# Gain Based Delay Balancing in the Deep Submicron Era

Ryusuke EGAWA[1] , Jubee TADA[2], Hiroaki KOBAYASHI[1] and Gensuke GOTO[2]
[1] Cyberscience Center, Tohoku University
6-3 Aramaki Aza Aoba, Aoba-ku, Sendai 980-8578, Japan
[2] Graduate School of Science and Engineering, Yamagata University
4-3-16 Jonan, Yonezawa 992-8510, Japan
E-mail: [1]{egawa, koba}@isc.tohoku.ac.jp, [2]{jubee, gengoto}@yz.yamagata-u.ac.jp

**Abstract:** This paper presents a novel technique to balance the delays in a combinational logic circuit for an equal delay circuit design. The delay balancing technique relies on a gain based delay model and the availability of a continuous size delay elements library. Based on the concept of logical effort in very large scale integrated circuits in the deep submicron era, our proposed technique attempts to minimize a delay difference of combinational logic circuits. Delay balancing tools based on our proposal are developed, and the effectiveness of tools is evaluated. Experimental results show that our proposal achieves 9.96% delay variations on average in combinational logic circuits.

## 1. Introduction

An equal delay circuit has strong requirements for asynchronous circuit design and wave pipelined circuit design. Also, a combinational logic circuit with less delay variation is expected to avoid hold time restrictions of output registers in the deep submicron era. However, the equal delay circuit design, also known as "Delay Balancing" is a tough task because this process has to target and equalize all path's delays in a combinational circuit. The difficulty in the delay balancing is one of the main reasons that asynchronous and wave pipelines cannot be a dominant in recent microprocessor design, in spite of their high potentials. So far, many researchers have proposed some delay balancing techniques [2][9]. However these proposals are not suits for recent advanced CMOS technologies due to enlarging delay difference between logical design and physical design phases [3]. In addition, recent logic synthesis tools are employing heuristic approaches with low accuracy and a large number of feedbacks due to uncertainness of a wire load delay model.

Under this situation, to realize accurate delay balancing in the deep submicron era, we have proposed a delay balancing technique that employs a delay elements insertion and a gate sizing based on an effort delay model [6]. The gain based delay model can estimate delay of a circuit easily with high accuracy in advanced CMOS technologies [1][8]. Also, this model can control the delay of combinational logic circuit by sizing transistors in a deterministic way to optimize the delay of paths in the circuit. As shown in the previous studies concerned with the delay balancing, a combination of delay elements insertion and gate sizing is a common approach. However, there are few approaches for the equal delay circuit designs with gain based delay model [9]. In this paper, we develop delay balancing tools based on our proposal and demonstrate the effectiveness of our approach. Also, we evaluate the delay balanced circuits as wave-pipelined circuits compared with conventional pipelined circuits. These results show the validity of our proposal and high-speed and low-power features of wave pipelines.

In the next section, an overview and a basic concept of proposed delay balancing technique are described. Experimental results are shown in Section 3, and Section 4 concludes this paper.

## 2. Concept of Gain based Delay Balancing

### 2. 1 Outline of Delay Balacing

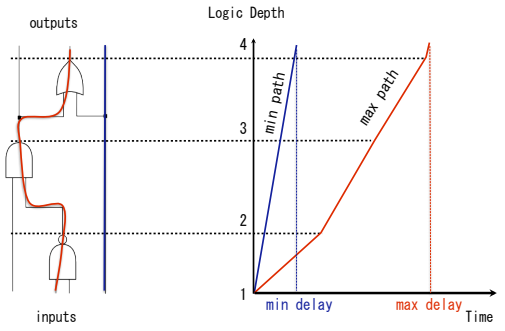
In this section, we describe an outline and a basic concept of our proposed delay balancing. As a common delay balancing approach, we employ two phase delay balancing: rough tuning and fine tuning [2]. Delay elements insertion is used for rough tuning and a gate sizing based on effort delay model is used for a fine tuning. Figure 1 shows a basic scenario of our delay balancing. In our approach, as the first step, logic depth of all paths in a combinational circuit is equalized by delay element insertions, and the circuit is reconstructed to reduce the number of fanouts. After the rough tunings, the process moved to the fine tuning phase. All path's delay are estimated by gain based delay model, and gate widths of delay elements are resized. The left side of each figure shows a combinational logic circuit for delay balancing, and the right side of each figure shows *contour-model* (time-space diagram) of signal propagations on the circuit [2]. The vertical axis of the diagram indicates logic depth and horizontal axis indicates time. The blue line denotes minimum delay path (min-path) in a circuit and red line denotes maximum delay path (max-path) of the circuit. As shown in this figure, carrying out above processes, the minimum path delay and the maximum path delay approach gradually.

