DIGITAL COMMUNICATION RECEIVER USING VERILOG

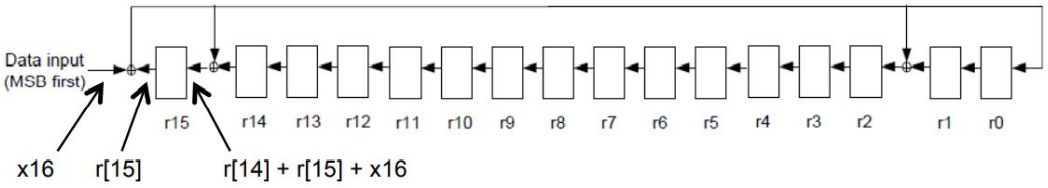
INTRODUCTION:

In the current world, communication has got many applications such as telephonic conversations in which the messages are encoded into the communication channel and then decoding it at the receiver end. Different types of errors are encountered during data transmission because of physical defects in the communication medium as well as environmental interference.Digital communication system is used to transport an information bearing signal from the source to a user destination via a communication channel. Error coding is used for error free communication in different digital data comm. Systems such as network comm., satellite and cellular comm. and deep space communication.

Error coding uses mathematical formulae to encode data bits at the source end into longer bit words called code words for transmission. In this project we will implement forward error correction which has CRC as its front and an interleaver at its tail.

When facing a tough problem, coming up with a solution is easier when the problem is broken down to simpler smaller blocks. This project is divided into 4 tasks which will be described one by one in detail in this report.

CRC: *Cyclic Redundancy Check (CRC) is used to detect errors in data transmission and is capable of detecting all single and double errors and many multiple errors with a small number of bits. Here message is interpreted as polynomial and is divided by a generator polynomial. Then the remainder of the division is added to the actual message polynomial to form a code polynomial. For our application, the generator polynomial is CRC16 (x16 + x15 + x2 + 1). We initialize the register to 16'hffff to ensure that leading 0s in front of a packet are protected by the CRC. All the registers are clocked with common clock*

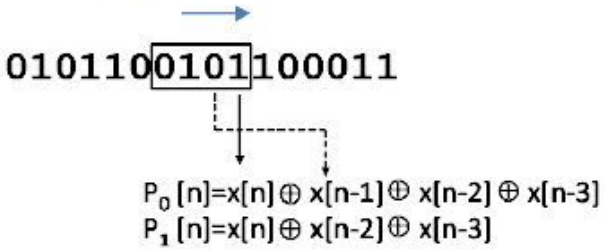


Example: input [32 bytes] ***03 01 02 03***
After 32 bits are shifted in, the value in r[15:0] is the CRC ***[30 3A]****.*

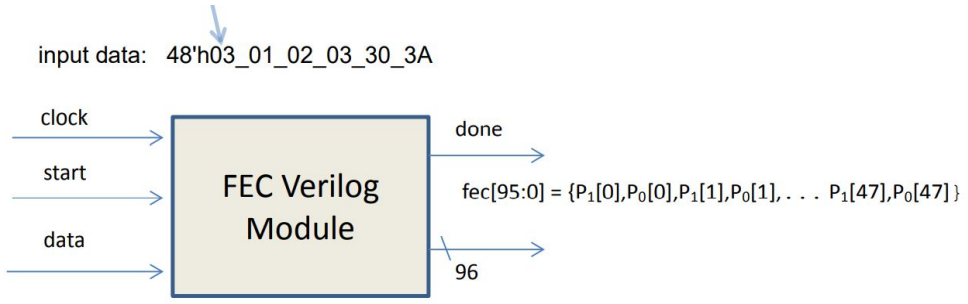
*The CRC calculation is such that sending the data and appended CRC (6 bytes) through the same hardware used to generate the CRC will give a CRC of [**00**].*

***FEC:** Forward Error Correction (FEC) code is a system of adding redundant data, or parity data, to a message, such that it can be recovered by a receiver even when a number of errors (up to the capability of the code being used) were introduced. Since the receiver does not have to ask the sender for retransmission of the data, a backchannel is not required in forward error correction, and it is therefore suitable for simplex communication such as broadcasting. FEC can be used when it is known beforehand that a received signal will be very weak and likely to contain errors resulting in a high bit error rate (BER). This could be when the signal is transmitted over long distances. One example of FEC is convolution codes . Convolution codes involve calculating parity bits and then sending only the parity bits.A convolution encoder uses a sliding window to calculate the parity bits.*
- The size of the window is the constraint k (the larger k is, the larger a particular bit is used when calculating the parity bit which inturn implies greater redundancy and better error correction possibilities)
-The code rate of a convolutional code tells you how many parity bits are sent for each message bit(1/r , r parity bits/message bits).
-We'll transmit the parity sequences, not the message itself. Each message bit is "spread across" K elements of each parity sequence, so the parity sequences are better protection against bit errors than the message sequence itself.
-In our case k and 1/r are 4 and ½ i.e., each input bit is encoded into two output bits, thus doubling the amount of data that must be transmitted.
-gi is the K-element generator for parity bit pi.
g0 = 1,1,1,1 and g1 = 1,1,0,1. The parity bits are then

