International Journal of Security and Its Applications
Vol.9, No.8 (2015), pp.417-426
http://dx.doi.org/10.14257/ijsia.2015.9.8.37

Timing-reliable Design of High-speed TigerSHARC 201 Link Transceiver

Peng Wang[1,2*], Ke Gong[1], You-chao Tu[1], Xiao-li Zhang[1] and Xin-wu Chen[1]

[1] *College of Physics & Electronics, Xinyang Normal University, Xinyang, China*
[2] *Communication ASIC Design Centre, Tongji University, Shanghai, China*
[*] *pengpwn1985@163.com*

Abstract

TigerSHARC 201 link interface is a very efficient double data rate protocol. However, its inconsecutive characteristic of clock adds great difficulty for FPGA implementation when transmission rate is too high. In this paper, with specially-designed clock tree, ingenious rate decreasing strategy, plus proper control on FPGA place & routing processing, 8 group link transceivers with data rate over 500MB/s per lane were designed successfully on Xilinx Virtex6 XC6VLX130. Besides, no sophistic IOSERDES component was utilized. Whole design was completed via Verilog RTL code and Xilinx user constrained file. Presented design also has experienced harsh environment test, proved its efficiency and reliability.

Keywords: *TigerSHARC 201; Link Protocol; Transceiver; FPGA*

1. Introduction

Although has been manufactured by Analog Devices for many years, TigerSHARC 201 DSP (TS201) is still widely used especially in radar/wireless communication systems [1-2]. Its core frequency can attain 600MHz. Besides, each TS201 incorporates 4 group full-duplex link interfaces, with highest transmission rate 500MB/s per lane [3]. In many complex real-time processing systems, performances of single or even array TS201 are still insufficient, so FPGA becomes another key component. Unlike DSP, FPGA's internal programmable resources are mainly LUTs, registers, RAMs, multipliers, etc. Hence, FPGA can be regarded as collection of many independent basic function elements, which interprets the most important characteristic of FPGA: parallel processing ability. In many digital signal processing systems, DSP plus FPGA has become the most attractive strategy.

For TS201, convenient method to complete mass data exchange is external bus interface and link interface. Unfortunately, TS201 external bus highest frequency can only attain 125MHz [4]. Besides, only 1 group external bus is available for single TS201 chip, indicating all the TS201 peripherals have to share the same external bus via Time Division Multiplexing (TDM) strategy. In this situation, link interface shows its great advantages: TS201 link interface is an enhanced version of TS101 link interface, all the TS201 link data/clock ports belong to LVDS electrical level (as a contrast, TS101 link interfaces belong to single-ended LVTTL level), so that link interface highest frequency was improved greatly to 500MHz with Double Data Rate (DDR) characteristic. There are 4 group link interfaces in single TS201 chip, each group link interface can complete full-duplex communications under 4-wire or 1-wire mode. Therefore, especially for multiple TS201s plus FPGAs processing platform, link interface becomes the best choice for mass exchange between DSPs and FPGAs. Furthermore, link protocol is also suitable for high-throughput data exchanges between FPGAs [5-6].

For quite a long time, Moore's law is still effective in FPGA realm [7], and architecture of FPGA becomes more and more complicated. Some high-end FPGA can include more

ISSN: 1738-9976 IJSIA

than 500,000 LUT/Register pairs and 50Mbit SRAMs, which is abundant for complex algorithm implementation, whereas FPGA's internal fabric work frequency hasn't been improved synchronously. One of the most vital key is routing delay, especially when design is very considerable, routing delay can attain 3 to 4 ns easily, which limits highest work frequency of FPGA directly. Even for the newest and fastest FPGAs, like Virtex7 [8] (Xilinx Corporation), Stratix V [9] (Altera Corporation), it's still quite difficult to ensure FPGA working steadily with frequency over 200MHz. As a result, the most common work frequency of FPGA is still 100~200MHz or even lower. For application with clock frequency less than 100MHz, it's relatively simple to design link transceiver. However, once TS201 link interface clock frequency is too high, designers have to transmit/receive link data at the rate of double clock frequency, resulting in great challenge. Consequently, the gap between high frequency of interface and low work frequency of FPGA becomes the key point for many high-speed signal processing systems.

