



กทปส

รายงานฉบับสมบูรณ์

โครงการขอรับการส่งเสริมและสนับสนุนจากเงินกองทุนวิจัยและพัฒนากิจการกระจายเสียง
กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ

โครงการพัฒนาอุปกรณ์เข้ารหัสช่องสัญญาณมาตรฐาน 5G
สำหรับสื่อการสอนและการต่อยอดเชิงพาณิชย์

Development of the channel coding in 5G standard for
education tool and initial commercial product

เวริต ภาคย์พิสุทธิ์

ธันวาคม 2564

กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ
(สำนักงาน กสทช.)

รายงานฉบับสมบูรณ์

ทุนส่งเสริมและสนับสนุนการวิจัยและพัฒนา
สัญญารับทุนเลขที่ B2-021/1-63

โครงการพัฒนาอุปกรณ์เข้ารหัสช่องสัญญาณมาตรฐาน 5G
สำหรับสื่อการสอนและการต่อยอดเชิงพาณิชย์
Development of the channel coding in 5G standard for education tool
and initial commercial product

นักวิจัย

- | | |
|------------------------|----------------|
| 1. เวธิต ภาคย์พิสุทธิ์ | หัวหน้าโครงการ |
| 2. กฤษณะพงศ์ พันธุ์ศรี | นักวิจัยร่วม |

ได้รับทุนอุดหนุนจาก
กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ
(สำนักงาน กสทช.)

ธันวาคม 2564

บทสรุปผู้บริหาร

โครงการพัฒนาอุปกรณ์เข้ารหัสช่องสัญญาณมาตรฐาน 5G สำหรับสื่อการสอน
และการต่อยอดเชิงพาณิชย์
ธันวาคม 2564

โครงการนี้ นักวิจัยมีเป้าหมาย 2 มิติ ได้แก่ มิติด้านการศึกษาวิศวกรรมโทรคมนาคม และมิติของการพัฒนาอุปกรณ์โทรคมนาคมต้นแบบ โดยมีมิติด้านการศึกษาที่นักวิจัยนี้ได้สังเกตเห็นปัญหาสำคัญของระบบการศึกษาทั่วโลก ที่เทคโนโลยีการสื่อสารปัจจุบันมีความซับซ้อนมากกว่าเนื้อหาในวิชาเรียน ดังนั้น นักวิจัยจึงต้องการพัฒนาอุปกรณ์ที่สามารถสนับสนุนการเรียนรู้ของนักศึกษาสาขาวิชาวิศวกรรมโทรคมนาคมหรือสาขาอื่น ๆ ที่เกี่ยวข้อง โดยอุปกรณ์สามารถแสดงให้เห็นถึงขั้นตอนของระบบสื่อสารมาตรฐาน 5G นอกจากนี้ หนึ่งในปัญหาสำคัญของอุตสาหกรรมโทรคมนาคมในประเทศไทย คือ ศักยภาพของการพัฒนาอุปกรณ์โทรคมนาคมที่มีความซับซ้อนสูง นักวิจัยจึงต้องการนำประสบการณ์ด้านงานวิจัยมาพัฒนาเป็นอุปกรณ์ที่สอดคล้องกับมาตรฐาน 5G โดยนักวิจัยคาดหวังว่าอุปกรณ์ต้นแบบที่พัฒนาขึ้นจะสามารถนำไปสู่การอุปกรณ์ที่มีประสิทธิภาพสูงสำหรับการใช้งานเชิงพาณิชย์

ทีมวิจัยได้พัฒนาชุดซอฟต์แวร์และอุปกรณ์เข้ารหัสช่องสัญญาณมาตรฐาน 5G เพื่อใช้เป็นสื่อการสอน และนำไปต่อยอดเชิงพาณิชย์ เนื้อหาของโครงการสามารถแบ่งออกเป็น 4 ส่วนหลัก ได้แก่ 1) ชุดซอฟต์แวร์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G 2) ชุดอุปกรณ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G 3) ชุดการเรียนรู้การเข้ารหัสช่องสัญญาณมาตรฐาน 5G และ 4) เว็บไซต์เผยแพร่ความรู้และชุดซอฟต์แวร์และอุปกรณ์เข้ารหัสช่องสัญญาณมาตรฐาน 5G โดยผลงานของโครงการได้นำไปใช้ประกอบการเรียนการสอนของภาควิชาวิศวกรรมโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

โครงการพัฒนาอุปกรณ์เข้ารหัสช่องสัญญาณมาตรฐาน 5G สำหรับสื่อการสอน
และการต่อยอดเชิงพาณิชย์
เวธิต ภาคย์พิสุทธิ์
ธันวาคม 2564

เทคโนโลยีการสื่อสารไร้สายยุค 5G มีความสามารถในการส่งข้อมูลความเร็วสูง ความหน่วงต่ำ และรองรับการส่งข้อมูลปริมาณมากในพื้นที่จำกัด นอกจากนี้ เทคโนโลยี 5G เรียกร้องความน่าเชื่อถือในการส่งข้อมูลในระดับอัตราเฟรมผิดพลาดน้อยกว่า 10^{-5} ด้วยเหตุนี้ เทคโนโลยี 5G จึงใช้งานรหัสช่องสัญญาณที่มีประสิทธิภาพสูงเพื่อตอบสนององระดับความผิดพลาดดังกล่าว ในปัจจุบัน การศึกษาในระดับอุดมศึกษาในภาควิชาที่เกี่ยวข้องกับวิศวกรรมการสื่อสารยังมีข้อจำกัดในเรื่องทดสอบการเข้ารหัสช่องสัญญาณในระบบสื่อสารยุคสมัยใหม่ เนื่องจากอุปกรณ์ที่เกี่ยวข้องมีราคาแพง บทความนี้ จึงนำเสนอการพัฒนาอุปกรณ์เข้ารหัสช่องสัญญาณมาตรฐาน 5G โดยมีเป้าหมายเพื่อใช้เป็นสื่อการสอนในด้านวิศวกรรมโทรคมนาคม ประกอบไปด้วยการพัฒนาชุดซอฟต์แวร์และชุดอุปกรณ์ลอจิกแบบโปรแกรมได้ (FPGA) ในการเข้ารหัสช่องสัญญาณมาตรฐาน 5G ซึ่งมีการออกแบบส่วนประสานกับผู้ใช้ (GUI) ให้มีความง่ายและเป็นมิตรต่อผู้ใช้งาน

**Development of the channel coding in 5G standard for education tool
and initial commercial product**

Watid Phakphisut

December 2021

Fifth Generation (5G) communication technology allows for high-speed data transmission, low latency, and high-volume data transmission in restricted area. Moreover, it requires high-reliability transmission with a frame error rate (FER) of less than 10^{-5} . As a results, the channel coding with high efficiency has been used to achieve a target reliability. Nowadays, higher education in the field of communication engineering is limited in testing channel coding for the modern communication systems, due to the high cost of the associated equipment. In this article, we then develop equipment for channel encoding process in accordance with the 5 G communication standard, with the intention of using it as a teaching tool in communication engineering. Our developed equipment includes a software package as well as hardware in the form of a field-programmable gate array (FPGA). In addition, we develop a user-friendly graphical user interface (GUI) to display the encoding results.

สารบัญ

| | หน้า |
|--|-----------|
| บทสรุปผู้บริหาร | ก |
| บทคัดย่อภาษาไทย | ข |
| บทคัดย่อภาษาอังกฤษ | ค |
| สารบัญตาราง | ช |
| สารบัญภาพ | ฅ |
| | |
| บทที่ 1. บทนำ | 1 |
| 1.1 ที่มา และความสำคัญของโครงการ | 1 |
| 1.2 วัตถุประสงค์ | 2 |
| 1.3 ขอบเขตของโครงการ | 2 |
| 1.4 ประโยชน์ที่คาดว่าจะได้รับ | 11 |
| | |
| บทที่ 2. ทฤษฎี และงานวิจัยที่เกี่ยวข้อง | 12 |
| 2.1 พื้นฐานรหัสช่องสัญญาณ | 12 |
| 2.2 การออกแบบมาตรฐานการเข้ารหัสช่องสัญญาณของ 3GPP | 13 |
| 2.3 การประชุมของ 3GPP เพื่อออกแบบรหัสช่องสัญญาณในมาตรฐาน 5G | 15 |
| 2.4 รายละเอียดการเข้ารหัสช่องสัญญาณชนิด LDPC codes มาตรฐาน 5G | 23 |
| 2.4.1 ภาพรวมการเข้ารหัสแอลดีพีซีมาตรฐาน 5G | 23 |
| 2.4.2 การเข้ารหัส CRC | 23 |
| 2.4.3 การแบ่งส่วนข้อมูล (Segmentation) | 27 |
| 2.4.4 รหัสแอลดีพีซี (LDPC) | 28 |
| 2.4.5 การปรับอัตรารหัส (Rate Matching) | 39 |
| 2.4.6 การแทรกสลับบิต (Bit Interleaving) | 41 |
| 2.4.7 การต่อเรียงบล็อกรหัส (Code Block Concatenation) | 43 |
| 2.5 รายละเอียดการเข้ารหัสช่องสัญญาณชนิด Polar codes มาตรฐาน 5G | 44 |
| 2.5.1 ภาพรวมการเข้ารหัสโพลาร์มาตรฐาน 5G | 44 |
| 2.5.2 การแบ่งย่อยบล็อกรหัส | 48 |
| 2.5.3 การเข้ารหัส CRC | 48 |
| 2.5.4 การสแครมบลิง CRC | 49 |

สารบัญ (ต่อ)

| | หน้า |
|---|-----------|
| 2.5.5 การแทรกสลับ CRC | 50 |
| 2.5.6 การลำดับช่องสัญญาณย่อย | 51 |
| 2.5.7 การคำนวณบิดพาริตี | 58 |
| 2.5.8 การเข้ารหัสโพลาร์ | 58 |
| 2.5.9 การแทรกสลับบล็อกย่อย | 59 |
| 2.5.10 การปรับอัตรารหัส | 60 |
| 2.5.11 การแทรกสลับบิตรหัส | 61 |
| 2.5.12 การต่อบล็อกรหัส | 62 |
| บทที่ 3. ระเบียบวิธีวิจัย | 63 |
| 3.1 รายละเอียดการออกแบบชุดซอฟต์แวร์และวงจร เข้ารหัสช่องสัญญาณชนิด LDPC codes | 63 |
| 3.1.1 ส่วนประกอบชุดซอฟต์แวร์และวงจร | 63 |
| 3.1.2 แผนผังการทำงานของชุดซอฟต์แวร์และวงจรเข้ารหัสช่องสัญญาณ | 63 |
| 3.2 รายละเอียดการออกแบบชุดซอฟต์แวร์และวงจร เข้ารหัสช่องสัญญาณชนิด Polar codes | 77 |
| 3.2.1 ส่วนประกอบชุดซอฟต์แวร์และวงจร | 77 |
| 3.2.2 รายละเอียดการออกแบบชุดซอฟต์แวร์และวงจรเข้ารหัส ช่องสัญญาณชนิด Polar codes มาตรฐาน 5G | 78 |
| 3.3 รายละเอียดการออกแบบชุดซอฟต์แวร์การเรียนรู้การสื่อสารดิจิทัล ที่มีการเข้ารหัสช่องสัญญาณมาตรฐาน 5G | 97 |
| 3.3.1 ส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อควบคุมวงจรเข้ารหัส ช่องสัญญาณชนิด LDPC Codes | 97 |
| 3.3.2 ส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อควบคุมวงจรเข้ารหัส ช่องสัญญาณชนิด Polar Codes | 100 |
| 3.4 รายละเอียดการออกแบบชุดวงจรการเรียนรู้การสื่อสารดิจิทัลที่มีการเข้ารหัส ช่องสัญญาณมาตรฐาน 5G | 103 |
| 3.4.1 ชุดวงจรเข้ารหัสร่วมกับส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อการเข้ารหัส ช่องสัญญาณชนิด LDPC Codes | 104 |

สารบัญ (ต่อ)

| | หน้า |
|--|------------|
| 3.4.2 ชุดวงจรเข้ารหัสร่วมกับส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อการเข้ารหัส ช่องสัญญาณชนิด Polar Codes | 108 |
| บทที่ 4. ผลการวิจัย และการวิจารณ์ผล | 113 |
| 4.1 ผลการทดสอบชุดซอฟต์แวร์และวงจรเข้ารหัสช่องสัญญาณชนิด LDPC codes | 113 |
| 4.1.1 วิธีการใช้งานและผลการทดสอบชุดซอฟต์แวร์ MATLAB สำหรับการเข้ารหัสช่องสัญญาณ | 113 |
| 4.1.2 วิธีการใช้งานและผลการทดสอบชุดซอฟต์แวร์ PYTHON สำหรับการเข้ารหัสช่องสัญญาณ | 120 |
| 4.1.3 วิธีการใช้งานและผลการทดสอบวงจรเข้ารหัสช่องสัญญาณ | 126 |
| 4.2 ผลการทดสอบชุดซอฟต์แวร์และวงจรเข้ารหัสช่องสัญญาณชนิด Polar codes | 139 |
| 4.2.1 การใช้งานซอฟต์แวร์และการกำหนดพารามิเตอร์เริ่มต้น | 139 |
| 4.2.2 ผลการทดสอบชุดซอฟต์แวร์เข้ารหัสโพลาร์ตามมาตรฐาน 5G | 144 |
| 4.2.3 การใช้งานชุดซอฟต์แวร์ PYTHON และกำหนดพารามิเตอร์เริ่มต้น | 168 |
| 4.2.4 ผลการทดสอบชุดซอฟต์แวร์ PYTHON | 170 |
| 4.2.5 วิธีการใช้งานวงจรเข้ารหัส FPGA สำหรับการเข้ารหัสช่องสัญญาณ | 181 |
| 4.2.6 ผลการทดสอบวงจรเข้ารหัสโพลาร์ตามมาตรฐาน 5G | 181 |
| 4.3 ผลการทดสอบชุดซอฟต์แวร์การเรียนรู้การสื่อสารดิจิทัลที่มีการเข้ารหัส ช่องสัญญาณมาตรฐาน 5G | 204 |
| 4.3.1 ส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อควบคุมวงจรเข้ารหัสช่องสัญญาณ ชนิด LDPC Codes | 204 |
| 4.3.2 ส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อควบคุมวงจรเข้ารหัสช่องสัญญาณ ชนิด Polar Codes | 209 |
| 4.4 ผลการทดสอบชุดวงจรการเรียนรู้การสื่อสารดิจิทัลที่มีการเข้ารหัสช่องสัญญาณ มาตรฐาน 5G | 215 |
| 4.4.1 ชุดวงจรเข้ารหัสร่วมกับส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อการเข้ารหัส ช่องสัญญาณชนิด LDPC Codes | 215 |
| 4.4.2 ชุดวงจรเข้ารหัสร่วมกับส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อการเข้ารหัส ช่องสัญญาณชนิด Polar Codes | 219 |

สารบัญ (ต่อ)

| | หน้า |
|---|------|
| บทที่ 5. สรุปผลการวิจัย และข้อเสนอแนะ | 229 |
| บรรณานุกรม | 231 |
| ภาคผนวก | 232 |
| ภาคผนวก ก ซอฟต์แวร์เข้ารหัสช่องสัญญาณด้วย MATLAB | 232 |
| ภาคผนวก ข ซอฟต์แวร์เข้ารหัสช่องสัญญาณด้วย PYTHON | 234 |
| ภาคผนวก ค รายละเอียดของอุปกรณ์ FPGA รุ่น Altera DE2-115 | 236 |
| ภาคผนวก ง การเผยแพร่ผลผลิตของโครงการ | 240 |
| ภาคผนวก จ ข้อเสนอแนะแนวทางการนำชุดการเรียนรู้และอุปกรณ์เข้ารหัส ช่องสัญญาณไปต่อยอด | 270 |
| ภาคผนวก ฉ รายงานผลการดำเนินงานฉบับย่อสำหรับตีพิมพ์ในวารสาร สำนักงาน กสทช. | 271 |

สารบัญตาราง

| | หน้า | |
|---------------|---|----|
| ตารางที่ 2.1 | รายละเอียดการประชุมของ 3GPP เพื่อออกแบบรหัสช่องสัญญาณในมาตรฐาน 5G | 16 |
| ตารางที่ 2.2 | ผลการปฏิบัติด้วยโอเปอเรเตอร์ Exclusive-OR | 24 |
| ตารางที่ 2.3 | การกำหนดพารามิเตอร์ในการแบ่งส่วนข้อมูล | 27 |
| ตารางที่ 2.4 | รายละเอียดเมทริกซ์ฐาน | 34 |
| ตารางที่ 2.5 | รายละเอียดค่า $k = 0, \dots, E_r - 1$ | 35 |
| ตารางที่ 2.6 | รายละเอียดค่า $V_{i,j}$ สำหรับกราฟฐาน 1 ของ LDPC | 36 |
| ตารางที่ 2.7 | รายละเอียดค่า $V_{i,j}$ สำหรับกราฟฐาน 2 ของ LDPC | 38 |
| ตารางที่ 2.8 | ตำแหน่งเริ่มต้น k_0 ของ rv ที่แตกต่างกัน | 41 |
| ตารางที่ 2.9 | การเข้ารหัสช่องสัญญาณของแต่ละประเภทช่องสัญญาณ | 46 |
| ตารางที่ 2.10 | พารามิเตอร์และขอบเขตของรหัสโพลาร์สำหรับช่องสัญญาณต่าง ๆ ในมาตรฐาน 5G | 47 |
| ตารางที่ 2.11 | ลำดับการแทรกสลับ $\Pi_L^{\max}(i)$ สำหรับการแทรกสลับ CRC | 51 |
| ตารางที่ 2.12 | ลำดับความน่าเชื่อถือของช่องสัญญาณ (polar sequence) | 53 |
| ตารางที่ 2.13 | ลำดับการแทรกสลับ สำหรับการแทรกสลับบล็อกย่อ | 60 |

สารบัญภาพ

| | หน้า |
|--|------|
| รูปที่ 1.1 ตัวอย่างการพัฒนาชุดซอฟต์แวร์การเข้ารหัสช่องสัญญาณมาตรฐาน โดยใช้ภาษา MATLAB และ Python | 3 |
| รูปที่ 1.2 บอร์ด FPGA รุ่น Altera DE2-115 | 4 |
| รูปที่ 1.3 ส่วนต่อประสานกับผู้ใช้ GUI รับข้อมูลจากชุดซอฟต์แวร์ Python (ส่วน 5G LDPC Encoder – Downlink/Uplink) | 6 |
| รูปที่ 1.4 ส่วนต่อประสานกับผู้ใช้ GUI รับข้อมูลจากชุดซอฟต์แวร์ Python (ส่วน 5G Polar Encoder – Downlink) | 7 |
| รูปที่ 1.5 ส่วนต่อประสานกับผู้ใช้ GUI รับข้อมูลจากชุดซอฟต์แวร์ Python (ส่วน 5G Polar Encoder – Uplink) | 7 |
| รูปที่ 1.6 ส่วนต่อประสานกับผู้ใช้ GUI รับข้อมูลจากชุดซอฟต์แวร์ Python (ส่วน 5G Polar Encoder – Broadcast) | 8 |
| รูปที่ 1.7 ส่วนต่อประสานกับผู้ใช้ GUI ที่รับข้อมูลจากชุดอุปกรณ์ FPGA (ส่วน 5G LDPC Encoder – Downlink/Uplink) | 8 |
| รูปที่ 1.8 ส่วนต่อประสานกับผู้ใช้ GUI ที่รับข้อมูลจากชุดอุปกรณ์ FPGA (ส่วน 5G Polar Encoder – Downlink) | 9 |
| รูปที่ 1.9 ส่วนต่อประสานกับผู้ใช้ GUI ที่รับข้อมูลจากชุดอุปกรณ์ FPGA (ส่วน 5G Polar Encoder – Uplink) | 9 |
| รูปที่ 1.10 ส่วนต่อประสานกับผู้ใช้ GUI ที่รับข้อมูลจากชุดอุปกรณ์ FPGA (ส่วน 5G Polar Encoder – Broadcast) | 10 |
| รูปที่ 1.11 เว็บไซต์ www.channelcoding.com | 11 |
| รูปที่ 2.1 แผนภาพการเข้ารหัสแอลดีพีซีมาตรฐาน 5G | 23 |
| รูปที่ 2.2 โพลีโนเมียล และการแทนค่าตัวหาร | 24 |
| รูปที่ 2.3 ขั้นตอนการทำงานของ CRC | 25 |
| รูปที่ 2.4 โครงสร้างแต่ละบล็อก ในกรณีมีจำนวนบล็อกมากกว่า 1 บล็อก | 28 |
| รูปที่ 2.5 โครงสร้างของรหัสบล็อกเชิงเส้นแบบ (N, K) | 29 |
| รูปที่ 2.6 โครงสร้างพื้นฐานเมทริกซ์ฐานของรหัส LDPC ตามมาตรฐาน 5G | 32 |
| รูปที่ 2.7 กราฟฐาน 1 | 33 |
| รูปที่ 2.8 กราฟฐาน 2 | 34 |

สารบัญญภาพ (ต่อ)

| | | หน้า |
|-------------|---|------|
| รูปที่ 2.9 | การปรับอัตราสำหรับรหัส LDPC ตามมาตรฐาน 5G | 40 |
| รูปที่ 2.10 | ตัวอย่างวิธีแทรกสลับบิต | 42 |
| รูปที่ 2.11 | ตัวอย่างการแทรกสลับบิตและการแทรกสลับบิตกลับ | 43 |
| รูปที่ 2.12 | กระบวนการแทรกสลับบิตสำหรับรหัส LDPC ตามมาตรฐาน 5G | 43 |
| รูปที่ 2.13 | กระบวนการเข้ารหัสโพลาไรในมาตรฐาน 5G | 44 |
| รูปที่ 2.14 | โครงสร้างการเข้ารหัส CRC | 48 |
| รูปที่ 2.15 | ตัวอย่างลำดับการแทรกสลับ CRC สำหรับช่องสัญญาณ BCH ที่ $A = 32$ $L = 24$ และ $K = 56$ | 50 |
| รูปที่ 2.16 | ตัวอย่างการลำดับช่องสัญญาณย่อยสำหรับช่องสัญญาณ BCH ที่เวกเตอร์ \mathbf{c}' ความยาว $K = 56$ และเวกเตอร์ \mathbf{u} ความยาว $N = 2^n = 512$ หรือ $n = n_{\max} = 9$ | 51 |
| รูปที่ 2.17 | โครงสร้างการเข้ารหัสโพลาไรที่ความยาว $N = 2$ และ $N = 4$ ตามลำดับ | 58 |
| รูปที่ 2.18 | ลำดับการแทรกสลับบล็อกย่อย | 59 |
| รูปที่ 2.19 | บัพเฟอร์วงกลมสำหรับการปรับอัตรารหัส | 60 |
| รูปที่ 2.20 | รูปแบบการแทรกสลับแบบสามเหลี่ยมขึ้นบันได | 61 |
| รูปที่ 3.1 | แผนภาพบล็อกการเข้ารหัส CRC | 63 |
| รูปที่ 3.2 | แผนภาพบล็อกการเลือกกราฟฐาน | 64 |
| รูปที่ 3.3 | แผนภาพบล็อกการกำหนดค่าความยาวบล็อกรหัส | 65 |
| รูปที่ 3.4 | แผนภาพบล็อกการคำนวณจำนวนบล็อกย่อย | 65 |
| รูปที่ 3.5 | แผนภาพบล็อกการกำหนดจำนวนบิตข้อมูลของแต่ละบล็อกรหัส | 66 |
| รูปที่ 3.6 | แผนภาพบล็อกการคำนวณอัตราขยายขนาดของเมทริกซ์ฐาน | 67 |
| รูปที่ 3.7 | แผนภาพบล็อกการแบ่งย่อยบล็อกข้อมูล | 68 |
| รูปที่ 3.8 | แผนภาพบล็อกการเข้ารหัส CRC ของบล็อกรหัส | 69 |
| รูปที่ 3.9 | แผนภาพบล็อกการกำหนดความยาวคำรหัส | 69 |
| รูปที่ 3.10 | แผนภาพบล็อกการสร้างเมทริกซ์พาริตีเช็ก H | 70 |
| รูปที่ 3.11 | แผนภาพบล็อกการเติมบิตข้อมูลในบล็อกรหัส | 71 |
| รูปที่ 3.12 | แผนภาพบล็อกการเข้ารหัส LDPC | 71 |
| รูปที่ 3.13 | แผนภาพบล็อกการกำหนดความยาวคำรหัสของบล็อกรหัส | 72 |
| รูปที่ 3.14 | แผนภาพบล็อกการคำนวณข้อมูลการส่งกลุ่มของบล็อกรหัส ($CBGTI$) | 73 |

สารบัญญภาพ (ต่อ)

| | | หน้า |
|-------------|--|------|
| รูปที่ 3.15 | แผนภาพบล็อกการคำนวณความยาวข้อมูลเอาต์พุตของกระบวนการปรับอัตราหัส | 74 |
| รูปที่ 3.16 | แผนภาพบล็อกการเลือกบิตเพื่อส่ง | 75 |
| รูปที่ 3.17 | แผนภาพบล็อกการแทรกสลับบิต | 76 |
| รูปที่ 3.18 | แผนภาพบล็อกการต่อเรียงบล็อกรหัส | 76 |
| รูปที่ 3.19 | แผนผังการทำงานการแบ่งย่อยบล็อกรหัส | 79 |
| รูปที่ 3.20 | แผนผังการทำงานการเชื่อมบิต CRC ท้ายข้อมูล | 80 |
| รูปที่ 3.21 | แผนผังการทำงานการสแครมบลิง CRC | 81 |
| รูปที่ 3.22 | แผนผังการทำงานการแทรกสลับ CRC | 82 |
| รูปที่ 3.23 | แผนผังการทำงานการสร้างลำดับการแทรกสลับ CRC | 83 |
| รูปที่ 3.25 | แผนผังการทำงานการหาความยาวคำรหัสแม่ | 85 |
| รูปที่ 3.26 | แผนผังการทำงานการลำดับช่องสัญญาณย่อย | 87 |
| รูปที่ 3.27 | แผนผังการทำงานการลำดับช่องสัญญาณย่อย (ต่อ) | 89 |
| รูปที่ 3.28 | แผนผังการทำงานการคำนวณบิตพาริตี | 90 |
| รูปที่ 3.29 | แผนผังการทำงานการเข้ารหัสโพลาร์ | 91 |
| รูปที่ 3.30 | แผนผังการทำงานการแทรกสลับบล็อกย่อย | 92 |
| รูปที่ 3.31 | แผนผังการทำงานการสร้างลำดับการแทรกสลับบล็อกย่อย | 93 |
| รูปที่ 3.32 | แผนผังการทำงานการปรับอัตราหัส | 94 |
| รูปที่ 3.33 | แผนผังการทำงานการแทรกสลับบิตรหัส | 95 |
| รูปที่ 3.34 | แผนผังการทำงานการสร้างลำดับการแทรกสลับบิตรหัส | 96 |
| รูปที่ 3.35 | แผนผังการทำงานการต่อบล็อกรหัส | 97 |
| รูปที่ 3.36 | หน้าต่างส่วนต่อประสานกราฟิกกับผู้ใช้สำหรับรหัสช่องสัญญาณชนิด LDPC Codes | 98 |
| รูปที่ 3.37 | แผนผังการทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้สำหรับรหัสช่องสัญญาณชนิด LDPC Codes | 99 |
| รูปที่ 3.38 | หน้าต่างส่วนต่อประสานกราฟิกกับผู้ใช้สำหรับรหัสช่องสัญญาณชนิด Polar Codes (Downlink) | 101 |
| รูปที่ 3.39 | หน้าต่างส่วนต่อประสานกราฟิกกับผู้ใช้สำหรับรหัสช่องสัญญาณชนิด Polar Codes (Uplink) | 101 |
| รูปที่ 3.40 | หน้าต่างส่วนต่อประสานกราฟิกกับผู้ใช้สำหรับรหัสช่องสัญญาณชนิด Polar Codes (Broadcast) | 102 |

สารบัญญภาพ (ต่อ)

| | | หน้า |
|-------------|--|------|
| รูปที่ 3.41 | แผนผังการทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้สำหรับรหัสช่องสัญญาณชนิด Polar Codes | 103 |
| รูปที่ 3.42 | ชุดวงจรการเรียนรู้การสื่อสารดิจิทัลที่มีการเข้ารหัสช่องสัญญาณมาตรฐาน 5G ประกอบด้วยส่วนต่อประสานกราฟิกกับผู้ใช้และ FPGA สำหรับรหัส LDPC | 104 |
| รูปที่ 3.43 | แผนผังการทำงานการสื่อสารระหว่างคอมพิวเตอร์สู่วงจรเข้ารหัส FPGA สำหรับรหัส LDPC | 105 |
| รูปที่ 3.44 | แผนผังการทำงานการสื่อสารระหว่างวงจรเข้ารหัส FPGA สู่คอมพิวเตอร์ สำหรับรหัส LDPC | 106 |
| รูปที่ 3.45 | ชุดวงจรการเรียนรู้การสื่อสารดิจิทัลที่มีการเข้ารหัสช่องสัญญาณมาตรฐาน 5G | 108 |
| รูปที่ 3.46 | แผนผังการทำงานการสื่อสารระหว่างคอมพิวเตอร์สู่วงจรเข้ารหัส FPGA สำหรับรหัส Polar ทุกช่องสัญญาณ | 109 |
| รูปที่ 3.47 | แผนผังการทำงานการสื่อสารระหว่างวงจรเข้ารหัส FPGA สู่คอมพิวเตอร์ สำหรับรหัส Polar ทุกช่องสัญญาณ | 110 |
| รูปที่ 4.1 | ชุดซอฟต์แวร์เข้ารหัสช่องสัญญาณชนิด LDPC codes โดยใช้โปรแกรม MATLAB | 113 |
| รูปที่ 4.2 | ตัวอย่างการเรียกใช้งานซอฟต์แวร์และการกำหนดพารามิเตอร์ของไฟล์ชื่อ test_5G_Transmission | 114 |
| รูปที่ 4.3 | ผลลัพธ์การเข้ารหัส CRC โดยใช้โปรแกรม MATLAB | 115 |
| รูปที่ 4.4 | ผลลัพธ์เวกเตอร์หลังจากการแบ่งส่วนย่อยข้อมูลโดยใช้โปรแกรม MATLAB | 115 |
| รูปที่ 4.5 | ผลลัพธ์พารามิเตอร์อื่น ๆ หลังจากการแบ่งส่วนย่อยข้อมูลโดยใช้โปรแกรม MATLAB | 116 |
| รูปที่ 4.6 | ผลลัพธ์เวกเตอร์หลังจากการเข้ารหัส CRC ของบล็อกรหัสโดยใช้โปรแกรม MATLAB | 117 |
| รูปที่ 4.7 | ผลลัพธ์เวกเตอร์หลังจากการเข้ารหัส LDPC โดยใช้โปรแกรม MATLAB | 118 |
| รูปที่ 4.8 | ผลลัพธ์เวกเตอร์หลังจากการปรับอัตรารหัสโดยใช้โปรแกรม MATLAB | 119 |
| รูปที่ 4.9 | ผลลัพธ์เวกเตอร์หลังจากการแทรกสลับบิตโดยใช้โปรแกรม MATLAB | 119 |
| รูปที่ 4.10 | ผลลัพธ์เวกเตอร์หลังจากการต่อเรียงบล็อกรหัสโดยใช้โปรแกรม MATLAB | 120 |
| รูปที่ 4.11 | ชุดซอฟต์แวร์เข้ารหัสช่องสัญญาณชนิด LDPC codes โดยใช้ภาษา PYTHON | 121 |
| รูปที่ 4.12 | ตัวอย่างการเรียกใช้งานซอฟต์แวร์ภาษา PYTHON และการกำหนดพารามิเตอร์ของไฟล์ชื่อ UserDataChannelCoding | 122 |

สารบัญญภาพ (ต่อ)

| | | หน้า |
|-------------|--|------|
| รูปที่ 4.13 | ผลลัพธ์การเข้ารหัส CRC โดยใช้ภาษา PYTHON | 123 |
| รูปที่ 4.14 | ผลลัพธ์เวกเตอร์หลังจากการแบ่งส่วนย่อยข้อมูลโดยใช้ภาษา PYTHON | 123 |
| รูปที่ 4.15 | ผลลัพธ์พารามิเตอร์อื่น ๆ หลังจากการแบ่งส่วนย่อยข้อมูลโดยใช้ภาษา PYTHON | 123 |
| รูปที่ 4.16 | ผลลัพธ์เวกเตอร์หลังจากการเข้ารหัส CRC ของบล็อกรหัสโดยใช้ภาษา PYTHON | 124 |
| รูปที่ 4.17 | ผลลัพธ์เวกเตอร์หลังจากการเข้ารหัส LDPC โดยใช้ภาษา PYTHON | 124 |
| รูปที่ 4.18 | ผลลัพธ์เวกเตอร์หลังจากการปรับอัตรารหัสโดยใช้ภาษา PYTHON | 125 |
| รูปที่ 4.19 | ผลลัพธ์เวกเตอร์หลังจากการแทรกสลับบิตโดยใช้ภาษา PYTHON | 125 |
| รูปที่ 4.20 | ผลลัพธ์เวกเตอร์หลังจากการต่อเรียงบล็อกรหัสโดยใช้ภาษา PYTHON | 125 |
| รูปที่ 4.21 | แสดงการสร้างโปรเจคใหม่บนหน้าต่างเริ่มแรก | 126 |
| รูปที่ 4.22 | หน้าต่างสำหรับกำหนดค่าเริ่มต้นของโปรเจค | 127 |
| รูปที่ 4.23 | หน้าต่างสำหรับกำหนดประเภทของโปรเจค | 127 |
| รูปที่ 4.24 | หน้าต่างสำหรับเลือกไฟล์ของโปรเจค | 128 |
| รูปที่ 4.25 | หน้าต่างสำหรับเลือกอุปกรณ์ที่ใช้กับโปรเจค | 128 |
| รูปที่ 4.26 | หน้าต่างสำหรับเลือกเครื่องมือในการสังเคราะห์วงจรและจำลองผลของโปรเจค | 129 |
| รูปที่ 4.27 | ตำแหน่งเมนูที่ใช้ในการสังเคราะห์วงจรบนหน้าต่างเริ่มแรกของโปรแกรม | 129 |
| รูปที่ 4.28 | ลักษณะของหน้าต่างเริ่มแรกเมื่อการสังเคราะห์วงจรเสร็จสมบูรณ์ | 130 |
| รูปที่ 4.29 | ขั้นตอนการเลือกเมนูเพื่อทำการจำลองผลของวงจร | 130 |
| รูปที่ 4.30 | ตำแหน่งของเมนูในการเลือก Entity เพื่อจำลองผลบนหน้าต่างเริ่มแรก ของโปรแกรม | 131 |
| รูปที่ 4.31 | ชื่อ Entity ทั้งหมดในโปรเจคภายในเมนู Work | 131 |
| รูปที่ 4.32 | ขั้นตอนเพิ่มสัญญาณจาก Entity ที่เลือกเข้าสู่หน้าต่าง Wave | 132 |
| รูปที่ 4.33 | ขั้นตอนการตั้งค่าสัญญาณนาฬิกา | 132 |
| รูปที่ 4.34 | หน้าต่างการตั้งค่าสัญญาณนาฬิกา | 133 |
| รูปที่ 4.35 | เมนูสำหรับการกำหนดค่าสัญญาณใด ๆ | 133 |
| รูปที่ 4.36 | หน้าต่างการกำหนดค่าของสัญญาณใด ๆ | 134 |
| รูปที่ 4.37 | ตำแหน่งของเมนูที่ใช้สำหรับเริ่มการจำลองการทำงานของวงจรบนหน้าต่างเริ่มแรก | 134 |
| รูปที่ 4.38 | ปุ่มสำหรับการจำลองการทำงานของวงจร | 135 |
| รูปที่ 4.39 | การทำงานของวงจรขณะสัญญาณ rst มีค่าเท่ากับ 0 | 135 |
| รูปที่ 4.40 | การทำงานของวงจรขณะสัญญาณ rst มีค่าเท่ากับ 1 | 135 |

สารบัญญภาพ (ต่อ)

| | หน้า |
|---|------|
| รูปที่ 4.41 ผลการทดสอบวงจรเข้ารหัสช่องสัญญาณ LDPC แบบรวมทุกส่วนประกอบจากโปรแกรม ModelSim-Altera | 136 |
| รูปที่ 4.42 สัญญาณ codeword (บางส่วน) | 136 |
| รูปที่ 4.43 ผลการทดสอบวงจรเข้ารหัส CRC จากโปรแกรม ModelSim-Altera | 136 |
| รูปที่ 4.44 ผลการทดสอบวงจรแบ่งย่อยบล็อกข้อมูลจากโปรแกรม ModelSim-Altera | 137 |
| รูปที่ 4.45 ผลการทดสอบวงจรเข้ารหัส CRC ของบล็อกรหัสจากโปรแกรม ModelSim-Altera | 137 |
| รูปที่ 4.46 ผลการทดสอบวงจรเข้ารหัส LDPC จากโปรแกรม ModelSim-Altera | 137 |
| รูปที่ 4.47 สัญญาณ from_top_circulant (บางส่วน) | 138 |
| รูปที่ 4.48 ผลการทดสอบวงจรปรับอัตรารหัสจากโปรแกรม ModelSim-Altera | 138 |
| รูปที่ 4.49 สัญญาณ from_RateMatching (บางส่วน) | 138 |
| รูปที่ 4.50 ผลการทดสอบวงจรแทรกสลับบิตจากโปรแกรม ModelSim-Altera | 139 |
| รูปที่ 4.51 สัญญาณ bitInterleavedCodeword (บางส่วน) | 139 |
| รูปที่ 4.52 ผลการทดสอบวงจรต่อเรียงบล็อกรหัสจากโปรแกรม ModelSim-Altera | 139 |
| รูปที่ 4.53 สัญญาณบางส่วนของ from_concat | 139 |
| รูปที่ 4.54 ไฟล์โมดูลหลัก 5 โมดูลและไฟล์ชุดคำสั่งเพื่อกำหนดพารามิเตอร์และเรียกใช้โมดูลฟังก์ชัน | 140 |
| รูปที่ 4.55 ชุดคำสั่งเพื่อกำหนดพารามิเตอร์และเรียกใช้โมดูลฟังก์ชัน | 140 |
| รูปที่ 4.56 ตำแหน่งการกดปุ่มสร้างไฟล์สคริป | 141 |
| รูปที่ 4.57 ไฟล์สคริปที่ถูกสร้าง | 141 |
| รูปที่ 4.58 ตำแหน่งการกดปุ่มสร้างไฟล์สคริปที่มีเทมเพลตฟังก์ชัน | 142 |
| รูปที่ 4.59 ไฟล์สคริปที่มีเทมเพลตฟังก์ชันที่ถูกสร้าง | 142 |
| รูปที่ 4.60 การบันทึกไฟล์สกุล m | 143 |
| รูปที่ 4.61 การเขียนคำสั่งเพื่อกำหนดพารามิเตอร์ | 143 |
| รูปที่ 4.62 การเขียนคำสั่งเพื่อเรียกใช้โมดูลฟังก์ชัน | 144 |
| รูปที่ 4.63 บิตข้อมูลขาเข้า | 145 |
| รูปที่ 4.64 บิตข้อมูลขาออก (บางส่วน) | 145 |
| รูปที่ 4.65 เวกเตอร์ c | 145 |
| รูปที่ 4.66 เวกเตอร์ c' | 146 |

สารบัญญภาพ (ต่อ)

| | | หน้า |
|-------------|---|------|
| รูปที่ 4.67 | เวกเตอร์ u (บางส่วน) | 146 |
| รูปที่ 4.68 | เวกเตอร์ d (บางส่วน) | 147 |
| รูปที่ 4.69 | เวกเตอร์ y (บางส่วน) | 148 |
| รูปที่ 4.70 | เวกเตอร์ e (บางส่วน) | 148 |
| รูปที่ 4.71 | บิตข้อมูลขาเข้า | 149 |
| รูปที่ 4.72 | บิตข้อมูลขาออก (บางส่วน) | 150 |
| รูปที่ 4.73 | เวกเตอร์ c ที่ได้จากการเข้ารหัส CRC | 150 |
| รูปที่ 4.74 | เวกเตอร์ c ที่ได้จากการสแครมบลิง CRC | 151 |
| รูปที่ 4.75 | เวกเตอร์ c' | 151 |
| รูปที่ 4.76 | เวกเตอร์ u (บางส่วน) | 152 |
| รูปที่ 4.77 | เวกเตอร์ d (บางส่วน) | 152 |
| รูปที่ 4.78 | เวกเตอร์ y (บางส่วน) | 153 |
| รูปที่ 4.79 | เวกเตอร์ e (บางส่วน) | 153 |
| รูปที่ 4.80 | บิตข้อมูลขาเข้า | 155 |
| รูปที่ 4.81 | บิตข้อมูลขาออก (บางส่วน) | 155 |
| รูปที่ 4.82 | เวกเตอร์ a' บล็อกแรก (บางส่วน) | 156 |
| รูปที่ 4.83 | เวกเตอร์ a' บล็อกท้าย (บางส่วน) | 156 |
| รูปที่ 4.84 | เวกเตอร์ c บล็อกแรก (บางส่วน) | 157 |
| รูปที่ 4.85 | เวกเตอร์ c บล็อกท้าย (บางส่วน) | 157 |
| รูปที่ 4.86 | เวกเตอร์ u บล็อกแรก (บางส่วน) | 158 |
| รูปที่ 4.87 | เวกเตอร์ u บล็อกท้าย (บางส่วน) | 158 |
| รูปที่ 4.88 | เวกเตอร์ d บล็อกแรก (บางส่วน) | 159 |
| รูปที่ 4.89 | เวกเตอร์ d บล็อกท้าย (บางส่วน) | 159 |
| รูปที่ 4.90 | เวกเตอร์ y บล็อกแรก (บางส่วน) | 160 |
| รูปที่ 4.91 | เวกเตอร์ y บล็อกท้าย (บางส่วน) | 160 |
| รูปที่ 4.92 | เวกเตอร์ e บล็อกแรก (บางส่วน) | 161 |
| รูปที่ 4.93 | เวกเตอร์ e บล็อกท้าย (บางส่วน) | 161 |
| รูปที่ 4.94 | เวกเตอร์ f บล็อกแรก (บางส่วน) | 162 |

สารบัญญภาพ (ต่อ)

| | หน้า |
|--|------|
| รูปที่ 4.95 เวกเตอร์ f บล็อกท้าย (บางส่วน) | 162 |
| รูปที่ 4.96 เวกเตอร์ g (บางส่วน) | 163 |
| รูปที่ 4.97 บิตข้อมูลขาเข้า | 164 |
| รูปที่ 4.98 บิตข้อมูลขาออก (บางส่วน) | 164 |
| รูปที่ 4.99 เวกเตอร์ c | 164 |
| รูปที่ 4.100 เวกเตอร์ u ที่ได้จากการลำดับช่องสัญญาณย่อย | 165 |
| รูปที่ 4.101 เวกเตอร์ u ที่ได้จากการคำนวณบิตพาริตีตรวจสอบ | 166 |
| รูปที่ 4.102 เวกเตอร์ d | 166 |
| รูปที่ 4.103 เวกเตอร์ y | 167 |
| รูปที่ 4.104 เวกเตอร์ e | 167 |
| รูปที่ 4.105 เวกเตอร์ f | 168 |
| รูปที่ 4.106 ตัวอย่างการเรียกใช้งานซอฟต์แวร์ภาษา PYTHON และการกำหนดพารามิเตอร์ | 169 |
| รูปที่ 4.107 ผลลัพธ์การเข้ารหัส CRC สำหรับช่องสัญญาณ PBCH | 170 |
| รูปที่ 4.108 ผลลัพธ์การแทรกสลับ CRC สำหรับช่องสัญญาณ PBCH | 171 |
| รูปที่ 4.109 ผลลัพธ์การลำดับช่องสัญญาณย่อยสำหรับช่องสัญญาณ PBCH | 171 |
| รูปที่ 4.110 ผลลัพธ์การเข้ารหัสโพลาร์สำหรับช่องสัญญาณ PBCH | 172 |
| รูปที่ 4.111 ผลลัพธ์การแทรกสลับบล็อกย่อยสำหรับช่องสัญญาณ PBCH | 172 |
| รูปที่ 4.112 ผลลัพธ์การปรับอัตรารหัสสำหรับช่องสัญญาณ PBCH | 173 |
| รูปที่ 4.113 ผลลัพธ์การเข้ารหัส CRC สำหรับช่องสัญญาณ PDCCH | 173 |
| รูปที่ 4.114 ผลลัพธ์การสแคมบลิง CRC สำหรับช่องสัญญาณ PDCCH | 174 |
| รูปที่ 4.115 ผลลัพธ์การแทรกสลับ CRC สำหรับช่องสัญญาณ PDCCH | 174 |
| รูปที่ 4.116 ผลลัพธ์การลำดับช่องสัญญาณย่อยสำหรับช่องสัญญาณ PDCCH | 174 |
| รูปที่ 4.117 ผลลัพธ์การเข้ารหัสโพลาร์สำหรับช่องสัญญาณ PDCCH | 175 |
| รูปที่ 4.118 ผลลัพธ์การแทรกสลับบล็อกย่อยสำหรับช่องสัญญาณ PDCCH | 175 |
| รูปที่ 4.119 ผลลัพธ์การปรับอัตรารหัสสำหรับช่องสัญญาณ PDCCH | 176 |
| รูปที่ 4.120 ผลลัพธ์การแบ่งบล็อกย่อยสำหรับช่องสัญญาณ PUCCH | 176 |
| รูปที่ 4.121 ผลลัพธ์การเข้ารหัส CRC สำหรับช่องสัญญาณ PUCCH | 177 |
| รูปที่ 4.122 ผลลัพธ์การลำดับช่องสัญญาณย่อยสำหรับช่องสัญญาณ PUCCH | 177 |
| รูปที่ 4.123 ผลลัพธ์การคำนวณบิตพาริตีสำหรับช่องสัญญาณ PUCCH | 178 |

สารบัญญภาพ (ต่อ)

| | หน้า |
|--|------|
| รูปที่ 4.124 ผลลัพธ์การเข้ารหัสโพลาไรส์สำหรับช่องสัญญาณ PUCCH | 178 |
| รูปที่ 4.125 ผลลัพธ์การแทรกสลับบล็อกย่อยสำหรับช่องสัญญาณ PUCCH | 179 |
| รูปที่ 4.126 ผลลัพธ์การปรับอัตรารหัสสำหรับช่องสัญญาณ PUCCH | 179 |
| รูปที่ 4.127 ผลลัพธ์การแทรกสลับบิตรหัสสำหรับช่องสัญญาณ PUCCH | 180 |
| รูปที่ 4.128 ผลลัพธ์การต่อบล็อกรหัสสำหรับช่องสัญญาณ PUCCH | 180 |
| รูปที่ 4.129 สัญญาณบิตข้อมูลขาเข้า (บางส่วน) | 182 |
| รูปที่ 4.130 บิตข้อมูลขาเข้า (บางส่วน) | 182 |
| รูปที่ 4.131 สัญญาณบิตข้อมูลขาออก (บางส่วน) | 182 |
| รูปที่ 4.132 บิตข้อมูลขาออก (บางส่วน) | 182 |
| รูปที่ 4.133 สัญญาณ sInfoWithCRC (บางส่วน) | 182 |
| รูปที่ 4.134 บิต sInfoWithCRC (บางส่วน) | 183 |
| รูปที่ 4.135 สัญญาณ sInterleavedInfoWithCRC (บางส่วน) | 183 |
| รูปที่ 4.136 บิต sInterleavedInfoWithCRC (บางส่วน) | 183 |
| รูปที่ 4.137 สัญญาณ sEncodingBits (บางส่วน) | 183 |
| รูปที่ 4.138 บิต sEncodingBits (บางส่วน) | 184 |
| รูปที่ 4.139 สัญญาณ sMotherCodeword (บางส่วน) | 184 |
| รูปที่ 4.140 บิต sMotherCodeword (บางส่วน) | 184 |
| รูปที่ 4.141 สัญญาณ sSubBlockIntlMotherCodeword (บางส่วน) | 184 |
| รูปที่ 4.142 บิต sSubBlockIntlMotherCodeword (บางส่วน) | 185 |
| รูปที่ 4.143 สัญญาณ sRateMatchedCodeword (บางส่วน) | 185 |
| รูปที่ 4.144 บิต sRateMatchedCodeword (บางส่วน) | 185 |
| รูปที่ 4.145 สัญญาณบิตข้อมูลขาเข้า (บางส่วน) | 186 |
| รูปที่ 4.146 บิตข้อมูลขาเข้า (บางส่วน) | 186 |
| รูปที่ 4.147 สัญญาณบิตข้อมูลขาออก (บางส่วน) | 186 |
| รูปที่ 4.148 บิตข้อมูลขาออก (บางส่วน) | 187 |
| รูปที่ 4.149 สัญญาณ sInfoWithCRC (บางส่วน) | 187 |
| รูปที่ 4.150 บิต sInfoWithCRC (บางส่วน) | 187 |
| รูปที่ 4.151 สัญญาณ sScambledInfoWithCRC (บางส่วน) | 187 |
| รูปที่ 4.152 บิต sScambledInfoWithCRC (บางส่วน) | 188 |

สารบัญญภาพ (ต่อ)

| | หน้า |
|---|------|
| รูปที่ 4.153 สัญญาณ sInterleavedInfoWithCRC (บางส่วน) | 188 |
| รูปที่ 4.154 บิต sInterleavedInfoWithCRC (บางส่วน) | 188 |
| รูปที่ 4.155 สัญญาณ sEncodingBits (บางส่วน) | 188 |
| รูปที่ 4.156 บิต sEncodingBits (บางส่วน) | 189 |
| รูปที่ 4.157 สัญญาณ sMotherCodeword (บางส่วน) | 189 |
| รูปที่ 4.158 บิต sMotherCodeword (บางส่วน) | 189 |
| รูปที่ 4.159 สัญญาณ sSubBlockIntlMotherCodeword (บางส่วน) | 189 |
| รูปที่ 4.160 บิต sSubBlockIntlMotherCodeword (บางส่วน) | 190 |
| รูปที่ 4.161 สัญญาณ sRateMatchedCodeword (บางส่วน) | 190 |
| รูปที่ 4.162 บิต sRateMatchedCodeword (บางส่วน) | 190 |
| รูปที่ 4.163 สัญญาณบิตข้อมูลขาเข้า (บางส่วน) | 191 |
| รูปที่ 4.164 บิตข้อมูลขาเข้า (บางส่วน) | 191 |
| รูปที่ 4.165 สัญญาณบิตข้อมูลขาออก (บางส่วน) | 191 |
| รูปที่ 4.166 บิตข้อมูลขาออก (บางส่วน) | 192 |
| รูปที่ 4.167 สัญญาณ sSegmentedBits บล็อกแรก (บางส่วน) | 192 |
| รูปที่ 4.168 สัญญาณ sSegmentedBits บล็อกท้าย (บางส่วน) | 192 |
| รูปที่ 4.169 บิต sSegmentedBits บล็อกแรก (บางส่วน) | 192 |
| รูปที่ 4.170 บิต sSegmentedBits บล็อกท้าย (บางส่วน) | 193 |
| รูปที่ 4.171 สัญญาณ sInfoWithCRC บล็อกแรก (บางส่วน) | 193 |
| รูปที่ 4.172 สัญญาณ sInfoWithCRC บล็อกท้าย (บางส่วน) | 193 |
| รูปที่ 4.173 บิต sInfoWithCRC บล็อกแรก (บางส่วน) | 193 |
| รูปที่ 4.174 บิต sInfoWithCRC บล็อกท้าย (บางส่วน) | 194 |
| รูปที่ 4.175 สัญญาณ sEncodingBits บล็อกแรก (บางส่วน) | 194 |
| รูปที่ 4.176 สัญญาณ sEncodingBits บล็อกท้าย (บางส่วน) | 194 |
| รูปที่ 4.177 บิต sEncodingBits บล็อกแรก (บางส่วน) | 194 |
| รูปที่ 4.178 บิต sEncodingBits บล็อกท้าย (บางส่วน) | 195 |
| รูปที่ 4.179 สัญญาณ sMotherCodeword บล็อกแรก (บางส่วน) | 195 |
| รูปที่ 4.180 สัญญาณ sMotherCodeword บล็อกท้าย (บางส่วน) | 195 |
| รูปที่ 4.181 บิต sMotherCodeword บล็อกแรก (บางส่วน) | 195 |

สารบัญญภาพ (ต่อ)

| | หน้า |
|--|------|
| รูปที่ 4.182 บิต sMotherCodeword บล็อกท้าย (บางส่วน) | 196 |
| รูปที่ 4.183 สัญญาณ sSubBlockIntlMotherCodeword บล็อกแรก (บางส่วน) | 196 |
| รูปที่ 4.184 สัญญาณ sSubBlockIntlMotherCodeword บล็อกท้าย (บางส่วน) | 196 |
| รูปที่ 4.185 บิต sSubBlockIntlMotherCodeword บล็อกแรก (บางส่วน) | 196 |
| รูปที่ 4.186 บิต sSubBlockIntlMotherCodeword บล็อกท้าย (บางส่วน) | 197 |
| รูปที่ 4.187 สัญญาณ sRateMatchedCodeword บล็อกแรก (บางส่วน) | 197 |
| รูปที่ 4.188 สัญญาณ sRateMatchedCodeword บล็อกท้าย (บางส่วน) | 197 |
| รูปที่ 4.189 บิต sRateMatchedCodeword บล็อกแรก (บางส่วน) | 197 |
| รูปที่ 4.190 บิต sRateMatchedCodeword บล็อกท้าย (บางส่วน) | 198 |
| รูปที่ 4.191 สัญญาณ sCodedBitInterleavedCodeword บล็อกแรก (บางส่วน) | 198 |
| รูปที่ 4.192 สัญญาณ sCodedBitInterleavedCodeword บล็อกท้าย (บางส่วน) | 198 |
| รูปที่ 4.193 บิต sCodedBitInterleavedCodeword บล็อกแรก (บางส่วน) | 198 |
| รูปที่ 4.194 บิต sCodedBitInterleavedCodeword บล็อกท้าย (บางส่วน) | 199 |
| รูปที่ 4.195 สัญญาณ sConcatenatedCodeword (บางส่วน) | 199 |
| รูปที่ 4.196 บิต sConcatenatedCodeword (บางส่วน) | 199 |
| รูปที่ 4.197 สัญญาณบิตข้อมูลขาเข้า (บางส่วน) | 200 |
| รูปที่ 4.198 บิตข้อมูลขาเข้า (บางส่วน) | 200 |
| รูปที่ 4.199 สัญญาณบิตข้อมูลขาออก (บางส่วน) | 200 |
| รูปที่ 4.200 บิตข้อมูลขาออก (บางส่วน) | 201 |
| รูปที่ 4.201 สัญญาณ sInfoWithCRC (บางส่วน) | 201 |
| รูปที่ 4.202 บิต sInfoWithCRC (บางส่วน) | 201 |
| รูปที่ 4.203 สัญญาณ sEncodingWOParityBits (บางส่วน) | 201 |
| รูปที่ 4.204 บิต sEncodingWOParityBits (บางส่วน) | 202 |
| รูปที่ 4.205 สัญญาณ sEncodingBits (บางส่วน) | 202 |
| รูปที่ 4.206 บิต sEncodingBits (บางส่วน) | 202 |
| รูปที่ 4.207 สัญญาณ sMotherCodeword (บางส่วน) | 202 |
| รูปที่ 4.208 บิต sMotherCodeword (บางส่วน) | 203 |
| รูปที่ 4.209 สัญญาณ sSubBlockIntlCodeword (บางส่วน) | 203 |
| รูปที่ 4.210 บิต sSubBlockIntlCodeword (บางส่วน) | 203 |

สารบัญญภาพ (ต่อ)

| | หน้า |
|---|------|
| รูปที่ 4.211 สัญญาณ sRateMatchedCodeword (บางส่วน) | 203 |
| รูปที่ 4.212 บิต sRateMatchedCodeword (บางส่วน) | 204 |
| รูปที่ 4.213 สัญญาณ sCodedBitInterleaveCodeword (บางส่วน) | 204 |
| รูปที่ 4.214 บิต sCodedBitInterleaveCodeword (บางส่วน) | 204 |
| รูปที่ 4.215 หน้าแรกของส่วนต่อประสานกราฟิกกับผู้ใช้ | 205 |
| รูปที่ 4.216 การเลือกชนิดของตัวเข้ารหัสกรณิรหัสช่องสัญญาณชนิด LDPC Codes | 206 |
| รูปที่ 4.217 หน้าต่างแสดงรายการพารามิเตอร์กรณิรเข้ารหัสช่องสัญญาณ โดยใช้ 5G LDPC Encoder – Downlink/Uplink | 207 |
| รูปที่ 4.218 การเลือกวิธีการเข้ารหัส | 208 |
| รูปที่ 4.219 ผลลัพธ์การเข้ารหัสช่องสัญญาณโดยใช้ 5G – LDPC Encoder Downlink/Uplink | 208 |
| รูปที่ 4.220 หน้าแรกของส่วนต่อประสานกราฟิกกับผู้ใช้ | 209 |
| รูปที่ 4.221 การเลือกชนิดของตัวเข้ารหัสกรณิรหัสช่องสัญญาณชนิด Polar Codes | 209 |
| รูปที่ 4.222 หน้าต่างแสดงรายการพารามิเตอร์กรณิรการเข้ารหัสช่องสัญญาณโดยใช้ 5G Polar Encoder – Downlink | 209 |
| รูปที่ 4.223 หน้าต่างแสดงรายการพารามิเตอร์กรณิรการเข้ารหัสช่องสัญญาณโดยใช้ 5G Polar Encoder – Uplink | 211 |
| รูปที่ 4.224 หน้าต่างแสดงรายการพารามิเตอร์กรณิรการเข้ารหัสช่องสัญญาณโดยใช้ 5G Polar Encoder – Broadcast | 212 |
| รูปที่ 4.225 การเลือกวิธีการเข้ารหัส | 213 |
| รูปที่ 4.226 ผลลัพธ์การเข้ารหัสช่องสัญญาณโดยใช้ 5G Polar Encoder – Downlink | 213 |
| รูปที่ 4.227 ผลลัพธ์การเข้ารหัสช่องสัญญาณโดยใช้ 5G Polar Encoder – Uplink | 214 |
| รูปที่ 4.228 ผลลัพธ์การเข้ารหัสช่องสัญญาณโดยใช้ 5G Polar Encoder – Broadcast | 214 |
| รูปที่ 4.229 หน้าแรกของส่วนต่อประสานกราฟิกกับผู้ใช้ | 215 |
| รูปที่ 4.230 ผู้ใช้งานเลือกวิธีการเข้ารหัส “5G LDPC Encoder - Downlink/Uplink” และเลือกตัวเข้ารหัส “FPGA” | 216 |
| รูปที่ 4.231 ผู้ใช้งานรอกค่าพารามิเตอร์และอินพุตในส่วนการรับค่าพารามิเตอร์และอินพุต สำหรับรหัส LDPC | 216 |
| รูปที่ 4.232 การกดปุ่ม “Key1” เพื่อล้างค่าที่ค้างในวงจรเข้ารหัส FPGA | 217 |
| รูปที่ 4.233 การกดปุ่ม “RUN” เพื่อส่งข้อมูลจากคอมพิวเตอร์ไปยังวงจรเข้ารหัส FPGA | 217 |

| | | |
|--------------|--|-----|
| รูปที่ 4.234 | หลอด LED 8 ดวงทางขวาติด เพื่อป้องกันบวมถึงการเข้ารหัสในวงจรเข้ารหัส FPGA เสร็จสิ้น | 218 |
| รูปที่ 4.235 | การกดปุ่ม “Key0” เพื่อส่งผลลัพธ์จากวงจรเข้ารหัส FPGA ไปยังคอมพิวเตอร์ | 218 |
| รูปที่ 4.236 | ส่วนต่อประสานกราฟิกกับผู้ใช้แสดงผลของแต่ละกระบวนการเข้ารหัส LDPC | 219 |
| รูปที่ 4.237 | หน้าแรกของส่วนต่อประสานกราฟิกกับผู้ใช้ | 220 |
| รูปที่ 4.238 | ผู้ใช้งานเลือกวิธีการเข้ารหัส “5G Polar Encoder - Broadcast” และเลือกตัวเข้ารหัส “FPGA” | 221 |
| รูปที่ 4.239 | ผู้ใช้งานเลือกวิธีการเข้ารหัส “5G Polar Encoder - Downlink” และเลือกตัวเข้ารหัส “FPGA” | 221 |
| รูปที่ 4.240 | ผู้ใช้งานเลือกวิธีการเข้ารหัส “5G Polar Encoder - Uplink” และเลือกตัวเข้ารหัส “FPGA” | 222 |
| รูปที่ 4.241 | ผู้ใช้งานรอกค่าพารามิเตอร์และอินพุตในส่วนการรับค่าพารามิเตอร์ และอินพุต สำหรับรหัส Polar ช่องสัญญาณ Broadcast | 222 |
| รูปที่ 4.242 | ผู้ใช้งานรอกค่าพารามิเตอร์และอินพุตในส่วนการรับค่าพารามิเตอร์และอินพุต สำหรับรหัส Polar ช่องสัญญาณ Downlink | 223 |
| รูปที่ 4.243 | ผู้ใช้งานรอกค่าพารามิเตอร์และอินพุตในส่วนการรับค่าพารามิเตอร์และอินพุต สำหรับรหัส Polar ช่องสัญญาณ Uplink | 223 |
| รูปที่ 4.244 | การกดปุ่ม “Key1” เพื่อล้างค่าที่ค้างในวงจรเข้ารหัส FPGA | 224 |
| รูปที่ 4.245 | การกดปุ่ม “RUN” เพื่อส่งข้อมูลจากคอมพิวเตอร์ไปยังวงจรเข้ารหัส FPGA สำหรับรหัส Polar ช่องสัญญาณ Broadcast | 224 |
| รูปที่ 4.246 | การกดปุ่ม “RUN” เพื่อส่งข้อมูลจากคอมพิวเตอร์ไปยังวงจรเข้ารหัส FPGA สำหรับรหัส Polar ช่องสัญญาณ Downlink | 225 |
| รูปที่ 4.247 | การกดปุ่ม “RUN” เพื่อส่งข้อมูลจากคอมพิวเตอร์ไปยังวงจรเข้ารหัส FPGA สำหรับรหัส Polar ช่องสัญญาณ Uplink | 225 |
| รูปที่ 4.248 | หลอด LED 8 ดวงทางขวาติด เพื่อป้องกันบวมถึงการเข้ารหัสในวงจรเข้ารหัส FPGA เสร็จสิ้น | 226 |
| รูปที่ 4.249 | การกดปุ่ม “Key0” เพื่อส่งผลลัพธ์จากวงจรเข้ารหัส FPGA ไปยังคอมพิวเตอร์ | 226 |
| รูปที่ 4.250 | ส่วนต่อประสานกราฟิกกับผู้ใช้แสดงผลของแต่ละกระบวนการเข้ารหัส Polar ช่องสัญญาณ Broadcast | 227 |
| รูปที่ 4.251 | ส่วนต่อประสานกราฟิกกับผู้ใช้แสดงผลของแต่ละกระบวนการเข้ารหัส Polar ช่องสัญญาณ Downlink | 228 |
| รูปที่ 4.252 | ส่วนต่อประสานกราฟิกกับผู้ใช้แสดงผลของแต่ละกระบวนการเข้ารหัส Polar ช่องสัญญาณ Uplink | 228 |

บทที่ 1

บทนำ

1.1 ที่มา และความสำคัญของโครงการ

การสื่อสารไร้สายจัดเป็นองค์ประกอบสำคัญของอุปกรณ์ไอทีในปัจจุบัน สังเกตได้จากอุปกรณ์ไอทีต่าง ๆ เช่น โทรศัพท์มือถือ โน้ตบุ๊ก หรืออุปกรณ์ IoT ต่าง ๆ จะมีส่วนประกอบหนึ่งก็คือการสื่อสารไร้สายเสมอ การสื่อสารไร้สายที่สำคัญของมนุษย์ก็คือการสื่อสารผ่านโครงข่ายโทรศัพท์มือถือ โดยจะพบว่าพฤติกรรมของมนุษย์เปลี่ยนแปลงไปจากอดีตเมื่อเทคโนโลยีสื่อสารไร้สายยุค 4G ถูกนำมาใช้อย่างกว้างขวาง เทคโนโลยีสื่อสารไร้สายในยุค 5G นั้นถือเป็นวิวัฒนาการครั้งสำคัญของระบบสื่อสารโทรคมนาคมของโลก เนื่องจากการสื่อสารจะไม่จำกัดอยู่แต่เพียงโทรศัพท์มือถือเท่านั้น แต่จะขยายขอบเขตไปยังอุปกรณ์ต่าง ๆ จำพวกอุปกรณ์อินเทอร์เน็ตของสรรพสิ่ง (Internet of Things, IoT) ซึ่งการสื่อสารระหว่างอุปกรณ์ในยุค 5G จะเป็นแรงผลักดันไปสู่รูปแบบหรือมิติใหม่ของมนุษย์ในการดำรงชีวิต การทำงาน และความสัมพันธ์ระหว่างกันและกัน รูปแบบการใช้งานหรือ use case ที่จะเกิดขึ้น เช่น การรับชมวิดีโอสตรีมมิงความคมชัดสูง การใช้งานยานพาหนะไร้คนขับ การเชื่อมต่ออุปกรณ์ IoT หลายพันล้านตัวเข้ากับโครงสร้างพื้นฐานคลาวด์นำไปสู่การให้บริการรูปแบบใหม่ ๆ ที่ชาญฉลาดและมีประสิทธิภาพ สามารถตอบสนองกับวิถีชีวิตในยุคของการเปลี่ยนแปลงจากเทคโนโลยีดิจิทัล หนึ่งในเทคโนโลยีที่สำคัญของการสื่อสารไร้สายยุค 3G 4G และ 5G คือ เทคโนโลยีการเข้ารหัสช่องสัญญาณที่ทำให้การสื่อสารระหว่างอุปกรณ์ปราศจากความผิดพลาด เทคโนโลยีการเข้ารหัสช่องสัญญาณมีหน้าที่ปกป้องบิตข้อมูลจากสัญญาณรบกวนต่าง ๆ ในระบบสื่อสาร ในมาตรฐาน 5G นั้นได้มีการเปลี่ยนแปลงเทคโนโลยีการเข้ารหัสช่องสัญญาณครั้งใหญ่เนื่องจากข้อกำหนดของ ITU ดังนั้น จึงเป็นที่มาของโครงการที่นำเสนอนี้ ซึ่งมีเป้าหมายคือ

1) มิติด้านการศึกษาวิศวกรรมโทรคมนาคม

ทีมวิจัยนี้ได้สังเกตเห็นปัญหาสำคัญของระบบการศึกษาทั่วโลก ที่เทคโนโลยีการสื่อสารปัจจุบันมีความซับซ้อนมากกว่าเนื้อหาในวิชาเรียน ปัญหาดังกล่าวเกิดจากอุปกรณ์การสอนสมัยใหม่ที่มีราคาแพงและต้องนำเข้าจากต่างประเทศ อีกทั้ง ผู้ผลิตอุปกรณ์การสอนมักจะเน้นพัฒนาอุปกรณ์การสอนสำหรับวิชาพื้นฐานทางวิศวกรรมโทรคมนาคม ทำให้ขาดอุปกรณ์การสอนที่สอดคล้องกับระบบการสื่อสารปัจจุบัน ดังนั้น ทีมวิจัยจึงต้องการพัฒนาอุปกรณ์ที่สามารถสนับสนุนการเรียนรู้ของนักศึกษาสาขาวิชาวิศวกรรมโทรคมนาคมหรือสาขาอื่น ๆ ที่เกี่ยวข้อง โดยทีมวิจัยได้เลือกเทคโนโลยีการเข้ารหัสช่องสัญญาณในมาตรฐาน 5G ซึ่งมีการเปลี่ยนแปลงจากมาตรฐาน 3G และ 4G ไปมาก จึงเกิด

เป็นข้อเสนอโครงการที่ต้องการพัฒนาอุปกรณ์เข้ารหัสช่องสัญญาณที่นักศึกษาสามารถเรียนรู้ผลลัพธ์ แต่ละขั้นตอนของการเข้ารหัสช่องสัญญาณตามมาตรฐาน 5G ได้โดยง่าย

2) มิติของการพัฒนาอุปกรณ์โทรคมนาคมต้นแบบ

หนึ่งในปัญหาสำคัญของอุตสาหกรรมโทรคมนาคมในประเทศไทย คือ ศักยภาพของการพัฒนาอุปกรณ์โทรคมนาคมที่มีความซับซ้อนสูง ที่มิวิจยสังเกตเห็นว่าการพัฒนาอุปกรณ์โทรคมนาคมในระดับการสร้างและขายอุปกรณ์โทรคมนาคม จำเป็นต้องแข่งขันกับประเทศต่าง ๆ ที่มีความพร้อมในด้านแรงงานและงบประมาณที่สูง ดังนั้น การพัฒนาอุปกรณ์โทรคมนาคมในระดับขาย IP หรือทรัพย์สินทางปัญญาที่ด้านการออกแบบโครงการสร้างภายในมีความเป็นไปได้ในการแข่งขันกับประเทศต่าง ๆ มากกว่า ที่มิวิจยจึงต้องการพัฒนาโครงสร้างการเข้ารหัสช่องสัญญาณตามมาตรฐาน 5G เพื่อให้เป็นอุปกรณ์ต้นแบบในการนำไปสู่การพัฒนาโครงสร้างที่มีประสิทธิภาพสูงสำหรับการใช้งานเชิงพาณิชย์

1.2 วัตถุประสงค์

- 1) เพื่อพัฒนาอุปกรณ์สื่อการสอนของการเข้ารหัสช่องสัญญาณตามมาตรฐาน 5G
- 2) เพื่อพัฒนาอุปกรณ์ต้นแบบของการเข้ารหัสช่องสัญญาณตามมาตรฐาน 5G

1.3 ขอบเขตของโครงการ

โครงการสามารถแบ่งออกเป็น 4 ส่วนหลัก ได้แก่ 1) ชุดซอฟต์แวร์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G 2) ชุดอุปกรณ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G 3) ชุดการเรียนรู้การเข้ารหัสช่องสัญญาณมาตรฐาน 5G และ 4) เว็บไซต์เผยแพร่ความรู้และชุดซอฟต์แวร์และอุปกรณ์เข้ารหัสช่องสัญญาณมาตรฐาน 5G รายละเอียดสามารถอธิบายโดยสังเขป ดังนี้

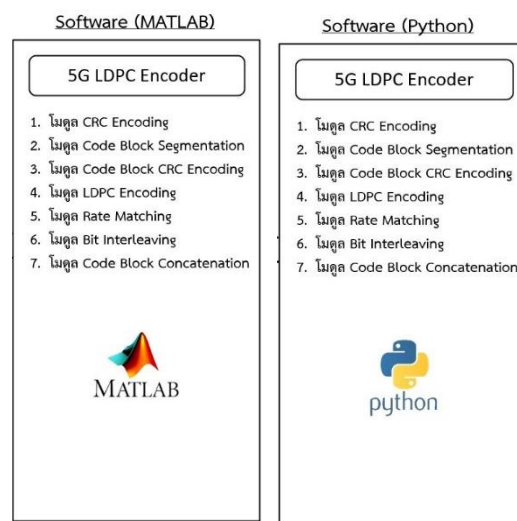
1) การพัฒนาชุดซอฟต์แวร์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G

ในส่วนแรกนี้ ที่มิวิจยเลือกใช้ภาษา MATLAB และ Python ในการพัฒนาชุดซอฟต์แวร์สำหรับเข้ารหัสช่องสัญญาณมาตรฐาน 5G ซึ่งประกอบด้วย 4 ส่วน ได้แก่

- 5G LDPC Encoder สำหรับช่องสัญญาณ Downlink และ Uplink
- 5G Polar Encoder สำหรับช่องสัญญาณ Downlink
- 5G Polar Encoder สำหรับช่องสัญญาณ Uplink
- 5G Polar Encoder สำหรับช่องสัญญาณ Broadcast

ซึ่งเป็นไปตามมาตรฐานของ 3GPP ที่กำหนดใช้ในระบบสื่อสารไร้สายยุค 5G โดยการพัฒนาชุดซอฟต์แวร์ 5G LDPC Encoder สำหรับช่องสัญญาณ Downlink และ Uplink จะประกอบไปด้วยการทำงานทั้งหมด 7 ส่วน ซึ่งแสดงในรูปที่ 1 ได้แก่ การเข้ารหัส CRC การแบ่งข้อมูลเป็นบล็อกย่อย

การเข้ารหัส CRC ให้บล็อกข้อมูลย่อย การเข้ารหัส LDPC ให้บล็อกข้อมูลย่อย การปรับอัตรารหัสโดย การเลือกบิตที่จะส่ง การแทรกสลับบิตข้อมูลของบล็อกข้อมูลย่อย และการนำบล็อกข้อมูลย่อยมาต่อ เรียงและส่งออก การพัฒนาชุดซอฟต์แวร์ 5G Polar Encoder สำหรับช่องสัญญาณ Downlink Uplink และ Broadcast จะประกอบไปด้วยการทำงานทั้งหมด 11 ส่วน ซึ่งแสดงในรูปที่ 1.1 ได้แก่ การแบ่งย่อยบล็อกรหัส การเข้ารหัส CRC การสแครมบลิง CRC การแทรกสลับ CRC การลำดับ ช่องสัญญาณย่อย การคำนวณบิตพาริตี การเข้ารหัส Polar การแทรกสลับบล็อกย่อย การปรับอัตรา รหัส การแทรกสลับบิตรหัส และการต่อบล็อกรหัส



