International Journal of Advances in Computer Science and Its Applications– IJCSIA
Volume 4: Issue 2 *[ISSN: 2250-3765]*
Publication Date : 25 June 2014

Software for automated SoC design based on Hardware/Software CoDesign Concept

Maman Abdurohman[1], Adiwijaya, Gandeva Bayu
Unified Communication Laboratory
Telkom University
Bandung, Indonesia

Arif Sasongko[2], Ricky Rawung
School of Electrical Engineering and Informatics
Bandung Institute of Technology
Bandung, Indonesia

***Abstract*—Productivity gap on SoC design is a challenge to address in design method research area. Main research topics in this area address this problem using many schemes and methods. Software for automation in a system design and implementation is one solution to address design time efficiency. The second one is parallelism concept in hardware/software co-design. This approach addresses both design time efficiency and flexibility. This paper proposes an automated SoC design software with respect to parallelism and automation concept. Hardware/software co-design is an implementation of parallelism concept. In the traditional design, the main concept is hardware design and then software design. In this paper we did both design in parallel mode. The result showed that the proposed method can reduce much more time than before**

***Keywords*—productivity gap, SoC design, hardware/software, codesign**

I. Introduction

There are three important aspects that affect SoC design and implementation, i.e. Integrated Circuit(IC) technology, processor and design technology [12]. Moore law shows the accurate prediction with respect to the capacity of IC technology. Its capacity is doubling every 18 months [8]. The fact shows the capacity of processor chip doubled in few years. The growth of IC technology was irreversible. It constantly increases inline to the market necessity. On the other side, productivity of hardware designers was not as fast as IC technology growth.

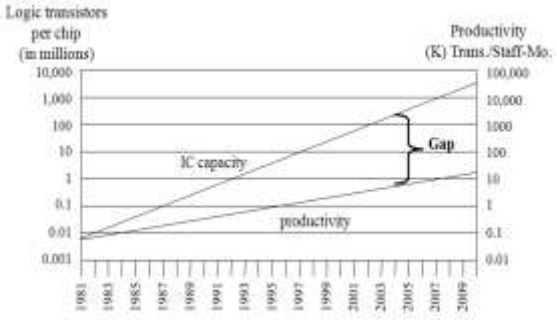

Figure 1. Productivity Gap[12]

There is a wide gap between technology and productivity. It means that there is an opportunity lost on IC technology usability. It is quite understandable because of the irreversible nature of IC technology. It will continue to evolve and never return. However, the designers are still dominated by those who have a productive age limit. Designers who have advanced skill will eventually disappear along with the course of time. Transfer of skills is done by knowledge transfer and research collaboration.

On the other hand, the right thing to do is to reform the technological design. The idea is to construct SoC design framework that accelerates designing process. That will maximize the productivity.

SoC framework consists of two parts, i.e. hardware and software. Traditional approach was conducted by designing hardware and software separately. That approach takes longer time because of the serial process. The improvement of that approach was carried out by designing both hardware and software together in parallel, which is by hardware/software co-design concept. The second approach to do is developing tool for time efficiency. Mapping processes between source code and object can be simplified by using automatic generator.

II. Hardware/Software CoDesign Method

Efforts to improve the performance of design technology have been carried out. The fundamental concept used to improve design technology is by enhancing abstraction level, such as Architectural model[4], High Level Synthesis[2], System level Synthesis[3], Transaction Level Modeling[6], System Level Design[7] and System Level Modeling[10]. The second one is parallelism concept such as Hardware/software co-design model[9].

Designers concern to improve the design performance for addressing the productivity of gap problem. In this paper, hardware / software co-design, as a basic concept is used to increase performance of the SoC design process. The used approach in the model of hardware/software codesign is parallelism concept.

SEEK DIGITAL LIBRARY

There are two major sub-systems in SoC implementation : hardware and software. Many functions and processes implemented as part of hardware, as well as software. The different feature between hardware and software lies in the way of operating. Basically all the process can be implemented as both hardware and software. Depend on the user requirement. Each has advantages and disadvantages. Hardware has a good speed while software has a low-cost.

