

การเข้ารหัสลับโดยใช้ซินโดรมของบล็อกโค้ดเชิงเส้นและการสลับบิตโดย

อาศัยการกำเนิดแบบสุ่ม

**DATA SCRAMBLER USING SYNDROME OF LINEAR BLOCK  
CODE AND PSEUDO-RANDOM BIT ALLOCATION**



นายคเชนทร์ แจ่มกมล

Mr.KACHEN JAMKAMOL

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า

บัณฑิตวิทยาลัย

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

พ.ศ. 2539

ISBN 974-621-776-3

ลิขสิทธิ์ของบัณฑิตวิทยาลัย สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เลขหมู่.....

เลขทะเบียน..... 27275

วัน, เดือน, ปี 18 ส.ค. 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**DATA SCRAMBLER USING SYNDROME OF LINEAR BLOCK  
CODE AND PSEUDO-RANDOM BIT ALLOCATION**



**A THESIS SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENTS FOR THE DEGREE  
MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING  
SCHOOL OF GRADUATE STUDIES  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**1986**

**ISBN 974-621-776-3**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์

การเข้ารหัสลับโดยใช้จีนโดรมของบล็อกเค้ดเชิงเส้นและการ  
สลับบิทโดยอาศัยการกำเนิดแบบสุ่ม

นักศึกษา

นายคเชนทร์ แจ่มกมล

อาจารย์ผู้ควบคุมวิทยานิพนธ์

รศ.ดร.พุทศักดิ์ ชิวสุวิทย์

ระดับการศึกษา

วิศวกรรมศาสตรมหาบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

ภาควิชา

วิศวกรรมเทคโนโลยีการวัดคุมทางอุตสาหกรรม

พ.ศ.

2539

บทคัดย่อ

การสื่อสารข้อมูลเป็นที่ยอมรับกันทั่วไปว่ามีความสำคัญต่อชีวิตประจำวันมาก แนวโน้ม  
ของการแลกเปลี่ยนข้อมูลส่วนตัวมักจะส่งผ่านโครงข่ายของการสื่อสาร จุดนี้เองเป็นสาเหตุที่จะทำ  
ให้เกิดการรั่วไหลของข้อมูล เนื่องจากทำการจารกรรมข้อมูลซึ่งเป็นความเสี่ยงต่อความปลอดภัย  
ของข้อมูลและการละเมิดสิทธิส่วนบุคคล ดังนั้นในวิทยานิพนธ์นี้ได้เสนอเทคนิคการเข้ารหัสลับ  
ข้อมูลก่อนส่งออกไปในตัวกลางการสื่อสาร เพื่อป้องกันมิให้ข่าวสารรั่วไหลหรือถูกจารกรรม โดย  
เทคนิคการเข้ารหัสลับนี้ประกอบด้วยการใช้จีนโดรมของบล็อกเค้ดเชิงเส้น และการ  
กำเนิดแบบสุ่ม ผลของการเข้ารหัสนี้ได้แสดงให้เห็นอย่างชัดเจนว่าสามารถปกปิดข้อมูลได้อย่าง  
สมบูรณ์

**Thesis Title**                    **Data scrambler using syndrome of linear block code and pseudo-random bit allocation**

**Student**                         **Mr. Kachen Jamkamol**

**Thesis advisor**                **Assoc.Prof.Dr.Fusak Cheevasuvit**

**Level of study**                 **Master of Engineering in Electrical Engineering**

**Department**                  **Industrial Instrumentation Technology King Mongkut's Institute of Technology Ladkrabang**

**Year**                              **1996**



### **Abstract**

**Data communication is accepted as an important part of our everyday life. The trend of proprietary data exchange is through the communication networks. It will be caused the loss of data information due to data trapping , which will be remained the leading risk to data security and privacy. Therefore, this paper proposes a technique data scrambler in order to prevent the secret data from trapping. The syndrome information of linear block code and the pseudo random bit allocation are applied to the proposed technique. The result of scrambled data shows clearly that the original data will be hidden completely.**

## กิตติกรรมประกาศ

ขอขอบพระคุณ รศ.ดร.ฟูศักดิ์ ชิวสุวิทย์ เป็นอย่างสูงที่ได้ให้การประสิทธิ์ประสาทวิชา ให้คำแนะนำในเรื่องต่างๆ และให้คำปรึกษาในเรื่องต่างๆแก่ผู้เขียน ขอขอบพระคุณคณะกรรมการ สอบทุกท่านที่ได้สละเวลาอันมีค่า ขอขอบพระคุณ คุณพ่อคุณแม่และขอบคุณเพื่อนๆ ที่ให้ความช่วยเหลือและให้กำลังใจในการทำวิทยานิพนธ์ และขอขอบคุณสำนักงานพัฒนาวิทยาศาสตร์และเทคโนโลยีแห่งชาติที่ได้มอบทุนการศึกษาในการทำวิจัยช่วยให้งานวิจัยสำเร็จไปได้ด้วยดี

คเชนทร์ แจ่มกมล



# สารบัญ

	หน้า
บทคัดย่อภาษาไทย.....	I
บทคัดย่อภาษาอังกฤษ.....	II
กิตติกรรมประกาศ.....	III
สารบัญ.....	IV
สารบัญตาราง.....	VI
สารบัญภาพ.....	VIII
บทที่	
1. บทนำ.....	1
ความเป็นมาและความสำคัญของปัญหา.....	1
วัตถุประสงค์ของการวิจัย.....	1
ขอบเขตของการวิจัย.....	2
2. หลักการเข้ารหัสและบล็อกโค้ดเชิงเส้น.....	3
วิธีการสร้างรหัสลับ.....	3
การคำนวณทางคณิตศาสตร์ของรหัสแบบไบนารี.....	5
การตรวจสอบพาริตี.....	6
ความสามารถในการตรวจแก้บิตที่ผิดในรหัสเชิงเส้น.....	7
บล็อกโค้ดเชิงเส้น.....	10
แมทริกซ์ตัวกำเนิด.....	10
แมทริกซ์ในการตรวจสอบพาริตี.....	12
ซินโดรม.....	14
3. การออกแบบตัวเข้ารหัสและตัวถอดรหัสลับ.....	19
การออกแบบตัวเข้ารหัสลับ.....	19
ส่วนของการเข้ารหัส.....	20

## สารบัญ(ต่อ)

บทที่	หน้า
ส่วนของรูปแบบพิศพลาด.....	21
ส่วนของการกำเนิดสัญญาณสู่่มเทียม.....	22
ส่วนของการสลับตำแหน่งบิท.....	23
การออกแบบตัวถอดรหัสลับ.....	23
ส่วนของการสลับตำแหน่งบิทกลับ.....	24
ส่วนของการคำนวณหาซึนโครม.....	25
4. การประยุกต์ใช้อุปกรณ์ประเภทโปรแกรมเมเบิลลอจิกดีไวซ์.....	29
พื้นฐานของโปรแกรมเมเบิลลอจิกดีไวซ์.....	29
การออกแบบฟังก์ชันลอจิกโดยใช้ PLD.....	31
ส่วนของการเข้ารหัส.....	31
ส่วนของรูปแบบพิศพลาด.....	34
ส่วนของการกำเนิดสัญญาณสู่่มเทียม.....	36
ส่วนของการสลับตำแหน่งบิท.....	38
ส่วนของการสลับตำแหน่งบิทกลับ.....	42
ส่วนของการคำนวณหาค่าซึนโครม.....	46
5. ผลการทดลอง.....	51
6. สรุปผลและแนวทางการพัฒนา.....	56
บรรณานุกรม.....	57
ภาคผนวก.....	58
ภาคผนวก ก.....	59
ภาคผนวก ข.....	60
ประวัติผู้เขียน.....	63

## สารบัญตาราง

ตารางที่	หน้า
1 รูปแบบการคำนวณทางคณิตศาสตร์ของรภัส.....	5
2 รหัสพาริตีแบบคี่.....	6
3 รหัสพาริตีแบบคู่.....	7
4 แสดงขึ้น โครมที่ได้จากรูปแบบรหัสที่ผิดไปหนึ่งบิต.....	17
5 รูปแบบผิดพลาด.....	22
6 รูปแบบการกำเนิดแบบสุ่ม.....	23
7 การสลับตำแหน่งบิต.....	24
8 การสลับตำแหน่งบิตกลับ.....	26



## สารบัญญภาพ

	หน้า
1 แสดงการเข้ารหัสเสียงแบบแอนะล็อก.....	3
2 แสดงการเข้ารหัสแอนะล็อกชนิดมีการประมวลสัญญาณแบบดิจิทัล.....	4
3 แสดงการเข้ารหัสแบบดิจิทัล.....	4
4 แสดงรูปแบบรหัสคำของบล็อกโค้ดเชิงเส้น.....	10
5 แสดงแผนภาพการเข้ารหัสลับ.....	19
6 แสดงเมทริกซ์ตัวกำเนิด.....	20
7 แสดงการหารหัสคำ.....	21
8 แสดงแผนภาพการถอดรหัสลับ.....	25
9 แสดงเมทริกซ์พาริตี.....	25
10 แสดงการหาค่าซินโดรม.....	27
11 แสดงโครงสร้างของ PROM ที่โปรแกรมทางอาร์เรย์ OR.....	30
12 แสดงโครงสร้างของ PAL ที่ทำการโปรแกรมในอาร์เรย์ AND และ OR.....	30
13 แสดงโครงสร้างของ PAL ที่ทำการโปรแกรมที่อาร์เรย์ AND.....	31
14 แสดงรูปแบบสมการของส่วนเข้ารหัส.....	32
15 แสดงข้อมูลของ GAL16V8 ในการนำมาใช้งานส่วนเข้ารหัส.....	33
16 แสดงรูปแบบสมการของส่วนรูปแบบผิดพลาด.....	34
17 แสดงข้อมูลของ PAL16L8 ในการนำมาใช้งานส่วนรูปแบบผิดพลาด.....	35
18 แสดงรูปแบบสมการของการกำเนิดแบบสุ่ม.....	36
19 แสดงข้อมูลของ PAL16R4 ในการนำมาใช้งานส่วนการกำเนิดแบบสุ่ม.....	37
20 แสดงการสลับบิตที่ตำแหน่ง 0000.....	38
21 แสดงรูปแบบสมการส่วนของการสลับตำแหน่งบิต 0-3.....	39
22 แสดงรูปแบบสมการส่วนของการสลับตำแหน่งบิต 4-7.....	39
23 แสดงข้อมูลของ GAL20V8 ในการนำมาใช้งานส่วนสลับตำแหน่งบิต 0-3.....	40

## สารบัญญภาพ (ต่อ)

หน้า

24	แสดงข้อมูลของ GAL20V8 ในการนำมาใช้งานส่วนสลับตำแหน่งบิท 4-7.....	41
25	แสดงรูปแบบสมการส่วนของการสลับตำแหน่งบิทกลับบิท 0-3.....	43
26	แสดงรูปแบบสมการส่วนของการสลับตำแหน่งบิทกลับบิท 4-7.....	43
27	แสดงข้อมูลของ GAL20V8 ในการนำมาใช้งานส่วนสลับตำแหน่งบิทกลับบิท 0-3.....	44
28	แสดงข้อมูลของ GAL20V8 ในการนำมาใช้งานส่วนสลับตำแหน่งบิทกลับบิท 4-7.....	45
29	แสดงรูปแบบของสมการส่วนหาซินโดรม 1 .....	47
30	แสดงรูปแบบของสมการส่วนหาซินโดรม 2 .....	47
31	แสดงข้อมูลของ GAL20V8 ในการนำมาใช้งานส่วนหาซินโดรม 1.....	48
32	แสดงข้อมูลของ GAL16V8 ในการนำมาใช้งานส่วนหาซินโดรม 2.....	49
33a	แสดงรูปแบบสัญญาณเสียงดนตรีต้นแบบ.....	52
33b	แสดงรูปแบบสัญญาณเสียงดนตรีที่ผ่านการเข้ารหัสลับ.....	52
33c	แสดงรูปแบบสัญญาณเสียงดนตรีที่ผ่านการถอดรหัสลับ.....	52
34a	รูปแบบของสัญญาณเสียงพูดต้นแบบ.....	53
34b	รูปแบบของสัญญาณเสียงพูดที่ผ่านการเข้ารหัส.....	53
34c	รูปแบบของสัญญาณเสียงพูดที่ผ่านการถอดรหัสลับ.....	54
35a	แสดงข้อมูลข่าวสารต้นแบบ.....	54
35b	แสดงข้อมูลที่ผ่านการเข้ารหัส.....	55
35	แสดงข้อมูลที่ผ่านการถอดรหัสลับ.....	55
36	แสดงวงจรรวมของแผงวงจรตัวเข้ารหัสลับ .....	61
37	แสดงวงจรรวมของแผงวงจรตัวถอดรหัสลับ.....	62

# บทที่ 1

## บทนำ

### ความเป็นมาและความสำคัญของปัญหา

เนื่องจากปัจจุบันการติดต่อสื่อสารไม่ว่าจะเป็นทางโทรศัพท์ โทรทัศน์ คอมพิวเตอร์ วิทยุ ได้เข้ามามีบทบาทสำคัญอย่างสูงในสังคมปัจจุบัน จะมีกลุ่มของผู้ใช้บริการการสื่อสารในรูปแบบต่าง ๆ สำหรับการดำเนินธุรกิจ การทหาร หรือข้อมูลส่วนตัว ผ่านตัวกลางการสื่อสารในรูปแบบต่าง ๆ โดยข่าวสารที่ใช้ในการติดต่อสื่อสารจะมีความสำคัญของข่าวสารต่าง ๆ ตามลำดับ ซึ่งความสำคัญของข่าวสารนั้นจะขึ้นอยู่กับผู้ส่งสารจะเป็นผู้ให้ความสำคัญของข่าวสารนั้น เช่น ข่าวสารทางการทหารหรือข่าวสารทางด้านธุรกิจจะมีความสำคัญมาก และจะมีความล่อแหลมในการถูกดักฟังหรือจารกรรมจากผู้ที่ต้องการทำลาย ทำให้ผู้ส่งสารต้องพยายามหาวิธีการในการป้องกันข้อมูลข่าวสารของตนเองให้รอดพ้นจากการถูกจารกรรม วิธีการหนึ่งที่ถูกนำมาใช้คือ การเข้ารหัสข้อมูล (Scramble) ก่อนทำการส่งผ่านในตัวกลางของการสื่อสารและทำการถอดรหัส (Descramble) ข้อมูลที่ได้รับกลับคืนมาจากผู้รับข่าวสารนั้น ๆ ซึ่งทำให้ข้อมูลข่าวสารรอดพ้นจากการถูกจารกรรมได้ ถ้าผู้ต้องการจารกรรมข้อมูลไม่สามารถทำการถอดรหัสข้อมูลเหล่านั้นออกมาได้ หรือถอดมาได้แต่ข่าวสารนั้นถูกนำมาใช้แล้วและไม่มีผลต่อการดำเนินการใด ๆ ในการได้รับข่าวสารนั้น ในการวิจัยนี้ได้นำเสนอวิธีการเข้ารหัส โดยใช้หลักการของบล็อกโค้ดเชิงเส้นมาใช้สำหรับการเข้ารหัสลับ และการถอดรหัสโดยอาศัยซินโดรมของบล็อกโค้ดเชิงเส้นมาใช้สำหรับการป้องกันข้อมูลข่าวสารให้ได้รับความปลอดภัย

### วัตถุประสงค์ของการวิจัย

1. เพื่อเป็นการศึกษาหาแนวทางในการออกแบบชุดตัวเข้ารหัสลับ และชุดตัวถอดรหัสลับ และนำมาทำการสร้างเป็นแผงวงจร สำหรับทำงานตามโครงสร้างที่ทำการออกแบบ

2. เพื่อเป็นแนวทางสำหรับการพัฒนาในการออกแบบวิธีการสร้าง การเข้ารหัสลับและวิธีการถอดรหัสลับ โดยเป็นแนวทางสำหรับผู้สนใจสร้างอุปกรณ์ทางด้านการป้องกันข้อมูลข่าวสารให้ปลอดภัย และเป็นการช่วยประหยัดเวลาในการเริ่มศึกษาทางด้านนี้

3. สามารถนำแผงวงจรตัวเข้ารหัสลับและตัวถอดรหัสลับที่ทำการออกแบบ ไปใช้งานสำหรับป้องกันให้ข้อมูลให้มีความปลอดภัย

### ขอบเขตของการวิจัย

ขอบเขตของการวิจัยจะเป็นการสร้างแผงวงจรการเข้ารหัสลับและแผงวงจรการถอดรหัสลับโดยอาศัยหลักการของบล็อกโค้ดเชิงเส้นและการหาซิงโครม ซึ่งเป็นหลักวิธีการในการนำมาใช้ในการออกแบบวิธีการเข้ารหัสลับและวิธีการถอดรหัสลับ และนำวิธีการที่ได้มาทำการออกแบบสร้างเป็นแผงวงจรชุดสร้างรหัสลับและแผงวงจรชุดถอดรหัสลับ สำหรับการนำมาใช้งาน สามารถแบ่งเนื้อหากล่าวแยกเป็นบท ๆ ได้ดังนี้

บทที่ 1 บทนำ กล่าวถึงความเป็นมาและความสำคัญของงานวิจัย วัตถุประสงค์ในการทำวิจัยและรายละเอียดเนื้อหาในบทต่าง ๆ

บทที่ 2 ระบบการเข้ารหัสลับและทฤษฎี ในรายละเอียดจะเป็นการกล่าวถึงพื้นฐานของการเข้ารหัสแบบต่าง ๆ และทฤษฎีของบล็อกโค้ดเชิงเส้น

บทที่ 3 หลักการของตัวเข้ารหัสลับและตัวถอดรหัสลับ ในรายละเอียดจะกล่าวถึงการนำวิธีการของบล็อกโค้ดเชิงเส้นมาใช้ในการออกแบบวิธีการเข้ารหัสลับและวิธีการถอดรหัสลับและขั้นตอนต่าง ๆ ของการเข้ารหัส

บทที่ 4 การนำอุปกรณ์ประเภท PLD มาใช้งาน ในรายละเอียดเป็นการกล่าวถึงการนำอุปกรณ์ประเภท PLD มาใช้สำหรับสร้างแผงวงจรตัวเข้ารหัสลับและตัวถอดรหัสลับ รวมถึงหลักการของอุปกรณ์ PLD

บทที่ 5 แสดงผลการทดลอง โดยทำการเข้ารหัสและถอดรหัสกับสัญญาณในรูปแบบแตกต่างกัน เช่น สัญญาณเสียงพูดเสียงดนตรีและเพิ่มข้อมูลจากคอมพิวเตอร์

บทที่ 6 บทสรุปผลและแนวทางการพัฒนา กล่าวสรุปเนื้อหาของงานวิจัยที่สามารถรวมถึงปัญหาที่ประสบในการทำวิจัย

## บทที่ 2

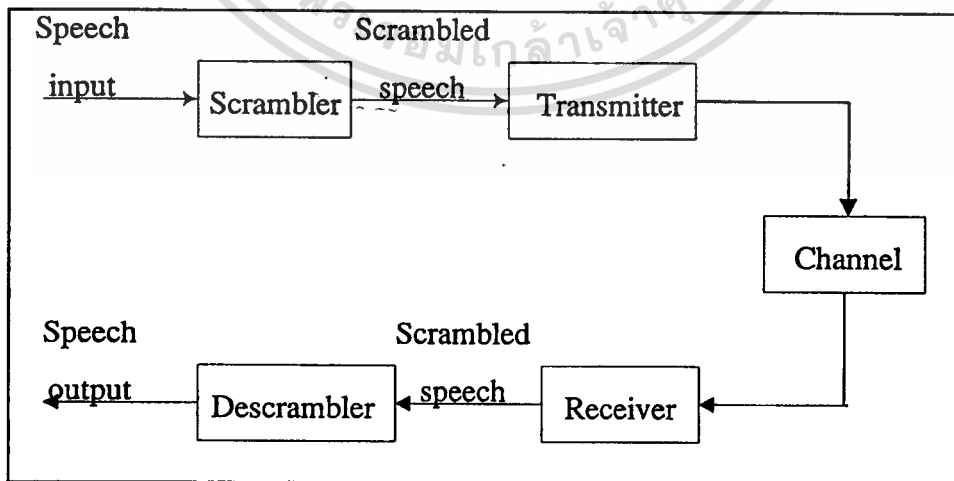
### หลักการเข้ารหัสและบล็อกโค้ดเชิงเส้น

#### วิธีการสร้างรหัสลับ

โดยทั่วไปมักมีการกล่าวอ้างถึงวิธีการเข้ารหัสลับ (Scramble) แบบพื้นฐาน 2 วิธี คือ แบบแอนะล็อก (Analog) และแบบดิจิทัล (Digital) ส่วนใหญ่เครื่องเข้ารหัสสัญญาณที่มีความซับซ้อนมากๆ มักใช้กระบวนการของการประมวลผลสัญญาณดิจิทัล (Digital signal processing) กระบวนการนี้ทำงานโดยการแปลงสัญญาณแอนะล็อกไปเป็นสัญญาณดิจิทัลก่อนแล้วจึงทำการเข้ารหัส ส่วนในระบบที่เป็นแบบแอนะล็อกโดยส่วนใหญ่จะมีความปลอดภัยค่อนข้างน้อย

