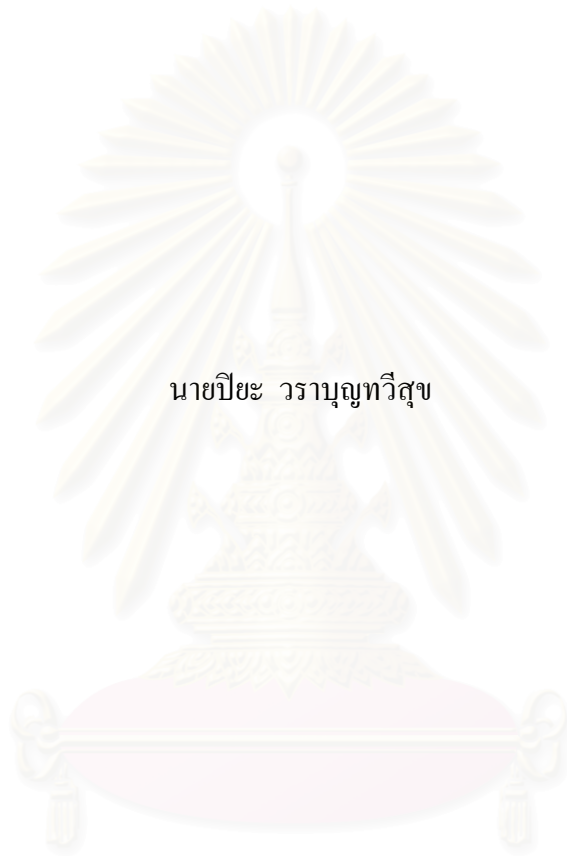


การออกแบบวงจรคูณเลขอิงตรรกะนี้แบบอสมวาร โดยใช้อัลกอริทึมการคูณแบบเชื่อมต่อตรง



นายปิยะ วราบุญทวีสุข

สถาบันวิทยบริการ

จุฬาลงกรณ์มหาวิทยาลัย

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมคอมพิวเตอร์ ภาควิชาวิศวกรรมคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย

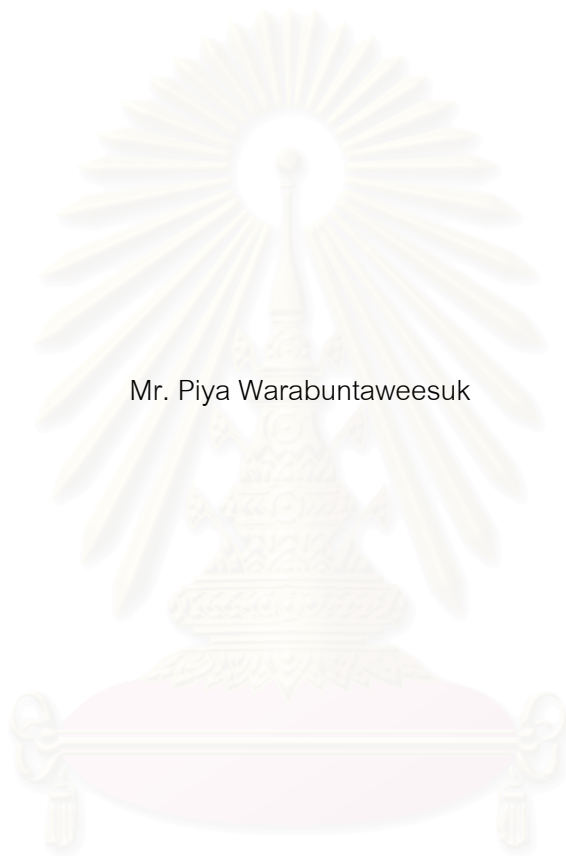
ปีการศึกษา 2546

ISBN 974-17-3611-8

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

DESIGN OF ASYNCHRONOUS FLOATING-POINT MULTIPLIER
USING ON-LINE MULTIPLICATION ALGORITHM

Mr. Piya Warabuntaweasuk



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Engineering in Computer Engineering

Department of Computer Engineering

Faculty of Engineering

Chulalongkorn University

Academic Year 2003

ISBN 974-17-3611-8

หัวข้อวิทยานิพนธ์	การออกแบบวงจรคูณเลขเชิงตรรกะแบบผสมวงโดยใช้อัลกอริทึมการคูณแบบเชื่อมตรง
โดย	นายปิยะ วรบุญทวีสุข
สาขาวิชา	วิศวกรรมคอมพิวเตอร์
อาจารย์ที่ปรึกษา	อาจารย์ ดร. อาทิตย์ ทองทักษ์
อาจารย์ที่ปรึกษาร่วม	อาจารย์ ดร. อรรถสิทธิ์ สุรฤกษ์

คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย อนุมัติให้บัณฑิตวิทยาลัยรับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

..... คณบดีคณะวิศวกรรมศาสตร์
(ศาสตราจารย์ ดร.สมศักดิ์ ปัญญาแก้ว)

คณะกรรมการสอบวิทยานิพนธ์

..... ประธานกรรมการ
(รองศาสตราจารย์ ดร.ประภาส จงสฤษดิ์วัฒนา)

..... อาจารย์ที่ปรึกษา
(อาจารย์ ดร.อาทิตย์ ทองทักษ์)

..... อาจารย์ที่ปรึกษาร่วม
(อาจารย์ ดร.อรรถสิทธิ์ สุรฤกษ์)

..... กรรมการ
(นายชำนาญ ปัญญาใส)

ปิยะ วราบุญทวีสุข : การออกแบบวงจรคูณเลขอิงตรรกะชนิดแบบอสมวารโดยใช้อัลกอริทึมการคูณแบบเชื่อมตรง. (DESIGN OF ASYNCHRONOUS FLOATING-POINT MULTIPLIER USING ON-LINE MULTIPLICATION ALGORITHM) อ. ที่ปรึกษา : อ. ดร.อาทิตย์ ทองทัชย์, อ.ที่ปรึกษาร่วม : อ. ดร. อรรถสิทธิ์ สุรฤกษ์, 51 หน้า. ISBN 974-17-3611-8.

ในระบบเชิงเลขวีแอลเอสไอ ตรรกะเชิงเลขแบบอสมวารเป็นอีกทางเลือกหนึ่งที่เหมาะสมเปรียบเทียบกับตรรกะเชิงเลขแบบสมวาร เพราะตรรกะเชิงเลขแบบอสมวารจะไม่มีปัญหาการแกว่งของสัญญาณนาฬิกา และการดำเนินการต่างๆจะใช้เวลาในการทำงานเป็นแบบเฉลี่ย ปัจจุบันได้มีการนำเสนออัลกอริทึม สถาปัตยกรรม และเทคโนโลยีมากมาย เพื่อพัฒนาการสร้างวงจรมคูณ ซึ่งวิธีการคำนวณจะมีผลกระทบต่อประสิทธิภาพของการสร้างวงจรถูกกัน โดยวิธีการคำนวณหนึ่งที่น่าสนใจ คือ การคำนวณแบบเชื่อมตรง โดยการคำนวณแบบเชื่อมตรงจะทำการแทนตัวเลขที่มีเครื่องหมายกับตัวดำเนินการและผลลัพธ์ของการคำนวณ ซึ่งไหลผ่านหน่วยคำนวณครั้งละหลัก โดยเริ่มต้นจากหลักที่มีนัยสำคัญมากที่สุด

ในวิทยานิพนธ์นี้ได้นำเสนอวิธีการใหม่โดยประยุกต์ใช้อัลกอริทึมการคูณแบบเชื่อมตรงมาออกแบบวงจรมคูณเลขอิงตรรกะชนิดแบบอสมวารขนาด 16 บิต ซึ่งอัลกอริทึมการคูณแบบเชื่อมตรงจะผลิตผลลัพธ์เป็นค่าประมาณที่มีค่ามากกว่าหรือเท่ากับผลลัพธ์ที่แน่นอน โดยไม่ต้องใช้อัลกอริทึมการปิดค่า และใช้ภาษาอธิบายฮาร์ดแวร์วีเอชดีแอลในการอธิบายโครงสร้างของวงจรมคูณ และนำไปสร้างบนเอฟพีจีเอ จากผลการทดลองสามารถสร้างวงจรมคูณแบบอสมวารขนาด 16 บิตมาใช้ได้จริง โดยอัตราปริมาณงานมีค่าเท่ากับ 27.15 เมกกะเฮิร์ต

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาควิชา.....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่อนิสิต.....
สาขาวิชา.....วิศวกรรมคอมพิวเตอร์..... ลายมือชื่ออาจารย์ที่ปรึกษา.....
ปีการศึกษา.....2546..... ลายมือชื่ออาจารย์ที่ปรึกษาร่วม.....

4470414121 : MAJOR COMPUTER ENGINEERING

KEY WORD: ASYNCHRONOUS DESIGN / ASYNCHRONOUS MULTIPLIER / FLOATING-POINT MULTIPLIER / ON-LINE MULTIPLIER

PIYA WARABUNTAWEESUK : DESIGN OF ASYNCHRONOUS FLOATING-POINT MULTIPLIER USING ON-LINE MULTIPLICATION ALGORITHM. THESIS ADVISOR : ARTHIT THONGTAK, Ph.D., THESIS COADVISOR : ATHASIT SURARERKS, Ph.D., 51 pp. ISBN 974-17-3611-8.

Asynchronous digital logic shows to be a valid alternative to its synchronous counterparts in VLSI digital systems, since it is free of clock skew problems. The asynchronous methodology is used to achieve the average case delay operations. Many algorithms, architectures, and technologies have been proposed to improve the implementation of multiplier. Computation mode also has an influence on the effectiveness of the implementation. One interesting mode is an on-line computation where operands and results flow through arithmetic units in a digit serial number starting with the most significant digit, combining with the signed digit number representation.

In this thesis, a novel approach for designing of asynchronous multiplier is proposed particularly in 16-bit floating-point numbers. An on-line algorithm is applied in our approach. The algorithm produces, without rounding algorithm, the estimated result which is equal to or greater than the exact value. VHDL, a hardware description language, is used to structural describe the multiplier implemented by FPGA. The experimental results demonstrate that our 16-bit asynchronous multiplier can be realized with the throughput rate at 27.15 MHz.

Department...Computer Engineering... Student's signature.....

Field of study...Computer Engineering... Advisor's signature.....

Academic year2003..... Co-advisor's signature.....

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี จากความกรุณาอย่างยิ่งของอาจารย์ ดร. อาทิตย์ ทองทักษ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ และอาจารย์ ดร. อรรถสิทธิ์ สุรฤกษ์ อาจารย์ที่ปรึกษาวิทยานิพนธ์ร่วม ซึ่งได้สละเวลาให้แนวคิด และคำแนะนำต่างๆที่เป็นประโยชน์ต่อการทำวิจัยอย่างต่อเนื่อง

ขอขอบคุณ นายกี วัฒนะวิรุณ และนางสาวปัญจภา เรืองสินทรัพย์ ที่ให้คำแนะนำเกี่ยวกับทฤษฎีเบื้องต้นต่างๆ

ขอขอบคุณ ห้องปฏิบัติการ Digital System Engineering Laboratory ที่เอื้อเพื่อสถานที่ และอุปกรณ์ในการทำวิจัย

ขอขอบคุณ เพื่อนนิสิตปริญญาโททุกคน ที่ได้คอยช่วยเหลือการทำวิจัยแก่ผู้วิจัยมาโดยตลอด

ขอขอบคุณ นางสาวภาณภัทร์ แก้วบุญชร ที่คอยให้กำลังใจ ตลอดจนถึงเดือนสติให้ผู้วิจัยกระทำในสิ่งที่ควรกระทำ

ท้ายที่สุด ผู้วิจัยขอกราบขอบพระคุณ บิดา มารดา และพี่สาว ที่สนับสนุน ห่วงใย และให้กำลังใจเสมอมา

ปิยะ วรบุญทวีสุข

27 สิงหาคม 2546

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ง
บทคัดย่อภาษาอังกฤษ	จ
กิตติกรรมประกาศ	ฉ
สารบัญ	ช
สารบัญภาพ	ญ
สารบัญตาราง	ฎ
บทที่	
1 บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของการวิจัย	2
1.3 ขอบเขตของการวิจัย	3
1.4 ประโยชน์ที่ได้รับ	3
1.5 ขั้นตอนดำเนินการวิจัย	3
1.6 ลำดับขั้นตอนในการเสนอผลการวิจัย	4
1.7 ผลงานที่ตีพิมพ์จากงานวิจัย	4
2 แนวคิดและทฤษฎีที่เกี่ยวข้อง	5
2.1 อัลกอริทึมการคูณแบบเชื่อมตรง	5
2.1.1 ระบบตัวเลขซ้ำซ้อนที่มีเครื่องหมาย (Redundant Sign-Digit Number System)	5
2.1.2 อัลกอริทึมการคูณแบบข้อผิดพลาดติดลบ (Negative Error On-line Multiplication Algorithm)	6
2.2 การออกแบบวงจรแบบอสมวารโดยใช้รหัสรางคู่	8
2.2.1 การออกแบบวงจรรางคู่โดยใช้แผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับ (Reduced-Ordered Binary Decision Diagram (ROBDD) Implementation)	10
2.2.2 การออกแบบวงจรตอบรับ	11
2.3 การออกแบบและสร้างวงจรอสมวารโดยใช้เอฟพีจีเอ	11

บทที่	หน้า
2.3.1	12
2.3.2	12
2.3.3	13
2.4	13
2.5	15
สรุป	15
3	16
3.1	16
3.1.1	16
3.1.2	17
3.2	20
3.2.1	20
3.2.2	25
สรุป	27
4	28
4.1	28
4.2	30
4.3	33
4.3.1	33
4.3.2	33
4.3.3	34
สรุป	34
5	35
5.1	35
5.2	36

บทที่

หน้า

รายการอ้างอิง.....	37
ภาคผนวก.....	39
ก ผลการจำลองการทำงานของวงจรคุณสมบัตินแบบเชื่อมต่อตรง	40
ข การคำนวณหาอัตราปริมาณงานของวงจรไปป์ไลน์แบบอสมมาตร.....	48
ค คำแปลศัพท์ที่ใช้ในวิทยานิพนธ์.....	49
ประวัติผู้เขียนวิทยานิพนธ์	51



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญภาพ

ภาพประกอบ	หน้า
2.1 การแสดงค่า 19 บนระบบตัวเลขฐานสองที่มีเครื่องหมาย	5
2.2 โครงสร้างวงจรแบบอสมวาร	9
2.3 ลักษณะการเปลี่ยนระดับสัญญาณของวงจรรางคู่แบบอสมวารในการทำงาน แบบสองขั้วกลับคืนสู่ศูนย์	9
2.4 แผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับสำหรับฟังก์ชัน $F = AB + BC + AC$	10
2.5 การออกแบบส่วนวงจรรางคู่โดยใช้แผนภาพตัดสินใจแบบทวิภาคชนิด มีการลดทอนอันดับ	11
2.6 ขั้นตอนการออกแบบวงจรแบบอสมวารโดยใช้เอฟพีจีเอ	13
2.7 แผนภาพการทำงานของระบบไปป์ไลน์แบบอสมวาร	14
2.8 โปรโตคอลลำดับสัญญาณแบบ 2 ขั้นตอน	14
2.9 โปรโตคอลลำดับสัญญาณแบบ 4 ขั้นตอน	14
2.10 การแปลงตัวเลขห้าขั้วที่มีเครื่องหมายแบบอนเดอะฟลาย	15
3.1 โครงสร้างของวงจรคูณอสมวารแบบเชื่อมตรง	16
3.2 โครงสร้างรุ่นที่ 1 ของวงจรคูณเชื่อมตรง	17
3.3 โครงสร้างรุ่นที่ 2 ของวงจรคูณเชื่อมตรง	18
3.4 แผนภาพตัดสินใจแบบทวิภาคของวงจรเข้ารหัสค่าของผลคูณแบบที่ 1	21
3.5 แผนภาพตัดสินใจแบบทวิภาคของวงจรเข้ารหัสเครื่องหมายของผลคูณแบบที่ 1	22
3.6 วงจรเข้ารหัสค่าของผลคูณแบบที่ 1	22
3.7 วงจรเข้ารหัสค่าเครื่องหมายของผลคูณแบบที่ 1	22
3.8 แผนภาพตัดสินใจแบบทวิภาคของวงจรเข้ารหัสค่าของผลคูณแบบที่ 2	24
3.9 แผนภาพตัดสินใจแบบทวิภาคของวงจรเข้ารหัสเครื่องหมายของผลคูณแบบที่ 2	24
3.10 วงจรเข้ารหัสค่าของผลคูณแบบที่ 2	24
3.11 วงจรเข้ารหัสค่าเครื่องหมายของผลคูณแบบที่ 2	25
3.12 แผนภาพตัดสินใจแบบทวิภาคเพื่อหาค่าสถานะขาออก ของวงจรแปลงอนเดอะฟลาย	26
3.13 แผนภาพตัดสินใจแบบทวิภาคเพื่อหาค่าข้อมูลขาออก ของวงจรแปลงอนเดอะฟลาย	26

สารบัญญภาพ (ต่อ)

ภาพประกอบ	หน้า
3.14 วงจรหาค่าสถานะขาออกของการแปลงตัวเลขแบบอนเดอะฟลาย	26
3.15 วงจรหาค่าข้อมูลขาออกของการแปลงตัวเลขแบบอนเดอะฟลาย	27
4.1 การสังเคราะห์วงจรโดยใช้โปรแกรม Leonardo Spectrum	28
4.2 การสร้างวงจรระดับสังเคราะห์บนซอฟต์แวร์โดยใช้โปรแกรม Design Manager	29
4.3 การกำหนดไฟล์นำทางให้กับวงจรที่จะทำการสร้างใหม่	29
4.4 การสร้างไฟล์สกุล .VHD และไฟล์สกุล .SDF ของโปรแกรม Design Manager	30



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

สารบัญตาราง

ตาราง	หน้า
2.1 การหาค่าผลคูณจากอัลกอริทึมการคูณเชื่อมตรงแบบข้อผิดพลาดติดลบ	7
2.2 ค่ารหัสวางคู่ในสาย (X,X') ที่ใช้แทนค่าตรรกะในบิต X	8
2.3 สถานะและเอาต์พุตของการแปลงตัวเลขแบบอนเดอะฟลาย	15
3.1 จำนวนขั้นตอนของวงจรคูณเชื่อมตรงเมื่อออกแบบโดยใช้รูปแบบบรรทัดฐาน	19
3.2 ตารางค่าความจริงของวงจรเข้ารหัสแบบที่ 1	21
3.3 ตารางค่าความจริงของวงจรเข้ารหัสแบบที่ 2	23
3.4 ตารางค่าความจริงของการแปลงตัวเลขแบบอนเดอะฟลาย	25
4.1 ผลการทดลองของวงจรคูณสมวารแบบเชื่อมตรง	31
4.2 เวลาที่ใช้ในแต่ละขั้นตอนของวงจรคูณสมวารแบบเชื่อมตรงซึ่งออกแบบด้วย โครงสร้างรุ่นที่ 1	32
4.3 เวลาที่ใช้ในแต่ละขั้นตอนของวงจรคูณสมวารแบบเชื่อมตรงซึ่งออกแบบด้วย โครงสร้างรุ่นที่ 2	32

บทที่ 1

บทนำ

การทำงานของวงจรรวมมีผลอย่างมากต่อประสิทธิภาพของไมโครโปรเซสเซอร์ และระบบการประมวลผลสัญญาณเชิงเลข (Digital Signal Processing System: DSP) อื่นๆ เพราะวงจรรวมเป็นวงจรมีขั้นตอนการทำงานค่อนข้างซับซ้อนและมีการทำงานหลายขั้นตอน ดังนั้นความเร็วของวงจรรวมจะเป็นตัวกำหนดอัตราปริมาณงาน (Throughput Rate) ของไมโครโปรเซสเซอร์ และระบบการประมวลผลสัญญาณเชิงเลข นอกจากนี้การประมวลผลภาพสามมิติจะใช้วงจรรวมเชิงตรรกะความเร็วสูงในการประมวลผลโปรแกรมประยุกต์ภาพสามมิติแบบเวลาจริง (Real-Time 3D Graphic Applications) [7] เพราะบ่อยครั้งที่การประมวลผลภาพสามมิติจะต้องคูณเลขเชิงตรรกะจำนวนมาก ด้วยเหตุผลเหล่านี้ควรออกแบบวงจรรวมให้มีประสิทธิภาพในการทำงานสูงเพื่อตอบสนองต่อความต้องการใช้งาน

1.1 ความเป็นมาและความสำคัญของปัญหา

ในการออกแบบวงจรรวมแบบสมวาร (Synchronous Circuit) ซึ่งเป็นวงจรมีสัญญาณนาฬิกาในการควบคุมการทำงานของวงจรรวม เพื่อลดความซับซ้อนในการออกแบบวงจรรวม และทำให้วงจรรวมสามารถทำงานได้อย่างถูกต้อง แต่การใช้สัญญาณนาฬิกาทำให้เกิดปัญหาต่างๆ [17] เช่น ปัญหาการแกว่งของสัญญาณนาฬิกา (Clock Skew) ปัญหาการทำงานบนความหน่วงที่ช้าที่สุด (Worst-Case Delay Operation) ปัญหาการออกแบบวงจรรวมในแต่ละส่วน (Modular Design) ปัญหาการสิ้นเปลืองพลังงาน (Power Dissipation) เป็นต้น จากปัญหาเหล่านี้ทำให้เกิดแนวความคิดในการออกแบบวงจรรวมแบบไม่สมวาร (Asynchronous Circuit) ซึ่งไม่ใช้สัญญาณนาฬิกาในการควบคุมการทำงานของวงจรรวม ทำให้ไม่เกิดปัญหาการแกว่งของสัญญาณนาฬิกา เวลาที่ใช้ในการทำงานเป็นเวลาที่สั้น ใช้พลังงานต่ำ และสามารถออกแบบแต่ละส่วนของวงจรรวมอย่างเป็นอิสระต่อกันได้

งานวิจัยนี้ได้เลือกอัลกอริทึมการคูณแบบเชื่อมตรง (On-Line Multiplication Algorithm) [1] เป็นอัลกอริทึมการคูณเลขเชิงตรรกะหนึ่งที่น่าสนใจ เนื่องจากการคำนวณของอัลกอริทึมแบบเชื่อมตรง (On-Line Arithmetic) [13] เป็นการคำนวณแบบหลักต่อหลัก (Computation Overlapped with Digit-By-Digit) โดยเริ่มต้นทำงานจากหลักที่มีนัยสำคัญมากที่สุดไปยังหลักที่มีนัยสำคัญน้อยที่สุด (Most Significant Digit First: MSDF) นอกจากนี้การหาผลคูณของอัลกอริทึมการคูณแบบเชื่อมตรง จะหาจากค่าของผลคูณครั้งหนึ่งจากผลคูณทั้งหมด โดยค่าผลคูณที่ได้จะเป็นค่าประมาณ (Estimated Value) ที่มากกว่าหรือเท่ากับค่าที่แน่นอน (Exact Value) ของผลคูณ

เสมอ ซึ่งการหาผลคูณของวงจรมูลเลขเชิงตรรกะที่ใช้กันอยู่ในปัจจุบัน จะหาจากค่าของผลคูณครั้งหนึ่งจากผลคูณทั้งหมดเช่นกัน แต่ไม่สามารถรับประกันได้ว่า ค่าผลคูณที่ได้จะมีค่ามากกว่า น้อยกว่า หรือเท่ากับค่าที่แน่นอนของผลคูณ