Xilinx has released its recommended strategy XAPP727 to solve this problem [10], both regular and featured design were presented in XAPP727. But there are still several defects: strict location constraint for many primitives/pins, tense timing slack for high-frequency situation (regular design), or sophistic IOSERDES primitives plus clock divider have to be instanced(featured design). To overcome these defects, a new architecture for link transceiver was designed. Whole design's data rate can attain over 500MB/s per lane steadily. Currently 8 group link transceivers have been implemented on Xilinx Virtex6 XC6VLX130 successfully with others sophistic processing logic together. The main design strategy was to restrict high-rate clock domain in FPGA I/O registers and neighboring resources, then after ingenious serial to parallel conversion without IOSERDES primitive, others module can access link data under very low clock frequency. Static Timing Analysis was also completed without any timing violation.

2. TS201 Link Protocol

Generally, there are 12 group signals in each TS201 link interface, but it should be stressed that LxACKI/LxACKO, LxBCMPI/LxBCMPO pins are all optional, only LxDATI/LxDATO and LxCLKOUT/LxCLKIN are absolutely necessary for full-duplex communication, this strategy was also adopted in this paper. Detailed pin information of LxDATI, LxDATO, LxCLKOUT and LxCLKIN were listed in Table 1.

Table 1. Key I/O Pin Lists of TS201 Interface

Pin name	Input/Output type	Function
LxDATO3-0P	Output	Output data Transmit port LVDS P terminal
LxDATO3-0N	Output	Output data Transmit port LVDS N terminal
LxCLKOUTP	Output	link output clock port LVDS P terminal
LxCLKOUTN	Output	link output clock port LVDS N terminal
LxDATI3-0P	Input	Input data Transmit port LVDS P terminal
LxDATI3-0N	Input	Input data Transmit port LVDS N terminal
LxCLKINP	Input	link input clock port LVDS P terminal
LxCLKINN	Input	link input clock port LVDS N terminal

The shortest transmission length of link protocol is 128-bit, and 4-wire mode are the most common for link transmission, corresponding timing diagram can be found in Figure 1. It's obvious that DDR technology is utilized to accelerate data transmission rate. Another important characteristic is inconsecutiveness of LxCLOCKOUT, which is only valid when data is transmitting, after that, LxCLOCKOUT returns to low level (idle state).

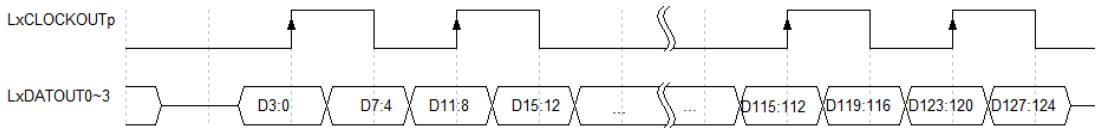

Figure 1. 4-bit Mode Link Transmission Timing Pattern

3. Link Transmitter Design

Traditional design strategy for link transmitter is to generate a clock whose frequency is twice of link clock, so that FPGA fabric can update LxDATO data only in positive edge of new clock, and DDR was transformed to SDR mode [11], which is convenient for HDL synthesis. However, since design goal is to attain transmission rate with 500MB/s per lane, indicating LxCLOCKOUT should be 500MHz, while LxDATO3-0 updating frequency should be 1000MHz because of DDR mode. Such high frequency exceeds FPGA normal work frequency greatly.

To solve this confliction, advanced SelectIO resources of Xilinx FPGA can be used, each FPGA I/O pin contains 1 ILOGIC/OLOGIC pairs. Both ILOGIC and OLOGIC can be configured as ISERDES and OSERDES, which is specially designed for high-speed data transfer. Therefore, OSERDES components are quite suitable for LxCLKOUT/LxDATO signal generator, and widely utilized in our design.

