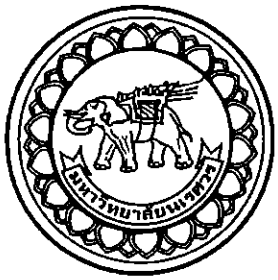
อภินันทนาการ



สำนักหอสมุด

# การศึกษาประสิทธิภาพของรหัสคอนโวลูชัน

# THE STUDY OF CONVOLUTIONAL CODE EFFICIENCY

นายอภิสิทธิ์ ครุฑศรี รหัส 55361052

สำนักหอสมุด มหาวิทยาลัยนเรศวร
วันลงทะเบียน 24 มิ.ย. 2561
เลขทะเบียน 1722007X
เลขเรียกหนังสือ ปร
ว1 CD
0216 ก
2559
CD-STL 66

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2559

# ใบรับรองปริญญานิพนธ์

**ชื่อหัวข้อโครงงาน** การศึกษาประสิทธิภาพของรหัสคอนโวลูชัน
**ผู้ดำเนินโครงงาน** นายอภิสิทธิ์ ครุฑศรี รหัส 55361052
**ที่ปรึกษาโครงงาน** ผู้ช่วยศาสตราจารย์ ดร. สุรเชษฐ์ กานต์ประชา
**สาขาวิชา** วิศวกรรมไฟฟ้า
**ภาควิชา** วิศวกรรมไฟฟ้าและคอมพิวเตอร์
**ปีการศึกษา** 2559

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....ที่ปรึกษาโครงงาน
(ผู้ช่วยศาสตราจารย์ ดร. สุรเชษฐ์ กานต์ประชา)

.....กรรมการ
(ผู้ช่วยศาสตราจารย์ ดร. อัครพันธ์ วงศ์กังแห)

.....กรรมการ
(ดร.ชัยรัตน์ พินทอง)

| | |
|---|---|
| **ชื่อหัวข้อโครงงาน** | การศึกษาประสิทธิภาพของรหัสคอนโวลูชัน |
| **ผู้ดำเนินโครงงาน** | นายอภิสิทธิ์ ครุฑศรี รหัส 55361052 |
| **ที่ปรึกษาโครงงาน** | ผู้ช่วยศาสตราจารย์ ดร. สุรเชษฐ์ กานต์ประชา |
| **สาขาวิชา** | วิศวกรรมไฟฟ้า |
| **ภาควิชา** | วิศวกรรมไฟฟ้าและคอมพิวเตอร์ |
| **ปีการศึกษา** | 2559 |

.....

## บทคัดย่อ


โครงงานนี้เป็นการศึกษาประสิทธิภาพของรหัสคอนโวลูชันซึ่งเป็นวิธีการหนึ่งเพื่อเพิ่มความสามารถในการแก้ไขความผิดพลาดในกระบวนการส่งข้อมูลไปยังภาครับบนช่องส่งสัญญาณและถูกสร้างขึ้นโดยตัวเข้ารหัสที่ภาคส่ง สัญญาณรบกวนบนช่องส่งสัญญาณถูกจำลองขึ้นด้วยการสุ่มบิตที่ผิดพลาดจากข้อมูลที่ทำการส่ง โดยรูปแบบที่ถูกสร้างในการสุ่มเพื่อใช้ทดสอบบิตของคำรหัสคอนโวลูชันที่ถูกเลือกแบบสุ่มนี้ด้วยค่าความน่าจะเป็นซึ่งเป็นตัวแปรที่ถูกกำหนดค่าใช้กับช่องส่งสัญญานเพื่อทดสอบความผิดพลาดด้วยบิตที่ถูกเลือกแบบสุ่มจากชุดบิตข้อมูลที่ทำการส่งออกไป โดยใช้ค่าอัตราบิตผิดพลาดที่ได้รับเป็นตัวชี้วัดประสิทธิภาพของกระบวนการในการตรวจสอบ และแก้ไขความผิดพลาดที่เกิดขึ้น

โดยที่การเข้ารหัสคอนโวลูชัน การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีโดยการตัดสินใจแบบฮาร์ดนั้นเป็นโปรแกรมจำลองที่สร้างขึ้นด้วยแมตแล็บ ทำให้พบว่าอัตรารหัสมีความสัมพันธ์แบบแปรผกผันกับประสิทธิภาพของความสามารถในการแก้ไขความผิดพลาด

| | |
|---|---|
| **Project title** | The Study of Convolutional Code Efficiency |
| **Name** | Mr. Apisit Krutsri ID. 55361052 |
| **Project advisor** | Asst. Prof. Surachet Kanprachar, Ph.D. |
| **Major** | Electrical Engineering |
| **Department** | Electrical and Computer Engineering |
| **Academic year** | 2016 |


## Abstract

This project is a study of Convolutional code efficiency. Convolutional code is a method for adding error-correcting ability to the processing of information transmitted to a receiver on a transmission channel, and is generated by an encoder on the transmitter. Channel noise is simulated which may randomly cause bit errors in the transmitted data. A randomly generated bit pattern is used to test randomly selected bits in the convolutional code. A probability value, a given input variable, is applied to the channel to test for errors in randomly selected bits in the transmission bit stream. A bit error rate is obtained which indicates the efficiency of the error detection and correction method.

To encode the convolutional code, the hard-decision Viterbi decoding algorithm was applied in the simulation program created with MATLAB. It was seen that the code rate was varied inversely with the error correcting ability efficiency.

# กิตติกรรมประกาศ

โครงงานฉบับนี้สามารถสำเร็จลุล่วงได้ด้วยดี จากความอนุเคราะห์ของบุคคลหลายท่าน ซึ่งผู้ดำเนินโครงงานจึงขอกราบขอบพระคุณบุคคลต่างๆ ดังนี้

ขอขอบพระคุณ บิดา มารดา และญาติพี่น้องที่คอยให้กำลังใจ และการสนับสนุนเสมอมา

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. สุรเชษฐ์ กานต์ประชา อาจารย์ที่ปรึกษาโครงงานที่คอยให้การอบรมสั่งสอน คำแนะนำ และคำปรึกษาทั้งในเรื่องการดำเนินโครงงาน เรื่องเรียน และเรื่องอื่นๆ รวมทั้งโอกาสต่างๆ ที่ทางข้าพเจ้าได้รับเป็นอย่างดี

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ ดร. อัครพันธ์ วงศ์กังแห และดร.ชัยรัตน์ พินทอง กรรมการในการสอบโครงงานฉบับนี้ที่ได้ให้ความอนุเคราะห์แก่ข้าพเจ้า

ขอขอบพระคุณบุคลากรทุกท่าน ภาควิชาวิศวกรรมไฟฟ้า และคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวรที่ได้อบรมให้ความรู้ ความช่วยเหลือ และคำแนะนำต่างๆ ต่อข้าพเจ้าเป็นอย่างดีเสมอมา

สุดท้ายนี้ขอขอบคุณรุ่นพี่ เพื่อนทุกคน และบุคคลอื่นๆ ที่ไม่ได้ถูกกล่าวถึงที่ได้ให้ความอนุเคราะห์ต่อข้าพเจ้าในการดำเนินโครงงานจนสำเร็จ

นายอภิสิทธิ์ ครุฑศรี

# สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์ ..... ก
บทคัดย่อภาษาไทย ..... ข
บทคัดย่อภาษาอังกฤษ ..... ค
กิตติกรรมประกาศ ..... ง
สารบัญ ..... จ
สารบัญตาราง ..... ช
สารบัญรูป ..... ฌ

บทที่ 1 บทนำ ..... 1
1.1 ที่มาและความสำคัญโครงงาน ..... 1
1.2 วัตถุประสงค์ของโครงงาน ..... 2
1.3 ขอบเขตของโครงงาน ..... 2
1.4 ขั้นตอนและแผนการดำเนินงาน ..... 2
1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน ..... 3
1.6 งบประมาณ ..... 3

บทที่ 2 หลักการและทฤษฎีที่เกี่ยวข้อง ..... 4
2.1 โครงสร้างของรหัสคอนโวลูชัน และการเข้ารหัสคอนโวลูชัน [1,2] ..... 4
2.1.1 แผนภาพสถานะ (State diagram) [1,4] ..... 11
2.1.2 แผนภาพต้นไม้ (Tree diagram) [1,4] ..... 12
2.1.3 แผนภาพเทรลลิส (Trellis diagram) [1,4] ..... 13
2.2 ระยะฟรี (Free distance หรือ $d_{free}$) [1,5,6] ..... 14
2.3 การถอดรหัสคอนโวลูชัน (Convolutional decoding) [4,6] ..... 18
2.3.1 การถอดรหัสด้วยขั้นตอนวิธีของวิเทอร์บีด้วยการตัดสินใจแบบฮาร์ด ..... 19
2.3.2 การถอดรหัสด้วยขั้นตอนวิธีของวิเทอร์บีด้วยการตัดสินใจแบบซอฟต์ ..... 19

# สารบัญ (ต่อ)

หน้า


บทที่ 3 การออกแบบโครงงานและวิธีการดำเนินงาน ..... 24

3.1 ศึกษาการสร้างบิตข้อมูล ..... 24

3.2 ศึกษาการเข้ารหัสคอนโวลูชัน ..... 24

3.3 ศึกษาการสุ่มตำแหน่งบิตที่ผิดพลาด ..... 24

3.4 ศึกษาการรวมข้อมูลคำรหัสคอนโวลูชัน กับตำแหน่งบิตที่ผิดพลาด ..... 24

3.5 ศึกษาการถอดคำรหัสคอนโวลูชันโดยขั้นตอนวิธีของวิเทอร์บี ..... 25

3.6 การจำลองการทำรหัสคอนโวลูชัน ..... 25

3.6.1 กรณีศึกษาของระบบ [3,1,2] ..... 25

3.6.2 กรณีศึกษาของระบบ [4,1,2] ..... 25

3.6.3 กรณีศึกษาของระบบ [5,1,2] ..... 26

3.6.4 กรณีของระบบ [3,2,1] ..... 26

3.7 การสร้างโปรแกรมจำลองด้วยโปรแกรมแมตแล็บ ..... 26

3.7.1 การสร้างบิตข้อมูล ..... 26

3.7.2 การเข้ารหัสคอนโวลูชัน ..... 26

3.7.3 การสุ่มตำแหน่งบิตที่ผิดพลาด และการรวมข้อมูล ..... 26

3.7.4 การถอดคำรหัสคอนโวลูชัน ..... 27

3.7.5 จำนวนบิตที่ผิดพลาดที่เกิดขึ้นภายในช่องส่งสัญญาณ และภายหลังการถอดคำรหัส ..... 27

3.8 วิเคราะห์ประสิทธิภาพของระบบด้วยอัตราบิตที่ผิดพลาด (BER) ..... 27

บทที่ 4 ผลการดำเนินโครงงาน ..... 28

4.1 ผลการศึกษาการเข้ารหัสคอนโวลูชัน และการถอดคำรหัสคอนโวลูชัน ..... 28

4.2 ผลการศึกษาการสุ่มตำแหน่งบิตที่ผิดพลาด และการรวมข้อมูลในช่องส่งสัญญาณ ..... 28

4.3 ผลการจำลองการทำรหัสคอนโวลูชัน ..... 28

4.3.1 กรณีศึกษาของระบบ [3,1,2] ..... 29

4.3.2 กรณีศึกษาของระบบ [4,1,2] ..... 51

4.3.3 กรณีศึกษาของระบบ [5,1,2] ..... 56

# สารบัญ (ต่อ)

หน้า

4.3.4 กรณีศึกษาของระบบ [3,2,1] ..... 60
4.4 ประสิทธิภาพของรหัสคอนโวลูชัน ..... 65

บทที่ 5 สรุปผลการดำเนินงาน ..... 68
5.1 สรุปผลการดำเนินงาน ..... 68
5.1.1 สรุปผลการจำลองการทำรหัสคอนโวลูชันของกรณีศึกษาต่างๆ ..... 68
5.1.2 สรุปผลประสิทธิภาพของรหัสคอนโวลูชัน ..... 69
5.2 ปัญหาที่พบขณะดำเนินโครงงาน ..... 69
5.3 ข้อเสนอแนะ ..... 69

เอกสารอ้างอิง ..... 70

ภาคผนวก การสุ่มตำแหน่งบิตที่ผิดพลาด ..... 71

ประวัติผู้ดำเนินโครงงาน ..... 73

# สารบัญตาราง

ตารางที่ หน้า

2.1 ตารางแสดงค่าสถานะก่อน และหลังข้อมูลเข้าสู่วงจร และค่าคำรหัสที่ได้.....10
2.2 ตารางแสดงค่าต่างๆ สำหรับอัตรารหัส 1/2.....15
2.3 ตารางแสดงค่าต่างๆ สำหรับอัตรารหัส 1/3.....16
2.4 ตารางแสดงค่าต่างๆ สำหรับอัตรารหัส 1/4.....17
4.1 ตารางแสดงค่า อินพุต สถานะเดิม คำรหัส และสถานะถัดไปในกรณีที่อินพุตคือ 0.....31
4.2 ตารางแสดงค่า อินพุต สถานะเดิม คำรหัส และสถานะถัดไปในกรณีที่อินพุตคือ 1.....31
4.3 ตารางสรุปความสัมพันธ์ของระบบ [3,1,2].....32
4.4 ตารางแสดง อินพุต การเปลี่ยนสถานะ และคำรหัสจากตัวอย่างอินพุต.....33
4.5 ตารางแสดงความสัมพันธ์ของระบบ [4,1,2].....52
4.6 ตารางแสดง อินพุต การเปลี่ยนสถานะ และคำรหัสจากตัวอย่างอินพุต.....54
4.7 ตารางแสดงความสัมพันธ์ของระบบ [5,1,2].....57
4.8 ตารางแสดง อินพุต การเปลี่ยนสถานะ และคำรหัสจากตัวอย่างอินพุต.....58
4.9 ตารางแสดงความสัมพันธ์ของระบบ [3,2,1].....61
4.10 ตารางแสดง อินพุต การเปลี่ยนสถานะ และคำรหัสจากตัวอย่างอินพุต.....63

# สารบัญรูป

รูปที่ หน้า

2.1 รูปแสดงวงจรเข้ารหัสคอนโวลูชัน....4
2.2 รูปแสดงวงจรเข้ารหัสคอนโวลูชันแบบอนุกรม....5
2.3 รูปแสดงวงจรเข้ารหัสคอนโวลูชันแบบขนาน....5
2.4 รูปแสดงตัวอย่างแผนภาพแบบบล็อกของวงจรเข้ารหัส....8
2.5 รูปแสดงแผนภาพสถานะของตัวอย่างที่ 2.1....12
2.6 รูปแสดงแผนภาพต้นไม้ของตัวอย่างที่ 2.1....13
2.7 รูปแสดงแผนภาพเทรลลิสของตัวอย่างที่ 2.1....14
2.8 รูปแสดงแผนภาพเทรลลิส....20
2.9 รูปแสดงการหาระยะแฮมมิงที่แต่ละเส้นทางของแต่ละระดับ....21
2.10 รูปแสดงการหาผลรวมของแต่ละเส้นทางที่เป็น "Survivor"....22
2.11 รูปแสดงการหาผลรวมที่น้อยที่สุดที่ได้....22
4.1 แผนภาพจำลองระบบ [3,1,2]....29
4.2 แผนภาพสถานะของระบบ [3,1,2]....32
4.3 แผนภาพเทรลลิสของระบบ [3,1,2]....34
4.4 การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 0....34
4.5 การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 1....35
4.6 การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 2....36
4.7 การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 3....37
4.8 การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 4....38
4.9 การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 5....39
4.10 การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 6....40
4.11 การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 7....41
4.12 การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 8....42
4.13 การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 9....43
4.14 การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 10....44
4.15 การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 11....45
4.16 แผนภาพเทรลลิสการถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บี
ของระบบ [3,1,2]....45

# สารบัญรูป (ต่อ)

รูปที่ หน้า

4.17 แผนภาพเทรลลิสแสดงการถอดคำรหัสในกรณีที่ 2.2 ..... 46
4.18 แผนภาพเทรลลิสแสดงการถอดคำรหัสในกรณีที่ 2.3 ..... 47
4.19 แผนภาพเทรลลิสแสดงการถอดคำรหัสในกรณีที่ 2.4 ..... 47
4.20 แผนภาพเทรลลิสแสดงการถอดคำรหัสในกรณีที่ 2.5 ..... 48
4.21 แผนภาพเทรลลิสแสดงการถอดคำรหัสในกรณีที่ 2.6 ..... 49
4.22 แผนภาพเทรลลิสแสดงการถอดคำรหัสในกรณีที่ 2.7 ..... 49
4.23 แผนภาพเทรลลิสแสดงการถอดคำรหัสในกรณีที่ 2.8 ..... 50
4.24 แผนภาพเทรลลิสแสดงการถอดคำรหัสในกรณีที่ 2.9 ..... 51
4.25 แผนภาพจำลองระบบ [4,1,2] ..... 52
4.26 แผนภาพสถานะของระบบ [4,1,2] ..... 53
4.27 แผนภาพเทรลลิสของระบบ [4,1,2] ..... 55
4.28 แผนภาพเทรลลิสแสดงการถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บี ของระบบ [4,1,2] ..... 55
4.29 แผนภาพจำลองระบบ [5,1,2] ..... 56
4.30 แผนภาพสถานะของระบบ [5,1,2] ..... 58
4.31 แผนภาพเทรลลิสของระบบ [5,1,2] ..... 59
4.32 แผนภาพเทรลลิสแสดงการถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บี ของระบบ [5,1,2] ..... 60
4.33 แผนภาพจำลองระบบ [3,2,1] ..... 61
4.34 แผนภาพสถานะของระบบ [3,2,1] ..... 63
4.35 แผนภาพเทรลลิสของระบบ [3,2,1] ..... 64
4.36 แผนภาพเทรลลิสแสดงการถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บี ของระบบ [3,2,1] ..... 65
4.37 แผนภูมิแสดงผลของอัตราบิตที่ผิดพลาดจากกรณีศึกษา ..... 66
1 แผนผังงานการสุ่มตำแหน่งบิตที่ผิดพลาด ..... 72

# บทที่ 1
# บทนำ

## 1.1 ที่มาและความสำคัญโครงงาน

ในปัจจุบันมนุษย์ใช้ประโยชน์จากการสื่อสารในด้านต่างๆมากมาย เช่น ด้านการศึกษา ด้านความบันเทิง ด้านการประกอบธุรกิจ การติดต่อสื่อสารระหว่างบุคคล ด้านการแพทย์ และการรับข้อมูลข่าวสารทั่วไป เป็นต้น ดังนั้นการสื่อสารจึงมีความสำคัญต่อการดำเนินชีวิตประจำวันของมนุษย์เป็นอย่างมาก จึงมีการพัฒนาเทคโนโลยีทางการสื่อสารเพื่อให้มีประสิทธิภาพในการใช้งานสูงที่สุด

การสื่อสารข้อมูลที่มีมากมายหลายรูปแบบที่ถูกใช้กันอย่างแพร่หลายในปัจจุบัน หนึ่งในนั้นคือ การสื่อสารดิจิตอล ซึ่งได้รับความนิยมเป็นอย่างมาก โดยทั่วไปแล้วในการสื่อสารข้อมูลนั้นความสำคัญลำดับต้นๆของระบบการสื่อสารข้อมูลนั้นจะให้ความสำคัญในเรื่องของความถูกต้องของข้อมูล แต่ในการส่งสัญญาณสื่อสารนั้นก็มักพบปัญหาในเรื่องของความผิดพลาดในการสื่อสารข้อมูล อันเนื่องจากสัญญาณรบกวนต่างๆ ระหว่างการรับ-ส่งข้อมูลซึ่งมีผลทำให้เกิดความผิดเพี้ยนไปของข้อมูล ซึ่งอาจจะทำให้การตีความข้อมูลที่ภาครับได้รับนั้นไม่ถูกต้อง จึงมีการนำเทคนิคต่างๆมาใช้เพื่อเพิ่มประสิทธิภาพในการสื่อสารข้อมูล เช่นการเพิ่มความสามารถในการตรวจพบความผิดพลาดของข้อมูล และความสามารถในการแก้ไขความผิดพลาดนั้นได้

ดังที่กล่าวมาข้างต้นนั้น และเทคนิคที่ใช้ในการเพิ่มความสามารถต่างๆนี้เองก็มีประสิทธิภาพที่แตกต่างกัน ดังนั้นโครงงานจึงมีความต้องการศึกษาประสิทธิภาพของหนึ่งในวิธีการเพิ่มความสามารถในการตรวจพบความผิดพลาดของข้อมูล และความสามารถในการแก้ไขความผิดพลาดของข้อมูลได้ในระหว่างการรับ-ส่งข้อมูลในระบบสื่อสารให้มีความถูกต้อง โดยกระบวนการเข้ารหัสคอนโวลูชัน (Convolutional Codes) ซึ่งมีวิธีการทำงานคือการนำข้อมูลดิจิตอลที่ต้องการจะส่งผ่านระบบสื่อสารมาทำการแบ่งข้อมูลออกเป็นชุดข้อมูล แล้วนำมาเข้ารหัสให้อยู่ในรูปของคำรหัส (Codeword) ของการเข้ารหัสคอนโวลูชัน (Convolutional Codes)

โครงงานนี้ได้นำเสนอการศึกษาประสิทธิภาพของกระบวนการเข้ารหัสคอนโวลูชัน (Convolutional Codes) โดยโปรแกรมแมตแล็บ (MATLAB) ถูกนำมาใช้ในการสร้างการจำลองการเข้ารหัสคอนโวลูชัน (Convolutional Codes)

## 1.2 วัตถุประสงค์ของโครงงาน

1) เพื่อศึกษาการเข้ารหัสและการถอดรหัสของรหัสคอนโวลูชัน

2) เพื่อศึกษาประสิทธิภาพของรหัสคอนโวลูชัน

## 1.3 ขอบเขตของโครงงาน

1) ศึกษากระบวนการทำรหัสคอนโวลูชัน

2) ศึกษาการเขียนโปรแกรมแมตแล็บ (MATLAB)

## 1.4 ขั้นตอนและแผนการดำเนินงาน

| รายละเอียด | พ.ศ.2558 | | | | | พ.ศ.2559 | | | |
|---|---|---|---|---|---|---|---|---|---|
| | ส.ค. | ก.ย. | ต.ค. | พ.ย. | ธ.ค. | ม.ค. | ก.พ. | มี.ค. | เม.ย. |
| 1) รวบรวมข้อมูล | | | | | | | | | |
| 2) ศึกษากระบวนการทำรหัสคอนโวลูชัน | | | | | | | | | |
| 3) สร้างแบบจำลองกระบวนการทำรหัสคอนโวลูชันด้วยโปรแกรมแมตแล็บ | | | | | | | | | |
| 4) วิเคราะห์และสรุปผลการศึกษาประสิทธิภาพของรหัสคอนโวลูชัน | | | | | | | | | |
| 5) จัดทำปริญญานิพนธ์ | | | | | | | | | |

## 1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน

1) เข้าใจกระบวนการทำรหัสคอนโวลูชัน

2) เพิ่มทักษะการเขียนโปรแกรมแมตแล็บ (MATLAB)

3) ทราบถึงตัวแปรที่มีความสำคัญต่อประสิทธิภาพของการเข้ารหัสแบบรหัสคอนโวลูชัน

## 1.6 งบประมาณ

| | | |
|---|---|---|
| 1) ค่าเอกสารคู่มือที่ใช้ในการศึกษา | 300 | บาท |
| 2) ค่าถ่ายเอกสารและเข้าเล่มปริญญานิพนธ์ | 700 | บาท |
| รวมเงินทั้งสิ้น(หนึ่งพันบาทถ้วน) | 1,000 | บาท |

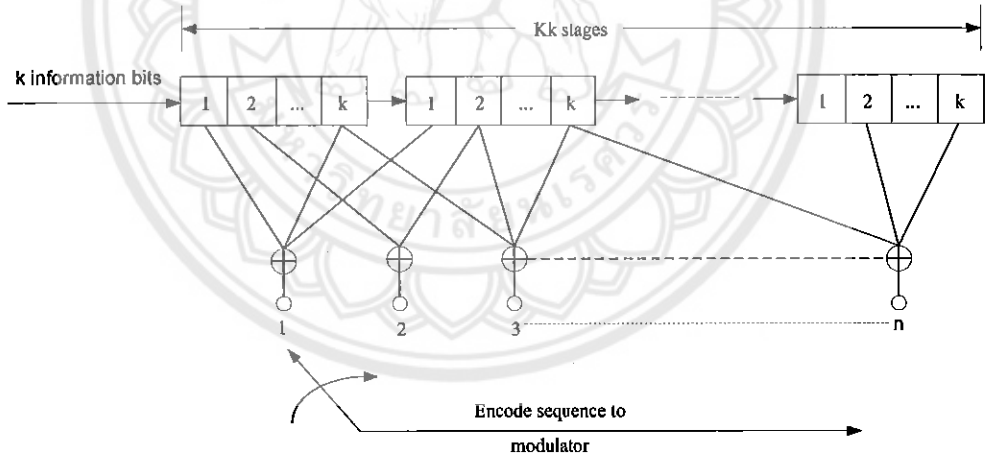
หมายเหตุ: ถัวเฉลี่ยทุกรายการ

# บทที่ 2
# หลักการและทฤษฎีที่เกี่ยวข้อง

การเข้ารหัสข้อมูลแบบรหัสคอนโวลูชันนั้นเป็นการเข้ารหัสข้อมูลดิจิตอลแบบไบนารีที่มีหน่วยความจำ กล่าวคือข้อมูลบิตก่อนหน้าข้อมูล ณ เวลาใดๆ จำนวนหนึ่งนั้นจะถูกนำเข้าไปประมวลผลร่วมกับบิตข้อมูลที่เข้ามายังวงจรเข้ารหัส ณ เวลานั้นๆ เพื่อหาค่ารหัสของรหัสคอนโวลูชัน

ที่โดยในบทนี้จะกล่าวถึงทฤษฎีต่างๆ ที่เกี่ยวข้องกับรหัสคอนโวลูชันดังนี้ โครงสร้างของวงจรเข้ารหัสคอนโวลูชัน รวมถึงหลักการที่ใช้วิเคราะห์ระบบของวงจรเข้ารหัส เช่น แผนภาพสถานะ (State diagram) แผนภาพต้นไม้ (Tree diagram) และแผนภาพเทรลลิส (Trellis diagram) เป็นต้น ระยะฟรีซึ่งเป็นค่าที่บ่งบอกถึงประสิทธิภาพของรหัสคอนโวลูชันที่ได้จากวงจรเข้ารหัส และการถอดรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีโดยการตัดสินใจแบบฮาร์ด

## 2.1 โครงสร้างของรหัสคอนโวลูชัน และการเข้ารหัสคอนโวลูชัน [1,2]




**รูปที่ 2.1** รูปแสดงวงจรเข้ารหัสคอนโวลูชัน[1]

รูปที่ 2.1 ข้อมูลไบนารีจำนวน k บิต จะถูกส่งผ่านวงจรเลื่อนข้อมูลจำกัดสถานะเชิงเส้นที่มีจำนวนชุดของวงจรเลื่อนข้อมูลจำนวน K ชุด แต่ละชุดประกอบไปด้วยวงจรเลื่อนข้อมูลจำนวน k ตัว ผ่านมายังตัวกำเนิดที่เป็นฟังก์ชันพีชคณิตเชิงเส้นที่มีจำนวน n ตัวที่จะถูกนำค่าที่ได้ในแต่ละตัวกำเนิดมาเรียงต่อกันตามลำดับ ซึ่งจะเห็นได้ว่าเมื่อมีข้อมูลเข้าจำนวน k บิต จะได้คำรหัสออกจำนวน n บิต ซึ่งค่า K นี้เองถูกเรียกว่า "Constraint length of the convolutional code" หรือความยาวจำกัดของรหัสคอนโวลูชันนั่นเอง

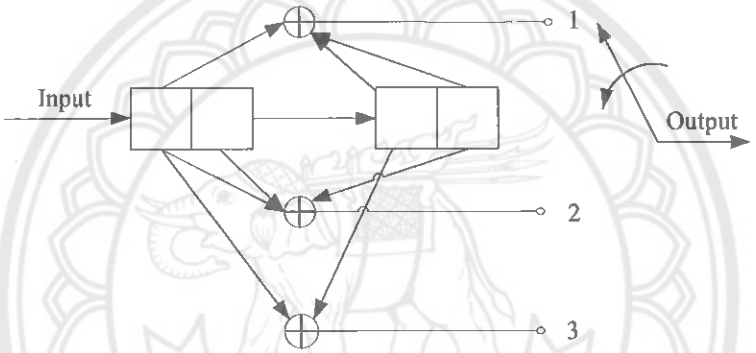
โดยในโครงงานนี้จะนำเสนอรหัสคอนโวลูชันที่วงจรของตัวเข้ารหัสเป็นแบบขนานและระบบของตัวเข้ารหัสถูกนิยามด้วย (n,k,m) โดยกำหนดความหมายของตัวแปรต่างๆ ดังต่อไปนี้

n คือ จำนวนบิตของ Codeword หรือคำรหัสต่อจำนวนบิตข้อมูลขาเข้า k

k คือ จำนวนบิตของข้อมูลขาเข้าในแต่ละครั้ง

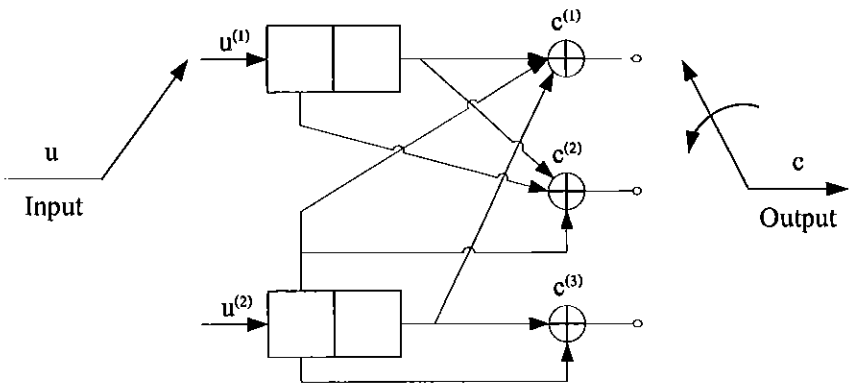
m คือ จำนวนวงจรเลื่อนข้อมูลที่เป็นหน่วยความจำ หรือก็คือจำนวนวงจรเลื่อนข้อมูลที่เก็บค่าของข้อมูลที่เป็นอดีตของชุดข้อมูลที่เข้าสู่วงจรเข้ารหัส มีจำนวน K-1 ตัว

โดยการเปลี่ยนระบบจากระบบแบบอนุกรมไปเป็นระบบแบบขนานดังที่แสดงในรูปที่ 2.2 และรูปที่ 2.3




**รูปที่ 2.2** รูปแสดงวงจรเข้ารหัสคอนโวลูชันแบบอนุกรม[1]

รูปที่ 2.2 เป็นภาพแสดงวงจรเข้ารหัสคอนโวลูชันแบบอนุกรมที่นิยามระบบได้ดังนี้ (3,2,1) คือวงจรเข้ารหัสมีการรับข้อมูลเข้าครั้งละ 2 บิต ได้คำรหัสครั้งละ 3 บิต และมีหน่วยความจำ 1 บิต โดยมีค่า K = 2




**รูปที่ 2.3** รูปแสดงวงจรเข้ารหัสคอนโวลูชันแบบขนาน[1]

รูปที่ 2.3 เป็นภาพแสดงวงจรเข้ารหัสคอนโวลูชันแบบขนานที่นิยามระบบได้ดังนี้ (3,2,1) คือวงจรเข้ารหัสมีการรับข้อมูลเข้าครั้งละ 2 บิต คือ $u^{(1)}$ และ $u^{(2)}$ ได้คำรหัสครั้งละ 3 บิต และมีหน่วยความจำ 1 บิต โดยค่า K = 2

