Australian Journal of Basic and Applied Sciences, 3(4): 4125-4131, 2009
ISSN 1991-8178

Design and Implementation of a High Bit Rate HDLC Transceiver Based on a Modified MT8952B Controller

S. Hamed Javadi and Ali Peiravi

Department of Electrical Engineering, School of Engineering, Ferdowsi University of Mashhad, Mashhad, IRAN.

Abstract: To transmit and receive data over any network successfully, a protocol is required to manage the flow. High-level Data Link Control (HDLC) protocol is defined in Layer 2 of OSI model and is one of the most commonly used layer 2 protocols. HDLC supports both full-duplex and half-duplex data transfer. In addition, it offers error control and flow control. Using a modified MT8952B controller design, the current research presents a new method for implementing an ultra high bit rate HDLC Controller that is compatible with ST-BUS format using Xilinx Virtex FPGA as the target technology using VHDL for implementation. The HDLC Transceiver is used to transmit and receive the HDLC frames. Implementing the HDLC protocol transceiver in FPGA offers the flexibility, upgradeability and customization benefits of programmable logic and also reduces the total cost of every project which involves HDLC protocol controllers.

Key words: HDLC, FPGA, VHDL, CRC-CCITT

INTRODUCTION

HDLC is one of the most important data link control protocols which is widely used. It is the basis for many other important data link control protocols, such as LAPB, LAPD and PPP, which use the same or similar formats and the same mechanisms employed in HDLC (William Stallings, 2007). Some key applications of this protocol include frame relay switches, error correction in modems, packet data switches and data link controllers (Amit Dhir, 2000).

Fast hardware implementation of new concepts and innovative thoughts in order to check their validity and possible advantage is possible by using modern field programmable gates such as the digital FPGAs and the analog FPAAs. Examples include reliability improvement, application of Genetic Algorithms in design of unmanned aerial vehicles, linear feedback shift register based stream ciphers, and multiple reference frames compensation in the H264 coder. Peiravi (2009) presented integration of discrete parts using modern VLSI gates such as FPAAs and FPGAs to improve the reliability of the analog computer of a gyroscopic naval navigation system. Allaire *et al.* (2009) presented a parallel FPGA implementation of a Genetic Algorithm solution of the re-planning requirements of path-planning in unmanned aerial vehicles to achieve real-time applicability and claimed to have attained an excellent autonomous path planner. Deepthi *et al.* (2009) presented the FPGA implementation of nonlinear combination generators and clock-controlled generators which are two very commonly used schemes in linear feedback shift register (LFSR) based stream ciphers. Hachicha *et al.* (2009) presented both DSP and FPGA implementations for multiple reference frames compensation in the H264 coder to improve the coding efficiency for sequences which contain uncovered backgrounds, repetitive motions and highly textured areas using a technique based on Markov Random Fields Algorithm relying on robust moving pixel segmentation. They claimed to process 50 frames (128 × 128)/s on the EP1S10 FPGA paltform and 35 frames (128 × 128)/s on the ADSP BF533.

Many researchers have attempted to improve existing designs of various types of data processors using FPGA implementation. For example, Gheorghiu *et al.* (2008) presented an FPGA implementation of a frequency domain equalizer design. Meng *et al.* (2009) proposed the use of FPGAs and an adaptive data prefetching scheme to avoid reconfigurable processing coprocessor stalls due to data unavailability through profiling methodologies and quantitative analysis for processing biological data and show a 42% improvement.

