

FPGA Area Reduction with Uncommon Usage of Reset Signals

Hiroshi Nakatsuka[†]Kenji Kise[‡]Department of Computer Science, Tokyo Institute of Technology[†]
Graduate School of Information Science and Engineering, Tokyo Institute of Technology[‡]

1 Introduction

An FPGA (Field Programmable Gate Array) is an LSI which can be reconfigured to desired logic circuits after manufacturing. Engineers can realize developed logic circuits on FPGAs. In recent years, FPGAs have been commonly used for both prototyping SoC (System-on-Chip), ASIC and implementing low volume production devices. A circuit implemented on an FPGA is composed of logic blocks and switch matrix for connecting the logic blocks. In FPGAs, logic blocks are expensive hardware resources, and the scale of the circuit to be implemented depends on the number of logical blocks. Therefore, focusing on the internal structure of an FPGA to reduce the circuit area, thereby reducing the consumption of logic blocks is an important issue. In this paper, we propose a method for reducing the circuit area by leveraging reset signals of logic blocks' flip-flops and BlockRAMs' output registers. In order to evaluate the usefulness of the proposed method, we apply the proposed method to implement a soft processor, and compare the hardware resource usage with the original implementation.

2 Background

In FPGAs, a logic block generally consists of a few logic cells. Each logic cell typically consists of look-up tables (LUTs), flip-flops (FFs), carry chain, and multiplexers (MUXs). This paper focuses on the Xilinx Spartan-3E FPGA device. In the FPGA, a logic cell is called a slice. Reducing the number of slices is equivalent to reducing the area of logic blocks. Figure 1 depicts architecture of a slice in the FPGA device. As illustrated in Figure 1, each slice of the device consists of two 4-input LUTs, two FFs, carry chain (MULT_AND, XORCY, MUXCY), 2-input MUXs (MUXF5), etc. [1]. In addition to clock signal (CK), clock enable signal (CE) and input data signal (Q), each FF has a set-reset signal (SR) which is an input signal. Based on the design written in Hardware Description Language (HDL) such as Verilog HDL and VHDL, logic synthesis tools infer whether or not to use these FPGA primitives.

In order to implement relatively large storages such as caches and register files, in addition to the FFs in the logic cells, FPGA has multiple synchronous write and synchronous read memory blocks called BlockRAMs (BRAMs). Each BRAM of the Spartan-3E device is a dual-port memory consisting of 18K bits SRAM, including 2K bits parity. In each port of the BRAM, each output register has a clock enable signal and a reset signal.

3 FPGA Area Reduction Method using Reset Signals

Our method is targeting the Xilinx Spartan-3E FPGA device. Since each LUT of the Spartan-3E device is a 4-input LUT, a 3-input or 4-input MUX can be implemented by connecting two LUTs in a series as

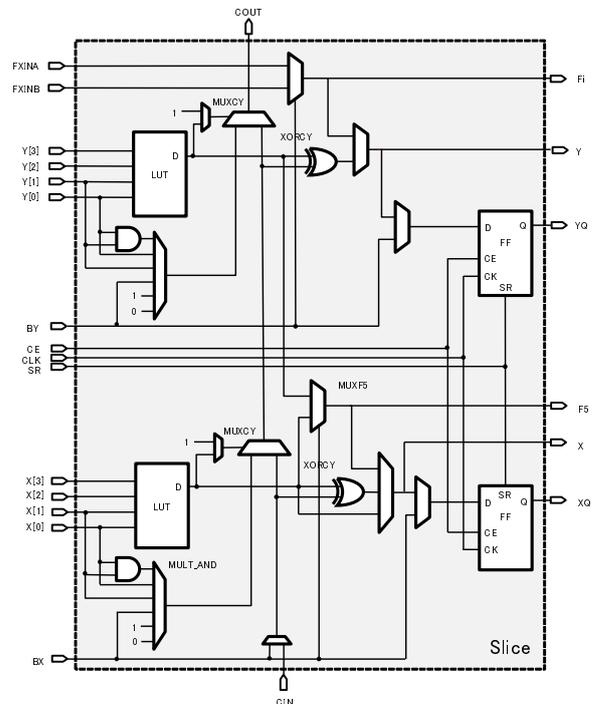


Figure 1: Slice of Spartan-3E FPGA

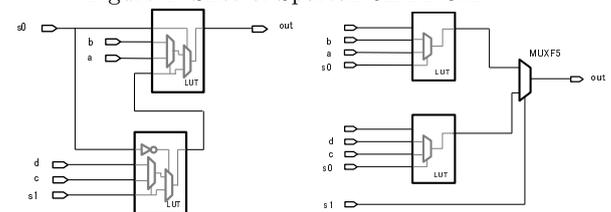


Figure 2: 4-input MUX implementation (1)

Figure 3: 4-input MUX implementation (2)

shown in Figure 2, or connecting two LUTs in parallel using MUXF5 primitive as shown in Figure 3. We consider the case in which all inputs of the MUX are outputs from the slice's FFs. In this case, we reset all FFs which does not output the required signal, and perform OR operation on all FFs' output signals in a single LUT as shown in Figure 4. By this way, it is possible to achieve a circuit which is equivalent to a 3-input or 4-input MUX using only one LUT. This means that a 3-input or 4-input MUX logic that needs two LUTs to be implemented can be configured with a single LUT by using reset signals. As a result, we can expect the reduction of slices to be used in the implementation of MUXs. Even if there is an output from a BRAM to the input of a MUX, it is possible to use a similar approach by using a reset signal of the output register of the BRAM.

