

RCA on FPGAs Designed by RTL Design Methodology and Wave-Pipelined Operation

Tomoaki Sato¹, Member, Sorawat Chivapreecha²,
Phichet Moungnoul³, and Kohji Higuchi⁴, Non-members

ABSTRACT

Field-programmable gate arrays (FPGAs) are used in various systems with reconfigurable functions. Conventional FPGAs have been developed using a transistor level description for minimizing routing delay. Although FPGAs developed with a register transfer level (RTL) design methodology provide various benefits to the designers of a system-on-a-chip (SoC), they have not been realized. Therefore, the authors advanced their development. They should be shown to operate with practical throughput. For this purpose, circuits on these device need to be designed and evaluated. In this paper, a ripple-carry adder (RCA) was designed and the throughput of the RCA was evaluated. The resulting throughput was applicable to network processors. Additionally, a wave-pipelined operation without changing the RCA revealed that the problem of routing delay in FPGA developed by RTL methodology was mitigated. The contributions of this paper are to clarify that a 4-bit adder can be implemented on FPGAs and their throughput can be improved by wave-pipelined operations.

Keywords: FPGAs, RTL Design Methodology, RCA, Wave-pipeline, SoC

1. INTRODUCTION

A variety of equipment, such that used in networks [1] and firewalls [2] run using FPGAs because they are reconfigurable. These features enable changes in circuit configurations on the FPGAs as in a software program when it is required to add or modify a function. Therefore, if the processing speed, power consumption or cost is not suitable for processing in a central processing unit (CPU), the option of using FPGAs can be very useful.

To achieve a reconfigurable features in conventional FPGAs, a routing path is controlled by a tran-

sistor acting as a switch. This is significantly different from a SoC developed using a standard cell library. Therefore, a SoC circuit configuration cannot be changed. Alternatively, operations of FPGAs consume more power and work at a lower frequency than a SoC.

FPGAs can be used to solve these problems, [3-4] achieve high speed operation and low power consumption at the architecture and transistor levels. In [5], the problems of operating speed and power consumption of static random access memory (SRAM) used with FPGAs were described and solved. These studies are very useful in conventional FPGAs developed at the transistor level.

Moreover, the development of SoCs build in FPGAs is currently much in demand. It is essential to embed the FPGA at the RTL level for large-scale SoCs and to simultaneously shorten the design period. However, the routing path delays of FPGAs designed by a RTL design methodology are larger than conventional FPGAs. This is a reason why a study of FPGAs using RTL design methodology was not done.

Studies were done [6-9] to realize FPGAs designed by the RTL design methodology. The advantage is that the FPGAs themselves can be developed using a HDL (hardware description language). Additionally, circuits on FPGAs can be designed using a HDL as well as conventional FPGAs.

It should be made clear that these FPGAs are usable for practical applications. Thus, it is necessary to design and evaluate circuits on these FPGAs. In this paper, a 4-bit ripple-carry adder (RCA) was configured on an FPGA. The throughput of the 4-bit RCA was evaluated. Additionally, wave-pipelined operations were done without changing the circuit configuration of the 4-bit RCA and this contributes to an easing of the routing delay problem.

This paper is organized as follows. Section 2 presents outlines of CPUs and FPGAs used in packet processing. Next, Section 3 discusses FPGAs that are designed by the RTL design methodology. In Section 4, the 4-bit RCA circuit is designed on the FPGAs and wave-pipelined operation of this device is described in Section 5. Then, the 4-bit RCA is evaluated in Section 6.

Manuscript received on August 31, 2016 ; revised on December 23, 2016.

Final manuscript received on February 21, 2017.

¹ The author is with the Computing and Networking Center, Hirosaki University, Japan, E-mail: t-sato@hokusei.ac.jp

^{2,3} The authors are with the Faculty of Engineering, KMITL, Thailand, E-mail: sorawat@telecom.kmitl.ac.th and phichet@telecom.kmitl.ac.th

⁴ The author is with the Graduate School of Informatics and Engineering, UEC, Japan, E-mail: higuchi@ee.uec.ac.jp

Table 1: CPU Time in the CPUs of [16].

Frequency of the CPU	Minimum size frame (ns)	Standard size frame (ns)	Jumbo frame of 3000 KB (ns)	Maximum jumbo frame (ns)
500MHz	760.0	15280.0	30280.0	160080.0
800MHz	380.0	7640.0	15140.0	80040.0
1GHz	237.5	4775.0	9462.5	50025.0
1.2GHz	190.0	3820.0	7570.0	40020.0
1.5GHz	158.3	3183.3	6308.3	33350.0
2.0GHz	126.7	2546.7	5046.7	26680.0
3.0GHz	95.0	1910.0	3785.0	20010.0
4.0GHz	63.3	1273.3	2523.3	13340.0

2. NEEDS AND PROBLEMS OF FPGAS FOR PACKET PROCESSING IN MOBILE DEVICES

Packet processing in mobile devices should be executed at high speed with low power consumption. The use of an FPGA is a better solution in this regard. However, conventional FPGA devices have a problem in that a CPU developed by an FPGA designer cannot be used as an ASIC. In this section, CPUs and FPGAs in packet processing are explained. Subsequently, problems and solutions of packet processing on a mobile CPU are clarified based on our research results. Finally, the necessity of a CPU architecture that specializes in packet processing and an FPGA developed by the RTL design methodology is described.

