รายงานโครงการวิจัย

เรื่อง

การเข้ารหัสลับเพื่อป้องกันการรั่วไหลของข้อมูลข่าวสาร

(Data scrambler for preventing the secret information from trapping)

โดย

รศ.ดร.ฟูศักดิ์ ชีวสุวิทย์

ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมสาตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ทุนอุดหนุนการวิจัยปีงบประมาณ 2540

โครงการวิจัยเรื่อง "การเข้ารหัสลับเพื่อป้องกันการรั่วไหลของข้อมูลข่าวสาร"

(Data scramble for preventing the secret information from trapping)

บทคัดย่อ

การสื่อสารข้อมูลเป็นที่ยอมรับกันทั่วไปว่ามีความสำคัญต่อชีวิตประจำวันมาก แนวโน้มของการแลกเปลี่ยนข้อมูลส่วนตัวมักจะส่งผ่านโครงข่ายของการสื่อสาร จุดนี้เองเป็นสาเหตุที่จะทำให้เกิดการรั่วไหลของข้อมูล เนื่องจากทำการจารกรรมข้อมูลซึ่งเป็นความเสี่ยงต่อความปลอดภัยของข้อมูลและการละเมิดสิทธิส่วนบุคคล ดังนั้นในโครงการวิจัยได้เสนอเทคนิคการเข้ารหัสลับข้อมูลก่อนส่งออกไปในตัวกลางการสื่อสาร เพื่อป้องกันมิให้ข่าวสารรั่วไหลหรือถูกจารกรรม โดยเทคนิคการเข้ารหัสลับนี้ประกอบด้วยการประยุกต์ใช้ซีนโดรมของบล็อคโค้ดเชิงเส้น และการกำเนิดแบบสุ่ม ผลของการเข้ารหัสนี้ได้แสดงให้เห็นอย่างชัดเจนว่าสามารถปกปิดข้อมูลได้อย่างสมบูรณ์

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.

บทที่ 1
บทนำ

ความเป็นมาและความสำคัญของปัญหา

เนื่องจากปัจจุบันการติดต่อสื่อสารไม่ว่าจะเป็นทางโทรทัศน์ โทรศัพท์ คอมพิวเตอร์ วิทยุ ได้เข้ามามีบทบาทสำคัญอย่างสูงในสังคมปัจจุบัน จะมีกลุ่มของผู้ใช้บริการการสื่อสารในรูปต่าง 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 ในการวิจัยนี้ได้นำเสนอวิธีการเข้ารหัสโดยใช้หลักการของบล็อคโค้ดเชิงเส้นมาใช้สำหรับการเข้ารหัสลับและการถอดรหัสโดยอาศัยซีนโดรมของบล็อคโค้ดเชิงเส้นมาใช้สำหรับการป้องกันข้อมูลข่าวสารให้ได้รับความปลอดภัย

<u>วัตถุประสงค์ของการวิจัย</u>

1. เพื่อเป็นการศึกษาหาแนวทางในการออกแบบชุดตัวเข้ารหัสลับ และชุดตัวถอดรหัสลับและนำมาทำการสร้างเป็นแผงวงจร สำหรับทำงานตามโครงสร้างที่ทำการออกแบบ

2. เพื่อเป็นแนวทางสำหรับการพัฒนาในการออกแบบวิธีการสร้าง การเข้ารหัสลับและวิธีการถอดรหัสลับ โดยเป็นแนวทางสำหรับผู้สนใจสร้างอุปกรณ์ทางด้านการป้องกันข้อมูลข่าวสารให้ปลอดภัย และเป็นการช่วยประหยัดเวลาในการเริ่มศึกษาทางด้านนี้

3. สามารถนำแผงวงจรตัวเข้ารหัสลับและตัวถอดรหัสลับที่ทำการออกแบบ ไปใช้งานสำหรับป้องกันให้ข้อมูลให้มีความปลอดภัย

ขอบเขตของการวิจัย

ขอบเขตของการวิจัยจะเป็นการสร้างแผงวงจรการเข้ารหัสลับและแผงวงจรการถอดรหัสลับโดยอาศัยหลักการของบล็อคโค้ดเชิงเส้นและการหาซีนโดรม ซึ่งเป็นหลักวิธีการในการนำมาใช้ในการออกแบบวิธีการเข้ารหัสลับและวิธีการถอดรหัสลับ และนำวิธีการที่ได้มาทำการออกแบบสร้างเป็นแผงวงจรชุดสร้างรหัสลับและแผงวงจรชุดถอดรหัสลับ สำหรับการนำมาใช้งาน สามารถแบ่งเนื้อหากล่าวแยกเป็นบท ๆ ได้ดังนี้

บทที่ 1 บทนำ กล่าวถึงความเป็นมาและความสำคัญของงานวิจัย วัตถุประสงค์ในการทำวิจัยและรายละเอียดเนื้อหาในบทต่าง ๆ

บทที่ 2 ระบบการเข้ารหัสลับและทฤษฎี ในรายละเอียดจะเป็นการกล่าวถึงพื้นฐานของการเข้ารหัสแบบต่าง ๆ และทฤษฎีของบล็อคโค้ดเชิงเส้น

บทที่ 3 หลักการของตัวเข้ารหัสลับและตัวถอดรหัสลับ ในรายละเอียดจะกล่าวถึงการนำวิธีการของบล็อคโค้ดเชิงเส้นมาใช้ในการออกแบบวิธีการเข้ารหัสลับและวิธีการถอดรหัสลับและขั้นตอนต่าง ๆ ของการเข้ารหัส

บทที่ 4 การนำอุปกรณ์ประเภท PLD มาใช้งาน ในรายละเอียดเป็นการกล่าวถึงการนำอุปกรณ์ประเภท PLD มาใช้สำหรับสร้างแผงวงจรตัวเข้ารหัสลับและตัวถอดรหัสลับ รวมหลักการของอุปกรณ์ PLD

บทที่ 5 แสดงผลการทดลอง โดยทำการเข้ารหัสและถอดรหัสกับสัญญาณในรูปแบบแตกต่างกัน เช่น สัญญาณเสียงพูดเสียงดนตรีและแฟ้มข้อมูลจากคอมพิวเตอร์

บทที่ 6 บทสรุปผลและแนวทางการพัฒนา กล่าวสรุปเนื้อหาของงานวิจัยที่ทำมารวมถึงปัญหาที่ประสบในการทำวิจัย

บทที่ 2

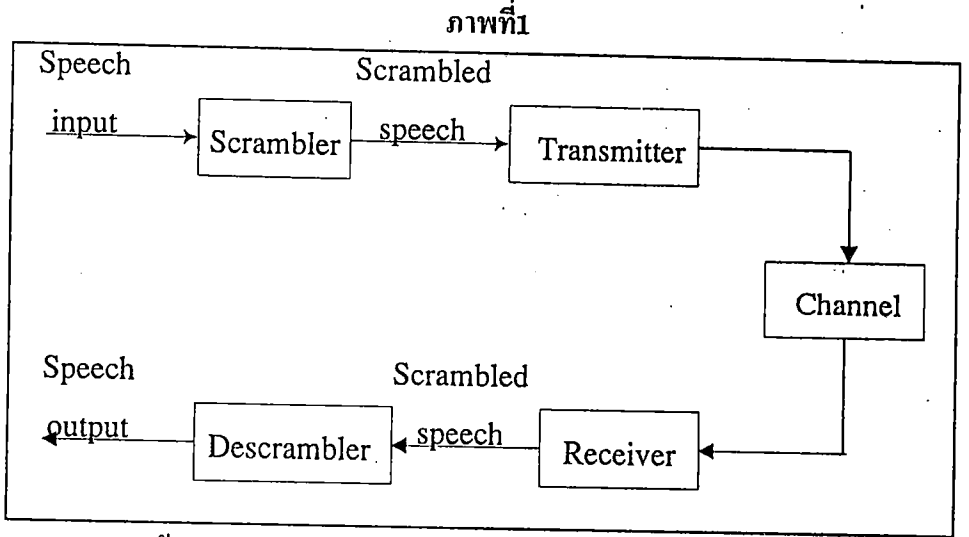
หลักการเข้ารหัสและบล็อคโค้ดเชิงเส้น

วิธีการสร้างรหัสลับ

โดยทั่วไปมักมีการกล่าวอ้างถึงวิธีการเข้ารหัสลับ (Scramble) แบบพื้นฐาน 2 วิธี คือ แบบแอนะล็อก (Analog) และแบบดิจิตอล (Digital) ส่วนใหญ่เครื่องเข้ารหัสสัญญาณที่มีความซับซ้อนมากๆ มักใช้กระบวนการของการประมวลผลสัญญาณดิจิตอล (Digital signal processing) กระบวนการนี้ทำงานโดยการแปลงสัญญาณแอนะล็อกไปเป็นสัญญาณดิจิตอลก่อนแล้วจึงทำการเข้ารหัส ส่วนในระบบที่เป็นแบบแอนะล็อกโดยส่วนใหญ่จะมีความปลอดภัยค่อนข้างน้อย

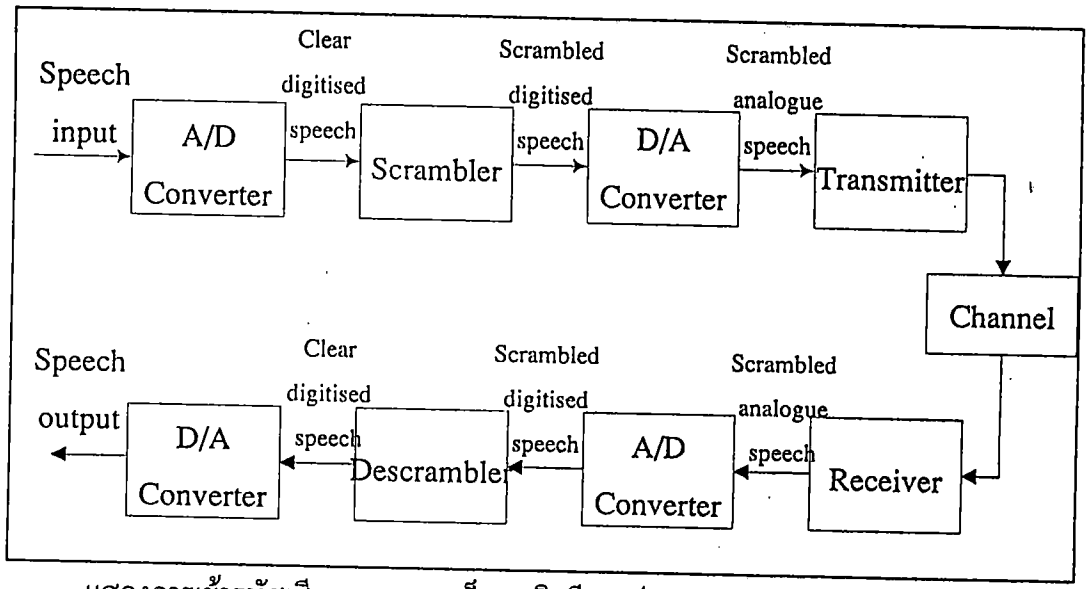
ข้อแตกต่างที่เห็นได้ชัดเจนระหว่างการเข้ารหัสแบบแอนะล็อกและแบบดิจิตอลคือ รูปแบบที่เกี่ยวกับตัวส่งผ่านซึ่งเป็นอุปกรณ์ที่ใช้ในการส่งสัญญาณที่เข้ารหัสแล้ว วัตถุประสงค์ของระบบที่มีการเข้ารหัสแบบแอนะล็อกคือ ส่งผ่านข่าวสารที่มีการเปลี่ยนแปลงอย่างต่อเนื่อง ในทางตรงข้ามระบบเข้ารหัสลับแบบดิจิตอลจะส่งผ่านข่าวสารด้วยสัญญาณที่สามารถนำข่าวสารไปได้เฉพาะตามจำนวนที่จำกัดไว้ ดังแสดงได้ในแผนภาพดังต่อไปนี้

