

## รายการอ้างอิง

1. Lavagno, L. Synthesis and testing of bounded wire delay asynchronous circuits from Signal Transition Graphs. Doctoral dissertation, Department of Electrical Engineering and Computer Science, University of California at Berkeley, 1992.
2. Chu, T. A. Synthesis of self-timed VLSI circuits from graph-theoretic specifications. Doctoral dissertation, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology (MIT), June 1987.
3. Park, S. B. Synthesis of asynchronous VLSI circuits from Signal Transition Graph specifications. Doctoral dissertation, Department of Engineering-Computer Science, Tokyo Institute of Technology (TIT), 1996.
4. Hauck, S. Asynchronous design methodologies: An overview. IEEE Transactions on Computers Vol. 83 No. 1 (January 1995): 69-93.
5. David, I.; Ginosar, R.; and Yoeli, M. An efficient implementation of boolean functions as self-timed circuits. IEEE Transactions on Computers Vol. 41 No. 1 (January 1992): 2-11.
6. Muller, D. E.; Bartky, W. S. A theory of asynchronous circuits. Proceedings on Theory and Switching (1959): 204-243.
7. Seitz, C. L. Introduction to VLSI Systems. New York: Addison Wesley, 1981.
8. Martin, A. J. The limitations to Delay-Insensitivity in asynchronous circuits. Proceedings on Advanced Research in VLSI 6 (1990): 263-278.
9. Takamura, A., et al. TITAC-2: An asynchronous 32-bit microprocessor based on Scalable-Delay-Insensitive model. IEEE Transactions on Computers (Oct 1997): 288-294.
10. Takamura, A., et al. TITAC-2: An asynchronous 32-bit microprocessor. IEEE Transactions on Computers (1998): 319-320.

11. Nanya, T. Asynchronous VLSI System Design [electronic bulletin board]. Research Center of Advanced Science and Technology, University of Tokyo. Available from <http://nanya-www.cs.titech.ac.jp/>; INTERNET. March 1998.
12. Wu, T. Y.; and Vrudhula, S. B. K. A design of a fast and area efficient multi-input Muller C-element. IEEE Transactions on VLSI Systems Vol. 1 No. 2 (June 1993): 215-219.
13. Kuwako, M.; and Nanya, T. Timing-reliability evaluation of asynchronous circuits based on different delay models. Proceedings on Advanced Research in Asynchronous Circuits and Systems 1 (1994): 22-31.
14. Nanya, T.; and Kuwako, M. On signal transition causality for self-timed Implementation of boolean functions. Proceedings of the Hawaii International Conference on System Sciences 26 (1993): 359-368.
15. Akers, S. B. Binary Decision Diagrams. IEEE Transactions on Computers Vol. C-27 No. 6 (June 1978): 509-516.
16. Bryant, R. E. Graph-based algorithms for boolean function manipulation. IEEE Transactions on Computers Vol. C-35 No. 8 (August 1986): 677-691.
17. Nanya, T.; Ueno, Y.; Kagotani, H.; Kuwako, M.; and Takamura, A. TITAC: Design of a Quasi-Delay-Insensitive microprocessor. IEEE Design & Test of Computers Vol. 11 No. 2 (1994): 50-63.
18. Varshavsky, V. I., ed. Self-timed control of concurrent processes. New York: Kluwer Academic Publishers, (n.d.).
19. Kishinevsky, M.; Kondratyev, A.; Taubin, A.; and Varshavsky, V. Concurrent hardware: The theory and practice of self-timed design. New York: Series in Parallel Computing, John Wiley & Sons, 1994.
20. Lisanke, R. Logic synthesis benchmark circuits for the International Workshop on logic synthesis. Available from <ftp://ftp.mcnc.org/>; INTERNET. March 1998.
21. Yang, S. Logic synthesis and optimization benchmarks userguide, version 3.0. Available from <ftp://ftp.mcnc.org/>; INTERNET. March 1998.

22. Poon, R. K. Computer circuits electrical design. New York: Addison Wesley, 1995.
23. Geiger, R.L.; Allen P.E.; and Strader N.R. VLSI: Design techniques for analog and digital circuits. New York: McGRAW-HILL, 1990.

ภาคผนวก

ภาคผนวก ก

การแทนส่วนวงจรรางคู่ที่ออกแบบโดยใช้ตรรกะรางคู่ที่ไรต์ัวผกผัน

รูปแบบ

ส่วนของอินพุตของส่วนวงจรรางคู่

$i$  ค่าความหน่วงสายอินพุต  $i_1$  ของวงจร , ค่าความหน่วงสายอินพุต  $i'_1$  ของวงจร

$i$  ค่าความหน่วงสายอินพุต  $i_2$  ของวงจร , ค่าความหน่วงสายอินพุต  $i'_2$  ของวงจร

⋮

⋮

⋮

ส่วนคู่เกตแอนดออร์ของส่วนวงจรรางคู่

$a$   $Gd_{0 \rightarrow 1}, Gd_{1 \rightarrow 0}$  ( [หมายเลขสายอินพุต : ค่าความหน่วงสาย] , หมายเลขสายเอาต์พุต )

$o$   $Gd_{0 \rightarrow 1}, Gd_{1 \rightarrow 0}$  ( [หมายเลขสายอินพุต : ค่าความหน่วงสาย] , หมายเลขสายเอาต์พุต )

⋮

⋮

⋮

ข้อกำหนด

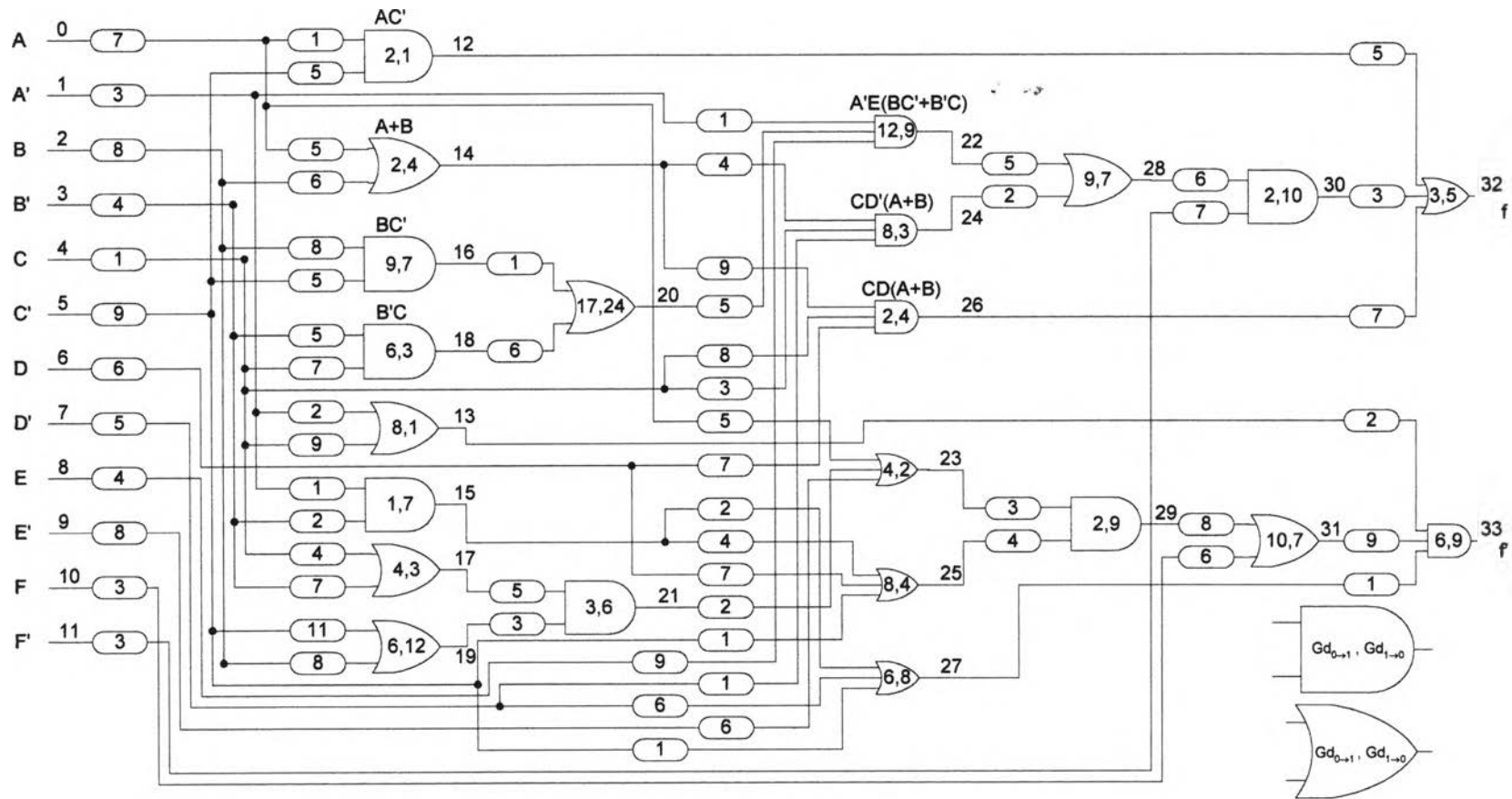
1. คู่สายที่เป็นอินพุตของวงจรคู่ที่  $x$  หรือ  $(i_x, i'_x)$  มีหมายเลขเป็น  $(2x-2, 2x-1)$
2. แต่ละคู่เกตแอนดออร์ในวงจรแทนด้วย
  - $a$   $Gd_{0 \rightarrow 1}, Gd_{1 \rightarrow 0}$  ( [หมายเลขสายอินพุต : ค่าความหน่วงสาย] , หมายเลขสายเอาต์พุต )
  - $o$   $Gd_{0 \rightarrow 1}, Gd_{1 \rightarrow 0}$  ( [หมายเลขสายอินพุต : ค่าความหน่วงสาย] , หมายเลขสายเอาต์พุต )

โดย  $a$  คือ เกตแอนด์

$o$  คือ เกตออร์

$Gd_{0 \rightarrow 1, 1 \rightarrow 0}$  คือ ค่าความหน่วงเกตสำหรับการเปลี่ยนระดับสัญญาณ  $0 \rightarrow 1$  และ  $1 \rightarrow 0$   
ทั้งนี้ หมายเลขสายเอาต์พุตของคู่เกตแอนดออร์จะเป็นเลขคู่กันระหว่าง  $(2x-2, 2x-1)$

ตัวอย่างการแทนส่วนวงจรรางคู่ที่ออกแบบโดยใช้ตรรกะรางคู่ที่ไรต์ัวผกผันของฟังก์ชัน  $f = AC' + CD(A+B) + F'(CD'(A+B) + A'E(BC'+B'C))$  แสดงในรูปแบบที่ ก.1



