FPGA Based HDLC Controllers: Comparative Review

S. D. Samudra
Bharati Vidyapeeth Deemed University, Pune, Maharashtra, India

Abstract
For communication of data over a network a protocol is required. The layer 2 of OSI model defines such protocols. HDLC is one of these protocols. This protocol is commonly used in networking. Many chips have been designed for HDLC controllers. Not all the features of these controllers are always needed. With the increasing popularity of FPGA, controller chips are being designed according to the needs of the communication system. This paper discusses the design and implementation of some such HDLC controllers.

Keywords
HDLC Controller, FPGA, Customization

I. Introduction
HDLC protocol is a common protocol in networking. HDLC is the mother of all LANs and WANs. SDLC protocol was adopted by IEEE and defined as HDLC. HDLC is defined in the second layer of OSI model. It is preferred because of the advantages offered namely supports both half duplex and full duplex communication lines, offers error control (used for error correction in modems), efficient and powerful synchronization features. Due to these features HDLC controller chips have been designed. Many networking applications use HDLC controller for communication. For faster implementation HDLC controllers were designed and developed according to the requirements of the basic communication system using FPGA. Researchers have presented work where more number of channels is implemented using FPGA and also simple controller designs as a part of a large system eg. the implementation for 128 channels based on FPGA, implementation where HDLC controller is a part of an SOC. In this paper, a comparative review of these applications of HDLC controller has been presented. Section 2 discusses the HDLC frame structure. In section 3, the various designs and implementations are discussed along with their results.

II. HDLC Frame Structure
The data transmitted is termed as a frame. The frame has to follow a predefined format called as frame structure.

<table>
<thead>
<tr>
<th>Flag</th>
<th>Address</th>
<th>Control</th>
<th>Data</th>
<th>FCS</th>
<th>Flag</th>
</tr>
</thead>
</table>

The frame starts and ends with same bit string (01111110) called flag. These are called header and frame end. Address field holds the address of the receiving station. Control field holds the control code. Data field is the information to be transferred. FCS adopts the CRC as calculating polynomial. Bit stuffing is performed on the frame i.e. a ‘0’ is inserted after a sequence of five ‘1’s. Thus ‘01111110’ is the longest sequence that will appear in the frame other than the flag. After the fifth ‘1’ the next bit is checked for a ‘1’ or ‘0’. If a ‘1’ is found the code is taken as a flag or else as a data. The control field varies depending on the mode of operation. There are three modes of operation for a HDLC controller.

III. Study of Various Designs and Implementations
The requirement for communication in each application is different. Hence the design for the HDLC controller for each application is different. Some papers have designed for half duplex transmission line, some are using full duplex. The clock requirement, synchronization requirement for every design is different. The CRC polynomial selected can be 16-bit or 32-bit. Mostly CRC-16 has been used.

Most papers have used FSMs to design the transmitter and receiver of their controller. There is a slight variation in the states and state transitions according to the requirements.

Next part of this section presents a review of the controller designs. The papers selected for review are on the basis of application of the controller.

Chen Zhifeng et al have proposed a broadband radar system communication [5], the requirements are few: baud rate not less than 2 Mb/s, the communication distance does not exceed 10 meters, the error rate less than 10-6. In view of the demands for baud rate and stability, the controller is based on a simple register design. Hence, very few hardware resources are needed and resources are saved for large-scale data processing of the wideband radar system. Also to ease the access of the radar signal processor to the controller, a SRAM interface is added.

The controller is in control of the communication between the radar signal processor and the control computer through 485 bus. While processing data, the control computer sends the control instructions to the radar signal processor. The radar signal processor sends back the processed results to the control computer. The data processing is done by following an algorithm.

A half duplex communication system is being used for communication. The HDLC controller operates in three states: idle state, sending state and receiving state. These state transitions are controlled through a controller software program. The HDLC controller has three main modules: control module, send module and receive module.

The HDLC controller is designed using top-down methodology using VHDL in ISE 10.1. The design is simulated in Modelsim 6.3. Programs are written on radar signal processor software to achieve transmit HDLC receiving data. The simulation results show that the frame data are successfully received, with less CRC errors. The controller is synthesized on xc4cfx60-10ff672 of Xilinx FPGA. The hardware resources consumption of the controller have the following statistics:

- Number of Slice Flip Flops: 274 out of 50,560; 1%
- Number of 4 input LUTs: 451 out of 50,560; 1%
- Total Number of 4 input LUTs: 497 out of 50,560; 1%
- Number of BUFG/BUFGCTRLs: 4 out of 32; 12%
- Number of FIFO16/RAMB16s: 2 out of 232; 1%

This HDLC controller has the following features: receiving and sending module programming control, query/interrupt working mode, 8-bit station address freely set, data overflow/CRC error detection, SRAM memory and registers can be accessed by SRAM interface, storage resource can be expanded.