High throughput of networks used in mobile devices has been advanced. In Wi-Fi LANs, the development of IEEE802.11ad was done. It will finally achieve a throughput of 6.8 Gbps [10-11]. Moreover, studies on 5G is advanced in mobile data communications. The purpose of the current study is to achieve data communication at rate of more than 10 Gbps [12]. Packet processing in mobile devices in such high-speed data communications is essential to ASIC or FPGA processing.

Super pipelining, parallel processing and processing that does not depend on the word size are easily achieved using a FPGA. Therefore, an FPGA is very beneficial for packet processing. In [13], a packet classification engine achieved a very high throughput by super pipelining on an FPGA. Moreover, the use of an FPGA is beneficial for low power consumption [14]. [15] reconfigured FPGA features for packet processing.

However, detecting processing due to unauthorized access needs not only a packet classification engine, but also complex processing. Circuits on an ASIC or FPGA for complex processing pose a problem that requires a huge amount of hardware. Additionally, it cannot take advantage of software resources for unauthorized access. That is, a CPU for packet and detection processing is required.

The authors estimated the performance of a CPU that is needed for packet processing [16]. According to the results, packet processing with a 1 Gbps throughput and normal data communications needs

a CPU with a MIPS64 5K architecture operating at 1 GHz. In contrast, packet processing of a repeating packet frame of the shortest size and a packet frame with a standard size needs a clock frequency of 4 GHz. Table 1 shows the CPU times required for packet transfer processing. If it is unable to accomplish such a process, it becomes vulnerable to a denial of service (DoS) attack.

In [16], to protect a CPU with 1 GHz operations from a DoS attack, out-of-order packet execution inside the CPU has been proposed. This function should be built inside the CPU as an ASIC circuit. Additionally, the CPU needs to be developed as an ASIC for low power operations because the processing speed and power consumption of the CPU influences the entire system. If the CPU has sufficient processing capacity for packet processing, it facilitates the change in the content of processing by software. In that case, reconfigurable features such as an FPGA are not required.

When incorporating such a unique architecture on an FPGA, the CPU was only on the FPGA as a soft-core processor. Furthermore, circuits as an ASIC are often needed for low-power consumption and high-speed processing. Therefore, an FPGA using RTL design methodology, as in this study, facilitates the development of CPUs and circuits as an ASIC. In other words, ASIC-FPGA codesign is realized.

3. FPGAS DESIGNED BY THE RTL DESIGN METHODOLOGY

In this section, details of the FPGAs that the authors developed are explained. Next, logic synthesis is executed to investigate delay times of the FPGAs. Developed environments are described.

Table 2: Design environments.

OS	Cent OS 5.9 x86
CPU	Intel Core 2 Duo E6600 (2.4GHz)
Memory	2 GBytes
Logic synthesis	Synopsys Design Compiler H-2013.03-SP2
Technology	Rohm 180 nm C-MOS
Standard cell library	The library provided by Rohm

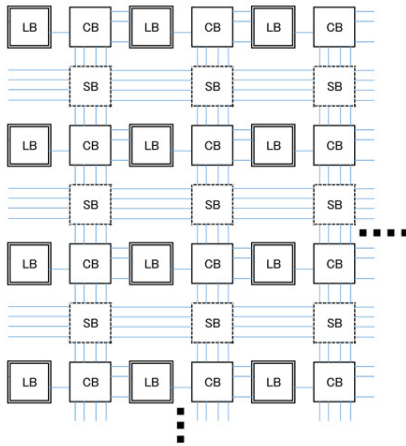


Fig.1: Architecture of FPGAs designed using RTL design methodology.

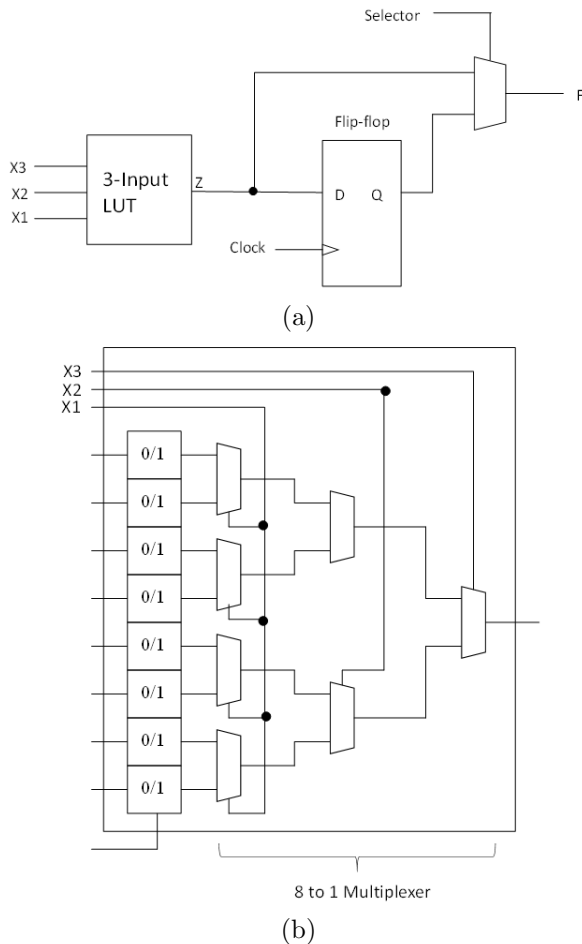


Fig.2: Structure of the LB. (a) Basic component and (b) 3-input and 1-output LUT.