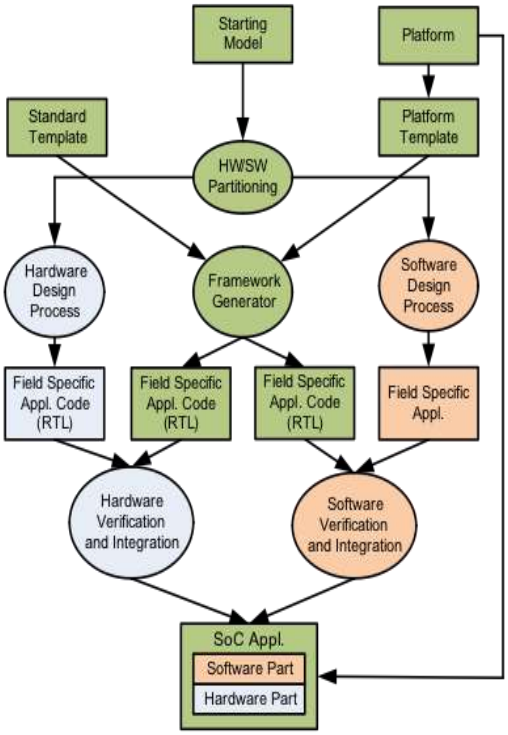

Figure 2. Hardware/Software CoDesign Process

In hardware/software codesign, the designing process is conducted in parallel. In the beginning, there is a hardware/ software partitioning which is aim to divide the software and hardware parts. RTL code is designed in hardware parts, while the function is designed in software parts. Thus, in the end of the process, the combination of both is conducted.

Starting model of design requires many characteristics. The model should be simple as easy as possible to be understood. So, many kind people with various background can discussed it. It is important to transform user/project owner requirement to the model accurately.

Complex model will rise the difficulty in the first step and often leads to inconsistence respect to user requirement. The second is that the model has to fit the syntax and format for automated generation of the framework. Lack of preciseness may lead to ambiguities. The third is that the model has to support modularity. Each part of the system can be designed independently. Engineers can design many function/module in the same time.

III.Software of SoC Automated Generator

The second approach in improving the performance of the design is the automation of the design process. Software development is conducted to replace the manual process. This approach is commonly done to speed up the process.

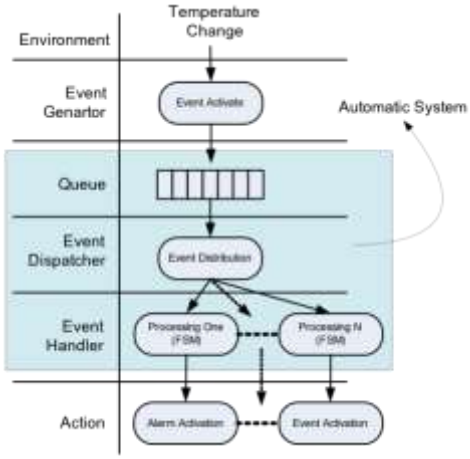

Figure 3. Automation Framework

There are library templates that are designed as an enabler for the framework generator. The library is called when executing the design. The main process performed by the automation software is read in a state chart design to be transformed into C framework files. These results are used for SoC implementation which consists of hardware and software parts.

The basic process in the development of software for design automation starts with handling events coming from the external system. These data queue for dispatching to suitable handler. Handler send action signal to output system. There are many output systems depend on definition of the system. Calculator application, for example, shows the result of calculation on the screen or seven segments. Temperature control system send action signal to control air conditioner turn on/off.

Code generation process starts from the defined description of state chart and templates in the source. Results from the system are the C files which is used for system configuration and the main file that is the implementation of the FSM. The last but not least are header files.

Frameworks that produced by the system configuration files are independent to the system. The others files are system file itself. System file that created in this software are hardware part and software part files. This manner of implementation shows the hardware/software codesign concept.

