Indonesian Journal of Electrical Engineering and Computer Science
Vol. 13, No. 2, February 2019, pp. 485~491
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v13.i2.pp485-491

Implementation of an ARM-Based system using a Xilinx ZYNQ SoC

Omar Salem Baans, Asral Bahari Jambek
School of Microelectronic Engineering, Universiti Malaysia Perlis, Perlis, Malaysia

Article Info

Article history:

Received Sep 24, 2018
Revised Nov 25, 2018
Accepted Dec 9, 2018

Keywords:

ARM
Cortex-A9
SDK
SoC
Vivado
ZYNQ board

ABSTRACT

ARM processors are widely used in embedded systems. They are often implemented as microcontrollers, field-programmable gate arrays (FPGAs) or systems-on-chip. In this paper, a variety of ARM processor platform implementations are reviewed, such as implementation into a microcontroller, a system-on-chip and a hybrid ARM-FPGA platform. Furthermore, the implementation of a specific ARM processor, the Cortex-A9 processor, into a system-on-chip (SoC) on an FPGA is discussed using Xilinx's Vivado and SDK software system and execution on a Xilinx Zynq Board.

Corresponding Author:

Omar Salem Baans,
School of Microelectronic Engineering,
Universiti Malaysia Perlis, Perlis, Malaysia.
Email: omersalim4901@gmail.com

1. INTRODUCTION

Currently, embedded systems are used in various applications such as automotive, consumer electronics and system control. Embedded systems perform dedicated functions and are normally small and cheap. In developing an embedded system for a specific application, the main design considerations are the system hardware, the application software and the cost. The specifications of the hardware include the system design, operation speed, memory, communication interfaces and power consumption. An embedded system can be implemented as a microcontroller chip, as in [1] and [2]. However, an embedded system can also be developed into a system-on-chip (SoC) for higher performance [3-4].

For improved flexibility, an embedded system may be implemented into a field-programmable gate array (FPGA) chip. One of the popular processor types employed in embedded systems is the ARM processor. The ARM processor is widely utilised by researchers in SoC development [3–7].

This paper is organised as follows. Section 2 discusses several microprocessor-based systems, while Section 3 presents the design methodology in implementing an ARM Cortex-M9 processor-based system in an FPGA. In Section 4, the experimental results are analysed and discussed. Finally, Section 5 concludes the paper.

There have been many studies on the implementation of ARM processors into FPGAs. This section will discuss and elaborate on the existing designs. The Altera SoC is one example of a heterogeneous multiprocessor (Hetero-MP) [8-9], as illustrated in Figure 1. It is integrated with an ARM-based hard-core processor and an FPGA fabric. Several Nios II soft-core processors can be implemented into the FPGA and communicate with the ARM processor.

Another FPGA board that implemented an Hetero-MP was discussed in [10]. A Xilinx Virtex5 ML505 board [11] was used for the development of the architecture as shown in Figure 2. The FPGA was based on a processing unit that included 3 PU architectures: PU1 realised the microarray image enhancement;

PU2 computed the microarray image addressing; and PU3 used spatial parallelism for image segmentation. The hardware architecture is shown in Figure 3. Each custom processing element from the proposed design was connected to a fast simple link (FSL) data bus as a coprocessor for the soft-core MicroBlaze 100-MHz microprocessor.

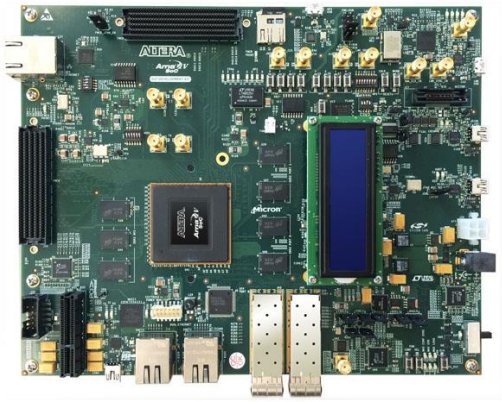

Figure 1. Arria® V SoC Development Kit from Altera [8]

