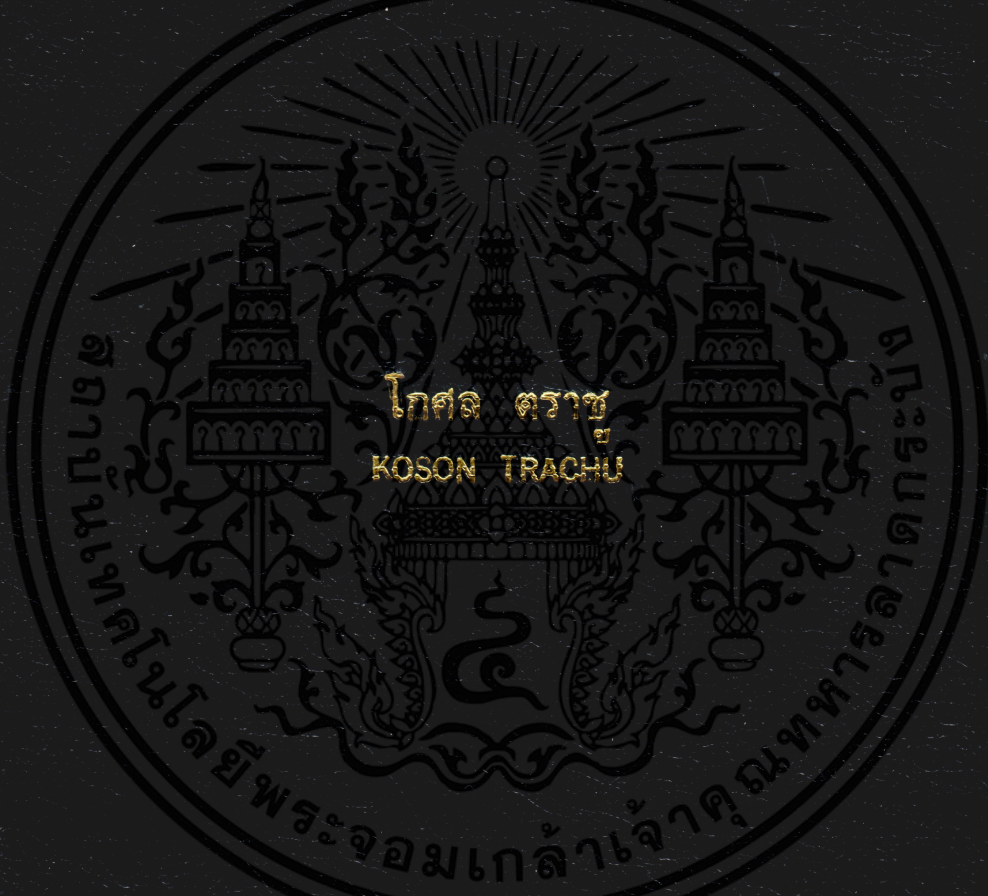
การปกปิดข้อมูลด้วยการเข้ารหัสบล็อคโค้ด และสัญญาณรบกวนแบบลำดับสุ่มเทียม ที่สร้างบน FPGA

DATA SCRAMBLE BASED ON BLOCK CODE AND PSEUDO RANDOM SEQUENCE IMPLEMENTED ON FPGA



โกศล ตราชู

KOSON TRACHU

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2544

ISBN 974-648-156-8

การปกปิดข้อมูลด้วยการเข้ารหัสบล็อคโค้ด และสัญญาณรบกวนแบบลำดับสุ่มเทียม ที่สร้างบน FPGA

DATA SCRAMBLE BASED ON BLOCK CODE AND PSEUDO RANDOM SEQUENCE IMPLEMENTED ON FPGA

โกศล ตราชู

KOSON TRACHU

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2544

ISBN 974-648-156-8

DATA SCRAMBLE BASED ON BLOCK CODE AND PSEUDO RANDOM SEQUENCE IMPLEMENTED ON FPGA

KOSON TRACHU

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING

SCHOOL OF GRADUATE STUDIES

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2001

ISBN 974-648-156-8

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองวิทยานิพนธ์

หัวข้อวิทยานิพนธ์ การปกปิดข้อมูลด้วยการเข้ารหัสบล็อคโค้ด และสัญญาณรบกวนแบบลำดับสุ่มเทียม ที่สร้างบน FPGA

DATA SCRAMBLE BASED ON BLOCK CODE AND PSEUDO RANDOM SEQUENCE IMPLEMENTED ON FPGA

ชื่อนักศึกษา นายโกศล ตราชู

รหัสประจำตัว 41061142

ปริญญา วิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชา วิศวกรรมไฟฟ้า

อาจารย์ผู้ควบคุมวิทยานิพนธ์ รศ.ดร.ฟูศักดิ์ ชีวสุวิทย์

คณะกรรมการสอบวิทยานิพนธ์		ลายมือชื่อ
รศ.พิพัฒน์	เลาหสงคราม	
ผศ.ประภาษ	อุดคิมาพันธุ์	
รศ.วิทยา	ทิพย์สุวรรณพร	
รศ.ดร.กอบชัย	เดชหาญ	
รศ.ดร.ฟูศักดิ์	ชีวสุวิทย์	

วัน/เดือนปี ที่สอบ 11 เมษายน 2544 เวลา 12.00-13.00 น.

สถานที่สอบ ณ อาคาร 12 ชั้น ชั้น 4 (ห้อง E12-404)

บัณฑิตวิทยาลัยรับรองแล้ว

(รศ.ดร.บุญวัฒน์ อัตชู)

คณบดีบัณฑิตวิทยาลัย

วันที่ 1 **เดือน** เมษายน **พ.ศ.** 2544

หัวข้อวิทยานิพนธ์	การปกปิดข้อมูลด้วยการเข้ารหัสบล็อคโค้ด และสัญญาณรบกวนแบบลำดับสุ่มเทียม ที่สร้างบน FPGA
นักศึกษา	นายโกศล ตราชู
รหัสประจำตัว	41061142
ปริญญา	วิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชา	วิศวกรรมไฟฟ้า
พ.ศ.	2544
อาจารย์ผู้ควบคุมวิทยานิพนธ์	รศ.ดร.ฟูศักดิ์ ชีวสุวิทย์

บทคัดย่อ

วิทยานิพนธ์นี้ เสนอวิธีการปกปิดข้อมูลและสัญญาณเสียงโดยการเข้ารหัสแบบบล็อคโค้ดเชิงเส้นและสัญญาณรบกวนแบบสุ่มเทียมที่สร้างบน FPGA ข่าวสารจะถูกส่งผ่านเมตริกซ์ตัวกำเนิดของตัวเข้ารหัสบล็อคโค้ดเชิงเส้น ซึ่งจะได้พาริตี้ออกมาจำนวนหนึ่ง หลังจากนั้นจึงรวมสัญญาณรบกวนแบบสุ่มเข้ากับรหัสคำ พร้อมทั้งใช้พาริตี้เป็นตัวเปิดตารางการสลับตำแหน่งบิตในรหัสคำ ทำให้ได้ข่าวสารที่ผิดเพี้ยนไปจากเดิม แล้วถูกส่งออกไปในช่องการสื่อสารครั้งละ 8 บิต ในส่วนของภาครับ จะย้อนกระบวนการของภาคส่ง ประกอบด้วยการแก้รหัสที่ผิดและการสลับตำแหน่งบิตกลับคืน ทำให้ได้รหัสข่าวสารเดิมที่เหมือนกับต้นทาง ระบบปกปิดข้อมูลนี้ถูกสร้างบนชิป FPGA ที่มีขนาดเล็ก ปลอดภัยจากการคัดลอกวงจรต้นแบบและสามารถทำงานที่เวลาจริง

Thesis Title	Data Scramble Based On Block Code And Pseudo Random Sequence Implemented On Fpga
Student	Mr.Koson Trachu
Student ID.	41061142
Degree	Master of Engineering
Programme	Electrical Engineering
Year	2001
Thesis Advisor	Assoc. Prof. Dr. Fusak Cheevasuwit

ABSTRACT

This thesis proposes a method of data scramble by using linear block code and pseudo random sequence noise. The message code will be passed to the generator matrix of linear block code for producing a code vector. The obtained code vector will be consisted of the parity digits. Then, the pseudo random sequence noise will be summed by exclusive-or to code vector in order to generate the error syndrome. After that, the process of bits permutation is applied to the message bits of each code vector. The parity bits in each code word is a key for accessing to the look up table which already assigned the new position of message bits. The sequence of scramble code vector will be divided into 8 bits for each word before transmitting them into the transmission channel. The process will be reversely treated in the encoder side for recovering the original message bits. Here, the encoder and decoder have been implemented onto the FPGA chip for reduce size and realtime processing.

กิตติกรรมประกาศ

วิทยานิพนธ์ฉบับนี้ สำเร็จลุล่วงได้เป็นอย่างดี ด้วยคำแนะนำและให้คำปรึกษาอย่างดีของ รศ.ดร. ฟูศักดิ์ ชีวสุวิทย์ ซึ่งเป็นอาจารย์ผู้ควบคุมวิทยานิพนธ์ ผู้ทำวิจัยรู้สึกซาบซึ้งในความกรุณาและขอกราบขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณคุณพ่อคุณแม่เป็นอย่างยิ่ง ที่ให้ทุกสิ่งทุกอย่างเพื่อการศึกษาและการทำงานวิจัยด้วยดีที่สุด ขอบพระคุณ อาจารย์ กิติพงค์ มะโน ในคำแนะนำที่เป็นประโยชน์ รวมถึงเอื้อเฟื้ออุปกรณ์สำหรับการทดลอง ขอขอบพระคุณ อาจารย์สักรียา ชิตวงศ์ ที่คอยเป็นพี่เลี้ยงในการทำงานวิจัยนี้จนสำเร็จลุล่วงได้ด้วยดี ขอขอบพระคุณ คุณอาโมทย์ สมบูรณ์แก้ว และทุกๆ คนใน EOL ที่คอยให้คำชี้แนะและเอื้อเฟื้อสถานที่ในการประกอบต้นแบบงานวิจัยจนสำเร็จลุล่วง ขอขอบคุณ คุณศวัสกร ไชยสุนทร ที่เป็นธุระในการจัดพิมพ์บางส่วนของต้นฉบับและเป็นกำลังใจที่ดีตลอดระยะเวลาการทำวิทยานิพนธ์ ขอบคุณ พี่ๆ เพื่อนๆ และน้องๆ ทุกคน ที่มีส่วนร่วมในงานวิจัยและวิทยานิพนธ์ฉบับนี้จนสำเร็จลุล่วงด้วยดี

และขอขอบคุณ ทบวงมหาวิทยาลัย ในฐานะเจ้าของทุนการศึกษาและวิจัย

โกศล ตราชู

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VII

สารบัญรูป VIII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและความสำคัญของปัญหา 1

1.2 วัตถุประสงค์ของการศึกษา 1

1.3 สมมติฐานของการศึกษา 2

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย 2

1.5 ขอบเขตการวิจัย 2

1.6 เนื้อหาภายในวิทยานิพนธ์ 2

บทที่ 2 หลักการและทฤษฎี 4

2.1 การเข้ารหัสลับ 4

2.1.1 ขอบเขตความต้องการของผู้ใช้ 4

2.1.2 ประสิทธิภาพและคุณภาพของสัญญาณ 5

2.1.3 การใช้งานอุปกรณ์สื่อสารที่ติดตั้งระบบเข้ารหัสลับ 6

2.1.4 การบริการและสนับสนุนการใช้งาน 7

2.2 รูปแบบการเข้ารหัสลับ 8

2.3 การแก้รหัสที่ผิด 10

2.3.1 การคำนวณทางคณิตศาสตร์ที่ใช้ในการแก้รหัสที่ผิด 10

2.3.2 การตรวจสอบความถูกต้องของรหัสคำ 10

2.3.3 ความสามารถในการตรวจแก้บิตที่ผิดในรหัสเชิงเส้น 12

2.4 บล็อคโค้ดเชิงเส้น 14

2.4.1 ข้อกำหนดของบล็อคโค้ดเชิงเส้น 14

2.4.2 เวคเตอร์สเปช 15

2.4.3 เมตริกซ์ตัวกำเนิด 16

2.4.4 เมตริกซ์สำหรับตรวจสอบพาริตี้ 17

สารบัญ(ต่อ)

หน้า

2.4.5 ซีนโดรม 18
2.4.6 การกำเนิดลำดับการนับแบบสุ่มเทียม 21
2.4.7 ตัวกำเนิดลำดับแบบชิฟท์รีจิสเตอร์ 22
2.4.8 เมตริกซ์ของการเปลี่ยนสถานะ 23
2.4.9 ตัวกำเนิดลำดับการนับแบบง่ายและแบบโมดูล่าร์ 23
2.5 อุปกรณ์ FPGAs 24
2.5.1 ส่วนองค์ประกอบของลอจิก 26
2.5.2 ส่วนที่ติดต่อกับภายนอกของ FPGAs (Input/Output Block) 26
2.5.3 กระบวนการในการพัฒนาระบบที่ประกอบด้วย FPGAs 27
2.5.4 โหมดของการโหลดโปรแกรมลงสู่ FPGAs 28

บทที่ 3 การออกแบบและการสร้าง 32
3.1 การออกแบบภาคส่ง 33
3.1.1 วงจรเข้ารหัสบล็อคโค้ดเชิงเส้น 33
3.1.2 วงจรนับแบบสุ่มเทียม 34
3.1.3 วงจรกำเนิดรูปแบบที่ผิด 35
3.1.4 วงจรสลับตำแหน่งบิตภาคส่ง 37
3.1.5 วงจรจัดรูปแบบข้อมูลสำหรับส่ง 38
3.2 การออกแบบภาครับ 39
3.1.5 วงจรจัดรูปแบบข้อมูลสำหรับภาครับ 39
3.2.1 วงจรสลับตำแหน่งบิตภาครับ 40
3.2.2 วงจรถอดรหัสบล็อคโค้ดเชิงเส้นโดยอาศัยซีนโดรม 41
3.3 วงจรสำหรับการทดลองภายนอก FPGA 43
3.4 สรุป 44

บทที่ 4 การทดลองและผลการทดลอง 45
4.1 กล่าวนำ 45
4.2 การทดสอบส่วนเข้าและถอดรหัสบล็อคโค้ดเชิงเส้น 45
4.3 การทดลองส่วนการสร้างรูปแบบที่ผิด 45

สารบัญ(ต่อ)

หน้า

4.4 การทดลองส่วนสลับตำแหน่งบิตข้อมูล 46
4.5 การทดลองส่วนจัดเฟรมข้อมูล 46
4.6 การทดสอบส่วนประกอบโดยรวมทั้งระบบ 47
4.7 การทดลองเข้ารหัสลับสัญญาณเสียง 47
4.8 การทดลองเข้ารหัสลับข้อมูลภาพ 50
4.9 สรุป 51

บทที่ 5 สรุปผลการวิจัยและข้อเสนอแนะ 52
5.1 กล่าวนำ 52
5.2 ปัญหาที่พบในการทำวิจัย 52
5.3 วิธีแก้ไขปัญหา 52
5.4 ข้อเสนอแนะในการพัฒนา 52
5.5 สรุป 53
เอกสารอ้างอิง 54
ภาคผนวก ก ผลงานที่ได้รับการตีพิมพ์ 55
ภาคผนวก ข วงจรในภาคส่ง 56
ภาคผนวก ค วงจรในภาครับ 76
ภาคผนวก ง วงจร FPGA และวงจรประกอบการทดลอง 95
ประวัติผู้เขียน 99

สารบัญตาราง

ตารางที่ หน้า

2.1 พาริตี้คี่ 11

2.2 พาริตี้คู่ 11

2.3 ซีนโดรมที่ได้จากรหัสที่ผิดไปหนึ่งบิตของรหัส (6,3) 20

2.4 รายละเอียดของ FPGAs บางรุ่น ที่ผลิตโดยบริษัทไซลิงก์ 25

2.5 โหมดต่างๆ ในการโหลดคอนฟิกกูเรชั่นเข้าสู่ FPGAs..... 29

3.1 รูปแบบรหัสที่ผิด ที่เลือกใช้ในการสร้างระบบปกปิดข้อมูล 35

3.1 (ต่อ) รูปแบบรหัสที่ผิด ที่เลือกใช้ในการสร้างระบบปกปิดข้อมูล..... 36

3.2 ตารางความจริงของวงจรสร้างรูปแบบรหัสที่ผิด..... 36

3.3 ตารางความจริงของวงจรสลับตำแหน่งบิตภาคส่ง 37

3.4 ตารางความจริงของวงจรสลับตำแหน่งบิตภาครับ 41

สารบัญรูป

รูปที่ หน้า

2.1 การเข้ารหัสลับสัญญาณแอนะลอกชนิดไม่มีการประมวลผลสัญญาณดิจิตอล 8

2.2 การเข้ารหัสลับสัญญาณแอนะลอกชนิดมีการประมวลผลสัญญาณดิจิตอล 9

2.3 การเข้ารหัสลับสัญญาณเสียงแบบดิจิตอล 9

2.4 การคำนวณทางคณิตศาสตร์ของ Galois field 10

2.5 การเพิ่มพาริตี้บิตให้กับรหัสข่าวสาร 11

2.6 การเข้ารหัสข่าวสาร 15

2.7 รูปแบบของตัวกำเนิดลำดับแบบสุ่มเทียม..... 24

2.8 รายละเอียดภายใน CLB ของ FPGAs เบอร์ XC4005E 27

2.9 ส่วนที่ติดต่อกับอุปกรณ์ภายนอกของ FPGAs ตระกูล XC4000..... 27

2.10 กระบวนการและขั้นตอนในการออกแบบระบบที่มี FPGAs..... 28

2.11 การต่อใช้งานในลักษณะสเลฟซีเรียล 30

2.12 แผนผังเวลาการป้อนข้อมูลโปรแกรมคอนฟิกในลักษณะสเลฟซีเรียล..... 30

2.13 การต่อใช้งานในลักษณะมาสเตอร์ซีเรียล 31

2.14 การต่อใช้งานในลักษณะมาสเตอร์พาราแรล 31

3.1 ผังวงจรของระบบปกปิดข้อมูล 32

3.2 วงจรเข้ารหัสบล็อคโค้ดเชิงเส้น..... 34

3.3 ผังวงจรนับแบบสุ่มเทียมโดยใช้โพลิโนเมียลตั้งต้นอันดับ 8 35

3.4 วงจรจัดเฟรมข้อมูลภาคส่ง..... 39

3.5 ผังวงจรและผังเวลาของการจัดเฟรมข้อมูลภาครับ..... 40

3.6 วงจรหาซีนโดรม 43

3.7วงจรเปิดตารางซีนโดรม 43

3.8 ผังวงจรการเชื่อมต่อภายนอก เพื่อทดสอบการปกปิดสัญญาณเสียง 44

3.9 การประกอบระบบเพื่อทดลองปกปิดข้อมูลภาพ 44

4.1 การทดสอบการเข้ารหัสบล็อคโค้ดเชิงเส้น 45

4.2 ผังวงจรการทดสอบส่วนสร้างรูปแบบที่ผิด..... 46

4.3 ผังวงจรการทดสอบร่วมกับส่วนแก้รหัสที่ผิด 46

4.4 ผังวงจรการสลับตำแหน่งบิตข้อมูล..... 46

4.5 การทดสอบการจัดเฟรมข้อมูลภาคส่งและภาครับ 47

4.6 การทดสอบการส่งข้อมูลผ่านวงจรทั้งระบบ..... 47

สารบัญรูป(ต่อ)

รูปที่ หน้า

4.7 วงจรทดสอบการเข้ารหัสลับสัญญาณเสียง 47
4.8 การทดลองเข้ารหัสลับสัญญาณสามเหลี่ยม 48
4.9 การทดลองเข้ารหัสลับสัญญาณไซน์ ความถี่ 311.5 Hz 48
4.10 การทดลองเข้ารหัสลับสัญญาณไซน์ ความถี่ 2 kHz 48
4.11 สัญญาณเสียงพูด “เอ” และสัญญาณที่ผ่านระบบปกปิดข้อมูล 49
4.12 การทดลองส่งเสียงเพลงผ่านระบบปกปิดข้อมูล 49
4.13 ผลการปกปิดข้อมูลภาพสีเทา 256 ระดับ ขนาด 256 x 256 จุด 50
4.14 การปกปิดข้อมูลภาพขาวดำ ขนาด 256 x 256 จุด 50
4.15 การปกปิดข้อมูลภาพขาวดำ ขนาด 600 x 100 จุด 50
4.15 (ต่อ) การปกปิดข้อมูลภาพขาวดำ ขนาด 600 x 100 จุด 51
ข.1. ผังวงจรของภาคส่ง 57
ข.2 วงจรเข้ารหัสบล็อคโค้ดเชิงเส้น 58
ข.3 วงจรกำเนิดลำดับการนับแบบสุ่มเทียมขนาด 8 บิต 59
ข.4 วงจรสร้างรูปแบบที่ผิด 60
ข.5 วงจรบวกเลขโมโดลู 2 ขนาด 8 บิต 61
ข.6 ผังวงจรสลับตำแหน่งบิตภาคส่งและรับ 61
ข.7 วงจรสลับตำแหน่งบิตภาคส่ง บล็อค 0 62
ข.8 วงจรสลับตำแหน่งบิตภาคส่ง บล็อค 1 63
ข.9 วงจรสลับตำแหน่งบิตภาคส่ง บล็อค 2 64
ข.10 วงจรสลับตำแหน่งบิตภาคส่ง บล็อค 3 65
ข.11 วงจรสลับตำแหน่งบิตภาคส่ง บล็อค 4 66
ข.12 วงจรสลับตำแหน่งบิตภาคส่ง บล็อค 5 67
ข.13 วงจรสลับตำแหน่งบิตภาคส่ง บล็อค 6 68
ข.14 วงจรสลับตำแหน่งบิตภาคส่ง บล็อค 7 69
ข.15 วงจรจัดเฟรมข้อมูลภาคส่ง 70
ข.16 บัฟเฟอร์สำหรับจัดเฟรมภาคส่ง 71
ข.17 วงจรกำหนดช่องเวลาการเขียน/อ่านข้อมูล 72
ข.18 วงจรมัลติเพล็กซ์ เข้า 3 ออก 1 ขนาด 8 บิต 73
ข.19 วงจรมัลติเพล็กซ์ เข้า 3 ออก 1 ขนาด 1 บิต 73

สารบัญรูป(ต่อ)

รูปที่ หน้า

ข.20 วงจรหารความถี่ เพื่อจัดช่องเวลาให้สอดคล้องกับความเร็วข้อมูล 74

ข.21 บัฟเฟอร์ข้อมูลขาออกของภาคส่ง 75

ค.1. ผังวงจรของภาครับ 77

ค.2 ผังวงจรการจัดเฟรมข้อมูล 78

ค.3 วงจรนับ 48 และหารความถี่ของภาครับ 79

ค.4 วงจรสร้างสัญญาณการเขียนและอ่านข้อมูล 80

ค.5 หน่วยความจำข้อมูลสำหรับจัดเรียงข้อมูล 8 บิตx 3 เป็น 12 บิต x 2 81

ค.6 วงจรมัลติเพล็กซ์ 2 ออก 1 ขนาด 12 บิต 82

ค.7 บัฟเฟอร์ข้อมูลเอาท์พุตของวงจรจัดเฟรมข้อมูลภาครับ 83

ค.8 ผังวงจรสลับตำแหน่งบิตภาครับ 83

ค.9 วงจรสลับตำแหน่งบิตภาครับ บล็อก 0 84

ค.10 วงจรสลับตำแหน่งบิตภาครับ บล็อก 1 85

ค.11 วงจรสลับตำแหน่งบิตภาครับ บล็อก 2 86

ค.11 วงจรสลับตำแหน่งบิตภาครับ บล็อก 3 87

ค.12 วงจรสลับตำแหน่งบิตภาครับ บล็อก 4 88

ค.13 วงจรสลับตำแหน่งบิตภาครับ บล็อก 5 89

ค.14 วงจรสลับตำแหน่งบิตภาครับ บล็อก 6 90

ค.15 วงจรสลับตำแหน่งบิตภาครับ บล็อก 7 91

ค.16 วงจรหาซีนโดรม 92

ค.17 วงจรเปิดตารางซีนโดรม 93

ค.18 วงจรบวกเลขแบบโมโดลู 2 สำหรับกลับค่าบิตที่ผิด 94

ง.1 วงจรกรองความถี่ต่ำผ่าน ความถี่ตัดที่ 3,400 เฮิรตซ์ 96

ง.2 วงจรสำหรับคอนฟิกูเรชั่น FPGA 97

ง.3 วงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอล 98

ง.4 วงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอก 98

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบัน มีการสื่อสารด้วยสียงพูด โทรสาร หรือข้อมูลข่าวสารต่างๆ ผ่านข่ายสายโทรศัพท์สาธารณะเพิ่มมากขึ้น ในจำนวนเหล่านั้น มีช่องการสื่อสารจำนวนไม่น้อยที่เสี่ยงต่อการถูกจารกรรม ดักฟังและถูกทำลายโดยผู้ประสงค์ร้ายต่อข้อมูลหรือผู้ต้องการลักลอบนำข่าวสารไปใช้ประโยชน์ โดยไม่ได้ขออนุญาต ทำให้เกิดความเสียหายต่อธุรกิจหรือความเป็นส่วนตัว

มีการนำเทคโนโลยีสมัยใหม่ ทั้งด้านการประมวลผลข้อมูลและการเข้ารหัสลับ มาช่วยในการรักษาความปลอดภัยของข่าวสารที่ถูกส่งผ่านสื่อสาธารณะ โดยระบบต่างๆ เหล่านั้นถูกพัฒนาขึ้นบนพื้นฐานของความประหยัด สะดวกต่อการใช้งาน มีขนาดเล็ก และมีหลักการที่เชื่อถือได้รองรับ เพื่อให้เกิดความถูกต้องและปลอดภัยของข้อมูลสูงสุด ระบบการสื่อสารที่ใช้ระบบปกปิดข้อมูลในปัจจุบัน ได้แก่ ระบบสื่อสารของธนาคาร การสื่อสารข้อมูลของทางทหาร การส่งข้อมูลผ่านเส้นใยแก้วนำแสง การให้บริการเคเบิลทีวี เป็นต้น

วิทยานิพนธ์ฉบับนี้ เสนอวิธีการสร้างระบบปกปิดข้อมูล โดยอาศัยหลักการแก้รหัสที่ผิดร่วมกับการสลับบิตข้อมูล และสร้างสัญญาณรบกวนแบบสุ่ม บน FPGA ทำให้ได้ขนาดวงจรรวมที่กระทัดรัด นอกจากนี้ระบบดังกล่าวถูกออกแบบให้สนับสนุนการส่งและรับข้อมูลแบบอนุกรม ตามมาตรฐาน RS232 ทำให้สามารถนำข้อมูลที่ถูกปกปิดแล้วส่งผ่านวงจรสื่อสารสาธารณะโดยใช้โมเด็มได้ทันที ทั้งยังสามารถถอดรหัสและนำข้อมูลเดิมกลับมาที่ปลายทางได้อย่างถูกต้องครบถ้วน

1.2 วัตถุประสงค์ของการศึกษา

1. เพื่อศึกษาทฤษฎีการแก้รหัสที่ผิด การกำเนิดแบบสุ่ม และ FPGA
2. เพื่อออกแบบระบบปกปิดข้อมูลโดยใช้หลักการแก้รหัสที่ผิด การกำเนิดสัญญาณรบกวนแบบสุ่ม บน FPGA
3. เพื่อทดลองสร้างระบบระบบปกปิดข้อมูลโดยใช้หลักการแก้รหัสที่ผิด การกำเนิดสัญญาณรบกวนแบบสุ่ม บน FPGA
4. ทดลองและพัฒนาระบบระบบปกปิดข้อมูลโดยใช้หลักการแก้รหัสที่ผิด การกำเนิดสัญญาณรบกวนแบบสุ่ม บน FPGA เพื่อสร้างต้นแบบที่ใกล้เคียงกับระบบที่ใช้งานได้จริง

1.3 สมมติฐานของการศึกษา

การพัฒนาระบบปกปิดข้อมูลในระบบดิจิตอลมีข้อดีกว่าระบบแอนะลอก เนื่องจากมีขนาดเล็กและให้คุณภาพของสัญญาณที่ดีกว่า[1] อุปกรณ์โปรแกรมได้ที่มีความจุสูงและแก้ไขโปรแกรมได้ง่าย เช่น FPGA มีความเหมาะสมที่จะนำมาทดลองสร้างเป็นระบบปกปิดข้อมูล เพราะจะทำให้ประหยัดเวลาและค่าใช่จ่ายสำหรับตัวอุปกรณ์ในการประกอบวงจร

1.4 ทฤษฎีหรือแนวคิดที่ใช้ในการวิจัย

เมื่อข่าวสารเดินทางผ่านสื่อที่มีสัญญาณรบกวนสูง ข่าวสารที่รับได้ที่ปลายทาง อาจมีรูปแบบที่ต่างไปจากเดิม ทำให้ผู้รับไม่สามารถนำข้อมูลที่ถูกต้องไปใช้งานจริงได้ การแก้รหัสที่ผิด ได้มีส่วนช่วยในการขจัดข่าวสารที่ผิดอันเกิดจากสัญญาณรบกวนนั้น และส่งข่าวสารที่ถูกต้องออกมา

ระบบการสื่อสารในปัจจุบัน ได้พัฒนาขึ้นมาก จนทำให้มีความผิดพลาดของข่าวสารที่เกิดจากช่องสัญญาณน้อยลงมาก อีกทั้งอุปกรณ์ปลายทาง (Terminal) ในปัจจุบันได้รวมระบบการแก้รหัสที่ผิดไว้ในตัว ซึ่งถือเป็นระบบที่มีความถูกต้องของข้อมูลสูงมาก (Error free channel)

งานวิจัยนี้ ได้รวมข้อดีของการแก้รหัสที่ผิดและคุณสมบัติที่ปราศจากการผิดพลาดของข้อมูลของช่องสัญญาณในปัจจุบัน มาสร้างเป็นระบบปกปิดข้อมูล โดยทำการเพิ่มสัญญาณรบกวนเข้าไปที่ต้นทาง ส่งผ่านระบบสื่อสาร (ที่ไม่ทำให้ข่าวสารผิดไปจากเดิม) และขจัดสัญญาณรบกวนนั้นออกที่ปลายทาง ทำให้ผู้ประสงค์ร้าย ที่ดักจับข้อมูลในช่องสื่อสาร ได้รับข่าวสารที่ผิดไปเนื่องจากมีสัญญาณรบกวนรวมอยู่ด้วย และไม่สามารถนำไปใช้งานได้

1.5 ขอบเขตการวิจัย

งานวิจัยนี้ ต้องการพัฒนาและทดสอบระบบปกปิดข้อมูล โดยอาศัยการแก้รหัสที่ผิดแบบบล็อคโค้ดเชิงเส้น บนชิพ FPGA ซึ่งเดิมถูกสร้างอยู่บนไอซีแบบแยกส่วน เพื่อเปรียบเทียบความแตกต่าง ระหว่างข่าวสารที่ส่งผ่านช่องสัญญาณโดยตรง กับข่าวสารที่ผ่านการปกปิดข้อมูลและข่าวสารที่ได้รับการถอดหัสที่ปกปิดแล้ว

1.6 เนื้อหาภายในวิทยานิพนธ์

เนื้อหาภายในวิทยานิพนธ์ฉบับนี้ ถูกแบ่งออกเป็นส่วนๆ เพื่อให้ผู้ที่สนใจมีความสะดวกแก่การศึกษาและทำความเข้าใจ โดยแบ่งออกเป็นบทต่างๆ ตามลำดับดังนี้

บทที่ 2 กล่าวถึงหลักการปกปิดข้อมูล แบบ Private-key ซึ่งเป็นวิธีที่เลือกใช้ในวิทยานิพนธ์นี้ การเข้ารหัสแบบบล็อคโค้ดเชิงเส้น การกำเนิดลำดับการนับแบบสุ่มโดยใช้โพลิโนเมียล

บทที่ 3 การออกแบบและการสร้าง กล่าวถึงวิธีการนำหลักการทางคณิตศาสตร์ในบทที่ 2 มาออกแบบเป็นวงจรดิจิตอล เพื่อสร้างวงจรบนชิพ FPGA และอุปกรณ์รอบข้าง จนได้ระบบที่

สมบูรณ์ ตั้งแต่การนำสัญญาณเสียงเข้ามาทางอินพุทที่ต้นทาง จนกระทั่งได้สัญญาณเสียงออกมาที่ปลายทาง

บทที่ 4 กล่าวถึงการทดสอบวงจรในส่วนต่างๆ ทั้งที่ทดสอบแยกเป็นภาคย่อยๆ และทดสอบวงจรทั้งหมดรวมกัน ซึ่งแสดงให้เห็นถึงประสิทธิภาพของวงจรเข้ารหัสลับที่ได้สร้างขึ้นมา

บทที่ 5 สรุปและวิจารณ์ กล่าวถึงแนวทางในการพัฒนางานวิจัยนี้ เพื่อให้สามารถนำไปใช้งานได้จริงในอนาคต

บทที่ 2
หลักการและทฤษฎี

เนื้อหาในบทนี้ กล่าวถึงหลักการที่นำมาใช้ในงานวิจัย ประกอบด้วย การเข้ารหัสบล็อคโค้ดเชิงเส้น การนับแบบลำดับสุ่ม รวมทั้ง FPGA ซึ่งเป็นอุปกรณ์หลักที่นำมาสร้างเป็นชิ้นงาน

2.1 การเข้ารหัสลับ

โครงข่ายการสื่อสารสาธารณะในปัจจุบันเช่น โครงข่ายโทรศัพท์พื้นฐาน (Public Switch Telephone Network) เป็นที่นิยมใช้งานกันมาก ด้วยเหตุผลต่างๆ หลายประการ อาทิ มีคุณภาพของเสียงพูดสูง ใช้งานง่าย อัตราค่าบริการต่ำ แต่ในทางกลับกัน ระบบสื่อสารสาธารณะ ขาดความปลอดภัยในการรักษาความลับของข้อมูล ซึ่งการดักฟังทำได้ง่าย วิทยานิพนธ์ฉบับนี้จึงศึกษาระบบเข้ารหัสลับที่จะนำมาใช้กับระบบการสื่อสารดังกล่าว

การสร้างระบบเข้ารหัสลับเสียงพูดหรือข้อมูลข่าวสาร เพื่อนำมาใช้ร่วมกับช่องสื่อสารที่มีอยู่ในปัจจุบัน หรือการสร้างเพื่อรวมเข้าไว้ในช่องการสื่อสารที่จะมีในอนาคตนั้น จะต้องพิจารณาถึงส่วนประกอบหลายประการด้วยกัน [1] ได้แก่

- ขอบเขตความต้องการของผู้ใช้
- ขอบเขตของอุปกรณ์และวงจรที่มีอยู่ในปัจจุบัน
- ความสามารถของผู้ผลิตหรือผู้ออกแบบระบบเข้ารหัสลับ
- ความยากง่ายในการติดตั้งและโอกาสที่จะเกิดการทำงานที่ผิดพลาดของระบบ

2.1.1 ขอบเขตความต้องการของผู้ใช้

โดยทั่วไปมักจะมีความขัดแย้งระหว่างระดับของความปลอดภัย คุณภาพของสัญญาณเสียง และราคาของระบบ ผู้ใช้มักต้องการระบบที่มีความปลอดภัยของเสียงพูดหรือข่าวสารระดับสูง ในราคาที่ประหยัดซึ่งเป็นเรื่องที่เป็นไปได้ค่อนข้างยาก สามารถจำแนกประเภทของความต้องการเหล่านั้นได้เป็น

- ระดับของความปลอดภัย

ในการเข้ารหัสลับเพื่อรักษาความปลอดภัยสำหรับระบบสื่อสารในระดับต่างๆ เช่นระดับผู้นำประเทศ กิจการทหาร กิจการธนาคาร หรือใช้ส่วนบุคคล จะมีระดับความยากง่ายต่างกันออกไป ในการสื่อสารส่วนบุคคล อาจใช้วิธีการเข้ารหัสลับแบบง่ายๆ ในขณะที่ผู้นำประเทศแล้ว จะต้องใช้

ระบบที่แทบจะไม่มีวิธีการดักฟังได้เลย ในการออกแบบระบบเข้ารหัสลับเพื่อให้รองรับกับความต้องการดังกล่าว มีสิ่งที่จะต้องพิจารณา คือ

- ผู้ดักฟัง

เป็นเรื่องยากที่จะบอกได้ว่าสามารถจะป้องกันการดักฟังจากผู้ใด เช่น ในกรณีกิจการทหารในสงคราม จะต้องมีการป้องกันการดักฟังจากข้าศึก ส่วนในกรณีของตำรวจ อาจใช้ป้องกันการดักฟังจากผู้กระทำผิดกฎหมายที่มีเครื่องมือสื่อสารชนิดเดียวกัน ซึ่งสามารถรู้ตัวก่อนและหลบเลี่ยงการจับกุมได้ทัน แต่ถ้าเป็นกรณีบุคคลทั่วไปที่ไม่ประสงค์ทำผิดกฎหมาย การดักฟังข่าวสารของตำรวจก็ถือเป็นเรื่องที่ไม่มีความผิดร้ายแรง

- ความสามารถของผู้ดักฟัง

ระบบเข้ารหัสลับจะมีประสิทธิภาพเพียงใด ขึ้นอยู่กับผู้ออกแบบและผู้ดักฟัง ถ้าผู้ดักฟังเป็นผู้ที่ได้รับการศึกษาและฝึกฝนมาเป็นอย่างดี ก็คงไม่สามารถป้องกันการดักฟังได้ ตัวแปรสำคัญที่ทำให้การดักฟังทำได้สำเร็จมีอยู่หลายประการ เช่น ทรัพยากร อุปกรณ์ในการดักฟัง เวลา และปริมาณข้อมูลที่ผู้ดักฟังรับได้ (ผู้ดักฟังอาจบันทึกเสียงไว้เพื่อฟังหลายๆ ครั้ง จนจับประเด็นได้)

- ปริมาณข่าวสารที่ผู้ดักฟังได้รับ

ระยะเวลาในการส่งข่าวสารอย่างต่อเนื่อง มีผลโดยตรงต่อระดับความปลอดภัยของระบบ ถ้าผู้ดักฟังได้รับข่าวสารต่อเนื่องเป็นเวลานานพอ อาจทำให้สามารถคิดหาวิธีการถอดรหัสลับได้

- ระยะเวลาในการได้รับข่าวสารของผู้ดักฟัง

ระบบสื่อสารที่ไม่มีการเปลี่ยนแปลงรูปแบบของการเข้ารหัสลับ ทำให้เกิดข้อเสียคือ ผู้ดักฟังมีเวลาในการคิดหาวิธีการถอดรหัสลับ ข่าวสารจะปลอดภัยอยู่ตราบเท่าที่ผู้ดักฟังยังไม่รู้วิธีการเข้ารหัสลับ ดังนั้นผู้ใช้ระบบไม่ควรมั่นใจในความปลอดภัยของระบบมากเกินไป และต้องเปลี่ยนรูปแบบของการเข้ารหัสลับให้บ่อยที่สุดเท่าที่จะทำได้ เหตุผลสำคัญอีกประการคือ เมื่อผู้ใช้มั่นใจว่าระบบสื่อสารนั้นไม่มีการเข้ารหัส ก็มักจะพยายามใช้คำพูดที่เข้ารหัสเพื่อปกปิดข่าวสารกันเอง แต่ถ้าผู้ใช้มีความไว้วางใจว่าระบบนั้นถูกเข้ารหัสลับแล้ว ก็จะมีการส่งข่าวสารถึงกันโดยเปิดเผย จึงเป็นจุดอ่อนสำคัญที่ทำให้ผู้ดักฟังสามารถถอดรหัสลับได้ในที่สุด

2.1.2 ประสิทธิภาพและคุณภาพของสัญญาณ

ระบบการสื่อสารสาธารณะส่วนใหญ่ เป็นช่องสัญญาณที่มีความปลอดภัยของข้อมูลต่ำ หากมีการเพิ่มความปลอดภัยให้กับช่องสัญญาณด้วยการติดตั้งระบบเข้ารหัสลับ จะทำให้คุณภาพของสัญญาณลดต่ำลง ในทางปฏิบัติไม่สามารถบอกได้ว่าตัวเข้ารหัสแต่ละตัวมีประสิทธิภาพและคุณภาพของสัญญาณดีหรือไม่ดี เนื่องจากความปลอดภัยของข้อมูลในระบบที่เข้ารหัสลับจะเป็นส่วนกลับกับคุณภาพของสัญญาณเสมอ

ในการสื่อสารผ่านช่องสัญญาณที่ไม่มีการเข้ารหัสลับ ผู้ใช้จะคิดหาวิธีการเข้ารหัสคำพูดของตนเอง เช่นการใช้รหัสวิทยุสื่อสาร ผู้รับจะได้ยินเสียงที่ชัดเจน และถอดรหัสตามความหมายที่ได้ตกลงกันไว้ล่วงหน้า แต่ในกรณีที่ติดตั้งตัวเข้ารหัสลับ คุณภาพของสัญญาณจะลดลงจนทำให้ไม่สามารถสื่อสารกันได้เข้าใจ ประสิทธิภาพในการสื่อสารจะขึ้นอยู่กับคุณภาพของสัญญาณเสียง คุณภาพของช่องการสื่อสาร และการรู้จำของผู้ฟัง โดยทั่วไปแล้ว การเพิ่มความซับซ้อนของตัวเข้ารหัสลับ จะทำให้คุณภาพของสัญญาณเสียงที่รับได้ลดลง

2.1.3 การใช้งานอุปกรณ์สื่อสารที่ติดตั้งระบบเข้ารหัสลับ

เหตุผลที่ใช้พิจารณาระบบหรืออุปกรณ์สื่อสารที่ต้องการเข้ารหัสลับ ได้แก่

1. ระบบนั้นถูกนำไปใช้ทางยุทธวิธีหรือในสถานะสงครามหรือไม่ ในการใช้งานระบบสื่อสารส่วนบุคคลอาจไม่ต้องการรักษาความลับของข้อมูล ในขณะที่การติดต่อสื่อสารในสนามรบต้องการความปลอดภัยของข่าวสารที่สูงมาก ซึ่งระบบที่นำมาใช้จะต้องคำนึงถึง
   - ความลับของอุปกรณ์หรือตัววงจร ในกรณีที่เครื่องมือสื่อสารถูกจับกุมได้โดยฝ่ายตรงข้าม ฝ่ายตรงข้ามสามารถศึกษากลวิธีในการเข้ารหัสลับและสามารถถอดรหัสลับได้ทั้งเครือข่ายการสื่อสาร
   - น้ำหนัก
   - ความทนทาน
   - ความยากง่ายในการใช้งาน
   - ความสะดวกในการเชื่อมต่ออุปกรณ์เข้ารหัสลับกับเครื่องรับ-ส่ง
   - การบำรุงรักษา
2. การนำอุปกรณ์เข้ารหัสลับไปใช้งาน ในบริเวณที่มีสัญญาณรบกวนสูง อาจทำให้ไม่สามารถถอดรหัสลับได้
3. ในแต่ละเครื่องส่ง อาจมีตัวเข้ารหัสลับได้เป็นจำนวนมาก เพื่อเพิ่มประสิทธิภาพในการเข้ารหัสลับ หรือในกรณีที่มีการสื่อสารหลายช่วง (multi hop) อาจจำเป็นต้องเข้ารหัสลับในแต่ละช่วงให้แตกต่างกัน
4. การใช้กุญแจรหัสแบบปลายทางถึงปลายทาง (End-to-End) หรือแบบเป็นช่วงๆ (Link by Link) อาจใช้วิธีการเข้ารหัสลับได้ 2 ลักษณะ คือ จะใช้การเข้ารหัสลับด้วยกุญแจแบบปลายทางถึงปลายทาง เมื่อผู้ใช้ทั้งสองด้านไม่ไว้วางใจวงจรสื่อสารช่วงกลางหรือต้องการปกปิดกุญแจรหัส แต่ถ้าผู้ใช้ทั้งหมดในระบบมีความไว้วางใจกัน หรือมีกุญแจรหัสจำนวนมากพอ ก็อาจใช้ระบบรหัสกุญแจสาธารณะได้
5. ปริมาณการสื่อสารในวงจร