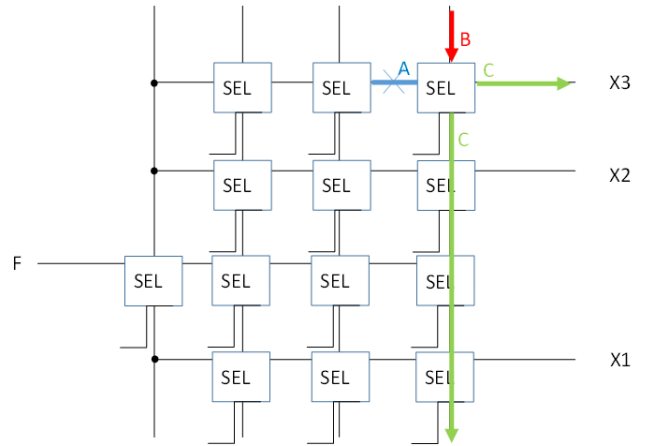


Fig.3: Structure of the CB.

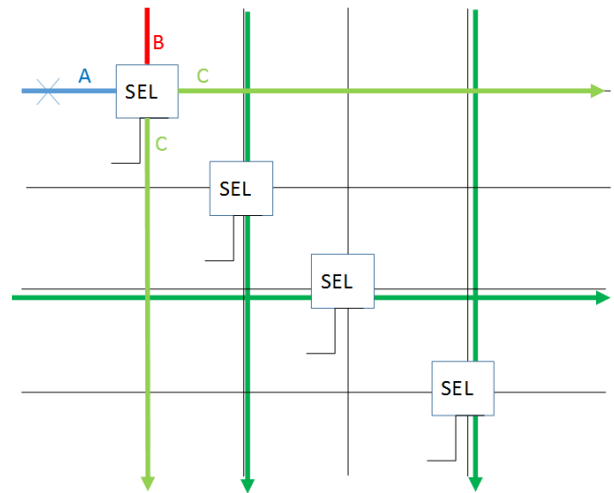


Fig.4: Structure of the SB.

The architecture of FPGAs is shown in Fig. 1. FPGAs have three components. These are a logic block (LB) as shown in Fig. 2, a connection block (CB) (Fig. 3), and a switch block (SB) (Fig. 4). The CB and SB are very different from conventional FPGAs. These switches are not transistors, but rather they are selectors. This is because the direction of the FPGA routing is different from that of conventional FPGAs.

As indicated by the selectors of Fig. 3 and Fig. 4, the selected result of A or B is output to C. In the case of Fig. 3 and Fig. 4, the signal of B is selected and flows to C. The signal A cannot be used. Fig. 4 shows that 4 lines can cross simultaneously. Therefore, it is confirmed that routings on the FPGA are possible [9].

An advantage of the FPGAs is that they can be developed using RTL design methodology. It allows not only ASIC-FPGA codesign, but also arrangement of the FPGA architecture. If a large circuit is designed

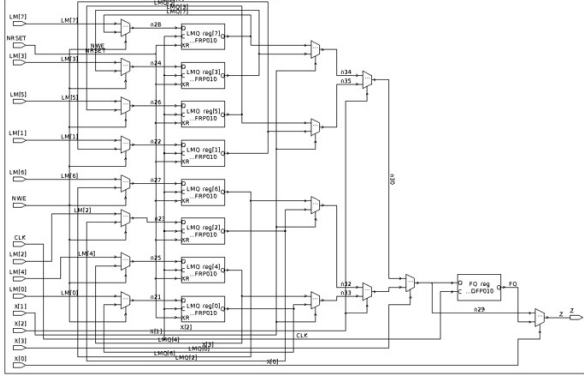


Fig.5: Logic synthesis result of Fig. 2.

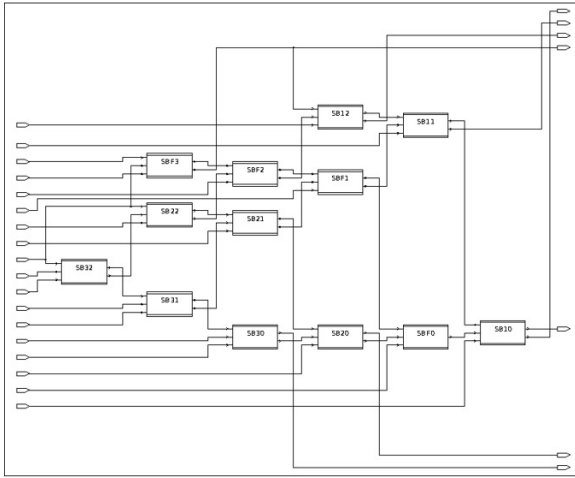


Fig.6: Logic synthesis result of Fig. 3.

on an FPGA, routing may not be possible. In this case, increasing the number of wires solves the problem of routing. It is easy to realize other arithmetic circuits.

For the evaluation carried out in Section 6, the authors ran a logic synthesis of the parts of the FPGAs using the design environments of Table 2. In this paper, the authors chose the standard cell library released by Rohm, Inc.

The CB of this paper was different from that of [7]. The number of selectors was optimized. Logic synthesis results of Fig. 2, Fig.3 and Fig. 4 are shown in Fig. 5, Fig. 6 and Fig. 7, respectively.

