สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# การปรับปรุงประสิทธิภาพระบบโทรศัพท์เคลื่อนที่ DS-CDMA โดยใช้ระบบเข้ารหัสแบบเทอร์โบ

# PERFORMANCE IMPROVEMENT OF DS-CDMA MOBILE TELEPHONE SYSTEM BASED ON TURBO-CODE ENCODING SYSTEM

T051883

ภัทรพงษ์ ผาสุขกิจ
PATTARAPONG PHASUKKIT

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
บัณฑิตวิทยาลัย
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ. 2547

ISBN 974-9680-96-0

วพ.
ภ366ก
2547
เลขหมู่.......
เลขทะเบียน 51883
วัน,เดือน,ปี - 4 ส.ค. 2547
b. 11401256X
i.
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# PERFORMANCE IMPROVEMENT OF DS-CDMA MOBILE TELEPHONE SYSTEM BASED ON TURBO-CODE ENCODING SYSTEM

PATTARAPONG PHASUKKIT

A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF ENGINEERING IN TELECOMMUNICATION ENGINEERING
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2004

ISBN 974-9680-96-0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COPYRIGHT 2004
SCHOOL OF GRADUATE STUDIES
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| **หัวข้อวิทยานิพนธ์** | การปรับปรุงประสิทธิภาพระบบโทรศัพท์เคลื่อนที่ DS-CDMA โดยใช้ระบบเข้ารหัสแบบเทอร์โบ |
| **นักศึกษา** | นายภัทรพงษ์ ผาสุขกิจ |
| **รหัสประจำตัว** | 44611207 |
| **ปริญญา** | วิศวกรรมศาสตรมหาบัณฑิต |
| **สาขาวิชา** | วิศวกรรมโทรคมนาคม |
| **พ.ศ.** | 2547 |
| **อาจารย์ผู้ควบคุมวิทยานิพนธ์** | รศ.สมยศ จุณณะปิยะ |

# บทคัดย่อ


วิทยานิพนธ์ฉบับนี้ นำเสนอวิธีการเพิ่มประสิทธิภาพให้กับระบบ DS-CDMA มาตรฐาน IS-95 โดยนำวิธีการเข้ารหัสข้อมูลแบบเทอร์โบ โดยใช้อัลกอริทึม MAP ในการถอดรหัสเข้ามาแทนที่ระบบเข้ารหัสแบบคอนโวลูชั่นแบบเดิมที่ใช้งานในระบบ DS-CDMA มาตรฐาน IS-95 โดยจำลองการทำงาน โดยเขียนโปรแกรมจำลองการทำงาน และแสดงประสิทธิภาพของระบบออกมาอยู่ในรูปของอัตราบิทข้อมูลผิดพลาด (Bit-Error-Rate) เทียบกับค่าพลังงานบิท Eb/No โดยทำการเปรียบเทียบประสิทธิภาพของระบบ DS-CDMA เมื่อนำการเข้ารหัสข้อมูลแบบเทอร์โบเข้ามาใช้แทนที่การเข้ารหัสแบบเดิม (Convolution Codes) และทำการเปลี่ยนแปลงขนาดเฟรมข้อมูล โดยเปลี่ยนแปลงอัตราการส่งข้อมูลตามมาตรฐาน IS-95 ที่ 9.6 kbps,4.8 kbps และ 2.4 kbps และเปลี่ยนรูปแบบการทำอินเทอร์ลีฟเวอร์ ทั้งอินเทอร์ลีฟเวอร์ภายในชุดเข้ารหัส แ ละอินเทอร์ลีฟเวอร์ภายนอก ซึ่งเ ป็นอินเทอร์ลีฟเวอร์หลังจากที่ได้เข้ารหัสข้อมูลตามมาตรฐาน IS-95 และเปลี่ยนแปลงจำนวนรอบของชุดถอดรหัสข้อมูลในการเข้ารหัสข้อมูลแบบเทอร์โบ เพื่อให้ได้ประสิทธิภาพของระบบ DS-CDMA ออกมาดีที่สุด




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|---|---|
| **Thesis Title** | Performance improvement of DS-CDMA mobile telephone system based on turbo-code encoding system |
| **Student** | Mr. Pattarapong Phasukkit |
| **Student ID.** | 44611207 |
| **Degree** | Master of Engineering |
| **Program** | Telecommunication Engineering |
| **Year** | 2004 |
| **Thesis Advisor** | Assoc. Prof. Somyot Junnapiya |


# ABSTRACT

This thesis presents the way to improve the performance of DS-CDMA (IS-95 Standard) by using turbo-code system to act as old encoding (convolution-code). Simulations of system are used computer programming. Results are presented in performance format BER (Bit-Error-Rate) versus Eb/No. By using turbo-code will change parameters, iteration of decoding system, format of interleave data (internal and external) and data rate of DS-CDMA system at 9.6 kbps, 4.8 kbps and 2.4 kbps in DS-CDMA system so as good as optimal performance.




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ


วิทยานิพนธ์ ฉบับนี้สำเร็จลุล่วงได้อย่างดีด้วยคำแนะนำ และคำปรึกษาเกี่ยวกับการดำเนินการศึกษาและวิจัยจากท่าน รศ.สมยศ จุณณะปิยะ อาจารย์ผู้ควบคุมวิทยานิพนธ์ ข้าพเจ้ารู้สึกทราบซึ้งในความอนุเคราะห์จากท่านและขอกราบขอบพระคุณอย่างสูง

ขอขอบพระคุณ ท่าน รศ.ดร. ปัญญา ฐิติมัชฌิมา ภาควิชาวิศวกรรมโทรคมนาคม ที่ช่วยเหลือให้คำแนะนำ และกำลังใจ เป็นสิ่งที่มีคุณค่า ต่อข้าพเจ้ามาก

ขอขอบพระคุณ ท่าน ดร.พรชัย ทรัพย์นิธิ ภาควิชาวิศวกรรมโทรคมนาคม ผู้ที่ซึ่งให้ความกระจ่าง สิ่งที่ข้าพเจ้าไม่เข้าใจ ผู้ที่ทำให้ข้าพเจ้ารู้สึกว่าปัญหาที่ไม่เข้าใจ สามารถเข้าใจปัญหาได้ ถ้าหากว่ามีผู้ที่รู้อย่างแท้จริงมาให้ความกระจ่าง

ขอขอบพระคุณ หน่วยงานสำนักวิจัยและบริการคอมพิวเตอร์ ที่สนับสนุนเครื่องมือต่าง ๆ คุณสุกุมา มุ่ยสีผู้มีส่วนสำคัญทำให้วิทยานิพนธ์เป็นรูปเล่มเสร็จสมบูรณ์ขึ้นมาได้ พี่ชายที่แสนดี คุณนเรศ ศรีจาด และน้องผานิจ ละมุล ผู้ที่คอยให้กำลังใจมาตลอด

ขอขอบคุณเทคโนโลยีต่าง ๆ และงานวิจัยของท่านต่าง ๆ ที่เกี่ยวข้องที่ทำให้ผู้วิจัยสามารถเข้าใจ และเรียนรู้ได้ทันเหตุการณ์

สุดท้ายขอกราบขอบพระคุณ คุณพ่อสมศักดิ์ คุณแม่ประเทือง ผาสุขกิจ คุณแม่โสภา มุ่ยสี และคุณแม่สวง ภู่ม่วง ผู้สนับสนุนการศึกษาโดยตลอด

คุณค่าและประโยชน์อันพึงมีจากวิทยานิพนธ์ฉบับนี้ ผู้วิจัยขอมอบแด่ผู้มีพระคุณทุกท่าน

ภัทรพงษ์ ผาสุขกิจ




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ

หน้า

บทคัดย่อภาษาไทย........................................................................................I
บทคัดย่อภาษาอังกฤษ....................................................................................II
กิตติกรรมประกาศ........................................................................................III
สารบัญ........................................................................................................ IV
สารบัญตาราง..............................................................................................VII
สารบัญภาพ...............................................................................................VIII
บทที่ 1 บทนำ....................................................................................................1
1.1 ความเป็นมาและความสำคัญของปัญหา......................................................1
1.2 เหตุผลที่ทำการศึกษา..............................................................................1
1.3 หลักการที่นำเสนอ...................................................................................2
1.4 การเปรียบเทียบกับหลักการที่มีอยู่เดิม......................................................2
1.5 ขอบเขตการวิจัย......................................................................................2

บทที่ 2 ระบบ DS-CDMA และเทอร์โบโค้ด..........................................................4
2.1 พื้นฐานระบบ DS-CDMA.............................................................................4
2.1.1 โค้ด Maximal Sequence..........................................................................4
2.1.2 การสร้างโค้ด m-sequence.......................................................................8
2.1.3 การสเปรดสเปกตรัมแบบ Direct Sequence.............................................13
2.1.4 วงจรภาคส่งของระบบ Direct Sequence ที่ใช้การมอดูเลตแบบ BPSK ......16
2.1.5 วงจรภาครับของระบบ Direct Sequence ที่ใช้การมอดูเลตแบบ BPSK ......17
2.2 ระบบโทรศัพท์เคลื่อนที่ CDMA..................................................................18
2.2.1 ประเภทของโค้ดที่ใช้ในระบบ CDMA (IS-95)...........................................19
2.2.2 โค้ด Walsh...........................................................................................20
2.2.3 โค้ด Long PN.......................................................................................23
2.2.4 โค้ด Short PN.......................................................................................23
2.2.5 การส่งสัญญาณจากสถานีฐานไปหาโทรศัพท์...........................................23
2.2.6 ช่องสัญญาณไพลอต.............................................................................24
2.2.7 ช่องสัญญาณซิงโครไนเซชั่น...................................................................25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ (ต่อ)

หน้า

2.2.8 ช่องสัญญาณเพจจิง ....29
2.2.9 ช่องสัญญาณทราฟฟิก ....30
2.2.10 การส่งสัญญาณจากโทรศัพท์ไปหาสถานีฐาน ....32
2.2.11 ช่องสัญญาณแอกเซส ....33
2.2.12 ช่องสัญญาณทราฟฟิกขากลับ ....38
2.2.13 คุณลักษณะและข้อดีของระบบ CDMA ....41
2.2.14 ขนาดความจุของระบบ CDMA ....42
2.2.15 การคำนวณค่า Co-Channel Interference ในระบบ CDMA ....43
2.2.16 ประเภทของไดเวอร์ซิตีที่มีใช้ในระบบ CDMA ....44
2.3 คอนโวลูชันแนลโค้ด ....44
2.3.1 คุณสมบัติทั่ว ๆ ไปของคอนโวลูชันแนลโค้ด ....45
2.3.2 การเข้ารหัสของคอนโวลูชันแนลโค้ด ....50
2.3.3 การคำนวณหาซินโดรม ....53
2.3.4 การถอดรหัสของคอนโวลูชันแนลโค้ดและการแพร่กระจายของรหัสที่ผิด..54
2.3.5 โครงสร้างแบบต้นไม้ และคุณสมบัติของระยะห่างของคอนโวลูชันแนลโค้ด....58
2.4 การถอดรหัสคอนโวลูชั่น ....66
2.5 รหัสเทอร์โบ ....71
2.5.1 บทนำ ....71
2.5.2 วงจรเข้ารหัสเทอร์โบ ....71
2.5.3 วงจรถอดรหัสเทอร์โบ ....76
2.5.4 การคำนวณอัตราส่วนความน่าเป็นจริง ....77
2.6 การออกแบบอินเทอร์ลีฟ ....82
2.6.1 การทำอินเทอร์ลีฟแบบ Block Interleave ....84
2.6.2 การทำอินเทอร์ลีฟแบบ Random ....84
2.6.3 เซอร์คูล่าร์-ชีฟ อินเทอร์ลีฟ ....85
2.6.4 การทำอินเทอร์ลีฟเวอร์แบบเซมิแรนดอม ....86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ (ต่อ)

หน้า


บทที่ 3 วิธีการดำเนินงาน....88

3.1 ชุดสร้างข้อมูลไบนารี่....88

3.2 ชุดเข้ารหัสเทอร์โบและอัลกอริทึมแบบ MAP ....89

3.2.1 ชุดเข้ารหัสเทอร์โบ....89

3.2.2 การจำลองการทำงานของชุดสลับข้อมูลแบบ Random และแบบ Block .....90

3.2.3 การออกแบบชุดถอดรหัสเทอร์โบแบบใช้อัลกอริทึมแบบ MAP ....92

3.3 การออกแบบระบบ DS-CDMA มาตรฐาน IS-95 ....94

บทที่ 4 ผลการวิจัย ....110

บทที่ 5 สรุปผลการทดลอง....123

5.1 สรุปผลการวิจัย....123

5.2 ปัญหาและข้อเสนอแนะในการพัฒนางานวิจัยต่อไป....125

เอกสารอ้างอิง....126

ภาคผนวก ....129

ภาคผนวก ก อัลกอริทึมในการถอดรหัสเทอร์โบ ....130

ภาคผนวก ข แสดงตัวอย่างในการคำนวณการถอดรหัสของอัลกอรึทึม MAP ....141

ภาคผนวก ค รายชื่อผลงานวิจัยที่เกี่ยวข้อง ....147

ประวัติผู้เขียน ....89



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง

ตารางที่ หน้า


2.1 64 –ary Walsh Functions ....22

2.2 โครงสร้างของแพ็กเกต....38

2.3 การหาระยะแฮมมิงระหว่างชุดบิทที่รับได้ 101000001 เฉพาะ 9 บิทแรก
กับรหัสทั้ง 8 เส้นทาง ....67

2.4 การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิทที่ 10-12 ซึ่งมีค่าเท่ากับ 011....68

2.5 การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิทที่ 13-15 ซึ่งมีค่าเท่ากับ 101 ....69

2.6 การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิทที่ 16-18 ซึ่งมีค่าเท่ากับ 011....69

2.7 การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิทที่ 19-21 ซึ่งมีค่าเท่ากับ 101....70

2.8 การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิทที่ 22-24 ซึ่งมีค่าเท่ากับ 100 ....70

2.9 แสดงชุดลำดับข้อมูลทางด้านอินพุท และเอาท์พุท ของชุดเข้ารหัส ....83

4.1 พารามิเตอร์ของชุดเข้ารหัสเทอร์โบ ....110

4.2 พารามิเตอร์ของชุดเข้ารหัสคอนโวลูชั่น ....110

4.3 พารามิเตอร์ของระบบ DS-CDMA (มาตรฐาน IS-95) ....111

ข.1 Trellis section at $u_k$=0 ....142

ข.2 Trellis section at $u_k$=1 ....142

ข.3 Decoder inputs ....142

ข.4 แสดงผลลัพธ์ของ $\gamma(s',s)$ at $u_k = 0$ ....143

ข.5 แสดงผลลัพธ์ของ $\gamma(s',s)$ at $u_k = 1$ ....143

ข.6 แสดงผลลัพธ์ $\gamma^e(s',s)$ at $u_k = 0$ ....143

ข.7 แสดงผลลัพธ์ของ $\gamma^e(s',s)$ at $u_k = 1$ ....144

ข.8 แสดงผลลัพธ์ของ $\tilde{\alpha}(s)$ ....144

ข.9 แสดงผลลัพธ์ของ $\tilde{\beta}(s)$ ....145

ข.10 แสดงผลลัพธ์ของ $\sigma(s)$ ที่ $u_k = 0$ ....145

ข.11 แสดงผลลัพธ์ของ $\sigma(s)$ ที่ $u_k = 1$ ....145

ข.12 แสดงผลลัพธ์ของ $L_extrinsic$ ....146

ข.13 การตัดสินใจขั้นสุดท้าย....146



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ

ภาพที่ หน้า


2.1 ค่าอัตสหสัมพันธ์ของโค้ด m-sequence ที่มีการเลื่อนไปของโค้ดที่ค่าต่าง ๆ ......................6
2.2 โครงสร้างทั่วไปของวงจรสร้างโค้ด m-sequence ........................................................8
2.3 ตัวอย่างการทำสเปรดสเปกตรัม ..................................................................................15
2.4 วงจรภาคส่งของระบบ DS ที่ใช้การมอดูเลตแบบ BPSK ...............................................16
2.5 ตัวอย่างของสัญญาณที่เกิดขึ้นในวงจรภาคส่งของระบบ DS ที่ใช้ BPSK ........................16
2.6 วงจรภาครับของระบบ DS ที่ใช้การมอดูเลตแบบ BPSK ...............................................18
2.7 ตัวอย่างของสัญญาณที่เกิดขึ้นในวงจรภาครับของระบบ DS ที่ใช้ BPSK ........................18
2.8 วงจร 15-stage LFSR สำหรับ I ....................................................................................23
2.9 วงจร 15-stage LFSR สำหรับ Q ...................................................................................23
2.10 การแบ่งช่องสัญญาณที่ส่งออกโดยสถานีฐาน .............................................................24
2.11 วงจรสร้างสัญญาณไพลอต ........................................................................................25
2.12 วงจรสร้างสัญญาณซิงโครไนเซชั่น .............................................................................26
2.13 วงจรเข้ารหัสคอนโวลูชั่นสำหรับช่องสัญญาณซิงก์ ......................................................26
2.14 โครงสร้างเฟรมและรูปแบบของข้อมูลที่ใช้ในช่องสัญญาณซิงก์ ...................................27
2.15 โครงสร้างของโพลิโนเมียลที่ใช้ในการเข้ารหัส CRC ของช่องสัญญาณซิงก์ ...................28
2.16 โครงสร้างของวงจรเข้ารหัส CRC สำหรับช่องสัญญาณซิงก์ .........................................28
2.17 ขั้นตอนการสร้างสัญญาณเพจจิง ................................................................................29
2.18 Long code generator ...................................................................................................30
2.19 รายละเอียดรูปแบบของ paging channel long code mask ............................................30
2.20 ขั้นตอนการสร้างช่องสัญญาณทราฟฟิก .....................................................................31
2.21 รายละเอียดรูปแบบของ forward traffic channel public long code mask .......................32
2.22 รายละเอียดของ ESN ..................................................................................................32
2.23 การจัดช่องสัญญาณที่ส่งออกโดยเครื่องโทรศัพท์เคลื่อนที่ ............................................33
2.24 ขั้นตอนการสร้างสัญญาณสำหรับส่งผ่านช่องสัญญาณแอกเซสในทิศทางที่จะส่งจาก
โทรศัพท์เคลื่อนที่ไปยังสถานีฐาน .................................................................................33
2.25 โครงสร้างของเฟรมที่ส่งออกผ่านช่องสัญญาณแอกเซส ..............................................34
2.26 โครงสร้างวงจรเข้ารหัสแบบคอนโวลูชั่น ......................................................................34
2.27 การทำอินเตอร์ลีฟสำหรับ reverse link .........................................................................36


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ (ต่อ)


ภาพที่ หน้า

2.28 การสเปรดสัญญาณด้วย Long PN Code ....................................................37

2.29 Long code generator ....................................................37

2.30 Access channel long code mask ....................................................38

2.31 ขั้นตอนการสร้างสัญญาณทราฟฟิกในทิศทางจากโทรศัพท์เคลื่อนที่ไปยังสถานีฐาน ..........39

2.32 ตัวอย่างการเปิดปิดเกตสำหรับการส่งที่อัตราบิทที่ค่าต่าง ๆ ....................................................40

2.33 รายละเอียดรูปแบบของ Public long code mask ....................................................40

2.34 กลุ่มของเซลล์ที่อยู่รอบ ๆ เซลล์ที่กำลังพิจารณา ....................................................44

2.35 ลักษณะทั่วไปของตัวเข้ารหัสแบบคอนโวลูชันแนล ....................................................45

2.36 วงจรเข้ารหัสที่ใช้ชีฟท์รีจิสเตอร์ (n-k)(N-1) ตัว (วงจรเข้ารหัสแบบที่ 1) ....................................................51

2.37 วงจรเข้ารหัสที่ใช้ชีฟท์รีจิสเตอร์ k(N-1) ตัว (วงจรเข้ารหัสแบบที่ 2) ....................................................52

2.38 วงจรเข้ารหัสแบบที่ 1 สำหรับ (2,1) คอนโวลูชันแนลโค้ดของตัวอย่างที่ 2.5 ....................................................53

2.39 วงจรเข้ารหัสแบบที่ 2 สำหรับ (2,1) คอนโวลูชันแนลโค้ดของตัวอย่างที่ 2.5 ....................................................53

2.40 ตัวคำนวณหาซินโดรมสำหรับคอนโวลูชันแนลโค้ด ....................................................54

2.41 วงจรถอดรหัสทั่ว ๆ ไปสำหรับคอนโวลูชันแนลโค้ด ....................................................56

2.42 การแบ่งครั้งแรก ....................................................59

2.43 การแบ่งครั้งที่สอง ....................................................60

2.44 ตารางที่แสดงการพัฒนาของต้นไม้ที่ใช้สำหรับ (n,1) คอนโวลูชันแนลโค้ด ....................................................60

2.45 ลักษณะการกระจายกิ่งของ (n,1) คอนโวลูชันแนลโค้ด ....................................................61

2.46 โครงสร้างของ (3,1) คอนโวลูชันแนลโค้ดของตัวอย่างที่ 2.6 ....................................................62

2.47 ลักษณะต้นไม้ที่ใช้สำหรับ (n,k) คอนโวลูชันแนลโค้ด ....................................................63

2.48 ตัวอย่างวงจรเข้ารหัสคอนโวลูชั่น ....................................................67

2.49 โครงสร้างพื้นฐานของวงจรเข้ารหัสเทอร์โบ ....................................................72

2.50 แผนภาพการเปลี่ยนสถานะของวงจรเข้ารหัสย่อยในรูปที่ 2.49 ....................................................73

2.51 แผนภาพเทรลลิสของวงจรเข้ารหัสย่อยในรูปที่ 2.49 ....................................................73

2.52 ตัวอย่างวงจรเข้ารหัสย่อย ....................................................74

2.53 ตัวอย่างขั้นตอนการเข้ารหัส ....................................................75

2.54 แผนภาพเทรลลิสเมื่อเข้ารหัสวงจรเข้ารหัสย่อยด้วยข้อมูล 11011100 โดยบิทหางมีค่าเท่ากับ 111 ผลลัพธ์ที่ได้คือ ....................................................75



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ (ต่อ)


ภาพที่ หน้า

2.55 แผนภาพการเปลี่ยนสถานะเมื่อเข้ารหัสด้วยข้อมูล 11011100 โดยบิทหางมีค่าเท่ากับ 111.....76

2.56 โครงสร้างพื้นฐานของวงจรถอดรหัสเทอร์โบ ..........77

2.57 แสดงการทำอินเตอร์ลีฟ ประกอบด้วยชุดเข้ารหัสแบบ RSC จำนวน 2 ชุด ..........82

2.58 แสดงตัวอย่างของรหัสที่ได้จากชุดเข้ารหัสเทอร์โบโค้ด ..........83

2.59 แสดงการทำอินเตอร์ลีฟเวอร์แบบบล๊อค Block Interleave ..........84

2.60 แสดงการทำอินเตอร์ลีฟเวอร์แบบ Random โดยความมีความยาวของชุดข้อมูลเท่ากับ 8 ......85

2.61 แสดงการทำอินเตอร์ลีฟแบบเซอร์คูล่าร์ เมื่อ L=8, a=3 และ s=0 ..........85

2.62 แสดงการทำอินเตอร์ลีฟแบบเซมิแรนด้อม โดยมีค่า L=16 และ S=2 ..........86

3.1 แสดง Trellis ไดอะแกรม ของชุดเข้ารหัสเทอร์โบ [1,05/07] ..........89

3.2 แสดงรูปแบบการอ่านข้อมูลแบบ Block Interleave ..........90

3.3 แสดง Flow chart ในการออกแบบการทำอินเทอร์ลีฟเวอร์แบบ Block ..........91

3.4 แสดงขั้นตอนการสลับข้อมูลแบบ Random ..........92

3.5 แสดงโครงสร้างของระบบ DS-CDMA มาตรฐาน IS-95 ในทิศทาง Reverse Link ..........94

3.6 แสดงโครงสร้างของชุดเข้ารหัสคอนโวลูชั่นที่ใช้ในระบบ DS-CDMA ..........95

3.7 แสดง Flow Chart ในการทำงานของการถอดรหัสข้อมูลแบบ Hard decision Viterbi ..........96

3.8 แสดง Flow Chart ในการทำงานของการถอดรหัสข้อมูลแบบ Soft decision Viterbi ..........97

3.9 แสดงประสิทธิภาพของระบบที่เข้ารหัสข้อมูลแบบคอนโวลูชั่น ความยาวของชุดเข้ารหัสเท่ากับ 9 ที่ใช้อัลกอริทึมในการถอดรหัสแบบ Soft และ แบบ Hard เทียบกับระบบที่ไม่ได้เข้ารหัส ..........98

3.10 แสดงการทำอินเทอร์ลีฟข้อมูลแบบ Block ..........99

3.11 แสดงโครงสร้างของชุด Correlation ทางด้านชุดรับ ..........100

3.12 แสดงโครงสร้างของ LFSR ที่ใช้ในการสร้าง Long PN-Sequence ..........102

3.13 แสดงการนำเอา Long PN Code มาทำการสเปรดข้อมูล ..........103

3.14 แสดง Access Long Code Mask ..........103

3.15 แสดงการคูณข้อมูลด้วย Short PN ..........104

3.16 แสดง วงจร 15-Stage LFSR สำหรับ I ..........104

3.17 แสดง วงจร 15-Stage LFSR สำหรับ Q ..........104



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ (ต่อ)


ภาพที่ หน้า

3.18 แสดงเฟสของ OQPSK ........105

3.19 เฟสของ QPSK ........106

3.20 กราฟแสดงประสิทธิภาพของ QPSK และ OQPSK บนช่องสัญญาณ AWGN ........106

3.21แสดงฮิสโตรแกรมของข้อมูล WGN ........107

3.22 (a) แสดงเอลเวลโลปสัญญาณมัลติพาธ ........108

3.22 (b) แสดงฮิสโตรแกรมของสัญญาณมัลติพาธ ........108

4.1 แสดงประสิทธิภาพของระบบการเข้ารหัสเทอร์โบบนช่องสัญญาณ AWGN data frame=580 บิท ........112

4.2 แสดงประสิทธิภาพของระบบเข้ารหัสแบบเทอร์โบโดยเปลี่ยนแปลงจำนวนรอบ บนช่องสัญญาณ มัลติพาธ เรย์ลี data frame=580 บิท ........113

4.3 แสดงประสิทธิภาพของระบบเข้ารหัสแบบเทอร์โบ เมื่อเปลี่ยนแปลงขนาดของเฟรมข้อมูล บนช่องสัญญาณ AWGN จำนวน Iteration=6 ........114

4.4 แสดงประสิทธิภาพของระบบเข้ารหัสแบบเทอร์โบ เมื่อเปลี่ยนแปลงขนาดของเฟรมข้อมูล บนช่องสัญญาณมัลติพาธ เรย์ลี ........115

4.5 แสดงประสิทธิภาพของระบบเทอร์โบที่ใช้รูปแบบการทำการสลับตำแหน่งข้อมูล แบบ Random และแบบ Block จำนวน Iteration=6 ........116

4.6 แสดงประสิทธิภาพของระบบ DS-CDMA มาตรฐาน IS-95 โดยใช้การเข้ารหัส แบบคอนโวลูชั่น ........117

4.7 แสดงประสิทธิภาพของระบบ DS-CDMA มาตรฐาน IS-95 โดยใช้การเข้ารหัส แบบเทอร์โบ โดยเปลี่ยนแปลงจำนวนรอบของการถอดรหัสข้อมูล ........118

4.8 แสดงประสิทธิภาพของระบบ DS-CDMA เมื่อใช้การเข้ารหัสแบบเทอร์โบ โดยเปลี่ยนแปลงอัตราการส่งข้อมูลที่ 9600 bps บนช่องสัญญาณมัลติพาธเรย์ลี ........119

4.9 แสดงประสิทธิภาพของระบบ DS-CDMA เมื่อใช้การเข้ารหัสแบบเทอร์โบ โดยเปลี่ยนแปลงอัตราการส่งข้อมูลที่ 4800 bps บนช่องสัญญาณมัลติพาธเรย์ลี ........120

4.10 แสดงประสิทธิภาพของระบบ DS-CDMA เมื่อใช้การเข้ารหัสแบบเทอร์โบ โดยเปลี่ยนแปลงอัตราการส่งข้อมูลที่ 2400 bps บนช่องสัญญาณมัลติพาธเรย์ลี ........120

4.11 แสดงประสิทธิภาพของระบบ DS-CDMA มาตรฐาน IS-95 โดยใช้การเข้ารหัส แบบเทอร์โบ เทียบกับคอนโวลูชั่น ........121



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ (ต่อ)


ภาพที่ หน้า

4.12 แสดงประสิทธิภาพของระบบ DS-CDMA เมื่อเปลี่ยนรูปแบบการสลับข้อมูลของ อินเทอร์ลีฟเวอร์ภายนอก แบบ Block และ Random ....122

ก.1 แสดงโครงสร้างภายในของชุดเข้ารหัส แบบ RSC G(D)= [1,05/07] ที่มีค่าความจุภายในเท่ากับ 2 ....130

ก.2 แสดงไดอะแกรม Trellis ของชุดเข้ารหัสแบบ RSC ซึ่งมีค่าเจนเนอร์เรเตอร์ เท่ากับ G(D)= [1,05/07] ที่มีค่าความจุภายในเท่ากับ 2 ....131

ข.1 โครงสร้างของชุดเข้ารหัสแบบ RSC ....141

ข.2 Trellis ไดอะแกรมของวงจรเข้ารหัส RSC ....141



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 1
# บทนำ

## 1.1 ความเป็นมาและความสำคัญของปัญหา

ระบบโทรศัพท์เคลื่อนที่ระบบ DS-CDMA นั้น แต่ละผู้ใช้จะใช้ความถี่คลื่นพาห์ความถี่เดียวกัน โดยแต่ละผู้ใช้ ถูกทำการคูณด้วยข้อมูล PN แบบ Random และแต่ละผู้ใช้จะมีค่าของข้อมูล PN ที่แตกต่างกัน และระดับค่าของพลังงานที่ใช้ในการติดต่อสื่อสารในระบบ DS-CDMA นั้นมีระดับพลังงานที่ต่ำมาก เนื่องมาจากสาเหตุที่ต้องการให้อายุการใช้งานของเครื่องโทรศัพท์มีระยะเวลาที่นาน และด้านความปลอดภัยของผู้ใช้ต่อกำลังส่งของเครื่องโทรศัพท์เคลื่อนที่ จะเห็นว่าเนื่องจากในระบบ DS-CDMA แต่ละผู้ใช้จะใช้ความถี่คลื่นพาห์ความถี่เดียวกัน และกำลังส่งของเครื่องโทรศัพท์เคลื่อนที่ DS-CDMA อยู่ในระดับที่ต่ำ ดังนั้นการเกิดการรบกวนเนื่องจาก การเกิดอินเตอร์เฟอร์เร้นท์ในระบบ DS-CDMA จึงมีโอกาสเกิดการรบกวนสูง ส่งผลกระทบให้ประสิทธิภาพของระบบลดลง ซึ่งวิธีการที่จะเพิ่มประสิทธิภาพให้กับระบบนั้น มีวิธีการอยู่ 2 วิธีด้วยกัน วิธีการแรกใช้เทคนิคในการจัดการทางด้านสายอากาศเข้ามาเพิ่มประสิทธิภาพให้กับระบบ วิธีที่สอง คือวิธีการเพิ่มบิทข้อมูลเข้ามา เพื่อเพิ่มภูมิต้านทานต่อการรบกวนที่จะเข้ามา ซึ่งเป็นสาเหตุทำให้ประสิทธิภาพของระบบลดลง ซึ่งวิธีการนี้ ก็คือวิธีการเข้า-ถอดรหัสข้อมูล มีรูปแบบ เป็นการจัดการเกี่ยวกับบิทข้อมูล ซึ่งในวิทยานิพนธ์ นำวิธีการนี้เข้ามาใช้ในระบบ DS-CDMA เพื่อเพิ่มประสิทธิภาพของระบบให้ดีขึ้น

## 1.2 เหตุผลที่ทำการศึกษา

เหตุผลที่สนใจปัญหาในระบบ DS-CDMA เนื่องจากว่าระบบ DS-CDMA เป็นระบบโทรศัพท์ที่มีการพัฒนาเข้าสู่ยุคของการสื่อสารยุคที่ 3 ในอนาคตจะถูกนำมาใช้แทนที่ระบบโทรศัพท์ที่มีอยู่ในปัจจุบัน เป็นเหตุผลที่ทำให้ผู้วิจัยสนใจที่จะศึกษาเกี่ยวกับการเพิ่มประสิทธิภาพของระบบ DS-CDMA ในส่วนของการลดความผิดพลาดของระบบให้ลดน้อยลง ซึ่งวิธีการที่น่าสนใจคือวิธีการเข้ารหัส-ถอดรหัสข้อมูล และระบบการเข้ารหัสเทอร์โบโค้ด ถือว่าที่มีประสิทธิภาพดีที่สุดในปัจจุบัน(โอกาสเกิดความผิดพลาดของข้อมูลน้อย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.3 หลักการที่นำเสนอ

โดยปรกติโครงสร้างภายในระบบ DS-CDMA มาตรฐาน IS-95[1] นั้น มีชุดเข้ารหัส-ถอดรหัสข้อมูลทางด้านต้นทางและปลายทาง ซึ่งเป็นวิธีการเข้ารหัส-ถอดรหัส แบบ Convolutional Code[2,3] ในวิทยานิพนธ์นี้จะนำเอาการเข้ารหัสแบบ Turbo-Code[4,5,6,7] ซึ่งให้ประสิทธิภาพที่ดีกว่าการเข้ารหัส-ถอดรหัส แบบ Convolutional และจะทำการปรับค่าพารามิเตอร์ให้เหมาะสมเมื่อนำเอาการเข้ารหัสแบบ Turbo-Code เข้ามาแทนระบบของการเข้ารหัสแบบเก่า

## 1.4 การเปรียบเทียบกับหลักการที่มีอยู่เดิม

หลักการเดิมที่มีอยู่จะใช้ระบบของการเข้ารหัสแบบ Convolution มีพารามิเตอร์ของชุดเข้ารหัสแบบ Convolution คือ Constraint length=9 โพลีนอเมียล เจเนอร์เรเตอร์ 378, 1945[8] ส่วนทางด้านชุดรับจะใช้อัลกอริทึมในการถอดรหัสแบบ Viterbi[9,10] ซึ่งเมื่อนำเอาระบบของการเข้ารหัสแบบ Turbo-code เข้ามาแทนระบบของการเข้ารหัสแบบ convolution ทำการเปรียบเทียบประสิทธิภาพทั้งสองระบบ เพื่อแสดงให้เห็นถึงประสิทธิภาพที่ดีขึ้นโดยใช้รูปแบบการเข้ารหัสแบบเดิมเป็นตัวเปรียบเทียบ และจะทำการปรับพารามิเตอร์ของชุดเข้ารหัสแบบ Turbo-code ให้มีความเหมาะสมกับระบบ DS-CDMA

## 1.5 ขอบเขตการวิจัย

วิทยานิพนธ์นี้ แสดงวิธีการเพิ่มประสิทธิภาพ แนวทางหนึ่งในระบบ DS-CDMA มาตรฐาน IS-95 โดยใช้ระบบ การเข้ารหัส-ถอดรหัส เทอร์โบโค้ด แทนวิธีการเข้ารหัสแบบเดิม (Convolution) ซึ่ง มีการเปลี่ยนแปลง พารามิเตอร์ ต่าง ๆ ในระบบ DS-CDMA มาตรฐาน IS-95 โดยแปรเปลี่ยนขนาดของการทำอินเตอร์ลีฟ ภายในระบบ ซึ่ง ในระบบ DS-CDMA เมื่อใช้การเข้ารหัสแบบเทอร์โบ แล้วจะมีการทำอินเตอร์ลีฟข้อมูลสองส่วน คือ ส่วนแรก คือการทำอินเตอร์ลีฟเวอร์ ในระบบการเข้ารหัสเทอร์โบโค้ด ส่วนที่สอง เป็นการทำอินเตอร์ลีฟภายนอก ซึ่ง เป็นการทำอินเตอร์ลีฟเวอร์ข้อมูลที่ผ่านการเข้ารหัสข้อมูลแบบเทอร์โบโค้ด แล้วซึ่งเป็นโครงสร้างเดิมในระบบ DS-CDMA มาตรฐาน IS-95

บทที่ 2 กล่าวถึงมาตรฐานของระบบ DS-CDMA และ พื้นฐานในการเข้ารหัส-ถอดรหัส ซึ่งประกอบด้วย การเข้ารหัสแบบคอนโวลูชั่น อัลกอริทึมในการถอดรหัส ในระบบเข้ารหัสคอนโวลูชั่น การเข้ารหัสเทอร์โบโค้ด ประกอบด้วย ชุดเข้ารหัส แบบคลาสสิค คอนโวลูชั่น สเตจไดอะแกรมของชุดเข้ารหัสแบบคลาสิค คอนโวลูชั่น ชุดเข้ารหัสแบบที่มีการป้อนกลับ และสเตดไดอะแกรม รูปแบบการทำอินเตอร์ลีฟเวอร์แบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 ในบทนี้เราจะจำลองการทำงานระบบ DS-CDMA มาตรฐาน IS-95 ในทิศทางจากโทรศัพท์เคลื่อนที่ไปยังสถานีฐาน โดยจะทำการแบ่งการออกแบบออกเป็นส่วน แต่ละส่วนแสดงขั้นตอนการจำลองการทำงานระบบ โดยอธิบายแสดงขั้นตอนการทำงานของระบบ สุดท้ายจะนำระบบที่ได้ทำการออกแบบแต่ละส่วนมารวมกันเป็นระบบ DS-CDMA มาตรฐาน IS-95 (แบบที่ใช้ชุดเข้ารหัสข้อมูลแบบเดิม ซึ่งเป็นแบบคอนโวลูชั่น และแบบที่ใช้การเข้ารหัสข้อมูลแบบเทอร์โบ ซึ่งเป็นแบบที่ผู้วิจัยได้นำเสนอ) ซึ่งจะใช้ในการวิเคราะห์ประสิทธิภาพของระบบในบทถัดไป

บทที่ 4 กล่าวถึง วิธีการทดลอง ผลการทดลองและวิเคราะห์

บทที่ 5 บทสรุป และ ปัญหาและข้อเสนอแนะในการพัฒนางานวิจัยต่อไป

ในส่วนของภาคผนวก ก แสดงตัวอย่างการคำนวณในการถอดรหัสแบบละเอียด ภาคผนวก ข แสดงตัวอย่างในการคำนวณการถอดรหัสของอัลกอรีทึม MAP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2

# ระบบ DS-CDMA และเทอร์โบโค้ด

## 2.1 พื้นฐานระบบ DS-CDMA [1].

ในบทนี้เราจะกล่าวถึงหลักการของการทำสเปรดสเปกตรัม (spread spectrum) สำหรับการใช้งานในระบบโทรศัพท์เคลื่อนที่ CDMA (Code Division Multiple Access) การทำสเปรดสเปกตรัมในยุคแรก ๆ นั้นมีจุดมุ่งหมายเพื่อนำมาใช้งานในวงการทหารสำหรับป้องกันการรบกวนของข้าศึกต่อการรับส่งสัญญาณ (enemy jamming) หากแต่ในปัจจุบันวิธีการนี้ได้นำมาใช้ประโยชน์ในระบบสื่อสารข้อมูลแบบไร้สาย โดยประโยชน์สำคัญที่ได้รับจากวิธีนี้คือสามารถเพิ่มอัตราการรับส่งข้อมูลให้สูงขึ้น ซึ่งในทางทฤษฎีแล้วมีความเป็นไปได้ที่จะสามารถเพิ่มอัตราการส่งข้อมูลให้สูงกว่าระบบที่มีการใช้งานกันอยู่อย่างแพร่หลายในขณะนี้คือ FDMA และ TDMA

วิธีการทำสเปรดสเปกตรัมสามารถแบ่งออกได้เป็น 2 วิธีหลักคือ

- Direct Sequence
- Frequency Hopping

ทั้งสองวิธีมีความแตกต่างกันโดยหลักการ หากแต่ว่าทั้งคู่อาศัยโค้ดชนิดเดียวกันที่มีชื่อเรียกว่าโค้ด PN (Pseudo-random Noise) เป็นส่วนสำคัญ ดังนั้นจึงมีความจำเป็นที่เราจะต้องเข้าใจถึงแนวทางการสร้างและคุณลักษณะของโค้ด PN ก่อนที่จะกล่าวถึงหลักการของการทำสเปรดสเปกตรัมแบบ Direct Sequence.

โค้ด PN ก็คือ ลำดับสัญญาณ (sequence) ที่ถูกสร้างขึ้นมาโดยวิธีการที่มีลำดับขั้นตอนที่ชัดเจน หากแต่โค้ดหรือลำดับสัญญาณที่ได้มีคุณสมบัติที่คล้ายคลึงกับสัญญาณแรนดอมมาก โค้ดชนิดนี้ถูกนำมาใช้งานในทางปฏิบัติอย่างกว้างขวาง ยกตัวอย่างเช่น นำมาใช้ในการซิงโครไนซ์สัญญาณ (signal synchronization) การเข้ารหัสลับข้อมูล (cryptography) และการทำสเปรดสเปกตรัม (spread spectrum) เป็นต้น การสร้างโค้ดประเภทนี้สามารถกระทำได้หลายวิธี เช่น m-sequence, Gold และ Kasami เป็นต้น วิธีการหนึ่งที่ได้รับความนิยมมากเป็นพิเศษคือ วิธีที่เรียกว่า binary maximal sequence หรือที่เรียกกันทั่วไป binary m-sequence ซึ่งในบทนี้จะกล่าวถึงโค้ดประเภทนี้ในรายละเอียด

### 2.1.1 โค้ด Maximal Sequence

โค้ด maximal-sequence หรือ m-sequence เป็นโค้ดที่สร้างขึ้นจากชิฟต์รีจิสเตอร์จำนวนหนึ่งที่มาต่อเชื่อมกันแบบอนุกรมโดยที่มีการป้อนกลับของสัญญาณจากชิฟต์รีจิสเตอร์อย่างน้อย 2 ตำแหน่งกลับไปยังที่ขาเข้าของชิฟต์รีจิสเตอร์ตัวแรก โดยนิยามแล้วโค้ดชนิดนี้สามารถสร้างลำดับ

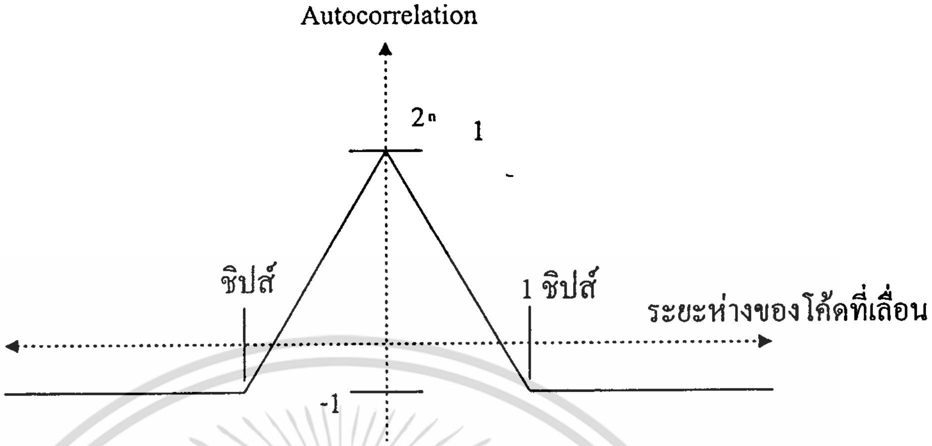
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่มีความยาวมากที่สุดได้คือมีความยาวเท่ากับ $2^n$-1 เมื่อ n คือจำนวนของชิฟต์รีจิสเตอร์ที่ใช้ ชุดโค้ดทุกชุดที่สร้างขึ้นจะมีคุณสมบัติเฉพาะที่เหมือนกันดังต่อไปนี้

- โค้ดหรือลำดับสัญญาณที่ได้จะมีชิปส์ (chips) ที่มีค่าเป็นหนึ่งจำนวนมากกว่าชิปส์ที่มีค่าเป็นศูนย์อยู่หนึ่งชิปส์เสมอ เช่น ชุดโค้ดที่ใช้ชิฟต์รีจิสเตอร์จำนวน 10 ตัว ให้ลำดับสัญญาณที่มีความยาวเท่ากับ $2^{10}$-1 = 1,023 ชิปส์ จะประกอบด้วยชิปส์ที่มีค่าเป็นศูนย์จำนวน 511 ชิปส์ และชิปส์ที่มีค่าเป็นหนึ่งจำนวน 512 ชิปส์
- ภายในลำดับสัญญาณแต่ละชุดจะมีช่วงของชิปส์ที่มีค่าเป็น ศูนย์หรือหนึ่ง ต่อเนื่องกันจำนวนหนึ่ง ส่วนที่ต่อเนื่องกันนี้จะเรียกว่า รัน (run) โดยความยาวของแต่ละรันจะมีขนาดที่แตกต่างกันไป สำหรับโค้ด m-sequence จะมีคุณสมบัติของรันที่พิเศษคือ จะมีรันของศูนย์และหนึ่งขนาด P ชิปส์เกิดขึ้นในลำดับสัญญาณทั้งสิ้น $2^{n-(p+2)}$ ครั้ง โดย P มีค่าอยู่ระหว่าง 1 ถึง n-2 เมื่อ n คือจำนวนของชิฟต์รีจิสเตอร์ที่ใช้ในการสร้างลำดับสัญญาณ นอกจากนี้จะมีรันของศูนย์ขนาด n-1 จำนวน 1 รัน และรันของหนึ่งขนาด n จำนวน 1 รันเสมอ
- อัตสหสัมพันธ์ (autocorrelation) ของชุดโค้ด m-sequence จะมีค่าเท่ากับ –1 เสมอหากมีการเลื่อนของชุดโค้ดทั้งสองชุดที่เหมือนกันทุกประการให้มีขนาดห่างกันมากกว่า ±1 ชิปส์ แต่ถ้าหากการเลื่อนชุดโค้ดทั้งสองอยู่ในช่วง 0±1 ค่าของอัตสหสัมพันธ์จะแปรเปลี่ยนไปตามระยะเลื่อนไปของชุดโค้ดทั้งสองอย่างเชิงเส้นโดยจะมีค่าอยู่ระหว่าง –1 ถึง $2^n$-1 ดูรูปที่ 2.1 ประกอบ หมายเหตุในการหาค่าอัตสหสัมพันธ์สามารถกระทำได้โดยนำค่าของชิปส์จากลำดับสัญญาณ 2 ชุดที่ตำแหน่งเดียวกันมาคูณกัน แล้วนำผลคูณที่ได้ของแต่ละตำแหน่งมาบวกรวมกันทั้งหมด ผลที่ได้ก็คือค่าอัตสหสัมพันธ์ ทั้งนี้ในการคูณกันนั้นจะกำหนดให้ใช้ –1 แทนชิปส์ที่มีค่าเป็นศูนย์และหนึ่ง แทนชิปส์ที่มีค่าเป็นหนึ่ง ในการหาค่าอัตสหสัมพันธ์นั้นจะต้องใช้ลำดับสัญญาณ 2 ชุดที่เหมือนกัน แต่อาจมีการเลื่อนชิปส์ของลำดับสัญญาณชุดหนึ่งให้ต่างจากลำดับสัญญาณของอีกชุดหนึ่งด้วยระยะทางต่าง ๆ ตามต้องการ
- เมื่อนำโค้ด m-sequence ชุดหนึ่งมาบวกมอดูโล 2 กับโค้ดชุดเดิมที่มีการเลื่อนชิปส์ไป จะได้โค้ดชุดใหม่ที่มีคุณลักษณะเดียวกันกับโค้ดชุดเดิมแต่มีการเลื่อนของชิปส์ไปด้วยจำนวนที่แตกต่างจากโค้ดเริ่มต้นทั้งสองชุด
- หากพิจารณาค่าของชิฟต์รีจิสเตอร์แต่ละตัว ณ จังหวะเวลาใดเวลาหนึ่ง จะพบว่าค่าหรือสถานะของชิฟต์รีจิสเตอร์เหล่านี้จะมีรูปแบบที่แตกต่างไปจากที่จังหวะเวลาอื่น ๆ ทั้งหมดเมื่อพิจารณาภายในช่วงเวลา 1 คาบของลำดับสัญญาณ นั่นคือรูปแบบสถานะของชิฟต์รีจิสเตอร์เหล่านี้จะไม่มีการเกิดซ้ำภายใน 1 คาบสัญญาณ อีกทั้งจะไม่เกิดกรณีที่สถานะของชิฟต์รีจิสเตอร์ทุกตัวมีค่าเป็นศูนย์หมด เพราะในสถานะดังกล่าววงจรชิฟต์รีจิสเตอร์จะไม่สามารถสร้างลำดับสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขึ้นได้ ดังนั้นในการสร้างลำดับสัญญาณนั้นจะต้องแน่ใจว่าค่าตั้งต้นของชิฟต์รีจิสเตอร์ทุกตัวมีค่าที่ไม่เป็นศูนย์พร้อมกันหมด




**รูปที่ 2.1** ค่าอัตสหสัมพันธ์ของโค้ด m-sequence ที่มีการเลื่อนไปของโค้ดที่ค่าต่าง ๆ

คุณสมบัติต่าง ๆ ที่กล่าวมานี้ทำให้โค้ดประเภทนี้มีประโยชน์อย่างมากในทางปฏิบัติ เช่น การที่โค้ดมีจำนวนชิปส์ที่เป็นศูนย์ใกล้เคียงกับชิปส์ที่เป็นหนึ่ง ทำให้เวลานำมาใช้งานกับระบบสื่อสารบางประเภทเช่นระบบที่มอดูเลตสัญญาณ โดยมีการ Suppress carrier จะไม่มีปัญหาในเรื่องขององค์ประกอบไฟฟ้ากระแสตรงเกิดขึ้น สังเกตว่าโค้ดที่มีความยาวมากขึ้น ปัญหาในเรื่องดังกล่าวก็จะลดลงตามไปด้วย

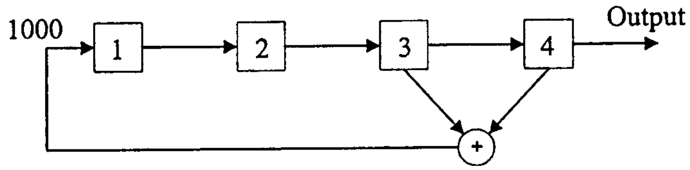
จากรูปแบบและลักษณะการเกิดของรันที่ได้กล่าวมาข้างต้นนั้น ได้มีการพิสูจน์และพบว่าโค้ดประเภทนี้มีคุณสมบัติที่เหมือนกับสัญญาณแรนดอมจริง ๆ ทั้ง ๆ ที่โค้ด m-sequence เป็นสัญญาณที่มีลักษณะเป็นรายคาบคือจะมีรูปแบบที่ซ้ำเดิมเมื่อครบรอบหนึ่งคาบ

คุณสมบัติอัตสหสัมพันธ์ของโค้ด m-sequence ก็มีประโยชน์อย่างมากกับระบบสื่อสารโทรศัพท์เคลื่อนที่ CDMA เพราะลำดับสัญญาณชุดหนึ่งที่มีคาบยาวมากเพียงพอสามารถนำมาใช้ในการระบุหมายเลขประจำตัวของสถานีฐานได้เป็นจำนวนมากโดยอาศัยการเลื่อนชิปส์ของลำดับสัญญาณไปด้วยจำนวนที่แตกต่างกัน เช่นในมาตรฐานระบบโทรศัพท์เคลื่อนที่ CDMA (IS-95) ใช้ชิฟต์รีจิสเตอร์จำนวน 15 ตัวในการสร้างโค้ดที่มีคาบความยาวเท่ากับ $2^{15}-1 = 32,768$ ชิปส์ สถานีฐานแต่ละแห่งจะใช้วงจรสร้างโค้ดแบบเดียวกันแต่จะมีการเลื่อนชิปส์ของลำดับสัญญาณออกไปทีละ 64 ชิปส์ เพราะฉะนั้นก็จะมีโค้ดที่แตกต่างกันสำหรับการใช้งานได้ทั้งหมด 32,768/64 = 512 ชุด สำหรับใช้ระบุหมายเลขประจำตัวของสถานีฐานได้มากถึง 512 แห่ง

เช่น รูปข้างล่างแสดงวงจรกำเนิดโค้ด m-sequence โดยใช้ชิฟต์รีจิสเตอร์ 4 ตัว (n=4) และวงจรบวกเลขฐานสอง ซึ่งทำหน้าที่รวมสัญญาณจากชิฟต์รีจิสเตอร์ตัวที่ 3 และตัวที่ 4 จำนวน 1 ชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรนี้จะสามารถสร้างลำดับสัญญาณที่มีความยาวเท่ากับ $2^n-1 = 15$ ชิปส์ ถ้าหากตั้งค่าเริ่มต้นของชิฟต์รีจิสเตอร์ด้วยค่า 1000 ผลที่ได้จะเป็นลำดับสัญญาณที่มีค่าเป็น 000100110101111



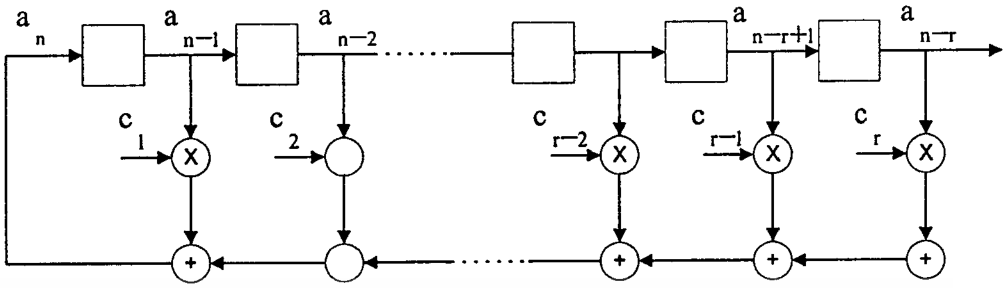

จากชุดโค้ดที่ได้ สามารถนำมาวิเคราะห์คุณสมบัติต่าง ๆ ได้ดังนี้คือ

- ชุดโค้ดมีชิปส์ที่มีค่าเป็นศูนย์ทั้งหมด 7 ตัวและชิปส์ที่มีค่าเป็นหนึ่ง 8 ตัว นั่นคือจำนวนศูนย์และหนึ่งของโค้ดมีค่าต่างกันเพียงแค่หนึ่ง
- ชุดโค้ดประกอบด้วยรันของศูนย์จำนวน 4 รัน และรันของหนึ่งอีก 4 รัน โดยรันที่มีความยาวเท่ากับ 1 ของที่เป็นศูนย์และหนึ่งมีทั้งหมดอย่างละ 2 รัน หรือเทียบเท่ากับ ½ ของจำนวนรันทั้งหมด รันที่มีความยาวเท่ากับ 2 ของที่เป็นศูนย์และหนึ่งมีทั้งหมดอย่างละ 1 รัน หรือเทียบเท่ากับ ¼ ของจำนวนรันทั้งหมด รันของศูนย์ที่มีความยาวเท่ากับ 3 จำนวน 1 รัน และรันของหนึ่งที่มีความยาวเท่ากับ 4 จำนวน 1 รัน
- ค่าอัตสหสัมพันธ์ของชุดโค้ดมีคุณลักษณะดังที่แสดงในตารางข้างล่าง

| จำนวนชิปส์ที่เลื่อนจากกัน | ลำดับสัญญาณ | อัตสหสัมพันธ์ |
|---|---|---|
| 0 | 000100110101111 | 15 |
| 1 | 100010011010111 | -1 |
| 2 | 110001001101011 | -1 |
| 3 | 111000100110101 | -1 |
| 4 | 111100010011010 | -1 |
| 5 | 011110001001101 | -1 |
| 6 | 101111000100110 | -1 |
| 7 | 010111100010011 | -1 |
| 8 | 101011110001001 | -1 |
| 9 | 110101111000100 | -1 |
| 10 | 011010111100010 | -1 |
| 11 | 001101011110001 | -1 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.2 การสร้างโค้ด m-sequence




รูปที่ 2.2 โครงสร้างทั่วไปของวงจรสร้างโค้ด m-sequence

พิจารณาลักษณะโครงสร้างของวงจรสร้างโค้ด m-sequence ที่ประกอบด้วยชิฟต์รีจิสเตอร์จำนวน r ตัวและวงจรบวกสัญญาณแบบมอดูโล 2 อีกจำนวนหนึ่งในรูปที่ 2.2 จากในรูปจะสามารถเขียนความสัมพันธ์ของ $a_n$ กับสัญญาณค่าอื่น ๆ ได้เป็น

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \ldots + c_{r-1} a_{n-r+1} + c_r a_{n-r} = \sum_{i=1}^{r} c_i a_{n-i} \qquad (2.1)$$

ตัวแปรต่าง ๆ ในสมการนี้มีค่าได้เพียง 2 รูปแบบคือ 0 หรือ 1 โดย $c_i$ $(i = 1,2,\ldots,r)$ มีค่าเป็นศูนย์ เมื่อไม่มีการป้อนกลับของสัญญาณจากชิฟต์รีจิสเตอร์ตัวที่ i กลับไปที่ขาเข้าของชิฟต์รีจิสเตอร์ตัวแรก และมีค่าเป็น 1 เมื่อมีการป้อนกลับของสัญญาณที่ตำแหน่งดังกล่าว ทั้งนี้ $c_r$ จะต้องมีค่าเป็น 1 เสมอ ส่วนของวงจรบวกจะเป็นการบวกแบบมอดูโล 2 นั่นคือ 0+1 = 1+0 = 1 และ 0+0 = 1+1 = 0 ซึ่งก็คือการทำ Exclusive-OR นั่นเอง

สมมติให้ลำดับสัญญาณจากวงจรสร้างโค้ด m-sequence สามารถเขียนอยู่ในรูปของ generating function ได้เป็น

$$G(D) = a_0 + a_1 D + a_2 D^2 + \ldots = \sum_{n=0}^{\infty} a_n D^n \qquad (2.2)$$

โดย D คือการหน่วงเวลาไปหนึ่งหน่วยหรือหนึ่งจังหวะเวลา ส่วนค่ายกกำลังของ D คือจำนวนหน่วยเวลาที่มีการหน่วงเวลาไป

จากสมการ 2.1 และ 2.2 เราสามารถเขียนสมการ 2.3 ได้เป็น

$$\begin{aligned} G(D) &= \sum_{n=0}^{\infty} a_n D^n = \sum_{n=0}^{\infty}\sum_{i=1}^{r} c_i a_{n-i} D^n \\ &= \sum_{i=1}^{r} c_i D^i \left[ \sum_{n=0}^{\infty} a_{n-i} D^{n-i} \right] \\ &= \sum_{i=1}^{r} c_i D^i [a_{-i} D^{-i} + \ldots + a_{-1} D^{-1} + G(D)] \end{aligned} \qquad (2.3)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ 2.3 ย้ายข้างของ G(D) ใหม่ได้เป็น

$$G(D)\left(1-\sum_{i=1}^{r}c_iD^i\right)=\sum_{i=1}^{r}c_iD^i(a_{-i}D^{-i}+...+a_{-1}D^{-1})$$

หรือ

$$G(D)=\sum_{i=1}^{r}c_iD^i(a_{-i}D^i+...+a_{-1}D^{-1})/\left(1-\sum_{i=1}^{r}c_iD^i\right)=g_0(D)/f(D) \qquad (2.4)$$

โดย $f(D)=\left(1-\sum_{i=1}^{r}c_iD^i\right)$ เป็นค่าที่เรียกว่า characteristic polynomial ของวงจรชิฟต์รีจิสเตอร์ที่ใช้ โดยค่านี้จะขึ้นอยู่กับค่า $c_0$, $c_1$, …, $c_r$ เท่านั้น ซึ่งเป็นตัวกำหนดว่าภายในวงจรจะมีการป้อนกลับของสัญญาณจากชิฟต์รีจิสเตอร์ตัวใดบ้าง สำหรับ $g_0(D)$ เป็นส่วนที่ขึ้นอยู่กับค่าเริ่มต้นของชิฟต์รีจิสเตอร์แต่ละตัว $a_{-r}$, $a_{-r+1}$, …, $a_{-1}$ ก่อนที่ค่าแรก $a_0$ ของลำดับสัญญาณจะโผล่ออกมา เราสามารถเขียน $g_0(D)$ ใหม่ได้เป็น

$$\begin{aligned} g_0(D) &= \sum_{i=1}^{r}c_i(a_{-i}+a_{-i+1}D+...+a_{-1}D^{i-1}) \\ &= c_1a_{-1} \\ &\quad +c_2(a_{-2}+a_{-1}D) \\ &= \quad +c_3(a_{-3}+a_{-2}D+a_{-1}D^2) \\ &\quad +... \\ &\quad +c_r(a_{-r}+a_{-r+1}D+...+a_{-1}D^{r-1}) \end{aligned} \qquad (2.5)$$

ในขั้นต้นนี้เราจะพิจารณากรณีพิเศษคือสมมติให้

$a_{-r}=1$ และ $a_{-r+1}=a_{-r+2}=...=a_{-2}=a_{-1}=0$

และจากนั้นจะขยายการอธิบายให้ครอบคลุมในกรณีทั่วไปภายหลัง สำหรับกรณีพิเศษนี้สมการ 2.5 จะเขียนได้เป็น

$$g_0(D)=1, G(D)=1/f(D) \qquad (2.6)$$

จากสมการที่ 2.4 และ 2.6 เราสามารถพิสูจน์ถึงคุณสมบัติที่สำคัญ 3 อย่างดังต่อไปนี้

**คุณสมบัติที่ 1 :** ลำดับสัญญาณที่สร้างขึ้นจะเป็นสัญญาณรายคาบโดยที่มีคาบความยาวเป็น

$$P\le 2^r-1 \qquad (2:7)$$

คุณสมบัตินี้สามารถพิสูจน์ได้จากการที่เราทราบว่าค่าของชิฟต์รีจิสเตอร์ทั้งหมด r ตัวจะมีรูปแบบที่เป็นไปได้ทั้งหมดเท่ากับ $2^r$-1 แบบ และมีรูปแบบอีกหนึ่งรูปแบบที่ทุกตัวมีค่าเป็นศูนย์ทั้งหมด สำหรับกรณีที่ชิฟต์รีจิสเตอร์มีค่าเป็นศูนย์ทั้งหมดคาบของลำดับสัญญาณจะมีค่าเป็น 1 คือชิฟต์รี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จิสเตอร์ทุกตัวก็ยังคงจะมีค่าเป็นศูนย์ไปตลอด ส่วนในกรณีที่เหลือนั้นรูปแบบของค่ารีจิสเตอร์ทั้งหมดจะต้องมีการซ้ำเกิดขึ้นหลังจากเวลาผ่านไป $2^r$-1 จังหวะเวลาอย่างแน่นอน เมื่อเกิดการซ้ำกันขึ้นของรูปแบบของค่าในรีจิสเตอร์หลังจากที่ผ่านไป $P \leq 2^r - 1$ จังหวะเวลา ค่าต่าง ๆ เหล่านี้ก็จะมีการซ้ำกันเกิดขึ้นอีกในทุก ๆ ช่วงเวลา P เพราะจากสมการที่ 2.4 จะเห็นว่าลำดับการเกิดขึ้นของค่าในรีจิสเตอร์ในแต่ละจังหวะเวลาจะถูกกำหนดจากค่าเริ่มต้น $g_0(D)$ เท่านั้น เพราะฉะนั้นจึงสรุปได้ว่าลำดับสัญญาณที่ได้จะเป็นสัญญาณรายคาบที่มีคาบความยาวเท่ากับ $P \leq 2^r - 1$

**คุณสมบัติที่ 2 :** คาบความยาวของ G(D) คือเลขจำนวนเต็ม P ที่เล็กที่สุดจะมีค่ามากกว่าศูนย์ โดยที่ f(D) สามารถหาร $1\text{-}D^p$ ลงตัว ทั้งนี้ $g_0(D)$ และ f(D) ต้องไม่มีตัวประกอบร่วมกัน

ในการพิสูจน์คุณสมบัติข้อนี้จะเริ่มจากกรณีที่ง่ายก่อนคือกรณีที่ $g_0(D) = 1$ จากนั้นจึงจะขยายการพิสูจน์ให้ครอบคลุมทุก ๆ กรณี ในการพิสูจน์นั้นเราแสดงให้เห็นว่า ถ้า G(D) มีคาบความยาวเท่ากับ P แล้ว f(D) จะต้องหาร $1\text{-}D^p$ ลงตัวเสมอ และในทางกลับกันถ้า f(D) หาร $1\text{-}D^p$ ลงตัวแล้ว G(D) ก็จะต้องมีคาบความยาวเท่ากับ P ด้วย ถ้าหากว่า G(D) มีคาบความยาวเท่ากับ P และ $g_0(D) = 1$ แล้วเราจะสามารถเขียนสมการแสดงความสัมพันธ์ได้เป็น

$$\frac{1}{f(D)} = G(D) = (a_0 + a_1 D + \dots + a_{p-1} D^{p-1}$$
$$+ D^p (a_0 + a_1 D + \dots + a_{p-1} D^{p-1})$$
$$+ D^{2p} (a_0 + a_1 D + \dots + a_{p-1} D^{p-1}) + \dots$$
$$= \frac{1}{1 - D^p} (a_0 + a_1 D + \dots + a_{p-1} D^{p-1})$$

ดังนั้น

$$\frac{1 - D^p}{f(D)} = (a_0 + a_1 D + \dots + a_{p-1} D^{p-1})$$

ซึ่งแสดงว่า f(D) หาร $1\text{-}D^p$ ลงตัว

ในทางกลับกันถ้า f(D) หาร $1\text{-}D^p$ ลงตัวก็สามารถเขียนเป็นสมการได้เป็น

$$\frac{1 - D^p}{f(D)} = (a_0 + a_1 D + \dots + a_{p-1} D^{p-1})$$

เนื่องจาก $g_0(D) = 1$ ดังนั้น

$$G(D) = \frac{1}{f(D)} = \frac{1}{1 - D^p} (a_0 + a_1 D + \dots + a_{p-1} D^{p-1})$$
$$= (1 + D^p + D^{2p} + \dots)(a_0 + a_1 D + \dots + a_{p-1} D^{p-1})$$

สมการนี้แสดงให้เห็นว่า G(D) เป็นสัญญาณรายคาบที่มีคาบความยาวเท่ากับ P

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับกรณีทั่วไปที่ $g_0(D)$ มีค่าเป็นเท่าใดก็ได้นั้น ก็สามารถพิสูจน์ได้เช่นกัน จากที่ถ้า G(D) มีคาบความยาวเท่ากับ P แล้วจะเขียนความสัมพันธ์ได้ดังนี้

$$\frac{g_0(D)}{f(D)} = G(D) = \frac{1}{1-D^p}(a_0 + a_1 D + \ldots + a_{p-1} D^{p-1})$$

**ซึ่งเขียนใหม่ได้เป็น**

$$g_0(D)(1-D^p) = (a_0 + a_1 D + \ldots + a_{p-1} D^{p-1}) f(D)$$

**จากเงื่อนไขที่ $g_0(D)$ และ f(D) จะต้องไม่มีตัวประกอบร่วมกัน ดังนั้นสมการข้างต้นจะเป็นจริงได้ก็** ต่อเมื่อ f(D) ต้องสามารถหาร $(1\text{-}D^p)$ ลงตัว

สำหรับการพิจารณาในทางกลับกันคือถ้า f(D) สามารถหาร $(1\text{-}D^p)$ ลงตัวแล้ว จะได้ว่าใน กรณีที่ $g_0(D) = 1$ สมการความสัมพันธ์ที่ได้คือ

$$\frac{1}{f(D)} = G(D) = (a_0 + a_1 D + \ldots + a_{p-1} D^{p-1})(1 + D^p + D^{2p} + \ldots)$$

แต่สำหรับกรณีที่ $g_0(D)$ มีค่าที่ต่างไป คือ $g_0(D) = g_0 + g_1 D + \ldots + g_{r-1} D_{r-1}$ โดยที่ $g_k = 1$ อย่างน้อยหนึ่ง ค่าของ k ที่ไม่เท่ากับ 0 เราจะสามารถเขียนสมการของ G(D) ได้ในรูปต่อไปนี้

$$\begin{aligned} G(D) = \frac{g_0(D)}{f(D)} &= g_0(a_0 + a_1 D + \ldots + a_{p-1} D^{p-1})(1 + D^p + D^{2p} + \ldots) \\ &+ g_1 D(a_0 + a_1 D + \ldots + a_{p-1} D^{p-1})(1 + D^p + D^{2p} + \ldots) \\ &+ \ldots\ldots \\ &+ g_{r-1} D^{r-1}(a_0 + a_1 D + \ldots + a_{p-1} D^{p-1})(1 + D^p + D^{2p} + \ldots) \end{aligned}$$

จากสมการที่ได้จะเห็นว่าในแต่ละเทอมนั้นเป็นสัญญาณรายคาบที่มีคาบความยาวเท่ากับ P ซึ่งเมื่อ รวมสัญญาณคาบแต่ละเทอมเข้าด้วยกันแล้วก็จะได้สัญญาณรายคาบที่มีคาบความยาวเท่าเดิมคือ เท่ากับ P สังเกตว่าถ้าหาก f(D) สามารถแยกออกเป็นตัวประกอบย่อยได้แล้ว อาจจะเกิดกรณีที่ $g_0(D)$ บางรูปแบบมีตัวประกอบย่อยแบบเดียวกันกับ f(D) ในกรณีแบบนี้จะทำให้คาบความยาวที่ได้มี ขนาดเล็กลงนั่นคือ $P \leq 2^r - 1$

**คุณสมบัติที่ 3 :** เงื่อนไขที่จำเป็นในการที่วงจรกำเนิดลำดับสัญญาณจะสามารถให้สัญญาณ ที่มีคาบความยาวสูงสุดเท่ากับ $P = 2^r - 1$ คือ f(D) ต้องไม่สามารถแยกออกเป็นตัวประกอบย่อยได้ อีก (irreducible)

ในการพิสูจน์คุณสมบัติข้อนี้ เราจะสมมติให้ $g_0(D) = 1$ ได้โดยที่ไม่ส่งผลต่อการพิสูจน์ สำหรับกรณีทั่วไปเพราะหากลำดับสัญญาณที่ได้มีคาบความยาวเท่ากับ $P = 2^r - 1$ แล้ว เมื่อเวลา ผ่านไปช่วงเวลาหนึ่งค่าของรีจิสเตอร์แบบต่าง ๆ ที่เป็นไปได้ทั้งหมดจะต้องเกิดขึ้นอย่างแน่นอน และ ณ จังหวะเวลานั้นเราสามารถกำหนดให้รูปแบบนั้นเป็นค่าตั้งต้น $g_0(D)$ ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าหาก f(D) สามารถนำมาแยกตัวประกอบได้เป็น s(D) และ t(D) ที่มีดีกรีเป็น $r_s$ และ $r_t$ ตามลำดับโดยที่ $r_s \geq 1$ และ $r_t \geq 1$และ $r = r_s + r_t$ เมื่อทำการแยกตัวประกอบของ 1/f(D) ก็จะได้

$$G(D) = \frac{1}{f(D)} = \frac{\alpha(D)}{s(D)} + \frac{\beta(D)}{t(D)}$$

โดย

$$\text{period}\left[\frac{\alpha(D)}{s(D)}\right] \leq 2^{r_s} - 1, \text{period}\left[\frac{\beta(D)}{t(D)}\right] \leq 2^{r_t} - 1$$

เพราะฉะนั้น

$$\text{period}[G(D)] \leq \text{period}\left[\frac{\alpha(D)}{s(D)}\right].\text{period}\left[\frac{\beta(D)}{t(D)}\right]$$

$$\leq (2^{r_s} - 1)(2^{r_t} - 1) = 2^r - 2^{r_s} - 2^{r_t} + 1 \leq 2^r - 3$$

จากผลลัพธ์ที่ได้จะเห็นว่าขัดกันกับที่ได้ระบุไว้ข้างต้น เพราะฉะนั้น f(D) จะต้องมีคุณสมบัติที่ไม่สามารถแยกตัวประกอบได้ (irreducible)

การที่ f(D) มีคุณสมบัติดังกล่าวเพียงอย่างเดียวมิได้หมายความว่าคาบของสัญญาณที่ได้จะต้องมีขนาดสูงสุดเท่ากับ $P = 2^r - 1$ เสมอไป ยกตัวอย่างเช่น ในกรณี r = 4 ซึ่งควรจะให้คาบสัญญาณที่มีความยาวสูงสุดเท่ากับ $P = 2^4 - 1 = 15$ แต่เมื่อพิจารณา $f(D) = 1+D+D^2+D^3+D^4$ ซึ่งไม่สามารถแยกตัวประกอบย่อยได้อีก กลับพบว่าลำดับสัญญาณที่ได้มีคาบความยาวเท่ากับ 5 เท่านั้น ทั้งนี้ก็เพราะ f(D) สามารถหาร $1\text{-}D^5$ ลงตัว หากต้องการให้ได้ลำดับสัญญาณที่มีคาบความยาวเท่ากับ 15 ก็จะต้องใช้ $f(D) = 1+D+D^4$ ชุด f(D) นี้หาร $1\text{-}D^{15}$ ลงตัวแต่จะหาร $1\text{-}D^k$ สำหรับ k < 15 ไม่ลงตัว