For OSERDES, all the incoming parallel data have to pass through its external ports D1~D6 under writing clock, then these data are converted to high-speed serial data internally. Although single OSERDES's serialization ratio can be set from 2:1 to 6:1, the only supportable DDR serialization ratio for Virtex6 is 4:1 mode [12]. Desired frequency of LxCLOCKOUT is 500MHz, indicating OSERDES writing clock should be 250MHz for 4:1 DDR mode, this frequency is too high for routine FPGA project, so asynchronous FIFO (asFIFO) with independent reading and writing port width mode was used to decrease data rate further. AsFIFO's output data width was 16-bit under 250MHz clock, while input data width can be set to 32-bit or 64-bit, corresponding writing clock was decreased to 125MHz/62.5MHz, which is a rather appropriate frequency for FPGA.

With scenario depicted above, link transmitter can be implemented easily, external 32-bit data were written into transmitter under the rate of 125MHz, then after asFIFO and 4:1 DDR OSERDES processing, an equivalent 8:1 DDR OSERDES was constructed. LxDATO were output from OSERDES under 500MHz DDR mode, and converted to LVDS signal via OBUFDS primitives. LxCLOCKOUT can also be generated similarly: defined a 4-bit width register variable lxclk_div under 250MHz clock domain, whose default was 0b0000, when 16-bit data from asFIFO was available, lxclk_div became 0b1010, else lxclk_div restored its default value, then lxclk_div was transmitted into OSERDES, so that 500MHz clock signal lxclk_tmp was generated. Nevertheless, this temporary clock's phase was aligned with lxdat(3:0), while the best phase relation is: LxCLOCKOUT's phase was delayed 1/4 cycle compared with LxDATO. For 500MHz transmission rate, 1/4 cycle=500ps. Such accurate control in FPGA is nearly impossible via classic delay strategy like inserting multi-stage inverters, or manual placement. Fortunately, Xilinx embedded specialized delay element IODELAYE1 in many FPGA chips. Depending on different delay step (78ps or 52ps), IODELAYE1's work frequency can be set 200MHz or 300MHz. Furthermore, IODELAYE1's control module can reduce

the effects of voltage and temperature variations automatically. Hence, IODELAYE1 is a marvellous choice to modify signal delay.

Basically, IODELAYE1 can be seen as a 31-tap wraparound delay element, and 200MHz work clock is sufficient in this design. For 500MHz link transmitter, ideal tap No. is 6. However, once IODELAYE1 is used, added delay time because of routing also has to be calculated, which results in larger delay value. To attain accurate timing parameter of specific I/O pin, Xilinx OFFSET output constraint can be added on User Constrained File (UCF). Both OFFSET OUT and FROM TO constraint can complete this function [13]. Therefore designer can impose these constraints on LxCLOCKOUT and LxDATO I/O pins to get accurate delay time from professional STA tools. After that, insert IODELAYE1 primitive with proper tap No., and 1/4 phase delay between LxCLOCKOUT and LxDATO was attained successfully.

4. Link Receiver Design

Compared with transmitter, receiver design is much more difficult because of LxCLKIN's inconsecutiveness, which adds great difficulty on cross time-domain processing and data rate decreasing. Generally, the most common strategy to complete cross time-domain transfer is asFIFO. However, asFIFO needs multi-stage synchronizing mechanism to ensure data correction, once LxCLKIN is chosen as asFIFO writing clock, it can't be ensured that all the LxDATIN data are written into FIFO and read out correctly. Hence asFIFO can't be adopted for link receiver cross time-domain buffer. Another difficult is LxCLKIN's frequency and DDR mode, which is too fast for normal FPGA fabric processing.

According to TS201 data sheet, LxCLKIN's phase has been delayed 1/4 cycle (0.5ns for 500MHz) compared with LxDATIN. Consequently, if both PCB and FPGA internal routing paths of LxCLKIN and LxDATIN are length-equal generally, LxDATIN should be registered correctly with LxCLKIN as trigger clock. This phase relationship has to be kept for next processing.

Firstly, all the LxDATIN ports should be assigned into the same I/O bank to improve routing result when possibly. And LxCLKIN can be used as high-speed register clock locally, indicating it should be assigned to Clock Capable (CC) pin, so that dedicated clock resource BUFIO/BUFR/BUFG is available for LxCLKIN.