งานวิจัยเกี่ยวกับวงจรมูลเลขเชิงตรรกะได้นำเสนอ [8] ซึ่งบทความได้เสนอโครงสร้างเพื่อรวมหลักการของโครงสร้างแบบแถวลำดับ (Array) และโครงสร้างแบบต้นไม้ (Tree) เพื่อให้ได้วงจรมูลที่ใช้เวลาในการทำงานต่ำและค่าใช้จ่าย (Overhead) ในการเดินสายและใช้พื้นที่ของวงจรมูลสูงมากนัก ส่วนงานวิจัยเกี่ยวกับวงจรมูลแบบอสสมวารเป็นการออกแบบวงจรมูลอสสมวารแบบไปป์ไลน์ไขว้ (Asynchronous Cross-Pipelined Multiplier) [11] โดยนำเสนอโครงสร้างแถวลำดับแบบใหม่ซึ่งสามารถทำงานได้เร็วขึ้น และใช้รหัสรางคู่ (Dual-rail Code) ในการออกแบบวงจรมูลเนื่องจากวงจรมูลที่ออกแบบได้จะไม่มีเกตผกผัน (Inverter gate) ทำให้ไม่เกิดข้อผิดพลาดขึ้นบนการทำงานของวงจรมูล นอกจากนี้ได้มีงานวิจัย [15] ซึ่งออกแบบวงจรมูลด้วยอัลกอริทึมของบูท (Booth's Algorithm) โดยใช้รหัสรางคู่ในการออกแบบวงจรมูล และเปรียบเทียบวงจรมูลที่ออกแบบด้วยโครงสร้าง 3 แบบคือ โครงสร้างแบบเซลฟีทิม์ โครงสร้างแบบขนาน และโครงสร้างแบบไปป์ไลน์ ซึ่งพบว่า วงจรมูลที่ออกแบบด้วยโครงสร้างแบบไปป์ไลน์มีประสิทธิภาพดีที่สุด เมื่อเทียบกับโครงสร้างอีก 2 แบบ

ในปัจจุบันไม่มีการนำอัลกอริทึมการคูณแบบเชื่อมตรงมาสร้างวงจรมูลที่สามารถนำไปใช้งานได้จริง ดังนั้นงานวิจัยนี้ได้มุ่งประเด็นที่การออกแบบวงจรมูลเลขเชิงตรรกะแบบอสสมวารโดยใช้อัลกอริทึมการคูณแบบเชื่อมตรง เพื่อให้สามารถนำไปใช้กับไมโครโปรเซสเซอร์แบบอสสมวารได้ ซึ่งโครงสร้างที่จะนำมาทดลองในงานวิจัยนี้ คือ โครงสร้างแบบไปป์ไลน์ และใช้ภาษาวีเอชดีแอล (VHDL) ในการอธิบายลักษณะของวงจรมูล และทำการทดสอบการทำงานของวงจรมูลจากวงจรมูลระดับสังเคราะห์บนเอฟพีจีเอ (Field Programmable Gate Array: FPGA) โดยการจำลองการทำงาน (Simulation) แบบมีความหน่วง

1.2 วัตถุประสงค์ของการวิจัย

1. เพื่อศึกษาการทำงานของวงจรมูลแบบอสสมวาร
2. เพื่อศึกษาการสร้างวงจรมูลโดยวิธีการไปป์ไลน์แบบอสสมวาร
3. เพื่อศึกษาถึงแนวทางการสร้างวงจรมูลโดยใช้อัลกอริทึมการคูณแบบเชื่อมตรง

1.3 ขอบเขตของการวิจัย

1. ออกแบบโครงสร้างของวงจรคุณเลขเชิงตรรกษณ์แบบอสมวาร โดยใช้อัลกอริทึมการคุณแบบเชื่อมตรงขนาดไม่ต่ำกว่า 16 x 16 บิต และได้ผลคุณไม่ต่ำกว่า 16 บิต
2. ใช้ภาษาวีเอชดีแอลในการอธิบายวงจร และใช้เอพพีจีเอในการอ้างอิงวงจรระดับสังเคราะห์
3. ทดสอบความถูกต้องของวงจรโดยใช้การจำลองการทำงานแบบมีความหน่วงจากวงจรระดับสังเคราะห์บนเอพพีจีเอ

1.4 ประโยชน์ที่ได้รับ

1. เป็นแนวทางการวิจัยเริ่มต้นของการออกแบบวงจรคุณแบบอสมวาร
2. เป็นแนวทางการวิจัยเริ่มต้นของการใช้วิธีการไปป์ไลน์บนวงจรแบบอสมวาร
3. เป็นแนวทางการวิจัยเริ่มต้นในการปรับเปลี่ยนอัลกอริทึมแบบเชื่อมตรงมาใช้กับระบบอสมวาร
4. สามารถนำวงจรคุณที่ได้ไปใช้ร่วมกับไมโครโปรเซสเซอร์แบบอสมวารที่ยังไม่มีวงจรคุณได้

1.5 ขั้นตอนดำเนินการวิจัย

1. ศึกษาและค้นคว้างานวิจัยเกี่ยวกับวงจรคุณแบบอสมวารและวงจรคุณเลขเชิงตรรกษณ์
2. ศึกษาแนวคิดและทฤษฎีที่เกี่ยวข้อง ได้แก่
 - อัลกอริทึมการคุณแบบเชื่อมตรง
 - การออกแบบวงจรแบบอสมวารโดยใช้รหัสรางคู่
 - วิธีการไปป์ไลน์แบบอสมวาร
 - การออกแบบส่วนวงจรรางคู่ และส่วนวงจรตอบรับ
 - การออกแบบและสร้างวงจรอสมวารโดยใช้เอพพีจีเอ
 - การแปลงตัวเลขแบบอนเดอะฟลาย
3. ออกแบบโครงสร้างของวงจรคุณให้เหมาะสมกับระบบอสมวารโดยใช้อัลกอริทึมการคุณแบบเชื่อมตรง
4. เขียนโปรแกรมอธิบายฮาร์ดแวร์ของวงจรคุณโดยใช้ภาษาวีเอชดีแอล และทำการตรวจสอบความถูกต้องของวงจรโดยการจำลองการทำงาน

5. ทดสอบการทำงานของวงจรรคูณจากวงจรระดับสังเคราะห์บนเอฟพีจีเอ โดยการจำลองการทำงานแบบมีความหน่วง

1.6 ลำดับขั้นตอนในการเสนอผลการวิจัย

วิทยานิพนธ์นี้แบ่งเนื้อหาออกเป็น 5 บท คือ บทที่ 1 เป็นบทนำซึ่งกล่าวถึงความสำคัญและที่มาของปัญหา ตลอดจนวัตถุประสงค์ของงานวิจัย บทที่ 2 เป็นการสรุปแนวคิดและเนื้อหาที่เกี่ยวข้องกับงานวิจัย บทที่ 3 จะนำเสนอการออกแบบและพัฒนาโครงสร้าง ซึ่งใช้ในการสร้างวงจรรคูณสมวารแบบเชื่อมต่อตรง บทที่ 4 เสนอการทดสอบวงจร โดยทำการจำลองการทำงานของวงจรแบบมีความหน่วง ซึ่งอ้างอิงจากวงจรระดับสังเคราะห์บนเอฟพีจีเอ และบทที่ 5 เป็นการสรุปผลการวิจัยและข้อเสนอแนะ

1.7 ผลงานที่ตีพิมพ์จากงานวิจัย

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้ตีพิมพ์เป็นบทความทางวิชาการ ในหัวข้อ “Design of Asynchronous On-Line Multiplier” โดยปิยะ วราบุญทวีสุข อาทิตย์ ทองทักษ์ และอรรถสิทธิ์ สุรฤกษ์ ในงานประชุมวิชาการ “The 7th Annual National Symposium on Computational Science and Engineering (ANSCSE7)” ซึ่งจัดโดยภาควิชาวิทยาศาสตร์เคมี คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ณ คณะวิทยาศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในระหว่างวันที่ 24-26 มีนาคม 2546

ส่วนหนึ่งของวิทยานิพนธ์นี้ได้ตีพิมพ์เป็นบทความทางวิชาการ ในหัวข้อ “การปรับปรุงวงจรรคูณสมวารแบบเชื่อมต่อตรงโดยใช้เทคนิคการทำให้เป็นบรรทัดฐาน” โดยปิยะ วราบุญทวีสุข อาทิตย์ ทองทักษ์ และอรรถสิทธิ์ สุรฤกษ์ ในงานประชุมวิชาการ “The 7th National Computer Science and Engineering Conference (NCSEC 2003)” ซึ่งจัดโดยภาควิชาวิทยาศาสตร์คอมพิวเตอร์ คณะวิทยาศาสตร์ มหาวิทยาลัยบูรพา ณ มหาวิทยาลัยบูรพา จังหวัดชลบุรี ในระหว่างวันที่ 28-30 ตุลาคม 2546

บทที่ 2

แนวคิดและทฤษฎีที่เกี่ยวข้อง

ในบทนี้จะกล่าวถึงทฤษฎีต่างๆที่เกี่ยวข้องกับงานวิจัยซึ่งได้แก่ อัลกอริทึมการคูณแบบ เชื่อมตรง การออกแบบวงจรแบบอสถวารโดยใช้รหัสรางคู่ การออกแบบวงจรอสถวารโดยใช้ เอฟพีจีเอ วิธีการไปป์ไลน์แบบอสถวาร และการแปลงตัวเลขแบบอนเดอะฟลาย

2.1 อัลกอริทึมการคูณแบบเชื่อมตรง

อัลกอริทึมการคูณแบบเชื่อมตรงเป็นอัลกอริทึมที่เริ่มต้นทำการคูณจากหลักที่มีนัยสำคัญ มากที่สุดไปยังหลักที่มีนัยสำคัญน้อยที่สุด ทั้งนี้อัลกอริทึมการคูณแบบเชื่อมตรงจะมีความหน่วง เชื่อมตรง (On-Line Delay) δ ซึ่งเป็นเลขจำนวนเต็มขนาดเล็กที่สามารถทำให้อัลกอริทึมทำงานได้ อย่างถูกต้อง โดยทำให้อัลกอริทึมสามารถหาค่าของผลลัพธ์ n หลักแรกได้ก็ต่อเมื่ออัลกอริทึมได้รับ อินพุตเป็น $n + \delta$ หลักแรกแล้ว เช่น อัลกอริทึมมีค่า $\delta = 1$ อัลกอริทึมจะสามารถผลิตเอาต์พุต ตั้งแต่หลักที่ 13 ถึงหลักที่ 3 ได้ก็ต่อเมื่อได้รับอินพุต 4 หลักแรกแล้ว เป็นต้น ค่าของความหน่วง เชื่อมตรงจะขึ้นอยู่กับช่วงของอินพุตหรือตัวดำเนินการ (Operand) ซึ่งจะมีค่ามากกว่า 0 และน้อยกว่า $\frac{1}{\beta^\delta}$ โดย β เป็นฐานของระบบตัวเลข (Base Number System) ที่ใช้ในการคูณ ซึ่งเป็นเลข จำนวนเต็มมีค่ามากกว่าหรือเท่ากับ 2 และเรียก $\frac{1}{\beta^\delta}$ ว่า ค่าขอบเขตของตัวดำเนินการ [5] (Operand Bound)

2.1.1 ระบบตัวเลขซ้ำซ้อนที่มีเครื่องหมาย (Redundant Sign-Digit Number System)

เนื่องจากอัลกอริทึมการคูณแบบเชื่อมตรงจะใช้ระบบตัวเลขที่มีเครื่องหมายเพื่อแสดงค่า ผลคูณที่ได้จากอัลกอริทึม ซึ่งตัวเลขทุกค่าจะมีการแสดงได้มากกว่าหนึ่งรูปแบบ [2] เมื่อฐานของ ระบบตัวเลขมีค่าเท่ากับตัวเลขแต่ละหลักจะมีค่าอยู่บนเซต $\mathcal{E} = \{-1, 0, 1\}$ เช่น 19 จะสามารถแสดง ได้โดยค่า 10011 หรือ 1010 $\bar{1}$ โดย $\bar{1}$ แทนค่าของ -1 เป็นต้นดังแสดงในรูปที่ 2.1

β^4	β^3	β^2	β^1	β^0
16	8	4	2	1
1	0	0	1	1
1	0	1	0	$\bar{1}$

รูปที่ 2.1 การแสดงค่า 19 บนระบบตัวเลขฐานสองที่มีเครื่องหมาย

2.1.2 อัลกอริทึมการคูณเชื่อมตรงแบบข้อผิดพลาดติดลบ (Negative Error On-line Multiplication Algorithm)

อัลกอริทึมการคูณแบบเชื่อมตรงที่ใช้ในวิทยานิพนธ์นี้คือ อัลกอริทึมการคูณเชื่อมตรงแบบข้อผิดพลาดติดลบ โดยมี A และ B เป็นตัวดำเนินการหรืออินพุต ซึ่งรูปแบบของ A และ B เป็นดังนี้ $A = (.a_{-1}a_{-2}\dots)_\beta$ และ $B = (.b_{-1}b_{-2}\dots)_\beta$ โดยแต่ละหลักของตัวดำเนินการจะอยู่บนเซต $C = \{c \in \mathbb{Z} \mid 0 \leq c \leq \beta - 1\}$ และผลคูณหรือเอาต์พุตแทนด้วย X ซึ่งอยู่บนระบบตัวเลขซ้ำซ้อนที่มีเครื่องหมาย และมีค่าอยู่บนเซต $\mathcal{E} = \{e \in \mathbb{Z} \mid -\beta + 1 \leq e \leq \beta - 1\}$ การทำงานของอัลกอริทึมการคูณเชื่อมตรงแบบข้อผิดพลาดติดลบเป็นดังนี้

input:

$$A = (.a_{-1}a_{-2}\dots)_\beta \text{ where } a_j \in C \text{ and}$$

$$B = (.b_{-1}b_{-2}\dots)_\beta \text{ where } b_j \in C$$

$$a_{-1} = a_{-2} = \dots = a_{-s} = 0 \text{ and } b_{-1} = b_{-2} = \dots = b_{-s} = 0$$

output:

$$X = (.x_{-1}x_{-2}\dots)_\beta \text{ where } x_j \in \mathcal{E} \text{ such that}$$

$$\|X\| = \sum_{j \leq -1} x_j \beta^j = \|A\| \cdot \|B\|$$

begin

$$W_0 = 0;$$

$$B_0 = 0;$$

$$x_0 = 0;$$

$$j = -1;$$

while $j \leq -1$ do

$$W_j = \beta (W_{j+1} - x_{j+1}) + A_j b_j + B_{j+1} a_j; \quad (2.1)$$

$$x_j = \lceil W_j \rceil; \quad (2.2)$$

$$j = j - 1;$$

enddo;

end;

โดย $\beta (W_{j+1} - x_{j+1})$ ในสมการที่ (2.1) คือ ฟังก์ชันเศษเหลือ [13] (Residual Function) ของอัลกอริทึมการคูณแบบเชื่อมตรง และ $A_j = (.a_{-1}a_{-2}\dots a_j)_\beta$ สำหรับ W_j คือ ผลการคูณบางส่วน (Partial Product) ซึ่งจะมีค่าอยู่บนช่วง $(-\beta, 2\beta^\delta (\beta - 1)]$

เมื่ออยู่บนเลขฐานสองอินพุตของอัลกอริทึมจะอยู่บนเซต $\{0,1\}$ และเอาต์พุตอยู่บนเซต $\{\bar{1}, 0, 1\}$ ซึ่ง $\bar{1}$ แทนค่า -1 อัลกอริทึมจะทำงานได้อย่างถูกต้อง เมื่อ δ มีค่าเท่ากับ 1 ขอบเขตของตัวดำเนินการบนระบบเลขฐานสองมีค่าเท่ากับ $1/2$ หรือ 0.5 และค่าของ W_j จะเป็น $-2 < W_j \leq 1$ ดังนั้นการปัดค่าขึ้นของสมการที่ (2.2) จะทำดังนี้ เมื่อ $-2 < W_j \leq -1$ ค่าของ x_j จะเท่ากับ $\bar{1}$ เมื่อ $-1 < W_j \leq 0$ ค่าของ x_j จะเท่ากับ 0 และเมื่อ $0 < W_j \leq 1$ ค่าของ x_j จะเท่ากับ 1 และหากต้องการให้ตัวดำเนินการซึ่งเป็นเลขเชิงตรรกะนี้มีค่ามากกว่า 0.5 และทำให้อัลกอริทึมทำงานได้อย่างถูกต้องสามารถทำได้โดยเปลี่ยนสมการที่ (2.1) เป็นสมการที่ (2.3)

$$W_j = \beta(W_{j+1} - \beta x_{j+2}) + A_j b_j + B_{j+1} a_j; \quad (2.3)$$

$$x_{j+1} = \lceil W_j / \beta \rceil; \quad (2.4)$$

เมื่อเปลี่ยนสมการ (2.1) เป็นสมการ (2.3) การปัดค่าในสมการ (2.2) จะเปลี่ยนเป็นสมการ (2.4) โดยมีการปัดค่าดังนี้ เมื่อ $-4 < W_j \leq -2$ ค่าของ x_{j+1} จะเท่ากับ $\bar{1}$ เมื่อ $-2 < W_j \leq 0$ ค่าของ x_{j+1} จะเท่ากับ 0 และเมื่อ $0 < W_j \leq 2$ ค่าของ x_{j+1} จะเท่ากับ 1 การหาค่าผลคูณของอัลกอริทึมการคูณเชื่อมตรงแบบข้อผิดพลาดติดลบแสดงดังตัวอย่างที่ 2.1

ตัวอย่างที่ 2.1 เมื่อ $A = B = (.01101111)_2$

ตารางที่ 2.1 การหาค่าผลคูณจากอัลกอริทึมการคูณเชื่อมตรงแบบข้อผิดพลาดติดลบ

j	$A_j b_j + B_{j+1} a_j$	W_j	x_{j+1}	$2(W_{j+1} - 2x_{j+2})$
-1	0	0	0	0
-2	0.01	00.01	1	-11.1
-3	0.101	-10.111	$\bar{1}$	-01.11
-4	0	-01.11	0	-11.100
-5	0.11001	-10.10111	$\bar{1}$	-01.0111
-6	0.110101	-00.100111	0	-01.00111
-7	0.1101101	-00.0101111	0	-00.101111
-8	0.11011101	00.00100001	1	-11.1011111
-9	0	-11.1011111	$\bar{1}$	-11.011111

จากตารางที่ 2.1 ค่าของ X คือ $(.1\bar{1}0\bar{1}001\bar{1}\dots)_2$ ซึ่งมีค่าเท่ากับ 0.19140625 และค่าที่แน่นอนของผลคูณจากตัวอย่างที่ 2.1 คือ 0.18800354 จะเห็นได้ว่า ค่าที่ได้จากการหาค่าผลคูณจากอัลกอริทึมการคูณเชื่อมตรงแบบข้อผิดพลาดติดลบจะเป็นค่าประมาณที่มากกว่าค่าที่แน่นอน

และอาจมีบางกรณีที่มีค่าประมาณจากการหาค่าผลคูณจากอัลกอริทึมการคูณเชื่อมตรงแบบข้อผิดพลาดติดลบจะมีค่าเท่ากับค่าที่แน่นอน

2.2 การออกแบบวงจรแบบอสมวารโดยใช้รหัสรางคู่

ในการออกแบบวงจรแบบอสมวารจะไม่มีกำหนดค่าความหน่วงของเกตและสายภายในวงจร ทำให้การออกแบบไม่สามารถทราบค่าความหน่วงจริง (Actual Delay) ของการเปลี่ยนระดับสัญญาณภายในวงจร และในการส่งสัญญาณข้อมูลจะไม่สามารถระบุถึงการเปลี่ยนแปลงของระดับสัญญาณซึ่งคงที่ในช่วงเวลาหนึ่งได้ เช่น การเปลี่ยนระดับสัญญาณ $1 \rightarrow 1$ เป็นต้น ดังนั้นการออกแบบวงจรจะต้องทำการเข้ารหัสข้อมูลด้วยรหัสรางคู่ ซึ่งเป็นรหัสที่ใช้ในการส่งและแยกค่าระดับสัญญาณข้อมูลในแต่ละรอบการทำงานของวงจร โดยค่ารหัสรางคู่เป็นดังตารางที่ 2.2

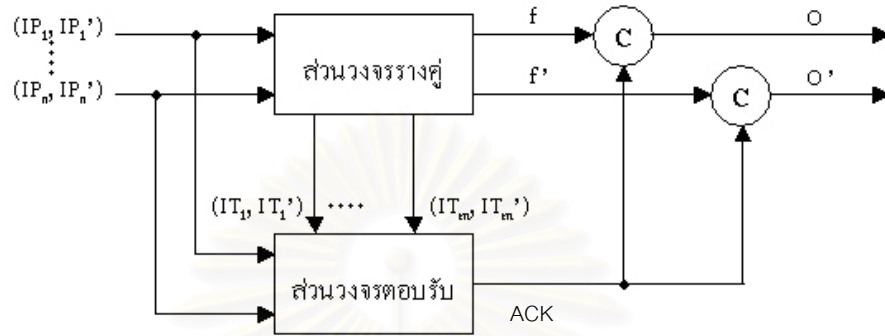
ตารางที่ 2.2 ค่ารหัสรางคู่ในสาย (X,X') ที่ใช้แทนค่าตรรกะในบิต X

ค่ารหัสรางคู่ในสาย (X,X')	ค่าตรรกะในบิต X
(0,1)	0
(1,0)	1
(0,0)	ตัวแบ่งรอบการทำงาน (Spacer)
(1,1)	ไม่ใช่

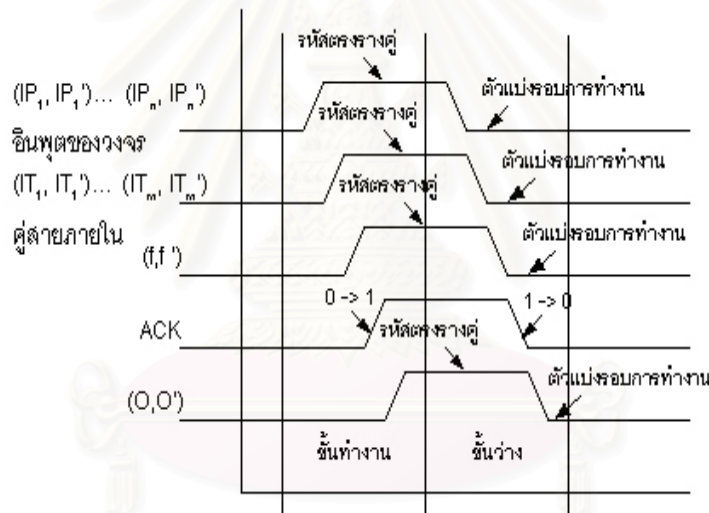
จากตารางที่ 2.2 รหัสรางคู่จะแสดงสัญญาณข้อมูลแต่ละบิตด้วยสายสัญญาณสองเส้น (X,X') โดยแทนค่าตรรกะศูนย์และค่าตรรกะหนึ่งในบิต X ด้วยรหัสรางคู่ (2-Rail Codeword) มีค่าเท่ากับ (0,1) และ (1,0) ตามลำดับ และแทนค่าที่ใช้แยกการส่งค่าตรรกะของแต่ละรอบการทำงาน หรือเรียกว่า ตัวแบ่งรอบการทำงาน ด้วยค่า (0,0) ดังนั้นสามารถแบ่งการทำงานในแต่ละรอบของวงจรออกเป็น 2 ขั้นตอนคือ ขั้นตอนการทำงาน (Working Phase) ซึ่งจะได้รับอินพุตเป็นรหัสรางคู่ และขั้นว่าง (Idle Phase) ซึ่งจะได้รับอินพุตเป็นตัวแบ่งรอบการทำงาน โดยทั้งสองขั้นตอนจะทำงานสลับกัน เรียกว่า การทำงานรางคู่แบบสองขั้นชนิดกลับสู่ศูนย์ (2-Rail 2-Phase Return-to-Zero Operation)

