HARDWARE COST REDUCTION IN FAULT DETECTION MECHANISM FOR CONSTRUCTIVE TIMING VIOLATION TECHNIQUE

Kazuhiro Mima† Toshinori Sato†‡

Department of Artificial Intelligence, Kyushu Institute of Technology†
Center for Microelectronic Systems, Kyushu Institute of Technology‡
680-4 Kawazu, Iizuka, 820-8502 Japan
{kazuhiro, tsato}@mickey.ai.kyutech.ac.jp

Abstract : In this paper, we propose to reduce the hardware cost of an ALU that utilizes Constructive Timing Violation (CTV) technique. The hardware cost required by the previously proposed CTV technique is about three times bigger than that of the baseline ALU. In order to reduce the cost, we propose to share a part of ALU between the main and cheker parts in the CTV mechanism, and to utilize pipelining technique in the fault detection circuit. We implement the proposed mechanism in a carry select adder using Verilog-HDL and logic synthesis. From the detailed simulations, it is observed there is the potential that clock frequency can be boosted by 1.4 – 1.6 times. However, in practice, there are several problems found.

1 INTRODUCTION

In recent years, high performance and low power microprocessors are required not only by large-scale computers but also by mobile devices such as PDAs (Personal Digital Assistance). For large-scale computers, increasing clock frequency is one of the simple ideas in order to gain processor performance. However, heating due to increasing power comsumption probably destroys processors. On the other hand, high perfomance microprocessors are required by recent mobile devices. For example, JAVA applications and video processing are often exectued on such devices. In addition, since the devices are battery-operated, reducing power consumption is an important factor.

The active power P_{active} and gate delay t_{pd} of a CMOS circuit are given by

$$P_{active} \propto f C_{load} V_{dd}^2 \quad (1)$$

$$t_{pd} \propto \frac{V_{dd}}{(V_{dd} - V_{th})^{\alpha}} \quad (2)$$

where f is clock frequency, C_{load} is load capacitance, V_{dd} is supply voltage, and V_{th} is threshold voltage of the device. α is a factor depending upon the carrier velocity saturation and is about 1.3 – 1.5 in advanced MOSFETs. Based on Eq.(1), to reduce power consumption and to keep high perfomance, decreasing power supply voltage, V_{dd}, is required without decreasing clock frequency. However, Eq.(2) tells us that power supply voltage reduction increases gate delay, and thus clock frequency has to be decreased. Therefore, keeping clock frequency causes timing violations due to unsatisfied timing constraints.

In order to relax the timing constraints, we proposed Constructive Timing Violation (CTV) technique [1]. The CTV exploits the facts that critical path in a circuit is not always active and that there are considerable margins in LSI design. That is, we propose an LSI design methodology focusing on typical cases rather than worrying about very rare worst cases. It has a fault tolerant mechanism for timing violation based on a kind of parallelism; that is space redundancy. However, there is much hardware in its timing violation detection mechanism. Hardware cost required by the proposed technique is about three times bigger than that of the conventional one. It is a serious problem for not only low cost microprocessors but also for high performance microprocessors.

This paper considers hardware cost reduction in the fault detection mechanism for the CTV technique, and evaluates the proposed mechanism adopted in an ALU. In Section2, we describe the ALU utilizing the original CTV technique. In Section 3, we describe an ALU structure utilizing a new CTV technique which requires hardware cost. In Section 4, we describe the evaluation environment for the ALU utilizing new CTV technique. In Section 5, we describe the evaluation results. And last, we present our conclusion.

2 Overview of CTV Technique

Figure 1 shows the ALU structure utilizing the original CTV technique. It consists of a main part and a checker part. The main part includes an ALU and works at f_{dd} that is higher than that meets the critical path delay. Therefore, the main part is implemented so that it can maintain throughput and low latency, but timing violations probably occur. On the other hand, the checker part includes two ALUs and two comparators in order to check if the main part is safe. Each pair of an ALU and a comparator in the checker part works at $\frac{f_{dd}}{2}$, and thus timing violations do not occur. Clocks provided for the pairs have different phase, and they alternatively check the main part to detect timing violation.

Circuits utilizing the CTV technique can work at higher frequency or with less power consumption. Working at higher frequency will be easily understood. It is assumed that currently the circuit works at $\frac{f_{dd}}{2}$ without timing violations, and it is expected that the clock is increased to f_{dd}. Since the proposed mechanism detects timing violations and a recovery mechanism tolerates them, it is possible for the circuit to work at f_{dd} with maintaining its throughput and latency.

