

A Connection Block Implemented in the RTL Design for Delay Time Equalization of Wave-Pipelining

Tomoaki SATO

Computing and Networking Center, Hirosaki University
Hirosaki 036-8561 Japan

and

Sorawat CHIVAPREECHA

Department of Telecommunication Engineering,
Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang
Bangkok 10520 Thailand

and

Phichet MOUNGNOUL

Department of Telecommunication Engineering,
Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang
Bangkok 10520 Thailand

ABSTRACT

Field-programmable gate arrays (FPGAs) which have many advantages are used in various devices. Use of the FPGAs is not only prototyping and verification of circuits but also an important part of the commercial products. A CPU of hardware is required in the FPGAs. But it has a problem with the architecture of the CPU is limited. The method of solving these problems is developing a system on a chip (SoC) which is equipped with FPGAs and a customized CPU. From the view point of ease of design and shortening a design period, development techniques on a register-transfer level (RTL) using a standard cell library are essential. On the other hand, applying this method without using a design technique has a problem in terms of throughput. In this paper, a connection block for routing using wave-pipeline technique is proposed to solve the throughput problems. This block is evaluated, and it is shown that it is useful for wave pipeline operation.

Keywords: Connection Block, Field-Programmable Gate Arrays, Delay Time Equalization, Wave-Pipelines, IPS, Reconfigurable Circuits.

1. INTRODUCTION

Field-programmable gate arrays (FPGAs) that easily achieves specialized circuits by using hardware description languages (HDLs) are used for various devices. Usage of the FPGAs is not only prototyping [1] and verification of circuits [2] but also an important part of the commercial products [3]. Using FPGAs has advantages as follows:

- Easy change of circuits
- Can be verified in circuits
- Significant reduction of development time
- Cost reduction of a small number of products

Furthermore, a central processing unit (CPU) core is built into some FPGAs. On the CPU is capable of running an operating

system (OS) such as Linux. The OS running enables the managements of task, memory, file control and peripherals. After that, because an application software can share the interface by the provision of an application programming interface (API), the software development becomes easy.

Architecture and micro-architecture of the CPU built into the FPGA chip is determined by the FPGA manufacturer, and the FPGA users cannot customize it. As customizable CPU by the FPGA users, a soft-core CPU is used. This CPU works on FPGAs, which is disadvantageous in terms of operating frequency and power consumption than the CPU built into the FPGA chip.

The method of solving these problems is developing a system on a chip (SoC) which is equipped with FPGAs and a customized CPU. In particular, from the view point of ease of design and shortening a design period, development techniques on a register-transfer level (RTL) using a standard cell library is essential [4-7]. Conventional FPGAs is developed at the RTL. Therefore, FPGAs developed by the RTL is more disadvantageous than conventional FPGAs in the point of a throughput.

Wave-pipeline technique [8-10] is used as a design method to improve the throughput of circuits on FPGAs [11]. The technique leads pipelined operations without using registers. Therefore, the advantage that power consumption doesn't increase is possessed. This technique achieves high-speed pipeline operations by reducing the difference between the maximum delay time and the minimum delay time. However, the use of the connection blocks in FPGAs has the problem that the delay time difference extends.

In this paper, a connection block for routing in wave-pipeline operations is proposed for the purpose of solving this problem. The use of this block achieves that to reduce the significant impact delay time difference in high throughput in wave-pipelined operations. Furthermore, the connection block has to

