# LAB EXPERIMENT 11

**NAME: RAHIL SHARMA**

**PRN: 18070123062**

**BATCH: EA-3;G2**

**AIM:** To perform the operations of Reed Soloman error correcting.(Encoder and Decoder)

**THEORY:** Reed Solomon code is a linear cyclic systematic non-binary block code.RS codes operate on the information by dividing themessage stream into blocks of data, adding redundancy perblock depending only on the current inputs. It is capable to correct both burst errors and erasures. RS codes are generally represented as an RS (n, k), with m-bit symbols, whereBlock Length: nNo. of Original Message symbols: kNumber of Parity Digits: n -k = 2tThe relationship between the symbol size, m, and the size of the codeword n, is given by

$$n=2^m-1$$

Reed-Solomon codes are most widely used to correcting burst errors. Coding gain is very high and less then LDPC and TURBO codes.coding gainis the measure in the difference between thesignal-to-noise ratio(SNR) levels between the uncoded system and coded system needed to achieve a given Bit Error Probability.the coding rate is very high for Reed Solomon code so it is suitable for many applications including storage and transmission.

$$\text{Coding Rate}=k/n$$

Applications of Reed Soloman:

1. Storage devices (including tape, Compact Disk, DVD etc)
2. Wireless or mobile communications ( cellular telephones, microwave links, etc)
3. Satellite communications
4. Digital television

```
%Matlab Code for RS coding and decoding

n=7; k=3; % Codeword and message word lengths

m=3; % Number of bits per symbol

msg = gf([5 2 3; 0 1 7;3 6 1],m) % Three k-symbol message words
```

```
msg = GF(2^3) array. Primitive polynomial = D^3+D+1 (11 decimal)

Array elements =

   5   2   3
   0   1   7
   3   6   1
```

```
% message vector is defined over a Galois field where the number must

%range from 0 to 2^m-1

codedMessage = rsenc(msg,n,k)% Encoding the message vector using RS coding
```

```
codedMessage = GF(2^3) array. Primitive polynomial = D^3+D+1 (11 decimal)

Array elements =

   5   2   3   5   4   4   2
   0   1   7   6   6   0   7
   3   6   1   7   4   0   2
```

```
dmin=n-k+1 % display dmin
```

```
dmin = 5
```

```
t=(dmin-1)/2 % diplay error correcting capability of the code
```

```
t = 2
```

```
% Generate noise - Add 2 contiguous symbol errors with first word;

% 2 symbol errors with second word and 3 distributed symbol

% errors to last word

noise=gf([0 0 0 2 3 0 0 ;6 0 1 0 0 0 0 ;5 0 6 0 0 4 0],m)
```

```
noise = GF(2^3) array. Primitive polynomial = D^3+D+1 (11 decimal)

Array elements =

   0   0   0   2   3   0   0
   6   0   1   0   0   0   0
   5   0   6   0   0   4   0
```

```
received = noise+codedMessage
```

```
received = GF(2^3) array. Primitive polynomial = D^3+D+1 (11 decimal)

Array elements =

   5   2   3   7   7   4   2
   6   1   6   6   6   0   7
   6   6   7   7   4   4   2
```

```
%decoded contains the decoded message and cnumerr contains the number of

%symbols errors corrected for each row. Also if cnumerr(i) = -1 it indicates that the i
```

```
contains unrecoverable error
```

```
ans = logical
   0
```

```
[decoded,cnumerr] = rsdec(received,n,k)
```

```
decoded = GF(2^3) array. Primitive polynomial = D^3+D+1 (11 decimal)

Array elements =

   5   2   3
   0   1   7
   6   6   7
cnumerr = 3×1
     2
     2
    -1
```

**CONCLUSION: From this experiment we have learnt how to perform the operations of Reed Soloman Error Correcting.**