

A Proposal of Methodologies for Implementing Digital Chips in the Latest Processes

Hye-Seung Sun¹ and In-Shin Cho²

¹Department of Semiconductor Design, University of Korea Polytechnics

²Integrated Circuit Design Education Center, Korea Advanced Institute of Science and Technology

E-mail : shspoly@kopo.ac.kr, ischo@idec.or.kr

Abstract - In this paper, design methodologies suitable for implementing digital systems at various processes are suggested. Important issues such as Multi Corner Multi Mode, Hierarchical Design, adoption of CCS model, and changes in design flow must be considered for ultra-fine processes. The Cortex-M0 SoC Platform is implemented, taking into account these important issues, and the results using various digital libraries are compared.

All implemented platforms meet specifications and operate normally with both hardware and software. The fastest clock cycle that can be synthesized is 4ns for Samsung 28nm process.

Keywords – Methodology, Digital ASIC, Cortex-M0, Samsung 28nm

I. INTRODUCTION

IDEC (IC Design Education Center) has opened the opportunity to design chips using the 28nm process. There are several important changes with 28nm.

First, it is possible to design chips in a cloud server environment. Through the introduction of the cloud, data that is considered sensitive from the foundry company's perspective, including PDK (Process Design Kit) and digital libraries, can be stored and protected. It is possible to keep it safe, and designers do not have to spend expensive money to buy servers and install the OS, EDA tools, and license daemon themselves. Designers can design chips by simply connecting to cloud systems at any time.

IDEC, which must prepare and operate all cloud environments, has also been able to make great progress in security and network control. It is now possible to control users accessing EDA (Electronic Design Automation) tool licenses based on IP addresses. To prevent indiscriminate access to the cloud server, it is now possible to block everything except the IP address of the designer's PC. It is including the blocking of Internet and FTP communications.

When designing chips in a cloud environment, the smoothness of the network is most important. For this purpose, a dedicated line was installed to ensure that there is no decrease in speed. In general, if communication of about

20Mbps is guaranteed, there will be no problem for analog designers to work on schematics and layouts while looking at the screen. As the result, the introduction of the cloud is essential to enter the cutting-edge process, protect the intellectual property rights of the process company, such as network and EDA licenses. Moreover, integrating with existing infrastructure can be said to be the biggest change in the design environment.

Second, there is a change in the process library from the use of the NLDM (Non-Linear Delay Model) model to the CCS (Composite Current Source) model. Both models are delay models for standard cells created and distributed by foundry companies. The file extension of delay models for digital libraries is .lib, which contains the operating speed of a digital standard cell. It is based on statistics obtained through Spice simulation from semiconductor foundry.

In fact, when a PDK production team includes a lot of timing information during Spice simulation, the size of the .lib file will increase and become more detailed. The biggest issue is that the size of data can grow infinitely and it can make the Linux environment heavy when operating tools for chip implementation. Therefore, Cell Delay is commonly expressed in the form of a horizontal and vertical two-dimensional table for Input Transition and Output Load. Net Delay is expressed according to the size of the design area [1].

Details about Cell Delay are as follows. If a PDK production team reduces the difference between each value of Input Transition and Output Load in the Row and Column in the table, then the data to be included in .lib file becomes too large, so there is some difference between values. As a result, Cell Delay has the characteristic of being expressed as data in the form of a dotted line rather than a linear straight line. It is named Non-Linear Delay Model. Both Cell Delay and Net Delay information in .lib are created using statistical values, and NLDM is mainly used in three-digit processes around 110nm. However, it is recommended to use the CCS model in the 28nm process provided for Korea domestic universities. The NLDM model is based on voltage and CCS model is a current-based model. Because CCS model contains larger information than that of NLDM, a simple comparison of the file size shows that it is more than 10 times larger, and CCS makes the digital library capacity reach hundreds of gigabytes but it is recommended to use CCS model in 28nm processes because it is important to

Manuscript Received Feb. 21, 2024, Revised Mar. 14, 2024, Accepted Mar. 14, 2024

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/4.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

implement using the detailed information in latest processes.