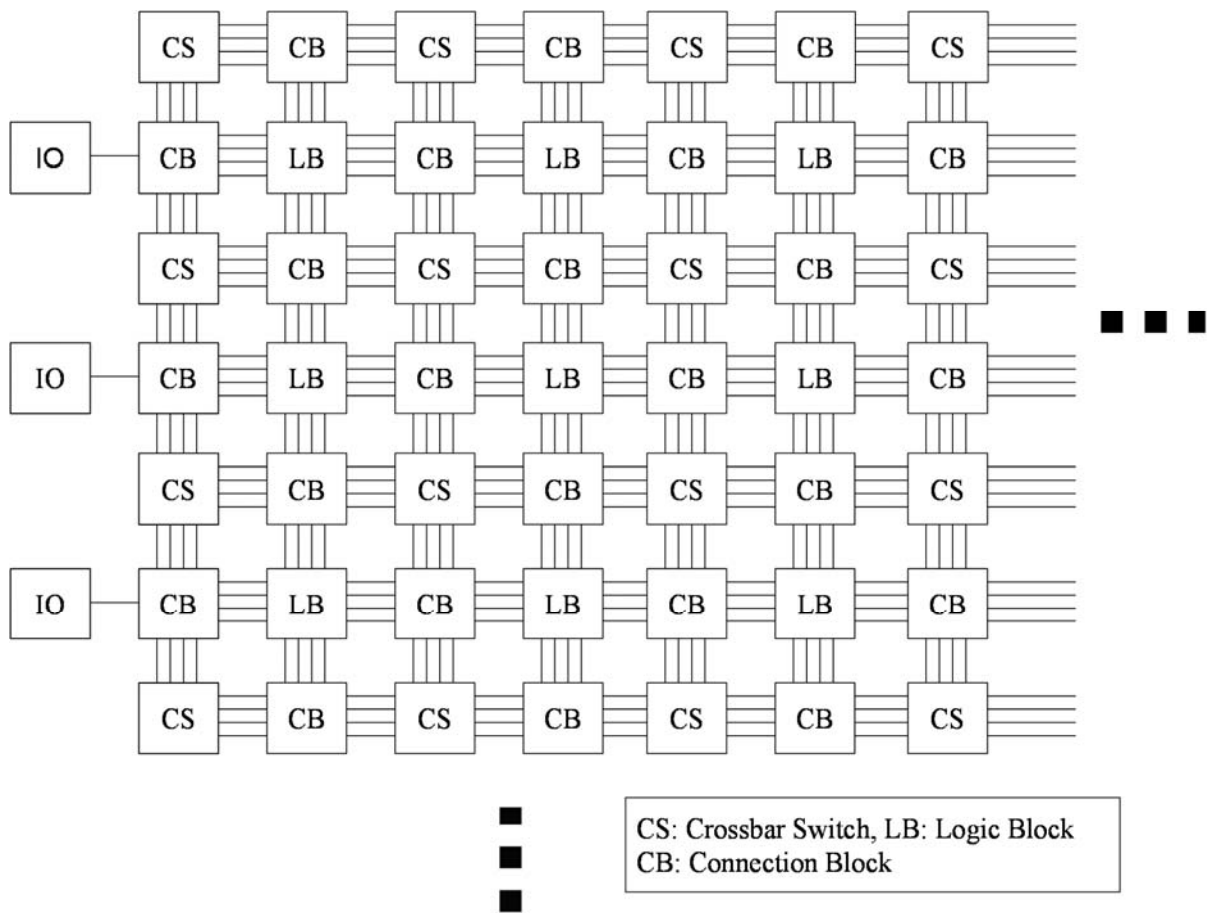


Figure 1. The structure of conventional FPGAs

be developed in the RTL. Therefore, developed algorithm is required to be applied to the RTL design. In this paper, the development procedure of this block is shown.

This paper is organized as follows. Section 2 describes the outline of FPGAs implemented in the RTL design and the Connection Blocks for the FPGAs. Section 3 proposes the delay time equalization of the connection block for high-throughput operations in the FPGAs. The connection block is evaluated in Section 4. Finally, Section 5 presents our conclusions.

2. CONNECTION BLOCKS ON FPGAs IMPLEMENTED IN THE RTL DESIGN

A connection block is used to connect routing wires and logic blocks. Transistors are used in conventional FPGAs as the switch of the connection block. In the FPGAs that we design using standard cells, the switch of the connection block is a selector.

