

# A Survey of Hardware and Software Co-design Issues for System on Chip Design

J. Kokila, N. Ramasubramanian and S. Indrajeet

**Abstract** The modern embedded system needs to be designed to meet the tremendous changes due to high speed and advancement in technologies. Encapsulating user needs into a small area and enhancing the specification of today's requirement is really a challenging task for System on chip designer. This review presents the practical view of Hardware/Software co-design and its main design issues tackled in recent years. The role of SOC cannot be limited to a single task, since it is an ocean of streams like Computer Architecture, VLSI and Embedded System, which has been joined together to deal with a current multi-tasking environment. The key motive of this survey is to cover the design issues faced by SOC designer and applying all these in Hardware Software co-design. A comparative study is made to show how the specific design is used efficiently and goals are reached in each proposal with its metrics and drawbacks.

**Keywords** SOC design · Driving factors · Methodologies · Hardware software co-design

## 1 Introduction

In Today's world 98 % of systems are using embedded systems. Hardware has to be cautiously designed and software has to be efficiently programmed in parallel in an embedded system so that overall system performance increases, reduction in power consumption and high Integrity has to be maintained. An embedded system

---

J. Kokila (✉) · N. Ramasubramanian · S. Indrajeet  
Department of Computer Science & Engineering, National Institute of Technology,  
Trichy, India  
e-mail: jk.cse09@gmail.com

N. Ramasubramanian  
e-mail: nrs@nitt.edu

S. Indrajeet  
e-mail: indrajeet95@gmail.com

is an electronic system that has a programming module which is embedded in computer hardware. It is used everywhere in modern life, starting from Consumer Electronics, Education, Telecommunication, Home Appliance, Transportation, Industry, Medical and Military. Software method with higher quality code, fixing errors and optimization is more important for the performance raise. In hardware building, more and more complex designs in RTL are becoming increasingly difficult. System-on-a-chip (SOC) is an Integrated Circuit, which houses all the critical elements of the electronic system on a single microchip. It is widely used across the embedded industry due to their small-form-factor, computational excellence and low power consumption. The hardware/Software co-design has some intention and purpose based on prefix 'co'. They are co-ordination, concurrency, correctness and complexity. The survey of historical details and future direction about this strategy has been discussed in detail from the scratch for advanced electronic systems [1]. The current and future direction is also very much related to the driving factors which has been examined in this work.

The remaining of this paper is organized as follows. Starting with the recent issues of the SOC design in Sect. 2 with latest comments, followed by the discussion of various proposals in Sect. 3 with different variants of hardware/software co-design, that have simultaneously emerged from 2013 to 2015. In Sect. 4 the designing strategies of SOC with its overview has been explained and highlighted the importance of core method. Finally, Sect. 5 remarks the main conclusion of this survey.

## 2 Design Issues

The design of the system will be affected by the choice of hardware and the type of software they use. The requirements of choice can be divided into two basic groups which may be used and achieve system goals. The users wish certain apparent properties in a system that is a system should be fast, convenient to use, reliable, easy to learn and to use, and safe. The system should be comfy to design, implement, and maintain and it should be precise for the below mentioned goals.

### 2.1 Performance

The demand for performance is increasing from data processing in documentation to high-speed computing and clustering, which in recent years plays main role in research and complex designing. The Performance is considered as the most important factor for design issues which may be due to double 'r' that is reconfiguration [2, 3] and reliability [3]. Multiprocessor SOC could provide immense computational capacity, achieving high performance in such systems is highly challenging.

## **2.2 Complexity**

Progress in Embedded System guides us to grow a finite number of feature at low power integration and complexity into a single chip. This dispute requires a set of ways to integrate heterogeneous systems into the same chip to reduce area, power and delay. The five facets of complexity that range from functional complexity, architectural and verification challenges to design team communication and deep submicron (DSM) implementation.

## **2.3 Power Management**

The SOC power issues have forced a reorganization of methodologies all over the design flow to explain power-related effects There are two types of power in SOC design one is static which is for memory and dynamic is for logic. Overall power in a chip may be dynamic power consumption, short-circuit current and leakage current will change dramatically. Increase in speed and decrease in size will lead to head dissipation and reliability issues. This may cause electro migration and IR drop for wide range of design applications.

## **2.4 Reliability**

The parts of an SOC, Intellectual property and blocks related to memory, can go on to function while other parts do not. Some may work irregularly, or at lower speeds, others may increase in temperature to the point where they either shut down or slow down. This leads to some interesting issues across a variety of the electronics industry, ranging from legal liability, to what differentiates any functioning system especially critical systems from those that are non-functional or marginally functional. Just as systems are getting more complex, we need more and more metrics for analyzing it.