Secondly, IDDR primitives were utilized to decrease data transfer rate by 50%. There are 3 type different IDDR modes: OPPOSITE_EDGE mode, SAME_EDGE mode, and SAME_EDGE_PIPELINED mode. If IDDR clock is consecutive, SAME_EDGE/ SAME_EDGE_PIPELINED should be adopted for better timing slack. Still because of LxCLKIN's inconsecutiveness, only OPPOSITE_EDGE mode can be chosen, whose output timing pattern is shown in Figure 2.

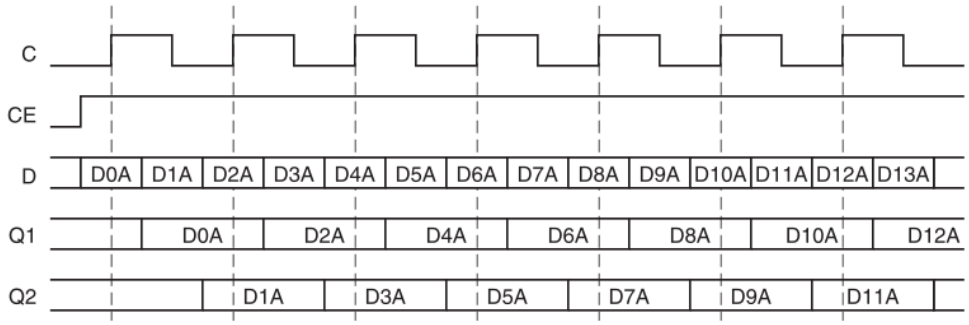

Figure 2. IDDR Timing in OPPOSITE_EDGE Mode

For CC Pin, both BUFIO and BUFR can be used for timing driving. Although BUFIO can only drive I/O Logic components, it has lower routing delay. Therefore, BUFIO was chosen to drive IDDR. To offset routing delay of BUFIO, so that phase relation between LxDATIN and LxCLKIN can be kept, it's necessary to add 0-tap LODELAYE1 on LxDATIN signal path.

Because of very short data valid length (only 1ns for 500MHz DDR mode), routing delay of LxDATIN and LxCLKIN has to be controlled precisely. With the help of PlanAhead tool integrated with ISE suite [14], optimal LODELAYE1 location of each LxDATIN was assigned via UCF. So that fabulous uniformity was attained. (detailed comparison under certain pin location can be found in Table 2 and Table 3)

Table 2. LxDATIN Delay Report without LODELAYE1 Location UCF

Constraint	Check	Worst Case Slack	Best Case Achievable
TS_0A = MAXDELAY FROM TIMEGRP "LINK_DAT0" TO TIMEGRP "FFS" 3 ns	SETUP HOLD	0.144ns 1.479ns	2.856ns
TS_1A = MAXDELAY FROM TIMEGRP "LINK_DAT1" TO TIMEGRP "FFS" 3 ns	SETUP HOLD	0.509ns 1.275ns	2.491ns
TS_2A = MAXDELAY FROM TIMEGRP "LINK_DAT2" TO TIMEGRP "FFS" 3 ns	SETUP HOLD	-0.069ns 1.613ns	3.069ns
TS_3A = MAXDELAY FROM TIMEGRP "LINK_DAT3" TO TIMEGRP "FFS" 3 ns	SETUP HOLD	0.013ns 1.570ns	2.987ns

Table 3. LxDATIN Delay Report with LODELAYE1 Location UCF

Constraint	Check	Worst Case Slack	Best Case Achievable
TS_0A = MAXDELAY FROM TIMEGRP "LINK_DAT0" TO TIMEGRP "FFS" 3 ns	SETUP HOLD	0.606ns 1.222ns	2.394ns
TS_1A = MAXDELAY FROM TIMEGRP "LINK_DAT1" TO TIMEGRP "FFS" 3 ns	SETUP HOLD	0.575ns 1.233ns	2.425ns
TS_2A = MAXDELAY FROM TIMEGRP "LINK_DAT2" TO TIMEGRP "FFS" 3 ns	SETUP HOLD	0.545ns 1.274ns	2.455ns
TS_3A = MAXDELAY FROM TIMEGRP "LINK_DAT3" TO TIMEGRP "FFS" 3 ns	SETUP HOLD	0.570ns 1.252ns	2.430ns