รูปที่ ก.1 ตัวอย่างการแทนส่วนวงจรรางคู่ที่ออกแบบโดยใช้ตรรกะรางคู่ที่ไรต์วอล์ก

```

i7,3
i8,4
i1,9
i6,5
i4,8
i3,3
a2,1(0:1,5:5,12)
o8,1(1:2,4:9,13)
o2,4(0:5,2:6,14)
a1,7(1:1,3:2,15)
a9,7(2:8,5:5,16)
o4,3(3:4,4:7,17)
a6,3(3:5,4:7,18)
o6,12(2:11,5:8,19)
o17,24(16:1,18:6,20)
a3,6(17:5,19:3,21)
a12,9(1:1,20:5,8:9,22)
o4,2(0:5,21:2,9:6,23)
a8,3(14:4,4:3,7:1,24)
o8,4(15:4,5:1,6:7,25)
a2,4(4:9,14:8,6:7,26)
o6,8(5:6,15:2,7:3,27)
o9,7(22:5,24:2,28)
a2,9(23:3,25:4,29)
a2,10(28:6,11:7,30)
o10,7(29:8,10:6,31)
o3,5(12:5,30:3,26:7,32)
a6,9(13:2,31:9,27:1,33)

```

รูปที่ ก.1 ตัวอย่างการแทนส่วนวงจรรางคู่ที่ออกแบบโดยใช้ตรรกะรางคู่ที่ไรต์แวมกผัน (ต่อ)

ภาคผนวก ข

การแทนส่วนวงจรร่างคู่ที่ออกแบบโดยใช้  
แผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับ

รูปแบบ

ส่วนของอินพุตของส่วนวงจรร่างคู่

i ค่าความหน่วงสายอินพุต  $i_1$  ของวงจรถูก , ค่าความหน่วงสายอินพุต  $i'_1$  ของวงจรถูก

i ค่าความหน่วงสายอินพุต  $i_2$  ของวงจรถูก , ค่าความหน่วงสายอินพุต  $i'_2$  ของวงจรถูก

·  
·  
·

ส่วนเส้นทางต่อเชื่อมเกตแอนดีในส่วนวงจรร่างคู่

a  $Gd_{0 \rightarrow 1}$  ,  $Gd_{1 \rightarrow 0}$  ( [หมายเลขสายอินพุต : ค่าความหน่วงสาย] , หมายเลขสายเอาต์พุต ) ,

$Gd_{0 \rightarrow 1}$  ,  $Gd_{1 \rightarrow 0}$  ( [หมายเลขสายอินพุต : ค่าความหน่วงสาย] , หมายเลขสายเอาต์พุต )

·  
·  
·

ส่วนชุดของเกตออร์ในส่วนวงจรร่างคู่

o  $Gd_{0 \rightarrow 1}$  ,  $Gd_{1 \rightarrow 0}$  ( [หมายเลขสายอินพุต : ค่าความหน่วงสาย] , หมายเลขสายเอาต์พุต )

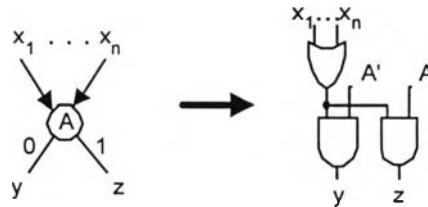
·  
·  
·

e



### ข้อกำหนด

1. คู่สายที่เป็นอินพุตของวงจรรวมที่  $x$  หรือ  $(i_x, i_x')$  มีหมายเลขเป็น  $(2x-2, 2x-1)$
2. สำหรับการแปลงบัพภายในที่เป็นตัวแปรเดียวกันของแผนภาพตัดสินใจแบบทวิภาคชนิดที่มีการลดทอนอันดับเป็นส่วนวงจรรวมคู่ ดังแสดงในรูปที่ ข.1



รูปที่ ข.1 การแปลงแผนภาพตัดสินใจแบบทวิภาคชนิดที่มีการลดทอนอันดับเป็นส่วนวงจรรวมคู่

จะได้ว่า กลุ่มเกตที่แปลงได้จะประกอบด้วย 2 ส่วนคือ

#### ก. ชุดของเกตออร์

แต่ละเกตออร์แทนด้วย

$0 \text{ } Gd_{0 \rightarrow 1}, Gd_{1 \rightarrow 0}$  ( [หมายเลขสายอินพุต : ค่าความหน่วงสาย] , หมายเลขสายเอาต์พุต )

โดย  $0$  คือ เกตออร์

$Gd_{0 \rightarrow 1, 1 \rightarrow 0}$  คือ ค่าความหน่วงเกตสำหรับการเปลี่ยนระดับสัญญาณ  $0 \rightarrow 1$  และ  $1 \rightarrow 0$  ทั้งนี้ หมายเลขสายเอาต์พุตของแต่ละเกตออร์จะเป็นเลขเรียง  $x, x+1, \dots$

และ เมื่อเรียงเกตออร์สำหรับแต่ละบัพภายในที่เป็นตัวแปรเดียวกันจบ ให้ใช้  $e$  แสดงการจบชุดเกตออร์

#### ข. เกตแอนด์ที่เป็นคู่สำหรับแต่ละเกตออร์

คู่ของเกตแอนด์ แทนด้วย

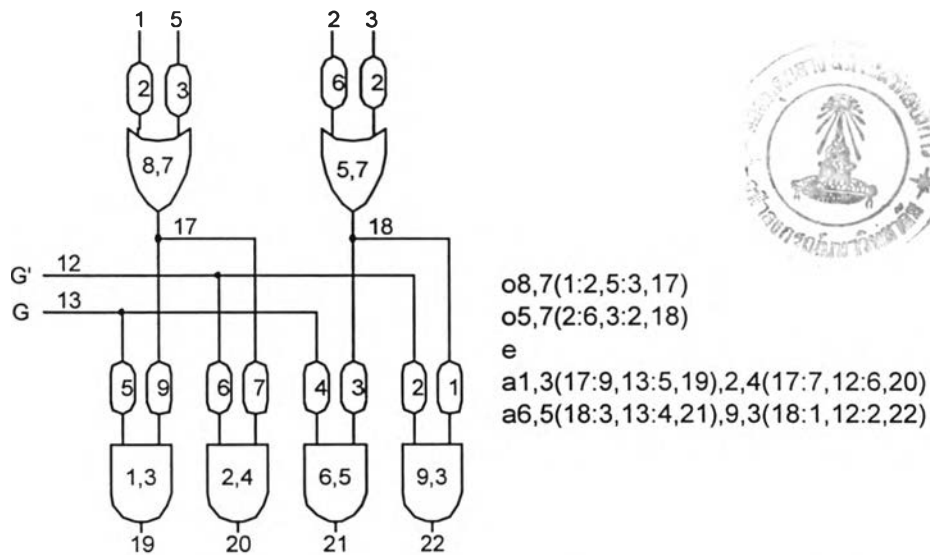
$a \text{ } Gd_{0 \rightarrow 1}, Gd_{1 \rightarrow 0}$  ( [หมายเลขสายอินพุต : ค่าความหน่วงสาย] , หมายเลขสายเอาต์พุต ) ,

$Gd_{0 \rightarrow 1}, Gd_{1 \rightarrow 0}$  ( [หมายเลขสายอินพุต : ค่าความหน่วงสาย] , หมายเลขสายเอาต์พุต )

โดย  $a$  คือ เกตแอนด์

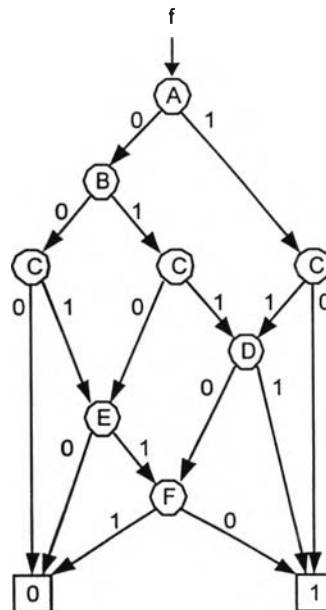
$Gd_{0 \rightarrow 1, 1 \rightarrow 0}$  คือ ค่าความหน่วงเกตสำหรับการเปลี่ยนระดับสัญญาณ  $0 \rightarrow 1$  และ  $1 \rightarrow 0$

ทั้งนี้ หมายเลขสายเอาต์พุตของเกตแอนดที่เป็นคู่จะเป็นตัวเลข  $(x, x+1)$  และบังคับให้สายอินพุตแรกของเกตแอนดที่เป็นคู่ต้องเป็นหมายเลขเดียวกันดังตัวอย่างในรูปที่ ข.2



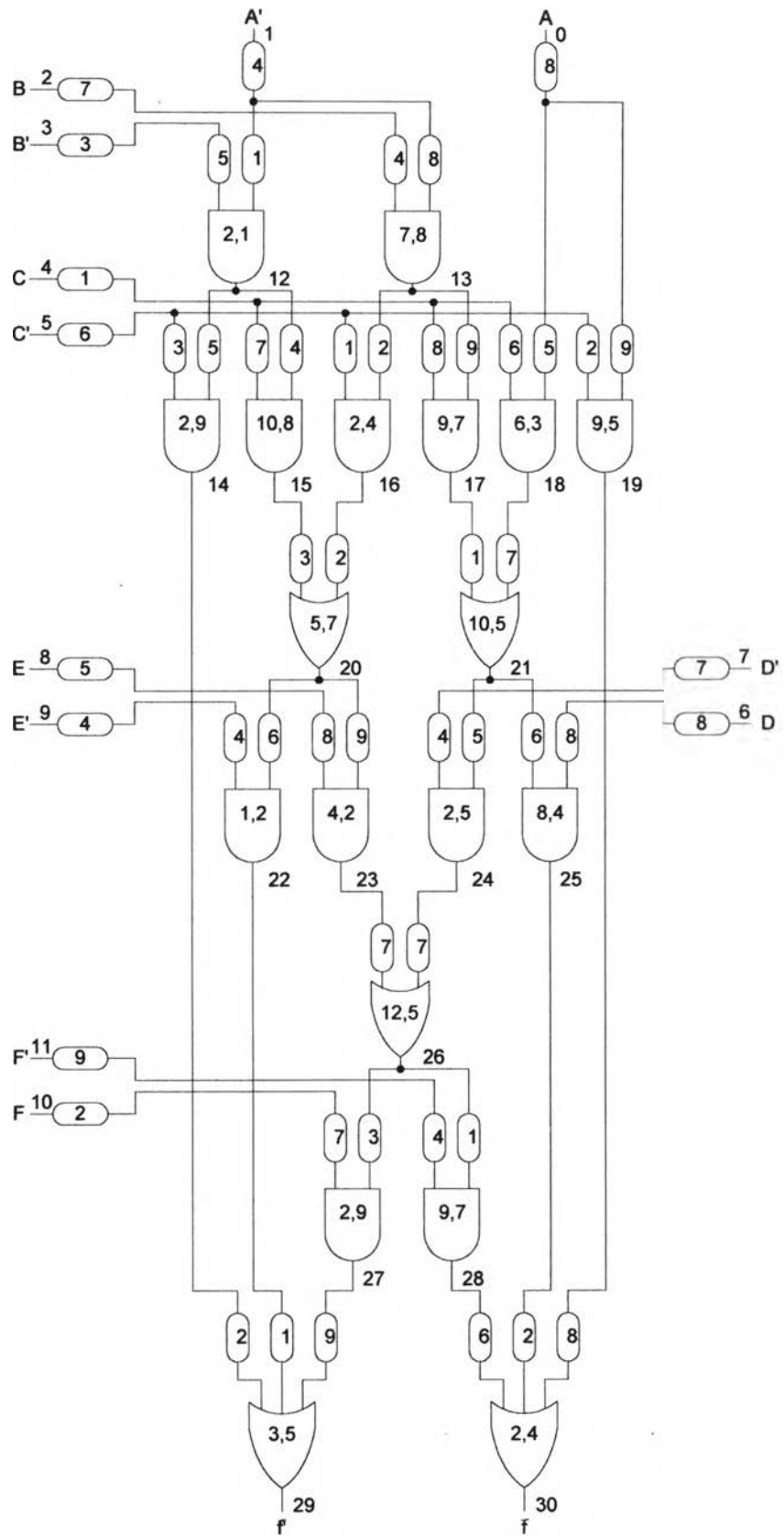
รูปที่ ข.2 ตัวอย่างการแทนชุดของเกตออร์และเกตแอนดคู่ของส่วนวงจรรางคู่ที่ออกแบบโดยใช้แผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับ

ตัวอย่างการแทนส่วนวงจรรางคู่ที่ออกแบบโดยใช้แผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับของฟังก์ชัน  $f = AC' + CD(A+B) + F'(CD'(A+B) + A'E(BC'+B'C))$  แสดงในรูปที่ ข.3



ก. แผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับ

รูปที่ ข.3 ตัวอย่างการแทนส่วนวงจรรางคู่ที่ออกแบบโดยใช้แผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับ



ข. ส่วนวงจรรางคู่

รูปที่ ข.3 ตัวอย่างการแทนส่วนวงจรรางคู่ที่ออกแบบโดยใช้แผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับ (ต่อ)

```

i8,4
i7,3
i1,6
i8,7
i5,4
i2,9
a2,1(1:1,3:5,12),7,8(1:8,2:4,13)
a2,9(12:5,5:3,14),10,8(12:4,4:7,15)
a2,4(13:2,5:1,16),9,7(13:9,4:8,17)
a6,3(0:5,4:6,18),9,5(0:9,5:2,19)
o5,7(15:3,16:2,20)
o10,5(17:1,18:7,21)
e
a1,2(20:6,9:4,22),4,2(20:9,8:8,23)
a2,5(21:4,7:5,24),8,4(21:6,6:8,25)
o12,5(23:7,24:7,26)
e
a2,9(26:3,10:7,27),9,7(26:1,11:4,28)
o3,5(14:2,22:1,27:9,29)
o2,4(28:6,25:2,19:8,30)
e

```

รูปที่ ข.3 ตัวอย่างการแทนส่วนวงจรรางคู่ที่ออกแบบโดยใช้  
แผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับ (ต่อ)

## ภาคผนวก ค

### การแทนส่วนวงจรตอบรับสำหรับการจำลองการทำงาน

#### รูปแบบ

ส่วนแต่ละกลุ่มส่วนประกอบวงจร

[ หมายเลขสายที่ประกั้นการสิ้นสุดการเปลี่ยนระดับสัญญาณ<sub>j</sub>, ]

C [ หมายเลขสายที่ได้รับการประกั้นการสิ้นสุดการเปลี่ยนระดับสัญญาณ<sub>j</sub>, ]

.

.

.

ส่วนเกตออร์ในส่วนวงจรตอบรับ

OR จำนวนสายที่เลือก

[ ค่าความหน่วงเกตออร์ ], [หมายเลขสายอินพุต<sub>k</sub>, ]

.

.

.

#### ข้อกำหนด

1. สำหรับแต่ละกลุ่มส่วนประกอบวงจร

ในกรณีนี้ที่สายหมายเลข  $W_{11}, W_{12}, \dots$  ใช้สำหรับการประกั้นการสิ้นสุดการเปลี่ยนระดับสัญญาณที่สายหมายเลข  $W_{21}, W_{22}, \dots$  แทนด้วย

$W_{11}, W_{12}, \dots$

C  $W_{21}, W_{22}, \dots$

ในการออกแบบโดยใช้ตรรกะวางคูที่ไรต์ตัวผกผัน

กลุ่มส่วนประกอบวงจรหมายถึง แต่ละคู่เกตแอนด์ออร์

โดย  $W_{11}$  เป็น สายเอาต์พุตของเกตออร์สำหรับแต่ละคู่เกตแอนด์ออร์

$W_{12} \dots W_{1m}$  เป็น สายอินพุตของเกตแอนด์สำหรับแต่ละคู่เกตแอนด์ออร์

$W_{21} \dots W_{2n}$  เป็น สายอินพุตของทั้งเกตแอนด์และออร์ในแต่ละคู่เกตแอนด์ออร์

ในการออกแบบโดยใช้แผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับ

กลุ่มส่วนประกอบวงจร หมายถึง ชุดเส้นทางต่อเชื่อมเกตแอนด์สำหรับแต่ละเกตออร์

โดย  $W_{11} \dots W_{1m}$  เป็น สายหลักของเกตออร์

และ สายในกลุ่มเส้นทางต่อเชื่อมเกตแอนด์ที่มีค่าความหน่วงการ

เปลี่ยนระดับสัญญาณ  $1 \rightarrow 0$  ต่ำกว่า

$W_{21} \dots W_{2n}$  เป็น สายทั้งหมดในกลุ่มเส้นทางต่อเชื่อมเกตแอนด์

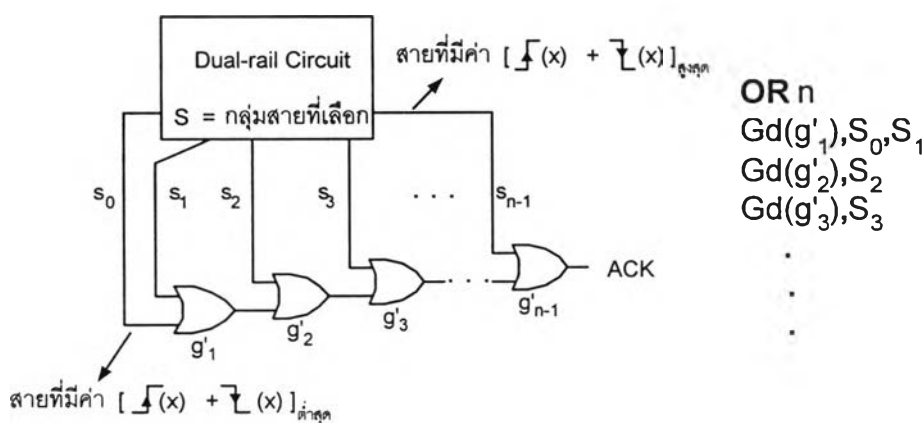
2. สำหรับส่วนเกตออร์ในส่วนวงจรตอบรับ

แทนด้วย

OR จำนวนสายที่เลือก

[ ค่าความหน่วงเกตออร์ ], [หมายเลขสายอินพุต  $x_k$  , ]

ตัวอย่างการแทนส่วนวงจรตอบรับแสดงในรูปที่ ค.1



รูปที่ ค.1 ตัวอย่างการแทนส่วนวงจรตอบรับ

## ภาคผนวก ง

### ผลงานที่ตีพิมพ์จากงานวิจัย

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้ตีพิมพ์เป็นบทความทางวิชาการ ในหัวข้อ "A synthesis of acknowledgement network of Scalable-Delay-Insensitive (SDI) asynchronous combinational circuits" โดย Raschada Nootjarat และ Arthit Thongtak ในงานประชุมวิชาการ "The third Annual National Symposium on Computational Science and Engineering (ANSCSE'99)" ซึ่งจัดโดยสำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติ (NSTDA) ณ. คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในวันที่ 24-26 มีนาคม พ.ศ. 2542

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้ตีพิมพ์เป็นบทความทางวิชาการ ในหัวข้อ "A delay approximation of dual-rail network for Scalable-Delay-Insensitive asynchronous combinational circuits" โดย Raschada Nootjarat และ Arthit Thongtak ในงานประชุมวิชาการ "The National Computer Science and Engineering Conference (NCSEC'99)" ซึ่งจัดโดยคณะวิทยาศาสตร์และเทคโนโลยี มหาวิทยาลัยอัสสัมชัญ ณ. โรงแรมแลนด์มาร์ค ถนนสุขุมวิท ในวันที่ 16-17 ธันวาคม พ.ศ. 2542

# A Synthesis of Acknowledgement Network of Scalable Delay Insensitive (SDI) Asynchronous Combinational Circuits

Raschada Nootjarat

Arthit Thongtak

Department of Computer Engineering, Chulalongkorn University, Bangkok Thailand.

Tel. (662) 218-6956 Fax. (662) 218-6955

[u37rnj@cp.eng.chula.ac.th](mailto:u37rnj@cp.eng.chula.ac.th) , [arthit@cp.eng.chula.ac.th](mailto:arthit@cp.eng.chula.ac.th)

## Abstract

In the near future of VLSI technology, the dominant wire delay continues to be a major setback to utilize high frequency clock in synchronous circuit design. Asynchronous circuit design offers a vital solution to this problem. This paper proposes a mathematical model of delay approximation for the synthesis of asynchronous combinational circuits based on Scalable Delay Insensitive (SDI) model. Using boolean algebra analysis, we proved that the hazard-free acknowledgement network can be implemented by using only OR gate to guarantee all the timing reliability. To estimate the hardware, we compare with Quasi-Delay Insensitive (QDI) circuit. Let  $n$  and  $m$  be the number of primary input and internal wires. In the worst case of our approach, a wire set selected that guarantees the timing reliability contains  $i+g$  wires. This cost is in between the Fundamental Mode (FM) circuit,  $2i$  wires, and Input-Output Mode (IOMode) circuit,  $2i+2g$  wires. From our experimental results, our synthesized circuits operate faster than QDI circuits with the same timing reliability.

## 1. Introduction

Asynchronous circuits have been proven an efficient implementation on the scale-down VLSI technology. When the device size is scaled down, wire-delay increases so that it becomes the dominant term in the layout design. The dominant wire-delay causes many functional restrictions on synchronous design such as global synchronization and clock timing control. In asynchronous design that completely avoids these problems, we achieve ease of modular composition, timing fault tolerance on circuit layout, potentially fast operation with average-case delay, and low power consumption. However, asynchronous circuits are, by definition, sensitive to all signal changes, whether they are intentional (i.e. part of the specification) or not (hazards). Therefore, the variable transitions must be instantaneously verified for correct operation. This results in the design schematic as a *self-timed* manner [1].

*Delay model* [1] plays an essential role in asynchronous circuit design as a layout abstraction. In the *Unbounded Delay Model* such as *Delay Insensitive (DI)*[2] and *Quasi-Delay Insensitive (QDI)*[3], there is no estimation on *actual delay*. The logic design must be painstakingly validated all unlikely transitions. This gives the resulting circuits occasion to tolerate on a harsh environment but suffer from abundant hardware cost. On the other hand, *Bounded Delay Model* and *Speed Independent (SI)*[4] take a gamble on delay estimation. The design is obviously straightforward. The obtained circuits are smaller and faster, but possibly have an error operation.

*Scalable Delay Insensitive (SDI)* [5][6], the novel delay model, is based on *timing reliability* [7]. By means of timing reliability, SDI encompasses the delay models from Bounded Delay Model to QDI. Furthermore, SDI analyzes the relation of *estimated delay* and *actual delay* which substantiates the practical characteristics of signal propagation timing.



Hence, the circuits designed on SDI are the most practical one. Presently, there is no systematic method to synthesize the SDI circuits.

