Research Paper

Volume : 2 | Issue : 5 | May 2013 • ISSN No 2277 - 8179

A High Speed Serial Bus Fire Wire Using FPGA

Engineering

KEYWORDS : link layer controller, 1394, CRC, PHY interface, host interface.

Rakesh. B. Patel	Master of Engineering student, L.D. College of Engineering, Ahmedabad
Pankaj P. Prajapati	Asst.Professor, L.D.College of Engineering, Ahmedabad
Rashesh D. Prajapati	Asst.Professor, M.L.Institute for Diploma Studies, Bhandu(Mehsana)

ABSTRACT

In this paper, the design of Fire Wire for a high speed serial bus is possible with a link layer controller. Starting from basic overview of fire wire bus, I describe the various blocks of the Link Layer Controller. The link layer is works between Physical layer and Transaction layer. The various modules for the link layer are design in HDL language and implemented on the FPGA.

1. Introduction

FireWire, originally developed by Apple Computer. It is defined by the IEEE 1394-1995 [FireWire 400], IEEE 1394a-2000 [FireWire 800] and IEEE 1394b standards-that move large amounts of data between computers and peripheral devices. It features simplified cabling, hot swapping and transfer speeds of up to 800 megabits per second. FireWire is a high-speed serial input/output (I/O) technology for connecting peripheral devices to a computer or to each other. It is one of the fastest peripheral standards ever developed and now, at 800 megabits per second (Mbps), it's even faster.

FireWire can connect up to 63 peripherals in a tree or daisy-chain topology (as opposed to Parallel SCSI's electrical bus topology). It allows peer-to-peer device communication - such as communication between a scanner and a printer - to take place without using system memory or the CPU. FireWire also supports multiple hosts per bus. It is designed to support plug and play but not hot swapping. The copper cable it uses in its most common implementation can be up to 4.5 meters (15 ft) long and is more flexible than most parallel SCSI cables. In its six-circuit or nine-circuit variations, it can supply up to 45 watts of power per port at up to 30 volts, allowing moderate-consumption devices to operate without a separate power supply.

II. Link Layer Controller Architecture

Link-layer controller has to support 100Mbps, 200Mbps and 400Mbps three speed modes and finish the CRC coding or verification simultaneously. As illustrated in Fig. 1 , the controller is divided into 6 sub function modules including PHY interface(physical), CRC, host interface, FIFO, transmit and receiving modules.

A. Physical Interface

Link-layer controller communicates with the Physical chips through the Physical interface and handles the bus arbitration and the data transmission or receiving issues. The Physical Interface signals are

(1) Ctl(control signal) is the control signals droved by both linklayer controller and Physical layer chip, LReq(link request) is the request signal driving by link-layer, Ctl and LReq works cooperatively to finish the communication controls between link-layer and Physical layer. The data transmission between the two layers through the D [7:0] signals (8 bits wide in 400Mbps mode).

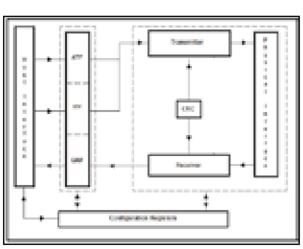

Figure1. Link Layer Controller Architecture

B. CRC-32 bit

The CRC module generates 32-bit CRC. The CRC is calculated to transmit along with the message and to check CRC of the received packet. The purpose of the error detection technique is to enable the receiver, of the message transmitted through a noisy (error-introducing) channel, to determine whether the message has been corrupted or not.

The arithmetic can be described in the polynomial form::

F(x) A degree k-1 polynomial that is used to represent the k bits of the packet covered by the CRC. For the purposes of the CRC, the coefficient of the highest order term shall be the first bit transmitted.

L(x) A degree 31 polynomial with all of the coefficients equal to one, i.e.,

$L(x) = x^{31} + x^{30} + x^{29} + \ldots x^2 + x^1 + 1$

G(x): The standard generator polynomial

$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

R(x) The remainder polynomial that is of degree less than 32.

P(x) The remainder polynomial on the receive checking side that is of degree less than 32.

CRC The CRC polynomial that is of degree less than 32.

Q(x) The greatest common multiple of G(x) in $[x^{32} F(x) + x^k L(x)]$. $Q^*(x)\ x^{32} Q(x)$.

M(x) The sequence that is transmitted.

M*(x) The sequence that is received.

C(x) a unique polynomial remainder produced by the receiver upon reception of an error-free sequence. This polynomial has the value: $C(x) = x^{32}L(x)/G(x)$

$C(x) = x^{31} + x^{30} + x^{26} + x^{25} + x^{24} + x^{18} + x^{15} + x^{14} + x^{12} + x^{11} + x^{10} + x^8 + x^6 + x^5 + x^4 + x^3 + x + 1$

CRC generation equations.

The equations that are used to generate the CRC sequence from F(x) are:

a) CRC = L(x) + R(x) = RS(x) where

RS(x) is the one's complement of R(x).

b) $[x^{32} F(x)+x^k L(x)]/G(x)=Q(x)+R(x)/G(x)$.

c) $M(x) = [x^{32} F(x)] + CRC$.