ชุดโพลิโนเมียล f(D) ที่สามารถให้กำเนิดลำดับสัญญาณที่มีคาบความยาวเท่ากับ $P = 2^r - 1$ ได้เรียกว่า primitive ซึ่งชุดโพลิโนเมียลที่มีคุณสมบัติพิเศษนี้ได้มีการค้นพบและจัดทำเป็นตารางสำหรับใช้งานไว้แล้ว อีกทั้งเราสามารถคำนวณหาจำนวนโพลิโนเมียลที่มีคุณสมบัติ primitive ได้โดยอาศัยสูตรง่าย ๆ ต่อไปนี้

$$N_p(r) = \frac{2^r - 1}{r}\prod_{i=1}^{J}\frac{P_i - 1}{P_i} \tag{2.8}$$

ทั้งนี้ $\{P_i, i = 1,2,....,J\}$ คือตัวประกอบที่เป็นจำนวนเฉพาะของ $2^r\text{-}1$ นั่นคือ

$$2^r - 1 = \prod_{i=1}^{J} P_i^{e_i}$$

โดย $e_i$ คือตัวเลขจำนวนเต็มค่าหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยกตัวอย่างในกรณีที่ r=12 ถึง 12 จะได้ผลดังนี้

| R | $2^r$-1 | J | $N_p(r)$ |
|---|---|---|---|
| 2 | 3 | 1 | 3/2•2/3=1 |
| 3 | 7 | 1 | 7/3•6/7=2 |
| 4 | 3•5 | 2 | 15/4•4/5•2/3=2 |
| 5 | 31 | 1 | 31/5•30/31=6 |
| 6 | $3^2$•7 | 2 | 63/6•2/3•6/7=6 |
| 7 | 127 | 1 | 127/7•126/127=18 |
| 8 | 3•5•17 | 3 | 255/8•2/3•4/5•16/17=16 |
| 9 | 7•73 | 2 | 511/9•6/7•72/73=48 |
| 10 | 3•11•31 | 3 | 1023/10•2/3•10/11•30/31=60 |
| 11 | 23•89 | 2 | 2047/11•22/23•88/89=176 |
| 12 | $3^2$•5•7•13 | 4 | 4095/12•2/3•4/5•6/7•12/13=48 |

**2.1.3 การสเปรดสเปกตรัมแบบ Direct Sequence**

ในส่วนแรกนี้เราจะอธิบายถึงหลักการพื้นฐานของวิธีการทำสเปรดสเปกตรัมแบบ Direct Sequence เพื่อการใช้งานในระบบ (CDMA) โดยเราจะยกตัวอย่างง่าย ๆ สำหรับใช้ในการบรรยายถึงกรรมวิธีการทำสเปรดสเปกตรัมในแต่ละขั้นตอน สมมติว่าในระบบที่เราสนใจมีจำนวนผู้ใช้ทั้งหมด 3 คน ซึ่งแต่ละคนก็จะได้รับชุดโค้ด PN ที่แตกต่างกัน ก่อนที่ผู้ใช้แต่ละคนจะส่งข้อมูลออกจะนำบิทข้อมูลเหล่านั้นไปทำการสเปรดสเปกตรัมโดยการนำบิทข้อมูลไปคูณกับชุดโค้ด PN ของตนเอง ซึ่งจะทำให้อัตราบิทมีขนาดสูงขึ้น หมายเหตุ : โดยทั่วไปอัตราบิทหลังจากการทำสเปรดสเปกตรัมแล้วจะเรียกว่าอัตราชิปส์ (chip rate) ดังที่แสดงในรูปที่ 2.3 ซึ่งในตัวอย่างนี้อัตราชิปส์มีขนาดเพิ่มขึ้นเป็น 4 เท่าตัวของอัตราบิทข้อมูลของผู้ใช้ สำหรับตัวอย่างนี้เราสมมติให้ผู้ใช้หมายเลข 1, 2 และ 3 ต้องการส่งข้อมูล 11, 10 และ 01 ตามลำดับ สัญญาณที่ผ่านการสเปรดสเปกตรัมแล้วจะมีลักษณะดังที่แสดงในรูปที่ 2.3 สังเกตว่าสัญญาณ 1 จะแทนด้วยแรงดันเป็น 1 และสัญญาณ 0 จะแทนด้วยแรงดันเป็น –1 สัญญาณที่ทำการสเปรดแล้วทั้ง 3 สัญญาณนี้จะถูกส่งออกไปพร้อมกันบนคลื่นความถี่เดียวกัน นั่นหมายความว่าที่ภาครับจะได้รับสัญญาณที่เป็นผลรวมของ 3 สัญญาณรวมกันซึ่งดูเสมือนว่าสัญญาณเหล่านี้รบกวนซึ่งกันและกันอยู่และยากต่อการแยกแยะออกมา ดูรูปที่ 2.3 ประกอบ

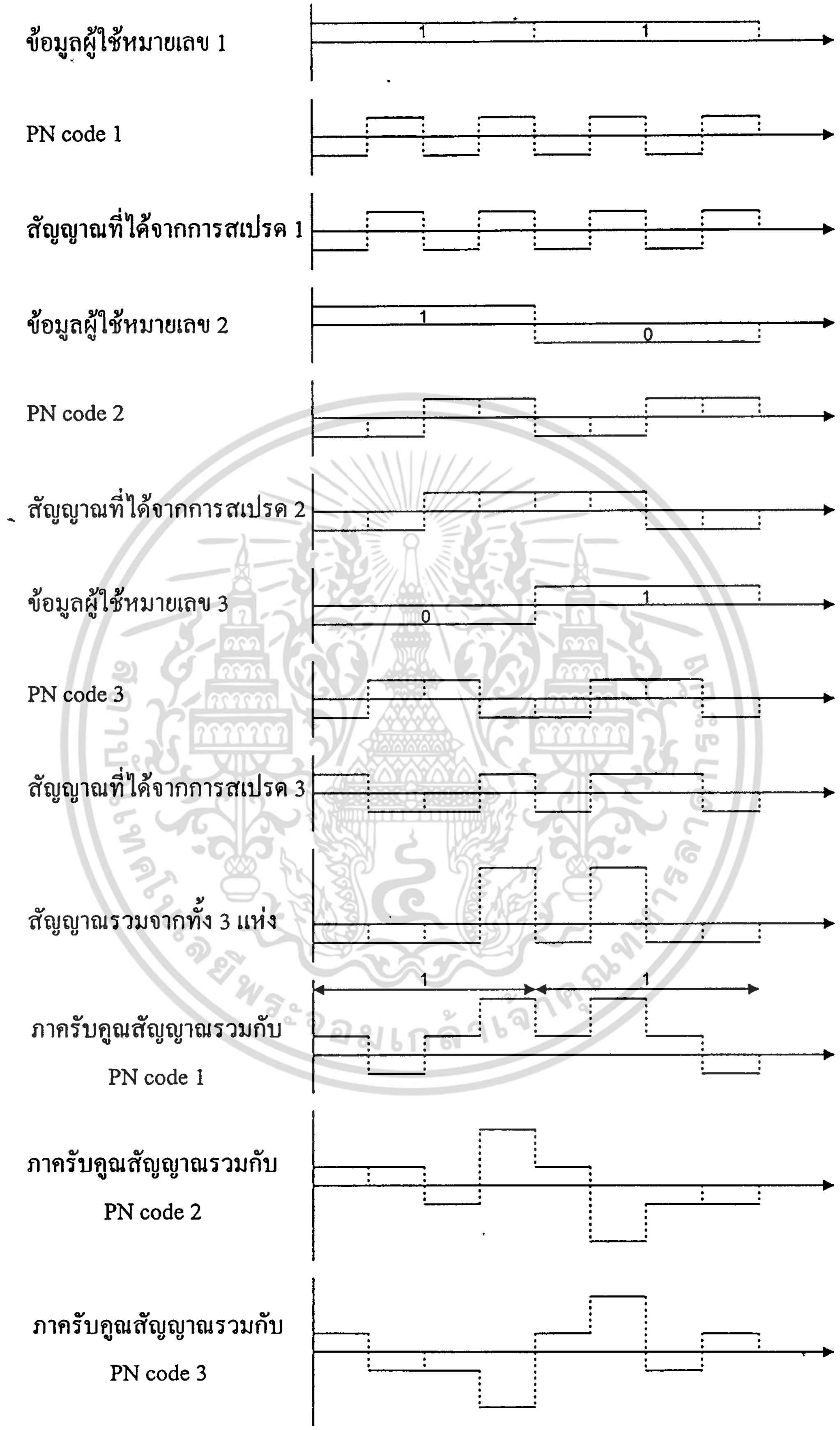
สำหรับขั้นตอนในการแยกสัญญาณของแต่ละคนออกมานั้นก็สามารถทำได้โดยการนำชุดโค้ด PN ของตนเองไปคูณกับสัญญาณที่รับได้และทำการบวกผลคูณเหล่านั้นทีละ 4 บิต หรือเท่ากับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความยาวของหนึ่งบิทข้อมูลนั่นเอง ค่าที่คำนวณได้เหล่านี้จะนำไปใช้ในการตัดสินใจต่อไป ยกตัวอย่างเช่นถ้าเรานำโค้ดของผู้ใช้หมายเลข 1 ไปคูณกับสัญญาณรวม จะได้ผลรวมของผลคูณออกมาเป็น 4 สำหรับข้อมูลทั้งสองบิท ในขณะที่ถ้าผู้ใช้หมายเลข 2 กระทำการในลักษณะเดียวกันจะได้ผลออกมาเป็น 4 และ –4 จะเห็นว่าถ้าผลลัพธ์ที่ออกมามีค่าเท่ากับ 4 แสดงว่าบิทข้อมูลที่ส่งออกมามีค่าเป็น 1 และถ้าผลลัพธ์ที่ได้เป็น –4 แสดงว่าบิทข้อมูลนั้นมีค่าเป็น 0

สังเกตว่าการที่ผู้ใช้แต่ละคนสามารถจะดึงสัญญาณข้อมูลของตนเองออกมาจากสัญญาณรวมได้ก็เพราะว่า เมื่อนำโค้ด PN ของผู้ใช้คนหนึ่งไปคูณกับส่วนสัญญาณที่สเปรดออก โดยผู้ใช้อีกคนหนึ่ง เช่นถ้าเรานำชุดโค้ด PN ของผู้ใช้หมายเลข 1 ไปคูณกับส่วนของสัญญาณที่สเปรดออกโดยผู้ใช้หมายเลข 2 จะพบว่าค่าผลรวมของการคูณในทุก ๆ 4 บิทจะได้ค่าที่เป็น 0 หมด ซึ่งจริง ๆ และถ้าหากทำการทดสอบกับทุกกรณีจะได้ผลที่เหมือนกันคือค่าผลรวมที่ได้มีค่าเป็น 0 เสมอ ตัวอย่างนี้แสดงให้เห็นว่าสัญญาณที่สเปรดออกด้วยชุดโค้ด PN ชุดหนึ่งจะไม่สามารถดีสเปรดออกด้วยชุดโค้ด PN ชุดอื่น ซึ่งทำให้เราสามารถส่งสัญญาณของผู้ใช้หลาย ๆ คนลงบนคลื่นพาห์ความถี่เดียวกันได้ สังเกตว่าจุดสำคัญของการทำสเปรดสเปกตรัมก็คือการเลือกชุดโค้ดที่มีคุณสมบัติพิเศษดังที่อธิบายไว้ในเบื้องต้นนั่นเอง

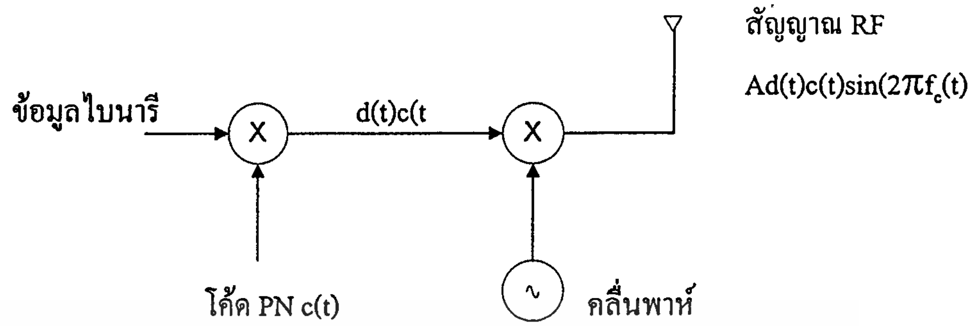
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



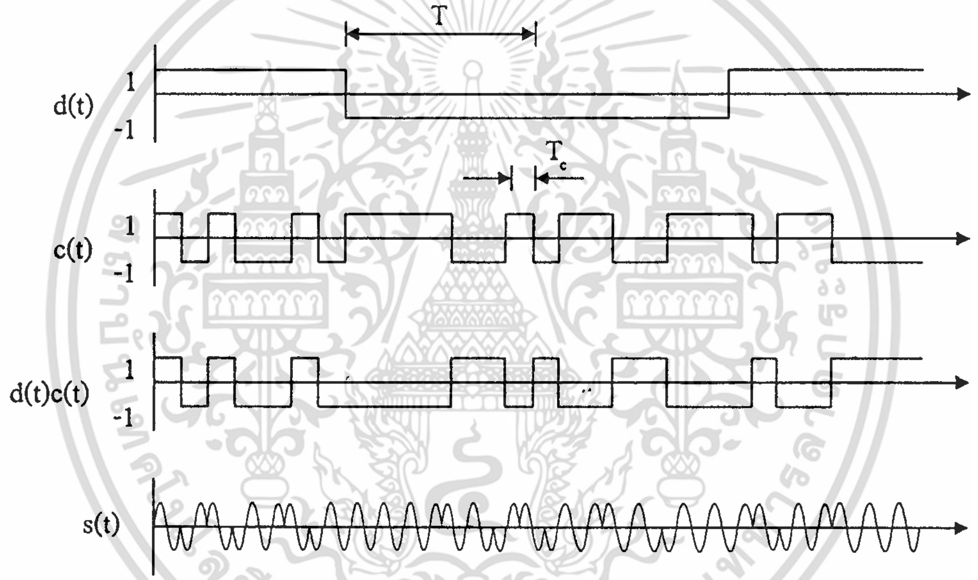

**รูปที่ 2.3 ตัวอย่างการทำสเปรดสเปกตรัม**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.1.4 วงจรภาคส่งของระบบ Direct Sequence ที่ใช้การมอดูเลตแบบ BPSK




รูปที่ 2.4 วงจรภาคส่งของระบบ DS ที่ใช้การมอดูเลตแบบ BPSK




รูปที่ 2.5 ตัวอย่างของสัญญาณที่เกิดขึ้นในวงจรภาคส่งของระบบ DS ที่ใช้ BPSK

วงจรภาคส่งของระบบ DS ที่ใช้การมอดูเลตแบบ BPSK (Binary Phase Shift Keying) ประกอบด้วย 2 ส่วนสำคัญคือวงจรคูณสัญญาณข้อมูลกับโค้ด PN และส่วนของวงจรมอดูเลตเตอร์แบบ BPSK ดังที่แสดงในรูปที่ 2.4 สัญญาณขาเข้า d(t) ซึ่งมีอัตราของบิทข้อมูลที่ต่ำโดยให้สัญญาณนี้มีคาบเวลาเป็น T ถูกนำไปคูณกับโค้ด PN c(t) ที่มีคาบเวลาของหนึ่งชิปส์เท่ากับ $T_c$ ซึ่งสั้นกว่า T มาก ขั้นตอนนี้เรียกว่าการสเปรดสเปกตรัมของสัญญาณข้อมูลจากเดิมที่มีอัตราของบิทข้อมูลต่ำซึ่งใช้แบนด์วิดท์ของช่องสัญญาณขนาดเล็กในการส่งมาเป็นข้อมูลที่มีอัตราของชิปส์ที่สูงขึ้นกว่าเดิมมากซึ่งหมายถึงว่าต้องใช้แบนด์วิดท์ของช่องสัญญาณในการส่งที่ใหญ่ขึ้นมากด้วยจากนั้นก็นำสัญญาณ d(t)c(t) ที่ได้นี้ไปทำการมอดูเลตแบบ BPSK โดยใช้คลื่นพาห์ความถี่สูง รูปที่ 2.5 แสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



ตัวอย่างของสัญญาณที่ได้ในแต่ละขั้นตอน โดยในตัวอย่างนี้ได้กำหนดให้ $T = 7T_c$ และคลื่นพาห์มีความถี่เท่ากับ $f_c = 1/T_c$

### 2.1.5 วงจรภาครับของระบบ Direct Sequence ที่ใช้การมอดูเลตแบบ BPSK

รูปที่ 2.6 แสดงองค์ประกอบของภาครับระบบ DS ที่ใช้การมอดูเลตแบบ BPSK วงจรนี้ใช้ในการดึงสัญญาณข้อมูล d(t) ออกมาจากสัญญาณ RF ที่รับได้ซึ่งโดยปกติแล้วสัญญาณนี้จะมีการดีเลย์เนื่องมาจากการแพร่ของสัญญาณ (propagation delay) ดังนั้นสัญญาณที่รับได้สามารถเขียนในรูป

$$s(t-\tau) = d(t-\tau)c(t-\tau)\sin(2\pi f_c(t-\tau))$$

โดย $\tau$ คือ ค่าดีเลย์ไทม์ ดูรูปที่ 2.7 ประกอบการอธิบาย

สัญญาณ $s(t-\tau)$ นี้ถูกนำไปคูณกับโค้ด PN ชุดเดียวกันกับที่ใช้ในภาคส่ง หากแต่ว่าวงจรภาครับต้องสามารถประมาณหาค่าดีเลย์ไทม์ให้ใกล้เคียงกับค่า $\tau$ มากที่สุด ผลที่ได้จากการคูณของสัญญาณทั้งสองคือ

$$c(t-\tau)s(t-\tau) = d(t-\tau)A\sin(2\pi f_c(t-\tau))$$

ทั้งนี้เพราะ $c(t-\tau)c(t-\tau) = 1$ ณ ที่เวลา t ใด ๆ

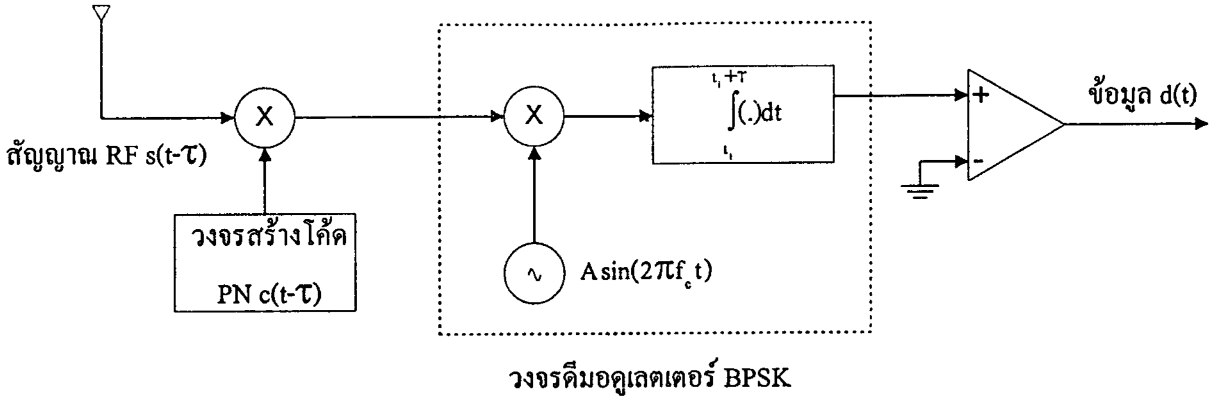
ขั้นตอนนี้คือการดีสเปรดของสัญญาณข้อมูลที่ต้องการออกมา จากนั้นเมื่อนำสัญญาณนี้ไปเข้าวงจรดีมอดูเลเตอร์แบบ BPSK ก็จะได้สัญญาณข้อมูลกลับคืนมา ในวงจรดีมอดูเลเตอร์ประกอบด้วย 2 ส่วนหลักคือวงจรคูณและวงจรอินทิเกรเตอร์ สัญญาณที่ได้หลังจากผ่านวงจรคูณแล้วสามารถแสดงได้ดังนี้

$$d(t-\tau)A\sin(2\pi f_c(t-\tau))\sin(2\pi f_c(t-\tau)) = d(t-\tau)A\{1-\cos(4\pi f_c(t-\tau))\}/2$$

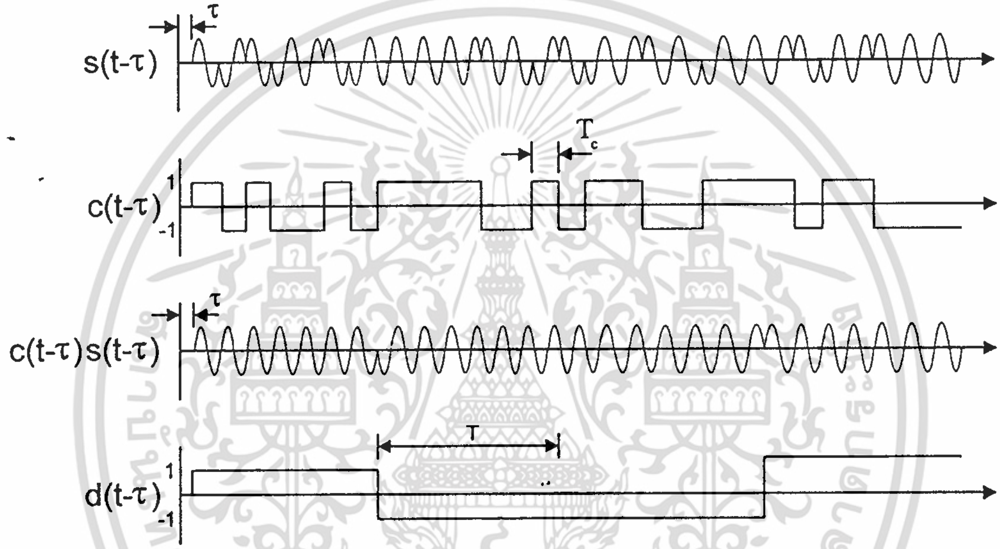
เมื่อนำสัญญาณเข้าวงจรอินทิเกรเตอร์ที่ทำการอินทิเกรตใหม่ทุก ๆ คาบเวลา T ซึ่งโดยปกติแล้วจะมีช่วงเวลายาวกว่าคาบเวลา $T_c$ มาก ดังนั้นสัญญาณ $\sin(4\pi f_c(t-\tau))$ เมื่อถูกอินทิเกรตแล้วก็จะมีค่าใกล้ศูนย์มาก เพราะฉะนั้นสัญญาณที่เหลือจากวงจรอินทิเกรตจึงมีเพียงสัญญาณ $d(t-\tau)$ ซึ่งเมื่อนำมาเข้าวงจรเปรียบเทียบแล้วก็จะได้ข้อมูลเดิมที่มีค่าเป็น 0 และ 1 กลับคืนมา

หมายเหตุ : ขั้นตอนการคูณโค้ด PN กับส่วนของวงจรดีมอดูเลเตอร์สามารถนำมาสลับกันได้ โดยที่ยังคงได้ผลลัพธ์เหมือนกัน จากตัวอย่างนี้จะเห็นว่าถ้าหากภาครับไม่รู้ชุดโค้ด PN ที่ใช้ที่ภาคส่งก็จะไม่สามารถแยกสัญญาณข้อมูลออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.6** วงจรภาครับของระบบ DS ที่ใช้การมอดูเลตแบบ BPSK




**รูปที่ 2.7** ตัวอย่างของสัญญาณที่เกิดขึ้นในวงจรภาครับของระบบ DS ที่ใช้ BPSK

## 2.2 ระบบโทรศัพท์เคลื่อนที่ CDMA

การวิจัยและพัฒนาระบบโทรศัพท์เคลื่อนที่แบบ CDMA เริ่มขึ้นอย่างจริงจังเมื่อประมาณต้นปี ค.ศ. 1989 และเพียงไม่กี่ปีต่อมาคือในปลายปี ค.ศ. 1992 ผลงานวิจัยและพัฒนาเหล่านี้ก็ได้รับการเสนอและยอมรับให้เป็นมาตรฐานที่เรียกว่า IS-95 ในปัจจุบันระบบโทรศัพท์เคลื่อนที่แบบ CDMA นี้ได้เริ่มมีการนำมาใช้งานจริงกันในหลายประเทศทั่วโลก สำหรับในประเทศไทยนั้นการสื่อสารแห่งประเทศไทยได้นำระบบนี้เข้ามาให้บริการโทรศัพท์เคลื่อนที่ในราวเดือนเมษายนปี ค.ศ. 1998 หลักการเบื้องต้นของระบบ CDMA ได้จัดแบ่งความถี่ที่มีอยู่ทั้งหมดออกเป็นช่วง โดยแต่ละช่วงมีความกว้างเท่ากับ 1.25 MHz โดยวิธี FDMA สังเกตว่าคลื่นพาห์หนึ่งความถี่มีขนาดแบนด์วิดท์ที่กว้างกว่าของระบบอื่น ๆ เช่น GSM (200 kHz) มาก ด้วยขนาดแบนด์วิดท์ที่กว้างขนาดนี้ CDMA อนุญาตให้ผู้ใช้โทรศัพท์จำนวนมากจำนวนหนึ่งส่งสัญญาณเสียงลงบนคลื่นความถี่เดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ ซึ่งหมายความว่าผู้ใช้ทุก ๆ คนที่ใช้คลื่นพาห์เดียวกันจะรบกวนในเชิงความถี่ตลอดเวลาการใช้งาน หากแต่ว่าโทรศัพท์แต่ละเครื่องจะสามารถแยกแยะว่าสัญญาณส่วนใดที่เป็นของตนเองได้โดยอาศัยชุดโค้ดที่แตกต่างกันในการรับและส่งข้อมูล ด้วยเหตุนี้จึงเรียกระบบนี้ว่า Code Division Multiple Access (CDMA) ชุดโค้ดที่ใช้ในการทำสเปรดสเปกตรัมเหล่านี้มีชื่อเรียกว่า PN (Pseudorandom Noise) ซึ่งชุดโค้ดแต่ละชุดที่ใช้จะต้องมีคุณสมบัติที่เรียกว่า orthogonal กับชุดโค้ดอื่น ๆ ทั้งหมด

ขั้นตอนการสร้างสัญญาณสำหรับการส่งออกเริ่มด้วยการนำสัญญาณเสียงที่อยู่ในรูปของสัญญาณดิจิตอลซึ่งมีอัตราของบิทข้อมูลค่อนข้างต่ำ (เช่น 9600 bps) มาคูณกับสัญญาณ PN ซึ่ง มีอัตราของบิทข้อมูลที่สูงกว่ามาก (1.228 Mcps) ผลลัพธ์ที่ได้ก็คือข้อมูลที่อัตราการส่งสูงกว่าสัญญาณเสียงที่ต้องการจะส่งมาก ด้วยเหตุนี้จึงเรียกว่ามีการสเปรดสเปกตรัม (spread spectrum) เกิดขึ้นและแบนด์วิดท์ของช่องสัญญาณที่ใช้ในการส่งข้อมูลนี้ก็ต้องมีขนาดใหญ่ขึ้นด้วยคือ 1.25 MHz หมายเหตุ : สัญญาณที่ผ่านกระบวนการสเปรดสเปกตรัมแล้วมักจะใช้ mcps แทน mbps ส่วนของภาครับสัญญาณที่ปลายทางก็จะทำการดีสเปรด (despread) สัญญาณที่รับได้เพื่อดึงสัญญาณเสียงเดิมกลับคืนมา ในการดีสเปรดสัญญาณนั้นจะอาศัยวงจรคอรีเลเตอร์ (correlator) ซึ่งทำหน้าที่คูณสัญญาณที่ได้รับกับโค้ด PN ชุดเดิมซึ่งเหมือนกันกับการคูณที่ภาคส่ง และจากนั้นก็หาค่าเฉลี่ยของสัญญาณทุก ๆ หนึ่งคาบของอัตราบิทข้อมูลของสัญญาณเสียง

หมายเหตุ : สัญญาณข้อมูลที่ต้องการส่งออกมิจำเป็นต้องเป็นสัญญาณเสียงเท่านั้น อาจเป็นสัญญาณข้อมูลแบบอื่นเช่นข้อมูลประเภทดาต้าก็ได้

การที่เครื่องโทรศัพท์เคลื่อนที่แต่ละเครื่องที่ส่งผ่านสัญญาณเสียงหรือข้อมูลดาต้าในรูปของสัญญาณดิจิตอลโดยใช้ชุดโค้ด PN ที่แตกต่างกันและมีคุณสมบัติที่เป็น orthogonal ซึ่งกันและกันทำให้การรับและดึงสัญญาณของเครื่องโทรศัพท์เครื่อง ๆ หนึ่งจะไม่ไปดีสเปรดสัญญาณที่ถูกสเปรดออกด้วยโค้ด PN ชุดอื่น ด้วยเหตุนี้เครื่องโทรศัพท์แต่ละเครื่องจึงสามารถแยกแยะสัญญาณที่เป็นของตนเองได้ ค่าพารามิเตอร์ค่าหนึ่งที่สำคัญคือ Processing gain ซึ่งได้กำหนดให้มีค่าเท่ากับอัตราส่วนระหว่างอัตราบิทข้อมูลของโค้ด PN ต่ออัตราการส่งของข้อมูลจริง ในกรณีที่ส่งสัญญาณข้อมูลด้วยอัตรา 9600 bps จะได้ค่า Processing gain เท่ากับ 1.228 Mbps/9600 bps = 128 หรือเท่ากับ 21 dB

### 2.2.1 ประเภทของโค้ดที่ใช้ในระบบ CDMA (IS-95)

ในระบบ CDMA ตามมาตรฐาน IS-95 มีการใช้โค้ดทั้งหมด 3 ประเภท คือ

1. Walsh code
2. Long PN code
3. Short PN code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โค้ดแต่ละประเภทได้ถูกนำมาใช้งานในระบบ IS-95 ในลักษณะที่แตกต่างกัน คุณสมบัติและรายละเอียดของโค้ดทั้ง 3 ประเภทจะได้กล่าวถึงต่อไป

### 2.2.2 โค้ด Walsh

โค้ด Walsh มีชื่อเรียกอีกอย่างว่า Hadamard code ประกอบด้วยโค้ดที่มีคุณสมบัติ orthogonal ระหว่างกันทั้งหมด 64 ชุด โดยที่โค้ดแต่ละชุดมีความยาวเท่ากับ 64 บิต โค้ดประเภทนี้ออกแบบมาเพื่อใช้สำหรับสเปรดช่องสัญญาณที่มีอัตราการส่ง 1.228 Mbps

หมายเหตุ : โค้ด 2 ชุดจะมีคุณสมบัติ orthogonal ระหว่างกันก็ต่อเมื่อค่าสหสัมพันธ์ไขว้ (cross correlation) ระหว่างโค้ดคู่นี้มีค่าเป็นศูนย์ การคำนวณหาค่าสหสัมพันธ์ไขว้สามารถหาได้จาก $\sum x_i y_i = 0$ โดย $x_i$ และ $y_i$ คือค่าของบิทที่ i ของโค้ดชุดแรกและชุดที่สองในรูปของไบโพลาร์ นั่นคือว่า $x_i$ , $y_i$ = -1 ถ้าบิทที่ i นั้นมีค่าเป็นศูนย์ และ $x_i$ , $y_i$ = 1 ถ้าบิทที่ i นั้นมีค่าเป็นหนึ่ง ตัวอย่างของโค้ดที่มีคุณสมบัติ orthogonal กันเช่น X = 0011 และ Y = 0110

โค้ด Walsh สามารถสร้างได้จาก Hadamard matrices $H_{2m}$ โดยที่ m คือค่า 2 ยกกำลังตัวเลขจำนวนเต็ม (m = 1,2,4,......) ค่าเริ่มต้นของ Hadamard matrices ที่ m = 1 กำหนดให้เป็น

$$H_2 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$$

และค่า $H_{2m}$ อื่น ๆ สร้างจากความสัมพันธ์ต่อไปนี้

$$H_{2m} = \begin{bmatrix} H_m & H_m \\ H_m & \overline{H}_m \end{bmatrix}$$

ตัวอย่างเช่นกรณี m = 2 ได้

$$H_4 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และกรณี m = 4 ได้ $H_8 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระบบ CDMA ใช้ Hadamard code $H_{64}$ ดังที่แสดงในตารางที่ 2.1

**ตารางที่ 2.1** 64 –ary Walsh Functions

| Index | Walh codes |
|---|---|
| 0 | 0000000000000000000000000000000000000000000000000000000000000000 |
| 1 | 0101010101010101010101010101010101010101010101010101010101010101 |
| 2 | 0011001100110011001100110011001100110011001100110011001100110011 |
| 3 | 0110011001100110011001100110011001100110011001100110011001100110 |
| 4 | 0000111100001111000011110000111100001111000011110000111100001111 |
| 5 | 0101101001011010010110100101101001011010010110100101101001011010 |
| 6 | 0011110000111100001111000011110000111100001111000011110000111100 |
| 7 | 0110100101101001011010010110100101101001011010010110100101101001 |
| 8 | 0000000011111111000000001111111100000000111111110000000011111111 |
| 9 | 0101010110101010010101011010101001010101101010100101010110101010 |
| 10 | 0011001111001100001100111100110000110011110011000011001111001100 |
| 11 | 0110011010011001011001101001100101100110100110010110011010011001 |
| 12 | 0000111111110000000011111111000000001111111100000000111111110000 |
| 13 | 0101101010100101010110101010010101011010101001010101101010100101 |
| 14 | 0011110011000011001111001100001100111100110000110011110011000011 |
| 15 | 0110100110010110011010011001011001101001100101100110100110010110 |
| 16 | 0000000000000000111111111111111100000000000000001111111111111111 |
| 17 | 0101010101010101101010101010101001010101010101011010101010101010 |
| 18 | 0011001100110011110011001100110000110011001100111100110011001100 |
| 19 | 0110011001100110100110011001100101100110011001101001100110011001 |
| 20 | 0000111100001111111100001111000000001111000011111111000011110000 |
| 21 | 0101101001011010101001011010010101011010010110101010010110100101 |
| 22 | 0011110000111100110000111100001100111100001111001100001111000011 |
| 23 | 0110100101101001100101101001011001101001011010011001011010010110 |
| 24 | 0000000011111111111111110000000000000000111111111111111100000000 |
| 25 | 0101010110101010101010100101010101010101101010101010101001010101 |
| 26 | 0011001111001100110011000011001100110011110011001100110000110011 |
| 27 | 0110011010011001100110010110011001100110100110011001100101100110 |
| 28 | 0000111111110000111100000000111100001111111100001111000000001111 |
| 29 | 0101101010100101101001010101101001011010101001011010010101011010 |
| 30 | 0011110011000011110000110011110000111100110000111100001100111100 |
| 31 | 0110100110010110100101100110100101101001100101101001011001101001 |
| 32 | 0000000000000000000000000000000011111111111111111111111111111111 |
| 33 | 0101010101010101010101010101010110101010101010101010101010101010 |
| 34 | 0011001100110011001100110011001111001100110011001100110011001100 |
| 35 | 0110011001100110011001100110011010011001100110011001100110011001 |
| 36 | 0000111100001111000011110000111111110000111100001111000011110000 |
| 37 | 0101101001011010010110100101101010100101101001011010010110100101 |
| 38 | 0011110000111100001111000011110011000011110000111100001111000011 |
| 39 | 0110100101101001011010010110100110010110100101101001011010010110 |
| 40 | 0000000011111111000000001111111111111111000000001111111100000000 |
| 41 | 0101010110101010010101011010101010101010010101011010101001010101 |
| 42 | 0011001111001100001100111100110011001100001100111100110000110011 |
| 43 | 0110011010011001011001101001100110011001011001101001100101100110 |
| 44 | 0000111111110000000011111111000011110000000011111111000000001111 |
| 45 | 0101101010100101010110101010010110100101010110101010010101011010 |
| 46 | 0011110011000011001111001100001111000011001111001100001100111100 |
| 47 | 0110100110010110011010011001011010010110011010011001011001101001 |
| 48 | 0000000000000000111111111111111111111111111111110000000000000000 |
| 49 | 0101010101010101101010101010101010101010101010100101010101010101 |
| 50 | 0011001100110011110011001100110011001100110011000011001100110011 |
| 51 | 0110011001010011010011001100110011001100110011001100101100110011 |
| 52 | 0000111100001111111100001111000011110000111100000000111100001111 |
| 53 | 0101101001011010101001011010010110100101101001010101101001011010 |
| 54 | 0011110000111100110000111100001111000011110000110011110000111100 |
| 55 | 0110100101101001100101101001011010010110100101100110100101101001 |
| 56 | 0000000011111111111111110000000011111111000000000000000011111111 |
| 57 | 0101010110101010101010100101010110101010010101010101010110101010 |
| 58 | 0011001111001100110011000011001111001100001100110011001111001100 |
| 59 | 0110011010011001100110010110011010011001011001100110011010011001 |
| 60 | 0000111111110000111000000000111111110000000011010010110101010010 |
| 61 | 0101101010100101010110100101010110101010010101011010010110101001 |
| 62 | 0011110011000011110000110011110011000011001111000011110011000011 |
| 63 | 0110100110010110100101100110100110010110011010010110100110010110 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.3 โค้ด Long PN

โค้ดประเภท Long PN สร้างจากชิฟต์รีจิสเตอร์ขนาด 42 บิท โดยต่อเชื่อมกันให้มีโครงสร้างเป็น maximum length LFSRs และใช้โพลิโนเมียลต่อไปนี้

$$p(x) = x^{42} + x^{35} + x^{33} + x^{31} + x^{27} + x^{26} + x^{25} + x^{22} + x^{21} + x^{19} + x^{18} + x^{17} + x^{16} + x^{10} + x^{7} + x^{6} + x^{5} + x^{3} + x^{2} + x + 1$$

ผลที่ได้คือโค้ดที่มีความยาวถึง $2^{42}-1 = 4.398 \times 10^{12}$ เนื่องจากอัตราการสร้างโค้ดนี้มีค่าเท่ากับ 1.228 Mbps ดังนั้นโค้ดนี้จะมีการซ้ำทุก ๆ 41.425 วัน

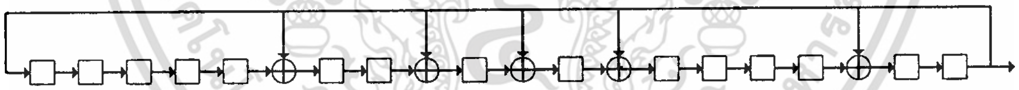
### 2.2.4 โค้ด Short PN

โค้ด Short PN นี้มีอยู่ 2 ชุดคือ I และ Q โดยแต่ละชุดสร้างจากการใช้ชิฟต์รีจิสเตอร์ขนาด 15 บิท โดยต่อเชื่อมกันให้มีโครงสร้างเป็น maximum length LFSRs และใช้โพลิโนเมียล 2 ชุดต่อไปนี้

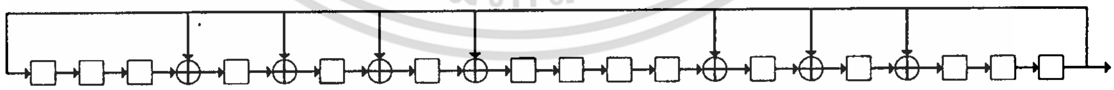
$$I(x) = x^{15} + x^{13} + x^{9} + x^{8} + x^{7} + x^{5} + 1$$

$$Q(x) = x^{15} + x^{12} + x^{11} + x^{10} + x^{6} + x^{5} + x^{4} + x^{3} + 1$$

โครงสร้างของวงจรสร้างชุดโค้ด Short PN มีลักษณะดังที่แสดงในรูปที่ 2.8 และ 2.9 วงจรทั้งสองสามารถสร้างชุดโค้ดที่มีความยาวเท่ากับ $2^{15}-1 = 32,767$ จากนั้นจะมีการเพิ่มบิท 0 อีกหนึ่งบิทเข้าไปเมื่อพบโค้ดที่มีค่า 0 ติดต่อกัน 14 ตัว เพื่อให้ได้โค้ดที่มีความยาวเท่ากับ 32,768 เนื่องจากอัตราการสร้างโค้ดนี้มีค่าเท่ากับ 1.228 Mbps ดังนั้นโค้ดนี้จะมีการซ้ำทุก ๆ 26.66 ms



**รูปที่ 2.8** วงจร 15-stage LFSR สำหรับ I



**รูปที่ 2.9** วงจร 15-stage LFSR สำหรับ Q

### 2.2.5 การส่งสัญญาณจากสถานีฐานไปหาโทรศัพท์

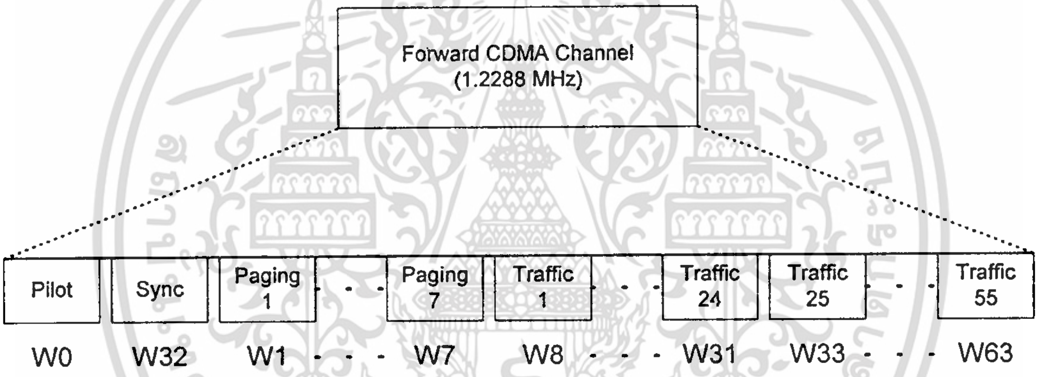
สัญญาณที่ส่งในทิศทางจากสถานีฐานไปยังเครื่องโทรศัพท์เคลื่อนที่หรือที่เรียกว่า Forward Link นั้น ประกอบด้วยช่องสัญญาณทั้งหมด 4 ประเภท คือ

- ช่องสัญญาณไพลอต (Forward pilot channel)
- ช่องสัญญาณซิงโครไนเซชั่น (Forward synchronization channel)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ช่องสัญญาณเพจจิง (Forward paging channel)
- ช่องสัญญาณทราฟฟิก (Forward traffic channel)

โดยคลื่นพาห์หนึ่งความถี่สามารถใช้ส่งสัญญาณทั้ง 4 ประเภทรวมกันได้ทั้งหมดไม่เกิน 64 ช่องสัญญาณ ทั้งนี้จะต้องมีช่องสัญญาณไพลอต (pilot channel) จำนวน 1 ช่องเสมอ มีช่องสัญญาณซิงโครไนเซชั่น (synchronization channel) ไม่เกิน 1 ช่อง มีช่องสัญญาณเพจจิง (paging channel) ไม่เกิน 7 ช่อง และมีช่องสัญญาณทราฟฟิกอีกจำนวนหนึ่ง ช่องสัญญาณทั้ง 64 ช่องสามารถแยกออกจากกันได้โดย Walsh code 64 ชุด ดูตัวอย่างการแบ่งช่องสัญญาณที่มักจะมีการใช้งานตามปกติในรูปที่ 2.10 สังเกตว่าช่องสัญญาณไพลอตจะใช้ Walsh code ชุดที่ 0 (W0) เสมอ ส่วนช่องสัญญาณซิงโครไนเซชั่น ถ้ามีการใช้งานก็จะใช้ Walsh code ชุดที่ 32 (W32) เสมอ แต่ถ้าหากเป็นช่องสัญญาณเพจจิงก็จะมีการใช้ Walsh code ชุดที่ 1-7 (W1-W7) เรียงตามลำดับ และสำหรับช่องสัญญาณทราฟฟิกจะใช้ W8-W31 และ W33-W64



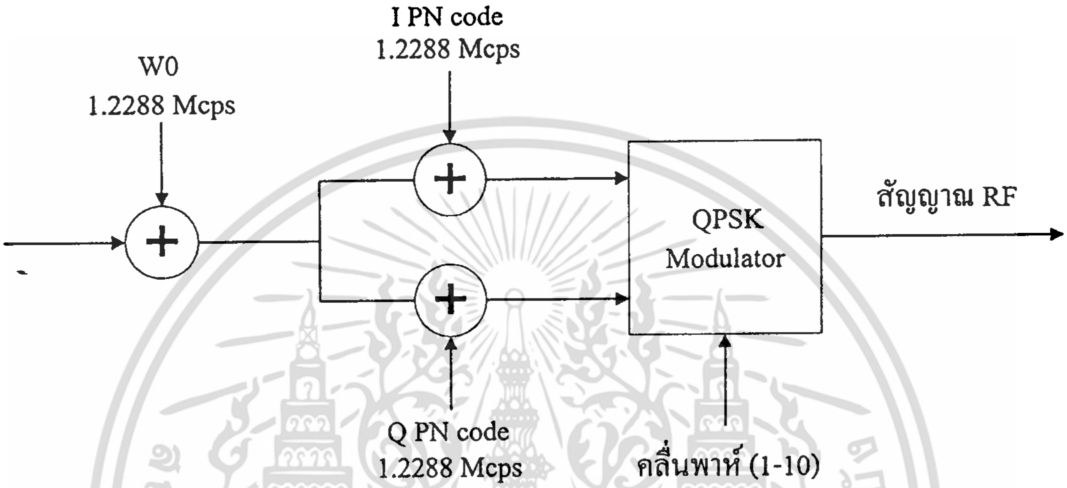

รูปที่ 2.10 การแบ่งช่องสัญญาณที่ส่งออกโดยสถานีฐาน

### 2.2.6 ช่องสัญญาณไพลอต

โดยปกติสถานีฐานของแต่ละเซลล์จะส่งสัญญาณไพลอต (pilot signal) ออกไปให้เครื่องโทรศัพท์ที่อยู่รอบ ๆ ข้างตลอดเวลา เพื่อใช้เป็นสัญญาณคลื่นพาห์อ้างอิงสำหรับการดีมอดูเลตแบบโคฮีเรนต์โดยที่ไม่มีส่วนของข้อมูลจากผู้ใช้มอดูเลตเข้าไปเลย การส่งสัญญาณนี้จะใช้กำลังส่งที่สูงกว่าสัญญาณประเภทอื่น ๆ ทั้งนี้เพื่อให้อุปกรณ์โทรศัพท์รอบข้างสามารถตรวจจับสัญญาณได้โดยง่าย ในการส่งสัญญาณไพลอตนั้นจะใช้ Walsh code ชุดที่ประกอบด้วยศูนย์ทั้งหมด 64 บิท (W0) เพราะฉะนั้นสัญญาณนี้ประกอบด้วยข้อมูลเพียงอย่างเดียวคือ Short PN code ของ I กับ Q ซึ่งข้อมูลส่วนนี้ใช้เป็นตัวบ่งบอกได้ว่าสัญญาณไพลอตนี้ถูกส่งออกมาจากสถานีฐานใด ดังนั้นเครื่องโทรศัพท์เพียงแต่อ่านข้อมูลของ Short PN code จนครบหนึ่งคาบ (26.66 ms) ทราบว่าเป็นสัญญาณที่ได้รับนี้เป็นของสถานีฐานใด ซึ่งโดยปรกติแล้วทันทีที่มีการเปิดเครื่องโทรศัพท์เคลื่อนที่ตัวเครื่องโทรศัพท์จะทำการค้นหาและวัดกำลังของสัญญาณไพลอต ที่ส่งออกจากสถานีฐานนั้นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไป ทุกสถานีฐานจะใช้ Short PN code ชุดเดียวกันหมดเพียงแต่จะมีการเลื่อนของชิปส์ออกไปด้วยจำนวนที่แตกต่างกัน ในการเลื่อนชิปส์นั้นจะเลื่อนออกไปทีละ 64 ชิปส์ เพราะฉะนั้นก็จะมีโค้ดที่แตกต่างกันสำหรับการใช้งานได้ทั้งหมด 32,768/64=512 ชุด โค้ดชุดหนึ่งอาจมีการนำมาใช้งานซ้ำในหลาย ๆ เซลล์ได้ หากแต่เซลล์เหล่านั้นจะต้องจัดอยู่ห่างกันมากเพียงพอที่จะไม่รบกวนซึ่งกันและกัน รูปที่ 2.11 แสดงรายละเอียดขั้นตอนของการสร้างสัญญาณไพลอต



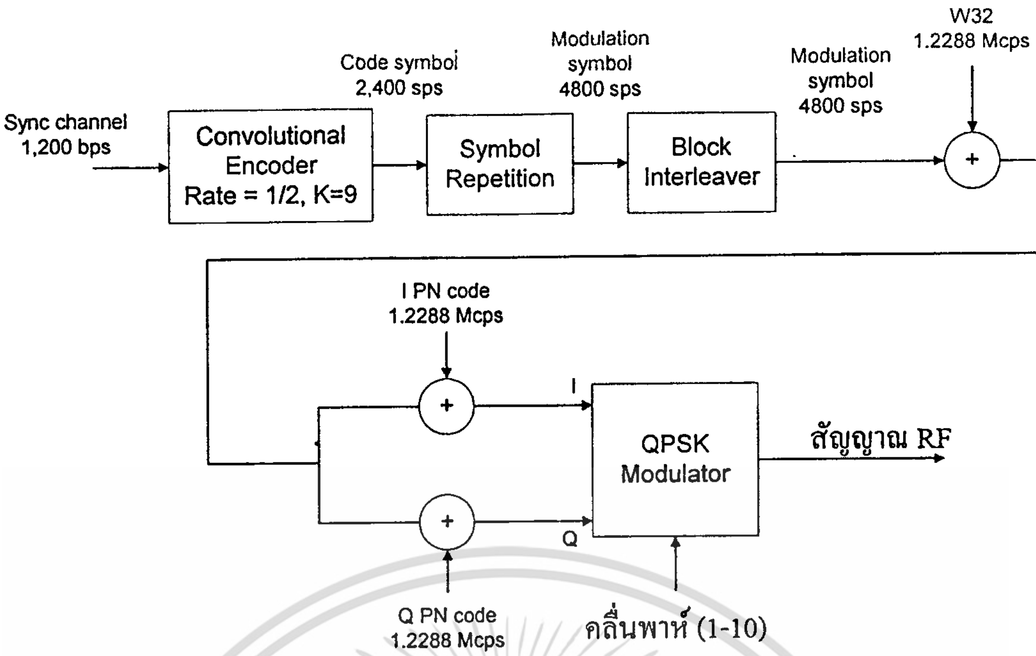

รูปที่ 2.11 วงจรสร้างสัญญาณไพลอต

สัญญาณไพลอตยังมีประโยชน์สำหรับใช้ในการควบคุมกำลังส่งของสัญญาณ (power control) โดยทันทีที่เครื่องโทรศัพท์ได้รับสัญญาณไพลอตก็จะทำการคำนวณหาค่ากำลังเฉลี่ยของสัญญาณที่รับได้เพื่อใช้สำหรับประมาณค่ากำลังสูญเสียระหว่างทางของสัญญาณ ซึ่งค่านี้จะนำมาใช้ในการกำหนดขนาดของกำลังที่เหมาะสมของสัญญาณที่โทรศัพท์จะใช้ในการส่งออก

### 2.2.7 ช่องสัญญาณซิงโคไนเซชั่น

ช่องสัญญาณซิงโคไนเซชั่น (Synchronization channel) มีไว้สำหรับส่งข้อมูลที่จำเป็นในการติดต่อในช่วงเริ่มแรก เพื่อที่เครื่องโทรศัพท์จะได้สามารถซิงโครไนกับสัญญาณที่ส่งออกจากสถานีฐานได้อย่างถูกต้อง ข้อมูลเหล่านี้ประกอบด้วย จังหวะเวลาในการส่งสัญญาณ หมายเลขประจำตัวของระบบโครงข่ายที่เครื่องโทรศัพท์ติดต่ออยู่ นอกจากนี้ก็ยังบอกถึงเวลาของนาฬิกาที่ระบบใช้อยู่ ช่องสัญญาณนี้ใช้ Walsh Code ชุดที่ W32 เป็นตัวบอกถึงชนิดของช่องสัญญาณ สังเกตุว่าช่องสัญญาณซิงโครไนเซชั่น คล้ายกับช่องสัญญาณไพลอตตรงที่ไม่มีการใช้ long PN code เพราะเป็นสัญญาณที่แพร่ออกสำหรับออกโทรศัพท์เคลื่อนที่ทุกๆ เครื่อง ดูรายละเอียดของขั้นตอนการสร้างสัญญาณซิงโคไนเซชั่นได้ในรูปที่ 2.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



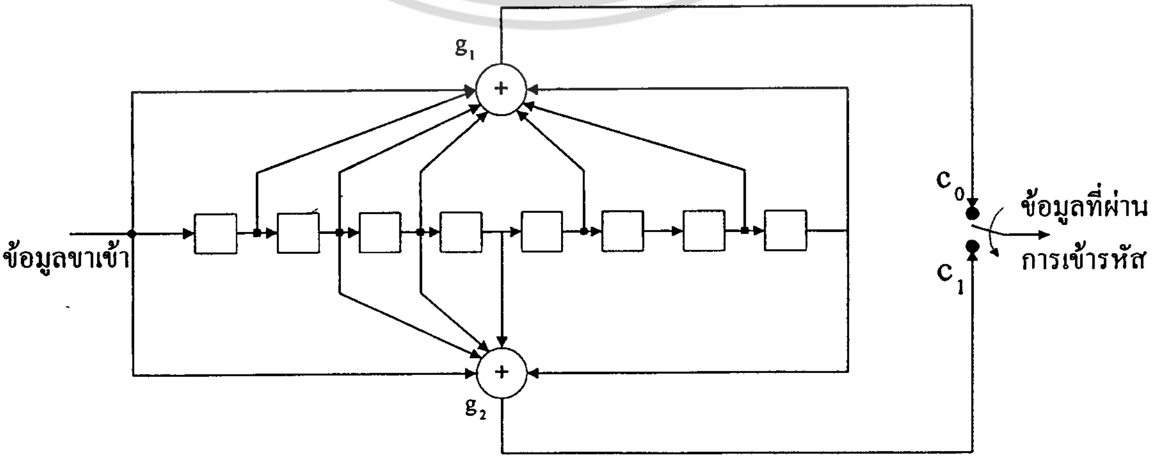

รูปที่ 2.12 วงจรสร้างสัญญาณซิงโครไนเซชั่น

จากรูปที่ 2.12 ในขั้นตอนแรกบิทข้อมูลที่ส่งในช่องสัญญาณซิงก์จะถูกนำไปผ่านกระบวนการเข้ารหัสช่องสัญญาณคอนโวลูชั่นที่มีอัตราการเข้ารหัส ½ และมี constraint span เท่ากับ 9 โดยชุดโพลีโนเมียลที่ใช้ประกอบด้วย

$g_1(x) = 1+x+x^2+x^3+x^5+x^7+x^8$

$g_2(x) = 1+x^2+x^3+x^4+x^8$

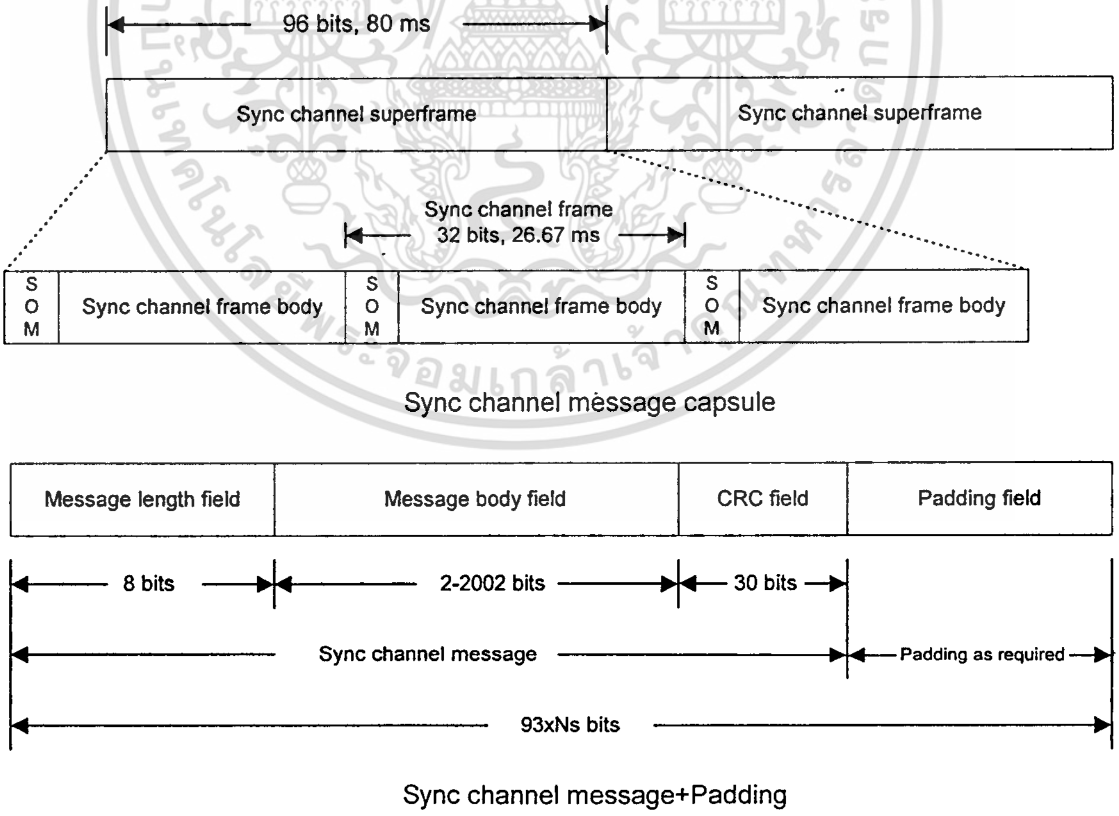
สำหรับโครงสร้างของวงจรเข้ารหัสคอนโวลูชั่นนี้ได้แสดงดังรูปที่ 2.13 หลังจากที่ผ่านการเข้ารหัสช่องสัญญาณแล้วอัตราบิทของข้อมูลเพิ่มขึ้นเท่าตัวเป็น 2.4 ksps ข้อมูลเหล่านี้จะถูกนำไปผ่านกระบวนการ Symbol repettition เพื่อให้ได้ชุดข้อมูลที่มีอัตราบิทเพิ่มสูงขึ้นเป็น 4.8 Ksps จากนั้นนำไปเข้าวงจรอินเตอร์ลีฟ (Block interleave) เพื่อลดผลกระทบของความผิดพลาดแบบเบิรสต์ก่อนที่จะถูกสเปรดออกด้วย Walsh Code ชุดที่ 32 ที่อัตราชิปส์ 1.2288 Mcps




รูปที่ 2.13 วงจรเข้ารหัสคอนโวลูชั่นสำหรับช่องสัญญาณซิงก์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

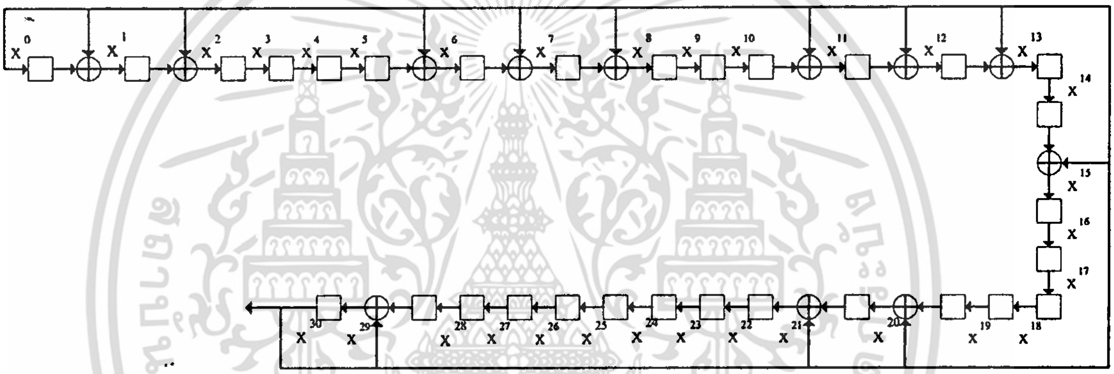
อัตราที่ใช้ในการส่งสัญญาณของช่องสัญญาณซิงโคไนเซชั่นมีค่าเท่ากับ 1,200 bps โดยรูปแบบโครงสร้างของข้อมูลที่จะส่งออกแบ่งออกเป็นช่วงเวลาที่เรียกว่าซุปเปอร์เฟรม(Superframe) ซึ่งมีขนาดเท่ากับ 80 ms และใช้ส่งข้อมูล 96 บิท ดูรูปที่ 2.15 ประกอบ ในหนึ่งซูเปอร์เฟรมจะบรรจุเฟรม (frame) ขนาด 32 บิท จำนวน 3 เฟรม ซึ่งแต่ละเฟรมจะมีความยาวเท่ากับ 26.66 ms ภายในแต่ละเฟรมประกอบด้วย 2 ส่วน คือ บิท SOM (Start of Message) และ Sync channel frame body ในส่วนของ SOM มีความยาว 1 บิท โดยมีหน้าที่เป็นตัวบ่งชี้ถึงว่าเฟรมดังกล่าวนี้บรรจุข้อความ (Message) ชุดใหม่หรือไม่ นั่นคือถ้า SOM=1 แสดงว่าเฟรมนี้บรรจุจุดเริ่มต้นของข้อความชุดใหม่ และถ้า SOM = 0 แสดงว่าเป็นข้อความส่วนที่เหลือ ข้อความที่จะมาบรรจุลงในช่องสัญญาณซิงก์นี้มีรูปแบบดังแสดงในส่วนล่างของรูปที่ 2.14 ในส่วนของบิทแรก (Message length filed) มีไว้สำหรับบอกถึงความยาวของข้อความ จากนั้นตามด้วย Message body filed ซึ่งบรรจุข้อมูลต่างๆ ที่สถานีฐานต้องการจะส่งออกไปให้เครื่องโทรศัพท์ ในส่วนนี้มีขนาดความยาวที่ไม่แน่นอนโดยมีค่าระหว่าง 2-2002 บิท ส่วนถัดมาคือ CRC Filed ที่มีไว้สำหรับใช้ในการตรวจจับความผิดพลาดของการส่งข้อมูลค่า CRC field คำนวณจาก message length filed และ message body field โดยอาศัยโพลีโนเมียล



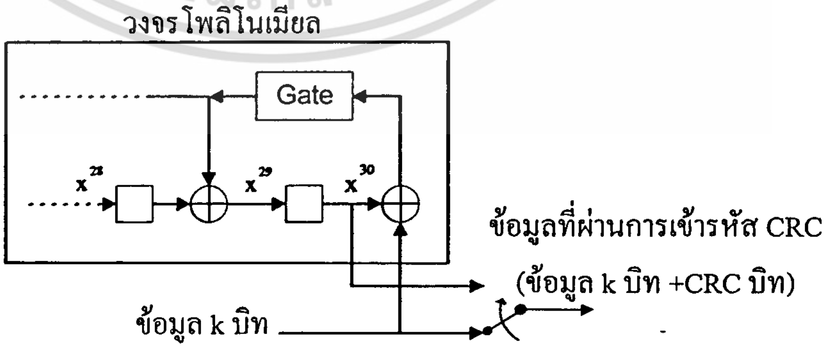

**รูปที่ 2.14 โครงสร้างเฟรมและรูปแบบของข้อมูลที่ใช้ในช่องสัญญาณซิงก์**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของวงจรสำหรับโพลิโนเมียลข้างต้นมีลักษณะดังที่แสดงในรูปที่ 2.15 ส่วนวงจรเข้ารหัส CRC โดยรวมมีโครงสร้างดังในรูปที่ 2.16 ลำดับขั้นตอนการทำงานของวงจรเข้ารหัส CRC มีรายละเอียดดังนี้ คือ ขั้นแรกระบบจะทำการตั้งค่าของชิฟต์รีจิสเตอร์แต่ละตัวให้มีค่าเป็น 1 ทั้งหมด จากนั้นปรับสวิตช์ที่ขาออกให้ตั้งอยู่ในตำแหน่งที่รับข้อมูลโดยตรงจากขาเข้าของวงจรและทำการเชื่อมวงจร gate เข้าด้วยกัน หลังจากที่ระบบได้ทำงานไปเป็นจำนวน k รอบให้ทำการปรับตำแหน่งของสวิตช์ที่ขาออกไปต่อเชื่อมกับส่วนของวงจรโพลิโนเมียลแทน พร้อมกันนั้นให้ทำการเปิด gate ออก และปล่อยให้วงจรทำงานในลักษณะนี้ไปอีก 30 รอบ ซึ่งในช่วงนี้เองที่ระบบจะได้ค่า CRC ออกมาจำนวนทั้งสิ้น 30 บิท ที่ขาออก ทั้งนี้ข้อมูล k บิท ที่กล่าวมานี้ประกอบด้วย message length field จำนวน 8 บิท และ message body field อีกจำนวน 2-2002 บิท



รูปที่ 2.15 โครงสร้างของโพลิโนเมียลที่ใช้ในการเข้ารหัส CRC ของช่องสัญญาณซิงก์

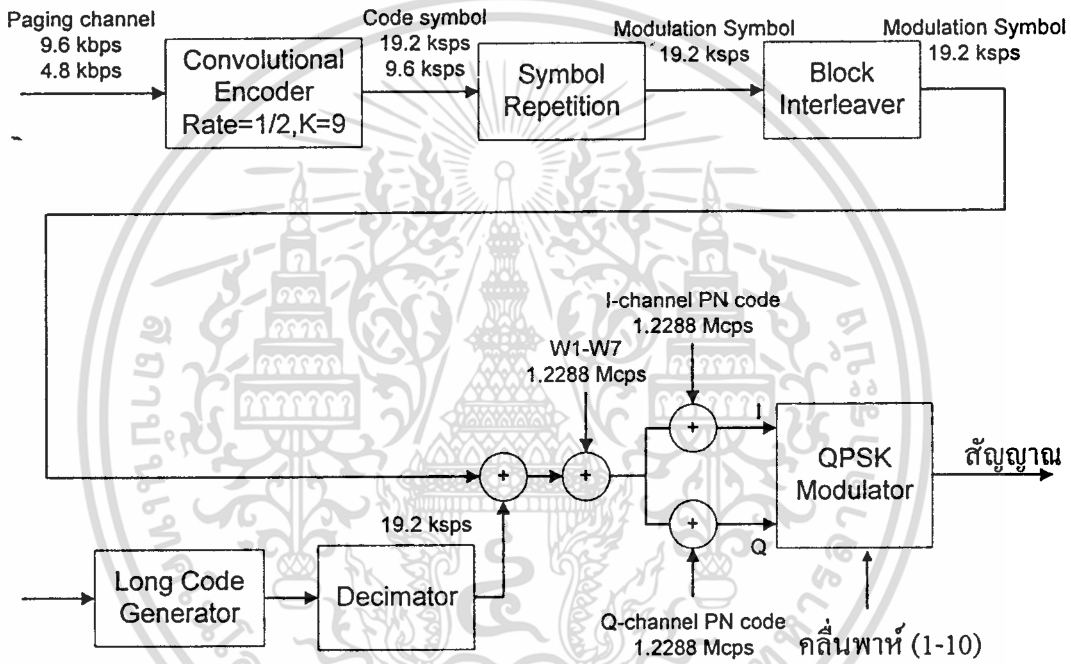



รูปที่ 2.16 โครงสร้างของวงจรเข้ารหัส CRC สำหรับช่องสัญญาณซิงก์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.2.8 ช่องสัญญาณเพจจิง

ช่องสัญญาณเพจจิง (paging channel) มีไว้สำหรับให้สถานีฐานใช้ในการเพจ (page) หรือค้นหาตำแหน่งของเครื่องโทรศัพท์ที่ต้องการได้แล้วก็จะจัดสรรช่องสัญญาณทราฟฟิก (Traffic channel) สำหรับใช้ในการติดต่อรับส่งข้อมูลของผู้ใช้ต่อไป อัตราบิทข้อมูลที่ใช้ส่งมี 2 ขนาดคือ 9,600 หรือ 4,800 bps โดยมีขนาดความยาวของเฟรมเท่ากับ 20 ms ชุด Walsh code ที่ใช้คือ W1-W7 หมายเหตุ : ในกรณีที่ชุดโค้ด W1-W7 มิได้มีการใช้งานอยู่ระบบก็สามารถที่จะใช้โค้ดเหล่านี้ในการส่งสัญญาณข้อมูลของผู้ใช้ได้ คือนำมาใช้เป็นช่องสัญญาณทราฟฟิก (traffic channel) แทน ดูขั้นตอนการสร้างสัญญาณเพจจิงได้ในรูปที่ 2.17




รูปที่ 2.17 ขั้นตอนการสร้างสัญญาณเพจจิง

ในขั้นแรกสัญญาณเพจจิงจะถูกนำมาเข้ารหัสช่องสัญญาณคอนโวลูชั่นที่มีอัตราการเข้ารหัสเท่ากับ R = 1/2 และมี contraint span เท่ากับ 9 โดยชุดโพลิโนเมียลที่ใช้ประกอบด้วย

$g_1(x) = 1+x+x^2+x^3+x^5+x^7+x^8$

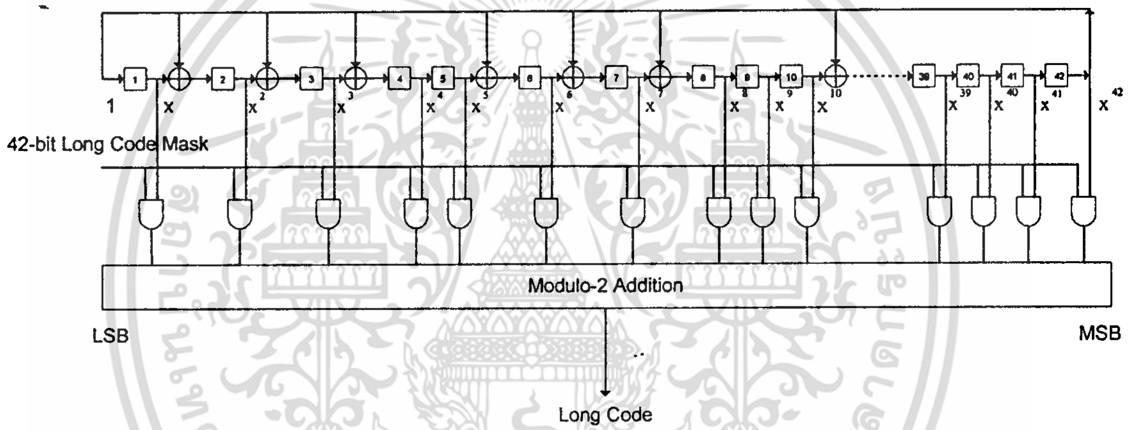
$g_2(x) = 1+x^2+x^3+x^4+x^8$

ผลที่ได้คือข้อมูลที่มีอัตราการบิทเพิ่มขึ้น 2 เท่า คือเท่ากับ 9,600 sps และ 19.2 ksps จากนั้นนำไปผ่านวงจร Symbol Repetition เพื่อทำให้อัตราการส่งของทั้ง 2 ประเภทมีค่าเท่ากันเป็น 19.6 ksps และทำการอินเตอร์ลีฟข้อมูลคราวละ 20 ms เพื่อลดผลกระทบของความผิดพลาดแบบเบิรสต์ที่อาจเกิดขึ้นจากช่องสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่ออกจากวงจรอินเตอร์ลีฟจะถูกนำไปสแกรมเบิล (scramble) โดยการนำไปบวก (modulo-2) กับลำดับสัญญาณที่สร้างขึ้นจากวงจร long code generator และ decimator ส่วนของวงจร long code generator มีลักษณะโครงสร้างดังแสดงในรูปที่ 2.18 ภายในวงจรประกอบด้วยส่วนของ 42-bit Long Code Mask ที่ทำหน้าที่ในการกำหนดรูปแบบของการสแกรมเบิลข้อมูล รูปแบบของ code mask ชุดนี้มีโครงสร้างดังที่แสดงในรูปที่ 2.19 วงจร long code generator นี้ให้กำเนิดลำดับสัญญาณที่มีอัตราบิทอยู่ที่ 1.2288 kcps ดังนั้นเมื่อผ่านเข้าสู่วงจร decimator ซึ่งทำหน้าที่ในการดึงเฉพาะบิทแรกของลำดับสัญญาณทุก ๆ 64 บิท ผลที่ได้คือสัญญาณที่มีอัตราบิทลดลงเหลือเท่า ๆ กับสัญญาณที่ออกจากวงจรอินเตอร์ลีฟคือ 19.2 ksps เพื่อนำไปทำการสแกรมเบิลต่อไป

สำหรับขั้นตอนที่เหลือก็มีลักษณะที่คล้ายกับช่องสัญญาณ 2 ประเภทแรกที่ได้กล่าวไว้ก่อนหน้า โดยใช้ Walsh code ชุดที่ W1-W7




**รูปที่ 2.18** Long code generator

| 41 … 29 | 28 … 24 | 23 … 21 | 20 … 9 | 8 … 0 |
|---|---|---|---|---|
| 1100011001101 | 00000 | PCN | 000000000000 | PILOT-PN |

PCN: Paging Channel Number

PILOT-PN : Pilot sequence offset index for the forward CDMA Channel

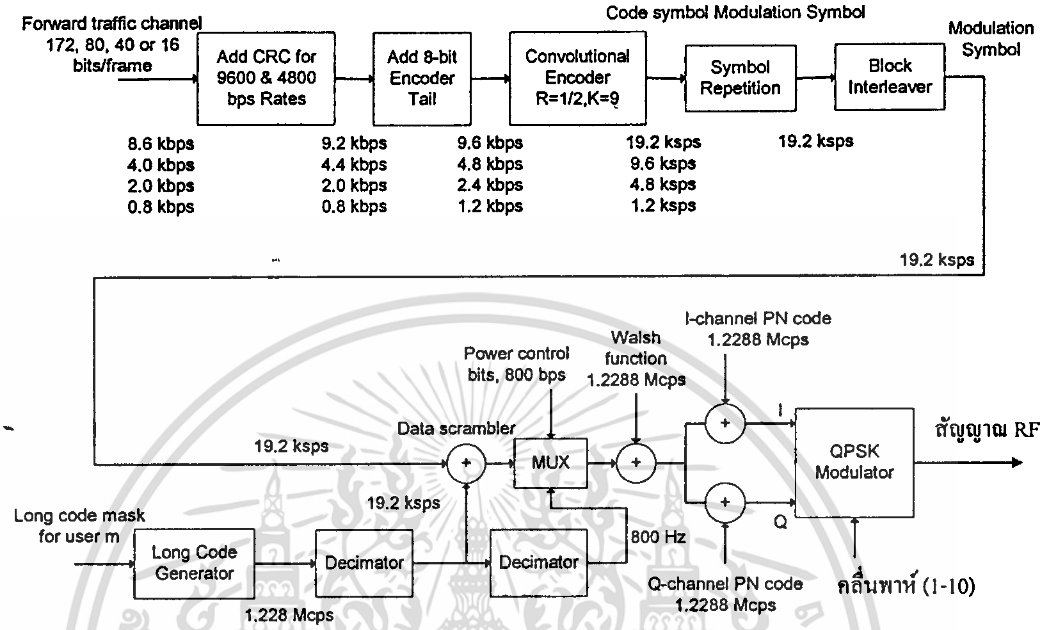
**รูปที่ 2.19** รายละเอียดรูปแบบของ paging channel long code mask

