Tag-less compression for FPGA configuration data

Souhei Takagi†, Naoya Niwa†, Yusuke Yanai†, Hideharu Amano†, Masaki Amagasaki‡, Yuya Nakazato‡, Masahiro Iida‡
†Dept. of Information and Computer Science, Keio University, Japan
‡Faculty of Advanced Science and Technology, Kumamoto University, Japan

I. INTRODUCTION

This research proposes a configuration compression method for SLM (Scalable Logic Module) [1]and its decompression hardware. SLM is a new FPGA develpoed by Kumamoto University in 2014, and is characterized by its small size of configuration information. The SLMs are embedded in a chip called SLMLET jointly developed by Kumamoto University and Keio University in the CREST project "Development of multi-node integrated system for MEC". By integrating RISC-V CPU, memory, and SLMs in a single chip, multiple sets of configuration information are stored in the memory of the chip. By sending it from the memory to the SLM, switching logic circuits can be improved than that for common FPGAs. On the other hand, since the memory size in a chip is limited, we need to compress the configuration information as small as possible to store a large number of sets of the configuration data in the memory. FPGA configuration information is large and contains many iterative patterns, and thus, a method of compressing it has been researched since the 90's. As recent commercially available FPGAs have a powerful I/O mechanisms like PCIe for its host, they tend to rely its high throughput rather than data compression. Thus, most of the studies on configuration compression assume specific FPGAs [2] [3] [4], and the proposed methods are not suitable for the SLM which has also a specific architecture. Here, a simple compression method called tag less compression (TLC) is proposed for the SLM, and its compression ratio is evaluated. Considering the SLMLET chip implementation, a hardware cost is also evaluated.

II. PROPOSAL OF THE COMPRESSION METHOD

A. Tag Less Compression

For SLMLET, compression can be done outside the chip, but on-the-fly decompression is needed to be done inside the chip. Thus, the compression method with a simple decompressing hardware is required. Also, the compressed configuration data must be efficiently stored in the memory for RISC-V CPU. Considering these requirements, we propose a compression method called TLC (Tag Less Compression). Here, we explain with a 4-bit version shown in Fig. 1).

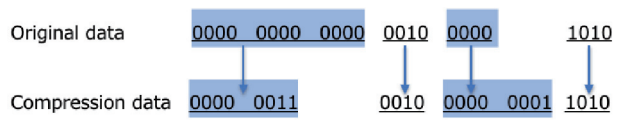

Fig. 1. Tag Less Compression (TLC)

TLC is a type of run length compression, which targets patterns with 4-bit continuous zeros. When more than four continuous zeros are found in the original data, '0000' is first outputted. After that, the number of consecutive patterns of '0000' is specified by 4 bits. In the example of Fig. 1, the pattern '0000 0000 0000' is compressed into '0000 0011'. '0011' shows the number of continuous '0000's. Patterns not the target of compression are outputted as they are. One of the features of this method is that it does not use tags. By not using tags, data are handled in 4-bit units and easy to be handled with most of CPUs. This ability facilitates both compression and decompression implementations. The problem of TLC is that it increases the data amount if '0000' appears only once as Fig. 1 shows. When comparing the compression ratios, we implemented and examined 3-bit and 8-bit versions in addition to the 4-bit version described above.

B. Evaluation of compression rate

During the evaluation, the compression ratio was measured for the circuits implemented on the actual SLM fine-grained reconfigurable logic. The target SLM has 256 tiles, and approximately 5k gates can be mounted as a whole. It was mounted on the first test chip developed in 2021, and now is working on the test board. Two SLM blocks have been mounted on the SLMLET chip currently being developed. We prepared multiple design samples to check the compression ratio for the SLM configuration information. "alu", "adder", and "counter" are simple and small logic, while "lpd", "tmul", "tdiv", and "div" are more complicated and large circuits. The tool for generating the configuration information is a combination of free software [5], and a tool developed by Kumamoto University. The compression method is a 3-bit, 4-bit, or 8-bit version of TLC. For comparison, we chose the 'gzip' on Linux as an example of Lampel Zip, and frequent pattern compression or FPC [6] as an example of a run length compression which allows on-the fly decompression. For reference, we also compared the compression ratio when they are applied to a common FPGA. We adopted Xilinx SPARTAN3-XC3S50, whose implementation process and available gate size are almost the same. Its design was implemented with Xilinx ise14.7.