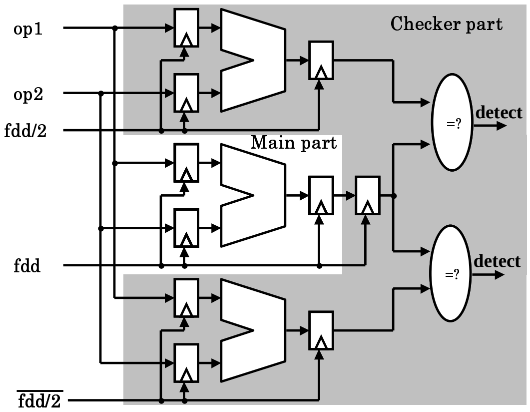


Figure 1: ALU design utilizing CTV

We explain how power consumption is reduced as follows. In this estimation, it is assumed that the ALU safely works at f_{dd} whan its supply voltage is V_{dd}. It is easily found that decreasing power supply voltage is the most effective way to reduce power consumption from Eq.(1). In order to reduce power consumption, we would like to decrease the supply voltage to $\frac{V_{dd}}{2}$. Usually, the clock frequency should be decreased to $\frac{f_{dd}}{2}$ determined by the supply voltage $\frac{V_{dd}}{2}$, but we keep it as f_{dd} in the main part. The checker part is used for detecting the timing violations and thus it works at the frequency $\frac{f_{dd}}{2}$ with the supply voltage $\frac{V_{dd}}{2}$ without timing violations. Because the energy reduction due to the lower supply voltage is larger than the increase of energy consumption caused by amount of parallelism, the CTV can decrease energy consumption efficiently. In this case, the total power consumption of the three ALUs in the main and the checker parts is half that of the conventional one. We have already evaluated the improvements in performance and energy efficiency via circuit simulations [2, 3].

The additional checker part has twice bigger hardware than the main part does. Therefore, the hardware overhead cannot be ignored even if the amount of circuit required for implementing the CTV technique is small. In the next section, we explain how to reduce hardware cost of the checker part, which is the essential component in the CTV to detect timing violations.

3 Reducing Hardware Cost in Fault Detection Mechanism

Figure 2 shows an n-bit ALU structure utilizing CTV technique with reduced hardware cost. The ALU consists of a main part and a checker part. The main part is divided into a lower n/2-bit ALU and an upper n/2-bit ALU, and works at f_{dd}. The checker part is also divided into a lower n/2-bit ALU, an upper n/2-bit ALU, and a comparator. The lower ALU is shared by the main part and the checker part. In the checker part, the lower and upper ALUs are pipelined so that they are able to work at f_{dd} without timing violation and so that the checker part keeps up with the progress in the main part; i.e. the main and checker parts' throughput are same. As you can see, the additional circuit only requires the upper ALU, the comparator, and a few flipflops for pipelining.

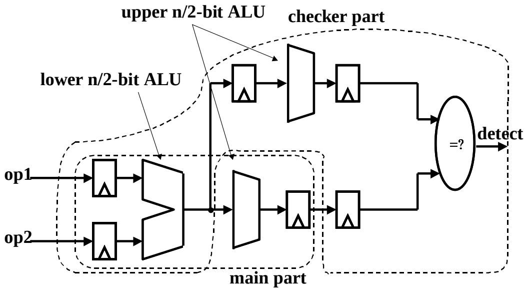


Figure 2: ALU design utilizing improved CTV

The improved CTV technique has three advantages over the original one in addition to the hardware cost reduction in the checker part.

- It requires only one clock domain, f_{dd}, in order to drive whole circuit. Therefore, the chip area is significantly reduced since the clock generator of $\frac{f_{dd}}{2}$ and complementary two clock lines are not required.
- The static power consumption due to leakage current is decreased due to the reduction in the number of transistors.
- The active power consumption of the chekcer part ALU is reduced due to sharing the lower ALU between the main and the checker parts.

We explain the detail of third advantage of the improved CTV under the same situation of the example in the previous section. It is assumed that the circuit in Figure 2 works at f_{dd} with supply voltage V_{dd}. We would like to decrease supply voltage to $\frac{V_{dd}}{2}$. Under this condition, it is assumed that timing violations probably occur in the main part and do not occur in the checker part. Based on Eq.(1), the active power of the ALUs in the main and checker parts is as follows

$$\begin{aligned}
&P_{main} + P_{check} \\
&\quad= N f C_{load} (\frac{V_{dd}}{2})^2 + \frac{N}{2} f C_{load} (\frac{V_{dd}}{2})^2 \\
&\quad= \frac{3}{8} N f C_{load} V_{dd}^2 \qquad (3)
\end{aligned}$$

where N is number of transistors in an ALU. It is assumed that the number of transistors in the checker ALU is half that of the main ALU. Since the power consumption of a conventional ALU is $N f C_{load} V_{dd}^2$, the power consumtion of the improved CTV technique is reduced by 62.5% from that of the conventional ALU. Moreover, utilizing the improved CTV technique can expect 12.5% power reduction compared to the original CTV technique.