4. RCA ON THE FPGAS

The FPGAs were verified that they operated at a practical rate. A 4-bit RCA circuit was developed on the FPGAs in this section. The LUT of the FPGAs was a 3-input and 1-output format and could store data according to the table shown in Fig. 8. Therefore, a full adder (FA) was designed with two LBs.

The 4-bit RCA circuit is shown in Fig. 9. Routing was made based on the explanation in Section 3. The operational frequency was calculated using the

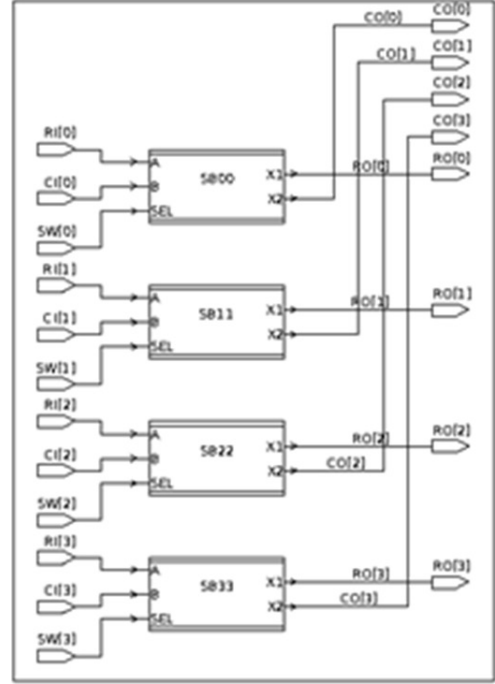


Fig.7: Logic synthesis result of Fig. 4.

CIN	A[0]	B[0]	S[0]
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

(a)

CIN	A[0]	B[0]	CIN[1]
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

(b)

Fig.8: Truth table for the full adder in the LUTs. (a) Sum, and, (b) Carry.

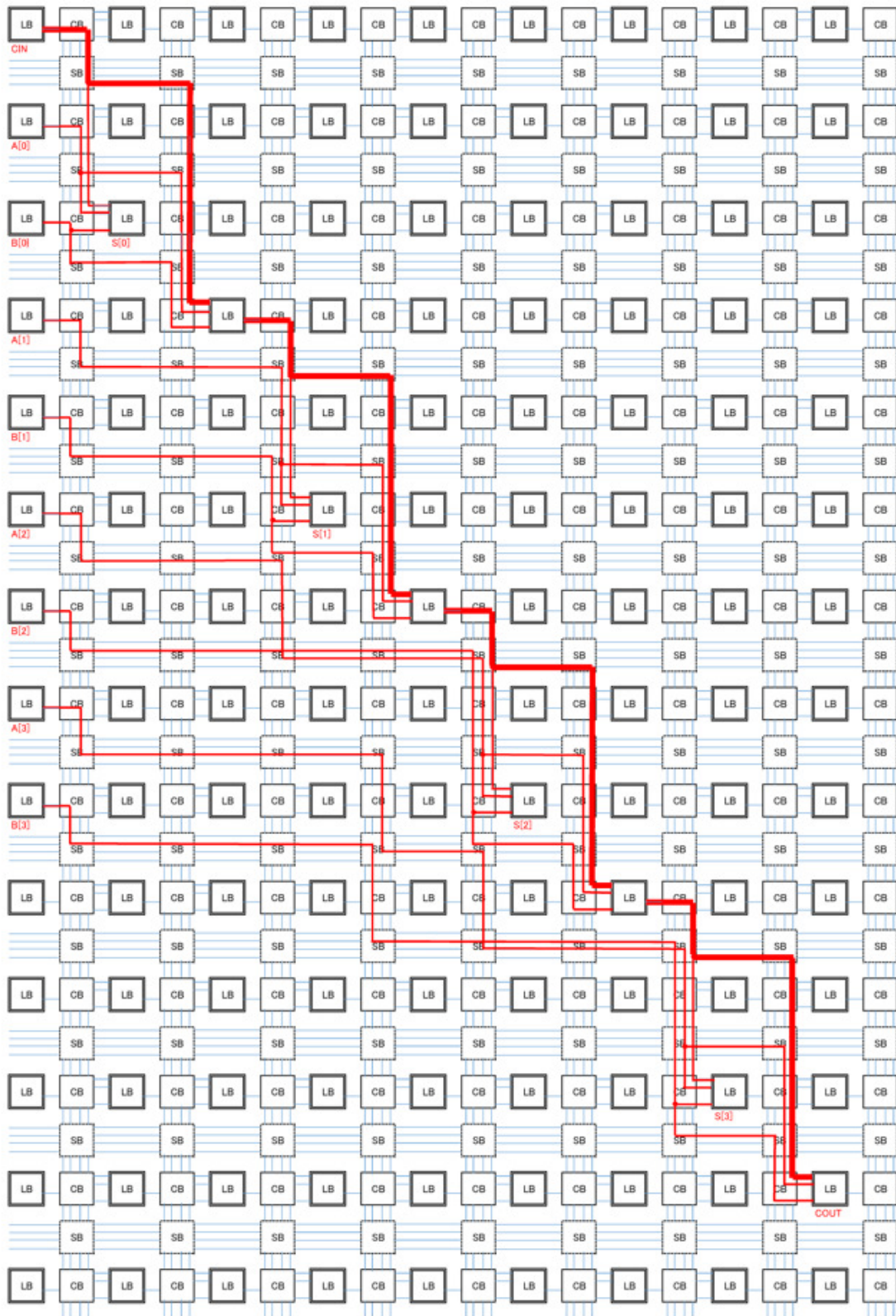


Fig.9: 4-bit RCA on the FPGAs.

results of logical synthesis. For the calculation of the delay times, the authors wrote a software program. According to these results, the routing with the maximum delay time was from CIN to COUT, shown as the heavy line in Fig. 9. The maximum delay time was 24.12 ns.

