Case Study: First-Time Success ASIC Design Methodology Applied to a Multi-Processor System-on-Chip

Arya Wicaksana, Dareen Kusuma Halim,
Dicky Hartono, Felix Lokananta, Sze-Wei Lee,
Mow-Song Ng and Chong-Ming Tang

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/intechopen.79855

Abstract

Achieving first-time success is crucial in the ASIC design league considering the soaring cost, tight time-to-market window, and competitive business environment. One key factor in ensuring first-time success is a well-defined ASIC design methodology. Here we propose a novel ASIC design methodology that has been proven for the RUMPS401 (Rahman University Multi-Processor System 401) Multiprocessor System-on-Chip (MPSoC) project. The MPSoC project is initiated by Universiti Tunku Abdul Rahman (UTAR) VLSI design center. The proposed methodology includes the use of Universal Verification Methodology (UVM). The use of electronic design automation (EDA) software during each step of the design methodology is also presented. The first-time success RUMPS401 demonstrates the use of the proposed ASIC design methodology and the good of using one. Especially this project is carried on in educational environment that is even more limited in budget, resources and know-how, compared to the business and industrial counterparts. Here a novel ASIC design methodology that is tailored to first-time success MPSoC is presented.

Keywords: ASIC design, first-time success, design methodology, MPSoC, UVM

1. Introduction

The ever-changing and ever-challenging ASIC design environment consistently brings up new challenges into the ASIC design league. There are long existing challenges [1] such as soaring cost (design, verification, fabrication), high design complexity, tight time-to-market window and competitive business environment. The complexity of design and verification has grown tremendously in search for more computational power and capacity, which is also in accordance to the Moore's law. Nowadays the implementation of specific application such as face recognition in a System-on-Chip (SoC) has become a practice [2]. Another technology advancement is the development and use of Network-on-Chip (NoC) for communication inside the chip [3]. All of the advancement in technology today without doubt has enriched the ASIC products while at the same time introduced more and more challenges into the ASIC design and verification.

The long and many steps in the design and verification process of an ASIC may often lead to first silicon failure [4]. This is due to factors such as human errors, immature electronic design automation (EDA) software, lack of experience and discipline [5]. The first silicon failure is most often not acceptable in many business environments where a design re-spin is costly and may harm the continuity of the business. Hence, ensuring first-time success is fundamental to survive in the ASIC design league. The key success that is proposed here is the first-time success ASIC design methodology that describes the design and verification process in a well-defined manner starting from the user requirements (system specifications) until the fabrication (tape-out) of the ASIC. The proposed methodology also borrows the Universal Verification Methodology (UVM) [6] for the verification part of the design. The main objective of the proposed methodology is to serve as the guideline for achieving first-time success. The methodology has been used and proven for a Multiprocessor System-on-Chip (MPSoC) project initiated by the VLSI Design Center in Universiti Tunku Abdul Rahman (UTAR) as a demonstration of concept and viability.

The MPSoC project produces a first-time working silicon (first-time success) that is named RUMPS401. The RUMPS401 MPSoC is developed by using the proposed design methodology and with the use of relevant industrial standard EDA software tools (Synopsys, Mentor Graphics and Cadence). The first-time success RUMPS401 has proven that the proposed design methodology or the "UTAR first-time success ASIC design methodology" ensures first silicon success. Notably, the development of the RUMPS401 took place in educational environment where budget, resources and know-how are limited compared to the business and industrial counterparts. The novel UTAR first-time success ASIC design methodology is presented in the next section.

2. UTAR first-time success ASIC design methodology

The methodology is comprised of two major parts: front end and back end. Every development of a commercial electronic product is derived from the user requirements, which is translated into system specification. This system specification is the entry point in the proposed ASIC design methodology and the final design step of the methodology is tape-out for fabrication. At the front end part of the methodology, the end is the equivalence check process and the resulting netlist from the process is then passed for the back end step. The back end step starts with floorplanning using the netlist that has passed the equivalence check process earlier. The process is carried on through the final step that is the tape-out. Here is the proposed ASIC design methodology as shown in **Figure 1**.

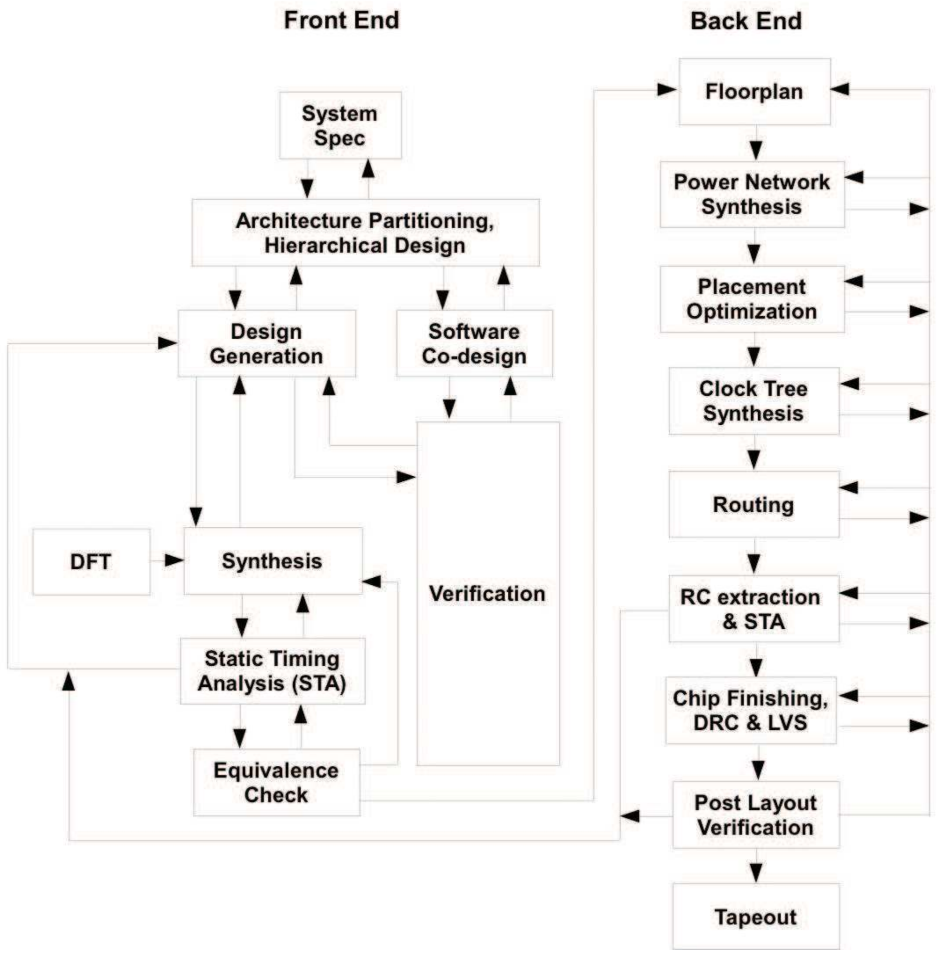

Figure 1. The proposed ASIC design methodology.