The first step in the system is read configuration file. There are many definition in the configuration file for configuring the system. Next step is initiation all of FSMs, states and events. After all function initiated then declaration will finished. Another process is creating FSM.c files. After all

SEEK DIGITAL LIBRARY

FSMs and states are executed then the result combine with declaration files.

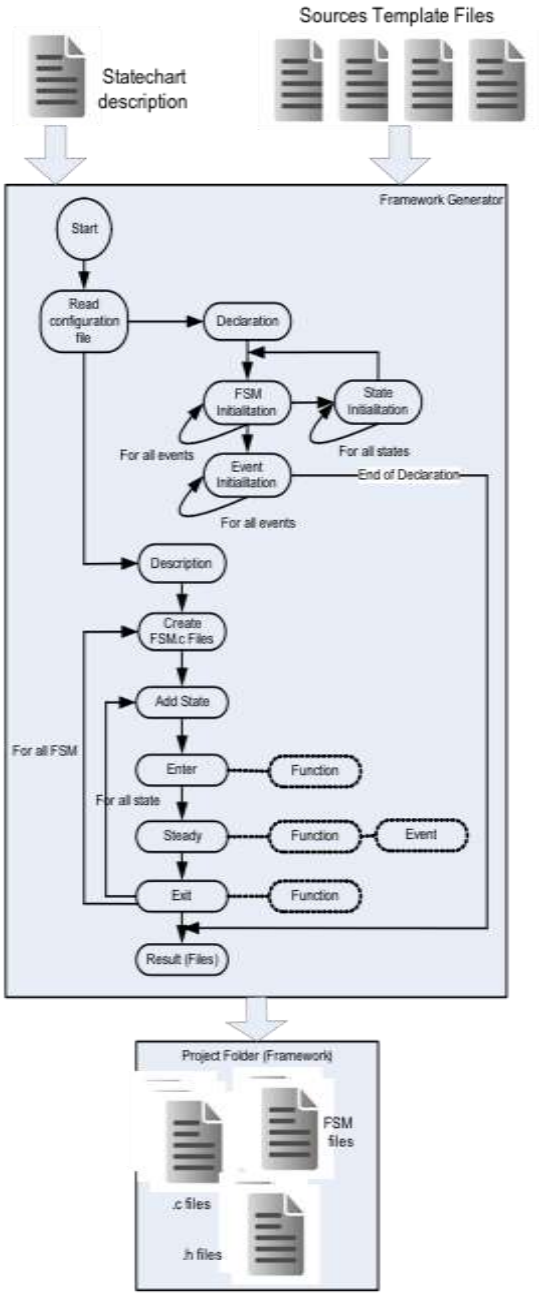

Figure 4. Software Process Diagram

In the end of the process, there are three kinds files as the result of the system such as FSM files, .c fileas and .h files.

IV. Experiment and Result

In the system, a program for calculator has been created. The followings are the files resulted from using hardware/software co-design approach. Calculator is the model application to prove the efficacy of SoC automation application. In calculator, there are some function unit and process. Function units, such as subtraction, adder, and multiplier, are implemented as hardware. Supporting functions, such as addButton, subButton, and mulButton are implemented as software.