From Table 2 and Table 3, it's clearly that data path disparity was decreased from 0.578ns to 0.061ns, so that even 500MHz DDR data also can be registered correctly via IDDR components. After that, 4 group IDDR output data Q1 and Q2 can be split into 2 parts: all the Q1 data constituted dat_pos. All the Q2 data constituted dat_neg. Dat_pos updated on lxclkin_bufio positive edge, while dat_neg only updated on lxclkin_bufio negative edge. Thus data rate was decreased to SDR 500MHz (see Figure 3). This was an obvious improvement compared with XAPP727, since valid data length can be increased twice, corresponding I/O location also can be more flexible.

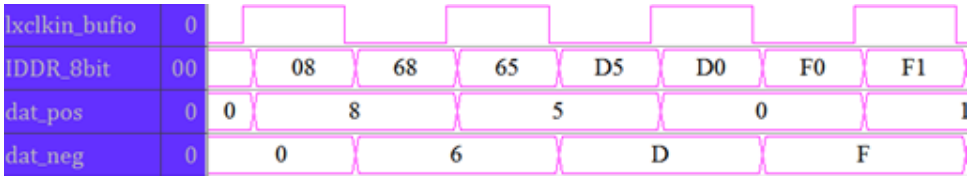

Figure 3. IDDR Timing in OPPOSITE_EDGE Mode

Dat_pos/dat_neg from IDDR should be written into RAM instantly. Theoretically both distributed RAM and Block RAM (BRAM) are all available. However, BRAM is more precious, while distributed RAM is rather plentiful, indicating routing delay can be controlled more easily. Besides, BRAM needs longer setup/hold time, which adds difficulty further because of 500MHz high frequency. Finally, distributed RAM primitive RAM32X1D and RAM32X1D_1 were utilized, the former is positive edge writing mode,

and the latter is negative edge writing mode, hence both dat_pos and dat_neg can be written into RAM with the same clock. This is another important advantage of distributed RAM.

Minimum link data length is 128 bits, so corresponding minimum valid dat_pos and dat_neg No. is 16, therefore RAM32X1D and RAM32X1D_1 can be used for pingpong buffer. There were 4 RAM32X1D and 4 RAM32X1D_1 modules in 1 link receiver, each corresponding to 1 bit data of dat_pos/dat_neg. For the same reason depicted in IODELAYE1 design of LxDATIN, optimal RAM32X1D and RAM32X1D_1 location also should be assigned via UCF according to pin location. Figure 4 showed contrast without UCF and with UCF clearly (white arrow line represented data path from IDDR to distributed RAM).

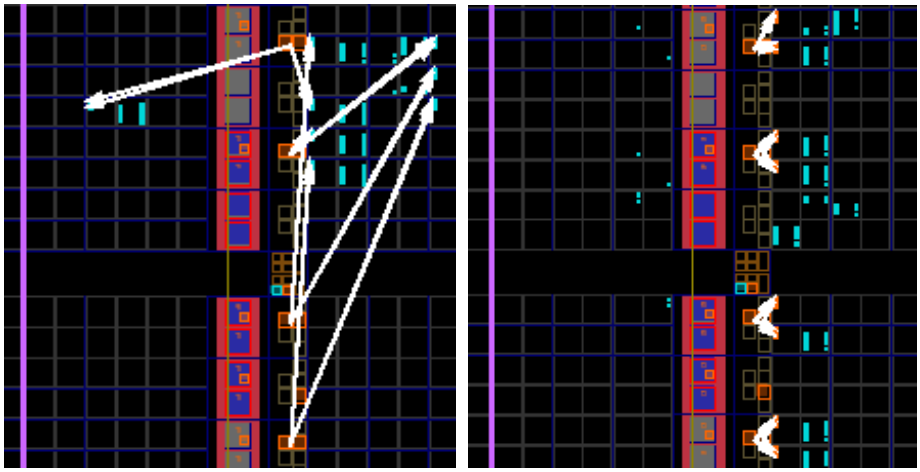

Figure 4. FPGA Placed Results without Distributed RAM UCF (left) and with Proper UCF (right)