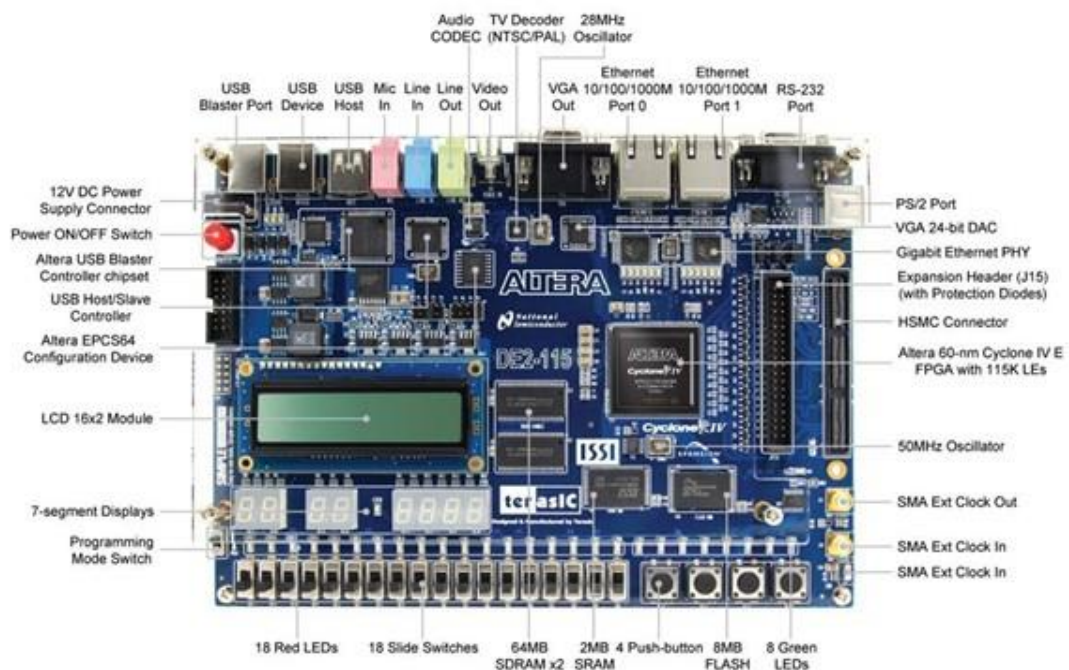
รูปที่ 1.1 ตัวอย่างการพัฒนาชุดซอฟต์แวร์การเข้ารหัสช่องสัญญาณมาตรฐานโดยใช้ภาษา MATLAB และ Python

2) การพัฒนาชุดอุปกรณ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G

ในส่วนที่สอง ทีมวิจัยจะพัฒนาชุดอุปกรณ์เข้ารหัสช่องสัญญาณมาตรฐาน 5G บนอุปกรณ์ FPGA โดยใช้ภาษา VHSIC Hardware Description Language (VHDL) ซึ่งประกอบไปด้วยวงจรต่อไปนี้

- 5G LDPC Encoder สำหรับช่องสัญญาณ Downlink และ Uplink
- 5G Polar Encoder สำหรับช่องสัญญาณ Downlink
- 5G Polar Encoder สำหรับช่องสัญญาณ Uplink
- 5G Polar Encoder สำหรับช่องสัญญาณ Broadcast

โดยการพัฒนาวงจรในข้างต้น จะมีขั้นตอนที่สอดคล้องกับชุดซอฟต์แวร์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ที่มิจิวิจัยเลือกใช้บอร์ด FPGA รุ่น Altera DE2-115 แสดงในรูปที่ 1.2 ซึ่งประกอบไปด้วยอินเทอร์เฟซที่รองรับการเชื่อมต่อที่หลากหลาย บอร์ด DE2-115 มีชิพ Cyclone EP4CE115 ที่ประกอบไปด้วย 114,480 ลอจิก อีกทั้งบนบอร์ด DE2-115 ยังประกอบไปด้วยหน่วยความจำ สวิตช์ จอแสดงผล LCD ซึ่งเหมาะสมในการนำไปใช้ประกอบการเรียนการสอนของวิศวกรรมโทรคมนาคม



รูปที่ 1.2 บอร์ด FPGA รุ่น Altera DE2-115

3) การพัฒนาชุดการเรียนรู้การเข้ารหัสช่องสัญญาณมาตรฐาน 5G

ที่มิจิวิจัยที่มิจิวิจัยเลือกใช้ภาษา Python เพื่อสร้างส่วนต่อประสานกับผู้ใช้ (Graphic User Interface: GUI) ในการแสดงผลการเข้ารหัสช่องสัญญาณจากชุดซอฟต์แวร์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ในหัวข้อที่ 1 รวมถึงผลลัพธ์การเข้ารหัสช่องสัญญาณจากชุดอุปกรณ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ในหัวข้อที่ 2 ดังนี้

ส่วนประสานผู้ใช้ (GUI) ที่รับผลลัพธ์จากชุดซอฟต์แวร์

ที่มิจิวิจัยออกแบบหน้าต่าง GUI เพื่อแสดงผลการเข้ารหัสช่องสัญญาณจากชุดซอฟต์แวร์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ที่พัฒนาด้วยภาษา Python โดยหน้าต่าง GUI ถูกออกแบบให้ผู้ใช้

สามารถใช้ได้อย่างง่ายดาย ผู้ใช้สามารถเลือกรูปแบบการเข้ารหัสที่ต้องการในหัวข้อ 5G Channel Encoder และป้อนค่าพารามิเตอร์ที่จำเป็นในหัวข้อ Configuration parameters

กรณีผู้ใช้งานเลือกการเข้ารหัสแบบ 5G LDPC Encoder – Downlink/Uplink ดังแสดงในรูปแบบที่ 1.3 ข้อมูลอินพุตที่กรอกลงไปจะถูกนำไปเข้าสู่กระบวนการทำงานของโมดูลต่าง ๆ เช่น โมดูล CRC Encoding โมดูล Code Block Segmentation โมดูล Code Block CRC Encoding โมดูล LDPC Encoding โมดูล Rate Matching โมดูล Bit Interleaving และโมดูล Code Block Concatenation ผลลัพธ์ที่ได้จากแต่ละโมดูลจะแสดงผลแก่ผู้ใช้ในรูปแบบบล็อกไดอะแกรมบนหน้าต่าง GUI ในหัวข้อ Block Diagram of 5G Channel Encoder

กรณีผู้ใช้งานเลือกการเข้ารหัส 5G Polar Encoder ในรูปแบบ Downlink ดังแสดงในรูปแบบที่ 1.4 โดยโปรแกรมจะเรียกใช้งานฟังก์ชัน 5G Polar Encoder ซึ่งประกอบไปด้วยโมดูล CRC Encoding โมดูล CRC Scrambling โมดูล CRC Interleaving โมดูล Polar Sequencing โมดูล Polar Encoding โมดูล Sub Block Interleaving และโมดูล Rate Matching

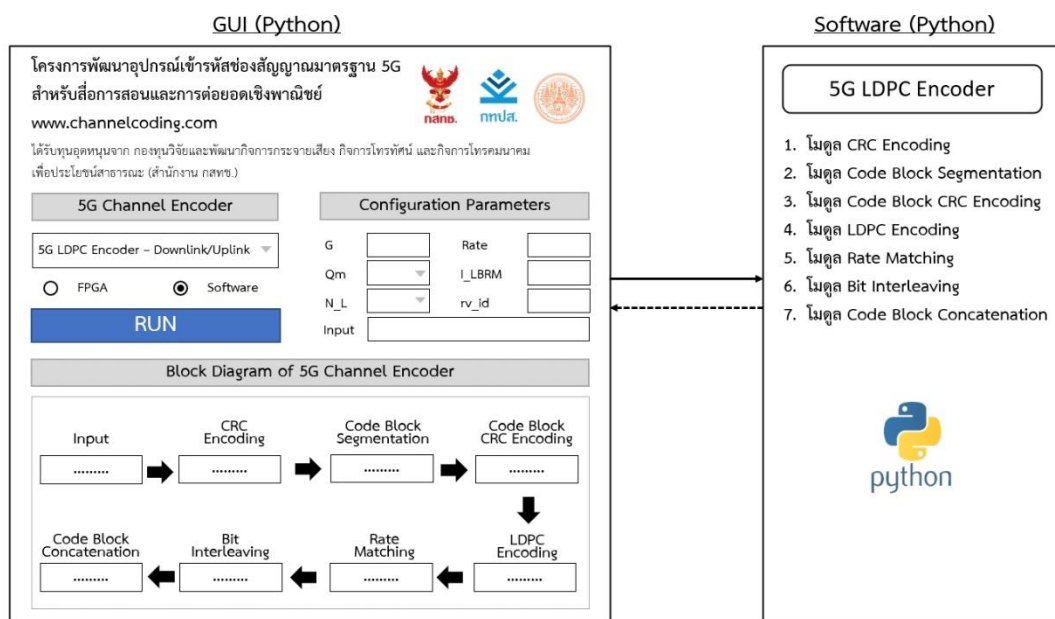
กรณีผู้ใช้งานเลือกการเข้ารหัส 5G Polar Encoder ในรูปแบบ Uplink ดังแสดงในรูปแบบที่ 1.5 โดยโปรแกรมจะเรียกใช้งานฟังก์ชัน 5G Polar Encoder ในรูปแบบ Uplink ซึ่งประกอบไปด้วยการใช้งานโมดูลต่าง ๆ ที่เพิ่มขึ้นมาจากการใช้งานในรูปแบบ Downlink เช่น โมดูล Code Block Segmentation โมดูล Parity-Check โมดูล Code bit interleaving และโมดูล Code Block Concatenation โดยผลลัพธ์ของแต่ละโมดูลยังคงแสดงในรูปแบบบล็อกไดอะแกรมบนหน้าต่าง GUI

กรณีผู้ใช้งานเลือกการเข้ารหัส 5G Polar Encoder ในรูปแบบ Broadcast ดังแสดงในรูปแบบที่ 1.6 โดยโปรแกรมจะเรียกใช้งานฟังก์ชัน 5G Polar Encoder ในรูปแบบ Broadcast ซึ่งเมื่อผู้ใช้งานกดเลือกใช้งานในส่วนนี้ โปรแกรมจะเรียกใช้งานฟังก์ชัน 5G Polar Encoder ซึ่งประกอบไปด้วยโมดูลคล้ายกับการเข้ารหัสในรูปแบบ Downlink เช่น โมดูล CRC Encoding โมดูล CRC Interleaving โมดูล Polar Sequencing โมดูล Polar Encoding โมดูล Sub Block Interleaving และโมดูล Rate Matching

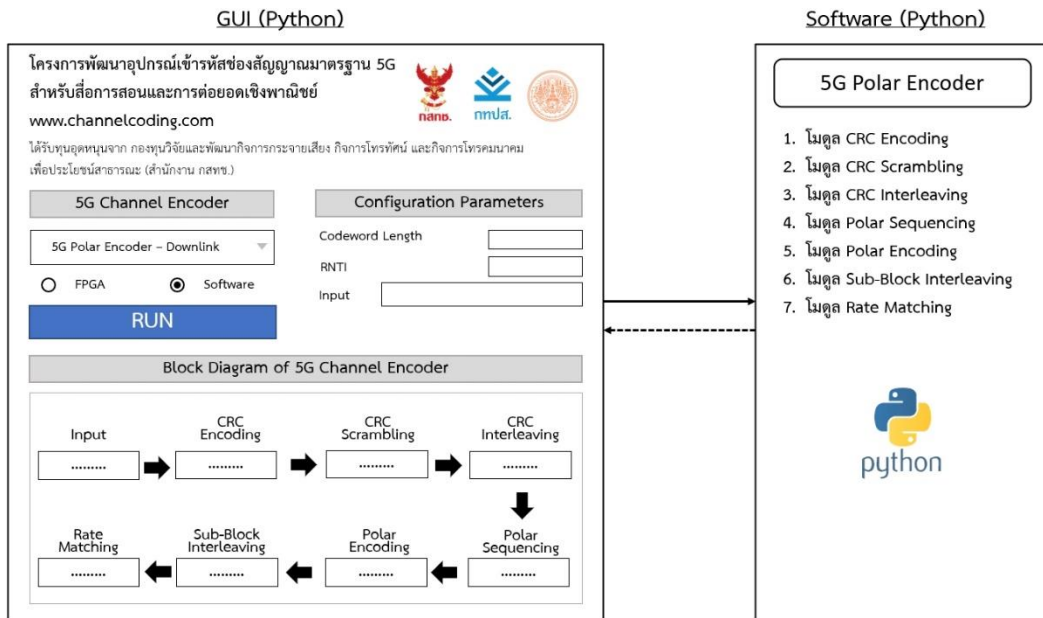
ส่วนประสานผู้ใช้ (GUI) ที่รับผลลัพธ์จากชุดอุปกรณ์ FPGA

ทีมวิจัยได้พัฒนาส่วนต่อประสานกับผู้ใช้ (GUI) เพื่อแสดงผลการเข้ารหัสจากอุปกรณ์ FPGA ทีมวิจัยได้ออกแบบหน้าต่าง GUI ให้คล้ายคลึงกับการแสดงผลจากชุดซอฟต์แวร์ Python ในกรณีนี้

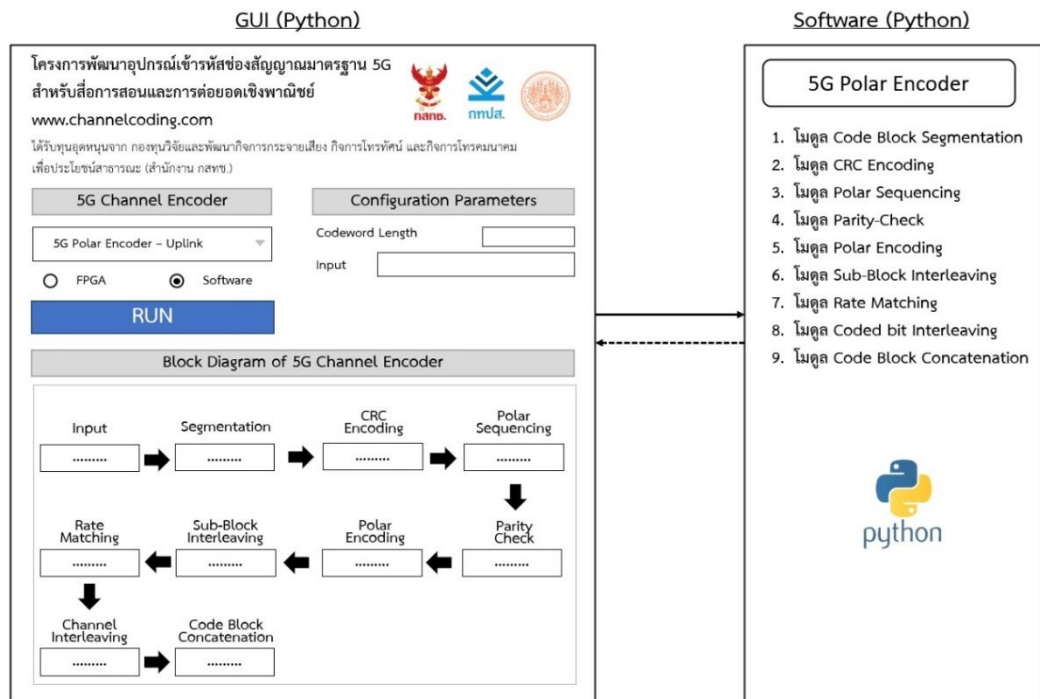
ผู้ใช้เลือกการเข้ารหัสแบบ 5G LDPC Encoder – Downlink/Uplink แสดงดังรูปที่ 1.7 บอร์ด FPGA จะใช้โมดูล 5G LDPC Encoder ที่เขียนด้วยภาษา VHDL ในการเข้ารหัสข้อมูลที่กำหนด หลังจากเข้ารหัสเสร็จสิ้น บอร์ด FPGA จะส่งผลลัพธ์ที่เกิดขึ้นในแต่ละขั้นตอนของกระบวนการเข้ารหัสผ่าน Serial Port กลับไปยังคอมพิวเตอร์ และแสดงผลลัพธ์แก่ผู้ใช้งานบน GUI ในทำนองเดียวกัน ผู้ใช้งานสามารถเลือกการเข้ารหัสแบบ 5G Polar Encoder เพื่อทดสอบผลลัพธ์การเข้ารหัส Polar ในรูปแบบ Downlink Uplink และส่วน Broadcast ดังแสดงดังแสดงในรูปที่ 1.8 1.9 และ 1.10 ตามลำดับ



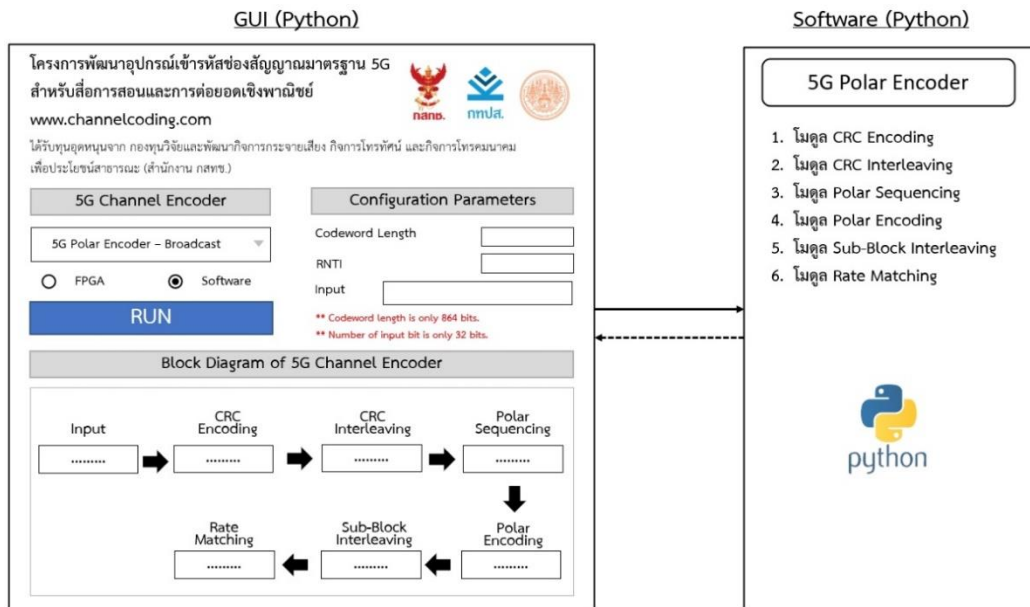
รูปที่ 1.3 ส่วนต่อประสานกับผู้ใช้ GUI รับข้อมูลจากชุดซอฟต์แวร์ Python (ส่วน 5G LDPC Encoder – Downlink/Uplink)



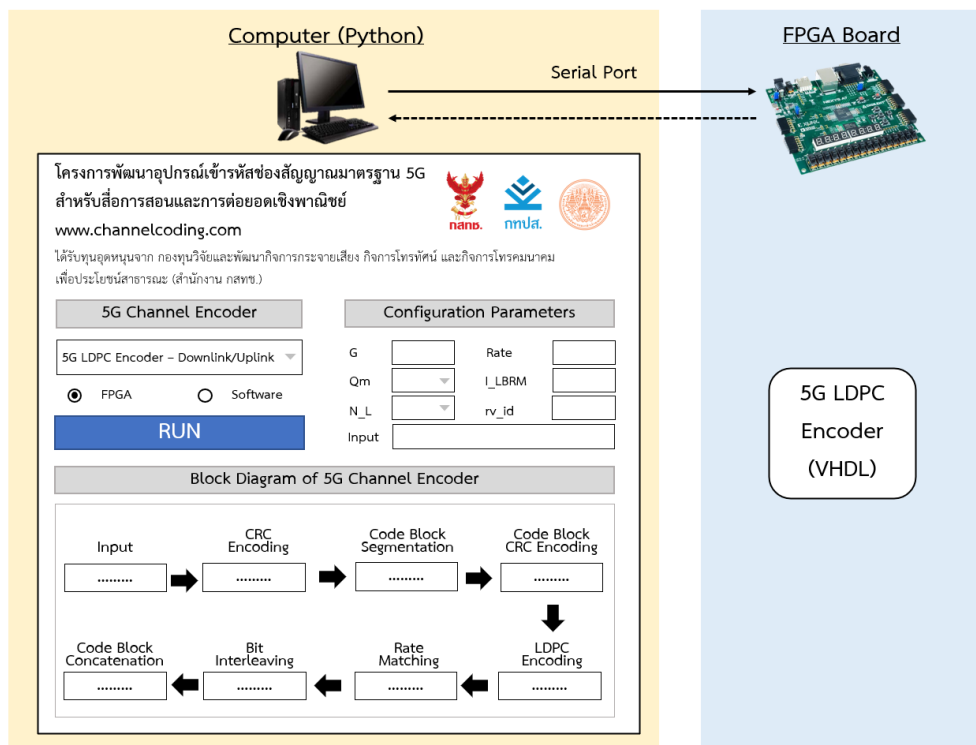
รูปที่ 1.4 ส่วนต่อประสานกับผู้ใช้ GUI รับข้อมูลจากชุดซอฟต์แวร์ Python
(ส่วน 5G Polar Encoder – Downlink)



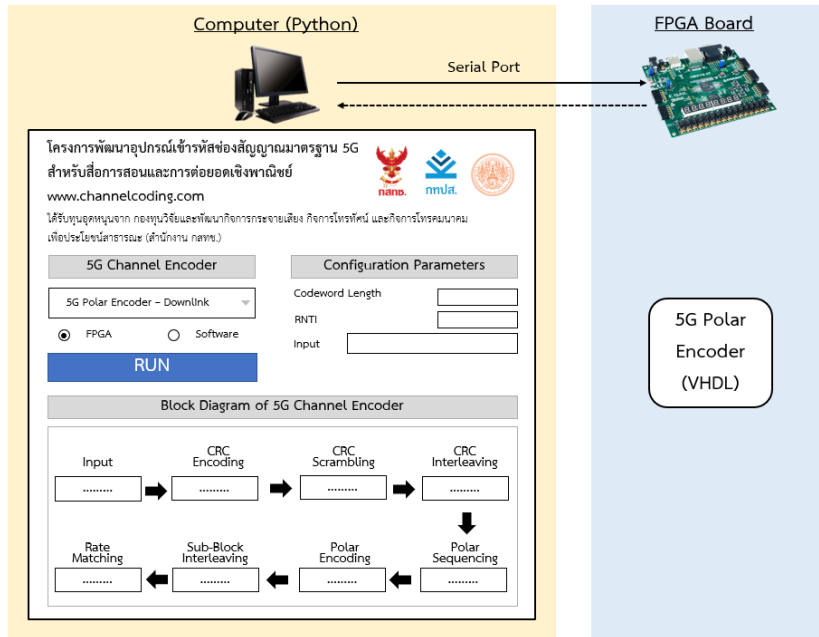
รูปที่ 1.5 ส่วนต่อประสานกับผู้ใช้ GUI รับข้อมูลจากชุดซอฟต์แวร์ Python
(ส่วน 5G Polar Encoder – Uplink)



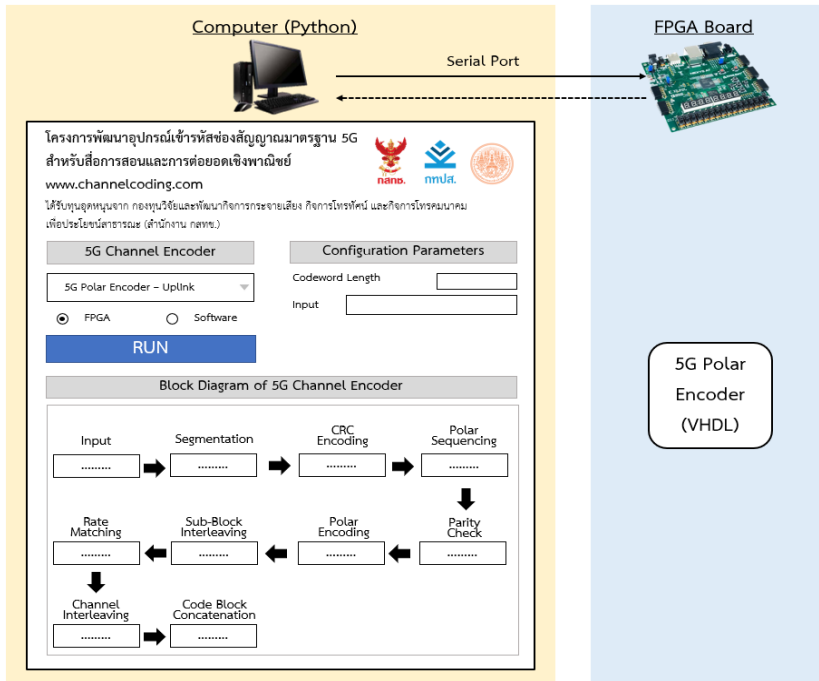
รูปที่ 1.6 ส่วนต่อประสานกับผู้ใช้ GUI รับข้อมูลจากชุดซอฟต์แวร์ Python (ส่วน 5G Polar Encoder – Broadcast)



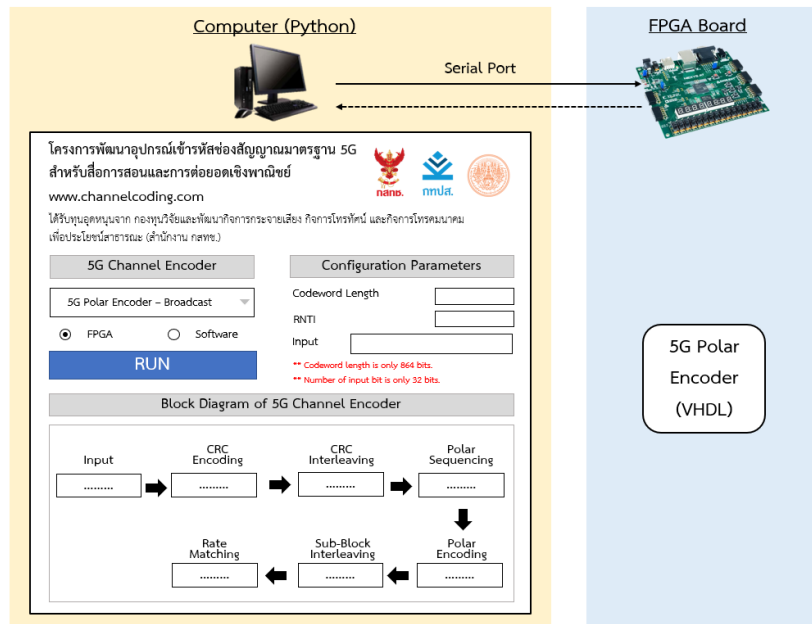
รูปที่ 1.7 ส่วนต่อประสานกับผู้ใช้ GUI ที่รับข้อมูลจากชุดอุปกรณ์ FPGA (ส่วน 5G LDPC Encoder – Downlink/Uplink)



รูปที่ 1.8 ส่วนต่อประสานกับผู้ใช้ GUI ที่รับข้อมูลจากชุดอุปกรณ์ FPGA (ส่วน 5G Polar Encoder – Downlink)



รูปที่ 1.9 ส่วนต่อประสานกับผู้ใช้ GUI ที่รับข้อมูลจากชุดอุปกรณ์ FPGA (ส่วน 5G Polar Encoder – Uplink)



รูปที่ 1.10 ส่วนต่อประสานกับผู้ใช้ GUI ที่รับข้อมูลจากชุดอุปกรณ์ FPGA (ส่วน 5G Polar Encoder – Broadcast)

4) การพัฒนาเว็บไซต์เผยแพร่ความรู้การเข้ารหัสช่องสัญญาณมาตรฐาน 5G
 ทีมวิจัยได้พัฒนาเว็บไซต์ www.channelcoding.com ดังรูปที่ 11 เพื่อนำเสนอการใช้งาน “รหัสช่องสัญญาณ” ในมาตรฐานการสื่อสารต่าง ๆ เช่น มาตรฐาน 5G (3GPP NR) และ Wi-Fi (IEEE 802.11) รวมถึงการเผยแพร่ความรู้และชุดซอฟต์แวร์และอุปกรณ์เข้ารหัสช่องสัญญาณมาตรฐาน 5G



รูปที่ 1.11 เว็บไซต์ www.channelcoding.com

1.4 ประโยชน์ที่คาดว่าจะได้รับ

- 2) มหาวิทยาลัยที่จัดการเรียนการสอนด้านวิศวกรรมโทรคมนาคม ได้รับอุปกรณ์และซอฟต์แวร์ การเข้ารหัสช่องสัญญาณที่ทันสมัยและตรงตามมาตรฐาน 3GPP release 15 หรือ 5G new radio
- 3) ผู้ประกอบการด้านอุปกรณ์การสอน ได้เห็นแนวทางการพัฒนาอุปกรณ์การสอนที่สอดคล้องกับความต้องการของมหาวิทยาลัย
- 4) ผู้ประกอบการด้านอุปกรณ์โทรคมนาคมได้อุปกรณ์เข้ารหัสช่องสัญญาณต้นแบบที่นำไปสู่การพัฒนาโครงสร้างแบบใหม่ๆ เพื่อใช้ในจุด IP หรือทรัพยากรสิ้นทางปัญญา

บทที่ 2

ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

2.1 พื้นฐานรหัสช่องสัญญาณ

รหัสช่องสัญญาณเป็นองค์ประกอบสำคัญในระบบสื่อสารดิจิทัลที่ขาดไม่ได้ เนื่องจากรหัสช่องสัญญาณทำให้การรับส่งข้อมูลดิจิทัลปราศจากความผิดพลาด เทคโนโลยีเครือข่ายโทรศัพท์เคลื่อนที่ยุคที่ 4 หรือ 4G ได้ประยุกต์ใช้รหัสช่องสัญญาณที่เรียกว่า turbo codes ใน data channel และ tail-biting convolutional codes (TBCC) ใน control channel [1] รหัสช่องสัญญาณ turbo codes เป็นรหัสช่องสัญญาณที่มีความน่าสนใจอย่างยิ่ง เนื่องจากเป็นรหัสช่องสัญญาณชนิดแรกที่มีสมรรถนะเข้าใกล้ทฤษฎีความจุช่องสัญญาณ ซึ่งเป็นเหตุผลสำคัญที่ทำให้ turbo code ถูกประยุกต์ใช้ในมาตรฐาน 3G และ 4G (รหัสช่องสัญญาณ turbo codes ถูกคิดค้นโดย Claude Berrou Alain Glavieux และ รศ.ดร.ปัญญา รุติมีชฌิมา [3] ในปี ค.ศ. 1993) จากการค้นพบ turbo codes ทำให้นักวิจัยทั่วโลกทำการศึกษารหัสช่องสัญญาณอื่น ๆ ที่มีสมรรถนะที่ใกล้เคียงหรือดีกว่า turbo codes จนกระทั่งค้นพบว่ารหัสช่องสัญญาณ low-density parity-check codes (LDPC codes) มีสมรรถนะเข้าใกล้ทฤษฎีความจุช่องสัญญาณเช่นเดียวกับ turbo codes (รหัสช่องสัญญาณ LDPC codes ถูกคิดค้นโดย Robert Gallager [4] ตั้งแต่ปี ค.ศ. 1962 แต่ในช่วงเวลาดังกล่าวประสิทธิภาพของระบบคอมพิวเตอร์ยังไม่ดีเพียงพอสำหรับการจำลองหาสมรรถนะการแก้ไขบิตผิดของ LDPC codes) แม้ว่า LDPC codes จะมีสมรรถนะเข้าใกล้ทฤษฎีความจุช่องสัญญาณเช่นเดียวกับ turbo codes แต่การพัฒนางจรเข้ารหัสและถอดรหัสนของ LDPC codes ให้มีความซับซ้อนที่เหมาะสมกับการประยุกต์ใช้งานยังเป็นโจทย์สำคัญที่ต้องได้รับการพัฒนาในช่วงเวลาดังกล่าว ปัจจุบัน รหัสช่องสัญญาณ LDPC codes ได้รับการพัฒนาให้วงจรเข้ารหัสและถอดรหัสมีความซับซ้อนต่ำ ทำให้ LDPC codes ถูกนำมาใช้งานในระบบสื่อสารมาตรฐานต่าง ๆ เช่น ระบบเครือข่ายคอมพิวเตอร์ไร้สายมาตรฐาน IEEE 802.11 [5] ระบบโทรทัศน์ดิจิทัลมาตรฐาน DVB-S2 และ DVB-T2 [6] ทำให้ การออกแบบรหัสช่องสัญญาณในมาตรฐาน 5G รหัสช่องสัญญาณ LDPC codes จะถูกนำมาใช้หรือไม่ เป็นประเด็นที่ได้รับความสนใจจากนักวิจัยทั่วโลก นอกจากนี้ รหัสช่องสัญญาณ polar codes ซึ่งเป็นรหัสช่องสัญญาณที่สามารถพิสูจน์ได้ว่ามีสมรรถนะเข้าใกล้ทฤษฎีความจุช่องสัญญาณก็เป็นอีกตัวเลือกที่น่าสนใจสำหรับการประยุกต์ใช้ในมาตรฐาน 5G (รหัสช่องสัญญาณ polar codes ถูกคิดค้นโดย Erdal Arkan [7] ในปี ค.ศ. 2009 ได้รับความสนใจจากนักวิจัยจำนวนมากเนื่องจากการวิธีการเข้ารหัส polar codes สามารถพิสูจน์ได้โดยตรงว่ามีสมรรถนะ

เข้าใกล้ทฤษฎีความจุช่องสัญญาณ ต่างจาก turbo codes และ LDPC codes ที่ไม่สามารถพิสูจน์ได้ โดยตรงว่ามีสมรรถนะเข้าใกล้ทฤษฎีความจุช่องสัญญาณ

2.2 การออกแบบมาตรฐานการเข้ารหัสช่องสัญญาณของ 3GPP

สหภาพโทรคมนาคมระหว่างประเทศ (international telecommunication union, ITU) ได้แบ่งรูปแบบการใช้งานเครือข่ายโทรศัพท์เคลื่อนที่ยุคที่ 5 หรือ 5G ไว้ 3 ประเภท [8] ได้แก่ enhanced mobile broadband (eMBB), ultra-reliable and low latency communications (URLLC) และ massive machine type communications (mMTC) สำหรับการใช้งานแบบ eMBB จะมุ่งเน้นการใช้งานในลักษณะที่ต้องการการส่งข้อมูลความเร็วสูงในระดับกิกะบิตต่อวินาที ตัวอย่างเช่น การรับชมวิดีโอความละเอียด 4K หรือการสื่อสารแบบเรียลไทม์ เป็นต้น สำหรับการใช้งานแบบ URLLC ซึ่งมุ่งเน้นไปที่การส่งข้อมูลที่มีความเสถียรมาก มีความหน่วงในการส่งข้อมูลต่ำ ตัวอย่างเช่น การรับส่งข้อมูลระหว่างรถยนต์ที่ต้องการรับส่งข้อมูลที่มีการหน่วงเวลาต่ำเพื่อตอบสนองเหตุการณ์ต่าง ๆ สำหรับการใช้งานแบบ mMTC ซึ่งมุ่งเน้นการเชื่อมต่ออุปกรณ์จำนวนมากในบริเวณพื้นที่จำกัด ตัวอย่างเช่น การรับส่งข้อมูลของอุปกรณ์ IoT

จากรูปแบบการใช้งานทั้ง 3 ที่ถูกกำหนดโดย ITU จะเห็นได้ว่าการสื่อสารยุคไร้สายยุค 5G ถูกคาดหวังว่าจะเป็นยุคของการสื่อสารกับทุกสรรพสิ่ง ไม่ว่าจะเป็นระหว่างมนุษย์กับมนุษย์ มนุษย์กับเครื่องจักร หรือเครื่องจักรกับเครื่องจักร การสื่อสารของสรรพสิ่งเหล่านี้ต่างก็เรียกร้องคุณภาพในการสื่อสารที่ดี กล่าวคือปราศจากความผิดพลาดในการส่งข้อมูล อย่างไรก็ตาม เนื่องจากในสภาพอากาศมีสิ่งรบกวนจำนวนมาก เช่น การรบกวนจากฝน สัญญาณรบกวนแบบจางหาย รวมถึงความร้อนจากวงจรอิเล็กทรอนิกส์ในอุปกรณ์รับ/ส่ง ดังนั้นการที่ระบบสื่อสารจะปราศจากข้อผิดพลาดเลยจึงเป็นไปได้ยาก โดยปกติแล้วในการส่งข้อมูลผ่านช่องสัญญาณที่คาดว่าจะมีความผิดพลาดเกิดขึ้นจะอาศัยสิ่งที่เรียกว่า “รหัสช่องสัญญาณ” หรือเรียกอีกชื่อหนึ่งว่า “รหัสแก้ไขความผิดพลาด” เพื่อแก้ไขความผิดพลาดในการส่งข้อมูล รหัสช่องสัญญาณเกิดจากการเข้ารหัสข้อมูลที่ภาคส่งโดยอาศัยวงจรเข้ารหัสเพื่อเพิ่มบิตตรวจสอบหรือที่เรียกกันว่า “บิตพาริตี (Parity bits)” เข้าไปกับข้อมูลต้นฉบับ เพื่อส่งข้อมูลเหล่านั้นไปยังภาครับ โดยหน้าที่ของภาครับคือการถอดรหัสข้อมูลที่ได้รับมา โดยอาศัยบิตพาริตีในการตรวจสอบแก้ไขความผิดพลาดที่เกิดขึ้น ในปัจจุบันนี้มีรหัสช่องสัญญาณให้เลือกใช้อย่างหลากหลายตามความเหมาะสมกับแอปพลิเคชันต่าง ๆ เช่น

- รหัสแฮมมิง (Hamming Codes: 1950) ถูกใช้ในหน่วยความจำของคอมพิวเตอร์ RAM และ DRAM
- รหัสคอนโวลูชัน (Convolution Codes: 1955) ถูกใช้ในการสื่อสารดาวเทียม

- รหัสเทอร์โบ (Turbo Codes: 1993) การคิดค้นรหัสดังกล่าวนี้ถือเป็นการเริ่มต้นของการสื่อสารยุคใหม่ เนื่องจากรหัสดังกล่าวได้ถูกพิสูจน์ว่ามีสมรรถนะที่เข้าใกล้ขีดจำกัดทางทฤษฎีหรือ “ขอบเขตของแชนนอน (Shannon’s limit)” รหัสดังกล่าวได้ถูกใช้กันอย่างแพร่หลายในโทรศัพท์มือถือยุค 3G - 4G รวมถึงยานอวกาศนิวฮอไรซอนส์ (New Horizons)
- รหัสแอลดีพีซี (LDPC codes: 1962) รหัส LDPC นี้ได้ถูกพิสูจน์ด้วยเช่นกันว่ามีสมรรถนะที่เข้าใกล้ขีดจำกัดของแชนนอนเช่นเดียวกันกับรหัส Turbo อีกทั้งยังมีความซับซ้อนในการเข้าและถอดรหัสที่ต่ำกว่ารหัส Turbo อีกด้วย ดังนั้นรหัส LDPC จึงถูกใช้กันอย่างแพร่หลายเป็นอย่างมากในปัจจุบัน เช่นใน Wireless Network, Digital broadcasting, Hard disk drive และ Solid-state drive
- รหัสโพลาร์ (Polar codes: 2009) เป็นรหัสที่เพิ่งถูกคิดค้นขึ้นในปี 2009 ซึ่งรหัสชนิดนี้สามารถพิสูจน์ได้ทางคณิตศาสตร์ว่ามีสมรรถนะเข้าใกล้ทฤษฎีความจุช่องสัญญาณเช่นเดียวกับรหัส Turbo และรหัส LDPC

ในส่วนและเทคโนโลยี 5G การสื่อสารระหว่างสถานีฐานและโทรศัพท์เคลื่อนที่จะประกอบไปด้วยข้อมูลอยู่ 2 ชนิด ชนิดแรกคือ User data ซึ่งเป็นข้อมูลที่แท้จริงของผู้ใช้งานที่พยายามส่งไปที่ผู้รับปลายทาง ชนิดที่สองคือ Control data ซึ่งเป็นข้อมูลที่โทรศัพท์เคลื่อนที่ส่งข้อความตอบกลับไปยังสถานีฐานว่าได้รับข้อมูลถูกต้องหรือไม่ หรือสภาพสัญญาณรบกวนบริเวณนั้นเป็นเช่นไร รหัสช่องสัญญาณที่จะถูกนำมาใช้ใน 5G ต้องมีความเหมาะสมสำหรับการใช้งานทั้ง 3 ด้านของ 5G ไม่ว่าจะเป็น eMBB, mMTC และ URLLC ดังนี้

- อัตราบิดผิดพลาดภายหลังการถอดรหัสต้องต่ำ คุณสมบัติพื้นฐานของรหัสช่องสัญญาณที่จะถูกนำมาใช้ในทุกแอปพลิเคชันรวมถึงการสื่อสารยุค 5G คือต้องมีสมรรถนะในการแก้ไขบิดผิดที่ดี เพราะฉะนั้นรหัสช่องสัญญาณที่มีสมรรถนะที่เข้าใกล้ความจุช่องสัญญาณเช่น รหัส Turbo รหัส LDPC และรหัส Polar จะถูกนำมาพิจารณาใช้สำหรับการสื่อสารยุค 5G เป็นอันดับแรก
- ความซับซ้อนในการออกแบบฮาร์ดแวร์ รหัสช่องสัญญาณที่นำมาใช้ต้องมีโครงสร้างฮาร์ดแวร์ที่ยอมรับได้ในเชิงปฏิบัติ เนื่องจากความซับซ้อนของฮาร์ดแวร์เกี่ยวข้องกับการใช้งานแบตเตอรี่ในอุปกรณ์ ซึ่งหากอุปกรณ์มีการใช้แบตเตอรี่มากเกินไปจะไม่เหมาะกับการนำมาใช้งานในอุปกรณ์ IoT รหัสช่องสัญญาณที่มีโครงสร้างการเข้าและถอดรหัสที่มีความซับซ้อนต่ำคือรหัส LDPC และ Polar

- วงจรถอดรหัสมีการทำงานแบบขนาน เป็นที่ทราบกันดีว่าหากวงจรถอดรหัสมีการทำงานแบบขนาน (Parallel processing) จะส่งผลให้ค่าความหน่วงหรือดีเลย์ของการส่งข้อมูลจะมีค่าลดต่ำไปด้วย ซึ่งค่าดีเลย์ในการถอดรหัสต่ำลงจะช่วยให้การสื่อสารเป็นไปอย่างรวดเร็วฉะนั้นรหัสช่องสัญญาณที่จะถูกนำมาใช้ต้องสามารถประมวลผลแบบขนานได้
- รองรับการทำงาน IR-HARQ (Hybrid ARQ with Incremental Redundancy) โดย IR-HARQ ถูกพัฒนามาจากเทคนิค Hybrid ARQ ที่เป็นการทำงานร่วมกันระหว่างเทคนิค ARQ error-control และการเข้ารหัสช่องสัญญาณด้วยอัตรารหัสที่ปรับได้ ซึ่งความพิเศษของเทคนิค IR-HARQ ที่แตกต่างจากเทคนิค Hybrid ARQ คือเมื่อภาครับได้รับรู้ว่าข้อมูลที่ภาคส่งมีความผิดพลาดเกิดขึ้น ภาครับจะทำการส่งการแจ้งเตือน NACK กลับไปยังภาคส่ง โดยภาคส่งจะมีการเข้ารหัสข้อมูลเดิมด้วยอัตรารหัสที่ต่ำลงเพื่อเพิ่มจำนวนบิตพาริตีให้มากขึ้น และทำการส่งเฉพาะบิตพาริตีอันใหม่มาที่ภาครับเท่านั้นเพื่อลดความซ้ำซ้อนในการส่งข้อมูลเดิมซ้ำ ฉะนั้นเงื่อนไขที่สำคัญสำหรับการออกแบบรหัสช่องสัญญาณสำหรับ 5G จะต้องรองรับการทำงาน IR-HARQ ได้
- รองรับความยาวบิตข้อมูลและอัตรารหัส สิ่งที่หลีกเลี่ยงไม่ได้เลยในการออกแบบรหัสช่องสัญญาณสำหรับเทคโนโลยีการสื่อสารยุค 5G คือการที่ต้องออกแบบกระบวนการเข้ารหัสข้อมูลให้สามารถรองรับความยาวคำรหัสต่าง ๆ อีกทั้งอัตรารหัสต้องมีความยืดหยุ่นปรับเปลี่ยนได้ตามสถานการณ์ความผิดพลาดที่เกิดขึ้นได้

2.3 การประชุมของ 3GPP เพื่อออกแบบรหัสช่องสัญญาณในมาตรฐาน 5G

องค์กร The 3rd Generation Partnership Project หรือ 3GPP คือหน่วยงานตามข้อตกลงความร่วมมือในการพัฒนารายละเอียดด้านมาตรฐานทางเทคนิค (Standards Development Organizations: SDOs) และการจัดทำรายงานต่างๆ ระหว่างองค์กรพัฒนามาตรฐานกับหน่วยงานอื่นๆ ที่เกี่ยวข้องในเครือข่ายระบบโทรศัพท์เคลื่อนที่ที่ใช้กันทั่วโลก องค์กร 3GPP เป็นการรวมกลุ่มอุตสาหกรรมโทรคมนาคมยักษ์ใหญ่ของโลก เช่น สมาคมอุตสาหกรรมวิทยุและธุรกิจญี่ปุ่น (Association of Radio Industries and Businesses: ARIB) สหพันธ์เพื่อทางออกอุตสาหกรรมโทรคมนาคมสหรัฐอเมริกา (Automatic terminal information service: ATIS) สมาคมมาตรฐานการสื่อสารจีน (China Communications Standards Association : CCSA) สถาบันมาตรฐานโทรคมนาคมยุโรป (European Telecommunication Standards Institute : ETSI) สมาคมเทคโนโลยีโทรคมนาคมเกาหลี (Telecommunications Technology Association : TTA) และคณะกรรมการเทคโนโลยีโทรคมนาคมญี่ปุ่น (Telecommunication Technology Committee :

TTC) ไม่นานมานี้องค์กร 3GPP ได้จัดตั้งการประชุมโดยมีจุดประสงค์เพื่อออกแบบมาตรฐาน “รหัสช่องสัญญาณ” ให้มีความเหมาะสมกับเทคโนโลยี 5G มากที่สุด องค์กร 3GPP ได้เริ่มศึกษารหัสช่องสัญญาณสำหรับมาตรฐาน 5G ในการประชุม RAN WG1 Meeting #84bis และเสร็จสิ้นในการประชุม RAN WG1 Meeting #91 ดังมีรายละเอียดการประชุมในแต่ละครั้งดังนี้

ตารางที่ 2.1 รายละเอียดการประชุมของ 3GPP เพื่อออกแบบรหัสช่องสัญญาณในมาตรฐาน 5G

| ชื่อการประชุม | วันและสถานที่ | สรุปข้อมูลสำคัญ |
|---------------------------------------|--|--|
| 3GPP TSG RAN Meeting #71 [9] | 7-10 March 2016 Göteborg, Sweden | วางกำหนดการและกลุ่มงานเพื่อศึกษาเทคโนโลยีสำหรับมาตรฐาน 5G |
| 3 GPP TSG RAN WG1 Meeting #84bis [10] | 11-15 April 2016 Busan, Korea | <ol style="list-style-type: none"> รหัสช่องสัญญาณที่อยู่ในตัวเลือกสำหรับมาตรฐาน 5G ได้แก่ <ul style="list-style-type: none"> - LDPC code - Polar code - Convolutional code - Turbo code - Erasure code เงื่อนไขการเลือกรหัสช่องสัญญาณ ได้แก่ <ul style="list-style-type: none"> - สมรรถนะการแก้ไขบิตผิด - ความซับซ้อนในการออกแบบฮาร์ดแวร์ - การหน่วงเวลาของวงจรเข้ารหัสและถอดรหัส - ความยืดหยุ่นในการปรับอัตราหัสและความยาวรหัส |
| 3 GPP TSG RAN WG1 Meeting #8 5 [11] | 23-27 May 2016 Nanjing, China | <p>เนื่องจากผลการจำลองสมรรถนะรหัสช่องสัญญาณของผู้ยื่นข้อเสนอมีความใกล้เคียงกัน จำเป็นต้องศึกษาคุณสมบัติอื่นๆ เพิ่มเติม โดยกำหนดให้ผู้ที่ต้องการยื่นข้อเสนอรหัสช่องสัญญาณที่จะใช้ในมาตรฐาน 5G จะต้องให้ข้อมูลวิธีการออกแบบรหัสช่องสัญญาณเพื่อให้ผู้ยื่นข้อเสนออื่น ๆ สามารถทำการจำลองอัตราบิตผิดซ้ำได้ และการแสดงอัตราบิตผิดของรหัสช่องสัญญาณที่นำเสนอจะต้องมีการปรับพารามิเตอร์ดังต่อไปนี้</p> <ul style="list-style-type: none"> - E_s/N_0 E_b/N_0 - Code rate - Information block length - iterations หรือ list size - CRC bits |

| ชื่อการประชุม | วันและสถานที่ | สรุปข้อมูลสำคัญ |
|---------------------------------------|---|---|
| 3 GPP TSG RAN WG1 Meeting #8 6 [12] | 22-26 August 2016 Gothenburg, Sweden | <ol style="list-style-type: none"> 1. ให้ผู้ยื่นข้อเสนอส่งข้อมูลผลการจำลองสมรรถนะโดยใช้รูปแบบไฟล์ excel เหมือนกัน โดยรหัสช่องสัญญาณบน data channel ในมาตรฐาน 5G ที่มีการใช้งานแบบ eMBB จะถูกเลือกในการประชุมครั้งนี้ 2. ให้ผู้ยื่นข้อเสนอทำการศึกษาความยืดหยุ่นในการปรับอัตราหัสและความยาวรหัส รวมถึงศึกษาความซับซ้อนของวงจรเข้ารหัสและถอดรหัส 3. รหัสช่องสัญญาณสำหรับ data channel จะต้องรองรับ Incremental Redundancy (IR) HARQ และ Chase Combining (CC) HARQ 4. ให้ผู้ยื่นข้อเสนอทำการศึกษารหัสช่องสัญญาณบน control channel ซึ่งมีขนาดเฟรมข้อมูลไม่เกิน 200 บิต |
| 3 GPP TSG RAN WG1 Meeting #86bis [13] | 10-14 October 2016 Lisbon, Portugal | <ol style="list-style-type: none"> 1. แสดงผลการจำลองการเปรียบเทียบสมรรถนะของรหัสช่องสัญญาณต่าง ๆ รวมถึงประเด็นอื่นที่เกี่ยวข้องกับการประชุมครั้งก่อน ซึ่งได้ข้อสังเกตของรหัสช่องสัญญาณ ดังนี้ <ul style="list-style-type: none"> <u>ประเด็นเรื่องสมรรถนะ</u> ไม่สามารถตัดสินว่า LDPC codes Polar codes และ Turbo codes ตัวเลือกไหนมีสมรรถนะดีสุดเนื่องจากข้อมูลที่มีขึ้นอยู่กับความซับซ้อนของการออกแบบวงจร <u>ประเด็นเรื่องความยืดหยุ่นของอัตราหัสและความยาวเฟรมข้อมูล LDPC codes</u> <ul style="list-style-type: none"> - หนึ่งบริษัทกังวลความซับซ้อนที่เกิดจากการประยุกต์ใช้ LDPC codes สำหรับ IR-HARQ <u>Polar codes</u> <ul style="list-style-type: none"> - หลายบริษัทกังวลเกี่ยวกับการใช้ HARQ ร่วมกับ Polar codes <u>Turbo codes</u> <ul style="list-style-type: none"> - รองรับการใช้ IR-HARQ หรือ CC-HARQ <u>ประเด็นเรื่องความซับซ้อนในการออกแบบวงจร LDPC codes</u> <ul style="list-style-type: none"> - ประสิทธิภาพจะลดลงที่อัตราหัสต่ำ - การถอดรหัสสามารถใช้ parallelization ซึ่งให้ผลการหน่วงเวลาที่ต่ำ |

| ชื่อการประชุม | วันและสถานที่ | สรุปข้อมูลสำคัญ |
|-------------------------------------|---------------------------------|--|
| | | <p>Polar codes</p> <ul style="list-style-type: none"> - การถอดรหัสแบบ list จะมีความซับซ้อนเพิ่มขึ้นมากเมื่อข้อมูลมีขนาดใหญ่ - ยังมีความกังวลเกี่ยวกับประสิทธิภาพของการใช้พลังงานเนื่องจาก Polar codes ยังไม่มีการประยุกต์ใช้งานจริงในมาตรฐานอื่น ๆ <p>Turbo codes</p> <ul style="list-style-type: none"> - การประยุกต์ใช้งานในวงจรมีความน่าสนใจในมุมมองของพื้นที่และพลังงาน แต่ throughput ต่ำ โดยเฉพาะอัตรารหัสต่ำ <p><u>ประเด็นเรื่องการหน่วงเวลา</u></p> <p>การออกแบบของ LDPC codes และ Turbo codes สามารถลดการหน่วงเวลาได้ แต่การออกแบบ Polar codes อาจลดการหน่วงเวลาได้เฉพาะเฟรมข้อมูลขนาดสั้น</p> <p>2. การตัดสินใจเบื้องต้นของรหัสช่องสัญญาณสำหรับ eMBB บน data channel จะเลือกใช้ LDPC codes เมื่อขนาดข้อมูลมีความยาวมากกว่า 1024 บิต</p> |
| 3 GPP TSG RAN WG1 Meeting #8 7 [14] | 14-18 November 2016 Reno, USA | <p>1. การออกแบบ LDPC codes</p> <ul style="list-style-type: none"> - ใช้หลักการ code extension เช่น lower-triangular extension ใน LDPC codes เพื่อรองรับ IR HARQ - ใช้ QC-LDPC codes ที่ sub-block มี circulant weight ≤ 2 - บริษัทที่สนใจออกแบบ LDPC codes ตามหลักเกณฑ์เบื้องต้นให้แสดง protomatrix ที่ออกแบบพร้อมผลอัตราบิดเบิดในการประชุมครั้งต่อไป <p>2. สรุปรหัสช่องสัญญาณสำหรับ eMMB จากผลการศึกษาของบริษัทต่าง ๆ</p> <ul style="list-style-type: none"> - เลือกใช้ LDPC codes บน data channel - เลือกใช้ polar codes บน control channel กรณีที่ขนาดเฟรมข้อมูลเล็กมากให้ใช้ repetition/block coding |
| 3 GPP TSG RAN WG1 | 16-20 January 2017 Spokane, USA | 1. Parity-check matrix ของ LDPC codes ประกอบไปด้วย sub-matrices A, B, C, D, E ดังนี้ |

| ชื่อการประชุม | วันและสถานที่ | สรุปข้อมูลสำคัญ | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------------------------------|------------------------------------|---|---|---|---|---|--|---|---|---|--|--|--|--|---|---|--|--|---|--|---|---|--|--|--|--|---|---|---|--|--|--|---|
| Meeting #AH1_NR [15] | | <table border="1" data-bbox="746 398 975 533"> <tr> <td>A</td> <td>B</td> <td>C</td> </tr> <tr> <td colspan="2">D</td> <td>E</td> </tr> </table> <p data-bbox="746 555 1412 683">โดยที่ C เป็นเมตริกซ์ศูนย์ และ E เป็นเมตริกซ์เอกลักษณ์ และเมตริกซ์ B จะมีหลักที่มี weight = 3 ตามด้วย dual diagonal structure ดังนี้</p> <table border="1" data-bbox="746 701 986 887"> <tr><td>1</td><td>1</td><td></td><td></td><td></td></tr> <tr><td></td><td>1</td><td>1</td><td></td><td></td></tr> <tr><td>1</td><td></td><td>1</td><td>1</td><td></td></tr> <tr><td></td><td></td><td></td><td>1</td><td>1</td></tr> <tr><td>1</td><td></td><td></td><td></td><td>1</td></tr> </table> <ol data-bbox="703 907 1412 1339" style="list-style-type: none"> ใช้ QC-LDPC codes ที่ sub-block มี circulant weight = 1 อัตรารหัสต่ำสุดของ LDPC codes เท่ากับ 1/5 การทำ rate matching ของ LDPC codes จะใช้หลักการ circular buffer แบบเดียวกับมาตรฐาน LTE การอธิบาย polar codes จะไม่มีกระบวนการ bit reversal ในวงจรเข้ารหัสซึ่งไม่มีผลต่อการออกแบบวงจร ตัวเลือกสำหรับการเข้ารหัส polar codes ได้แก่ <ul data-bbox="746 1220 1197 1339" style="list-style-type: none"> - CRC + polar codes - CRC + distributed CRC + polar codes - CRC + concatenated polar code | A | B | C | D | | E | 1 | 1 | | | | | 1 | 1 | | | 1 | | 1 | 1 | | | | | 1 | 1 | 1 | | | | 1 |
| A | B | C | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| D | | E | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| | | | 1 | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 1 | | | | 1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 3 GPP TSG RAN WG1 Meeting #8 8 [16] | 13-17 February 2017 Athens, Greece | <ol data-bbox="703 1355 1412 1568" style="list-style-type: none"> Parity-check matrix ของ LDPC codes ซึ่งประกอบไปด้วย sub-matrices A, B, C, D, E มีรายละเอียดดังนี้ <ul data-bbox="746 1444 1197 1568" style="list-style-type: none"> - A เกี่ยวข้องกับ systematic bits - B เกี่ยวข้องกับ parity bits - กรณีหลักแรกหรือหลักสุดท้ายมี weight = 1 <div data-bbox="794 1601 1273 1713"> </div> <p data-bbox="794 1742 1013 1780">กรณีไม่มี weight = 1</p> <div data-bbox="794 1814 989 1926"> </div> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

| ชื่อการประชุม | วันและสถานที่ | สรุปข้อมูลสำคัญ |
|---------------------------------------|--------------------------------|---|
| | | 2. LDPC codes จะมีจำนวน base graph น้อยกว่าหรือเท่ากับ 2 3. ให้ศึกษาการเข้ารหัส polar codes โดยยึดรูปแบบดังนี้ <ul style="list-style-type: none"> - J bits CRC + J' bits CRC + polar codes - J bits CRC + J' bits distributed CRC + polar codes - J bits CRC + J' PC bits + polar codes - J bits CRC + J' Hash sequence + polar codes - $(J + J')$ bits CRC + polar codes 4. ความยาวสูงสุดของ polar code เท่ากับ <ul style="list-style-type: none"> - $N = 512$ สำหรับ downlink - $N = 1024$ สำหรับ uplink |
| 3 GPP TSG RAN WG1 Meeting #88bis [17] | 3-7 April 2017 Spokane, USA | 1. LDPC codes มีขนาด CRC ≤ 24 บิต 2. การประชุมครั้งนี้จะตัดสิน base graph โดยระหว่างนี้ให้ผู้ยื่นข้อเสนอส่ง base graph โดยมี shift value matrices ≤ 8 3. การตัดสินเลือก base graph จะมีตัวเลือกดังนี้ <ul style="list-style-type: none"> - ถ้าจำนวน base graph = 1 จะต้องมีอัตรารหัสเท่ากับ $1/5 \leq R \leq 8/9$ - ถ้าจำนวน base graph = 2 จะมีเงื่อนไข คือ base graph#1 จะมีอัตรารหัส $1/3 \leq R \leq 8/9$ base graph#2 จะมีอัตรารหัส $1/5 \leq R \leq 2/3$ - พิจารณา BLER ที่ต่ำสุดโดยใช้การถอดรหัส SPA แบบ floating point flooding และใช้จำนวน iteration เท่ากับ 50 โดยสามารถหยุดกระบวนการถอดรหัสก่อนครบจำนวนรอบได้ 4. การออกแบบ polar codes จะต้องทำให้กระบวนการถอดรหัสสามารถหยุดได้เมื่อตรวจพบบิตผิดแม้ว่ากระบวนการถอดรหัสยังดำเนินการไม่เสร็จสิ้น 5. จำนวนบิต CRC สำหรับ polar codes อาจแตกต่างกันระหว่าง downlink กับ uplink และขนาดของเฟรมข้อมูล |
| 3 GPP TSG RAN WG1 Meeting #8 9 [18] | 15-19 May 2017 Hangzhou, China | 1. LDPC codes จะมีจำนวน base graph เท่ากับ 2 โดยมีตัวเลือกได้แก่ <ul style="list-style-type: none"> - สำหรับ base graph#1 จะมีขนาด 46×68 ซึ่งมีตัวเลือก 7 ตัว ได้แก่ A-G - สำหรับ base graph#2 จะมีขนาด 42×52 - การประชุมครั้งนี้จะทำการตัดสินเลือก base graph 2. การเลือก bit selection ของ polar codes มีตัวเลือกได้แก่ |

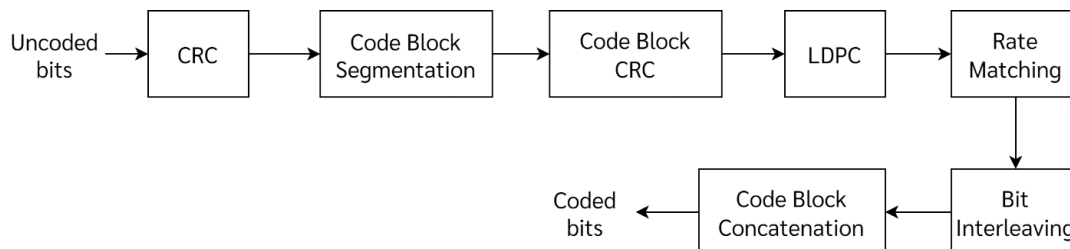
| ชื่อการประชุม | วันและสถานที่ | สรุปข้อมูลสำคัญ |
|--|--|---|
| | | <ul style="list-style-type: none"> - ออกแบบ fixed sequence ของแต่ละขนาด mother codes - แต่ละขนาด mother code ใช้ fixed sequence ที่หาได้จาก reference mother codes 3. Rate matching ของ polar codes จะใช้วิธีต่อไปนี้ <ul style="list-style-type: none"> - ใช้ puncturing สำหรับอัตราหัสต่ำ - ใช้ shortening สำหรับอัตราหัสสูง 4. ใช้ polar codes บน Broadcast channel โดยมีความยาวสูงสุดเท่ากับ 512 บิต |
| 3 GPP TSG RAN WG1 Meeting #AH2_NR [19] | 27-30 June 2017 Qingdao, China | <ol style="list-style-type: none"> 1. โครงสร้างการเข้ารหัสของ LDPC codes สำหรับ eMBB จะใช้ base graph#1 และ base graph#2 ในไฟล์ excel ของเอกสาร R1-1711982 2. โครงสร้างการเข้ารหัสของ Polar codes สำหรับ eMBB ได้แก่ <ul style="list-style-type: none"> - สำหรับ downlink ใช้ CRC + Interleaver + Polar codes - สำหรับ uplink CRC + 3 parity bits + Polar codes 3. แต่ละขนาด mother code ของ Polar codes ใช้ fixed sequence แบบเดียวกัน โดยมีวิธีออกแบบ ได้แก่ <ul style="list-style-type: none"> - Polarization Weight (PW) - Optimized- combined-and-nested (CN) - Mutual information density evolution (MI-DE) 4. การประชุมครั้งนี้จะตัดสินใจเลือกวิธีการออกแบบ fixed sequence โดยพิจารณาจากค่าBLER เมื่อกำหนดให้ list <= 16 |
| 3 GPP TSG RAN WG1 Meeting #90 [20] | 21-25 August 2017 Prague, Czech Republic | <ol style="list-style-type: none"> 1. LDPC codes ใช้ CRC เท่ากับ 24 บิต 2. การเลือกใช้ base graph ของ LDPC codes จะใช้เกณฑ์ต่อไปนี้ <ul style="list-style-type: none"> - ใช้ base graph#1 เมื่อขนาดข้อมูลมากกว่า 3840 บิต หรือ อัตราหัสเริ่มต้นมากกว่า 0.67 - ใช้ base graph#2 เมื่อขนาดข้อมูลน้อยกว่า 3840 บิต และ อัตราหัสเริ่มต้นน้อยกว่า 0.67 3. การออกแบบ fixed sequence ของ polar codes จะใช้วิธี Polarization Weight (PW) |
| 3 GPP TSG RAN WG1 Meeting #AH3_NR [21] | 18-21 September 2017 Nagoya, Japan | <ol style="list-style-type: none"> 1. Channel interleaver ของ LDPC codes มีลักษณะแบบ row-column interleaver ซึ่งมีจำนวนแถวเท่ากับ modulation order โดยมีกระบวนการเขียนในแนวแถวและอ่านในแนวหลัก |

| ชื่อการประชุม | วันและสถานที่ | สรุปข้อมูลสำคัญ |
|---------------------------------------|---|--|
| | | 2. Channel interleaver ของ Polar codes มีลักษณะแบบ triangular interleaver |
| 3 GPP TSG RAN WG1 Meeting #90bis [22] | 9-13 October 2017 Prague, Czech Republic | 1. เงื่อนไขเพิ่มเติมสำหรับการเลือก base graph ของ LDPC codes ได้แก่ - ใช้ base graph#1 เมื่ออัตรารหัสมากกว่า 0.95 - ใช้ base graph#2 เมื่อขนาดข้อมูลน้อยกว่า 308 bits 2. Polar codes สำหรับ downlink จะไม่ใช่ channel interleaver 3. Polar codes สำหรับ downlink จะใช้ CRC จำนวน 24 บิต |
| 3 GPP TSG RAN WG1 Meeting #91 [23] | 27-1 December 2017 Reno, USA | 1. เงื่อนไขเพิ่มเติมสำหรับการเลือก base graph ได้แก่ กำหนดใช้ base graph#1 และ base graph#2 เมื่ออัตรารหัสมากกว่า 0.95 2. Polar code สำหรับ uplink จะใช้จำนวนบิต CRC ดังนี้ - ใช้ CRC = 6 บิต เมื่อขนาดข้อมูลเท่ากับ $12 \leq K \leq 19$ - ใช้ CRC = 11 บิต เมื่อขนาดข้อมูลเท่ากับ $K > 19$ |
| 3GPP TSG RAN meeting #78 [24] | 18-21 December 2017 Lisbon, Portugal | จัดทำ technical specification ของรหัสช่องสัญญาณในมาตรฐาน Release 15 (5G Phase 1) |

ผลสรุปการประชุมทั้งสิ้น 14 ครั้ง องค์กร 3GPP ได้มีข้อสรุปว่าจะมีการนำรหัสแอลดีพีซี (LDPC codes: 1962) มาใช้สำหรับข้อมูลในส่วน User data เนื่องจากรหัส LDPC มีประสิทธิภาพของปริมาณการส่งผ่านข้อมูล (throughput) ต่อพื้นที่ และมีปริมาณการส่งผ่านข้อมูลสูงสุดที่สามารถใช้งานได้สูงกว่าในรหัส Turbo อีกทั้งรหัส LDPC ยังสามารถลดความซับซ้อน และความหน่วงเวลา (latency) จากการประมวลผลแบบขนานในการถอดรหัสได้ โดยเฉพาะอย่างยิ่งเมื่อใช้งานกับรหัสที่มีอัตรารหัสสูง ในส่วนการส่งข้อมูล Control data จะมีการใช้รหัสปอลาร์ (Polar codes: 2009) เนื่องจากรหัส Polar สามารถใช้งานร่วมการตรวจสอบความผิดพลาดแบบ CRC และมีการถอดรหัสแบบ Successive Cancellation List (SCL) ซึ่งจะมีประสิทธิภาพในการถอดรหัสดีกว่ารหัส Turbo โดยเฉพาะเมื่อใช้สำหรับการส่งข้อมูลที่มีความยาวไม่เกิน 250 บิต ซึ่งเป็นขนาดข้อมูลที่เพียงพอสำหรับการส่งข้อมูล Control data ภายหลังจากได้ข้อสรุปจากการประชุมแล้วนั้นจึงเกิดมาตรฐานที่ชื่อว่า 3GPP NR ขึ้นมาและจะนำส่งไปพิจารณาที่ IMT-2020 ต่อไป

2.4 รายละเอียดการเข้ารหัสช่องสัญญาณชนิด LDPC codes มาตรฐาน 5G

2.4.1 ภาพรวมการเข้ารหัสแอลดีพีซีมาตรฐาน 5G



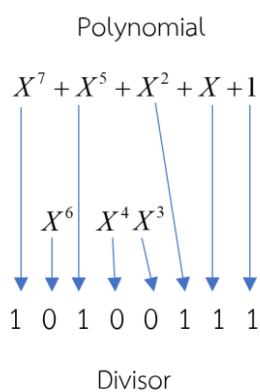
รูปที่ 2.1 แผนภาพการเข้ารหัสแอลดีพีซีมาตรฐาน 5G

โครงการนี้จะพัฒนาการเข้ารหัสแอลดีพีซีมาตรฐาน 5G ตามเอกสาร 3GPP 38.212 [1] รูปที่ 2.1 แสดงแผนภาพบล็อกของกระบวนการเข้ารหัสช่องสัญญาณสำหรับส่วนช่องสัญญาณข้อมูล โดยการทำงานจะเริ่มต้นจากการนำข้อมูลที่ต้องการส่งไปเข้ารหัสด้วยรหัสตรวจจับความผิดพลาด ที่เรียกว่า CRC (Cyclic Redundancy Check) ก่อนจะนำผลลัพธ์ที่ได้ไปผ่านกระบวนการแบ่งส่วนข้อมูล (Segmentation) เพื่อแบ่งข้อมูลให้มีขนาดของข้อมูลให้เป็นไปตามมาตรฐานของการสื่อสาร 5G โดยข้อมูลจะถูกแบ่งออกเป็นบล็อกย่อย ๆ ที่มีขนาดเท่ากัน หลังจากนั้นแต่ละบล็อกย่อยจะถูกนำไปเข้ารหัส CRC ซ้ำอีกครั้ง แล้วนำผลลัพธ์ที่ได้ไปเข้ารหัส LDPC โดยการแทรกบิตพาริตีไปกับข้อมูล เพื่อให้สามารถแก้ไขความผิดพลาดของข้อมูลที่ภาครับได้อย่างถูกต้อง ผลลัพธ์ที่ได้จากการเข้ารหัส LDPC จะถูกเรียกว่าคำรหัส (Codeword) จากนั้นคำรหัสจะถูกส่งต่อไปยังวงจรปรับอัตรารหัส (Rate Matching) เพื่อปรับความยาวของคำรหัสให้เหมาะสมกับการส่ง จากนั้นวงจรแทรกสลับบิต (Bit Interleaving) จะถูกใช้งานเพื่อแทรกสลับบิตข้อมูลในคำรหัสเพื่อป้องกันความผิดพลาดแบบติดกัน (Burst Error) และสุดท้ายจะมีการต่อเรียงบล็อกคำรหัสเพื่อส่งต่อไปมอดูเลตและส่งข้อมูลออกไปตามลำดับ

2.4.2 การเข้ารหัส CRC

รหัส CRC (Cyclic Redundancy Check) เป็นรหัสที่มีความสามารถในการตรวจสอบความผิดพลาดของข้อมูล โดยมีลักษณะคล้าย Hash แต่มีความซับซ้อนน้อยกว่าและมีขนาดของผลลัพธ์การคำนวณน้อยกว่า (ขนาดไม่เกิน 64 บิต ในขณะที่ Hash มีขนาดมากกว่า 128 บิต) รหัส CRC นิยมใช้ในการรับส่งข้อมูลที่มีการเข้าจังหวะ (Synchronous) และการสื่อสารข้อมูลที่ต้องการความรวดเร็วในการตรวจสอบความผิดพลาดของข้อมูล

หลักการของ CRC [2] จะใช้พหุนาม (Polynomial) มากำหนดเป็นค่าตัวหาร (Divisor) ดังตัวอย่างรูปที่ 2.2 ซึ่งมีข้อกำหนดว่าบิตซ้ายสุดและบิตขวาสุดต้องมีค่าเป็น 1 เสมอ อีกทั้งพหุนามจะต้องมีจำนวนบิตน้อยกว่าจำนวนบิตของข้อมูล ในการหาค่า CRC สามารถทำได้โดยอาศัยตัวดำเนินการ Exclusive-OR หากบิตตรงกันผลที่ได้จะเท่ากับ 0 ในขณะที่บิตต่างกัน ผลที่ได้ก็จะเท่ากับ 1 ซึ่งผลลัพธ์ของ Exclusive-OR แสดงในตารางที่ 2.2

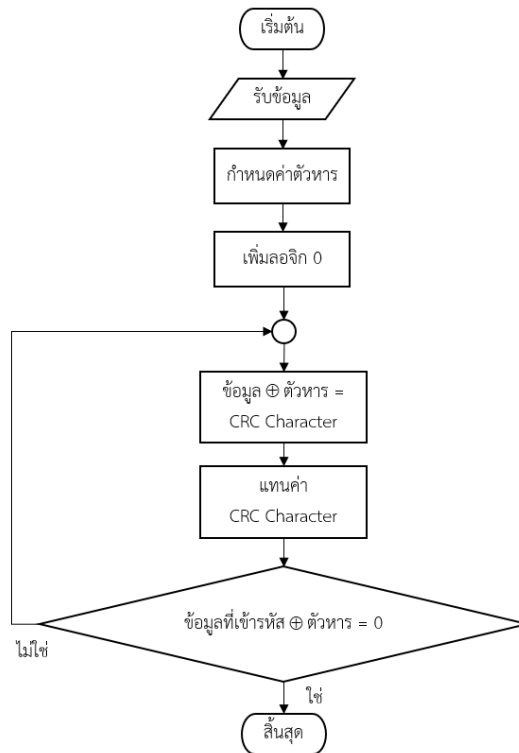


รูปที่ 2.2 พหุนามและการแทนค่าตัวหาร

ตารางที่ 2.2 ผลการปฏิบัติด้วยโอเปอเรเตอร์ Exclusive-OR

| First Input | Second Input | XOR Output |
|-------------|--------------|------------|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

ขั้นตอนการทำงานของ CRC



รูปที่ 2.3 ขั้นตอนการทำงานของ CRC

รูปที่ 2.3 แสดงขั้นตอนการทำงานของ การเข้ารหัส CRC [3] โดยเริ่มต้นจากการรับข้อมูลที่ต้องการเข้ารหัส และกำหนดค่าตัวหาร (Divisor) ในรูปของฟังก์ชันพหุนาม (Polynomial) ที่หารด้วย $x+1$ ลงตัว จากนั้นทำการเพิ่มลอจิก 0 ในส่วนท้ายของข้อมูลตามจำนวนบิตของการเข้ารหัส CRC

หลังจากที่มีการเติมบิต 0 ลงไปที่ข้อมูลแล้ว ข้อมูลจะถูกนำมาทำการ Exclusive-OR กับค่าตัวหาร เพื่อหาค่าเศษที่ได้จากการคำนวณซึ่งจะเรียกว่าค่า CRC Character จากนั้นนำ CRC Character แทนค่าลอจิก 0 ในส่วนท้ายของข้อมูลที่ใส่เข้าไปในตอนแรก โดยผลลัพธ์ที่ได้คือข้อมูลที่ผ่านการเข้ารหัส CRC หมายความว่าในการตรวจสอบความถูกต้องในการทำ CRC นั้น เมื่อนำค่าข้อมูลที่เพิ่มค่า CRC Character แล้วมาทำการ Exclusive-OR กับค่าตัวหาร ถ้าผลลัพธ์ที่ได้เป็น 0 แสดงว่าข้อมูลที่ผ่านการเข้ารหัส CRC มีความถูกต้อง

การเข้ารหัสแบบ CRC ในมาตรฐาน 5G [4] สำหรับแบบ Uplink และ Downlink ใช้ การเข้ารหัส CRC ขนาด 24, 16, 11 และ 6 บิต โดยมีการกำหนดโพลิโนเมียลตัวหารดังนี้