Third, the specifications of the design server have become an important issue. Regarding the hardware specifications for each of the cloud servers mentioned above, in the past, it was possible to install Linux and design on a personal computer. However, when designing chips using the 28nm process, a workstation or server has become essential. An 8 Core CPU running at 2GHz and 16GB RAM is not suitable for design. If a designer considers a workstation or server, it is advantageous to use more than 16 CPU cores in multi-threading, 128GB or more of memory, and the larger the hard disk capacity. The clock frequency of the CPU needs to be around 3GHz. If ultra-fine processes of 14 nm or less are carried out, the hardware specifications of such design servers will become more important.

In this paper, based on the three contents described above, changes in the basic design flow and the introduction of the Multi-Corner Multi-Mode and Hierarchical Design methodology, which are essential for the latest processes and SoC (System On Chip) microsystems, will be described..

II. DESIGN METHODOLOGY

A. Changes in Basic Chip Design Flow

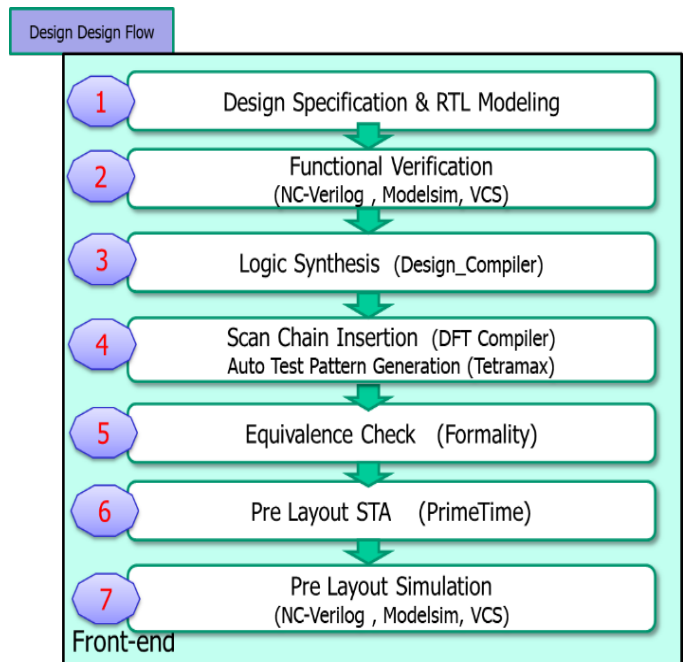
CADENCE, SYNOPSIS, SIEMENS EDA have digital design flows and EDA tools that encompass all processes. In addition to the three major companies, tools made by ANSYS and Scientific Analog are also necessary for chip design. Additionally, each foundry companies creates and applies its own in-house tools. As the number of essential tools increases and becomes more complex, the burden on designers trying to design chips is increasing. In universities, designers perform everything from ideation to front-end implementation and back-end implementation, so chip design is possible only with all levels of EDA licenses, PDKs, and digital libraries. Moreover, professors and students at universities cannot afford both in-house tools and expensive commercial tools due to limited budget, so the chip designer must set the appropriate standards and follow the guidance of the foundry company as much as possible.

The first thing to look at when setting standards is licensing. The chip designer must select a tool based on the license a designer can purchase and use right away. The willing of the tool vendor is also important in this regard. In the case of SYNOPSIS, CADENCE, SIEMENS EDA and Scientific Analog, a license equivalent to a commercial license has been provided to domestic universities for a long time. It has been supplied and is active in education, so it is easy for university professors and students to apply it to their processes. Recently, CADENCE is also required for all digital design stages. It is entering an era where people can design with tools that suit their needs, as licenses are provided.

