การเข้ารหัสโดย DSP(DIGITAL SIGNAL PROCESSING)

DATA CODING USING DSP(DIGITAL SIGNAL PROCESSING)

นายกฤษฎา ตาทอง
นายนิพนธ์ พัชรสถิตย์
นายอนวัช มงคลวิสุทธิ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2544

DATA CODING USING DSP(DIGITAL SIGNAL PROCESSING)

KRISDA TATONG

NIPON PATCHARASATID

ANAWAT MONGKHONWISUT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

DEPARTMENT OF INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2001

ภาควิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์ การเข้ารหัสโดย DSP(DIGITAL SIGNAL PROCESSING)

DATA CODING USING DSP(DIGITAL SIGNAL PROCESSING)

นักศึกษาผู้จัดทำ นายกฤษฎา ตาทอง **รหัสประจำตัว** 42015421

นายนิพนธ์ พัชรสถิตย์ **รหัสประจำตัว** 42015437

นายอนวัช มงคลวิสุทธิ์ **รหัสประจำตัว** 42015459

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

ปีการศึกษา 2544

อาจารย์ผู้ควบคุมปริญญานิพนธ์	**ลายมือชื่อ**
รศ.ดร.ฟูศักดิ์ ชีวสุวิทย์	

วัน/เดือน/ปี ที่สอบ วันอังคารที่ 26 มีนาคม พ.ศ. 2545

สถานที่สอบ ณ ห้องสอบปริญญานิพนธ์ ภาควิชาวิศวกรรมการวัดคุม

ภาควิชารับรองแล้ว

(ผศ.ประสิทธิ์ จุลเสรีวงศ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

หัวข้อปริญญานิพนธ์ การเข้ารหัสโดยใช้ DSP(DIGITAL SIGNAL PROCESSING)
DATA CODING USING DSP(DIGITAL SIGNAL PROCESSING)
นักศึกษาผู้จัดทำ นายกฤษฎา ตาทอง
นายนิพนธ์ พัชรสถิตย์
นายอนวัช มงคลวิสุทธิ์
อาจารย์ที่ปรึกษา รศ.ดร.ฟูศักดิ์ ชีวสุวิทย์
ปีการศึกษา 2544

บทคัดย่อ

การสื่อสารข้อมูลเป็นที่ยอมรับกันทั่วไปว่ามีความสำคัญต่อชีวิตประจำวันมากยิ่งขึ้น ซึ่งแนวโน้มของการแลกเปลี่ยนข้อมูลส่วนตัวมักจะส่งผ่านเครือข่ายของการสื่อสาร จุดนี้เองเป็นสาเหตุที่จะทำให้เกิดการรั่วไหลของข้อมูล เนื่องจากการทำการจารกรรมข้อมูลซึ่งเป็นความเสี่ยงต่อความปลอดภัยของข้อมูลและการละเมิดสิทธิส่วนบุคคล ดังนั้นในปริญญานิพนธ์นี้จึงได้นำเสนอเทคนิคการเข้ารหัสลับข้อมูลก่อนส่งออกไปในตัวกลางการสื่อสาร เพื่อป้องกันมิให้ข่าวสารรั่วไหลหรือถูกจารกรรม โดยเทคนิคการเข้ารหัสลับนี้จะประกอบด้วยการประยุกต์ใช้ซีนโดรมของ บล็อคโค้ดเชิงเส้น และการกำเนิดแบบสุ่ม ผลของการเข้ารหัสนี้ได้แสดงให้เห็นชัดเจนว่าสามารถปกปิดข้อมูลได้อย่างสมบูรณ์

Thesis Title DATA CODING USING DSP(DIGITAL SIGNAL PROCESSING)

Authors Mr.Krisda Tatong

Mr.Nipon Patcharasatid

Mr.Anawat Mongkhonwisut

Thesis Advisor Assoc.Prof.Dr.Fusak Cheevasuvit

Year 2001

ABSTRACT

Data communication is accepted as an important part of our everyday lift. 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

ปริญญานิพนธ์ฉบับนี้ สำเร็จลุล่วงได้เป็นอย่างดีด้วยคำแนะนำและให้คำปรึกษาอย่างดีของ รศ.ดร.ฟูศักดิ์ ชีวสุวิทย์ ซึ่งเป็นอาจารย์ที่ปรึกษา คณะผู้จัดทำรู้สึกซาบซึ้งในความกรุณาและกราบขอบพระคุณเป็นอย่างสูง

กราบขอบพระคุณคุณพ่อคุณแม่เป็นอย่างยิ่ง ที่ให้ทุกสิ่งทุกอย่างเพื่อการศึกษาและการทำปริญญานิพนธ์ด้วยดีที่สุด ขอขอบพระคุณอาจารย์ สุรพันธุ์ แย้มมั่น ภาควิชาฟิสิกส์อุตสาหกรรม สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ ในคำแนะนำที่เป็นประโยชน์ รวมถึงเอื้อเฟื้ออุปกรณ์สำหรับการทดลอง และขอขอบคุณเพื่อน ๆ ที่ให้ความช่วยเหลือและให้กำลังใจในการทำปริญญานิพนธ์จนสำเร็จลุล่วงด้วยดี

คณะผู้จัดทำ

สารบัญ

หน้า

บทคัดย่อภาษาไทย I

บทคัดย่อภาษาอังกฤษ II

กิตติกรรมประกาศ III

สารบัญ IV

สารบัญตาราง VII

สารบัญภาพ VIII

บทที่ 1 บทนำ 1

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย 1

1.2 วัตถุประสงค์ของปริญญานิพนธ์ 1

1.3 ขอบเขตของปริญญานิพนธ์ 2

1.4 ขั้นตอนการศึกษา 2

บทที่ 2 ความรู้เบื้องต้นเกี่ยวกับอุปกรณ์ประเภท DSP 3

2.1 กล่าวนำ 3

2.2 สถาปัตยกรรมของ TMS320C31 3

2.2.1 หน่วยประมวลผลกลาง (CPU) 3

2.2.2 รีจีสเตอร์และหน้าที่ 4

2.1.1 หน่วยความจำ 7

บทที่ 3 หลักการและทฤษฎีที่ใช้ในการออกแบบเข้ารหัสลับ 10

3.1 กล่าวนำ 10

3.2 การแก้รหัสที่ผิด 12

3.2.1 การคำนวณทางคณิตศาสตร์ที่ใช้ในการแก้รหัสที่ผิด 12

3.2.2 การตรวจสอบความถูกต้องของรหัสคำ 12

3.2.3 การตรวจสอบพาริตี้ (Parity check) 13

3.2.4 ความสามารถในการตรวจแก้บิตที่ผิดในรหัสเชิงเส้น 15

สารบัญ (ต่อ)

หน้า

3.3 บล็อคโค้ดเชิงเส้น (Linear block code) 17

3.3.1 แมทริกซ์ตัวกำเนิด (Generator matrix) 18

3.3.2 แมทริกซ์ในการตรวจสอบพาริตี้ (Parity check matrix) 20

3.3.3 ซีนโดรม 21

บทที่ 4 การออกแบบตัวเข้ารหัสและตัวถอดรหัสลับ.....26

4.1 กล่าวนำ26

4.2 การประยุกต์ใช้ DSP ตามหลักการและทฤษฎี26

4.2.1 การออกแบบตัวเข้ารหัสลับ(Encoder) 26

4.2.2 ส่วนของรูปแบบผิดพลาด (Error Pattern) 28

4.2.3 ส่วนของการสลับตำแหน่งบิต (Bit Allocation) 28

4.2.4 การออกแบบตัวถอดรหัสลับ (Decoder) 29

4.2.5 ส่วนของการสลับตำแหน่งบิตกลับ(Bit Reallocation) 30

4.2.6 ส่วนของการคำนวณหาซีนโดรม.....30

บทที่ 5 การทดลองและผลการทดลอง 33

5.1 กล่าวนำ.....33

5.2 ผลการทดลอง.....34

5.3 สรุปผลการทดลอง 38

บทที่ 6 สรุปผลการวิจัยและข้อเสนอแนะ.....39

6.1 สรุปผลการทดลอง 39

6.2 ปัญหาและอุปสรรคที่พบ.....39

6.3 วิธีแก้ไข 40

6.4 ข้อเสนอแนะและแนวทางการพัฒนา.....40

บรรณานุกรม 41

สารบัญ (ต่อ)

หน้า

ภาคผนวก.....42

สารบัญตาราง

ตารางที่ หน้า

2.1 ชื่อและหน้าที่ต่าง ๆ ของรีจีสเตอร์ใน CPU 6
2.2 ชื่อและหน้าที่ต่าง ๆ ของรีจีสเตอร์ใน CPU (ต่อ) 7
3.1 รหัสพาริตี้แบบคี่ 14
3.2 รหัสพาริตี้แบบคู่ 14
3.3 แสดงซีนโดรมที่ได้จากรูปแบบรหัสที่ผิดไปหนึ่งบิต 24
4.1 รูปแบบผิดพลาด 28
4.2 การสลับตำแหน่งบิต 29
4.3 การสลับตำแหน่งบิตกลับ 30

สารบัญภาพ

ภาพที่ หน้า

2.1 แสดงสถาปัตยกรรมของไอซีเบอร์TMS320C3X....3
2.2 แสดงหน่วยความจำของไอซีเบอร์TMS320C31....7
2.3 แสดงการจัดหน่วยความจำของไอซีเบอร์TMS320C31....9
3.1 แสดงการเข้ารหัสเสียงแบบอนาล็อกชนิดไม่มีการประมวลสัญญาณดิจิตอล....10
3.2 แสดงการเข้ารหัสอนาล็อกชนิดมีการประมวลสัญญาณดิจิตอล....11
3.3 แสดงการเข้ารหัสเสียงแบบดิจิตอล....11
3.4 การคำนวณทางคณิตศาสตร์ของ Galois field (ก) การบวก (ข) การคูณ....12
3.5 การเพิ่มพาริตี้บิตให้กับรหัสข่าวสาร....13
3.6 แสดงรูปรหัสคำของบล็อคโค้ดเชิงเส้น....18
4.1 แสดงแผนภาพการเข้ารหัสลับ....26
4.2 แสดงแมทริกซ์ตัวกำเนิด....27
4.3 แสดงการสร้างรหัสคำ....27
4.4 แสดงแผนภาพการถอดรหัสลับ....29
4.5 แสดงแมทริกซ์พาริตี้....31
4.6 แสดงการหาค่าซีนโดรม....31
5.1 แสดงการทำงานโดยรวม....33
5.2 แสดงการทำงานในหนึ่งรอบ....34
5.3 แสดงรูปแบบสัญญาณเสียงพูดต้นแบบ....35
5.4 แสดงรูปแบบสัญญาณเสียงพูดที่ผ่านการเข้ารหัสลับ....35
5.5 แสดงรูปแบบสัญญาณเสียงพูดที่ผ่านการถอดรหัสลับ....35
5.6 แสดงรูปแบบสัญญาณเสียงดนตรีต้นแบบ....36
5.7 แสดงรูปแบบสัญญาณเสียงดนตรีที่ผ่านการเข้ารหัสลับ....36
5.8 แสดงรูปแบบสัญญาณเสียงดนตรีที่ผ่านการถอดรหัสลับ....36
5.9 แสดงรูปแบบสัญญาณไซน์ต้นแบบ....37
5.10 แสดงรูปแบบสัญญาณไซน์ที่ผ่านการเข้ารหัสลับ....37
5.11 แสดงรูปแบบสัญญาณไซน์ที่ผ่านการถอดรหัสลับ....37

บทที่ 1
บทนำ

1.1 ความเป็นมาและเหตุจูงใจของการวิจัย

เนื่องจากปัจจุบันการติดต่อสื่อสารไม่ว่าจะเป็นทางโทรทัศน์ โทรศัพท์ คอมพิวเตอร์ วิทยุ ได้เข้ามามีบทบาทสำคัญอย่างสูง ในสังคมปัจจุบัน จะมีกลุ่มของผู้ให้บริการการสื่อสารข้อมูลในรูปแบบต่าง ๆ สำหรับการดำเนินธุรกิจ การทหาร หรือข้อมูลส่วนตัว ที่ผ่านตัวกลางการสื่อสารรูปแบบต่าง ๆ โดยข่าวสารที่ใช้ในการติดต่อสื่อสารจะมีความสำคัญของข่าวสารต่าง ๆ ตามลำดับ ซึ่งความสำคัญของข่าวสารนั้นจะขึ้นอยู่กับผู้ส่งสารจะเป็นผู้ให้ความสำคัญของข่าวสารนั้น เช่น ข่าวสารทางการทหาร หรือข่าวสารทางด้านธุรกิจจะมีความสำคัญมาก และจะมีความล่อแหลมในการถูกดักฟัง หรือการจารกรรมจากผู้ที่ต้องการทำลาย ทำให้ผู้ส่งสารต้องพยายามหาวิธีการในการป้องกันข้อมูลข่าวสารของตนเองให้รอดพ้นจากการถูกจารกรรม วิธีการหนึ่งที่ถูกนำมาใช้ คือ การเข้ารหัสข้อมูล (Scramble) ก่อนทำการส่งผ่านในตัวกลางของการสื่อสารและทำการถอดรหัส (Descramble) ข้อมูลที่ได้รับกลับคืนมาจากผู้รับข่าวสารนั้น ๆ ซึ่งทำให้ข้อมูลข่าวสารรอดพ้นจากการถูกจารกรรมได้ ทำให้ผู้ที่ต้องการจารกรรมข้อมูลไม่สามารถทำการถอดรหัสข้อมูลเหล่านั้นออกมาได้ ในปริญญานิพนธ์นี้ได้นำเสนอวิธีการเข้ารหัสโดยใช้หลักการบล็อคโค้ดเชิงเส้นมาใช้สำหรับการเข้ารหัสลับและการถอดรหัสลับโดยอาศัยซีนโดรมของบล็อคโค้ดเชิงเส้นเพื่อใช้สำหรับการป้องกันข้อมูลข่าวสารให้ได้รับความปลอดภัย

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อเป็นการศึกษาทฤษฎีการแก้รหัสที่ผิดแบบบล็อคโค้ดเชิงเส้น และ DSK
2. เพื่อออกแบบระบบปกปิดข้อมูลเสียงโดยใช้หลักการแก้รหัสที่ผิด ที่ใช้งานบน DSK
3. เพื่อทดลองและสร้างระบบปกปิดข้อมูลเสียงโดยใช้หลักการแก้รหัสที่ผิด ที่ใช้งานบน DSK
4. เพื่อเป็นแนวทางสำหรับการพัฒนาในการออกแบบวิธีการสร้าง การเข้ารหัสลับและวิธีการถอดรหัสลับ โดยเป็นแนวทางสำหรับผู้สนใจทางด้านการป้องกันข้อมูลเสียงให้ปลอดภัย และเป็นการช่วยประหยัดเวลาในการเริ่มศึกษาทางด้านนี้
5. สามารถนำผลงานการทดลองนี้ไปใช้งานสำหรับป้องกันข้อมูลเสียงให้มีความปลอดภัยเบื้องต้นได้

1.3 ขอบเขตของปริญญานิพนธ์

ขอบเขตของวิทยานิพนธ์จะเป็นการประยุกต์ใช้ DSP มาใช้ประมวลผลสัญญาณ โดยอาศัยหลักการของบล็อคโค้ดเชิงเส้นและการหาซีนโดรม ซึ่งเป็นหลักวิธีในการนำมาใช้ในการออกแบบวิธีการเข้ารหัสลับและวิธีการถอดรหัสลับ สามารถแบ่งเนื้อหากล่าวแยกเป็นบท ๆ ได้ดังนี้

บทที่ 1 บทนำ กล่าวถึงความเป็นมาและความสำคัญ วัตถุประสงค์และรายละเอียดเนื้อหาในบทต่าง ๆ