4 Evaluation Environment

In order to evaluate the improved CTV technique, we impremented a carry select adder (CSLA)[4] using Verilog-HDL. After verifying the correctness of its function by logic simulations, we logic-synthesized it in order to estimate gate delay. The clock frequency is boosted from that determined by estimated critical path delay, and we measure error rates in the main part. At the same time, we also measure what times the clock frequency is boosted without errors in the checker part. Figure 3 shows the evaluation circuit. The circuit enclosed by dotted line is the CSLA utlizing the improved CTV technique, that includes delay information. The lower and upper 16-bit CSLAs construct the 32-bit main CSLA. A 16-bit CSLA and a comparator are included by the checker part. The three comparators in the right side of Figure 3 check whether the main CSLA, the checker CSLA and the comparator in the checker part are safe. The "ideal result" shown at the top and bottom comparators are correct values delivered by the 32-bit CSLA without delay information. Another "ideal result" is from the whole evaluation circuit without delay information.

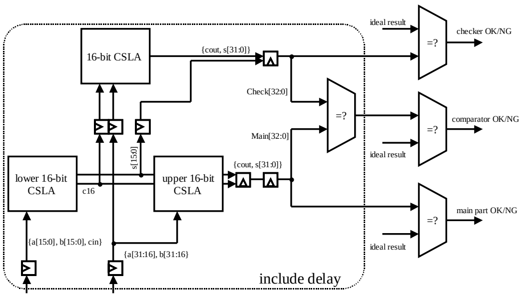


Figure 3: Evaluation Circuit

Cadence Verilog-XL was used for logic simulation. In order to perform logic simulation, we made test vectors using an instruction level simulator, sim-safe, included in SimpleScalar/PISA tool set [6] and SPEC2000 CINT benchmark. The test vectors were created for additions and subtractions executed as add, sub, load and store instructions. We made 10 thousand patterns for each instruction. The logic-synthesis is performed by Synopsis DesignCompiler using Hitachi's $0.18\mu m$ CMOS process technology provided by VDEC[5].

5 Evaluation Results

In this section, we present evaluation results. We measure error rates of the main part without errors of the checker part. They are shown in Figure 4. The horizonal axis of each graph shows boosting ratio of the clock frequency, and the vertical line shows error rates of the main part. In the cases of `164.gzip` `176.cc1`,`197.parser`, `256.bzip2`, and `300.twolf`, it is observed that the frequency of the CSLA can be boosted by 1.4-1.6 times and the error rate is less than 30%. Our previous studies unveiled [1, 2, 3] that power reduction or performance improvement is expected if error ratio is less than 30%. Therefore, the validity of the improved CTV technique can be found in this case. Moreover, since the error rate of all benchmarks is 0% until the boosting ratio is increased to 1.3, the clock frequency can be boosted without any penalty due to recoveries from faults.

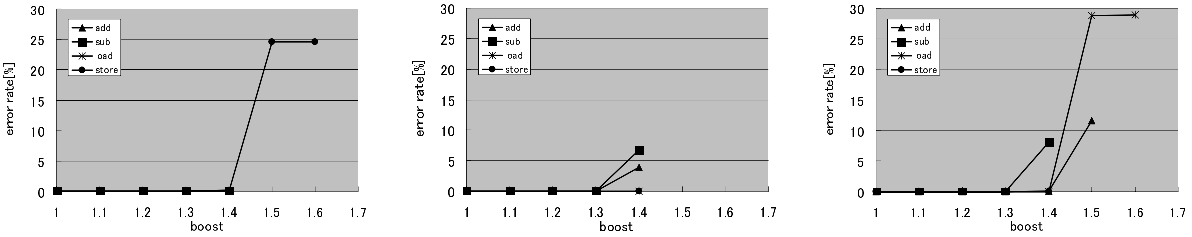
On the other hand, in the cases of `175.vpr`, `181.mcf` and `255.vortex`, the frequency can be only boosted by 1.4 times. Beyond this point, errors are found in the checker part. The error rate is about 10% when boosting ratio is 1.4. Therefore, the potential of CTV cannot be exploited. This problem is caused by pipelining the checker part. We consider the problem as follows. According to the timing-report of logic-synthesis, the critical path delay in the main part is 1.54 ns, and that in the checker part is 1.44 ns. Therefore, using pipelining, the delay in the checker part can be reduced by only 10% of the critical path delay in the main part. In other words, the checker part might cause timing violations when the boosting ratio is bigger than 1.1. The reasons are as follows. First, pipelining the CSLA to the two-stage CSLA can reduce just the delay of a 2-to-1 multiplexer. Second, adding flipflops required by piplelining increases the delay of each pipeline stage in the checker part. In the logic simulation, timing violations was not found in the checker part when the boosting ratio is less than 1.4. However, boosting ratio of the frequency should be 1.1 in order to execute with keeping safety. This problem can be solved by re-examining the optimization constraints for the logic-synthesis, but we don't think that it is a fundamental solution.

