International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 11 (2018) pp. 8991-8997
© Research India Publications. http://www.ripublication.com

Proficient Design Space Exploration of ZYNQ SoC using VIVADO Design Suite: Custom Design of High Performance AXI Interface for High speed data transfer between PL and DDR Memory using Hardware-Software Co-Design

Rikin J Nayak[1], Jaiminkumar B Chavda[2]
[1]Assistant Professor, V.T.Patel Department of E & C, Chandubhai S. Patel Institute of Technology,
Charotar University of Science & Technology, Changa, Gujarat, India.

[2]Assistant Professor, IT Department, Chandubhai S. Patel Institute of Technology,
Charotar University of Science & Technology, Changa, Gujarat, India.

Abstract

In the recent era of Massive scale computing, the new trend of deploying application design in which, not only utilizing the software but also integrate and utilization of custom logic design on dedicated Hardware platform is gaining the focus. This design approach allows to fuse the features of software and hardware based systems in a single package. This paper discusses the design of High Speed Data Acquisition System (DAQ) based on Hardware-Software(HW-SW) Co-Design methodology and its experiments on ZYNQ-7000 SoC(System on Chip) based platform (Avnet ZED Board). This novel experimental setup incorporate the design and development of Custom VIVADO Design for ZYNQ-FPGA(Programmable Logic-PL) and move the data to the Onboard DDR3 Memory using AXI4(Advanced Extensible Interface Ver. 4). Detailed design and results are discussed in this paper.

Keywords: VIVADO, ZYNQ SoC, Hardware-Software Co-Design, PS-PL Interface, First in first out (FIFO), AXI4

INTRODUCTION

The objective of the system is to design custom AXI based interface which controls the data transfer between PL-PS in ZYNQ SoC Architecture by providing control and data bus signals. This system is to be used as a Data Acquisition System (DAQS) which acquires data from multiple external sources/lines by sampling using clock signal and it utilizes the bandwidth of DAQ system. To handle the signals with variable data rates (Kbps to Mbps to Gbps), Asynchronous & Synchronous data, and multiple channels (1-n), with configurability in accommodating in a single system design. The constraints in designing configurable DAQS are imposed by underlying Hardware usage in I/O Speed, Synchronizers, Buffers, Memory Modules, FPGA Logical components, etc. In DAQS, the basic problem is to accurately sample the asynchronous/synchronous high speed data lines from multiple channels at the input interface of the system. The proposed system is designed to have configurability using reconfigurable software and hardware components, which allows flexibility of software with the high performance hardware using high speed computing fabrics. For satisfying both the requirements of handling high speed data with reconfigurable computing, a computing architecture with FPGA and Microprocessor is preferred over the conventional processors and Flexible computing architecture over the Application Specific Integrated Circuits.

In traditional systems, high frequency parallel acquisition tasks with reliability are achieved by using Programmable FPGAs as a front-end part in data acquisition systems, while the processor is used for controlling the data movement with standard interfaces like SPI, SDIO, UART, Ethernet or USB among memory components. Using open source device driver, one can easily program interface using ‘C’ and other high level languages. For onboard data storage, memories of various sizes (Few KB to Few GBs) with variable speed grade is available i.e. SRAM, DDR RAM, FLASH memories and others.

The design of low cost real time DAQ systems are well discussed in various articles [Mehmet et.al., 2008 , B. Nkom et al., 2009, Jason Bank et al., 2012]. Data acquisition process starts with measurement of physical phenomenon. This physical phenomenon could be the room temperature wind flow parameters, light source intensity, stresses and strain, the pressure in a chamber, force applied to an object, or many others. In earlier times, data acquisition system were based on microcontroller unit, but now due to rise of tremendous demand of samples, resolution, and development of semiconductor technology, it is possible to design High performance digital systems which can fuse the available modern processors, digital signal processors(DSP) and FPGAs as a processing unit in the advanced data acquisition system. The all three computing blocks have their own pros and cons in their operational application fields as an individual [Anju P. Raju et al.,2012, Jan Dolinay et al., 2011].

In the design of many DAQ systems combination of FPGA and Processor is being proposed [S. Li et al., 2010] and developed where major challenges includes the designing interface (communication link) between FPGA and processor and verifying the Integrity of data. Depending upon the operating challenges, Data Interface could be PCI/PCIE, USB or any other custom protocol.

