Answer to the question no: 11

We can say that

Vulnerable bits = data rate x burst duration

a. Vulnerable bits = $1500 \times (2 \times 10^{-3}) = 3$ bits

b. Vulnerable bits = $(12 \times 10^{3}) \times (2 \times 10^{-3}) = 24$ bits

c. Vulnerable bits = $(100 \times 10^{3}) \times (2 \times 10^{-3}) = 200$ bits

d. Vulnerable bits = $(100 \times 10^{6}) \times (2 \times 10^{-3}) = 200,000$ bits

The last example shows how a noise of small duration can effect so many bits if the data rate is high.

Answer to the question no: 12

(a)
$$\begin{array}{r} 10001 \\ \oplus\, 10000 \\ \hline 00001 \end{array}$$

(b)
$$\begin{array}{r} 10001 \\ \oplus\, 10001 \\ \hline 00000 \end{array}$$

(c)
$$\begin{array}{r} 11100 \\ \oplus\, 00000 \\ \hline 11100 \end{array}$$

(d)
$$\begin{array}{r} 10011 \\ \oplus\, 11111 \\ \hline 01100 \end{array}$$

Answer to the question no: 13

The codeword for dataword 10 is 101. This code word will be changed to 010 if a 3-bits burst error occurs. This pattern is not one of the Valid Codewords, So the receiver detects the error and discards the received pattern.

Answer to the question no: 14

The codeword for dataword 10 is 10101. This codeword will be changed to 01001 if a 3 bit burst error occurs. This pattern is not one of the Valid codewords, so the receiver discards the received pattern.

Answer to the question no:15

a. $d(10000, 00000) = 1$

$$\begin{array}{r} 10000 \\ \oplus\, 00000 \\ \hline 10000 \end{array}$$

b. $d(10101, 10000) = 2$

$$\begin{array}{r} 10101 \\ \oplus\, 10000 \\ \hline 00101 \end{array}$$

c. $d(11111, 11111) = 0$

$$\begin{array}{r} 11111 \\ \oplus\, 11111 \\ \hline 00000 \end{array}$$

d. $d(000, 000) = 0$

$$\begin{array}{r} 000 \\ \oplus\, 000 \\ \hline 000 \end{array}$$

part c and d show that the distance a codeword and itself is 0.

Answer to the question no: 16

a. For error detection → $d_{min} = s+1 = 2+1 = 3$

b. For error Correction → $d_{min} = 2t+1 = 2\times2+1 = 5$

c. For error section → $d_{min} = s+1 = 3+1 = 4$

For error correction → $d_{min} = 2t+1 = 2\times2+1 = 5$

d. For error detection → $d_{min} = s+1 = 6+1 = 7$

For error correction → $d_{min} = 2t+1 = 2\times2+1 = 5$

Therefore d_{min} should be 7.

Answer to the question no: 17

a. 01

b. error

c. 00

e. error

Answer to the question no:18

we show that the exclusive-or of the second and third code word

$$\begin{array}{r} 01011 \\ \oplus\, 10111 \\ \hline 11100 \end{array}$$

is not in the code. The code is not linear.

Answer to the question no:19

I. (1st) $\oplus$ (2nd) = (2nd)

II. (2nd) $\oplus$ (3th) = (4th)

III. (3rd) $\oplus$ (4th) = (2nd)

IV. (4th) $\oplus$ (5th) = (8th)

V. (5th) $\oplus$ (6th) = (2nd)

Answer to the question no: 20

a. Dataword: 0100 → codeword: 0100011 → Corrupted: 0010011
This pattern isn't in the table → Correctly discarded

b. Dataword: 0111 → Codeword: 0111001 → Corrupted: 1111000
This pattern isn't in the table → Correctly discared

c. Dataword: 1111 → Codeword: 1111111 → Corrupted: 0101110
This pattern is in the table → Erroneously accepted as 0101

d. Dataword: 0000 → Codeword: 0000000 → Corrupted: 1101000
This pattern is in the table → Erroneously accepted as. 1101

Answer to the question no: 21

a. Dataword: 0100 → Codeword: 0100011 → Corrupted: 1100011
→ $S_2S_1S_0 = 110$ → Corrected Codeword: 0100011 →
dataword: 0100
The dataword is correctly founded.

b. Dataword: 0111 → Codeword: 0111001 → Corrupted: 0011001 → $S_2S_1S_0$: 011 → Corrected dataword: 0111001 → dataword: 0111
This dataword is correctly founded.

c. Dataword: 1111 → Codeword: 1111111 → Corrupted: 0111110 → $S_2S_1S_0$: 111 → Corrected dataword: 0101110 → dataword: 0101
This dataword is found but incorrect.

d. Dataword: 0000 → Codeword: 0000000 → Corrupted: 1100001 → $S_2S_1S_0$: 100 → Corrected dataword: 1100101 → dataword: 1100
This dataword is found but incorrect.

