

Performance Improvement of General-Synchronous Circuits by Variable Latency Technique using Dynamic Timing-Error Detection

Shimpei Sato

Dept. of Information and
Communications Engineering
Tokyo Institute of Technology
Meguro-ku, Tokyo, Japan
satos@ict.e.titech.ac.jp

Hiroshi Nakatsuka

Dept. of Communications and
Computer Engineering
Tokyo Institute of Technology
Meguro-ku, Tokyo, Japan
nakatsuka@eda.ce.titech.ac.jp

Atsushi Takahashi

Dept. of Information and
Communications Engineering
Tokyo Institute of Technology
Meguro-ku, Tokyo, Japan
atsushi@ict.e.titech.ac.jp

Abstract— General-synchronous circuits have a better performance compared to a complete-synchronous circuit. The performance of them is expected to be further improved by allowing speculative execution. In this paper, a high performance general-synchronous circuits with speculative execution is realized as a variable latency circuit by adopting error-detection and correction mechanism. In our proposed method, a circuit is designed by combining clock scheduling, delay insertion, and speculation effectively. In experiments, we confirmed that 6.7% performance improvement is achieved compared to a general-synchronous circuit with fixed latency.

I. INTRODUCTION

Most of digital circuits nowadays are designed as clock synchronous circuits with global clocks. In a typical clock synchronous circuit implementation, a global clock is designed to be inputted to every flip-flop simultaneously, and every primitive computation is executed in one clock cycle. In such implementation, the performance of a circuit depends on the clock period, and the maximum delay of primitive computations gives a lower bound of the clock period. Therefore, in order to maximize the performance of a circuit, the reduction of the maximum delay of primitive computations is pursued if primitive computations executed in one clock cycle are given as input [1].

In this paper, we expand the concept of general-synchronous circuit [2, 3, 4, 5] to handle speculative execution of a primitive computation. In general-synchronous circuit, the clock is assumed to be distributed periodically to each individual register though is not necessarily to all the registers simultaneously. Performance improvement methods [3, 4, 5] in general-synchronous circuit proposed so far do not take speculative execution of primitive computation into account. By allowing speculative execution in general-synchronous circuit, a smaller clock period of circuit is allowed. Even though the number of clock cycles required to finish computations is increased, the performance is expected to be improved by adopting an appropriate clock period.

Delay insertion [3, 4] is a remarkable performance improvement method of general-synchronous circuit. In general-

synchronous circuit, the increase of the minimum delay between registers leads to the clock period minimization. Even though we do not consider the detailed delay insertion methods, the delay insertion will be realized by replacing a large module with a small module synthesized under looser delay constraints, by using smaller transistors and narrower wires, and by deleting buffers from long interconnects, as well as by inserting buffers to short interconnects.

Variable latency circuit [6, 7, 8, 9] in which each primitive computation is not done in fixed latency has potential to ease the constraints which limit the clock period minimization and to improve the circuit performance remarkably. A dynamic timing error detection/correction mechanism [9], which is a implementation of variable latency circuit, enables a circuit works correctly under assumption of a timing-error occurrence. Using this mechanism, a circuit with a minimized clock period executes its primitive computation speculatively. When it detects a timing-error, a time for error correction is required. Therefore, the performance of a circuit using the mechanism depends on the timing-error rate.

In this paper, we propose a performance improvement method of general-synchronous circuits using the variable latency technique. The dynamic timing-error detection/correction mechanism is used as a variable latency technique. Delay insertion method is also used for a timing-error reduction. In the evaluation, we show the performance improvement of a circuit by gate level simulation.

II. LIMITATION OF THE MINIMUM CLOCK PERIOD OF GENERAL-SYNCHRONOUS CIRCUIT

A. General-synchronous circuit

In this paper, we consider a circuit consisting of registers and gates, and wires connecting them. We refer to them as elements. A circuit is represented by the graph $G = (V_g, E_g)$, where V_g is the vertex set corresponding to elements in the circuit and E_g is the directed edge set corresponding to signal propagation in the circuit. In this paper, we assume that the maximum delay of each element is equal to its minimum delay. Let $d(v)$ be the weight of $v \in V_g$ which corresponds to the delay

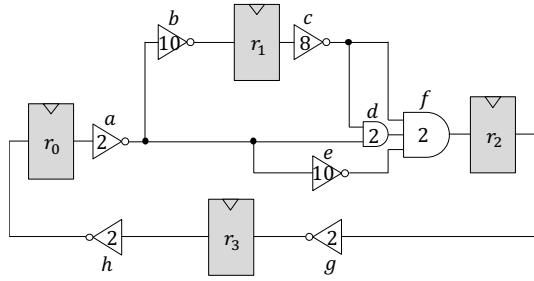


Fig. 1. An example of a circuit G .