- สำหรับการเข้ารหัส CRC 24 บิต

$$g_{\text{CRC24A}}(X) = X^{24} + X^{23} + X^{18} + X^{17} + X^{14} + X^{11} + X^{10} + X^7 + X^6 + X^5 + X^4 + X^3 + X + 1$$

$$g_{\text{CRC24B}}(X) = X^{24} + X^{23} + X^6 + X^5 + X + 1$$

$$g_{\text{CRC24C}}(X) = X^{24} + X^{23} + X^{21} + X^{20} + X^{17} + X^{15} + X^{13} + X^{12} + X^8 + X^4 + X^2 + X + 1$$

- สำหรับการเข้ารหัส CRC 16 บิต

$$g_{\text{CRC16}}(X) = X^{16} + X^{12} + X^5 + 1$$

- สำหรับการเข้ารหัส CRC 11 บิต

$$g_{\text{CRC11}}(X) = X^{11} + X^{10} + X^9 + X^5 + 1$$

- สำหรับการเข้ารหัส CRC 6 บิต

$$g_{\text{CRC6}}(X) = X^6 + X^5 + 1 \text{ ในการเข้ารหัส}$$

โดยขนาดของการเข้ารหัส CRC และโพลีโนเมียลตัวหารดังกล่าวได้มีการพิสูจน์แล้วว่าเพียงพอสำหรับการใช้ตรวจสอบความผิดพลาดข้อมูลสำหรับช่องสัญญาณข้อมูล

อย่างไรก็ตามสำหรับกระบวนการเข้ารหัสช่องสัญญาณสำหรับช่องสัญญาณข้อมูล ในมาตรฐาน 5G ประกอบด้วยการเข้ารหัส CRC 2 ส่วนได้แก่ การเข้ารหัส CRC ในส่วนของบล็อกการขนส่ง (Transport Block) และการเข้ารหัส CRC ในส่วนของบล็อกรหัส ดังนี้

การเข้ารหัส CRC ในส่วนของบล็อกการขนส่ง เป็นการเข้ารหัส CRC ข้อมูลทั้งหมดที่ต้องการส่งไปยังภาครับ โดยมีการแบ่งออกเป็น 2 กรณีได้แก่ กรณีข้อมูลมีความยาวมากกว่า 3824 บิต กำหนดให้ใช้การเข้ารหัส CRC ขนาด 24 บิต มีโพลีโนเมียลเป็น $g_{\text{CRC24A}}(X)$ หากเป็นกรณีอื่น กำหนดให้ใช้การเข้ารหัส CRC ขนาด 16 บิตมีโพลีโนเมียลเป็น $g_{\text{CRC16}}(X)$

การเข้ารหัส CRC ในส่วนของบล็อกรหัสเป็นการเข้ารหัส CRC ข้อมูลแต่ละบล็อกย่อย ๆ หลังผ่านกระบวนการ Segmentation เนื่องจากเมื่อทำการแบ่งข้อมูลเป็นบล็อกย่อย ข้อมูลบางบล็อกเป็นเพียงข้อมูลบางส่วนของข้อมูลเริ่มต้นที่ต้องการส่งไปยังภาครับ ดังนั้นจึงมีการเข้ารหัส CRC ข้อมูลแต่ละบล็อกย่อย ๆ เพื่อใช้ในการตรวจสอบความผิดพลาดของข้อมูลแต่ละบล็อกย่อยในภาครับ โดยมีการกำหนดให้ใช้การเข้ารหัส CRC ขนาด 24 บิต มีโพลีโนเมียลเป็น $g_{\text{CRC24B}}(X)$

2.4.3 การแบ่งส่วนข้อมูล (Segmentation)

การแบ่งส่วนข้อมูล (Segmentation) [4] เป็นกระบวนการแบ่งบล็อกข้อมูลให้เป็นบล็อกย่อย เมื่อบล็อกข้อมูลมีขนาดมากเกินไปขอบเขตในกระบวนการเข้ารหัส LDPC มาตรฐาน 5G ที่กำหนด บล็อกข้อมูลจะถูกแบ่งเป็นบล็อกรหัส (Code Block) แต่ละบล็อกรหัสจะถูกส่งไปเข้ารหัสด้วยรหัส CRC ซึ่งจำนวนบิตข้อมูลสูงสุดต่อบล็อก (K_{cb}) ที่มาตรฐาน 5G รองรับจะขึ้นอยู่กับกราฟฐาน (Base Graph) ที่เลือกใช้

กรณีที่ใช้กราฟฐาน 1 ขนาดบิตสูงสุดต่อบล็อก $K_{cb} = 8448$ บิต

กรณีที่ใช้กราฟฐาน 2 ขนาดบิตสูงสุดต่อบล็อก $K_{cb} = 3840$ บิต

กระบวนการ Segmentation เริ่มต้นจากการพิจารณาจำนวนบิตอินพุตที่เข้ามาโดยบิตอินพุตที่เข้ามา (B) ซึ่งเป็นบิตข้อมูลที่ผ่านมาการเข้ารหัส CRC แล้ว 1 ครั้ง หาก B มีขนาดมากกว่า K_{cb} จะทำการแบ่งบิตข้อมูลออกเป็นบล็อกรหัส (C) และทำการส่งแต่ละบล็อกไปเข้ารหัส CRC จำนวน 24 บิตอีกครั้ง แต่หาก B น้อยกว่าหรือเท่ากับ K_{cb} จะไม่มีการเข้ารหัส CRC อีกครั้งและจะมีบล็อกข้อมูลเพียงหนึ่งบล็อก และบิตทั้งหมดที่ต้องส่ง (B) จะเท่ากับ $n = \max(\min(n_1, n_2, n_{\max}), n_{\min})$ การกำหนดค่าพารามิเตอร์สำหรับทั้ง 2 กรณี แสดงดังตารางที่ 2.3

ตารางที่ 2.3 การกำหนดพารามิเตอร์ในการแบ่งส่วนข้อมูล

| กรณี $B > K_{cb}$ | กรณี $B \leq K_{cb}$ |
|-------------------|------------------------------------|
| $L = 0$ | $L = 24$ |
| $C = 1$ | $C = \lceil B / K_{cb} - L \rceil$ |
| $B' = B$ | $B' = B + C \cdot L$ |

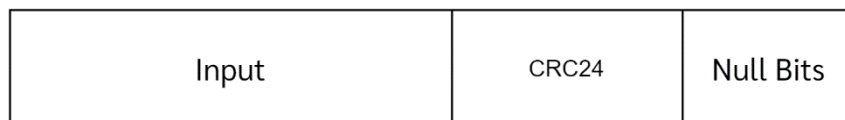
หลังจากผ่านกระบวนการ Segmentation บิตข้อมูลจะถูกแบ่งออกเป็นบล็อกรหัสขนาด K บิต ซึ่งแต่ละบล็อกประกอบด้วย $C_{r_0}, C_{r_1}, \dots, C_{r_{(K_r-1)}}$ เมื่อ $0 \leq r < C$ โดยที่ r คือลำดับบล็อกรหัสและ K_r คือ จำนวนบิตทั้งหมดในบล็อกที่ r โดยในแต่ละบล็อกจะถูกแบ่งให้มีจำนวนบิตข้อมูลเท่ากันทุกบล็อก หรือ $K_r = K$ โดยจำนวนบิตในแต่ละบล็อกจะมีขนาด K' บิต ซึ่งสามารถหาได้จาก

$$K' = B' / C \tag{1}$$

มาตรฐาน 5G มีการกำหนดจำนวนบิตในแต่ละบล็อกให้อยู่ภายใต้เงื่อนไข $K_b \cdot Z_c \geq K'$ จากนั้นแทนค่า Z_c ที่ได้จากเงื่อนไขดังกล่าวเพื่อหาค่า K โดยมีข้อกำหนดดังนี้ หากเลือกใช้กราฟฐาน 1 จะใช้ $K = 22Z_c$ และหากเลือกใช้กราฟฐาน 2 จะใช้ $K = 10Z_c$ จึงส่งผลให้ต้องมีการปรับขนาดของบิตข้อมูลที่แบ่งในตอนต้น จากเดิมขนาด K' บิต เป็นขนาดบิตข้อมูลตามมาตรฐานกำหนดขนาด K บิต โดย Z_c จะถูกกำหนดมาจากการเลือกค่า Z ที่ต่ำที่สุดในตารางที่ 2.9 และอยู่ภายใต้เงื่อนไขดังกล่าว ส่วน K_b จะขึ้นกับกราฟฐานที่เลือกใช้ โดยมีเงื่อนไขการพิจารณา ดังนี้

- กรณีใช้กราฟฐาน 1 ใช้ $K_b = 22$ บิต
- กรณีใช้กราฟฐาน 2 และ $B > 640$ บิต ใช้ $K_b = 10$ บิต
- กรณีใช้กราฟฐาน 2 และ $B > 560$ บิต ใช้ $K_b = 9$ บิต
- กรณีใช้กราฟฐาน 2 และ $B > 192$ บิต ใช้ $K_b = 8$ บิต
- กรณีอื่น ๆ ใช้ $K_b = 6$ บิต

หลังจากทำการแบ่งข้อมูลเป็นบล็อกย่อย ๆ แต่ละบล็อกจะมีองค์ประกอบ 2 ส่วน ได้แก่ บิตข้อมูล และบิตว่าง (Null Bits) บิตข้อมูลอินพุตจะถูกใส่เข้าไปในตำแหน่งที่ 0 ถึง $K' - L - 1$ ของบล็อก ในกรณีที่ $C < 1$ หรือมีจำนวนบล็อกย่อยมากกว่า 1 บล็อก ข้อมูลภายในบล็อกจะถูกส่งไปเข้ารหัสตรวจจับความผิดพลาด หรือ CRC โดยทำการคำนวณหา CRC พาริตี 24 บิต จากโพลีโนเมียล $g_{\text{CRC24B}}(D)$ และถูกใส่ในตำแหน่งที่ $K' - L$ ถึง $K' - 1$ จากนั้นจะมีการเติมบิตว่างตั้งแต่บิตที่ H ถึง $K - 1$ เพื่อให้ครบจำนวนบิตข้อมูลในแต่ละบล็อก (K) ดังรูปที่ 2.4



รูปที่ 2.4 โครงสร้างแต่ละบล็อก ในกรณีมีจำนวนบล็อกมากกว่า 1 บล็อก

2.4.4 รหัสแอลดีพีซี (LDPC)

รหัสพาริตีเชิงความหนาแน่นต่ำหรือรหัสแอลดีพีซี (Low-Density Parity-Check: LDPC) [5] เป็นรหัสบล็อกเชิงเส้น (Linear Block Code) ที่ถูกนำเสนอโดย R. Gallager และถูกนำมาศึกษาและพัฒนาต่อยอดทำให้เป็นที่สนใจอีกครั้งโดย D. Mackay และ R. Neal ซึ่งนักวิจัยทั้งสองได้มีการพิสูจน์ว่ารหัส LDPC มีสมรรถนะเข้าใกล้ความจุช่องสัญญาณหรือลิมิตของแชนนอน (Shannon's Limit) ภายใต้การถอดรหัสแบบวนซ้ำด้วยการแพร่กระจายความเชื่อมั่น (Belief Propagation: BP) นอกจากนี้รหัส LDPC ยังมีความซับซ้อนน้อยและสามารถนำมาใช้ในกับการเข้ารหัสข้อมูลที่มีขนาด

ใหญ่ได้อย่างมีประสิทธิภาพทำให้รหัส LDPC ที่ได้รับความนิยมและถูกนำมาใช้ในระบบสื่อสารดิจิทัลหลายรูปแบบ รหัส LDPC ถูกนิยามด้วยเมทริกซ์พาริตีเชิงความหนาแน่นต่ำ กล่าวคือในเมทริกซ์ดังกล่าวมีจำนวนสมาชิกที่มีค่าเป็น 1 น้อยมากเมื่อเทียบกับขนาดของเมทริกซ์พาริตีเชิง ด้วยลักษณะดังกล่าวจึงถูกเรียกว่ารหัสพาริตีเชิงความหนาแน่นต่ำ

2.4.4.1 รหัสบล็อกเชิงเส้น

รหัสบล็อกเชิงเส้น (Linear Block Code) [6] สามารถเขียนแทนด้วยสัญลักษณ์ (N, K) ซึ่ง N คือความยาวคำรหัสและ K คือความยาวบิตข้อมูล ดังนั้นจึงสรุปได้ว่าภายใต้คำรหัสจะประกอบไปด้วย 2 ส่วน ส่วนแรกคือบิตข้อมูลที่ต้องการส่งไปยังภาครับ (Message Bit) จะมีอยู่จำนวน K บิต และสามารถเขียนอยู่ในรูป $[u_0, u_1, u_2, \dots, u_{K-1}]$ ส่วนที่สองคือบิตพาริตี (Parity Bit) มีจำนวน $N - K$ บิต และสามารถเขียนอยู่ในรูป $p = [u_0, u_1, u_2, \dots, u_{N-K}]$ โดยคำรหัสทั้งหมดที่ประกอบด้วยส่วนทั้งสองสามารถเขียนในรูป $c = [u_0, u_1, u_2, \dots, u_{K-1}, p_0, p_1, p_2, \dots, p_{N-K}]$ ซึ่งสามารถเขียนทุกส่วนรวมในเวกเตอร์ดังรูปที่ 2.5

$$[c_0, c_1, c_2, \dots, c_{N-1}] = \boxed{\begin{array}{|l|l|} \hline [u_0, u_1, u_2, \dots, u_{K-1}] & [p_1, p_2, \dots, p_{N-K}] \\ \hline \end{array}}$$

รูปที่ 2.5 โครงสร้างของรหัสบล็อกเชิงเส้นแบบ (N, K)

รหัสบล็อกเชิงเส้นยังมีคุณสมบัติคือ เมื่อนำคำรหัส 2 คำรหัสใด ๆ มาบวกกันแบบมอดุโล 2 ผลลัพธ์ที่ได้จะเป็นอีกคำรหัสหนึ่งเสมอ นอกจากนี้การถอดรหัสบล็อกเชิงเส้นจะถอดทีละบล็อกซึ่งขนาดของบล็อกจะขึ้นอยู่กับแต่ละงานประยุกต์ที่ใช้ในระบบสื่อสาร อัตราส่วนของบิตข้อมูลกับจำนวนบิตทั้งหมดในคำรหัสนั้นเรียกว่า “อัตรารหัส” (Code Rate: R) นิยามได้ดังนี้

$$R = \frac{K}{N} \tag{2}$$

โดยค่าอัตรารหัสที่เลือกใช้ขึ้นอยู่กับปริมาณสัญญาณรบกวน (Noise) นั่นคือในการสื่อสารที่มีสัญญาณรบกวนมากจะเลือกใช้อัตรารหัสที่มีค่าต่ำ หากการสื่อสารที่มีสัญญาณรบกวนน้อยจะเลือกใช้อัตรารหัสที่มีค่าสูง

2.4.4.2 มุมมองของเมทริกซ์

รหัส LDPC จะถูกกำหนดโดยเมทริกซ์ 2 ตัว คือเมทริกซ์ตัวกำเนิด (Generator Matrix: \mathbf{G}) และเมทริกซ์พาริตีเช็ค (Parity Check Matrix: \mathbf{H}) เมทริกซ์ \mathbf{G} มีหน้าที่ในการสร้างคำรหัสจากบิตข้อมูลที่เข้ามาและเมทริกซ์ \mathbf{H} มีหน้าที่ในการถอดรหัสจากบิตรหัสที่ได้รับจากช่องสัญญาณ

1.4.2.1 เมทริกซ์กำเนิด

เมทริกซ์ \mathbf{G} มีไว้สำหรับเข้ารหัสบิตข้อมูล ซึ่งกระบวนการเข้ารหัสทำได้โดยการนำบิตข้อมูล \mathbf{u} มาคูณเข้ากับเมทริกซ์ \mathbf{G} โดยเมทริกซ์ \mathbf{G} แบ่งออกเป็น 2 ส่วน จากสมการ (3) จะเห็นว่าส่วนแรกคือเมทริกซ์เอกลักษณ์ (Identity Matrix) และส่วนที่ 2 คือเมทริกซ์พาริตี

$$\mathbf{G}_{K \times N} = [\mathbf{I}_{K \times K} \mid \mathbf{P}_{K \times (N-K)}] \quad (3)$$

ซึ่งคำรหัสที่ได้นั้นก็จะมีขนาด $1 \times N$ ในส่วนของการเข้ารหัสนั้นก็จะเป็นดังสมการที่ (4) ดังนี้

$$\mathbf{c} = \mathbf{u}\mathbf{G} = [u_0, u_1, u_2, \dots, u_{K-1}, p_0, p_1, \dots, p_{N-K-1}] = [c_0, c_1, c_2, \dots, c_{N-1}] \quad (4)$$

สมการที่ (4) แสดงให้เห็นว่าบิตข้อมูลเป็นส่วนหนึ่งของคำรหัสที่มีพาริตีเพิ่มเข้ามาซึ่งพาริตีเหล่านี้มีหน้าที่ในการแก้ไขความผิดพลาดของคำรหัสเมื่อถูกส่งผ่านช่องสัญญาณที่มีการรบกวนจากสัญญาณรบกวนต่าง ๆ

2.4.2.2 เมทริกซ์ตรวจสอบพาริตี

เมทริกซ์ตรวจสอบพาริตี \mathbf{H} มีผลอย่างมากต่อประสิทธิภาพของรหัส LDPC นอกจากนี้เมทริกซ์ \mathbf{H} ยังมีความสัมพันธ์กับเมทริกซ์ \mathbf{G} ตามสมการที่ (5)

$$\mathbf{H}\mathbf{G}^T = \mathbf{0} \quad (5)$$

จากสมการที่ (5) จะเห็นว่าทั้งเมทริกซ์ \mathbf{G} และ \mathbf{H} เมื่อนำมาคูณกันแล้วจะได้เวกเตอร์ 0 นอกจากนี้หากสมการที่ (3) คือเมทริกซ์ \mathbf{G} ที่อยู่ในรูปแบบที่เป็นระบบ (Systematic) แล้วเมทริกซ์ \mathbf{H} ที่อยู่ในรูปแบบที่เป็นระบบเช่นกัน รูปแบบที่เป็นระบบคือมีส่วนของเมทริกซ์พาริตีจะอยู่ส่วนหน้าและต่อกับเมทริกซ์เอกลักษณ์ที่อยู่ส่วนหลังตามสมการที่ (6)

$$\mathbf{H}_{(N-K) \times N} = [\mathbf{P}^T \mid \mathbf{I}_{(N-K) \times (N-K)}] \quad (6)$$

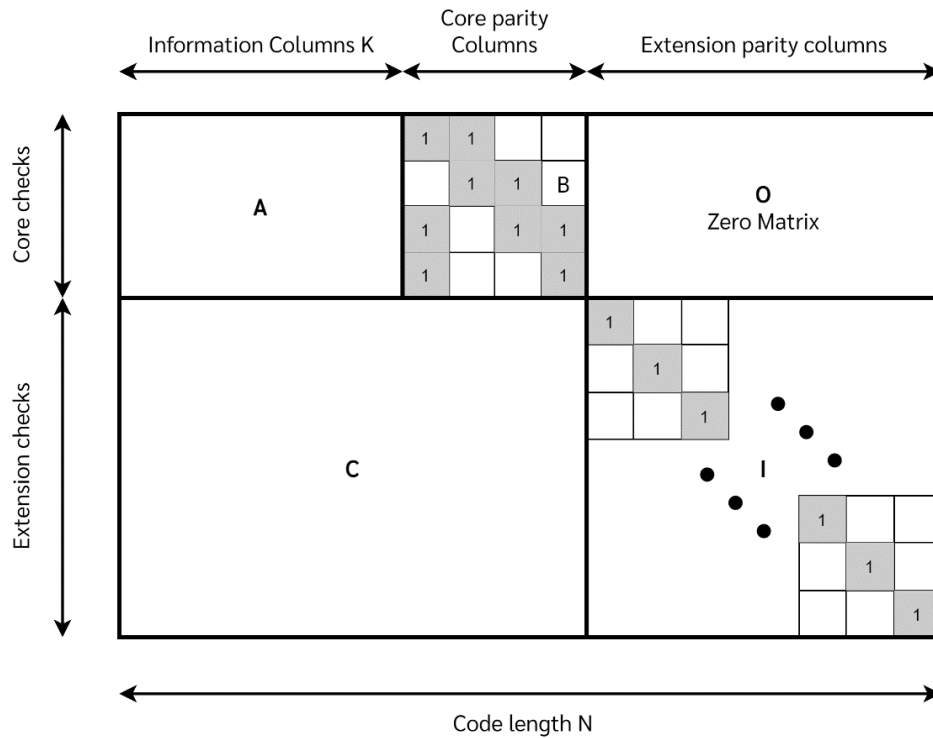
เมทริกซ์ \mathbf{H} ยังมีความสัมพันธ์กับค่ารหัสซึ่งเป็นไปตามสมการที่ (7)

$$\mathbf{Hc}^T = \mathbf{HG}^T \mathbf{u}^T = \mathbf{0} \quad (7)$$

ความสัมพันธ์ดังกล่าวสามารถใช้ในการตรวจสอบความถูกต้องของค่ารหัสที่ได้จากกระบวนการเข้ารหัส LDPC ที่ภาคส่ง หรือกล่าวอีกนัยหนึ่งคือหากนำเมทริกซ์ \mathbf{H} มาคูณกับทรานสโพสเมทริกซ์ (Transpose Matrix) ของค่ารหัสแล้วมีค่าเท่ากับ 0 นั่นคือค่ารหัสที่ได้จากกระบวนการเข้ารหัส LDPC มีความถูกต้อง นอกจากนี้ความสัมพันธ์ดังกล่าวยังสามารถนำไปใช้เป็นต้นแบบในการถอดรหัสด้วยวิธีต่าง ๆ ที่ภาครับได้เช่นกัน

2.4.4.3 การประยุกต์ใช้งานรหัสแอลดีพีซีสำหรับช่องสัญญาณข้อมูลในมาตรฐาน release 15 (5G Phase 1)

รหัส LDPC มีบทบาทสำคัญในการสื่อสารยุค 5G โดยได้รับการยอมรับให้เป็นรูปแบบการเข้ารหัสช่องสัญญาณบนช่องสัญญาณข้อมูลสำหรับการสื่อสารไร้สาย ในการประชุมมาตรฐาน 3GPP ได้ข้อสรุปว่าโครงสร้างเมทริกซ์ฐานของรหัส LDPC ที่ใช้สำหรับการสื่อสารไร้สาย 5G มีลักษณะดังรูปที่ 2.6 ประกอบด้วยเมทริกซ์ย่อย 5 เมทริกซ์คือ \mathbf{A} , \mathbf{B} , \mathbf{O} , \mathbf{C} และ \mathbf{I} โดย เมทริกซ์ย่อย \mathbf{A} เกี่ยวข้องกับบิตข้อมูล เมทริกซ์ย่อย \mathbf{B} เกี่ยวข้องกับบิตพาริตี (Parity Bits) เมทริกซ์ย่อย \mathbf{O} เป็นเมทริกซ์ศูนย์ เมทริกซ์ย่อย \mathbf{C} สอดคล้องกับแถว SPC (Single Parity check) และเมทริกซ์ย่อย \mathbf{I} เป็นเมทริกซ์เอกลักษณ์ โดยเมทริกซ์ย่อย \mathbf{A} และเมทริกซ์ย่อย \mathbf{B} รวมกันเรียกว่าเคอร์เนล (Kernel) และเมทริกซ์ย่อยส่วนอื่น ๆ (\mathbf{O} , \mathbf{C} และ \mathbf{I}) เรียกว่าส่วนขยาย



รูปที่ 2.6 โครงสร้างพื้นฐานเมทริกซ์ฐานของรหัส LDPC ตามมาตรฐาน 5G

2.4.4.3.1 กระบวนการเข้ารหัส LDPC

กระบวนการการเข้ารหัส LDPC สำหรับช่องสัญญาณข้อมูลในมาตรฐาน release 15 เริ่มต้นจากการพิจารณาค่า MCS (Modulation and Coding Scheme) เพื่อให้ทราบอัตรารหัสที่ต้องใช้ในการเข้ารหัส แล้วทำการเลือกกราฟฐานของรหัส LDPC ซึ่งกราฟฐาน 1 ใช้สำหรับบล็อกข้อมูลขนาดใหญ่ ($44 \leq K \leq 8448$) และอัตรารหัสสูงในช่วง ($1/3 \leq R \leq 8/9$) ในขณะที่กราฟฐาน 2 ใช้สำหรับบล็อกข้อมูลขนาดเล็ก ($20 \leq K \leq 3840$) และอัตรารหัสที่ต่ำในช่วง ($1/5 \leq R \leq 10/13$) โดยมีเงื่อนไขในการพิจารณาเลือกดังนี้

- 1) กรณีข้อมูลเท่ากับ $K \leq 308$ บิต ใช้กราฟฐาน 2
- 2) กรณีอัตรารหัสเท่ากับ $R < 1/4$ ใช้กราฟฐาน 2
- 3) กรณีอัตรารหัสเท่ากับ $R < 2/3$ และขนาดข้อมูล $K < 3840$ บิต ใช้กราฟฐาน 2
- 4) กรณีอื่น ๆ ใช้ base graph 1

2.4.4.3.2 รายละเอียดกราฟฐาน ของรหัส LDPC

โครงสร้างของรหัส LDPC ในมาตรฐาน release 15 [4] หรือ มาตรฐาน 5G phase 1 แสดงได้โดยใช้เมทริกซ์พาริตีเช็คราฟฐาน 1 ดังรูปที่ 2.7 และกราฟฐาน 2 ดังรูปที่ 2.8 ซึ่งมีรายละเอียดของเมทริกซ์ฐานของทั้งกราฟฐาน 1 และกราฟฐาน 2 แสดงดังตารางที่ 2.4 โดยช่องสี่ขาวคือเมทริกซ์ศูนย์ขนาด $Z \times Z$ และช่องสีอื่นๆ คือเมทริกซ์ที่มีการสลับเปลี่ยนแบบเวียนหมุน (Circular Permutation Matrix) ขนาด $Z \times Z$ การเลือกค่า Z จากตารางที่ 2.5 สามารถหาได้ตามสมการที่ (8) และ (9)

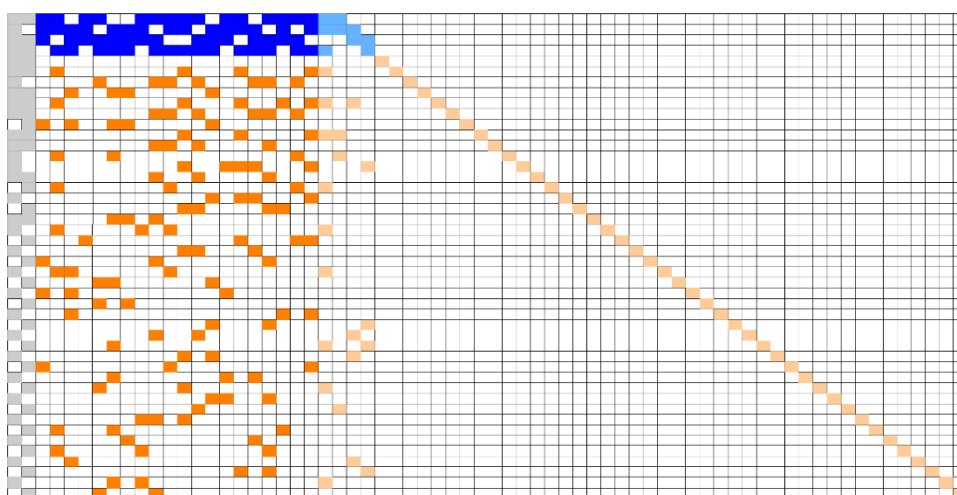
$$Z = K / 22 \text{ สำหรับกราฟฐาน 1} \quad (8)$$

$$Z = K / 10 \text{ สำหรับกราฟฐาน 1} \quad (9)$$

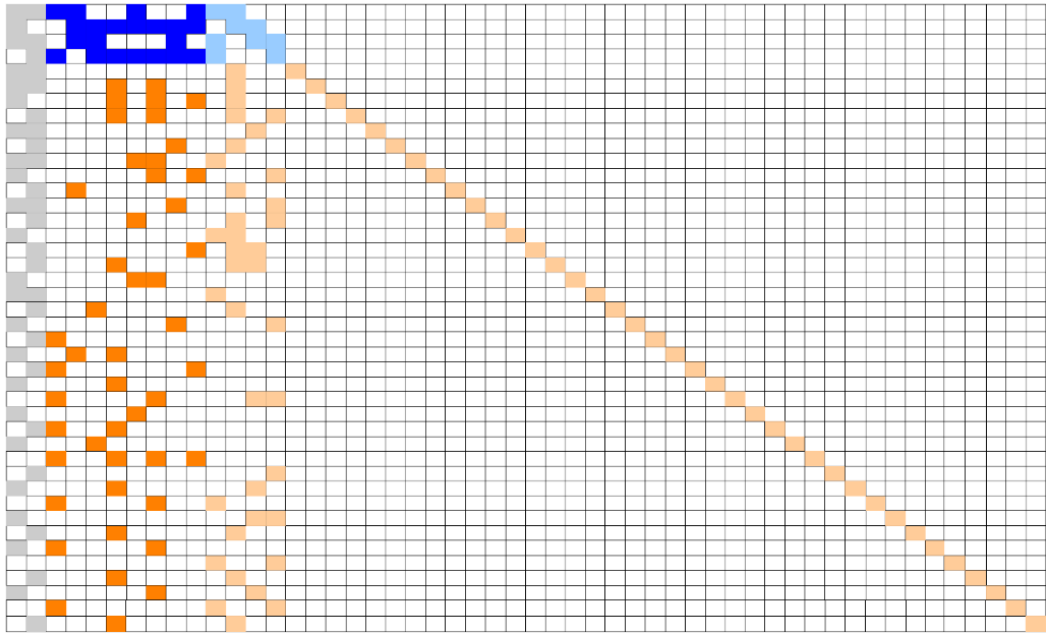
เมื่อทราบค่า Z แล้วจึงสามารถสร้างเมทริกซ์ที่มีการสลับเปลี่ยนแบบเวียนหมุนได้จากการนำเมทริกซ์เอกลักษณ์มาดำเนินการหมุนแบบวนกลับ โดยจำนวนครั้งของการหมุนสามารถหาได้ตามสมการที่ (10)

$$P_{i,j} = V_{i,j} \oplus Z \quad (10)$$

โดยค่า $V_{i,j}$ หาได้จากตารางที่ 2.5 และ 2.6 โดยขึ้นอยู่กับค่าดัชนีเซต (Set index) และกราฟฐานของ LDPC



รูปที่ 2.7 กราฟฐาน 1



รูปที่ 2.8 กราฟฐาน 2

ตารางที่ 2.4 รายละเอียดเมทริกซ์ฐาน

| รายละเอียดของเมทริกซ์ฐาน | กราฟฐาน 1 | กราฟฐาน 2 |
|----------------------------------|-----------|-----------|
| อัตรารหัส (R) ต่ำสุด | 1/3 | 1/5 |
| ขนาดของเมทริกซ์ | 44x68 | 42x52 |
| หลักที่เกี่ยวข้องกับบิตข้อมูล | 1-22 | 1-10 |
| หลักที่เกี่ยวข้องกับบิตพาริตี | 23-68 | 11-52 |
| หลักที่เกี่ยวข้องกับบิต puncture | 1-2 | 1-2 |

ตารางที่ 2.5 รายละเอียดค่า $k = 0, \dots, E_r - 1$

| ดัชนีเซต (Set index) E_r | เซตของขนาดในการขยายกราฟฐาน (g_k) |
|----------------------------|--------------------------------------|
| 0 | {2, 4, 8, 16, 32, 64, 128, 256} |
| 1 | {3, 6, 16, 24, 48, 96, 192, 384} |
| 2 | {5, 10, 20, 40, 80, 160, 320} |
| 3 | {7, 14, 28, 56, 112, 224} |
| 4 | {9, 18, 36, 72, 144, 288} |
| 5 | {11, 22, 44, 88, 176, 352} |
| 6 | {13, 26, 52, 104, 208} |
| 7 | {15, 30, 60, 120, 240} |

ตารางที่ 2.6 รายละเอียดค่า $V_{i,j}$ สำหรับกราฟฐาน 1 ของ LDPC

| H_{BG} | | $V_{i,j}$ | | | | | | | H_{BG} | | $V_{i,j}$ | | | | | | | | |
|-----------|--------------|--------------------|-----|-----|-----|-----|-----|-----|-----------|--------------|--------------------|-----|-----|-----|-----|-----|-----|-----|-----|
| Row index | Column index | Set index i_{LS} | | | | | | | Row index | Column index | Set index i_{LS} | | | | | | | | |
| i | j | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | i | j | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | 0 | 250 | 307 | 73 | 223 | 211 | 294 | 0 | 135 | 15 | 1 | 96 | 2 | 290 | 120 | 0 | 348 | 6 | 138 |
| | 1 | 69 | 19 | 15 | 16 | 198 | 118 | 0 | 227 | | 10 | 65 | 210 | 60 | 131 | 183 | 15 | 81 | 220 |
| | 2 | 226 | 50 | 103 | 94 | 188 | 167 | 0 | 126 | | 13 | 63 | 318 | 130 | 209 | 108 | 81 | 182 | 173 |
| | 3 | 159 | 369 | 49 | 91 | 186 | 330 | 0 | 134 | | 18 | 75 | 55 | 184 | 209 | 68 | 176 | 53 | 142 |
| | 5 | 100 | 181 | 240 | 74 | 219 | 207 | 0 | 84 | | 25 | 179 | 269 | 51 | 81 | 64 | 113 | 46 | 49 |
| | 6 | 10 | 216 | 39 | 10 | 4 | 165 | 0 | 83 | | 37 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 9 | 59 | 317 | 15 | 0 | 29 | 243 | 0 | 53 | | 1 | 64 | 13 | 69 | 154 | 270 | 190 | 88 | 78 |
| | 10 | 229 | 288 | 162 | 205 | 144 | 250 | 0 | 225 | | 3 | 49 | 338 | 140 | 164 | 13 | 293 | 198 | 152 |
| | 11 | 110 | 109 | 215 | 216 | 116 | 1 | 0 | 205 | | 11 | 49 | 57 | 45 | 43 | 99 | 332 | 160 | 84 |
| | 12 | 191 | 17 | 164 | 21 | 216 | 339 | 0 | 128 | | 20 | 51 | 289 | 115 | 189 | 54 | 331 | 122 | 5 |
| | 13 | 9 | 357 | 133 | 215 | 115 | 201 | 0 | 75 | | 22 | 154 | 57 | 300 | 101 | 0 | 114 | 182 | 205 |
| | 15 | 195 | 215 | 298 | 14 | 233 | 53 | 0 | 135 | | 38 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 16 | 23 | 106 | 110 | 70 | 144 | 347 | 0 | 217 | | 0 | 7 | 260 | 257 | 56 | 153 | 110 | 91 | 183 |
| | 18 | 190 | 242 | 113 | 141 | 95 | 304 | 0 | 220 | | 14 | 164 | 303 | 147 | 110 | 137 | 228 | 184 | 112 |
| | 19 | 35 | 180 | 16 | 198 | 216 | 167 | 0 | 90 | | 16 | 59 | 81 | 128 | 200 | 0 | 247 | 30 | 106 |
| | 20 | 239 | 330 | 189 | 104 | 73 | 47 | 0 | 105 | | 17 | 1 | 358 | 51 | 63 | 0 | 116 | 3 | 219 |
| | 21 | 31 | 346 | 32 | 81 | 261 | 188 | 0 | 137 | | 21 | 144 | 375 | 228 | 4 | 162 | 190 | 155 | 129 |
| | 22 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | | 39 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 1 | 42 | 130 | 260 | 199 | 161 | 47 | 1 | 183 |
| | 0 | 2 | 76 | 303 | 141 | 179 | 77 | 22 | 96 | | 12 | 233 | 163 | 294 | 110 | 151 | 286 | 41 | 215 |
| | 2 | 239 | 76 | 294 | 45 | 162 | 225 | 11 | 236 | | 13 | 8 | 280 | 291 | 200 | 0 | 246 | 167 | 180 |
| | 3 | 117 | 73 | 27 | 151 | 223 | 96 | 124 | 136 | | 18 | 155 | 132 | 141 | 143 | 241 | 181 | 68 | 143 |
| | 4 | 124 | 288 | 261 | 46 | 256 | 338 | 0 | 221 | | 19 | 147 | 4 | 295 | 186 | 144 | 73 | 148 | 14 |
| | 5 | 71 | 144 | 161 | 119 | 160 | 268 | 10 | 128 | | 40 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 7 | 222 | 331 | 133 | 157 | 76 | 112 | 0 | 92 | | 0 | 60 | 145 | 64 | 8 | 0 | 87 | 12 | 179 |
| 8 | 104 | 331 | 4 | 133 | 202 | 302 | 0 | 172 | 1 | 73 | 213 | 181 | 6 | 0 | 110 | 6 | 108 | | |
| 9 | 173 | 178 | 80 | 87 | 117 | 50 | 2 | 56 | 7 | 72 | 344 | 101 | 103 | 118 | 147 | 166 | 159 | | |
| 11 | 220 | 295 | 129 | 206 | 109 | 167 | 16 | 11 | 8 | 127 | 242 | 270 | 198 | 144 | 258 | 184 | 138 | | |
| 12 | 102 | 342 | 300 | 93 | 15 | 253 | 60 | 189 | 10 | 224 | 197 | 41 | 8 | 0 | 204 | 191 | 196 | | |
| 14 | 109 | 217 | 76 | 79 | 72 | 334 | 0 | 95 | 41 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 15 | 132 | 99 | 266 | 9 | 152 | 242 | 6 | 85 | 0 | 151 | 187 | 301 | 105 | 265 | 89 | 6 | 77 | | |
| 16 | 142 | 354 | 72 | 118 | 158 | 257 | 30 | 153 | 3 | 186 | 206 | 162 | 210 | 81 | 65 | 12 | 187 | | |
| 17 | 155 | 114 | 83 | 194 | 147 | 133 | 0 | 87 | 9 | 217 | 264 | 40 | 121 | 90 | 155 | 15 | 203 | | |
| 19 | 255 | 331 | 260 | 31 | 156 | 9 | 168 | 163 | 11 | 47 | 341 | 130 | 214 | 144 | 244 | 5 | 167 | | |
| 21 | 28 | 112 | 301 | 187 | 119 | 302 | 31 | 216 | 22 | 160 | 59 | 10 | 183 | 228 | 30 | 30 | 130 | | |
| 22 | 0 | 0 | 0 | 0 | 0 | 0 | 105 | 0 | 42 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 249 | 205 | 79 | 192 | 64 | 162 | 6 | 197 | | |
| 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 5 | 121 | 102 | 175 | 131 | 46 | 264 | 86 | 122 | | |
| 0 | 106 | 205 | 68 | 207 | 258 | 226 | 132 | 189 | 16 | 109 | 328 | 132 | 220 | 266 | 346 | 96 | 215 | | |
| 1 | 111 | 250 | 7 | 203 | 167 | 35 | 37 | 4 | 20 | 131 | 213 | 283 | 50 | 9 | 143 | 42 | 65 | | |
| 2 | 185 | 328 | 80 | 31 | 220 | 213 | 21 | 225 | 21 | 171 | 97 | 103 | 106 | 18 | 109 | 199 | 216 | | |
| 4 | 63 | 332 | 280 | 176 | 133 | 302 | 180 | 151 | 43 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 5 | 117 | 256 | 38 | 180 | 243 | 111 | 4 | 236 | 0 | 64 | 30 | 177 | 53 | 72 | 280 | 44 | 25 | | |
| 6 | 93 | 161 | 227 | 186 | 202 | 265 | 149 | 117 | 12 | 142 | 11 | 20 | 0 | 189 | 157 | 58 | 47 | | |
| 7 | 229 | 267 | 202 | 95 | 218 | 128 | 48 | 179 | 13 | 188 | 233 | 55 | 3 | 72 | 236 | 130 | 126 | | |
| 8 | 177 | 160 | 200 | 153 | 63 | 237 | 38 | 92 | 17 | 158 | 22 | 316 | 148 | 257 | 113 | 131 | 178 | | |
| 9 | 95 | 63 | 71 | 177 | 0 | 294 | 122 | 24 | 44 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 10 | 39 | 129 | 106 | 70 | 3 | 127 | 195 | 68 | 1 | 156 | 24 | 249 | 88 | 180 | 18 | 45 | 185 | | |
| 13 | 142 | 200 | 295 | 77 | 74 | 110 | 155 | 6 | 2 | 147 | 89 | 50 | 203 | 0 | 6 | 18 | 127 | | |
| 14 | 225 | 88 | 283 | 214 | 229 | 286 | 28 | 101 | 10 | 170 | 61 | 133 | 168 | 0 | 181 | 132 | 117 | | |
| 15 | 225 | 53 | 301 | 77 | 0 | 125 | 85 | 33 | 18 | 152 | 27 | 105 | 122 | 165 | 304 | 100 | 199 | | |
| 17 | 245 | 131 | 184 | 198 | 216 | 131 | 47 | 96 | 45 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 18 | 205 | 240 | 246 | 117 | 269 | 163 | 179 | 125 | 0 | 112 | 298 | 289 | 49 | 236 | 38 | 9 | 32 | | |
| 19 | 251 | 205 | 230 | 223 | 200 | 210 | 42 | 67 | 3 | 86 | 158 | 280 | 157 | 199 | 170 | 125 | 178 | | |
| 20 | 117 | 13 | 276 | 90 | 234 | 7 | 66 | 230 | 4 | 236 | 235 | 110 | 64 | 0 | 249 | 191 | 2 | | |
| 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11 | 116 | 339 | 187 | 193 | 266 | 288 | 28 | 156 | | |
| 25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 22 | 222 | 234 | 281 | 124 | 0 | 194 | 6 | 58 | | |
| 0 | 121 | 276 | 220 | 201 | 187 | 97 | 4 | 128 | 46 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 1 | 89 | 87 | 208 | 18 | 145 | 94 | 6 | 23 | 1 | 23 | 72 | 172 | 1 | 205 | 279 | 4 | 27 | | |
| 3 | 84 | 0 | 30 | 165 | 166 | 49 | 33 | 162 | 6 | 136 | 17 | 295 | 166 | 0 | 255 | 74 | 141 | | |
| 4 | 20 | 275 | 197 | 5 | 108 | 279 | 113 | 220 | 7 | 116 | 383 | 96 | 65 | 0 | 111 | 16 | 11 | | |
| 6 | 150 | 199 | 61 | 45 | 82 | 139 | 49 | 43 | 14 | 182 | 312 | 46 | 81 | 183 | 54 | 28 | 181 | | |
| 7 | 131 | 153 | 175 | 142 | 132 | 166 | 21 | 186 | 47 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 8 | 243 | 56 | 79 | 16 | 197 | 91 | 6 | 96 | 0 | 195 | 71 | 270 | 107 | 0 | 325 | 21 | 163 | | |
| 10 | 136 | 132 | 281 | 34 | 41 | 106 | 151 | 1 | 2 | 243 | 81 | 110 | 176 | 0 | 326 | 142 | 131 | | |
| 11 | 86 | 305 | 303 | 155 | 162 | 246 | 83 | 216 | 4 | 215 | 76 | 318 | 212 | 0 | 226 | 192 | 169 | | |
| 12 | 246 | 231 | 253 | 213 | 57 | 345 | 154 | 22 | 15 | 61 | 136 | 67 | 127 | 277 | 99 | 197 | 98 | | |
| 13 | 219 | 341 | 164 | 147 | 36 | 269 | 87 | 24 | 48 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 14 | 211 | 212 | 53 | 69 | 115 | 185 | 5 | 167 | 1 | 25 | 194 | 210 | 208 | 45 | 91 | 98 | 165 | | |
| 16 | 240 | 304 | 44 | 96 | 242 | 249 | 92 | 200 | 6 | 104 | 194 | 29 | 141 | 36 | 326 | 140 | 232 | | |
| 17 | 76 | 300 | 28 | 74 | 165 | 215 | 173 | 32 | 8 | 194 | 101 | 304 | 174 | 72 | 268 | 22 | 9 | | |
| 18 | 244 | 271 | 77 | 99 | 0 | 143 | 120 | 235 | 49 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 20 | 144 | 39 | 319 | 30 | 113 | 121 | 2 | 172 | 0 | 128 | 222 | 11 | 146 | 275 | 102 | 4 | 32 | | |
| 21 | 12 | 357 | 68 | 158 | 108 | 121 | 142 | 219 | 4 | 165 | 19 | 293 | 153 | 0 | 1 | 1 | 43 | | |
| 22 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 19 | 181 | 244 | 50 | 217 | 155 | 40 | 40 | 200 | | |
| 25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 21 | 63 | 274 | 234 | 114 | 62 | 167 | 93 | 205 | | |
| 0 | 157 | 332 | 233 | 170 | 246 | 42 | 24 | 64 | 50 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 1 | 102 | 181 | 205 | 10 | 235 | 256 | 204 | 211 | 1 | 86 | 252 | 27 | 150 | 0 | 273 | 92 | 232 | | |
| 26 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 14 | 236 | 5 | 308 | 11 | 180 | 104 | 136 | 32 | | |
| 0 | 205 | 195 | 83 | 164 | 261 | 219 | 185 | 2 | 18 | 84 | 147 | 117 | 53 | 0 | 243 | 106 | 118 | | |
| 1 | 236 | 14 | 292 | 59 | 181 | 130 | 100 | 171 | 25 | 6 | 78 | 29 | 68 | 42 | 107 | 6 | 103 | | |
| 3 | 194 | 115 | 50 | 86 | 72 | 251 | 24 | 47 | 51 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 12 | 231 | 166 | 318 | 80 | 283 | 322 | 65 | 143 | 0 | 216 | 159 | 91 | 34 | 0 | 171 | 2 | 170 | | |
| 16 | 28 | 241 | 201 | 182 | 254 | 295 | 207 | 210 | 10 | 73 | 229 | 23 | 130 | 90 | 16 | 88 | 199 | | |

| H_{BG} | | $V_{i,j}$ | | | | | | | | H_{BG} | | $V_{i,j}$ | | | | | | | | |
|---------------|------------------|--------------------|-----|-----|-----|-----|-----|-----|-----|---------------|------------------|--------------------|-----|-----|-----|-----|-----|-----|-----|---|
| Row index i | Column index j | Set index i_{LS} | | | | | | | | Row index i | Column index j | Set index i_{LS} | | | | | | | | |
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| 6 | 21 | 123 | 51 | 267 | 130 | 79 | 258 | 161 | 180 | 31 | 13 | 120 | 260 | 105 | 210 | 252 | 95 | 112 | 26 | |
| | 22 | 115 | 157 | 279 | 153 | 144 | 283 | 72 | 180 | | 24 | 9 | 90 | 135 | 123 | 173 | 212 | 20 | 105 | |
| | 27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 52 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 6 | 0 | 183 | 278 | 289 | 158 | 80 | 294 | 6 | 199 | 32 | 1 | 95 | 100 | 222 | 175 | 144 | 101 | 4 | 73 | |
| | 6 | 22 | 257 | 21 | 119 | 144 | 73 | 27 | 22 | | 7 | 177 | 215 | 308 | 49 | 144 | 297 | 49 | 149 | |
| | 10 | 28 | 1 | 293 | 113 | 169 | 330 | 163 | 23 | | 22 | 172 | 258 | 66 | 177 | 166 | 279 | 125 | 175 | |
| | 11 | 67 | 351 | 13 | 21 | 90 | 99 | 50 | 100 | | 25 | 61 | 256 | 162 | 128 | 19 | 222 | 194 | 108 | |
| | 13 | 244 | 92 | 232 | 63 | 59 | 172 | 48 | 92 | | 53 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 17 | 11 | 253 | 302 | 51 | 177 | 150 | 24 | 207 | | 0 | 221 | 102 | 210 | 192 | 0 | 351 | 6 | 103 | |
| | 18 | 157 | 18 | 138 | 136 | 151 | 284 | 38 | 52 | | 12 | 112 | 201 | 22 | 209 | 211 | 265 | 126 | 110 | |
| | 20 | 211 | 225 | 235 | 116 | 108 | 305 | 91 | 13 | | 14 | 199 | 175 | 271 | 58 | 36 | 338 | 63 | 151 | |
| 7 | 28 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 33 | 24 | 121 | 287 | 217 | 30 | 162 | 83 | 20 | 211 | | |
| | 0 | 220 | 9 | 12 | 17 | 169 | 3 | 145 | | 77 | 54 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 1 | 44 | 62 | 88 | 76 | 189 | 103 | 88 | | 146 | 1 | 2 | 323 | 170 | 114 | 0 | 56 | 10 | 199 | |
| | 4 | 159 | 316 | 207 | 104 | 154 | 224 | 112 | | 209 | 2 | 187 | 8 | 20 | 49 | 0 | 304 | 30 | 132 | |
| | 7 | 31 | 333 | 50 | 100 | 184 | 297 | 153 | | 32 | 11 | 41 | 361 | 140 | 161 | 76 | 141 | 6 | 172 | |
| | 8 | 167 | 290 | 25 | 150 | 104 | 215 | 159 | | 166 | 21 | 211 | 105 | 33 | 137 | 18 | 101 | 92 | 65 | |
| | 14 | 104 | 114 | 76 | 158 | 164 | 39 | 76 | | 18 | 55 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | 127 | 230 | 187 | 82 | 197 | 60 | 4 | 161 | |
| 8 | 0 | 112 | 307 | 295 | 33 | 54 | 348 | 172 | 181 | 34 | 7 | 167 | 148 | 296 | 186 | 0 | 320 | 153 | 237 | |
| | 1 | 4 | 179 | 133 | 95 | 0 | 75 | 2 | 105 | | 15 | 164 | 202 | 5 | 68 | 108 | 112 | 197 | 142 | |
| | 3 | 7 | 165 | 130 | 4 | 252 | 22 | 131 | 141 | | 17 | 159 | 312 | 44 | 150 | 0 | 54 | 155 | 180 | |
| | 12 | 211 | 18 | 231 | 217 | 41 | 312 | 141 | 223 | | 56 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 16 | 102 | 39 | 296 | 204 | 98 | 224 | 96 | 177 | | 1 | 161 | 320 | 207 | 192 | 199 | 100 | 4 | 231 | |
| | 19 | 164 | 224 | 110 | 39 | 46 | 17 | 99 | 145 | | 6 | 197 | 335 | 158 | 173 | 278 | 210 | 45 | 174 | |
| | 21 | 109 | 368 | 269 | 58 | 15 | 59 | 101 | 199 | | 12 | 207 | 2 | 55 | 26 | 0 | 195 | 168 | 145 | |
| | 22 | 241 | 67 | 245 | 44 | 230 | 314 | 35 | 153 | | 22 | 103 | 266 | 285 | 187 | 205 | 268 | 185 | 100 | |
| 9 | 24 | 90 | 170 | 154 | 201 | 54 | 244 | 116 | 38 | 35 | 57 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | 30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 37 | 210 | 259 | 222 | 216 | 135 | 6 | 11 | |
| | 0 | 103 | 366 | 189 | 9 | 162 | 156 | 6 | 169 | | 14 | 105 | 313 | 179 | 157 | 16 | 15 | 200 | 207 | |
| | 1 | 182 | 232 | 244 | 37 | 159 | 88 | 10 | 12 | | 15 | 51 | 297 | 178 | 0 | 0 | 35 | 177 | 42 | |
| | 10 | 109 | 321 | 36 | 213 | 93 | 293 | 145 | 206 | | 18 | 120 | 21 | 160 | 6 | 0 | 188 | 43 | 100 | |
| | 11 | 21 | 133 | 286 | 105 | 134 | 111 | 53 | 221 | | 58 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 13 | 142 | 57 | 151 | 89 | 45 | 92 | 201 | 17 | | 1 | 198 | 269 | 298 | 81 | 72 | 319 | 82 | 59 | |
| | 17 | 14 | 303 | 267 | 185 | 132 | 152 | 4 | 212 | | 13 | 220 | 82 | 15 | 195 | 144 | 236 | 2 | 204 | |
| 10 | 18 | 61 | 63 | 135 | 109 | 76 | 23 | 164 | 92 | 36 | 23 | 122 | 115 | 115 | 138 | 0 | 85 | 135 | 161 | |
| | 20 | 216 | 82 | 209 | 218 | 209 | 337 | 173 | 205 | | 59 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 31 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 167 | 185 | 151 | 123 | 190 | 164 | 91 | 121 | |
| | 1 | 98 | 101 | 14 | 82 | 178 | 175 | 126 | 116 | | 9 | 151 | 177 | 179 | 90 | 0 | 196 | 64 | 90 | |
| | 2 | 149 | 339 | 80 | 165 | 1 | 253 | 77 | 151 | | 10 | 157 | 289 | 64 | 73 | 0 | 209 | 198 | 26 | |
| | 4 | 167 | 274 | 211 | 174 | 28 | 27 | 156 | 70 | | 12 | 163 | 214 | 181 | 10 | 0 | 246 | 100 | 140 | |
| | 7 | 160 | 111 | 75 | 19 | 267 | 231 | 16 | 230 | | 60 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 8 | 49 | 383 | 161 | 194 | 234 | 49 | 12 | 115 | | 1 | 173 | 258 | 102 | 12 | 153 | 236 | 4 | 115 | |
| 11 | 14 | 58 | 354 | 311 | 103 | 201 | 267 | 70 | 84 | 37 | 3 | 139 | 93 | 77 | 77 | 0 | 264 | 28 | 188 | |
| | 32 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 7 | 149 | 346 | 192 | 49 | 165 | 37 | 109 | 168 | |
| | 0 | 77 | 48 | 16 | 52 | 55 | 25 | 184 | 45 | | 19 | 0 | 297 | 208 | 114 | 117 | 272 | 188 | 52 | |
| | 1 | 41 | 102 | 147 | 11 | 23 | 322 | 194 | 115 | | 61 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 12 | 83 | 8 | 290 | 2 | 274 | 200 | 123 | 134 | | 0 | 157 | 175 | 32 | 67 | 216 | 304 | 10 | 4 | |
| | 16 | 182 | 47 | 289 | 35 | 181 | 351 | 16 | 1 | | 8 | 137 | 37 | 80 | 45 | 144 | 237 | 84 | 103 | |
| | 21 | 78 | 188 | 177 | 32 | 273 | 166 | 104 | 152 | | 17 | 149 | 312 | 197 | 96 | 2 | 135 | 12 | 30 | |
| | 22 | 252 | 334 | 43 | 84 | 39 | 338 | 109 | 165 | | 62 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 12 | 23 | 22 | 115 | 280 | 201 | 26 | 192 | 124 | 107 | 38 | 1 | 167 | 52 | 154 | 23 | 0 | 123 | 2 | 53 | |
| | 33 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 3 | 173 | 314 | 47 | 215 | 0 | 77 | 75 | 189 | |
| | 0 | 160 | 77 | 229 | 142 | 225 | 123 | 6 | 186 | | 9 | 139 | 139 | 124 | 60 | 0 | 25 | 142 | 215 | |
| | 1 | 42 | 186 | 235 | 175 | 162 | 217 | 20 | 215 | | 18 | 151 | 288 | 207 | 167 | 183 | 272 | 128 | 24 | |
| | 10 | 21 | 174 | 169 | 136 | 244 | 142 | 203 | 124 | | 63 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 11 | 32 | 232 | 48 | 3 | 151 | 110 | 153 | 180 | | 0 | 149 | 113 | 226 | 114 | 27 | 288 | 163 | 222 | |
| | 13 | 234 | 50 | 105 | 28 | 238 | 176 | 104 | 98 | | 4 | 157 | 14 | 65 | 91 | 0 | 83 | 10 | 170 | |
| | 18 | 7 | 74 | 52 | 182 | 243 | 76 | 207 | 80 | | 24 | 137 | 218 | 126 | 78 | 35 | 17 | 162 | 71 | |
| 13 | 34 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 39 | 64 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| | 0 | 177 | 313 | 39 | 81 | 231 | 311 | 52 | | 220 | 1 | 151 | 113 | 228 | 206 | 52 | 210 | 1 | 22 | |
| | 3 | 248 | 177 | 302 | 56 | 0 | 251 | 147 | | 185 | 16 | 163 | 132 | 69 | 22 | 243 | 3 | 163 | 127 | |
| | 7 | 151 | 266 | 303 | 72 | 216 | 265 | 1 | | 154 | 18 | 173 | 114 | 176 | 134 | 0 | 53 | 99 | 49 | |
| | 20 | 185 | 115 | 160 | 217 | 47 | 94 | 16 | | 178 | 25 | 139 | 168 | 102 | 161 | 270 | 167 | 98 | 125 | |
| | 23 | 62 | 370 | 37 | 78 | 36 | 81 | 46 | | 150 | 65 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 35 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 0 | 0 | 139 | 80 | 234 | 84 | 18 | 79 | 4 | 191 | |
| | 0 | 206 | 142 | 78 | 14 | 0 | 22 | 1 | | 124 | 7 | 157 | 78 | 227 | 4 | 0 | 244 | 6 | 211 | |
| 14 | 12 | 55 | 248 | 299 | 175 | 186 | 322 | 202 | 144 | 40 | 9 | 163 | 163 | 259 | 9 | 0 | 293 | 142 | 187 | |
| | 15 | 206 | 137 | 54 | 211 | 253 | 277 | 118 | 182 | | 22 | 173 | 274 | 260 | 12 | 57 | 272 | 3 | 148 | |
| | 16 | 127 | 89 | 61 | 191 | 16 | 156 | 130 | 95 | | 66 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 17 | 16 | 347 | 179 | 51 | 0 | 66 | 1 | 72 | | 1 | 149 | 135 | 101 | 184 | 168 | 82 | 181 | 177 | |
| | 21 | 229 | 12 | 258 | 43 | 79 | 78 | 2 | 76 | | 6 | 151 | 149 | 228 | 121 | 0 | 67 | 45 | 114 | |
| | 36 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 10 | 167 | 15 | 126 | 29 | 144 | 235 | 153 | 93 | |
| | 0 | 40 | 241 | 229 | 90 | 170 | 176 | 173 | 39 | | 45 | 67 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 15 | 0 | 40 | 241 | 229 | 90 | 170 | 176 | 173 | | | 39 | 67 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

ตารางที่ 2.7 รายละเอียดค่า $V_{i,j}$ สำหรับกราฟฐาน 2 ของ LDPC

| H_{BG} | | $V_{i,j}$ | | | | | | | | H_{BG} | | $V_{i,j}$ | | | | | | | |
|---------------|------------------|--------------------|-----|-----|-----|-----|-----|-----|-----|---------------|------------------|--------------------|-----|-----|-----|-----|-----|-----|-----|
| Row index i | Column index j | Set index i_{LS} | | | | | | | | Row index i | Column index j | Set index i_{LS} | | | | | | | |
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | 0 | 9 | 174 | 0 | 72 | 3 | 156 | 143 | 145 | 16 | 26 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1 | 117 | 97 | 0 | 110 | 26 | 143 | 19 | 131 | 17 | 1 | 254 | 158 | 0 | 48 | 120 | 134 | 57 | 196 |
| | 2 | 204 | 166 | 0 | 23 | 53 | 14 | 176 | 71 | 17 | 5 | 124 | 23 | 24 | 132 | 43 | 23 | 201 | 173 |
| | 3 | 26 | 66 | 0 | 181 | 35 | 3 | 165 | 21 | 11 | 114 | 9 | 109 | 206 | 65 | 62 | 142 | 195 | |
| | 6 | 189 | 71 | 0 | 95 | 115 | 40 | 196 | 23 | 12 | 64 | 6 | 18 | 2 | 42 | 163 | 35 | 218 | |
| | 9 | 205 | 172 | 0 | 8 | 127 | 123 | 13 | 112 | 27 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 10 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 220 | 186 | 0 | 68 | 17 | 173 | 129 | 128 | |
| | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 6 | 194 | 6 | 18 | 16 | 106 | 31 | 203 | 211 | |
| | 0 | 167 | 27 | 137 | 53 | 19 | 17 | 18 | 142 | 7 | 50 | 46 | 86 | 156 | 142 | 22 | 140 | 210 | |
| | 3 | 166 | 36 | 124 | 156 | 94 | 65 | 27 | 174 | 28 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| | 4 | 253 | 48 | 0 | 115 | 104 | 63 | 3 | 183 | 0 | 87 | 58 | 0 | 35 | 79 | 13 | 110 | 39 | |
| 5 | 125 | 92 | 0 | 156 | 66 | 1 | 102 | 27 | 1 | 20 | 42 | 158 | 138 | 28 | 135 | 124 | 84 | | |
| 6 | 226 | 31 | 88 | 115 | 84 | 55 | 185 | 96 | 10 | 185 | 156 | 154 | 86 | 41 | 145 | 52 | 88 | | |
| 7 | 156 | 187 | 0 | 200 | 98 | 37 | 17 | 23 | 29 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 8 | 224 | 185 | 0 | 29 | 69 | 171 | 14 | 9 | 1 | 26 | 76 | 0 | 6 | 2 | 128 | 196 | 117 | | |
| 9 | 252 | 3 | 55 | 31 | 50 | 133 | 180 | 167 | 4 | 105 | 61 | 148 | 20 | 103 | 52 | 35 | 227 | | |
| 11 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 11 | 29 | 153 | 104 | 141 | 78 | 173 | 114 | 6 | | |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 30 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 0 | 81 | 25 | 20 | 152 | 95 | 98 | 126 | 74 | 0 | 76 | 157 | 0 | 80 | 91 | 156 | 10 | 238 | | |
| 1 | 114 | 114 | 94 | 131 | 106 | 168 | 163 | 31 | 8 | 42 | 175 | 17 | 43 | 75 | 166 | 122 | 13 | | |
| 3 | 44 | 117 | 99 | 46 | 92 | 107 | 47 | 3 | 13 | 210 | 67 | 33 | 81 | 81 | 40 | 23 | 11 | | |
| 4 | 52 | 110 | 9 | 191 | 110 | 82 | 183 | 53 | 31 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 8 | 240 | 114 | 108 | 91 | 111 | 142 | 132 | 155 | 1 | 222 | 20 | 0 | 49 | 54 | 18 | 202 | 195 | | |
| 10 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 2 | 63 | 52 | 4 | 1 | 132 | 163 | 126 | 44 | | |
| 12 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 32 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 23 | 106 | 0 | 156 | 68 | 110 | 52 | 5 | | |
| 1 | 8 | 136 | 38 | 185 | 120 | 53 | 36 | 239 | 3 | 235 | 86 | 75 | 54 | 115 | 132 | 170 | 94 | | |
| 2 | 58 | 175 | 15 | 6 | 121 | 174 | 48 | 171 | 5 | 238 | 95 | 158 | 134 | 56 | 150 | 13 | 111 | | |
| 4 | 158 | 113 | 102 | 36 | 22 | 174 | 18 | 95 | 33 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 5 | 104 | 72 | 146 | 124 | 4 | 127 | 111 | 110 | 1 | 46 | 182 | 0 | 153 | 30 | 113 | 113 | 81 | | |
| 6 | 209 | 123 | 12 | 124 | 73 | 17 | 203 | 159 | 2 | 139 | 153 | 69 | 88 | 42 | 108 | 161 | 19 | | |
| 7 | 54 | 118 | 57 | 110 | 49 | 89 | 3 | 199 | 9 | 8 | 64 | 87 | 63 | 101 | 61 | 88 | 130 | | |
| 8 | 18 | 28 | 53 | 156 | 128 | 17 | 191 | 43 | 34 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 9 | 128 | 186 | 46 | 133 | 79 | 105 | 160 | 75 | 0 | 228 | 45 | 0 | 211 | 128 | 72 | 197 | 66 | | |
| 10 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 1 | 5 | 156 | 21 | 65 | 94 | 63 | 136 | 194 | 95 | | |
| 13 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 35 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 0 | 179 | 72 | 0 | 200 | 42 | 86 | 43 | 29 | 2 | 29 | 67 | 0 | 90 | 142 | 36 | 164 | 146 | | |
| 1 | 214 | 74 | 136 | 16 | 24 | 67 | 27 | 140 | 7 | 143 | 137 | 100 | 6 | 28 | 38 | 172 | 66 | | |
| 11 | 71 | 29 | 157 | 101 | 51 | 83 | 117 | 180 | 12 | 160 | 55 | 13 | 221 | 100 | 53 | 49 | 190 | | |
| 14 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 122 | 85 | 7 | 6 | 133 | 145 | 161 | 86 | | |
| 0 | 231 | 10 | 0 | 185 | 40 | 79 | 136 | 121 | 36 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 1 | 41 | 44 | 131 | 138 | 140 | 84 | 49 | 41 | 0 | 8 | 103 | 0 | 27 | 13 | 42 | 168 | 64 | | |
| 5 | 194 | 121 | 142 | 170 | 84 | 35 | 36 | 169 | 6 | 151 | 50 | 32 | 118 | 10 | 104 | 193 | 181 | | |
| 7 | 159 | 80 | 141 | 219 | 137 | 103 | 132 | 88 | 37 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 11 | 103 | 48 | 64 | 193 | 71 | 60 | 62 | 207 | 1 | 98 | 70 | 0 | 216 | 106 | 64 | 14 | 7 | | |
| 15 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 101 | 111 | 126 | 212 | 77 | 24 | 186 | 144 | | |
| 0 | 155 | 129 | 0 | 123 | 109 | 47 | 7 | 137 | 5 | 135 | 168 | 110 | 193 | 43 | 149 | 46 | 16 | | |
| 5 | 228 | 92 | 124 | 55 | 87 | 154 | 34 | 72 | 38 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 7 | 45 | 100 | 99 | 31 | 107 | 10 | 198 | 172 | 0 | 18 | 110 | 0 | 108 | 133 | 139 | 50 | 25 | | |
| 9 | 28 | 49 | 45 | 222 | 133 | 155 | 168 | 124 | 4 | 28 | 17 | 154 | 61 | 25 | 161 | 27 | 57 | | |
| 11 | 158 | 184 | 148 | 209 | 139 | 29 | 12 | 56 | 39 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 16 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 71 | 120 | 0 | 106 | 87 | 84 | 70 | 37 | | |
| 1 | 129 | 80 | 0 | 103 | 97 | 48 | 163 | 86 | 5 | 240 | 154 | 35 | 44 | 56 | 173 | 17 | 139 | | |
| 5 | 147 | 186 | 45 | 13 | 135 | 125 | 78 | 186 | 7 | 9 | 52 | 51 | 185 | 104 | 93 | 50 | 221 | | |
| 7 | 140 | 16 | 148 | 105 | 35 | 24 | 143 | 87 | 9 | 84 | 56 | 134 | 176 | 70 | 29 | 6 | 17 | | |
| 11 | 3 | 102 | 96 | 150 | 108 | 47 | 107 | 172 | 40 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 13 | 116 | 143 | 78 | 181 | 65 | 55 | 58 | 154 | 1 | 106 | 3 | 0 | 147 | 80 | 117 | 115 | 201 | | |
| 17 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 1 | 170 | 20 | 182 | 139 | 148 | 189 | 46 | | |
| 0 | 142 | 118 | 0 | 147 | 70 | 53 | 101 | 176 | 41 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 1 | 94 | 70 | 65 | 43 | 69 | 31 | 177 | 169 | 0 | 242 | 84 | 0 | 108 | 32 | 116 | 110 | 179 | | |
| 12 | 230 | 152 | 87 | 152 | 88 | 161 | 22 | 225 | 5 | 44 | 8 | 20 | 21 | 89 | 73 | 0 | 14 | | |
| 18 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 12 | 166 | 17 | 122 | 110 | 71 | 142 | 163 | 116 | | |
| 1 | 203 | 28 | 0 | 2 | 97 | 104 | 186 | 167 | 42 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 8 | 205 | 132 | 97 | 30 | 40 | 142 | 27 | 238 | 2 | 132 | 165 | 0 | 71 | 135 | 105 | 163 | 46 | | |
| 10 | 61 | 185 | 51 | 184 | 24 | 99 | 205 | 48 | 7 | 164 | 179 | 88 | 12 | 6 | 137 | 173 | 2 | | |
| 11 | 247 | 178 | 85 | 83 | 49 | 64 | 81 | 68 | 10 | 235 | 124 | 13 | 109 | 2 | 29 | 179 | 106 | | |
| 19 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 43 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 0 | 11 | 59 | 0 | 174 | 46 | 111 | 125 | 38 | 0 | 147 | 173 | 0 | 29 | 37 | 11 | 197 | 184 | | |
| 1 | 185 | 104 | 17 | 150 | 41 | 25 | 60 | 217 | 12 | 85 | 177 | 19 | 201 | 25 | 41 | 191 | 135 | | |
| 6 | 0 | 22 | 156 | 8 | 101 | 174 | 177 | 208 | 13 | 36 | 12 | 78 | 69 | 114 | 162 | 193 | 141 | | |
| 7 | 117 | 52 | 20 | 56 | 96 | 23 | 51 | 232 | 44 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 20 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 57 | 77 | 0 | 91 | 60 | 126 | 157 | 85 | | |
| 0 | 11 | 32 | 0 | 99 | 28 | 91 | 39 | 178 | 5 | 40 | 184 | 157 | 165 | 137 | 152 | 167 | 225 | | |
| 7 | 236 | 92 | 7 | 138 | 30 | 175 | 29 | 214 | 11 | 63 | 18 | 6 | 55 | 93 | 172 | 181 | 175 | | |
| 9 | 210 | 174 | 4 | 110 | 116 | 24 | 35 | 168 | 45 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 13 | 56 | 154 | 2 | 99 | 64 | 141 | 8 | 51 | 0 | 140 | 25 | 0 | 1 | 121 | 73 | 197 | 178 | | |
| 21 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 38 | 151 | 63 | 175 | 129 | 154 | 167 | 112 | | |
| 1 | 63 | 39 | 0 | 46 | 33 | 122 | 18 | 124 | 7 | 154 | 170 | 82 | 83 | 26 | 129 | 179 | 106 | | |
| 3 | 111 | 93 | 113 | 217 | 122 | 11 | 155 | 122 | 46 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 11 | 14 | 11 | 48 | 109 | 131 | 4 | 49 | 72 | 10 | 219 | 37 | 0 | 40 | 97 | 167 | 181 | 154 | | |
| 22 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 13 | 151 | 31 | 144 | 12 | 56 | 38 | 193 | 114 | | |
| 0 | 83 | 49 | 0 | 37 | 76 | 29 | 32 | 48 | 47 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | |
| 1 | 2 | 125 | 112 | 113 | 37 | 91 | 53 | 57 | 1 | 31 | 84 | 0 | 37 | 1 | 112 | 157 | 42 | | |
| 8 | 38 | 35 | 102 | 143 | 62 | 27 | 95 | 167 | 5 | 66 | 151 | 93 | 97 | 70 | 7 | 173 | 41 | | |

| H_{BG} | | $V_{i,j}$ | | | | | | | H_{BG} | | $V_{i,j}$ | | | | | | | | | |
|---------------|------------------|--------------------|-----|-----|-----|-----|-----|-----|---------------|------------------|--------------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Row index i | Column index j | Set index i_{LS} | | | | | | | Row index i | Column index j | Set index i_{LS} | | | | | | | | | |
| | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | | | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | |
| | 13 | 222 | 166 | 26 | 140 | 47 | 127 | 186 | 219 | | 11 | 38 | 190 | 19 | 46 | 1 | 19 | 191 | 105 | |
| | 23 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 48 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 14 | 1 | 115 | 19 | 0 | 36 | 143 | 11 | 91 | 82 | 39 | 0 | 239 | 93 | 0 | 106 | 119 | 109 | 181 | 167 | |
| | 6 | 145 | 118 | 138 | 95 | 51 | 145 | 20 | 232 | | | 7 | 172 | 132 | 24 | 181 | 32 | 6 | 157 | 45 |
| | 11 | 3 | 21 | 57 | 40 | 130 | 8 | 52 | 204 | | | 12 | 34 | 57 | 138 | 154 | 142 | 105 | 173 | 189 |
| | 13 | 232 | 163 | 27 | 116 | 97 | 166 | 109 | 162 | | | 49 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 24 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | | 2 | 0 | 103 | 0 | 98 | 6 | 160 | 193 | 78 |
| 15 | 0 | 51 | 68 | 0 | 116 | 139 | 137 | 174 | 38 | 40 | 10 | 75 | 107 | 36 | 35 | 73 | 156 | 163 | 67 | |
| | 10 | 175 | 63 | 73 | 200 | 96 | 103 | 108 | 217 | | | 13 | 120 | 163 | 143 | 36 | 102 | 82 | 179 | 180 |
| | 11 | 213 | 81 | 99 | 110 | 128 | 40 | 102 | 157 | | | 50 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 25 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | | 41 | 1 | 129 | 147 | 0 | 120 | 48 | 132 | 191 | 53 |
| 1 | 203 | 87 | 0 | 75 | 48 | 78 | 125 | 170 | | 5 | | 229 | 7 | 2 | 101 | 47 | 6 | 197 | 215 | |
| 9 | 142 | 177 | 79 | 158 | 9 | 158 | 31 | 23 | | 11 | | 118 | 60 | 55 | 81 | 19 | 8 | 167 | 230 | |
| 11 | 8 | 135 | 111 | 134 | 28 | 17 | 54 | 175 | | 51 | | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 16 | 12 | 242 | 64 | 143 | 97 | 8 | 165 | 176 | 202 | | | | | | | | | | | |

2.4.5 การปรับอัตราหัส (Rate Matching)

การปรับอัตราหัส (Rate matching) คือกระบวนการปรับขนาดคำรหัสให้มีขนาดเท่ากับขนาดบล็อกการขนส่ง (Transport Block) สำหรับ downlink และ uplink เนื่องจากในมาตรฐาน 5G ได้มีการนำเทคนิค Incremental Redundancy - Hybrid Automatic Repeat request (IR-HARQ) [7] มาใช้งานในการแก้ไขความผิดพลาด (Correct Errors) ในการส่งข้อมูลระหว่างโหนด และตัวอุปกรณ์ของผู้ใช้งาน (User's Device) เมื่อตัวอุปกรณ์ได้มีการร้องขอให้ส่งข้อมูลใหม่โหนดจะมีการส่งข้อมูลเฉพาะในส่วนบิตพาริตีที่เพิ่มขึ้นไปให้ตัวอุปกรณ์เพื่อใช้ในการแก้ไขข้อมูล แต่เนื่องจากปัญหาในเรื่องของขนาดบัพเฟอร์กระบวนการ HARQ ที่จำกัด กระบวนการปรับอัตราหัสจึงเป็นเทคนิคที่ถูกนำมาใช้งานในการแก้ปัญหานี้ โดยจำเป็นต้องประมาณค่าทรูพุตสูงสุด (Maximum Throughput) ซึ่งสามารถคำนวณได้ดังนี้

$$\text{กรณี } j \leq C' - \text{mod } G / N_L \cdot Q_m, C' - 1$$

$$E_r = N_L \cdot Q_m \cdot \left\lfloor \frac{G}{N_L \cdot Q_m \cdot C'} \right\rfloor \quad (11)$$