In this paper, we propose a synthesis method for the hazard-free SDI combinational circuits. Based on the functional analysis of *dual-rail network*, the mathematical delay approximations are derived. They are used in both synthesis procedure and performance measurement. Comparing to other combinational circuits that have been proposed, our synthesized circuits operate faster with the minimum hardware cost. This paper is organized as follows. Section 2 and 3 describe an asynchronous combinational circuit and the concept of SDI. Section 4 explicates our synthesis method. Section 5 outlines the performance and hardware cost evaluation. The experimental results and conclusion are stated at the last.

## 2. Asynchronous Combinational Circuit

Hazard-Free asynchronous combinational circuits [8] must use *dual-rail code*, a type of *self-synchronizing code* [9], to encode data path. That is, the logic 0 and 1 are represented as *2-rail codeword* (0,1) and (1,0) in each pair wire ( $x, x'$ ), respectively. The *non-codeword* (0,0) (*spacer*) is used to separate the *operational cycle*. Each *operational cycle* composes of *working phase* and *idle phase* alternately performed that is called *2-rail 2-phase (return-to-zero) operation* [8]. Figure 1 illustrates the asynchronous combinational circuit and figure 2 shows its operation.

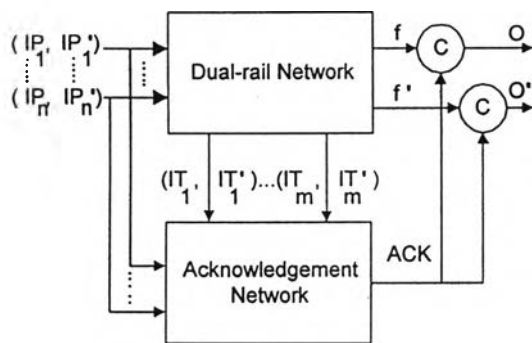


Figure 1. Asynchronous Combinational Circuit

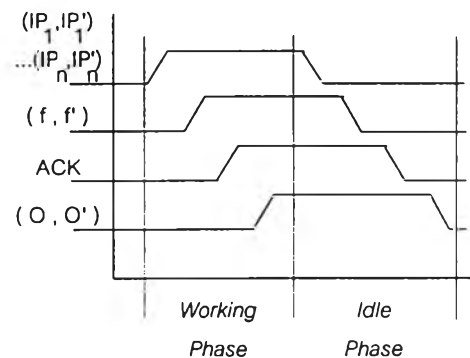


Figure 2. 2-rail 2-phase (return-to-zero) operation

In *working phase*, dual-rail network computes  $(f, f')$  as the output of boolean function corresponding to the applied input pattern. In *idle phase*, all pair wires ( $x, x'$ ) and  $(f, f')$  of dual-rail network will be reset to *spacer*. The circuit that works in this operation has only *rising transition* ( $0 \rightarrow 1$ ) in *working phase* and *falling transition* ( $1 \rightarrow 0$ ) in *idle phase*. This is called *monotonic change property* which is hazard-free equivalence.

In figure 1, the dual-rail network is an *inverter-free 2-rail logic implementation* [8] (see in figure 3) of boolean function. The acknowledgement network generates a *completion signal* (ACK) to indicate the *stability* of dual-rail network. In *working (idle) phase*, the *rising (falling) transition* on ACK is used to propagate the output  $(f, f')$  to  $(o, o')$  through *C-Elements* (CE) [4].

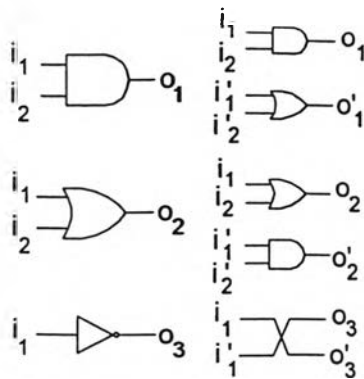


Figure 3. Inverter-free 2-rail logic implementation

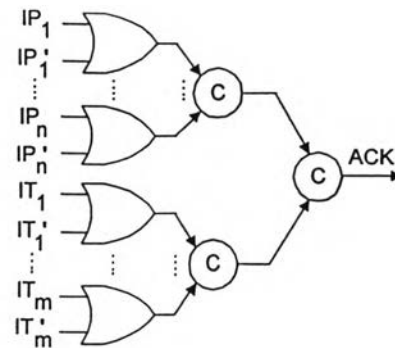


Figure 4. Quasi-Delay Insensitive (QDI) Acknowledgement Network

In order to construct the acknowledgement network, the *environment operation* [1] must be considered. The *environment operation* delineates how the environment should interact with a circuit in order to obtain the correct input-output behavior. In the *fundamental mode (FM) environment* [8][10], the environment changes the inputs and holds them fixed until the circuit has completely stabilized. Only after the circuit has reached a stable state, the environment is allowed to give the next input change. In the *Input-Output mode (IOMode) environment* [11][12][13], the environment does not have to wait until the circuit has stabilized completely. An input change may be made as soon as the network has given an appropriate response to the previous input. Hence, the circuit operated in *IOMode* is more robust than in *FM*. Figure 4 shows a *Quasi-Delay Insensitive (QDI)* acknowledgement network that works in *IOMode*. In *FM*, we can omit the dual-rail network internal wires ( $IT_1, IT_1'$ ), ..., ( $IT_m, IT_m'$ ).

### 3. Scalable Delay Insensitive (SDI) Model

SDI [5][6] is an unbounded delay model with the bounded relative delay ratio. Unlike DI and QDI, SDI imposes a restriction on *relative variation ratio (R)*.  $R$  is the ratio of *Actual Relative Delay ( $D_a$ )* and *Estimated Relative Delay ( $D_e$ )*.  $D_a$  is realized when the circuit is operated. However, it can be estimated as  $D_e$  depending on the layout implementation. Thus,  $R$  represents the estimation error that, in the circuit implementation, is bounded to *Maximum Variation Ratio (K)* as  $1 / K < R < K$ . If wire delays are dominant for the possible delay distribution, a circuit in the smaller area gives lower  $K$ , which in turn makes the logic design easier.

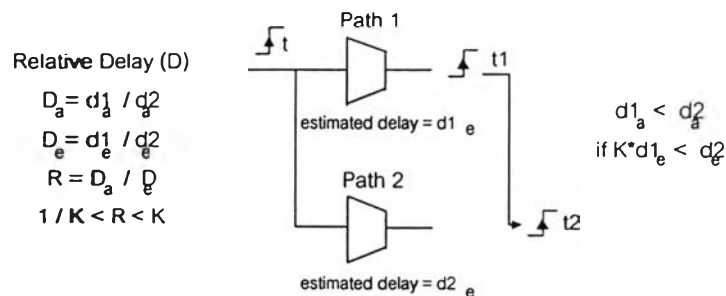


Figure 5. SDI circuit paths, when  $t_1$  occurs before  $t_2$

In the specification of a circuit, if two transitions (e.g.  $t_1$  and  $t_2$ ) have to occur in sequence, based on SDI, the circuit is implemented as two paths operate sequentially as

shown in figure 5. That is, although  $t1$  is  $K$  times slower, the resulting circuit still operates correctly. From this practical viewpoint, the essential  $K$  value can be used to obtain timing reliability.

#### 4. Synthesis of Acknowledgement Network based on SDI model

When analyzing asynchronous combinational circuits working in 2-rail 2-phase (return-to-zero), the completion signal ACK of acknowledgement network is generated once the dual-rail network is stable. After combining this behavior with the implementation of the circuit, the last transition of the dual-rail network is performed as the transition  $t1$ . We can synthesize the acknowledgement network that generates ACK as the transition  $t2$ . In our synthesis procedure, we approximate the delay of each dual AND-OR gate. This delay approximation is analyzed with the monotonic change property in order to compute the dual-rail network delay as  $d1$ . From delay  $d1$ , the delay of acknowledgement network ( $d2$ ) is computed that guarantees maximum variation ratio  $K$  as  $d2 = Kd1$ . Then, the acknowledgement network is constructed by using only OR gate. The input wires of this OR gate are selected from dual-rail network. These wires are proven to cover the transitions in each phase by using the functional analysis. Consequently, the synthesized circuit is hazard-free and can operate in *IOMode*.

##### 4.1 Analysis of dual AND-OR for delay approximation

From *DeMorgan's Theorem*,  $\{f(i_1, i_2, \dots, i_r, 0, 1, +, \bullet)\}' = \{f(i_1', i_2', \dots, i_r', 1, 0, \bullet, +)\}$ , the dual-rail network can be implemented by using only dual AND-OR gates (see in figure 3).

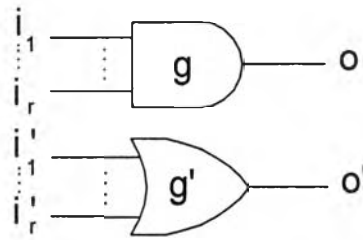


Figure 6. dual AND-OR gate

Figure 6 shows each dual AND-OR gate of dual-rail network and will be used as the notation throughout this paper. In the 2-rail 2-phase operation, each output ( $o, o'$ ) of the dual AND-OR gate will have logic 1(0) in working (idle) phase on these conditions.

Working phase : one of ( $o, o'$ ) has logic "1",

$$\bigwedge_{p=1}^r i_p = 1 \quad \text{if and only if} \quad i_k = 1 \quad \forall (k = 1 \dots r) \quad (1)$$

$$\bigvee_{p=1}^r i_p' = 1 \quad \text{if and only if} \quad i_k' = 1 \quad \exists (k = 1 \dots r) \quad (2)$$

Idle phase : both ( $o, o'$ ) have logic "0",

$$\bigwedge_{p=1}^r i_p = 0 \quad \text{if and only if} \quad i_k = 0 \quad \exists (k = 1 \dots r) \quad (3)$$

$$\bigvee_{p=1}^r i_p' = 0 \quad \text{if and only if} \quad i_k' = 0 \quad \forall (k = 1 \dots r) \quad (4)$$

Equations 1 to 4 are represented as the delay approximation equations. Let  $Set(x)$  and  $Reset(x)$  are the approximated delay of rising and falling transition of wire  $x$ .  $Y_{Set}$  and  $Y_{Reset}$  are the *switching delay* of gate  $Y$  that produces rising (falling) transition on its output.  $Max-Min(a_1, a_2, \dots, a_n)$  are used for finding the maximum-minimum value of  $(a_1, a_2, \dots, a_n)$ .

Working Phase

$$Set(o) = Max( Set(i_1), Set(i_2), \dots, Set(i_r) ) + g_{Set} \quad (5)$$

$$Set(o') = Min ( Set(i_1'), Set(i_2'), \dots, Set(i_r') ) + g'_{Set} \quad (6)$$

Idle Phase

$$Reset(o) = Min ( Reset(i_1), Reset(i_2), \dots, Reset(i_r) ) + g_{Reset} \quad (7)$$

$$Reset(o') = Max( Reset(i_1'), Reset(i_2'), \dots, Reset(i_r') ) + g'_{Reset} \quad (8)$$

#### 4.2 Analysis of monotonic change property for delay computation of dual-rail network

In working phase, the logic value of  $(o, o')$  are formed as two cases below.

Case 1  $(o, o') = (1, 0)$ , after all logic 1(0) of  $i_p(i_p')$  ( $p=1\dots r$ ) had passed  $g(g')$ .