เมื่อออกแบบวงจรโดยใช้รหัสรางคู่ วงจรที่ออกแบบได้จะประกอบด้วยสองส่วนคือ ส่วนวงจรรางคู่ (Dual-Rail Circuit) ทำหน้าที่คำนวณตามฟังก์ชันตรรกะที่ต้องการ และส่วนวงจรตอบรับ (Acknowledgement Circuit) ทำหน้าที่ตรวจสอบการสิ้นสุดการเปลี่ยนระดับสัญญาณในวงจรรางคู่ โดยจะพิจารณาการเลือกสายสัญญาณจากส่วนวงจรรางคู่ไปยังส่วนวงจรตอบรับตาม

แบบจำลองความหน่วงที่ไม่ไวต่อความหน่วงชนิดปรับมาตราส่วนได้ (Scalable Delay Insensitive Model: SDI Model) [3] และมีอุปกรณ์ชนิดซี (C-Element) ทำหน้าที่เชื่อมต่อผลลัพธ์จากทั้งสองส่วนเป็นเอาต์พุตของวงจรถัดไปแสดงในรูปที่ 2.2 วงจรแบบผสมวารที่ออกแบบได้จะมีลักษณะการเปลี่ยนระดับสัญญาณดังในรูปที่ 2.3 และมีลักษณะการทำงานดังนี้



รูปที่ 2.2 โครงสร้างวงจรแบบผสมวาร



รูปที่ 2.3 ลักษณะการเปลี่ยนระดับสัญญาณของวงจรรางคู่แบบผสมวารในการทำงาน

แบบสองขั้นกลับคืนสู่ศูนย์

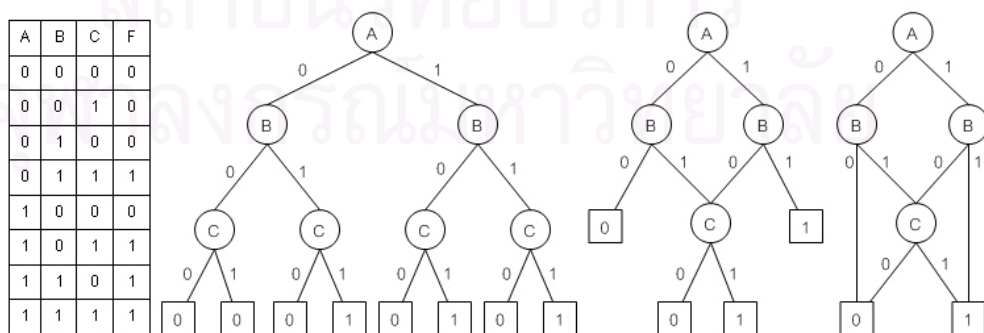
ขั้นทำงาน ส่วนวงจรรางคู่จะได้รับอินพุตเป็นรหัสตรงรางคู่ ทำให้คู่สายภายในวงจรบางเส้นมีการเปลี่ยนระดับสัญญาณ $0 \rightarrow 1$ และส่งผลให้เอาต์พุต (f, f') เป็นรหัสตรงรางคู่ตามฟังก์ชันตรรกะ และเมื่อทุกการเปลี่ยนระดับสัญญาณ $0 \rightarrow 1$ สิ้นสุดที่ค่าระดับสัญญาณ 1 แล้ว ส่วนวงจรตอบรับจะทำให้สัญญาณแสดงความบริบูรณ์ (Completion Signal : ACK) เปลี่ยนระดับสัญญาณ $0 \rightarrow 1$ เพื่อทำหน้าที่แลตช์ (Latch) อุปกรณ์ชนิดซีให้ส่งค่ารหัสตรงรางคู่ที่ (f, f') ไปเป็นค่าเอาต์พุตของวงจร (O, O')

ขั้นวาง ส่วนวงจรวางคู่จะได้รับอินพุตเป็นตัวแบ่งรอบการทำงาน ทำให้สายทุกเส้นที่มีค่าระดับสัญญาณเป็น 1 ในขั้นการทำงาน มีการเปลี่ยนแปลงระดับสัญญาณ $1 \rightarrow 0$ และให้เอาท์พุต (f,f') เป็นตัวแบ่งรอบการทำงาน และเมื่อสายทุกเส้นมีค่าระดับสัญญาณ 0 แล้วส่วนวงจรตอรับ จะให้สัญญาณแสดงความบริบูรณ์มีการเปลี่ยนระดับสัญญาณ $1 \rightarrow 0$ เพื่อแลตช์อุปกรณ์ชนิดซีให้ส่งค่าตัวแบ่งรอบการทำงานที่ (f,f') ออกเป็นเอาท์พุต (O,O') เพื่อแสดงว่า วงจรพร้อมที่จะเริ่มทำงานในขั้นการทำงานรอบถัดไป

จากลักษณะการทำงานดังกล่าว วงจรแบบอสมวารที่ออกแบบได้จะมีการเปลี่ยนระดับสัญญาณ $0 \rightarrow 1$ ในขั้นการทำงาน และการเปลี่ยนระดับสัญญาณ $1 \rightarrow 0$ ในขั้นวางเท่านั้น ทำให้สามารถรับประกันได้ว่า ไม่เกิดข้อผิดพลาด (Hazard) ขึ้นบนการทำงานของวงจรที่ออกแบบ

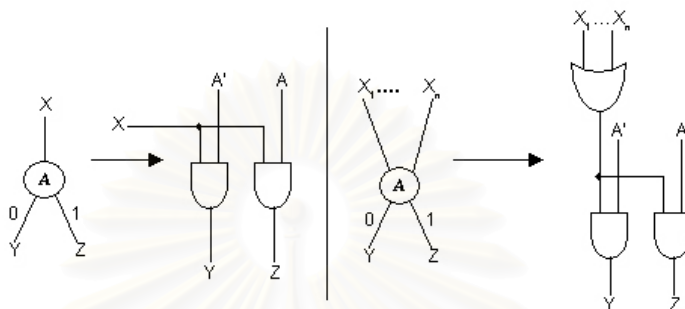
2.2.1 การออกแบบวงจรวางคู่โดยใช้แผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับ (Reduced-Ordered Binary Decision Diagram (ROBDD) Implementation)

แผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับ เป็นแผนภาพที่มีการกำหนดลำดับตัวแปรและสามารถที่จะลดขนาดของแผนภาพลงได้ และใช้แทนฟังก์ชันตรรกะเพื่อช่วยในการออกแบบและสังเคราะห์วงจรขนาดใหญ่ โดยจากลักษณะของแผนภาพ อินพุตแต่ละแบบจะมีเส้นเชื่อมเพียงหนึ่งเส้นเท่านั้นที่ให้อาท์พุตของฟังก์ชัน สามารถเขียนฟังก์ชันของตัวแปรอินพุตที่อยู่ในเส้นเชื่อมไปยังเอาท์พุตค่าตรรกะศูนย์และค่าตรรกะหนึ่งได้เป็นผลคูณของผลรวม (Product-of-Sum) และผลรวมของผลคูณ (Sum-of-Product) ตามลำดับ โดยรูปที่ 2.4 เป็นตัวอย่างการสร้างแผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับ ของฟังก์ชัน $F = AB + BC + AC$ ซึ่งเริ่มต้นด้วยการสร้างแผนภาพตัดสินใจแบบทวิภาคจากตารางค่าความจริงของฟังก์ชัน แล้วจึงทำการลดทอนอันดับ



รูปที่ 2.4 แผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับ สำหรับฟังก์ชัน $F = AB + BC + AC$

การออกแบบส่วนวงจรรางคู่จะทำการสร้างแผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับจากตารางค่าความจริงที่ต้องการ จากนั้นจะทำการแปลงแผนภาพตัดสินใจนั้นให้เป็นส่วนวงจรรางคู่ โดยเชื่อมต่อดวงจรดังรูปที่ 2.5 ส่วนวงจรรางคู่ประกอบด้วยเกตแอนด์ ซึ่งมีการจัดเรียงเป็นเส้นทางเทียบเท่ากับเส้นเชื่อมในแผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับเรียกว่า เส้นทางต่อเชื่อมเกตแอนด์ และเกตออร์ใช้รวมเส้นทางต่อเชื่อมเกตแอนด์



รูปที่ 2.5 การออกแบบส่วนวงจรรางคู่โดยใช้แผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับ

2.2.2 การออกแบบวงจรตอบรับ

การเปลี่ยนระดับสัญญาณที่สัญญาณแสดงความบริบูรณ์มีลักษณะตรงกับการเปลี่ยนระดับสัญญาณภายในวงจรรางคู่ทั้งในชั้นทำงานและชั้นว่าง จึงสามารถออกแบบส่วนวงจรตอบรับโดยให้สายอินพุตได้มาจากการเลือกกลุ่มของสายสัญญาณ ซึ่งสามารถครอบคลุมทุกการเปลี่ยนระดับสัญญาณภายในส่วนวงจรรางคู่ ทั้งในชั้นทำงานและชั้นว่าง และใช้เกตออร์ในการตรวจสอบการสิ้นสุดการเปลี่ยนระดับสัญญาณ เพื่อให้การเปลี่ยนระดับสัญญาณที่สัญญาณแสดงความบริบูรณ์มีการทำงานสัมพันธ์กับการเปลี่ยนระดับสัญญาณภายในส่วนวงจรรางคู่ โดยต้องรับประกันได้ว่าสัญญาณแสดงความบริบูรณ์จะต้องเกิดขึ้นหลังจากการเปลี่ยนระดับสัญญาณสุดท้ายภายในส่วนวงจรรางคู่ทั้งในชั้นทำงานและชั้นว่าง

2.3 การออกแบบและสร้างวงจรสมวารโดยใช้เอพพีจีเอ

ในการออกแบบวงจรสมวารโดยใช้เอพพีจีเอนั้น เราควรทราบถึงโครงสร้างภายในของเอพพีจีเอ รวมทั้งภาษาที่ใช้ในการอธิบายวงจร เพื่อให้สามารถนำวงจรที่ได้ไปสังเคราะห์ลงบนเอพพีจีเอได้

2.3.1 เอฟพีจีเอ [19,20]

เอฟพีจีเอเป็นอุปกรณ์ตรรกะที่สามารถโปรแกรมได้หลายครั้ง ส่วนใหญ่ใช้สำหรับสร้างวงจรต้นแบบอย่างรวดเร็ว (Rapid Prototype) หรือวงจรดิจิทัล (Digital Circuit) โดยข้อดีของการออกแบบวงจรลงบนเอฟพีจีเอคือ สามารถออกแบบและตรวจสอบวงจรได้ง่าย ค่าใช้จ่ายในการออกแบบและแก้ไขต่ำ เพราะสามารถตรวจสอบและแก้ไขวงจรให้ถูกต้องได้ด้วยการทำงานบนเครื่องคอมพิวเตอร์ส่วนบุคคล จนกระทั่งการทำงานของวงจรถูกต้องจริง แล้วจึงโปรแกรมลงบนเอฟพีจีเอ โครงสร้างภายในของเอฟพีจีเอ ประกอบด้วย 3 ส่วนหลักคือ

- กลุ่มระเบียบโครงแบบตรรกะ (Configuration Logic Block: CLB) เป็นส่วนที่ใช้การสร้างวงจรตรรกะ ซึ่งภายในประกอบด้วย ฟลิปฟลอป (Flip-flop), อุปกรณ์รวมส่งสัญญาณ (Multiplexer) และตารางค้นหา (Look-up Table) ซึ่งตารางค้นหาเป็นส่วนที่ใช้ในการสร้างฟังก์ชันตรรกะ
- กลุ่มระเบียบอินพุตหรือเอาต์พุต (I/O Block) เป็นส่วนที่ใช้สร้างอินพุตและเอาต์พุตของวงจร
- ส่วนเชื่อมโยงที่สามารถโปรแกรมได้ (Programmable Interconnect) เป็นส่วนที่ใช้ในการเชื่อมต่อสายสัญญาณของกลุ่มระเบียบโครงแบบตรรกะเข้าด้วยกัน

2.3.2 วีเอชดีแอล (VHDL) [10]

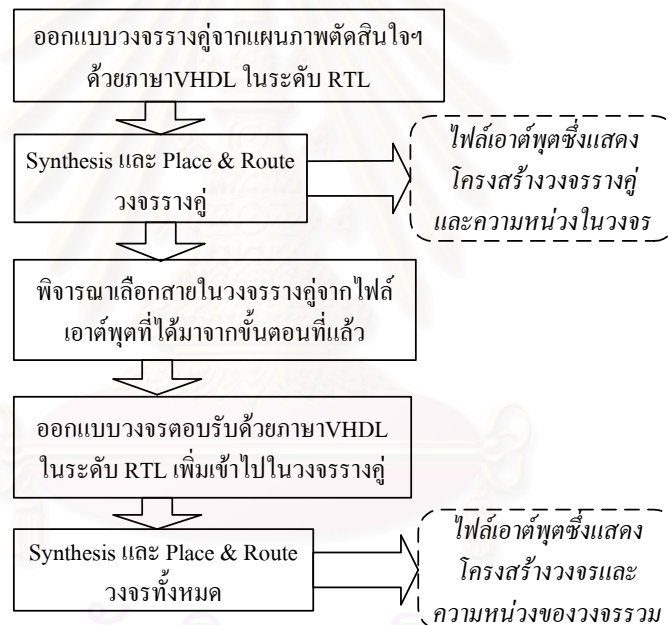
ภาษาวีเอชดีแอล เป็นภาษาอธิบายฮาร์ดแวร์ใช้สำหรับออกแบบวงจรดิจิทัล ลักษณะโครงสร้างทางภาษาเป็นการผสมกันระหว่างภาษาเชิงวัตถุ (Object Oriented Language) และภาษาโปรแกรมที่ทำงานพร้อมกัน (Concurrent Programming Language) โดยมองส่วนต่างๆ เป็นโมดูล (Module) หรือวัตถุ (Object) หนึ่ง โดยมีสัญญาณเชื่อมต่อระหว่างโมดูล ซึ่งการส่งสัญญาณระหว่างโมดูลจะเกิดขึ้นพร้อมกัน เช่นเดียวกับการแยกไหลของกระแสไฟฟ้าไปยังส่วนต่างๆ ของวงจร โดยภาษา VHDL สามารถเขียนอธิบายการทำงานของวงจรได้เป็น 3 ระดับคือ

- ระดับพฤติกรรม (Behavioral Level) เป็นการเขียนอธิบายถึงพฤติกรรมหรือฟังก์ชันการทำงานของวงจร
- ระดับการส่งผ่านระหว่างรีจิสเตอร์ (Register Transfer Level : RTL) เป็นการเขียนอธิบายถึงลักษณะการส่งผ่านข้อมูลระหว่างรีจิสเตอร์หรือเกตในวงจร

- ระดับโครงสร้าง (Structural Level) เป็นการเขียนอธิบายถึงโครงสร้างภายในวงจร รวมทั้งการติดต่อระหว่างคอมโพเนนท์ (Component) หรือโมดูลต่างๆภายในวงจร

2.3.3 ขั้นตอนการออกแบบวงจรและสร้างวงจรแบบอสมวารโดยใช้เอชพีจีเอ [21]

การออกแบบวงจรแบบอสมวารโดยใช้เอชพีจีเอ จะเริ่มจากการออกแบบวงจรร่างคู่โดยใช้แผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับดังที่ได้กล่าวไปข้างต้น และใช้ภาษาวีเอสดีแอลเขียนอธิบายการทำงานของส่วนวงจรร่างคู่ แล้วนำไปสังเคราะห์และสร้างเป็นส่วนวงจรร่างคู่ที่สามารถนำไปโปรแกรมลงบนเอชพีจีเอ หลังจากนั้นจึงทำการออกแบบส่วนวงจรตอบรับเพิ่มเติมเข้าไป โดยการออกแบบวงจรในส่วนตอบรับนั้นจะอาศัยวิธีการเลือกสายสัญญาณที่นำเสนอบน [21] แล้วนำวงจรใหม่ไปสังเคราะห์และสร้างเป็นวงจรแบบอสมวารที่สามารถนำไปโปรแกรมลงบนเอชพีจีเออีกครั้งดังรูปที่ 2.6

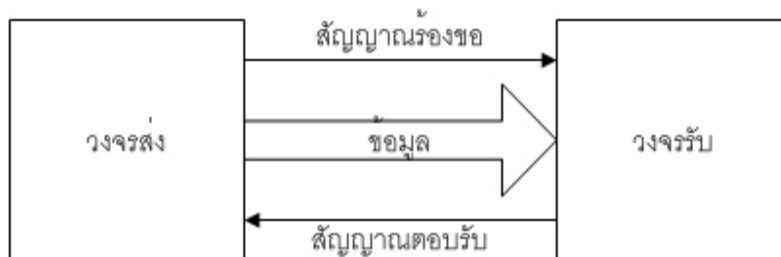


รูปที่ 2.6 ขั้นตอนการออกแบบวงจรแบบอสมวารโดยใช้เอชพีจีเอ [21]

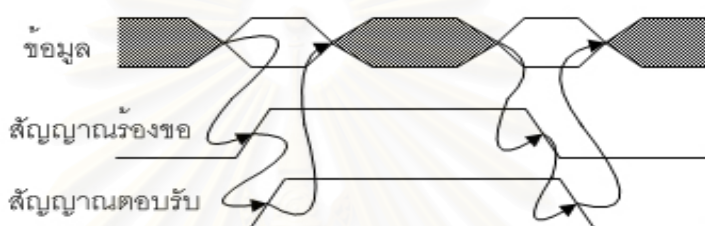
2.4 วิธีการไปป์ไลน์แบบอสมวาร (Asynchronous Pipeline Approach) [9,16]

วิธีการไปป์ไลน์แบบอสมวารจะใช้สัญญาณร้องขอ (Request Signal) และสัญญาณตอบรับ (Acknowledgement Signal) เป็นสัญญาณควบคุมแทนการใช้สัญญาณนาฬิกาในการควบคุมการทำงานของวงจร โดยการทำงานของไปป์ไลน์แบบอสมวารเป็นดังรูปที่ 2.7 เมื่อวงจรส่งทำงานเสร็จสิ้นจะส่งสัญญาณร้องขอไปยังวงจรรับ เพื่อให้วงจรรับนำข้อมูลไปทำงานต่อ เมื่อวงจรรับได้รับข้อมูลแล้วจะส่งสัญญาณตอบรับกลับไปให้วงจรส่ง โดยมีโปรโตคอล (Protocol) ที่ใช้ควบคุมการรับส่งข้อมูลของไปป์ไลน์แบบอสมวาร 2 รูปแบบ คือ โปรโตคอลสถานะสัญญาณแบบ 2

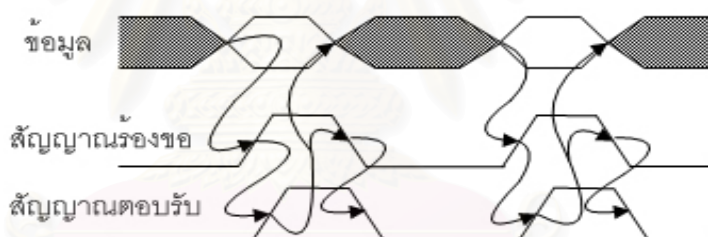
ขั้นตอน (2-Phase Handshake Protocol) และโปรโตคอลลำดับสัญญาแบบ 4 ขั้นตอน (4-Phase Handshake Protocol) ดังแสดงในรูปที่ 2.8 และ 2.9 ตามลำดับ



รูปที่ 2.7 แผนภาพการทำงานของระบบไปป์ไลน์แบบอสมวาร



รูปที่ 2.8 โปรโตคอลลำดับสัญญาแบบ 2 ขั้นตอน

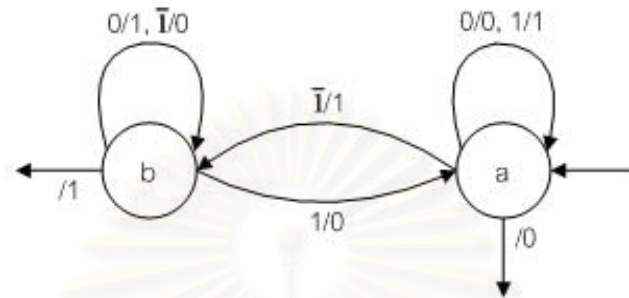


รูปที่ 2.9 โปรโตคอลลำดับสัญญาแบบ 4 ขั้นตอน

จากรูปที่ 2.7 เมื่อวงจรถูกส่งทำการคำนวณเสร็จสิ้น วงจรถูกส่งจะส่งสัญญาขอรับส่งไปยังวงจรถูกรับ และคงค่าของข้อมูลไว้จนกว่าจะได้รับสัญญาตอบรับจากวงจรถูกรับ เมื่อเป็นโปรโตคอลลำดับสัญญาแบบ 2 ขั้นตอน หากวงจรถูกส่งได้รับสัญญาตอบรับจากวงจรถูกรับ วงจรถูกส่งจะเริ่มทำการคำนวณรอบต่อไปและคงค่าของสัญญาขอรับส่งไว้จนกว่าจะเสร็จสิ้นการทำงานในรอบต่อไป แต่ ถ้าเป็นโปรโตคอลลำดับสัญญาแบบ 4 ขั้นตอน วงจรถูกส่งจะทำการตั้งสัญญาขอรับส่งกลับสู่ศูนย์ พร้อมกับเริ่มทำการคำนวณในรอบถัดไป

2.5 การแปลงตัวเลขแบบออนเดอะฟลาย (On-the-Fly Conversion) [4, 12]

การแปลงเลขที่อยู่ในระบบตัวเลขซ้ำซ้อนที่มีเครื่องหมายให้เป็นตัวเลขแบบทั่วไปสามารถทำได้ โดยใช้การแปลงตัวเลขแบบออนเดอะฟลาย ซึ่งการทำงานของ การแปลงตัวเลขแบบออนเดอะฟลายแสดงโดยออโตมาตา (Automata) ดังรูปที่ 2.10



รูปที่ 2.10 การแปลงตัวเลขซ้ำซ้อนที่มีเครื่องหมายแบบออนเดอะฟลาย

เมื่อ a และ b เป็นสถานะ (State) ของการแปลงตัวเลข โดยเริ่มทำงานที่สถานะ a และทำงานต่อไป และใช้ค่าจาก ตารางที่ 2.3 ซึ่งสถานะสุดท้าย (Terminal State) ของ a มีค่าเท่ากับ 0 และสถานะสุดท้ายของ b มีค่าเท่ากับ 1

ตารางที่ 2.3 สถานะและเอาต์พุตของการแปลงตัวเลขแบบออนเดอะฟลาย

อินพุต	0	1	$\bar{1}$
$A[k+1]$	$(A[k], 0)$	$(A[k], 1)$	$(B[k], 1)$
$B[k+1]$	$(B[k], 1)$	$(A[k], 0)$	$(B[k], 0)$

สรุป

งานวิจัยนี้ได้นำทฤษฎีต่างๆที่ได้กล่าวข้างต้น ไปประยุกต์ใช้ในการออกแบบวงจรคูณเชื่อมตรงให้เหมาะสมกับระบบอสมวาร โดยจะทดลองจากวงจรระดับสังเคราะห์บนเอพพีซีไอ ซึ่งในบทที่ 3 จะกล่าวถึงการออกแบบโครงสร้างและวงจรรายในบางส่วนที่สำคัญ เพื่อให้สามารถนำการออกแบบที่ได้ไปใช้ในการเขียนภาษาอธิบายฮาร์ดแวร์ของวงจรรคูณอสมวารแบบเชื่อมตรงได้