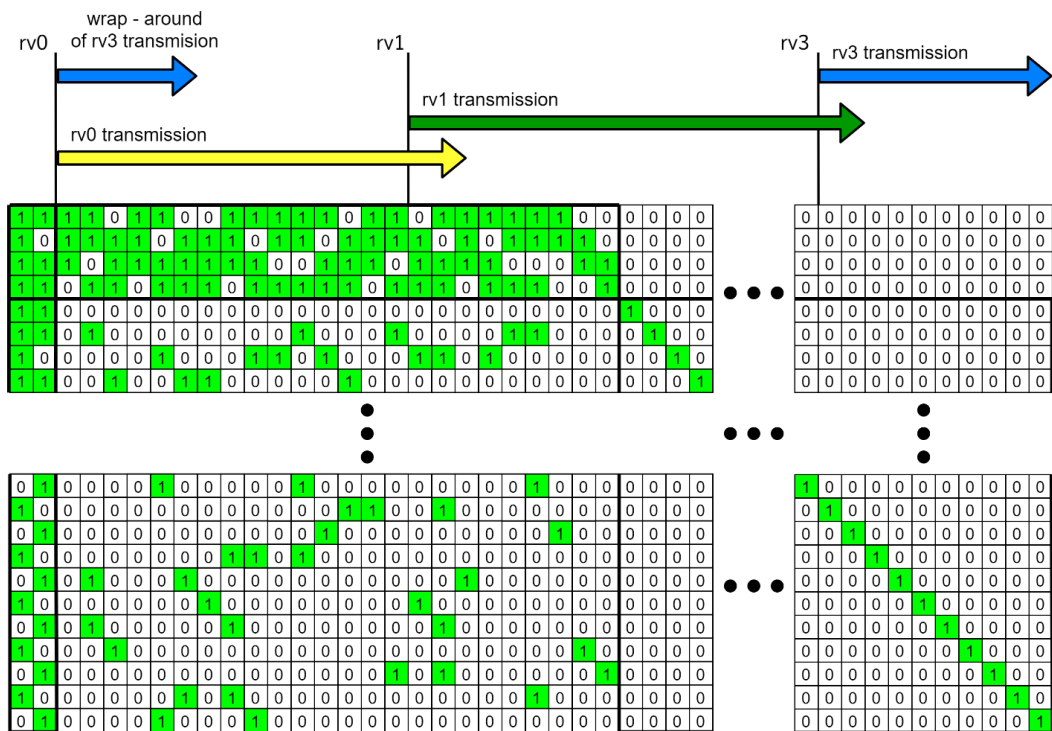
กรณีอื่น ๆ

$$E_r = N_L \cdot Q_m \cdot \left\lceil \frac{G}{N_L \cdot Q_m \cdot C'} \right\rceil \quad (12)$$

- เมื่อ j คือหมายเลขของบล็อกจากการแบ่งส่วนข้อมูล (Segmentation)
 E_r คือความยาวเอาต์พุตของการปรับอัตราหัส (Rate Matching)
 N_L คือลำดับของชั้นการขนส่ง (Transport layer) มีค่าเป็น 1, 2, 3 หรือ 4

- Q_m คือลำดับการมอดูเลต มีค่า 1 สำหรับ BPSK, ค่า 2 สำหรับ QPSK, ค่า 4 สำหรับ 16-QAM, ค่า 6 สำหรับ 64-QAM หรือ ค่า 8 สำหรับ 256-QAM
- G คือจำนวนบิตทั้งหมดที่สามารถใช้ได้สำหรับการส่งของบล็อกการขนส่ง
- C' คือจำนวนบิตทั้งหมดที่ผ่านการแบ่งส่วนข้อมูล

การปรับอัตรารหัสสำหรับรหัส LDPC ตามมาตรฐาน 5G จะมีการกำหนดค่า Redundancy Version (rv) ซึ่ง rv จะมีความสอดคล้องกับตำแหน่งคอลัมน์ของกราฟฐาน โดยกราฟฐานจะถูกแบ่งออกเป็น 4 ส่วนโดยไม่รวมบิต puncture ใน 2 คอลัมน์แรก บิตข้อมูลที่ส่งออกจะประกอบไปด้วยเอาต์พุตการเข้ารหัสเริ่มต้นจากตำแหน่งคอลัมน์ดังรูปที่ 2.9 โดยสามารถคำนวณตำแหน่งคอลัมน์เริ่มต้น (k_0) ของแต่ละ rv ได้จากตารางที่ 2.8 ตามค่าของ rv_{id} และกราฟฐานของ LDPC



รูปที่ 2.9 การปรับอัตราสำหรับรหัส LDPC ตามมาตรฐาน 5G

ตารางที่ 2.8 ตำแหน่งเริ่มต้น k_0 ของ rv ที่แตกต่างกัน

| rv_{id} | k_0 | |
|-----------|---|---|
| | กราฟฐาน 1 | กราฟฐาน 2 |
| 0 | 0 | 0 |
| 1 | $\left\lfloor \frac{17N_{cb}}{66Z_c} \right\rfloor Z_c$ | $\left\lfloor \frac{13N_{cb}}{50Z_c} \right\rfloor Z_c$ |
| 2 | $\left\lfloor \frac{33N_{cb}}{66Z_c} \right\rfloor Z_c$ | $\left\lfloor \frac{25N_{cb}}{50Z_c} \right\rfloor Z_c$ |
| 3 | $\left\lfloor \frac{56N_{cb}}{66Z_c} \right\rfloor Z_c$ | $\left\lfloor \frac{43N_{cb}}{50Z_c} \right\rfloor Z_c$ |

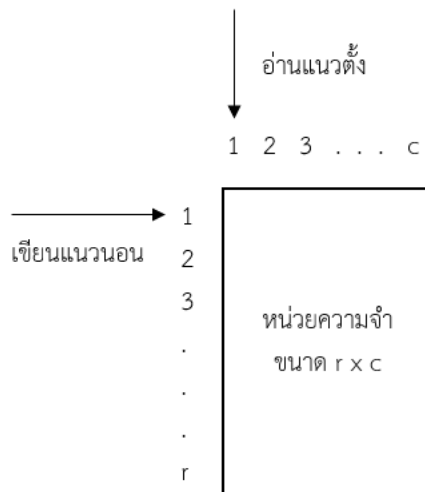
เมื่อ rv_{id} คือหมายเลข rv สำหรับการส่งข้อมูลมีค่าเป็น 0, 1, 2 หรือ 3

k_0 คือตำแหน่งเริ่มต้นในการส่งในแต่ละ rv

การส่งครั้งแรกจะต้องเริ่มต้นด้วย rv_0 ซึ่งเป็นตำแหน่งคอลัมน์ที่ 3 โดยในการส่งข้อมูลซ้ำแต่ละครั้งอาจมีคอลัมน์ที่ทับซ้อนกันหรือไม่ก็ได้ หากความยาวการส่งเริ่มต้นจากตำแหน่งคอลัมน์ของ rv ขนาดของเมทริกซ์ตรวจสอบพาริตี ดังนั้นการส่งข้อมูลดังกล่าวจะรวมเอาต์พุตการเข้ารหัสของคอลัมน์ก่อนหน้าโดยไม่รวมบิต Puncture 2 คอลัมน์แรก

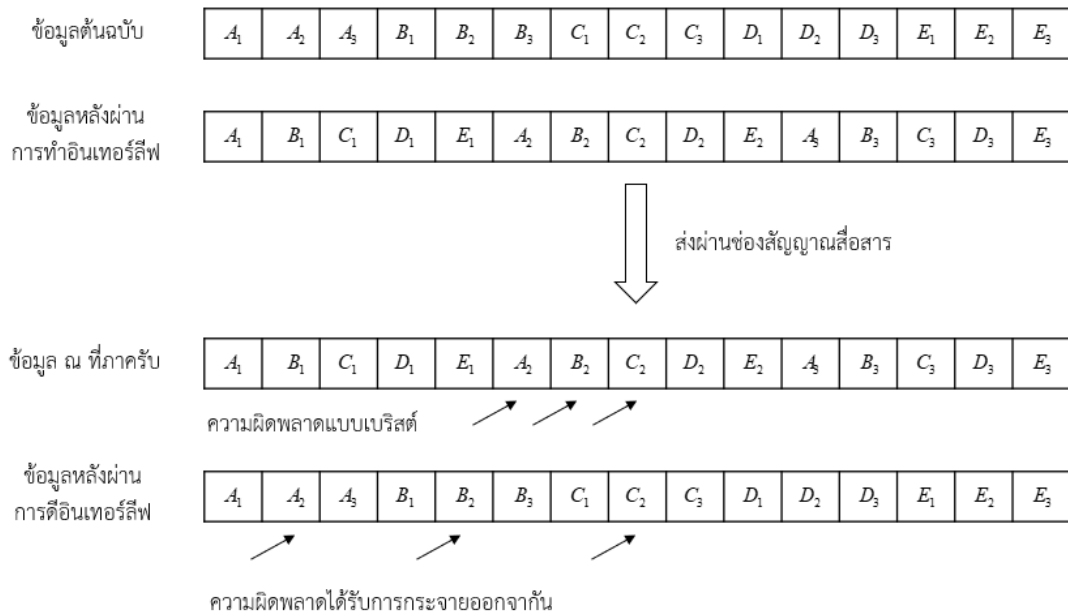
2.4.6 การแทรกสลับบิต (Bit Interleaving)

เทคนิคการแทรกสลับ (interleaving) [8] คือการป้องกันความผิดพลาดของข้อมูลที่ส่งไปบนช่องสัญญาณด้วยกระบวนการสลับบิตหรือไบนารีข้อมูลทีภาคส่ง หากมีสัญญาณรบกวนเกิดขึ้นบนช่องสัญญาณเป็นผลให้ข้อมูลที่ส่งนั้นมีความผิดพลาดติดกัน เมื่อทำกระบวนการแทรกสลับบิตกลับ (De-Interleaving) จะทำให้บิตผิดพลาดที่ติดกันมีการกระจายตัวออก



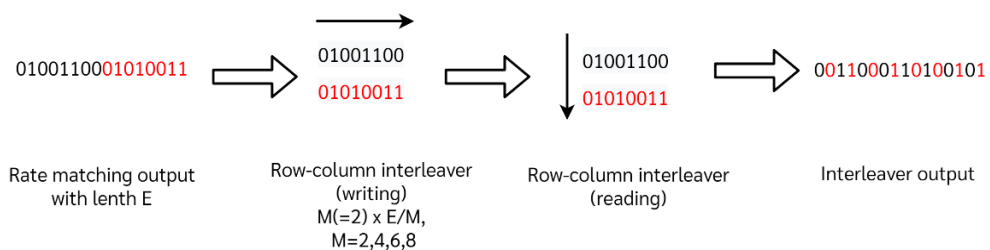
รูปที่ 2.10 ตัวอย่างวิธีแทรกสลับบิต

หลักการของการแทรกสลับบิต [9] คือจะแบ่งบิตข้อมูลออกเป็นบล็อกย่อย ๆ ขนาด $r \times c$ บิต โดย r คือ จำนวนแถว และ c คือ จำนวนคอลัมน์ จากนั้นก็จะทำการสลับตำแหน่งของแต่ละบิต ภายในบล็อกเดียวกันก่อนการส่งออก โดยอาศัยวิธีการคือ นำข้อมูลที่จะส่งมาเขียนลงในหน่วยความจำทีละแถวตามลำดับของข้อมูลที่เข้ามาจนครบหนึ่งบล็อก จากนั้นทำการอ่านข้อมูลในแนวตั้งเพื่อส่งออกทีละคอลัมน์จนหมด ดังที่แสดงในรูปที่ 2.10 และเมื่อบิตข้อมูลเหล่านี้ถึงที่ภาครับแล้วก็จะทำการสลับตำแหน่งของบิตให้กลับเป็นปกติโดยอาศัยกระบวนการที่กลับกันกับที่ภาส่งดังตัวอย่างรูปที่ 2.11 โดยข้อมูลที่จะส่งถูกแบ่งออกเป็นบล็อกขนาด 5×3 บิต หลังจากที่ได้ทำการอินเทอร์ลีฟ ลำดับการส่งบิตข้อมูลจะแตกต่างไปจากข้อมูลเดิม สังเกตว่าถึงแม้บิตข้อมูลเหล่านี้จะได้รับผลกระทบจากความผิดพลาดแบบเบริสต์ (Burst Error) เนื่องจากความไม่เป็นอุดมคติของช่องสัญญาณ อย่างไรก็ตามเมื่อข้อมูลมาถึงภาครับจะมีการแทรกสลับบิตกลับเพื่อให้บิตข้อมูลมีลำดับที่ถูกต้องเหมือนเดิม หมายเหตุว่ากระบวนการแทรกสลับบิตจะสามารถทำให้ความผิดพลาดมีการกระจายตัวตลอดทั้งบล็อกข้อมูลในรูปแบบที่ต่างไปจากเดิม ซึ่งการกระจายของความผิดพลาดนี้ช่วยให้ระบบไม่ต้องอาศัยวิธีการเข้ารหัสที่มีขีดความสามารถในการแก้บิตที่ผิดพลาดที่สูงนัก



รูปที่ 2.11 ตัวอย่างการแทรกสลับบิตและการแทรกสลับบิตกลับ

ในรหัส LDPC สำหรับมาตรฐาน 5G [10] กระบวนการแทรกสลับบิตถูกนำไปใช้หลังจากการเลือกบิตในการส่งซึ่งทำโดยการแทรกระหว่างแถวและคอลัมน์ดังรูปที่ 2.12 ซึ่งขนาดแถวขึ้นอยู่กับลำดับการมอดูเลต 4/16/64/256 ของการมอดูเลตเชิงตัวเลขทางขนาดและเฟส (Quadrature Amplitude Modulation: QAM)



รูปที่ 2.12 กระบวนการแทรกสลับบิตสำหรับรหัส LDPC ตามมาตรฐาน 5G

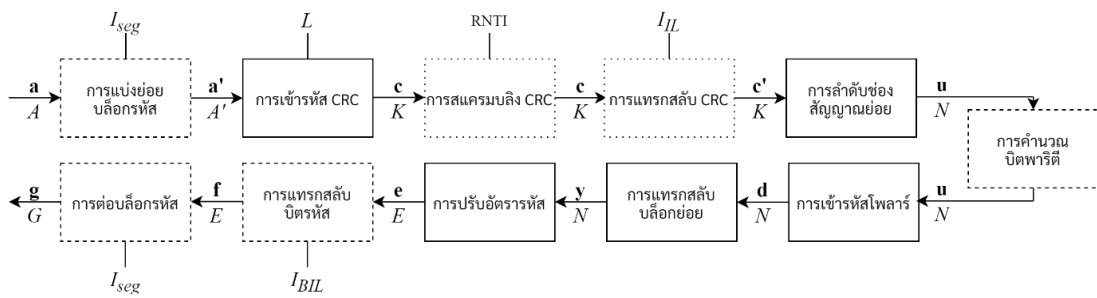
2.4.7 การต่อเรียงบล็อกรหัส (Code Block Concatenation)

การต่อเรียงบล็อกรหัสประกอบไปด้วยการต่อเรียงบล็อกรหัสเอาต์พุตที่ได้จากการปรับอัตรารหัสซึ่งมีการทำงานโดยจะกำหนดให้ลำดับอินพุตบิตสำหรับกระบวนการต่อเรียงบล็อกรหัสคือลำดับ f_{rk} สำหรับ $r=0, \dots, C-1$ และ $k=0, \dots, E_r-1$ เมื่อ E_r คือจำนวนบิตหลังมีการปรับอัตรารหัสแล้วสำหรับบล็อกรหัสลำดับที่ r จะได้ลำดับเอาต์พุตของบิตจากกระบวนการต่อเรียงบล็อกรหัสคือลำดับ g_k สำหรับ $k=0, \dots, G-1$

2.5 รายละเอียดการเข้ารหัสช่องสัญญาณชนิด Polar codes มาตรฐาน 5G

2.5.1 ภาพรวมการเข้ารหัสโพลาร์มาตรฐาน 5G

โครงการนี้จะพัฒนาการเข้ารหัสโพลาร์มาตรฐาน 5G ตามเอกสาร 3GPP 38.212 [1] โดย การเข้ารหัสช่องสัญญาณควบคุมจะมีความแตกต่างกันในกรณีช่องสัญญาณ downlink และ uplink ดังรูปที่ 2.13 และสามารถสรุปค่าพารามิเตอร์ของรหัสโพลาร์ในมาตรฐาน 5G [11]



รูปที่ 2.13 กระบวนการเข้ารหัสโพลาร์ในมาตรฐาน 5G

จากรูปที่ 2.13 กระบวนการเข้ารหัสในกรณีช่องสัญญาณ uplink จะมีกระบวนการที่เพิ่มเติม จากกรณีช่องสัญญาณ downlink และกรณีช่องสัญญาณ downlink จะมีกระบวนการแทรกกลับ CRC เข้ามา โดยกระบวนการที่เหมือนกันทั้งสองกรณีจะมีกระบวนการทำงานที่คล้ายกัน กระบวนการเข้ารหัสจะเข้ารหัสตามเงื่อนไขของค่าพารามิเตอร์ต่าง ๆ ตามจำนวนข้อมูลที่จะนำมาเข้ารหัสและ ประเภทช่องสัญญาณ พารามิเตอร์ที่สำคัญประกอบด้วย

1) ความยาวเพย์โหลดขาเข้า A จะถูกกำหนดจากข้อมูลที่ต้องการเข้ารหัสช่องสัญญาณจากการสื่อสารขึ้นบน มีความยาวสูงสุด $A \leq 1706$

2) ความยาวบิต CRC L ตามมาตรฐานได้กำหนดพหุนาม CRC ไว้ดังนี้

$$g_{24A}(x) = x^{24} + x^{23} + x^{18} + x^{17} + x^{14} + x^{11} + x^{10} + x^7 + x^6 + x^5 + x^4 + x^3 + x + 1$$

$$g_{24B}(x) = x^{24} + x^{23} + x^6 + x^5 + x + 1$$

$$g_{24C}(x) = x^{24} + x^{23} + x^{21} + x^{20} + x^{17} + x^{15} + x^{13} + x^{12} + x^8 + x^4 + x^2 + x + 1$$

$$g_{16}(x) = x^{16} + x^{12} + x^5 + 1$$

$$g_{11}(x) = x^{11} + x^{10} + x^9 + x^5 + 1$$

$$g_6(x) = x^6 + x^5 + 1$$

โดยพหุนามที่ใช้กับรหัสโพลาร์ในช่องสัญญาณ UCI คือพหุนาม g_6 ที่ความยาว $L=6$ และ g_{11} ที่ความยาว $L=11$ และช่องสัญญาณ DCI คือพหุนาม g_{24C} ที่ความยาว $L=24$

3) code dimension K คือจำนวนบิตที่ทำการเข้ารหัสโพลาร์ สามารถประกอบไปด้วยบิตข้อมูล บิต CRC รวมถึงบิตพาริตี โดยที่ $K < N$

4) ความยาวคำรหัสแม่ (mother code length) $N = 2^n$ ซึ่งเป็นข้อจำกัดความยาวคำรหัสของการเข้ารหัสโพลาร์ โดยค่า n ที่จะใช้งานจะถูกคำนวณโดยสมการที่ 13

$$n = \max(\min(n_1, n_2, n_{\max}), n_{\min}) \quad (13)$$

โดยที่ n_{\min} และ n_{\max} คือขอบเขตเลขชี้กำลังล่างและบนตามลำดับ สำหรับช่องสัญญาณ downlink $n_{\min} = 5$ และ $n_{\max} = 9$ สำหรับช่องสัญญาณ uplink $n_{\min} = 5$ และ $n_{\max} = 10$ สำหรับ n_1 ถือเป็นขอบเขตที่มีเงื่อนไขตามการปรับอัตราหัสและ n_2 เป็นขอบเขตที่มีเงื่อนไขตามอัตราหัสต่ำสุด และค่า K คือจำนวนบิตก่อนการเข้ารหัสตั้งสมการที่ 14 และ 15

$$n_1 = \begin{cases} \lfloor \log_2 E \rfloor & \text{หาก } E \leq (9/8) \cdot 2^{\lfloor \log_2 E \rfloor} \text{ และ} \\ \lceil \log_2 E \rceil & \text{ถ้า } K/E < 9/16 \\ & \text{อื่น ๆ} \end{cases} \quad (14)$$

$$n_2 = \lceil \log_2 (K/R_{\min}) \rceil \quad \text{โดยที่ } R_{\min} = 1/8 \quad (15)$$

5) ความยาวคำรหัส E คือความยาวคำรหัสหลังการปรับอัตราหัสจากความยาวคำรหัสแม่ N ให้ได้อัตราหัส R ที่ต้องการ มีความยาวสูงสุด $E \leq 8192$ หากความยาวคำรหัส $E < N$ คำรหัสแม่จะถูกทำการพังก์เจอร์ (puncturing) หรือการช็อตเทน (shortening) หาก $E > N$ คำรหัสแม่บางบิตจะถูกส่งซ้ำ ในกระบวนการปรับอัตราหัส ความยาวคำรหัส E ที่ถูกกำหนดจากการสื่อสารชั้นถัดไป

6) อัตราหัส $R = A/E$ คืออัตราส่วนระหว่างความยาวเพย์โหลดขาเข้าต่อความยาวคำรหัสที่ผ่านการปรับอัตราหัส

7) ความยาวเพย์โหลดขาออก G เป็นความยาวของบิตหลังผ่านกระบวนการเข้ารหัสช่องสัญญาณทั้งหมด สามารถมีความยาวได้เท่ากับหรือมากกว่าความยาวคำรหัส E ตามประเภทช่องสัญญาณการสื่อสาร บิตหลังผ่านกระบวนการเข้ารหัสช่องสัญญาณทั้งหมดจะถูกบรรจุใส่เพย์โหลดเพื่อผ่านระบบการสื่อสารต่อไป

8) ความยาวบิตที่ตัดออก U สำหรับการปรับอัตรารหัสในกรณีการฟังก์ชันเซอร์และการช็อตเทน

9) ความยาวตำแหน่งบิตแก้ไขสำหรับการปรับอัตรารหัส สำหรับวิธีการฟังก์ชันเซอร์จะมีความยาว T และวิธีการช็อตเทนจะมีความยาว U เท่ากับความยาวบิตที่ตัดออก

10) ความยาวบิตข้อมูลและบิตแก้ไขจะแทนด้วย $|Q_I|$ และ $|Q_F|$ ตามลำดับ และให้ n_{PC} คือความยาวบิตพาริตี

ประเภทช่องสัญญาณและการเข้ารหัสช่องสัญญาณที่ใช้งานสามารถสรุปได้ดังตารางที่ 2.9 ค่าพารามิเตอร์ที่แตกต่างกันในแต่ละช่องสัญญาณสามารถสรุปได้ดังตารางที่ 2.10

ตารางที่ 2.9 การเข้ารหัสช่องสัญญาณของแต่ละประเภทช่องสัญญาณ

| ช่องสัญญาณข้อมูล (transport channel: TrCH) | การเข้ารหัสช่องสัญญาณ |
|--|--------------------------------------|
| uplink shared channel (UL-SCH) | low-density parity-check (LDPC) code |
| downlink shared channels (DL-SCH) | |
| paging channel (PCH) | |
| broadcast channel (BCH) | polar code |
| ช่องสัญญาณควบคุม (control information: CI) | การเข้ารหัสช่องสัญญาณ |
| downlink control information (DCI) | polar code |
| uplink control information (UCI) | |
| | block code |

ตารางที่ 2.10 พารามิเตอร์และขอบเขตของรหัสโพลาร์สำหรับช่องสัญญาณต่าง ๆ ในมาตรฐาน 5G

| | | UCI | | | | DCI | BCH |
|---|---------------|---|--|---------------------|---------------|------|-----|
| | | $A \geq 20$ | | $12 \leq A \leq 19$ | | | |
| | | $(A \geq 1013) \vee$ $(A \geq 360 \wedge G \geq 1088)$ | $(A < 360) \vee$ $(A < 1013 \wedge G < 1088)$ | $E - A \leq 175$ | $E - A > 175$ | | |
| เลขชี้กำลัง สูงสุดความ ยาวรหัสแม่ | n_{\max} | 10 | | | | 9 | |
| ตัวบ่งชี้การ แทรกกลับ CRC | I_{IL} | 0 | | | | 1 | |
| ตัวบ่งชี้การ แทรกกลับ บิตรหัส | I_{BL} | 1 | | | | 0 | |
| ตัวบ่งชี้การ แบ่งย่อย บล็อกรหัส | I_{seg} | 1 | 0 | | | 0 | |
| ความยาวเพย์ โหลดขาออก สูงสุด | G_{\max} | 16384 | 8192 | | | 8192 | 864 |
| ความยาวเพย์ โหลดขาออก ต่ำสุด | G_{\min} | 31 | | 18 | | 25 | 864 |
| ความยาวเพย์ โหลดขาเข้า สูงสุด | A_{\max} | 1706 | | | | 140 | 32 |
| ความยาวเพย์ โหลดขาเข้า ต่ำสุด | A_{\min} | 12 | | | | 1 | 32 |
| ความยาวบิต CRC | L | 11 | | 6 | | 24 | |
| ความยาวบิต พาริตี ตรวจสอบ | n_{PC} | 0 | | 3 | | 0 | |
| ความยาวบิต พาริตี ตรวจสอบ โดยเลือกจาก น้ำหนักแถวที่ ต่ำสุด | n_{PC}^{wm} | 0 | | 0 | 1 | 0 | |

2.5.2 การแบ่งย่อยบล็อกรหัส

การแบ่งย่อยบล็อกรหัสเป็นการแบ่งย่อยข้อมูลเพย์โหลด **a** ออกเป็น 2 ส่วนและทำการเข้ารหัสแยกกัน เพื่อไม่ให้เกิดความซับซ้อนในการเข้ารหัสมากเกินไป (เกิดจากความยาวข้อมูล)

กรณีช่องสัญญาณ UCI และมีเงื่อนไขว่า $(A \geq 360 \wedge E \geq 1088) \vee A \geq 1013$ ระบายวนนี้จะทำงานและกำหนดตัวบ่งชี้การแบ่งย่อยบล็อกรหัส $I_{seg} = 1$ นอกจากนั้นกำหนดให้ $I_{seg} = 0$ เวกเตอร์ **a** ความยาว A จะถูกแบ่งเป็น 2 เวกเตอร์ **a'** ความยาว $A' = A/2$ หากความยาว A เป็นจำนวนคี่ **a'** ส่วนแรกจะแบ่ง $\lfloor A/2 \rfloor$ บิตแรกจาก **a** และเติมบิต 0 ไว้ด้านหน้า

สำหรับช่องสัญญาณ BCH และ DCI ระบายวนการนี้จะไม่ทำงานและกำหนดให้ $I_{seg} = 0$ เวกเตอร์ **a** จะผ่านไปยังระบายวนการเข้ารหัส CRC ถัดไป

2.5.3 การเข้ารหัส CRC



รูปที่ 2.14 โครงสร้างการเข้ารหัส CRC

การเข้ารหัส CRC จะทำการคำนวณบิต CRC ความยาว L บิตก่อน จากนั้นจะนำบิต CRC ที่คำนวณได้ต่อท้ายกับเวกเตอร์ **a** ความยาว A จากอินพุต (หรือ **a'** ความยาว A' ในกรณีที่ $I_{seg} = 1$) ดังรูปที่ 2.14 โดยจะได้เวกเตอร์ **c** ที่มีความยาว $K = A + L$ (หรือ **c'** ความยาว $K = A' + L$ ในกรณีที่ $I_{seg} = 1$) การเข้ารหัส CRC จะกระทำอยู่ในรูปแบบ systematic อยู่ใน $GF(2)$

กรณีช่องสัญญาณ UCI

ใช้พหุนาม g_6 ในการคำนวณบิต CRC ความยาว $L = 6$ ในกรณีที่เวกเตอร์ **a** มีความยาว $12 \leq A \leq 19$ หากเป็นกรณีที่ $A \geq 20$ จะใช้พหุนาม g_{11} ในการคำนวณบิต CRC ความยาว $L = 11$ สุดท้ายจะได้ผลลัพธ์เป็นเวกเตอร์ **c**

กรณีช่องสัญญาณ BCH

พหุนามที่ใช้คำนวณบิต CRC คือ g_{24C} ซึ่งจะได้บิต CRC ความยาว $L = 24$ เวกเตอร์ **a** ความยาว $A = A_{\min} = A_{\max} = 32$ (ถูกกำหนดตายตัวตามมาตรฐาน) จะถูกสแครมบลิงด้วยเวกเตอร์

ขนาดเท่ากันก่อนหน้ากระบวนการเข้ารหัสช่องสัญญาณ โดยการสแครมบลิงคือการนำบิตของสอง
เวกเตอร์มาทำกระบวนการเอ็กซ์คลูซีฟออร์ จากนั้นจะเข้ารหัส CRC ตามปกติ จะได้ผลลัพธ์เป็น
เวกเตอร์ **c** ความยาว $K = 56$ บิต

กรณีช่องสัญญาณ DCI

พหุนามที่ใช้คำนวณ CRC คือ g_{24C} ซึ่งจะได้บิต CRC ความยาว $L = 24$ การเข้ารหัส CRC
จะแตกต่างจากการเข้ารหัส CRC ทั่วไป โดยตัวตั้งในการคำนวณ CRC จะนำเวกเตอร์ 1 ขนาดเท่ากับ
ความยาวบิต CRC ไว้ด้านหน้าและตามด้วยเวกเตอร์ **a** ความยาว A (ซึ่งต่างจากวิธีทั่วไปที่จะนำ
เวกเตอร์ **a** ไว้ด้านหน้าและตามหลังด้วยเวกเตอร์ 0 ขนาดเท่ากับความยาวบิต CRC ไว้ด้านหลัง) เมื่อ
คำนวณบิต CRC จะนำบิต CRC ต่อด้านหลังเวกเตอร์ **a** ตามปกติ เวกเตอร์ **a** สามารถมีความยาว
ได้ $A_{\min} = 1$ ถึง $A_{\max} = 140$ จะได้ผลลัพธ์เป็นเวกเตอร์ **c**

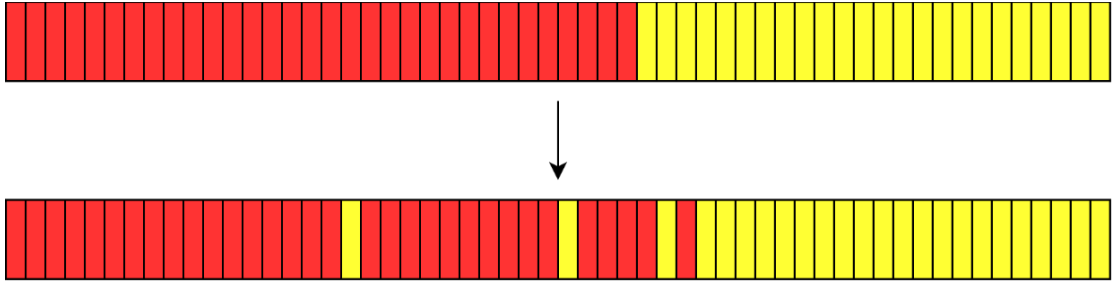
2.5.4 การสแครมบลิง CRC

เวกเตอร์ **c** จากกระบวนการก่อนจะถูกสแครมบลิง ซึ่งจะถูกใช้งานในการทำ (blind
detection หรือ blind decoding) ของการส่งข้อมูลระหว่าง radio access network (RAN) และ
อุปกรณ์ผู้ใช้งาน (user equipment: UE) ในการส่งข้อมูลควบคุมผ่านช่องสัญญาณ DCI อุปกรณ์ฝั่ง
RAN จะไม่มีการใส่ข้อมูลส่วนหัว (header) ในการระบุตัวตนของอุปกรณ์ผู้ใช้งาน แต่จะทำการสแครม
บลิงรหัสเฉพาะตัวที่ตรงกับรหัสเฉพาะตัวของอุปกรณ์ผู้ใช้งานที่ต้องการสื่อสาร จากนั้นอุปกรณ์ RAN
จะส่งข้อมูลในลักษณะออกอากาศให้อุปกรณ์ผู้ใช้งานโดยรอบ จากนั้นอุปกรณ์ผู้ใช้งานจะทำการ
ถอดรหัสของข้อมูล blind detection การสแครมบลิงจะทำให้ข้อมูลที่ไม่ตรงกับอุปกรณ์ผู้ใช้งานมี
โอกาสถอดรหัสผิดสูง และข้อมูลที่ตรงกับอุปกรณ์ผู้ใช้งานสามารถถอดรหัสถูกต้องและสามารถรับ
ข้อมูลได้ ซึ่งรหัสเฉพาะดังกล่าวจะถูกกำหนดโดย radio network temporary identifier (RNTI)
เทคนิค blind detections ดังกล่าวมีการใช้งานตั้งแต่มาตรฐาน 4G

กรณีช่องสัญญาณ DCI ที่มีการเข้ารหัส CRC 24 บิต c_{A+8}^{A+L-1} (บิต CRC 16 บิตท้ายสุด) จะ
ถูกสแครมบลิงด้วยบิต RNTI ความยาวที่เท่ากัน 16 บิต ซึ่งบิต RNTI จะถูกเลือกจากประเภทของ
ช่องสัญญาณและประเภทหน้าที่ของข้อมูล

สำหรับช่องสัญญาณ BCH และ UCI กระบวนการนี้จะไม่ทำงานเวกเตอร์ **c** จะผ่านไปยัง
กระบวนการแทรกสลับ CRC ถัดไป

2.5.5 การแทรกสลับ CRC



รูปที่ 2.15 ตัวอย่างลำดับการแทรกสลับ CRC สำหรับช่องสัญญาณ BCH ที่ $A = 32$ $L = 24$ และ $K = 56$ โดยสีแดงและสีเหลืองคือบิตข้อมูลและบิต CRC ตามลำดับ

เวกเตอร์ **c** ที่ผ่านกระบวนการก่อนหน้าจะถูกแทรกสลับตามลำดับดังตารางที่ 2.11 เพื่อที่จะลดความซับซ้อนในการถอดรหัสจากเทคนิค early termination ซึ่งจะหยุดการถอดรหัสระหว่างทางหากพบความผิดพลาดระหว่างการถอดรหัส ตัวอย่างการแทรกสลับ CRC สำหรับช่องสัญญาณ BCH ได้แสดงดังรูปที่ 2.15

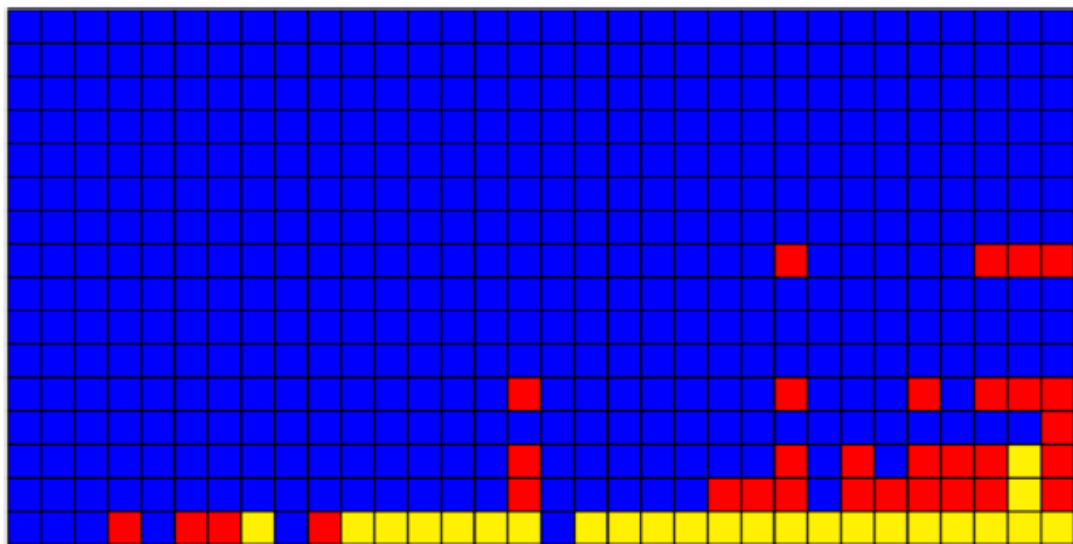
กรณีช่องสัญญาณ BCH และ DCI เนื่องจากเวกเตอร์ **a** สามารถมีหลายความยาวได้ ดังนั้นเวกเตอร์ **c** จึงไม่จำเป็นต้องมีขนาดเท่ากับความยาวลำดับบิตแทรกสลับสูงสุด $K_L^{\max} = 164$ วิธีการแทรกสลับจะนำเวกเตอร์ **c** ต่อท้ายด้วยบิตว่าง (null) จนมีความยาวเท่ากับ K_L^{\max} จากนั้นทำการแทรกสลับตามลำดับการแทรกสลับดังตารางที่ 2.11 และนำบิตว่างออกจากเวกเตอร์ที่แทรกสลับจะได้ผลลัพธ์เป็นเวกเตอร์ **c'**

สำหรับช่องสัญญาณ UCI กระบวนการนี้จะไม่ทำงานและกำหนด $I_{LL} = 0$ เวกเตอร์ **c** จะผ่านไปยังกระบวนการลำดับช่องสัญญาณย่อยถัดไป

ตารางที่ 2.11 ลำดับการแทรกสลับ $\Pi_L^{\max}(i)$ สำหรับการแทรกสลับ CRC
(เรียงค่า i จากซ้ายไปขวา บนลงล่าง)

| | | | | | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 2 | 4 | 7 | 9 | 14 | 19 | 20 | 24 | 25 | 26 | 28 | 31 |
| 34 | 42 | 45 | 49 | 50 | 51 | 53 | 54 | 56 | 58 | 59 | 61 | 62 |
| 65 | 66 | 67 | 69 | 70 | 71 | 72 | 76 | 77 | 81 | 82 | 83 | 87 |
| 88 | 89 | 91 | 93 | 95 | 98 | 101 | 104 | 106 | 108 | 110 | 111 | 113 |
| 115 | 118 | 119 | 120 | 122 | 123 | 126 | 127 | 129 | 132 | 134 | 138 | 139 |
| 140 | 1 | 3 | 5 | 8 | 10 | 15 | 21 | 27 | 29 | 32 | 35 | 43 |
| 46 | 52 | 55 | 57 | 60 | 63 | 68 | 73 | 78 | 84 | 90 | 92 | 94 |
| 96 | 99 | 102 | 105 | 107 | 109 | 112 | 114 | 116 | 121 | 124 | 128 | 130 |
| 133 | 135 | 141 | 6 | 11 | 16 | 22 | 30 | 33 | 36 | 44 | 47 | 64 |
| 74 | 79 | 85 | 97 | 100 | 103 | 117 | 125 | 131 | 136 | 142 | 12 | 17 |
| 23 | 37 | 48 | 75 | 80 | 86 | 137 | 143 | 13 | 18 | 38 | 144 | 39 |
| 145 | 40 | 146 | 41 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 |
| 156 | 157 | 158 | 159 | 160 | 161 | 162 | 163 | | | | | |

2.5.6 การลำดับช่องสัญญาณย่อย



รูปที่ 2.16 ตัวอย่างการลำดับช่องสัญญาณย่อยสำหรับช่องสัญญาณ BCH ที่เวกเตอร์ \mathbf{c}' ความยาว $K = 56$ และเวกเตอร์ \mathbf{u} ความยาว $N = 2^n = 512$ หรือ $n = n_{\max} = 9$ โดยการปรับอัตรารหัส $E = N$ สีน้ำเงิน สีแดงและสีเหลืองคือบิตซ้ำแข็ง บิตข้อมูลและบิต CRC ตามลำดับ

ระบบจะทำการสร้างรหัสโพลาร์ ซึ่งเป็นเวกเตอร์ \mathbf{u} ความยาว N โดยการเลือกตำแหน่งของบิตซ้ำแข็ง Q_F และบิตข้อมูล Q_I บิตข้อมูลหรือเวกเตอร์ \mathbf{c} (หรือ \mathbf{c}') จากกระบวนการก่อนถูกวางไว้สำหรับตำแหน่งบิตข้อมูล ส่วนตำแหน่งบิตซ้ำแข็งจะถูกกำหนดค่าเป็นบิต 0 การเลือกความยาว N จะมีเกณฑ์การเลือกตามสมการที่ 13 14 และ 15 โดยจะเลือกค่า n สุดท้ายเพื่อให้ได้ความยาว $N = 2^n$ การเลือกตำแหน่งของบิตซ้ำแข็งจะสอดคล้องกับเงื่อนไขการปรับอัตรารหัส โดยการปรับอัตรารหัสเป็นการปรับขนาดคำรหัสก่อนส่งผ่านช่องสัญญาณ โดยจะกำหนดตัวแปร U เป็นความยาวในการปรับอัตรารหัส รูปที่ 16 คือตัวอย่างการลำดับช่องสัญญาณย่อยของช่องสัญญาณ BCH ซึ่งสามารถมีความยาว A หรือ K ได้เพียงค่าเดียวตามตารางที่ 2.11

1) การเลือกตำแหน่งบิตซ้ำแข็งสำหรับทุกช่องสัญญาณ สรุปลงขั้นตอนได้ 2 ขั้นตอนดังนี้

1.1) เลือกตำแหน่งบิตซ้ำแข็งสำหรับการปรับอัตรารหัส หาก $K/E \leq 7/16$ จะปรับอัตรารหัสด้วยการฟังก์เจอร์ ซึ่งจะกำหนดบิตซ้ำแข็งที่ T ตำแหน่งแรก ในกรณีอื่นจะปรับอัตรารหัสด้วยการช็อตเทน ซึ่งจะกำหนดบิตซ้ำแข็งที่ $U = N - E$ ตำแหน่งสุดท้าย T คำนวณได้ดังสมการที่ 16

$$T = \begin{cases} \left\lceil \frac{3}{4}N - \frac{E}{2} \right\rceil - 1 & \text{หาก } E \geq \frac{3}{4}N \\ \left\lceil \frac{9}{16}N - \frac{E}{4} \right\rceil - 1 & \text{อื่น ๆ} \end{cases} \quad (16)$$

สาเหตุที่ใช้ความยาว T เพิ่มเติมจาก U เพื่อป้องกันไม่ให้อัตราข้อมูลถูกฟังก์เจอร์ไปในการปรับอัตรารหัส

1.2) หากจำนวนบิตซ้ำแข็งสำหรับการปรับอัตรารหัสยังมีขนาดไม่เกินจำนวนบิตซ้ำแข็งที่มีได้ จะทำการกำหนดตำแหน่งบิตซ้ำแข็งเพิ่มเติมตามจำนวนที่เหลือและกำหนดตามลำดับความน่าเชื่อถือของช่องสัญญาณดังตารางที่ 2.12 ซึ่งสามารถสรุปลงขั้นตอนการเลือกตำแหน่งบิตซ้ำแข็งตามลำดับความน่าเชื่อถือของช่องสัญญาณได้ 2 ขั้นตอนดังนี้

1.2.1) เลือกลำดับช่องสัญญาณ Q จากต่ำ (ค่าน้อย) ไปสูง (ค่ามาก) (ลำดับตามตารางไม่จำเป็นต้องติดกัน) ตามความยาวคำรหัสแม่ N ซึ่งจะได้ลำดับความน่าเชื่อถือ $W(Q)$ ที่คู่กับลำดับช่องสัญญาณ

1.2.2) ตำแหน่งของบิตซ้ำแข็งจะอยู่ในตำแหน่งของลำดับช่องสัญญาณที่คู่กับลำดับความน่าเชื่อถือที่ต่ำที่สุดตามจำนวนบิตซ้ำแข็ง $|Q_F|$ ที่ต้องการ

ตัวอย่างเช่น กรณี $N=8$ ประกอบด้วยบิตซ้ำและบิตข้อมูลเท่ากัน จะเลือกลำดับช่องสัญญาณมา 8 ลำดับที่ 0 1 2 4 3 5 6 7 (ตามคอลัมน์ลำดับช่องสัญญาณ) ซึ่งจะได้ลำดับความน่าเชื่อถือเป็น 0 1 2 3 7 8 11 24 (ตามคอลัมน์ลำดับความน่าเชื่อถือ) มาตามลำดับคู่กัน จะได้ตำแหน่งบิตซ้ำที่ 0 1 2 4 เนื่องจากที่ลำดับช่องสัญญาณ 0 1 2 4 คู่กับลำดับความน่าเชื่อถือที่ต่ำที่สุด (0 1 2 3) และได้ตำแหน่งบิตข้อมูลที่ 3 5 6 7

กรณีที่ $N=16$ ประกอบด้วยบิตซ้ำและบิตข้อมูลเท่ากัน จะเลือกลำดับช่องสัญญาณมา 16 ลำดับที่ 0 1 2 4 8 3 5 9 6 10 12 7 11 13 14 15 (ตามคอลัมน์ลำดับช่องสัญญาณ) ซึ่งจะได้ลำดับความน่าเชื่อถือเป็น 0 1 2 3 4 7 8 10 11 13 16 24 28 33 35 76 (ตามคอลัมน์ลำดับความน่าเชื่อถือ) มาตามลำดับคู่กัน จะได้ตำแหน่งบิตซ้ำที่ 0 1 2 4 8 3 5 9 เนื่องจากที่ลำดับช่องสัญญาณ 0 1 2 4 8 3 5 9 คู่กับลำดับความน่าเชื่อถือที่ต่ำที่สุด (0 1 2 3 4 7 8 10) และได้ตำแหน่งบิตข้อมูลที่ 6 10 12 7 11 13 14 15

2) การเลือกจำนวนช่องสัญญาณย่อย

กรณีช่องสัญญาณ UCI ที่เงื่อนไข $12 \leq A \leq 19$ จะมีการใช้งานบิตพาริตี ซึ่งตำแหน่งของบิตพาริตี Q_{PC} จะถูกแทรกไปในบิตข้อมูลจำนวน $n_{PC} = 3$ บิต หากรหัสมีเงื่อนไข $E - A \leq 175$ จะเลือกจากตำแหน่งที่ลำดับความน่าเชื่อถือต่ำที่สุดทั้ง 3 บิตจากตารางที่ 2.12 ภายในตำแหน่งของบิตข้อมูล แต่หากรหัสมีเงื่อนไข $E - A > 175$ จะเลือกจากตำแหน่งที่ลำดับความน่าเชื่อถือต่ำที่สุดเพียง 2 บิต โดยอีก 1 บิตพาริตีจะเลือกตำแหน่งจากตำแหน่งที่มีน้ำหนักแถวของเมทริกซ์ G_N ต่ำที่สุด ซึ่งหากมีตำแหน่งดังกล่าวมากกว่าหนึ่งตำแหน่ง จะเลือกตำแหน่งดังกล่าวที่ความน่าเชื่อถือสูงที่สุดด้วย ทำให้จำนวนของบิตซ้ำจะลดลงเหลือ $|Q_F| = N - (K + 3)$

สำหรับช่องสัญญาณ UCI ที่เงื่อนไข $A \geq 20$ ช่องสัญญาณ BCH และ DCI จะไม่ใช้งานบิตพาริตี ซึ่งจะเลือกบิตซ้ำตามจำนวน $|Q_F| = N - K$

ตารางที่ 2.12 ลำดับความน่าเชื่อถือของช่องสัญญาณ (polar sequence)

โดยที่ $W(Q)$ คือลำดับความน่าเชื่อถือและ Q คือลำดับช่องสัญญาณ

| $w(Q)$ | Q | $w(Q)$ | Q | $w(Q)$ | Q | $w(Q)$ | Q | $w(Q)$ | Q | $w(Q)$ | Q | $w(Q)$ | Q | $w(Q)$ | Q |
|--------|-----|--------|-----|--------|-----|--------|-----|--------|-----|--------|-----|--------|-----|--------|-----|
| 0 | 0 | 128 | 518 | 256 | 94 | 384 | 214 | 512 | 364 | 640 | 414 | 768 | 819 | 896 | 966 |
| 1 | 1 | 129 | 54 | 257 | 204 | 385 | 309 | 513 | 654 | 641 | 223 | 769 | 814 | 897 | 755 |
| 2 | 2 | 130 | 83 | 258 | 298 | 386 | 188 | 514 | 659 | 642 | 663 | 770 | 439 | 898 | 859 |
| 3 | 4 | 131 | 57 | 259 | 400 | 387 | 449 | 515 | 335 | 643 | 692 | 771 | 929 | 899 | 940 |
| 4 | 8 | 132 | 521 | 260 | 608 | 388 | 217 | 516 | 480 | 644 | 835 | 772 | 490 | 900 | 830 |

| $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ |
|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|
| 5 | 16 | 133 | 112 | 261 | 352 | 389 | 408 | 517 | 315 | 645 | 619 | 773 | 623 | 901 | 911 |
| 6 | 32 | 134 | 135 | 262 | 325 | 390 | 609 | 518 | 221 | 646 | 472 | 774 | 671 | 902 | 871 |
| 7 | 3 | 135 | 78 | 263 | 533 | 391 | 596 | 519 | 370 | 647 | 455 | 775 | 739 | 903 | 639 |
| 8 | 5 | 136 | 289 | 264 | 155 | 392 | 551 | 520 | 613 | 648 | 796 | 776 | 916 | 904 | 888 |
| 9 | 64 | 137 | 194 | 265 | 210 | 393 | 650 | 521 | 422 | 649 | 809 | 777 | 463 | 905 | 479 |
| 10 | 9 | 138 | 85 | 266 | 305 | 394 | 229 | 522 | 425 | 650 | 714 | 778 | 843 | 906 | 946 |
| 11 | 6 | 139 | 276 | 267 | 547 | 395 | 159 | 523 | 451 | 651 | 721 | 779 | 381 | 907 | 750 |
| 12 | 17 | 140 | 522 | 268 | 300 | 396 | 420 | 524 | 614 | 652 | 837 | 780 | 497 | 908 | 969 |
| 13 | 10 | 141 | 58 | 269 | 109 | 397 | 310 | 525 | 543 | 653 | 716 | 781 | 930 | 909 | 508 |
| 14 | 18 | 142 | 168 | 270 | 184 | 398 | 541 | 526 | 235 | 654 | 864 | 782 | 821 | 910 | 861 |
| 15 | 128 | 143 | 139 | 271 | 534 | 399 | 773 | 527 | 412 | 655 | 810 | 783 | 726 | 911 | 757 |
| 16 | 12 | 144 | 99 | 272 | 537 | 400 | 610 | 528 | 343 | 656 | 606 | 784 | 961 | 912 | 970 |
| 17 | 33 | 145 | 86 | 273 | 115 | 401 | 657 | 529 | 372 | 657 | 912 | 785 | 872 | 913 | 919 |
| 18 | 65 | 146 | 60 | 274 | 167 | 402 | 333 | 530 | 775 | 658 | 722 | 786 | 492 | 914 | 875 |
| 19 | 20 | 147 | 280 | 275 | 225 | 403 | 119 | 531 | 317 | 659 | 696 | 787 | 631 | 915 | 862 |
| 20 | 256 | 148 | 89 | 276 | 326 | 404 | 600 | 532 | 222 | 660 | 377 | 788 | 729 | 916 | 758 |
| 21 | 34 | 149 | 290 | 277 | 306 | 405 | 339 | 533 | 426 | 661 | 435 | 789 | 700 | 917 | 948 |
| 22 | 24 | 150 | 529 | 278 | 772 | 406 | 218 | 534 | 453 | 662 | 817 | 790 | 443 | 918 | 977 |
| 23 | 36 | 151 | 524 | 279 | 157 | 407 | 368 | 535 | 237 | 663 | 319 | 791 | 741 | 919 | 923 |
| 24 | 7 | 152 | 196 | 280 | 656 | 408 | 652 | 536 | 559 | 664 | 621 | 792 | 845 | 920 | 972 |
| 25 | 129 | 153 | 141 | 281 | 329 | 409 | 230 | 537 | 833 | 665 | 812 | 793 | 920 | 921 | 761 |
| 26 | 66 | 154 | 101 | 282 | 110 | 410 | 391 | 538 | 804 | 666 | 484 | 794 | 382 | 922 | 877 |
| 27 | 512 | 155 | 147 | 283 | 117 | 411 | 313 | 539 | 712 | 667 | 430 | 795 | 822 | 923 | 952 |
| 28 | 11 | 156 | 176 | 284 | 212 | 412 | 450 | 540 | 834 | 668 | 838 | 796 | 851 | 924 | 495 |
| 29 | 40 | 157 | 142 | 285 | 171 | 413 | 542 | 541 | 661 | 669 | 667 | 797 | 730 | 925 | 703 |
| 30 | 68 | 158 | 530 | 286 | 776 | 414 | 334 | 542 | 808 | 670 | 488 | 798 | 498 | 926 | 935 |
| 31 | 130 | 159 | 321 | 287 | 330 | 415 | 233 | 543 | 779 | 671 | 239 | 799 | 880 | 927 | 978 |
| 32 | 19 | 160 | 31 | 288 | 226 | 416 | 555 | 544 | 617 | 672 | 378 | 800 | 742 | 928 | 883 |
| 33 | 13 | 161 | 200 | 289 | 549 | 417 | 774 | 545 | 604 | 673 | 459 | 801 | 445 | 929 | 762 |
| 34 | 48 | 162 | 90 | 290 | 538 | 418 | 175 | 546 | 433 | 674 | 622 | 802 | 471 | 930 | 503 |
| 35 | 14 | 163 | 545 | 291 | 387 | 419 | 123 | 547 | 720 | 675 | 627 | 803 | 635 | 931 | 925 |
| 36 | 72 | 164 | 292 | 292 | 308 | 420 | 658 | 548 | 816 | 676 | 437 | 804 | 932 | 932 | 878 |

| $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ |
|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|
| 37 | 257 | 165 | 322 | 293 | 216 | 421 | 612 | 549 | 836 | 677 | 380 | 805 | 687 | 933 | 735 |
| 38 | 21 | 166 | 532 | 294 | 416 | 422 | 341 | 550 | 347 | 678 | 818 | 806 | 903 | 934 | 993 |
| 39 | 132 | 167 | 263 | 295 | 271 | 423 | 777 | 551 | 897 | 679 | 461 | 807 | 825 | 935 | 885 |
| 40 | 35 | 168 | 149 | 296 | 279 | 424 | 220 | 552 | 243 | 680 | 496 | 808 | 500 | 936 | 939 |
| 41 | 258 | 169 | 102 | 297 | 158 | 425 | 314 | 553 | 662 | 681 | 669 | 809 | 846 | 937 | 994 |
| 42 | 26 | 170 | 105 | 298 | 337 | 426 | 424 | 554 | 454 | 682 | 679 | 810 | 745 | 938 | 980 |
| 43 | 513 | 171 | 304 | 299 | 550 | 427 | 395 | 555 | 318 | 683 | 724 | 811 | 826 | 939 | 926 |
| 44 | 80 | 172 | 296 | 300 | 672 | 428 | 673 | 556 | 675 | 684 | 841 | 812 | 732 | 940 | 764 |
| 45 | 37 | 173 | 163 | 301 | 118 | 429 | 583 | 557 | 618 | 685 | 629 | 813 | 446 | 941 | 941 |
| 46 | 25 | 174 | 92 | 302 | 332 | 430 | 355 | 558 | 898 | 686 | 351 | 814 | 962 | 942 | 967 |
| 47 | 22 | 175 | 47 | 303 | 579 | 431 | 287 | 559 | 781 | 687 | 467 | 815 | 936 | 943 | 886 |
| 48 | 136 | 176 | 267 | 304 | 540 | 432 | 183 | 560 | 376 | 688 | 438 | 816 | 475 | 944 | 831 |
| 49 | 260 | 177 | 385 | 305 | 389 | 433 | 234 | 561 | 428 | 689 | 737 | 817 | 853 | 945 | 947 |
| 50 | 264 | 178 | 546 | 306 | 173 | 434 | 125 | 562 | 665 | 690 | 251 | 818 | 867 | 946 | 507 |
| 51 | 38 | 179 | 324 | 307 | 121 | 435 | 557 | 563 | 736 | 691 | 462 | 819 | 637 | 947 | 889 |
| 52 | 514 | 180 | 208 | 308 | 553 | 436 | 660 | 564 | 567 | 692 | 442 | 820 | 907 | 948 | 984 |
| 53 | 96 | 181 | 386 | 309 | 199 | 437 | 616 | 565 | 840 | 693 | 441 | 821 | 487 | 949 | 751 |
| 54 | 67 | 182 | 150 | 310 | 784 | 438 | 342 | 566 | 625 | 694 | 469 | 822 | 695 | 950 | 942 |
| 55 | 41 | 183 | 153 | 311 | 179 | 439 | 316 | 567 | 238 | 695 | 247 | 823 | 746 | 951 | 996 |
| 56 | 144 | 184 | 165 | 312 | 228 | 440 | 241 | 568 | 359 | 696 | 683 | 824 | 828 | 952 | 971 |
| 57 | 28 | 185 | 106 | 313 | 338 | 441 | 778 | 569 | 457 | 697 | 842 | 825 | 753 | 953 | 890 |
| 58 | 69 | 186 | 55 | 314 | 312 | 442 | 563 | 570 | 399 | 698 | 738 | 826 | 854 | 954 | 509 |
| 59 | 42 | 187 | 328 | 315 | 704 | 443 | 345 | 571 | 787 | 699 | 899 | 827 | 857 | 955 | 949 |
| 60 | 516 | 188 | 536 | 316 | 390 | 444 | 452 | 572 | 591 | 700 | 670 | 828 | 504 | 956 | 973 |
| 61 | 49 | 189 | 577 | 317 | 174 | 445 | 397 | 573 | 678 | 701 | 783 | 829 | 799 | 957 | 1000 |
| 62 | 74 | 190 | 548 | 318 | 554 | 446 | 403 | 574 | 434 | 702 | 849 | 830 | 255 | 958 | 892 |
| 63 | 272 | 191 | 113 | 319 | 581 | 447 | 207 | 575 | 677 | 703 | 820 | 831 | 964 | 959 | 950 |
| 64 | 160 | 192 | 154 | 320 | 393 | 448 | 674 | 576 | 349 | 704 | 728 | 832 | 909 | 960 | 863 |
| 65 | 520 | 193 | 79 | 321 | 283 | 449 | 558 | 577 | 245 | 705 | 928 | 833 | 719 | 961 | 759 |
| 66 | 288 | 194 | 269 | 322 | 122 | 450 | 785 | 578 | 458 | 706 | 791 | 834 | 477 | 962 | 1008 |
| 67 | 528 | 195 | 108 | 323 | 448 | 451 | 432 | 579 | 666 | 707 | 367 | 835 | 915 | 963 | 510 |
| 68 | 192 | 196 | 578 | 324 | 353 | 452 | 357 | 580 | 620 | 708 | 901 | 836 | 638 | 964 | 979 |

| $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ |
|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|
| 69 | 544 | 197 | 224 | 325 | 561 | 453 | 187 | 581 | 363 | 709 | 630 | 837 | 748 | 965 | 953 |
| 70 | 70 | 198 | 166 | 326 | 203 | 454 | 236 | 582 | 127 | 710 | 685 | 838 | 944 | 966 | 763 |
| 71 | 44 | 199 | 519 | 327 | 63 | 455 | 664 | 583 | 191 | 711 | 844 | 839 | 869 | 967 | 974 |
| 72 | 131 | 200 | 552 | 328 | 340 | 456 | 624 | 584 | 782 | 712 | 633 | 840 | 491 | 968 | 954 |
| 73 | 81 | 201 | 195 | 329 | 394 | 457 | 587 | 585 | 407 | 713 | 711 | 841 | 699 | 969 | 879 |
| 74 | 50 | 202 | 270 | 330 | 527 | 458 | 780 | 586 | 436 | 714 | 253 | 842 | 754 | 970 | 981 |
| 75 | 73 | 203 | 641 | 331 | 582 | 459 | 705 | 587 | 626 | 715 | 691 | 843 | 858 | 971 | 982 |
| 76 | 15 | 204 | 523 | 332 | 556 | 460 | 126 | 588 | 571 | 716 | 824 | 844 | 478 | 972 | 927 |
| 77 | 320 | 205 | 275 | 333 | 181 | 461 | 242 | 589 | 465 | 717 | 902 | 845 | 968 | 973 | 995 |
| 78 | 133 | 206 | 580 | 334 | 295 | 462 | 565 | 590 | 681 | 718 | 686 | 846 | 383 | 974 | 765 |
| 79 | 52 | 207 | 291 | 335 | 285 | 463 | 398 | 591 | 246 | 719 | 740 | 847 | 910 | 975 | 956 |
| 80 | 23 | 208 | 59 | 336 | 232 | 464 | 346 | 592 | 707 | 720 | 850 | 848 | 815 | 976 | 887 |
| 81 | 134 | 209 | 169 | 337 | 124 | 465 | 456 | 593 | 350 | 721 | 375 | 849 | 976 | 977 | 985 |
| 82 | 384 | 210 | 560 | 338 | 205 | 466 | 358 | 594 | 599 | 722 | 444 | 850 | 870 | 978 | 997 |
| 83 | 76 | 211 | 114 | 339 | 182 | 467 | 405 | 595 | 668 | 723 | 470 | 851 | 917 | 979 | 986 |
| 84 | 137 | 212 | 277 | 340 | 643 | 468 | 303 | 596 | 790 | 724 | 483 | 852 | 727 | 980 | 943 |
| 85 | 82 | 213 | 156 | 341 | 562 | 469 | 569 | 597 | 460 | 725 | 415 | 853 | 493 | 981 | 891 |
| 86 | 56 | 214 | 87 | 342 | 286 | 470 | 244 | 598 | 249 | 726 | 485 | 854 | 873 | 982 | 998 |
| 87 | 27 | 215 | 197 | 343 | 585 | 471 | 595 | 599 | 682 | 727 | 905 | 855 | 701 | 983 | 766 |
| 88 | 97 | 216 | 116 | 344 | 299 | 472 | 189 | 600 | 573 | 728 | 795 | 856 | 931 | 984 | 511 |
| 89 | 39 | 217 | 170 | 345 | 354 | 473 | 566 | 601 | 411 | 729 | 473 | 857 | 756 | 985 | 988 |
| 90 | 259 | 218 | 61 | 346 | 211 | 474 | 676 | 602 | 803 | 730 | 634 | 858 | 860 | 986 | 1001 |
| 91 | 84 | 219 | 531 | 347 | 401 | 475 | 361 | 603 | 789 | 731 | 744 | 859 | 499 | 987 | 951 |
| 92 | 138 | 220 | 525 | 348 | 185 | 476 | 706 | 604 | 709 | 732 | 852 | 860 | 731 | 988 | 1002 |
| 93 | 145 | 221 | 642 | 349 | 396 | 477 | 589 | 605 | 365 | 733 | 960 | 861 | 823 | 989 | 893 |
| 94 | 261 | 222 | 281 | 350 | 344 | 478 | 215 | 606 | 440 | 734 | 865 | 862 | 922 | 990 | 975 |
| 95 | 29 | 223 | 278 | 351 | 586 | 479 | 786 | 607 | 628 | 735 | 693 | 863 | 874 | 991 | 894 |
| 96 | 43 | 224 | 526 | 352 | 645 | 480 | 647 | 608 | 689 | 736 | 797 | 864 | 918 | 992 | 1009 |
| 97 | 98 | 225 | 177 | 353 | 593 | 481 | 348 | 609 | 374 | 737 | 906 | 865 | 502 | 993 | 955 |
| 98 | 515 | 226 | 293 | 354 | 535 | 482 | 419 | 610 | 423 | 738 | 715 | 866 | 933 | 994 | 1004 |
| 99 | 88 | 227 | 388 | 355 | 240 | 483 | 406 | 611 | 466 | 739 | 807 | 867 | 743 | 995 | 1010 |
| 100 | 140 | 228 | 91 | 356 | 206 | 484 | 464 | 612 | 793 | 740 | 474 | 868 | 760 | 996 | 957 |

| $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ | $w(\varrho)$ | ϱ |
|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|--------------|-----------|
| 101 | 30 | 229 | 584 | 357 | 95 | 485 | 680 | 613 | 250 | 741 | 636 | 869 | 881 | 997 | 983 |
| 102 | 146 | 230 | 769 | 358 | 327 | 486 | 801 | 614 | 371 | 742 | 694 | 870 | 494 | 998 | 958 |
| 103 | 71 | 231 | 198 | 359 | 564 | 487 | 362 | 615 | 481 | 743 | 254 | 871 | 702 | 999 | 987 |
| 104 | 262 | 232 | 172 | 360 | 800 | 488 | 590 | 616 | 574 | 744 | 717 | 872 | 921 | 1000 | 1012 |
| 105 | 265 | 233 | 120 | 361 | 402 | 489 | 409 | 617 | 413 | 745 | 575 | 873 | 501 | 1001 | 999 |
| 106 | 161 | 234 | 201 | 362 | 356 | 490 | 570 | 618 | 603 | 746 | 913 | 874 | 876 | 1002 | 1016 |
| 107 | 576 | 235 | 336 | 363 | 307 | 491 | 788 | 619 | 366 | 747 | 798 | 875 | 847 | 1003 | 767 |
| 108 | 45 | 236 | 62 | 364 | 301 | 492 | 597 | 620 | 468 | 748 | 811 | 876 | 992 | 1004 | 989 |
| 109 | 100 | 237 | 282 | 365 | 417 | 493 | 572 | 621 | 655 | 749 | 379 | 877 | 447 | 1005 | 1003 |
| 110 | 640 | 238 | 143 | 366 | 213 | 494 | 219 | 622 | 900 | 750 | 697 | 878 | 733 | 1006 | 990 |
| 111 | 51 | 239 | 103 | 367 | 568 | 495 | 311 | 623 | 805 | 751 | 431 | 879 | 827 | 1007 | 1005 |
| 112 | 148 | 240 | 178 | 368 | 832 | 496 | 708 | 624 | 615 | 752 | 607 | 880 | 934 | 1008 | 959 |
| 113 | 46 | 241 | 294 | 369 | 588 | 497 | 598 | 625 | 684 | 753 | 489 | 881 | 882 | 1009 | 1011 |
| 114 | 75 | 242 | 93 | 370 | 186 | 498 | 601 | 626 | 710 | 754 | 866 | 882 | 937 | 1010 | 1013 |
| 115 | 266 | 243 | 644 | 371 | 646 | 499 | 651 | 627 | 429 | 755 | 723 | 883 | 963 | 1011 | 895 |
| 116 | 273 | 244 | 202 | 372 | 404 | 500 | 421 | 628 | 794 | 756 | 486 | 884 | 747 | 1012 | 1006 |
| 117 | 517 | 245 | 592 | 373 | 227 | 501 | 792 | 629 | 252 | 757 | 908 | 885 | 505 | 1013 | 1014 |
| 118 | 104 | 246 | 323 | 374 | 896 | 502 | 802 | 630 | 373 | 758 | 718 | 886 | 855 | 1014 | 1017 |
| 119 | 162 | 247 | 392 | 375 | 594 | 503 | 611 | 631 | 605 | 759 | 813 | 887 | 924 | 1015 | 1018 |
| 120 | 53 | 248 | 297 | 376 | 418 | 504 | 602 | 632 | 848 | 760 | 476 | 888 | 734 | 1016 | 991 |
| 121 | 193 | 249 | 770 | 377 | 302 | 505 | 410 | 633 | 690 | 761 | 856 | 889 | 829 | 1017 | 1020 |
| 122 | 152 | 250 | 107 | 378 | 649 | 506 | 231 | 634 | 713 | 762 | 839 | 890 | 965 | 1018 | 1007 |
| 123 | 77 | 251 | 180 | 379 | 771 | 507 | 688 | 635 | 632 | 763 | 725 | 891 | 938 | 1019 | 1015 |
| 124 | 164 | 252 | 151 | 380 | 360 | 508 | 653 | 636 | 482 | 764 | 698 | 892 | 884 | 1020 | 1019 |
| 125 | 768 | 253 | 209 | 381 | 539 | 509 | 248 | 637 | 806 | 765 | 914 | 893 | 506 | 1021 | 1021 |
| 126 | 268 | 254 | 284 | 382 | 111 | 510 | 369 | 638 | 427 | 766 | 752 | 894 | 749 | 1022 | 1022 |
| 127 | 274 | 255 | 648 | 383 | 331 | 511 | 190 | 639 | 904 | 767 | 868 | 895 | 945 | 1023 | 1023 |

2.5.7 การคำนวณบิตพาริตี

กรณีช่องสัญญาณ UCI

หากช่องสัญญาณ UCI มีการใช้งานบิตพาริตี หลังจากการเลือกตำแหน่งของบิตพาริตีทั้ง 3 บิตแล้ว การคำนวณบิตพาริตีสามารถทำได้โดยซีพรีจิสเตอร์วนซ้ำขนาด 5 บิต โดยตั้งค่าบิตเริ่มต้นเป็น 0 บิตพาริตีจะคำนวณโดยการเอ็กซ์คลูซีฟออร์กับบิตลำดับหน้าเว้นครั้งละ 5 โดยจะเอ็กซ์คลูซีฟออร์เฉพาะบิตข้อมูลที่ไม่รวมบิตพาริตีลำดับก่อนและไม่รวมบิตซ้ำซ้ำ สามารถสรุปการคำนวณบิตพาริตี u_i ได้ดังสมการที่ 17

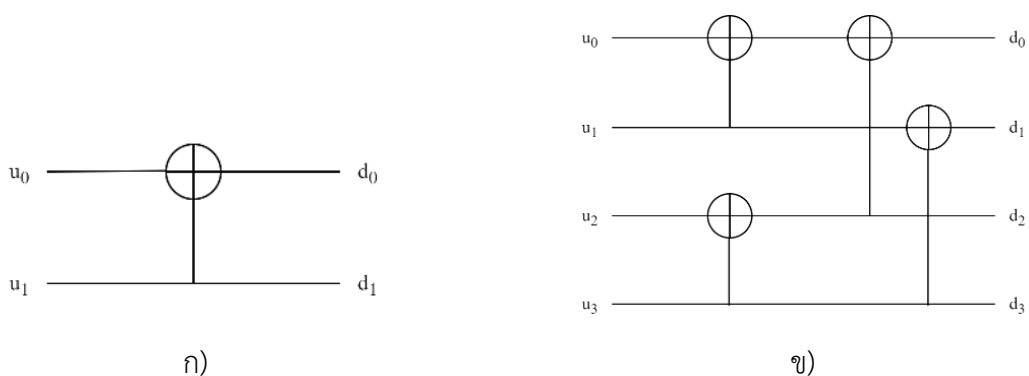
$$u_i = \bigoplus_{j=\lfloor i_{PC}/5 \rfloor}^{q-1} u_{5j+p} \quad (17)$$

โดยที่ $q = \lfloor i/5 \rfloor$ $p = \text{mod}(i, 5)$ และ $i_{PC} \in Q_{PC}$ คือค่าตำแหน่งบิตพาริตีสูงสุดที่น้อยกว่า i ที่ $\text{mod}(i_{PC}, 5) = p$ หากไม่มีค่าตำแหน่งพาริตีนั้นจะกำหนดให้ $i_{PC} = 0$

จากนั้นเวกเตอร์ \mathbf{u} จะถูกส่งผ่านไปเข้ารหัสโพลาร์ในกระบวนการถัดไป

สำหรับช่องสัญญาณ BCH และ DCI กระบวนการนี้จะไม่ทำงาน เวกเตอร์ \mathbf{u} จากกระบวนการก่อนหน้าจะถูกส่งผ่านไปเข้ารหัสโพลาร์ในกระบวนการถัดไป

2.5.8 การเข้ารหัสโพลาร์



รูปที่ 2.17 โครงสร้างการเข้ารหัสโพลาร์ที่ความยาว $N = 2$ และ $N = 4$ ตามลำดับ

การเข้ารหัสโพลาร์สำหรับทุกช่องสัญญาณสามารถดำเนินการทางคณิตศาสตร์โดยสมการที่

18

$$\mathbf{d} = \mathbf{uG}_N \quad (18)$$

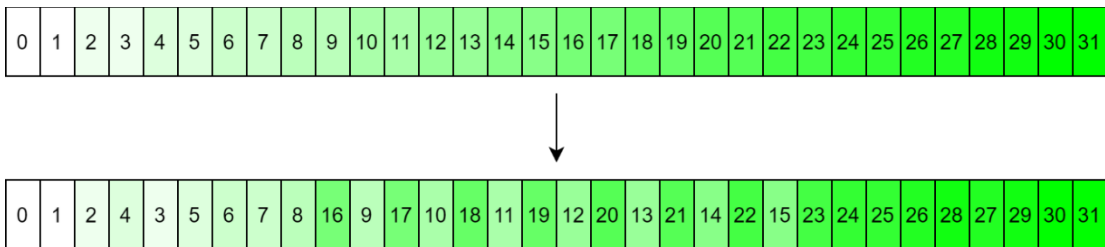
โดยที่ $\mathbf{G}_N = \mathbf{G}_2^{\otimes n}$ และ $\mathbf{G}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ มีอินพุตคือบิตที่จะเข้ารหัส \mathbf{u} ความยาว N และได้ผลลัพธ์เป็นคำรหัสแม่ \mathbf{d} ความยาว N ซึ่งในทางปฏิบัติสามารถดำเนินการได้โดยตัวดำเนินการเอ็กซ์คลูซีฟออร์ตามโครงสร้างดังรูปที่ 2.17 ในกรณีที่ $N=2$ สามารถเขียนสมการโดยสมการที่ 19 และ 20

$$d_0 = u_0 \oplus u_1 \quad (19)$$

$$d_1 = u_1 \quad (20)$$

โครงสร้างการเข้ารหัสสามารถขยายได้ในลักษณะเรียกซ้ำ (recursive) จำนวนสองเท่าได้เรื่อย ๆ ซึ่งเป็นสาเหตุที่ความยาวคำรหัสแม่ต้องมีขนาด $N=2^n$

2.5.9 การแทรกสลับบล็อกย่อย



รูปที่ 2.18 ลำดับการแทรกสลับบล็อกย่อย

เวกเตอร์ \mathbf{d} ความยาว N จากกระบวนการก่อนหน้าสำหรับทุกช่องสัญญาณจะถูกแบ่งเป็น 32 บล็อกย่อย ความยาวบล็อกละ $N/32$ แทนลำดับบิตในการแทรกสลับโดย $J(j)$ ผลลัพธ์จะได้เวกเตอร์ \mathbf{y} ความยาว N โดยที่ $y_j = d_{J(j)}$ และ $j=0,1,\dots,N-1$ สามารถคำนวณลำดับบิตในการแทรกสลับดังสมการที่ 21

$$J(j) = \left(P \left(\left\lfloor 32 \frac{j}{N} \right\rfloor \right) \cdot \frac{N}{32} \right) + \text{mod} \left(j, \frac{N}{32} \right) \quad (21)$$