## **2.5 Security**

If all the technology works as designed, there are yawning security holes in designs at every process node and in almost every IoT design even if it's only what a well-design piece of hardware or software is connected to. It is obvious that a device which is compromised is no longer reliable. But a device that can be flexible is not reliable, either. All of the processor companies have been active in securing their cores. ARM has its Trust Zone technology for compartmentalizing memory and processes. Software security alone is not needed for security but hardware plays a major role because hardware attacks are very much concerned With SOC.

### 2.6 Time-to-Market

The SOC designer can improve time-to-market and evade costly re-spins by using FPGA-based physical prototypes to speed up system validation. FPGA-based prototypes enable embedded software developers to use a real hardware earlier in the design cycle, allowing pre-silicon validation and hardware/software integration prior to chip fabrication. The use FPGA and simulation make the designing process easy and efficient.

## 3 Co-design: A Glance at Existing Approaches

The diagram (Fig. 1) targets the enhancement of SOC design as the effect of the design issue (cause) which in turn, become the effect of selected papers in this survey. A comparative study has been made in the Table 1 and each design issues are matched with some recent proposals.

### 3.1 Performance Increase Using Reconfiguration

The megablock is a basic repetitive block, the detection of which aids in finding the loops which can be mapped into a Reconfigurable Processing Unit (RPU), consisting of a specialized reconfigurable array of Functional Units. The synthesis of the RPU is done offline while the reconfiguration is done online. This work can be

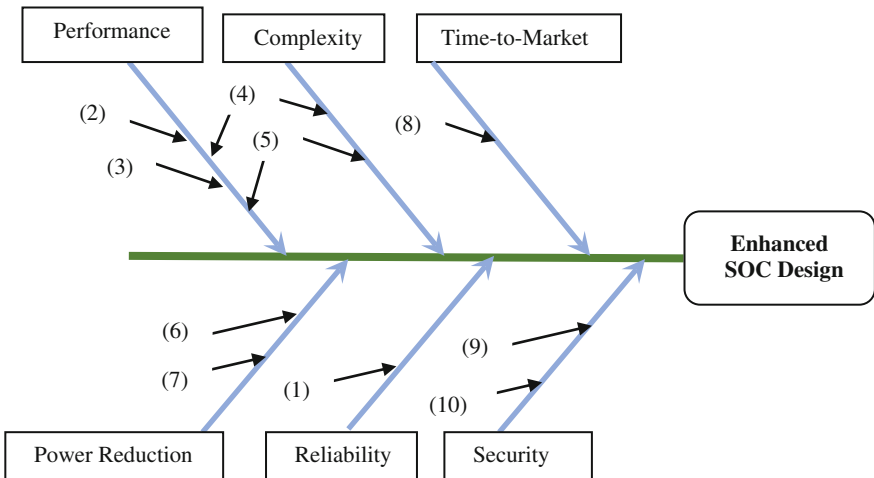


Fig. 1 Cause effective diagram for driving factors

**Table 1** Comparative study of driving forces

S. No.	Design issues	Methods adapted	Hardware	Software	Metrics	Limitation
1	Performance	ZyCAP [4]	ZyCap-controller	Device configuration interface	Resource utilization performances	Need more resources
2	Performance	Reconfigurable HW/SW systems [3]	RPU	Megablocks	Execution time communication over-head	Blocks the processor during reconfiguration
3	Performance and complexity	Hardware software coherence protocols [5]	Local memory	Compiler	Speedup energy reduction	Uses of CPU cache
4	Performance and complexity	Accelerator coherency port using Xilinx Zynq [6]	Hardware accelerator and DMA	Application to transfer data between accelerator and memory	Energy efficiency, speed	Critical application will overloads cache space
5	Power reduction	Optimize battery life in SOC device [7]	Power management	Features and flexibility	Power variance	Lack of SOC IP block for power consumption
6	Power reduction	Device runtime power management [8].	Monitor to check bits	Central PM agent	Execution time and resources	Increase hardware complexity and not addressed the QOS
7	Reliability and complexity	Cardio [6]	Fault detection	System reconfiguration	Scalability performance	Overhead in broadcasting message at least one system should be active
8	Security	Authentications on FPGAs using PUF [7]	PUF	Authentication	Threshold error rate	Hardware resource occupied
9	Security	Locking phenomena using PUF [9]	TERO-PUF	Chip identification	Response time inter device variation	Architecture depend on clock input
10	Time-to-market	A metaprogrammed C++ framework [4]	Hw code Netlist (NOC)	SW code binary files	Memory FPA resource and latency	High through-put and cross-domain communication

carried out for cache memory, reduction in resource, floating point operation and for developing on line megablock for individual adaptable embedded systems.