5. WAVE-PIPELINED OPERATION

A wave-pipeline [17], [18] is a design method that does not use registers for pipeline operations, and for this reason, it is superior in terms of power consumption. In circuits on FPGAs, design techniques for high-speed operations are limited. In such a situation, a wave-pipeline is effective [19].

Fig. 10 shows an overview of pipelines. The conventional pipelines shown in Fig. 10 (a) require registers for pipelined operations. Only one set of signals can operate in the circuit between the pipeline registers. Alternatively, pipeline registers are not used for wave-pipelines. Therefore, it is essential to make a collision-free interval so that signals do not collide. Wave-pipelined operations provide for a condition where two or more signals exist between registers.

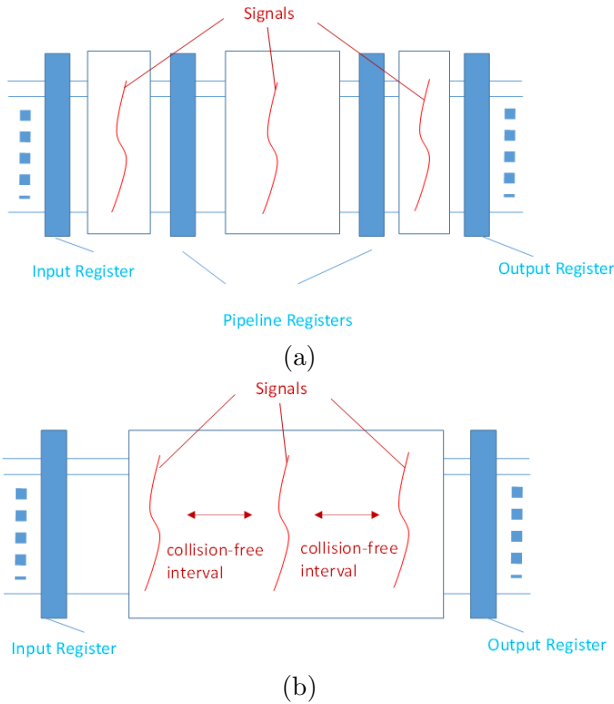


Fig.10: *Overviews of pipelines. (a) Conventional pipeline (b) Wave-pipeline.*

A wave-pipeline is also used in commercial processors. Circuits constituting FPGAs for wave-pipelines have been studied [6-9, 21]. However, this was not achieved in the arithmetic circuit on the FPGA constructed by RTL proposed by us.

A clock cycle time for wave-pipelining, T_{CK} , was calculated from the following equation [20].

$$T_{CK} = (D_{MAX} - D_{MIN}) + T_{OV} \quad (1)$$

Wave pipelined operations are achieved if this expression is satisfied. In a circuit for wave-pipelined operations, D_{MAX} is the maximum delay time and D_{MIN} is the minimum delay time. T_{OV} was set as a margin. The margin means the influence of the conditions of chip fabrication and operating conditions such as temperature and voltage.

The novelty of the wave-pipeline in this paper is that it allows wave-pipelined operations without changing the circuit configuration of Fig. 9. Here, route adjustments of only the outputs shown in the heavy line of Fig. 11 are executed. This led to design simplification of the proposed wave-pipelined circuit. As discussed in Section 4, the maximum delay time was 24.12 ns. Also, the minimum delay route on a route could not be adjusted from B[3] to COUT. Here, the minimum delay time was 10.07 ns. From these conditions, the routes of the outputs of S[0], S[1], S[2] and S[3] were derived from the following equation:

Here, the route adjustments of only the outputs shown in the heavy line of Fig. 11 are executed. This is led to the design simplification of a wave-pipelined circuit. According to the Sec. 4, the maximum delay time is 24.12 ns. Also, the minimum delay route on a route cannot be adjusted is from B[3] to COUT. The minimum delay time is 10.07 ns. From these conditions, the routes of the outputs of S[0], S[1], S[2] and S[3] are derived from the following equation.

$$10.07 \leq D_{OUTPUT} \leq 24.12 \quad (2)$$

D_{OUTPUT} is a delay time for the outputs. All the outputs of Fig. 9 satisfy Equation (2).

The FPGAs enable ASIC-FPGA co-design. Therefore, arithmetic circuits such as an ASIC solve the problems of operation speed on the FPGAs. However, circuits as an ASIC cannot be changed and added after chip fabrication. Wave-pipelined operations on the FPGAs are needed for this reason.

6. EVALUATIONS

The FPGAs designed by the RTL design methodology were evaluated using the operation frequencies of Fig. 9 and Fig. 11. An operating frequency in normal operations of the RCA in Fig. 9 can be obtained from the maximum delay time. Alternatively, an operating frequency in wave-pipelined operations of the RCA in Fig. 11 was derived from Equation (1). T_{OV} , the overhead time in wave-pipelined operations, was set to 2.0 ns.