6. ระบบสื่อสารเป็นแบบทางเดียว (Simplex) สองทาง (Duplex) หรือสองทางในเวลาเดียวกัน (Full duplex)

2.1.4 การบริการและสนับสนุนการใช้งาน

ในการใช้งานระบบเข้ารหัสลับให้มีประสิทธิภาพ มีองค์ประกอบต่างๆ ที่ต้องพิจารณา เช่น ความรู้ทางด้านคณิตศาสตร์และวิศวกรรมเกี่ยวกับระบบเข้ารหัสลับ วิธีการติดตั้ง การซ่อมและบำรุงรักษา การอบรมการใช้งาน เอกสารประกอบการใช้และวิธีการเปลี่ยนกุญแจรหัส

โดยทั่วไป ระบบเข้ารหัสลับจะถูกออกแบบมาเป็นอย่างดี โดยผู้เชี่ยวชาญด้านการเข้ารหัส สิ่งที่ต้องพิจารณาคือ การติดตั้ง เนื่องจากผู้ใช้ส่วนใหญ่ต้องการให้ติดตั้งในบริเวณที่เป็นความลับจริงๆ จึงอาจส่งผลถึงการซ่อมบำรุงรักษา ตลอดจนกรณีที่อุปกรณ์ในระบบอาจต้องถูกซ่อม หรือถอดเปลี่ยนโดยผู้ที่ไม่มีความรู้ด้านการซ่อมบำรุง (เช่นในสนามรบ เป็นต้น)

ระบบเข้ารหัสส่วนมากจะใช้กุญแจรหัส ซึ่งเป็นหัวใจสำคัญของระบบ ผู้ใช้จะต้องจดจำ และดูแลรักษากุญแจรหัส รวมถึงการสร้าง การแจกจ่ายและการบริหารกุญแจรหัส และต้องระลึกเสมอว่าผู้ดักฟังก็อาจมีกุญแจรหัสที่ใช้งานได้และกำลังใช้งานอยู่ด้วย

เนื่องจากกุญแจรหัสเป็นสิ่งที่สำคัญ จึงต้องพิจารณาถึงความปลอดภัยในการสร้าง และการใช้งานกุญแจรหัส

1) การสร้างกุญแจรหัส

ปัญหาประการแรกที่พบคือ ผู้ใช้จะกำหนดกุญแจรหัสได้อย่างไร โดยปกติในระบบที่เข้ารหัสลับ จะอนุญาตให้ผู้ใช้กำหนดรหัสกุญแจเอง แต่ผู้ใช้ส่วนใหญ่มักตั้งรหัสกุญแจตามชื่อของสิ่งของ หรือบุคคลที่มีความสำคัญสำหรับตน ทำให้ผู้ดักฟังสามารถคาดเดาได้โดยง่าย อีกวิธีหนึ่งที่ใช้กันก็คือ การสร้างกุญแจรหัสโดยวิธีการมาตรฐาน ซึ่งมีอันตรายในกรณีที่ผู้ดักฟังรู้กรรมวิธีในการสร้างกุญแจรหัส อันจะทำให้เขารู้กุญแจรหัสของทั้งระบบเลยทีเดียว

วิธีที่ดีในการสร้างกุญแจรหัส คือการกำเนิดรหัสกุญแจแบบสุ่ม ซึ่งทำให้ผู้ดักฟังสามารถคาดเดารูปแบบของรหัสกุญแจได้ยากขึ้น ดังนั้น ความปลอดภัยของระบบจึงขึ้นอยู่กับการทำให้ผู้ดักฟังไม่สามารถทราบกุญแจรหัสได้นั่นเอง

2) การแจกจ่ายกุญแจรหัส

เมื่อมีการสร้างกุญแจรหัส จะต้องแจกจ่ายไปยังผู้ใช้ อาจส่งโดยบุคคลหรือส่งผ่านทางระบบสื่อสาร ถ้าระบบสื่อสารมีขนาดใหญ่มาก หรือมีการเปลี่ยนแปลงกุญแจรหัสบ่อยครั้ง ก็จะทำให้เกิดปัญหายุ่งยากไม่น้อย โดยเฉพาะอย่างยิ่งต้องมั่นใจว่ากุญแจรหัสส่งไปถูกที่และถูกเวลา ในทางปฏิบัตินั้น ไม่อาจคาดหวังได้ว่าผู้ใช้จะมีความรู้เป็นอย่างดีในการใช้งานระบบสื่อสารที่มีการเข้ารหัสลับ จึงทำให้เกิดข้อผิดพลาดจากผู้ใช้ (human error) ขึ้นบ่อยครั้ง และเป็นจุดอ่อนที่ทำให้ผู้ดัก

ฟังสามารถถอดรหัสลับได้ การออกแบบระบบและวิธีการสร้างกุญแจรหัสเป็นงานที่ทำได้ยากมาก การแจกจ่ายรหัสกุญแจเป็นงานที่มีต้นทุนสูง ต้องใช้บุคลากรหรือทรัพยากรในระบบสื่อสารจำนวนมาก ต้องเชื่อมั่นและไว้วางใจในตัวสื่อนำกุญแจรหัสไปแจกจ่ายด้วย ซึ่งอาจแก้ปัญหาได้โดยการเข้ารหัสลับให้กับรหัสกุญแจนั้นอีกชั้นหนึ่ง

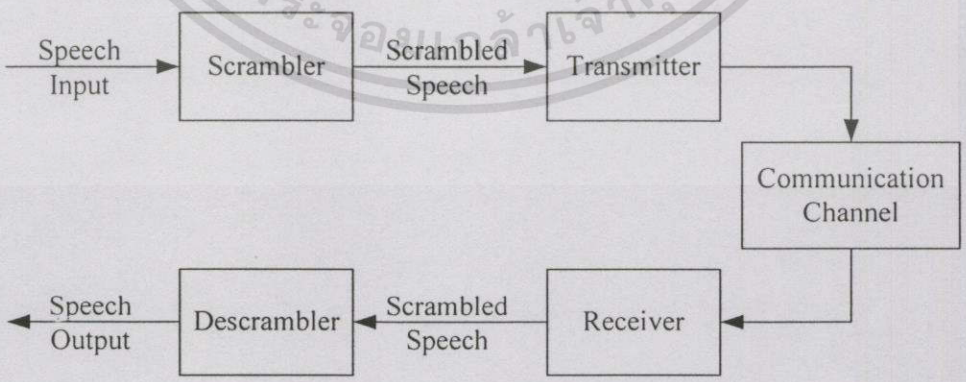
3) การบริหารกุญแจรหัส

การบริหารกุญแจรหัส หมายถึง กระบวนการในการเก็บรักษากุญแจรหัสทั้งหมดในระบบ และต้องสามารถแจกจ่ายในเวลาที่ผู้ใช้ต้องการได้ โดยมีความปลอดภัยสูง รวมทั้งการสร้างกุญแจรหัสขึ้นใหม่ ในกรณีที่ของเดิมถูกทำลาย หรือคาดว่ากุญแจรหัสจะถูกขโมยจากผู้ดักฟัง

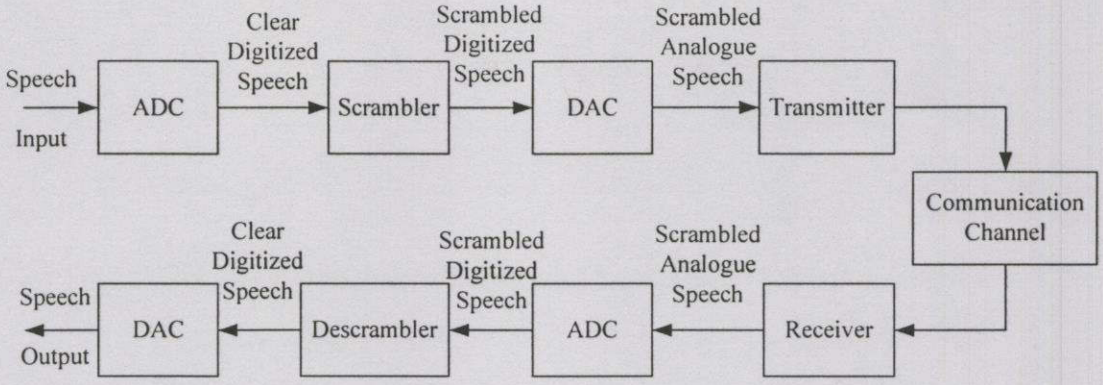
2.2 รูปแบบการเข้ารหัสลับ

ในการพิจารณาสร้างหรือนำระบบเข้ารหัสลับมาใช้งาน จะต้องพิจารณาถึงอุปกรณ์ที่จะนำมาสร้างเป็นระบบ โดยปกติจะพูดถึงตัวเข้ารหัสลับใน 2 ระบบใหญ่ๆ คือ ระบบแอนะลอกและระบบดิจิตอล ในปัจจุบันมีการนำการประมวลผลสัญญาณดิจิตอล (Digital Signal Processing) มาใช้อย่างกว้างขวาง โดยอาศัยการแปลงสัญญาณแอนะลอกเป็นสัญญาณดิจิตอล ทำให้สามารถเปลี่ยนแปลงรูปแบบในการเข้ารหัสได้ง่าย ส่วนวงจรเข้ารหัสลับแบบแอนะลอกนั้น นอกจากจะออกแบบวงจรยุ่งยากซับซ้อนแล้ว ยังมีระดับความปลอดภัยต่ำ

ข้อแตกต่างที่เห็นได้ชัดระหว่างการเข้ารหัสลับแบบดิจิตอลและแบบแอนะลอกคือ ตัวส่งสัญญาณที่เข้ารหัสลับแล้ว ในวงจรแอนะลอก จะต้องส่งสัญญาณอย่างต่อเนื่อง ส่วนในวงจรดิจิตอลจะส่งสัญญาณได้จำนวนจำกัด ขึ้นอยู่กับปริมาณข่าวสารที่บรรจุได้ในรหัสดิจิตอลเท่านั้น โดยรูปที่ 2.1 ถึง 2.3 แสดงให้เห็นถึงผังวงจรของการเข้ารหัสลับแบบต่างๆ

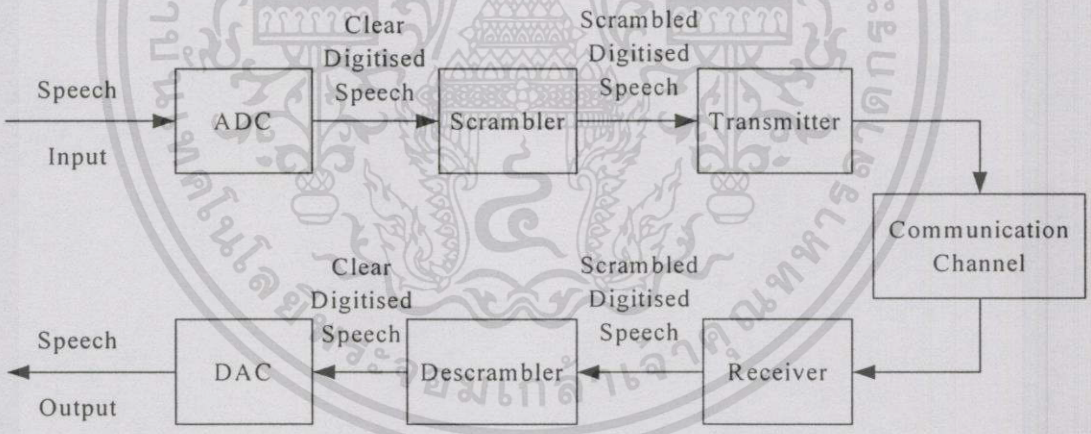


รูปที่ 2.1 การเข้ารหัสลับสัญญาณแอนะลอกชนิดไม่มีการประมวลผลสัญญาณดิจิตอล



รูปที่ 2.2 การเข้ารหัสลับสัญญาณแอนะลอกชนิดมีการประมวลผลสัญญาณดิจิตอล

รูปที่ 2.1 และ 2.2 เป็นการเข้ารหัสลับแบบแอนะลอก ความแตกต่างระหว่างรูปทั้งสอง คือ รูปแบบของสัญญาณที่ทำการเข้ารหัส ในรูปที่ 2.1 สัญญาณจะเป็นแอนะลอกตลอดกระบวนการ ส่วนระบบในรูปที่ 2.2 สัญญาณจะถูกเปลี่ยนให้อยู่ในรูปแบบของดิจิตอลก่อนเข้ารหัสลับ แล้วจึงแปลงกลับเป็นแอนะลอก เพื่อส่งในช่องสัญญาณที่เป็นแอนะลอก หลังจากนั้นจึงย้อนกระบวนการ โดยแปลงสัญญาณที่รับเข้ามาเป็นดิจิตอล ทำการถอดรหัสลับในรูปแบบดิจิตอล แล้วจึงแปลงเป็นสัญญาณเสียงพูดแอนะลอกอีกครั้ง



รูปที่ 2.3 การเข้ารหัสลับสัญญาณเสียงแบบดิจิตอล

ผังวงจรในรูปที่ 2.3 เป็นรูปแบบที่ใช้ในวิทยานิพนธ์ฉบับนี้ จัดเป็นการเข้ารหัสลับแบบดิจิตอลโดยเสียงพูดจะถูกแปลงให้อยู่ในรูปแบบดิจิตอล หลังจากนั้นจะถูกเข้ารหัสลับและส่งในรูปแบบของดิจิตอลไปยังภาครับ ที่ภาครับจะทำการถอดรหัสลับที่รับเข้ามาได้ แล้วจึงแปลงกลับเป็นสัญญาณแอนะลอกในขั้นสุดท้าย ระบบนี้จัดเป็นระบบที่ง่ายที่สุดในสามรูปแบบ จากที่ยกตัวอย่างมานั้น การพิจารณานำระบบใดไปใช้นั้นขึ้นอยู่กับว่าเสียงพูดที่ส่งในช่องทางการสื่อสารมีความสำคัญมากน้อยเพียงใด และช่องสัญญาณที่ใช้อยู่เป็นชนิดใด

2.3 การแก้รหัสที่ผิด

2.3.1 การคำนวณทางคณิตศาสตร์ที่ใช้ในการแก้รหัสที่ผิด

การออกแบบวงจรเข้ารหัสลับในวิทยานิพนธ์ฉบับนี้ อาศัยการแก้รหัสที่ผิดแบบบล็อคโค้ดเชิงเส้น อาศัยการประมวลผลสัญญาณเลขฐานสอง ซึ่งมีคณิตศาสตร์เฉพาะงานเรียกว่า Galois field [2], [3] สามารถออกแบบเป็นวงจรทางฮาร์ดแวร์ได้ง่าย เนื่องจากมีลักษณะเป็นลอจิก 0 และ 1

0	+	0	=	0
0	+	1	=	1
1	+	0	=	1
1	+	1	=	0

(ก)

0	*	0	=	0
0	*	1	=	0
1	*	0	=	0
1	*	1	=	1

(ข)

รูปที่ 2.4 การคำนวณทางคณิตศาสตร์ของ Galois field (ก) การบวก (ข) การคูณ

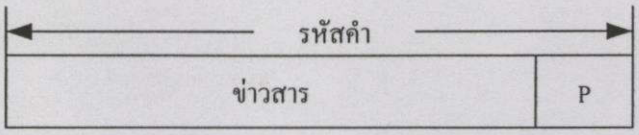
ใน Galois field จะมีการกระทำทางคณิตศาสตร์สองลักษณะคือ การบวกและการคูณ ของจำนวนเลขที่นับได้ (finite number) จะสามารถทำได้ก็ต่อเมื่อ จำนวนเลขเหล่านั้นเป็นกำลัง (power) ของจำนวนเต็มที่หารไม่ลงตัว (prime number) ดังนั้นจึงสามารถใช้กฎทางคณิตศาสตร์โดยทั่วไปมาใช้กับการบวกและการคูณใน Galois field ได้ ในกรณีนำมาสร้างเป็นวงจรลอจิกที่มีระดับ 0 และ 1 สามารถบวกและคูณตามกฎเกณฑ์ดังในรูปที่ 2.4

การคูณและการบวกในรูปที่ 2.4 เรียกว่าการบวกและการคูณแบบโมโดลู-2 (modolu-2) ซึ่งค่า 2 จะมีค่าเท่ากับ 0 และ 1 มีค่าเท่ากับ -1 ทั้งนี้ สัญลักษณ์ของการบวก การคูณ ตัวเลข 0 และตัวเลข 1 จะรวมกันเป็นฟิลด์ (field) ซึ่งเรียกว่า binary field และสามารถเขียนเป็น GF(2)

ในการสร้างวงจรลอจิกจากการบวกและคูณแบบโมโดลู 2 จากรูปที่ 2.4 สามารถใช้เกท เอ็กคลูซีฟท์-ออร์ สำหรับการบวก และใช้เกทแอนด์ สำหรับการคูณ

2.3.2 การตรวจสอบความถูกต้องของรหัสคำ

วิธีการตรวจสอบความถูกต้องของรหัสคำแบบพื้นฐานที่นิยมใช้กันคือ การตรวจสอบพาริตี้ (parity check) การตรวจสอบความถูกต้องของข้อมูลในลักษณะนี้ จะทำการเพิ่มข้อมูลตามหลังรหัสข่าวสาร (message) จำนวน 1 บิต ให้เป็นรหัสคำ (code word) ดังรูปที่ 2.5



รูปที่ 2.5 การเพิ่มพาริตี้บิตให้กับรหัสข่าวสาร

การตรวจแบบพาริตี้จะกระทำได้ 2 ลักษณะ คือ

1. กำหนดให้ผลรวมแบบโมดูล 2 ของทุกบิตในรหัสคำมีค่าเท่ากับหนึ่ง เป็นการเพิ่มพาริตี้ด้วยค่าที่ทำให้จำนวนของเลข 1 ในรหัสคำเป็นจำนวนคี่ ซึ่งเรียกว่า พาริตี้คี่ (odd parity) นั่นเอง

2. กำหนดให้ผลรวมแบบโมดูล 2 ของทุกบิตในรหัสคำมีค่าเท่ากับศูนย์ จำนวนของเลข 1 ในรหัสคำหลังจากเพิ่มพาริตี้บิตแล้วจะเป็นจำนวนคู่ เรียกว่าพาริตี้คู่ (even parity)

การตรวจสอบความถูกต้องของรหัสคำด้วยพาริตี้ทั้ง 2 ลักษณะ จะมีข้อแตกต่างกัน ดังแสดงในตารางที่ 2.1 และ 2.2

ตารางที่ 2.1 พาริตี้คี่

ข่าวสาร	รหัสคำ
0 0 0	0 0 0 1
0 0 1	0 0 1 0
0 1 0	0 1 0 0
0 1 1	0 1 1 1
1 0 0	1 0 0 0
1 0 1	1 0 1 1
1 1 0	1 1 0 1
1 1 1	1 1 1 0

ตารางที่ 2.2 พาริตี้คู่

ข่าวสาร	รหัสคำ
0 0 0	0 0 0 0
0 0 1	0 0 1 1
0 1 0	0 1 0 1
0 1 1	0 1 1 0
1 0 0	1 0 0 1
1 0 1	1 0 1 0
1 1 0	1 1 0 0
1 1 1	1 1 1 1

จากตารางทั้งสองพบว่า ในตารางที่ 2.1 ไม่ปรากฏแถวของรหัสคำที่เป็นศูนย์ทั้งหมด จึงไม่จัดเป็นรหัสแบบเชิงเส้น ดังนั้นการตรวจสอบแบบพาริตี้คี่ จะทำให้เกิดรหัสที่ไม่เป็นเชิงเส้น ในทาง

ตรงกันข้าม แถวแรกของรหัสคำในตารางที่ 2.2 มีค่าเป็นศูนย์ แสดงว่าการตรวจสอบแบบพาริตี้คี่ทำให้เกิดรหัสเชิงเส้น

การสร้างพาริตี้คู่ ทำได้โดยการนำบิตทั้งหมดของข่าวสารมาบวกกับแบบโมโดลู 2 เช่น ข่าวสารมีค่าเป็น 1 1 0 เมื่อนำมาบวกแบบโมโดลู 2 จะได้ผลลัพธ์เป็น 0 เมื่อนำมาเป็นพาริตี้ก็จะได้รหัสคำเป็น 1 1 0 0 ซึ่งมีบิตที่เป็น 1 ทั้งหมดเป็นจำนวนคู่

การตรวจสอบแบบพาริตี้นี้สามารถตรวจสอบการผิดพลาดของรหัสคำที่รับได้เพียงบิตเดียวเท่านั้น ถ้ามีจำนวนบิตที่ผิดมากกว่านั้นก็จะไม่สามารถคาดหวังผลได้ ต้องใช้วิธีการแก้รหัสที่ผิดแบบอื่นๆ เช่นการเข้ารหัสแบบบล็อคโค้ดเชิงเส้น ดังที่จะได้กล่าวรายละเอียดในหัวข้อถัดไป

2.3.3 ความสามารถในการตรวจแก้บิตที่ผิดในรหัสเชิงเส้น

เนื้อหาในส่วนนี้จะกล่าวถึงข้อกำหนดพื้นฐานของการแก้รหัสที่ผิดแบบบล็อคโค้ดเชิงเส้น เพื่อให้สามารถทำความเข้าใจหลักการที่นำมาใช้ในวิทยานิพนธ์นี้ได้ดียิ่งขึ้น

เมื่อกล่าวถึงเวคเตอร์ n-ทูเปิลแล้ว จะมีนิยามที่สำคัญคือน้ำหนัก (weight) [4] ของแฮมมิ่ง (Hamming) ซึ่งเรียกว่า $w(v)$ หมายถึงผลรวมของจำนวนบิตในรหัส v ที่มีค่าไม่เป็นศูนย์ เช่น ถ้า $v=[1\,0\,0\,1\,0\,1\,1\,0\,1]$ แล้ว จะได้ $w(v)=5$

ถ้าให้ u และ v เป็นเวคเตอร์ n-ทูเปิล ค่าระยะห่าง (distance) ระหว่าง u และ v เขียนได้เป็น $d(u,v)$ ของสองเวคเตอร์ใดๆ คือจำนวนบิตของรหัส "1" ที่แตกต่างของเวคเตอร์ทั้งสอง เช่น

$$u=[1\quad 0\quad 0\quad 1\quad 0\quad 1\quad 1\quad 0]$$

$$v=[0\quad 1\quad 0\quad 0\quad 1\quad 1\quad 0\quad 1]$$

จะได้

$$d(u,v)=6$$

ถ้านำเวคเตอร์ u และ v มาบวกกันโดยคณิตศาสตร์ของ Galois field แบบไบนารี่จะได้

$$u+v=[1\quad 1\quad 0\quad 1\quad 1\quad 0\quad 1\quad 1]$$

และสามารถหาน้ำหนักของเวคเตอร์ที่รวมกันแล้วได้เป็น

$$w(u+v)=6$$

กล่าวโดยสรุปก็คือ ระยะห่างแบบแฮมมิ่งของสองเวคเตอร์ u และ v จะมีค่าเท่ากับน้ำหนักของแฮมมิ่งที่ได้จากการรวมสองเวคเตอร์เข้าด้วยกัน คือ

$$d(u,v)=w(u+v) \tag{2.1}$$

เมื่อมีรหัสคำหรือเวคเตอร์ 2 เวคเตอร์ สามารถหาระยะห่างต่ำสุดได้เป็น $d_{\min}$ ถ้า u และ v เป็นเวคเตอร์รหัสที่อยู่ในรหัสเชิงเส้น และ $u+v$ ก็เป็นรหัสเชิงเส้นด้วย (เพราะเซ็ทของทุกเวคเตอร์รหัสต่างก็เป็นซับสเปซของ n-ทูเปิล) จากนิยามที่ว่าระยะห่างระหว่างเวคเตอร์รหัสทั้งสอง

คือ น้ำหนักของเวคเตอร์รหัสที่สาม จะได้ระยะห่างต่ำสุด $d_{\min}$ ของรหัสเชิงเส้น เท่ากับน้ำหนักต่ำสุดของเวคเตอร์รหัสที่ไม่เป็นศูนย์ ค่าระยะห่างต่ำสุดและค่าน้ำหนักต่ำสุด จะเป็นตัวกำหนดความสามารถในการแก้รหัสที่ผิดของรหัสเชิงเส้น

ในช่องสัญญาณของระบบสื่อสาร รหัสที่ผิดจะเกิดขึ้นอย่างสุ่มและอย่างเป็นอิสระ ซึ่งถือว่าเป็นสัญญาณผิดพลาดแบบสุ่ม (random error) [5], [6] การแก้รหัสที่ผิดที่ถูกออกแบบมาใช้กับช่องสัญญาณดังกล่าวเรียกว่า การแก้รหัสที่ผิดแบบสุ่ม (random error correcting codes) นอกจากนี้ยังมีการส่งข่าวสารผ่านช่องสัญญาณโทรศัพท์ การบันทึกข้อมูลลงบนเทปแม่เหล็ก แผ่นดิกส์แม่เหล็ก หรือแผ่นซีดีรอม ที่จะถูกรบกวนหรือเกิดการผิดพลาดของข้อมูลขึ้นโดยไม่รู้จำนวนและเวลาที่แน่นอน หรือเรียกว่าเป็นเกิดการผิดพลาดขึ้น เป็นช่วงระยะเวลาติดต่อกัน การแก้รหัสที่ผิดที่ถูกออกแบบมาให้ใช้กับการผิดพลาดแบบดังกล่าวเรียกว่า burst error correcting codes แต่ในส่วนนี้จะกล่าวถึงการแก้รหัสที่ผิดแบบสุ่มเท่านั้น ส่วนการแก้รหัสที่ผิดแบบอื่นๆ สามารถศึกษาได้จากหนังสือเกี่ยวกับการแก้รหัสที่ผิดทั่วๆ ไป

เมื่อพิจารณาถึงรหัสที่ส่งผ่านช่องสัญญาณ โดยให้ $v=(v_1,v_2,...,v_n)$ เป็นเวคเตอร์สำหรับส่ง และให้ $r=(r_1,r_2,...,r_n)$ เป็นเวคเตอร์รหัสที่รับได้ (ซึ่งอาจจะมีค่าใดๆ อันเกิดจากการรบกวนในช่องสัญญาณ) ที่อยู่ใน 2^n เวคเตอร์ของ n-ทูเปิล สามารถหาความแตกต่างระหว่าง u และ v ได้เป็น e คือ

$$\begin{aligned} e &= (e_1,e_2,\ldots,e_n) \\ &= r+v \\ e &= (r_1,r_2,\ldots,r_n)+(v_1,v_2,\ldots,v_n) \\ &= (r_1+v_1,r_2+v_2,\ldots,r_n+v_n) \end{aligned}$$

เมื่อ e เป็นรูปแบบของรหัสที่ผิด (error pattern หรือ error vector) ที่เกิดจากการรบกวนในช่องสัญญาณ หาก $e_i=v_i+r_i=1$ แล้ว จะได้ว่า เวคเตอร์รหัสเกิดการผิดพลาดขึ้นที่ตำแหน่งบิตที่ i เนื่องจากในเวคเตอร์รหัสใดๆ มีจำนวน n บิต จึงทำให้เกิดรูปแบบรหัสที่ผิดได้ทั้งสิ้นเป็น 2^n-1 รูปแบบ (ไม่นับรูปแบบที่มีทุกบิตเป็นศูนย์)

ตัวถอดรหัสที่ภาครับจะมีหน้าที่ตรวจหาเวคเตอร์รหัสที่ส่งมา โดยใช้เวคเตอร์ r ที่รับเข้ามา สำหรับการถอดรหัสโดยวิธีหารหัสที่คล้ายกันมากที่สุด (maximum likelihood decoding) ตัวถอดรหัสจะหาเวคเตอร์ที่มีค่าเท่ากับ v ในภาคส่ง โดยการตรวจระยะห่างของแฮมมิ่งกับเวคเตอร์ r ตัวถอดรหัสสามารถทำการแก้ไขรหัสที่ผิดจำนวน t บิต ให้กับเวคเตอร์รหัส r ที่รับเข้ามา (นั่นคือ $d(v,r)$ ต้องมีค่าต่ำสุด) ถ้ารหัสที่มีระยะห่างต่ำสุด $d_{\min}$ เป็น $2t+1$ แล้วก็จะสามารถแก้รหัสที่ผิดแบบสุ่มอยู่จำนวน t บิตได้เสมอ

จากเวคเตอร์ r ที่รับได้ ให้ v เป็นเวคเตอร์ที่ต้องการส่ง และ u เป็นเวคเตอร์ใดๆ ระยะห่างของแฮมมิ่งระหว่าง u, v และ r จะต้องเป็นไปตามสมการต่อไปนี้

$$d(v,r) + d(u,r) \geq d(u,v) \tag{2.2}$$

ถ้ามีรหัสที่ผิดอยู่จำนวน t' บิต (โดยที่ $t' \leq t$) ระยะห่างแฮมมิ่งระหว่างเวคเตอร์รหัสที่ส่ง v กับเวคเตอร์รหัสที่รับได้ r จะเป็น $d(v,r) = t'$ เมื่อ $d(u,v) \geq d_{\min} \geq 2t+1$ แล้ว จาก (2.2) จะได้

$$\begin{aligned} d(u,r) &\geq 2t+1-t' \\ d(u,r) &\geq t+1 \\ d(u,r) &\geq t' \end{aligned} \tag{2.3}$$

จากสมการ (2.2) ถ้ารูปแบบของรหัสมีบิตที่ผิดจำนวน t บิตหรือน้อยกว่า เวคเตอร์ r ที่รับได้จะมีค่าใกล้เคียงกับเวคเตอร์รหัส v มากกว่าเวคเตอร์รหัส u ดังนั้น ตัวถอดรหัสจะสามารถแก้รหัสที่ผิดและนำเวคเตอร์ที่ถูกต้องกลับมาได้ อย่างไรก็ตาม จากสมการข้างบน ตัวถอดรหัสจะไม่สามารถแก้รูปแบบของรหัสที่ผิดจำนวน l บิต เมื่อ $l \geq t+1$ ซึ่งในกรณีนี้เวคเตอร์ r จะมีค่าเข้าใกล้เวคเตอร์อื่นมากกว่าเวคเตอร์ v จากภาคส่ง และทำให้ตัวถอดรหัสทำงานผิดพลาด โดยปกติการแก้รหัสที่ผิดแบบเชิงเส้นจะทำงานแก้ที่ผิดได้อย่างถูกต้องก็ต่อเมื่อ

$$t = \frac{(d_{\min} - 1)}{2} \tag{2.4}$$

โดย t เป็นจำนวนเต็มไม่คิดทศนิยม และสามารถตรวจสอบรหัสที่ผิดในแต่ละรหัสคำได้เป็นจำนวน $(d_{\min} - 1)$ การแก้รหัสที่ผิดได้เป็นจำนวน t บิต มักนิยมเรียกว่า t-error-correcting code

2.4 บล็อคโค้ดเชิงเส้น (Linear block codes)

2.4.1 ข้อกำหนดของบล็อคโค้ดเชิงเส้น

สมมติว่าข้อมูลที่จะส่งอยู่ในรูปของเลขฐานสอง การเข้ารหัสบล็อคโค้ดเชิงเส้นจะประกอบด้วย 2 ขั้นตอนย่อย คือ

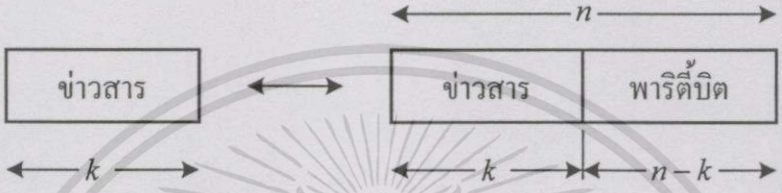
(1) แบ่งข้อมูลออกเป็นบล็อคย่อย แต่ละบล็อคจะประกอบด้วยข่าวสาร m ที่มีขนาดเป็น k บิต

(2) ทำการเข้ารหัสตามกฎของบล็อคโค้ดเชิงเส้น คือ เปลี่ยน (transform) บล็อคของข่าวสาร m เป็นรหัสคำ u ขนาด n บิต ($n > k$)

เนื่องจากแต่ละบล็อคของข่าวสารมีข้อมูลขนาด k บิต จึงมีรูปแบบของข่าวสารที่ปรากฏที่เอาต์พุตของตัวเข้ารหัสแตกต่างกันทั้งสิ้น 2^k รหัส กลุ่มของรหัสคำจำนวน 2^k นี้ถูกเรียกว่า

บล็อคโค้ด รหัสคำมีชื่อที่นิยมเรียกอีกอย่างหนึ่งว่า เวคเตอร์รหัส (code vector) เนื่องจากมีเวคเตอร์ทุกตัวขนาด n-ทูเปิลในสเปซ v_n

รหัสข่าวสารในแต่ละบล็อคขนาด k บิต ให้ชุดของรหัสที่แตกต่างกันทั้งสิ้น $2^k - 1$ รหัส (เนื่องจากจะไม่มีการนำรหัสที่เป็นศูนย์ทุกบิตมาใช้) รหัสแต่ละบล็อคจะถูกนำมาเข้ารหัสเป็นบล็อคขนาด n บิต โดยการเพิ่มบิตตรวจสอบ (เรียกว่าพาริตี้หรือรหัสแก้ไข) เข้าไปจำนวน $n-k$ บิต เพื่อใช้ในการตรวจสอบและแก้ไขที่ผิดของรหัสคำที่ภาครับ จำนวนของ $n-k$ ขึ้นอยู่กับขนาดของข่าวสารและความต้องการจำนวนบิตในการแก้รหัสที่ผิด ดังแสดงในรูปที่ 2.6



รูปที่ 2.6 การเข้ารหัสข่าวสาร

บล็อคข่าวสารขนาด n บิต นี้เรียกว่ารหัสคำ u (code word) ถ้าหากปรากฏรหัสข่าวสาร k บิต ในรูปแบบเดิมอย่างครบถ้วนอยู่ในรหัสคำ u จะเรียกว่าเป็นการเข้ารหัสแบบซิสเต็มเมติกส์ (systematic code) และยิ่งกว่านั้น ถ้ารหัสคำทั้ง 2^k เกิดจากการรวมกันของ k เวคเตอร์ของรหัสแบบอิสระเชิงเส้น (linear independence) จะเรียกรหัสคำดังกล่าวว่ารหัสบล็อคโค้ดเชิงเส้น

2.4.2 เวคเตอร์สเปซ (Vector space)

จากที่ได้กล่าวในหัวข้อ 2.3.1 ว่า $GF(2)$ ประกอบด้วยสัญญาณเลข "0" และ "1" พร้อมทั้งการ "บวก" และ "คูณ" นั้น ต่อไปให้พิจารณาลำดับของเลขฐานสอง

$$v = (v_1, v_2, v_3, ..., v_n)$$

เมื่อ v_i ในลำดับที่ i มีค่า 0 หรือ 1 เท่านั้น ลำดับดังกล่าวนี้ถูกเรียกว่า n-ทูเปิล (n-tuple) และสามารถมีลำดับของรหัส v ได้ทั้งสิ้น 2^n ลำดับที่แตกต่างกัน สามารถแสดงการบวกของรหัส 2 ตัวใดๆ ได้เป็น

$$v = (v_1, v_2, v_3, ..., v_n)$$
$$u = (u_1, u_2, u_3, ..., u_n)$$
$$v + u = (v_1 + u_1, v_2 + u_2, v_3 + u_3, ..., v_n + u_n)$$

เมื่อการรวมกันของ v และ u กระทำในไบนารี่ฟิลด์ดังในรูปที่ 2.4 $v+u$ ในลำดับที่ i ก็จะเป็นรหัส n-ทูเปิลเช่นเดียวกัน สามารถใช้กฎการสลับที่ในการบวกกับเวคเตอร์ได้เช่นกัน กล่าวคือ

$$v+u=u+v$$

ในกรณีการคูณเลขจำนวนเต็มกับเวคเตอร์ในไบนารี่ฟิลด์ขนาด n-ทูเปิล สามารถเขียนได้เป็น

$$\sigma(v_1,v_2,v_3,\ldots,v_n)=(\sigma v_1,\sigma v_2,\sigma v_3\ldots,\sigma v_n)$$

เมื่อ σv_i ที่ตำแหน่ง i ใดๆ มีค่าเป็น 0 หรือ 1 เท่านั้น และเป็นไปตามกฎการคูณดังรูปที่ 2.4 (ข)

2.4.3 เมตริกซ์ตัวกำเนิด (Generator matrix)

การสร้างพาริตี้ตรวจสอบสำหรับรหัสเชิงเส้นแบบซิสเต็มเมติกส์ (n,k) สามารถอธิบายด้วยเมตริกซ์ขนาด $k\times n$ ดังรูปแบบต่อไปนี้

โดยกำหนดให้ $p_{ij}=0$ หรือ 1 และให้ $\mathbf{I}_k$ เป็นเมตริกซ์เอกลักษณ์ (identity matrix) สำหรับ p_{ij} แล้ว เมตริกซ์ตัวกำเนิด (generator matrix) ของการเข้ารหัสแบบซิสเต็มเมติกส์สามารถเขียนได้เป็น

$$G=[\mathbf{I}_k\ \ \mathbf{P}] \tag{2.5}$$

เมื่อพิจารณาถึงบล็อคของข่าวสาร $m=(m_1,m_2,\ldots,m_k)$ โดยการใช้เมตริกซ์ตามรูปแบบใน (2.5) จะได้รหัสคำเป็น

$$\begin{aligned} u&=(u_1,u_2,\ldots,u_n)\\ &=(m_1,m_2,\ldots,m_k)G \qquad (2.6)\\ &=(m_1,m_2,\ldots,m_k)\begin{bmatrix} 1 & 0 & 0 & 0 & \ldots & 0 & p_{11} & p_{12} & \cdots & p_{1,n-k}\\ 0 & 1 & 0 & 0 & \ldots & 0 & P_{21} & p_{22} & \cdots & p_{2,n-k}\\ \vdots & & & & & \vdots & \vdots & \vdots & & \vdots\\ 0 & 0 & 0 & 0 & \ldots & 1 & P_{k,1} & p_{k,2} & \cdots & p_{k,n-k}\end{bmatrix}\end{aligned}$$

จากกฎการคูณของเมตริกซ์จะได้

$$u_i=m_i \ \text{ เมื่อ } i=1,2,\ldots,k \tag{2.7 ก}$$

และ

$$u_{k+j}=p_{1j}m_1+p_{2j}m_2+\ldots+p_{kj}m_k \tag{2.7 ข}$$

จากสมการ พบว่า ที่ $j = 1,2,...,n-k$ นั้น รหัส k บิตแรกในรหัสคำเป็นรหัสข่าวสาร m ที่ต้องการส่ง และรหัส $n-k$ บิตหลังเป็นฟังก์ชั่นเชิงเส้นของรหัสข่าวสาร เรียกรหัสจำนวน $n-k$ บิตในรหัสคำ u ว่าเป็นข้อมูลสำหรับตรวจสอบพาริตี้ของรหัสคำ เรียกสมการ (2.7 ข) ว่าเป็นสมการตรวจสอบพาริตี้ (parity-check equations) ของรหัสคำ

2.4.4 เมตริกซ์สำหรับตรวจสอบพาริตี้ (Parity-check matrix)

เมตริกซ์สำหรับตรวจสอบพาริตี้ **H** ใช้ในการตรวจสอบที่ผิดในรหัสคำที่รับได้ กล่าวคือ เมตริกซ์ **G** ขนาด $k \times n$ ในตัวเข้ารหัส จะมีเมตริกซ์ **H** ขนาด $(n-k) \times n$ ซึ่งโรว์สเปซของ **G** จะตั้งฉากอยู่กับ **H** และอินเนอร์โปรดักท์ของเวคเตอร์ในโรว์สเปซของ **G** และ **H** จะมีค่าเป็นศูนย์

ให้

$$H = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{n-k} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n} \\ \vdots & \vdots & & \vdots \\ h_{n-k,1} & h_{n-k,2} & & h_{n-k,n} \end{bmatrix} \tag{2.9}$$

และให้ $u = (u_1, u_2, ..., u_n)$ เป็นเวคเตอร์ในโรว์สเปซของ **G** จะได้

$$\mathbf{u}\mathbf{H}^{\mathrm{T}} = (0 \quad 0 \ldots 0) \tag{2.10}$$

หรือ

$$u \cdot h_i = u_1 h_{i1} + u_2 h_{i2} + u_n h_{in} = 0 \tag{2.11}$$

เมื่อ $u = (u_1, u_2, ..., u_n)$

จึงสามารถสรุปได้ว่า u จะเป็นรหัสคำที่ได้จาก **G** ถ้าเพียงแต่ $u \cdot \mathbf{H}^{\mathrm{T}} = 0$ แมทริกซ์ **H** นี้เรียกว่าเมตริกซ์สำหรับตรวจสอบพาริตี้ หรือเรียกย่อๆว่า พาริตี้แมทริกซ์ ถ้าแมทริกซ์ตัวกำเนิดของรหัสได้มาจากสมการ (2.5) พาริตี้แมทริกซ์ที่ได้จะเป็น

$$H = \begin{bmatrix} p_{11} & p_{21} & p_{k1} \\ p_{12} & p_{22} & p_{k2} \\ p_{1,n-k} & p_{2,n-k} & p_{k,n-k} \end{bmatrix}$$

$$= \left[\mathbf{P}^{\mathrm{T}} \quad I_{n-k}\right] \tag{2.12}$$

เมื่อ $\mathbf{P}^{\mathrm{T}}$ เป็นทรานสโพสเมตริกซ์ของ $\mathbf{P}$ และสามารถสร้างสมการตรวจสอบพาริตี้ (2.7ข) ได้จากเมตริกซ์ $\mathbf{H}$ โดยให้ $u=(u_1,u_2,...,u_n)$ เป็นรหัสคำของข่าวสาร $m=(m_1,m_2,...,m_k)$ เมื่อ $u_i=m_i$ ที่ตำแหน่งบิต $i=1,2,...,k$ เนื่องจาก

$$u\cdot\mathbf{H}^{\mathrm{T}}=0 \tag{2.13}$$

จะได้ว่า

$$\begin{aligned} u_{k+j} &= p_{1j}u_1+p_{2j}u_2+...+p_{kj}u_k \\ &= p_{1j}m_1+p_{2j}m_2+...+p_{kj}m_k \end{aligned}$$

เมื่อ $j=1,2,...,n-k$ ในการออกแบบรหัสเชิงเส้นนั้น ต้องทำการเลือกเมตริกซ์ $\mathbf{P}$ ที่เหมาะสมเพื่อให้มีคุณสมบัติในการแก้ที่ผิดได้ครบตามความสามารถของการแก้ที่ผิด

2.4.5 ซีนโดรม (Syndrome)