Xilinx ZYNQ System on Chip(SoC) is a programmable device which incorporates dual core ARM Cortex MP-Core Processor(PS) and Artrix-7 FPGA(PL) on a single chip package with foot print as small as 13mm X 13mm and it is benefitted from the 28nm fabrication technology [Tao Xue et al.,2014]. Along with, Xilinx Vivado Design Suite is a software tool suite by Xilinx which supports the synthesis and analysis of HDL designs with other features for development and high-level synthesis on ZYNQ SoC. Xilinx provides VIVADO HLS tools for converting high-level language (such as C, Matlab, etc) logical description to a FPGA hardware module, which could accelerate the computing process by software to improve the system's real time performance [Xilinx Inc.UG702, 2012].

The rest of paper is organized as follow: In section 2, related work is described with architecture of ZYNQ and its components.. In section 3, a custom design is explained and realized as well as the results are discussed. In section 4, conclusion of the experiment is placed.

CONSTITUTION OF ZYNQ AND RELATED WORK

The internal architecture of ZYNQ contains PS System, PL System and Communication interfaces with PS-PL Interface as well as with the External Environment using 2 AXI Switches. PS Systems consists of Application processing unit (APU) and Communication Interface Controllers (for DDR and Other Memory units ,RS323-UART, Serial Peripheral Interface(SPI), USB2.0, SD Card Interface (SDIO), Gigabyte Ethernet, etc). PS system incorporates dual core ARM Cortex A9 processor with ARMv7-A architecture. To support media and signal processing architecture, which provides performance for audio, video, image and voice processing and 3D graphics applications Neon coprocessor in included in design [Xilinx Inc. UG925,2012]. In the ZYNQ SoC, all the device controllers are interfaced with the system bus of ZYNQ SoC using AXI Switches. PS is also connected with the DDR Memory using the AXI Dynamic Memory Controller on the AXI Bus System. Zynq SoC provides 4 HP Port Slave interface in PS System.

The PL system is based on an Artix-7/ Kintex 7 FPGA with the programmable resources: configurable logic blocks (CLBs) along with block RAMs and digital signal processing blocks. It also provides analog-to-digital converter (ADC) which gives the opportunity to get sample analog signals from the real-life “analog world” [Xilinx Inc. DS190,2016]. There are 9 Interfaces are available for the communication using AXI Bus Protocol i.e. (1) 4 GP Ports (2 Master and 2 Slave), (ii) 4 HP Port (All Slave) and (iii) 1 ACP Port. In the SoC architecture GP ports implements AXI-Lite Interface while Other Ports Implements AXI- FULL Interface. GP ports are generally used for peripherals controls while HP ports are used for moving a large volume of data at higher rate. GP port is a memory mapped AXI slave interface, on the other side HP ports are AXI master interfaces which communicate with mapped memory regions from DDR, OCM etc. PS can control operation of custom IP in PL using 32 bit registers via AXI GP ports.

This architecture provides rich resources which can be utilized for solving many problems related to parallel processing, different application developed on ZYNQ architectures are [Edavoor et al., 2017, Xue, Tao, et al.,2018]. Author in [Coseriu et al., 2016] used ZYNQ architecture for image defogging by restoring contrast. Recently applications development using such SOCs is developing rapidly. One of application is related to driver awareness monitoring system was developed by author [Schwiegelshohn et al.,2014]. They used car simulator for proving their method. Author [Barbareschi et al., 2015] designed open source project called zedroid which allows android OS to execute on ZYNQ platform. For performing on board processing it is required to have standard protocol for data transfer between host and memory. ZYNQ Architecture has different data mover IP available to transfer data from FPGA to memory without interfering processor. In [Silva et al.,2015] author discussed about available interfaces between PL-PS on a zynq board and based on data rate which one gives effective throughput. Author in [Sklyarov et al., 2017] explored performance of HP ports under different traffic conditions. In the previous work in [Nayak, Rikin et al., 2017] was related to comparison of HP port and ACP (accelerator coherency port) port for data transfer between PL and PS. In this paper, a custom design used for data transfer between PL and PS is discussed in brief. For internal data transfer AXI interface is used which is on chip, however data transfer between two devices with measurable distance provides the engineering critical challenges when the link operates at high frequency.