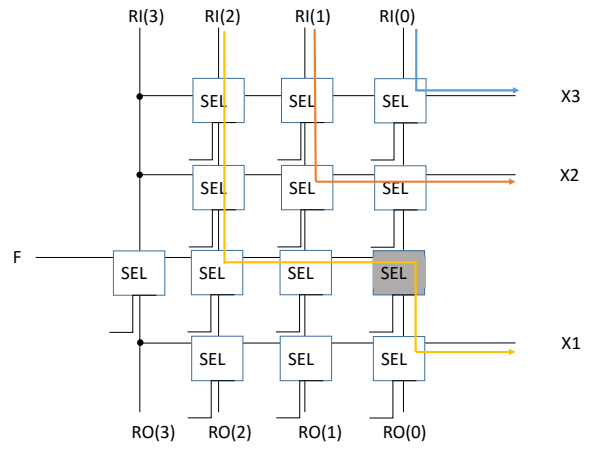


Figure 2. A connection block developed in the RTL design.

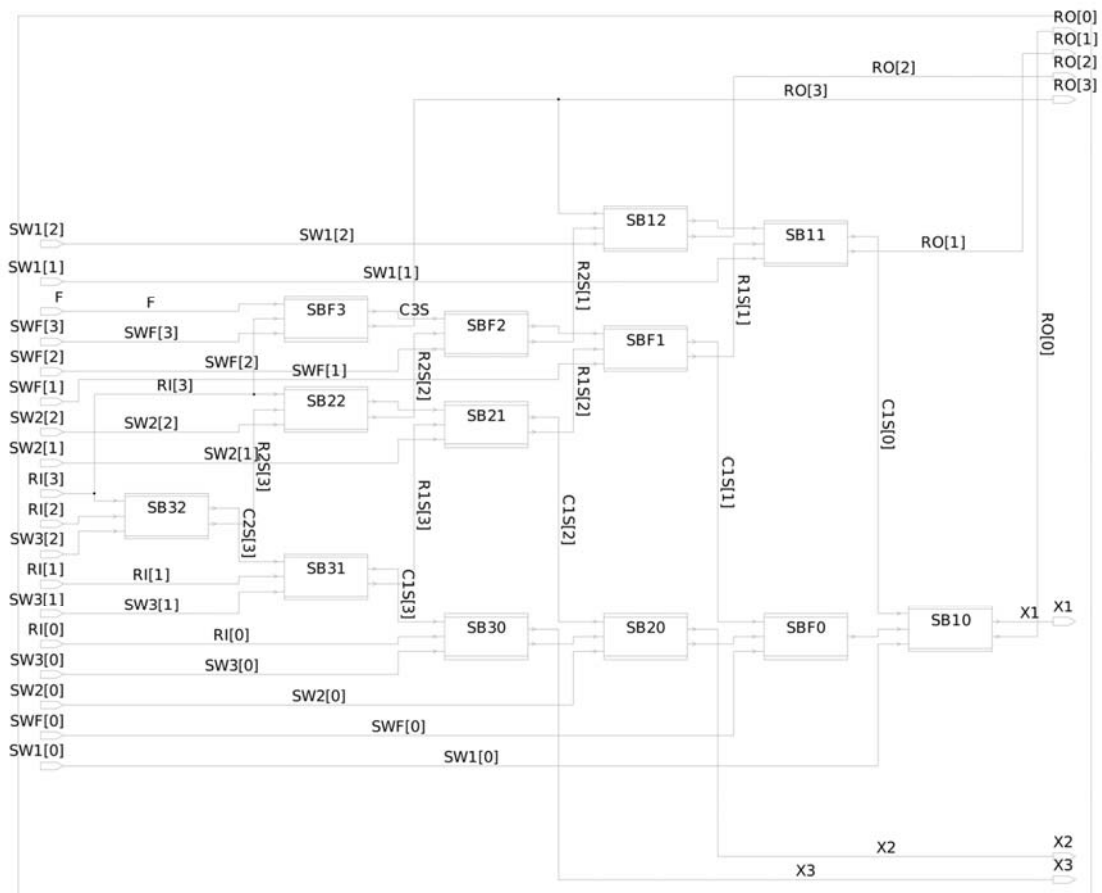


Figure 3. A connection block developed by the RTL design.