In a system with multiple independent accelerators, a full reconfiguration forces reconfiguration of all of them even if not required. PR overcomes these limitations, and hence a key enabler for this paradigm of reconfigurable computing. Partial Reconfiguration in hybrid FPGA platforms like the Xilinx Zynq have definitely shown advantages and zycap, is a controller that significantly improves reconfiguration throughput in Zynq system over standard methods, while allowing overlapped executions, resulting in improved overall system performance. Zycap also plays a significant role in automating Partial Reconfiguration development in hybrid FPGAs.

### ***3.2 Complexity Measure Using Memory and Accelerator***

This paper focuses on coherence protocol for hybrid memory which may affect the power consumption and scalability. The two main concerns are the important quantity of power consumed in the cache hierarchy and the absence of scalability of current cache coherence protocols [5]. The optimized compiler are used mainly for generating machine language and it should be adaptable for any processor. The hybrid memory has incoherence between Local and cache hierarchy which is solved very efficiently by hardware-software protocols. The compiler generates an assembly code, even in memory aliasing problem. A comparative study has been made for cache-based architecture and hybrid memory. This work will increase average speedup of 38 % and energy reduction of 27 %.

Hardware accelerators can achieve significant performance for many applications. The motivation for enhanced execution speed and power efficiency for their specific computational workloads have been discussed in reference [10]. To perform a particular processing task using an accelerator there is a need to access memory. Xilinx Zynq provides three types of memory interface methods: GP (General Purpose), HP (High performance), and ACP (Accelerator Coherency Port). In the first approach hardware accelerator is connected to ARM CPU located on Zynq through GP interface for memory access through DMA. This brings down the performance. In the next approach accelerator is connected to ARM CPU through HP port, which increases the performance because HP ports have direct connection with DDR memory controllers on Zynq PS. In the third approach accelerator is connected through ACP port which further increases the performance because through ACP DMA can access caches of ARM cores which in turn gives low latency.

### ***3.3 Power Management and Optimizing Battery Life***

SOC contains many non CPU devices which are evaluated by drivers, so a device runtime Power management (PM) becoming very essential. PM is not used in

drivers because to save chip area same power and clock source is used by different devices, if any one device PM performance is poor then the whole domain should be in enable which increases power loss. The solution for this is relieving drivers from PM responsibility by replacing the PM code in all drivers with a single kernel module called as Central PM Agent. Central PM Agent have two parts called monitor and controller. The monitor will check whether devices are in busy or idle mode using software and hardware approaches. Based on monitor information controller will keep the device in ENABLED or DISABLED mode. The software approach will monitoring whether a device register is accessed by CPU for threshold time. We can also simplify the driver source code by using a Power Advisor which suggests where to add PM calls in driver.

The surfacing of battery-constrained SOC device such as Mobile phones, laptops, modern electronic devices and tablets, requires hardware-software co-design to optimize battery life. The Offloading behavior from processor to dedicated engine on the SOC can help decrease power consumption. The Data Acquisition unit along with instrumental platform has been used to assemble the power consumption of various SOC IP blocks. In this work hardware component is designed in Hardware and adding features and flexibility was included through software module. This work can be extended by the software to design an energy aware application by collecting software based IP.

### ***3.4 Reliability in CMP***

The permanent failures in Chip Multiprocessor is a big issue in terms of reliability which have been rising above by cardio system. A Distribute Resource manager was developed to collect data about CMP Components which will test the hardware components at regular intervals and information's are collected and analyzed for future failure. If any core failure is detected then the core will be reconfigured and at least one core should be non-faulty to do the task at runtime. An experiment was conducted to measure performance and response time of Cardio used C++ simulator, gem5 and cacti 5.3. The benefit of this system is more reliable for distributed resources, fault—aware OS and scalable for CMP. This proposed work was compared with hardware only technique such as Immune in NOC. The Cardio can be embedded in microcontroller and heterogeneous SOC design for more efficient fault-tolerance systems.

### ***3.5 Security Using PUF***

The Hardware and software components communication using remote method Invocation in an object oriented programming concept. A Meta-programming framework for reducing the gap between hardware/software boundaries using

Remote method Invocation of object oriented approach [4]. The Proxy/Agent Mechanism have been implemented in software and hardware components to generate binary file and netlist. The communication framework has to focus on two important issues that is components mapping and framework portability. Several template-based implementations of serialize and reverse may be defined in order to provide the optimal marshalling for each data type. Synthesis constraints were set in both CatapultC and ISE with the goal of minimizing circuit area considering a target operating frequency of 100 MHz.