CUSTOM DESIGNED AXI GP/HP INTERFACE SYSTEM: IMPLEMENTATION AND RESULTS

In the fig. 1, brief design of custom AXI High Performance Interface System, which samples the signals from the external source is visually represented. Here, A data generator is designed on Virtex 5 FPGA which generates the traffic of cumulative 144 Mbps on, 3 separate data lines with speed as 64,64 and 16 Mbps respectively. Each line of data generator source implements 3 wire interface protocol for communication: i. *clock* for synchronization, ii. *Strobe* for stream data start/ stop and iii. *Data* line for stream data. On the Zedboard (ZYNQ 7000 series) signals are received on differential IO pins and sampled in parallel. Since data arrives in a stream format, it requires buffering the temporary storage. The sampled data is serialized and buffered at the FIFO (Block RAM) using PL resources for each of the line.

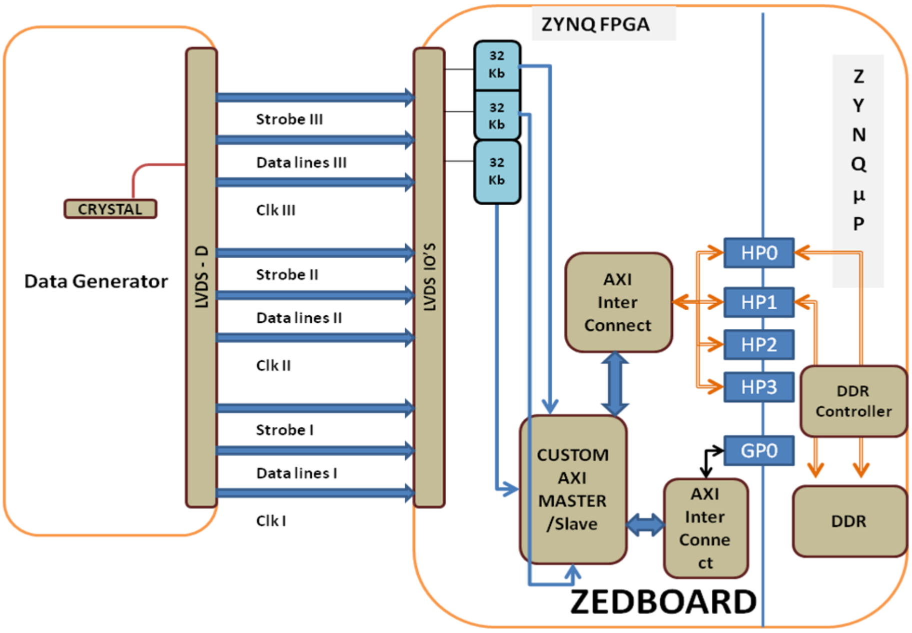


Figure 1. System Design

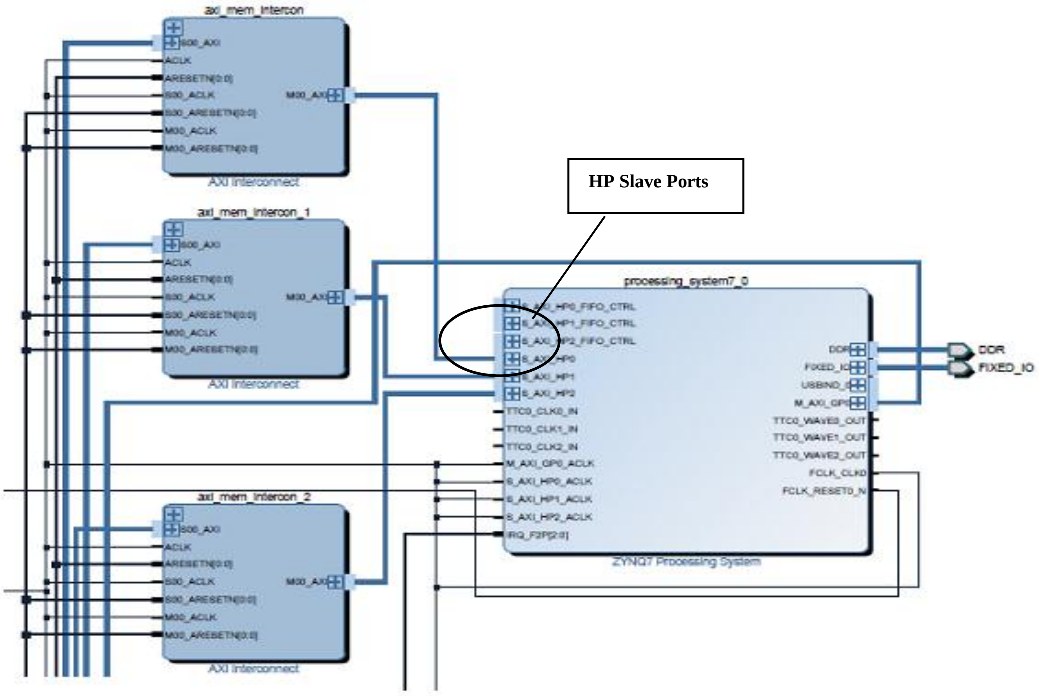


Figure 2(A). The ZYNQ PS Design (Vivado Design Suite)

To access DDR RAM from PL buffer, custom IP logic is designed and developed. . In this design, A custom IP is developed at PL System which comprises of (i) 3 HP Port AXI FULL Master Interfaces, (ii) One AXI LITE Slave interface and (iii) 3 Interrupt Lines. In this design, Three HP Master PL System interfaces are associated with Three HP Slave interface of PS System. Each HP port is mapped to separate address range in onboard DDR3 RAM at Zedboard. At PS, AXI Full Slave port is available which is responsible to transfer data in to and from DDR RAM. AXI Lite Slave Interface is associated with AXI Lite Master Interface using GP Port. The design includes the separate interrupt lines for each of data line to verify the volume of data using Concat IP block and served by Generic Interrupt Controller-GIC at the PS. In ZYNQ SoC, As per design requirements, HP ports satisfies the needs of high speed data transfer. For controlling state machine mechanism, GP ports serves the register based state control.

In Fig 2, a block design using VIVADO Suite is represented. Here, for 3 data lines individual internal FIFO serves as buffers and 3 separate HP ports are linked with each of the input data lines. To Communicate with PS, PL has to program one of the interfaces as AXI Master. As Here, PL drives the communication link working as bus master, PL initiates transfer transaction using *INIT_AXI_TXN* pulse, which transfers the block of data from FIFO buffer and writes the data into DDR3 RAM at designated mapped address. The custom designed IP is handling the separate address *M_AXI_AWDDR* and Data signals *M_AXI_WDATA* for reading and writing the DDR address locations using HP Master ports.