Answer to the question no: 22

a. If we rotate 0101100 one bit, the result is 0010110, which is in the code. If we rotate 0101100 two bits the result is 0001011, which is in the code.

b. The XORing of the two codewords

$$
\begin{array}{r}
0010110 \\
\oplus\, 1111111 \\
\hline
1101001
\end{array}
$$

which is in the code.

Answer to the question no: 23

We need to find $k = 2^m - 1 - m \geq 11$. We use trial and error to found the right answer.

a. Let $m=1$ $k = 2^m - 1 - m = 2^1 - 1 - 1 = 0$ (not acceptable)

b. Let $m=2$ $k = 2^2 - 1 - 2 = 1$ (not acceptable)

c. Let $m=3$ $k = 2^3 - 1 - 2 = 4$ (not acceptable)

d. Let $m=4$ $k = 2^4 - 1 - 4 = 11$ (acceptable)

The code is $C(15,11)$ with $d_{min} = 3$

Answer to the question no: 24

a. $(x^3+x^2+x+1)+(x^4+x^2+x+1) = x^4+x^3$

b. $(x^3+x^2+x+1)-(x^4+x^2+x+1) = x^4+x^3$

c. $(x^3+x^2) \times (x^4+x^2+x+1) = x^7+x^6+x^5+x^2$

d. $(x^3+x^2+x+1) / (x^2+1) = x+1$ (reminder is 0)

Answer to the question no: 25

a. 101110 → $x^5+x^3+x^2+x$

b. 101110 → 10111000 (Three zero's added to the right)

c. $x^3 \times (x^5+x^3+x^2+x) = x^8+x^6+x^5+x^4$

d. 101110 → 10 (The four rightmost bits are deleted)

e. $x^{-4} \times (x^5+x^3+x^2+x) = x$ (Negative power are deleted)

Answer to the question no: 26

To detect single bit errors, a CRC generator must have at least two terms and the co-efficient of x^0 must be non-zero.

a. x^3+x+1 → It meets both crities

b. x^4+x^2 → It meets the first criteria, but not the second.

c. 1 → It meets the second criteria, but not the first.

d. x^2+1 → It meets both criteria.

Answer to the question no : 27

CRC-8 generator is x^8+x^2+x+1

a. It has more than one term and the co-efficient of x^0 is 1. It can detect a Single bit error.

b. The polynomial is of degree 8, which means the number of checkbits $r=8$. It will detect all the brust errors of size 8 or less.

c. Burst error of size 9 are detected most of the time, but they slip by with probability $(1/2)^{r-1}$ or $(1/2)^{8-1} \approx 0.008$. This means 8 out of 1000 burst errors of size 9 are left undetected.

d. Burst ~~of~~ errors of size 15 are detected most of the time, but slip by with probability $(\frac{1}{2})^r$ or $(\frac{1}{2})^8 \approx 0.004$. This means 4 out of 1000 burst errors of size 15 are left undetected.

Answer to the question no: 28

This generator is $x^{32}+x^{26}+x^{23}+x^{22}+x^{16}+x^{12}+x^{11}+x^{10}+x^{8}+x^{7}+x^{5}+x^{4}+x^{2}+x+1$

a. It has more than one term and the co-efficient of x^0 is 1. It detects all single bit error.

b. The polynomial is of degree 32, which means that the number of checkbits $r=32$. It will detect all burst errors of size 32 or less.

c. Burst errors of size 33 are detected most of the time, but they slip by with probability $(\frac{1}{2})^{r-1}$ or $(\frac{1}{2})^{32-1} \approx 465 \times 10^{-12}$. This means 465 out of 10^{12} burst errors of size 33 are lefte Undetected.

d. Burst errors of size 55 are detected most of the time, but they are slipped with probability $(\frac{1}{2})^{r}$ or $(\frac{1}{2})^{32} \approx 233 \times 10^{-12}$. This means 233 out of 10^{12} burst errors of size 55 are left undetected

Answer to the question no: 29

We need to add all bits modulo-2 (XORing). However it is Simpler to count the number of 1s and make them even by adding a 0 or a 1. we have shown the parity bit in the codeword in color separate for emphasis.