ภาพที่1



แสดงการเข้ารหัสเสียงแบบแอนะล็อกชนิดไม่มีการประมวลผลสัญญาณดิจิตอล

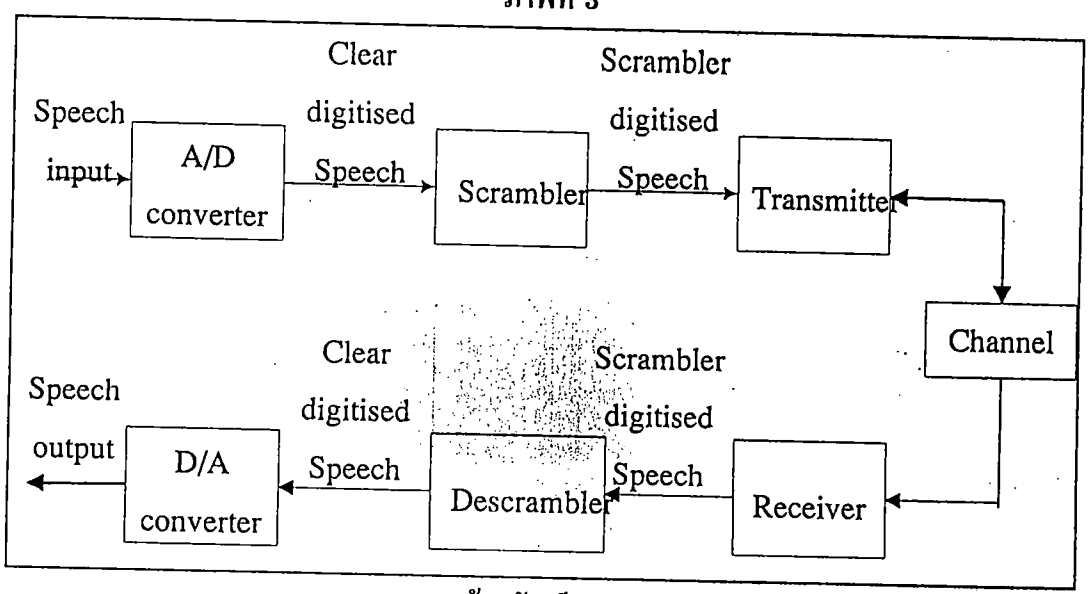
ภาพที่ 2



แสดงการเข้ารหัสเสียงแบบแอนะล็อกชนิดมีการประมวลผลสัญญาณแบบดิจิตอล

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

ภาพที่ 3



แสดงการเข้ารหัสเสียงแบบดิจิตอล

ไปอยู่ในรูปของสัญญาณดิจิตอลก่อนจะทำการเข้ารหัส แล้วถูกเปลี่ยนกลับให้อยู่ในรูปสัญญาณแอนะล็อกก่อนทำการส่งผ่านและหลังการส่งผ่านจะถูกเปลี่ยนกลับให้เป็นสัญญาณดิจิตอลอีกครั้งหนึ่ง แล้วจึงจะทำการถอดรหัสลับและท้ายสุดจะถูกเปลี่ยนให้เป็นสัญญาณแอนะล็อกอีกครั้งหนึ่ง

ในภาพที่ 3 แสดงให้เห็นถึงระบบดิจิตอลอีกประเภทหนึ่ง ความแตกต่างระหว่างระบบนี้และระบบในภาพที่ 2 คือไม่มีตัวแปลงสัญญาณจากดิจิตอลไปเป็นแอนะล็อก (D/A Converter) ในทันทีก่อนเข้าสู่เครื่องส่งผ่าน และไม่มีตัวแปลงสัญญาณจากแอนะล็อกไปเป็นดิจิตอลในทันทีหลังผ่านเครื่องรับ สำหรับในวิทยานิพนธ์นี้จะใช้รูปแบบดังแสดงในภาพที่ 3

การคำนวณทางคณิตศาสตร์ของรหัสแบบไบนารี

สำหรับการเข้ารหัสในวิทยานิพนธ์ฉบับนี้ จะใช้คณิตศาสตร์ที่มีความซับซ้อนน้อย ทั้งนี้เพื่อให้สามารถนำมาสร้างวงจรทางฮาร์ดแวร์ได้ง่าย คณิตศาสตร์ที่ว่านี้รู้จักกันในชื่อของ Galois field [1] เนื่องจากคณิตศาสตร์ที่ใช้มีลักษณะเป็นลอจิก คือมีสัญญลักษณ์เป็น 0 กับ 1 บางครั้งคณิศาสตร์ดังกล่าวนี้ถูกเรียกว่า Binary field ซึ่งเขียนย่อเป็น $GF(2)$ การทำงานของคณิศาสตร์ดังกล่าวแสดงผลตามตารางที่ 1 โดยการบวกจะมีลักษณะเป็นการทำเอกซ์คลูซีฟออร์ (Exclusive OR) ของลอจิก ส่วนการคูณจะมีลักษณะเป็นการ AND ของลอจิก ในการเข้ารหัสนี้จะมีการนำเอาเวคเตอร์ของข้อมูลข่าวสาร (Massage) คูณกับเมทริกซ์ที่ทำการออกแบบไว้ ผลลัพธ์ที่ได้จะเกิดจากการ AND และ EX-OR ของข้อมูลที่เป็นลอจิก จากเวคเตอร์ข้อมูลและแมทริกซ์ที่กำหนด

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

<u>การตรวจสอบพาริตี้</u> (Parity check)

เพื่อให้เข้าใจถึงการสร้างรหัสเชิงเส้น ให้ดูตัวอย่างง่ายๆ จากการสร้างรหัสคำ (Code word) เริ่มจากข้อมูลโดยปล่อยให้ผ่านไปในรหัสคำโดยตรง และจะมีบิทตามหลังแบบบิทเดียว ซึ่งเป็นการคำนวณจากบิทข้อมูลทั้งหมด มีวิธีการกำหนดบิทสุดท้ายที่ตามหลังมา 2 วิธี คือ

1. กำหนดบิทสุดท้ายเพื่อให้ผลรวมแบบโมดูโล (Modulo) 2 ของบิททั้งหมดในรหัสคำ มีค่าเท่ากับหนึ่ง

2. กำหนดบิทสุดท้ายเพื่อให้ผลรวมแบบโมดูโล 2 ของบิททั้งหมดในรหัสคำ มีค่าเท่ากับศูนย์

ในกรณีแรกรหัสคำมีพาริตี้แบบคี่ (Odd parity) กล่าวคือ จำนวนบิทที่เป็นหนึ่งในรหัสคำเป็นจำนวนคี่ ส่วนในกรณีที่สองมีจำนวนบิทที่เป็นหนึ่งในรหัสคำเป็นคู่ (Even parity) บิทที่เพิ่มขึ้นนอกเหนือจากบิทข้อมูลเรียกว่าพาริตี้เช็ค (Parity check) และอาจเรียกได้ว่าเป็นการตรวจสอบแบบพาริตี้คู่หรือพาริตี้คี่

รหัสแบบพาริตี้คี่และคู่จะแสดงในตารางที่ 2 และ 3 ตามลำดับ สำหรับกรณีที่มีบิทข้อมูล 3 ตำแหน่ง จะสังเกตเห็นว่ารหัสของตารางที่ 2 ไม่มีแถวที่เป็นศูนย์ทั้งหมด ซึ่งต้องเป็นส่วนหนึ่งของรหัสที่เป็นเชิงเส้น

ตารางที่ 2

ข้อมูลข่าวสาร	ข้อมูลเข้ารหัส
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

รหัสพาริตี้แบบคี่

ดังนั้นการตรวจสอบพาริตี้แบบคี่ทำให้เกิดรหัสแบบไม่เป็นเชิงเส้น ในทางตรงกันข้ามรหัสที่ใช้พาริตี้ในตารางที่ 3 จะเป็นรหัสเชิงเส้น ระบบที่มีการผลิตการตรวจสอบพาริตี้แบบคู่โดย

การเพิ่มบิทที่เป็นพาริตี้นั้นได้จากการบวกแบบโมดูโล 2 ของบิทในรหัสของข้อมูลข่าวสาร ตัวอย่าง เช่น รหัสข้อมูลข่าวสาร 101 เมื่อทำการบวกแบบโมดูโล 2 ของบิทที่มีค่าเป็น 1,0 และ 1 จะได้ผลลัพธ์คือ 0 ซึ่งจะเป็นค่าของบิทที่จะนำมาทำการต่อข้างท้ายของรหัสข้อมูลข่าวสาร กลายเป็น 1010

ตารางที่ 3

ข้อมูลข่าวสาร	ข้อมูลเข้ารหัส
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

รหัสพาริตี้แบบคู่

ในการเพิ่มพาริตี้อีกหนึ่งบิทให้รหัสของข้อมูลข่างสารนั้นปกติมักจะนำมาตรวจสอบได้ถ้าบิทผิดไปเพียงบิทเดียวเท่านั้น แต่ไม่สามารถนำมาทำการแก้ไขบิทที่ผิดได้ ในการแก้ไขบิทที่ผิดไปจะต้องใช้วิธีการแข้ารหัสที่เพิ่มพาริตี้บิทให้มากขึ้น อย่างเช่นการเข้ารหัสแบบบล็อคโค้ดเชิงเส้นที่จะได้กล่าวต่อไป

ความสามารถในการตรวจแก้บิทที่ผิดในรหัสเชิงเส้น

ในส่วนนี้จะได้กล่าวถึงคำศัพท์พื้นฐานที่ใช้ในการแก้บิทที่ผิดของรหัสเชิงเส้น

เวท (Weight) ของแฮมมิ่งสำหรับเวคเตอร์ v n-ทูเปิ้ลส์ คือ $\omega(\mathrm{v})$ ซึ่งหมายถึงผลรวมของจำนวนบิทของรหัสของ v ที่ไม่เป็นศูนย์ ตัวอย่างเช่น ถ้า v=[1 0 1 0 1 1 0 0 0 1] จะได้ $\omega(\mathrm{v}) = 5$

ให้ u และ v เป็นเวคเตอร์ n-ทูเปิ้ลส์ ค่าระยะห่างระหว่าง u และ v เขียนได้เป็น $d(\mathrm{u},\mathrm{v})$ ระยะห่างของสองเวคเตอร์ใด ๆ คือ จำนวนบิทรหัส “1” ที่แตกต่างกันของเวคเตอร์ทั้งสองเช่น

u=[1 0 0 1 0 1 1 0 0 0 1]
v=[1 1 0 0 1 0 1 0 1 0 1]

จะได้ $d(\mathrm{u},\mathrm{v})=5$
ถ้านำเวคเตอร์ u และเวคเตอร์ v มาบวกกัน โดยการบวกไบนารีเป็นการทำ EX-OR ดังนั้น

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

เวคเตอร์รวมที่ได้ถ้านำมาหาเวทของแฮมมิ่งจะได้ว่า $\omega(\mathrm{u}+\mathrm{v})=5$
ดังนั้นพอสรุปได้ว่าระยะห่างแบบแฮมมิ่งระหว่างเวคเตอร์ u และ v จะเท่ากับเวทของแฮมมิ่งจากเวคเตอร์รวม กล่าวคือ

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

สำหรับรหัสเชิงเส้นในการหาระยะห่างของแต่ละคู่ของรหัสคำ ระยะห่างที่น้อยที่สุดเขียนย่อเป็น $d_{\min}$ ถ้า u และ v เป็นโค้ดเวคเตอร์สองชุดของรหัสเชิงเส้น โดย u+v ก็ยังเป็นโค้ดเวคเตอร์เพราะเซทของทุกโค้ดเวคเตอร์เป็นซับสเปช (Subspace) ของทุก n-ทูเปิ้ลส์ ดังนั้น จากคำนิยามที่ว่า ระยะห่างระหว่างโค้ดเวคเตอร์ทั้งสองคือ เวทของโค้ดเวคเตอร์ที่ 3 ก็จะได้ระยะห่างน้อยสุดของรหัสเชิงเส้นเท่ากับเวทต่ำสุดของโค้ดเวคเตอร์ที่ไม่เป็นศูนย์ ค่าระยะห่างน้อยสุด และเวทต่ำสุดจะเป็นตัวกำหนดความสามารถในการแก้รหัสบิทที่ผิดของรหัสเชิงเส้น[1]

พิจารณาจากรหัสที่ใช้ส่งโดยให้ $\mathrm{v}=(v_1,v_2,.....,v_n)$ เป็นโค้ดเวคเตอร์ที่ใช้ส่งและให้ $\mathrm{r}=(r_1,r_2,.....,r_n)$ เป็นเวคเตอร์ที่ได้รับจากการส่ง แต่เนื่องจากเวคเตอร์ r ที่รับได้จะเป็นอะไรก็ได้ใน 2^n เวคเตอร์ของ n-ทูเปิ้ลส์ ความแตกต่างระหว่าง r และ v คือ e

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

ซึ่ง e เป็นรูปแบบของรหัสที่ผิด (error pattern หรือ error vector) เมื่อ $e_i = r_i + v_i = 1$ นั้นก็หมายความว่าโค้ดเวคเตอร์เกิดความผิดผลาดตำแหน่งบิทที่ i^{th} แต่เนื่องจากในหนึ่งโค้ดเวคเตอร์มีรหัสอยู่ n บิท จึงทำให้เกิดความผิดผลาด $2^n -$ รูปแบบที่แตกต่างกัน ไม่นับรูปแบบที่มีทุกบิทเป็นศูนย์