### 2. 2 Delay Elements Insertion

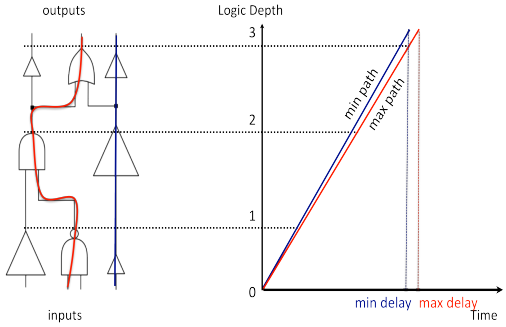
In the delay elements insertion phase, according to our previous studies, we concentrate on adjusting the logic depth of all paths in a combinational logic circuit [6]. The delay is not taken into account in this phase, thus delays of each path and delay elements are not considered. In other words, delay elements are handled as dummy buffers. Then, the proposed technique reconstructs the circuit to reduce a number of fanouts to realize flexible balancing in the next step according to Klass's approach [3]. Figure 2 shows

delay elements insertion, and the first process carries out as follows:

*Step 1. Align logic gates according to the logic depth*
*Step 2. Find out traverse nets*
*Step 3. Insert delay elements on the traverse nets to adjust a logic depth*
*Step 4. Reduce number of fanouts by logic reconstruction.*




(a) Before delay balancing.




(b) After delay balancing.

Figure 1. Delay balancing scenario.

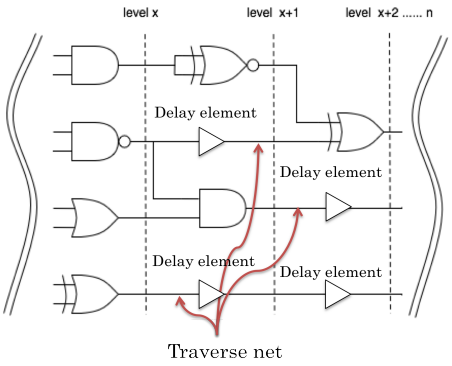



Figure 2. Delay element insertion.

### 2. 3 Gain based Delay Balancing

After logic depth adjustment of the circuit and fanouts reduction, the process moves to a gate sizing phase. Note that our approach is just sizing delay elements, and the delay element is consists of two sequential inverters. In this gate sizing procedure, a gain based delay model that is a delay model used in Logical Effort Theory (LE)[1] is used. The gain based delay model has higher accuracy to estimate delay time in advanced CMOS technologies than a wire load model being used most now [7]. LE refers to the inherent cost of computations in logic gates, and is a characterization of the complexity of a logic gate. It provides a method to estimate delay of a CMOS circuit, and develop a scheme for sizing the transistors by back of the envelope calculations. The main features of LE are briefly described in the following paragraphs. The detail of LE can be obtained from [1].

LE takes into account the fact that the speed of a digital circuit block is dependent on its fanout ($C_{out}$) and its fanin ($C_{in}$). Further, LE introduces technology independence by normalizing the speed to that of a minimal size inverter:

$$d_{abs} = d\tau \quad (1).$$

Where, $d_{abs}$= absolute delay, $d$ = unit less delay, $\tau$ = delay of an inverter driving an identical inverter with no parasitic. The delay expression of a logic block in LE is given as:

$$d = f + p \quad (2).$$

Where $p$ = parasitic delay, $f$ = effort or stage delay. Furthermore $f = gh$ where $g$ is defined as a logical effort and has an electrical effort. The electrical effort $h$ is equal to the ratio of input capacitance and output capacitance : $h = C_{in}/C_{out}$. Thus:

$$d = gh + p \quad (3).$$

Based on this delay model, Figure 3 shows the basic concept of gate sizing to balance a delay. In Figure 3, $C_{in}$ denotes the input capacitance of each input pin, $D_Q$ and $D_R$ indicate the delay at Q and R, a number above each transistor indicates a width of the transistor. In this case, we try to balance $D_Q$ and $D_R$. Here, $D_Q$ = 9 and $D_R$ = 6, thus a Delay Difference ($DD$) is 3. To balance the delays, our proposal sizes an inverter x2 of the delay element based on above equations. If the size of two transistors in the x2 set to five times large compared with initial state, $C_{in}$ of the inverter x1 changes to 15. Then, according to Equation (3), the delay of x1 is changed to 6, and the delay of x2 is changed to 1.4. Eventually, $D_R$ becomes 8.4, and we can obtain DD = 0.6.

