ISSN:2582-1458
https://doi.org/10.31838/jvcs/03.01.03

Research Article

Hardware/Software Co-Design using ZYNQ SoC

CH.V.S. RAGHUVEERA KUMAR[1], DR. USHA RANI.NELAKUDITI[2]
[1]Ph.D. Scholar, Department of ECE VFSTR, Vadlamudi
[2]Senior Member, IEEE Professor, ECE, VFSTR, Vadlamudi
Email: raghuchandu432@gmail.com[1], usharani.nsai@gmail.com[2]
Received: 02.07.20, Revised: 08.08.21, Accepted: 14.09.21

ABSTRACT
A Programmable Logic (PL) (FPGA) and Processing Subsystem (PS) (ARM Cortex-A9) make up the Xilinx ZYNQ-7000 SoC.. The data transfer arrangement between the PL and PS is an important part of the ZYNQ Architecture. In comparison to other existing solutions, the AXI Interconnect serves as a vital communication link between PL and PS for bi-directional data transfer. This paper explores configuration between the PL and PS with in the ZYNQ-7000 SoC. Implemented a logic with configuration of both PS and PL and only PL.

Keywords: Xilinx, Zynq 7000 Soc, FPGA, Arm Cortex-A9, AXI Interconnect

Introduction

Prior to the Zynq, processors were connected to a Field Programmable Gate Array (FPGA), which complicated communication between the Programmable Logic (PL) and the Processing System (PS). The Xilinx ZYNQ-7000 SoC is the most recent addition to the new invention, and it provides an alternate platform for traditional SoC and ASIC users. The ZYNQ-7000 SoC is a new class of product and the first in the industry to combine an industry-standard application processor Cortex A9 dual-core Processing System (PS) from ARM and Xilinx 28nm Programmable Logic (PL) on the same chip, combining the performance, fully programmable nature, and power savings of hard intellectual property (ARM IP) with the flexibility of programmable logic. (FPGA).

In comparison to Telemetry Systems, the ZYNQ SoC-based secure data transmission implementation in an embedded system indicates that ZYNQ SoC-based implementation requires less space and weight. Major considerations such as area and weight may have an impact on overall system performance.

In any embedded system, especially in mission-critical applications, high performance and low latency are vital. The ZYNQ SoC provides shared memory access, allowing for improved performance and lower latency in Embedded Multiprocessor architectures [2]. The PLB bus appears to be a viable communication channel between the ARM and the FPGA in a ZYNQ- based Multi-Core Embedded Systems system. Also demonstrates an increase in communication speed [3].

The internal architecture of the ZYNQ-7000 allows custom IPs and logics to be implemented in the PL. It also allows for the installation of bespoke software in the PS. It makes developing distinct and differentiated functional systems much easier. Because of their limited amount of I/O as well as bandwidth, latency, and power needs, multi-chip based solutions (e.g., an FPGA with an ASSP) cannot provide the same degree of performance and dependability.

Configuration Between Pl And Ps

ZYNQ SoC devices employ a variety of communication interface-based interconnect approaches that are tuned and tailored to the functional blocks' specific communication requirements. All of the connections available to the system designer are included in the PL to PS interface. [5] will be the classification for these interfaces.

- Functional Interfaces: Interconnect AXI, DMA controller, extended MIO interfaces (EMIO) for most I/O peripherals, interrupts, timers, and debug interfaces are all included. In PL, the functional interface signals can be used to link user-designed IP blocks.
- Configuration Signals: Contains the processor configuration access port (PCAP), configuration status, single event upset (SEU) and Program/Done/Init. Configuration signals are connected to fixed logic in the PL configuration block, providing PS control.

  Transmission logic/interfaces are a sort of functional interface, and the AXI interconnect is the most common interface for data communication between PL and PS. The ZYNQ SoC has 9 independent PS-to-PL AXI

Interconnect based Interfaces, allowing for data transmission speeds of up to 100Gbps.