3. RUMPS401 MPSoC project

The objective of the RUMPS401 project is to be able to successfully implement the MPSoC to a ready-for-tape-out state. To fulfill this target, the "First-Time-Success" ASIC design methodology is strictly followed through the design flow. ASIC design methodology adoption affects the decisions in every design steps. This and the following sections elaborates the implementation of the methodology through the design process of the RUMPS401. In addition, the usages of the commercial ASIC design tools are discussed as well.

3.1. Design specification

The first design process of the RUMPS401 MPSoC is defining the system level functional specifications. RUMPS401 chip is targeted for microcontroller-based applications that perform relatively high data processing. To support this requirement, the chip is designed to contain multiple processor cores. RUMPS401 consists of four ARM Cortex-M0 processors. M0 processor implements Advanced High-Performance Bus (AHB) protocol to connect with the peripheral modules. Network-on-Chip (NoC) interconnect is selected as the inter-processors communication media. NoC architecture is preferred over the bus-based interconnect. It is more suitable for an MPSoC system due to it scalability property and capability to deliver better performance than it bus-based counterparts [7–10].

Several targeted applications—such as: coarse-grain Advanced Encryption Standard (AES) encryption and Software-Defined Radio (SDR)—are used to derive the functional specifications. For instance, in the coarse-grain AES encryption application, the targeted functional features are:

- The SoC retrieves a continuous stream of plaintext as the input.
- The plaintext are divided into data blocks and distributed to multiple processor cores through the inter-processors communication architecture.
- Multiple processor cores perform parallel coarse-grain encryption on the plaintext.
- The system accumulates the ciphertext as the result of the encryption process and arranges it in the correct order.
- The SoC transmits out the ciphertext as the output.

The design is partitioned into hardware and software in accordance to the functional specification. The hardware design of the RUMPS401 is partitioned into sub-systems level of design and furthermore into the modules level of design. Hierarchical design approach simplifies the design and verification process in the early stage of the design flow. Modularity design technique also enables the reusability of the modules in different part of the design. Some modules are designed specifically to enable and accelerate the targeted applications. For example, the AES accelerator and multiply-accumulate (MAC) module are targeted for the encryption and SDR applications respectively. The other modules such as the Flash and SRAM memories controllers support the basic functionality of the MPSoC. The defined specifications of all specific modules are summarized and written into specification document.

In the evaluation process of the design specification of the modules, several important aspects are factored into the analysis. It includes: performance, area and power consumption. Some key analyses are also discussed in the few next sub-sections.

3.2. Design partitioning

The RUMPS401 chip functional features are partitioned into smaller and more specific tasks. These tasks are then assigned to the four processor cores. For that reason, each of the processors has different sets of hardware modules. With this architecture implementation, RUMPS401 can be categorized as a heterogeneous MPSoC. One of the processor cores, called IO-control core, is designated to handle Inter-chip communication tasks. The core is equipped with communication modules such as parallel-port controller and has the largest number of IO pins. It is also utilized as the main coordinator core. The other cores are used for more specific functions. Two of the cores are named normal cores and equipped with AES accelerators. Likewise the DSP core contains MAC module.

The hardware implementation of parallel port controller, AES accelerator and MAC modules help the processor cores to perform repetitive tasks. In the application, these tasks especially take the most processing time. The processor cores are used to handle the less demanding tasks. Application software is designed to run the processor cores to execute these tasks. Scheduling and synchronization of tasks executions are also implemented in the software. These two aspects are increasingly important in the system that contains multiple processor cores.

The processor cores require AHB (Advanced High-performance Bus) to NoC (ahb2noc) module as the on-chip communication bridge. This ahb2noc bridge is specifically created to interface the M0 processors in the RUMPS401 that use AHB-lite protocol and the NoC interconnect protocol. The data buffering processes in the ahb2noc are implemented in hardware to help speeding up the processor cores. Similar data buffering mechanism is applied to the parallel port module. These modules are used to transmit and receive multiple numbers of data. The modules are designed to register the received data in their buffer and generate interrupt signal to inform the processor cores about the availability of data. With this scheme, the processor cores are only needed to serve the Interrupt Service Routines (ISRs) once for some period of time. The mechanism helps the processor cores to save numerous processing times in compared with the mechanism without physical buffers.

RUMPS401 MPSoC does not contain any analog design block. If an analog design module is required such as Analog-to-Digital Converter (ADC), the development of the analog block is started together with the development of the digital hardware module and the application software. Behavioral model of the analog block is required to be prepared for the simulation purpose.

3.3. Network-on-Chip

NoC-based System-on-Chip is divided into multiple Processing Blocks (PBs). Each PB could contain one or a few components, and all of PBs are connected to each other via routers. A router connects to one or more neighbor routers depends on the topology of the network.

NoC used packet-switched method to send data from each PB to one another. All of the routers have an exclusive address and routing algorithm. Data and header are wrapped as a flit. Flit header consist of destination address and type of the data. Each PB sends flits to destination router through the network. The router’s routing algorithm will determine the paths for the flit to reach its destination.

NoC has become the solution of choice for System-on-Chip traditional interconnects problem [11]. Each router is directly connected to neighbor router, as a result interconnect wire can be kept short. NoC interconnects are also able to handle multiple communication flows in the form of multiple flits and thus providing high communication bandwidth.

The design of NoC are divided in various aspects such as the topology, routing algorithm, flow control, and router microarchitecture. Topology determines physical layout, connections between nodes, channels in the network, and has an important role in routing strategy and application mapping [12]. Topology design starts with basic typical network topology for example, tree, ring, star, or mesh topology. Optimization and additional feature are implemented to

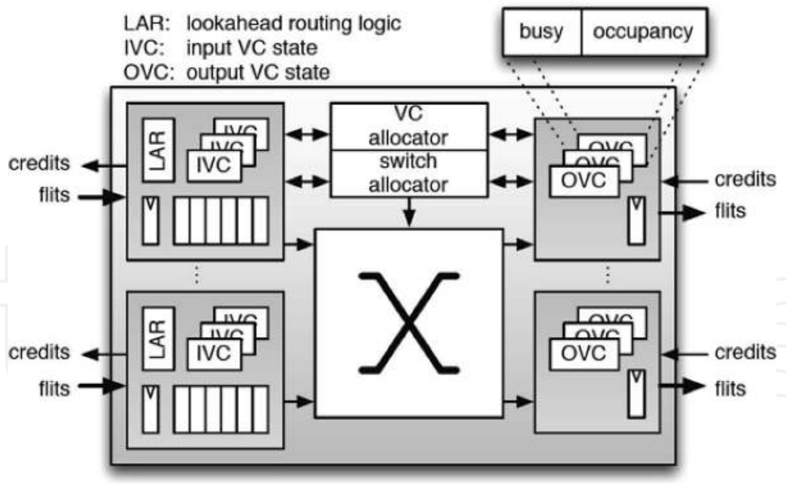

Figure 2. Router microarchitecture overview [15].

improve these basic topologies, as result new and more complex interconnects are formed. For example, the Quarc NoC architecture is an improved ring NoC topology [13]. Floor-planning and layout of the NoC architecture need to be considered as well [14].