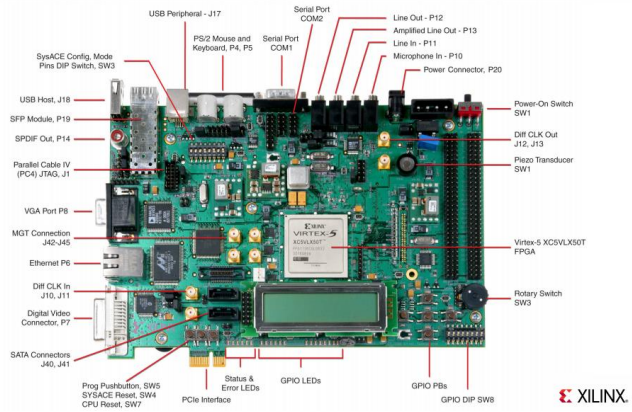

Figure 2. Xilinx board Virtex5 ML505 [11]

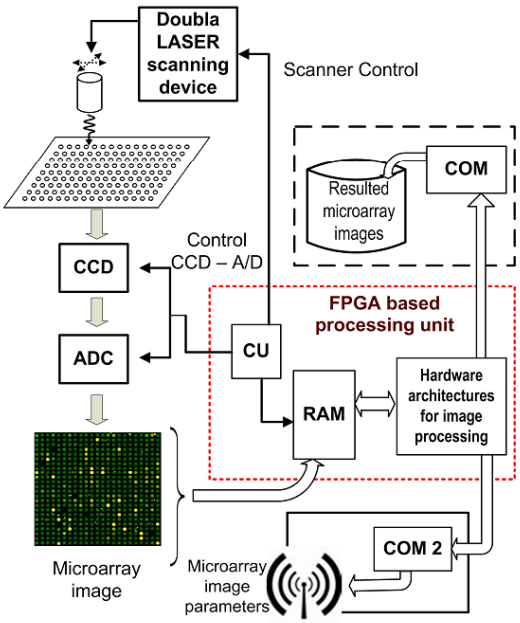

Figure 3. System design block diagram for [10]

In [12], the authors proposed an embedded robot controller based on an ARM and an FPGA. The controller was implemented using a high-speed and high-processing-capability ARM microcontroller and a flexible and parallel-computing FPGA. Figure 4 shows the main structure of the controller, where a flexible static memory controller was used to connect the ARM microcontroller with the FPGA. Additional components used in the controller system included a joint test action group (JTAG), a universal synchronous asynchronous receiver transmitter (USART), a universal serial bus (USB) and a general-purpose input and output (GPIO). The ARM microcontroller was used as the main controller, while the FPGA was used to control the servo motor. The robot controller was implemented and tested on a 6-degree-of-freedom robot arm.

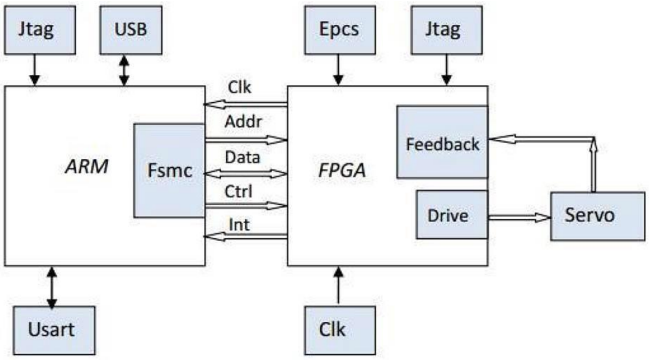

Figure 4. Main structure of the controller [12]

In [13] and [14], a BioServer was utilised that consisted of two separate physical boards. The first part uses a Xilinx ML510 Embedded Development Platform. This was the basis for the embedded system with two PowerPC 440 microprocessors called BioSys. The second part was called the Biometrical Computation Unit (BioCU), and its main components included four digital signal processors, two of which were fixed-point (CPU0 and CPU1) and two of which were floating-point (FPU0 and FPU1), as shown in Figure 5. The details of the BioServer device are presented in Figure 6. The BioCU board was inserted into one of the PCI 32-bit slots on the ML510 BioSys platform.