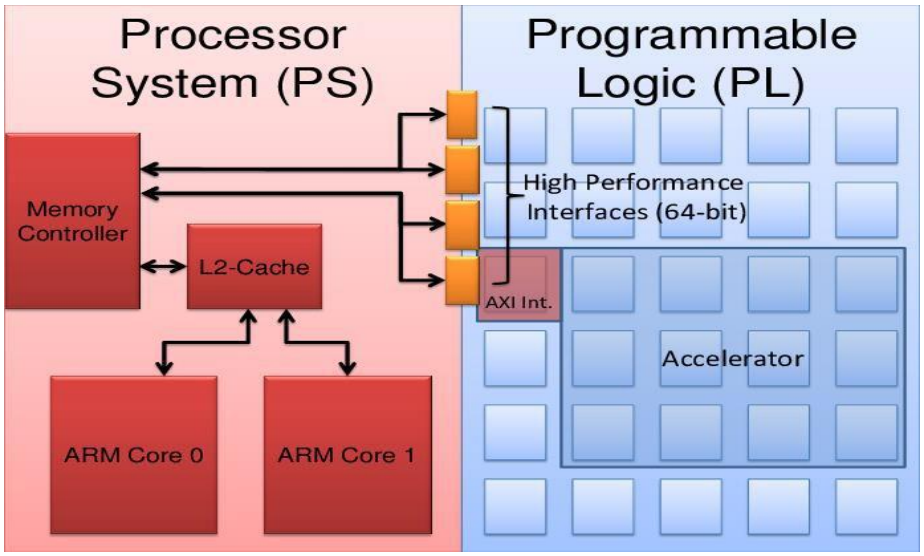

Fig.1: AXI interconnect interfaces

The above figure shows the interconnection with the PS and PL. In this architecture shows the high performance interface which means very speed.

ZYNQ Hardware and Software Design Flow

The below figure briefly shows the overall steps involved in executing the ZYNQ based designs, which includes both hardware and software design phases.

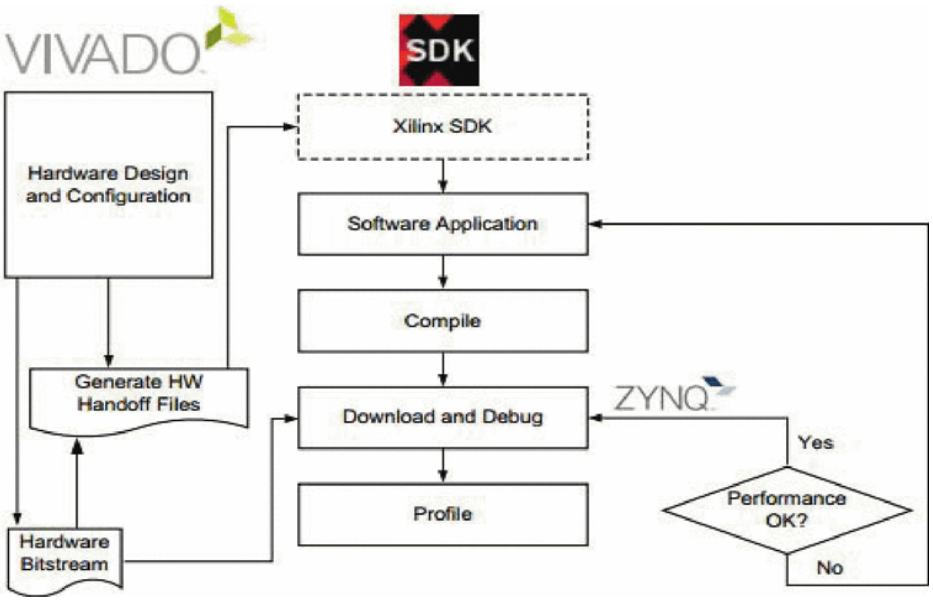

Fig.2: ZYNQ hardware and software design flow

Some steps in the design flow for the Zynq architecture are similar to those in a standard FPGA. The first step is to define the system's specs and requirements. The different tasks (functions) are then assigned to implementation in either PL or PS during the system design stage, which is known as task partitioning. This stage is critical since the overall system's performance is dependent on tasks/functions being assigned to the most appropriate technology: hardware or software.

The next step is to develop and test the hardware and software. In terms of the PL, the objective is to identify the required functional blocks in order to accomplish the design criteria, as well as to construct them as IPs and connect them appropriately. Similarly, the software activity entails writing code for the PlayStation. As a result, system integration and testing are necessary to complete the design. The Zynq SoC design pipeline is depicted in Figure 2.

For software the program was written in SDK which means the Software Development Kit. In that wite the program in C language by taking the new application system then we have to load the bitmap file which is shown in above figure

Implementation

In this implementation NAND gate was implemented with the both PS with PL and only PL the floor plan was shown in figure 3 and 4 which is shown in below figure