### 2.2.9 ช่องสัญญาณทราฟฟิก

ช่องสัญญาณทราฟฟิก (traffic channel) มีไว้สำหรับรับส่งสัญญาณเสียงและสัญญาณอื่น ๆ ตลอดช่วงเวลาที่มีการใช้งานของผู้ใช้ Walsh code ชุดที่สามารถนำมาใช้งานได้ก็คือ W8-W31, W33-W63 และ W1-W7 (ใช้เฉพาะเวลาที่มิได้นำไปใช้งานอย่างอื่น) อัตราบิทข้อมูลที่ใช้ในการส่งมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งหมด 4 แบบคือ 1,200, 2,400, 4,800 และ 9,600 bps โดยที่ขนาดของหนึ่งเฟรมมีความยาวเท่ากับ 20 ms




**รูปที่ 2.20 ขั้นตอนการสร้างช่องสัญญาณทราฟฟิก**

รูปที่ 2.20 แสดงขั้นตอนการสร้างช่องสัญญาณทราฟฟิก จะเห็นว่าในขั้นตอนแรก ๆ จะมีลักษณะคล้ายกับช่องสัญญาณประเภทอื่นที่ได้อธิบายไว้แล้ว คือมีการเข้ารหัสช่องสัญญาณแบบคอนโวลูชั่น การทำ Symbol repertition และการทำอินเตอร์ลีฟ หลังจากนั้นสัญญาณที่ได้ก็จะนำมาทำการสแกรมเบิลเหมือนในกรณีของช่องสัญญาณเพจจิง หากแต่ในกรณีของช่องสัญญาณทราฟฟิกจะมีการใช้ long code mask ที่แตกต่างไป โดยโครงสร้างที่ใช้มีลักษณะดังแสดงในรูปที่ 2.21 ส่วนของ Permuted ESN นั้นได้มาจากการสลับตำแหน่งของบิทใน ESN แสดงในรูปที่ 2.22 โดยมีลำดับดังนี้คือ

Permuted ESN = $\{E_0, E_{31}, E_{22}, E_{13}, E_4, E_{26}, E_{17}, E_8, E_{30}, E_{21}, E_{12}, E_3,$

$E_{25}, E_{16}, E_7, E_{29}, E_{20}, E_{11}, E_2, E_{24}, E_{15}, E_6, E_{28}, E_{19},$

$E_{10}, E_1, E_{23}, E_{14}, E_5, E_{27}, E_{18}, E_9\}$

เหตุผลที่ต้องมีการสลับตำแหน่งก็เพื่อลดความสัมพันธ์ (correlation) ระหว่าง long code แต่ละชุดข้อมูลที่ส่งผ่านช่องสัญญาณทราฟฟิกนี้ จะได้รับการปกป้องจากการดักฟังได้โดยการใช้ private long code mask แทน public long code mask ที่ได้อธิบายไว้ ซึ่งโดยปกติแล้วการที่ระบบจะเปลี่ยนไปใช้ private code mask ได้นั้นจะต้องมีการทำกระบวนการ authentication ก่อนเสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| 41          32 | 31          0 |
|---|---|
| 1100011000 | Permuted ESN |

**รูปที่ 2.21** รายละเอียดรูปแบบของ forward traffic channel public long code mask

| 31      24 | 23      18 | 17      0 |
|---|---|---|
| MFR Code | FCC Allocation Number | Serial Number |

**รูปที่ 2.22** รายละเอียดของ ESN

นอกจากนี้ระบบยังมีการแทรกบิทพิเศษลงในช่องสัญญาณทราฟฟิก สำหรับใช้ในการควบคุมกำลังส่งของเครื่องโทรศัพท์ให้อยู่ในระดับเหมาะสม โดยในการส่งนั้นจะส่งเพียง 1 บิท ทุกๆ ช่วงเวลา 1.25 ms(800 bps) หากบิทที่ส่งออกมีค่าเป็น 0 ก็หมายความว่าเครื่องโทรศัพท์จะต้องทำการปรับค่าเฉลี่ยกำลังของกำลังส่งขึ้นหนึ่งระดับ แต่หากบิทที่ส่งออกมีค่าเป็น 1 เครื่องโทรศัพท์ก็จะต้องปรับลดกำลังส่งลง

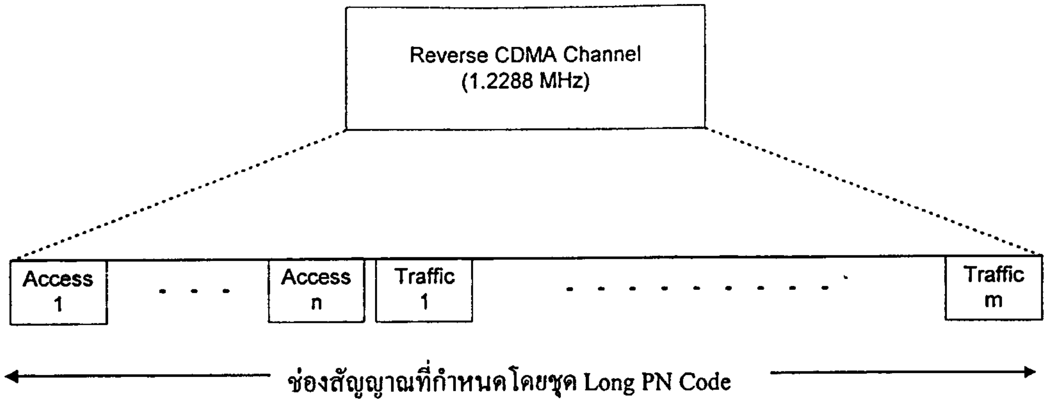
**2.2.10 การส่งสัญญาณจากโทรศัพท์ไปหาสถานีฐาน**

สำหรับการส่งสัญญาณในทิศทางจากโทรศัพท์เคลื่อนที่ไปยังสถานีฐาน หรือที่เรียกว่า reverse link นั้นมีความซับซ้อนกว่าในทิศทางตรงกันข้าม เพราะกำลังของสัญญาณที่เครื่องโทรศัพท์ส่งออกมีขนาดอ่อนกว่า เมื่อเทียบกับกำลังของสัญญาณที่ส่งออกจากสถานีฐาน ดังนั้นจึงต้องมีการเข้ารหัสช่องสัญญาณที่ซับซ้อนกว่า อีกทั้งที่สถานีฐานยังมีการใช้สายอากาศหลายชุดเพื่อช่วยเพิ่มสมรรถภาพในการรับสัญญาณ การส่งสัญญาณในทิศทางนี้มีเพียง 2 ประเภท คือ

- ช่องสัญญาณแอกเซส (Access channel)
- ช่องสัญญาณทราฟฟิกขากลับ (Reverse traffic channel)

ดูลักษณะของการแบ่งช่องสัญญาณในทิศทาง reverse link ได้ในรูปที่ 2.23 สังเกตว่าในทิศทาง reverse link นั้นอาศัยชุดโค้ด long PN เป็นตัวระบุประเภทของช่องสัญญาณ ซึ่งแตกต่างจากในกรณีของ forward link ที่ใช้โค้ด Walsh เป็นตัวบ่งบอกถึงประเภทของช่องสัญญาณ

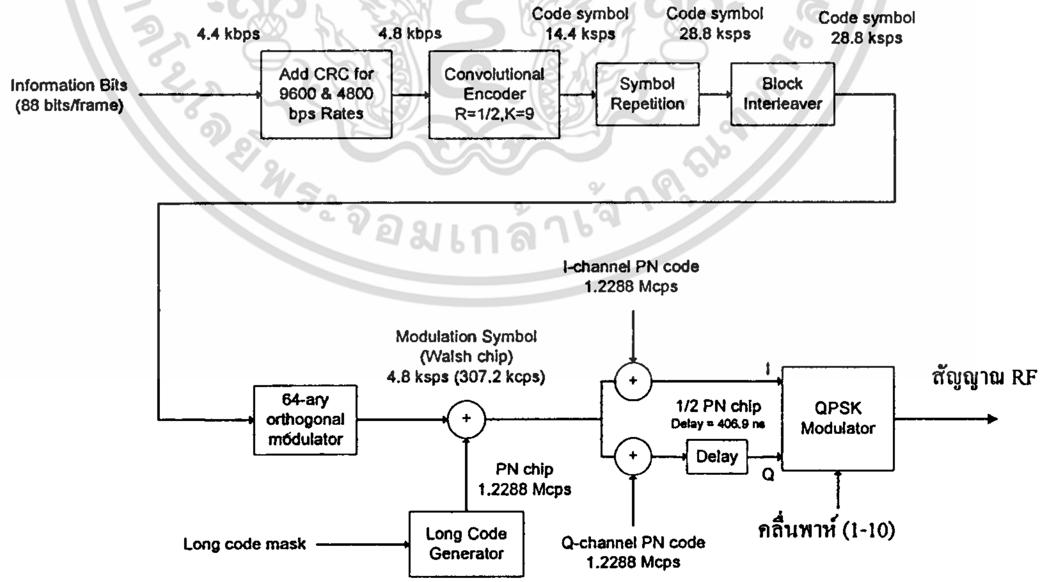
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.23** การจัดช่องสัญญาณที่ส่งออกโดยเครื่องโทรศัพท์เคลื่อนที่

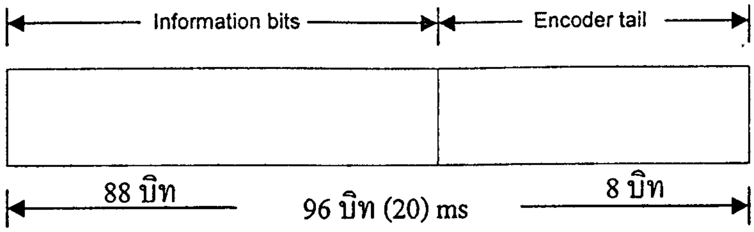
### 2.2.11 ช่องสัญญาณแอกเซส

ในส่วนของช่องสัญญาณแอกเซส (Access channel) มีหน้าที่หลัก คือ ใช้ในการส่งสัญญาณซิกแนลลิงออกไปที่สถานีฐาน เพื่อขอเริ่มการติดต่อในกรณีที่ผู้ใช้ต้องการจะโทรออก และใช้ช่องสัญญาณนี้ในการโต้ตอบกับสัญญาณซิกแนลลิงที่ส่งออกมาจากสถานีฐานผ่านทางช่องสัญญาณเพจจิง อัตราการส่งของช่องสัญญาณ access กำหนดเป็นค่าคงที่ไว้ที่ 4,800 bps โดยรูปแบบในการส่งนั้นใช้เฟรมขนาดเท่ากับ 20 ms และในแต่ละเฟรมนั้นจะต้องเริ่มที่เวลาที่ตรงกับเวลาของระบบที่สามารถหารด้วย 20 ms ลงตัว สัญญาณที่จะส่งออกผ่านช่องสัญญาณ access จะต้องผ่านกระบวนการต่าง ๆ ดังที่แสดงในรูปที่ 2.24



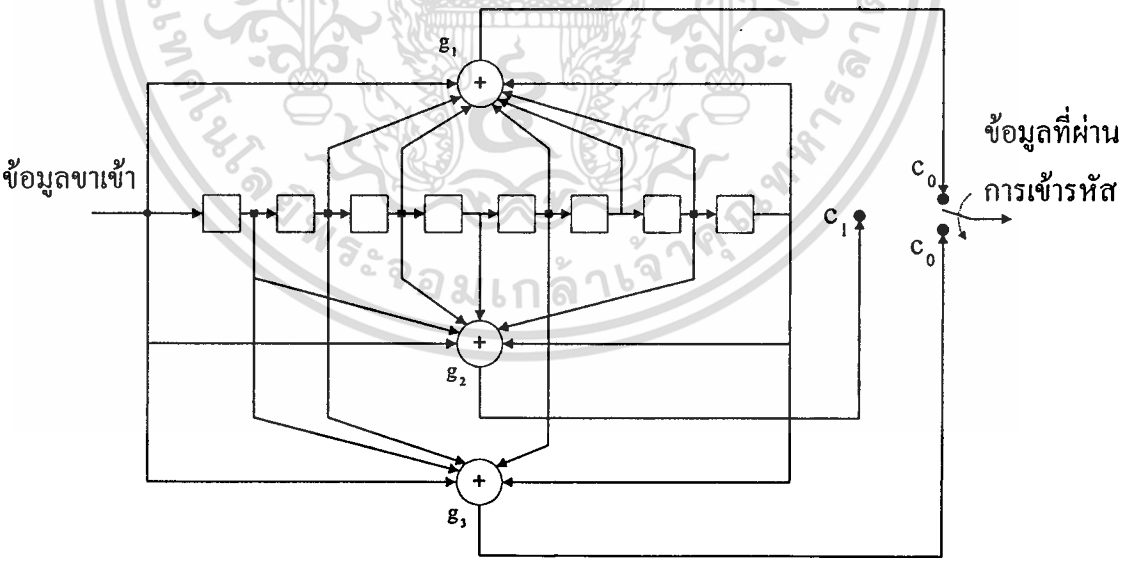

**รูปที่ 2.24** ขั้นตอนการสร้างสัญญาณสำหรับส่งผ่านช่องสัญญาณแอกเซสในทิศทางที่จะส่งจากโทรศัพท์เคลื่อนที่ไปยังสถานีฐาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.25** โครงสร้างของเฟรมที่ส่งออกผ่านช่องสัญญาณแอกเซส

โครงสร้างของเฟรมแบ่งออกเป็น 2 ส่วน คือ information bits จำนวน 88 บิท และ encoder tail อีก 8 บิท ดูรูปที่ 2.25 ประกอบ ดังนั้นอัตราบิทที่ส่งมีค่าเท่ากับ (88+8)/20 ms = 4,800 bps จากนั้นจะนำไปผ่านการเข้ารหัสแบบคอนโวลูชั่นทีละ 88 บิท โดยอาศัยวงจรในรูปที่ 2.26 ซึ่งเป็นวงจรเข้ารหัสที่มีอัตราการเข้ารหัสเท่ากับ 1/3 และมี constraint length เท่ากับ 9 ส่วนของ encode tail จำนวน 8 บิท จะกำหนดให้มีค่าเป็นศูนย์ทั้งหมดก่อนที่จะทำการเข้ารหัส จากในรูปจะเห็นว่าวงจรเข้ารหัสที่ใช้ประกอบด้วยชิฟต์รีจิสเตอร์ทั้งหมด 8 ตัว วงจรบวกมอดูโล 2 อีก 3 ตัว และมีอุปกรณ์สวิตช์อีก 1 ตัวซึ่งทำหน้าที่วนรอบรับค่าจากผลลัพธ์ที่ได้จากวงจรบวกมอดูโล 2 ทีละตัว หลังจากที่ผ่านการเข้ารหัสแล้วอัตราบิทข้อมูลจะสูงขึ้น 3 เท่ากลายเป็น 14,400 bps




**รูปที่ 2.26** โครงสร้างวงจรเข้ารหัสแบบคอนโวลูชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่น จงเข้ารหัสข้อมูลสำหรับช่องสัญญาณ access จำนวน 80 บิท ต่อไปนี้โดยใช้วงจรการเข้ารหัสแบบคอนโวลูชั่นตามรูปที่ 2.26

1100101111001101010110100100111011011010011 1

1010111100100110111010101100110101000110000 1

วิธีทำ

ผลที่ได้จากการเข้ารหัสคือ

111 100 110 011 011 100 011 000 010 101 110 010 101 001 110 001

110 010 100 010 110 111 111 100 100 111 011 001 010 000 011 100

000 010 101 011 101 000 010 101 011 111 100 101 100 100 100 110

111 010 110 011 001 010 101 110 101 100 111 100 000 110 001 001

100 011 110 010 101 110 111 000 000 101 000 110 110 010 100 101

010 110 111 101 101 011 111 110 001 100 001 010 101 100 110 111

เนื่องจากอัตราการส่งที่ใช้มีค่าเพียง 4,800 bps ดังนั้นจึงต้องมีการส่งบิทข้อมูลเดิมซ้ำออกไปทั้งหมด 2 ครั้งติดต่อกัน จากนั้นจึงทำการส่งต่อไปเข้าวงจรอินเตอร์ลีฟ จากตัวอย่างข้างต้นหลังจากที่ผ่านการเพิ่มบิทซ้ำเข้าไปก็จะได้บิทข้อมูลที่มีลักษณะดังต่อไปนี้

111111 110000 111100 001111 001111 110000 001111 000000

001100 110011 111100 001100 110011 000011 111100 000011

111100 001100 110000 001100 111100 111111 111111 110000

110000 111111 001111 000011 001100 000000 001111 110000

000000 001100 110011 001111 110011 000000 001100 110011

001111 111111 110000 110011 110000 110000 110000 111100

111111 001100 111100 001111 000011 001100 110011 111100

110011 110000 111111 110000 000000 111100 000011 000011

110000 001111 111100 001100 110011 111100 111111 000000

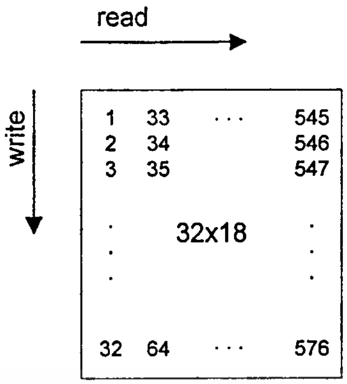
000000 110011 000000 111100 111100 001100 110000 110011

001100 111100 111111 110011 110011 001111 111111 111100

000011 110000 000011 001100 110011 110000 111100 111111

จุดมุ่งหมายหลักของการทำอินเตอร์ลีฟก็เพื่อจัดการกับปัญหาของความผิดพลาดของบิทข้อมูลแบบเบริสต์ การทำอินเตอร์ลีฟจะนำข้อมูลทีละ 20 ms ซึ่งจะมีทั้งสิ้น 576 บิท มาจัดเรียงเป็น 32 แถว และ 18 คอลัมน์ดังที่แสดงในรูปที่ 2.27 ในการเขียนบิทข้อมูลลงในหน่วยความจำก็จะกระทำในแนวตั้ง แต่ว่าในการอ่านออกก็จะอ่านออกทีละแถวในแนวนอน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.27** การทำอินเตอร์ลีฟสำหรับ reverse link

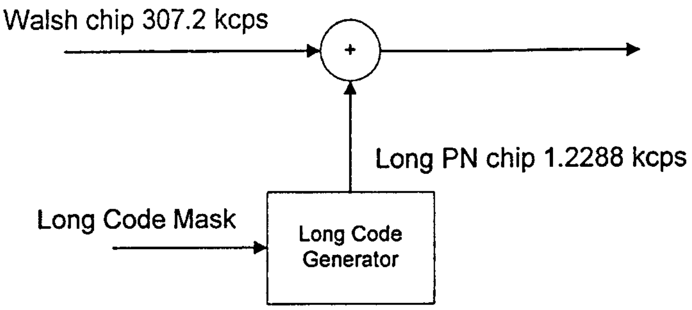
หากใช้ตัวอย่างบิทข้อมูลข้างต้นเราจะได้ชุดบิทข้อมูลที่มีลักษณะดังนี้

100111 000111 110011 000010 100011 001100 011110 101110
111111 000111 110001 100100 101010 000110 011011 101100
111110 101011 101100 100110 100111 110000 111110 100101
100110 001101 001110 011000 010000 000100 000011 010010
101010 110111 011000 011001 110011 011000 010110 111011
110110 010111 100000 011101 100101 111110 000011 101001
100111 000111 110011 000010 100011 001100 011110 101110
111111 000111 110001 100100 101010 000110 011011 101100
111110 101011 101100 100110 100111 110000 111110 100101
100110 001101 001110 011000 010000 000100 000011 010010
101010 110111 011000 011001 110011 011000 010110 111011
110110 010111 100000 011101 100101 111110 000011 101001

จากบิทข้อมูลที่ได้ในขณะนี้มีอัตราบิทอยู่ที่ 28.8 ksps ในขั้นตอนต่อไปนี้จะแปลงบิทข้อมูลทีละ 6 บิท ให้กลายเป็น 64 บิท ที่มีรูปแบบตามรหัส Walsh ผลที่ได้คือข้อมูลที่มีอัตราส่งเท่ากับ (28.8/6) x 64 = 307.2 kcps ดูตัวอย่างของการเข้ารหัสข้อมูล

| ข้อมูล | index | 64-symbol putput |
|---|---|---|
| 100111 | 57 | 0101010110101010101010100101010110101010010101010101010110101010 |
| 000111 | 56 | 0000000011111111111111110000000011111111000000000000000011111111 |
| 110011 | 51 | 0110011001100110100110011001100110011001100110010110011001100110 |
| 000010 | 16 | 0000000000000000111111111111111100000000000000001111111111111111 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

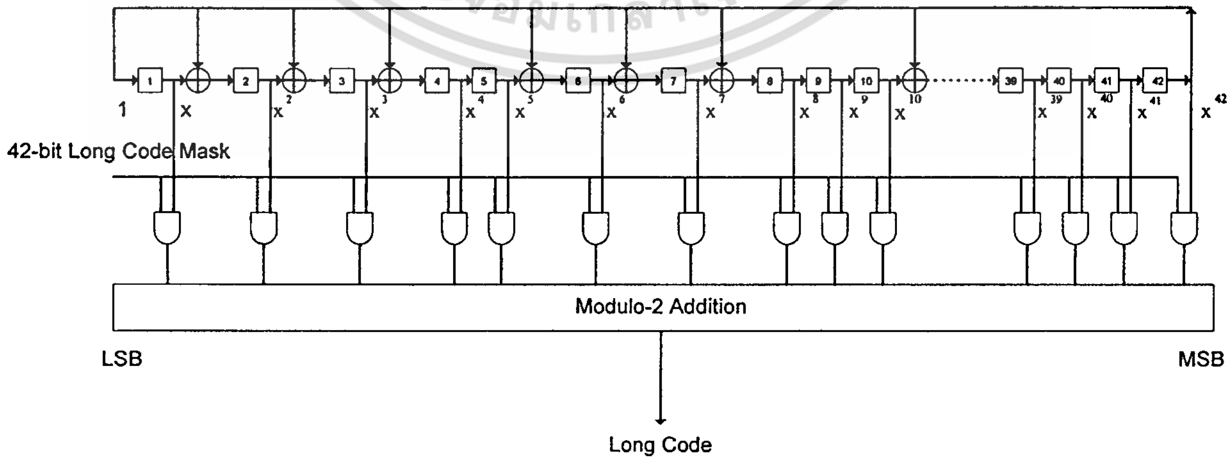



**รูปที่ 2.28** การสเปรดสัญญาณด้วย Long PN Code

ขั้นตอนถัดมาจะนำข้อมูลมาทำการสเปรดสัญญาณออกด้วยชุด Long PN code ที่มีคาบความยาวเท่ากับ $2^{42}-1$ โดยอาศัยวงจรชิฟต์รีจิสเตอร์ที่ต่อกันเป็นโพลิโนเมียล

$$p(x) = 1 + x + x^2 + x^3 + x^5 + x^6 + x^7 + x^{10} + x^{16} + x^{17} + x^{18} + x^{19} + x^{21} + x^{22} + x^{25} + x^{26} + x^{27} + x^{31} + x^{33} + x^{35} + x^{42}$$

ในการสเปรดสัญญาณนั้นกระทำโดยอาศัยวงจรบวกแบบมอดูโล 2 ดังที่แสดงในรูปที่ 2.29 ในส่วนของวงจร Ling Code Generator นั้นประกอบด้วย 2 ส่วนหลัก คือ ชุดชิฟต์รีจิสเตอร์จำนวน 42 ตัวที่ต่อเชื่อมกันตามโพลิเนียล p(x) และส่วนของ 42-bit Access Long Code Mask ที่ทำหน้าที่แยกแยะและบ่งบอกถึงพารามิเตอร์ที่สำคัญในการติดต่อ ซึ่งประกอบด้วย หมายเลขของช่องสัญญาณ access หมายเลขของช่องสัญญาณเพจจิงที่กำลังติดต่ออยู่ด้วย หมายเลขของสถานีฐานที่ต้องการจะติดต่อด้วย และหมายเลขของ PILOT-PN ที่ใช้งานอยู่ ดูโครงสร้างของ 42-bit Access Long Code Mask ได้รูปที่ 2.30




**รูปที่ 2.29** Long code generator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากที่ได้ผ่านกระบวนการสเปรดสัญญาณด้วย long PN code แล้ว ก็นำสัญญาณที่ได้ไปสเปรดด้วย short PN code ทั้ง I และ Q สังเกตว่าในการมอดูเลตสัญญาณจะต่างไปจากกรณี forward link เพราะมีการดีเลย์ข้อมูลในส่วนของ Q เป็นเวลาครึ่งหนึ่งของคาบเวลาหนึ่งชิปส์ ลักษณะการมอดูเลตแบบนี้มีชื่อเรียกว่า OQPSK (Offset QPSK)

| 41 … 33 | 32 … 28 | 27 … 25 | 24 … 9 | 8 … 0 |
|---|---|---|---|---|
| 110001111 | ACN | PCN | BASED-ID | PILOT-PN |

MSB LSB

ACN : Access Channel Number

PCN : Paging Channel Number

BASD-ID : Based Station Identification

PILOT-PN : PN offset for Forward CDMA Channel

**รูปที่ 2.30** Access channel long code mask

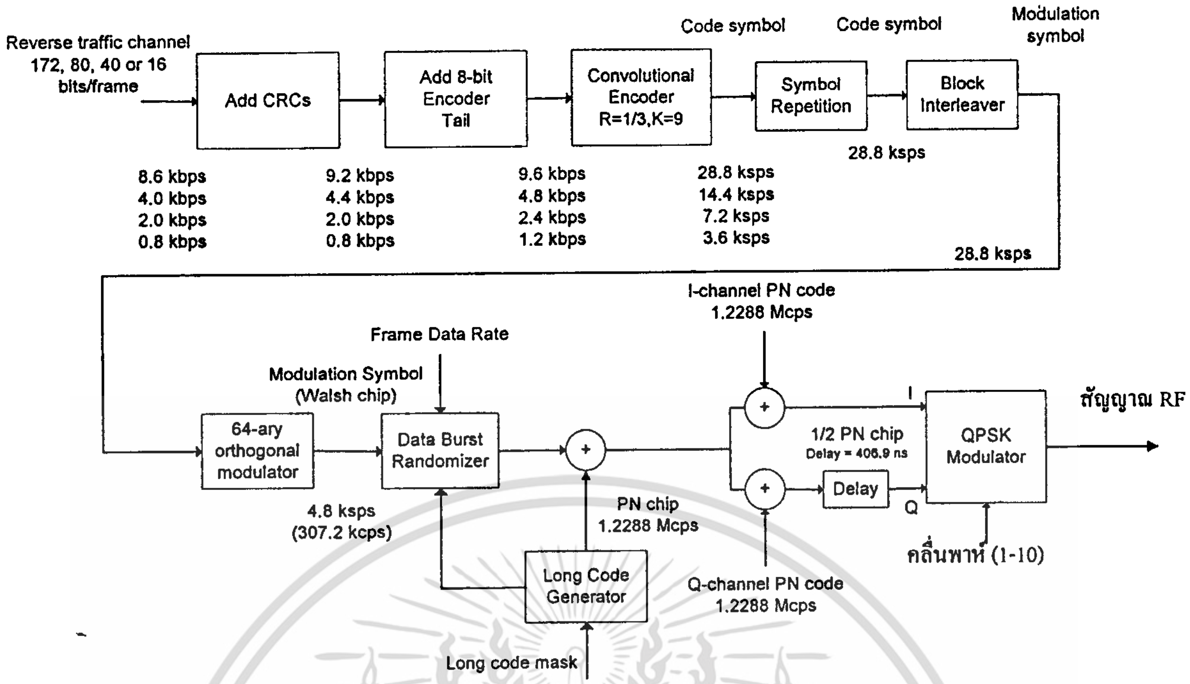
**2.2.12 ช่องสัญญาณทราฟฟิกขากลับ**

รูปที่ 2.31 แสดงโครงสร้างขั้นตอนของการสร้างช่องสัญญาณทราฟฟิกที่จะส่งจากโทรศัพท์เคลื่อนที่ไปยังสถานีฐาน วงจรเข้ารหัสสัญญาณเสียงที่ใช้ใน IS-95 จะกำเนิดแพ็กเกต (packet) จำนวนหนึ่งแพ็กเกตทุก ๆ 20 ms โดยที่ขนาดความยาวของแพ็กเกตมีค่าไม่คงที่ทั้งนี้ขึ้นอยู่กับปริมาณการพูดของผู้ใช้โทรศัพท์ โดย IS-95 ได้แบ่งชนิดของแพ็กเกตออกเป็นทั้งหมด 4 แบบ ดังที่แสดงไว้ในตารางที่ 2.2 สัญญาณที่ได้จากที่แสดงในตารางที่ 2.2 จะนำไปเข้ารหัสแบบคอนโวลูชั่นที่มีอัตราการเข้ารหัสเท่ากับ 1/3 และมีค่า constraint length เท่ากับ 9 ซึ่งทำให้จำเป็นต้องเพิ่ม code tail อีก 8 บิท ต่อท้ายดังที่แสดงในตารางเดียวกัน สังเกตว่ากรณีที่อัตราการส่งข้อมูล 4,800 และ 9,600 bps จะมีการใช้โค้ด CRC เพิ่มเติมพิเศษเพื่อช่วยในการตรวจสอบว่ามีความผิดพลาดในข้อมูลเกิดขึ้นหรือไม่ วงจรเข้ารหัสคอนโวลูชั่นที่ใช้มีโครงสร้างเช่นเดียวกันกับในกรณีของช่องสัญญาณแอกเซสดังที่แสดงไว้ในรูปที่ 2.29

**ตารางที่ 2.2** โครงสร้างของแพ็กเกต

| Information (bits) | Signaling (bits) | CRC (bits) | Code tail (bits) | Total (bits) | Bit rate (bps) |
|---|---|---|---|---|---|
| 171 | 1 | 12 | 8 | 192 | 9,600 |
| 80 | 0 | 8 | 8 | 96 | 4,800 |
| 40 | 0 | 0 | 8 | 48 | 2,400 |
| 16 | 0 | 0 | 8 | 24 | 1,200 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 2.31 ขั้นตอนการสร้างสัญญาณทราฟฟิกในทิศทางจากโทรศัพท์เคลื่อนที่ไปยังสถานีฐาน

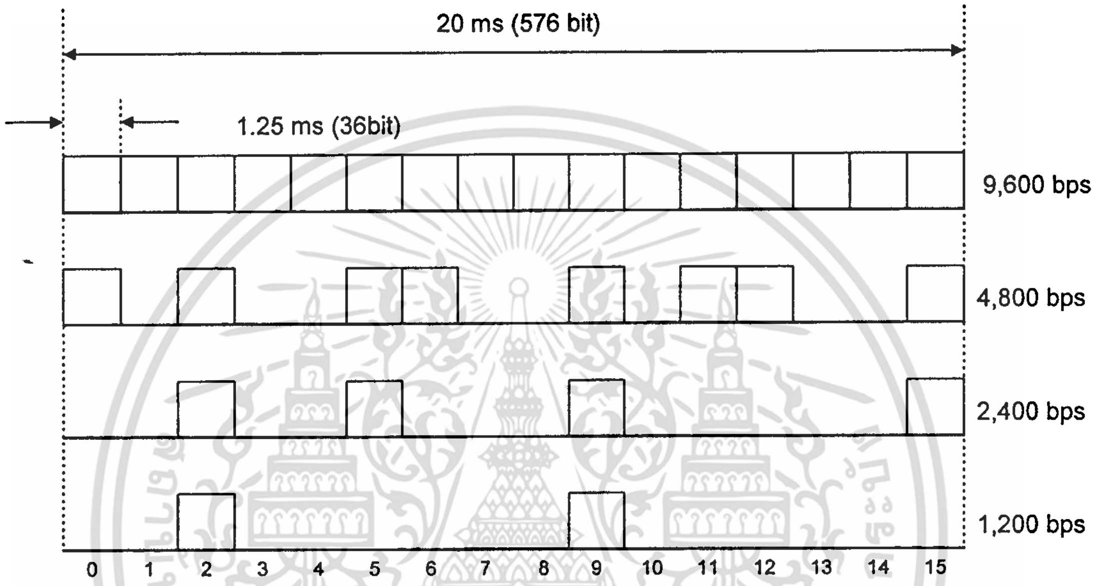
สัญญาณที่ได้จากวงจรเข้ารหัสช่องสัญญาณคอนโวลูชั่นนั้นจะนำไปเข้าวงจร symbol repetition ซึ่งทำหน้าที่เพิ่มบิทข้อมูลซ้ำเพื่อให้อัตราบิทที่ขาออกมีค่าเท่ากับ 28.8 kbps เพราะฉะนั้นในกรณีที่อัตราบิทขาเข้ามีอัตราบิทเป็น R จะมีการส่งบิทนั้นซ้ำอีก 9,600/R-1 ครั้ง เช่น ถ้าอัตราบิทขาเข้ามีค่าเป็น 4,800 bps จะมีการส่งแต่ละบิทซ้ำอีกหนึ่งบิท จากนั้นสัญญาณที่ได้นี้จะนำไปทำการอินเตอร์ลีฟด้วยช่วงความลึกเท่ากับความยาวของหนึ่งเฟรมหรือ 20 ms เนื่องจากอัตราบิทก่อนการทำอินเตอร์ลีฟมีค่าเท่ากับ 28.8 kbps ดังนั้นในช่วงเวลา 20 ms จะมีจำนวนบิททั้งสิ้น 576 บิท (32x18) ซึ่งจะนำมาทำการอินเตอร์ลีฟด้วยวิธีการเดียวกับในกรณีของช่องสัญญาณแอกเซสที่แสดงไว้ก่อนหน้านี้ในรูปที่ 2.27

ในลำดับถัดมาคือในส่วนของวงจร 64-ary orthogonal modulator จะแปลงข้อมูลทีละ 6 บิทให้เป็น Walsh code ที่มีความยาวเท่ากับ 64 ชิปส์ การที่ใช้ 6 บิท ก็เนื่องมาจากการที่ Walsh code ที่ใช้มีทั้งหมด $2^6 = 64$ ชุดนั่นเอง ผลที่ได้ก็คือข้อมูลที่มีอัตราชิปส์เท่ากับ 28,800/6 x 64 = 307.2 k Walsh cps (kcps) หรือได้ 4.8 k Walsh sps สังเกตว่าการนำ Walsh code มาใช้งานในการส่งข้อมูลในทิศทางจากเครื่องโทรศัพท์ไปยังสถานีฐาน (reverse link) แตกต่างจากการใช้งานโค้ดนี้ในการส่งข้อมูลในทิศตรงข้าม (forward link) อย่างชัดเจน ในกรณีของ forward link โค้ดเหล่านี้ใช้สำหรับแบ่งแยกช่องสัญญาณออกเป็น 64 ช่อง

สัญญาณที่ได้จากวงจร 64-ary orthogonal modulator จะนำไปเข้าวงจร Data burst randomizer ซึ่งทำหน้าที่เปิดและปิดเกตเพื่อตัดที่สัญญาณที่ซ้ำซ้อนออกไป ซึ่งหมายความว่าวงจร

เอกสารนี้เป็นเอกสารที่ใช้ในเฉพาะหรือการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้เพื่อประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนนี้จะมีผลกระทบต่อสัญญาณก็เฉพาะกรณีที่อัตราบิทของข้อมูลมีค่าต่ำกว่า 9,600 bps เท่านั้น โดยจะมีการเปิดและปิดเกตในทุก ๆ ช่วงเวลา 1.25 ms หรือทุก ๆ 36 บิท ช่วงเวลาการเปิดปิดจะมีลักษณะดังที่แสดงในรูปที่ 2.32 จะเห็นได้ว่า duty cycle ของการส่งที่อัตราบิท 9,600, 4,800, 2,400 และ 1,200 bps จะมีค่าเท่ากับ 100, 50, 25, และ 12.5% ตามลำดับ การที่ไม่มีการส่งสัญญาณออกไปในบางช่วงเวลานั้นมีประโยชน์เพราะจะช่วยลดปริมาณสัญญาณรบกวนภายในระบบลงได้ ซึ่งจะส่งผลโดยตรงให้ระบบสามารถรองรับจำนวนผู้ใช้บริการได้มากขึ้น




**รูปที่ 2.32** ตัวอย่างการเปิดปิดเกตสำหรับการส่งที่อัตราบิทที่ค่าต่าง ๆ

ในขั้นตอนถัดมาสัญญาณที่ได้จะนำไปสเปรดออกด้วยโค้ด long PN ที่มีอัตราชิปส์เท่ากับ 1.2288 Mcps โดยอาศัยวงจรแบบเดียวกับในกรณีของช่องสัญญาณแอกเซส ซึ่งการสเปรดสัญญาณนั้นโทรศัพท์แต่ละเครื่องจะใช้ชุด public long mask ที่แตกต่างกัน ดูโครงสร้างรายละเอียดของ public long code mask ที่ใช้ในช่องสัญญาณ forward traffic ได้ในรูปที่ 2.33

| 41 … 32 | 31 … 0 |
|---|---|
| 1100011000 | Permuted ESN |

**รูปที่ 2.33** รายละเอียดรูปแบบของ Public long code mask

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2.2.13 คุณลักษณะและข้อดีของระบบ CDMA**

- โดยธรรมชาติแล้วในระหว่างการสนทนาของมนุษย์จะมีเพียงประมาณ 35% ของเวลาทั้งหมดเท่านั้นที่มีสัญญาณเสียงของผู้พูด ส่วนเวลาที่เหลือจะเป็นการฟัง ในระบบ CDMA นั้นเนื่องจากผู้ใช้บริการทุกคนจะถูกจัดให้ใช้ช่องสัญญาณช่องเดียวกันหมด ดังนั้นถ้ามีผู้ใช้บางส่วนที่ไม่มีการพูดก็หมายถึงว่าปริมาณสัญญาณรบกวนระหว่างกันของผู้ใช้ ณ เวลานั้น ๆ ก็ลดลงด้วย ซึ่งโดยหลักการแล้วอาจจะลดลงได้ถึง 65 % นั่นหมายความว่าเราสามารถเพิ่มขนาดความจุของระบบขึ้นถึงประมาณ 3 เท่า
- ในการส่งข้อมูลที่อัตราการส่งสูง ๆ เช่น มากกว่า 10 kbps มาก ๆ นั้น ระบบ FDMA และ TDMA จำเป็นต้องอาศัยวงจรอีควอไลเซอร์ในการลดสัญญาณที่เรียกว่า intersymbol interference ซึ่งเกิดจากการแผ่ของสัญญาณเนื่องจากการสเปรดของดีเลย์ ในขณะที่วงจรภาครับของระบบ CDMA ไม่มีความจำเป็นต้องใช้วงจรอีควอไลเซอร์สำหรับการปรับแก้รูปสัญญาณ แต่ต้องอาศัยวงจรคอร์รีเลเตอร์ (correlator) เพื่อใช้ในการดีสเปรด (despread) สัญญาณกลับมา ซึ่งวงจรคอร์รีเลเตอร์นั้นมีความซับซ้อนน้อยกว่าวงจรอีควอไลเซอร์
- ระบบ CDMA ใช้วงจรรับส่งวิทยุเพียงชุดเดียวสำหรับครอบคลุมบริเวณ ๆ หนึ่ง ดังนั้นจึงง่ายและสะดวกในการติดตั้ง อีกทั้งยังประหยัดเนื้อที่ในการจัดเก็บอุปกรณ์ด้วย
- เนื่องจากในแต่ละเซลล์ใช้คลื่นวิทยุความถี่เดียวกันจึงไม่มีความจำเป็นต้องมีการแฮนด์ออฟจากความถี่หนึ่งไปยังอีกค่าความถี่หนึ่งในกรณีที่ตัวโทรศัพท์เคลื่อนที่จากเซลล์หนึ่งไปสู่เซลล์ข้างเคียง การแฮนด์ออฟแบบนี้เรียกว่า soft handoff ซึ่งต่างจาก hard handoff ตรงที่ไม่มีความจำเป็นต้องเปลี่ยนความถี่ของคลื่นพาห์
- ในระบบ TDMA มีความจำเป็นที่จะต้องเว้นช่องว่างที่เรียกว่า guard time ไว้ขนาดหนึ่งเพื่อป้องกันมิให้ข้อมูลที่อยู่คนละไทม์สล็อตเกิดการทับกัน ส่วนระบบ CDMA นั้นไม่มีความจำเป็นที่ต้องมี guard time เลย เพราะฉะนั้นช่วงเวลาตรงนี้สามารถนำมาใช้เพิ่มประสิทธิภาพการส่งข้อมูลให้สูงขึ้นได้
- ระบบ CDMA ไม่มีความจำเป็นต้องจัดแบ่งความถี่สำหรับการใช้งานที่ต่างกันสำหรับเซลล์ที่อยู่ใกล้กัน นั่นคือเซลล์ที่อยู่ติดกันก็สามารถใช้คลื่นความถี่เดียวกันได้ ซึ่งต่างจากระบบอื่น เช่น FDMA เพราะระบบเหล่านี้จะต้องมีการจัดแบ่งความถี่ออกเป็นกลุ่มๆ และเซลล์ที่ใช้คลื่นความถี่กลุ่มเดียวกันจำเป็นจะต้องอยู่ห่างกันเพียงพอเพื่อลดผลกระทบของ Co-Channel Interference ดังนั้นการใช้ประโยชน์จากสเปรกตรัมความถี่ที่มีอยู่อย่างจำกัดของระบบ CDMA จึงมีประสิทธิภาพที่ดีกว่า ให้ขนาดความจุที่สูงกว่าและรองรับจำนวนผู้ใช้บริการได้มากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ปัญหาและผลกระทบของเฟดดิงต่อการส่งสัญญาณแบบแถบความถี่กว้าง (wideband) มีน้อยกว่าเมื่อเทียบกับการส่งข้อมูลแบบแถบความถี่แคบ (narrowband) ดังนั้นการออกแบบระบบเพื่อจัดการกับปัญหาของเฟดดิงจึงมีความซับซ้อนน้อยกว่า

**2.2.14 ขนาดความจุของระบบ CDMA**

ในระบบ TDMA/FDMA นั้นขนาดความจุของระบบขึ้นอยู่กับขนาดของสเปกตรัมความถี่ที่มีอยู่ และขนาดแบนด์วิดท์ของสัญญาณแต่ละช่องเป็นหลัก โดยการคำนวณหาขนาดความจุนั้นค่อนข้างง่ายและตรงไปตรงมา ยกตัวอย่างเช่นในกรณีของระบบ FDMA ขนาดความจุของแต่ละเซลล์คำนวณจากขนาดสเปกตรัมความถี่ที่มีอยู่หารด้วยขนาดแบนด์วิดท์ของช่องสัญญาณเสียงแต่ละช่อง และสำหรับระบบที่มีการจัดวางเซลล์เป็นกลุ่มละ 7 เซลล์เพื่อลดปัญหา Co-Channel Interteference ก็ให้นำผลที่ได้มาหารด้วย 7 ผลลัพธ์ที่ได้คือขนาดความจุของแต่ละเซลล์ภายในระบบ

แต่สำหรับในกรณีของ CDMA ขนาดความจุของระบบขึ้นอยู่กับระดับของสัญญาณรบกวน (interference) เป็นหลัก นั่นคือระบบจะยังคงสามารถทำงานได้โดยปกติตราบใดที่ปริมาณของสัญญาณรบกวนยังอยู่ในระดับที่ไม่สูงเกินไป สัญญาณรบกวนที่กล่าวถึงนี้เกิดจากผู้ใช้อื่น ๆ ที่อยู่ในบริเวณรอบข้างที่มีการใช้งานที่ความถี่เดียวกัน

พิจารณาเซลล์หนึ่งเซลล์ที่มีผู้ใช้ทั้งหมดจำนวน k คน กำลังของสัญญาณรบกวนจากผู้ใช้อื่น ๆ ทั้งหมดภายในเซลล์มีค่าเท่ากับ P(k-1) เมื่อ P คือกำลังของสัญญาณที่สเปรดแล้ว ค่าความหนาแน่นของสัญญาณรบกวน $N_o$ มีค่าเท่ากับ P(k-1)/W โดย W คือค่าแบนด์วิดท์ของช่องสัญญาณที่ใช้ พลังงานของสัญญาณข้อมูลหนึ่งบิท $E_b$ มีค่าเท่ากับ P/R โดย R คือค่าอัตราการส่ง (bps)
ดังนั้น

$$E_b/N_o = (P/R)/(P(k-1)/W) = W/(R(k-1))$$

ถ้ากำหนดให้ Processing gain G = W/R

จะได้ $k = 1+G/(E_b/N_0)$

สมการนี้คิดเฉพาะผลของสัญญาณรบกวนจากผู้ใช้ในเซลล์เดียวกันเท่านั้น เพราะฉะนั้นถ้าคิดถึงผลกระทบของเซลล์รอบข้างด้วยขนาดของสัญญาณรบกวนจะเพิ่มขึ้นประมาณ 1.5 เท่า ดังนั้นจะได้

$$k = 1+G/[1.5(E_b/N_0)]$$

ถ้าหากสายอากาศที่ใช้เป็นแบบที่กำหนดทิศทางการส่งให้ครอบคลุมเพียง 1/3 ของพื้นที่ทั้งหมดได้ (sectoring) ก็จะสามารถลดสัญญาณรบกวนลงได้ ซึ่งเราสามารถประมาณการลดทอนของสัญญาณรบกวนได้โดยการคูณสัญญาณรบกวนกับค่าตัวประกอบ S ซึ่งมีค่าประมาณ 2.4 เท่า นอกจากนี้ในระบบ CDMA สามารถใช้ประโยชน์จากการที่ผู้ใช้โทรศัพท์จะมีช่วงเวลาที่ไม่พูดเช่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ช่วงเวลาที่ฟังอยู่ ช่วงที่ผู้ใช้เงียบไปนี้ก็จะไม่สร้างสัญญาณรบกวนต่อผู้ใช้อื่น ๆ เพราะฉะนั้นค่าสัญญาณรบกวนก็ลดลงด้วยค่าตัวประกอบ V โดย 1/V คือเปอร์เซ็นต์ของเวลาที่ไม่มีสัญญาณเสียง เมื่อคิดถึงผลกระทบขององค์ประกอบต่าง ๆ ที่กล่าวมานี้ก็จะได้

$k = 1+G/[1.5(E_b/N_0)]SV$

เช่น จงคำนวณหาจำนวนของผู้ใช้ในหนึ่งเซลล์ที่ระบบ CDMA ระบบหนึ่งสามารถรองรับได้ โดยกำหนดให้แบนด์วิดท์ที่ใช้มีค่าเท่ากับ 1.228 MHz อัตราการส่งข้อมูลที่ 9.6 kbps และค่า $E_b/N_o$ ที่ระบบยังสามารถทำงานได้ดีมีค่าเท่ากับ 10 dB ทั้งนี้ให้ใช้ค่า S = 2.4 และ V = 1/0.6

วิธีทำ

$G = 1.228/9.6 = 127.9$

$E_b/N_o$ = 10 dB = 10 เท่า

จะได้ $k = 1 + [127.9/(1.5x10)] x 2.4/0.6 = 35.1$

จำนวนผู้ใช้ทั้งหมดเท่ากับในหนึ่งเซลล์ที่มีจำนวนช่องสัญญาณ 10 ช่องคือ 10 x 35.1 = 351

### 2.2.15 การคำนวณค่า Co-Channel Interference ในระบบ CDMA

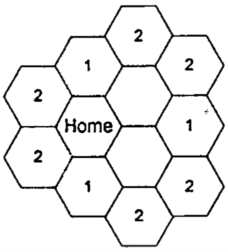
ค่าความจุของระบบโทรศัพท์เคลื่อนที่โดยปกติแล้วจะถูกจำกัดโดยค่าอัตราส่วนระหว่างกำลังของสัญญาณข้อมูลต่อกำลังของสัญญาณรบกวนจากเซลล์รอบข้างหรือที่เรียกว่าค่า Carrier to Interference (C/I) สำหรับระบบที่ใช้วิธี FDMA นั้นจะลดปัญหานี้ลงด้วยการแบ่งความถี่ออกเป็นกลุ่ม ๆ และจัดสรรความถี่ที่ไม่ซ้ำกันให้กับเซลล์ที่อยู่ติดกัน ซึ่งการจัดความถี่แบบนี้มีผลทำให้ความจุของระบบลดลง ยกตัวอย่างเช่นในกรณีของ AMPS มีการแบ่งความถี่ออกเป็น 7 กลุ่ม สำหรับใช้งานซึ่งทำให้จำนวนคลื่นความถี่ที่ใช้ในแต่ละเซลล์จะลดลง 7 เท่า แต่สำหรับระบบ CDMA นั้นไม่มีการแบ่งความถี่ออกเป็นกลุ่มนั่นคืออนุญาตให้เซลล์ทุก ๆ เซลล์แม้กระทั่งเซลล์ที่อยู่ติดกันใช้ความถี่เดียวกันได้

พิจารณารูปที่ 2.34 จะเห็นว่าตำแหน่งที่จะให้ค่า C/I ต่ำที่สุดคือจุดมุมที่บริเวณขอบเซลล์ โดยกำหนดให้ระยะทางระหว่างจุดนี้กับจุดกึ่งกลางของเซลล์ (Home) มีค่าเท่ากับ R จากรูปนี้เราสามารถแยกแหล่งกำเนิดสัญญาณรบกวนออกเป็นกลุ่ม ๆ ดังต่อไปนี้

- ภายในเซลล์ที่กำลังพิจารณาจะมีสัญญาณรบกวนจากผู้ใช้อื่น ๆ ทั้งหมดยกเว้นตัวเอง นั่นคือมีทั้งสิ้น k-1 แหล่ง โดยที่แหล่งกำเนิดสัญญาณรบกวนเหล่านี้มีระยะห่างประมาณโดยเฉลี่ยเท่ากับ R
- สัญญาณรบกวนจากผู้ใช้ในเซลล์ข้างเคียงที่ติดกันจำนวนทั้งสิ้น 2 เซลล์ แต่ละเซลล์มีแหล่งกำเนิดสัญญาณรบกวนจำนวน k แหล่ง โดยมีระยะห่างออกไปเท่ากับ R
- สัญญาณจากเซลล์รอบข้างชั้นที่ 1 จำนวน 3 เซลล์ แต่ละเซลล์มีแหล่งกำเนิดสัญญาณรบกวนจำนวน k แหล่ง โดยมีระยะห่างออกไปเท่ากับ 2R

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สัญญาณจากเซลล์รอบข้างชั้นที่ 2 จำนวน 6 เซลล์ แต่ละเซลล์มีแหล่งกำเนิดสัญญาณรบกวนจำนวน k แหล่ง โดยมีระยะห่างออกไปเท่ากับ 2.64R




**รูปที่ 2.34** กลุ่มของเซลล์ที่อยู่รอบ ๆ เซลล์ที่กำลังพิจารณา

**2.2.16 ประเภทของไดเวอร์ซิตีที่มีใช้ในระบบ CDMA**

- ในระบบ CDMA มีการใช้ไดเวอร์ซิตีหลากหลายรูปแบบในการเพิ่มคุณภาพการรับส่งสัญญาณดังต่อไปนี้

- Frequency diversity เนื่องจากขนาดของสเปกตรัมของระบบ CDMA มีความกว้างมาก จึงทำให้ผลกระทบของการสะท้อนของสัญญาณจากหลายทิศทาง หรือที่เรียกว่าเฟดดิง (fading) มีน้อย ทั้งนี้เพราะว่าจะมีสัญญาณเพียงบางส่วนเท่านั้นที่ได้รับการรบกวนในขณะใดขณะหนึ่ง ดังนั้นสัญญาณในระบบ CDMA จึงสามารถรับมือกับปัญหาเฟดดิงได้ดีกว่าระบบอื่น
- Antenna diversity ที่สถานีฐานจะใช้สายอากาศ 2 ชุดในการรับสัญญาณ
- Time diversity เกิดจากการที่ระบบอาศัยวงจรรับแบบ Rake ซึ่งสามารถใช้ประโยชน์จากสัญญาณข้อมูลเดียวกันหลายชุดเดินทางที่มาถึงภาครับด้วยเวลาที่แตกต่างกัน
- Space diversity เกิดขึ้นจากการที่ระบบ CDMA ใช้การแฮนด์ออฟแบบที่เรียกว่า soft handoff ซึ่งการแฮนด์ออฟแบบนี้เครื่องโทรศัพท์เคลื่อนที่สามารถเลือกสัญญาณจากสถานีฐานมากกว่าหนึ่งแห่งที่มีคุณภาพดีกว่ามาใช้งาน

## 2.3 คอนโวลูชันแนลโค้ด [2,11,12]

Elias เป็นบุคคลแรกที่ได้ให้ความสนใจในคอนโวลูชันแนลโค้ด และได้เสนอออกมาสู่สายตาประชาชนในราวปี ค.ศ. 1955 รหัสลักษณะนี้จะแตกต่างไปจากบล็อคโค้ดที่ได้อธิบายไว้บทก่อน ๆ ในกรณีของบล็อคโค้ดนั้นแต่ละบล็อคจะมีรหัสอยู่ n บิทที่ได้จากตัวเข้ารหัสในเวลานั้น ซึ่งรหัส n บิท จะขึ้นกับรหัสของข่าวสาร k บิทในหน่วยของเวลานั้น อย่างไรก็ดีสำหรับคอนโวลูชันแนลโค้ดในแต่ละบล็อคจะมีรหัสอยู่ n บิท ที่ได้จากตัวเข้ารหัส ณ เวลานั้น รหัสนี้ไม่เพียงแต่จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขึ้นกับรหัสของข่าวสาร k บิทในเวลานั้น แต่จะขึ้นกับรหัสของข่าวสารในช่องเวลา N-1 ที่ผ่านมา (N>1)

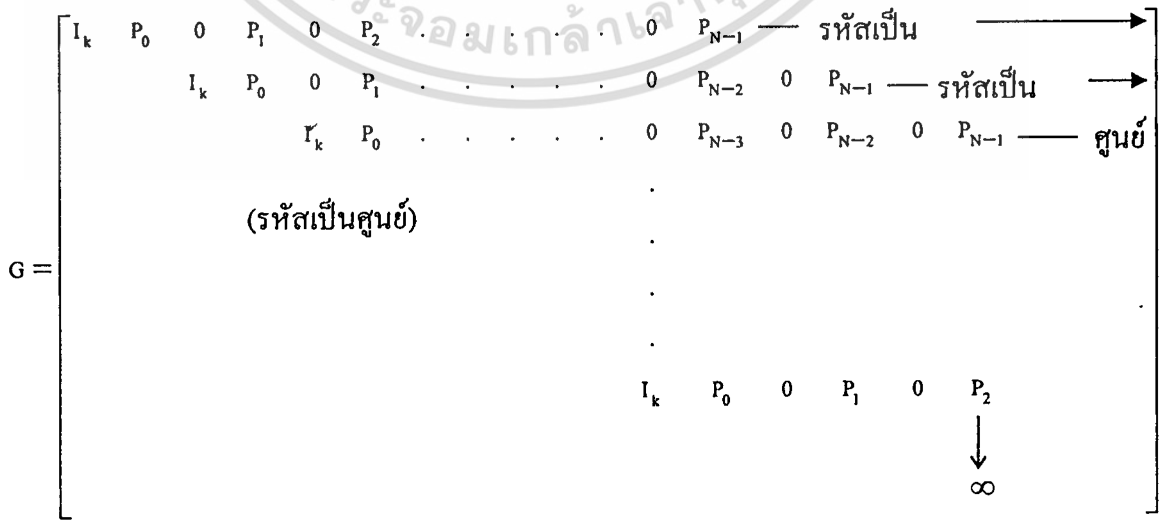
คอนโวลูชันแนลโค้ดถูกใช้ในการแก้รหัสที่ผิดอย่างแรนดอมหรือเบอสท์ การเข้ารหัสของคอนโวลูชันแนลโค้ดนี้สร้างได้โดยใช้ชีฟท์รีจิสเตอร์และได้มีการศึกษาวิจัยพบว่าประสิทธิภาพของคอนโวลูชันแนลโค้ดจะดีกว่าบล็อคโค้ด

ในบทนี้จะได้แสดงโครงสร้างอย่างง่ายของคอนโวลูชันแนลโค้ด สัญญลักษณ์ที่ใช้อาจจะยุ่งยากแต่ก็ไม่สามารถที่จะหลีกเลี่ยงได้

### 2.3.1 คุณสมบัติทั่ว ๆ ไปของคอนโวลูชันแนลโค้ด

การทำงานของตัวเข้ารหัสของคอนโวลูชันแนลโค้ดดังแสดงในรูปที่ 2.35 ซึ่งอธิบายได้ดังนี้ คือ สัญญาณเข้าที่ตัวรหัสจะเป็นลำดับของข่าวสาร m ที่จุดสัญญาณออกจะได้ลำดับของรหัสสัญญาณ c ณ. หน่วงเวลาใด ๆ บล็อคของข่าวสาร k บิทจะถูกป้อนเข้าตัวเข้ารหัสจะได้บล็อคของรหัส n บิทที่จุดสัญญาณออกของตัวเข้ารหัสโดยที่ k<n รหัส n บิทจะขึ้นกับบล็อคของข่าวสาร k บิท ณ.หน่วยเวลาเดียวกันและจะขึ้นกับ (n-1) บล็อคของข่าวสารที่ผ่านมาด้วยรหัสที่ได้จากตัวเข้ารหัสดังกล่าวเรียกว่า (n,k) คอนโวลูชันแนลโค้ดที่มีคอนสแตรนเล็งธ์ (Constraint length) เป็น n บล็อค (หรือ n N บิท) อัตราความเร็วของรหัสคือ R = k/n

รูปที่ 2.35 ลักษณะทั่วไปของตัวเข้ารหัสแบบคอนโวลูชันแนล




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือพูดได้ว่า

$$\{g_\infty(1), g_\infty(2), \ldots, g_\infty(k), Dg_\infty(1), Dg_\infty(2), \ldots Dg_\infty(k),$$
$$D^2 g_\infty(1), D^2 g_\infty(2), \ldots, D^2 g_\infty(k), \ldots\}$$

เป็นแถว (บรรทัด) ต่าง ๆ

ให้ m เป็น semi-infinite ของลำดับรหัสข่าวสารที่ถูกเข้ารหัส ลำดับข่าวสารนี้จะถูกแบ่งเป็นบล็อค ดังนี้

$$m = (m_0(1)m_0(2)\ldots m_0(k)m_1(1)m_1(2)\ldots m_1(k)m_2(1)m_2(2)\ldots m_2(k)\ldots) \quad (2.9)$$

บล็อคที่ $\lambda$รหัสข่าวสาร คือ $m_\lambda$จะประกอบด้วยรหัสข่าวสาร k บิท

$$m_\lambda = m_\lambda(1)m_\lambda(2)\ldots\ldots m_\lambda(k)$$

ลำดับของรหัสสัญญาณที่สอดคล้องกับ m คือ

$$C = mG^\infty$$
$$= m(1)g_\infty(1) + m(2)g_\infty(2) + \ldots + m(k)g_\infty(k) +$$
$$m(1)Dg_\infty(1) + m(2)Dg_\infty(2) + \ldots + m(k)Dg_\infty(k) +$$
$$\cdot$$
$$m_\lambda(1)D^\lambda g_\infty(1) + m_\lambda(2)D^\lambda g_\infty(2) + \ldots + m_\lambda(k)D^\lambda g_\infty(k) +$$
$$\cdot$$
$$= \sum_{\lambda=0}^{\infty} m_c(1)D^\lambda g_\infty(1) + \sum_{\lambda=0}^{\infty} m_c(2)D^\lambda g_\infty(2) + \ldots + \sum_{\lambda=0}^{\infty} m_c(k)D^\lambda g_\infty(k) \quad (2.10)$$

พิจารณาเซ็ทของเวคเตอร์ จำนวน k(n-k) เวคเตอร์ของรหัสฐานสอง N หลัก

$$g(i,j) = [g_0(i,j), g_1(i,j), g_2(i,j), \ldots, g_{N-1}(i,j)] \quad (2.11)$$

สำหรับ i = 1,2,….,k และ j = 1,2,….n-k จาก (n-k) เวคเตอร์เขียนเป็นแมททริกซ์ของ k แถวได้เป็น

$$Q_0 = \begin{bmatrix} g_\infty(1) \\ g_\infty(2) \\ \cdot \\ \cdot \\ g_\infty(k) \end{bmatrix} = [I_k \quad P_0 \quad 0 \quad P_1 \quad 0 \quad P_2 \quad 0 \quad \ldots \quad 0 \quad P_{N-1} \quad \text{ศูนย์} \quad \infty] \quad (2.12)$$

เมื่อ (1) $I_k$ เป็นไอเด็นติตี้แมททริกซ์ (identity matrix) ขนาด kxk

(2) 0 เป็นซีโร่แมททริกซ์ (zero matrix) ขนาด kxk

(3) $P_c$ เป็นแมททริกซ์ขนาด k x(n-k) โดยที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P_e = \begin{bmatrix} g_e(1,1) & g_e(1,2) & g_e(1,3) & \cdots & g_e(1,n-k) \\ g_e(2,1) & g_e(2,2) & g_e(2,3) & \cdots & g_e(2,n-k) \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & & & & \\ g_e(k,1) & g_e(k,2) & g_e(k,3) & \cdots & g_e(k,n-k) \end{bmatrix} \quad (2.13)$$

สำหรับ $0 \leq e \leq N-1$

ดังนั้นแต่ละแถวของ $g_\infty(i)$ เป็น semi-infinite vector โดยจะมีส่วนที่ไม่เป็นศูนย์อยู่ใน nN บิทแรก สำหรับค่าบวกจำนวนเต็ม $\lambda$ ใด ๆ สามารถกำหนดตัวที่ใช้ในการเลื่อนคือ $D^\lambda$ ดังนี้ เมื่อคูณ $D^\lambda$ เข้ากับลำดับของสัญญาณ $g_\infty(i)$ โดยจะทำให้ทุกรหัสใน $g_\infty(i)$ เลื่อนไปทางขวาเป็นจำนวน $\lambda n$ ตำแหน่ง รหัสบิทที่อยู่ก่อนหน้า $g_\infty(i)$ จะกลายเป็นศูนย์หมด

$\lambda n \quad g_\infty(i)$

$$D^\lambda g_\infty(i) = (0\ 0 \ldots .0 \qquad\qquad ) \quad (2.14)$$

รหัส (n,k) คอนโวลูชันแนลโค้ดเป็นรหัสเชิงเส้นที่สร้างได้จากแมททริกซ์ข้างล่างนี้ ลำดับของรหัสสัญญาณ c ประกอบด้วย

$$c = (c_0(1)\, c_0(2) \ldots c_0(n)\, c_1(1)\, c_1(2) \ldots c_1(n)\, c_2(1) c_2(2) \ldots \ldots c_2(n) \ldots) \quad (2.15)$$

เมื่อโค้ดในบล็อคที่ $\lambda$ คือ $c_\lambda$ ประกอบด้วยรหัส n หลัก

$$c_\lambda = c_\lambda(1) c_\lambda(2) \ldots c_\lambda(n)$$

จากสมการ (2.14) และ (2.15) รหัส n หลักในบล็อคที่ $\lambda$ คือ

$$c_\lambda(i) = m_\lambda(i) \text{ สำหรับ } i = 1,2,\ldots,$$

และ

$$c_\lambda(k+j) = \sum_{i=1}^{k} m_\lambda(i) g_0(i,j) + \sum_{i=1}^{k} m_{\lambda-1}(i) g_1(i,j) + \ldots + \sum_{i=1}^{k} m_{\lambda-N+1}(i) g_{N-1}(i,j) \quad (2.16)$$

สำหรับ j – 1,2,...., n-k โปรดสังเกตว่า $m_t(i)$ จะถือว่าเป็นศูนย์เมื่อ $t < 0$ จากสมการที่ (2.15) จะพบว่า k หลักแรกของรหัสในบล็อคที่ $\lambda$ จะเป็นรหัสของข่าวสาร k บิท ที่ถูกป้อนเข้าไปในตัวเข้ารหัส ณ หน่วยเวลาที่ $\lambda$ ส่วนรหัส (n-k) ของบล็อคที่ $\lambda$ คือรหัสของพาริตี้เช็คในสมการที่ (2.16) แสดงให้เห็นว่ารหัสพารีตี้เช็ค n-k บิท ไม่เพียงจะได้จากบล็อคของข่าวสารปัจจุบันเท่านั้นแต่ยังได้แสดงถึง N-1 บล็อคของข่าวสารที่ผ่านมา ดังนั้นการเข้ารหัสของโค้ดแบบคอนโวลูชันแนลนั้นแต่ละบิทใน k บิทของบล็อคของข่าวสารมีผลต่อโค้ดเวคเตอร์ที่จะตามมาอีก N บล็อก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังได้แสดงให้เห็นแล้วว่า (n,k) คอนโวลูชั่นโค้ดเกิดจากเซ็ทของ k(n-k) เวคเตอร์ในสมการที่ (2.10) เวคเตอร์ k(n-k) เหล่านี้เรียกว่าซับเจนเนอเรเตอร์ (sub-geneator) ของ (n,k) คอนโวลูชันแนลโค้ด ให้ g(i) เป็นเวคเตอร์ที่ประกอบด้วย nN บิทของ $g_{\infty}(i)$ ได้จากสมการ (2.11) เวคเต่อร์ k ชุดคือ g(1),g(2),…..,g(k) เรียกว่าเจนเนอเรเตอร์ (หรือลำดับของเจนเนอเรเตอร์) ของรหัสเจนเนอเรเตอร์ทั้ง k ตัวนี้ได้จากการผสมผสานกันระหว่าง k(n-k) ซับเจนเนอเรเตอร์ที่สอดคล้องกับสมการที่ (2.11) และ (2.12) ถ้าเจนเนอเรเตอร์ k ชุดของรหัส (n-k) ถูกกำหนดให้ ดังนั้น k(n-k) ซับเจนเนอเรเตอร์ของรหัสสามารถหาได้จากการแยกเจนเนอเรเตอร์แมททริกซ์ $G_{\infty}$ ในสมการที่ (2.14) เรียกว่าเจนเนอเรเตอร์แมททริกซ์ของรหัส ค่า N คือ ค่าคอนสแตรนเล็งธ์ เนื่องจากที่หน่วยเวลาใด ๆ รหัส k บิทแรกคือรหัสของข่าวสารที่ไม่มีการเปลี่ยนแปลงรหัสจะมีลักษณะเป็นรูปแบบระบบ สำหรับรหัสที่ไม่เป็นรหัสแบบระบบ ได้จาก k แถวแรก (ไม่เป็นศูนย์ใน nN บิทแรก) ของ $G_{\infty}$ ไม่อยู่ในรูปแบบของสมการที่ (2.11)

เช่น พิจารณา (2,1) คอนโวลูชันแนลโค้ด เมื่อคอนสแตรนเล็งธ์ N = 4 โดยมีซับเจนเนอเรเตอร์เป็น

$g(1,1) = (1\ 1\ 0\ 1)$

วิธีทำ $K = 1, n = 2$ และ $N = 4$

$g(1,1) = (g_0(1,1), g_1(1,1), g_2(1,1), g_3(1,1))$

ทำให้ได้ $g_0(1,1) = 1$ , $g_1(1,1) = 1$

$g_2(1,1) = 0$ , $g_3(1,1) = 1$

แต่เนื่องจาก $Q_0 = g(1) = [I_k\ P_0\ 0\ P_1\ 0\ P_2\ 0\ P_3]$

$I_k = 1$

$P_0 = g_0(1,1) = 1$

$P_1 = g_1(1,1) = 1$

$P_2 = g_2(1,1) = 0$

$P_3 = g_3(1,1) = 1$

$g(1) = [\ 1\ 1\ 0\ 1\ 0\ 0\ 0\ 1\ ]$

จะได้เจนเนอเรเตอร์แมททริกซ์เป็น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$G=\begin{bmatrix} g(1) \\ Dg(1) \\ D^{2}g(1) \\ D^{3}g(1) \\ D^{4}g(1) \\ \downarrow \\ \ \end{bmatrix}=\begin{bmatrix} 11 & 01 & 00 & 01 & & & & \\ & 11 & 01 & 00 & 01 & & & \\ & & 11 & 01 & 00 & 01 & & \\ & & & 11 & 01 & 00 & 01 & \\ & & & & 11 & 01 & 00 & 01 \\ & & & & & \downarrow & & \\ & & & & & \infty & & \end{bmatrix}$$

ให้ m = (1 0 0 1 1 ........) เป็นลำดับของรหัสข่าวสารที่ต้องการเข้ารหัส รหัสที่ออกจากจุดสัญญาณออกของวงจรเข้ารหัสคือ

$C = m\ G_{\infty} = (11\ 01\ 00\ 10\ 10.....)$

เช่น จากรหัส (3,2) โดยมี N = 3 และกำหนดให้ซับเจนเนอเรเตอร์เป็น

$g(1,1) = 101$

$g(2,1) = 110$

มีลำดับของรหัสสัญญาณเข้าของตัวเข้ารหัสเป็น m = (11 00 10....) จงหาลำดับของรหัสสัญญาณที่ออกจากตัวเข้ารหัส

วิธีทำ $k = 2, n = 3, N = 3$ และ $n\text{-}k = 3\text{-}2 = 1$

$g(1,1) = (g_0(1,1), g_1(1,1), g_2(1,1)) = (1\ 0\ 1)$

$g(2,1) = (g_0(2,1), g_1(2,1), g_2(2,1)) = (1\ 1\ 0)$

แต่เนื่องจาก $Q_0 = \begin{bmatrix} g(1) \\ g(2) \end{bmatrix} = [I_k \quad P_0 \quad 0 \quad P_1 \quad 0 \quad P_2]$

$$I_k = I_2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad 0 = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$$

$$P_e = \begin{bmatrix} g_e(11) & g_e(1,2)....... & g_e(1,n-k) \\ g_e(2,1) & & \\ . & & \\ g_e(k,1) & ..................... & g_e(k,n-k) \end{bmatrix}$$

$$P_0 = \begin{bmatrix} g_0(1,1) \\ g_0(2,1) \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$$P_1 = \begin{bmatrix} g_1(1,1) \\ g_1(2,1) \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P_2 = \begin{bmatrix} g_2(1,1) \\ g_2(2,1) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$$G_\infty = \begin{bmatrix} 101 & 000 & 001 & & \\ 011 & 001 & 000 & & \\ & 101 & 000 & 001 & \\ & 011 & 001 & 000 & \\ & & 101 & 000 & 001 \\ & & 011 & 001 & 000 \\ & & \downarrow & & \\ & & \infty & & \end{bmatrix}$$

จะได้ลำดับของสัญญาณออกจากตัวเข้ารหัสคือ

$C = m\ G_\infty = (110\,001\,100....)$

เนื่องจาก (n,k) คอนโวลูชันแนลโค้ดเป็นรหัสเชิงเส้น ทำให้สามารถหาพาริตี้เช็คแมททริกซ์ ได้เป็น

$$H_\infty = \begin{bmatrix} P_0^T & I_k & & & & & & & & & \\ P_1^T & 0 & P_0^T & I_k & & & & & & & \\ P_2^T & 0 & P_1^T & 0 & P_0^T & I_k & & & & & \\ \cdot & & & & & & & & & & \\ \cdot & & & & & & & & & & \\ P_{N-1}^T & 0 & P_{N-2}^T & 0 & P_{N-3}^T & 0 & \cdots\cdots & P_0 & I_k & & \\ & & P_{N-1}^T & 0 & P_{N-2}^T & 0 & \cdots\cdots & P_1^T & 0 & P_0^T & I_k \\ & & & & P_{N-1}^T & 0 & \cdots\cdots & P_2^T & 0 & P_1^T & 0 \\ & & & & & & & & & & \\ \infty & & \infty & & \infty & & & & & & \end{bmatrix} \qquad (2.17)$$

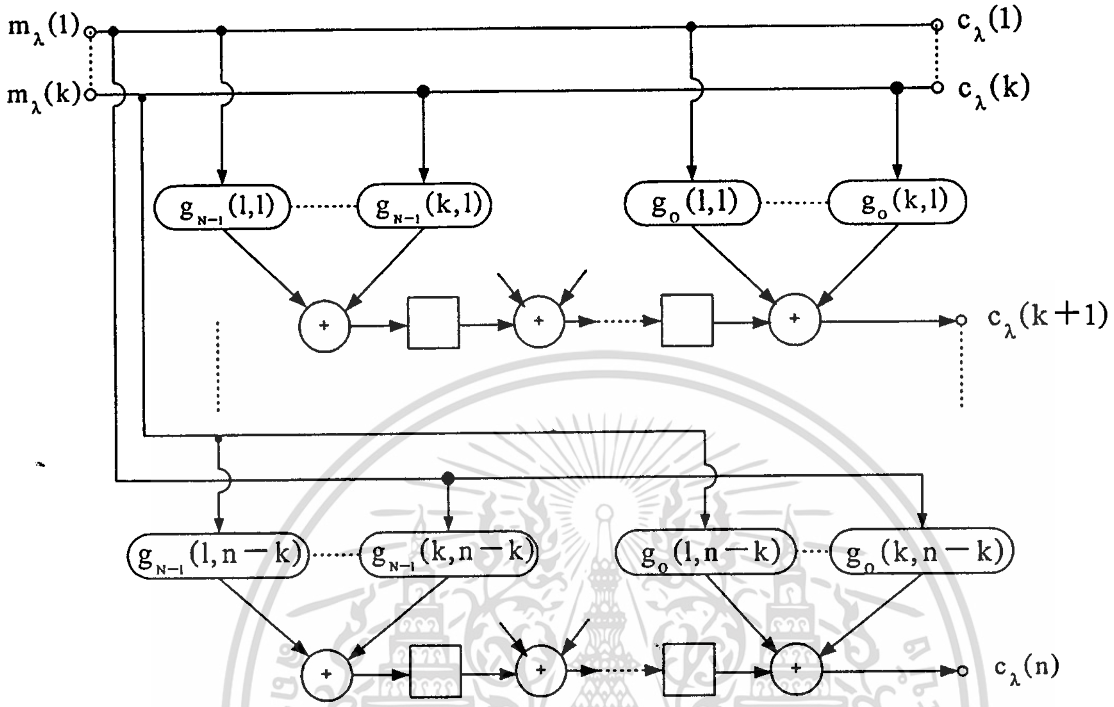
เมื่อ $P_\lambda^T$ เป็นทรานสโพส ของ $P_\lambda$ โดยจะเห็นได้ชัดว่า $G_\infty \ H_\infty^T = 0$ ดังนั้น C จะเป็นลำดับของสัญญาณ ถ้าและเพียงแต่ถ้า $CH^T = 0$