Figure 5. View of the BioServer device [13-14]

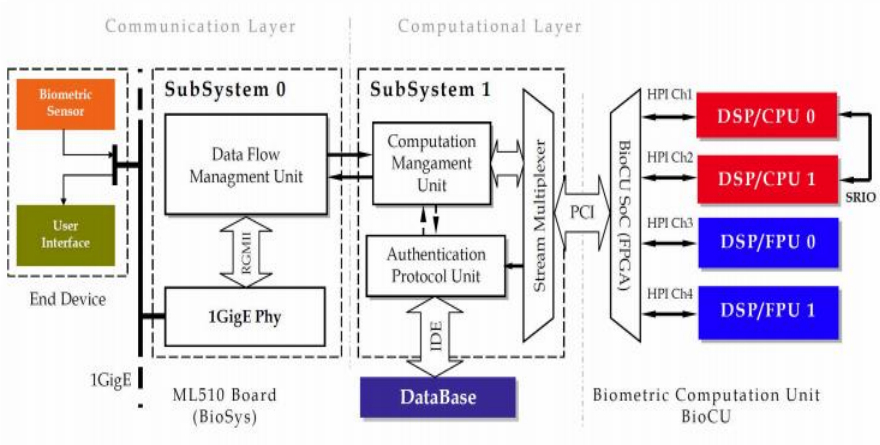

Figure 6. Block diagram of the BioServer device [13-14]

Xilinx All-Programmable SoCs (AP SoCs) are processor-centric platforms that offer software, hardware and I/O programmability in a single chip [15]. The Zynq-7000 family is based on AP SoC architecture. The architecture is divided into two parts: the processing system (PS) and the programmable logic (PL), as shown in Figure 7. The PS is the main processing unit and includes components such as an ARM Cortex-A9 processor, on-chip memory and various peripherals. The PL consists of a number of hardware accelerators and a partial reconfiguration region (PRR) controller. Between the PS and PL, an interconnection is used to allow the system to communicate.

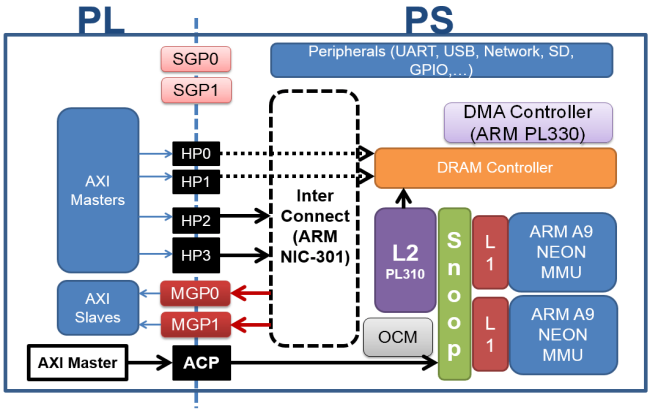

Figure 7. Architecture of a Xilinx ZYNQ board [15]

Several existing processor-based systems have been reviewed and discussed in this section. Table 1 summarises the findings. From Table 1, it can be seen that an ARM processor was used in [8-9], [12] and [15]. The ARM in [15] had a dual ARM core, while the [8-9] and [12] boards had only one ARM core, other than the Nios II processor.

Table 1. Comparison of Existing ARM Processor Based System

Features	Year	Architecture	Platform	Processor	Software
[8], [9]	2015	Microcontroller	Altera Board DE2-70	ARM Cortex-A9 and Nios II	Quartz
[10], [11]	2011, 2012	Microcontroller	Xilinx Board Virtex5 ML505	Soft-core Micro Blaze microprocessor	Xilinx
[12]	2014	Microcontroller	ARM and FPGA Development Board	ARM Cortex M3	STMicroelectronic
[13], [14]	2010, 2011	FPGA & DSP	Bioserver	-	Xilinx & DSP
[15]	2015	Hard processor system	Xilinx ZC702	Dual core ARM Cortex-A9	Xilinx