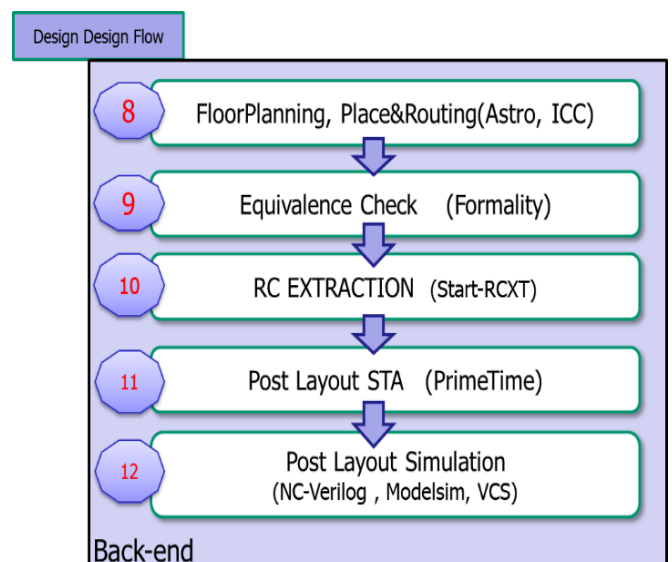
In addition to licensing issues, an important issue concerns the level of supply of PDKs and libraries from the foundry companies. Even if there is a tool a designer wants to use, the foundry company must supply the tech file or necessary information files needed for the tool. If the foundry company only supplies PDK in OA format, there is

no choice but to use only the CADENCE tool for analog design. In this case, if iPDK has not been developed or is not provided by the foundry company, a designer should give up using SYNOPSIS' Custom Compiler tool.

Fig. 1 shows the basic flow for digital chip design. It consists of a total of 12 steps, it can be seen as a result that satisfies all of the items described above. This basic flow can be applied up to the 65nm process at domestic universities.



(a)



(b)

Fig. 1. The basic flow for digital chip design (a) Front-End (b) Back-End.

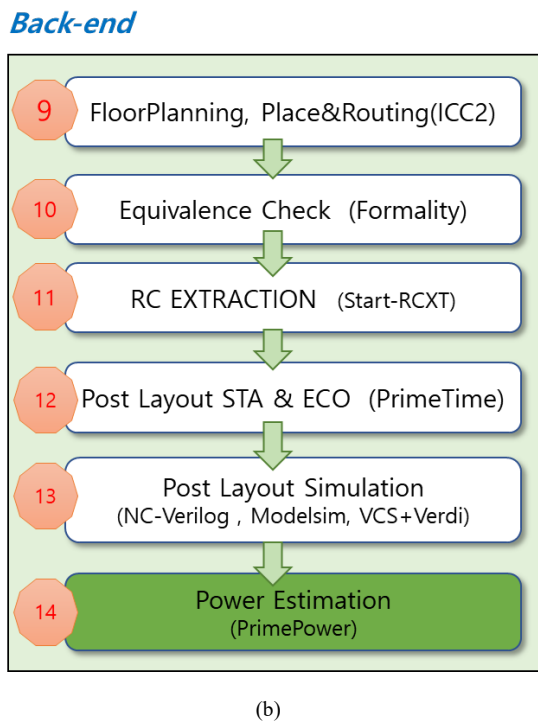
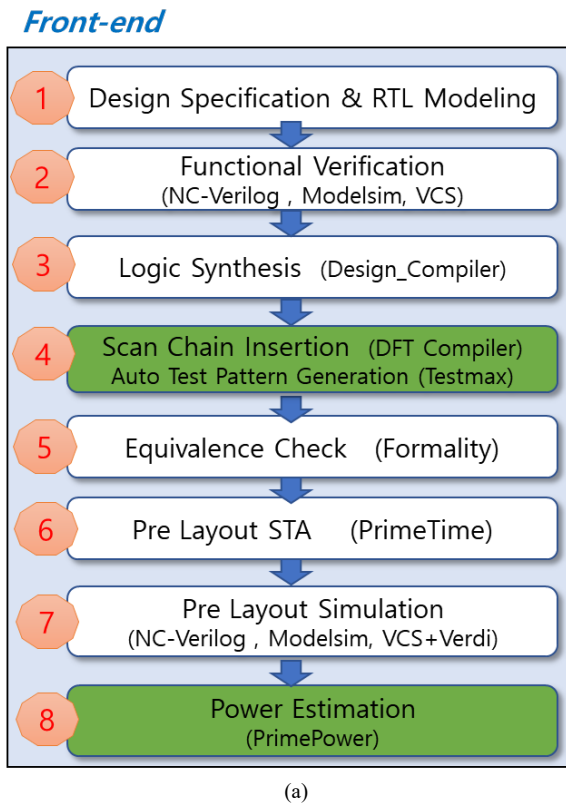


