

Data Communication

Error Detection and Correction

Error correction method based on multiple parity bits, can correct one bit error. However, the receiver should make sure that the corruption has happened in only one bit. One method to do this is using an error detection and an error correction methods together.

Assume data is 8 bits. Use multiple parity to create error correction code. Then find the CRC for the codeword (data+parity) and append it to end of the codeword. At the receiver:

1. If CRC check does not fail then accept the data by separating the parity from data bits
2. If CRC fails, then assume only one bit is in error, make the correction and repeat CRC check
3. If the second CRC check fails, then ask for re-transmission of the message (print a message on the screen)
4. Otherwise, assume only one bit was in error, and the message has been corrected successfully.

Use the following sample code for creating CRC code. Note that data is given as a vector of 1s and 0s. Test your code by changing 1 or some bits.

CRC code generation in MATLAB

```
msg=[1 1 1 0 0 0 1 1 ]
% the coefficients of the generating polynomial
poly=[1 1 0 0 1 1]
[M N]=size(poly);
mseg=[msg zeros(1,N-1)];
[q r]=deconv(mseg,poly);
r=abs(r);
for i=1:length(r)
    a=r(i);
    if ( mod(a,2)== 0 )
        r(i)=0;
    else
        r(i)=1;
    end
end

crc=r(length(msg)+1:end)
%frame variable gives data concatenated by CRC
frame = bitor(mseg,r)
```