Designing data transfer link between FPGAs based processor system and a pc computer using ethernet IP core

Mazin Rejab Khalil , Ahmed Mudhafar Mohsin Technical Engineering College Northern Technical University Mosul, Iraq e-mail: <u>mazinrk56@gmail.com</u>

Abstract—The paper aims to construct a fast communication link between any personal computer and a processor system that is configured on field programmable gate arrays to exchange data between them. The exchanged data can be image data, audio signals or any other types of data. The processor system implies a soft core processor with its peripherals and Ethernet IP core along with 128 MB DDR-SDRAM memory to cope with high data density transfer. Wire shark analyzer is used in the personal computer side to capture and analyze the transferred data. C language is used to program the processor system while C# is used to transmit data packets from personal computer. The system performance is tested by transferring certain image data from the personal computer to the processor system.

Keywords—embedded design techniques; ethernet IP core; soft core processor; wire shark software.

I. INTRODUCTION

Processor systems that are implemented and configured on field programmable gate arrays (FPGAs) are usually used to perform certain processing mission in real time operation. The data to be operated on is usually acquired via universal asynchronous receiver transmitter (UART) core that is connected to the processor data bus and a USB to UART bridge device that permits connection to the personal computer(pc) with USB cable. The data transfer speed in this technique is limited to the operating baud rate of the UART core that transfer data serially bit after bit which makes the data transfer slow for high density data items like images. The suggested solution in this paper is to add an Ethernetlite media access controller (MAC) core (IP core) to the processor system that communicates to the processor via the processor bus. The core provides (10-100) Mbps data transfer speed.

The data transfer link design procedure comprises: (1) designing soft core processor system to be configured on Spartan 6 FPGAs using embedded design techniques, (2) adding Ethernet core to the designed processor system, (3) installing wireshark software on the pc under consideration, and (4) testing the system performance by transferring data of certain image between the personal computer and the designed processor system.

II. PROCESSOR SYSTEM HARDWARE DEVELOPMENT

Embedded design techniques were used to construct the hardware part of the processor system in a platform studio[1]. Soft core processor system is developed using the procedure adopted in [2]. Fig. 1 shows the block diagram of the hard ware part of the designed processor system. It consists of Microblaze soft core processor with enabled floating point unit to deal with 32 bits floating point numbers[3], processor local bus (PLB) version 4.6 which is connected to the processor via data interface circuit (DPLB interface) and instruction interface circuit (IPLB interface)[4], multi port memory controller(MPMC) that act as interfacing circuit to 64MB dual data rate synchronous dynamic RAM(DDR SDRAM) which is necessary to enable the system to deal with signals containing large number of samples[5], RS232 terminal connected to the PLB via UART interfacing circuit[5], general purpose 8bits input/ output port (gpio) with its interfacing circuit[6], microprocessor debug module (mdm) with its interface circuit, and a boot memory in the form of a block RAM [7]. Chipscope logic analyzer (ILA) is implanted in the system to display the output signals. Each of the above hardware components are instantiated in the form of intellectual property(IP) with customizable parameters to operate in conformity with each others.





III. ETHERNET IP CORE

Α. Principles of Operation

Fig. 2 displays the block diagram of the Ethernetlite to be added to the designed soft core processor system. It is added to the processor local bus(PLB) through the PLB interface module. The PING transmitter (TX) and receiver (RX) buffers and the PONG (TX) and (RX) buffers are 2 KB dual port memories. The management data input/output (MDIO) master interface module provides access to the physical layer (PHY). The MAC unit consist of transmit and receive modules[8],[9].

Fig. 3 shows Ethernet data frame, the fields in the frame and the bits within the frame are transmitted from left to right. The transmit control multiplexer(MUX) arranges this frame and sends preamble, start frame delimiter (SFD), frame data, padding and cyclic redundancy check (CRC) to the transmit FIFO in a determined order. The preamble field contain seven bytes with pattern "10101010". The (SFD) marks the start of the frame and contains the pattern "10101011". The destination address field is 6 bytes long, its least significant bit determines if the address is individual or group. The six bytes long source address field is provided in the packet data for transmission and retained in receive packet data. The value in Type/length field represent the number of bytes in the following data field, since the maximum length of the data field is 1500 bytes, the number exceeding 1500 in this field represents the frame type. The pad field varies from 0-46 bytes and used to ensure that the frame is at least 64 bytes long. The value in the frame check sequence (FCS) is calculated using cyclic redundancy check(CRC) to ensure data size exchange coincidence[8].