2. RESEARCH METHOD

In this section, the method of implementing an ARM processor-based system will be explained. In the hardware implementation, the system components including an ARM Cortex-A9 processor are integrated into the system. The system architecture, along with the Hex file, is compiled using Xilinx Vivado software. To implement the systems into an FPGA, the input and output for the system are set. Once the compilation is completed, the architecture and the Hex file are downloaded using a Xilinx ZYNQ FPGA chip on a ZC702 board. When the system is executing on the FPGA, the experiment result can be observed. The system block diagram consists of two main parts, the PS and PL, as shown in Figure 8. In this system, the PS is an ARM Cortex-A9 and 64k-bit memory while the PL part consists of a GPIO interface and input and output ports.

Next, the HDL wrapper is created before the bitstream of the system is downloaded onto the FPGA board. Finally, Xilinx SDK software is used to create and execute the C programming code into the ARM processor. The program controls the blinking of the LED using a button switch on the board. At the same time, it can perform addition for any two numbers the user saves into a created register in the FPGA.

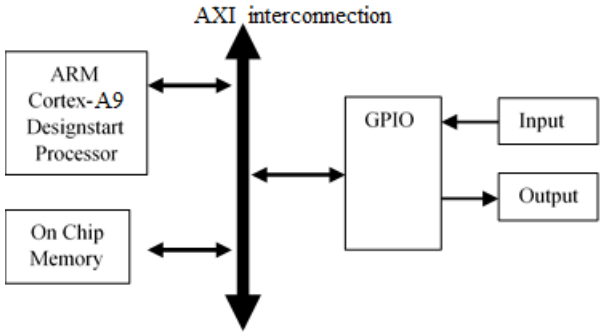

Figure 8. Block diagram of the ARM processor system

3. RESULTS AND ANALYSIS

In this section, the results of the experiment will be discussed. The design file was successfully downloaded into a Xilinx ZYNQ ZC702 FPGA board. The on-chip memory of the system was loaded with an LED toggle program and a half-adder function. In the system architecture, the GPIO was connected to an on-board LED and adder. These programs were simulated using a laptop computer (Intel(R) Core(TM) i3 processor, 1.80 GHz, 6.00 GB RAM).

Figure 9 shows the simulation results for adding A, B and C in hexadecimal numbers in three sequence periods. For example, in the first 5 nanoseconds, A = 1, B = 19 (13 in hex) and C = 0, so the result was 20 (14 in hex). The same was true for the other two outputs.

Figure 10 shows the SDK terminal that was connected through a UART cable from the board. The screen terminal showed the results, as seen in Figure 10. The screen terminal shows the results of adding A, B and C in hexadecimal numbers, where the first line was A = 1, B = 19 and C = 0. This gave a result of 20. The same was true for the other two outputs. Finally, the LED results on the board showed increasing values as reflected on the LED when the user pressed the button. These are shown in Figure 11 and 12. The same decimal values of that number could be shown on the SDK terminal on the PC screen, as shown in Figure 13.

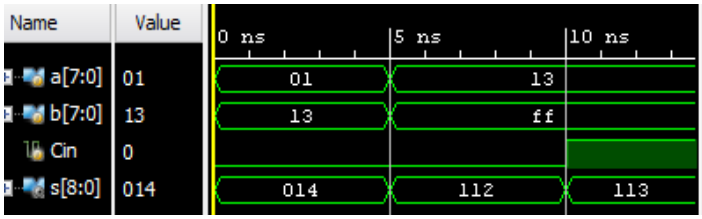

Figure 9. The simulation results for adding A, B and C in hexadecimal numbers

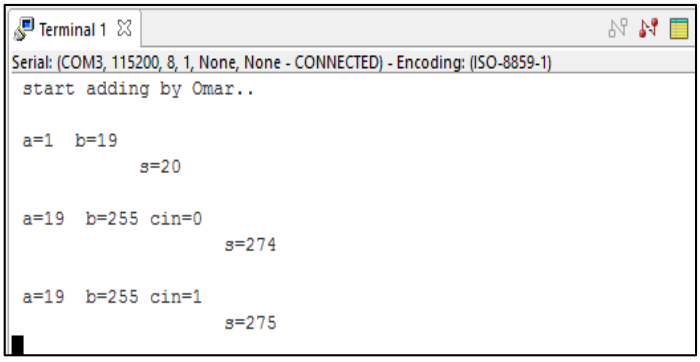

Figure 10. The screen terminal shows the results of adding A, B and C in hexadecimal numbers