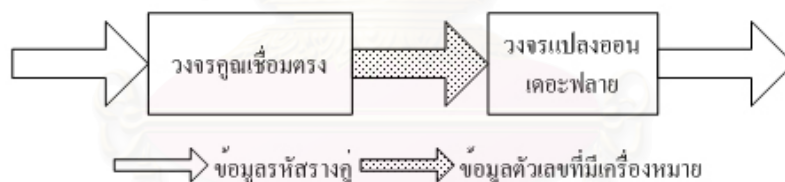
บทที่ 3

แนวคิดในการออกแบบและพัฒนางจร

ในบทนี้จะกล่าวถึง การนำอัลกอริทึมการคูณแบบเชื่อมตรงมาประยุกต์ใช้กับการออกแบบและพัฒนางจรให้เหมาะสมกับระบบอสมวาร โดยแบ่งเป็นการออกแบบโครงสร้างของวงจรมคูณอสมวารแบบเชื่อมตรง และการออกแบบวงจรมคูณในบางส่วน

3.1 การออกแบบโครงสร้างของวงจรมคูณอสมวารแบบเชื่อมตรง

จากอัลกอริทึมการคูณแบบเชื่อมตรง สามารถออกแบบโครงสร้างของวงจรมคูณได้ดังรูปที่ 3.1 ซึ่งเรียกกวงจรมคูณที่ออกแบบด้วยอัลกอริทึมการคูณแบบเชื่อมตรงว่า “วงจรมคูณอสมวารแบบเชื่อมตรง” โดยแบ่งวงจรมคูณออกเป็น 2 ส่วนคือ วงจรมคูณเชื่อมตรงทำหน้าที่คำนวณหาผลคูณตามอัลกอริทึมการคูณแบบเชื่อมตรง ซึ่งผลลัพธ์ที่ได้จากการคำนวณของวงจรมคูณเชื่อมตรงจะอยู่ในรูปของข้อมูลตัวเลขห้าชั้นที่มีเครื่องหมาย และวงจรมคูณแปลงอนเดอะฟลายจะทำหน้าที่แปลงผลลัพธ์ที่ได้นี้ให้อยู่ในรูปของข้อมูลรหัสสร้างคู่ ซึ่งวงจรมคูณอสมวารแบบเชื่อมตรงจะใช้โปรโตคอลถาดนิตินิยามแบบ 4 ชั้นตอน [16] ในการควบคุมการทำงานระหว่างขั้นตอนของวงจรมคูณ โดยการพัฒนาวงจรมคูณจะมุ่งเน้นไปที่การปรับปรุงโครงสร้างของวงจรมคูณเชื่อมตรง และสามารถแบ่งโครงสร้างได้เป็นรุ่นต่างๆดังนี้

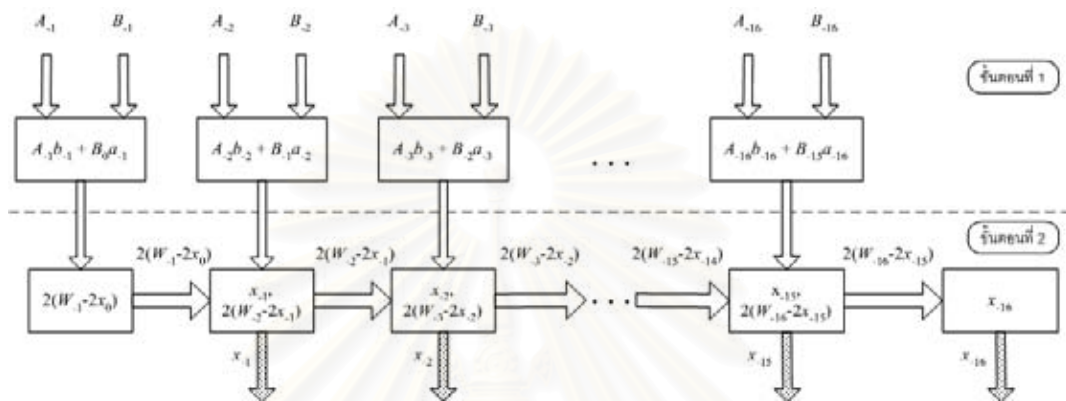


รูปที่ 3.1 โครงสร้างของวงจรมคูณอสมวารแบบเชื่อมตรง

3.1.1 โครงสร้างรุ่นที่ 1 (Version 1)

ในการออกแบบวงจรมคูณแบบเชื่อมตรงนั้น เบื้องต้นได้ทำการออกแบบตรงตามสมการที่ 2.3 และ 2.4 ซึ่งเรียกว่า โครงสร้างรุ่นที่ 1 แสดงดังรูปที่ 3.2 โดยขั้นตอนที่ 1 จะทำการหาค่าของ $A_j b_j + B_{j+1} a_j$ ซึ่งอยู่ในสมการที่ 2.3 สำหรับขั้นตอนที่ 2 จะหาค่าผลลัพธ์ของการคูณตามสมการที่ 2.4 และนำค่าของผลคูณที่ได้ไปหาค่าฟังก์ชันเศษเหลือ เพื่อส่งไปให้กับวงจรมคูณในบิตถัดไป ดังนั้นวงจรมคูณอสมวารแบบเชื่อมตรงที่ออกแบบด้วยโครงสร้างรุ่นที่ 1 จะมีขั้นตอนการทำงาน 3 ขั้นตอนคือ ขั้นตอนที่ 1 และ 2 จากรูปที่ 3.2 และวงจรมคูณแปลงอนเดอะฟลายเป็นขั้นตอนสุดท้าย

จากโครงสร้างรุ่นที่ 1 จะเห็นได้ว่า การคำนวณฟังก์ชันเศษเหลือในขั้นตอนที่ 2 น่าจะทำให้วงจรในขั้นตอนที่ 2 มีความหน่วงสูงมาก เพราะต้องรอการคำนวณหาค่าของผลลัพธ์แต่ละบิตให้เสร็จก่อนที่จะสามารถทำการคำนวณหาค่าฟังก์ชันเศษเหลือ ซึ่งจะมีขนาดใหญ่ขึ้นเมื่อจำนวนบิตที่นำมาทำการคำนวณมากขึ้น ดังนั้นทำให้เกิดแนวคิดที่จะกระจายค่าของฟังก์ชันเศษเหลือให้มีขนาดเล็กลง เพื่อลดความหน่วงในการทำงานของขั้นตอนที่ 2 ซึ่งจะอธิบายต่อไปในโครงสร้างรุ่นที่ 2



รูปที่ 3.2 โครงสร้างรุ่นที่ 1 ของวงจรมคูณเชื่อมตรง

3.1.2 โครงสร้างรุ่นที่ 2

ในโครงสร้างรุ่นที่ 2 นี้ ได้แนวคิดจากการกระจายค่าของฟังก์ชันเศษเหลือ โดยใช้เทคนิคการทำให้เป็นบรรทัดฐาน (Normalization Technique) ซึ่งสามารถแสดงได้ดังนี้

จากสมการที่ (2.3) เมื่ออยู่บนระบบเลขฐานสอง และพิจารณาที่ $j+2, j+1$ และ j

$$W_{j+2} = 2(W_{j+3} - 2x_{j+4}) + A_{j+2}b_{j+2} + B_{j+3}a_{j+2}$$

$$\begin{aligned} W_{j+1} &= 2(W_{j+2} - 2x_{j+3}) + A_{j+1}b_{j+1} + B_{j+2}a_{j+1} \\ &= 4W_{j+3} - 8x_{j+4} + 2A_{j+2}b_{j+2} + 2B_{j+3}a_{j+2} - 4x_{j+3} + A_{j+1}b_{j+1} + B_{j+2}a_{j+1} \end{aligned}$$

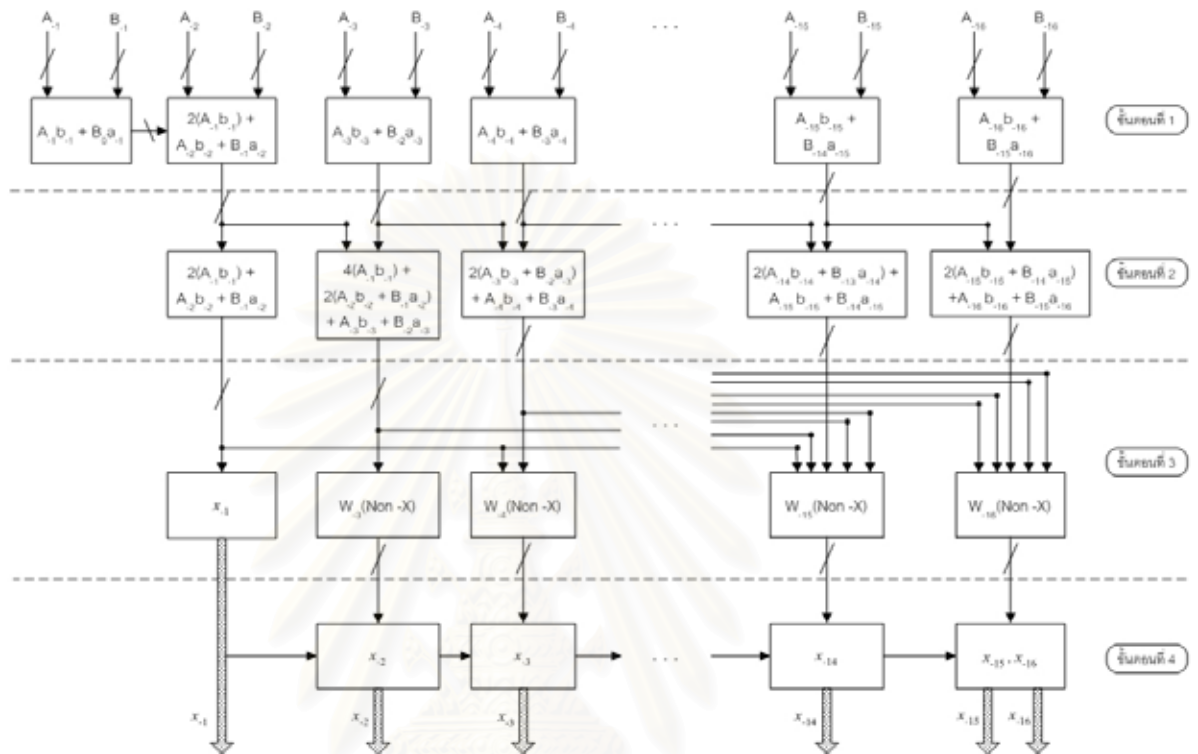
$$\begin{aligned} W_j &= 2(W_{j+1} - 2x_{j+2}) + A_j b_j + B_{j+1} a_j \\ &= 8W_{j+3} - 16x_{j+4} + 4A_{j+2}b_{j+2} + 4B_{j+3}a_{j+2} - 8x_{j+3} + 2A_{j+1}b_{j+1} + 2B_{j+2}a_{j+1} - 4x_{j+2} + \\ &\quad A_j b_j + B_{j+1} a_j \end{aligned}$$

เมื่อ $W_{j+3} = W_0 = 0$ และพจน์ของ x_{j+4} และ x_{j+3} จะมีค่าเกินจากขอบเขตของ $W_j [1,5]$ ซึ่งทำให้พจน์เหล่านั้นมีค่าเป็น 0 ดังนั้นสามารถตัดค่าเหล่านั้นออกได้ ทำให้

$$W_j = 4A_{j+2}b_{j+2} + 4B_{j+3}a_{j+2} + 2A_{j+1}b_{j+1} + 2B_{j+2}a_{j+1} + A_j b_j + B_{j+1} a_j - 4x_{j+2}$$

$$W_j = \sum_{i=\lfloor \frac{2+j}{2} \rfloor}^j 2^{i-j} (A_i b_i + B_{i+1} a_i) - 4x_{j+2} \quad (3.1)$$

เมื่อตัดพจน์ต่างๆที่ค่าเกินจากขอบเขตของ W_j จะทำให้ค่าของ i อยู่บนช่วงของ $\left[\left\lceil \frac{2+j}{2} \right\rceil, j\right]$ ดังนั้นสมการที่ (3.1) จะเป็นรูปแบบบรรทัดฐาน (Normal Form) ของสมการที่ (2.3) ซึ่งจากสมการที่ (3.1) นี้ สามารถทำการออกแบบโครงสร้างรุ่นที่ 2 ได้ดังรูปที่ 3.3



รูปที่ 3.3 โครงสร้างรุ่นที่ 2 ของวงจรมอดูมเชื่อมต่อตรง

จากการออกแบบโครงสร้างรุ่นที่ 2 คาดว่าเวลาที่ใช้ในการทำงานในแต่ละขั้นตอนจะมีความสมดุลกันมากกว่าโครงสร้างรุ่นที่ 1 เนื่องจากการทำงานในขั้นตอนที่ 4 จะมีการส่งค่า x_{j+2} เพียงบิดเดียวเท่านั้น ทำให้เวลาที่ใช้ในการทำงานน่าจะน้อยกว่าเวลาที่ใช้ในการทำงานของโครงสร้างรุ่นที่ 1 ดังนั้นวงจรมอดูมแบบเชื่อมต่อตรงที่ออกแบบด้วยโครงสร้างรุ่นที่ 2 จะแบ่งออกเป็น 5 ขั้นตอน คือขั้นตอนที่ 1 ถึง 4 จากรูปที่ 3.3 และขั้นตอนที่ 5 คือ วงจรแปลงอนเดอะฟลาย

จากรูปแบบบรรทัดฐานในสมการที่ (3.1) ทำให้ทราบถึงจำนวนพจน์ของอินพุตที่จำเป็นต่อการคำนวณ โดยจำนวนพจน์จะเพิ่มขึ้นตามจำนวนบิตของอินพุตที่เพิ่มขึ้น และสามารถนำจำนวนพจน์นี้มาคำนวณหาจำนวนขั้นตอนของวงจรมอดูมได้ ซึ่งจะพิจารณาจากการกำหนดจำนวนอินพุตในแต่ละบิตของขั้นตอนที่ 3 เป็นต้นไป เพื่อหาว่าควรกำหนดจำนวนอินพุตเป็นเท่าใดเพื่อให้ได้เวลาที่ในการทำงานของแต่ละขั้นตอนสมดุลกันมากที่สุด ซึ่งสามารถคำนวณหาจำนวนขั้นตอน

ของวงจรมีได้ดังสมการที่ (3.2) และตารางที่ 3.1 แสดงจำนวนของขั้นตอนเมื่อจำนวนบิตของอินพุตเพิ่มขึ้น

$$S = \begin{cases} \left\lceil \log_{in} \frac{4+B}{4} \right\rceil + 2 & \text{เมื่อ } \left\lceil \log_{in} \frac{4+B}{4} \right\rceil \neq 1 \\ \left\lceil \log_{in} \frac{4+B}{4} \right\rceil + 3 & \text{เมื่อ } \left\lceil \log_{in} \frac{4+B}{4} \right\rceil = 1 \end{cases} \quad (3.2)$$

เมื่อ S เป็นจำนวนขั้นตอนทั้งหมดของวงจร
 in เป็นจำนวนอินพุตในแต่ละบล็อก (ไม่รวม x_{j+2})
 B เป็นจำนวนบิตของอินพุต

ตารางที่ 3.1 จำนวนขั้นตอนของวงจรมุ่งเชื่อมต่อเมื่อออกแบบโดยใช้รูปแบบบรรทัดฐาน

จำนวน อินพุตในแต่ละบล็อก \diagdown จำนวนบิต	8	16	32	64
≤ 2	4	5	6	7
≤ 3	4	4	4	5
≤ 4	4	4	4	5
≤ 5	4	4	4	4

พิสูจน์สมการที่ (3.2)

จากสมการที่ (3.1) พจน์ที่มีค่าทั้งหมดจะอยู่ตั้งแต่ $\frac{2+j}{2}$ ถึง j และมีจำนวนพจน์ได้มากที่สุดเมื่อ $j = -B$ ดังนั้น

$$\text{จำนวนพจน์ทั้งหมด} = N = \frac{4+B}{2}$$

เมื่อผ่านขั้นตอนที่ 2

$$\text{จำนวนพจน์เมื่อผ่านขั้นตอนที่ 2} = P = \frac{N}{2} = \frac{4+B}{4}$$

เมื่อผ่านขั้นตอนถัดไปจำนวนพจน์จะลดลงดังสมการที่ (3.3)

$$P_i = \frac{P_{i-1}}{in} \quad (3.3)$$

เมื่อ i คือ ขั้นตอน และกำหนดให้ C เป็นจำนวนขั้นตอนที่เหลือ โดยมีข้อกำหนดว่า จะต้องมียุคมากกว่าหรือเท่ากับ 1 และ $C = S - 3$ จะถือว่าขั้นตอนใดๆเป็นขั้นตอนรองสุดท้ายเมื่อจำนวนพจน์ที่เหลือในขั้นตอนนั้นน้อยกว่าหรือเท่ากับจำนวนอินพุตในแต่ละบล็อกที่กำหนดไว้ ดังนั้น

$$\begin{aligned} in &\geq \frac{P}{in^c} \\ in^{c+1} &\geq P \end{aligned}$$

$$C+1 \geq \log_{in} P$$

$$C \geq \log_{in} P - 1$$

$$\therefore C = \lceil \log_{in} P \rceil - 1$$

เมื่อ $\lceil \log_{in} P \rceil = 1$ จะทำให้ค่าของ $C = 0$ ซึ่งไม่เป็นไปตามข้อกำหนดที่ได้ตั้งไว้ ดังนั้นจะต้องเพิ่มเงื่อนไข เพื่อให้เป็นไปตามข้อกำหนดที่ตั้งไว้ ดังนี้

$$C = \lceil \log_{in} P \rceil - 1 \quad \text{เมื่อ } \lceil \log_{in} P \rceil \neq 1$$

$$\text{และ } C = \lceil \log_{in} P \rceil \quad \text{เมื่อ } \lceil \log_{in} P \rceil = 1$$

เมื่อแทนค่า P จากสมการที่ (3.3) และค่า $C = S - 3$ จะได้จำนวนขั้นตอนทั้งหมดของวงจรเป็นดังสมการที่ (3.2)

3.2 การออกแบบวงจรภายใน

ในหัวข้อนี้จะอธิบายการออกแบบวงจรภายใน ซึ่งจะอธิบาย 2 ส่วนคือ การออกแบบวงจรเข้ารหัสเพื่อหาผลคูณ ตามสมการที่ (2.4) และ การออกแบบวงจรแปลงอนเดอะฟลาย

3.2.1 การออกแบบวงจรเข้ารหัสเพื่อหาผลคูณ

วงจรเข้ารหัสนี้เป็นวงจรที่ใช้ในการหาค่าของผลคูณ โดยการใช้การบิดค่าของ W_j ซึ่งในเบื้องต้นจะทำการเข้ารหัสจากสมการที่ (2.4) ดังนี้

$$\text{เมื่อ } -4 < W_j \leq -2 \quad \text{แล้ว } x_{j+1} = \bar{1} = -1$$

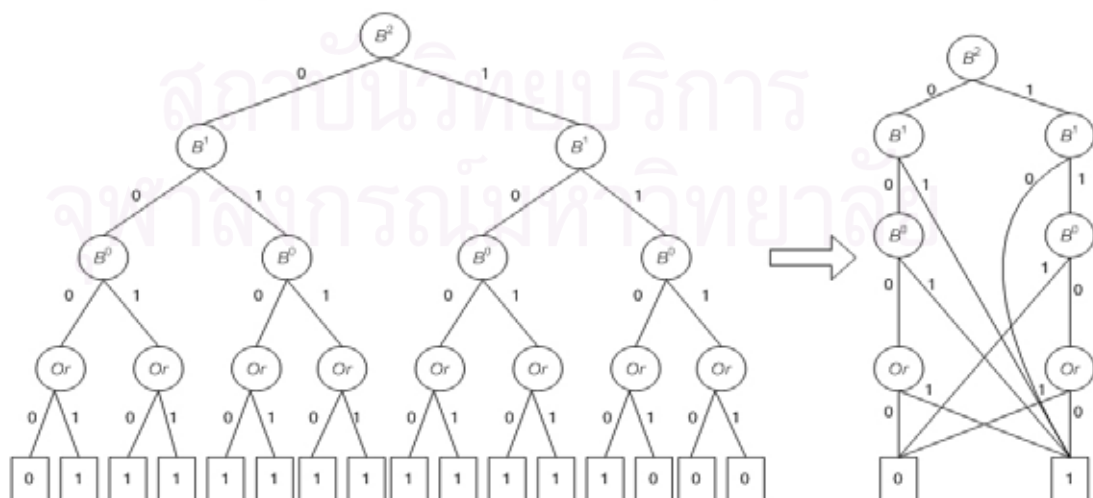
$$-2 < W_j \leq 0 \quad \text{แล้ว } x_{j+1} = 0$$

$$0 < W_j \leq 2 \quad \text{แล้ว } x_{j+1} = 1$$

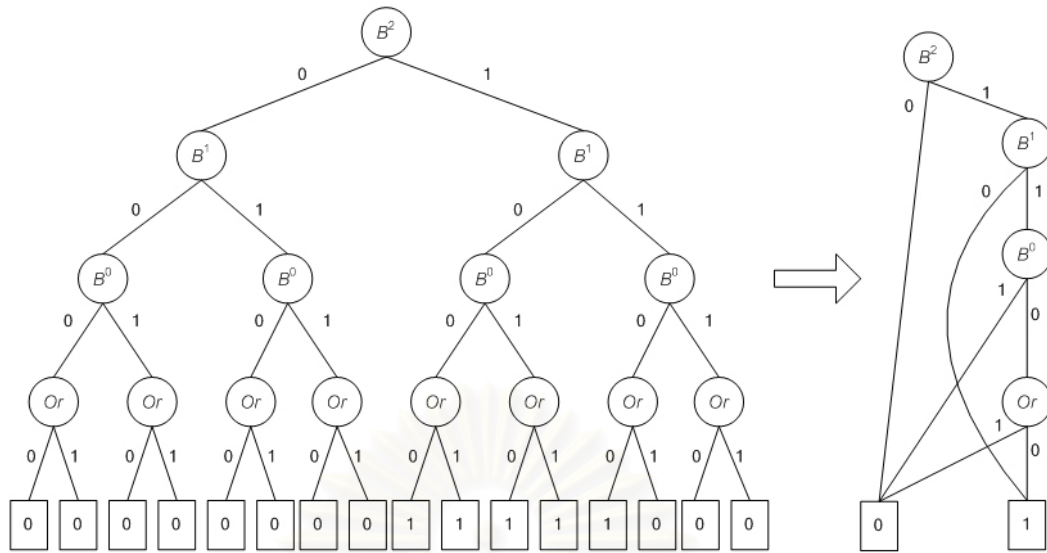
จากการเข้ารหัสนี้สามารถเขียนเป็นตารางค่าความจริง (Truth Table) ได้ดังตารางที่ 3.2 และเรียกวงจรเข้ารหัสนี้ว่า วงจรเข้ารหัสแบบที่ 1 ซึ่งค่าของ B^2 , B^1 และ B^0 ค่าของ W ตำแหน่งที่ 2 1 และ 0 ตามลำดับ และค่าของ Or คือ ค่าที่เกิดจากตรรกะหรือ (Or) ของค่า W ตำแหน่งที่ -1 จนถึง j ทั้งหมด เมื่อนำค่าความจริงจากตารางที่ 3.2 ไปออกแบบวงจร จะได้แผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับสำหรับค่าของผลคูณและเครื่องหมายของผลคูณเป็นดังรูปที่ 3.4 และ 3.5 ตามลำดับ สำหรับวงจรรหัสหาค่าของผลคูณแบบที่ 1 และวงจรรหัสหาค่าเครื่องหมายของผลคูณแบบที่ 1 แสดงดังรูปที่ 3.6 และ 3.7 ตามลำดับ