Case 2  $(o, o') = (0, 1)$ , after the first rising transition on  $i_p'$  ( $p=1\dots r$ ) had passed  $g'$ .

Thus, the first rising transition on one output wire of  $(o, o')$  is the correct *2-rail codeword* output. If this output is  $(f, f')$  of dual-rail network (see in fig.1), the acknowledgement network can generate rising transition on ACK to propagate  $(f, f')$  through CE as output of circuit. As a result, despite the lack of stability in working phase, the circuit can produce the correct output and operate faster. However, the unstable rising transitions may exist in idle phase. The appearing of both rising and falling transitions in idle phase causes the circuit to lose monotonic change property and the hazards can occur. To preserve the monotonic change property, the delay of dual-rail network ( $d1$ ) should be the maximum delay of all transitions which is computed as follows. Let  $FSet(x)$  is the minimum approximated delay of rising transition on wire  $x$ .

$$d1 = Max ( Set(a), Reset(a) ) \quad (9)$$

The properties of wire 'a' are :

P1.  $Set(a) > Min( Set(f), Set(f') )$ ,  $a \in$  internal wires of dual-rail network

P2.  $Set(a) > FSet(o')$ ,  $a \in \{ i_1', i_2', \dots, i_r' \} \cup \{ i_1, i_2, \dots, i_r \}$  of each dual AND-OR gate, and

$$FSet(o') = Min ( FSet(i_1'), FSet(i_2'), \dots, FSet(i_r') ) + g' \quad (10)$$

$$FSet(o) = Min ( FSet(i_1), FSet(i_2), \dots, FSet(i_r) ) + g \quad (11)$$

$$FSet(x) = Set(x) \quad x \text{ is the input wire of the dual-rail network.} \quad (12)$$

In property P1, 'a' are internal wires that contain the rising transition while one of  $(f, f')$  has logic 1. In property P2, 'a' are input wires of each dual AND-OR gate that one of its output  $(o, o')$  has logic 1.

### 4.3 Acknowledgement Network implementation

We implement the acknowledgement network by only OR gate. The reasons are stated as follows. Let 'A' refer to all wires in the dual-rail network. We realize that :

$$\text{Working Phase} \quad \exists(A) = 1 \rightarrow \text{Completion Signal} = 1 \quad (13)$$

$$\text{Idle Phase} \quad \forall(A) = 0 \rightarrow \text{Completion Signal} = 0 \quad (14)$$

which is the same behavior as OR gate. Therefore, we can use only OR gate to implement the acknowledgement network, as illustrated in figure 7. Let  $X_1 \dots X_s$  cover all rising and falling transitions in the dual-rail network.

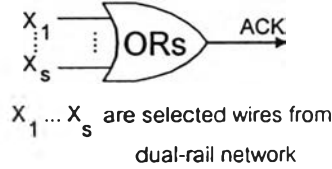


Figure 7. Our synthesized Acknowledgement Netwo

In working phase, ACK can be generated by the first rising transition in  $X_1 \dots X_s$ . In idle phase, all the falling transitions must be verified for preventing the hazards before ACK is produced. To guarantee Maximum Variation Ratio (K) and to assure the stability of the dual-rail network, the delay of acknowledgement network ( $d_2$ ) and 'ORs' are computed as follows.

$$d_2 = K.d_1 \quad (15)$$

$$\text{'ORs' delay} = d_2 - \text{Max}(\text{Reset}(X_1), \text{Reset}(X_2), \dots, \text{Reset}(X_s)) \quad (16)$$

### 4.4 Functionality analysis for selecting dual-rail network wires

From the functionality of dual AND-OR gate described in section 4.1. In working phase, the hazard-free is realized by using the practical delay approximation (see in section 4.2-4.3). To validate the hazard-free property of circuit in idle phase, we consider the behavior of OR gate (analyzed in section 4.3) with its input wires. For this reason, the main objective of wire selection is to cover the falling transitions in idle phase. So we consider only the wires that contains logic 1 in working phase which reset to logic 0 in idle phase. Evidently, these wires can also fulfill the rising transitions in working phase as well. There are two cases of the 2-rail codeword on  $(o, o')$ .

Case 1  $(o, o') = (1, 0)$ . The input wires  $i_p (p=1 \dots r)$  and output wire  $o$  have logic 1. To indicate the ending of falling transitions in all the input and output wires of AND gate, the selection of only one wire that has maximum delay of falling transition is adequate.

Case 2  $(o, o') = (0, 1)$ . Some input wires of both AND and OR gate have logic 1.

From the functionality of OR gate, the ending of all falling transitions in the input wires  $i_p' (p=1 \dots r)$  can be observed from the falling transition at the output wire  $o'$ . For AND gate, all input wires  $i_p (p=1 \dots r)$  that can not be guaranteed by  $o'$  must be selected. These additional wires are the wires that have delay of falling transition more than  $o'$ .

As a conclusion, the conditions for wire selection are :

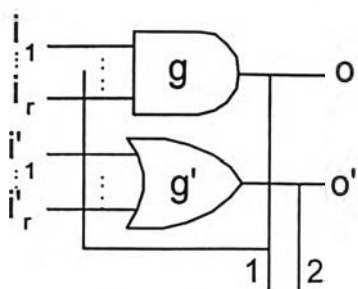


Figure 8. The selection of wires in dual AND-OR gate

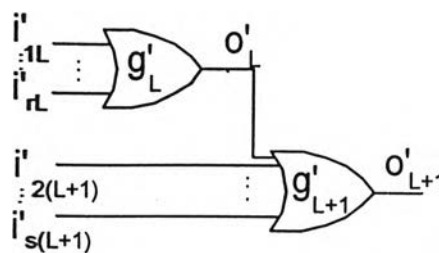


Figure 9. The selection of wires in OR gates connected in series

In figure 8, for each dual AND-OR gate

- Select the wire  $x$  (number 1) when  $\text{Reset}(x) > \text{Reset}(o')$   $x \in \{i_1, i_2, \dots, i_n\} \cup o$   
At least one wire that has the maximum  $\text{Reset}(x)$  must be selected.
- Select the wire  $o'$  (number 2)

In figure 9, for OR gates connected in series, the falling transition on  $O_L'$  can be observed from  $O_{L+1}'$ . Hence,

- $o'$  is not selected when it is between OR gate at level  $L$  and  $L+1$ . But it must be selected if it is input wire of AND gate.

#### 4.5 The hazard-free operation in input-output mode environment

In the synthesis procedure, we analyze the functionality of each dual AND-OR gate in the dual-rail network. From this analysis, both the mathematical approximation of delay and the selection of wires are delineated. It is proved in subsection 4.1 to 4.4 that the acknowledgement network accurately validate the stability of dual-rail network and satisfy maximum variation ratio ( $K$ ). As a result, it can operate in input-output mode environment with hazard-free operation.

### 5. Performance and Hardware Cost Comparison

To compare the acknowledgement network,  $n$ -input OR gate and CE are replaced by their two-input equivalence. Let  $G_n$  and  $Gd_n$  denote  $n$ -input gates and its delay. Since the OR and CE function are associative, we use the scheme [5][8]

$$1G_n = (n-1)G_2 \quad (17)$$

$$Gd_n = Gd_2 \lceil \log_2 n \rceil \quad (18)$$

We measure the performance and hardware cost as follows. Let dual-rail network that has  $2n$  input wires (IP) and  $2m$  internal wires (IT).

#### 5.1 Performance

The performance is estimated from the using time-units in both working phase (WKP) and idle phase (IDP). FMQDI and IOQDI are QDI acknowledgement networks (see in figure 4) operate in FM and IOMode environment. In QDI, OR gate checks the rising (falling) transition of each dual-wire in working (idle) phase, and CE ascertains same logic value of all

OR gate outputs. Table 1 shows the equation of WKP and IDP. **OurSDI** is our synthesized SDI acknowledgement network (see in figure 7) that operates in IOMode. In worst case, the total selected wires from dual-rail network is  $n + m$ .

Table 1. WKP and IDP of each implementation.

Network	WKP	IDP
FMQDI	$\text{Max}(\text{Set}(\text{IP})) + \text{OR}d_2 + \text{CE}d_n$	$\text{Max}(\text{Reset}(\text{IP})) + \text{OR}d_2 + \text{CE}d_n$
IOQDI	$\text{Max}(\text{Set}(\text{IP}), \text{Set}(\text{IT})) + \text{OR}d_2 + \text{CE}d_{n+m}$	$\text{Max}(\text{Reset}(\text{IP}), \text{Reset}(\text{IT})) + \text{OR}d_2 + \text{CE}d_{n+m}$
OurSDI	$\text{Min}(\text{Set}(\text{selected wire})) + \text{OR}d_{n+m}$	$\text{Max}(\text{Reset}(\text{selected wire})) + \text{OR}d_{n+m}$

Stated in [4], the actual  $\text{CE}d_2$  is approximately 1.5 times of  $\text{OR}d_2$ . The equations in table 1 clarify that our acknowledgement network operates faster. This is due to the use of simple OR gates.

## 5.2 Hardware Cost

To measure cost of interface with dual-rail network and acknowledgement network implementation, we calculate both total selected wires (SW) and total internal wires of acknowledgement network (ANW) (see table 2). The worst-case selection of wires (SW) in our method (see figure 10) is that all input wires of AND gate must be selected. The cost is between **FMQDI** and **IOQDI** if  $m > n$ .

Table 2. SW and ANW of each implementation

Network	SW	ANW
FMQDI	$2n$	$4n - 2$
IOQDI	$2n + 2m$	$4n + 4m - 2$
OurSDI (In the worst case)	$n + m$	$2n + 2m - 2$

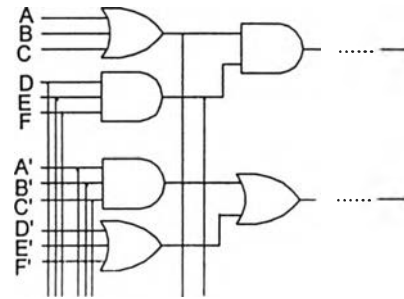


Figure 10. The selection of wires (In the worst case)

## 6. Experimental Results

In order to realize the performance and hardware cost, a 4-bit asynchronous full-adder is synthesized with  $K=2$  [5][6]. For QDI,  $K$  is the maximum variation ratio on the *isochronic fork*. In this circuit, the dual-rail network have 18 input wires and 80 internal wires. Therefore, SW of **FMQDI**, our circuit (**OurSDI**) and **IOQDI** are 18, 38 and 98 wires, respectively. For ANW, **FMQDI**, **OurSDI** and **IOQDI** are 34, 74 and 194. Figure 11 and 12 show WKP and IDP of all input-pattern simulation. Let  $(f, f')$  is output of dual-rail network.