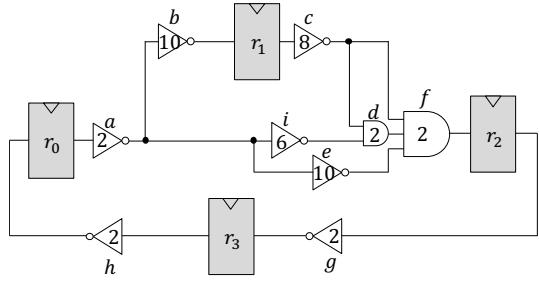


Fig. 3. An example of a circuit G_1 . Delay element i is inserted to the circuit G .

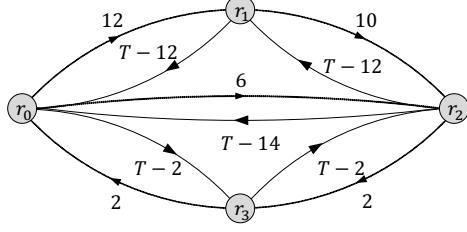


Fig. 2. The constraint graph $H(G)$ of the circuit G .

of corresponding element. Let V_r be a register set. Necessarily, the register set is a subset of V_g .

Fig. 1 shows an example of a circuit. In Fig. 1, $\{r_0, r_1, r_2, r_3\}$ is the register set, and the figure in each vertex except registers represents its weight. In general-synchronous circuit, the clock arrival timing of a register may be different from other registers. The clock timing $S(r)$ of a register r is defined as the difference in clock arrival time between r and an arbitrary chosen reference register. Moreover, the set of clock timing of all the registers S is called *clock schedule*.

A circuit works correctly with a clock period T if the following two types of constraints are satisfied for every register pair with signal propagation's [10].

Setup Constraint

$$S(u) - S(v) \leq T - D_{\max}(u, v) \quad (1)$$

Hold Constraint

$$S(v) - S(u) \leq D_{\min}(u, v) \quad (2)$$

where $D_{\max}(u, v)$ is the maximum delay and $D_{\min}(u, v)$ is the minimum delay from a register u to v .

Since a clock ticks all the register simultaneously in complete-synchronous circuit, the clock period must be larger than or equal to the maximum delay between registers. On the other hand, in general-synchronous circuit, circuits can work correctly with the clock period which is smaller than the maximum delay between registers, if all the register pair with signal path satisfies two types of constraints.

A *constraint graph* can be defined from a given circuit. The constraint graph $H(G) = (V_r, E_r)$ for the circuit G is shown in

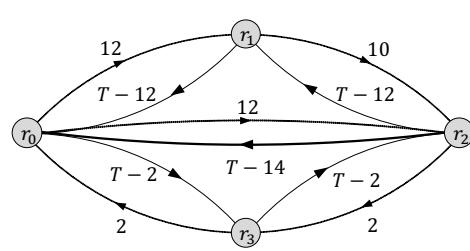


Fig. 4. The constraint graph $H(G_1)$ of the circuit G_1 .

Fig. 2. Where vertex set V_r corresponds to registers in G and directed edge set E_r corresponds to two types of constraints [2, 11]. In the constraint graph, a register pair is connected by two edges. One is an edge corresponds to the hold constraint (a dotted arrow in the graph) and the other is an edge corresponds to the setup constraint (a solid arrow in the graph).

Let $T_S(G)$ be the minimum clock period of a general-synchronous circuit G under the assumption that the clock can be inputted to each register at an arbitrary designated timing. Hereafter, we simply call $T_S(G)$ the minimum clock period of G . $T_S(G)$ is determined by the constraint graph $H(G)$. Let the weight of a directed cycle in $H(G)$ be the sum of edge weights on the directed cycle. It is known that the minimum clock period $T_S(G)$ is the smallest clock period such that there is no cycle with negative weight in the constraint graph $H(G)$.