Fig. 2. The advanced flow for digital chip design at 28nm process (a) Front-End (b) Back-End.

Fig. 2 shows the chip design flow for the 28nm process. As a total 14 step, more tools are introduced. Necessary changes while advancing the 28nm process are in implementation tools. It is about synthesis and Auto PnR(Place and Route) and there are 4 important changes.

Firstly, changes in the synthesis tool are made by reading the Techfile and Tluplus files required for physical implementation. After synthesis, a designer can proceed with floorplan and macro placement. In the synthesis tool. The reason for using the synthesis tool up to the back-end area is the finer the process, the more important it is to consider timing based on the location of the macro or IP block at the synthesis step. In the case of SYNOPSIS Design Compiler, a designer can issue the floorplan command by executing the tool with the `dc_shell -topo` command. There is the other method for the same purpose. With the help of a license, ICC or ICC2 can be invoked from Design Compiler topographical mode and the location of macros or IP blocks can be specified at ICC or ICC2. Once finished Floorplan and Placing the macro blocks, the situation is brought back to the Design Compiler and incremental compilation can be performed with the value of location of Macro blocks to meet the timing requirement. In those cases, Design Compiler can move the positions of blocks automatically to improve timing.

The two methods described above are fundamentally for consistency between front-end and back-end. The timing was satisfied in the synthesis stage, but in the PnR stage, where the timing is not satisfied due to various reasons such as the figure of the floorplan, the location of IP blocks, congestion issues, etc. As the process becomes more detailed, it becomes more difficult to solve timing problem. Many designers need to consider design consistency and timing.

Secondly, it is about the Auto PNR tool. The biggest change is the new introduction of the SYNOPSIS IC Compiler 2. Since the ICC tool is scheduled to be discontinued, it is essential to use ICC2 at this time. It is a newly released tool for fine processing, and is known to have a performance about 7 times faster than ICC. It is necessary to prepare the library by converting it into a format called ndm rather than the existing Milkyway. An issue when switching tools is that the ICC tool script cannot be applied to ICC2 and must be created anew. The ndm format can be converted using a lef file, and the related conversion script set can be referenced by downloading the Reference Methodology from the solvnet site. Scripts required for tool operation can also be downloaded and referenced from the solvnet site [2].

The Auto PnR tool is also available in CADENCE, and the extension of the back-end library distributed by foundry companies is mainly LEF. The CADENCE tool, innovus is also good to apply to the Auto PnR stage, but the tools for the next stages, Parasitic Extraction, STA, and ECO, mainly use the SYNOPSIS tools. There were compatibility issues with tools from different vendors, and there were also shortcomings in whether or not licenses were provided. Recently CADENCE tool licenses have begun to be fully provided, it is expected to improve.

Thirdly, there is a noticeable change in power estimation at the 28nm process. Traditionally, the method of measuring power by converting wave-shaped simulation results into VCD format and then extracting them into saif format is widely used. Recently, the fsdb format, an improved compression method is used. SYNOPSIS PrimePower is

used as a tool to measure power consumption. Dynamic Simulation includes Function Simulation, Pre-Layout Simulation, and Post-Layout Simulation and each saif file is extracted and used as inputs of PrimePower. Because the setting procedures of PrimePower are the same as PrimeTime, existing Primetime users can operate it without much difference. Moreover, using saif file means that the designer has established a test bench or test scenario, so the power estimation method is closest to the actual driving situation of the fab-out chip.