พิจารณาถึงรหัสเชิงเส้น (n,k) ที่มีเมตริกซ์ตัวกำเนิด $\mathbf{G}$ และเมตริกซ์ตรวจสอบพาริตี้ $\mathbf{H}$ ให้ u เป็นรหัสคำที่ถูกส่งผ่านช่องสัญญาณที่มีการรบกวน ที่ภาครับ ได้เวคเตอร์รหัส r ซึ่งประกอบด้วยรหัสคำเดิม u และเวคเตอร์รหัสที่ผิด e นั่นคือ

$$r=u+e \tag{2.14}$$

ที่ภาครับ จะยังไม่สามารถบอกได้ว่าเวคเตอร์ใดคือ u หรือ e ดังนั้นหน้าที่ของตัวถอดรหัส คือการหาเวคเตอร์รหัส u จากรหัสคำที่รับได้ r ซึ่งทำได้โดยการคำนวณตามสมการ

$$\begin{aligned} S &= r\cdot\mathbf{H}^{\mathrm{T}} \\ &= (e+u)\mathbf{H}^{\mathrm{T}}=e\mathbf{H}^{\mathrm{T}}+u\mathbf{H}^{\mathrm{T}} \end{aligned} \tag{2.15}$$

ถ้า u เป็นรหัสคำที่ถูกต้อง จะได้

$$u\mathbf{H}^{\mathrm{T}}=0 \tag{2.16}$$

และ

$$\mathbf{s}=\mathbf{e}\mathbf{H}^{\mathrm{T}} \tag{2.17}$$

เรียก s ว่าเป็นซีนโดรมของรหัสคำที่รับได้ r

จากสมการที่ (2.15) นั้น พบว่าซีนโดรมจะมีค่าเป็นศูนย์ เมื่อ r มีค่าตรงกับ u และมีค่าไม่เป็นศูนย์ เมื่อ r มีค่ารหัสอย่างอื่น สามารถพิสูจน์ได้โดย การสมมติรูปแบบของเวคเตอร์รหัสที่ต่างกัน แต่มีซีนโดรมเดียวกัน เช่น เมื่อมีรูปแบบที่ผิดเป็น e_1 และ e_2 จะได้ว่า

$$S_1=e_1\mathbf{H}^{\mathrm{T}} \tag{2.18 ก}$$

$$S_2=e_2\mathbf{H}^{\mathrm{T}} \tag{2.18 ข}$$

แต่ $S_1 = S_2$ จะให้

$$e_1\mathbf{H}^{\mathrm{T}} = e_2\mathbf{H}^{\mathrm{T}} \tag{2.19}$$

หรือ

$$(e_1 + e_2)\mathbf{H}^{\mathrm{T}} = 0 \tag{2.20}$$

เนื่องจาก $\mathbf{H}^{\mathrm{T}}$ จะมีค่าเป็นศูนย์ไม่ได้ ดังนั้น $e_1 + e_2$ จะต้องมีค่าเป็นศูนย์เพื่อให้สมการที่ (2.20) เป็นจริง

$$e_1 + e_2 = 0$$

ผลต่างของ e_1 และ e_2 จะมีค่าเป็นศูนย์ได้ก็ต่อเมื่อทุกบิตของ e_1 มีค่าตรงกับ e_2 จึงสรุปได้ว่า

$$e_1 = e_2$$

ซึ่งไม่ตรงตามสมมติฐานที่ว่า e_1 และ e_2 เป็นเวคเตอร์รหัสที่ต่างกัน

เมื่อนำรหัสคำ (n,k) มาคำนวณหาค่าซินโดรม จะได้ซินโดรมขนาด $n-k$ บิต ดังนั้น จะมีซินโดรมที่แตกต่างกันเป็นจำนวน $2^{n-k}-1$ และรูปแบบรหัสที่ผิดจะมีความสอดคล้องกับซินโดรมแบบหนึ่งต่อหนึ่ง

ในตัวถอดรหัสจะมีตารางซินโดรมและรูปแบบของรหัสที่ผิด ที่สอดคล้องกับซินโดรมเก็บเอาไว้ ขั้นตอนในการถอดรหัสที่ภาครับจะมีทั้งสิ้น 4 ขั้นตอนด้วยกันคือ

1. คำนวณซินโดรมของรหัสคำที่รับได้ ด้วยสมการ $s = r\mathbf{H}^{\mathrm{T}}$
2. เปิดตารางซินโดรมเพื่อดึงรูปแบบที่ผิด ที่สอดคล้องกับซินโดรมจะได้ เวคเตอร์รหัส e จากตาราง
3. คำนวณรหัสคำที่ถูกต้องจากสมการ $u = r + e$
4. ดึง k บิตแรกจากรหัสคำ r ซึ่งจะได้ข่าวสาร m ที่ถูกต้องออกมา

ตัวอย่าง ในการแก้รหัสที่ผิดมีเมตริกซ์ตัวกำเนิดเป็น

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \tag{2.21}$$

สามารถหาเมตริกซ์ตรวจสอบพาริตี้ได้เป็น

$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{2.22}$$

ถ้ารหัสคำขนาด 6 ทูเปิลส์ $u=(1\quad 1\quad 1\quad 0\quad 0\quad 0)$ เป็นเวคเตอร์รหัสที่ได้จากข่าวสาร $(1\quad 1\quad 1)$ แล้ว

$$s=u\mathbf{H}^{\mathrm{T}}=(1\quad 1\quad 1\quad 0\quad 0\quad 0)\begin{bmatrix}0&1&1\\1&0&1\\1&1&0\\1&0&0\\0&1&0\\0&0&1\end{bmatrix} \tag{2.23}$$

$$=(0\quad 0\quad 0)$$

สมมติว่า เพิ่มเวคเตอร์รหัสที่ผิด $e=(0\quad 0\quad 0\quad 0\quad 0\quad 1)$ เข้าไปในรหัสคำเดิมจะได้ $u=(1\quad 1\quad 1\quad 0\quad 0\quad 1)$ และจะได้ซีนโดรมเป็น

$$s=(1\quad 1\quad 1\quad 0\quad 0\quad 1)\begin{bmatrix}0&1&1\\1&0&1\\1&1&0\\1&0&0\\0&1&0\\0&0&1\end{bmatrix}=(0\quad 0\quad 1)$$

สามารถหารูปแบบของรหัสที่ผิดไปเป็นจำนวนหนึ่งบิตและซีนโดรมที่สอดคล้องกันได้เป็น

ตารางที่ 2.3 ซีนโดรมที่ได้จากรหัสที่ผิดไปหนึ่งบิตของรหัส (6,3)

รูปแบบรหัสที่ผิด	ซีนโดรม
0000000	000
0000001	001
0000010	010
0000100	100
0001000	110
0010000	101
0100000	011
0010001	111

ถ้าทางด้านส่งต้องการส่งข่าวสาร $m=(1\quad 0\quad 1)$ จะได้รหัสคำ u ที่ส่งออกไปในช่องสัญญาณเป็น

$$u = mG = \begin{pmatrix}1 & 0 & 1\end{pmatrix}\begin{bmatrix}1 & 0 & 0 & 0 & 1 & 1\\ 0 & 1 & 0 & 1 & 0 & 1\\ 0 & 0 & 1 & 1 & 1 & 0\end{bmatrix}$$

$$= \begin{pmatrix}1 & 0 & 1 & 1 & 0 & 1\end{pmatrix}$$

สมมติว่าที่ภาครับได้รับรหัสคำ r เป็น $\begin{pmatrix}1 & 0 & 0 & 1 & 0 & 1\end{pmatrix}$ ทางด้านรับจะคำนวณหาค่าซินโดรมจาก r ถ้าค่าซินโดรมเท่ากับศูนย์หมดทุกบิต แสดงว่า r ที่รับได้ตรงกับรหัสคำ u แต่ถ้าซินโดรมไม่เท่ากับศูนย์ ก็จะต้องเปิดตารางที่ 2.3 เพื่อหารูปแบบเวคเตอร์รหัสที่ผิดโดย

$$s = r\mathbf{H}^{\mathrm{T}} = \begin{pmatrix}1 & 0 & 0 & 1 & 0 & 1\end{pmatrix}\begin{bmatrix}0 & 1 & 1\\ 1 & 0 & 1\\ 1 & 1 & 0\\ 1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 1\end{bmatrix} = \begin{pmatrix}1 & 1 & 0\end{pmatrix}$$

หลังจากการนำซินโดรมไปเปิดตารางที่ 2.3 แล้ว พบว่ารูปแบบของเวคเตอร์รหัสที่ผิด คือ $\begin{pmatrix}0 & 0 & 1 & 0 & 0 & 0\end{pmatrix}$ จึงสามารถแก้ไขที่ผิดเพื่อให้ได้ u คืนมาโดย

$$u = r + e = \begin{pmatrix}1 & 0 & 0 & 1 & 0 & 1\end{pmatrix} + \begin{pmatrix}0 & 0 & 1 & 0 & 0 & 0\end{pmatrix}$$

$$= \begin{pmatrix}1 & 0 & 1 & 1 & 0 & 1\end{pmatrix}$$

ซึ่งเป็นรหัสคำที่ถูกต้อง เช่นเดียวกับที่ส่งมา

2.4.6 การกำเนิดลำดับการนับแบบสุ่มเทียม (Pseudo random sequence generator)

วงจรกำเนิดลำดับการนับแบบสุ่มเทียม เป็นวงจรดิจิตอลซึ่งประกอบด้วยชิฟท์รีจิสเตอร์และเกทชนิดเอ็กคลูซีฟท์-ออร์ ใช้กำเนิดลำดับของเลขฐานสองแบบสุ่มเทียม (Pseudo-random binary sequence) ถูกนำมาใช้กับระบบเข้ารหัสลับ เพื่อเพิ่มความซับซ้อนของรหัสที่ใช้ส่ง และจะนำวงจรลักษณะเดียวกันมาใช้ที่ภาครับ เพื่อหารหัสเดิมกลับคืนมา ลำดับการนับแบบสุ่มเทียมของภาคส่งและภาครับจะต้องมีการเข้าจังหวะ (Synchronize) ซึ่งกันและกัน ทั้งนี้ อาจมีตัวนับแบบสุ่มอยู่เฉพาะที่ภาคส่งเพียงด้านเดียวหรือมีอยู่ทั้งที่ภาคส่งและภาครับก็ได้

ในการเข้ารหัสลับเพื่อให้รหัสคำมีรูปแบบของการสุ่มมากที่สุดนั้น ตัวกำเนิดลำดับการนับแบบสุ่มเทียมจะต้องมีการกระจายของลำดับการนับด้วยตัวมันเองที่เพียงพอ แต่ในทางปฏิบัติแล้วการกำเนิดลำดับการนับแบบสุ่มเทียมประกอบด้วยชิฟท์รีจิสเตอร์เป็นจำนวนจำกัด จึงสามารถคาดเดาหรือทำนายลำดับการนับได้เสมอ (เหตุนี้จึงเรียกว่าเป็นการสุ่มเทียม) ฉะนั้น จึงต้องศึกษาคุณ

สมบัติของการสุ่มเพื่อให้เข้าใจถึงข้อจำกัดในการกำเนิดลำดับการนับแบบสุ่มเทียม โดยการเทียบเคียงจากการปั่นเหรียญดังนี้

1. จำนวนของเหรียญที่ปั่นแล้วได้หัว จะมีจำนวนใกล้เคียงกับจำนวนของเหรียญที่ปั่นแล้วได้ก้อย
2. โอกาสที่จะเกิดหัวหรือก้อยซ้ำๆ กันเป็นเวลาสั้นๆ จะมีบ่อยครั้งกว่าการเกิดหัวหรือก้อยติดกันเป็นเวลานานๆ กล่าวคือ การเกิดหัวหรือก้อยซ้ำกันหนึ่งครั้ง จะมีโอกาสครึ่งหนึ่ง การเกิดติดกันสองครั้งจะมีโอกาสเป็นหนึ่งในสี่ และการเกิดติดกันสามครั้งจะมีโอกาสเป็นหนึ่งในแปด เป็นต้น
3. ลำดับของเลขฐานสองที่เกิดขึ้นจากการเกิดหัวหรือก้อยในการปั่นเหรียญทั้งหมด เป็นฟังก์ชันสหสัมพันธ์ (correlative function) ชนิดพิเศษ ซึ่งจะมีลักษณะการกระจายเป็นรูประฆังคว่ำ ที่มียอดสูงสุดอยู่ตรงกลาง และมีปริมาณลดลงอย่างรวดเร็วที่ปลายทั้งสองด้าน

ถ้าให้การเกิดหัวมีค่าเป็น 1 และการเกิดก้อยมีค่าเป็น 0 ดังตัวอย่างข้างต้นแล้ว สามารถสรุปคุณสมบัติเหล่านั้น ออกมาในลักษณะของลำดับการนับแบบสุ่มเทียม s_k ที่มีคาบเวลา τ ได้ดังนี้

1. ในคาบเวลา τ ใดๆ จำนวนของ 1 หรือ 0 ที่เกิดขึ้นจะใกล้เคียงกัน
2. ในคาบเวลา τ ใดๆ จำนวนของ 1 หรือ 0 ที่เกิดขึ้นจะมีโอกาสครึ่งหนึ่ง โอกาสที่จะเกิดติดกันสองครั้งเป็นหนึ่งในสี่ โอกาสที่ติดกันสามครั้งเป็นหนึ่งในแปด เป็นต้น
3. ค่าที่ได้จากฟังก์ชันสหสัมพันธ์มี 2 ชนิดคือ
   3.1 จำนวนของ 1 ในแต่ละคาบเวลาของผลรวมของลำดับ $\{s_k + s_{k+m}\}$ จะมีค่าตรงกับทุกๆ กรณีที่ $m \neq 0$
   3.2 ผลรวมของลำดับสำหรับ $m = 0$ จะต่างไปจากกรณีของ 3.1

2.4.7 ตัวกำเนิดลำดับแบบชิฟท์รีจิสเตอร์

ตัวกำเนิดลำดับแบบชิฟท์รีจิสเตอร์ (Shift Register Generator : SRG) เป็นระบบที่ทำงานได้ด้วยตนเอง และไม่แปรผันตามเวลา (time invariant system) โดยจะประกอบด้วยชิฟท์รีจิสเตอร์และเกทชนิดเอ็กซ์คลูซีฟ-ออร์ ซึ่งสามารถหาค่าสถานะปัจจุบันของชิฟท์รีจิสเตอร์ได้โดยการบวกแบบโมดูล 2 กับสถานะก่อนหน้านี้

ความยาว (length) ของตัวกำเนิดลำดับแบบชิฟท์รีจิสเตอร์ L สามารถหาได้จากจำนวนของรีจิสเตอร์ในระบบ เวกเตอร์สถานะ (state vector) ของ SRG ที่มีความยาว L และสถานะของเวคเตอร์ลำดับที่ k หรือเขียนย่อว่า d_k ซึ่งกำหนดได้โดยเวคเตอร์ความยาว L ที่แทนสถานะของชิฟท์รีจิสเตอร์ใน SRG ที่เวลา k นั่นคือ

$$d_k \equiv \begin{bmatrix} d_{0,k} & d_{1,k} & \cdots & d_{L-1,k} \end{bmatrix}^T \tag{2.24}$$

เมื่อสถานะ $d_{i,k}$, $i = 0,1,...,L-1$ แทนค่ารีจิสเตอร์ตำแหน่งที่ i ใน SRG ที่เวลา k

2.4.8 เมตริกซ์ของการเปลี่ยนสถานะ (State transition matrix)

สำหรับ SRG ที่มีความยาว L เมตริกซ์ของการเปลี่ยนสถานะ $\mathbf{T}$ จะถูกกำหนดโดยเมตริกซ์ขนาด $L \times L$ ที่แสดงความสัมพันธ์ระหว่างเวคเตอร์สถานะ d_k และ d_{k-1} หรืออาจเขียนได้เป็น

$$d_k = \mathbf{T} \cdot d_{k-1} \tag{2.25}$$

ตัวกำเนิดลำดับการนับแบบชิฟท์รีจิสเตอร์ คือตัวกำเนิดค่าที่ได้จากเมตริกซ์ของการเปลี่ยนสถานะ $\mathbf{T}$ และเวคเตอร์สถานะเริ่มต้น d_0 เมตริกซ์ $\mathbf{T}$ จะเป็นตัวกำหนดรูปแบบของ SRG และเวคเตอร์สถานะเริ่มต้น d_0 จะเป็นตัวกำหนดสถานะเริ่มต้นของลำดับที่กำเนิดออกมา หรือกล่าวได้อีกนัยหนึ่งก็คือ ลักษณะเฉพาะของ SRG แต่ละตัวถูกกำหนดจากเมตริกซ์ของการเปลี่ยนสถานะ $\mathbf{T}$ เพียงอย่างเดียว และลำดับการนับที่ได้จาก SRG ก็จะเกิดจากการกำหนดเวคเตอร์เริ่มต้น d_0 ในครั้งแรกเท่านั้น

2.4.9 ตัวกำเนิดลำดับการนับแบบง่ายและแบบโมดูล่าร์

ในหัวข้อนี้ จะพิจารณาลักษณะของตัวกำเนิดลำดับการนับแบบสุ่มเทียม (SRG) 2 ชนิด คือตัวกำเนิดลำดับการนับแบบสุ่มเทียมแบบง่าย (Simple SRG : SSRG) และ ตัวกำเนิดลำดับการนับแบบสุ่มเทียมแบบโมดูล่าร์ (Modular SRG : MSRG)

ตัวกำเนิดลำดับการนับแบบสุ่มเทียมแบบง่าย มีเมตริกซ์ของการเปลี่ยนสถานะเป็น

$$\mathbf{T}_s = \mathbf{A}^t \Psi(x) \tag{2.26}$$

สำหรับสเปซของลำดับ $V[\Psi_{(x)}]$

มีหลายวิธีที่จะใช้กำหนด SSRG โดยปกติ มักจะใช้โพลิโนเมียลคุณลักษณะ (characteristic polynonial) $C(x)$ หรือที่เรียกว่ารีเซ็ตโพลิโนเมียล (reset polynomial) ความสัมพันธ์ระหว่างสเปซของลำดับ $\Psi(x)$ และ $C(x)$ หาได้จาก

$$\Psi(x) = x^L C(x^{-1}) \tag{2.27}$$

ตัวกำเนิดการนับแบบสุ่มเทียมแบบโมดูล่าร์ มีเมตริกซ์ของการเปลี่ยนสถานะเป็น

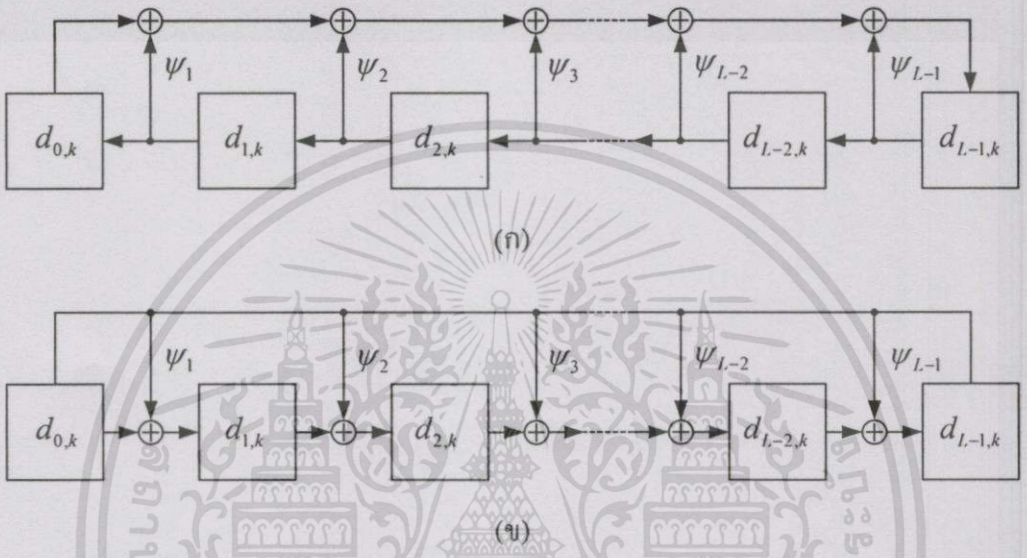
$$\mathbf{T}_M = A\Psi(x) \tag{2.28}$$

สำหรับสเปซของลำดับ $V[\Psi_{(x)}]$

วิธีการที่จะใช้กำหนดค่า MSRG มีหลากหลายรูปแบบเช่นเดียวกับ SSRG แต่ที่นิยมใช้กันก็คือ เมตริกซ์ตัวกำเนิด (generator polynomial) $G(x)$ ความสัมพันธ์ระหว่าง $\Psi(x)$ และ $G(x)$ หาได้จากสมการข้างล่างนี้

$$\Psi(x) = G(x) \tag{2.29}$$

การนำตัวกำเนิดลำดับแบบสุ่มเทียมทั้งสองชนิดมาสร้างเป็นวงจรดิจิตอล แสดงดังรูปที่ 2.7



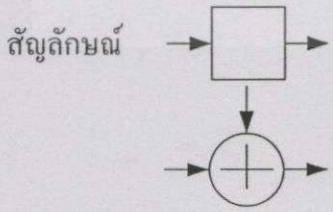
รูปที่ 2.7 รูปแบบของตัวกำเนิดลำดับแบบสุ่มเทียม ในสเปซของลำดับ $V\left[\sum_{i=0}^{L}\Psi_i x^i\right]$

(ก) ตัวกำเนิดลำดับการนับแบบสุ่มเทียมแบบง่าย

(ข) ตัวกำเนิดลำดับการนับแบบสุ่มเทียมแบบโมดูลาร์

จากรูปที่ 2.7 จะพบว่า

ψ_i เมื่อ $i = 0,1,2,\ldots, L-1$ เป็นการเชื่อมต่อสำหรับเกทเอ็กคลูซีฟ-ออร์ ซึ่งจะมีการเชื่อมต่อเกทดังกล่าวเมื่อ $\psi_i = 1$ และไม่มีการเชื่อมต่อเมื่อ $\psi_i = 0$ โดยที่



2.5 อุปกรณ์ FPGAs (Field Programmable Gate Arrays)

FPGAs เป็นวงจรรวมแบบโปรแกรมได้ขนาดใหญ่ สามารถโปรแกรมให้ทำงานเป็นวงจรเชิงเลข โดยการโหลดข้อมูลจากภายนอกลงบนสแตติกแรมที่อยู่ภายใน และสามารถโปรแกรมใหม่

ได้หลังจากการรีเซ็ตด้วยสัญญาณไฟฟ้า FPGAs เป็นอุปกรณ์ที่ประหยัดพลังงานไฟฟ้า มีความจุเกทพื้นฐานสูง อีกทั้งยังสามารถรักษาความลับของวงจรภายในได้เป็นอย่างดี

วงจรรวม FPGAs ที่นำมาใช้ในงานวิจัยนี้ ผลิตโดยบริษัทไซลิงก์ (Xilinx) [11], [12] และ [13] ซึ่งค้นคว้าร่วมกับบริษัทเอ็มเอ็มไอ (MMI) สร้างเป็นอุปกรณ์ที่บรรจุวงจรรวมที่มีความหนาแน่นของเกท 600 ถึง 2500 เกท ดังตารางที่ 2.4

ตารางที่ 2.4 รายละเอียดของ FPGAs บางรุ่น ที่ผลิตโดยบริษัทไซลิงก์

FPGAs	จำนวนเกทโดยประมาณ	จำนวน I/Os	จำนวนฟลิบฟลอบ	แรม bits	จำนวนของ CLBs
XC2064	1,000	58	122	0	64
XC2018	1,500	74	174	0	100
XC3020/3120	1,800	64	256	0	64
XC3030/3130	2,700	80	360	0	100
XC3042/3142	3,700	96	480	0	144
XC3064/3164	5,500	120	688	0	244
XC3195	9,000	176	1,320	0	484
XC4002A	2,000	64	256	2,048	64
XC4003/4003A	3,000	80	360	3,200	100
XC4003H	3,000	160	200	3,200	100
XC4004	4,000	960	480	4,608	144
XC4005/4005A	5,000	122	616	6,072	196
XC4005H	5,000	192	392	6,272	196
XC4006	6,000	128	768	8,192	256
XC4010	10,000	160	1,120	12,800	400
XC4013	13,000	192	1,536	18,432	576
XC4025	25,000	256	2,560	32,768	1,024

FPGAs มีโครงสร้างภายในใกล้เคียงกับสถาปัตยกรรมของเกทอะเรย์ (Gate Array Logic; GAL) มาก สามารถโปรแกรมและลบคอนฟิกกูเรชั่น (configuration) ภายในสแตติกแรม (static RAM) ได้โดยใช้สัญญาณไฟฟ้าและการดึงข้อมูลเลขฐานสิบหกมาจากภายนอก เช่น parallel EPROM หรือ serial PROM นับว่ามีความแตกต่างจาก EPLD หรือ PAL ที่มี EPROM บรรจุอยู่ภายในตัวเอง

ภายใน FPGAs จะมีการจัดเรียงลอจิกเซลล์เป็นเมตริกซ์และล้อมรอบด้วยอินพุต-เอาท์พุต เซลล์ ซึ่งเรียกว่า CLB (Configuration Logic Block) FPGAs ตัวแรกที่ผลิตโดยบริษัทไซลิงก์คือ เบอร์ XC2064 (ตระกูล XC2000) ซึ่งมีเซลล์จำนวน 64 เซลล์ หลังจากนั้นจึงได้ผลิต FPGAs ตระกูล XC3000 และ XC4000 ที่มีจำนวนเกทมากขึ้นและทำงานได้ดีขึ้นตามลำดับ

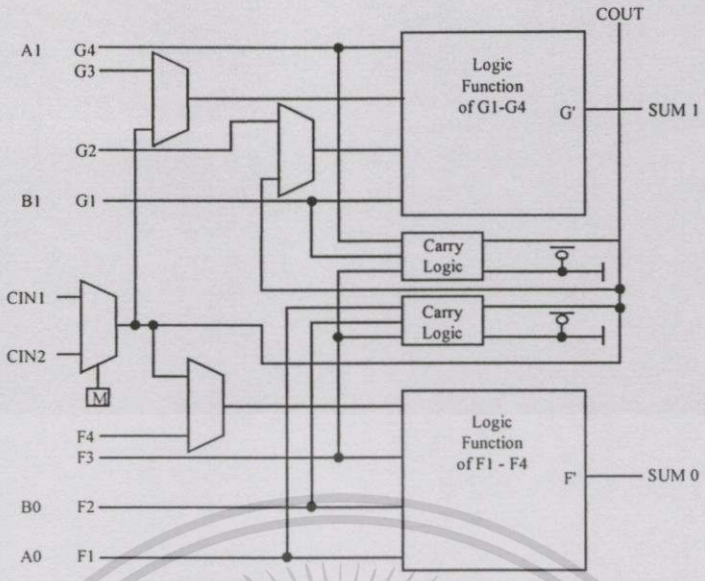
2.5.1 ส่วนองค์ประกอบของลอจิก (Configuration Logic Block)

CLB จะถูกจัดเรียงกันเป็นแบบเมตริกซ์อะเรย์ขนาด $M \times N$ ในขั้นตอนของการออกแบบนั้น จะมีการจัดวาง CLB และเชื่อมต่อ CLB ถึงกันโดยผ่านทางสายสัญญาณแนวนอนและแนวตั้ง การโปรแกรม FPGAs ก็คือการกำหนดว่าแต่ละ CLB จะเชื่อมต่อถึงกันในรูปแบบใด อาจทำได้โดยการโปรแกรมด้วยมือ หรือใช้โปรแกรมสำเร็จรูปทำให้โดยอัตโนมัติ ซึ่งจะได้ผลลัพท์เป็นข้อกำหนดการทำงานของ FPGAs อยู่ในแฟ้มข้อมูลที่เรียกว่า configuration file ที่บรรจุโครงร่าง (net list) ภายใน FPGAs ตามลักษณะของวงจร โปรแกรมสำเร็จรูปรุ่นใหม่ๆ จะมีความสามารถในการลดรวม (optimize) สมการลอจิกให้กระชับ เป็นผลให้สามารถบรรจุวงจรขนาดใหญ่ขึ้นลงใน FPGAs ตัวเดิมได้

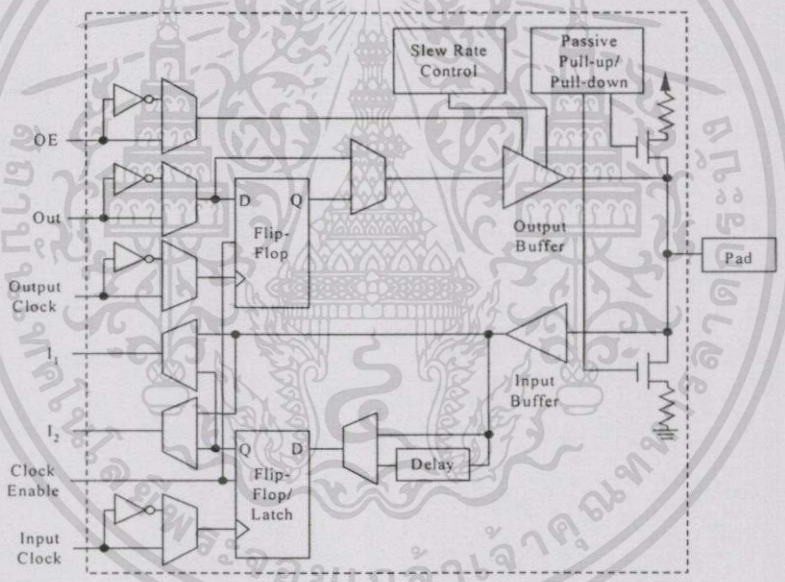
การใช้โปรแกรมสำเร็จรูปช่วยออกแบบและสร้างโครงข่ายภายใน FPGAs นั้น จะได้แฟ้มข้อมูลเอาท์พุทที่สำคัญคือ แฟ้มกระแสข้อมูล (bit stream) ซึ่งสามารถนำไปโหลดลงสู่หน่วยความจำของ FPGAs หรือโปรแกรมลงบน EPROM เพื่อให้ FPGAs ดึงไปใช้งานขณะเริ่มทำงานได้ รูปที่ 2.8 แสดงรายละเอียดภายใน CLB ของ FPGAs ตระกูล XC4000

2.5.2 ส่วนที่ติดต่อกับภายนอกของ FPGAs (Input/Output Block)

ส่วนติดต่อกับวงจรภายนอกของ FPGAs สร้างขึ้นจาก programmable input/output devices (IOBs) IOB แต่ละตัวสามารถถูกแก้ไขได้อย่างอิสระโดยจะให้เป็นอินพุทแบบ 3 สถานะ หรือพอร์ทแบบสองทิศทางก็ได้ นอกจากนั้น ยังสามารถโปรแกรมให้รู้จักระดับสัญญาณแบบ TTL และ CMOS ได้ IOB แต่ละตัวจะมีฟลิบฟลอบที่สามารถใช้เป็นบัฟเฟอร์ได้ทั้งอินพุทและเอาท์พุท FPGAs จะมี IOB ตั้งแต่ 64 ถึง 144 ตัว ขึ้นอยู่กับตระกูลของมัน รูปที่ 2.9 แสดง IOBs ของ FPGAs ตระกูล XC4000



รูปที่ 2.8 รายละเอียดภายใน CLB ของ FPGAs เบอร์ XC4005E



รูปที่ 2.9 ส่วนที่ติดต่อกับอุปกรณ์ภายนอกของ FPGAs ตระกูล XC4000

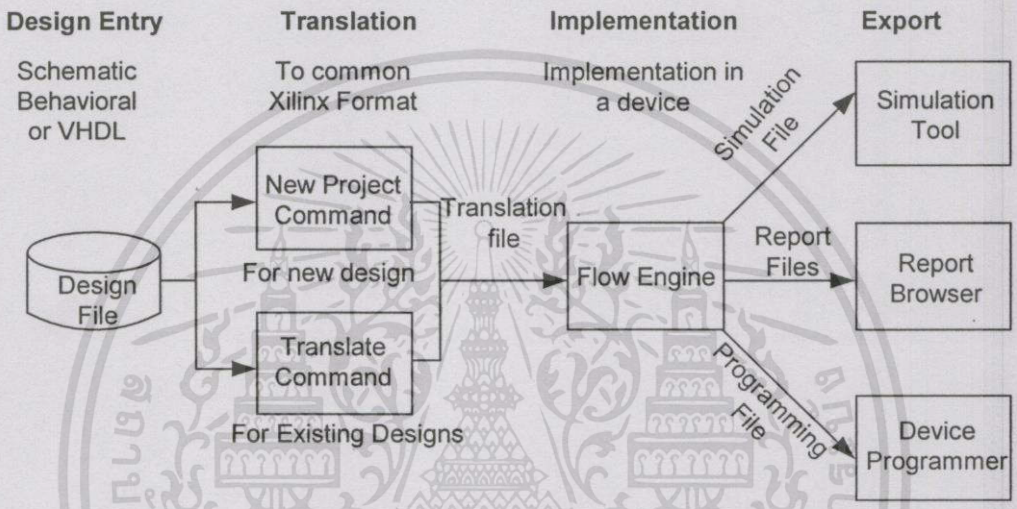
2.5.3 กระบวนการในการพัฒนาระบบที่ประกอบด้วย FPGAs

ในการสร้างระบบที่มี FPGAs เป็นส่วนประกอบ จะมีขั้นตอนในการทำงานที่ชัดเจน กล่าวคือจะเริ่มต้นจากการสร้างแฟ้มการออกแบบ (design file) ซึ่งอาจจะออกแบบวงจรด้วย Viewlogic หรือ OrCAD การบรรยายการเชื่อมต่อด้วย Xilinx ABEL หรือ PLUSASM รวมถึงการเขียนด้วยภาษา VHDL แล้วแปลด้วย Viewlogic Synthesis เป็นต้น

หลังจากขั้นตอนการออกแบบแล้ว ก็จะแปลงเป็นแฟ้มสำหรับโหลดสู่ FPGAs ด้วยขั้นตอนพื้นฐานต่อไปนี้

1. แปลงแฟ้มต้นฉบับเป็นแฟ้มชนิด XFF (Translation)
2. แปลแฟ้มให้เข้ากับอุปกรณ์ FPGAs เป้าหมาย (Implementation)
3. ส่งออกแฟ้มข้อมูลไปยังซอฟท์แวร์วิเคราะห์ผังเวลา (Export)
4. รายงานสถานะของแต่ละขั้นตอนในการออกแบบ (Report Generation)
5. วิเคราะห์ผังเวลาของการออกแบบ (Timing Analysis)

รูปที่ 2.10 แสดงกระบวนการและขั้นตอนในการออกแบบ รวมทั้งแฟ้มขาเข้าและขาออกที่ได้จากกระบวนการพัฒนาระบบด้วยซอฟท์แวร์



รูปที่ 2.10 กระบวนการและขั้นตอนในการออกแบบระบบที่มี FPGAs

2.5.4 โหมดของการโหลดโปรแกรมลงสู่ FPGAs

เมื่อได้ออกแบบและสร้างวงจรแล้ว จะต้องทำการโหลดโครงข่ายของวงจรลงสู่หน่วยความจำของ FPGAs เพื่อให้ FPGAs เริ่มต้นทำงาน สำหรับ FPGAs ที่ใช้ในงานวิจัยนี้ สามารถโหลดแฟ้มที่มีนามสกุล .BIT (โดยจะต้องคอมไพล์แฟ้ม LCA โดยโปรแกรม MAKEBIT แล้วจึงสามารถโหลดลงสู่ FPGAs) ได้โดยวิธีการต่างๆ ดังตารางที่ 2.5

เมื่อเริ่มจ่ายไฟเข้าตัว FPGAs จะทำการลบข้อมูลหน่วยความจำที่ใช้ในคอนฟิก (configuration memory) ตรวจสอบลักษณะการคอนฟิกตามตารางที่ 2.5 ว่าเป็นแบบอนุกรมหรือขนาน หลังจากนั้นจะเริ่มทำการโปรแกรมคอนฟิกสัญญาณ DONE/PROGRAM เป็น “0” ซึ่งอยู่ระหว่างโปรแกรมและเมื่อข้อมูลตรงกับที่ส่วนหัวของข้อมูลคอนฟิกสัญญาณ DONE/PROGRAM จะเป็น “1” ซึ่งหมายถึงโปรแกรมทำการคอนฟิกเสร็จสิ้น

ตารางที่ 2.5 โหมดต่างๆ ในการโหลดคอนฟิกกูเรชั่นเข้าสู่ FPGAs

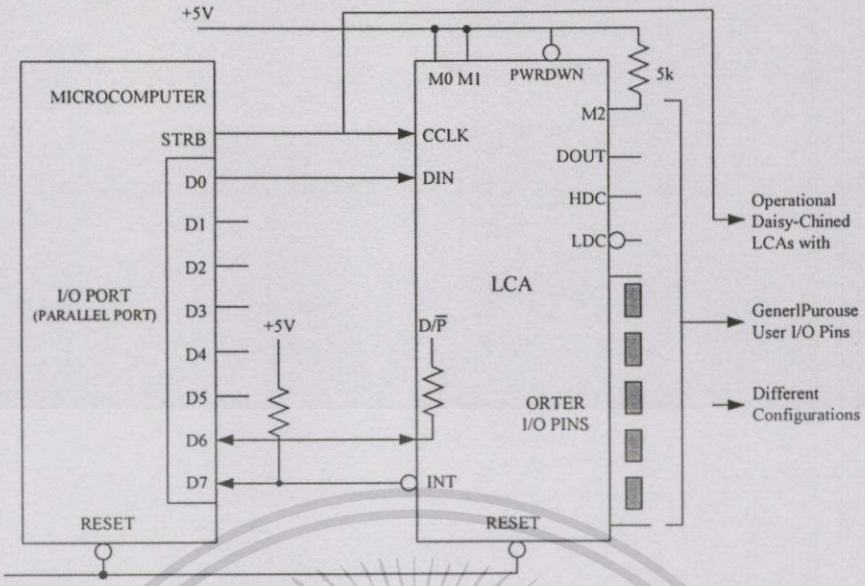
Mode	M2	M1	M0	CCLK	DATA
Slave Serial	1	1	1	input	Bit – Serial
Master Serial	0	0	0	output	Bit – Serial
Master Parallel up	1	0	0	output	Byte - Wide , 00000 up
Master Parallel down	1	1	0	output	Byte – Wide , 00000 down
Peripheral Synchr.	0	1	1	input	Byte - Wide
Peripheral Asynchr.	1	0	1	output	Byte - Wide
Reserved	0	1	0	-	-
Reserved	0	0	1	-	-

ก. การใช้งานในลักษณะสเลฟซีเรียล

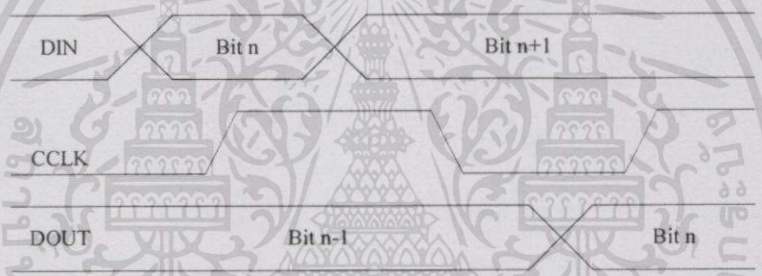
การต่อในลักษณะสเลฟซีเรียลเหมาะสมกับวงจรที่ออกแบบมาเพื่อทำงานร่วมกับไมโครคอมพิวเตอร์ ทั้งนี้เพราะ FPGAs ได้ใช้ความสามารถของไมโครคอมพิวเตอร์ในการเก็บและส่งข้อมูลคอนฟิกให้ เพียงแต่ต้องเขียนโปรแกรมเพื่อส่งโปรแกรมคอนฟิกให้เพิ่มลักษณะการต่อในลักษณะนี้เป็นดังรูปที่ 2.11 ซึ่งไมโครคอมพิวเตอร์จะสร้างสัญญาณเพื่อทำการคอนฟิกให้กับอุปกรณ์ FPGAs การป้อนโปรแกรมคอนฟิกให้ FPGAs ทำได้โดยต่อสัญญาณ Strobe เข้ากับขา CCLK และพอร์ต D0 เข้ากับขา DIN สร้างสัญญาณคล็อกป้อนที่ขา CCLK และป้อนโปรแกรมคอนฟิกแบบอนุกรมเข้าที่ขา DIN ดังผังเวลาในรูปที่ 2.12

ข. การใช้งานในลักษณะมาสเตอร์ซีเรียล

ในส่วนของการต่อใช้งานแบบมาสเตอร์ซีเรียล จะต่างจากการต่อแบบสเลฟซีเรียล คือใช้ PROM เบอร์ XC17XXX เป็นตัวเก็บโปรแกรม ทำให้ไม่ต้องเสียเวลาเขียนโปรแกรมเพื่อทำการคอนฟิกทุกครั้งที่มีการใช้งาน วิธีการเขียนโปรแกรมคอนฟิกลง PROM สามารถทำตามขั้นตอนดังนี้ คือ เมคบิต (MakeBits) เพื่อสร้างแฟ้ม .BIT จากวงจรที่ออกแบบ และใช้โปรแกรม MakePROM สร้างแฟ้ม .Hex แล้วทำการโปรแกรมลง PROM ด้วยอุปกรณ์โปรแกรม PROM ที่มาพร้อมกับตัวโปรแกรมของ ไซลิงค์



รูปที่ 2.11 การต่อใช้งานในลักษณะสเลฟซีเรียล



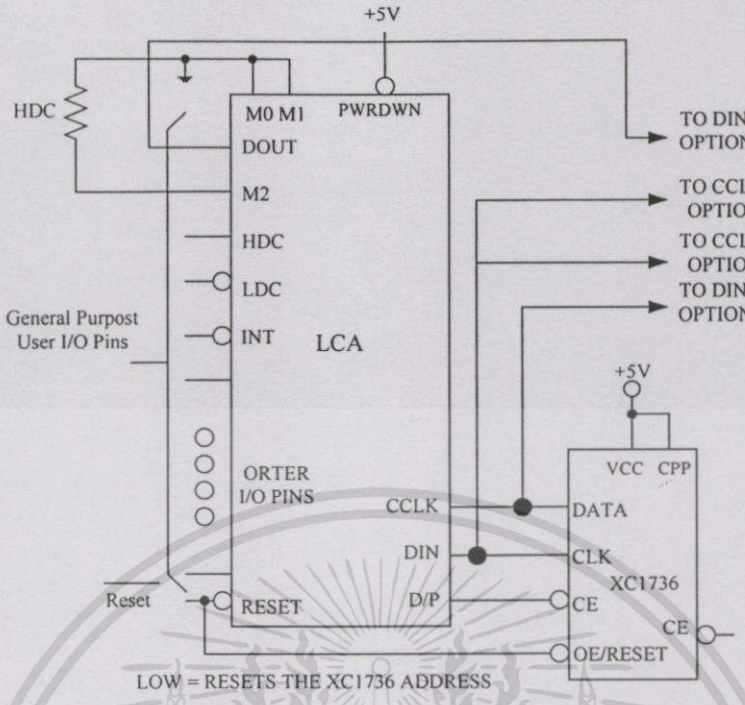
รูปที่ 2.12 แผนผังเวลาการป้อนข้อมูลโปรแกรมคอนฟิกในลักษณะสเลฟซีเรียล