Actually, circuits on an ASIC fabricated in a 0.18 μm CMOS process operate at 2.0 GHz [22]. Therefore, this value is very reasonable.

The clock cycle time of Fig. 11 in wave-pipelined operations, T_{RCA} was calculated from the following

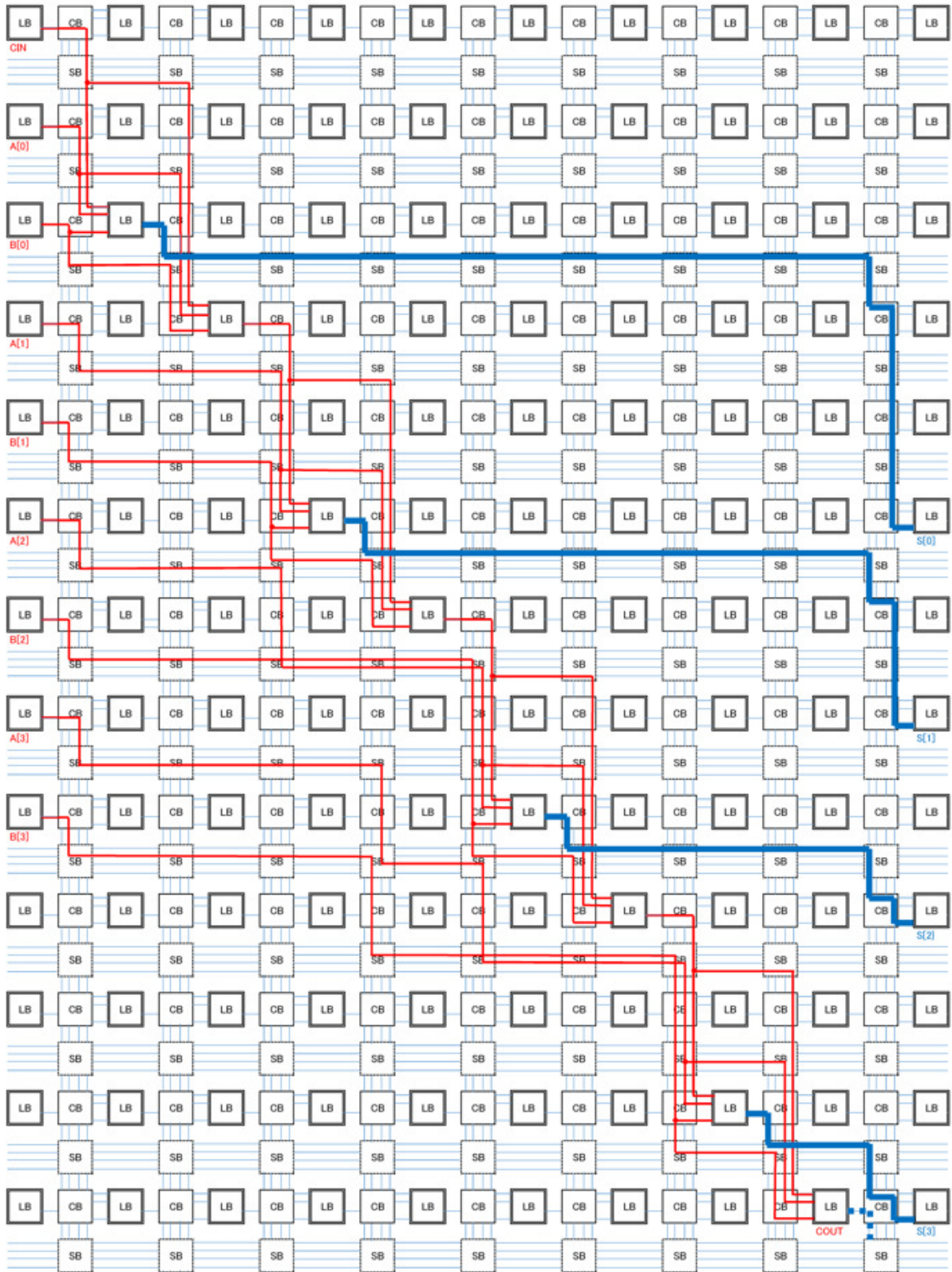


Fig.11: 4-bit RCA for wave-pipelined operations on FPGAs.

equation.

$$T_{RCA} = (24.12 - 10.07) + 2.0 \quad (3)$$

These results are shown in Fig. 12. The operational frequency greatly depends on the process technology. MAX II of Altera's complex programmable logic device (CPLD) has been implemented in a 180 nm CMOS technology [23]. The technology is same as the FPGAs. Operational frequencies of the Internal Oscillator of the CPLD were 13.33-22.22 MHz. That is, the operation frequency of the FPGA was higher than the CPLD.