All results are shown in TABLE I, and results of TLCs and FPC which can be implemented into a chip are picked up and shown in a Fig.2. Here, the compression rate is computed by (the configuration data amount)/(compressed data amount), so the larger is the better. Although 'gzip' is the best, it needs complicated hardware and hard to be implemented in a small chip like SLMLET.

Fig. 2 shows that the rate is better for simple designs, and degraded for complicated designs. It is because the complicated design includes larger number of 1's. However, the 3bit version and 4bit version of TLC are more resistant than other TLCs and FPCs. They achieved a certain compression rate even for complicated design 'ldp' which uses almost all resources on the SLM. Considering the easiness of implementation with the CPU, we adopted 4bit version of TLC for SLMLET.

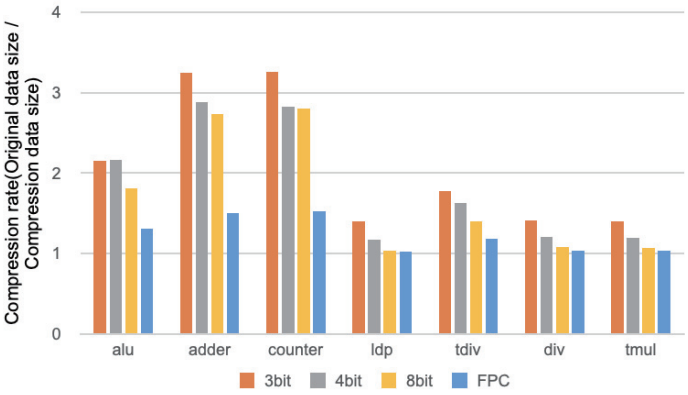

Fig. 2. Results of Compression

TABLE I. RESULTS OF COMPRESSION

original data	size(kbyte)	compression rate				
		3bit	4bit	8bit	gzip	FPC
alu	14.8	2.15	2.16	1.81	3.13	1.31
adder	14.8	3.25	2.88	2.73	15.7	1.50
counter	14.6	3.26	2.83	2.80	6.53	1.53
ldp	14.6	1.40	1.17	1.04	1.38	1.02
tmul	14.8	1.40	1.20	1.07	1.43	1.04
tdiv	14.8	1.78	1.63	1.40	2.11	1.18
div	14.8	1.41	1.21	1.08	1.46	1.04
counter(SPARTAN3)	55.0	3.63	6.71	25.4	36	1.83

III. HARDWARE IMPLEMENTATION

The logic circuit that decompresses the data compressed by the 4-bit version was described in Verilog HDL, and the operation was confirmed by simulating it with iverilog. In addition, assuming a USJC 55nm process, we used Synopsys' Design Compiler N-2017.09-SP1 for logic synthesis. For comparison, the decompression circuit for FPC [6] was also evaluated in the area and delay with the same process technology. Also, here we should check the elements required for decompression. Since only one bit of decompressed configuration information is read per clock, what is important is not the decompression speed but the ability to read without stopping.

A. Implementation of 4 bit TLC

The decompression algorithm executes the opposite of compression with the hardware shown in Fig. 3.