บทที่ 2 ความรู้เบื้องต้นเกี่ยวอุปกรณ์ประเภท DSP

บทที่ 3 ระบบการเข้ารหัสลับและทฤษฎี ในรายละเอียดจะเป็นการกล่าวถึงพื้นฐานของการเข้ารหัสแบบต่าง ๆ และทฤษฎีของบล็อคโค้ดเชิงเส้น

บทที่ 4 หลักการของตัวเข้ารหัสลับและถอดรหัสลับ ในรายละเอียดจะกล่าวถึงการนำวิธีการบล็อคโค้ดเชิงเส้นมาใช้ในการออกแบบวิธีการเข้ารหัสลับและวิธีการถอดรหัสลับและขั้นตอนต่าง ๆ ของการเข้ารหัส และการนำ DSP มาใช้งาน ในรายละเอียดเป็นการกล่าวถึงการนำ DSP มาใช้สำหรับประมวลผลสัญญาณเพื่อทำการเข้ารหัสลับและถอดรหัสลับ

บทที่ 5 แสดงผลการทดลอง โดยทำการเข้ารหัสและถอดรหัสกับสัญญาณเสียง

บทที่ 6 บทสรุปและแนวทางการพัฒนา กล่าวสรุปเนื้อหาที่ทำมารวมถึงปัญหาที่ประสบ

1.4 ขั้นตอนการศึกษา

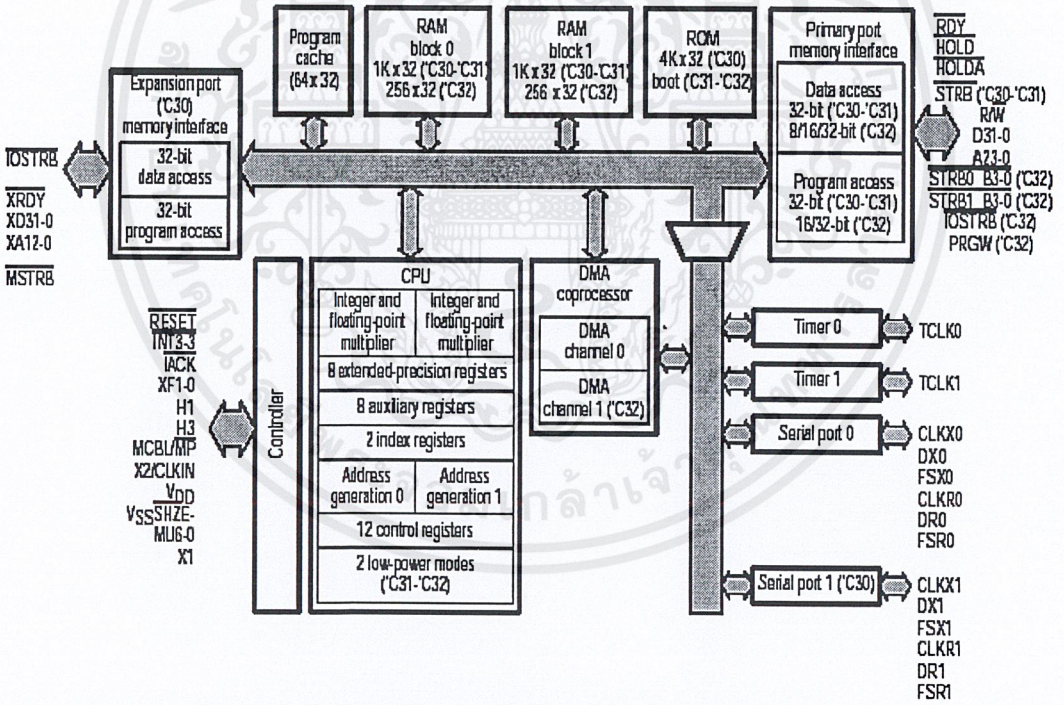
การทำโครงงานวิจัยปริญญาในปริญญานิพนธ์ฉบับนี้มีขั้นตอนการศึกษาเริ่มจาก การศึกษาการทำงานของDSK เช่น การทำงานของระบบ, หน่วยประมวลผลของระบบ, หน่วยความจำ, คำสั่ง, โปรแกรม Debugger ตลอดจนการศึกษาโปรแกรมการทำงานเบื้องต้นสำหรับ DSK จากนั้นได้ทำการศึกษาถึงหลักการเข้ารหัสลับโดยใช้ซีนโดรมของบล็อกโค้ดเชิงเส้น จากนั้นได้ทำการประยุกต์หลักการเข้ารหัสลับโดยใช้ซีนโดรมของบล็อกโค้ดเชิงเส้นเพื่อนำมาเขียนโปรแกรมสำหรับ DSK จากนั้นทำการประมวลผลในการเข้ารหัสและถอดรหัสตามรูปแบบที่ได้กำหนดไว้

บทที่ 2
ความรู้เบื้องต้นเกี่ยวกับอุปกรณ์ประเภท DSP

2.1 กล่าวนำ

ในการทดลองนี้จะใช้ DSK ที่มี ไอซีเบอร์ TMS320C31 ของ TEXAS INSTRUMENTS ซึ่งถูกสร้างและออกแบบมาเพื่อตอบสนองความต้องการทางด้านการประมวลผลบนพื้นฐานของคณิตศาสตร์ชั้นสูงและการแก้ปัญหาทาง Hard ware และ Solf ware

ประสิทธิ์ภาพอันสูงของ TMS320C31 เกิดจากความแน่นอนและความกว้างทางไดนามิกของส่วนคำนวณทางทศนิยม มีหน่วยความจำขนาดใหญ่ การทำงานแบบขนานและมี DMA Controller เพื่อการทำงานที่มีประสิทธิ์ภาพมากขึ้น



ภาพที่ 2.1 แสดงสถาปัตยกรรมของไอซีเบอร์ TMS320C3X

2.2 สถาปัตยกรรม TMS320C31 ประกอบด้วยส่วนต่างๆสำคัญดังนี้

2.2.1หน่วยประมวลผลกลาง (CPU)

หน่วยประมวลผลกลางจะประกอบไปด้วย

- Multiplier

มีหน้าที่ในการคูณ และสามารถคูณจำนวนเต็ม 24 bit และจำนวนทศนิยม 32 บิต โดยใช้เวลาเพียง 1 รอบ ซึ่งการคำนวณจำนวนทศนิยมจะใช้เวลา 50 ns ต่อรอบ และสามารถประมวลผลแบบขนานได้เพื่อที่จะเพิ่มความสามารถประมวลผลข้อมูล โดยใช้คำสั่งแบบขนานให้ทำการคูณและการทำงานของ ALU ภายในหนึ่งรอบ

เมื่อ Multiplier ทำการคำนวณทศนิยม อินพุตที่ใช้มีขนาด 32 บิต และผลลัพธ์ที่ได้จะมีขนาด 40 บิต แต่ถ้า Multiplier ทำการคูณจำนวนเต็มอินพุตมีขนาดเพียง 24 บิต และจะได้ผลลัพธ์ขนาด 32 บิต

- หน่วยช่วยคำนวณทางคณิตศาสตร์ (Arithmetic Logic Unit (ALU))

ใน 1 รอบการทำงานของ ALU จะประมวลผลจำนวนเต็มได้ 32 บิต และทศนิยมได้ 40 บิต โดยผลลัพธ์จาก ALU ยังคงมีขนาด 32 บิตถ้าเป็นจำนวนเต็ม และ 40 บิตถ้าเป็นทศนิยม barrel shifter จะมีหน้าที่ในการเลื่อนข้อมูล 32 บิต ซ้ายและขวาใน 1 รอบการทำงาน

บัสภายใน(CPU1/CPU2 และ REG1/REG2) จะเป็นตัวพาข้อมูลที่จะนำมาประมวลผล 2 ตัว จากหน่วยความจำและอีก 2 ตัวจากรีจีสเตอร์ ดังนั้นจึงสามารถทำการคูณ บวก หรือลบแบบขนานได้ ใน 1 รอบการทำงาน

- หน่วยช่วยคำนวณทางคณิตศาสตร์ (Auxiliary Register Arithmetic Unit (ARAUs))

ARAUs เป็นรีจีสเตอร์ที่ช่วยการทำงานแบบขนานกับตัวคูณและ ALU โดย ARAUs จะมีอยู่ 2 ตัวคือ ARAU0 และ ARAU1 ARAU จะสามารถเก็บแอดเดรส ได้ 2 แอดเดรสใน 1 รอบการทำงานเพื่อใช้ในการอ้างตำแหน่งแบบต่างๆ

- CPU Register File

ใน TMS320C31 จะมีรีจีสเตอร์ 28 ตัว ซึ่งตรงกับ CPU โดยรีจิสเตอร์ทั้งหลายเหล่านี้จะทำงานกับ multiplier และ ALU และยังสามารถใช้เพื่อวัตถุประสงค์อื่นอีกด้วย อย่างไรก็ตามรีจีสเตอร์เหล่านี้ก็ยังมีหน้าที่พิเศษบางอย่างด้วย เช่น ใช้สำหรับการทำงานเกี่ยวกับจำนวนเลขทศนิยม และรีจีสเตอร์ช่วย (Auxiliary register) ทั้ง 8 ตัวยังใช้ในการอ้างแอดเดรสแบบ Indirect Addressing Register ด้วย รีจีสเตอร์ที่เหลือจะทำหน้าที่เกี่ยวกับระบบต่างๆ เช่น การอ้างแอดเดรสการจัดการเกี่ยวกับแสตก(stack) การบอกสถานะของโปรเซสเซอร์ การอินเตอร์รัพต์ และการเคลื่อนย้ายข้อมูลเป็นบล็อก

2.2.2 รีจีสเตอร์และหน้าที่

- Exten - precision register (R7-R0) :

สามารถใช้เก็บค่าจำนวนเต็ม 32 บิต และทศนิยม 40 บิตโดยจะเก็บค่าลงในบิตที่ 0-39 แต่ถ้าจำนวนเต็มแบบมีเครื่องหมายก็จะทำการเก็บที่บิต 0-31 ส่วนบิตที่ 32-39 จะไม่ใช้

- Auxiliary register (AR7-AR0) :

จะถูกใช้โดย CPU และจะถูกควบคุมโดย ARAU ทั้ง 2 ตัว หน้าที่แรกของ AR ยังใช้ให้ส่วนที่เก็บตัวนับในการวนลูปหรือจะใช้เพื่องานอื่นๆ ที่เกียวกับตัวมัลติเพล็กเซอร์และ ALU

- Data page pointer (DP) :

ส่วนรีจีสเตอร์ขนาด 32 บิต โดยที่ 5 บิตแรก จะถูกใช้ในการอ้างแอดเดรสแบบ direct addressing mode ซึ่งใช้เป็นตัวชี้ (pointer) ไปยังหน้า (page) ต่างๆของข้อมูล

- Index register (IR0,IR1) :

จะถูกใช้โดย ARAU ในการเก็บค่าของดัชนีในการชี้แอดเดรส

- Block size register (BK) :

มีขนาด 32 บิตซึ่ง ARAU จะใช้ในการอ้างแอดเดรสแบบเก็บ Circular Addressing เพื่อเป็นตัวเก็บค่าขนาดของบล็อกข้อมูล

- System stack pointer (SP) :

เป็นรีจีสเตอร์ขนาด 32 บิต ซึ่งเป็นค่าของแอดเดรสของยอดของแสตก โดยปกติ SP จะชี้ไปยังค่าสุดท้ายซึ่งเก็บอยู่บนแสตก การ push จะกระทำ ก่อนที่จะเพิ่มค่า SP และการ POP หลังการลดค่าของ SP คำสั่งที่จะทำงานกับ SP ได้คือ interrupts, traps, return, PUSH และ POP

- Status register (ST) :

เป็นรีจีสเตอร์ที่เก็บข้อมูลต่างๆที่แสดงสถานะของ CPU เช่นค่าของ flag ซึ่งจะแสดงสถานะของผลลัพธ์ที่ได้จากการคำนวณทางคณิตศาสตร์ ว่าเป็น ศูนย์ ลบ ฯลฯ รวมถึงการคำนวณทางลอจิกด้วย

- CPU/DMA interrupt enable register (IE) :

เป็นรีจีสเตอร์ขนาด 32 บิต ใช้สำหรับการกำหนดการเอ็นนาเบิ้ลอินเตอร์รัพต์ของ CPU โดยการเอ็นนาเบิ้ลของ CPU จะใช้บิตที่ 0-10 และการเอ็นนาเบิ้ลของ DMA จะใช้บิตที่ 16-26 " 1 " จะเป็นการเอ็นนาเบิ้ล และ " 0 " จะเป็นการดีสเอเบิ้ล

- I/O Flag register (IOF) :

เป็นรีจีสเตอร์ในการควบคุมขา XF0 และ XF1

- Repeat counter (RC) :

เป็นรีจีสเตอร์ขนาด 32 บิตใช้เก็บค่าของจำนวนเท่าหรือจำนวนครั้งของบล็อกข้อมูลซึ่งถูกกระทำ เมื่อโปรเซสเซอร์ทำงานในโหมดของการกระทำในข้อมูลซ้ำ (repeat mode) repeat star address register (RS) เป็นรีจีสเตอร์ขนาด 32 บิต จะถูกใช้ในการเก็บแอดเดรสเริ่มต้นของบล็อกข้อมูลที่จะถูกกระทำซ้ำ และ repeat end address register (RE) : ซึ่งเป็นรีจีสเตอร์ขนาด 32 บิต จะถูกเก็บแอดเดรสสุดท้ายของบล็อกข้อมูลซึ่งจะถูกกระทำซ้ำ

- Program counter (PC) :

เป็นรีจีสเตอร์ขนาด 32 บิต : ซึ่งเก็บแอดเดรสของคำสั่งที่จะถูกนำมาอ่านคำสั่งถัดไป

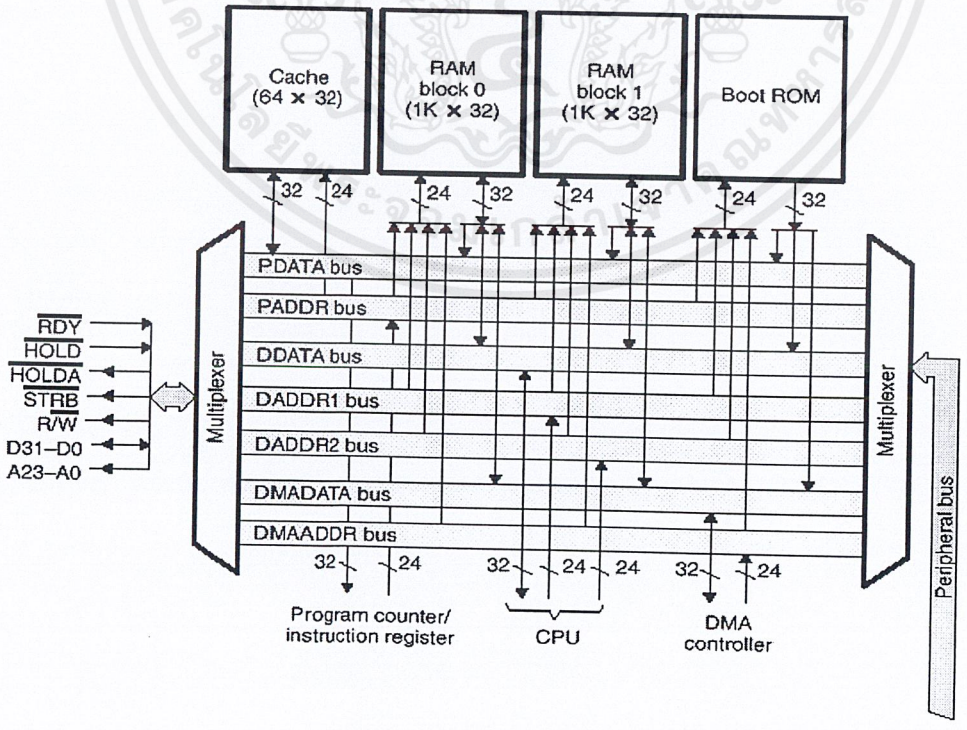
ตารางที่ 2.1 ชื่อและหน้าที่ต่างๆของรีจีสเตอร์ใน CPU