ในการหาค่าของฟังก์ชันตัวก่อกำเนิดนั้นเราทำได้โดยกำหนดเป็นเวคเตอร์จำนวน n เวคเตอร์ โดยแต่ละเวคเตอร์แทนวงจรบวกแบบมอดูโลทูแต่ละตัว โดยที่เรากำหนดให้ค่าของวงจรเลื่อนข้อมูลทุกตัวในวงจรมีค่าเป็น 0 ทั้งหมด และให้ข้อมูลเข้าจำนวน k บิตชุดแรกเป็น 1 ทั้งหมด นอกจากนั้นให้เป็น 0 ทั้งหมด เพื่อหาผลตอบสนองของวงจรบวกแบบมอดูโลทูแต่ละตัว อาจกล่าวได้ว่ามีค่าเป็น 1 เมื่อวงจรเลื่อนข้อมูลเชื่อมต่อกับวงจรบวกแบบมอดูโลทู และมีค่าเป็น 0 เมื่อวงจรเลื่อนข้อมูลไม่ได้เชื่อมต่อกับวงจรบวกแบบมอดูโลทู

โดยการกำหนดค่าตัวแปรต่างๆ ดังนี้

ข้อมูลขาเข้า

$$u = (u_0, u_1, u_2, \ldots, u_L) \tag{2.1}$$

สมการที่ 2.1 คือข้อมูลเข้า u ที่มีความยาว L+1 โดยมีลำดับตั้งแต่บิตที่ 0 ไปจนถึงบิตที่ L

ฟังก์ชันก่อกำเนิดย่อย

$$g^{(i)} = \left(g_0^{(i)}, g_1^{(i)}, g_2^{(i)}, \ldots, g_m^{(i)}\right) \tag{2.2}$$

สมการที่ 2.2 คือฟังก์ชันก่อกำเนิดย่อย g อันดับที่ i ที่มีความยาว K บิต โดยมีลำดับตั้งแต่วงจรเลื่อนข้อมูล ตัวที่ 0 ไปจนถึงตัวที่ m โดยที่ i คืออันดับย่อยของคำรหัส มีค่าอยู่ในช่วง $1 \le i \le n$

คำรหัสคอนโวลูชันย่อย

$$c^{(i)} = \sum_{h=1}^{k} u^{(h)} * g_h^{(i)} \tag{2.3}$$

โดยเครื่องหมาย * คือเครื่องหมายตัวกระทำการคอนโวลูชันแบบไม่ต่อเนื่อง

สมการที่ 2.3 คือคำรหัสคอนโวลูชันย่อย c อันดับที่ i สามารถหาได้จากผลรวมของการคอนโวลูชันแบบไม่ต่อเนื่องระหว่างข้อมูลเข้า u อันดับที่ h กับฟังก์ชันก่อกำเนิดย่อย g อันดับที่ i ที่ได้รับผลจากข้อมูลเข้า u อันดับที่ h ผลของสมการที่ 2.3 ทำให้สามารถเขียนแสดงผลลัพธ์ได้ว่าคำรหัสคอนโวลูชันย่อยสามารถเขียนได้เป็น

$$c^{(i)} = \left(c_0^{(i)}, c_1^{(i)}, c_2^{(i)}, \ldots, c_{m+L}^{(i)}\right) \tag{2.4}$$

สมการที่ 2.4 คือคำรหัสคอนโวลูชันย่อย c อันดับที่ i ที่มีความยาว m+L+1 โดยมีลำดับตั้งแต่บิตที่ 0 ไปจนถึงบิตที่ m+L

คำรหัสคอนโวลูชัน

$$c = \left(c_0^{(1)}, c_0^{(2)}, \ldots, c_0^{(n)}, c_1^{(1)}, c_1^{(2)}, \ldots, c_1^{(n)}, c_2^{(1)}, c_2^{(2)}, \ldots, c_2^{(n)}, \ldots, c_{m+L}^{(1)}, c_{m+L}^{(2)}, \ldots, c_{m+L}^{(n)}\right) \quad (2.5)$$

สมการที่ 2.5 คือคำรหัสคอนโวลูชัน c ที่เรียงลำดับจากอันดับของคำรหัสย่อยที่ i อันเกิดจากผลกระทบของข้อมูลเข้า u ลำดับที่ l-m โดยที่ l อยู่ในช่วง $0 \le l \le L$ โดยเมื่อ l-m น้อยกว่า 0 ให้ u มีค่าเป็น 0 และไม่นำมาคิด หรืออาจสามารถหาค่าของ c แต่ละตัวในสมการ 2.5 ได้จากความสัมพันธ์ของการคอนโวลูชันแบบไม่ต่อเนื่องของสมการที่ 2.3 ได้ว่า

$$c_l^{(i)} = \sum_{h=1}^{k} \sum_{j=0}^{m} u_{l-j}^{(h)} g_{j,h}^{(i)} \quad (2.6)$$

สมการที่ 2.6 คือสมการที่หาค่าของคำรหัสคอนโวลูชัน c แต่ละตัวโดยที่ l คือลำดับของข้อมูลเข้า u และ i คืออันดับของคำรหัสย่อย อันเป็นผลจากผลรวมของ $\sum_{j=0}^{m} u_{l-j}^{(h)} g_{j,h}^{(i)}$ โดยที่ h = 1,2,...,k คืออันดับของข้อมูลเข้า u และ $\sum_{j=0}^{m} u_{l-j}^{(h)} g_{j,h}^{(i)}$ คือผลรวมของผลคูณของสายของข้อมูลเข้าอันดับที่ h ลำดับที่ l-j โดยที่ j = 0,1,...,m

เช่น จากรูปที่ 2.2 ได้ค่าของฟังก์ชันก่อกำเนิดที่มีความยาว Kk ดังนี้

$$\begin{aligned} g^{(1)} &= [1011] \\ g^{(2)} &= [1101] \\ g^{(3)} &= [1010] \end{aligned} \quad (2.7)$$

จากสมการที่ 2.3 ทำให้สามารถหาค่าคำรหัสคอนโวลูชันย่อยได้ดังนี้

$$\begin{aligned} c^{(1)} &= u * g^{(1)} \\ c^{(2)} &= u * g^{(2)} \\ c^{(3)} &= u * g^{(3)} \end{aligned} \quad (2.8)$$

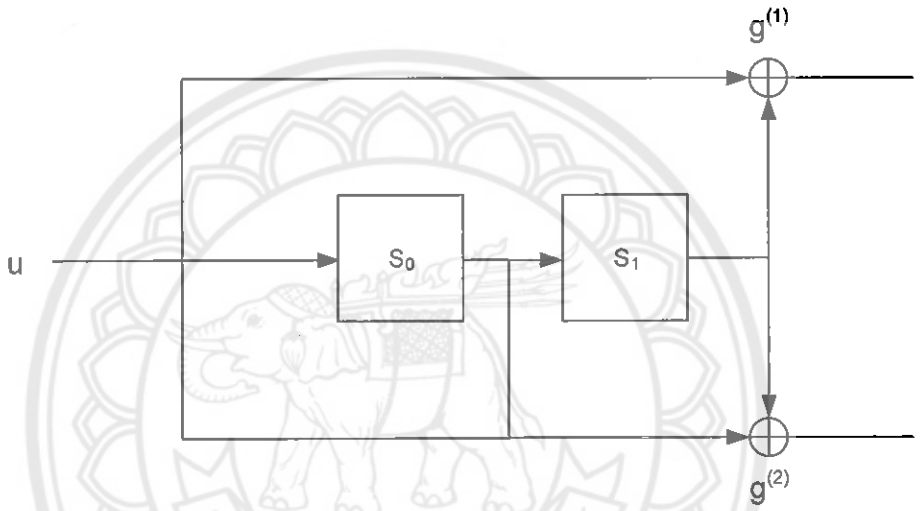
แต่เมื่อแปลงให้อยู่ในรูปของระบบแบบขนานในรูปที่ 2.3 แล้วนั้น จะได้ค่าของฟังก์ชันก่อกำเนิดที่มีความยาว K ดังนี้

$$\begin{aligned} g_1^{(1)} &= [0\ 1] & g_2^{(1)} &= [1\ 1] \\ g_1^{(2)} &= [1\ 1] & g_2^{(2)} &= [1\ 0] \\ g_1^{(3)} &= [0\ 0] & g_2^{(3)} &= [1\ 1] \end{aligned} \quad (2.9)$$

จากสมการที่ 2.3 ทำให้สามารถหาค่าคำรหัสคอนโวลูชันย่อยได้ดังนี้

$$c^{(1)} = u^{(1)} * g_1^{(1)} + u^{(2)} * g_2^{(1)}$$
$$c^{(2)} = u^{(1)} * g_1^{(2)} + u^{(2)} * g_2^{(2)} \quad (2.10)$$
$$c^{(3)} = u^{(1)} * g_1^{(3)} + u^{(2)} * g_2^{(3)}$$

**ตัวอย่างที่ 2.1** [3] ของวงจรเข้ารหัสคอนโวลูชันเมื่อกำหนดข้อมูลเข้าคือ u = 10110




**รูปที่ 2.4** รูปแสดงตัวอย่างแผนภาพแบบบล็อกของวงจรเข้ารหัส[3]

รูปที่ 2.4 เป็นวงจรเข้ารหัสที่มีระบบเป็น (2,1,2) คือมีการนำเข้าข้อมูลครั้งละ 1 บิตสู่วงจรเข้ารหัส และจะได้ผลลัพธ์เป็นคำรหัสจำนวนครั้งละ 2 บิต และค่า K = 3 คือการคำนวณหาคำรหัสของวงจรโดยมีข้อมูลที่ถูกนำเข้า 1 บิต และข้อมูลในอดีตที่อยู่ในวงจรเลื่อนข้อมูล 2 บิตและมี 2 ฟังก์ชันตัวก่อกำเนิดย่อยที่แต่ละฟังก์ชันมีความยาว 3 บิตได้แก่ สมการที่ 2.11 และ 2.12 ดังนี้

$$g^{(1)} = [1\,0\,1] \quad (2.11)$$

$$g^{(2)} = [1\,1\,1] \quad (2.12)$$

ซึ่งการคำนวณหาคำรหัสค่าของฟังก์ชันก่อกำเนิดย่อยทั้ง 2 ฟังก์ชันจะถูกนำมาคำนวณโดยการนำข้อมูลที่ต้องการทำการเข้ารหัสคอนโวลูชัน นั้นมาคอนโวลูท กับฟังก์ชันก่อกำเนิดย่อยแต่ละฟังก์ชัน ซึ่งจะได้ผลลัพธ์ย่อยออกมา ซึ่งจากสมการที่ 2.3 จะได้ผลลัพธ์ดังสมการที่ 2.13 และ 2.14 ดังนี้

$$c^{(1)} = u * g^{(1)} \quad (2.13)$$

$$c^{(1)} = (10110) * (101) = 1001110$$

$$c^{(2)} = u * g^{(2)} \qquad (2.14)$$

$$c^{(2)} = (10110) * (111) = 1100010$$

จากนั้นนำคำรหัสย่อยมาเรียงต่อกันโดยความสัมพันธ์ของสมการที่ 2.5 ได้ว่า เมื่อข้อมูลที่ทำการเข้ารหัสคอนโวลูชันคือ u = 10110 จะได้คำรหัสคอนโวลูชันคือ c = 11010010101100

หรือ อีกวิธีสามารถหาได้โดยอาศัยความสัมพันธ์ของฟังก์ชันก่อกำเนิด[1]

ในตัวอย่างที่ 2.1 จากความสัมพันธ์ของสมการที่ 2.3 สมการที่ 2.5 และสมการที่ 2.6 แล้วทำให้ได้ว่า จากสมการที่ 2.6 นั้น

จากคำรหัสย่อยอันดับที่ 1 ได้ว่าสามารถหาคำรหัสบิตที่ 1 ได้จาก สมการที่ 2.15

$$c_l^{(1)} = u_l g_0^{(1)} + u_{l-1} g_1^{(1)} + u_{l-2} g_2^{(1)} \qquad (2.15)$$

$$= u_l + u_{l-2} \qquad (2.16)$$

เนื่องจากผลการแทนค่าทำให้ได้ผลลัพธ์ดังสมการที่ 2.16 จะพบว่าคำรหัสย่อยอันดับที่ 1 ได้รับผลจากข้อมูลเข้า ณ เวลานั้น และข้อมูลในอดีตที่อยู่ในวงจรเลื่อนข้อมูล $S_1$ เนื่องจาก $g_1^{(1)} = 0$ ทำให้พจน์ $u_{l-1} g_1^{(1)} = 0$

จากคำรหัสย่อยอันดับที่ 2 ได้ว่าสามารถหาคำรหัสบิตที่ 1 ได้จาก สมการที่ 2.17

$$c_l^{(2)} = u_l g_0^{(2)} + u_{l-1} g_1^{(2)} + u_{l-2} g_2^{(2)} \qquad (2.17)$$

$$= u_l + u_{l-1} + u_{l-2} \qquad (2.18)$$

เนื่องจากผลการแทนค่าทำให้ได้ผลลัพธ์ดังสมการที่ 2.18 จะพบว่าคำรหัสย่อยอันดับที่ 1 ได้รับผลจากข้อมูลเข้า ณ เวลานั้น ข้อมูลในอดีตที่อยู่ในวงจรเลื่อข้อมูล $S_0$ และวงจรเลื่อนข้อมูล $S_1$

และจากความสัมพันธ์ของสมการที่ 2.3 และสมการที่ 2.5 จึงกำหนดพารามิเตอร์ G ขึ้น เรียกว่าเมตริกซ์ก่อกำเนิด (Generator matrix : G) ซึ่งจากที่กล่าวมานี้ได้ G มีค่าดังสมการที่ 2.19

$$G = \begin{bmatrix} g_0^{(1)} & g_0^{(2)} & g_1^{(1)} & g_1^{(2)} & g_2^{(1)} & g_2^{(2)} & & & & & & & & \\ & & g_0^{(1)} & g_0^{(2)} & g_1^{(1)} & g_1^{(2)} & g_2^{(1)} & g_2^{(2)} & & & & & & \\ & & & & g_0^{(1)} & g_0^{(2)} & g_1^{(1)} & g_1^{(2)} & g_2^{(1)} & g_2^{(2)} & & & & \\ & & & & & & g_0^{(1)} & g_0^{(2)} & g_1^{(1)} & g_1^{(2)} & g_2^{(1)} & g_2^{(2)} & & \\ & & & & & & & & g_0^{(1)} & g_0^{(2)} & g_1^{(1)} & g_1^{(2)} & g_2^{(1)} & g_2^{(2)} \end{bmatrix} \qquad (2.19)$$

โดย G เกิดจากการนำค่าของฟังก์ชันก่อกำเนิดมาเรียงต่อสลับกันแล้วนำมาวางตามแถวโดยวางตัวแรกของแต่ละแถวห่างจากตัวแรกของแถวก่อนหน้าตามจำนวน m ซึ่งจำนวนแถวนี้ขึ้นอยู่กับจำนวนของข้อมูลเข้านั่นเอง เมื่อแทนค่า ฟังก์ชันก่อกำเนิดจากสมการที่ 2.11 และ 2.12 แล้วจะได้ G มีค่าดังสมการที่ 2.20

$$= \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \quad (2.20)$$

จากสมการที่ 2.3 เมื่อ u = 10110 จะได้ว่า สามารถหาคำรหัสได้ดังสมการที่ 2.21

$$c = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \quad (2.21)$$

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

สังเกตเห็นได้ว่าคำรหัสที่ได้จากข้อมูลชุดนี้จะได้ 10 บิตแรกมีค่าเหมือนกันกับวิธีการก่อนหน้านี้ แต่ 4 บิตหลังนั้นเป็นผลของการทำคอนโวลูชันแบบไม่ต่อเนื่อง จึงทำให้ในวิธีการก่อนหน้านี้ มีการเพิ่มเติมบิตหางเพื่อขจัดค่าต่างๆ ของข้อมูลชุดนี้ออกจากข้อมูลชุดต่อไปนั่นเอง

จากการวิเคราะห์ระบบในรูปที่ 2.4 พบว่าค่า m = 2 นั้นมีสถานะจากวงจรเลื่อนข้อมูล ที่เป็นไปได้ทั้งหมด $2^2 = 4$ สถานะ ดังนี้ 00 01 10 และ 11 จึงสามารถนำมาสร้างตารางที่แสดงถึงข้อมูลขาเข้า สถานะก่อนที่ข้อมูลจะเข้าสู่วงจรเข้ารหัส สถานะที่เปลี่ยนไปหลังจากที่ข้อมูลเข้าสู่วงจรเข้ารหัส และคำรหัสคอนโวลูชันในแต่ละกรณี ดังตารางที่ 2.1

**ตารางที่ 2.1** ตารางแสดงค่าสถานะก่อน และหลังข้อมูลเข้าสู่วงจร และค่าคำรหัสที่ได้

| ข้อมูล | สถานะเริ่มต้น | | สถานะที่เปลี่ยนไปด้วยข้อมูล | | คำรหัส (c) |
|---|---|---|---|---|---|
| | $S_0$ | $S_1$ | $S_0$ | $S_1$ | |
| 0 | 0 | 0 | 0 | 0 | 00 |
| 1 | 0 | 0 | 1 | 0 | 11 |
| 0 | 0 | 1 | 0 | 0 | 11 |

**ตารางที่ 2.1 (ต่อ)** ตารางแสดงค่าสถานะก่อน และหลังข้อมูลเข้าสู่วงจร และค่าคำรหัสที่ได้

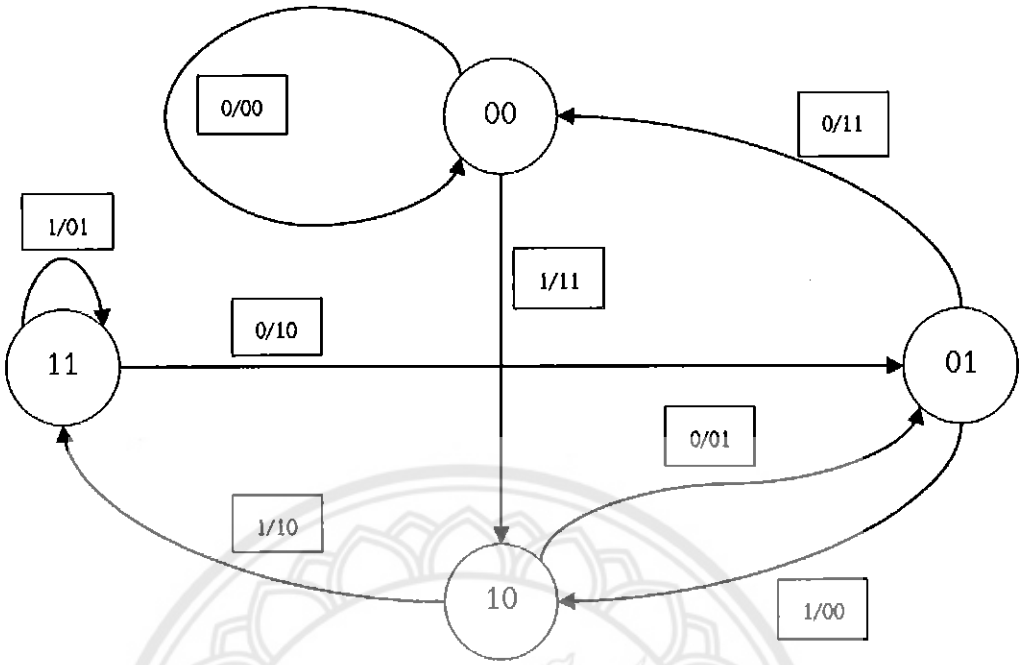
| ข้อมูล | สถานะเริ่มต้น | | สถานะที่เปลี่ยนไปด้วยข้อมูล | | คำรหัส (c) |
|---|---|---|---|---|---|
| | $S_0$ | $S_1$ | $S_0$ | $S_1$ | |
| 1 | 0 | 1 | 1 | 0 | 00 |
| 0 | 1 | 0 | 0 | 1 | 01 |
| 1 | 1 | 0 | 1 | 1 | 10 |
| 0 | 1 | 1 | 0 | 1 | 10 |
| 1 | 1 | 1 | 1 | 1 | 01 |

จากตารางที่ 2.1 เนื่องจากข้อมูลเป็นแบบไบนารี จะเห็นได้ว่าในหนึ่งสถานะนั้นจะมีโอกาสที่ข้อมูล 0 หรือ 1 เข้ามาสู่ระบบก็ได้ เช่น สมมติว่า ถ้าสถานะเริ่มต้นของวงจรเลื่อนข้อมูลในรูปที่ 2.4 มีค่า $S_0S_1 = 00$ เมื่อมีข้อมูลเข้า ณ เวลานั้นคือ $u = 0$ ค่าสถานะที่เปลี่ยนไปของวงจรเลื่อนข้อมูลคือ $S_0S_1 = 00$ ได้คำรหัส $c^{(1)}c^{(2)} = 00$ และถ้าสถานะเริ่มต้นของวงจรเลื่อนข้อมูลในรูปมีค่า $S_0S_1 = 00$ เมื่อมีข้อมูลเข้า ณ เวลานั้นคือ $u = 1$ ค่าสถานะที่เปลี่ยนไปของวงจรเลื่อนข้อมูลคือ $S_0S_1 = 10$ ได้คำรหัส $c^{(1)}c^{(2)} = 11$

จากนี้จะกล่าวถึงทฤษฎีที่ใช้วิเคราะห์วงจรเข้ารหัสคอนโวลูชัน อันได้แก่ แผนภาพสถานะ (State diagram) แผนภาพต้นไม้ (Tree diagram) และแผนภาพเทรลลิส (Trellis diagram)

### 2.1.1 แผนภาพสถานะ (State diagram) [1,4]

แผนภาพสถานะเป็นวิธีหนึ่งที่ใช้อธิบายกาเข้ารหัสคอนโวลูชัน โดยจะแสดงค่าของข้อมูลที่อยู่ภาพในวงจรเลื่อนข้อมูล ค่าข้อมูลเข้า และผลลัพธ์คำรหัสที่ได้ โดยจะให้ค่าสถานะอยู่ภายในวงกลมนั้น คือสถานะของวงจรเลื่อนข้อมูล และลูกศรแสดงความสัมพันธ์ของการเปลี่ยนแปลงของสถานะหนึ่งไปเป็นอีกสถานะหนึ่งเมื่อได้รับข้อมูลเข้าไปในวงจร ณ เวลานั้น จากตารางที่ 2.1 สามารถนำมาสร้างแผนภาพสถานะได้ดังรูปที่ 2.5

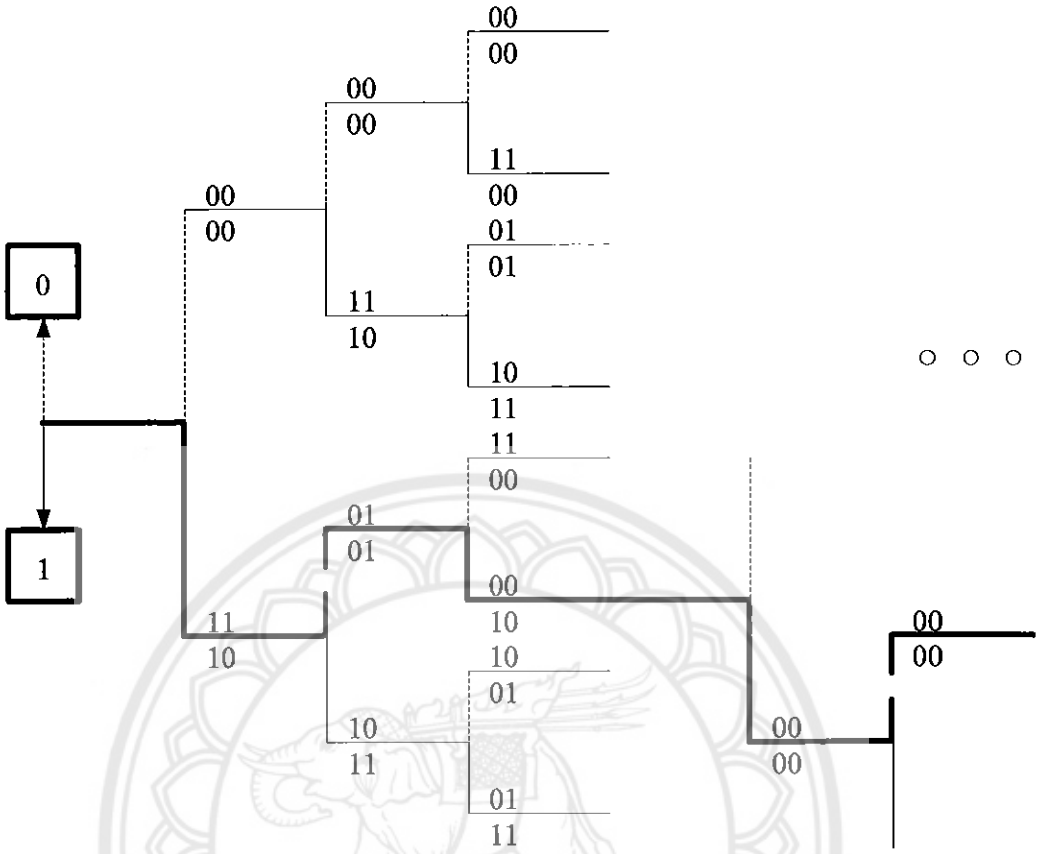



**รูปที่ 2.5** รูปแสดงแผนภาพสถานะของตัวอย่างที่ 2.1[1]

รูปที่ 2.5 สถานะที่เป็นไปได้ทั้งหมด 4 สถานะ ถูกนำไปใส่ไว้ในวงกลมแต่ละวงโดยมีลูกศรชี้ถึงการเปลี่ยนแปลงของสถานะหนึ่งไปเป็นอีกสถานะหนึ่งของวงจรเลื่อนข้อมูลด้วยผลของข้อมูลที่เข้ามายังวงจร และที่ลูกศรแต่ละเส้นจะมีถูกกำกับ ข้อมูล/คำรหัส โดยที่ทิศทางลูกศรที่ชี้เข้าหาวงกลมแต่ละวงนั้นจะเท่ากับค่า $2^k = 2^1 = 2$ เส้น และทิศทางลูกศรที่ชี้ออกจากวงกลมแต่ละวงนั้นจะเท่ากับค่า $n = 2$ เส้น เช่น ที่สถานะ 00 เมื่อมีข้อมูลเข้ามาเป็น 0 จะได้คำรหัส 00 และสถานะต่อไปคือสถานะ 00 เมื่อข้อมูลเข้ามาเป็น 1 จะได้คำรหัส 11 และสถานะต่อไปคือสถานะ 10 และที่สถานะ 10 เมื่อมีข้อมูลเข้ามาเป็น 0 จะได้คำรหัส 01 และสถานะต่อไปคือสถานะ 01 เมื่อข้อมูลเข้ามาเป็น 1 จะได้คำรหัส 10 และสถานะต่อไปคือสถานะ 11 เป็นต้น

### 2.1.2 แผนภาพต้นไม้ (Tree diagram) [1,4]

แผนภาพต้นไม้นั้นเป็นอีกวิธีหนึ่งที่จะสามารถใช้อธิบายการเข้ารหัสคอนโวลูชัน โดยพิจารณาเส้นทางเดินของข้อมูลเข้า คำรหัสที่ได้ สถานะที่เปลี่ยนแปลงไปของวงจรเลื่อนข้อมูล ของวงจรเข้ารหัสในตัวอย่างที่ 2.1 สามารถสร้างแผนภาพต้นไม้ได้ดังรูปที่ 2.6

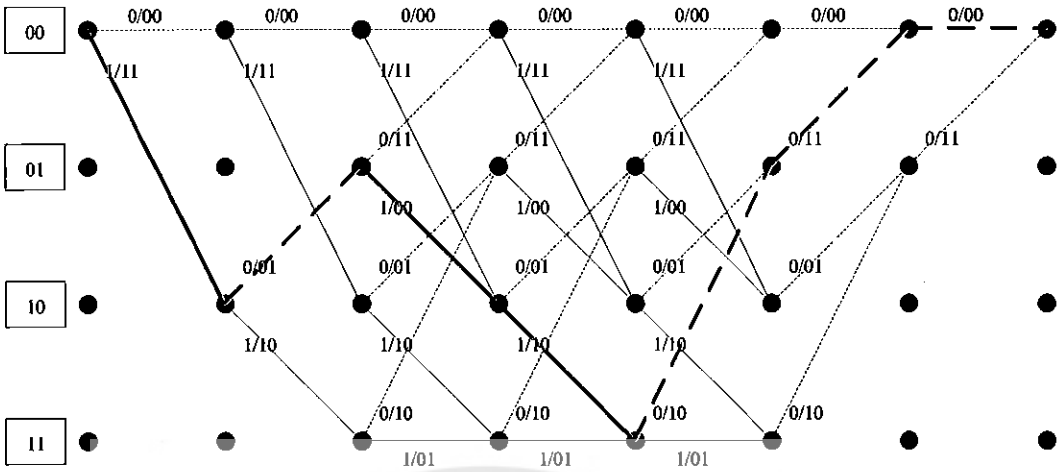



**รูปที่ 2.6** รูปแสดงแผนภาพต้นไม้ของตัวอย่างที่ 2.1[1]

รูปที่ 2.6 แผนภาพต้นไม้นี้ กำหนดให้เมื่อข้อมูลที่เข้ามาเป็น 0 นั้นจะกำหนดเส้นทางให้มีทิศขึ้นบน และเมื่อข้อมูลที่เข้ามาเป็น 1 นั้นจะกำหนดเส้นทางให้มีทิศลงล่าง และจะสังเกตเห็นว่ามีการกำกับไว้โดย คำรหัส/สถานะ โดยเส้นทางเดินจะกำหนดให้เดินทางจากซ้ายไปขวา เช่น จากรูปนี้จะแสดงการเข้ารหัสด้วยระบบจากตัวอย่างที่ 2.1 เมื่อข้อมูลเข้าคือ u = 10110 แสดงโดยเส้นสีแดง ซึ่งจะได้คำรหัสว่า c = 11 01 00 10 10

### 2.1.3 แผนภาพเทรลลิส (Trellis diagram) [1,4]

แผนภาพเทรลลิสก็เป็นอีกหนึ่งวิธีที่ใช้อธิบายการเข้ารหัสคอนโวลูชัน โดยจะเป็นการแสดงข้อมูลต่างๆ ที่เข้ามาในวงจรเข้ารหัสครั้งละ k บิต ผลลัพธ์คำรหัสที่ได้จำนวน n บิต สถานะของวงจรเลื่อนข้อมูล สถานะที่เปลี่ยนไปของวงจรเลื่อนข้อมูล และทางเดินของข้อมูลต่างๆ จากตารางที่ 2.1 สามารถแสดงได้ดังรูปที่ 2.7




รูปที่ 2.7 รูปแสดงแผนภาพเทรลลิสของตัวอย่างที่ 2.1[1]

รูปที่ 2.7 ด้านซ้ายมือของรูปแสดงถึงสถานะที่เป็นไปได้ทั้งหมด 4 สถานะคือ 00 01 10 และ 11 ถูกกำกับไว้ เมื่อข้อมูลเข้ามายังวงจรเข้ารหัส k = 1 บิตแล้วจึงมีความเป็นไปได้ที่ข้อมูลจะเข้ามาได้ทั้งหมด 2 สถานะ คือ 0 หรือ 1 เท่านั้นจึงทำให้ในหนึ่งสถานะจะมีเส้นกิ่งออกมา 2 เส้น โดยแต่ละเส้นกิ่งจะถูกกำกับด้วย ข้อมูล/คำรหัส จะเห็นได้ว่าเส้นกิ่งเหล่านี้จะเชื่อมถึงกันระหว่างสถานะหนึ่งไปยังอีกสถานะหนึ่ง เมื่อชุดข้อมูลที่เข้ามามีความยาวเป็นจำนวนชุด k บิตจะทำให้เกิดเส้นทางเดินขึ้น เส้นทางเดินนี้จะบ่งบอกถึงสถานะ คำรหัสที่ได้จากทางเดินของข้อมูลนี้ โดยเริ่มต้นจากทางด้านซ้ายมือ เริ่มที่สถานะของวงจรเลื่อนข้อมูลมีค่าเป็น 00 แล้วเดินต่อไปทางขวามือ และจะจบลงที่สถานะ 00 เช่นเดิม ซึ่งจากตัวอย่างที่ 2.1 นั้นข้อมูลเข้า u = 10110 ได้เดินทางตามเส้นสีแดงจากการเปลี่ยนแปลงสถานะดังนี้คือ $00 \rightarrow 10 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 01 \rightarrow 00 \rightarrow 00$ และคำรหัสที่ได้คือ c = 11 01 00 10 10 11 00 จะเห็นได้สถานะของวงจรเข้ารหัสจะกลับไปเป็นสถานะ 00 ดังเดิม ซึ่งโดยทั่วไปแล้วจะมีการเติมบิตหางเข้าไปจำนวน m = 2 บิตเข้าต่อจากข้อมูลขาเข้า เพื่อทำให้ค่าสถานะของวงจรกลับเป็น 00 ดังเดิม