Routing algorithm determines the path selected by a flit to reach its destination. A good routing algorithm is able to balance the traffic inside the network so that the performance and throughput of the network will be optimized. Flow control regulates the resource allocation process to a flit as they travel through the network. Congestion control, energy and power management and fault tolerance are communication design problem that need to be resolve as well [12].

Application modeling and optimization is another challenge in a NoC-based MPSoC. Every application has different communication traffic pattern. Different application mappings will significantly affect the network traffic. A good understanding of the traffic patterns and system requirements helps in determining the optimal network topology, and this has a huge impact on design costs, power, and performance [12].

Router performance, cost and efficiency depend on its microarchitecture [15]. Router micro architecture made up of input buffer, router state, routing logic, allocator and a crossbar. Input buffered routers have become the primary choice in current NoC research. Flits that cannot be forwarded are temporarily held in input buffers at the router until they can proceed to next hop. **Figure 2** shows a typical NoC router microarchitecture.

4. Front-end design

The RUMPS401 design is carried on in Register-Transfer Level (RTL). The Verilog Hardware Description Language (HDL) is used to implement the module blocks. The ARM Cortex-M0 processor cores RTL design is obtained through the ARM's University Program. The program allows the usages of the M0 processor in university research projects. The SRAM modules that are attached to each core are generated by using ARM's Artisan SRAM generator, while the Flash memories for program storages are proprietary to Silicon Storage Technology (SST).

The Widely used commercial ASIC design tools by Synopsys are utilized through the entire design process. Universal Verification Methodology (UVM)-based Verification Platform is constructed to verify the modules and the system level functionality. The RUMPS401 design is then synthesized into gate-level netlist. The design is targeted to use Silterra's 180 nm CMOS Technology. Equivalent checks and gate-level simulations are performed to the gate-level netlist against the original RTL design.

4.1. UVM-based verification platform

The verification platform is a structure consisting of testbenches. The testbenches are designed in a hierarchical structure. The most basic testbenches are created to verify individual RTL modules. As multiple RTL modules are combined together to form a more complex subsystem, the components of the module level testbenches can be reused to build the subsystem level testbench [11]. The RUMPS401 MPSoC system level testbench are the top most level of this hierarchical verification platform. The testbenches are built based on the UVM that used System Verilog Hardware Verification Language (HVL). System Verilog adopts the Object Oriented Programming (OOP) concept to provide more robust and reusable features to write testbenches as compared to Verilog based testbenches. The testbenches based on UVM concept are composed of reusable verification components that are called Universal Verification Components (UVCs).

4.1.1. UVC

An UVC is an encapsulated, ready-to-use, configurable verification environment for a specific interface protocol, design sub-module or a full system [6]. Each UVC is designed for a specific design or protocol. It generates stimulus, performs checking and collects coverage information. The RTL modules or systems that are verified are referred as Device-Under-Test (DUT). Some examples of UCVs that are designed to verify the RUMPS401 verification platform are: AHB-Lite, NoC and parallel port UVCs.

An UVC is comprised of sub-components that are called Agents. Inside a UVC, each agent handles a specific functionality but is still related to each other. These Agents have another level of sub-components: Driver, Monitor and Sequencer. Driver is an active component that drives the signals to stimulate the DUT. A series of stimulus to be executed by the driver is generated by Sequencer. These test stimuli are organized as items called sequences. Monitor acts passively by sampling the DUT signals.

4.1.2. Testbench architecture

Depending on the complexity of the DUT, a testbench architecture can contain one or more UVCs. It is not uncommon that some DUTs require other RTL module(s) to help the testbench to run the simulation. **Figure 3** shows the testbench architecture for the Parallel-port module. The RTL module of AHB-Lite bus is integrated to the testbench architecture together with the

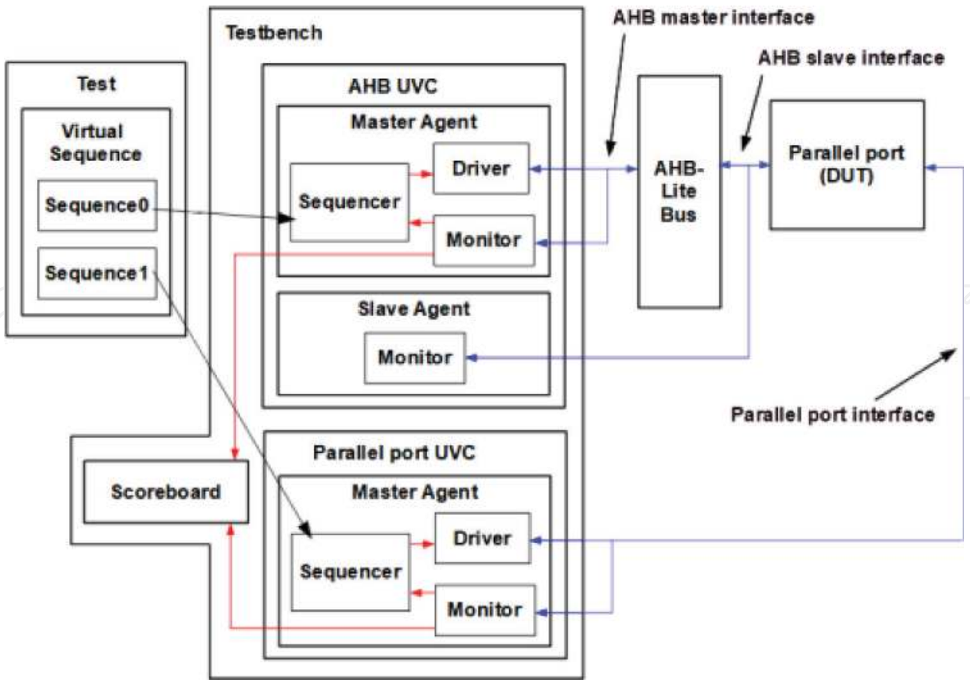

Figure 3. Parallel port module testbench architecture.

parallel port module. The AHB-Lite bus provides the necessary signals to access the AHB-Lite slave port of the parallel port module. An AHB-Lite Master Agent inside AHB-Lite UVC is connected to the AHB-Lite master port of the AHB-Lite bus module. The AHB-Lite Master Agent drives AHB-Lite transfers to access the registers inside the parallel port module. A passive AHB-Lite Slave Agent is attached to the connection between the AHB-Lite bus and the parallel port modules. This passive Agent only contains a Monitor and is only attached to collect the transfers. A parallel port UVC is connected to the parallel port interface of the parallel port module. The parallel port UVC trades data with the parallel port module using the parallel port data transfer protocol.