### 2.3.2 การเข้ารหัสของคอนโวลูชันแนลโค้ด

การเข้ารหัสของ (n,k) คอนโวลูชันแนลโค้ดสามารถทำได้ง่าย ๆ ด้วยชีฟท์รีจิสเตอร์ที่ต่อไว้สอดคล้องกับชุดของ k(n-k) ซับเจนเนอเรเตอร์ที่กำหนดให้ จากสมการที่ (2.16) และ (2.17) จะให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรเข้ารหัสที่ใช้ชีฟท์รีจิสเตอร์ (n-k)(N-1) ตัวแสดงในรูปที่ 2.36 ซึ่งออกแบบโดย Wozencraft และ Reifter [2]




$g_s(i,j)$ : จะมีการต่อเมื่อมีค่า = 1 และจะไม่มีการต่อเมื่อมีค่า = 0

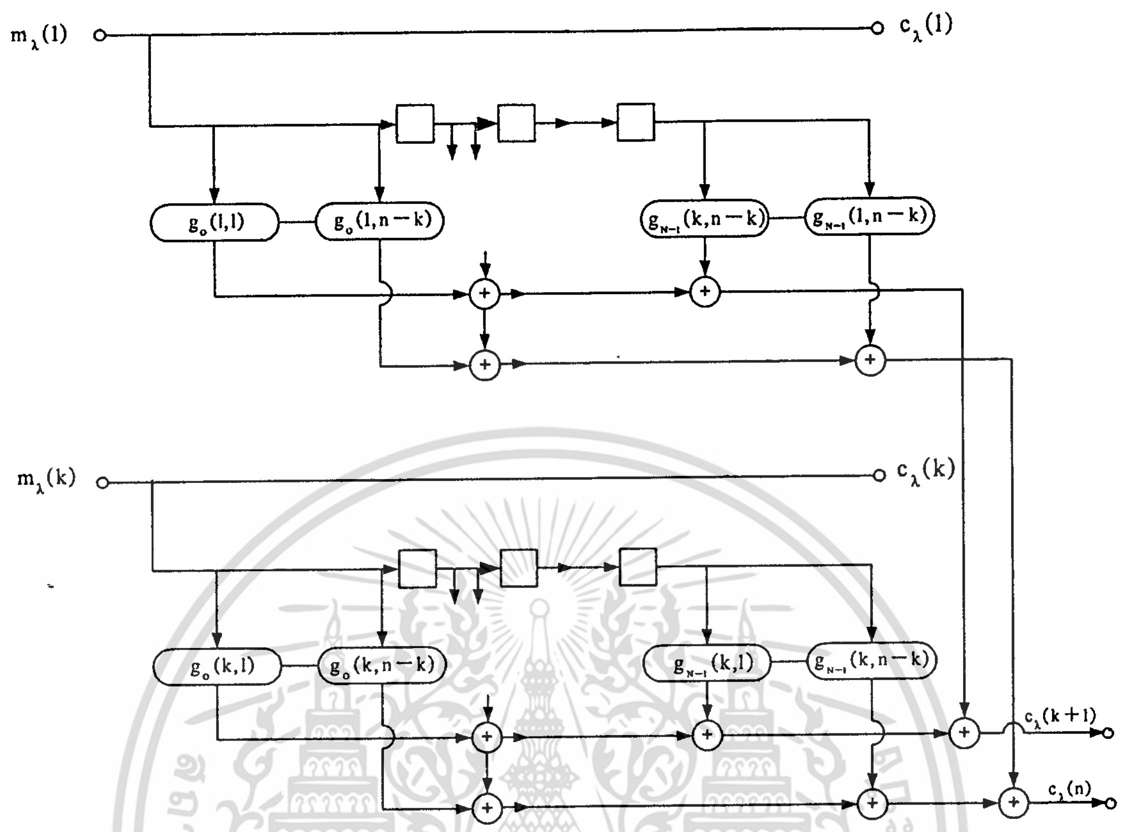
**รูปที่ 2.36** วงจรเข้ารหัสที่ใช้ชีฟท์รีจิสเตอร์ (n-k)(N-1) ตัว (วงจรเข้ารหัสแบบที่ 1)

จากสมการที่ (2.16) สามารถเขียนใหม่ได้เป็น

$$C_\lambda(k+j) = \sum_{t=0}^{N-1} m_{\lambda-t}(1)g_t(1,j) + \sum_{t=0}^{N-1} m_{\lambda-t}(2)g_t(2,j) + \dots\dots + \sum_{t=0}^{N-1} m_{\lambda-t}(k)g_t(k,j) \quad (2.18)$$

จากสมการที่ (2.16) และ (2.17) จะได้วงจรเข้ารหัสแบบที่ 2 ที่ใช้ชีฟท์รีจิสเตอร์ k(N-1) ตัว ดังแสดงในรูปที่ 2.37 ซึ่งได้รับการปรับปรุงจาก Massey [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.37** วงจรเข้ารหัสที่ใช้ชีฟท์รีจิสเตอร์ k(N-1) ตัว (วงจรเข้ารหัสแบบที่ 2)

เนื่องจากวงจรเข้ารหัสทั้งสองแบบจะรับรหัสข่าวสารในลักษณะขนานดังนั้นที่จุดก่อนจุดสัญญาณเข้าของวงจรเข้ารหัสทั้งสองนี้จะต้องมี Commutator ช่วยแปลงรหัสสัญญาณอนุกรมให้เป็นแบบขนานโดยจะช่วยแปลงลำดับของรหัสข่าวสาร m ให้รหัส k หลักที่จะป้อนเข้าวงจรได้พร้อม ๆ กัน

ลำดับสัญญาณย่อยจะกลายเป็น

$$m^{(1)} = m_0(1)m_1(1)m_2(1) \ldots\ldots\ldots\ldots m_e(1)$$

$$m^{(2)} = m_0(2)m_1(2)m_2(2) \ldots\ldots\ldots\ldots m_e(2)$$

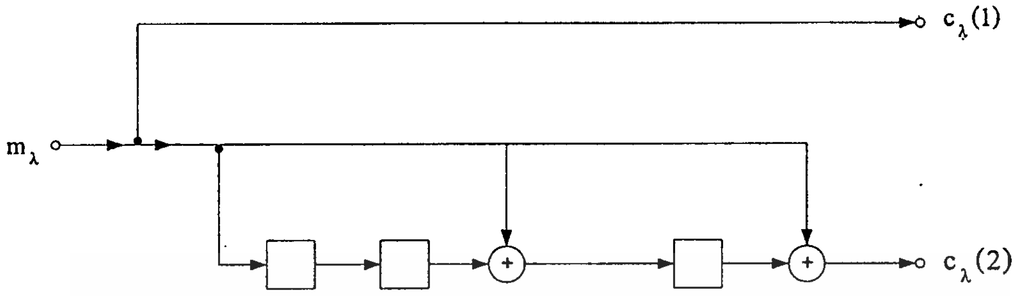
$$\vdots \qquad (2.19)$$

$$m^{(k)} = m_0(k)m_1(k)m_2(k) \ldots\ldots\ldots\ldots m_e(k)$$

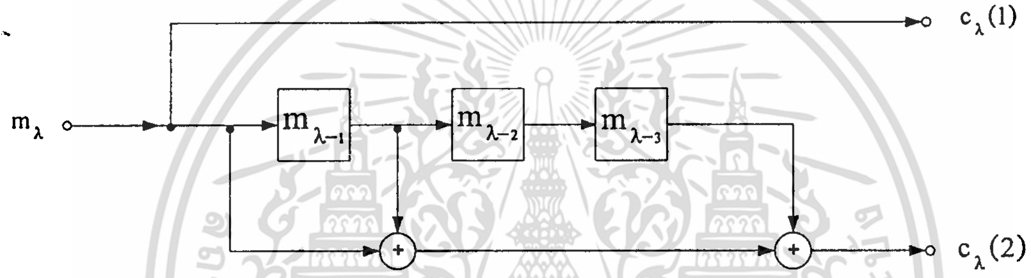
ในระหว่างหน่วยเวลาที่ $\lambda$ บล็อคของข่าวสารคือ $(m_\lambda(1)\ m_\lambda(2),\ldots\ldots\ldots, m_\lambda(k))$ จะถูกป้อนเข้าวงจรเข้ารหัส รีจิสเตอร์ต่าง ๆ ของวงจรเลื่อนข้อมูลไปหนึ่งครั้ง และบล็อคของโค้ดลำดับที่ $\lambda$คือ $(c_\lambda(1), c_\lambda(2),\ldots, c_\lambda(n))$ ถูกสร้างจากจุดสัญญาณออก n จุดของวงจร ถ้ารหัสถูกส่งออกแบบอนุกรม จะต้องใช้ commutator ช่วงแปลงรหัสแบบขนานให้เป็นแบบอนุกรมเสียก่อน

เอกสารนี้มีไว้เพื่อการศึกษาเท่านั้น ห้ามนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับรหัสแบบอัตราส่งสูง (k > n-k) วงจรแบบที่ 1 จะประหยัดกว่าวงจรแบบที่ 2 ส่วนถ้ารหัสเป็นแบบอัตราส่งต่ำแล้ววงจรแบบที่ 2 จะประหยัดกว่าแบบที่ 1 วงจรทั้งสองแบบที่ใช้สำหรับรหัส (2,1) ในตัวอย่างที่ 2.5 ดังแสดงในรูปที่ 2.38 และ 2.39 ตามลำดับ




รูปที่ 2.38 วงจรเข้ารหัสแบบที่ 1 สำหรับ (2,1) คอนโวลูชันแนลโค้ดของตัวอย่างที่ 2.5




รูปที่ 2.39 วงจรเข้ารหัสแบบที่ 2 สำหรับ (2,1) คอนโวลูชันแนลโค้ดของตัวอย่างที่ 2.5

**2.3.3 การคำนวณหาซินโดรม**

ให้ C เป็นลำดับของรหัสที่ถูกส่งออกมาและ e เป็นลำดับของสัญญาณรบกวนในช่องส่ง ดังนั้นลำดับของรหัสที่รับได้จะเป็น

$$r = c+e$$

โดยที่

$$r = (r_0(1)r_0(2).....r_0(n)r_1(1)r_1(2).....r_1(n)r_2(1)r_2(2).....$$
$$.....r_2(n)...... r_\lambda(1)\ r_\lambda(2)......\ r_\lambda(n).....)$$

เมื่อ $r_\lambda(i) = c_\lambda(i)+e_\lambda(i)$

สำหรับ i = 1,2,.....,n ซินโดรมของลำดับของรหัสที่รับได้คือ

$$s = rH_\infty^T \quad (2.20)$$

นั้นคือ s ก็เป็นลำดับสัญญาณแบบ semi-infinite ด้วย

$$s = (s_0(1)s_0(2)......s_0(n\text{-}k)s_1(1)s_1(2)......s_1(n\text{-}k)$$
$$......s_\lambda(1)\ s_\lambda(2).......\ s_\lambda(n\text{-}k).......) \quad (2.21)$$

เมื่อบล็อคที่ λ ประกอบด้วยซินโดรม n-k หลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$s_\lambda(1)\ s_\lambda(2)\ldots\ldots\ldots\ s_\lambda(n\text{-}k)$$

จากสมการที่ (2.20) ซีนโดรม n-k บิทของบล็อคที่ $\lambda$ ได้จาก

$$s_\lambda(j) = r_\lambda(k+j) + \sum_{i=1}^{k} r_\lambda(i) g_0(i,j) + \sum_{i=1}^{k} r_{\lambda-1}(i) g_1(i,j) + \ldots\ldots + \sum_{i=1}^{k} r_{\lambda-N+1}(i) g_{N-1}(i,j) \qquad (2.22)$$

สำหรับ j = 1,2,.....,n-k แต่ r = c+e และ $CH_\infty^T = 0$ จะได้

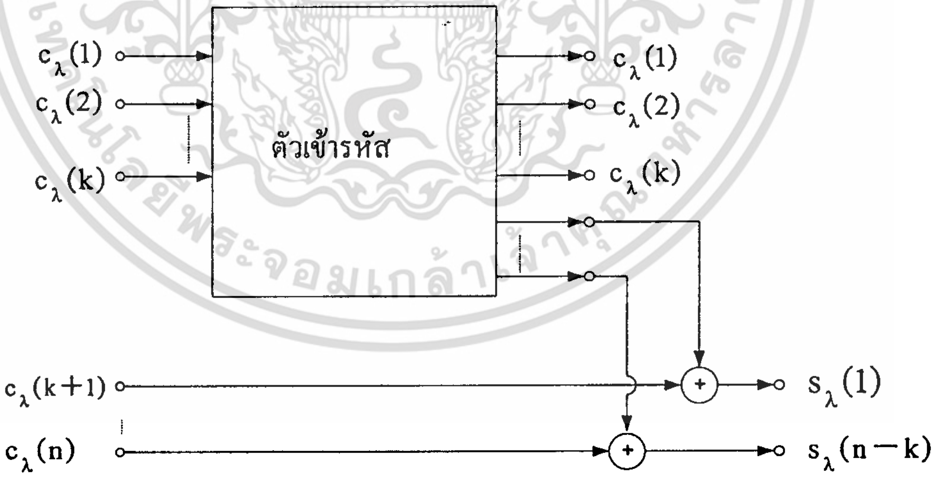
$$\begin{aligned} S &= rH_\infty^T \\ &= (c+e)H_\infty^T \\ &= eH_\infty^T \end{aligned} \qquad (2.23)$$

จากสมการที่ (2.23) บิทของซีนโดรมเขียนให้อยู่ในเทอมของรหัสบิทที่ผิด คือ

$$s(i) = e_\lambda(k+j) + \sum_{i=1}^{k} e_\lambda(i) g_0(i,j) + \sum_{i=1}^{k} e_{\lambda-1}(i) g_1(i,j) + \ldots\ldots + \sum_{i=1}^{k} e_{\lambda-N+1}(i) g_{N-1}(i,j) \qquad (2.24)$$

สำหรับ j = 1,2,.......,n-k ($e_t = 0$ สำหรับ $t < 0$) ซีนโดรมของลำดับรหัสสัญญาณที่รับได้ประกอบด้วยข้อมูลของลำดับของรหัสที่ผิดและสามารถนำไปใช้ในการถอดรหัส

ซีนโดรมสามารถคำนวณได้ในตัวถอดรหัสด้วยการเข้ารหัสของรหัสข่าวสารที่รับได้และบวกผลลัพธ์ของรหัสพารีตี้เช็คเข้ากับรหัสพารีตี้เช็คที่รับได้ดังแสดงในรูปที่ 2.40




**รูปที่ 2.40** ตัวคำนวณหาซีนโดรมสำหรับคอนโวลูชันแนลโค้ด

### 2.3.4 การถอดรหัสของคอนโวลูชันแนลโค้ดและการแพร่กระจายของรหัสที่ผิด

เนื่องจากการเข้ารหัสของคอนโวลูชันแนลโค้ดจะทำทีละบล็อค (ซึ่งมีรหัสข่าวสารอยู่ k บิท และความยาวของโค้ดคือ n บิท) ลำดับของรหัสที่รับได้ก็จะถูกถอดรหัสทีละบล็อคของรหัส n บิท ดังได้แสดงไว้ในหัวข้อที่ 2.3.3 แล้วว่าซีนโดรมของลำดับรหัสที่รับได้ประกอบด้วยข้อมูลเกี่ยวกับลำดับของรหัสที่ผิด ซึ่งเกิดจากสัญญาณรบกวนในช่องส่ง จึงสามารถนำมาประมาณรหัสที่ผิดได้

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพื่อที่จะแสดงถึงการถอดรหัสของบล็อคหนึ่งที่รับเข้ามามีรหัสอยู่ n หลักสมการที่ (2.24) เขียนใหม่ได้เป็น

$$s_0(j)=e_0(k+j)+\sum_{i=1}^{k}e_0(i)g_0(i,j)$$

$$s_1(j)=e_1(k+j)+\sum_{i=1}^{k}e_1(i)g_0(i,j)+\sum_{i=1}^{k}e_0(i)g_1(i,j)$$

.

.

$$s_{N-1}(j)=e_{N-1}(k+j)+\sum_{i=1}^{k}e_{N-1}(i)g_0(i,j)+\ldots\ldots$$

$$\ldots\ldots\ldots+\sum_{i=1}^{k}e_0(i)g_{N-1}(i,j)$$

$$s_N(j)=e_N(k+j)+\sum_{i=1}^{k}e_N(i)g_0(i,j)+\ldots+\sum_{i=1}^{k}e_1(i)g_{N-1}(k,j)$$

$$s_{N+1}(j)=e_{N+1}(k+j)+\sum_{i=1}^{k}e_{N+1}(i)g_0(i,j)+\ldots$$

$$\ldots+\sum_{i=1}^{k}e_2(i)g_{N-1}(i,j)$$

(2.25)

เมื่อ j = 1,2,.......,n-k ในการตรวจสอบเซ็ทของสมการเหล่านี้ จะสังเกตุพบว่ารหัสที่ผิดคือ $e_0$ (1), $e_0$ (2),......,$e_0$(n) ในบล็อคที่ 0 ของลำดับของรหัสที่ผิดจะถูกตรวจสอบเพียงใน N(n-k) สมการแรก ดังนั้นในการเข้ารหัสของบล็อคที่ 0 ของลำดับของรหัสที่รับได้จะมีเพียง N(n-k) บิทของซีนโดรม จากบล็อคที่ 0 ถึงบล็อคที่ N-1 ที่จะถูกนำมาใช้ หลังจากที่รหัสผิดถูกตรวจพบในบล็อคที่ 0 แล้ว ผลกระทบที่มีต่อซีนโดรมจะหมดไป การกำจัดผลกระทบจากรหัสที่ผิดให้หมดไปจากซีนโดรมนี้เรียกว่าซีนโดรมรีเซทติ้ง (syndrome resetting) จากการถอดรหัสในบล็อคที่ 0 ตัวถอดรหัสจะเริ่มถอดรหัสบล็อคที่ 1 ของลำดับรหัสที่รับได้ เนื่องจากรหัสผิดในบล็อคที่ 1 จะถูกตรวจสอบด้วยซีนโดรม N(n-k) บิทจากบล็อคที่ 1 ถึงบล็อคที่ N ของลำดับรหัสที่รับได้ ดังนั้นบล็อคที่ 1 ของลำดับรหัสที่รับได้จะถูกถอดรหัสด้วยการตรวจสอบซีนโดรมที่ถูกเปลี่ยนแปลงจากบล็อคที่ 1 ถึง บล็อคที่ N ถ้าการถอดรหัสที่บล็อคที่ 0 ที่รับได้ถูกต้อง ซีนโดรมรีเซทติ้งจะมีผลในการเปลี่ยนแปลงลำดับของรหัสซีนโดรมดังนี้

$$s'_0(j)=0$$

$$s'_1(j)=e_1(k+j)+\sum_{i=1}^{k}e_1(i)g_0(i,j)$$

$$s'_2(j)=e_2(k+j)+\sum_{i=1}^{k}e_2(i)g_0(i,j)+\sum_{i=1}^{k}e_1(i)g_1(i,j)$$

.

$$s'_{N-1}(j)=e_{N-1}(k+j)+\sum_{i=1}^{k}e_{N-1}(i)g_0(i,j)+\ldots\ldots\ldots.$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\ldots\ldots + \sum_{i=1}^{k} e_1(i) g_{N-2}(i,j)$$

$$s'_N(j) = e_N(k+j) + \sum_{i=1}^{k} e_N(i) g_0(i,j) + \ldots\ldots\ldots.$$

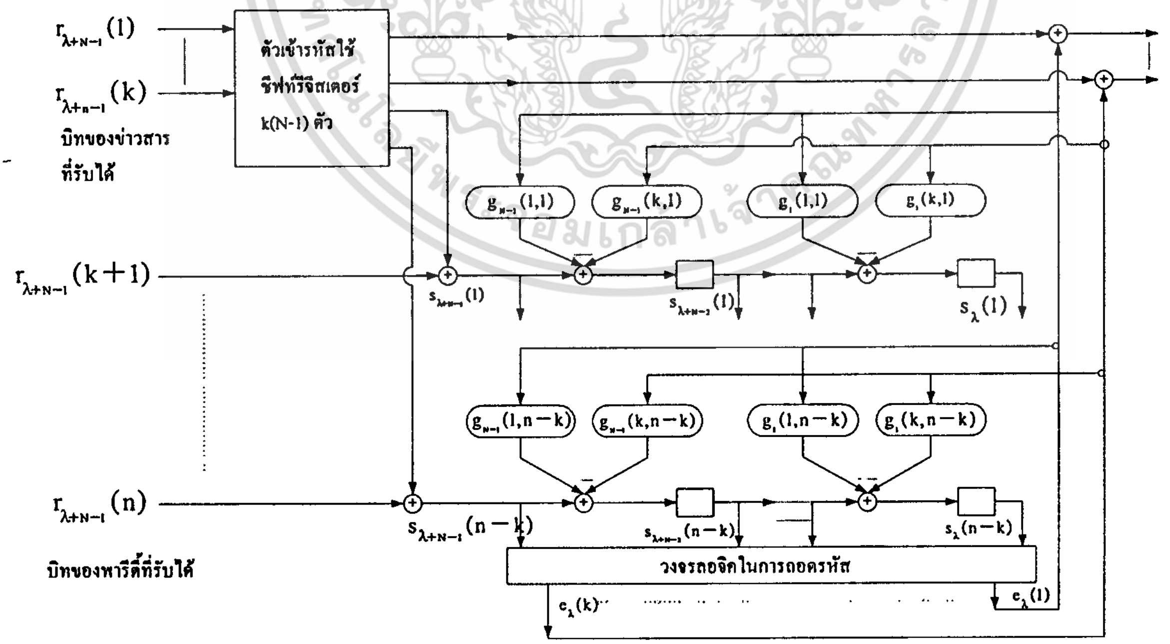
$$\ldots\ldots + \sum_{i=1}^{k} e_1(i) g_{N-1}(i,j)$$

$$s'_{N+1}(j) = e_{N+1}(k+j) + \sum_{i=1}^{k} e_{N+1}(i) g_0(i,j) + \ldots\ldots\ldots.$$

$$\ldots\ldots + \sum_{i=1}^{k} e_2(i) g_{N-1}(i,j) \tag{2.26}$$

โดยที่จะมีเพียงบิทต่าง ๆ ของซีนโดรมในบล็อคที่ 0 ถึงบล็อคที่ N-1 ที่เปลี่ยนแปลง จะสังเกตพบว่าบิทต่าง ๆ ของซีนโดรมที่เปลี่ยนจากบล็อคที่ 1 ถึงบล็อคที่ N จะถูกนำมาใช้สำหรับการถอดรหัสในบล็อคที่ 1 ที่รับได้จะมีรูปแบบเหมือนกันกับบิทต่าง ๆ ของซีนโดรมที่ใช้ในการถอดรหัสของ บล็อคที่ 0 ที่รับได้หลังจากบล็อคที่ 1 ของลำดับรหัสสัญญาณที่รับได้ผ่านการถอดรหัสเรียบร้อยแล้ว ซีนโดรมจะถูกปรับเปลี่ยนใหม่อีกครั้งหนึ่ง การถอดรหัสก็จะทำงานในลักษณะนี้ไปเรื่อย ๆ ในการถอดรหัสของบล็อคที่ $\lambda$ ของลำดับรหัสที่รับได้ ซีนโดรม(ที่ผ่านการกำจัดผลกระทบอันเกิดจากบิทที่ผิดก่อน ๆ) จากบล็อคที่ $\lambda$ ถึงบล็อคที่ $\lambda$-N+1 จะถูกนำมาใช้

ตัวถอดรหัสทั่ว ๆ ไปของ (n,k) คอนโวลูชันแนลโค้ด ดังแสดงในรูปที่ 2.41




**รูปที่ 2.41** วงจรถอดรหัสทั่ว ๆ ไปสำหรับคอนโวลูชันแนลโค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของวงจรถอดรหัสมีดังนี้

ขั้นที่ 1 สำหรับการถอดรหัส บล็อคที่ 0 ของลำดับรหัสที่รับเข้ามา รหัสของซินโดรม N(n-k) บิทแรกจากบล็อคที่ 0 ถึงบล็อคที่ N-1 ของลำดับรหัส(ซินโดรม s) คำนวณได้จากการเลื่อน N บล็อคแรกของลำดับรหัสที่รับได้เข้าสู่ซินโดรมเจนเนอเรเตอร์ (รีจิสเตอร์ทุกตัวจะเป็นศูนย์ในสภาวะเริ่มต้น)

ขั้นที่ 2 รหัสของซินโดรมทั้ง N(n-k) บิท จะถูกอ่านเข้าไปในวงจรลอจิกและถูกทดสอบสำหรับหาบิทที่ผิดในบล็อคที่ 0 ของลำดับสัญญาณที่รับได้ โดยปกติจะกล่าวถึงเฉพาะบิทที่ผิดที่เกิดขึ้นในตำแหน่งต่าง ๆ ของรหัสข่าวสาร k บิท ดังนั้นจุดสัญญาณออกของวงจรลอจิกจะใช้ประมาณหาค่าบิทที่ผิดในตำแหน่งทั้ง k บิทของรหัสข่าวสารของบล็อคที่ 0

ขั้นที่ 3 รหัสข่าวสาร k หลัก คือ $r_0(1),r_0(2),\ldots,r_0(k)$ ถูกดึงออกจากตัวเข้ารหัสและถูกแก้ด้วยสัญญาณรหัสที่ออกจากวงจรลอจิก การแก้รหัสทำได้โดยการใช้เกทแบบ exclusive OR จำนวน k ตัว ในเวลาเดียวกันรหัสที่ผิดจะถูกป้อนกลับไปยังซินโดรมรีจิสเตอร์ เพื่อจะกำจัดผลกระทบของรหัสผิดที่มีต่อลำดับรหัสของซินโดรม (syndrome resetting)

ขั้นที่ 4 ทันทีที่รหัสข่าวสาร k บิทที่รับได้เคลื่อนออกจากตัวเข้ารหัส ซินโดรมรีจิสเตอร์ทั้ง (n-k) ตัวจะถูกเลื่อนไปทางขวาหนึ่งครั้ง บล็อคที่ N ของลำดับรหัสที่รับได้จะถูกเลื่อนเข้าไปในซินโดรมเจนเนอเรเตอร์ โดยจะสอดคล้องกับ (n-k) บิทของซินโดรมที่ถูกคำนวณและเก็บไว้ในตัวซ้ายสุดของ (n-k) ซินโดรมรีจิสเตอร์

ขั้นที่ 5 ในซินโดรมรีจิสเตอร์จะประกอบด้วย รหัสของซินโดรมที่เปลี่ยนแปลงไป N(n-k) บิทจากบล็อคที่ 1 ถึงบล็อคที่ N ตัวถอดรหัสจะเริ่มทำการถอดรหัสของบล็อคที่ 1 ของลำดับรหัสที่รับได้ด้วยวิธีการเดียวกันกับการถอดรหัสในบล็อคที่ 0 ตัวถอดรหัสจะทำงานซ้ำขั้นที่ 2,3 และ 4

ขั้นที่ 6 ตัวถอดรหัสจะทำงานด้วยวิธีการทีละขั้นและจะถอดรหัสทั้ง n บิทในเวลาเดียวกัน

การปรับเปลี่ยนค่าซินโดรมใหม่หลังจากการถอดรหัสแต่ละครั้งผลลัพธ์จะถูกป้อนกลับไปยังตัวถอดรหัส ตัวถอดรหัสแบบนี้บางครั้งเรียกว่า feedback decoder และการทำงานของซินโดรมรีเซทติ้งนี้บางครั้งก็เรียกว่า feedback decdoding

ข้อเสียเปรียบของการถอดรหัสแบบป้อนกลับ คือแนวโน้มของความผิดพลาดในการถอดรหัสอันเนื่องจากความผิดพลาดจากการถอดรหัสที่ผ่านมา ปรากฏการณ์นี้เรียกว่าการเกิดการแพร่ของบิทที่ผิด (error propagation) วิธีการที่จะจำกัดการแพร่ของบิทที่ผิดคือให้สมมติว่าตัวถอดรหัสกำลังถอดรหัสในบล็อคที่ $\lambda$ ของลำดับรหัสที่รับได้ จากสมการที่ (2.16) สังเกตพบว่ารหัสที่ผิดของช่องส่งที่ปรากฏก่อนหน่วยเวลาที่ ($\lambda$-N+1) ไม่มีผลกระทบต่อรหัสของซินโดรมหลังจากเวลาผ่านไป ($\lambda$+1) หน่วย ถ้าการถอดรหัสผิดไปก่อนเวลาหน่วยที่ ($\lambda$-N+1) และไม่เกิดการถอดรหัสผิดจากหน่วยเวลาที่ ($\lambda$-N+1) ถึง ($\lambda$-1) ดังนั้นการปรับค่าซินโดรมใหม่ จะให้ค่าซินโดรมที่ถูกต้องสำหรับการถอดรหัสในบล็อคที่ $\lambda$ นั้นคือถ้าการถอดรหัสผิดไปจะตามหลังด้วยการ

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ห้ามนำไปใช้เพื่อการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถอดรหัสที่ถูกต้องจำนวน N-1 บิท ตัวถอดรหัสจะกลับสู่สภาวะการทำงานที่ปกติ การแพร่ของบิทที่ผิดก็จะหยุดเพียงแค่นั้น

### 2.3.5 โครงสร้างแบบต้นไม้ และคุณสมบัติของระยะห่าง ของคอนโวลูชันแนลโค้ด

คอนโวลูชันแนลโค้ดมีโครงสร้างที่น่าสนใจซึ่งจัดลำดับของรหัสเป็นแบบ semi-infinite ที่สามารถเขียนให้อยู่ในรูปโครงสร้างแบบต้นไม้ได้ แต่ละต้นจะมี n ตา (node) แต่ละตาจะมี $2^k$ กิ่ง (branch) แต่ละกิ่งจะมีหนึ่งบล็อคของรหัสขนาด n บิทที่สอดคล้องกับบล็อคของข่าวสาร k บิท ดังนั้นลำดับของสัญญาณจะแสดงได้ด้วยเส้นทางของโครงสร้างต้นไม้นี้ไปจนถึงตำแหน่งที่อนันต์

เพื่อให้ง่ายต่อการเข้าใจจะได้พิจารณาถึง (n,1) คอนโวลูชันแนลโค้ดก่อน ดังที่ได้แสดงในหัวข้อที่ 2.3.1 (n,1) คอนโวลูชันแนลโค้ดที่ถูกกำหนดไว้ด้วย n-1 ซับเจนเนอเรเตอร์

$$g(1,j)=(g_0(1,j),g_1(1,j),g_2(1,j),\ldots g_{N-1}(1,j))$$

สำหรับ j = 1,2,….,n-1 ตัวเจนเนอเรเตอร์ของรหัสนี้คือ

$$\begin{aligned} g(1)=(&1g_0(1,1)g_0(1,2)\ldots g_0(1,n-1)0g_1(1,1)g_1(1,2) \\ &\ldots g_1(1,n-1)0g_2(1,1)g_2(1,2)\ldots g_2(1,n-1) \\ &\ldots 0g_{N-1}(1,1)g_{N-1}(1,2)\ldots g_{N-1}(1,n-1)) \end{aligned} \quad (2.27)$$

เพื่อความสะดวกเราจะใช้สัญญลักษณ์ดังนี้

$$\underline{g}_0 = 1g_0(1,1)g_0(1,2)\ldots g_0(1,n-1)$$

และ

$$\underline{g}_\lambda = 0g_\lambda(1,1)g_\lambda(1,2)\ldots g_\lambda(1,n-1)$$

สำหรับ $\lambda = 1,2\ldots, N-1$ ดังนั้นเจนเนอเรเตอร์ g(1) เขียนใหม่ได้เป็น

$$g(1) = \underline{g}_0 \quad \underline{g}_1 \quad \underline{g}_2 \quad \ldots \quad \underline{g}_{N-1}$$

เมื่อ $\underline{g}_\lambda$ กิ่งที่ $\lambda$(บล็อคที่ $\lambda$ ) ของ g(1) เจนเนอเรเตอร์แมททริกซ์ของรหัสคือ

$$G = \begin{bmatrix} g_\infty(1) \\ Dg_\infty(1) \\ D^2g_\infty(1) \\ D^3g_\infty(1) \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \begin{bmatrix} \underline{g}_0 & \underline{g}_1 & \underline{g}_2 & \underline{g}_3 & \ldots & \underline{g}_{N-1} & 0 & 0 & 0 & \ldots \\ 0 & \underline{g}_0 & \underline{g}_1 & \underline{g}_2 & \ldots & \underline{g}_{N-2} & \underline{g}_{N-1} & 0 & 0 & \ldots \\ 0 & 0 & \underline{g}_0 & \underline{g}_1 & \ldots & \underline{g}_{N-3} & \underline{g}_{N-2} & \underline{g}_{N-1} & 0 & \ldots \\ 0 & 0 & 0 & \underline{g}_0 & \ldots & \underline{g}_{N-4} & \underline{g}_{N-3} & \underline{g}_{N-2} & \underline{g}_{N-1} & \ldots \\ & & & & \downarrow & & & & & \\ & & & & \infty & & & & & \end{bmatrix} \quad (2.28)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ 0 คือ บล็อคของค่าศูนย์ ให้ $m = (m_0\ m_1\ m_2\ m_3\ \ldots)$ เป็นลำดับของรหัสข่าวสารที่จะนำมาเข้ารหัสให้เป็น $c = (c_0\ c_1\ c_2\ c_3\ \ldots)$ จะได้ว่า $c_\lambda$ คือ รหัสบล็อคที่ $\lambda$

$$\underline{C}_\lambda = C_\lambda(1)C_\lambda(2).....C_\lambda(n)$$

นั้นคือ

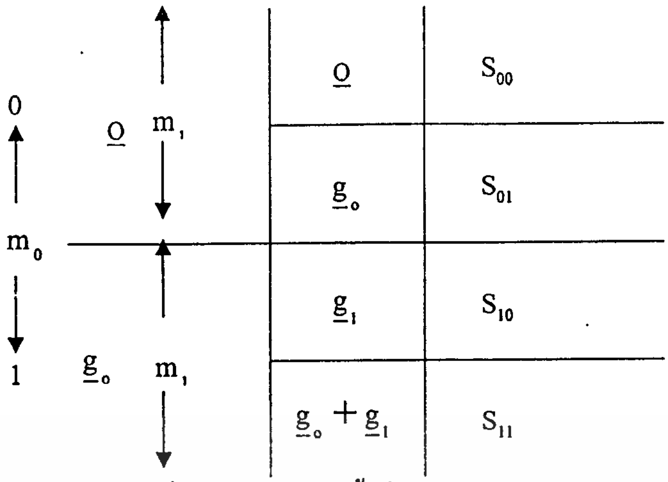
$$C = (\underline{C}_0\underline{C}_1\underline{C}_2\underline{C}_3\ldots.)$$
$$= mG_\infty$$
$$= mG_\infty$$
$$= mG_\infty = m_0 g_\infty(1) + m_1 Dg_\infty(1) + m_2 D^2 g_\infty(1) + \qquad (2.29)$$

จากการทดสอบสมการที่ (2.20) หรือ (2.21) จะสังเกตพบว่า $\underline{C}_0 = \underline{0}$ ถ้า $m_0 = 0$ และ $\underline{C}_0 = \underline{g}_0$ ถ้า $m_0 = 1$ นั้นคือรหัสสามารถแบ่งออกเป็นสองซับเซทที่ขนาดเท่ากัน ซับเซทแรกคือ $S_0$ ที่ประกอบทุกลำดับของรหัสที่สอดคล้องกับ $m_0 = 0$ ดังนั้นทุกลำดับของรหัสในซับเซทนี้จะนำหน้าด้วย 0 ส่วนอีกซับเซทหนึ่งคือ $S_1$ ซึ่งประกอบด้วยลำดับของรหัสที่สอดคล้องกับ $m_0 = 1$ ทุกลำดับของรหัสที่อยู่ในซับเซทนี้จะนำหน้าด้วย $\underline{g}_0$ การแบ่งชุดของสัญญาณดังแสดงในรูปที่ 2.42 ขณะเดียวกันก็สามารถที่จะแบ่ง $S_0$ ออกเป็นซับเซทที่เท่ากันได้อีกสองซับเซทคือ $S_{00}$ และ $S_{01}$ ชุดย่อยของ $S_{00}$ ประกอบด้วยทุกลำดับของรหัสที่สอดคล้องกับ $m_0 = m_1 = 0$ ดังนั้นทุกลำดับของรหัสใน $S_{00}$ จะมีรหัสข้างหน้าเป็น $\underline{0}\ \underline{0}$ ส่วนชุดของ $S_{01}$ จะประกอบด้วยลำดับของสัญญาณที่สอดคล้องกับ $\underline{m} = 0$ และ $m = 1$ ทุกลำดับของรหัสใน $S_{01}$ จะถูกเติมข้างหน้าด้วย $\underline{0}\ \underline{g}_0$ ทำนองเดียวกัน $S_1$ ก็จะถูกแบ่งออกเป็นซับเซทที่เท่ากัน 2 ชุดคือ $S_{10}$ และ $S_{11}$ $S_{10}$ นี้จะให้ลำดับของรหัสที่สอดคล้องกับ $m_0 = 1$ และ $m_1 = 0$ ทุกลำดับของรหัสใน $S_{10}$ จะนำหน้าด้วย $\underline{g}_0\ \underline{g}_1$ ส่วนใน $S_{11}$ จะประกอบด้วยลำดับของรหัสที่สอดคล้องกับ $m_0 = m_1 = 1$ ทุกลำดับของรหัสใน $S_{11}$ จะมีรหัสนำหน้าที่เหมือนกันคือ $\underline{g}_0\ \underline{g}_0 + \underline{g}_1$ การแบ่งครั้งที่สองดังแสดงในรูปที่ 2.43 การแบ่งทำได้เรื่อย ๆ จนถึงอนันต์

| | |
|---|---|
| $\underline{0}$ | $S_0$ |
| $\underline{g}_0$ | $S_1$ |

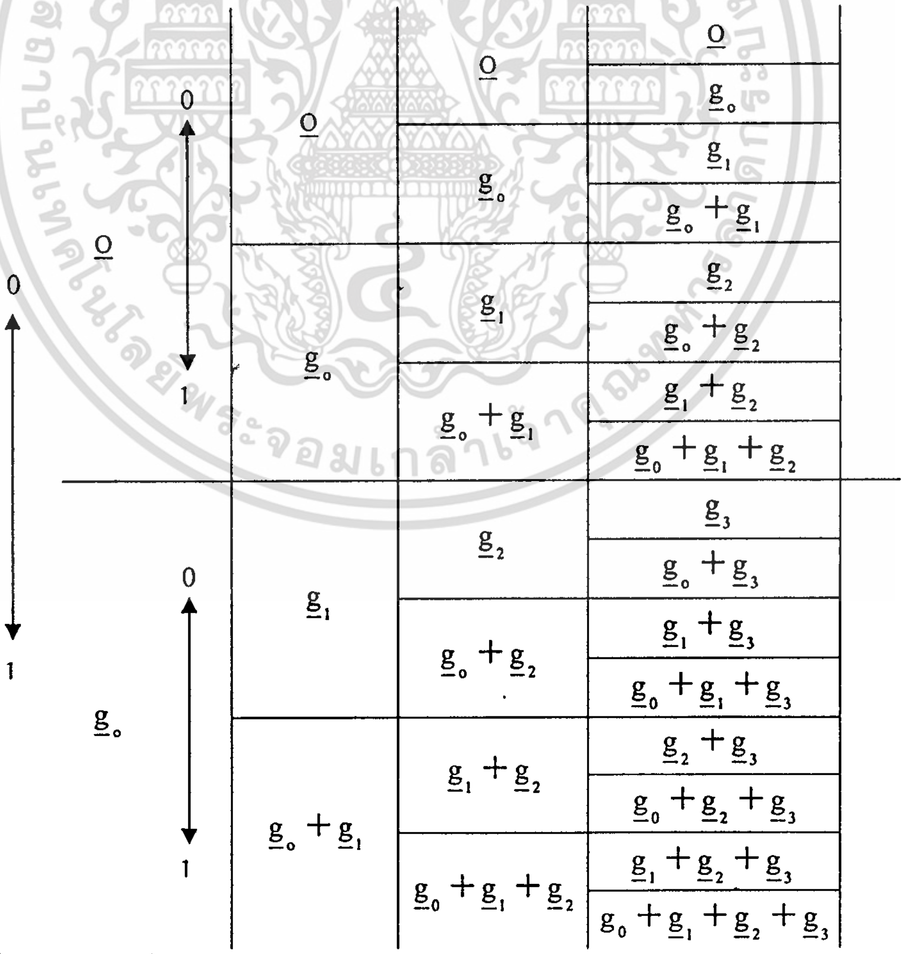
**รูปที่ 2.42** การแบ่งครั้งแรก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



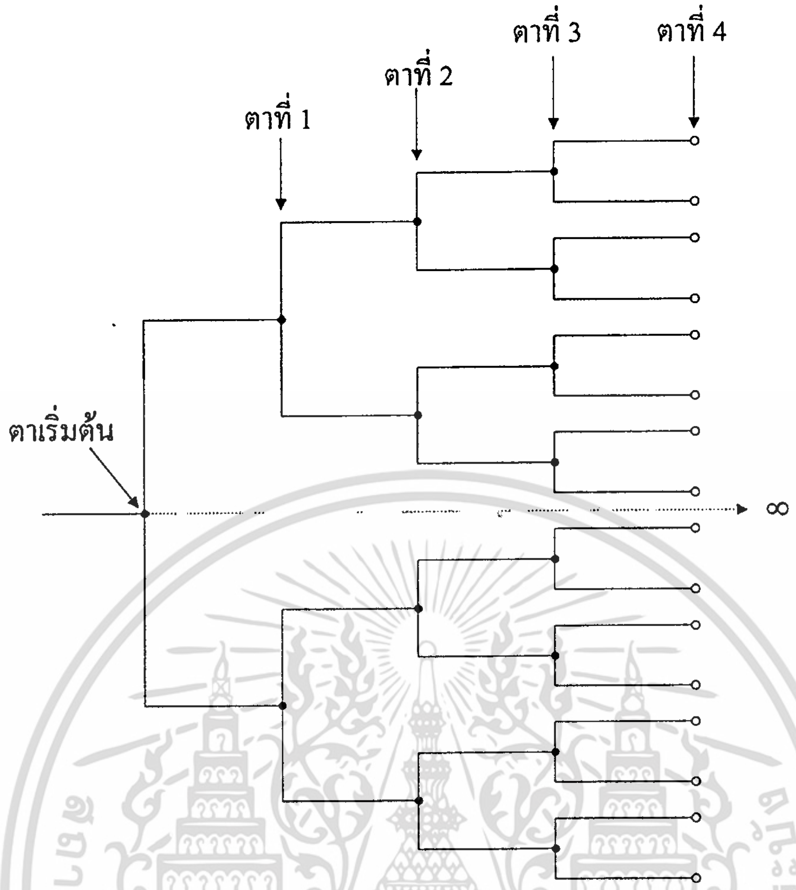

**รูปที่ 2.43** การแบ่งครั้งที่สอง

ทุกลำดับของรหัสสามารถเขียนอยู่ในรูปต้นไม้ที่มี n ตา แต่ละตาจะแยกออกได้ 2 กิ่ง แต่ละกิ่งจะมีรหัส n บิทประกอบกันเป็นบล็อคที่สอดคล้องกับรหัสข่าวสาร ลำดับของรหัสจะผ่านต้นไม้นี้ รูปที่ 2.44 และ 2.45 แสดงให้เห็นถึงลักษณะของรหัสในส่วนต่าง ๆ ของต้นไม้




**รูปที่ 2.44** ตารางที่แสดงการพัฒนาของต้นไม้ที่ใช้สำหรับ (n,1) คอนโวลูชันแนลโค้ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.45** ลักษณะการกระจายกิ่งของ (n,1) คอนโวลูชันแนลโค้ด

เช่นกำหนดให้มีรหัส (3,1)เมื่อ N = 4 โดยการใช้ซับเจนเนอเรเตอร์

g (1,1) = 1 0 1 1

g (1,2) = 1 1 0 1

จะได้เจนเนอเรเตอร์ของรหัสเป็น

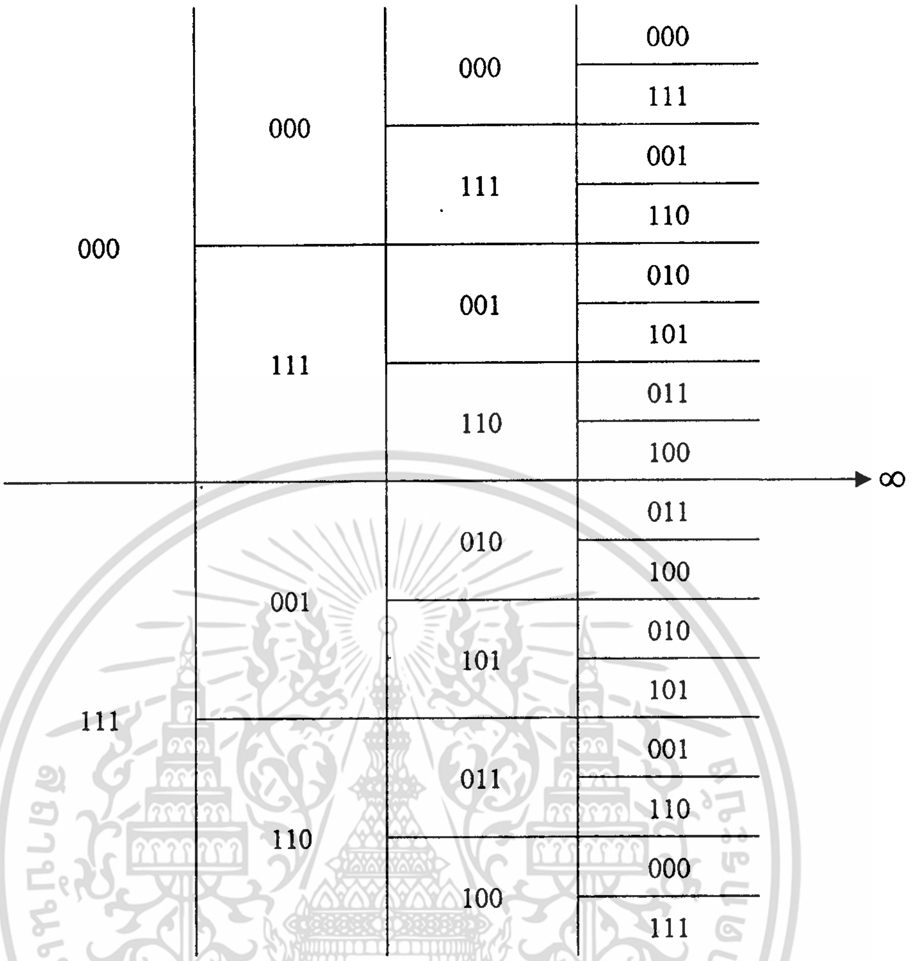
g(1) = (111 001 010 011)

ดังนั้นเจนเนอเรเตอร์แมททริกซ์คือ

$$G_{\infty} = \begin{bmatrix} 1&1&1&0&0&1&0&1&0&0&1&1&&&&&&&&& \\ &&&1&1&1&0&0&1&0&1&0&0&1&1&&&&&& \\ &&&&&&1&1&1&0&0&1&0&1&0&0&1&1&&& \\ &&&&&&&&&1&1&1&0&0&1&0&1&0&0&1&1 \\ &&&&&&&&&&\downarrow&&&&&&&&&& \\ &&&&&&&&&&\infty&&&&&&&&&& \end{bmatrix}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปของต้นไม้ที่สอดคล้องกับรหัสที่ได้ดังแสดงในรูปที่ 2.46




รูปที่ 2.46 โครงสร้างของ (3,1) คอนโวลูชันแนลโค้ดของตัวอย่างที่ 2.6

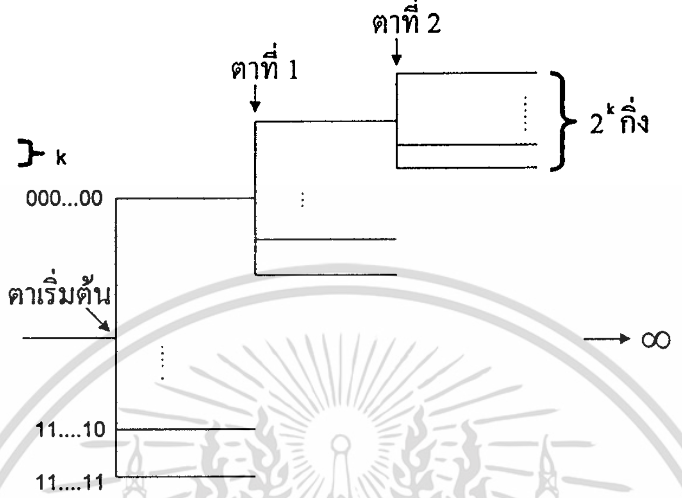
การทำงานในการเข้ารหัสทำได้โดยการป้อนรหัสข่าวสารผ่านไปตามกิ่งของต้นไม้แต่ละบิทของรหัสข่าวสารให้คำสั่งการทำงานดังต่อไปนี้ ที่ตาที่กำหนดให้ตัวถอดรหัสจะเลือกกิ่งบนเป็นบล็อคของรหัสออกถ้ารหัสข่าวสารเข้าเป็น 0 ในทางตรงกันข้ามตัวถอดรหัสจะเลือกกิ่งล่าง ดังนั้นลำดับของการเลือกตาที่ตามมาจะเหมือน ๆ กันทำให้ได้ลำดับของรหัสออกจากตัวเข้ารหัสจากต้นไม้ในรูปที่ 2.46 ถ้ามีลำดับของรหัสข่าวสารเป็น m = (1 0 0 1.....) ก็จะได้

C = (111 001 010 100….)

จากต้นไม้ที่มีกิ่งอนันต์ที่ได้จาก (n,1) คอนโวลูชันแนลโค้ดนี้สามารถขยายให้เป็น (n,k) คอนโวลูชันแนลโค้ดได้โดยจะมีความน่าจะเป็นไปได้ $2^k$ แบบจากรหัสข่าวสาร k บิท ณ หน่วยเวลาที่ 0 ทุกลำดับของรหัสจะแบ่งออกเป็น $2^k$ ซับเซทซึ่งมี $S_0$ , $S_1$ , $S_2$ ,.....S ในเซทของ $S_i$ ก็สามารถแบ่งย่อยออกไปได้อีก $2^k$ ซับเซท การแบ่งตาอย่างนี้สามารถทำไปได้เรื่อย ๆ จนถึงอนันต์ ดังแสดงในรูปที่ 2.47 ที่หน่วยเวลา $\lambda$ ตัวถอดรหัสจะอยู่ ณ ตาอันดับที่ $\lambda$ ถ้ารหัสข่าวสารเข้ามา เป็นบล็อค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คือ $m_\lambda(1).....m_\lambda(k)$ ตัวถอดรหัสจะเลือกกิ่งที่ประกอบด้วยบล็อคของข่าวสารเป็น บล็อคของรหัสออก ลักษณะต้นไม้ที่ใช้กับคอนโวลูชันแนลโค้ดมีประโยชน์มากในการใช้อธิบายถึงความน่าจะเป็นของการถอดรหัส ที่เป็นที่รู้จักกันในชื่อของ sequential decoding




รูปที่ 2.47 ลักษณะต้นไม้ที่ใช้สำหรับ (n,k) คอนโวลูชันแนลโค้ด

คุณสมบัติของระยะห่างของคอนโวลูชั่นโค้ด เมื่อพิจารณาจากต้นไม้ที่มีตาจำกัด โดยที่มีความยาวของกิ่งเป็น N ซึ่งเป็นการตัดกิ่งของต้นไม้ที่มีความยาวอนันต์ ต้นไม้ $\Omega$ ที่มีความยาวจำกัดนี้จะมีทางเดินของรหัสอยู่ $2^{kN}$ ของความยาว N กิ่ง ทางเดินของรหัสใน $\Omega$ เรียกว่า initial code vectors และทุก ๆ เวคเตอร์จะให้ (nN,kN) รหัสเชิงเส้นที่ได้จากซับแมทริกซ์ (kN x nN) ที่อยู่มุมบนซ้ายมือของเจนเนอเรเตอร์แมทริกซ์ $G_\infty$ ของสมการที่ (2.12)

$$G=\begin{bmatrix} I_k & P_0 & 0 & P_1 & 0 & P_2 & \cdots & 0 & P_{N-1} \\ & & I_k & P_0 & 0 & P_1 & \cdots & 0 & P_{N-2} \\ & & & & I_k & P_0 & \cdots & 0 & P_{N-3} \\ & & & & & & & & \vdots \\ & & & & & & & I_k & P_0 \end{bmatrix} \qquad (2.30)$$

พาริตี้เช็คแมทริกซ์ คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$H = \begin{bmatrix} P_0^T & I_{n-k} & & & & & & & \\ P_1^T & 0 & P_0^T & I_{n-k} & & & & & \\ P_2^T & 0 & P_1^T & 0 & P_0^T & I_{n-k} & & & \\ . & & & & & & & & \\ . & & & & & & & & \\ P_{N-1}^T & 0 & P_{N-2}^T & 0 & P_{N-3} & 0 & \ldots & P_0^T & I_{n-k} \end{bmatrix} \qquad (2.31)$$

โดยพารีตี้เช็คแมททริกซ์มีขนาด (n-k)N x nN ซึ่งเป็นซับแมททริกซ์ที่อยู่มุมบนซ้ายมือของ $H_\infty$ ในสมการที่ (2.10) โดยจะได้ $GH^T = 0$ ดังนั้นเวคเตอร์ V จะเป็นโค้ดเวคเตอร์เริ่มต้น (initial code vector) ขนาด nN บิท ที่อยู่ใน $\Omega$ ถ้าและเพียงแต่ถ้า

$$V.H^T = 0$$

รหัสต้นไม้ $\Omega$ ที่มีความยาวจำกัดสามารถแบ่งออกเป็น $2^k$ ซับเซท $[S_0]$, $[S_1]$,...$[s\ 2^k - 1]$ แต่ละซับเซท $[S_i]$ ประกอบด้วย $2^{k(N-1)}$ โค้ดเวคเตอร์เริ่มต้นที่มีบล็อคที่ 0 เหมือน ๆ กัน สำหรับโค้ดเวคเตอร์เริ่มต้นจากสองซับเซทที่ต่างกันจะแตกต่างกันในบล็อคที่ 0 ให้ d(U,V) เป็นระยะห่างแบบแฮมมิ่งระหว่างโค้ดเวคเตอร์เริ่มต้นทั้งสองคือ U กับ V ซึ่งแตกต่างกันในบล็อคที่ 0

คำนิยามที่ 1.1 ระยะห่างต่ำสุด $d_{min}$ ของ (n,k) คอนโวลูชันแนลโค้ดจะเท่ากับระยะห่างแบบแฮมมิ่ง d(U,V) ที่มีค่าน้อยที่สุดระหว่างโค้ดเวคเตอร์เริ่มต้นทั้งสองคือ U กับ V ที่แตกต่างกัน ในบล็อคที่ 0 ทางคณิตศาสตร์เขียนได้เป็น

$$d_{min} = \min_{\substack{i,j \\ i \neq j}} \{d(U,V) : U \varepsilon [S_i] \text{ และ } V \varepsilon [S_j]\} \qquad (2.32)$$

ให้ Z = U+V ดังนั้น Z ก็ยังคงเป็นโค้ดเวคเตอร์เริ่มต้น แต่เนื่องจาก U กับ V แตกต่างกันในบล็อคที่ 0 บล็อคที่ 0 ของ z ต้องไม่เป็นศูนย์หมด ให้ w(z) เป็นเวทของ z จะได้ว่า d(U,V) = w(z) ซึ่งจะเป็นไปตามคำนิยามข้างบนที่กล่าวไว้ว่าระยะห่างต่ำสุดของคอนโวลูชันแนลโค้ด จะเท่ากับเวทต่ำสุดของโค้ดเวคเตอร์เริ่มต้นของบล็อคที่ 0 ที่ไม่เป็นศูนย์ สำหรับการแก้รหัสที่ผิดอย่างแรนดอมระยะห่างต่ำสุดเป็นพารามิเตอร์ที่สำคัญที่จะบ่งบอกถึงความสามารถในการแก้รหัสที่ผิด

ให้ e เป็นลำดับของรหัสที่ผิดที่สามารถแก้ได้ใน (n,k) คอนโวลูชันแนลโค้ดจากการตรวจสอบซีนโดรม N(n-k) บิทจากหน่วยเวลาที่ $\lambda$ ถึง $(\lambda+N-1)$ ตัวถอดรหัสจะประมาณบิทที่ผิดได้ถูกต้องในบล็อคที่ $\lambda$ของ e สำหรับ $\lambda$=0,1,2,... เนื่องจากการถอดรหัสของบล็อคที่รับได้ใด ๆ จะเหมือนกันกับการถอดรหัสในบล็อคที่ 0 ที่รับได้ ดังนั้น ณ ที่นี้จะพิจารณาถึงการถอดรหัสในบล็อคที่ 0 ที่รับได้เท่านั้น ให้ $e_1$ และ $e_2$ เป็นลำดับของบิทที่ผิดที่แก้ได้ซึ่งจะแตกต่างกันในบล็อคที่ 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความต้องการที่ใช้ในการถอดรหัสได้อย่างถูกต้องในบล็อคที่ 0 คือซินโดรม N(n-k) บิทที่ได้จาก $e_1$ และ $e_2$ จากหน่วยเวลาที่ 0 ถึงที่ (N-1) จะต้องไม่เหมือนกัน มิฉะนั้นแล้วจากการ ตรวจสอบซินโดรมตัวถอดรหัสจะไม่สามารถบอกได้ว่าปรากฏบล็อคที่ 0 ของ $e_2$ สมมติว่าลำดับของบิทที่ผิดทั้งสองแตกต่างกันในบล็อคที่ 0 ให้ซินโดรมที่แตกต่างกันจากหน่วยเวลาที่ 0 ถึง (N-1) ดังนั้นสามารถแสดงให้เห็นถึงการถอดรหัสของบิทที่ผิดในบล็อคที่ 0 จะเป็นหนึ่งเดียวและการถอดรหัสจะถูกต้อง ถ้าลำดับของรหัสที่ผิดไปเกิดจากสัญญาณรบกวนในช่องส่งที่สามารถแก้ไขได้จะเห็นได้จากการที่ป้อนทุกลำดับของบิทที่ผิดเข้าไปในซับเซทของ $E_i$ แต่ละซับเซท $E_i$ ประกอบด้วยทุกลำดับของบิทที่ผิดที่มีรหัสในบล็อคที่ 0 เหมือนกัน $\underline{e}_0^{(i)} = (e_0(1)e_0(2)............e_0(n))^{(i)}$ ดังนั้น $E_i$ จะสอดคล้องกับ $\underline{e}_0^{(i)}$ แบบหนึ่งต่อหนึ่ง สำหรับแต่ละ $E_i$ ต้องคำนวณซินโดรม $A_i$ ที่สอดคล้องจากเงื่อนไขที่กำหนดให้จะได้ความสอดคล้องระหว่าง $A_i$ และ $e_0^{(i)}$ เป็นแบบหนึ่งต่อหนึ่ง จากการตรวจสอบซินโดรม N(n-k) บิท ตัวถอดรหัสสามารถบอกรูปแบบของบิทที่ผิดในบล็อคที่ 0 ที่รับได้และการถอดรหัสจะถูกต้อง ดังนั้นพอสรุปได้ว่าเงื่อนไขที่จำเป็นและพอเพียงสำหรับเซทของลำดับรหัสที่ผิดที่แตกต่างกันใน บล็อคที่ 0 จะสามารถแก้ได้โดยจะให้ซินโดรมที่แตกต่างกันจากหน่วยเวลาที่ 0 ถึงหน่วยเวลาที่ (N-1) ด้วยพื้นฐานแห่งความเป็นจริงนี้รหัสแบบคอนโวลูชันแนลสามารถออกแบบเพื่อใช้แก้รหัสที่ผิดแบบแรนดอมหรือแบบเบอสท์หรือทั้งสองแบบ

ให้ $d_{min}$ เป็นระยะห่างต่ำสุดของ (n,k) คอนโวลูชันแนลโค้ด ความสามารถในการแก้รหัสที่ผิดแบบแรนดอมของรหัสนี้ได้จากทฤษฎีที่ 1.1

ทฤษฎีที่ 1.1 ให้ $e_1$ และ $e_2$ เป็นลำดับของรหัสที่ผิดซึ่งมีความยาว N บล็อคที่แตกต่างกันในบล็อคที่ 0 ถ้าทั้ง $e_1$ และ $e_2$ ประกอบรหัสที่ผิดไปจำนวนเท่ากับหรือน้อยกว่า $[(d_{min}-1)]$ ดังนั้น $e_1$ และ $e_2$ จะมีซินโดรมที่ต่างกันกล่าวคือ

$$e_1H^T \neq e_2H^T$$

พิสูจน์ เนื่องจาก $e_1$ และ $e_2$ แตกต่างกันในบล็อคที่ 0 ดังนั้นบล็อคที่ 0 ของ $e_1 + e_2$ ต้องไม่เป็นศูนย์ สมมติว่า $e_1H^T = e_2H^T$ จะได้

$$(e_1 + e_2)H^T = 0$$

เพราะฉะนั้น $e_1 + e_2$ ต้องเป็นโค้ดเวคเตอร์เริ่มต้นที่ไม่เป็นศูนย์ในบล็อคที่ 0 แต่เป็นไปไม่ได้เนื่องจากเวทของ $e_1 + e_2$ นี้น้อยกว่า $d_{min}$ ซึ่งจะเท่ากับเวทต่ำสุดของโค้ดเวคเตอร์เริ่มต้นที่ไม่เป็นศูนย์ในบล็อคที่ 0 ดังนั้นจะได้ $e_1H^T \neq e_2H^T$

ทฤษฎีนี้แสดงให้เห็นว่าลำดับของรหัสที่ผิดใด ๆ ที่มีจำนวนบิทที่ผิดเท่ากับหรือน้อยกว่า $[(d_{min}-1)/2]$ ที่ขยายออกไปของ nN ตำแหน่งที่ต่อเนื่องกันสามารถแก้ไขได้ และสามารถแสดงต่อไปได้ว่า สำหรับลำดับของรหัสที่ผิดที่มี $[(d_{min}-1)/2]+1$ บิท ในการขยายครั้งที่ 1 ของ nN ตำแหน่งที่ต่อเนื่องกันจะไม่สามารถแก้ไขได้ ให้ U เป็นโค้ดเวคเตอร์เริ่มต้นที่ไม่เป็นศูนย์ในบล็อค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ 0 ที่มีเวท w(U) เท่ากับ $d_{min}$ เราสามารถสร้างลำดับของรหัสที่ผิด $e_1$ และ $e_2$ จาก U โดยมีคุณสมบัติดังนี้

1) $e_1 + e_2 = U$

2) เซทของรหัสที่ผิดใน $e_1$ และ เซทของรหัสที่ผิดใน $e_2$ ต่างก็ไม่ต่อเนื่องกัน (เห็นได้อย่างชัดเจนว่า $e_1$ และ $e_2$ แตกต่างกันในบล็อคที่ 0)

3) เวท $w(e_1)$ ของ $e_1$ เท่ากับ $[(d_{min}-1)/2]+1$ และเวท $w(e_2)$ของ $e_2$ เท่ากับ $d_{min}-[(d_{min}-1)/2]-1$

จากคุณสมบัติข้อแรกจะได้ $UH^T = (e_1 + e_2)H^T = 0$ คือ $e_1H^T = e_2H^T$ เนื่องจาก $e_1$ และ $e_2$ ได้ซีนโดรมเหมือนกัน อย่างน้อยที่สุดหนึ่งในสองลำดับของรหัสที่ผิดจะไม่สามารถแก้ไขได้ มี 2 กรณีที่ถูกพิจารณา กรณีแรกคือถ้า $d_{min}$ เป็นเลขคี่จะได้

$$\frac{d_{min}-1}{2} = w(e_2) < w(e_1) = \frac{d_{min}-1}{2} + 1$$

หมายความว่า $e_2$ สามารถแก้ได้ ดังนั้น $e_1$ ที่มีบิทที่ผิดอยู่เท่ากับ $[(d_{min}-1)/2]+1$ จะไม่สามารถแก้ไขได้ กรณีที่สองถ้า $d_{min}$ เป็นเลขคู่จะได้

$$w(e_1) = w(e_2) = \frac{d_{min}-1}{2} + 1$$

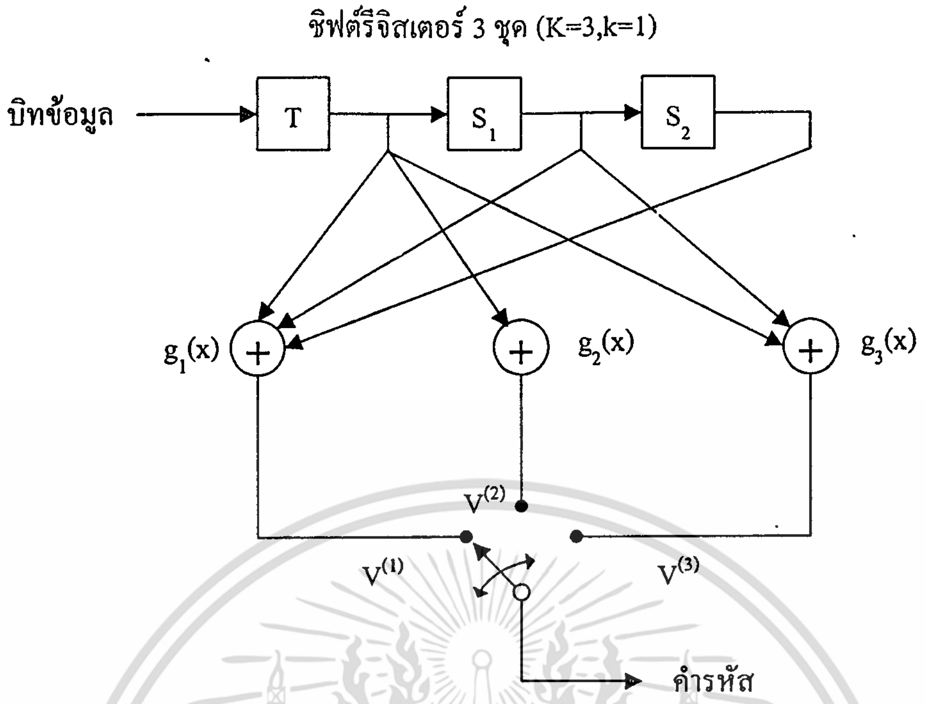
ในกรณีนี้อย่างน้อยที่สุดหนึ่งในสองไม่สามารถแก้ไขได้ ดังนั้นสรุปได้ว่ารหัสแบบคอนโวลูชั่นแนลที่มีระยะห่างต่ำสุด $d_{min}$ จะสามารถแก้ลำดับของรหัสที่ผิดใด ๆ ที่มีจำนวนบิทที่ผิดเท่ากับหรือน้อยกว่า $[(d_{min}-1)/2]$ ในตำแหน่งที่ต่อเนื่องกัน nN ตำแหน่งที่ขยายออกไป

จะเห็นได้ชัดว่า (n,k) คอนโวลูชันแนลโค้ดสามารถแก้รหัสที่ผิดอย่างแรนดอมได้โดยรหัสควรจะมีค่า $d_{min}$ สูงพอภายใต้เงื่อนไขบังคับของความยาว N อย่างไรก็ดียังไม่มีวิธีการอัตโนมัติที่จะสร้างรหัสแบบคอนโวลูชันแนลที่ให้ $d_{min}$ ภายใต้เงื่อนไขบังคับของความยาว N

## 2.4 การถอดรหัสคอนโวลูชั่น

การถอดรหัสที่ภาครับสามารถกระทำได้หลายวิธี แต่วิธีที่จะนำเสนอในที่นี้คือวิธีของ Viterbi เพราะเป็นวิธีที่มีประสิทธิภาพและนำมาใช้งานได้ดี การทำงานของวิธี Viterbi อาศัยแผนภาพเทรลลิสเป็นกลไกพื้นฐานในกระบวนการถอดรหัส เราจะอาศัยตัวอย่างการถอดรหัสของคำรหัสที่มีขนาดสั้น ๆ เพื่อช่วยประกอบในการอธิบายกรรมวิธีการถอดรหัสแต่ละขั้นโดยละเอียด คำรหัสตัวอย่างที่จะใช้ในการถอดรหัส คือ 111 010 001 011 101 011 101 100 ซึ่งได้จากการเข้ารหัสข้อมูล 110101 โดยอาศัยวงจรเข้ารหัสในรูปที่ 2.48 เพื่อให้เห็นภาพว่ารหัสคอนโวลูชั่นมีขีดความสามารถในการแก้ไขความผิดพลาดที่อาจเกิดขึ้นระหว่างการส่งผ่านช่องสัญญาณ ในที่นี้จะสมมติให้คำรหัสที่ได้รับมีความผิดพลาดเกิดขึ้น 2 บิท คือ บิทที่ 2 และ 5 ฉะนั้นชุดบิทที่จะนำไปผ่านการถอดรหัสจึงมีค่าเท่ากับ 1<u>0</u>1 0<u>0</u>0 001 011 101 011 101 100

เอกสารนี้เป็นเอกสารที่มีลิขสิทธิ์ ให้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.48 ตัวอย่างวงจรเข้ารหัสคอนโวลูชั่น**

ในการถอดรหัสเราจะต้องเริ่มพิจารณาที่สถานะ A เสมอ ขั้นแรกให้มองลึกเข้าไปในแผนภาพเทรลลิส 3 ระดับ และพิจารณาดูเส้นทางแต่ละเส้นซึ่งมีทั้งหมด 8 เส้นทางว่ามีความคล้ายคลึงกับชุดบิทที่ได้รับมากน้อยเท่าใด โดยอาศัยการวัดค่าระยะแฮมมิงระหว่างเส้นทางเหล่านั้นกับชุดบิทที่รับได้ 9 บิทแรก รายละเอียดของการเปรียบเทียบทั้งหมดได้แสดงไว้ในตารางที่ 2.3

**ตารางที่ 2.3** การหาระยะแฮมมิงระหว่างชุดบิทที่รับได้ 101000001 เฉพาะ 9 บิทแรกกับรหัสทั้ง 8 เส้นทาง

| เส้นทาง | รหัส | ระยะแฮมมิง | Survivor? |
|---|---|---|---|
| AAAA | 000000000 | 3 | Yes |
| ACBA | 111101100 | 5 | No |
| ACDB | 111010001 | 2 | Yes |
| AACB | 000111101 | 6 | No |
| AAAC | 000000111 | 4 | No |
| ACBC | 111101011 | 4 | Yes |
| ACDD | 111010110 | 5 | Yes |
| AACD | 000111010 | 7 | No |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากค่าระยะแฮมมิงที่ได้ก็นำมาเปรียบเทียบเป็นคู่ทั้งหมด 4 คู่ เช่นคู่แรกที่สิ้นสุดที่สถานะ A เส้นทาง AAAA มีขนาดของระยะแฮมมิงเท่ากับ 3 ซึ่งสั้นกว่าเส้นทาง ACBA ดังนั้นเส้นทางที่สั้นกว่านี้จะถูกเลือก หรือกล่าวว่าเส้นทางนี้เป็นเส้นทาง Survivor คู่ที่สองที่ไปสิ้นสุดที่สถานะ B เส้นทาง ACDB มีค่าของระยะแฮมมิงที่สั้นกว่าเส้นทาง AACB อย่างชัดเจน ดังนั้นเส้นทางที่จะเลือกเป็น Survivor คือเส้นทาง ACDB ในกรณีของคู่ที่สิ้นสุดที่สถานะ C เส้นทางทั้งสองมีค่าระยะแฮมมิงที่เท่ากัน คือ 4 ให้เลือกเส้นทางใดเส้นทางหนึ่งเป็นเส้นทาง Survivor ก็ได้ ในตัวอย่างได้เลือกเส้นทาง ACBC ส่วนกรณีสุดท้ายได้เลือกเส้นทาง ACDD ไว้เพราะมีค่าระยะแฮมมิงที่สั้นกว่าเส้นทาง AACD ผลลัพธ์ที่ได้คือมี 4 เส้นทางที่เป็นเส้นทางที่เป็นไปได้

**ตารางที่ 2.4** การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิทที่ 10-12 ซึ่งมีค่าเท่ากับ 011

| เส้นทาง | ระยะแฮมมิงของ survivors เดิม | ส่วนของเส้นทางใหม่ | ระยะแฮมมิงเฉพาะส่วนของเส้นทางใหม่ | ระยะแฮมมิงของเส้นทางทั้งหมด | Survivor? |
|---|---|---|---|---|---|
| AAAAA | 3 | AA | 2 | 5 | No |
| ACDBA | 2 | BA | 3 | 5 | Yes |
| ACBCB | 4 | CB | 2 | 6 | No |
| ACDDB | 4 | DB | 1 | 5 | Yes |
| AAAAC | 3 | AC | 1 | 4 | No |
| ACDBC | 2 | BC | 0 | 2 | Yes |
| ACBCD | 4 | CD | 1 | 5 | Yes |
| ACDDD | 5 | DD | 2 | 7 | No |