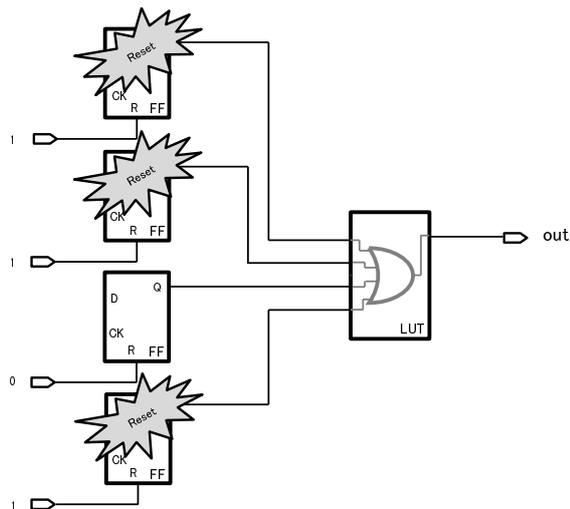


Figure 4: 4-input MUX implementation using reset signals

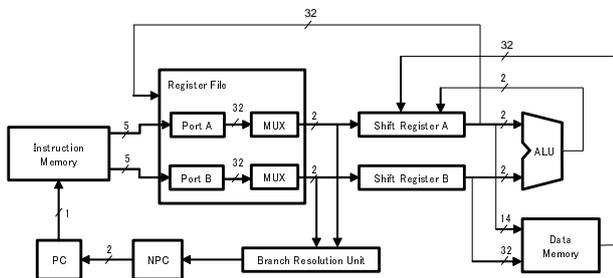


Figure 5: Data path of the Ultrasmall

4 Evaluation and Discussion

Since we have not already done the evaluation of the proposed method alone, we applied the proposed method to Ultrasmall Soft Processor (Ultrasmall) [2]. Ultrasmall is a soft processor that executes a subset of the MIPS instruction. In order to reduce the use area of logical blocks, Ultrasmall adopts 2-bit serial architecture. Figure 5 shows the data path of Ultrasmall. In this paper, we focus on the path from the Register File to the Shift Register A, B of Ultrasmall and apply the method to this path. In Ultrasmall, Data Memory and Register File are implemented in BRAMs, and the Shift Registers are composed of 32-bit FFs. By outputting directly the 32-bit outputs of the Register File to the Shift Registers, the MUXs that select 2-bit from 32-bit do not need. Thus, the number of used slices and LUTs are reduced. In such case, input of each Shift Register, a 3-input MUX which select one signal from the output of the Register File, the output of the Data Memory and the internal transition of the Shift Register is required. However, by applying the proposed method, each of these 3-input MUXs can be realized by using a single LUT. From the connection relationship of FFs and LUTs in a slice as illustrated in Figure 1, we can conclude that the number of used slices is not increased by the addition of these MUXs. Figure 6 shows the data path of the Ultrasmall applied the proposed method.

In order to evaluate the usefulness of the proposed method, we compared the hardware resource usage of the Ultrasmall and Ultrasmall+ (Ultrasmall with the proposed method). The target device is Xilinx Spartan-3E XC3S500E-5VQ. We described Ultrasmall+ and Ultrasmall circuit by Verilog HDL. The

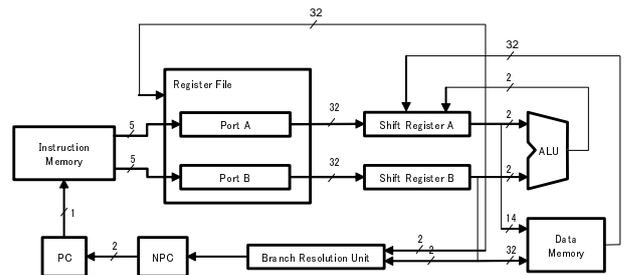


Figure 6: Data path of the Ultrasmall applied the proposed method

	Slices	Slice FF	LUTs	BRAM
Ultrasmall	163	141	244	14
Ultrasmall+	149	142	244	14

Table 1: Comparison between Ultrasmall and after applying the method

Xilinx ISE design suit 14.7 was used for synthesizing of the circuits. We set the synthesis option as following: Optimization Goal to Area, and Optimization Effort to High. Table 1 shows the hardware resource usage of Ultrasmall, and Ultrasmall+. As shown in Table 1, the number of used slices of Ultrasmall+ is about 9% smaller than Ultrasmall. On the other hand, the number of used LUTs of Ultrasmall and Ultrasmall+ is not changed. By applying the proposed method, it is possible to reduce the MUXs in the Register File. However, control logics for generating the reset signals are required. Therefore, the number of used LUTs is not changed. Further, the number of used FFs of Ultrasmall+ increases one compared to Ultrasmall. This is also considered due to the additional control logics for generating the reset signals.

5 Conclusion

In this paper, we presented a method for reducing the circuit area and conserving logic blocks of an FPGA by using reset signals. We applied the proposed method to a soft processor, and compared the hardware resource usage to evaluate the significance of the proposed method. By using the proposed method, it was possible to reduce 9% of used slices, but the number of used LUTs of them did not change. It is believed that this is due to the increase of the control logics for generating the reset signals. As future work we plan to investigate methods for reducing the hardware resources usage of the non-MUX logics.

References

- [1] Xilinx DS312 Spartan-3E FPGA Family Data Sheet
- [2] Tanaka, Y. and Sato, S. and Kise, K. The Ultrasmall Soft Processor HEART, 2013