The test scenario to be run on the testbench architecture is defined in a component called Test. It selects the sequences to be run on all the sequencers inside the testbench to create a complete test scenario. When the test is running, scoreboard component is collecting all the necessary information. To obtain this information, scoreboard is connected to the monitors. The information are then used to perform comparison and check. Scoreboard reports failures if incorrect functionalities are performed by the DUT.

With the hierarchical approach, the module level testbenches are combined to form sub-system level testbenches. The sequences that generate some specific stimulus scenarios for the specific modules can also be reused. The same rules are applied when the sub-system RTLs are combined to form the MPSoCs [11].

4.1.3. Verification plan and coverages

Verification plan is constructed as the main guideline in the verification process. It contains multiple sets of tests for all possible test scenarios, especially all corner cases to stress the design. Some tests is designed as directed tests where the stimulus to drive the DUT are predefined. More advance tests generates constrained random stimulus. In this scenario, tests are randomized with the guide of constraints to make the stimulus to be more relevant.

There are two important goals in designing test scenarios. The first one is to test all possible functionalities of the module. The tests to cover these functionalities are derived from the RUMPS401 design specifications. 'Functional coverage' term is used to determine how much the functionalities of the design have been verified. Functional coverage reaches 100% when all of the functionalities are exercised. The second goal is to achieve a highest possible level of fault coverage. Ideally, good test scenarios are able to toggle all interconnect in the gate level netlist of the DUT and every failure must be observable at the output ports. However, due to the advancement of the geometry technology nodes that consistently increases the complexity of SoCs, fault simulation is no longer feasible. Code coverage concept is brought up as an alternative. Code coverage tracks how many line of code have been executed (line coverage), which paths through the code and expressions have been executed (path coverage), which single-bit variable have had the value 0 or 1 (toggle coverage) and which states and transition in a state machine have been visited (FSM coverage).

Verification coverage reviews are periodically performed through the design process. The verification coverage is applied as the parameter of the design progression. The periodic reviews are important for design evaluations and verification plan updates. Often new test scenarios are initiated as the result of analyzing the functional and code coverage reports. 100% verification coverage of the RUMPS401 chip is achieved towards the end cycle of the verification review as one of the signoff criteria.

4.2. Synthesis and design for testability

To ensure the synthesizability of the RUMPS401 RTL design, synthesizable coding guidelines are strictly followed. However, the synthesizability of an RTL module cannot be verified by simulation tools such as VCS. The Design Compiler (DC) tool from Synopsys is used to check design syntax and eliminate the non-synthesizable codes.

4.2.1. Design synthesis

The synthesis process is done to the Verilog RTL of RUMPS401 that are completed and fully verified. Design synthesis requires vendor specific synthesis libraries. The libraries contain information related to the available logic gates which are used as the target to translate and optimize the RTL design during the synthesis process. These libraries are prepared and loaded into DC tool. The RTL modules which are stored in multiple Verilog files are then read in and analyzed. DC reports errors when there are problems related to library or RTL synthesizability issues.

In order to guide the synthesis process, design constraints are applied to DC. The RUMPS401 design is targeted to operate with 16 MHz clock frequency. However for margin consideration, the design is constrained to 22.2 MHz. In addition 0.35 ns clock uncertainty constraint is applied to consider clock skew and jitter. For I/O-related timing constraints, time budgeting method is applied. For the chip input ports, DC requires information regarding the clock period and the output propagation delay of the signal from the device it interfaces with. For the outputs, other than the clock period, DC must also be provided with the input setup time specification of the device that is driven. With the time budgeting rule applied, 60% of the time allocation is dedicated to the external devices, and the remaining 40% is used to constraint internal IO controllers. The clock and reset signals are set as ideal nets therefore DC does not have to contend with clock skew issues at this stage. Clock and reset signals skew are particularly handled during the clock tree synthesis process in the physical design process.

In defining design constraints, cost consideration has to be balanced against performance. In the case where cost is the critical factor, the area constraint of the chip is set to some very small value or even zero. This constraint setup lets the DC to optimize the design as small as possible. In order to push the performance of the targeted technology, the clock period is set aggressively. Nevertheless sufficient timing margins must be included to consider physical variations, timing uncertainties and physical layout timing closure [11]. All these considerations are accounted to the limitation of the targeted 180 nm technology. Smaller process nodes allow tighter constraint such as higher system frequency and smaller clock margin to enable better performance. However smaller devices dimensions introduce worse leakage power that needs to be taken into consideration.

The final synthesized gate level netlist of the RUMPS401 design managed to meet the targeted constraints. There is a number of timing paths that violate setup requirement. However the slacks are in the range of 0.1–0.3 ns and are considered pretty small. These violations are left to be solved by the layout tools. The synthesized netlist requires 8.93 mm^2 total area. This value is the best case result that could be provided by DC due to the zero are constraint that is applied.

4.2.2. Design for testability

Physical defects can occur during the manufacturing process of chips. These physical defects could cause faults on any part of the chips. The design for testability (DFT) concept provides the capability to detect the occurrence of faults in all parts of the chips [11]. Most often for cost consideration, chips only provide limited number of I/O pins. DFT techniques are developed to facilitate the uses of limited I/O pins to apply test vectors and to verify the results. DFT is driven by the need to provide 100% controllability and observability.

The main DFT technique that is implemented in the RUMPS401 design is scan chain insertion. With this technique, all of the flip-flops in the design is replaced with scannable flip-flops that have an additional multiplexer at its input pins. In normal chip operation mode, flip-flops are interconnected through functional logic gates. During the scan mode, the output of a flip-flop is linked directly to the input of the next flip-flop. This scenario creates a long shift registers chain. For some cases where the chip has a large number of flip-flops, multiple scan chains are constructed to speed up the test process. RUMPS401 chip contains four scan chains. The start and the end points of these chains are brought to the I/O pins of the chip.

DC provides a feature to add scan chain into the design that is synthesized. With this option enabled, the scan chains are inserted to the gate level netlist of the RUMPS401 MPSoC. An additional file is also generated by DC that contains the detail information regarding the scan chains structure in RUMPS401. This file and the gate level netlist are used as the input to the Automated Test Pattern Generation (ATPG) tool. The tool analyzes the scan chain structure and generates all possible combination of test patterns. The Synopsys Tetramax ATPG tool is used for RUMPS401 test pattern generation.

