Cyclic Redundancy Check

Pramith, Sumanth

3/8/2019

Outline

Introduction

Implementation

Hardware(LFSR) to be implemented in verilog

Data input to FPGA

Introduction

- CRC is an Error detection technique widely used in communication protocols
- cyclic redundancy check technique: where you add parity bits to the input binary data and transmit it. After receiving the data it checks for the error
- the parity bits are generated by dividing the input binary data with selected generator polynomial

Examples

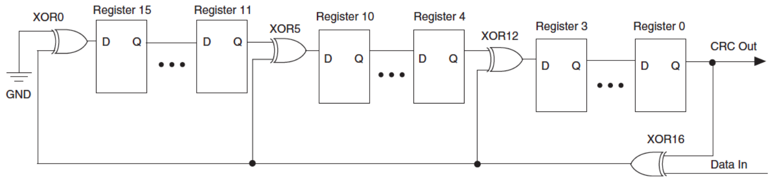
- $CRC24(X) = X^{24} + X^{23} + X^{21} + X^{20} + X^{17} + X^{15} + X^{13} + X^{12} + X^{8} + X^{4} + X^{2} + X + 1$
- $CRC16(X) = X^{16} + X^{12} + X^{5} + 1$

Implementation

- parity bits generation is done by implementing linear feedback shift register of length equal to the length of the generator polynomial on fpga
- linear feedback in lfsr is implemented at the position of non zero coefficients of the crc polynomial
- after the input binary is processed through the lfsr sequentially on fpga CRC is updated on the lfsr
- we add this crc binary bits to the end of the input data and transmit it through the channel
- after receiving the data which may be distorted by noise we process the received data through the lfsr(of generator polynomial) and check for for the CRC(nothing but remainder)
- if remainder is zero then message is not distorted otherwise distorted

Hardware(LFSR) to be implemented in verilog

Figure 1. LFSR Configuration for CCITT CRC-16 Generator ($X^{16} + X^{12} + X^{5} + X^{0}$)



Data input to FPGA

- arduino reads data from raspi initially and is sent to fpga sequentially
- output parity bits are colleted from fpga and appended to the end of the input binary signal
- same algorithm is implemented to detect the errors in the received signal where crc is zero if no error is occured during the transmission
- the error detection can be shown on an arduino serial monitor after collecting the data from the lfsr