Register Name	Assigned Function
AR0	Auxiliary register 0
AR1	Auxiliary register 1
AR2	Auxiliary register 2
AR3	Auxiliary register 3
AR4	Auxiliary register 4
AR5	Auxiliary register 5
AR6	Auxiliary register 6
AR7	Auxiliary register 7
DP	Data-page pointer
IR0	Index register 0
IR1	Index register 1
BK	Block-size register
SP	System-stack pointer
ST	Status register
IE	CPU/DMA interrupt-enable
IF	CPU interrupt Flag
IOF	I/O Flag

ตารางที่ 2.1(ต่อ) ชื่อและหน้าที่ต่างๆของรีจีสเตอร์ใน CPU

Register Name	Assigned Function
RS	Repeat start-address
RE	Repeat end-address
RC	Repeat counter
PC	Program counter
R0	Extended-precision register 0
R1	Extended-precision register 1
R2	Extended-precision register 2
R3	Extended-precision register 3
R4	Extended-precision register 4
R5	Extended-precision register 5
R6	Extended-precision register 6
R7	Extended-precision register 7

2.2.3 หน่วยความจำ (Memory Organization)



ภาพที่ 2.2 แสดงหน่วยความจำของไอซีเบอร์ TMS320C31

หน่วยความจำของ TMS320C31 จะมีขนาด 16M และมีขนาดของ 1 word เท่ากับ 32 บิต โดยในหน่วยความจำนี้จะใช้กับโปรแกรมข้อมูล และการอินพุท เอาท์พุท ดังนั้นค่าของสัมประสิทธิ์โปรแกรมหรือข้อมูลจะถูกเก็บได้ทั้ง RAM และ ROM

- Ram, Rom

ภาพที่ 2.2 จะแสดงการจัดหน่วยความจำภายใน TMS320C31 โดยแรมจะแบ่งออกเป็น บล็อก 0 และ 1 ซึ่งจะมีขนาดบล็อกละ 1K x 32 และ ROM จะมีขนาด 4K x 32 ทั้ง RAM และ ROM สามารถถูกเข้าใช้ได้โดย CPU 2 ครั้งใน 1 รอบของการทำงานได้ดังนั้นการที่มีบัสของโปรแกรมแยกกันและบัสของ DMA ที่สามารถทำงานแบบขนานได้

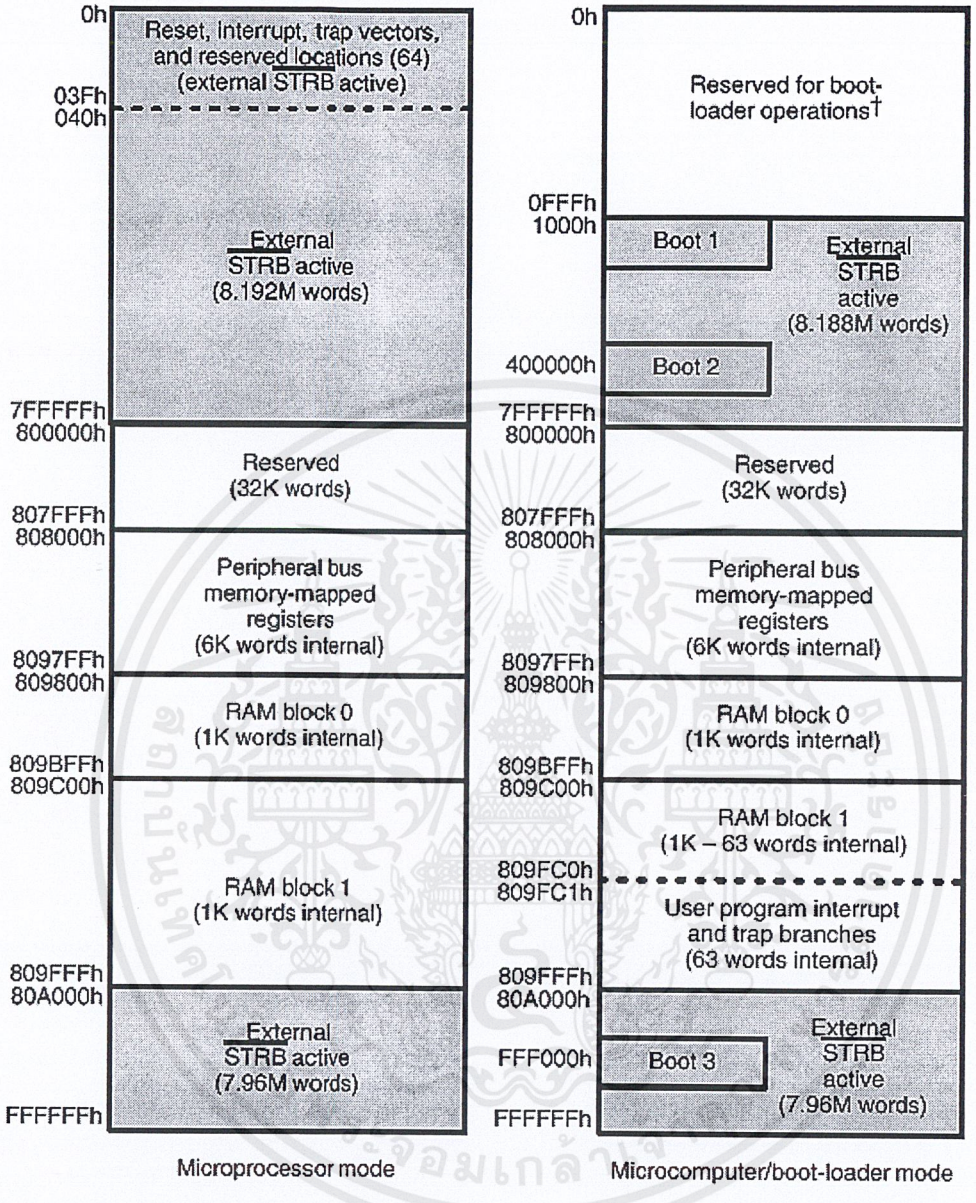
- Memory Maps

ตารางหน่วยความจำจะขึ้นอยู่กับว่าจะให้โปรเซสเซอร์ประมวลผลในโหมดของไมโครโปรเซสเซอร์ หรือไมโครคอมพิวเตอร์ แต่ในที่นี้จะให้โปรเซสเซอร์ประมวลผลในโหมดของไมโครโปรเซสเซอร์ แสดงในภาพที่ 2.3 ในแอดเดรสที่ 800000h จนถึง 801FFFh จะถูกใช้สำหรับ expansion bus ซึ่งจะถูกเข้าถึงได้โดยให้สัญญาณ / MSTRB แอคตีฟ แอดเดรสที่ 80200h จนถึง 803FFFh จะไม่ใช้แอดเดรสที่ 80400h ถึง 805FFFh จะถูกใช้สำหรับ expansion bus ซึ่งจะถูกทำให้เข้าถึงโดยให้สัญญาณ /IOSTRB แอคตีฟ แอดเดรส 80600h ถึง 807FFFh จะไม่ใช้ทุกๆรีจีสเตอร์จะอยู่ที่แอดเดรส 80800h ถึง 8097FFh RAM บล็อก 0 จะอยู่ที่ 809800h ถึง 809bFFh และ บล็อก 1 จะอยู่ที่ 809C00h ถึง 809FFh แอดเดรส 80A00h จนถึง 0FFFFFFh จะถูกใช้โดยอุปกรณ์ภายนอก (เมื่อสัญญาณ / STRB แอคตีฟ)

- Memory Address Mode

TMS32C31 จะแบ่งโหมดของการอ้างแอดเดรสออกเป็น 5 กลุ่ม ดังนี้

1. General addressing mode
2. Three – operand addressing mode
3. Parallel addressing mode
4. Long – immediate addressing mode
5. Contional branch addressing mode



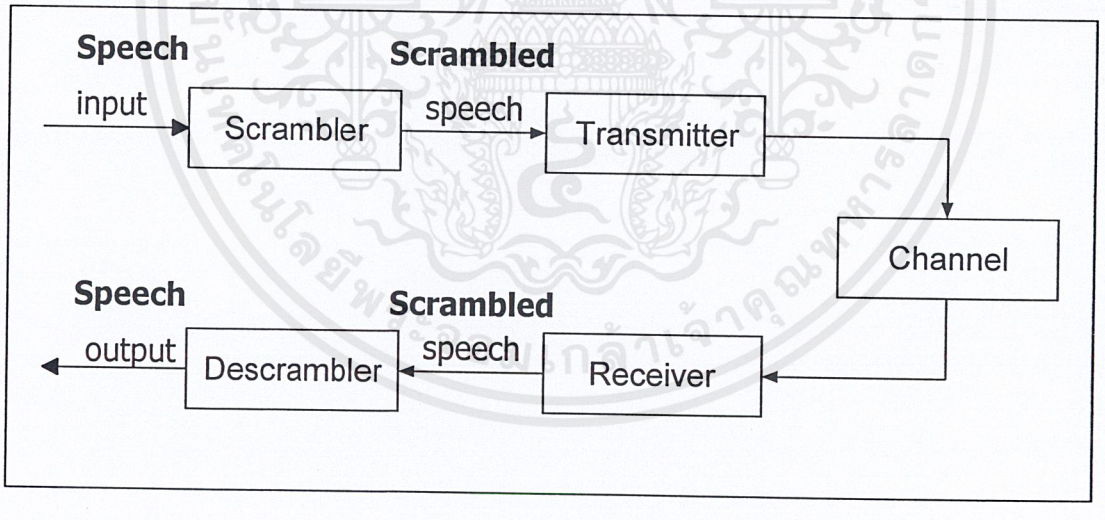
ภาพที่ 2.3 แสดงการจัดหน่วยความจำของไอซี TMS320C31

บทที่ 3
หลักการและทฤษฎีที่ใช้ในการออกแบบเข้ารหัสลับ

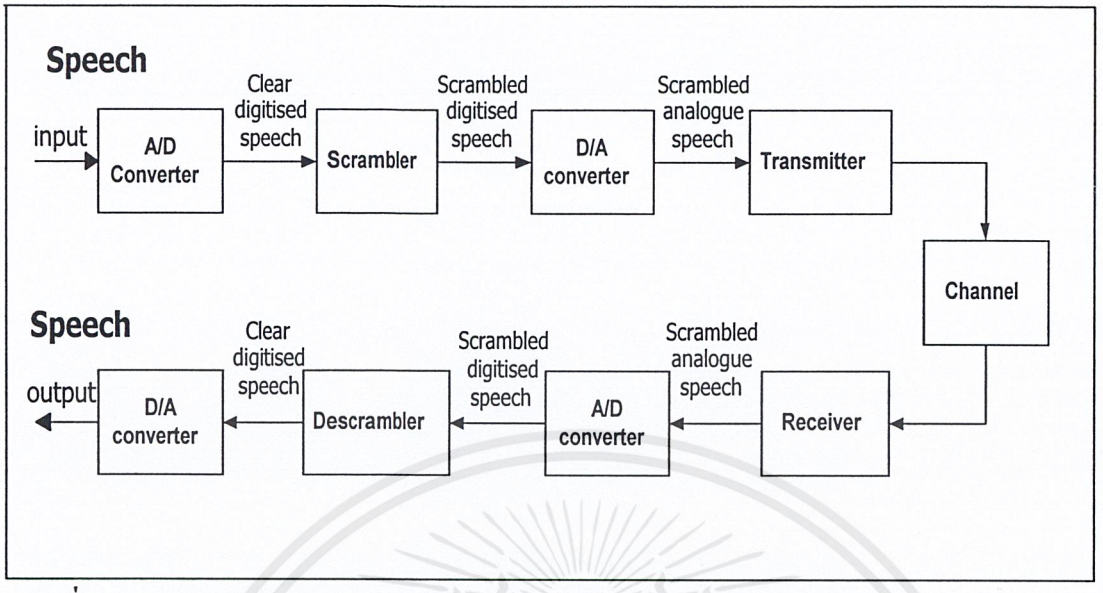
3.1 กล่าวนำ

โดยทั่วไปมักมีการกล่าวอ้างถึงวิธีการเข้ารหัสลับ (Scramble) ที่มีแบบพื้นฐาน 2 วิธี คือ แบบอนาล็อก (Analog) และแบบดิจิตอล (Digital) ส่วนใหญ่เครื่องเข้ารหัสสัญญาณที่มีความซับซ้อนมาก ๆ มักใช้กระบวนการของการประมวลผลสัญญาณดิจิตอล (Digital Signal Processing) กระบวนการนี้ทำงานโดยการแปลงสัญญาณอนาล็อกไปเป็นสัญญาณดิจิตอลก่อนแล้วจึงทำการเข้ารหัส ส่วนในระบบที่เป็นอนาล็อกโดยส่วนใหญ่จะมีความปลอดภัยค่อนข้างน้อย

ข้อแตกต่างที่เห็นได้ชัดเจนระหว่างการเข้ารหัสแบบอนาล็อกและแบบดิจิตอลคือ รูปแบบที่เกี่ยวกับตัวส่งผ่านซึ่งเป็นอุปกรณ์ที่ใช้ในการส่งสัญญาณที่เข้ารหัสแล้ว วัตถุประสงค์ของระบบที่มีการเข้ารหัสแบบอนาล็อกคือ ส่งผ่านข่าวสารที่มีการเปลี่ยนแปลงอย่างต่อเนื่อง ในทางตรงข้ามระบบเข้ารหัสลับแบบดิจิตอลจะส่งผ่านข่าวสารด้วยสัญญาณที่สามารถนำข่าวสารไปได้เฉพาะตามจำนวนที่จำกัดไว้ ดังแสดงได้ในแผนภาพดังต่อไปนี้