ตารางที่ 3.2 ตารางค่าความจริงของวงจรเข้ารหัสแบบที่ 1

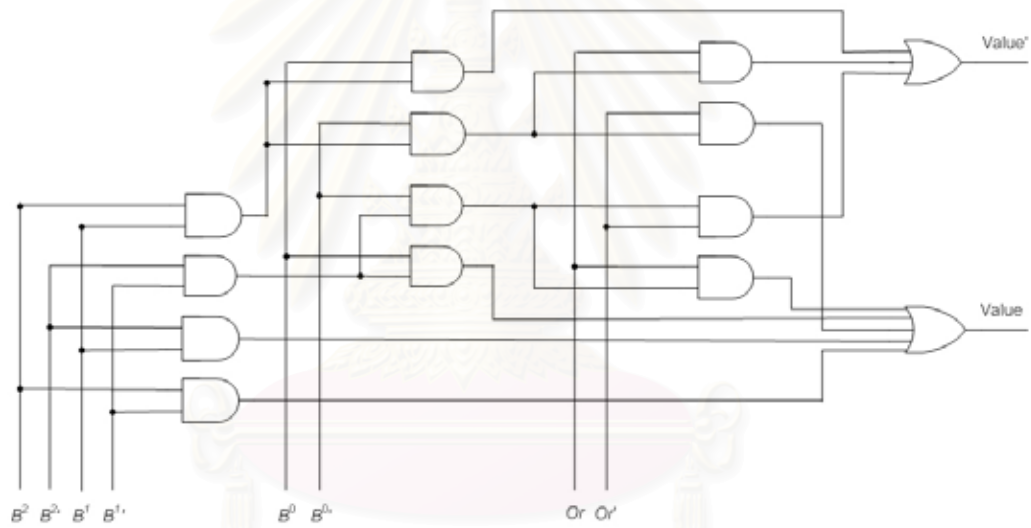
B^2	B^1	B^0	Or	เครื่องหมายของผลคูณ (Sign)	ค่าของผลคูณ (Value)
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	0	1
0	1	0	1	0	1
0	1	1	0	0	1
0	1	1	1	0	1
1	0	0	0	1	1
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	0	0
1	1	1	0	0	0
1	1	1	1	0	0



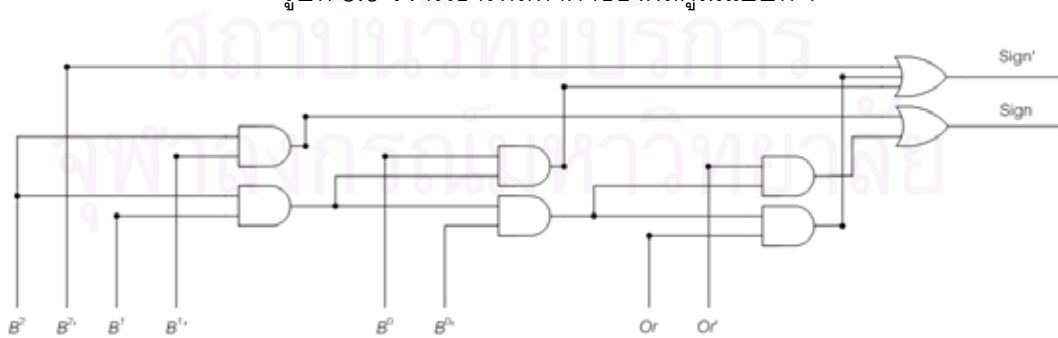
รูปที่ 3.4 แผนภาพตัดสินใจแบบทวิภาคของวงจรเข้ารหัสค่าของผลคูณแบบที่ 1



รูปที่ 3.5 แผนภาพตัดสินใจแบบทวิภาคของวงจรเข้ารหัสเครื่องหมายของผลคูณแบบที่ 1



รูปที่ 3.6 วงจรเข้ารหัสหาค่าของผลคูณแบบที่ 1



รูปที่ 3.7 วงจรเข้ารหัสหาค่าเครื่องหมายของผลคูณแบบที่ 1

เมื่อพิจารณาวงจรเข้ารหัสแบบที่ 1 จะเห็นว่า มีขนาดใหญ่ ซึ่งน่าจะทำให้มีความหน่วงสูง ทำให้เกิดแนวคิดในการปรับปรุงวงจรเข้ารหัสแบบที่ 2 ขึ้น โดยพิจารณาจากสมการที่ (2.4) ซึ่งสามารถลดตัวแปรลงได้ โดยการเลื่อน (Shift) ค่าของ W ไปทางขวา 1 บิต ดังนั้น

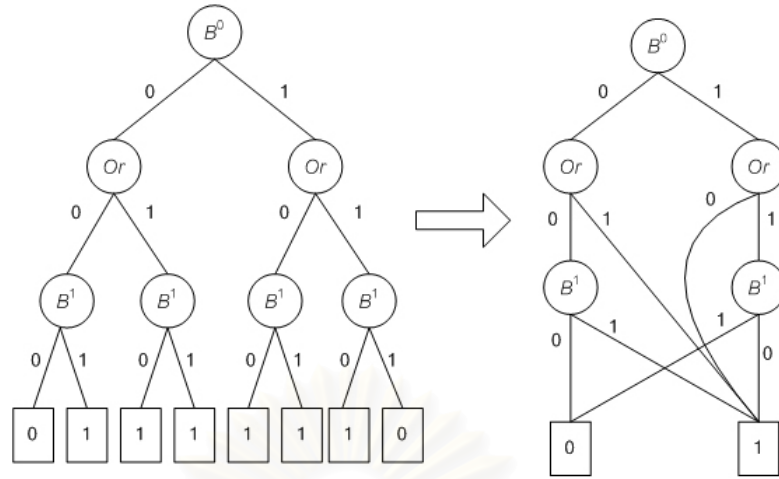
$$\begin{aligned} \text{เมื่อ } -2 < W_j \leq -1 & \quad \text{แล้ว } x_{j+1} = \bar{1} = -1 \\ -1 < W_j \leq 0 & \quad \text{แล้ว } x_{j+1} = 0 \\ 0 < W_j \leq 1 & \quad \text{แล้ว } x_{j+1} = 1 \end{aligned}$$

จากการเข้ารหัสแบบที่ 2 จะสามารถเขียนตารางค่าความจริงได้ดังตารางที่ 3.3 และสามารถนำค่าความจริงจากตารางที่ 3.3 ไปเขียนแผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับสำหรับค่าของผลคูณและเครื่องหมายของผลคูณของวงจรเข้ารหัสแบบที่ 2 เป็นดังรูปที่ 3.8 และ 3.9 ตามลำดับ และออกแบบวงจรเข้ารหัสค่าของผลคูณและวงจรเข้ารหัสเครื่องหมายของผลคูณแบบที่ 2 ได้ดังแสดงในรูปที่ 3.10 และ 3.11 ตามลำดับ

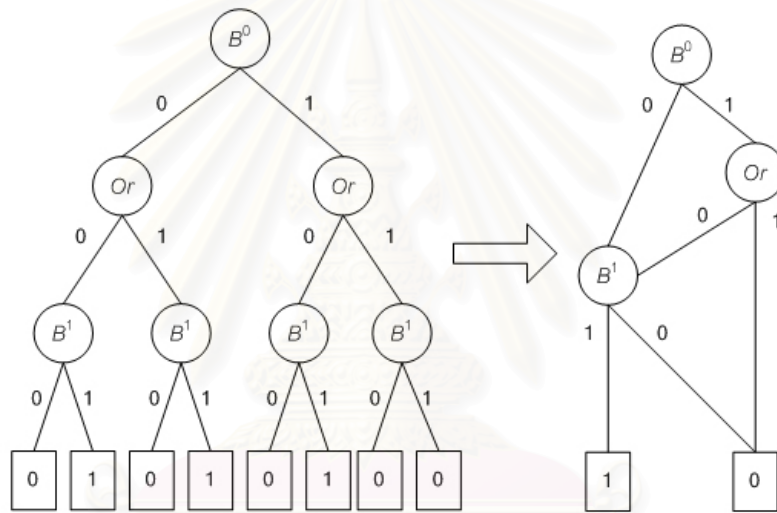
ตารางที่ 3.3 ตารางค่าความจริงของวงจรเข้ารหัสแบบที่ 2

B^0	Or	B^1	เครื่องหมายของผลคูณ (Sign)	ค่าของผลคูณ (Value)
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	1	1
1	0	0	0	1
1	0	1	1	1
1	1	0	0	1
1	1	1	0	0

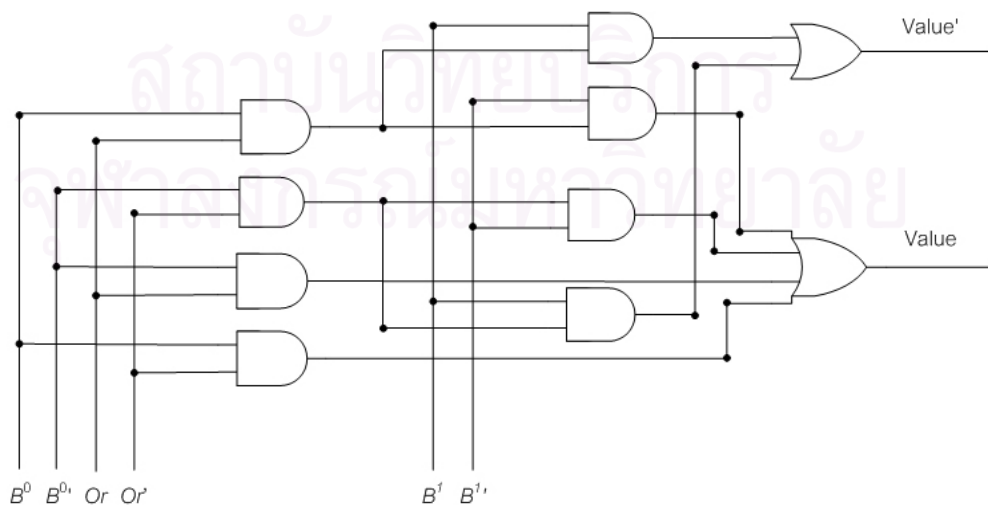
จากรูปที่ 3.10 และ 3.11 จะเห็นได้ว่า วงจรเข้ารหัสแบบที่ 2 นั้นมีขนาดเล็กกว่าแบบที่ 1 มาก ดังนั้นจะคาดคะเนได้ว่า วงจรเข้ารหัสแบบที่ 2 จะใช้เวลาในการเข้ารหัสน้อยกว่าวงจรเข้ารหัสแบบที่ 1 ซึ่งในการทดลองจะทำการเปรียบเทียบผลลัพธ์ระหว่างวงจรมอดุลาร์แบบเชื่อมตรงที่ออกแบบโดยใช้วงจรเข้ารหัสแบบที่ 1 และ 2 ว่าเป็นไปตามความคาดหมายหรือไม่



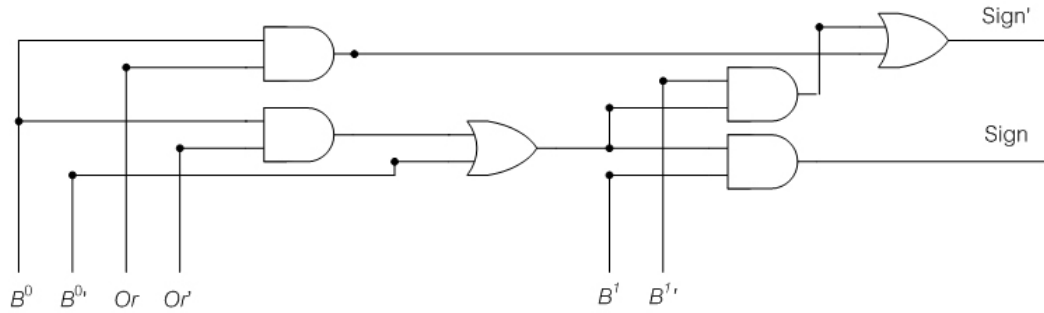
รูปที่ 3.8 แผนภาพตัดสินใจแบบทวิภาคของวงจรเข้ารหัสค่าของผลคูณแบบที่ 2



รูปที่ 3.9 แผนภาพตัดสินใจแบบทวิภาคของวงจรเข้ารหัสเครื่องหมายของผลคูณแบบที่ 2



รูปที่ 3.10 วงจรเข้ารหัสค่าของผลคูณแบบที่ 2



รูปที่ 3.11 วงจรเข้ารหัสหาค่าเครื่องหมายของผลคูณแบบที่ 2

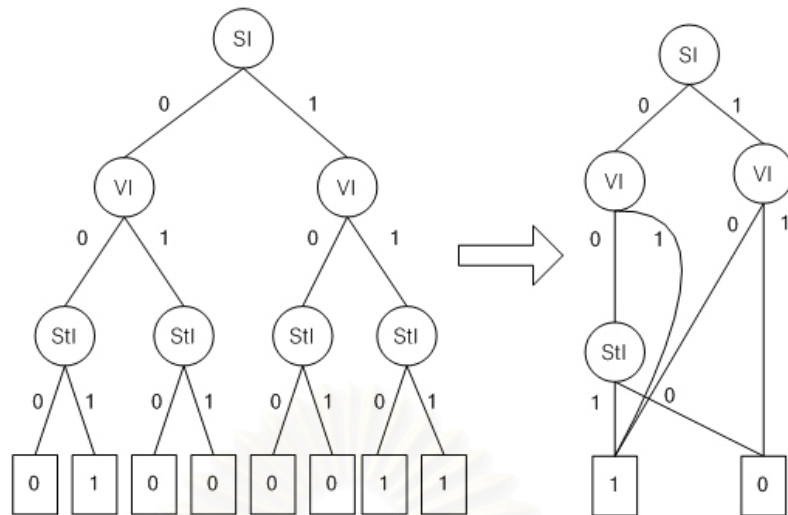
3.2.2 การออกแบบวงจรแปลงอนเดอะฟลาย

ในวงจรคูณอสมวารแบบเชื่อมตรง จำเป็นจะต้องมีวงจรอนเดอะฟลาย เพื่อแปลงค่าเอาต์พุตของวงจรคูณเชื่อมตรง ซึ่งอยู่ในรูปของข้อมูลตัวเลขซ้ำซ้อนที่มีเครื่องหมาย ให้เป็นข้อมูลรหัสรางคู่เพื่อให้สามารถนำวงจรคูณอสมวารแบบเชื่อมตรงนี้ไปใช้ร่วมกับไมโครโปรเซสเซอร์แบบอสมวารได้ โดยจากอโตเมตาในรูปที่ 2.10 สามารถเขียนเป็นตารางค่าความจริงได้ดังตารางที่ 3.4

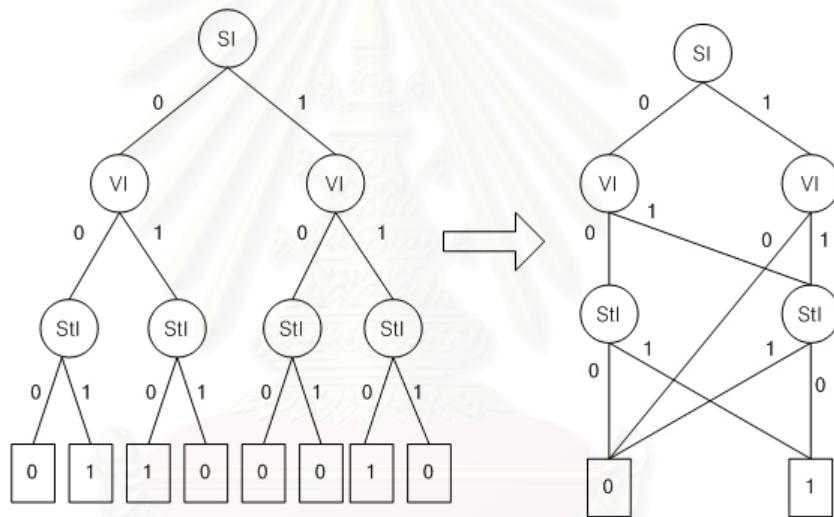
ตารางที่ 3.4 ตารางค่าความจริงของการแปลงตัวเลขแบบอนเดอะฟลาย

เครื่องหมายขาเข้า (Sign Input : SI)	ค่าผลคูณขาเข้า (Value Input : VI)	สถานะขาเข้า (State Input : StI)	สถานะขาออก (State Output : SO)	ข้อมูลขาออก (Data Output : DO)
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	0	0
1	0	1	0	0
1	1	0	1	1
1	1	1	1	0

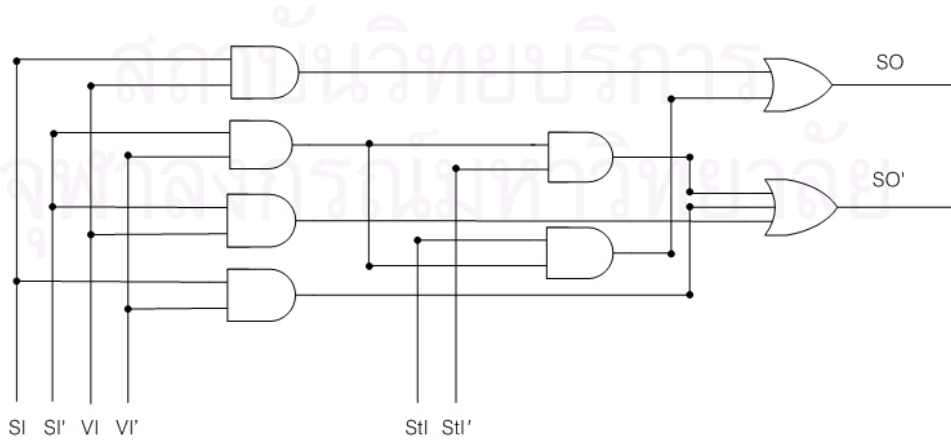
สถานะ a และ b จากอโตเมตาในรูปที่ 2.10 แทนค่าสถานะขาเข้า 0 และ 1 ในตารางที่ 3.4 ซึ่งสามารถนำข้อมูลจากตารางนี้ไปสร้างแผนภาพตัดสินใจแบบทวิภาคชนิดมีการลตทอนอันดับ ได้ดังรูปที่ 3.12 และ 3.13 และออกแบบวงจรได้ดังรูปที่ 3.14 และ 3.15



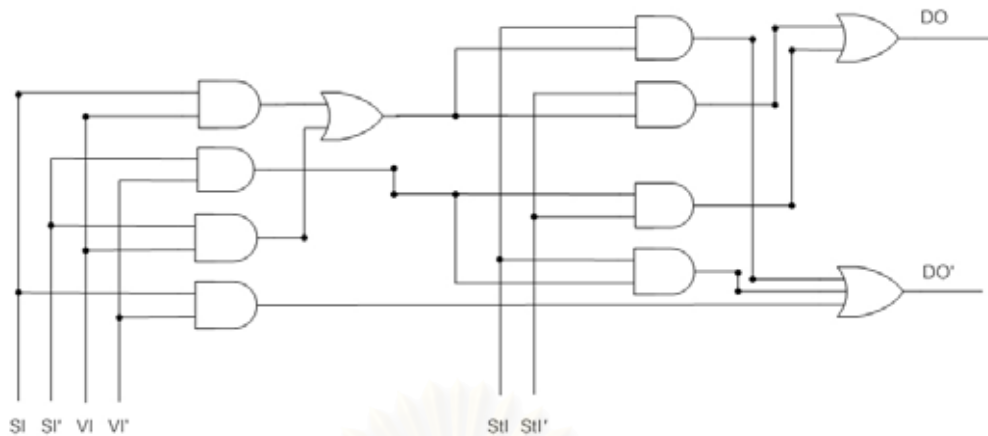
รูปที่ 3.12 แผนภาพตัดสินใจแบบทวิภาคเพื่อหาค่าสถานะขาออกของวงจรแปลงอนเดอะฟลาย



รูปที่ 3.13 แผนภาพตัดสินใจแบบทวิภาคเพื่อหาค่าข้อมูลขาออกของวงจรแปลงอนเดอะฟลาย



รูปที่ 3.14 วงจรหาค่าสถานะขาออกของการแปลงตัวเลขแบบอนเดอะฟลาย



รูปที่ 3.15 วงจรหาค่าข้อมูลขาออกของการแปลงตัวเลขแบบอนเดอะฟลาย

สรุป

งานวิจัยนี้ทำการออกแบบและพัฒนาวงจรมอดูลสมวารแบบเชื่อมตรง โดยการออกแบบจะแบ่งเป็นโครงสร้างรุ่นที่ 1 และโครงสร้างรุ่นที่ 2 ซึ่งโครงสร้างทั้ง 2 แบบจะใช้โปรโตคอลอานัติสัญญาณแบบ 4 ขั้นตอนในการควบคุมการทำงานระหว่างขั้นตอนภายในวงจรมอดูล โดยมีสมมติฐานว่า โครงสร้างรุ่นที่ 2 จะให้อัตราปริมาณงาน (Throughput Rate) ดีกว่าโครงสร้างรุ่นที่ 1 และวงจรถ่ายรหัสแบบที่ 2 จะใช้เวลาในการทำงานน้อยกว่าวงจรถ่ายรหัสแบบที่ 1 ซึ่งในบทที่ 4 จะเป็นการนำเสนอการทดลองและผลของการทดลองเบื้องต้น เพื่อตรวจสอบสมมติฐานว่าถูกต้องหรือไม่

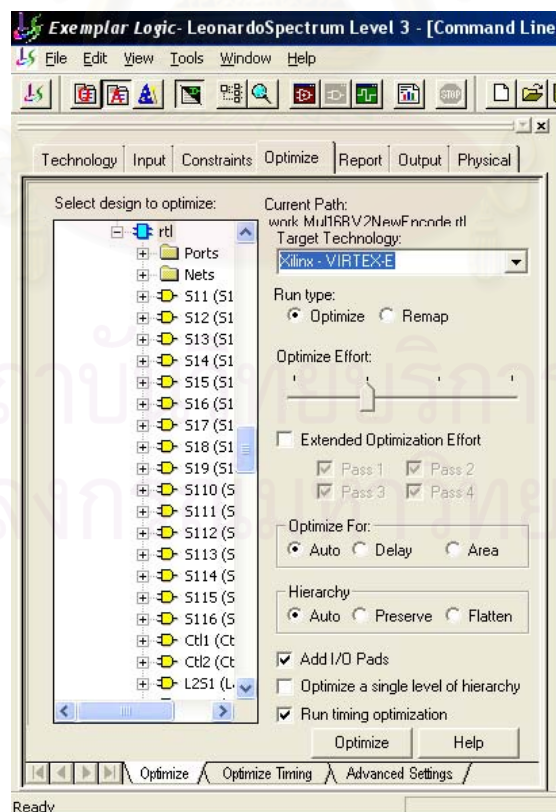
บทที่ 4

การทดสอบวงจร

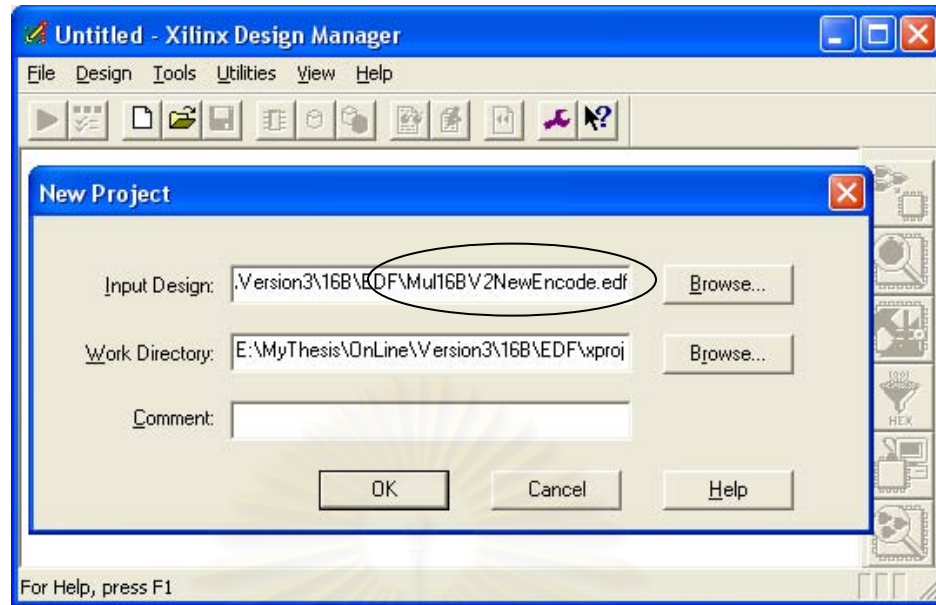
หลังจากที่ได้ทำการออกแบบวงจรคุณ โดยใช้โครงสร้างรุ่นที่ 1 และโครงสร้างรุ่นที่ 2 แล้ว จะทำการจำลองการทำงาน เพื่อตรวจสอบความถูกต้องและแก้ไขข้อบกพร่องของวงจร ก่อนที่จะนำไปสังเคราะห์บนเอฟพีจีเอ XCV1000Efg1156-6 หลังจากนั้นจะทำการจำลองการทำงานแบบมีความหน่วงจากวงจรระดับสังเคราะห์บนเอฟพีจีเอ ดังนั้นในบทนี้จะอธิบายถึงขั้นตอนการสร้างวงจรระดับสังเคราะห์บนเอฟพีจีเอ ผลการจำลองการทำงานแบบมีความหน่วงของวงจรคุณ อสมวารแบบเชื่อมตรง และผลการเปรียบเทียบการทำงานของวงจรคุณอสมวารแบบเชื่อมตรง