Once IODELAYE1 and distributed RAM were located precisely, dat_pos and dat_neg can be written into distributed RAM reliably. Nevertheless, RAM writing clock has to be generated properly. Phase of this clock should be controlled carefully to ensure data/address timing slack (Detailed timing parameter of distributed RAM can be found in FPGA data sheet [15]). According to previous description, dat_pos/dat_neg was triggered by clock from BUFIO. Thereby a natural strategy is to generate properly-delayed version of BUFIO clock, under this situation, IODELAYE1 showed its value again: IBUFDS output LxCLKIN signal were routed to BUFIO and IODELAYE1 simultaneously, and then IODELYE1 outputting signal can be routed to BUFR/BUFG primitives. According to Virtex6 data sheet, BUFR highest frequency can attain 500MHz only for -3 speed grade, while BUFG highest can attain 700MHz even for -1 speed grade, but BUFG is a rather rare resource. Hence whether BUFR or BUFG should be adopted can be decided by designer flexibly. Since -3 speed grade was chosen in presented design, BUFR was finally instanced as distributed RAM writing clock. Besides, this clock was also used to generate writing address (0~31 5-bit counter in fact), while writing enable signal can be set to constant high level because of inconsecutiveness of LxCLKIN.

Figure 5 showed key diagram for receiver design. With IODELAYE1 and distributed RAM located properly via UCF, this architecture can ensure timing slack even for 500MHz high-frequency. Dat_rdclk, dat_rdaddr, and dat_8bit_out were distributed RAM external read port, and Dat_rdclk can be a free-running 500MHz clock drived by MMCM, so that cross time-domain process was completed, too.