Corresponding Author: Ali Peiravi, Department of Electrical Engineering School of Engineering Ferdowsi University of Mashhad Mashhad IRAN P.O.Box 91775-1111 Postal Code: 9177948974
Phone: 098-511-881-5100 Fax: 098-511-8763302
E-mali: Ali_peiravi@yahoo.com

FPGA implementation of HDLC transceivers is another such example. Lu *et al*. (2002) presented a multi-channel high speed HDLC data processor design which could process 128 logic channel HDLC data simultaneously using Xilinx X2S200 FPGAs. They compared their design with other communication chips of this kind and showed that their design takes more advantages of chip's resources. Qiao *et al*. (2003) presented an FPGA implementation of a new method to converge the HDLC data frames from 12 independent data communication channels into one time-division multiplexing data link using XILINX xc2s200pq208. Chen *et al*. (2003) presented a high speed low-power ASIC design of HDLC controller based on RS-485 bus indicating die area and dynamic power reduction as well as improvement in system reliability. Qasim *et al*. (2003) presented VHDL modeling and FPGA implementation of a single-channel HDLC transmitter using a Xilinx v50bg256 device. They achieved a running frequency of 59.2 MHz. Gao *et al*. (2005) presented the VHDL modeling and design of an HDLC transceiver using Xilinx Virtex FPGA and claimed to have attained a simple, flexible, and easy to use system compared to existing systems. Ying *et al*. (2008) presented a new HDLC protocol controller based on the FPGA parallel technology to guarantee the radar data communication in the air traffic control system and claimed to have an improvement in the robustness and reliability of the whole system.

The HDLC controller presented in this paper offers the bit rate of up to 85 MHz. The signals of our controller are compatible with MT8952 produced by Zarlink which is currently abundantly used in industry. It has two operational modes: normal mode and the internal mode. In the normal mode the data streams are transmitted at a bit rate equal with the system clock frequency. In the internal mode, 32 time channels - channel 0 through channel 31 - are considered and the HDLC frames are transmitted during just channel 1 with bit rate of 2.048 Mbps. During the other channels the transmitting line is held at high impedance. This feature makes this transceiver useful in ISDN for its D-Channel and also in packet switching.

The presented HDLC controller is designed to use the available resources of the target technology efficiently, so it is possible to integrate it with other systems. Our design is also flexible, so it is easily modifiable.

The remainder of this paper is organized as follows: section II describes briefly HDLC frame format. Section III discusses the system of implemented HDLC controller. Implementation of system on Virtex target is described in section IV. A comparison of the proposed HDLC controller with some well-known HDLC controllers is presented in section V.

II. HDLC Frames:

Two HDLC controllers exchange their information via HDLC frames. The structure of HDLC frames is shown in figure 1.

Flag	Address	Control	Data	FCS	Flag

Fig. 1: Structure of HDLC frames (Andrew S. Tanenbaum, 2003).

Flag fields delimit the frame at both ends with the unique pattern 01111110. To avoid the appearance of the flag pattern inside the frame, a procedure known as “bit stuffing” is used. In bit-stuffing, the transmitter will always insert an extra 0 bit after each occurrence of five 1s in the frame between the transmission of the starting and ending flags. In the receiver side, after detecting a starting flag, the receiver monitors the bit stream. When a pattern of five 1s appears, the sixth bit is tested to see whether or not it is a 0 in which case it is deleted. If the sixth bit is a 1 and the seventh bit is a 0, the combination is accepted as a flag. If the sixth and seventh bits are both 1, it indicates an abort condition by the sender.

The address field that is not needed for point-to-point links identifies the secondary station that is to receive the frame. Although not always needed, it is always included for the sake of uniformity (William Stallings, 2007).

HDLC defines three types of frames each of which with a different control field format. Information frames or (I-frames) carry the data for the user. Flow- and error-control data, using the ARQ mechanism, are attached to the information frame, generally refereed to as piggybacking in the literature. Supervisory or S-frames provide the ARQ mechanism when piggybacking is not used. Unnumbered U-frames provide supplemental link control functions. The first one or two bits of the control field serves to identify the frame type. The remaining bit positions are organized into subfields.

The data field of HDLC frame is present only in I-frames and some U-frames. The field may hold any sequence of bits. However, it must consist of an integral number of octets. The length of the information field is variable limited to a system-defined maximum. The frame check sequence or FCS is an error-detecting code that is usually calculated based on CRC-CCITT.