จะเห็นได้ว่าไม่ว่าจะใช้วิธีการใดไม่ว่าจะเป็น แผนภาพสถานะ แผนภาพต้นไม้ แผนภาพเทรลลิส หรือวิธีการทางคณิตศาสตร์ในการวิเคราะห์วงจรเข้ารหัสคอนโวลูชันจะได้ผลลัพธ์ของคำรหัสที่ได้เหมือนกัน

## 2.2 ระยะฟรี (Free distance หรือ $d_{free}$) [1,5,6]

พารามิเตอร์หนึ่งที่มีผลต่อประสิทธิภาพของรหัสคอนโวลูชันคือ ระยะฟรี (Free distance) ซึ่งเป็นระยะห่างที่น้อยที่สุด ระหว่างคำรหัสสองคำในรหัสคอนโวลูชันหนึ่งๆ ระยะฟรีนี้จะมีผลคล้ายกับระยะใกล้ที่สุด (Minimum distance) ของรหัสแบบบล็อก โดยถ้าระยะนี้มีค่ามากจะทำให้

รหัสมีโอกาสที่จะแก้ไขความผิดพลาดได้มากขึ้น เนื่องจากเมื่อคำรหัสแต่ละคำมีความแตกต่างกันมากเพียงพอที่จะทำให้ตัวถอดรหัสตัดสินใจเลือกคำรหัสที่ถูกต้องได้ง่ายขึ้น ในรูปสมการอาจเขียนระยะฟรีได้ดังนี้

$$d_{free} = \min\{d(r', r'') : u' \neq u''\} \tag{2.22}$$

สมการที่ 2.22 ระยะฟรีคือระยะแฮมมิ่งที่น้อยที่สุด เมื่อ r' และ r" คือคำรหัสของข้อมูล u' และ u" ตามลำดับ โดยที่ $u' \neq u''$ และ $d(r', r'')$ คือระยะแฮมมิ่งระหว่างคำรหัส r' และ r"

จากการที่รหัสคอนโวลูชันเป็นรหัสเชิงเส้น เนื่องจากระบบการเข้ารหัสเป็นระบบเชิงเส้นดังที่ได้อธิบายไว้แล้ว ดังนั้นระยะฟรีจึงสามารถหาได้จากน้ำหนักของคำรหัสที่ไม่เป็นศูนย์ใดๆ โดยไม่ขึ้นกับความยาว เพราะน้ำหนักของคำรหัสไบนารีคือจำนวนบิตที่มีค่าเป็นหนึ่ง ซึ่งจำนวนบิตที่เป็นหนึ่งนี้ก็เท่ากับระยะจากคำรหัสศูนย์

$$d_{free} = \min\{w(r) : v \neq 0\} \tag{2.23}$$

สมการที่ 2.23 ระยะฟรี คือน้ำหนักแฮมมิ่งที่น้อยที่สุดของคำรหัส v เมื่อ $v \neq 0$

เนื่องจากคำรหัสคอนโวลูชันมีความยาวไม่จำกัด แต่ในการใช้งานจริงที่ข้อมูลมีค่าจำกัดนั้น จะมีการป้อนบิต 0 เข้าไป เพื่อให้ข้อมูลในหน่วยความจำถูกเข้ารหัสออกมาเป็นบิตหาง และปรับให้หน่วยความจำมีค่าเป็นศูนย์ทั้งหมด ดังนั้นสามารถพิจารณาได้ว่าคำรหัสเริ่มจากสถานะศูนย์ และสิ้นสุดที่สถานะศูนย์เสมอ โดยสามารถใช้แผนภาพเทรลลิสช่วยในการหาคำรหัสเหล่านี้ได้ซึ่งเป็นวิธีที่เห็นภาพชัดเจน นอกจากนั้นยังสามารถใช้หลักการซ้อนทับของระบบเชิงเส้นซึ่งเป็นวิธีที่รวดเร็ว รวมทั้งยังมีวิธีที่ใช้กราฟการไหลของสัญญาณร่วมกับสูตรการหาค่าอัตราขยายของเมสันซึ่งจะทำให้ข้อมูลครบถ้วนมากยิ่งขึ้นโดยสามารถแจกแจงได้ว่าคำรหัสกี่คำที่มีค่าน้ำหนักนั้นๆ

โดยสามารถจัดทำตารางแสดงค่าระยะฟรีที่น้อยที่สุด ค่าของฟังก์ชันก่อกำเนิด และค่าความยาวจำกัดสำหรับอัตรารหัสที่หลากหลาย ได้ดังตารางที่ 2.2-2.4 เป็นต้น

**ตารางที่ 2.2** ตารางแสดงค่าต่างๆ สำหรับอัตรารหัส 1/2 [1]

| ค่าความยาวจำกัด K | ฟังก์ชันก่อกำเนิดในเลขฐาน 8 | | $d_{free}$ |
|---|---|---|---|
| 3 | 5 | 7 | 5 |
| 4 | 15 | 17 | 6 |
| 5 | 23 | 35 | 7 |

**ตารางที่ 2.2 (ต่อ)** ตารางแสดงค่าต่างๆ สำหรับอัตรารหัส 1/2 [1]

| ค่าความยาวจำกัด K | ฟังก์ชันก่อกำเนิดในเลขฐาน 8 | | $d_{free}$ |
|---|---|---|---|
| 6 | 53 | 75 | 8 |
| 7 | 133 | 171 | 10 |
| 8 | 247 | 371 | 10 |
| 9 | 561 | 753 | 12 |
| 10 | 1,167 | 1,545 | 12 |
| 11 | 2,335 | 3,661 | 14 |
| 12 | 4,335 | 5,723 | 15 |
| 13 | 10,533 | 17,661 | 16 |
| 14 | 21,675 | 27,123 | 16 |

จากตารางที่ 2.2 ได้ดังนี้ เมื่ออัตรารหัสคือ 1/2 ค่าของเมตริกซ์ก่อกำเนิดในตารางถูกกำหนดในรูปของเลขฐาน 8 การหาค่าพารามิเตอร์ของเมตริกซ์ก่อกำเนิดนั้นสามารถหาได้ดังเช่น เมื่อความยาวจำกัดของวงจรเข้ารหัส K = 3 ได้ค่าของระบบคือ (2,1,2) ค่า $d_{free}$ น้อยที่สุดที่ได้คือ 5 จะได้เมตริกซ์ก่อกำเนิด มีค่า 5 และ 7 ซึ่งจะได้ค่าของเมตริกซ์ก่อกำเนิดในรูปของเลขฐาน 2 คือ 101 และ 111 ในรูปแบบของวงจรเข้ารหัสแบบขนาน เมื่อความยาวจำกัดของวงจรเข้ารหัส K = 4 ได้ค่าของระบบคือ (2,1,3) ค่า $d_{free}$ น้อยที่สุดที่ได้คือ 6 จะได้เมตริกซ์ก่อกำเนิด มีค่า 15 และ 17 ซึ่งจะได้ค่าของเมตริกซ์ก่อกำเนิดในรูปของเลขฐาน 2 คือ 1101 และ 1111 ในรูปแบบของวงจรเข้ารหัสแบบขนาน เป็นต้น

**ตารางที่ 2.3** ตารางแสดงค่าต่างๆ สำหรับอัตรารหัส 1/3 [1]

| ค่าความยาวจำกัด K | ฟังก์ชันก่อกำเนิดในเลขฐาน 8 | | | $d_{free}$ |
|---|---|---|---|---|
| 3 | 5 | 7 | 7 | 10 |
| 4 | 13 | 15 | 17 | 13 |

**ตารางที่ 2.3 (ต่อ)** ตารางแสดงค่าต่างๆ สำหรับอัตรารหัส 1/3 [1]

| ค่าความยาวจำกัด K | ฟังก์ชันก่อกำเนิดในเลขฐาน 8 | | | $d_{free}$ |
|---|---|---|---|---|
| 5 | 25 | 33 | 37 | 16 |
| 6 | 47 | 53 | 75 | 18 |
| 7 | 133 | 145 | 175 | 20 |
| 8 | 225 | 331 | 367 | 22 |
| 9 | 557 | 663 | 711 | 24 |
| 10 | 1,117 | 1,365 | 1,633 | 27 |
| 11 | 2,353 | 2.671 | 3,175 | 29 |
| 12 | 4,767 | 5,723 | 6,265 | 32 |
| 13 | 10,533 | 10,675 | 17,661 | 33 |
| 14 | 21,645 | 35,661 | 37,133 | 36 |

จากตารางที่ 2.3 ได้ดังนี้ เมื่ออัตรารหัสคือ 1/3 ค่าของเมตริกซ์ก่อกำเนิดในตารางถูกกำหนดในรูปของเลขฐาน 8 การหาค่าพารามิเตอร์ของเมตริกซ์ก่อกำเนิดนั้นสามารถหาได้ดังเช่น เมื่อความยาวจำกัดของวงจรเข้ารหัส K = 3 ได้ค่าของระบบคือ (3,1,2) ค่า $d_{free}$ น้อยที่สุดที่ได้คือ 10 จะได้เมตริกซ์ก่อกำเนิด มีค่า 5, 7 และ 7 ซึ่งจะได้ค่าของเมตริกซ์ก่อกำเนิดในรูปของเลขฐาน 2 คือ 101, 111 และ 111 ในรูปแบบของวงจรเข้ารหัสแบบขนาน เมื่อความยาวจำกัดของวงจรเข้ารหัส K = 4 ได้ค่าของระบบคือ (3,1,3) ค่า $d_{free}$ น้อยที่สุดที่ได้คือ 13 จะได้เมตริกซ์ก่อกำเนิด มีค่า 13, 15 และ 17 ซึ่งจะได้ค่าของเมตริกซ์ก่อกำเนิดในรูปของเลขฐาน 2 คือ 1011, 1101 และ 1111 ในรูปแบบของวงจรเข้ารหัสแบบขนาน เป็นต้น

**ตารางที่ 2.4** ตารางแสดงค่าต่างๆ สำหรับอัตรารหัส 1/4 [1]

| ค่าความยาวจำกัด K | ฟังก์ชันก่อกำเนิดในเลขฐาน 8 | | | | $d_{free}$ |
|---|---|---|---|---|---|
| 3 | 5 | 7 | 7 | 7 | 10 |

**ตารางที่ 2.4 (ต่อ)** ตารางแสดงค่าต่างๆ สำหรับอัตรารหัส 1/4 [1]

| ค่าความยาวจำกัด K | ฟังก์ชันก่อกำเนิดในเลขฐาน 8 | | | | $d_{free}$ |
|---|---|---|---|---|---|
| 4 | 13 | 15 | 15 | 17 | 13 |
| 5 | 25 | 27 | 33 | 37 | 16 |
| 6 | 53 | 67 | 71 | 75 | 18 |
| 7 | 135 | 135 | 147 | 163 | 20 |
| 8 | 235 | 275 | 313 | 357 | 22 |
| 9 | 463 | 535 | 733 | 745 | 24 |
| 10 | 1,117 | 1,365 | 1,633 | 1,653 | 27 |
| 11 | 2,327 | 2,353 | 2,671 | 3,175 | 29 |
| 12 | 4,767 | 5,723 | 6,265 | 7,455 | 32 |
| 13 | 11,145 | 12,477 | 15,537 | 16,727 | 33 |
| 14 | 21,113 | 23,175 | 35,527 | 35,537 | 36 |

จากตารางที่ 2.4 ได้ดังนี้ เมื่ออัตรารหัสคือ 1/4 ค่าของเมตริกซ์ก่อกำเนิดในตารางถูกกำหนดในรูปของเลขฐาน 8 การหาค่าพารามิเตอร์ของเมตริกซ์ก่อกำเนิดนั้นสามารถหาได้ดังเช่น เมื่อความยาวจำกัดของวงจรเข้ารหัส K = 3 ได้ค่าของระบบคือ (4,1,2) ค่า $d_{free}$ น้อยที่สุดที่ได้คือ 10 จะได้เมตริกซ์ก่อกำเนิด มีค่า 5, 7, 7 และ 7 ซึ่งจะได้ค่าของเมตริกซ์ก่อกำเนิดในรูปของเลขฐาน 2 คือ 101, 111, 111 และ 111 ในรูปแบบของวงจรเข้ารหัสแบบขนาน เมื่อความยาวจำกัดของวงจรเข้ารหัส K = 4 ได้ค่าของระบบคือ (4,1,3) ค่า $d_{free}$ น้อยที่สุดที่ได้คือ 13 จะได้เมตริกซ์ก่อกำเนิด มีค่า 13, 15, 15 และ 17 ซึ่งจะได้ค่าของเมตริกซ์ก่อกำเนิดในรูปของเลขฐาน 2 คือ 1011, 1101, 1101 และ 1111 ในรูปแบบของวงจรเข้ารหัสแบบขนาน เป็นต้น

## 2.3 การถอดรหัสคอนโวลูชัน (Convolutional decoding) [4,6]

การถอดรหัสคอนโวลูชันนั้นมีความซับซ้อนมากกว่าการเข้ารหัส จึงได้มีการนำเสนอกระบวนการถอดรหัสอยู่หลายวิธี หนึ่งในวิธีที่ได้รับความนิยม และถูกกล่าวถึงในโครงงานนี้ คือ ขั้นตอนวิธีของวิเทอร์บี (Viterbi algorithm) โดยแบ่งเป็น 2 วิธีการ คือการถอดรหัสด้วยการ

ตัดสินใจแบบฮาร์ด และการถอดรหัสด้วยการตัดสินใจแบบซอฟต์ โดยทั้ง 2 วิธีจะมีลักษณะคล้ายๆกัน แต่จะใช้ระยะห่างที่แตกต่างกันซึ่งในการตัดสินใจแบบฮาร์ดนั้นจะใช้ระยะแฮมมิ่ง แต่ในการตัดสินใจแบบซอฟต์จะใช้ระยะยูคลิดแล้วทำการหาเส้นทางในแผนภาพเทรลลิสต่อไปดังนี้

### 2.3.1 การถอดรหัสด้วยขั้นตอนวิธีของวิเทอร์บีด้วยการตัดสินใจแบบฮาร์ด

การตัดสินใจแบบฮาร์ด (Hard-decision) เป็นการถอดรหัสด้วยวิธีความเป็นไปได้สูงสุด (Maximum likelihood) ซึ่งจะพิจารณาว่าเส้นทางใดใกล้เคียงกับคำรหัสที่ได้รับมากที่สุด และจะทำการตัดสินข้อมูลโดยมีการตั้งค่าเทรสโฮลด์ไว้ โดยจะตัดสินข้อมูลที่ได้รับมาว่าเป็น 0 หรือ 1 แล้วนำไปใช้ในการหาเส้นทางเดินที่ดีที่สุด

### 2.3.2 การถอดรหัสด้วยขั้นตอนวิธีของวิเทอร์บีด้วยการตัดสินใจแบบซอฟต์

การตัดสินใจแบบซอฟต์ (Soft-decision) จะมีกระบวนการที่ซับซ้อนกว่าการตัดสินใจแบบฮาร์ด โดยจะพิจารณาถึงข้อมูลที่ได้รับมาโดยยังไม่ตัดสินว่าข้อมูลที่ได้รับมาเป็นบิต 0 หรือ 1 แต่นำข้อมูลที่ได้รับมาแล้วนั้น นำไปคำนวณหาเมตริกซ์ที่ใช้คำนวณหาทางเดินที่เป็นไปได้สูงที่สุดเปรียบเทียบกับเส้นทางเดินต่างๆ ของข้อมูลที่ได้รับก่อนหน้า

โดยขั้นตอนวิธีของวิเทอร์บีจะคำนวณหาระยะห่างของข้อมูลที่ได้รับเข้ามา และแทนค่าเหล่านั้นลงในเส้นทางต่างๆ เพื่อใช้ในการตัดสินใจในการเลือกเส้นทางที่มีความเป็นไปได้สูงที่สุด โดยค้นหาเส้นทางที่อยู่ในแผนภาพเทรลลิสที่มีความใกล้เคียงกับข้อมูลที่ได้รับมามากที่สุด เพื่อนำข้อมูลในเส้นทางนั้นมาคำนวณ โดยมีกระบวนการทำงานดังนี้

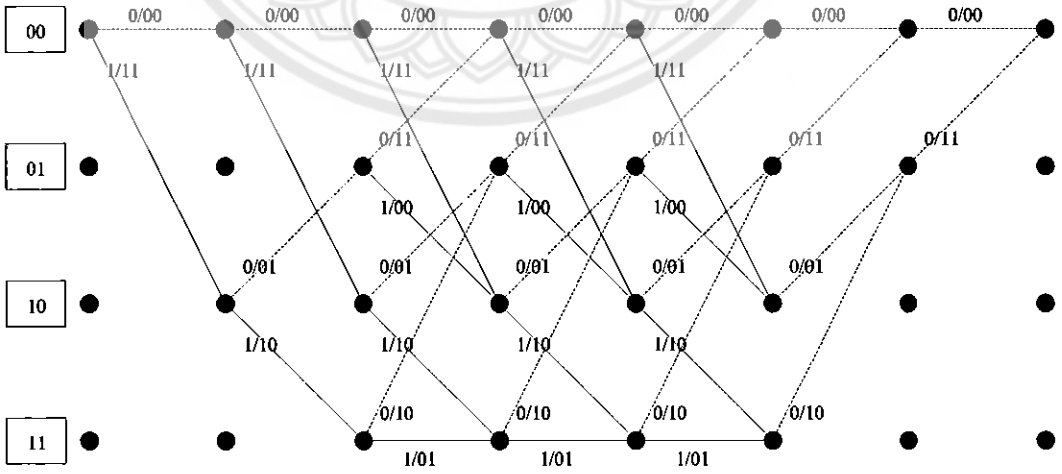
1. พิจารณาแบ่งข้อมูลที่ได้รับมาเป็นข้อมูลย่อยจำนวน L ชุด แต่ละชุดมีจำนวน n บิต
2. ทำการวาดแผนภาพเทรลลิสที่มีจำนวนสถานะเท่ากับ $2^m$ สถานะ มีจำนวนระดับในการทำงานยาว L+1 ระดับ โดยที่ m = K-1 (K คือความยาวจำกัด หรือ Constraint length) และจะพิจารณาเฉพาะเส้นทางที่มีความเป็นไปได้ที่จะถูกส่งมาเท่านั้น โดยระดับที่ l = L-K+1 ขึ้นไปของแผนภาพเทรลลิสนั้น ให้วาดเฉพาะเส้นทางที่เข้าหาสถานะของวงจรเป็น 0 ทั้งหมด
3. กำหนดค่าตัวแปรระดับ l = 0 เป็นค่าเริ่มต้น และทำการกำหนดค่าเริ่มต้นของตัวแปรเมตริกซ์ในสภาวะเริ่มต้นที่มีข้อมูลเป็น 0 ทั้งหมดให้เมตริกซ์มีค่าเป็น 0
4. ทำการหาค่าระยะของข้อมูลที่ได้รับชุดที่ l+1 กับข้อมูลในเส้นทางในการเปลี่ยนแปลงสถานะของแผนภาพเทรลลิสในระดับที่ l ไปยังระดับที่ l+1
5. นำค่าระยะที่ได้ไปบวกกับเมตริกซ์สะสม เพื่อใช้ในการตัดสินใจเลือกเส้นทางที่เป็นไปได้มากที่สุดในการเปลี่ยนจากสถานะในระดับที่ l ไปยังอีกสถานะในระดับที่ l+1

6. พิจารณาในแต่ละสถานะของระดับที่ l นั้นในแต่ละระดับจะเลือกเส้นทางที่มีค่าเมตริกซ์สะสมน้อยที่สุดที่เข้าหาแต่ละสถานะโดยเส้นทางที่ถูกเลือกนั้นจะถูกเรียกว่า "Survivor" ซึ่งจะเป็นเส้นทางที่ถูกเก็บไว้ในการคำนวณครั้งต่อไป และสำหรับเส้นทางที่ไม่ถูกเลือกนั้นจะถูกเรียกว่า "Forgetting" โดยจะถูกละทิ้งไปจากกระบวนการตัดสินใจ และให้ l เพิ่มขึ้น 1 ระดับ แล้วกลับไปทำข้อที่ 4 วนไปเรื่อยๆ จนเมื่อ l+1 มากกว่า L-K+1 ขึ้นไป จึงทำข้อ 7 ต่อไป
7. เริ่มต้นพิจารณา ณ ระดับที่ l เมื่อ l+1 มากกว่า L-K+1 ที่ระดับนี้ข้อมูลเป็น 0 ทั้งหมด ทำการเลือกเส้นทางที่เป็น "Survivor" ซึ่งเป็นเส้นทางที่ถูกเลือกที่เหลืออยู่ย้อนกลับไปจนถึงสถานะเริ่มต้นของการทำงานที่มีสถานะเป็น 0 ทั้งหมด ซึ่งเส้นทางที่ได้นั้นจะเป็นเส้นทางของข้อมูลที่ใกล้เคียงกับข้อมูลที่ได้รับเข้ามา จึงจะถูกนำไปใช้คำนวณหาข้อมูลโดยข้อมูลที่จะถูกส่งออกไปจากภาคถอดรหัสนั้นจะเป็นข้อมูลทั้งหมดที่อยู่ในเส้นทางที่ถูกเลือก ยกเว้นบิตหางจะถูกตัดทิ้งไป

ตัวอย่างที่ 2.2[3] การถอดรหัสด้วยขั้นตอนวิธีของวิเทอร์บีด้วยการตัดสินใจแบบฮาร์ด

ดังผลที่ได้จากตัวอย่างที่ 2.1 เมื่อมีข้อมูลเข้า u = 10110 เข้าไปในวงจรเข้ารหัสคอนโวลูชันดังกล่าวแล้วนั้น จะได้คำรหัสคอนโวลูชัน c = 11010010101100 แล้วกระทำตามกระบวนการถอดรหัสดังนี้

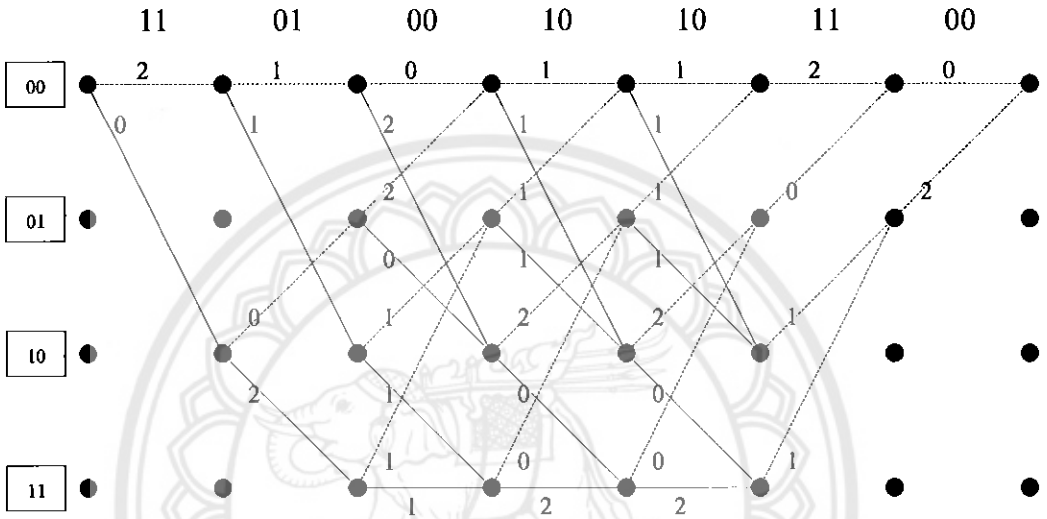
1) แบ่งคำรหัสออกเป็น 7 ชุด ชุดละ 2 บิต ได้ 11 01 00 10 10 11 00 และสร้างแผนภาพเทรลลิสได้ดังแสดงในรูปที่ 2.8 ดังนี้




**รูปที่ 2.8** รูปแสดงแผนภาพเทรลลิส

รูปที่ 2.8 รูปภาพนี้แสดงถึงแผนภาพเทรลลิสของวงจรถอดรหัส ซึ่งเหมือนกันกับแผนภาพเทรลลิสของวงจรเข้ารหัส

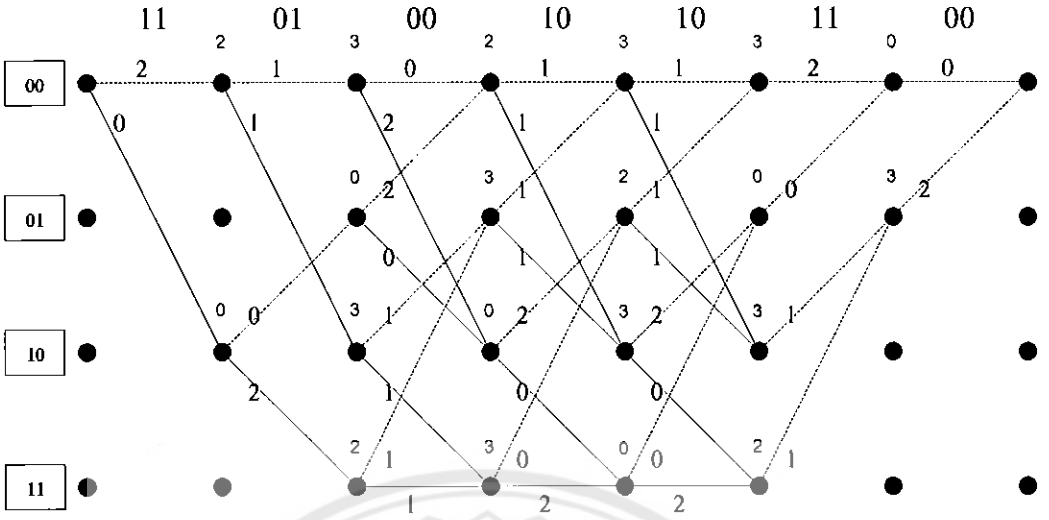
2) นำแต่ละชุดข้อมูลมาเปรียบเทียบหาค่าระยะแฮมมิงในแต่ละเส้นทางของแต่ละระดับโดยจะแสดงข้อมูลไว้เป็นตัวเลขสีแดงข้างบนของแผนภาพ และแสดงค่าระยะในแต่ละเส้นทางของแต่ละระดับเป็นตัวเลขสีเขียวกำกับไว้กับเส้นทางนั้นๆ ดังรูปที่ 2.9




รูปที่ 2.9 รูปแสดงการหาระยะแฮมมิงที่แต่ละเส้นทางของแต่ละระดับ

รูปที่ 2.9 แผนภาพเทรลลิสที่ถูกหาค่าระยะแฮมมิงโดยเทียบกับข้อมูลจะทำการถอดรหัสกำกับไว้บนภาพเป็นตัวเลขที่เขียว

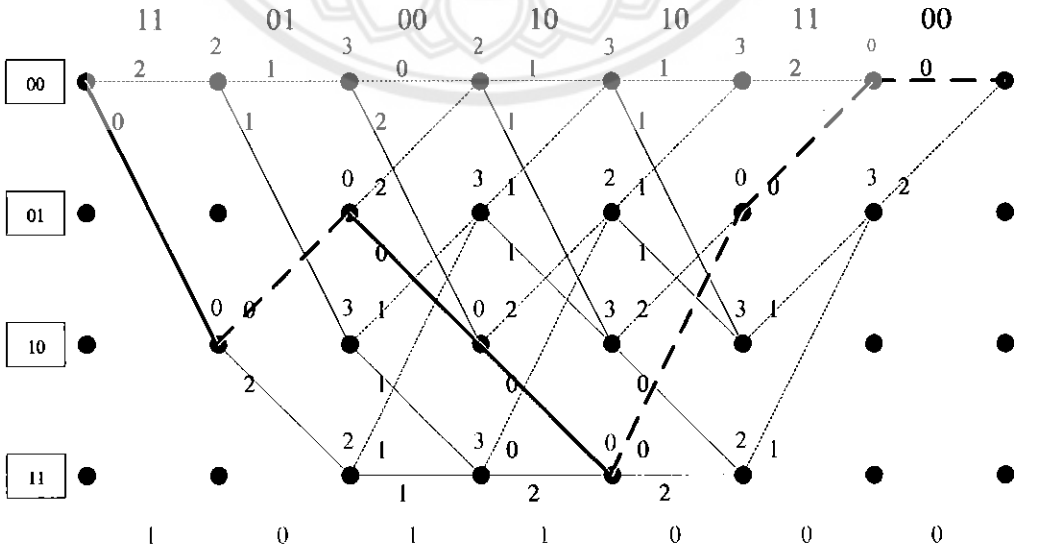
3) หาผลรวมของระยะห่างรวมของทุกเส้นทางที่เป็น "Survivor" โดยกำกับไว้เป็นตัวเลขสีฟ้า บนจุดสถานะของแต่ละระดับ ดังรูปที่ 2.10




**รูปที่ 2.10** รูปแสดงการหาผลรวมของแต่ละเส้นทางที่เป็น "Survivor"

รูปที่ 2.10 แผนภาพเทรลลิสที่หาค่าผลรวมของเส้นทางที่เป็น "Survivor" กำกับไว้บนภาพเป็นตัวเลขสีฟ้า

4) หาเส้นทางที่ผลรวมน้อยที่สุด และหาค่าของข้อมูลที่ถอดรหัสได้จากวงจรถอดรหัส โดยเส้นทางที่มีผลรวมของระยะแฮมมิงที่น้อยที่สุดมีค่าผลรวมเป็น 0 ทำให้ได้ว่าข้อมูลที่ได้จากการถอดรหัสคือ 1011000 โดยแสดงเป็นตัวเลขสีส้มด้านล่างของแผนภาพ ซึ่งในที่นี้ 2 บิตสุดท้าย 00 คือบิตหางไม่ใช่ข้อมูลที่ต้องการจะได้ จึงได้ว่าข้อมูลที่ได้จากการถอดรหัสคือ 10110 ซึ่งได้ผลลัพธ์ของข้อมูลที่ตรงกับ u ของวงจรเข้ารหัส ดังรูปที่ 2.11




**รูปที่ 2.11** รูปแสดงการหาผลรวมที่น้อยที่สุดที่ได้

รูปที่ 2.11 แผนภาพเทรลลิสที่ทำการถอดรหัสได้ข้อมูลที่ต้องการออกมาโดยกำกับด้วยตัวเลขสีส้มด้านล่างของแผนภาพ

โดยสรุปแล้วในบทที่ 2 นี้ได้กล่าวถึงหลักการ และทฤษฎีที่เกี่ยวข้องกับการเข้ารหัสคอนโวลูชัน ซึ่งมีการกล่าวถึงการวิเคราะห์วงจรเข้ารหัสด้วยแผนภาพ 3 ชนิดดังนี้ แผนภาพสถานะ (State diagram) แผนภาพต้นไม้ (Tree diagram) และแผนภาพเทรลลิส (Trellis diagram) ระยะฟรี ซึ่งบ่งบอกถึงประสิทธิภาพของคำรหัสคอนโวลูชันที่ได้จากวงจรเข้ารหัสนั้นมีความแตกต่างกันเพียงพอที่จะทำให้วงจรถอดรหัสสามารถตัดสินใจเลือกคำรหัสที่ถูกต้องได้ และการถอดรหัสคอนโวลูชันให้ได้ข้อมูลที่ต้องการ โดยขั้นตอนวิธีของวิเทอร์บีด้วยการตัดสินใจแบบฮาร์ด

# บทที่ 3
# การออกแบบโครงงานและวิธีการดำเนินงาน

ในบทนี้จะกล่าวถึงขั้นตอนในการดำเนินโครงงาน โครงสร้าง และการออกแบบโปรแกรมจำลองการเข้ารหัสแบบคอนโวลูชัน และการถอดรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บี

## 3.1 ศึกษาการสร้างบิตข้อมูล

ในการสื่อสารข้อมูลดิจิตอลนั้นข้อมูลจะถูกแปลงให้อยู่ในรูปของบิตข้อมูลแบบไบนารีหรือเลขฐานสองจะถูกสร้างขึ้นที่ภาคส่ง ส่งผ่านช่องส่งสัญญาณไปยังภาครับ โดยที่ในหนึ่งบิตข้อมูลจะถูกแบ่งออกเป็นสองสถานะคือ 0 หรือ 1

## 3.2 ศึกษาการเข้ารหัสคอนโวลูชัน

ในขั้นตอนนี้จะทำการศึกษาถึงหลักการทำงาน โครงสร้าง ทฤษฎีที่ใช้ในการออกแบบ และวิเคราะห์พารามิเตอร์ต่างๆ ในวงจรเข้ารหัสคอนโวลูชัน ซึ่งเป็นวิธีหนึ่งที่เพิ่มความสามารถในการตรวจสอบ และแก้ไขความผิดพลาดที่เกิดจากผลกระทบของสัญญาณรบกวนภายในช่องส่งสัญญาณ โดยในการเข้ารหัสจะทำให้ชุดคำรหัสที่ได้มีจำนวนบิตที่มากกว่าข้อมูลที่ถูกนำมาเข้ารหัส โดยวิธีการที่ใช้ทำการเข้ารหัสนั้นได้ถูกแสดงไว้ในบทที่ 2

## 3.3 ศึกษาการสุ่มตำแหน่งบิตที่ผิดพลาด

ในช่องส่งสัญญาณนั้นจะมีสัญญาณอันไม่เป็นที่พึงประสงค์ ซึ่งจะทำให้ข้อมูลที่ต้องการจะส่งผ่านช่องส่งสัญญาณนั้นผิดเพี้ยนไป โดยในการทดลองนี้จะจำลองสัญญาณรบกวน โดยกำหนดให้สัญญาณรบกวนเป็นชุดข้อมูลที่เกิดจากการสุ่มแล้วนำไปผ่านตัววัดระดับ ซึ่งจะทำการกำหนดค่าระดับด้วยค่าของความน่าจะเป็นของการเกิดบิตที่ผิดพลาด ($P_e$) โดยให้ตำแหน่งของข้อมูลของสัญญาณรบกวนที่มีค่าน้อยกว่าค่าระดับเป็นตำแหน่งที่เกิดการผิดเพี้ยนขึ้นของข้อมูลที่ถูกส่งผ่านช่องส่งสัญญาณแบบสมมาตรไบนารีซึ่งจะเป็นไปตามวิธีการในภาคผนวก

## 3.4 ศึกษาการรวมข้อมูลคำรหัสคอนโวลูชัน กับตำแหน่งบิตที่ผิดพลาด

ในขั้นตอนนี้เป็นการจำลองถึงผลกระทบของสัญญาณรบกวนที่มีต่อสัญญาณข้อมูลที่ต้องการจะส่งไปยังภาครับนั้นผิดเพี้ยนไป โดยในช่องส่งสัญญาณนั้นสัญญาณของข้อมูล และ

1722098



สำนักหอสมุด
24 ส.ค. 2561

สัญญาณรบกวนจะบวกทับซ้อนกันเป็นสัญญาณเดียวซึ่งเป็นผลทำให้เกิดความผิดเพี้ยนดังที่ได้กล่าวมาข้างต้น โดยการนำข้อมูลคำรหัสคอนโวลูชันมาบวกกับชุดข้อมูลตำแหน่งบิตที่ผิดพลาด

### 3.5 ศึกษาการถอดคำรหัสคอนโวลูชันโดยขั้นตอนวิธีของวิเทอร์บี

หลังจากสัญญาณผ่านช่องส่งสัญญาณมาแล้วจนถึงภาครับจะทำการถอดคำรหัสคอนโวลูชัน เพื่อให้ได้ข้อมูลจริงที่ภาคส่งต้องการจะส่งมายังภาครับออกมาโดยขั้นตอนวิธีของวิเทอร์บีด้วยการตัดสินใจแบบฮาร์ดดังที่แสดงไว้ในบทที่ 2

### 3.6 การจำลองการทำรหัสคอนโวลูชัน

ในโครงงานนี้จะทำการศึกษาระบบตัวอย่างอันได้แก่ ระบบ [3,1,2] ระบบ [4,1,2] ระบบ [5,1,2] และระบบ [3,2,1] เป็นระบบที่ใช้ในการจำลอง โดยกำหนดให้ข้อมูลคือ 1011100101 จากนั้นทำการจำลองโดยแบ่งเป็นกรณีศึกษาได้ดังนี้

#### 3.6.1 กรณีศึกษาของระบบ [3,1,2]

1. การเข้ารหัสคอนโวลูชัน
2. การถอดคำรหัสคอนโวลูชัน
   - 2.1 กรณีไม่มีบิตผิดพลาด
   - 2.2 กรณีบิตผิดพลาดจำนวน 1 บิต
   - 2.3 กรณีบิตผิดพลาดจำนวน 2 บิตที่ไม่ติดกัน
   - 2.4 กรณีบิตผิดพลาดจำนวน 2 บิตที่ติดกัน และอยู่ต่างคำรหัส
   - 2.5 กรณีบิตผิดพลาดจำนวน 2 บิตที่ติดกัน และอยู่ในคำรหัสเดียวกัน
   - 2.6 กรณีบิตผิดพลาดจำนวน 3 บิต โดยมี 2 บิตที่อยู่ติดกัน แต่อยู่ต่างคำรหัส
   - 2.7 กรณีบิตผิดพลาดจำนวน 3 บิต โดยที่มี 2 บิตติดกัน และอยู่ในคำรหัสเดียวกัน
   - 2.8 กรณีบิตผิดพลาดจำนวน 3 บิตไม่ติดกัน
   - 2.9 กรณีบิตผิดพลาดจำนวน 3 บิตติดกัน

#### 3.6.2 กรณีศึกษาของระบบ [4,1,2]

1. การเข้ารหัสคอนโวลูชัน

2. การถอดคำรหัสคอนโวลูชัน กรณีไม่มีบิตผิดพลาด

### 3.6.3 กรณีศึกษาของระบบ [5,1,2]

1. การเข้ารหัสคอนโวลูชัน

2. การถอดคำรหัสคอนโวลูชัน กรณีไม่มีบิตผิดพลาด

### 3.6.4 กรณีของระบบ [3,2,1]

1. การเข้ารหัสคอนโวลูชัน

2. การถอดคำรหัสคอนโวลูชัน กรณีไม่มีบิตผิดพลาด

## 3.7 การสร้างโปรแกรมจำลองด้วยโปรแกรมแมตแล็บ

ศึกษาวิธีการใช้งานโปรแกรมแมตแล็บเบื้องต้นจะพบว่าสามารถแยกส่วนโปรแกรมออกเป็นส่วนย่อยได้โดยการเขียนฟังก์ชัน จากนั้นในโปรแกรมหลักจะนำแต่ละฟังก์ชันมาใช้งานรวมกัน และการเขียนโปรแกรมจำลองการทำรหัสคอนโวลูชันด้วยโปรแกรมแมตแล็บทำได้ดังนี้

### 3.7.1 การสร้างบิตข้อมูล

ชุดข้อมูลไบนารีจะถูกสร้างจำนวน 1 ล้านบิตโดยโปรแกรม Information_gen.m ด้วยคำสั่ง randi จะเป็นการสุ่มตัวเลขจำนวนเต็มโดยถูกกำหนดให้สุ่มตัวเลข 0 หรือ 1 ด้วยการแจกแจงเอกรูป จากนั้นนำผลลัพธ์ไปเก็บไว้ในไฟล์ Information.txt

### 3.7.2 การเข้ารหัสคอนโวลูชัน

เปิดใช้งานโปรแกรมหลักโดยใช้ชื่อว่า Convolutional_Coding.m ทำการเรียกข้อมูลจากไฟล์ Information.txt จากนั้นแบ่งข้อมูลเป็นชุด ชุดละ 100 บิตแล้วนำไปทำการเข้ารหัสคอนโวลูชันด้วยค่าของระบบที่ถูกกำหนดโดยพารามิเตอร์ (n,k,m) เพื่อให้ได้ผลลัพธ์คำรหัสคอนโวลูชันตามกระบวนการเข้ารหัส ดังที่กล่าวมาแล้วในบทที่ 2 โดยการใช้งานฟังก์ชันย่อยชื่อว่า ConvolutionalEncoder.m

### 3.7.3 การสุ่มตำแหน่งบิตที่ผิดพลาด และการรวมข้อมูล

ในการสุ่มตำแหน่งบิตที่ผิดพลาดนั้น จะทำการสุ่มข้อมูลด้วยคำสั่ง rand ซึ่งเป็นการสุ่มข้อมูลที่มีค่าตั้งแต่ 0 ไปจนถึง 1 ด้วยการแจกแจงเอกรูป จากนั้นกำหนดค่าความน่าจะเป็นของการ

เกิดบิตที่ผิดพลาดให้มีค่า 0.001 0.002 0.004 0.006 0.008 0.01 0.02 0.04 0.06 0.08 0.1 เป็นกรณีศึกษา โดยตั้งค่าระดับด้วยค่าความน่าจะเป็นของการเกิดบิตที่ผิดพลาดเป็นค่าระดับ เมื่อข้อมูลในการสุ่มตำแหน่งบิตที่ผิดพลาด ตำแหน่งใดมีค่าน้อยกว่าค่าระดับ จะกำหนดให้มีค่าเป็น 1 นอกจากนี้จะเป็น 0 ตามวิธีการที่ได้กล่าวไว้ในภาคผนวก แล้วนำตำแหน่งบิตที่ผิดพลาดที่ได้มาบวกกันแบบมอดูโลทู จะทำให้ได้ข้อมูลที่ผ่านช่องส่งสัญญาณมาแล้ว

### 3.7.4 การถอดคำรหัสคอนโวลูชัน

นำข้อมูลที่ได้รับจากช่องส่งสัญญาณมาทำการถอดคำรหัสคอนโวลูชันด้วยการใช้ขั้นตอนวิธีของวิเทอร์บีโดยการตัดสินใจแบบฮาร์ด ซึ่งเป็นฟังก์ชันย่อยชื่อว่า ConvolutionalDecoder.m เพื่อหาข้อมูลที่ภาคส่งสร้างขึ้น

### 3.7.5 จำนวนบิตที่ผิดพลาดที่เกิดขึ้นภายในช่องส่งสัญญาณ และภายหลังการถอดคำรหัส

เมื่อได้ชุดข้อมูลที่เป็นข้อมูลภายในช่องส่งสัญญาณแล้วนั้น จะทำการหาจำนวนบิตที่แตกต่างไปจากคำรหัสคอนโวลูชันที่ภาคส่ง และภายหลังจากการถอดคำรหัสจะได้ชุดข้อมูลนำมาทำการหาจำนวนบิตที่แตกต่างกันกับข้อมูลที่ภาคส่งสร้างขึ้น เพื่อนำไปคำนวณหาค่าอัตราบิตผิดพลาด (BER)

## 3.8 วิเคราะห์ประสิทธิภาพของระบบด้วยอัตราบิตที่ผิดพลาด (BER)

ในการสื่อสารดิจิตอลนั้นจะใช้อัตราบิตที่ผิดพลาดในการระบุถึงประสิทธิภาพของระบบว่ามีประสิทธิภาพในการแก้ไขความผิดเพี้ยนของข้อมูลที่เกิดจากผลกระทบของสัญญาณรบกวน เมื่อส่งผ่านช่องส่งสัญญาณในรูปของสัญญาณดิจิตอลมากน้อยเพียงใด โดยในโครงงานนี้จะทำการเปรียบเทียบผลลัพธ์ของอัตราบิตที่ผิดพลาดของทุกระบบในกรณีศึกษา กับผลลัพธ์จากทฤษฎีในรูปของแผนภูมิ

# บทที่ 4
# ผลการดำเนินโครงงาน

ในบทนี้จะกล่าวถึงผลลัพธ์การดำเนินโครงงานจากการจำลองการเข้ารหัสคอนโวลูชันด้วยระบบที่ใช้เป็นกรณีศึกษา และการถอดคำรหัสด้วยขั้นตอนวิธีของวิเทอร์บีด้วยการตัดสินในแบบฮาร์ดจากความน่าจะเป็นของการเกิดบิตที่ผิดพลาดกรณีต่างๆ

## 4.1 ผลการศึกษาการเข้ารหัสคอนโวลูชัน และการถอดคำรหัสคอนโวลูชัน

ผลการศึกษาในขั้นตอนนี้ทำให้ทราบถึงการสร้างบิตข้อมูล หลักการทำงาน โครงสร้าง ทฤษฎีที่ใช้ในการออกแบบ และวิเคราะห์พารามิเตอร์ต่างๆ ในการเข้ารหัสคอนโวลูชัน และการถอดคำรหัสคอนโวลูชันโดยได้ถูกกล่าวถึงไว้แล้วในบทที่ 2

## 4.2 ผลการศึกษาการสุ่มตำแหน่งบิตที่ผิดพลาด และการรวมข้อมูลในช่องส่งสัญญาณ

ผลการศึกษาในขั้นตอนนี้ทำให้ทราบถึงการเกิดขึ้นของสัญญาณรบกวน และพารามิเตอร์ที่ส่งผลถึงตำแหน่งบิตที่ผิดพลาด คือค่าความน่าจะเป็นของการเกิดบิตที่ผิดพลาด และลักษณะของการรวมกันของข้อมูลจากภาคส่ง กับตำแหน่งบิตที่ผิดพลาดภายในช่องส่งสัญญาณ

## 4.3 ผลการจำลองการทำรหัสคอนโวลูชัน

ค่าพารามิเตอร์ระบบ [n,k,m] มีความหมายดังนี้

n คือจำนวนบิตของคำรหัส หรือเอาต์พุตต่อหนึ่งหน่วยเวลา

k คือจำนวนบิตของข้อมูล หรืออินพุตที่เข้าสู่ระบบต่อหนึ่งหน่วยเวลา

m คือจำนวนชิฟรีจิสเตอร์ที่เป็นหน่วยความจำของระบบ

ในที่นี้กำหนดให้เมทริกซ์ก่อกำเนิดคือ ค่าที่อธิบายลักษณะการเชื่อมต่อของชิฟรีจิสเตอร์กับตัวบวกแบบมอดูโลทูโดยค่า 1 หมายถึงมีการเชื่อมต่อระหว่างกัน และค่า 0 หมายถึงไม่มีการเชื่อมต่อระหว่างกันโดยค่าของเมทริกซ์ก่อกำเนิดจะอยู่ในรูปของเลขฐานแปดซึ่งจะต้องนำมาทำการแปลงไปเป็นเลขฐานสอง โดยจะแสดงผลการจำลองการทำรหัสคอนโวลูชันของกรณีศึกษา 4 กรณี คือ กรณีศึกษาของระบบ [3,1,2] [4,1,2] [5,1,2] และ [3,2,1] ดังต่อไปนี้

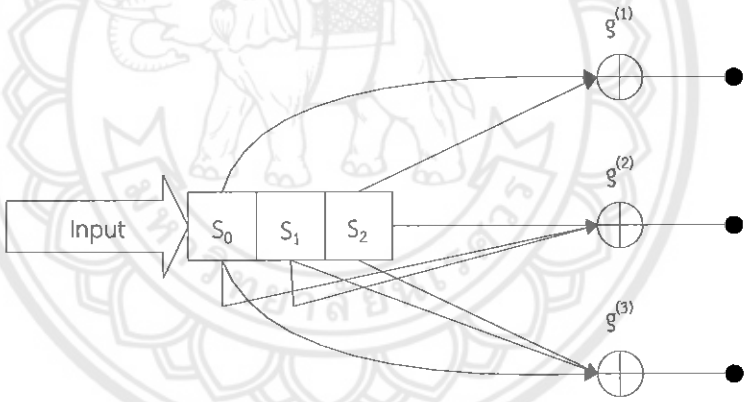
### 4.3.1 กรณีศึกษาของระบบ [3,1,2]

#### 1. การเข้ารหัสคอนโวลูชัน

ระบบ [3,1,2] มีความหมายว่า 3 คือมีจำนวนบิตของคำรหัส 3 บิตต่อหนึ่งหน่วยเวลา
1 คือมีจำนวนบิตข้อมูลที่เข้าสู่ระบบ 1 บิตต่อหนึ่งหน่วยเวลา
2 คือมีจำนวนชิฟรีจิสเตอร์ที่เป็นหน่วยความจำ 2 ชิฟ

ซึ่งจะได้เมทริกซ์ก่อกำเนิด [5,7,7] (ฐาน 8) แปลงได้เป็น [101,111,111] (ฐาน 2)
โดยกำหนดให้ $g^{(1)}$ คือ 101, $g^{(2)}$ คือ 111 และ $g^{(3)}$ คือ 111
และกำหนดให้ $S_0$ คืออินพุต
$S_1$ คือชิฟรีจิสเตอร์ชิฟที่ 1
$S_2$ คือชิฟรีจิสเตอร์ชิฟที่ 2

จะได้แผนภาพจำลองระบบ [3,1,2] ดังรูปที่ 4.1




**รูปที่ 4.1** แผนภาพจำลองระบบ [3,1,2]

รูปที่ 4.1 คือรูปของแผนภาพการจำลองระบบ [3,1,2] จะเห็นได้ว่าระบบนี้อินพุตจะเข้าครั้งละ 1 บิตที่ $S_0$ และได้คำรหัสจำนวน 3 บิต และการเชื่อมต่อกันระหว่างชิฟรีจิสเตอร์กับตัวบวกแบบมอดูโลทูจะได้ค่าของฟังก์ชันก่อกำเนิด เช่น $g^{(1)}$ ซึ่งมีการเชื่อมต่อกับ $S_0$ และ $S_2$ จะมีค่า 101 เป็นต้น

จากรูปที่ 4.1 จะได้ว่า
อินพุต $S_0$ จำนวน 1 บิตมีสัญลักษณ์จำนวน 2 สัญลักษณ์ คือ 0,1
สถานะของหน่วยความจำ $S_1S_2$ จำนวน 2 ชิฟมี 4 สถานะ คือ 00,01,10,11
คำรหัสจำนวน 3 บิต

การคำนวณหาค่าคำรหัส และสถานะถัดไปสามารถหาได้ดังต่อไปนี้

ในกรณีเมื่ออินพุตคือ 0

การหาค่าสถานะถัดไปทำได้โดยให้สถานะเดิมคือ $S_1S_2$ เมื่อมีอินพุต $S_0$ เข้าสู่ระบบจะทำให้ได้ $S_0S_1S_2$ ซึ่ง $S_0S_1$ คือค่าของสถานะถัดไปนั่นเอง

ในกรณีที่สถานะเดิมคือ $S_1S_2 = 00$ เมื่อมีอินพุต 0 จะได้ $S_0S_1S_2 = 000$ ซึ่งจะได้ค่าสถานะถัดไปคือ 00
ในกรณีที่สถานะเดิมคือ $S_1S_2 = 01$ เมื่อมีอินพุต 0 จะได้ $S_0S_1S_2 = 001$ ซึ่งจะได้ค่าสถานะถัดไปคือ 00
ในกรณีที่สถานะเดิมคือ $S_1S_2 = 10$ เมื่อมีอินพุต 0 จะได้ $S_0S_1S_2 = 010$ ซึ่งจะได้ค่าสถานะถัดไปคือ 01
ในกรณีที่สถานะเดิมคือ $S_1S_2 = 11$ เมื่อมีอินพุต 0 จะได้ $S_0S_1S_2 = 011$ ซึ่งจะได้ค่าสถานะถัดไปคือ 01

สามารถคำนวณหาค่ารหัสได้โดย

เมื่อสถานะเดิมคือ $S_1S_2 = 00$ จะได้ $S_0S_1S_2 = 000$
ในกรณีของคำรหัสบิตที่ 1 สามารถหาได้จากการนำ $S_0S_1S_2$ คูณกับ $g^{(1)}$ แบบตัวต่อตัวแล้วนำมาบวกกัน ถ้าได้ค่าผลบวกเกิน 1 ให้หาเศษเหลือเมื่อหารด้วย 2
คำรหัสบิตที่ 1 จะได้ว่า $S_0xg_0^{(1)}+S_1xg_1^{(1)}+S_2xg_2^{(1)} = 0x1+0x0+0x1 = 0$
คำรหัสบิตที่ 2 จะได้ว่า $S_0xg_0^{(2)}+S_1xg_1^{(2)}+S_2xg_2^{(2)} = 0x1+0x1+0x1 = 0$
คำรหัสบิตที่ 3 จะได้ว่า $S_0xg_0^{(3)}+S_1xg_1^{(3)}+S_2xg_2^{(3)} = 0x1+0x1+0x1 = 0$
จะได้คำรหัสคือ 000

เมื่อสถานะเดิมคือ $S_1S_2 = 01$ จะได้ $S_0S_1S_2 = 001$
คำรหัสบิตที่ 1 จะได้ว่า $S_0xg_0^{(1)}+S_1xg_1^{(1)}+S_2xg_2^{(1)} = 0x1+0x0+1x1 = 1$
คำรหัสบิตที่ 2 จะได้ว่า $S_0xg_0^{(2)}+S_1xg_1^{(2)}+S_2xg_2^{(2)} = 0x1+0x1+1x1 = 1$
คำรหัสบิตที่ 3 จะได้ว่า $S_0xg_0^{(3)}+S_1xg_1^{(3)}+S_2xg_2^{(3)} = 0x1+0x1+1x1 = 1$
จะได้คำรหัสคือ 111

เมื่อสถานะเดิมคือ $S_1S_2 = 10$ จะได้ $S_0S_1S_2 = 010$
คำรหัสบิตที่ 1 จะได้ว่า $S_0xg_0^{(1)}+S_1xg_1^{(1)}+S_2xg_2^{(1)} = 0x1+1x0+0x1 = 0$
คำรหัสบิตที่ 2 จะได้ว่า $S_0xg_0^{(2)}+S_1xg_1^{(2)}+S_2xg_2^{(2)} = 0x1+1x1+0x1 = 1$
คำรหัสบิตที่ 3 จะได้ว่า $S_0xg_0^{(3)}+S_1xg_1^{(3)}+S_2xg_2^{(3)} = 0x1+1x1+0x1 = 1$
จะได้คำรหัสคือ 011

เมื่อสถานะเดิมคือ $S_1S_2 = 11$ จะได้ $S_0S_1S_2 = 011$
คำรหัสบิตที่ 1 จะได้ว่า $S_0xg_0^{(1)}+S_1xg_1^{(1)}+S_2xg_2^{(1)} = 0x1+1x0+1x1 = 1$
คำรหัสบิตที่ 2 จะได้ว่า $S_0xg_0^{(2)}+S_1xg_1^{(2)}+S_2xg_2^{(2)} = 0x1+1x1+1x1 = 2 = 0$
คำรหัสบิตที่ 3 จะได้ว่า $S_0xg_0^{(3)}+S_1xg_1^{(3)}+S_2xg_2^{(3)} = 0x1+1x1+1x1 = 2 = 0$
จะได้คำรหัสคือ 100

จากผลลัพธ์ข้างต้น สามารถสร้างเป็นตารางแสดงความสัมพันธ์ระหว่าง อินพุต สถานะเดิม คำรหัส และสถานะถัดไปได้ดังตารางที่ 4.1

**ตารางที่ 4.1** ตารางแสดงค่า อินพุต สถานะเดิม คำรหัส และสถานะถัดไปในกรณีที่อินพุตคือ 0

| อินพุต($S_0$) | สถานะเดิม($S_1S_2$) | คำรหัส | สถานะถัดไป($S_0S_1$) |
|---|---|---|---|
| 0 | 00 | 000 | 00 |
| 0 | 01 | 111 | 00 |
| 0 | 10 | 011 | 01 |
| 0 | 11 | 100 | 01 |

ตารางที่ 4.1 คือตารางแสดงค่า อินพุต สถานะเดิม คำรหัส และสถานะถัดไปในกรณีที่อินพุตคือ 0 โดยการนำผลลัพธ์จากการคำนวณต่างๆ มาสร้างเป็นตารางแสดงความสัมพันธ์ เช่น เมื่อสถานะเดิมคือ 11 เมื่อมีอินพุต เข้าสู่ระบบจะได้คำรหัสคือ 100 และสถานะถัดไปคือ 01 เป็นต้น

และให้ทำด้วยวิธีการเดียวกันในกรณีเมื่ออินพุตคือ 1 จะสามารถสร้างเป็นตารางแสดงความสัมพันธ์ระหว่าง อินพุต สถานะเดิม คำรหัส และสถานะถัดไปได้ดังตารางที่ 4.2

**ตารางที่ 4.2** ตารางแสดงค่า อินพุต สถานะเดิม คำรหัส และสถานะถัดไปในกรณีที่อินพุตคือ 1

| อินพุต($S_0$) | สถานะเดิม($S_1S_2$) | คำรหัส | สถานะถัดไป($S_0S_1$) |
|---|---|---|---|
| 1 | 00 | 111 | 10 |
| 1 | 01 | 000 | 10 |
| 1 | 10 | 100 | 11 |
| 1 | 11 | 011 | 11 |

ตารางที่ 4.2 คือตารางแสดงค่า อินพุต สถานะเดิม คำรหัส และสถานะถัดไปในกรณีที่อินพุตคือ 1 โดยการนำผลลัพธ์จากการคำนวณต่างๆ มาสร้างเป็นตารางแสดงความสัมพันธ์ เช่น เมื่อสถานะเดิมคือ 11 เมื่อมีอินพุตเข้าสู่ระบบจะได้คำรหัสคือ 011 และสถานะถัดไปคือ 11 เป็นต้น

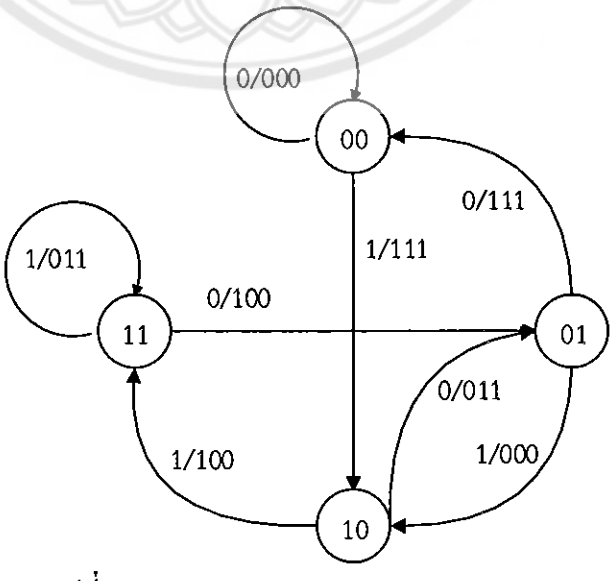
จากตารางที่ 4.1 และ 4.2 สามารถสรุปเป็นตารางแสดงความสัมพันธ์ของระบบ [3,1,2] ได้ดังตารางที่ 4.3

**ตารางที่ 4.3** ตารางสรุปความสัมพันธ์ของระบบ [3,1,2]

| อินพุต($S_0$) | สถานะเดิม($S_1S_2$) | คำรหัส | สถานะถัดไป($S_0S_1$) |
|---|---|---|---|
| 0 | 00 | 000 | 00 |
| 0 | 01 | 111 | 00 |
| 0 | 10 | 011 | 01 |
| 0 | 11 | 100 | 01 |
| 1 | 00 | 111 | 10 |
| 1 | 01 | 000 | 10 |
| 1 | 10 | 100 | 11 |
| 1 | 11 | 011 | 11 |

ตารางที่ 4.3 คือตารางสรุปค่าความสัมพันธ์ของระบบ [3,1,2] ซึ่งได้จากการรวมกันของตารางที่ 4.1 และ 4.2 อธิบายความสัมพันธ์ได้ว่า เช่นให้สถานะเดิมคือ 00 ในกรณีที่อินพุตคือ 0 เข้าสู่ระบบจะได้ผลลัพธ์คำรหัสคือ 000 สถานะถัดไปคือ 00 ในกรณีที่อินพุตคือ 1 เข้าสู่ระบบจะได้ผลลัพธ์คำรหัสคือ 111 สถานะถัดไปคือ 10 เป็นต้น

จากตารางที่ 4.3 สามารถนำมาสร้างเป็นแผนภาพสถานะได้ดังรูปที่ 4.2




**รูปที่ 4.2** แผนภาพสถานะของระบบ [3,1,2]

รูปที่ 4.2 คือแผนภาพสถานะของระบบ [3,1,2] สามารถอธิบายได้เช่น เมื่อที่สถานะเดิมคือ 00 ในกรณีที่อินพุตคือ 0 สถานะถัดไปคือ 00 และได้คำรหัสคือ 000 ในกรณีที่อินพุตคือ 1 สถานะถัดไปคือ 10 และได้คำรหัสคือ 111 เป็นต้น

ตัวอย่างแสดงการเข้ารหัสของระบบ [3,1,2] ที่มีเมทริกซ์ก่อกำเนิดคือ [5,7,7]

กำหนดให้อินพุตคือ 1011100101 และสถานะเริ่มต้นคือ 00 เมื่ออินพุตเข้าสู่ระบบจะทำให้ได้การเปลี่ยนสถานะ และคำรหัสดังตารางที่ 4.4 (โดยในขั้นตอนสุดท้ายจะทำการเติมบิตหางเพื่อปรับให้สถานะของหน่วยความจำกลับไปเป็น 00)

ตารางที่ 4.4 ตารางแสดง อินพุต การเปลี่ยนสถานะ และคำรหัสจากตัวอย่างอินพุต

| อินพุต | 1 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| การเปลี่ยนสถานะ | 10 | 01 | 10 | 11 | 11 | 01 | 00 | 10 | 01 | 10 | 01 | 00 |
| คำรหัส | 111 | 011 | 000 | 100 | 011 | 100 | 111 | 111 | 011 | 000 | 011 | 111 |

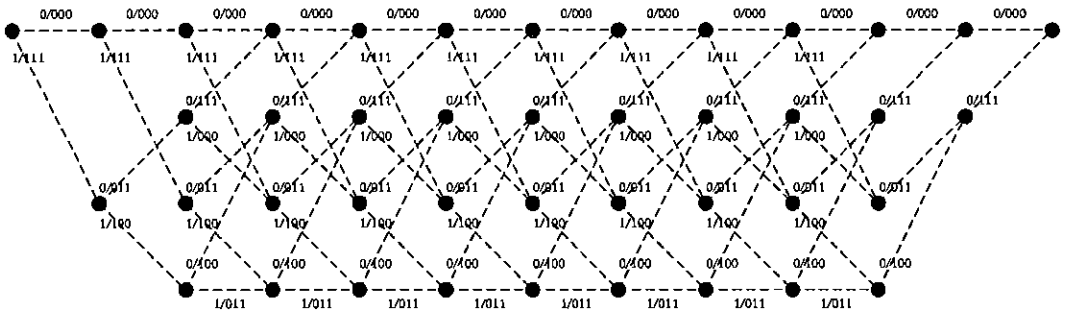
ตารางที่ 4.4 จากการนำอินพุตคือ 1011100101 มาเข้ารหัสคอนโวลูชันจะได้คำรหัสคือ 111 011 000 100 011 100 111 111 011 000 011 111

2. การถอดคำรหัสคอนโวลูชัน

2.1 กรณีไม่มีบิตผิดพลาด

เมื่อนำคำรหัส 111011000100011100111111011000011111 ดังตารางที่ 4.4 มาถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บี จะสามารถแบ่งคำรหัสเป็นชุด จำนวน 12 ชุด ชุดละ 3 บิต ได้ว่าคำรหัสคือ 111 011 000 100 011 100 111 111 011 000 011 111

จากนั้นทำการสร้างแผนภาพเทรลลิสด้วยข้อมูลในตารางที่ 3 จะสามารถสร้างแผนภาพเทรลลิสจำนวน 13 ระดับ ตั้งแต่ระดับที่ 0 ถึงระดับที่ 12 ได้ดังรูปที่ 4.3



