472 MHz throughput asynchronous FIFO design on a Virtex-5 FPGA device

Jeong-Gun Lee\textsuperscript{1a)}, Deok-Young Lee\textsuperscript{1}, Myeong-Hoon Oh\textsuperscript{2}, and Young-Woong Ko\textsuperscript{1b)}

\textsuperscript{1} Dept. of Computer Engineering, Hallym University
39 Hallym-daehakgil, Chuncheon, Gangwondo 200–702, Korea
\textsuperscript{2} Electronics and Telecommunications Research Institute,
138 Gajeongno, Yuseong-gu, Daejeon, 304–700, Korea
\textsuperscript{a}) jeonggun.lee@hallym.ac.kr
\textsuperscript{b}) yuko@hallym.ac.kr

Abstract: In this paper, we design and analyze an asynchronous pipelined FIFO called a micropipeline with the awareness of “place & route” (P&R) on an FPGA device. We use a commercially available 65 nm Virtex-5 devices and design a high-speed implementation of the asynchronous four-phase micropipeline with considering its layout on the device. The layout of our design is modified manually to meet timing constraints and to accelerate the speed of circuits. The asynchronous FIFO implemented on the Virtex-5 device shows 452 MHz throughput and 648 ps per-stage latency at the simulation under the worst case operating condition and around 472 MHz throughput is observed at the actual measurement on a real working chip at room temperature.

Keywords: asynchronous circuit, FPGA, place & route, high-speed, Virtex-5, micropipeline

Classification: Science and engineering for electronics

References
1 Introduction

In the international technology roadmap for semiconductor (ITRS) [1], asynchronous circuit design techniques are considered as a promising design alternative for resolving the design problems particularly such as a circuit reliability issue caused by process, voltage and thermal variation occurred in a nanometer CMOS technology. In general, full custom or ASIC design techniques have been used widely for the implementation of asynchronous circuits. In a current design technology, however, an FPGA device is not only used for a prototyping platform anymore. The FPGA device is getting much attraction from industry in the name of “reconfigurable device” and those reconfigurable devices are expected to be used more frequently in the future chip market.

There has been only little work on the implementation of asynchronous circuits on FPGA devices. It is important for both synchronous and asynchronous designers to implement asynchronous circuits on those promising devices to exploit the traditional benefits of asynchronous circuits such as low power consumption, low electromagnetic interference, average case performance, and delay insensitivity. The implementation of the asynchronous circuits on commercial FPGA devices is rare due to the hardness of timing control for signal propagation delays. For the implementation of asynchronous circuits on an FPGA device, there have been two types of researches: (1) the design of new FPGA architectures for easily adapting asynchronous circuits [2, 3, 4], (2) the implementation of asynchronous circuits on currently available FPGA devices [5]. Recently, in [4], the proposed architecture has been commercialized but its impact on FPGA design society is marginal.

In this paper, we design and analyze a simple asynchronous pipelined FIFO called a micropipeline with awareness of “place & route” (P&R) on an FPGA device in order to show the feasibility of high-speed implementation of asynchronous circuits. We use 65 nm Xilinx Virtex-5 devices to design and implement the FIFO with layout adjustments to meet timing constraints which the circuits have to satisfy for their correct operations. The asynchronous FIFO implemented on the Virtex-5 device shows 452 MHz throughput and 648 ps per-stage latency at the simulation with worst case operating condition, while 472 MHz average-throughput is observed at the real measurement on a working chip at room temperature.
Design

In this section, we design high-speed asynchronous pipelined FIFO using an FPGA device and investigate various aspects that have to consider when high-speed asynchronous circuits are mapped onto the FPGA device.

2.1 Micropipeline design on an FPGA

Figure 1 (a) shows a micropipeline FIFO design target architecture. In the architecture, the most important circuit is a C-gate which has a role of synchronizing asynchronous signals between stages [6]. To implement the C-gates only with combinational gates and feedback signals in an FPGA, designers should care about the feedback signals which are automatically routed by commercial synchronous FPGA synthesis and P&R tools.