Forthly, the most important issue in the latest process is test coverage. When a designer relies on human-created test scenarios and test benches to verify the design, it is widely known that full coverage of the design is not possible and the chip finally malfunctions. Moreover, as the process becomes more advanced, the size of the system that can be integrated increases, and it is reached an era where about 1 billion gates can be integrated, so a new special verification method is needed. Recently, it has become essential to create a test bench using System Verilog and verify it by applying UVM. The tools used for this are Verdi and VCS tools from SYNOPSIS. Since VCS is a tool for compilation and simulation, and Verdi is a debugging tool, it is recommended that designers use them as the essential tools.

B. Multi Corner Multi Mode

The digital library for the 28nm process includes many .lib files. This means that it contains a very large number of corner information. When the chip designer sets the specifications for the chip, it is needed to think more about how many operating conditions will be satisfied. The TIV (Temperature Inversion) corner must be considered, and it must be decided whether to make a commercial chip or a military chip. The chip designer should consider whether to select the best Corner, which considers situations where high voltage is applied, and conversely, whether to select the worst corner, which considers situations where low voltage is applied.

```

set scenario mode_norm_0C_rvt_ss_0p900v_125c
create_scenario ${scenario}
# Read in scenario-specific constraints file
puts "RM-Info: Sourcing script file [which [dcrmm_mcm filename ${DCRM_CONSTRAINT
S_INPUT_FILE} ${scenario}]] \n"
source [dcrmm_mcm filename ${DCRM_CONSTRAINTS_INPUT_FILE} ${scenario}]
set_operating_conditions ss_0p900v_125c -library sc9_cmos28lpp_base_rvt_ss_nomin
al_max_0p900v_125c_sadhm -analysis_type on_chip_variation
set_timing_derate -early 0.964
set_tlu_plus_files -tech2itf_map ${MAP_FILE}
                    -max_tluplus $TLUPLUS_MAX_FILE #
                    -min_tluplus $TLUPLUS_MIN_FILE #
                    -tech2itf_map ${MAP_FILE}
check_tlu_plus_files
    
```

Fig. 3. An example of Multi Corner Multi Mode.

Additionally, a test mode must be created by adding testing-consideration coding to the synthesized netlist or verilog RTL source code. If the design considers a multi-power domain, a power idle mode must also be created. The guide to the latest process is that it is necessary to define multi-scenarios. Construct a scenario with several modes

and corners, if a designer combines all the constraints considering the setup for each scenario, whether hold is considered, and the derate value presented by the engineering company, MCMM is completed. The designer can check this through the Fig. 3.

Considering the derate value means that there is a faster corner or a slightly slower corner in one corner by considering OCV (On Chip Variation). This makes the synthesis tool to consider that there are more corners. This means that variation is taken into consideration when calculating timing. If the foundry company provides an AOCV (Advanced OCV) model, it is a model to which the derate value has already been applied, so the designer can easily apply OCV by reading it in the tool.

If multi-scenarios are created using MCMM, they are useful at the synthesis stage. When examining synthesis results, designers should check for timing issues across all scenarios. If there are no problems, all scenarios can be passed on to the Back-End stage as is. In the case of the Design Compiler tool, there is a convenient way to transfer to ICC2 through the `write_icc2_files` command.

Technically, there are the biggest difficulties in the post-layout STA and ECO stages. It is common to prepare the number of folders for running PrimeTime, the first and second sign off tool, to match the number of scenarios. If there are three scenarios in total, STA using PrimeTime must be run in three folders individually. If one or more scenarios do not meet the setup and hold time requirements, layout will be modified. While trying to solve the hold time issue, the design may encounter a setup time issue, so many iterations will be required.