III. System Description:

MT8952B is a popular HDLC controller which is used abundantly in industry (HDLC). The proposed HDLC controller is based on a modified MT8952B controller that is a scheme compatible with this popular HDLC controller as shown in figure 2. This transceiver is designed to work in two modes: Normal Mode and Internal Control Mode. In normal mode, the data stream is transmitted in HDLC format in bit rate of CLK frequency. In internal mode, the data stream is transmitted in ST-BUS format using a bit rate of 2.048 Mbps (ISDN). There are 32 channels in ST-BUS format. Each channel is used for sending a different data stream.

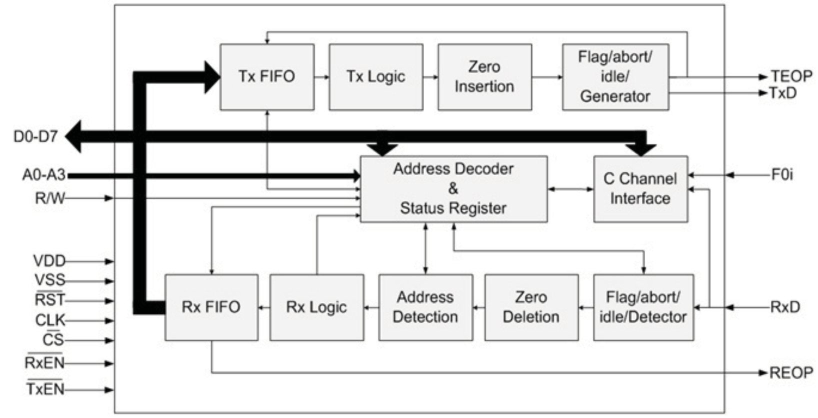


Fig. 2: block diagram of the implemented HDLC Controller.

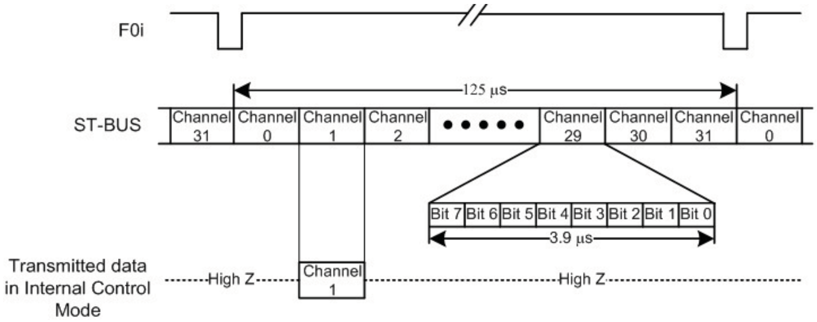


Fig. 3: ST-BUS format (ISDN).

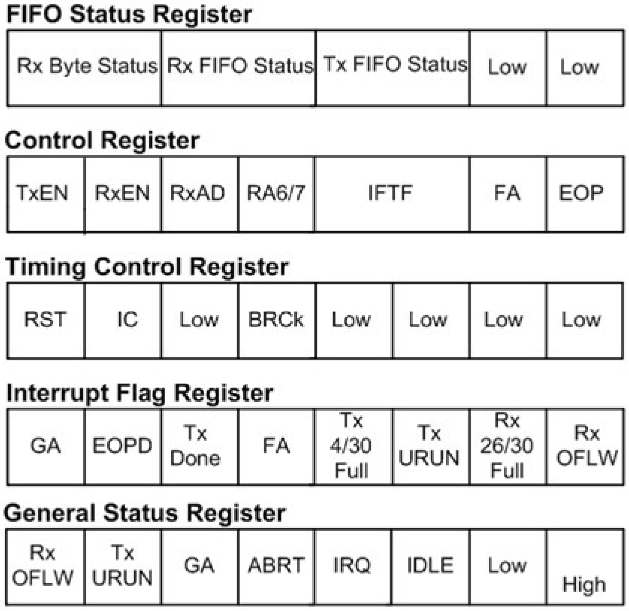
The timing diagram for transmitted data in internal control mode in ST-BUS format is shown in figure 3. The signal F0i is a pulse which defines the beginning of ST-BUS. In internal mode, the data stream is transmitted over channel 1 of ST-BUS and it is high-impedance over other channels. This format makes this transceiver useful in ISDN for its D-Channel and also in packet switching.

