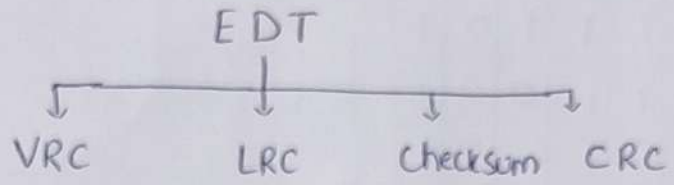
Error detection Techniques:



VRC (Vertical Redundancy Check) : It is also known as parity check.

1 1 0 0 0 0 1 → Even parity Generator → 1 → 1 | 1100001

Data — VRC — Data Txed

* It is simple technique.
* It can detect single bit error.
* It can detect burst error only if the number of error is odd.

→ Sender : 11100001 → Txn Error 10100001 → Rxr rejects this data

→ Sender : 11100001 → Txn Error 10100101 → Rxr accepts this data.

LRC (Longitudinal Redundancy Check):

* In LRC, a block of bits is organized in rows & columns.
* The parity bit is calculated for each column & sent along with the data.
* The block of parity acts as the redundant bits.

Ex: Find the LRC for data blocks 11100111, 11011101, 00111001, 10101001. &

LRC :

	1	1	1	0	0	1	1	1
	1	1	0	1	1	1	0	1
	0	0	1	1	1	0	0	1
	1	0	1	0	1	0	0	1
LRC →	1	0	1	0	1	0	1	0

* LRC increases the likelihood of detecting burst errors.
* If two bits in one data unit are damaged and two bits in exactly the same positions in another data unit are also corrupted, the LRC checker will not detect an error.

	1	1	1	0	0	1	1	1
	1	10	0	1	1	1	0	1
	0	01	1	1	1	0	0	1
	1	0	1	0	1	0	0	1
	1	0	1	0	1	0	1	0

Checksum : check the sum.

Checksum creation :

*) Break the original message into 'k' number of blocks with 'n' bits in each block.
* sum all the 'k' data blocks.
* Add the carry to the sum if any.
* Do 1's compliment to the sum = checksum.

* Consider the data unit to be transmitted is

10D1100111100010001001001000100.

Carry: 1 1 1 1 1

1	0	0	1	1	0	0	1
1	1	1	0	0	0	1	0
0	0	1	0	0	1	0	0
1	0	0	0	0	1	0	0
1 0	0	0	1	0	0	0	1	1
						1	0
	0	0	1	0	0	1	0	1
	1	1	0	1	1	0	1	0 - 1's compliment = checksum

* Collect All the data blocks including checksum.
* Sum all the data blocks and checksum.
* If the result is all 1's, accept else reject.

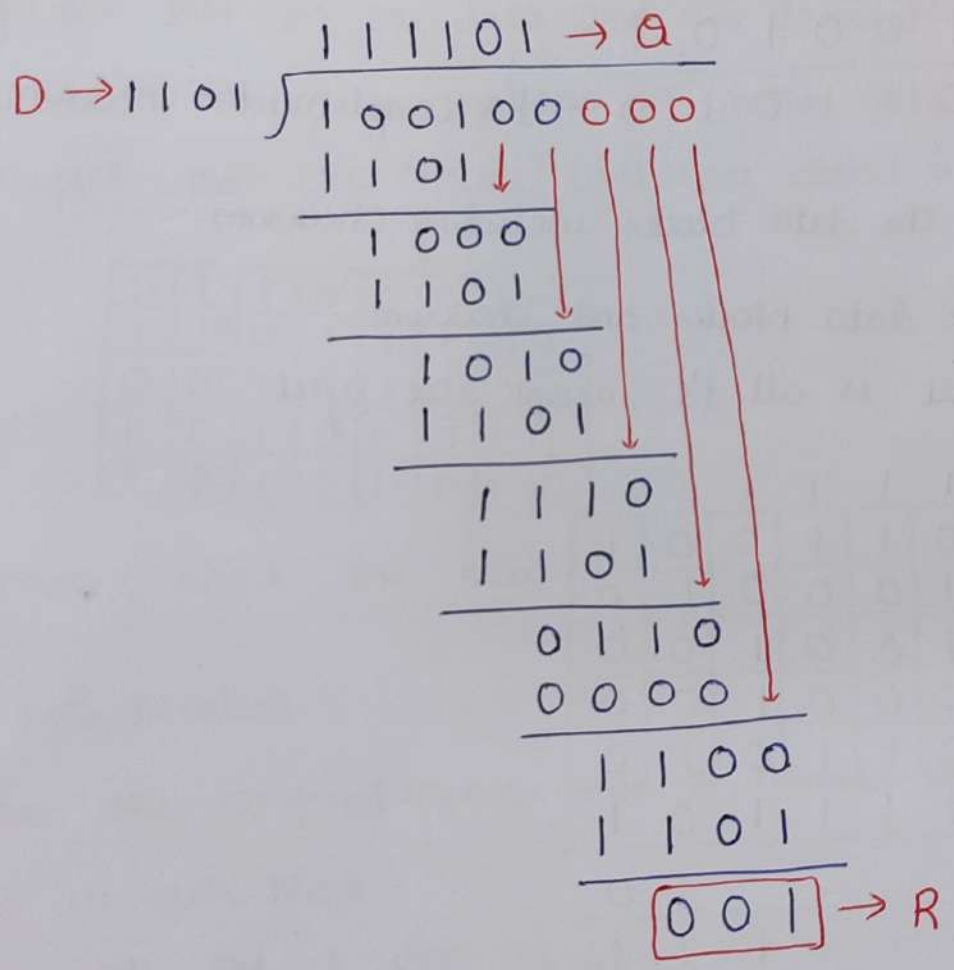
	1	1	1	1	1	1	
1	0	0	1	1	0	0	1
1	1	1	0	0	0	1	0
0	0	1	0	0	1	0	0
1	0	0	0	0	1	0	0
1	1	0	1	1	0	1	0
10	1	1	1	1	1	1	0	1
						1	0
	1	1	1	1	1	1	1	1

* The checksum detects all errors involving an odd number of bits.
* It detects most errors involving an even number of bits.

CRC [Cyclic Redundancy Check]:

Find the CRC for the data blocks 100100 with the divisor 1101

* Find the length of the divisor 'r'
* Append "r-1" bits to the original message
* Perform binary division operation [EX-OR operation]
* Remainder of the division = CRC [r-1 bits]



CRC : 001

Data Transmitted : 100100001

* Data Received : 100100001

```
        1111101
    ____________
1101 ) 100100001
       1101
       ----
        1000
        1101
        ----
         1010
         1101
         ----
          1110
          1101
          ----
           0110
           0000
           ----
            1101
            1101
            ----
             000  → Data accepted.
```

* Data Received : 100100101

```
        1111101
    ____________
1101 ) 100100101
       1101
       ----
        1000
        1101
        ----
         1010
         1101
         ----
          1111
          1101
          ----
           0100
           0000
           ----
            1001
            1101
            ----
             100  → Data Rejected.
```

* Non zero remainder indicates an error in the data.