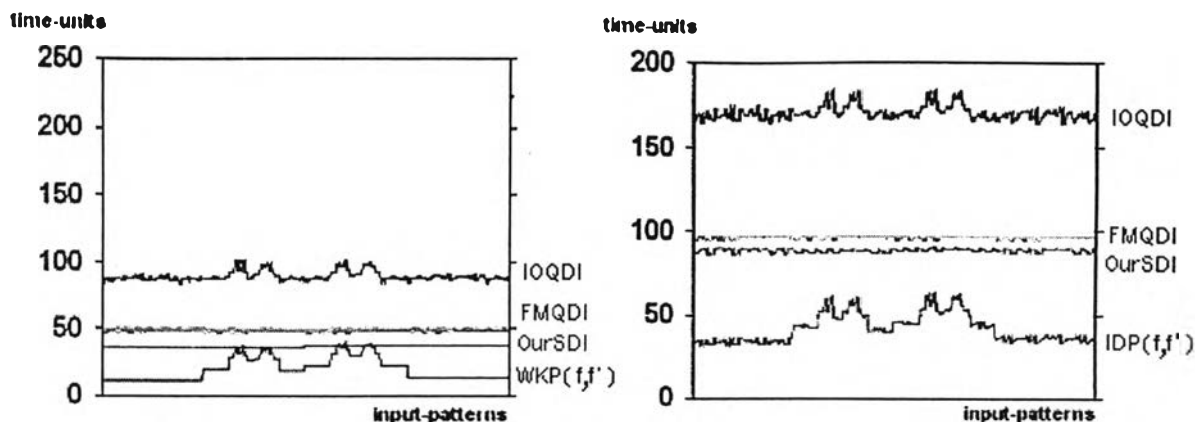


Figure 11. WKP of 4-bit FA ( $K=2.0$ ,  $CE\_delay = 1.5 \times OR\_delay$ ) Figure 12. IDP of 4-bit FA ( $K=2.0$ ,  $CE\_delay = 1.5 \times OR\_delay$ )

From figure 11 and 12, **OurSDI** operates faster than **FMQDI** and **IOQDI** 1.28 and 2.38 times in WKP. In IDP, it operates 1.08 and 1.94 times faster than **FMQDI** and **IOQDI**. Then, the average time-units on WKP ( $WKP_{AV}$ ) and IDP ( $IDP_{AV}$ ) of circuits are computed for comparing **FMQDI** and **IOQDI** with **OurSDI**. Figure 13 and 14 show this ratio on  $K$  variation.

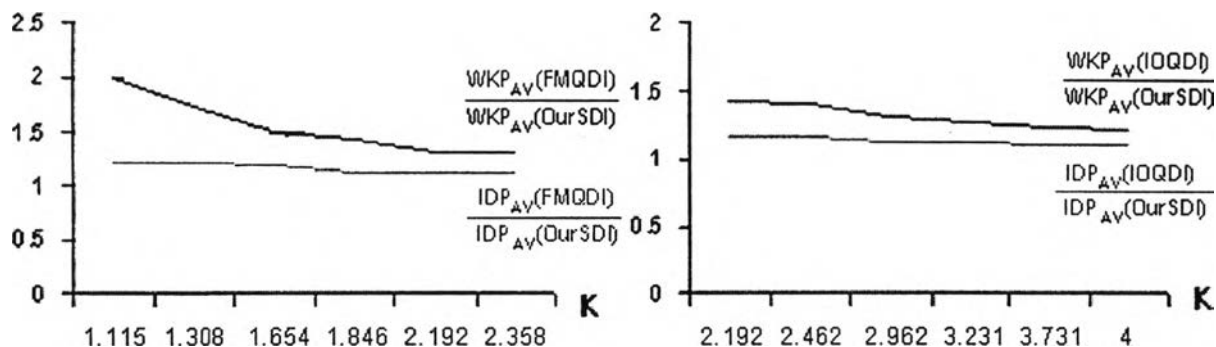


Figure 13.  $WKP_{AV}$  ratio,  $IDP_{AV}$  ratio between **FMQDI** and **OurSDI** Figure 14.  $WKP_{AV}$  ratio,  $IDP_{AV}$  ratio between **IOQDI** and **OurSDI**

Figures 13 and 14 show that the synthesized circuit performs faster than **FMQDI** and **IOQDI** in all variation ratio  $K$ . The timing reliability of **FMQDI** is also less than **IOQDI**.

After 282 circuits including this 4-bit full adder had been tested, we found that our synthesized circuits operate faster than **FMQDI** and **IOQDI** 1.48 and 2.03 times in working phase, 1.15 and 1.56 times in idle phase. As for hardware cost,  $SW(\text{OurSDI})$  is less than  $SW(\text{FMQDI})$  and  $SW(\text{IOQDI})$  1.29 and 3.33 times.  $ANW(\text{OurSDI})$  is less than  $ANW(\text{FMQDI})$  and  $ANW(\text{IOQDI})$  1.30 and 3.35 times.

## 7. Conclusion

This paper proposes the first practical method for synthesis the acknowledgement network of asynchronous combinational circuit based on SDI. The proposed mathematical approximations of delay and functionality analysis are proven to achieve the efficient circuit on every maximum variation ratio ( $K$ ). Furthermore, our hazard-free synthesized circuits can operate in input-output mode environment. Compared to other implementation, our circuits operate faster with less hardware cost.



## 8. Acknowledgement

This research was supported in part by the National Science and Technology Development Agency (NSTDA), contact number CO-B-11-44-09-109M.

## References

- [1] Scott Hauck, "Asynchronous Design Methodologies : An overview", *Procs of the IEEE*, Vol.83, No.1, January 1995, pp.69-93.
- [2] C.L.Seitz, "Introduction to VLSI Systems", Chapter 7 in C.Mead and L.Conway, Addison Wesley, 1981.
- [3] A.J.Martin, "The Limitations to Delay-Insensitivity in Asynchronous Circuits", *Advanced Research in VLSI (Proc.6<sup>th</sup> MIT Conf.)*, 1990, pp.263-278.
- [4] D.E.Muller, W.S.Bartky, "A Theory of Asynchronous Circuits", *Proc.on Theory and Switching*, 1959, pp.204-243.
- [5] Akihiro Takamura, Masashi Kuwako, Masashi Imai, Taro Fujii, Motokazu Ozawa, Izumi Fukasaku, Yoichiro Ueno, Takashi Nanya, "TITAC-2: An asynchronous 32-bit microprocessor based on Scalable-Delay-Insensitive model", *Proc. of ICCD, IEEE 1997*, pp.288-294.
- [6] Akihiro Takamura, Masashi Imai, Motokazu Ozawa, Izumi Fukasaku, Taro Fujii, Masashi Kuwako, Yoichiro Ueno, Takashi Nanya, "TITAC-2: An asynchronous 32-bit microprocessor", *IEEE 1998*, pp.319-320.
- [7] Masashi Kuwako, Takashi Nanya, "Timing-Reliability Evaluation of Asynchronous Circuits Based on Different Delay Models", *Proceedings in 1<sup>st</sup> International Symposium on Advanced Research in Asynchronous Circuits and Systems*, IEEE 1994, pp.22-31.
- [8] Ilana David, Ran Ginosar, Michael Yoeli, "An Efficient Implementation of Boolean Functions as Self-Timed Circuits", *IEEE Trans. on Computers*, January 1992, 41(1) : 2-11.
- [9] Victor I.Varshavsky (Ed.), "Self-Timed Control of Concurrent Processes", Kluwer Academic Publishers, pp.43-113.
- [10] Janusz A.Brzozowski and Jo C.Ebergen, "Recent developments in the design of asynchronous circuits", *Proc. Fundamentals of Computation Theory 1989, Lecture Notes in Computer Science*, August 1989, Vol.380, pp.78-94.
- [11] Michael Kishinevsky, Alex Kondratyev, Alexander Taubin and Victor Varshavsky, "Concurrent Hardware: The Theory and Practice of Self-Timed Design", *Series in Parallel Computing*, John Wiley & Sons, 1994.
- [12] Janusz A.Brzozowski and Jo C.Ebergen, "On the Delay-Sensitivity of Gate Networks", *IEEE Trans. on Computers*, November 1992, 41(11) : 1349-1360.
- [13] Takashi Nanya and Masashi Kuwako, "On Signal Transition Causality for Self-Timed Implementation of Boolean Functions", *System Sciences, Procs. of the 26<sup>th</sup> Hawaii International Conference on Published*, 1993, pp. 359-368.

# A Delay Approximation of Dual-rail Network For Scalable-Delay-Insensitive Asynchronous Combinational Circuits

Raschada Nootjarat

Arthit Thongtak

Department of Computer Engineering, Chulalongkorn University, Bangkok Thailand.

Tel.(662) 218-6956 Fax.(662) 218-6955

Email: [u37rni@cp.eng.chula.ac.th](mailto:u37rni@cp.eng.chula.ac.th) , [arthit@cp.eng.chula.ac.th](mailto:arthit@cp.eng.chula.ac.th)

## Abstract

This paper presents a delay approximation method for implementing the hazard-free circuit on Scalable-Delay-Insensitive (SDI). The method is based on the analysis of signal propagation. With the actual delay associated with each component in a circuit, the method is quite realistic. It enables the approximation of highly complicated dual-rail network and then guarantees the stability to be robust against the delay variation. Shown in the experimental results, the integrity of operation is guaranteed.

## I. Introduction

SCALABLE-DELAY-INSENSITIVE (SDI) [1-3] has become a topic of great interest in asynchronous circuit design [4]. It is motivated by the fact that the complexity on design can be simplified a great deal by considering only a *delay variation*. It is evident that a severe task on design is due to hazards [5][6], informally defined as any deviations of the implementation from the specification. On SDI, the physical reality of the circuit is closely modeled to perform the delay approximation. The delay variation which exploits the estimation error is treated as a specification to make the variable transitions being instantaneously verified for a correct

operation. On layout that the circuit is sensitive to all signal changes, the hazards are removed for ensuring the integrity of operations. Therefore, the *timing reliability* [7] is preserved.

The SDI combinational circuit [8] is divided into two subnetworks: *dual-rail network* and *acknowledgement network*. The dual-rail network is performed for giving the *2-rail codeword* [9][10] output and then it is identified the stability by the acknowledgement network. In the circuit specification that its delay variation is bounded to *maximum delay variation ratio (K)*, the robustness against unpredictable hazards is provided by the acknowledgement network design having K-time delay more than on dual-rail network. The delay used by such an approach is certainly on the stable state, which is only complete when all the states are carefully simulated with the applied delay variation. Since the circuit is not small in general, the simulation of the circuit may not applicable. One approach on this problem is to analyze the characteristic of the transitions on the dual-rail network. This paper proposes a method to approximate the delay for designing the SDI combinational circuit.

In next section, we describe some of the relevant concepts and backgrounds. Section 3 then details the delay approximation. The experimental results and conclusions are contained in section 4 and 5, respectively.

## II. Preliminaries and Backgrounds

This section summarizes some concepts related to Scalable-Delay-Insensitive.

### A. Scalable-Delay-Insensitive (SDI) Model

SDI [1-3] is an *unbounded delay model* with the bounded *relative variation ratio (R)*. The delay variations that give rise to many possible deviations from the specified behavior, are concentrated on the interface between two levels in design. The signal propagation delay on circuit layout is analyzed as *actual relative delay ( $D_a$ )* and it is estimated as *estimated relative delay ( $D_e$ )* for the gate level. R is measured on the estimation error which is bounded between

$1/K$  and  $K$ , while  $K$  is *maximum delay variation ratio*. As shown in figure 1, the implementation of delay for  $t_2$  provides the correct operation. Consequently, the essential  $K$  value means the *timing reliability* [7] used to robust on the delay variations.