In the proposed design, several registers are implemented to control the functions of the transceiver and report its status. These registers are listed in Table 1.

Two 128-bit FIFOs are implemented in transmitter and receiver. The status of these FIFOs is reported by FIFO Status register. Timing Control register is used for changing the transceiver's mode from normal to Internal Control Mode and vise-versa. It is also used for resetting the IC. After resetting the IC which should be provided via a start-up circuit, all bits of Control register and Timing Control register will be cleared (logic 0). So the transceiver is in Normal mode and the transmitter sends idle stream (7FFF$_{\text{Hex}}$). Also the transmitter and receiver are disabled, so Control register should be written using address bus and data bus to enable either transmitter or receiver.

Table 1: The various Registers for control and status reporting purposes together with their addresses.

A0-A3	Registers	
	Read	Write
0	FIFO Status	-
1	Receive Data	Transmit Data
10	Control	Control
11	Receive Address	Receive Address
100	C Channel Control	C Channel Control
101	Timing Control	Timing Control
110	Interrupt Flag	-
111	Interrupt Enable	Interrupt Enable
1000	General Status	-
1001	C Channel Status	-



FIFO Status Register

Rx Byte Status	Rx FIFO Status	Tx FIFO Status	Low	Low

Control Register

TxEN	RxEN	RxAD	RA6/7	IFTF	FA	EOP

Timing Control Register

RST	IC	Low	BRCk	Low	Low	Low	Low

Interrupt Flag Register

GA	EOPD	Tx Done	FA	Tx 4/30 Full	Tx URUN	Rx 26/30 Full	Rx OFLW

General Status Register

Rx OFLW	Tx URUN	GA	ABRT	IRQ	IDLE	Low	High

Fig. 4: Registers used for controlling or reporting the status of HDLC transceiver.

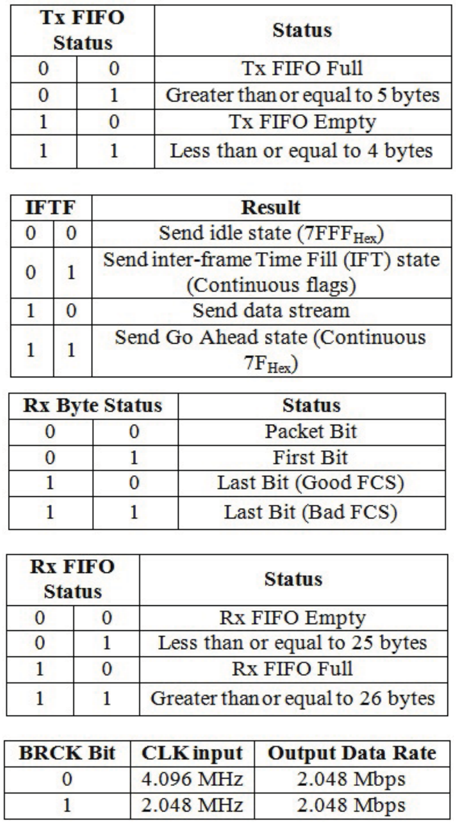
The data is written in transmitter FIFO by addressing Transmit Data register. The first byte of the written data is considered as the first byte of stream. The last byte of stream is determined by setting EOP (End Of Packet) in Control register. If transmitter is enabled, the FCS of data is calculated as the data of transmitter FIFO is serially transferred to Tx Logic. CRC-CCITT is used to calculate the CRC (Andrew S. Tanenbaum, 2003). CRC calculation is implemented by Linear Feedback Shift Reigster (LFSR) method (figure 6). In this method CRC is calculated via distributed EXOR gates.