4.1 การสร้างวงจรระดับสังเคราะห์บนเอฟพีจีเอ

ในงานวิจัยนี้ใช้ภาษาวีเอชดีแอลในการอธิบายลักษณะของวงจร ดังนั้นเมื่อเขียนโปรแกรมอธิบายวงจรเสร็จแล้ว จะนำมาสังเคราะห์โดยใช้โปรแกรม Leonardo Spectrum [6] ดังแสดงในรูปที่ 4.1 จากนั้นจะนำไฟล์สกุล .EDF ที่ได้ไปสร้างวงจรระดับสังเคราะห์บนเอฟพีจีเอ โดยใช้โปรแกรม Design Manager ซึ่งเป็นโปรแกรมย่อยของโปรแกรม Xilinx Foundation Series 3.1 ดังแสดงในรูปที่ 4.2

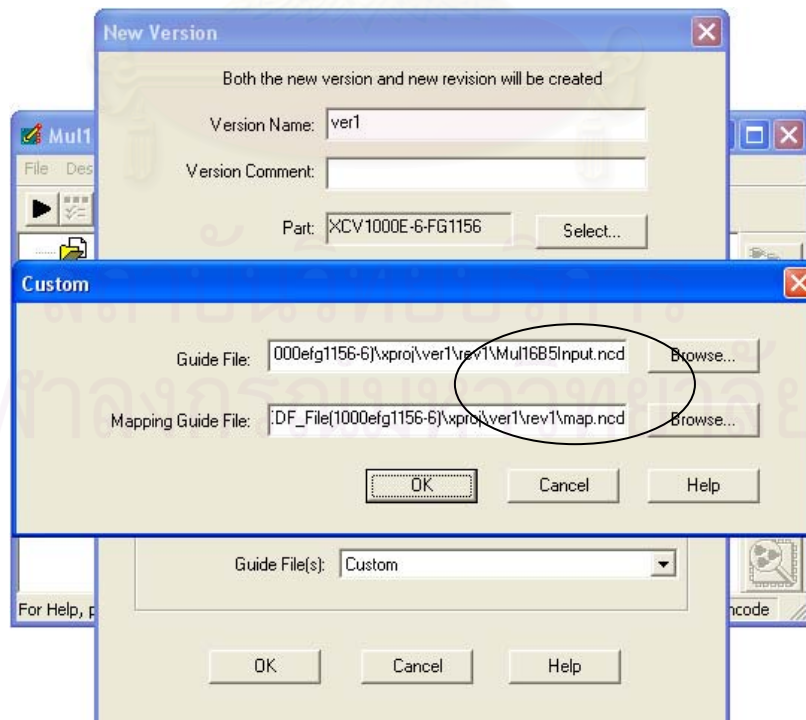


รูปที่ 4.1 การสังเคราะห์วงจรโดยใช้โปรแกรม Leonardo Spectrum



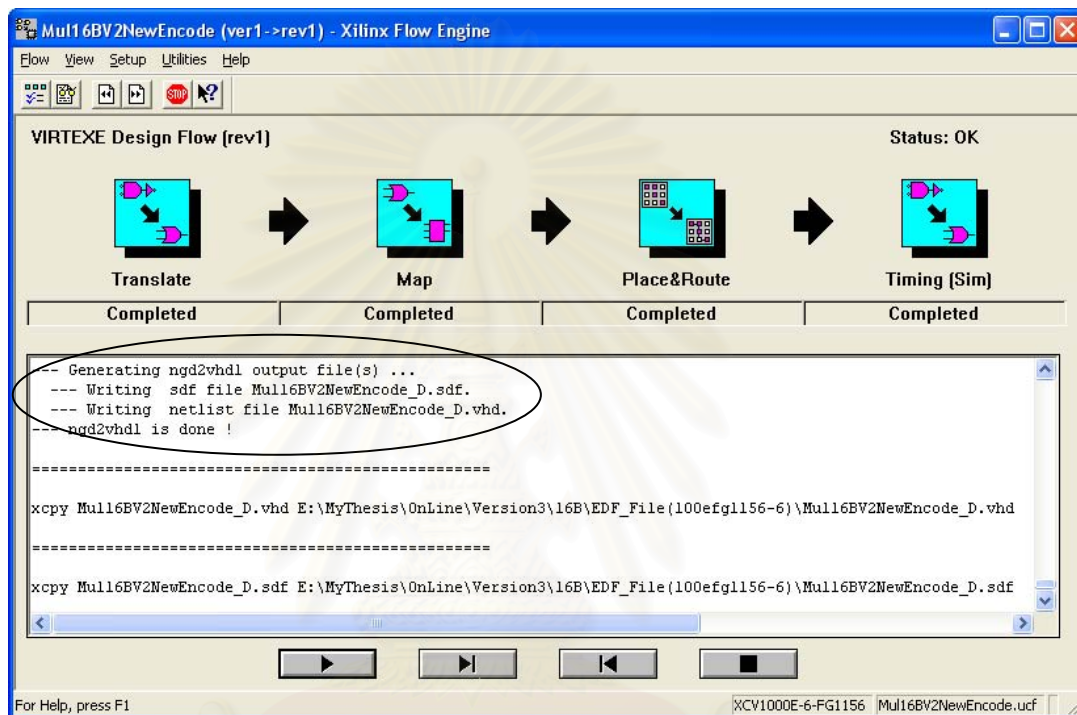
รูปที่ 4.2 การสร้างวงจรระดับสังเคราะห์บนเอฟพีจีเอโดยใช้โปรแกรม Design Manager

เมื่อสร้างวงจรระดับสังเคราะห์เสร็จแล้ว หากต้องการแก้ไขข้อบกพร่องบางส่วนของวงจรที่สามารถที่จะทำได้โดยไม่ต้องทำการเพลสและเราต์ (Place and Route) วงจรใหม่ทั้งหมด โดยการกำหนดไฟล์นำทาง (Guide File) ให้กับวงจรที่จะทำการสร้างใหม่ดังรูปที่ 4.3 เพราะการกำหนดไฟล์นำทางนั้นจะทำให้โปรแกรม Design Manager ทำการเพลสและเราต์กับวงจรเฉพาะส่วนที่เกิดการเปลี่ยนแปลงขึ้นเท่านั้น ซึ่งสามารถใช้วิธีนี้ในการสร้างส่วนวงจรตอบรับของวงจรได้ด้วย



รูปที่ 4.3 การกำหนดไฟล์นำทางให้กับวงจรที่จะทำการสร้างใหม่

เมื่อทำการสร้างวงจรระดับสังเคราะห์บนเฟิร์มแวร์แล้ว โปรแกรม Design Manager จะสร้างไฟล์สกุล .VHD และไฟล์สกุล .SDF ดังแสดงในรูปที่ 4.4 เพื่อให้สามารถนำไฟล์ข้อมูลที่ได้นี้ไปจำลองการทำงานแบบมีความหน่วงก่อนที่จะนำไปทดสอบบนบอร์ดจริง (กรณีที่ต้องทำการทดสอบบนบอร์ด) เมื่อได้ไฟล์สกุล .VHD และไฟล์สกุล .SDF จะสามารถทำการจำลองการทำงานแบบมีความหน่วงได้ ซึ่งในหัวข้อถัดไปจะกล่าวถึงผลการทดลองจากการจำลองการทำงานแบบมีความหน่วง



รูปที่ 4.4 การสร้างไฟล์สกุล .VHD และไฟล์สกุล .SDF ของโปรแกรม Design Manager

4.2 ผลการจำลองการทำงานของวงจรคุณสมบัตินแบบเชื่อมต่อตรง

ในงานวิจัยนี้จะใช้โปรแกรม Model Sim [14] ในการจำลองการทำงานของวงจรคุณสมบัตินแบบเชื่อมต่อตรง โดยวงจรที่นำมาทดสอบ จะออกแบบโดยใช้โครงสร้างรุ่นที่ 1 และโครงสร้างรุ่นที่ 2 รวมทั้งใช้วงจรเข้ารหัสแบบที่ 1 และแบบที่ 2 กับการออกแบบโครงสร้างรุ่นที่ 2 โดยวงจรที่ออกแบบจะมี 2 ขนาด คือ ขนาด 8 บิตและขนาด 16 บิต ซึ่งวงจรคุณขนาด 8 บิต จะแบ่งออกเป็น 2 วงจรคือ วงจรที่มีจำนวนอินพุตตั้งแต่ขั้นตอนที่ 3 เป็นต้นไปไม่เกิน 3 และไม่เกิน 2 โดยการทดลองจะทำการสุ่มค่าอินพุต 40 คู่ เพื่อนำมาหาค่าของผลคูณ ซึ่งจะผลคูณมีขนาดเท่ากับขนาดของอินพุตที่ป้อนเข้าไปในวงจร และผลการทดลองจะประกอบด้วย จำนวนเกตที่ใช้ในการสร้างวงจร เวลาที่ใช้ในการทำงาน ซึ่งเป็นเวลาเฉลี่ย และอัตราปริมาณงาน ดังแสดงในตารางที่ 4.1

ตารางที่ 4.1 ผลการทดลองของวงจรรูปร่างสมมาตรแบบเชื่อมตรง

วงจรรูปร่าง	จำนวนเกต (เกต)	เวลาที่ใช้ในการทำงาน (นาโนวินาที : ns)	อัตราปริมาณงาน (เมกะเฮิรตซ์ : MHz)
วงจรรูปร่างขนาด 8 บิต โดยใช้โครงสร้างรุ่นที่ 1 และใช้วงจรรูปร่างแบบที่ 1	8,577	118.13	15.67
วงจรรูปร่างขนาด 8 บิต โดยใช้โครงสร้างรุ่นที่ 2 ใช้วงจรรูปร่างแบบที่ 1 และกำหนดจำนวนอินพุต ≤ 3	12,759	160.15	30.36
วงจรรูปร่างขนาด 8 บิต โดยใช้โครงสร้างรุ่นที่ 2 ใช้วงจรรูปร่างแบบที่ 1 และกำหนดจำนวนอินพุต ≤ 2	13,092	176.15	27.18
วงจรรูปร่างขนาด 8 บิต โดยใช้โครงสร้างรุ่นที่ 2 ใช้วงจรรูปร่างแบบที่ 2 และกำหนดจำนวนอินพุต ≤ 3	12,864	148.03	35.59
วงจรรูปร่างขนาด 16 บิต โดยใช้โครงสร้างรุ่นที่ 1 และใช้วงจรรูปร่างแบบที่ 1	30,792	200.60	11.30
วงจรรูปร่างขนาด 16 บิต โดยใช้โครงสร้างรุ่นที่ 2 ใช้วงจรรูปร่างแบบที่ 1 และกำหนดจำนวนอินพุต ≤ 5	50,133	250.50	17.26
วงจรรูปร่างขนาด 16 บิต โดยใช้โครงสร้างรุ่นที่ 2 ใช้วงจรรูปร่างแบบที่ 2 และกำหนดจำนวนอินพุต ≤ 5	50,772	189.00	27.15

ในการทดลองได้ทำการเก็บผลเวลาที่วงจรรูปร่างที่ใช้ในแต่ละขั้นตอน โดยตารางที่ 4.2 แสดงเวลาที่วงจรรูปร่างซึ่งออกแบบด้วยโครงสร้างรุ่นที่ 1 ใช้ในแต่ละขั้นตอน โดยแบ่งเป็น 3 ขั้นตอน และตารางที่ 4.3 แสดงเวลาที่วงจรรูปร่างซึ่งออกแบบด้วยโครงสร้างรุ่นที่ 2 ใช้ในแต่ละขั้นตอน และแบ่งเป็น 5 ขั้นตอน

ตารางที่ 4.2 เวลาที่ใช้ในแต่ละขั้นตอนของวงจรรูปร่างแบบเชื่อมตรง
ซึ่งออกแบบด้วยโครงสร้างรุ่นที่ 1

วงจรรูปร่าง	ขั้นตอนที่ 1 (ns)	ขั้นตอนที่ 2 (ns)	ขั้นตอนที่ 3 (ns)
วงจรรูปร่างขนาด 8 บิต	20.28	64.23	28.90
วงจรรูปร่างขนาด 16 บิต	31.60	135.35	30.60

ตารางที่ 4.3 เวลาที่ใช้ในแต่ละขั้นตอนของวงจรรูปร่างแบบเชื่อมตรง
ซึ่งออกแบบด้วยโครงสร้างรุ่นที่ 2

วงจรรูปร่าง	ขั้นตอน ที่ 1 (ns)	ขั้นตอน ที่ 2 (ns)	ขั้นตอน ที่ 3 (ns)	ขั้นตอน ที่ 4 (ns)	ขั้นตอน ที่ 5 (ns)
วงจรรูปร่างขนาด 8 บิต วงจรรูปร่างแบบที่ 1 จำนวนอินพุต ≤ 3	24.78	22.73	30.18	44.50	32.00
วงจรรูปร่างขนาด 8 บิต วงจรรูปร่างแบบที่ 1 จำนวนอินพุต ≤ 2	25.03	21.58	29.85	62.63	32.30
วงจรรูปร่างขนาด 8 บิต วงจรรูปร่างแบบที่ 2 จำนวนอินพุต ≤ 3	24.78	22.70	30.20	32.38	31.98
วงจรรูปร่างขนาด 16 บิต วงจรรูปร่างแบบที่ 1 จำนวนอินพุต ≤ 5	28.25	30.80	43.43	116.13	27.88
วงจรรูปร่างขนาด 16 บิต วงจรรูปร่างแบบที่ 2 จำนวนอินพุต ≤ 5	24.65	31.68	41.13	54.83	31.78

จากผลการทดลองในตารางที่ 4.1 วงจรคุณขนาด 8 บิตที่กำหนดจำนวนอินพุตไม่เกิน 3 จะมีค่าอัตราปริมาณงานมากกว่าวงจรคุณขนาด 8 บิตที่กำหนดจำนวนอินพุตไม่เกิน 2 ทั้งนี้ เนื่องจากการกำหนดจำนวนอินพุตให้น้อยลงจะเป็นการลดภาระการทำงานไปให้กับขั้นตอนที่ 4 ที่ใช้เวลาในการทำงานสูงอยู่แล้ว สามารถสังเกตได้เวลาที่ใช้ในการทำงานของขั้นตอนที่ 4 จาก ตารางที่ 4.3 ซึ่งวงจรคุณขนาด 8 บิตที่กำหนดจำนวนอินพุตไม่เกิน 3 จะใช้เวลาในขั้นตอนนี้้น้อยกว่าวงจรคุณขนาด 8 บิตที่กำหนดจำนวนอินพุตไม่เกิน 2 สำหรับผลการจำลองการทำงานอย่างละเอียดของวงจรคุณที่ออกแบบในงานวิจัยนี้ จะแสดงในภาคผนวก ก

4.3 ผลการเปรียบเทียบการทำงานของวงจรคุณสมวารแบบเชื่อมต่อตรง

งานวิจัยนี้จะเปรียบเทียบวงจรคุณสมวารแบบเชื่อมต่อตรง ซึ่งออกแบบด้วยโครงสร้างที่ต่างกัน ในด้านขนาด เวลาที่ใช้ในการทำงานของแต่ละขั้นตอนและทั้งวงจร และอัตราปริมาณงานของวงจร ดังนี้

4.3.1 ขนาดของวงจร

จากตารางที่ 4.1 วงจรคุณที่ออกแบบด้วยโครงสร้างรุ่นที่ 1 จะมีขนาดเล็กกว่าวงจรคุณที่ออกแบบด้วยโครงสร้างรุ่นที่ 2 คิดเป็น 32.78 - 34.49 เปอร์เซ็นต์ สำหรับวงจรคุณขนาด 8 บิต และคิดเป็น 38.58 - 39.35 เปอร์เซ็นต์ สำหรับวงจรคุณขนาด 16 บิต ทั้งนี้เนื่องจากวงจรคุณที่ออกแบบด้วยโครงสร้างรุ่นที่ 2 นั้นมีขั้นตอนการทำงานที่มากกว่า ทำให้ต้องใช้เกตในการสร้างแลตช์ (Latch) และวงจรควบคุมมากกว่าวงจรคุณที่ออกแบบด้วยโครงสร้างรุ่นที่ 1

4.3.2 เวลาที่ใช้ในการทำงาน

จากตารางที่ 4.1 พบว่า วงจรคุณขนาด 8 บิต ที่ออกแบบด้วยโครงสร้างรุ่นที่ 1 จะใช้เวลาที่ใช้ในการทำงานน้อยกว่าวงจรคุณขนาด 8 บิต ที่ออกแบบด้วยโครงสร้างรุ่นที่ 2 ทั้งวงจรคุณที่ใช้วงจรเข้ารหัสแบบที่ 1 และแบบที่ 2 คิดเป็น 20.20 - 32.94 เปอร์เซ็นต์ และเมื่อเปรียบเทียบวงจรคุณขนาด 16 บิต พบว่า วงจรคุณที่ออกแบบด้วยโครงสร้างรุ่นที่ 1 จะใช้เวลาที่ใช้ในการทำงานน้อยกว่าวงจรคุณที่ออกแบบด้วยโครงสร้างรุ่นที่ 2 และใช้วงจรเข้ารหัสแบบที่ 1 คิดเป็น 19.92 เปอร์เซ็นต์ แต่จะใช้เวลาในการทำงานมากกว่าวงจรคุณที่ออกแบบด้วยโครงสร้างรุ่นที่ 2 และใช้วงจรเข้ารหัสแบบที่ 2 คิดเป็น 6.14 เปอร์เซ็นต์

จากตารางที่ 4.2 และ 4.3 สังเกตได้ว่า วงจรคุณที่ออกแบบด้วยโครงสร้างรุ่นที่ 2 จะใช้เวลาการทำงานในแต่ละขั้นตอนใกล้เคียงกัน มากกว่าวงจรคุณที่ออกแบบด้วยโครงสร้างรุ่นที่ 1

และ ในขั้นตอนที่ 4 ของวงจรคุณที่ใช้วงจรเข้ารหัสแบบที่ 2 จะใช้เวลาในการทำงานน้อยกว่า วงจรคุณที่ใช้วงจรเข้ารหัสแบบที่ 1 เพราะวงจรเข้ารหัสแบบที่ 2 มีขนาดเล็กกว่าวงจรเข้ารหัสแบบที่ 1 และแผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับ มีการเรียงตัวแปรที่ดีต่อการสร้างวงจรมากกว่า ทำให้วงจรคุณที่ใช้วงจรเข้ารหัสแบบที่ 2 ใช้เวลาในการทำงานน้อยกว่าวงจรคุณที่ใช้วงจรเข้ารหัสแบบที่ 1 โดยเฉพาะวงจรคุณขนาด 16 บิตที่ออกแบบด้วยโครงสร้างรุ่นที่ 2 และใช้วงจรเข้ารหัสแบบที่ 2 สามารถทำงานได้เร็วกว่าวงจรคุณขนาดเท่ากัน ซึ่งออกแบบด้วยโครงสร้างรุ่นที่ 1

4.3.3 อัตราปริมาณงาน

จากตารางที่ 4.1 วงจรคุณขนาด 8 บิตที่ออกแบบด้วยโครงสร้างรุ่นที่ 2 และใช้วงจรเข้ารหัสแบบที่ 1 สามารถให้อัตราปริมาณงานมากกว่าวงจรคุณขนาด 8 บิตที่ออกแบบด้วยโครงสร้างรุ่นที่ 1 ถึง 93.75 เปอร์เซ็นต์ เนื่องจากโครงสร้างรุ่นที่ 1 จะใช้เวลาเป็นอย่างมากในการหาค่าของฟังก์ชันเศษเหลือ ซึ่งโครงสร้างรุ่นที่ 2 ได้ทำการปรับปรุงโดยใช้เทคนิคการทำให้เป็นบรรทัดฐานดังที่ได้อธิบายแล้วในบทที่ 3 และวงจรคุณขนาด 8 บิตที่ออกแบบด้วยโครงสร้างรุ่นที่ 2 และใช้วงจรเข้ารหัสแบบที่ 2 สามารถให้อัตราปริมาณงานได้สูงยิ่งกว่าวงจรคุณที่ออกแบบด้วยโครงสร้างรุ่นที่ 2 และใช้วงจรเข้ารหัสแบบที่ 1 เป็น 17.23 เปอร์เซ็นต์

สำหรับวงจรคุณขนาด 16 บิตที่ออกแบบด้วยโครงสร้างรุ่นที่ 2 และใช้วงจรเข้ารหัสแบบที่ 1 จะให้อัตราปริมาณงานสูงกว่าวงจรคุณขนาด 16 บิตที่ออกแบบด้วยโครงสร้างรุ่นที่ 1 ถึง 52.74 เปอร์เซ็นต์ และเมื่อเปรียบเทียบกับวงจรคุณขนาด 16 บิตที่ออกแบบด้วยโครงสร้างรุ่นที่ 2 และใช้วงจรเข้ารหัสแบบที่ 2 จะให้อัตราปริมาณงานสูงกว่าวงจรคุณขนาด 16 บิตที่ออกแบบด้วยโครงสร้างรุ่นที่ 2 และใช้วงจรเข้ารหัสแบบที่ 1 และวงจรคุณขนาด 16 บิตที่ออกแบบด้วยโครงสร้างรุ่นที่ 1 ถึง 57.30 และ 140.27 เปอร์เซ็นต์ ตามลำดับ

สรุป

งานวิจัยนี้ได้ทำการออกแบบและทดสอบวงจรคุณอสมวารแบบเชิงสมตรง โดยได้ทำการออกแบบโครงสร้างเป็น 2 รุ่น และใช้วงจรเข้ารหัส 2 แบบ ซึ่งจากผลการทดลองพบว่า เป็นไปตามสมมติฐานที่ตั้งไว้ในบทที่ 3 และเมื่อออกแบบวงจรคุณด้วยโครงสร้างรุ่นที่ 2 และใช้วงจรเข้ารหัสแบบที่ 2 จะให้อัตราปริมาณงานสูงขึ้น แม้ว่าเมื่อขนาดของอินพุตจะเพิ่มขึ้นแล้วทำให้ขนาดของวงจรมีขนาดใหญ่ขึ้นมากก็ตาม แต่อัตราปริมาณงานที่ได้ก็มีค่าอยู่ในเกณฑ์ที่ดี ดังนั้นวงจรคุณอสมวารที่ได้ออกแบบนี้ เหมาะสมแก่การเป็นอีกทางเลือกหนึ่งในการนำวงจรไปใช้ร่วมกับไมโครโปรเซสเซอร์แบบอสมวารหรือระบบประมวลผลเชิงเลขอื่นๆ