ในขั้นที่สองเราจะมาดูข้อมูลทีละ 3 บิท หรือเพียงหนึ่งขั้นของการเปลี่ยนสถานะจากเส้นทาง 4 เส้นทางที่ได้คัดเลือกเป็น Survivor จากขั้นแรก โดยเราจะนำสถานะสุดท้ายของแต่ละเส้นมาเป็นจุดเริ่มต้นเพื่อใช้ในการพิจารณาหาเส้นทางต่อไป ค่าระยะแฮมมิงของเส้นทางใหม่ที่ได้ให้นำไปรวมกับค่าระยะแฮมมิงของค่าเดิมที่คำนวณได้ในขั้นก่อน แล้วก็ทำการตัดสินใจเก็บเส้นทางที่สั้นกว่าไว้ รายละเอียดของขั้นตอนนี้สำหรับการพิจารณา 3 บิทถัดมาคือบิทที่ 9-12 ได้แสดงในตารางที่ 2.4 กระบวนการที่ใช้ในขั้นตอนนี้ให้นำมาใช้ซ้ำในการพิจารณาต่อคราวละ 3 บิท จนกระทั่งชุดบิทที่ต้องการถอดรหัสหมดไป สำหรับการพิจารณาบิทที่ 13-15, 16-18, 19-21 และ 22-24 สามารถดูได้จากตารางที่ 2.5-2.8 ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 2.5** การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิทที่ 13-15 ซึ่งมีค่าเท่ากับ 101

| เส้นทาง | ระยะแฮมมิงของ survivors เดิม | ส่วนของเส้นทางใหม่ | ระยะแฮมมิงเฉพาะส่วนของเส้นทางใหม่ | ระยะแฮมมิงของเส้นทางทั้งหมด | Survivor? |
|---|---|---|---|---|---|
| ACDBAA | 5 | AA | 2 | 7 | No |
| ACDDBA | 5 | BA | 1 | 6 | Yes |
| ACDBCB | 2 | CB | 0 | 2 | Yes |
| ACBCDB | 5 | DB | 1 | 6 | No |
| ACDBAC | 5 | AC | 1 | 6 | Yes |
| ACDDBC | 5 | BC | 2 | 7 | No |
| ACDBCD | 2 | CD | 3 | 5 | Yes |
| ACBCDD | 5 | DD | 2 | 7 | No |

**ตารางที่ 2.6** การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิทที่ 16-18 ซึ่งมีค่าเท่ากับ 011

| เส้นทาง | ระยะแฮมมิงของ survivors เดิม | ส่วนของเส้นทางใหม่ | ระยะแฮมมิงเฉพาะส่วนของเส้นทางใหม่ | ระยะแฮมมิงของเส้นทางทั้งหมด | Survivor? |
|---|---|---|---|---|---|
| ACDDBAA | 6 | AA | 2 | 8 | No |
| ACDBCBA | 2 | BA | 3 | 5 | Yes |
| ACDBACB | 6 | CB | 2 | 8 | No |
| ACDBCDB | 5 | DB | 1 | 6 | Yes |
| ACDDBAC | 6 | AC | 1 | 7 | No |
| ACDBCBC | 2 | BC | 0 | 2 | Yes |
| ACDBACD | 6 | CD | 1 | 7 | Yes |
| ACDBCDD | 5 | DD | 2 | 7 | No |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ 2.7** การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิทที่ 19-21 ซึ่งมีค่าเท่ากับ 101

| เส้นทาง | ระยะแฮมมิงของ survivors เดิม | ส่วนของเส้นทางใหม่ | ระยะแฮมมิงเฉพาะส่วนของเส้นทางใหม่ | ระยะแฮมมิงของเส้นทางทั้งหมด | Survivor? |
|---|---|---|---|---|---|
| ACDBCBAA | 5 | AA | 2 | 7 | No |
| ACDBCDBA | 6 | BA | 1 | 7 | Yes |
| ACDBCBCB | 2 | CB | 0 | 2 | Yes |
| ACDBACDB | 7 | DB | 1 | 8 | No |
| ACDBCBAC | 5 | AC | 1 | 6 | Yes |
| ACDBCDBC | 6 | BC | 2 | 8 | No |
| ACDBCBCD | 2 | CD | 3 | 5 | Yes |
| ACDBACDD | 7 | DD | 2 | 9 | No |

**ตารางที่ 2.8** การคำนวณในขั้นที่ 2 ของวิธี Viterbi พิจารณาบิทที่ 22-24 ซึ่งมีค่าเท่ากับ 100

| เส้นทาง | ระยะแฮมมิงของ survivors เดิม | ส่วนของเส้นทางใหม่ | ระยะแฮมมิงเฉพาะส่วนของเส้นทางใหม่ | ระยะแฮมมิงของเส้นทางทั้งหมด | Survivor? |
|---|---|---|---|---|---|
| ACDBCDBAA | 7 | AA | 1 | 8 | No |
| ACDBCBCBA | 2 | BA | 0 | 2 | Yes |
| ACDBCBACB | 6 | CB | 1 | 7 | Yes |
| ACDBCBCDB | 5 | DB | 2 | 7 | No |
| ACDBCDBAC | 7 | AC | 2 | 9 | No |
| ACDBCBCBC | 2 | BC | 3 | 5 | Yes |
| ACDBCBACD | 6 | CD | 2 | 8 | No |
| ACDBCBCDD | 5 | DD | 1 | 6 | Yes |

เมื่อได้ทำตามขั้นตอนที่ 2 จนครบทุกบิทแล้วจะได้เส้นทาง Survivor ทั้งหมด 4 เส้นทางดังนี้ คือ ACDBCBCBCBA, ACDBCBACB, ACDBCBCBC, ACDBCBCDD ซึ่งแต่ละเส้นทางมีระยะแฮมมิงเท่ากับ 2, 7, 5 และ 6 ตามลำดับ เมื่อเปรียบเทียบเส้นทางทั้ง 4 เส้นพบว่าเส้นทางที่สั้นที่สุดก็คือเส้นทาง ACDBCBCBA ซึ่งมีค่าระยะแฮมมิงเท่ากับ 2 ฉะนั้นเราจึงตัดสินใจว่าเส้นทางดังกล่าวนี้คือเส้นทางแทนคำรหัสที่ถูกต้อง และเมื่อแปลงเส้นทางนี้ให้อยู่ในรูปของชุดบิทก็จะได้เป็น 111 010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

001 011 101 011 101 100 ถูกต้องตรงกับคำรหัสที่ต้นทางได้ส่งออก ซึ่งหมายความว่าบิทตำแหน่งที่ 2 และ 5 ซึ่งผิดไปในระหว่างการส่งผ่านช่องสัญญาณได้รับการแก้ไขให้ถูกต้องแล้ว

## 2.5 รหัสเทอร์โบ

### 2.5.1. บทนำ

รหัสเทอร์โบ (Turbo code) เป็นกรรมวิธีในการเข้าและถอดรหัสช่องสัญญาณประเภทใหม่ เพิ่งจะได้รับการค้นพบและพัฒนาขึ้นในปี ค.ศ.1993 โดย C. Berrou, A. Glavieux และ P. Thitimajshima เริ่มเผยแพร่และตีพิมพ์ครั้งแรกในงานประชุมวิชาการนานาชาติ Proceedings of IEEE International Conference on Communications (ICC 1993) หน้า 1064-1070 รหัสเทอร์โบมีคุณสมบัติพิเศษคือ สามารถทำงานได้ดีแม้ช่องสัญญาณจะมีอัตราส่วน $E_b/N_o$ ที่ต่ำมาก มีขีดความสามารถในการจัดการกับปัญหาความผิดพลาดได้อย่างมีประสิทธิภาพ และที่น่าสนใจเป็นพิเศษ คือมีสมรรถนะเข้าใกล้ค่าที่กำหนดจากทฤษฎีของ Shannon โดยอาศัยกระบวนการเข้าและถอดรหัสที่ไม่ซับซ้อนมาก ซึ่งในอดีตที่ผ่านมายังไม่มีวิธีการเข้ารหัสช่องสัญญาณใดสามารถทำได้ หรือถ้าทำได้ก็ต้องใช้อัตราการเข้ารหัสที่สูงประกอบกับวงจรถอดรหัสที่ซับซ้อนมาก ด้วยเหตุนี้การค้นพบรหัสเทอร์โบจึงนับว่าเป็นการค้นพบที่สำคัญและได้เปลี่ยนแปลงงานวิจัยทางด้านการเข้ารหัสช่องสัญญาณไปอย่างมาก ในหลายปีที่ผ่านมาผลงานวิจัยและพัฒนาที่เกี่ยวข้องกับรหัสเทอร์โบจึงมีให้เห็นกันมากมาย การประยุกต์ใช้งานในทางปฏิบัติก็เกิดขึ้นอย่างแพร่หลาย เช่น ในระบบโทรศัพท์เคลื่อนที่ยุคที่สาม (Third generation mobile system) ได้นำรหัสเทอร์โบมากำหนดใช้เป็นมาตรฐานสำหรับการติดต่อระหว่างสถานีฐานและเครื่องโทรศัพท์เคลื่อนที่แล้ว

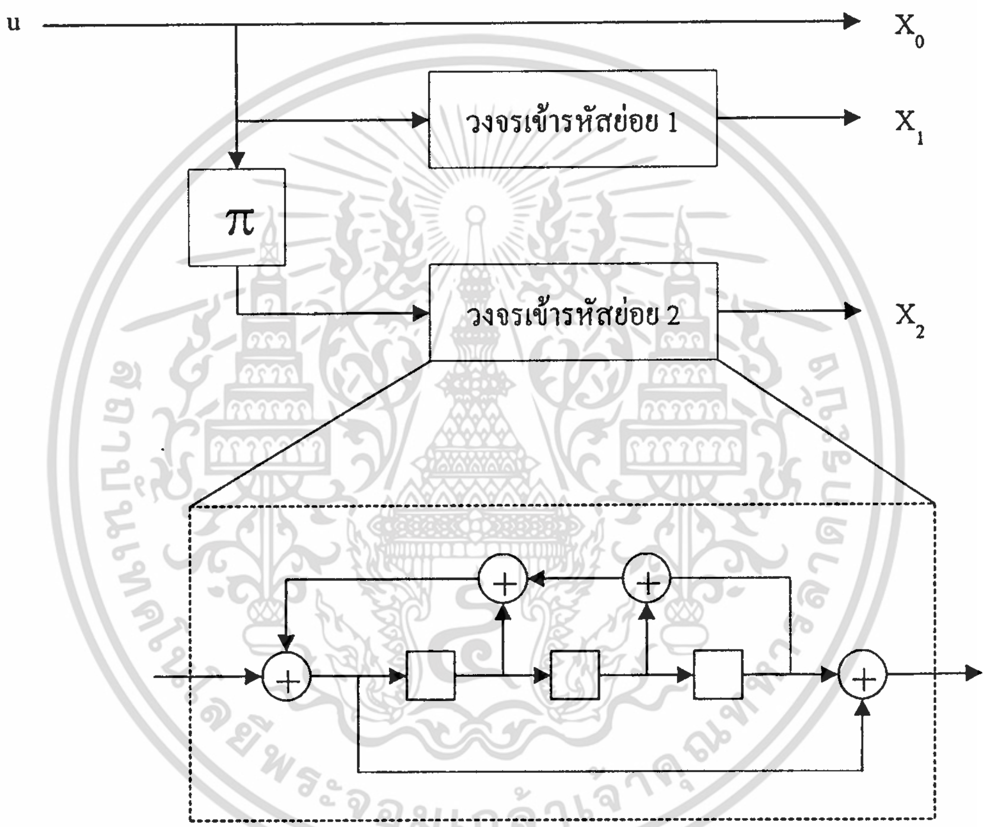
หลักการสำคัญสามประการที่เป็นโครงสร้างพื้นฐานของรหัสเทอร์โบและมีคุณสมบัติที่แตกต่างจากวิธีการเข้าและถอดรหัสช่องสัญญาณประเภทอื่น คือ การเข้ารหัสแบบต่อขนาน (parallel concatenated encoding) การใช้วงจรเข้ารหัสแบบป้อนกลับ และการถอดรหัสแบบวนซ้ำ (iterative decoding) สำหรับในภาคผนวกนี้เราจะกล่าวถึงกระบวนการเข้าและถอดรหัสเทอร์โบโดยจะอาศัยตัวอย่างประกอบการอธิบาย

### 2.5.2 วงจรเข้ารหัสเทอร์โบ

วงจรเข้ารหัสเทอร์โบมีโครงสร้างพื้นฐานดังแสดงในรูปที่ 2.49 จากรูปชุดข้อมูลที่จะเข้ารหัส u ได้รับการป้อนเข้าทางด้านซ้ายของวงจร เพื่อนำไปเป็นสัญญาณเข้าในแต่ละองค์ประกอบย่อยของวงจรเข้ารหัสเทอร์โบทั้ง 3 ส่วน ได้แก่ ส่วนที่แปลงสัญญาณ u ให้เป็นสัญญาณ $X_0$, $X_1$, และ$X_2$ ตามลำดับที่ด้านออกของวงจร สำหรับตัวอย่างโครงสร้างที่ใช้ในการอธิบายนี้เป็นวงจรเข้ารหัสที่มีอัตราการเข้ารหัสเท่ากับ 1/3

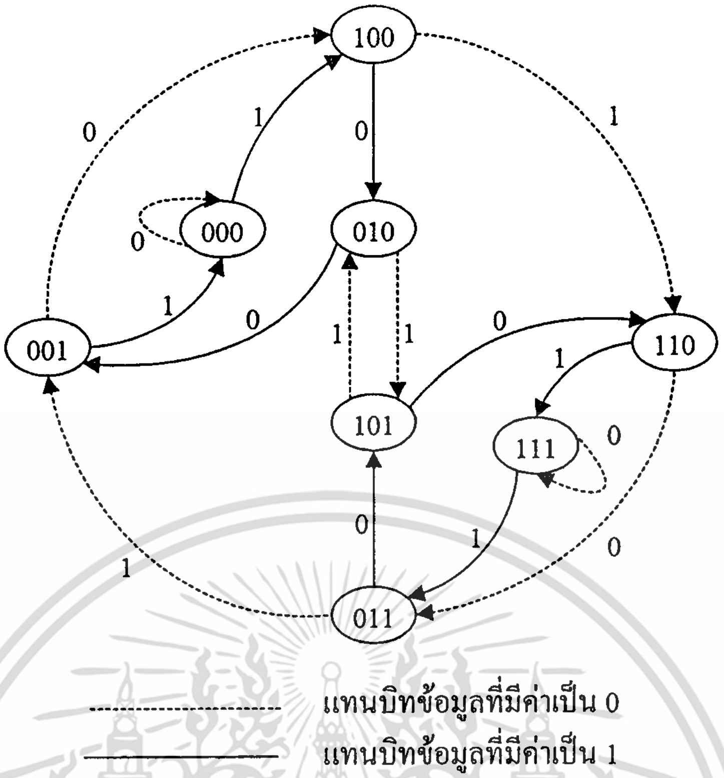
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.49 จะเห็นว่าสัญญาณ $X_0$ เป็นค่าเดียวกับสัญญาณข้อมูลด้านเข้า สำหรับสัญญาณ $X_1$ ได้จากการผ่านชุดข้อมูล u เข้าวงจรเข้ารหัสย่อยหมายเลข 1 สำหรับโครงสร้างภายในของวงจรเข้ารหัสย่อยนี้มีลักษณะดังในรูปที่ 2.50 ส่วนสัญญาณ $X_2$ ได้จากการป้อนชุดบิทข้อมูล u เข้าสู่วงจรอินเทอร์ลีฟเวอร์เพื่อสลับลำดับของชุดบิทข้อมูล u ให้อยู่ในรูปแบบที่แตกต่างจากเดิม จากนั้นป้อนบิทข้อมูลเหล่านี้เข้าสู่วงจรเข้ารหัสย่อยหมายเลข 2 ซึ่งอาจมีโครงสร้างพื้นฐานเหมือนหรือแตกต่างจากวงจรเข้ารหัสหมายเลข 1

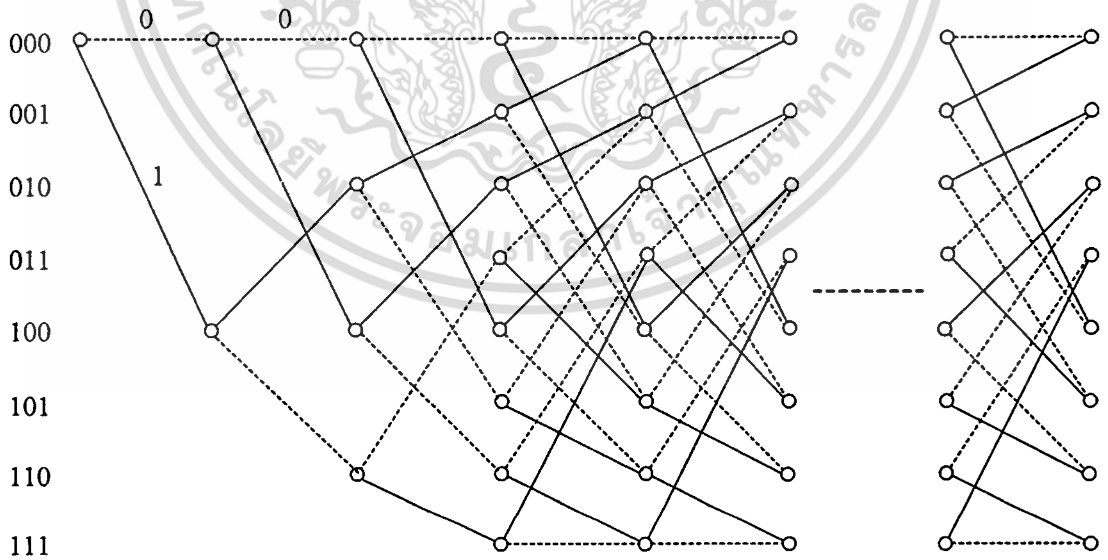



**รูปที่ 2.49 โครงสร้างพื้นฐานของวงจรเข้ารหัสเทอร์โบ**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



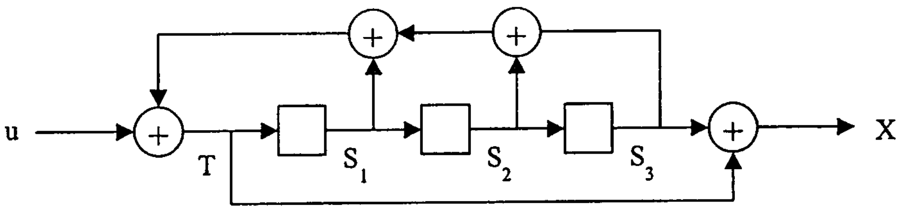

**รูปที่ 2.50** แผนภาพการเปลี่ยนสถานะของวงจรเข้ารหัสย่อยในรูปที่ 2.49




**รูปที่ 2.51** แผนภาพเทรลลิสของวงจรเข้ารหัสย่อยในรูปที่ 2.49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่นจงแสดงขั้นตอนการเข้ารหัสวงจรย่อยหนึ่งตามรูปข้างล่างของรหัสเทอร์โบ โดยกำหนดบิทข้อมูลให้มีค่าเป็น 11011100




**รูปที่ 2.52 ตัวอย่างวงจรเข้ารหัสย่อย**

วิธีทำ

จากรูปที่ 2.52 เมื่อนำบิทข้อมูลมาเข้ารหัสเทอร์โบจะมีขั้นตอนและรายละเอียดดังนี้

ขั้นแรก รีเซตให้สถานะของชิฟต์รีจิสเตอร์ทั้งหมดมีค่าเป็นศูนย์ในที่นี้คือ $S_1$, $S_2$ และ $S_3$ ในขั้นนี้เป็นเพียงการเตรียมความพร้อมของวงจรเข้ารหัสและยังไม่มีการป้อนบิทข้อมูลเข้า ฉะนั้นค่าที่ได้ที่ตำแหน่ง T และ X จึงยังไม่ต้องสนใจ

ขั้นที่สอง เริ่มป้อนบิทแรกซึ่งมีค่าเป็น 1 เข้าสู่วงจร ทำให้ค่าที่ตำแหน่ง $T = u \oplus S_1 \oplus S_2 \oplus S_3 = 1 \oplus 0 \oplus 0 \oplus 0 = 1$ และค่าของ $X = T \oplus S_3 = 1 \oplus 0 = 1$ ซึ่งคือผลการเข้ารหัสของบิทแรกนั่นเอง

ขั้นที่สาม เริ่มป้อนบิทที่สองซึ่งมีค่าเป็น 1 เข้าสู่วงจร ค่าอื่น ๆ ในวงจรก็จะเลื่อนไปหนึ่งบิททั้งหมดดังรูป ทำให้ค่าที่ตำแหน่ง $T = u \oplus S_1 \oplus S_2 \oplus S_3 = 1 \oplus 1 \oplus 0 \oplus 0 = 0$ และค่าของ $X = T \oplus S_3 = 0 \oplus 0 = 0$ ซึ่งคือผลการเข้ารหัสของบิทสอง

ขั้นที่สี่ เริ่มป้อนบิทที่สามซึ่งมีค่าเป็น 0 เข้าสู่วงจร ค่าอื่น ๆ ในวงจรก็จะเลื่อนไปหนึ่งบิทในลักษณะเดิม ทำให้ค่าที่ตำแหน่ง $T = u \oplus S_1 \oplus S_2 \oplus S_3 = 0 \oplus 0 \oplus 1 \oplus 0 = 1$ และค่าของ $X = T \oplus S_3 = 1 \oplus 0 = 1$ ซึ่งคือผลการเข้ารหัสของบิทสาม

ขั้นที่ห้า ป้อนบิทข้อมูลเข้าไปในวงจรทีละบิทจนครบด้วยวิธีการเหมือนเดิมจะได้ผลลัพธ์เป็น 10101110 สังเกตว่าสถานะของวงจรเข้ารหัสเทอร์โบมิได้กลับไปสู่สถานะเริ่มต้นที่เป็นศูนย์หมด ในรูปอยู่ในสถานะ 001 ด้วยเหตุนี้จึงต้องมีการเตรียมบิทหางที่เหมาะสมอีก 3 บิท เพื่อปรับวงจรให้กลับไปสู่สถานะที่ศูนย์ทั้งหมดกระบวนการเข้ารหัสจึงจะเสร็จสมบูรณ์

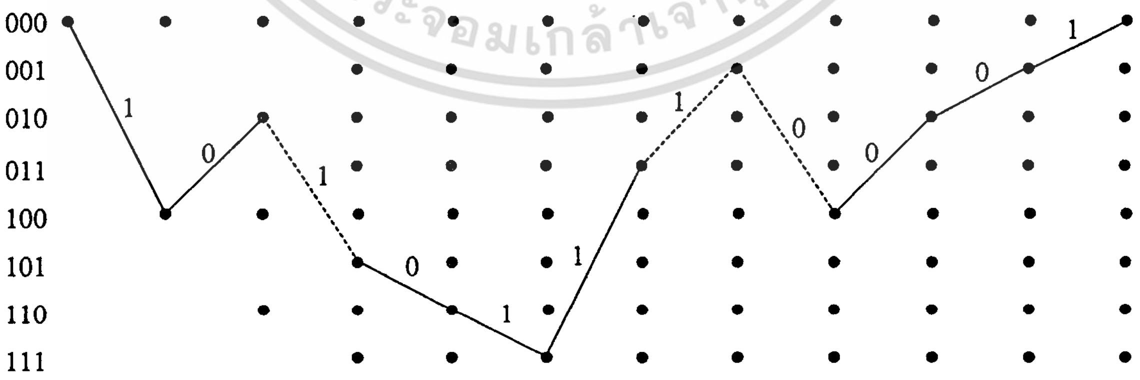
ขั้นสุดท้าย ในการเลือกค่าของบิทหางที่เหมาะสมมีหลักการง่าย ๆ ดังนี้คือให้เลือกป้อนค่า u ที่ทำให้ค่าของบิท T = 0 ในตัวอย่างได้ป้อน u = 1 เข้าไปจึงทำให้ T = 0 ได้ตามต้องการ ทำเช่นนี้ไปจนครบ 3 ครั้ง วงจรเข้ารหัสเทอร์โบก็จะกลับไปสู่สถานะเริ่มต้นที่ศูนย์อีกครั้ง เป็นอันเสร็จสิ้นการเข้ารหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | u | T | $S_1$ | $S_2$ | $S_3$ | X |
|---|---|---|---|---|---|---|
| เซตค่าเริ่มต้น | x | x | 0 | 0 | 0 | x |
| n = 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| n = 2 | 1 | 0 | 1 | 0 | 0 | 0 |
| n = 3 | 0 | 1 | 0 | 1 | 0 | 1 |
| n = 4 | 1 | 1 | 1 | 0 | 1 | 0 |
| n = 5 | 1 | 1 | 1 | 1 | 0 | 1 |
| n = 6 | 1 | 0 | 1 | 1 | 1 | 1 |
| n = 7 | 0 | 0 | 0 | 1 | 1 | 1 |
| n = 8 | 0 | 1 | 0 | 0 | 1 | 0 |
| n = 9 บิทหาง | 1 | 0 | 1 | 0 | 0 | 0 |
| n = 10 | 1 | 0 | 0 | 1 | 0 | 0 |
| n = 11 | 1 | 0 | 0 | 0 | 1 | 1 |
| เลื่อนอีกครั้ง | x | x | 0 | 0 | 0 | x |

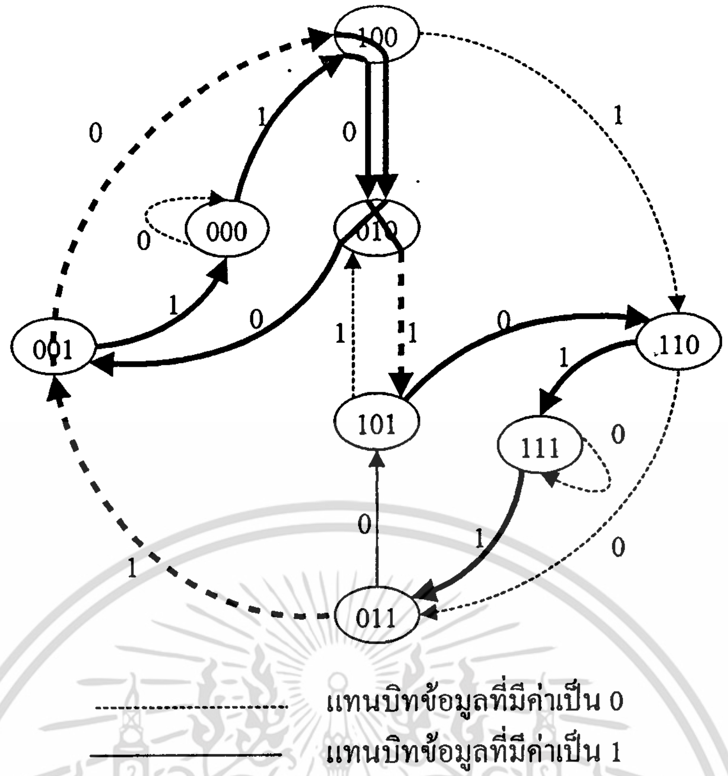
**รูปที่ 2.53** ตัวอย่างขั้นตอนการเข้ารหัส

ตัวอย่างการเข้ารหัสที่ได้อธิบายมานี้สามารถนำมาแสดงในรูปของแผนภาพเทรลลิสจะมีลักษณะดังรูปที่ 2.54 หรือถ้าจะดูจากแผนภาพการเปลี่ยนสถานะก็จะมีลักษณะเป็นดังรูปที่ 2.55




**รูปที่ 2.54** แผนภาพเทรลลิสเมื่อเข้ารหัสวงจรเข้ารหัสย่อยด้วยข้อมูล 11011100 โดยบิทหางมีค่าเท่ากับ 111 ผลลัพธ์ที่ได้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.55** แผนภาพการเปลี่ยนสถานะเมื่อเข้ารหัสด้วยข้อมูล 11011100 โดยบิทหางมีค่าเท่ากับ 111

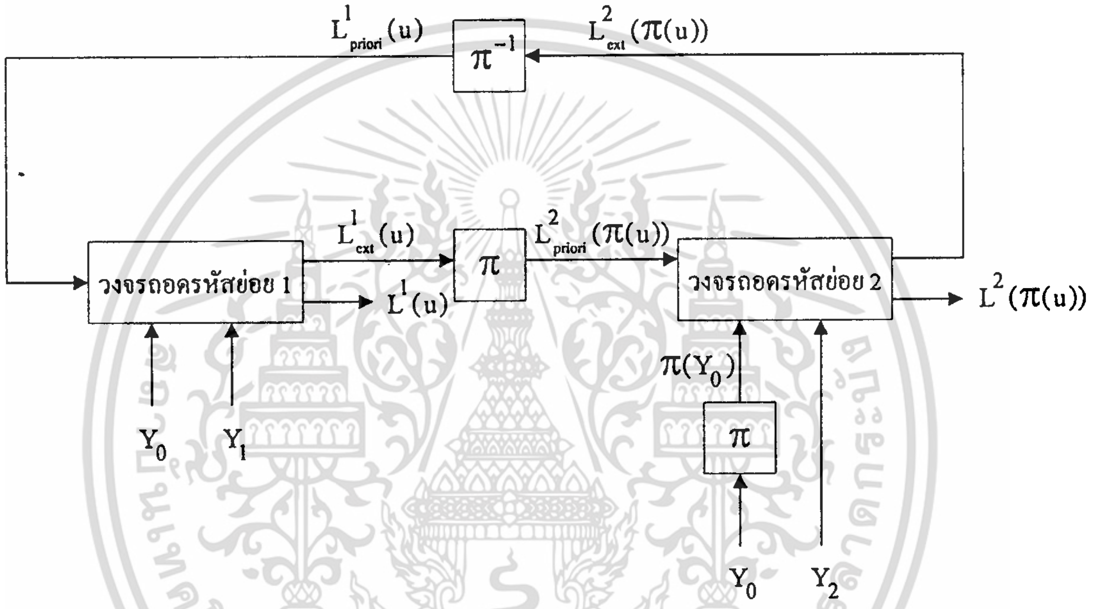
**2.5.3 วงจรถอดรหัสเทอร์โบ**

ดังที่ได้เกริ่นไว้ก่อนหน้าแล้วว่ารหัสเทอร์โบมีคุณสมบัติสำคัญที่แตกต่างจากรหัสแบบอื่นอยู่ 3 ประการ และหนึ่งในนั้นคือกรรมวิธีในการถอดรหัส กล่าวคือในรหัสเทอร์โบกระบวนการถอดรหัสจะเป็นแบบวนซ้ำ (iterative decoding) คือ แทนที่จะมีวงจรถอดรหัสเพียงตัวเดียวและทำการถอดรหัสเพียงรอบเดียว วงจรถอดรหัสเทอร์โบประกอบด้วยวงจรถอดรหัสย่อยมากกว่าหนึ่งตัว ดูรูปที่ 2.56 และลักษณะการทำงานของวงจรถอดรหัสย่อยแต่ละตัวจะทำงานสลับกันไปมา เมื่อตัวหนึ่งอยู่ระหว่างการถอดรหัสอีกตัวหนึ่งจะรอ และผลการถอดรหัสของวงจรหนึ่งจะถูกส่งต่อไปให้ตัวถอดรหัสอีกตัวหนึ่งเพื่อใช้เป็นข้อมูลประกอบในการถอดรหัสในรอบต่อไป การทำงานของวงจรถอดรหัสทั้งสองจะสลับกันไปเป็นรอบ ๆ จนกระทั่งได้ผลลัพธ์ที่ลู่เข้าสู่ค่าที่เหมาะสม ทั้งนี้สังเกตว่าวงจรถอดรหัสย่อยจะมีจำนวนเท่ากับวงจรเข้ารหัสย่อยที่ภาคส่ง โดยจะทำงานสอดคล้องสัมพันธ์กัน รายละเอียดขั้นตอนการทำงานเป็นดังนี้คือ

ขั้นแรกให้นำข้อมูลส่วนที่ผ่านการเข้ารหัสด้วยวงจรเข้ารหัสย่อยที่ 1 $Y_1$ กับข้อมูลที่ไม่ผ่านการเข้ารหัสใด ๆ เลย $Y_0$ มาป้อนเข้าวงจรถอดรหัสย่อยที่ 1 ผลที่ได้จากการถอดรหัสประกอบด้วย 2 ส่วนคือ ส่วนแรกคือค่าอัตราส่วนความน่าเป็นจริง (likelihood ratio) $L^1(u)$ ซึ่งเป็นค่าที่จะนำไปใช้ในการตัดสินใจว่าบิทข้อมูลแต่ละบิทน่าจะมีค่าเป็น 0 หรือ 1 ส่วนที่สองเรียกว่าอัตราส่วนความน่าเป็นจริงของข่าวสารเอกซ์ทรินซิก $L^1_{ext}(u)$ (extrinsic information) ซึ่งเป็นค่าที่จะนำไปป้อนให้วงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถอดรหัสย่อยที่ 2 ในรูปของข่าวสาร priori $L^2_{priori}(\pi(u))$ สังเกตว่าการที่ต้องมีกระบวนการทำอินเทอร์ลีฟ $\pi(u)$ ระหว่างตัวถอดรหัสย่อยตัวที่ 1 และตัวที่ 2 ก็เพื่อให้ลำดับของข่าวสารของบิทได้รับการสลับให้สอดคล้องตรงกันลำดับของบิทที่ใช้ในวงจรถอดรหัสย่อยที่ 2 เช่นเดียวกันกับวงจรถอดรหัสย่อยที่ 2 ด้านเข้าของวงจรถอดรหัสที่ 1 ก็มีการป้อนข่าวสาร $L^1_{priori}(u)$ ด้วยเช่นกัน ซึ่งได้มาจากผลการคำนวณได้อัตราส่วนความน่าเป็นจริงของข่าวสารเอกซ์ทรินซิกจากวงจรถอดรหัสย่อยที่ 2 เพียงแต่ในรอบแรกของการทำงานเรามักจะกำหนดค่าเริ่มต้นของ $L^1_{priori}(u)$ ให้มีค่าเท่ากับ 1 เพื่อแทนความหมายว่าบิทข้อมูล 0 และ 1 มีโอกาสเกิดขึ้นด้วยความน่าจะเป็นที่เท่ากัน




รูปที่ 2.56 โครงสร้างพื้นฐานของวงจรถอดรหัสเทอร์โบ

### 2.5.4 การคำนวณอัตราส่วนความน่าเป็นจริง

สำหรับกรณีของการรับส่งแบบไบนารี คือใช้สัญญลักษณ์ในการส่งที่ต่างกันเพียง 2 รูปแบบคือ 0 และ 1 ค่าอัตราส่วนความน่าเป็นจริง (likelihood ratio) สามารถแสดงได้ดังนี้คือ

$$L(u_n) = \frac{P(u_n = 1|O)}{P(u_n = 0|O)} \tag{2.33}$$

โดย $P(u_n = 1|O)$ คือความน่าจะเป็นที่บิทข้อมูลที่ n มีค่าเป็น 1 เมื่อภาครับได้รับทราบสัญลักษณ์ทั้งหมด O ที่ส่งผ่านช่องสัญญาณ ยกตัวอย่าง เช่น ข้อมูลสัญลักษณ์ที่รับได้ O ของวงจรถอดรหัสย่อยที่ 1 ซึ่งประกอบด้วย $Y_0$ และ $Y_1$ นั่นเอง และสำหรับกรณีของวงจรถอดรหัสย่อยที่ 2 ข้อมูลสัญลักษณ์ที่รับได้ O หมายถึง $\pi(Y_0)$ และ $Y_2$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ $P(u_n = 0|O)$ คือความน่าจะเป็นที่บิทข้อมูลที่ n มีค่าเป็น 0 เมื่อภาครับได้รับทราบสัญญลักษณ์ทั้งหมด O ที่ส่งผ่านช่องสัญญาณ

อาศัยกฎเกณฑ์ของ Baye's เราสามารถเขียนสมการที่ (2.33) ใหม่ได้เป็น

$$L(u_n) = \frac{P(u_n = 1, O)/P(O)}{P(u_n = 0, O)/P(O)} = \frac{P(u_n = 1, O)}{P(u_n = 0, O)} \quad (2.34)$$

สังเกตว่าค่าอัตราส่วนความน่าเป็นจริง $L(u_n)$ มีค่าอยู่ระหว่าง 0 ถึง $\infty$ ค่าที่คำนวณได้นี้นำมาใช้ในการตัดสินใจว่าบิท $u_n$ น่าจะมีค่าอย่างไร สำหรับกรณีที่ $L(u_n) = 0$ หมายความว่าบิทข้อมูลที่ n มีความน่าจะเป็นที่มีค่าเป็น 0 สูงมาก หากพบว่า $L(u_n) = 1$ แสดงว่าบิทที่ n มีค่าเป็น 0 หรือ 1 ด้วยความน่าจะเป็นเท่า ๆ กันนั่นคือตัดสินใจได้ยาก และหาก $L(u_n) = \infty$ มีความหมายว่าบิทที่ n มีโอกาสที่จะมีค่าเป็น 1 สูงมาก กระบวนการตัดสินใจสามารถเขียนในรูปคณิตศาสตร์ได้เป็น

$u_n = 0$ ถ้า $L(u_n) < 1$

$u_n = 1$ ถ้า $L(u_n) > 1$

ค่าอัตราส่วนความน่าเป็นจริงตามความสัมพันธ์ในสมการที่ (2) สามารถพรรณนาในอีกลักษณะหนึ่งได้เป็น

$$L(u_n) = \frac{\sum_{(s',s):u_n=1} P(S_{n-1} = s'S_n = s, O)}{\sum_{(s',s):u_n=0} P(S_{n-1} = s', S_n = s, O)} \quad (2.35)$$

เมื่อ $(s',s):u_n = 1$ หมายถึงกรณีหรือเหตุการณ์ที่วงจรเข้ารหัสจะเปลี่ยนสถานะจาก s' ไปเป็นสถานะ s เมื่อมีการป้อนบิท $u_n = 1$ และ $(s',s) : u_n = 0$ หมายถึงกรณีหรือเหตุการณ์ที่วงจรเข้ารหัสจะเปลี่ยนสถานะจาก s' ไปเป็นสถานะ s เมื่อมีการป้อนบิท $u_n = 0$

รูปแบบการเปลี่ยนสถานะเหล่านี้ขึ้นกับโครงสร้างของวงจรเข้ารหัสและสังเกตว่าคุณลักษณะของการเปลี่ยนสถานะไม่ขึ้นกับเวลา n นั่นคือการเปลี่ยนสถานะของวงจรเข้ารหัสขึ้นอยู่กับบิทข้อมูลที่ป้อนเข้ามาเท่านั้นซึ่งมีรูปแบบเฉพาะตายตัว ดังนั้นเราจึงกล่าวว่าวงจรเข้ารหัสมีคุณสมบัติไทม์อินแวเรียนต์มาคอฟเชน (time invariant Markov chain)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการคำนวณหาอัตราส่วนความน่าเป็นจริงมักจะมีการแปลงรูปความสัมพันธ์หรือจัดให้อยู่ในรูปแบบที่คำนวณได้โดยง่ายและสามารถใช้กรรมวิธีรีเคอร์ซีฟ (recursive) มาช่วยในการคำนวณ ได้ดังต่อไปนี้

$$L(u_n) = \frac{\sum_{(s',s):u_n=1} \alpha_{n-1}(s').\gamma_n(s',s).\beta_n(s)}{\sum_{(s',s):u_n=0} \alpha_{n-1}(s').\gamma_n(s'.s).\beta_n(s)} \tag{2.36}$$

โดยที่

$$\alpha_n(s) = P(S_n = s, O_1^n)$$
$$\beta_n(s) = P(O_{n+1}^N \mid S_n = s; O_1^n)$$
$$\gamma_n(s',s) = P(S_n = s, O_n \mid S_{n-1} = s', O_1^{n-1})$$

และ

$$O = (O_1, O_2, O_3, ..., O_N)$$
$$O_1^n = (O_1, O_2, O_3, ..., O_n)$$
$$O_{n+1}^N = (O_{n+1}, O_{n+2}, O_{n+3}, ..., O_N)$$

สำหรับส่วนต่อไปนี้จะกล่าวถึงการพิสูจน์การแสดงสมการที่ (2.35) ด้วยสมการที่ (2.36)

$$\begin{aligned} P(S_{n-1} = s', S_n = s, O) &= P(S_{n-1} = s', O_1^{n-1}) P(S_n = s, O_n^N \mid S_{n-1} = s', O_1^{n-1}) \qquad (2.37) \\ &= P(S_{n-1} = s', O_1^{n-1}) P(S_n = s, O_n \mid S_{n-1} = s', O_1^{n-1}) P(O_{n+1}^N \mid S_{n-1} = s', S_n = s, O_1^n) \\ &= P(S_{n-1} = s', O_1^{n-1}) P(S_n = s, O_n \mid S_{n-1} = s', O_1^{n-1}) P(O_{n+1}^N \mid S_n = s, O_1^n) \\ &= \alpha_{n-1}(s') \gamma_n(s',s) \beta_n(s) \end{aligned}$$

การเปลี่ยนรูปสมการจากบรรทัดแรกและบรรทัดที่สองนั้นอาศัยกฎเกณฑ์ของ Baye's ส่วนการที่สามารถลดรูปพจน์ที่สามของบรรทัดที่สามได้เพราะอาศัยคุณสมบัติมาคอฟเชนของวงจรเข้ารหัส กล่าวคือถ้าเราสามารถทราบสถานะของวงจรเข้ารหัสว่า $S_n = s$ ที่เวลา n แล้ว เหตุการณ์ $S_{n-1} = s'$ ที่เกิดขึ้นที่เวลา n-1 ไม่ส่งผลอีกต่อไป

โดยทั่วไปในการคำนวณค่า $\alpha_n(s)$ มักจะอาศัยวิธีการแบบรีเคอร์ซีฟ (recursive) ดังนี้

$$\begin{aligned} \alpha_n(s) &= P(S_n = s, O_1^n) \\ &= \sum_{s' \in \zeta} P(S_{n-1} = s', S_n = s, O_1^n) && \text{อาศัย marginal probability} \\ &= \sum_{s' \in \zeta} P(s_{n-1} = s', O_1^{n-1}) P(S_n = s, O_n \mid S_{n-1} = s', O_1^{n-1}) && \text{อาศัยกฎเกณฑ์ของ Baye's} \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$= \sum_{s'\in\zeta} \alpha_{n-1}(s').\gamma_n(s',s) \tag{2.38}$$

เช่นเดียวกันกับ $\alpha_n(s)$ ในการคำนวณค่า $\beta_n(s)$ มักจะอาศัยวิธีการแบบรีเคอร์ซีฟ (revursive) ดังนี้

$$\beta_n(s) = P(O_{n+1}^N | S_n = s, O_1^n)$$

$$= \sum_{s'\in\zeta} P(S_{n+1} = s', O_{n+1}^N | S_n = s, O_1^n)$$ อาศัย marginal probability

$$= \sum_{s'\in\zeta} P(O_{n+2}^N | S_n = s, S_{n+1} = s', O_1^{n+1}) P(S_{n+1} = s', O_{n+1} | S_n = s, O_1^n)$$ อาศัยกฎเกณฑ์ของ Baye's

$$= \sum_{s'\in\zeta} P(O_{n+2}^N | S_{n+1} = s', O_1^{n+1}) P(S_{n+1} = s', O_{n+1} | S_n = s, O_1^n)$$ อาศัยคุณสมบัติมาคอฟเชน

$$= \sum_{s'\in\zeta} \beta_{n+1}(s').\gamma_{n+1}(s,s') \tag{2.39}$$

โดยเงื่อนไขขอบเขตของสมการความสัมพันธ์ของ $\alpha_n(s)$ และ $\beta_n(s)$ เป็นดังนี้คือ

$\alpha_0(0) = 1$

$\alpha_0(s) = 0$ เมื่อ $s \neq 0$

$\beta_N(0) = 1$

$\beta_N(s) = 0$ เมื่อ $s \neq 0$

เงื่อนไขขอบเขตเหล่านี้พิจารณาจากกลไกการทำงานของวงจรเข้ารหัสที่กำหนดว่าให้ชิฟต์รีจิสเตอร์ทุกตัวเริ่มต้นที่สถานะเป็นศูนย์เสมอ และให้จบสิ้นการเข้ารหัสที่สถานะเป็นศูนย์ด้วย

สำหรับกรณีเฉพาะที่ช่องสัญญาณไม่ก่อให้เกิดการรบกวนกันระหว่างสัญลักษณ์พจน์ $\gamma_n(s',s)$ และ $\beta_n(s)$ สามารถลดรูปลงได้เป็นดังนี้

$$\beta_n(s) = P(O_{n+1}^N | S_n = s) \tag{2.40}$$

$$\gamma_n(s',s) = P(S_n = s, O_n | S_{n-1} = s') \tag{2.41}$$

$$= P(S_n = s, | S_{n-1} = s') P(O_n | S_{n-1} = s', S_n = s)$$

ที่ผ่านมาข้างต้นเราอธิบายถึงสมการสำหรับการถอดรหัสเพื่อหาค่าอัตราส่วนความน่าเป็นจริง $L(u_n)$ สำหรับวงจรถอดรหัสใด ๆ แต่เนื่องจากกรรมวิธีการถอดรหัสเทอร์โบเป็นโครงสร้างการถอดรหัสแบบวนซ้ำ คืออาศัยวงจรถอดรหัสย่อยมากกว่าหนึ่งตัว ดังรูปที่ 2.40 การวิเคราะห์ที่ผ่านมาจึงเป็นขั้นตอนการทำงานที่เกิดขึ้นในวงจรถอดรหัสย่อยแต่ละตัวเท่านั้น เช่นในกรณีของวงจรถอดรหัสย่อยที่ 1 ผลที่ได้จากวิเคราะห์ข้างต้นก็คือค่า $L^1(u)$ สังเกตว่ามีข่าวสารอีกส่วนหนึ่งที่ยังไม่ได้แสดงการคำนวณคือ $L_{ext}^1(u)$ ซึ่งเป็นส่วนของข่าวสารที่เรียกว่า ข่าวสารเอกซ์ทรินซิก (extrinsic information) ข่าวสารส่วนนี้จะถูกนำไปป้อนให้วงจรถอดรหัสย่อยที่ 2 ในรูปของ $L_{priori}^2(\pi(u))$ และเราเรียกข่าวสารส่วนนี้ว่า priori information ฉะนั้นในส่วนต่อไปนี้จะกล่าวถึงการดึงข่าวสารส่วนนี้สำหรับวงจรถอดรหัสแต่ละตัวโดยจะพิจารณากรณีของวงจรถอดรหัสย่อยที่ 1 เป็นตัวอย่าง เนื่องจากที่ผ่านมาการอธิบายสมการสำหรับการถอดรหัสไม่มีพารามิเตอร์ใดที่ระบุว่าใช้วงจร

เอกสารนี้เป็นเอกสารที่มีลิขสิทธิ์... ไม่ใช่เพื่อวัตถุประสงค์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถอดรหัสย่อยตัวใดอยู่ ฉะนั้นในส่วนต่อไปนี้จึงต้องมีการเพิ่มพารามิเตอร์กำกับหมายเลขของวงจรถอดรหัสย่อยด้วย เช่น $\gamma_n(s',s)$ จะเขียนใหม่ได้เป็น $\gamma_{1,n}(s',s)$ เป็นต้น สำหรับรูปสัญญาณที่รับได้สำหรับกรณีวงจรถอดรหัสย่อยที่ 1 ได้แก่

$$O=(Y_0,Y_1)$$

$$O_n=(Y_{0,n},Y_{1,n})$$

$$\gamma_{1,n}(s',s)=P(S_{1,n}=s,O_n\,|\,S_{1,n-1}=s')$$
$$=P(S_{1,n}=s,|\,S_{1,n-1}=s')P(O_n\,|\,S_{1,n-1}=s',S_{1,n}=s)$$

สังเกตว่า $\gamma_{1,n}(s',s)=0$ ถ้าวงจรเข้ารหัสไม่สามารถเปลี่ยนจากสถานะ s' ไปเป็นสถานะ s และสำหรับกรณีที่การเปลี่ยนสถานะมีความเป็นไปได้ $\gamma_{1,n}(s',s)$ จะมีค่าเท่ากับ

$$\gamma_{1,n}(s',s)=P(u_n)P(Y_{0,n}\,|\,X_{0,n})_{X_{0,n}=\text{data}\quad\text{of}(s_{1,n-1}=s',S_{1,n}=s)}$$
$$P(Y_{1,n}\,|\,X_{1,n})_{x_{1,n}=\text{parity}\quad\text{of}(S_{1,n-1}=s',S_{1,n}=s)}$$

จากสมการ 2.36 ค่าความน่าเป็นจริงของวงจรถอดรหัสย่อยที่ 1 แสดงได้เป็น

$$L^1(u_n)=\frac{P(u_n=1\,|\,O,\text{วงจรเข้ารหัสย่อย1})}{P(u_n=0\,|\,O,\text{วงจรเข้ารหัสย่อย1})}$$

$$=\frac{\sum_{(S',s):u_n=1}\alpha_{1,n-1}(s').\gamma_{1,n}(s',s).\beta_{1,n}(s)}{\sum_{(S',s):u_n=0}\alpha_{1,n-1}(s').\gamma_{1,n}(s',s).\beta_{1,n}(s)}$$

$$=\frac{P(u_n=1)\;P(Y_{0,n}\,|\,X_{0,n}=1)\sum_{(S',s):u_n=1}[\alpha_{1,n-1}(s').P(Y_{1,n}\,|\,X_{1,n}).\beta_{1,n}(s)]_{X_{1,n}=\text{parity}\quad\text{of}(S_{1,n-1}=s',S_{1,n}=s)}}{P(u_n=0)\;P(Y_{0,n}\,|\,X_{0,n}=0)\sum_{(S',s):u_n=0}[\alpha_{1,n-1}(s').P(Y_{1,n}\,|\,X_{1,n}).\beta_{1,n}(s)]_{X_{1,n}=\text{parity}\quad\text{of}(S_{1,n-1}=s',S_{1,n}=s)}}$$

$$=L^1_{priori}(u_n)L^1_{channel}(u_n)L^1_{ext}(u_n)$$

$$=L^1_{int}(u_n)L^1_{ext}(u_n)\qquad(2.42)$$

ในกรณีที่เป็นวงจรถอดรหัสย่อยทำงานรอบแรก ข่าวสาร $L^1_{priori}(u_n)=\frac{P(u_n=1)}{P(u_n=0)}$ มักจะมีค่าเท่ากับ $\frac{1/2}{1/2}=1$ ซึ่งหมายความว่าความน่าจะเป็นที่บิทข้อมูลมีค่าเป็นศูนย์หรือหนึ่งจะมีค่าเท่า ๆ กัน แต่สำหรับการทำงานในรอบต่อ ๆ ไป วงจรถอดรหัสย่อยจะใช้ข่าวสาร $\frac{P(u_n=1)}{P(u_n=0)}$ จากข่าวสารที่คำนวณได้จากวงจรถอดรหัสย่อยตัวที่ 2 ดังสมการต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$P(u_n = 1) = \frac{L_{ext}^2(u_n)}{1 + L_{ext}^2(u_n)} \tag{2.43}$$

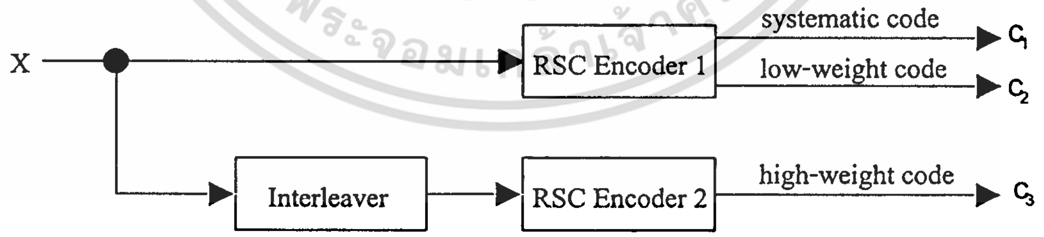
$$P(u_n = 0) = \frac{1}{1 + L_{ext}^2(u_n)}$$

ข่าวสารในพจน์ $L_{channel}^1(u_n) = \frac{P(Y_{0,n} \mid X_{0,n} = 1)}{P(Y_{0,n} \mid X_{0,n} = 0)}$ คือข่าวสารที่ขึ้นโดยตรงกับคุณลักษณะช่องสัญญาณ (channel information) ส่วนข่าวสาร $L_{ext}^1(u_n)$ คือข่าวสารขึ้นกับโครงสร้างของการเข้ารหัส สังเกตว่าข่าวสารที่ส่งผ่านระหว่างวงจรถอดรหัสย่อยจะมีเฉพาะส่วนของ $L_{ext}^1(u_n)$ เท่านั้น การแลกเปลี่ยนเฉพาะข่าวสารส่วนนี้ระหว่างวงจรถอดรหัสย่อยนับว่าเป็นกุญแจหนึ่งของความสำเร็จของรหัสเทอร์โบ

การวิเคราะห์ข่าวสารที่ได้จากวงจรถอดรหัสย่อยที่ 2 ก็สามารถทำได้ด้วยวิธีที่คล้ายคลึงกับการวิเคราะห์ข่าวสารของวงจรถอดรหัสย่อยที่ 1 แต่ลำดับของบิทข้อมูล $u_n$ จะต้องถูกสลับลำดับด้วยวงจรอินเทอร์ลีฟเวอร์

## 2.6 การออกแบบอินเตอร์ลีฟ

สำหรับระบบเทอร์โบโค้ด ได้นำเอาการทำอินเตอร์ลีฟเวอร์เข้ามาใช้ระหว่างองค์ประกอบภายใน ชุดเข้ารหัสแบบ RSC ทั้ง 2 ชุด โดยที่กระบวนการของการทำงานของอินเตอร์ลีฟเวอร์นั้นจะจัดการตำแหน่งของข้อมูล ให้มีการจัดเรียงตำแหน่ง แตกต่างไปจากชุดข้อมูลเดิม ซึ่งจะทำให้ค่าของน้ำหนักของ โค้ดเวอร์ดนั้นมีค่าเพิ่มมากขึ้น ดังแสดงในรูปที่ 2.57



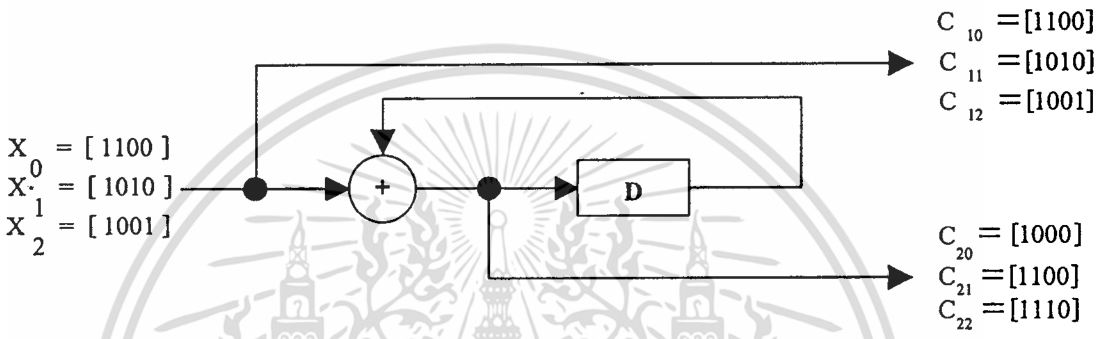

**รูปที่ 2.57 การทำอินเตอร์ลีฟ ประกอบด้วยชุดเข้ารหัสแบบ RSC จำนวน 2 ชุด**

จากรูปที่ 2.57 ค่าของลำดับของข้อมูลทางด้านขาเข้า ของชุดลำดับข้อมูล x ในชุดเข้ารหัสแบบคอนโวลูชั่นที่มีการป้อนกลับ และค่าน้ำหนักของข้อมูลเป็น Low-Weight code ได้เอาท์พุทออกมา คือ C2 สำหรับชุดเข้ารหัสชุดที่ 1 และถ้าเรานำเอาข้อมูลทางด้านขาเข้าตัวเดียวกัน เข้ามายังชุดเข้ารหัส RSC ตัวที่ 2 จะให้ผลออกมาเป็น Low-Weight code เช่นเดียวกัน แต่เราไม่ต้องการได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Low-Weight code อีก เราจึงมีวิธีการ โดยเราต้องทำการจัดการข้อมูลทางด้านอินพุท ให้ข้อมูลทางด้านอินพุทนั้นแตกต่างกัน โดยนำเอาวิธีการทำอินเตอร์ลีฟเข้ามาจัดการข้อมูลก่อนที่จะส่งเข้ามายังชุดเข้ารหัสชุดที่ 2 เพื่อที่จะได้รหัส High-Weight code ออกมา C3

ดังนั้น ระบบการเข้ารหัสแบบเทอร์โบโค้ด จะประกอบด้วยโค้ด ที่มีค่าของทั้ง Low และ High ทั้งสอง สรุปได้ว่า Low-weight Code จะได้มาจากชุดเข้ารหัสชุดที่ 1 ส่วน High-Weight code จะได้มาจากชุดเข้ารหัสชุดที่ 2 ในรูปที่ 2.58 แสดงตัวอย่างของรหัสที่ได้จาก ระบบเข้ารหัสแบบเทอร์โบโค้ด




**รูปที่ 2.58** ตัวอย่างของรหัสที่ได้จากชุดเข้ารหัสเทอร์โบโค้ด

จากรูปที่ 2.58 สัญญาณที่เข้ามาทางด้านอินพุท คือ Xi จะได้เอาท์พุทออกมา C1I และ C2I ตามลำดับ ข้อมูลอินพุท X1 และ X2 จะมีรูปแบบที่แตกต่างกันกับ X0 ซึ่งในตารางที่ 2.9 ได้แสดงรายละเอียดผลลัพธ์ คือ โค้ดเวอร์ด และ Weight

**ตารางที่ 2.9** ชุดลำดับข้อมูลทางด้านอินพุท และเอาท์พุท ของชุดเข้ารหัส

| | Input Sequence $X_I$ | Output Sequence $C_{1i}$ | Output Sequence $C_{2i}$ | Codeword Weight i |
|---|---|---|---|---|
| i = 0 | 1100 | 1100 | 1000 | 3 |
| i = 1 | 1010 | 1010 | 1100 | 4 |
| i = 2 | 1001 | 1001 | 1110 | 5 |

ในตารางนั้นแสดงให้เห็นว่าโค้ดเวอร์ด และ Weight นั้นสามารถเพิ่มขึ้นได้ใช้การทำอินเตอร์ลีฟเวอร์

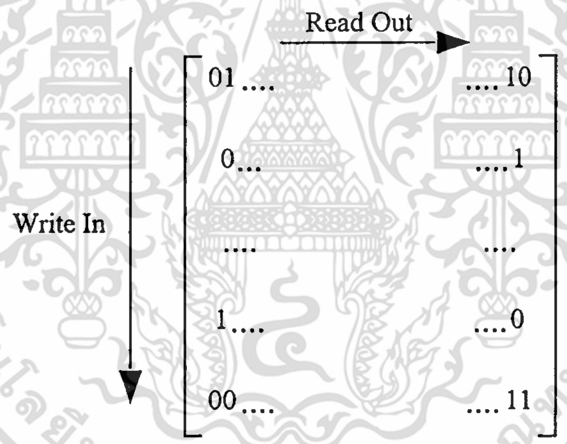
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลกระทบของอินเตอร์ลีฟเวอร์นั้นจะมีผลต่อสมรรถนะของระบบการเข้ารหัสเทอร์โบโค้ด เพราะว่ามีผลกระทบโดยตรงต่อคุณสมบัติของระยะของการเข้ารหัส ซึ่งแสดงรายละเอียดในเอกสารอ้างอิง[1].

เมื่อทำการหลีกเลี่ยงไม่ให้เกิดค่า Low-Weight ค่าอัตราการผิดพลาดบิทข้อมูล(BER) ของระบบเข้ารหัสแบบเทอร์โบโค้ดสามารถที่จะปรับปรุงให้ดีขึ้นได้ ดังนั้นจึงได้มีการค้นคว้าวิธีออกแบบการทำอินเตอร์ลีฟเกิดขึ้น ซึ่งรายละเอียดต่อจากนี้ไป จะกล่าวถึงรูปแบบในการทำอินเตอร์ลีฟแบบต่างๆ โดยส่วนใหญ่นั้น จะนิยมใช้ในระบบเข้ารหัสเทอร์โค้ด

### 2.6.1 การทำอินเตอร์ลีฟแบบ Block Interleave

การทำอินเตอร์ลีฟแบบบล๊อค (Block Interleaver) นั้นเป็นที่นิยมมากในระบบการสื่อสารในปัจจุบันนี้ ซึ่งวิธีการของมันนั้นจะทำการจัดการจัดตำแหน่งของข้อมูลโดย ให้ข้อมูลออกทางด้านเอาท์พุท เป็นข้อมูลที่ได้จากการอ่านข้อมูลในแนวตั้งในทิศทางจากบนลงล่าง ซึ่งแสดงรายละเอียดในรูปที่ 2.59




รูปที่ 2.59 การทำอินเตอร์ลีฟเวอร์แบบบล๊อค Block Interleave

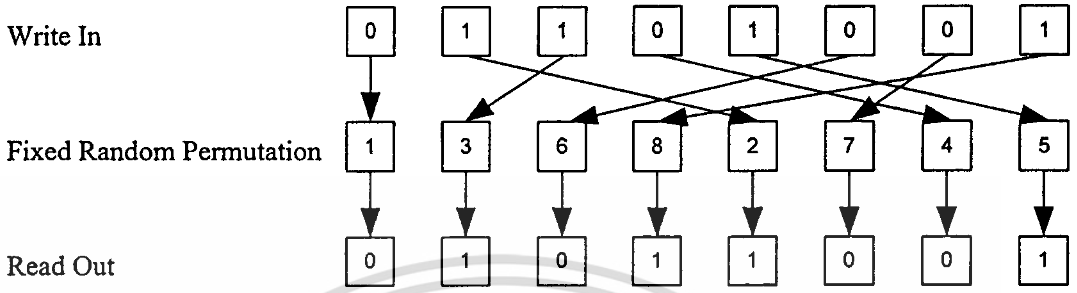
จากรูปที่ 2.59 นั้น อินพุทที่เข้ามาคือ[0 1 ... 1 0 0 ... 1 ...1 ... 0 0 0 ... 1 1] และได้ข้อมูลใหม่ออกมาทางด้านเอ้าพุทคือ [0 0 ... 1 0 1 ... 0...1...1 0 1... 0 1]

### 2.6.2 การทำอินเตอร์ลีฟเวอร์แบบ แรนดอม Random(Pseudo-Random) Interleaver

แรนดอมอินเตอร์ลีฟเวอร์นั้น มีวิธีการในการจัดตำแหน่งของข้อมูลโดยกำหนดค่าตำแหน่งการวางข้อมูลโดยทำการสุ่มตำแหน่งแบบแรนดอม แล้วเก็บค่าตำแหน่งดังกล่าวนั้นในหน่วยความจำ เพื่อใช้ในขบวนการย้อนกลับเอาข้อมูลกลับมาทางด้านปลายทาง ตำแหน่งที่ได้จาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การสุ่มนั้น จะนำมาเป็นตำแหน่งที่ใช้ในการอ้างอิงในการจัดวางตำแหน่ง ในรูปนั้นเราจะยกตัวอย่าง โดยสมมุติให้ความยาวของชุดข้อมูลมีค่าเท่ากับ Lซึ่งแสดงในรูปที่ 2.60




รูปที่ 2.60 การทำอินเตอร์ลีฟเวอร์แบบ Random โดยความมีความยาวของชุดข้อมูลเท่ากับ 8

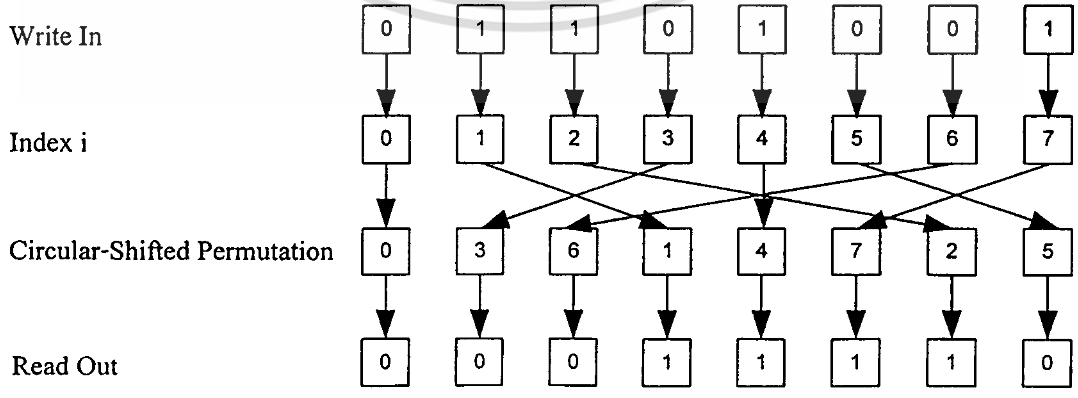
ในรูปที่ 2.60 นั้นผลของการทำอินเตอร์ลีฟเวอร์จะได้ชุดข้อมูลออกมาคือ [0 1 0 1 1 0 0 1] เมื่อชุดข้อมูลทางด้านอินพุทคือ [0 1 1 0 1 0 0 1] โดยถูกการจัดวางตำแหน่งดังรูปที่ 2.60

### 2.6.3 เซอร์คูล่าร์-ชีพ อินเตอร์ลีฟ

การเปลี่ยนแปลง ค่าของ p ในรูปแบบของ เซอร์คูล่า-ชีฟ อินเตอร์ลีฟ นั้นสามารถกำหนดได้โดย

$$p(i) = (ai + s) \bmod L$$

โดยที่ a< L,:ซึ่ง a คือค่า relatively prime กับค่า L ซึ่ง ค่า s<L เมื่อ I คือค่าอินเด็กส์ และ a คือค่าของลำดับขนาดของลำดับขั้น และ s คือค่าของ ออฟเซ็ท ซึ่งแสดงรายละเอียดในอ้างอิง [2] .ในรูปที่ 2.61 นั้นแสดงการทำอินเตอร์ลีฟแบบเซอร์คูล่า-ชีฟ ซึ่งมีค่า L=8,a=3 และ s=0




**รูปที่ 2.61** การทำอินเตอร์ลีฟแบบเซอร์คูล่าร์ เมื่อ L=8, a=3 และ s=0.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากในรูปที่ 2.61 การทำอินเตอร์ลีฟนั้นจะทำการส่งข้อมูลออกเป็น [0 1 1 0 1 0 0 1] ซึ่งค่าที่ป้อนเข้ามาทางด้านอินพุทคือ [0 0 0 1 1 1 1 0] ดังนั้น เราสามารถที่จะทำการปรับค่าโดยทำการจัดแบ่งออกเป็น 3 หรือ 5 ซึ่งชนิดของการทำอินเตอร์ลีฟเวอร์นั้นแสดงให้เห็นถึง ผลลัพธ์ที่ได้ออกมานั้นออกมาดีมาก ในการเปลี่ยนแปลง ค่าน้ำหนักทางด้านอินพุทเป็น 2 ซึ่งชุดของลำดับข้อมูลนั้นจะเป็น โค้ดเวอร์ดที่มีค่าน้ำหนักต่ำ เปลี่ยนแปลงไป เป็นค่าน้ำหนักเท่ากับ 2 ของลำดับของชุดข้อมูล จะได้ค่าของโค้ดเวอร์ดที่มีค่าสูง อย่างไรก็ตาม โดยปรกตินั้น จะมีค่า (3 หรือ 5 ซึ่งเราจะทำการเปลี่ยนแปลงค่าของบิทโดยทำการแบ่งออกมาในรูปที่ 5) ซึ่งโดยธรรมชาตินั้นชนิดของการทำอินเตอร์ลีฟเวอร์ชนิดนี้ จะมีความยุ่งยาก ในการที่จะจัดการค่าของโค้ดเวอร์ดให้มีค่าน้ำหนักสูง ( ค่า weight > 2) ค่าของลำดับของชุดข้อมูล ที่ได้ออกมามีค่าโค้ดเวอร์ดต่ำนั้น และค่าของโค้ดเวอร์ดที่มีค่าสูงนั้นดูได้จากเอกสารอ้างอิง[3].