After calculating FCS, bit stuffing is carried out in Zero Insertion block. Ultimately, the total stream is limited by flags in Flag/Abort/Idle Generator block. The stream, which is now in HDLC format, is transmitted over TXD. TEOP is logic 1 during the last bit of transmitting stream. Any stream can be aborted by setting FA bit in Control register. After aborting a stream, abort sequence (011111111) is sent.

The data to be sent in Internal Timing mode of HDLC transceiver is written in C Channel Control register. The data received in this mode is stored in C Channel Status register.

In receiver side, data is received on RxD. Then, depending on which kind of stream is received, Flag/Abort/Idle Generator reports the function of receiver to General Status register. If a flag is detected, it is ignored and all other bits of the stream are delivered to Zero Deletion block, which deletes each 0 after each five sequence of 1s. After zero deletion, address field of stream is checked by Address Detection block. This block could be either enabled or disabled by RxAD bit in the Control register. Also RA6/7 bit determines the number of bits in address field of received data to be checked.

If checking the address is OK or if it is disabled, the FCS of received data is calculated. Otherwise, the received data is ignored. Whether the FCS of the received data is OK or not, data stream is stored in RX FIFO. The status of FCS is reported by Rx Byte Status bits in FIFO Status register.



Tx FIFO Status		Status
0	0	Tx FIFO Full
0	1	Greater than or equal to 5 bytes
1	0	Tx FIFO Empty
1	1	Less than or equal to 4 bytes

IFTF		Result
0	0	Send idle state (7FFF$_{Hex}$)
0	1	Send inter-frame Time Fill (IFT) state (Continuous flags)
1	0	Send data stream
1	1	Send Go Ahead state (Continuous 7F$_{Hex}$)

Rx Byte Status		Status
0	0	Packet Bit
0	1	First Bit
1	0	Last Bit (Good FCS)
1	1	Last Bit (Bad FCS)

Rx FIFO Status		Status
0	0	Rx FIFO Empty
0	1	Less than or equal to 25 bytes
1	0	Rx FIFO Full
1	1	Greater than or equal to 26 bytes

BRCK Bit	CLK input	Output Data Rate
0	4.096 MHz	2.048 Mbps
1	2.048 MHz	2.048 Mbps

Fig. 5: Bits of registers

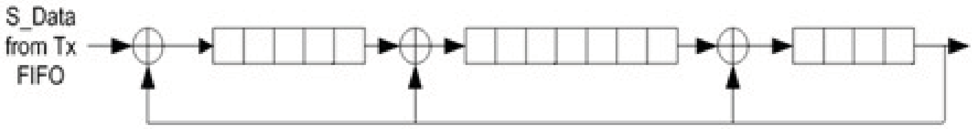


Fig. 6: CRC-CCITT implementation (Glaise, R.J., X. Jacquart, 1993).

IV. System Implementation:

The HDLC transceiver was designed and implemented using VHDL codes. The codes were simulated by ModelSim 6.1b and successfully implemented in Xilinx VirtexII FPGA using Xilinx ISE 9.1 tool. The reason for using Virtex FPGA is its various built-in features that help the designer throughout the process of design. This family provides a broad capability for chip-to-chip communications through programmable support for the latest I/O standards, digital Delay-Locked Loops (DLLs) for clock signal synchronization on the FPGA and on the board, and a memory hierarchy to manage fast access to RAM on and off the chip . Device utilization is shown in Table 2. Maximum bit rate of the implemented HDLC transceiver is 85 Mbps.