ค. การใช้งานในลักษณะขนาน

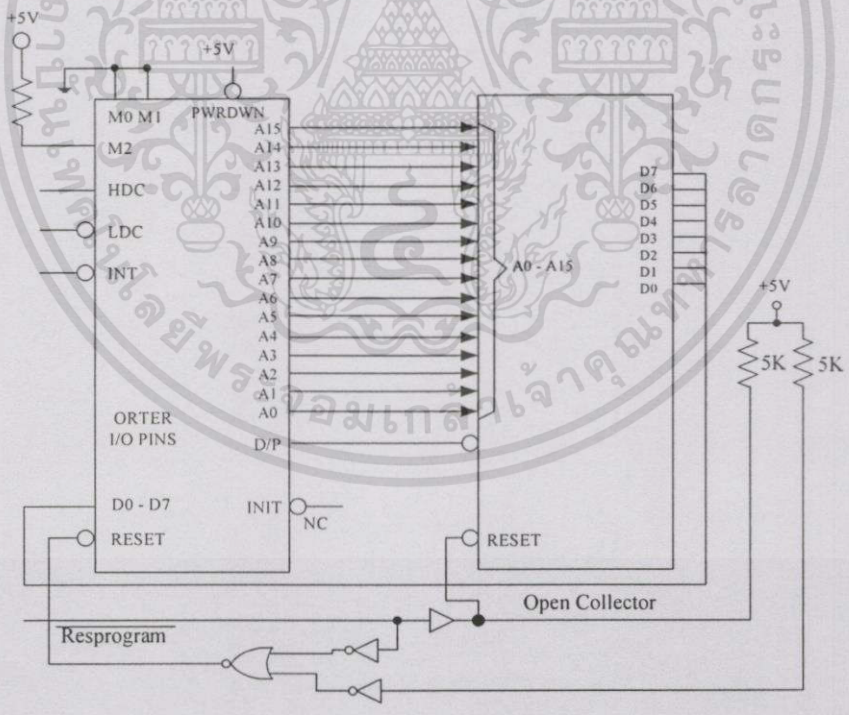
รูปที่ 2.14 แสดงวงจรสำหรับการคอนฟิก FPGAs แบบขนาน D0-D7 เป็นขารับข้อมูลที่ใช้ในการคอนฟิก A0-A15 เป็นแอดเดรสที่ FPGAs สร้างให้กับ EPROM เพื่ออ่านข้อมูลมาเก็บไว้ในสแตติกแรม (static ram) แอดเดรสทั้ง 16 เส้น ไม่จำเป็นต้องต่อให้ครบก็ได้ ขึ้นอยู่กับขนาดหน่วยความจำ EPROM ที่ใช้และสามารถกำหนดให้นับขึ้นหรือลงได้

ง. การดาวน์โหลดด้วย XChecker

กระบวนการพัฒนาในงานวิจัยนี้ ใช้เครื่องคอมพิวเตอร์โหลดแฟ้ม .BIT ลงสู่หน่วยความจำของ FPGAs ผ่านทางพอร์ตสื่อสารอนุกรม ชนิด DB-9 และยังสามารถอ่านค่าลอจิกภายในตัว FPGAs เพื่อทดสอบความถูกต้องในการออกแบบวงจรและการทำงานที่เวลาจริงได้ โดยการใช้ XChecker และซอฟท์แวร์ design manager ของไซลิงก์



รูปที่ 2.13 การต่อใช้งานในลักษณะมาสเตอร์ซีเรียล

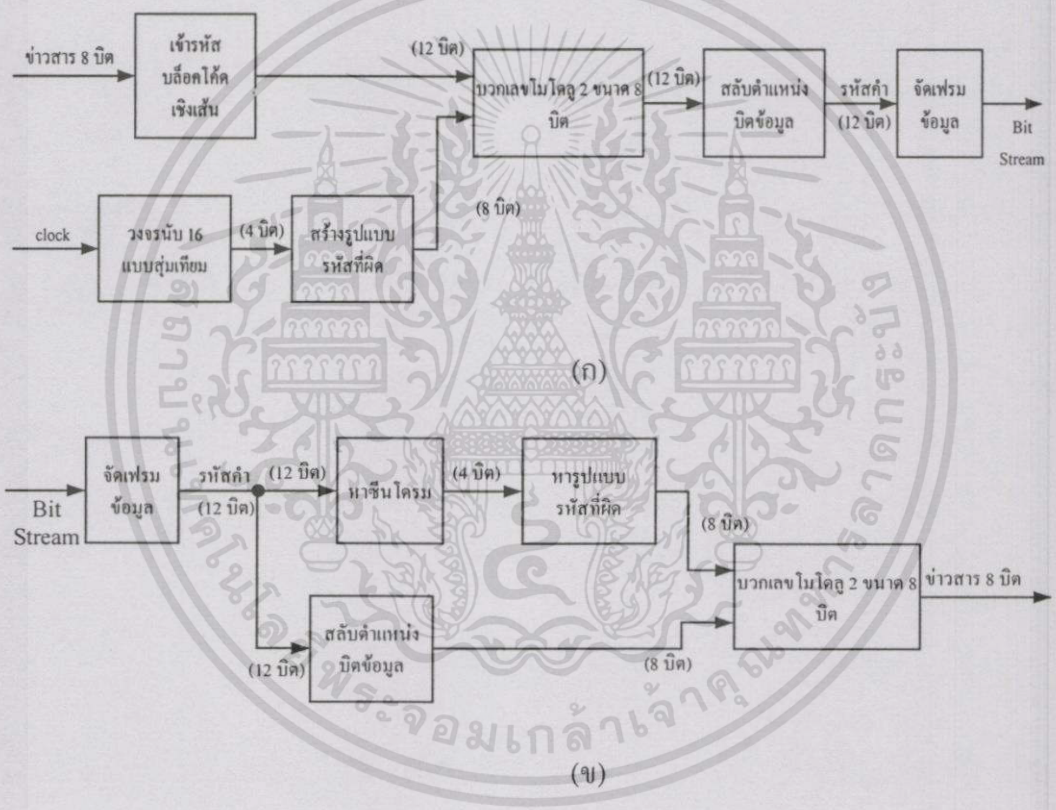


รูปที่ 2.14 การต่อใช้งานในลักษณะมาสเตอร์พาราแรล

บทที่ 3

การออกแบบและการสร้าง

บทนี้จะกล่าวถึงการออกแบบและการสร้างระบบปกปิดข้อมูลบน FPGA ซึ่งประกอบด้วยวงจรในภาคต่างๆ คือ วงจรเข้ารหัสแบบบล็อคโค้ดเชิงเส้น วงจรนับแบบสุ่มเทียม วงจรกำเนิดรูปแบบที่ผิด วงจรแก้รหัสที่ผิดแบบบล็อคโค้ดเชิงเส้นโดยอาศัยซินโดรม วงจรสลับตำแหน่งบิตข้อมูล ภาคส่งและภาครับ ดังผังวงจรในรูปที่ 3.1 (ก) และ (ข) โดยวงจรทั้งหมด จะถูกสร้างรวมอยู่ภายใน FPGA เพียงตัวเดียว



รูปที่ 3.1 ผังวงจรของระบบปกปิดข้อมูล (ก) ภาคส่ง (ข) ภาครับ

ในการทดสอบระบบ ได้สร้างวงจรแปลงสัญญาณจากแอนะลอกเป็นดิจิตอลและแปลงกลับ เพื่อทดสอบ เปรียบเทียบการทำงานและวัดประสิทธิภาพของระบบ ทั้งในกรณีที่ปกปิดข้อมูลและกรณีที่ถอดรหัสปกปิดข้อมูลแล้ว โดยทดสอบกับสัญญาณแอนะลอกย่านความถี่เสียงพูด (ความถี่ 300 ถึง 3400 เฮิรตซ์)

ในรูปที่3.1 (ก) ข่าวสารจะถูกเข้ารหัสแบบบล็อคโค้ดเชิงเส้น เพื่อนำไปรวมกับรหัสที่ผิดจากวงจรกำเนิดรูปแบบที่ผิดแบบสุ่มเทียม หลังจากนั้นจะถูกส่งผ่านวงจรสลับตำแหน่งบิต เพื่อเพิ่ม

ความซับซ้อนของรหัสคำ และส่งไปยังภาครับโดยผ่านวงจรจัดลำดับบล็อกข้อมูลให้มีขนาด 8 บิต รูปที่ 3.1 (ข) เป็นวงจรภาครับ การทำงานเริ่มต้นจากการจัดเฟรมข้อมูลกลับ สลับตำแหน่งบิตกลับ และหาซีนโดรมเพื่อแก้รหัสที่ผิดแบบสุ่มเทียม ตามลำดับ ซึ่งรายละเอียดในการออกแบบและสร้างวงจรในภาคต่างๆ จะได้กล่าวในลำดับต่อไปนี้

3.1 การออกแบบภาคส่ง

3.1.1 วงจรเข้ารหัสบล็อคโค้ดเชิงเส้น

วงจรในส่วนนี้ ทำหน้าที่แปลงข่าวสาร m ขนาด k บิต ด้วยการเข้ารหัสบล็อคโค้ดเชิงเส้น โดยการคูณกับเมตริกซ์ตัวกำเนิด **G** (Generator matrix) จะได้รหัสคำ u ขนาด n บิต จำนวนบิตของรหัสที่เพิ่มขึ้นมาขนาด $(n-k)$ บิตนี้เรียกว่าพาริตี้ ถูกนำไปใช้ตรวจสอบความถูกต้องและแก้รหัสที่ผิดในภาครับ เมตริกซ์ตัวกำเนิด **G** ที่ใช้ในงานวิจัยนี้คือ

$$\mathbf{G}=\left[\begin{array}{cccccccc:cccc} 1&0&0&0&0&0&0&0&0&0&1&1\\ 0&1&0&0&0&0&0&0&1&0&0&1\\ 0&0&1&0&0&0&0&0&1&1&0&0\\ 0&0&0&1&0&0&0&0&0&1&1&0\\ 0&0&0&0&1&0&0&0&1&1&1&0\\ 0&0&0&0&0&1&0&0&0&1&1&1\\ 0&0&0&0&0&0&1&0&1&0&1&1\\ 0&0&0&0&0&0&0&1&1&1&0&1 \end{array}\right] \tag{3.1}$$

เมื่อนำข่าวสารขนาด 8 บิตมาคูณกับแมทริกซ์ตัวกำเนิด **G** จะได้ผลลัพธ์เป็นรหัสคำ **u** ขนาดกว้าง 12 บิต คือ

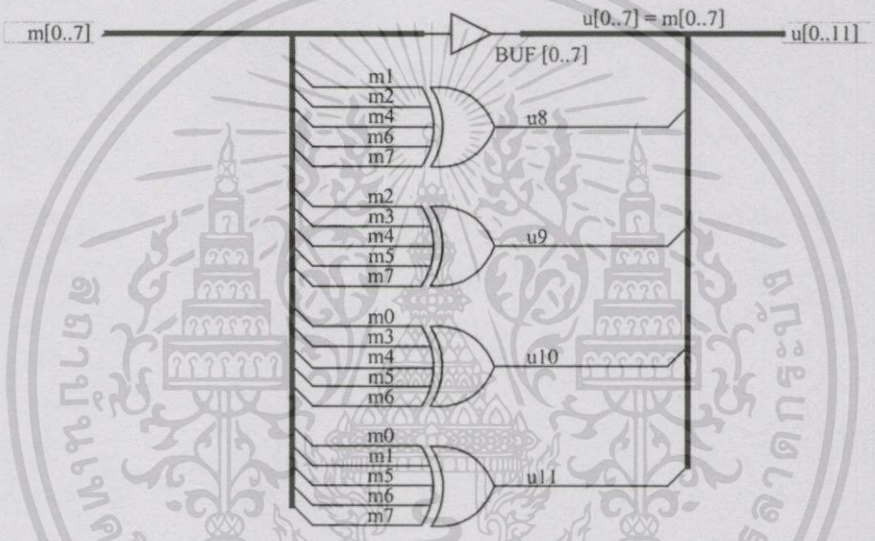
$$\mathbf{u}=\mathbf{mG} \tag{3.2}$$

$$u=\begin{bmatrix} m_0 & m_1 & m_2 & m_3 & m_4 & m_5 & m_6 & m_7 \end{bmatrix}\left[\begin{array}{cccccccc:cccc} 1&0&0&0&0&0&0&0&0&0&1&1\\ 0&1&0&0&0&0&0&0&1&0&0&1\\ 0&0&1&0&0&0&0&0&1&1&0&0\\ 0&0&0&1&0&0&0&0&0&1&1&0\\ 0&0&0&0&1&0&0&0&1&1&1&0\\ 0&0&0&0&0&1&0&0&0&1&1&1\\ 0&0&0&0&0&0&1&0&1&0&1&1\\ 0&0&0&0&0&0&0&1&1&1&0&1 \end{array}\right] \tag{3.3}$$

สามารถนำ (3.3) มาเขียนเป็นสมการลอจิกได้เป็น (3.4)

$$u_0 = m_0,\ u_1 = m_1, u_2 = m_2,\ u_3 = m_3, \tag{3.4}$$
$$u_4 = m_4,\ \ u_5 = m_5, u_6 = m_6,\ u_7 = m_7$$
$$u_8 = m_1 \oplus m_2 \oplus m_4 \oplus m_6 \oplus m_7,$$
$$u_9 = m_2 \oplus m_3 \oplus m_4 \oplus m_5 \oplus m_7$$
$$u_{10} = m_0 \oplus m_3 \oplus m_4 \oplus m_5 \oplus m_6,$$
$$u_{11} = m_0 \oplus m_1 \oplus m_5 \oplus m_6 \oplus m_7$$

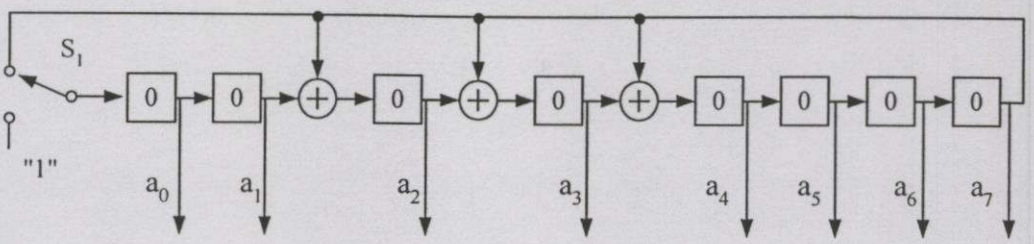
จากสมการ (3 4) สามารถออกแบบเป็นวงจรเกทซึ่งประกอบด้วยเกทเอ็กซ์คลูซีฟ-ออร์ ขนาด 5 อินพุต ดังรูปที่ 3.2



รูปที่ 3.2 วงจรเข้ารหัสบล็อคโค้ดเชิงเส้น

3.1.2 วงจรนับแบบสุ่มเทียม

ในระบบปกปิดข้อมูลนี้จะใช้รหัสขนาด 4 บิตในการสร้างรูปแบบที่ผิดแบบสุ่ม จากการออกแบบและคำนวณ พบว่าวงจรนับแบบสุ่มเทียมที่ใช้โพลิโนเมียลตั้งต้นอันดับ 8 ดังรูปที่ 3.3 จะให้รูปแบบการนับจำนวน 255 ลำดับที่ไม่ซ้ำกันคือ $P(x) = X^8 + X^4 + X^3 + X^2 + 1$ ซึ่งให้ลำดับการนับแบบสุ่มที่ยาวกว่าวงจรนับที่ใช้โพลิโนเมียลตั้งต้นอันดับ 4 และต้องสร้างวงจรสำหรับตั้งค่าสภาวะเริ่มต้นให้กับวงจรนับแบบสุ่มเทียมด้วย เนื่องจากในขณะรีเซ็ต ค่าในรีจิสเตอร์ทั้งหมดจะเป็นศูนย์ ทำให้วงจรไม่สามารถนับแบบสุ่มเทียมได้



รูปที่ 3.3 ผังวงจรนับแบบสุ่มเทียมโดยใช้โพลิโนเมียลตั้งต้นอันดับ 8

รูปที่ 3.3 เป็นผังวงจรนับแบบสุ่มเทียมขนาด 8 บิต แต่เลือกใช้เพียง 4 บิตเพื่อให้เข้ากับวงจรกำเนิดรูปแบบที่ผิดได้ จากการคำนวณพบว่าวงจรนี้ให้ลำดับการนับแบบสุ่มที่ยาวกว่าวงจรนับขนาด 4 บิต ซึ่งมีลำดับการนับที่วนรอบในจำนวน 16 ลำดับเท่านั้น วงจรสมบูรณ์แสดงไว้ในภาคผนวก ก ตัวเลข "0" ที่ปรากฏในชิฟท์รีจิสเตอร์คือค่าขณะรีเซ็ตของวงรนับแบบสุ่มเทียม

3.1.3 วงจรกำเนิดรูปแบบที่ผิด

รหัสคำ u ที่ได้จากการเข้ารหัสแบบซิสเต็มเมติกส์ จะประกอบด้วยข่าวสาร m และพาริตี้จำนวน 4 บิต การที่ผู้ประสงค์ร้ายจะดักจับข่าวสาร ก็สามารถทำได้ง่ายโดยการตัดพาริตี้ทิ้งไปเท่านั้น เพื่อเป็นการเพิ่มความซับซ้อนให้กับรหัสคำ วงจรในส่วนนี้จะทำหน้าที่กำเนิดรูปแบบรหัสที่ผิดจำนวน 2 บิท โดยอาศัยสัญญาณจากวงจรนับแบบลำดับสุ่ม r_p ขนาด 4 บิต จากวงจรในรูปที่ 3.3 เป็นตัวกำหนดรูปแบบที่ผิด แล้วรวมเข้ากับรหัสคำ u ซึ่งจะได้ให้รหัสคำที่ต่างไปจากข่าวสาร m

จากการหาความเป็นไปได้ของรหัสที่ผิดไปเป็นจำนวน 1 และ 2 บิต ในข้อมูลขนาด 8 บิต พบว่าสามารถสร้างรูปแบบที่ผิดได้ทั้งสิ้น 28 รูปแบบ พร้อมทั้งหาซินโดรมที่สอดคล้องกับรูปแบบที่ผิดดังกล่าวได้ ดังตารางที่ 3.1

ตารางที่ 3.1 รูปแบบรหัสที่ผิด ที่เลือกใช้ในการสร้างระบบปกปิดข้อมูล

ค่าซินโดรม	ตำแหน่งบิตที่ผิด 2 บิต	รูปแบบที่เลือก	รูปแบบของรหัส
0001	(3,5) และ (2,7)	(2,7)	001000010000
0010	(1,6) และ (2,4)	(1,6)	010000100000
0011	(4,7)	(4,7)	000010010000
0100	(0,5) และ (1,7)	(0,5)	100010000000
0101	(0,3), (1,2) และ (4,6)	(4,6)	000010100000
0110	(6,7)	(6,7)	000000110000

ตารางที่ 3.1 (ต่อ)

ค่าซีนโดรม	ตำแหน่งบิตที่ผิด 2 บิต	รูปแบบที่เลือก	รูปแบบของรหัส
1010	(0,1), (2,3) และ (5,7)	(5,7)	000001010000
1011	(2,5) และ (3,7)	(3,7)	000100010000
1100	(5,6)	(5,6)	000001100000
1101	(0,4) และ (3,6)	(0,4)	100010000000
1110	(0,7) และ (1,5)	(0,7)	100000010000
1111	(0,2) และ (1,3)	(1,3)	010100000000

รูปแบบที่เลือกในตารางที่ 3.1 คือรูปแบบที่ทำให้การรบกวนของสัญญาณแอนะล็อกสูงกว่ารูปแบบอื่น และจะได้ตารางความจริงของวงจรถอดรหัสสำหรับสร้างรูปแบบที่ผิดดังตารางที่ 3.2

ตารางที่ 3.2 ตารางความจริงของวงจรสร้างรูปแบบรหัสที่ผิด

ลำดับ	อินพุต				เอาท์พุต (ที่นำไปใช้กำหนดตำแหน่งบิตที่ผิด)							
	D	C	B	A	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
1	0	0	0	1	0	0	1	0	0	0	0	1
2	0	0	1	0	0	1	0	0	0	0	1	0
3	0	0	1	1	0	0	0	0	1	0	0	1
4	0	1	0	0	1	0	0	0	0	1	0	0
5	0	1	0	1	0	0	0	0	1	0	1	0
6	0	1	1	0	0	0	0	0	0	0	1	1
7	0	1	1	1	0	1	0	0	1	0	0	0
8	1	0	0	0	0	0	0	1	1	0	0	0
9	1	0	0	1	0	0	0	0	1	1	0	0
10	1	0	1	0	0	0	0	0	0	1	0	1
11	1	0	1	1	0	0	0	1	0	0	0	1
12	1	1	0	0	0	0	0	0	0	1	1	0
13	1	1	0	1	1	0	0	0	1	0	0	0
14	1	1	1	0	1	0	0	0	0	0	0	1
15	1	1	1	1	0	1	0	1	0	0	0	0

จากตารางที่ 3.2 สามารถเขียนเป็นสมการลอจิกได้ดัง (3.5)

$$Y_0 = \overline{A}B\overline{C}\overline{D} + AB\overline{C}D + ABC\overline{D} \quad (3.5)$$

$$Y_1 = \overline{A}\overline{B}C\overline{D} + BCD$$

$$Y_2 = \overline{A}\overline{B}\overline{C}D + A\overline{B}CD$$

$$Y_3 = A\overline{B}\overline{C}\overline{D} + ABCD$$

$$Y_4 = \overline{A}CD + A\overline{B}\overline{C} + B\overline{C}D$$

$$Y_5 = A\overline{B}C + A\overline{B}D + B\overline{C}\,\overline{D}$$

$$Y_6 = \overline{A}B\overline{C}D + \overline{A}C\overline{D} + AB\overline{C}\,\overline{D}$$

$$Y_7 = \overline{A}\,\overline{B}D + AC\overline{D} + B\overline{C}\,\overline{D}$$

3.1.4 วงจรสลับตำแหน่งบิตภาคส่ง

วงจรสลับตำแหน่งบิตในภาคส่ง จะใช้พารีตี้ที่ได้จากการเข้ารหัสในหัวข้อ 3.1.1 มาควบคุมการสลับตำแหน่งบิต พารีตี้ที่นำมาใช้สลับตำแหน่งบิตจะขึ้นอยู่กับข่าวสาร ข่าวสารจะถูกสลับตำแหน่งทั้ง 8 บิต ในที่นี้จะมีรูปแบบของการสลับตำแหน่งบิตที่ต่างกัน 16 รูปแบบดังตารางที่ 3.3 และสามารถเขียนเป็นสมการลอจิกได้ดัง (3.6)

ตารางที่ 3.3 ตารางความจริงของวงจรสลับตำแหน่งบิตภาคส่ง

อินพุต				ตำแหน่งการสลับบิต							
A	B	C	D	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
0	0	0	0	O_0	O_2	O_3	O_4	O_7	O_6	O_1	O_6
0	0	0	1	O_2	O_4	O_6	O_1	O_0	O_3	O_7	O_5
0	0	1	0	O_5	O_7	O_0	O_4	O_2	O_6	O_1	O_3
0	0	1	1	O_6	O_4	O_0	O_3	O_7	O_5	O_1	O_2
0	1	0	0	O_4	O_0	O_6	O_3	O_7	O_1	O_2	O_5
0	1	0	1	O_5	O_1	O_6	O_2	O_4	O_3	O_7	O_0
0	1	1	0	O_7	O_0	O_1	O_2	O_6	O_4	O_5	O_3
0	1	1	1	O_6	O_2	O_1	O_0	O_7	O_3	O_4	O_5
1	0	0	0	O_3	O_1	O_4	O_0	O_6	O_2	O_5	O_7
1	0	0	1	O_2	O_5	O_4	O_1	O_6	O_3	O_0	O_7
1	0	1	0	O_1	O_7	O_3	O_4	O_5	O_2	O_0	O_6
1	0	1	1	O_5	O_1	O_6	O_2	O_7	O_0	O_3	O_4
1	1	0	0	O_6	O_4	O_0	O_3	O_1	O_2	O_7	O_5
1	1	0	1	O_7	O_0	O_1	O_2	O_3	O_4	O_5	O_6
1	1	1	0	O_4	O_0	O_7	O_3	O_2	O_1	O_5	O_6
1	1	1	1	O_5	O_0	O_4	O_1	O_2	O_3	O_7	O_6

$$\begin{aligned} O_0 = {} & \left(A\overline{B}\,\overline{C}\,\overline{D}\right)I_0 + \left(A\overline{B}C\overline{D}\right)I_1 + \left(\overline{B}C\overline{D}\right)I_2 + \left(A\overline{B}C\overline{D}\right)I_3 + \left(\overline{A}B\overline{C}\,\overline{D} + ABC\overline{D}\right)I_4 + \\ & \left(\overline{A}\,\overline{B}C\overline{D} + \overline{A}B\overline{C}D + ACD\right)I_5 + \left(\overline{A}CD + AB\overline{C}\,\overline{D}\right)I_6 + \left(\overline{A}BC\overline{D} + AB\overline{C}D\right)I_7 \end{aligned} \tag{3.6}$$

$$\begin{aligned} O_1 = {} & \left(\overline{A}B\overline{D} + BC\overline{D} + ABD\right)I_0 + \left(\overline{A}B\overline{C}D + A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}CD\right)I_1 + \\ & \left(\overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + \overline{A}BCD\right)I_2 + \left(A\overline{B}\,\overline{C}\,\overline{D} + ABC\overline{D}\right)I_3 + \left(A\overline{B}\,\overline{C}D\right)I_5 + \left(\overline{B}C\overline{D}\right)I_7 \end{aligned}$$

$$\begin{aligned} O_2 = {} & \left(\overline{A}\,\overline{B}C + AB\overline{C}\,\overline{D}\right)I_0 + \left(\overline{A}BC + AB\overline{C}D\right)I_1 + \left(\overline{A}\,\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}C\overline{D}\right)I_2 + \\ & \left(A\overline{B}\,\overline{C}\,\overline{D} + ABCD\right)I_4 + \left(\overline{A}B\overline{C} + \overline{A}\,\overline{C}D + A\overline{B}CD\right)I_6 + \left(ABC\overline{D}\right)I_7 \end{aligned}$$

$$\begin{aligned} O_3 = {} & \left(\overline{A}BCD + A\overline{B}\,\overline{C}\,\overline{D}\right)I_0 + \left(\overline{B}\,\overline{C}D + ABCD\right)I_1 + \left(\overline{A}BC\overline{D} + A\overline{B}CD + B\overline{C}D\right)I_2 + \\ & \left(\overline{A}\,\overline{B}CD + B\overline{C}\,\overline{D} + AB\overline{D}\right)I_3 + \left(\overline{A}B\overline{D} + \overline{B}C\overline{D}\right)I_4 \end{aligned}$$

$$O_4 = (\overline{A}B\overline{C}D)I_0 + (AB\overline{C}D)I_1 + (\overline{A}B\overline{C}\overline{D} + ABC)I_2 + (AB\overline{C}D)I_3 + (\overline{A}B\overline{C}D)I_4 + (A\overline{B}C\overline{D})I_5 + (\overline{A}BC\overline{D} + A\overline{B}\overline{C})I_6 + (\overline{A}\overline{C}\overline{D} + \overline{A}CD + \overline{B}CD)I_7$$

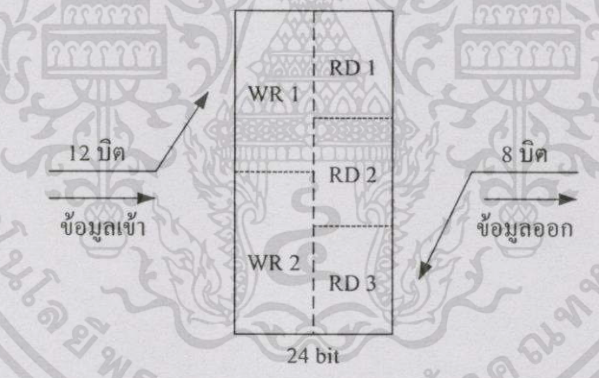
$$O_5 = (A\overline{B}CD)I_0 + (\overline{A}B\overline{C}D + ABC\overline{D})I_1 + (A\overline{B}\overline{D} + A\overline{C}\overline{D})I_2 + (\overline{A}BD + \overline{B}\overline{C}D + BCD)I_3 + (\overline{A}BC\overline{D} + AB\overline{C}D)I_4 + (\overline{A}\overline{B}CD)I_5 + (\overline{A}\overline{B}\overline{D})I_6$$

$$O_6 = (A\overline{B}\overline{C}D + A\overline{B}C\overline{D})I_0 + (\overline{A}B\overline{D} + \overline{A}\overline{B}C)I_1 + (\overline{A}BC\overline{D})I_2 + (A\overline{B}CD)I_3 + (\overline{A}BCD)I_4 + (A\overline{B}\overline{C}\overline{D} + AB\overline{C}D + BC\overline{D})I_5 + (\overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C}D + AB\overline{C}\overline{D} + ABCD)I_7$$

$$O_7 = (\overline{A}B\overline{C}D)I_0 + (\overline{A}\overline{B}CD)I_2 + (\overline{A}C\overline{D})I_3 + (A\overline{B}CD)I_4 + (\overline{A}\overline{B}\overline{C} + \overline{A}BCD + B\overline{C}\overline{D})I_5 + (AC\overline{D} + ABD)I_6 + (A\overline{B}\overline{C})I_7$$

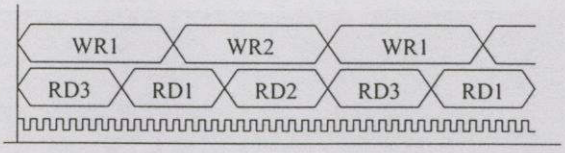
3.1.5 วงจรจัดรูปแบบข้อมูลสำหรับส่ง

เนื่องจากรหัสคำ u มีขนาด 12 บิต ทำให้ไม่สามารถส่งผ่านระบบการสื่อสารโดยทั่วไปขนาด 8 บิตได้ จึงออกแบบวงจรจัดรูปแบบข้อมูล ก่อนที่จะส่งเข้าสู่ช่องการสื่อสาร วงจรนี้อาศัยหลักการของสวิทซ์หน่วยความจำ ซึ่งใช้ที่พักข้อมูลขาเข้าขนาด 12 บิตจำนวน 2 บล็อค แล้วทยอยอ่านออกมาในขนาด 8 บิตจำนวน 3 บล็อค มีผังวงจรและผังเวลาดังรูปที่ 3.4



(ก) ผังวงจร

รูปที่ 3.4 วงจรจัดเฟรมข้อมูลภาคส่ง



(ข) ผังเวลา

รูปที่ 3.4 (ต่อ) วงจรจัดเฟรมข้อมูลภาคส่ง

เมื่อ WR 1 และ WR 2 คือช่วงเวลาในการเขียนข้อมูล 12 บิตลงในบัฟเฟอร์ภาคส่ง

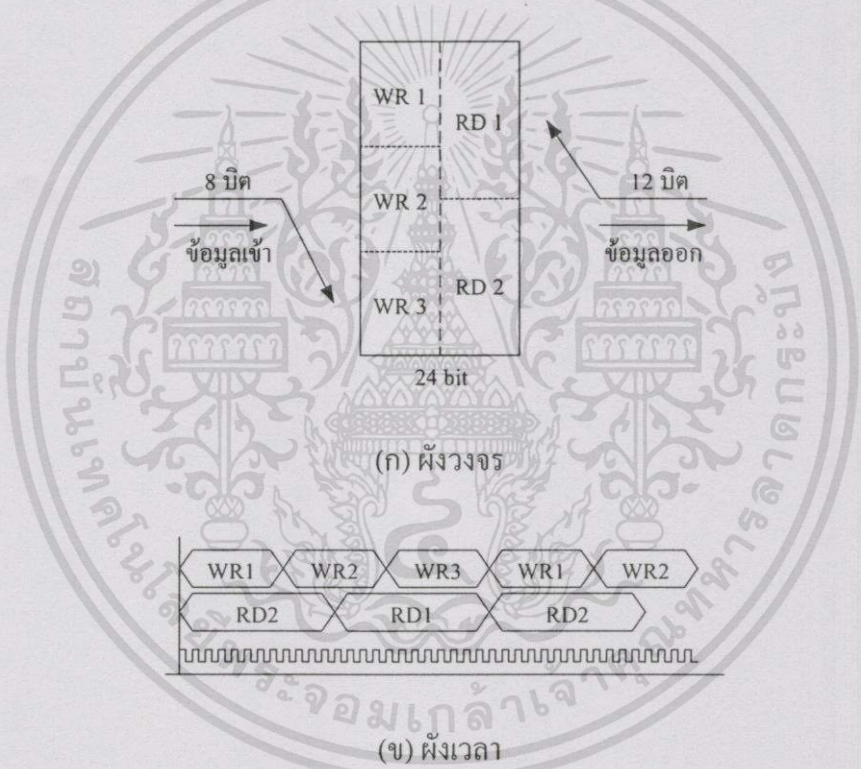
RD 1 RD 2 และ RD 3 คือช่วงเวลาในการอ่านข้อมูล 8 บิตจากบัฟเฟอร์ภาคส่ง

3.2 การออกแบบภาครับ

วงจรในภาครับ จะมีลักษณะสมมาตรกับภาคส่ง กล่าวคือ จะเริ่มจากการสลับตำแหน่งบิต แล้วผ่านการแก้ไขรหัสที่ผิด ประกอบด้วยการหาซีนโดรมและเปิดตารางรูปแบบรหัสที่ผิด ทำให้สามารถหารหัสเดิมออกมาได้อย่างถูกต้องครบถ้วน การออกแบบและการสร้างวงจรในภาครับสามารถกล่าวได้โดยลำดับดังนี้

3.1.5 วงจรจัดรูปแบบข้อมูลสำหรับภาครับ

วงจรจัดรูปแบบเฟรมข้อมูลในภาครับ มีลักษณะการทำงานตรงข้ามกับภาคส่ง คือจะจัดเรียงข้อมูลขนาด 8 บิต ที่รับเข้ามา ให้อยู่ในรูปแบบของข้อมูลขนาด 12 บิต เพื่อส่งให้ภาคอื่นๆ ต่อไป ดังผังวงจรและผังเวลาในรูปที่ 3.5



รูปที่ 3.5 ผังวงจรและผังเวลาของการจัดเฟรมข้อมูลภาครับ

เมื่อ WR 1 WR 2 และ WR 3 คือช่วงเวลาในการเขียนข้อมูล 8 บิตลงในบัฟเฟอร์ภาครับ
RD 1 และ RD 2 คือช่วงเวลาในการอ่านข้อมูล 12 บิตจากบัฟเฟอร์ภาครับ

จากรูปที่ 3.4 และ 3.5 ทั้งภาคส่งและภาครับ จะต้องทำงานสัมพันธ์กัน ในการออกแบบจะต้องมีการส่งสัญญาณเข้าจังหวะ (synchronous) จากภาคส่งไปยังภาครับโดยตรง แต่ถ้าส่งสัญญาณโดยไม่มีสายสัญญาณเข้าจังหวะดังกล่าว จะต้องออกแบบวงจรค้นหาสัญญาณนาฬิกาและสัญญาณเข้าจังหวะ (clock recovery) ที่ภาครับด้วย

3.2.1 วงจรสลับตำแหน่งบิตภาครับ

วงจรสลับตำแหน่งบิตในภาครับ ทำหน้าที่เช่นเดียวกับวงจรสลับตำแหน่งบิตในภาคส่ง จากตารางความจริงของการสลับบิตภาคส่งในตารางที่ 3.3 สามารถออกแบบตารางความจริงของการสลับตำแหน่งบิตภาครับ ได้ดังตารางที่ 3.4 และสมการลอจิกใน (3.7)

ตารางที่ 3.4 ตารางความจริงของวงจรสลับตำแหน่งบิตภาครับ

อินพุต				ตำแหน่งการสลับบิต							
A	B	C	D	I_0	I_1	I_2	I_3	I_4	I_5	I_6	I_7
0	0	0	0	O_0	O_6	O_1	O_2	O_3	O_7	O_5	O_4
0	0	0	1	O_4	O_3	O_0	O_5	O_1	O_7	O_2	O_1
0	0	1	0	O_2	O_4	O_6	O_7	O_3	O_0	O_5	O_6
0	0	1	1	O_2	O_6	O_7	O_3	O_1	O_5	O_0	O_4
0	1	0	0	O_1	O_5	O_6	O_3	O_0	O_7	O_2	O_4
0	1	0	1	O_7	O_1	O_3	O_5	O_4	O_0	O_2	O_6
0	1	1	0	O_1	O_2	O_3	O_7	O_5	O_6	O_4	O_0
0	1	1	1	O_3	O_2	O_1	O_5	O_6	O_7	O_0	O_4
1	0	0	0	O_3	O_1	O_5	O_0	O_2	O_6	O_4	O_7
1	0	0	1	O_6	O_3	O_5	O_5	O_2	O_1	O_4	O_7
1	0	1	0	O_6	O_0	O_0	O_2	O_3	O_4	O_7	O_1
1	0	1	1	O_5	O_1	O_3	O_6	O_7	O_0	O_2	O_4
1	1	0	0	O_2	O_4	O_5	O_3	O_1	O_7	O_0	O_6
1	1	0	1	O_1	O_2	O_3	O_4	O_5	O_6	O_7	O_0
1	1	1	0	O_1	O_5	O_4	O_3	O_0	O_6	O_7	O_2
1	1	1	1	O_1	O_3	O_4	O_5	O_2	O_0	O_7	O_6

$$O_0 = (\overline{A}\,\overline{B}C\overline{D})I_0 + (\overline{A}B\overline{D} + BC\overline{D} + ABD)I_1 + (\overline{A}\,\overline{B}C + AB\overline{C}\,\overline{D})I_2 + (\overline{A}BCD + A\overline{B}C\overline{D})I_3 + (\overline{A}\,\overline{B}\,\overline{C}D)I_4 + (A\overline{B}CD)I_5 + (A\overline{B}\,\overline{C}D + A\overline{B}C\overline{D})I_6 + (\overline{A}B\overline{C}D)I_7 \quad (3.7)$$

$$O_1 = (A\overline{B}C\overline{D})I_0 + (\overline{A}B\overline{C}D + A\overline{B}\,\overline{C}\,\overline{D} + A\overline{B}CD)I_1 + (\overline{A}BC + AB\overline{C}D)I_2 + (B'C'D + ABCD)I_3 + (AB\overline{C}\,\overline{D})I_4 + (\overline{A}B\overline{C}\,\overline{D} + ABC\overline{D})I_5 + (\overline{A}B\overline{D} + \overline{A}\,\overline{B}C)I_6$$

$$O_2 = (\overline{B}\,\overline{C}D)I_0 + (\overline{A}BCD)I_1 + (\overline{A}BC\overline{D} + A\overline{B}CD + B\overline{C}D)I_3 + (\overline{A}\,\overline{B}C\overline{D} + ABC)I_4 + (A\overline{B}\,\overline{D} + A\overline{C}\,\overline{D})I_5 + (\overline{A}B\overline{C}\,\overline{D})I_6 + (\overline{A}\,\overline{B}CD)I_7$$

$$O_3 = (A\overline{B}\,\overline{C}\,\overline{D})I_0 + (\overline{A}B\overline{C}\,\overline{D} + A\overline{B}CD)I_2 + (\overline{A}\,\overline{B}CD + B\overline{C}\,\overline{D} + AB\overline{D})I_3 + (AB\overline{C}D)I_4 + (\overline{A}B\overline{D} + \overline{B}\,\overline{C}D + BCD)I_4 + (A\overline{B}CD)I_6 + (\overline{A}C\overline{D})I_7$$

$$O_4 = (\overline{A}B\overline{C}\,\overline{D} + ABC\overline{D})I_0 + (\overline{A}\,\overline{B}D + AB\overline{C}\,\overline{D})I_1 + (A\overline{B}\,\overline{C} + ABCD)I_2 + (\overline{A}\,\overline{B}\,\overline{D} + \overline{B}C\overline{D})I_3 + (\overline{A}B\overline{C}D)I_4 + (\overline{A}BC\overline{D} + ABC'D)I_5 + (\overline{A}BCD)I_6 + (A\overline{B}CD)I_7$$

$$O_5 = (\overline{A}\,\overline{C}D + ACD)I_0 + (A\overline{B}\,\overline{C}D)I_1 + (A\overline{B}C\overline{D})I_4 + (\overline{A}\,\overline{B}CD)I_5 + (A\overline{B}\,\overline{C}\,\overline{D} + AB\overline{C}D + BC\overline{D})I_6 + (\overline{A}B\overline{C} + \overline{A}BCD + B\overline{C}\,\overline{D})I_7$$

$$O_6 = (\overline{A}CD + AB\overline{C}D)I_0 + (\overline{A}B\overline{C} + \overline{A}\overline{C}D + A\overline{B}CD)I_2 + (\overline{A}BC\overline{D} + A\overline{B}\overline{C})I_4 + (\overline{A}\overline{B}D)I_5 + (AC\overline{D} + ABD)I_7$$

$$O_7 = (\overline{A}BC\overline{D} + AB\overline{C}D)I_0 + (\overline{B}C\overline{D})I_1 + (ABC\overline{D})I_2 + (\overline{A}\overline{C}\overline{D} + \overline{A}CD + \overline{B}CD))I_4 + (\overline{A}\overline{C}D + AB\overline{C}\overline{D} + ABCD)I_6 + (A\overline{B}\overline{C})I_7$$

3.2.2 วงจรถอดรหัสบล็อคโค้ดเชิงเส้นโดยอาศัยซีนโดรม

วงจรถอดรหัสบล็อคโค้ดเชิงเส้น มีหน้าที่ตรวจสอบและแก้ไขความผิดพลาดของรหัสคำ u ที่รับเข้ามา โดยจะประกอบด้วย 3 วงจรย่อย คือ วงจรหาค่าซีนโดรม วงจรปิดตารางหารูปแบบที่ผิด และวงจรบวกแบบโมโดลูสอง

1. วงจรหาค่าซีนโดรม

วงจรนี้จะหาค่าซีนโดรม โดยการสร้างขึ้นเป็นวงจรลอจิกจากสมการ

$$S = r \cdot \mathbf{H}^{\mathrm{T}} \tag{3.8}$$

และ จากเมตริกซ์ $\mathbf{H}$ ที่สอดคล้องกับเมตริกซ์ $\mathbf{G}$ ในสมการที่ 3.1

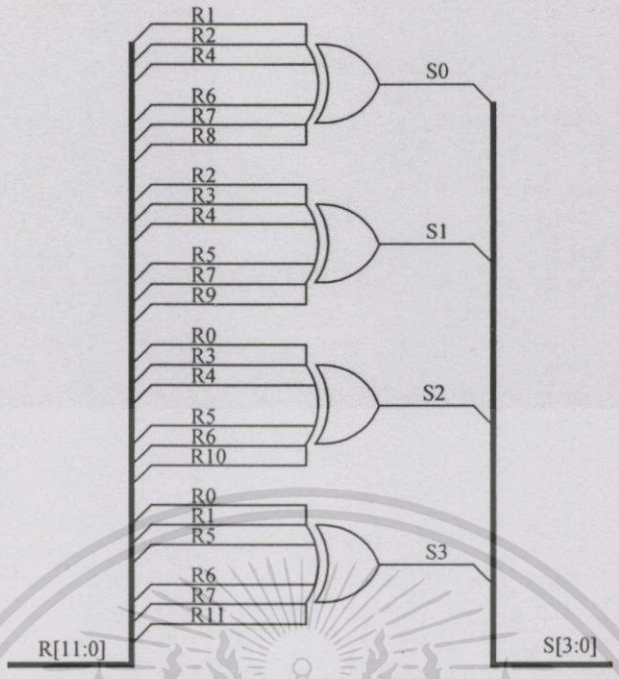
$$\mathbf{H} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.9}$$

จะได้

$$\begin{bmatrix} s_0 \\ s_1 \\ s_2 \\ s_3 \end{bmatrix} = [r_0, \ldots, r_{11}] \begin{bmatrix} 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}^{\mathrm{T}} \tag{3.10}$$