### 2.6.4 การทำอินเตอร์ลีฟเวอร์แบบ เซมิแรนดอม

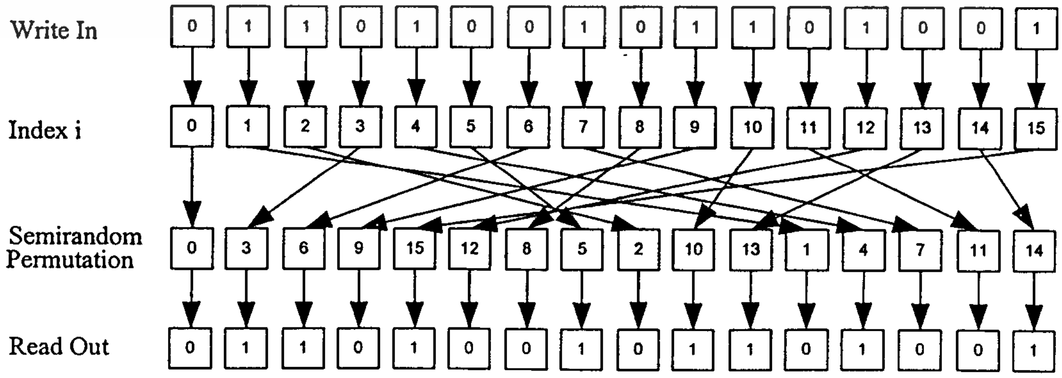
การทำอินเตอร์ลีฟเวอร์แบบ เซมิแรนดอม หรือ เรียกว่าแบบกึ่งแรนดอม นั้นจะเป็นการทำอินเตอร์ลีฟแบบผสมผสานระหว่าง การทำอินเตอร์ลีฟเวอร์แบบแรนดอม และการทำอินเตอร์ลีฟเวอร์แบบ “designed” อินเตอร์ลีฟเวอร์ ดังเช่นการทำอินเตอร์ลีฟเวอร์แบบบล็อค และการทำอินเตอร์ลีฟเวอร์แบบเซอร์คูล่า-ซีฟ อินเตอร์ลีฟเวอร์ [4] ในการสับเปลี่ยนตำแหน่งโดยอัลกอริทึมแบบเซมิแรนดอมนั้น ได้อธิบายในเอกสารอ้างอิง[5] ดังมีขั้นตอนดังนี้

ขั้นตอนที่ 1 ทำการเลือกค่าอินเด็กแบบแรนดอม $i \in [0, L-1]$

ขั้นตอนที่ 2 ทำการเลือกค่าของ S ที่มีค่าเป็นบวก $S < \sqrt{\frac{L}{2}}$

ขั้นตอนที่ 3 ทำการเปรียบเทียบ I กับค่า S ที่ผ่านมา ซึ่งค่าแต่ละค่าของ S นั้นจะทำการเปรียบเทียบค่า I ที่ปรากฏ โดยค่า I จะมีค่าอยู่ในช่วงของ $\pm S$ ถ้า I ภายในนั้นอยู่ในช่วงที่กำหนดให้กลับไปยังขั้นตอนที่ 1 แต่ถ้าปรากฏค่า I ค่าอื่นๆ ออกมาให้เก็บค่า I นั้นเอาไว้

ขั้นตอนที่ 4 กลับไปยังขั้นตอนที่ 1 จนกระทั่งค่า L นั้นถูกกำหนดหมดทุกตัว




**รูปที่ 2.62 การทำอินเตอร์ลีฟแบบเซมิแรนดอม โดยมีค่า L=16 และ S=2**

เอกสารนี้เป็นเอกสารที่สงวนสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป 2.62 การทำอินเตอร์ลีฟนั้นค่าที่ได้ทางด้านเอาท์พุทคือ [ 0 1 1 0 1 0 0 1 0 1 10 1 0 0 1] เมื่อทำการป้อนอินพุทเข้ามาเป็น [ 0 0 0 1 1 1 0 0 1 1 0 1 1 1 0 0] การทำอินเตอร์ลีฟเวอร์แบบเซมิแรนดอมนั้นไม่สามารถให้การรับประกันได้ว่าสามารถที่ จะจัดการได้สำเร็จ ดังที่ถูกกล่าวไว้ในอ้างอิง[7]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3
# วิธีการดำเนินงาน

ในบทนี้เราจะจำลองการทำงานระบบ DS-CDMA มาตรฐาน IS-95 ในทิศทางจากโทรศัพท์เคลื่อนที่ไปยังสถานีฐาน โดยจะทำการแบ่งการออกแบบออกเป็นส่วน แต่ละส่วนแสดงขั้นตอนการจำลองการทำงานระบบ โดยอธิบายแสดงขั้นตอนการทำงานของระบบ สุดท้ายจะนำระบบที่ได้ทำการออกแบบแต่ละส่วนมารวมกันเป็นระบบ DS-CDMA มาตรฐาน IS-95 (แบบที่ใช้ชุดเข้ารหัสข้อมูลแบบเดิม ซึ่งเป็นแบบคอนโวลูชั่น และแบบที่ใช้การเข้ารหัสข้อมูลแบบเทอร์โบ ซึ่งเป็นแบบที่ผู้วิจัยได้นำเสนอ) ซึ่งจะใช้ในการวิเคราะห์ประสิทธิภาพของระบบในบทถัดไป

## 3.1 ชุดสร้างข้อมูลไบนารี่

การออกแบบชุดสร้างข้อมูลไบนารี่ ทำการออกแบบโดยสร้างข้อมูลแบบสุ่มโดยที่ในแต่ละเฟรมของข้อมูลจะมีค่าข้อมูลแบบสุ่มที่ไม่ซ้ำกัน ซึ่งสามารถยกตัวอย่างการจำลองการทำงานโดยเขียนโปรแกรมภาษา C ได้ดังนี้

ส่วนของโปรแกรมสร้างข้อมูลไบนารี่

```
Unseigned time t;
Srand(t);
//--{กำหนดค่าเริ่มต้นของข้อมูลแต่ละชุดให้มีค่าเริ่มต้นที่แตกต่างกัน โดยใช้ค่าเวลานำมาเป็นค่าอ้างอิงในการสร้างข้อมูลแบบสุ่ม -----//
For(i=0;i<data_frame;i++)
{
*(data+i)=rand() % 2;
//---สร้างข้อมูลที่มีค่าแบบสุ่มโดยมีการเปลี่ยนแปลงค่าได้ 2 ค่า {'0' และ '1'}
}
```

ผลการ Run โปรแกรม (กำหนดให้ data frame =10)

**1 0 0 1 0 1 0 1 0**

ผลของการ Run โปรแกรมจะได้ข้อมูลที่มีลักษณะแบบ Random ที่มีการเปลี่ยนแปลงอยู่ 2 ระดับคือ "0" และ "1"

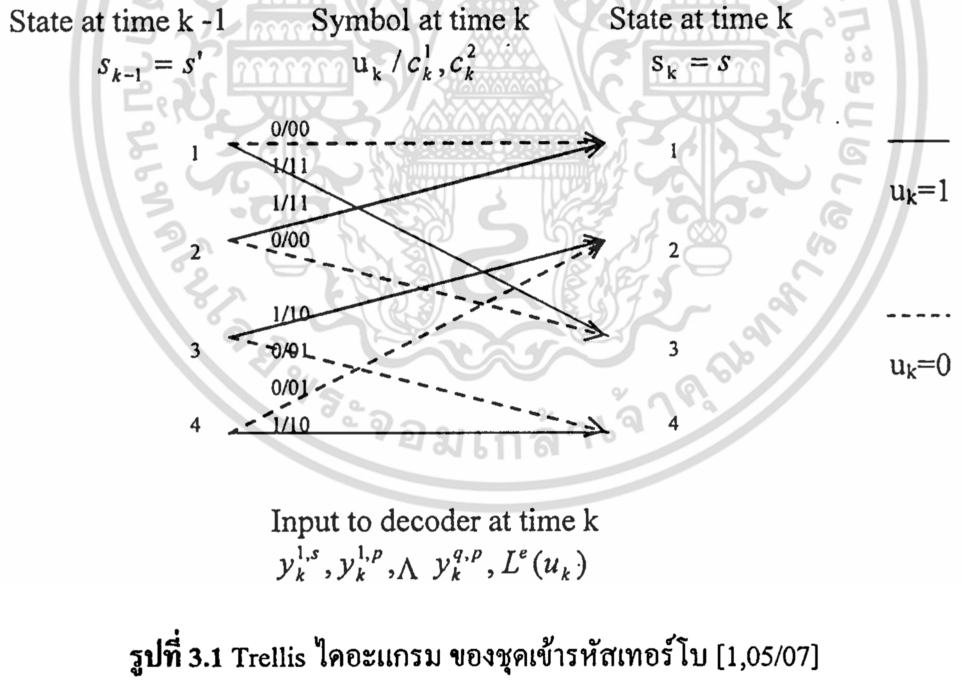
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.2 ชุดเข้ารหัสเทอร์โบ และอัลกอริทึมแบบ MAP

ในวิทยานิพนธ์ฉบับนี้ อัลกอริทึมในการถอดรหัสของชุดเข้ารหัสเทอร์โบจะใช้ MAP ในการถอดรหัส เนื่องจากต้องการความถูกต้องมากที่สุด ฉะนั้นผู้วิจัยจึงหลีกเลี่ยงการประมาณค่า โดยใช้อัลกอริทึมแบบ Log-MAP และ MAX-Log-MAP ซึ่งมีข้อดีในส่วนของการประหยัดจำนวนหน่วยความจำ ซึ่งเป็นประโยชน์ในการจำลองการทำงานโดยโปรแกรมคอมพิวเตอร์ แต่ผลที่ได้นั้นให้ประสิทธิภาพออกมาด้อยกว่า อัลกอริทึมแบบ MAP รายละเอียดในส่วนหัวข้อนี้ จะแบ่งออกเป็นสองส่วน คือ ส่วนแรกจะกล่าวถึงวิธีการในการเข้ารหัสแบบเทอร์โบ และส่วนที่สองจะกล่าวถึงอัลกอริทึมในการถอดรหัสของเทอร์โบ ดังจะกล่าวดังต่อไปนี้

### 3.2.1 ชุดเข้ารหัสเทอร์โบ

ในส่วนของชุดเข้ารหัสแบบเทอร์โบ นั้นจะประกอบด้วย ชุดเข้ารหัสแบบคอนโวลูชั่นแบบรีเคอร์ซีฟ จำนวน 2 ชุด ซึ่งขั้นตอนการออกแบบจะคล้ายกับชุดเข้ารหัสแบบคอนโวลูชั่น ดังหัวข้อที่ผ่านมาข้างต้น รูปต่อไปนี้แสดง Trellis ไดอะแกรมของชุดเข้ารหัสแบบเทอร์โบ ซึ่งจะยกตัวอย่างขนาดของเจนเนอร์เรเตอร์ขนาดเล็ก คือ 1,5/7 ซึ่งจะแสดง รูปไดอะแกรม Trellis ได้ดังนี้



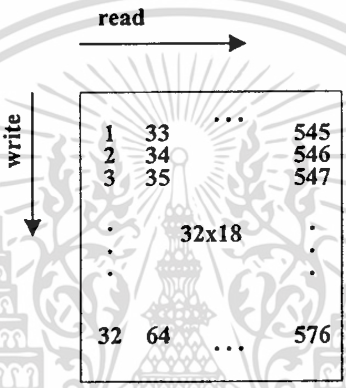

**รูปที่ 3.1** Trellis ไดอะแกรม ของชุดเข้ารหัสเทอร์โบ [1,05/07]

หลังจากที่ได้ไดอะแกรม Trellis แล้ว นำเอา Trellis ที่ได้ไปใช้ในการถอดรหัสข้อมูลทางภาครับ ซึ่งเป็นประโยชน์อย่างมาก โดยที่ทางด้านรับต้องทราบ Trellis ของชุดเข้ารหัสที่เข้ามาโดยพารามิเตอร์ที่มีความสำคัญที่ได้จากแผนภาพ Trellis คือ State ก่อนและหลังเข้ารหัสข้อมูลที่ได้แต่ละเสตจของการเข้ารหัส โครงสร้างของชุดเข้ารหัสแบบเทอร์โบ สามารถดูเพิ่มเติมได้จากในบทที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.2.2 การจำลองการทำงานของชุดสลับข้อมูลแบบ Random และแบบ Block**

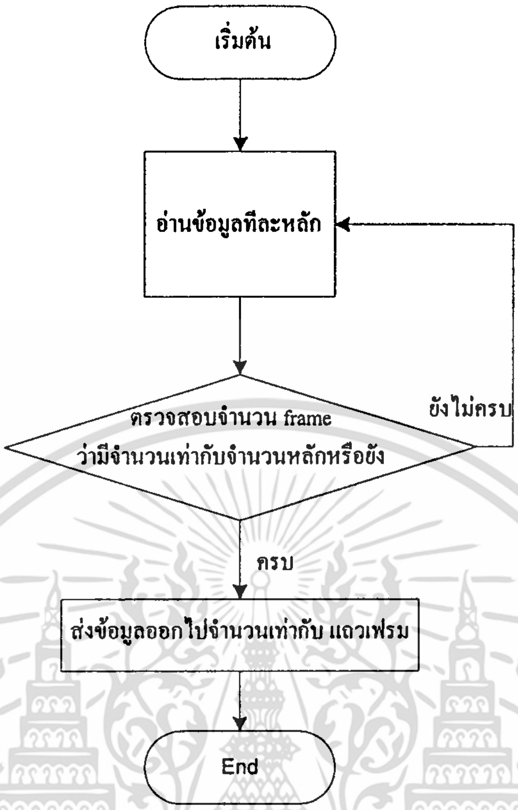
ในการออกแบบในส่วนนี้ได้กล่าวไว้อย่างละเอียดในบทที่ 2 หัวข้อการทำอินเทอร์ลีฟข้อมูล ซึ่งในวิทยานิพนธ์ฉบับนี้จะนำเอาการทำการสลับข้อมูลเพียง 2 รูปแบบมาใช้ คือการทำอินเทอร์ลีฟแบบ Random ซึ่งนำมาใช้ในโครงสร้างของชุดเข้ารหัสแบบเทอร์โบ และชุดทำอินเทอร์ลีฟแบบ Block ซึ่งในระบบ DS-CDMA มาตรฐาน IS-95 นั้น ในโครงสร้างจะมีการทำการอินเทอร์ลีฟข้อมูลหลังจากที่ผ่านชุดเข้ารหัสแล้ว ด้วยการทำการสลับข้อมูลแบบ Block ซึ่งจะมีรูปแบบการทำอินเทอร์ลีฟแบบ 32*18 (หมายความว่าจะอ่านข้อมูลออกไปทีละ 32 บิท จำนวน 18 ชุด) โดยที่ข้อมูลที่ออกไปจะมีการสลับตำแหน่งดังรูป




รูปที่ 3.2 รูปแบบการอ่านข้อมูลแบบ Block Interleave

เพื่อให้เกิดความเข้าใจมากขึ้น ดูรายละเอียดขั้นตอนการทำงานจาก Flow Chart จำลองการทำงานของการทำอินเทอร์ลีฟเวอร์แบบ Block แสดงได้ดังนี้

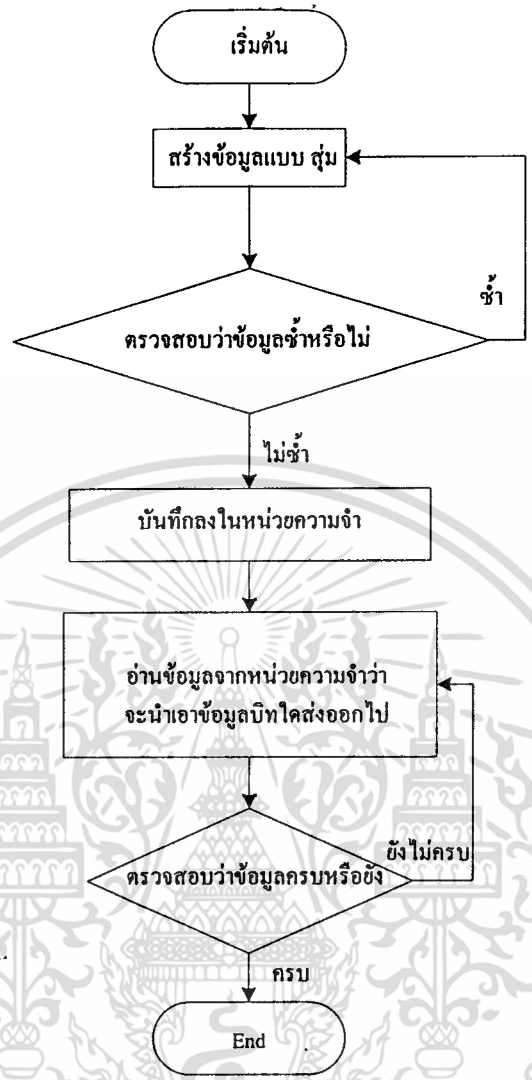
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.3** Flow chart ในการออกแบบการทำอินเทอร์ลีฟเวอร์แบบ Block

จากรูปแสดง Flow Chart การทำงานของการทำอินเทอร์ลีฟเวอร์แบบ Block มีขั้นตอนการทำงานคือ ข้อมูลจะถูกอ่านเข้ามาเก็บยังหน่วยความจำ เพื่อที่จะส่งข้อมูลออกไป โดยข้อมูลที่ส่งออกไปจะถูกอ่านออกไปทีละหลัก จะเห็นว่าในรูปแบบการสลับข้อมูลแบบ Block ข้อมูลที่อ่านออกไปแต่ละเฟรม ประกอบไปด้วยข้อมูลที่มาจากข้อมูลในแต่ละเฟรม โดยข้อมูลตำแหน่งที่หนึ่งจะมาจากข้อมูลในเฟรมที่หนึ่งและข้อมูลที่ตำแหน่งที่สองจะมาจากข้อมูลในเฟรมที่สองที่อ่านเข้ามาเป็นลำดับถัดไปจนถึงข้อมูลชุดสุดท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



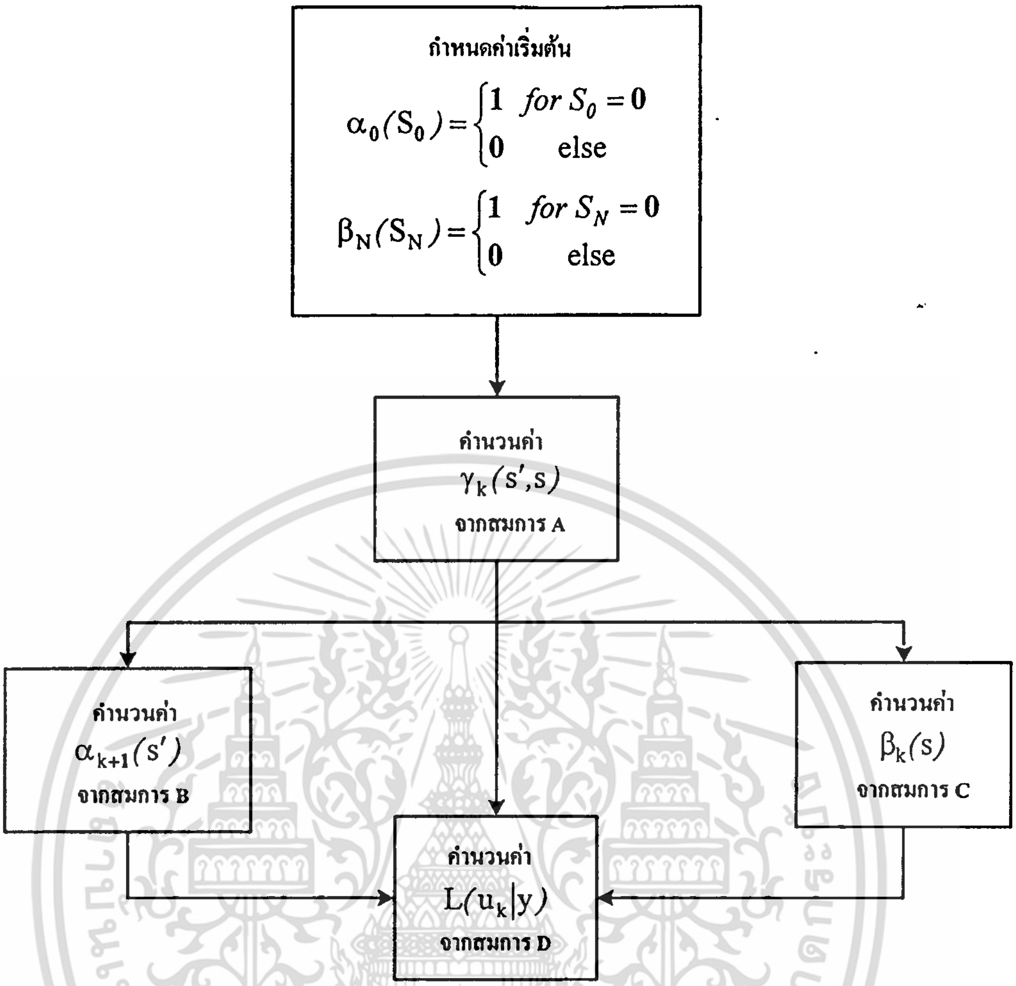

**รูปที่ 3.4 ขั้นตอนการสลับข้อมูลแบบ Random**

จากรูปแสดงขั้นตอนการจัดการสลับตำแหน่งข้อมูลแบบ Random เช่นเดียวกับในการสลับตำแหน่งข้อมูลแบบ Block ข้อมูลจะถูกอ่านเข้ามาเก็บยังหน่วยความจำ แล้วจะส่งข้อมูลออกไปด้วยค่าตำแหน่งข้อมูลที่มีตำแหน่งแบบ Random โดยก่อนจะส่งจะทำการตรวจสอบว่าตำแหน่งที่ทำการส่งซ้ำกับตำแหน่งที่เคยส่งหรือไม่ ถ้าซ้ำจะทำการหาตำแหน่งในการส่งใหม่ที่ไม่ซ้ำ

### 3.2.3 การออกแบบชุดถอดรหัสเทอร์โบ แบบใช้อัลกอริทึมแบบ MAP

รายละเอียดโดยละเอียดอยู่ในส่วนของภาคผนวก ในวิทยานิพนธ์ฉบับนี้นำเอาชุดเข้ารหัสแบบเทอร์โบ ซึ่งมีโครงสร้างแบบขนาน ดังนั้นในส่วนของชุดถอดรหัสก็จะมีลักษณะที่เป็นชุดถอดรหัสจำนวน 2 ชุดต่อขนานกัน ดังโครงสร้างแสดงดังรูป จากรายละเอียดใน ภาคผนวก เมื่อนำมาเขียนเป็น Flow chart แสดงขั้นตอนในการคำนวณได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




สมการที่ใช้ทั้ง 4 สมการแสดงได้ดังนี้

$$\gamma_k(s',s) = C.e^{(u_k L(u_k)/2)}.exp(\frac{E_b}{2\sigma^2}2a\sum_{l=1}^{n} y_{kl}x_{yl}) \qquad \text{สมการ A}$$

ค่า C คือค่าคงที่ ซึ่งไม่ขึ้นกับเครื่องหมาย summation เราสามารถที่จะแยกออกมาได้

$$\alpha_k(S_k) = \frac{\sum_{S_{k-1}}\sum_{i=0}^{1}\gamma_i(y_k,S_{k-1},S_k).\alpha_{k-1}(S_{k-1})}{\sum_{S_k}\sum_{S_{k-1}}\sum_{i=0}^{1}\gamma_i(y_k,S_{k-1},S_k).\alpha_{k-1}(S_{k-1}).\beta_k(S_k)} \qquad \text{สมการ B}$$

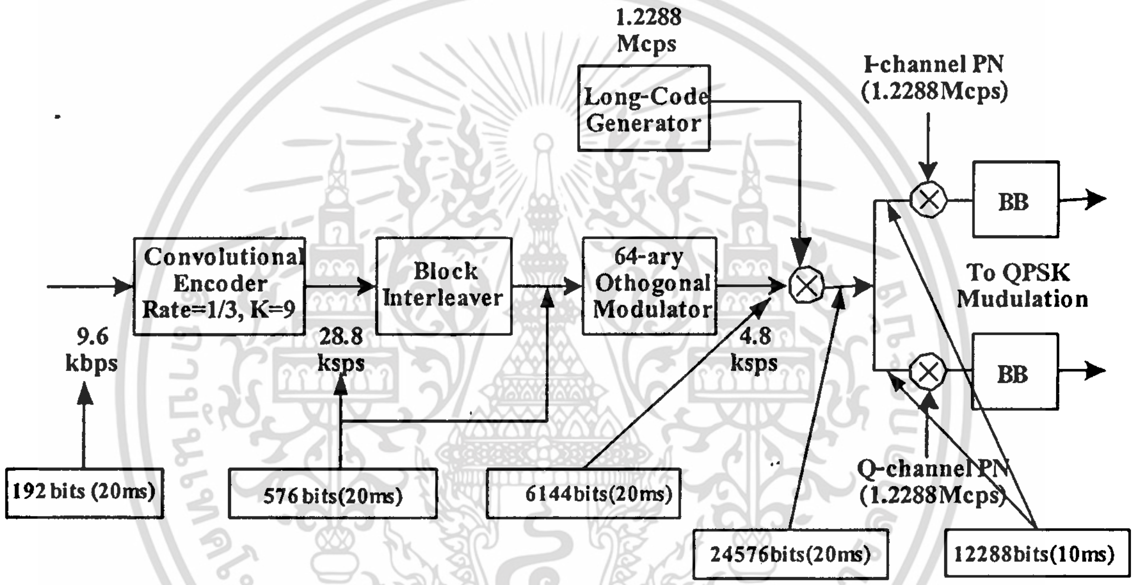
$$\beta_k(S_k) = \frac{\sum_{S_{k-1}}\sum_{i=0}^{1}\gamma_i(y_{k+1},S_k,S_{k+1}).\beta_{k+1}(S_{k+1})}{\sum_{S_k}\sum_{S_{k-1}}\sum_{i=0}^{1}\gamma_i(y_{k+1},S_k,S_{k+1}).\alpha_k(S_k)} \qquad \text{สมการ C}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$L(u_k|y)=ln\left(\frac{\sum\limits_{\substack{(s',s)\Rightarrow\\u_k=+1}}\alpha_{k-1}(s').\gamma_k(s',s).\beta_k(s)}{\sum\limits_{\substack{(s',s)\Rightarrow\\u_k=-1}}\alpha_{k-1}(s').\gamma_k(s',s).\beta_k(s)}\right) \qquad \text{สมการ D}$$

## 3.3 การออกแบบระบบ DS-CDMA มาตรฐาน IS-95

ในระบบ DS-CDMA มาตรฐาน IS-95 ในการส่งในทิศทางจากโทรศัพท์เคลื่อนที่ไปยังสถานีฐานนั้น (Reverse Link) จะมีรูปแบบของข้อมูลที่ส่งดังในไดอะแกรม




รูปที่ 3.5 โครงสร้างของระบบ DS-CDMA มาตรฐาน IS-95 ในทิศทาง Reverse Link

สิ่งที่จำเป็นของระบบ DS-CDMA มาตรฐาน IS-95 ที่ควรทำความเข้าใจก่อน คือ ระบบรูปแบบของการจัดเฟรมข้อมูลของ CDMA มาตรฐาน IS-95 ดังแสดงในรูป ซึ่งจะใช้ในการทำความเข้าใจในแต่ละส่วนที่มีการเปลี่ยนแปลงขนาดของเฟรมข้อมูล รายละเอียดจะแสดงได้ดังต่อไปนี้

ขั้นตอนของการส่งข้อมูลในระบบ DS-CDMA มาตรฐาน IS-95 ในทิศทาง Reverse Link จะมีขั้นตอนดังนี้

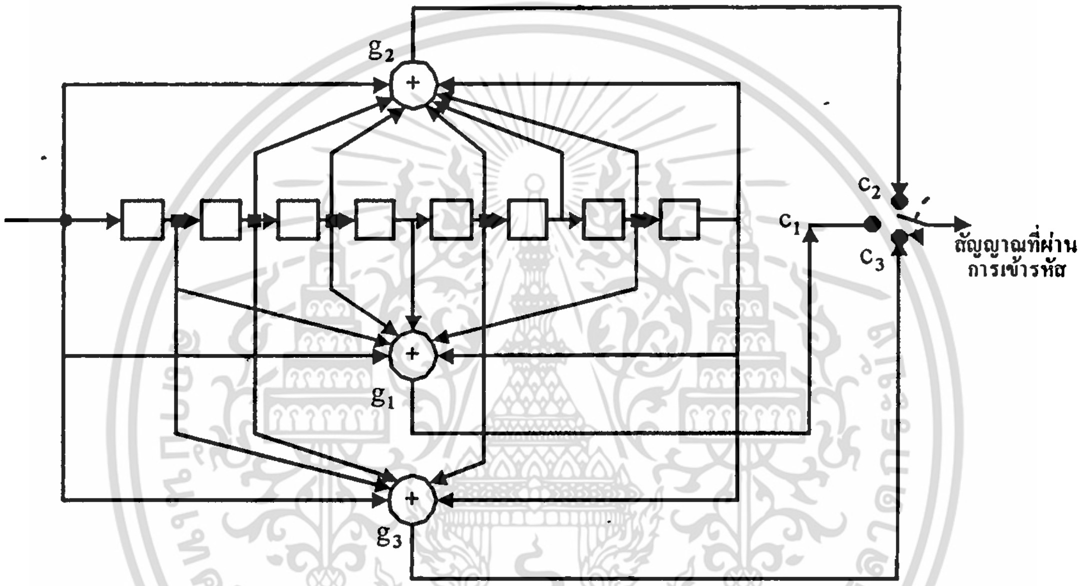
1. ข้อมูลขนาด 88 บิท สำหรับการส่งที่อัตราการส่งข้อมูลเท่ากับ 4800 บิทต่อวินาที และ 192 บิทสำหรับอัตราการส่งข้อมูลที่ 9600 บิทต่อวินาที ถูกส่งเข้าส่วนเข้ารหัสข้อมูลที่มีอัตราการเข้ารหัสเท่ากับ 1/3 ต่อไปนี้จะยกตัวอย่างการส่งข้อมูลที่ 4800 บิทต่อวินาที (88 บิท) โดยจะมีการเติมบิทข้อมูลเข้าไปเพื่อเคลียร์ค่าในหน่วยความจำของชุดเข้ารหัสเท่ากับ 8 บิท จะได้ข้อมูลที่เข้าชุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เข้ารหัสเท่ากับ 88+8=96 บิท ซึ่งทำการส่งในระยะเวลา 20 ms ฉะนั้น อัตราเร็วในการส่งข้อมูลจะมีค่าเท่ากับ 96/20 ms=4800 bps

1.1 การจำลองการทำงานชุดเข้ารหัสคอนโวลูชั่นที่ใช้ในระบบ DS-CDMA มาตรฐาน IS-95

ในการออกแบบชุดเข้ารหัสแบบคอนโวลูชั่นนั้น สิ่งที่จำเป็นอยู่สองส่วน ซึ่งเป็นส่วนสำคัญในการจำลองการทำงานคือ สเตจไดอะแกรมของชุดเข้ารหัสคอนโวลูชั่น และส่วนประกอบของโครงสร้างของชุดเข้ารหัสที่ใช้ ซึ่งต่อไปนี้จะแสดงตัวอย่างของชุดเข้ารหัสแบบคอนโวลูชั่น ซึ่งมีเจนเนอร์เรเตอร์ g1=373,g2=557, g3=711




รูปที่ 3.6 โครงสร้างของชุดเข้ารหัสคอนโวลูชั่นที่ใช้ในระบบ DS-CDMA

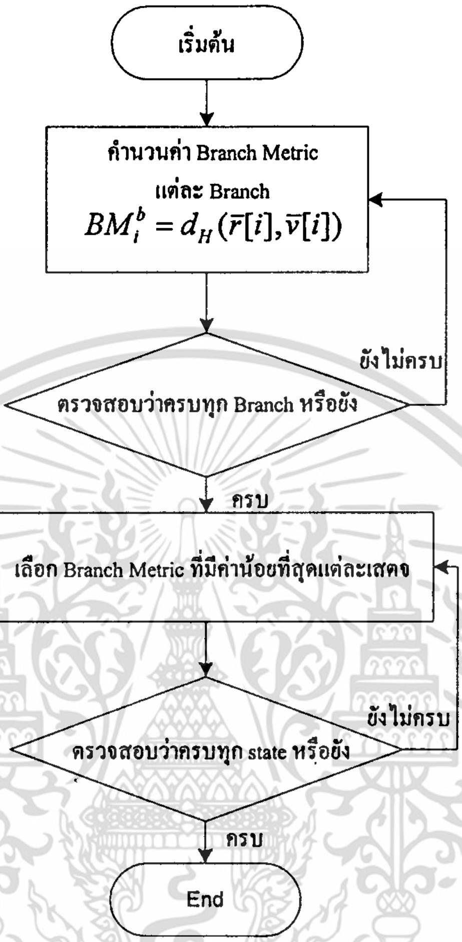
หลังจากได้สร้าง Trellis แล้ว ต่อไปจะทำการสร้างในส่วนของชุดเข้ารหัสคอนโวลูชั่น ซึ่งผลของข้อมูลทางด้านเอาท์พุทจะมีค่ามากขึ้นตามอัตราการเข้ารหัส เช่น ถ้าชุดเข้ารหัสเป็นแบบคอนโวลูชั่น ที่มีอัตราการเข้ารหัส =1/3 ซึ่งหมายความว่าถ้ามีข้อมูลเข้ามา 1 บิท จะมีข้อมูลออกมา 3 บิท

1.2 การออกแบบชุดถอดรหัสคอนโวลูชั่น แบบใช้อัลกอริทึมแบบไวตาบิในการถอดรหัส

วิธีการในการจำลองการทำงานในส่วนของชุดถอดรหัสข้อมูลแบบไวตาบิ จะแบ่งออกเป็น 2 ส่วนด้วยกันคือ การถอดรหัสแบบ Soft decision Viterbi และ Hard decision Viterbi ซึ่งจะแสดงการจำลองการทำงานในรูปของ Flow Chart ในการออกแบบได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ 1 แสดง Flow Chart ในการจำลองการทำงานของ Hard Decision Viterbi แสดงรายละเอียดได้ดังนี้

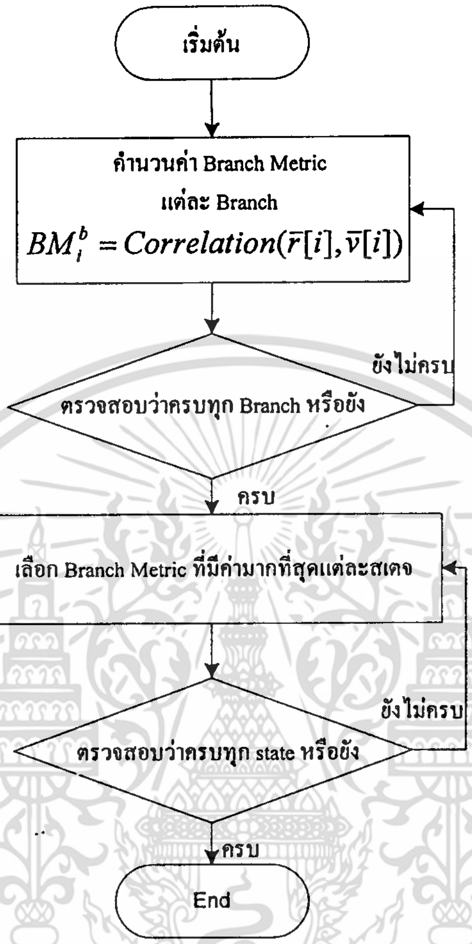



รูปที่ 3.7 Flow Chart ในการทำงานของการถอดรหัสข้อมูลแบบ Hard decision Viterbi

ขั้นตอนของ Hard decision จาก Flow Chart อัลกอริทึมในการถอดรหัส Viterbi ขั้นตอนในการเลือก path metric อัลกอริทึมในการถอดรหัสแบบ Hard decision ขั้นตอนในการเลือก path metric จะมีขั้นตอน 3 ขั้นตอน คือ ACS(ADD Compare Select) ซึ่งมีความหมายว่าจะทำการคำนวณ path metric โดยทำการหาค่าความแตกต่างกันของ branch กับ ข้อมูลที่รับได้ โดย path metric=ข้อมูลที่รับได้ที่เวลา k ลบด้วย Brach metric ทุก Branch ทั้งหมดยกกำลัง 2 เมื่อผ่านขั้นตอนการคำนวณจะได้ค่าของ Metric ทุก Branch ต่อไปจะทำการเลือก Branch Metric ที่สเตจถัดไป โดยจะเลือก Branch ที่เข้ามายังสเตจนั้นที่มีค่าน้อย โดยถือว่ามีความแตกต่างกันน้อยที่สุด สุดท้ายในแต่ละสเตจจะมีเพียง branch เดียวเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

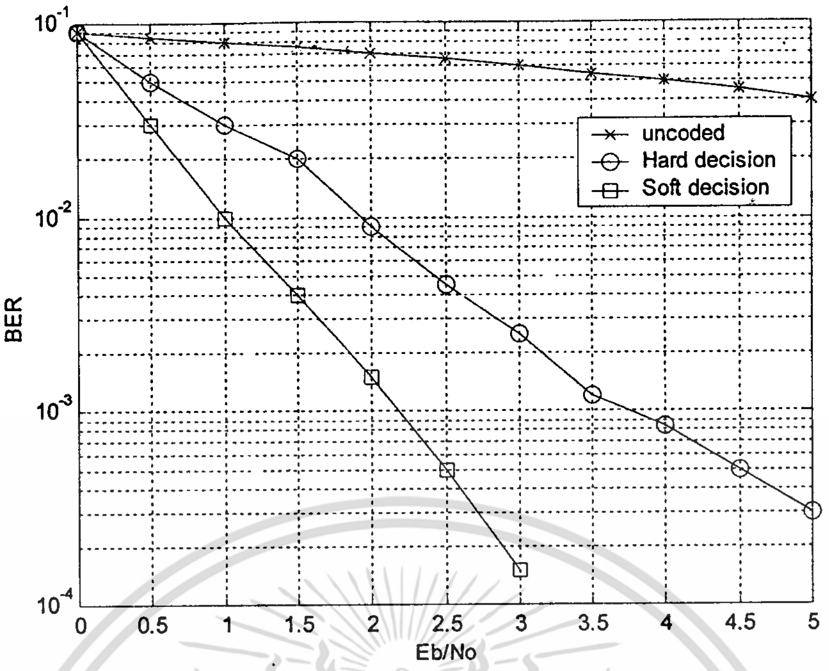
ส่วนที่ 2 แสดง Flow Chart ในการจำลองการทำงานของ Soft Decision Viterbi แสดงรายละเอียดได้ดังนี้




รูปที่ 3.8 Flow Chart ในการทำงานของการถอดรหัสข้อมูลแบบ Soft decision Viterbi

ขั้นตอนของ Soft decision จะแตกต่างกับแบบ Hard decision ในส่วนของการคำนวณหาค่า Branch Metric ตรงที่ในวิธีการแบบ Soft decision จะคำนวณ Branch Metric โดยใช้วิธีการ Correlation ข้อมูลที่รับได้กับค่าข้อมูลแต่ละ Branch ซึ่งในขั้นตอนเลือกจะทำการเลือกค่าสูงสุด ซึ่งถือว่ามีความเหมือนมากที่สุด ซึ่งจะแตกต่างกับแบบ Hard decision ซึ่งจะแสดงประสิทธิภาพของระบบที่ใช้การอัลกอริทึมแบบ Soft และแบบ Hard ในการถอดรหัส ได้ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



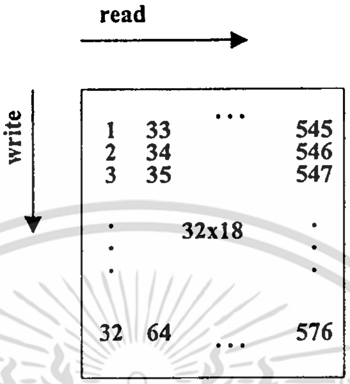

รูปที่ 3.9 ประสิทธิภาพของระบบที่เข้ารหัสข้อมูลแบบคอนโวลูชั่น ความยาวของชุดเข้ารหัสเท่ากับ 9 ที่ใช้อัลกอริทึมในการถอดรหัสแบบ Soft และ แบบ Hard เทียบกับระบบที่ไม่ได้เข้ารหัส

จากรูปจะเห็นว่าประสิทธิภาพของระบบที่ใช้การถอดรหัสแบบ Soft จะให้ประสิทธิภาพออกมาดีกว่าระบบที่ใช้การเข้ารหัส แบบ Hard มาก ซึ่งในวิทยานิพนธ์จะใช้การถอดรหัสข้อมูลที่ใช้การเข้ารหัสแบบคอนโวลูชั่นด้วย Soft Decision Viterbi ซึ่งตรงกับมาตรฐานที่ใช้ในระบบ DS-CDMA มาตรฐาน IS-95

2. จำนวนข้อมูลที่ออกจากชุดเข้ารหัสจะมีจำนวนข้อมูลเท่ากับ 96x3=288 บิท ซึ่งจะมีอัตราการส่งข้อมูลเท่ากับ 288/20ms=14.4 kbps จะถูกส่งผ่านวงจรปรับอัตราการส่งข้อมูลให้มีค่าเท่ากับ 28.8 kbps ซึ่งจะต้องทำการส่งข้อมูลซ้ำอีกจำนวน 1 ชุด ซึ่งทำให้อัตราการส่งข้อมูลสูงขึ้นเป็น 14.4kbps x 2=28.8 kbps

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. หลังจากผ่านวงจรปรับอัตราการส่งข้อมูลเป็นที่เรียบร้อยแล้ว จะผ่านชุดทำการสลับตำแหน่งข้อมูล Interleave ซึ่งข้อมูลที่ออกจากวงจรปรับอัตราการส่งข้อมูลจะมีจำนวนบิทเท่ากับ 288X2=576 บิท ซึ่งยังคงส่งในเวลา 20 ms โดยรูปแบบของการทำอินเทอร์ลีฟ เป็นแบบ บล๊อคอินเทอร์ลีฟ มีการจัดรูปแบบข้อมูลเป็นเมตริกส์ขนาด 32*18 สมาชิกของเมตริกส์จะมีค่าเท่ากับชุดของบิทข้อมูลที่ป้อนเข้าอินเทอร์ลีฟ ซึ่งมีค่าเท่ากับ 576 บิท




รูปที่ 3.10 การทำอินเทอร์ลีฟข้อมูลแบบ Block

4. หลังจากผ่านขั้นตอนการสลับตำแหน่งข้อมูลแล้ว อัตราการส่งข้อมูลยังคงเท่าเดิม 28.8 kbps ต่อไปจะถูกส่งเข้าวงจร Hadamard Modulator ซึ่งเป็นการมอดูเลตแบบ noncoherent โดยที่ทางด้านชุดรับสามารถรับข้อมูลได้อย่างถูกต้องโดยทำการหา Correlation ของสัญญาณที่ได้รับกับ Hardamard ทั้ง 64 ชุด ว่ามีความเหมือนกับ Hardamard ชุดใดมากที่สุด หลักการในระบบ DS-CDMA มาตรฐาน IS-95 ในทิศทาง Reverse Link จะแบ่งข้อมูลที่รับมาจากวงจรการทำอินเทอร์ลีฟจำนวน 576 บิท มาทำการแบ่งข้อมูลออกเป็นชุดละ 6 บิท ซึ่งจะได้ทั้งหมดจำนวน 576/6=96 ชุด โดยที่แต่ละชุดจะถูก Map เข้ากับ Hardamard ทั้ง 64 ชุด โดยวิธีการ Map จะมีวิธีการดังนี้

สมมุติข้อมูล 6 บิท มีค่าเท่ากับ 100101 ซึ่งมีค่าเท่ากับ 37 ในระบบเลขฐานสิบก็จะทำการ MAP ข้อมูล 100101 เข้ากับ Hardamard ลำดับที่ 37 ข้อมูลของ Hardamard ลำดับที่ 37 จะถูกส่งไปยังเครื่องรับทางด้านเครื่องรับจะทำการกู้ข้อมูลกลับมาได้โดยทำการหาค่า Correlation ข้อมูลขนาด 64 บิท ที่รับได้กับ Hardamard ทั้ง 64 ชุด แล้วจะตัดสินใจว่ามีค่าเหมือนกับ Hardamard ชุดใด จากค่าของ Correlation ที่มีค่าสูงที่สุด และจะทำการ Map ข้อมูลจาก 64 บิท กลับเป็น 6 บิท โดยดูจาก Order ของ Hardamard ที่มีค่าสูงสุด ดังแสดงได้ในตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**การออกแบบ Harddamard**

ต่อไปนี้เราจะสร้าง เมตริกส์ Hardamard ขนาด 64*64 ซึ่งเป็นขนาดที่ในระบบ DS-CDMA มาตรฐาน IS-95 ใช้ มีขั้นตอนการออกแบบดังนี้

**การออกแบบ Hardamard matrix**

วิธีการออกแบบจะทำการออกแบบ Hardamard Matrix โดยใช้ Hardamard ขนาด 2x2 เป็นหลักในการออกแบบ โดยจะทำการออกแบบ Hardamard ขนาด 4x4 และ 8x8 , 16x16 , 32x32 , 64x64 ซึ่งหลักการออกแบบจะทำการออกแบบตามลำดับดังต่อไปนี้

H2→H4→H8→H16→H32→H64

รูป Matrix ขนาด 2x2

$$H2=\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}$$

ผลที่ได้จะได้ข้อมูลจำนวน 64 ชุด ๆละ 64 บิท ซึ่งข้อมูลแต่ละแถวของ Hardamard จะถูกเรียกว่า Walh Code ซึ่งความหมาย W0 มีความหมายว่า เป็นข้อมูล Hardamard เมตริกส์แถวที่ 0 ซึ่งจะมี Walh Code ทั้งสิ้นจำนวน 64 ชุด{W0,W1,W2,.......W63}

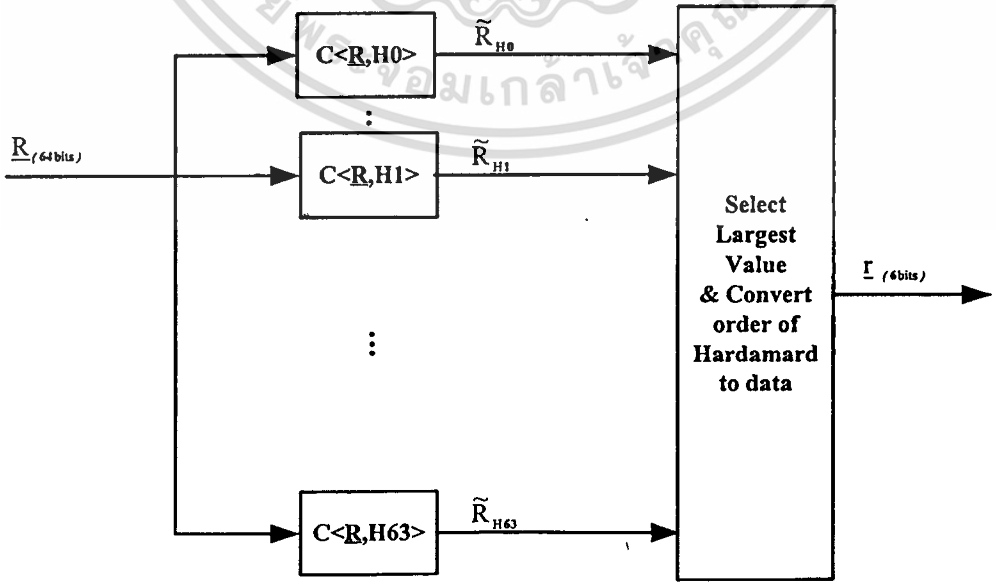
**ตัวอย่างของการกู้ข้อมูลคืนทางด้านปลายทาง**

ข้อมูลทางด้านต้นทาง = 100101

MAP ได้เท่ากับ Hardamard order 37=100000000000000000000

เมื่อมาถึงยังเครื่องรับสัญญาณที่รับได้

R=1100000000000000001000000010000 ส่วนที่ขีดเส้นใต้คือข้อมูลที่ผิดไปโดยมาจากการถูกรบกวนสัญญาณ R จะถูกนำมา Correlate กับ Hardamard ทั้ง 64 ชุด ดังแสดงได้ใน Diagram




**รูปที่ 3.11** โครงสร้างของชุด Correlation ทางด้านชุดรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการ Correlations ของ Hardamard แต่ละชุดมีค่าเท่ากับ

| | | | |
|---|---|---|---|
| C<R,H0>= -4 | C<R,H1>= 12 | C<R,H2>= 31 | C<R,H3>=26 |
| C<R,H4>= 1 | C<R,H5>= 13 | C<R,H6>= 19 | C<R,H7>=27 |
| C<R,H8>= -6 | C <R,H9>= 14 | C<R,H10>= 7 | C<R,H11>=29 |
| C<R,H12>= -8 | C<R,H13>= 15 | C<R,H14>= -6 | C<R,H15>=2 |
| C<R,H16>= -5 | C<R,H17>= 20 | C<R,H18>= -3 | C<R,H19>=-5 |
| C<R,H20>= 3 | C<R,H21>= 22 | C<R,H22>= -9 | C<R,H23>= -18 |
| C<R,H24>= 2 | C<R,H25>= -25 | C <R,H26>= -21 | C<R,H27>= -30 |
| C<R,H28>= -7 | C<R,H29>= 28 | C<R,H30>= - 2 | C<R,H31>= -27 |
| C<R,H32>= -9 | C<R,H33>= 32 | C<R,H34>= 7 | C<R,H35>= -16 |
| C<R,H36>= -16 | C<R,H37>= 62 | C<R,H38>= 12 | C<R,H39>= -10 |
| C<R,H40>= 20 | C<R,H41>= 15 | C<R,H42>= -8 | C<R,H43>=-6 |
| C<R,H44>= -18 | C<R,H45>= 14 | C<R,H46>= 12 | C<R,H47>=15 |
| C<R,H48>= -15 | C<R,H49>= 18 | C<R,H50>= 11 | C<R,H51>= 14 |
| C<R,H52>= -19 | C<R,H53>= 25 | C<R,H54>= 10 | C<R,H55>= 11 |
| C<R,H56>= -30 | C<R,H57>=36 | C<R,H58>= 9 | C<R,H59>= 12 |
| C<R,H60>= 8 | C<R,H61>= 42 | C<R,H62>= -8 | C<R,H63>=-9 |

จากการทำ Correlation จะพบว่าค่า correlation ของ <R,H37> จะมีค่ามากที่สุด ดังนั้นเราจึงตัดสินใจว่าข้อมูลที่ส่งมาทางด้านปลายทางมีความเหมือนกับ Hardamard Order 37 มากที่สุด ซึ่งผลของข้อมูลจะมีค่าจำนวน order ของ Hardamard ซึ่งมีค่าเท่ากับ 37 ในระบบเลขฐาน 10 แปลงเป็นระบบเลขฐานสองได้ 100101 ซึ่งตรงกับข้อมูลที่ส่งมาทางด้านต้นทาง หลังจากผ่านชุด Orthogonal Hardamard Modulation แล้วจำนวนบิทข้อมูลที่ส่งมีค่าเท่ากับ $\frac{576}{6} * 64 = 6144$ บิท ซึ่งยังคงส่งในช่วงเวลา 20ms เหมือนเดิม ฉะนั้นอัตราในการส่งข้อมูลจะสูงขึ้นเท่ากับ 6144/20 ms=307,200 bps หรือเท่ากับ 307.2 Kbps

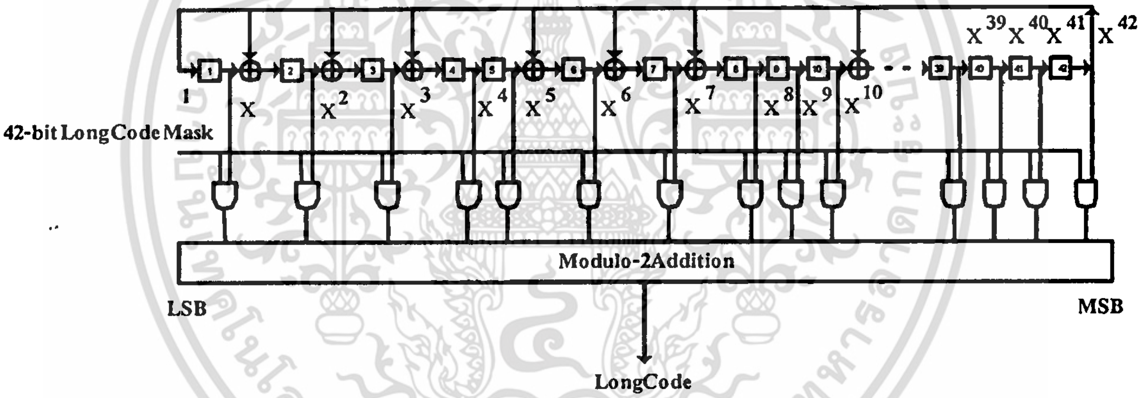
5. หลังจากผ่าน Hardamard Modulator แล้ว ข้อมูลจะถูกทำการเสปรดด้วย PN –Sequence ซึ่งในระบบโทรศัพท์ IS-95 ในทิศทาง Reverse นั้นจะใช้ PN –Sequence จำนวน 2 ชุด ซึ่ง PN-Sequence ชุดแรกจะถูกสร้างมาจาก LFSR ที่มี Order ของการสร้างข้อมูลจำนวน $2^{41} - 1$ ค่า ซึ่งเป็นข้อมูลที่มีความยาวมาก จึงเรียกว่า Long PN-Sequence การออกแบบจะแสดงในส่วนถัดไป ชุด PN-Sequence ชุดที่สอง จะถูกสร้างขึ้นมาจาก LFSR เช่นเดียวกับแบบแรก แต่ในแบบที่สองนี้จะมี PN จำนวนสองชุดสำหรับ I และ Q รายละเอียดแสดงในส่วนถัดไปซึ่ง PN ทั้งสองชุดมีลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของโพลีโนเมียลสูงสุดเท่ากับ 15 ฉะนั้น สามารถสร้างข้อมูล PN ได้ $2^{15}-1$ ซึ่งมีความยาวข้อมูลเท่ากับ 32677 บิท (จะทำการเพิ่มบิทสุดท้าย บิท 0 เข้าไปเพื่อให้ได้จำนวนบิทเท่ากับ 32678) ซึ่งเรียกการสร้าง PN แบบนี้ว่า Short-PN-Sequence รายละเอียดในการสร้างจะแสดงดังต่อไปนี้

ในระบบ DS-CDMA มาตรฐาน IS-95 จะเรียก Long –PN—sequence ในทิศทาง Reverse Link ว่า Scrambling Codes และ Short PN ว่า Channelization Codes ซึ่งกล่าวไว้ในเอกสารอ้างอิง [spread Codes for Direct Sequence CDMA and Wideband CDMA Cellular Network] เนื่องมาจากหน้าที่ในการทำงานของ PN Sequence แต่ละตัวจะแตกต่างกันโดย Long PN-Sequence จะมีหน้าที่แบ่งเฟรมของข้อมูลในทิศทาง reverser Link ซึ่งแต่ละเฟรมจะถูกคูณด้วย PN ที่แตกต่างกัน เป็นการป้องกันการเกิด ISI ภายใน และ Short PN-Sequence มีประโยชน์ในการแบ่งแยกเซลล์แต่ละเซลล์โดยที่แต่ละเซลล์จะถูกคูณด้วย Short PN คนละตัวกัน ซึ่งในระบบ DS-CDMA จะมีสถานีฐานได้มากที่สุดเท่ากับ 512 สถานีฐาน ซึ่งแต่ละสถานีฐานจะมี Short PN ที่ต่างกัน 64

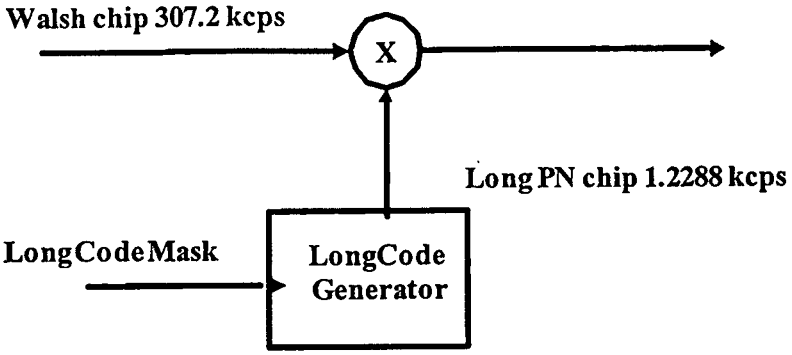
**การสร้าง PN Sequence**




รูปที่ 3.12 โครงสร้างของ LFSR ที่ใช้ในการสร้าง Long PN-Sequence

ข้อมูลที่ออกมาจาก Hardamard Modulator จะถูกทำการ Spread ข้อมูลด้วย Long PN-Sequence ซึ่งข้อมูลขนาด 6144 บิท ที่มีอัตราการส่งข้อมูลเท่ากับ 307.2 Kbps(ส่ง 20ms) จะถูกทำการยืดบิทข้อมูลแต่ละบิทออกไปอีก 4 เท่า ซึ่งหมายความว่า processing Gain มีค่าเท่ากับ 4 ซึ่งการยืดข้อมูลออกแสดงได้ดังตัวอย่าง

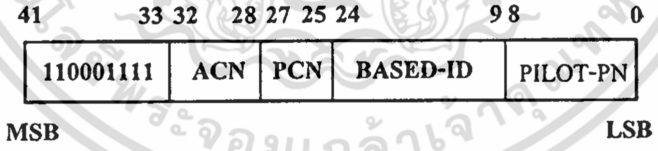
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.13** การนำเอา Long PN Code มาทำการสเปรดข้อมูล

ข้อมูล 1001001 ถูกทำการยืดข้อมูลเท่ากับ 4 ( Processing Gain =4) จะมีค่าเท่ากับ 1111 0000 0000 1111 0000 0000 1111 ซึ่งผลของการยืดข้อมูลออกเท่ากับ 4 เท่าจะทำให้อัตราการส่งข้อมูลสูงขึ้น เท่ากับ 307.2 Kbps * 4 =1.2288 Mbps (ส่งข้อมูลทั้งหมดภายในเวลา 20 ms ) ซึ่งจำนวนของบิทข้อมูลที่ส่งไปจะเท่ากับ 6144*4=24576 บิท ซึ่งจะถูกทำการสเปรดด้วยข้อมูลที่สร้างมาจาก Long-PN อัตราการส่งข้อมูลจะมีค่าเท่ากับ 1.2288 Mbps (ส่ง 24576 บิท ในช่วงเวลา 20 ms)

ส่วนที่สำคัญอีกส่วนหนึ่งคือ Long Code Mask แสดงได้ดังรูปที่ 3.14 ซึ่งมีหน้าที่เหมือนหน้ากากของข้อมูล ซึ่งจะมีขนาด 42 บิท ซึ่งจะมาแอนด์กับบิทข้อมูลของแต่ละสถานะผลของการแอนด์แต่ละบิทจะมากระทำการบวกแบบมอดูโล 2 ซึ่งจะได้เอาท์พุทออกมาเป็นบิทข้อมูลทีละ 1 บิท ซึ่งนำมาคูณกับข้อมูลที่ออกมาจากชุด Orthogonal Hardamard Modulation



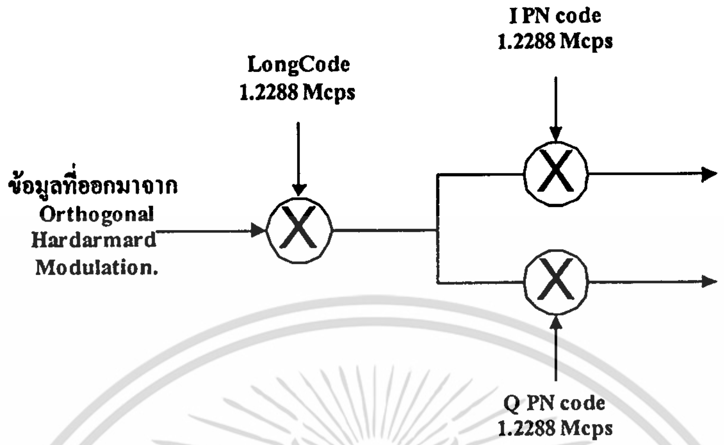

**ACN:Acess Channel Number**
**PCN:Paging Channel Number**
**BASED-ID:Based Station Identification**
**PILOT-PN:PN offset for Forward CDMA Channel**

**รูปที่ 3.14** Access Long Code Mask

หมายเหตุ : **ในวิทยานิพนธ์ฉบับนี้กำหนดให้** ACN=11111 และ PCN=111, BASED-ID=0000000011111111 และ PILOT-PN=000000000 ซึ่งเฟรมของ Mask จะได้เท่ากับ 110001111 11111 111 0000000011111111 000000000 ซึ่งจะนำเอาไปกระทำแอนด์กับโพลีโนเมียลของ Long Code ซึ่งค่าของ Mask จะมีค่าที่เปลี่ยนแปลงตามพารามิเตอร์ ทั้ง 4 ตัว (ACN,PCN,BASED-ID,PILOT-PN)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. หลังจากผ่านการสเปรดข้อมูลด้วย Long-PN แล้วข้อมูลจะถูกแบ่งออกเป็น I และ Q เพื่อที่จะนำไปมอดูเลต แบบ OQPSK ซึ่งข้อมูลก่อนที่จะมอดูเลตแบบ OQPSK จะถูกทำการคูณด้วย Short PN ทั้งข้อมูลชุด I และข้อมูลชุด Q ดังแสดงในไดอะแกรม



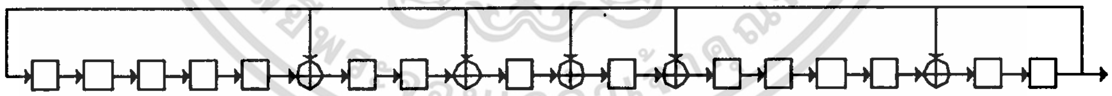

รูปที่ 3.15 การคูณข้อมูลด้วย Short PN

ชุดสร้าง Short PN มีอยู่ 2 ชุดคือ I และ Q โดยแต่ละชุดจะสร้างจากการใช้รีจิสเตอร์ขนาด 15 บิท โดยเชื่อมต่อให้มีโครงสร้างเป็น Maximum length LFSRs ซึ่งจะใช้โพลีโนเมียล 2 ชุด ดังต่อไปนี้

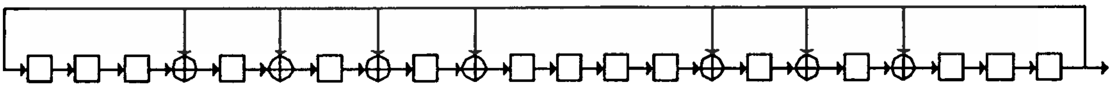
$$I(x) = x^{15} + x^{13} + x^{9} + x^{8} + x^{7} + x^{5} + 1$$

$$Q(x) = x^{15} + x^{12} + x^{11} + x^{10} + x^{6} + x^{5} + x^{4} + x^{3} + 1$$

โดยที่โครงสร้างของ LFSRs ทั้ง 2 ชุด สามารถแสดงได้ดังนี้



รูปที่ 3.16 วงจร 15-Stage LFSR สำหรับ I



รูปที่ 3.17 วงจร 15-Stage LFSR สำหรับ Q

**หน้าที่ของ Short PN**

ดังที่ได้กล่าวไว้ในตอนต้นแล้วว่า Short PN มีหน้าที่ Channelization Codes มีความหมายว่า Short PN จะทำหน้าที่ระบุหมายเลขประจำตัวของสถานีฐาน ซึ่งจำนวนของข้อมูลที่สร้างได้จาก Short PN จะมีได้เท่ากับ 2^15-1=32.768 ชิปส์ ซึ่งในระบบ DS-CDMA มาตรฐาน IS-95 กำหนดให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานีฐานแต่ละสถานีมีอัตราชิปส์ห่างกัน 64 ชิปส์ เพราะฉะนั้นก็จะมีโค้ดที่แตกต่างกันสำหรับการใช้งานทั้งหมด 32,768/64 =512 สำหรับระบุหมายเลขประจำตัวของสถานีฐานได้มากถึง 512 แห่ง

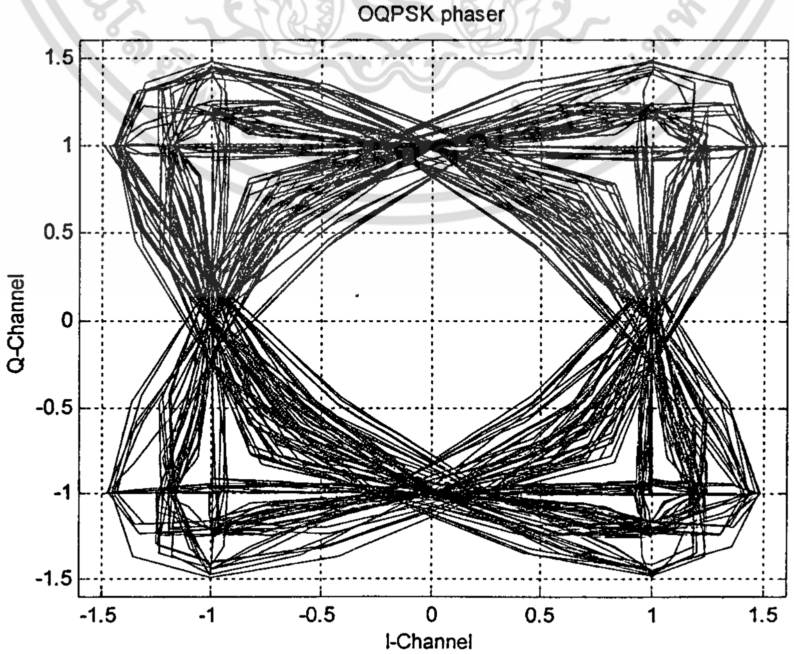
ในการ Simulation เราสามารถทำการแยกชุดข้อมูล I และ Q ได้โดยแยกชุดข้อมูลที่มีดัชนีที่เป็นเลขคี่ได้เป็น I และแยกชุดข้อมูลที่มีดัชนีที่เป็นเลขคู่ Q ข้อมูลที่ผ่านการมอดูเลตแบบ OQPSK ชุดข้อมูล I จะมีอัตราการส่งข้อมูล 1.2288 Mbps เนื่องจากว่าจะส่งข้อมูลทั้งหมดที่เวลาลดลงครึ่งหนึ่ง(10 ms) และข้อมูลของ Q ก็เช่นเดียวกัน

7. การสร้าง OQPSK (Offsets Quadrature Phase Shift Keying) เอกสารอ้างอิง [Simulation and Software Radio for Mobile Communications บทที่ 3 หน้า 98-100]

รายละเอียดในการสร้าง OQPSK เราจะแบ่งออกเป็น 2 หัวข้อย่อย ๆ คือ ส่วนแรกจะกล่าวถึง OQPSK และส่วนที่สองจะพูดถึงการจำลองการทำงานโดยโปรแกรมคอมพิวเตอร์

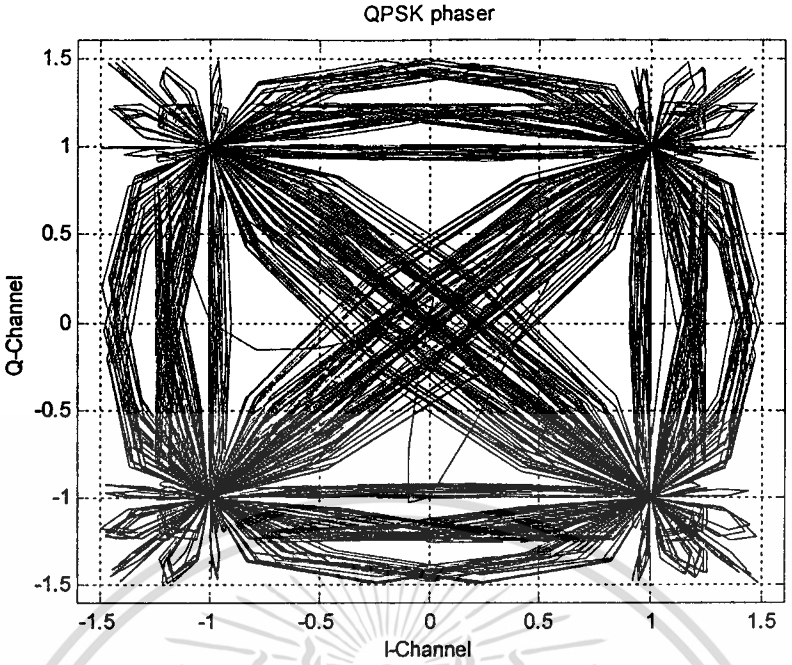
**พื้นฐานของ OQPSK**

ในโครงสร้างของการส่งข้อมูลแบบ QPSK นั้น จะประกอบด้วยข้อมูลจำนวน 2 ชุด ชุดแรกเป็นชุดข้อมูลของ Ich (Inphase) และส่วนที่สองเป็นชุดของ Qch(Quadrature) ซึ่งแสดงได้ในรูป $(d_I, d_Q)$ ซึ่งมีรูปแบบของการเปลี่ยนแปลงข้อมูลได้ จำนวน 4 รูปแบบ คือ (1,1),(1,-1), (-1,-1) และ (-1,1) ซึ่งรูปร่างของสัญญาณที่ทำการส่งแสดงได้ดังรูปที่ 3.18 จากรูปจะเห็นได้ว่าในการส่งสัญญาณนั้นบางครั้งสัญญาณจะตัดผ่านที่จุดกำเนิด ซึ่งหมายความว่ามีการเปลี่ยนแปลงของเฟสไป 180 องศา ซึ่งค่าสเปรคตรัมของสัญญาณจะมีค่ามากขึ้นเมื่อเทียบกับระบบของ PSK ทั่วไป ซึ่งวิธีการที่จะลดปัญหาที่เกิดนี้ลงได้ สามารถทำได้โดยใช้ระบบ OQPSK แต่ค่าของ BER จะได้ออกมาเหมือน กับ BER ของ QPSK

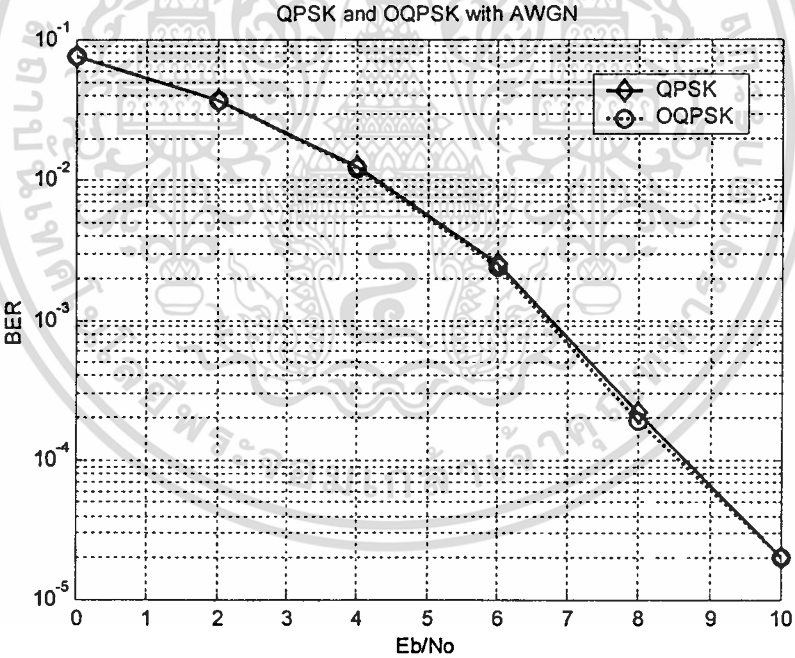



**รูปที่ 3.18 เฟสของ OQPSK**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 3.19 เฟสของ QPSK**




**รูปที่ 3.20 ประสิทธิภาพของ QPSK และ OQPSK บนช่องสัญญาณ AWGN**

จากกราฟแสดงให้เห็นว่าประสิทธิภาพของ QPSK และ OQPSK ในการจำลองการทำงานจะให้ประสิทธิภาพออกมาไม่แตกต่างกัน แต่ในทางปฏิบัติ ดังที่กล่าวไว้ในตอนต้น ดังนั้นในการจำลองการทำงานโดยใช้โปรแกรมคอมพิวเตอร์ ทางผู้วิจัยจึงเลือกเอา QPSK มาใช้แทน OQPSK (ตามมาตรฐาน IS-95) ซึ่งถือว่าให้ประสิทธิภาพออกมาไม่แตกต่างกัน

ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. ชุดสร้างสัญญาณรบกวนแบบ Additive White Gaussians Noise

การออกแบบชุดสร้างสัญญาณรบกวนมีขั้นตอนการสร้างดังนี้

ขั้นตอนที่ 1 สร้างข้อมูลแบบสุ่ม 2 ชุด

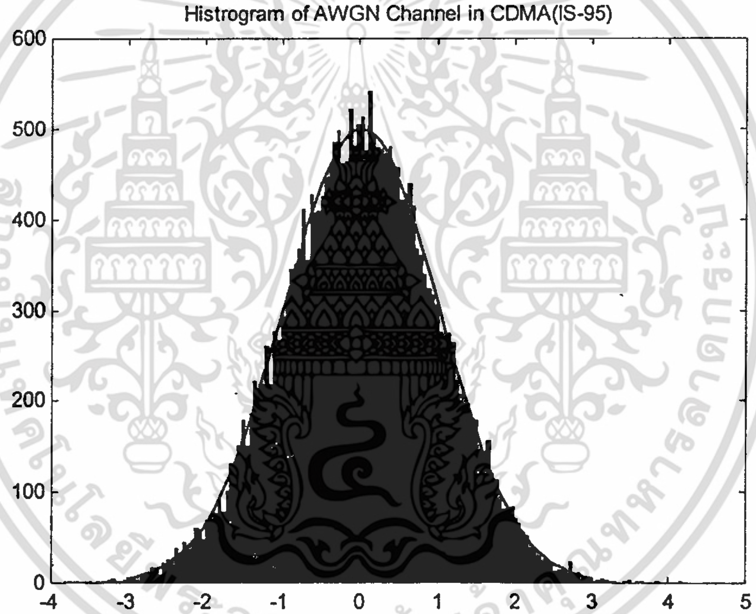
ขั้นตอนที่ 2 ข้อมูลสุ่มชุดแรก แทนด้วย ฟังก์ชั่น $f$ ข้อมูลชุดที่สอง แทนด้วยฟังก์ชั่น $g$

ฟังก์ชั่น $f$ มีค่าเท่ากับ $\sqrt{-\log(a)}$

ฟังก์ชั่น $g$ มีค่าเท่ากับ $\sqrt{2}\cos(2\pi b)$

โดยที่ a และ b คือ ค่าข้อมูลแบบสุ่ม

นำฟังก์ชั่น $f$ และ $g$ มาคูณกันจะได้เป็นข้อมูลแบบสุ่ม ซึ่งเป็นข้อมูลที่มีการกระจายแบบเกาส์เซี่ยน ซึ่งสามารถตรวจสอบโดยดูพล็อตค่าฮิสโตรแกรม ของข้อมูลว่ามีการกระจายตัวแบบเกาส์เซี่ยน




รูปที่ 3.21 ฮิสโตรแกรมของข้อมูล WGN

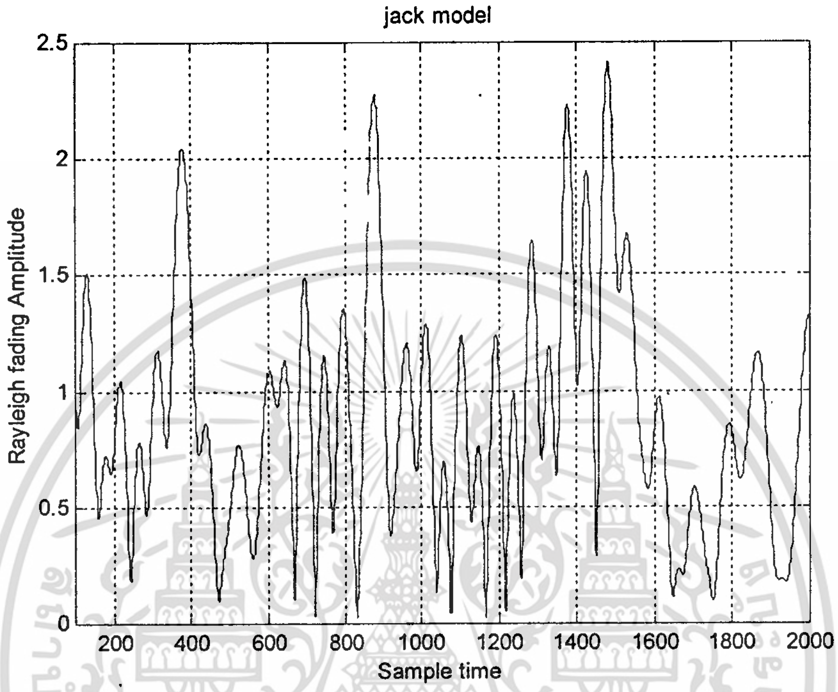
หลังจากได้ข้อมูลแบบสุ่มที่มีการกระจายข้อมูลแบบเกาส์เซี่ยนแล้ว ในการจำลองการทำงานจะถือว่าข้อมูลที่ได้นี้เป็นข้อมูลที่มารบกวนระบบ โดยการรบกวนในการจำลองการทำงานจะนำมาบวกเข้ากับข้อมูลที่ส่งทางด้านต้นทาง ฉะนั้นข้อมูลทางด้านชุดรับจะประกอบด้วยข้อมูลที่ส่งจากทางด้านชุดส่งรวมกับข้อมูลที่สร้างขึ้น (สัญญาณรบกวน)

ต่อไปจะกล่าวถึง การสร้างข้อมูลที่เข้ามารบกวนระบบอีกรูปแบบหนึ่ง ซึ่งจะกล่าวถัดจากหัวข้อนี้

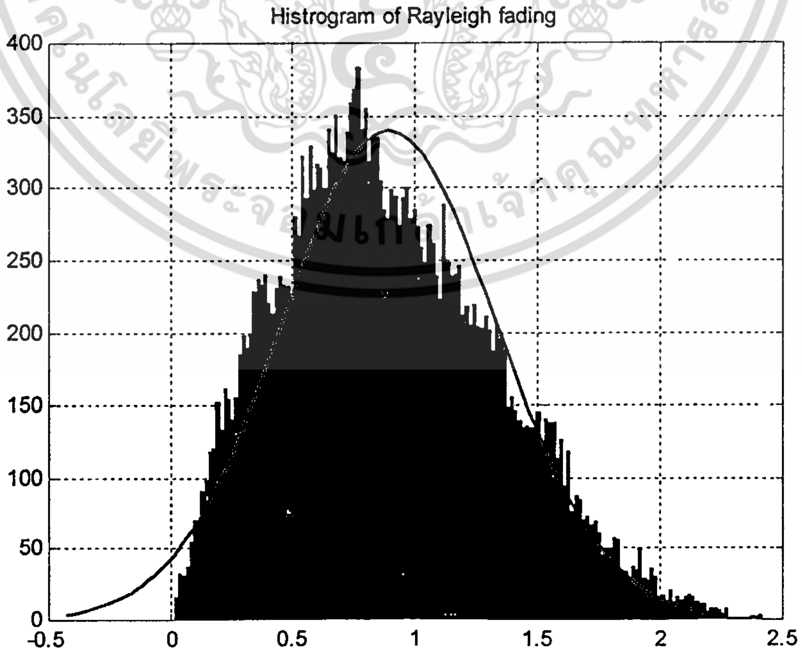
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. การออกแบบช่องสัญญาณแบบมัลติพาธ เรย์ลี

นิยามของช่องสัญญาณแบบมัลติพาธ คือ ผลคูณของตัวแปรแบบสุ่มที่มีการกระจายตัวแบบเกาส์เซี่ยน จำนวน 2 ตัว ซึ่งสามารถสร้างในรูปแบบของโปรแกรมมิ่ง โดยใช้โมเดลของ Jake's มาทำการสร้างได้โดยสามารถแสดงรูปแบบของการกระจายตัวแบบ Rayleigh ได้ดังรูป