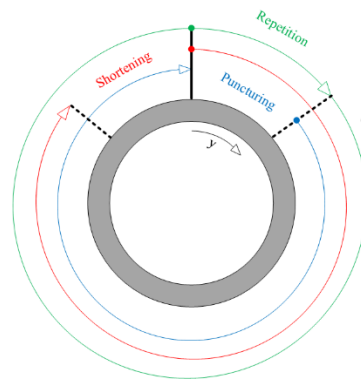
โดยที่ลำดับการแทรกสลับ $P(i)$ แสดงดังตารางที่ 2.13 และแสดงลำดับการแทรกสลับดังรูปที่ 18

ตารางที่ 2.13 ลำดับการแทรกสลับ $P(i)$ สำหรับการแทรกสลับบล็อกย่อย

(เรียงค่า i จากซ้ายไปขวา บนลงล่าง)

| | | | | | | | | | | | | | | | |
|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 0 | 1 | 2 | 4 | 3 | 5 | 6 | 7 | 8 | 16 | 9 | 17 | 10 | 18 | 11 | 19 |
| 12 | 20 | 13 | 21 | 14 | 22 | 15 | 23 | 24 | 25 | 26 | 28 | 27 | 29 | 30 | 31 |

2.5.10 การปรับอัตราหัส



รูปที่ 2.19 บัฟเฟอร์วงกลมสำหรับการปรับอัตราหัส โดยให้วงกลมสีเทาเป็นความยาวของเวกเตอร์ y และลูกศรสีต่าง ๆ เป็นเวกเตอร์ e ที่จะเลือกบิตในเวกเตอร์ y ส่งไปยังกระบวนการถัดไป

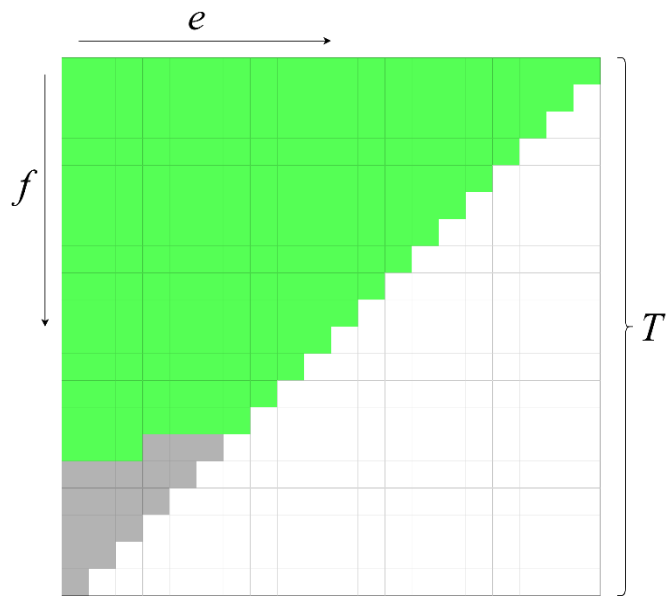
การปรับอัตราหัสจะถูกกำหนดโดยค่ารหัส e ที่ความยาว E ซึ่งเป็นความยาวที่ถูกกำหนดจากการสื่อสารขั้นถัดไป เนื่องจากกระบวนการลำดับช่องสัญญาณย่อยได้ลำดับบิตแช่แข็งในตำแหน่งที่จะทำการตัดบิตออกแล้ว กระบวนการนี้จะทำการตัดบิตจากเวกเตอร์ y ออก ในกรณีที่ใช้การปรับอัตราหัสแบบการพังก์เจอร์หรือการชอร์ตเทน หรือส่งบิตเวกเตอร์ y เข้าในกรณีที่ใช้การส่งซ้ำ รายละเอียดของการปรับอัตราหัสทั้ง 3 รูปแบบสรุปได้ดังรูปที่ 2.19 และมีรายละเอียดดังนี้

1) การพังก์เจอร์ (puncturing) จะทำเมื่อ $E < N$ และ $K/E \leq 7/16$ โดยจะไม่ทำการส่งบิต $U = N - E$ แรก จะได้ผลลัพธ์เวกเตอร์ e ที่ความยาว E โดยที่ $e_i = y_{i+U}$ และ $i = 0, 1, \dots, E-1$

2) การชอร์ตเทน (shortening) จะทำเมื่อ $E < N$ และ $K/E > 7/16$ โดยจะไม่ทำการส่งบิต $U = N - E$ สุดท้าย จะได้ผลลัพธ์เวกเตอร์ e ที่ความยาว E โดยที่ $e_i = y_i$ และ $i = 0, 1, \dots, E-1$

3) การส่งซ้ำ (repetition) จะทำเมื่อ $E > N$ โดยจะส่งบิต $U = N - E$ แรกซ้ำ จะได้ผลลัพธ์เวกเตอร์ e ที่ความยาว E โดยที่ $e_i = y_{\text{mod}(i, N)}$ และ $i = 0, 1, \dots, E-1$

2.5.11 การแทรกสลับบิตรหัส



รูปที่ 2.20 รูปแบบการแทรกสลับแบบสามเหลี่ยมขั้นบันได

ก่อนการนำคำรหัสที่ผ่านการปรับอัตราหัสส่งไปยังการสื่อสารขั้นถัดไปเพื่อทำการมอดูเลชันเวกเตอร์ \mathbf{e} จะถูกแทรกสลับอีกครั้งโดยการแทรกสลับแบบสามเหลี่ยมขั้นบันได การแทรกสลับนี้ถือมีความสำคัญเนื่องจากสามารถช่วยเพิ่มการขยายไดเวอร์ซิตี (diversity gain) ส่งผลให้มีอัตราบิตผิดพลาดที่ต่ำขึ้นในการมอดูเลชันลำดับสูง

กรณีช่องสัญญาณ UCI

การแทรกสลับจะทำงานและให้ $I_{BL} = 1$ รูปแบบการแทรกสลับจะสร้างจากโครงสร้างสามเหลี่ยมขั้นบันไดที่มีความกว้าง-ยาวขนาด $T \times T$ ดังรูปที่ 20 T จะมีความยาวเท่ากับจำนวนเต็มค่าน้อยสุดที่ $T(T+1)/2 \geq E$ เวกเตอร์ \mathbf{e} แต่ละบิตจะถูกเขียนเข้าโครงสร้างสามเหลี่ยมแทนด้วยเมทริกซ์สามเหลี่ยมคว่ำ \mathbf{v} ขนาด $T \times T$ ดังรูปที่ 2.20 จากทิศทางซ้ายไปขวาและตามด้วยบนลงล่าง จากนั้นจะสร้างผลลัพธ์คำรหัสที่ผ่านการแทรกสลับบิตรหัส \mathbf{f} โดยการอ่านค่าจากโครงสร้างสามเหลี่ยมจากทิศทางบนลงล่างและตามด้วยซ้ายไปขวา โดยมองในรายละเอียดเพิ่มเติม ส่วนที่ไว้ของโครงสร้างสามเหลี่ยมนั้นเป็นการแทนค่าในเมทริกซ์ \mathbf{v} ด้วย null จากการแทนด้วยค่าของเวกเตอร์ \mathbf{e} สามารถเขียนได้ดังสมการที่ 22

$$v_{i,j} = \begin{cases} \text{null} & \text{หาก } i+j \geq T \text{ หรือ} \\ e_{j+iT-i((i+1)/2)} & j+iT-i((i+1)/2) \geq E \text{ อื่น ๆ} \end{cases} \quad (22)$$

โดยที่ $i=0,1,\dots,T$ และ $j=0,1,\dots,T$

สำหรับช่องสัญญาณ BCH และ DCI กระบวนการนี้จะไม่ทำงานและกำหนด $I_{BIL} = 0$ เวกเตอร์ \mathbf{e} จากกระบวนการปรับอัตราหัสจะถูกบรรจุเข้าเพย์โหลดขาออก \mathbf{g} สำหรับการสื่อสารชั้นถัดไป

2.5.12 การต่อบล็อกรหัส

กรณีช่องสัญญาณ UCI หากกระบวนการข้างต้นมีการแบ่งบล็อกรหัส หรือมีเงื่อนไขว่า $(A \geq 360 \wedge E \geq 1088) \vee A \geq 1013$ และ $I_{seg} = 1$ เวกเตอร์ \mathbf{f} สองเวกเตอร์จะถูกต่อกันตามลำดับเดิมได้ผลลัพธ์เวกเตอร์ \mathbf{g} ความยาว G หาก $G = 2E + 1$ จะเติมบิต 0 ไว้ด้านหลังเวกเตอร์ที่สอง

สำหรับช่องสัญญาณ BCH และ DCI กระบวนการนี้จะไม่ทำงานและมีตัวบ่งชี้ $I_{BIL} = 0$ อยู่แล้ว เวกเตอร์ \mathbf{e} จากกระบวนการปรับอัตราหัสจะถูกบรรจุเข้าเพย์โหลดขาออก \mathbf{g} สำหรับการสื่อสารชั้นถัดไป

บทที่ 3 ระเบียบวิธีวิจัย

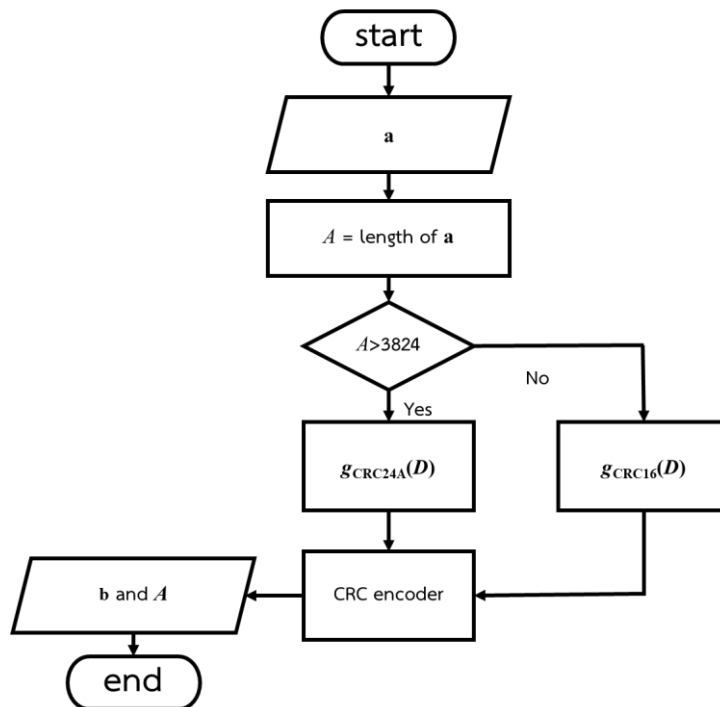
3.1 รายละเอียดการออกแบบชุดซอฟต์แวร์และวงจรเข้ารหัสช่องสัญญาณชนิด LDPC codes

3.1.1 ส่วนประกอบชุดซอฟต์แวร์และวงจร

กระบวนการเข้ารหัสช่องสัญญาณชนิด LDPC codes ประกอบด้วยโมดูลที่สำคัญทั้งหมด 7 ส่วน ได้แก่ 1. การเข้ารหัส CRC 2. การแบ่งข้อมูลเป็นบล็อกย่อย 3. การเข้ารหัส CRC ให้บล็อกข้อมูลย่อย 4. การเข้ารหัส LDPC ให้บล็อกข้อมูลย่อย 5. การปรับอัตรารหัสโดยการเลือกบิตที่จะส่ง 6. การแทรกสลับบิตข้อมูลของบล็อกข้อมูลย่อย และ 7. การนำบล็อกข้อมูลย่อยมาต่อเรียงและส่งออก ภายในโมดูลหลักจะมีโมดูลย่อยที่ทำหน้าที่ประมวลผลค่าพารามิเตอร์ที่สำคัญอยู่ภายใน

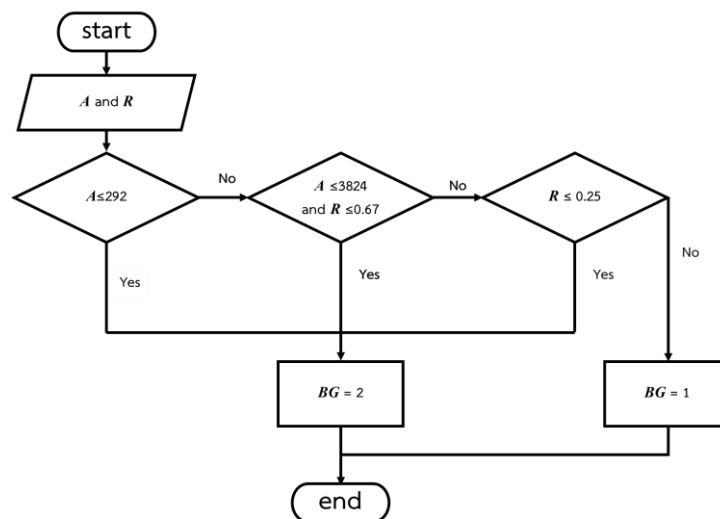
3.1.2 แผนผังการทำงานของชุดซอฟต์แวร์และวงจรเข้ารหัสช่องสัญญาณ

3.1.2.1 การเข้ารหัส CRC



รูปที่ 3.1 แผนภาพบล็อกการเข้ารหัส CRC

ในส่วนของโมดูลหลักการเข้ารหัส CRC ประกอบด้วยโมดูลย่อย 2 โมดูลได้แก่ การเข้ารหัส CRC และการเลือกกราฟฐาน (Base graph) รูปที่ 3.1 แสดงแผนภาพการทำงานของซอฟต์แวร์ เริ่มต้นจากรับบิตข้อมูล **a** ในรูปแบบเวกเตอร์ แล้วทำการคำนวณค่าความยาวของข้อมูล (A) หลังจากนั้นจะมีการตัดสินใจว่าจะเลือกใช้โพลีโนเมียลความยาว 24 ($g_{CRC24A}(D)$) หรือ 16 บิต ($g_{CRC16A}(D)$) โดยพิจารณาจากความยาวข้อมูล (A) ที่เข้ามา ถ้ามากกว่า 3824 บิต จะใช้โพลีโนเมียลความยาว 24 บิต แต่ถ้าน้อยกว่าหรือเท่ากับ 3824 บิต จะใช้โพลีโนเมียลความยาว 16 บิต ถัดมากจะมีการเข้ารหัส CRC โดยใช้โพลีโนเมียลที่ตรงตามเงื่อนไขค่ารหัสที่ได้จะแทนด้วยเวกเตอร์ **b**

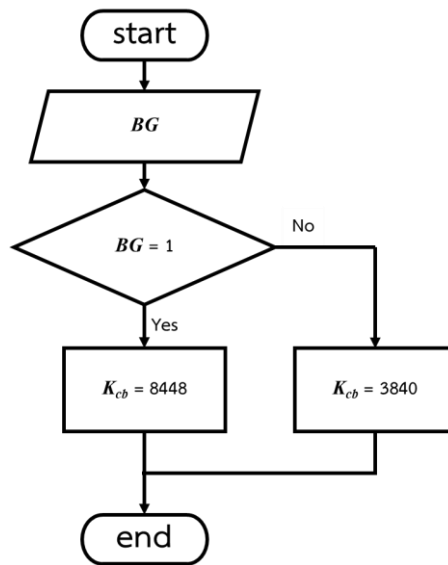


รูปที่ 3.2 แผนภาพบล็อกการเลือกกราฟฐาน

รูปที่ 3.2 แสดงแผนภาพการทำงานของซอฟต์แวร์ เริ่มจากรับค่าความยาวข้อมูล (A) ที่ได้จากแผนภาพในรูปที่ 3.1 และรับค่าอัตรารหัส (R) หลังจากนั้นจะตัดสินใจเลือกกราฟฐาน (BG) โดยพิจารณาจากความยาวข้อมูลและอัตรารหัสโดยมีเงื่อนไขดังนี้ ถ้าความยาวข้อมูลน้อยกว่าหรือเท่ากับ 292 บิต หรือ ความยาวข้อมูลน้อยกว่า 3824 บิต และอัตรารหัสน้อยกว่าหรือเท่ากับ 0.67 หรือ อัตรารหัสน้อยกว่าหรือเท่ากับ 0.25 จะเลือกใช้กราฟฐาน 2 ส่วนกรณีอื่นจะเลือกใช้กราฟฐาน 1

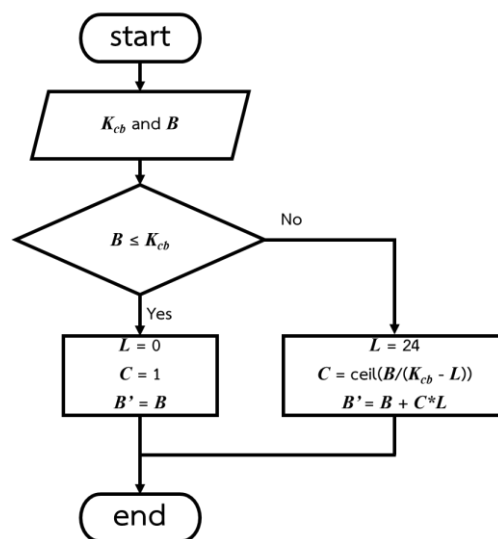
3.1.2.2 การแบ่งย่อยบล็อกข้อมูล

โมดูลหลักการแบ่งย่อยประกอบด้วย 5 โมดูลย่อยได้แก่ การกำหนดค่าความยาวบล็อกรหัส การคำนวณจำนวนบล็อกย่อย การกำหนดจำนวนบิตข้อมูลของแต่ละบล็อกรหัส การคำนวณอัตราขยายขนาดของเมทริกซ์ฐาน และการแบ่งย่อยบล็อกข้อมูล



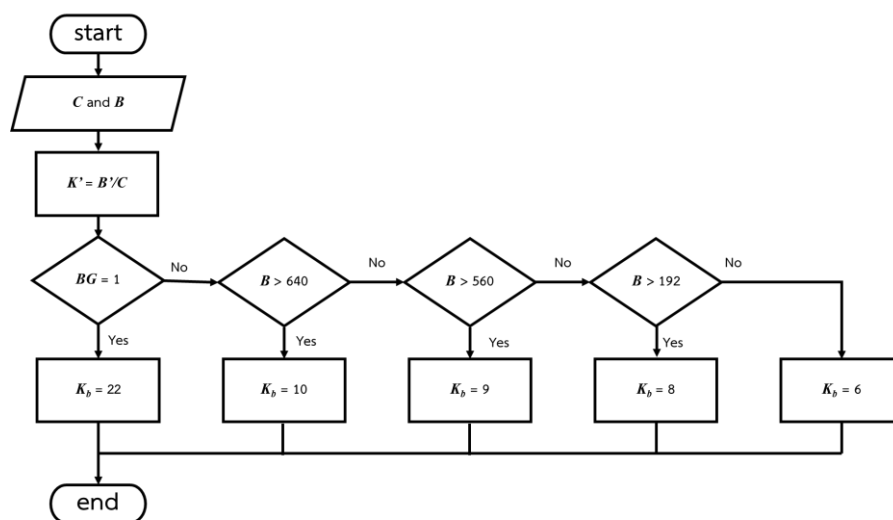
รูปที่ 3.3 แผนภาพบล็อกการกำหนดค่าความยาวบล็อกรหัส

รูปที่ 3.3 แสดงแผนภาพบล็อกการกำหนดค่าความยาวบล็อกรหัส การทำงานเริ่มจากรับค่ากราฟฐาน (BG) ที่ได้จากรูปที่ 3.2 หลังจากนั้นจะกำหนดจำนวนบิตข้อมูลในแต่ละบล็อกรหัส (K_{cb}) โดยพิจารณาจากกราฟฐาน (BG) ถ้ากราฟฐานเท่ากับ 1 จะกำหนด K_{cb} เป็น 8448 บิต ถ้าเป็นกราฟฐานเป็น 2 จะกำหนด K_{cb} เป็น 3840 บิต



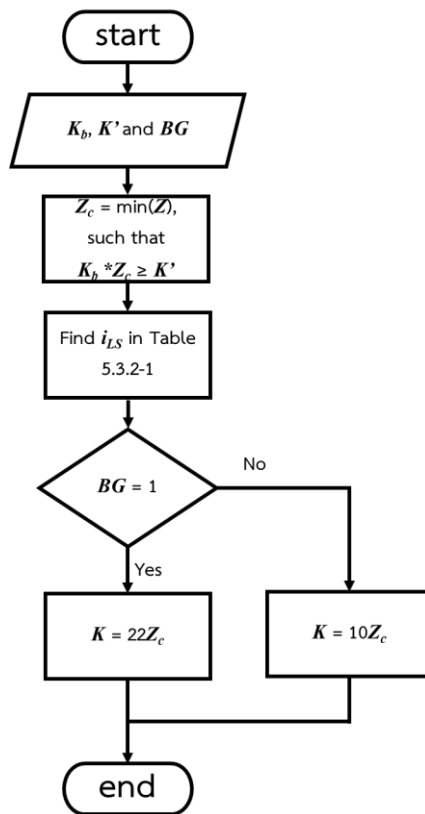
รูปที่ 3.4 แผนภาพบล็อกการคำนวณจำนวนบิตบล็อกย่อย

รูปที่ 3.4 แสดงแผนภาพบล็อกการคำนวณจำนวนบล็อกล้อย การทำงานเริ่มจากรับค่าความยาวคำรหัส (B) ที่ได้จากรูปที่ 3.1 และจำนวนบิตข้อมูลในแต่ละบล็อกรหัส (K_{cb}) ที่ได้จากรูปที่ 3.3 ถ้าความยาวคำรหัส (B) น้อยกว่าหรือเท่ากับจำนวนบิตข้อมูลในแต่ละบล็อกรหัส (K_{cb}) ที่กำหนดไว้ จำนวนบล็อกรหัส (C) จะมีบล็อกรหัสเพียง 1 บล็อก จำนวนบิตพริติของ CRC (L) จะเท่ากับ 0 บิต และจำนวนบิตข้อมูลทั้งหมด (B') จะเท่ากับความยาวคำรหัส (B) แต่ถ้าความยาวคำรหัส (B) มากกว่าจำนวนบิตข้อมูลในแต่ละบล็อกรหัส (K_{cb}) ที่กำหนดไว้ จำนวนบล็อกรหัส (C) จะถูกคำนวณโดย $\lceil B / (K_{cb} - L) \rceil$ จำนวนบิตพริติของ CRC (L) จะเท่ากับ 24 บิต ส่งผลให้จำนวนบิตข้อมูลทั้งหมด (B') จะเท่ากับ $(B + C * L)$



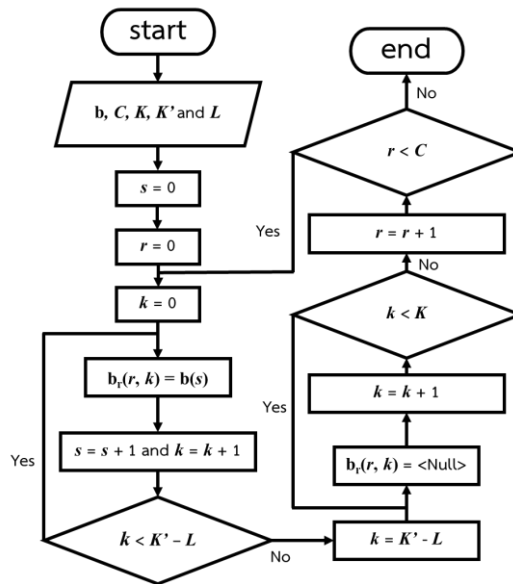
รูปที่ 3.5 แผนภาพบล็อกการกำหนดจำนวนบิตข้อมูลของแต่ละบล็อกรหัส

รูปที่ 3.5 แสดงแผนภาพบล็อกการกำหนดจำนวนบิตข้อมูลของแต่ละบล็อกรหัส การทำงานเริ่มจากรับค่าความยาวคำรหัส (B) ที่ได้จากรูปที่ 3.1 และจำนวนบล็อกรหัส (C) ที่ได้จากรูปที่ 3.4 จำนวนบิตข้อมูลที่ต้องการในแต่ละบล็อกรหัส (K_b) จะพิจารณาจากเงื่อนไขดังนี้ ถ้าเลือกใช้กราฟฐาน (BG) 1 จำนวนบิตข้อมูลที่ต้องการในแต่ละบล็อกรหัส (K_b) คือ 22 บิต ถ้าความยาวคำรหัส (B) มากกว่า 640 บิต จำนวนบิตข้อมูลที่ต้องการในแต่ละบล็อกรหัส (K_b) คือ 10 บิต ถ้าความยาวคำรหัส (B) มากกว่า 560 บิต จำนวนบิตข้อมูลที่ต้องการในแต่ละบล็อกรหัส (K_b) คือ 9 บิต ถ้าความยาวคำรหัส (B) มากกว่า 192 บิต จำนวนบิตข้อมูลที่ต้องการในแต่ละบล็อกรหัส (K_b) คือ 8 บิต และถ้านอกเหนือจากเงื่อนไขเหล่านี้ จำนวนบิตข้อมูลที่ต้องการในแต่ละบล็อกรหัส (K_b) คือ 6 บิต



รูปที่ 3.6 แผนภาพบล็อกการคำนวณอัตราขยายขนาดของเมทริกซ์ฐาน

รูปที่ 3.6 แสดงแผนภาพบล็อกการคำนวณอัตราขยายขนาดของเมทริกซ์ฐาน การทำงานเริ่มจากรับค่าจำนวนบิตข้อมูลที่ต้องการในแต่ละบล็อกรหัส (K_b) จำนวนบิตข้อมูลจริงแต่ละบล็อกรหัส (K') จากรูปที่ 3.5 และกราฟฐาน (BG) ที่ได้จากรูปที่ 3.2 ถัดมาคำนวณหาอัตราขยายขนาดของเมทริกซ์ฐาน (Z_c) โดยได้จากการหาค่าต่ำสุดของอัตราขยายขนาดของเมทริกซ์ฐาน (Z) ในตารางที่ 2.5 โดยต้องสอดคล้องกับเงื่อนไขบังคับ $K_b \cdot Z_c \geq K'$ หลังจากนั้นตรวจสอบค่า i_{LS} ของอัตราขยายขนาดของเมทริกซ์ฐาน (Z_c) ที่ได้ ถัดมาจะพิจารณาค่าความยาวบิตข้อมูลหลังการขยายขนาดเมทริกซ์ฐาน (K) ถ้าเลือกใช้กราฟฐาน 1 ค่าความยาวบิตข้อมูลหลังการขยายขนาดเมทริกซ์ฐานคือ $22Z_c$ บิต แต่ถ้าเลือกใช้กราฟฐาน 2 ค่าความยาวบิตข้อมูลหลังการขยายขนาดเมทริกซ์ฐาน (K) คือ $10Z_c$ บิต

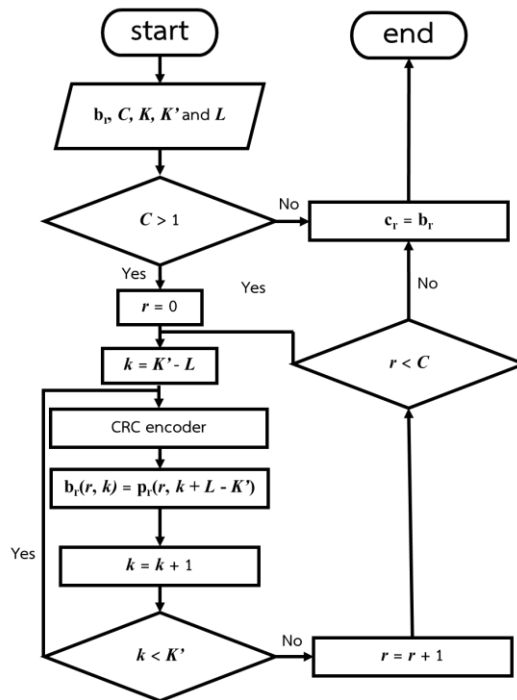


รูปที่ 3.7 แผนภาพบล็อกการแบ่งย่อยบล็อกข้อมูล

รูปที่ 3.7 แสดงแผนภาพบล็อกการแบ่งย่อยบล็อกข้อมูล การทำงานเริ่มจากรับค่าเวกเตอร์คำรหัส **b** จากโมดูลหลักการเข้ารหัส CRC จำนวนบล็อกรหัส (C) ความยาวบิตข้อมูลหลังการขยายขนาดเมทริกซ์ฐาน (K) จำนวนบิตข้อมูลจริงแต่ละบล็อกรหัส (K') และจำนวนบิตพาริตีของ CRC (L) จากรูปที่ 3.4 หลังจากนั้นจะเป็นการนำบิตข้อมูลในเวกเตอร์ **b** ไปเก็บในแต่ละบล็อกรหัสย่อย **b_r** ในตำแหน่งที่ 0 ถึง $K' - L - 1$ ส่วนในตำแหน่งที่ $K' - L$ ถึง K จะเป็นบิตว่าง <NULL>

3.1.2.3 การเข้ารหัส CRC ของบล็อกรหัส

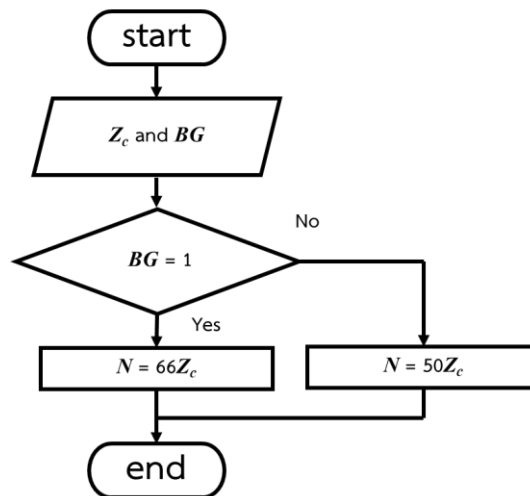
โมดูลการเข้ารหัส CRC ครั้งที่ 2 จะเป็นการเข้ารหัส CRC ให้กับบล็อกรหัส รูปที่ 3.8 แสดงแผนภาพบล็อกการเข้ารหัส CRC ของบล็อกรหัส การทำงานเริ่มจากรับค่าเวกเตอร์บล็อกรหัส **b_r**, จำนวนบล็อกรหัส (C) ความยาวบิตข้อมูลหลังการขยายขนาดเมทริกซ์ฐาน (K) จำนวนบิตข้อมูลจริงแต่ละบล็อกรหัส (K') และจำนวนบิตพาริตีของ CRC (L) ถัดมาจะพิจารณาเงื่อนไขการเข้ารหัส CRC ของบล็อกรหัสดังนี้ ถ้าบล็อกรหัสย่อยมีเพียง 1 บล็อกจะไม่มีกรเข้ารหัส CRC และเวกเตอร์บล็อกรหัสอันใหม่ **c_r** จะกำหนดให้เท่ากับ **b_r** ในกรณีที่จำนวนบล็อกรหัสย่อยมีมากกว่า 1 บล็อกจะทำการเข้ารหัส CRC ให้กับบล็อกรหัสทีละบล็อกจะได้เวกเตอร์ **p_r**, หลังจากนั้นจะนำพาริตีมาในตำแหน่ง $K' - L$ ถึง K' เติมให้กับเวกเตอร์ **b_r** ในตำแหน่งที่ $K' - L$ ถึง K' และในขั้นตอนสุดท้ายเวกเตอร์บล็อกรหัสอันใหม่ **c_r** จะกำหนดให้เท่ากับ **b_r**



รูปที่ 3.8 แผนภาพบล็อกการเข้ารหัส CRC ของบล็อกรหัส

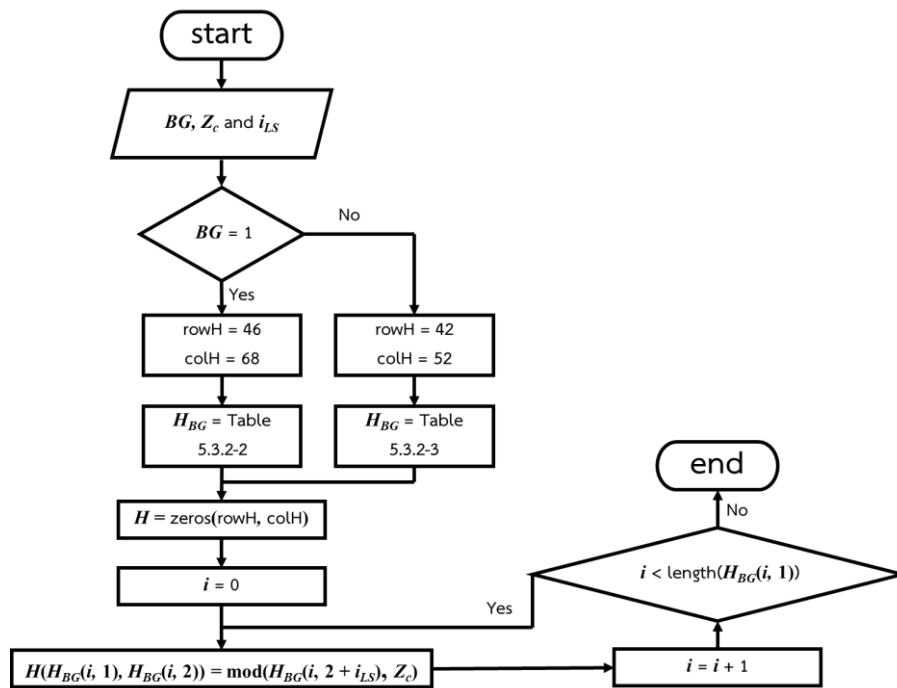
3.1.2.4 การเข้ารหัส LDPC

โมดูลหลักการเข้ารหัส LDPC ประกอบไปด้วย 4 โมดูลย่อยได้แก่ การกำหนดความยาวคำรหัส การสร้างเมทริกซ์พาริตีเช็ก H การเติมบิตข้อมูลในบล็อกรหัส และการเข้ารหัส LDPC



รูปที่ 3.9 แผนภาพบล็อกการกำหนดความยาวคำรหัส

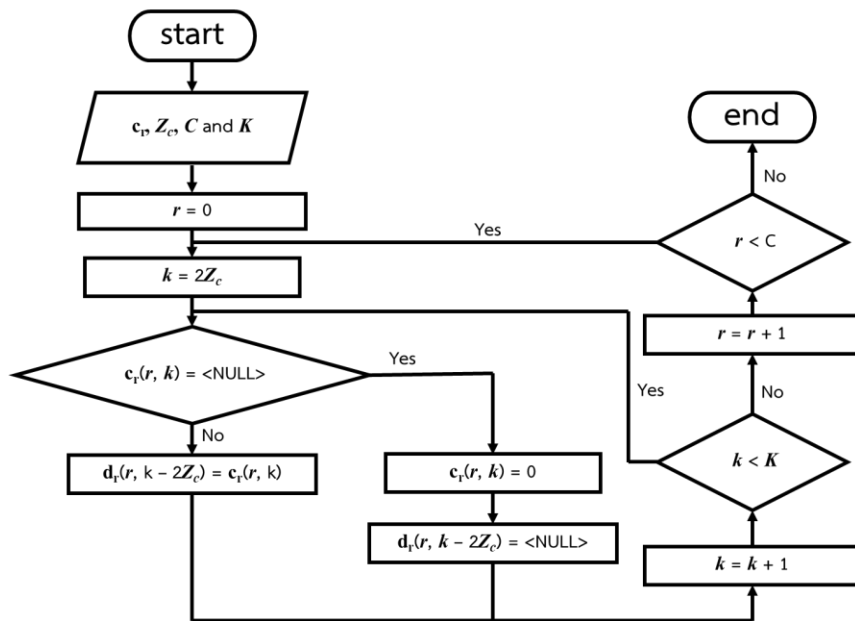
รูปที่ 3.9 แสดงแผนภาพบล็อกการกำหนดความยาวคำรหัส การทำงานเริ่มจากรับค่า Z_c และ BG หลังจากนั้นจะพิจารณาเงื่อนไขต่อไปนี้ ถ้าเลือกใช้กราฟฐาน 1 ความยาวคำรหัส N จะเท่ากับ $66Z_c$ บิต ถ้าเลือกใช้กราฟฐาน 2 ความยาวคำรหัส N จะเท่ากับ $50Z_c$ บิต



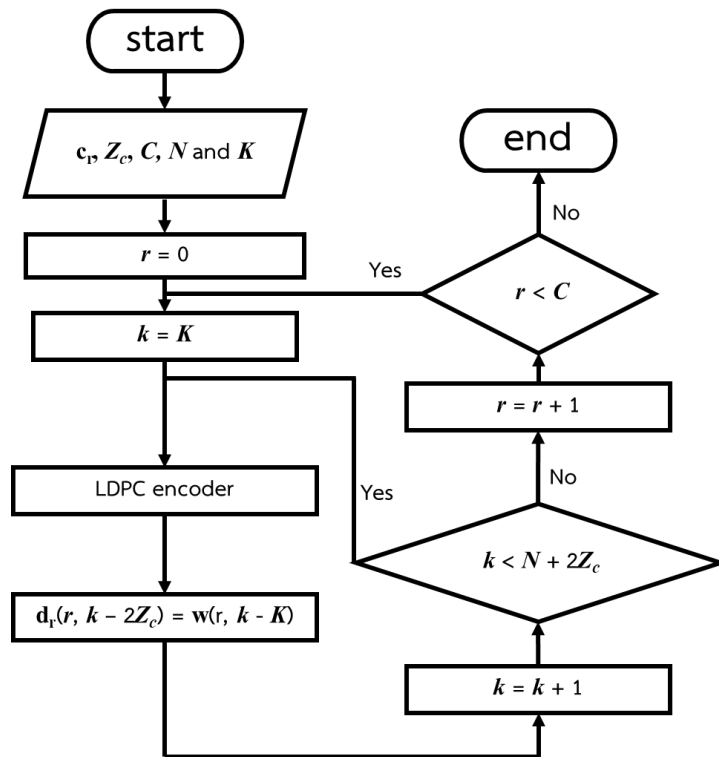
รูปที่ 3.10 แผนภาพบล็อกการสร้างเมทริกซ์พาริตีเช็ค H

รูปที่ 3.10 แสดงแผนภาพบล็อกการสร้างเมทริกซ์พาริตีเช็ค H การทำงานเริ่มจากรับค่า BG Z_c และ i_{LS} ถัดมาเป็นการกำหนดขนาดของเมทริกซ์ H และการเลือกใช้ตารางการสร้าง H โดยพิจารณาจากเงื่อนไขต่อไปนี้ ถ้าเลือกใช้กราฟฐาน 1 ขนาดของเมทริกซ์ H คือ 46×68 และเลือกใช้ค่าในตารางที่ 2.6 ถ้าเลือกใช้กราฟฐาน 2 ขนาดของเมทริกซ์ H คือ 42×52 และเลือกใช้ค่าในตารางที่ 2.7 หลังจากนั้นจะสร้างเมทริกซ์ H ให้มีค่า 0 ทั้งหมดโดยมีขนาดที่กำหนดตามเงื่อนไขข้างต้น หลังจากนั้นทำการเติมค่าของเมทริกซ์ในตำแหน่งที่กำหนดในตาราง โดยค่าจำนวนเต็มในตารางจะสอดคล้องกับ Z_c ดังสมการที่ (10)

รูปที่ 3.11 แสดงแผนภาพบล็อกการเติมบิตข้อมูลในบล็อกรหัส การทำงานเริ่มจากรับค่าเวกเตอร์ c_r Z_c C และ K ถัดมาเป็นการเติมบิตข้อมูล c_r ลงไปในเวกเตอร์ d_r ในตำแหน่งที่ $2Z_c$ ถึง K เนื่องจากข้อมูลใน 2 หลักแรกจะไม่ถูกส่งออกไป ขั้นตอนถัดมาจะเป็นการพิจารณาค่าในเวกเตอร์ c_r ถ้าบิตในตำแหน่งดังกล่าวเป็นบิตว่าง <NULL> จะเติมบิต 0 ลงไปแทน



รูปที่ 3.11 แผนภาพบล็อกการเติมบิตข้อมูลในบล็อกรหัส

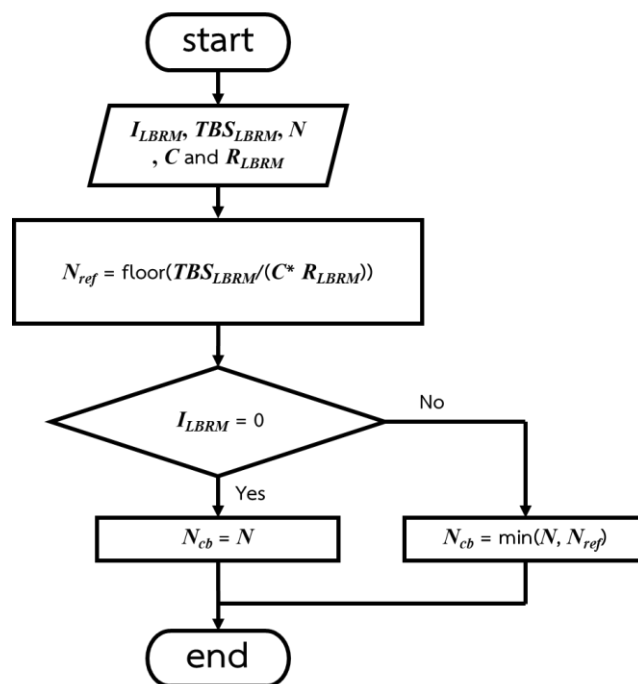


รูปที่ 3.12 แผนภาพบล็อกการเข้ารหัส LDPC

รูปที่ 3.12 แสดงแผนภาพบล็อกการเข้ารหัส LDPC การทำงานเริ่มจากรับค่าเวกเตอร์ \mathbf{c} , Z_c , C , N และ K ถัดมาเป็นกระบวนการเข้ารหัส LDPC แต่ละบล็อกรหัสจะได้คำรหัส \mathbf{w} และนำบิตพาริตีในเวกเตอร์ \mathbf{w} เติมในเวกเตอร์ $\mathbf{A} \geq 1013 \vee ()$ ในตำแหน่งที่ K ถึง $N + 2Z_c$

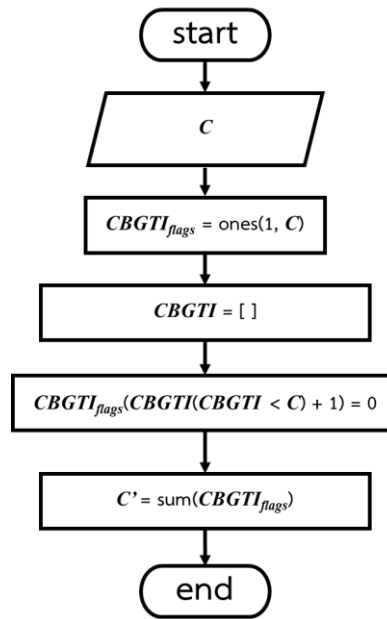
3.1.2.5 การปรับอัตรารหัส

โมดูลหลักการการปรับอัตรารหัส ประกอบไปด้วย 4 โมดูลย่อยได้แก่ การกำหนดความยาวคำรหัสของบล็อกรหัส การคำนวณข้อมูลการส่งกลุ่มของบล็อกรหัส ($CBGTI$) การคำนวณความยาวข้อมูลเอาต์พุตของกระบวนการปรับอัตรารหัส และการเลือกบิตเพื่อส่ง



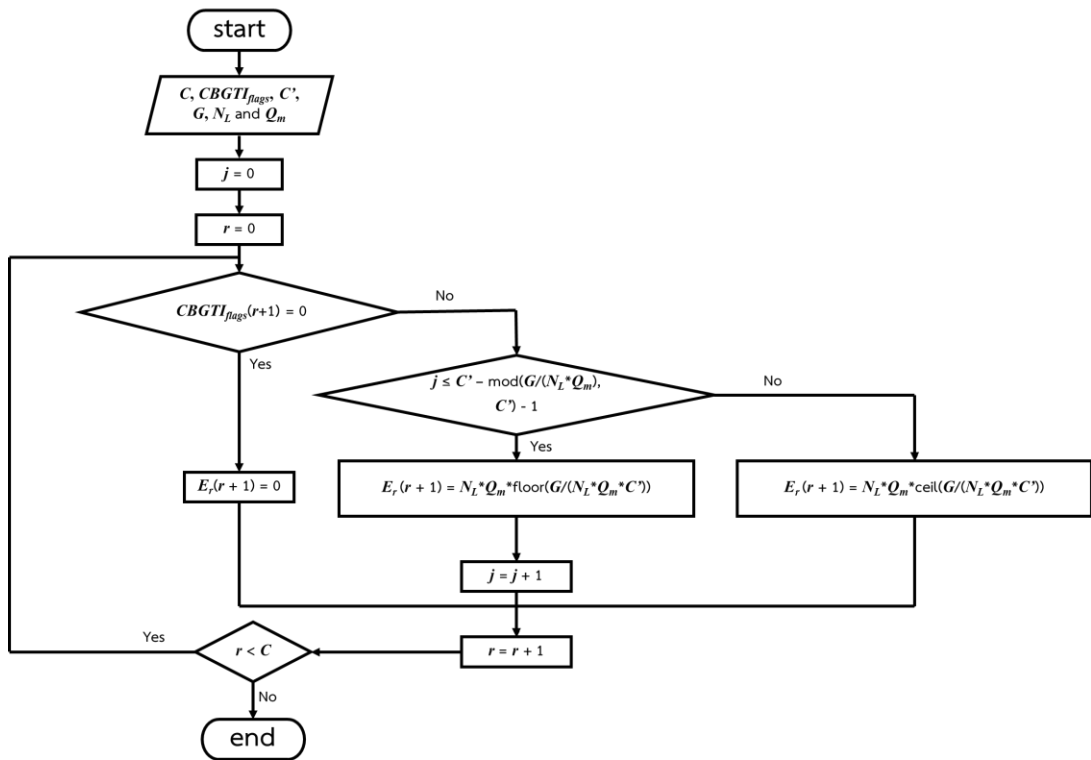
รูปที่ 3.13 แผนภาพบล็อกการกำหนดความยาวคำรหัสของบล็อกรหัส

รูปที่ 3.13 แสดงแผนภาพบล็อกการกำหนดความยาวคำรหัสของบล็อกรหัส การทำงานเริ่มจากรับค่า I_{LBRM} , TBS_{LBRM} , N , C และ R_{LBRM} ถัดมาคำนวณ N_{ref} หลังจากนั้นพิจารณาเงื่อนไขตรรกะของบัพเฟอร์ที่ถูกจำกัด ถ้า I_{LBRM} เท่ากับ 0 ความยาวคำรหัสของบล็อกรหัส (N_{cb}) กำหนดให้เท่ากับความยาวคำรหัสของ LDPC (N) แต่ถ้า I_{LBRM} ไม่เท่ากับ 0 ความยาวคำรหัสของบล็อกรหัส (N_{cb}) กำหนดให้เท่ากับค่าต่ำสุดระหว่างความยาวคำรหัสของ LDPC (N) และ N_{ref}



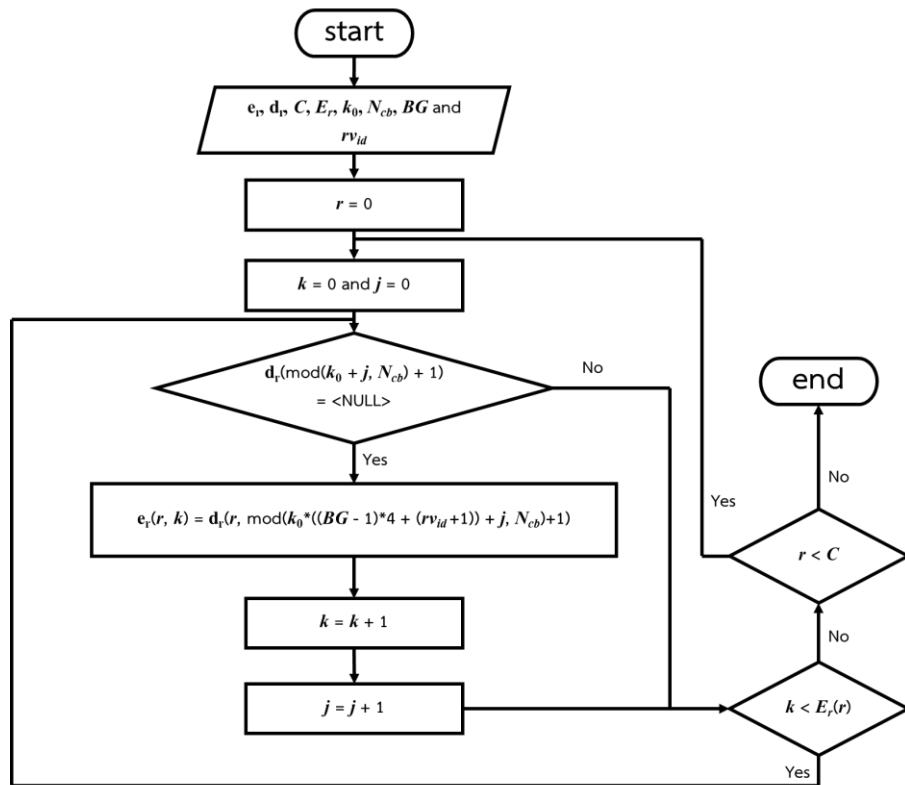
รูปที่ 3.14 แผนภาพบล็อกการคำนวณข้อมูลการส่งกลุ่มของบล็อกรหัส (CBGTI)

รูปที่ 3.14 แสดงแผนภาพบล็อกการคำนวณข้อมูลการส่งกลุ่มของบล็อกรหัส (CBGTI) การทำงานเริ่มจากรับค่า C ถัดมากำหนดให้ $CBGTI_{flags}$ แต่ละบล็อกรหัสมีค่า 1 และกำหนดให้ $CBGTI$ เป็นเซตว่าง หลังจากนั้นเติมบิต 0 ให้กับ $CBGTI_{flags}$ ในตำแหน่งที่ $CBGTI$ น้อยกว่า C สุดท้าย C คำนวณมาจากผลรวมของ $CBGTI_{flags}$



รูปที่ 3.15 แผนภาพบล็อกการคำนวณความยาวข้อมูลเอาต์พุตของกระบวนการปรับอัตราหัส

รูปที่ 3.15 แสดงแผนภาพบล็อกการคำนวณความยาวข้อมูลเอาต์พุตของกระบวนการปรับอัตราหัส การทำงานเริ่มจากรีค่า C $CBGTI_{flags}$ C' G N_L และ Q_m ถัดมาจะพิจารณาเงื่อนไขแรก ถ้าค่า $CBGTI_{flags}$ ของบล็อกรหัสเท่ากับ 0 กำหนดให้ความยาวบล็อกรหัส E_r เท่ากับ 0 ถ้าค่า $CBGTI_{flags}$ ของบล็อกรหัสไม่ใช่ค่า 0 จะพิจารณาเงื่อนไข $j \leq C' - \text{mod}(G/N_L \cdot Q_m, C') - 1$ ถ้าเป็นไปตามเงื่อนไขจะคำนวณค่า E_r ตามสมการที่ (11) ถ้าไม่เป็นไปตามเงื่อนไขจะคำนวณค่า E_r ตามสมการที่ (12)

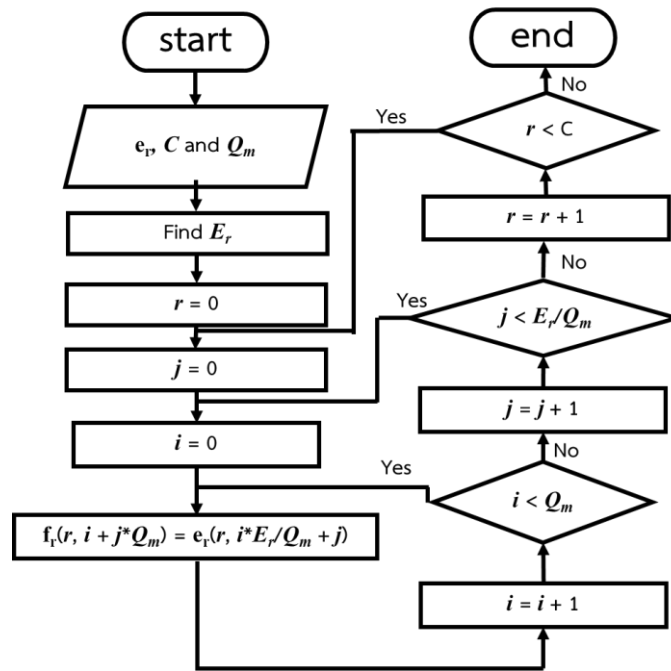


รูปที่ 3.16 แผนภาพบล็อกการเลือกบิตเพื่อส่ง

รูปที่ 3.16 แสดงแผนภาพบล็อกการเลือกบิตเพื่อส่ง การทำงานเริ่มจากรับค่า e_r , d_r , C , E_r , k_0 , N_{cb} , BG และ rv_{id} ถัดมา พิจารณาเงื่อนไขต่อไปนี้ ถ้าเวกเตอร์ d_r เป็นบิตว่าง <NULL> จะนำค่าบิตในตำแหน่ง $\text{mod}(k_0(4(BG-1) + rv_{id} + 1) + j, N_{cb})$ มาใส่ในเวกเตอร์ e_r ถ้าเวกเตอร์ d_r ไม่ใช่บิตว่าง <NULL> จะไม่ทำอะไรกับข้อมูล

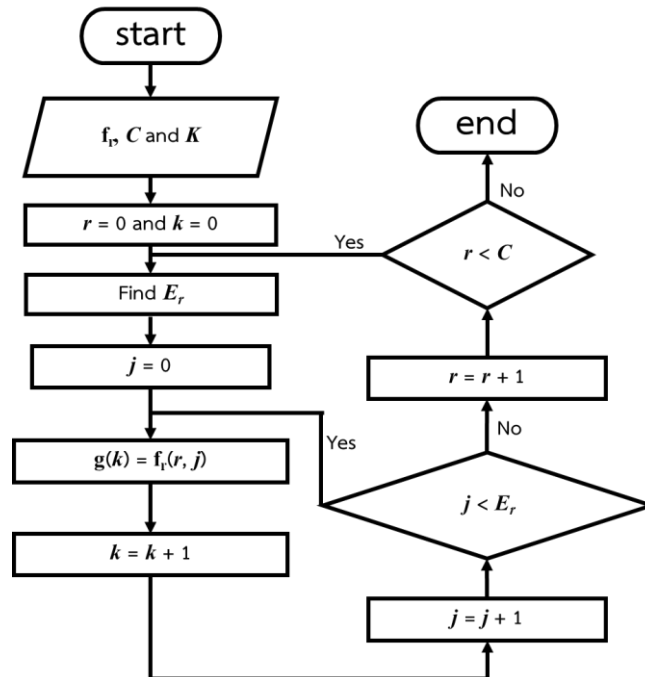
3.1.2.6 การแทรกสลับบิต

รูปที่ 3.17 แสดงแผนภาพบล็อกการแทรกสลับบิต การทำงานเริ่มจากรับค่าเวกเตอร์ e_r , C และ Q_m ถัดมาคำนวณความยาวข้อมูล E_r หลังจากนั้นจะแทรกสลับบิตข้อมูลโดยนำบิตข้อมูลของ e_r ในตำแหน่ง $i * E_r / Q_m + j$ ไปใส่ในเวกเตอร์ f_r ในตำแหน่ง $i + j * Q_m$



รูปที่ 3.17 แผนภาพบล็อกการแทรกสลับบิต

3.1.2.7 การต่อเรียงบล็อกรหัส



รูปที่ 3.18 แผนภาพบล็อกการต่อเรียงบล็อกรหัส

รูปที่ 3.18 แสดงแผนภาพบล็อกการต่อเรียงบล็อกรหัส การทำงานเริ่มจากรับค่าเวกเตอร์ \mathbf{f} , C และ K ถัดมาคำนวณค่าความยาวบล็อกรหัส E , หลังจากนั้นบล็อกรหัสย่อยจะถูกนำไปเรียงต่อกันในเวกเตอร์ \mathbf{g} ตามลำดับ

3.2 รายละเอียดการออกแบบชุดซอฟต์แวร์และวงจรเข้ารหัสช่องสัญญาณชนิด Polar codes

3.2.1 ส่วนประกอบชุดซอฟต์แวร์และวงจร

ส่วนประกอบชุดซอฟต์แวร์และวงจรการเข้ารหัสจะประกอบไปด้วยกระบวนการทำงาน 11 ส่วนดังนี้ 1) การแบ่งย่อยบล็อกรหัส 2) การเข้ารหัส CRC 3) การสแครมบลิง CRC 4) การแทรกสลับ CRC 5) การลำดับช่องสัญญาณย่อย 6) การคำนวณบิตพาริตี 7) การเข้ารหัสโพลาร์ 8) การแทรกสลับบล็อกย่อย 9) การปรับอัตรารหัส 10) การแทรกสลับบิตรหัส 11) การต่อบล็อกรหัส ซึ่งจำนวนโมดูลหลักของซอร์สโค้ดมีจำนวน 5 โมดูลหลักและมีโมดูลย่อยภายในโมดูลหลักประกอบการทำงาน ซึ่งมีจำนวนครอบคลุมเท่ากับกระบวนการทำงานทั้งหมด

โมดูลหลักทั้งหมด 5 โมดูล ประกอบไปด้วยไฟล์สกุล m ดังนี้

1) CodeBlockSegmentation.m ของกระบวนการแบ่งย่อยบล็อกรหัส

2) CRCAttachment.m ของกระบวนการเข้ารหัส การสแครมบลิง และการแทรกสลับ CRC และมีโมดูลย่อยประกอบไปด้วย

2.1) CRCAttachment

2.2) CRCInterleaver

2.3) CRCFunction

2.4) RNTISelection

3) PolarSequencing.m ของกระบวนการลำดับช่องสัญญาณย่อยและการคำนวณบิตพาริตี และมีโมดูลย่อยประกอบไปด้วย

3.1) PolarSequencing

3.2) SubBlockInterleaver

3.3) PolarGeneratorMatrix

3.4) ParityCheckEncoding

4) PolarEncoding.m ของกระบวนการเข้ารหัสโพลาร์ การแทรกสลับบล็อกย่อย และการปรับอัตรารหัส และมีโมดูลย่อยประกอบไปด้วย

5) CodedBitInterleaver.m ของกระบวนการแทรกสลับบิตรหัสและการต่อบล็อกรหัส

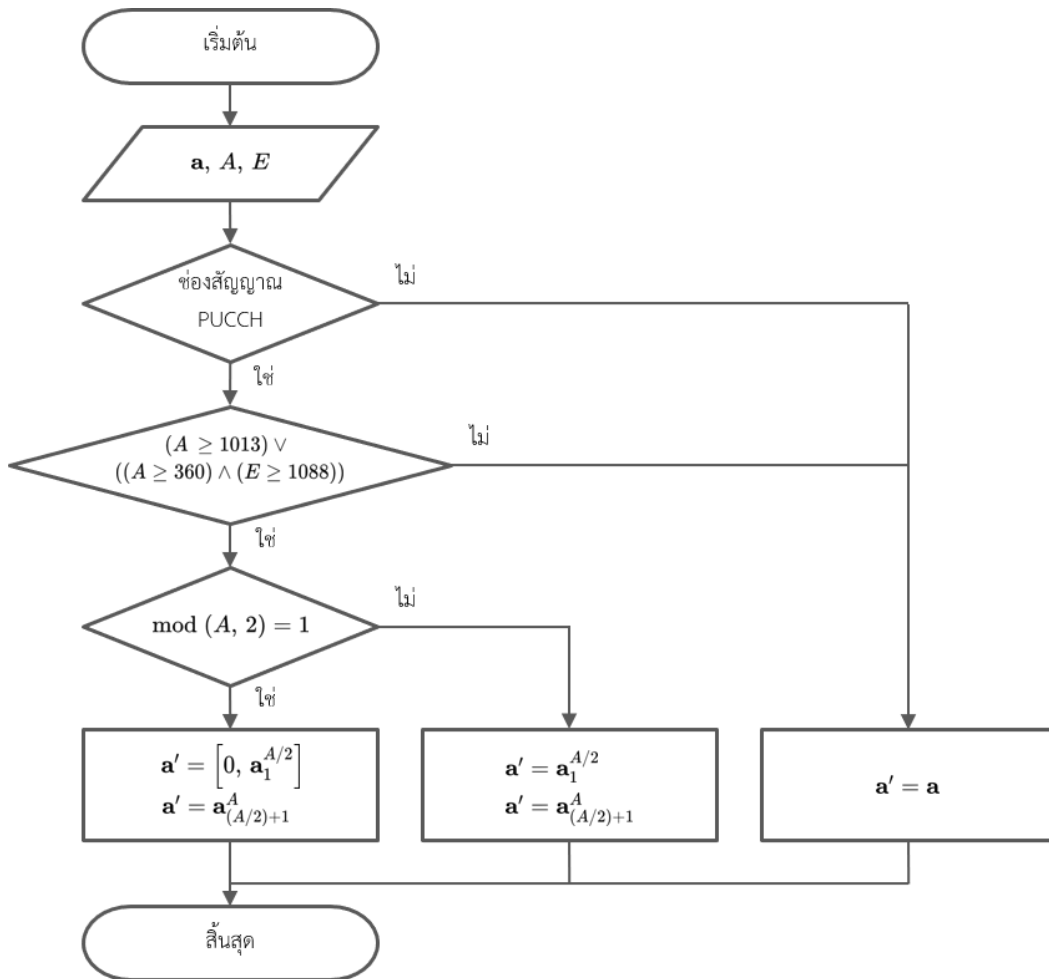
3.2.2 รายละเอียดการออกแบบชุดซอฟต์แวร์และวงจรเข้ารหัสช่องสัญญาณชนิด Polar codes มาตรฐาน 5G

รายละเอียดการออกแบบชุดซอฟต์แวร์จะแบ่งหัวข้อย่อยตามซอฟต์แวร์โมดูลหลัก 5 หลัก และจะอธิบายรายละเอียดของซอฟต์แวร์โมดูลย่อยในโมดูลหลักภายใต้หัวข้อย่อย

3.2.2.1 การแบ่งย่อยบล็อกรหัส

โมดูลการแบ่งย่อยบล็อกรหัสสามารถเข้าถึงได้ที่ไฟล์ CodeBlockSegmentation.m

การแบ่งย่อยบล็อกรหัสแบ่งข้อมูลขาเข้าเป็นสองส่วน โดยเวกเตอร์ \mathbf{a} ความยาว A จะถูกแบ่งครึ่งออกเป็นสองเวกเตอร์ \mathbf{a}' ความยาว A' โมดูลจะทำงานเฉพาะช่องสัญญาณ PUCCH ในกรณี ที่ $A \geq 1013 \vee (A \geq 360 \wedge E \geq 1088)$ สำหรับกรณีอื่นและช่องสัญญาณ PBCH และ PDCCH โมดูลจะไม่ทำงานและแทนเวกเตอร์ $\mathbf{a}' = \mathbf{a}$ ที่ความยาว $A' = A$ แผนภาพการทำงานสามารถแสดง ได้ดังรูปที่ 3.19



รูปที่ 3.19 แผนผังการทำงานการแบ่งย่อยบิตรอกซ์

3.2.2.2 การเข้ารหัส การสแครมบลิง และการแทรกสลับ CRC

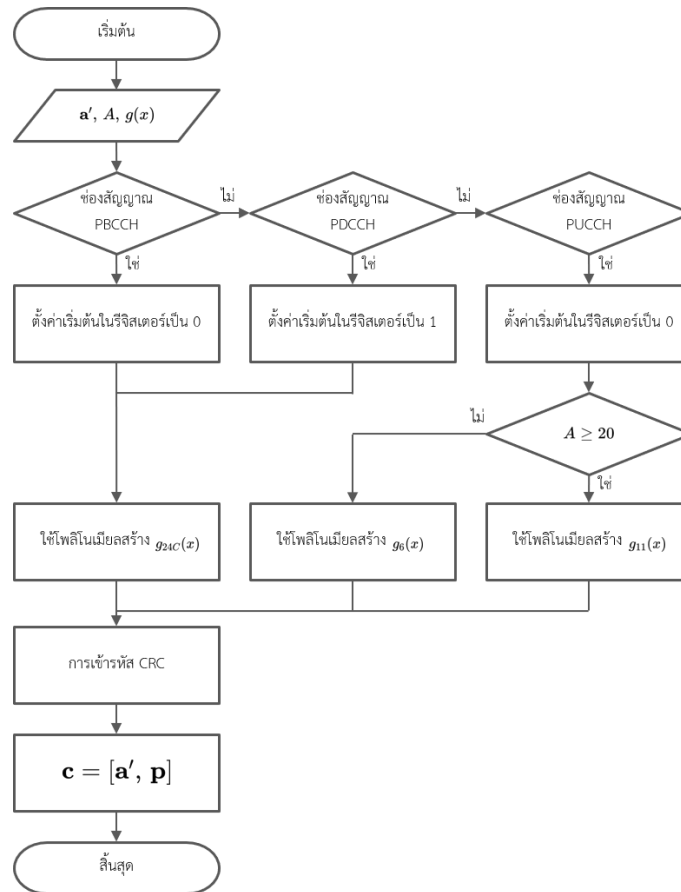
โมดูลการเข้ารหัส การสแครมบลิง และการแทรกสลับ CRC สามารถเข้าถึงได้ที่ไฟล์ CRCAttachment.m

3.2.2.2.1 การเชื่อมบิต CRC ท้ายข้อมูล

ระบบจะทำการเข้ารหัส CRC ซึ่งจะได้บิต CRC หรือเวกเตอร์ \mathbf{p} ความยาว L และต่อท้ายเวกเตอร์ \mathbf{a}' ได้ผลลัพธ์เป็นเวกเตอร์ \mathbf{c} ความยาว $K = A' + L$ แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 3.20

สำหรับช่องสัญญาณ PBCH และ PDCCH โพลีโนเมียลสร้างที่จะเลือกใช้คือ $g_{24C}(x)$ ที่มีความยาว $L = 24$ และสำหรับช่องสัญญาณ PDCCH ค่าเริ่มต้นในรีจิสเตอร์จะตั้งค่าเป็น 1 ทั้งหมด

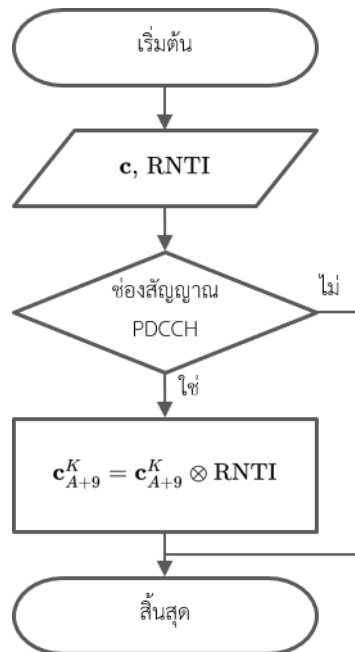
สำหรับช่องสัญญาณ PUCCH หากความยาว $A \geq 20$ จะเลือกใช้โพลีโนเมียลสร้าง $g_{11}(x)$ ที่มีความยาว $L=11$ และหากความยาว $A \leq 19$ จะเลือกใช้โพลีโนเมียลสร้าง $g_6(x)$ ที่มีความยาว $L=6$



รูปที่ 3.20 แผนผังการทำงานการเชื่อมบิต CRC ท้ายข้อมูล

3.2.2.2.2 การสแครมบลิง CRC

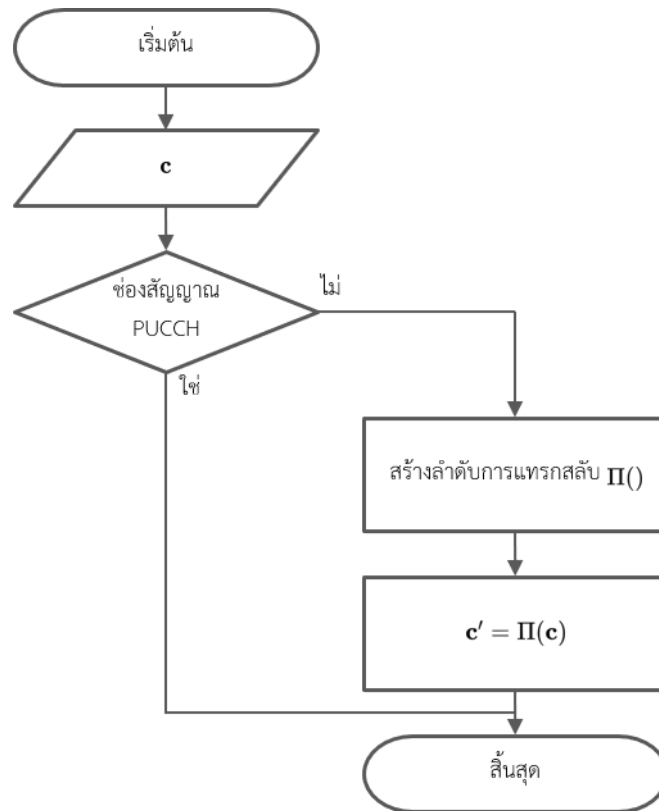
การสแครมบลิง CRC จะนำบิต RNTI ความยาว 16 บิตมาทำกระบวนการเอ็กซ์คลูซีฟออร์กับ บิต CRC 16 บิตสุดท้ายหรือเวกเตอร์ $c_{A'+9}^K$ แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 3.21



รูปที่ 3.21 แผนผังการทำงานการสแครมบลิง CRC

3.2.2.2.3 การแทรกสลับ CRC

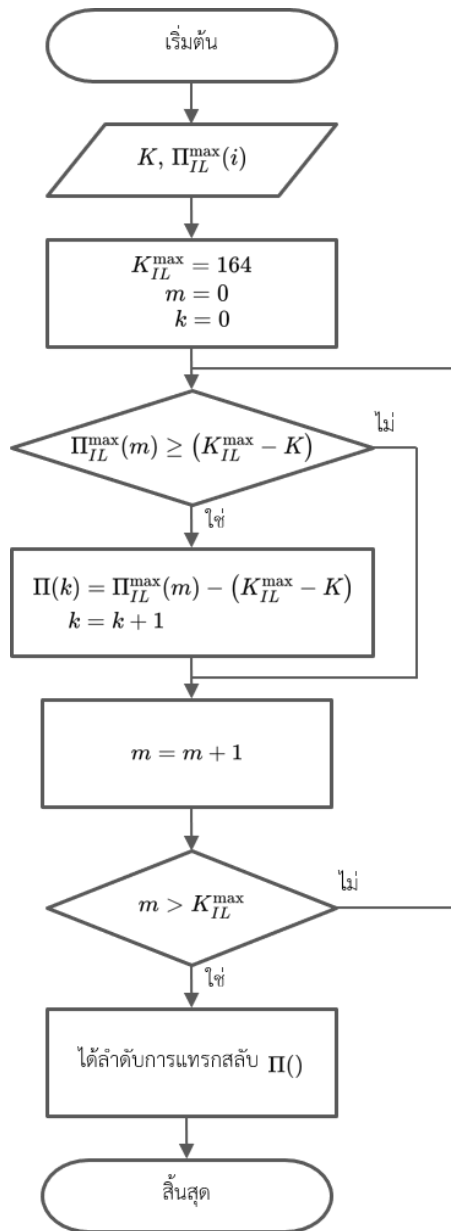
การแทรกสลับ CRC จะนำเวกเตอร์ \mathbf{c} ที่ผ่านการเข้ารหัส CRC หรือการสแครมบลิงมาแทรกสลับตำแหน่งบิต ได้ผลลัพธ์เป็นเวกเตอร์ \mathbf{c}' ที่มีความยาว K ที่มีความยาวเท่ากับเวกเตอร์ \mathbf{c} แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 3.22 โมดูลจะทำงานเฉพาะช่องสัญญาณ PBCH และ PDCCH ที่มีการเข้ารหัส CRC ด้วยโพลีโนเมียลสร้าง $g_{24C}(x)$ สำหรับช่องสัญญาณ PUCCH โมดูลจะไม่ทำงาน



รูปที่ 3.22 แผนผังการทำงานการแทรกสลับ CRC

3.2.2.2.4 การสร้างลำดับการแทรกสลับ CRC

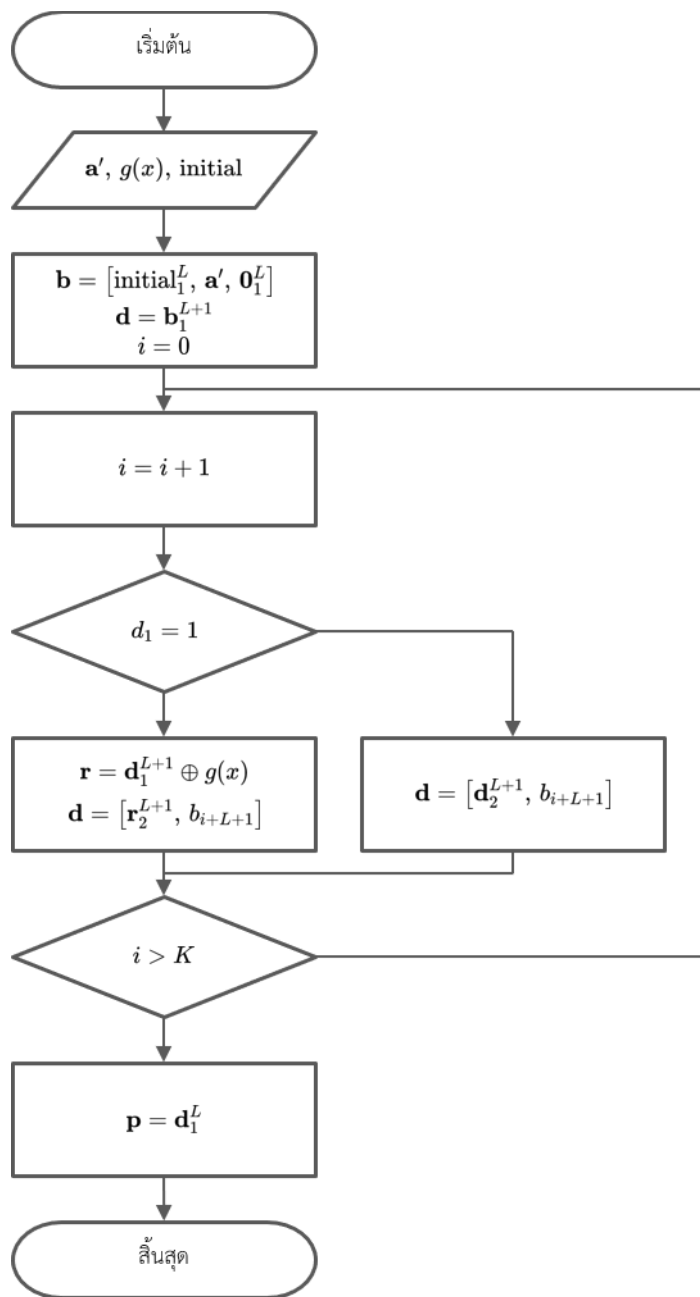
ลำดับการแทรกสลับ CRC ถูกสร้างโดยอัลกอริทึมที่ได้ระบุไว้ตามมาตรฐานร่วมกับลำดับการแทรกสลับตามตารางที่ 10 แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 3.23