ทางด้านรับตัวถอดรหัสมีหน้าที่ในการตรวจหาโค้ดเวคเตอร์ที่ส่งมาจากโค้ดเวคเตอร์ r ที่รับได้ สำหรับการถอดรหัสโดยใช้วิธีแม็กซิมั่มไลค์รีฮูดนั้น ตัวถอดรหัสจะตรวจสอบว่า v เป็นเวคเตอร์ที่ใช้ในการส่ง ซึ่งจะมีค่าเข้าใกล้เวคเตอร์ r โดยอาศัยการดูจากระยะห่างของแฮมมิ่ง ตัวถอดรหัสสามารถทำการแก้ไขรหัสที่ผิดจำนวน t บิทในโค้ดเวคเตอร์ที่รับเข้ามา โดยที่ $2t+2 \geq d_{\min} \geq 2t+1$ ตัวถอดรหัสสามารถทำการแก้ทุกรูปแบบที่ผิดไป t บิท จากเวคเตอร์ r ที่รับได้ ซึ่งแสดงให้เห็นได้ดังนี้ ให้ v เป็นโค้ดเวคเตอร์ที่ต้องการส่งและ u เป็นโค้ดเวคเตอร์ใดๆ ระยะห่างของแฮมมิ่งระหว่าง u,v และ r จะต้องเป็นไปตามสมการข้างล่างนี้

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

ถ้าสมมุติว่าเกิดรหัสผิดไป t' บิท $(t' \leq t)$ ดังนั้นระยะห่างของแฮมมิ่งระห่างโค้ดเวคเตอร์ที่ส่ง v กับโค้ดเวคเตอร์ที่รับ r คือ $d(\mathrm{v},\mathrm{r}) = t'$ แต่ $d(\mathrm{u},\mathrm{v}) \geq d_{\min} \geq 2t+1$ สมการที่ 2.16 จะให้

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

จากสมการที่ 2.3 แสดงให้เห็นว่ารูปแบบของรหัสที่มีบิทผิดไป t บิทหรือน้อยกว่า เวคเตอร์ r ที่รับได้จะเข้าใกล้โค้ดเวคเตอร์ v กว่าโค้ดเวคเตอร์ u ดังนั้นตัวถอดรหัสจะสามารถแก้ไขรหัสผิดได้ถูกต้องตามความผิดพลาดของบิทที่ผิดไป ตัวถอดรหัสไม่สามารถจะแก้ทุกรูปแบบที่ผิดไป ℓ บิท เมื่อ $\ell \geq t+1$ โดยปกติแล้วการแก้รหัสผิดของโค้ดเชิงเส้นจะทำได้เมื่อ

$$t = \left(d_{\min} - 1\right)/2 \tag{2.4}$$

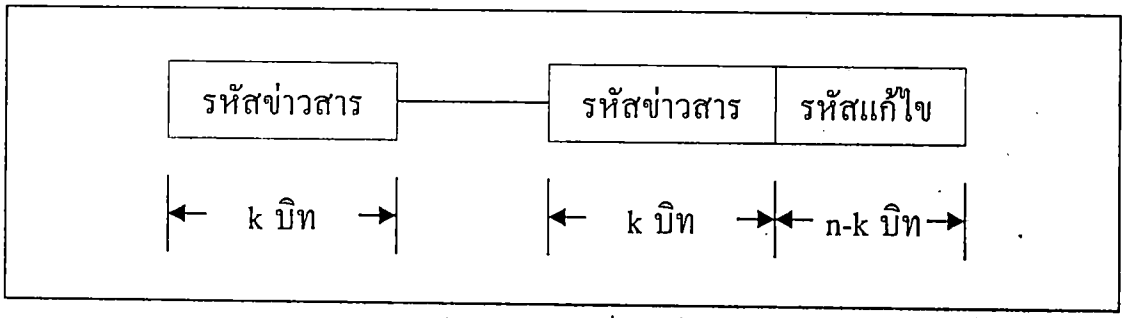
โดย $t = \left(d_{\min} - 1\right)/2$ เป็นค่าจำนวนเต็มไม่คิดทศนิยม และสามารถตรวจสอบรหัสผิดที่เกิดขึ้นถึง $\left(d_{\min} - 1\right)$ บิทในแต่ละรหัสคำ[1]

บล็อคโค้ดเชิงเส้น (Linear block code)

จากรหัสข่าวสาร (Massage) ที่แต่ละบล็อคมีขนาด k บิท ซึ่งพบว่าจะให้ข่าวสารที่แตกต่างกันได้ถึง $2^k - 1$ ข่าวสาร (ยกเว้นบล็อคที่มีรหัสข่าวสารเป็นศูนย์หมดจะไม่มีการนำมาใช้) แต่ละบล็อคข่าวสารจะถูกนำมาเข้ารหัสเป็นบล็คขนาด n บิท โดยจะมี $n-k$ บิท ที่เพิ่มเข้าไปให้รหัสข่าวสาร บิทเหล่านี้ที่เพิ่มเข้าไปในแต่ละบล็อคจะเป็นพาริตี้หรือบางทีเรียกว่ารหัสแก้ไข ที่จะถูกนำมาใช้ในการตรวจสอบและตรวจสอบบิทที่ผิดไป และค่าของ $n-k$ บิทจะขึ้นกับรหัสข่าวสารโดยตรง

บล็อคข่าวสารขนาด n บิทที่ได้จากการเข้ารหัสนี้จะเรียกว่ารหัสคำ (Code word) ถ้าหากว่ารหัสคำมีบิทของข่าวสารเดิมปรากฏอยู่ใน k บิทเริ่มต้นรหัสนั้น จะเรียกว่าซิสเต็มเมติกซ์ (Systematic code) ยิ่งไปกว่านั้นถ้าหากแต่ละรหัสคำจากจำนวน 2^k รหัสคำที่เข้ารหัสไว้ เกิดจากการรวมกันของ k เวคเตอรของรหัสแบบอิสระเชิงเส้น (linearly independent) รหัสดังกล่าวจะเรียกว่ารหัสบล็อคโค้ดเชิงเส้น [11] รูปแบบของบล็อคโค้ดเชิงเส้นแสดงได้ดังภาพที่ 4

ภาพที่ 4



แสดงรูปรหัสคำของบล็อคโค้ดเชิงเส้น

แมทริกซ์ตัวกำเนิด (Generator matrix)

สำหรับซับสเปซ S ของ V_n และแต่ละ n-ทูเปิ้ลส์ (Tuples) ของ S เป็นการรวมแบบเชิงเส้นของ $v_1, v_2, ..., v_k$ กล่าวคือ

$$u = m_1 v_1 + m_2 v_2 + ... + m_k v_k \qquad (2.5)$$

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

รหัสคำโดยเซ็ทของ k โค้ดเวคเตอร์ที่เป็นข้อกำหนดอิสระเชิงเส้น ถ้าจัด k รหัสคำเป็นอิสระต่อกันได้แมทริกซ์ $k \times n$

$$G = \begin{bmatrix} \mathrm{v}_1 \\ \mathrm{v}_2 \\ \vdots \\ \mathrm{v}_k \end{bmatrix} = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & & & \\ v_{k1} & v_{k2} & \cdots & v_{kn} \end{bmatrix} \tag{2.6}$$

เมื่อ $\mathrm{v}_i = (v_{i1}, v_{i2}, v_{i3}, \ldots\ldots, v_{in})$ สำหรับ $i = 1,2,\ldots\ldots,k$ ให้ $\mathrm{m} = (m_1, m_2, \ldots\ldots\ldots, m_k)$ เป็นบล็อคของข่าวสาร รหัสคำจะได้จาก

$$\begin{aligned} \mathrm{u} &= \mathrm{m}G \\ &= (m_1, m_2, \ldots\ldots\ldots, m_k) \begin{bmatrix} \mathrm{v}_1 \\ \mathrm{v}_2 \\ \vdots \\ \mathrm{v}_k \end{bmatrix} \\ &= m_1\mathrm{v}_1 + m_2\mathrm{v}_2 + \ldots\ldots\ldots + m_k\mathrm{v}_k \end{aligned} \tag{2.7}$$

ดังนั้นรหัสที่สอดคล้องกับชุดข่าวสาร $(m_1, m_2, \ldots\ldots, m_k)$ เกิดจากการรวมแบบเชิงเส้นของแถวใน G กลุ่มแถวต่างๆ ของแมทริกซ์ G จะเป็นตัวผลิตรหัสเชิงเส้น และเราเรียกแมทริกซ์ G ว่าแมทริกซ์ตัวกำเนิดของรหัส รหัสเชิงเส้นที่กล่าวนี้เรียกว่ารหัส (n,k) โดยในแต่ละบล็อคจะมีข่าวสารอยู่ k บิท ที่ถูกเข้ารหัสเป็นรหัสคำที่มีความยาวขนาด่ n บิท

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

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

โดย P_{ij} = 1 หรือ 0 ให้ I_k เป็นแมทริกซ์เอกลักษณ์ (Identity matrix) ขนาด $k \times k$ และให้ P เป็นแมทริกซ์ขนาด $k \times (n-k)$ ที่มีอิลิเมนท์เป็น P_{ij} ดังนั้นแมทริกซ์ตัวกำเนิดของรหัสระบบเขียนใหม่ได้เป็น

$$G = [I_k \vdots P]$$

พิจารณาถึงบล็อคของข่าวสาร $m = (m_1, m_2, \ldots\ldots, m_k)$ เมื่อได้แมทริกซ์ตัวกำเนิดของสมการ (2.8) จะได้โค้ดเวคเตอร์เป็น

$$\begin{aligned} u &= (u_1, u_2, u_3, \ldots\ldots, u_n) \\ &= (m_1, m_2, \ldots\ldots, m_k)G \end{aligned}$$

$$= (m_1, m_2, \ldots\ldots, m_k) \begin{bmatrix} 1 & 0 & 0 & 0 & \ldots & 0 & p_{11} & p_{12} & \cdots & p_{1k} \\ 0 & 1 & 0 & 0 & \ldots & 0 & p_{21} & p_{22} & \cdots & p_{2k} \\ \vdots & & & & & & \vdots & & & \\ 0 & 0 & 0 & 0 & \ldots & 1 & p_{k1} & p_{k2} & \cdots & p_{k,n-k} \end{bmatrix} \quad (2.9)$$

จากการคูณของแมทริกซ์จะได้

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

และ

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

สำหรับ $j = 1,2,\ldots\ldots, n-k$ จากสมการที่ 2.10a และ 2.10b จะพบว่ารหัส k บิทแรกของรหัสคำ คือ รหัสของข่าวสาร ส่วน $(n-k)$ บิทหลังเป็นฟังก์ชันเชิงเส้นของรหัสข่าวสารซึ่งเรียกว่ารหัสแก้ไข $(n-k)$ บิทของ u หรือ รหัสตรวจสอบพาริตี้ (Parity check digits) ของรหัสคำ สมการที่ 2.10b เรียกว่าสมการพาริตี้ของรหัส

แมทริกซ์ในการตรวจสอบพาริตี้ (Parity check matrix) [8]

จากที่กล่าวว่าแมทริกซ์ G ขนาด $k \times n$ จะมีแมทริกซ์ H ขนาด $(n-k) \times n$ ซึ่งโรว์สเปซของ G จะตั้งฉากอยู่กับ H อินเนอร์โปรดักท์ของเวคเตอร์ในโรว์สเปซของ G กับแถวของ H จะเป็นศูนย์

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

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

$$uH^T=\begin{pmatrix}0 & 0 & \cdots & 0\end{pmatrix} \tag{2.12}$$

หรือ

$$u.h_i=u_ih_{i1}+u_2h_{i2}+.....+u_nh_{in}=0 \tag{2.13}$$

สำหรับ $u=(u_1,u_2,.....,u_n)$ จึงสรุปได้ว่า u จะเป็นรหัสคำที่ได้จาก G ถ้าเพียงแต่ $uH^T=0$ แมทริกซ์ H นี้เรียกว่าแมทริกซ์ในการตรวจสอบพาริตี้ หรือเรียกย่อๆ ว่าพาริตี้แมทริกซ์ ถ้าแมทริกซ์ตัวกำเนิดของรหัสได้มาจากสมการที่ 2.8 พาริตี้แมทริกซ์ของรหัสคือ

