Linear Block Coding & Decoding

clc;

clear all;

n=input('Enter value of n=');

k=input('Enter value of k=');

m=n-k;

P=input('Enter parity matrix');

disp(P);

I=eye(k,k);

disp('Generator Matrix G=');

G=[I P];

disp(G);

d=0:(2^k)-1;

u=dec2bin(d,3);

disp('Display all possible messages');

disp(u);

cws=u\*G;

cw=rem(cws,2);

disp('Generated all codewords for the messages are');

disp(cw);

B=input('Enter the messages');

disp('entered message is');

disp('B');

C=B\*G;

Code\_C=rem(C,2);

disp('Generated codeword for the message is');

disp(Code\_C);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Decoding

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

clc;clear all; close all;

n=input('Enter the codeword length,n= ');

k=input('Enter the no of message length of the codeword,k= ');

P=input('Enter the coeff of parity matrix :');

r=input('Enter the received codeword');

a=P' %Transpose of parity matrix,P

I=eye(n-k) %Identity matrix of the order of (n-k)

H=cat(2,a,I) %Parity Check Matrix H

TransposeH=H'

s=r\*TransposeH %Syndrome vector,s=received codeword \* H'

l=rem(s,2)

if( l==0)

disp('no need of correction');

else

e=eye(n) %returns the n-by-n identity matrix.

el=e\*TransposeH

sl=rem(el,2)

for i=1:n

if l==sl(i,:)

break;

end

end

correctedCodeword=xor(r,e(i,:)) %correctedCodeword=r xor e

mes=correctedCodeword(1:k)

end

Additional Encoder

clc;clear all; close all;

n=input('Enter the length of the codeword, n: ');

k=input('Enter the message length of the codeword, k: ');

p=input('Enter the Parity Matrix: '); %order of Parity Matrix k X(n-k)

I=eye(k); %Identity matrix of the order of kXk

disp('Generator Marix');

%Calculate the Generator Marix by concatenating I and p, G=[Ik | P]

G=cat(2,I,p) %Concatenate arrays along specified dimension.cat(dim, A, B)

m=0:(2^k)-1 %No. of message bit =2^k

%d=dec2bin(m,k) %produces a binary representation with at least k bits

d=de2bi(m,k,'left-msb') %Convert decimal numbers to binary vectors

C=d\*G; %Codeword=Dataword \*Generator Marix

% m=input('Enter the value , m: ');

% C=m\*G

Codeword=rem(C,2) %Remainder after division