	Dataword	Number of 1s	parity	Codeword
a.	1001011 →	4 (even) →	0 →	0 1001011
b.	0001100 →	2 (even) →	0 →	0 0001100
c.	1000000 →	1 (odd) →	1 →	1 1000000
d.	1110111 →	6 (even) →	0 →	0 1110111

Answer to the question no: 33

4	5	6	7	
B	A	9	8	
0	0	0	0	checksum (initial)
F	F	F	F	Sum
0	0	0	0	checksum (to Send).

Answer to the question no: 30

Datawond: 10100111

Divison : 10111

Sender

```
10111) 101001110000
       10111
       ------
       000111
        00000
       ------
         01111
         00000
         -----
          11111
          10111
          -----
           10000
           10111
           -----
            01110
            00000
            -----
             11100
             10111
             -----
             010110
             ϕϕ0111
             ------
              00001   (Reminden)
```

Codeword: 101001110001

Receiven

```
10111) 101001110001
       10111
       ------
       000111
       000000
       ------
         01111
         00000
         -----
          11111
          10111
          -----
          010000
          ϕϕ0111
          ------
          ϕ00111 0
             00000
             -----
              11100
              10111
              -----
              010111
               10111
               -----
               00000
```

Remind (no ennon)

Datawond: 10100111

Answer to the question no: 31

Sender:

Dataword: $x^7+x^5+x^2+x+1$

Divison : x^4+x^2+x+1

$$\begin{array}{r} x^7+x^4+x^3+x+1 \\ x^4+x^2+x+1 \,\big)\, x^{11}+x^9+x^6+x^5+x^4 \\ x^{11}+x^9 \qquad\qquad +x^8+x^7 \\ \hline x^6+x^5+x^4+x^8+x^7 \\ x^6+x^5+x^4+x^8 \\ \hline x^7 \\ x^7+x^5+x^4+x^3 \\ \hline x^5+x^4+x^3 \\ x^5 \qquad +x^3+x^2+x \\ \hline x^4+x^2+x \\ x^4+x^2+x+1 \\ \hline \text{Reminder } 1 \end{array}$$

Receiver:

Codeword: $x^{11}+x^9+x^6+x^5+x^4+1$

$$\begin{array}{r} x^7+x^4+x^3+x+1 \\ x^4+x^2+x+1 \,\big)\, x^{11}+x^9+x^6+x^5+x^4+1 \\ x^{11}+x^9 \qquad\qquad +x^8+x^7 \\ \hline x^6+x^5+x^4+1+x^8+x^7 \\ x^6+x^5+x^4 \qquad x^8 \\ \hline x^7+1 \\ x^7 \quad +x^5+x^4+x^3 \\ \hline x^5+x^4+x^3+1 \\ x^5 \qquad +x^3 \quad +x^2+x \\ \hline x^4+x^2+x+1 \\ x^4+x^2+x+1 \\ \hline \text{Reminder } 0 \end{array}$$

Dataword: $x^7+x^5+x^2+x+1$

Answer to the question no: 32

a. checksum at the Sender Site

1	2	2	2	Carries
3	4	5	6	
A	B	C	C	
0	2	B	C	
E	E	E	E	
0	0	0	0	checksum (initial)
D	1	C	C	Sum (partial)
			1	
D	1	C	D	Sum
2	E	3	2	checksum (to Send)

b. checksum at the receiver Site (no error)

1	2	2	2	Carries
3	4	5	6	
A	B	C	C	
0	2	B	C	
E	E	E	E	
2	E	3	2	checksum (received)
F	F	F	E	sum (partial)
			1	
F	F	F	F	Sum
0	0	0	0	checksum.

c. checksum at the receiver site (one caught error)

2 3 3 3 Carries

3 4 5 6

A B C E → Error here

0 2 B C

E E E F

2 E 3 2 checksum (received)

0 0 0 0 Sum (partial)

→ 2

0 0 0 2 Sum

F F F D checksum

d. checksum at the receiver site (two error, but not caught).

1 2 2 2 Carries.

3 4 5 6 ~~→ Error here~~

A B C E → error here

0 2 B A → error here

E E E E

2 E 3 2 checksum (received)

F F F E Sum (partial)

→ 1

F F F F Sum

0 0 0 0 checksum.