This block is developed by using the development environment and the standard cell library shown in Table 1. Figure 2 shows the connection block for the RTL design. Figure 3 is the connection block for Figure 2 developed by the RTL design. This is the results of the logic synthesis using the logic synthesis tool and the standard cell library in Table 1. This block is not applied equalization of delay time. Therefore, there is a problem that the delay time varies greatly depending on the routing path.

Six selectors are used in the route from RI(2) of the input port to X1 of the output port shown in Figure 1. Reason for such a route is to secure the three signal lines. Thus, if the route uses the connection block developed in RTL, the route should use a minimum of three selectors of the vertical column. As a result, each of the delay time of the signal lines shown in Figure 1 is significantly different. The difference is shown in Table 2. The delay time difference further expands by multiple use this connection block.

Table 1. Design environments

OS	Cent OS 5.9 x86
CPU	Intel Core 2 Duo E6600
Memory	2 GBytes
Logic synthesis	Synopsys Design Compiler H-2013.03-SP2
Technology	Rohm 180 nm C-MOS
Standard cell library	Tamaru/Onodera Lab. of Kyoto Univ. [12]

Table 2. Delay time difference of Figure 2

Routes	Delay times
RI(0) -> X3	0.27
RI(1) -> X2	0.87
RI(2) -> X1	1.76

3. DELAY TIME EQUALIZATION

The authors adjust delay times in buffer insertions in order to reduce the delay time difference. Previous study of the connection block for wave-pipelined operations is being conducted by us in [7]. This study is needed to timing adjustment in wave-pipelined circuit designs. Meanwhile, this paper is the study to reduce the delay time difference of the routing.

In this delay adjustment, standard cells with a buffer or inverter are used. The specific development procedure is as follows:

- The delay times of each path are examined using a logic synthesis tool
- The delay time difference of each path is calculated
- Delay time adjustment circuits in accordance with the insertion of delay time elements are made based on the delay time difference
- Delay time adjustment circuits are inserted into the connection block

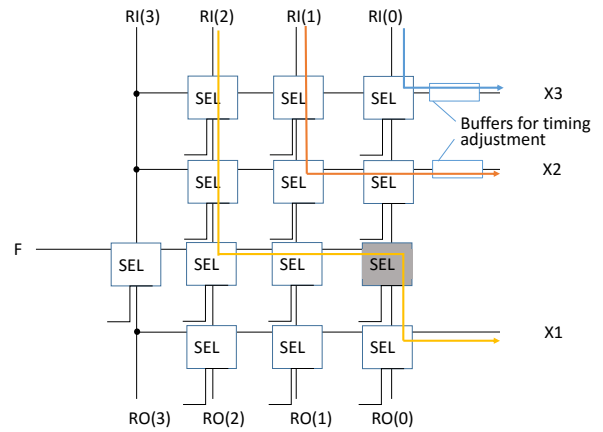


Figure 4. Delay elements inserted into the connection block.

Delay elements are inserted into the respective X1 and X2 of output ports shown in Figure 4. Figure 5 is the results of the logic synthesis using the design environments of Table 1. Each delay