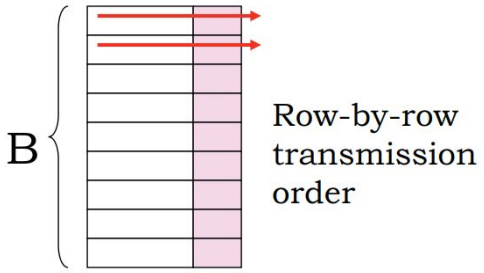
$$p_i[n] = \left(\sum_{j=0}^{k-1} g_i[j]x[n-j]\right) \bmod 2 .$$



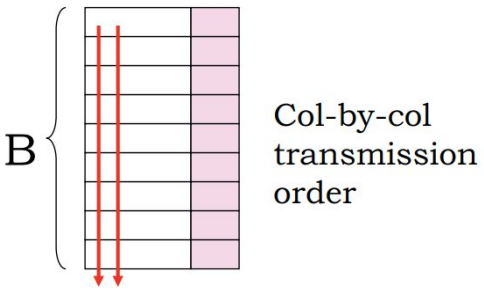
The parity bits are then sent as a single data stream:
P1[0],P0[0],P1[1],P0[1],P1[2],P0[2], ...
*Using the input data with crc appended (48'h03_01_02_03_30_3A) we generate parity bits and store the output in **fec[95:0]**. The input start pulses high for one clock cycle when data is available. When all the parity bits are generated, done is asserted with fec[95:0] containing encoded data streams. (set x[-1]= x[-2] = x[-3] = 0)*



Interleaver: *In many situations errors come in bursts: correlated multi-bit errors (e.g., fading or burst of interference on wireless channels, damage to storage media etc.) which wipes out the large number of adjacent data bits - defeating the convolution code. A simple solution is to interleave the data bits of a four byte packet so that adjacent data bits are spaced out in the transmitted sequence. Instead of sending all 8 bits of the byte 0, the low order bit pair of bytes 3, 2, 1, and 0 (starting at the LSB end) are transmitted followed by the next set of bit pairs until all bits are transmitted. This is implemented in many satellite communication systems.*

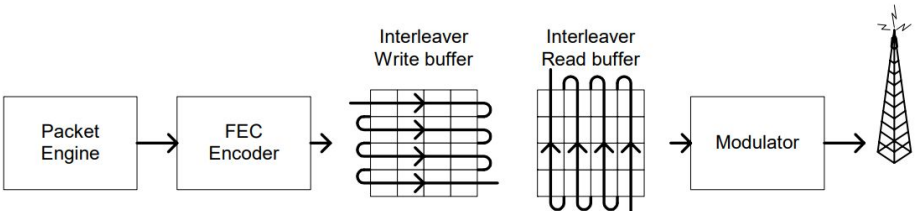


B-bit burst produces multiple bit errors

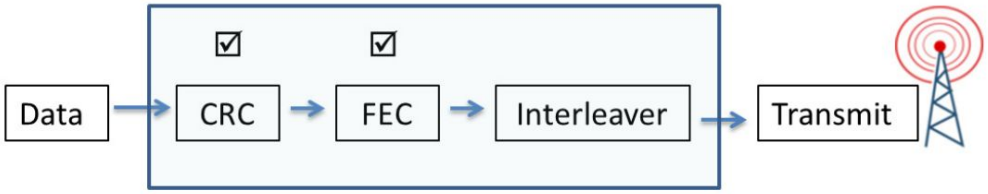


B-bit burst produces 1-bit error in B different codewords

For example, the data packet 00 0E 8C 03 ie
00 00 00 00
00 00 11 10
10 00 11 00
00 00 00 11
should be interleaved and transmitted as C8 3C 00 20 ie
11 00 10 00
00 11 11 00
00 00 00 00
00 10 00 00

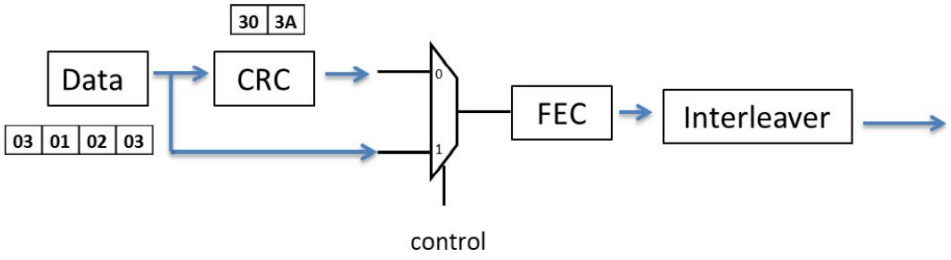


The final task is to establish a communication system with these modules.
The complete process, in serial mode, requires a 32 clock cycle to calculate the CRC followed by 48 clock cycles for the convolution encoder totaling 80 clock cycles.