In the example of the constraint graph $H(G)$, the clock period that satisfies all the cycles in $H(G)$ do not have negative weight is 9. The weight of the directed cycle (r_1, r_0, r_2, r_1) is negative when the clock period is less than 9. Then the minimum clock period $T_S(G)$ in the example is 9.

B. Limitation of the minimum clock period of general-synchronous circuit

Here, we consider the limitation of the minimum clock period of general-synchronous circuit which clock period is reduced by delay insertion [4]. Fig. 3 shows a circuit G_1 in which a delay element i is inserted on the path from r_0 to r_2 of the circuit G . The weight of the delay element is 6. Fig. 4 shows the constraint graph $H(G_1)$ for the circuit G_1 . By delay inser-

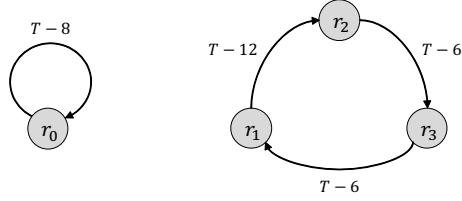


Fig. 5. Examples of a cycle consisting of setup constraint edges.

tion, the weight of the hold constraint edge between r_0 and r_2 changes from 6 to 12.

In the example of the circuit G , the minimum clock period is limited by the directed cycle (r_1, r_0, r_2, r_1) . In the graph $H(G_1)$, the weight of the hold constraint edge (r_0, r_2) is 12 and the clock period expected from the directed cycle (r_1, r_0, r_2, r_1) is 6. In this way, when the minimum clock period of a general-synchronous circuit is limited by a directed cycle including a hold constraint edge, the limitation to the clock period can be relaxed by delay insertion to the edge.

The minimum clock period of the circuit G_1 is 7 which is limited by the directed cycle $(r_0, r_3, r_2, r_1, r_0)$. The cycle is composed without including a hold constraint edge. When the minimum clock period of a general-synchronous circuit is limited by a cycle which is not including a setup constraint edge such as examples in Fig. 5, delay insertion has no effect to reduce the clock period of the circuit. Therefore, there is a theoretical limitation of clock period minimization of general-synchronous circuit under the assumption that a timing-error does not occur in the circuits.

III. GENERAL-SYNCHRONOUS CIRCUIT WITH VARIABLE LATENCY TECHNIQUE

Here, we propose a general-synchronous circuit using variable latency technique. The dynamic timing-error detection/correction mechanism [9] is used to realize a variable latency circuit. Using this mechanism, we aim to achieve the performance improvement of a general-synchronous circuit by running it under a small clock period which is assumed to occur a timing-error.

A. Variable latency technique using dynamic timing-error detection

The latency of a circuit is the time required to generate the outputs after the inputs are given. Generally the clock period of a circuit is determined by it. Dynamic timing-error detection mechanism changes the latency according to the time required to generate the output signals. Using this mechanism, circuits work correctly with a clock period which is smaller than the maximum latency of the circuits. If no timing-error is detected, circuits with this mechanism runs correctly. When a timing error is detected, the wrong values has to be corrected by the mechanism to keep circuits work correctly.

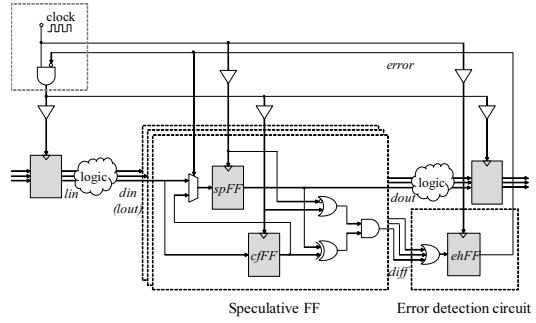


Fig. 6. Implementation of the dynamic timing-error detection/correction mechanism.

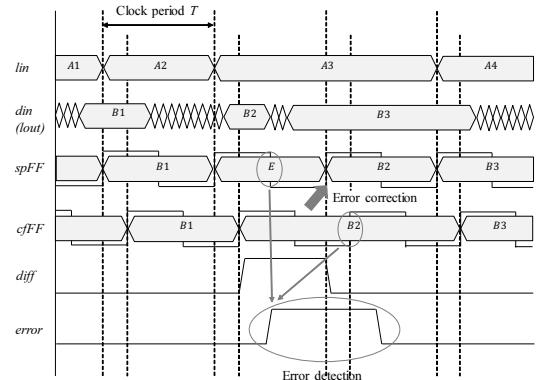


Fig. 7. A timing chart in the case of timing-error.