### B. Scalable-Delay-Insensitive (SDI) Combinational Circuit

The hazard-free circuit [8] is divided into two subnetworks: dual-rail network and acknowledgement network, as shown in figure 2. The *self-synchronizing property* [9][10] is applied to the circuit by using *dual-rail code* to encode a data path. The function of dual-rail network is to compute the logical output while acknowledgement network verifies the stability of the dual-rail network. Both networks are operated in *working phase* and *idle phase* alternately, which is called *2-rail 2-phase (return-to-zero) operation* [11].

In working phase, every inputs of dual-rail network ( $i, i'$ ) are converted from *spacer*  $(0, 0)$  to *2-rail codeword*  $(0, 1)$  or  $(1, 0)$ , producing the codeword output  $(f, f')$ . The *completion signal* (ACK) is generated once the dual-rail network is stable to propagate the output from  $(f, f')$  to  $(0, 0')$  through *C-element* [12][13]. Then, in idle phase, all wires and  $(f, f')$  of the dual-rail network will be reset to spacer for restarting next cycle. The stability is again assured by ACK.

There are two practical approaches to implement the dual-rail network. In *Inverter-free 2-rail logic implementation* [8][11], we replace each wire such as  $i$  by the pair wires  $(i, i')$ , and use AND-OR gates as shown in figure 3. For *Reduced-Ordered-Binary Decision Diagram (ROBDD) implementation* [8][14], each node in ROBDD is replaced by AND-OR gates as shown in figure 4.

### III. The delay approximation of Dual-rail Network

For ensuring the correct operation, the completion signal ACK produced by the acknowledgement network is used to guarantee the stability of the dual-rail network. The signal must be generated once the dual-rail network becomes stable; inasmuch as, we approximate the delay from the last transition on the dual-rail network and then construct the acknowledgement network. In case of the delay variation bounded to the maximum delay variation ratio (K), the acknowledgement network must be implemented with K-time delay.

One significant point on the circuit design is the complicated in delay approximation. Even though we correctly model the actual delay of gate and wire on layout, the delay approximation of dual-rail network may have an error caused from a plenty of tangled paths. In this research, we calculate the delay approximation of dual-rail network designed on Inverter-free 2-rail logic implementation and Reduced-Ordered-Binary Decision Diagram (ROBDD) implementation. Both implementations have only 'AND' and 'OR' gates, so the procedure of delay approximations are equivalent.

#### A. The delay approximation of transitions on dual-rail network

Definitions:

$Gd$  = the actual gate switching delay

$Wd$  = the actual wire delay

$\uparrow(x)$  = the delay of the **rising** transition at node x

$\downarrow(x)$  = the delay of the **falling** transition at node x

From the functionality of each gate, we approximate the transition delay as follows

For AND gate

$$\uparrow(\text{output}) = \uparrow_{\max}(\text{input}) + Gd_{\text{set}} \quad (1)$$

$$\downarrow(\text{output}) = \downarrow_{\min}(\text{input}) + Gd_{\text{reset}} \quad (2)$$

For OR gate

$$\uparrow(\text{output}) = \uparrow_{\min}(\text{input}) + Gd_{\text{set}} \quad (3)$$

$$\downarrow(\text{output}) = \downarrow_{\max}(\text{input}) + Gd_{\text{reset}} \quad (4)$$

When the transition is propagated on a wire, its delay at the terminal node is added by  $Wd$  from a delay at the starting node.

### B. The delay approximation of dual-rail network

After calculate the delay of transitions propagated from inputs to output ( $f, f'$ ) as shown in section A, we apparently categorize the network according to the analysis of the delay variation as

#### Case 1: The network without delay variation

We found that the network in working phase gives the correct codeword and then guarantees the stability. However, spacer at the output can not assure the stability in idle phase.

The delay for guarantee the stability is summarized as

$$\text{Delay of working phase} = \uparrow_{\min}(f, f') \quad (5)$$

$$\text{Delay of idle phase} = \downarrow_{\max}(\text{all nodes}) \quad (6)$$

Consequently, the delay of working phase is the minimal path delay of the rising transitions at the output. In idle phase, it is the maximum delay of the falling transitions on the network.

#### Case 2: The network with delay variation

Although the codeword in working phase can guarantee the stability of the operation as the deduction in previous case, but it can not be used when the delay variation occurs between paths in the network. Contemplately, the delay of working phase must be replaced by

$$\text{Delay of working phase} = \uparrow_{\max}(\text{all nodes}) \quad (7)$$

Ultimately, the equation to reckon the delay approximation is

$$\text{Delay of dual-rail network} = (\uparrow, \downarrow)_{\max}(\text{all nodes}) \quad (8)$$

## IV. Experimental Results

To evaluate the delay approximation method, the gate-level circuits are implemented from the MCNC benchmark circuits [13][14]. Table 1 and Table 2 show the experimental results.

Table 1. The delay approximation of dual-rail network without delay variation

Circuit	Inverter-free 2-rail logic implementation				Reduced-Ordered-Binary Decision Diagram (ROBDD)			
	Working Phase		Idle Phase		Working Phase		Idle Phase	
	Approx.	Sim*	Approx.	Sim*	Approx.	Sim*	Approx.	Sim*
Sao2	15.00	16.25	27.25	24.75	7.25	7.25	108.00	<b>108.00</b>
Z9sym	22.00	22.00	75.00	75.00	35.60	35.60	111.80	<b>111.80</b>
9Sym	12.00	12.00	30.00	30.00	48.00	48.00	119.00	<b>119.00</b>
Clip	18.40	18.40	25.20	24.20	75.20	75.20	201.00	<b>201.00</b>
Con1	12.50	12.50	15.50	15.50	20.00	20.00	105.50	<b>105.50</b>
Ex5p	15.40	15.40	38.38	37.50	12.28	12.28	103.15	<b>103.15</b>
Rd84	18.50	18.50	33.25	31.25	32.67	32.67	86.33	<b>80.33</b>
Sqrt8	13.00	13.00	18.25	18.25	23.50	23.50	103.00	<b>103.00</b>

Table 2. The delay approximation of dual-rail network with *maximum delay variation ratio* ( $K$ ) = 2.00\*\*

Circuit	Inverter-free 2-rail logic implementation				Reduced-Ordered-Binary Decision Diagram (ROBDD)			
	Working Phase		Idle Phase		Working Phase		Idle Phase	
	Approx.	Sim*	Approx.	Sim*	Approx.	Sim*	Approx.	Sim*
Sao2	32.75	32.75	58.75	50.50	21.00	21.00	214.00	<b>214.00</b>
Z9sym	44.00	44.00	172.00	172.00	97.00	97.00	211.00	<b>211.00</b>
9Sym	40.00	40.00	53.00	53.00	119.00	119.00	205.00	<b>205.00</b>
Clip	31.60	31.60	54.60	49.40	64.00	64.00	210.00	<b>210.00</b>
Con1	29.50	29.50	34.50	34.50	54.50	54.50	184.00	<b>184.00</b>
Ex5p	31.79	31.79	76.59	76.13	22.06	22.06	189.84	<b>189.84</b>
Rd84	37.25	37.25	67.25	67.00	63.50	63.50	172.50	<b>124.00</b>
Sqrt8	28.33	28.33	40.00	40.00	49.25	49.25	192.50	<b>192.50</b>

\* Sim is the maximum delay from the simulation measuring when the circuit becomes stable.

\*\* K is maximum delay variation ratio on the fabricated circuit using 3 layer metal and 0.5 micron rule CMOS technology [1-3].

For the case without delay variation in table 1; since the codeword at the output of dual-rail network in working phase guarantees the stability, both greater and lower errors have no effect to the circuit operation. However, the circuit wrongly operates in case of the underestimated approximation in idle phase. For table 2; the delay variation causes the codeword can not

guarantee the stability of dual-rail network. So, the approximation is not admissible when it is lower than the correctness.

Shown in table 1, the approximation in working phase of circuit 'Sao2' are lower than the real one; inasmuch as, a trend of the approximation seeks for the minimal path delay. Analyzing these circuits, the path delay is activated when the output  $(f,f')$  of dual-rail network is  $(1,1)$  which will not actually happen.

About the overestimation in both tables, they are caused from the circuits without optimizing on redundant paths. The maximum delay on the approximation equals the selection of the maximum delay when comparing all paths. Therefore, the approximation selects the redundant path that is certainly not operated. After the circuit is optimized, the correct approximation is fulfilled.

## V. Conclusions

This research proposes the methodology for approximating the delay of dual-rail network that is crucial in the acknowledgement network designing on SDI. Consider on the characteristic of dual-rail network operation without delay variation, codeword as the output in working phase can be used to guarantee the stability. Therefore, we approximate the delay in working phase from the minimal path delay of the rising transitions at the output. In idle phase that spacer can not ensure the stability, the delay is approximated from the maximum delay of the falling transitions in the circuit. In case of the network with delay variation, we found that the codeword can not be longer used for guarantee the stability. Instead, the stability in working phase is guaranteed by the maximum delay of the rising transitions. As a result, the delay of dual-rail network is the maximum delay of all transitions in both phases.

We are implementing the delay approximation method based on the analysis of signal propagation which will verify the stability of dual-rail network and that the acknowledgement network correctly satisfies maximum delay variation ratio  $(K)$  based on SDI.



## Acknowledgement

This research was supported by the National Science and Technology Development Agency (NSTDA), contact number CO-B-11-44-09-109M.

## References

- [1] Akihiro Takamura, Masashi Kuwako, Masashi Imai, Taro Fujii, Motokazu Ozawa, Izumi Fukasaku, Yoichiro Ueno and Takashi Nanya. "TITAC-2 : An asynchronous 32-bit microprocessor based on Scalable-Delay-Insensitive model", *Proceedings of ICCD, IEEE*, pp.288-294, Oct 1997.
- [2] Akihiro Takamura, Masashi Imai, Motokazu Ozawa, Izumi Fukasaku, Taro Fujii, Masashi Kuwako, Yoichiro Ueno and Takashi Nanya, "TITAC-2 : An asynchronous 32-bit microprocessor", *IEEE*, pp.319-320, 1998.
- [3] Takashi Nanya, "Asynchronous VLSI System Design", *Research Center of Advanced Science and Technology, University of Tokyo*, <http://nanya-www.cs.titech.ac.jp/>, March 1998.
- [4] Scott Hauck, "Asynchronous Design Methodologies : An overview", *Proc. of the IEEE*, Vol.83, No.1, pp.69-93, January 1995.
- [5] C.L.Seitz, "Introduction to VLSI Systems", *Chapter 7 in C.Mead and L.Conway, Addison Wesley*, 1981.
- [6] A.J.Martin, "The Limitations to Delay-Insensitivity in Asynchronous Circuits", *Advanced Research in VLSI (Proc. 6<sup>th</sup> MIT Conf.)*, pp.263-278, 1990.
- [7] Masashi Kuwako and Takashi Nanya, "Timing-Reliability Evaluation of Asynchronous Circuits Based on Different Delay Models", *Proceedings in First International Symposium on Advanced Research in Asynchronous Circuits and Systems, IEEE*, pp.22-31, 1994.
- [8] Raschada Nootjarat and Arthit Thongtak, "A Synthesis of Acknowledgement Network of Scalable-Delay-Insensitive (SDI) Asynchronous Combinational Circuits", *3<sup>rd</sup> Annual National Symposium on Computational Science and Engineering, Chulalongkorn University*, March, 1999.
- [9] Victor I.Varshavsky (Ed.), "Self-Timed Control of Concurrent Processes", *Kluwer Academic Publishers*, pp.43-113.
- [10] Michael Kishinevsky, Alex Kondratyev, Alexander Taubin, and Victor Varshavsky, "Concurrent Hardware: The Theory and Practice of Self-Timed Design", *Series in Parallel Computing, John Wiley & Sons*, 1994.
- [11] Ilana David, Ran Ginosar and Michael Yoeli, "An Efficient Implementation of Boolean Functions as Self-Timed Circuits", *IEEE Transactions on Computers*, 41(1) : 2-11, January 1992.