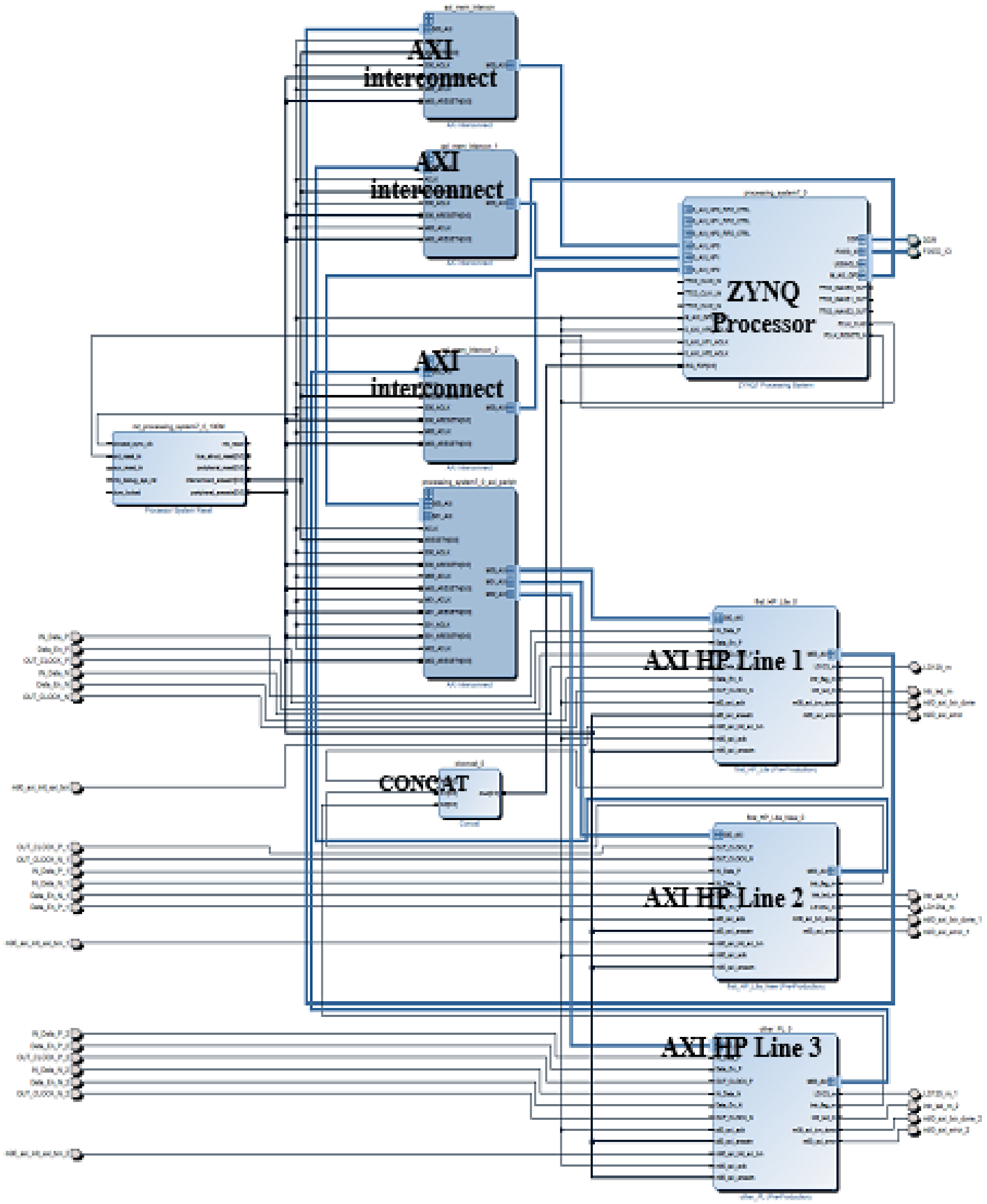


Figure 2(B) The Complete Design (Vivado Design Suite)

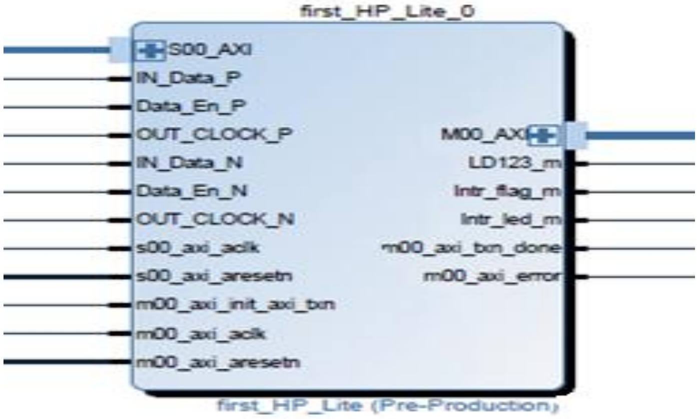


Figure 2(C) The Custom HP Design (Vivado Design Suite)

Steps for designing Hardware-Software based DAQ is as follows:

Hardware Component

1. A custom logic IP which consist of 1 GP port and 3 HP ports.
2. Asynchronous data sampling is achieved and controlled with 3 Wire interface protocol (Data, Clock, Strobe)from external source.
3. Sampled serialized data is buffered at to 32 x 1K FIFO (4KBytes)(block RAM) in PL.
4. As FIFO memory gets filled with 60 % of maximum depth, FIFO_Full signal is raised and FIFO readout operation is carried out which transfer the data from FIFO to pre mapped memory region at DDR RAM. In case of FIFO is partially filled and not data is sampled for sufficient long time, then internal timer timeout event also empties FIFO.
5. After writing suitable minimum block size of memory Blocks (pages) to RAM, interrupt signal is raised (HIGH) by custom IP in FPGA which is acknowledge by processor. Processor executes Interrupt Service Routine (ISR) in response of interrupt using synchronized driver software.

Steps for the software programming are as below.