For half duplex operation, the carrier sense signal (PHY_crs) is used to sense the signal from external PHY. In full duplex operation the (PHY_crs) is not used[8].



Fig. 2. Ethernetlite IP.

							_
7	1	6	6	2	0 - 1500	0 - 46	4
Preamble	Start of Frame Delimiter (SFD)	Destination Address	Source Address	Type/Length	Data	Pad	Frame Check Sequence
		•		- 64 - 1518	ovtes		•

Fig. 3. Ethernetlite IP data frame.

Adding Ethernet IP Core to the System В. Hardware

The Ethernet IP core is added to the designed soft core processor system hardware by using embedded design techniques tool. The procedure of introducing an IP core to the processor system is explained in[10] and[11]. Fig.4 shows the block diagram of the designed soft core processor system after adding the Ethernet IP core. Fig. 5 displays the system assembly view issued by the platform studio for the soft core processor system with Ethernet IP core. Fig.6 shows the address map of the resultant designed system.

IV. SYSTEM SOFTWARE DEVELOPMENT

Software Structure Α.

Embedded design techniques tools implies software development platform through which the application programs are developed in C language with necessary libraries and compilers. The application program is prepared here to transmit and receive data via Ethernet IP core to and from the pc. The chipscope logic analyzer (ILA) is used to display, in the processor environment, the transferred data from the personal computer to the soft core processor system. Hyper terminal port is created and connected to the processor system via UART interface to print the data on soft core processor side.

On PC side C# language application programmer interface functions are used to transmit and receive data packets[12],[13]. Wireshark network packet analyzer is installed to capture transferred packet data and analyze them.

Application Program R

The prepared application program is developed with steps as follows.

1) Step 1: Define transmitter (Tx) frame length and receiver (Rx) frame length.

- 2) Step 2: Initialize FIFO.
- 3) Step 3: Initialize ethernet hardware.



Fig. 4. Processor system with ethernetlite IP.

LLP	Bus Interfaces	Ports	Addresses		
M M L	Name	Bus Name	IP 1	ype	IP Version
	dlmb		*	lmb_v10	2.00.b
	ilmb		1 🗙	lmb_v10	2.00.b
S S	mb_plb			plb_v46	1.05.a
	microblaze_0		*	microblaze	8.20.a
▋▋▋▶⋺▶┴┤	🗄 lmb_bram			bram_block	1.00.a
	dlmb_cntlr		- 📩 📩	Imb_bram_i	3.00.b
	ilmb_cntlr		*	Imb_bram_i	3.00.b
····· •	MCB_DDR2		- 📩 🖈	mpmc	6.04.a
	i mdm_0		- 🛨 🖈	mdm	2.00.b
<u> </u>	Ethernet_MAC		- 📩 🖈	xps_etherne	4.00.a
•	GPIO_HDR		- 📩 🖈	xps_gpio	2.00.a
é	RS232_Uart_1		- 📩 🖈	xps_uartlite	1.02.a
	clock_gener		- 📩 🖈	clock_gene	4.02.a
	proc_sys_re		- 📩 🖈	proc_sys_re	3.00.a

Fig. 5. Designed processor system assembly view.

4) Step 4: Set MAC Address and operating speed.

5) Step 5: Check Tx and Rx status.

6) Step 6: Start ethernet device.

7) Step 7: Set up packets to be transmitted.

8) Step 8: Calculate transmitter frame length in FIFO (Tx length).

9) Step 9: Compare the data size in Tx FIFO (Tx length) with defined Tx frame length.

10) Step 10: If Tx length does not equal Tx frame length, write frame data to Tx FIFO.

11) Step 11: Go to step 8.

12) Step 12: If Tx length equals to Tx frame length, initiate transmission

- 13) Step 13: Receive procedure.
- 14) Step 14: Wait for packets to be received.
- 15) Step 15: Check Rx FIFO Occupancy.
- *16)* Step 16: Get the length of arrived packets.
- 17) Step 17: Read received packet data.
- 18) Step 18: Verify the received frame lengh.
- V. RESULTS

Fig.7 represents a sample image whose data are to be transferred from pc side to the soft core processor system. Fig. 8 displays the values of the first ten pixels of the first three rows of the image data quoted from mat lab software on pc side. Fig. 9 shows samples of captured packets in capture (P.Cap) window displayed by wireshark analyzer. Fig.10 displays samples of the data in the captured packets, the samples pointed inside red rectangles are the data of fig. 8. The transferred data to the soft core processor system during acquisition process are shown in fig.11 by using