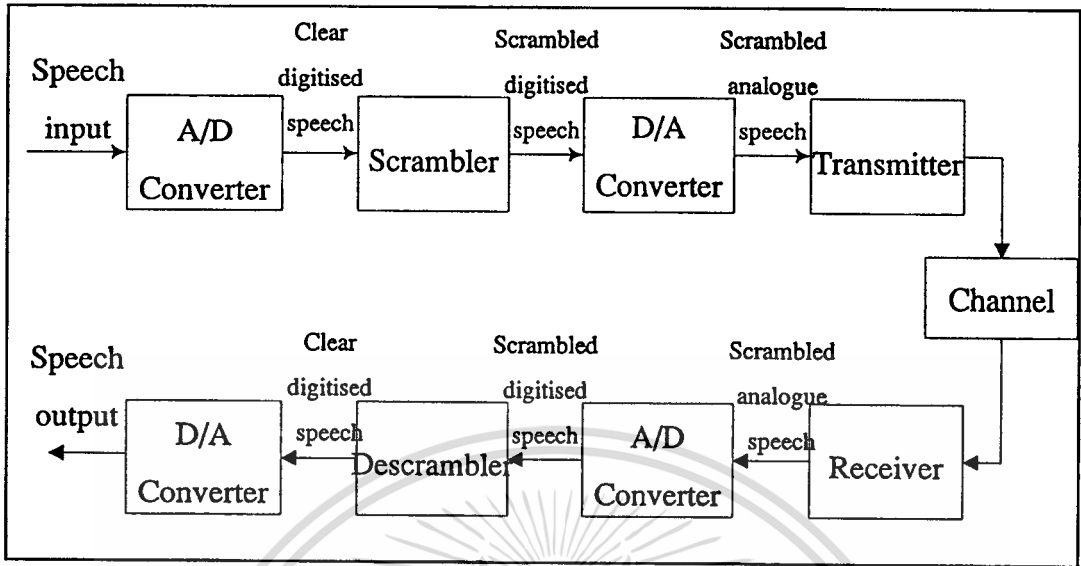
ข้อแตกต่างที่เห็นได้ชัดเจนระหว่างการเข้ารหัสแบบแอนะล็อกและแบบดิจิทัลคือ รูปแบบที่เกี่ยวกับตัวส่งผ่านซึ่งเป็นอุปกรณ์ที่ใช้ในการส่งสัญญาณที่เข้ารหัสแล้ว วัตถุประสงค์ของระบบที่มีการเข้ารหัสแบบแอนะล็อกคือ ส่งผ่านข่าวสารที่มีการเปลี่ยนแปลงอย่างต่อเนื่อง ในทางตรงข้ามระบบเข้ารหัสแบบดิจิทัลจะส่งผ่านข่าวสารด้วยสัญญาณที่สามารถนำข่าวสารไปได้เฉพาะตามจำนวนที่จำกัดไว้ ดังแสดงได้ในแผนภาพดังต่อไปนี้

ภาพที่ 1



แสดงการเข้ารหัสเสียงแบบแอนะล็อกชนิดไม่มีการประมวลผลสัญญาณดิจิทัล

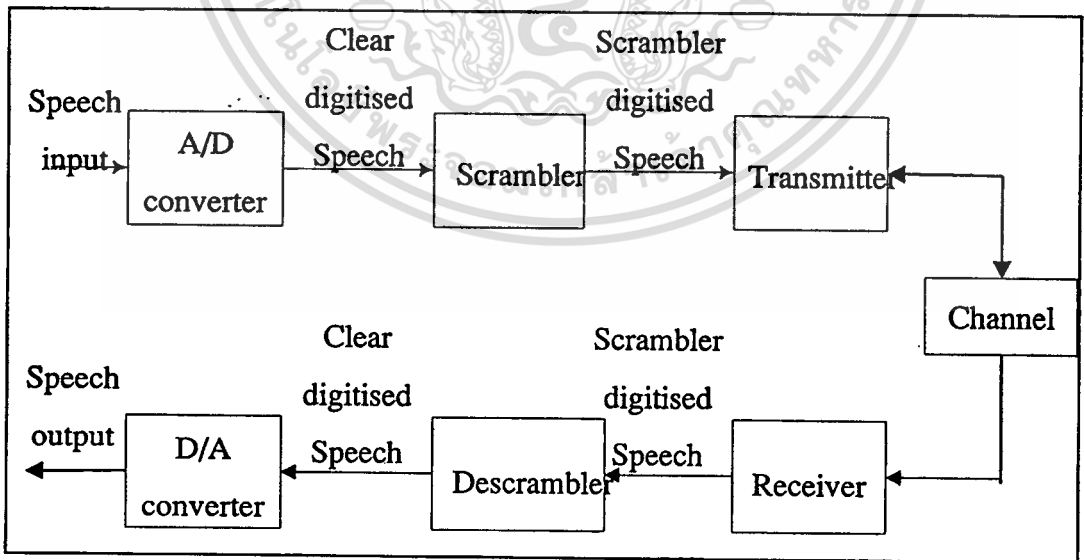
ภาพที่ 2



แสดงการเข้ารหัสเสียงแบบแอนะล็อกชนิดมีการประมวลผลสัญญาณแบบดิจิทัล

ในภาพที่ 1 และ 2 แสดงให้เห็นถึงการเข้ารหัสสัญญาณแบบแอนะล็อก ความแตกต่างระหว่างทั้ง 2 รูปแบบคือ รูปแบบของสัญญาณที่ผ่านการเข้ารหัสแล้ว เช่นในรูปภาพที่ 1 ยังคงมีสัญญาณแบบแอนะล็อกทั้งกระบวนการเข้ารหัส และในระบบตามภาพที่ 2 สัญญาณถูกเปลี่ยน

ภาพที่ 3



แสดงการเข้ารหัสเสียงแบบดิจิทัล

ไปอยู่ในรูปของสัญญาณดิจิทัลก่อนจะทำการเข้ารหัส แล้วถูกเปลี่ยนกลับให้อยู่ในรูปสัญญาณแอนะล็อกก่อนทำการส่งผ่านและหลังการส่งผ่านจะถูกเปลี่ยนกลับให้เป็นสัญญาณดิจิทัลอีกครั้งหนึ่ง แล้วจึงจะทำการถอดรหัสลับและท้ายสุดจะถูกเปลี่ยนให้เป็นสัญญาณแอนะล็อกอีกครั้งหนึ่ง

ในภาพที่ 3 แสดงให้เห็นถึงระบบดิจิทัลอีกประเภทหนึ่ง ความแตกต่างระหว่างระบบนี้และระบบในภาพที่ 2 คือ ไม่มีตัวแปลงสัญญาณจากดิจิทัลไปเป็นแอนะล็อก (D/A Converter) ในทันทีก่อนเข้าสู่เครื่องส่งผ่าน และไม่มีตัวแปลงสัญญาณจากแอนะล็อกไปเป็นดิจิทัลในทันทีหลังผ่านเครื่องรับ สำหรับในวิทยานิพนธ์นี้จะใช้รูปแบบดังแสดงในภาพที่ 3

### การคำนวณทางคณิตศาสตร์ของรหัสแบบไบนารี

สำหรับการเข้ารหัสในวิทยานิพนธ์ฉบับนี้ จะใช้คณิตศาสตร์ที่มีความซับซ้อนน้อย ทั้งนี้เพื่อให้สามารถนำมาสร้างวงจรทางฮาร์ดแวร์ได้ง่าย คณิตศาสตร์ที่ว่ามีรู้จักกันในชื่อของ Galois field [1] เนื่องจากคณิตศาสตร์ที่ใช้มีลักษณะเป็นลอจิก คือมีสัญลักษณ์เป็น 0 กับ 1 บางครั้งคณิตศาสตร์ดังกล่าวนี้ถูกเรียกว่า Binary field ซึ่งเขียนย่อเป็น  $GF(2)$  การทำงานของคณิตศาสตร์ดังกล่าวแสดงผลตามตารางที่ 1 โดยการบวกจะมีลักษณะเป็นการทำเอกซ์คลูซีฟออร์ (Exclusive OR) ของลอจิก ส่วนการคูณจะมีลักษณะเป็นการ AND ของลอจิก ในการเข้ารหัสนี้จะมีการนำเอาเวกเตอร์ของข้อมูลข่าวสาร (Message) คูณกับเมทริกซ์ที่ทำการออกแบบไว้ ผลลัพธ์ที่ได้จะเกิดจากการ AND และ EX-OR ของข้อมูลที่เป็นลอจิก จากเวกเตอร์ข้อมูลและเมทริกซ์ที่กำหนด

ตารางที่ 1

$0+0=0$	$0 \times 0=0$
$0+1=1$	$0 \times 1=0$
$1+0=1$	$1 \times 0=0$
$1+1=0$	$1 \times 1=1$

รูปแบบการคำนวณทางคณิตศาสตร์ของรหัส

## การตรวจสอบพาริตี ( Parity check)

เพื่อให้เข้าใจถึงการสร้างรหัสเชิงเส้น ให้ดูตัวอย่างง่ายๆ จากการสร้างรหัสคำ (Code word) เริ่มจากข้อมูลโดยปล่อยให้ผ่านไปในรหัสคำโดยตรง และจะมีบิตตามหลังแบบบิตเดียว ซึ่งเป็นการคำนวณจากบิตข้อมูลทั้งหมด มีวิธีการกำหนดบิตสุดท้ายที่ตามหลังมา 2 วิธี คือ

1. กำหนดบิตสุดท้ายเพื่อให้ผลรวมแบบโมดูลอ (Modulo) 2 ของบิตทั้งหมดในรหัสคำ มีค่าเท่ากับหนึ่ง
2. กำหนดบิตสุดท้ายเพื่อให้ผลรวมแบบโมดูลอ 2 ของบิตทั้งหมดในรหัสคำ มีค่าเท่ากับศูนย์

ในกรณีแรกรหัสคำมีพาริตีแบบคี่ (Odd parity) กล่าวคือ จำนวนบิตที่เป็นหนึ่งในรหัสคำ เป็นจำนวนคี่ ส่วนในกรณีที่สองมีจำนวนบิตที่เป็นหนึ่งในรหัสคำเป็นคู่ (Even parity) บิตที่เพิ่มขึ้นนอกเหนือจากบิตข้อมูลเรียกว่าพาริตีเช็ก (Parity check) และอาจเรียกได้ว่าเป็นการตรวจสอบแบบพาริตีคี่หรือพาริตีคู่

รหัสแบบพาริตีคี่และคู่จะแสดงในตารางที่ 2 และ 3 ตามลำดับ สำหรับกรณีที่มิบบิตข้อมูล 3 ตำแหน่ง จะสังเกตเห็นว่ารหัสของตารางที่ 2 ไม่มีแถวที่เป็นศูนย์ทั้งหมด ซึ่งต้องเป็นส่วนหนึ่งของรหัสที่เป็นเชิงเส้น

ตารางที่ 2

ข้อมูลข่าวสาร	ข้อมูลเข้ารหัส
000	0001
001	0010
010	0100
011	0111
100	1000
101	1011
110	1101
111	1110

รหัสพาริตีแบบคี่

ดังนั้นการตรวจสอบพาริตีแบบคี่ทำให้เกิดรหัสแบบไม่เป็นเชิงเส้น ในทางตรงกันข้าม รหัสที่ใช้พาริตีในตารางที่ 3 จะเป็นรหัสเชิงเส้น ระบบที่มีการผลิตการตรวจสอบพาริตีแบบคู่โดยเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเพิ่มบิตที่เป็นพาริตีนั้นได้จากการบวกแบบโมดูโล 2 ของบิตในรหัสของข้อมูลข่าวสาร ตัวอย่างเช่น รหัสข้อมูลข่าวสาร 101 เมื่อทำการบวกแบบโมดูโล 2 ของบิตที่มีค่าเป็น 1,0 และ 1 จะได้ผลลัพธ์คือ 0 ซึ่งจะเป็นค่าของบิตที่จะนำมาทำการต่อข้างท้ายของรหัสข้อมูลข่าวสาร กลายเป็น 1010

ตารางที่ 8

ข้อมูลข่าวสาร	ข้อมูลเข้ารหัส
000	0000
001	0011
010	0101
011	0110
100	1001
101	1010
110	1100
111	1111

รหัสพาริตีแบบคู่

ในการเพิ่มพาริตีอีกหนึ่งบิตให้รหัสของข้อมูลข่าวสารนั้นปกติมักจะนำมาตรวจสอบได้ถ้าบิตผิดไปเพียงบิตเดียวเท่านั้น แต่ไม่สามารถนำมาทำการแก้ไขบิตที่ผิดได้ ในการแก้ไขบิตที่ผิดไปจะต้องใช้วิธีการเข้ารหัสที่เพิ่มพาริตีบิตให้มากขึ้น อย่างเช่นการเข้ารหัสแบบบล็อกโค้ดเชิงเส้นที่จะได้กล่าวต่อไป

### ความสามารถในการตรวจแก้บิตที่ผิดในรหัสเชิงเส้น

ในส่วนนี้จะได้กล่าวถึงคำศัพท์พื้นฐานที่ใช้ในการแก้บิตที่ผิดของรหัสเชิงเส้น

เวท (Weight) ของแฮมมิงสำหรับเวกเตอร์  $v$   $n$ -ทิวเปิลส์ คือ  $\omega(v)$  ซึ่งหมายถึงผลรวมของจำนวนบิตของรหัสของ  $v$  ที่ไม่เป็นศูนย์ ตัวอย่างเช่น ถ้า  $v=[1\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 1]$  จะได้

$$\omega(v) = 5$$

ให้  $u$  และ  $v$  เป็นเวกเตอร์  $n$ -ทิวเปิลส์ ค่าระยะห่างระหว่าง  $u$  และ  $v$  เขียนได้เป็น  $d(u,v)$  ระยะห่างของสองเวกเตอร์ใด ๆ คือ จำนวนบิตรหัส "1" ที่แตกต่างกันของเวกเตอร์ทั้งสองเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned} u &= [1\ 0\ 0\ 1\ 0\ 1\ 1\ 0\ 0\ 0\ 1] \\ v &= [1\ 1\ 0\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1] \end{aligned}$$

$$\text{จะได้ } d(u, v) = 5$$

ถ้านำเวกเตอร์  $u$  และเวกเตอร์  $v$  มาบวกกัน โดยการบวกไบนารีเป็นการทำ EX-OR ดังนั้น

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

เวกเตอร์รวมที่ได้ถ้านำมาหาเวทของแฮมมิงจะได้ว่า  $\omega(u+v) = 5$

ดังนั้นพอสรุปได้ว่าระยะห่างแบบแฮมมิงระหว่างเวกเตอร์  $u$  และ  $v$  จะเท่ากับเวทของแฮมมิงจากเวกเตอร์รวม กล่าวคือ

$$d(u, v) = \omega(u+v) \quad (2.1)$$

สำหรับรหัสเชิงเส้นในการหาระยะห่างของแต่ละคู่ของรหัสคำ ระยะห่างที่น้อยที่สุดเขียนย่อเป็น  $d_{\min}$  ถ้า  $u$  และ  $v$  เป็นโค้ดเวกเตอร์สองชุดของรหัสเชิงเส้น โดย  $u+v$  ก็ยังเป็นโค้ดเวกเตอร์เพราะเซตของทุกโค้ดเวกเตอร์เป็นซับสเปซ (Subspace) ของทุก  $n$ -ทิวเปิ้ลส์ ดังนั้นจากคำนิยามที่ว่า ระยะห่างระหว่างโค้ดเวกเตอร์ทั้งสองคือ เวทของโค้ดเวกเตอร์ที่ 3 ก็จะได้ระยะห่างน้อยสุดของรหัสเชิงเส้นเท่ากับเวทต่ำสุดของโค้ดเวกเตอร์ที่ไม่เป็นศูนย์ ค่าระยะห่างน้อยสุด และเวทต่ำสุดจะเป็นตัวกำหนดความสามารถในการแก้รหัสบิตที่ผิดของรหัสเชิงเส้น[1]

พิจารณาจากรหัสที่ใช้ส่งโดยให้  $v = (v_1, v_2, \dots, v_n)$  เป็นโค้ดเวกเตอร์ที่ใช้ส่งและให้  $r = (r_1, r_2, \dots, r_n)$  เป็นเวกเตอร์ที่ได้รับจากการส่ง แต่เนื่องจากเวกเตอร์  $r$  ที่รับได้จะเป็นอะไรก็ได้ใน  $2^n$  เวกเตอร์ของ  $n$ -ทิวเปิ้ลส์ ความแตกต่างระหว่าง  $r$  และ  $v$  คือ  $e$

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

ซึ่ง  $e$  เป็นรูปแบบของรหัสที่ผิด (error pattern หรือ error vector) เมื่อ  $e_j = r_j + v_j = 1$  นั่นก็หมายความว่าโค้ดเวกเตอร์เกิดความผิดพลาดตำแหน่งบิตที่  $j^{\text{th}}$  แต่เนื่องจากในหนึ่งโค้ดเวกเตอร์มีรหัสอยู่  $n$  บิต จึงทำให้เกิดความผิดพลาด  $2^n -$  รูปแบบที่แตกต่างกัน ไม่นับรูปแบบที่มีทุกบิตเป็นศูนย์

ทางด้านรับตัวถอดรหัสมีหน้าที่ในการตรวจหาโค้ดเวกเตอร์ที่ส่งมาจากโค้ดเวกเตอร์  $r$  ที่รับได้ สำหรับการถอดรหัสโดยใช้วิธีแมกซ์อิมัมไลค์รีซูดนั้น ตัวถอดรหัสจะตรวจสอบว่า  $v$  เป็นเวกเตอร์ที่ใช้ในการส่ง ซึ่งจะมีค่าเข้าใกล้เวกเตอร์  $r$  โดยอาศัยการดูจากระยะห่างของแฮมมิง ตัวถอดรหัสสามารถทำการแก้ไขรหัสที่ผิดจำนวน  $t$  บิตในโค้ดเวกเตอร์ที่รับเข้ามา โดยที่  $2t + 2 \geq d_{\min} \geq 2t + 1$  ตัวถอดรหัสสามารถทำการแก้ทุกรูปแบบที่ผิดไป  $t$  บิต จากเวกเตอร์  $r$  ที่รับได้ ซึ่งแสดงให้เห็นได้ดังนี้ ให้  $v$  เป็นโค้ดเวกเตอร์ที่ต้องการส่งและ  $u$  เป็นโค้ดเวกเตอร์ใดๆ ระยะห่างของแฮมมิงระหว่าง  $u, v$  และ  $r$  จะต้องเป็นไปตามสมการข้างล่างนี้

$$d(v, r) + d(u, r) \geq d(u, v) \quad (2.2)$$

ถ้าสมมติว่าเกิดรหัสผิดไป  $t'$  บิต ( $t' \leq t$ ) ดังนั้นระยะห่างของแฮมมิงระหว่างโค้ดเวกเตอร์ที่ส่ง  $v$  กับโค้ดเวกเตอร์ที่รับ  $r$  คือ  $d(v, r) = t'$  แต่  $d(u, v) \geq d_{\min} \geq 2t + 1$  สมการที่ 2.16 จะให้

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

จากสมการที่ 2.3 แสดงให้เห็นว่ารูปแบบของรหัสที่มีบิตผิดไป  $t$  บิตหรือน้อยกว่า เวกเตอร์  $r$  ที่รับได้จะเข้าใกล้โค้ดเวกเตอร์  $v$  กว่าโค้ดเวกเตอร์  $u$  ดังนั้นตัวถอดรหัสจะสามารถแก้ไขรหัสผิดได้ถูกต้องตามความผิดพลาดของบิตที่ผิดไป ตัวถอดรหัสไม่สามารถจะแก้ทุกรูปแบบที่ผิดไป  $\ell$  บิต เมื่อ  $\ell \geq t + 1$  โดยปกติแล้วการแก้รหัสผิดของโค้ดเชิงเส้นจะทำได้เมื่อ

$$t = (d_{\min} - 1) / 2 \quad (2.4)$$

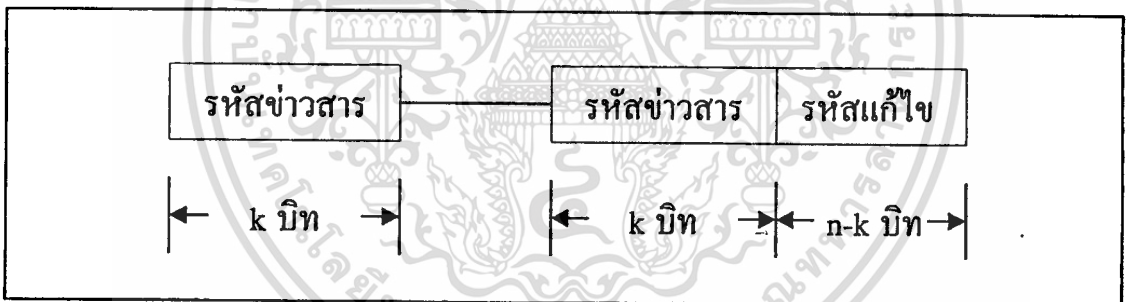
โดย  $t = (d_{\min} - 1) / 2$  เป็นค่าจำนวนเต็มไม่กิตกศนิยม และสามารถตรวจสอบรหัสผิดที่เกิดขึ้นถึง  $(d_{\min} - 1)$  บิตในแต่ละรหัสคำ[1]