In order to avoid this problem, increasing the number of pipeline stages is one of the simple ideas. This solution increases clock frequency. However, since the latency of the checker part is increased, the penalty for the recoveries due to faults is also increased.

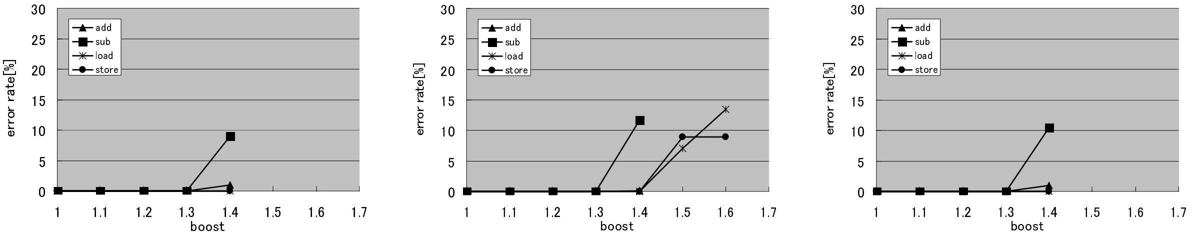
6 Conclusion

In this paper, we proposed to reduce the hardware cost of an ALU that utilizes CTV technique. The CSLA utilizing the improved CTV technique was impremented by Verilog-HDL. From the logic simulation, the clock frequency of the CSLA can be boosted by 1.4 –1.6 times with low error rates. However, in practice, we can boost the clock frequency by only 1.1 times according to the timing-report of the logic-synthesis. Therefore, we have to re-examine the pipelining in the checker part in order to execute at higher clock frequency.

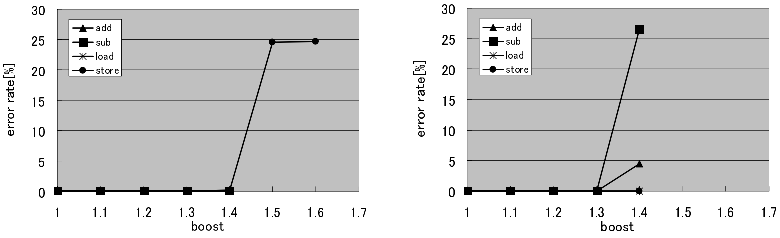
Future studies regarding the CTV technique are as follows. We have to design and to evaluate the deeper pipelined ALUs to have a gain in clock frequency. We are currently considering some technique to mitigates the penalties due to timing errors, which is increased according to the deeper pipeline. It can detect every error at the earlier stages. We are also interested in exploiting the profile information on occurrence of errors. We are investigating a hardware profiler resembling the branch target



(a)164.gzip (b) 175.vpr (c) 176.cc1



(d)181.mcf (e) 197.parser (f) 255.vortex



(g) 256.bzip2 (h) 300.twolf

Figure 4: Simulation Result

buffers. If we can predict if a timing error occurs, we use the outcome from the cheker part instead of that from the main part. This also mitigates the penalties due to the errors.

Acknowledgement

This work is supported in part by a financial gift from Toshiba Corporation semiconductor company and Grant-in-Aid for Scientific Research from Ministry of Education, Culture, Sports & Technology (No. 15650010).

References

[1] T.Sato, I.Arita, “Constructive Timing Violation for Improving Energy Efficiency,” in L.Benini, M.Kandemir, J. Ramanujam, “Compilers and Operating Systems for Low Power,” Kluwer Academic Publishers, September 2003.

[2] A.Tanino, T.Sato, “Evaluating the Potential of an Energy Reduction Technique Based on Timing Constraint Speculation,” 4th Workshop on Compilers and Operating Systems for Low Power, September 2003.

[3] A.Tanino, T.Sato, “Simplifying High-Frequency Microprocessor Design via Timing Constraint Speculation, ” 16th International Conference on Computer Applications in Industry and Engineering, November 2003.

[4] A. Chandrakasan, W. J. Bowhill, F. Fox, “Design of High-Performance Microprocessor Circuits,” IEEE Press, 2000.

[5] VLSI Design and Education Center (VDEC), the University of Tokyo, http://www.vdec.u-tokyo.ac.jp.

[6] T.Austin, E.Larson, D.Ernst, “SimpleScalar: An Infrastructure for Computer System Modeling,” IEEE Computer, February 2002.