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

$$=\left[P^TI_{n-k}\right]$$

P^T เป็นทรานสโพส (Transpose) ของแมทริกซ์ P สมการพาริตี้ 2.10b ได้จากแมทริกซ์ H นั้นคือ $u=(u_1,u_2,.....,u_n)$ เป็นรหัสคำของรหัสข้อมูล $m=(m_1,m_2,.....,m_k)$ เมื่อ $u_i=m_i$ สำหรับ $i=1,2,....,k$ แต่

$$uH^T=0$$

จะได้ว่า

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

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

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

ซีนโดรม

บล็อคของรหัสคำขนาด n บิท เมื่อทำการส่งออกไปในตัวกลางจะเกิดสัญญานรบกวนทำให้บางบิทของข้อมูลผิดไป ซึ่งรูปแบบของบิทที่ผิดไปบางครั้งเรียกว่าโคเซ็ท (Coset) มีหลายรูปแบบถ้าหาก u เป็นรหัสคำที่ต้องการส่ง โดย u มีระยะห่างต่ำสุดตามเงื่อนไขสมการ (2.4) และ e_ℓ เป็นรูปแบบของบิทที่ผิดไปในระหว่างการติดต่อสื่อสาร ทางด้านรับจะได้รับรหัสคำเป็น r กล่าวคือ

$$r = e_\ell + \underline{u} \qquad (2.16)$$

การคำนวนหาซีนโดรมของรหัสคำที่รับได้ทำได้โดย

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

ถ้าหาก u เป็นรหัสคำที่ถูกต้องจะพบว่า

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

ดังนั้นซีนโดรมคือ $S = e_\ell H^T$

โดยปกติแล้วแต่ละรูปแบบของบิทที่ผิดไปถ้าไม่ซ้ำกันจะให้ค่าซีนโดรมที่ไม่เท่ากัน ในการพิสูจน์ทำได้โดยถ้าสมมติว่ารูปแบบของบิทที่ผิดไปคนละรูปแบบแต่ให้ซีนโดรนเท่ากัน อย่างเช่นรูปแบบของบิทที่ผิดไป e_ℓ กับ e_lเมื่อซีโดรนคือ

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

$$S_2 = e_t H^T$$

แต่ $S_1 = S_2$ จะให้

$$e_\ell H^T = e_t H^T \tag{2.19}$$

หรือ

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

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

$$e_\ell - e_t = 0$$

ผลต่างของ e_ℓ กับ e_t จะเป็นศูนย์ก็ต่อเมื่อทุกบิทใน e_ℓ กับ e_t จะเหมือนกันแบบบิทต่อบิทจึงสรุปได้ว่า

$$e_\ell = e_t$$

ซึ่งขัดกับสมมติฐานที่ว่า e_ℓ กับ e_t เป็นคนละรูปแบบ

ดังนั้นพอสรุปได้ว่าถ้าหากรูปแบบบิทที่ผิดไปคนละรูปแบบจะให้ค่าซีนโดรมที่แตกต่างกันออกไป กล่าวคือ

$$e_\ell H^T \neq e_t H^T \tag{2.20}$$

ถ้าหากรหัสคำ (n,k) ถูกนำมาคำนวณหาซีนโดรม จะได้ซีนโดรมขนาด $n-k$ บิท ดังนั้นซีนโดรมที่แตกต่างกันจะมีจำนวน 2^{n-k} ค่า จะพบว่ารูปแบบของบิทที่ผิดไปหนึ่งรูปแบบกับซีนโดรนที่สอดคล้องจะเป็นแบบหนึ่งต่อหนึ่ง ปกติแล้วทางด้านรับจะมีตารางของซีนโดรมและรูปแบบของบิทที่ผิดที่สอดคล้องกันกับซีนโดรมเก็บเอาไว้ ดังนั้นทางด้านรับจะมีขั้นตอนการทำงาน 4 ขั้นตอนกล่าวคือ

1.คำนวณซินโดรมของรหัสคำ r ที่ได้รับจาก $S = rH^T$

2.เปิดตารางของซินโดรมเพื่อดึงเอารูปแบบที่ผิดที่ให้ซินโดรมเหมือนกับที่คำนวณได้ ถ้าหากเป็นรูปแบบของ e_ℓ

3.รหัสที่ถูกต้องคำนวณได้จาก $u = r + e_\ell$

4.ดึง k บิทแรกจากรหัสคำของ u ซึ่งจะเป็นรหัสข่าวสาร m ที่ส่งมา

ตัวอย่างเช่น ถ้าหากมีเมทริกซ์ตัวกำเนิด G เป็น

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

สามารถแปลงเป็นเมตริกซ์ตรวจสอบพาริตี้ H เป็น

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

ถ้าสมมติว่ารูปแบบรหัสที่ผิดไปเพียงหนึ่งบิท เป็น e_ℓ =[0,0,0,0,0,1] จะให้ซินโดรมเป็น

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

สำหรับรูปแบบต่างๆของรหัสที่ผิดไปเพียงบิทเดียวกับซินโดรมที่สอดคล้องพอสรุปได้ดังตารางข้างล่าง

ตารางที่ 4

ซีนโดรม	รูปแบบรหัสที่ผิดไปเพียงบิทเดียว
001	000001
010	000010
100	000100
110	001000
101	010000
011	100000

แสดงซีนโดรมที่ได้จากรูปแบบรหัสที่ผิดไปหนึ่งบิท

ถ้าสมมติว่าทางด้านส่งต้องการส่งรหัสข่าวสาร $m=[101]$ จะได้รหัสคำ u

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

เมื่อทำการส่งรหัสคำ u ผ่านช่องส่งสัญญาณที่มีการรบกวน จะพบว่าทางด้านรับได้รับรหัสคำ r เป็น $[1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1]$ ทางด้านรับจะทำการคำนวณหาค่าซีนโดรมจากรหัสค่า r ถ้าค่าซีนโดรมเป็นศูนย์หมดทุกบิทแสดงว่ารหัสคำ r ที่รับกับรหัสคำ u ที่ส่งเป็นรหัสคำเดียวกัน แต่ถ้าหากค่าซีนโดรมไม่เป็นศูนย์ก็ต้องทำการเปิดตารางที่ 4 เพื่อหารูปแบบรหัสที่ผิดไปที่สอดคล้องการคำนวณซีนโดรมทำได้โดย

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

ซีนโดรมที่ได้จะนำไปตรวจสอบกับตาราง พบว่ารูปแบบของรหัสที่ผิดไปคือ[0,0,1,0,0,0] ดังนั้นการแก้ไขรหัส r เพื่อให้ได้รหัสคำ u ที่ส่งมาทำได้โดย

$$u = r + e = [1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1] + [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0]$$

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

จากวิธีการดึงรหัสคำที่ถูกต้องกลับคืนมาจากรหัสคำที่ผิด จึงได้ถูกนำมาใช้เป็นหลักการเข้ารหัสลับให้กับข้อมูลในวิทยานิพนธ์ฉบับนี้ หลักการทำงานของการแก้รหัสลับคือ จากรหัสคำที่มีอยู่จะถูกนำมาบวกกับรูปแบบรหัสที่ผิดต่าง ๆ ที่กำหนดไว้ โดยแต่ละรูปแบบของรหัสที่ผิดจะมีซีนโดรมที่สอดคล้องเก็บเป็นตารางข้อมูลเอาไว้ ซึ่งรหัสคำที่ผิดมีหลายรูปแบบ ดังนั้นการบวกรหัสคำกับรูปแบบรหัสที่ผิด จะทำการเลือกรูปแบบของรหัสที่ผิดแบบสุ่มเทียม (Pseudo random) วิธีการนี้ทางด้านรับจะมีรหัสคำที่ผิดอยู่ตลอดเวลา แต่จะสามารถนำเอารหัสคำที่ถูกต้องกลับคืนมาได้ถ้าหากรู้ว่าแมทริกซ์ G คืออะไร ในการเพิ่มความซับซ้อนการเข้ารหัสลับ จะมีการเรียงลำดับตำแหน่งบิทในแต่ละรหัสคำที่ผ่านการบวกรูปแบบบิทที่ผิดไปเรียบร้อยแล้ว รูปแบบของการเรียงสลับบิทที่ผิดจะมีอยู่หลายรูปแบบเช่นกัน การเลือกรูปแบบการเรียงสลับบิทจะใช้ค่าของซีนโดรมเป็นตัวเลือก สำหรับรายละเอียดในการเข้ารหัสลับจะได้กล่าวในบทถัดไป

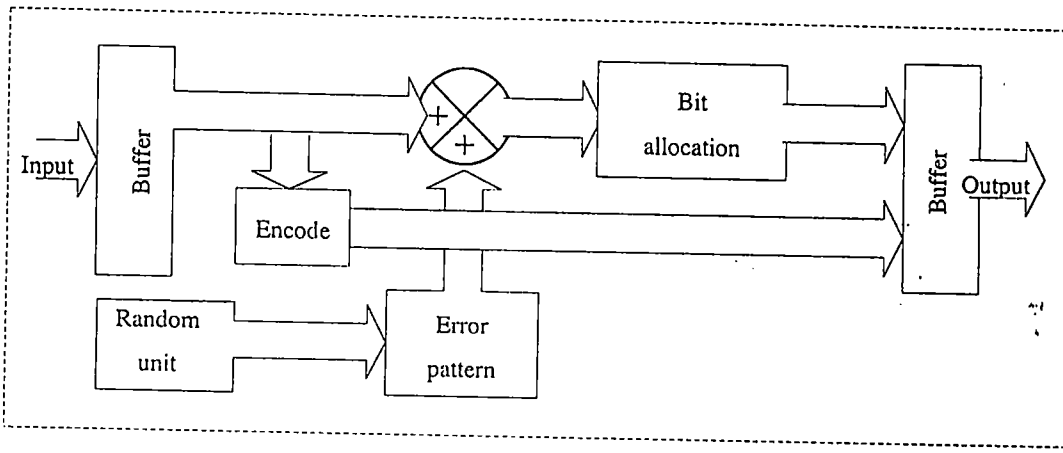
บทที่ 3

การออกแบบตัวเข้ารหัสและตัวถอดรหัสลับ

ในการออกแบบตัวเข้ารหัสและถอดรหัสลับแบบดิจิตอลนี้ สัญญาณที่จะนำมาป้อนเข้าทางอินพุทสำหรับทำการเข้ารหัสจะต้องเป็นสัญญาณที่อยู่ในรูปของสัญญาณดิจิตอล แต่ถ้าอินพุทที่ถูกป้อนเข้ามาอยู่ในรูปของสัญญาณแอนะล็อก จะต้องทำการแปลงสัญญาณเหล่านั้นให้อยู่ในรูปของสัญญาณดิจิตอล ก่อนที่จะทำการส่งเข้าทางด้านอินพุทของตัวเข้ารหัส ส่วนทางด้านตัวถอดรหัสลับก็จะได้รับข้อมูลที่ถูกส่งผ่านเข้ามาทางอินพุทที่อยู่ในรูปสัญญาณดิจิตอลเช่นกัน และทำการถอดรหัสกลับแต่ถ้ารูปสัญญาณเดิมเป็นแบบแอนะล็อกก็จะต้องใช้ตัวแปลงสัญญาณดิจิตอลกลับเป็นสัญญาณแอนะล็อกกลับคืนมาเช่นเดิม ในส่วนนี้จะกล่าวถึงวิธีการเฉพาะส่วนของตัวเข้ารหัสลับและตัวถอดรหัสลับเท่านั้น

การออกแบบตัวเข้ารหัสลับ

ภาพที่ 5



แสดงแผนภาพการเข้ารหัสลับ

ในการออกแบบตัวสร้างรหัสลับจะเป็นการนำข่าวสารข้อมูลที่เข้ามาทางด้านอินพุทมาทำการแปลงให้มีรูปแบบของข้อมูลแตกต่างไปจากข้อมูลเดิม โดยทำการแปลงข้อมูลข่าวสารที่เข้ามาทางด้านอินพุทที่มีขนาด k บิท (ในที่นี้ใช้ 8 บิท) ให้เป็นรหัสคำ ที่มีขนาด n บิท (12 บิท) แล้วทำการบวกรหัสคำที่ได้กับรูปแบบผิดพลาดที่ทำการสร้างขึ้นจำนวน 15 รูปแบบหลังจากนั้นจะทำการสลับตำแหน่งของรหัสคำซึ่งมีรูปแบบการสลับตำแหน่ง 16 รูปแบบที่ทำการบวกรูปแบบผิดพลาดแล้ว ก็จะทำให้ได้รหัสข้อมูลลับที่ทำการเข้ารหัสแล้ว ซึ่งสามารถแบ่งเป็นส่วนๆ ของการสร้างรหัสลับดังแสดงในภาพที่ 5 ได้ดังนี้

ส่วนของการเข้ารหัส (Encoder)