## บล็อกโค้ดเชิงเส้น (Linear block code)

จากรหัสข่าวสาร (Message) ที่แต่ละบล็อกมีขนาด  $k$  บิต ซึ่งพบว่าจะให้ข่าวสารที่แตกต่างกันได้ถึง  $2^k - 1$  ข่าวสาร (ยกเว้นบล็อกที่มีรหัสข่าวสารเป็นศูนย์หมดจะไม่มีการนำมาใช้) แต่ละบล็อกข่าวสารจะถูกนำมาเข้ารหัสเป็นบล็อกขนาด  $n$  บิต โดยจะมี  $n-k$  บิต ที่เพิ่มเข้าไปให้รหัสข่าวสาร บิตเหล่านี้ที่เพิ่มเข้าไปในแต่ละบล็อกจะเป็นพริตตีหรือบางที่เรียกว่ารหัสแก้ไข ที่จะถูกนำมาใช้ในการตรวจสอบและตรวจสอบบิตที่ผิดไป และค่าของ  $n-k$  บิตจะขึ้นกับรหัสข่าวสารโดยตรง

บล็อกข่าวสารขนาด  $n$  บิตที่ได้จากการเข้ารหัสนี้จะเรียกว่ารหัสคำ (Code word) ถ้าหากว่ารหัสคำมีบิตของข่าวสารเดิมปรากฏอยู่ใน  $k$  บิตเริ่มต้นรหัสนั้น จะเรียกว่าซิสเต็มเมติกซ์ (Systematic code) ยิ่งไปกว่านั้นถ้าหากแต่ละรหัสคำจากจำนวน  $2^k$  รหัสคำที่เข้ารหัสไว้ เกิดจากการรวมกันของ  $k$  เวกเตอร์ของรหัสแบบอิสระเชิงเส้น (linearly independent) รหัสดังกล่าวจะเรียกว่ารหัสบล็อกโค้ดเชิงเส้น [11] รูปแบบของบล็อกโค้ดเชิงเส้นแสดงได้ดังภาพที่ 4

ภาพที่ 4



แสดงรูปรหัสคำของบล็อกโค้ดเชิงเส้น

### แมทริกซ์ตัวกำเนิด (Generator matrix)

สำหรับซับสเปซ  $S$  ของ  $V_n$  และแต่ละ  $n$ -ทิวเปิ้ล (Tuples) ของ  $S$  เป็นการรวมแบบเชิงเส้นของ  $v_1, v_2, \dots, v_k$  กล่าวคือ

$$u = m_1 v_1 + m_2 v_2 + \dots + m_k v_k \quad (2.5)$$

เมื่อ  $m_i = 0$  สำหรับ  $i = 1, 2, \dots, k$  ซับสเปซนี้มีขนาด  $k$  มิติของ  $V_n$  ซึ่งประกอบด้วย  $2^k$  ของ  $n$ -ทิวเปิ้ล จากข้อกำหนดที่กล่าวมาซึ่งสามารถอธิบายถึงรหัสเชิงเส้นของ  $2^k$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสคำโดยเซตของ  $k$  โค้ดเวกเตอร์ที่เป็นข้อกำหนดอิสระเชิงเส้น ถ้าจัด  $k$  รหัสคำเป็นอิสระต่อกันได้เมทริกซ์  $k \times n$

$$G = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_k \end{bmatrix} = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \cdots & \vdots \\ v_{k1} & v_{k2} & \cdots & v_{kn} \end{bmatrix} \quad (2.6)$$

เมื่อ  $v_i = (v_{i1}, v_{i2}, v_{i3}, \dots, v_{in})$  สำหรับ  $i = 1, 2, \dots, k$  ให้  $m = (m_1, m_2, \dots, m_k)$  เป็นบล็อกของข่าวสาร รหัสคำจะได้จาก

$$\begin{aligned} u &= mG \\ &= (m_1, m_2, \dots, m_k) \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_k \end{bmatrix} \\ &= m_1 v_1 + m_2 v_2 + \dots + m_k v_k \end{aligned} \quad (2.7)$$

ดังนั้นรหัสที่สอดคล้องกับชุดข่าวสาร  $(m_1, m_2, \dots, m_k)$  เกิดจากการรวมแบบเชิงเส้นของแถวใน  $G$  กลุ่มแถวต่างๆ ของเมทริกซ์  $G$  จะเป็นตัวผลิตรหัสเชิงเส้น และเราเรียกเมทริกซ์  $G$  ว่าเมทริกซ์ตัวกำเนิดของรหัส รหัสเชิงเส้นที่กล่าวนี้เรียกว่ารหัส  $(n, k)$  โดยในแต่ละบล็อกจะมีข่าวสารอยู่  $k$  บิต ที่ถูกเข้ารหัสเป็นรหัสคำที่มีความยาวขนาด  $n$  บิต

ลักษณะของเมทริกซ์ตัวกำเนิดขนาด  $k \times n$  ที่ใช้สร้างรหัสเชิงเส้น  $(n, k)$  แสดงได้ดังสมการที่ (2.8)

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & p_{11} & p_{12} & \cdots & p_{1,n-k} \\ 0 & 1 & 0 & 0 & \cdots & 0 & p_{21} & p_{22} & \cdots & p_{2,n-k} \\ 0 & 0 & 1 & 0 & \cdots & 0 & p_{31} & p_{32} & \cdots & p_{3,n-k} \\ \vdots & & & & & & \vdots & & & \\ 0 & 0 & 0 & 0 & \cdots & 1 & p_{k1} & p_{k2} & \cdots & p_{k,n-k} \end{bmatrix} \quad (2.8)$$

โดย  $P_{ij} = 1$  หรือ  $0$  ให้  $I_k$  เป็นเมทริกซ์เอกลักษณ์ (Identity matrix) ขนาด  $k \times k$  และให้  $P$  เป็นเมทริกซ์ขนาด  $k \times (n-k)$  ที่มีลิเมนต์เป็น  $P_{ij}$  ดังนั้นเมทริกซ์ตัวกำเนิดของรหัสระบบเขียนใหม่ได้เป็น

$$G = [I_k : P]$$

พิจารณาถึงบล็อกของข่าวสาร  $m = (m_1, m_2, \dots, m_k)$  เมื่อได้เมทริกซ์ตัวกำเนิดของสมการ (2.8) จะได้โค้ดเวกเตอร์เป็น

$$\begin{aligned} u &= (u_1, u_2, u_3, \dots, u_n) \\ &= (m_1, m_2, \dots, m_k)G \\ &= (m_1, m_2, \dots, m_k) \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 & p_{11} & p_{12} & \dots & p_{1k} \\ 0 & 1 & 0 & 0 & \dots & 0 & p_{21} & p_{22} & \dots & p_{2k} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & p_{k1} & p_{k2} & \dots & p_{k,n-k} \end{bmatrix} \end{aligned} \quad (2.9)$$

จากการคูณของเมทริกซ์จะได้

$$u_i = m_i \quad \text{สำหรับ } i = 1, 2, \dots, k \quad (2.10a)$$

และ

$$u_{k+j} = p_{1j}m_1 + p_{2j}m_2 + \dots + p_{kj}m_k \quad (2.10b)$$

สำหรับ  $j = 1, 2, \dots, n-k$  จากสมการที่ 2.10a และ 2.10b จะพบว่ารหัส  $k$  บิตแรกของรหัสคำ คือ รหัสของข่าวสาร ส่วน  $(n-k)$  บิตหลังเป็นฟังก์ชันเชิงเส้นของรหัสข่าวสารซึ่งเรียกว่ารหัสแก้ไข  $(n-k)$  บิตของ  $u$  หรือ รหัสตรวจสอบพาริตี (Parity check digits) ของรหัสคำ สมการที่ 2.10b เรียกว่าสมการพาริตีของรหัส

**เมทริกซ์ในการตรวจสอบพาริตี (Parity check matrix) [8]**

จากที่กล่าวว่าเมทริกซ์  $G$  ขนาด  $k \times n$  จะมีเมทริกซ์  $H$  ขนาด  $(n-k) \times n$  ซึ่งไร้วัดของ  $G$  จะตั้งฉากอยู่กับ  $H$  อินเนอร์โปรดักต์ของเวกเตอร์ในไร้วัดของ  $G$  กับแถวของ  $H$  จะเป็นศูนย์

$$H = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{n-k} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ h_{n-k,1} & h_{n-k,2} & \cdots & h_{n-k,n} \end{bmatrix} \quad (2.11)$$

และให้  $u = (u_1, u_2, \dots, u_n)$  เป็นเวกเตอร์ในโรว์สเปซของ  $G$  จะได้

$$uH^T = (0 \ 0 \ \cdots \ 0) \quad (2.12)$$

หรือ

$$u \cdot h_j = u_1 h_{j1} + u_2 h_{j2} + \cdots + u_n h_{jn} = 0 \quad (2.13)$$

สำหรับ  $u = (u_1, u_2, \dots, u_n)$  จึงสรุปได้ว่า  $u$  จะเป็นรหัสคำที่ได้จาก  $G$  ถ้าเพียงแต่  $uH^T = 0$  เมทริกซ์  $H$  นี้เรียกว่าเมทริกซ์ในการตรวจสอบพาริตี หรือเรียกย่อๆ ว่าพาริตีเมทริกซ์ ถ้าเมทริกซ์ตัวกำเนิดของรหัสได้มาจากสมการที่ 2.8 พาริตีเมทริกซ์ของรหัสคือ

$$H = \begin{bmatrix} p_{11} & p_{21} & \cdots & p_{k1} & 1 & 0 & 0 & \cdots & 0 \\ p_{12} & p_{22} & \cdots & p_{k2} & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ p_{1,n-k} & p_{2,n-k} & \cdots & p_{k,n-k} & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \quad (2.14)$$

$$= [P^T I_{n-k}]$$

$P^T$  เป็นทรานสโพส (Transpose) ของเมทริกซ์  $P$  สมการพาริตี 2.10b ได้จากเมทริกซ์  $H$  นั่นคือ  $u = (u_1, u_2, \dots, u_n)$  เป็นรหัสคำของรหัสข้อมูล  $m = (m_1, m_2, \dots, m_k)$  เมื่อ  $u_i = m_j$  สำหรับ  $i = 1, 2, \dots, k$  แต่

$$uH^T = 0$$

จะได้ว่า

$$u_{k+j} = p_{1j}u_1 + p_{2j}u_2 + \cdots + p_{kj}u_k$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$= p_{1j}m_1 + p_{2j}m_2 + \dots + p_{kj}m_k \quad (2.15)$$

เมื่อ  $j = 1, 2, \dots, n - k$  ซึ่งสมการข้างบนเป็นสมการเดียวกันกับสมการที่ 2.10b ในการออกแบบรหัสเชิงเส้นนั้นแมทริกซ์  $P$  จะถูกเลือกเพื่อให้มีคุณสมบัติในการแก้บิทที่ผิด

### ซินโดรม

บิตของรหัสคำขนาด  $n$  บิต เมื่อทำการส่งออกไปในตัวกลางจะเกิดสัญญาณรบกวนทำให้บางบิทของข้อมูลผิดไป ซึ่งรูปแบบของบิทที่ผิดไปบางครั้งเรียกว่าโคเซต (Coset) มีหลายรูปแบบถ้าหาก  $u$  เป็นรหัสคำที่ต้องการส่ง โดย  $u$  มีระยะห่างต่ำสุดตามเงื่อนไขสมการ (2.4) และ  $e_\ell$  เป็นรูปแบบของบิทที่ผิดไปในระหว่างการติดต่อสื่อสาร ทางด้านรับจะได้รับรหัสคำเป็น  $r$  กล่าวคือ

$$r = e_\ell + u \quad (2.16)$$

การคำนวณหาซินโดรมของรหัสคำที่รับได้ทำได้โดย

$$\begin{aligned} S &= uH^T \\ &= (e_\ell + u)H^T = e_\ell H^T + uH^T \end{aligned} \quad (2.17)$$

ถ้าหาก  $u$  เป็นรหัสคำที่ถูกต้องจะพบว่า

$$uH^T = 0 \quad (2.18)$$

ดังนั้นซินโดรมคือ  $S = e_\ell H^T$

โดยปกติแล้วแต่ละรูปแบบของบิทที่ผิดไปถ้าไม่ซ้ำกันจะให้ค่าซินโดรมที่ไม่เท่ากัน ในการพิสูจน์ทำได้โดยถ้าสมมติว่ารูปแบบของบิทที่ผิดไปคนละรูปแบบแต่ให้ซินโดรมเท่ากัน อย่างเช่นรูปแบบของบิทที่ผิดไป  $e_\ell$  กับ  $e_r$  เมื่อซินโดรมคือ

$$S_1 = e_\ell H^T$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$S_2 = e_r H^T$$

แต่  $S_1 = S_2$  จะให้

$$e_\ell H^T = e_r H^T \quad (2.19)$$

หรือ

$$(e_\ell - e_r) H^T = 0$$

เนื่องจาก  $H^T$  ไม่เป็นศูนย์ ดังนั้น  $e_\ell - e_r$  จะมีค่าเป็นศูนย์

$$e_\ell - e_r = 0$$

ผลต่างของ  $e_\ell$  กับ  $e_r$  จะเป็นศูนย์ก็ต่อเมื่อทุกบิตใน  $e_\ell$  กับ  $e_r$  จะเหมือนกันแบบบิตต่อบิตจึงสรุปได้ว่า

$$e_\ell = e_r$$

ซึ่งขัดกับสมมติฐานที่ว่า  $e_\ell$  กับ  $e_r$  เป็นคนละรูปแบบ

ดังนั้นพอสรุปได้ว่าถ้าหากรูปแบบบิตที่ผิดไปคนละรูปแบบจะให้ค่าซินโดรมที่แตกต่างกันออกไป กล่าวคือ

$$e_\ell H^T \neq e_r H^T \quad (2.20)$$

ถ้าหากรหัสคำ  $(n, k)$  ถูกนำมาคำนวณหาซินโดรม จะได้ซินโดรมขนาด  $n - k$  บิต ดังนั้นซินโดรมที่แตกต่างกันจะมีจำนวน  $2^{n-k}$  ค่า จะพบว่ารูปแบบของบิตที่ผิดไปหนึ่งรูปแบบกับซินโดรมที่สอดคล้องจะเป็นแบบหนึ่งต่อหนึ่ง ปกติแล้วทางด้านรับจะมีตารางของซินโดรมและรูปแบบของบิตที่ผิดที่สอดคล้องกันกับซินโดรมเก็บเอาไว้ ดังนั้นทางด้านรับจะมีขั้นตอนการทำงาน 4 ขั้นตอนกล่าวคือ

1. คำนวณซินโดรมของรหัสคำ  $r$  ที่ได้รับจาก  $S = rH^T$
2. เปิดตารางของซินโดรมเพื่อดึงเอารูปแบบที่ผิดที่ให้ซินโดรมเหมือนกับที่คำนวณได้ ถ้าหากเป็นรูปแบบของ  $e_\ell$
3. รหัสที่ถูกต้องคำนวณได้จาก  $u = r + e_\ell$
4. ดึง  $k$  บิตแรกจากรหัสคำของ  $u$  ซึ่งจะเป็นรหัสข่าวสาร  $m$  ที่ส่งมา ตัวอย่างเช่น ถ้าหากมีเมทริกซ์ตัวกำเนิด  $G$  เป็น

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

สามารถแปลงเป็นเมทริกซ์ตรวจสอบพาริตี  $H$  เป็น

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

ถ้าสมมติว่ารูปแบบรหัสที่ผิดไปเพียงหนึ่งบิต เป็น  $e_\ell = [0, 0, 0, 0, 0, 1]$  จะให้ซินโดรมเป็น

$$S = e_\ell H^T = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$$

สำหรับรูปแบบต่างๆของรหัสที่ผิดไปเพียงบิตเดียวกับซินโดรมที่สอดคล้องพอสรุปได้ดังตารางข้างล่าง

ตารางที่ 4

ซินโครม	รูปแบบรหัสที่ผิดไปเพียงบิตเดียว
001	000001
010	000010
100	000100
110	001000
101	010000
011	100000

แสดงซินโครมที่ได้จากรูปแบบรหัสที่ผิดไปหนึ่งบิต

ถ้าสมมติว่าทางด้านส่งต้องการส่งรหัสข่าวสาร  $m = [101]$  จะได้รับรหัสค่า  $u$

$$u = mG = [101] \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \\ = [1 \ 0 \ 1 \ 1 \ 0 \ 1]$$

เมื่อทำการส่งรหัสค่า  $u$  ผ่านช่องส่งสัญญาณที่มีการรบกวน จะพบว่าทางด้านรับได้รับรหัสค่า  $r$  เป็น  $[1 \ 0 \ 0 \ 1 \ 0 \ 1]$  ทางด้านรับจะทำการคำนวณหาค่าซินโครมจากรหัสค่า  $r$  ถ้าค่าซินโครมเป็นศูนย์หมดทุกบิตแสดงว่ารหัสค่า  $r$  ที่รับกับรหัสค่า  $u$  ที่ส่งเป็นรหัสค่าเดียวกัน แต่ถ้าหากค่าซินโครมไม่เป็นศูนย์ก็จะต้องทำการเปิดตารางที่ 4 เพื่อหารูปแบบรหัสที่ผิดไปที่สอดคล้องการคำนวณซินโครมทำได้โดย

$$S = rH^T = [1 \ 0 \ 0 \ 1 \ 0 \ 1] \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = [1 \ 1 \ 0]$$

จีนโดรมที่ได้จะนำไปตรวจสอบกับตาราง พบว่ารูปแบบของรหัสที่คิดไปคือ  $[0,0,1,0,0,0]$  ดังนั้นการแก้ไขรหัส  $r$  เพื่อให้ได้รหัสดำ  $u$  ที่ส่งมาทำได้โดย

$$\begin{aligned} u &= r + e = [1 \ 0 \ 0 \ 1 \ 0 \ 1] + [0 \ 0 \ 1 \ 0 \ 0 \ 0] \\ &= [1 \ 0 \ 1 \ 1 \ 0 \ 1] \end{aligned}$$

จากวิธีการดึงรหัสคำที่ถูกต้องกลับคืนมาจากรหัสคำที่ผิด จึงได้ถูกนำมาใช้เป็นหลักการเข้ารหัสลับให้กับข้อมูลในวิทยานิพนธ์ฉบับนี้ หลักการทำงานของกรเข้ารหัสลับคือ จากรหัสคำที่มีอยู่ จะถูกนำมาบวกกับรูปแบบรหัสที่ผิดต่าง ๆ ที่กำหนดไว้ โดยแต่ละรูปแบบของรหัสที่ผิดจะมีจีนโดรมที่สอดคล้องเก็บเป็นตารางข้อมูลเอาไว้ ซึ่งรหัสคำที่ผิดมีหลายรูปแบบ ดังนั้นการบวกรหัสคำกับรูปแบบรหัสที่ผิด จะทำการเลือกรูปแบบของรหัสที่ผิดแบบสุ่มเทียม (Pseudo random) วิธีการนี้ทางด้านรับจะมีรหัสคำที่ผิดอยู่ตลอดเวลา แต่จะสามารถนำเอารหัสคำที่ถูกต้องกลับคืนมาได้ถ้าหากรู้ว่าแมทริกซ์  $G$  คืออะไร ในการเพิ่มความซับซ้อนการเข้ารหัสลับ จะมีการเรียงลำดับตำแหน่งบิตในแต่ละรหัสคำที่ผ่านการบวกรูปแบบบิตที่ผิดไปเรียบร้อยแล้ว รูปแบบของการเรียงสลับบิตที่ผิดจะมีอยู่หลายรูปแบบเช่นกัน การเลือกรูปแบบการเรียงสลับบิตจะใช้ค่าของจีนโดรมเป็นตัวเลือกสำหรับรายละเอียดในการเข้ารหัสลับจะได้กล่าวในบทถัดไป

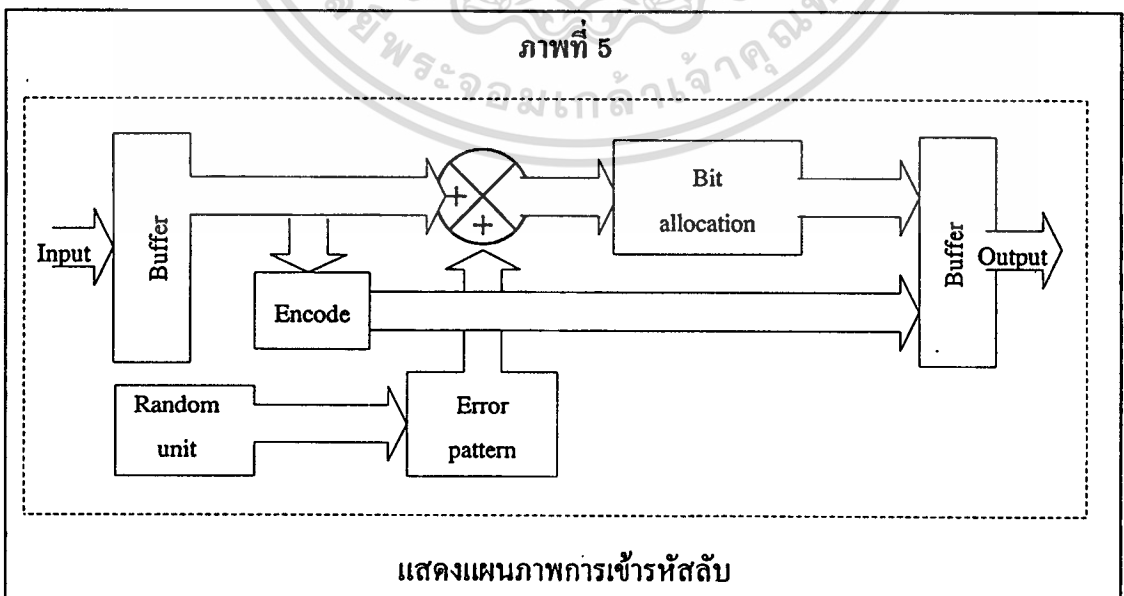
## บทที่ 3

### การออกแบบตัวเข้ารหัสและตัวถอดรหัสลับ

ในการออกแบบตัวเข้ารหัสและถอดรหัสลับแบบดิจิทัลนี้ สัญญาณที่จะนำมาป้อนเข้าทางอินพุตสำหรับการเข้ารหัสจะต้องเป็นสัญญาณที่อยู่ในรูปของสัญญาณดิจิทัล แต่ถ้าอินพุตที่ถูกป้อนเข้ามาอยู่ในรูปของสัญญาณแอนะล็อก จะต้องทำการแปลงสัญญาณเหล่านั้นให้อยู่ในรูปของสัญญาณดิจิทัล ก่อนที่จะทำการส่งเข้าทางด้านอินพุตของตัวเข้ารหัส ส่วนทางด้านตัวถอดรหัสลับก็จะได้รับข้อมูลที่ถูส่งผ่านเข้ามาทางอินพุตที่อยู่ในรูปสัญญาณดิจิทัลเช่นกัน และทำการถอดรหัสกลับแต่ถ้ารูปสัญญาณเดิมเป็นแบบแอนะล็อกก็จะต้องใช้ตัวแปลงสัญญาณดิจิทัลกลับเป็นสัญญาณแอนะล็อกกลับคืนมาเช่นเดิม ในส่วนนี้จะกล่าวถึงวิธีการเฉพาะส่วนของตัวเข้ารหัสลับและตัวถอดรหัสลับเท่านั้น