**รูปที่ 4.3** แผนภาพเทรลลิสของระบบ [3,1,2]

รูปที่ 4.3 คือแผนภาพเทรลลิสของระบบ [3,1,2] จำนวน 13 ระดับ โดยเริ่มต้นจากสถานะ 0 ณ ระดับที่ 0 และระดับที่ 10 ขึ้นไปจะวาดเฉพาะเส้นทางที่เข้าหาสถานะของวงจรเป็น 0 ทั้งหมด

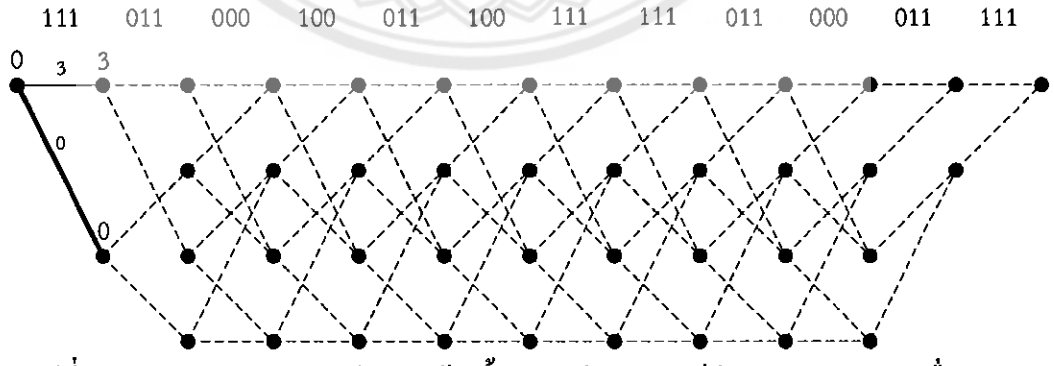
จากนั้นเริ่มต้นด้วยการกำหนดค่าตัวแปรระดับ l = 0 และกำหนดค่าของตัวแปรเมทริกซ์สะสมมีค่า 0 เป็นค่าเริ่มต้น จากนั้นหาค่าระยะระหว่างชุดข้อมูลคำรหัสที่ 1 กับข้อมูลในแต่ละเส้นทางการเปลี่ยนแปลงเข้าสู่แต่ละสถานะในระดับที่ 1 สามารถหาได้ดังนี้

ในระดับที่ 1 หาค่าระยะแฮมมิ่งของชุดคำรหัสที่ 1 กับข้อมูลในแต่ละเส้นทาง

ที่สถานะ 0 เส้นทางจากสถานะ 0 ในระดับที่ 0 หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 000 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 0+3=3

ที่สถานะ 2 เส้นทางจากสถานะ 0 ในระดับที่ 0 หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 111 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 0+0=0

ได้ค่าตัวแปรระดับ l = 1 และได้ค่าเมทริกซ์สะสมใหม่ดังรูปที่ 4.4




**รูปที่ 4.4** การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 0

รูปที่ 4.4 จะเห็นได้ว่าเมื่อผ่านกระบวนการไป 1 ระดับจากระดับที่ 0 ไปยังระดับที่ 1 แล้ว จะได้ค่าระยะบนแต่ละเส้นทาง และค่าเมทริกซ์สะสมใหม่ที่สถานะ 0 คือ 3 และสถานะ 2 คือ 0

**ในระดับที่ 2** หาค่าระยะแฮมมิ่งของชุดคำรหัสที่ 2 กับข้อมูลในแต่ละเส้นทาง

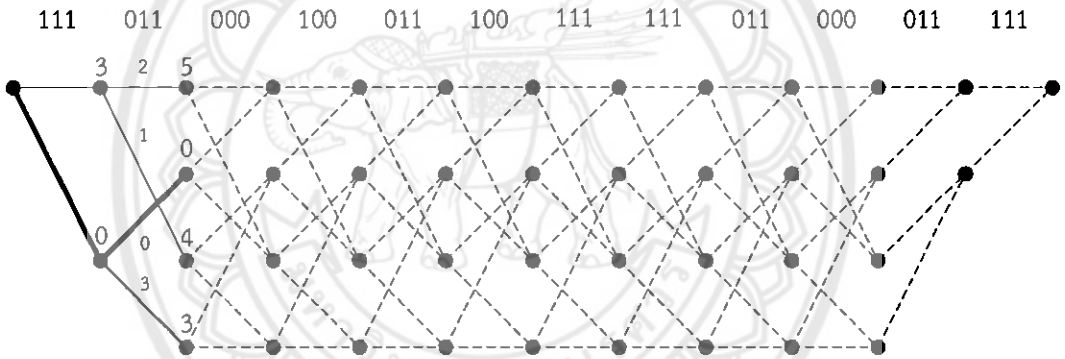
ที่สถานะ 0 เส้นทางจากสถานะ 0 ในระดับที่ 1 หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 000 ได้ 2 หาค่าเมทริกซ์สะสมได้ว่า 3+2=5

ที่สถานะ 1 เส้นทางจากสถานะ 2 ในระดับที่ 1 หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 011 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 0+0=0

ที่สถานะ 2 เส้นทางจากสถานะ 0 ในระดับที่ 1 หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 111 ได้ 1 หาค่าเมทริกซ์สะสมได้ว่า 3+1=4

ที่สถานะ 3 เส้นทางจากสถานะ 2 ในระดับที่ 1 หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 100 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 0+3=3

ได้ค่าตัวแปรระดับ l = 2 และได้ค่าเมทริกซ์สะสมใหม่ดังรูปที่ 4.5




**รูปที่ 4.5** การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 1

รูปที่ 4.5 จะเห็นได้ว่าเมื่อผ่านกระบวนการไปอีก 1 ระดับจากระดับที่ 1 **ไปยังระดับที่ 2 แล้ว จะ**ได้ค่าระยะบนแต่ละเส้นทาง และค่าเมทริกซ์สะสมใหม่ที่สถานะ 0 คือ 5 **สถานะ 1 คือ 0 สถานะ 2 คือ 4 และสถานะ 3 คือ 3**

**ในระดับที่ 3** หาค่าระยะแฮมมิ่งของชุดคำรหัสที่ 3 กับข้อมูลในแต่ละเส้นทาง

ที่สถานะ 0 เส้นทางจากสถานะ 0 และสถานะ 1 ในระดับที่ 2
หาค่าระยะแฮมมิ่งระหว่าง 000 กับ 000 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 5+0=5
หาค่าระยะแฮมมิ่งระหว่าง 000 กับ 111 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 0+3=3 ถูกเลือกเป็นเส้นทางที่เหลือรอด

ที่สถานะ 1 เส้นทางจากสถานะ 2 และสถานะ 3 ในระดับที่ 2
หาค่าระยะแฮมมิ่งระหว่าง 000 กับ 011 ได้ 2 หาค่าเมทริกซ์สะสมได้ว่า 4+2=6

หาค่าระยะแฮมมิ่งระหว่าง 000 กับ 100 ได้ 1 หาค่าเมทริกซ์สะสมได้ว่า 3+1=4 ถูกเลือกเป็นเส้นทางที่เหลือรอด

ที่สถานะ 2 เส้นทางจากสถานะ 0 และสถานะ 1 ในระดับที่ 2

หาค่าระยะแฮมมิ่งระหว่าง 000 กับ 111 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 5+3=8

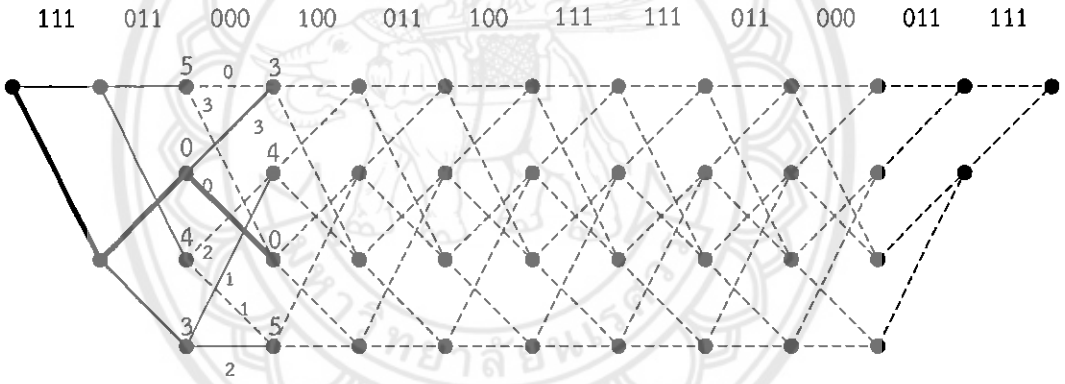
หาค่าระยะแฮมมิ่งระหว่าง 000 กับ 000 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 0+0=0 ถูกเลือกเป็นเส้นทางที่เหลือรอด

ที่สถานะ 3 เส้นทางจากสถานะ 2 และสถานะ 3 ในระดับที่ 2

หาค่าระยะแฮมมิ่งระหว่าง 000 กับ 100 ได้ 1 หาค่าเมทริกซ์สะสมได้ว่า 4+1=5

หาค่าระยะแฮมมิ่งระหว่าง 000 กับ 011 ได้ 2 หาค่าเมทริกซ์สะสมได้ว่า 3+2=5 ถูกเลือกเป็นเส้นทางที่เหลือรอด

ได้ค่าตัวแปรระดับ 1 = 3 และได้ค่าเมทริกซ์สะสมใหม่ดังรูปที่ 4.6




**รูปที่** 4.6 การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 2

รูปที่ 4.6 จะเห็นได้ว่าเมื่อผ่านกระบวนการไปอีก 1 ระดับจากระดับที่ 2 ไปยังระดับที่ 3 แล้ว จะได้ค่าระยะบนแต่ละเส้นทาง และค่าเมทริกซ์สะสมใหม่ที่สถานะ 0 คือ 3 สถานะ 1 คือ 4 สถานะ 2 คือ 0 และสถานะ 3 คือ 5

**ในระดับที่ 4** หาค่าระยะแฮมมิ่งของชุดคำรหัสที่ 4 กับข้อมูลในแต่ละเส้นทาง

ที่สถานะ 0 เส้นทางจากสถานะ 0 และสถานะ 1 ในระดับที่ 3

หาค่าระยะแฮมมิ่งระหว่าง 100 กับ 000 ได้ 1 หาค่าเมทริกซ์สะสมได้ว่า 3+1=4 ถูกเลือกเป็นเส้นทางที่เหลือรอด

หาค่าระยะแฮมมิ่งระหว่าง 100 กับ 111 ได้ 2 หาค่าเมทริกซ์สะสมได้ว่า 4+2=6

ที่สถานะ 1 เส้นทางจากสถานะ 2 และสถานะ 3 ในระดับที่ 3

หาค่าระยะแฮมมิ่งระหว่าง 100 กับ 011 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 0+3=3 ถูกเลือกเป็นเส้นทางที่เหลือรอด

หาค่าระยะแฮมมิ่งระหว่าง 100 กับ 100 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 5+0=5

ที่สถานะ 2 เส้นทางจากสถานะ 0 และสถานะ 1 ในระดับที่ 3

หาค่าระยะแฮมมิ่งระหว่าง 100 กับ 111 ได้ 2 หาค่าเมทริกซ์สะสมได้ว่า 3+2=5 ถูกเลือกเป็นเส้นทางที่เหลือรอด

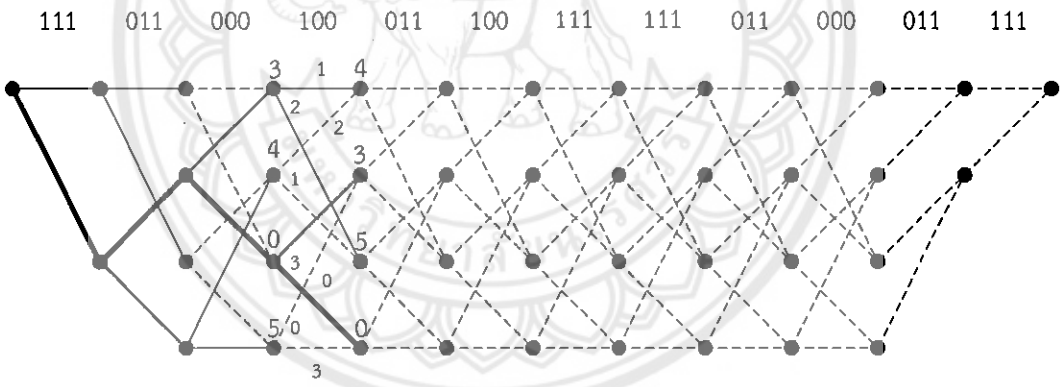
หาค่าระยะแฮมมิ่งระหว่าง 100 กับ 000 ได้ 1 หาค่าเมทริกซ์สะสมได้ว่า 4+1=5

ที่สถานะ 3 เส้นทางจากสถานะ 2 และสถานะ 3 ในระดับที่ 3n

หาค่าระยะแฮมมิ่งระหว่าง 100 กับ 100 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 0+0=0 ถูกเลือกเป็นเส้นทางที่เหลือรอด

หาค่าระยะแฮมมิ่งระหว่าง 100 กับ 011 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 5+3=8

ได้ค่าตัวแปรระดับ l = 4 และได้ค่าเมทริกซ์สะสมใหม่ดังรูปที่ 4.7




**รูปที่ 4.7** การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 3

รูปที่ 4.7 จะเห็นได้ว่าเมื่อผ่านกระบวนการไปอีก 1 ระดับจากระดับที่ 3 ไปยังระดับที่ 4 แล้ว จะได้ค่าระยะบนแต่ละเส้นทาง และค่าเมทริกซ์สะสมใหม่ที่สถานะ 0 คือ 4 สถานะ 1 คือ 3 สถานะ 2 คือ 5 และสถานะ 3 คือ 0

**ในระดับที่ 5** หาค่าระยะแฮมมิ่งของชุดคำรหัสที่ 5 กับข้อมูลในแต่ละเส้นทาง

ที่สถานะ 0 เส้นทางจากสถานะ 0 และสถานะ 1 ในระดับที่ 4

หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 000 ได้ 2 หาค่าเมทริกซ์สะสมได้ว่า 4+2=6

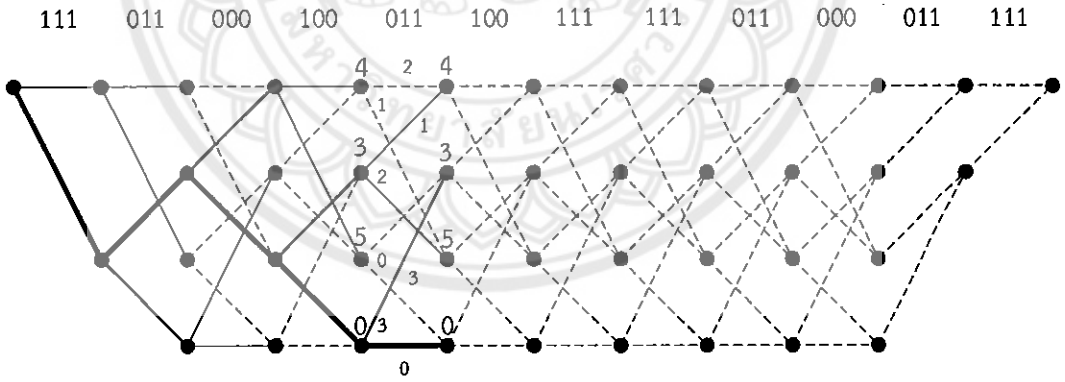
หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 111 ได้ 1 หาค่าเมทริกซ์สะสมได้ว่า 3+1=4 ถูกเลือกเป็นเส้นทางที่เหลือรอด

ที่สถานะ 1 เส้นทางจากสถานะ 2 และสถานะ 3 ในระดับที่ 4
หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 011 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 5+0=5
หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 100 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 0+3=3 ถูกเลือกเป็นเส้นทางที่เหลือรอด

ที่สถานะ 2 เส้นทางจากสถานะ 0 และสถานะ 1 ในระดับที่ 4
หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 111 ได้ 1 หาค่าเมทริกซ์สะสมได้ว่า 4+1=5
หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 000 ได้ 2 หาค่าเมทริกซ์สะสมได้ว่า 3+2=5 ถูกเลือกเป็นเส้นทางที่เหลือรอด

ที่สถานะ 3 เส้นทางจากสถานะ 2 และสถานะ 3 ในระดับที่ 4
หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 100 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 5+3=8
หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 011 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 0+0=0 ถูกเลือกเป็นเส้นทางที่เหลือรอด

ได้ค่าตัวแปรระดับ 1 = 5 และได้ค่าเมทริกซ์สะสมใหม่ดังรูปที่ 4.8




**รูปที่ 4.8** การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 4

รูปที่ 4.8 จะเห็นได้ว่าเมื่อผ่านกระบวนการไปอีก 1 ระดับจากระดับที่ 4 ไปยังระดับที่ 5 แล้ว จะได้ค่าระยะบนแต่ละเส้นทาง และค่าเมทริกซ์สะสมใหม่ที่สถานะ 0 คือ 4 สถานะ 1 คือ 3 สถานะ 2 คือ 5 และสถานะ 3 คือ 0

**ในระดับที่ 6 หาค่าระยะแฮมมิ่งของชุดคำรหัสที่ 6 กับข้อมูลในแต่ละเส้นทาง**

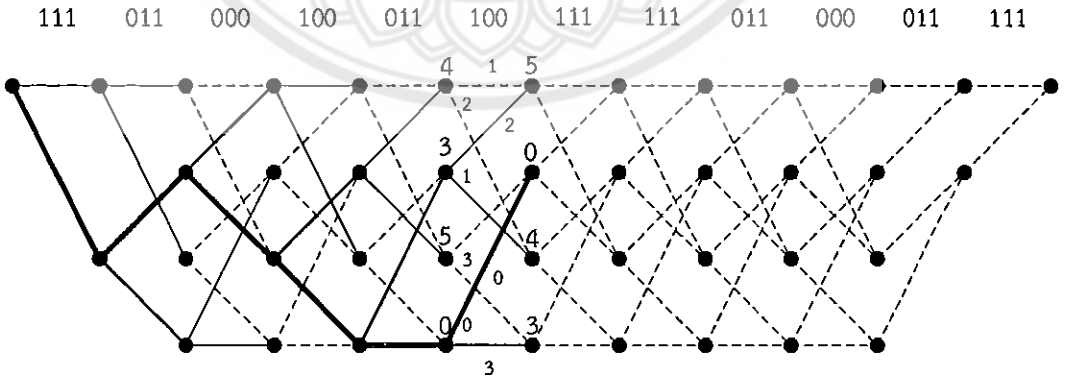
**ที่สถานะ 0 เส้นทางจากสถานะ 0 และสถานะ 1 ในระดับที่ 5**
**หาค่าระยะแฮมมิ่งระหว่าง 100 กับ 000 ได้ 1 หาค่าเมทริกซ์สะสมได้ว่า 4+1=5**
**หาค่าระยะแฮมมิ่งระหว่าง 100 กับ 111 ได้ 2 หาค่าเมทริกซ์สะสมได้ว่า 3+2=5 ถูกเลือกเป็นเส้นทางที่เหลือรอด**

**ที่สถานะ 1 เส้นทางจากสถานะ 2 และสถานะ 3 ในระดับที่ 5**
**หาค่าระยะแฮมมิ่งระหว่าง 100 กับ 011 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 5+3=8**
**หาค่าระยะแฮมมิ่งระหว่าง 100 กับ 100 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 0+0=0 ถูกเลือกเป็นเส้นทางที่เหลือรอด**

**ที่สถานะ 2 เส้นทางจากสถานะ 0 และสถานะ 1 ในระดับที่ 5**
**หาค่าระยะแฮมมิ่งระหว่าง 100 กับ 111 ได้ 2 หาค่าเมทริกซ์สะสมได้ว่า 4+2=6**
**หาค่าระยะแฮมมิ่งระหว่าง 100 กับ 000 ได้ 1 หาค่าเมทริกซ์สะสมได้ว่า 3+1=4 ถูกเลือกเป็นเส้นทางที่เหลือรอด**

**ที่สถานะ 3 เส้นทางจากสถานะ 2 และสถานะ 3 ในระดับที่ 5**
**หาค่าระยะแฮมมิ่งระหว่าง 100 กับ 100 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 5+0=5**
**หาค่าระยะแฮมมิ่งระหว่าง 100 กับ 011 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 0+3=3 ถูกเลือกเป็นเส้นทางที่เหลือรอด**

**ได้ค่าตัวแปรระดับ l = 6 และได้ค่าเมทริกซ์สะสมใหม่ดังรูปที่ 4.9**




**รูปที่ 4.9** การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 5

รูปที่ 4.9 จะเห็นได้ว่าเมื่อผ่านกระบวนการไปอีก 1 ระดับจากระดับที่ 5 ไปยังระดับที่ 6 แล้ว จะได้ค่าระยะบนแต่ละเส้นทาง และค่าเมทริกซ์สะสมใหม่ที่สถานะ 0 คือ 5 สถานะ 1 คือ 0 สถานะ 2 คือ 4 และสถานะ 3 คือ 3

**ในระดับที่ 7** หาค่าระยะแฮมมิ่งของชุดคำรหัสที่ 7 กับข้อมูลในแต่ละเส้นทาง

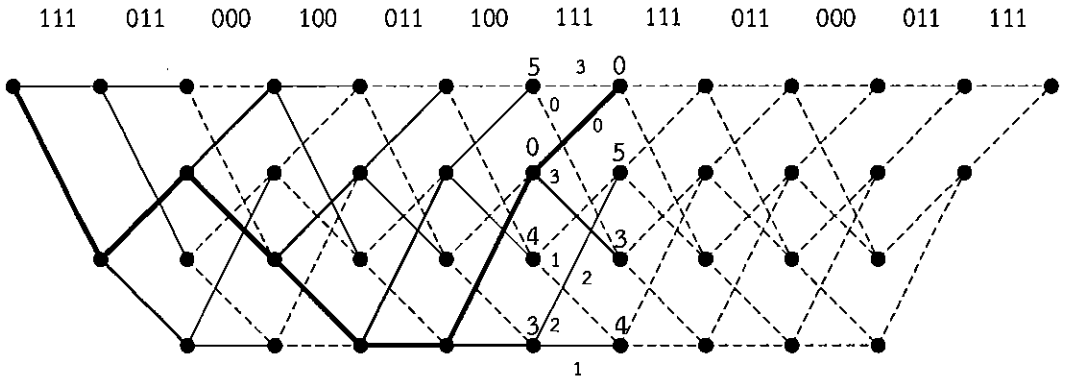
ที่สถานะ 0 เส้นทางจากสถานะ 0 และสถานะ 1 ในระดับที่ 6
หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 000 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 5+3=8
หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 111 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 0+0=0 ถูกเลือกเป็นเส้นทางที่เหลือรอด

ที่สถานะ 1 เส้นทางจากสถานะ 2 และสถานะ 3 ในระดับที่ 6
หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 011 ได้ 1 หาค่าเมทริกซ์สะสมได้ว่า 4+1=5
หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 100 ได้ 2 หาค่าเมทริกซ์สะสมได้ว่า 3+2=5 ถูกเลือกเป็นเส้นทางที่เหลือรอด

ที่สถานะ 2 เส้นทางจากสถานะ 0 และสถานะ 1 ในระดับที่ 6
หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 111 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 5+0=5
หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 000 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 0+3=3 ถูกเลือกเป็นเส้นทางที่เหลือรอด

ที่สถานะ 3 เส้นทางจากสถานะ 2 และสถานะ 3 ในระดับที่ 6
หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 100 ได้ 2 หาค่าเมทริกซ์สะสมได้ว่า 4+2=6
หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 011 ได้ 1 หาค่าเมทริกซ์สะสมได้ว่า 3+1=4 ถูกเลือกเป็นเส้นทางที่เหลือรอด

ได้ค่าตัวแปรระดับ l = 7 และได้ค่าเมทริกซ์สะสมใหม่ดังรูปที่ 4.10




**รูปที่ 4.10** การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 6

รูปที่ 4.10 จะเห็นได้ว่าเมื่อผ่านกระบวนการไปอีก 1 ระดับจากระดับที่ 6 ไปยังระดับที่ 7 แล้ว จะได้ค่าระยะบนแต่ละเส้นทาง และค่าเมทริกซ์สะสมใหม่ที่สถานะ 0 คือ 0 สถานะ 1 คือ 5 สถานะ 2 คือ 3 และสถานะ 3 คือ 4

**ในระดับที่ 8** หาค่าระยะแฮมมิ่งของชุดคำรหัสที่ 8 กับข้อมูลในแต่ละเส้นทาง

ที่สถานะ 0 เส้นทางจากสถานะ 0 และสถานะ 1 ในระดับที่ 7

หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 000 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 0+3=3 ถูกเลือกเป็นเส้นทางที่เหลือรอด

หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 111 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 5+0=5

ที่สถานะ 1 เส้นทางจากสถานะ 2 และสถานะ 3 ในระดับที่ 7

หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 011 ได้ 1 หาค่าเมทริกซ์สะสมได้ว่า 3+1=4 ถูกเลือกเป็นเส้นทางที่เหลือรอด

หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 100 ได้ 2 หาค่าเมทริกซ์สะสมได้ว่า 4+2=6

ที่สถานะ 2 เส้นทางจากสถานะ 0 และสถานะ 1 ในระดับที่ 7

หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 111 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 0+0=0 ถูกเลือกเป็นเส้นทางที่เหลือรอด

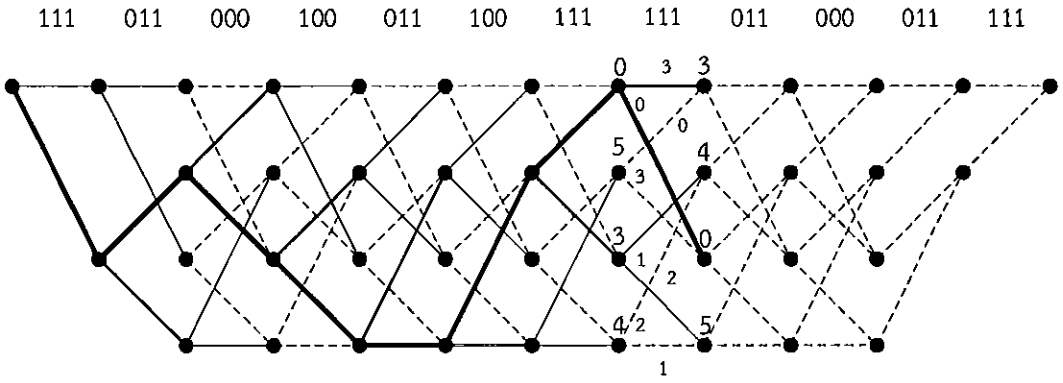
หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 000 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 5+3=8

ที่สถานะ 3 เส้นทางจากสถานะ 2 และสถานะ 3 ในระดับที่ 7

หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 100 ได้ 2 หาค่าเมทริกซ์สะสมได้ว่า 3+2=5 ถูกเลือกเป็นเส้นทางที่เหลือรอด

หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 011 ได้ 1 หาค่าเมทริกซ์สะสมได้ว่า 4+1=5

ได้ค่าตัวแปรระดับ l = 8 และได้ค่าเมทริกซ์สะสมใหม่ดังรูปที่ 4.11




**รูปที่ 4.11** การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 7

รูปที่ 4.11 จะเห็นได้ว่าเมื่อผ่านกระบวนการไปอีก 1 ระดับจากระดับที่ 7 ไปยังระดับที่ 8 แล้ว จะได้ค่าระยะบนแต่ละเส้นทาง และค่าเมทริกซ์สะสมใหม่ที่สถานะ 0 คือ 3 สถานะ 1 คือ 4 สถานะ 2 คือ 0 และสถานะ 3 คือ 5

**ในระดับที่ 9** หาค่าระยะแฮมมิ่งของชุดคำรหัสที่ 9 กับข้อมูลในแต่ละเส้นทาง

ที่สถานะ 0 เส้นทางจากสถานะ 0 และสถานะ 1 ในระดับที่ 8

หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 000 ได้ 2 หาค่าเมทริกซ์สะสมได้ว่า 3+2=5 ถูกเลือกเป็นเส้นทางที่เหลือรอด

หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 111 ได้ 1 หาค่าเมทริกซ์สะสมได้ว่า 4+1=5

ที่สถานะ 1 เส้นทางจากสถานะ 2 และสถานะ 3 ในระดับที่ 8

หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 011 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 0+0=0 ถูกเลือกเป็นเส้นทางที่เหลือรอด

หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 100 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 5+3=8

ที่สถานะ 2 เส้นทางจากสถานะ 0 และสถานะ 1 ในระดับที่ 8

หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 111 ได้ 1 หาค่าเมทริกซ์สะสมได้ว่า 3+1=4 ถูกเลือกเป็นเส้นทางที่เหลือรอด

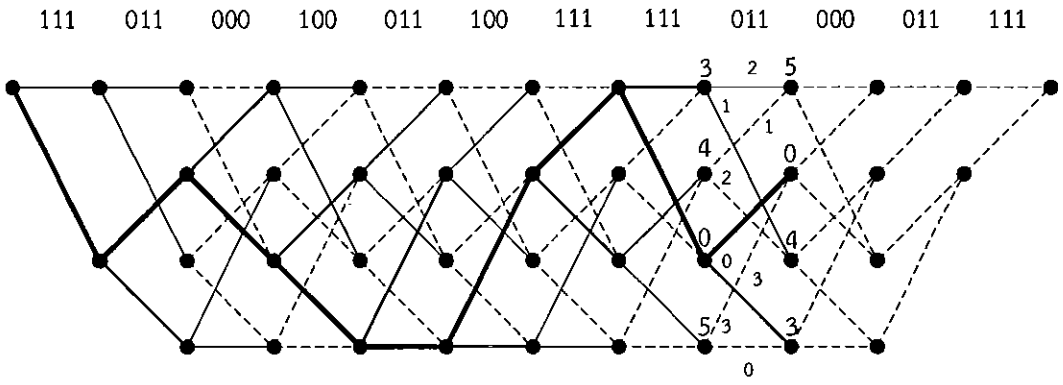
หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 000 ได้ 2 หาค่าเมทริกซ์สะสมได้ว่า 4+2=6

ที่สถานะ 3 เส้นทางจากสถานะ 2 และสถานะ 3 ในระดับที่ 8

หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 100 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 0+3=3 ถูกเลือกเป็นเส้นทางที่เหลือรอด

หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 011 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 5+0=5

ได้ค่าตัวแปรระดับ l = 9 และได้ค่าเมทริกซ์สะสมใหม่ดังรูปที่ 4.12




**รูปที่ 4.12** การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 8

รูปที่ 4.12 จะเห็นได้ว่าเมื่อผ่านกระบวนการไปอีก 1 ระดับจากระดับที่ 8 ไปยังระดับที่ 9 แล้ว จะได้ค่าระยะบนแต่ละเส้นทาง และค่าเมทริกซ์สะสมใหม่ที่สถานะ 0 คือ 5 สถานะ 1 คือ 0 สถานะ 2 คือ 4 และสถานะ 3 คือ 3

