An Evaluation of Constructive Timing Violation via CSLA Design *

Asami Tanino† Toshinori Sato†‡ Itsujiro Arita†

†Department of Artificial Intelligence, Kyushu Institute of Technology
‡Center for Microelectronic Systems, Kyushu Institute of Technology
680-4 Kawazu, Iizuka, 820-8502 Japan
Tel:+81-948-29-7624 Fax:+81-948-29-7601
toshinori.sato@computer.org

1 Constructive Timing Violation

It is the situation that wire delay in circuit determines clock frequency of a chip rather than gate delay, with progress of semiconductor process technology. Moreover, frequency of microprocessor continues improving and it is becoming difficult to satisfy timing constraints. Recently, we proposed Constructive Timing Violation (CTV) technique [1] in order to relax timing constraints in nanometer scale design. It relies on a fault-tolerant mechanism based on speculative execution technique. If we provide some fault-tolerant mechanism for timing faults, it is not necessary to keep the constraints.

We explain our proposal using an adder. It is assumed that currently the adder executes at clock frequency of f, and it is expected that the clock is increased by two times, that is $2f$. First, the adder is duplicated by three times. One adder, which we call main adder, works at $2f$ and the remaining two adders, which we call checker adders, work at f. Thus, the checker adders are free from timing errors and are used for verifying the operation of the main adder. Since the clock signals of the checker adders are complementary with each other, they work alternatively to verify the main adder. The verification is based on comparing two execution results from the main adder and corresponding one of the checker adders. If they do not match, a timing fault is detected. In such cases, any recovery action should be initiated. In order to revert processor state to a safe point where the error is detected, we propose to utilize the recovery mechanism used in modern microprocessors for speculative execution. In other words, a timing fault of an instruction is regarded as a mispredicted branch instruction. Thus, there are no hardware overhead in the recovery mechanism.

2 Evaluation

In order to model the relationship between boosted clock frequency and timing error rate, we designed Carry Select Adder (CSLA). A similar evaluation on Carry Lookahead Adder has been done by other research group [2]. We implemented our CSLA using Verilog-HDL and logic-synthesized it using DesignCompiler. The synthesized CSLA includes estimated gate and wire delay based on a standard ASIC library. In order to perform logic simulation, we made test vectors from functional simulation of SPEC CINT2000 benchmark suit. We classified operations regarding adders into four categories: `add`, `sub`, `load`, and `store`. Figure 1 shows logic simulation results in the case of `176.gcc`, when we boosted processor clock frequency by the order of $1.1 - 3.0$. The horizontal line indicates the frequency and the vertical line indicates timing error rate. We can find that error rate is less than 20% when clock frequency is two times faster than that satisfies timing constraints due to critical paths. In our preliminary evaluation, we have already evaluated how timing violations affect processor performance, and found that 30% of timing error rate did not have severe impact on performance. Thus, processors may tolerate the measured 20% of error rate, and it is confirmed that the CTV technique is effective on boosting processor performance and on simplifying timing design.

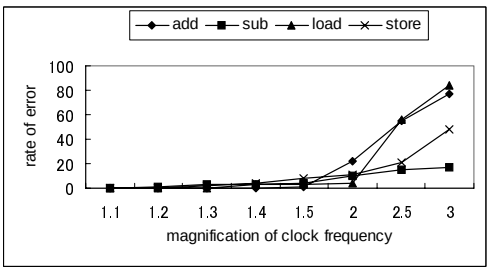

Figure 1: Simulation results (gcc)

*This work is supported in part by a financial gift from Toshiba Corporation.

References

[1] T. Sato and I. Arita, "Give up meeting timing constraints, but tolerate violations," Cool Chips IV, 2001.

[2] T. Matsuo, et al., "Dependable pipelining," Technical Report of IEICE, DC2002-20, 2002 (in Japanese).