Memory devices such as Flash and SRAM modules in RUMPS401 are designed to have very high density of memory cells. These memory devices are prone to suffer from physical defects. Studies have been done to classify faults that commonly appear in the memory devices. Some of the common memory related faults are: stuck-at faults, transition faults, coupling faults, bridging faults and state coupling faults. Test algorithms are developed to perform faults detection in memories modules.

A dedicated Memory Built-In-Self-Test (MBIST) circuit is designed to implement the memory test algorithms. The module is embedded into the SRAM controller modules. The MBIST module implements checkerboard algorithm, where it fills the entire array of memory cells with checkerboard pattern. The memory cells are alternately filled with one and zero values in vertical and horizontal direction. The MBIST module then reads back the values to detect any defect in the memory. The same test is repeated with the inversed pattern.

Similar to the MBIST, Flash wrapper interface is designed to provide direct access to the Flash memories in the RUMPS401. The Flash wrapper module implements multiplexer that is used to select one of the four Flash modules. The selected Flash module I/O pins are exposed by the Flash wrapper to the chip's I/O. Through the wrapper, test algorithms to erase, program and read the Flash memory cells are conducted. In addition, this Flash wrapper interface is also utilized to initialize the Flash modules after the manufacturing proses. The Flash initialization process must be done to setup the operating parameters of the Flash memories before it can be used to store program codes.

Other DFT methods are considered with the incorporate use of the M0 cores inside the RUMPS401 chip. One of the areas that are covered by this DFT method is the data buffers, such as in the NoC routers. Test software is written to be executed by the processor cores to pump data through the NoC buffers. The data is then read back and verified. The same method is also applied to perform the test to the peripheral modules, by looping back the output pins of the peripherals back to its inputs.

4.3. Equivalence check and gate-level simulation

With the rise of design complexity of SoCs, formality verification becomes critically important to ensure the correctness of every design step. The equivalence check is a common verification step in the design flow that utilizes EDA tools. This check is performed to verify the equivalence of designs that are implemented in different abstraction level. Therefore equivalence check is usually performed after the synthesis process is done and after the physical design implementation is finished.

In this RUMPS401 project, the equivalence check is done with the Formality tool from Synopsys. The gate-level netlist of the RUMPS401 from the synthesis process is used as the input. The original RTL design is also included as the reference design. The Formality tool implements mathematical techniques to perform logic circuits comparison of the RTL and gate-level designs. The tool then generates equivalence check reports base on the comparison results. A configuration script is required to guide the Formality tools to perform all the tasks.

Simulations are also conducted to the gate-level netlist of the RUMPS401 chip. These gate-level simulations are implemented as the double check to ensure the functionalities of the gate-level netlist are the same as the RTL design. The gate-level simulations utilize the same verification environment that is constructed to verify the RTL design. The RTL design is substituted with the gate-level netlist and then all of the test scenarios are applied and verified. The simulation results show all the generated logic gates of RUMPS401 design are function correctly.

4.4. Static Timing Analysis

DC tool provides embedded timing analysis feature to evaluate the logic path delays during synthesis process. However a more accurate Static Timing Analysis (STA) is needed to be performed to the gate-level netlist. Several factors such as variation of manufacturing process, supply voltage level and working temperature affects the timing characteristic of the cells and interconnects inside the chip. STA is conducted to verify that the design meets the timing requirements across the corner variation of Process-Voltage-Temperature (PVT) [11]. The RUMPS401 chip is targeted to operate with ±10% supply voltage variation (2.97–3.63 V) and in the environment with temperature ranged from −40 to 125°C.

STA is performed to the RUMPS401 design by using Prime Time (PT) tool from Synopsys. PT implements mathematical methods to calculate accurate timing delays. The gate-level netlist, vendor specific libraries and design constraints are read into PT. PT calculates the timing delays for all possible paths and uses it to perform the analysis. It generates the timing reports as the result. Critical timing paths are listed in these reports. One of the worst critical timing path is located at the output data bus of the Flash memory. The Flash memory module requires maximum of 33 ns time to generate the output data. With the design is constrained to use 22.2 MHz clock frequency, the output data bus of the Flash memory only lefts 12 ns time allocation. Refinements on the RTL design or synthesis process are done to reduce severe timing violations. In the Flash memory output data bus case, tight constraint is applied to the data path and the standard cells with the highest drive strength are used with the expense of chip area.

5. Back-end design

The back-end design process translates the logical RTL design into a layout, which defines how the gates and IP blocks are physically laid out in silicon wafer, as well as the connection between them. The end-result of this process is a layout file that foundry requires for the silicon fabrication. This section describes the back-end design steps done with Synopsys ICC tool and general view of the RUMPS401 physical layout design.

5.1. Initial design setup and floorplanning

Back-end design process starts by setting up the ICC with vendor-specific technology files and libraries. Technology files define the set of layers used in fabrication and design rules that must be followed. Libraries define the cells and IP blocks (logic gates) physical properties such as operating condition, gate delay, size, and pin locations. Additionally, the setup process also loads TLU+ files provided by vendor, which contain the parasitic RC model. Most importantly, the setup loads netlist and timing constraint files resulted from the design synthesis process.

The process continues by moving to the floorplanning step, in which the general layout or footprints of the chip is defined. This includes the total silicon area, core area, IO pads arrangement, power bus area, standard cells area, and IP blocks placement. The IP blocks placement is necessary as they usually have much larger size compared to the standard cells (gates) and are put on certain area to allow easier routing process. Normally, the IP blocks placement are done manually by the designer along with their blockages to prevent standard cells from being placed around the IP blocks, to reduce the congestion around the area.

Once the IP blocks are in place, standard cells are coarsely placed into the defined area. At the end floorplanning, the design will have all standard cells, IO pads, and IP blocks placed inside the defined silicon area. It is advisable to run multiple type of checks regarding congestion and core utilization. Core utilization is the ratio of area consumed by standard cells and IP blocks against total core area. Generally, it is kept at 40–50% to allow sufficient spaces for routing connections between the cells.

5.2. Power network synthesis (PNS)

Power is a crucial element for any chip as it is impossible to function without any electricity. The same goes for the cells inside the design. Hence, power network is laid out first to ensure that there is enough space for power lines, and that they utilize highest level of metal layer that is thicker than others for the best possible electrical characteristic. The goal of this step is to lay out power straps for the core area, from which the standard cells and IP blocks can tap into. In some design, due to operating voltage or current condition IP blocks may require separated power straps that tap directly into IO pads that supplies power.