Fig. 6 shows the overview of the implementation of dynamic timing-error detection/correction mechanism. In this implementation, a conventional deterministic flip-flop is replaced by a speculative flip-flop. A speculative flip-flop contains two conventional deterministic flip-flops, called $spFF$ and $cfFF$. A timing-error caused at $spFF$ is allowed, while no timing-error is allowed at $cfFF$. The value stored at $spFF$ is erroneous but is available earlier. While, the value stored at $cfFF$ is errorless but is available later. The values stored at $spFF$ is used as an “output” signal of this circuit, and the following primitive computations start earlier.

A timing-error is detected by comparing the values of stored at $spFF$ and $cfFF$ in the error detection circuit. The result of the comparison is stored at $ehFF$ and the flip-flop controls the error correction process. When $ehFF$ outputs a error signal, all of deterministic flip-flops stop updating their state by gating the clock signal supply for one clock cycle. In the speculative FFs, the value at the $spFF$ is replaced by the value at the $cfFF$ while stopping deterministic flip-flops. Then, the circuit returns to execute its primitive computations from the next clock cycle. Fig. 7 shows a timing chart that indicates a behavior of timing-error detection/correction of the circuit.

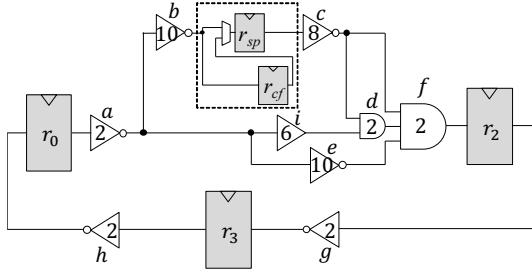


Fig. 8. A circuit G_{sp} using variable latency technique.

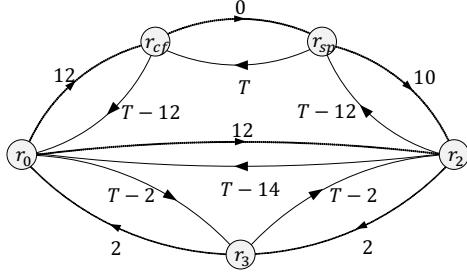


Fig. 9. The constraint graph $H(G_{sp})$ of the circuit G_{sp} .

B. Clock period reduction of general-synchronous circuit using variable latency technique

Here, general-synchronous circuit using variable latency technique that we propose is explained. Using our proposed method, the clock period of a circuit can be reduced smaller than the minimum clock period of the general-synchronous circuit. Reducing the clock period enables a circuit works faster. However, possibility of timing-error occurrence will increase. We aim to achieve performance improvement of a circuit considering the performance trade-off between clock period and timing error rate.

A circuit G_{sp} shown in Fig. 8 is a general-synchronous circuit using variable latency technique. The circuit G_{sp} is implemented by applying the variable latency technique to the circuit G_1 shown in Fig. 3. The register r_1 in the circuit G_1 is replaced by r_{sp} and r_{cf} for dynamic timing-error detection. Note that the figure shows only the speculative FF and the error detection circuit is omitted.

Fig. 9 shows the constraint graph $H(G_{sp})$ of the circuit G_{sp} . The register r_0 and the register r_{sp} are wired in the circuit. However there is no constraints between them because the register r_{sp} updates the state speculatively. On the other hand, the register r_{sp} and the register r_2 are required to satisfy the constraints. Also the register r_0 and the register r_{cf} are required to satisfy the constraints. Considering the timing constraint between the register r_{cf} and the register r_{sp} , the weight of the setup constraint edge between them is set to be T and the weight of the hold constraint edge between them is set to be 0.

In the constraint graph $H(G_1)$ shown in Fig. 4, the cycle which determines the minimum clock period is the cycle $(r_0, r_3, r_2, r_1, r_0)$. By introducing the speculative FF as the alternative of the register r_1 , the constraint is relaxed to use a smaller clock period than the minimum clock period of G_1 . The total weights of the cycle $(r_0, r_3, r_2, r_{sp}, r_{cf}, r_0)$ in $H(G_{sp})$ is larger than the total weights of the cycle $(r_0, r_3, r_2, r_1, r_0)$ in $H(G_1)$ under the same T . That is, a smaller clock period can be set in the circuit G_{sp} than the circuit G_1 .