รูป 3.22 (a) เอลเวลโลปสัญญาณมัลติพาธ




รูป 3.22 (b) ฮิสโตรแกรมของสัญญาณมัลติพาธ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.22 (a) แสดงเอลเวลโลปของช่องสัญญาณ Rayleigh fading ที่นำมาใช้เป็นตัวทดสอบประสิทธิภาพของระบบ DS-CDMA มาตรฐาน IS-95 โดยกำหนดให้จำนวนข้อมูลมีค่าเท่ากับจำนวนบิทข้อมูลที่ออกจากเครื่องส่ง ซึ่งมีค่าเท่ากับ 24576 (เท่ากับข้อมูลที่ผ่านการเสปรดด้วย PN) (b) แสดงฮิสโตรแกรมของ Rayleigh fading channel ที่จำลองขึ้นจากรูป (a) พารามิเตอร์ของ Rayleigh fading channel ที่ใช้แบบจำลองของ Jake ในการสร้างมีพารามิเตอร์ดังนี้

| **พารามิเตอร์** | **Value** |
|---|---|
| **จำนวนออสซิลเลเตอร์** | $M = 8$ |
| จำนวนเส้นทาง | $N=(M*2)+1$=17 |
| ความถี่คลื่นพาห์ | 840 MHz |
| ความเร็วของโมบายล์ | 60ไมล์/ชั่วโมง |

หมายเหตุ : ค่าพารามิเตอร์ที่มีความสำคัญของช่องสัญญาณแบบ Raleigh คือ

1. ค่าของจำนวนของออสซิลเลเตอร์ $M$
2. จำนวนของเส้นทางของสัญญาณ มีความสัมพันธ์กันตามสมการ

$$M = \frac{1}{2}(N-1)$$

เมื่อ $M$ คือค่าของจำนวนของออสซิลเลเตอร์ และ N คือค่าของจำนวนเส้นทางของสัญญาณ

3. ค่าของความถี่ ดรอปเปอร์ $f_d$ : มีค่าเท่ากับ

$$f_d = \frac{v}{c}\cos\theta_n$$

เมื่อ $f_d$ คือค่าของความถี่ดรอปเปอร์

$v$ คือค่าของความเร็วของโมบายล์

$c$ คือค่าของความเร็วแสง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 4
# ผลการวิจัย

ในหัวข้อนี้จะทำการจำลองการทำงานของระบบ DS-CDMA มาตรฐาน IS-95 ในทิศทางจากโมบายล์ไปยังสถานีฐาน โดยพารามิเตอร์ที่ใช้งานที่มีความสำคัญในการจำลองการทำงานของระบบ ในแต่ละหัวข้อ จะมีอยู่ด้วยกันอยู่ 3 ตาราง ดังนี้

**ตารางที่ 4.1** พารามิเตอร์ของชุดเข้ารหัสเทอร์โบ

| พารามิเตอร์ | ค่าพารามิเตอร์ |
|---|---|
| อัตราการเข้ารหัส | 1/3 |
| เจนเนอร์เรเตอร์ | จำนวน 3 ชุด<br>g1=1<br>g2=21<br>g3=37 |
| ความยาวของชุดหน่วยความจำภายใน | M=5 |
| จำนวนสเตจในการเข้ารหัส | 16 |
| อัลกอริทึมในการถอดรหัส | MAP |
| Interleaver | Random |

**ตารางที่ 4.2** พารามิเตอร์ของชุดเข้ารหัสคอนโวลูชั่น

| พารามิเตอร์ | ค่าพารามิเตอร์ |
|---|---|
| อัตราการเข้ารหัส | 1/3 |
| เจนเนอร์เรเตอร์ | จำนวน 3 ชุด<br>g1=557<br>g2=663<br>g3=711 |
| ความยาวของชุดเข้ารหัส | L=9 |
| จำนวนสเตจในการเข้ารหัส | 256 |
| ชุดถอดรหัส | ไวตาบิ แบบ Soft decision |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

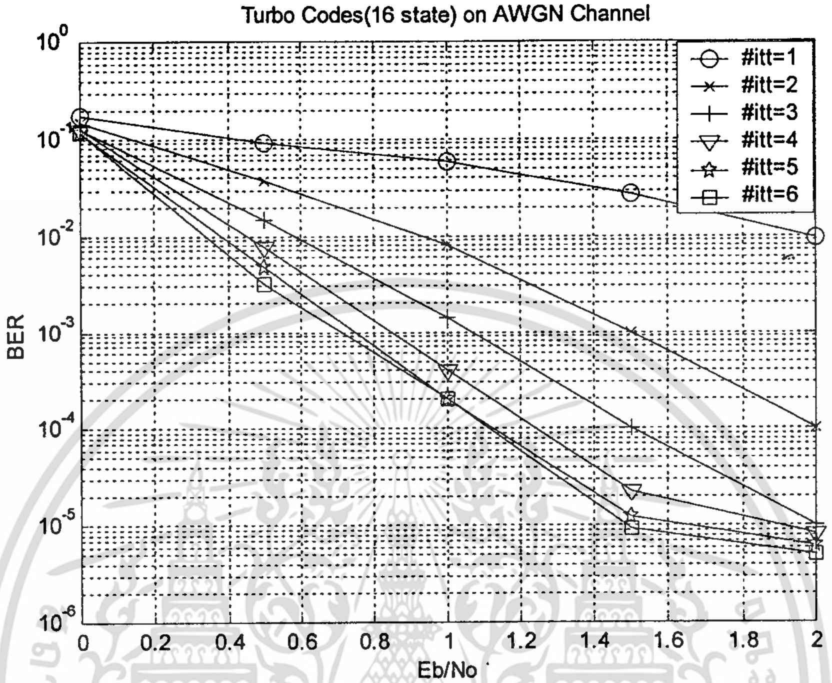
**ตารางที่ 4.3** พารามิเตอร์ของระบบ DS-CDMA (มาตรฐาน IS-95)

| พารามิเตอร์ | ค่าพารามิเตอร์ |
|---|---|
| Uplink frequency band | 824-849 MHz |
| Downlink frequency band | 869-894 MHz |
| Number of carrier/band | 20 |
| Bandwidth/carrier | 1.25 MHz |
| Multiple-access, method | CDMA |
| Number of user/carrier | 60 |
| Chip rate | 1.2288 Mcps |
| Speech coder | Variable rate CELP |
| Speech rate | 9600,4800,2400,1200 bps |
| Channel encoder Method | Convolution Code<br>R=1/2, L=9(D)<br>R=1/3,L=9(U) |
| Interleaver | Block |
| Modulation | BPSK with QPSK<br>Spreading(D)<br>64-ary Orthogonal with QPSK<br>Spreading(U) |
| Signature sequence | Hadamard(Walsh) of length 64 |
| PN sequence | N=2^42-1(Long code)<br>N=2^15-1(Spreading Code) |

1. ส่วนแรก จะทำการวิเคราะห์ระบบการเข้ารหัสข้อมูลแบบเทอร์โบ โดยจะทำการจำลองการทำงานของระบบโดยทำการเปลี่ยนแปลงพารามิเตอร์ต่าง ๆ ในระบบการเข้ารหัสแบบเทอร์โบ ซึ่งส่วนแรกจะทำการวิเคราะห์ประสิทธิภาพของระบบเข้ารหัสแบบเทอร์โบ โดยแปรเปลี่ยนจำนวนรอบของการเข้ารหัสแบบเทอร์โบ ส่วนที่สอง ทำการจำลองการทำงานของระบบเข้ารหัสแบบเทอร์โบ โดยเปลี่ยนแปลงขนาดของเฟรมข้อมูล ให้มีขนาดต่าง ๆ ส่วนที่สาม ทำการจำลองการทำงานระบบเข้ารหัสข้อมูลแบบเทอร์โบ โดยเปลี่ยนรูปแบบการสลับข้อมูล (Interleaver)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

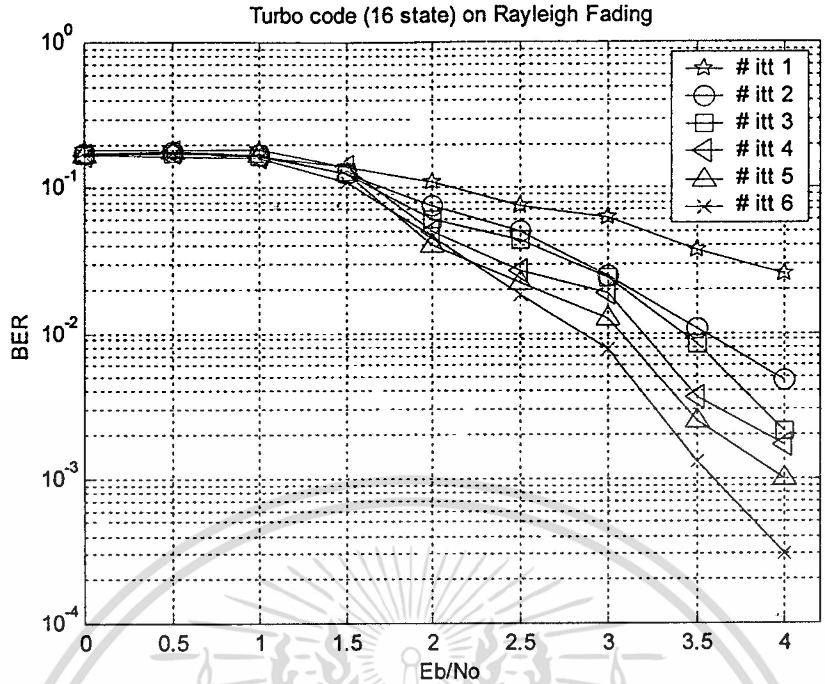
1.1 วิเคราะห์ประสิทธิภาพของระบบเข้ารหัสแบบเทอร์โบ โดยใช้เจนเนอร์เรเตอร์ ขนาด (1,21/35) โดยใช้อัลกอริทึมแบบ MAP ในการถอดรหัส โดยทำการเปลี่ยนแปลงจำนวนรอบของการถอดรหัส




**รูปที่ 4.1** ประสิทธิภาพของระบบการเข้ารหัสเทอร์โบบนช่องสัญญาณ AWGN data frame=580 บิท

จากรูปที่ 4.1 จะเห็นว่าประสิทธิภาพของชุดเข้ารหัสแบบเทอร์โบบนช่องสัญญาณ AWGN จะให้ประสิทธิภาพออกมาดีเมื่อเพิ่มจำนวนรอบของการตัดสินใจให้มีค่ามากขึ้น โดยจะเห็นความแตกต่างมาก ๆ ที่จำนวนรอบน้อย ๆ และจากรูปจะเห็นว่าช่วงจำนวนรอบที่ 4-6 ประสิทธิภาพจะมีความแตกต่างกันน้อยมากประมาณ 0.5 dB ซึ่งเราสามารถเลือกจำนวนรอบที่ 4 ไปใช้ได้เพราะประสิทธิภาพใกล้เคียงกับจำนวนรอบที่ 5 และ 6 แต่เวลาในการตัดสินใจข้อมูลจะลดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

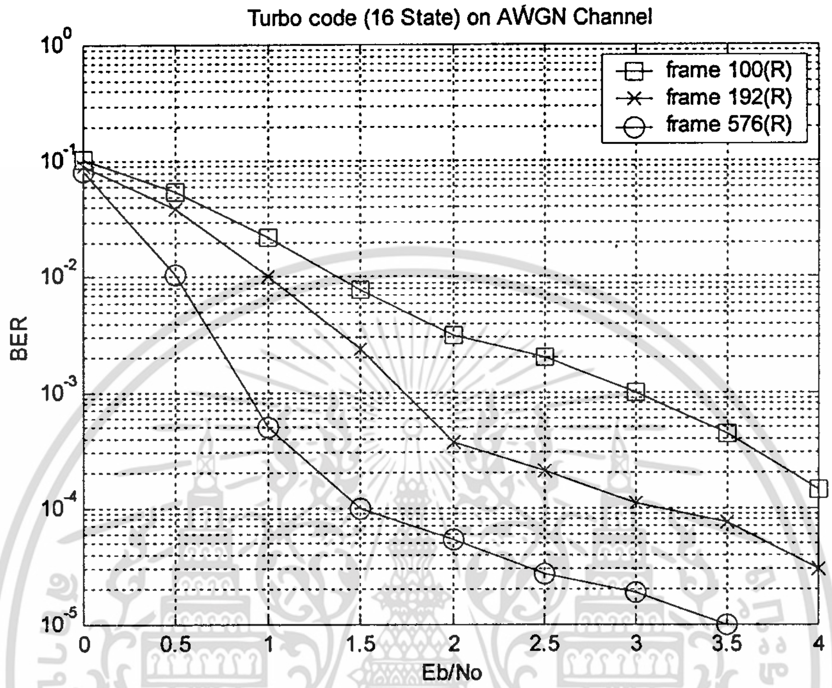



**รูปที่ 4.2** ประสิทธิภาพของระบบเข้ารหัสแบบเทอร์โบโดยเปลี่ยนแปลงจำนวนรอบ บนช่องสัญญาณ มัลติพาธ เรย์ลี data frame=580 บิท

จากรูปที่ 4.2 จะเห็นว่าประสิทธิภาพของชุดเข้ารหัสแบบเทอร์โบ ในช่องสัญญาณมัลติพาธ จะให้ประสิทธิภาพออกมาดีเมื่อเพิ่มจำนวนรอบของการตัดสินใจให้มีค่ามากขึ้น เช่นเดียวกันกับในช่องสัญญาณ AWGN จะมีความแตกต่างกันของจำนวนรอบที่ 4, 5 และ 6 ประมาณ 0.5 dB แต่กราฟแสดงประสิทธิภาพของระบบบนช่องสัญญาณมัลติพาธ จะได้ประสิทธิภาพแย่กว่า AWGN แต่ระบบเข้ารหัสแบบเทอร์โบสามารถที่จะเพิ่มประสิทธิภาพของระบบได้ดีเช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

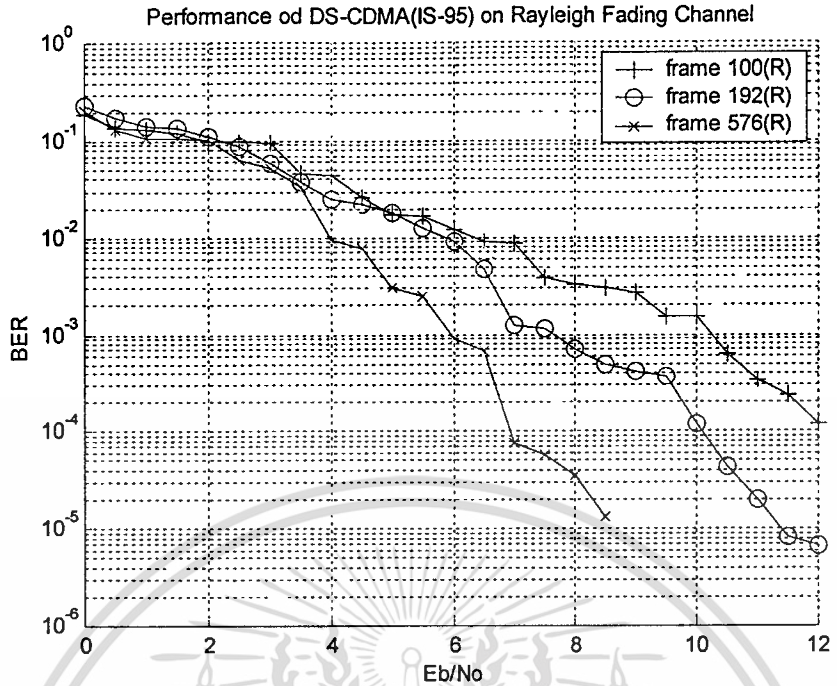
1.2 วิเคราะห์ประสิทธิภาพของระบบเทอร์โบ ซึ่งจะดูผลกระทบของการเปลี่ยนแปลงขนาดของเฟรมข้อมูล โดยจะทำการแปรเปลี่ยนขนาดของเฟรมข้อมูลเท่ากับ 100, 192 (มาตรฐาน IS-95), 576 บิทต่อเฟรม ซึ่งจะได้ผลการทดลองออกมาดังนี้




รูปที่ 4.3 ประสิทธิภาพของระบบเข้ารหัสแบบเทอร์โบ เมื่อเปลี่ยนแปลงขนาดของเฟรมข้อมูล บนช่องสัญญาณ AWGN จำนวน Iteration=6

จากรูปที่ 4.3 ใช้ชุดเข้ารหัสแบบเทอร์โบ ซึ่งใช้การสลับตำแหน่งของข้อมูลเป็นแบบ Random จะเห็นได้ว่าประสิทธิภาพของระบบชุดเข้ารหัสแบบเทอร์โบ เมื่อทำการเพิ่มขนาดของเฟรมให้ขนาดของข้อมูลที่มีความยาวมากขึ้นจะเห็นได้ว่าประสิทธิภาพของระบบจะดีขึ้น เนื่องมาจากว่า ในโครงสร้างของชุดเข้ารหัสจะมีส่วนที่ทำการสลับตำแหน่งของข้อมูล (Interleaver) เมื่อทำการสลับตำแหน่งของข้อมูลในแต่ละเฟรมจะทำให้ลดการผิดพลาดของข้อมูลที่มีความต่อเนื่องกัน ซึ่งทางด้านปลายทางสามารถที่จะแก้ไขได้ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

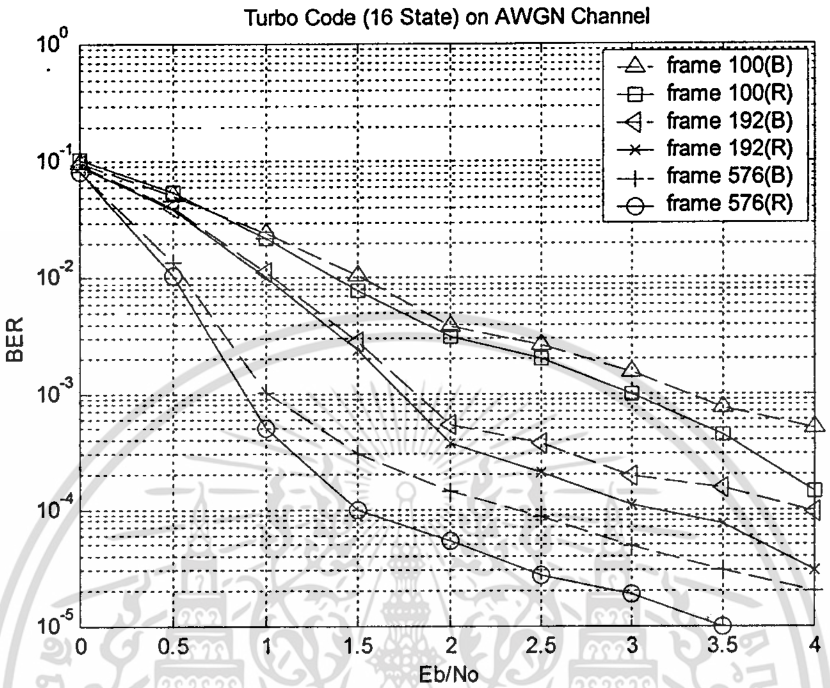



รูปที่ 4.4 ประสิทธิภาพของระบบเข้ารหัสแบบเทอร์โบ เมื่อเปลี่ยนแปลงขนาดของเฟรมข้อมูล บนช่องสัญญาณมัลติพาธ เรย์ลี

จากรูปเมื่อทำการเปลี่ยนแปลงขนาดของเฟรมข้อมูล จะเห็นว่าประสิทธิภาพของระบบที่ใช้เฟรมข้อมูลขนาดใหญ่ จะให้ประสิทธิภาพดีกว่า ประสิทธิภาพของระบบชุดเข้ารหัสแบบเทอร์โบที่ใช้ขนาดข้อมูลขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 วิเคราะห์ประสิทธิภาพของระบบเข้ารหัสแบบเทอร์โบ โดยทดลองเปลี่ยนแปลงวิธีการสลับตำแหน่งข้อมูล (Interleave) แบบ Random และแบบ Block



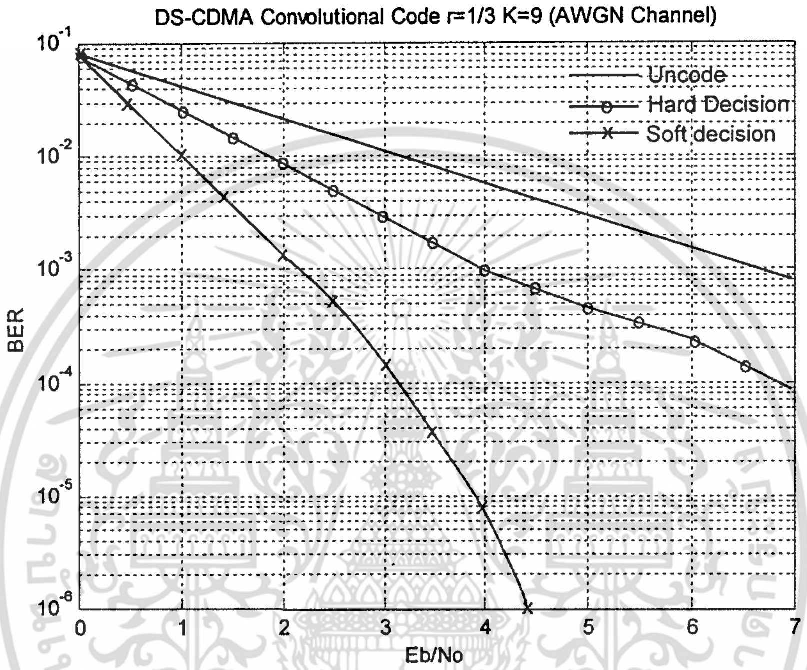

**รูปที่ 4.5** ประสิทธิภาพของระบบเทอร์โบที่ใช้รูปแบบการทำการสลับตำแหน่งข้อมูล แบบ Random และแบบ Block จำนวน Iteration=6

จากรูปแสดงการเปรียบเทียบประสิทธิภาพของระบบเข้ารหัสแบบเทอร์โบที่ใช้ขนาดของข้อมูล และรูปแบบการทำอินเทอร์ลีฟเวอร์แบบ Block และ Random โดยจะเห็นได้ว่าประสิทธิภาพของระบบที่ใช้ข้อมูลที่มีความยาวมาก ๆ และใช้รูปแบบการทำอินเทอร์ลีฟเวอร์แบบ Random จะให้ประสิทธิภาพออกมาดี ฉะนั้นในการจำลองการทำงานจะใช้รูปแบบการทำอินเทอร์ลีฟเวอร์ของชุดเข้ารหัสเทอร์โบเป็นแบบ Random

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การวิเคราะห์ประสิทธิภาพของระบบ DS-CDMA มาตรฐาน IS-95 (ในทิศทาง Reverse Link) แบบเดิมที่ใช้รูปแบบการเข้ารหัสข้อมูลแบบคอนโวลูชั่นโค้ด ซึ่งพารามิเตอร์ตามมาตรฐานที่ใช้ใน IS-95 จะใช้อัตราการเข้ารหัสข้อมูลเท่ากับ 1/3 เจนเนอร์เรเตอร์ g1=557 , g2=663 , g3=711

2.1 แสดงประสิทธิภาพของระบบ DS-CDMA มาตรฐาน IS-95 โดยใช้การเข้ารหัสแบบคอนโวลูชั่น ขนาดของความยาวของชุดเข้ารหัสเท่ากับ 9 ใช้อัลกอริทึมแบบ Soft decision ในการถอดรหัส

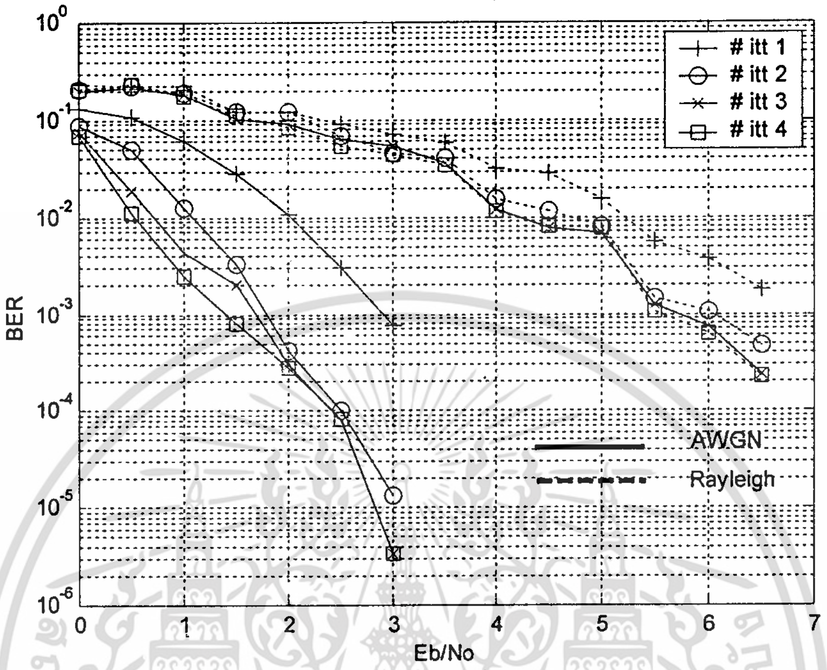



รูปที่ 4.6 ประสิทธิภาพของระบบ DS-CDMA มาตรฐาน IS-95 โดยใช้การเข้ารหัสแบบคอนโวลูชั่น

จากรูปเป็นไปตามทฤษฎี ที่ประสิทธิภาพของการถอดรหัสข้อมูลที่ใช้รูปแบบการถอดรหัสแบบ Soft decision จะให้ประสิทธิภาพออกมาดีกว่า Hard decision และจากรูปเราใช้ชุดเข้ารหัสคอนโวลูชั่น ที่มีความยาวของชุดเข้ารหัสเท่ากับ 9 ซึ่งมีจำนวนเสตจในการเข้ารหัสถึง 512 สเตจ ซึ่งเป็นชุดเข้ารหัสที่มีความยุ่งยากมาก แต่จะให้ประสิทธิภาพออกมาดี ซึ่งในการจำลองการทำงานในระบบ DS-CDMA มาตรฐาน IS-95 ได้ใช้ชุดเข้ารหัสที่มีความยาวของชุดเข้ารหัสเท่ากับ 9 เช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 วิเคราะห์ประสิทธิภาพของระบบ DS-CDMA มาตรฐาน IS-95 เมื่อใช้การเข้ารหัสแบบเทอร์โบ โดยทดลองเปลี่ยนแปลงจำนวนรอบของการตัดสินใจของเทอร์โบโค้ดบนช่องสัญญาณ AWGN โดยค่าพารามิเตอร์ของชุดเข้ารหัสแบบเทอร์โบ เป็นไปดังตารางที่ 4.1

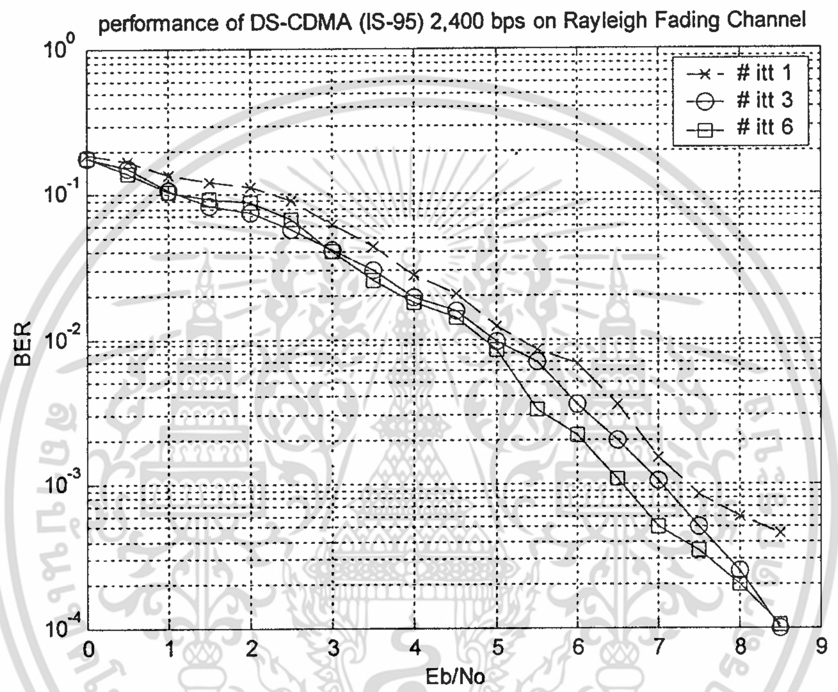



**รูปที่ 4.7** ประสิทธิภาพของระบบ DS-CDMA มาตรฐาน IS-95 โดยใช้การเข้ารหัสแบบเทอร์โบ โดยเปลี่ยนแปลงจำนวนรอบของการถอดรหัสข้อมูล

จากรูปที่ 4.7 ช่องสัญญาณแบบเรย์ลี มี $F_c$=840 MHz ตามมาตรฐาน IS-95 และขนาดของอินเทอร์ลีฟเวอร์ภายในแบบ Random ขนาด 192 บิท และอินเทอร์ลีฟเวอร์ภายนอกขนาด 600 บิท เป็นแบบ Block ผลที่ได้จะเห็นว่าประสิทธิภาพของระบบ DS-CDMA มาตรฐาน IS-95 จะดีขึ้นเมื่อใช้ระบบการเข้ารหัสแบบเทอร์โบ

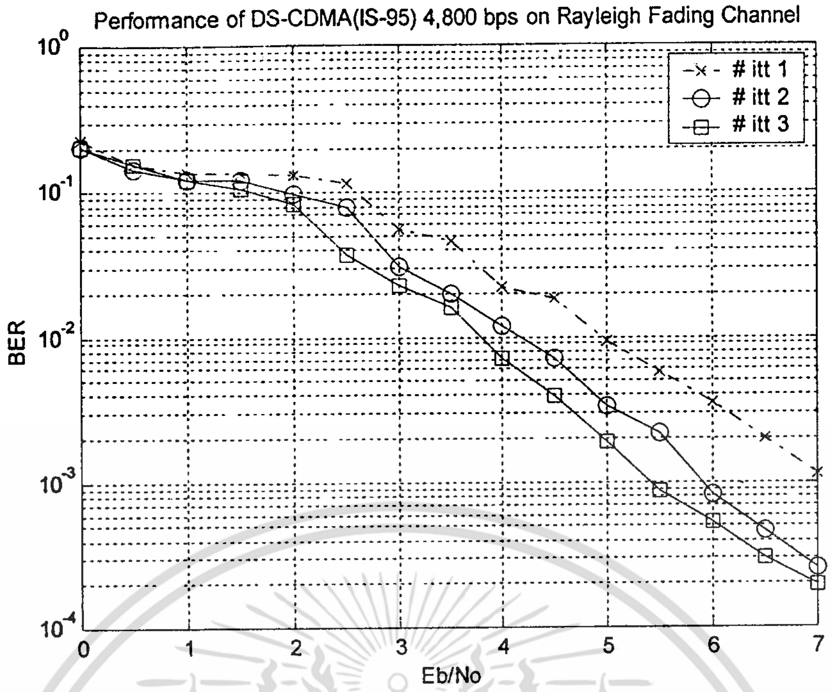
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 หลังจากในหัวข้อ 1.2 ได้ทำการวิเคราะห์ผลของการเปลี่ยนแปลงขนาดของเฟรมข้อมูลของการเข้ารหัสแบบเทอร์โบ แล้วจะเห็นว่าประสิทธิภาพของระบบที่ใช้ขนาดของเฟรมข้อมูลที่มีขนาดใหญ่จะให้ประสิทธิภาพออกมาดีกว่าระบบที่ใช้ขนาดของเฟรมข้อมูลที่มีขนาดเล็ก ในส่วนนี้เราจะทำการเปลี่ยนแปลงขนาดของเฟรมข้อมูลเช่นเดียวกับข้อ 1.2 แต่จะวิเคราะห์ในระบบ DS-CDMA มาตรฐาน IS-95 ที่ใช้การเข้ารหัสแบบเทอร์โบ ที่มีพารามิเตอร์ดังในตารางที่ 4.1

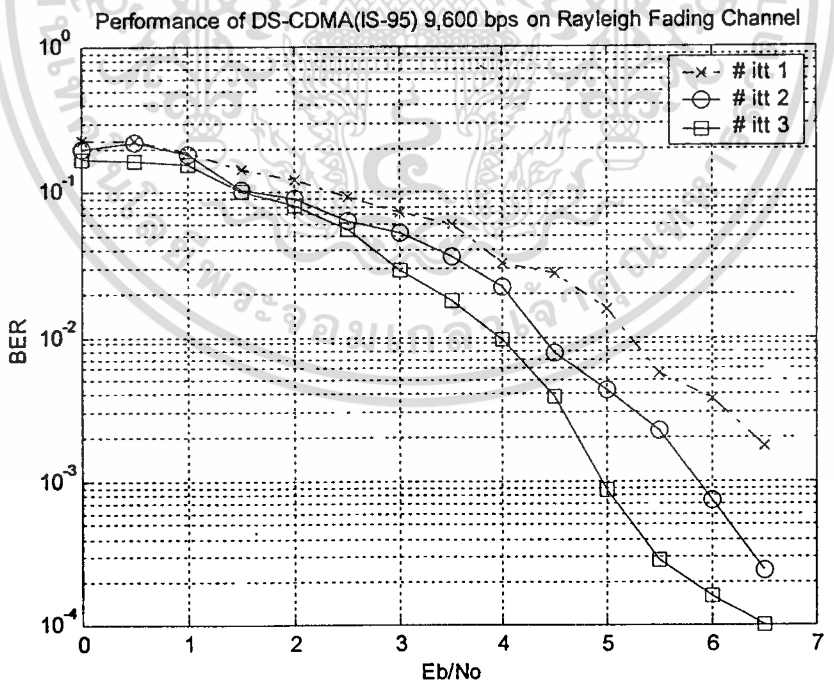



รูปที่ 4.8 ประสิทธิภาพของระบบ DS-CDMA เมื่อใช้การเข้ารหัสแบบเทอร์โบ โดยเปลี่ยนแปลงอัตราการส่งข้อมูลที่ 9600 bps บนช่องสัญญาณมัลติพาธเรย์ลี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.9** ประสิทธิภาพของระบบ DS-CDMA เมื่อใช้การเข้ารหัสแบบเทอร์โบ โดยเปลี่ยนแปลงอัตราการส่งข้อมูลที่ 4800 bps บนช่องสัญญาณมัลติพาธเรย์ลี

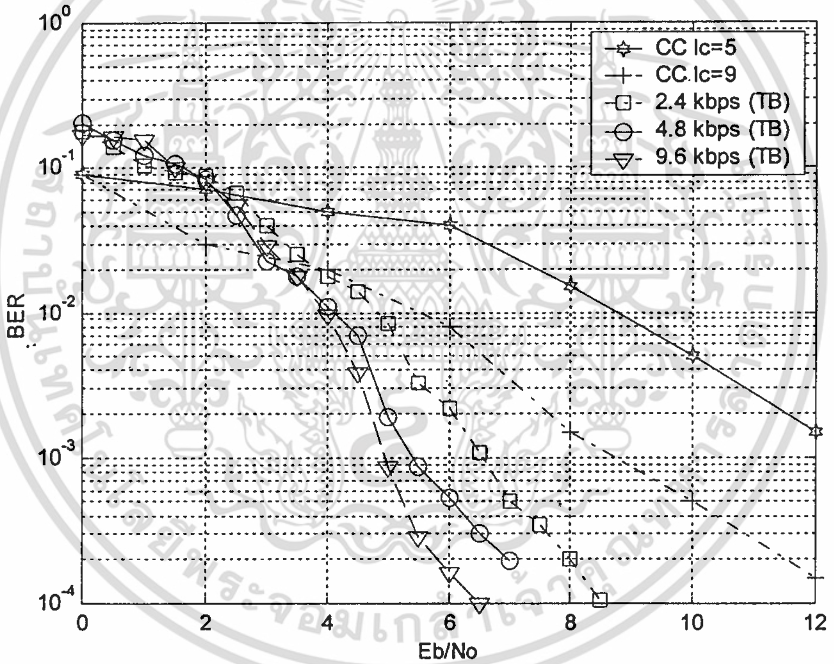



**รูปที่ 4.10** ประสิทธิภาพของระบบ DS-CDMA เมื่อใช้การเข้ารหัสแบบเทอร์โบ โดยเปลี่ยนแปลงอัตราการส่งข้อมูลที่ 2400 bps บนช่องสัญญาณมัลติพาธเรย์ลี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปทั้ง 3 เมื่อทำการส่งข้อมูลที่อัตราการส่งข้อมูลที่แตกต่างกัน คือ 9600 bps , 4800 bps และ 2400 bps จะเห็นว่าในระบบ DS-CDMA มาตรฐาน IS-95 จะให้ประสิทธิภาพออกมาไม่แตกต่างกัน หรือพูดได้ว่าประสิทธิภาพของระบบ DS-CDMA มาตรฐาน IS-95 จะให้ประสิทธิภาพของระบบออกมาเท่ากันทุกอัตราการส่งข้อมูลที่เป็นมาตรฐานของระบบ ซึ่งไม่ว่าเราจะเลือกส่งที่ความเร็วในการส่งข้อมูลเท่าใดก็จะให้ประสิทธิภาพออกมาเท่ากัน

2.4 วิเคราะห์ประสิทธิภาพของระบบ DS-CDMA มาตรฐาน IS-95 เมื่อใช้การเข้ารหัสแบบคอนโวลูชั่น เทียบกับการเข้ารหัสแบบเทอร์โบ (ที่นำเสนอ) โดยพารามิเตอร์ของชุดเข้ารหัสแบบเทอร์โบใช้เจนเนอร์เรเตอร์ (1,21/37) โดยชุดเข้ารหัสแบบคอนโวลูชั่น ใช้พารามิเตอร์เช่นเดียวกับ ข้อ 2.1 แสดงประสิทธิภาพได้ดังนี้



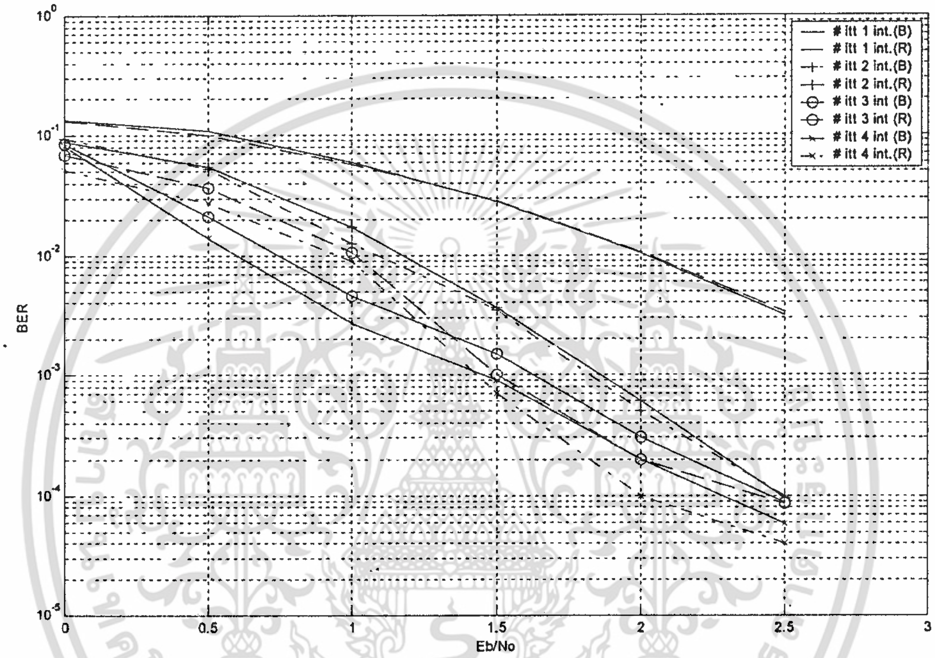

**รูปที่ 4.11** ประสิทธิภาพของระบบ DS-CDMA มาตรฐาน IS-95 โดยใช้การเข้ารหัสแบบเทอร์โบ เทียบกับคอนโวลูชั่น

จากรูปได้ทำการเปรียบเทียบประสิทธิภาพของรูปแบบการเข้ารหัส ในระบบ DS-CDMA มาตรฐาน IS-95 ซึ่งใช้การเข้ารหัสข้อมูลตามมาตรฐาน ในตารางที่ 4.3 และระบบ DS-CDMA มาตรฐาน IS-95 ที่ใช้การเข้ารหัสแบบเทอร์โบ (ที่นำเสนอ) โดยมีพารามิเตอร์ของชุดเข้ารหัสดังตารางที่ 4.1 จะเห็นว่าประสิทธิภาพของระบบ DS-CDMA ที่ใช้การเข้ารหัสแบบเทอร์โบจะให้ประสิทธิภาพออกมาดีกว่าระบบเดิมที่ใช้การเข้ารหัสแบบคอนโวลูชั่น โดยสามารถเพิ่ม

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ห้ามมิให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประสิทธิภาพของระบบ DS-CDMA ที่ใช้การเข้ารหัสแบบเทอร์โบ ด้วยการเปลี่ยนแปลงจำนวนรอบของการเข้ารหัส และเปลี่ยนแปลงรูปแบบการทำอินเทอร์ลีฟเวอร์ภายในชุดเข้ารหัส ดังที่จำลองการทำงานในหัวข้อที่ 1 ซึ่งจะให้ประสิทธิภาพของระบบออกมาดีขึ้น

2.5 วิเคราะห์ประสิทธิภาพของระบบ DS-CDMA มาตรฐาน IS-95 โดยใช้การเข้ารหัสแบบเทอร์โบ โดยเปลี่ยนรูปแบบของการสลับข้อมูลบนช่องสัญญาณ AWGN




**รูปที่ 4.12** ประสิทธิภาพของระบบ DS-CDMA เมื่อเปลี่ยนรูปแบบการสลับข้อมูลของอินเทอร์ลีฟเวอร์ภายนอก แบบ Block และ Random

ผลที่ได้จะเห็นว่า ประสิทธิภาพของระบบ DS-CDMA มาตรฐาน IS-95 เมื่อเปลี่ยนแปลงรูปแบบการสลับตำแหน่งข้อมูลภายนอกนั้น มีผลน้อยเมื่อเทียบระหว่างการทำอินเทอร์ลีฟเวอร์ภายนอกแบบ Block และการทำอินเทอร์ลีฟเวอร์ภายนอกแบบ Random ซึ่งตามมาตรฐานของ IS-95 จะใช้รูปแบบการทำอินเทอร์ลีฟเวอร์ข้อมูลภายนอกเป็นแบบ Block ฉะนั้นในระบบ DS-CDMA มาตรฐาน IS-95 รูปแบบในการทำอินเทอร์ลีฟเวอร์ภายนอกจะให้ผลออกมาไม่แตกต่างกันเท่าไหร่ ฉะนั้น ในการจำลองการทำงานที่ผ่านมาได้กำหนดรูปแบบการทำอินเทอร์ลีฟเวอร์ภายนอก ตามมาตรฐาน IS-95 โดยไม่จำเป็นต้องเปลี่ยนรูปแบบการทำอินเทอร์ลีฟเวอร์ ผลที่ได้นี้แตกต่างกับอินเทอร์ลีฟเวอร์ภายในชุดเข้ารหัสเทอร์โบ ประสิทธิภาพจะได้ออกมาดีที่สุดเมื่อใช้การทำอินเทอร์ลีฟเวอร์แบบ Random ในชุดเข้ารหัสเทอร์โบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5
# สรุปผลการทดสอบ

## 5.1 สรุปผลการวิจัย

ผลการจำลองการทำงานในบทที่ 4 ที่ได้ทำการวิจัย สรุปรายละเอียดได้ดังนี้

ในรูปที่ 4.1 ถึง 4.5 จะเห็นว่าในระบบของการเข้ารหัสแบบเทอร์โบ ในช่องสัญญาณแบบ AWGN และแบบมัลติพาธ เรย์ลีประสิทธิภาพของระบบที่เข้ารหัสแบบเทอร์โบ จะให้ประสิทธิภาพออกมาดี เมื่อใช้ขนาดของการทำอินเทอร์ลีฟเวอร์ที่มีขนาดใหญ่ และจำนวนรอบของการถอดรหัสข้อมูลมีจำนวนรอบที่มากขึ้น ยิ่งจำนวนรอบในการถอดรหัสมีจำนวนมากเท่าไหร่ยิ่งทำให้ประสิทธิภาพในระบบออกมาดี ทั้งในช่องสัญญาณ AWGN และแบบมัลติพาธ เรย์ลี และรูปแบบในการทำอินเทอร์ลีฟข้อมูลในชุดเข้ารหัสแบเทอร์โบ แบบ Random จะให้ประสิทธิภาพออกมาดีกว่าการทำอินเทอร์ลีฟข้อมูลแบบ Block

ในรูปที่ 4.6 แสดงประสิทธิภาพของระบบ DS-CDMA มาตรฐาน IS-95 ที่ใช้การเข้ารหัสแบบคอนโวลูชั่น เปรียบเทียบประสิทธิภาพของอัลกอริทึมในการถอดรหัส จะเห็นได้ว่าอัลกอริทึมในการถอดรหัสข้อมูลแบบ Soft Decision จะให้ประสิทธิภาพออกมาดีกว่าแบบ Hard Decision ซึ่งเราจะใช้ระบบ DS-CDMA ที่ใช้การเข้ารหัสแบบคอนโวลูชั่นถอดรหัสแบบ Soft Decision ในการจำลองการทำงานเพื่อใช้ในการเปรียบเทียบกับระบบ DS-CDMA ที่ใช้ชุดเข้ารหัสแบบเทอร์โบ (ที่นำเสนอ)

รูปที่ 4.7 แสดงประสิทธิภาพของระบบ DS-CDMA มาตรฐาน IS-95 ซึ่งเราพิจารณามาตรฐานในการส่งทิศทางจากโทรศัพท์เคลื่อนที่ไปยังสถานีฐาน จากรูปแสดงให้เห็นว่าประสิทธิภาพของระบบ DS-CDMA เมื่อใช้การเข้ารหัสแบบเทอร์โบ ประสิทธิภาพของระบบสามารถเปลี่ยนแปลงได้ โดยเพิ่มจำนวนรอบของการถอดรหัสของชุดถอดรหัสให้มีจำนวนรอบมากขึ้น โดยที่จากการจำลองการทำงานได้ทำการเปลี่ยนแปลงจำนวนรอบจาก 1 ถึง 6 ในช่องสัญญาณ AWGN และช่องสัญญาณมัลติพาธ จะเห็นได้ว่าประสิทธิภาพของระบบ DS-CDMA สามารถเปลี่ยนแปลงได้โดย ทำการเปลี่ยนจำนวนรอบของการตัดสินใจของชุดถอดรหัสให้มีจำนวนรอบมากขึ้น ยิ่งจำนวนรอบมากก็ยิ่งทำให้ประสิทธิภาพระบบ DS-CDMA ยิ่งดีขึ้น และจำนวนรอบที่มีความเหมาะสม คือจำนวนรอบที่ 4 ซึ่งในจำนวนรอบที่ 5 และ 6 ให้ประสิทธิภาพออกมาไม่ค่อยแตกต่างกันมากนัก ฉะนั้นจำนวนรอบที่ 4 มีความเหมาะสม เนื่องจากเวลาในการตัดสินข้อมูลของชุดถอดรหัสจะลดลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 4.8 ถึง 4.11 จำลองการทำงานโดยเปลี่ยนแปลงอัตราการส่งข้อมูลตามมาตรฐาน IS-95 คือ ที่ 9600 บิทต่อวินาที , 4800 บิทต่อวินาที และ 2400 บิทต่อวินาที โดยจำนวนบิทที่เข้าชุดเข้ารหัสจะมีขนาดของเฟรมเปลี่ยนไป ผลที่ได้จากการจำลองการทำงาน จะเห็นได้ว่าประสิทธิภาพของระบบ DS-CDMA ให้ผลที่ได้ออกมาไม่แตกต่างกันมากนัก เหตุผลอาจจะมาจากโครงสร้างของระบบ DS-CDMA ถ้าส่งข้อมูลต่ำกว่า 9600 บิท จะมีการส่งข้อมูลซ้ำ เช่นถ้าส่งที่ 4800 จะทำการส่งข้อมูลซ้ำ 2 ครั้ง และที่ 2400 บิท จะทำการส่งข้อมูลซ้ำ 4 ครั้ง ซึ่งเป็นการชดเชยการส่งข้อมูลที่มีขนาดของเฟรมข้อมูลที่มีขนาดเล็ก ซึ่งจะได้ผลของประสิทธิภาพออกมาไม่แตกต่างกันเท่าไหร่ ในการส่งข้อมูลที่อัตราการส่งที่แตกต่างกัน ฉะนั้นในการจำลองการทำงานเราจะใช้การส่งข้อมูลที่ 9600 บิทต่อวินาทีในการส่งข้อมูล

ในรูปที่ 4.12 แสดงประสิทธิภาพของระบบ DS-CDMA มาตรฐาน IS-95 ที่มีการเปลี่ยนแปลงรูปแบบของการทำอินเทอร์ลีฟเวอร์ภายนอก ซึ่งเป็นการทำอินเทอร์ลีฟเวอร์ข้อมูลหลังจากที่ผ่านการเข้ารหัสข้อมูล โดยทดลองเปลี่ยนแปลงรูปแบบของการทำอินเทอร์ลีฟแบบ Block ซึ่งเป็นแบบที่ใช้ในมาตรฐาน IS-95 เทียบกับแบบ Random ผลที่ได้จะให้ผลออกมาไม่แตกต่างกันมาก หรือจะพูดได้ว่าแทบไม่แตกต่างกันเลย ซึ่งต่างกับในชุดเข้ารหัสแบบเทอร์โบ ที่การทำอินเทอร์ลีฟเวอร์แบบ Block จะให้ประสิทธิภาพออกมาด้อยกว่าการทำอินเทอร์ลีฟข้อมูลแบบ Random โดยในระบบ DS-CDMA มาตรฐาน IS-95 ใช้การสลับตำแหน่งข้อมูลแบบ Block โดยถ้ามองในรูปแบบในทางปฏิบัติแล้ว วงจรการทำงานมีความซับซ้อนน้อยกว่า ฉะนั้น จึงไม่จำเป็นต้องเปลี่ยนรูปแบบการทำอินเทอร์ลีฟเวอร์ข้อมูล

จากการทดสอบทั้งหมดที่ได้ทำการทดสอบสามารถที่จะสรุปได้ ดังนี้

1. ระบบ DS-CDMA มาตรฐาน IS-95 เมื่อนำเอาระบบเข้ารหัสแบบเทอร์โบ เข้ามาใช้ในระบบจะให้ประสิทธิภาพออกมาดีกว่าระบบ DS-CDMA ที่ใช้การเข้ารหัสแบบคอนโวลูชั่น
2. ประสิทธิภาพของระบบ DS-CDMA ที่ใช้ชุดเข้ารหัสแบบเทอร์โบสามารถเพิ่มประสิทธิภาพได้โดยเพิ่มจำนวนรอบในการตัดสินใจของชุดถอดรหัส
3. พารามิเตอร์ต่าง ๆ ในระบบ DS-CDMA นอกเหนือจากชุดเข้ารหัส เมื่อทำการเปลี่ยนแปลงตามมาตรฐานจะไม่มีผลต่อประสิทธิภาพของระบบ หรือมีผลน้อยมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.2 ปัญหาและข้อเสนอแนะในการพัฒนางานวิจัยต่อไป

จากที่ได้ทำการทดสอบเพิ่มประสิทธิภาพให้กับระบบ DS-CDMA มาตรฐาน IS-95 โดยนำเอาชุดเข้ารหัสแบบเทอร์โบ มาใช้แทนชุดเข้ารหัสแบบคอนโวลูชั่น อุปสรรคในการวิจัยอยู่ที่การจำลองการทำงานของระบบทำได้ค่อนข้างยาก เนื่องจากถ้ามีความต้องการจำลองการทำงานของระบบทั้งหมด ผู้วิจัยขอแนะนำให้แยกการจำลองแบ่งออกเป็นส่วน ๆ เพื่อให้ง่ายต่อการตรวจสอบและแก้ไข ปัญหาอีกอย่างหนึ่งคือ เมื่อจำลองการทำงานของระบบใหญ่ ๆ จะใช้เวลาในการ Run Program นานมาก ฉะนั้นควรใช้เครื่องมือที่มีประสิทธิภาพ เช่นควรใช้ภาษา C หรือ JAVAในการจำลองการทำงาน ซึ่งเป็นภาษาที่ค่อนข้างมีความเหมาะสมเป็นอย่างมาก ผู้ที่มีความประสงค์นำโปรแกรมในวิทยานิพนธ์ไปพัฒนาในระบบอื่น ๆ เช่น OFDM,CDMA 2000, MC-CDMA

ในวิทยานิพนธ์ฉบับนี้ได้จำลองการทำงานระบบ DS-CDMA มาตรฐาน IS-95 บนช่องสัญญาณ AWGN และมัลติพาธ เรย์ลี โดยไม่ได้ทำการวิเคราะห์ผลของระบบหลายผู้ใช้ (Multi-User) ซึ่งสามารถที่จะนำไปพัฒนาต่อได้ โดยวิเคราะห์ระบบโดยพิจารณาในการใช้งานหลายผู้ใช้ โดยอาจจะทดลองเปลี่ยนพารามิเตอร์ต่าง ๆ ได้แก่ รูปแบบของ PN ทั้ง Long PN และ Short PN ที่ใช้ และทดลองดูผลที่เกิดขึ้นในระบบ ในรูปแบบ Multi-User

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# เอกสารอ้างอิง


[1] ลัญฉกร วุฒิสิทธิกุลกิจ. **หลักการระบบโทรศัพท์เคลื่อนที่.** กรุงเทพมหานคร : โรงพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย. 2542

[2] ลัญฉกร วุฒิสิทธิกุลกิจ. **เทคโนโลยีโทรคมนาคม ทฤษฎีข่าวสารและการเข้ารหัส.** กรุงเทพมหานคร : โรงพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย. 2546.

[3] ฟูศักดิ์ ชีวสุวิทย์. **การแก้รหัสที่ผิด.** กรุงเทพมหานคร : คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง. 2533

[4] มัลลิกา ศรีกมล. **ทฤษฎีรหัส.** เชียงใหม่. : คณะวิทยาศาสตร์ มหาวิทยาลัยเชียงใหม่. 2527

[5] V. K. Garg, **IS-95 CDMA and CDMA2000,.** USA, Prentice-Hall Publishers, 2000

[6] V. K. Grarg, **Wireless Network Evolution 2G to 3G,.** USA, Prentice-Hall Publishers, 2002.

[7] S. Glisic, B. Vucetic, **Spread Spectrum CDMA Systems for Wireless Communications,.** London, Artech House, Inc. 1997.

[8] B. Vucetic , J. Yuan, **Turbo Codes Principles and Applications,** USA, Kluwer Academic Publishers. Second Printing 2001.

[9] C. Berrou, A. Glavieux and P. Thitimajshima, “Communications 1993,ICC ’93 Geneva. Technical Program ,Conferences Recode, “Near Shannon limit error-correcting coding and decoding; Turbo-codes, “IEEE international Conference on ,Vol 2. 1993, pp 1064-1070 vol.2

[9] C. Berrou and A. Glavieux, “Near Optimum error correcting coding and Decoding:Turbo-codes, “IEEE Trans, On Volume:44 Issuse:10, Oct.1996,pp 1261-1271.

[10] B. Sklar, “ Turbo Code concepts made easy, or how I learned to concatenate and reiterate, “ Milcom 97 Proceeding,.Vol 1,1997,pp 1261-1271.

[11] B. Sklar, “ A Primer on Turbo Code Concepts”, “IEEE Communications Magazine, December 1997,pp 94-102

[12] V. K. Garg, K. Smolik and J. E. Wilkes, **Applications of CDMA in Wireless/Personal Commnunications,** UK, Prentice-Hall Publishers, 2002.

[13] N. Sone, M. Mohri, M. Mori and H. Sasano, “On Good Convolutional Codes with Optimal Free Distance for Rate ½,1/3 and ¼, “IEICE Trans. Commu., Vol.E84-B.,NO.1 Jan 2001,pp 116-119


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[14] J. Hagenauer and P. Hoeher , "A Viterbi Algorithm with Soft-Decision Outputs and Its Applications, "GLOBLECOM 1989, Dallas, Texas,pp 1680-1686, Nov. 1989.

[15] A. J. Viterbi, "Convolutional Codes and Their Performance in Communication Systems, "IEEE Transactions on Communications Technology, Vol. COM-19, No. 5, pp. 751-772, October 1971.

[16] J. G. Proakis, M. Salehi, **Communication system engineering 2nd** ,. USA, Prentice-Hall Publishers, 2002

[17] M. Y. Rhee , **Error Correcting Coding Theory,** New York, McGraw-Hill, 1989.

[18] B. Sklar, **Digital Communications Fundamentals and Applications 2nd** , New Jersey, Prentice-Hall Publishers, 2001

[19] G. D. Forney and E. K. Bower, "A High Speed Sequential Decoder: Prototype Design and Test, "IEEE Trans. Commun. Technol., Vol. COM19, no.5, Oct. 1971,pp. 821-835

[20] S. B. Wicker, **Error Control Systems for Digital Communication and Storage,** Englewood Cliffs : Prentice Hall, 1995.

[21] C. Berrou, P. Adde, E. Angui and S. Faudeli, " A Low Complexity soft-Output Viterbi Decoder Architecture, " Proceeding of ICC 1993, Geneva, Switzerland, pp. 737-740, May 1993.

[22] J. Hagenauer, P. Robertson and L. Papke, "Iterative ("Turbo") Decoding of Systematic Convolutional Codes with MAP and SOVA Algorithms, "Proceeding of ITG, pp. 21-29, Oct. 1994

[23] J. Hagenauer and L. Papke, "Decoding "Turbo " Codes with the Soft Output Viterbi Algorithm (SOVA)," Proceeding of International Symposium on Information Theory, Trondheim, Norway, p. 164, June1994.

[24] G. Battail, C. Berrou and A. Glavieux, "Psuedo-Random Recursive Convolutional Coding for Near-Capacity Performance," GLOBECOM 1993, pp.23-27, Dec. 1993.

[25] D. Divsalar and F. Pollara, "Turbo Codes for PCS Applications," Proceeding of ICC 1995, Seattle, WA., pp. 54-59 ,June 1995.

[26] S. Dolinar and D. Divsalar, "Weight Distributions for Turbo Codes Using Random and Nonrandom Permutations," JPL TDA Progress Report 42-122, Aug. 15, 1995.

[27] D. Divsalar and F. Pollara, "On the Design of Turbo Codes," TDA Progress rep. 42-123, Jet Propulsion Lab., Pasadena, CA, Nov.15, 1995, pp. 99-121

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[28] D.Divsalar and R. J. McEliece, "Effective Free Distances of Turbo Codes," Elect, Lett., Vol.32, no. 5, Feb. 29, 1996,pp. 56-65.

[29] C. Berrou and A. Glavieux, "Near Optimum Error Correcting Coding and Decoding:Turbo-Codes," IEEE Transactions on Communications, Vol. 44, No. 10, pp. 1261-1271, Oct. 10, 1996.

[30] L.R. Balh, J. Cocke, F. Jelinek and P. Raviv, "Optimal Decoding of linear codes for minimizing symbol error rate," IEEE Trans. Inform. Theory, Vol.IT-20,pp 284-287, March 1974.

[31] R. H. Morelos-Zaragoza, **The Art of Error Correcting Coding,** Canada, John Wiley & Son, LTD, 2002.

[32] P. Sweeney, **ERROR CONTROL CODING From Theory to Practice,** UK , John Wiley & Son, LTD, 2002.

[33] A.D. Hougton, **The Engineering's Error Coding Handbook,** London, Chapman & Hall, 1997.

[33] W. E. Ryan, "A turbo code tutorial,"http://www.ee.virginia.edu/research/CCSP/turbo_codes/tcodes-bib/turbo2c.ps

[34] L. Hanzo, T. H. Liew and B.L. Yeap, **Turbo Coding Turbo Equalisation and Space-Time Coding for Transmission over Fading Channels,** USA, John Wiley & Son, LTD, 2002.

[35] D. Spuler, **C++ and C efficiency: How to improve program speed and memory usage,** New Jersey, Prentice-Hall, 1969.

[36] D. Spuler, **Comprehensive C.**, New Jersey, Prentice-Hall, 1969.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

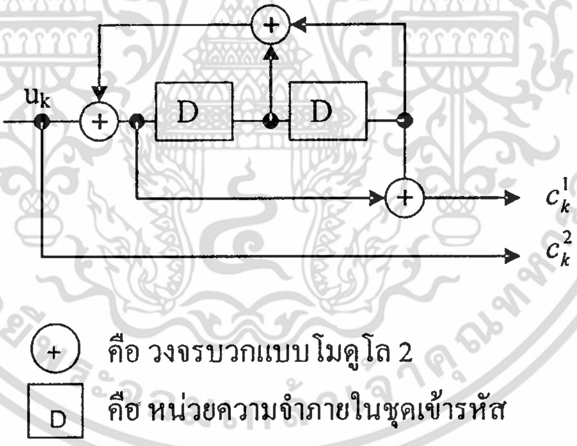
## ภาคผนวก ก

# อัลกอริทึมในการถอดรหัสเทอร์โบ

ในส่วนของหัวข้อนี้จะทำการอธิบายรายละเอียดของอัลกอริทึมในการทอดรหัสแบบ MAP ซึ่งมีรายละเอียดดังนี้

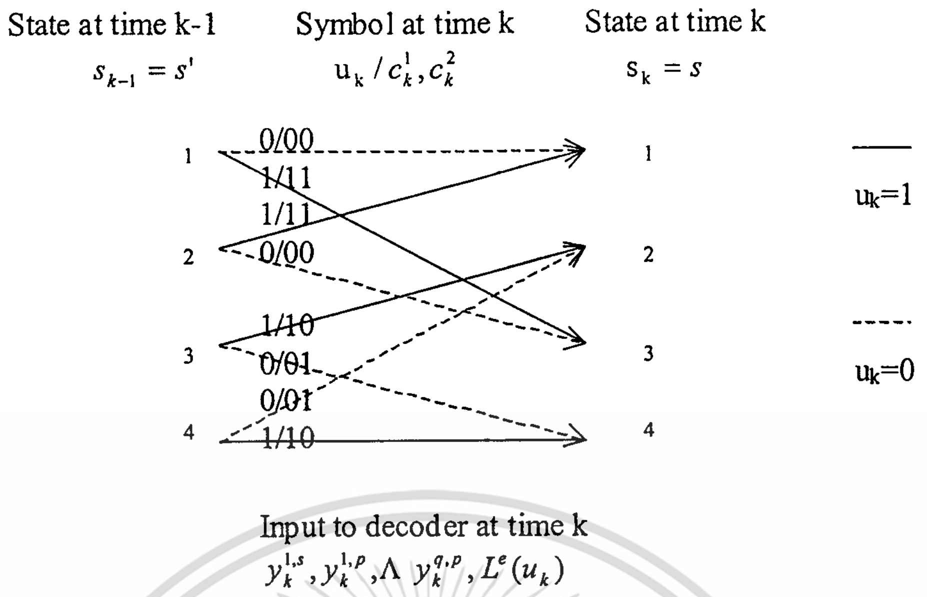
### ก.1 กล่าวนำ

ในรูป ก.1 และรูป ก.2 นั้นเป็นตัวอย่างที่แสดงความสัมพันธ์ระหว่างสัญญาณและไดอะแกรม Trellis ซึ่งในตัวอย่างที่ทำการแสดงนี้ ชุดถอดรหัสแบบ RSC มีค่าเจนเนอร์เรเตอร์ของชุดเข้ารหัส คือ G(D)= [1,05/07] ซึ่งเขียนอยู่ในรูปของเลขฐานแปด และมีขนาดของหน่วยความจำเท่ากับ 2 ณ ตำแหน่งของเวลาที่ k จะมีอินพุทที่ป้อนเข้ามายังชุดเข้ารหัสคือ $u_k$ $c_k$ คือ ค่าโค้ดซิมโบล $y_k$ คือค่าของข้อมูลบางส่วน และ $L^e(u_k)$ คือค่าของข้อความ priori ของ $u_k$ ซึ่งถูกจัดการจากชุดถอดรหัสชุดที่ผ่านมา




**รูปที่ ก.1** โครงสร้างภายในของชุดเข้ารหัส แบบ RSC G(D)= [1,05/07] ที่มีค่าความจุภายในเท่ากับ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ ก.2** ไดอะแกรม Trellis ของชุดเข้ารหัสแบบ RSC ซึ่งมีค่าเจนเนอร์เตอร์เท่ากับ G(D)= [1,05/07] ที่มีค่าความจุภายในเท่ากับ 2

ก่อนที่จะทำการแก้สมการแบบซิมโบลต่อซิมโบลของอัลกอริทึมแบบ MAP นั้นเราทำการกำหนดค่าต่าง ๆ และนิยามตามนี้

N คือ ค่าขนาดของความยาวของเฟรมของซิมโบลที่ส่ง

K คือ ค่าอินเด็ก ซึ่งมีค่าอยู่ในช่วง k∈ {1,N}.

$c_k = \{c_k^1, c_k^2, K, c_k^q\}$ มีความหมายว่าเป็น โค้ดซิมโบลที่ถูกสร้างขึ้นโดยชุดเข้ารหัสแบบคอนโวลูชั่น $c_k^m \in (-1,1)$, m ∈ (1,q)

- $x_k = (x_k^{1,s}, x_k^{2,p}, x_k^{3,p}, \ldots x_k^{q,p}), \{x_k^{1,s}, x_k^{2,p}, x_k^{3,p}, \Lambda, x_k^{q,p}\} \in (-A, A)$ คือ ค่าของซิมโบลที่ผ่านการมอดูเลตแล้ว
- A คือค่าของเกณฑ์ของช่องสัญญาณ ซึ่งในช่องสัญญาณแบบ AWGN นั้นเราจะสมมุติให้ค่าของ A=1
- $y_k = (y_k^{1,s}, y_k^{2,p}, y_k^{3,p}, \ldots y_k^{q,p})$ คือค่าของซิมโบลที่รับได้
- $y_1^N = (y_1, y_2, K\ y_N)$ คือ เฟรมของซิมโบลที่รับได้
- Trellis section
- **ดังที่แสดงในรูป 3.15 นั้นค่าของ Trellis นั้น จะมีพารามิเตอร์ที่มีลักษณะเฉพาะ จำนวน 4 พารามิเตอร์ คือ** parameters $u_k$, $c_k$, $s'$ **และ** s **เมื่อ** k **คือค่าของตัวอินเด็กส์ และ** $c_k$, $s'$ and s **คือค่าของอินพุท ,เอาท์พุท และเสตจเริ่มต้น และสเตจสิ้นสุดของ** trellis **ตามลำดับ ในระบบของซิมเมตริกส์คอนโวลูชั่น จะทำการกำหนดเป็นคู่** ($s'$,u), (s,u), ($s'$,c), **ซึ่งได้ทำการแสดงในรูปที่ 3.15 ซึ่งค่าของ** i∈ (1, $2^M$) **ซึ่งก็คือสถานะของ** Trellis **เขียนได้เป็น** S={$s_i$, i∈ (1, $2^M$)}.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

M คือ ค่าขนาดของหน่วยความจำ ซึ่งมีค่า M=K-1 เมื่อ K คือค่าของความยาวคอนสเตรทของชุดเข้ารหัสแบบคอนโวลูชั่น

ส่วน $s'=s_{k-1}$ คือค่าของสเตจของไดอะแกรม Trellis ที่ตำแหน่ง k-1 หรือที่ตำแหน่งของสเตจเริ่มต้น ของ $s_k$ กับค่าของอินพุท $u_k$. $s' \in S$.

. $s=s_k$ คือค่าของสเตจของ trellis ที่ตำแหน่ง k หรือ ตำแหน่งสิ้นสุดกับค่าของอินพุท $u_k$. $s \in S$.

### ก.2 สูตรและสมการที่จำเป็น

สำหรับเหตุการณ์ของ A และ B ค่าของ Joint probability ของ A และ B เขียนในรูปของ P(A,B) และคุณสมบัติอื่นๆ ดังนี้

$$P(A,B) = P(A)P(B/A) \ , \qquad [A]$$

$$P(A,B/C) = P(A/C)P(B/A,C) \ , \qquad [B]$$

$$P(A) = \sum_{B} P(A,B). \qquad [C]$$

ถ้าการปรากฏขึ้นของ A ไม่ขึ้นกับการปรากฏขึ้นของ B หรือกล่าวได้ว่าเหตุการณ์ A และ B นั้นไม่ขึ้นต่อกัน ความน่าจะเป็นแบบเงื่อนไขสามารถที่จะเขียนได้ในรูป P(A/B) ซึ่งมีความหมายว่า ความน่าจะเป็นที่จะเกิดเหตุการณ์ A นั้นขึ้นอยู่กับเหตุการณ์ของ B ซึ่งสามารถเขียนได้ดังนี้

$$P(A/B) = P(A), \qquad [D]$$

$$P(AB) = P(A)P(B). \qquad [E]$$

### ก.3 อัลกอริทึม MAP

ขั้นแรกเราจะทำการหาที่มาของอัลกอริทึมแบบ MAP โดยใช้เงื่อนไขแบบซิมโบลต่อซิมโบล ดังนั้นอัลกอริทึมในการถอดรหัสนั้น จะมีการกระทำในระดับบิท

#### ก.3.1 การถอดรหัสโดย MAP อัลกอริทึมแบบซิมโบลต่อซิมโบล

อัตราส่วนในทางลอกการิทึมของความน่าจะเป็น $L(u_k)$ ของข้อมูลแบบไบนารี่แบบสุ่มของตัวแปร $u_k$ นั้นสามารถที่จะกำหนดได้โดย

$$L(u_k) \triangleq \log\left(\frac{P(u_k = 1/y_1^N)}{P(u_k = 0/y_1^N)}\right). \qquad (ก\text{-}1)$$

การตัดสินใจนั้นจะพิจารณาจากเครื่องหมายของ L(u),

$$\tilde{u}_k = sign[L(u_k)]. \qquad (ก\text{-}2)$$

เมื่อนำเอาสมการที่ [A] และ [C] มาแทนลงในสมการที่ (ก-1) เราสามารถเขียนใหม่ได้เป็น

$$L(u_k) = \log\left(\frac{P(u_k = 1, y_1^N)/P(y_1^N)}{P(u_k = 0, y_1^N)/P(y_1^N)}\right)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$= \log[\frac{\sum_{s'} P(s_{k-1}=s',u_k=1,y_1^N)/P(y_1^N)}{\sum_{s'} P(s_{k-1}=s',u_k=0,y_1^N)/P(y_1^N)}]. \quad (ก\text{-}3)$$

เราจะทำการพิจารณาคู่ $(s_{k-1},u_k)$, ซึ่ง $s_k$ คือค่าที่ถูกกำหนดให้เป็นสาขาพิเศษของ Trellis ดังนั้นผลของการรวม $s'$ สำหรับค่าของ joint probability ในส่วนของเศษ (ส่วนบน) นั้นจะได้ค่าที่เหมือนกับผลของการรวมกันของ $u^+$ ซึ่งเป็นกลุ่มคู่ของ $(s_{k-1}=s')\rightarrow(s_k=s)$ ซึ่งมีสาเหตุมาจากการกำหนดให้ $u_k=1$,ซึ่งคล้ายๆกับ $u^-$ ซึ่งมันก็คือค่าของลำดับของคู่ $(s_{k-1}=s')\rightarrow(s_k=s)$ เมื่อทำการกำหนดค่าของอินพุท $u_k=0$

ต่อมาค่าของซิมโบลที่รับได้สามารถที่จะแบ่งแยกออกได้ 3 ส่วน ส่วนแรกนั้นจะเป็นส่วนที่มีการบรรจุค่าที่สังเกตุการณ์ก่อนเวลาที่ k ลำดับที่สองนั้นจะเป็นส่วนที่ทำการบรรจุค่าปัจจุบันที่พิจารณา ณ ตำแหน่ง k และลำดับที่สาม นั้นคือส่วนที่ทำการพิจารณา ณ ตำแหน่งที่จะเกิดขึ้นในอนาคต ณ เวลา k ซึ่งสามารถที่จะเขียนได้ดังนี้

$$y_1^N=\{y_1^{k-1},y_k,y_{k+1}^N\}. \quad (ก\text{-}4)$$

ทำการแทนสมการ (ก-4) ลงในสมการ (ก-3) ซึ่งจะได้สมการ (ก-3) ในรูปใหม่ คือ

$$L(u_k)=\log[\frac{\sum_{u^+} P(s_{k-1}=s',u_k=1,y_1^{k-1},y_k,y_{k+1}^N)/P(y_1^N)}{\sum_{u^-} P(s_{k-1}=s',u_k=0,y_1^{k-1},y_k,y_{k+1}^N)/P(y_1^N)}] \quad (ก\text{-}5)$$

เมื่อค่า $\sum_{u^+}()$ คือค่าของผลรวมของคู่สาขาที่สามารถที่จะทำการส่งผ่านได้ $(s_{k-1}, s_k)$ ณ ตำแหน่ง k โดยกำหนดให้ค่าของอินพุท $u_k=1$, และค่าของ $\sum_{u^-}()$ คือ ค่าของผลรวมของคู่สาขาทั้งหมดที่สามารถที่จะเกิดขึ้นได้ $(s_{k-1}, s_k)$ ณ ตำแหน่งที่ k โดยกำหนดให้ค่าของอินพุทมีค่าเท่ากับ $u_k=0$.

