Performance Evaluation & Design Methodologies for Automated 32 Bit CRC Checking for 32 bit address Using HDLC Block

Neeraj Kumar Misra,
(Assistant professor, Dept. of ECE, R D Foundation Group of Institution Ghaziabad, India
Email: neeraj.mishra3@gmail.com)

ABSTRACT
For design methodology of CRC or cyclic redundancy check is very used technique for error checking and shows the transmission reliability we are using the HDLC block. HDLC block is very useful in data communication these block operated in data link layer. For design methodology of CRC is to generate the CRC polynomial using XOR’s gate and shift register these polynomial are implement on software Xilinx Plan Ahead 13.1 and verify for simulation result for random testing of CRC bit on receiver side same result are obtained to show that it is more reliable.

Keywords – CRC, XILINX, HDLC, LFSR, OSI

I. INTRODUCTION
In OSI model first layer is physical layer. This layer takes care of getting data on the wire and off of it again. At the data link layer, we must take this incoming stream of data from higher or lower layers and create frames from it. Handling the data requires a solid protocol that can perform better error checking and more efficient throughputs. That data is performed by HDLC block. Basically ISO termed it as High level Data Link Control (HDLC). HDLC has own standard frame format. In the framing of HDLC one block is CRC [Cyclic Redundancy Checking] it is basically error checking number that the Destination can use to verify that the packet is error free. This is usually done by the data link protocol and calculated CRC is appended to the end of the data link layer frame.

II. LITERATURE SURVEY
Main focus of the survey is to understand the data link layer and develop a protocol which can offer its services to the layer above it i.e. is the network layer and the layer below it i.e. the physical layer. Function of this protocol controller is to perform a number of separate activities like physical addressing, to check for errors, flow control etc.

III. COMPETITIVE OVERVIEW OF CRC AND HDLC BLOCK
In HDLC overview HDLC device contains a full-duplex transceiver with independent transmit and receive sections for bit-level HDLC protocol operations. The core is designed for easy integration into wide range of applications implemented on most ASIC’s and FPGA’s technologies. Now the controllers generate and detect flags that indicate the HDLC status. They provide 32-bit CRC on data packets using defined polynomial, and recognize the single byte address in the received frame.
In transmitting and receiving section of HDLC the packet data are serially, while providing the data transparency through zero insertion and deletion. These controllers generate and detect flags that indicate the HDLC status. They provide 32-bit CRC on data packets using the CRC defined polynomial, and recognize the single byte address in the received frame.

**CRC 32bit Polynomial** [32 bits] = \( X^{32} + X^{26} + X^{23} + X^{16} + X^{12} + X^{11} + X^{10} + X^{8} + X^{7} + X^{5} + X^{4} + X^{2} + X + 1 \) HEX CODE 04C11DB7

In CRC overview it is powerful method for detecting errors in the received data is by grouping the bytes of data into a block and calculating a Cyclic Redundancy Check (CRC). This is usually done by the data link protocol and calculated CRC is appended to the end of the data link layer frame. The CRC is calculated by performing a modulo 2 division of the data by a generator polynomial and recording the remainder after division.

**IV. SOFTWARE APPROACH INTENTION OF MODULAR DIVISION OF CRC**

Division may be performed in software, it usually performed using a shift register and X-OR gates. The hardware solution for implementing a CRC is much simpler than a software approach. For example for a CRC-32bit is:

Depending upon the data of the MSB register in the Linear Feedback shift register (LFSR), shifting and XOR operations occur. This serial LFSR implementation is converted into a one shot or single cycle operation. Now this approach is converted into polynomial that is CRC polynomial. Suppose polynomial \( x^3+x+1 \) is generated the generated circuit for CRC is shown fig e.

**V. DESIGN METHODOLOGIES OF CRC USING HDLC BLOCK**

A practical implementation of a decoder also requires a method to initialize the encoder prior to transmission of the first bit of data in a frame, and to flush the encoder after sending the last byte. In the example below (which uses a different representation of the schematics for X-OR gates and shift register elements), the process starts by initializing the encoder with zero bits, by setting the switch to B. Some CRC’s initialize the register to a non-
zero value, which can give added detection capability when the first set of bits in a frame may themselves be zero. Then the switch is moved to position A and one data bit enter the encoder for each clock cycle. The data bits are immediately available at the output. After the last bit has been sent, the switch is returned to position B and the contents of the encoder are sent to the output. This is often called flushing the encoder and requires one clock cycle per bit held in the shift register.

On reception, the process is reversed. The CRC register is first set to zero (or the initial value on transmission, if non-zero). The bits (this time including the CRC) are fed into the register on each clock cycle. If the CRC contains the value zero (assuming initialization was zero), the CRC is valid, if not it has detected an error. The CRC-32 is able to detect all single errors, all double errors, all odd numbers of errors and all errors with burst less than 32 bits in length. In addition 99.9984 % of other error patterns will be detected. Protocols at the network layer and higher (e.g. IP, UDP, TCP) usually use a simpler checksum to verify that the data being transported has not been corrupted by the processing performed by the nodes in the network.

After the packet with the polynomial the remainder comes out to be zero that means the transmission and reception are error free and in case the remainder is not zero that means an error has occurred during the process and hence the packet is discarded and the whole packet is retransmitted.

VI. SOFTWARE USED PARAMETER OF CRC

Initially, the digital design (block diagram) will be drafted showing the basic functioning of the hardware.
in terms of the blocks. This will then be coded in a hardware description language (VHDL). The functioning of the coded design is to be simulated on simulation software (e.g. ModelSim). After proper simulation, the design is to be synthesized and then translated to a structural architecture in terms of the components on the target FPGA device (Spartan 3) and perform the post-translate simulation in order to ensure the proper functioning of the design after translation.

**CRC Parameter taken**

<table>
<thead>
<tr>
<th>data&lt;=11110000</th>
</tr>
</thead>
<tbody>
<tr>
<td>16bitaddress</td>
</tr>
<tr>
<td>txaddressinlo&lt;=11110000, txaddressinhi&lt;=11110000, txaddressout&lt;=1111000011110000</td>
</tr>
<tr>
<td>clock=1,reset=0, wrtaddresshi=1, wrtaddresslo=1.</td>
</tr>
</tbody>
</table>

**VII. PERFORMANCE EVALUATION OF CRC CHECKING**

CRC parameter taken to simulate the simulation software modelsim after the address and the data are attached together, we divide them with a constant polynomial of 32 bits and append the remainder of the division along the data and address.

In this simulation coded in Xilinx software transmitter crc32 bit is 1100011011………that crc 32 bit is verified in same in receiver obtained a crc32<=00000000000000000000000000000000 which indicates an error free transmission and simulation result is shown below.

---

**Fig j. Simulation result for CRC at transmit section**

**Fig k. Simulation result for CRC at receiver section**
VIII. FUTURE IMPROVEMENT

Design methodology of CRC is checking properly using HDLC block. The main agenda is HDLC it improve performance of HDLC like it more features can be added to increase its utility. There are several modifications, which can further improve its performance. These are listed below:
It can be further enhanced to 256 Independent, Bi-directional HDLC Channels, Large 16kB FIFO in Both Receive and Transmit Directions and Transmit Packet Priority Setting.

IX. CONCLUSION

This paper explains an easy and efficient method of generating automated CRC generations for synthesizable use software Xilinx Plan Ahead 13.1. The check frame sequence generation using cyclic redundancy checks CRC-32 to ensure error-free transmission.

REFERENCE

Journal Papers:

Books:

Proceedings Papers: