Prof N.K.Mittal, Mr Mohd. Ahmed, Ms Aafia Zafar / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 6, November- December 2012, pp.1424-1428

Conception And Implementation Of Medium Access Control Protocol Of IEEE 802.3 Transmitter Using VHDL

Prof N.K.Mittal[#1], Mr Mohd. Ahmed[#2], Ms Aafia Zafar[#3]
Department of Electronics & Communication, Oriental Institute of Science & Technology, RGTU Bhopal, India

Abstract

This paper focuses on the design and implementation of IEEE 802.3 medium access control protocol for transmitter. In this paper we design the Ethernet connection oriented LAN medium access control protocol that converts 32 bit data into 4 bit data for transmission. The behaviour of MAC circuit is described using VHSIC Hardware Description Language (VHDL). A synthesized VHDL model of the chip is developed and implemented on the target technology. This paper will concentrate on the testability features that increase product reliability. It focuses on the design of MAC transmitter chip with embedded Built-In-Self-Test (BIST) architecture using FPGA technology.

Keywords— Local Area Network (LAN), Medium Access Control (MAC), Linear feed Back Register, Logical Link Control (LLC), VHISC Hardware Description Language (VHDL).

I. INTRODUCTION

The **Media Access Control** (MAC) data communication protocol sub-layer, also known as the Medium Access Control, is a part of the data link layer specified in the seven layers of OSI model (layer 2). It provides addressing and channel access control mechanisms that make it possible for several terminals or network nodes to communicate within a multipoint network, typically with a local area network (LAN) or Metropolitan area network (MAN). A MAC protocol is not required in full-duplex point-to-point communication. In single channel point-to-point communications full-duplex can be emulated. This emulation can be considered a MAC layer. The MAC sub-layer acts as an interface between the Logical Link Control sub layer and the network's physical layer. The MAC layer provides an addressing mechanism called physical address or MAC address. This is a unique serial number assigned to each network adapter, making it possible to deliver data packets to a destination within a sub network, i.e. a physical network without routers, for example an Ethernet network. FPGA area and speed optimization to implement computer network protocol is subject of research mainly due to its importance to network performance. The objective of resource utilization of field programming gate array (FPGA) is to allocate contending to embed maximum intricate functions. This approach makes design cost effective and maximizes IEEE 802.3 MAC performance. Binary Exponential Back-Off Algorithm, Very high speed integrated circuit hardware description language (VHSIC-HDL) VHDL coding to implement synchronous counters and FSM coding style influences performance of MAC transmitter [1][3]. However performance of IEEE 802.3 MAC transmitter can be optimized using linear feedback shift register, one hot finite machine (FSM) state encoding style, VHDL coding style and synthesis constraints [1].

ASSUMPTIONS

1. The Size of Transmit buffer is assumed to be equal to maximum allowed size of frame 1500(data) + 6 (Destination Address) + 2(length) = 1508 bytes.
2. Simulation model of LLC and PLS will be used for testing.

II. MAC TRANSMITTER

On receiving 'STRT_XMIT' from the upper layer (LLC) this block makes the 'X_BUSY' signal active and starts the process of monitoring the channel for 'CARRIERS SENSE'. This process is called 'DEFER'. The signal 'CARRIER SENSE' is provided by the physical layer. Defer block monitors the channel for inter-frame gap period, which's 96 bit period. The period is split up into two different slot 60 bit period and 36-bit period [1].During the 60-bit period if it receives 'CARRIER SENSE' as active then the timer is restarted. After 60 bit time period is elapsed, the transmitter does not monitor 'CARRIER SENSE' for next 36 bit period [1] and gives the signal 'XMIT_FRAME'. Once the transmission is started it waits for XMIT_OVER or START_DEF to be asserted and goes to start of defer when either is asserted.

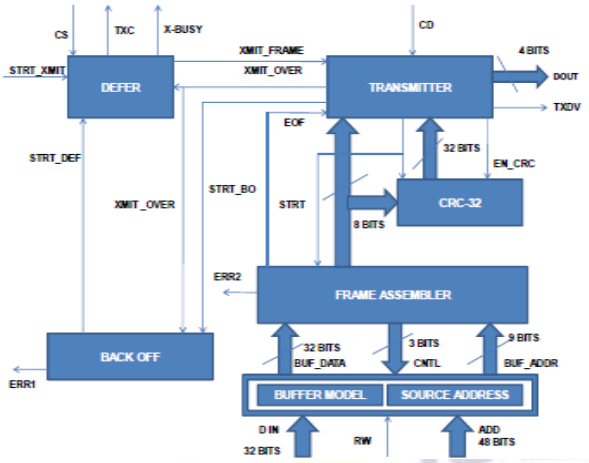

Fig. 1 Block diagram of MAC transmitter

III. OPTIMIZED IMPLEMENTATION OF BACK OFF

When a transmission attempt has been terminated due to collision, it is retried by the transmitter until it is successful or a maximum number of sixteen attempts have been made. The scheduling of the retransmission is determined by a controlled randomization process known as "Truncated Binary Exponential Back Off" [1][4]. After the end of enforcing a Collision (jamming), the transmitter delay before attempting to retransmit the frame delay is an integer multiple of slot times to delay before the retransmission attempt is chosen as a uniformly distributed random integer r in the range 0<=r<=2^k, where k=min (n, 10). If all attempt limits fail, then this event is reported as an error [5].

IV. IMPLEMENTATION OF BACK OFF WITH FINITE STATE MACHINE BASED APPROACH

The function of a frame assembler is to contain the information of the different components of the frame viz. destination address, source address and data, and supply this to the transmitter as well as the CRC block. Hence, the frame Assembler assembles all the fields over which FCS is determined [8]. The Frame Assembler block is controlled by the transmitter block through the STRT signal. This signal is low when the transmitter is idle. The STRT signal function as an enable for the Frame Assembler assembles block. The buffer width is 32 bits. Since the entire frame data is assumed to be present inside the MAC, the Buffer has to accommodate at least 1500 bytes, which is the maximum data size in an 802.3 frame. Here, it is assumed that the entire data contents of a frame reside with the MAC before transmission. In order to address a 32 bit wide buffer of 1500 bytes a 9 bit address is required. The CNTRL (2) is connected to RD pin of the buffer while CNTRL (1) and CNTRL (0) control data to be read from either buffer location of the source address register (1).

TABLE I: CNTRL Data

CNTRL[2:0]	BUF_DATA
011	Source Address (47 to 16)
001	Source Address(15 to 0) on the higher 16 bits
101	Buffer Data

The function of a frame assembler is to construct the 802.3 frame from the following:
1. Destination address stored in first 6 bytes of buffer.
2. Source address that is hardwired onto the MA.
3. Length stored in the lower word of the second location of buffer.
4. Data bytes stored in the subsequent buffer locations.
5. Pad bytes if length is less than 46 bytes to make up a total of 46 bytes of data and pad.

Hence the frame Assembler is tested as follows:-
1. The buffer is filled with some destination address and a length. A source address is also stored in its register.
2. The subsequent buffer locations are filled with randomly generated data vectors
3. The frame assembler is activated by the STRT signal.
4. The byte output of the frame assembler is stored into an output file through text input output.
5. The frame generated is analyzed to isolate its individual components viz. destination Address, Source Address, Length data + Pad. Each field is verified for correct operation.

The total length of the data must match the Length specified in the length field for length between 46 and 1500,where as it should be 46 bytes including data and padding for length field or less than 46. If the length specified is greater than 1500 the frame Assembler should generate an error condition and exit.

The Frame Assembler is tested for the following length conditions Length=0, Length<46, Length=46, Length>46, and Length<1500, Length =1500, Length >1500.

V. OPTIMIZED IMPLEMENTATION OF TRANSMITTER

On receiving 'STRT_XMIT' active from 'DEFER' block this starts transmitting 4 bits at a time. At the same time it gives signal 'Transmit Valid' (TXDV) to the physical layer. First it transmits 7-bytes of 'Preamble' then 1 byte of 'Start Frame Delimiter'(SFD) is transmitted and it also gives 'STRT' signal to 'Frame Assembler' until it receives 'END of Frame'(EOF) signal from the 'Frame Assembler'. Then it transmits 32 bit of 'CRC' and give the signal 'Transmit Over' (XMIT_OVER) to

'DEFER' block and de-asserts the signal 'TXDV' and 'STRT'. Since CRC block works on bytes, 'Frame Assembler' gives 8-bit of data at the

output; whereas 'Transmitter' gives nibble at the output. So it reads from 'Frame Assembler' after 2 clock cycles and 'Frame Assembler' Block also gives output on every 2 clock cycles.

This block also monitor's the signal 'Collision Detected' (CD) provided by the physical layer. If it detects CD during transmitting 'Preamble' then it completes transmitting 'Preamble' and then it transmits 4 bytes of 'JAM' sequence. It also asserts the signal 'Start Back Off' (STRT_BO) and de-asserts signal 'STRT'. If collision is detected anywhere else other than 'Preamble' then 'Transmitter' stops transmitting and sends JAM sequence. It also asserts the signal 'Start Back Off' (STRT_BO) and desserts signal 'STRT'.

VI. OPTIMIZED IMPLEMENTATION OF CRC GENERATOR

A Cyclic Redundancy Check [CRC] is used by the transmit and receive algorithms to generate a CRC value for the FCS field, The frame check sequence [FCS] field contains a 4-octet CRC value, This value is computed as a function of the contents of the Source Address, Destination Address , Length, LLC data and pad. The encoding is defined by the following generating polynomial:

$X32+X26+X23+X22+X16+X12+X11+X10+X8+X7+X5+X4+X2+X1+1$ [4].

Mathematically, the CRC value corresponding to a given frame is defined by the following procedure [4].

1. The first 32 bits of the frame are complemented.
2. The n bits of the frame are then considered to be the coefficients of a polynomial M (x) of degree [n-1]. (The first bit of the destination Address field corresponds to the X (n-1) term and last bit of data field corresponds to the X term)
3. M (x) is multiplied by X32 and divided by G (x), producing a remainder R (x) of degree < 31.
4. The coefficients of R (x) are considered to be a 32-bit sequence.
5. The bit sequence is completed and it results in the CRC.

TABLE II: DEVICE UTILIZATION SUMMARY
Selected Device: SPARTAN II XC2S15cs144-6

S.No.	PARAMETERS	TOTAL NO.	PERCENTAGE
1	Number of slices	50 out of 192	26%
2	Number of slice flip flops	44 out of 384	11%
3	Number of 4 input LUTs	87 out of 384	22%
4	Number of bonded IOBs	17 out of 90	18%
5	Number of GCLKs	01 out of 04	25%