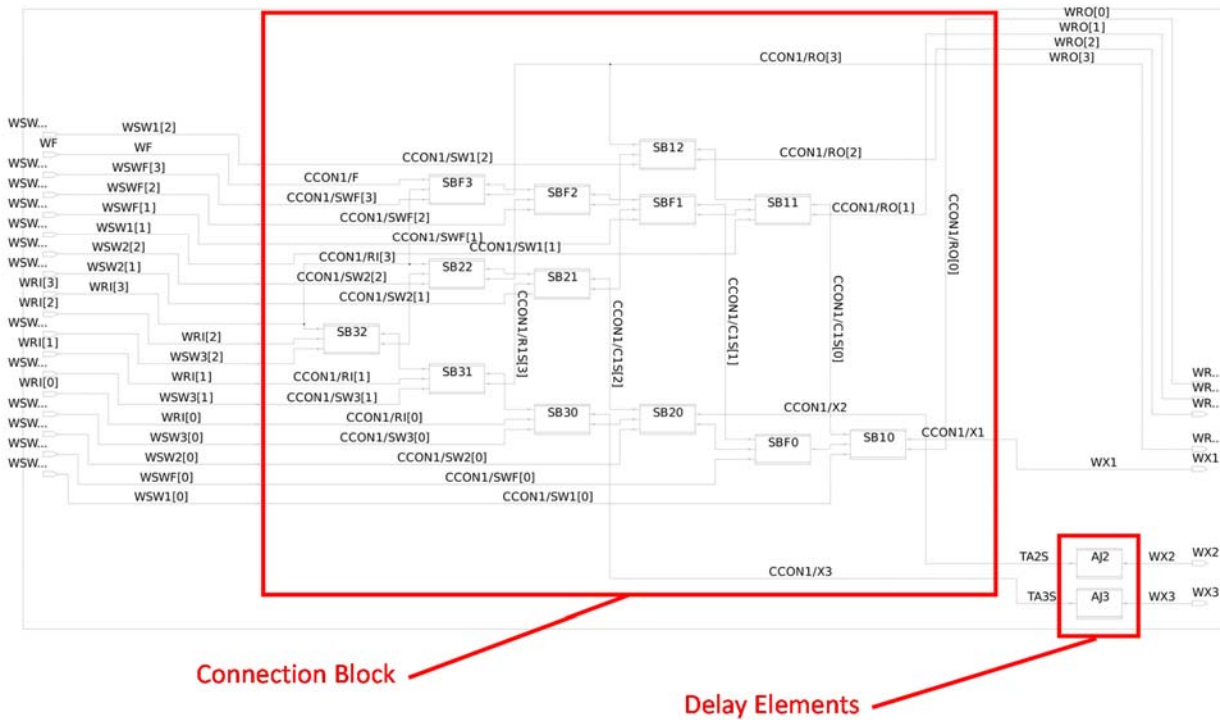


Figure 5. A connection block for delay time equalization.

time is calculated based on the output delay time of the X1. Insert buffers created based on this calculation is shown in Figure 6. The delay times of the delay elements are given in Table 3.



(a)



(b)

Figure 6. Delay elements (a) Inserted to X3 (b) Inserted to X2.

All of these are possible to run on Design Compiler of Synopsys. In other words, the design of the connection block in based on the equalized delay time can be made automatically by creating a script on the CAD. This is essential in the development of FPGAs by the RTL.

Table 3. Delay times of Figure 6

Delay elements	Delay times [ns.]
Inserted to X3	1.10 - 1.26
Inserted to X2	0.50 - 0.66

4. EVALUATIONS

The connection block for delay time equalization in wave-pipelining developed in this study is shown in figure 5. Using the development environment of table 1, the delay times of each path are calculated. The delay times of each route are shown in Table 4.

These results reveal that the different delay time is reduced to approximately 1/5. Since these results are a delay time difference per block, actual circuits are further expanded the delay time difference.

Table 4. Delay time difference of Figure 4

Routes	Delay times	Delay type
RI(0) -> X3	1.42	Minimum
RI(1) -> X2	1.42	Minimum
RI(2) -> X1	1.70	Maximum

The clock cycle time of wave-pipelining, T_{CK} , is given by the following equation.

$$T_{CK} = (D_{MAX} - D_{MIN}) + T_{OV}. \quad (2)$$

Here,

D_{MAX} : The maximum delay path,

D_{MIN} : The minimum delay path,

T_{OV} : Overhead time.