Software Component

1. In the software, allocate the Buffers of 16MB for each of the line in DDR with initial default values.
2. Initialize Generic Interrupt Controller device and register Interrupt and Interrupt Service Routines.
3. Initialize the custom IP AXI registers to default state.(Device ready to receive external data)
4. Wait for Event in sleep mode till the data of 64KB (16 pages each of 4KB size), which will be notified by interrupt.
5. On event of interrupt, Interrupt Service Routing (ISR) starts serving execution. In ISR, custom IP registers are modified, so that the interrupt signal is reset to default state (LOW) (i.e. software writes the IP registers, Fabric logic updates the interrupt line status to down). Execute Interrupt handlers to modify Custom IP Registers in to update device Status(Interrupt Served and Ready for next interrupt) In the ISR, low priority tasks like data processing and other can be scheduled based on critical time constraints.

After every 64KB block of sampled data is captured in buffered memory at DDR, data verification is performed every memory location as per the known pattern. and data read from DDR memory is sent to RS232-UART for display at relatively slow speed. Fig 3 shows data verification mechanism.

Resource utilization for the above hardware is as shown in Table 1.

Table 1. Resource Utilization

Resource	MAX Available	Utilization(%)
Slice LUTs	53200	8247 (16%)
Slice Registers	106400	4520 (4%)
IO	202	31 (15%)
Clocking	32	4 (13%)

Data verification Methodology

1. Counting Total number of Interrupts generated and served to find the volume of data written to DDR memory. (Currently Interrupt generated at 64KByte data)
2. Generating the Test Data 32bit numbers which increments by one and verifying same at received data pattern block using software programming.
3. Verification of data on altering the combination of individual lines, to verify the volume and values of

each destination buffers.

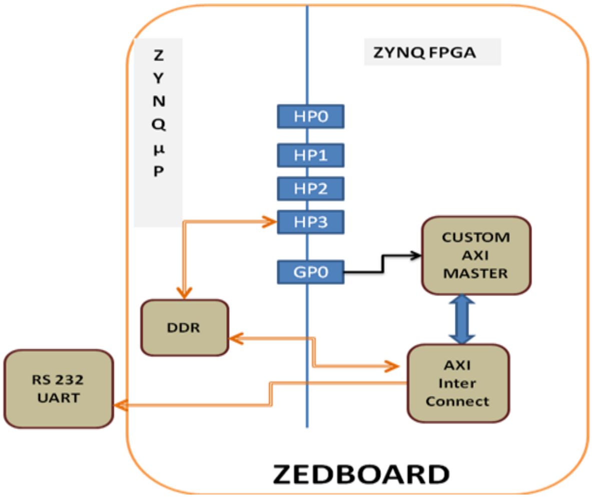
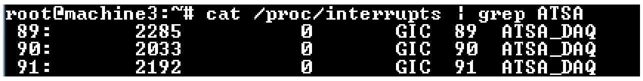


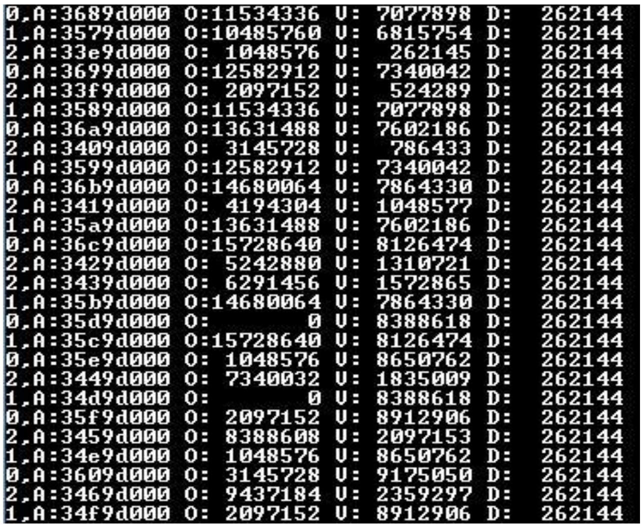
Figure 3. Data verification mechanism

Data is transferred from Custom IP Logic from PL to PS and notification of each 64KB data is verified using an interrupt. Here for three inputs, three interrupts are registered for transfer parallel data to PS. On each 16 pages, each page has 1024x32 bit data, one interrupt is generated on three the line simultaneously. Below fig 4 (A) shows screen shot of registered interrupt no 91, 90 and 89 for three lines with screen shot of received data. Here in Fig 4 (B), image shows, five columns are displayed. each of the columns indicates data line no, address(A), relative offset(O) in 16 MB buffer, Value(V) and difference(D) of the values between two continuous 1 MB block in total 16 MB block respectively.



```
root@machine3:~# cat /proc/interrupts | grep ATSA
 89:        2285          0        GIC  89  ATSA_DAQ
 90:        2033          0        GIC  90  ATSA_DAQ
 91:        2192          0        GIC  91  ATSA_DAQ
```