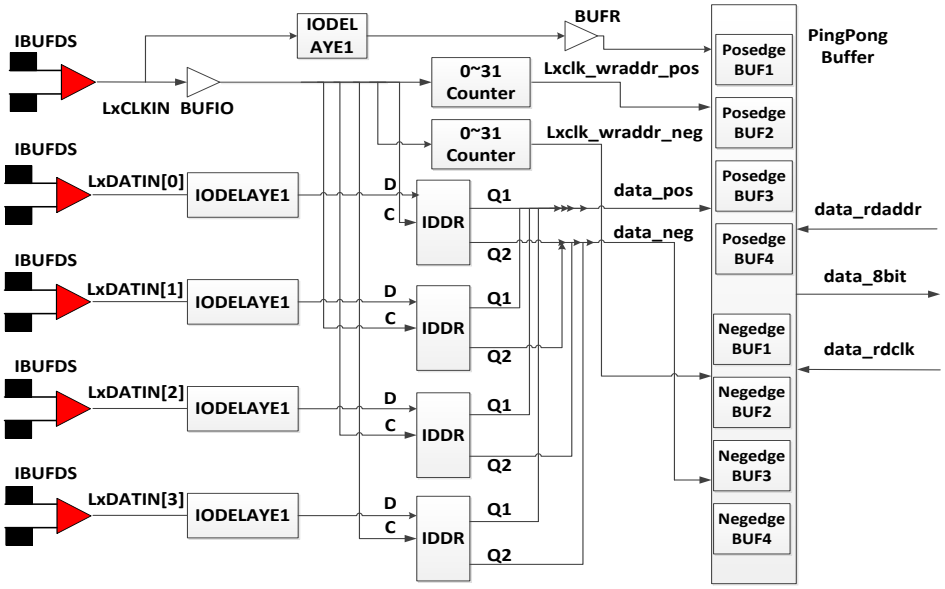

Figure 5. Link Receiver LxCLKIN Clock Domain Architecture Diagram

Up to now, there was still 1 problem in Figure 5. Since LxCLKIN has stopped when link transmission terminated, how can pingpong buffer notice available link data? If it can't output data timely, new received data would cover old data soon. To solve this problem, writing address Lxclk_wraddr_neg changing pattern can be utilized. Link data transfer length is N*128bit according to TS201 data sheet, therefore once successive blocks of link data arrived, Lxclk_wraddr_neg would increase from 0 to 15, then from 16 to 31, 0 to 15...etc. So positive edge of Lxclk_wraddr_neg bit3 can be used to indicate address increasing from 7 to 8 or 23 to 24. And address change from 7 to 8 should always occur first. This edge can be detected reliably by free-running data_rdclk via multi-stage synchronizing and checking mechanism. Once detected, data_rdaddr can start increasing from 0 to 15 to output received 8-bit link data, when data_rdaddr lower 4-bit values attain 15, lower 4-bit value of Lxclk_wraddr_neg after synchronized should be checked again. If link transfer has been terminated, Lxclk_wraddr_neg(3:0) would be 0b0000 because of link transfer length pattern. In this situation, reading operation should be paused until positive edge of Lxclk_wraddr_neg bit3 occurs again. Else if link transfer is still working, Lxclk_wraddr_neg(3:0) would retain increasing pattern, and reading operation should also continue accordingly. With this check mechanism, distributed RAM pingpong buffer data can be transferred to next-stage module automatically.

Pingpong buffer depicted above can output 8-bit link data under free-running 500MHz clock, but this frequency is still too high for FPGA. Hence, BRAM with independent reading and writing port width mode was used to decrease data rate further. Although asynchronous FIFO can complete similar processing, it needs more logic elements, and FIFO's routing delay control is much more difficult for 500MHz work frequency (if FIFO strategy was utilized, maximum frequency is about 330MHz in actual project). Consequently, BRAM with 8-bit write port and 32-bit read port was utilized for 2nd rate decreasing processing. With this strategy, BRAM read clock frequency was decreased to 125MHz, which is a rather low frequency for most FPGA. This BRAM can also be seen 2nd pingpong buffer. And with similar strategy like distributed RAM read address controlling mechanism, BRAM can output 32-bit data timely once link data is ready.

After 2nd buffer processing via BRAM, 32-bit link data under 125MHz clock is available. This rate can be decreased to 62.5MHz (with 64-bit data width) further if necessary. Since data rate has been decreased greatly, others FPGA modules can access

received data without any difficulty. Excellent timing performance of whole design was achieved, too.

5. Implementation Results

With strategy depicted above, 8 group ADSP TS201 link transceivers were implemented on XC6VLX130 successfully with others processing logic. Both transmitter and receiver highest frequency can attain 500MHz steady. Environment experiment & test was also passed without any errors. Besides, if presented design was used for FPGA inter-connection, frequency over 500MHz was also practical, the only limitation was BRAM's maximum frequency. Detailed resource usage of single link transmitter/receiver can be found in Table 4. Compared with existing design, Liu Yao and his colleges only attained 300MB/s on FPGA platform with sophistic constraint [16]. While XAPP727 strategy has no sufficient preponderance in timing slack and I/O pin flexibility. Therefore, strategy depicted in this paper is very appealing in many systems.

Table 4. Resource Lists for Single Transmitter/Receiver

Resource	Register	LUT (Logic)	LUT (RAM)	18kb BRAM	BUFIO	BUFR	IODELAYE1	OSERDES
Transmitter	17	1	-	1	-	-	5	6
Receiver	63	40	16	1	1	1	5	-

6. Conclusion

Currently, TS201 link transceiver based on FPGA was still widely used in digital signal processing systems. Since classical straight implement strategy cannot attain very high speed, and Xilinx recommended strategy XAPP727 also has its disadvantages, presented design show its great advantages in many aspects, especially for high-speed application system. With ingenious internal architecture adopted, both high-speed clock tree and reliable data path were constructed successfully, so that marvelous transmission ability can be achieved. Besides, although presented design's was completed on Virtex6 platform, with proper modification, it can also be implemented in others FPGA like Virtex5/Kintex7, or Stratix III/IV of Altera, indicating excellent compatibility for multi-platform applications. Presented design is especially suitable for radar signal processor and others high-speed real-time processing systems.