เป็นส่วนที่ทำหน้าที่ในการแปลงรหัสข้อมูลข่าวสารที่เข้ามาทางอินพุทที่มีขนาดของข้อมูล k บิท มาทำการแปลงให้เป็นรหัสคำ ในการแปลงนี้จะใช้วิธีการของรหัสบล็อคโค้ดเชิงเส้น โดยจะเป็นการนำรหัสข่าวสาร $\underline{\mathrm{m}}$ ที่มีขนาด k บิทไปทำการคูณกับแมทริกซ์ตัวกำเนิด G (Generator matrix) จะได้ผลลัพธ์เป็นรหัสคำที่มีขนาด n บิท จะมีจำนวนของบิทที่เพิ่มขึ้นมาจากรหัสของข้อมูลข่าวสาร $(n-k)$ บิท (4 บิท) โดยรหัสที่เพิ่มขึ้นมานี้จะถูกนำมาใช้ในการแก้รหัสผิดซึ่งจะกล่าวในหัวข้อต่อไป

ภายในแมทริกซ์ตัวกำเนิด G จะประกอบไปด้วยแมทริกซ์เอกลักษณ์ I_k ที่มีขนาด $k \times k$ และแมทริกซ์พาริตี้ P ที่มีขนาด $k \times (n-k)$ จะแสดงได้ในสมการที่3.1

$$\mathrm{G} = [\mathrm{I}_k : \mathrm{P}] \tag{3.1}$$

แมทริกซ์ตัวกำเนิด G ที่ใช้แสดงในภาพที่ 6

ภาพที่ 6

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

แสดงแมทริกซ์ตัวกำเนิด

จากรหัสข้อมูลข่าวสาร $\underline{m}$ ที่มีขนาด 8 บิท นำมาทำการคูณกับแมทริกซ์ตัวกำเนิด G จะได้รหัสคำมีขนาด 12 บิท ดังแสดงในสมการที่ 3.2

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

ทำให้สามารถแสดงวิธีการสร้างรหัสคำ u ในภาพที่ 7

ภาพที่ 7

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

แสดงการสร้างรหัสคำ

ส่วนของรูปแบบผิดพลาด (Error Pattern)

สำหรับแมทริกซ์ตัวกำเนิดในภาพที่ 6 นั้นจะแก้รหัสผิดได้เพียงบิทเดียวแต่ในการเข้ารหัสลับนี้จะใช้รูปแบบของบิทที่ผิดไปสองบิท ใน 8 บิทแรกของรหัสคำซึ่งเป็นบิทของข่าวสาร ดังนั้นจึงไม่เป็นไปตามกฎของซีนโดรม กล่าวคืออาจมีรูปแบบของบิทที่ผิดไป มากกว่าหนึ่งรูปแบบที่ให้ซีนโดรมเหมือนกัน ดังแสดงในตารางที่ 5 เนื่องจากการเข้ารหัสบล็อคโค้ดเชิงเส้น (12,8) จะมีซีนโดรมได้ 4 บิท หรือมีซีนโดรมได้ 15 รูปแบบที่แตกต่างกัน (ไม่นับซีนโดรมที่ทุกบิทเป็นศูนย์หมด) เนื่องจากรหัสที่ผิดไปสองบิทในแปดบิทแรก จะมีรูปแบบที่ไม่เหมือนกันได้เป็น 28 รูปแบบดังแสดงในคอลัมน์ที่สองของตารางที่ 5 ในกรณีที่รูปแบบบิทที่ผิดมีหลายรูปแบบจะเลือกเอารูปแบบบิทที่ผิดที่ให้ค่าสูงที่สุด อย่างเช่นรูปแบบ (3,5) ซึ่งมีบิทที่ผิดคือบิทที่ 3 กับ 5 จาก 8 บิทแรกของรหัสคำ กับรูปแบบ (2,7) ซึ่งมีบิทที่ผิดคือ บิทที่ 2 กับบิทที่ 7 ใน 8 บิทแรกของรหัสคำ จะพบว่ารูปแบบ (2,7) มีค่าสูงสุดกว่า (3,5) จึงเลือกรูปแบบ (2,7) เนื่องจากการเข้ารหัสลับสัญญาณเสียง ถ้าเลือกบิทที่ผิดมีค่าสูงจะทำให้แอมปลิจูด (Amplitude) ของสัญญาณเสียงที่รวมกับรหัสที่ผิดแล้วเกิด

การเปลี่ยนแปลงสูงดังนั้น รูปแบบบิทที่ผิดที่ได้เลือกไว้ และซีนโดรมที่สอดคล้องพอสรุปได้ดังตารางที่ 5

$$r = u \oplus e \tag{3.3}$$

ตารางที่ 5

ค่าซีนโดรม	ตำแหน่งบิทที่ผิด	เลือกรูปแบบ	รูปแบบบิทที่ผิด
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 โดยรูปแบบของสัญญาณแบบสุ่มเทียมใช้พริมิทีฟโพลิโนเมียล (Primitive polynomial) อันดับสี่ คือ $X^4 + X + 1$ ทำให้ได้รูปแบบของการเรียงลำดับเป็นดังตารางที่ 6

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

ส่วนของการสลับตำแหน่งบิท (Bit allocation)

เป็นการสลับตำแหน่งของข้อมูลที่เป็นรหัสคำที่ทำการบวกกับรูปแบบผิดพลาดแล้วโดยจะทำการสลับเฉพาะ 8 บิทแรกเท่านั้น เพื่อเพิ่มความซับซ้อนในการป้องกันข้อมูลให้มีความปลอดภัยยิ่งขึ้น โดยการนำ 4 บิทหลังที่ได้จากรหัสคำมาเป็นตัวเลือกรูปแบบของการสลับตำแหน่งบิทข้อมูล ซึ่งมีรูปแบบการสลับบิท 16 รูปแบบดังแสดงในตารางที่ 7

การออกแบบตัวถอดรหัสลับ

ในการออกแบบตัวถอดรหัสลับเป็นการนำข้อมูลข่าวสารที่ผ่านการเข้ารหัสลับแล้ว กลับคืนมาให้ได้ข้อมูลเดิม โดยทำการถอดรหัสข้อมูลข่าวสาร จากข่าวสารที่ทำการเข้ารหัสลับแล้วมีขนาด n บิท เข้ามาทางอินพุทโดยเริ่มจากการสลับตำแหน่งของข้อมูลกลับ ให้ข้อมูลที่ได้มีตำแหน่งเหมือนเดิม โดยใช้ข้อมูล 4 บิทที่เพิ่มเข้ามาเป็นตัวเลือกรูปแบบที่จะทำการสลับตำแหน่งบิทกลับ แล้วทำการคำนวณหาค่าซินโดรมที่จะมาเป็นตัวเลือกรูปแบบที่ผิดพลาดดังแสดงในตารางที่ 4 มาทำการบวกแบบเอกคลูซีฟออร์กับข้อมูลข่าวสารนั้น ทำให้ได้ข้อมูลข่าวสารเดิมกลับคืนมา แบ่งเป็นแต่ละขั้นตอนดังแสดงในภาพที่ 8

ตารางที่ 7

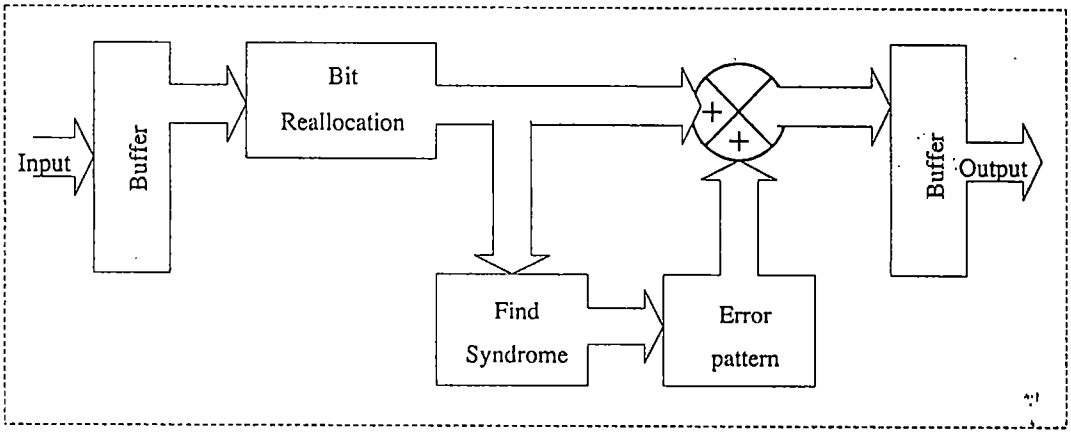
ลำดับรูปแบบ	ตำแหน่งบิทที่ทำการสลับ							
	LSB							MSB
0000	0	2	3	4	7	6	1	5
0001	2	4	6	1	0	3	7	5
0010	5	7	0	4	2	6	1	3
0011	6	4	0	3	7	5	1	2
0100	4	0	6	3	7	1	2	5
0101	5	1	6	2	4	3	7	0
0110	7	0	1	2	6	4	5	3
0111	6	2	1	0	7	3	4	5
1000	3	1	4	0	6	2	5	7
1001	2	5	4	1	6	3	0	7
1010	1	7	3	4	5	2	0	6
1011	5	1	6	2	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

การสลับตำแหน่งบิท

ส่วนของการสลับตำแหน่งบิทกลับ (Bit reallocation)

เป็นส่วนที่ทำหน้าที่ในการสลับตำแหน่งของบิทข้อมูลที่เป็นรหัสคำที่ผิดไปกลับคืนมาให้ได้ตำแหน่งที่ถูกต้องตรงกับตำแหน่งเดิม โดยจะทำการสลับตำแหน่งข้อมูลกลับเฉพาะ 8 บิทแรกที่ถูกสลับไว้เท่านั้น และ 4 บิทที่เหลือจะเป็นตัวเลือกรูปแบบของการสลับตำแหน่งข้อมูลกลับ ซึ่งจะมีรูปแบบการสลับตำแหน่งบิทข้อมูลดังแสดงในตารางที่ 8

ภาพที่ 8



แสดงแผนภาพการถอดรหัสลับ

ส่วนของการคำนวณหาซีนโดรม

ในส่วนของการคำนวณหาซีนโดรมประกอบไปด้วยแมทริกซ์ตรวจสอบพาริตี้ H ที่ถูกนำมาใช้ในการคำนวณหาค่าซีนโดรมสำหรับนำมาใช้ในการถอดรหัสที่ผิดไป โดยแมทริกซ์ตรวจสอบพาริตี้ H ประกอบขึ้นจากแมทริกซ์ทรานสโพสของแมทริกซ์พาริตี้ P ที่มีขนาด $(n-k)\times k$ และแมทริกซ์เอกลักษณ์ I_{n-k} คือ

$$\mathrm{H}=\left[\mathrm{P}^{\mathrm{T}}\mathrm{I}_{n-k}\right] \tag{3.4}$$

แมทริกซ์ตรวจสอบพาริตี้ H แสดงในภาพที่ 8

ภาพที่ 9

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

แสดงแมทริกซ์พาริตี้

ตารางที่ 8

ลำดับรูปแบบ	ตำแหน่งบิทที่ทำการสลับ							
	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

การสลับตำแหน่งบิทกลับ

ในการคำนวณหาซีนโดรม S ได้จากการนำรหัสคำที่ได้รับมาทำการคูณกับแมทริกซ์ทรานสโพสของ H ซึ่ง H^T เป็นแมทริกซ์โดย 4 แถวสุดท้ายของ H^T จะเป็นแมทริกซ์เอกลักษณ์ โดยใน 8 แถวแรกได้จากการทำทรานสโพสแมทริกซ์ P ของแมทริกซ์ตัวกำเนิด โดยทั้ง 8 แถวจะแตกต่างกัน และ r เป็นรหัสที่ได้รับเข้ามาซึ่งสามารถทำการคำนวณได้ดังแสดงในสมการที่ 3.5

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

ซีนโดรมสามารถทำการหาได้ดังภาพที่ 10

ภาพที่ 10

$$S = [r_0 \quad r_1 \quad \cdots \quad r_{10} \quad r_{11}] \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

แสดงการหาค่าซีนโดรม

แต่เนื่องจากแต่ละแถว (Row) ของแมทริกซ์ G และแมทริกซ์ H ตั้งฉากกันจะให้อินเนอร์โปรดัก (Inner product) เป็นศูนย์ ก็หมายความว่า $uH^T = 0$ นั่นเอง ทำการแทนค่า r ด้วยสมการที่ 3.5 ได้ว่า