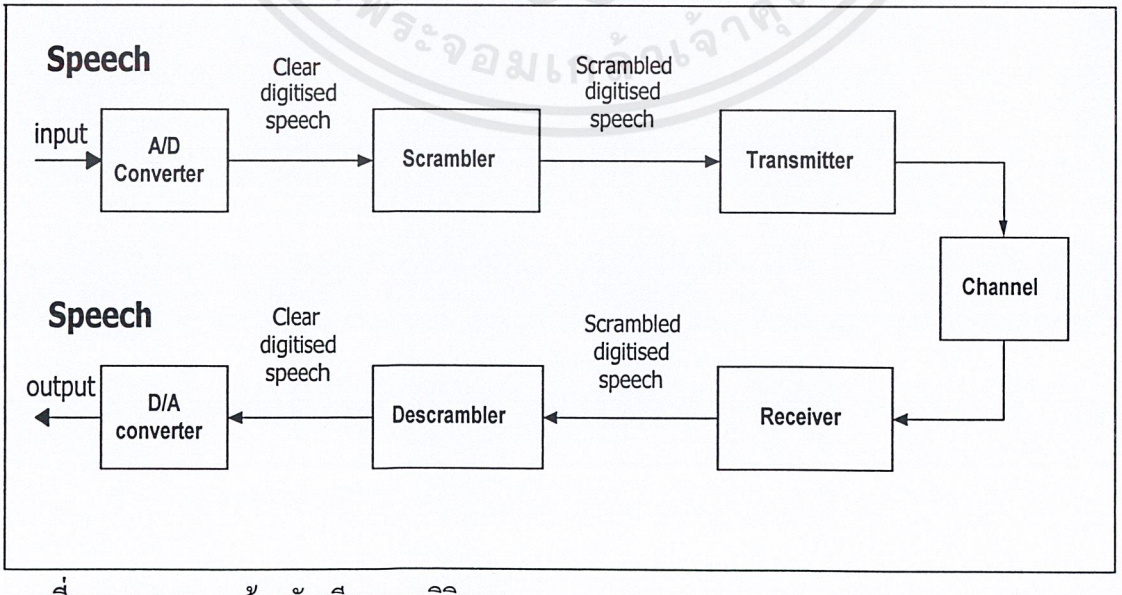


ภาพที่ 3.1 แสดงการเข้ารหัสเสียงแบบอนาล็อกชนิดไม่มีการประมวลสัญญาณดิจิตอล



ภาพที่ 3.2 แสดงการเข้ารหัสเสียงแบบอนาล็อกชนิดมีการประมวลผลสัญญาณดิจิตอล

ในภาพที่ 3.1 และ 3.2 แสดงให้เห็นถึงการเข้ารหัสสัญญาณแบบอนาล็อก ความแตกต่างระหว่างทั้ง 2 รูปแบบคือ รูปแบบของสัญญาณที่ผ่านการเข้ารหัสแล้ว เช่นในภาพที่ 3.1 และยังคงมีสัญญาณแบบอนาล็อกทั้งกระบวนการเข้ารหัส และในระบบตาม ภาพที่ 3.2 สัญญาณถูกเปลี่ยนไปอยู่ในรูปของสัญญาณดิจิตอลก่อนจะทำการเข้ารหัส แล้วถูกเปลี่ยนกลับให้อยู่ในรูปสัญญาณอนาล็อกก่อนทำการส่งผ่านและหลังจากการส่งผ่านจะถูกเปลี่ยนกลับให้เป็นสัญญาณดิจิตอลอีกครั้งหนึ่ง แล้วจึงจะทำการถอดรหัสลับและท้ายสุดจะถูกเปลี่ยนให้เป็นสัญญาณอนาล็อกอีกครั้งหนึ่ง



ภาพที่ 3.3 แสดงการเข้ารหัสเสียงแบบดิจิตอล

ในภาพที่ 3.3 แสดงให้เห็นถึงระบบดิจิตอลอีกประเภทหนึ่ง ความแตกต่างระหว่างระบบนี้และระบบในภาพที่ 3.2 คือไม่มีตัวแปลงสัญญาณจากดิจิตอลไปเป็นอนาล็อก (D/A Converter) ในทันทีก่อนเข้าสู่เครื่องส่งผ่าน และไม่ตัวแปลงสัญญาณจากอนาล็อกไปเป็นดิจิตอล (A/D Converter) ในทันทีหลังผ่านเครื่องรับ

3.2 การแก้รหัสที่ผิด

3.2.1 การคำนวณทางคณิตศาสตร์ที่ใช้ในการแก้รหัสที่ผิด

การออกแบบวงจรเข้ารหัสลับในปริญญานิพนธ์ฉบับนี้อาศัยการแก้รหัสที่ผิดแบบบล็อคโค้ดเชิงเส้น อาศัยการประมวลผลสัญญาณเลขฐานสอง ซึ่งมีคณิตศาสตร์เฉพาะงานเรียกว่า Galois field [2] , [3] สามารถออกแบบเป็นวงจรทางฮาร์ดแวร์ได้ง่ายเนื่องจากมีลักษณะเป็นลอจิก 0 และ 1

0 + 0 = 0	0 * 0 = 0
0 + 1 = 1	0 * 1 = 0
1 + 0 = 1	1 * 0 = 0
1 + 1 = 0	1 * 1 = 1
(ก)	(ข)

ภาพที่ 3.4 การคำนวณทางคณิตศาสตร์ของ Galois field (ก) การบวก (ข) การคูณ

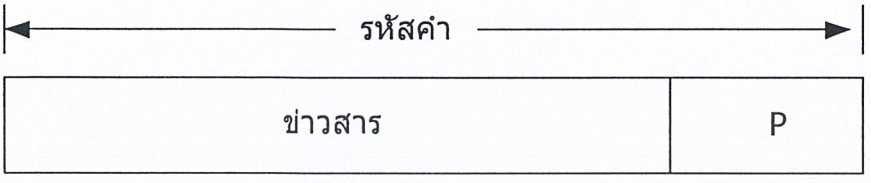
ใน Galois field จะมีการกระทำทางคณิตศาสตร์สองลักษณะคือ การบวกและการคูณ ของจำนวนเลขที่นับได้ (finite number) จะทำได้ก็ต่อเมื่อจำนวนเลขเหล่านั้นเป็นกำลัง (power) ของจำนวนเต็มที่หารไม่ลงตัว (prime number) ดังนั้นจึงสามารถใช้กฎทางคณิตศาสตร์ โดยทั่วไปมาใช้กับการบวกและการคูณใน Galois field ได้ ในกรณีนำมาสร้างเป็นวงจรลอจิกที่มีระดับ 0 และ 1 สามารถบวกและคูณตามกฎเกณฑ์ดังในภาพที่ 3.4

การคูณและการบวกในภาพที่ 3.4 เรียกว่าการบวกและการคูณแบบโมดูโล-2 (modulo-2) ซึ่งค่า 2 จะมีค่าเท่ากับ 0 และ 1 มีค่าเท่ากับ –1 ทั้งนี้ สัญลักษณ์ของการบวก การคูณ ตัวเลข 0 และตัวเลข 1 จะรวมกันเป็นฟิลด์ (field) ซึ่งเรียกว่า binary field และสามารถเขียนเป็น GF(2)

ในการสร้างวงจรลอจิกจากการบวกและการคูณแบบโมดูโล 2 จากภาพที่ 3.4 สามารถใช้เกจเอ็กคลูซีฟท์-ออร์ สำหรับการบวก และใช้เกจแอนด์ สำหรับการคูณ

3.2.2 การตรวจสอบความถูกต้องของรหัสคำ

วิธีการตรวจสอบความถูกต้องของรหัสคำแบบพื้นฐานที่นิยมใช้กันทั่ว ๆ ไปคือ การตรวจสอบพาริตี้ (parity check) การตรวจสอบความถูกต้องของข้อมูลในลักษณะนี้ จะทำการเพิ่มข้อมูลตามหลังรหัสข่าวสาร (message) จำนวน 1 บิต ให้เป็นรหัสคำ (code word) ดังภาพที่ 3.5



ภาพที่ 3.5 การเพิ่มพาริตี้บิตให้กับรหัสข่าวสาร

การตรวจสอบพาริตี้จะทำได้ 2 ลักษณะ คือ

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

2. กำหนดให้ผลรวมแบบโมดูโล 2 ของทุกบิตในรหัสคำมีค่าเท่ากับศูนย์ จำนวนของเลข 1 ในรหัสคำหลังจากเพิ่มพาริตี้แล้วจะเป็นจำนวนคู่ เรียกว่า พาริตี้คู่ (even parity)

3.2.3 การตรวจสอบพาริตี้ (Parity check)

เพื่อให้เข้าใจถึงการสร้างรหัสเชิงเส้นให้ดูตัวอย่างง่ายๆ โดยจากการสร้างรหัสคำ (Code word) เริ่มจากข้อมูลซึ่งปล่อยให้ผ่านไปในรหัสคำโดยตรง และจะมีบิตตามหลังแบบบิตเดียว ซึ่งเป็นการคำนวณจากบิตข้อมูลทั้งหมด มีวิธีการกำหนดบิตสุดท้ายที่ตามหลังมา 2 วิธี คือ

1.กำหนดบิตสุดท้ายเพื่อให้ผลรวมแบบโมดูโล (Modulo) 2 ของบิตทั้งหมดในรหัสคำมีค่าเท่ากับหนึ่ง

2.กำหนดบิตสุดท้ายเพื่อให้ผลรวมแบบโมดูโล 2 ของบิตทั้งหมดในรหัสคำมีค่าเท่ากับศูนย์

ในกรณีแรกรหัสคำมีพาริตี้แบบคี่ (Odd parity) กล่าวคือ จำนวนบิตที่เป็นหนึ่งในรหัสคำเป็นจำนวนคี่ ส่วนในกรณีที่สองมีจำนวนบิตที่เป็นหนึ่งในรหัสคำเป็นคู่ (Even parity) บิตที่เพิ่มขึ้นนอกเหนือจากบิตข้อมูลเรียกว่าพาริตี้เช็ค (Parity check) และอาจเรียกได้ว่าเป็นการตรวจสอบแบบพาริตี้คู่หรือพาริตี้คี่

รหัสแบบพาริตี้คี่และคู่ จะแสดงในตารางที่ 3.1 และ 3.2 ตามลำดับ สำหรับกรณีที่มีบิตข้อมูล 3 ตำแหน่ง จะสังเกตเห็นว่ารหัสของตารางที่ 3.1 ไม่มีแถวที่เป็นศูนย์ทั้งหมด ซึ่งต้องเป็นส่วนหนึ่งของรหัสที่เป็นเชิงเส้น

ตารางที่ 3.1 รหัสพาริตี้แบบคี่

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

ดังนั้นการตรวจสอบพาริตี้แบบคี่ทำให้เกิดรหัสแบบไม่เป็นเชิงเส้น ในทางตรงกันข้ามรหัสที่ใช้พาริตี้ในตารางที่ 3.2 จะเป็นรหัสเชิงเส้น ระบบที่มีการผลิตการตรวจสอบพาริตี้แบบคู่โดยการเพิ่มบิตที่เป็นพาริตี้นั้นได้จากการบวกแบบโมดูโล 2 ของบิตในรหัสของข้อมูลข่าวสาร ตัวอย่างเช่น รหัสข้อมูลข่าวสาร 101 เมื่อทำการบวกแบบโมดูโล 2 ของบิตที่มีค่าเป็น 1,0 และ 1 จะได้ ผลลัพธ์คือ 0 ซึ่งจะเป็นค่าของบิตที่จะนำมาทำการต่อข้างท้ายของรหัสข้อมูลข่าวสาร กลายเป็น 1010

ตารางที่ 3.2 รหัสพาริตี้แบบคู่

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

ในการเพิ่มพาริตี้อีกหนึ่งบิตให้รหัสของข้อมูลข่าวสารนั้นปกติมักจะนำมาตรวจสอบได้ถ้าบิตผิดไปเพียงบิตเดียวเท่านั้น แต่ไม่สามารถนำมาแก้ไขบิตที่ผิดได้ ในการแก้ไขบิตที่ผิดไปจะต้อง

ใช้วิธีการเข้ารหัสที่เพิ่มพาริตี้บิตให้มากขึ้น อย่างเช่นการเข้ารหัสแบบบล็อคโค้ดเชิงเส้นที่จะได้กล่าวต่อไป

3.2.4 ความสามารถในการตรวจแก้บิตที่ผิดในรหัสเชิงเส้น

ในส่วนนี้จะได้กล่าวถึงคำศัพท์พื้นฐานที่ใช้ในการแก้บิตที่ผิดของรหัสเชิงเส้น

เวท (Weight) ของแฮมมิ่งสำหรับเวคเตอร์ v η - ทูเปิ้ลส์ 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 เป็นเวคเตอร์ η - ทูเปิ้ลส์ ค่าระยะห่างระหว่าง u และ v เขียนได้เป็น d(u,v) ระยะห่างของสองเวคเตอร์ใด ๆ คือ จำนวนบิตรหัส " 1 " ที่แตกต่างกันของเวคเตอร์ทั้งสอง เช่น

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

ดังนั้นพอสรุปได้ว่าระยะห่างแบบแฮมมิ่งระหว่างเวคเตอร์ u และ v จะเท่ากับเวทของแฮมมิ่งจากเวคเตอร์รวมกล่าวคือ

$$d(u + v) = \omega(u + v) \qquad (3.1)$$

สำหรับในรหัสเชิงเส้นในการหาระยะห่างของแต่ละคู่ในรหัสคำ ระยะห่างที่น้อยที่สุดเขียนย่อเป็น d_{min} ถ้า u และ v เป็นโค้ดเวคเตอร์สองชุดของรหัสเชิงเส้น โดย u + v ก็ยังเป็นโค้ดเวคเตอร์ เพราะเซทของทุกโค้ดเวคเตอร์เป็นซับสแปช (Subspace) ของทุก η - ทูเปิ้ลส์ ดังนั้น จากคำนิยาม

ที่ว่า ระยะห่างระหว่างโค้ดเวคเตอร์ทั้งสองคือ เวทของโค้ดเวคเตอร์ที่ 3 ก็จะได้ระยะห่างน้อยที่สุดของรหัสเชิงเส้นเท่ากับเวทต่ำสุดของโค้ดเวคเตอร์ที่ไม่เป็นศูนย์ ค่าระยะห่างน้อยที่สุด และเวทต่ำสุดจะเป็นตัวกำหนดความสามารถในการแก้รหัสบิตที่ผิดของรหัสเชิงเส้น

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

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

ซึ่ง e เป็นรูปแบบของรหัสที่ผิด (error pattern หรือ error vector) เมื่อ $e = r_i + v_i = 1$ นั้นก็หมายความว่าโค้ดเวคเตอร์เกิดความผิดพลาดตำแหน่งบิตที่ i^{th} แต่เนื่องจากในหนึ่งโค้ดเวคเตอร์มีรหัสอยู่ n บิต จึงทำให้เกิดความผิดพลาด 2^n รูปแบบที่แตกต่างกัน ไม่นับรูปแบบที่มีทุกบิตเป็นศูนย์

ทางด้านรับตัวถอดรหัสมีหน้าที่ในการตรวจหาโค้ดเวคเตอร์ที่ส่งมาจากโค้ดเวคเตอร์ r ที่รับได้ สำหรับการถอดรหัสโดยใช้วิธีแม็กซิมั่มไลด์รีฮูดนั้น ตัวถอดรหัสจะตรวจสอบว่า v เป็นเวคเตอร์ที่ใช้ในการส่ง ซึ่งจะมีค่าเข้าใกล้เวคเตอร์ r โดยอาศัยดูจากระยะห่างของแฮมมิ่ง ตัวถอดรหัสสามารถทำการแก้ไขรหัสที่ผิดจำนวน t บิตในโค้ดเวคเตอร์ที่รับเข้ามา โดยที่ $2t + 2 \geq d_{min} \geq 2t + 1$ ตัวถอดรหัสสามารถทำการแก้ทุกรูปแบบที่ผิดไป t บิต จากเวคเตอร์ r ที่รับได้ ซึ่งแสดงให้เห็นได้ดังนี้ ให้ v เป็นโค้ดเวคเตอร์ที่ต้องการส่งและ u เป็นโค้ดเวคเตอร์ใด ๆ ระยะห่างของแฮมมิ่งระหว่าง u, v และ r จะต้องเป็นไปตามสมการ

$$d(v,r) + d(u,r) \geq d(u,v) \qquad (3.2)$$

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

$$d(u,r) \geq 2t + 1 - t'$$
$$d(u,r) \geq t + 1 \qquad (3.3)$$
$$d(u,r) \geq t'$$

จากสมการที่ 2.3 แสดงให้เห็นว่ารูปแบบของรหัสที่มีบิตผิดไป t บิตหรือน้อยกว่า เวคเตอร์ r ที่รับได้จะเข้าใกล้เวคเตอร์ v กว่าโค้ดเวคเตอร์ u ดังนั้นตัวถอดรหัสจะสามารถแก้ไขรหัสที่ผิดได้ถูกต้องตามความผิดพลาดของบิตที่ผิดไป ตัวถอดรหัสไม่สามารถจะแก้ทุกรูปแบบที่ผิดไป 1 บิต เมื่อ $1 \geq t+1$ โดยปกติแล้วการแก้รหัสผิดของโค้ดเชิงเส้นจะทำได้เมื่อ