0x03_01_02_03 + CRC: 30_3A

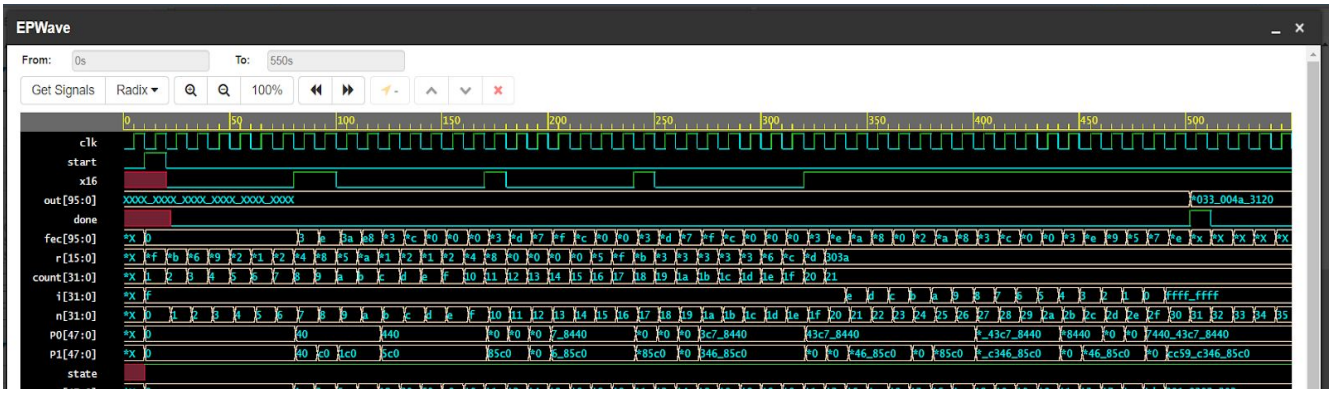
But notice that the CRC result is not required while the convolution encoder is processing the first four bytes of data. So we will send the first 4 bytes of data to both CRC and FEC parallely. After the last data bit has been shifted in, CRC has been computed and is available. At this time the input to the convolution encoder can be switched by control to the output of the CRC registers. CRC generation and convolution encoding takes 32 cycles and 16 more clock cycles to complete convolution encoding. By taking advantage of concurrent processing total processing is reduced from 80 clock cycles to 48 clock cycles.



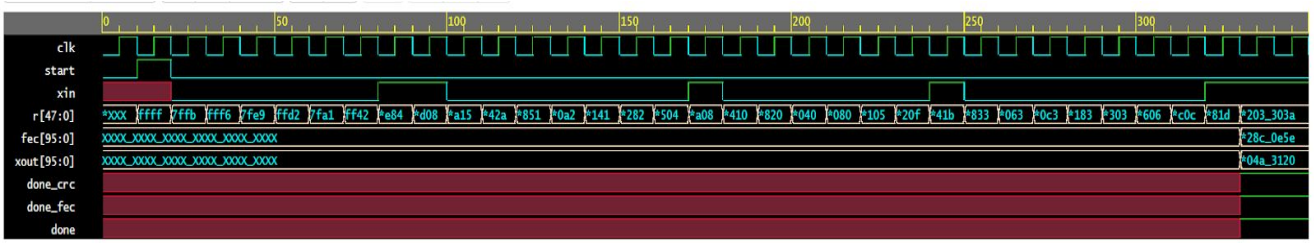
Total time can even be reduced further. At t=32, since all 16 bits of CRC are available, the remaining 32 parity bits can be computed in parallel in one clock cycle!

Result:-

48 clock cycles -



32 clock cycles -



All data in hexadecimal base:
Input: [***4*** *bytes]* ***03 01 02 03***
Appended CRC: [***6*** *bytes]* ***03 01 02 03 30 3A***
FEC encoder output: [***12*** *bytes]* ***00 0E 8C 03 7C 0D F0 0E 82 8C 0E 5E***
Interleaver output: [***12*** *bytes]* ***A2 84 C8 FC CF 3C 40 33 00 4A 31 20***

Conclusion:

Digital communication system is used to transport an information bearing signal from the source to a user destination via a communication channel.Every communication channel is exposed to different kinds of noise sources which cause some loss or changes in the data transmitted over the channel. The problem arises when you discover that the received data has been corrupted and you have to retransmit the whole data again. However retransmission is not an acceptable idea. This project will allow the receiver to detect and correct them without the need of retransmission. Deep space and satellite communications network, broadband modems, digital televisions etc., fields have the application of this project.