

Evaluation of the Second FPGA-IP Prototype for MEC Devices and its Improvements

MORIHIRO KUGA[†]
SEN GEN[†]

MASAHIRO SUMITA[†]
MASAHIRO IIDA[†]

Abstract: Multi-access Edge Computing has become increasingly important in the IoT environment. By integrating eFPGA within MEC devices, it becomes possible to accelerate processing effectively based on specific application’s demands. This paper reports the evaluation results of the 2nd FPGA-IP prototype and its improvement policy to avoid the current issues.

Keywords: MEC (Multi-access Edge Computing), embedded FPGA, FPGA-IP

1. Introduction

Multi-access Edge Computing (MEC) has become increasingly important in the Internet of Things (IoT) environment. MEC aims to offload some of the processing at edge nodes to distribute the load across the entire environment. In particular, Field Programmable Gate Array (FPGA) is highly valuable for MEC applications due to its ability to reconfigure circuits according to user requirements. By integrating embedded FPGA (eFPGA) within MEC devices, it becomes possible to accelerate processing effectively based on specific application’s demands.

This paper reports the evaluation results of the 2nd FPGA-IP prototype and its improvement policy to avoid the current issues.

2. FPGA architecture and CAD flow

2.1 FPGA architecture

We have prototyped two FPGA-IPs, named TEG1[1] and TEG2, for MEC devices. TEG1 was the first prototype and TEG2 was 2nd prototype based on TEG1, enhancing the dedicated ripple carry adders, multipliers, and the number of I/O ports.

TEG2 has a configuration where the logic tiles (TILE) are arranged in a 16x16 array shown in Fig.1. Around the perimeter of each logic tile, there are 16 I/O Blocks (IOBs) positioned on

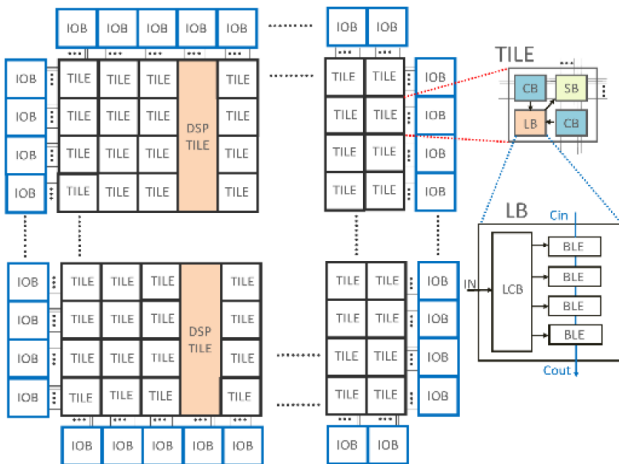


Fig.1: Overview of the 2nd FPGA-IP prototype named TEG2. all four sides, enabling input and output operations for the implemented logic circuits on the FPGA. The TILE section consists of a Logic Block (LB), a Switch Block (SB), and two Connection Blocks (CBs). All of these components are placed on the tile. In 2nd prototype, the TILES are arranged in a 16x16 array, except for the DSPTILE columns.

The LB is composed of a Local Connection Block (LCB) and four Basic Logic Elements (BLEs) shown on the right side of Fig.1. Each BLE includes a 5-input Scalable Logic Module (5-SLM)[1], a full adder, and a D-FF (Flip-Flop) shown in Fig.2. SLM is a logic cell architecture with reduced configuration memory (CF) compared to LUT (Look Up Table) with the same number of inputs. By connecting BLEs in a carry chain, a dedicated 4-bit ripple carry adder (RCA) can be constructed. Furthermore, by connecting tiles, it is possible to build a multi-bit RCA. In addition, to accommodate a dedicated multiplication block, TEG2 incorporates an unsigned 8-bit multiplier unit as a processing element in DSPTILE, placed in the 3rd and 13th columns total of 8 modules. The multiplier takes two 8-bit inputs and produces 16-bit outputs within the area of four tiles, as shown in Fig.1.

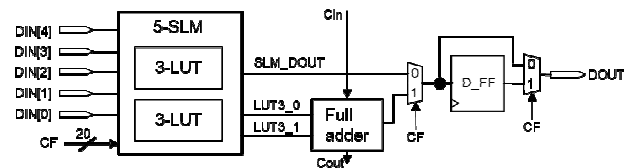


Fig.2: Structure of BLE.

2.2 CAD flow

We organized the CAD flow based on the VTR (Verilog to Routing) tool[2], tailored for TEG2 as shown in Fig.3[3]. First, we described the circuit in RTL (Register Transfer Level) and performed logic synthesis using ODIN II and Yosys, based on the specified architecture file. Next, we performed technology mapping using an improved version of ABC[4] that had a technology mapper for SLM. Based on the results of technology mapping, we proceeded with clustering and placement using vpr8.1 and vpr8.0. Finally, EasyRouter[1] carried out to route the interconnection of the logic resources and finally to generate configuration data for FPGA-IP.

[†] Kumamoto University, JAPAN

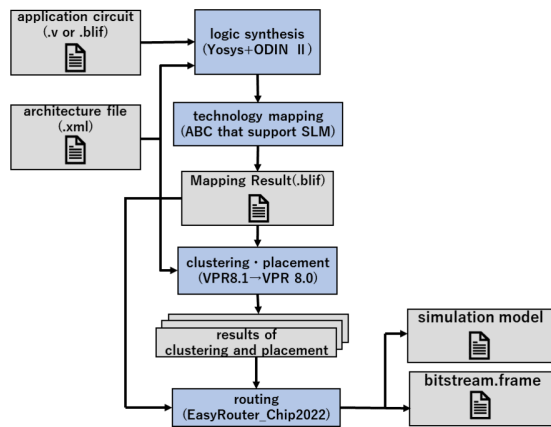


Fig.3: CAD flow for TEG2.