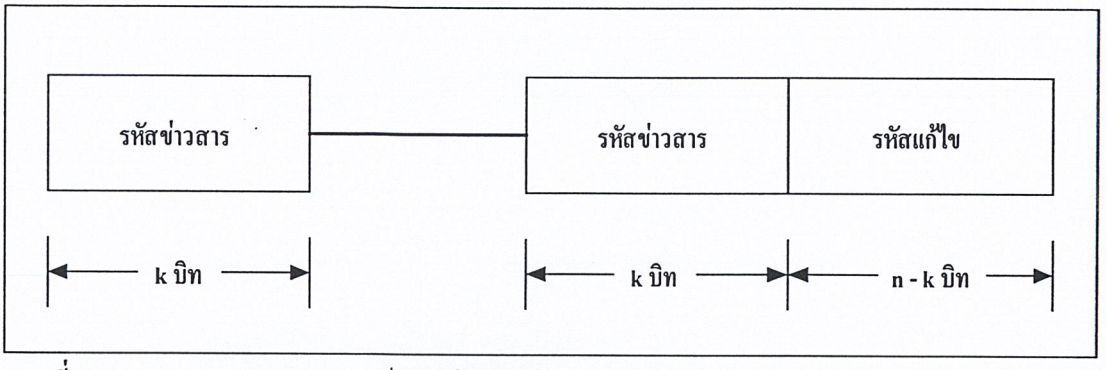
$$t = (d_{min} - 1) / 2 \qquad (3.4)$$

โดย $t = (d_{min} - 1) / 2$ เป็นค่าจำนวนเต็มไม่คิดทศนิยม และสามารถตรวจสอบรหัสผิดที่เกิดขึ้นถึง $(d_{min} - 1)$ บิตในแต่ละรหัสคำ

3.3 บล็อคโค้ดเชิงเส้น (Linear block code)

จากรหัสข่าวสาร (Message) ที่แต่ละบล็อคมีขนาด k บิตซึ่งพบว่าจะให้ข่าวสารที่แตกต่างกันได้ถึง $2^k - 1$ ข่าวสาร (ยกเว้นบล็อคที่มีรหัสข่าวสารหมดจะไม่มีการนำมาใช้) แต่ละบล็อคข่าวสารจะถูกนำมาเข้ารหัสเป็นบล็อคขนาด n บิต โดยจะมี $n - k$ บิต ที่เพิ่มเข้าไปให้รหัสข่าวสาร บิตเหล่านี้ที่เพิ่มเข้าไปในแต่ละบล็อคจะเป็นพาริตี้หรือบางทีเรียกว่ารหัสแก้ไข ที่จะถูกนำมาใช้ในการตรวจสอบและตรวจสอบบิตที่ผิดไป และค่าของ $n - k$ บิตจะขึ้นกับรหัสข่าวสารโดยตรง

บล็อคข่าวสารขนาด n บิตที่ได้จากการเข้ารหัสนี้จะ เรียกว่ารหัสคำ (Code word) ถ้าหากว่ารหัสคำมีบิตของข่าสารเดิมปรากฏอยู่ใน k บิตเริ่มต้นรหัสนั้น จะเรียกว่าซิสเต็มเมติกซ์ (Systematic code) ยิ่งไปกว่านั้นถ้าหากแต่รหัสคำจากจำนวน 2^k รหัสคำที่เข้ารหัสไว้ เกิดจากการรวมกันของ k เวคเตอร์ ของรหัสแบบอิสระเชิงเส้น (linearly independent) รหัสดังกล่าวจะเรียกว่า รหัสบล็อคโค้ดเชิงเส้น [11] ซึ่งรูปแบบของบล็อคโค้ดเชิงเส้นจะแสดงได้ดังภาพที่ 2.4



ภาพที่ 3.6 แสดงรูปรหัสคำของบล็อคโค้ดเชิงเส้น

3.3.1 แมทริกซ์ตัวกำเนิด (Generator matrix)

สำหรับซับสเปซ S ของ V_n และแต่ละ η- ทูเปิ้ลส์ (Tuples) ของ S เป็นการรวมแบบเชิงเส้นของ $v_1,v_2,\ldots,v_k$ กล่าวคือ

$$u = m_1v_1 + m_2v_2 + \ldots + m_kv_k \tag{3.5}$$

เมื่อ $m_i = 0$ สำหรับ $i = 1,2,\ldots,k$ ซับสเปซนี้มีขนาด k มิติของ V_n ซึ่งประกอบด้วย 2^k ของ η – ทูเปิ้ลส์ จากข้อกำหนดที่ผ่านมาซึ่งสามารถอธิบายถึงรหัสเชิงเส้นของ 2^k รหัสคำโดยเซทของ k โค้ดเวคเตอร์ที่เป็นข้อกำหนดอิสระเชิงเส้น ถ้าจัด k รหัสคำเป็นอิสระต่อกันได้แมทริกซ์ $k \times n$

$$G = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_k \end{bmatrix} = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & & & \\ v_{k1} & v_{k2} & \cdots & v_{kn} \end{bmatrix} \tag{3.6}$$

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

$$u = mg$$

$$= (m_1,m_2,\ldots\ldots,m_k)\begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_3 \end{bmatrix} \tag{3.7}$$

$$= m_1v_1 + m_2v_2 + \ldots\ldots + m_kv_k$$

ดังนั้นรหัสที่สอดคล้องกับชุดข่าวสาร $(m_1,m_2,\ldots\ldots m_k)$ เกิดจากการรวบรวมแบบเชิงเส้นของแถวใน G กลุ่มแถวต่างๆของแมทริกซ์ G จะเป็นตัวผลิตรหัสเชิงเส้น และเราเรียกแมทริกซ์ G ว่าแมทริกซ์ตัวกำเนิดรหัส รหัสเชิงเส้นที่กล่าวนี้เรียกว่ารหัส (n,k) โดยในแต่ละบล็อกจะมีข่าวสารอยู่ k บิต ที่ถูกเข้ารหัสเป็นรหัสคำที่มีความยาวขนาด n บิต

ลักษณะของแมทริกซ์ตัวกำเนิดขนาด k× n ที่ใช้สร้างรหัสเชิงเส้น (n,k) แสดงได้ดังสมการที่ (2.8)

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & p_{11} & p_{12} & \cdots & p_{1,n-k} \\ 0 & 1 & 0 & 0 & \cdots & 0 & p_{21} & p_{22} & \cdots & p_{2,n-k} \\ 0 & 0 & 1 & 0 & \cdots & 0 & p_{31} & p_{32} & \cdots & p_{3,n-k} \\ \vdots & & & & & & \vdots & & & \\ 0 & 0 & 0 & 0 & \cdots & 1 & p_{k1} & p_{k2} & \cdots & p_{k,n-k} \end{bmatrix} \qquad (3.8)$$

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

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

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

$$u = (u_1,u_2,u_3,\ldots\ldots,u_n)$$
$$= (m_1,m_2,\ldots\ldots,m_k)\, G$$

$$= (m_1,m_2,\ldots\ldots,m_k) \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 & p_{11} & p_{12} & \cdots & p_{1k} \\ 0 & 1 & 0 & 0 & \cdots & 0 & p_{21} & p_{22} & \cdots & p_{2k} \\ \vdots & & & & & & \vdots & & & \\ 0 & 0 & 0 & 0 & \cdots & 1 & p_{k1} & p_{k2} & \cdots & p_{k,n-k} \end{bmatrix} \qquad (3.9)$$

จากการคูณของแมทริกซ์จะได้

$$u_i = m_i \qquad \text{สำหรับ } i = 1,2,.....k \qquad (3.10a)$$

และ

$$u_{k+j} = p_{1j}m_1 + p_{2j}m_2 + ...+ p_{kj}m_k \qquad (3.10b)$$

สำหรับ $j = 1,2,.....,n-k$ จากสมการที่ 2.10a และ 2.10b จะพบว่ารหัส k บิตแรกของรหัสคำ คือ รหัสของข้อมูลข่าวสาร ส่วน (n-k) บิตหลังเป็นฟังก์ชั่นเชิงเส้นของรหัสข่าวสาร ซึ่งเรียกว่า รหัสแก้ไข (n-k) บิตของ u หรือ รหัสตรวจสอบพาริตี้ (Parity check digits) ของรหัสคำ สมการที่ 2.10b เรียกว่าสมการพาริตี้ของรหัส

3.3.2 แมทริกซ์ในการตรวจสอบพาริตี้ (Parity check matrix)

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

$$H = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{n-k} \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1n} \\ h_{21} & h_{22} & \cdots & h_{2n} \\ \vdots & & & \\ h_{n-k,1} & h_{n-k,2} & \cdots & h_{n-k,n} \end{bmatrix} \qquad (3.11)$$

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

$$uH^T = \begin{pmatrix} 0 & 0 & \cdots & 0 \end{pmatrix} \qquad (3.12)$$

หรือ

$$u.h_i = u_ih_{i1} + u_2h_{i2} + u_nh_{in} = 0 \qquad (3.13)$$

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

$$H = \begin{bmatrix} p_{11} & p_{21} & \cdots & p_{k1} & 1 & 0 & 0 & \cdots & 0 \\ p_{21} & p_{22} & \cdots & p_{k2} & 0 & 1 & 0 & \cdots & 0 \\ \vdots & & & & \vdots & & & & \\ p_{1,n-k} & p_{2,n-k} & \cdots & p_{k,n-k} & 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \qquad (3.14)$$

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

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

$$\begin{aligned} U_{k+j} &= p_{1j}u_1 + p_{2j}u_2 + ...+p_{kj}u_k \\ &= p_{1j}m_1 + p_{2j}m_2 + ...+ p_{kj}m_k \end{aligned} \qquad (3.15)$$

เมื่อ $j = 1,2,.....,n-k$ ซึ่งสมการข้างบนเป็นสมการเดียวกันกับสมการที่ 2.10b ในการออกแบบรหัสเชิงเส้นนั้นแมทริกซ์ P จะถูกเลือกเพื่อให้มีคุณสมบัติในการแก้บิตที่ผิด

3.3.3 ซีนโดรม

บล็อกของขนาดคำขนาด n บิต เมื่อทำการส่งออกไปในตัวกลางจะเกิดสัญญาณรบกวนทำให้บางบิตของข้อมูลผิดไป ซึ่งรูปแบบของบิตที่ผิดไป บางครั้งเรียกว่า โคเซ็ท (Coset) มีหลายรูปแบบถ้าหาก u เป็นรหัสคำที่ต้องการส่ง โดย u มีระยะห่างต่ำสุดตามเงื่อนไขสมการ (2.4) และ e_ℓ เป็นรูปแบบที่ผิดไปในระหว่างการติดต่อสื่อสาร ทางด้านรับจะได้รหัสคำเป็น r กล่าวคือ

$$r = e_\ell + u \tag{3.16}$$

การคำนวณหาซีนโดรมของรหัสคำที่รับได้ ทำได้โดย

$$\begin{aligned} S &= uH^T \\ &= (e_\ell + u)H^T = e_\ell H^T + uH^T \end{aligned} \tag{3.17}$$

ถ้าหาก u เป็นรหัสคำที่ถูกต้องจะพบว่า

$$uH^T = 0 \tag{3.18}$$

ดังนั้นซีนโดรมคือ $S = e_\ell H^T$

โดยปกติแล้วแต่ละรูปแบบของบิตที่ผิดไปถ้าไม่ซ้ำกันจะให้ค่าซีนโดรมที่ไม่เท่ากัน ในการพิสูจน์ทำได้โดยถ้าสมมติว่ารูปแบบของบิตที่ผิดไปคนละรูปแบบแต่ให้ซีนโดรมเท่ากัน อย่างเช่น รูปแบบของบิตที่ผิดไป e_ℓ กับ e_t เมื่อซีนโดรมคือ

$$S_1 = e_\ell H^T$$
$$S_2 = e_\ell H^T$$

แต่ $S_1 = S_2$ จะให้

$$e_\ell H^T = e_t H^T \tag{3.19}$$

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 จะเป็นศูนย์ก็ต่อเมื่อทุกบิตใน e_ℓ กับ e_t จะเหมือนกันแบบบิตต่อบิต จึงสรุปได้ว่า

$$e_\ell = e_t$$

ซึ่งขัดกับสมมติฐานที่ว่า e_ℓ กับ e_t เป็นคนละรูปแบบ

ดังนั้นพอสรุปได้ว่าถ้าหากรูปแบบบิตที่ผิดไปคนละรูปแบบจะให้ค่าซินโดรมที่แตกต่างกันออกไป กล่าวคือ

$$e_\ell H^T \neq e_t H^T \tag{3.20}$$

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

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

2. เปิดตารางของซินโดรมเพื่อดึงเอารูปแบบที่ผิดที่ให้ซินโดรมเหมือนกับที่คำนวณได้ถ้าหากเป็นรูปแบบของ e_ℓ

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

4. ดึง k บิตแรกของรหัสคำของ u ซึ่งจะเป็นรหัสข่าวสาร m ที่ส่งมา

ตัวอย่างเช่น ถ้าหากมีแมทริกซ์ตัวกำเนิด G เป็น

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

สามารถแปลงเป็นแมทริกซ์ตรวจสอบพาริตี้ H เป็น

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

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

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

สำหรับรูปแบบต่างๆของรหัสที่ผิดไปเพียงบิตเดียวกับซีนโดรมที่สอดคล้องพอสรุปได้ดังตารางข้างล่าง

ตารางที่ 3.3 แสดงซีนโดรมที่ได้จากรูปแบบรหัสที่ผิดไปหนึ่งบิต

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

ถ้าสมมติว่าทางด้านส่งรหัสข่าวสาร m = [101] จะได้รหัสคำ u

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

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

เมื่อทำการส่งรหัสคำ u ผ่านช่องส่งสัญญาณที่มีการรบกวน จะพบว่าทางด้านรับจะได้รับรหัสคำ r เป็น $\begin{bmatrix}1 & 0 & 0 & 1 & 0 & 1\end{bmatrix}$ ทางด้านรับจะทำการคำนวณหาค่าซีนโดรมจากรหัสค่า r ถ้าค่าซีนโดรมเป็นศูนย์หมดทุกบิตแสดงว่ารหัสคำ r ที่ได้รับกับรหัสคำ u ที่ส่งเป็นรหัสคำเดียวกัน

แต่ถ้าหากซีนโดรมไม่เป็นศูนย์ก็ต้องทำการเปิดตารางที่ 3.3 เพื่อหารูปแบบรหัสที่ผิดไปที่สอดคล้องกับการคำนวณซีนโดรมทำได้โดย

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

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

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

จากวิธีการดึงรหัสคำที่ถูกต้องกลับคืนมาจากรหัสคำที่ผิด จึงได้ถูกนำมาใช้เป็นหลักการเข้ารหัสลับให้กับข้อมูลในวิทยานิพนธ์ฉบับนี้ หลักการทำงานของการแก้รหัสลับคือ จากรหัสคำที่มีอยู่จะถูกนำมาบวกกับรูปแบบรหัสที่ผิดต่างๆ ที่กำหนดไว้ โดยแต่ละรูปแบบของรหัสที่ผิดจะมีซีนโดรมที่สอดคล้องเก็บเป็นตารางข้อมูลเอาไว้ ซึ่งรหัสคำที่ผิดมีหลายรูปแบบ วิธีการนี้ทางด้านรับจะมีรหัสคำที่ผิดอยู่ตลอดเวลา แต่จะสามารถนำเอารหัสคำที่ถูกต้องกลับคืนมาได้ถ้าหากรู้ว่าแมทริกซ์ G คืออะไร ในการเพิ่มความซับซ้อนการเข้ารหัสลับ จะมีการเรียงลำดับตำแหน่งบิตในแต่ละรหัสคำที่ผ่านการบวกรูปแบบบิตที่ผิดไปเรียบร้อยแล้ว รูปแบบของการเรียงสลับบิตที่ผิดจะมีอยู่หลายรูปแบบเช่นกัน การเลือกรูปแบบการเรียงสลับบิตจะใช้ค่าของซีนโดรมเป็นตัวเลือก สำหรับรายละเอียดในการเข้ารหัสลับจะได้กล่าวในบทต่อไป