Designer may choose to lay the straps manually or automatically with ICC's PNS feature. Either way, IR-drop analysis must be performed to ensure that the power straps is able to supply sufficient power to every cell in the design. It measures the voltage drop caused by strap resistance as a function of its length and width. The process of laying out straps and running IR-drop analysis is performed in cycles until the voltage drop level is within an acceptable range. In RUMPS401 design, the IR drop is kept below 50 mV across the chip. The chip uses 1.8 V CMOS technology for its internal cells and designed with ±10% tolerance for supply voltage, hence the maximum allowed IR drop is kept at lower value to accommodate the supply voltage variations.

The RUMPS401 power network are laid on the two highest layers of Silterra's six metal layers technology. Ground straps are laid on the sixth metal layer, while the 1.8 V straps lie on the fifth metal layer. The main power rings and are 100 and 30 μm wide, respectively. Standard cells power busses tap directly into the power straps through vias. There are 6–7 pairs of power straps laid vertically throughout the cell area to provide even power distribution and IR drop. Instead of tapping to the power ring or straps, the flash memories are powered individually via the power IO pads spread on every side of the chip. This is done in accordance to vendor's application note.

IR drop simulation were run separately for the 1.8 and 3.3 V supplies. Worst IR drop on the 1.8 V line is observed on the power straps going into the bottom-right flash, measure at 33.5 mV. An IR drop of 25.2 mV is measured on the 3.3 V power straps going into the top-right flash. Both are caused by longer distance from the IO pads, but are still below the 50 mV cap. As for the standard cells, the worst IR drop is measured at about 20 mV, located around the lower part of the chip. This is again due to the farther distance from the LDO located at the chip topmost. Verification of design constraints such as timing is performed after every step to ensure that there are no violation carried to the next design stage.

During the RUMPS401 CTS process, the clock optimization command was run in few optimization stages based on the verification result. After the initial CTS step, an optimization is performed for congestion and DFT, which result are checked with the global route congestion map. Re-optimization towards congestion were not performed as there is almost no congestion after the CTS. At this stage, the hold time constraint at 16 MHz clock path was violated by 0.02 ns and fixed by running the CTS optimization with the hold time prioritized, followed by the actual clock nets routing. Check on the congestion and timing were performed between every stage.

5.3. Placement optimization and clock tree synthesis (CTS)

At this point, the design physical layout has its standard cells, IP blocks, IO pads, and power straps in place. Placement of standard cells is optimized with ICC automation feature, in which the optimization parameters such as timing critical paths can be specified. The optimization groups together cells that belong to certain modules. It is intended to minimize the length of connection among cells, to reduce congestion, and to make the design routable.

Clock tree is a clock network that sources from a single IO pad and branches to every flip-flop in the design. Clock tree synthesis process attempts to generate the network with minimum clock skew, in which the clock signals arrived on every flip flops as evenly as possible. If required, buffers are placed to minimize clock delay on longer wire. This step is carried by running ICC automated CTS feature which works based on the DC timing constraint. The RUMPS401 clock tree is synthesized with 0.35 ns clock uncertainty derived from the design constraint defined during netlist synthesis. The CTS process is split into three steps. In the first step, the clock tree is synthesized without any optimization preferences. The second step optimizes CTS in terms of cell congestion and DFT. This step includes 1–2 rounds of optimizations to fix hold time violation. The actual routing of clock tree nets is performed in the last step, preceded by synthesis of reset signal using the CTS tool with the same constraints as the clock signal and target skew of 0.05 μs.

5.4. Routing

In this step, real wire connection among cells are made based on the logical connection defined in the netlist. Connections are formed via metal layers whose number depends on the foundry process. Prior to the ICC automated routing process, designer is required to specify certain parameters such as the routing priority, routing computational power. The automated routing process is done in three incremental steps. Leakage-power is the optimization target for the initial routing step. The second routing step optimizes further by attempting to optimize crosstalk reduction, while the final step cleans up any design-rule-checking (DRC) violations. DRC defines set of rules to be adhered by designers, such as minimum metal width, metal-to-metal spacing, antenna.

After every routing step, timing checks must be performed again to ensure that the timing condition is still met. Should it be violated, the routing may be carried again with timing optimization as priority. Generally, the design may not be clean of DRC violations after the first iteration of routing. The third routing step may be performed iteratively until the DRC is clean. Should there be violations that cannot be cleaned by the automated routing, they must be fixed manually by the designer.

At any point of the routing process, the physical design can be exported into physical netlist which is used for logical verification against the front-end design as well as timing analysis. Should there be a problem, changes are made on either front-end or back-end part. It may be big changes yielding to the repetition of whole back-end process, or small changes that can be solved by Engineering Change Order (ECO) process.

5.5. Engineering Change Order

This step is commonly performed when the back-end design is nearing its final step and changes in the design is minor. ICC provides ECO process where the changes can be accommodated by spare cells prepared by front-end designer. These cells are already defined in RTL level and takes form of common standard cells which are placed inside the core area but has no connection other than power. Should an ECO process occur, ICC will perform place optimization and minor re-routing to connect the design to these spare cells based on the ECO netlist.

5.6. Post-layout Static Timing Analysis, equivalent check and simulation

After the physical design of the RUMPS401 MPSoC is completed, Resistance and Capacitance (RC) characteristics can be extracted from the metal routes. These RC numbers provide more accurate timing delays across the signal paths. Another round of Static Timing Analysis is performed to consider these RC delays information. Adjustment is done to the physical design to eliminate any reported timing violations. In the first STA run, many hold time violations are reported, especially in the scan chain test mode. These are due to the direct path between the output pin of the flip-flops to the scan input of the next flip-flops. The parasitic RC delays affect the clock skew and the path delays. However, the violations are very minor and can be in the fractions of nanoseconds. Delay cells insertions are done to fix these violations. After all violations are fixed, the final timing reports are then used for design sign-off.

In this stage, equivalence check is required to ensure the whole back-end design processes do not alter the functionality of the chip. The post-layout netlist of RUMPS401 is extracted from the physical design. It is analyzed and compared against the gate-level netlist from DC as the reference design.

Post-layout simulations are performed as the final verification process of the RUMPS401 design. The post-layout simulations have very similar setup to the gate-level simulations. The extracted post-layout netlist is used as the DUT. RC characteristics of the design are also included to the simulation. Post-layout simulations verify that the design is still able to perform all the specified functionality with the additional parasitic delays added to the signals path. The final post-layout netlist of RUMPS401 chip with the parasitic timing delays manages to pass all the functional tests.

5.7. Chip finishing

To prepare the design for tape-out, there are few necessary finishing steps that must be applied to the physical layout. Filler cells are inserted into the layout to fill gaps among standard cells due to the manufacturing process requiring continuity of silicon within certain amount. IO pads fillers are inserted as well for the power ring continuation along the IO pads. To improve the routing wire quality, route tracks are widened and redundant via are created. After the finishing, various checks are run against the layout to ensure that it is free of any violation such as timing and DRC. The clean layout is then exported into a layout file (GDSII). Up to this point, the back-end design processes are performed in ICC tool environment.