### การออกแบบตัวเข้ารหัสลับ

ภาพที่ 5



ในการออกแบบตัวสร้างรหัสลับจะเป็นการนำข่าวสารข้อมูลที่เข้ามาทางด้านอินพุตมาทำการแปลงให้มีรูปแบบของข้อมูลแตกต่างไปจากข้อมูลเดิม โดยทำการแปลงข้อมูลข่าวสารที่เข้ามาทางด้านอินพุตที่มีขนาด  $k$  บิต (ในที่นี้ใช้ 8 บิต) ให้เป็นรหัสคำ ที่มีขนาด  $n$  บิต (12 บิต) แล้วทำการบวกรหัสคำที่ได้กับรูปแบบผิดพลาดที่ทำการสร้างขึ้นจำนวน 15 รูปแบบหลังจากนั้นจะทำการสลับตำแหน่งของรหัสคำซึ่งมีรูปแบบการสลับตำแหน่ง 16 รูปแบบที่ทำการบวกรูปแบบผิดพลาดแล้ว ก็จะทำให้ได้รหัสข้อมูลลับที่ทำการเข้ารหัสแล้ว ซึ่งสามารถแบ่งเป็นส่วนๆ ของการสร้างรหัสลับดังแสดงในภาพที่ 5 ได้ดังนี้

### ส่วนของการเข้ารหัส (Encoder)

เป็นส่วนที่ทำหน้าที่ในการแปลงรหัสข้อมูลข่าวสารที่เข้ามาทางอินพุตที่มีขนาดของข้อมูล  $k$  บิต มาทำการแปลงให้เป็นรหัสคำ ในการแปลงนี้จะใช้วิธีการของรหัสบล็อกไค้ดเชิงเส้น โดยจะเป็นการนำรหัสข่าวสาร  $\underline{m}$  ที่มีขนาด  $k$  บิตไปทำการคูณกับเมทริกซ์ตัวกำเนิด  $G$  (Generator matrix) จะได้ผลลัพธ์เป็นรหัสคำที่มีขนาด  $n$  บิต จะมีจำนวนของบิตที่เพิ่มขึ้นมาจากรหัสของข้อมูลข่าวสาร  $(n - k)$  บิต (4 บิต) โดยรหัสที่เพิ่มขึ้นมานี้จะถูกนำมาใช้ในการแก้รหัสผิดซึ่งจะกล่าวในหัวข้อต่อไป

ภายในเมทริกซ์ตัวกำเนิด  $G$  จะประกอบไปด้วยเมทริกซ์เอกลักษณ์  $I_k$  ที่มีขนาด  $k \times k$  และเมทริกซ์พาริตี  $P$  ที่มีขนาด  $k \times (n - k)$  จะแสดงได้ในสมการที่ 3.1

$$G = [I_k : P] \quad (3.1)$$

เมทริกซ์ตัวกำเนิด  $G$  ที่ใช้แสดงในภาพที่ 6

ภาพที่ 6

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

แสดงเมทริกซ์ตัวกำเนิด

จากรหัสข้อมูลข่าวสาร  $\underline{m}$  ที่มีขนาด 8 บิต นำมาทำการคูณกับเมทริกซ์ตัวกำเนิด  $G$  จะได้รับรหัสคำมีขนาด 12 บิต ดังแสดงในสมการที่ 3.2

$$u = mG \quad (3.2)$$

ทำให้สามารถแสดงวิธีการสร้างรหัสคำ  $u$  ในภาพที่ 7

ภาพที่ 7

$$u = [m_0 \ m_1 \ m_2 \ m_3 \ m_4 \ m_5 \ m_6 \ m_7] \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

แสดงการสร้างรหัสคำ

#### ส่วนของรูปแบบผิดพลาด (Error Pattern)

สำหรับเมทริกซ์ตัวกำเนิดในภาพที่ 6 นั้นจะแก้รหัสผิดได้เพียงบิตเดียวแต่ในการเข้ารหัสลับนี้จะใช้รูปแบบของบิตที่ผิดไปสองบิต ใน 8 บิตแรกของรหัสคำซึ่งเป็นบิตของข่าวสาร ดังนั้นจึงไม่เป็นไปตามกฎของซินโดรม กล่าวคืออาจมีรูปแบบของบิตที่ผิดไป มากกว่าหนึ่งรูปแบบที่ให้ซินโดรมเหมือนกัน ดังแสดงในตารางที่ 5 เนื่องจากการเข้ารหัสบล็อกโค้ดเชิงเส้น (12,8) จะมีซินโดรมได้ 4 บิต หรือมีซินโดรมได้ 15 รูปแบบที่แตกต่างกัน (ไม่นับซินโดรมที่ทุกบิตเป็นศูนย์หมด) เนื่องจากรหัสที่ผิดไปสองบิตในแปดบิตแรก จะมีรูปแบบที่ไม่เหมือนกันได้เป็น 28 รูปแบบดังแสดงในคอลัมน์ที่สองของตารางที่ 5 ในกรณีที่รูปแบบบิตที่ผิดมีหลายรูปแบบจะเลือกเอารูปแบบบิตที่ผิดที่ให้ค่าสูงที่สุด อย่างเช่นรูปแบบ (3,5) ซึ่งมีบิตที่ผิดคือบิตที่ 3 กับ 5 จาก 8 บิตแรกของรหัสคำ กับรูปแบบ (2,7) ซึ่งมีบิตที่ผิดคือ บิตที่ 2 กับบิตที่ 7 ใน 8 บิตแรกของรหัสคำ จะพบว่ารูปแบบ (2,7) มีค่าสูงสุกว่า (3,5) จึงเลือกรูปแบบ (2,7) เนื่องจากการเข้ารหัสลับสัญญาณเสียง ถ้าเลือกบิตที่ผิดมีค่าสูงจะทำให้แอมพลิจูด (Amplitude) ของสัญญาณเสียงที่รวมกับรหัสที่ผิดแล้วเกิด

การเปลี่ยนแปลงสูงดังนั้น รูปแบบบิตที่ผิดที่ได้เลือกไว้ และซินโครมที่สอดคล้องพอสรุปได้ดัง ตารางที่ 5

$$r = u \oplus e \quad (3.3)$$

ตารางที่ 5

ค่าซินโครม	ตำแหน่งบิตที่ผิด	เลือกรูปแบบ	รูปแบบบิตที่ผิด
0001	(3,5) (2,7)	(2,7)	001000010000
0010	(1,6) (2,4)	(1,6)	010000100000
0011	(4,7)	(4,7)	000010010000
0100	(0,5) (1,7)	(0,5)	100001000000
0101	(0,3) (1,2) (4,6)	(4,6)	000010100000
0110	(6,7)	(6,7)	000000110000
0111	(1,4) (2,6)	(1,4)	010010000000
1000	(0,6) (3,4)	(3,4)	000110000000
1001	(4,5)	(4,5)	000011000000
1010	(0,1) (2,3) (5,7)	(5,7)	000001010000
1011	(2,5) (3,7)	(2,5)	001001000000
1100	(5,6)	(5,6)	000001100000
1101	(0,4) (3,6)	(0,4)	100010000000
1110	(0,7) (1,5)	(0,7)	100000010000
1111	(0,2) (1,3)	(1,3)	010100000000

รูปแบบผิดพลาด

### ส่วนของการกำเนิดสัญญาณสุ่มเทียม (Pseudo random sequence)

เป็นส่วนของตัวกำเนิดสัญญาณสุ่มเทียม มาทำหน้าที่เป็นตัวเลือกรูปแบบผิดพลาดที่จะมาทำการบวกแบบเอกคลูซีฟออร์กับรหัสคำที่ได้ โดยรูปแบบของสัญญาณแบบสุ่มเทียมใช้พหุนามลิโนเมียล (Primitive polynomial) อันดับสี่ คือ  $X^4 + X + 1$  ทำให้ได้รูปแบบของการเรียงลำดับเป็นดังตารางที่ 6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 6

ลำดับ	รูปแบบ	ลำดับ	รูปแบบ
$x^0$	0001	$x^8$	0101
$x^1$	0010	$x^9$	1010
$x^2$	0100	$x^{10}$	0111
$x^3$	1000	$x^{11}$	1110
$x^4$	0011	$x^{12}$	1111
$x^5$	0110	$x^{13}$	1101
$x^6$	1100	$x^{14}$	1001
$x^7$	1011		

รูปแบบการกำเนิดแบบสุ่ม

### ส่วนของการสลับตำแหน่งบิต (Bit allocation)

เป็นการสลับตำแหน่งของข้อมูลที่เป็นรหัสคำที่ทำการบวกกับรูปแบบผิดพลาดแล้วโดยจะทำการสลับเฉพาะ 8 บิตแรกเท่านั้น เพื่อเพิ่มความซับซ้อนในการป้องกันข้อมูลให้มีความปลอดภัยยิ่งขึ้น โดยการนำ 4 บิตหลังที่ได้จากรหัสคำมาเป็นตัวเลือกรูปแบบของการสลับตำแหน่งบิตข้อมูล ซึ่งมีรูปแบบการสลับบิต 16 รูปแบบดังแสดงในตารางที่ 7

### การออกแบบตัวถอดรหัสลับ

ในการออกแบบตัวถอดรหัสลับเป็นการนำข้อมูลข่าวสารที่ผ่านการเข้ารหัสลับแล้ว กลับคืนมาให้ได้ข้อมูลเดิม โดยทำการถอดรหัสข้อมูลข่าวสาร จากข่าวสารที่ทำการเข้ารหัสลับแล้วมีขนาด  $n$  บิต เข้ามาทางอินพุทโดยเริ่มจากการสลับตำแหน่งของข้อมูลกลับ ให้ข้อมูลที่ได้มีตำแหน่งเหมือนเดิม โดยใช้ข้อมูล 4 บิตที่เพิ่มเข้ามาเป็นตัวเลือกรูปแบบที่จะทำการสลับตำแหน่งบิตกลับ แล้วทำการคำนวณหาค่าซินโดรมที่จะมาเป็นตัวเลือกรูปแบบที่ผิดพลาดดังแสดงในตารางที่ 4 มาทำการบวกแบบเอกคลูซีฟออร์กับข้อมูลข่าวสารนั้น ทำให้ได้ข้อมูลข่าวสารเดิมกลับคืนมา แบ่งเป็นแต่ละขั้นตอนดังแสดงในภาพที่ 8

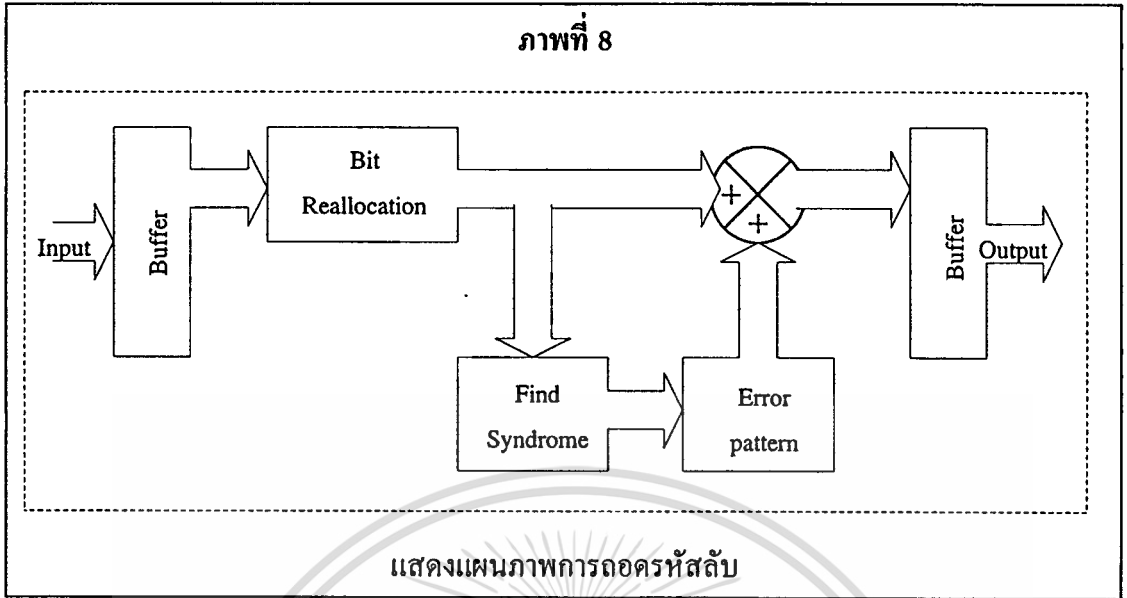
## ตารางที่ 7

ลำดับรูปแบบ	ตำแหน่งบิตที่ทำการสลับ							
	LSB						MSB	
0000	0	2	3	4	7	6	1	5
0001	2	4	6	1	0	3	7	5
0010	5	7	0	4	2	6	1	3
0011	6	4	0	3	7	5	1	2
0100	4	0	6	3	7	1	2	5
0101	5	1	6	2	4	3	7	0
0110	7	0	1	2	6	4	5	3
0111	6	2	1	0	7	3	4	5
1000	3	1	4	0	6	2	5	7
1001	2	5	4	1	6	3	0	7
1010	1	7	3	4	5	2	0	6
1011	5	1	6	2	7	0	3	4
1100	6	4	0	3	1	2	7	5
1101	7	0	1	2	3	4	5	6
1110	4	0	7	3	2	1	5	6
1111	5	0	4	1	2	3	7	6

### การสลับตำแหน่งบิต

#### ส่วนของการสลับตำแหน่งบิตกลับ (Bit reallocation)

เป็นส่วนที่ทำหน้าที่ในการสลับตำแหน่งของบิตข้อมูลที่เป็นรหัสคำที่ผิดไปกลับคืนมาให้ได้ตำแหน่งที่ถูกต้องตรงกับตำแหน่งเดิม โดยจะทำการสลับตำแหน่งข้อมูลกลับเฉพาะ 8 บิตแรกที่ถูกสลับไว้เท่านั้น และ 4 บิตที่เหลือจะเป็นตัวเลือกรูปแบบของการสลับตำแหน่งข้อมูลกลับ ซึ่งจะมีรูปแบบการสลับตำแหน่งบิตข้อมูลดังแสดงในตารางที่ 8



**ส่วนของการคำนวณหาซินโดรม**

ในส่วนของการคำนวณหาซินโดรมประกอบไปด้วยเมทริกซ์ตรวจสอบพาริตี H ที่ถูกนำมาใช้ในการคำนวณหาค่าซินโดรมสำหรับนำมาใช้ในการถอดรหัสที่ผิดไป โดยเมทริกซ์ตรวจสอบพาริตี H ประกอบขึ้นจากเมทริกซ์ทรานสโพสของเมทริกซ์พาริตี P ที่มีขนาด  $(n-k) \times k$  และเมทริกซ์เอกลักษณ์  $I_{n-k}$  คือ

$$H = [P^T I_{n-k}] \tag{3.4}$$

เมทริกซ์ตรวจสอบพาริตี H แสดงในภาพที่ 8

**ภาพที่ 9**

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

**แสดงเมทริกซ์พาริตี**

## ตารางที่ 8

ลำดับรูปแบบ	ตำแหน่งบิตที่ทำการสลับ							
	LSB							MSB
0000	0	6	1	2	3	7	5	4
0001	4	3	0	5	1	7	2	6
0010	2	6	4	7	3	0	5	1
0011	2	6	7	3	1	5	0	4
0100	1	5	6	3	0	7	2	4
0101	7	1	3	5	4	0	2	6
0110	1	2	3	7	5	6	4	0
0111	3	2	1	5	6	7	0	4
1000	3	1	5	0	2	6	4	7
1001	6	3	0	5	2	1	4	7
1010	6	0	5	2	3	4	7	1
1011	5	1	3	6	7	0	2	4
1100	2	4	5	3	1	7	0	6
1101	1	2	3	4	5	6	7	0
1110	1	5	4	3	0	6	7	2
1111	1	3	4	5	2	0	7	6

การสลับตำแหน่งบิตกลับ

ในการคำนวณหาซินโดรม  $S$  ได้จากการนำรหัสคำที่ได้รับมาทำการคูณกับเมทริกซ์ทรานสโพสของ  $H$  ซึ่ง  $H^T$  เป็นเมทริกซ์โดย 4 แถวสุดท้ายของ  $H^T$  จะเป็นเมทริกซ์เอกลักษณ์ โดยใน 8 แถวแรกได้จากการทำทรานสโพสเมทริกซ์  $P$  ของเมทริกซ์ตัวกำเนิด โดยทั้ง 8 แถวจะแตกต่างกัน และ  $r$  เป็นรหัสที่ได้รับเข้ามาซึ่งสามารถทำการคำนวณได้ดังแสดงในสมการที่ 3.5

$$S = rH^T \quad (3.5)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซินโดรมสามารถทำการหาได้ดังภาพที่ 10

ภาพที่ 10

$$S = \begin{bmatrix} r_0 & r_1 & \dots & r_{10} & r_{11} \end{bmatrix} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

แสดงการหาค่าซินโดรม

แต่เนื่องจากแต่ละแถว (Row) ของเมทริกซ์  $G$  และเมทริกซ์  $H$  ตั้งฉากกันจะให้ อินเนอร์โปรดัก (Inner product) เป็นศูนย์ ก็หมายความว่า  $uH^T = 0$  นั่นเอง ทำการแทนค่า  $r$  ด้วย สมการที่ 3.5 ได้ว่า

$$\begin{aligned} S &= (u \oplus e)H^T \\ S &= uH^T \oplus eH^T \\ S &= eH^T \end{aligned} \quad (3.6)$$

จากสมการที่ 3.5 และ 3.6 เป็นตัวบอกให้ทราบว่าค่าซินโดรมที่คำนวณได้จากรหัสคำที่ ผิดไปเนื่องจากรูปแบบผิดพลาด  $e$  ที่คูณกับ  $H^T$  นั้นให้ค่าซินโดรมเหมือนกันกับการนำรูปแบบผิดพลาด  $e$  ไปคูณโดยตรงกับ  $H^T$  ด้วยเหตุนี้ค่าของซินโดรมจึงเป็นตัวบ่งบอกว่ารหัส  $r$  ที่ได้รับมีรูปแบบผิดพลาด  $e$  รูปแบบใดปรากฏอยู่ ดังนั้นรูปแบบการเข้ารหัสคำที่ถูกส่ง  $u$  กลับคืนมาทำได้โดยการนำ  $e$  ไปทำการบวกเอกคลูซีฟออร์กัбрรหัส  $r$  อีกครั้งหนึ่ง ดังแสดงในสมการที่ 3.7 ซึ่งทำให้สามารถทำการถอดรหัสได้และได้ข้อมูลข่าวสารที่ถูกต้องกลับคืนมา

$$\begin{aligned}
 \underline{r} \oplus e &= (u \oplus e) \oplus e \\
 &= u \oplus (e \oplus e) \\
 &= u
 \end{aligned}
 \tag{3.7}$$



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 4

### การประยุกต์ใช้อุปกรณ์ประเภทโปรแกรมเมเบิลลอจิกดีไวซ์

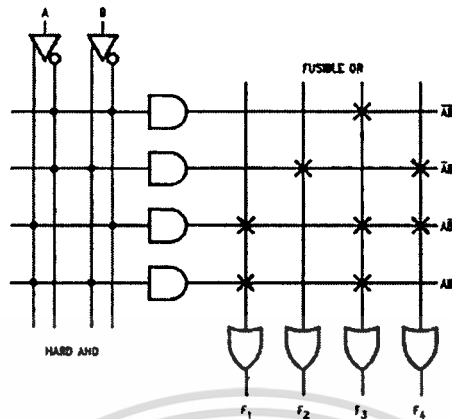
อุปกรณ์ประเภทโปรแกรมเมเบิลลอจิกดีไวซ์ (Programmable logic device) หรือ PLD ถูกผลิตขึ้นหลายบริษัทด้วยกันซึ่ง PLD เป็นอุปกรณ์ที่ถูกออกแบบมาใช้สำหรับทำงานตามสมการฟังก์ชันลอจิกที่ทำการออกแบบไว้ PLD แต่ละชนิดจะมีโครงสร้างภายในที่แตกต่างกัน ซึ่งเป็นสิ่งที่ผู้ออกแบบสมการฟังก์ชันลอจิกต้องพิจารณาสำหรับการเลือกอุปกรณ์ประเภทนี้มาใช้งาน