บทที่ 4
การออกแบบตัวเข้ารหัสและตัวถอดรหัสลับ

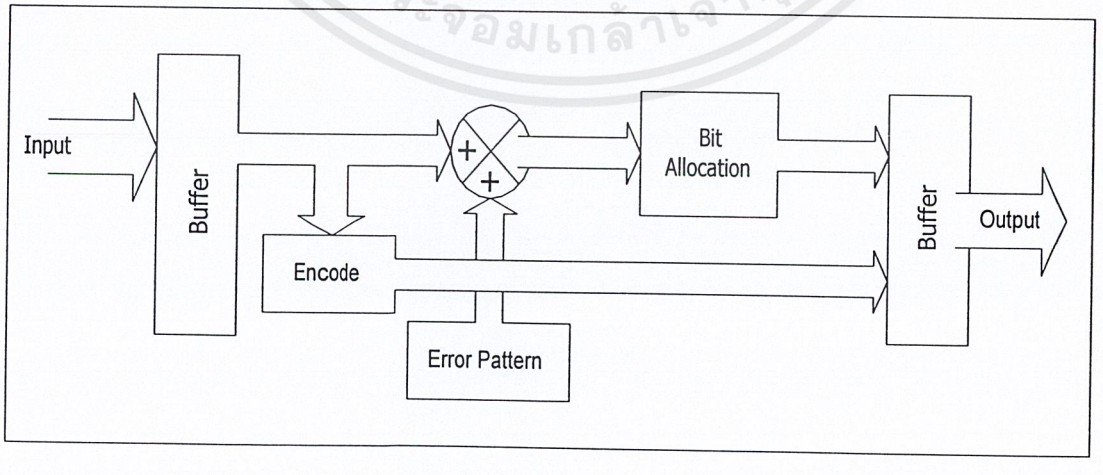
4.1 กล่าวนำ

ในบทนี้จะกล่าวถึงการออกแบบและการสร้างตัวเข้ารหัสและถอดรหัสลับ ซึ่ง จะมีส่วนต่าง ๆ ได้แก่ ภาคแปลงสัญญาณอนาล็อกให้เป็นสัญญาณดิจิตอล(ADC), ภาคแปลงสัญญาณดิจิตอลให้เป็นสัญญาณอนาล็อก(DAC), ภาคกำเนิดรูปแบบที่ผิด, ภาคสลับตำแหน่งบิตข้อมูล, ภาคแก้รหัสที่ผิดแบบบล็อกโค้ดเชิงเส้นโดยอาศัยซีนโดรม, ภาคสลับตำแหน่งบิตข้อมูลกลับ

4.2 การประยุกต์ใช้ DSP ตามหลักการและทฤษฎี

ในการออกแบบนั้นจะเป็นการนำข้อมูลเสียงเข้ามาทางอินพุต (Input Port) ของDSK มาทำการแปลงให้มีรูปแบบของข้อมูลแตกต่างไปจากข้อมูลเดิม โดยทำการแปลงข้อมูลข่าวสารที่เข้ามาทางด้านอินพุตขนาด 5 บิตให้เป็นรหัสคำที่มีขนาด 8 บิต แล้วทำการบวกรหัสคำที่ได้กับรูปแบบที่ผิดพลาดที่สร้างขึ้นจำนวน 7 รูปแบบหลังจากนั้นจะทำการสลับตำแหน่งของรหัสคำซึ่งมีรูปแบบการสลับตำแหน่ง 8 รูปแบบที่ทำการบวกรูปแบบที่ผิดพลาดแล้ว ก็จะทำให้ได้รหัสข้อมูลที่ทำการเข้ารหัสแล้ว ซึ่งสามารถแบ่งเป็นส่วน ๆ ของการสร้างรหัสลับตามบล็อกไดอะแกรมภาพที่ 4.1 ได้ดังนี้

4.2.1 การออกแบบตัวเข้ารหัสลับ(Encoder)



ภาพที่ 4.1 แสดงแผนภาพการเข้ารหัสลับ

เป็นส่วนที่ทำหน้าที่ในการแปลงรหัสข้อมูลเสียงที่เข้ามาทางอินพุตที่มีขนาดของข้อมูล 5 บิต มาทำการเปลี่ยนแปลงให้เป็นรหัสคำ ในการแปลงนี้จะใช้วิธีการของรหัสบล็อคโค้ดเชิงเส้น โดยจะเป็นการนำข้อมูลเสียง $\underline{m}$ ที่มีขนาด 5 บิต ไปทำการคูณกับแมทริกซ์ตัวกำเนิด G (Generator matrix) จะได้ผลลัพธ์เป็นรหัสคำที่มีขนาด 8 บิต จะมีจำนวนของบิตที่เพิ่มขึ้นมาจากรหัสของข้อมูลเสียง 3 บิต โดยรหัสที่เพิ่มขึ้นมานี้จะถูกนำมาใช้ในการแก้รหัสผิดซึ่งจะ กล่าวในหัวข้อต่อไป ภายในแมทริกซ์ตัวกำเนิด G จะประกอบด้วยแมทริกซ์เอกลักษณ์ I_k ที่มีขนาด 5 x 5 และแมทริกซ์พาริตี้ P ที่มีขนาด 5 x 3 จะแสดงในสมการที่ 4.1

$$G = [I_k : P] \tag{4.1}$$

แมทริกซ์ตัวกำเนิด G ที่ใช้แสดงในภาพที่ 4.2

$$G = \left[\begin{array}{ccccc:ccc} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{array}\right]$$

ภาพที่ 4.2 แสดงแมทริกซ์ตัวกำเนิด

จากรหัสข้อมูลข่าวสาร $\underline{m}$ ที่มีขนาด 5 บิต นำมาทำการคูณกับแมทริกซ์ตัวกำเนิด G จะทำให้ได้รหัส 8 บิต ดังแสดงในสมการที่ 4.2

$$U = mG \tag{4.2}$$

ทำให้สามารถแสดงวิธีการสร้างรหัสคำ U ในภาพที่ 4.3

$$U = [\, m_0 \;\; m_1 \;\; m_2 \;\; m_3 \;\; m_4 \,] \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

ภาพที่ 4.3 แสดงการสร้างรหัสคำ

4.2.2 ส่วนของรูปแบบผิดพลาด (Error Pattern)

สำหรับแมทริกซ์ตัวกำเนิดในภาพที่ 4.2 นั้นจะแก้รหัสผิดได้เพียงบิตเดียวเท่านั้น แต่ในการเข้ารหัสลับนี้จะใช้รูปแบบของบิตที่ผิดไปสองบิต ใน 5 บิตแรกของรหัสคำซึ่งเป็นบิตของข่าวสาร ดังนั้นจึงไม่เป็นไปตามกฎของซีนโดรม กล่าวคืออาจมีรูปแบบของบิตที่ผิดไปมากกว่าหนึ่งรูปแบบที่ให้ซีนโดรมเหมือนกัน ดังแสดงในตารางที่ 4.1 เนื่องจากการเข้ารหัสบล็อคโค้ดเชิงเส้น (8,5) จะมีซีนโดรมได้ 3 บิต หรือมีซีนโดรมได้ 7 รูปแบบที่แตกต่างกัน(ไม่นับซีนโดรมที่เป็นศูนย์หมด) เนื่องจากรหัสที่ผิดไปสองบิตใน 5 บิตแรก จะมีรูปแบบที่ไม่เหมือนกัน 10 รูปแบบ ดังแสดงในคอลัมม์ที่สองของตารางที่ 4.1 ในกรณีที่รูปแบบที่ผิดมีหลายรูปแบบจะเลือกเอารูปแบบบิตที่ให้ค่าสูงสุด เนื่องจากการเข้ารหัสลับสัญญาณเสียง ถ้าเลือกบิตที่มีค่าสูงจะให้แอมปลิจูด (Amplitude) ของสัญญาณเสียงที่รวมกับรหัสที่ผิดแล้วเกิดการเปลี่ยนแปลงสูง ดังนั้นรูปแบบผิดที่เลือกไว้และซีนโดรมที่สอดคล้องพอสรุปได้ดังตารางที่ 4 .1

$$r = u \oplus e \tag{4.3}$$

ตารางที่ 4.1 รูปแบบผิดพลาด

ค่าซีนโดรม	ตำแหน่งบิตที่ผิด	เลือกรูปแบบ	รูปแบบบิตที่ผิด
001	(0,1)(0,4)	(0,1)	11000000
010	(3,4)(0,2)	(0,2)	10100000
011	(1,2)	(1,2)	01100000
100	(2,4)(0,3)	(0,3)	10010000
101	(1,3)	(1,3)	01010000
110	(2,3)	(2,3)	00110000
111	(1,4)	(1,4)	01001000

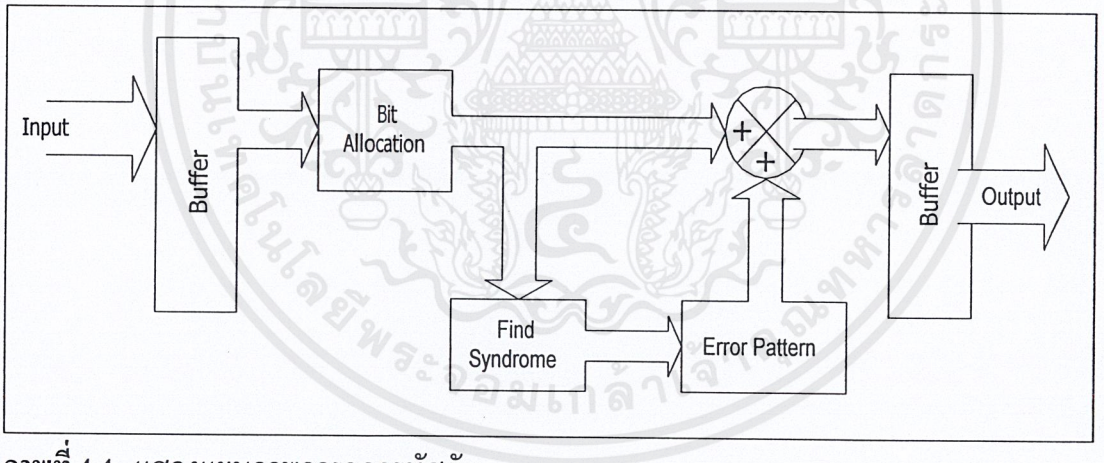
4.2.3 ส่วนของการสลับตำแหน่งบิต (Bit Allocation)

เป็นการสลับตำแหน่งของข้อมูลที่เป็นรหัสคำที่ทำการบวกกับรูปแบบผิดพลาดแล้วโดยจะทำการสลับเฉพาะ 5 บิตแรกเท่านั้น เพื่อเพิ่มความซับซ้อนในการป้องกันข้อมูลให้มีความปลอดภัยยิ่งขึ้น โดยการนำ 3 บิตหลังที่ได้จากรหัสคำมาเป็นตัวเลือกรูปแบบของการสลับตำแหน่งบิตข้อมูลซึ่งมีรูปแบบการสลับบิต 8 รูปแบบดังแสดงในตารางที่ 4.2

ตารางที่ 4.2 การสลับตำแหน่งบิต

ลำดับรูปแบบ	ตำแหน่งบิตที่ทำการสลับ				
	MSB				LSB
000	5	3	4	7	6
001	6	4	3	5	7
010	4	3	7	6	5
011	5	7	3	6	4
100	6	3	4	7	5
101	3	4	5	6	7
110	5	3	6	7	4
111	3	4	7	5	6

4.2.4 การออกแบบตัวถอดรหัสลับ (Decoder)



ภาพที่ 4.4 แสดงแผนภาพการถอดรหัสลับ

ในการออกแบบตัวถอดรหัสลับเป็นการนำข้อมูลเสียงที่ผ่านการเข้ารหัสแล้วกลับคืนมาให้ได้ข้อมูลเดิม โดยทำการถอดรหัสข้อมูลข่าวสาร จากข่าวสารที่ทำการเข้ารหัสแล้วมีขนาด 8 บิต เข้ามาทางอินพุตโดยเริ่มจากการสลับตำแหน่งของข้อมูลกลับให้ข้อมูลที่ได้มีตำแหน่งเหมือนเดิมโดยใช้ข้อมูล 3 บิตที่เพิ่มเข้ามาเป็นตัวเลือกรูปแบบที่จะทำการสลับตำแหน่งบิตกลับแล้วทำการคำนวณหาค่าซินโดรมที่จะมาเป็นตัวเลือกรูปแบบที่ผิดพลาดดังแสดงในตารางที่ 4.1 มาทำการบวกแบบเอกคลูซีฟออร์กับข้อมูลข่าวสารนั้น ทำให้ได้ข้อมูลข่าวสารเดิมกลับมาแบ่งเป็นแต่ละขั้นตอนดังแสดงในภาพที่ 4.4

4.2.5 ส่วนของการสลับตำแหน่งบิตกลับ(Bit Reallocation)

เป็นส่วนที่ทำหน้าที่ในการสลับตำแหน่งของบิตข้อมูล ที่เป็นรหัสคำที่ผิดไปกลับคืนมาให้ได้รับตำแหน่งที่ถูกต้องตรงกับตำแหน่งเดิม โดยจะทำการสลับตำแหน่งข้อมูลกลับเฉพาะ 5 บิตแรกที่ถูกสลับไว้เท่านั้น และ 3 บิตที่เหลือจะเป็นตัวเลือกรูปแบบของการสลับตำแหน่งข้อมูลกลับซึ่งจะมีรูปแบบการสลับตำแหน่งบิตข้อมูลดังแสดงในตารางที่ 4.3

ตารางที่ 4.3 การสลับตำแหน่งบิตกลับ

ลำดับรูปแบบ	ตำแหน่งบิตที่ทำการสลับ				
	MSB				LSB
000	4	3	7	5	6
001	3	7	4	6	5
010	5	4	3	7	6
011	6	4	7	3	5
100	4	7	3	5	6
101	3	4	5	6	7
110	4	5	7	3	6
111	5	3	4	6	7

4.2.6 ส่วนของการคำนวณหาซินโดรม

ในส่วนของการคำนวณหาซินโดรมประกอบไปด้วย แมทริกซ์ตรวจสอบพาริตี้ H ที่ถูกนำมาใช้ในการคำนวณหาค่าซินโดรมสำหรับนำมาใช้ในการถอดรหัสที่ผิดไป โดยแมทริกซ์ตรวจสอบพาริตี้ H ประกอบขึ้นจากแมทริกซ์ทรานสโพสของแมทริกซ์พาริตี้ P ที่มีขนาด $(n-k) \times k$ และแมทริกซ์เอกลักษณ์ I_{n-k} คือ

$$H = [\, P^T I_{n-k} \,] \tag{4.4}$$

แมทริกซ์ในการตรวจสอบพาริตี้ H แสดงในภาพที่ 4.5

$$H = \left[\begin{array}{ccccc:ccc} 1 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{array}\right]$$

ภาพที่ 4.5 แสดงแมทริกซ์พาริตี้

ในการคำนวณหาซีนโดรม S ได้จากการนำรหัสคำที่ได้รับมาทำการคูณกับแมทริกซ์ทรานโพสของ H ซึ่ง H^T เป็นแมทริกซ์โดย 4 แถวสุดท้ายของ H^T จะเป็นแมทริกซ์เอกลักษณ์ โดยใน 8 แถวแรกได้จากการทำทรานโพสแมทริกซ์ P ของแมทริกซ์ตัวกำเนิด โดยทั้ง 8 แถวจะแตกต่างกันและ r เป็นรหัสที่ได้รับเข้ามาซึ่งสามารถทำการคำนวณได้ดังสมการที่ 4.5

$$S = r H^T \tag{4.5}$$

ซีนโดรมสามารถทำการหาได้ดังภาพที่ 4.6

$$S = [r_0 \quad r_1 \quad \cdots \quad r_6 \quad r_7] \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