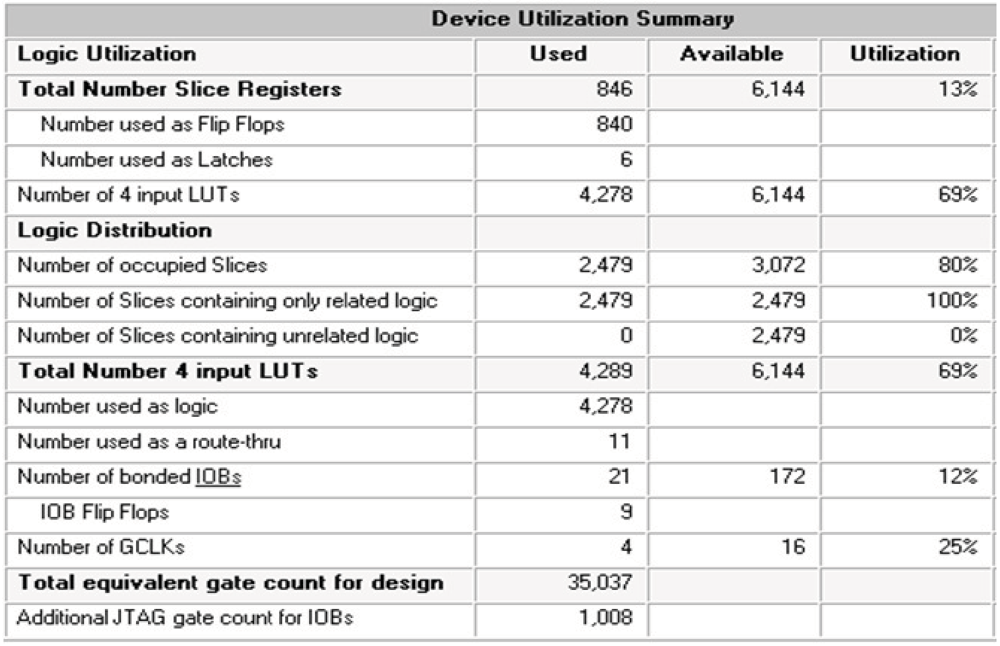
A comparison between the implemented HDLC transceiver in this research and some of the major existing products is presented in Table 3.

Conclusions:

The HDLC transceiver was successfully implemented in Xilinx VitexII FPGA. The most important advantages of this HDLC transceiver are:

- Maximum bit rate of 85 Mbps in normal mode
- Compatibility with ST-BUS format
- Compatibility with MT8952, Zarlink HDLC Controller

Table 2: Implementation of HDLC transceiver in XC2V500.



Device Utilization Summary			
Logic Utilization	**Used**	**Available**	**Utilization**
Total Number Slice Registers	846	6,144	13%
Number used as Flip Flops	840		
Number used as Latches	6		
Number of 4 input LUTs	4,278	6,144	69%
Logic Distribution			
Number of occupied Slices	2,479	3,072	80%
Number of Slices containing only related logic	2,479	2,479	100%
Number of Slices containing unrelated logic	0	2,479	0%
Total Number 4 input LUTs	4,289	6,144	69%
Number used as logic	4,278		
Number used as a route-thru	11		
Number of bonded IOBs	21	172	12%
IOB Flip Flops	9		
Number of GCLKs	4	16	25%
Total equivalent gate count for design	35,037		
Additional JTAG gate count for IOBs	1,008		

Table 3: A comparison between the implemented HDLC transceiver and major existing products.

	KS32C50100	MC92460TS	MT8952	Implemented HDLD
	(Samsung) [	(Motorola) (2002)	(Zarlink)	Controller
Maximum bit rate (Mbps)	10	66.7	2.5	85
ST-BUS Compatibility			✓	✓
Programmable Interrupts	✓	✓	✓	✓
Address detection		4 bytes	1 byte	1 byte
CRC-CCITT	✓	✓	✓	✓
CRC-16	✓			
CRC-32	✓			
No-CRC		✓		

Given the desirability of ST-BUS compatibility which makes this transceiver useful in packet switching, and ISDN for its D-channel, there is a 34-fold improvement in bit rate in this design compared with MT8952. Moreover, a comparison of the performance of the proposed HDLC controller with other major existing controllers shows that the proposed design offers several advantages over existing products in addition to a higher bit rate.