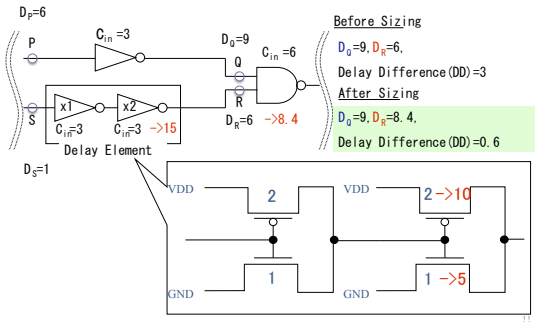



Figure 3. Basic concept of gain based balancing.

To perform gain based delay balancing efficiently, we prepare two options; *redundant delay elements deletion* and *delay element insertion*. A long chain of delay elements sometime enlarges a circuit scale, thus in this phase, our approach detects long delay elements chains and deletes redundant delay elements. Also, if the delay difference is larger than a delay that can be realized by gate sizing, an additional delay element is inserted. Gate sizing procedure

is shown in Figure 4. The delay balancing is performed from input to output of a circuit with checking output capacitance of the next gates. This process is repeated till tracing all paths. In this figure, $n_{max}$ and $n_{min}$ indicate the maximum and the minimum delay of delay elements that can be achieved by the gate sizing. Also this figure assumes $D_Q > D_R$. When the $DD$ ($D_Q$ -$D_R$) is larger than $n_{max}$, the delay element is inserted. On the other hand, if the $DD$ is smaller than $n_{min}$, the delay element is deleted. In the other case, we carry out gain base gate sizing described in the previous paragraph.

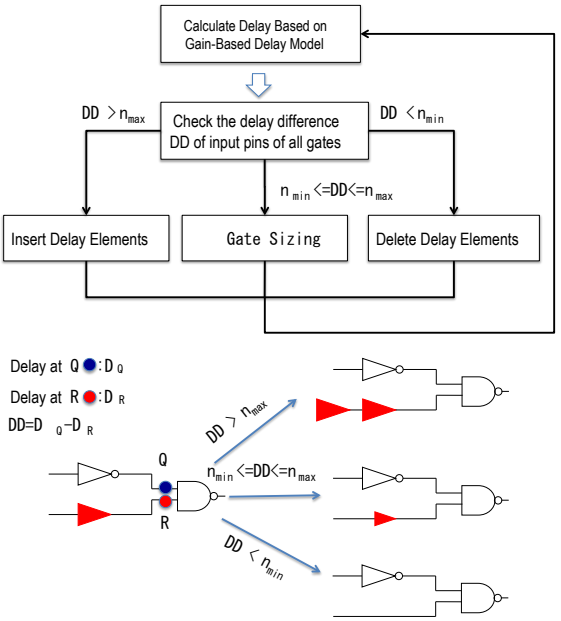



Figure 4. Gate sizing and delay elements control.

## 3. Experimental Results

In this section, our proposal is applied to some adders to confirm the effectiveness.

### 3. 1 Design and Evaluation Flow

Figure 5 shows an evaluation flow and tools that we have implemented. Shaded boxes indicate the tools developed by perl languages based on our proposal. First, combinational logic circuits are designed by VHDL, and logic synthesis is carried out by *Synopsys Design Vision* to obtain a netlist of the circuit and path information. Then these informations are input to a *Logic depth equalizer* that work as shown in Section2.2. In this tool, two inverters with the smallest size are used as the delay element. Note that, so far, delays are not considered. Next, delays are calculated using a gain based delay model by a *Gain-based delay analyzer*, and gate sizing described in Section 2.3 are performed by a *Gain-based gate sizing tool*. From these processes, we get a netlist of the balanced circuit. With this balanced netlist, cell information is input to a *VHDL-SPICE Converter* to obtain a balanced SPICE netlist. Finally, we evaluate the circuit using a *Synopsys Nanosim* simulator (SPICE compatible simulator) in terms of a delay and power of the circuits.