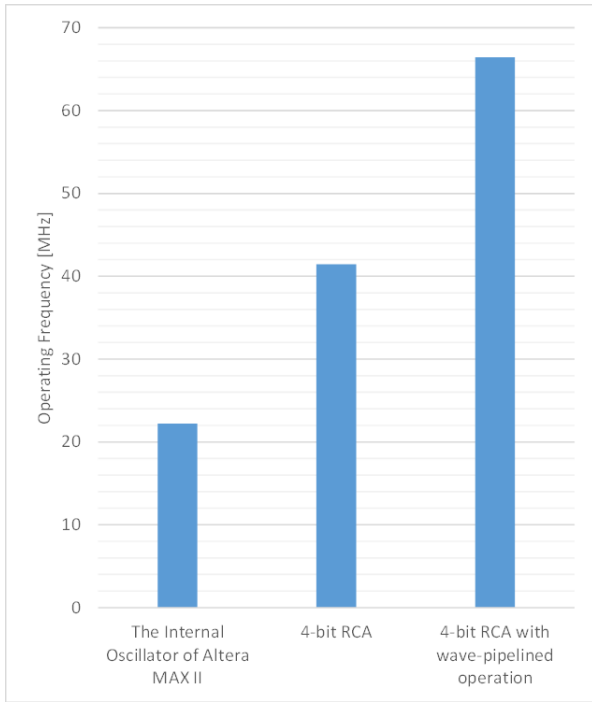


Fig.12: Operating frequencies of the CPLD and 4-bit RCAs on the FPGAs.

When packet processing of a computer network is executed on FPGAs, processing in packet frame units is possible. Here, the operating frequency in the FPGAs was set to 60 MHz. In the case of 1 Gbps, a word width of 17 bits or more enables the processing. Thus, it is clear that the process is practical.

7. CONCLUSIONS

The FPGAs designed by the RTL design methodology have the following advantages:

- Easy integration of FPGA functions in a SoC is possible.
- Significant shortening of the design period of a SoC.
- Allows the selection of process rules.

The authors developed FPGAs to capitalize on these advantages. In this paper, a 4-bit RCA on a FPGA was designed and evaluated to demonstrate the practicality of the approach.

FPGAs were not developed for this purpose since there they have a larger delay than conventional FPGAs. This problem was relaxed by wave-pipelined operations without changing the circuit configuration of RCAs. Wave-pipelined operations are very suitable for patterned circuits like those in RCAs. That is, they are considered applicable in multiple circuits.

Wave-pipelined circuits have the advantage of not increasing power consumption because they do not require pipeline registers. Additionally, the delay time of the entire circuit in wave-pipelined operations is the same as in normal operations.

Therefore, the contributions of this paper are as follows:

- It was shown that the FPGAs can be put to practical use.
- A 4-bit adder can be implemented on FPGAs.
- The problem of routing delay can be solved by improving the throughput by easy wave-pipelined operations.

Future work will involve fabrication of FPGA chips using the 180 nm CMOS standard cell library and evaluations by measurements of the chip.

ACKNOWLEDGEMENT

This work was supported in part by VLSI Design and Education Center (VDEC), the University of Tokyo, in collaboration with Synopsys, Inc., Rohm Corporation, Toppan Printing Corporation and KAKENHI Grant Number 25330149.

References

- [1] Y. R. Qu and V. K. Prasanna, "High-Performance and Dynamically Updatable Packet Classification Engine on FPGA," *IEEE Trans. Parallel Distrib. Sys.*, vol. 27, no. 1, pp. 197-209, 2016.
- [2] T. Sato, S. Imaruoka and M. Fukasse, "Verifying Firewall Circuits by Wave-Pipelined Operations," in *Proc. IEEE TENCON 2009*, pp. WED3.P.14.1-WED3.P.14.6, 2009.
- [3] S. Redif and S. Kasap, "Novel Reconfigurable Hardware Architecture for Polynomial Matrix Multiplications," *IEEE Trans. VLSI*, vol. 23, no. 3, pp. 254-265, 2015.
- [4] H. Marzouqi, M. Al-Qutayri, K. Salah, D. Schinianakis, and T. Stouraitis, "A High-Speed FPGA Implementation of an RSD-Based ECC Processor," *IEEE Trans. VLSI*, vol. 24, no. 1, pp. 151-164, 2016.
- [5] K. Huang, R. Zhao, W. He and Y. Lian, "High-Density and High-Reliability Nonvolatile Field-Programmable Gate Array With Stacked 1D2R RRAM Array," *IEEE Trans. VLSI*, vol. 24, no. 1, pp. 139-150, 2016.
- [6] T. Sato, S. Chivapreecha and P. Moungnou, "A Crossbar Switch Circuit Design for Reconfig-