REFERENCES

Amit Dhir, 2000. “HDLC Controller Solutions with Spartan-II FPGAs,” Xilinx Inc.

Allaire, F.C.J., M. Tarbouchi, G. Labonte and G. Fusina, 2009. "FPGA implementation of genetic algorithm for UAV real-time path planning," Journal of Intelligent and Robotic Systems: Theory and Applications, 54(1-3): 495-510.

Andrew S. Tanenbaum, 2003. Computer Networks, 4th edition, Prentice Hall.

“Application notes: Virtex, the Xilinx Virtex Series: Redefining FPGAs – A product background,” www.nalanda.nitc.ac.in/industry/appnotes/xilinx/documents/prs_rls/vtxbackg.htm.

Chen, H. and Y.Q. Han, 2003. "ASIC design of high-speed low-power HDLC controller," Journal of Beijing Institute of Technology (English Edition), 12(SUPPL.): 66-69.

Deepthi, P.P., D.S. John and P.S. Sathidevi, 2009. "Design and analysis of a highly secure stream cipher based on linear feedback shift register," Computers and Electrical Engineering, 35(2): 235-243.

Gao, Zhen-bin and Jian-Fei Liu, 2005. "FPGA implementation of a multi-channel HDLC protocol transceiver", In Proceedings of the 2005 International Conference on Communications, Circuits and Systems, 2: 1300-1302.

Gheorghiu, V., S. Kameda, T. Takagi, K. Tsubouchi and F. Adachi, 2008. "Implementation of frequency domain equalizer for single carrier transmission," In Proceedings of the 4th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM '08.

Glaise, R.J., X. Jacquart, 1993. “Fast CRC calculation,” IEEE International Conference on Computer Design: VLSI in Computer and Processors.

Hachicha, K., D. Faura, O. Romain and P. Garda, 2009. "Accelerating the multiple reference frames compensation in the H.264 video coder," Journal of Real-Time Image Processing, 4(1): 55-65.

“Implementing an ISDN Architecture Using the ST-BUS,” Zarlink Semiconductor.

“KS32C50100 RISC MICROCONTROLLER,” Datasheet, Samsung Electronics.

Lu, Y., Z. Wang, L. Qiao and B. Huanq, 2002. "Design and implementation of multi-channel high speed HDLC data processor," IEEE International Conference on Communications, Circuits and Systems, and West Sino Expositions, 2: 1471-1475.

Meng, X. and V. Chaudary, 2009. "Boosting data throughput for sequence database similarity searches on FPGAs using an adaptive buffering scheme," Parallel Computing, 35(1): 1-11.

“MT8952B HDLC protocol controller,” Datasheet, Zarlink Semiconductor.

“MC92460 HDLC Controller Technical Summary,” Datasheet, Freescale Semiconductor Inc, Rev.1, 2002.

Peiravi, A., 2009. "Reliability improvement of the analog computer of a naval navigation system by derating and accelerated life testing," Journal of Applied Sciences (JAS), 9(1): 173-177.

Qiao, L., Z. Wang, J. Jing, T. Huang and X. Wang, 2003. "Design of DCC multiplexer used in complex SDH network elements," Journal of Southeast University (Natural Science Edition), 33(6): 703-706.

Syed Manzoor Qasim and Shuja A. Abbasi, 2003. “FPGA Implementation of a single-channel HDLC layer-2 protocol transmitter using VHDL,” In Proceedings of the 15th International Conference on Microelectronics, ICM2003, pp: 265-268.

William Stallings, 2007. Data and Computer Communications, Eight edition, Prentice Hall, pp: 222.

Ying, S.C. and X. Zhang, 2008. "New HDLC protocol controller based on the FPGA," Journal of Sichuan University (Engineering Science Edition), 40(3): 116-120.