The clock cycle time of Figure 2, T_{CK1} , is given by the following equation.

$$T_{CK1} = (1.76 - 0.27) + T_{OV}. \quad (2)$$

Also, the clock cycle time of Figure 4, T_{CK2} , is provide by the following equation.

$$T_{CK2} = (1.70 - 1.42) + T_{OV}. \quad (3)$$

It can be expressed as

$$T_{CK2} > T_{CK1}. \quad (4)$$

Thus, the connection block which has been proposed in this study is excellent in terms of throughput.

5. CONCLUSIONS

It is imperative for circuits on FPGAs that are developed by RTL to increase the throughput of the routing. In this paper, delay equalizations of the connection block for the FPGAs were performed in order to increase the throughput of wave-pipelined operations. The delay time difference of this connection block was confirmed by using 0.18um C-MOS technology. The delay time difference is reduced to 1/5 of the previous state of delay equalizations. That is, the delay time difference of the routing for connecting between the logic blocks is reduced, it becomes possible to increase the throughput of circuits on the FPGA.

In future research, the proposed algorithm is described as a script for the synthesis tool. As a result, the connection blocks for delay equalization are implemented by an automated design.

ACKNOWLEDGMENT

This work has been supported in part by VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys, Inc. and KAKENHI Grant Numbers 25330149. The standard cell library used on this research was developed by Tamaru/Onodera Lab. of Kyoto Univ. and released by Prof. Kobayashi of Kyoto Inst. of Tech.

REFERENCES

- [1] M. Gschwind, V. Salapura and D. Maurer "FPGA prototyping of a RISC processor core for embedded applications," **IEEE Trans. Very Large Scale**

- Integration (VLSI) Systems**, vol. 9, no. 2, pp.241-250, 2001.
- [2] C. Xia-Tao, W.-K. Huang, N. Park, F. J. Meyer, and F. Lombardi, "Design verification of FPGA implementations," **IEEE Design & Test of Computers**, vol. 16, no. 2, pp. 66 - 73, 1999.
 - [3] R. Duncan, P. Jungck, A. Norton, K. Ross and G. Triplett, "FPGA-Driven Table System to Accelerate Network Flows," **Proc. of 2013 16th International Conference on Network-Based Information Systems (NBIS)**, pp. 1 - 8 , 2013.
 - [4] T. Sato, S. Chivapreecha and P. Moungnoul, "Wiring Control by RTL Design for Reconfigurable Wave-Pipelined Circuits," **Proc. of APSIPA ASC 2014**, pp. WP1-3-1-WP1-3-6, 2014.
 - [5] T. Sato, S. Chivapreecha and P. Moungnoul, "A Crossbar Switch Circuit Design for Reconfigurable Wave-Pipelined Circuits," **Proc. of WMSCI2014**, vol. II, pp. 200-2052014.
 - [6] T. Sato, S. Chivapreecha and P. Moungnoul, "A Logic Block for Wave-Pipelining," **Proc. of IMETI 2013**, Jul. 2013, pp. 130-134.
 - [7] T. Sato, S. Chivapreecha and P. Moungnoul, "Fine-Tuning of Wave-Pipelines on FPGAs Developed by the RTL Design," **Proc. of ECTI-CON 2015** (To be published).
 - [8] L. Cotton, "Maximum Rate Pipelining Systems," in **Proc. AFIPS Spring Joint Computer Conference**, pp. 581-586, 1969.
 - [9] F. Klass and M. J. Flynn, "Comparative Studies of Pipelined Circuits," Stanford University Technical Report, no. CSL-TR-93-579, 1993.
 - [10] W. P. Burlison, 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.
 - [11] 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.
 - [12] H. Onodera, A. Hirata, A. Kitamura, K. Kobayashi, K. Tamaru, "P2Lib:Process Portable Library and Its Generation System," **J. Information Processing**, vol.40, no. 4, pp. 1660-1669, 1999, in Japanese.