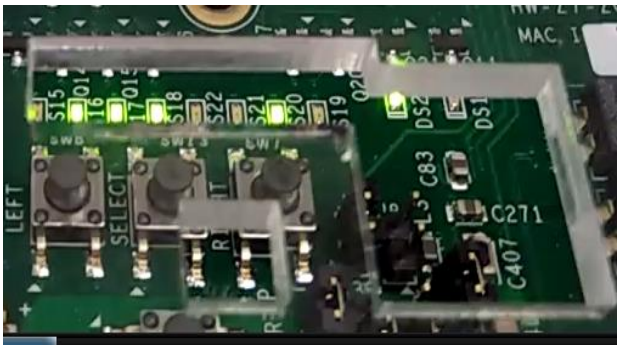

Figure 11. The LED results on the board showed increasing values as reflected on the LED when the user pressed the button

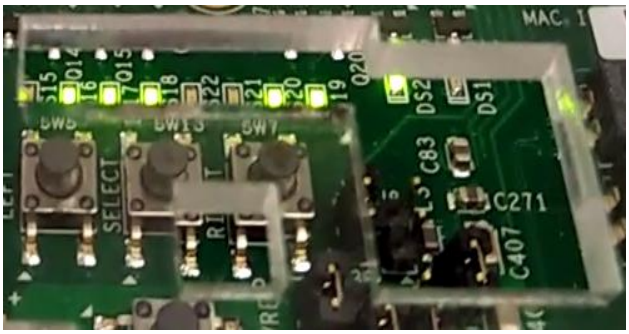

Figure 12. The LED results on the board showed increasing values as reflected on the LED when the user pressed the button

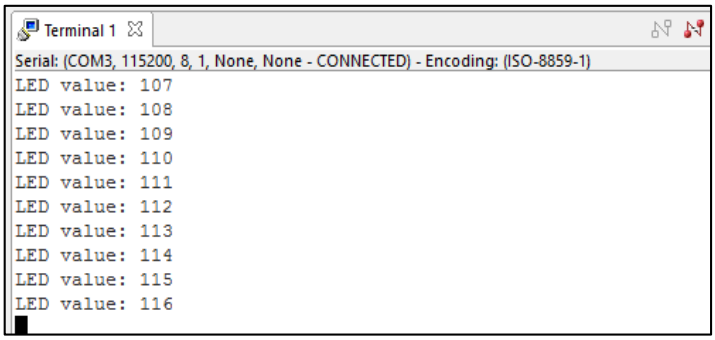

Figure 13. The equivalent decimal values shown on the sdk terminal

4. CONCLUSION

In this paper, a method of implementing an ARM microprocessor system was presented and discussed. An ARM microprocessor system was designed using Vivado 2016.4 and SDK software. The ARM Cortex-A9 processor is used in the microprocessor system is. In this system, the AXI interconnection buses were used to connect the ARM processor with other components. The application program was compiled and assembled into Verilog code using an ARM tool chain. Lastly, the microprocessor was successfully implemented into a Xilinx ZYNQ board for demonstration.

ACKNOWLEDGEMENTS

The author would like to acknowledge the support from the Sciencefund under a grant number of 03-01-15-SF0229 from the Ministry of Science, Technology & Innovation, Malaysia.

REFERENCES

[1] A. Peter, R. K. Karne, and A. L. Wijesinha, "A bare machine sensor application for an ARM processor," in *IEEE International Conference on Electro-Information Technology*, EIT 2013, 2013, no. 2, pp. 1–6.