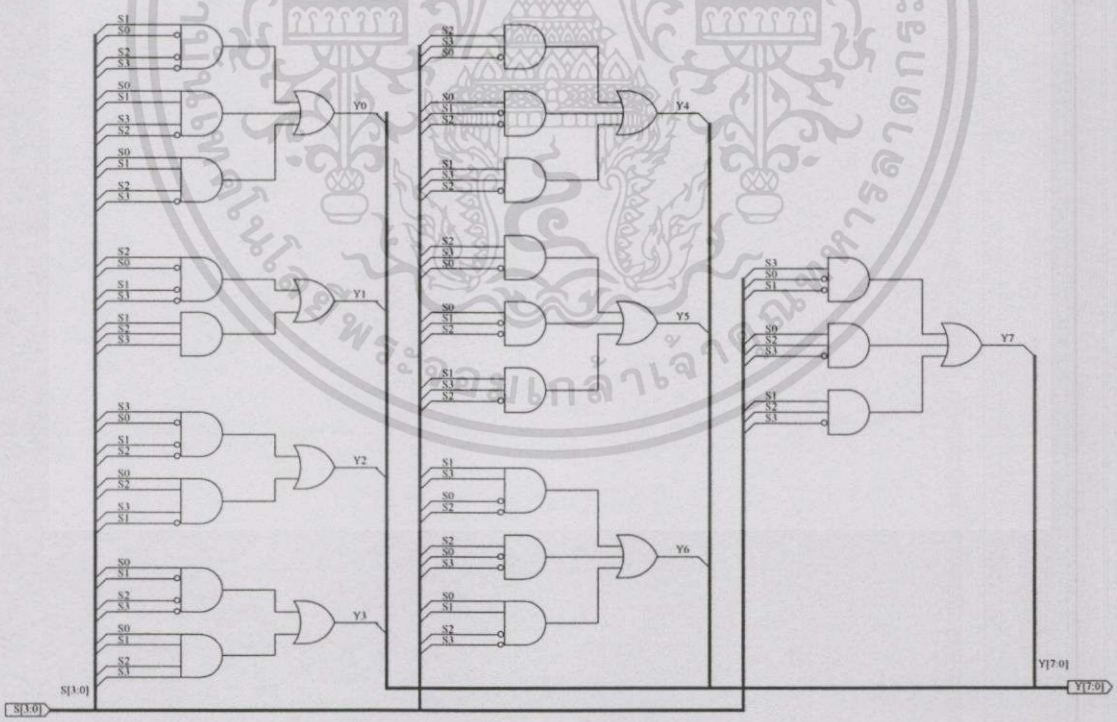
$$\begin{aligned} s_0 &= r_1 \oplus r_2 \oplus r_4 \oplus r_6 \oplus r_7 \oplus r_8 \\ s_1 &= r_2 \oplus r_3 \oplus r_4 \oplus r_5 \oplus r_7 \oplus r_9 \\ s_3 &= r_0 \oplus r_3 \oplus r_4 \oplus r_5 \oplus r_6 \oplus r_{10} \\ s_4 &= r_0 \oplus r_1 \oplus r_5 \oplus r_6 \oplus r_7 \oplus r_{11} \end{aligned} \tag{3.11}$$

จากสมการที่ (3.11) สามารถนำมาสร้างเป็นวงจรลอจิกได้ดังรูปที่ 3.6



รูปที่ 3.6 วงจรหาซีนโดรม

2. วงจรเปิดตารางเพื่อหารูปแบบที่ผิด



รูปที่ 3.7 วงจรเปิดตารางซีนโดรม

ค่าซีนโดรม s[0:3] ที่ได้จากวงจรในรูปที่ 3.6 จะถูกนำไปเปิดตารางหาตำแหน่งบิตที่ผิดด้วยวงจรในรูปที่ 3.7 ซึ่งจะทำให้ได้รหัสออกมาตามตารางที่ 3.2

3. วงจรแก้ไขบิตที่ผิด

การแก้ไขบิดที่ผิด ทำได้โดยการกลับค่าลอจิกของบิตนั้น จากรูปแบบที่ผิด ที่ได้จากการเปิดตารางซีนโดรม นำมาบวกแบบโมโดลูสอง กับรหัสคำ r ขนาด 12 บิต ในที่นี้จะใช้บิตที่ตรงกับตำแหน่งข่าวสารเท่านั้น ส่วนพาริตี้บิตจะถูกตัดทิ้งไป

3.3 วงจรสำหรับการทดลองภายนอก FPGA

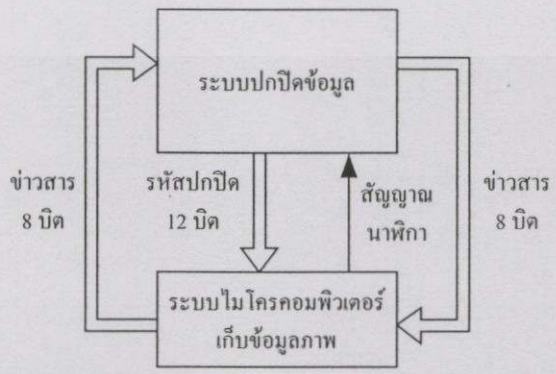
จากที่กล่าวมา เป็นวงจรที่อยู่ภายใน FPGA เพื่อเป็นการทดสอบความถูกต้องของการปกปิดข้อมูล จึงได้สร้างวงจรประกอบ โดยมีอินพุตและเอาท์พุต สำหรับข้อมูลดิจิตอลขนาด 8 บิต สำหรับทดสอบกับสัญญาณเสียงและข้อมูลภาพ ซึ่งผลการทดสอบจะกล่าวไว้ในบทที่ 4

การเชื่อมต่อ FPGA กับวงจรทดสอบภายนอก

การทดลองปกปิดข้อมูลเสียงพูด ทำได้โดยการเชื่อมต่อระบบแปลงสัญญาณแอนะลอกและดิจิตอลร่วมกับระบบปกปิดข้อมูล ดังรูปที่ 3.8

รูปที่ 3.8 ผังวงจรการเชื่อมต่อภายนอก เพื่อทดสอบการปกปิดสัญญาณเสียง

สัญญาณเสียงจะถูกสุ่มที่ความเร็วคงที่ (ประมาณ 8,000 ครั้งต่อวินาที) โดยวงจรหารความถี่ภายใน FPGA อาศัยคริสตอลความถี่ 4 MHz จากภายนอก ส่วนการทดสอบกับข้อมูลภาพ จะมีการทดลองรับและส่งสัญญาณภาพแบบบิตแมปผ่านระบบปกปิดข้อมูล โดยใช้เครื่องคอมพิวเตอร์ทยอยส่งข้อมูลครั้งละไบต์ ดังรูปที่ 3.9 ซึ่งผลที่ได้จะกล่าวถึงในบทที่ 4



รูปที่ 3.9 การประกอบระบบเพื่อทดลองปกปิดข้อมูลภาพ

3.4 สรุป

การออกแบบและการสร้างระบบปกปิดข้อมูล ได้สร้างวงจรภายใน FPGA เพื่อให้ได้วงจรรวมที่มีขนาดเล็ก และรักษาความลับของตัววงจรไว้ได้ นอกจากนั้น ในงานวิจัยนี้ยังได้สร้างวงจรประกอบและระบบย่อยต่างๆ เพื่อให้สะดวกในการทดลองและทดสอบ ซึ่งรายละเอียดวงจรทั้งหมดจะกล่าวถึงในภาคผนวก ก

บทที่ 4

การทดลองและผลการทดลอง

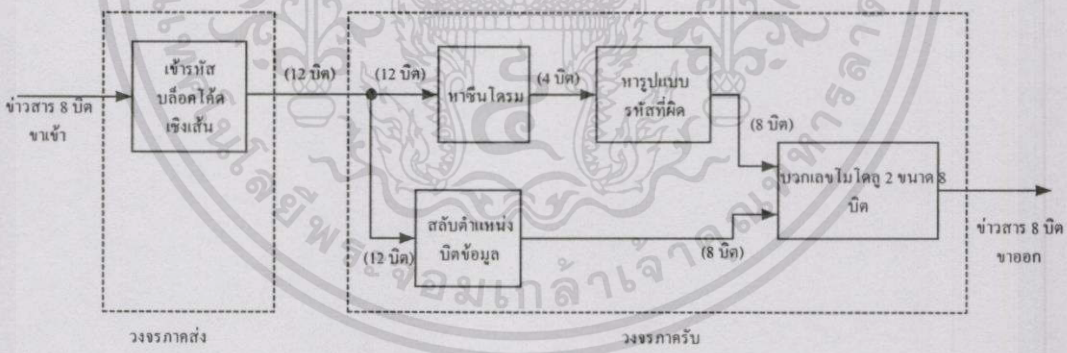
เนื้อหาในบทนี้ กล่าวถึงการทดลองและผลการทดลองของระบบปกปิดข้อมูล โดยการทดลองกับซอฟท์แวร์จำลองการทำงาน และทดสอบบนอุปกรณ์ FPGA เบอร์ XC4005E ที่เวลาจริง

4.1 กล่าวนำ

ในการทดลองระบบปิดข้อมูล นอกจากการทดสอบทุกส่วนประกอบย่อยแล้ว ได้มีการสร้างวงจรภายนอกเพิ่มเติมเป็นวงจรแปลงสัญญาณจากแอนะลอกเป็นดิจิตอลและแปลงกลับ ดังรูปที่ 3.8 เพื่อเปรียบเทียบการทำงาน ทั้งในกรณีที่ไม่มีการปกปิดข้อมูล กรณีที่มีการปกปิดข้อมูลและกรณีที่ถอดรหัสการปกปิดข้อมูลแล้ว โดยทดสอบกับสัญญาณแอนะลอกย่านความถี่เสียงพูด (ความถี่ 300 ถึง 3,400 เฮิรตซ์) อีกทั้งได้ทดลองปกปิดข้อมูลภาพสีเทา 256 ระดับ ซึ่งให้ผลในระดับที่น่าพอใจ

4.2 การทดสอบส่วนเข้ารหัสและถอดรหัสบล็อคโค้ดเชิงเส้น

การทดสอบส่วนเข้าและถอดรหัสบล็อคโค้ดเชิงเส้น ทำได้โดยการนำวงจรทั้งสองมาต่อด้วยกันโดยตรง ดังรูปที่ 4.1 ผลที่ได้พบว่า ค่าทางขาออกที่ได้ จะตรงกับทางขาเข้าทั้งหมด



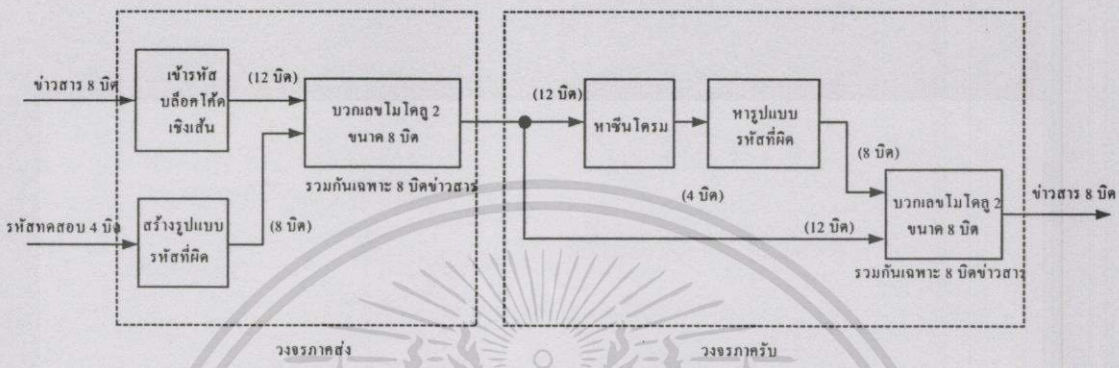
รูปที่ 4.1 การทดสอบการเข้ารหัสบล็อคโค้ดเชิงเส้น

4.3 การทดลองส่วนการสร้างรูปแบบที่ผิด

การทดสอบส่วนสร้างรหัสที่ผิด สามารถแบ่งขั้นตอนการทดสอบออกเป็น 2 ขั้น คือ การทดสอบน้ำหนักของรหัสที่ได้และการทดสอบร่วมกับวงจรแก้รหัสที่ผิด

การทดสอบน้ำหนักของรหัสทำได้โดยการป้อนอินพุตขนาด 4 บิตทั้ง 16 รูปแบบดังผังวงจรในรูปที่ 4.2 จากการทดลองพบว่ารหัสที่ได้จะมีน้ำหนักเป็น 2 เสมอ ส่วนการทดลองร่วมกับวงจรแก้รหัสที่ผิด แสดงผังวงจรไว้ในรูปที่ 4.3 จากการทดลอง พบว่าวงจรแก้รหัสที่ผิดสามารถแก้รหัสรหัสที่ผิดได้อย่างถูกต้องครบถ้วน

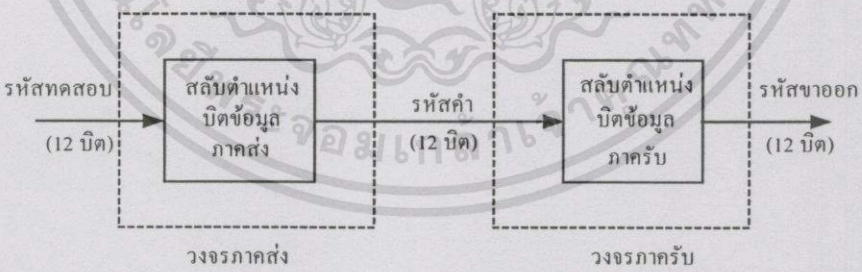
รูปที่ 4.2 ผังวงจรการทดสอบส่วนสร้างรูปแบบที่ผิด



รูปที่ 4.3 ผังวงจรการทดสอบร่วมกับส่วนแก้รหัสที่ผิด

4.4 การทดลองส่วนสลับตำแหน่งบิตข้อมูล

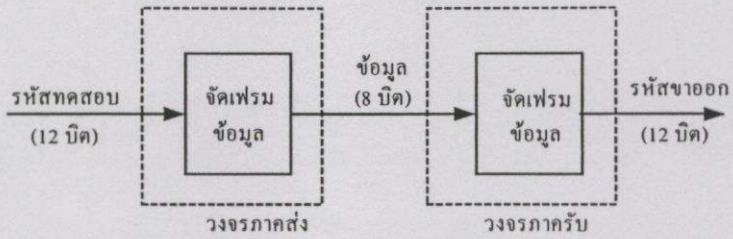
การทดสอบความถูกต้องในการทำงานของวงจรสลับตำแหน่งบิตข้อมูลทำได้โดยการประกอบวงจรตามผังในรูปที่ 4.4 จากการทดลอง พบว่าทั้งวงจรสลับตำแหน่งในภาครับและภาคส่งสามารถทำงานตามที่ออกแบบไว้ในตารางที่ 3.3 และ 3.4 และทำงานร่วมกันได้โดยมีความถูกต้องทุกประการ



รูปที่ 4.4 ผังวงจรการสลับตำแหน่งบิตข้อมูล

4.5 การทดลองส่วนจัดเฟรมข้อมูล

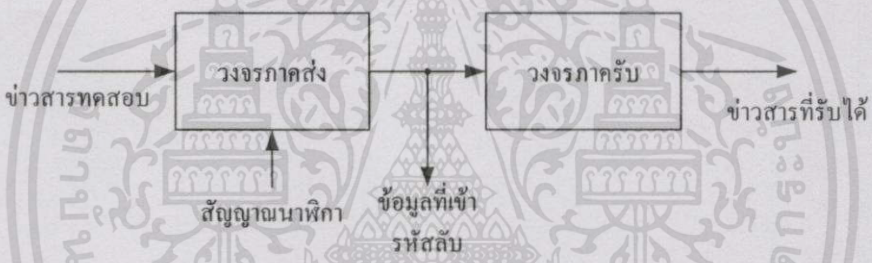
ในการทดสอบการจัดเฟรมข้อมูล ได้นำวงจรจัดเฟรมข้อมูลภาครับและภาคส่งมาต่อถึงกันโดยตรง แล้วทำการทดสอบโดยการป้อนสัญญาณดิจิตอลขนาด 12 บิต ตั้งแต่ 000000H ถึง 0FFFFFFH พร้อมทั้งป้อนสัญญาณนาฬิกาเพื่อกระตุ้นฟลิบฟลอบภายใน พบว่าข้อมูลเอาท์พุตที่ได้ตรงกับข้อมูลอินพุตทุกประการ



รูปที่ 4.5 การทดสอบการจัดเฟรมข้อมูลภาคส่งและภาครับ

4.6 การทดสอบส่วนประกอบโดยรวมทั้งระบบ

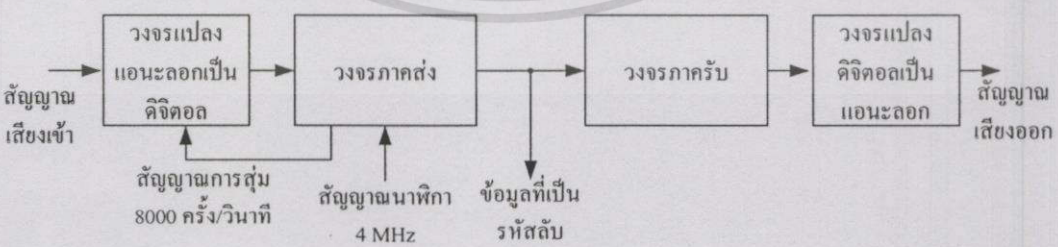
เมื่อทดสอบวงจรในภาคต่างๆ เรียบร้อยแล้ว ก็เป็นการทดสอบวงจรรวมทั้งระบบ โดยการประกอบวงจรตามรูปที่ 4.6 แล้วป้อนสัญญาณดิจิตอลอินพุตและสัญญาณนาฬิกา ด้วยเครื่องคอมพิวเตอร์ จากการทดสอบพบว่าข้อมูลที่เอาท์พุตตรงกับข้อมูลทางอินพุตทุกตัว แต่ข้อมูลที่ผ่านการเข้ารหัสลับไม่มีส่วนเหมือนกับข้อมูลอินพุตเลย



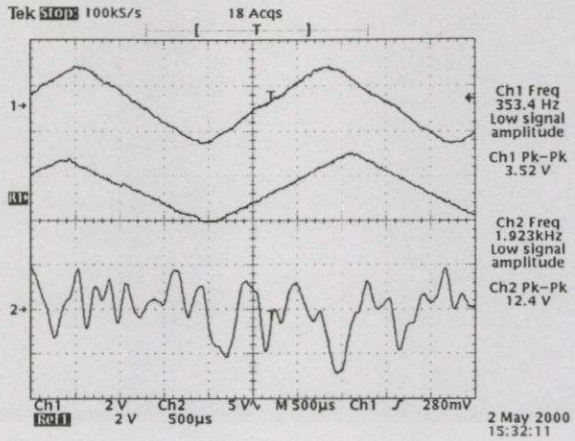
รูปที่ 4.6 การทดสอบการส่งข้อมูลผ่านวงจรทั้งระบบ

4.7 การทดลองเข้ารหัสลับสัญญาณเสียง

ระบบปกปิดข้อมูลนี้ สามารถนำไปใช้ได้กับสัญญาณเสียงพูดในย่านความถี่ 300 ถึง 3,400 เฮิตรซ์ จากการทดลองกับสัญญาณในรูปแบบต่างๆ แสดงผลที่ได้ดังรูปที่ 4.7 ถึง 4.10

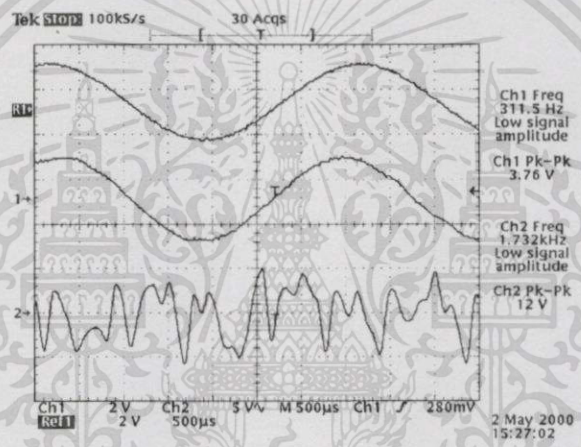


รูปที่ 4.7 วงจรทดสอบการเข้ารหัสลับสัญญาณเสียง



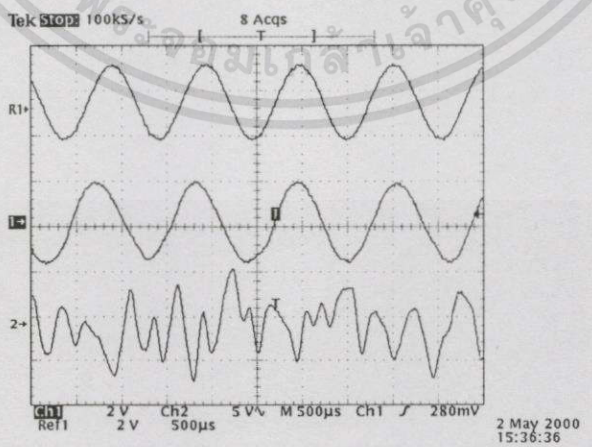
รูปที่ 4.8 การทดลองเข้ารหัสลับสัญญาณสามเหลี่ยม

บน:สัญญาณต้นฉบับ กลาง:สัญญาณเอาท์พุต ล่าง:สัญญาณที่ผ่านการปกปิด



รูปที่ 4.9 การทดลองเข้ารหัสลับสัญญาณไซน์ ความถี่ 311.5 Hz

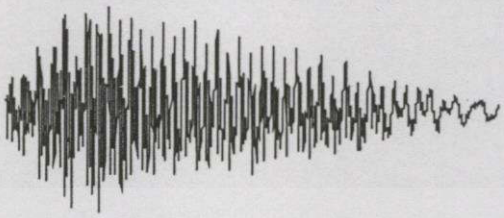
บน:สัญญาณต้นฉบับ กลาง:สัญญาณเอาท์พุต ล่าง:สัญญาณที่ผ่านการปกปิด



รูปที่ 4.10 การทดลองเข้ารหัสลับสัญญาณไซน์ ความถี่ 2 kHz

บน:สัญญาณต้นฉบับ กลาง:สัญญาณเอาท์พุต ล่าง:สัญญาณที่ผ่านการปกปิด

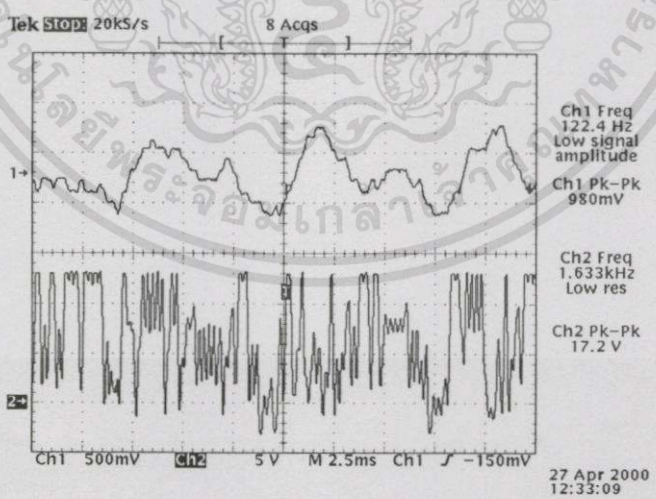
นอกจากการทดสอบด้วยสัญญาณจากเครื่องกำเนิดสัญญาณฟังก์ชันแล้ว จากการทดสอบด้วยเสียงพูด ระบบก็ยังคงให้การเข้ารหัสลับในระดับที่น่าพอใจ ดังผลการทดลองในรูปที่ 4.11 และ 4.12



(ก) สัญญาณเสียงพูดต้นฉบับ

(ข) สัญญาณเสียงพูดที่เข้ารหัสลับ

รูปที่ 4.11 สัญญาณเสียงพูด "เอ" และสัญญาณที่ผ่านระบบปกปิดข้อมูล

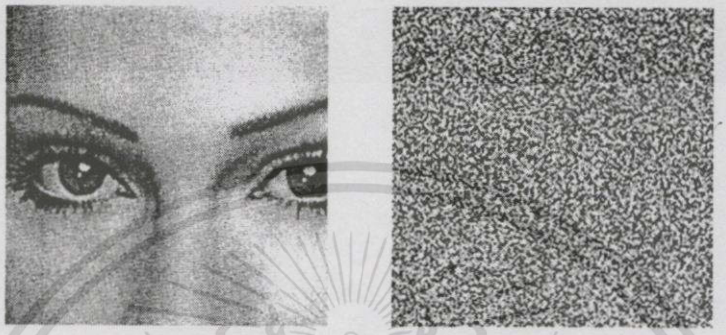


รูปที่ 4.12 การทดลองส่งเสียงเพลงผ่านระบบปกปิดข้อมูล

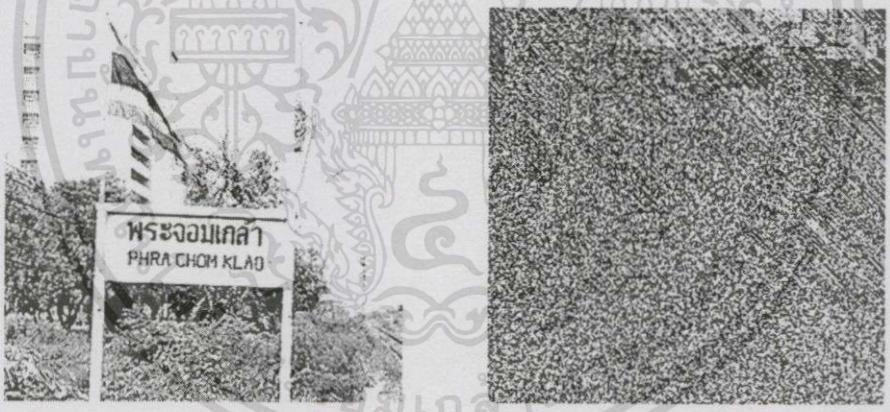
บน:สัญญาณต้นฉบับ ล่าง:สัญญาณที่ผ่านการปกปิด

4.8 การทดลองเข้ารหัสลับข้อมูลภาพ

การทดสอบระบบปกปิดข้อมูลภาพ โดยการต่อวงจรทดสอบดังรูปที่ 3.9 ซึ่งทำได้ในลักษณะการจำลองไม่ใช่เวลาจริง แต่ผลที่ได้นั้นคาดว่าสามารถนำไปใช้ได้กับการปกปิดข้อมูลภาพที่เวลาจริง ดังรูปที่ 4.13 เป็นการปกปิดข้อมูลภาพสีเทา 256 ระดับ โดยมีขนาด 256 x 256 จุด และการปกปิดข้อมูลภาพขาวดำ แสดงไว้ในรูปที่ 4.14 และ 4.15



รูปที่ 4.13 ผลการปกปิดข้อมูลภาพสีเทา 256 ระดับ ขนาด 256 x 256 จุด
(ก) ภาพต้นฉบับ (ข) ภาพที่ผ่านการเข้ารหัส



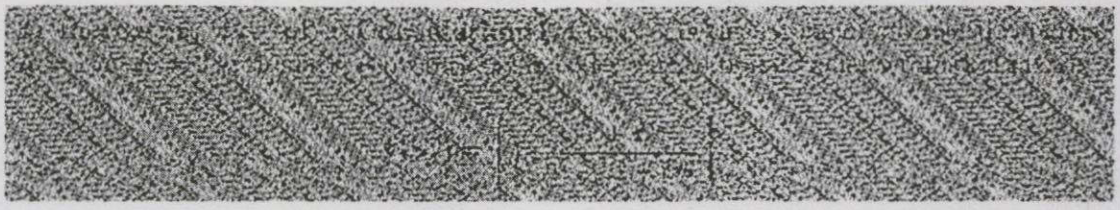
รูปที่ 4.14 การปกปิดข้อมูลภาพขาวดำ ขนาด 256 x 256 จุด

3) **Increasing d_{free} of a Convolutional Code.** In this problem we use the binary $R = 1/2$, $K = 3$ systematic convolutional encoder with rational transfer function matrix

$$\mathbf{G}(D) = \left[1 \quad \frac{1 + D^2}{1 + D + D^2} \right].$$

(ก) ภาพต้นฉบับ

รูปที่ 4.15 การปกปิดข้อมูลภาพขาวดำ ขนาด 600 x 100 จุด



(ข)ภาพที่ผ่านการปกปิด

รูปที่ 4.15 (ต่อ)

4.9 สรุป

เนื่องจากระบบที่ออกแบบบน FPGA เป็นวงจรดิจิตอลที่มีความเร็วสูง การนำมาใช้กับความถี่เสียงที่เวลาจริง ซึ่งมีอัตราการสุ่มประมาณ 8,000 ครั้งต่อวินาที จึงไม่มีขีดจำกัดอันเนื่องมาจากความเร็วของอุปกรณ์ประมวลผลสัญญาณ แต่ในการทดลองกับข้อมูลภาพนั้น มีข้อจำกัดในการรับส่งข้อมูลระหว่างเครื่องคอมพิวเตอร์ที่นำมาร่วมทดสอบกับระบบปกปิดข้อมูล จึงยังไม่สามารถทดลองที่เวลาจริงได้ อย่างไรก็ตามผลการทดลองดังกล่าว สามารถนำไปใช้อธิบายผลที่จะได้รับขณะส่งข้อมูลภาพที่เวลาจริง

บทที่ 5

สรุปผลการวิจัยและข้อเสนอแนะ

5.1 กล่าวนำ

งานวิจัยนี้ เป็นการพัฒนาต่อเนื่องจากงานวิจัย ซึ่งสร้างชิ้นงานขึ้นด้วยอุปกรณ์แยกส่วน มาเป็นระบบปกปิดข้อมูลที่สร้างบน FPGA ซึ่งมีขนาดเล็ก สามารถรักษาความลับของวงจรภายในได้ดี อีกทั้งยังสามารถแก้ไขและเปลี่ยนแปลงลักษณะการทำงานของวงจรได้ง่าย เนื่องจากส่วนประกอบที่กำหนดโครงสร้างลอจิกบล็อคภายใน FPGA เป็นหน่วยความจำแรม แก้ไขได้ไม่จำกัดจำนวนครั้ง

5.2 ปัญหาที่พบในการทำวิจัย

ปัญหาส่วนใหญ่ที่พบในงานวิจัยคือ

1. ด้านเครื่องมือในการออกแบบและสร้างวงจร เนื่องจากในขณะที่สร้างชิ้นงาน ระบบพัฒนาวงจรบน FPGA ยังคงมีราคาสูง

2. เครื่องมือสำหรับทดสอบความถูกต้องของวงจรที่สร้างขึ้น เนื่องจากวงจรที่ถูกสร้างขึ้นจะอยู่ภายใน FPGA จึงไม่สามารถวัดหรือทดสอบได้ตามกระบวนการปกติ

5.3 วิธีแก้ไขปัญหา

1. ด้านเครื่องมือในการวิจัย ได้รับการอนุเคราะห์จากภาควิชาต้นสังกัดของผู้ทำวิจัยและศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

2. การทดสอบความถูกต้องของวงจร อาศัยการโปรแกรมวงจรย่อยลงใน FPGA และป้อนรหัสทดสอบจนครบทุกตัว บันทึกผลและวิเคราะห์ข้อผิดพลาดที่เกิดขึ้นจนแก้ไขได้ทั้งหมด

5.4 ข้อเสนอแนะในการพัฒนา

งานวิจัยนี้ เป็นการปกปิดข้อมูลที่เพิ่มภาระให้กับระบบสื่อสาร คือเพิ่มปริมาณข้อมูลขึ้นถึง 1.5 เท่า ในการพัฒนางานวิจัยควรเพิ่มวงจรบีบอัดข้อมูลเพื่อให้สอดคล้องกับอัตราเร็วในการสื่อสารปัจจุบัน

สามารถนำอัลกอริทึมในงานวิจัยนี้ ไปสร้างบนชิพที่โปรแกรมได้ประเภทไมโครโปรเซสเซอร์ ไมโครคอนโทรลเลอร์ หรือ DSP เพื่อให้สามารถรักษาความลับในการสื่อสารในระบบเครือข่ายคอมพิวเตอร์โดยทั่วไป

5.5 สรุป

การทำงานวิจัยนี้มีข้อจำกัดหลายประการทั้งในด้านอุปกรณ์และเครื่องมือในการวิเคราะห์วงจรดิจิตอลขนาดใหญ่ ที่รองรับการทำงานกับข้อมูลปริมาณมาก จึงทำให้ต้องหาวิธีการทดสอบแบบต่างๆ ดังที่ได้กล่าวในบทที่ 3 และ 4 เพื่อให้ได้ความถูกต้องและเที่ยงตรงของวงจรมากที่สุด อย่างไรก็ตาม การเข้ารหัสบล็อคโค้ดเชิงเส้นที่เลือกใช้ในงานวิจัยนี้ ทำให้มีปริมาณข่าวสารเพิ่มขึ้นจากเดิมถึง 1.5 เท่าตัว (เนื่องจากข่าวสารขนาด 8 บิต ถูกเข้ารหัสเป็นรหัสคำขนาด 12 บิต) ซึ่งถือว่าเป็นการลดประสิทธิภาพของการสื่อสาร ดังนั้นในอนาคตอาจต้องมีการเพิ่มเติมวงจรบีบอัดข้อมูลเพื่อลดปริมาณข่าวสารลงให้ใกล้เคียงกับระบบที่ไม่ผ่านการเข้ารหัสลับ

เอกสารอ้างอิง

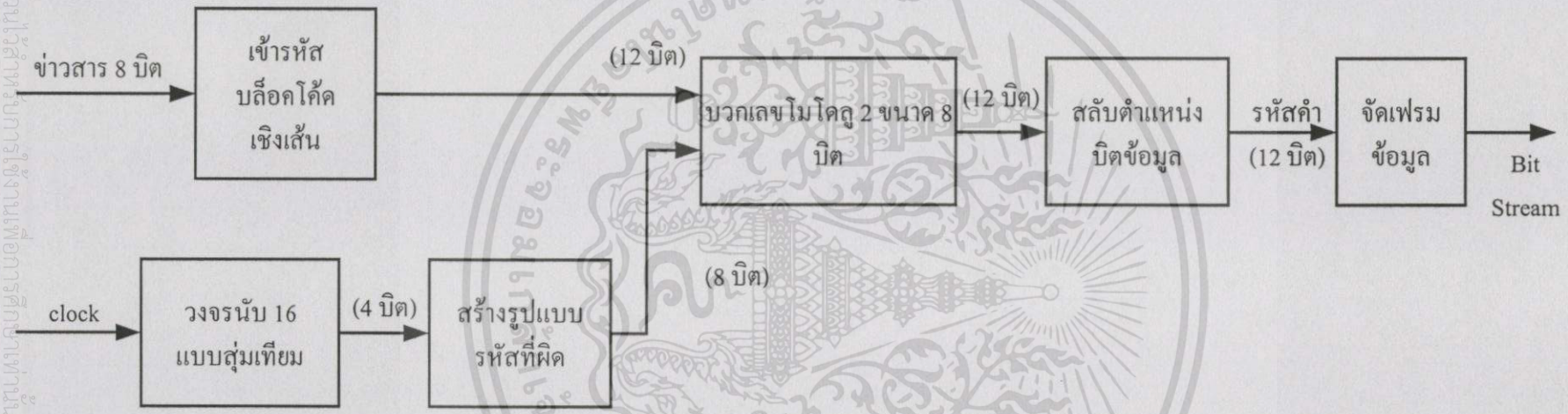
1. Henry J.Beker and Fred C.Piper. “Secure Speech Communications.” Academic Press,1985.
2. Stephen B. Wicker. “Error control system for digital communication and storage.” Prentic Hall, 1995.
3. Shu Lin. "An Introduction to Error Correcting Code." Prentice-Hall, Inc. Englewood Cliffs, New Jersey, 1970.
4. Vera Pless. "Introduction to the theory of error-correcting codes." John Wiley&Sons, Inc. 1982
5. Peter Sweeney. “Error control coding and introduction.” Prentic Hall, 1991.
6. George C. Clark, Jr. and J.Bibb Cain. “Error-Correction Coding for Digital Communications.” Plenum Press, 1981.
7. ฟูศักดิ์ ชีวสุวิทย์. "การแก้รหัสที่ผิด." คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง พ.ศ. 2528
8. ฟูศักดิ์ ชีวสุวิทย์ และคเชนทร์ แจ่มกมล. "การเข้ารหัสลับโดยใช้ซีนโดรมของลิเนียร์บล็อคโค้ดและการสลับบิตโดยอาศัยเพอซีโดแรนดอม" วารสารทางวิชาการของสมาคมคอมพิวเตอร์แห่งประเทศไทย
9. ฟูศักดิ์ ชีวสุวิทย์ และคเชนทร์ แจ่มกมล."การออกแบบตัวเข้ารหัสแบบเวลาจริงโดยใช้ซีนโดรมของลิเนียร์บล็อคโค้ดและการสลับบิตโดยการกำเนิดแบบสุ่ม." การประชุมวิชาการทางไฟฟ้า ครั้งที่ 18 มหาวิทยาลัยเทคโนโลยีมหานคร, หน้า 1034-1038, พฤศจิกายน 2538
10. ฟูศักดิ์ ชีวสุวิทย์ และ โกศล ตราชู, “การปกปิดข้อมูลด้วยการเข้ารหัสบล็อคโค้ดและสัญญาณรบกวนแบบลำดับสุ่มเทียม ที่สร้างบน FPGA”, วิศวสารลาดกระบัง. 17(3) : 34-39
11. Xilinx. "The Programable Logic Data Book." 3rd ed. Xilinx, Inc. 1994
12. Viewlogic System. "Workview Plus on Windows." Viewlogic Systems, Inc. 1993
13. Xilinx. "XACT Hardware&Pheripherals Giude." Xilinx, Inc. 1994

ภาคผนวก ก
ผลงานที่ได้รับการตีพิมพ์

1. ฟูศักดิ์ ชีวสุวิทย์ และ โกศล ตราชู, "การปกปิดข้อมูลด้วยการเข้ารหัสบล็อคโค้ดและสัญญาณรบกวนแบบลำดับสุ่มเทียม ที่สร้างบน FPGA", **วิศวสารลาดกระบัง.** 17(3) : 34-39