C. Hierarchical Design

In the case of universities, if some designers participate in the 28nm process, they can be allocated an area of 4mm x 4mm. It is not recommended to implement all designs in such a large area at once using the flatten method. Each operation unit must be implemented in modules or IP units and then proceed in a bottom-up manner. A simple design is shown in Fig. 4. From the top module's perspective, the area where hierarchical design will be performed must be defined.

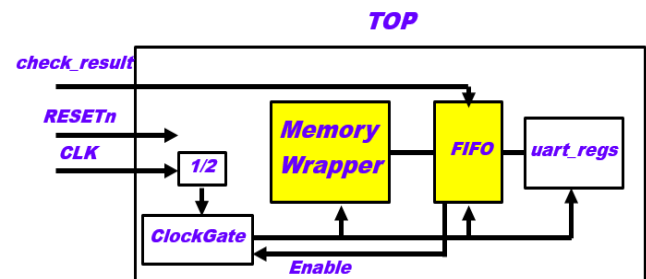


Fig. 4. An example of hierarchical design.

There is a memory wrapper which is decided to implement containing memory and a simple logic module in advance and they are marked in yellow. From the top module's perspective, MCMM scenarios will be selected and operating conditions will be selected for each scenario. Accordingly, the constraints of the top module will be

created, and thereby the constraints from the unit module perspective will be created naturally. At this time, the designer must consider the relationship between the top module and submodules. For example, if the main clock is 50MHz, the clock which is divided by 2 enters each submodule, so when creating constraints for the submodule, the clock must be declared as 25MHz. Also, from the sub block's perspective, the input and output ports are pins connected through the internal net from the TOP module's perspective, so the submodule should be implemented bottom up. It is important that once combined in the top module, constraints on submodules are no longer needed. This is because when implementing the top module, constraints from the top module perspective are only needed. However, the implementation of the sub design itself is used without change. So the commands **reset_design** and **set_dont_touch** are important to use.

Fig. 5 shows the sequence to implement bottom up. The designer can proceed from FE to BE of the submodule and check at a higher level. Alternatively, the designer can proceed with FE only and first include the approved ddc file in the FE stage of the higher module and monitor it in the BE stage. From the top module's perspective, it is important to ensure that there are no timing issues when connecting the pre-implemented submodules. To pass the setup time violations, the distance between submodules must be close. In order to implement it closely, it is advantageous to implement the top module itself in a small size. Additionally, in order to pass both setup and hold time violations, the buffer must be able to enter even in close situations. so a guide to proper sizing is to keep PnR utilization around 70%.

Fig. 6 shows another issue in the PnR phase. By comparing the two pictures, it is obvious that it is advantageous to connect the submodules through a simple and short path when placing them. Therefore, the first thing a designer should do is to draw an overall picture from the perspective of the TOP module or upper module. It will be helpful in determining the location and number of pins for close connection, and predicting and designing the locations of many modules in advance will make designs clear. Conversely, if a designer does not draw a picture, the location and number of pins are not specified in advance, so the path between modules will eventually become longer, causing timing issues and antenna errors.

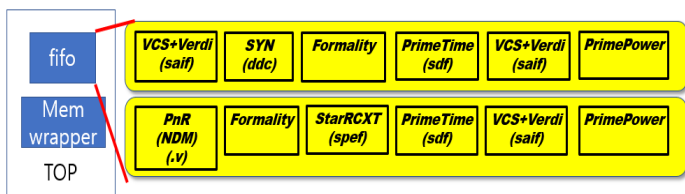


Fig. 5. The sequence to implement bottom-up.

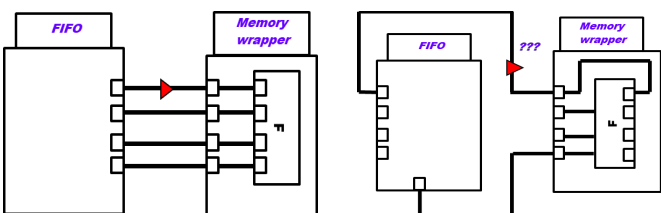


Fig. 6. Differences between the shape of sub modules.