**ในระดับที่ 10** หาค่าระยะแฮมมิ่งของชุดคำรหัสที่ 10 กับข้อมูลในแต่ละเส้นทาง

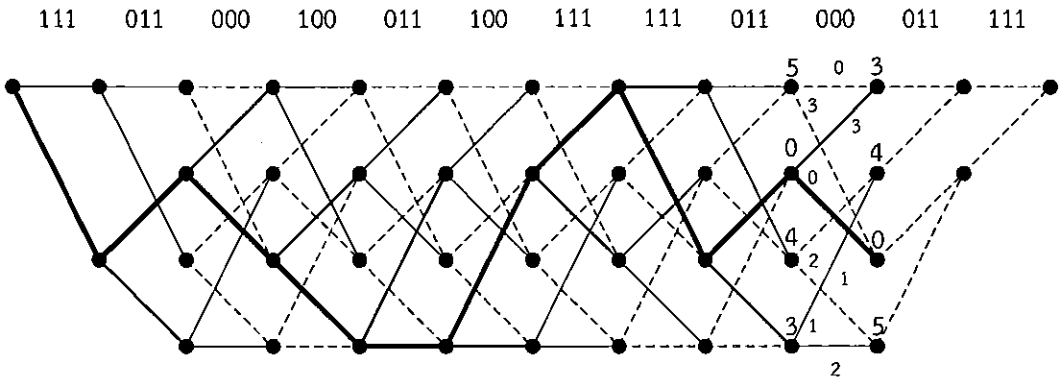
ที่สถานะ 0 เส้นทางจากสถานะ 0 และสถานะ 1 ในระดับที่ 9
หาค่าระยะแฮมมิ่งระหว่าง 000 กับ 000 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 5+0=5
หาค่าระยะแฮมมิ่งระหว่าง 000 กับ 111 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 0+3=3 ถูกเลือกเป็นเส้นทางที่เหลือรอด

ที่สถานะ 1 เส้นทางจากสถานะ 2 และสถานะ 3 ในระดับที่ 9
หาค่าระยะแฮมมิ่งระหว่าง 000 กับ 011 ได้ 2 หาค่าเมทริกซ์สะสมได้ว่า 4+2=6
หาค่าระยะแฮมมิ่งระหว่าง 000 กับ 100 ได้ 1 หาค่าเมทริกซ์สะสมได้ว่า 3+1=4 ถูกเลือกเป็นเส้นทางที่เหลือรอด

ที่สถานะ 2 เส้นทางจากสถานะ 0 และสถานะ 1 ในระดับที่ 9
หาค่าระยะแฮมมิ่งระหว่าง 000 กับ 111 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 5+3=8
หาค่าระยะแฮมมิ่งระหว่าง 000 กับ 000 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 0+0=0 ถูกเลือกเป็นเส้นทางที่เหลือรอด

ที่สถานะ 3 เส้นทางจากสถานะ 2 และสถานะ 3 ในระดับที่ 9
หาค่าระยะแฮมมิ่งระหว่าง 000 กับ 100 ได้ 1 หาค่าเมทริกซ์สะสมได้ว่า 4+1=5
หาค่าระยะแฮมมิ่งระหว่าง 000 กับ 011 ได้ 2 หาค่าเมทริกซ์สะสมได้ว่า 3+2=5 ถูกเลือกเป็นเส้นทางที่เหลือรอด

ได้ค่าตัวแปรระดับ l = 10 และได้ค่าเมทริกซ์สะสมใหม่ดังรูปที่ 4.13




**รูปที่ 4.13** การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 9

รูปที่ 4.13 จะเห็นได้ว่าเมื่อผ่านกระบวนการไปอีก 1 ระดับจากระดับที่ 9 ไปยังระดับที่ 10 แล้ว จะได้ค่าระยะบนแต่ละเส้นทาง และค่าเมทริกซ์สะสมใหม่ที่สถานะ 0 คือ 3 สถานะ 1 คือ 4 สถานะ 2 คือ 0 และสถานะ 3 คือ 5

**ในระดับที่ 11** หาค่าระยะแฮมมิ่งของชุดคำรหัสที่ 11 กับข้อมูลในแต่ละเส้นทาง

ที่สถานะ 0 เส้นทางจากสถานะ 0 และสถานะ 1 ในระดับที่ 10

หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 000 ได้ 2 หาค่าเมทริกซ์สะสมได้ว่า 3+2=5 ถูกเลือกเป็นเส้นทางที่เหลือรอด

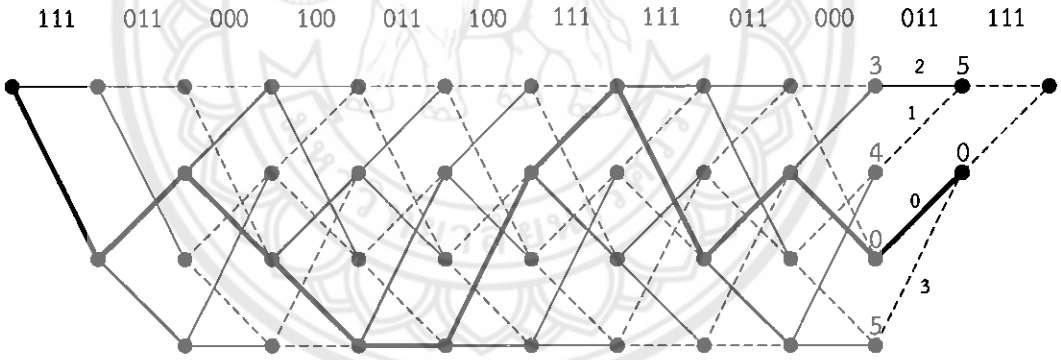
หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 111 ได้ 1 หาค่าเมทริกซ์สะสมได้ว่า 4+1=5

ที่สถานะ 1 เส้นทางจากสถานะ 2 และสถานะ 3 ในระดับที่ 10

หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 011 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 0+0=0 ถูกเลือกเป็นเส้นทางที่เหลือรอด

หาค่าระยะแฮมมิ่งระหว่าง 011 กับ 100 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 5+3=8

ได้ค่าตัวแปรระดับ l = 11 และได้ค่าเมทริกซ์สะสมใหม่ดังรูปที่ 4.14




**รูปที่ 4.14** การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ l = 10

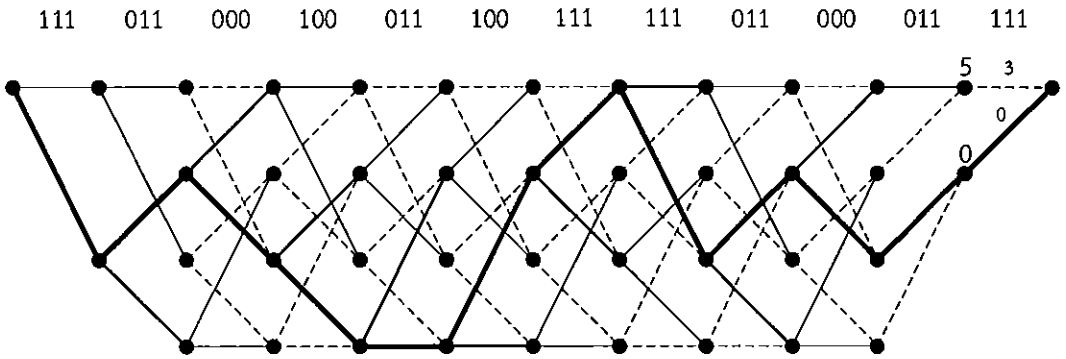
รูปที่ 4.14 จะเห็นได้ว่าเมื่อผ่านกระบวนการไปอีก 1 ระดับจากระดับที่ 10 ไปยังระดับที่ 11 แล้ว จะได้ค่าระยะบนแต่ละเส้นทาง และค่าเมทริกซ์สะสมใหม่ที่สถานะ 0 คือ 5 และสถานะ 1 คือ 0

**ในระดับที่ 12** หาค่าระยะแฮมมิ่งของชุดคำรหัสที่ 12 กับข้อมูลในแต่ละเส้นทาง

ที่สถานะ 0 เส้นทางจากสถานะ 0 และสถานะ 1 ในระดับที่ 11

หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 000 ได้ 3 หาค่าเมทริกซ์สะสมได้ว่า 5+3=8

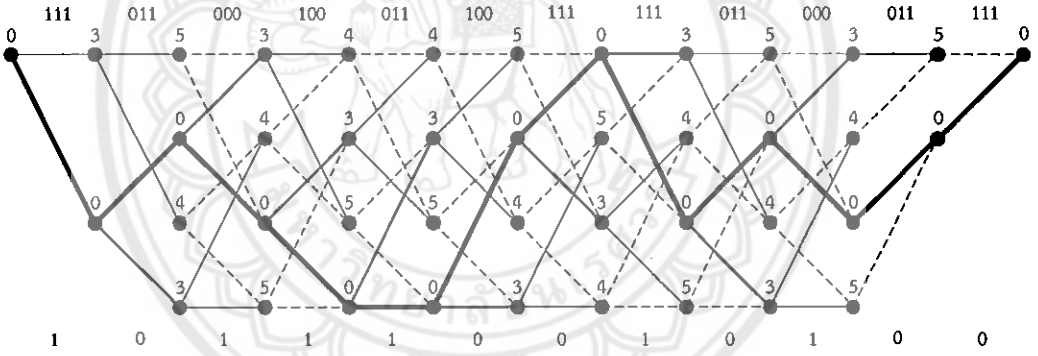
หาค่าระยะแฮมมิ่งระหว่าง 111 กับ 111 ได้ 0 หาค่าเมทริกซ์สะสมได้ว่า 0+0=0 ถูกเลือกเป็นเส้นทางที่เหลือรอด ดังรูปที่ 4.15




**รูปที่ 4.15** การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] เมื่อ I = 11

รูปที่ 4.15 จะเห็นได้ว่าเมื่อผ่านกระบวนการไปถึงระดับสุดท้าย คือจากระดับที่ 11 ไปยังระดับที่ 12 แล้ว จะได้ค่าระยะบนแต่ละเส้นทาง และค่าเมทริกซ์สะสมใหม่ที่สถานะ 0 คือ 0

จากการกระทำข้างต้นที่ได้กล่าวมาแล้ว จะทำให้ได้ผลลัพธ์ของเส้นทาง และข้อมูลก่อนเข้ารหัสคอนโวลูชัน ซึ่งสามารถแสดงผ่านแผนภาพเทรลลิสได้ดังรูปที่ 4.16



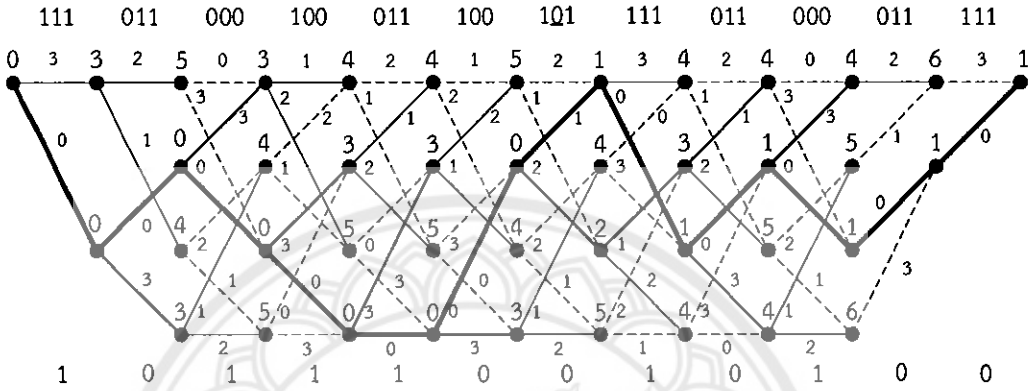

**รูปที่ 4.16** แผนภาพเทรลลิสการถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2]

รูปที่ 4.16 รูปแสดงแผนภาพเทรลลิสแสดงการถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,1,2] โดยที่บนจุดจะถูกกำหนดค่าไว้ด้วยค่าเมทริกซ์สะสมของแต่ละสถานะในทุกระดับ ซึ่งจะได้ผลลัพธ์เป็นเส้นทางบนแผนภาพเทรลลิส นำมาใช้ในการถอดคำรหัสได้ว่า 101110010100 ซึ่งเมื่อตัดบิตหางออกจะได้ว่าข้อมูลก่อนเข้ารหัสคอนโวลูชันคือ 1011100101

สำหรับระบบ [3,1,2] เมื่ออินพุตคือ 1011100101 จะได้คำรหัสว่า 111 011 000 100 011 100 111 111 011 000 011 111 แต่เมื่อส่งผ่านช่องส่งสัญญาณผลกระทบจากสัญญาณรบกวนจะก่อให้เกิดความผิดพลาดขึ้น ดังนั้นเพื่อจำลองผลกระทบดังกล่าวจะลองทำการเปลี่ยนค่าของคำรหัสด้วยความผิดพลาดตามกรณีต่างๆ ดังต่อไปนี้

**2.2 กรณีบิตผิดพลาดจำนวน 1 บิต**

กรณีนี้กำหนดให้บิตที่ 20 จาก 1 เป็น 0 จำนวน 1 บิตเป็นบิตผิดพลาด จะทำให้ได้สัญญาณที่ฝั่งตัวรับคือ 111 011 000 100 011 100 101 111 011 000 011 111 เมื่อนำมาทำการคำนวณเพื่อหาอินพุตด้วยขั้นตอนวิธีของวิเทอร์บีจะได้ดังรูปที่ 4.17

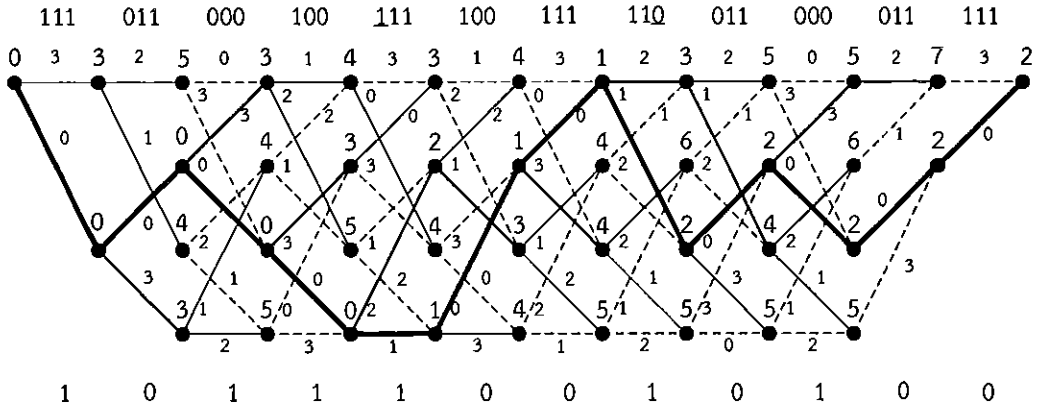



**รูปที่ 4.17** แผนภาพเทรลลิสแสดงการถอดคำรหัสในกรณีที่ 2.2

รูปที่ 4.17 เป็นแผนภาพเทรลลิสที่แสดงถึงการถอดคำรหัสคอนโวลูชันในกรณีที่มีบิตผิดพลาดจำนวน 1 บิต โดยจะแสดงค่าระยะแฮมมิ่งในแต่ละเส้นทาง และค่าเมทริกซ์สะสมในแต่ละจุดของสถานะ ซึ่งจะได้ผลลัพธ์คือ 1011100101 ซึ่งเมื่อนำมาเปรียบเทียบกับอินพุตจริง ข้างต้นจะได้จำนวนบิตที่ผิดพลาดจำนวน 0 บิตซึ่งหมายความว่ายังสามารถแก้ไขความผิดพลาดที่เกิดขึ้นได้ โดยที่ 00 สองบิตสุดท้ายคือบิตหาง

**2.3 กรณีบิตผิดพลาดจำนวน 2 บิตที่ไม่ติดกัน**

กรณีนี้กำหนดให้บิตที่ 13 จาก 0 เป็น 1 และ บิตที่ 24 จาก 1 เป็น 0 จำนวน 2 บิตเป็นบิตผิดพลาด จะทำให้ได้สัญญาณที่ฝั่งตัวรับคือ 111 011 000 100 111 100 111 110 011 000 011 111 เมื่อนำมาทำการคำนวณเพื่อหาอินพุตด้วยขั้นตอนวิธีของวิเทอร์บีจะได้ดังรูปที่ 4.18

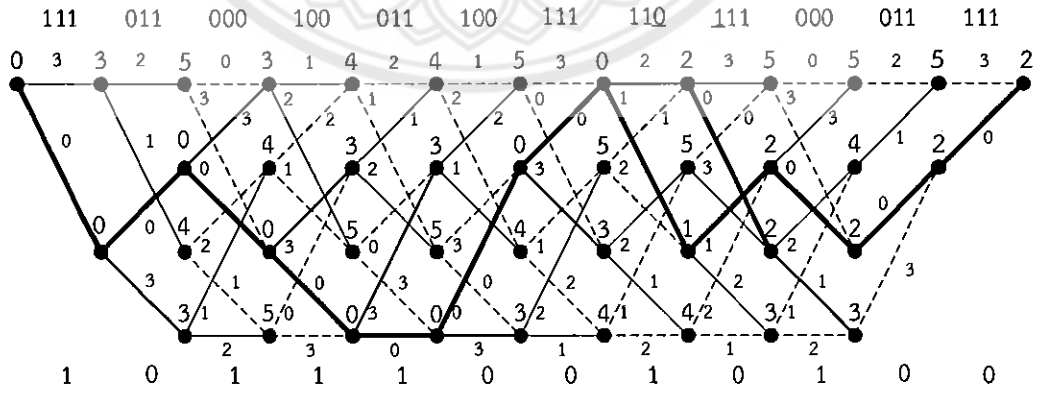



**รูปที่ 4.18** แผนภาพเทรลลิสแสดงการถอดคำรหัสในกรณีที่ 2.3

รูปที่ 4.18 เป็นแผนภาพเทรลลิสที่แสดงถึงการถอดคำรหัสคอนโวลูชันในกรณีที่มีบิตผิดพลาดจำนวน 2 บิตที่ไม่ติดกัน โดยจะแสดงค่าระยะแฮมมิ่งในแต่ละเส้นทาง และค่าเมทริกซ์สะสมในแต่ละจุดของสถานะ ซึ่งจะได้ผลลัพธ์คือ 1011100101 ซึ่งเมื่อนำมาเปรียบเทียบกับอินพุตจริง ข้างต้นจะได้จำนวนบิตที่ผิดพลาดจำนวน 0 บิตซึ่งหมายความว่ายังสามารถแก้ไขความผิดพลาดที่เกิดขึ้นได้ โดยที่ 00 สองบิตสุดท้ายคือบิตหาง

### 2.4 กรณีบิตผิดพลาดจำนวน 2 บิตที่ติดกัน และอยู่ต่างคำรหัส

กรณีนี้กำหนดให้บิตที่ 24 จาก 1 เป็น 0 และ บิตที่ 25 จาก 0 เป็น 1 จำนวน 2 บิตเป็นบิตผิดพลาด จะทำให้ได้สัญญาณที่ฝั่งตัวรับคือ 111 011 000 100 011 100 111 110 111 000 011 111 เมื่อนำมาทำการคำนวณเพื่อหาอินพุตด้วยขั้นตอนวิธีของวิเทอร์บีจะได้ดังรูปที่ 4.19



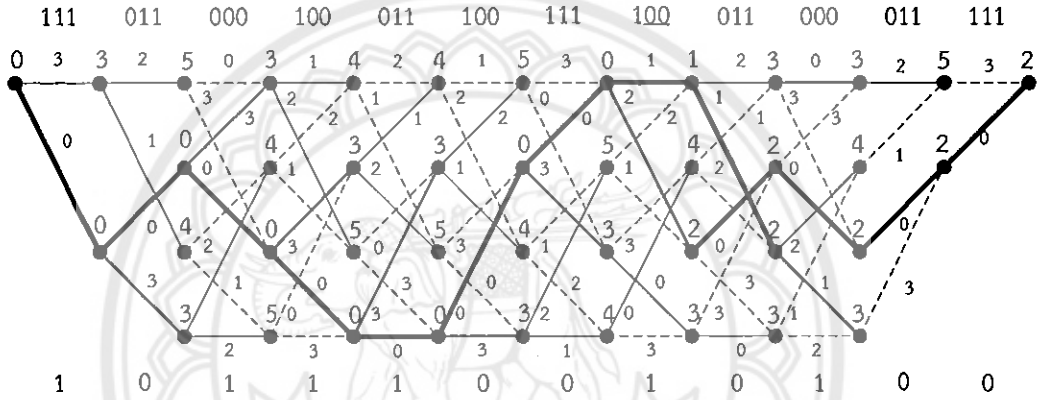

**รูปที่ 4.19** แผนภาพเทรลลิสแสดงการถอดคำรหัสในกรณีที่ 2.4

รูปที่ 4.19 เป็นแผนภาพเทรลลิสที่แสดงถึงการถอดคำรหัสคอนโวลูชันในกรณีที่มีบิตผิดพลาดจำนวน 2 บิตที่ติดกัน และอยู่ต่างคำรหัส โดยจะแสดงค่าระยะแฮมมิ่งในแต่ละเส้นทาง และ

ค่าเมทริกซ์สะสมในแต่ละจุดของสถานะ ซึ่งจะได้ผลลัพธ์คือ 1011100101 ซึ่งเมื่อนำมาเปรียบเทียบกับอินพุตจริง ข้างต้นจะได้จำนวนบิตที่ผิดพลาดจำนวน 0 บิตซึ่งหมายความว่ายังสามารถแก้ไขความผิดพลาดที่เกิดขึ้นได้ โดยที่ 00 สองบิตสุดท้ายคือบิตหาง

### 2.5 กรณีบิตผิดพลาดจำนวน 2 บิตที่ติดกัน และอยู่ในคำรหัสเดียวกัน

กรณีนี้กำหนดให้บิตที่ 23 จาก 1 เป็น 0 และบิตที่ 24 จาก 1 เป็น 0 จำนวน 2 บิตเป็นบิตผิดพลาด จะทำให้ได้สัญญาณที่ฝั่งตัวรับคือ 111 011 000 100 011 100 111 100 011 000 011 111 เมื่อนำมาทำการคำนวณเพื่อหาอินพุตด้วยขั้นตอนวิธีของวิเทอร์บีจะได้ดังรูปที่ 4.20

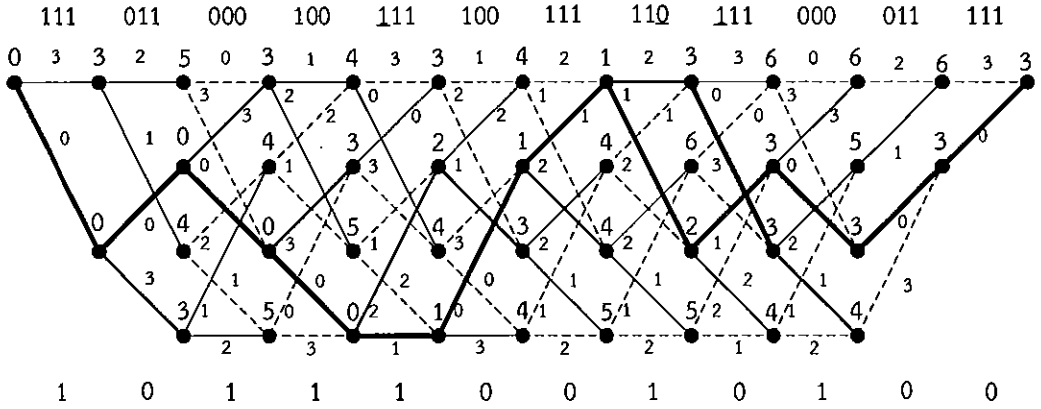



**รูปที่ 4.20** แผนภาพเทรลลิสแสดงการถอดคำรหัสในกรณีที่ 2.5

รูปที่ 4.20 เป็นแผนภาพเทรลลิสที่แสดงถึงการถอดคำรหัสคอนโวลูชันในกรณีที่มีบิตผิดพลาดจำนวน 2 บิตที่ติดกัน และอยู่ในคำรหัสเดียวกัน โดยจะแสดงค่าระยะแฮมมิ่งในแต่ละเส้นทาง และค่าเมทริกซ์สะสมในแต่ละจุดของสถานะ ซึ่งจะได้ผลลัพธ์คือ 1011100101 ซึ่งเมื่อนำมาเปรียบเทียบกับอินพุตจริง ข้างต้นจะได้จำนวนบิตที่ผิดพลาดจำนวน 0 บิตซึ่งหมายความว่ายังสามารถแก้ไขความผิดพลาดที่เกิดขึ้นได้ โดยที่ 00 สองบิตสุดท้ายคือบิตหาง

### 2.6 กรณีบิตผิดพลาดจำนวน 3 บิต โดยมี 2 บิตที่อยู่ติดกัน แต่อยู่ต่างคำรหัส

กรณีนี้กำหนดให้บิตที่ 13 จาก 0 เป็น 1 บิตที่ 24 จาก 1 เป็น 0 และ บิตที่ 25 จาก 0 เป็น 1 จำนวน 3 บิตเป็นบิตผิดพลาด จะทำให้ได้สัญญาณที่ฝั่งตัวรับคือ 111 011 000 100 111 100 111 110 111 000 011 111 เมื่อนำมาทำการคำนวณเพื่อหาอินพุตด้วยขั้นตอนวิธีของวิเทอร์บีจะได้ดังรูปที่ 4.21

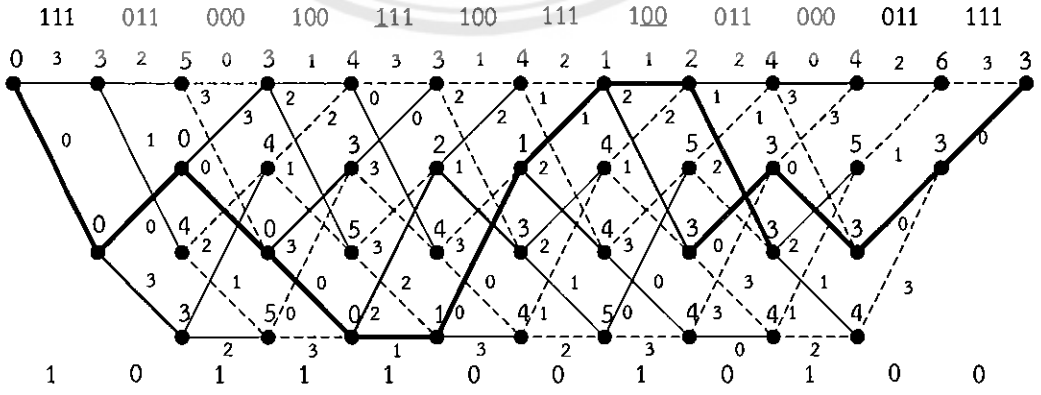



**รูปที่ 4.21** แผนภาพเทรลลิสแสดงการถอดคำรหัสในกรณีที่ 2.6

รูปที่ 4.21 เป็นแผนภาพเทรลลิสที่แสดงถึงการถอดคำรหัสคอนโวลูชันในกรณีที่มีบิตผิดพลาดจำนวน 3 บิต โดยมี 2 บิตที่อยู่ติดกัน แต่อยู่ต่างคำรหัส โดยจะแสดงค่าระยะแฮมมิ่งในแต่ละเส้นทาง และค่าเมทริกซ์สะสมในแต่ละจุดของสถานะ ซึ่งจะได้ผลลัพธ์คือ 1011100101 ซึ่งเมื่อนำมาเปรียบเทียบกับอินพุตจริง ข้างต้นจะได้จำนวนบิตที่ผิดพลาดจำนวน 0 บิตซึ่งหมายความว่ายังสามารถแก้ไขความผิดพลาดที่เกิดขึ้นได้ โดยที่ 00 สองบิตสุดท้ายคือบิตหาง

### 2.7 กรณีบิตผิดพลาดจำนวน 3 บิต โดยที่มี 2 บิตติดกัน และอยู่ในคำรหัสเดียวกัน

กรณีนี้กำหนดให้บิตที่ 13 จาก 0 เป็น 1 บิตที่ 23 จาก 1 เป็น 0 และ บิตที่ 24 จาก 1 เป็น 0 จำนวน 3 บิตเป็นบิตผิดพลาด จะทำให้ได้สัญญาณที่ฝั่งตัวรับคือ 111 011 000 100 111 100 111 100 011 000 011 111 เมื่อนำมาทำการคำนวณเพื่อหาอินพุตด้วยขั้นตอนวิธีของวิเทอร์บีจะได้ดังรูปที่ 4.22

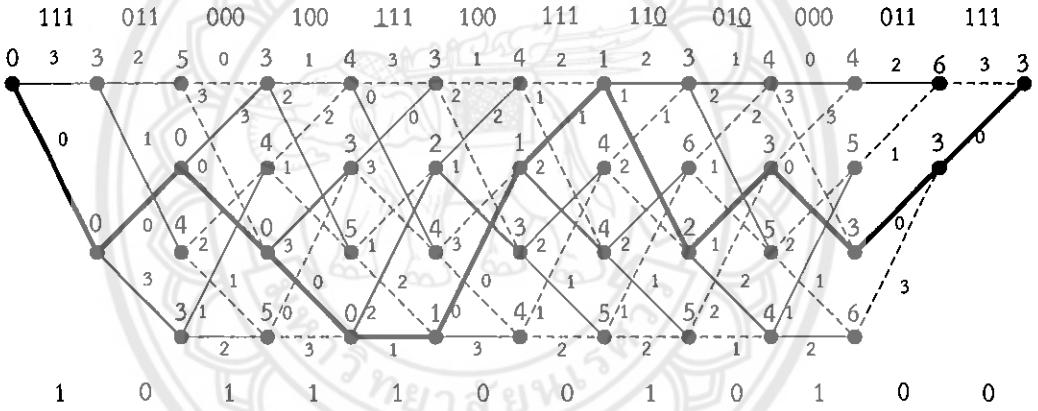



**รูปที่ 4.22** แผนภาพเทรลลิสแสดงการถอดคำรหัสในกรณีที่ 2.7

รูปที่ 4.22 เป็นแผนภาพเทรลลิสที่แสดงถึงการถอดคำรหัสคอนโวลูชันในกรณีที่มีบิตผิดพลาดจำนวน 3 บิต โดยที่มี 2 บิตติดกัน และอยู่ในคำรหัสเดียวกัน โดยจะแสดงค่าระยะแฮมมิ่งในแต่ละเส้นทาง และค่าเมทริกซ์สะสมในแต่ละจุดของสถานะ ซึ่งจะได้ผลลัพธ์คือ 1011100101 ซึ่งเมื่อนำมาเปรียบเทียบกับอินพุตจริง ข้างต้นจะได้จำนวนบิตที่ผิดพลาดจำนวน 0 บิตซึ่งหมายความว่ายังสามารถแก้ไขความผิดพลาดที่เกิดขึ้นได้ โดยที่ 00 สองบิตสุดท้ายคือบิตหาง

### 2.8 กรณีบิตผิดพลาดจำนวน 3 บิตไม่ติดกัน

กรณีนี้กำหนดให้บิตที่ 13 จาก 0 เป็น 1 บิตที่ 24 จาก 1 เป็น 0 และ บิตที่ 27 จาก 1 เป็น 0 จำนวน 3 บิตเป็นบิตผิดพลาด จะทำให้ได้สัญญาณที่ฝั่งตัวรับคือ 111 011 000 100 111 100 111 110 010 000 011 111 เมื่อนำมาทำการคำนวณเพื่อหาอินพุตด้วยขั้นตอนวิธีของวิเทอร์บีจะได้ดังรูปที่ 4.23



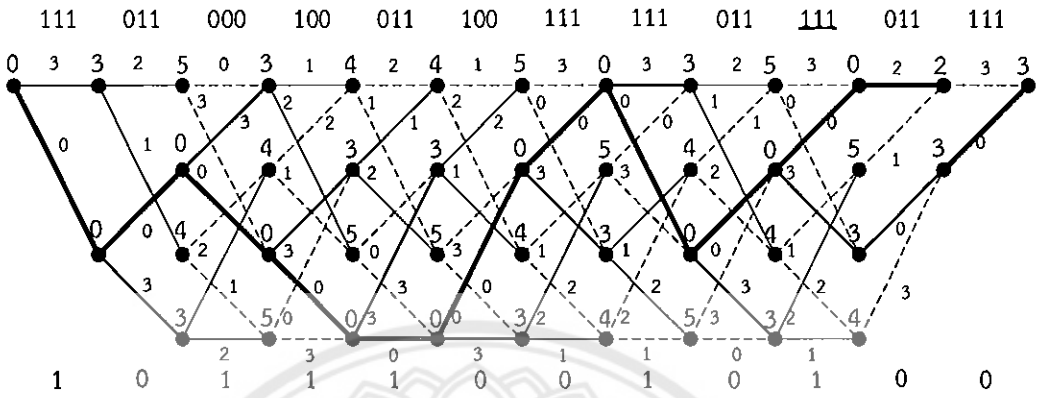

**รูปที่ 4.23** แผนภาพเทรลลิสแสดงการถอดคำรหัสในกรณีที่ 2.8

รูปที่ 4.23 เป็นแผนภาพเทรลลิสที่แสดงถึงการถอดคำรหัสคอนโวลูชันในกรณีที่มีบิตผิดพลาดจำนวน 3 บิตไม่ติดกัน โดยจะแสดงค่าระยะแฮมมิ่งในแต่ละเส้นทาง และค่าเมทริกซ์สะสมในแต่ละจุดของสถานะ ซึ่งจะได้ผลลัพธ์คือ 1011100101 ซึ่งเมื่อนำมาเปรียบเทียบกับอินพุตจริง ข้างต้นจะได้จำนวนบิตที่ผิดพลาดจำนวน 0 บิตซึ่งหมายความว่ายังสามารถแก้ไขความผิดพลาดที่เกิดขึ้นได้ โดยที่ 00 สองบิตสุดท้ายคือบิตหาง