Figure 4 (A). Registered interrupts



```
0,A:3689d000 O:11534336 V: 7077898 D:  262144
1,A:3579d000 O:10485760 V: 6815754 D:  262144
2,A:33e9d000 O: 1048576 V:  262145 D:  262144
0,A:3699d000 O:12582912 V: 7340042 D:  262144
2,A:33f9d000 O: 2097152 V:  524289 D:  262144
1,A:3589d000 O:11534336 V: 7077898 D:  262144
0,A:36a9d000 O:13631488 V: 7602186 D:  262144
2,A:3409d000 O: 3145728 V:  786433 D:  262144
1,A:3599d000 O:12582912 V: 7340042 D:  262144
0,A:36b9d000 O:14680064 V: 7864330 D:  262144
2,A:3419d000 O: 4194304 V: 1048577 D:  262144
1,A:35a9d000 O:13631488 V: 7602186 D:  262144
0,A:36c9d000 O:15728640 V: 8126474 D:  262144
2,A:3429d000 O: 5242880 V: 1310721 D:  262144
2,A:3439d000 O: 6291456 V: 1572865 D:  262144
1,A:35b9d000 O:14680064 V: 7864330 D:  262144
0,A:35d9d000 O:       0 V: 8388618 D:  262144
1,A:35c9d000 O:15728640 V: 8126474 D:  262144
0,A:35e9d000 O: 1048576 V: 8650762 D:  262144
2,A:3449d000 O: 7340032 V: 1835009 D:  262144
1,A:34d9d000 O:       0 V: 8388618 D:  262144
0,A:35f9d000 O: 2097152 V: 8912906 D:  262144
2,A:3459d000 O: 8388608 V: 2097153 D:  262144
1,A:34e9d000 O: 1048576 V: 8650762 D:  262144
0,A:3609d000 O: 3145728 V: 9175050 D:  262144
2,A:3469d000 O: 9437184 V: 2359297 D:  262144
1,A:34f9d000 O: 2097152 V: 8912906 D:  262144
```

Figure 4 (B). Acquired data (screenshot)

CONCLUSION

In the paper, A custom Designed AXI High Performance Interface System with data acquisition system is developed on Xilinx's Zedboard, which having advantages of on board microprocessor and Field Programmable gate array. AXI interface between PL-PS provides high speed links which eliminates the extra harness between PL and PS. System has several advantages like configurability, portability, lower cost and easy interfaces with high processing power of ARM Cortex processor which can be utilize for processing. System also provides on board data storage up to 512 MB DDR RAM which can be extending up to 1 GB. During the experiment, total data of 256 MB were transferred on three lines simultaneously. In future implementation DAQ system output interface will be designed to transfer high speed data to computer like Ethernet or can store the big size data on onboard SD card; SD card has its own limits on storing the data continuously, which could limit the sampling data rate.

REFERENCES

[1] Xue, Tao, Weibin Pan, Guanghua Gong, Ming Zeng, Hui Gong, and Jianmin Li. "Design of Giga bit Ethernet readout module based on ZYNQ for HPGe." In Real Time Conference (RT), 2014 19th IEEE-NPSS, pp. 1-4. IEEE, 2014.

[2] Guide, Partial Reconfiguration User. "UG702 (v14. 1)." Xilinx Inc., Apr 24 (2012).

[3] Xilinx Inc. UG925, "Zynq-7000 EPP ZC702 Base Targeted Reference Design" 2012

[4] Zynq, Xilinx. "7000." Zynq-7000 all programmable soc overview, advance product specification-ds190 (v1. 2) available on: http://www. xilinx. com/support/documentation-/data sheets/-ds190-Zynq-7000-Overview. pdf," August (2012).