Fig. 7 shows the implementation purpose of the memory wrapper. During operating ICC2 at 28nm process, it is realized that the connection to memory is difficult. In many cases, pins with adjacent 1'b1 or 1'b0 values are applied in common, but if routing is performed without any control, a pattern may be formed inside the memory after digging into the memory macro block. This phenomenon can commonly appear in designs that use macros. Since memory is used a lot in recent AI designs, creating a memory wrapper for each type of memory used will help in design by preventing memory routing issues in advance [3].

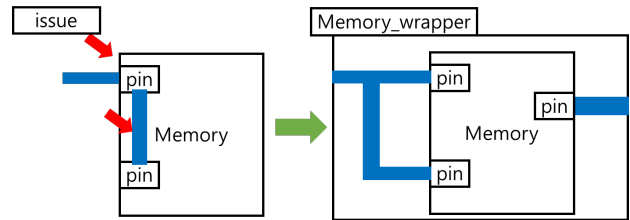


Fig. 7. Purpose of the memory wrapper.

III. RESULTS AND DISCUSSIONS

In this paper, SoC implementation based on Cortex-M0 will be covered. Cortex-M0 is supported through ARM University Program (AUP) and the SoC Platform can be received in partially encrypted Verilog Language. According to the enclosed user guide, the Cortex-M0 core was developed to operate at 50MHz on the TSMC 180nm process. The encrypted part is the decoder part of the core, and the remaining parts and the AMBA bus and peripheral controllers are implemented in Verilog language, making them suitable for implementation and simulation in ASIC [4]. The SoC Platform includes Cortex-M0, AHB, APB bus, GPIO, UART, Timer, and Watchdog. Internal RAM and ROM are provided in the Behavioral model, so when proceeding with ASIC, they must be replaced with memory provided by the foundry company. The entire package also includes a software development environment for running the Cortex-M0 SoC. Using the *u*vision tool created by Keil, developers can compile software codes that can test interrupts, timers, UART, Watchdog, etc. to create the final hex file. This hex file can be used for simulation, and if there is implemented hardware, it can be downloaded to the on-chip memory and then run the system.

Fig. 8 and TABLE I show the implementation results using digital libraries of Samsung 65nm, Samsung 28nm, and SYNOPSIS 28nm. The fastest clock cycle that can be synthesized is 12ns for Samsung 65nm, 7ns for SYNOPSIS 28nm, and 4ns for Samsung 28nm. This was applied as is when creating constraints.

For all three processes, hierarchical design was performed by implementing the memory wrapper first and then using it in the upper module. The biggest reason why the implemented areas are different is because the sizes of the memory and standard cells included in the digital library are different. In Samsung 65nm, it is clear that the RAM and ROM are large in size, but in Samsung 28nm, you can see that they are all contained in a small Core area.