PLD มีข้อดีที่สามารถลดค่าใช้จ่ายในการออกแบบแผงวงจร เพราะ PLD แต่ละตัวสามารถออกแบบให้ทำงานตามสมการฟังก์ชันแตกต่างกันได้ โดยทำการโปรแกรมลงไปในเซลล์ (Cell) ภายในของตัวอุปกรณ์ PLD ด้วยการใช้เครื่องมือในการโปรแกรมเท่านั้นก็จะได้ PLD ที่ทำงานตามฟังก์ชันที่ผู้ออกแบบต้องการ

#### พื้นฐานของโปรแกรมเมเบิลลอจิกดีไวซ์

การนำ PLD มาใช้ในการทำงานตามฟังก์ชันที่ทำการออกแบบวงจรดิจิทัล เริ่มต้นจากการระเบิดฟิวส์ (Fuse) ที่อยู่ภายในตัว PLD ซึ่งเป็นเทคโนโลยีประเภทเดียวกันกับที่ใช้ในอุปกรณ์จำพวก Programmable read only memory (PROM) โดยโครงสร้างภายในของ (PROM) มีการตีโค้ดทางด้านอินพุต ในลักษณะเป็นเมทริกซ์ของเทอม AND แบบกำหนดรูปแบบตายตัว และสามารถทำการโปรแกรมลงไปได้ในเมทริกซ์เทอม OR ดังแสดงในภาพที่ 11 ซึ่งจะ เป็นข้อเสียของการใช้ PROM มาเป็นอุปกรณ์ทางด้านลอจิก เนื่องจากวงจรทางด้านลอจิกมีความหลากหลายมาก จำนวนของโปรดักต์เทอม (Product-term) ที่นำมาใช้คือ  $2^n$  เมื่อ  $n$  คือ จำนวนตัวแปรทางด้านอินพุต ซึ่งเป็นขนาดของเมทริกซ์ที่จะต้องมีความเป็นสองเท่าของ  $n$  เสมอ

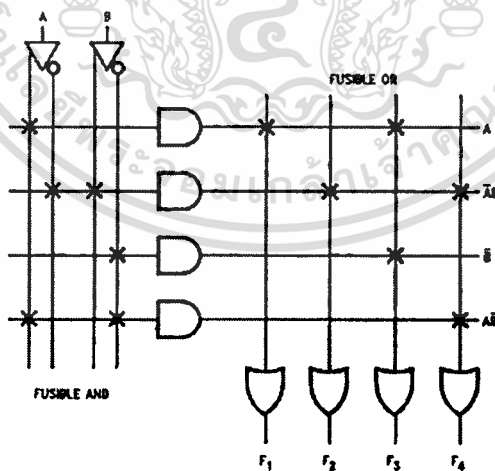
ภาพที่ 11



แสดงโครงสร้างของ PROM ที่โปรแกรมทางอาร์เรย์ OR

การแก้ไขทำได้โดยการใช้ฟิลด์แบบเมทริกซ์แทนที่การดีโค้ดแบบมีการกำหนดรูปแบบตายตัว ซึ่งจะยอมให้มีการเลือกโปรดัคเทอมที่ใช้ในการออกแบบ ทำให้การใช้เมทริกซ์แบบโปรแกรมเป็นไปอย่างมีประสิทธิภาพ เช่นเดียวกับ PROM ที่เกิดจากการใช้ฟิลด์ที่สามารถจัดรูปแบบในฟิลด์ (Field) ทำให้ถูกเรียกว่าฟิลด์โปรแกรมเมเบิลลอจิกอาร์เรย์ (Field Programmable logic array, FPAL หรือ PAL) ลักษณะโครงสร้างพื้นฐานภายในของ PAL ดังแสดงในภาพที่ 12 ในการใช้งานจะทำการ

ภาพที่ 12

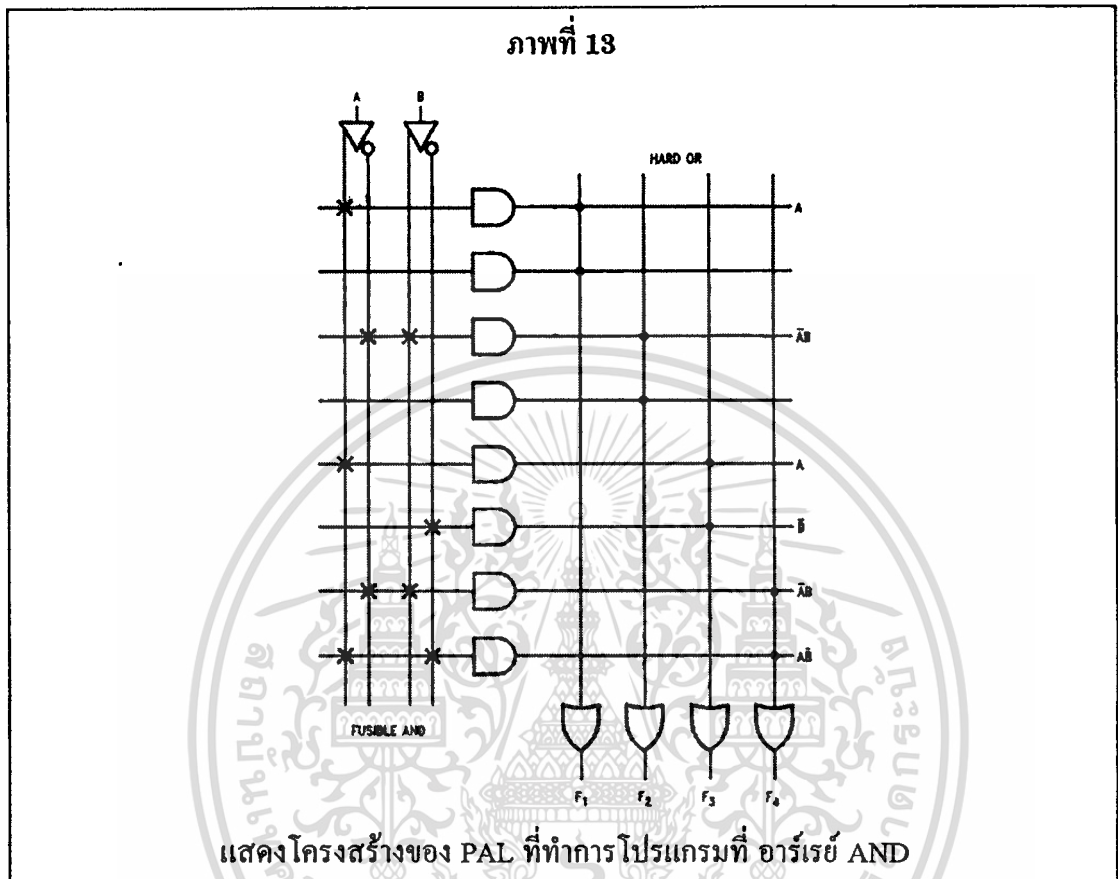


แสดงโครงสร้างของ PAL ที่ทำการโปรแกรมในอาร์เรย์ AND และ OR

โปรแกรมเฉพาะอาร์เรย์ของแอน (AND) เท่านั้น ดังแสดงในภาพที่ 13 ลักษณะของ PAL จะเป็นในลักษณะแบบ One-time programming (OTP) ซึ่งสามารถทำการโปรแกรมให้ PAL ทำงานตามฟังก์ชันที่ต้องการได้เพียงครั้งเดียว และไม่สามารถทำการลบหรือโปรแกรมใหม่ได้อีก ด้วยการพัฒนาทางเทคโนโลยีของการออกแบบเซลล์ (Cell) ทำให้สามารถผลิต PLD ที่สามารถทำการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมให้ทำงานตามฟังก์ชันลอจิกและสามารถทำการลบแล้วทำการโปรแกรมใหม่โดยเรียกว่า Generic array logic (GAL)



### การออกแบบฟังก์ชันลอจิกโดยใช้ PLD

เป็นการนำเอาอุปกรณ์ประเภท PLD มาใช้งานในส่วนต่างๆ สำหรับสร้างชุดตัวเข้ารหัสลับและตัวถอดรหัสลับเพื่อเป็นการประหยัดและสามารถลดขนาดของแผ่นวงจร ซึ่งจะทำการออกแบบสำหรับใช้งานในส่วนต่าง ๆ ดังนี้

#### ส่วนของการเข้ารหัส (Encoder)

เป็นการสร้างรหัสคำซึ่งจะเป็นไปตามสมการที่ 3.2 โดยที่  $m$  คือรหัสข้อมูลข่าวสารที่เข้ามาทางอินพุตมีขนาด 8 บิต ที่นำมาทำการคูณกับเมทริกซ์ตัวกำเนิด  $G$  ส่วน  $n$  คือรหัสคำที่ปรากฏทางเอาต์พุตซึ่งจะมีขนาด 12 บิต ทำให้สามารถเขียนเป็นสมการคูณกันของข้อมูลข่าวสารที่เข้ามา กับเมทริกซ์ตัวกำเนิดดังแสดงในสมการที่ 4.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\begin{aligned}
 u_0 &= m_0 & u_1 &= m_1 \\
 u_2 &= m_2 & u_3 &= m_3 \\
 u_4 &= m_4 & u_5 &= m_5 \\
 u_6 &= m_6 & u_7 &= m_7 \\
 u_8 &= m_1 \oplus m_2 \oplus m_4 \oplus m_6 \oplus m_7 \\
 u_9 &= m_2 \oplus m_3 \oplus m_4 \oplus m_5 \oplus m_7 \\
 u_{10} &= m_0 \oplus m_3 \oplus m_4 \oplus m_5 \oplus m_6 \\
 u_{11} &= m_0 \oplus m_1 \oplus m_5 \oplus m_6 \oplus m_7
 \end{aligned} \tag{4.1}$$

จากสมการที่ 4.1 จะเห็นได้ว่า  $u_0 - u_7$  มีค่าเท่ากับ  $m_0 - m_8$  และในส่วนของ  $u_8 - u_{11}$  จะถูกนำมาใช้ในการสร้างสมการลอจิก สำหรับการโปรแกรมดั่งแสดงในภาพที่ 14

ภาพที่ 14

```

ENCODE BLOCK
CHIP ENCODE GAL16V8
NC M1 M2 M3 M4 M5 M6 M7 M8 GND
NC E4 E3 EA4 EA3 EA2 EA1 E2 E1 VCC
EQUATIONS
EA1 = /M2*/M3*M5+/M2*M3*/M5+M2*/M3*/M5+M2*M3*M5
E1 = /EA1*/M7*M8+/EA1*/M7*/M8+EA1*/M7*/M8+EA1*/M7*M8

EA2 = /M3*/M4*M5+/M3*M4*/M5+M3*/M4*/M5+M3*M4*M5
E2 = /EA2*/M6*M8+/EA2*/M6*/M8+EA2*/M6*/M8+EA2*/M6*M8

EA3 = /M1*/M4*M5+/M1*M4*/M5+M1*/M4*/M5+M1*M4*M5
E3 = /EA3*/M6*M7+/EA3*/M6*/M7+EA3*/M6*/M7+EA3*/M6*M7

EA4 = /M1*/M2*M6+/M1*M2*/M6+M1*/M2*/M6+M1*M2*M6
E4 = /EA4*/M7*M8+/EA4*/M7*/M8+EA4*/M7*/M8+EA4*/M7*M8

```

แสดงรูปแบบสมการส่วนเข้ารหัส

ในการออกแบบได้ทำการเลือกใช้ GAL16V8 ซึ่งมีอินพุตสูงสุดถึง 8 อินพุต และมี 8 เอาท์พุต จากสมการลอจิกในภาพที่ 14 จะเห็นว่า EA1-EA4 เป็นฟังก์ชันที่ใช้ในการป้อนกลับเนื่องจาก GAL16V8 สามารถทำโปรแกรมโปรดักเทอม (Product-term) ได้สูงสุดเพียง 7 เทอมต่อหนึ่งเอาท์พุต ทำให้สามารถเขียนสมการในการโปรแกรมลงบน GAL16V8 ได้ดังภาพที่ 14 และเอาท์พุตที่ได้คือ E1-E4 เทียบได้กับ  $u_7 - u_{11}$  ของสมการที่ 4.1

## ภาพที่ 15

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)

Copyright (R) National Semiconductor Corporation 1990,1991

Document file for PALCODE.eqn

Device: 16V8

\$LABELS 20 NC M1 M2 M3 M4 M5 M6 M7 M8 GND NC E4 E3 EA4 EA3 EA2 EA1 E2 E1 VCC

Pin	Label	Type
1	NC	unused
2	M1	com input
3	M2	com input
4	M3	com input
5	M4	com input
6	M5	com input
7	M6	com input
8	M7	com input
9	M8	com input
10	GND	ground pin
11	NC	unused
12	E4	pos,trst,com output
13	E3	pos,trst,com output
14	EA4	pos,trst,com feedback
15	EA3	pos,trst,com feedback
16	EA2	pos,trst,com feedback
17	EA1	pos,trst,com feedback
18	E2	pos,trst,com output
19	E1	pos,trst,com output
20	VCC	power pin

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)

Copyright (R) National Semiconductor Corporation 1990,1991

Device Utilization:

No of dedicated inputs used : 8/10 (80.0%)

No of dedicated outputs used : 2/2 (100.0%)

No of feedbacks used as dedicated outputs : 2/6 (33.3%)

No of feedbacks used : 4/6 (66.7%)

Pin	Label	Terms Usage
19	E1	5/8 (62.5%)
18	E2	5/8 (62.5%)
17	EA1	5/8 (62.5%)
16	EA2	5/8 (62.5%)
15	EA3	5/8 (62.5%)
14	EA4	5/8 (62.5%)
13	E3	5/8 (62.5%)
12	E4	5/8 (62.5%)
Total		40/64 (62.5%)

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)

Copyright (R) National Semiconductor Corporation 1990,1991

Chip diagram (DIP)

NC	1	20	VCC
M1	2	19	E1
M2	3	18	E2
M3	4	17	EA1
M4	5	16	EA2
M5	6	15	EA3
M6	7	14	EA4
M7	8	13	E3
M8	9	12	E4
GND	10	11	NC

## แสดงข้อมูลของ GAL16V8 ในการนำมาใช้งานส่วนเข้ารหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทำการแปลงโปรแกรม (Compile) สมการที่แสดงในภาพที่ 14 ด้วยซอฟต์แวร์ OPAL จะได้ไฟล์ข้อมูลนามสกุล .LOG ซึ่งเป็นแฟ้มข้อมูลที่ใช้สำหรับเก็บรายละเอียดของการทำงานของ GAL16V8 และตำแหน่งของขาสำหรับการนำไปใช้งานในแผงวงจรที่ออกแบบ ดังปรากฏในภาพที่ 15

ในส่วนการนำสมการฟังก์ชันลอจิกที่ทำการออกแบบไว้มาทำการโปรแกรมลงบนเซลล์ของ GAL16V8 สำหรับให้ GAL16V8 ทำงานตามฟังก์ชันที่ออกแบบนั้น ต้องใช้เครื่องมือสำหรับทำการโปรแกรมโดยเฉพาะ โดยจะนำแฟ้มข้อมูล .JED ที่ได้จากการแปลงโปรแกรม แล้วทำการโปรแกรมลงบนเซลล์ของ GAL16V8 จะทำให้ได้ไอซี GAL16V8 ที่สามารถทำงานตามฟังก์ชันสำหรับทำหน้าที่ในการแปลงข้อมูลข่าวสารเป็นรหัสคำเพียงอย่างเดียว

### ส่วนของรูปแบบผิดพลาด (Error pattern)

เป็นการออกแบบโดยนำเอา PLD มาทำหน้าที่ในการสร้างรูปแบบที่ผิดพลาดที่สามารถทำการเลือกได้ซึ่งจะปรากฏทางเอาต์พุตที่มีขนาด 8 บิตและมีอินพุตขนาด 4 บิตสำหรับเป็นตัวเลือกรูปแบบดังแสดงในมตารางที่ 5 เป็นอินพุต สามารถนำมาเขียนเป็นสมการลอจิกสำหรับทำการโปรแกรมลงบนเซลล์ของ PLD ดังแสดงในภาพที่ 16

ภาพที่ 16

```

BIT ERROR
CHIP BIT_ERROR PAL16L8
A B C D NC NC NC NC NC GND
NC Y8 Y7 Y6 Y5 Y4 Y3 Y2 Y1 VCC
EQUATION
/Y1 =A*/C*/D+A*/B+A*/C*/D+A*/D+A*/C
/Y2 =/A*/B*/D+B*/C+A*/C+A*/B+B*/C*/D
/Y3 =B+/A*/C+A*/C+C*/D
/Y4 =B*/C+/A*/D+C*/D+A*/B*/D
/Y5 =C*/D+A*/C+B*/C*/D+/A*/B*/C*/D
/Y6 =/A*/D+/A*/C+B*/D+B*/C+A*/B*/C*/D
/Y7 =A*/B+C*/D+A*/D+A*/C+/A*/B*/D+A*/B*/C*/D
/Y8 =/A*/B*/C*/D+B*/C+A*/C+B*/D+A*/D

```

### แสดงรูปแบบสมการของส่วนรูปแบบผิดพลาด

จากสมการมีตัวแปร A,B,C,D เป็นอินพุตที่เป็นตัวเลือกรูปแบบผิดพลาดซึ่งมี A เป็นบิตสูง และ D เป็นบิตต่ำ ส่วนทางด้าน Y1-Y8 เป็นเอาต์พุตของรูปแบบผิดพลาดที่จะนำไปบวกแบบเอกคลูซิฟออร์กับข้อมูลข่าวสาร u มี Y1 เป็นบิตสูงและ Y8 เป็นบิตต่ำ ดังแสดงในภาพที่ 16 โดย

ทำการเลือก PLD เบอร์ PAL16L8 ซึ่งมี 8 อินพุตและมี 8 เอาต์พุตเป็นตัวทำหน้าที่ในการเก็บรูปแบบผิดพลาด

ภาพที่ 17

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
 Copyright (R) National Semiconductor Corporation 1990,1991

Document file for BITER.EQN  
 Device: 16L8

\$LABELS 20 A B C D NC NC NC NC NC NC GND NC Y8 Y7 Y6 Y5 Y4 Y3 Y2 Y1 VCC

Pin	Label	Type
1	A	com input
2	B	com input
3	C	com input
4	D	com input
5	NC	unused
6	NC	unused
7	NC	unused
8	NC	unused
9	NC	unused
10	GND	ground pin
11	NC	unused
12	Y8	neg.com output
13	Y7	neg.com output
14	Y6	neg.com output
15	Y5	neg.com output
16	Y4	neg.com output
17	Y3	neg.com output
18	Y2	neg.com output
19	Y1	neg.com output
20	VCC	power pin

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
 Copyright (R) National Semiconductor Corporation 1990,1991

Device Utilization:

No of dedicated inputs used : 4/10 (40.0%)  
 No of dedicated outputs used : 2/2 (100.0%)  
 No of feedbacks used as dedicated outputs : 6/6 (100.0%)

Pin	Label	Terms Usage
19	Y1	6/8 (75.0%)
18	Y2	6/8 (75.0%)
17	Y3	5/8 (62.5%)
16	Y4	5/8 (62.5%)
15	Y5	5/8 (62.5%)
14	Y6	6/8 (75.0%)
13	Y7	7/8 (87.5%)
12	Y8	6/8 (75.0%)
Total		46/64 (71.9%)

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
Copyright (R) National Semiconductor Corporation 1990,1991

Chip diagram (DIP)

A	1	20	VCC
B	2	19	Y1
C	3	18	Y2
D	4	17	Y3
NC	5	16	Y4
NC	6	15	Y5
NC	7	14	Y6
NC	8	13	Y7
NC	9	12	Y8
GND	10	11	NC

แสดงข้อมูลของ PAL16L8 ในการนำมาใช้งานส่วนของรูปแบบผิดพลาด

เมื่อนำสมการที่สร้างขึ้นในภาพที่ 16 มาทำการแปลงโปรแกรมจะได้เพิ่มข้อมูลแสดงข้อมูลการใช้งานและตำแหน่งของขาในการนำมาใช้งานดังแสดงในภาพที่ 17 และเพิ่มข้อมูลที่เป็นรายละเอียดของแต่ละฟิลต์ที่ทำการระเบิดฟิลต์อาร์เรย์ภายในของ PAL16L8 และใช้ไฟล์นี้ในการโปรแกรมสำหรับกำหนดให้ทำงานตามฟังก์ชันลอจิกที่ต้องการ

ส่วนของการกำเนิดสัญญาณแบบสุ่มเทียม (Pseudo Random )

ใช้พหุนามที่พหุนาม (Primitive Polynomial) อันดับสี่ พหุนามที่ได้อคือ  $X^4 + X + 1$  จะให้ลำดับของรูปแบบดังแสดงในตารางที่ 6