- urable Wave-Pipelined Circuits,” in *Proc. WM-SCI 2014*, vol. II, pp. 200-205, 2014.
- [7] T. Sato, S. Chivapreecha and P. Moungnou, “Wiring Control by RTL Design for Reconfigurable Wave-Pipelined Circuits,” in *Proc. AP-SIPA ASC 2014*, pp. WP1-3-1-WP1-3-6, 2014.
- [8] T. Sato, S. Chivapreecha and P. Moungnou, “Fine-Tuning of Wave-Pipelines on FPGAs Developed by the RTL Design,” in *Proc. ECTI-CON 2015*, pp. 1230.1-1230.6, 2015.
- [9] T. Sato, S. Chivapreecha and P. Moungnou, “The Potential of Routes Configured with the Switch Matrix by RTL,” *Applied Mechanics and Materials J.*, vol. 781, pp. 189-192, 2015.
- [10] W. Hong, K.-H. Baek and A. Goudelev, “Multi-layer Antenna Package for IEEE 802.11ad Employing Ultralow-Cost FR4,” *IEEE Trans. Antennas and Propagation*, vol. 60, no. 12, pp. 5932-5938, 2012.
- [11] H. Sawada; S. Takahashi and S. Kato, “Disconnection Probability Improvement by Using Artificial Multi Reflectors for Millimeter-Wave Indoor Wireless Communications,” *IEEE Trans. Antennas and Propagation*, vol. 61, no. 4, pp. 1868-1875, 2013.
- [12] A. H. Fazlollahi and J. Chen, “Copper Makes 5G Wireless Access to Indoor Possible,” in *Proc. 2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1-5, 2015.
- [13] Y. R. Qu and V. K. Prasanna, “High-Performance and Dynamically Updatable Packet Classification Engine on FPGA,” *IEEE Trans. Parallel and Distributed Systems*, vol. 27, no. 1, pp. 197-209, 2016.
- [14] A. Kennedy and X. Wang, “Ultra-High Throughput Low-Power Packet Classification,” *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 22, no. 2, pp. 286-299, 2016.
- [15] G. Brebner and W. Jiang, “High-Speed Packet Processing using Reconfigurable Computing,” *IEEE Micro*, vol. 34, no. 1, pp. 8-18, 2014.
- [16] T. Sato, P. Moungnou, S. Chivapreecha and K. Higuchi, “Performance Estimates of an Embedded CPU for High-Speed Packet Processing,” in *Proc. ECTI-CON 2014*, pp.1298.1-1298-5, 2014.
- [17] L. Cotton, “Maximum Rate Pipelining Systems,” in *Proc. AFIPS Spring Joint Computer Conference*, pp. 581-586, 1969.
- [18] F. Klass and M. J. Flynn, “Comparative Studies of Pipelined Circuits,” *Stanford University Technical Report*, no. CSL-TR-93-579, 1993.
- [19] I. B. Eduardo, L. Sergio and M. M. Juan, “Some Experiments About Wave Pipelining on FPGA’s,” *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 6, no. 2, pp. 232-237, 1998.
- [20] W. P. Burleson, M. Ciesielski, F. Klass, and W. Liu, “Wave-Pipelining: A Tutorial and Research Survey,” *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 6, no. 3, pp. 464-474, 1998.
- [21] T. Sato, P. Moungnou, S. Chivapreecha and K. Higuchi, “A Connection Block Implemented in the RTL Design for Delay Time Equalization of Wave-Pipelining,” *J. Systemics, Cybernetics and Informatics*, vol. 14, no. 1, pp. 49-54, 2016.
- [22] C. L. Jin, X. P. Yu and W.-Q. Sui, “1-2 GHz 2 mW injection-locked ring oscillator based phase shifter in 0.18 m CMOS technology,” *Electronics Letters*, vol. 52, no. 22, pp. 1858 - 1860, 2016.
- [23] Altera Inc., “Using the Internal Oscillator in Altera MAX Series,” https://www.altera.com/en_US/pdfs/literature/an/an496.pdf, 2014.



Tomoaki Sato received the B.S. and M.S. degrees from Hirosaki University, Japan, in 1996 and 1998 respectively, and the Ph.D. degree from Tohoku University, Japan, in 2001. From 2001 to 2005, he was an Assistant Professor of Sapporo Gakuin University, Japan. Since 2005, he has been an Associate Professor of Hirosaki University. Between 2012 and 2015, he was a Visiting Associate Professor of the Open University of Japan. From April 2017, he will become a Professor of Hokusei Gakuen University, Sapporo, Japan. His research interests include VLSI Design, Computer Hardware, Computer Architecture and Computer and Network Security. He is also a member of ECTI, IEEE, IEICE, IPSJ, and SSI.



Sorawat Chivapreecha received his B.Eng. degree in telecommunication engineering from the Suranaree University of Technology (SUT), Nakhon Ratchasima, Thailand, in 1998, M.Eng. degree and D.Eng. degree in electrical engineering from the King Mongkut's Institute of Technology Ladkrabang (KMUTL), Bangkok, Thailand, in 2001 and 2008, respectively. Since 2002, he has been a faculty member of the Department of Telecommunication Engineering, Faculty of Engineering, KMUTL, where he is currently an Assistant Professor. In 2011, he was awarded the distinguished lecturer from Faculty of Engineering, KMUTL. His research interests include digital filter design and implementation, VLSI for digital signal processing, information science and satellite engineering. He is also a member of IEEE and IEICE.



Phichet Moungnou received the B.Ind.Tech. degree in telecommunication technology, M.Eng and D.Eng degree in Electrical Engineering from King Mongkut's Institute of Technology Ladkrabang (KMUTL), Bangkok, THAILAND, in 1992, 1997 and 2001, respectively. Since 1998, he has been a member of the Department of Telecommunication at Faculty of Engineering,

KMITL, where he is currently an assistant professor of telecommunication. His research interests wireless and mobile communication and Broadband Communication.



Engineering and Digital Signal Processing.

Kohji Higuchi received his Ph.D. degree from Hokkaido University, Sapporo, Japan in 1981. In 1980 he joined the University of Electro-Communications, Tokyo, Japan, as a Research Associate, where he became an Assistant Professor in 1982 and currently an Associate Professor in the Dept. of Mechanical Engineering and Intelligent Systems, Electronic Control System Course. His interests include Power Electronics, Control