- [12] D.E.Muller, W.S.Bartky, "A Theory of Asynchronous Circuits", *Proc. on Theory and Switching*, pp.204-243, 1959.
- [13] Tzyh-Yung Wu and Sarma B.K. Vrudhula, "A design of a fast and area efficient multi-input Muller C-element", *IEEE Transactions on VLSI Systems*, 1(2) : 215-219, June 1993.
- [14] Takashi Nanya, Yoichiro Ueno, Hiroto Kagotani, Masashi Kuwako and Akihiro Takamura, "TITAC : Design of a Quasi-Delay-Insensitive Microprocessor", *IEEE Design & Test of Computers*, Vol.11, No.2, pp.50-63, Summer 1994.
- [15] R.Lisanke, "Logic synthesis benchmark circuits for the International Workshop on Logic Synthesis", [ftp://ftp.mcnc.org/](http://ftp.mcnc.org/), May 1989.
- [16] S.Yang, "Logic synthesis and optimization benchmarks userguide, version 3.0", *MCNC, Research Triangle Park, N.C. 27709*, 1991.

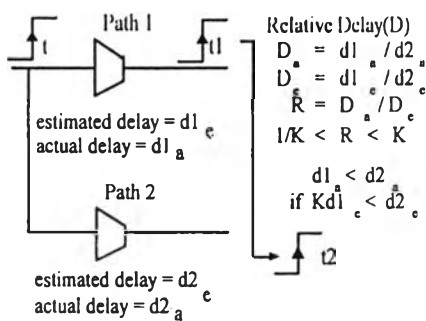


Fig.1 SDI circuit paths

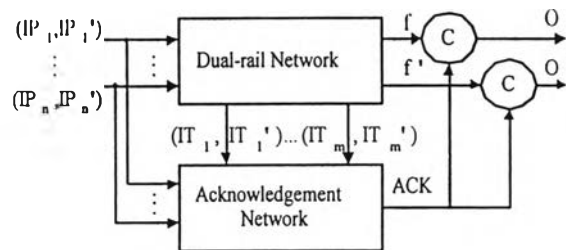


Fig.2 SDI Combinational Circuit

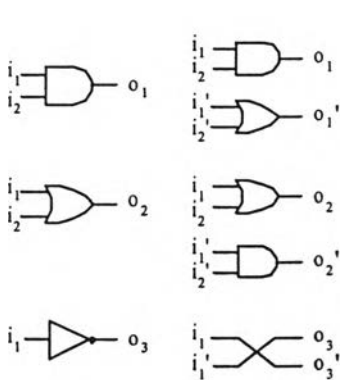
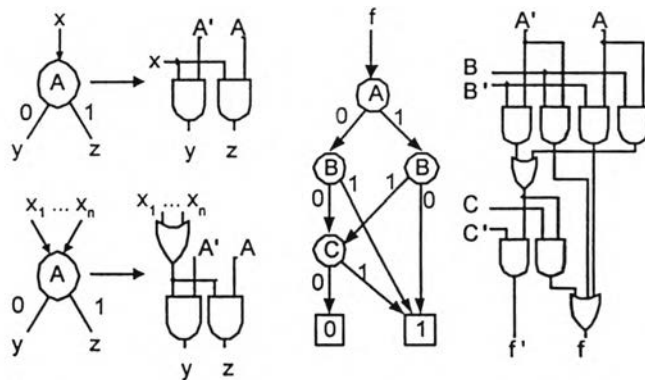


Fig.3 Inverter-free 2-rail logic implementation



(a) Dual-rail Network (b) Implementation example  
 Fig.4 Reduced-Ordered-Binary Decision Diagram (ROBDD) implementation

## ภาคผนวก จ

### ศัพท์เทคนิคที่ใช้ในวิทยานิพนธ์

#### ภาษาไทย

#### ภาษาอังกฤษ

กระบวนการเจือสาร	Fabrication Process
การแกว่งของสัญญาณนาฬิกา	Clock Skew
การเข้ารหัสข้อมูล	Data Encoding
การจำลองการทำงาน	Simulation
การทำงานรางคู่แบบสองขั้นชนิดกลับสู่ศูนย์	2-rail 2-phase (return-to-zero) operation
การเปลี่ยนระดับสัญญาณ	Signal Transition
การเปลี่ยนระดับสัญญาณขึ้น	rising transition
การเปลี่ยนระดับสัญญาณทางเดียว	Monotonic Change
การเปลี่ยนระดับสัญญาณลง	falling transition
การสิ้นสุดการเปลี่ยนระดับสัญญาณ	Stability
การออกแบบโดยใช้ตรรกะรางคู่ที่ไร้ตัวผกผัน	Inverter-free 2-rail logic implementation
การออกแบบโดยใช้แผนภาพตัดสินใจแบบทวิภาค ชนิดมีการลดทอนอันดับ	Reduced-Ordered-Binary Decision Diagram (ROBDD) implementation
การออกแบบวงจรระดับเกต	Gate Level Design
การออกแบบวงจรระดับเลย์เอาท์	Layout Level Design
กิ่งของสาย	Fork
กิ่งของสายเทียบเท่าตลอดช่วง	Isochronic Fork
เกตผกผัน	Inverter gate
เกตแอนด์	AND gate
ขั้น	Phase
ขั้นตอนวิธี	Algorithm

ขั้นทำงาน	Working Phase
ขั้นว่าง	Idle Phase
ความเชื่อถือได้เชิงเวลา	Timing Reliability
ความแปรปรวนความหน่วง	Delay Variation
ความหน่วงสัมพัทธ์	Relative Delay : D
ค่าความหน่วงเกต	Gate Delay
ค่าความหน่วงจริง	Actual Delay
ค่าความหน่วงจริงสัมพัทธ์	Actual Relative Delay : $D_a$
ค่าความหน่วงประมาณ	Estimated Delay
ค่าความหน่วงประมาณสัมพัทธ์	Estimated Relative Delay : $D_e$
ค่าความหน่วงสาย	Wire Delay
ค่าใช้จ่ายวงจร	Hardware Cost
ค่าตรรกะไม่นิยาม	Undefined Value
คุณสมบัติไม่เกิดฮาร์ด	Hazard-Free
ชุดลำดับเดียว	Unique-Successor-Set : USS
ชุดวัดเปรียบเทียบสมรรถนะ MCNC	MCNC Benchmark
ตัวแบ่งรอบการทำงาน	Spacer
เทคโนโลยีด้านวงจรรวมความจุสูงมาก	VLSI Circuit Technology
เทคโนโลยีเลย์เอาท์	Layout Technology
บัพใบ	Leaf Node
บัพภายใน	Internal Node
บัพราก	Root Node
แบบจำลองการทำงานสิ่งแวดล้อม	Environment Operation Model
แบบจำลองความหน่วง	Delay Model
แบบจำลองความหน่วงชนิดมีขอบเขต	Bounded Delay Model
แบบจำลองความหน่วงชนิดไม่มีขอบเขต	Unbounded Delay Model

แบบจำลองความหน่วงแบบไม่ขึ้นต่ออัตราเร็ว	Speed Independent : SI
แบบจำลองความหน่วงแบบไม่ไวต่อความหน่วง	Delay-Insensitive : DI
แบบจำลองความหน่วงแบบไม่ไวต่อความหน่วง ชนิดปรับมาตราส่วนได้	Scalable-Delay-Insensitive : SDI
แบบจำลองความหน่วงแบบไม่ไวต่อความหน่วง ชนิดเสมือน	Quasi-Delay-Insensitive : QDI
ผลคูณ	Product
ผลคูณของผลรวม	Product-of-Sum
ผลรวม	Sum
ผลรวมของผลคูณ	Sum-of-Product
แผนภาพตัดสินใจแบบทวิภาค	Binary Decision Diagram : BDD
แผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอน อันดับ	Reduced-Ordered-Binary Decision Diagram
แผนภาพตัดสินใจแบบทวิภาคชนิดมีอันดับ	Ordered-Binary Decision Diagram : OBDD
พีชคณิตแบบบูล	Boolean Algebra
ฟังก์ชันตรรกะสามค่า	Ternary Function
ภาวะแวดล้อมมูลฐาน	Fundamental Mode Environment : FM
ภาวะแวดล้อมรับเข้าส่งออก	Input-Output Mode Operation : IOMode
รหัสตรงรางคู่	2-rail Codeword
รหัสประสานจังหวะในตัว	Self-Synchronizing Code
รหัสไม่ตรงรางคู่	Non-Codeword
รหัสไม่มีอันดับ	Unordered Code
รหัสรางคู่	Dual-rail Code
รอบการทำงาน	Operational Cycle
แลตช์	Latch
วงจรเชิงผสมแบบอสมวาร	Asynchronous Combinational Circuits
วงจรแบบสมวาร	Synchronous Circuits

วงจรแบบอสมวาร	Asynchronous Circuits
สภาพแวดล้อมการทำงาน	Operating Environment
ส่วนจำเพาะอิสระ	Independent Module
ส่วนประกอบวงจร	Circuit Component
ส่วนวงจรตอบรับ	Acknowledgement Circuit
ส่วนวงจรรางคู่	Dual-rail Circuit
สัญญาณแสดงความบริบูรณ์	Completion Signal
สูตรการกระจายของแซนอน	Shannon Expansion Formula
เส้นเชื่อม, เส้นทางต่อเชื่อม	Path
เส้นทางต่อเชื่อมของส่วนประกอบวงจร	Path of circuit component
เส้นทางต่อเชื่อมเทียบเท่าตลอดช่วง	Isochronic path
เส้นทางส่งผ่านสัญญาณ	Signal Propagation Path
หนึ่งจากทั้งหมด	1-out-of-n
หนึ่งจากสอง	1-out-of-2
อัตราส่วนความแปรปรวนความหน่วงสูงสุด	Maximum Delay Variation Ratio : K
อัตราส่วนความหน่วงสัมพัทธ์	Relative Delay Ratio : R
อินพุต	input
อุปกรณ์ชนิดซี	C-Element
อุปกรณ์หน่วยเก็บ	Storage Element
เอาต์พุต	output
ฮাজারด์	Hazard

## ประวัติผู้เขียน



นางสาวรัชดา นุตจรัส เกิดวันที่ 29 มีนาคม พ.ศ. 2519 ที่จังหวัดกรุงเทพมหานคร สำเร็จการศึกษาระดับปริญญาวิทยาศาสตรบัณฑิต(เกียรตินิยมอันดับสอง) สาขาวิศวกรรมคอมพิวเตอร์ จากภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2540 และศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย ปีการศึกษา 2541