VII. RESULTS AND DISCUSSION

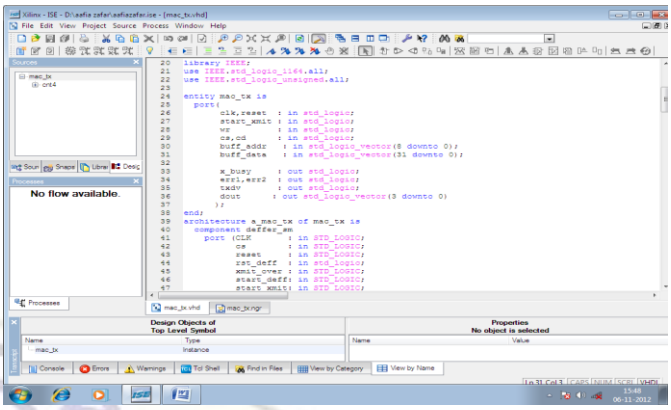

Fig. 2 Synthesis of MAC transmitter

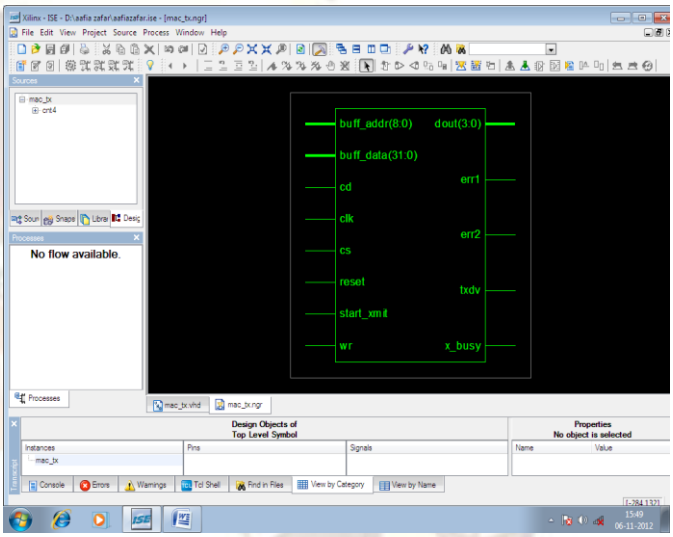

Fig. 3 RTL view of complete MAC transmitter

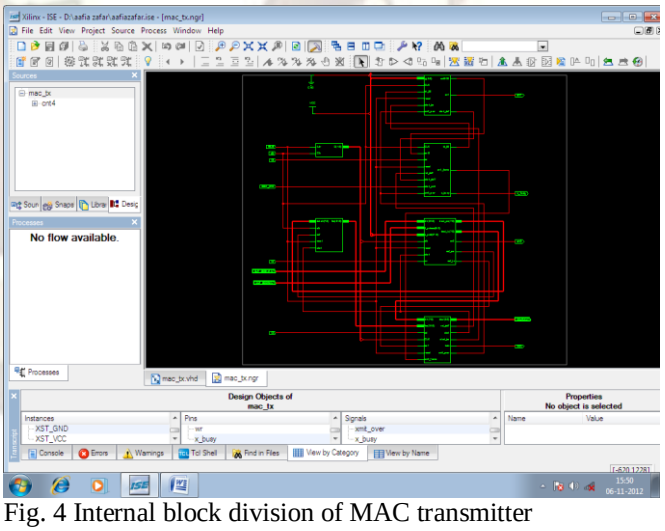

Fig. 4 Internal block division of MAC transmitter

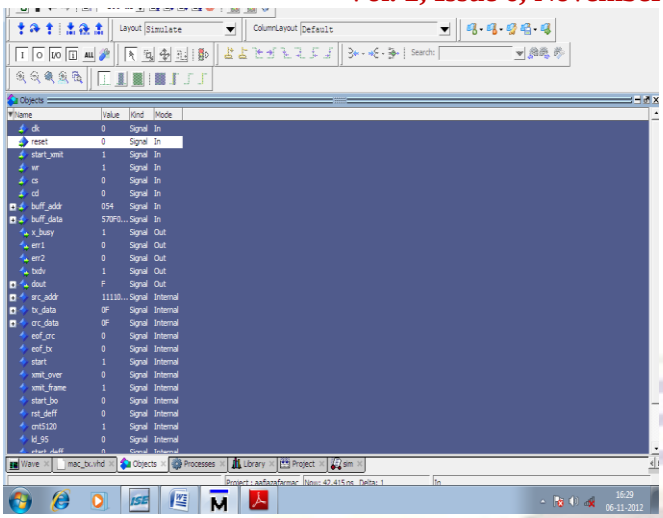

Fig. 5 Signal window of MAC transmitter

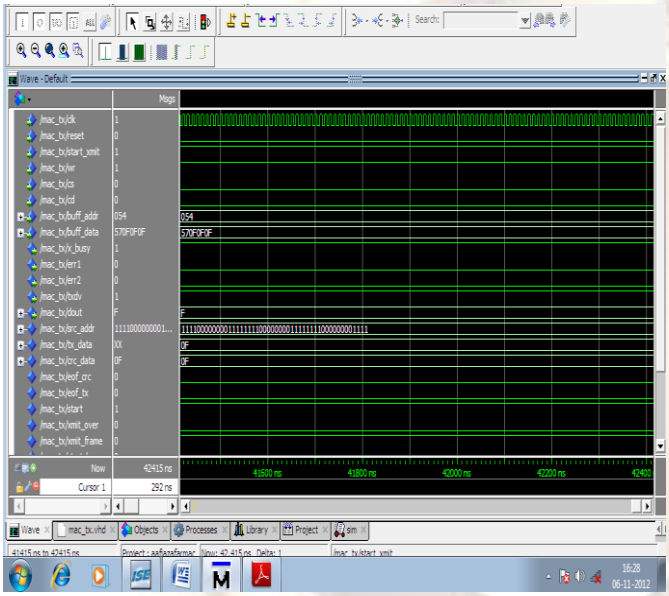

Fig. 6 Simulation waveform of MAC transmitter

The data transmission using MAC transmitter is shown in the above figure in which the time period of tx_clk 50 are 500, write signal is on low position (0) and reset is on high position (1). On changing the position of reset pin from 1 to 0 and making time period of tx_clk as 50, and then running the waveforms; the data is shifted from Buffer to Frame assembler and finally shifted to tx pin (4bit) with standard Frame data. On giving '0' value to reset it gives output as 0000, similarly on giving '1' value to reset and CD it gives output as ZZZZ, which is a don't care condition.