จากตารางที่ 6 รูปแบบของบิตสามารถนำมาเขียน ให้อยู่ในรูปแบบของซีควเอนซ์ (Sequence) ที่สามารถให้เอาท์พุทได้ตามตาราง สำหรับทำการเลือก PAL16R4 มาใช้ในการออกแบบสำหรับทำงานตามฟังก์ชัน โครงสร้างภายในของ PAL16R4 จะมีดีฟลิปฟลอป (D Flip-Flop) ภายใน 4 ตัว รูปแบบของฟังก์ชันที่ใช้ในการโปรแกรมให้ PAL16R4 ทำงานตามฟังก์ชันแสดงดังในภาพที่ 18

ภาพที่ 18

```
PSUDO
CHIP PSUDO PAL16R4
CLK NC NC NC NC NC NC NC NC NC GND
/EN NC NC Q4 Q3 Q2 Q1 NC NC VCC
EQUATIONS
/Q1 := Q1*/Q3*/Q4 + Q2*/Q3*/Q4 + Q3*Q4 + Q1*Q3*Q4 + Q2*Q3*Q4
/Q2 := /Q1
/Q3 := /Q2
/Q4 := /Q3
```

แสดงรูปแบบสมการของส่วนการกำเนิดแบบสุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่ 19

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
 Copyright (R) National Semiconductor Corporation 1990,1991  
 Document file for random.eqn  
 Device: 16R4

\$LABELS 20 CLK NC NC NC NC NC NC NC NC NC GND /EN NC NC Q4 Q3 Q2 Q1 NC NC VCC

Pin	Label	Type
1	CLK	clock pin
2	NC	unused
3	NC	unused
4	NC	unused
5	NC	unused
6	NC	unused
7	NC	unused
8	NC	unused
9	NC	unused
10	GND	ground pin
11	EN	enable pin
12	NC	unused
13	NC	unused
14	Q4	neg,reg feedback
15	Q3	neg,reg feedback
16	Q2	neg,reg feedback
17	Q1	neg,reg feedback
18	NC	unused
19	NC	unused
20	VCC	power pin

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
 Copyright (R) National Semiconductor Corporation 1990,1991  
 Device Utilization:  
 No of dedicated inputs used : 0/8 (0.0%)  
 No of feedbacks used : 4/8 (50.0%)

Pin	Label	Terms Usage
17	Q1	5/8 (62.5%)
16	Q2	1/8 (12.5%)
15	Q3	1/8 (12.5%)
14	Q4	1/8 (12.5%)
Total		8/64 (12.5%)

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
 Copyright (R) National Semiconductor Corporation 1990,1991

Chip diagram (DIP)

CLK	1	20   VCC
NC	2	19   NC
NC	3	18   NC
NC	4	17   Q1
NC	5	16   Q2
NC	6	15   Q3
NC	7	14   Q4
NC	8	13   NC
NC	9	12   NC
GND	10	11   /EN

แสดงข้อมูลของ PAL16R4 ในการนำมาใช้งานส่วนการกำเนิดแบบสุ่ม

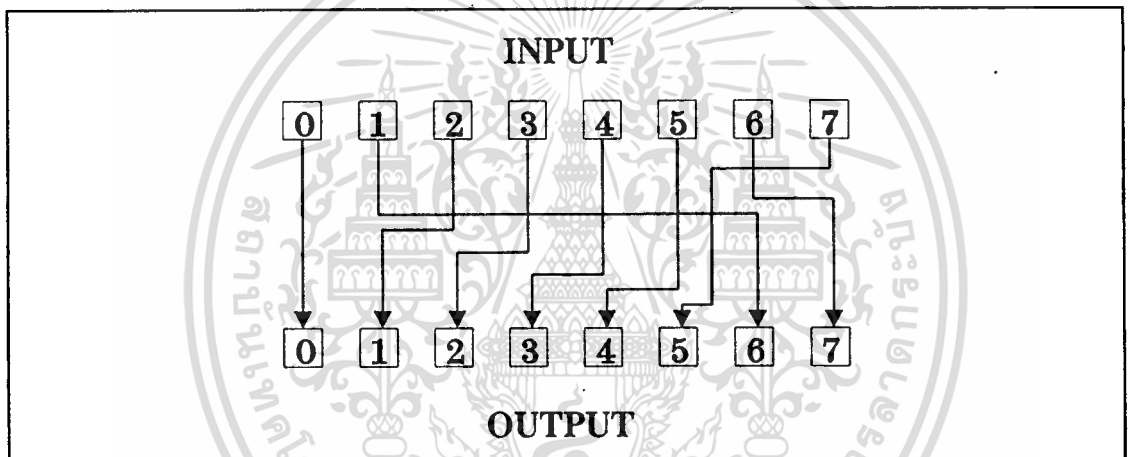
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การนำเสนอที่แสดงในภาพที่ 19 มาทำการแปลงโปรแกรมจะได้เพิ่มข้อมูลซึ่งจะเป็นไฟล์แสดงรายละเอียดในการใช้งาน PAL16R4 และแสดงตำแหน่งของขาสำหรับใช้งาน

#### ส่วนของการสลับตำแหน่งบิต (Bit allocation)

ในส่วนนี้เป็นการสลับตำแหน่งบิตของรหัสคำที่จะปรากฏทางเอาต์พุตตามรูปแบบที่ทำการกำหนดไว้ในตารางที่ 7 เช่นรูปแบบการสลับตำแหน่งของลำดับ 0000 เมื่อทำการสลับตำแหน่งจะแสดงได้ดังภาพที่ 20 ส่วนในลำดับอื่นๆ ก็เช่นเดียวกัน

ภาพที่ 20



แสดงการสลับบิตที่ตำแหน่ง 0000

ในการสลับตำแหน่งในตารางที่ 7 มีรูปแบบของการสลับอยู่ 16 รูปแบบที่ต้องทำการสลับ และแต่ละรูปแบบจะมีความแตกต่างกัน ข้อมูลอินพุตที่นำมาใช้ทั้ง 12 บิตนี้ มีเพียง 8 บิตแรกที่ถูกทำการสลับตำแหน่งส่วน 4 บิตหลังทำหน้าที่เลือกรูปแบบของการสลับตำแหน่งบิตข้อมูล 8 บิตแรก ซึ่งจะเป็นเอาต์พุต จากเงื่อนไขของอินพุตและเอาต์พุตนี้ทำให้พิจารณาเลือก GAL20V8 มาใช้งานซึ่ง GAL20V8 มีอินพุต 12 อินพุตและเอาต์พุต 8 เอาต์พุต และในการนำ GAL20V8 มาทำหน้าที่ในการสลับตำแหน่งบิต ต้องมีการใช้ GAL20V8 ถึงสองตัว เนื่องจากจากจำนวนของโปรดักเทอมที่ใช้มีมากกว่า 7 เทอมต่อหนึ่งเอาต์พุต ทำให้ต้องนำเอาวิธีการป้อนกลับมาใช้สำหรับการสร้างสมการที่ทำการโปรแกรมลงบน GAL20V8 ซึ่งต้องใช้การป้อนกลับถึง 4 เอาต์พุตของ GAL20V8 แต่ละตัว ฉะนั้นในการสลับตำแหน่งบิตจึงต้องใช้ GAL20V8 ถึงสองตัว โดยมีตัวแรกทำการสลับตำแหน่งบิตที่ 0-3 และตัวที่สองทำการสลับตำแหน่งบิตที่ 4-7 ทำให้สามารถเขียนเป็น

สมการสำหรับทำการโปรแกรมลง GAL20V8 ดังแสดงในรูปที่ 21 ซึ่งเป็นสมการของ GAL20V8 ตัวแรก และภาพที่ 22 เป็นการแสดงสมการของ GAL20V8 ตัวที่สอง

ภาพที่ 21

```

BIT ALLOCATION_A
CHIP ALLOCATION GAL20V8
NC A B C D E0 E1 E2 E3 E4 E5 GND
NC E6 Q4 Q3 SQ4 SQ3 SQ2 SQ1 Q2 Q1 E7 VCC
EQUATIONS

SQ1=/A*/B*/C*/D*E0+/B*/C*D*E2+/A*C*D*E6+/A*/B*C*/D*E5+/A*B*/C*/D*E4
+/A*B*/C*/D*E5+/A*B*C*/D*E7
Q1 =SQ1+A*B*/C*/D*E6+A*B*/C*D*E7+A*C*D*E5+A*B*C*/D*E4+A*/B*/C*/D*E5
+/A*/B*C*/D*E1

SQ2=/A*/B*/C*/D*E2+/A*/B*D*E4+/A*/B*C*/D*E7+/A*B*/C*/D*E0+/A*B*/C*/D*E1
+/A*B*C*/D*E2+B*C*/D*E0
Q2 =SQ2+A*B*/C*/D*E4+A*B*D*E0+A*/B*/C*/D*E1+A*/B*/C*/D*E5+A*/B*C*/D*E1
+/A*/B*C*/D*E7

SQ3=/A*/B*/C*/D*E3+/A*/B*/C*D*E6+/A*/B*C*E0+/A*B*/C*/D*E6+/A*B*/C*/D*E6
+/A*B*C*E1+A*B*/C*/D*E0
Q3 =SQ3+A*B*/C*/D*E1+A*B*C*D*E4+A*B*C*/D*E7+A*/B*/C*E4+A*/B*C*D*E6
+/A*/B*C*/D*E3

SQ4=/A*/B*/C*/D*E4+/A*/B*/C*D*E1+/A*/B*C*D*E3+B*/C*/D*E4+B*/C*/D*E3
+B*/C*/D*E2+/A*B*C*D*E0
Q4 =SQ4+A*B*/C*/D*E2+A*B*C*D*E1+A*B*C*/D*E3+A*/B*/C*/D*E0+A*/B*/C*/D*E1
+/A*/B*C*/D*E2

```

แสดงรูปแบบสมการส่วนของการสลับตำแหน่งบิต 0-3

ภาพที่ 22

```

BIT ALLOCATION_B
CHIP ALLOCATION GAL20V8
NC A B C D E0 E1 E2 E3 E4 E5 GND
NC E6 Q8 Q7 SQ8 SQ7 SQ6 SQ5 Q6 Q5 E7 VCC
EQUATIONS

SQ5=/A*/C*/D*E7+/A*/B*/C*D*E0+/A*C*D*E7+/A*/B*C*/D*E2+/A*B*/C*/D*E4
+/A*B*C*/D*E6+A*B*/C*/D*E1
Q5 =SQ5+A*B*/C*/D*E3+A*B*C*D*E2+A*B*C*/D*E2+A*/B*/C*E6+A*/B*C*D*E7
+/A*/B*C*/D*E5

SQ6=/A*/B*/D*E6+/A*/C*D*E3+/A*/B*C*D*E5+/A*B*/C*/D*E1+B*C*D*E3+/A*B*C*/D*E4
+A*B*/C*/D*E2
Q6 =SQ6+A*B*/C*/D*E4+A*B*C*/D*E1+A*/B*/C*/D*E2+A*/B*/C*/D*E3+A*/B*C*D*E0
+/A*/B*C*/D*E2

SQ7=/A*/B*/C*/D*E1+/A*/C*D*E7+/A*/B*C*E1+/A*B*/C*/D*E2+/A*B*C*D*E4+B*C*/D*E5
+A*B*/C*/D*E7
Q7 =SQ7+A*B*/C*/D*E5+A*B*C*D*E7+A*/B*/C*/D*E5+A*/B*/C*/D*E0+A*/B*C*D*E3
+/A*/B*C*/D*E0

SQ8=/A*/B*/C*/D*E5+/A*/B*/C*D*E5+/A*/B*C*D*E2+/A*/C*/D*E3+/A*B*/C*/D*E5
+/A*B*/C*/D*E0+/A*B*C*D*E5
Q8 =SQ8+A*B*/C*/D*E5+A*B*D*E6+A*/C*/D*E6+A*/B*/C*/D*E7+A*/B*/C*/D*E7
+/A*/B*C*/D*E4

```

แสดงรูปแบบสมการส่วนของการสลับตำแหน่งบิต 4-7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อนำเอาสมการในภาพที่ 21 และภาพที่ 22 มาทำการแปลงโปรแกรมจะได้เพิ่มข้อมูลที่แสดงการ GAL20V8 ในการทำงานตามฟังก์ชัน ซึ่งจะเก็บข้อมูลของการใช้งาน GAL20V8 และตำแหน่งของขาที่กำหนดการใช้งาน ดังแสดงในภาพที่ 23 ของ GAL20V8 ตัวแรกและภาพที่ 24 เป็นของ GAL20V8 ตัวที่สอง

ภาพที่ 24

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
 Copyright (R) National Semiconductor Corporation 1990,1991

Document file for MUXA.eqn  
 Device: 20V8

\$LABELS 24 NC A B C D E0 E1 E2 E3 E4 E5 GND G E6 Q4 SQ3 SQ2 SQ4 Q3 Q2 SQ1 Q1 E7 VCC

Pin	Label	Type
1	NC	unused
2	A	com input
3	B	com input
4	C	com input
5	D	com input
6	E0	com input
7	E1	com input
8	E2	com input
9	E3	com input
10	E4	com input
11	E5	com input
12	GND	ground pin
13	G	unused
14	E6	com input
15	Q4	pos,trst,com output
16	SQ3	pos,trst,com feedback
17	SQ2	pos,trst,com feedback
18	SQ4	pos,trst,com feedback
19	Q3	pos,trst,com output
20	Q2	pos,trst,com output
21	SQ1	pos,trst,com feedback
22	Q1	pos,trst,com output
23	E7	com input
24	VCC	power pin

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
 Copyright (R) National Semiconductor Corporation 1990,1991

Device Utilization:

No of dedicated inputs used : 12/14 (85.7%)  
 No of dedicated outputs used : 2/2 (100.0%)  
 No of feedbacks used as dedicated outputs : 2/6 (33.3%)  
 No of feedbacks used : 4/6 (66.7%)

Pin	Label	Terms Usage
22	Q1	8/8 (100.0%)
21	SQ1	8/8 (100.0%)
20	Q2	8/8 (100.0%)
19	Q3	8/8 (100.0%)
18	SQ4	8/8 (100.0%)
17	SQ2	8/8 (100.0%)
16	SQ3	8/8 (100.0%)
15	Q4	8/8 (100.0%)
<b>Total</b>		<b>64/64 (100.0%)</b>

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
 Copyright (R) National Semiconductor Corporation 1990,1991

Chip diagram (DIP)

NC		1		24		VCC
A		2		23		E7
B		3		22		Q1
C		4		21		SQ1
D		5		20		Q2
E0		6		19		Q3
E1		7		18		SQ4
E2		8		17		SQ2
E3		9		16		SQ3
E4		10		15		Q4
E5		11		14		E6
GND		12		13		G

แสดงข้อมูลของ GAL20V8 ในการนำมาใช้งานส่วนการสลับตำแหน่งบิต 0-3

ภาพที่ 24

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
 Copyright (R) National Semiconductor Corporation 1990,1991

Document file for MUXB.eqn  
 Device: 20V8

\$LABELS 24 NC A B C D E0 E1 E2 E3 E4 E5 GND NC E6 Q8 Q7 SQ8 SQ7 SQ6 SQ5 Q6  
 Q5 E7 VCC

Pin	Label	Type
1	NC	unused
2	A	com input
3	B	com input
4	C	com input
5	D	com input
6	E0	com input
7	E1	com input
8	E2	com input
9	E3	com input
10	E4	com input
11	E5	com input
12	GND	ground pin
13	NC	unused
14	E6	com input
15	Q8	pos,trst,com output
16	Q7	pos,trst,com output
17	SQ8	pos,trst,com feedback
18	SQ7	pos,trst,com feedback
19	SQ6	pos,trst,com feedback
20	SQ5	pos,trst,com feedback
21	Q6	pos,trst,com output
22	Q5	pos,trst,com output
23	E7	com input
24	VCC	power pin

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)

Copyright (R) National Semiconductor Corporation 1990,1991

Device Utilization:

No of dedicated inputs used : 12/14 (85.7%)  
 No of dedicated outputs used : 2/2 (100.0%)  
 No of feedbacks used as dedicated outputs : 2/6 (33.3%)  
 No of feedbacks used : 4/6 (66.7%)

Pin	Label	Terms	Usage
22	Q5	8/8	(100.0%)
21	Q6	8/8	(100.0%)
20	SQ5	8/8	(100.0%)
19	SQ6	8/8	(100.0%)
18	SQ7	8/8	(100.0%)
17	SQ8	8/8	(100.0%)
16	Q7	8/8	(100.0%)
15	Q8	8/8	(100.0%)
Total		64/64	(100.0%)

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)

Copyright (R) National Semiconductor Corporation 1990,1991

Chip diagram (DIP)

NC	1	24	VCC
A	2	23	E7
B	3	22	Q5
C	4	21	Q6
D	5	20	SQ5
E0	6	19	SQ6
E1	7	18	SQ7
E2	8	17	SQ8
E3	9	16	Q7
E4	10	15	Q8
E5	11	14	E6
GND	12	13	NC

แสดงข้อมูลของ GAL20V8 ในการนำมาใช้งานส่วนการสลับตำแหน่งบิต 4-7

### ส่วนของการสลับตำแหน่งบิตกลับ (Bit reallocation)

เป็นส่วนที่ทำหน้าที่ในการสลับตำแหน่งของบิตข้อมูลกลับ ทำงานในลักษณะตรงข้ามกับการสลับตำแหน่งบิตข้อมูลเพื่อให้ได้ข้อมูลที่มีตำแหน่งถูกต้องกลับคืนมา มีรูปแบบการสลับตำแหน่งบิตกลับตามตารางที่ 8

ในการเลือก PLD และการสร้างสมการลอจิกจากตารางการสลับบิตกลับ พิจารณาทางด้านอินพุตของการสลับบิตกลับจะมีอยู่ 12 อินพุต คือ มีข้อมูลที่ต้องการสลับตำแหน่ง 8 บิต และบิตที่เป็นตัวเลือกรูปแบบการสลับตำแหน่งบิตกลับ 4 บิต ส่วนเอาต์พุตที่ปรากฏจะมีขนาด 8 บิต เช่นเดียวกับการสลับตำแหน่งบิตทำให้พิจารณาเลือก GAL20V8 ซึ่งมีอินพุต 12 อินพุตและมีเอาต์พุต 8 เอาต์พุต แต่ต้องใช้ GAL20V8. ถึงสองตัวในการสลับบิตกลับเนื่องจากจำนวนของโปรดัคเทอม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของสมการที่ได้จากตารางที่ 8 มีจำนวนของเทอมมาก ทำให้ต้องใช้ GAL20V8 ถึงสองตัวโดยให้ตัวแรกทำการสลับตำแหน่งบิต 0-3 กลับ และตัวที่สองทำการสลับตำแหน่งบิต 4-7 กลับ และใช้เอาท์พุทที่เหลืออีก 4 เอาท์พุทของ GAL20V8 แต่ละตัวมาทำหน้าที่ในการป้อนกลับเพื่อให้สามารถใช้งานโปรดัคเทอมได้มากกว่า 7 เทอม ดังแสดงในภาพที่ 25 และภาพที่ 26

ภาพที่ 25

```

RE_BIT_ALLOCATION_A
CHIP RE_ALLOCATION GAL20V8
NC A B C D Q1 Q2 Q3 Q4 Q5 Q6 GND
NC Q7 D3 D2 SD3 SD2 SD1 SD0 D1 D0 Q8 VCC
EQUATIONS

SD0=/A*/B*/C*/D*Q1+/A*/B*/C*/D*Q5+/A*/B*/C*/Q3+/A*/B*/C*/D*Q2+/A*/B*/C*/D*Q8
+/A*/B*/C*/D*Q4+B*/C*/D*Q2
D0 =SD0+A*/B*/C*/D*Q3+A*/B*/D*Q2+A*/B*/C*/D*Q4+A*/B*/C*/D*Q7+A*/B*/C*/D*Q6
+A*/B*/C*/D*Q7
SD1=/A*/B*/C*/D*Q7+/B*/C*/D*Q4+/A*/B*/C*/Q7+/A*/B*/C*/D*Q6+/A*/B*/C*/D*Q2
+/A*/B*/C*/Q3+A*/B*/C*/D*Q5
D1 =SD1+A*/B*/C*/D*Q3+A*/B*/C*/D*Q4+A*/B*/C*/D*Q6+A*/B*/C*/D*Q2+A*/B*/C*/D*Q2
+A*/B*/C*/D*Q1
SD2=/A*/B*/C*/D*Q2+/B*/C*/D*Q1+/A*/B*/C*/D*Q8+/A*/B*/C*/D*Q5+/A*/B*/C*/D*Q7
+/A*/B*/C*/D*Q4+/A*/B*/C*/D*Q2
D2 =SD2+/A*/B*/C*/D*Q4+A*/B*/C*/D*Q6+A*/B*/C*/D*Q4+A*/B*/C*/Q5+A*/B*/C*/D*Q4
+A*/B*/C*/D*Q6
SD3=/A*/B*/C*/D*Q3+/A*/C*/D*Q6+/A*/B*/C*/D*Q4+/A*/C*/D*Q8+/A*/B*/C*/D*Q4
+B*/C*/D*Q6+A*/B*/C*/D*Q4
D3 =SD3+A*/B*/C*/D*Q5+A*/B*/C*/D*Q4+A*/B*/C*/D*Q1+A*/B*/C*/D*Q6+A*/B*/C*/D*Q7
+A*/B*/C*/D*Q3