ภาพที่ 4.6 แสดงการหาค่าซีนโดรม

แต่เนื่องจากแต่ละแถว (ROW) ของแมทริกซ์ G และแมทริกซ์ H ตั้งฉากกันจะให้อินเนอร์โปรดัก (Inner product) เป็นศูนย์ ก็หมายความว่า $u H^T = 0$ นั่นเอง ทำการแทนค่า r ด้วยสมการที่ 4.5 ได้ว่า

$$\begin{aligned} S &= (u \oplus e) H^T \\ S &= uH^T \oplus eH^T \\ S &= eH^T \end{aligned} \tag{4.6}$$

จากสมการที่ 4.5 และ 4.6 เป็นตัวบอกให้ทราบว่าค่าซีนโดรมที่คำนวณได้จากรหัสคำที่ผิดไปเนื่องจากรูปแบบผิดพลาด e ไปคูณกับ H^T นั้นจะให้ค่าซีนโดรมที่เหมือนกันกับการนำรูปแบบที่ผิดพลาด e ไปคูณโดยตรงกับ H^T ด้วยเหตุนี้ค่าของซีนโดรมจึงเป็นตัวบ่งบอกว่ารหัส r ที่ได้รับมีรูปแบบผิดพลาด e รูปแบบใดปรากฏอยู่ ดังนั้นรูปแบบการดึงรหัสคำที่ถูกส่ง u จะกลับคืนมาโดยการนำ e ไปทำการบวกเอกคลูซีฟออร์กับรหัส r อีกครั้งหนึ่ง ดังแสดงในสมการที่ 4.7 ซึ่งทำให้สามารถทำการถอดรหัสได้และได้ข้อมูลข่าวสารที่ถูกต้องกลับคืนมา

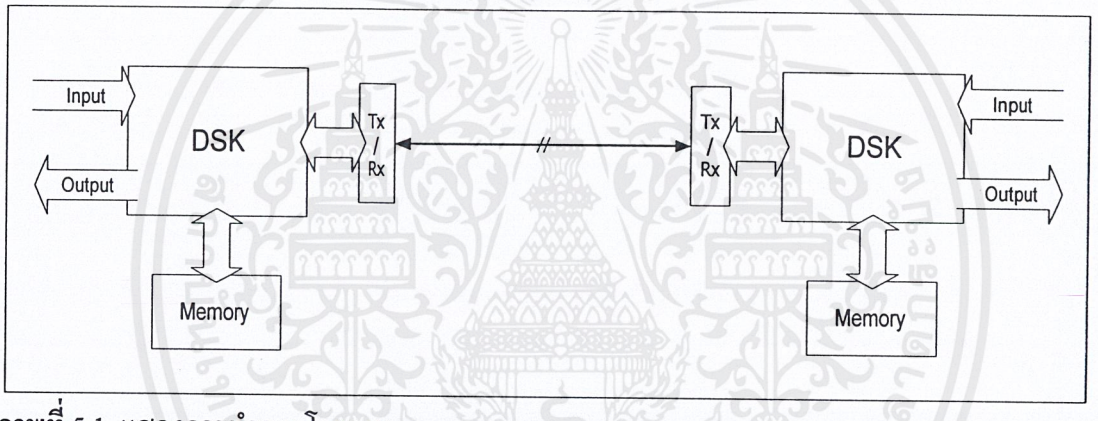
$$\begin{aligned} r \oplus e &= (u \oplus e) \oplus e \\ &= u \oplus (e \oplus e) \\ &= u \end{aligned} \tag{4.7}$$

บทที่ 5

ผลการทดลอง

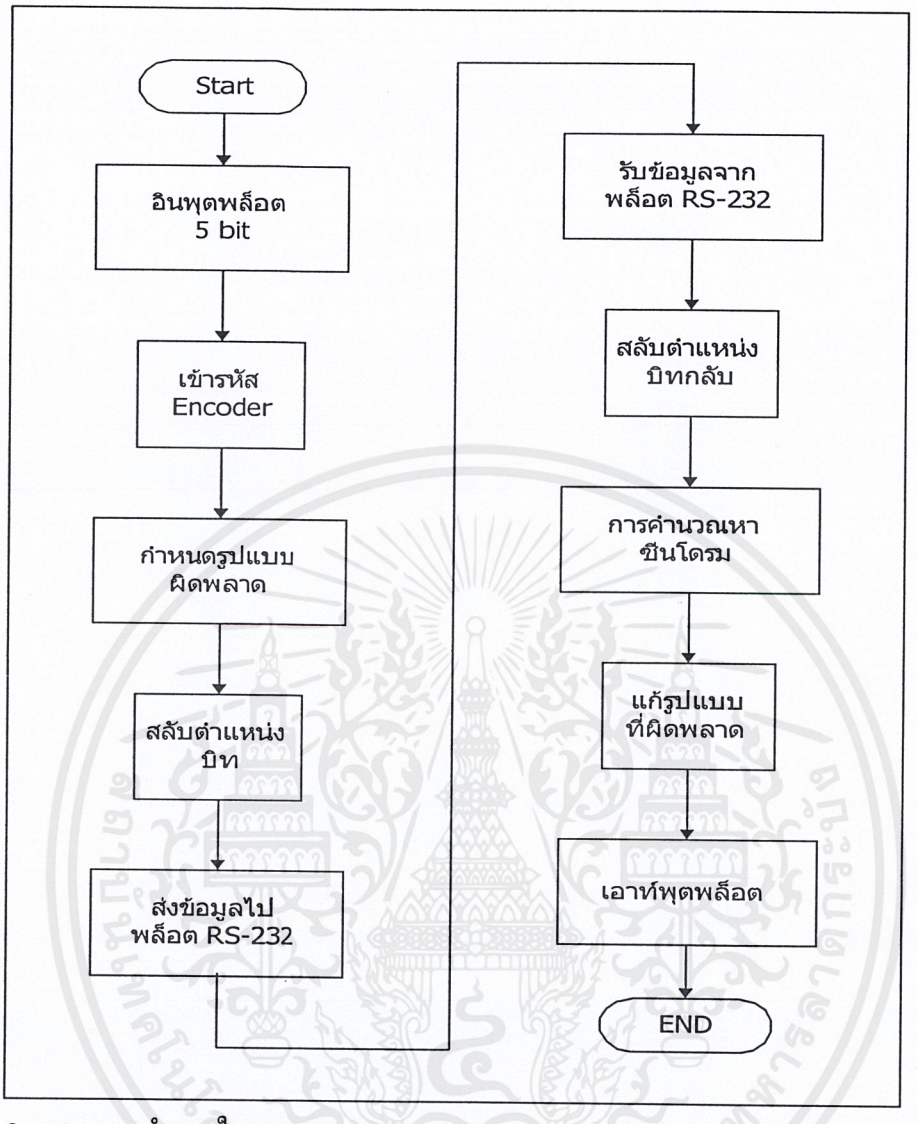
5.1 กล่าวนำ

การทดลองจะเป็นการประยุกต์ใช้งาน DSK รุ่น TMDS3200031 ใช้ Chip no. TMS320C31 ที่มีความเร็วในการประมวลผลสูง ซึ่งจะต้องเขียนโปรแกรมให้ DSK ทำการเข้ารหัสและถอดรหัสตามขั้นตอนที่กล่าวไว้ในบทที่ 3 โดยโปรแกรมที่เขียนนั้นจะเลือกใช้โปรแกรมภาษา C / C^{++} (ภาคผนวก ข) และได้มีการสร้างวงจรภายนอกเพิ่มเติมขึ้น เป็นวงจรในส่วนของการขยายหน่วยความจำและวงจรภาครับส่งข้อมูลดิจิตอล



ภาพที่ 5.1 แสดงการทำงานโดยรวม

ภาพที่ 5.1 จะเป็นการแสดงการทำงานโดยรวมของการเข้ารหัสสัญญาณ โดยจะทำการรับสัญญาณเสียงเข้ามาทางอินพุทพล็อตของ DSK เข้ามาประมวลผลตามโปรแกรมที่เขียนไว้ซึ่งจะเก็บอยู่ในส่วนของ Memory เนื่องจากข้อมูลที่ประมวลผลเสร็จจะเป็นสัญญาณดิจิตอลจึงต้องทำการส่งข้อมูลโดยใช้ IC ตระกูล UART (Universal Asynchronous Receiver / Transmitter) จะใช้เบอร์ 16C550 ที่มี FIFO ขนาด 16 Byte อีกข้างหนึ่งก็จะรับข้อมูลด้วยอัตรา Baud Rate ที่เท่ากัน จากนั้นส่งค่าที่ได้ไปให้ DSK ทำการ ถอดรหัสออกมาเป็นเสียงที่ถูกต้อง แล้วสัญญาณจะออกมาทางเอ้าท์พุตพล็อตของ DSK



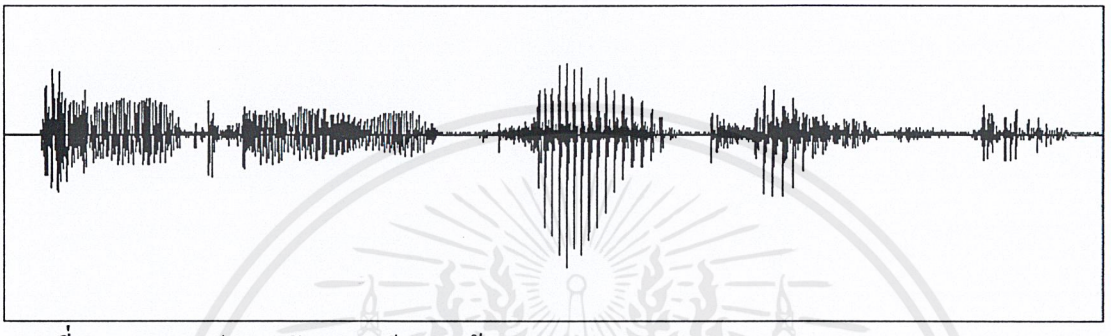
ภาพที่ 5.2 แสดงการทำงานใน 1 รอบ

การทดลองแผงวงจรตัวเข้ารหัสลับและถอดรหัสลับกระทำโดยการนำเอาสัญญาณที่อยู่ในรูปแบบต่าง ๆ มาทำการเข้ารหัสและถอดรหัสลับ สัญญาณอนาล็อก ได้แก่ สัญญาณไซน์ และสัญญาณเสียงพูด จากภาพที่ 5.2 แสดงการทำงานใน 1 รอบ ของการทำงานทั้งหมดซึ่งขั้นตอนต่าง ๆ จะกล่าวไว้ในบทที่ 4 ในส่วนของการเขียนโปรแกรมนั้นจะทำวนการทำงานในรูปแบบนี้ตลอด ซึ่งจะเสมือนเป็นการทำงานแบบ Full Duplex

5.2 ผลการทดลอง

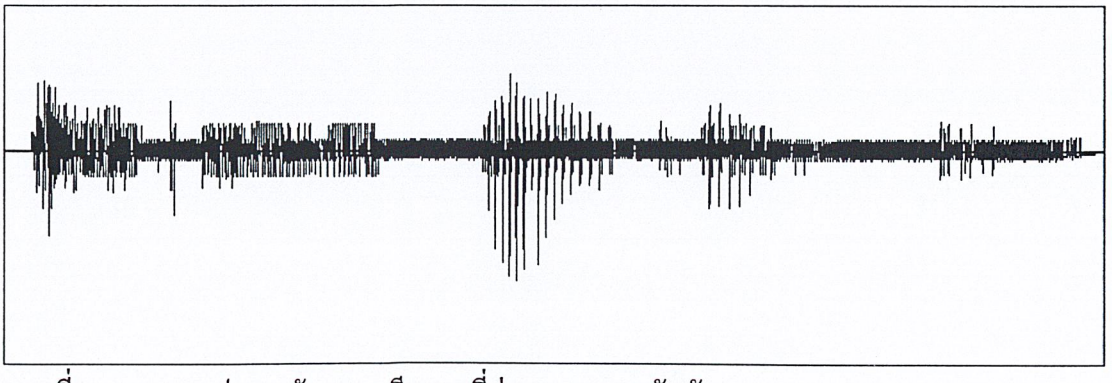
ผลการทดลองนั้นจะเป็นการเปรียบเทียบกันระหว่างสัญญาณ โดยจะเห็นว่ารูปร่างของสัญญาณมีความแตกต่างกันไปบ้างอันเนื่องมาจากการสูญเสียในระหว่างการแปลงสัญญาณ เนื่องจากใช้สัญญาณเพียง 5 บิตในการทดลองและสูญเสียในเรื่องของการรับส่งแบบอนุกรม

การทดลองในรูปแบบแรกใช้สัญญาณเสียงพูดเพียงอย่างเดียว โดยมีรูปแบบสัญญาณต้นแบบในภาพที่ 5.2 สำหรับนำมาทำการเข้ารหัสลับ โดยนำสัญญาณเข้าทาง Input Port ของ DSK แล้วทำการเข้ารหัสลับนำผลที่ได้จาก Output Port ดังแสดงในภาพที่ 5.3 และ ภาพที่ 5.4 เป็นการแสดงรูปของสัญญาณเสียงที่ผ่านการถอดรหัสแล้ว



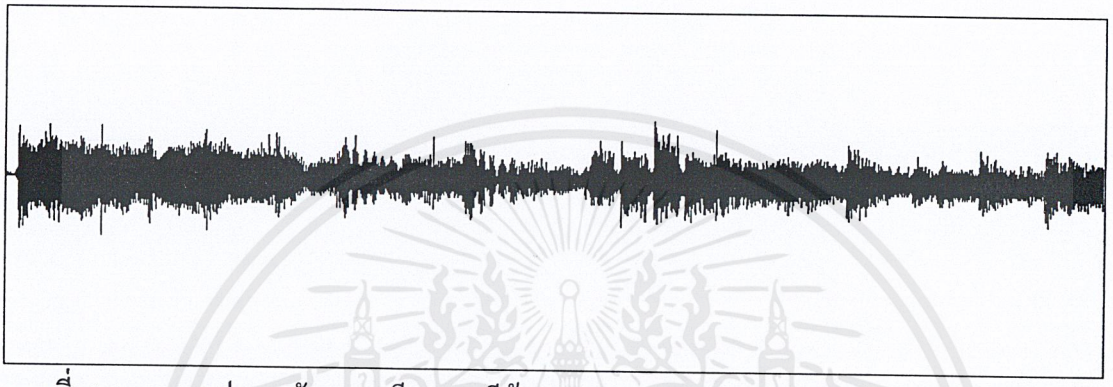
ภาพที่ 5.2 แสดงรูปแบบสัญญาณเสียงพูดต้นแบบ

ภาพที่ 5.3 แสดงรูปแบบสัญญาณเสียงพูดที่ผ่านการเข้ารหัสลับ



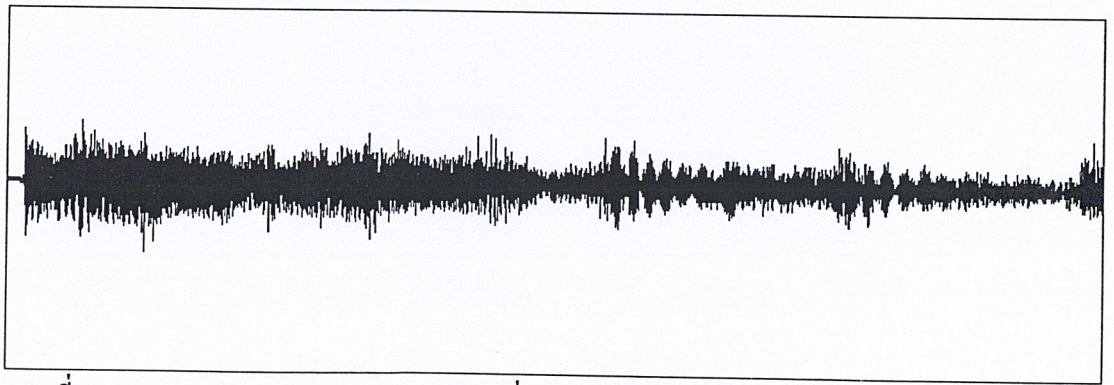
ภาพที่ 5.4 แสดงรูปแบบสัญญาณเสียงพูดที่ผ่านการถอดรหัสลับ