TABLE III: ANALYSIS OF DIFFERENT CONTROL SIGNALS

S.No.	SIGNALS	RESET 0	RESET 1	CD 1
1	clk	1	1	1
2	reset	0	1	1
3	start_xmit	1	1	1
4	cs	0	0	0
5	cd	0	0	1
6	src_addr	48 bit	48 bit	48 bit
7	buff_data	32bit	32bit	32bit
8	x_busy	1	0	0
9	rd	1	0	0
10	err1	0	0	0
11	err2	0	0	0
12	txdv	1	0	0
13	baddr	10001000	00000000	00000000
14	dout	0000	ZZZZ	ZZZZ
15	tx_data	00001111	00000000	00000000
16	crc_data	00001111	10101010	10101010
17	eof_crc	0	1	1
18	eof_tx	0	0	0
19	clken	0	0	0
20	start	1	0	0
21	xmit_over	0	0	0
22	xmit_frame	1	0	0
23	start_bo	0	0	0
24	rst_deff	0	0	0
25	cnt5120	1	1	1
26	start_deff	0	0	0
27	fcs	32 bit	32 bit	32 bit

VIII. CONCLUSION

The VHDL Implementation of MAC gives the improved digital design process, especially for FPGA design. A hardware description language allows a digital system to be designed and debugged at a higher level before conversion to the gate and flip-flop level. One of the most popular hardware description languages is VHISC hardware description language (VHDL) .It is used to describe and simulate the operation of variety IEEE 802.3 systems.

This paper has covered and discussed a software design, and implementation of a basic IEEE 802.3 (MAC Transmitter) system. The speed of data transmission is very high & it gives proper CRC bit for receiving correct data. The simulated waveforms give the reliability of the VHDL implementation to describe the characteristics and the architecture of the designed MAC with embedded BIST. The simulated waveforms also have shown the observer how long the test result can be achieved by using the Built- In-Self-Test technique (BIST) is completed at 39.2ms using 25 MHz clock speed transmitting at 100 Mbps. Even though it seems not to be as fast as it should be when BIST is implemented (the receiver needs to wait the signal from the transmitter), the MAC Transmitter module still takes advantage of the 100% fault coverage. This is the most important

thing that should not be left out by any designer to ensure the reliability of their design. The next target for this research is to verify the RTL, implement and download it on Xilinx's FPGA chip.

REFERENCES

[1] Dr. M.S. Sutone "Performance Evaluation of VHDL Coding Techniques for Optimized Implementation of IEEE 802.3"IEEE transaction on communication, pp-287-293, Jan 12, 2008.

[2] Fedrico Cali, Marco Conti, and Enrico Gregori "IEEE 802.11 protocol: design and performance evaluation of an adaptive Back Off mechanism" IEEE journal on selected areas in communications, vol.18.No.9, September 2000, pp1774-1778.

[3] P.M.Soni and A Chockalingam "IEEE analysis of link layer back off schemes on Point –to-point Markov fading links", IEEE Transaction on communication, vol.51, no.1, January 2003,pp 29-31.

[4] IEEE 802.3 Cyclic Redundancy Check, Xilinx, XAPP209 (v1.0) march 23, Application note: vertex series and vertex II family, 2001, Author by Chriss Borelli.

[5] Kenneth J. Christensen "A simulation study of enhanced arbitration methods for improving Ethernet performance" computer communications 21(1998)24-36 ELSEVIER.

[6] Douglas J. Smith "HDL Chip Design A Practical Guide for designing ,synthesis & simulating ASICs & FPGAs using VHDL or Verilog",3rd edition –MGH.pp179-183 and 198-201.

[7] Neil H.E.Weste and David Harris,"CMOS VLSI Design A circuit and systems perspective"3rd edition –PIE, pp 164-166.

[8] Paran Kurup & Taher Abbasi Kluwar, "Logic synthesis using synopsys" by,Academic publishers, pp. 39 & pp. 135-142.

AUTHORS BIOGRAPHY

Prof. N.K. Mittal has received his B.E. from G.E.C. Jabalpur and M.Tech in Digital Communication from MANIT, Bhopal. He has 21 years of experience in industry in the field of research and development & quality and above 12 years of academic experience. He is currently working as Asst. Professor in the Deptt. of Electronics & Communication in Oriental Institute of Science & Technology, Bhopal. His research interests fall in electronic circuits and optical communication.

Mr Mohammed Ahmed is an Asst Prof in the Deptt of Electronics and Communication in Oriental Institute of Science and Technology. Having done his Masters in Micro Electronics and VLSI Design, he has a vast experience in the field of Front end and backend VLSI Design. His active research areas include Analog and Mixed Signal Design, Real Time Operating Systems and Embedded Systems, VLSI Design, Pattern recognition using Image Processing.

Ms. Aafia Zafar has received her Bachelor of Engineering in Electronics & Engineering from the Oriental Institute of Science & Technology, Bhopal and presently she is an M.Tech scholar at Oriental Institute of Science & Technology, Bhopal