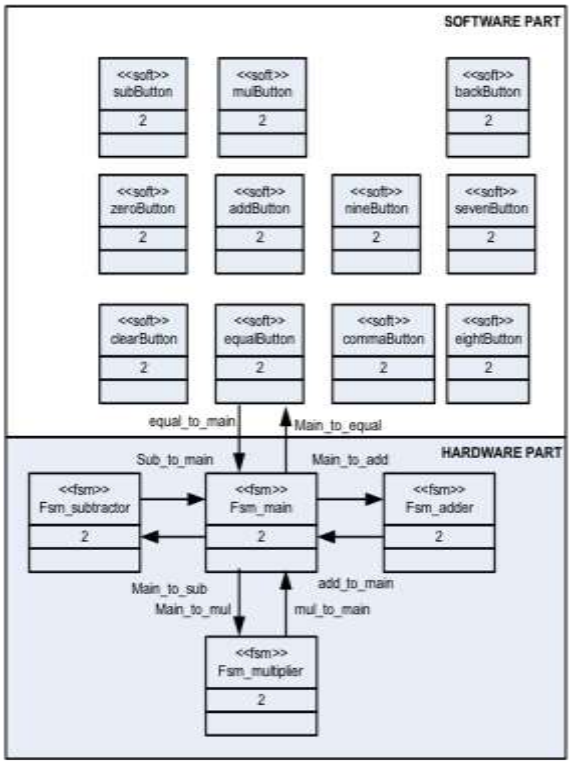

Figure 5. Hardware and Software Implementation

The main processes developed in calculator are addition, subtraction, and multiplication. These three processes require higher processing rapidity than other processes'. At the stage of hardware/software partitioning, those three are implemented as hardware and the rest are as software. The consideration of hardware and software separation needs to be adjusted to user requirement.

System that needs higher processing rapidity requires more hardware implementation than the saving system. Saving here means needing low power consumption and low production cost. Consequently, the rapidity will be lower than the hardware-oriented system.

The analysis of system efficacy is focused on the implementation of parallelism and automation concept in SoC design to improve design technology performance. There are, at least, two items addressed by using this approach. Parallelism concept that is implemented with hardware/software co-design can improve design efficacy and flexibility, which is in term of time efficiency and flexibility to meet user requirement. System can be designed in full hardware or hybrid between hardware and software part. Second one is software for automated SoC framework generator addressing the time consumption of design process. There are many overhead processes, and even though some cases create error bug, but they are still in the rational time frame.

SEEK DIGITAL LIBRARY

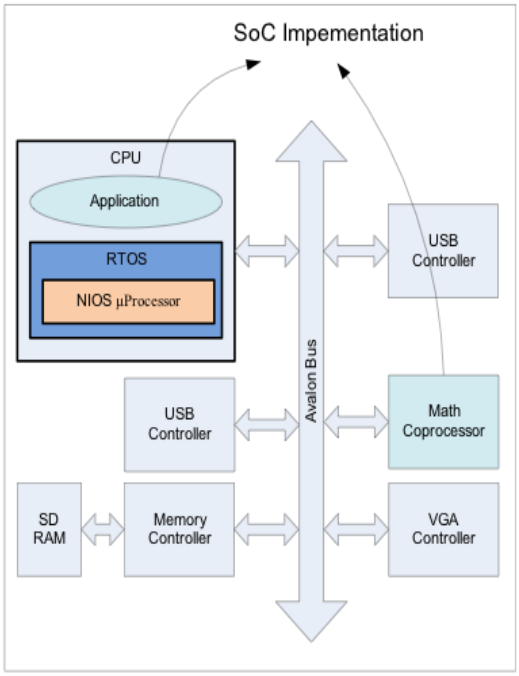

Figure 6. System Architecture

Calculator application as a prototype is implemented on an FPGA development board. Altera DE-2[11] and NIOS (soft) processor is used in this experiment to execute the software. So the platform consists of a VGA module, NIOS processor, uC/OS[5], and some peripherals.

The framework code as an output of the system consists of C files and VHDL files. The C files is implementation of the software parts and VHDL is implementation of the hardware part. There are 40 software files. These files consists of 19 software object files, 19 task state machines files, 1 initiation file, 1 file define the events and other environment. Each of these files has a header files.

The hardware part of the system code consists of 5 VHDL files. One is the top level file and connections to the NIOS processor. The rest files describe the hardware modules (the FSM part only).

The generated framework is a simple one and not completed yet. Some functions and blocks of hardware are added to complete an application prototype such as the graphic/picture, multiplication, hardware addition etc. These codes is then added to the platform code which consists of NIOS processor and library of uC/OS-II (RTOS). All of the codes are compiled and then the result is put on the Altera DE-2 board.