C. Timing constraint

For using the variable latency technique in a general-synchronous circuit, Two new constraints for the error detection/correction mechanism are required in addition to the setup constraint and the hold constraint. They are *Error signal constraint* and *Clock gating constraint*. Error signal constraint is the constraint required to put together the set of outputs from spFFs and cfFFs into a ehFF. Clock gating constraint is the constraint required to handle the clock supply when an error is detected.

Let F_{nr} be the set of deterministic flip-flops, F_{sp} be the set of sfFFs, and F_{cf} is the set of cfFF. $D_{\max}(u, v)$ is the maximum delay and $D_{\min}(u, v)$ is the minimum delay from a register u to v . $S(\cdot)$ is the clock scheduling of a register. A general-synchronous circuit using variable latency technique works correctly with a clock period T if the following constraints are satisfied for every register pair with signal propagation.

Normal constraints

$$P_{\text{normal}} = \{(u, v) | u \in F_{nr} \cup F_{sp}, v \in F_{nr} \cup F_{cf}\}$$

$$\begin{aligned} \forall (u, v) \in P_{\text{normal}}, \\ S(u) - S(v) &\leq T - D_{\max}(u, v) \\ S(v) - S(u) &\leq D_{\min}(u, v) \end{aligned}$$

Error signal constraint

$$\begin{aligned} \forall w \in F_{sp} \cup F_{cf}, \\ S(w) - S(\text{ehFF}) &\leq -D_{\max}(w, \text{ehFF}) \\ S(\text{ehFF}) - S(w) &\leq T + D_{\min}(w, \text{ehFF}) \end{aligned}$$

Clock gating constraint

$$\begin{aligned} \forall x \in F_{nr} \cup F_{cf}, \\ S(x) &\leq S(\text{ehFF}) \leq T + S(x) \end{aligned}$$

IV. EVALUATION

A. Methodology

Here, we show the performance of general-synchronous circuit using variable latency technique. For evaluations, we implement a 5-stage pipelined processor of MIPS I instruction set as a general-synchronous circuit with variable latency technique. The performance of the circuit is measured by gate level simulation while executing some applications on the processor.

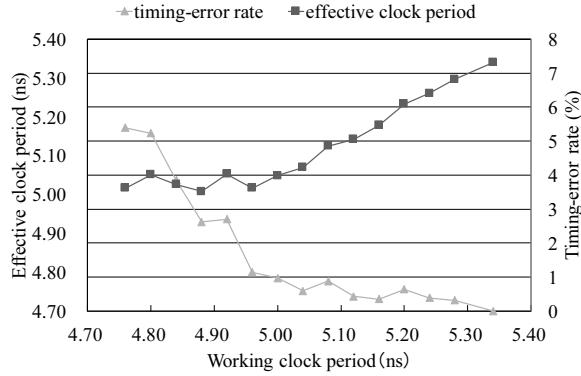


Fig. 10. Effective clock period and timing error rate.

We implemented the processor in Verilog HDL and synthesize it by Synopsys Design Compiler version I-2013.12-SP5 using ROHM 0.18 μm standard cell library. For the gate level simulation, we use Synopsys VCS version I-2014.03-SP1-5. Note that the wire delay is not considered in this simulation.

The performance of the circuit is measured by the executing time of application programs on it. As benchmark applications, we use “Bubble Sort”, “Quick Sort”, “Eight Queens”, “Towers of Hanoi”, “Puzzle”, and “Permutations” included in Stanford Integer Benchmark Suite [12]. The compiler for the application compilation is GCC version 4.3.3 with optimization option O2.

General-synchronous circuit requires to schedule the clock arrival timing to the registers in the circuit. We use a scheduling algorithm proposed in the paper [5].

B. Performance

In a variable latency circuit, the primitive computations of the circuit is stalled during the error correction process. Therefore, the performance of the circuit depends on the clock period and timing-error rate. To compare the performance of variable latency circuits, we introduce a metric called *Effective clock period* (T_{eff}). We call the clock period used for a circuit running as *Working clock period* for easy understanding. T_{eff} is defined as below:

$$T_{\text{eff}} = T(1 + \text{error_rate}) \quad (3)$$

Where T is the working clock period and *error_rate* is a timing-error rate of the circuit.