The next step in chip finishing are performed Cadence's Virtuoso and Calibre DRC/LVS tools. They are required to produce a complete GDSII layout file which contains the detail of every silicon masking layer as well as their geometry, which the foundry will refer to for actual fabrication. Virtuoso is used to perform merging between the ICC-exported GDSII and the complete layout of every cell, IO pad, and IP block used in the design. In ICC, those instances are abstracted as block with IO pins to reduce the processing load required by ICC during place and route process.

The layout is then checked against DRC rules with Calibre tools. This DRC check is generally the same as ones performed in ICC, but Calibre has been widely used as industrial standards hence it is more common to find foundries requiring designer to do the final DRC check with Calibre ruleset. Layout-versus-schematic (LVS) check is run to compare netlists produced at the front-end and back-end for their logical equivalence. Once the layout passes the checks, it is ready to be sent over to foundry for fabrication.

5.8. General view of the RUMPS401 physical layout

Figure 4 illustrates the RUMPS401 chip and physical layout in general. SRAM and flash are two largest IP blocks in the design, and they are placed on each end of the core area. Below and above the pairs are arrays of power switches, used for controlling power going into the flash. The RUMPS401 allows independent core sleep, in which it will turn off the flash to

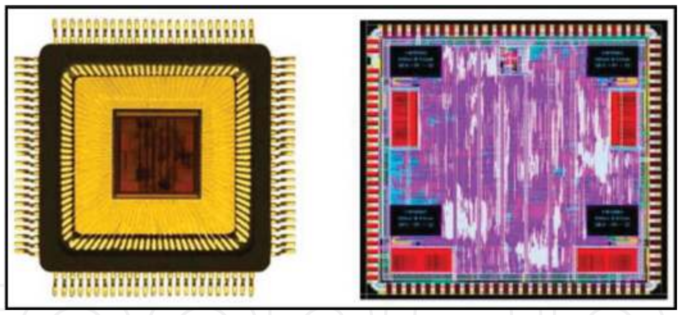

Figure 4. RUMPS401 chip (left) and RUMPS401 physical layout (right).

conserve power. Main power bus runs around the core area, supplying 1.8 V power source for standard cells. The 1.8 V is sourced from an internal 3.3 V-to-1.8 V low dropout voltage regulator (LDO) that receives 3.3 V power externally via IO pads. The 3.3 V power source is also used directly by the flash memory.

6. Result and application

Throughout the first-time success ASIC design methodology, check is performed on each design step to ensure that the design is free of error all the way to the fabricated silicon. When the physical chip is produced, proper testing procedure is performed first to verify that every gate in the silicon works, which includes but not limited to flash, RAM, and scan chain test. All gates are functioning verified by the scan-chain test. During normal operation, the RUMPS401 runs on the 16 MHz external clock source, consuming about 30 mA with all the cores running. The IO control core, normal cores, and DSP core individual current consumptions are at around 4, 7, and 4 mA, respectively. The NoC itself consumes about 10 mA and remains functional if at least one core is active. In a deep sleep mode all cores are put to sleep, the RUMPS401 switches to internal 32 kHz clock source and consumes only 13 μA. Ultimately, the goal of this thorough first-time success methodology is to have the chip working and performing its functionality on the first fabrication attempt. Hence, it is without doubt that the chip has been tested for real world application as a proof to the design methodology.

6.1. Software-defined radio

Software-defined radio (SDR) has been widely accepted as a solution for accommodating the availability of various yet ever evolving wireless standards. By shifting signal processing to programmable digital processors, a single radio transceiver can be reconfigured to operate on various standards without the need of hardware change. This is a desirable property for numerous applications, including Internet of Things (IoT). Industry players are competing on establishing wireless standards aimed for IoT, such as LoRa-WAN, SigFox, Bluetooth LE. During its lifespan, an IoT system may want to migrate to different wireless standard. As the number of nodes grow, replacing the transceivers would be problematic.

The RUMPS401 unique architecture offers an interesting platform for SDR implementation. Its low power and multi-processor architecture offers considerable processing capability if utilized properly whilst keeping power consumption to the minimum. Paired with a programmable radio frontend, the RUMPS401 can function as a complete IoT node, managing both SDR operation and transducers control. In [16], an SDR platform consisting of the RUMPS401 and programmable radio Lime LMS6002D was introduced. Implemented on the platform is an initial-stage transceiver software operating Turbo-coded, coherent binary phase-shift-keying (BPSK) modulation. Noting the absence of signal processing hardware from the RUMPS401 either in the ARM M0 processor or the peripherals, running coherent PSK modulation-demodulation and Turbo Code decoding is a challenging task for the MPSoC.

The Turbo Code software is implemented by dividing the algorithm based on its main tasks, i.e. the metrics calculation. Exploiting the heterogenous architecture, the tasks are split across the four cores based on each core's main functionality. The IO-control core is tasked with interfacing against the Lime LMS6002D and governs the sequence of data operation. The DSP core handles the computation of metrics that require extensive multiplications, while the two normal cores handle the rest of simpler but data-extensive metric computation. It was shown in [16] that such task structuring yields faster decoding compared to structure where each core performs more than one type of task due to the increased complexity on packet transfers among cores.

The coherent BPSK is configured to operate at a gross data rate of 2 kbps on 433 MHz carrier. Turbo encoding, pilot-bit stuffing and pulse shaping are applied to the data before transmission. The receiver performs pilot-aided timing and frequency synchronization on the incoming signal, then present the received data after passing it through the Turbo decoder. These functionalities are performed solely in the RUMPS401 via software, while the Lime LMS6002D only handles operations in radio frequency (RF) region.

Figure 5 illustrates the test setup with two SDR platform, one each for transmitter and receiver. Tests was carried in line-of-sight (LoS) configuration, in an empty room without any special conditioning. Excluding the pilot and Turbo code's parity bits, total of 10,000 pure data bits were generated and transmitted by the transmit side and checked on the receiver for its error rate. It is evident from **Figure 6** that the transceiver system exhibits the expected error rate behavior, as well as the Turbo Code that improves error rate by 20–40% on its third iteration.