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

จากสมการที่ 3.5 และ 3.6 เป็นตัวบอกให้ทราบว่าค่าซีนโดรมที่คำนวณได้จากรหัสคำที่ผิดไปเนื่องจากรูปแบบผิดพลาด e ที่คูณกับ H^T นั้นให้ค่าซีนโดรมเหมือนกันกับการนำรูปแบบผิดพลาด e ไปคูณโดยตรงกับ H^T ด้วยเหตุนี้ค่าของซีนโดรมจึงเป็นตัวบ่งบอกว่ารหัส r ที่ได้รับมีรูปแบบผิดพลาด e รูปแบบใดปรากฏอยู่ ดังนั้นรูปแบบการดึงรหัสคำที่ถูกส่ง u กลับคืนมาทำได้โดยการนำ e ไปทำการบวกเอกคลูซีฟออร์กับรหัส r อีกครั้งหนึ่ง ดังแสดงในสมการที่ 3.7 ซึ่งทำให้สามารถทำการถอดรหัสได้และได้ข้อมูลข่าวสารที่ถูกต้องกลับคืนมา

บทที่ 4
การประยุกต์ใช้อุปกรณ์ประเภทโปรแกรมเมเบิลลอจิกดีไวซ์

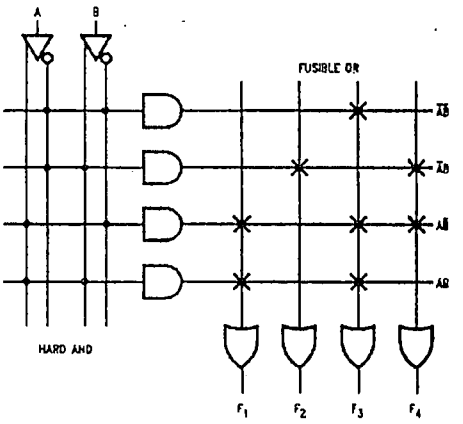
อุปกรณ์ประเภทโปรแกรมเมเบิลลอจิกดีไวซ์ (Programmable logic device) หรือ PLD ถูกผลิตขึ้นหลายบริษัทด้วยกันซึ่ง PLD เป็นอุปกรณ์ที่ถูกออกแบบมาใช้สำหรับทำงานตามสมการฟังก์ชันลอจิกที่ทำการออกแบบไว้ PLD แต่ละชนิดจะมีโครงสร้างภายในที่แตกต่างกัน ซึ่งเป็นสิ่งที่ผู้ออกแบบสมการฟังชันลอจิกต้องพิจารณาสำหรับการเลือกอุปกรณ์ประเภทนี้มาใช้งาน

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

พื้นฐานของโปรแกรมเมเบิลลอจิกดีไวซ์

การนำ PLD มาใช้ในการทำงานตามฟังก์ชันที่ทำการออกแบบวงจรดิจิตอล เริ่มต้นจากการระเบิดฟิวส์ (Fuse) ที่อยู่ภายในตัว PLD ซึ่งเป็นเทคโนโลยีประเภทเดียวกันกับที่ใช้ในอุปกรณ์จำพวก Programable read only memory (PROM) โดยโครงสร้างภายในของ (PROM) มีการดีโค้ดทางด้านอินพุท ในลักษณะเป็นแมทริกซ์ของเทอม AND แบบกำหนดรูปแบบตายตัว และสามารถทำการโปรแกรมลงไปได้ในแมทริกซ์เทอม OR ดังแสดงในภาพที่ 11 ซึ่งจะเป็นข้อเสียของการใช้ PROM มาเป็นอุปกรณ์ทางด้านลอจิก เนื่องจากวงจรทางด้านลอจิกมีความหลากหลายมาก จำนวนของโปรดัคเทอม (Product-term) ที่นำมาใช้คือ 2^n เมื่อ n คือ จำนวนตัวแปรทางด้านอินพุท ซึ่งเป็นขนาดของแมทริกซ์ที่จะต้องมีขนาดเป็นสองเท่าของ n เสมอ

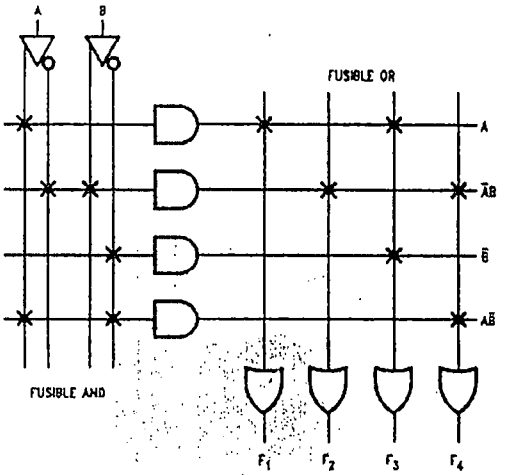
ภาพที่ 11



แสดงโครงสร้างของ PROM ที่โปรแกรมทางอาร์เรย์ OR

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

ภาพที่ 12

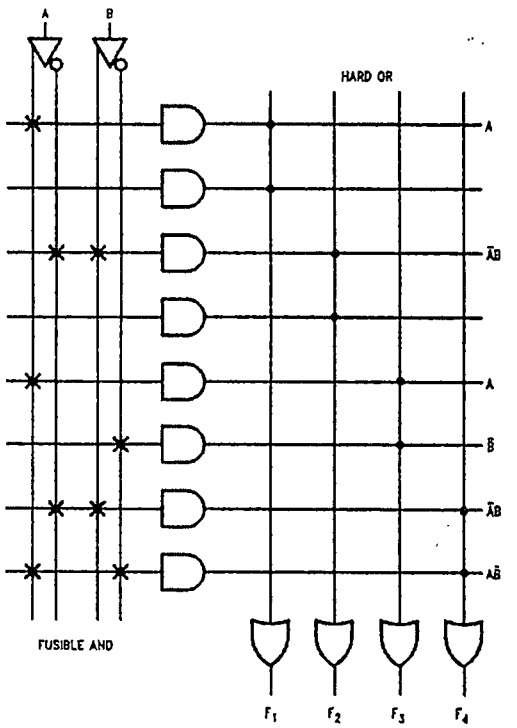


แสดงโครงสร้างของ PAL ที่ทำการโปรแกรมในอาร์เรย์ AND และ OR

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

โปรแกรมให้ทำงานตามฟังก์ชันลอจิกและสามารถทำการลบแล้วทำการโปรแกรมใหม่โดยเรียกว่า Generic array logic (GAL)

ภาพที่ 13



แสดงโครงสร้างของ PAL ที่ทำการโปรแกรมที่ อาร์เรย์ AND

การออกแบบฟังก์ชันลอจิกโดยใช้ PLD

เป็นการนำเอาอุปกรณ์ประเภท PLD มาใช้งานในส่วนต่างๆ สำหรับสร้างชุดตัวเข้ารหัสลับและตัวถอดรหัสลับเพื่อเป็นการประหยัดและสามารถลดขนาดของแผ่นวงจร ซึ่งจะทำการออกแบบสำหรับใช้งานในส่วนต่าง ๆ ดังนี้

ส่วนของการเข้ารหัส (Encoder)

เป็นการสร้างรหัสคำซึ่งจะเป็นไปตามสมการที่ 3.2 โดยที่ m คือรหัสข้อมูลข่าวสารที่เข้ามาทางอินพุทมีขนาด 8 บิท ที่นำมาทำการคูณกับแมทริกซ์ตัวกำเนิด G ส่วน u คือรหัสคำที่ปรากฏทางเอาท์พุทซึ่งจะมีขนาด 12 บิท ทำให้สามารถเขียนเป็นสมการคูณกันของข้อมูลข่าวสารที่เข้ามากับแมทริกซ์ตัวกำเนิดดังแสดงในสมการที่ 4.1

$$u_0 = m_0 \qquad u_1 = m_1$$
$$u_2 = m_2 \qquad u_3 = m_3$$
$$u_4 = m_4 \qquad u_5 = m_5$$
$$u_6 = m_6 \qquad u_7 = m_7$$
$$u_8 = m_1 \oplus m_2 \oplus m_4 \oplus m_6 \oplus m_7$$
$$u_9 = m_2 \oplus m_3 \oplus m_4 \oplus m_5 \oplus m_7$$
$$u_{10} = m_0 \oplus m_3 \oplus m_4 \oplus m_5 \oplus m_6$$
$$u_{11} = m_0 \oplus m_1 \oplus m_5 \oplus m_6 \oplus m_7 \qquad (4.1)$$

จากสมการที่ 4.1 จะเห็นได้ว่า $u_0 - u_7$ มีค่าเท่ากับ $m_0 - m_8$ และในส่วนของ $u_8 - u_{11}$ จะถูกนำมาใช้ในการสร้างสมการลอจิก สำหรับการโปรแกรมดังแสดงในภาพที่ 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 ซึ่งมีอินพุทสูงสุดถึง 8 อินพุท และมี 8 เอาท์พุท จากสมการลอจิกในภาพที่ 14 จะเห็นว่า EA1-EA4 เป็นฟังก์ชันที่ใช้ในการป้อนกลับเนื่องจาก GAL16V8 สามารถทำโปรแกรมโปรดัคเทอม (Product-term) ได้สูงสุดเพียง 7 เทอมต่อหนึ่งเอาท์พุท ทำให้สามารถเขียนสมการในการโปรแกรมลงบน GAL16V8 ได้ดังภาพที่ 14 และเอาท์พุทที่ได้คือ 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

เมื่อทำการแปลงโปรแกรม (Compile) สมการที่แสดงในภาพที่ 14 ด้วยซอฟแวร์ OPAL จะได้ไฟล์ข้อมูลนามสกุล .LOG ซึ่งเป็นแฟ้มข้อมูลที่ใช้สำหรับเก็บรายละเอียดของการใช้งาน GAL16V8 และตำแหน่งของขาสำหรับการนำไปใช้งานในแผงวงจรที่ออกแบบ ดังปรากฏในภาพที่ 15

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

ส่วนของรูปแบบผิดพลาด (Error pattern)