3. Evaluation results of TEG2

The TEG2 was an embedded FPGA-IP in a prototype chip named SLMLET[5]. The chip was fabricated by USCJ DDC (Deeply Depleted Channel) 55nm process. Two FPGA-IPs were embedded in the chip, and we confirmed that almost the function did the correct operation.

To evaluate the improvement of TEG2 from TEG1, we compare critical path delay between using the dedicated adder or multiplier and implementing the circuit solely with SLM[6]. This investigation aims to assess the extent of improvement in both adder and multiplier circuits. The test circuits used for comparison include 4-bit and 8-bit adders and an 8-bit multiplier. To find out the critical path delay with static timing analysis (STA), the critical path spans from the FF output holding input value to the FF input to hold the output value of each test circuit. To compare with resource utilization, we utilize the technology mapping report obtained from modified ABC[4] adapted for SLM. We used Prime Time 2019 03-SP3 by Synopsys for STA.

Table 1 shows the critical path delays in each implementation, where “SLM only” means the circuit implemented on only SLM resources, and “Dedicated HW” means the arithmetic operator mapped to the dedicated HW module, FAs with a carry chain for an addition, and DSPTILE for a multiply. In all test circuits, the critical path delay was reduced when dedicated HW was used compared to using only SLM, resulting in improved performance of addition and multiply operations.

Table 1: Result of critical path delay by STA.

Circuits	SLM only [ns]	Dedicated HW [ns]	Speedup
4-bit adder	3.702	2.579	1.435
8-bit adder	6.881	3.374	2.039
Multiplier	23.76	3.801	6.251

4. Improvement policy toward the 3rd prototype

During the evaluation process, we have identified issues that should be avoided in TEG2 architecture and its CAD flow. These issues were caused by the FPGA-IP architecture and CAD flow. Some major issues and these improvement policies are listed below.

- The SLMLET chip had two FPGA-IPs of 16x16 array, FPGA-IP1 was placed left side and FPGA-IP2 was placed

right side. Two FPGA-IPs were interconnected between FPGA-IP1’s right side I/Os and FPGA-IP2’s left side I/Os. Due to the layout of the IOB, the interconnection between the two FPGs had to be connected in bit-reverse order to ensure straight wiring. This layout led to an issue in the placing and the wiring processes. We will revise FPGA-IP with a large array such as a 32x16 size to avoid the issue.

- DSPTILE has an unsigned multiplier, but when implementing a signed multiplier, there is an issue that the implementation efficiency is decreased. Avoid this issue by implementing a signed multiplier for the DSPTILE.
- Although ODIN II and Yosys were used for logic synthesizer, some miss synthesis arose using a dedicated HW module when signed/unsigned multiplier and adder carry chain implementation. To avoid the issue, we will revise the logic synthesizer only to use Yosys utilizing its high functionality. Also, to enhance maintainability, we utilize the latest VPR 8.1 for the wiring process replacing EasyRouter.
- FPGA-IP2 could not be configured without lowering the core voltage. It looks like there was a hold timing violation in the layout of the configuration controller for FPGA-IP2. We should enough check with post-layout simulation.

5. Conclusion

In this paper, we introduced the FPGA-IP (TEG2) architecture and its CAD flow. The evaluation was conducted using STA. We confirmed that the critical path delay was reduced using dedicated HW compared to the only SLM implementation, resulting in improved performance. However, at the same time, we had identified issues that should be avoided in TEG2 architecture and its CAD flow. We summarized the issues and improvement policy toward the 3rd FPGA-IP prototype.

Reference

- [1] Morihiro KUGA, Qian ZHAO, Yuya NAKAZATO, Motoki AMAGASAKI, and Masahiro IIDA, “An eFPGA Generation Suite with Customizable Architecture and IDE,” IEICE Transactions on Fundamentals of Electronics, Communications and Computer Science, vol.E106A, No.3, pp560-574,2023.
- [2] J. Rose, J. Luu, C.W. Yu, O. Densmore, J. Goeders, A. Somerville, K.B. Kent, P. Jamieson, and J. Anderson, “The VTR project: Architecture and CAD for FPGAs from Verilog to routing,” Proc. ACM/SIGDA Int’l Symp. on Field Programmable Gate Arrays (FPGA’12), pp.77–86, 2012.
- [3] Sen Gen, Masahiro SUMITA, Morihiro KUGA, and Masahiro IIDA, “Adopting Open-source Logic Synthesizer for the original FPGA-IP,” Proc. 2023 Joint Conference of Electrical, Electronics and Information Engineers in Kyusyu, 09-2A-04, 2023.
- [4] Izumi KIUCHI, Yuya NAKASATO, Qian ZHAO, and Masahiro IIDA, “A Study on Technology Mapping Method for Scalable Logic Module,” IEICE Technical Report, vol.121, no.344, RECONF2021-76, pp. 108-113, Jan. 2022. (in Japanese)
- [5] T. Kojima, Y. Yanai, K. Okuhara, H. Amano, M. Kuga, and M. Iida, “Library Development for RISC-V FPGA SoCs,” IEICE Technical Report, RECONF2023-31, 2023. (in Japanese)
- [6] Masahiro SUMITA, Morihiro KUGA, and Masahiro IIDA, “Evaluation of the static timing analysis of an FPGA-IP prototype chip for MEC devices,” Proc. 2023 Joint Conference of Electrical, Electronics and Information Engineers in Kyusyu, 09-2A-03, 2023.