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 สรุปผลการวิจัย

วัตถุประสงค์หลักของงานวิจัยนี้คือ นำเสนอแนวทางการออกแบบวงจรควบคุมแบบอสมวารโดยใช้อัลกอริทึมการควบคุมแบบเชื่อมตรง เพื่อเป็นแนวทางเริ่มต้นในการสร้างวงจรควบคุมแบบอสมวารขึ้นมาใช้งานร่วมกับไมโครโปรเซสเซอร์แบบอสมวารที่ไม่มีวงจรคุมได้

จากการศึกษาการออกแบบวงจรแบบอสมวารเบื้องต้นพบว่า โครงสร้างแบบไปป์ไลน์ขนาน (Parallel Pipeline) นั้นสามารถให้อัตราปริมาณงานได้ดี และใช้เวลาในการทำงานไม่สูงมากนัก ดังนั้นงานวิจัยนี้ได้เลือกโครงสร้างนี้มาทำการออกแบบและสร้างวงจรควบคุมอสมวารแบบเชื่อมตรง ซึ่งสิ่งที่ต้องคำนึงถึงในการออกแบบคือ ความสมดุลของเวลาที่ใช้ในการทำงานของแต่ละขั้นตอน เพราะแต่ละขั้นตอนของวงจรไปป์ไลน์แบบอสมวารที่ออกแบบโดยใช้รหัสรางคู่ จะต้องทำการกำหนดค่าเริ่มต้นทุกครั้งที่เราเริ่มการทำงาน ดังนั้นความสมดุลของเวลานี้จะช่วยไม่ให้อัตราปริมาณงานลดลง

เมื่อทำการออกแบบและทำการจำลองการทำงานของวงจรคุมที่ได้ออกแบบแล้ว พบว่าโครงสร้างที่ออกแบบนั้นสามารถทำงานได้ดีพอสมควร แม้ว่าวงจรจะมีขนาดใหญ่ก็ตาม และสิ่งที่ได้ค้นพบอีกประการหนึ่งคือ วงจรที่ออกแบบด้วยโครงสร้างรุ่นที่ 2 และใช้วงจรเข้ารหัสแบบที่ 2 เมื่อเพิ่มขนาดของอินพุตขึ้น (จาก 8 บิตเป็น 16 บิต) เวลาที่ใช้ในการทำงานจะไม่เพิ่มขึ้นเป็น 2 เท่า และอัตราปริมาณงานที่ได้ไม่ลดลงมากนัก ดังนั้นน่าจะสามารรถนำโครงสร้างที่ได้นี้ไปใช้กับการออกแบบวงจรคุมที่มีอินพุตขนาดใหญ่ เช่น 32 บิต หรือ 53 บิต เป็นต้น

จากโครงสร้างรุ่นที่ 2 ซึ่งออกแบบได้นั้น สามารถทำการกำหนดจำนวนอินพุตตั้งแต่ขั้นตอนที่ 3 เป็นต้นไปได้ และเมื่อเกิดการเปลี่ยนแปลงขึ้นกับจำนวนอินพุต จะทำให้อัตราปริมาณงานและเวลาที่ใช้ในการทำงานของวงจรมีค่าเปลี่ยนแปลงตามไปด้วย ซึ่งในงานวิจัยได้ทำการทดลองเปลี่ยนจำนวนอินพุตกับวงจรคุมขนาด 8 บิตที่ใช้วงจรเข้ารหัสแบบที่ 1 พบว่า เมื่อกำหนดจำนวนอินพุตไม่เกิน 2 ค่าเวลาในการทำงานของขั้นตอนที่ 4 จะมากกว่าเมื่อกำหนดจำนวนอินพุตไม่เกิน 3 ค่า ซึ่งทำให้เวลาที่ใช้ในการทำงานของวงจรคุมที่กำหนดจำนวนอินพุตไม่เกิน 2 ค่านั้นมีค่ามากขึ้นและทำให้อัตราปริมาณงานที่ได้ลดลง

อย่างไรก็ตาม เมื่อเปลี่ยนวงจรเข้ารหัสเป็นวงจรเข้ารหัสแบบที่ 2 จะทำให้เวลาในการทำงานของขั้นตอนที่ 4 และของทั้งวงจรลดลง รวมทั้งอัตราปริมาณงานจะมากขึ้นด้วย เนื่องจาก

วงจรเข้ารหัสแบบที่ 2 มีขนาดเล็กกว่าวงจรเข้ารหัสแบบที่ 1 ซึ่งโดยปกติแล้ววงจรที่มีขนาดเล็กกว่า จะใช้เวลาในการทำงานน้อยกว่า และงานวิจัยนี้ยังไม่ได้ทำการทดลองเปลี่ยนจำนวนอินพุตกับ วงจรคุณที่ใช้วงจรเข้ารหัสแบบที่ 2 เพราะเวลาที่ใช้ในการทำงานของแต่ละขั้นตอนมีความสมดุล ใกล้เคียงกันดีอยู่แล้ว

จากผลการวิจัยสามารถสรุปได้ว่า วงจรคุณที่ออกแบบด้วยโครงสร้างรุ่นที่ 2 และใช้วงจร เข้ารหัสแบบที่ 2 เหมาะสมกับการนำอัลกอริทึมการคูณแบบเชื่อมตรงไปประยุกต์ใช้กับการ ออกแบบวงจรแบบอสมวาร เพราะอัตราปริมาณงานและเวลาที่ใช้ในการทำงานมีค่าอยู่ในเกณฑ์ที่ ดี อย่างไรก็ตามผลการทดลองที่ได้ทั้งอัตราปริมาณงานและเวลาที่ใช้ในการทำงานของวงจรจะ ขึ้นอยู่กับการเพรสและเรอต์ของโปรแกรม ดังนั้นหากทำการเพรสและเรอต์ใหม่อาจจะทำให้อัตรา ปริมาณงานและเวลาที่ใช้ในการทำงานเปลี่ยนแปลงได้เช่นกัน

5.2 ข้อเสนอแนะ

โครงสร้างทั้ง 2 รุ่นที่ได้นำเสนอในงานวิจัยนี้ สามารถนำไปประยุกต์ใช้กับการสร้างวงจร คุณที่สามารถหาค่าผลลัพธ์ของจำนวนเต็มและจำนวนเชิงตรรกะนี้ โดยเพิ่มวงจรเข้ารหัสของข้อมูล ส่วนที่เหลือเข้าไป จะทำให้สามารถหาค่าของเลขเลขจำนวนเต็มได้ ซึ่งคาดว่า จะใช้จำนวนเกต เพิ่มขึ้นไม่มากนัก แต่การออกแบบวงจรคุณขนาด 16 บิต โดยใช้โครงสร้างรุ่นที่ 2 นี้จะต้องใช้ จำนวนเกตถึง 50,772 เกต ซึ่งเป็นวงจรที่มีขนาดใหญ่มาก ดังนั้นหากต้องการออกแบบวงจรคุณ เพื่อรองรับกับขนาดของอินพุตที่เพิ่มขึ้น จะต้องคำนึงถึงการลดจำนวนเกตที่ต้องใช้ด้วย เพื่อให้ วงจรมีขนาดใหญ่พอที่จะรับกับค่าใช้จ่ายที่เพิ่มขึ้นได้ นอกจากนี้ควรมีการทดลองปรับเปลี่ยน จำนวนอินพุตของวงจรในขั้นตอนที่ 3 เป็นต้นไปกับวงจรคุณที่ใช้วงจรเข้ารหัสแบบที่ 2 เพื่อ ตรวจสอบว่า ควรกำหนดจำนวนอินพุตเป็นเท่าใด เพื่อให้วงจรคุณที่ใช้วงจรเข้ารหัสแบบที่ 2 มี ประสิทธิภาพสูงที่สุด โดยสามารถนำทฤษฎีที่ได้นำเสนอไว้ในบทที่ 3 มาประยุกต์ใช้ในการกำหนด จำนวนขั้นตอนของวงจรคุณได้

การออกแบบวงจรที่นำเสนอในงานวิจัยนี้ เป็นการนำอัลกอริทึมการคูณแบบเชื่อมตรงมา ประยุกต์ใช้กับระบบอสมวาร ซึ่งโครงสร้างของวงจรที่ออกแบบได้ จะไม่ตรงกับแนวคิดเบื้องต้นของ อัลกอริทึมการคูณแบบเชื่อมตรง ที่จะต้องทำให้ขนาดของวงจรมีขนาดเล็ก และสามารถทำการ คำนวณกับอินพุตที่มีขนาดไม่สิ้นสุดได้มากนัก ดังนั้นในอนาคตอันใกล้นี้ ควรมีการทำงานวิจัยที่ ใช้โครงสร้างไปป์ไลน์แบบลำดับ (Sequential Pipeline) เพื่อนำผลที่ได้มาเปรียบเทียบกับวงจรคุณ ที่ออกแบบโดยใช้โครงสร้างซึ่งนำเสนอในงานวิจัยนี้

รายการอ้างอิง

1. A. Surarerks. Arithmétique en ligne en base réelle et complexe. Thèse de Doctorat Université Pierre et Marie Curie, 2001.
2. A. Surarerks. Digit Set Conversion by On-line Finite Automata , Bulletin of the Belgian Mathematical Society, Simon Stevin 8, 2 (2001): 337-358.
3. A. Takamura, M. Imai, T. Fuji, M. Ozawa, I. Fukasaku, Y. Ueno and T. Nanya. TITAC-2: An asynchronous 32-bit microprocessor based on Scalable-Delay-Insensitive Model. Proceeding of ICCD IEEE (October 1997): 288 -294.
4. Ch. Frougny. On-the-fly algorithms and sequential machines. IEEE Transactions on Computers 49, 8 (August 2000): 859 –863.
5. Ch. Frougny and A. Surarerks. On-line multiplication in real and complex base, Proceedings of the 16th IEEE Symposium on Computer Arithmetic, Santiago de Compostela, Spain, (June 2003).
6. Exemplar Logic Incorporation. [Online], Available from: <http://www.mentor.com/leonardospectrum.>, (2003).
7. G. Even, S.M. Mueller and P.-M. Seidel. A dual mode IEEE multiplier. Second Annual IEEE International Conference on Innovative Systems in Silicon Proceedings (1997): 282-289.
8. H. Dhanesha, K. Falakshahi and M. Horowitz. Array-of-arrays architecture for parallel floating point multiplication. Proceeding of Advanced Research in VLSI (1995): 150-157.
9. I. E. Sutherland. Micropipelines. Communication of the ACM 32, 6 (June 1989): 720-738.
10. J. Bhasker. A VHDL Primer, Prentice-Hall, 1992.
11. J. Butas, Chiu-Sing Choy, J. Povazanec and Cheong-Fat Chan. Asynchronous cross-pipelined multiplier. IEEE Journal of Solid-State Circuits 36, 8 (August 2001): 1272-1275.
12. M.D. Ercegovac and T. Lang. On-the-Fly Conversion of Redundant into Conventional Representations. IEEE Transactions on Computers C-36, 7 (July 1987): 895-897.

13. M.D. Ercegovac and T. Lang. On-Line Arithmetic for DSP Application. Proceedings of the 32nd Midwest 1 (1990): 365-368.
14. Model Technology. [Online], Available from: <http://www.model.com.>, (2003).
15. P.Warabuntaweek and A.Thongtak. Design of Dual-rail Asynchronous Multiplier using Booth's Algorithm, Proceeding of the 6th National Computer Science and Engineering Conference, 2002, pp. 342-348.
16. S.B. Furber and P. Day. Four-phase micropipeline latch control circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 4, 2 (June 1996): 247 -253.
17. S. Hauck. Asynchronous Design Methodologies: An Overview. Proceeding of the IEEE 83, 1 (January 1995): 69-93.
18. T. Nanya, Y. Ueno, H. Kagotani, M. Kuwako and A. Takamura. TITAC: design of a quasi-delay insensitive microprocessor. IEEE Design & Test of Computers 11, 2 (1994): 50 -63.
19. Xilinx Incorporation. [Online], Available from: <http://www.xilinx.com.>, (2003).
20. Xilinx Incorporation. The Programmable Logic Data Book 2000. United State of America, 2000.
21. ปัญญา เรื่องสินทรัพย์. การออกแบบไมโครโปรเซสเซอร์ 8 บิต ที่ไม่ไวต่อความหน่วงชนิดปรับมาตราส่วนได้โดยใช้เอฟพีจีเอ. หลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมคอมพิวเตอร์ จุฬาลงกรณ์มหาวิทยาลัย, 2544.



ภาคผนวก

สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ก

ผลการจำลองการทำงานของวงจรมัลติเพล็กซ์แบบเชื่อมตรง

ผลการจำลองการทำงานของวงจรมัลติเพล็กซ์ขนาด 8 บิตที่ออกแบบด้วยโครงสร้างรุ่นที่ 1

ตัวคูณ (8Bit)	ตัวตั้ง (8Bit)	ผลลัพธ์ (Exact)	ผลลัพธ์ที่ได้ (8Bit)	WT (ns)	IT (ns)	CT (ns)	T _{st1} (ns)	T _{st2} (ns)	T _{st3} (ns)
0x69	0x76	0x30 66	0x31	122	40	162	20	68	30
0x53	0x68	0x21 B8	0x22	121	42	163	20	69	28
0x42	0xA1	0x29 82	0x2A	121	42	163	20	70	27
0x67	0xB0	0x46 D0	0x47	124	40	164	20	69	31
0x38	0xD6	0x2E D0	0x2F	119	38	157	20	64	31
0xA9	0xE0	0x93 E0	0x94	121	42	163	22	67	28
0xCD	0x78	0x60 18	0x61	122	40	162	21	67	30
0xF3	0x34	0x31 5C	0x32	114	40	154	20	62	28
0xE7	0x55	0x4C B3	0x4D	118	39	157	21	65	28
0x64	0x7E	0x31 38	0x32	118	41	159	20	66	28
0x88	0xB7	0x61 38	0x62	116	42	158	21	64	27
0x74	0x6D	0x31 64	0x32	118	41	159	20	66	28
0xCB	0xC9	0x9F 63	0xA0	114	38	152	20	60	30
0xB1	0x62	0x43 C2	0x44	120	42	162	20	68	28
0xFB	0xDA	0xD5 BE	0xD6	121	39	160	21	65	31
0x82	0x93	0x4A A6	0x4B	116	43	159	21	64	27
0x2D	0x3C	0x0A 8C	0x0B	114	41	155	20	62	28
0x9A	0x1A	0x0F A4	0x10	113	39	152	20	59	30
0x35	0xB7	0x25 E3	0x26	123	39	162	20	68	31
0x9A	0x1D	0x11 72	0x12	110	41	151	20	58	28
0xA4	0x6D	0x45 D4	0x46	125	39	164	20	70	31
0xD7	0x27	0x20 C1	0x21	116	39	155	20	62	30
0x84	0xC0	0x63 00	0x63	115	41	156	20	63	28
0xC3	0xB1	0x86 D3	0x87	124	40	164	21	68	31
0x76	0x49	0x21 A6	0x22	117	41	158	20	65	28
0x0B	0x97	0x06 7D	0x07	120	40	160	20	65	31
0x5D	0xA3	0x3B 37	0x3C	116	41	157	20	64	28
0x63	0x9B	0x3B F1	0x3C	116	41	157	20	64	28
0xF3	0xEF	0xE2 DD	0xE3	110	43	153	20	59	27
0x9B	0x79	0x49 43	0x4A	119	43	162	21	68	27
0x4D	0x5C	0x1B AC	0x1C	116	39	155	20	64	28
0xD8	0x58	0x4A 40	0x4B	117	42	159	20	65	28
0x93	0x9C	0x59 94	0x5A	116	43	159	20	65	27
0x8D	0x50	0x2C 10	0x2D	120	42	162	21	67	28
0x71	0x94	0x41 54	0x42	116	38	154	20	64	28
0xA1	0x5D	0x3A 7D	0x3B	119	42	161	21	66	28
0x40	0xF4	0x3D 00	0x3D	121	36	157	20	66	31
0x60	0x79	0x2D 60	0x2E	123	38	161	20	68	31
0x36	0x49	0x0F 66	0x10	115	39	154	20	31	30
0xA8	0x8E	0x5D 30	0x5E	119	38	157	20	64	31

Average		118.125	40.35	158.475	20.275	64.225	28.9
เวลาเฉลี่ยที่ใช้ในการคูณ	=	118.125	ns				
เวลาเฉลี่ยที่ใช้ในการกำหนดค่าเริ่มต้น	=	40.35	ns				
เวลาเฉลี่ยที่ใช้ในการทำงานต่อ 1 รอบ	=	158.475	ns				
อัตราปริมาณงาน	=	15.6694	MHz				

ผลการจำลองการทำงานของวงจรมัลติเพล็กซ์ขนาด 8 บิตที่ออกแบบด้วยโครงสร้างรุ่นที่ 2
ใช้วงจรเข้ารหัสแบบที่ 1 และกำหนดจำนวนอินพุตไม่เกิน 3

ตัวคูณ (8Bit)	ตัวตั้ง (8Bit)	ผลลัพธ์ที่ได้ (8Bit)	WT (ns)	IT (ns)	CT (ns)	T _{st1} (ns)	T _{st2} (ns)	T _{st3} (ns)	T _{st4} (ns)	T _{st5} (ns)
0x69	0x76	0x31	165	30	195	26	23	31	47	32
0x53	0x68	0x22	162	29	191	25	22	31	47	31
0x42	0xA1	0x2A	163	27	190	25	22	29	49	32
0x67	0xB0	0x47	169	27	196	25	23	35	46	34
0x38	0xD6	0x2F	163	28	191	25	24	28	46	34
0xA9	0xE0	0x94	165	30	195	25	22	38	43	31
0xCD	0x78	0x61	165	29	194	24	23	35	44	33
0xF3	0x34	0x32	160	28	188	25	22	28	48	31
0xE7	0x55	0x4D	157	31	188	25	22	27	46	31
0x64	0x7E	0x32	164	29	193	25	23	31	48	31
0x88	0xB7	0x62	159	29	188	25	23	28	44	33
0x74	0x6D	0x32	164	29	193	25	23	31	48	31
0xCB	0xC9	0xA0	157	27	184	24	22	31	41	33
0xB1	0x62	0x44	164	31	195	25	23	35	44	31
0xFB	0xDA	0xD6	156	29	185	24	22	27	45	32
0x82	0x93	0x4B	161	29	190	25	22	28	48	32
0x2D	0x3C	0x0B	160	30	190	24	22	31	46	31
0x9A	0x1A	0x10	159	30	189	24	24	31	42	32
0x35	0xB7	0x26	162	28	190	25	23	28	48	32
0x9A	0x1D	0x12	163	28	191	24	24	31	47	31
0xA4	0x6D	0x46	158	29	187	25	23	29	43	32
0xD7	0x27	0x21	162	28	190	25	23	31	44	33
0x84	0xC0	0x63	158	30	188	25	22	27	47	31
0xC3	0xB1	0x87	173	27	200	25	23	38	47	34
0x76	0x49	0x22	160	30	190	25	22	31	45	31
0x0B	0x97	0x07	164	27	191	25	22	31	46	34
0x5D	0xA3	0x3C	151	29	180	25	22	28	37	32
0x63	0x9B	0x3C	151	29	180	25	23	29	36	32
0xF3	0xEF	0xE3	158	28	186	25	23	28	43	33
0x9B	0x79	0x4A	158	30	188	24	24	29	43	32
0x4D	0x5C	0x1C	154	31	185	24	22	31	40	31
0xD8	0x58	0x4B	157	31	188	24	24	27	45	32
0x93	0x9C	0x5A	159	30	189	25	23	28	45	32
0x8D	0x50	0x2D	160	31	191	25	22	29	47	31
0x71	0x94	0x42	156	30	186	25	24	29	41	31
0xA1	0x5D	0x3B	154	29	183	25	22	29	40	32
0x40	0xF4	0x3D	156	28	184	25	22	29	41	33
0x60	0x79	0x2E	164	29	193	25	23	31	47	32
0x36	0x49	0x10	159	30	189	25	23	31	43	32
0xA8	0x8E	0x5E	156	30	186	24	23	28	43	32
Average			160.15	29.1	189.25	24.775	22.725	30.175	44.5	32

เวลาเฉลี่ยที่ใช้ในการคูณ = 160.15 ns

เวลาเฉลี่ยที่ใช้ในการกำหนดค่าเริ่มต้น = 29.1 ns

เวลาเฉลี่ยที่ใช้ในการทำงานต่อ 1 รอบ = 189.25 ns

อัตราปริมาณงาน = 30.3580 MHz

ผลการจำลองการทำงานของวงจรถูกขนาด 8 บิตที่ออกแบบด้วยโครงสร้างรุ่นที่ 2
ใช้วงจรเข้ารหัสแบบที่ 1 และกำหนดจำนวนอินพุตไม่เกิน 2

ตัวคูณ (8Bit)	ตัวตั้ง (8Bit)	ผลลัพธ์ที่ ได้ (8Bit)	WT (ns)	IT (ns)	CT (ns)	T _{st1} (ns)	T _{st2} (ns)	T _{st3} (ns)	T _{st4} (ns)	T _{st5} (ns)
0x69	0x76	0x31	183	27	210	26	22	31	66	33
0x53	0x68	0x22	182	25	207	25	22	34	64	32
0x42	0xA1	0x2A	177	24	201	25	21	25	69	32
0x67	0xB0	0x47	178	28	206	25	21	26	69	32
0x38	0xD6	0x2F	175	28	203	25	22	27	64	32
0xA9	0xE0	0x94	169	24	193	25	22	28	56	33
0xCD	0x78	0x61	173	27	200	25	21	31	58	33
0xF3	0x34	0x32	175	26	201	25	21	27	65	32
0xE7	0x55	0x4D	184	26	210	25	22	34	65	33
0x64	0x7E	0x32	182	26	208	25	22	31	67	32
0x88	0xB7	0x62	172	25	197	25	22	32	56	32
0x74	0x6D	0x32	182	26	208	25	22	31	67	32
0xCB	0xC9	0xA0	175	24	199	25	22	32	58	33
0xB1	0x62	0x44	179	23	202	25	21	27	68	33
0xFB	0xDA	0xD6	181	25	206	25	21	31	67	32
0x82	0x93	0x4B	179	27	206	25	21	31	65	32
0x2D	0x3C	0x0B	167	27	194	25	22	25	58	32
0x9A	0x1A	0x10	179	24	203	25	21	31	64	33
0x35	0xB7	0x26	186	25	211	25	22	34	68	32
0x9A	0x1D	0x12	180	26	206	25	22	31	65	32
0xA4	0x6D	0x46	178	27	205	25	22	36	68	32
0xD7	0x27	0x21	170	27	197	25	21	28	58	33
0x84	0xC0	0x63	163	26	189	25	21	25	53	32
0xC3	0xB1	0x87	181	28	209	25	21	34	64	32
0x76	0x49	0x22	172	25	197	25	22	26	62	32
0x0B	0x97	0x07	175	28	203	25	22	25	66	32
0x5D	0xA3	0x3C	175	24	199	25	22	31	59	33
0x63	0x9B	0x3C	178	24	202	25	22	33	60	33
0xF3	0xEF	0xE3	168	28	196	25	22	24	60	32
0x9B	0x79	0x4A	179	27	206	25	22	29	66	32
0x4D	0x5C	0x1C	175	24	199	25	22	32	58	33
0xD8	0x58	0x4B	179	26	205	25	21	31	65	32
0x93	0x9C	0x5A	171	25	196	25	22	26	61	32
0x8D	0x50	0x2D	179	26	205	25	21	31	64	33
0x71	0x94	0x42	179	26	205	25	21	33	63	32
0xA1	0x5D	0x3B	175	28	203	25	22	31	60	32
0x40	0xF4	0x3D	171	25	196	25	21	25	63	32
0x60	0x79	0x2E	181	25	206	25	21	31	67	32
0x36	0x49	0x10	169	24	193	25	21	32	55	32
0xA8	0x8E	0x5E	170	25	195	25	22	32	54	32
Average			176.15	25.775	201.925	25.025	21.575	29.85	62.625	32.3