[5] Coseriu, Bogdan, Mihai Negru, and Sergiu Nedevschi. "Contrast restoration of foggy images on the ZYNQ embedded platform." In Intelligent Computer Communication and Processing (ICCP), 2016 IEEE 12th International Conference on, pp. 207-214. IEEE, 2016.

[6] Schwiegelshohn, Fynn, and Michael Hübner. "Design of an attention detection system on the zynq-7000 soc." In ReConFigurable Computing and FPGAs (ReConFig), 2014 International Conference on, pp. 1-6. IEEE, 2014.

[7] Nayak, Rikin J., and Jaiminkumar B. Chavda. "Comparison of Accelerator Coherency Port (ACP) and High Performance Port (HP) for Data Transfer in DDR Memory Using Xilinx ZYNQ SoC." In International Conference on Information and Communication Technology for Intelligent Systems, pp. 94-102. Springer, Cham, 2017.

[8] National Semiconductors, “LVDS Owner’s Manual: LowVoltage Differential Signalling”, pp. 8-22, 3rd edition, 2004.

[9] Barbareschi, Mario, Antonino Mazzeo, and Antonino Vespoli. "Malicious traffic analysis on mobile devices: a hardware solution." International Journal of Big Data Intelligence 2, no. 2 (2015): 117-126.

[10] Intel.com “Signal Parameters”, http://goo.gl/P7wQ5N

[11] Li, S., Jiarong R. Luo, Yichun C. Wu, Guiming M. Li, Feng Wang, and Yong Wang. "Continuous and real-time data acquisition embedded system for EAST." IEEE Transactions on Nuclear Science 57, no. 2 (2010): 696-699.

[12] Demirtas, Mehmet, Ibrahim Sefa, Erdal Irmak, and Ilhami Colak. "Low-cost and high sensitive microcontroller based data acquisition system for renewable energy sources." In Power Electronics, Electrical Drives, Automation and Motion, 2008. SPEEDAM 2008. International Symposium on, pp. 196-199. IEEE, 2008.

[13] Nkom, B., and H. Musa. "Development of a Novel Microcontroller-based data logger." In Adaptive Science & Technology, 2009. ICAST 2009. 2nd International Conference on, pp. 314-324. IEEE, 2009.

[14] Bank, Jason, and Benjamin Kroposki. "Development of a real-time, high-speed distribution level data acquisition system." In Innovative Smart Grid Technologies (ISGT), 2012 IEEE PES, pp. 1-6. IEEE, 2012.

[15] Sivaram, A. T., Masashi Shimanouchi, Howard Maassen, and Robert Jackson. "Tester architecture for the source synchronous bus." In Test Conference, 2004. Proceedings. ITC 2004. International, pp. 738-747. IEEE, 2004.

[16] Raju, Anju P., and Ambika Sekhar. "Implementation of High Speed Distributed Data Acquisition System." International Journal of Advancements in Research & Technology 1, no. 4 (2012): 1-7.

[17] Dostálek, Petr, Jan Dolinay, and Vladimír Vašek. "Simple data acquisition unit for school laboratory exercises." In Carpathian Control Conference (ICCC), 2011 12th International, pp. 85-88. IEEE, 2011.

[18] Edavoor, Pranose J., Sithara Raveendran, and Amol D. Rahulkar. "Implementation of adaptive image compression algorithm using varying bit-length daubechies wavelet coefficient with three-level encryption on Zynq 7000." In Embedded Computing and System Design (ISED), 2017 7th International Symposium on, pp. 1-6. IEEE, 2017.

[19] Xue, Tao, Jinfu Zhu, Guanghua Gong, Liangjun Wei, Yang Luo, and Jianmin Li. "The Design and Data-throughput Performance of Readout Module based on ZYNQ SoC." IEEE Transactions on Nuclear Science (2018).

[20] Silva, João, Valery Sklyarov, and Iouliia Skliarova. "Comparison of on-chip communications in zynq-7000 all programmable systems-on-chip." IEEE Embedded Systems Letters 7, no. 1 (2015): 31-34.

[21] Sklyarov, Valery, and Iouliia Skliarova. "Exploration of high-performance ports in Zynq-7000 devices with different traffic conditions." In Electrical Engineering-Boumerdes (ICEE-B), 2017 5th International Conference on, pp. 1-4. IEEE, 2017.