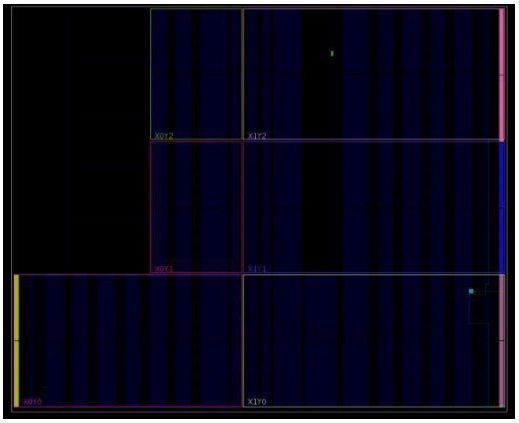

Fig.3: Floor plan of only PL section

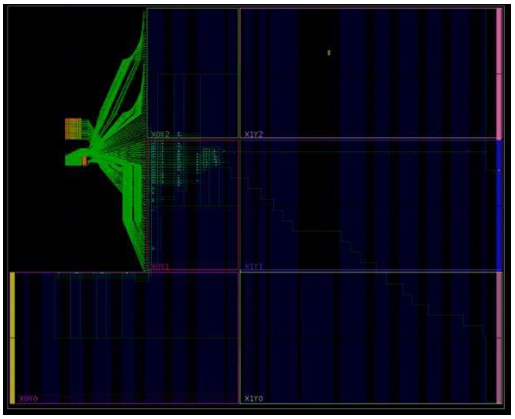

Fig.4: Floor plan of PL with PS section

Here it is configured PS and PL by a simple logic and compared the results between the only hardware and combing with the both hardware and software design.

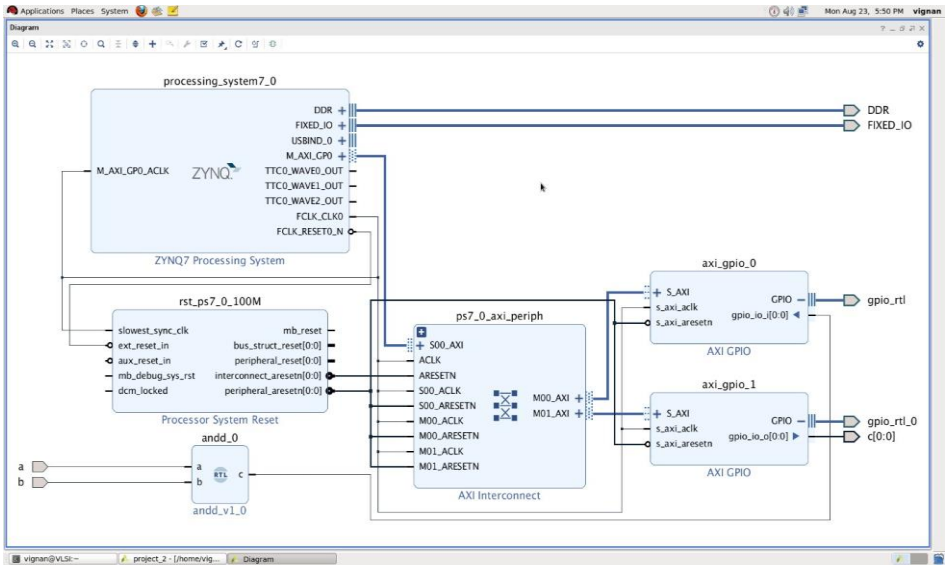

Fig.5: Implemented Nand logic by using block diagram

The written of AND gate in PL is regular method. It can write in both language either in VHDL or in Verilog. But in PS configuration the code should wite in either in C or C++ language.

Here the logic AND gate was implemented in PL which means in hardware section and logic NOT gate implemented in PS which means in software section. Where in hardware section the code was written in Verilog and in the section of software the code was written in C language.

By observing the above figures 3&4 can say that the one CLB used for PL configuration but in the PL with PS configuration so many CLBs are used and processing system also used for speed.

Verification

Verified the both the programmes which means the PL configuration and PS with PL configuratin outputs are verified by writing the costrain file. The below figures shows the test cases of the logic gates