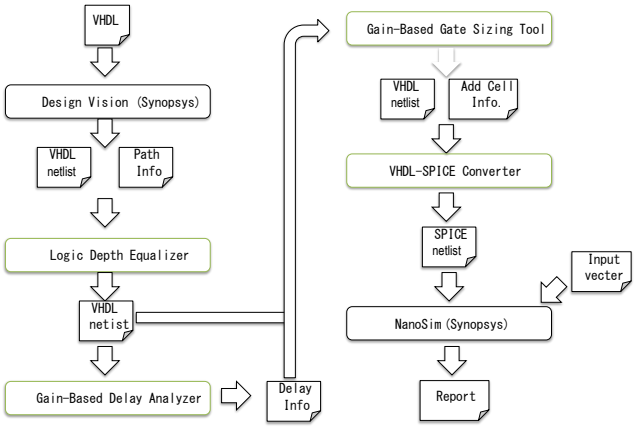



Figure 5. Design and evaluation flow.

### 3. 2 Experimental Setup

16-bit, 32-bit Ripple Carry Adders (RCA16, RCA32) and 16-bit, 32-bit Carry Look ahead Adders (CLA16, CLA32) are designed with Rohm 0.18um CMOS technology. To realize automatic delay balancing, we prepare 27 continuous cells at 0.27 µm-intervals for gate width of a nmos transistor and 0.33 µm -intervals for gate width of a pmos transistor with keeping the ratio of nmos and pmos, and all gate lengths are 0.18µm. The gate widths are varied from 1.35µm to 8.64µm for nmos, and 1.65µm to 10.56µm for pmos. After the calculation of appropriate gate size of the delay element in a gate sizing phase, our tools select the inverter that has the nearest size with the result. In these experiments the number of fanouts is limited to 2, and 10,000 random input vectors are used. The supply voltage is 1.8V for *Nanosim* simulations.

### 3. 3 Experimental Results

To confirm an effectiveness of our proposal, Figure 6 shows the maximum path delay and delay differences of before and after delay balancing. In the best case, 0.37 ns delay difference for RCA16, and the worst case, 0.97 ns delay difference for CLA 32 are achieved. We can confirm our proposal reduces delay difference well and they have just a 9.96% delay difference compared with original circuits on average. However, the maximum path delays are increased by 12.6% on average, due to the increase in total capacitance of the circuits. Table. 1 shows a number of transistors of each adder and its overhead (increase ratio of transistors). Although RCAs achieve significant delay difference reduction, the number of transistors is increased. However, in the case of CLAs, our proposal realizes almost same delay difference with smaller overheads compared with [3][9]. If we can adopt our gate balancing not only to delay elements (inverters), but also to another kind of logic gates, delay differences can be made smaller with small overheads.

Table.1 Number of transistors and overheads.

| | RCA16 | CLA16 | RCA32 | CLA32 |
|---|---|---|---|---|
| Not Balanced | 804 | 1794 | 1634 | 3166 |
| Balanced | 6516 | 2766 | 25512 | 8198 |
| Overhead (%) | 710.45 | 54.18 | 1461.32 | 158.94 |

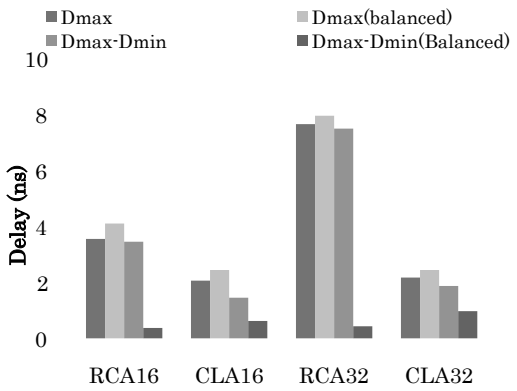



Figure 6. Delay difference of each adder.