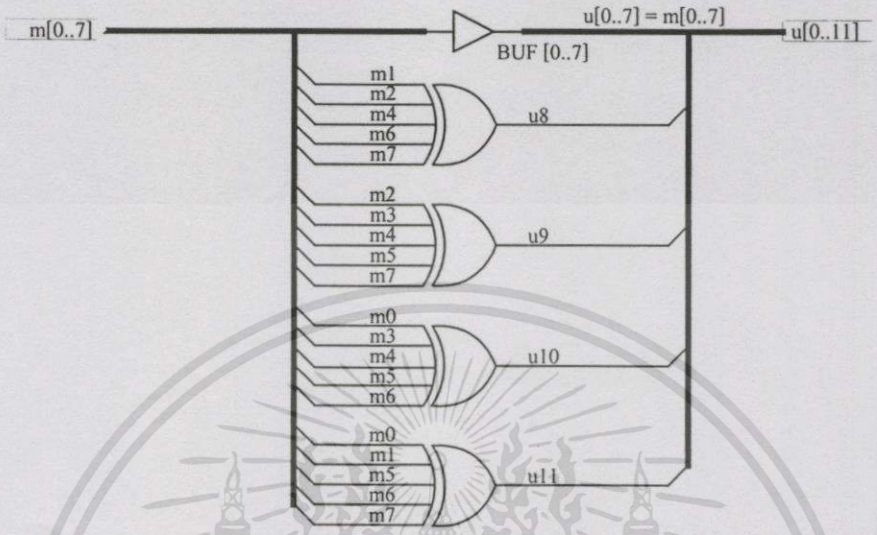
ภาคผนวก ข

วงจรในภาคส่ง

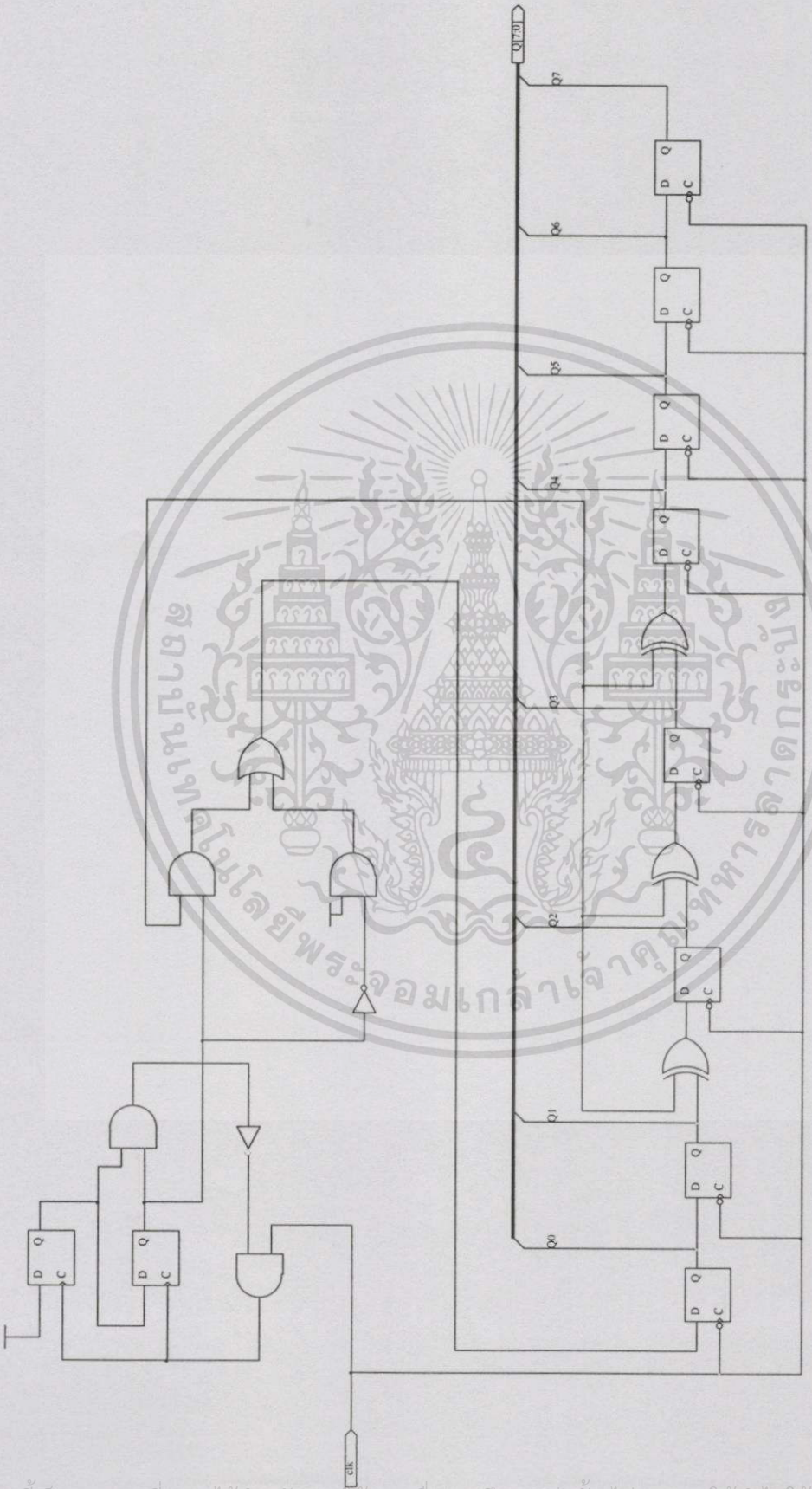


รูปที่ ข.1. ผังวงจรของภาคส่ง

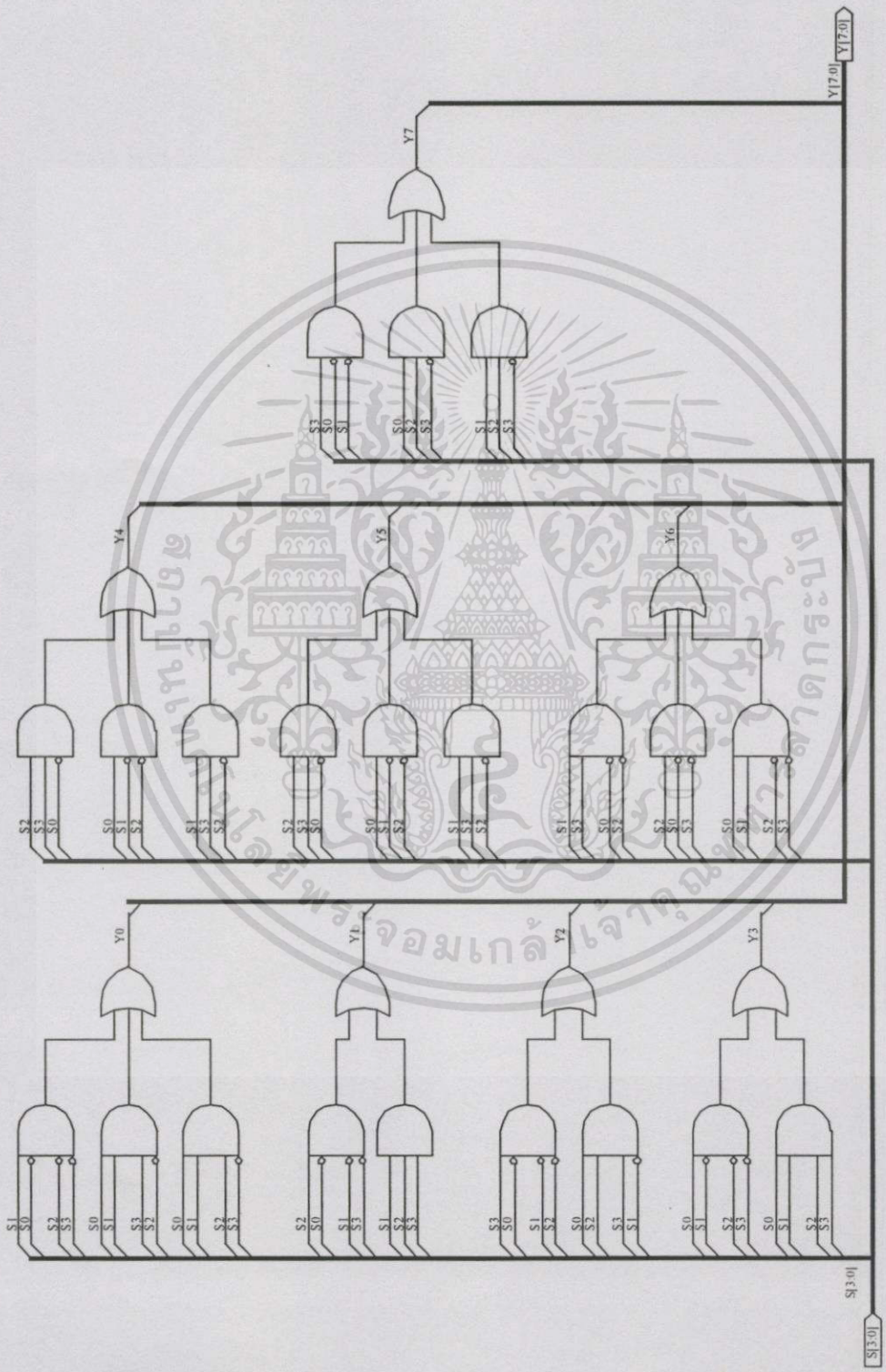
วงจรเข้ารหัสบล็อคโค้ดเชิงเส้น



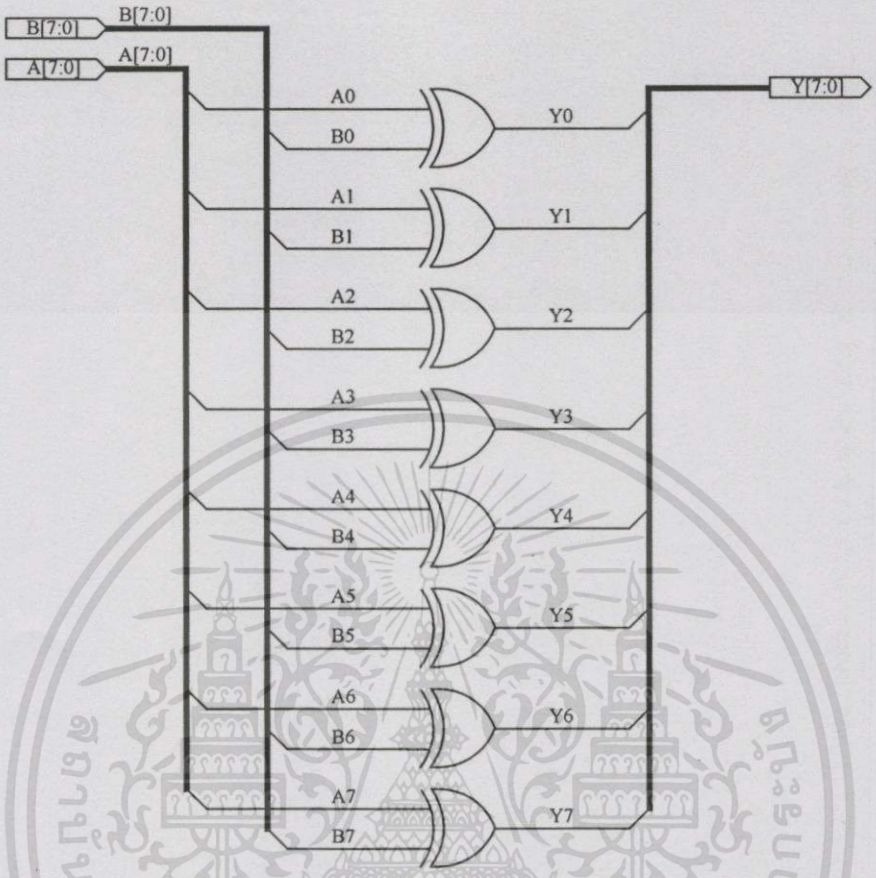
รูปที่ ข.2 วงจรเข้ารหัสบล็อคโค้ดเชิงเส้น



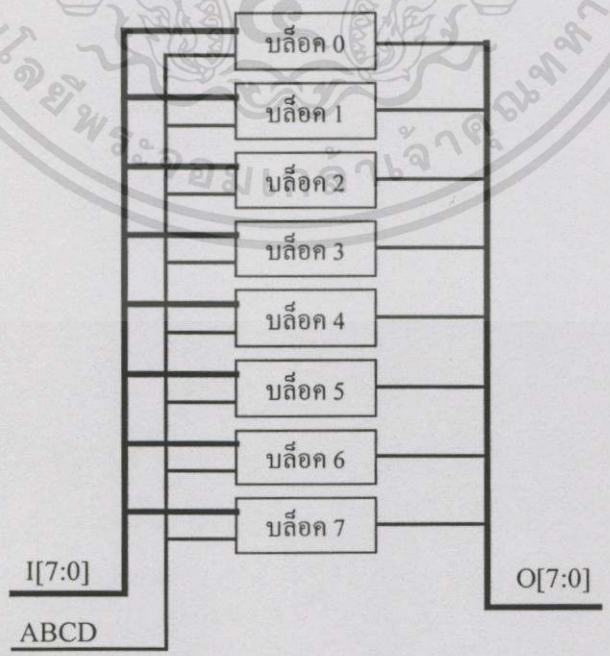
รูปที่ ข.3 วงจรกำเนิดลำดับการนับแบบสุ่มเทียมขนาด 8 บิต



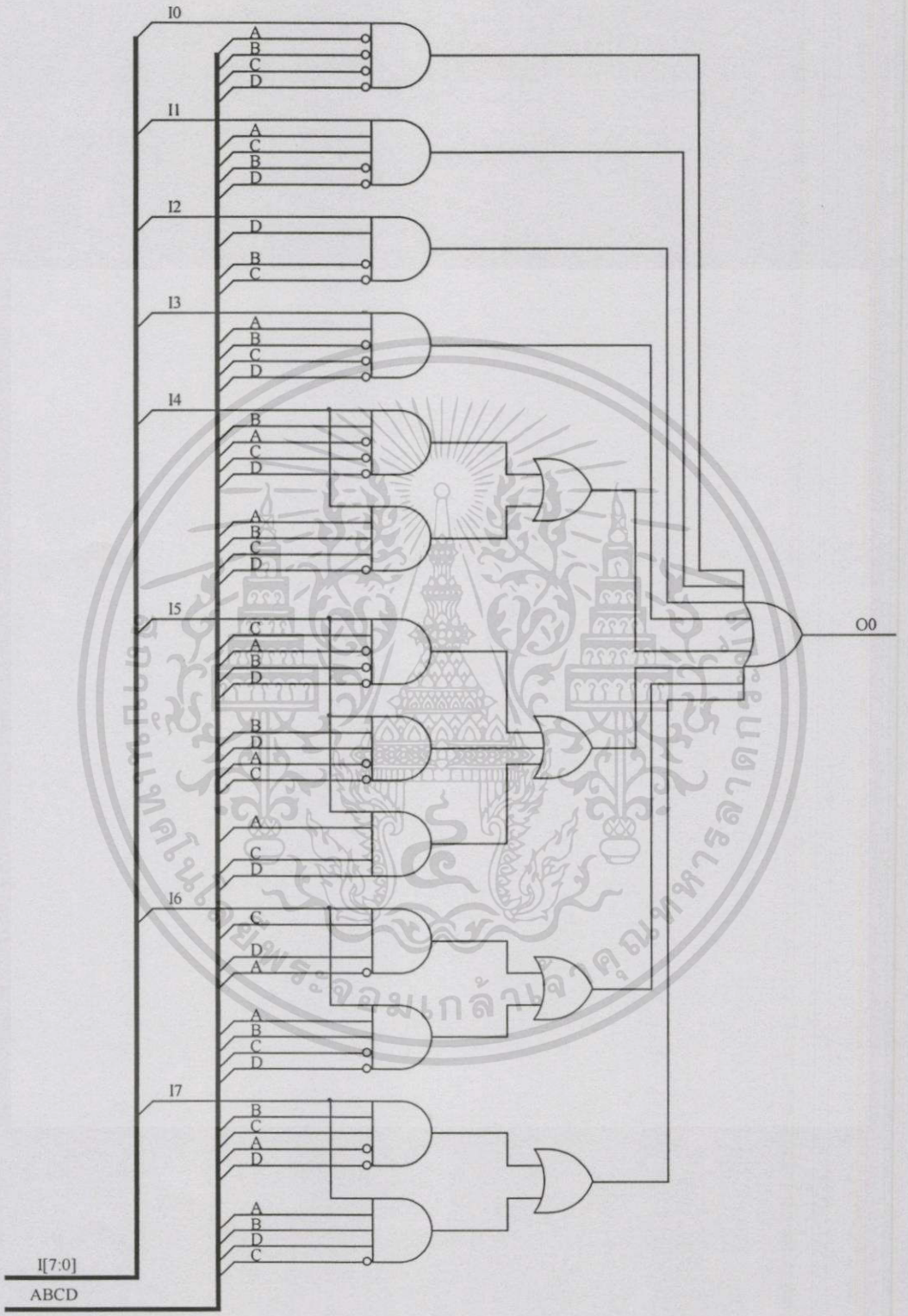
รูปที่ ข.4 วงจรสร้างรูปแบบที่ผิด



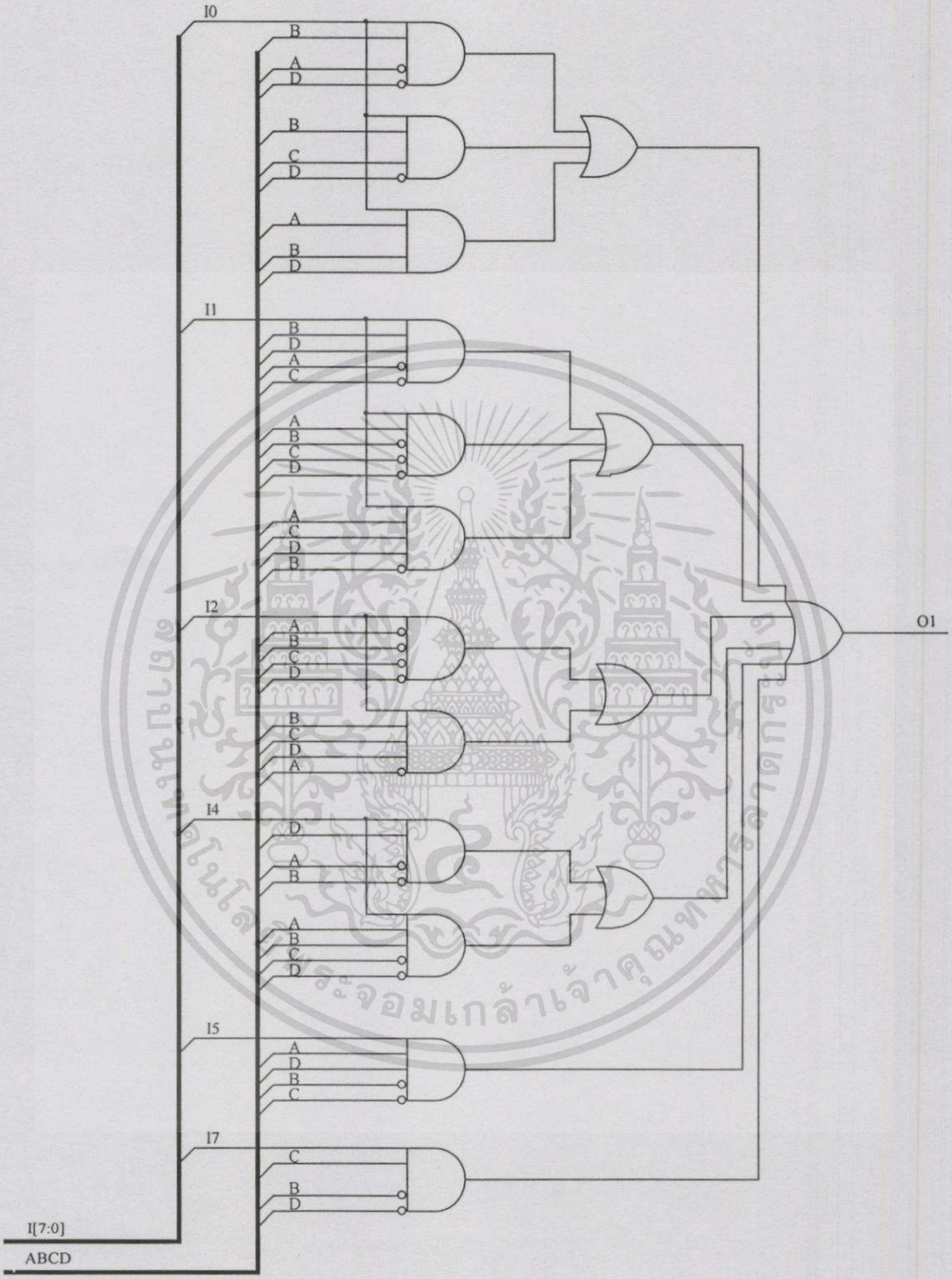
รูปที่ ข.5 วงจรบวกเลขโมดูล 2 ขนาด 8 บิต



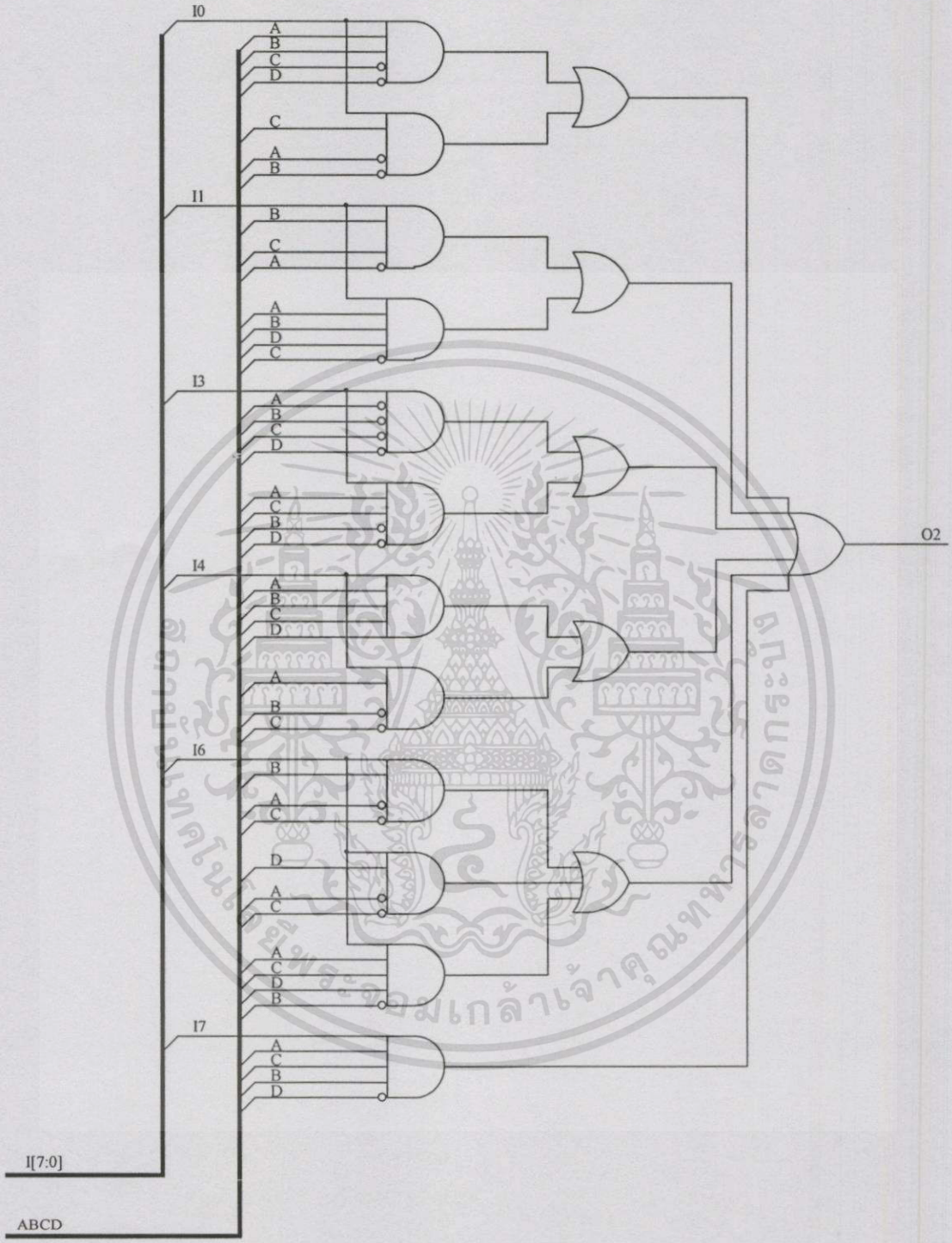
รูปที่ ข.6 ผังวงจรสลับตำแหน่งบิตภาคส่งและรับ



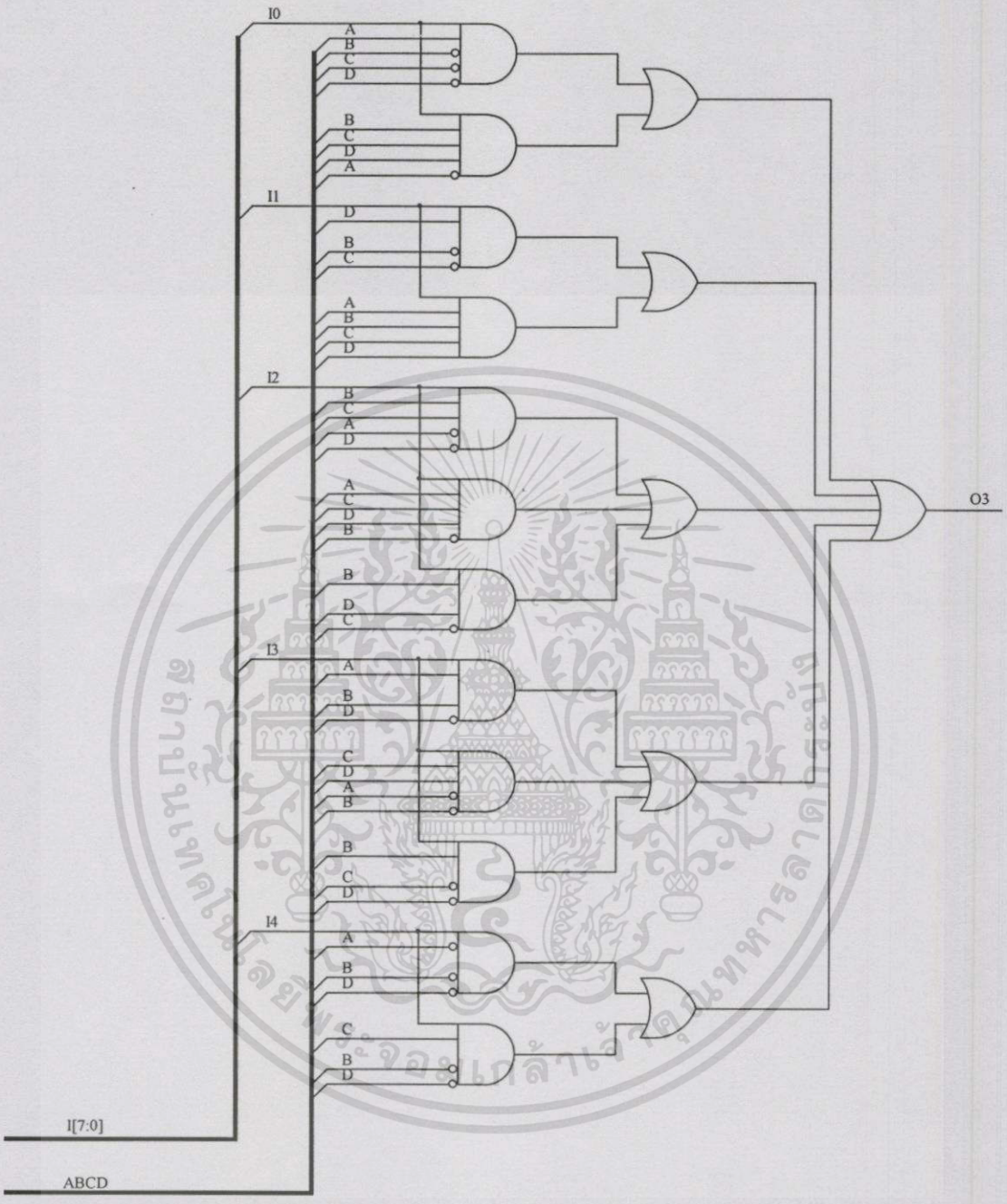
รูปที่ ข.7 วงจรสลับตำแหน่งบิตภาคส่ง บล็อค 0



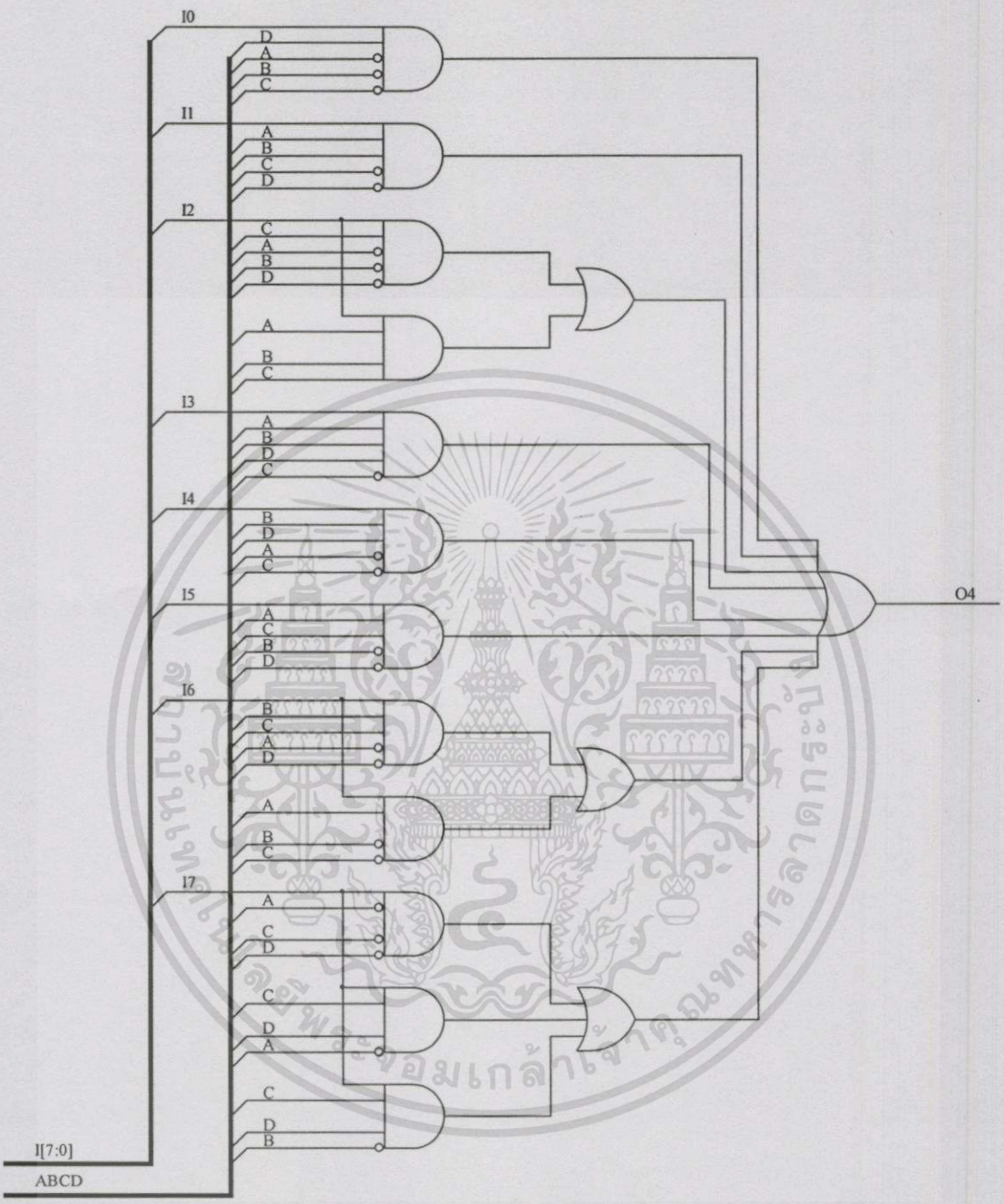
รูปที่ ข.8 วงจรสลับตำแหน่งบิตภาคส่ง บล็อค 1



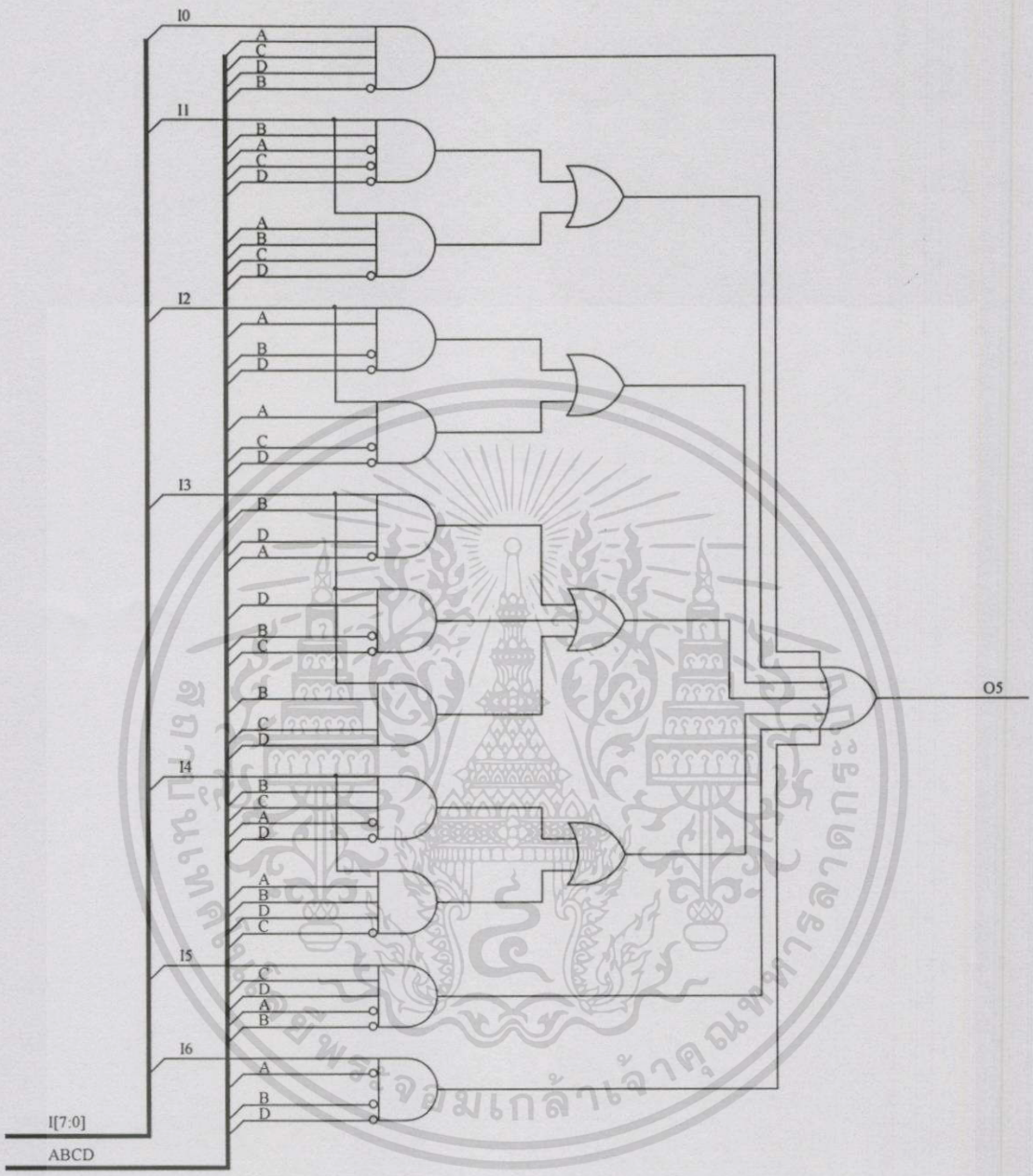
รูปที่ ข.9 วงจรสลับตำแหน่งบิตภาคส่ง บล็อค 2



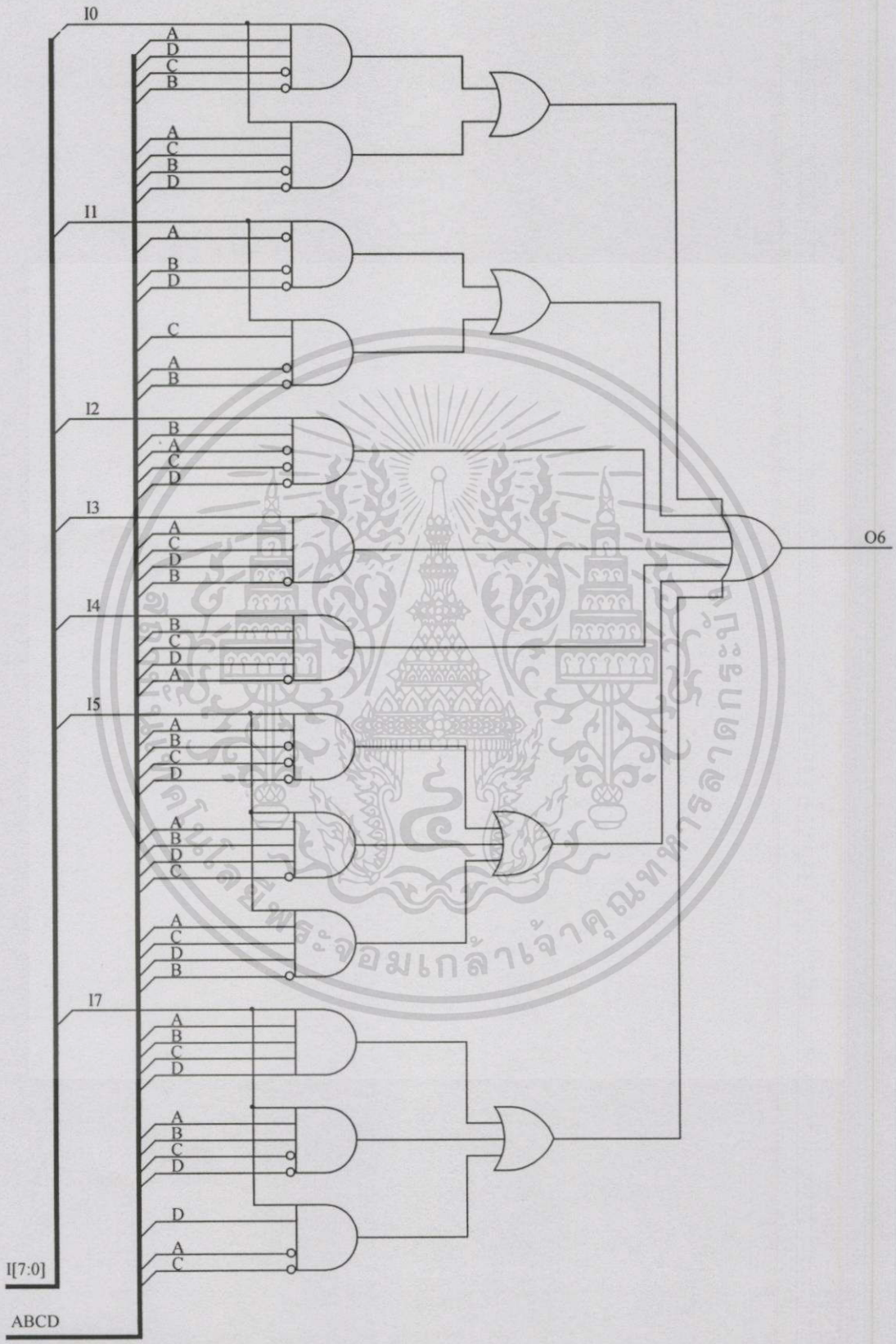
รูปที่ ข.10 วงจรสลับตำแหน่งบิตภาคส่ง บล็อค 3



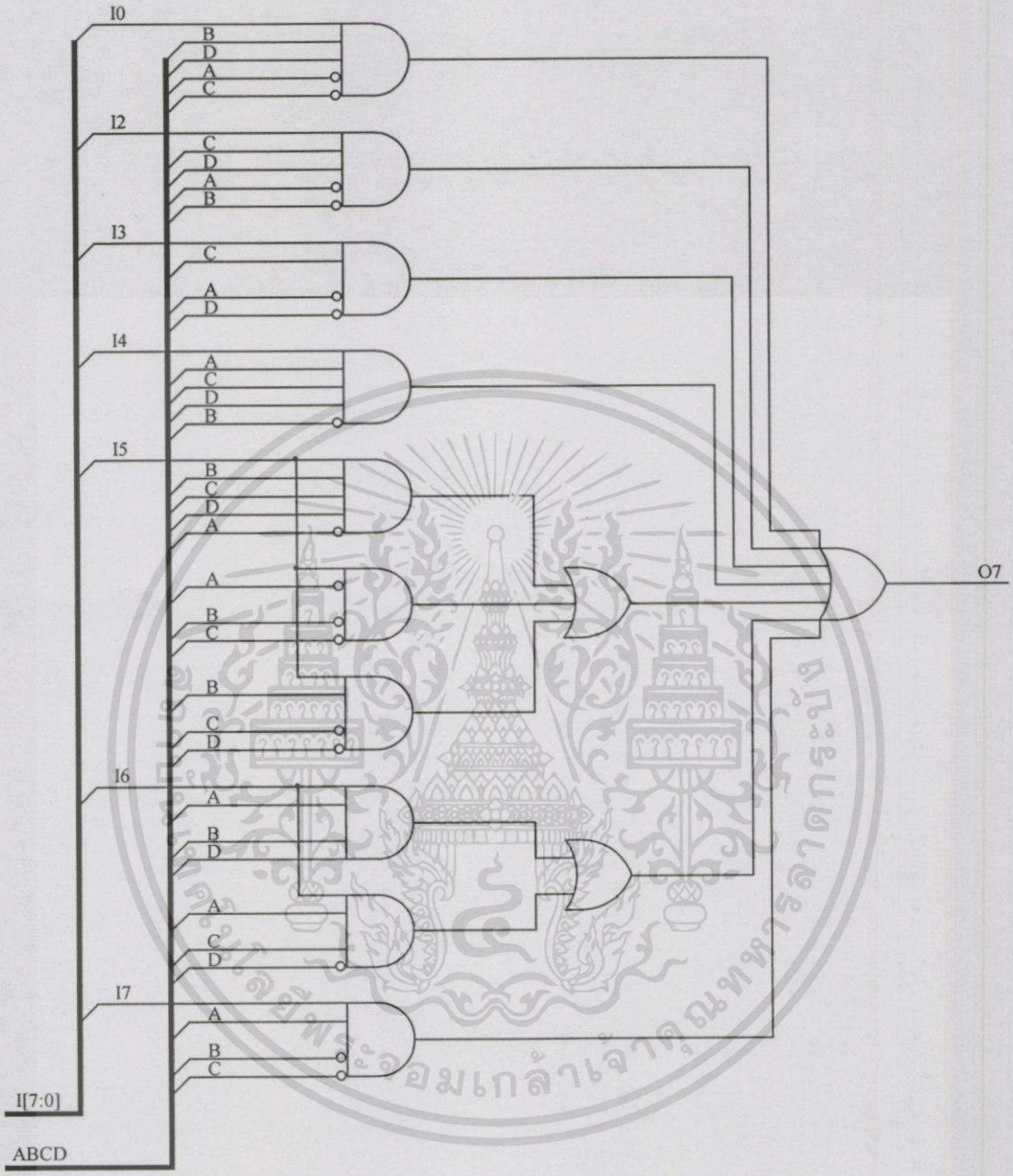
รูปที่ ข.11 วงจรสลับตำแหน่งบิตภาคส่ง บล็อค 4



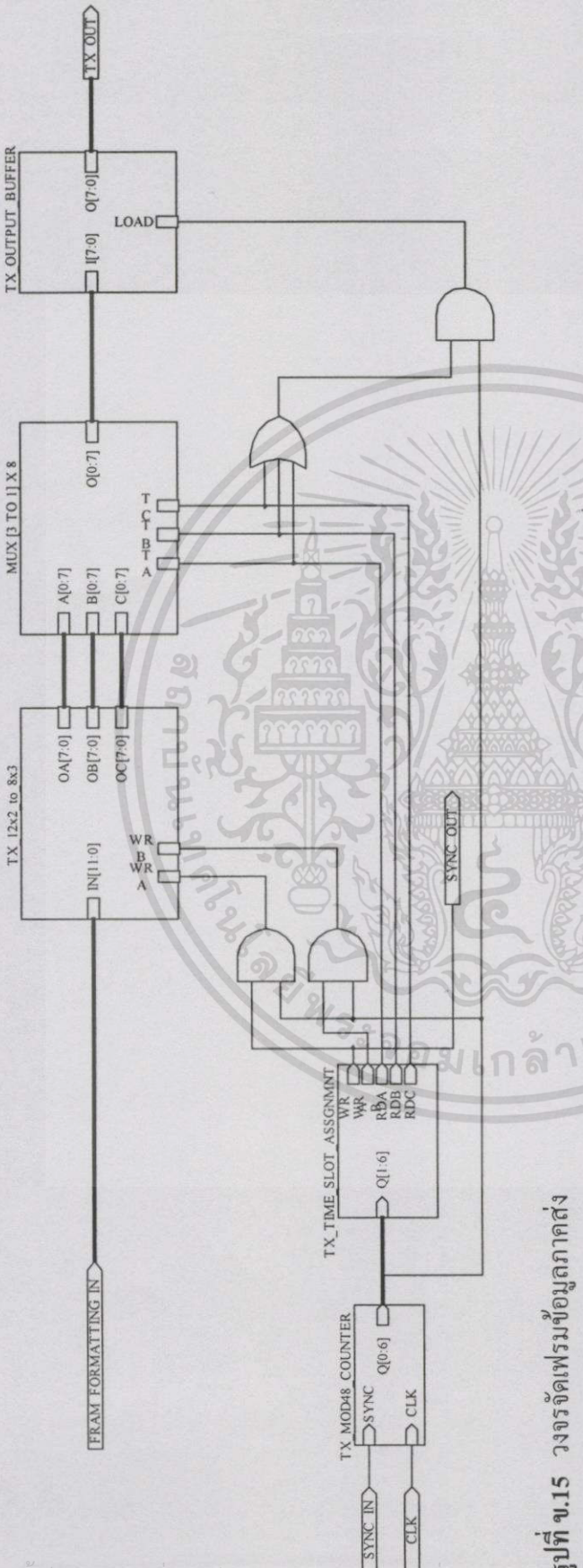
รูปที่ ข.12 วงจรสลับตำแหน่งบิตภาคส่ง บล็อค 5



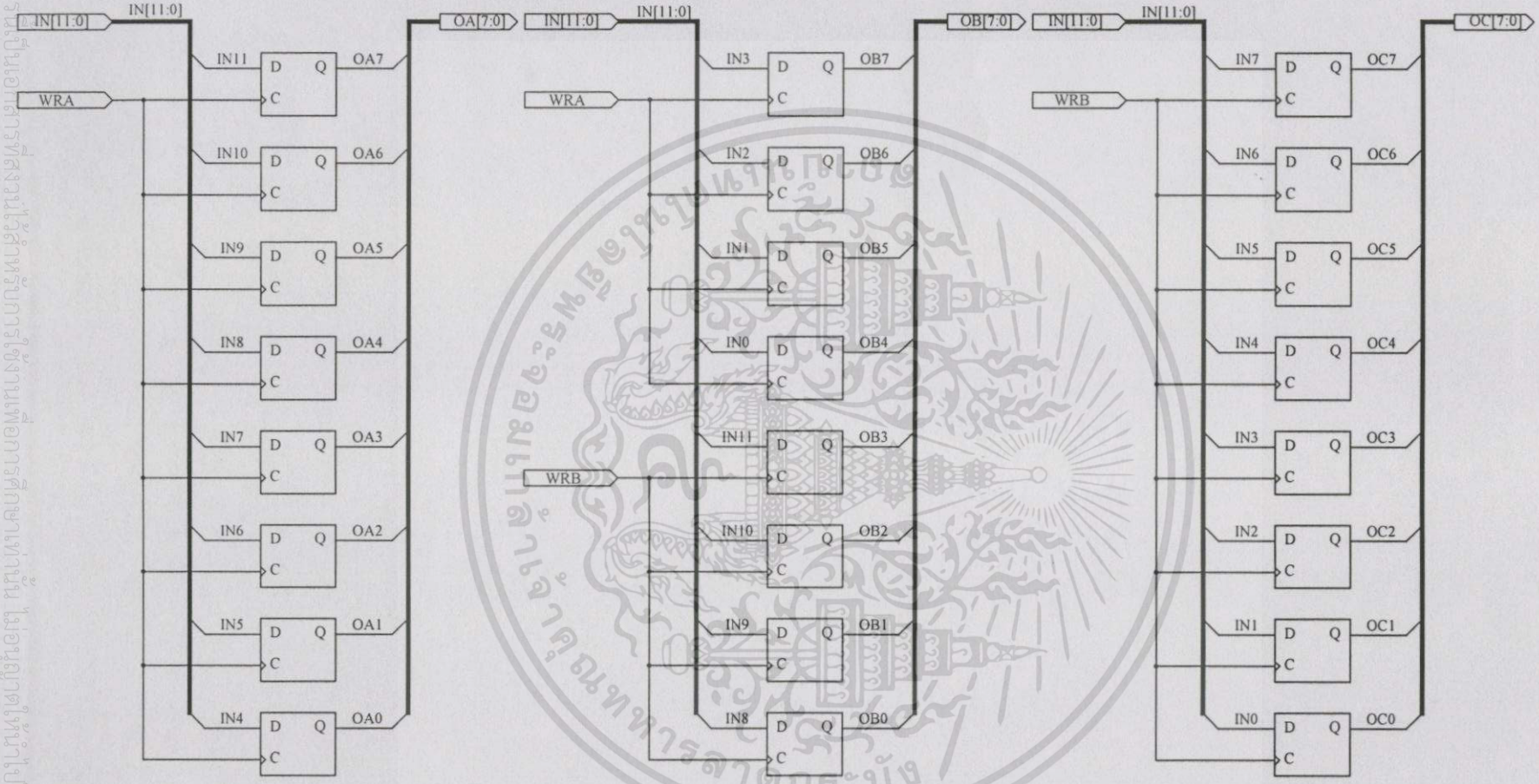
รูปที่ ข.13 วงจรสลับตำแหน่งบิตภาคส่ง บล็อค 6