เป็นการออกแบบโดยนำเอา PLD มาทำหน้าที่ในการสร้างรูปแบบที่ผิดพลาดที่สามารถทำการเลือกได้ซึ่งจะปรากฏทางเอาท์พุทที่มีขนาด 8 บิทและมีอินพุทขนาด 4 บิทสำหรับเป็นตัวเลือกรูปแบบดังแสดงในมตารางที่ 5 เป็นอินพุท 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
```

แสดงรูปแบบสมการของส่วนรูปแบบผิดผลาด

จากสมการมีตัวแปร A,B,C,D เป็นอินพุทที่เป็นตัวเลือกรูปแบบผิดพลาดซึ่งมี A เป็นบิทสูงและ D เป็นบิทต่ำ ส่วนทางด้าน Y1-Y8 เป็นเอาท์พุทของรูปแบบผิดพลาดที่จะนำไปบวกแบบเอกคลูซีฟออร์กับข้อมูลข่าวสาร u มี Y1 เป็นบิทสูงและ Y8 เป็นบิทต่ำ ดังแสดงในภาพที่ 16 โดย

ทำการเลือก PLD เบอร์ PAL16L8 ซึ่งมี 8 อินพุทและมี 8 เอาท์พุทเป็นตัวทำหน้าที่ในการเก็บรูปแบบผิดพลาด

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 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 ในการนำมาใช้งานส่วนของรูปแบบผิดผลาด

เมื่อนำสมการที่สร้างขึ้นในภาพที่ 16 มาทำการแปลงโปรแกรมจะได้แฟ้มข้อมูลแสดงข้อมูลการใช้งานและตำแหน่งของขาในการนำมาใช้งานดังแสดงในภาพที่ 17 และแฟ้มข้อมูลที่เป็นรายละเอียดของแต่ละฟิลด์ที่ทำการระเบิดฟิลด์อาร์เรย์ภายในของ PAL16L8 และใช้ไฟล์นี้ในการโปรแกรมสำหรับกำหนดให้ทำงานตามฟังก์ชันลอจิกที่ต้องการ

ส่วนของการกำเนิดสัญญาณแบบสุ่มเทียม (Pseudo Random)

ใช้พริมิทีฟโพลิโนเมียล (Primitive Polynomial) อันดับที่สี่ โพลิโนเมียลที่ได้คือ $X^4 + X + 1$ จะให้ลำดับของรูปแบบดังแสดงในตารางที่ 6

จากตารางที่ 6 รูปแบบของบิทสามารถนำมาเขียน ให้อยู่ในรูปแบบของซีเควนซ์ (Sequence) ที่สามารถให้เอาท์พุทได้ตามตาราง สำหรับทำการเลือก PAL16R4 มาใช้ในการออกแบบสำหรับทำงานตามฟังก์ชัน โครงสร้างภายในของ PAL16R4 จะมีดีฟลิบฟลอบ (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  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

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 GND /EN NC NC Q4 Q3 Q2 Q1 NC NC VCC

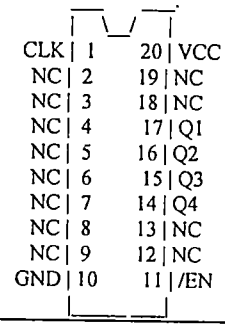
Pin	Label	Type
1	CLK	clock pin
2	NC	unused
3	NC	unused
4	NC	unused
5	NC	unused
6	NC	unused
7	NC	unused
8	NC	unused
9	NC	unused
10	GND	ground pin
11	EN	enable pin
12	NC	unused
13	NC	unused
14	Q4	neg,reg feedback
15	Q3	neg,reg feedback
16	Q2	neg,reg feedback
17	Q1	neg,reg feedback
18	NC	unused
19	NC	nused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



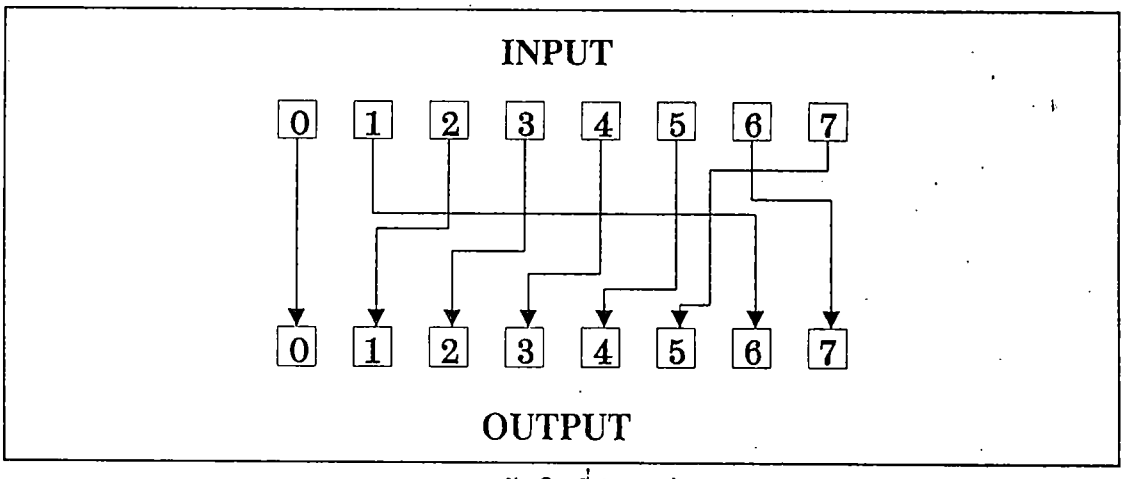
แสดงข้อมูลของ PAL16R4 ในการนำมาใช้งานส่วนการกำเนิดแบบสุ่ม

การนำสมการที่แสดงในภาพที่ 19 มาทำการแปลงโปรแกรมจะได้แฟ้มข้อมูลซึ่งจะเป็นไฟล์แสดงรายละเอียดในการใช้งาน PAL16R4 และแสดงตำแหน่งของขาสำหรับใช้งาน

ส่วนของการสลับตำแหน่งบิท (Bit allocation)

ในส่วนนี้เป็นการสลับตำแหน่งบิทของรหัสคำที่จะปรากฏทางเอาท์พุทตามรูปแบบที่ทำการกำหนดไว้ในตารางที่ 7 เช่นรูปแบบการสลับตำแหน่งของลำดับ 0000 เมื่อทำการสลับตำแหน่งจะแสดงได้ดังภาพที่ 20 ส่วนในลำดับอื่นๆ ก็เช่นเดียวกัน

ภาพที่ 20



แสดงการสลับบิทที่ตำแหน่ง 0000

ในการสลับตำแหน่งในตารางที่ 7 มีรูปแบบของการสลับอยู่ 16 รูปแบบที่ต้องทำการสลับ และแต่ละรูปแบบจะมีความแตกต่างกัน ข้อมูลอินพุทที่นำมาใช้ทั้ง 12 บิทนี้ มีเพียง 8 บิทแรกที่ถูกทำการสลับตำแหน่งส่วน 4 บิทหลังทำหน้าที่เลือกรูปแบบของการสลับตำแหน่งบิทข้อมูล 8 บิทแรก ซึ่งจะเป็นเอาท์พุท จากเงื่อนไขของอินพุทและเอาท์พุทนี้ทำให้พิจารณาเลือก GAL20V8 มาใช้งานซึ่ง GAL20V8 มีอินพุท 12 อินพุทและเอาท์พุท 8 เอาท์พุท และในการนำ GAL20V8 มาทำหน้าที่ในการสลับตำแหน่งบิท ต้องมีการใช้ GAL20V8 ถึงสองตัว เนื่องมาจากจำนวนของโปรดัคเทอมที่ใช้มีมากกว่า 7 เทอมต่อหนึ่งเอาท์พุท ทำให้ต้องนำเอาวิธีการป้อนกลับมาใช้สำหรับการสร้างสมการที่ทำการโปรแกรมลงบน GAL20V8 ซึ่งต้องใช้การป้อนกลับถึง 4 เอาท์พุทของ GAL20V8 แต่ละตัว ฉะนั้นในการสลับตำแหน่งบิทจึงต้องใช้ GAL20V8 ถึงสองตัว โดยมีตัวแรกทำการสลับตำแหน่งบิทที่ 0-3 และตัวที่สองทำการสลับตำแหน่งบิทที่ 4-7 ทำให้สามารถเขียนเป็น

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

ภาพที่ 21

```
BIT ALOCATION_A
CHIP ALOCATION 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*E3
  +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
```

แสดงรูปแบบสมการส่วนของการสลับตำแหน่งบิท 0-3

ภาพที่ 22

```
BIT ALOCATION_B
CHIP ALOCATION 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
```

แสดงรูปแบบสมการส่วนของการสลับตำแหน่งบิท 4-7

เมื่อนำเอาสมการในภาพที่ 21 และภาพที่ 22 มาทำการแปลงโปรแกรมจะได้แฟ้มข้อมูลที่แสดงการ 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%)
                    ------------------------------------
                    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 | 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 ในการนำมาใช้งานส่วนการสลับตำแหน่งบิท 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 ในการนำมาใช้งานส่วนการสลับตำแหน่งบิท 4-7

ส่วนของการสลับตำแหน่งบิทกลับ (Bit reallocation)

เป็นส่วนที่ทำหน้าที่ในการสลับตำแหน่งของบิทข้อมูลกลับ ทำงานในลักษณะตรงข้ามกับการสลับตำแหน่งบิทข้อมูลเพื่อให้ได้ข้อมูลที่มีตำแหน่งถูกต้องกลับคืนมา มีรูปแบบการสลับตำแหน่งบิทกลับตามตารางที่ 8

ในการเลือก PLD และการสร้างสมการลอจิกจากตารางการสลับบิทกลับ พิจารณาทางด้านอินพุทของการสลับบิทกลับจะมีอยู่ 12 อินพุท คือ มีข้อมูลที่ต้องการสลับตำแหน่ง 8 บิท และบิทที่เป็นตัวเลือกรูปแบบการสลับตำแหน่งบิทกลับ 4 บิท ส่วนเอาท์พุทที่ปรากฏจะมีขนาด 8 บิท เช่นเดียวกับการสลับตำแหน่งบิททำให้พิจารณาเลือก GAL20V8 ซึ่งมีอินพุท 12 อินพุทและมีเอาท์พุท 8 เอาท์พุท แต่ต้องใช้ GAL20V8 ถึงสองตัวในการสลับบิทกลับเนื่องจากจำนวนของโปรดัคเทอม

ของสมการที่ได้จากตารางที่ 8 มีจำนวนของเทอมมาก ทำให้ต้องใช้ GAL20V8 ถึงสองตัวโดยให้ตัวแรกทำการสลับตำแหน่งบิท 0-3 กลับ และตัวที่สองทำการสลับตำแหน่งบิท 4-7 กลับ และใช้เอาท์พุทที่เหลืออีก 4 เอาท์พุทของ GAL20V8 แต่ละตัวมาทำหน้าที่ในการป้อนกลับเพื่อให้สามารถใช้จำนวนโปรดัคเทอมได้มากกว่า 7 เทอม ดังแสดงในภาพที่ 25 และภาพที่ 26

ภาพที่ 25

```
RE_BIT ALOCATION_A
CHIP RE_ALOCATION 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*/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
```

แสดงรูปแบบของสมการส่วนของการสลับบิทกลับบิท 0-3

ภาพที่ 26