รูปที่ 3.23 แผนผังการทำงานการสร้างลำดับการแทรกกลับ CRC

3.2.2.2.5 การเข้ารหัส CRC

การเข้ารหัส CRC จะเข้ารหัส CRC แก่เวกเตอร์ \mathbf{a}' จากกระบวนการก่อน โดยโมดูลจะเลือก โพลีโนเมียลสร้างในการเข้ารหัส CRC จากเงื่อนไขของสัญญาณและความยาว A การเข้ารหัส CRC สามารถดำเนินการได้โดยการใช้งาน cyclic shift register ที่มีการตั้งค่าเริ่มต้นในรีจิสเตอร์เป็น 0 ทั้งหมดยกเว้นช่องสัญญาณ PDCCH ที่ตั้งค่าเริ่มต้นในรีจิสเตอร์เป็น 1 แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 3.24



รูปที่ 3.24 แผนผังการทำงานการเข้ารหัส CRC

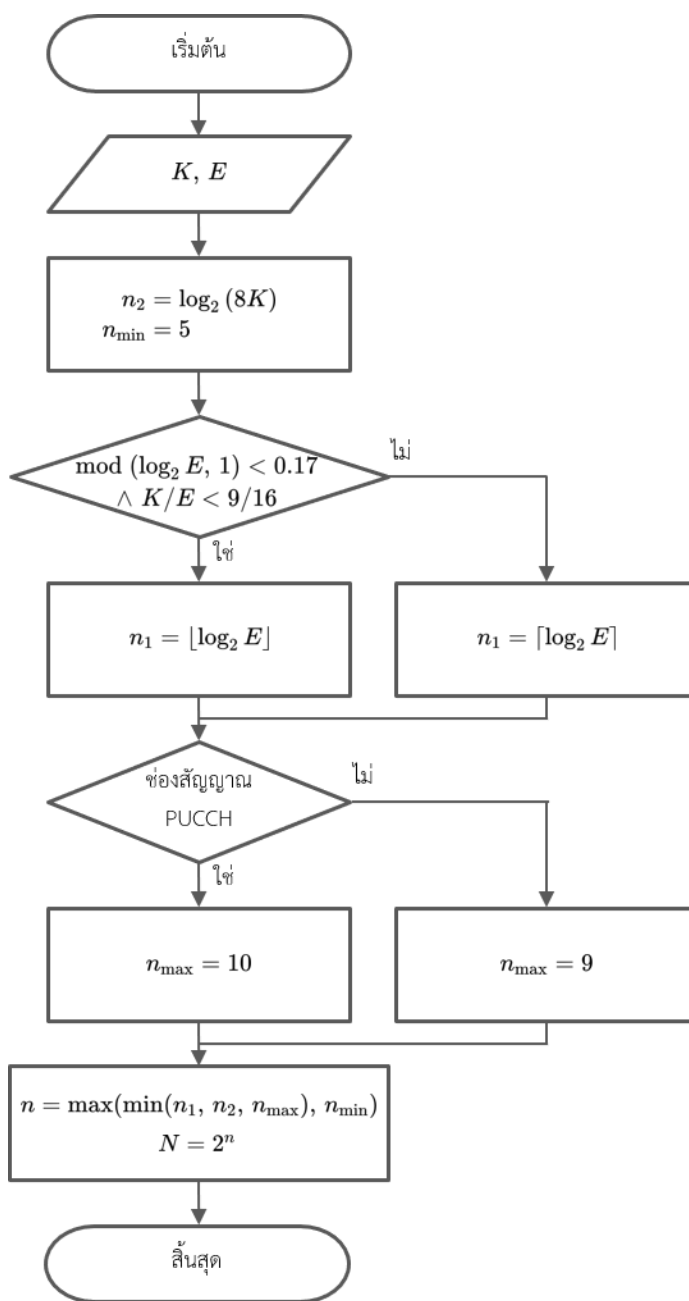
3.2.2.3 การลำดับช่องสัญญาณย่อยและการคำนวณบิตพาริตี

โมดูลการลำดับช่องสัญญาณย่อยและการคำนวณบิตพาริตีสามารถเข้าถึงได้ที่ไฟล์

PolarSequencing.m

3.2.2.3.1 การหาความยาวคำรหัสแม่

ก่อนการลำดับช่องสัญญาณย่อย ระบบจำเป็นต้องหาความยาวคำรหัสแม่ N สำหรับการลำดับช่องสัญญาณย่อยและการเข้ารหัสโพลาไร โดย N ต้องมีค่าเท่ากับสองกำลังใด ๆ การคำนวณความยาวคำรหัสแม่จะนำเวกเตอร์ \mathbf{c} ที่มีความยาว K และความยาวคำรหัส E มาพิจารณาเพื่อให้ได้ความยาวคำรหัสแม่ที่เหมาะสม แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 3.25



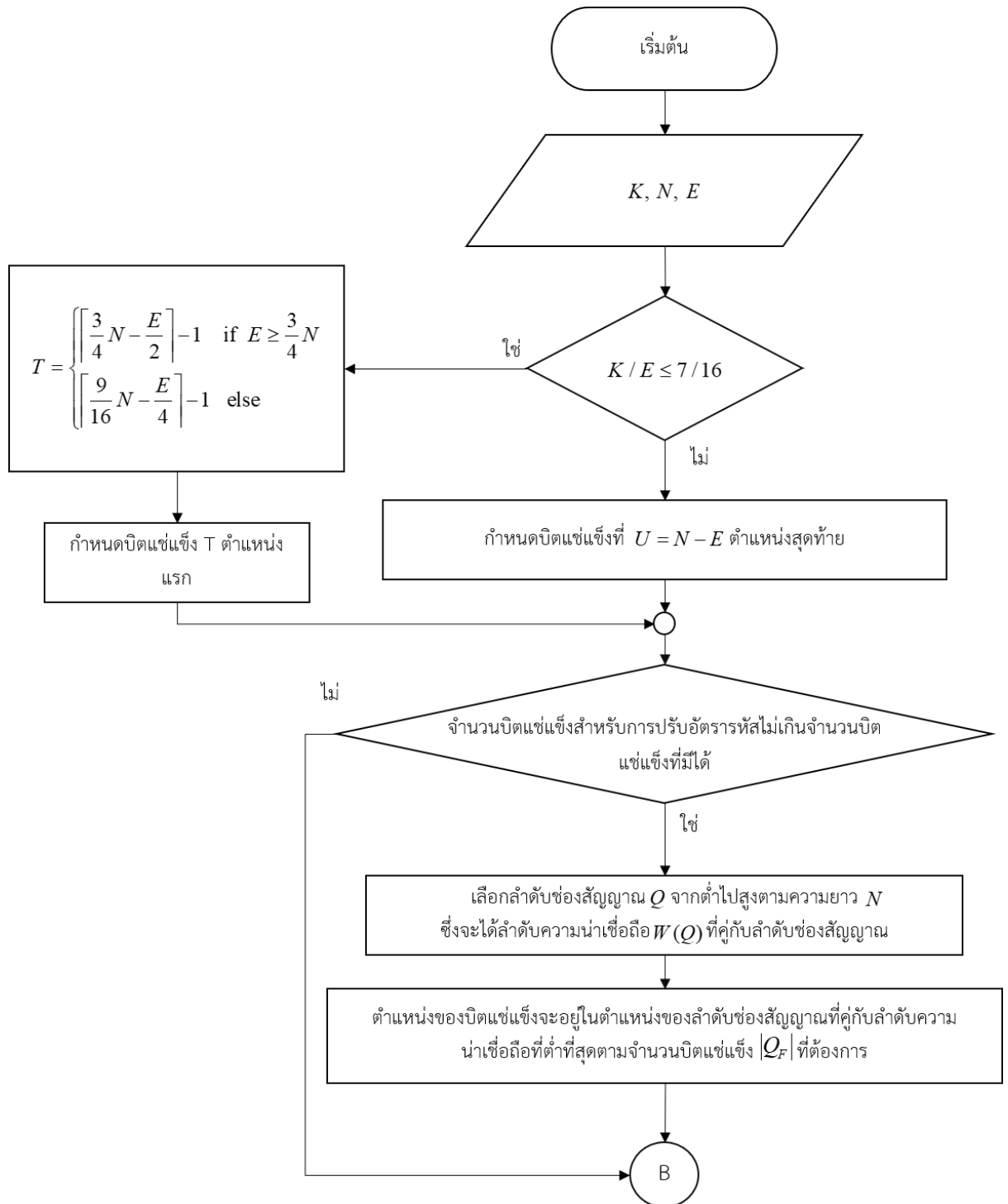
รูปที่ 3.25 แผนผังการทำงานการหาความยาวคำรหัสแม่

3.2.2.3.2 การลำดับช่องสัญญาณย่อย

การลำดับช่องสัญญาณย่อย ระบบจะค้นหาตำแหน่งบิตแชนแนลของรหัสโพลาร์ตามเงื่อนไขต่าง ๆ แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 3.26 และสามารถสรุปได้ดังนี้

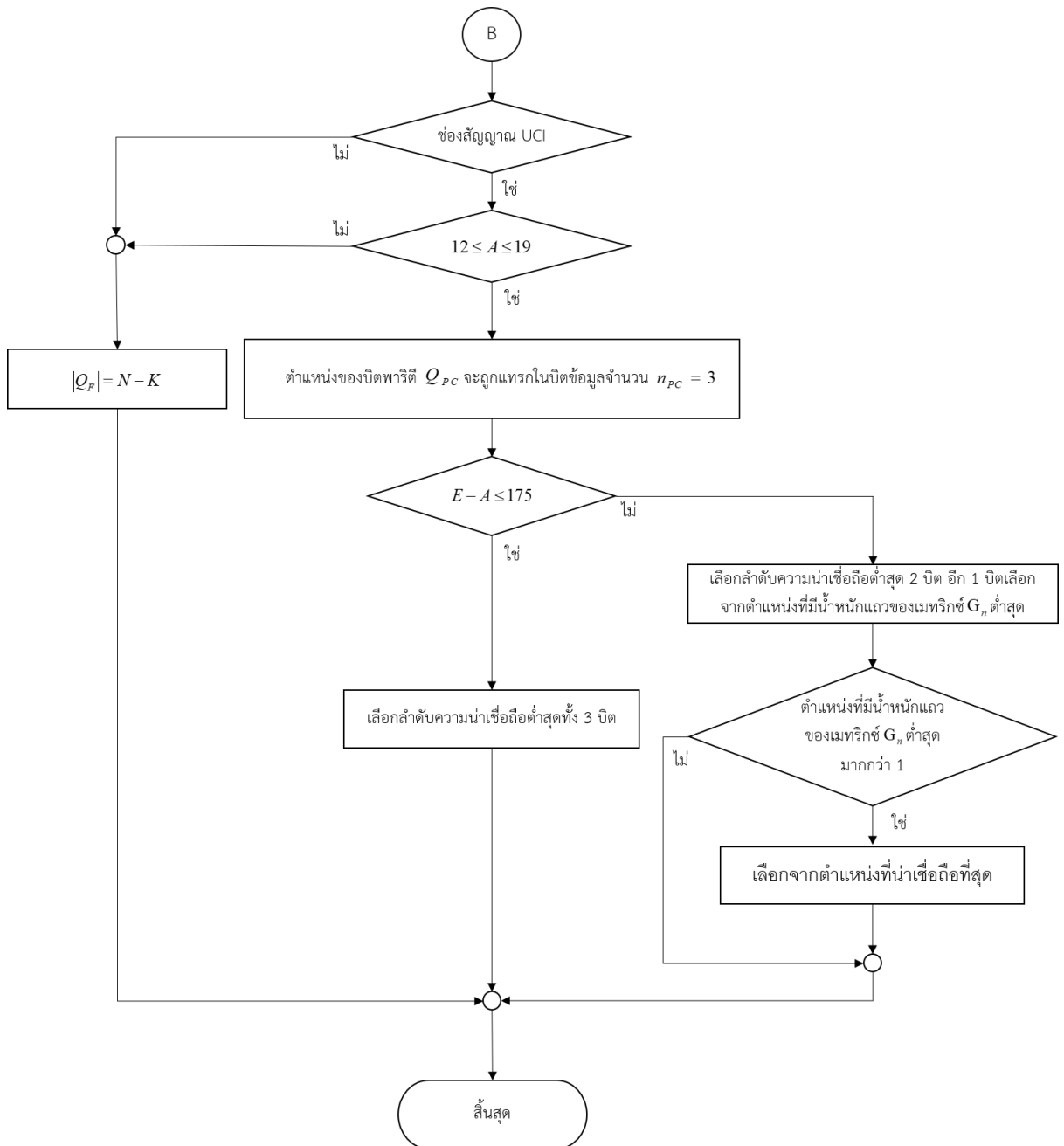
1) เลือกตำแหน่งบิตแชนแนลสำหรับการปรับอัตรารหัส หาก $K/E \leq 7/16$ จะปรับอัตรารหัสด้วยการฟังก์เจอร์ ซึ่งจะกำหนดบิตแชนแนลที่ U ตำแหน่งแรก ในกรณีอื่นจะปรับอัตรารหัสด้วยการช็อดเทน ซึ่งจะกำหนดบิตแชนแนลที่ U ตำแหน่งสุดท้าย โดย U คำนวณได้ดังสมการที่ 16

2) หากจำนวนบิตแชนแนลสำหรับการปรับอัตราหัสยังไม่เกินจำนวนบิตแชนแนลที่มีได้ จะเลือกบิตแชนแนลเพิ่มเติมตามลำดับความน่าเชื่อถือช่องสัญญาณตามตารางที่ 2.12



รูปที่ 3.26 แผนผังการทำงานการลำดับช่องสัญญาณย่อย

จากนั้นสำหรับช่องสัญญาณ PUCCH ที่ความยาว $A \leq 19$ ระบบจะเติมบิตพาริตีเข้าไปจำนวน $N_{PC} = 3$ ตามแผนภาพการทำงานแสดงได้ดังรูปที่ 3.27 โดยในกรณี $E - A \leq 175$ บิตพาริตีทั้ง 3 บิตจะถูกเลือกจากลำดับความน่าเชื่อถือที่ต่ำที่สุด ส่วนในกรณี $E - A > 175$ บิตพาริตี 2 บิตจะถูกเลือกจากลำดับความน่าเชื่อถือที่ต่ำที่สุดและบิตพาริตีสุดท้ายจะถูกเลือกโดยตำแหน่งที่มีน้ำหนักแถวของเมทริกซ์สร้าง \mathbf{G} ต่ำที่สุดในตำแหน่งที่ไม่ใช่ตำแหน่งบิตแข่ง $Q \in Q_F$ หากมีตำแหน่งของน้ำหนักแถวมากกว่าหนึ่ง จะเลือกตำแหน่งน้ำหนักแถวต่ำสุดที่มีความน่าเชื่อถือที่สุด

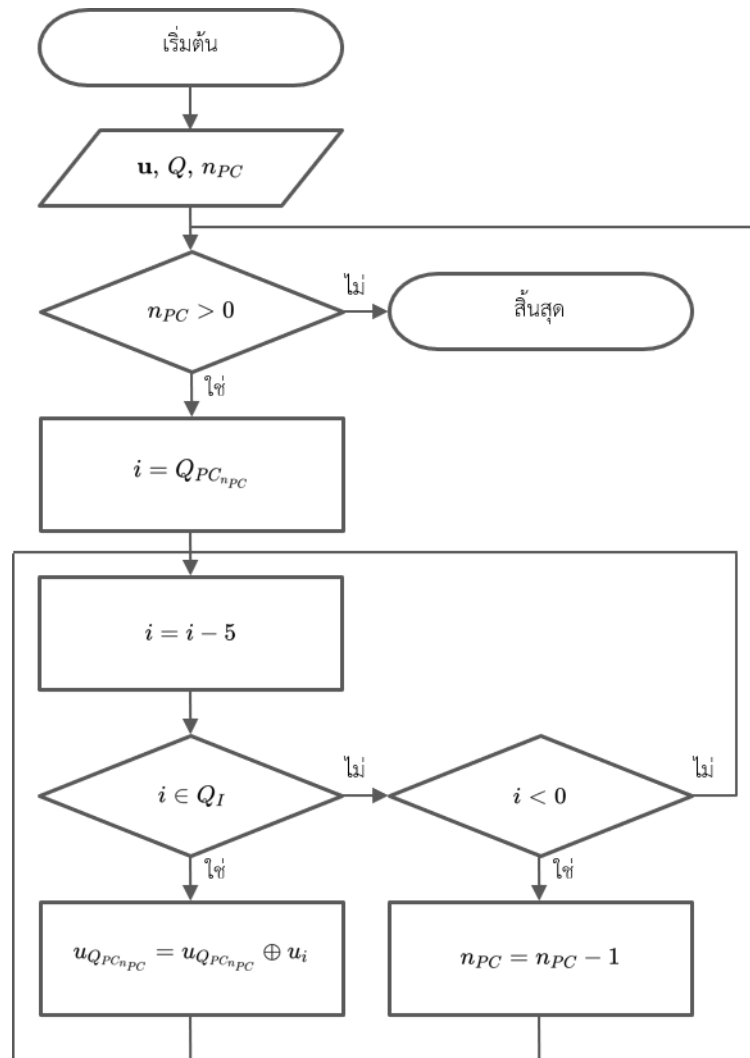


รูปที่ 3.27 แผนผังการทำงานการลำดับช่องสัญญาณย่อย (ต่อ)

3.2.2.3.3 การคำนวณบิตพาริตี

การคำนวณบิตพาริตีจะหาคำนวณค่าของบิตพาริตีหากกระบวนการก่อนหน้ามีการจองพื้นที่ให้บิตพาริตีจำนวน N_{PC} บิต ซึ่งโมดูลจะทำงานเฉพาะช่องสัญญาณ PUCCH ในกรณีที่ $A \leq 19$ สำหรับกรณีอื่นและช่องสัญญาณ PBCH และ PDCCH โมดูลจะไม่ทำงาน โดยบิตพาริตีที่อยู่ใน

ตำแหน่ง Q_{PC} จะมีความสัมพันธ์กับบิตก่อนหน้าถัดไปที่ละ 5 ตำแหน่งเรื่อย ๆ ไปยังจุดเริ่มต้น และ จะมีความสัมพันธ์เฉพาะบิตข้อมูลที่อยู่ในตำแหน่ง Q , โดยเฉพาะ แผนภาพการทำงานสามารถแสดง ได้ดังรูปที่ 3.28



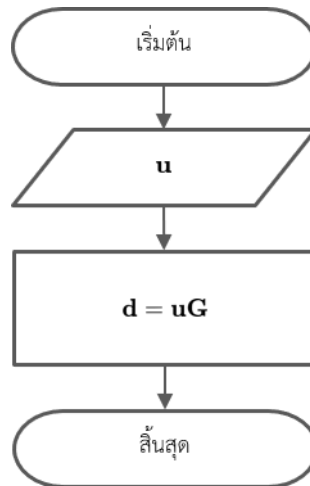
รูปที่ 3.28 แผนผังการทำงานการคำนวณบิตพาริตี

3.2.2.4 การเข้ารหัสโพลาร์ การแทรกสลับบิตเล็กน้อย และการปรับอัตรารหัส

โมดูลการเข้ารหัสโพลาร์ การแทรกสลับบิตเล็กน้อย และการปรับอัตรารหัสสามารถเข้าถึงได้ที่ ไฟล์ PolarEncoding.m

3.2.2.4.1 การเข้ารหัสโพลาร์

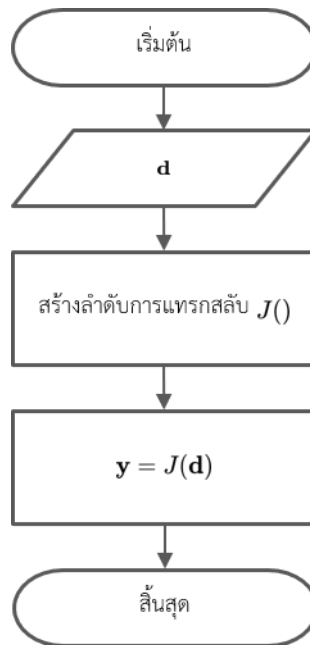
การเข้ารหัสโพลาร์จะทำการเข้ารหัสเวกเตอร์ \mathbf{u} ที่มีความยาว N ที่ผ่านกระบวนการก่อนหน้า โดย N ต้องมีค่าเท่ากับสองกำลังใด ๆ ได้ผลลัพธ์เป็นเวกเตอร์ \mathbf{d} ที่มีความยาว N เท่ากับเวกเตอร์ \mathbf{u} แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 3.29



รูปที่ 3.29 แผนผังการทำงานการเข้ารหัสโพลาร์

3.2.2.4.2 การแทรกสลับบล็อกย่อย

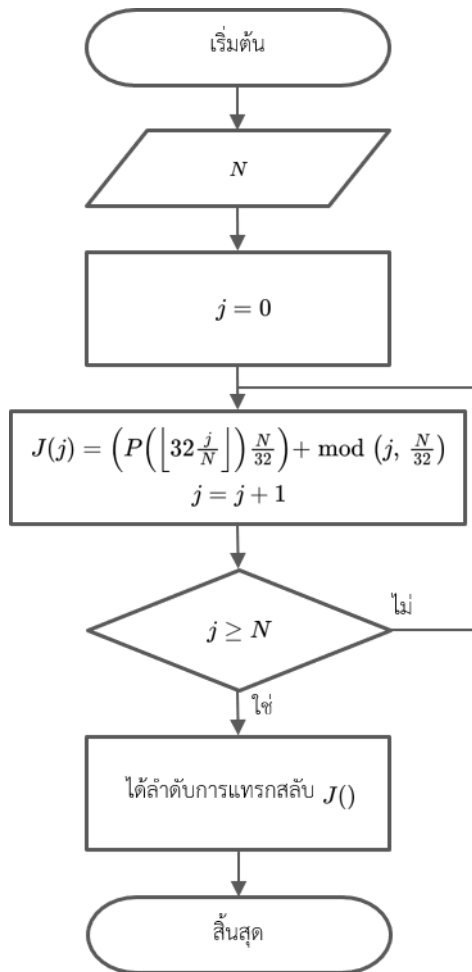
การแทรกสลับบล็อกย่อยจะนำเวกเตอร์ \mathbf{d} ที่ผ่านการเข้ารหัสโพลาร์มาแทรกสลับตำแหน่งบิต ได้ผลลัพธ์เป็นเวกเตอร์ $\mathbf{y} = J(\mathbf{d})$ ที่มีความยาว N ที่มีความยาวเท่ากับเวกเตอร์ \mathbf{d} แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 3.30



รูปที่ 3.30 แผนผังการทำงานการแทรกสลับล็อกย่อย

3.2.2.4.3 การสร้างลำดับการแทรกสลับล็อกย่อย

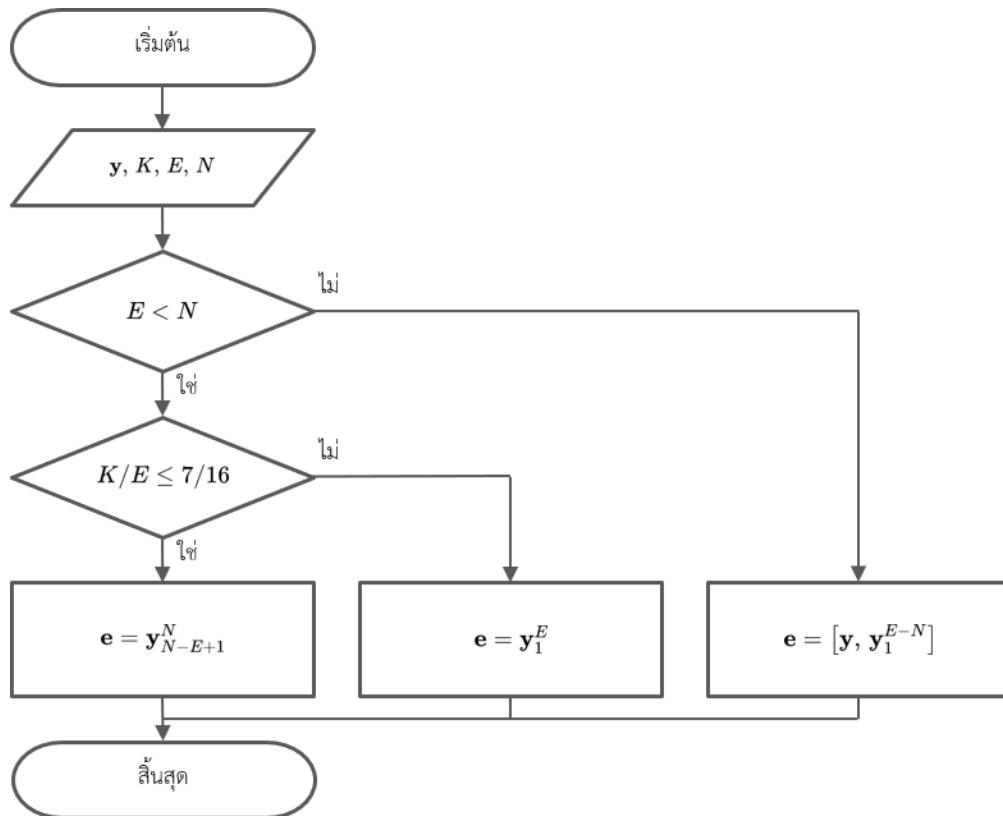
ลำดับการแทรกสลับล็อกย่อยถูกสร้างโดยอัลกอริทึมที่ได้ระบุไว้ตามมาตรฐานร่วมกับลำดับการแทรกสลัตามตารางที่ 12 ได้ผลลัพธ์เป็นลำดับการแทรกสลับล็อกย่อย $J()$ แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 3.31 ลำดับการแทรกสลับล็อกย่อยจะถูกคำนวณไว้ก่อนภายในกระบวนการลำดับช่องสัญญาณย่อยเพื่อใช้ในการกำหนดตำแหน่งบิตแน่ชัดสำหรับการปรับอัตรารหัสแบบฟังก์ชัน



รูปที่ 3.31 แผนผังการทำงานการสร้างลำดับการแทรกสลับบล็อกย่อย

3.2.2.4.4 การปรับอัตรารหัส

ระบบจะปรับความยาวคำรหัสแม้มให้มีมีความยาวเท่ากับความยาวคำรหัสที่ผู้ใช้งานต้องการ โดยเวกเตอร์ \mathbf{y} ที่มีความยาว N จากกระบวนการก่อนหน้าจะถูกปรับความยาวให้เท่ากับ E หากกรณี $E < N \wedge K/E \leq 7/16$ เวกเตอร์ \mathbf{y} จะถูกฟังก์ชันเซอร์ หากกรณี $E < N \wedge K/E > 7/16$ เวกเตอร์ \mathbf{y} จะถูกช้อนแทน และหากกรณี $E > N$ เวกเตอร์ \mathbf{y} จะถูกส่งซ้ำให้มีความยาวมากขึ้น โดยจะผลลัพธ์เป็นเวกเตอร์ \mathbf{e} ที่มีความยาว E แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 3.32



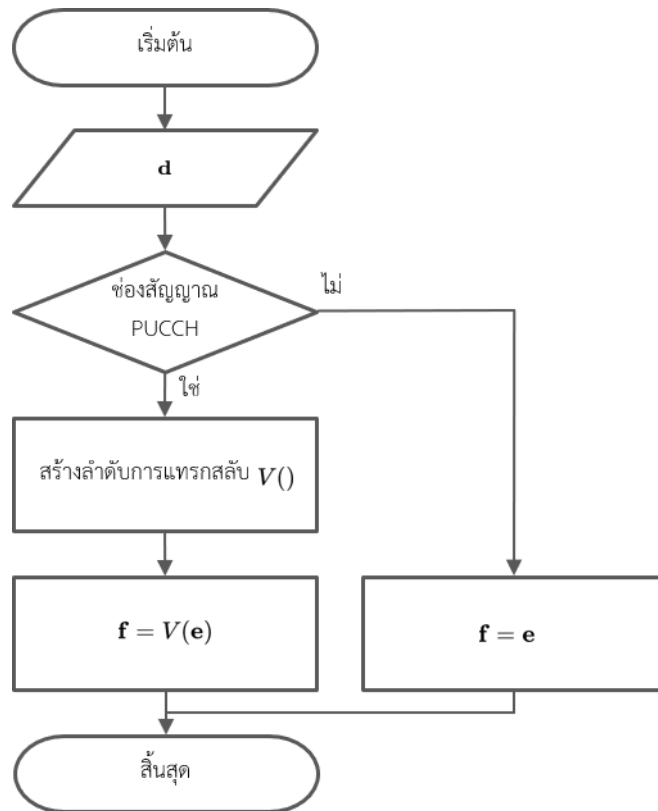
รูปที่ 3.32 แผนผังการทำงานการปรับอัตราหัส

3.2.2.5 การแทรกสลับบิตรหัสและการต่อบล็อกรหัส

โมดูลการแทรกสลับบิตรหัสและการต่อบล็อกรหัสสามารถเข้าถึงได้ที่ไฟล์ CodedBitInterleaver.m

3.2.2.5.1 การแทรกสลับบิตรหัส

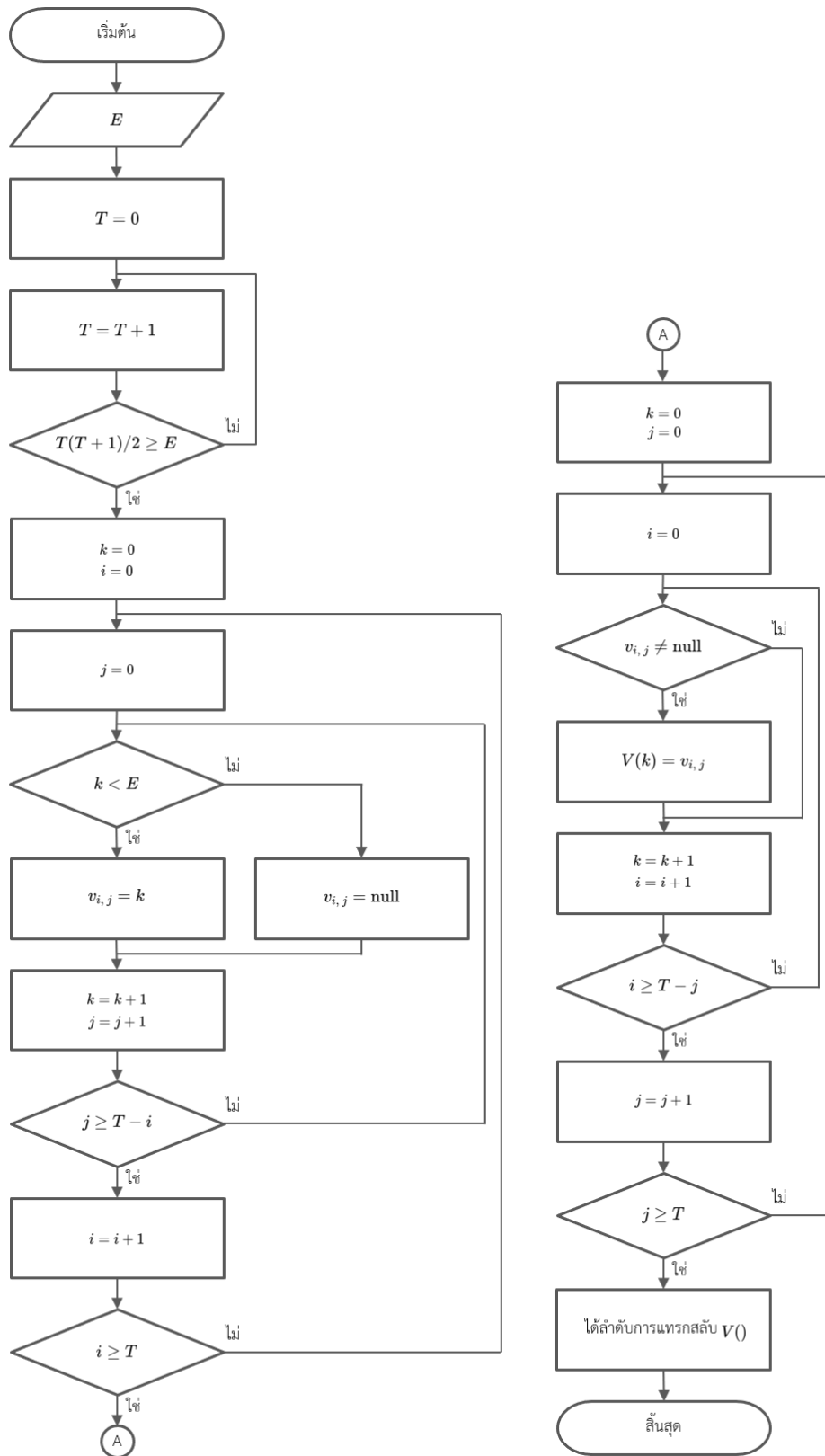
การแทรกสลับบิตรหัสจะนำเวกเตอร์ \mathbf{e} ที่ผ่านการปรับอัตราหัสมาแทรกสลับตำแหน่งบิต ได้ผลลัพธ์เป็นเวกเตอร์ $\mathbf{f} = V(\mathbf{e})$ ที่มีความยาว E ที่มีความยาวเท่ากับเวกเตอร์ \mathbf{e} แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 3.33 โมดูลจะทำงานเฉพาะช่องสัญญาณ PUCCH สำหรับช่องสัญญาณ PBCH และ PDCCH โมดูลจะไม่ทำงานและได้ผลลัพธ์เวกเตอร์ $\mathbf{f} = \mathbf{e}$



รูปที่ 3.33 แผนผังการทำงานการแทรกสลับบิตรหัส

3.2.2.5.2 การสร้างลำดับการแทรกสลับบิตรหัส

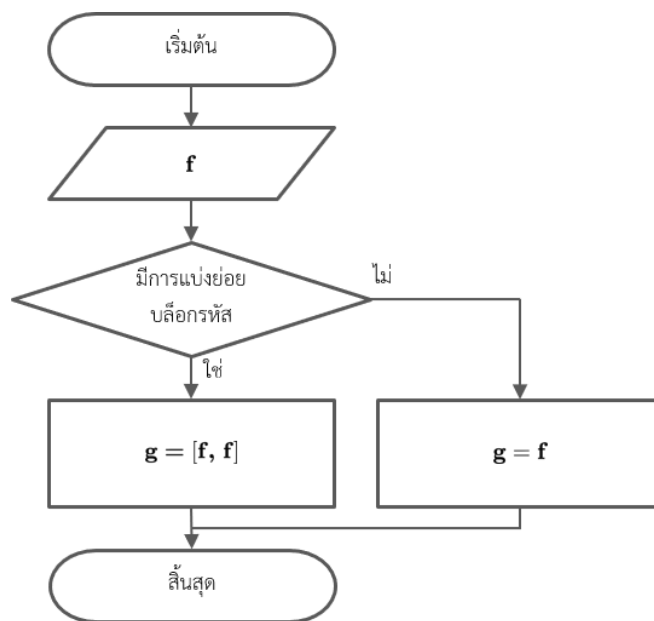
ลำดับการแทรกสลับบิตรหัสถูกสร้างโดยอัลกอริทึมที่ได้ระบุไว้ตามมาตรฐานร่วมกับลำดับการแทรกสลับที่คำนวณตามอัลกอริทึมดังรูปที่ 3.34 ได้ผลลัพธ์เป็นลำดับการแทรกสลับบิตรหัส $V()$ แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 3.34



รูปที่ 3.34 แผนผังการทำงานการสร้างลำดับการแทรกสลับบิตรหัส

3.2.2.5.3 การต่อบล๊อครหัส

หากกระบวนการระบบได้มีการแบ่งย่อยบล๊อครหัส โมดูลการต่อบล๊อครหัสจะทำงาน โดยจะนำเวกเตอร์ \mathbf{f} ทั้งสองเวกเตอร์ที่มีการแบ่งย่อยบล๊อครหัสก่อนหน้ามาต่อกันตามลำดับเดิม ได้ผลลัพธ์เป็นเวกเตอร์ \mathbf{g} ที่ความยาว $2E$ หากมีการแบ่งย่อยบล๊อครหัสก่อนหน้าและ E หากไม่มีการแบ่งย่อยบล๊อครหัสก่อนหน้า แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 3.35



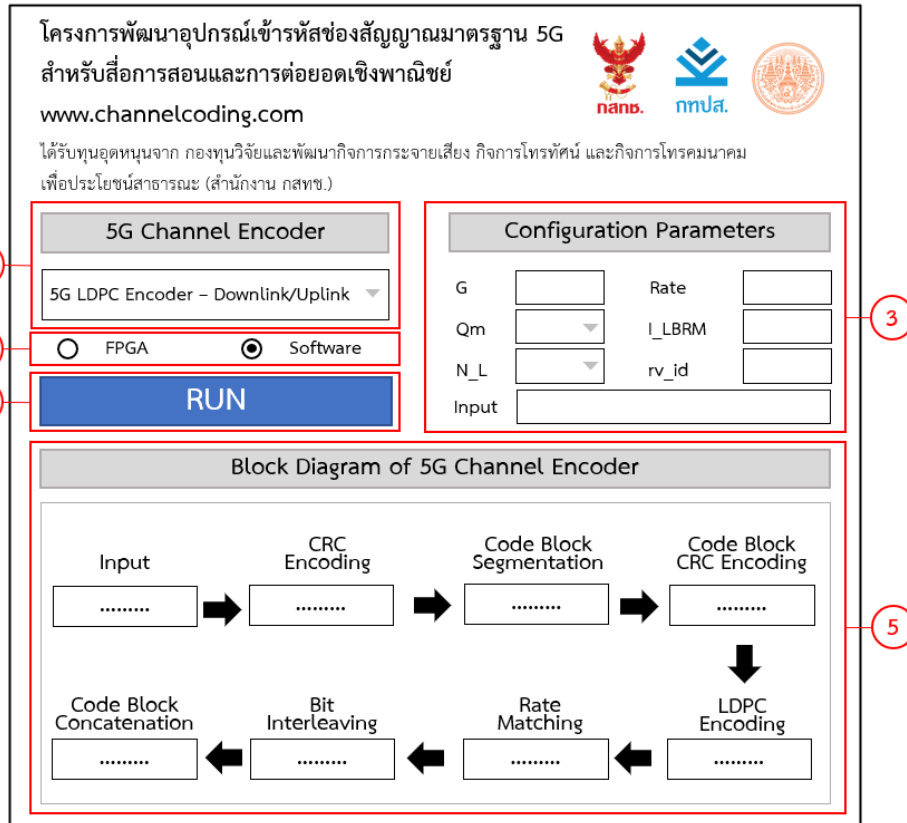
รูปที่ 3.35 แผนผังการทำงานการต่อบล๊อครหัส

3.3 รายละเอียดการออกแบบชุดซอฟต์แวร์การเรียนรู้การสื่อสารดิจิทัลที่มีการเข้ารหัส

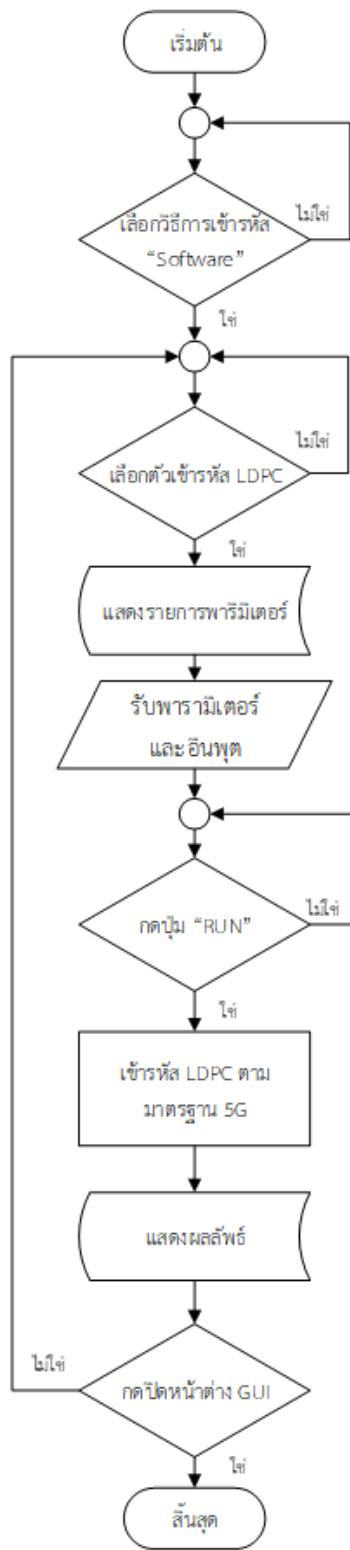
ช่องสัญญาณมาตรฐาน 5G

3.3.1 ส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อควบคุมวงจรเข้ารหัสช่องสัญญาณชนิด LDPC Codes

รายละเอียดการออกแบบส่วนต่อประสานกราฟิกผู้ใช้ (Graphical User Interface: GUI) เพื่อควบคุมวงจรเข้ารหัสช่องสัญญาณชนิด LDPC Codes ประกอบด้วย 5 ส่วน ได้แก่ การเลือกตัวเข้ารหัส การเลือกวิธีการเข้ารหัส การป้อนพารามิเตอร์และอินพุต การสั่งการเข้ารหัส และการแสดงผลลัพธ์ โดยหน้าต่างส่วนต่อประสานกราฟิกผู้ใช้ที่ได้จากการออกแบบและแผนผังการทำงานของโปรแกรมสำหรับรหัสช่องสัญญาณชนิด LDPC Codes ดังแสดงในรูปที่ 3.36 และ 3.37 ตามลำดับ



รูปที่ 3.36 หน้าต่างส่วนต่อประสานกราฟิกกับผู้ใช้สำหรับรหัสช่องสัญญาณชนิด LDPC Codes



รูปที่ 3.37 แผนผังการทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้สำหรับ รหัสช่องสัญญาณชนิด LDPC Codes

จากรูปที่ 3.36 และ 3.37 สามารถสรุปส่วนประกอบและการทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้สำหรับช่องสัญญาณชนิด LDPC Codes ได้ดังนี้

1) การเลือกตัวเข้ารหัส ในส่วนนี้ผู้ใช้สามารถเลือกชนิดของรหัสช่องสัญญาณที่ต้องการได้ โดยในกรณีรหัสช่องสัญญาณชนิด LDPC Codes การเข้ารหัสสัญญาณ Downlink และ Uplink จะมีกระบวนการเข้ารหัสเหมือนกัน

2) การเลือกวิธีการเข้ารหัส ในส่วนนี้ผู้ใช้สามารถเลือกวิธีการเข้ารหัสโดยใช้ชุดอุปกรณ์ (FPGA) หรือชุดซอฟต์แวร์ โดยในกรณีที่ต้องการใช้ชุดซอฟต์แวร์ผู้ใช้อาจเลือกที่ “Software”

3) การป้อนพารามิเตอร์และอินพุต เมื่อผู้ใช้เลือกชนิดของรหัสช่องสัญญาณในส่วนนี้จะแสดงรายการพารามิเตอร์ที่ผู้ใช้อาจจำเป็นต้องป้อนเพื่อใช้ในการเข้ารหัส พร้อมทั้งแสดงค่าเริ่มต้นของพารามิเตอร์และอินพุต โดยผู้ใช้สามารถปรับเปลี่ยนพารามิเตอร์และอินพุตตามต้องการได้

4) การสั่งการเข้ารหัส ในส่วนนี้เมื่อผู้ใช้ออกปุ่ม “RUN” ระบบจะนำพารามิเตอร์และอินพุตที่ผู้ใช้อป้อนในส่วนที่ 3 มาทำการเข้ารหัสช่องสัญญาณโดยใช้ซอฟต์แวร์การเข้ารหัส

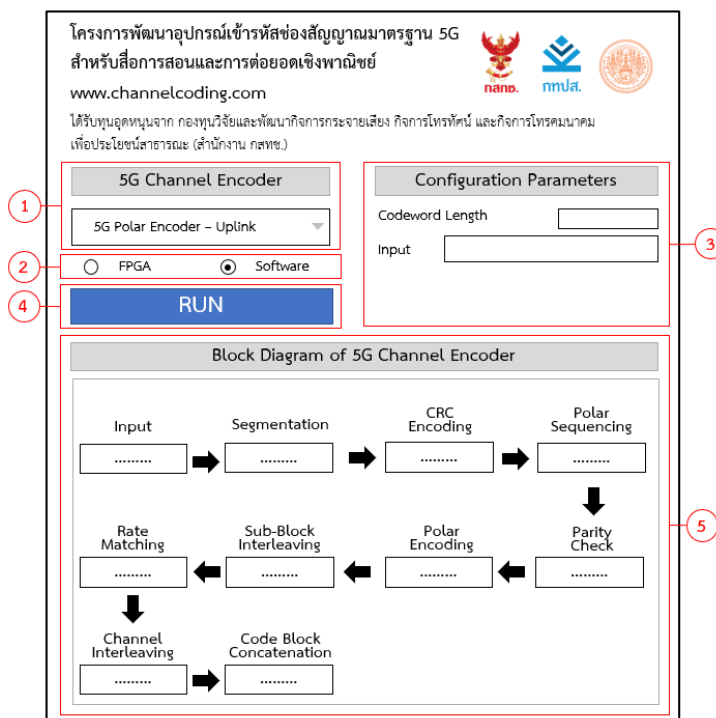
5) การแสดงผล หลังจากกระบวนการเข้ารหัสเสร็จสิ้น ในส่วนนี้จะแสดงผลที่ได้จากการเข้ารหัสในแต่ละขั้นตอนในรูปแบบของบล็อกไดอะแกรม

3.3.2 ส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อควบคุมวงจรเข้ารหัสช่องสัญญาณชนิด Polar Codes

รายละเอียดการออกแบบส่วนต่อประสานกราฟิกผู้ใช้เพื่อควบคุมวงจรเข้ารหัสช่องสัญญาณชนิด Polar Codes ประกอบด้วย 5 ส่วน ได้แก่ การเลือกตัวเข้ารหัส การเลือกวิธีการเข้ารหัส การป้อนพารามิเตอร์และอินพุต การสั่งการเข้ารหัส และการแสดงผล โดยหน้าต่างส่วนต่อประสานกราฟิกผู้ใช้สำหรับรหัสช่องสัญญาณชนิด Polar Code มี 3 รูปแบบตามลักษณะของสัญญาณ ซึ่งประกอบด้วย Downlink Uplink และ Broadcast ดังแสดงในรูปที่ 3.38-3.40 ตามลำดับ นอกจากนี้แผนผังการทำงานของโปรแกรมสำหรับ Polar Codes แสดงดังรูปที่ 3.41



รูปที่ 3.38 หน้าต่างส่วนต่อประสานกราฟิกกับผู้ใช้สำหรับ
 รหัสช่องสัญญาณชนิด Polar Codes (Downlink)



รูปที่ 3.39 หน้าต่างส่วนต่อประสานกราฟิกกับผู้ใช้สำหรับ
 รหัสช่องสัญญาณชนิด Polar Codes (Uplink)



รูปที่ 3.40 หน้าต่างส่วนต่อประสานกราฟิกกับผู้ใช้สำหรับ รหัสช่องสัญญาณชนิด Polar Codes (Broadcast)

จากรูปที่ 3.38-3.41 สามารถสรุปส่วนประกอบและการทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้สำหรับช่องสัญญาณชนิด Polar Codes ได้ดังนี้

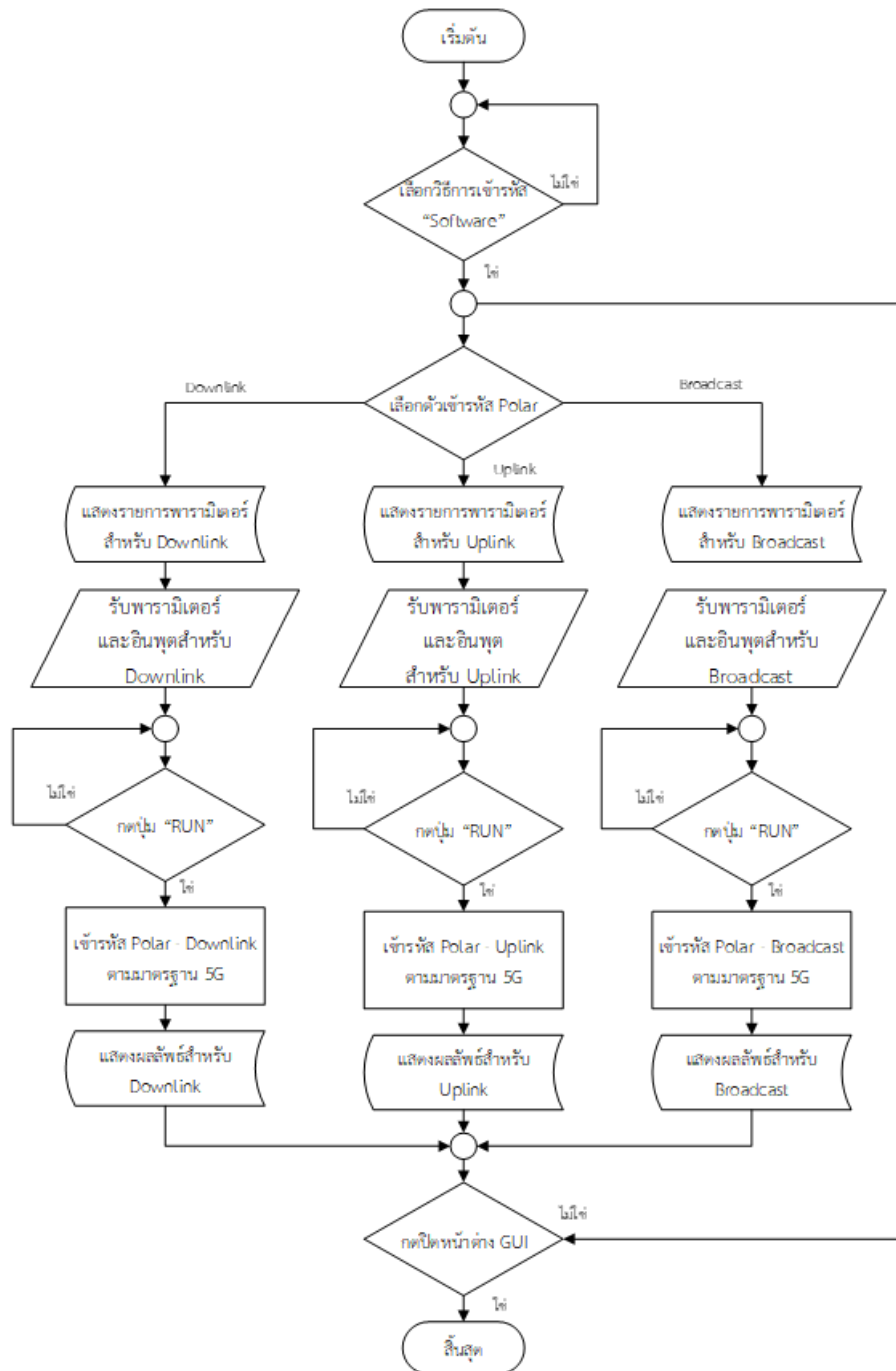
1) การเลือกตัวเข้ารหัส ในส่วนนี้ผู้ใช้สามารถเลือกชนิดของรหัสช่องสัญญาณที่ต้องการได้ โดยในกรณีรหัสช่องสัญญาณชนิด Polar Codes ผู้ใช้สามารถเลือกตัวเข้ารหัสตามลักษณะของสัญญาณได้ 3 รูปแบบ คือ Downlink Uplink และ Broadcast โดยแต่ละรูปแบบจะแสดงหน้าต่างส่วนต่อประสานกราฟิกกับผู้ใช้ที่แตกต่างกัน ดังแสดงในรูปที่ 3.38-3.40

2) การเลือกวิธีการเข้ารหัส ในส่วนนี้ผู้ใช้สามารถเลือกวิธีการเข้ารหัสโดยใช้ชุดอุปกรณ์ (FPGA) หรือชุดซอฟต์แวร์ โดยในกรณีที่ต้องการใช้ชุดซอฟต์แวร์ผู้ใช้อาจเลือกที่ “Software”

3) การป้อนพารามิเตอร์และอินพุต เมื่อผู้ใช้เลือกชนิดของรหัสช่องสัญญาณในส่วนนี้จะแสดงรายการพารามิเตอร์ที่ผู้ใช้จำเป็นต้องป้อนเพื่อใช้ในการเข้ารหัส พร้อมทั้งแสดงค่าเริ่มต้นของพารามิเตอร์และอินพุต โดยผู้ใช้สามารถปรับเปลี่ยนพารามิเตอร์และอินพุตตามต้องการได้

4) การสั่งการเข้ารหัส ในส่วนนี้เมื่อผู้ใช้กดปุ่ม “RUN” ระบบจะนำพารามิเตอร์และอินพุตที่ผู้ใช้ป้อนในส่วนที่ 3 มาทำการเข้ารหัสช่องสัญญาณโดยใช้ซอฟต์แวร์การเข้ารหัส

5) การแสดงผลลัพธ์ หลังจากกระบวนการเข้ารหัสเสร็จสิ้น ในส่วนนี้จะแสดงผลลัพธ์ที่ได้จากการเข้ารหัสในแต่ละขั้นตอนในรูปแบบของบล็อกไดอะแกรม



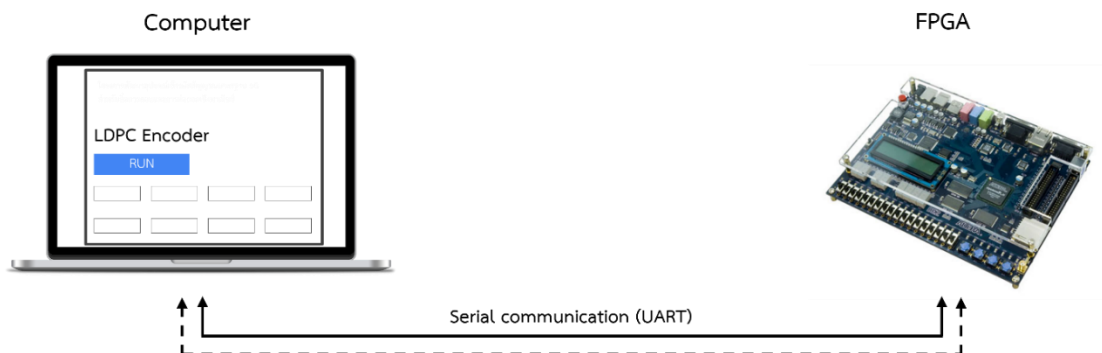
รูปที่ 3.41 แผนผังการทำงานของส่วนต่อประสานกราฟิกกับผู้ใช้สำหรับ

รหัสช่องสัญญาณชนิด Polar Codes

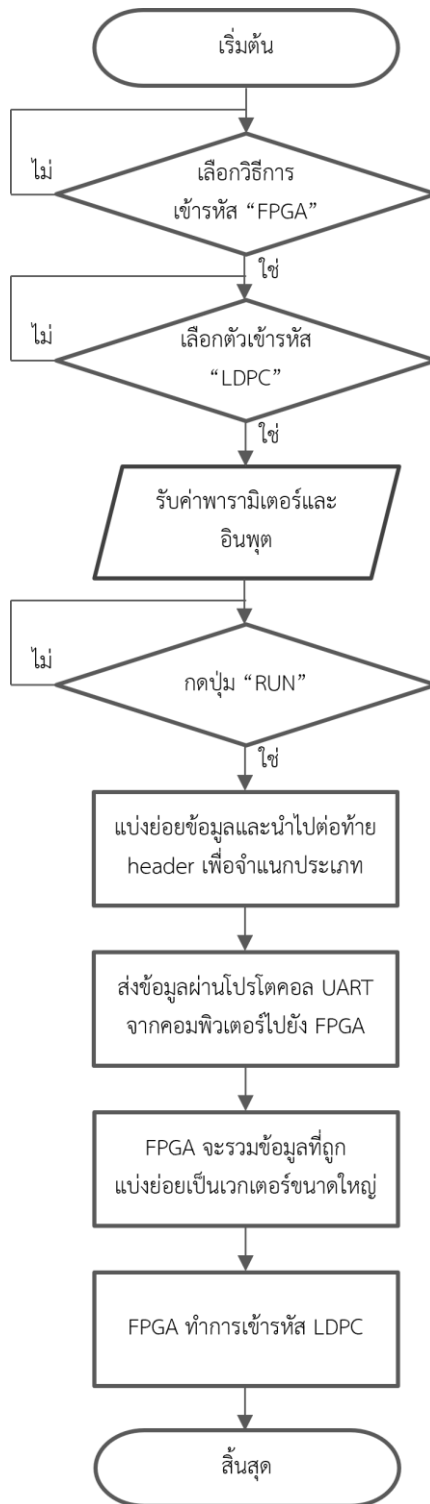
3.4 รายละเอียดการออกแบบชุดวงจรการเรียนรู้การสื่อสารดิจิทัลที่มีการเข้ารหัสช่องสัญญาณมาตรฐาน 5G

3.4.1 ชุดวงจรเข้ารหัสร่วมกับส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อการเข้ารหัสช่องสัญญาณชนิด LDPC Codes

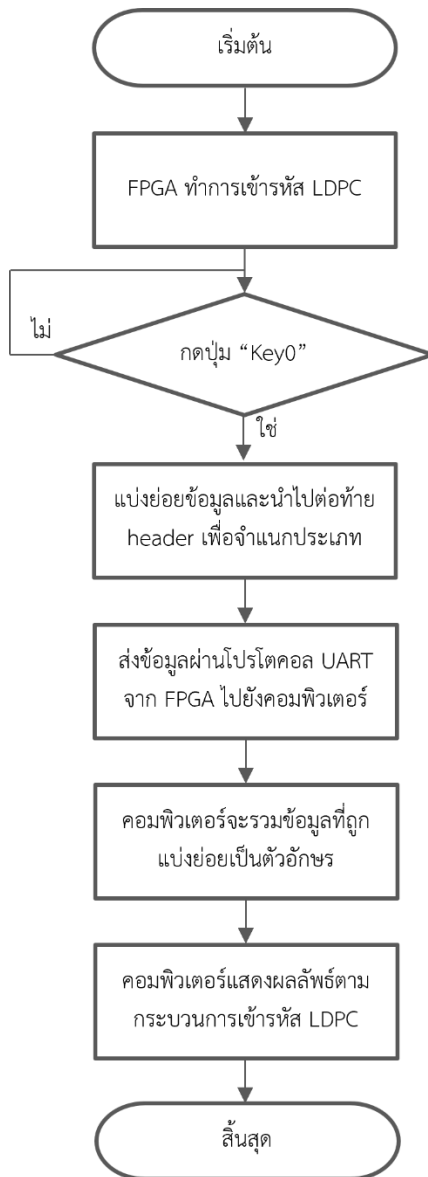
รายละเอียดการออกแบบชุดวงจรเข้ารหัสร่วมกับส่วนต่อประสานกราฟิกกับผู้ใช้ เพื่อเชื่อมต่อข้อมูลระหว่างผู้ใช้งานผ่านส่วนต่อประสานกราฟิกกับผู้ใช้กับวงจรเข้ารหัสช่องสัญญาณชนิด LDPC Codes คอมพิวเตอร์ ซึ่งแสดงส่วนต่อประสานกราฟิกกับผู้ใช้ จะรับข้อมูลการเข้ารหัสจากผู้ใช้งานที่ได้กรอก เพื่อจัดการข้อมูลและส่งต่อไปยังวงจรเข้ารหัส field programmable gate array หรือ FPGA ผ่านโปรโตคอล universal asynchronous receiver-transmitter หรือ UART แสดงชุดวงจรเข้ารหัส LDPC ที่ประกอบด้วยส่วนต่อประสานกราฟิกกับผู้ใช้และ FPGA ในระบบได้ดังภาพที่ 3.42 และแสดงแผนผังการทำงานได้ดังภาพที่ 3.43 และ 3.44



รูปที่ 3.42 ชุดวงจรการเรียนรู้การสื่อสารดิจิทัลที่มีการเข้ารหัสช่องสัญญาณมาตรฐาน 5G ประกอบด้วยส่วนต่อประสานกราฟิกกับผู้ใช้และ FPGA สำหรับรหัส LDPC



รูปที่ 3.43 แผนผังการทำงานการสื่อสารระหว่างคอมพิวเตอร์สู่วงจรรหัสเข้ารหัส FPGA สำหรับรหัส LDPC



รูปที่ 3.44 แผนผังการทำงานการสื่อสารระหว่างวงจรเข้ารหัส FPGA สู่คอมพิวเตอร์ สำหรับรหัส LDPC

จากรูปที่ 3.43 และ 3.44 จะแบ่งเป็นรายละเอียดการทำงานการสื่อสารระหว่างคอมพิวเตอร์สู่วงจรเข้ารหัส FPGA และระหว่างวงจรเข้ารหัส FPGA สู่คอมพิวเตอร์ ตามลำดับ การสื่อสารระหว่างคอมพิวเตอร์สู่วงจรเข้ารหัส FPGA สามารถสรุปขั้นตอนการทำงานได้ดังนี้

1) เลือกวิธีการเข้ารหัส “FPGA” โดยผู้ใช้งานจะเลือกผ่านส่วนต่อประสานกราฟิกกับผู้ใช้บนคอมพิวเตอร์ เมื่อเลือกตัวเลือกดังกล่าว คอมพิวเตอร์จะปรับการทำงานโดยการส่งข้อมูลไปให้วงจรเข้ารหัส FPGA ทำการเข้ารหัสแทนคอมพิวเตอร์เอง หากเลือกเป็น “Software”

2) เลือกตัวเข้ารหัส “5G LDPC Encoder - Downlink/Uplink” โดยผู้ใช้งานจะเลือกผ่านส่วนต่อประสานกราฟิกกับผู้ใช้บนคอมพิวเตอร์ เมื่อเลือกตัวเลือกดังกล่าว คอมพิวเตอร์จะเตรียมส่วนต่อประสานกราฟิกกับผู้ใช้ให้เหมาะสมกับรหัส LDPC โดยปรับส่วนการป้อนพารามิเตอร์และอินพุต และการแสดงผลเพื่อให้เหมาะสมรหัส LDPC ดังรูปที่ 3.36

3) ผู้ใช้งานกรอกค่าพารามิเตอร์และอินพุต สำหรับการเข้ารหัส LDPC

4) กดปุ่ม “RUN” เพื่อให้คอมพิวเตอร์เตรียมข้อมูลสำหรับการสื่อสารไปสู่วงจรเข้ารหัส FPGA โดยมีกระบวนการเตรียมข้อมูลตามขั้นตอนถัดไป

5) คอมพิวเตอร์จะแบ่งย่อยข้อมูลจากพารามิเตอร์และอินพุตเป็นบล็อกความยาวเท่ากับ 8 บิต และนำบล็อกข้อมูลย่อยไปต่อกับ header ที่คอมพิวเตอร์สร้างขึ้นสำหรับพารามิเตอร์และอินพุตนั้น ๆ หากข้อมูลบล็อกสุดท้ายมีความยาวไม่ถึง 8 บิต จะทำการเติมบิตศูนย์ต่อท้ายให้มีความยาวเท่ากับ 8 บิต

6) คอมพิวเตอร์ส่งข้อมูลไปยังวงจรเข้ารหัส FPGA ในรูปแบบซีเรียลผ่านโปรโตคอล UART โดยตั้งอัตราบิตต่อเวลาหรือ baud rate เท่ากับ 115200 บิตต่อวินาที ตั้งบิตเริ่มต้นเป็นบิต 1 ตั้งบิตสุดท้ายเป็นบิต 0 และไม่ใช้งานบิตพาริตี สำหรับการสื่อสารแบบซีเรียล

7) วงจรเข้ารหัส FPGA ที่ได้รับข้อมูลจากคอมพิวเตอร์ จะรับข้อมูลบล็อกละ 8 บิตเพื่อตรวจหา header เพื่อจำแนกข้อมูล จากนั้นจะนำบล็อกข้อมูลประเภทเดียวกันมาต่อกันเป็นเวกเตอร์ขนาดใหญ่เพื่อนำไปเข้ารหัสในขั้นตอนถัดไป

8) วงจรเข้ารหัส FPGA ทำการเข้ารหัส LDPC และการสื่อสารระหว่างวงจรเข้ารหัส FPGA สู่คอมพิวเตอร์ สามารถสรุปขั้นตอนการทำงานได้ดังนี้

1) FPGA ทำการเข้ารหัส LDPC

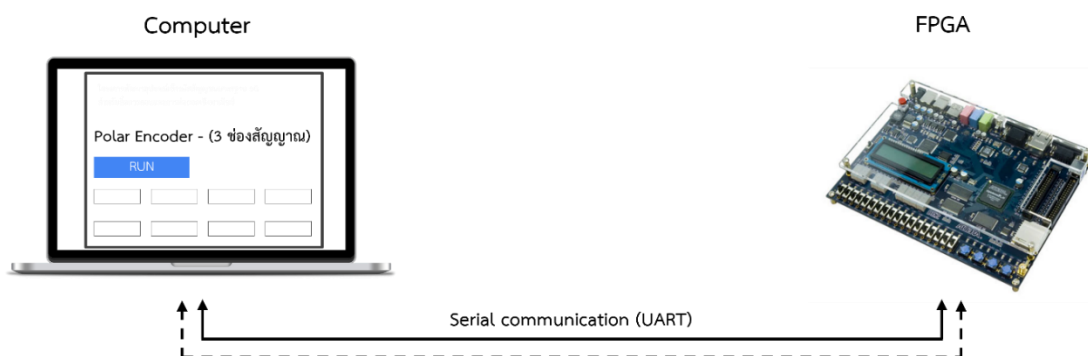
2) กดปุ่ม “Key0” เพื่อให้วงจรเข้ารหัส FPGA เตรียมข้อมูลสำหรับการสื่อสารไปสู่คอมพิวเตอร์ โดยมีกระบวนการเตรียมข้อมูลตามขั้นตอนถัดไป

3) วงจรเข้ารหัส FPGA จะแบ่งย่อยข้อมูลเป็นบล็อกความยาวเท่ากับ 8 บิต และนำบล็อกข้อมูลย่อยไปต่อกับ header ที่คอมพิวเตอร์สร้างขึ้นสำหรับข้อมูลประเภทนั้น ๆ หากข้อมูลบล็อกสุดท้ายมีความยาวไม่ถึง 8 บิต จะทำการเติมบิตศูนย์ต่อท้ายให้มีความยาวเท่ากับ 8 บิต

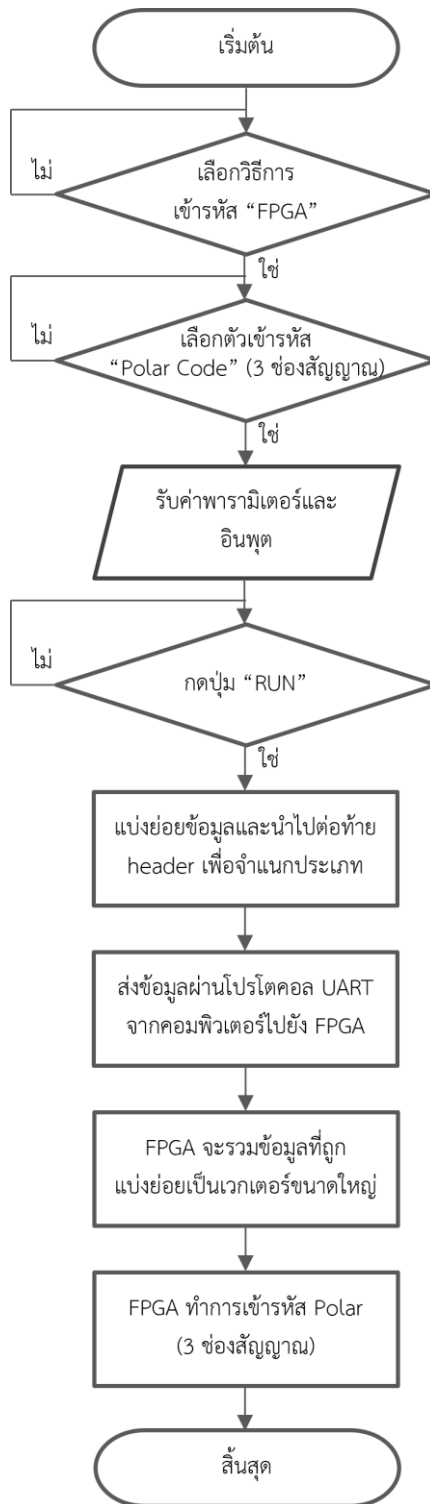
- 4) วงจรเข้ารหัส FPGA ส่งข้อมูลไปยังคอมพิวเตอร์ ในรูปแบบซีเรียลผ่านโปรโตคอล UART โดยตั้งอัตราบิตต่อเวลาหรือ baud rate เท่ากับ 115200 บิตต่อวินาที ตั้งบิตเริ่มต้นเป็นบิต 1 ตั้งบิตสุดท้ายเป็นบิต 0 และไม่ใช้งานบิตพาริตี สำหรับการสื่อสารแบบซีเรียล
- 5) คอมพิวเตอร์จะรวมข้อมูลข้อมูลที่ถูกระบุแบ่งย่อยเป็นตัวอักษรขนาดใหญ่สำหรับการแสดงผล โดยจะรับข้อมูลบล็อกละ 8 บิตเพื่อตรวจหา header เพื่อจำแนกข้อมูล
- 6) คอมพิวเตอร์แสดงผลลัพธ์ตามกระบวนการเข้ารหัส LDPC

3.4.2 ชุดวงจรเข้ารหัสร่วมกับส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อการเข้ารหัสช่องสัญญาณชนิด Polar Codes

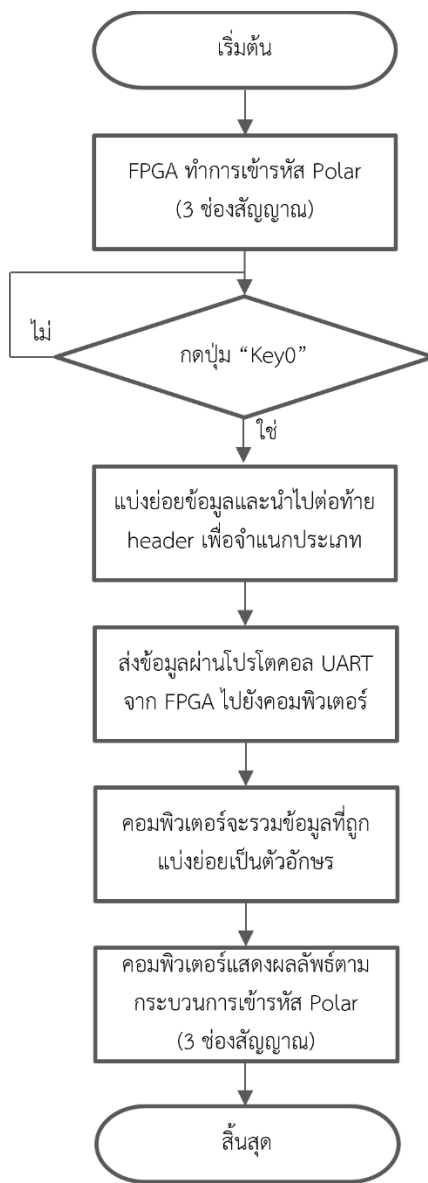
รายละเอียดการออกแบบชุดวงจรเข้ารหัสร่วมกับส่วนต่อประสานกราฟิกกับผู้ใช้ เพื่อเชื่อมต่อข้อมูลระหว่างผู้ใช้งานผ่านส่วนต่อประสานกราฟิกกับผู้ใช้กับวงจรเข้ารหัสช่องสัญญาณชนิด Polar Codes คอมพิวเตอร์ ซึ่งแสดงส่วนต่อประสานกราฟิกกับผู้ใช้ จะรับข้อมูลการเข้ารหัสจากผู้ใช้งานที่ได้กรอก เพื่อจัดการข้อมูลและส่งต่อไปยังวงจรเข้ารหัส field programmable gate array หรือ FPGA ผ่านโปรโตคอล universal asynchronous receiver-transmitter หรือ UART แสดงชุดวงจรเข้ารหัส Polar สำหรับทั้ง 3 ช่องสัญญาณ broadcast downlink และ uplink ที่ประกอบด้วยส่วนต่อประสานกราฟิกกับผู้ใช้และ FPGA ในระบบได้ดังภาพที่ 3.45 และแสดงแผนผังการทำงานได้ดังภาพที่ 3.46 และ 3.47



รูปที่ 3.45 ชุดวงจรการเรียนรู้การสื่อสารดิจิทัลที่มีการเข้ารหัสช่องสัญญาณมาตรฐาน 5G



รูปที่ 3.46 แผนผังการทำงานการสื่อสารระหว่างคอมพิวเตอร์สู่วงจรรหัสเข้ารหัส FPGA สำหรับรหัส Polar ทุกช่องสัญญาณ



รูปที่ 3.47 แผนผังการทำงานการสื่อสารระหว่างวงจรรหัสเข้ารหัส FPGA สู่คอมพิวเตอร์
สำหรับรหัส Polar ทุกช่องสัญญาณ

จากรูปที่ 3.46 และ 3.47 จะแบ่งเป็นรายละเอียดการทำงานการสื่อสารระหว่างคอมพิวเตอร์
สู่วงจรรหัสเข้ารหัส FPGA และระหว่างวงจรรหัสเข้ารหัส FPGA สู่คอมพิวเตอร์ ตามลำดับ การสื่อสารระหว่าง
คอมพิวเตอร์สู่วงจรรหัสเข้ารหัส FPGA สามารถสรุปขั้นตอนการทำงานได้ดังนี้

1) เลือกรหัสการเข้ารหัส “FPGA” โดยผู้ใช้งานจะเลือกผ่านส่วนต่อประสานกราฟิกกับผู้ใช้บนคอมพิวเตอร์ เมื่อเลือกตัวเลือกดังกล่าว คอมพิวเตอร์จะปรับการทำงานโดยการส่งข้อมูลไปให้วงจรเข้ารหัส FPGA ทำการเข้ารหัสแทนคอมพิวเตอร์เอง หากเลือกเป็น “Software”

2) เลือกตัวเข้ารหัส “5G Polar Encoder - Broadcast” “5G Polar Encoder - Downlink” หรือ “5G Polar Encoder - Uplink” ทั้ง 3 ช่องสัญญาณ โดยผู้ใช้งานจะเลือกผ่านส่วนต่อประสานกราฟิกกับผู้ใช้บนคอมพิวเตอร์ เมื่อเลือกตัวเลือกดังกล่าว คอมพิวเตอร์จะเตรียมส่วนต่อประสานกราฟิกกับผู้ใช้ให้เหมาะสมกับรหัส Polar ช่องสัญญาณนั้น ๆ โดยปรับส่วนการป้อนพารามิเตอร์และอินพุตและการแสดงผลลัพธ์ให้เหมาะสมรหัส Polar ช่องสัญญาณต่าง ๆ ดังรูปที่ 3.38-3.40