According to the synthesis report, the minimum working clock period of the processor as a general-synchronous circuit without using variable latency technique is 5.34 ns. We call this circuit as G_{org} . Delay elements are inserted appropriately not to limit the working clock period by a cycle including hold constraint edge. Therefore, the working clock period 5.34 ns is the smallest minimum working clock period of the processor available as a general-synchronous circuit without variable latency technique.

As a general-synchronous circuit using variable latency technique, we implemented 14 variations of the processor which were given different working clock period. The working clock period of the processors is set from 5.28 ns to 4.76 ns in decrements of 0.04 ns. For each implementation, we introduce speculative FFs and error correction mechanisms. Delay elements are also inserted appropriately. The smallest working clock period, it is 4.76 ns, is 10.8% smaller than the minimum working clock period of the G_{org} .

We compare the application performance of G_{org} and the 14 implementations. Fig. 10 shows the average effective clock period and the average timing-error rate for the six applications. Y-axes are the effective clock period and the timing-error rate. The results are average value for six applications. The result shows that the error rate increases by decreasing the working clock period. The effective clock period is saturated around 5.00 ns. The implementation with working clock period 4.88 ns achieves highest performance and its effective clock period is 5.00 ns. Comparing to the minimum working clock period of G_{org} , it achieves 6.7% performance improvement.

V. CONCLUSION

In this paper, we expanded general-synchronous circuit to improve the performance of circuits. By allowing a timing-error occurrence in general-synchronous circuit, circuits enables to work with a smaller clock period than the limited clock period. We used the dynamic-error detection/correction method for variable latency technique. From the evaluation using a 5-stage pipelined processor, we confirmed that the effective performance of our proposal improvement 6.7% in compared to the performance of a general-synchronous circuit without using variable latency technique.

REFERENCES

- [1] E.G. Friedman. Clock distribution networks in synchronous digital integrated circuits. *Proceedings of the IEEE*, 89(5):665–692, 2001.
- [2] A. Takahashi and Y. Kajitani. Performance and reliability driven clock scheduling of sequential logic circuits. In *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC '97*, pages 37–42, 1997.
- [3] Tomoyuki Yoda and Atsushi Takahashi. Clock period minimization of semi-synchronous circuits by gate-level delay insertion. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 82(11):2383–2389, 1999.
- [4] Yukihide Kohira and Atsushi Takahashi. Clock period minimization method of semi-synchronous circuits by delay insertion. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 88(4):892–898, 2005.

- [5] Atsushi Takahashi. Practical fast clock-schedule design algorithms. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, 89(4):1005–1011, 2006.
- [6] D. Ernst, Nam Sung Kim, S. Das, S. Pant, R. Rao, T. Pham, C. Ziesler, D. Blaauw, T. Austin, K. Flautner, and T. Mudge. Razor: a low-power pipeline based on circuit-level timing speculation. In *Proceedings of the 36th Annual IEEE/ACM International Symposium on Microarchitecture, MICRO-36*, pages 7–18, 2003.
- [7] T. Sato and Y. Kunitake. A simple flip-flop circuit for typical-case designs for DFM. In *Proceedings of the 8th International Symposium on Quality Electronic Design, ISQED '07*, pages 539–544, 2007.
- [8] D. Bull, S. Das, K. Shivashankar, G.S. Dasika, K. Flautner, and D. Blaauw. A power-efficient 32 bit ARM processor using timing-error detection and correction for transient-error tolerance and adaptation to PVT variation. *IEEE Journal of Solid-State Circuits*, 46(1):18–31, 2011.
- [9] Kenta Ando and Atsushi Takahashi. *SASIMI 2012 Proceedings*, pages 549–554, 2012.
- [10] J. P. Fishburn. Clock skew optimization. *IEEE Transactions on Computers*, 39(7):945–951, Jul 1990.
- [11] R. B. Deokar and S. S. Sapatnekar. A graph-theoretic approach to clock skew optimization. In *Proceedings of IEEE International Symposium on Circuits and Systems - ISCAS '94*, volume 1, pages 407–410 vol.1, May 1994.
- [12] John Hennessy and Peter Nye. Stanford integer benchmark suite. Stanford University, 1988.