[2] A. Bhargava and R. S. Ochawar, "Biometric access control implementation using 32 bit arm cortex processor," *Proc. - Int. Conf. Electron. Syst. Signal Process. Comput. Technol. ICESC 2014*, pp. 40–46, 2014.

[3] N. Van Helleputte, M. Konijnenburg *et. al.*, "A 345 µW Multi-Sensor Biomedical SoC With Bio-Impedance, 3-Channel ECG, Motion Artifact Reduction, and Integrated DSP," *IEEE J. Solid-State Circuits*, vol. 50, no. 1, pp. 230–244, Jan. 2015.

[4] T. Xia, J. Prévotet, and F. Nouvel, "An ARM-based Microkernel on Reconfigurable Zynq-7000 Platform," *Mediterr. Telecommun. J.*, vol. 5, no. 2, pp. 109–115, 2015.

[5] Y. Ando, Y. Ishida, S. Honda, *et. al.*, "Automatic Synthesis of Inter-heterogeneous-processor Communication for Programmable System-on-chip," *Inf. Media Technol.*, vol. 10, no. 3, pp. 415–419, 2015.

[6] D. Flynn, T. Wood, P. Dworsky, *et. al.*, "Teaching IC design with the ARM Cortex-M0 DesignStart processor and Synopsys 90nm Educational Design Kit," *Proc. 3rd Interdiscip. Eng. Des. Educ. Conf. IEDEC 2013*, pp. 36–38, 2013.

[7] J. Yiu, "ARM ® Cortex ® -M Processor based System Prototyping on FPGA," in *Embedded World Conference 2014*, 2014, pp. 1–7.

[8] Arria® V SoC Development Kit and Intel® SoC FPGA Embedded Development Suite (https://www.altera.com/products/boards_and_kits/dev-kits/altera/kit-arria-v-soc.html)

[9] Arria V SoC Development Kit User Guide.

[10] Belean, Bogdan, *et al.*, "FPGA based system for automatic cDNA microarray image processing." Computerized Medical Imaging and Graphics 36.5 (2012): 419-429.

[11] ML505/ML506/M L507 Evaluation Platform User Guide, UG347 (v3.1.2) May 16, 2011.

[12] X. He, Z. Wang, H. Fang, *et. al.*, "An embedded robot controller based on ARM and FPGA," in *2014 4th IEEE International Conference on Information Science and Technology*, 2014, pp. 702–705.

[13] Grabowski, Kamil, and Andrzej Napieralski. "Hardware architecture for advanced image processing." *Nuclear Science Symposium Conference Record (NSS/MIC)*, 2010 IEEE. IEEE, 2010.

[14] Grabowski, Kamil, and Andrzej Napieralski. "Hardware architecture optimized for iris recognition." Circuits and Systems for Video Technology, IEEE Transactions on 21.9 (2011): 1293-1303.

[15] ZC702 Evaluation Board for the Zynq-7000 XC7Z020 All Programmable SoC User Guide

BIOGRAPHIES OF AUTHORS

Omar Salem Nasser Baans is currently a Master student in School of Microelectronics Engineering, University Malaysia Perlis. He has his B.Eng. degree from International Islamic University Malaysia. From June 2014 to August 2014, he was a trainee at Fehm Technology Sdn. Bhd. in Kuala Lumpur for his internship training programme. His research project is developing system-on-chip image processing algorithm and architecture for DNA microarray analyser.

Asral Bahari Jambek received B.Eng. Hons. degree in Electronics Engineering from University of Southampton, United Kingdom, in 1998, Master of Science in Electronics Engineering from University Putra Malaysia, Malaysia, in 2002 and Ph.D in Electronics, University of Edinburgh, United Kingdom, in 2008. In 1998, he served in MIMOS Berhad, as research engineer. Since 2008, he joined University Malaysia Perlis, Malaysia as a senior lecturer in School of Microelectronics Engineering. Currently, he is an associate professor in Universiti Malaysia Perlis. His research interests include VLSI circuits and systems, digital signal processing (DSP), analogue and mixed signal (AMS) design, low power algorithms and architectures, and image and video processing.