เวลาเฉลี่ยที่ใช้ในการคูณ	=	176.15	ns
เวลาเฉลี่ยที่ใช้ในการกำหนดค่าเริ่มต้น	=	25.775	ns
เวลาเฉลี่ยที่ใช้ในการทำงานต่อ 1 รอบ	=	201.925	ns
อัตราปริมาณงาน	=	27.1777	MHz

ผลการจำลองการทำงานของวงจรมอดูลขนาด 8 บิตที่ออกแบบด้วยโครงสร้างรุ่นที่ 2
ใช้วงจรเข้ารหัสแบบที่ 2 และกำหนดจำนวนอินพุตไม่เกิน 3

ตัวคูณ (8Bit)	ตัวตั้ง (8Bit)	ผลลัพธ์ที่ ได้ (8Bit)	WT (ns)	IT (ns)	CT (ns)	T _{st1} (ns)	T _{st2} (ns)	T _{st3} (ns)	T _{st4} (ns)	T _{st5} (ns)
0x69	0x76	0x31	149	27	176	26	23	31	31	32
0x53	0x68	0x22	147	27	174	25	22	31	34	29
0x42	0xA1	0x2A	148	27	175	25	22	29	32	34
0x67	0xB0	0x47	159	25	184	25	23	35	36	34
0x38	0xD6	0x2F	153	21	174	25	24	28	36	34
0xA9	0xE0	0x94	156	29	185	25	22	38	35	30
0xCD	0x78	0x61	162	24	186	24	23	35	40	34
0xF3	0x34	0x32	142	33	175	25	22	28	32	29
0xE7	0x55	0x4D	138	30	168	25	22	27	28	30
0x64	0x7E	0x32	148	27	175	25	23	31	34	29
0x88	0xB7	0x62	152	20	172	25	23	28	34	36
0x74	0x6D	0x32	146	29	175	25	23	31	32	29
0xCB	0xC9	0xA0	157	21	178	24	22	31	40	34
0xB1	0x62	0x44	150	32	182	25	23	35	31	30
0xFB	0xDA	0xD6	142	29	171	24	22	27	32	31
0x82	0x93	0x4B	145	29	174	25	22	28	30	34
0x2D	0x3C	0x0B	144	28	172	24	22	31	30	31
0x9A	0x1A	0x10	148	30	178	24	24	31	31	32
0x35	0xB7	0x26	138	33	171	25	23	28	25	31
0x9A	0x1D	0x12	146	30	176	24	24	31	32	29
0xA4	0x6D	0x46	146	28	174	25	23	29	32	31
0xD7	0x27	0x21	149	29	178	25	23	31	30	34
0x84	0xC0	0x63	145	29	174	25	22	27	34	31
0xC3	0xB1	0x87	159	26	185	25	23	38	33	34
0x76	0x49	0x22	148	26	174	25	22	31	34	30
0x0B	0x97	0x07	142	29	171	25	22	31	24	34
0x5D	0xA3	0x3C	144	27	171	25	22	29	29	33
0x63	0x9B	0x3C	145	27	172	25	23	29	29	33
0xF3	0xEF	0xE3	146	23	169	25	23	28	28	36
0x9B	0x79	0x4A	141	30	171	24	24	29	24	34
0x4D	0x5C	0x1C	152	24	176	24	22	31	38	31
0xD8	0x58	0x4B	145	29	174	24	24	27	33	31
0x93	0x9C	0x5A	148	24	172	25	23	28	32	34
0x8D	0x50	0x2D	149	28	177	25	22	29	37	30
0x71	0x94	0x42	147	31	178	25	24	29	32	31
0xA1	0x5D	0x3B	144	30	174	25	22	29	29	33
0x40	0xF4	0x3D	152	25	177	25	22	29	37	33
0x60	0x79	0x2E	152	27	179	25	23	31	36	31
0x36	0x49	0x10	147	26	173	25	22	31	31	32
0xA8	0x8E	0x5E	150	24	174	24	23	28	38	31
Average			148.025	27.325	175.35	24.775	22.7	30.2	32.375	31.975

เวลาเฉลี่ยที่ใช้ในการคูณ	=	148.025	ns
เวลาเฉลี่ยที่ใช้ในการกำหนดค่าเริ่มต้น	=	27.325	ns
เวลาเฉลี่ยที่ใช้ในการทำงานต่อ 1 รอบ	=	175.35	ns
อัตราปริมาณงาน	=	35.5947	MHz

ผลการจำลองการทำงานของวงจรคุณขนาด 16 บิตที่ออกแบบด้วยโครงสร้างรุ่นที่ 1

ตัวคูณ (16 Bit)	ตัวตั้ง (16 Bit)	ผลลัพธ์ (Exact)	ผลลัพธ์ที่ได้ (16 Bit)	WT (ns)	IT (ns)	CT (ns)	T _{st1} (ns)	T _{st2} (ns)	T _{st3} (ns)
0xC3F7	0x0658	0x04DB 26E8	0x04DC	202	33	235	31	138	30
0xB125	0x8976	0x5F1E 740E	0x5F1F	205	31	236	32	138	32
0x67A9	0xC3F5	0x4F58 EFB D	0x4F59	211	29	240	31	145	32
0x832C	0xE46A	0x7509 8038	0x750A	191	31	222	31	125	32
0x5D4A	0x7650	0x2B1D 4320	0x2B1E	195	32	227	31	130	31
0x37BC	0x8310	0x1C88 AFC0	0x1C89	204	34	238	34	138	29
0x7634	0x3315	0x1796 0E44	0x1797	205	31	236	31	140	31
0x22FC	0x6832	0x0E3D 3538	0x0E3E	197	31	228	31	132	31
0x6371	0xF437	0x5EDD 1147	0x5EDE	200	29	229	31	134	32
0x789F	0xE56C	0x6C19 1E14	0x6C1A	193	29	222	31	129	30
0x897E	0x39CB	0x1F0A 14EA	0x1F0B	209	31	240	34	140	32
0xEBBF	0xFFCF	0xEB91 E071	0xEB92	201	28	229	32	135	31
0x25E1	0xC1C2	0x1CAB 5582	0x1CAC	199	32	231	34	133	29
0xDF62	0xC425	0xAB27 512A	0xAB28	196	33	229	31	133	29
0x6A48	0x5DF5	0x2701 DEE8	0x2702	194	28	222	31	127	31
0xF8B0	0x8A2E	0x863B 8FA0	0x863C	193	35	228	31	131	28
0xECC0	0x7400	0x6B47 0000	0x6B47	206	31	237	31	143	29
0xDF20	0x4DCA	0x43CC AF40	0x43CD	212	31	243	33	144	32
0xCC3E	0x8F64	0x7266 6A38	0x7267	208	33	241	31	145	29
0x67EB	0xA238	0x41D9 7168	0x41DA	200	28	228	31	136	30
0x0F58	0x8F52	0x0897 1230	0x0898	193	34	227	33	130	27
0x2B64	0xC666	0x21A0 A1D8	0x21A1	205	32	237	31	142	29
0xB295	0x5C3B	0x4056 B457	0x4057	203	29	232	31	137	32
0x968C	0xED8D	0x8BB2 871C	0x8BB3	198	32	230	31	134	30
0xF340	0x8800	0x813A 0000	0x813A	196	30	226	31	131	31
0x12BD	0xB8EE	0x0D89 43B6	0x0D8A	199	29	228	31	134	31
0x790D	0x2596	0x11C5 CE9E	0x11C6	197	30	227	33	132	29
0x94C3	0xF936	0x90D1 0C22	0x90D2	194	30	224	31	130	30
0xD614	0x0673	0x0564 A2FC	0x0565	218	30	248	31	152	32
0x2115	0x8C12	0x1219 CF7A	0x121A	197	32	229	31	134	29
0x6D16	0xF4F1	0x685F A9B6	0x6860	197	29	226	31	131	32
0xD800	0xC880	0xA92C 0000	0xA92C	201	33	234	31	138	29
0x26EA	0xCFD0	0x1F96 D420	0x1F97	206	29	235	31	139	33
0x83F4	0xCA39	0x683B E954	0x683C	189	31	220	31	125	30
0x21F4	0xEDAB	0x1F85 91FC	0x1F86	205	26	231	34	135	33
0x9A86	0x6BE2	0x411E 6C4C	0x411F	198	31	229	31	132	32
0x8F3F	0x8D4C	0x4F10 39B4	0x4F11	203	31	234	32	137	31
0xB48E	0xC1CA	0x88AD 860C	0x88AE	192	30	222	31	127	31
0xD9C5	0x6911	0x5960 4315	0x5961	208	29	237	33	140	32
0xCDE1	0xD76C	0xAD3E D1EC	0xAD3F	204	32	236	32	138	31
Average				200.6	30.725	231.325	31.6	135.35	30.6

เวลาเฉลี่ยที่ใช้ในการคูณ	=	200.6	ns
เวลาเฉลี่ยที่ใช้ในการกำหนดค่าเริ่มต้น	=	30.725	ns
เวลาเฉลี่ยที่ใช้ในการทำงานต่อ 1 รอบ	=	231.325	ns
อัตราปริมาณงาน	=	11.3027	MHz

ผลการจำลองการทำงานของวงจรมูลขนาด 16 บิตที่ออกแบบด้วยโครงสร้างรุ่นที่ 2
ใช้วงจรเข้ารหัสแบบที่ 1 และกำหนดจำนวนอินพุตไม่เกิน 5

ตัวคูณ (16 Bit)	ตัวตั้ง (16 Bit)	ผลลัพธ์ที่ได้ (16 Bit)	WT (ns)	IT (ns)	CT (ns)	T _{st1} (ns)	T _{st2} (ns)	T _{st3} (ns)	T _{st4} (ns)	T _{st5} (ns)
0xC3F7	0x0658	0x04DC	257	25	282	27	31	42	118	35
0xB125	0x8976	0x5F1F	253	36	289	27	31	47	119	25
0x67A9	0xC3F5	0x4F59	255	36	291	28	31	46	118	28
0x832C	0xE46A	0x750A	251	31	282	27	31	46	113	30
0x5D4A	0x7650	0x2B1E	249	35	284	28	31	40	120	26
0x37BC	0x8310	0x1C89	248	32	280	28	31	44	114	27
0x7634	0x3315	0x1797	249	35	284	28	31	41	116	29
0x22FC	0x6832	0x0E3E	252	33	285	28	31	46	116	27
0x6371	0xF437	0x5EDE	253	35	288	28	31	43	122	25
0x789F	0xE56C	0x6C1A	243	35	278	28	31	40	115	25
0x897E	0x39CB	0x1F0B	252	34	286	31	31	45	114	27
0xEBBF	0xFFCF	0xEB92	257	36	293	29	31	44	121	28
0x25E1	0xC1C2	0x1CAC	252	32	284	27	31	41	122	27
0xDF62	0xC425	0xAB28	251	34	285	28	31	46	115	27
0x6A48	0x5DF5	0x2702	247	30	277	31	31	40	111	30
0xF8B0	0x8A2E	0x863C	257	33	290	29	31	45	121	27
0xECC0	0x7400	0x6B47	243	40	283	28	30	41	116	24
0xDF20	0x4DCA	0x43CD	248	34	282	30	29	46	109	30
0xCC3E	0x8F64	0x7267	250	33	283	29	30	45	116	26
0x67EB	0xA238	0x41DA	246	35	281	27	31	41	114	29
0x0F58	0x8F52	0x0898	253	27	280	29	31	46	115	28
0x2B64	0xC666	0x21A1	250	37	287	27	31	41	120	27
0xB295	0x5C3B	0x4057	244	28	272	28	30	46	106	30
0x968C	0xED8D	0x8BB3	244	37	281	28	30	40	115	27
0xF340	0x8800	0x813A	257	31	288	27	31	45	117	33
0x12BD	0xB8EE	0x0D8A	254	36	290	28	31	45	118	28
0x790D	0x2596	0x11C6	249	37	286	28	31	42	117	27
0x94C3	0xF936	0x90D2	247	34	281	28	31	43	117	24
0xD614	0x0673	0x0565	260	29	289	28	31	41	121	35
0x2115	0x8C12	0x121A	262	33	295	28	31	46	126	27
0x6D16	0xF4F1	0x6860	246	28	274	28	30	41	112	31
0xD800	0xC880	0xA92C	254	37	291	29	31	41	122	27
0x26EA	0xCFD0	0x1F97	246	35	281	27	31	46	109	29
0x83F4	0xCA39	0x683C	251	34	285	28	31	42	120	26
0x21F4	0xEDAB	0x1F86	249	35	284	28	31	46	111	29
0x9A86	0x6BE2	0x411F	248	30	278	32	31	46	105	30
0x8F3F	0x8D4C	0x4F11	244	37	281	29	30	44	113	24
0xB48E	0xC1CA	0x88AE	254	33	287	27	31	43	123	26
0xD9C5	0x6911	0x5961	252	36	288	30	31	43	115	28
0xCDE1	0xD76C	0xAD3F	243	34	277	27	31	41	113	27
Average			250.5	33.55	284.05	28.25	30.8	43.425	116.125	27.875

เวลาเฉลี่ยที่ใช้ในการคูณ	=	250.5	ns
เวลาเฉลี่ยที่ใช้ในการกำหนดค่าเริ่มต้น	=	33.55	ns
เวลาเฉลี่ยที่ใช้ในการทำงานต่อ 1 รอบ	=	284.05	ns
อัตราปริมาณงาน	=	17.2591	MHz

ผลการจำลองการทำงานของวงจรคุณขนาด 16 บิตที่ออกแบบด้วยโครงสร้างรุ่นที่ 2
ใช้วงจรเข้ารหัสแบบที่ 2 และกำหนดจำนวนอินพุตไม่เกิน 5

ตัวคูณ (16 Bit)	ตัวตั้ง (16 Bit)	ผลลัพธ์ที่ได้ (16 Bit)	WT (ns)	IT (ns)	CT (ns)	T _{st1} (ns)	T _{st2} (ns)	T _{st3} (ns)	T _{st4} (ns)	T _{st5} (ns)
0xC3F7	0x0658	0x04DC	180	30	210	27	31	37	47	33
0xB125	0x8976	0x5F1F	195	29	224	24	32	44	57	33
0x67A9	0xC3F5	0x4F59	197	28	225	24	32	40	63	33
0x832C	0xE46A	0x750A	197	30	227	24	32	44	59	33
0x5D4A	0x7650	0x2B1E	187	28	245	24	31	43	51	33
0x37BC	0x8310	0x1C89	186	33	219	24	32	46	49	30
0x7634	0x3315	0x1797	198	24	222	24	31	40	63	35
0x22FC	0x6832	0x0E3E	191	29	220	24	32	40	57	33
0x6371	0xF437	0x5EDE	183	30	213	24	32	40	52	33
0x789F	0xE56C	0x6C1A	187	28	215	24	32	40	55	31
0x897E	0x39CB	0x1F0B	193	33	226	27	32	42	57	30
0xE8BF	0xFFCF	0xEB92	187	28	215	26	32	43	50	31
0x25E1	0xC1C2	0x1CAC	179	33	212	24	32	39	49	30
0xDF62	0xC425	0xAB28	183	29	212	24	32	41	53	28
0x6A48	0x5DF5	0x2702	190	25	215	27	31	39	55	33
0xF8B0	0x8A2E	0x863C	199	33	232	27	32	41	64	30
0xECC0	0x7400	0x6B47	187	28	214	24	32	40	53	33
0xDF20	0x4DCA	0x43CD	205	28	233	27	31	50	59	33
0xCC3E	0x8F64	0x7267	180	27	207	24	32	38	48	33
0x67EB	0xA238	0x41DA	193	28	221	24	31	41	59	33
0x0F58	0x8F52	0x0898	187	28	215	24	32	45	53	28
0x2B64	0xC666	0x21A1	175	29	204	24	31	38	47	30
0xB295	0x5C3B	0x4057	184	24	208	24	32	37	51	35
0x968C	0xED8D	0x8BB3	175	27	202	24	31	36	49	30
0xF340	0x8800	0x813A	194	32	226	24	32	40	62	31
0x12BD	0xB8EE	0x0D8A	188	32	220	24	32	38	58	31
0x790D	0x2596	0x11C6	176	30	206	24	32	40	46	29
0x94C3	0xF936	0x90D2	182	27	209	24	32	41	46	34
0xD614	0x0673	0x0565	182	24	206	24	32	37	50	34
0x2115	0x8C12	0x121A	174	33	207	24	32	36	47	30
0x6D16	0xF4F1	0x6860	199	27	226	24	32	45	60	33
0xD800	0xC880	0xA92C	190	33	223	27	32	40	56	30
0x26EA	0xCFD0	0x1F97	204	26	230	24	31	44	69	31
0x83F4	0xCA39	0x683C	200	30	230	24	31	38	69	33
0x21F4	0xEDAB	0x1F86	201	28	229	24	32	42	67	31
0x9A86	0x6BE2	0x411F	196	29	225	27	31	45	57	33
0x8F3F	0x8D4C	0x4F11	179	29	208	24	31	44	47	28
0xB48E	0xC1CA	0x88AE	186	29	215	24	32	43	49	33
0xD9C5	0x6911	0x5961	189	28	217	27	32	41	51	33
0xCDE1	0xD76C	0xAD3F	202	27	229	24	31	50	59	33
Average			189	28.8	217.8	24.65	31.675	41.125	54.825	31.775

เวลาเฉลี่ยที่ใช้ในการคูณ	=	189	ns
เวลาเฉลี่ยที่ใช้ในการกำหนดค่าเริ่มต้น	=	28.8	ns
เวลาเฉลี่ยที่ใช้ในการทำงานต่อ 1 รอบ	=	217.8	ns
อัตราปริมาณงาน	=	27.1524	MHz

- หมายเหตุ

WT	คือ เวลาที่ใช้ในการคูณ
IT	คือ เวลาที่ใช้ในการกำหนดค่าเริ่มต้นในขั้นตอนสุดท้ายของแต่ละวงจร
CT	คือ เวลาที่ใช้ในการทำงานต่อ 1 รอบ
$T_{st,i}$	คือ เวลาที่ใช้ในการทำงานในขั้นตอนที่ i ของแต่ละวงจร



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ภาคผนวก ข

การคำนวณหาอัตราปริมาณงานของวงจรไปป์ไลน์แบบอสสมวาร

การหาค่าอัตราปริมาณงานของวงจรไปป์ไลน์แบบอสสมวาร ซึ่งออกแบบโดยใช้รหัสรางคู่ จะหาจากเวลาที่ใช้ในการทำงาน ดังสมการ

$$\text{ThR} = \frac{N}{T}$$

ThR คือ อัตราปริมาณงาน

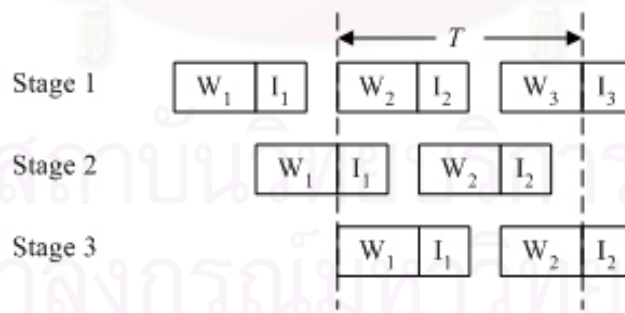
N คือ จำนวนเอาต์พุตที่วงจรหาค่าได้ในเวลา 1 รอบการทำงาน

T คือ เวลาที่ใช้ในการทำงานต่อ 1 รอบ

เนื่องจากวงจรที่ออกแบบโดยใช้รหัสรางคู่ จะมีการทำงานแบบขั้นตอนวนขั้นตอนดังรูป เพราะวงจรในแต่ละขั้นตอนจะต้องทำการกำหนดค่าเริ่มต้นเสมอ และต้องรอให้วงจรในขั้นตอนถัดไปทำงานเสร็จสิ้นก่อนที่จะเริ่มทำงานต่อไปได้ ดังนั้นค่าของ N จะไม่เท่ากับจำนวนขั้นตอนของวงจร แต่จะมีค่าเท่ากับจำนวนเอาต์พุตที่วงจรสามารถหาค่าได้ในเวลา 1 รอบการทำงานเท่านั้น ซึ่งจากรูปจะสามารถหาค่าของ N ได้ดังนี้

$$N = \left\lfloor \frac{S}{2} \right\rfloor$$

เมื่อ S คือ จำนวนขั้นตอนทั้งหมดของวงจร



W_i คือ เวลาในขั้นการทำงานลำดับที่ i

I_i คือ เวลาในขั้นว่างลำดับที่ i

ภาคผนวก ก
คำแปลศัพท์ที่ใช้ในวิทยานิพนธ์

คำศัพท์	คำแปลภาษาไทย
Digital Signal Processing (DSP) System	- ระบบการประมวลผลสัญญาณเชิงเลข
Throughput Rate	- อัตราปริมาณงาน
Real-Time 3D Graphic Applications	- โปรแกรมประยุกต์ภาพสามมิติแบบเวลาจริง
Synchronous Circuit	- วงจรแบบสมวาร
Asynchronous Circuit	- วงจรแบบอสมวาร
Clock Skew	- การแกว่งของสัญญาณนาฬิกา
On-Line Multiplication Algorithm	- อัลกอริทึมการคูณแบบเชื่อมต่อตรง
Inverter gate	- เกตผกผัน
On-Line Delay	- ความหน่วงเชื่อมต่อตรง
Base Number System	-ฐานของระบบตัวเลข
Redundant Sign-Digit Number System	- ระบบตัวเลขซ้ำซ้อนที่มีเครื่องหมาย
Operand Bound	- ค่าขอบเขตของตัวดำเนินการ
Negative Error On-line Multiplication Algorithm	- อัลกอริทึมการคูณเชื่อมต่อตรงแบบข้อผิดพลาดติดลบ
Residual Function	- ฟังก์ชันเศษเหลือ
Dual-rail Code	- รหัสรางคู่
Actual Delay	- ค่าความหน่วงจริง
2-Rail Codeword	- รหัสตรงรางคู่
Dual-Rail Circuit	- วงจรรางคู่
Acknowledgement Circuit	- วงจรตอบรับ
Scalable Delay Insensitive Model (SDI Model)	- แบบจำลองความหน่วงที่ไม่ไวต่อความหน่วงชนิดปรับมาตราส่วนได้
C-Element	- อุปกรณ์ชนิดซี
Completion Signal	- สัญญาณแสดงความบริบูรณ์
Reduced-Ordered Binary Decision Diagram (ROBDD)	- แผนภาพตัดสินใจแบบทวิภาคชนิดมีการลดทอนอันดับ

คำศัพท์

คำแปลภาษาไทย

Request Signal	- สัญญาณร้องขอ
Acknowledgement Signal	- สัญญาณตอบรับ
2-Phase Handshake Protocol	- โปรโตคอลการติดสัญญาแบบ 2 ขั้นตอน
4-Phase Handshake Protocol	- โปรโตคอลการติดสัญญาแบบ 4 ขั้นตอน
Normalization Technique	- เทคนิคการทำให้เป็นบรรทัดฐาน
Normal Form	- รูปแบบบรรทัดฐาน
Truth Table	- ตารางค่าความจริง



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย

ประวัติผู้เขียนวิทยานิพนธ์

นายปิยะ วราบุญทวีสุข เกิดเมื่อวันที่ 31 พฤษภาคม พ.ศ. 2522 ที่จังหวัด กรุงเทพมหานคร สำเร็จการศึกษาปริญญาวิทยาศาสตรบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ จากภาควิชาวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ในปีการศึกษา 2543 และได้เข้าศึกษาต่อในหลักสูตรวิทยาศาสตรมหาบัณฑิต สาขาวิศวกรรมคอมพิวเตอร์ ที่ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย ในปีการศึกษา 2544



สถาบันวิทยบริการ
จุฬาลงกรณ์มหาวิทยาลัย