Bus Interfaces Po	rts Addre	sses				
Instance		Base N	Vame	Base Address	High Address	
🖮 microblaze_0's Addr	ess Map					
dlmb_cntlr		C_BAS	EADDR	0x00000000	0x00001FFF	
ilmb_cntlr		C_BAS	EADDR	0x00000000	0x00001FFF	
Ethernet_MAC		C_BAS	EADDR	0x81000000	0x8100FFFF	
GPIO_HDR		C_BAS	EADDR	0x81400000	0x8140FFFF	
		C_BAS	EADDR	0x84000000	0x8400FFFF	
mdm_0		C_BAS	EADDR	0x84400000	0x8440FFFF	
MCB_DDR2		C_MP	MC_BASE	0x88000000	0x8FFFFFFF	

Fig. 6. Designed processor system address map.



Fig. 7. Sample image on pc side.

	128x128 uint8										
	1	2	3	4	5	6	7	8	9	10	11
1	111	102	106	102	102	104	99	99	100	101	
2	107	104	106	106	100	102	100	98	99	102	
3	106	97	100	96	95	92	92	95	90	99	
4	102	104	110	113	117	130	141	143	134	108	

Fig. 8. Samples of image pixels values on mat lab window.

chipscope analyzer where the values of the first ten samples are shown.

	Time	Sourc	e		Destination	Protocol	Length	Info	
	5 1843.0164				Broadcast	096620		Ethernet	
850	5 1843.0165	72 Hewl	ettP_e7	15:b5	Broadcast	Ethern_	1514	Ethernet	II
857	7 1843.0174	73 Hewl	ettP_e7	7:15:b5	Broadcast	Ethern_	1514	Ethernet	II.
858	8 1843.0182	40 Hewl	ettP_e7	7:15:b5	Broadcast	Ethern_	1514	Ethernet	II
859	9 1843.0190	43 Hewl	ettP_e7	7:15:b5	Broadcast	Ethern	1514	Ethernet	II
866	3 1843.0198	18 Hewl	ettP_e7	1:15:65	Broadcast	Ethern_	1514	Ethernet	II
863	1 1843.0285	51 Hewl	ettP_e7	7:15:b5	Broadcast	Ethern_	1514	Ethernet	II
862	2 1843.0232	77 Hewl	ettP_e7	7:15:65	Broadcast	Ethern_	1514	Ethernet	II
863	3 1843.0252	70 Hewl	ettP_e7	1:15:b5	Broadcast	Ethern_	1514	Ethernet	II
864	4 1843.0271	03 Hewl	ettP_e7	1:15:b5	Broadcast	Ethern_	1514	Ethernet	II
865	5 1843.0293	39 Hewl	ettP_e7	:15:b5	Broadcast	Ethern_	1514	Ethernet	II
866	5 1843.0311	13 Hewl	ettP_e7	1:15:65	Broadcast	Ethern	1514	Ethernet	
861	7 1843.0360	33 Hewl	ettP_e7	1:15:b5	Broadcast	Øxaabb	22	Ethernet	
868	8 1843.0372	12 Xili	nx_01:0	02:03	HewlettP e7:15:8	5 Avaaff	60	Ethernet	TT
rame ther	855: 33 b net II, Sr	ytes on w c: Hewlet	ire (26 tP_e7:1	4 bits), 5:b5 (b0	- 33 bytes captured :5a:da:e7:15:b5),	l (264 bits) o Dst: Broadcas	on inte	erface 0	ff:ff
rame ther De So 0 f 0 0	855: 33 b net II, Sr stination: urce: Hewl f ff ff ff 0 00 00 00 0	ytes on w c: Hewlet Broadcast ettP_e7:19 ff ff b0 00 00 00	ire (26 tP_e7:1 t (ff:f 5:b5 (b 5a da 00 00	4 bits), S:b5 (b0 f:ff:ff: 0:5a:da: e7 15 b 00 00 00	33 bytes captured 53:da:e7:15:b5), ff:ff) 5 aa 00 00 00 0 00 00 00 00	I (264 bits) o Dst: Broadcas	on inte it (ff:	erface 0 :ff:ff:ff:	ff:ff
rame ther > De > So 0 f 0 0	<pre>2 855: 33 b net II, Sr stination: urce: Hewl. f ff ff ff ff 0 00 00 00 00 0 paper.pcapng</pre>	ytes on w c: Hewlet Broadcast ettP_e7:19 ff ff b0 00 00 00	ire (26 tP_e7:1 t (ff:f 5:b5 (b) 5a da 00 00	4 bits), 5:b5 (b0 f:ff:ff: 0:5a:da:0 e7 15 b5 00 00 00	33 bytes captured (5a:da:e7:15:b5), fr:ff) e7:15:b5) 5 aa 00 00 00	I (264 bits) c Dst: Broadca: Z	on inte	erface 0 :ff:ff:ff:	ff:ff