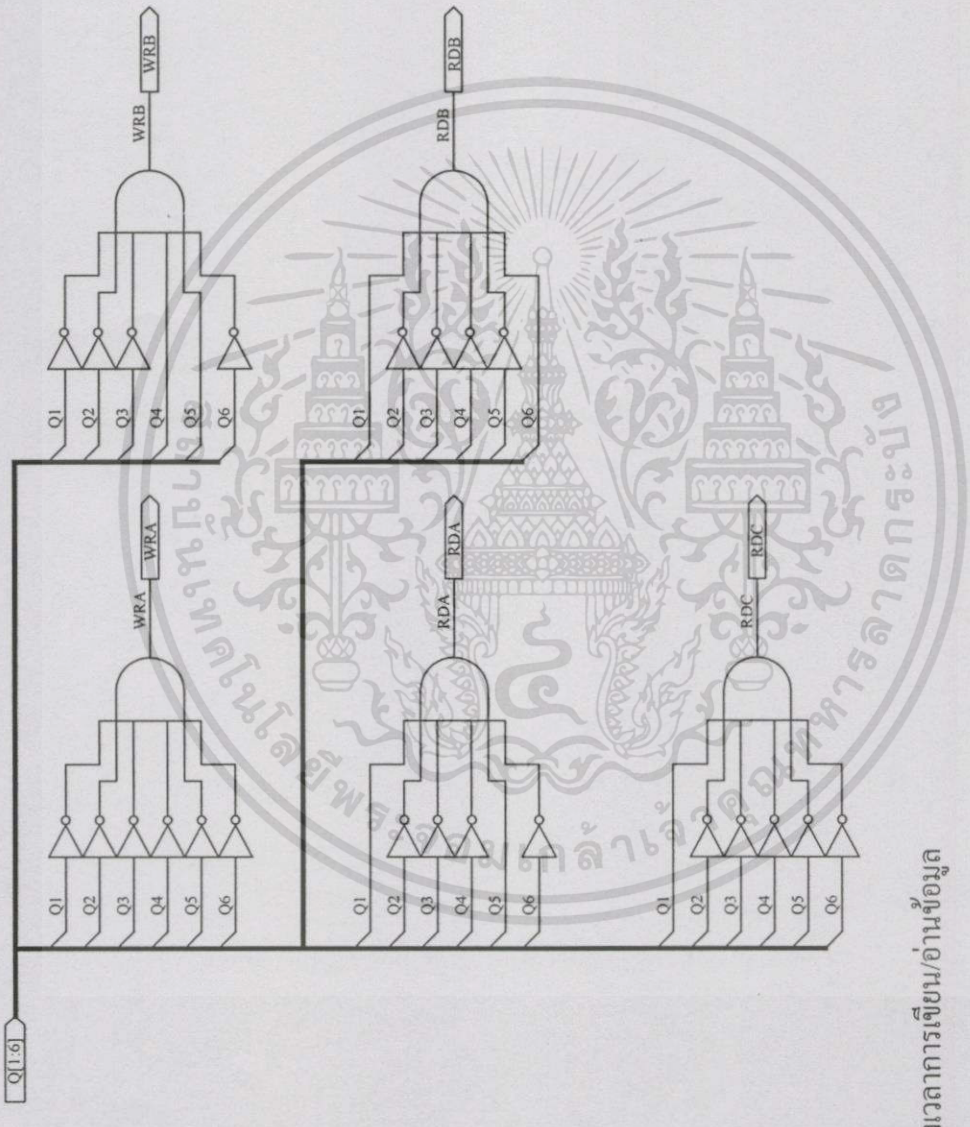
รูปที่ ข.14 วงจรสลับตำแหน่งบิตภาคส่ง บล็อค 7



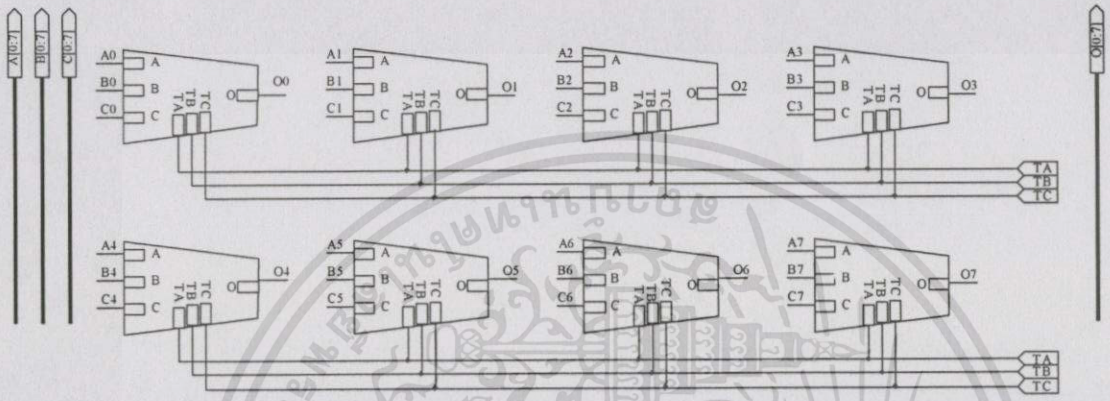
รูปที่ ข.15 วงจรจัดเฟรมข้อมูลภาคส่ง



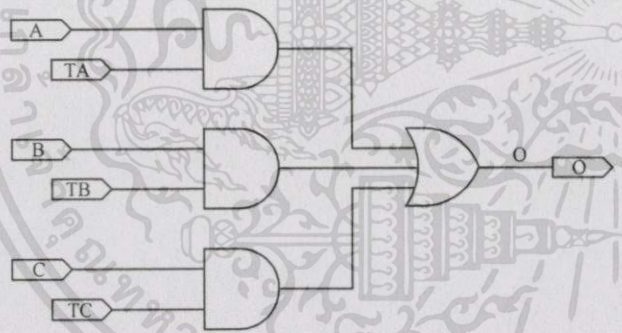
รูปที่ ข.16 บัฟเฟอร์สำหรับจัดเฟรมภาคส่ง



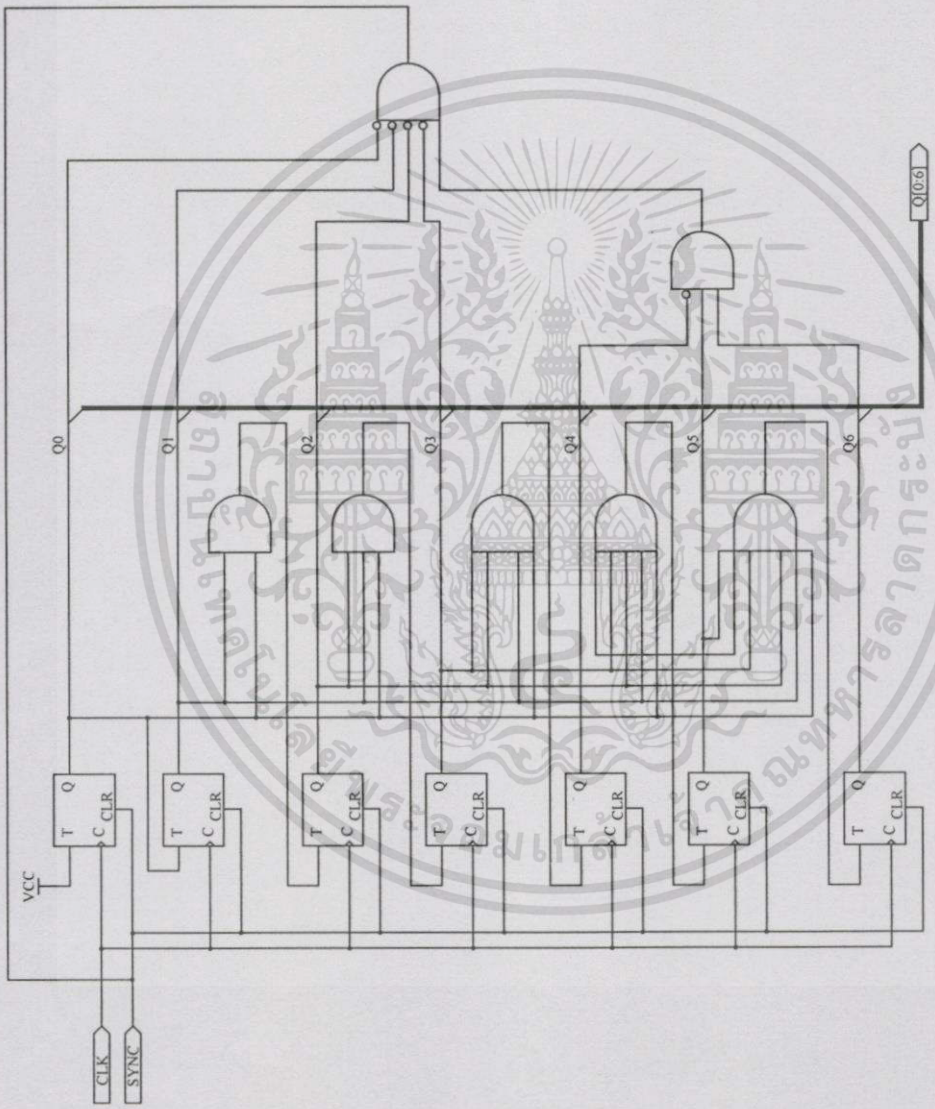
รูปที่ ข.17 วงจรกำหนดช่วงเวลาการเขียน/อ่านข้อมูล



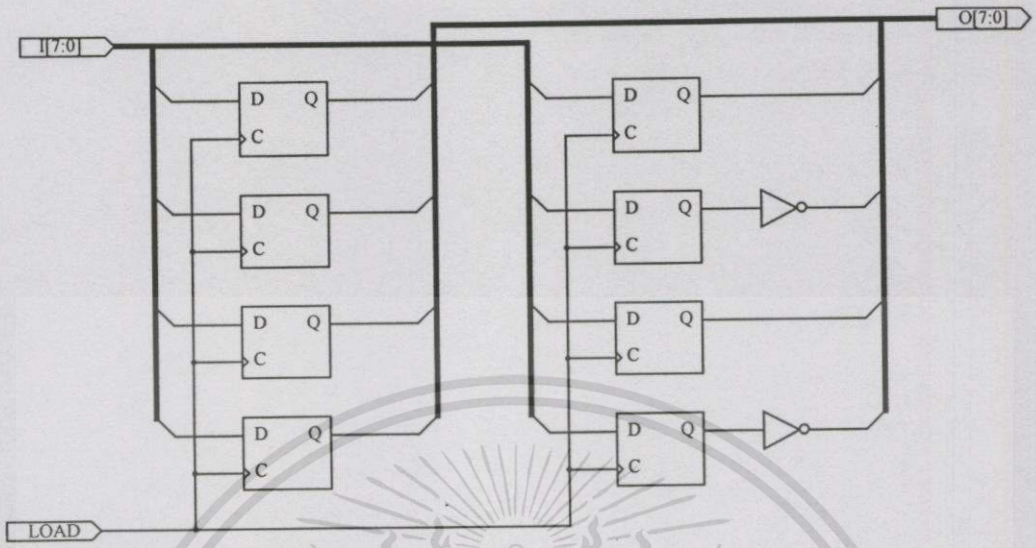
รูปที่ ข.18 วงจรมัลติเพล็กซ์ เข้า 3 ออก 1 ขนาด 8 บิต



รูปที่ ข.19 วงจรมัลติเพล็กซ์ เข้า 3 ออก 1 ขนาด 1 บิต



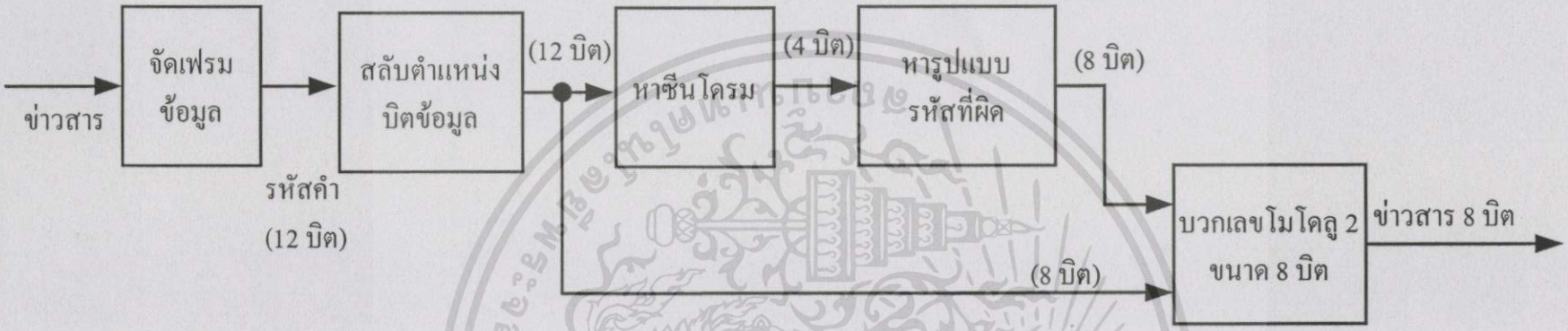
รูปที่ ข.20 วงจรหารความถี่เพื่อจัดช่องเวลาให้สอดคล้องกับความเร็วข้อมูล



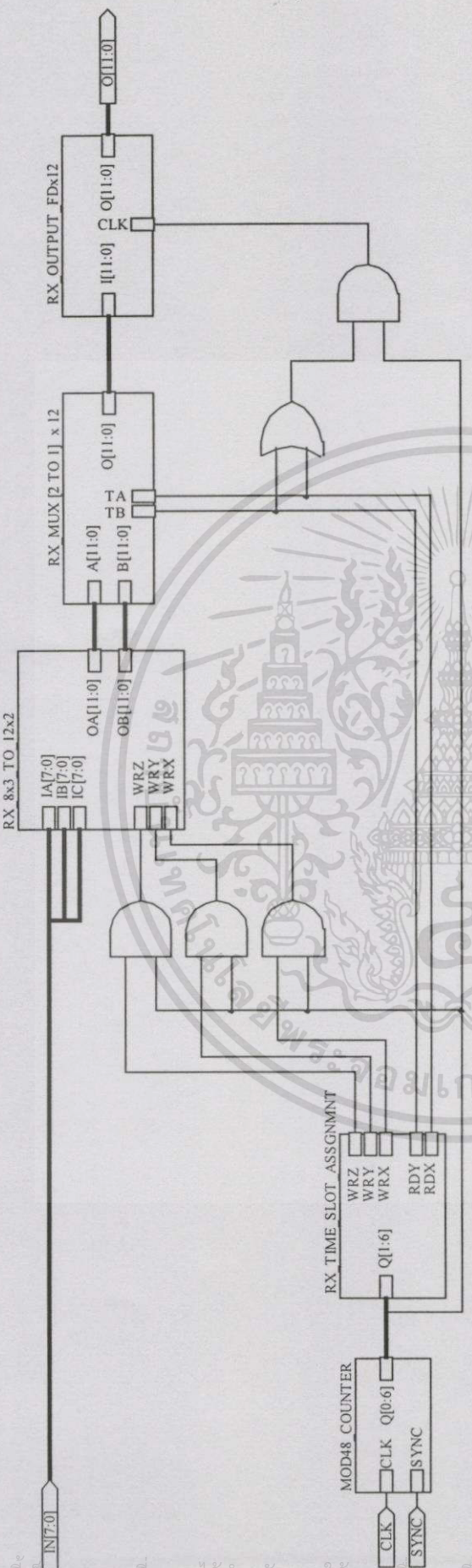
รูปที่ ข.21 บัฟเฟอร์ข้อมูลขาออกของภาคส่ง

ภาคผนวก ค

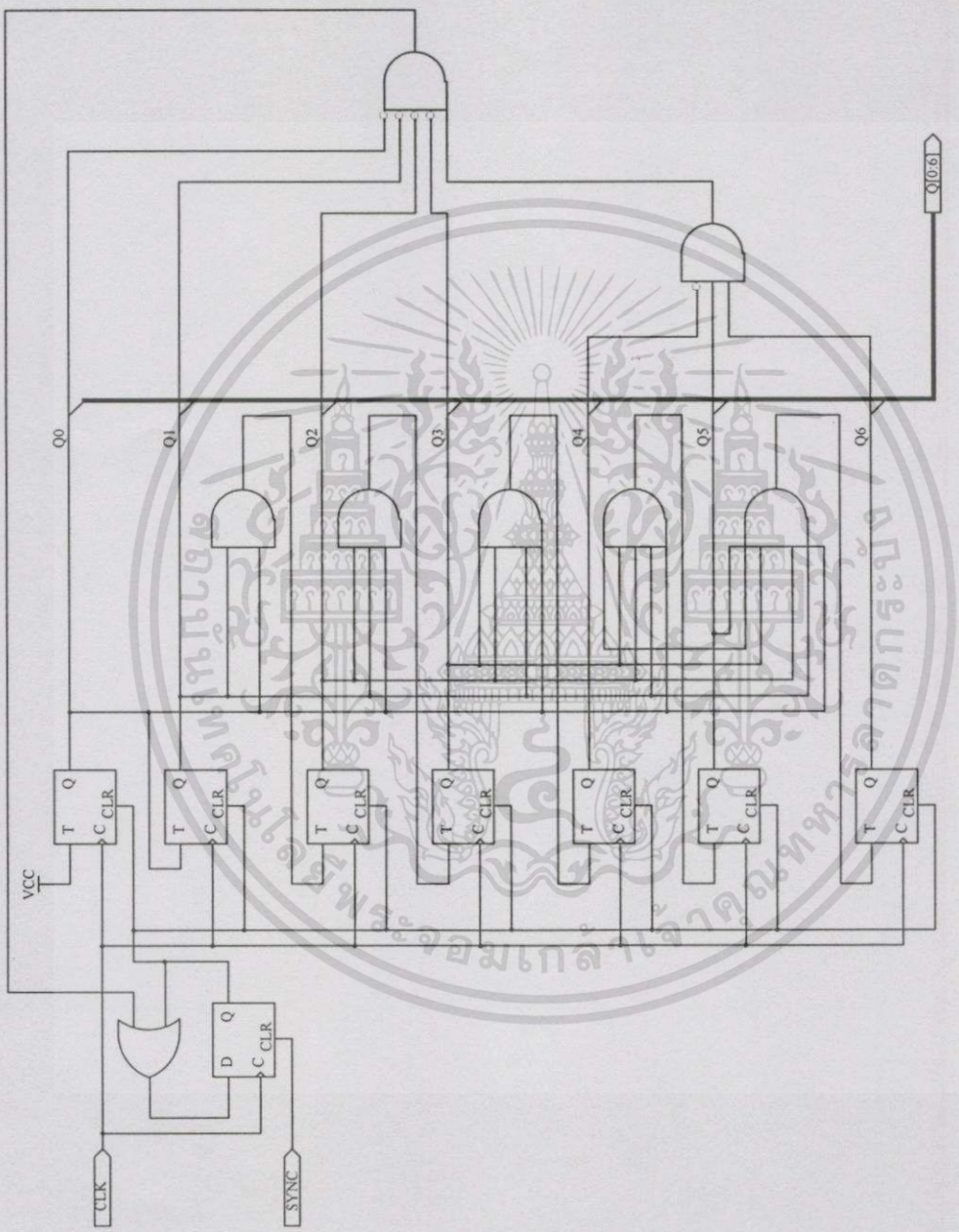
วงจรในภาครับ



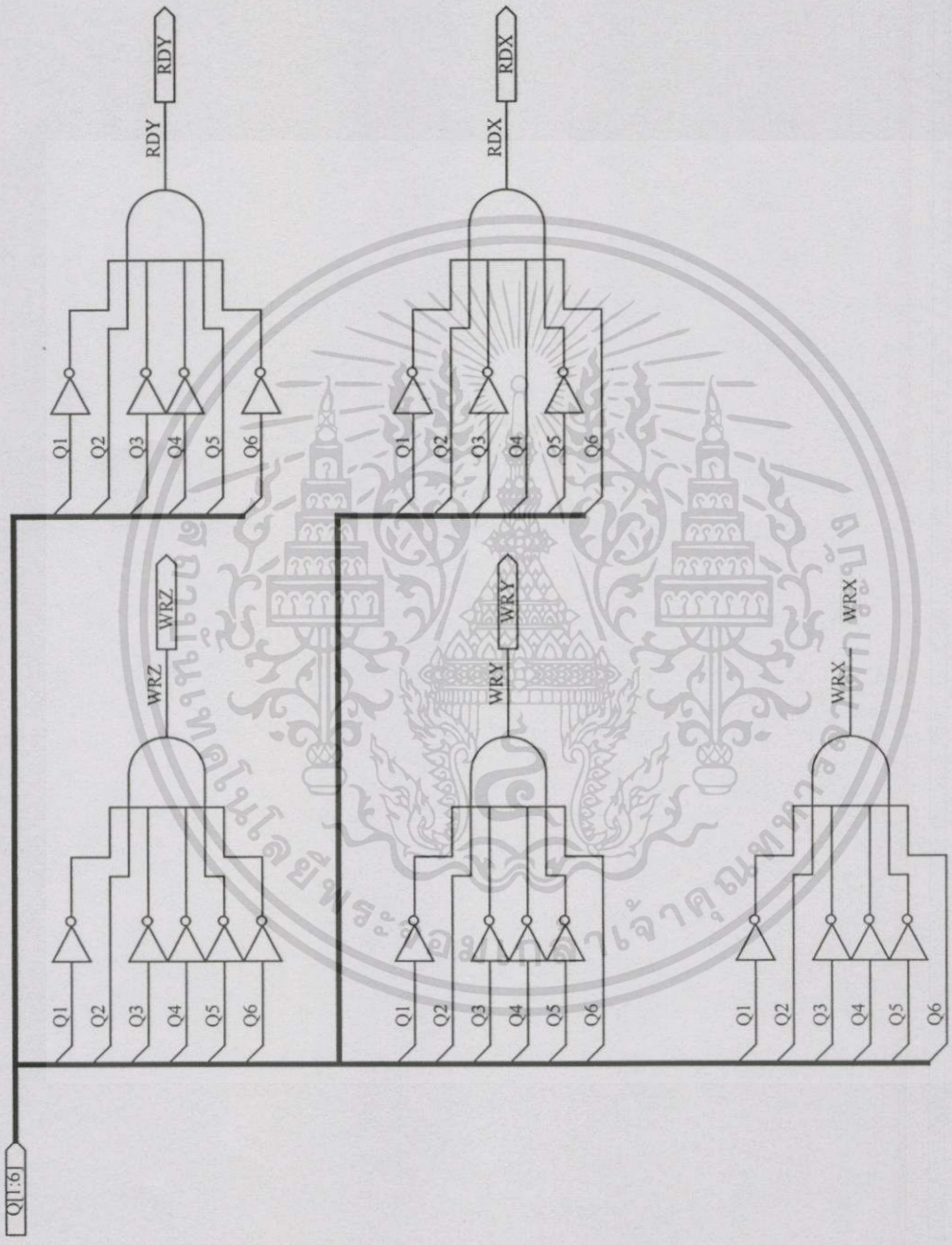
รูปที่ ค.1. ผังวงจรของภาครับ



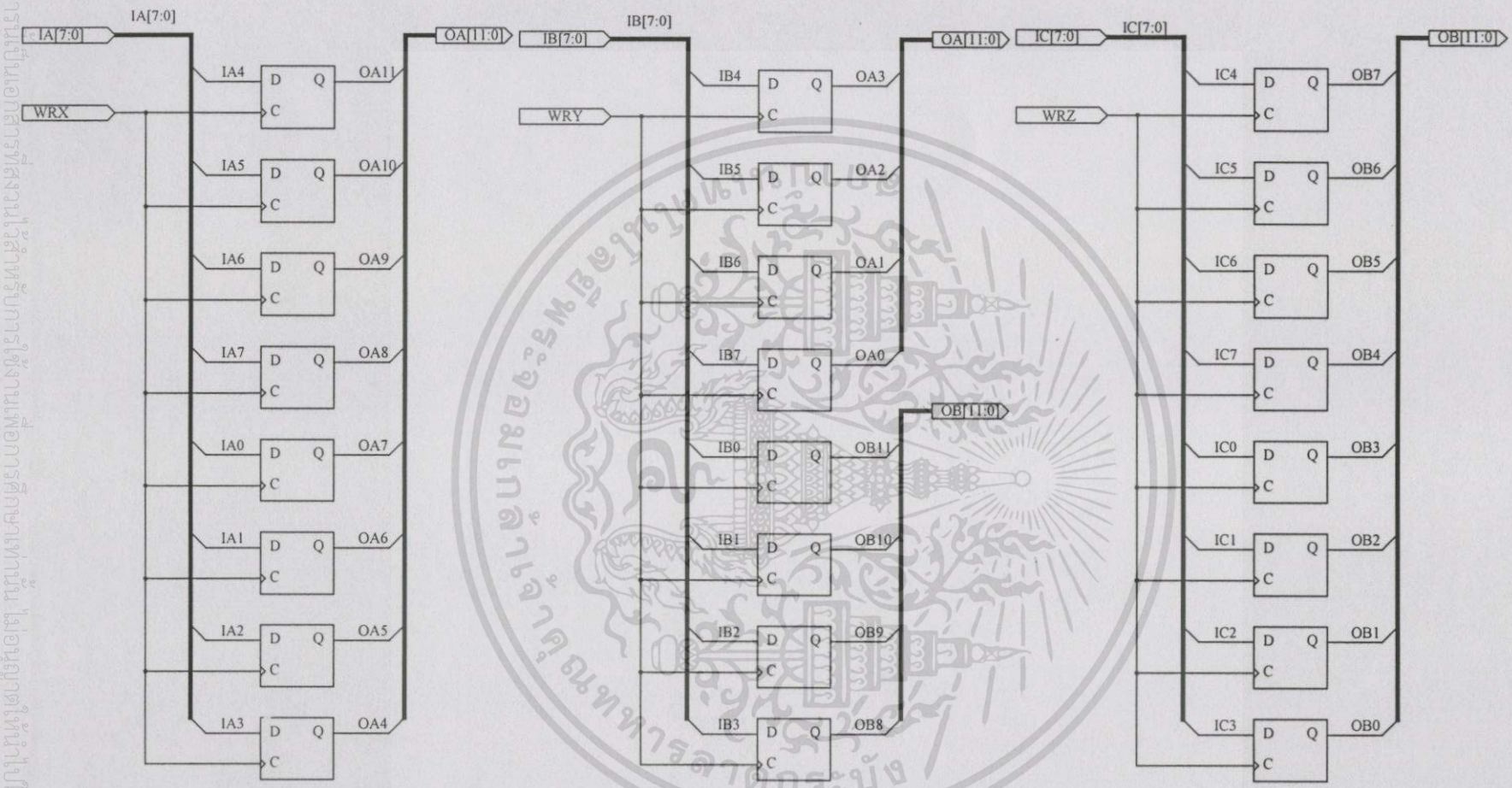
รูปที่ ค.2 ผังงงจรการจัดเฟรมข้อมูล



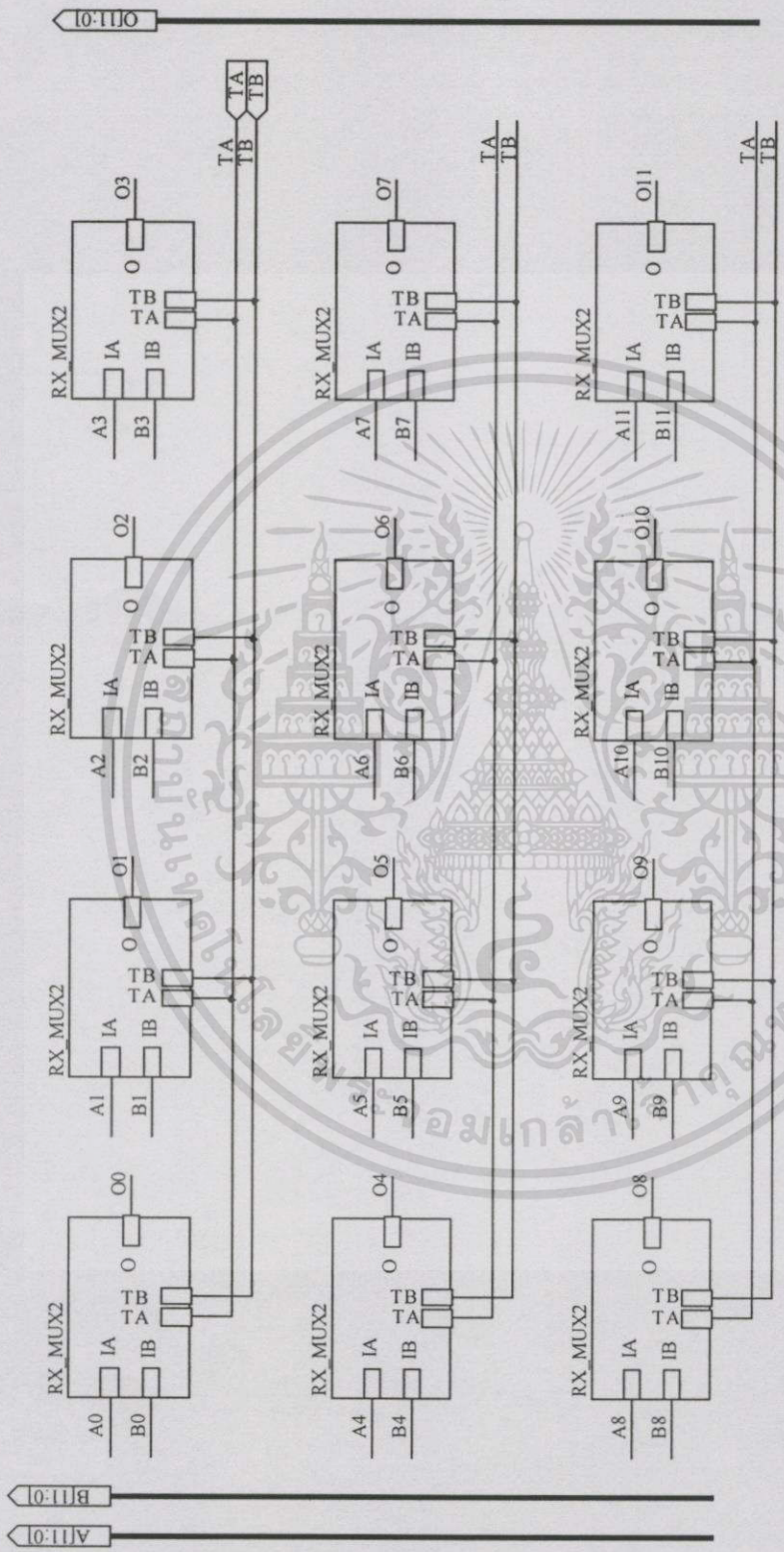
รูปที่ ค.3 วงจรนับ 48 และหารความถี่ของภาครับ



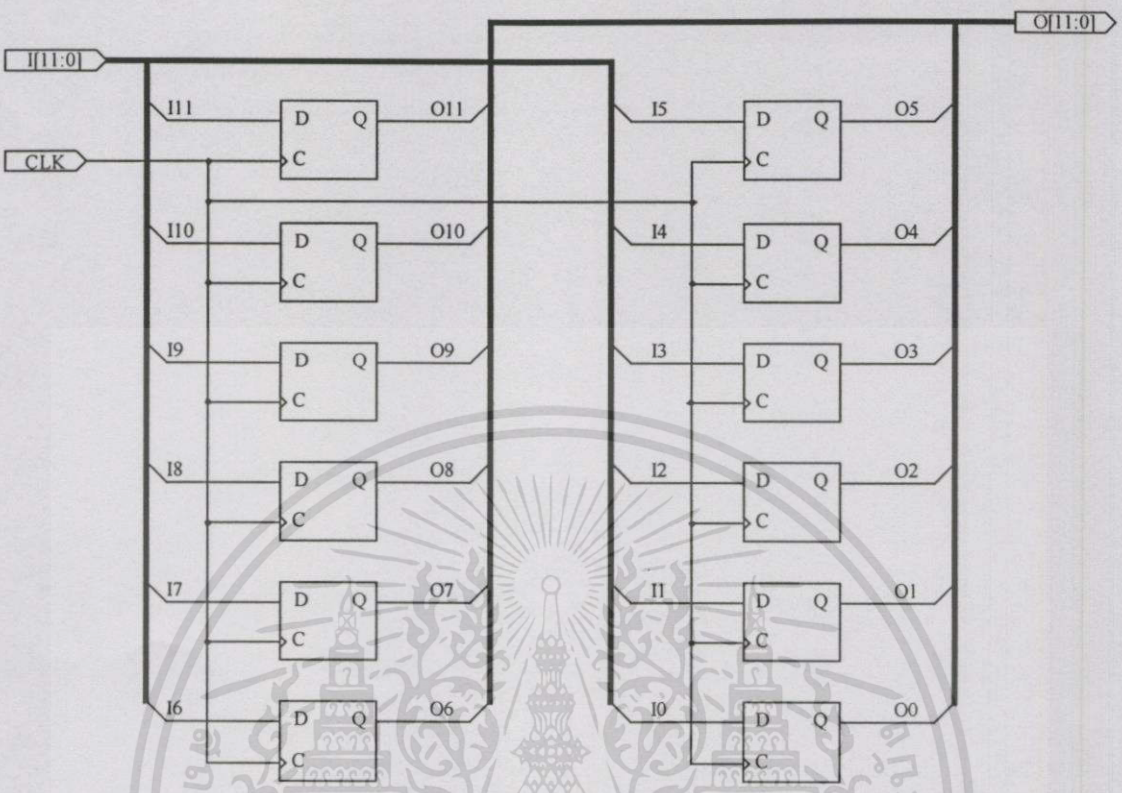
รูปที่ ค.4 วงจรสร้างสัญญาณเขียนและอ่านข้อมูล



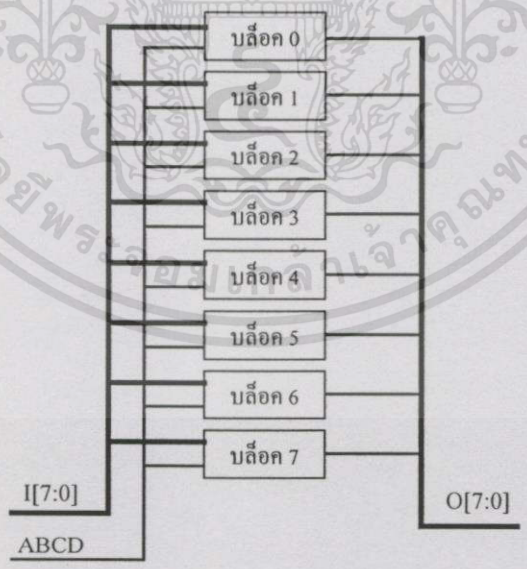
รูปที่ ค.5 หน่วยความจำข้อมูลสำหรับจัดเรียงข้อมูล 8 บิตx 3 เป็น 12 บิต x 2



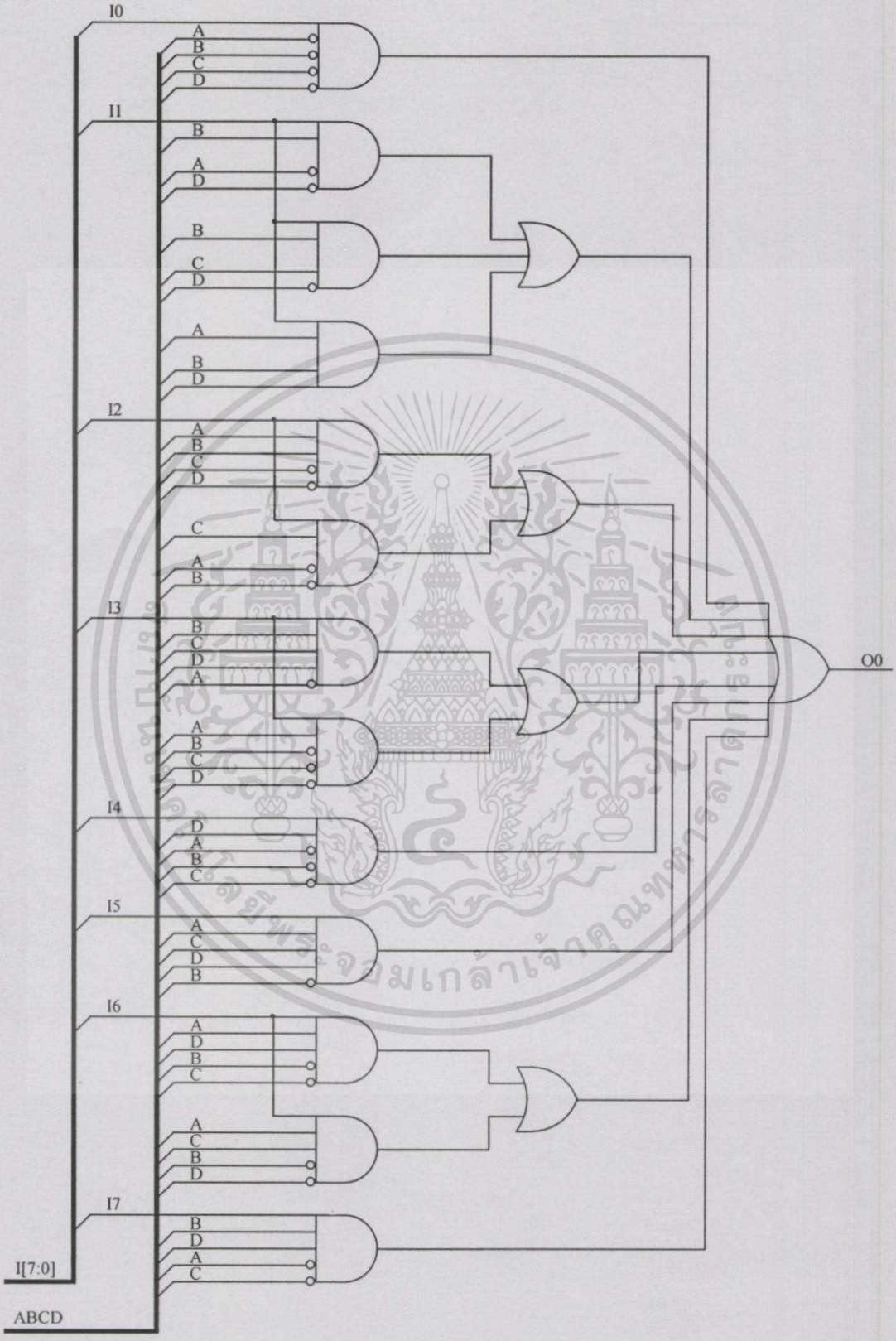
รูปที่ ค.6 วงจรมัลติเพล็กซ์ 2 ออก 1 ขนาด 12 บิต



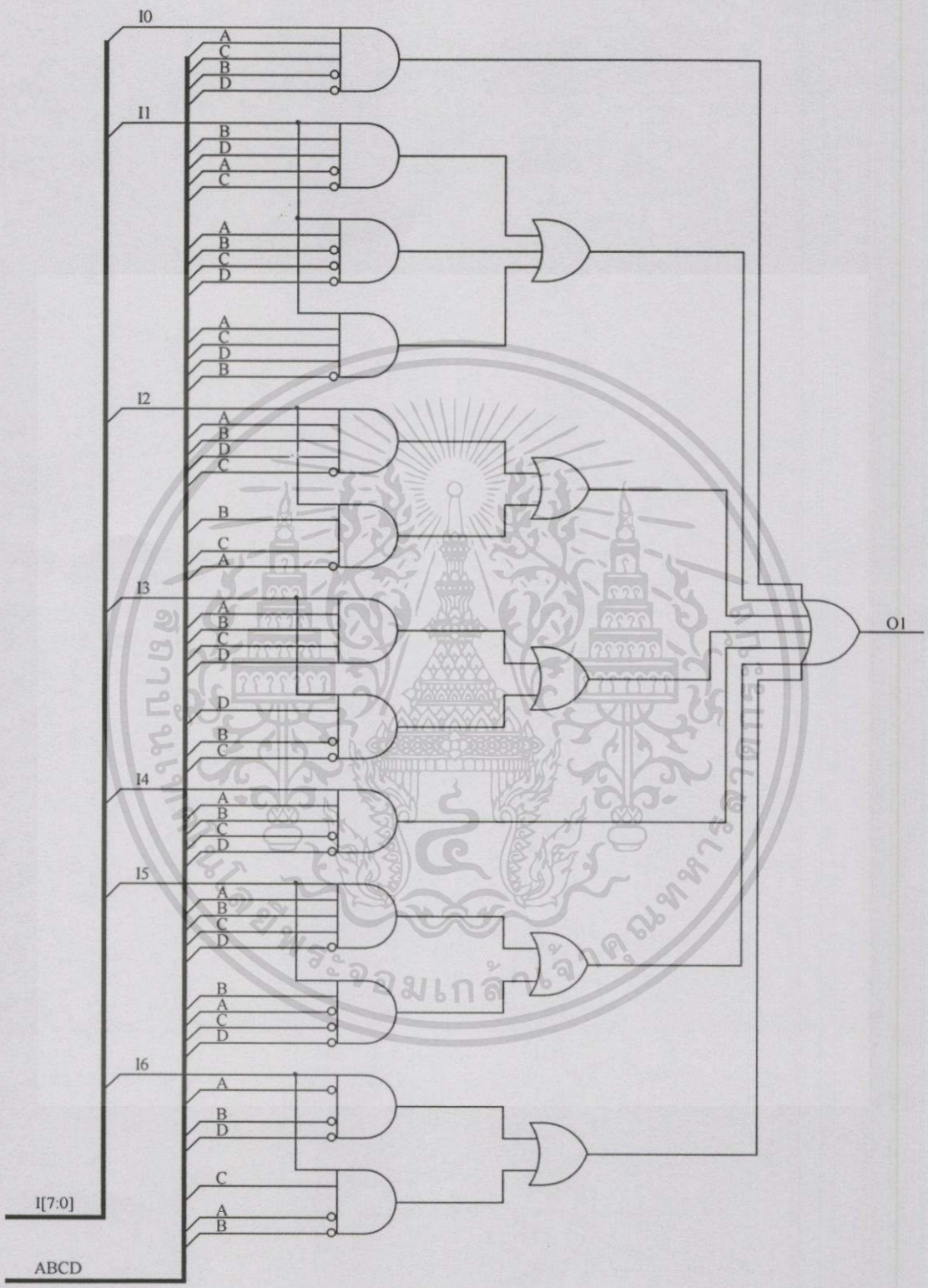
รูปที่ ค.7 บัฟเฟอร์ข้อมูลเอาท์พุตของวงจรจัดเฟรมข้อมูลภาครับ