V. Conclusion

The use of software with hardware/software co-design concept in SoC design addresses at least two design problems, time efficiency and flexibility. In hardware/software partitioning, user can flexibly decide some hardware part and software part as well as proportion of both in the system. Software for automation addresses the design efficiency. For the same system, it will need less time for designing than manual code generation. This paper shows that software for designing SoC framework based on hardware/software co-design approach can address the efficiency and flexibility of SoC design.

Acknowledgment

Maman Abdurohman would like to express his gratitude to Higher Education Directorate of Education and Cultural Ministry of Indonesia (DIKTI) for funding support in this research through “Hibah Bersaing” scheme. For local support, thanks to Unified Communication Laboratory of Telkom University and School of Electrical and Informatics Engineering, Bandung Institute of Technology for their research resources so that this research could be completed. This research is fully funded by Higher Education Directorate of Education and Cultural Ministry of Indonesia (DIKTI).

Refferences

[1] Abdurohman, M., Sasongko, A., Rawung, R. (2012) : "Automated SoC Framework Generator for Embedded System Design Using Event Driven Method". International Conference on Electrical, Computer, Electronics and Communication Engineering (ICECECE) Januari 2012. Zurich, Switzerland. Pp. 672-677.

[2] Arvind, Nikhil, R. Rosenband dan D. Dave, N. (2004) : “High Level Synthesis : An Essential Ingredient for Designing Complex ASICs”. International Conference on Computer Aided Design (ICCAD 2004). California. USA. 5-20.

[3] Cong, J. Fan, Y. Han, G. Jiang, W. dan Zhang, Z. (2007) : “Platform-based behavior-level and system-level synthesis”. University of California. Los Angeles. USA. 1-2.

[4] Coyle, F. dan Thornton, M. (2006) : “From UML to HDL : a Model Driven Architectureal Approach to Hardware-Software Co-Design”. Computer Science and Engineering Dept. Southern Methodist Univeristy. USA. 1-6.

[5] Jean J. Labrosse (1999). MicroC/OS-II : The Real-Time Kernel. R&D Books Lawrence.

[6] Kim, Y., T., Kim, T., Kim, Y., Shin, C., Chung, E., Y. Choi, K., M., Kong, J., T., dan Eo, S., K. (2005) : “Fast and Accurate Transaction Level Modeling of an Extended AMBA2.0 Bus Architecture”. Proceedings of the Design, Automation and Test in Europe Conference and Exhibition, IEEE. 1-2.

[7] Kogel, T., Wieferink, A. dan Kroll, H. (2002) : “Systemc Based Architecture Exploration Of A 3d Graphic Processor”. Integrated Signal Processing Systems Synopsys Inc. 1-8.

[8] Mathaikutty, D., A. (2007) : Metamodeling Driven IP Reuse for System-on-chip Integration and Microprocessor Design, Dissertation at Virginia Polytechnic Institute and State University. 10-20.

[9] Mooney III, John, V. (1998) : “Hardware/Software co-design of run-time systems”. Dissertation at Stanford University. 10-15.

[10] Pelkonen, A. Masselos, K. dan Cupak, M. (2003) : “System Level Modeling of Dynamically Reconfigurable Hardware with SystemC”. VTT Technical Research Center of Finland. 1-20.

[11] Pong P. Chu (2011). Embedded SOPC Design With Nios II Processor and VHDL Examples. Wiley.

[12] Vahid, F. dan Givargis, T. (2002) : Embedded system A Unified Hardware/Software Introduction, JohnWiley & Sons, Inc., New York. 1-2

SEEK DIGITAL LIBRARY

About Author (s):

Dr. Maman Abdurohman is a researcher in Telkom University, Indonesia. He is now head of Unified Communication Laboratory. His major research area ares Embedded System Design, M2M communication, and Design Automation.

SEEK DIGITAL LIBRARY