This SDR platform has highlighted one of vast possibilities offered by the RUMPS401 as a low-power MPSoC on complex applications and serves as a proof of the first-time success

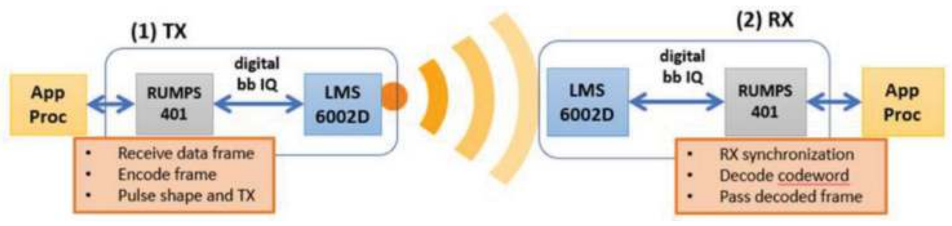

Figure 5. RUMPS401 SDR-based transceiver setup.

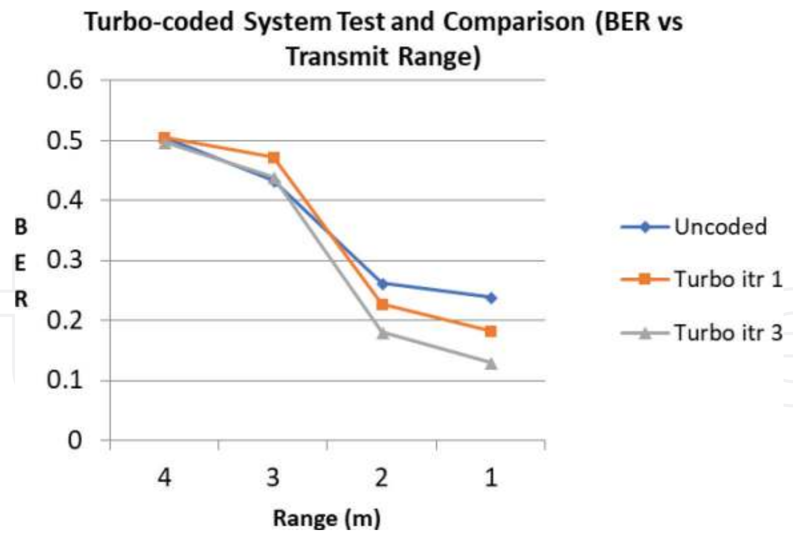

Figure 6. RUMPS401 SDR-based transceiver test result.

ASIC design methodology embraced by UTAR VLSI Research Center. Another example of RUMPS401 application is AES encryption as presented in [17]. The AES application has been successfully tested on the virtual prototyping platform and the RUMPS401 chip.

7. Conclusion

The first-time success RUMPS401 MPSoC has proven the proposed ASIC design methodology. The RUMPS401 is designed, implemented, and tested following the UTAR first-time success ASIC design methodology as presented throughout the book chapter. Design and verification processes are well defined and guided by the methodology which leads to first-time success. Corresponding EDA standard software tools are also mentioned including the use of UVM for functional verification.

Author details

Arya Wicaksana[1]*, Dareen Kusuma Halim[2], Dicky Hartono[3], Felix Lokananta[4], Sze-Wei Lee[4], Mow-Song Ng[3] and Chong-Ming Tang[3]

*Address all correspondence to: arya.wicaksana@umn.ac.id

1 Department of Informatics, Universitas Multimedia Nusantara, Indonesia

2 Department of Computer Engineering, Universitas Multimedia Nusantara, Indonesia

3 GLX Technologies, Malaysia

4 Department of Electronic Engineering, Universiti Tunku Abdul Rahman, Malaysia

References

[1] Kaeslin H. Digital Integrated Circuit Design. New York: Cambridge University Press; 2008. pp. 632-665

[2] ExtremeTech. What to Expect From Apple's Neural Engine in the A11 Bionic SoC [Internet]. 2017. Available from: https://www.extremetech.com/mobile/255780-apple-neural-engine-a11-bionic-soc [Accessed: 2018-06-03]

[3] Lusala AK, Manet P, Rousseau B, Legat JD. NoC implementation in FPGA using torus topology. In: 2007 International Conference on Field Programmable Logic and Applications. 2007

[4] Kharchenko VA. Problems of reliability of electronic components. Modern Electronic Materials. 2015;**1**:88-92. DOI: 10.1016/j.moem.2016.03.002

[5] Delta. SEM analysis reveals real cause of chip failure. Available from: https://asic.madebydelta.com/portfolio/sem-analysis-reveals-real-cause-of-chip-failure [Accessed: 2018-06-03]

[6] Accellera. Universal Verification Methodology (UVM) 1.1 User's Guide. California: Accellera; 2011

[7] Hartono D, Yap VV, Lee SW, Ng MS, Tang CM. A multicore system using NoC communication for parallel coarse-grain data processing. In: New Media Studies (ConMedia). 2013

[8] Dally WJ, Towles B. Route packets, not wires: On-chip interconnection networks. In: Proc Des. Autom. Conf. 2011. pp. 684-689

[9] Benini L, Micheli GD. Network on Chips: Technology and Tools. San Francisco: Morgan Kaufmann; 2006

[10] Wicaksana A, Tang CM. Virtual prototyping platform for multiprocessor system-on-chip hardware/software co-design and co-verification. In: Lee R, editor. Studies in Computational Intelligence. Vol. 719. Cham: Springer; 2011. pp. 93-108. DOI: 10.1007/978-3-319-60170-0_7

[11] Hartono D. The design and implementation of a scalable multi-processor system-on-chip using network communication for parallel coarse-grain data processings [dissertation]. Malaysia: Universiti Tunku Abdul Rahman; 2014

[12] Marculescu R, Ogras UY, Peh LS, Jerger NE, Hoskote Y. Outstanding research problems in NoC design: System, microarchitecture, and circuit perspectives. IEEE Transaction on Computer-Aided Design of Integrated Circuit and Systems. 2009. pp. 3-21

[13] Moadeli M, Maji P, Vanderbauwhede W. Quarc: A high-efficiency network on-chip architecture. In: International Conference on Advanced Information Networking and Applications. 2009. pp. 98-105

[14] Chatha KS, Srinivasan K. Layout aware design of mesh based NoC architectures. In: National Science Foundation and Consortium for Embedded System. 2006. pp. 136-141

[15] Becker DU. Efficient microarchitecture for network-on-chip routers. In: Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy. 2012

[16] Halim DK, Tang CM, Ng MS, Hartono D. Software-based turbo decoder implementation on low power multi-processor system-on-chip for Internet of Things. In: Proceedings of 2017 4th International Conference on New Media Studies (CONMEDIA); 8-10 November 2017. Yogyakarta: IEEE; 2018. pp. 137-141

[17] Wicaksana A, Tang CM, Ng MS. A scalable and configurable Multiprocessor System-on-Chip (MPSoC) virtual platform for hardware and software co-design and co-verification. In: 2015 3rd International Conference on New Media (CONMEDIA); 25-27 November 2015. Tangerang: IEEE. p. 2016