- **Micropipeline protocol selection**: There are many well-known micropipeline control circuits. Particularly in [6], several handshake control
circuits for the micropipelines have been proposed. Among the handshake control circuits, we choose “4-phase simple latch controller (SLC)” as our FIFO control circuits (shown in Figure 1(b)) to show the possible speed limits of asynchronous circuits on an FPGA. Note that the 4-phase SLC provides minimal cycle time (maximal throughput), but it allows only alternative stages to be occupied at most. The other advanced handshake controllers need more gates for their decoupled operations, and it seems that the complex circuits of the advanced handshake controllers cause the significant increase in the cycle time of the FIFO. The advanced handshake control circuits can be employed for better performance when combinational circuits are inserted in between micropipeline stages [6].

- **One LUT implementation of micropipeline handshake control circuits**: The cycle time of a micropipeline FIFO is proportional to the number of LUTs used in the implementation of handshake control circuits. In consequence, minimizing the number of LUTs for handshake control circuits is crucial to make high speed asynchronous circuits. Using two LUTs for a SLC in the stage control circuits causes the increase in the cycle time of FIFOs.

  For higher performance, both of the C-gate and the inverter in the SLC can be implemented in a single LUT by properly modifying programmable bits of the LUTs. Figure 1(c) shows the single LUT design of the SLC using an LUT primitive gate in the Virtex-5 library with Verilog-HDL [7]. In the description, “INIT(16’h00b2)” defines the configuration bitstream of the LUT which implements the logic equation for the SLC shown in Figure 1(b) with an output signal “out” and input signals “a”, “b”, “out”, “reset”.

  In this case, the cycle time of a stage is set to the sum of four LUTs delay and additional interconnect delays. The equation for the cycle time of a stage can be expressed in the following form.

  \[
  D_{\text{cycle}} = \frac{D_{\text{LUT}} + D_{\text{Fw}} + D_{\text{LUT}} + D_{\text{Bw}} + D_{\text{LUT}} + D_{\text{Fw}} + D_{\text{LUT}} + D_{\text{Bw}}}{\text{WorkingPhase}} + \frac{D_{\text{LUT}}}{\text{IdlingPhase}}
  \]

  Here, \(D_{\text{cycle}}\) is the cycle time of a FIFO stage and \(D_{\text{LUT}}\) is the propagation delay of a LUT. \(D_{\text{Fw}}\) and \(D_{\text{Bw}}\) are signal routing delays for forwarding a request to the next stage and backwarking an acknowledge to the previous stage, respectively. \(D_{\text{LUT}}\) is around 80 ps in 65 nm Virtex-5 FPGA devices. Finally, the cycle time of our micropipeline FIFO is determined by the longest cycle time among the cycle times of the micropipeline stages.

  In current advanced FPGA devices, interconnect delay is getting more dominant when compared to logic delay. In our timing analysis, the interconnect delay takes 83% of the worst cycle time delay in average even with P&P-awared local routings.

### 2.2 P&R design

In an FPGA device, it is hard to control timing delay among gate or circuit components. To make timing constraints be satisfied, special design
constraints should be given to synthesis and P&R optimization processes.

Xilinx synthesis and P&R tools support three useful constraints such as “LOC”, “RLOC” and “P-block” for controlling layout design [8]. In general, such a user-defined placement can increase the speed of circuits and makes die resources be used more efficiently. LOC and RLOC are the placement constraints specifying the absolute and relative positions of cells, respectively. The P-block constraint is supported by Xilinx PlanAhead and it allows to constraint circuit modules to a particular area of the FPGA device.

We can make a regular layout design with the manual settings of P-block and LOC constraints. Figure 1 (d)–(e) show the detailed layout view of the micropipeline FIFO shown in Figure 1 (a). Figure 1 (d) presents a placement of an SLC to an LUT in a Slice. Figure 1 (e) shows a detailed placed and routed design of our FIFO (from the 2nd stage to the 5th stage) mapped onto a Virtex 5 FPGA device. As shown in Figure 1 (e), control path (in the upper gray box) and datapath (in the lower gray box) circuit components are regularly placed. The layout design is performed using the Xilinx PlanAhead tool. The LUT/latch circuit components are manually placed to keep the relevant components closely and regularly be positioned.