### 2.9 กรณีบิตผิดพลาดจำนวน 3 บิตติดกัน

กรณีนี้กำหนดให้บิตที่ 28 จาก 0 เป็น 1 บิตที่ 29 จาก 0 เป็น 1 และ บิตที่ 30 จาก 0 เป็น 1 จำนวน 3 บิตเป็นบิตผิดพลาด จะทำให้ได้สัญญาณที่ฝั่งตัวรับคือ 111 011 000 100 011 100 111 111

011 111 011 111 เมื่อนำมาทำการคำนวณเพื่อหาอินพุตด้วยขั้นตอนวิธีของวิเทอร์บีจะได้ดังรูปที่ 4.24




**รูปที่ 4.24** แผนภาพเทรลลิสแสดงการถอดคำรหัสในกรณีที่ 2.9

รูปที่ 4.24 เป็นแผนภาพเทรลลิสที่แสดงถึงการถอดคำรหัสคอนโวลูชันในกรณีที่มีบิตผิดพลาดจำนวน 3 บิตติดกัน โดยจะแสดงค่าระยะแฮมมิ่งในแต่ละเส้นทาง และค่าเมทริกซ์สะสมในแต่ละจุดของสถานะ ซึ่งจะได้ผลลัพธ์คือ 1011100101 ซึ่งเมื่อนำมาเปรียบเทียบกับอินพุตจริงข้างต้นจะได้จำนวนบิตที่ผิดพลาดจำนวน 0 บิตซึ่งหมายความว่ายังสามารถแก้ไขความผิดพลาดที่เกิดขึ้นได้ โดยที่ 00 สองบิตสุดท้ายคือบิตหาง

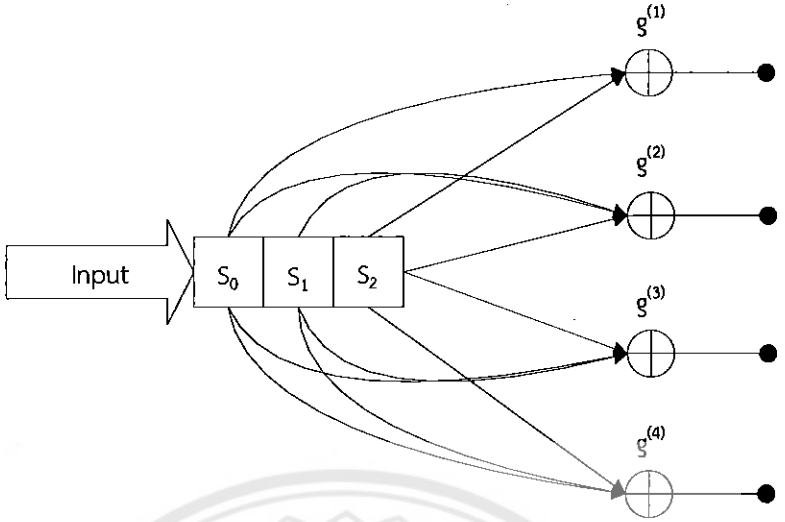
### 4.3.2 กรณีศึกษาของระบบ [4,1,2]

1. การเข้ารหัสคอนโวลูชัน

ระบบ [4,1,2] มีความหมายว่า 4 คือมีจำนวนบิตของคำรหัส 4 บิตต่อหนึ่งหน่วยเวลา
1 คือมีจำนวนบิตข้อมูลที่เข้าสู่ระบบ 1 บิตต่อหนึ่งหน่วยเวลา
2 คือมีจำนวนชิฟรีจิสเตอร์ที่เป็นหน่วยความจำ 2 ชิฟ

ซึ่งจะได้เมทริกซ์ก่อกำเนิด [5,7,7,7] (ฐาน 8) แปลงได้เป็น [101,111,111,111] (ฐาน 2)
โดยกำหนดให้ $g^{(1)}$ คือ 101, $g^{(2)}$ คือ 111, $g^{(3)}$ คือ 111, $g^{(4)}$ คือ 111
และกำหนดให้ $S_0$ คืออินพุต
$S_1$ คือชิฟรีจิสเตอร์ชิฟที่ 1
$S_2$ คือชิฟรีจิสเตอร์ชิฟที่ 2

จะได้แผนภาพจำลองระบบ [4,1,2] ดังรูปที่ 4.25




**รูปที่ 4.25** แผนภาพจำลองระบบ [4,1,2]

รูปที่ 4.25 คือรูปของแผนภาพการจำลองระบบ [4,1,2] จะเห็นได้ว่าระบบบนี้อินพุตจะเข้าครั้งละ 1 บิตที่ $S_0$ และได้คำรหัสจำนวน 4 บิต และการเชื่อมต่อกันระหว่างชิฟรีจิสเตอร์กับตัวบวกแบบมอดูโลทูจะได้ค่าของฟังก์ชันก่อกำเนิด เช่น $g^{(1)}$ ซึ่งมีการเชื่อมต่อกับ $S_0$ และ $S_2$ จะมีค่า 101 เป็นต้น

จากนั้นทำการหาอินพุต สถานะของหน่วยความจำ สถานะที่เปลี่ยนไปเมื่อมีอินพุตเข้าสู่วงจร คำรหัสที่ได้ ตามวิธีที่ได้กล่าวมาแล้วในกรณีศึกษาของระบบ [3,1,2] จะทำให้สามารถสร้างตารางแสดงความสัมพันธ์ของระบบ [4,1,2] ได้ดังตารางที่ 4.5

**ตารางที่ 4.5** ตารางแสดงความสัมพันธ์ของระบบ [4,1,2]

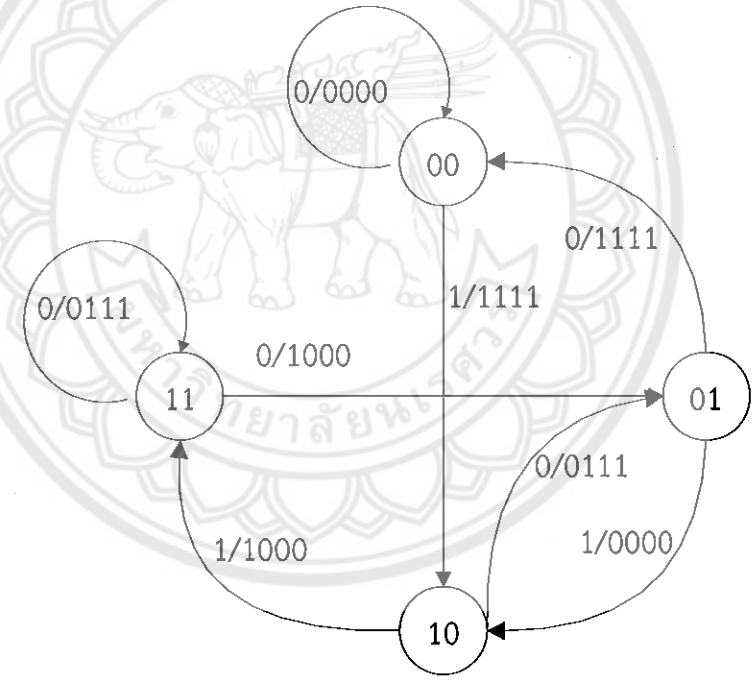
| อินพุต($S_0$) | สถานะเดิม($S_1S_2$) | คำรหัส | สถานะถัดไป($S_0S_1$) |
|---|---|---|---|
| 0 | 00 | 0000 | 00 |
| 0 | 01 | 1111 | 00 |
| 0 | 10 | 0111 | 01 |
| 0 | 11 | 1000 | 01 |
| 1 | 00 | 1111 | 10 |
| 1 | 01 | 0000 | 10 |
| 1 | 10 | 1000 | 11 |

**ตารางที่ 4.5 (ต่อ)** ตารางแสดงความสัมพันธ์ของระบบ [4,1,2]

| อินพุต($S_0$) | สถานะเดิม($S_1S_2$) | คำรหัส | สถานะถัดไป($S_0S_1$) |
|---|---|---|---|
| 1 | 11 | 0111 | 11 |

ตารางที่ 4.5 คือตารางแสดงค่าความสัมพันธ์ของระบบ [4,1,2] ซึ่งได้จากการคำนวณ อธิบายความสัมพันธ์ได้ว่า เช่นให้สถานะเดิมคือ 00 ในกรณีที่อินพุตคือ 0 เข้าสู่ระบบจะได้ผลลัพธ์คำรหัสคือ 0000 สถานะถัดไปคือ 00 ในกรณีที่อินพุตคือ 1 เข้าสู่ระบบจะได้ผลลัพธ์คำรหัสคือ 1111 สถานะถัดไปคือ 10 เป็นต้น

จากตารางที่ 4.5 สามารถนำมาสร้างเป็นแผนภาพสถานะได้ดังรูปที่ 4.26




**รูปที่ 4.26** แผนภาพสถานะของระบบ [4,1,2]

รูปที่ 4.26 คือแผนภาพสถานะของระบบ [4,1,2] สามารถอธิบายได้เช่น เมื่อที่สถานะเดิมคือ 00 ในกรณีที่อินพุตคือ 0 สถานะถัดไปคือ 00 และได้คำรหัสคือ 0000 ในกรณีที่อินพุตคือ 1 สถานะถัดไปคือ 10 และได้คำรหัสคือ 1111 เป็นต้น

ตัวอย่างแสดงการเข้ารหัสของระบบ [4,1,2] ที่มีเมทริกซ์ก่อกำเนิดคือ [5,7,7,7]

กำหนดให้อินพุตคือ 1011100101 และสถานะเริ่มต้นคือ 00 เมื่ออินพุตเข้าสู่ระบบจะทำให้ได้การเปลี่ยนสถานะ และคำรหัสดังตารางที่ 4.6 (โดยในขั้นตอนสุดท้ายจะทำการเติมบิตหางเพื่อปรับให้สถานะของหน่วยความจำกลับไปเป็น 00)

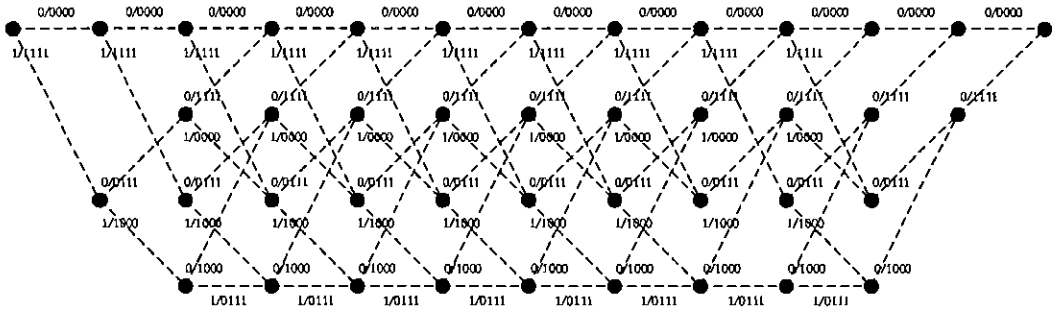
**ตารางที่ 4.6** ตารางแสดง อินพุต การเปลี่ยนสถานะ และคำรหัสจากตัวอย่างอินพุต

| อินพุต | การเปลี่ยนสถานะ | คำรหัส |
|---|---|---|
| 1 | 10 | 1111 |
| 0 | 01 | 0111 |
| 1 | 10 | 0000 |
| 1 | 11 | 1000 |
| 1 | 11 | 0111 |
| 0 | 01 | 1000 |
| 0 | 00 | 1111 |
| 1 | 10 | 1111 |
| 0 | 01 | 0111 |
| 1 | 10 | 0000 |
| 0 | 01 | 0111 |
| 0 | 00 | 1111 |

ตารางที่ 4.6 จากการนำอินพุตคือ 1011100101 มาเข้ารหัสคอนโวลูชันจะได้คำรหัสคือ 1111 0111 0000 1000 0111 1000 1111 1111 0111 0000 0111 1111

**2. การถอดคำรหัสคอนโวลูชัน กรณีไม่มีบิตผิดพลาด**

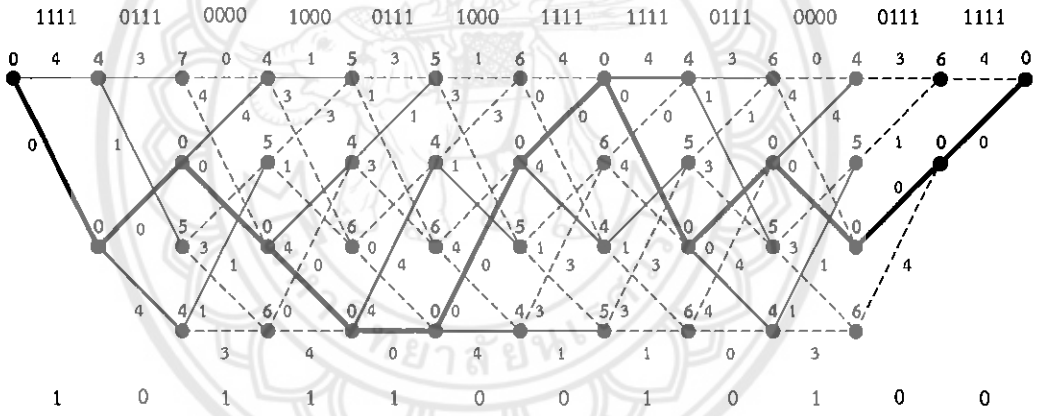
เมื่อนำคำรหัส 111101110000100001111000111111110111000001111111 ดังตารางที่ 4.6 มาถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บี โดยใช้ตารางที่ 4.5 ในการสร้างแผนภาพเทรลลิสของระบบ [4,1,2] ได้ดังรูปที่ 4.27



**รูปที่ 4.27** แผนภาพเทรลลิสของระบบ [4,1,2]

รูปที่ 4.27 คือแผนภาพเทรลลิสของระบบ [4,1,2] จำนวน 13 ระดับ โดยเริ่มต้นจากสถานะ 0 ณ ระดับที่ 0 และระดับที่ 10 ขึ้นไปจะวาดเฉพาะเส้นทางที่เข้าหาสถานะของวงจรเป็น 0 ทั้งหมด

จากนั้นทำการถอดคำรหัสคอนโวลูชันตามวิธีที่ได้กล่าวมาแล้วในกรณีศึกษาของระบบ [3,1,2] จะได้ผลลัพธ์จากการถอดคำรหัสดังรูปที่ 4.28




**รูปที่ 4.28** แผนภาพเทรลลิสแสดงการถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บี ของระบบ [4,1,2]

รูปที่ 4.28 รูปแสดงแผนภาพเทรลลิสแสดงการถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [4,1,2] โดยที่บนจุดจะถูกกำหนดค่าไว้ด้วยค่าเมทริกซ์สะสมของแต่ละสถานะในทุกระดับ ซึ่งจะได้ผลลัพธ์เป็นเส้นทางบนแผนภาพเทรลลิส นำมาใช้ในการถอดคำรหัสได้ว่า 101110010100 ซึ่งเมื่อตัดบิตหางออกจะได้ว่าข้อมูลก่อนเข้ารหัสคอนโวลูชันคือ 1011100101

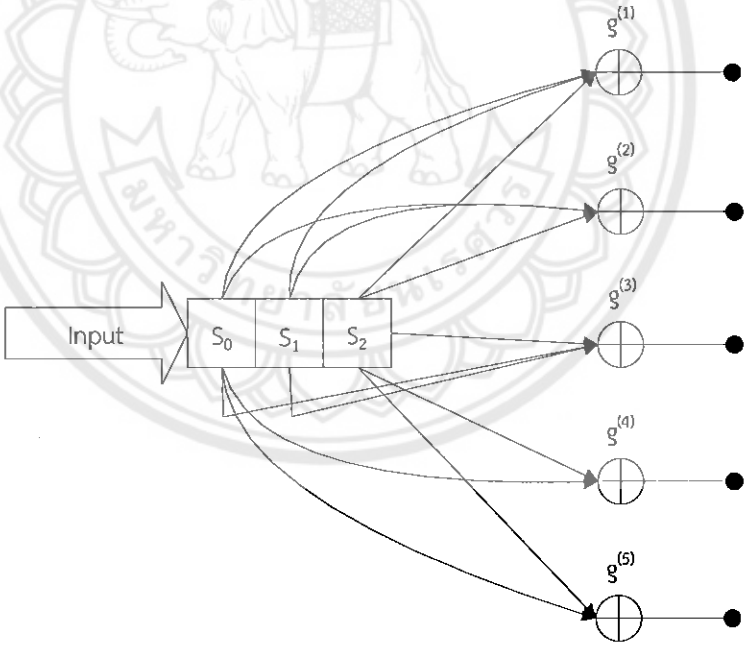
### 4.3.3 กรณีศึกษาของระบบ [5,1,2]

**1. การเข้ารหัสคอนโวลูชัน**

ระบบ [5,1,2] มีความหมายว่า 5 คือมีจำนวนบิตของคำรหัส 5 บิตต่อหนึ่งหน่วยเวลา
1 คือมีจำนวนบิตข้อมูลที่เข้าสู่ระบบ 1 บิตต่อหนึ่งหน่วยเวลา
2 คือมีจำนวนชิฟรีจิสเตอร์ที่เป็นหน่วยความจำ 2 ชิฟ

ซึ่งจะได้เมทริกซ์ก่อกำเนิด [7,7,7,5,5] (ฐาน 8) แปลงได้เป็น [111,111,111,101,101] (ฐาน 2)
โดยกำหนดให้ $g^{(1)}$ คือ 111, $g^{(2)}$ คือ 111, $g^{(3)}$ คือ 111, $g^{(4)}$ คือ 101, $g^{(5)}$ คือ 101
และกำหนดให้ $S_0$ คืออินพุต
$S_1$ คือชิฟรีจิสเตอร์ชิฟที่ 1
$S_2$ คือชิฟรีจิสเตอร์ชิฟที่ 2

จะได้แผนภาพจำลองระบบ [5,1,2] ดังรูปที่ 4.29




**รูปที่ 4.29** แผนภาพจำลองระบบ [5,1,2]

รูปที่ 4.29 คือรูปของแผนภาพการจำลองระบบ [5,1,2] จะเห็นได้ว่าระบบนี้อินพุตจะเข้าครั้งละ 1 บิตที่ $S_0$ และได้คำรหัสจำนวน 5 บิต และการเชื่อมต่อกันระหว่างชิฟรีจิสเตอร์กับตัวบวกแบบมอดูโลทูจะได้ค่าของฟังก์ชันก่อกำเนิด เช่น $g^{(1)}$ ซึ่งมีการเชื่อมต่อกับ $S_0$, $S_1$ และ $S_2$ จะมีค่า 111 เป็นต้น

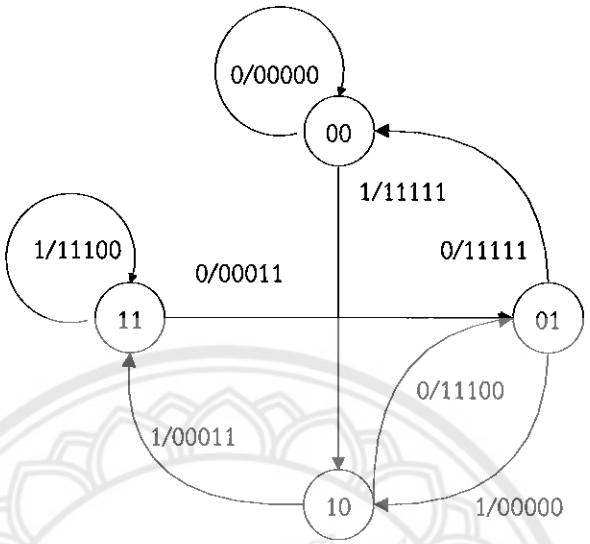
จากนั้นทำการหาอินพุต สถานะของหน่วยความจำ สถานะที่เปลี่ยนไปเมื่อมีอินพุตเข้าสู่วงจร คำรหัสที่ได้ ตามวิธีที่ได้กล่าวมาแล้วในกรณีศึกษาของระบบ [3,1,2] จะทำให้สามารถสร้างตารางแสดงความสัมพันธ์ของระบบ [5,1,2] ได้ดังตารางที่ 4.7

**ตารางที่ 4.7** ตารางแสดงความสัมพันธ์ของระบบ [5,1,2]

| อินพุต($S_0$) | สถานะเดิม($S_1S_2$) | คำรหัส | สถานะถัดไป($S_0S_1$) |
|---|---|---|---|
| 0 | 00 | 00000 | 00 |
| 0 | 01 | 11111 | 00 |
| 0 | 10 | 11100 | 01 |
| 0 | 11 | 00011 | 01 |
| 1 | 00 | 11111 | 10 |
| 1 | 01 | 00000 | 10 |
| 1 | 10 | 00011 | 11 |
| 1 | 11 | 11100 | 11 |

ตารางที่ 4.7 คือตารางแสดงค่าความสัมพันธ์ของระบบ [5,1,2] ซึ่งได้จากการคำนวณอธิบายความสัมพันธ์ได้ว่า เช่นให้สถานะเดิมคือ 00 ในกรณีที่อินพุตคือ 0 เข้าสู่ระบบจะได้ผลลัพธ์คำรหัสคือ 00000 สถานะถัดไปคือ 00 ในกรณีที่อินพุตคือ 1 เข้าสู่ระบบจะได้ผลลัพธ์คำรหัสคือ 11111 สถานะถัดไปคือ 10 เป็นต้น

จากตารางที่ 4.7 สามารถนำมาสร้างเป็นแผนภาพสถานะได้ดังรูปที่ 4.30




**รูปที่ 4.30** แผนภาพสถานะของระบบ [5,1,2]

รูปที่ 4.30 คือแผนภาพสถานะของระบบ [5,1,2] สามารถอธิบายได้เช่น เมื่อที่สถานะเดิมคือ 00 ในกรณีที่อินพุตคือ 0 สถานะถัดไปคือ 00 และได้คำรหัสคือ 00000 ในกรณีที่อินพุตคือ 1 สถานะถัดไปคือ 10 และได้คำรหัสคือ 11111 เป็นต้น

**ตัวอย่าง**แสดงการเข้ารหัสของระบบ [5,1,2] ที่มีเมทริกซ์ก่อกำเนิดคือ [7,7,7,5,5]

กำหนดให้อินพุตคือ 1011100101 และสถานะเริ่มต้นคือ 00 เมื่ออินพุตเข้าสู่ระบบจะทำให้ได้การเปลี่ยนสถานะ และคำรหัสดังตารางที่ 4.8 (โดยในขั้นตอนสุดท้ายจะทำการเติมบิตหางเพื่อปรับให้สถานะของหน่วยความจำกลับไปเป็น 00)

**ตารางที่ 4.8** ตารางแสดง อินพุต การเปลี่ยนสถานะ และคำรหัสจากตัวอย่างอินพุต

| อินพุต | การเปลี่ยนสถานะ | คำรหัส |
|---|---|---|
| 1 | 10 | 11111 |
| 0 | 01 | 11100 |
| 1 | 10 | 00000 |
| 1 | 11 | 00011 |
| 1 | 11 | 11100 |

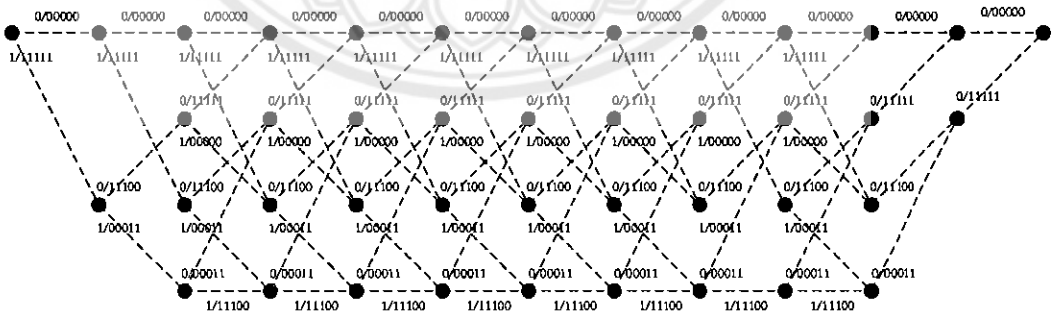
**ตารางที่ 4.8 (ต่อ)** ตารางแสดง อินพุต การเปลี่ยนสถานะ และคำรหัสจากตัวอย่างอินพุต

| อินพุต | การเปลี่ยนสถานะ | คำรหัส |
|---|---|---|
| 0 | 01 | 00011 |
| 0 | 00 | 11111 |
| 1 | 10 | 11111 |
| 0 | 01 | 11100 |
| 1 | 10 | 00000 |
| 0 | 01 | 11100 |
| 0 | 00 | 11111 |

ตารางที่ 4.8 จากการนำอินพุตคือ 1011100101 มาเข้ารหัสคอนโวลูชันจะได้คำรหัสคือ 11111 11100 00000 00011 11100 00011 11111 11111 11100 00000 11100 11111

2. การถอดคำรหัสคอนโวลูชัน กรณีไม่มีบิตผิดพลาด

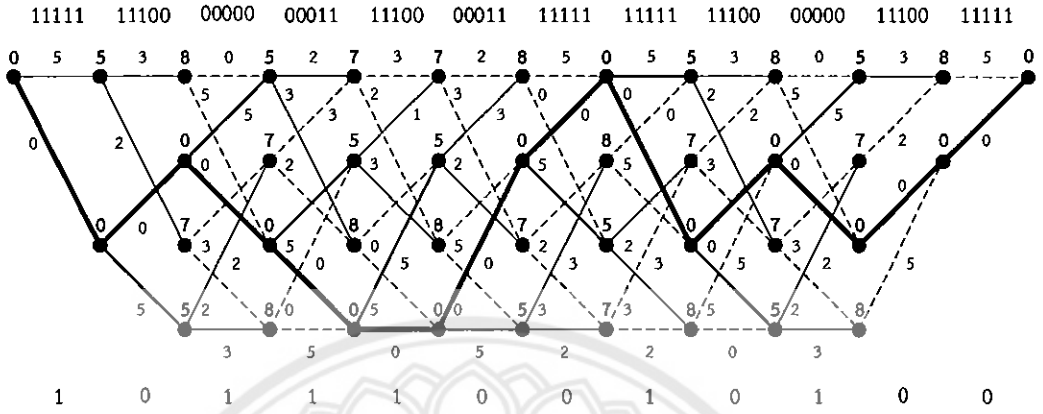
เมื่อนำคำรหัส 111111110000000000111110000011111111111111100000001110011111 ดังตารางที่ 4.8 มาถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บี โดยใช้ตารางที่ 4.7 ในการสร้างแผนภาพเทรลลิสของระบบ [5,1,2] ได้ดังรูปที่ 4.31



**รูปที่ 4.31** แผนภาพเทรลลิสของระบบ [5,1,2]

รูปที่ 4.31 คือแผนภาพเทรลลิสของระบบ [5,1,2] จำนวน 13 ระดับ โดยเริ่มต้นจากสถานะ 0 ณ ระดับที่ 0 และระดับที่ 10 ขึ้นไปจะวาดเฉพาะเส้นทางที่เข้าหาสถานะของวงจรเป็น 0 ทั้งหมด

จากนั้นทำการถอดคำรหัสคอนโวลูชันตามวิธีที่ได้กล่าวมาแล้วในกรณีศึกษาของระบบ [3,1,2] จะได้ผลลัพธ์จากการถอดคำรหัสดังรูปที่ 4.32




**รูปที่ 4.32** แผนภาพเทรลลิสแสดงการถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บี ของระบบ [5,1,2]

รูปที่ 4.32 รูปแสดงแผนภาพเทรลลิสแสดงการถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [5,1,2] โดยที่บนจุดจะถูกกำหนดค่าไว้ด้วยค่าแมทริกซ์สะสมของแต่ละสถานะในทุกระดับ ซึ่งจะได้ผลลัพธ์เป็นเส้นทางบนแผนภาพเทรลลิส นำมาใช้ในการถอดคำรหัสได้ว่า 101110010100 ซึ่งเมื่อตัดบิตหางออกจะได้ว่าข้อมูลก่อนเข้ารหัสคอนโวลูชันคือ 1011100101

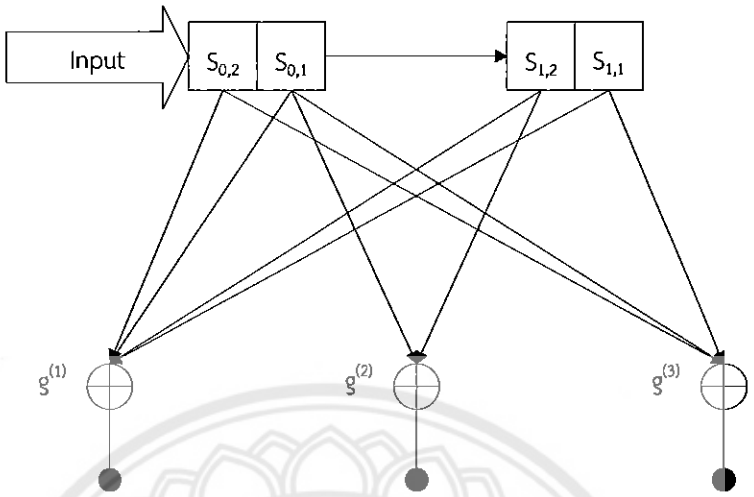
### 4.3.4 กรณีศึกษาของระบบ [3,2,1]

**1. การเข้ารหัสคอนโวลูชัน**

ระบบ [3,2,1] มีความหมายว่า 3 คือมีจำนวนบิตของคำรหัส 3 บิตต่อหนึ่งหน่วยเวลา
2 คือมีจำนวนบิตข้อมูลที่เข้าสู่ระบบ 2 บิตต่อหนึ่งหน่วยเวลา
1 คือมีจำนวนชุดชิฟรีจิสเตอร์ที่เป็นหน่วยความจำ 1 ชุด

ซึ่งจะได้เมทริกซ์ก่อกำเนิด [17,6,15] (ฐาน 8) แปลงได้เป็น [1111,0110,1101] (ฐาน 2)
โดยกำหนดให้ $g^{(1)}$ คือ 1111, $g^{(2)}$ คือ 0110, $g^{(3)}$ คือ 1101
และกำหนดให้ $S_{0,2}$ คืออินพุตบิตที่ 2
$S_{0,1}$ คืออินพุตบิตที่ 1
$S_{1,2}$ คือชิฟรีจิสเตอร์ชิฟที่ 1 ของอินพุตบิตที่ 2
$S_{1,1}$ คือชิฟรีจิสเตอร์ชิฟที่ 1 ของอินพุตบิตที่ 1

จะได้แผนภาพจำลองระบบ [3,2,1] ดังรูปที่ 4.33




**รูปที่ 4.33** แผนภาพจำลองระบบ [3,2,1]

รูปที่ 4.33 คือรูปของแผนภาพการจำลองระบบ [3,2,1] จะเห็นได้ว่าระบบนี้อินพุตจะเข้าครั้งละ 2 บิตที่ $S_{0,2}$ และ $S_{0,1}$ และได้คำรหัสจำนวน 3 บิต และการเชื่อมต่อกันระหว่างชิฟรีจิสเตอร์กับตัวบวกแบบมอดูโลทูจะได้ค่าของฟังก์ชันก่อกำเนิด เช่น $g^{(1)}$ ซึ่งมีการเชื่อมต่อกับ $S_{0,2}$, $S_{0,1}$, $S_{1,2}$ และ $S_{1,1}$ จะมีค่า 1111 เป็นต้น