ในสมการที่ [A] ค่าของ Joint probability ในสมการ (ก-5) สามารถที่จะแทนได้ โดยใช้ condition probability ซึ่งสามารถที่จะแสดงได้โดยการใช้สเตจของ trellis $s_{k-1}=s'$ ซึ่งเหตุการณ์ในอนาคตนั้นจะขึ้นอยู่กับเหตุการณ์ในอดีตที่ผ่านมา และค่าของ joint condition ของค่า $(s_{k-1}=s',y_1^{k-1})$ ซึ่งมีค่าของ สมมูลย์กันของสถานะ 2 สถานะ $s_{k-1}=s'$ ซึ่งค่าของอัตราส่วนของ Log-likelihood สามารถที่จะเขียนได้ใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$L(u_k) = \log[\frac{\sum_{u+} P(s_{k-1}=s', y_1^{k-1}) P(u_k = 1, y_k, y_{k+1}^N / s_{k-1}=s', y_1^{k-1}) / P(y_1^N)}{\sum_{u-} P(s_{k-1}=s', y_1^{k-1}) P(u_k = 0, y_k, y_{k+1}^N / s_{k-1}=s', y_1^{k-1}) / P(y_1^N)}]$$

(ก-6)

$$= \log[\frac{\sum_{u+} P(s_{k-1}=s', y_1^{k-1}) P(u_k = 1, y_k, y_{k+1}^N / s_{k-1}=s') / P(y_1^N)}{\sum_{u-} P(s_{k-1}=s', y_1^{k-1}) P(u_k = 0, y_k, y_{k+1}^N / s_{k-1}=s') / P(y_1^N)}].$$

ใช้สมการ [B] สามารถที่จะเขียนที่มาของสมการได้ดังนี้

$$P(u_k = u, y_k, y_{k+1}^N / s_{k-1} = s')$$

$$= P(y_{k+1}^N / s_{k-1} = s', u_k = u, y_k) P(u_k = u, y_k / s_{k-1} = s') . \quad \text{(ก-7)}$$

$$= p(y_{k+1}^N / s_k = s) P(u_k = u, y_k / s_{k-1} = s')$$

Let $\alpha_k(s) \triangleq P(s_k = s, y_1^k)$, $\beta_{k-1}(s') \triangleq P(y_k^N / s_{k-1} = s')$,

และ $\gamma_k(s', s) \triangleq P(u_k = u, y_k / s_{k-1} = s')$ โดยการแทน (ก-7) ในสมการ (ก-6) เราจะได้

$$L(u_k) = \log[\frac{\sum_{u+} \alpha_{k-1}(s') \beta_k(s) \gamma_k(s', s) / P(y_1^N)}{\sum_{u-} \alpha_{k-1}(s') \beta_k(s) \gamma_k(s', s) / P(y_1^N)}]. \quad \text{(ก-8)}$$

ก.3.2 คำนวณค่า $\alpha_k(s)$, $\beta_{k-1}(s')$, $\gamma_k(s', s)$

ค่า $\alpha_k(s)$, $\beta_{k-1}(s')$, และ $\gamma_k(s', s)$

เราจะทำการพิจารณาค่า $\gamma_k(s', s)$ คือ ค่าของความน่าจะเป็นของการส่งผ่านของสเตจที่มีการส่งผ่านจาก $s_{k-1}$ ถึง $s_k$ ที่เวลา k เมื่อมีข้อมูลอินพุท คือ $u_k$. ซึ่งค่าของความน่าจะเป็นของข้อมูลอินพุท P(u) นั้นจะไม่ขึ้นกับสเตจเริ่มต้น เงื่อนไขของข้อจำกัดนั้นจะลดระดับลง เราสามารถที่จะเขียนสมการ $\gamma_k(s', s)$ โดยใช้สมการ (3.69) สามารถเขียนได้ดังนี้

$$\gamma_k(s', s) \triangleq P(u_k = u, y_k / s_{k-1} = s')$$

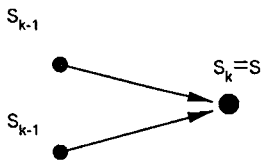
$$= P(y_k / s_{k-1} = s', u_k = u) P(u_k = u / s_{k-1} = s')$$

$$= P(y_k / c_k = c) P(u_k = u) \quad \text{(ก-9)}$$

ค่าของ $\alpha_k(s)$ คือ ค่าของ join probability ในสถานะ s ที่ตำแหน่ง k โดยเป็นค่าที่ได้ทำการสังเกตุผ่านมาแล้วในอดีต ใช้สมการ [C] และสมการ [A] ซึ่งมันสามารถที่จะทำการคำนวณค่าของรีเคอร์ซีฟ ได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\alpha_k(s) \underline{\underline{\Delta}} P(s_k = s, y_1^k) = \sum_{s'/s_k=s} P(s_{k-1} = s', u_k = u, y_1^{k-1}, y_k)$$




$$= \sum_{s'/s_k=s} P(s_{k-1} = s', y_1^{k-1}) P(u_k = u, y_k / s_{k-1} = s', y_1^{k-1})$$

$$= \sum_{s'/s_k=s} P(s_{k-1} = s', y_1^{k-1}) P(u_k = u, y_k / s_{k-1} = s')$$

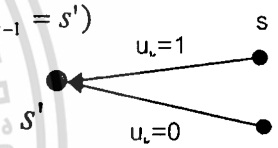
(ก-10)

$$= \sum_{s'/s_k=s} \alpha_{k-1}(s') \gamma_k(s', s) = \sum_{s'} \alpha_{k-1}(s') \gamma_k(s', s),$$

เมื่อ $\sum_{s'/s_k=s}()$ คือค่าของผลรวมของสเตจที่สามารถที่จะเป็นได้ $s_{k-1}$ ซึ่งคือสเตจสุดท้าย คือ $s_k=s$ เงื่อนไขของสถานะเริ่มต้น คือ $\alpha_0(s) = \begin{cases} 1 & \text{if } s=1 \\ 0 & otherwise \end{cases}$ ซึ่งความหมายของชุดถอดรหัส ค่าของ สเตจของ Trellis ทุกครั้งจะมีค่าเริ่มต้นเป็น 1

ค่าของ $\beta_{k-1}(s)$ คือค่า ความน่าจะเป็นแบบมีเงื่อนไขของจุดที่ทำการพิจารณาในอนาคต โดยกำหนดให้ สถานะ s'ที่ เวลา k-1 ซึ่งมันสามารถที่จะทำการคำนวณ ค่ารีเคอร์ซีฟได้ดังนี้

$$\beta_{k-1}(s') \underline{\underline{\Delta}} P(y_k^N / s_{k-1} = s') = \sum_{s/s_{k-1}=s'} P(u_k = u, y_{k+1}^N, y_k / s_{k-1} = s')$$




$$= \sum_{s/s_{k-1}=s'} P(y_{k+1}^N / s_{k-1} = s', u_k = u, y_k) P(u_k = u, y_k / s_{k-1} = s')$$

$$= \sum_{s/s_{k-1}=s'} P(y_{k+1}^N / s_k = s) P(u_k = u, y_k / s_{k-1} = s')$$

$$= \sum_{s/s_{k-1}=s'} \beta_k(s) \gamma_k(s', s), \qquad \text{(ก-11)}$$

เมื่อ $\sum_{s/s_{k-1}=s'}()$

คือค่าของผลรวมของสเตจที่ซึ่งเริ่มต้นจากสเตจ $s_{k-1} = s'$ ค่าเริ่มต้นของ The $\beta_k(s)$'s คือ

$$\beta_N = \begin{cases} 1 & \text{if } s=1 \\ 0 & \text{otherwise} \end{cases}$$

ซึ่งมีความหมายว่า ชุดเข้ารหัสนั้นมีค่าของสเตจสุดท้ายเท่ากับ 1

ตัวอย่างของการป้องกันข้อมูลเกิน ซึ่งจะเกิดขึ้นในการประยุกต์ใช้งานทางด้านฮาร์ดแวร์ ซึ่ง $\alpha_k(s)$ และ $\beta_{k-1}(s')$ นั้นต้องทำการนอมอลไลซ์ ค่าด้วยค่า $y_1^N$ ตัวอย่าง เช่น $\alpha_k(s)$ และ $\beta_{k-1}(s')$ ซึ่งจะทำการแบ่งออกได้โดย $P(y_1^N)$ เมื่อ

$$P(y_1^N) = P(y_1^{k-1}) P(y_k^N / y_1^{k-1}). \qquad \text{(ก-12)}$$

ทำการนอมอลไลซ์ ค่าในสมการ (ข-12)

$$L(u_k) = \log\left[\frac{\sum_{u+} \alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s) / P(y_1^N)}{\sum_{u-} \alpha_{k-1}(s') \gamma_k(s', s) \beta_k(s) / P(y_1^N)}\right]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$= \log[\frac{\dfrac{\sum_{u+}\alpha_{k-1}(s')\gamma_k(s',s)\beta_k(s)}{P(y_1^{k-1})P(y_k^N / y_1^{k-1})}}{\dfrac{\sum_{u-}\alpha_{k-1}(s')\gamma_k(s',s)\beta_k(s)}{P(y_1^{k-1})P(y_k^N / y_1^{k-1})}} = \log[\frac{\sum_{u+}\dfrac{\alpha_{k-1}(s')}{P(y_1^{k-1})}\gamma_k(s',s)\dfrac{\beta_k(s)}{P(y_k^N / y_1^{k-1})}}{\sum_{u-}\dfrac{\alpha_{k-1}(s')}{P(y_1^{k-1})}\gamma_k(s',s)\dfrac{\beta_k(s)}{P(y_k^N / y_1^{k-1})}}]. \quad (ก\text{-}13)$$

โดยกำหนดให้

$$\widetilde{\alpha}_{k-1}(s') \triangleq \frac{\alpha_{k-1}(s')}{P(y_1^{k-1})}, \quad \widetilde{\beta}_k(s) \triangleq \frac{\beta_k(s)}{P(y_k^N / y_1^{k-1})},$$ เราจะได้

$$L(u_k) = \log[\frac{\sum_{u+}\widetilde{\alpha}_{k-1}(s')\gamma_k(s',s)\widetilde{\beta}_k(s)}{\sum_{u-}\widetilde{\alpha}_{k-1}(s')\gamma_k(s',s)\widetilde{\beta}_k(s)}]. \quad (ก\text{-}14)$$

เพราะว่า

$$P(y_1^k) = \sum_s P(s_k = s, y_1^k) = \sum_s \alpha_k(s), \quad (ก\text{-}15)$$

ทำการหาร $\alpha_k(s)$ โดย $P(y_1^k)$ ดังนั้นเราจะได้

$$\widetilde{\alpha}_k(s) \triangleq \frac{\alpha_k(s)}{P(y_1^k)} = \frac{P(s_k = s, y_1^k)}{\sum_s P(s_k = s, y_1^k)} = \frac{\alpha_k(s)}{\sum_s \alpha_k(s)}$$

$$= \frac{\sum_{s'}\alpha_{k-1}(s')\gamma_k(s',s)}{\sum_s\sum_{s'}\alpha_{k-1}(s')\gamma_k(s',s)} = \frac{\sum_{s'}\widetilde{\alpha}_{k-1}(s')\gamma_k(s',s)}{\sum_s\sum_{s'}\widetilde{\alpha}_{k-1}(s')\gamma_k(s',s)}. \quad (ก\text{-}16)$$

เราสามารถที่จะแยกตัวประกอบ $P(y_1^N)$ ออกมาได้

$$P(y_1^N) = P(y_1^{k-1}, y_k^N) = P(y_1^{k-1})P(y_k^N / y_1^{k-1}) = P(y_{k-1}^N / y_1^{k-2})P(y_1^{k-2}) \quad (ก\text{-}17)$$

จากสมการ (ก-10) และสมการ (ก-15) และสมการ (ก-17) เราจะได้

$$P(y_{k-1}^N / y_1^{k-2}) = P(y_1^{k-1})\frac{P(y_k^N / y_1^{k-1})}{P(y_1^{k-2})}$$

$$= \sum_s\sum_{s'}\alpha_{k-2}(s')\gamma_{k-1}(s',s)\frac{P(y_k^N / y_1^{k-1})}{P(y_1^{k-2})}$$

$$= \sum_s\sum_{s'}\widetilde{\alpha}_{k-2}(s')\gamma_{k-1}(s',s)P(y_k^N / y_1^{k-1}) \quad (ก\text{-}18)$$

ทำการหารสมการ $\beta_{k-1}(s')$ จากสมการ (ก-11) โดยสมการ (ก-18) ซึ่งเราจะได้

$$\widetilde{\beta}_{k-1}(s') = \frac{\beta_{k-1}(s')}{P(y_{k-1}^N / y_1^{k-2})} = \frac{\beta_{k-1}(s')}{\sum_s\sum_{s'}\widetilde{\alpha}_{k-2}(s')\gamma_{k-1}(s',s)P(y_k^N / y_1^{k-1})} = \frac{\widetilde{\beta}_{k-1}(s')}{\sum_s\sum_{s'}\widetilde{\alpha}_{k-2}(s')\gamma_{k-1}(s',s)}$$

$$= \frac{\sum_s\widetilde{\beta}_k(s)\gamma_k(s',s)}{\sum_s\sum_{s'}\widetilde{\alpha}_{k-2}(s')\gamma_{k-1}(s',s)}. \quad (ก\text{-}19)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Note: เมื่อ $\widetilde{\beta}_{k-1}(s')$ คือค่าความแตกต่าง ซึ่งนำมาจากเอกสารอ้างอิง [33]

จาก (ก-9) เราสามารถสังเกตุได้ว่า $\gamma_k(s', s)$ จะมีส่วนประกอบ 2 ส่วนคือ

ให้ $P(c_k^1 = -1) = P_-$ and $P(c_k^1 = 1) = P_+$. ใช้ $P(c_k^1 = -1) + P(c_k^1 = 1) = 1$ และ $P(u_k) = P(c_k^1)$, ซึ่งเราสามารถที่จะทำการกำหนดได้

$$L^e(u_k) \triangleq \log\frac{P(u_k = 1)}{P(u_k = 0)} = \log\frac{P(c_k^1 = 1)}{P(c_k^1 = -1)} = \log\frac{P(c_k^1 = 1)}{1 - P(c_k^1 = 1)}$$

$$= L^e(c_k^1) = \log\frac{P_+}{P_-} = \log\frac{P_+}{1 - P_+}, \tag{ก-20}$$

เราจะได้ค่าดังนี้

$$P(c_k^1 = 1) = P_+ = \frac{\sqrt{P_-/P_+}}{1 + P_-/P_+}\sqrt{P_+/P_-} = \frac{\sqrt{P_-/P_+}}{1 + P_-/P_+}\left(\sqrt{P_+/P_-}\right)^{c_k^1}, \text{ และ} \tag{ก-21}$$

$$P(c_k^1 = -1) = P_- = \frac{\sqrt{P_-/P_+}}{1 + P_-/P_+}\left(\sqrt{P_+/P_-}\right)^{-1} = \frac{\sqrt{P_-/P_+}}{1 + P_-/P_+}\left(\sqrt{P_+/P_-}\right)^{c_k^1}. \tag{ก-22}$$

เมื่อนำสมการ (ก-21) และ (ก-22) มารวมกันเราจะได้

$$P(c_k^1) = \frac{e^{-\frac{L^e(c_k^1)}{2}}}{1 + e^{-L^e(c_k^1)}} e^{\frac{L^e(c_k^1)}{2} c_k^1} = A_k e^{\frac{L^e(c_k^1)}{2} c_k^1}, \tag{ก-23}$$

เมื่อ $A_k$ คือฟังก์ชั่นของ $L^e(c_k)$ แสดงได้ $A_k = \dfrac{e^{-\frac{L^e(c_k^1)}{2}}}{1 + e^{-L^e(c_k^1)}}$.

ทำการพิจารณาส่วนของชุดเข้ารหัส ซึ่งมีอัตราการเข้ารหัสเท่ากับ 1/q ( ซึ่งมีความหมายว่ามีอินพุทข้อมูลเข้ามา 1 บิท จะได้เอาท์พุทออกมา q บิท สำหรับอินพุทแต่ละบิท) ความน่าจะเป็นของเอาท์พุทที่ได้จากดีมอดดูเลเตอร์ $P(Y = y_k / c_k)$ เราสามารถที่จะทำการกำหนดให้ $c_k$ คือค่าที่ส่ง ซึ่งสามารถที่จะทำการประมาณค่าได้ ดังนี้

$$P(Y = y_k / c_k) = P(Y = y_k / u_k)$$

$$\approx \lim_{\Delta \to 0} P(y_k \le Y \le y_k + \Delta / U = u_k) \tag{ก-24}$$

$$= \lim_{\Delta \to 0} p(y_k / u_k)\Delta = \lim_{\Delta \to 0} p(y_k / x_k)\Delta,$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ $p(y_k/x_k)$ คือ ค่า probability density function ของ $y_k$ ซึ่ง $x_k$ คือข้อมูลในการส่ง $\Delta$ คือจำนวนบวกค่าเล็ก ๆ และเราสามารถที่จะตัดออกได้ ในการคำนวณ เมื่อ $P(y_k/u_k)$ คือปฏิภาคของ $p(y_k/x_k)$ ถ้าเรานำเอามาเขียนในรูปของ joint probability density functionจะได้

$$P(y_k/u_k)=P(y_k^{1,s},y_k^{2,p},y_k^{3,p},\Lambda/c_k^1,c_k^2,c_k^3,\Lambda)=P(y_k^{1,s},y_k^{2,p},y_k^{3,p},\Lambda/x_k^{1,s},x_k^{2,p},x_k^{3,p},\Lambda)$$

$$=P(y_k^{1,s}/x_k^{1,s})P(y_k^{2,p}/x_k^{2,p})P(y_k^{3,p}/x_k^{3,p})\Lambda=P(y_k^{1,s}/x_k^{1,s})\prod_{i=2}^{q}P(y_k^{i,p}/x_k^{i,p})$$

เมื่อช่องสัญญาณเป็นแบบ memoryless และ

$$P(y_k/u_k)\propto e^{-\frac{(y_k^{1,s}-x_k^{1,s})^2}{2\sigma_n^2}-\sum_{i=2}^{q}\frac{(y_k^{i,p}-x_k^{i,p})^2}{2\sigma_n^2}}=e^{-\frac{(y_k^{1,s})^2+(x_k^{1,s})^2}{2\sigma_n^2}-\sum_{i=2}^{q}\frac{(y_k^{i,p})^2+(x_k^{p,i})^2}{2\sigma_n^2}}\cdot e^{\frac{y_k^{1,s}x_k^{1,s}}{\sigma_n^2}+\sum_{i=2}^{q}\frac{y_k^{i,p}x_k^{i,p}}{\sigma_n^2}}$$

$$=B_k e^{\frac{y_k^{1,s}x_k^{1,s}}{\sigma_n^2}+\sum_{i=2}^{q}\frac{y_k^{i,p}x_k^{i,p}}{\sigma_n^2}}, \qquad (ก\text{-}25)$$

เมื่อ $B_k=e^{-\frac{(y_k^{1,s})^2+(x_k^{1,s})^2}{2\sigma_n^2}-\sum_{i=2}^{q}\frac{(y_k^{i,p})^2+(x_k^{p,i})^2}{2\sigma_n^2}}$.

โดยทำการรวม (ก-23) และ (ก-25), $\gamma_k(s',s)$ เราสามารถเขียนใหม่ได้

$$\gamma_k(s',s)=P(y_k/c_k=c)P(u_k=u)=B_k e^{\frac{y_k^{1,s}x_k^{1,s}}{\sigma_n^2}+\sum_{i=2}^{q}\frac{y_k^{i,p}x_k^{i,p}}{\sigma_n^2}}A_k e^{\frac{L^e(c_k^1)}{2}\cdot c_k^1}. \qquad (ก\text{-}26)$$

$A_k$ และ $B_k$ เราสามารถที่จะละเลยในการคำนวณได้ ซึ่งเราจะทำการกำหนดค่าของเกณฑ์ของช่องสัญญาณ คือ A ภายใต้ช่องสัญญาณแบบ AWGN เราจะได้

$$SNR_b=\frac{E_b}{N_0}=\frac{E_c}{rate\cdot N_0}. \qquad (ก\text{-}27)$$

เมื่อ SNR_b คืออัตราส่วนของค่าของพลังงานของสัญญาณบิทที่ไม่ได้ทำการเข้ารหัสต่อสเปรกตรัมของความหนาแน่นของกำลัง และ $E_c$ คือค่าของพลังงานบิตที่มีการเข้ารหัส และ $E_b$ คือค่าของพลังงานบิทที่ไม่ได้เข้ารหัส ซึ่ง $E_c=rate\times E_b$, และ

$$\sigma_n^2=\frac{N_0}{2}=\frac{E_c}{2\cdot rate\cdot SNR_b}=\frac{p}{2\cdot SNR_b}, \qquad p\triangleq\frac{1}{rate},$$

สมมุติให้ $E_c=1$ and A=1

โดยการแทนสมการ $\sigma_n^2$ ลงในสมการ (ก-26), เราจะได้

$$\gamma_k(s',s)=B_k\cdot e^{\frac{y_k^{1,s}\cdot x_k^{1,s}}{\sigma_n^2}+\sum_{i=2}^{q}\frac{y_k^{i,p}\cdot x_k^{i,p}}{\sigma_n^2}}\cdot A_k\cdot e^{\frac{L^e(c_k^1)}{2}\cdot c_k^1}=A_k\cdot B_k\cdot e^{\frac{y_k^{1,s}\cdot x_k^{1,s}}{\sigma_n^2}+\sum_{i=2}^{q}\frac{y_k^{i,p}\cdot x_k^{i,p}}{\sigma_n^2}}\cdot e^{\frac{L^e(c_k^1)}{2}\cdot c_k^1}$$

$$=A_k\cdot B_k\cdot\exp\left[\frac{4\cdot SNR_b}{p}\cdot\left(\frac{y_k^{1,s}\cdot A\cdot c_k^1}{2}+\sum_{i=2}^{q}\frac{y_k^{i,p}\cdot A\cdot c_k^i}{2}\right)+\frac{1}{2}\cdot L^e(c_k^1)\cdot c_k^1\right]$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$= A_k \cdot B_k \cdot \exp\left[\left(\frac{4 \cdot A \cdot SNR_b}{p} \cdot \frac{1}{2} \cdot \left(y_k^{1,s} \cdot c_k^1 + \sum_{i=2}^{q} y_k^{i,p} \cdot c_k^i\right) + \frac{1}{2} \cdot L^e(c_k^1) \cdot c_k^1\right)\right]$$

$$= A_k \cdot B_k \cdot \exp\left[\frac{1}{2} \cdot L^e(c_k^1) \cdot c_k^1 + \frac{4 \cdot A \cdot SNR_b}{p} \cdot \frac{1}{2} \cdot y_k^{1,s} \cdot c_k^1\right] \exp\left[\sum_{i=2}^{q}\left(\frac{4 \cdot A \cdot SNR_b}{p} \cdot \frac{1}{2} \cdot y_k^{i,p} \cdot c_k^i\right)\right]$$

$$= A_k \cdot B_k \cdot \exp\left[\frac{1}{2} \cdot L^e(c_k^1) \cdot c_k^1 + Lc \cdot \frac{1}{2} \cdot y_k^{1,s} \cdot c_k^1\right] \exp\left[\sum_{i=2}^{q}\left(Lc \cdot \frac{1}{2} \cdot y_k^{i,p} \cdot c_k^i\right)\right], \quad \text{(ก-28)}$$

เมื่อ

$$Lc \triangleq \frac{4 \cdot A \cdot SNR_b}{p}, \qquad \gamma^e(s',s) \triangleq \exp\left[\sum_{i=2}^{q}\left(Lc \cdot \frac{1}{2} \cdot y_k^{i,p} \cdot c_k^i\right)\right] \quad \text{(ก-29)}$$

สรุปได้ว่า

$$L(u_k) = \log\left[\frac{\sum_{u+} \tilde{\alpha}_{k-1}(s')\gamma_k(s',s)\tilde{\beta}_k(s)}{\sum_{u-} \tilde{\alpha}_{k-1}(s')\gamma_k(s',s)\tilde{\beta}_k(s)}\right]$$

$$\tilde{\alpha}_k(s) = \frac{\sum_{s'} \tilde{\alpha}_{k-1}(s')\gamma_k(s',s)}{\sum_{s}\sum_{s'} \tilde{\alpha}_{k-1}(s')\gamma_k(s',s)}, \text{initial value } \tilde{\alpha}_0(s) = \begin{cases} 1 & \text{if s} = 1 \\ 0 & otherwise \end{cases}$$

$$\tilde{\beta}_{k-1}(s') = \frac{\sum_{s} \tilde{\beta}_k(s)\gamma_k(s',s)}{\sum_{s}\sum_{s'} \tilde{\alpha}_{k-2}(s')\gamma_{k-1}(s',s)}, \text{with initial value } \tilde{\beta}_N(s) = \begin{cases} 1 & \text{if s} = 1 \\ 0 & otherwise \end{cases}$$

$$\gamma(s',s) \propto \exp\left[\frac{1}{2} \cdot L^e(c_k^1) \cdot c_k^1 + Lc \cdot \frac{1}{2} \cdot y_k^{1,s} \cdot c_k^1\right] \exp\left[\sum_{i=2}^{q}\left(Lc \cdot \frac{1}{2} \cdot y_k^{i,p} \cdot c_k^i\right)\right].$$

**ก.3.4 MAP decoding algorithm final result**

จากกฎของ Bayes's ค่าอัตราส่วน LAPP ของชุดถอดรหัส MAP สามารถที่จะเขียนได้ใหม่

$$L(u_k) = \log\left[\frac{P(u_k = 1/y_1^N)}{P(u_k = 0/y_1^N)}\right] = \log\left(\frac{P(y_1^N / u_k = 1)}{P(y_1^N / u_k = 0)}\right) + \log\left(\frac{P(u_k = 1)}{P(u_k = 0)}\right) \quad \text{(ก-30)}$$

โดยปรกติแล้ว $u_k$ นั้นจะเป็นค่าตัวแปรสุ่ม ซึ่งสามารถที่จะเขียนความน่าจะเป็นได้ $P(u_k=1) = P(u_k=0)$ ในกรณีการมอดูเลตเป็นแบบ BPSK ในระบบการถอดเข้ารหัสแบบเทอร์โบโค้ด ชุดถอดรหัส ชุดที่ 1 จะรับเอาข้อมูลบางส่วนของข้อมูลที่รับได้ $y_k = (y_k^{1,s}, y_k^{2,p}, \text{K } y_k^{q,p})$ สร้างการตัดสินใจและชุดถอดรหัส ชุดที่ 2 นั้นจะใช้ข้อมูลที่เหลือของข่าวสารที่รับได้ $y_k = (y_k^{\prime 1,s}, y_k^{\prime 2,p}, \text{K } y_k^{\prime q,p})$ สร้างการตัดสินใจชุดถอดรหัสชุดแรกจะรับเอาผลที่ได้จากการถอดรหัส ชุดที่สอง ซึ่งผลของการถอดรหัสนั้น จะได้เอาท์พุทเป็นข่าวสาร a priori จากสมการ (ก-14) และ (ก-28) เราจะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$L(u_k) = \log\left[\frac{\sum_{u^+}\tilde{\alpha}_{k-1}(s')\gamma_k(s',s)\tilde{\beta}_k(s)}{\sum_{u^-}\tilde{\alpha}_{k-1}(s')\gamma_k(s',s)\tilde{\beta}_k(s)}\right]$$

$$= \log\frac{\sum_{u^+}\tilde{\alpha}_{k-1}(s')\cdot\tilde{\beta}_k(s)\cdot\exp\left[\frac{1}{2}L^e(c_k^1)\cdot c_k^1 + \frac{1}{2}Lc\cdot y_k^{1,s}\cdot c_k^1\right]\cdot\exp\left[\sum_{i=2}^{q}\left(Lc\cdot\frac{1}{2}\cdot y_k^{i,p}\cdot c_k^i\right)\right]}{\sum_{u^-}\tilde{\alpha}_{k-1}(s')\cdot\tilde{\beta}_k(s)\cdot\exp\left[\frac{1}{2}L^e(c_k^1)\cdot c_k^1 + \frac{1}{2}Lc\cdot y_k^{1,s}\cdot c_k^1\right]\cdot\exp\left[\sum_{i=2}^{q}\left(Lc\cdot\frac{1}{2}\cdot y_k^{i,p}\cdot c_k^i\right)\right]} \quad (ก\text{-}31)$$

เมื่อ $L^e(c_k^1)$ *and* $Lc\cdot y_k^{1,s}$ คือค่าที่ขึ้นอยู่กับส่วนของ $u_k$ และค่าของสเตจของเราสามารถที่จะทำการแทนที่ลงในสมการ (ก-31) ดังนั้นเราจะได้

$$L(u_k) = \left[L^e(c_k^1) + Lc\cdot y_k^{1,s}\right] + \log\frac{\sum_{u^+}\tilde{\alpha}_{k-1}(s')\cdot\tilde{\beta}_k(s)\cdot\gamma_k^e(s',s)}{\sum_{u^-}\tilde{\alpha}_{k-1}(s')\cdot\tilde{\beta}_k(s)\cdot\gamma_k^e(s',s)}$$

$$= L_apriori + L_channel + L_extrinsic, \quad (ก\text{-}32)$$

เมื่อ $\gamma^e(s',s) \triangleq \exp\left[\sum_{i=2}^{q}\left(Lc\cdot\frac{1}{2}\cdot y_k^{i,p}\cdot c_k^i\right)\right]$,

$L^e(c_k^1)$ นั้นแทนด้วยข่าวสารแบบ a priori ของ $u_k$ ซึ่งถูกจัดการจากชุดถอดรหัสชุดที่ผ่านมา และ $Lc\cdot y_k^{1,s}$ คือค่าของผลลัพธ์ของช่องสัญญาณ ซึ่งเป็นผลมาจากการจัดการสัญญาณที่รับได้ ซึ่งตัวแปรสามตัวที่ได้กล่าวมาในตอนต้นเป็นตัวแปรที่เราสามารถแสดงลงในข่าวสารที่เป็นแบบ extrinsic ได้

$$L_extrinsic = \log\frac{\sum_{u^+}\tilde{\alpha}_{k-1}(s')\cdot\tilde{\beta}_k(s)\cdot\gamma_k^e(s',s)}{\sum_{u^-}\tilde{\alpha}_{k-1}(s')\cdot\tilde{\beta}_k(s)\cdot\gamma_k^e(s',s)} \quad (ก\text{-}33)$$

สำหรับตัวอย่างในกระบวนการทำการวนรอบในการถอดรหัสนั้น ในการคำนวณของชุดถอดรหัส ชุดที่ 1 ของค่า $L_1(c_k^1)$ เราสามารถทำการคำนวณ ได้โดย

$$L_1(c_k^1) = Lc\cdot y_k^{1,s} + L_{21}^e(c_k^1) + L_{12}^e(c_k^1), \quad (ก\text{-}34)$$

ดังนั้น

$$\tilde{u}_k = \text{sign}\{\text{L}_1(c_k^1)\}, \quad (ก\text{-}35)$$

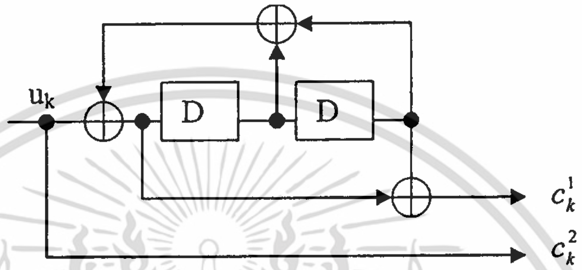
เมื่อ $L_{21}^e(u_k)$ คือค่าของข่าวสารแบบ Extrinsic ของชุดถอดรหัสชุดที่ 1 ซึ่งได้รับมาจากชุดถอดรหัส ชุดที่สอง และค่าของ $L_{12}^e(u_k)$ นั้นสามารถทำการคำนวณได้โดยใช้สมการ (ก-33) ซึ่งจะใช้ข่าวสาร extrinsic สำหรับชุดถอดรหัสชุดที่สอง ซึ่งได้ข้อมูลอินพุทมาจากชุดถอดรหัส ชุดที่หนึ่ง ผลลัพธ์ที่ได้ของค่า. $L_1(u_k)$ จะนำเอามาเป็นส่วนประกอบในการตัดสินใจ $\tilde{u}_k$.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

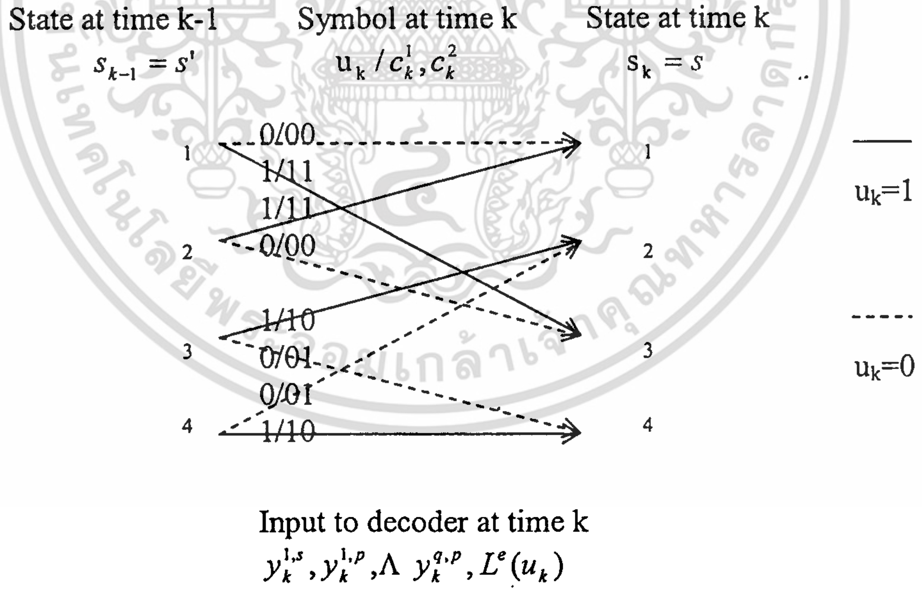
ภาคผนวก ข

# ตัวอย่างในการคำนวณการถอดรหัสของอัลกอรีทึม MAP

ในส่วนนี้จะแสดงตัวอย่างวิธีการในการคำนวณทีละขั้นตอน ในรูป ข.1 นั้นแสดงชุดเข้ารหัสแบบ RSC ซึ่งมีค่าของเจนเนอร์เรเตอร์ [1,05/07] ซึ่งมีขนาดของหน่วยความจำเท่ากับ 2 ในรูป ข.2 นั้นแสดงส่วนของ Trellis ของชุดเข้ารหัส RSC




รูปที่ ข.1 โครงสร้างของชุดเข้ารหัสแบบ RSC




**รูป ข.2** Trellis ไดอะแกรมของวงจรเข้ารหัส RSC

ตารางที่ ข.1 และ ข.2 นั้น แสดงเอาท์พุทของการส่งผ่านของชุดเข้ารหัส และแสดงสเตจของ Trellis ซึ่งได้รับข้อมูลอินพุท ส่วนในตารางที่ ข.1 นั้นแสดงอินพุทของชุดถอดรหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ ข.1** Trellis section at $u_k$=0

| State | $s_{k-1}$ | $s_k$ | $c_k^1$ | $c_k^2$ |
|---|---|---|---|---|
| 1 | 1 | 1 | -1 | -1 |
| 2 | 2 | 3 | -1 | -1 |
| 3 | 3 | 4 | -1 | 1 |
| 4 | 4 | 2 | -1 | 1 |

**ตารางที่ ข.2** Trellis section at $u_k$=1

| state | $s_{k-1}$ | $s_k$ | $c_k^1$ | $c_k^2$ |
|---|---|---|---|---|
| 1 | 1 | 3 | 1 | 1 |
| 2 | 2 | 1 | 1 | 1 |
| 3 | 3 | 2 | 1 | -1 |
| 4 | 4 | 4 | 1 | -1 |

**ตารางที่ ข.3** Decoder inputs

| input | k=1 | k=2 | K=3 | k=4 | k=5 | k=6 | K=7 |
|---|---|---|---|---|---|---|---|
| $Lc \cdot y_k^{1,s}$ | 3 | 2 | -7 | -3 | 5 | 2 | 6 |
| $Lc \cdot y_k^{2,p}$ | 2 | -1 | -6 | 1 | 0 | -5 | -7 |
| $L_k^e$ | 2 | -3 | -5 | 0 | 3 | -1 | -5 |

ต่อจากนี้ไปจะทำการคำนวณค่า ต่างๆ ดังนี้ $\alpha_k(s)$, $\beta_k(s')$, $\gamma_k(s',s)$, $L^e(u_k)$ and $L(u_k)$.

**ข.1 computing $\gamma(s',s)$**

$\gamma(s',s)$ ใช้สมการ (ก-28) ในการคำนวณ

$$\gamma(s',s) \propto \exp\left[\frac{1}{2}\cdot L^e(c_k^1)\cdot c_k^1 + Lc\cdot\frac{1}{2}\cdot y_k^{1,s}\cdot c_k^1\right]\exp\left[\sum_{i=2}^{q}\left(Lc\cdot\frac{1}{2}\cdot y_k^{i,p}\cdot c_k^i\right)\right].$$

ยกตัวอย่างให้ k=3,

$$\gamma_3(2,1) = \exp\left[\frac{1}{2}\cdot(-5)\cdot 1 + \frac{1}{2}\cdot(-7) + \frac{1}{2}\cdot(-6)\cdot 1\right] = 0.0001.$$

ตารางที่ ข.4 และ ข.5 แสดงผลลัพธ์ของ $\gamma(s',s)$ at $u_k = 0$ or $u_k = 1$ ตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ ข.4** ผลลัพธ์ของ $\gamma(s',s)$ at $u_k = 0$

| $\gamma(s',s)$ | k=1 | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 |
|---|---|---|---|---|---|---|---|
| $\gamma(1,1)$ | 0.0000 | 20.1 | 8103.1 | 2.7 | 0.0000 | 7.4 | 8103.1 |
| $\gamma(2,3)$ | 0 | 20.1 | 8103.1 | 2.7 | 0.0000 | 7.4 | 8103.1 |
| $\gamma(3,4)$ | 0.2 | 7.4 | 20.1 | 7.4 | 0.0000 | 0.0000 | 7.4 |
| $\gamma(4,2)$ | 0.2 | 7.4 | 20.1 | 7.4 | 0.0000 | 0.0000 | 7.4 |

**ตารางที่ ข.5** ผลลัพธ์ของ $\gamma(s',s)$ at $u_k = 1$

| $\gamma(s',s)$ | k=1 | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 |
|---|---|---|---|---|---|---|---|
| $\gamma(1,3)$ | 33.1155 | 0.0498 | 0.0001 | 0.3679 | 54.5982 | 0.1353 | 0.0001 |
| $\gamma(2,1)$ | 33.1155 | 0.0498 | 0.0001 | 0.3679 | 54.5982 | 0.1353 | 0.0001 |
| $\gamma(3,2)$ | 4.4817 | 0.1353 | 0.0498 | 0.1353 | 54.5982 | 20.0855 | 0.1353 |
| $\gamma(4,4)$ | 4.4817 | 0.1353 | 0.0498 | 0.1353 | 54.5982 | 20.0855 | 0.1353 |

**ข.2 computing** $\gamma^e(s',s)$

$\gamma^e(s',s)$ คือค่าที่ทำการคำนวณจากสมการ (ก-29)

$$\gamma^e(s',s) \triangleq \exp\left[\sum_{i=2}^{q}\left(Lc \cdot \frac{1}{2} \cdot y_k^{i,p} \cdot c_k^i\right)\right].$$

ยกตัวอย่างให้ k=3,

$$\gamma_3^e(2,3) = \exp\left[\frac{1}{2} \cdot (-6) \cdot (-1)\right] = 20.0855.$$

ตารางที่ ข.6 และ ข.7 แสดงผลลัพธ์ของ $\gamma^e(s',s)$ at $u_k = 0$ or $u_k = 1$ ตามลำดับ

**ตารางที่ ข.6** ผลลัพธ์ $\gamma^e(s',s)$ at $u_k = 0$

| $\gamma^e(s',s)$ | k=1 | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 |
|---|---|---|---|---|---|---|---|
| $\gamma^e(1,1)$ | 0.3679 | 1.6487 | 20.0855 | 0.6065 | 1.0000 | 12.1825 | 33.1155 |
| $\gamma^e(2,3)$ | 0.3679 | 1.6487 | 20.0855 | 0.6065 | 1.0000 | 12.1825 | 33.1155 |
| $\gamma^e(3,4)$ | 2.7183 | 0.6065 | 0.0498 | 1.6487 | 1.0000 | 0.0821 | 0.0302 |
| $\gamma^e(4,2)$ | 2.7183 | 0.6065 | 0.0498 | 1.6487 | 1.0000 | 0.0821 | 0.0302 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ตารางที่ ข.7** ผลลัพธ์ของ $\gamma^e(s',s)$ at $u_k = 1$

| $\gamma^e(s',s)$ | k=1 | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 |
|---|---|---|---|---|---|---|---|
| $\gamma^e(1,3)$ | 2.7183 | 0.6065 | 0.0498 | 1.6487 | 1.0000 | 0.0821 | 0.0302 |
| $\gamma^e(2,1)$ | 2.7183 | 0.6065 | 0.0498 | 1.6487 | 1.0000 | 0.0821 | 0.0302 |
| $\gamma^e(3,2)$ | 0.3679 | 1.6487 | 20.0855 | 0.6065 | 1.0000 | 12.1825 | 33.1155 |
| $\gamma^e(4,4)$ | 0.3679 | 1.6487 | 20.0855 | 0.6065 | 1.0000 | 12.1825 | 33.1155 |

**ข.3 computing $\widetilde{\alpha}(s)$**

$\widetilde{\alpha}(s)$ คือค่าที่คำนวณได้จากสมการ (ก-16)

$$\widetilde{\alpha}_k(s) = \frac{\sum_{s'} \widetilde{\alpha}_{k-1}(s')\gamma_k(s',s)}{\sum_s \sum_{s'} \widetilde{\alpha}_{k-1}(s')\gamma_k(s',s)}.$$

ยกตัวอย่างให้ k=3,

$$\widetilde{\alpha}_3(3) = \frac{0.0024 \cdot 0.0001 + 0.0179 \cdot 8103}{0.0024 \cdot (8103 + 0.0001) + 0.0179 \cdot (8103 + 0.0001) + 0 \cdot (20.1 + 0.0498) + 0.9796 \cdot (20.1 + 0.0498)}$$

$= 0.7868$

**ตารางที่ ข.8** ผลลัพธ์ของ $\widetilde{\alpha}(s)$

| $\widetilde{\alpha}(s)$ | k=0 | k=1 | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 |
|---|---|---|---|---|---|---|---|---|
| $\widetilde{\alpha}(1)$ | 1.0000 | 0.0009 | 0.0024 | 0.1065 | 0.0499 | 0.0165 | 0.0067 | 0.1079 |
| $\widetilde{\alpha}(2)$ | 0 | 0 | 0.0179 | 0.1065 | 0.0165 | 0.0502 | 0.0543 | 0.0136 |
| $\widetilde{\alpha}(3)$ | 0 | 0.9991 | 0 | 0.7868 | 0.0499 | 0.0499 | 0.0194 | 0.8780 |
| $\widetilde{\alpha}(4)$ | 0 | 0 | 0.9796 | 0.0003 | 0.8836 | 0.8833 | 0.9197 | 0.0005 |

**ข.4 computing $\widetilde{\beta}(s)$**

$\widetilde{\beta}(s)$ ใช้สมการ (ก-19) ในการคำนวณ

$$\widetilde{\beta}_{k-1}(s') = \frac{\sum_s \widetilde{\beta}_k(s)\gamma_k(s',s)}{\sum_s \sum_{s'} \widetilde{\alpha}_{k-2}(s')\gamma_{k-1}(s',s)}.$$

ยกตัวอย่างให้ k=3,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\widetilde{\beta}_3(3) = \frac{471.516 \cdot 0.1353 + 0.0029 \cdot 0.0074}{0.0024 \cdot (8103 + 0.0001) + 0.0179 \cdot (8103 + 0.0001) + 0 \cdot (20.1 + 0.0498) + 0.9796 \cdot (20.1 + 0.0498)}$$

$= 0.3454$

**ตารางที่ ข.9 ผลลัพธ์ของ $\widetilde{\beta}(s)$**

| $\widetilde{\beta}(s)$ | k=1 | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 |
|---|---|---|---|---|---|---|---|
| $\widetilde{\beta}(1)$ | 12.7944 | 20.9890 | 0.0195 | 0.1582 | 56.8192 | 419.980 | 1 |
| $\widetilde{\beta}(2)$ | 30.4839 | 371.441 | 0.1274 | 471.516 | 1.0407 | 0 | 0 |
| $\widetilde{\beta}(3)$ | 1.6201 | 50.2532 | 0.3454 | 8.6361 | 0 | 0 | 0 |
| $\widetilde{\beta}(4)$ | 82.8063 | 0.4640 | 18.8541 | 0.0029 | 0 | 0 | 0 |

**ข.5 computing $\sigma(s)$**

$\sigma(s)$ ใช้สมการดังต่อไปนี้ในการคำนวณ

$\sigma_k(s) = \widetilde{\alpha}_{k-1}(s') \cdot \gamma_k^e(s',s) \cdot \widetilde{\beta}_k(s)$ .

ยกตัวอย่างให้ k=3 มีข้อมูลอินพุท $u_k$=1, $\sigma_3(4) = 0.9796 \cdot 20.0855 \cdot 18.8541 = 370.979$ . ถ้าอินพุท $u_k$=0, $\sigma_3(3) = 0.0179 \cdot 20.0855 \cdot 0.3454 = 0.1245$ .

ตารางที่ ข.10 และ ข.11 แสดงผลลัพธ์ของ $\sigma(s)$ ที่ $u_k = 0$ หรือ $u_k = 1$ ตามลำดับ

**ตารางที่ ข.10 ผลลัพธ์ของ $\sigma(s)$ ที่ $u_k = 0$**

| $\sigma(s)$ | k=1 | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 |
|---|---|---|---|---|---|---|---|
| $\sigma(1)$ | 4.7068 | 0.0315 | 0.0010 | 0.0102 | 2.8379 | 84.3822 | 0.2208 |
| $\sigma(2)$ | 0 | 0 | 0.0062 | 0.2057 | 0.9196 | 0 | 0 |
| $\sigma(3)$ | 0 | 0 | 0.1245 | 0.5577 | 0 | 0 | 0 |
| $\sigma(4)$ | 0 | 0.2812 | 0 | 0.0038 | 0 | 0 | 0 |

**ตารางที่ ข.11 ผลลัพธ์ของ $\sigma(s)$ ที่ $u_k = 1$**

| $\sigma(s)$ | k=1 | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 |
|---|---|---|---|---|---|---|---|
| $\sigma(1)$ | 0 | 0 | 0 | 0.0278 | 0.9364 | 1.7315 | 0.0016 |
| $\sigma(2)$ | 0 | 611.846 | 0 | 225.010 | 0.0520 | 0 | 0 |
| $\sigma(3)$ | 4.4038 | 0.0278 | 0 | 1.5161 | 0 | 0 | 0 |
| $\sigma(4)$ | 0 | 0 | 370.979 | 0 | 0 | 0 | 0 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ข.6 computing $L_extrinsic$**

$L_extrinsic$ ใช้สมการ (ก-33)ในการคำนวณ

$$L_extrinsic = \log \frac{\sum_{u^+} \widetilde{\alpha}_{k-1}(s') \cdot \widetilde{\beta}_k(s) \cdot \gamma_k^e(s',s)}{\sum_{u^-} \widetilde{\alpha}_{k-1}(s') \cdot \widetilde{\beta}_k(s) \cdot \gamma_k^e(s',s)} = \log \left[ \frac{\sum_{u^+} \sigma_k(s)}{\sum_{u^-} \sigma_k(s)} \right],$$

เมื่อ $\sum_{u^+}()$ คือค่าของผลรวมของสาขาที่สามารถที่ทำการส่งผ่านได้ของคู่สาขา ($s_{k-1}$, $s_k$) ที่ตำแหน่ง k เมื่ออินพุท $u_k$=1, และ $\sum_{u^-}()$ คือค่าของผลรวมของสาขาที่สามารถที่ทำการส่งผ่านได้ของคู่สาขา ($s_{k-1}$, $s_k$) เมื่ออินพุท $u_k$=0. ยกตัวอย่างให้ k=3,

$$L_extrinsic_3 = \log(370.979) - \log(0.0062 + 0.1245 + 0.0010) = 7.9437.$$

ตารางที่ ข.12 แสดงผลลัพธ์ของ $L_extrinsic$.

**ตารางที่ ข.12 ผลลัพธ์ของ $L_extrinsic$**

| k=1 | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 |
|---|---|---|---|---|---|---|
| -0.0665 | 7.5790 | 7.9437 | 5.6748 | -1.3354 | -3.8864 | -4.9035 |

**ข.8 The estimation decision the decoder made**

ใช้สมการ (ก-34) ในการคำนวณ

$$L_1(c_k^1) = Lc \cdot y_k^{1,s} + L_{21}^e(c_k^1) + L_{12}^e(c_k^1),$$

ในการตัดสินใจขั้นสุดท้ายนั้นจะใช้สมการ (ก-35) ในการตัดสินใจ

$$\widetilde{u}_k = \text{sign}\{L_1(c_k^1)\}.$$

ยกตัวอย่างให้ k=3,

$$L(u_3) = -7 - 5 + 7.9437 = -4.06.$$

ผลลัพธ์สุดท้ายในการตัดสินใจแสดงในตารางที่ ข.13.

**ตารางที่ ข.13 การตัดสินใจขั้นสุดท้าย**

| | k=1 | k=2 | k=3 | k=4 | k=5 | k=6 | k=7 |
|---|---|---|---|---|---|---|---|
| $L(u_k)$ | 4.993 | 6.579 | -4.06 | 2.678 | 6.66 | -2.88 | -3.9 |
| $\widetilde{u}_k$ | 1 | 1 | 0 | 1 | 1 | 0 | 0 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ภาคผนวก ค

# ผลงานวิจัยที่ได้รับการตีพิมพ์

[1] P. Phasukkit, S. Junapiya and S. Muisee, "Performance Improvement of DS-CDMA Mobile Telephone System-based Turbo-code Encoding System". Proceedings 2003 ISCIT, September. 2003, pp. 145-149.

[2] ภัทรพงษ์ ผาสุขกิจ, สุกุมา มุ่ยสี และ สมยศ จุณณะปิยะ, "การปรับปรุงประสิทธิภาพระบบโทรศัพท์เคลื่อนที่ DS-CDMA (IS-95) โดยใช้การเข้ารหัสแบบเทอร์โบโค้ด", การประชุมวิชาการทางวิศวกรรมไฟฟ้า ครั้งที่ 26 (EECON 26), พฤศจิกายน 2546, หน้า 1564-1569.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ISCIT 2003

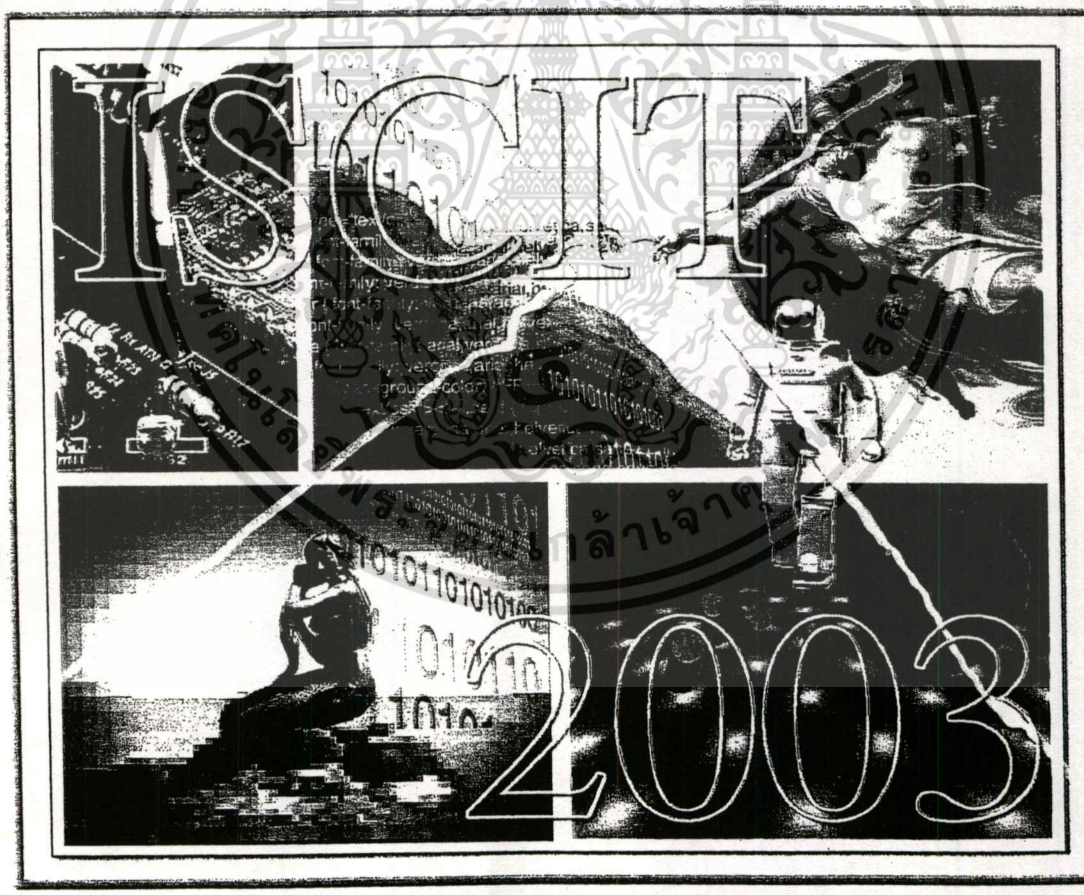
Volume I

# Proceedings

## The Third International Symposium on Communications and Information Technologies

September 3-5, 2003

BP Samila Beach Hotel and Resort, Songkhla, Thailand




ISBN 974-644-437-9

เอกสาร...ไว้...การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# PERFORMANCES IMPROVEMENT OF DS-CDMA MOBILE TELEPHONE SYSTEM-BASED TURBO-CODE ENCODING SYSTEM

*Pattarapong Phasukkit, *Somyos Junnapiya and **Sukuma Muisee*

*Department of Telecommunications Engineering

**Department of Computer Research and Service Center (CRSC)

King Mongkut's Institute of Technology Ladkrabang (KMITL)

Chalongkrung Road, Ladkrabang Bangkok Thailand 10520

Phone (+66-2)-3264329 Fax (+66-2)-3264329

Email:kppattar@kmitl.ac.th, kmsukuma@kmitl.ac.th

## Abstract

*This article present this way to improve performance of DS-CDMA(IS-95 standard) by used Turbo-Code system to act for old encoding (Convolution-Code). Simulation of system used computer programming. Result presented in performance format BER(Bit-Error-Rate). By using Turbo-Code will change size of Internal interleave in Turbo-Code system and External interleaves in DS-CDMA system so as good optimal performance.*

## 1. Introduction

The market of wireless communication is expected to increase dramatically in the late 1990s and beyond, and this expected high demand has led many service providers to investigate digital technology to satisfy the increasing demand. Spread spectrum has been used for a long time in military communications to resist intentional jamming and to achieve low-probability of detection. However, in recent years, spread spectrum has moved from military to commercial communications, culminating in the introduction of Interface Standard IS-95 *Code Division Multiple Access* (CDMA) technology as an alternative standard for commercial digital cellular and *Personal Communication System* (PCS) networks. Service providers, both cellular and PCS carriers, have deployed commercial CDMA systems in major metropolitan areas. The IS-95 CDMA is now being used in numerous cellular and PCS markets around the world. Service providers are deploying these systems in their markets, where there are mounting demands for higher capacity.

Multiple access systems share a fixed resource (i.e., frequency spectrum) to provide voice channels on demand to users. At first it seems counterintuitive that deliberately increasing the bandwidth required for transmission increases capacity. After all, in a traditional *Frequency Division Multiple Access* (FDMA) scheme, increasing the required bandwidth per user decreases the total number of users a fixed spectrum can support. We start with a definition of spread spectrum [1]. Spread-spectrum is a means of transmission in which the signal occupies a bandwidth in excess of the minimum necessary to send the information; the band spread is accomplished by means of a code that is independent of the data, and a synchronized reception with the code at the receiver is used for despreading and subsequent data recovery [2]

Turbo-Codes were first introduced in 1993 by Berrou,Glavieux, and Thitimajshima, and report in [3] ,where scheme is described that achieves a bit-error-probability of $10^{-5}$,using a rate ½ code over an *Additive White Gaussian Noise* (AWGN) channel and BPSK modulation at an $E_b / N_o$ of 0.7 *dB*. The codes are constructed by using two or more component codes on different interleaved versions of the same information sequence. Where as, for conventional codes, the final step at the decoder yields hard-decision decoded bits (or more generally, decode symbols), for a concatenated scheme, such as a turbo code, to work properly, the decoding algorithm should not limit itself to passing hard-decisions among the algorithm must effect an exchange, of soft decisions rather than hard decision. For a system with two component codes, the concept behind turbo decoding is to pass soft decisions from the output of one decoder to the input of the other decoder, and to iterate this process several times so as to produce more reliable decision [4]

## 2. Content

### 2.1 Multiple Access Using Spread Spectrum

Traditional ways of separating signals in time (i.e., *Time Division Multiple Access*, (TDMA)), or in frequency (i.e., FDMA) are relatively simple ways of making sure that the signals are orthogonal and noninterfering. However, in CDMA, different users occupy the same bandwidth at the same time, but are separated from each other via the use of a set of orthogonal waveforms, sequences, or codes. Two real-valued waveforms *x* and *y* are said to be orthogonal if their *cross-correlation* *Rxy*(0) over *T* is zero, where.

$$R_{xy}(0) = \int_0^T x(t)y(t)dt \qquad (1)$$

In discrete time, the two sequences **x** and **y** are orthogonal if their *cross-product* *R*xy(0) is zero. The cross product is defined as.

$$R_{xy}(0) = x^T y = \sum_{i=1}^{I} x_i y_i \qquad (2)$$

เอกสาร...ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Note that $T$ denotes the transpose of the column vector, which is another representation of a sequence of numbers. For example, the following two sequences or codes, **x** and **y**, are orthogonal because their cross-correlation is zero; that is, in order for the set of codes to be used in a multiple access scheme, we need two additional properties. In addition to the zero cross-correlation property, each code in the set of orthogonal codes must have an equal number of 1s and −1s. This second property gives that particular code the pseudorandom nature. The third property is that the *dot product* of each code scaled by the order of the code must equal to 1.

The order of the code is effectively the length of the code, and the dot product is defined as a scalar obtained by multiplying the sequence by itself and summing the individual terms; that is, the dot product of the code **x** is

$$R_{xx}(0) = x^T x = \sum_{i=0}^{l} x_i x_i \tag{3}$$

The two orthogonal codes in the previous example also satisfy the second and the third conditions. Both **x** and **y** have an equal number of 1s and −1s, we summarize the properties of the set of orthogonal codes to be used in DS-SS multiple access.

1. The cross-correlation should be zero or very small.
2. Each sequence in the set has an equal number of 1s and −1s, or the number of 1s differs from the number of −1s by at most 1.
3. The scaled dot product of each code should be equal to 1 [5].

## 2.2 Turbo Code Encoder

### 2.2.1 Turbo Coding

*A. Encoder* Turbo codes employ *Recursive Systematic Convolution* (RSC) codes which exhibit better performance than *Binary Convolution Encoders* (BCE) at low signal to noise ratios (SNR's), which are the power levels at which turbo codes are designed to operate [6]. These codes are recursive in nature, employing a feedback tap and also include a systematic bit in the output sequence. A MAP decoding scheme is suggested based on the original symbol-wise detection algorithm by Bahl et al [7]. This had a major drawback in terms of the complexity involved in the implementation of large constraint length codes. Concatenation provides an attractive alternative in terms of the relatively moderate complexity involved in the decoding process. *Parallel concatenation* was suggested as it permits the simultaneous processing of a bit by both decoders. In the simulation two K = 3 RSC codes are used. *Interleaving* of the information bit sequence is employed to introduce uncorrelatedness between the inputs to the two decoders. When decoded *iteratively* using the Log-Map algorithm near-optimum performance is achieved for the AWGN channel.

### 2.2.2 Log Map Algorithm

The MAP algorithm is a maximum a posteriori decoding algorithm in the sense that it computes the a posteriori probability of a bit being a zero or a one. In this sense it is a symbol-by-symbol decoding algorithm as opposed to the Viterbi algorithm, which is a Maximum-likelihood sequence Estimation (MLSE) algorithm. This algorithm minimizes the probability of bit or symbol error.

The origins of this algorithm are in [7] who applied it to the decoding of convolution codes. A modified version appears in [6], the key results of which are presented here, and implemented in our decoder with slight modifications.

A numerically stable and less complex version of the MAP algorithm, which achieves the same performance in theory but performs better in practice, performs all computations in the log domain. This is referred to as the log-MAP algorithm and is clearly explained in [8]. In this project, the Log-MAP algorithm has been implemented.

The MAP algorithm uses the LLR as a metric. The LLR is not immediately available and is computed by breaking it up into simpler terms.

Let dk be a bit being decoded and Rk be a channel observation consisting of the triplet (xk, y1k, y2k). A sequence of observations is denoted by *Rij*= {*Ri*,*Ri*+1,...,*Rj*}. The current state of the decoder is Sk N is the block size.

Using this notation it is possible to express the LLR as.

$$L(k) = \log \frac{\Pr\{d_k = 1 | R_1^N\}}{\Pr\{d_k = 0 | R_1^N\}} \tag{4}$$

The MAP algorithm decomposes the LLR into 3 terms, $\alpha$, $\beta$ and $\gamma$ which are possible to compute on observation of the entire received sequence. The result is

$$L(k) = \log \frac{\sum_m \sum_{m'} \gamma_1(R_k, m', m)\alpha_{k-1}(m')\beta_k(m)}{\sum_m \sum_{m'} \gamma_0(R_k, m', m)\alpha_{k-1}(m')\beta_k(m)} \tag{5}$$

with the parameters being defined as.

$$\alpha_k(m) = \Pr\{S_k = m | R_1^k\} \tag{6}$$

$$\beta_k(m) = \Pr\{R_{k+1}^N | S_k = m\} \tag{7}$$

$$\gamma_i(R_k, m', m) = \Pr\{d_k = i, S_k = m, R_k | S_{k-1} = m'\} \tag{8}$$

The computations of $\alpha$ and $\beta$ are recursive and are given in [BCJR] as.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 4. Simulation Setup

Simulation of the work has generated by composing the simulation program with C++ on Unix OS. The system parameters for this work are turbo codes of small data frame of 576 and large data frame of 65536. The type of internal and external block interleaver and the numbers of iteration of decoding set will be varied from #1 to #4 comparing the efficiency in the form of BER by varying of Eb/No.

Comparing DS-CDMA in up-link direction (Mobile to Base station) using convolution decoding having maximum distance and efficiency will do the comparison of the efficiencies. The used parameters of convolution are constrain length=5 and 7 (generator=$57_8$, $65_8$ $d_{free}$ = 5 and $1257_8$, $1551_8$ $d_{free}$=12) [9]. The Turbo-Code has the parameter generator ($31_8$, $27_8$), the formation of interleaver of Block and the encoding set of MAP algorithm.

## 5. Result

In figure 2 shows performance comparison of DS-CDMA when using convolution-codes by vary parameter constrain-length 5 and 9 with ones system that using turbo-codes this has size of frame data equal 65536. In figure 3 Show analyzed performance of DS-CDMA when vary number iteration decoding of turbo system by fixed size of both interleave (internal, external) to large (65536). In figure 4 shows that analysis performance of DS-CDMA by using turbo-codes fixed size of internal interleave equal 65536. Vary number of iteration and size of external interleaves equal small (576) and large (65536). In figure 5 that analysis effect of performance when vary size of both interleave (internal, external) on DS-CDMA system by using Turbo-Code and vary size of external interleave when using convolution-code.

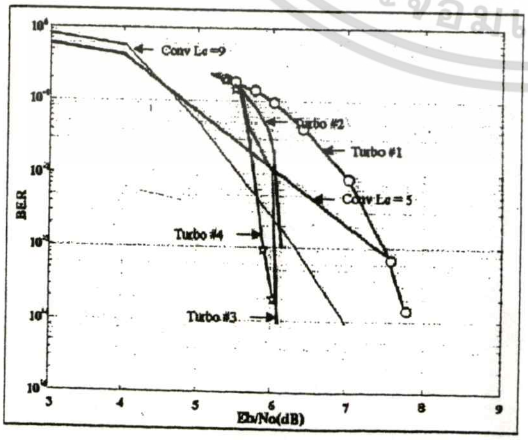



**Figure 2. Show comparison performance of DS-CDMA when using convolution-code with turbo-code system.**

Majority performance of system that using turbo-code better than convolution-code it summarize as.

1. Vary size of data frame to be large (65536) to get good performance.

2, Increase number of iteration of decoding to get goof performance.

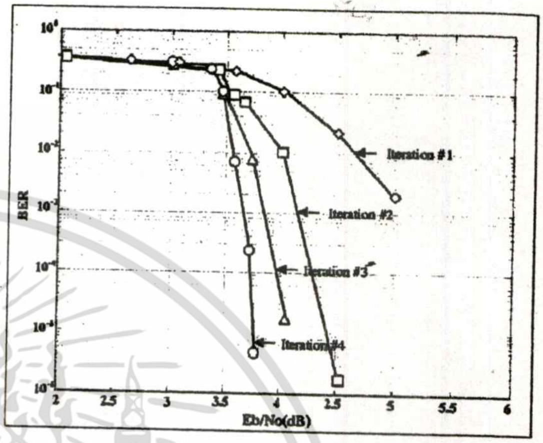



**Figure 3. Show performance of DS-CDMA When using turbo-code.**

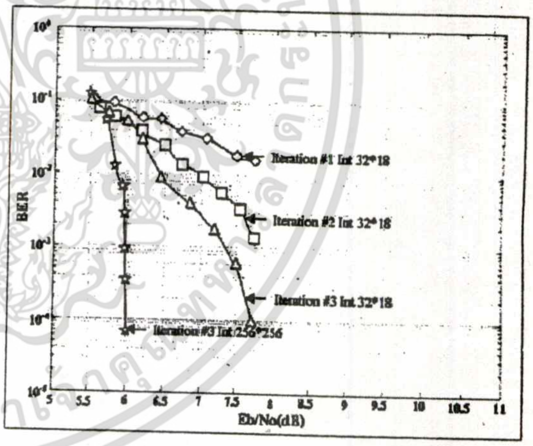



**Figure 4. Show performance of DS-CDMA when using turbo-code.**

Number of iteration increase as performance of system better

Size of interleave it large to get performance better than small, when increase number iteration of decoding performance to be good although size of interleave is small.

เอกสารนี้เป็นเอกสารที่ให้บริการเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\alpha_k(m) = \sum_{m'} \sum_{i=0}^{1} \gamma_i\{R_k, m', m)\alpha_{k-1}(m') \quad (9)$$

$$\beta(m) = \sum_{m'} \sum_{i=0}^{1} \gamma_i\{R_k, m, m'\}\beta_{k+1}(m') \quad (10)$$

The parameter $\gamma$ is easy to compute. A decomposition of this term into simpler probabilities yields.

$$\gamma_i(R_k, m', m) = \Pr\{x_k | d_k = i\}.$$
$$\Pr\{y_k | d_k = i, S_k = m, S_{k-1} = m'\}.$$
$$\Pr\{S_k | S_{k-1}\}.\Pr\{d_k = i | S_k = m, S_{k-1} = m'\} \quad (11)$$

The above decomposition takes into account the independence of systematic (xk) and redundant channel observations (yk). Thus the first two probabilities are channel transition probabilities which arise from knowledge of noise variance No. Our decoder implementation assumes the availability of this value though in practice it might not always be the case. The last term is 0 or 1 if the bit dk could cause the state transition. This is easy to obtain from the state diagram. The next to last term will initially be half. However, this term will more generally depend on the "a priori" information received as feedback by decoding stage. Such information, as explained in the previous section, cause this probability to favor particular bit logic over the other. Associating this computation with the general outline of iterative decoding explained in the last section, we see that it is this probability that serves to we see that it is this probability that serves to improve our decision, since $\gamma$ is dimple the probability that the bit $d_k$ caused the state transition.

From an intuitive standpoint, the parameter $\alpha$ is the probability of being in particular state $S_k$ given the first k channel observations (eqn.6).

Since the starting state is state 0, we have anintitial condition; so $\alpha$ can be calculated by forward recursion using (9).

The parameter $\beta$ is the probability of making a certain set of observations to the end of the trellis given a present state $S_k$. The RSC encoder is difficult to terminate in any state such as the zero-state. Therefore, we assume that the $\beta_N(m) = 1/(2^{K-1})$, where K is the constraint length, for all m [VAL]. Since the terminal values of $\beta$ by reverse recursion as in (10)[8].

## 3. Method

This paper shows the analysis of DS-CDMA as show in figures, with some high-lights, in two parts (Transmitter and receiver), the first one is the analysis of the efficiency by using Turbo-Codes encryption and decryption instead of using traditional method, Convolution Code, the second one is the analysis of the system while varying the values of external interleaving (in DS-CDMA system) and observation of the effect of the variation to the system. The output of this work will be shown in the form of BER (Bit-Error-Rate) and will be used for solution of the suitable parameters.

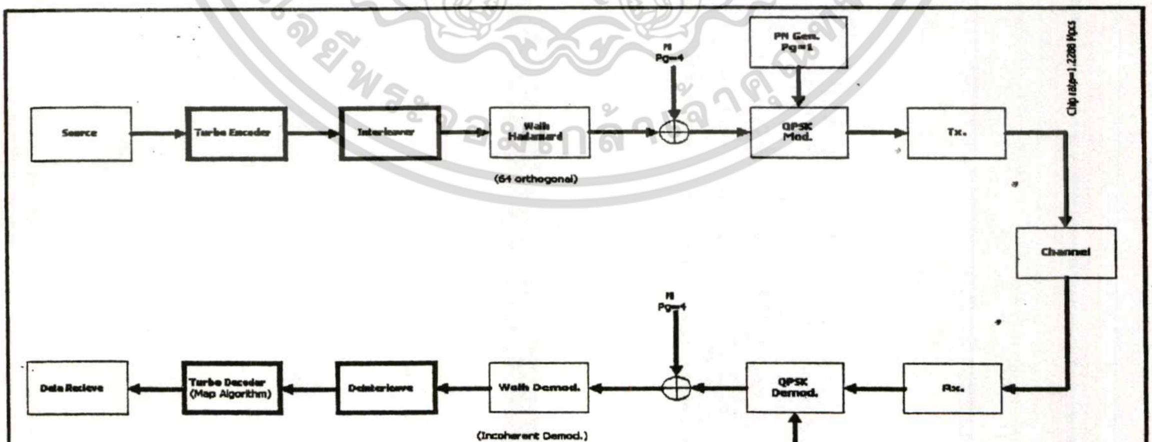



**Figure 1. Block Diagram System that Present**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

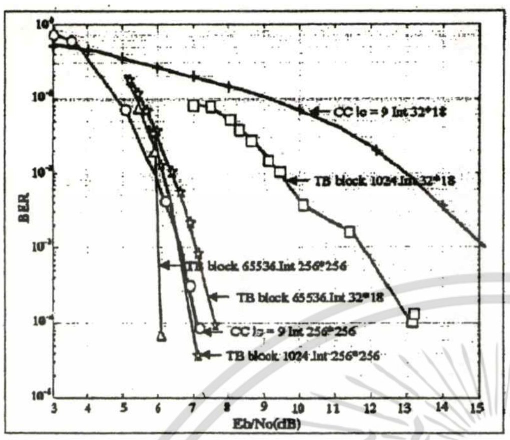



**Figure 5. Show performance of DS-CDMA when using Turbo-Code and convolution-codes.**

Summarized in Figure 5 As;

1). Performance of system to good when using size of interleaves to large both external and internal or internal, external only on system that used Turbo-Code.

2). Performance of system to good when using size of external interleaves to the large on system that used convolution-code.

## 6. Conclusions

In the article we analyses both the effect of the Turbo Interleaver size (i.e. internal interleaver) in DS-CDMA system for which the turbo performance improved as the turbo block size increases. Secondly we also condidered the presence of external interleaver (i.e. before the decoder) which function is to spread the error burst and make more effective the Turbo Decoding. The internal interleaver size depends on the transmission delay you can accept, the target QoS you want to achieve, etc. However, the sizes of two interleaver are independent. In particular the external interleaver size depends on the error burst length, the mobile speed, the power control, etc.

## 7. References


[1] Pickholtz, R. L., D. L. Schilling,and L. B. Milstein, "Theory of Spread-Spectrum Communications—A Tutorial,"*IEEE Trans. On Communications,*Vol. COM-30,No. 5,May,1982.

[2] Lee, E. A., and D. G. Messerschimitt, *Digital Communication,* Boston, MA:Kluwer Academic Publishers, 1990.

[3] Berrou, C. Glavieux, A. Thitimajshima, P., "Communications 1993,ICC '93 Geneva. Technical Program,Conference Recode,"Near Shannon limit error-correcting coding and decoding; Turbo-codes ,"*IEEE International Conference on,*Vol 2. 1993,pp 1064-1070 vol.2.

[4] Bernard Sklar, *Digital Communication Fundamentals and Applications 2nd*,pp 477, USA,Prentice -Hall Publisher,2001.

[5] Samuel C. Yang, *CDMA RF System Engineering,*.Boston, Artech House Publishers, 1998.

[6] Berrou, C.,Glavieux, A., "Near Optimum error correcting coding and Decoding:turbo-codes ,"*IEEE Trans. On Volume:44* Issuse:10, Oct.1996,pp 1261-1271.

[7] Bahl,L.R., Cocke , J. Jelinek, F., Raviv, P., "Optimal Decoding of linear codes for minimizing sysbol error rate," *IEEE Trans. Inform. Theory,*Vol.IT-20, pp 284-287, March 1974.

[8] Sklar, B. "Turbo code concepts made easy, or how I learned to concatenate and reiterate,"*Milcom 97 Proceedings,*.Vol 1,1997,pp 20-26 vol.1.

[9] Naoto SONE, Masami MOHRI, Masakatu MORI,and Hiroshi SASANO, "On Good Convolutional Codes with Optimal Free Distance for Rate ½,1/3 and ¼.,"*IEICE Trans.Cmmun.,*Vol.E84-B.,No.1 Jan 2001,pp 116-119.


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ประวัติผู้เขียน

นายภัทรพงษ์ ผาสุขกิจ เกิดเมื่อวันที่ 18 พฤษภาคม 2521 จังหวัดพระนครศรีอยุธยา สำเร็จการศึกษา วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมโทรคมนาคม จากคณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปีการศึกษา 2543 ปัจจุบันทำงานตำแหน่ง นักวิชาการคอมพิวเตอร์ ที่ทำงานสำนักวิจัยและบริการคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้