To check the interconnect wire routing, an FPGA editor is used [8]. Through the editor, we have checked the feedback signals in the SLCs implemented in LUTs are routed very locally so that the timing constraints for the correct SLC implementation are satisfied.

We extract all the net delays from our design using ISE timing analysis tool and then its worst cycle time is analyzed statically. Through the analysis, the worst case cycle time is found as 2.22 ns (its equivalence rate is 450.04 MHz) that is very similar to 2.21 ns (its equivalence rate is 452 MHz), observed at the post-P&R simulation. In this case, the error rate is less than 1% between analysis and simulation.

2.3 I/O Environment: pulse-based data generation circuit

Feeding data to our micropipeline through simulation benchmarks have to use IOB nodes that cause relatively larger propagation delay when compared to those of LUTs or local wires. Due to the large delay on the input/output (I/O) blocks of FPGA devices, high speed operation of our micropipeline is limited significantly by the delay of an I/O environment. To feed data to our micropipeline with a high speed cycle time and to verify the working stability of the operation in the micropipeline, we implement a high-speed data generation circuit on an FPGA device.

Figure 2 shows our “pulse-based data generation circuit” and the circuit is used as input environment as shown in the upper figure of Figure 1. Pulses are generated by an XOR-gate and a delay element “delay-P”. The pulses at the XOR-gate are used as clock events for capturing new data when “ack” signal is high (It means that the first stage gets the data so that the input generator needs to produce new data). The AND-gate in the figure is used to allow only low-to-high events on the ack signal work as the clock events.

The generated data are also feed back to an adder in order to produce
next data by adding “1”. The delay element, “delay-F”, is added to the feedback path as shown in Figure 2 for satisfying hold time constraints of the latches. The delay element is implemented by configuring a single LUT as a buffer gate.

3 Experimental results

To show the effect of layout awareness, we design two asynchronous micropipelines: one without P&R consideration and the other with considering P&R. Figure 3 (a) shows the signal waves in the design with considering P&R. In the figure, “d1”, “d2”, . . . , “d5” are data captured at the stage 1, stage 2, . . . , stage 5, respectively. The P&R aware design shows the correct FIFO operations and data items are evenly spaced. On the other hand, some data are missing during its operation in the P&R unaware design and, furthermore it shows many timing violations at the simulation as presented in Figure 3 (b).

Our asynchronous FIFO design on a Virtex-5 device shows 452 MHz throughput at the simulation. Note that the throughput data at simulation and analysis are derived with the worst-case operating condition (Voltage = 0.95 V, Temperature = 85 °C). Furthermore, average per-stage latency is observed as 648 ps. In general, a linear FIFO has a drawback of long latency but our design can achieve short latency while keeping its linear topology.

When the design is downloaded onto the FPGA, the measured working frequency of our FIFO is 472 MHz in average at room temperature.

- Impact of voltage/thermal variation: In order to investigate the voltage and thermal variation impact of the 65 nm technology further, we observe variation of the throughput performance while changing two key process parameters: voltage and temperature.

The best-case throughput performance, 502.25 MHz, is obtained under the operating condition, “Voltage = 1.05 V, Temperature = 0 °C” and the worst-case throughput performance, 452.55 MHz, is obtained under the operating condition, “Voltage = 0.95 V, Temperature = 85 °C”. The worst-case performance is about 1.1 times slower than the best-case performance in the given variation range of voltage and temperature. It is noteworthy that the
operating frequency are changed without losing the functional correctness of the asynchronous FIFO as voltage and temperature varies.

4 Conclusions

The high-speed asynchronous micropipeline FIFO design is the most fundamental topic since it shows the limit of timing overhead introduced in the design of asynchronous circuits. In this paper, we design and analyze a simple but high-speed asynchronous micropipeline FIFO with the “place & route” (P&R) awareness on an FPGA device. We use a commercially available 65nm Xilinx Virtex-5 device to implement the high-speed micropipeline FIFO with the detailed layout adjustment to meet timing constraints. The asynchronous FIFO mapped onto the Virtex-5 device shows 452 MHz throughput and 648 ps per-stage latency at the worst-case operating condition in the simulation. When the design is tested on the real working chip, it shows 472 MHz throughput performance in average.
Acknowledgments

This work was supported by ETRI.