จากนั้นทำการหาอินพุต สถานะของหน่วยความจำ สถานะที่เปลี่ยนไปเมื่อมีอินพุตเข้าสู่วงจร คำรหัสที่ได้ ตามวิธีที่ได้กล่าวมาแล้วในกรณีศึกษาของระบบ [3,1,2] จะทำให้สามารถสร้างตารางแสดงความสัมพันธ์ของระบบ [3,2,1] ได้ดังตารางที่ 4.9

**ตารางที่ 4.9** ตารางแสดงความสัมพันธ์ของระบบ [3,2,1]

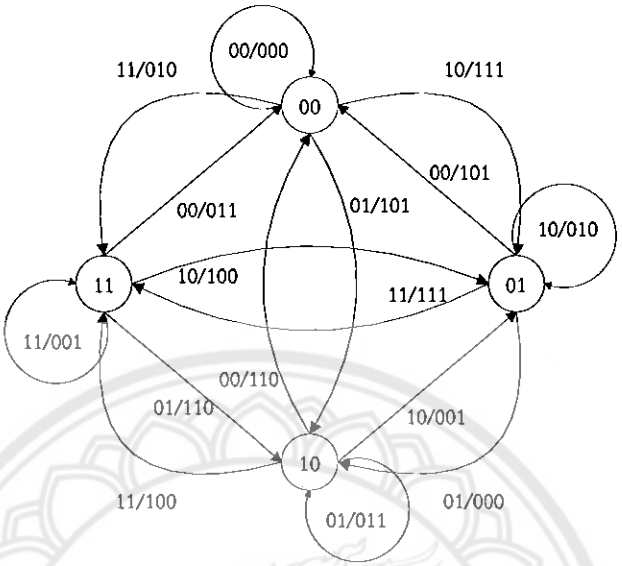
| อินพุต($S_{0,1}S_{0,2}$) | สถานะเดิม($S_{1,2}S_{1,1}$) | คำรหัส | สถานะถัดไป($S_{0,2}S_{0,1}$) |
|---|---|---|---|
| 00 | 00 | 000 | 00 |
| 00 | 01 | 101 | 00 |
| 00 | 10 | 110 | 00 |
| 00 | 11 | 011 | 00 |
| 01 | 00 | 101 | 10 |
| 01 | 01 | 000 | 10 |

**ตารางที่ 4.9 (ต่อ)** ตารางแสดงความสัมพันธ์ของระบบ [3,2,1]

| อินพุต($S_{0,1}S_{0,2}$) | สถานะเดิม($S_{1,2}S_{1,1}$) | คำรหัส | สถานะถัดไป($S_{0,2}S_{0,1}$) |
|---|---|---|---|
| 01 | 10 | 011 | 10 |
| 01 | 11 | 110 | 10 |
| 10 | 00 | 111 | 01 |
| 10 | 01 | 010 | 01 |
| 10 | 10 | 001 | 01 |
| 10 | 11 | 100 | 01 |
| 11 | 00 | 010 | 11 |
| 11 | 01 | 111 | 11 |
| 11 | 10 | 100 | 11 |
| 11 | 11 | 001 | 11 |

ตารางที่ 4.9 คือตารางแสดงค่าความสัมพันธ์ของระบบ [3,2,1] ที่ได้จากการคำนวณ สามารถอธิบายได้เช่น เมื่อที่สถานะเดิมคือ 00 ในกรณีที่อินพุตคือ 00 สถานะถัดไปคือ 00 และได้คำรหัสคือ 000 ในกรณีที่อินพุตคือ 01 สถานะถัดไปคือ 10 และได้คำรหัสคือ 101 ในกรณีที่อินพุตคือ 10 สถานะถัดไปคือ 01 และได้คำรหัสคือ 111 ในกรณีที่อินพุตคือ 11 สถานะถัดไปคือ 11 และได้คำรหัสคือ 010 เป็นต้น

จากตารางที่ 4.9 สามารถนำมาสร้างเป็นแผนภาพสถานะได้ดังรูปที่ 4.34




รูปที่ 4.34 แผนภาพสถานะของระบบ [3,2,1]

รูปที่ 4.34 คือแผนภาพสถานะของระบบ [3,2,1] สามารถอธิบายได้เช่น เมื่อที่สถานะเดิมคือ 00 ในกรณีที่อินพุตคือ 00 สถานะถัดไปคือ 00 และได้คำรหัสคือ 000 ในกรณีที่อินพุตคือ 01 สถานะถัดไปคือ 10 และได้คำรหัสคือ 101 ในกรณีที่อินพุตคือ 10 สถานะถัดไปคือ 01 และได้คำรหัสคือ 111 ในกรณีที่อินพุตคือ 11 สถานะถัดไปคือ 11 และได้คำรหัสคือ 010 เป็นต้น

**ตัวอย่าง**แสดงการเข้ารหัสของระบบ [3,2,1] ที่มีเมทริกซ์ก่อกำเนิดคือ [17,6,15]

กำหนดให้อินพุตคือ 1011100101 และสถานะเริ่มต้นคือ 00 เมื่ออินพุตเข้าสู่ระบบจะทำให้ได้การเปลี่ยนสถานะ และคำรหัสดังตารางที่ 4.10 (โดยในขั้นตอนสุดท้ายจะทำการเติมบิตหางเพื่อปรับให้สถานะของหน่วยความจำกลับไปเป็น 00)

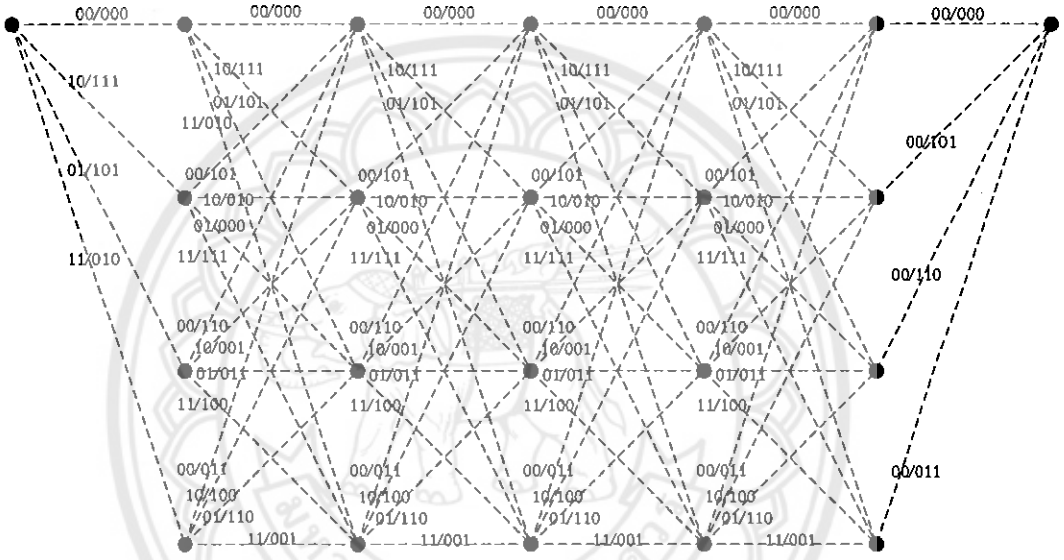
**ตารางที่ 4.10** ตารางแสดง อินพุต การเปลี่ยนสถานะ และคำรหัสจากตัวอย่างอินพุต

| อินพุต | 10 | 11 | 10 | 01 | 01 | 00 |
|---|---|---|---|---|---|---|
| การเปลี่ยนสถานะ | 01 | 11 | 01 | 10 | 10 | 00 |
| คำรหัส | 111 | 111 | 100 | 000 | 011 | 110 |

ตารางที่ 4.10 จากการนำอินพุตคือ 1011100101 มาเข้ารหัสคอนโวลูชันจะได้คำรหัสคือ 111 111 100 000 011 110

**2. การถอดคำรหัสคอนโวลูชัน กรณีไม่มีบิตผิดพลาด**

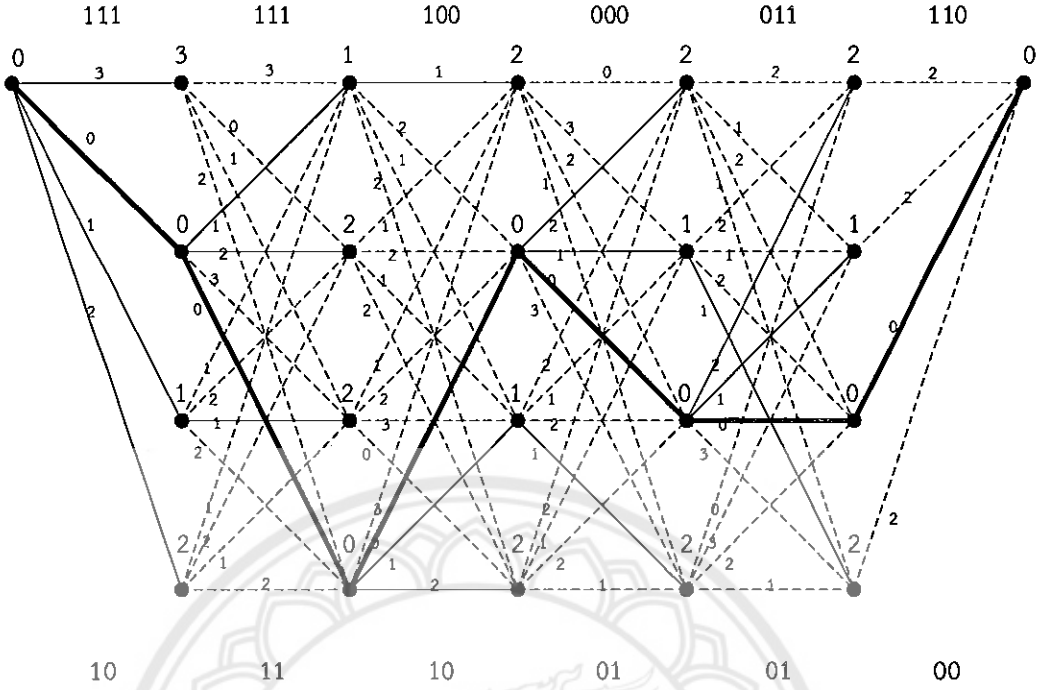
เมื่อนำคำรหัส 111111100000011110 ดังตารางที่ 4.10 มาถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บี โดยใช้ตารางที่ 4.9 ในการสร้างแผนภาพเทรลลิสของระบบ [3,2,1] ได้ดังรูปที่ 4.35




**รูปที่ 4.35** แผนภาพเทรลลิสของระบบ [3,2,1]

รูปที่ 4.35 คือแผนภาพเทรลลิสของระบบ [3,2,1] จำนวน 7 ระดับ โดยเริ่มต้นจากสถานะ 0 ณ ระดับที่ 0 และระดับที่ 5 ขึ้นไปจะวาดเฉพาะเส้นทางที่เข้าหาสถานะของวงจรเป็น 0 ทั้งหมด

จากนั้นทำการถอดคำรหัสคอนโวลูชันตามวิธีที่ได้กล่าวมาแล้วในกรณีศึกษาของระบบ [3,1,2] จะได้ผลลัพธ์จากการถอดคำรหัสดังรูปที่ 4.36

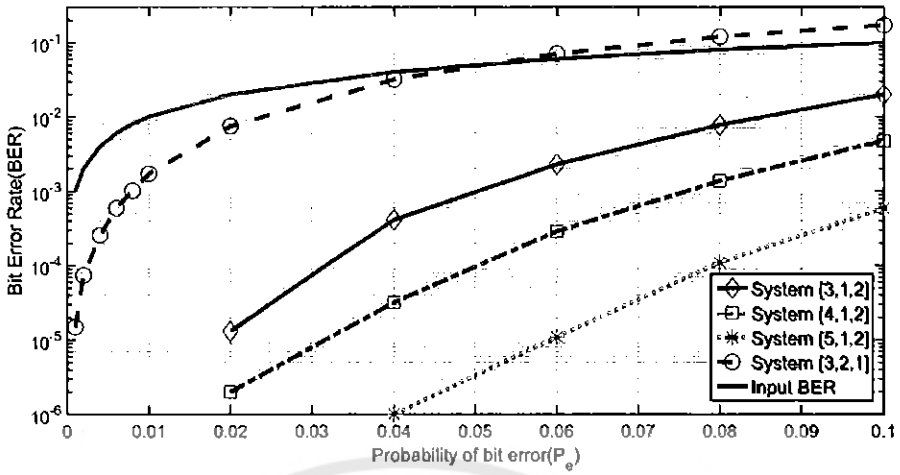



**รูปที่ 4.36** แผนภาพเทรลลิสแสดงการถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บี ของระบบ [3,2,1]

รูปที่ 4.36 รูปแสดงแผนภาพเทรลลิสแสดงการถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีของระบบ [3,2,1] โดยที่บนจุดจะถูกกำหนดค่าไว้ด้วยค่าเมทริกซ์สะสมของแต่ละสถานะในทุกระดับ ซึ่งจะได้ผลลัพธ์เป็นเส้นทางบนแผนภาพเทรลลิส นำมาใช้ในการถอดคำรหัสได้ว่า 101110010100 ซึ่งเมื่อตัดบิตหางออกจะได้ว่าข้อมูลก่อนเข้ารหัสคอนโวลูชันคือ 1011100101

## 4.4 ประสิทธิภาพของรหัสคอนโวลูชัน

หลังจากการสร้างโปรแกรมจำลองรหัสคอนโวลูชันโดยกรณีศึกษาคือการใช้งานรหัสคอนโวลูชันเข้ารหัสข้อมูลจำนวน 1,000,000 บิต ด้วยระบบ [3,1,2] ระบบ [4,1,2] ระบบ [5,1,2] และระบบ [3,2,1] โดยที่แต่ละระบบจะถูกจำลองด้วยความน่าจะเป็นของการเกิดบิตผิดพลาดคือ 0.001 0.002 0.004 0.006 0.008 0.01 0.02 0.04 0.06 0.08 0.1 ด้วยโปรแกรมแมตแล็บตามโครงร่างที่ได้กล่าวมาแล้วในบทที่ 3 นั้น จะสามารถนำผลของอัตราบิตผิดพลาดที่เกิดขึ้นมาเปรียบเทียบถึงประสิทธิภาพของความสามารถในการแก้ไขบิตที่ผิดพลาดโดยสร้างเป็นแผนภูมิดังรูปที่ 4.37




รูปที่ 4.37 แผนภูมิแสดงผลของอัตราบิตที่ผิดพลาดจากกรณีศึกษา

รูปที่ 4.37 แผนภูมิแสดงผลของอัตราบิตที่ผิดพลาดจากกรณีศึกษาทั้ง 4 กรณี โดยกำหนดให้แกนตั้งเป็นแกนแสดงค่าอัตราบิตที่ผิดพลาด (BER) ในรูปลอการิทึม และแกนนอนเป็นแกนแสดงค่าความน่าจะเป็นของการเกิดบิตที่ผิดพลาด ($P_e$) โดยที่เส้นสีฟ้าคือเส้นที่แสดงผลอัตราบิตที่ผิดพลาดของระบบ [3,1,2] เส้นสีส้มคือเส้นที่แสดงผลอัตราบิตที่ผิดพลาดของระบบ [4,1,2] เส้นสีเหลืองคือเส้นที่แสดงผลอัตราบิตที่ผิดพลาดของระบบ [5,1,2] เส้นสีม่วงคือเส้นที่แสดงผลอัตราบิตที่ผิดพลาดของระบบ [3,2,1] และเส้นสีเขียวคือเส้นที่แสดงอัตราบิตที่ผิดพลาดที่ถูกป้อน

วิเคราะห์ผลการจำลองการเข้ารหัสข้อมูลด้วยรหัสคอนโวลูชัน จากรูปที่ 4.37 พบว่าการเข้ารหัสข้อมูลด้วยระบบ [5,1,2] มีประสิทธิภาพดีที่สุด เนื่องจากมีอัตราบิตผิดพลาดน้อยที่สุดซึ่งบ่งชี้ได้ว่าการถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีโดยการตัดสินใจแบบฮาร์ดของระบบ [5,1,2] นั้นมีความสามารถในการแก้ไขความผิดพลาดซึ่งเกิดจากการสุ่มตำแหน่งบิตที่ผิดพลาดมากที่สุด ตามด้วยระบบ [4,1,2] ระบบ [3,1,2] และระบบ [3,2,1] ตามลำดับ จะสังเกตุได้ว่าทุกระบบที่ทำการจำลองนั้นมีจำนวนหน่วยความจำเท่ากันทุกระบบ แต่ค่าอัตรารหัส (Code Rate = k/n) นั้นมีค่าแตกต่างกัน ซึ่งถ้าเรียงค่าอัตรารหัสตามลำดับของประสิทธิภาพในการแก้ไขความผิดพลาดจะได้ว่าอัตรารหัสของระบบ [5,1,2] คือ 0.2 ระบบ [4,1,2] คือ 0.25 ระบบ [3,1,2] คือ 0.33 และระบบ [3,2,1] คือ 0.67 พบว่าระบบ [5,1,2] นั้นมีอัตรารหัสที่น้อยที่สุดแต่มีประสิทธิภาพในการแก้ไขความผิดพลาดมากที่สุด ตามด้วยระบบ [4,1,2] ระบบ [3,1,2] และระบบ [3,2,1] ตามลำดับ และจากผลการถอดคำรหัสคอนโวลูชันที่พบว่าเมื่ออัตรารหัสเพิ่มขึ้นจะทำให้ระยะแฮมมิงระหว่างคำรหัสลดลงจึงทำให้ความสามารถในการแก้ไขความผิดพลาดลดลงตามไปด้วย ซึ่งทำให้สามารถกล่าวสรุปได้ว่าเมื่ออัตรารหัสมีค่าเพิ่มมากขึ้นจะทำให้ประสิทธิภาพในการแก้ไขความผิดพลาดลดลง หรืออัตรารหัสมีความสัมพันธ์แบบแปรผกผันกันกับประสิทธิภาพในการแก้ไขความผิดพลาด

ซึ่งหมายความว่าเมื่อต้องการความสามารถในการแก้ไขความผิดพลาดที่สูงขึ้นจะต้องการจำนวนคำรหัสจำนวนมากต่อข้อมูลจำนวนน้อยอันเป็นผลให้ระบบต้องการช่องส่งสัญญาณที่มีความกว้าง (Bandwidth) มากซึ่งอาจจะทำให้สิ้นเปลืองทรัพยากรมากเกินไปจึงต้องเลือกใช้ระบบที่เหมาะสมกับประเภทของความต้องการความถูกต้องของข้อมูล แต่จากรูปก็ยังมีจุดที่น่าสนใจคือเส้นโค้งอัตราบิตผิดพลาดของระบบ [3,2,1] นั้น ที่ค่าความน่าจะเป็นของการเกิดบิตที่ผิดพลาดมากกว่า 0.05 โดยประมาณนั้นมีประสิทธิภาพแย่กว่าค่าอัตราบิตที่ผิดพลาดที่ถูกป้อน

# บทที่ 5
# สรุปผลการดำเนินงาน

## 5.1 สรุปผลการดำเนินงาน

จากการดำเนินงานด้วยการศึกษาการสร้างบิตข้อมูล การสุ่มตำแหน่งบิตที่ผิดพลาด การรวมข้อมูลภายในช่องส่งสัญญาณ การเข้ารหัสคอนโวลูชัน การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีโดยการตัดสินใจแบบฮาร์ด และการใช้งานโปรแกรมแมตแล็บเบื้องต้น เพื่อนำความรู้จากการศึกษาข้างต้นมาทำการจำลองการทำรหัสคอนโวลูชันของกรณีศึกษาต่างๆ และนำมาสร้างโปรแกรมเพื่อจำลองการทำรหัสคอนโวลูชันของกรณีศึกษาต่างๆ ด้วยโปรแกรมแมตแล็บเพื่อนำผลอัตราบิตผิดพลาดมาเปรียบเทียบด้วยแผนภูมิ เพื่อหาประสิทธิภาพของรหัสคอนโวลูชันในแต่ละกรณีโดยสามารถสรุปได้ดังนี้

### 5.1.1 สรุปผลการจำลองการทำรหัสคอนโวลูชันของกรณีศึกษาต่างๆ

กรณีศึกษาในการจำลองการทำรหัสคอนโวลูชันนั้นประกอบไปด้วยกรณีศึกษา 4 กรณีดังต่อไปนี้

1. กรณีศึกษาของระบบ [3,1,2] ประกอบไปด้วยการเข้ารหัสคอนโวลูชัน และการถอดคำรหัสคอนโวลูชันในกรณีที่ไม่มีบิตผิดพลาด และกรณีที่มีบิตผิดพลาด 8 กรณี
2. กรณีศึกษาของระบบ [4,1,2] ประกอบไปด้วยการเข้ารหัสคอนโวลูชัน และการถอดคำรหัสคอนโวลูชันในกรณีที่ไม่มีบิตผิดพลาด
3. กรณีศึกษาของระบบ [5,1,2] ประกอบไปด้วยการเข้ารหัสคอนโวลูชัน และการถอดคำรหัสคอนโวลูชันในกรณีที่ไม่มีบิตผิดพลาด
4. กรณีศึกษาของระบบ [3,2,1] ประกอบไปด้วยการเข้ารหัสคอนโวลูชัน และการถอดคำรหัสคอนโวลูชันในกรณีที่ไม่มีบิตผิดพลาด

จากผลการจำลองในบทที่ 4 สามารถสรุปผลได้ว่าค่าของเมทริกซ์สะสมในเส้นทางที่ถูกเลือกนั้นจะมีค่าขึ้นอยู่กับจำนวนบิตที่ผิดพลาดภายในช่องส่งสัญญาณ และด้วยการจำลองให้เกิดความผิดพลาดทั้ง 8 กรณีของกรณีศึกษาของระบบ [3,1,2] ในช่องส่งสัญญาณ พบว่าจะสามารถแก้ไขความผิดพลาดที่เกิดขึ้นได้ด้วยการถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีโดยการตัดสินใจแบบฮาร์ดได้ทุกกรณี และทำให้ทราบว่าการเลือกใช้เมทริกซ์ก่อกำเนิดที่ทำให้ระยะฟรีมีค่ามากนั้นจะทำให้ความสามารถทำให้การตัดสินใจเลือกเส้นทางที่ถูกดียิ่งขึ้น

### 5.1.2 สรุปผลประสิทธิภาพของรหัสคอนโวลูชัน

จากผลการจำลองการเข้ารหัสคอนโวลูชันด้วยระบบที่ให้เป็นกรณีศึกษาได้แก่ระบบ [3,1,2] ระบบ [4,1,2] ระบบ [5,1,2] และระบบ [3,2,1] โดยใช้ความน่าจะเป็นในการเกิดบิตผิดพลาดค่าต่างๆ ได้แก่ 0.001 0.002 0.004 0.006 0.008 0.01 0.02 0.04 0.06 0.08 0.1 เป็นกรณีศึกษาในแต่ละระบบ และใช้การถอดคำรหัสคอนโวลูชันด้วยขั้นตอนวิธีของวิเทอร์บีโดยการตัดสินใจแบบฮาร์ด สรุปได้ว่าการใช้ระบบ [5,1,2] ในการเข้ารหัสคอนโวลูชันมีประสิทธิภาพดีที่สุด ตามด้วยระบบ [4,1,2] ระบบ [3,1,2] และระบบ [3,2,1] ตามลำดับ และทำให้ทราบว่า 1 ในพารามิเตอร์ที่มีผลต่อประสิทธิภาพของการเข้ารหัสข้อมูลด้วยรหัสคอนโวลูชันคืออัตราคำรหัส (Code Rate = k/n) โดยที่ประสิทธิภาพของการเข้ารหัสคอนโวลูชันนั้นมีความสัมพันธ์แบบแปรผกผันกับอัตราคำรหัส

## 5.2 ปัญหาที่พบขณะดำเนินโครงงาน

เนื่องจากการสร้างโปรแกรมจำลองที่เกี่ยวกับความน่าจะเป็นที่จะต้องใช้ข้อมูลตัวอย่างที่เพียงพอในการจำลองซึ่งมีจำนวนมาก และด้วยการใช้ศึกษารหัสคอนโวลูชันซึ่งจะสามารถเลือกระบบที่มีหน่วยความจำจำนวนมากได้นั้น จะต้องใช้ทั้งเทคนิคในการเขียนโปรแกรมเพื่อลดภาระในการคำนวณ และการเก็บค่าตัวแปรต่างๆ ภายในโปรแกรมจำลอง และการใช้อุปกรณ์ที่มีทรัพยากรเพียงพอในการรองรับกับ 2 ปัจจัยข้างต้นเพื่อการทำงานที่รวดเร็ว และมีประสิทธิภาพ

## 5.3 ข้อเสนอแนะ

โครงงานนี้สามารถนำไปพัฒนาต่อเพื่อเพิ่มความสามารถต่างๆ เช่นความสามารถในการสร้างสัญญาณต่างๆ ภายในช่องส่งสัญญาณ การมอดูเลต การดีมอดูเลต ตัวกรอง และตัวตัดสินใจสัญญาณในภาครับ เป็นต้น

# เอกสารอ้างอิง


[1] Proakis, J. G., and Salehi M., *Digital Communication*, 5th ed., McGraw-Hill, New York, 2008.

[2] Schlegel, C. B., and Perez, L. C., *Trellis and Turbo Coding*, IEEE Press, United States of America, 2004.

[3] สรุเชษฐ์ กานต์ประชา และเศรษฐา ตั้งค้าวานิช, ประสิทธิภาพของระบบซีดีเอ็มเอแบบไดเร็กซีเควนซ์ที่มีการเข้ารหัส Performance of Direct Sequence Spread Spectrum CDMA systems, คณะวิศวกรรมศาสตร์, มหาวิทยาลัยนเรศวร, 2552.

[4] การันตร์ พันเลิศพานณิชย์, *ประสิทธิภาพของการเข้ารหัสแบบคอนโวลูชันโค้ดโดยมีการใช้อินเทอร์ลีฟ*, ปริญญานิพนธ์, ภาควิชาวิศวกรรมไฟฟ้า, คณะวิศวกรรมศาสตาร์, มหาวิทยาลัยนเรศวร, 2553.

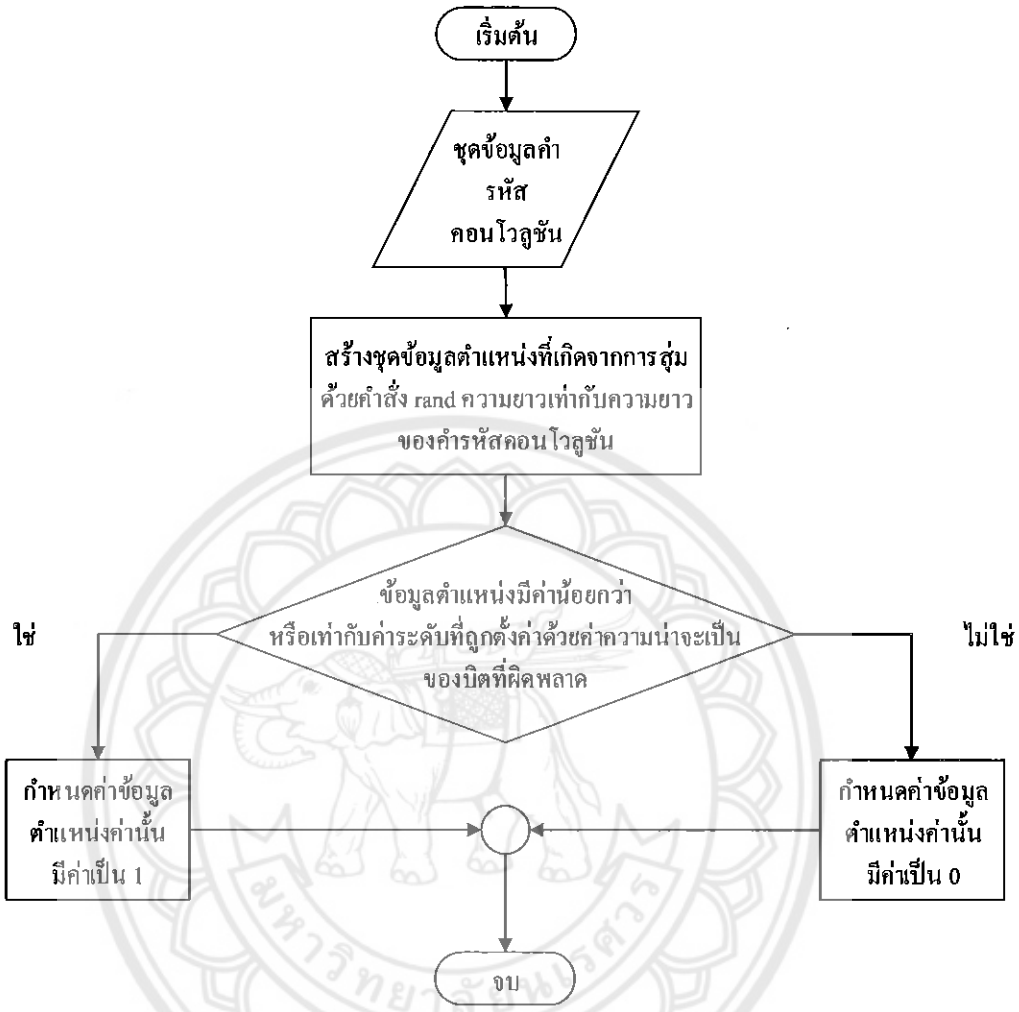
[5] Couch, L. W., *Digital and Analog Communication Systems*, 8th ed., Person Education, England, 2013.

[6] พิสิฐ วนิชชานันท์ และคณะ, *ทฤษฎีช่องส่งสัญญาณ Channel Coding Theory*, พิมพ์ครั้งที่ 1, สถาบันวิจัยและพัฒนาอุตสาหกรรมโทรคมนาคม, 2552

# ภาคผนวก

# การสุ่มตำแหน่งบิตที่ผิดพลาด

แผนภาพแสดงผังงานของวิธีการในการสุ่มตำแหน่งบิตที่ผิดพลาด




รูปที่ 1 แผนผังงานการสุ่มตำแหน่งบิตที่ผิดพลาด

รูปที่ 1 แผนภาพผังงานของการสุ่มตำแหน่งบิตที่ผิดพลาดทำให้กล่าวได้ว่า ในการสุ่มตำแหน่งบิตที่ผิดพลาดมีขั้นตอนในการทำงานดังนี้

1. รับค่าชุดข้อมูลคำรหัสคอนโวลูชันจากตัวเข้ารหัสคอนโวลูชัน

2. สร้างชุดข้อมูลที่มีความยาวเท่ากับความยาวของชุดข้อมูลคำรหัสคอนโวลูชัน ด้วยการสุ่มโดยใช้คำสั่ง rand ซึ่งจะเป็นการสุ่มค่าตั้งแต่ 0 ไปจนถึง 1 ด้วยความน่าจะเป็นแบบสม่ำเสมอ

3. สร้างเงื่อนไขเป็นค่าระดับซึ่งจะถูกตั้งค่าด้วยค่าความน่าจะเป็นของบิตที่ผิดพลาด โดยกำหนดให้ค่าของข้อมูลตำแหน่งที่มีค่าน้อยกว่าหรือเท่ากับค่าระดับมีค่าเป็น 1 และค่าของข้อมูลตำแหน่งที่มีค่ามากกว่าค่าระดับมีค่าเป็น 0

# ประวัติผู้ดำเนินโครงงาน

ชื่อ นายอภิสิทธิ์ ครุฑศรี

ภูมิลำเนา 53/31 ถ.บึงสีไฟ ต.ในเมือง อ.เมือง จ.พิจิตร

ประวัติการศึกษา

- จบการศึกษาในระดับมัธยมศึกษาจากโรงเรียนพิจิตรพิทยาคม
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรี ชั้นปีที่ 5 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

E-mail: apisitkr55@email.nu.ac.th