Balanced circuits are evaluated as wave pipelined circuits to validate the potential of delay balanced circuits. From delay differences shown in Figure 6 and 0.3ns register's delay of our cell library, clock periods are set 1 ns for RCAs and CLA 16, and 1.5 ns for CLA 32 according to [2]. To make a comparison with conventional pipelined circuits that are designed using *Sysnopsys Design Vision* (balanced registers command) with pipeline degree of 2. The clock cycle periods of conventional pipelines are 2.5ns for RCA16, 4.5ns for RCA 32 and 1.5ns for CLAs respectively. We use random10,000 input vectors for this evaluation, and we confirm wave pipelined behavior using *Nanosim Simulator*. Figure 7 shows power comsumtions of not balanced, conventional pipelined and balanced circuits. Althouh the power consumption of balanced circuits are increased due to delay elements overheads, we can confirm they have smaller power than conventional pipelines with smaller clock periods (higher clock rate). From these results, in more advanced technologies, pipeline registers overheads become larger, and wave pipelines could be consider as a candidate for future low-power and high-speed circuit design techniques.

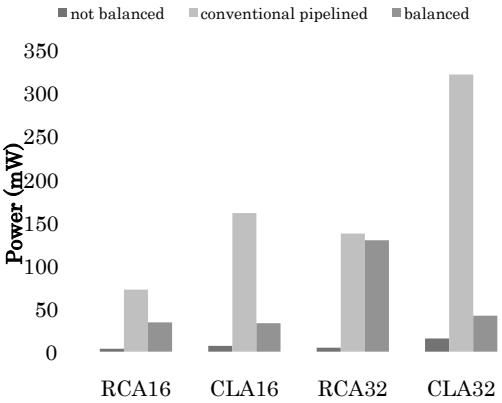



Figure 7. Power consumption.

## 4. Conclusion

In this paper we develop delay balancing tools for equal delay circuit design. Aiming at delay balancing in a deep submicron era, our delay balancing technique employs the gain based delay model. The experimental results show that our tools can realize delay balancing for two kind of adders with 9.96% delay variation on average. Also, considering balanced circuits as wave pipelined circuits, balanced circuits have low-power and high throughput characteristics compared with conventional pipelines. Our future work involves more detailed evaluations, quantitative comparisons with major previous delay balancing techniques [9] and enlarging the number of applicable logic gates.

## Acknowledgement

This research was partially supported by Grant-in-Aid for Young Scientific Research(B), the Ministry of Education, Culture, Sports, Science and Technology, No.19700037. This work is partially supported by VLSI Design and Education Center(VDEC), the University of Tokyo in collaboration with Synopsys, Inc. and ROHM CO. LTD..

## References

[1]. I. Sutherland, B. Sproull, and D. Harris, "Logical Effort: Designing Fast CMOS Circuit," MORGAN KAUFMANN PUBLISHERS, 1997.

[2]. W. P. Burleson, M. Ciesielski, F. Klass, and W. Liu, "Wave-Pipelining: A Tutorial and Research Survey," IEEE Trans. on Very Large Scale Integration (VLSI) Systems, Vol. 6, No. 3, pp. 464-474, 1998.

[3]. F. Klass, M. J. Flynn, "Balancing Circuits for Wave Pipelining," Technical Report CSL-TR-92-549, Stanford University, 1992.

[4]. Vojin G. Oklobdzija, Bart R. Zeydel, Hoang Dao, Sanu Mathew, Ram Krishnamurthy. "Energy-Delay Estimation Technique for High-Performance Microprocessor VLSI Adders," in proc. of 16th IEEE Symposium on Computer Arithmetic (ARITH-16 '03), pp. 272-277, 2003.

[5]. MAGAMA design automation Inc. "Gain based Synthsis: Speeding RTL to Silicon," white paper, http://www.magma-da.com/, 2002.

[6]. J. Tada, R. Egawa, K. Sano, G. Goto and T. Nakamura, "Gain based Delay Balancing Technique for Wave Pipelining", *Proc. of The 22nd International Technical Conference on Circuits/Systems*, (ITC-CSCC 2007), pp. 451-452, 2007.

[7]. H. Q. Dao, V. G. Oklobdzija, "Application of Logical Effort on Delay Analysis of 64-bit Static Carry-Lookahead Adder," *Proc. of 35th Annual Asilomar Conference on Signals, Systems and Computers*, November 4-7, 2001.

[8]. P. Rezvani, and M. Pedram, "Afanout optimization algorithm based on the effort delay model," Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on, , Vol. 22, Issue 12, pp. 1671-1678, 2003.

[9]. R. Tung, Y-B. Kim, "A novel delay balancing methodology for wave pipelined circuits," Proc of 48th Midwest Symposium on Circuit and Systems (MWSCAS), Vol.2 pp.1035 – 1038, 2005.