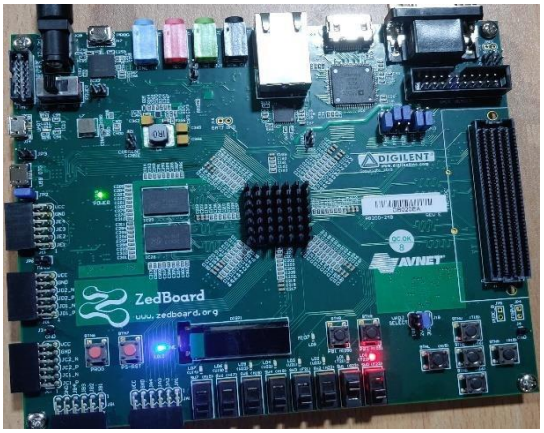

Fig.6: Both inputs are low

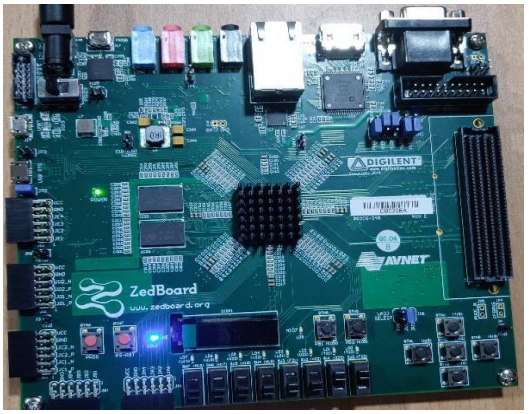

Fig.7: Both inputs are high

The connection was done in such a way that the output of AND gate is the input of NOT gate. It is achieved by using the AXI GPIO interface. In this two AXI GPIO are used for the both the input and outputs. The output of AND gate was connected to the input of processing system. In this AXI GPIO_0 act as the input which is shown in figure2.

The AXI GPIO_1 act as the output of the processing system. Here the output was taken. But here one essential point to be noted is the name of a port it should match to the constrain file. The declaration of input ports are very important it should match with the constrain otherwise it will throw an error. By combing both the gates it can achieve the NAND gate. Now the board work as a NAND gate by using both the PL and PS.

Results

In this section results are discussed in the both PL and combination of PS and PL configuration.

Table 1: Delays of both PS with configure PL and Only PL

		PS with PL	Only PL
Set up	Total delay	9.725	10.943
	Logic delay	3.189	3.657
	Net delay	6.536	7.286
Hold	Total delay	3.960	3.519
	Logic delay	1.294	1.351
	Net delay	2.666	2.169

The above table shows the various delays of logic. As implemented only NAND gate that too single bit, it shows the difference if it implemented in a high level logic definitely it will have huge various.

Conclusion & Future Scope

The above evidence shows that the PS configuration with PL is fast then the only PL. Here it is implemented for single bit, if it is implemented in multi bit and more complex the variation will be more.

Now implemented only NAND logic gate. In future high level logic circuits are implemented in PL and PS with the PL configuration. Then we can really enjoy the fruit of PS with PL configuration.

References

1. D. Dhana Laxmi “Implementation of secured data transmission system on customized Zynq SoC” IJSR Journal 2012.
2. Yang Nie, “Design and Implementation of Inter-core communication of Embedded Multi processor based on share memory”, IJSIA Journal, 2016
3. Yang Nie, “Research on the Design of multicore Embedded System based on micro blaze.” IJCA Journal 2016.
4. “Design of High-Speed Data Acquisition system based on FPGA” Yu Yanuin, Amm Journal,2014
5. UG585-Zynq-7000 AP SoC Technical Reference Manual Xilinx, Sept-2016.
6. R.F.Cordeiro, A. Prata, A. S. R. Oliveira, N. B. Carvalho and J. N. Vieira, "FPGA-based all-digital software defined radio system demonstration," 2015 25th International Conference on Field Programmable Logic and Applications (FPL), London, 2015, pp. 1-1.
7. S. M. Wright, M. P. McDougall, K. Feng, N. A. Hollingsworth, J. C. Bosshard and C. W. Chang, "Highly parallel transmit/receive systems for dynamic MRI," 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Minneapolis, MN, 2009, pp. 4053-4056.

8. J. F. Villena et al., "Fast Electromagnetic Analysis of MRI Transmit RF Coils Based on Accelerated Integral Equation Methods," in IEEE Transactions on Biomedical Engineering, vol. 63, no. 11, pp. 2250-2261, Nov. 2016.
9. A. Hassibi, A. Babakhani and A. Hajimiri, "A Spectral-Scanning Nuclear Magnetic Resonance Imaging (MRI) Transceiver," in IEEE Journal of Solid-State Circuits, vol. 44, no. 6, pp. 1805-1813, June 2009.
10. Z. Liu, C. Zhao, H. Zhou and H. Feng, "A Novel Digital Magnetic Resonance Imaging Spectrometer," 2006 International Conference of the IEEE Engineering in Medicine and Biology Society, New York, NY, 2006, pp. 280-283.
11. www.analog.com/media/en/technical-documentation/datasheets/AD9122/
12. www.xilinx.com/products/boards-and-kits/ek-z7-zc702-g.html/
13. www.xilinx.com/support/documentation/.../zc702.../ug850-zc702-evalbd.pdf/
14. www.google.co.in/mriquestions.com/Fradiofrequency-waves.html/
15. www.google.co.in/mriquestions.com/why-precession.html/