0000	ff	ff	ff	ff	ff	ff	b0	5a	da	e7	15	b5	aa	aa	6f	66
0010	6a	66	66	68	63	63	64	65	64	66	67	66	63	65	5c	5f
0020	61	58	62	99	c4	c6	81	3d	38	35	6e	8e	8c	b1	c3	c1
0030	c5	c5	c7	c6	c5	c4	c4	c 3	be	bb	b1	a5	84	9b	b9	bd
040	a8	82	73	7a	84	8b	95	9d	9a	97	89	82	79	69	82	ae
050	b7	bc	c1	c5	c4	c3	c5	c4	c4	c3	c8	CC	d1	d3	4d	49
060	ab	b 6	bf	ca	a3	53	58	58	58	5b	4c	59	68	5e	5b	5e
070	6f	75	8e	a1	a7	b4	b5	b3	b 8	b6	b7	b 8	b4	9e	5e	69
0806	74	63	5c	82	89	8e	92	91	9b	a6	b3	b1	ae	aa	6b	68
0000	6a	6a	64	66	64	62	63	66	66	64	69	67	66	60	66	65
0a0	61	5b	59	61	93	c4	d1	ab	7c	5c	7d	8c	7e	b1	c3	c2
0b0	c5	cd	cf	cc	c9	c7	c5	c5	bf	ad	8e	8c	a3	b 8	cb	65
10c0	3f	41	4e	4d	4e	4b	48	4a	50	52	57	5f	5c	8b	c1	bf
0b00	bf	c5	cb	c8	c 3	c9	c9	cb	cb	c9	c9	c9	d8	a5	2f	36
0906	b0	c5	a8	9e	80	a2	97	8c	84	4f	4b	54	51	57	5b	73
00f0	7c	8b	9e	a7	ab	b3	b3	b5	b3	bb	b 8	bc	bb	b 8	95	68
0100	72	5f	65	7a	88	8a	8a	90	9a	a6	ab	bØ	ae	ab	6a	61
0110	64	60	5f	5c	5c	5f	5a	63	64	66	65	68	61	6b	62	62
0120	65	66	63	60	58	83	bf	cf	cd	be	a1	82	84	ab	ba	c4

Fig. 10. Data inside the captured packets in P. cap window.



Fig. 11. Data of transferred samples on chipscope window.

Fig. 12 displays samples of the transferred image pixels printed on hyper terminal window after being written in the DDR-SDRAM memory of the processor system, the first group of the data represents the samples of fig. 8 (first ten pixels of the first three rows).

The above mentioned figures(fig. 8 to fig. 12) denote that the pixel values of the transferred image to the soft core processor system have the same values of the pixels of the image on pc side as displayed on mat lab window indicating successful transfer procedure. Table. I shows image pixels values for the first 10 samples of the image data of fig. 7 as displayed on PC side media(mat lab and wireshark windows) and on soft core processor media after being transferred (hyper terminal and chipscope windows).

VI. CONCLUSIONS

From the obtained results, the following conclusions are deduced:

- Ethernet IP core can be used in conjunction with soft core processor system to form a proficient communication link between personal computer and soft core processor system configured on FPGAs.
- Data can be transferred with high precision and high speed between the two environments. Precise transferred data lead to accurate results obtained from different processing operations performed on the data by the soft core processor system.