```

แสดงรูปแบบของสมการส่วนของการสลับบิตกลับบิต 0-3

ภาพที่ 26

```

BIT_REALLOCATION_B
CHIP BIT_REALLOCATION GAL20V8
NC A B C D Q1 Q2 Q3 Q4 Q5 Q6 GND
NC Q7 D7 D6 SD7 SD6 SD5 SD4 D5 D4 Q8 VCC
EQUATIONS

SD4=/A*/B*/C*/D*Q4+/A*/B*/D*Q2+B*/C*/D*Q4+/A*/B*/C*/D*Q1+/A*/B*/C*/D*Q5
+/A*/B*/C*/D*Q7+/A*/B*/C*/D*Q6
D4 =SD4+A*/B*/C*/D*Q2+A*/B*/C*/D*Q6+A*/B*/C*/D*Q3+A*/B*/C*/D*Q1+A*/B*/C*/Q3
+A*/B*/C*/D*Q8
SD5=/A*/B*/C*/Q8+/A*/B*/C*/D*Q6+/A*/B*/C*/D*Q1+B*/C*/D*Q8+/A*/B*/C*/D*Q1
+/A*/B*/C*/D*Q8+B*/C*/D*Q7
D5 =SD5+A*/B*/C*/D*Q7+A*/B*/C*/D*Q1+A*/B*/C*/D*Q7+A*/B*/C*/D*Q2+A*/B*/C*/D*Q1
+A*/B*/C*/D*Q5
SD6=/A*/B*/C*/D*Q6+/A*/C*/D*Q3+/A*/C*/D*Q1+/A*/B*/C*/D*Q6+/A*/B*/C*/D*Q3
+/A*/B*/C*/D*Q5+A*/B*/C*/D*Q1
D6 =SD6+A*/B*/D*Q8+A*/B*/C*/D*Q8+A*/B*/C*/D*Q5+A*/B*/C*/D*Q5+A*/B*/C*/D*Q3
+A*/B*/C*/D*Q8
SD7=/A*/C*/D*Q5+/A*/C*/D*Q7+/A*/C*/D*Q5+/A*/B*/C*/D*Q2+/A*/B*/C*/D*Q1
+A*/B*/C*/D*Q7+A*/B*/C*/D*Q1
D7 =SD7+A*/B*/C*/D*Q7+A*/B*/C*/D*Q3+A*/B*/C*/D*Q8+A*/B*/C*/D*Q8+A*/B*/C*/D*Q5
+A*/B*/C*/D*Q2

```

แสดงรูปแบบสมการส่วนของการสลับบิตกลับบิต 4-7

จากสมการในภาพที่ 25 และ 26 มี SD0-SD3 เป็นตัวป้อนกลับของ GAL20V8 ตัวแรก และ SD0-SD4 เป็นตัวป้อนกลับของ GAL20V8 ตัวที่สอง สำหรับเพิ่มโปรคักทอมในการโปรแกรมมีเอาต์พุต D0-D7 สำหรับ GAL20V8ตัวแรกให้อาต์พุต D0-D3 ส่วน GAL20V8 ตัวที่สองให้อาต์พุตเป็น D4-D7

ภาพที่ 27

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
 Copyright (R) National Semiconductor Corporation 1990,1991

Document file for RE\_MUXA.eqn  
 Device: 20V8  
 \$LABELS 24 NC A B C D Q1 Q2 Q3 Q4 Q5 Q6 GND NC Q7 D3 D2 SD3 SD2 SD1 SD0 D1 D0 Q8 VCC

Pin	Label	Type
1	NC	unused
2	A	com input
3	B	com input
4	C	com input
5	D	com input
6	Q1	com input
7	Q2	com input
8	Q3	com input
9	Q4	com input
10	Q5	com input
11	Q6	com input
12	GND	ground pin
13	NC	unused
14	Q7	com input
15	D3	pos,trst,com output
16	D2	pos,trst,com output
17	SD3	pos,trst,com feedback
18	SD2	pos,trst,com feedback
19	SD1	pos,trst,com feedback
20	SD0	pos,trst,com feedback
21	D1	pos,trst,com output
22	D0	pos,trst,com output
23	Q8	com input
24	_VCC	power pin

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
 Copyright (R) National Semiconductor Corporation 1990,1991

Device Utilization:

No of dedicated inputs used : 12/14 (85.7%)  
 No of dedicated outputs used : 2/2 (100.0%)  
 No of feedbacks used as dedicated outputs : 2/6 (33.3%)  
 No of feedbacks used : 4/6 (66.7%)

Pin	Label	Terms Usage
22	D0	8/8 (100.0%)
21	D1	8/8 (100.0%)
20	SD0	8/8 (100.0%)
19	SD1	8/8 (100.0%)
18	SD2	8/8 (100.0%)
17	SD3	8/8 (100.0%)
16	D2	8/8 (100.0%)
15	D3	8/8 (100.0%)
<b>Total</b>		<b>64/64 (100.0%)</b>

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
Copyright (R) National Semiconductor Corporation 1990,1991

Chip diagram (DIP)

NC	1	24	VCC
A	2	23	Q8
B	3	22	D0
C	4	21	D1
D	5	20	SD0
Q1	6	19	SD1
Q2	7	18	SD2
Q3	8	17	SD3
Q4	9	16	D2
Q5	10	15	D3
Q6	11	14	Q7
GND	12	13	NC

แสดงข้อมูลของ GAL20V8 ในการนำมาใช้งานส่วนการสลับตำแหน่งบิตกลับบิต 0-3

ภาพที่ 28

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)

Copyright (R) National Semiconductor Corporation 1990,1991

Document file for RE\_MUXB.eqn

Device: 20V8

\$LABELS 24 NC A B C D Q1 Q2 Q3 Q4 Q5 Q6 GND NC Q7 D7 D6 SD7 SD6 SD5 SD4 D5  
D4 Q8 VCC

Pin	Label	Type
1	NC	unused
2	A	com input
3	B	com input
4	C	com input
5	D	com input
6	Q1	com input
7	Q2	com input
8	Q3	com input
9	Q4	com input
10	Q5	com input
11	Q6	com input
12	GND	ground pin
13	NC	unused
14	Q7	com input
15	D7	pos,trst,com output
16	D6	pos,trst,com output
17	SD7	pos,trst,com feedback
18	SD6	pos,trst,com feedback
19	SD5	pos,trst,com feedback
20	SD4	pos,trst,com feedback
21	D5	pos,trst,com output
22	D4	pos,trst,com output
23	Q8	com input
24	VCC	power pin

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)

Copyright (R) National Semiconductor Corporation 1990,1991

Device Utilization:

No of dedicated inputs used : 12/14 (85.7%)  
No of dedicated outputs used : 2/2 (100.0%)  
No of feedbacks used as dedicated outputs : 2/6 (33.3%)  
No of feedbacks used : 4/6 (66.7%)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้ทำซ้ำโดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin	Label	Terms Usage
22	D4	8/8 (100.0%)
21	D5	8/8 (100.0%)
20	SD4	8/8 (100.0%)
19	SD5	8/8 (100.0%)
18	SD6	8/8 (100.0%)
17	SD7	8/8 (100.0%)
16	D6	8/8 (100.0%)
15	D7	8/8 (100.0%)
<b>Total</b>		<b>64/64 (100.0%)</b>

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
 Copyright (R) National Semiconductor Corporation 1990,1991  
 Chip diagram (DIP)

NC	1	24	VCC
A	2	23	Q8
B	3	22	D4
C	4	21	D5
D	5	20	SD4
Q1	6	19	SD5
Q2	7	18	SD6
Q3	8	17	SD7
Q4	9	16	D6
Q5	10	15	D7
Q6	11	14	Q7
GND	12	13	NC

แสดงข้อมูลของ GAL20V8 ในการนำไปใช้งานส่วนการสลับตำแหน่งบิตกลับบิต 4-7

ภาพที่ 27 และ 28 ซึ่งในแฟ้มข้อมูลนี้จะบอกรายละเอียดในการใช้งาน GAL20V8 และตำแหน่งของขาที่กำหนดใช้งาน

#### ส่วนของการคำนวณหาค่าซินโดรม (Syndrome)

ส่วนการหาค่าที่ใช้ในการเลือกรูปแบบที่ผิดพลาดมาทำการบวกแบบเอกคลูซิฟออร์กับข้อมูลข่าวสารจากการเข้ารหัส เพื่อให้ได้ข้อมูลที่ต้องการกลับคืนมา จากสมการที่ 3.5 สำหรับการคำนวณหาซินโดรมสามารถนำมาเขียนเป็นสมการรูปของลอจิกได้ดังนี้

$$S_0 = r_1 \oplus r_2 \oplus r_4 \oplus r_6 \oplus r_7 \oplus r_8$$

$$S_1 = r_2 \oplus r_3 \oplus r_4 \oplus r_5 \oplus r_7 \oplus r_9$$

$$S_2 = r_0 \oplus r_3 \oplus r_4 \oplus r_5 \oplus r_6 \oplus r_{10}$$

$$S_3 = r_0 \oplus r_1 \oplus r_5 \oplus r_6 \oplus r_7 \oplus r_1$$

(4.2)

จากสมการที่ 4.2 เป็นการคำนวณหาค่าของซินโดรม S0-S3 สามารถนำมาเขียนเป็นสมการลอจิกสำหรับโปรแกรมลงใน PLD ให้ทำงานตามฟังก์ชันดังแสดงในภาพที่ 29 และในภาพที่ 30 โดยมี 12 อินพุต คือ A1-A12 และ 4 เอาต์พุต คือ S0-S3 แต่เนื่องจากการแปลงสมการ 4.2 มีจำนวนของโปรดักเทอมมากทำให้ต้องเลือกใช้ GAL จำนวน 2 ตัว ตัวแรกใช้ GAL20V8 ซึ่งสามารถใช้ได้ 12 อินพุต และใช้ 8 เอาต์พุต โดยใช้ 4 เอาต์พุตเป็นตัวป้อนกลับสำหรับเพิ่มจำนวนโปรดักเทอมแต่ยังมีจำนวนโปรดักเทอมที่เหลือจึงใช้ GAL16V8 เพิ่มอีกหนึ่งตัวสำหรับทำการหาซินโดรม โดยรับอินพุตจาก GAL20V8 จะให้อาต์พุตเป็น S1-S4 เป็นค่าของซินโดรมที่ได้

ภาพที่ 29

```

SYNDROME
CHIP SYNDROME GAL20V8
NC A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 GND
NC A11 SO4 SO3 SE4 SE3 SE2 SE1 SO2 SO1 A12 VCC
EQUATIONS

SE1 = /A2*/A3*A5+/A2*A3*/A5+A2*/A3*/A5+A2*A3*A5
SO1 = /SE1*/A7*A8+/SE1*A7*/A8+SE1*/A7*/A8+SE1*A7*A8

SE2 = /A3*/A4*A5+/A3*A4*/A5+A3*/A4*/A5+A3*A4*A5
SO2 = /SE2*/A6*A8+/SE2*A6*/A8+SE2*/A6*/A8+SE2*A6*A8

SE3 = /A1*/A4*A5+/A1*A4*/A5+A1*/A4*/A5+A1*A4*A5
SO3 = /SE3*/A6*A7+/SE3*A6*/A7+SE3*/A6*/A7+SE3*A6*A7

SE4 = /A1*/A2*A6+/A1*A2*/A6+A1*/A2*/A6+A1*A2*A6
SO4 = /SE4*/A7*A8+/SE4*A7*/A8+SE4*/A7*/A8+SE4*A7*A8

```

แสดงรูปแบบของสมการส่วนหาซินโดรม 1

ภาพที่ 30

```

S_DROM2
CHIP SYNDROME GAL16V8
NC SO1 SO2 SO3 SO4 A9 A10 A11 A12 GND
NC NC NC NC NC S4 S3 S2 S1 VCC
EQUATIONS

S1 = SO1*/A9+/SO1*A9
S2 = SO2*/A10+/SO2*A10
S3 = SO3*/A11+/SO3*A11
S4 = SO4*/A12+/SO4*A12

```

แสดงรูปแบบของสมการส่วนหาซินโดรม 2

## ภาพที่ 31

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
 Copyright (R) National Semiconductor Corporation 1990,1991  
 Document file for S\_DROM1.eqn  
 Device: 20V8

\$LABELS 24 NC A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 GND NC A11 SO4 SO3 SE4 SE3 SE2  
 SE1 SO2 SO1 A12 VCC

Pin	Label	Type
1	NC	unused
2	A1	com input
3	A2	com input
4	A3	com input
5	A4	com input
6	A5	com input
7	A6	com input
8	A7	com input
9	A8	com input
10	A9	unused
11	A10	unused
12	GND	ground pin
13	NC	unused
14	A11	unused
15	SO4	pos,trst,com output
16	SO3	pos,trst,com output
17	SE4	pos,trst,com feedback
18	SE3	pos,trst,com feedback
19	SE2	pos,trst,com feedback
20	SE1	pos,trst,com feedback
21	SO2	pos,trst,com output
22	SO1	pos,trst,com output
23	A12	unused
24	VCC	power pin

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
 Copyright (R) National Semiconductor Corporation 1990,1991  
 Device Utilization:

No of dedicated inputs used : 8/14 (57.1%)  
 No of dedicated outputs used : 2/2 (100.0%)  
 No of feedbacks used as dedicated outputs : 2/6 (33.3%)  
 No of feedbacks used : 4/6 (66.7%)

Pin	Label	Terms Usage
22	SO1	5/8 (62.5%)
21	SO2	5/8 (62.5%)
20	SE1	5/8 (62.5%)
19	SE2	5/8 (62.5%)
18	SE3	5/8 (62.5%)
17	SE4	5/8 (62.5%)
16	SO3	5/8 (62.5%)
15	SO4	5/8 (62.5%)
<b>Total</b>		<b>40/64 (62.5%)</b>

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
 Copyright (R) National Semiconductor Corporation 1990,1991  
 Chip diagram (DIP)

NC		1		24		VCC
A1		2		23		A12
A2		3		22		SO1
A3		4		21		SO2
A4		5		20		SE1
A5		6		19		SE2
A6		7		18		SE3
A7		8		17		SE4
A8		9		16		SO3
A9		10		15		SO4
A10		11		14		A11
GND		12		13		NC

แสดงข้อมูลของ GAL20V8 ในการนำมาใช้งานส่วนหาซินโดรม 1

ภาพที่ 32

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
 Copyright (R) National Semiconductor Corporation 1990,1991

Document file for S\_DROM2.eqn  
 Device: 16V8

\$LABELS 20 NC SO1 SO2 SO3 SO4 A9 A10 A11 A12 GND NC NC NC NC NC NC S4 S3 S2 S1  
 VCC

Pin	Label	Type
1	NC	unused
2	SO1	com input
3	SO2	com input
4	SO3	com input
5	SO4	com input
6	A9	com input
7	A10	com input
8	A11	com input
9	A12	com input
10	GND	ground pin
11	NC	unused
12	NC	unused
13	NC	unused
14	NC	unused
15	NC	unused
16	S4	pos,com output
17	S3	pos,com output
18	S2	pos,com output
19	S1	pos,com output
20	VCC	power pin

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
 Copyright (R) National Semiconductor Corporation 1990,1991

Device Utilization:  
 No of dedicated inputs used : 8/10 (80.0%)  
 No of dedicated outputs used : 2/2 (100.0%)  
 No of feedbacks used as dedicated outputs : 2/6 (33.3%)

Pin	Label	Terms Usage
19	S1	2/8 (25.0%)
18	S2	2/8 (25.0%)
17	S3	2/8 (25.0%)
16	S4	2/8 (25.0%)
Total		8/64 (12.5%)

EQN2JED - Boolean Equations to JEDEC file assembler (Version V003)  
 Copyright (R) National Semiconductor Corporation 1990,1991  
 Chip diagram (DIP)

NC	1	20	VCC
SO1	2	19	S1
SO2	3	18	S2
SO3	4	17	S3
SO4	5	16	S4
A9	6	15	NC
A10	7	14	NC
A11	8	13	NC
A12	9	12	NC
GND	10	11	NC

แสดงข้อมูลของ GAL16V8 ในการนำมาใช้งานส่วนหาซิน โครม 2

ภาพที่ 31 และ ภาพที่ 32 ซึ่งเป็นไฟล์แสดงรายละเอียดของการใช้งานและตำแหน่งของขา  
 สำหรับการใช้งาน

## บทที่ 5

### การทดลองและผลการทดลอง

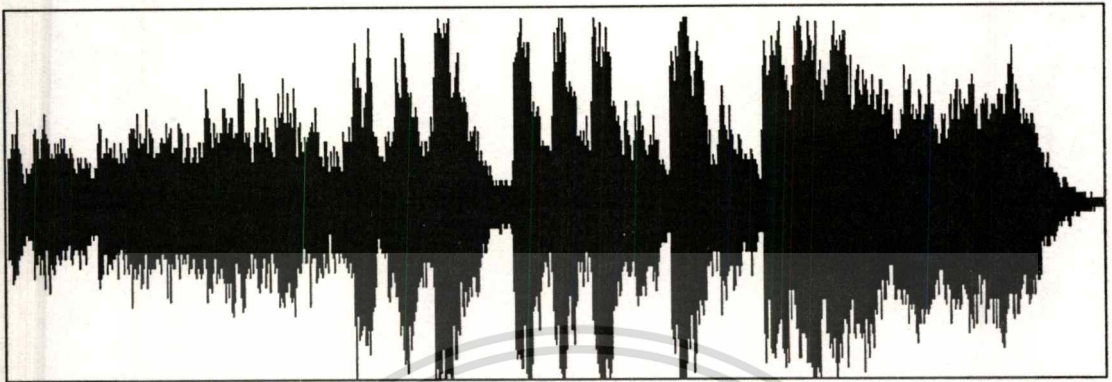
#### ผลการทดลอง

การทดลองแพลงจรวดตัวเข้ารหัสลับและตัวถอดรหัสลับกระทำโดยนำเอาสัญญาณที่อยู่ในรูปแบบต่าง ๆ มาทำการเข้ารหัสและถอดรหัสลับดังนี้ สัญญาณแอนะล็อกได้แก่ สัญญาณเสียงดนตรีและสัญญาณเสียงพูด และสัญญาณดิจิทัล

การทดลองในรูปแบบแรกใช้สัญญาณเสียงดนตรีที่อยู่ในรูปแบบสัญญาณแอนะล็อก มาทำการแปลงเป็นสัญญาณดิจิทัลก่อนทำการเข้ารหัสลับ ซึ่งสัญญาณเสียงดนตรีต้นแบบแสดงดังในภาพที่ 33a เป็นสัญญาณเสียงดนตรีที่นำมาทำการเข้ารหัสลับ แล้วนำผลของการเข้ารหัสลับที่ได้มาทำการแปลงให้อยู่ในรูปของสัญญาณแอนะล็อกดังแสดงในภาพที่ 33b และหลังจากนั้นได้นำสัญญาณที่ผ่านการเข้ารหัสแล้วมาทำการถอดรหัสลับ แล้วทำการแปลงให้อยู่ในรูปของสัญญาณแอนะล็อกดังแสดงในภาพที่ 33c

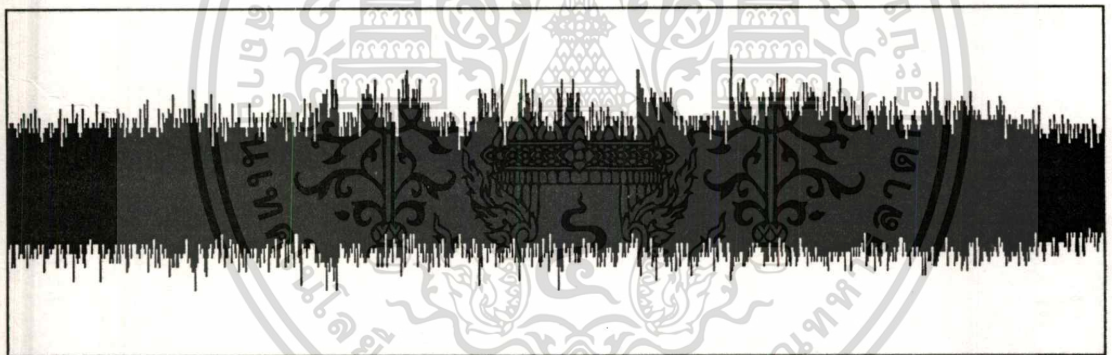
จากรูปของสัญญาณในภาพที่ 33c ซึ่งเป็นสัญญาณของเสียงดนตรี เปรียบเทียบกับรูปร่างของสัญญาณในภาพที่ 33b ที่ผ่านการเข้ารหัส แล้วจะเห็นว่ารูปร่างของสัญญาณมีความแตกต่างกันไปจากสัญญาณต้นแบบในภาพที่ 33a และเมื่อทำการถอดรหัสลับ จะได้รูปแบบของสัญญาณดังแสดงในภาพที่ 33c กลับมา สังเกตเห็นว่ารูปร่างของสัญญาณมีความคล้ายคลึงกับข้อมูลต้นแบบ จะมีส่วนแตกต่างกันบ้างอันเนื่องจากการสูญเสียในระหว่างการแปลงสัญญาณจากแอนะล็อกเป็นดิจิทัล เพราะว่าสัญญาณความถี่ของเสียงดนตรีค่อนข้างสูงและมีรายละเอียดค่อนข้างมาก