```
BIT REALOCATION_B
CHIP BIT_REALOCATION 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
```

แสดงรูปแบบสมการส่วนของการสลับบิทกลับบิท 4-7

จากสมการในภาพที่ 25 และ 26 มี SD0-SD3 เป็นตัวป้อนกลับของ GAL20V8 ตัวแรก และ SD0-SD4 เป็นตัวป้อนกลับของ GAL20V8 ตัวที่สอง สำหรับเพิ่มโปรดัคเทอมในการโปรแกรมมีเอาท์พุท D0-D7 สำหรับ GAL20V8ตัวแรกให้เอาท์พุท D0-D3 ส่วน GAL20V8 ตัวที่สองให้เอาท์พุทเป็น 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%)
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 | 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 ในการนำมาใช้งานส่วนการสลับตำแหน่งบิทกลับบิท 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%)
```

```
-------------------------------------------
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%)
-------------------------------------------
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| 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 ในการนำไปใช้งานส่วนการสลับตำแหน่งบิทกลับบิท 4-7

ภาพที่ 27 และ 28 ซึ่งในแฟ้มข้อมูลนี้จะบอกรายละเอียดในการใช้งาน GAL20V8 และตำแหน่งของขาที่กำหนดใช้งาน

ส่วนของการคำนวณหาค่าซีนโดรม (Syndrome)

ส่วนการหาค่าที่ใช้ในการเลือกรูปแบบที่ผิดพลาดมาทำการบวกแบบเอกคลูซีฟออร์กับข้อมูลข่าวสารจากการเข้ารหัส เพื่อให้ได้ข้อมูลที่ถูกต้องกลับคืนมา จากสมการที่ 3.5 สำหรับการคำนวณหาซีนโดรมสามารถนำมาเขียนเป็นสมการรูปของลอจิกได้ดังนี้

$$
\begin{aligned}
S_0 &= r_1 \oplus r_2 \oplus r_4 \oplus r_6 \oplus r_7 \oplus r_8 \\
S_1 &= r_2 \oplus r_3 \oplus r_4 \oplus r_5 \oplus r_7 \oplus r_9 \\
S_2 &= r_0 \oplus r_3 \oplus r_4 \oplus r_5 \oplus r_6 \oplus r_{10} \\
S_3 &= r_0 \oplus r_1 \oplus r_5 \oplus r_6 \oplus r_7 \oplus r_1
\end{aligned}
\qquad (4.2)
$$

จากสมการที่ 4.2 เป็นการคำนวณหาค่าของซินโดรม S0-S3 สามารถนำมาเขียนเป็นสมการลอจิกสำหรับโปรแกรมลงใน PLD ให้ทำงานตามฟังก์ชันดังแสดงในภาพที่ 29 และในภาพที่ 30 โดยมี 12 อินพุท คือ A1-A12 และ 4 เอาท์พุท คือ S0-S3 แต่เนื่องจากการแปลงสมการ 4.2 มีจำนวนของโปรดัคเทอมมากทำให้ต้องเลือกใช้ GAL จำนวน 2 ตัว ตัวแรกใช้ GAL20V8 ซึ่งสามารถใช้ได้ 12 อินพุท และใช้ 8 เอาท์พุท โดยใช้ 4 เอาท์พุทเป็นตัวป้อนกลับสำหรับเพิ่มจำนวนโปรดัคเทอมแต่ยังมีจำนวนโปรดัคเทอมที่เหลือจึงใช้ GAL16V8 เพิ่มอีกหนึ่งตัวสำหรับทำการหาซินโดรม โดยรับอินพุทจาก GAL20V8 จะให้เอาท์พุทเป็น 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 SYDROME 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%)
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      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 ในการนำมาใช้งานส่วนหาซีนโดรม 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 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 ในการนำมาใช้งานส่วนหาซีนโดรม 2

ภาพที่ 31 และ ภาพที่ 32 ซึ่งเป็นไฟล์แสดงรายละเอียดของการใช้งานและตำแหน่งของขา สำหรับการใช้งาน

บทที่ 5

การทดลองและผลการทดลอง

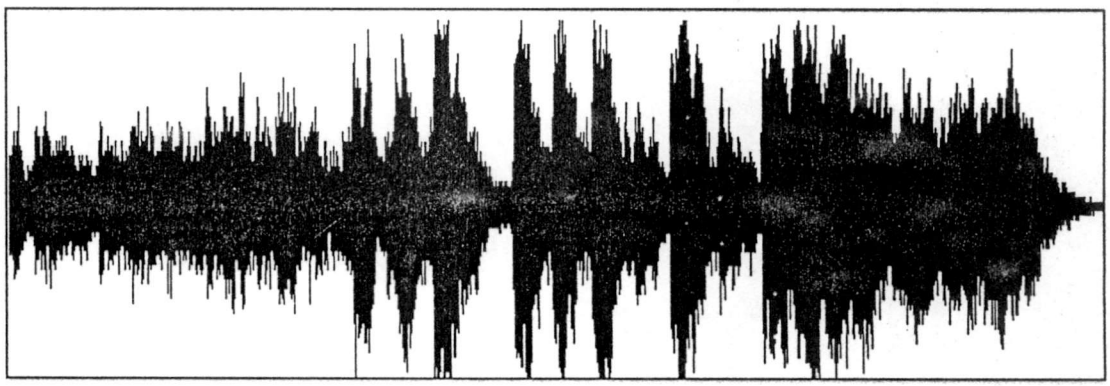
ผลการทดลอง

การทดลองแผงวงจรตัวเข้ารหัสลับและตัวถอดรหัสลับกระทำโดยนำเอาสัญญาณที่อยู่ในรูปแบบต่าง ๆ มาทำการเข้ารหัสและถอดรหัสลับดังนี้ สัญญาณแอนะล็อกได้แก่ สัญญาณเสียงดนตรีและสัญญาณเสียงพูด และสัญญาณดิจิตอล

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

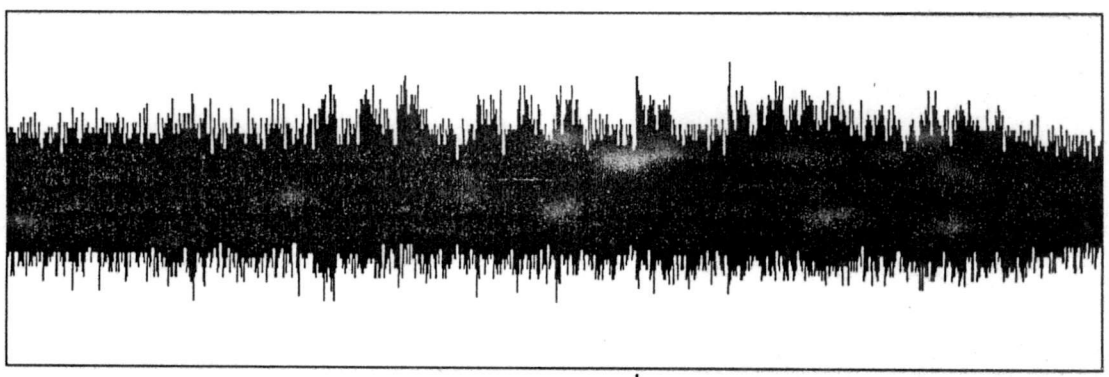
จากรูปของสัญญาณในภาพที่ 33c ซึ่งเป็นสัญญาณของเสียงดนตรี เปรียบเทียบกับรูปร่างของสัญญาณในภาพที่ 33b ที่ผ่านการเข้ารหัส แล้วจะเห็นว่ารูปร่างของสัญญาณมีความแตกต่างไปจากสัญญาณต้นแบบในภาพที่ 33a และมื่อทำการถอดรหัสกลับ จะได้รูปแบบของสัญญาณดังแสดงในภาพที่ 33c กลับมา สังเกตเห็นว่ามีรูปแบบของสัญญาณมีความคล้ายคลึงกับข้อมูลต้นแบบจะมีส่วนแตกต่างไปบ้างอันเนื่องจากการสูญเสียในระหว่างการแปลงสัญญาณจากแอนะล็อกเป็นดิจิตอล เพราะว่าสัญญาณความถี่ของเสียงดนตรีค่อนข้างสูงและมีรายละเอียดค่อนข้างมาก

ภาพที่ 33a



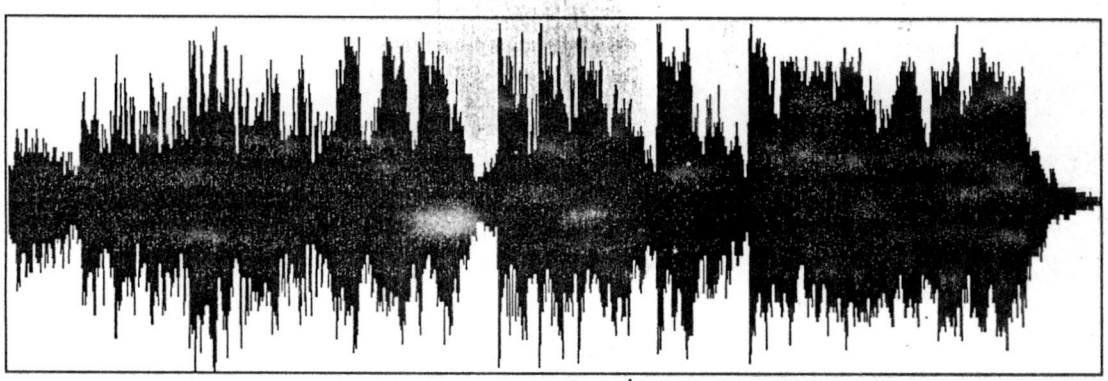
แสดงรูปแบบสัญญาณเสียงดนตรีต้นแบบ

ภาพที่ 33b



แสดงรูปแบบสัญญาณเสียงดนตรีที่ผ่านการเข้ารหัสลับ

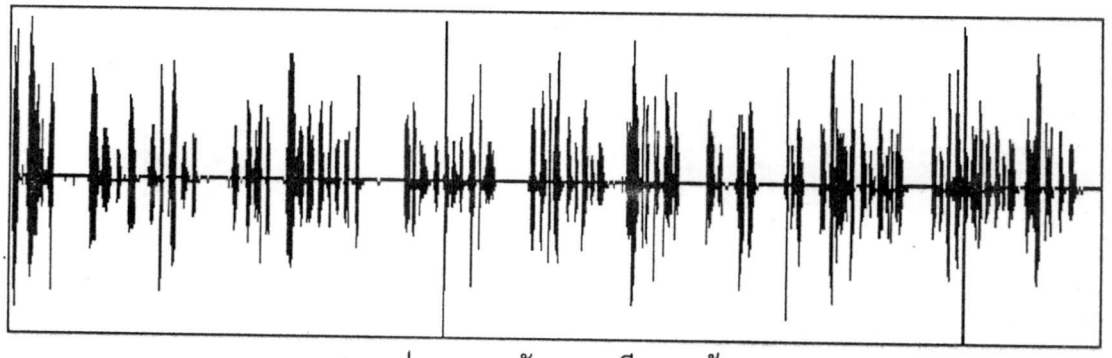
ภาพที่ 33c



แสดงรูปแบบสัญญาณเสียงดนตรีที่ผ่านการถอดรหัสลับ

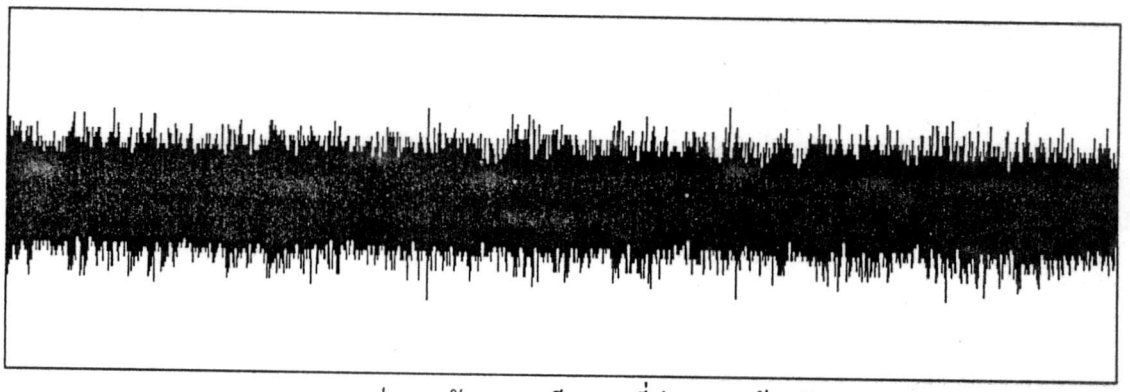
ในการทดลองรูปแบบที่สองใช้สัญญาณเสียงพูดเพียงอย่างเดียว โดยมีรูปแบบสัญญาณต้นแบบแสดงในภาพที่ 34a สำหรับนำมาทำการเข้ารหัสลับ โดยจะทำการแปลงสัญญาณเหล่านี้ให้อยู่ในรูปของสัญญาณดิจิตอล แล้วทำการเข้ารหัสลับนำผลที่ได้จากการเข้ารหัสมาทำการแปลงให้อยู่ในรูปสัญญาณแอนะล็อกดังแสดงในภาพที่ 34b และในภาพที่ 34c เป็นการแสดงรูปแบบของสัญญาณเสียงที่ผ่านการถอดรหัสลับแล้ว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

ภาพที่ 34a



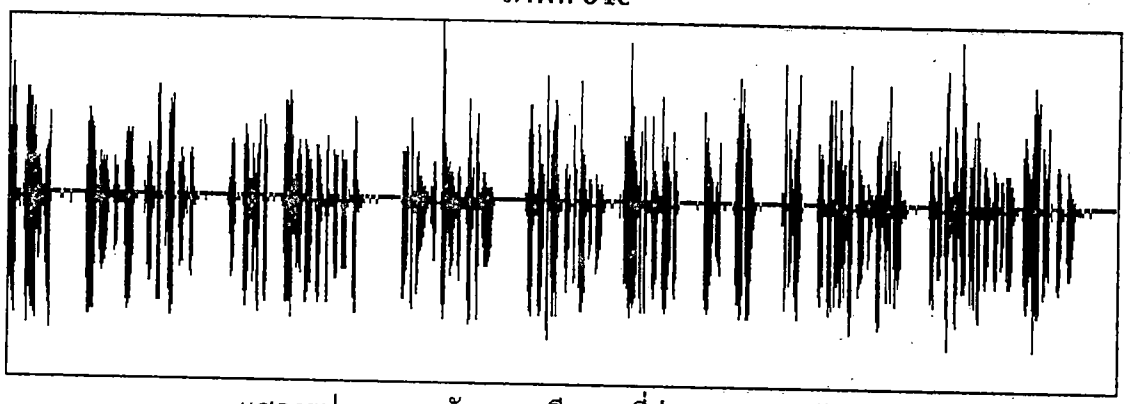
แสดงรูปแบบของสัญญาณเสียงพูดต้นแบบ

ภาพที่ 34b



แสดงรูปแบบสัญญาณเสียงพูดที่ผ่านการเข้ารหัส

ภาพที่ 34c



แสดงรูปแบบของสัญญาณเสียงพูดที่ผ่านการถอดรหัสลับ

จากผลการทดลองของรูปแบบที่สอง เมื่อทำการแปลงเสียงพูดต้นแบบในภาพที่ 34a ให้เป็นดิจิตอลแล้วนำมาทำการเข้ารหัสลับ และแปลงให้อยู่ในรูปของสัญญาณแอนะล็อกจะเห็นว่ามีรูปแบบสัญญาณที่แตกต่างไปจากสัญญาณต้นแบบแสดงในภาพที่ 34b และทำการถอดรหัสกลับจะได้รูปบบของสัญญาณแสดงดังในภาพที่ 34c ซึ่งมีความใกล้เคียงกับสัญญาณต้นแบบมาก

ในการทดลองรูปแบบที่สาม กระทำโดยนำข้อมูลที่อยู่ในรูปแบบของสัญญาณดิจิตอลซึ่งเป็นข้อมูลต้นแบบแสดงในภาพที่ 35a มาทำการเข้ารหัสลับ โดยข้อมูลที่ผ่านการเข้ารหัสแล้วแสดงในภาพที่ 35b และเมื่อนำมาทำการถอดรหัสกลับจะได้ข้อมูลกลับคืนมาดังแสดงในภาพที่ 35c

ภาพที่ 35a

ABCDEFGHIJKLMNOPQRSTUVWXYZ

แสดงข้อมูลข่าวสารต้นแบบ

ภาพที่ 35b

´ÒÇÒlpS;[WO/[¦Èð-®Ï©ì¯²¯Ä«E,Z®.”-°,«Èð

แสดงข้อมูลที่ผ่านการเข้ารหัส

ภาพที่ 35c

ABCDEFGHIJKLMNOPQRSTUVWXYZ

แสดงข้อมูลที่ผ่านการถอดรหัสลับ