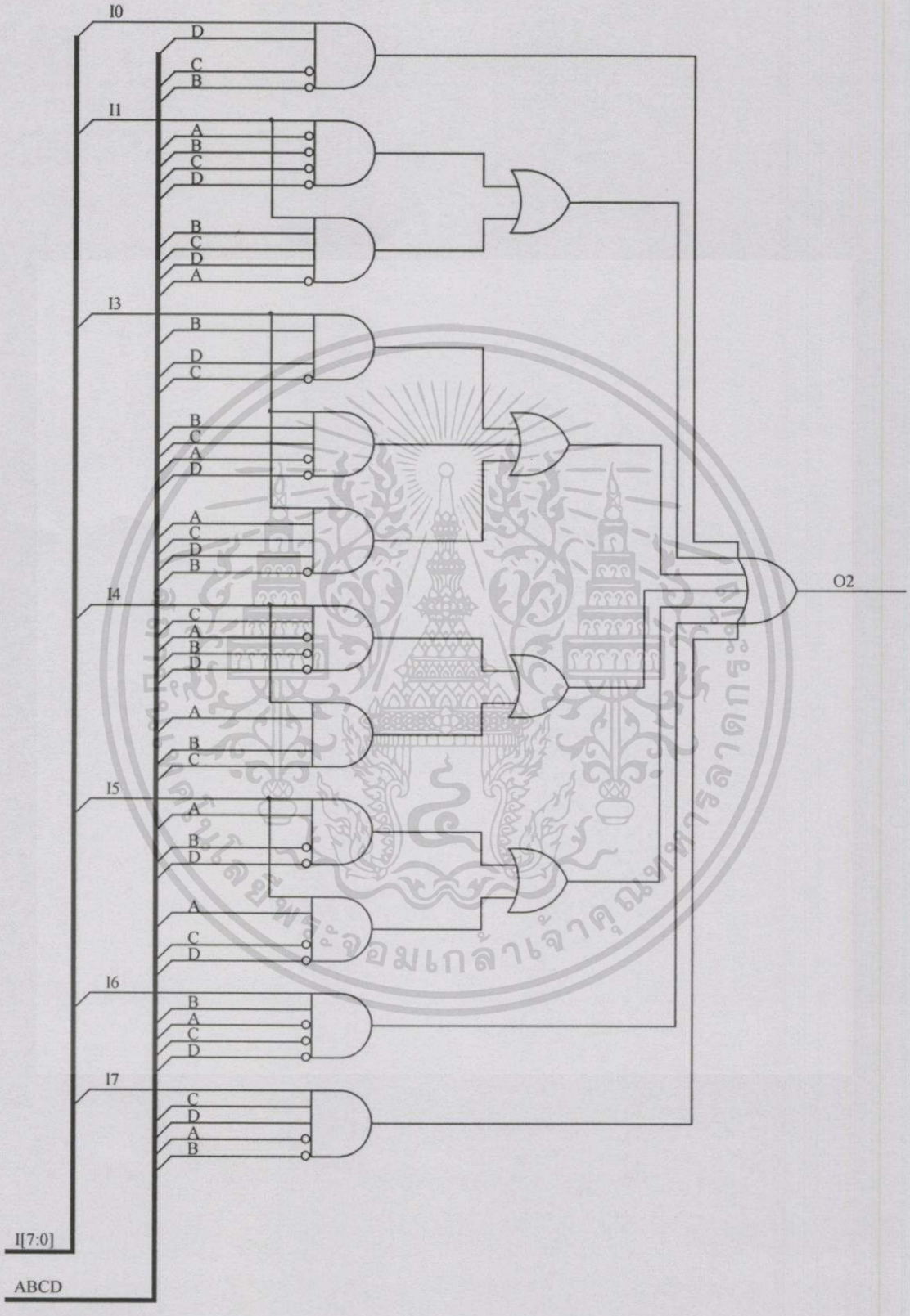
รูปที่ ค.8 ผังวงจรสลับตำแหน่งบิตภาครับ



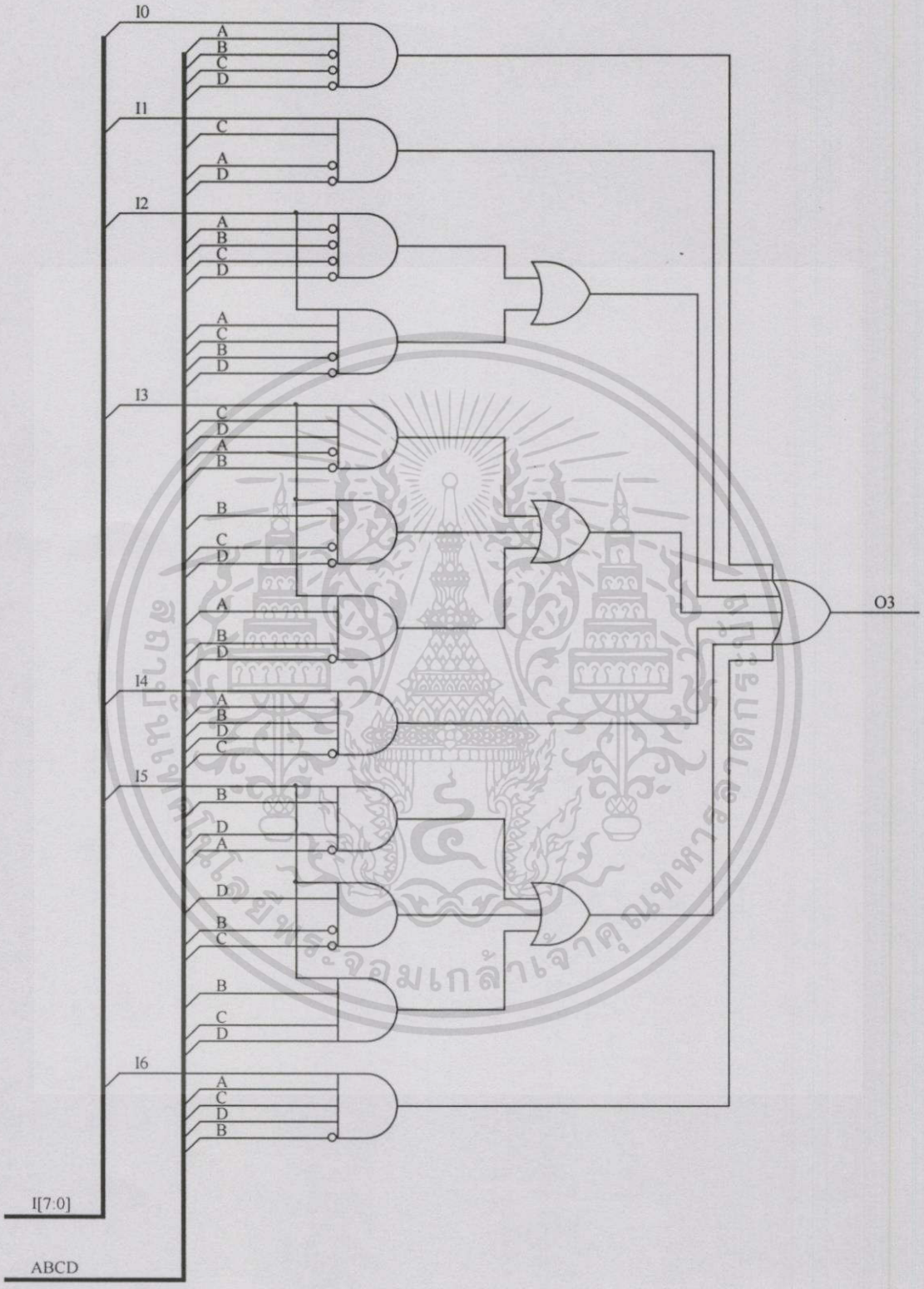
รูปที่ ค.9 วงจรสลับตำแหน่งบิตภาครับ บล็อก 0



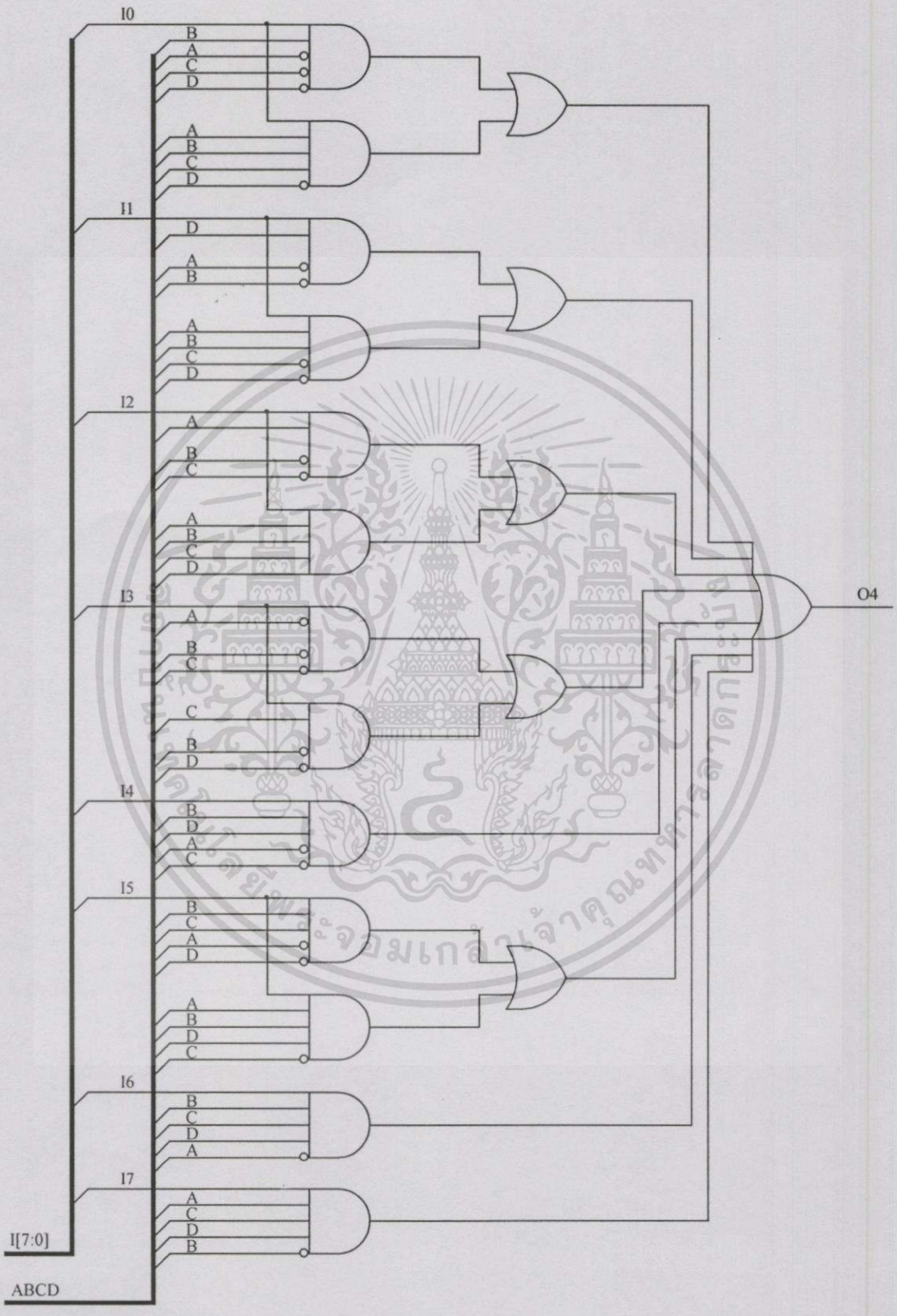
รูปที่ ค.10 วงจรสลับตำแหน่งบิตภาครับ บล็อก 1



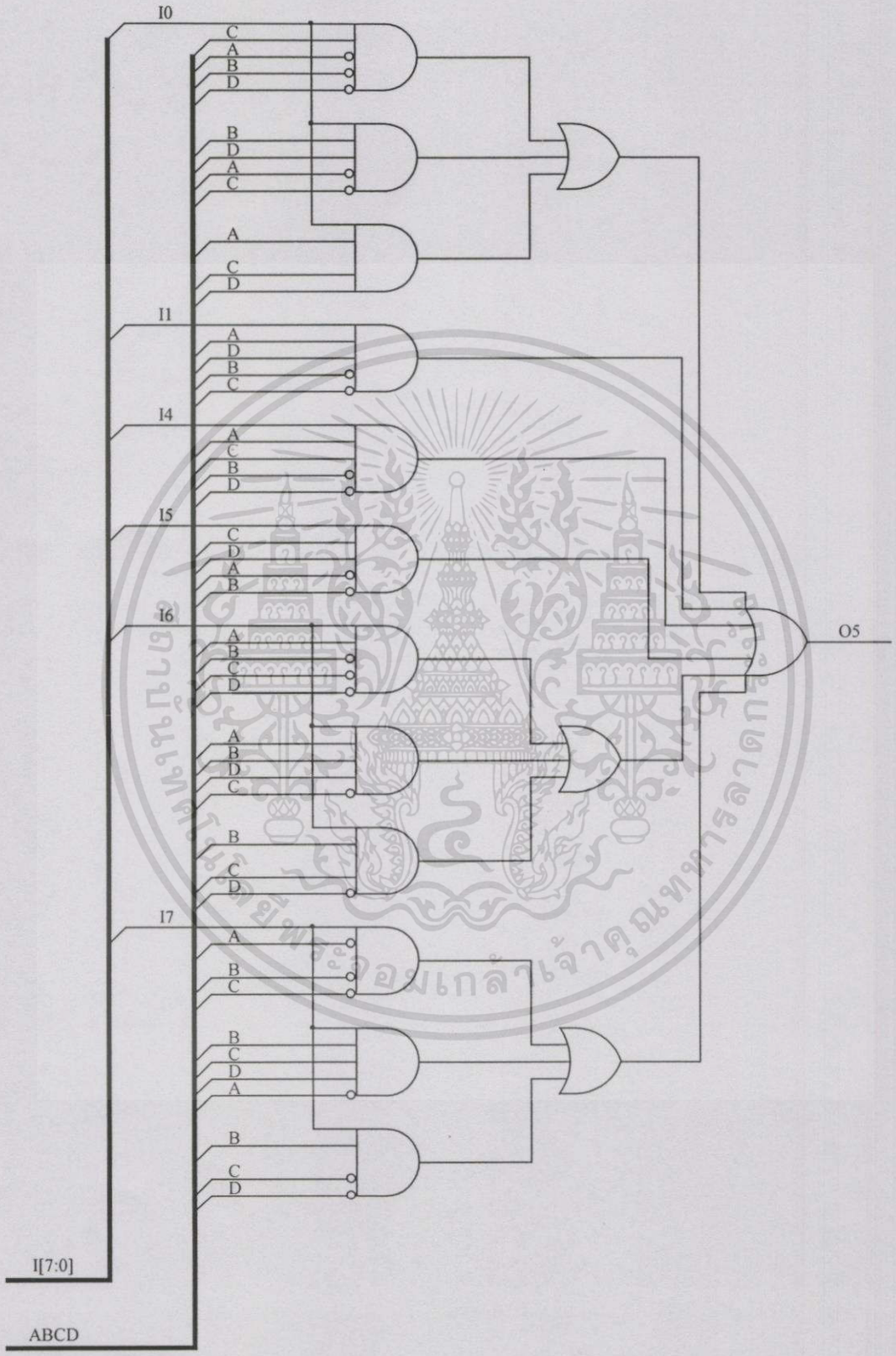
รูปที่ ค.11 วงจรสลับตำแหน่งบิตภาครับ บล็อก 2



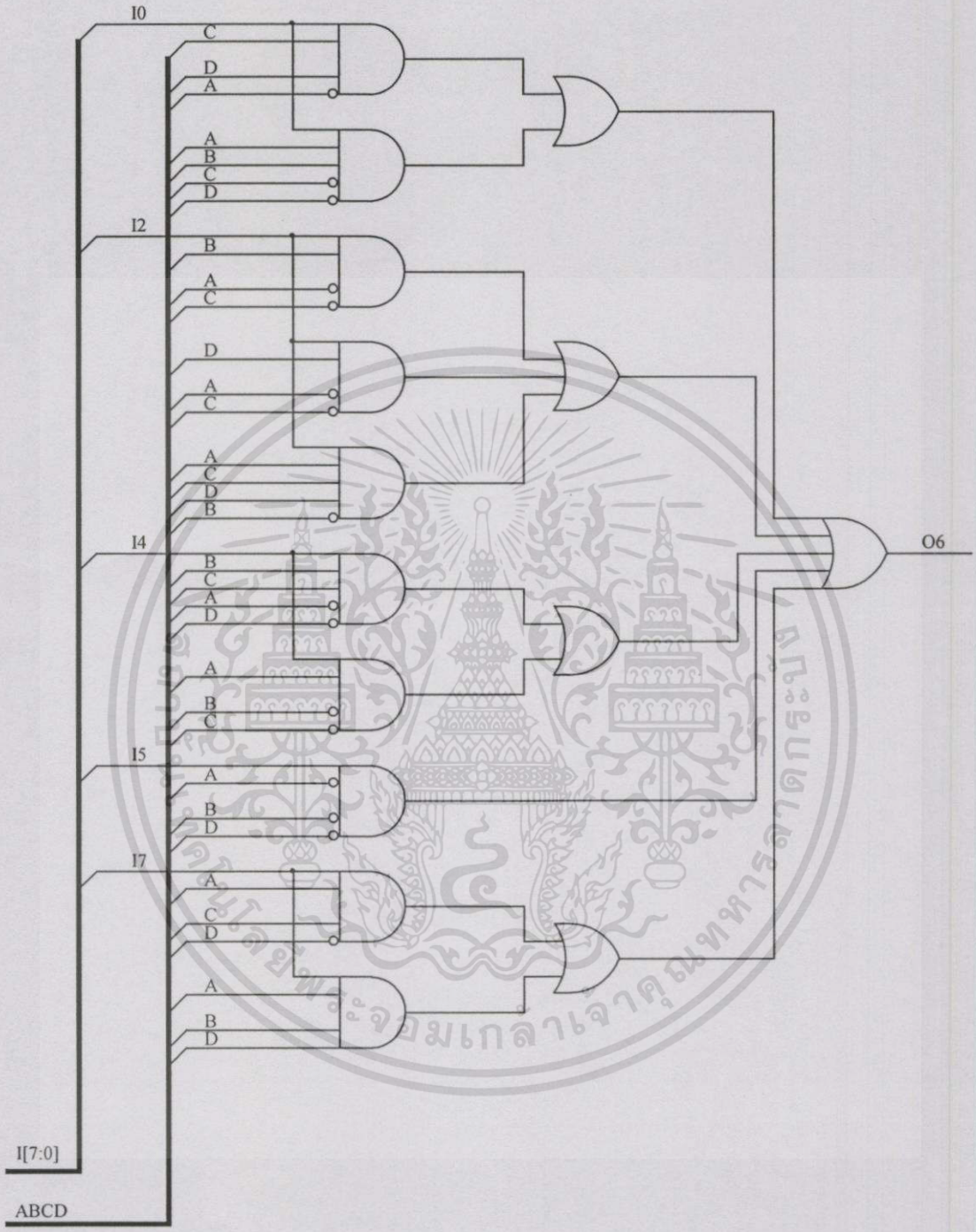
รูปที่ ค.11 วงจรสลับตำแหน่งบิตภาครับ บล็อก 3



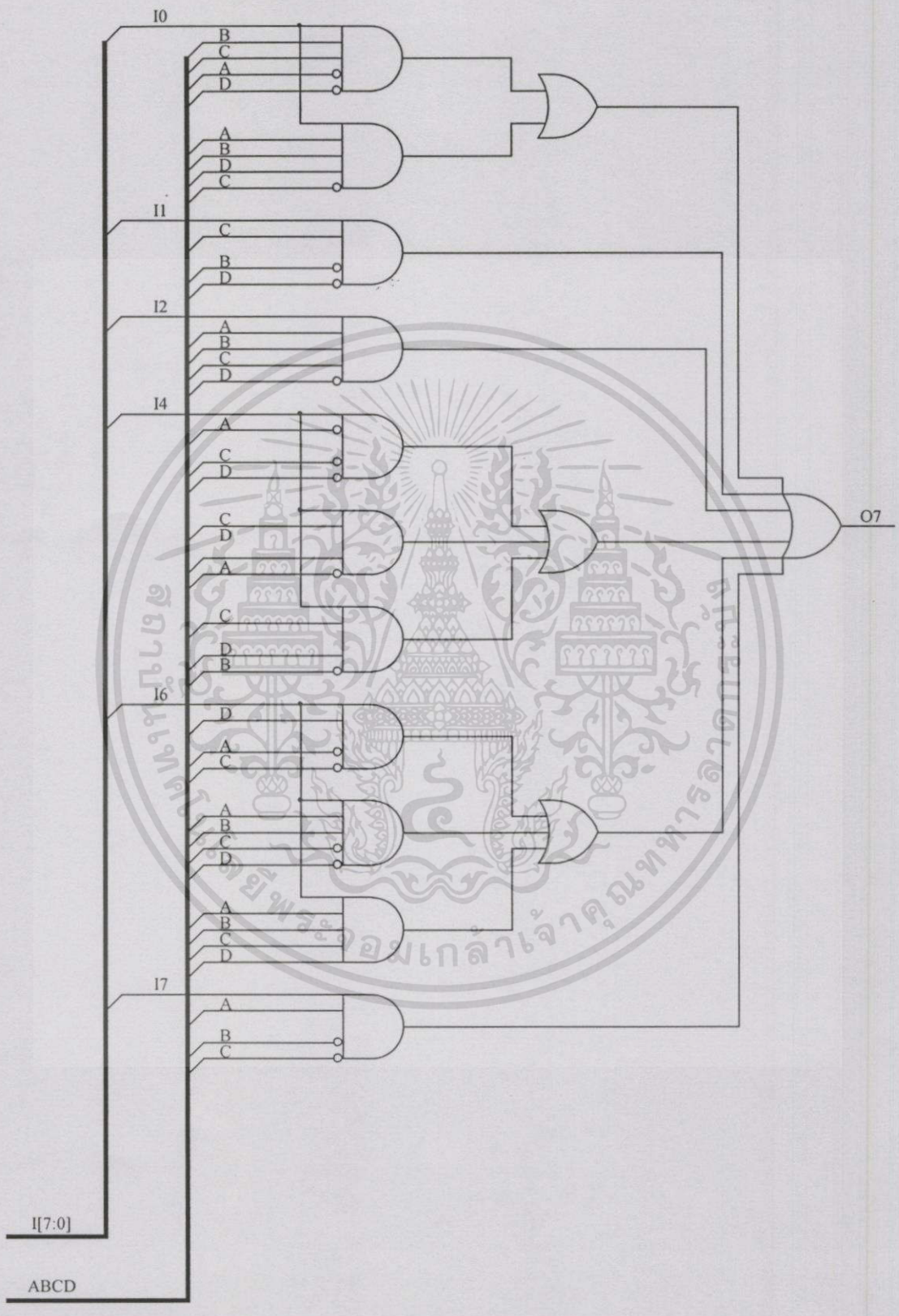
รูปที่ ค.12 วงจรสลับตำแหน่งบิตภาครับ บล็อก 4



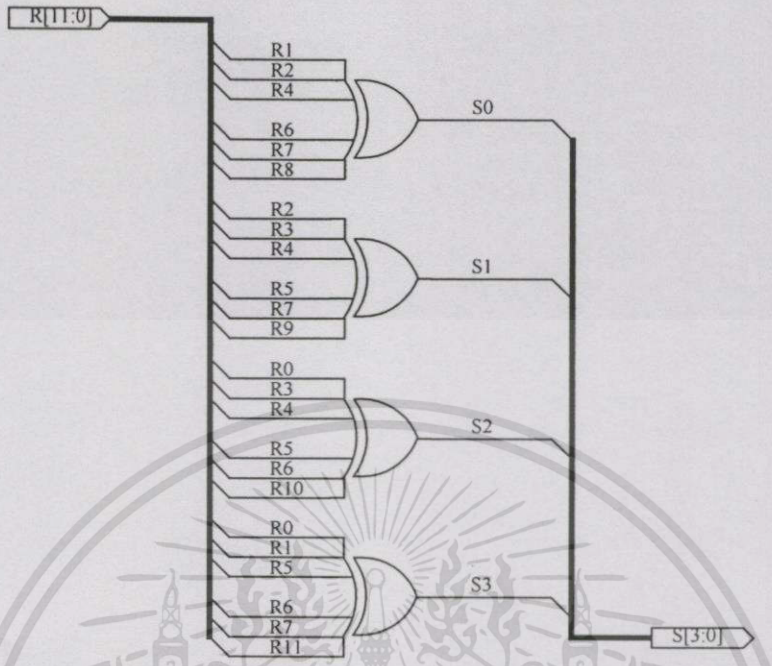
รูปที่ ค.13 วงจรสลับตำแหน่งบิตภาครับ บล็อก 5



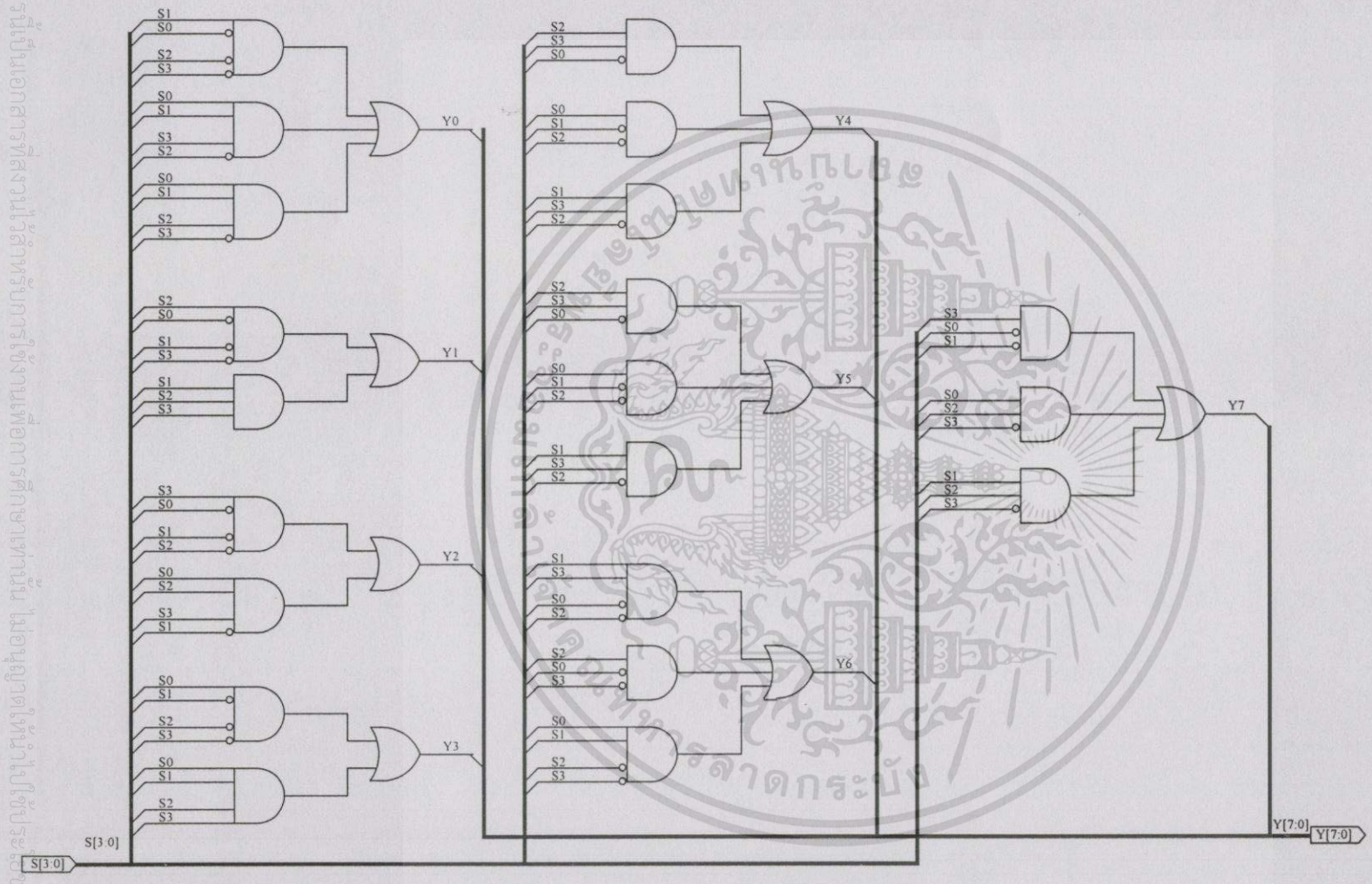
รูปที่ ค.14 วงจรสลับตำแหน่งบิตภาครับ บล็อก 6



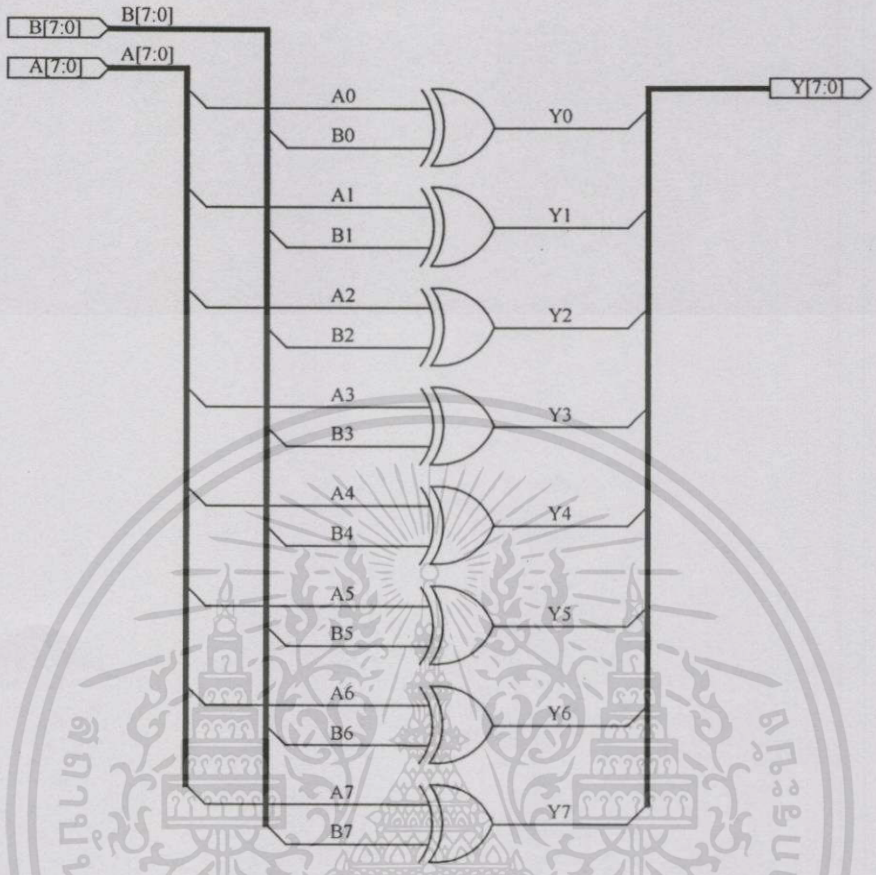
รูปที่ ค.15 วงจรสลับตำแหน่งบิตภาครับ บล็อก 7



รูปที่ ค.16 วงจรหาซีนโดรม



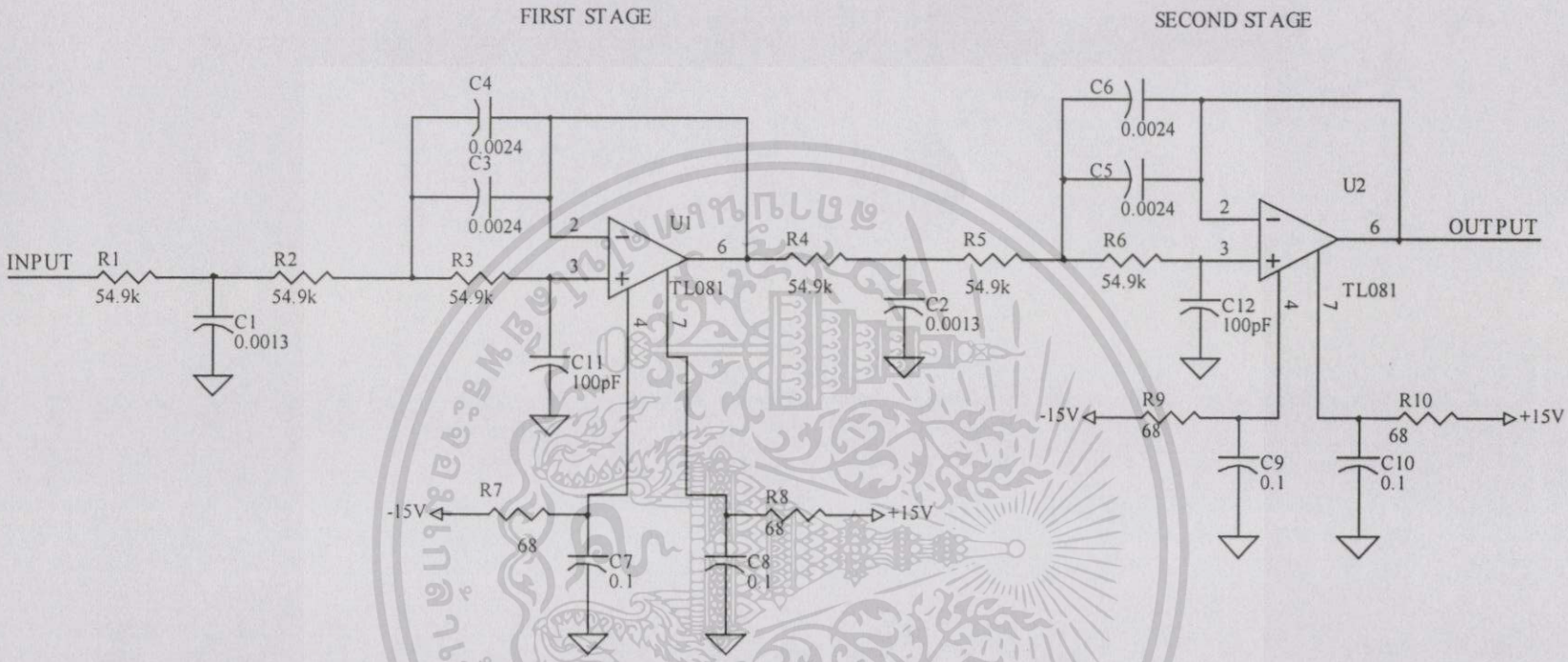
รูปที่ ค.17 วงจรเปิดตารางซีนโดรม



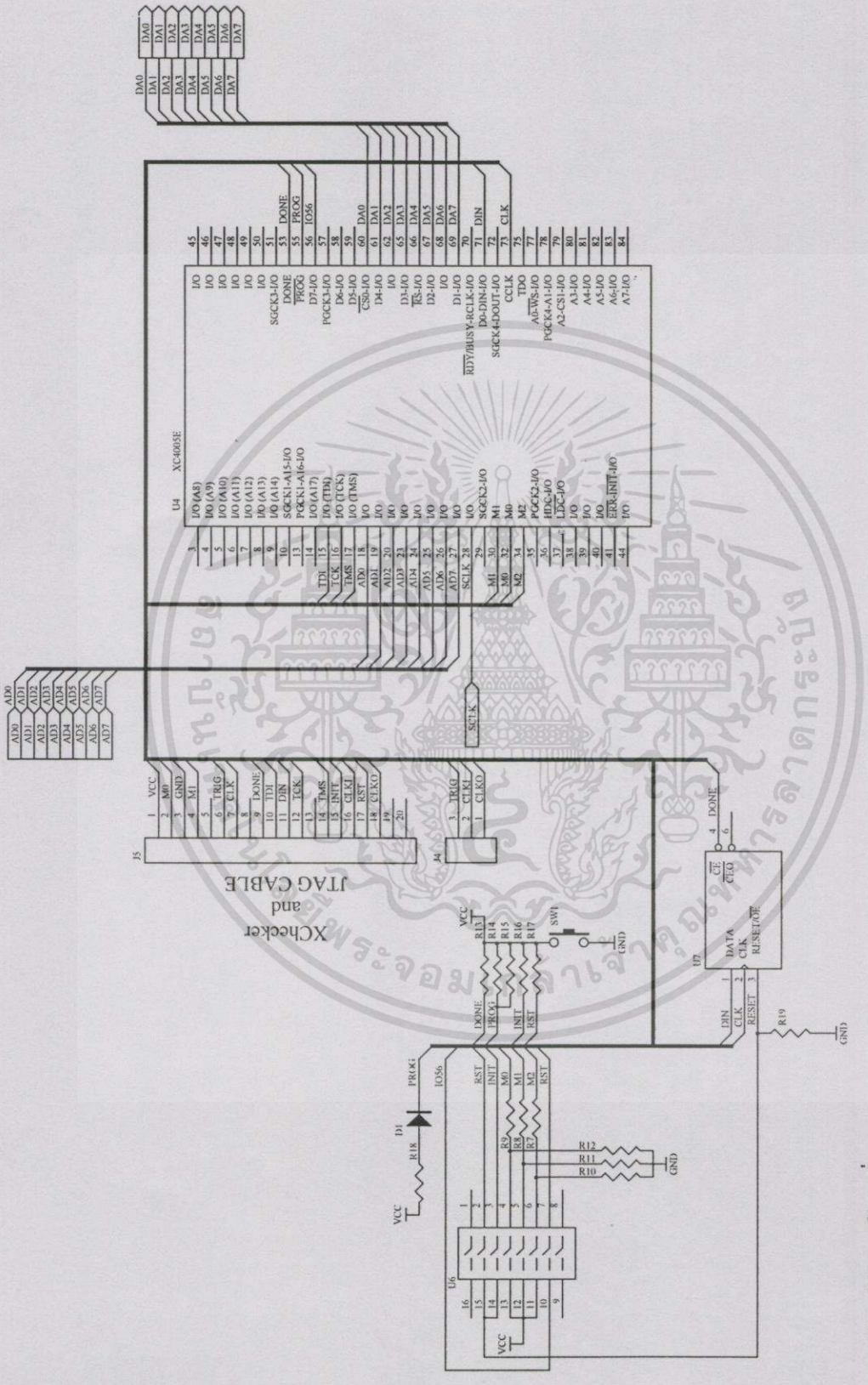
รูปที่ ค.18 วงจรบวกเลขแบบโมดูล 2 สำหรับกลับค่าบิตที่ผิด

ภาคผนวก ง

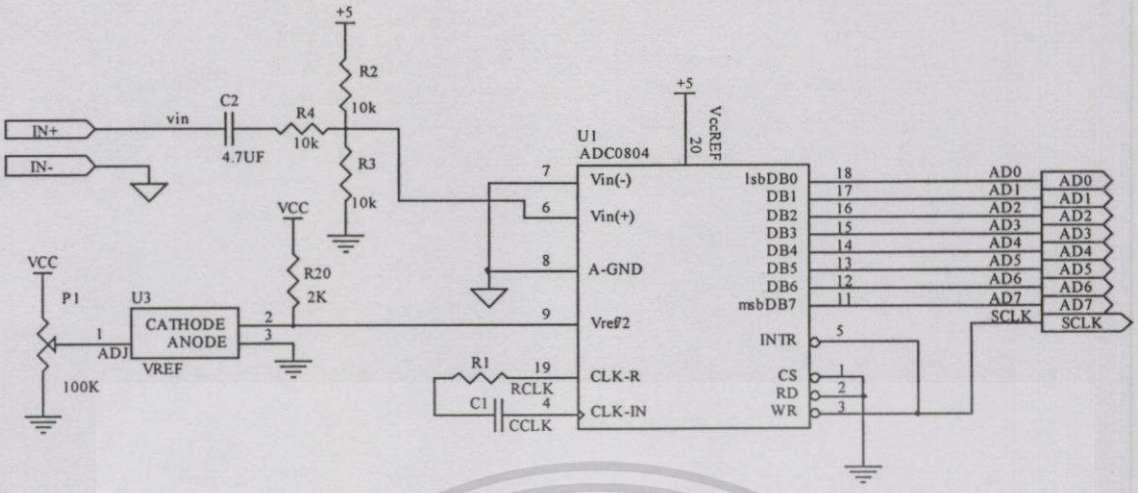
วงจร FPGA และวงจรประกอบการทดลอง



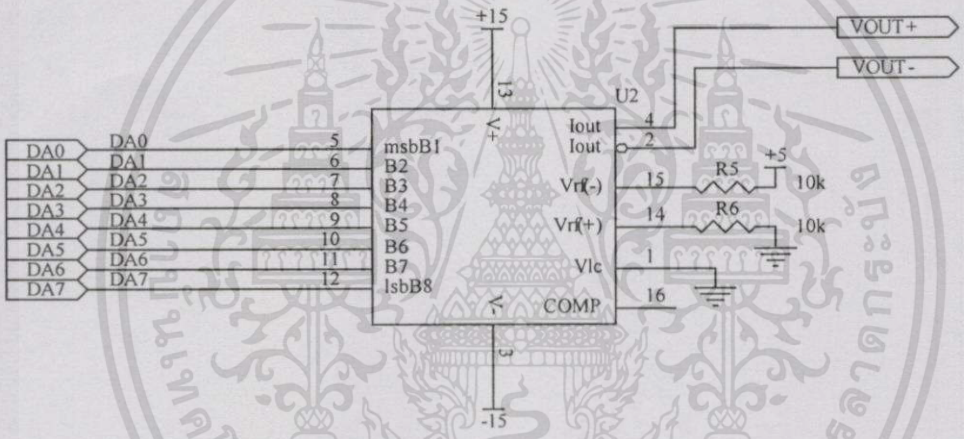
รูปที่ ง.1 วงจรกรองความถี่ต่ำผ่าน ความถี่ตัดที่ 3,400 เฮิรตซ์



รูปที่ ง.2 วงจรสำหรับดอนกิฤเรชั่น FPGA



รูปที่ ง.3 วงจรแปลงสัญญาณแอนะลอกเป็นดิจิตอล



รูปที่ ง.4 วงจรแปลงสัญญาณดิจิตอลเป็นแอนะลอก

ประวัติผู้เขียน

นายโกศล ตราชู เกิดเมื่อวันที่ 12 กันยายน 2515 ที่จังหวัดนครศรีธรรมราช สำเร็จการศึกษาครุศาสตรอุตสาหกรรมบัณฑิต(วิศวกรรมโทรคมนาคม) จากสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2536 และประกาศนียบัตรวิชาชีพชั้นสูง(อิเล็กทรอนิกส์) ปีการศึกษา 2534 จากวิทยาลัยเทคนิคนครศรีธรรมราช

ปี พ.ศ. 2537 เข้าทำงานในตำแหน่งอาจารย์ (ลูกจ้างชั่วคราว) สังกัดภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ตั้งแต่ ปี พ.ศ. 2540 ถึงปัจจุบัน รับราชการในตำแหน่งอาจารย์ สังกัดหน่วยงานเดิม ปี พ.ศ. 2540-2542 ได้รับทุนศึกษาต่อระดับปริญญาโทในประเทศจากทบวงมหาวิทยาลัย