## 4 Comparisons of Methodologies Adapted for SOC Design

The best process has to be selected and using them appropriately in correct position for specific design is again a big overhead. The methods can be used in single or in combination to form a higher level of design strategy to support a modern SOC. The following design strategies have been widely used for designing SOC from early days to today. The First and selected method is Hardware/Software co-design which refers to parallel design of hardware components using Hardware description language and software module using any programming language of heavily complex electronic real time systems. The second method is programmable core debits the changes in functionality of post fabrication in SOC. The increase in complexity and time-to-market pressures makes the post-fabrication changes more essential for modern SOC design. One of the primary ways in which these cores will be used is to allow designers to leave certain aspects of the design unspecified until after fabrication [11].

The chip designer may not have the detailed IP knowledge for combining intellectual property (IP) blocks from various sources. Although much effort has gone into determining best practices for integrating IP blocks into an SOC design is the third method, the effort involved in integrating stand-alone block-level timing constraints in the chip-level timing environment is often overlooked. The quality of the configuration and integration of complex IP blocks can have an important impact on a SOC's development schedule and performance. The solutions comprise a component related language, component selection needs algorithm, type matching and inferencing algorithms, temporal property based behavioral typing, and finally a software system on top of an existing model which is created on the different platform.

This survey comprises of various journal papers like IEEE, ACM and elsiver which uses Hardware Software co-design method to explore the goals of system design and is comparatively analyzed based on the flavor of computer science and engineering. Various approaches have been compared and its limitations are listed in the Table 1. They are mainly based on resource management in CPU, cache and clock for performance and security, cache utilization and adoption of critical application for complexity, QOS and Lack of IP for power Management, depending on at least one active processor for Reliability.



## 5 Conclusion

System-on-chip has been an imprecise term that magically holds out a lot of enthusiasm, and has been gaining stamina in the computer industry. Though the possibility is enormous, complexities are numerous, and answering these to offer successful designs is a true engineering challenge. All these issues create a path for future research works. Finally, it is also very auspicious to see from researcher's point of view that many positive, technical and fundamental questions will arise and that these excite and keep our lives busy for the next decades of years.

## References

1. Teich, J.: Hardware/software codesign: the past, the present, and predicting the future. In: Proceedings of the IEEE 100, Special centennial issue, pp. 1411–1430 (2012)
2. Vipin, K., Fahmy, S.: ZyCAP: efficient partial reconfiguration management on the Xilinx Zynq. *Embed. Syst. Lett. IEEE* **6**(3), 41–44 (2014)
3. Bispo, J., et al.: Transparent trace-based binary acceleration for reconfigurable HW/SW systems. *IEEE Trans. Ind. Inform.* **9**(3), 1625–1634 (2013)
4. Mück, T.R., Fröhlich, A.A.: A metaprogrammed C++ framework for hardware/software component integration and communication. *J. Syst. Archit.* **60**(10), 816–827 (2014)
5. Alvarez, L., et al.: Hardware-software coherence protocol for the coexistence of caches and local memories. *IEEE Trans. Comput.* **64**(1), 152–165 (2015)
6. Pellegrini, A., Bertacco, V.: Cardio: CMP adaptation for reliability through dynamic introspective operation. *IEEE Trans. Comput. Aided Des. Integr. Circuits Syst.* **33**(2), 265–278 (2014)
7. Aysu, A., Schaumont, P.: Hardware/software co-design of physical unclonable function based authentications on FPGAs. *Microprocess. Microsyst.* (2015)
8. Metri, G., et al.: Hardware/software codesign to optimize SoC device battery life. *Computer* **10**, 89–92 (2013)
9. Bossuet, L., et al.: A PUF based on a transient effect ring oscillator and insensitive to locking phenomenon. *IEEE Trans. Emerg. Top. Comput* **2**(1), 30–36 (2014)
10. Xu, C.: Automated OS-level Device Runtime Power Management. Rice University, Diss (2014)
11. Chen, Y.-K., Kung, S.-Y.: Trend and challenge on system-on-a-chip designs. *J. Sig. Process. Syst.* **53**(1–2), 217–229 (2008)
12. Sadri, M., et al.: Energy and performance exploration of accelerator coherency port using Xilinx ZYNQ. In: Proceedings of the 10th FPGAworld Conference. ACM (2013)
13. Rostami, M., et al.: Hardware security: threat models and metrics. In: Proceedings of the International Conference on Computer-Aided Design. IEEE Press (2013)