Acknowledgments

The authors would like to thank National 863 Foundation of China for their fund support (No. 2013AA7072012D). The authors also would like to thank Key research Foundation for University of Henan province, China for their fund support (No. 15A510011).

Particular thanks to Prof. Fuqiang Liu of Tongji University for his suggestion.

References

[1] Xiang, Hong., Wang, Jun and Zhang, Yuxi, "Parallel implementation of high resolution radar signal processing system based on multi-IC architecture" IET International Proc. on Radar, **(2013)** April, 1-4; Xi'an, China

[2] Baheeja, P,E. and Boominathan, M, Bhai, D.S, "DSP processor based oversampled beam former: An implementation technique". IEEE Proc. Ocean Electronics (SYMPOL), **(2013)** Septembpe, 35-40; Kochi, India

[3] Analog Device Inc. "ADSP-TS201 TigerSHARC Processor Hardware Reference". http://www.analog.com/media/en/dsp-documentation/processor-manuals/396096833ts201_hwr.pdf (accessed Oct 8, **2014**)

[4] Analog Device Inc. "ADSP-TS201S: TigerSHARC Embedded Processor Data Sheet" http://www.analog.com/media/en/technical-documentation/data-sheets/ADSP_TS201S.pdf (accessed Oct 8, **2014**)
[5] Wei, Wu., Jun, Wang and Wei, Li, "Design Methods of Multi-DSP Parallel Processing System", WRI World Congress on Computer Science and Information Engineering, **(2009)** March, 458–464; Los Angeles, USA
[6] Shenggang Yan, Xia Liu and Jianguo Huang, "Underwater target active tracking system based on DSP and FPGA", ICSP International Conference on Signal Processing, **(2008)** December, 2588–2591; Taiwan, China
[7] Mollick E. "Establishing Moore's Law", IEEE Annals of the History of Computing, vol. 28, no. 9, **(2006)**, pp. 62-75
[8] Xilinx Inc. "7 Series FPGAs Overview ver 1.16", http://www. xilinx.com/support/download/ DS180.pdf (accessed Oct 8, **2014**)
[9] Altera Inc. "Stratix V Device Overview ver **2014**.04.08", https://www.altera.com.cn /content/dam/ /global/en_US/pdfs/ literature/hb/stratix-v/stx5_51001.pdf (accessed December 10, **2013**)
[10] Xilinx Inc. "xapp727 application note", http://www.xilinx.com/support/download/xapp727.zip (accessed 4 December 2014)
[11] Pandey, B. and Pattanaik, M, "Low Power VLSI Circuit Design with Efficient HDL Coding". International Conference on Communication Systems and Network Technologies, **(2013)** April, 698-700; Gwalior, India
[12] Xilinx Inc. "Virtex-6 FPGA SelectIO Resources, ver 1.6", http://www. xilinx.com/support/ download/UG361.pdf (accessed November 7, **2014**)
[13] Xilinx Inc. "Timing Closure User Guide ver 14.3", http://www. xilinx.com/support/download/ UG361.pdf (accessed Oct 7, **2014**)
[14] Xilinx Inc. "PlanAhead User Guide, ver 14.6", http://www.xilinx.com /support/download /UG632.pdf (accessed June 26, **2014**)
[15] "Virtex-6 FPGA Data Sheet: DC and Switching Characteristics, ver 3.6", http://www. xilinx.com /support/download/ DS152.pdf (accessed March 18, **2015**)
[16] Liu Yao, Wang Wei, Gong Yu-zhen and Cai Hui-zhi, "Constraining Multiple High-Speed Link Ports in Altera's FPGA". Measurement & Control Technology vol. 31, no. 1, **(2012)**, pp. 116-120

Authors

Peng Wang, He was born in Taikang, Henan Province, China in 1985. He received Doctor Degree of Engineering from Tongji University in 2010. His main research realms are radar signal processing, VLSI deisgn. E-mail: pengpwn1985@163.com