การทดลองในรูปแบบแรกใช้สัญญาณเสียงดนตรี โดยมีรูปแบบสัญญาณต้นแบบในภาพที่ 5.5 สำหรับนำมาทำการเข้ารหัสลับ โดยนำสัญญาณเข้าทาง Input Port ของ DSK แล้วทำการเข้ารหัสลับนำผลที่ได้จาก Output Port ดังแสดงในภาพที่ 5.6 และ ภาพที่ 5.7 เป็นการแสดงรูปของสัญญาณดนตรีที่ผ่านการถอดรหัสแล้ว



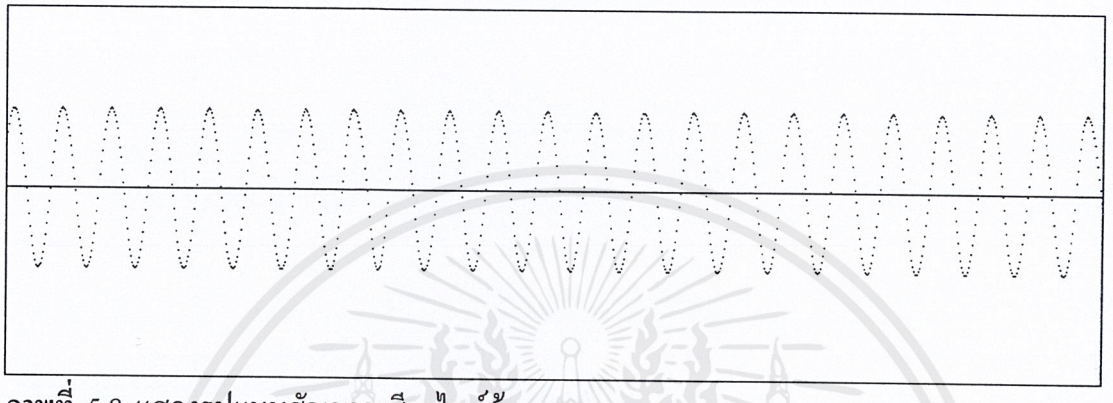
ภาพที่ 5.5 แสดงรูปแบบสัญญาณเสียงดนตรีต้นแบบ

ภาพที่ 5.6 แสดงรูปแบบสัญญาณเสียงดนตรีที่ผ่านการเข้ารหัสลับ

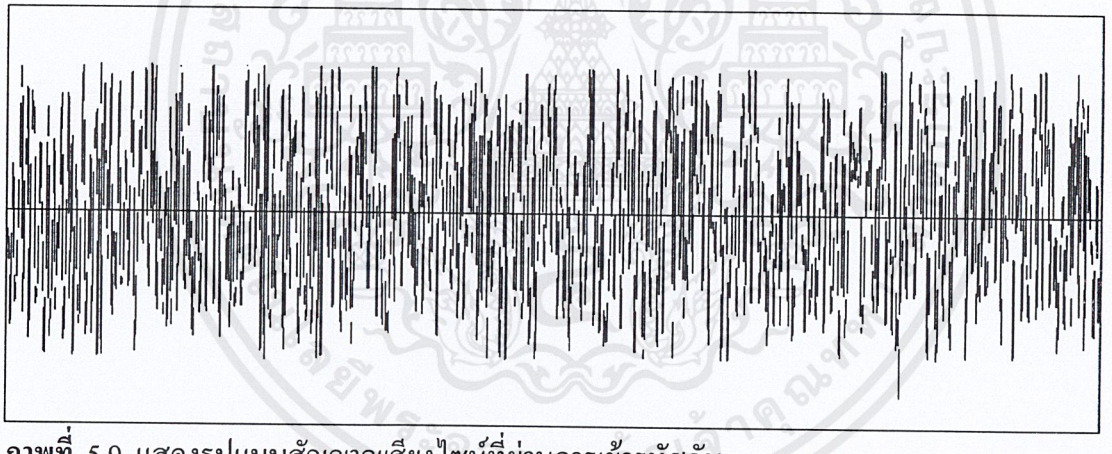


ภาพที่ 5.7 แสดงรูปแบบสัญญาณเสียงดนตรีที่ผ่านการถอดรหัสลับ

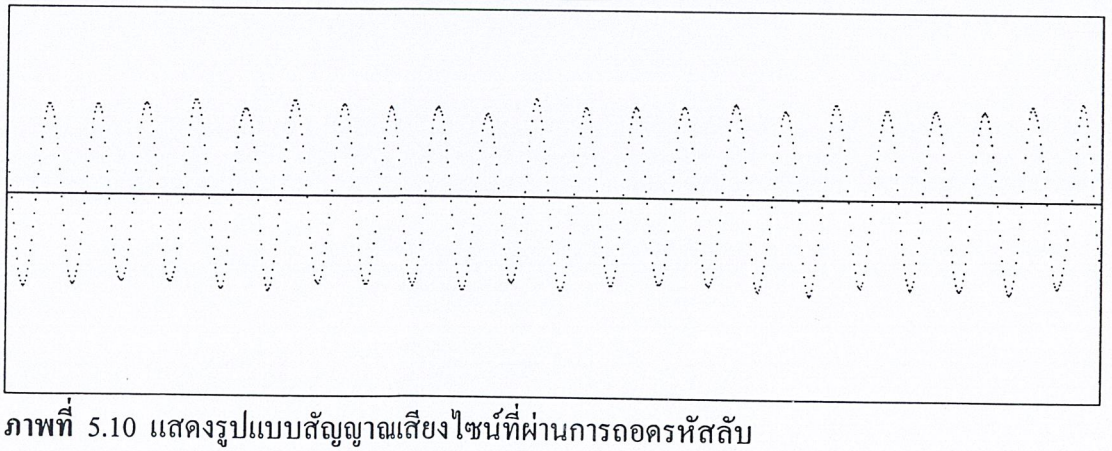
การทดลองในรูปแบบแรกใช้สัญญาณไซน์ โดยมีรูปแบบสัญญาณต้นแบบในภาพที่ 5.8 สำหรับนำมาทำการเข้ารหัสลับ โดยนำสัญญาณเข้าทาง Input Port ของ DSK แล้วทำการเข้ารหัสลับนำผลที่ได้จาก Output Port ดังแสดงในภาพที่ 5.9 และ ภาพที่ 5.10 เป็นการแสดงรูปของสัญญาณไซน์ที่ผ่านการถอดรหัสแล้ว



ภาพที่ 5.8 แสดงรูปแบบสัญญาณเสียงไซน์ต้นแบบ



ภาพที่ 5.9 แสดงรูปแบบสัญญาณเสียงไซน์ที่ผ่านการเข้ารหัสลับ



ภาพที่ 5.10 แสดงรูปแบบสัญญาณเสียงไซน์ที่ผ่านการถอดรหัสลับ

5.3 สรุปผลการทดลอง

เนื่องจากระบบของ DSK เป็นวงจรที่มีความเร็วสูงการนำมาใช้ที่เวลาจริงซึ่งมีอัตราสุ่มประมาณ 10,000 ครั้งต่อวินาที จากการทดลองโดยนำเสียงพูด , เสียงดนตรี และ สัญญาณไซน์ มาทำการเข้ารหัส ผลปรากฏว่าไม่ว่าจะเป็นเสียงพูด , เสียงดนตรี และ สัญญาณไซน์ ไม่สามารถที่จะแปลความหมายได้ และเมื่อทำการถอดรหัสกลับก็จะได้สัญญาณที่มีลักษณะที่ใกล้เคียงกับสัญญาณต้นแบบกลับคืนมา แต่สัญญาณจะมีความแตกต่างกันบ้างเล็กน้อยอันเป็นผลเนื่องมาจาก ขั้นตอนการแปลงสัญญาณอนาล็อกไปเป็นสัญญาณดิจิตอลและขั้นตอนการแปลงสัญญาณดิจิตอลไปเป็นสัญญาณอนาล็อก ดังผลการทดลองในภาพที่ 5.2 - 5.10

บทที่ 6
สรุปผลการวิจัยและข้อเสนอแนะ

6.1 สรุปผลการทดลอง

จากการทดลองที่ทำมาเป็นการนำเสนอรูปแบบทางอัลกอลิทึมและการนำมาใช้ร่วมกับ DSK ที่มีขนาดเล็กและมีความเร็วสูง โดยอาศัยหลักการของบล็อคโค้ดเชิงเส้นในการสร้างความปลอดภัยให้กับข้อมูลข่าวสาร ด้วยการนำรูปแบบผิดพลาดที่ถูกกำหนดรูปแบบไว้แล้ว มาทำการบวกแบบเอ็กครูซีฟออร์เข้ากับรหัสคำในส่วนที่เป็นข้อมูลแล้วนำรหัสคำที่ได้มาทำการสลับบิตในส่วนของข้อมูล โดยมีรหัสแก้ไขเป็นตัวเลือกรูปแบบการสลับทางด้านการถอดรหัส ใช้วิธีการคำนวณหาค่าซีนโดรมมาทำการถอดรหัสกลับคืนมาโดยวิธีการทั้งหมดได้กล่าวรายละเอียดในบทที่ 4 ซึ่งมีข้อจำกัดหลายประการที่จะให้ความถูกต้องและเที่ยงตรงของการเข้ารหัสและการถอดรหัสมากที่สุด อย่างไรก็ตามการเข้ารหัสบล็อกโค้ดเชิงเส้นที่เลือกใช้ในการทดลองนี้จะทำให้ปริมาณข่าวสารของข้อมูลเพิ่มขึ้นจากเดิมถึง 1.5 เท่าตัว(เนื่องจากข่าวสารขนาด 5 บิตถูกเข้าเป็นรหัสคำขนาด 8 บิต) ซึ่งถือว่าเป็นการลดประสิทธิภาพของการสื่อสารดังนั้นในอนาคตอาจต้องมีการพัฒนาทางด้านการเข้ารหัสเพื่อลดปริมาณข่าวสารลงให้ใกล้เคียงกับระบบที่ไม่ผ่านการเข้ารหัส

การเข้ารหัสเพียง 5 บิตนั้นจะทำให้รายละเอียดของสัญญาณขาดหายไป ทำให้เอาท์พุตที่ได้แตกต่างจากสัญญาณจริงแต่ก็ใกล้เคียง

จากการทดลองและศึกษาทางด้าน DSK จึงได้สร้างแผงวงจรในการเพิ่มหน่วยความจำให้มากขึ้นแล้วยังทำให้ DSK ใช้ประโยชน์ในการเขียนโปรแกรมในการใช้งาน และลดค่าใช้จ่ายในการซื้อ EVM BOARD ที่มีราคาแพงมาก ซึ่งจะเป็นแนวทางหนึ่งสำหรับผู้สนใจในการพัฒนาอุปกรณ์ประเภทนี้ขึ้นมาใช้งานโดยอาศัยเทคโนโลยีที่มีในประเทศ

6.2 ปัญหาและอุปสรรคที่พบ

1. เนื่องจากทำการเข้ารหัสแบบ 5 บิตทำให้ได้ข้อมูลที่ไม่ละเอียดดังนั้นสัญญาณที่ผ่านการถอดรหัสมาจะแตกต่างจากสัญญาณจริง

2. ส่วนของอุปกรณ์รับส่งข้อมูลดิจิตอลที่ไม่มีบน DSK

3. ส่วนของการควบคุม ADC และ DAC บน DSK ทำได้ยากเนื่องจากต้องใช้โปรมแกรมในการควบคุม ส่วนของอุปกรณ์รับส่งข้อมูลดิจิตอลที่ไม่มีบน DSK

4. ส่วนของการออกแบบการเข้ารหัสเสียงจะต้องอาศัยความเร็วในการประมวลผลและการรับส่งข้อมูลที่เข้ารหัสแล้วเร็วพอที่จะทำการถอดรหัสและแปลงออกมาเป็นสัญญาณเสียงที่ถูกต้อง

6.3 วิธีแก้ไข

1. เปลี่ยนวิธีการเข้ารหัสเป็นวิธีอื่น ๆ ที่มีประสิทธิภาพมากกว่านี้

2. สร้างวงจรในส่วนของการรับส่งข้อมูลโดยใช้ IC 16550 ซึ่งเป็น IC ที่ใช้ในการรับส่งข้อมูลดิจิตอลขนาด 8 บิต

3. ลดขนาดของโปรแกรมลงเพื่อไม่ให้เสียเวลาในการทำงานในส่วนของการประมวลผลของข้อมูลการเข้ารหัสและถอดรหัส

6.4 ข้อเสนอแนะและแนวทางการพัฒนา

จากการทดลองนี้จะเป็นการเข้ารหัสสัญญาณแบบบล็อกโค้ดเชิงเส้นซึ่งเป็นพื้นฐานของการเข้ารหัส ผู้ที่สนใจอาจทำการเข้ารหัสสัญญาณแบบต่าง ๆ ที่มีประสิทธิภาพในเรื่องของการ เข้ารหัสและถอดรหัสเพื่อให้ได้ข้อมูลที่มีประสิทธิภาพและทำให้เสียงที่ได้ชัดเจนมากขึ้น และส่งในระยะทางไกล ๆ ได้ และการทดลองนี้ได้ทำการสร้าง พล็อต RS-232 ไว้เพื่อเป็นการติดต่อสื่อสารและรองรับการเชื่อมต่อกับอุปกรณ์คอมพิวเตอร์หรือโมเด็ม ดังนั้นถ้าหากผู้สนใจที่จะพัฒนาต่อไปให้สามารถสื่อสารได้ไกลขึ้นเช่น ส่งผ่านข้ามเครือข่ายโทรศัพท์ ฯลฯ ก็สามารถทำได้

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."Acadamic 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,1998
7. ฟูศักด์ ชีวสุวิทย์. "การแก้รหัสผิด."คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง พ.ศ.2528
8. ฟูศักด์ ชีวสุวิทย์. และคเชนทร์ แจ่มกมล. "การเข้ารหัสโดยใช้ซีนโครมองลิเนียร์บล็อคโค้ดและการสลับบิตโดยอาศัยเพอซิโดแรนดอม."วารสารทางวิชาการของสมาคมคอมพิวเตอร์แห่งประเทศไทย
9. ฟูศักด์ ชีวสุวิทย์. และคเชนทร์ แจ่มกมล. "การออกแบบตัวเข้ารหัสลับแบบเวลาจริงโดยใช้ซีนโครมของลิเนียร์บล็อคโค้ดและการสลับบิตโดยอาศัยการกำเนิดแบบสุ่ม." การประชุมวิชาการทางไฟฟ้าครั้งที่ 18 มหาวิทยาลัยเทคโนโลยีมหานคร,หน้า 1034-1038, พฤศจิกายน 2538
10. K.Sam Shanmugam. "Digital and analog communication system." John Wiley & Sons,Inc,1978
11. Vera Pless. "Introduction to the theory of error-correcting codes." John Wiley & Sons,Inc,1978
12. Rulph Chassaing. "Digital Signal Processing Laboratory Experimants Using C and the TMS320C31 DSK" John Wiley & Sons,Inc,1999

ภาคผนวก

วงจร (Circuit)