Gao Zhen-Bin et al [6], have implemented a multi-channel HDLC controller on Xilinx Virtex. A double full duplex transceiver has been developed for general applications with built-up dual port RAM and an interrupt controller. It is controlled by the host CPU with few commands. The chip receives and transmits data frames automatically. The status is also provided after the completion of
operation. The control registers dominate the modes of operation and can be easily set. The baud rate of each channel can be changed. The transceiver consists of four main blocks: Interrupt controller, Control unit, Registers and Bus buffers, Two channels: channel A and channel B, Dual port RAM and RAM management unit. The top-down method is followed for designing the transceiver and a FSM is used for RAM management. The device utilization is as follows:

- 4 input LUTs : 854 out of 1536; 55%
- IOBs : 51 out of 98; 52%
- Block of RAMs: 8 out of 8; 100%
- Slice flip flops: 610 out of 1536; 39%

This transceiver can be applied in bit-oriented packet transmission, is suitable for Frame Relay switches, Cable Modem, Private packet data networks and switches etc.

Zhao Jinga et al [1] have implemented the HDLC controller in their FPGA airborne recorder. They have realized the communication between the recorder and the answering machine responder by using the HDLC controller due to its high efficiency for error detection and synchronous transmission. To avoid lose of data due to different machine times the header flag (01111110) is sent 4 times. RAM is used as receiving and transmitting buffers for data. The whole module is using a global clock. Here the HDLC protocol synchronizes the communication and validates the data. The controller signals the DSP that a valid data is available. The DSP fetches the data from the given address, reconstructs the data and stores it as one frame in FLASH.

Jun Wang et al [2] have implemented HDLC controller using VHDL and Altera Cyclone EP1C6T144C8. The defects of controllers based on ASIC have been discussed and the feasibility of an efficient device based on FPGA is proved. FSMs are used to develop the encoding-decoding modules and sending–receiving modules. The states of the FSM have been explained briefly along with the flowchart. 16-bit polynomial is used for calculating CRC. The results obtained are printed as the waveforms and also verified through analysis. This VHDL design offers following advantages:

- Flexible in programming and modification, suitable for different standards of HDLC procedures;
- Use of internal storage in FPGA, solves the problem of external storage devices requirement;
- anticipated function is perfectly implemented and easy to control;
- interfacing with other chips does not change.

S. Hamed Javadi and Ali Peiravi [7] have implemented a HDLC controller based on Modified MT8952B using Xilinx VirtexII FPGA. The transceiver designed works in two modes: normal speed mode and internal control mode. The design offered the following advantages:

- high bit rate 85 Mbps.
- compatible with MT8952, Zarlink HDLC controller
- compatible with ST Bus format.

This controller is suitable for packet switching and ISDN.

S.K. Ahn et al [3] have developed an application of HDLC for information transmission on MAGLEV. The data is transmitted between on-board and ground site in MAGLEV. To facilitate data transmission, a 1.1 km line was constructed with FSK and high level data link control protocol. The need here is swift exchange of information between on-board and ground site taking into consideration mass transportation and rapidity. The full duplex mode of communication is used. Data is encoded in NRZI (non return to zero inverted) for synchronization. CRC-16 is used to calculate the cyclic redundancy check. HDLC controller here consists of Intel 80c188 microprocessor, SCC (serial communication controller), Zilog 85c30 and 8256MUART. The main issue here is of synchronization. Transmitter and receiver use independent clocks for synchronization. The synchronous signal at the receiver is easily detected due to NRZI encoding. Hence in HDLC NRZI the clocking is reliably recovered from the data stream. To synchronize the data stream at the receiver a DPLL is used in NRZI. Hence the synchronization issue is solved in this application. The merits of HDLC namely high transmission efficiency, reliability and bit transparency are applied in this paper. The results of data exchange between on-board and ground-site have been verified through a user interface screen.

Hans Tiggeler et al [4] have designed a System-On-Chip for miniaturization of small satellites. The SoC is mapped on the OBC386 Reference Model. The prototype is based on a XILINX Virtex FPGA. It consists of SPARC V8 microprocessor core “LEON”, developed by European Space Agency (ESA). The HDLC controller used in this SoC is developed in-house and interfaced with the microprocessor via a system bus AMBA. The HDLC controller designed here is based on Zilog’s 8530 SCC controller. The reference model OBC386 shows that only a few capabilities of 8530 are needed. Hence a HDLC controller with fixed format HDLC controller with no clock recovery and a simplified error management is designed. The requirements of design were as follows:

- Flag Generation and Detection
- Abort and CRC Generation and Checking
- Transmit Underrun and Receive Overrun Flags
- Zero Insertion and Deletion
- 32-bit AMBA AHB Bus Interface
- 32-Byte synchronous FIFO to reduce interrupt latency
- Must support Back-to-Back Transmit and Receive Frames
- Programmable Baud rate Generator

All these requirements were satisfied by this FPGA based design. This single channel HDLC controller required only 6% of resources of XCV300 with a clock frequency of 98 MHz after place and route.

IV. Conclusion

The HDLC is the basic module for communication of data in bit-stream form. Through the study of the various applications it can be deduced that the device is still important for error-less data transfer. The FPGA is used due to its advantages of flexibility, upgradability and customization. The memory required is mostly available internally when an FPGA is used. This increases the speed of the controller and makes it compatible with today’s high speed technology. Also the controller can be simple or complicated depending on the features required. Due to these advantages of FPGA technology, the HDLC controller can be built as a standalone multi-channel chip or embedded in large systems like SoC.

References