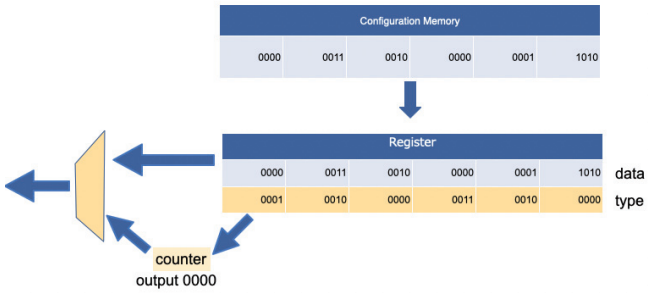

Fig. 3. Flow of Decompression on Hardware

The compressed data is read out from the memory into Register, assigned its type for every 4 bits, and the data is outputted in a serial manner. In SLMLET, the configuration information is stored in the 32-bit internal RAM, and given to the SLM core bit by bit in the same timing as the non-compressed data. As shown in the example in Fig. 3, type 0000 means that the data is transferred as it is, type 0001 shows that the counter follows, and type 0010 shows that it is a counter value for continuous '0000's. Based on the type, the data or continuous zero's are serially outputted. Since the pattern only one '0000' requires special operations, type 0011 is used for it. To cope with the case when the counter is carried over in the next word, the controller becomes a bit more complicated than shown in the figure, but still stays simple.

B. Hardware amount and delay

Since the target frequency is specified as 200MHz, a clock cycle is set to be 5000 psec. The results (TABLE. II) show that the extension circuit of TLC has a smaller area and a delay of 1.43 times shorter than that of the FPC. This result is due to the simplicity of the algorithm. Short delays are important for online elongation. As a result, the configuration time in SLMLET is expected to be 500 μ second to 1ms, much faster than that for common FPGAs.

TABLE II. HARDWARE AMOUNT AND DELAY

algorithm	amount (μm^2)	delay time(nsec)
TLC (4bit)	793.8	3.095
FPC	6476	4.422

IV. CONCLUSION

In this article, a novel configuration compression method called TLC is proposed, and its compression rate is evaluated. 4-bit version of TLC achieved more than 2 times compression rate for the configuration data for simple designs, and even for complicated designs it achieved a certain compression ratio. The rate is better than the compared run-length compression called FPC. Due to its simple operation, the area of TLC decompression circuit is only 793μm^2 of USJC 55nm CMOS process, and the delay is enough to work 200MHz clock cycle. The designed circuits were implemented on the real SLMLET chip which was taped out on this May and now under fabrication.

Acknowledgements

This work was supported by JST, CREST Grant Number JPMJCR19K1, Japan.

REFERENCES

[1] Qian Zhao, Kyosei Yanagida, Motoki Amagasaki, Masahiro Iida, Morihiro Kuga, and Toshinori Sueyoshi. A logic cell architecture exploiting the shannon expansion for the reduction of configuration memory. In *2014 FPL)*, pages 1–6, 2014.

[2] S. Hauck and W.D. Wilson. Runlength compression techniques for fpga configurations. In *7th FCCM*, pages 286–287, 1999.

[3] Xie Jing, Wang Yabin, Chen Liguang, Wang Jian, Wang Yuan, Lai Jinmei, and Tong Jiarong. Fast configuration architecture of fpga suitable for bitstream compression. In *2009 IEEE 8th International Conference on ASIC*, pages 126–130, 2009.

[4] K. Tanigawa, T. Kawasaki, and T. Hironaka. A coarse-grained reconfigurable architecture with low cost configuration data compression mechanism. In *Proceedings. 2003 ICFPT*, pages 311–314, 2003.

[5] Qian Zhao, Motoki Amagasaki, Masahiro Iida, Morihiro Kuga, and Toshinori Sueyoshi. Towards open-hw: A platform to design, share and deploy fpga accelerators in low cost. *IPSJ Trans. on SLDM*, 10:63–70, 2017.

[6] Naoya Niwa, Yoshiya Shikama, Hideharu Amano, and Michihiro Koibuchi. A case for low-latency network-on-chip using compression routers. In *2021 29th PDP*, pages 134–142, 2021.