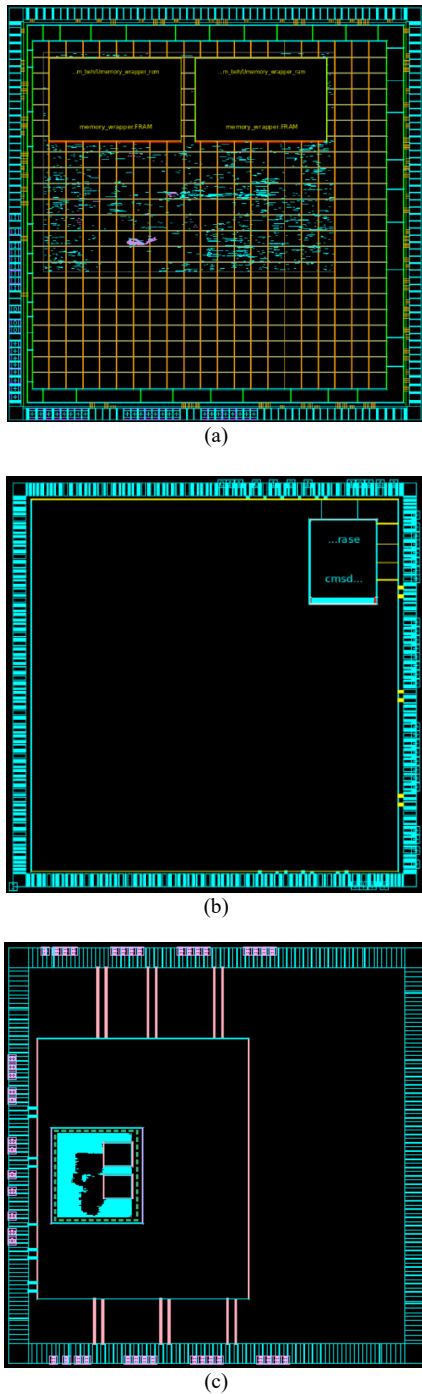


Fig. 8. Implementation of ARM Cortex-M0 (a) Samsung 65nm, (b) Samsung 28nm, (c) SYNOPSISYS 28nm.

TABLE I. Results of ARM Cortex-M0 implementation between various processes.

Process	Operating Frequency	Area(um)	Adoption of MCMM
SS65	12ns	2189 x 2118	X
SS28	4ns	650 x 767	O
SYNOPSISYS 28	7ns	760 x 860	O

MCMM is adopted at Samsung 28nm and SYNOPSISYS 28nm. MCMM is a new way to apply constraints and helps take more considerations into account during synthesis and PnR. CCS model is adopted at Samsung 28nm. The CCS model is a current-based model applied to Cell Delay measurement and Power Estimation and contains more detailed information than the NLDM model.

When synthesizing ARM Cortex-M0 or performing Auto PnR, results of the **report_timing** command can show that many internal combinational paths are very long that developers must pay attention to the critical path. Because Cortex-M0 is a downloaded SoC Platform and cannot be modified because it has an encrypted Verilog RTL source.

When a designer use Design Compiler, improvement is possible through the latest options such as **-spg** option and **-retime**. The latest features are only possible with the support of a license, and developers should check the SYNOPSISYS documents for correct use.

It was confirmed that the above three Cortex-M0 SoC Platforms operate well based on the constrained clock periods and software operations such as interrupt, uart, and timer are performed without any problems.

IV. CONCLUSION

This research proposes digital chip design methodologies which are suitable for implementation of SoC are suggested. Important issues about Multi Corner Multi Mode, Hierarchical Design, adoption of CCS model, and changes in design flow must be considered at ultra-fine processes. Based on the proposed methods, Cortex-M0 platforms are implemented and verified using SYNOPSISYS tools. Since all chips made by applying the proposed flow worked, university designers who make digital chips must apply all steps from start to finish, even if it takes a lot of time.

ACKNOWLEDGMENT

The chip fabrication and EDA tool were supported by the IC Design Education Center(IDEC), Korea.

REFERENCES

- [1] Synopsys, Design Compiler User guide. [Online]. Available : <http://www.synopsys.com>.
- [2] Synopsys, IC Compiler 2 User guide. [Online]. Available : <http://www.synopsys.com>.
- [3] Park, J. W., & Jeon, D. S. (2020). Designing Neuromorphic Processor with On-Chip Learning. *Journal of Integrated Circuits and Systems*, 6(2).
- [4] ARM, Arm University Program. [Online]. Available : <https://www.arm.com/resources/education/education-kits>.



Hye-Seung Sun received the B.S. degree in information and communication engineering from Hanbat National University, Daejeon, Korea, in 2007, and the M.S. degree in computer engineering from the same university in 2009. His research interests include digital ASIC design flow, with a current focus on low-

power IC design for ARM Cortex SoC Platform.



In-Shin Cho received the B.S. degree in information and communication engineering from Hanbat National University, Daejeon, Korea, in 2005, and the M.S. degree in computer engineering from the same university in 2007. His research interests include analog ASIC design flow, with a current focus on

low-power IC design.