ภาพที่ 33a



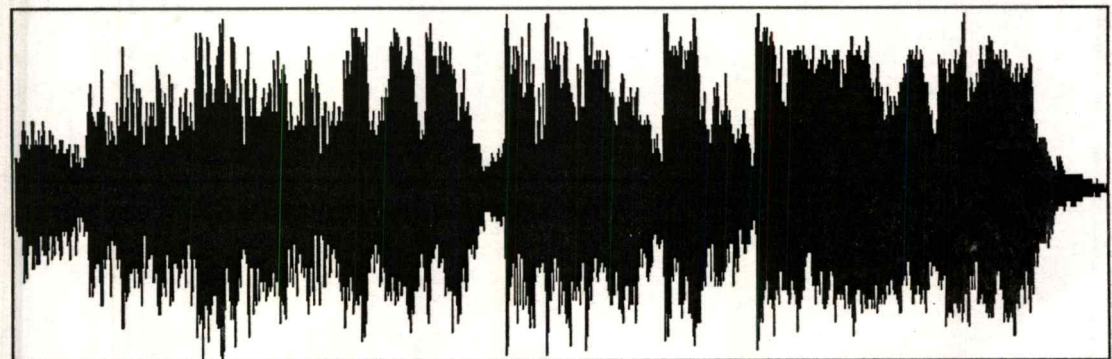
แสดงรูปแบบสัญญาณเสียงคนตรีต้นแบบ

ภาพที่ 33b



แสดงรูปแบบสัญญาณเสียงคนตรีที่ผ่านการเข้ารหัสลับ

ภาพที่ 33c



แสดงรูปแบบสัญญาณเสียงคนตรีที่ผ่านการถอดรหัสลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทดลองรูปแบบที่สองใช้สัญญาณเสียงพูดเพียงอย่างเดียว โดยมีรูปแบบสัญญาณต้นแบบแสดงในภาพที่ 34a สำหรับนำมาทำการเข้ารหัสลับ โดยจะทำการแปลงสัญญาณเหล่านี้ให้อยู่ในรูปของสัญญาณดิจิทัล แล้วทำการเข้ารหัสลับนำผลที่ได้จากการเข้ารหัสมาทำการแปลงให้อยู่ในรูปสัญญาณแอนะล็อกดังแสดงในภาพที่ 34b และในภาพที่ 34c เป็นการแสดงรูปแบบของสัญญาณเสียงที่ผ่านการถอดรหัสลับแล้ว และแปลงให้อยู่ในรูปแบบของสัญญาณแอนะล็อก

ภาพที่ 34a



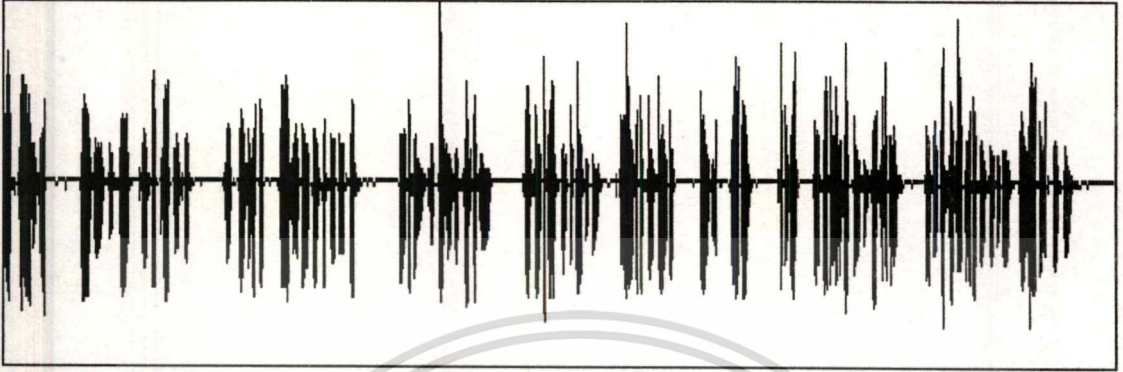
แสดงรูปแบบของสัญญาณเสียงพูดต้นแบบ

ภาพที่ 34b



แสดงรูปแบบสัญญาณเสียงพูดที่ผ่านการเข้ารหัส

ภาพที่ 34c



แสดงรูปแบบของสัญญาณเสียงพูดที่ผ่านการถอดรหัสกลับ

จากผลการทดลองของรูปแบบที่สอง เมื่อทำการแปลงเสียงพูดต้นแบบในภาพที่ 34a ให้เป็นดิจิทัลแล้วนำมาทำการเข้ารหัสลับ และแปลงให้อยู่ในรูปของสัญญาณแอนะล็อกจะเห็นว่า มีรูปแบบสัญญาณที่แตกต่างไปจากสัญญาณต้นแบบแสดงในภาพที่ 34b และทำการถอดรหัสกลับจะ ได้รูปแบบของสัญญาณแสดงดังในภาพที่ 34c ซึ่งมีความใกล้เคียงกับสัญญาณต้นแบบมาก

ในการทดลองรูปแบบที่สาม กระทำโดยนำข้อมูลที่อยู่ในรูปแบบของสัญญาณดิจิทัลซึ่งเป็นข้อมูลต้นแบบแสดงในภาพที่ 35a มาทำการเข้ารหัสลับ โดยข้อมูลที่ผ่านการเข้ารหัสแล้วแสดงในภาพที่ 35b และเมื่อนำมาทำการถอดรหัสกลับจะได้ข้อมูลกลับคืนมาดังแสดงในภาพที่ 35c

ภาพที่ 35a

ABCDEFGHIJKLMNOPQRSTUVWXYZ

แสดงข้อมูลข่าวสารต้นแบบ

ภาพที่ 35b

'ÒÇÒlpS;[WO/|Èð-®İ©i<sup>-2</sup>Ä«E,Z®.”-°,«Èð

แสดงข้อมูลผ่านการเข้ารหัส

ภาพที่ 35c

ABCDEFGHIJKLMNOPQRSTUVWXYZ

แสดงข้อมูลผ่านการถอดรหัสลับ



## บทที่ 6

### สรุปผลและแนวทางการพัฒนา

#### สรุปผลการวิจัย

จากการวิจัยที่ทำมาเป็นการนำเสนอรูปแบบทางอัลกอริทึมและการนำมาสร้างเป็นแผงวงจรตัวเข้ารหัสลับและชุดตัวถอดรหัสลับ โดยอาศัยหลักการของบล็อกโค้ดเชิงเส้นในการสร้างความปลอดภัยให้กับข้อมูลข่าวสาร ด้วยการนำรูปแบบผิดพลาดที่ถูกกำหนดรูปแบบไว้แล้ว มาทำการบวกแบบเอกคลูซีฟออร์เข้ากับรหัสค้ำในส่วนที่เป็นข้อมูลแล้วนำรหัสค้ำที่ได้มาทำการสลับบิตในส่วนของคุณค่า โดยมีการสลับบิตเป็นตัวเลือกรูปแบบการสลับทางด้านการถอดรหัส ใช้วิธีการคำนวณหาค่าขึ้นโคจรมาทำการถอดรหัสกลับคืนมาโดยวิธีการทั้งหมดได้กล่าวรายละเอียดในบทที่ 3

ในการวิจัยนี้ได้ศึกษาและนำมาสร้างเป็นแผงวงจรสำหรับชุดเข้ารหัสลับและชุดถอดรหัสลับ โดยได้นำอุปกรณ์ประเภท PLD (Programmable logic device) มาใช้สำหรับช่วยในการลดขนาดของแผงวงจรให้มีขนาดเล็กลง และค่าใช้จ่ายในการสร้างที่มีราคาต่ำ ซึ่งจะเป็นแนวทางหนึ่งสำหรับผู้สนใจในการสร้างอุปกรณ์ประเภทนี้ขึ้นมาใช้งาน โดยอาศัยเทคโนโลยีที่สามารถหาได้ในประเทศ

#### ปัญหาที่พบและข้อเสนอแนะ

เนื่องจากการออกแบบสร้างแผงวงจรตัวเข้ารหัสลับและตัวถอดรหัสลับ ใช้อุปกรณ์ไอซีตามที่พอหาซื้อได้ มาใช้สร้างแผงวงจร ทำให้ขนาดของแผงวงจรที่ได้มีขนาดไม่เล็กเท่าที่ควร ส่วนในการทดลองในการเข้ารหัสลับนั้นก็ไม่มีเครื่องมือที่ดีพอที่จะใช้ในการตรวจสอบบิตที่ทำการเข้ารหัสลับ ต้องใช้วิธีการตรวจเช็คด้วยการวัดตรวจสอบทีละบิต ทำให้มีความยุ่งยากและเสียเวลามากในการตรวจสอบการทำงานของแผงวงจร

สำหรับแนวทางในการพัฒนาต่อในการเข้ารหัสลับ ในด้านของอัลกอริทึมอาจต้องเพิ่มความซับซ้อนให้มากสำหรับสร้างความเชื่อถือในการนำไปใช้งานในระดับต่างๆ เพิ่มขึ้น และอาจจะต้องอาศัยชีพการประมวลผลสัญญาณดิจิทัลเข้ามาช่วยเพิ่มประสิทธิภาพในการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับให้เพียงในเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลเห็นประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บรรณานุกรม

1. Shu Lin. "An Introduction to Error\_Correcting Code." Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1970
2. Peter Sweeney. "Error control coding an introduction." Prentice Hall, 1991.
3. Henry J. Beker and Fred C. Piper. "Secure Speech Communications." Academic Press, 1985.
4. Richard S. Sandige. "Modern digital design." McGraw-Hill Publishing Company 1990.
5. National Semiconductor. "Programmable logic Devices Databook And Design Guide." 1990
6. D. Brook and R.J. Wynne. "Signal Processing : Principle and Applications." Edward Arnold 1988..
7. National Semiconductor. "Open Programable Architecture Language: OPAL junior." 1991
8. พุศักรดิ์ ชิวสุวิทย์. "การแก้รหัสผิด." คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง พ.ศ. 2528
9. พุศักรดิ์ ชิวสุวิทย์ และคเชนทร์ แจ่มกมล. "การเข้ารหัสลับโดยใช้ชิปไมโครของลิเนียร์บล็อกโค้ดและการสลับบิตโดยอาศัยเพอซีโดแรนคอม." วารสารทางวิชาการของสมาคมคอมพิวเตอร์แห่งประเทศไทย
10. พุศักรดิ์ ชิวสุวิทย์ และคเชนทร์ แจ่มกมล. "การออกแบบตัวเข้ารหัสลับแบบเวลาจริงโดยใช้ชิปไมโครของลิเนียร์บล็อกโค้ดและการสลับบิตโดยอาศัยการกำเนิดแบบสุ่ม." การประชุมวิชาการทางไฟฟ้า ครั้งที่ 18 มหาวิทยาลัยเทคโนโลยีมหานคร, หน้า 1034-1038, พฤศจิกายน 2538
11. K. Sam Shanmugam. "Digital and analog communication system." John Wiley & Sons, Inc, 1978
12. Vera Pless. "Introduction to the theory of error-correcting codes." John Wiley & Sons, Inc 1982



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ภาคผนวก ก**  
**ผลงานวิจัยที่ได้รับการตีพิมพ์**

- 1 พุศัคดี ชิวสุวิทย์ และคเชนทร์ แจ่มกมล, "การออกแบบตัวเข้ารหัสแบบเวลาจริงโดยใช้ซินโดรมของลิเนียร์บล็อกโค้ดและการสลับบิตโดยอาศัยการกำเนิดแบบสุ่ม", การประชุมวิชาการทางไฟฟ้า ครั้งที่ 18 มหาวิทยาลัยเทคโนโลยีมหานคร หน้า 1034-1038, พฤศจิกายน 2538
- 2 พุศัคดี ชิวสุวิทย์ และคเชนทร์ แจ่มกมล, "การเข้ารหัสลับโดยใช้ซินโดรมของลิเนียร์บล็อกโค้ดและการสลับบิตโดยอาศัยเพอซีโคเรนคอม", วารสารทางวิชาการของสมาคมคอมพิวเตอร์แห่งประเทศไทย
- 3 บัณฑิต สุมนวัฒน์เดช คเชนทร์ แจ่มกมล และ พุศัคดี ชิวสุวิทย์, "แผนวงจรควบคุมการเก็บภาพขนาด 512X512 จุดภาพ โดยใช้ตัวแปลงสัญญาณแอนะล็อกเป็นดิจิทัล 2 ชุดร่วมกันทำงาน", เอกสารการประชุมใหญ่ทางวิชาการประจำปี 2536 วิศวกรรมสถานแห่งประเทศไทย ในพระบรมราชูปถัมภ์, หน้า 630 - 641, 27-30 พฤศจิกายน 2536.

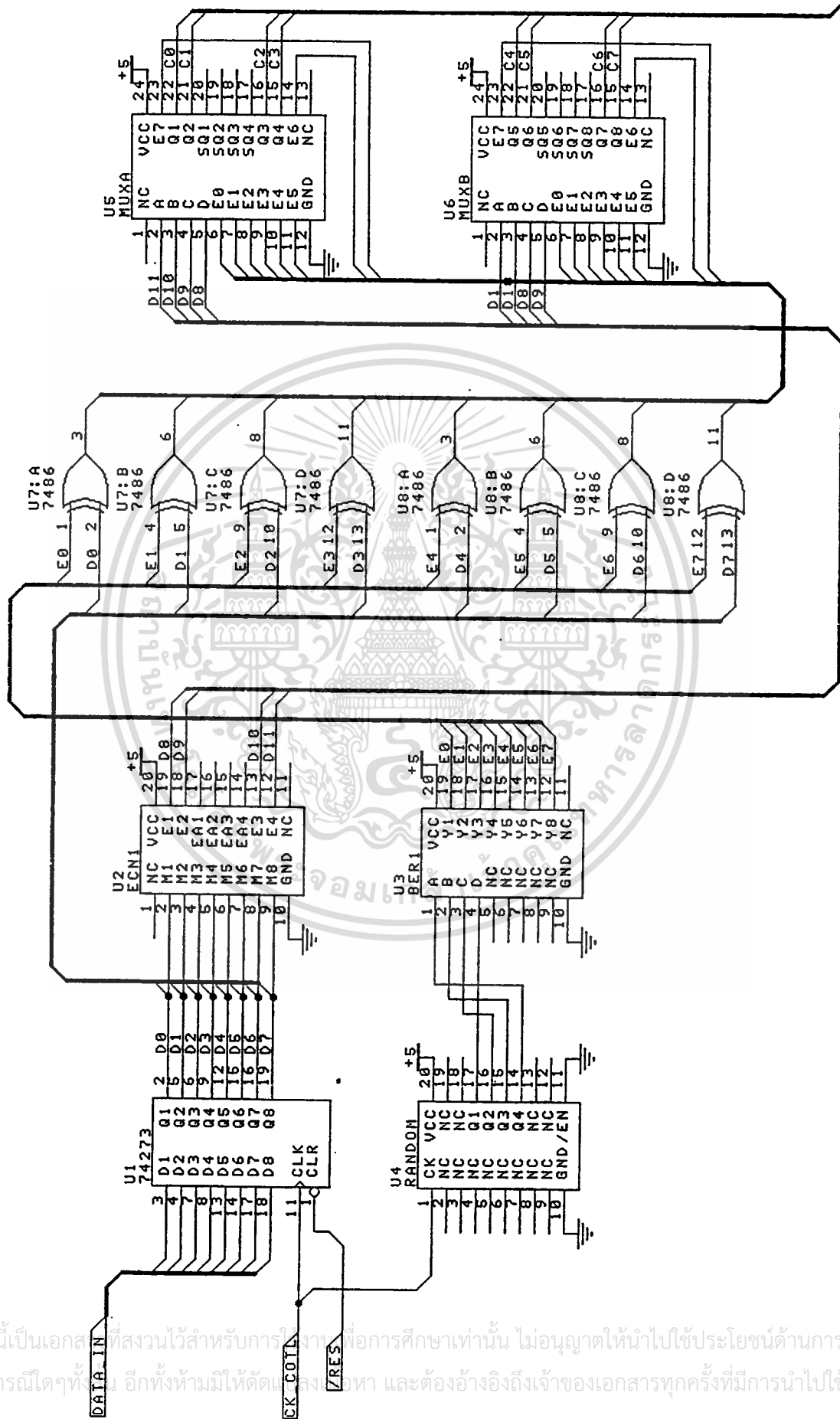
## ภาคผนวก ข

### วงจรรวมแผงวงจรตัวเข้ารหัสลับและตัวถอดรหัสลับ

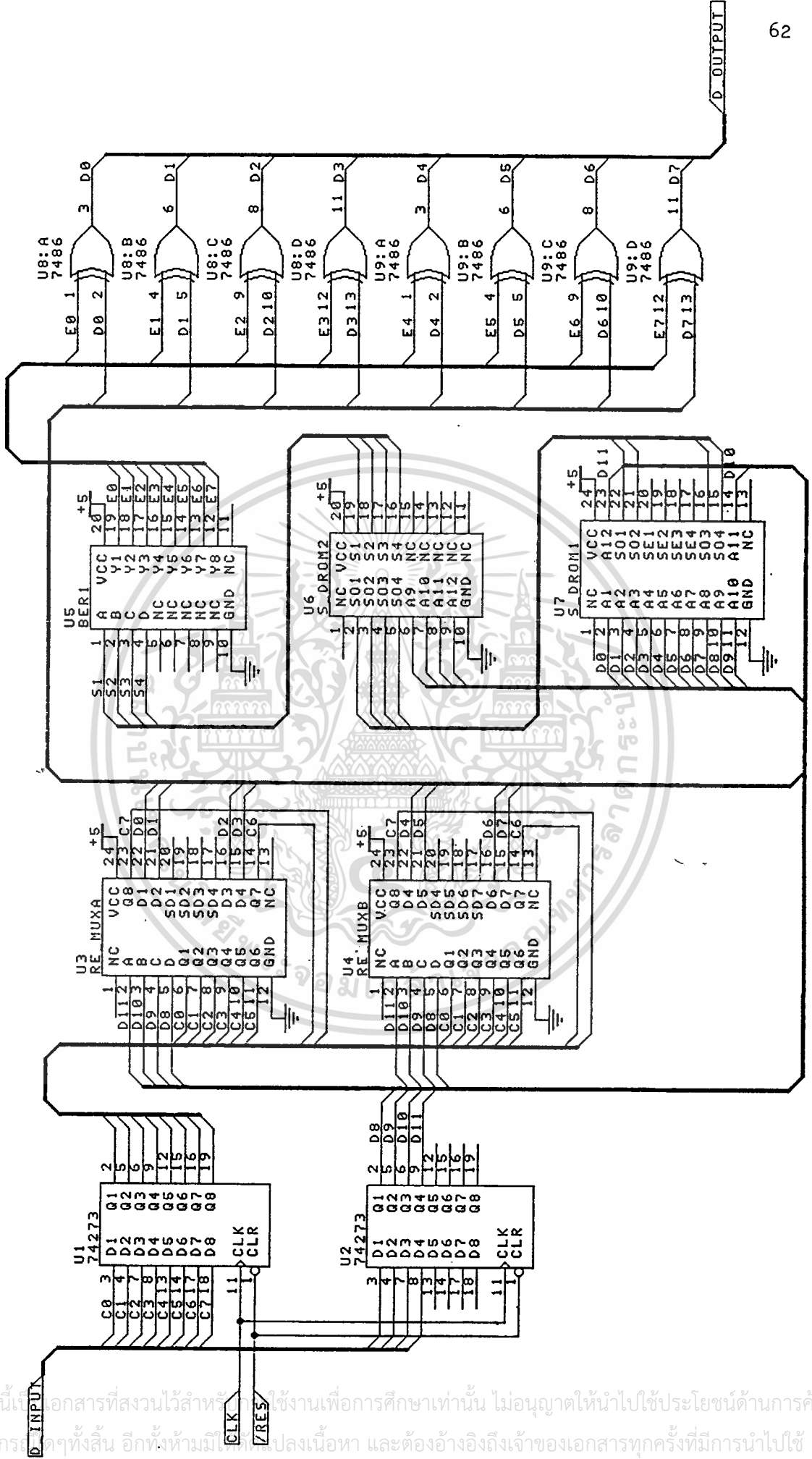
ภาพที่ ข-1 แสดงวงจรรวมของแผงวงจรตัวเข้ารหัสลับ

ภาพที่ ข-2 แสดงวงจรรวมของแผงวงจรตัวถอดรหัสลับ





แสดงวงรวมของแผงวงจรตัวเข้ารหัสลับ



แสดงวงจรรวมของแผงวงจรตัวถอดรหัสที่สิบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ไม่ควรนำเอกสารนี้ไปเผยแพร่ในที่สาธารณะโดยไม่ได้รับอนุญาต และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ประวัติผู้เขียน

ชื่อ	นายคเชนทร์ แจ่มกมล
เกิดวันที่	20 สิงหาคม 2512 ที่จังหวัดสุราษฎร์ธานี
การศึกษา	ปีการศึกษา 2518-2524 ระดับประถม โรงเรียนมานิตานุเคราะห์ สุราษฎร์ธานี ปีการศึกษา 2525-2527 ระดับมัธยมศึกษาตอนต้น โรงเรียนสุราษฎร์ธานี สุราษฎร์ธานี ปีการศึกษา 2528-2530 ระดับ ปวช. สาขาอิเล็กทรอนิกส์ วิทยาลัยเทคนิคสุราษฎร์ธานี สุราษฎร์ธานี ปีการศึกษา 2531-2532 ระดับ ปวส. สาขาคอมพิวเตอร์เทคโนโลยี สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ กทม. ปีการศึกษา 2533-2534 สาขา ระดับปริญญาตรีอุตสาหกรรมศาสตรบัณฑิต สาขาคอมพิวเตอร์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กทม.
ปัจจุบัน	ทำงานที่บริษัท เอ็นเอส อิเล็กทรอนิกส์กรุงเทพ (1993) จำกัด (NSEB) ในตำแหน่ง Test Engineer