C. Host Interface

The host interface handles the communication issues between the CPU and the link-layer controller. Through the host interface, the host CPU can access the link layer status registers, set the control registers; during data transmission, the data are pushed into the FIFO in link-layer controller in form of I/O.

D. FIFO

FIFO is used by the two modules:

i. The transmitter module,
ii. Receiver module.

The FIFO block includes two transmit blocks, Asynchronous Transmit FIFO (ATxFIFO) and the Isochronous Transmit FIFO (ITxFIFO). FIFO block includes one Receiving FIFO.

E. Transmitter

The transmitter retrieves data from either the ATF or the ITF and creates correctly formatted serial-bus packets to be transmitted through the physical interface.

The 1394 protocol supports both asynchronous and isochronous data transfers.

1. Asynchronous transfers*:*

Asynchronous transfers are targeted to a specific node with an explicit address. They are not guaranteed a specific amount of bandwidth on the bus, but they are guaranteed a fair shot at gaining access to the bus when asynchronous transfers are permitted. Asynchronous transfers are acknowledged and responded to. This allows error-checking and retransmission mechanisms to take place.

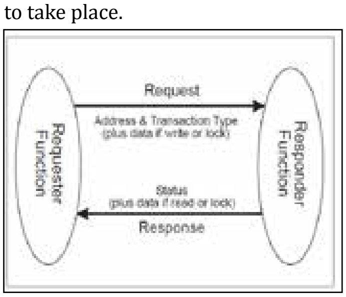

Figure2 . Asynchronous transfer mode.

2. Isochronous transfers*:*

Isochronous transfers are always broadcast in a one-to-one or one-to-many fashion. No error correction or retransmission is available for isochronous transfers. Up to 80% of the available bus bandwidth can be used for isochronous transfers. The delegation of bandwidth is tracked by a node on the bus that occupies the role of isochronous resource manager. This may or may not be the root node or the bus manager. The maximum amount of bandwidth an isochronous device can obtain is only limited by the number of other isochronous devices that have already obtained bandwidth from the isochronous resource manager.

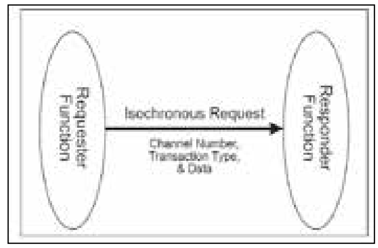

Figure3. Asynchronous transfer mode.

F. Receiver

The receiver gets the data packet from the PHY layer, after the "serial parallel transformation" has been done, data will be sent to a decoder where transformed the data to the format needed by host, then push them into FIFO. The CRC described in section B is also used here for data packet head and data CRC verification. The data has been settled FIFO will be cancelled once the CRC check fail.

V. Conclusion

This Paper present the basic blocks and their functions that needed for Link Layer Controller for high speed serial bus .The design of each block is done using HDL languages. After the simulation and verification the code can be loaded into FPGA, so supporting IEEE 1394 high performance serial bus on system can be much efficiently.

ACKNOWLEDGMENT

I express the great sense of gratitude towards almighty GOD for enormous blessings because of which I have been able to carry out my work. I would like to take this opportunity to express my gratitude to my Guide Prof. P.P.Prajapati for their generous guidance and support given to me. I would like to also thanks Prof. R.D.Prajapati for their guidance and supports.

REFERENCE

[1] T.K. Lewellen, R.S. Miyaoka, L.R. MacDonald, D. DeWitt; M. Haselman, S. Hauck: "Evolution of the Design of a Second Generation Fire Wire Based Data Acquisition System", 2010 IEEE Nuclear Science Symposium and Medical Imaging Conference (2010 NSSIMIC) pp. 2510 - 14 (2010). | | [2] Yu Yun; Wang Danghui; Yang Ke; Feng Zhihua: "Design of link layer for high speed serial bus", 2010 Information Science and Engineering (ICISE), 2nd International Conference on Communication, Networking & Broadcasting; Computing & Processing. | | [3] Texas Instruments, TSB12LV01B IEEE1394 Link- | layer Controller Data manual, Texas Instruments,2000. | | [4] RAMABADRAN T V, and GAITONDE S S, "A | tutorial on CRC computations," IEEE Micro, 1988. | [5] IEEE Std 1394-1995, IEEE Standard for a High | Performance Serial Bus, 1995. | | [6] IEEE Std 1394a-2000, IEEE Standard for a High | Performance Serial Bus-Amendment 1, 2000 | | [7] Zhand Dapu, "IEEE1394 Protocol and Interface | Design,"xduph, 2004 | | [8] Yang Ke, "Design of Asynchronous Transmission Mode | Link Control IP Core Based on IEEE1394,"2010. | | [9] Joung, J., Miyaoka, R., Kohlmyer, S.G., Lewellen, T.K., | A High Resolution Animal PET: Using continuous LSO with intrinsic spatial resoluton approaching 1 mm. J. Nucl Med 42(5):108P, 2001. | | [10] Yu, H., Analog Electronics Designs for High | Resolution PET Detector Systems., Ph.D. Thesis, 1999, University of Washington | |