● Output-HyperTerminal - □ × File Edit View Call Tantfer Help			
File Edit View Call Tander Help	🧞 Output - HyperTerminal	-	×
D	File Edit View Call Transfer Help		
Image receiving done ! Samples of received image first 10 pixel from rows(0,1,2) 111, 102, 106, 102, 102, 104, 99, 99, 100, 101, 107, 104, 106, 106, 106, 106, 106, 99, 99, 102, 106, 97, 106, 96, 95, 92, 92, 95, 90, 99, 101, 106, 106, 106, 106, 106, 106, 106			
C A LOO ALLO A LA LA LA DECO O ALLA CODOLLA CADO ALLA Contrata Deint solar	Image receiving done ! Samples of received image first 10 pixel from rows(0,1.2) 111, 102, 106, 102, 104, 99, 99, 100, 101, 107, 104, 106, 102, 104, 99, 99, 102, 106, 97, 100, 96, 95, 92, 92, 99, 99, 102, 106, 97, 100, 96, 55, 92, 92, 99, 99, 101, 155, 158, 151, 163, 165, 170, 173, 176, 182, 178, 187, 155, 158, 161, 164, 168, 172, 179, 183, 180, 189, 159, 159, 163, 169, 168, 179, 179, 183, 180, 189, 159, 159, 163, 169, 168, 179, 179, 183, 180, 189, 159, 159, 163, 169, 168, 179, 179, 183, 180, 189, 154, 155, 77, 104, 102, 90, 84, 0, 8, 44, 75, 78, 82, 81, 81, 81, 79, -		~

Fig. 12. Data of transferred samples on hyper terminal window.

 $TABLE \ I.$ SAMPLE OF IMAGE PIXEL VALUES BEFORE AND AFTER TRANSFER

		Image Pix	el Value			
Sample Number	P((Befoi	C media re Transfer)	Soft core processor media (After Transfer)			
	Mat lab	Wireshark (Hex)	Chip scope	Hyper terminal		
1	111	6f	111	111		
2	102	66	102	102		
3	106	6a	106	106		
4	102	66	102	102		
5	102	66	102	102		
6	104	68	104	104		
7	99	63	99	99		
8	99	63	99	99		
9	100	64	100	100		
10	101	65	101	101		

REFERENCES

[1] Xilinx, "EDK Concepts, tools and techniques," UG683, available on line. <u>https://www.xilinx.com</u>, 2011.

[2] M.R. Khalil , "Design of an interruptible soft core processor system to compute discrete cosine transform of audio signal," First International IEEE Conference of Electrical, Communication, Computer, Power and Control Engineering/ICECCPCE13, 2013.

[3] Xilinx, "Microblaze processor, reference guide," available on line: https://www.xilinx.com/support/documentation/sw ma nuals/xilinx14_7/mb_ref_guide.pdf; 2013, accessed 2019.

[4] Xilinx, " Pocessor local bus," on line : <u>https://www.xilinx.com/support/documentation/ip_documentation/plb_v46.pdf</u>;2010,accessed 2019.

[5] Xilinx, "Multiport memory controller," on line: <u>https://www.xilinx.com/support/documentation/ip_docu</u><u>mentation/mpmc_v6_06.pdf;</u> 2013,accessed 2019.

[6] Xilinx, "XPS UARTLITE," on line: <u>https://www.xilinx.com/support/documentation/ip_documentation/xps_uattlite/v1_02a/xps_uartlite.pdf;</u> 2011,accessed 2019.

[7] Xilinx, "Local memory bus BRAM controller," on line:

https://www.xilinx.com/support/documentation/ip_docu mentation/Imb bram if controller.pdf; 2011,accessed 2019.

[8] Xilinx, " XPS Ethernetlite Media Access Controller," on line: https://www.xilinx.com/support/documentation/ip_docu mentation/ethernet_media_access_controller.pdf; 2011,accessed 2019.

[9] Kailong Zhang, Liang Hu, Panfei Zuo, Xiao Wu, Kejian Miao, "Wireless extension mechanism and logic design for FPGA-based ethernet power link net," IEEE/ACIS 15 International Conference on Computer and Infofmation Science (ICIS), 2016.

[10] M. R. Khalil, M. A. Hamdoon, "Development of a scheme to connect PC monitor to soft-core Processor", The Mediterranean Journal Of Computers and Networks, Vol.7, No.1, 2011. [11] Mazin R. Khalil, Shaimaa M. Ali. "Designing FPGA Based VGA system to display the components of synthesized electrical circuit using embedded design techniques," First International IEEE Conference of Electrical, Communication, Computer, Power and Control Engineering/ICECCPCE13, 2013.

[12] P. Clausen, Introduction to Programming and The C# Language, bookboon, 2014,pp.289.

[13] S. Nakov, et al , Fundamental of Computer Programming with C#, Telerik Software Academy, 2013.