3) ผู้ใช้งานกรอกค่าพารามิเตอร์และอินพุต สำหรับการเข้ารหัส Polar ช่องสัญญาณต่าง ๆ

4) กดปุ่ม “RUN” เพื่อให้คอมพิวเตอร์เตรียมข้อมูลสำหรับการสื่อสารไปสู่วงจรเข้ารหัส FPGA โดยมีกระบวนการเตรียมข้อมูลตามขั้นตอนถัดไป

5) คอมพิวเตอร์จะแบ่งย่อยข้อมูลจากพารามิเตอร์และอินพุตเป็นบล็อกความยาวเท่ากับ 8 บิต และนำบล็อกข้อมูลย่อยไปต่อกับ header ที่คอมพิวเตอร์สร้างขึ้นสำหรับพารามิเตอร์และอินพุตนั้น ๆ หากข้อมูลบล็อกสุดท้ายมีความยาวไม่ถึง 8 บิต จะทำการเติมบิตศูนย์ต่อท้ายให้มีมีความยาวเท่ากับ 8 บิต

6) คอมพิวเตอร์ส่งข้อมูลไปยังวงจรเข้ารหัส FPGA ในรูปแบบซีเรียลผ่านโปรโตคอล UART โดยตั้งอัตราบิตต่อเวลาหรือ baud rate เท่ากับ 115200 บิตต่อวินาที ตั้งบิตเริ่มต้นเป็นบิต 1 ตั้งบิตสุดท้ายเป็นบิต 0 และไม่ใช้งานบิตพาริตี สำหรับการสื่อสารแบบซีเรียล

7) วงจรเข้ารหัส FPGA ที่ได้รับข้อมูลจากคอมพิวเตอร์ จะรับข้อมูลบล็อกละ 8 บิตเพื่อตรวจสอบ header เพื่อจำแนกข้อมูล จากนั้นจะนำบล็อกข้อมูลประเภทเดียวกันมาต่อกันเป็นเวกเตอร์ขนาดใหญ่เพื่อนำไปเข้ารหัสในขั้นตอนถัดไป

8) วงจรเข้ารหัส FPGA ทำการเข้ารหัส Polar ช่องสัญญาณต่าง ๆ และการสื่อสารระหว่างวงจรเข้ารหัส FPGA สู่คอมพิวเตอร์ สามารถสรุปขั้นตอนการทำงานได้ดังนี้

1) FPGA ทำการเข้ารหัส Polar ช่องสัญญาณต่าง ๆ

2) กดปุ่ม “Key0” เพื่อให้วงจรเข้ารหัส FPGA เตรียมข้อมูลสำหรับการสื่อสารไปสู่คอมพิวเตอร์ โดยมีกระบวนการเตรียมข้อมูลตามขั้นตอนถัดไป

3) วงจรเข้ารหัส FPGA จะแบ่งย่อยข้อมูลเป็นบล็อกความยาวเท่ากับ 8 บิต และนำบล็อกข้อมูลย่อยไปต่อกับ header ที่คอมพิวเตอร์สร้างขึ้นสำหรับข้อมูลประเภทนั้น ๆ หากข้อมูลบล็อกสุดท้ายมีความยาวไม่ถึง 8 บิต จะทำการเติมบิตศูนย์ต่อท้ายให้มีมีความยาวเท่ากับ 8 บิต

- 4) วงจรเข้ารหัส FPGA ส่งข้อมูลไปยังคอมพิวเตอร์ ในรูปแบบซีเรียลผ่านโปรโตคอล UART โดยตั้งอัตราบิตต่อเวลาหรือ baud rate เท่ากับ 115200 บิตต่อวินาที ตั้งบิตเริ่มต้นเป็นบิต 1 ตั้งบิตสุดท้ายเป็นบิต 0 และไม่ใช้งานบิตพาริตี สำหรับการสื่อสารแบบซีเรียล
- 5) คอมพิวเตอร์จะรวมข้อมูลข้อมูลที่ถูกแบ่งย่อยเป็นตัวอักษรขนาดใหญ่สำหรับการแสดงผล โดยจะรับข้อมูลบล็อกละ 8 บิตเพื่อตรวจหา header เพื่อจำแนกข้อมูล
- 6) คอมพิวเตอร์แสดงผลลัพธ์ตามกระบวนการเข้ารหัส Polar ช่องสัญญาณต่าง ๆ

บทที่ 4

ผลการวิจัย และการวิจารณ์ผล

4.1 ผลการทดสอบชุดซอฟต์แวร์และวงจรเข้ารหัสช่องสัญญาณชนิด LDPC codes

4.1.1 วิธีการใช้งานและผลการทดสอบชุดซอฟต์แวร์ MATLAB สำหรับการเข้ารหัสช่องสัญญาณ

4.1.1.1 การใช้งานซอฟต์แวร์ MATLAB และการกำหนดพารามิเตอร์เริ่มต้น

ในหัวข้อนี้จะกล่าวถึงการใช้งานชุดซอฟต์แวร์เข้ารหัสช่องสัญญาณชนิด LDPC codes ซึ่งสามารถดาวน์โหลดได้ที่เว็บไซต์ www.channelcoding.com ไฟล์ที่ได้มีชื่อว่า 5G_LDPC ซึ่งประกอบไปด้วยชุดซอฟต์แวร์ต่าง ๆ ดังแสดงในรูปที่ 4.1 ซอฟต์แวร์หลักที่สอดคล้องกับโมดูลรหัสช่องสัญญาณชนิด LDPC codes ประกอบไปด้วย 7 ซอฟต์แวร์ซึ่งแสดงในกรอบสีแดงเรียงลำดับการเรียงใช้งานหมายเลข 1 ถึงหมายเลข 7 ซอฟต์แวร์ test_5G_Transmission คือตัวอย่างการกำหนดพารามิเตอร์และเรียกใช้งานชุดซอฟต์แวร์ ส่วนซอฟต์แวร์ที่เหลือจะเป็นซอฟต์แวร์ย่อยที่จะถูกเรียกใช้โดยซอฟต์แวร์หลัก

| | | |
|---|------------------------|--------|
| 6 | BitInterleaving | M File |
| 7 | CodeBlockConcatenation | M File |
| | CRCEncoder0 | M File |
| 1 | CRCEncoder1 | M File |
| 3 | CRCEncoder2 | M File |
| | encodeLDPC5G | M File |
| 4 | LDPCEncoding | M File |
| | matrix5G | M File |
| | matrix5GH | M File |
| 5 | RateMatching | M File |
| 2 | Sementation | M File |
| | test_5G_Transmission | M File |

รูปที่ 4.1 ชุดซอฟต์แวร์เข้ารหัสช่องสัญญาณชนิด LDPC codes โดยใช้โปรแกรม MATLAB

รูปที่ 4.2 แสดงตัวอย่างการเรียกใช้งานซอฟต์แวร์และการกำหนดพารามิเตอร์ของไฟล์ชื่อ test_5G_Transmission โดยใช้โปรแกรม MATLAB ในซอฟต์แวร์จะประกอบไปด้วย 2 ส่วนคือ ส่วนที่กำหนดค่าพารามิเตอร์ และส่วนการเรียกใช้ซอฟต์แวร์ ในส่วนของการกำหนดพารามิเตอร์ ค่าพารามิเตอร์ที่สำคัญประกอบไปด้วยจำนวนบิตข้อมูล (A) เท่ากับ 16 บิต อัตรารหัส (R) เท่ากับ 0.5 เวกเตอร์ข้อมูล (a) เท่ากับ [1 0 1 1 1 1 1 0 1 0 1 1 0 0 1 1] ตรรกะของบัพเฟอร์ที่ถูกจำกัด (I_LBRM) เท่ากับ 0 ขนาดของบล็อกขนส่งข้อมูลสำหรับบัพเฟอร์ที่ถูกจำกัด (TBS_LBRM) เท่ากับค่าอนันต์ จำนวนการส่งข้อมูลเพิ่ม (rv_id) เท่ากับ 0 อันดับการมอดูเลชัน (Qm) เท่ากับ 4 จำนวนชั้นของการส่งข้อมูล (N_L) เท่ากับ 1 และจำนวนบิตที่ถูกส่งออก (G) เท่ากับ 100 บิต และส่วนที่ 2 คือ ส่วนของการเรียกใช้งานซอฟต์แวร์ โดยเริ่มจากการเรียกใช้ซอฟต์แวร์ CRCEncoder1 ในการเข้ารหัส CRC ให้กับข้อมูล

```
A = 16;
R = 0.5;
a = [1 0 1 1 1 1 1 0 1 0 1 1 0 0 1 1];
I_LBRM = 0;
TBS_LBRM = inf;
rv_id = 0;
Qm = 4;
N_L = 1;
G = 100;

[b, BG] = CRCEncoder1(a,R);
[br,K_prime,K,Z_c,i_LS,C,L] = Sementation(b,BG);
[c,Null] = CRCEncoder2(br,C,K_prime,L,K);
[d, N,Null_d] = LDPCEncoding(c,BG,K,Z_c,Null,i_LS);
[e,E_r]= RateMatching(d,Z_c,BG,C,N,Null_d,I_LBRM,TBS_LBRM,rv_id,Qm,N_L,G);
f = BitInterleaving(e,E_r,Qm,C);
g = CodeBlockConcatenation(f,G,E_r,C);
```

รูปที่ 4.2 ตัวอย่างการเรียกใช้งานซอฟต์แวร์และการกำหนดพารามิเตอร์ของไฟล์ชื่อ test_5G_Transmission

ถัดมาจะเรียกใช้ซอฟต์แวร์ Segmentation เพื่อแบ่งข้อมูลออกเป็นบล็อกรหัสย่อย ๆ หลังจากนั้นจะเรียกใช้ซอฟต์แวร์ CRCEncoder2 เพื่อเข้ารหัส CRC ให้กับบล็อกรหัส ถัดมาจะเรียกใช้ซอฟต์แวร์ LDPCEncoding ในการเข้ารหัส LDPC ให้กับบล็อกรหัส จากนั้นจะเรียกใช้ซอฟต์แวร์ RateMatching ในการเลือกส่งบางบิตซึ่งจำนวนบิตที่ส่งออกจะสอดคล้องกับอัตรารหัส ถัดมาจะเรียกใช้ซอฟต์แวร์ BitInterleaving เพื่อสลับตำแหน่งบิตข้อมูลซึ่งจะช่วยไม่ให้บิตผิดพลาดกันจำนวนมาก และสุดท้ายจะ

เรียกใช้ซอฟต์แวร์ COdeBlockConcatenation ในการต่อเรียงบล็อกรหัสย่อยก่อนจะส่งต่อไปยัง
วงจรมอดูเลชันและส่งออก

4.1.1.2 โมดูลการเข้ารหัส CRC

```
b =
Columns 1 through 16
   1   0   1   1   1   1   1   0   1   0   1   1   0   0   1   1
Columns 17 through 32
   1   0   1   0   1   0   1   0   1   0   1   1   1   0   1   0
BG =
   2
```

รูปที่ 4.3 ผลลัพธ์การเข้ารหัส CRC โดยใช้โปรแกรม MATLAB

รูปที่ 4.3 แสดงผลลัพธ์การเข้ารหัส CRC ซึ่งให้เอาต์พุต 2 ค่าได้แก่เวกเตอร์คำรหัส (b) ขนาด
32 บิต 16 บิตแรกเป็นบิตข้อมูล และ 16 บิตหลังเป็นบิตพาริตี เนื่องจากจำนวนบิตข้อมูลน้อยกว่า
3824 บิตจึงเลือกใช้โพลีโนเมียลความยาว (L) 16 บิต และเอาต์พุตกราฟฐาน (BG) เลือกใช้ 2

4.1.1.3 โมดูลการแบ่งส่วนย่อยข้อมูล

```
br =
Columns 1 through 16
   1   0   1   1   1   1   1   0   1   0   1   1   0   0   1   1
Columns 17 through 32
   1   0   1   0   1   0   1   0   1   0   1   1   1   0   1   0
Columns 33 through 48
   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
Columns 49 through 60
   0   0   0   0   0   0   0   0   0   0   0   0
```

รูปที่ 4.4 ผลลัพธ์เวกเตอร์หลังจากการแบ่งส่วนย่อยข้อมูลโดยใช้โปรแกรม MATLAB

รูปที่ 4.4 แสดงผลลัพธ์เวกเตอร์หลังจากการแบ่งส่วนย่อยข้อมูล เนื่องจากข้อมูลมีความยาว (B) เท่ากับ 32 บิต ดังนั้นบล็อกรหัสจึงมีจำนวน 1 บล็อก 32 บิตแรกได้จากเวกเตอร์คำรหัส (b) ส่วนตำแหน่งบิตที่เหลือกำหนดให้เป็นบิตว่าง <NULL> โดยการเติมบิต 0 ส่วนเอาต์พุตของพารามิเตอร์อื่น ๆ แสดงดังรูปที่ 4.5 ประกอบไปด้วย ความยาวข้อมูลจริง (K_prime) ได้เท่ากับ 32 บิต ความยาวข้อมูลที่ต้องการ (K) ได้เท่ากับ 60 บิต อัตราขยายของเมทริกซ์ฐาน (Z_c) ได้เท่ากับ 6 ค่า i_LS ได้เท่ากับ 2 และจำนวนบล็อกรหัส (C) ได้เท่ากับ 1

```

K_prime =
    32

K =
    60

Z_c =
    6

i_LS =
    2

C =
    1

L =
    0

```

รูปที่ 4.5 ผลลัพธ์พารามิเตอร์อื่น ๆ หลังจากการแบ่งส่วนย่อยข้อมูลโดยใช้โปรแกรม MATLAB

4.1.1.4 โมดูลการเข้ารหัส CRC ของบล็อกรหัส

```
c =  
  
Columns 1 through 13  
    1     0     1     1     1     1     1     0     1     0     1     1     0  
  
Columns 14 through 26  
    0     1     1     1     0     1     0     1     0     1     0     1     0  
  
Columns 27 through 39  
    1     1     1     0     1     0     0     0     0     0     0     0     0  
  
Columns 40 through 52  
    0     0     0     0     0     0     0     0     0     0     0     0     0  
  
Columns 53 through 60  
    0     0     0     0     0     0     0     0
```

รูปที่ 4.6 ผลลัพธ์เวกเตอร์หลังจากการเข้ารหัส CRC ของบล็อกรหัสโดยใช้โปรแกรม MATLAB

รูปที่ 4.6 แสดงผลลัพธ์เวกเตอร์ (c) หลังจากการเข้ารหัส CRC ของบล็อกรหัส จากรูปจะพบว่าบิตทั้งหมดใน c จะเหมือนกันกับเวกเตอร์ b เนื่องจากจำนวนบล็อกข้อมูลจากการแบ่งส่วนย่อยเป็นบล็อกรหัสมีจำนวน 1 บล็อกจึงไม่มีการเข้ารหัส CRC อีกครั้ง

4.1.1.5 โมดูลการเข้ารหัส LDPC

```

d =
Columns 1 through 19
  0  0  1  1  1  0  1  0  1  0  1  0  1  0  1  1  1  0  1
Columns 20 through 38
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Columns 39 through 57
  0  0  0  0  0  0  0  0  0  1  1  0  1  1  1  1  1  0  1
Columns 58 through 76
  0  1  1  0  0  1  1  1  0  1  0  1  0  1  0  1  0  1  1
Columns 77 through 95
  1  0  1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Columns 96 through 114
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Columns 115 through 133
  1  1  0  0  0  1  0  0  1  0  1  0  0  1  0  0  0  1  1
Columns 134 through 152
  1  1  0  0  1  1  0  0  1  0  1  1  0  0  0  1  1  0  1
Columns 153 through 171
  0  1  1  1  1  0  1  0  0  1  1  0  0  1  1  0  0  1  1
Columns 172 through 190
  0  0  1  1  0  1  0  1  0  1  1  0  0  0  0  1  1  1  1
Columns 191 through 209
  1  0  1  1  1  0  1  1  1  1  1  0  0  0  0  1  1  0  0
Columns 210 through 228
  1  1  1  1  0  1  0  1  0  1  1  1  1  0  1  0  0  0  0
Columns 229 through 247
  1  0  1  0  0  1  1  1  1  1  1  0  1  0  0  1  1  0  1
Columns 248 through 266
  0  0  0  0  1  0  1  1  1  1  1  1  1  1  0  0  1  1  0
Columns 267 through 285
  1  0  1  0  0  1  1  1  1  1  1  0  1  0  1  1  1  0  1
Columns 286 through 300
  0  0  1  0  0  0  1  1  0  0  1  1  0  0  0

```

รูปที่ 4.7 ผลลัพธ์เวกเตอร์หลังจากการเข้ารหัส LDPC โดยใช้โปรแกรม MATLAB

รูปที่ 4.7 ผลลัพธ์เวกเตอร์หลังจากการเข้ารหัส LDPC จากรูปจะพบว่าเวกเตอร์ข้อมูลจำนวน 60 บิตหลังเข้ารหัสแล้วจะได้เวกเตอร์ค้ำรหัส (d) มีจำนวน 300 บิต

4.1.1.6 โมดูลการปรับอัตรารหัส

```
e =  
Columns 1 through 19  
    0    0    1    1    1    0    1    0    1    0    1    0    1    0    1    1    1    0    1  
Columns 20 through 38  
    1    1    0    1    1    1    1    1    0    1    0    1    1    0    0    1    1    1    0  
Columns 39 through 57  
    1    0    1    0    1    0    1    0    1    1    1    0    1    0    0    0    0    0    0  
Columns 58 through 76  
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0  
Columns 77 through 95  
    0    0    0    0    0    0    0    0    0    0    1    1    0    0    0    1    0    0    1  
Columns 96 through 100  
    0    1    0    0    1
```

รูปที่ 4.8 ผลลัพธ์เวกเตอร์หลังจากการปรับอัตรารหัสโดยใช้โปรแกรม MATLAB

รูปที่ 4.8 ผลลัพธ์เวกเตอร์หลังจากการปรับอัตรารหัส จากรูปจะพบว่าเวกเตอร์ e จะมีจำนวนบิต 100 เนื่องจากเราเลือกบิตที่จะส่งออกไปจำนวน 100 บิต

4.1.7 โมดูลการแทรกสลับบิต

```
f =  
Columns 1 through 19  
    0    1    1    0    0    1    0    0    1    0    0    0    1    1    0    0    1    0    0  
Columns 20 through 38  
    0    0    1    0    0    1    1    0    0    0    0    0    0    1    0    0    0    0    1  
Columns 39 through 57  
    0    0    1    1    0    0    0    1    0    1    1    0    0    1    0    1    0    0    1  
Columns 58 through 76  
    0    0    0    1    1    0    0    1    0    0    1    0    1    0    0    1    0    0    0  
Columns 77 through 95  
    1    1    0    1    1    0    0    0    0    1    0    1    1    1    0    0    1    1    0  
Columns 96 through 100  
    0    1    0    0    1
```

รูปที่ 4.9 ผลลัพธ์เวกเตอร์หลังจากการแทรกสลับบิตโดยใช้โปรแกรม MATLAB

รูปที่ 4.9 ผลลัพธ์เวกเตอร์หลังจากการแทรกสลับบิต จากรูปจะพบว่าเวกเตอร์ f มีจำนวน 100 บิตเท่าเดิมแต่จำนวนบิตมีการสลับตำแหน่งกัน

4.1.8 โมดูลการต่อเรียงบล็อกรหัส

```
g =
Columns 1 through 19
    0    1    1    0    0    1    0    0    1    0    0    0    1    1    0    0    1    0    0
Columns 20 through 38
    0    0    1    0    0    1    1    0    0    0    0    0    0    1    0    0    0    0    1
Columns 39 through 57
    0    0    1    1    0    0    0    1    0    1    1    0    0    1    0    1    0    0    1
Columns 58 through 76
    0    0    0    1    1    0    0    1    0    0    1    0    1    0    0    1    0    0    0
Columns 77 through 95
    1    1    0    1    1    0    0    0    0    1    0    1    1    1    0    0    1    1    0
Columns 96 through 100
    0    1    0    0    1
```

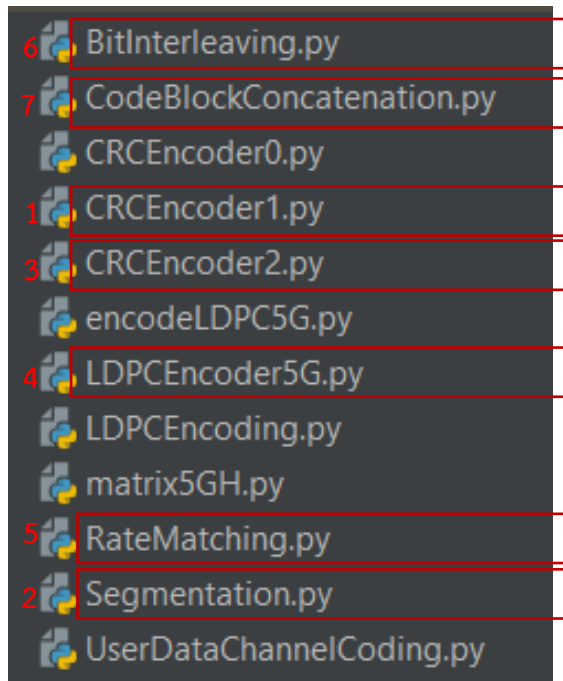
รูปที่ 4.10 ผลลัพธ์เวกเตอร์หลังจากการต่อเรียงบล็อกรหัสโดยใช้โปรแกรม MATLAB

รูปที่ 4.10 ผลลัพธ์เวกเตอร์หลังจากการแทรกสลับบิต จากรูปจะพบว่าเวกเตอร์มีจำนวน 100 บิตเท่าเดิม และตำแหน่งบิตเดิมเนื่องจากมีบล็อกรหัสเดียว

4.1.2 วิธีการใช้งานและผลการทดสอบชุดซอฟต์แวร์ PYTHON สำหรับการเข้ารหัสช่องสัญญาณ

4.1.2.1 การใช้งานซอฟต์แวร์ PYTHON และการกำหนดพารามิเตอร์เริ่มต้น

การใช้งานชุดซอฟต์แวร์ภาษา PYTHON สำหรับการเข้ารหัสช่องสัญญาณชนิด LDPC codes สามารถดาวน์โหลดได้ที่เว็บไซต์ www.channelcoding.com ไฟล์ที่ได้มีชื่อว่า 5G_LDPC ซึ่งประกอบไปด้วยชุดซอฟต์แวร์ต่าง ๆ ดังแสดงในรูปที่ 4.11 ซอฟต์แวร์หลักที่สอดคล้องกับโมดูลรหัสช่องสัญญาณชนิด LDPC codes ประกอบไปด้วย 7 ซอฟต์แวร์ซึ่งแสดงในกรอบสีแดงเรียงลำดับการเรียงใช้งานหมายเลข 1 ถึงหมายเลข 7 ซอฟต์แวร์ ส่วน UserDataChannelCoding เป็นตัวอย่างการกำหนดพารามิเตอร์และเรียกใช้งานชุดซอฟต์แวร์ย่อย สำหรับซอฟต์แวร์ที่เหลือจะเป็นซอฟต์แวร์ย่อยที่จะถูกเรียกใช้โดยซอฟต์แวร์หลัก



รูปที่ 4.11 ชุดซอฟต์แวร์เข้ารหัสช่องสัญญาณชนิด LDPC codes โดยใช้ภาษา PYTHON

รูปที่ 4.12 แสดงตัวอย่างการเรียกใช้งานซอฟต์แวร์ภาษา PYTHON และการกำหนดพารามิเตอร์ของไฟล์ชื่อ UserDataChannelCoding โดยใช้โปรแกรม PyCharm ในซอฟต์แวร์จะประกอบไปด้วย 3 ส่วนคือ ส่วนของการนำเข้าโมดูลย่อยสำหรับเรียกใช้ในโปรแกรมหลัก ส่วนที่กำหนดค่าพารามิเตอร์ และส่วนการเรียกใช้ซอฟต์แวร์ สำหรับการนำเข้าโมดูลย่อยสำหรับเรียกใช้ในโปรแกรมหลักจะประกอบไปด้วยการนำเข้าโมดูล CRCEncoder1, Segmentation, CRCEncoder2, LDPCEncoding, RateMatching, BitInterleaving และ CodeBlockConcatenation โดยใช้คำสั่ง import ในส่วนของการกำหนดพารามิเตอร์ ค่าพารามิเตอร์ที่สำคัญประกอบไปด้วยจำนวนบิตข้อมูล (A) เท่ากับ 16 บิต อัตรารหัส (R) เท่ากับ 0.5 เวกเตอร์ข้อมูล (a) เท่ากับ [1 0 1 1 1 1 0 1 0 1 1 0 0 1 1] ตรรกะของบัพเฟอร์ที่ถูกจำกัด (I_LBRM) เท่ากับ 0 ขนาดของบล็อกขนส่งข้อมูลสำหรับบัพเฟอร์ที่ถูกจำกัด (TBS_LBRM) เท่ากับค่าอนันต์


```

import numpy as np
import CRCEncoder1
import Segmentation
import CRCEncoder2
import LDPCEncoding
import RateMatching
import BitInterleaving
import CodeBlockConcatenation

A = 16
R = 0.5
a = np.array([1, 0, 1, 1, 1, 1, 1, 0, 1, 0, 1, 1, 0, 0, 1, 1])
I_LBRM = 0
TBS_LBRM = np.inf
rv_id = 0
Qm = 4
N_L = 1
G = 100

b, BG = CRCEncoder1.CRCEncoder11(a, R)
br, K_prime, K, Z_c, i_LS, C, L = Segmentation.Seg(b, BG)
c, Null = CRCEncoder2.CRCEncoder22(br, C, K_prime, L, K)
d, N, Null_d = LDPCEncoding.LDPCEncoder(c, BG, K, Z_c, Null, i_LS)
e, E_r = RateMatching.RM(d, Z_c, BG, C, N, Null_d, I_LBRM, TBS_LBRM, rv_id, Qm, N_L, G)
f = BitInterleaving.BI(e, E_r, Qm, C)
g = CodeBlockConcatenation.CBC(f, G, E_r, C)

```

รูปที่ 4.12 ตัวอย่างการเรียกใช้งานซอฟต์แวร์ภาษา PYTHON และการกำหนดพารามิเตอร์ของไฟล์ชื่อ UserDataChannelCoding

จำนวนการส่งข้อมูลเพิ่ม (rv_id) เท่ากับ 0 อันดับการมอดูเลชัน (Qm) เท่ากับ 4 จำนวนชั้นของการส่งข้อมูล (N_L) เท่ากับ 1 และจำนวนบิตที่ถูกส่งออก (G) เท่ากับ 100 บิต และส่วนที่ 2 คือส่วนของการเรียกใช้งานซอฟต์แวร์ โดยเริ่มจากการเรียกใช้ซอฟต์แวร์ CRCEncoder1 โดยมีฟังก์ชันย่อยภายในคือ CRCEncoder11 สำหรับการเข้ารหัส CRC ให้กับข้อมูล ถัดมาจะเรียกใช้ซอฟต์แวร์ Segmentation โดยมีฟังก์ชันย่อยภายในคือ Seg เพื่อแบ่งข้อมูลออกเป็นบล็อกรหัสย่อย ๆ หลังจากนั้นจะเรียกใช้ซอฟต์แวร์ CRCEncoder2 โดยมีฟังก์ชันย่อยภายในคือ CRCEncoder22 เพื่อเข้ารหัส CRC ให้กับบล็อกรหัส ถัดมาจะเรียกใช้ซอฟต์แวร์ LDPCEncoding โดยมีฟังก์ชันย่อยภายในคือ LDPCEncoder สำหรับการเข้ารหัส LDPC ให้กับบล็อกรหัส จากนั้นจะเรียกใช้ซอฟต์แวร์ RateMatching โดยมีฟังก์ชันย่อยภายในคือ RM สำหรับการเลือกส่งบางบิตซึ่งจำนวนบิตที่ส่งออกจะสอดคล้องกับอัตรารหัส ถัดมาจะเรียกใช้ซอฟต์แวร์ BitInterleaving โดยมีฟังก์ชันย่อยภายในคือ BI เพื่อสลับตำแหน่งบิตข้อมูลซึ่งจะช่วยไม่ให้บิตติดกันจำนวนมาก และสุดท้ายจะเรียกใช้ซอฟต์แวร์ CCodeBlockConcatenation โดยมีฟังก์ชันย่อยภายในคือ CBC สำหรับการต่อเรียงบล็อกรหัสย่อยก่อนจะส่งต่อไปยังวงจรมอดูเลชันและส่งออก

4.1.2.2 โมดูลการเข้ารหัส CRC

```
b = [1 0 1 1 1 1 1 0 1 0 1 1 0 0 1 1 1 0 1 0 1 0 1 0 1 0 1 1 1 0 1 0]
BG = 2
```

รูปที่ 4.13 ผลลัพธ์การเข้ารหัส CRC โดยใช้ภาษา PYTHON

รูปที่ 4.13 แสดงผลลัพธ์การเข้ารหัส CRC ซึ่งให้เอาต์พุต 2 ค่าได้แก่เวกเตอร์ค้ำรหัส (b) ขนาด 32 บิต 16 บิตแรกเป็นบิตข้อมูล และ 16 บิตหลังเป็นบิตพาริตี เนื่องจากจำนวนบิตข้อมูลน้อยกว่า 3824 บิตจึงเลือกใช้โพลีโนเมียลความยาว (L) 16 บิต และเอาต์พุตกราฟฐาน (BG) เลือกใช้ 2

4.1.2.3 โมดูลการแบ่งส่วนย่อยข้อมูล

```
br = [[1. 0. 1. 1. 1. 1. 1. 0. 1. 0. 1. 1. 0. 0. 1. 1. 1. 0. 1. 0. 1. 0. 1. 0.
1. 0. 1. 1. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

รูปที่ 4.14 ผลลัพธ์เวกเตอร์หลังจากการแบ่งส่วนย่อยข้อมูลโดยใช้ภาษา PYTHON

รูปที่ 4.14 แสดงผลลัพธ์เวกเตอร์หลังจากการแบ่งส่วนย่อยข้อมูล เนื่องจากข้อมูลมีความยาว (B) เท่ากับ 32 บิต ดังนั้นบล็อกรหัสจึงมีจำนวน 1 บล็อก 32 บิตแรกได้จากเวกเตอร์ค้ำรหัส (b) ส่วนตำแหน่งบิตที่เหลือกำหนดให้เป็นบิตว่าง <NULL> โดยการเติมบิต 0 ส่วนเอาต์พุตของพารามิเตอร์อื่น ๆ แสดงดังรูปที่ 4.15 ประกอบไปด้วย ความยาวข้อมูลจริง (K_prime) ได้เท่ากับ 32 บิต ความยาวข้อมูลที่ต้องการ (K) ได้เท่ากับ 60 บิต อัตราขยายของเมทริกซ์ฐาน (Z_c) ได้เท่ากับ 6 ค่า i_LS ได้เท่ากับ 2 และจำนวนบล็อกรหัส (C) ได้เท่ากับ 1

```
K_prime = 32.0
K = 60.0
Z_c = 6.0
i_LS = 1
C = 1
L = 0
```

รูปที่ 4.15 ผลลัพธ์พารามิเตอร์อื่น ๆ หลังจากการแบ่งส่วนย่อยข้อมูลโดยใช้ภาษา PYTHON

4.1.2.4 โมดูลการเข้ารหัส CRC ของบล็อกรหัส

```
c = [[1. 0. 1. 1. 1. 1. 1. 0. 1. 0. 1. 1. 0. 0. 1. 1. 1. 0. 1. 0. 1. 0. 1. 0.
1. 0. 1. 1. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```

รูปที่ 4.16 ผลลัพธ์เวกเตอร์หลังจากการเข้ารหัส CRC ของบล็อกรหัสโดยใช้ภาษา PYTHON

รูปที่ 4.16 แสดงผลลัพธ์เวกเตอร์ (c) หลังจากการเข้ารหัส CRC ของบล็อกรหัส จากรูปจะพบว่าบิตทั้งหมดใน c จะเหมือนกับกับเวกเตอร์ b เนื่องจากจำนวนบิตของข้อมูลจากการแบ่งส่วนย่อยเป็นบล็อกรหัสมีจำนวน 1 บล็อกจึงไม่มีการเข้ารหัส CRC อีกครั้ง

4.1.2.5 โมดูลการเข้ารหัส LDPC

```
d = [[0. 0. 1. 1. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 1. 1. 0. 1. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.
1. 0. 1. 1. 1. 1. 1. 0. 1. 0. 1. 1. 0. 0. 1. 1. 1. 0. 1. 0. 1. 0. 1. 0.
1. 0. 1. 1. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1.
0. 0. 1. 0. 1. 0. 0. 1. 0. 0. 0. 1. 1. 1. 1. 0. 0. 1. 1. 0. 0. 1. 0. 1.
1. 0. 0. 0. 1. 1. 0. 1. 0. 1. 1. 1. 1. 0. 1. 0. 0. 1. 1. 0. 0. 1. 1. 0.
0. 1. 1. 0. 0. 1. 1. 0. 1. 0. 1. 0. 1. 1. 0. 0. 0. 0. 0. 1. 1. 1. 1. 1. 0.
1. 1. 1. 0. 1. 1. 1. 0. 0. 0. 0. 1. 1. 0. 0. 0. 1. 1. 1. 1. 0. 1. 0.
1. 0. 1. 1. 1. 1. 0. 1. 0. 0. 0. 0. 1. 0. 1. 0. 0. 1. 1. 1. 1. 1. 0. 1.
0. 0. 1. 1. 0. 1. 0. 0. 0. 0. 0. 1. 0. 1. 1. 1. 1. 1. 1. 1. 1. 0. 0. 1.
1. 0. 1. 0. 1. 0. 0. 1. 1. 1. 1. 1. 0. 1. 0. 1. 1. 1. 0. 1. 0. 0. 1.
0. 0. 0. 1. 1. 0. 0. 1. 1. 0. 0. 0.]]
```

รูปที่ 4.17 ผลลัพธ์เวกเตอร์หลังจากการเข้ารหัส LDPC โดยใช้ภาษา PYTHON

รูปที่ 4.17 แสดงผลลัพธ์เวกเตอร์หลังจากการเข้ารหัส LDPC จากรูปจะพบว่าเวกเตอร์ข้อมูลจำนวน 60 บิตหลังเข้ารหัสแล้วจะได้เวกเตอร์เข้ารหัส (d) มีจำนวน 300 บิต

4.1.2.6 โมดูลการปรับอัตราหัส

```
e = [[0. 0. 1. 1. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 1. 1. 0. 1. 1. 1. 0. 1. 1.
1. 1. 1. 0. 1. 0. 1. 1. 0. 0. 1. 1. 1. 0. 1. 0. 1. 0. 1. 0. 1. 0. 1. 1.
1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0.
0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 1. 0. 0. 1. 0.
1. 0. 0. 1.]]
```

รูปที่ 4.18 ผลลัพธ์เวกเตอร์หลังจากการปรับอัตราหัสโดยใช้ภาษา PYTHON

รูปที่ 4.18 ผลลัพธ์เวกเตอร์หลังจากการปรับอัตราหัส จากรูปจะพบว่าเวกเตอร์ e จะมีจำนวนบิต 100 เนื่องจากเราเลือกบิตที่จะส่งออกไปจำนวน 100 บิต

4.1.2.7 โมดูลการแทรกสลับบิต

```
f = [[0. 1. 1. 0. 0. 1. 0. 0. 1. 0. 0. 0. 1. 1. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0.
1. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0. 1. 1. 0. 0. 0. 1. 0. 1.
1. 0. 0. 1. 0. 1. 0. 0. 1. 0. 0. 0. 1. 1. 0. 0. 1. 0. 0. 1. 0. 1. 0. 0.
1. 0. 0. 0. 1. 1. 0. 1. 1. 0. 0. 0. 0. 1. 0. 1. 1. 1. 0. 0. 1. 1. 0. 0.
1. 0. 0. 1.]]
```

รูปที่ 4.19 ผลลัพธ์เวกเตอร์หลังจากการแทรกสลับบิตโดยใช้ภาษา PYTHON

รูปที่ 4.19 ผลลัพธ์เวกเตอร์หลังจากการแทรกสลับบิต จากรูปจะพบว่าเวกเตอร์ f มีจำนวน 100 บิตเท่าเดิมแต่จำนวนบิตมีการสลับตำแหน่งกัน

4.1.2.8 โมดูลการต่อเรียงบล็อกรหัส

```
g = [[0. 1. 1. 0. 0. 1. 0. 0. 1. 0. 0. 0. 1. 1. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0.
1. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 0. 1. 1. 0. 0. 0. 1. 0. 1.
1. 0. 0. 1. 0. 1. 0. 0. 1. 0. 0. 0. 1. 1. 0. 0. 1. 0. 0. 1. 0. 1. 0. 0.
1. 0. 0. 0. 1. 1. 0. 1. 1. 0. 0. 0. 0. 1. 0. 1. 1. 1. 0. 0. 1. 1. 0. 0.
1. 0. 0. 1.]]
```

รูปที่ 4.20 ผลลัพธ์เวกเตอร์หลังจากการต่อเรียงบล็อกรหัสโดยใช้ภาษา PYTHON

รูปที่ 4.20 ผลลัพธ์เวกเตอร์หลังจากการแทรกสลับบิต จากรูปจะพบว่าเวกเตอร์มีจำนวน 100 บิตเท่าเดิม และตำแหน่งบิตเดิมเนื่องจากมีบล็อกรหัสเดียว

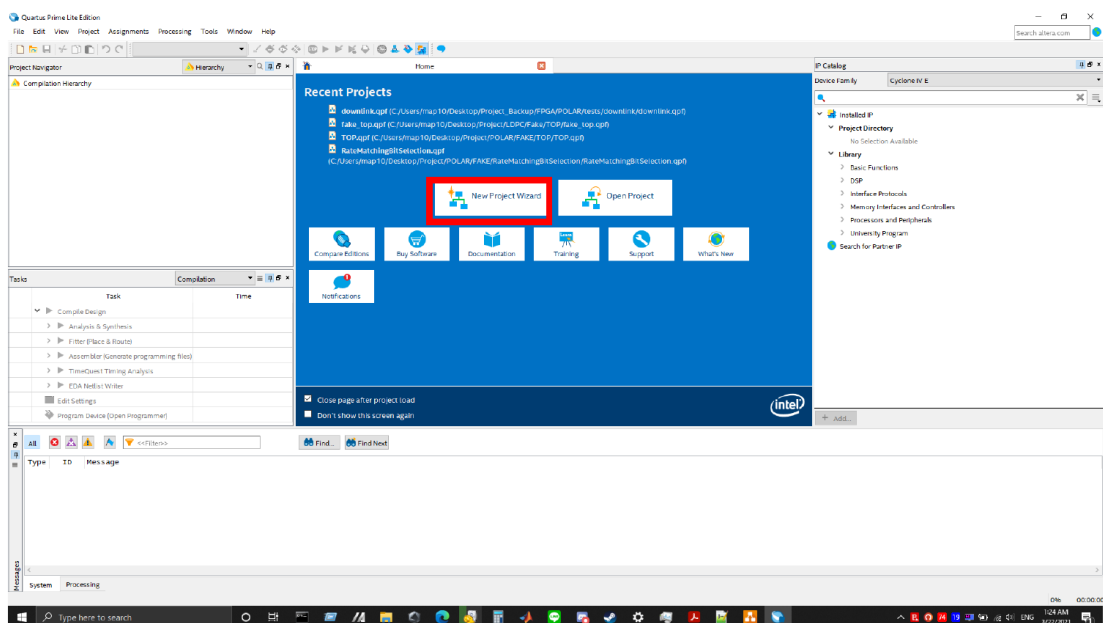
4.1.3 วิธีการใช้งานและผลการทดสอบวงจรเข้ารหัสช่องสัญญาณ

4.1.3.1 วิธีการใช้งานวงจรเข้ารหัสช่องสัญญาณ LDPC

วิธีการใช้งานวงจรเข้ารหัสช่องสัญญาณ LDPC ด้วยโปรแกรม Quartus Prime Lite Edition และโปรแกรม ModelSim-Altera มีดังนี้

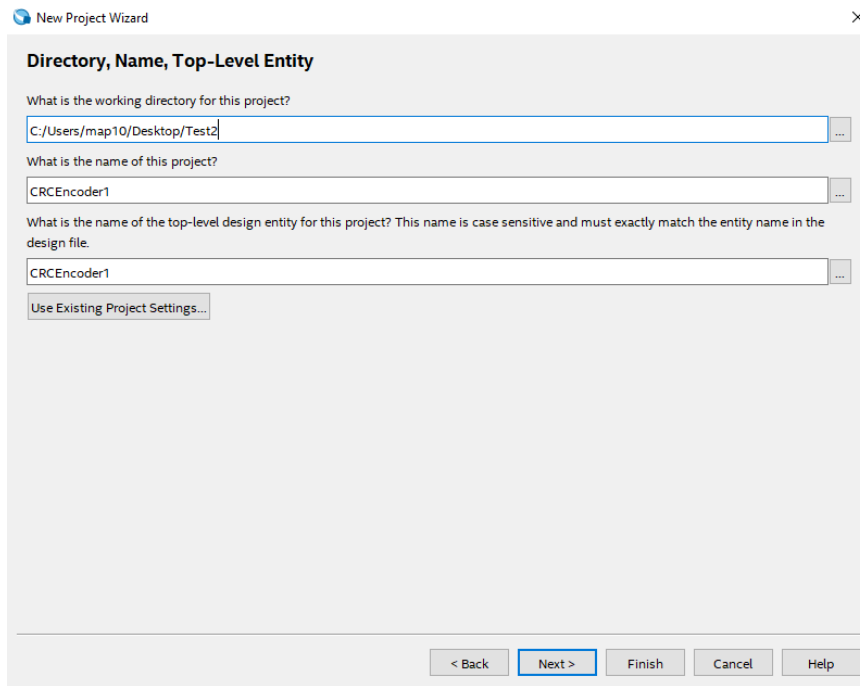
1) ทำการเปิดโปรแกรม Quartus Prime Lite Edition จะพบกับหน้าต่างเริ่มแรกของโปรแกรม

2) ทำการกดสร้างโปรเจคใหม่บนหน้าต่างเริ่มแรกของโปรแกรกดรูปที่ 4.21



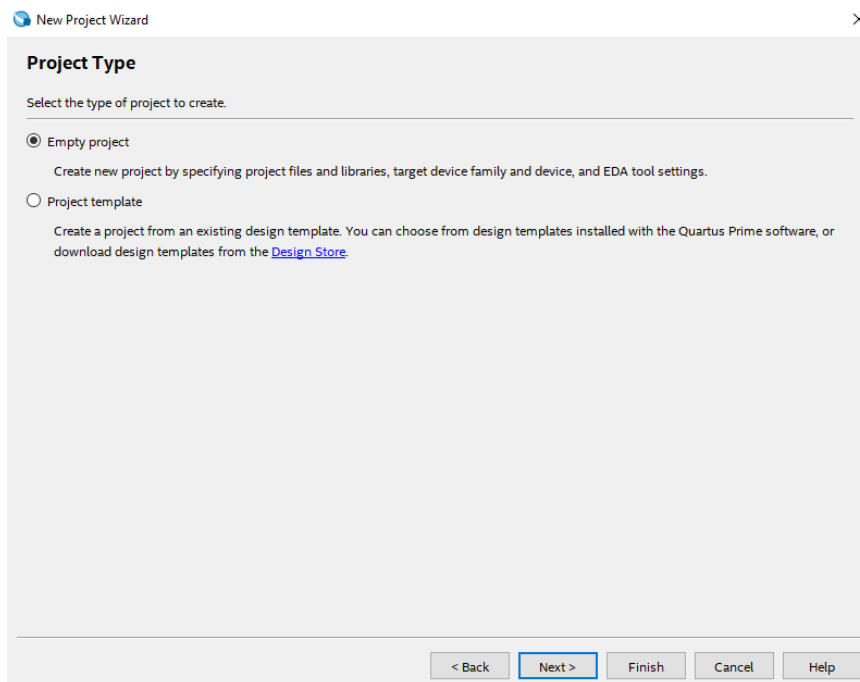
รูปที่ 4.21 แสดงการสร้างโปรเจคใหม่บนหน้าต่างเริ่มแรก

3) ทำการ Browse ตำแหน่งที่อยู่ของโปรเจคตามต้องการ ตั้งชื่อโปรเจคตามต้องการและตั้งชื่อ Top-Level Entity ตามชื่อ Entity ภายในไฟล์ .vhd หรือ .vhdl ที่ต้องการใช้งาน จากนั้นกด Next ดังรูปที่ 4.22



รูปที่ 4.22 หน้าต่างสำหรับกำหนดค่าเริ่มต้นของโปรเจกต์

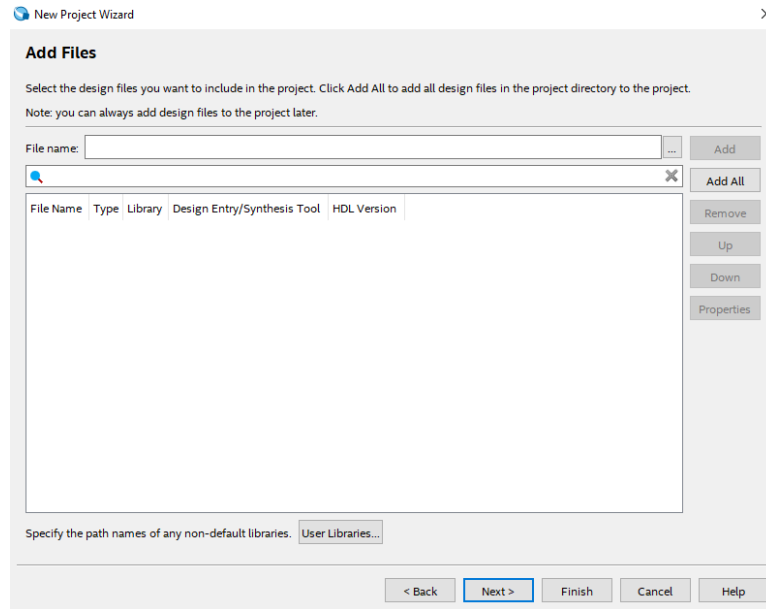
- 4) หน้าต่างนี้จะเป็นการเลือกประเภทของโปรเจกต์ ให้ทำการเลือกเป็น Empty project และกด Next ดังรูปที่ 4.23



รูปที่ 4.23 หน้าต่างสำหรับกำหนดประเภทของโปรเจกต์

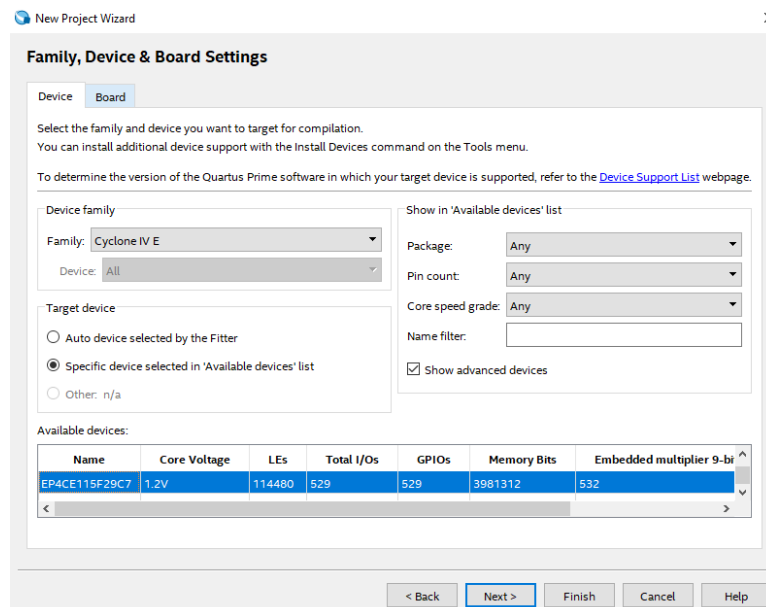
5) ทำการ Browse หาไฟล์ .vhd หรือ .vhdl ที่ต้องการใช้งาน จากนั้นกด Next ดังรูปที่

4.24



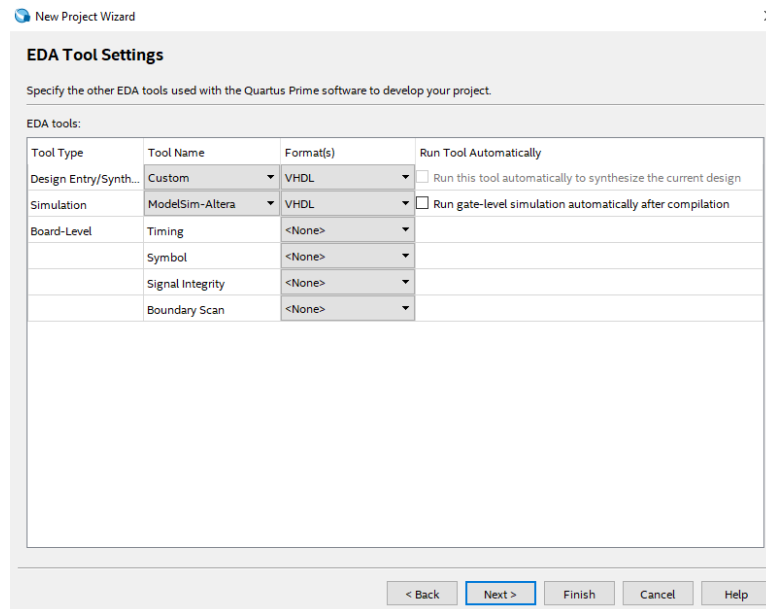
รูปที่ 4.24 หน้าต่างสำหรับเลือกไฟล์ของโปรเจค

6) ทำการเลือกอุปกรณ์ที่ต้องการใช้งาน ในช่อง Family ให้ทำการเลือกเป็น Cyclone IV E และให้ทำการเลือกอุปกรณ์ชื่อ EP4CE115F29C7 จากนั้นกด Next ดังรูปที่ 4.25



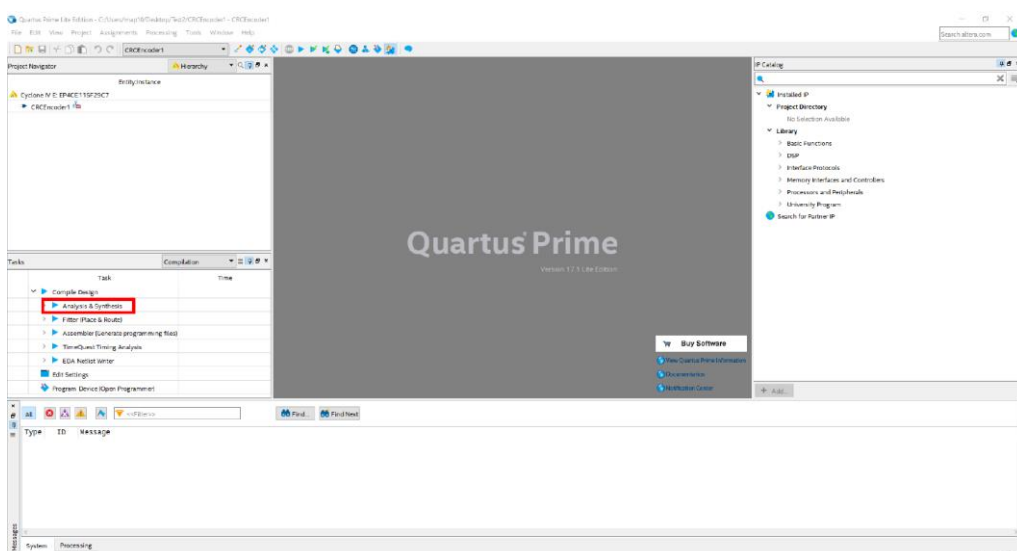
รูปที่ 4.25 หน้าต่างสำหรับเลือกอุปกรณ์ที่ใช้กับโปรเจค

7) ในช่อง Design Entry/Synthesis ให้เลือก Tool Name เป็น Custom แล้วเลือก Format(s) เป็น VHDL และในช่อง Simulation ให้เลือก Tool Name เป็น ModelSim-Altera แล้วเลือก Format(s) เป็น VHDL จากนั้นกด Finish ดังรูปที่ 4.26

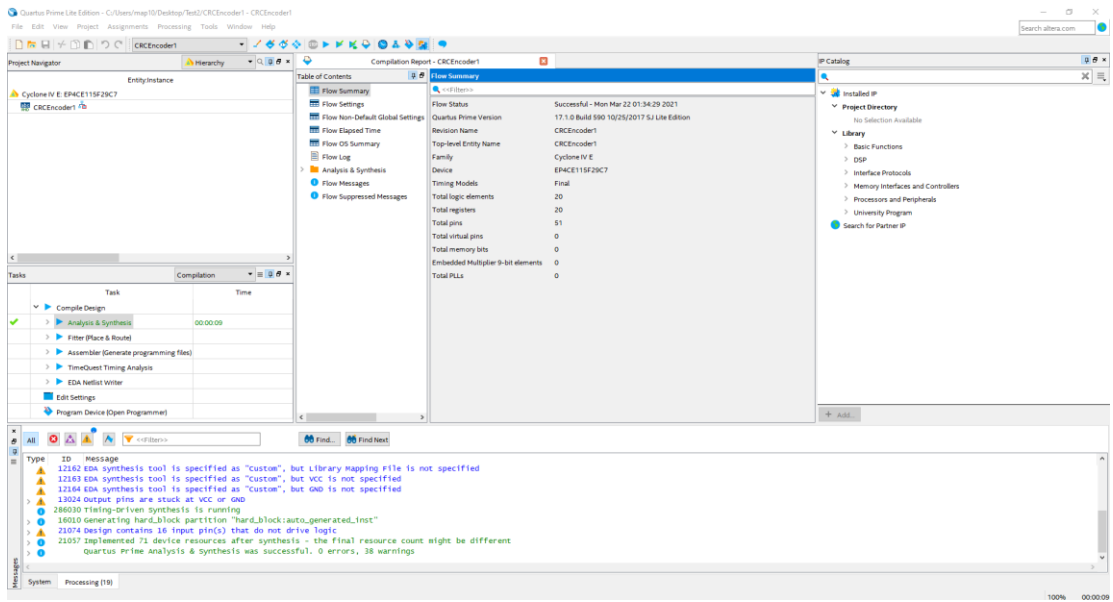


รูปที่ 4.26 หน้าต่างสำหรับเลือกเครื่องมือในการสังเคราะห์วงจรและจำลองผลของโปรเจก

8) ทำการกดดับเบิลคลิกที่ Analysis & Synthesis เพื่อสังเคราะห์วงจรดังรูปที่ 4.27 เมื่อกระบวนการเสร็จสิ้น จะได้นหน้าต่างเป็นเครื่องหมายถูกสีเขียวดังรูปที่ 4.28

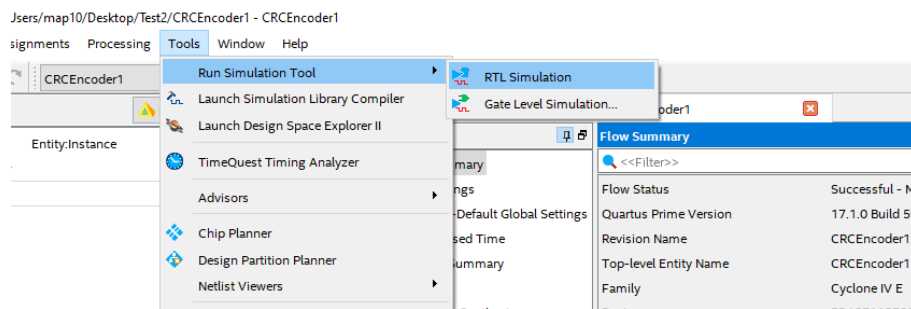


รูปที่ 4.27 ตำแหน่งเมนูที่ใช้ในการสังเคราะห์วงจรบนหน้าต่างเริ่มแรกของโปรแกรม



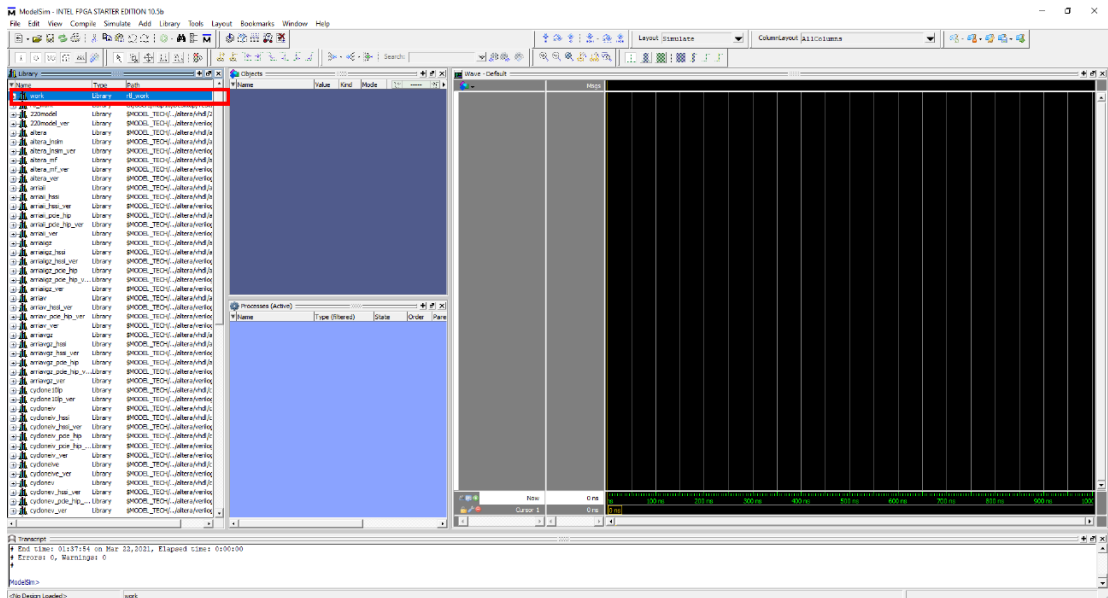
รูปที่ 4.28 ลักษณะของหน้าต่างเริ่มแรกเมื่อการสังเคราะห์วงจรเสร็จสมบูรณ์

9) เพื่อเริ่มทำการจำลองการใช้งานวงจร กดที่ Tools จากนั้นเลือก Run Simulation Tool แล้วกด RTL Simulation บนหน้าต่างของโปรแกรมดังรูปที่ 4.29



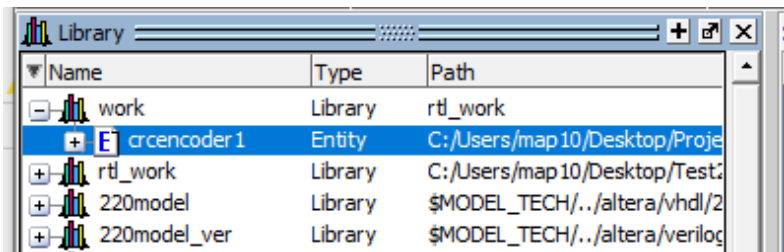
รูปที่ 4.29 ขั้นตอนการเลือกเมนูเพื่อทำการจำลองผลของวงจร

10) เมื่อพบหน้าต่างเริ่มแรกของโปรแกรม ModelSim-Altera ให้ทำการกดเครื่องหมาย + ข้างคำว่า work ดังรูปที่ 4.30 จะพบกับรายชื่อของ Entity ทั้งหมดที่มีอยู่ในโปรเจก



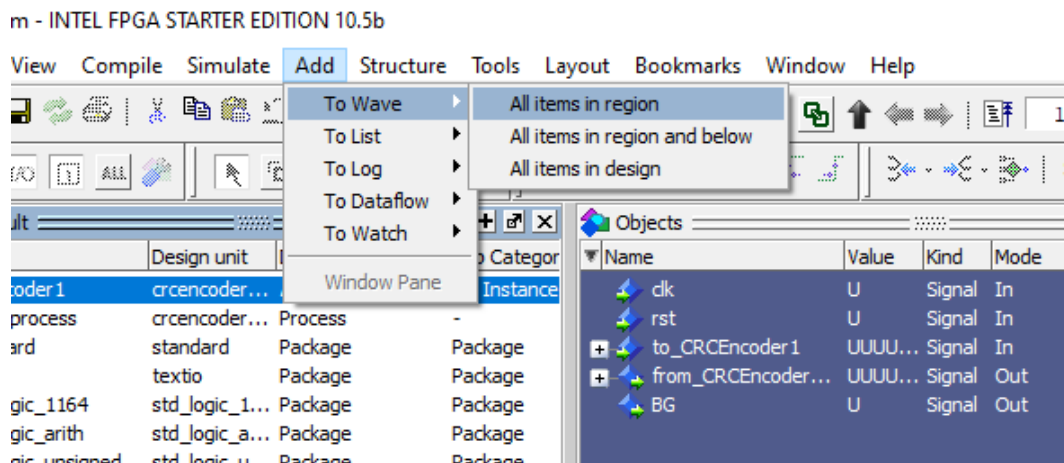
รูปที่ 4.30 ตำแหน่งของเมนูในการเลือก Entity เพื่อจำลองผลบนหน้าต่างเริ่มแรกของโปรแกรม

11) ทำการดับเบิลคลิกชื่อ Entity ของวงจรที่ต้องการใช้งานดังรูปที่ 4.31



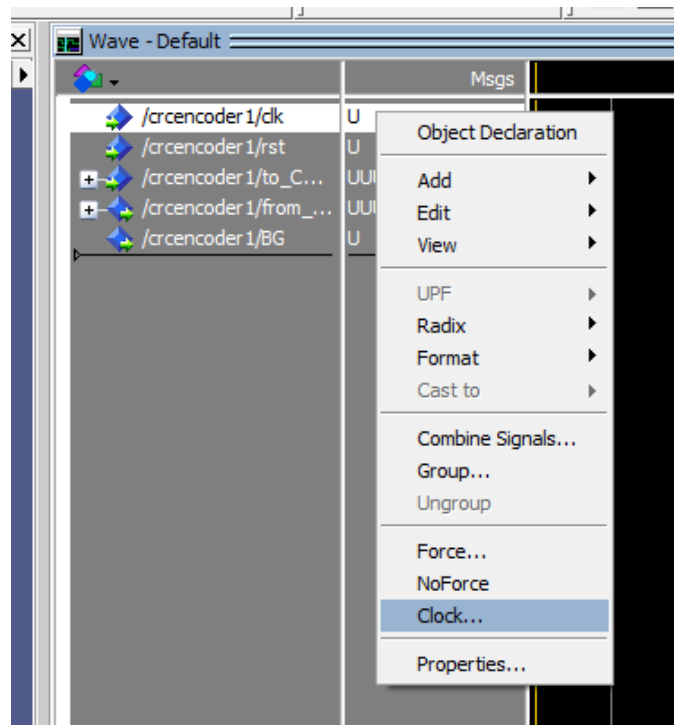
รูปที่ 4.31 ชื่อ Entity ทั้งหมดในโปรเจกต์ภายในเมนู Work

12) กด Add จากนั้นกด To Wave แล้วกด All items in region เพื่อเพิ่มสัญญาณทั้งหมดเข้าไปในหน้าต่าง Wave ดังรูปที่ 4.32 หากไม่ต้องการสัญญาณทั้งหมด ให้ทำการเลือกสัญญาณตามต้องการในหน้าต่าง Objects



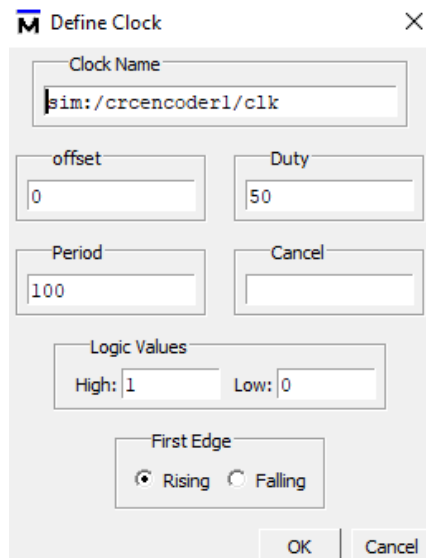
รูปที่ 4.32 ขั้นตอนเพิ่มสัญญาณจาก Entity ที่เลือกเข้าสู่หน้าต่าง Wave

13) ทำการตั้งค่าของสัญญาณ clk หรือสัญญาณนาฬิกาโดยการคลิกขวาที่สัญญาณ clk แล้วกด clock ดังรูปที่ 4.33



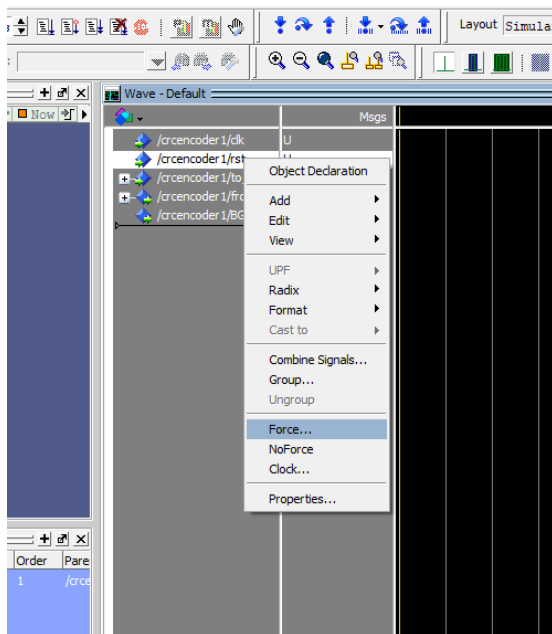
รูปที่ 4.33 ขั้นตอนการตั้งค่าสัญญาณนาฬิกา

14) ทำการตั้งค่าสัญญาณนาฬิกาตามต้องการ จากนั้นกด OK ดังรูปที่ 4.34



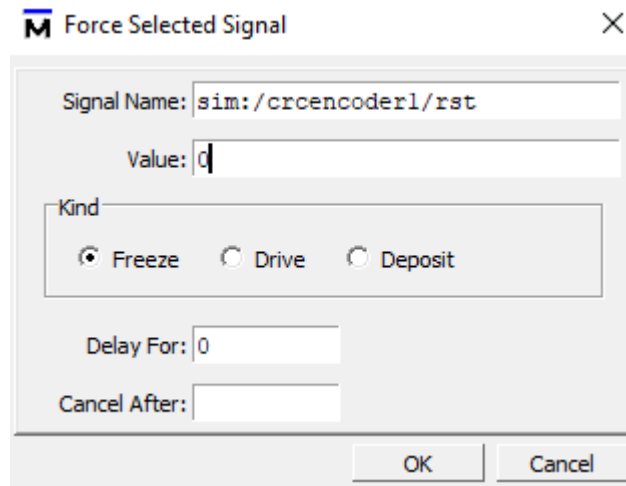
รูปที่ 4.34 หน้าต่างการตั้งค่าสัญญาณนาฬิกา

15) ทำการกำหนดค่าสัญญาณ rst หรือสัญญาณ Reset ด้วยการคลิกขวาที่สัญญาณ rst แล้วกด Force ดังรูปที่ 4.35



รูปที่ 4.35 เมนูสำหรับการกำหนดค่าสัญญาณใด ๆ

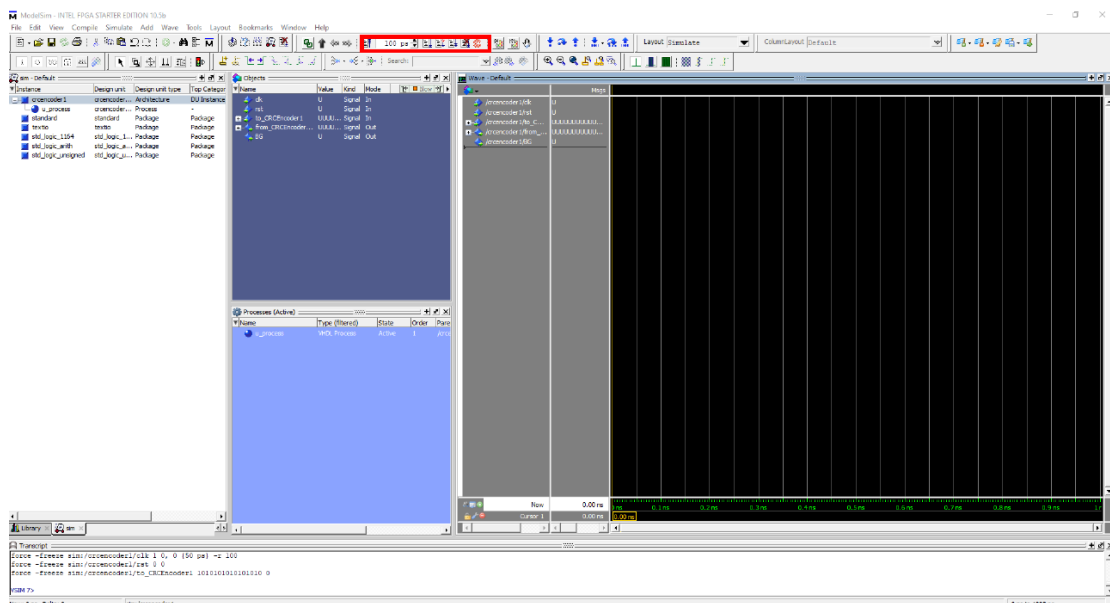
16) ทำการกำหนดค่าของสัญญาณ rst เป็น 0 เพื่อจำลองการกดปุ่ม Reset จากนั้นกด OK ดังรูปที่ 4.36



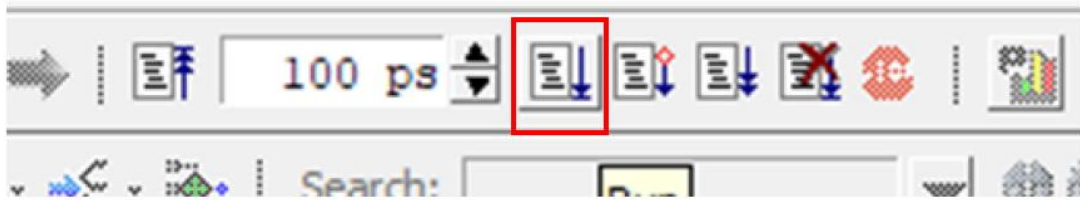
รูปที่ 4.36 หน้าต่างการกำหนดค่าของสัญญาณใด ๆ

17) ทำการกำหนดค่าของสัญญาณอินพุตอื่น ๆ ตามต้องการด้วยวิธีการเดียวกันกับหัวข้อที่ 16)

18) ณ บริเวณเมนูสำหรับการจำลองโปรแกรมดังรูปที่ 4.37 ให้ทำการกดปุ่มดังรูปที่ 4.38 เพื่อให้วงจรเริ่มทำงาน

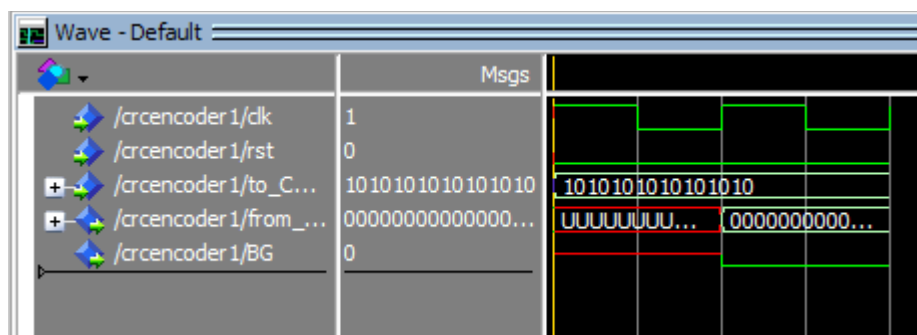


รูปที่ 4.37 ตำแหน่งของเมนูที่ใช้สำหรับการจำลองการทำงานของวงจรบนหน้าต่างเริ่มแรก



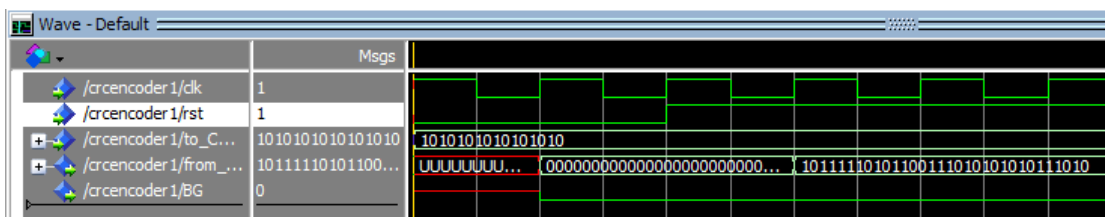
รูปที่ 4.38 ปุ่มสำหรับการจำลองการทำงานของวงจร

19) จะพบว่าเมื่อสัญญาณขึ้นมาในหน้าต่าง Wave ดังรูปที่ 4.39



รูปที่ 4.39 การทำงานของวงจรขณะสัญญาณ rst มีค่าเท่ากับ 0

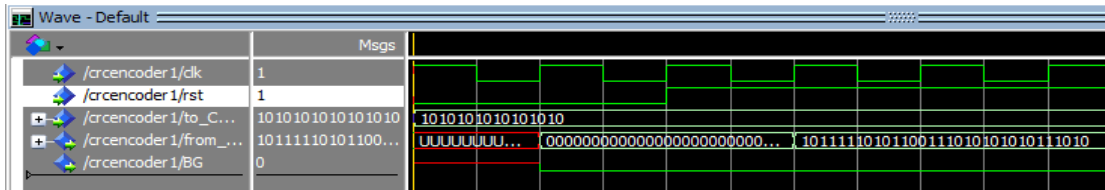
20) ทำการเซ็ตค่าของสัญญาณ rst เป็น 1 เพื่อจำลองการปล่อยปุ่ม Reset จะได้สัญญาณเอาต์พุตต่าง ๆ ดังรูปที่ 4.40



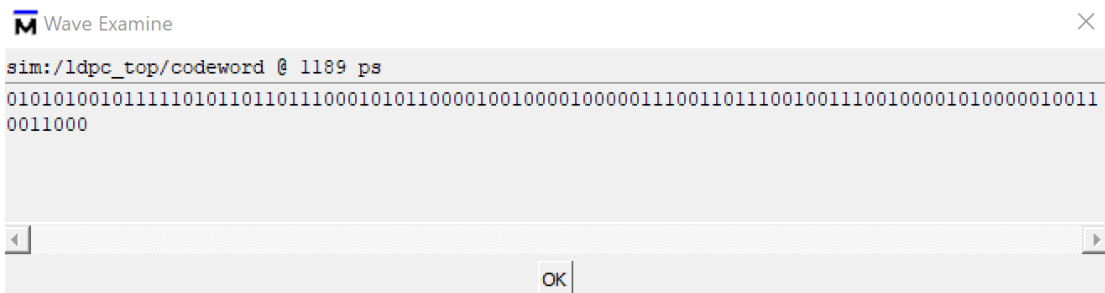
รูปที่ 4.40 การทำงานของวงจรขณะสัญญาณ rst มีค่าเท่ากับ 1

4.1.3.2 ผลการทดสอบวงจรเข้ารหัสช่องสัญญาณ LDPC

1) ผลการทดสอบวงจรเข้ารหัสช่องสัญญาณ LDPC แบบรวมทุกส่วนประกอบ ทำการป้อนอินพุตความยาว 8424 บิต ได้เอาต์พุตคือสัญญาณชื่อ codeword ดังรูปที่ 4.41 และรูปที่ 4.42

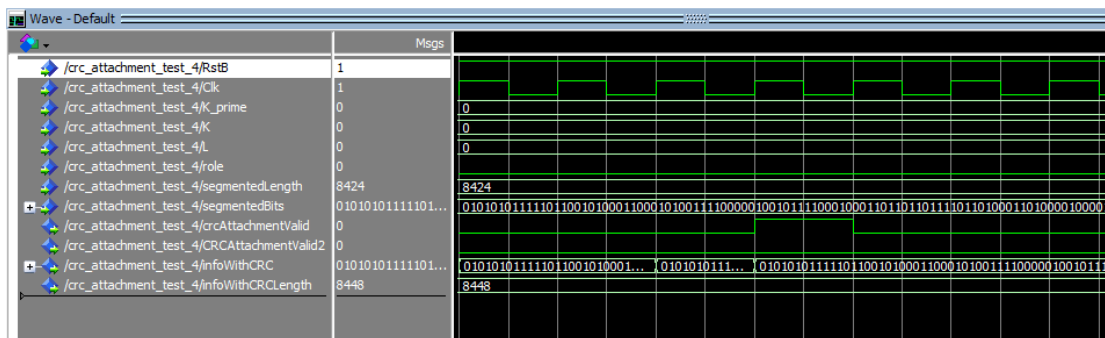


รูปที่ 4.41 ผลการทดสอบวงจรเข้ารหัสช่องสัญญาณ LDPC แบบรวมทุกส่วนประกอบจากโปรแกรม ModelSim-Altera



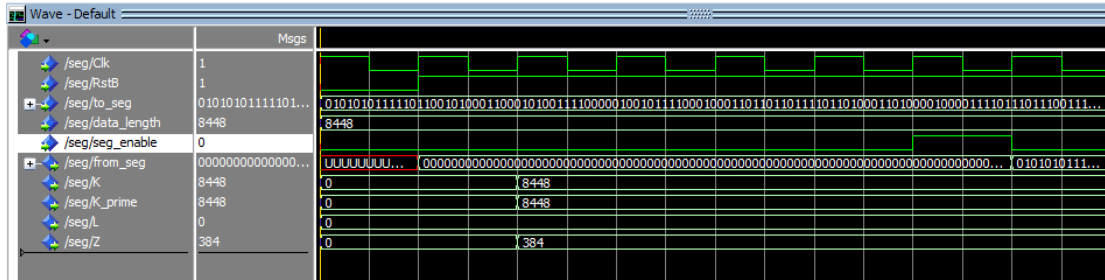
รูปที่ 4.42 สัญญาณ codeword (บางส่วน)

2) ผลการทดสอบวงจรเข้ารหัส CRC ทำการป้อนอินพุตความยาว 8424 ได้เอาต์พุตคือสัญญาณที่มีชื่อว่า infoWithCRC บิตดังรูปที่ 4.43



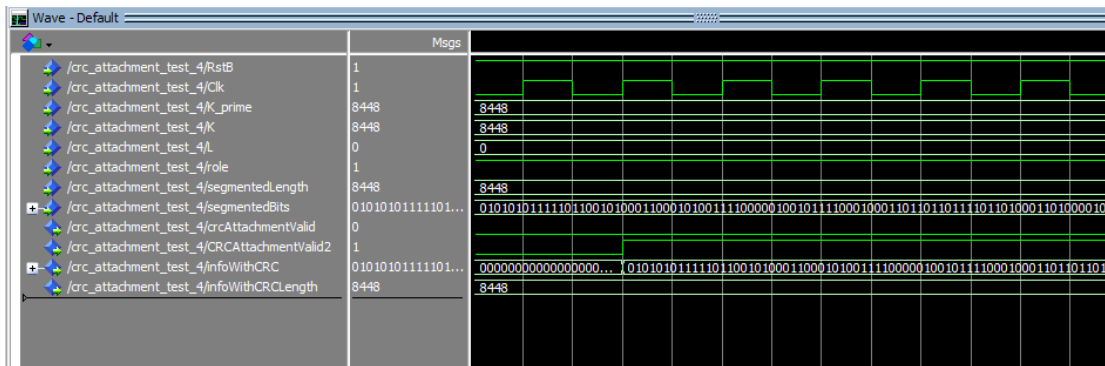
รูปที่ 4.43 ผลการทดสอบวงจรเข้ารหัส CRC จากโปรแกรม ModelSim-Altera

3) ผลการทดสอบวงจรแบ่งย่อยบล็อกข้อมูล ทำการป้อนอินพุตด้วยเอาต์พุตจากหัวข้อที่ 2) ได้เอาต์พุตคือสัญญาณชื่อ from_seg ดังรูปที่ 4.44



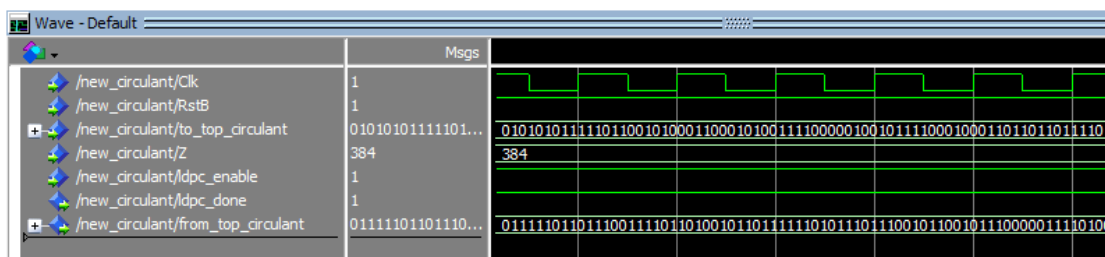
รูปที่ 4.44 ผลการทดสอบวงจรแบ่งย่อยบล็อกข้อมูลจากโปรแกรม ModelSim-Altera

4) ผลการทดสอบวงจรเข้ารหัส CRC ของบล็อกรหัส ทำการป้อนอินพุตด้วยเอาต์พุตจากหัวข้อที่ 3) ได้เอาต์พุตคือสัญญาณชื่อ infoWithCRC ดังรูปที่ 4.45

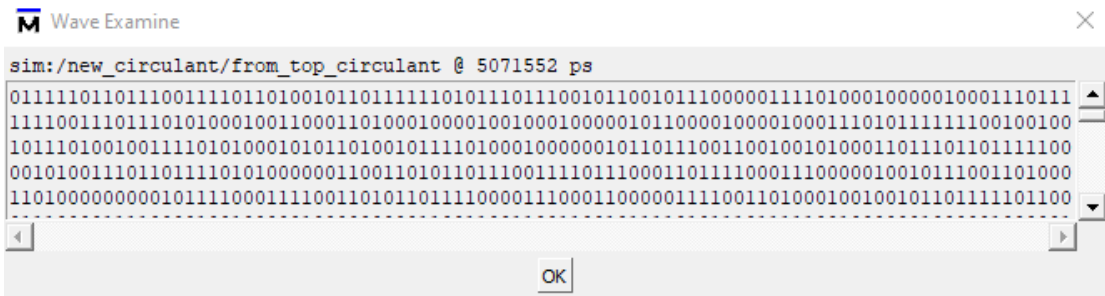


รูปที่ 4.45 ผลการทดสอบวงจรเข้ารหัส CRC ของบล็อกรหัสจากโปรแกรม ModelSim-Altera

5) ผลการทดสอบวงจรเข้ารหัส LDPC ทำการป้อนอินพุตด้วยเอาต์พุตจากหัวข้อที่ 4) ได้เอาต์พุตคือสัญญาณชื่อ from_top_circulant ดังรูปที่ 4.46 และรูปที่ 4.47

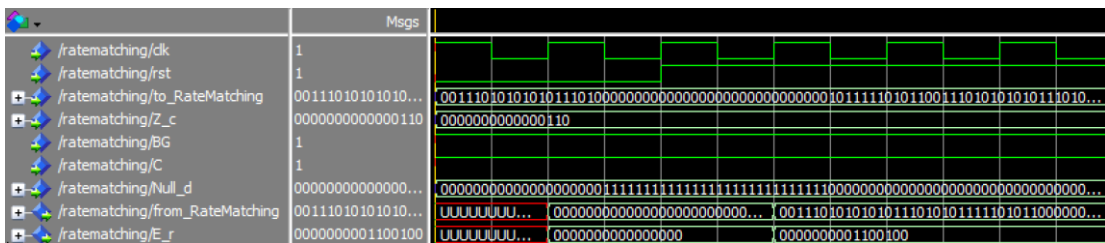


รูปที่ 4.46 ผลการทดสอบวงจรเข้ารหัส LDPC จากโปรแกรม ModelSim-Altera

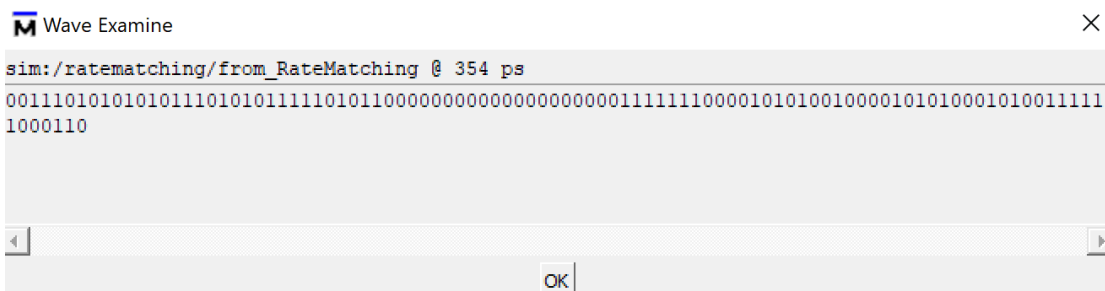


รูปที่ 4.47 สัญญาณ from_top_circulant (บางส่วน)

6) ผลการทดสอบวงจรปรับอัตรารหัส ทำการป้อนอินพุตด้วยเอาต์พุตจากหัวข้อที่ 5) ได้เอาต์พุตคือสัญญาณชื่อ from_RateMatching ดังรูปที่ 4.48 และรูปที่ 4.49

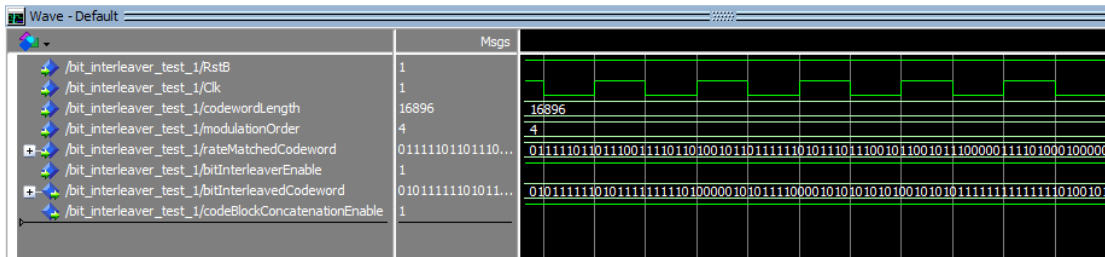


รูปที่ 4.48 ผลการทดสอบวงจรปรับอัตรารหัสจากโปรแกรม ModelSim-Altera

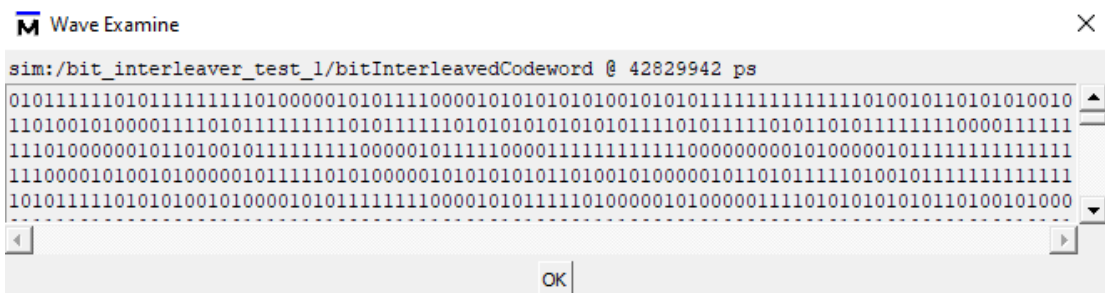


รูปที่ 4.49 สัญญาณ from_RateMatching (บางส่วน)

7) ผลการทดสอบวงจรแทรกสลับบิต ทำการป้อนอินพุตด้วยเอาต์พุตจากหัวข้อที่ 6) ได้เอาต์พุตคือสัญญาณชื่อ bitInterleavedCodeword ดังรูปที่ 4.50 และรูปที่ 4.51

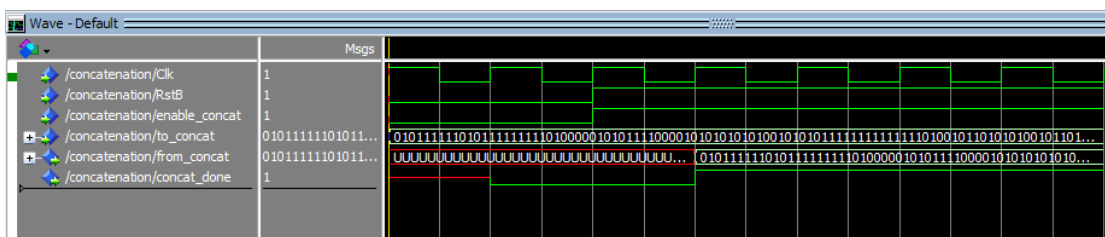


รูปที่ 4.50 ผลการทดสอบวงจรแทรกสลับบิตจากโปรแกรม ModelSim-Altera

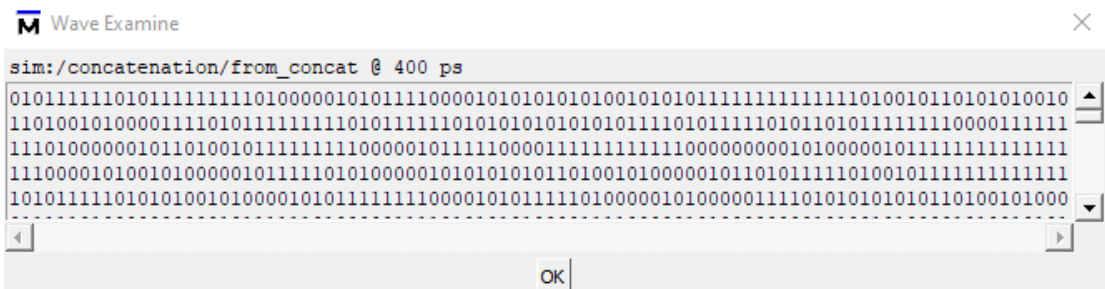


รูปที่ 4.51 สัญญาณ bitInterleavedCodeword (บางส่วน)

8) ผลการทดสอบวงจรต่อเรียงบล็อกรหัส ทำการป้อนอินพุตด้วยเอาต์พุตจากหัวข้อที่ 3) 6) และ 7) ได้เอาต์พุตคือสัญญาณชื่อ from_concat ขนาด 16896 บิตดังรูปที่ 4.52 และรูปที่ 4.53



รูปที่ 4.52 ผลการทดสอบวงจรต่อเรียงบล็อกรหัสจากโปรแกรม ModelSim-Altera



รูปที่ 4.53 สัญญาณบางส่วนของ from_concat

4.2 ผลการทดสอบชุดซอฟต์แวร์และวงจรเข้ารหัสช่องสัญญาณชนิด Polar codes

4.2.1 การใช้งานซอฟต์แวร์และการกำหนดพารามิเตอร์เริ่มต้น

ในหัวข้อนี้จะกล่าวถึงการใช้งานชุดซอฟต์แวร์เข้ารหัสช่องสัญญาณชนิด Polar codes ตามการออกแบบไว้ในหัวข้อที่ 3.2 ซึ่งสามารถดาวน์โหลดได้ที่เว็บไซต์ www.channelcoding.com ซึ่งประกอบไปด้วยชุดซอฟต์แวร์ 5 โมดูลหลักและไฟล์ชุดคำสั่ง แสดงดังรูปที่ 4.54



รูปที่ 4.54 ไฟล์โมดูลหลัก 5 โมดูลและไฟล์ชุดคำสั่งเพื่อกำหนดพารามิเตอร์และเรียกใช้โมดูลฟังก์ชัน

ไฟล์ชุดคำสั่งเพื่อกำหนดพารามิเตอร์และเรียกใช้โมดูลฟังก์ชันชื่อ polarEncoding.m สามารถแสดงชุดคำสั่งได้ดังรูปที่ 4.55

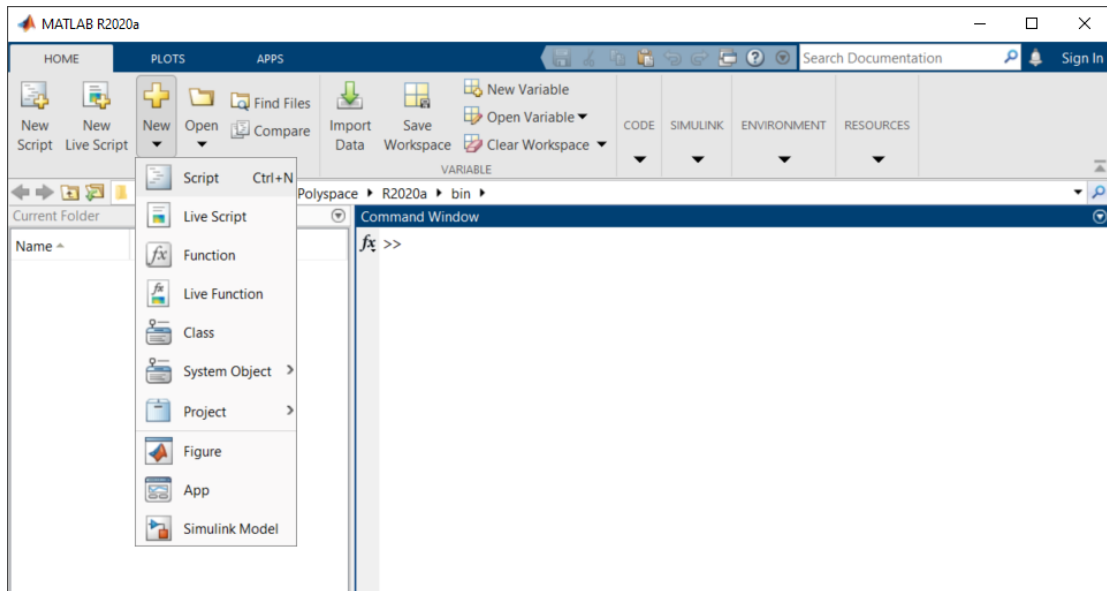
```
20
21 - channelFlag = 3;
22 - codewordLength = 72;
23 - rng(128);
24 - informationBits = ([1 1 1 0 1 0 1 0]);
25 - rntiBits = zeros(16, 1);
26
27 - [segmentedBits] = CodeBlockSegmentation(channelFlag, informationBits, codewordLength);
28 - [infoCRCBits, infoCRCBitsWOInt1, infoCRCBitsWOScrambling] = CRCAttachment(channelFlag, segmentedBits, rntiBits);
29 - [sourcewordBits, sourcewordBitsWOParity, subBlockInterleavingPattern, rateMatchingPattern, generatorMatrix, ...
30   frozenPositions] = PolarSequencing(channelFlag, infoCRCBits, codewordLength, length(informationBits));
31 - [codewordBits, subBlockInt1CodewordBits, motherCodewordBits] = PolarEncoding(sourcewordBits, ...
32   subBlockInterleavingPattern, rateMatchingPattern, generatorMatrix);
33 - [concatenatedBits, int1CodewordBits] = CodedBitInterleaver(channelFlag, codewordBits);
34
```

รูปที่ 4.55 ชุดคำสั่งเพื่อกำหนดพารามิเตอร์และเรียกใช้โมดูลฟังก์ชัน

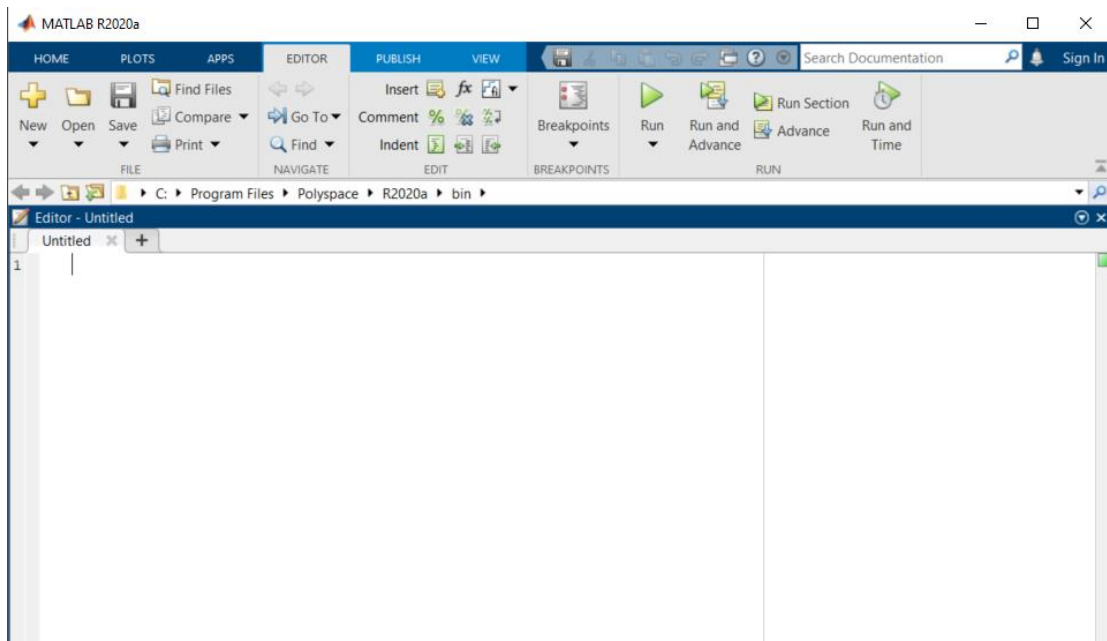
4.2.1.1 การสร้างไฟล์สกล m

ไฟล์สกล m เป็นสกลไฟล์ภายใต้ภาษา objective C ซึ่งสามารถใช้งานได้กับโปรแกรม MATLAB โดยไฟล์สกล m สามารถเป็นได้ทั้งไฟล์ที่เก็บชุดคำสั่ง MATLAB และสามารถเป็นไฟล์โมดูลฟังก์ชันได้

การสร้างไฟล์สกล m ทำได้โดยการสร้างไฟล์สคริปต์ขึ้นก่อน โดยกดที่ New > Script ดังรูปที่ 4.56 จากนั้นจะได้ไฟล์สคริปต์ขึ้นมาภายใต้หน้าต่าง editor ดังรูปที่ 4.57 ไฟล์สคริปต์จะสามารถเขียนชุดคำสั่ง MATLAB หรือสามารถเขียนเป็นโมดูลฟังก์ชันได้

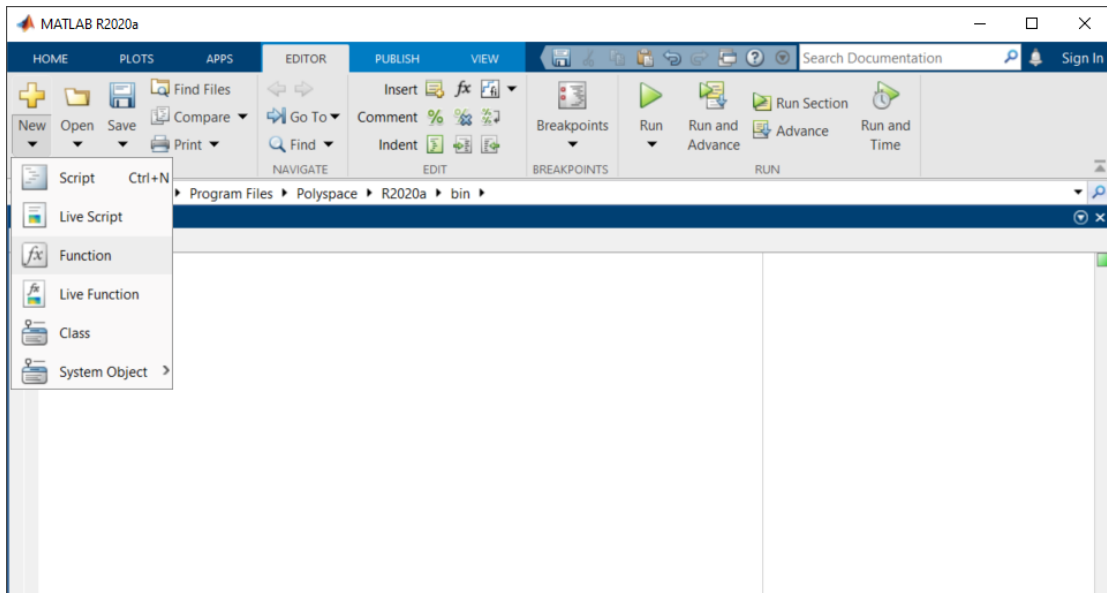


รูปที่ 4.56 ตำแหน่งการกดปุ่มสร้างไฟล์สคริป

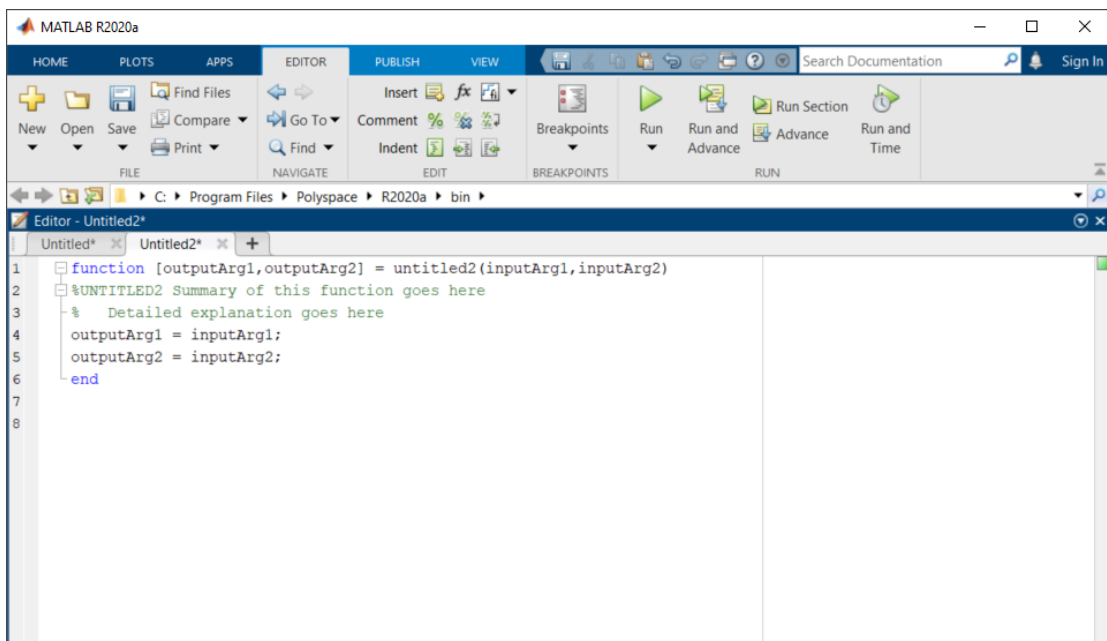


รูปที่ 4.57 ไฟล์สคริปที่ถูกสร้าง

นอกจากจะเขียนฟังก์ชันขึ้นมาจากไฟล์สคริปได้แล้ว โปรแกรมยังสามารถสร้างไฟล์สคริปที่มีเทมเพลตฟังก์ชันได้ทันที โดยกดที่ New > Function ดังรูปที่ 4.58 จากนั้นจะได้ไฟล์สคริปที่มีเทมเพลตฟังก์ชันขึ้นมาภายใต้หน้าต่าง editor ดังรูปที่ 4.59

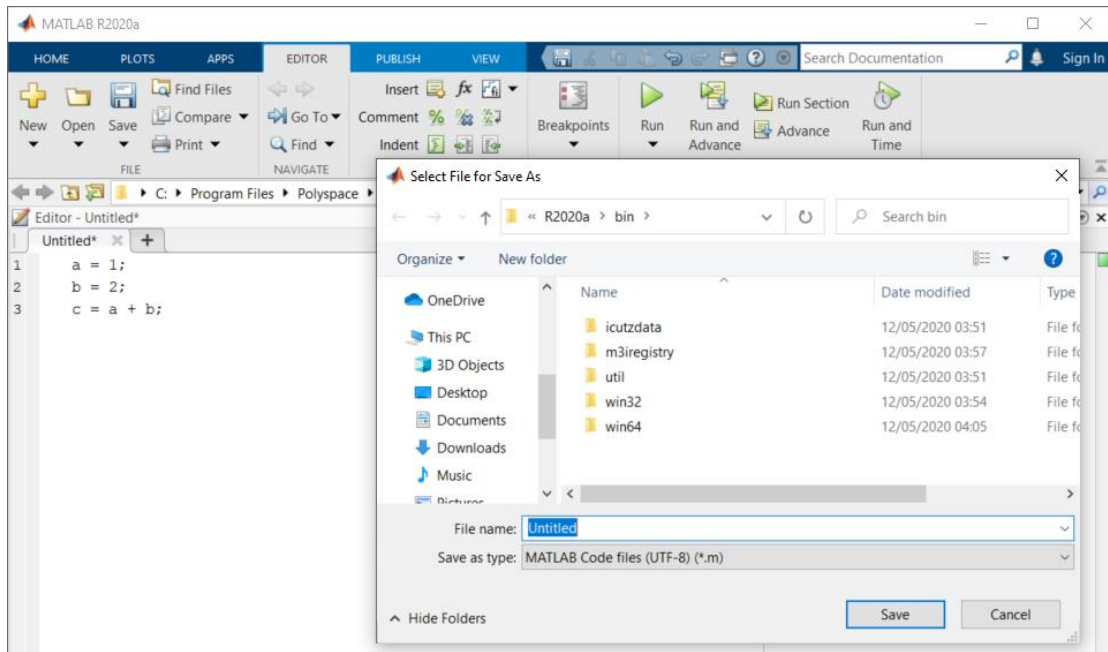


รูปที่ 4.58 ตำแหน่งการกดปุ่มสร้างไฟล์สคริปที่มีเทมเพลตฟังก์ชัน



รูปที่ 4.59 ไฟล์สคริปที่มีเทมเพลตฟังก์ชันที่ถูกสร้าง

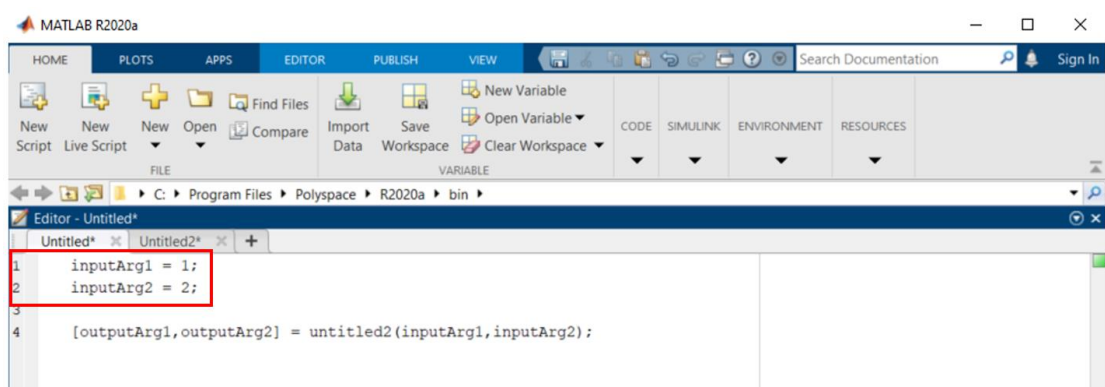
จากนั้นหากต้องการบันทึก ไฟล์จะถูกบันทึกเป็นไฟล์สกุล m ดังรูปที่ 4.60 โดยกดที่ Save ซึ่งหน้าต่าง Select File for Save As จะปรากฏขึ้นมาและสามารถกำหนดชื่อไฟล์และกดปุ่ม Save อีกครั้งเพื่อบันทึกไฟล์



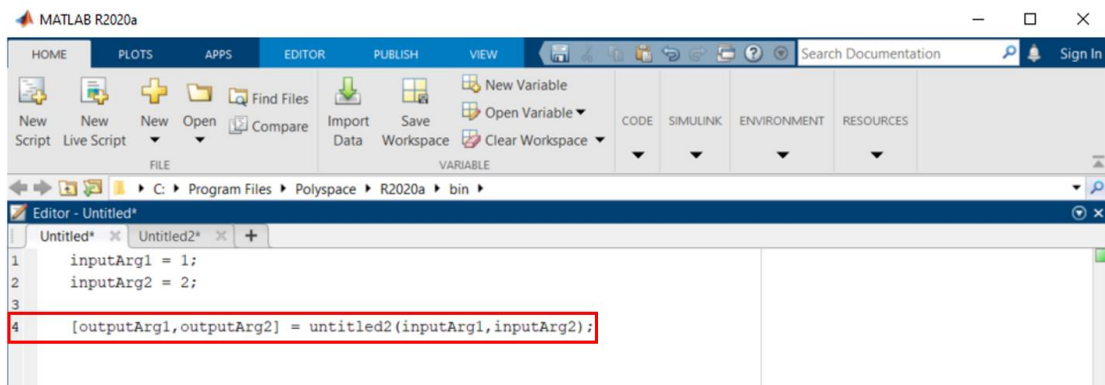
รูปที่ 4.60 การบันทึกไฟล์สกล m

4.2.1.2 การเขียนคำสั่งเพื่อกำหนดพารามิเตอร์และการเรียกใช้โมดูลฟังก์ชัน

ไฟล์สกล m ที่สร้างจากหัวข้อ 4.2.1.1 จะใช้สำหรับการกำหนดพารามิเตอร์และการเรียกใช้โมดูลฟังก์ชัน การกำหนดพารามิเตอร์สามารถทำได้โดยชุดคำสั่งบรรทัดที่ 1 และ 2 ดังรูปที่ 4.61 และการเรียกใช้งานโมดูลฟังก์ชันสามารถทำได้โดยชุดคำสั่งบรรทัดที่ 4 ดังรูปที่ 4.62 โดยฟังก์ชันจะทำการคืนค่าพารามิเตอร์ที่ปรากฏทางซ้ายของบรรทัด



รูปที่ 4.61 การเขียนคำสั่งเพื่อกำหนดพารามิเตอร์



รูปที่ 4.62 การเขียนคำสั่งเพื่อเรียกใช้โมดูลฟังก์ชัน

4.2.2 ผลการทดสอบชุดซอฟต์แวร์เข้ารหัสโพลาร์ตามมาตรฐาน 5G

การเก็บผลการทดสอบชุดซอฟต์แวร์เข้ารหัสโพลาร์ตามมาตรฐาน 5G จะทำการแบ่งส่วนตามประเภทของช่องสัญญาณ ซึ่งประกอบไปด้วย 3 ช่องสัญญาณ ช่องสัญญาณ PBCH ช่องสัญญาณ PDCCH และช่องสัญญาณ PUCCH

4.2.2.1 ผลการทดสอบชุดซอฟต์แวร์เข้ารหัสกรณีช่องสัญญาณ PBCH

สำหรับผลการทดสอบของช่องสัญญาณ PBCH หัวข้อในการเก็บผลจะถูกแบ่งออกตามจำนวนกระบวนการเข้ารหัสของช่องสัญญาณ PBCH ซึ่งประกอบด้วย 6 กระบวนการ และรวมถึงผลลัพธ์ของกระบวนการโดยรวมดังนี้

- 1) กระบวนการทำงานโดยรวม
- 2) การเข้ารหัส CRC
- 3) การแทรกสลับ CRC
- 4) การลำดับช่องสัญญาณย่อย
- 5) การเข้ารหัสโพลาร์
- 6) การแทรกสลับบล็อกย่อย
- 7) การปรับอัตรารหัส

1) ผลการทดสอบกระบวนการทำงานโดยรวม

เมื่อผู้ใช้งานกรอกบิตข้อมูลขาเข้าหรือเวกเตอร์ **a** แสดงดังรูปที่ 4.63 ระบบการเข้ารหัสโพลาร์สำหรับช่องสัญญาณ PBCH จะให้ผลลัพธ์บิตข้อมูลขาออกแสดงดังรูปที่ 4.64

```

informationBits =
Columns 1 through 24
  1   1   1   0   1   0   1   0   0   0   1   0   0   0   0   0   1   1   1   0   1   0   1   0
Columns 25 through 32
  0   0   1   0   0   0   0   0
>>

```

รูปที่ 4.63 บิตข้อมูลขาเข้า

```

concatenatedBits =
Columns 1 through 24
  0   1   0   1   0   0   0   0   1   1   1   1   0   1   0   1   0   1   1   1   1   0   1   0   0
Columns 25 through 48
  0   1   0   0   0   0   0   0   1   0   1   0   0   1   1   1   1   1   0   1   0   0   0   0   0
Columns 49 through 72
  0   0   0   0   1   0   1   1   1   0   1   1   0   0   0   0   1   1   1   1   1   1   1   0   1
Columns 73 through 96
  1   0   0   0   0   1   0   1   1   0   1   0   1   1   1   0   0   0   0   1   1   0   1   0
Columns 97 through 120
  0   0   0   0   0   0   1   0   0   1   1   1   0   1   0   1   0   1   0   1   1   0   0   0
Columns 121 through 144

```

รูปที่ 4.64 บิตข้อมูลขาออก (บางส่วน)

โดยรูปที่ 4.63 บิตข้อมูลขาเข้าเวกเตอร์ **a** มีความยาว $A = 32$ บิต โดยช่องสัญญาณ PBCH สามารถใช้ความยาวบิตข้อมูลได้เพียงความยาวเดียว และรูปที่ 4.64 ผลลัพธ์บิตข้อมูลขาออกมีความยาว $E = 864$ ซึ่งมีความยาวตรงกับความยาวคำรหัสของช่องสัญญาณ PBCH ซึ่งมีได้เพียงความยาวเดียว

2) ผลการทดสอบกระบวนการเข้ารหัส CRC

เมื่อป้อนอินพุตเวกเตอร์ **a** ความยาว A แสดงดังรูปที่ 4.63 ข้อมูลจะไม่ผ่านกระบวนการแบ่งย่อยบล็อกรหัสสำหรับช่องสัญญาณ PBCH ข้อมูลจะผ่านกระบวนการเข้ารหัส CRC และจะให้ผลลัพธ์เวกเตอร์ **c** ความยาว K แสดงดังรูปที่ 4.65

```

infoCRCBitsWOInt1 =
Columns 1 through 24
  1   1   1   0   1   0   1   0   0   0   1   0   0   0   0   0   1   1   1   0   1   0   1   0
Columns 25 through 48
  0   0   1   0   0   0   0   0   1   1   1   0   0   0   0   1   1   0   0   1   1   0   0   0
Columns 49 through 56
  1   0   1   0   0   1   0   0

```

รูปที่ 4.65 เวกเตอร์ **c**

โดยรูปที่ 4.65 เวกเตอร์ **c** มีความยาว $K = 56$ บิต ประกอบด้วยเวกเตอร์ **a** ความยาว $A = 32$ บิตและบิต CRC ความยาว $L = 24$ บิต

3) ผลการทดสอบกระบวนการแทรกสลับ CRC

เวกเตอร์ **c** จากกระบวนการเข้ารหัส CRC จะผ่านกระบวนการแทรกสลับ CRC และจะให้ผลลัพธ์เวกเตอร์ **c'** แสดงดังรูปที่ 4.66

```
infoCRCBits =
Columns 1 through 24
    1    1    0    0    0    1    0    0    0    0    1    0    0    0    1    0    0    1    1    1    1    0    0    1
Columns 25 through 48
    1    1    0    0    1    0    1    0    0    1    0    0    0    0    0    1    1    0    0    1    1    0    0    0
Columns 49 through 56
    1    0    1    0    0    1    0    0
>>
```

รูปที่ 4.66 เวกเตอร์ **c'**

โดยรูปที่ 4.66 เวกเตอร์ **c'** มีความยาวเท่ากับเวกเตอร์ **c** ที่ $K = 56$ บิต แต่จะมีลำดับบิตที่แตกต่างกันเนื่องเกิดการแทรกสลับ

4) ผลการทดสอบกระบวนการลำดับช่องสัญญาณย่อย

เวกเตอร์ **c'** จะไม่ผ่านกระบวนการสแครมเบิลสำหรับช่องสัญญาณ PBCH และเข้าไปยังกระบวนการลำดับช่องสัญญาณย่อยและให้ผลลัพธ์เวกเตอร์ **u** แสดงดังรูปที่ 4.67

```
sourcewordBits =
Columns 1 through 24
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
Columns 25 through 48
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
Columns 49 through 72
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
Columns 73 through 96
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
Columns 97 through 120
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
Columns 121 through 144
```

รูปที่ 4.67 เวกเตอร์ **u** (บางส่วน)

โดยรูปที่ 4.67 เวกเตอร์ **u** มีความยาว $N=512$ บิต ซึ่งถือเป็นความยาวคำรหัสแม่สูงสุดของช่องสัญญาณ PBCH โดยบิตในเวกเตอร์ **c'** จะอยู่ภายในตำแหน่งของเวกเตอร์ **u** ร่วมกับบิตซ้ำซ้ำ

5) ผลการทดสอบกระบวนการเข้ารหัสโพลาร์

เวกเตอร์ **u** จะไม่มีกระบวนการคำนวณบิดพริ้วสำหรับช่องสัญญาณ PBCH โดยจะผ่านกระบวนการเข้ารหัสโพลาร์ ซึ่งจะให้ผลลัพธ์เวกเตอร์ **d** แสดงดังรูปที่ 4.68

```

motherCodewordBits =
Columns 1 through 24
  0  1  0  1  0  0  0  1  1  1  1  0  1  0  1  0  1  1  1  1  0  1  0  0
Columns 25 through 48
  0  1  0  0  0  0  0  0  1  0  1  0  0  1  1  1  1  1  0  1  0  0  0  0
Columns 49 through 72
  1  1  1  1  1  1  0  1  1  0  0  0  0  1  0  1  0  0  0  0  1  0  1  1
Columns 73 through 96
  1  0  1  1  0  0  0  0  1  0  1  0  1  1  1  0  0  0  0  1  1  0  1  0
Columns 97 through 120
  0  0  0  0  0  0  1  0  0  1  1  1  0  1  0  1  0  1  0  1  1  0  0  0
Columns 121 through 144

```

รูปที่ 4.68 เวกเตอร์ **d** (บางส่วน)

โดยรูปที่ 4.68 เวกเตอร์ **d** มีความยาวเท่ากับเวกเตอร์ **u** ที่ $N=512$ บิต

6) ผลการทดสอบกระบวนการแทรกสลับบล็อกย่อย

หลังกระบวนการเข้ารหัสโพลาร์ เวกเตอร์ **d** จะผ่านกระบวนการแทรกสลับบล็อกย่อยและให้ผลลัพธ์เวกเตอร์ **y** แสดงดังรูปที่ 4.69

```

subBlockIntIcodewordBits =
Columns 1 through 24
  0  1  0  1  0  0  0  1  1  1  1  0  1  0  1  0  1  1  1  1  0  1  0  0
Columns 25 through 48
  0  1  0  0  0  0  0  0  1  0  1  0  0  1  1  1  1  1  0  1  0  0  0  0
Columns 49 through 72
  0  0  0  0  1  0  1  1  1  0  1  1  0  0  0  0  1  1  1  1  1  1  0  1
Columns 73 through 96
  1  0  0  0  0  1  0  1  1  0  1  0  1  1  1  0  0  0  0  1  1  0  1  0
Columns 97 through 120
  0  0  0  0  0  0  1  0  0  1  1  1  0  1  0  1  0  1  0  1  1  0  0  0
Columns 121 through 144

```

รูปที่ 4.69 เวกเตอร์ **y** (บางส่วน)

โดยรูปที่ 4.69 เวกเตอร์ **y** มีความยาวเท่ากับเวกเตอร์ **d** ที่ $N = 512$ บิต แต่จะมีลำดับบิตที่แตกต่างกันเนื่องเกิดการแทรกสลับ

7) ผลการทดสอบกระบวนการปรับอัตรารหัส

เวกเตอร์ **y** จะผ่านกระบวนการปรับอัตรารหัสและให้ผลลัพธ์เวกเตอร์ **e** แสดงดังรูปที่

4.70

```

codewordBits =
Columns 1 through 24
  0  1  0  1  0  0  0  1  1  1  1  0  1  0  1  0  1  1  1  1  0  1  0  0
Columns 25 through 48
  0  1  0  0  0  0  0  0  1  0  1  0  0  1  1  1  1  1  0  1  0  0  0  0
Columns 49 through 72
  0  0  0  0  1  0  1  1  1  0  1  1  0  0  0  0  1  1  1  1  1  1  0  1
Columns 73 through 96
  1  0  0  0  0  1  0  1  1  0  1  0  1  1  1  0  0  0  0  1  1  0  1  0
Columns 97 through 120
  0  0  0  0  0  0  1  0  0  1  1  1  0  1  0  1  0  1  0  1  1  0  0  0
Columns 121 through 144

```

รูปที่ 4.70 เวกเตอร์ **e** (บางส่วน)

ผลลัพธ์ของระบบการเข้ารหัสโพลาร์สำหรับช่องสัญญาณ PBCH จะสิ้นสุดที่กระบวนการนี้ โดยรูปที่ 4.70 เวกเตอร์ **e** จะเป็นผลลัพธ์บิตข้อมูลขาออกที่มีความยาว $E = 864$ บิต ซึ่งจะมีความยาวยาวมากกว่าเวกเตอร์ **y** ที่ $N = 512$ บิต เนื่องจากการปรับอัตรารหัสแบบส่งซ้ำ

4.2.2.2 ผลการทดสอบชุดซอฟต์แวร์เข้ารหัสกรณีช่องสัญญาณ PDCCH

สำหรับผลการทดสอบของช่องสัญญาณ PDCCH หัวข้อในการเก็บผลจะถูกแบ่งออกตามจำนวนกระบวนการเข้ารหัสของช่องสัญญาณ PDCCH ซึ่งประกอบด้วย 7 กระบวนการ และรวมถึงผลลัพธ์ของกระบวนการโดยรวมดังนี้

- 1) กระบวนการทำงานโดยรวม
- 2) การเข้ารหัส CRC
- 3) การสแครมบลิง CRC
- 4) การแทรกสลับ CRC
- 5) การลำดับช่องสัญญาณย่อย
- 6) การเข้ารหัสโพลาไรซ์
- 7) การแทรกสลับบล็อกย่อย
- 8) การปรับอัตรารหัส

1) ผลการทดสอบกระบวนการทำงานโดยรวม

เมื่อผู้ใช้งานกรอกบิตข้อมูลขาเข้าหรือเวกเตอร์ **a** แสดงดังรูปที่ 4.71 ระบบการเข้ารหัสโพลาไรซ์สำหรับช่องสัญญาณ PDCCH จะให้ผลลัพธ์บิตข้อมูลขาออกแสดงดังรูปที่ 4.72

```
informationBits =  
  
Columns 1 through 19  
    1    1    1    0    1    0    1    0    0    0    1    0    0    0    0    0    1    1    1  
  
Columns 20 through 32  
    0    1    0    1    0    0    0    1    0    0    0    0    0
```

รูปที่ 4.71 บิตข้อมูลขาเข้า

```

concatenatedBits =
Columns 1 through 19
  1   0   0   1   1   0   0   0   0   0   0   1   0   0   1   1   0   1   0   1
Columns 20 through 38
  0   0   1   1   1   1   1   0   1   1   0   1   0   1   1   0   0   0   0   1
Columns 39 through 57
  0   0   1   0   1   1   0   1   1   0   1   | 0   1   0   0   0   0   0   1   1
Columns 58 through 76
  1   1   0   1   1   1   1   0   1   1   0   1   1   0   1   0   0   0   1   0
Columns 77 through 95
  0   0   1   1   0   0   1   1   1   1   0   1   0   1   1   1   1   1   1   0

```

รูปที่ 4.72 บิตข้อมูลขาออก (บางส่วน)

โดยรูปที่ 4.71 บิตข้อมูลขาเข้าเวกเตอร์ **a** มีความยาว $A=32$ บิต โดยช่องสัญญาณ PDCCH สามารถใช้ความยาวบิตข้อมูลได้ตั้งแต่ 1 ถึง 140 บิต และรูปที่ 4.72 ผลลัพธ์บิตข้อมูลขาออกมีความยาว $E=432$ บิต ตามความต้องการของผู้ใช้

2) ผลการทดสอบกระบวนการเข้ารหัส CRC

เมื่อป้อนอินพุตเวกเตอร์ **a** ความยาว A แสดงดังรูปที่ 116 ข้อมูลจะไม่ผ่านกระบวนการแปลงย่อยบล็อกรหัสสำหรับช่องสัญญาณ PDCCH ข้อมูลจะผ่านกระบวนการเข้ารหัส CRC และจะให้ผลลัพธ์เวกเตอร์ **c** ความยาว K แสดงดังรูปที่ 4.73

```

infoCRCBitsWOScrambling =
Columns 1 through 19
  1   1   1   0   1   0   1   0   0   0   0   1   0   0   0   0   0   1   1   1
Columns 20 through 38
  0   1   0   1   0   0   0   1   0   0   0   0   0   0   1   0   1   1   0   0
Columns 39 through 56
  1   0   1   1   0   0   0   1   0   1   0   1   0   0   1   1   0   1

```

รูปที่ 4.73 เวกเตอร์ **c** ที่ได้จากการเข้ารหัส CRC

โดยรูปที่ 4.73 เวกเตอร์ **c** มีความยาว $K=56$ บิต ประกอบด้วยเวกเตอร์ **a** ความยาว $A=32$ บิตและบิต CRC ความยาว $L=24$ บิต

3) ผลการทดสอบกระบวนการสแครมbling CRC

หลังจากกระบวนการเข้ารหัส CRC ภายใต้ช่องสัญญาณ PDCCH บิต CRC 16 บิตสุดท้ายจะถูกทำการสแครมbling ได้ผลลัพธ์เวกเตอร์ \mathbf{c} ความยาว K แสดงดังรูปที่ 4.74

```

infoCRCBitsWOInt1 =
Columns 1 through 19
 1  1  1  0  1  0  1  0  0  0  1  0  0  0  0  0  1  1  1
Columns 20 through 38
 0  1  0  1  0  0  0  1  0  0  0  0  0  1  0  1  1  0  0
Columns 39 through 56
 1  0  0  0  1  1  1  0  1  0  1  0  1  1  0  0  1  0
    
```

รูปที่ 4.74 เวกเตอร์ \mathbf{c} ที่ได้จากการสแครมbling CRC

โดยรูปที่ 4.74 เวกเตอร์ \mathbf{c} มีความยาว $K = 56$ บิต เช่นเดียวกับกระบวนการเข้ารหัส CRC โดยที่บิตนอกเหนือจากบิต CRC 16 บิตสุดท้ายจะเหมือนกับกระบวนการก่อนหน้า

4) ผลการทดสอบกระบวนการแทรกสลับ CRC

เวกเตอร์ \mathbf{c} จากกระบวนการสแครมbling CRC จะผ่านกระบวนการแทรกสลับ CRC และจะให้ผลลัพธ์เวกเตอร์ \mathbf{c}' แสดงดังรูปที่ 4.75

```

infoCRCBits =
Columns 1 through 19
 1  1  0  0  0  1  0  0  0  0  1  0  0  0  1  0  0  1  1
Columns 20 through 38
 1  1  0  0  1  1  1  0  0  0  0  1  0  0  1  0  1  0  0
Columns 39 through 56
 1  0  0  0  1  1  1  0  1  0  1  0  1  1  0  0  1  0
    
```

รูปที่ 4.75 เวกเตอร์ \mathbf{c}'

โดยรูปที่ 4.75 เวกเตอร์ \mathbf{c}' มีความยาวเท่ากับเวกเตอร์ \mathbf{c} ที่ $K = 56$ บิต แต่จะมีลำดับบิตที่ต่างกันอย่างเนื่องเกิดการแทรกสลับ

5) ผลการทดสอบกระบวนการลำดับช่องสัญญาณย่อย

เวกเตอร์ **c'** จะผ่านไปยังกระบวนการลำดับช่องสัญญาณย่อยและให้ผลลัพธ์เวกเตอร์ **u** แสดงดังรูปที่ 4.76

```
sourcewordBits =
Columns 1 through 19
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Columns 20 through 38
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Columns 39 through 57
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Columns 58 through 76
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Columns 77 through 95
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

รูปที่ 4.76 เวกเตอร์ **u** (บางส่วน)

โดยรูปที่ 4.76 เวกเตอร์ **u** มีความยาว $N=512$ บิต โดยบิตในเวกเตอร์ **c'** จะอยู่ภายในตำแหน่งของเวกเตอร์ **u** ร่วมกับบิตซ้ำซ้ำ

6) ผลการทดสอบกระบวนการเข้ารหัสโพลาไร

เวกเตอร์ **u** จะไม่มีกระบวนการคำนวณบิตพริตตี้สำหรับช่องสัญญาณ PDCCH โดยจะผ่านกระบวนการเข้ารหัสโพลาไร ซึ่งจะให้ผลลัพธ์เวกเตอร์ **d** แสดงดังรูปที่ 4.77

```
motherCodewordBits =
Columns 1 through 19
0 1 0 1 1 1 1 0 1 1 1 0 1 1 1 1 1 1
Columns 20 through 38
0 0 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0
Columns 39 through 57
1 0 0 1 1 1 1 1 1 1 0 1 1 0 0 0 1 0
Columns 58 through 76
0 0 1 0 0 1 1 0 0 0 0 0 1 0 0 1 0 1
Columns 77 through 95
0 1 0 1 1 0 0 1 1 0 0 0 0 0 1 0 0 1
```

รูปที่ 4.77 เวกเตอร์ **d** (บางส่วน)

โดยรูปที่ 4.77 เวกเตอร์ **d** มีความยาวเท่ากับเวกเตอร์ **u** ที่ $N=512$ บิต

7) ผลการทดสอบกระบวนการแทรกสลับบล็อกย่อย

หลังกระบวนการเข้ารหัสโพลาร์ เวกเตอร์ **d** จะผ่านกระบวนการแทรกสลับบล็อกย่อยและให้ผลลัพธ์เวกเตอร์ **y** แสดงดังรูปที่ 4.78

```
subBlockInt1codewordBits =
Columns 1 through 19
0 1 0 1 1 1 1 0 1 1 1 0 1 1 1 1 1 0
Columns 20 through 38
0 0 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 0
Columns 39 through 57
1 0 0 1 1 1 1 1 1 1 0 0 0 0 0 1 0 0 1
Columns 58 through 76
0 1 1 0 1 0 1 0 1 1 0 0 0 0 1 0 0 0 1
Columns 77 through 95
0 0 1 1 1 0 0 1 1 0 0 0 0 0 1 0 0 1 1
```

รูปที่ 4.78 เวกเตอร์ **y** (บางส่วน)

โดยรูปที่ 4.78 เวกเตอร์ **y** มีความยาวเท่ากับเวกเตอร์ **d** ที่ $N = 512$ บิต แต่จะมีลำดับบิตที่แตกต่างกันเนื่องเกิดการแทรกสลับ

8) ผลการทดสอบกระบวนการปรับอัตราหัส

เวกเตอร์ **y** จะผ่านกระบวนการปรับอัตราหัสและให้ผลลัพธ์เวกเตอร์ **e** แสดงดังรูปที่ 4.79

```
codewordBits =
Columns 1 through 19
1 0 0 1 1 0 0 0 0 0 1 0 0 1 1 0 1 0 1
Columns 20 through 38
0 0 1 1 1 1 1 0 1 1 0 1 0 1 1 0 0 0 1
Columns 39 through 57
0 0 1 0 1 1 0 1 1 0 1 0 1 0 0 0 0 1 1
Columns 58 through 76
1 1 0 1 1 1 1 0 1 1 0 1 1 0 1 0 0 1 0
Columns 77 through 95
0 0 1 1 0 0 1 1 1 1 0 1 0 1 1 1 1 1 0
```

รูปที่ 4.79 เวกเตอร์ **e** (บางส่วน)

ผลลัพธ์ของระบบการเข้ารหัสโพลาร์สำหรับช่องสัญญาณ PDCCH จะสิ้นสุดที่กระบวนการนี้ โดยรูปที่ 4.79 เวกเตอร์ \mathbf{e} จะเป็นผลลัพธ์บิตข้อมูลขาออกที่มีความยาว $E = 432$ บิต ซึ่งจะมีความยาวน้อยกว่าเวกเตอร์ \mathbf{y} ที่ $N = 512$ บิต เนื่องจากการปรับอัตราหัสแบบฟังก์ชันเซอร์

4.2.2.3 ผลการทดสอบซอฟต์แวร์เข้ารหัสกรณีช่องสัญญาณ PUCCH ที่ความยาวบิตข้อมูลยาว

สำหรับผลการทดสอบของช่องสัญญาณ PUCCH ที่ความยาวบิตข้อมูลยาว หัวข้อในการเก็บผลจะถูกแบ่งออกตามจำนวนกระบวนการเข้ารหัสของช่องสัญญาณ PUCCH ที่ความยาวบิตข้อมูลยาว ซึ่งประกอบด้วย 8 กระบวนการ และรวมถึงผลลัพธ์ของกระบวนการโดยรวมดังนี้

- 1) กระบวนการทำงานโดยรวม
- 2) การแบ่งย่อยบล็อกรหัส
- 3) การเข้ารหัส CRC
- 4) การลำดับช่องสัญญาณย่อย
- 5) การเข้ารหัสโพลาร์
- 6) การแทรกสลับบล็อกย่อย
- 7) การปรับอัตราหัส
- 8) การแทรกสลับบิตรหัส
- 9) การต่อบล็อกรหัส

1) ผลการทดสอบกระบวนการทำงานโดยรวม

เมื่อผู้ใช้งานกรอกบิตข้อมูลขาเข้าหรือเวกเตอร์ \mathbf{a} แสดงดังรูปที่ 4.80 ระบบการเข้ารหัสโพลาร์สำหรับช่องสัญญาณ PUCCH จะให้ผลลัพธ์บิตข้อมูลขาออกแสดงดังรูปที่ 4.81

```

informationBits =
Columns 1 through 19
  1   1   1   0   1   0   1   0   0   0   1   0   0   0   0   0   1   1   1
Columns 20 through 38
  0   1   0   1   0   0   0   1   0   0   0   0   0   1   1   1   0   1   0
Columns 39 through 57
  1   0   0   0   1   0   0   0   0   0   1   1   1   0   1   0   1   0   0
Columns 58 through 76
  0   1   0   0   0   0   0   1   1   1   0   1   0   1   0   0   0   1   0
Columns 77 through 95
  0   0   0   0   1   1   1   0   1   0   1   0   0   0   1   0   0   0   0

```

รูปที่ 4.80 บิตข้อมูลขาเข้า

```

concatenatedBits =
Columns 1 through 19
  0   0   1   0   1   0   0   1   0   1   1   1   0   0   0   1   1   1   1
Columns 20 through 38
  0   1   1   1   1   1   0   1   0   0   1   1   1   1   0   1   0   1   0
Columns 39 through 57
  1   0   1   1   0   1   1   0   1   0   1   1   1   1   0   0   1   1   0
Columns 58 through 76
  0   1   1   1   0   0   1   1   0   1   0   0   1   0   1   0   1   1   0
Columns 77 through 95
  1   0   1   0   1   1   1   1   1   0   0   0   0   1   1   1   1   0   0

```

รูปที่ 4.81 บิตข้อมูลขาออก (บางส่วน)

โดยรูปที่ 4.81 บิตข้อมูลขาเข้าเวกเตอร์ **a** มีความยาว $A = 368$ บิต โดยช่องสัญญาณ PUCCH สามารถใช้ความยาวบิตข้อมูลได้ตั้งแต่ 12 ถึง 1706 บิต และรูปที่ 4.81 ผลลัพธ์บิตข้อมูลขาออกมีความยาว $G = 2176$ บิต โดยมีการแบ่งย่อยบิตกรหัสออกเป็น 2 ส่วน เช่นกัน

2) ผลการทดสอบกระบวนการแบ่งย่อยบิตกรหัส

เมื่อป้อนอินพุตเวกเตอร์ ความยาว A แสดงดังรูปที่ 4.80 ข้อมูลจะผ่านกระบวนการแบ่งย่อยบิตกรหัส และจะให้ผลลัพธ์เวกเตอร์ a' จำนวน 2 เวกเตอร์ ที่มีความยาว A' เท่ากัน แสดงดังรูปที่ 4.82 และ 4.83

```

segmentedBits =
Columns 1 through 19
  1  1  1  0  1  0  1  0  0  0  1  0  0  0  0  0  1  1  1
Columns 20 through 38
  0  1  0  1  0  0  0  1  0  0  0  0  0  1  1  1  0  1  0
Columns 39 through 57
  1  0  0  0  1  0  0  0  0  0  1  1  1  0  1  0  1  0  0
Columns 58 through 76
  0  1  0  0  0  0  0  1  1  1  0  1  0  1  0  0  0  1  0
Columns 77 through 95
  0  0  0  0  1  1  1  0  1  0  1  0  0  0  1  0  0  0  0

```

รูปที่ 4.82 เวกเตอร์ **a'** บล็อกแรก (บางส่วน)

```

segmentedBits =
Columns 1 through 19
  0  0  1  0  0  0  0  0  1  1  1  0  1  0  1  0  0  0  1
Columns 20 through 38
  0  0  0  0  0  1  1  1  0  1  0  1  0  0  0  1  0  0  0
Columns 39 through 57
  0  0  1  1  1  0  1  0  1  0  0  0  1  0  0  0  0  0  1
Columns 58 through 76
  1  1  0  1  0  1  0  0  0  1  0  0  0  0  0  1  1  1  0
Columns 77 through 95
  1  0  1  0  0  0  1  0  0  0  0  0  1  1  1  0  1  0  1

```

รูปที่ 4.83 เวกเตอร์ **a'** บล็อกท้าย (บางส่วน)

โดยรูปที่ 4.82 และ 4.83 เวกเตอร์ **a'** ที่ได้จากการแบ่งทั้ง 2 บล็อกจะมีความยาว $A'=184$ บิต ซึ่งมีขนาดครึ่งหนึ่งของความยาว A

3) ผลการทดสอบกระบวนการเข้ารหัส CRC

เวกเตอร์ **a'** ความยาว A' จากกระบวนการก่อนหน้าดังรูปที่ 4.82 และ 4.83 จะผ่านกระบวนการเข้ารหัส CRC และจะให้ผลลัพธ์เวกเตอร์ **c** ความยาว K แสดงดังรูปที่ 4.84 และ 4.85

```

infoCRCBits =
Columns 1 through 19
  1  1  1  0  1  0  1  0  0  0  1  0  0  0  0  0  1  1  1
Columns 20 through 38
  0  1  0  1  0  0  0  1  0  0  0  0  0  1  1  1  0  1  0
Columns 39 through 57
  1  0  0  0  1  0  0  0  0  0  1  1  1  0  1  0  1  0  0
Columns 58 through 76
  0  1  0  0  0  0  0  1  1  1  0  1  0  1  0  0  0  1  0
Columns 77 through 95
  0  0  0  0  1  1  1  0  1  0  1  0  0  0  1  0  0  0  0

```

รูปที่ 4.84 เวกเตอร์ **c** บล็อกแรก (บางส่วน)

```

infoCRCBits =
Columns 1 through 19
  0  0  1  0  0  0  0  0  1  1  1  0  1  0  1  0  0  0  1
Columns 20 through 38
  0  0  0  0  0  1  1  1  0  1  0  1  0  0  0  1  0  0  0
Columns 39 through 57
  0  0  1  1  1  0  1  0  1  0  0  0  1  0  0  0  0  0  1
Columns 58 through 76
  1  1  0  1  0  1  0  0  0  1  0  0  0  0  0  1  1  1  0
Columns 77 through 95
  1  0  1  0  0  0  1  0  0  0  0  0  1  1  1  0  1  0  1

```

รูปที่ 4.85 เวกเตอร์ **c** บล็อกท้าย (บางส่วน)

โดยรูปที่ 4.84 และ 4.85 เวกเตอร์ **c** มีความยาว $K=195$ บิต ประกอบด้วยเวกเตอร์ **a'** ความยาว $A'=184$ บิต และบิต CRC ความยาว $L=11$ บิต

4) ผลการทดสอบกระบวนการลำดับช่องสัญญาณย่อย

เวกเตอร์ **c** จะไม่มีกระบวนการสแครมบลิง CRC และกระบวนการแทรกสลับ CRC สำหรับช่องสัญญาณ PUCCH โดยจะผ่านไปยังกระบวนการลำดับช่องสัญญาณย่อยและให้ผลลัพธ์เวกเตอร์ **u** แสดงดังรูปที่ 4.86 และ 4.87

```

sourcewordBits =
Columns 1 through 19
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Columns 20 through 38
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Columns 39 through 57
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Columns 58 through 76
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Columns 77 through 95
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

```

รูปที่ 4.86 เวกเตอร์ **u** บล็อกแรก (บางส่วน)

```

sourcewordBits =
Columns 1 through 19
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Columns 20 through 38
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Columns 39 through 57
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Columns 58 through 76
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Columns 77 through 95
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0

```

รูปที่ 4.87 เวกเตอร์ **u** บล็อกท้าย (บางส่วน)

โดยรูปที่ 4.86 และ 4.87 เวกเตอร์ **u** มีความยาว $N = 1024$ บิต โดยบิตในเวกเตอร์ **c** จะอยู่ในตำแหน่งของเวกเตอร์ **u** ร่วมกับบิตซ้ำซ้ำ

5) ผลการทดสอบกระบวนการเข้ารหัสโพลาไร

เวกเตอร์ **u** จะไม่มีกระบวนการคำนวณบิตพาริตีสำหรับช่องสัญญาณ PUCCH ในกรณีความยาวบิตข้อมูลยาว โดยจะผ่านกระบวนการเข้ารหัสโพลาไร ซึ่งจะให้ผลลัพธ์เวกเตอร์ **d** แสดงดังรูปที่ 4.88 และ 4.89

```

motherCodewordBits =
Columns 1 through 19
  0  0  1  1  1  0  0  0  1  1  1  0  1  1  1  0  0  0  0
Columns 20 through 38
  0  0  0  1  0  1  1  0  1  1  0  0  0  0  0  0  1  1  0
Columns 39 through 57
  1  0  1  0  0  1  1  0  1  0  1  0  1  1  0  1  1  0  0
Columns 58 through 76
  0  1  1  0  1  0  1  0  0  1  1  1  1  0  0  0  1  1  1
Columns 77 through 95
  0  1  0  0  1  0  0  0  1  0  1  1  1  1  0  0  1  1  1

```

รูปที่ 4.88 เวกเตอร์ **d** บล็อกแรก (บางส่วน)

```

motherCodewordBits =
Columns 1 through 19
  1  1  1  1  0  1  1  0  1  1  0  0  1  1  0  0  1  1  0
Columns 20 through 38
  0  1  1  1  1  1  1  0  0  0  1  1  0  1  0  1  0  1  0
Columns 39 through 57
  0  1  1  1  0  0  0  1  1  0  1  0  0  1  0  0  1  1  0
Columns 58 through 76
  1  0  1  0  1  1  0  0  1  0  1  1  1  1  0  1  0  0  0
Columns 77 through 95
  1  1  1  1  0  1  1  1  0  1  1  0  1  0  1  0  0  1  1

```

รูปที่ 4.89 เวกเตอร์ **d** บล็อกท้าย (บางส่วน)

โดยรูปที่ 4.88 และ 4.89 เวกเตอร์ **d** มีความยาวเท่ากับเวกเตอร์ **u** ที่ $N = 1024$ บิต

6) ผลการทดสอบกระบวนการแทรกสลับบล็อกย่อย

หลังกระบวนการเข้ารหัสโพลาไร เวกเตอร์ **d** จะผ่านกระบวนการแทรกสลับบล็อกย่อยและให้ผลลัพธ์เวกเตอร์ **y** แสดงดังรูปที่ 4.90 และ 4.91

```

subBlockIntlcodewordBits =
Columns 1 through 19
  0  0  1  1  1  0  0  0  1  1  1  0  1  1  1  0  0  0  0
Columns 20 through 38
  0  0  0  1  0  1  1  0  1  1  0  0  0  0  0  0  1  1  0
Columns 39 through 57
  1  0  1  0  0  1  1  0  1  0  1  0  1  1  0  1  1  0  0
Columns 58 through 76
  0  1  1  0  1  0  1  0  0  1  1  1  1  0  0  0  1  1  1
Columns 77 through 95
  0  1  0  0  1  0  0  0  1  0  1  1  1  1  0  0  1  1  1

```

รูปที่ 4.90 เวกเตอร์ **y** บล็อกแรก (บางส่วน)

```

subBlockIntlcodewordBits =
Columns 1 through 19
  1  1  1  1  0  1  1  0  1  1  0  0  1  1  0  0  1  1  0
Columns 20 through 38
  0  1  1  1  1  1  1  0  0  0  1  1  0  1  0  1  0  1  0
Columns 39 through 57
  0  1  1  1  0  0  0  1  1  0  1  0  0  1  0  0  1  1  0
Columns 58 through 76
  1  0  1  0  1  1  0  0  1  0  1  1  1  1  0  1  0  0  0
Columns 77 through 95
  1  1  1  1  0  1  1  1  0  1  1  0  1  0  1  0  0  1  1

```

รูปที่ 4.91 เวกเตอร์ **y** บล็อกท้าย (บางส่วน)

โดยรูปที่ 4.90 และ 4.91 เวกเตอร์ **y** มีความยาวเท่ากับเวกเตอร์ **d** ที่ $N=1024$ บิต แต่จะมีลำดับบิตที่แตกต่างกันเนื่องเกิดการแทรกสลับ

7) ผลการทดสอบกระบวนการปรับอัตราหัส

เวกเตอร์ **y** จะผ่านกระบวนการปรับอัตราหัสและให้ผลลัพธ์เวกเตอร์ **e** แสดงดังรูปที่ 4.92 และ 4.93

```

codewordBits =
Columns 1 through 19
  0  0  1  1  1  0  0  0  1  1  1  0  1  1  1  0  0  0  0
Columns 20 through 38
  0  0  0  1  0  1  1  0  1  1  0  0  0  0  0  0  1  1  0
Columns 39 through 57
  1  0  1  0  0  1  1  0  1  0  1  0  1  1  0  1  1  0  0
Columns 58 through 76
  0  1  1  0  1  0  1  0  0  1  1  1  1  0  0  0  1  1  1
Columns 77 through 95
  0  1  0  0  1  0  0  0  1  0  1  1  1  1  0  0  1  1  1

```

รูปที่ 4.92 เวกเตอร์ **e** บล็อกแรก (บางส่วน)

```

codewordBits =
Columns 1 through 19
  1  1  1  1  0  1  1  0  1  1  0  0  1  1  0  0  1  1  0
Columns 20 through 38
  0  1  1  1  1  1  1  0  0  0  1  1  0  1  0  1  0  1  0
Columns 39 through 57
  0  1  1  1  0  0  0  1  1  0  1  0  0  1  0  0  1  1  0
Columns 58 through 76
  1  0  1  0  1  1  0  0  1  0  1  1  1  1  0  1  0  0  0
Columns 77 through 95
  1  1  1  1  0  1  1  1  0  1  1  0  1  0  1  0  0  1  1

```

รูปที่ 4.93 เวกเตอร์ **e** บล็อกท้าย (บางส่วน)

โดยรูปที่ 4.92 และ 4.93 เวกเตอร์ **e** จะมีความยาว $E=1088$ บิต ซึ่งจะมีความยาวมากกว่าเวกเตอร์ **y** ที่ $N=1024$ บิต เนื่องจากการปรับอัตรารหัสแบบส่งซ้ำ

8) ผลการทดสอบกระบวนการแทรกสลับบิตรหัส

หลังกระบวนการปรับอัตรารหัส เวกเตอร์ **e** จะผ่านกระบวนการแทรกสลับบิตรหัส ได้ผลลัพธ์เวกเตอร์ **f** แสดงดังรูปที่ 4.94 และ 4.95


```

intlCodewordBits =
Columns 1 through 19
  0  0  1  0  1  0  0  1  0  1  1  1  0  0  0  1  1  1  1
Columns 20 through 38
  0  1  1  1  1  1  0  1  0  0  1  1  1  1  0  1  0  1  0
Columns 39 through 57
  1  0  1  1  0  1  1  0  1  0  1  1  1  1  0  0  1  1  0
Columns 58 through 76
  0  1  1  1  0  0  1  1  0  1  0  0  1  0  1  0  1  1  0
Columns 77 through 95
  1  0  1  0  1  1  1  1  1  0  0  0  0  1  1  1  1  0  0

```

รูปที่ 4.94 เวกเตอร์ **f** บล็อกแรก (บางส่วน)

```

intlCodewordBits =
Columns 1 through 19
  1  0  1  1  1  0  0  0  0  1  1  1  0  0  1  0  0  1  1
Columns 20 through 38
  0  1  1  1  0  0  1  0  0  1  1  0  0  0  1  1  0  0  0
Columns 39 through 57
  1  1  1  1  1  0  1  0  1  0  1  0  0  0  0  1  0  1  0
Columns 58 through 76
  0  1  0  0  1  0  1  1  1  0  0  0  0  0  1  1  0  0  1
Columns 77 through 95
  0  0  1  0  1  0  1  0  1  1  1  0  1  0  1  1  1  0  1

```

รูปที่ 4.95 เวกเตอร์ **f** บล็อกท้าย (บางส่วน)

9) ผลการทดสอบกระบวนการต่อบล็อกรหัส

เวกเตอร์ **f** จะผ่านกระบวนการต่อบล็อกรหัสและให้ผลลัพธ์เวกเตอร์ **g** แสดงดังรูปที่ 4.96

```

concatenatedBits =
Columns 1 through 19
  0  0  1  0  1  0  0  1  0  1  1  1  0  0  0  1  1  1  1
Columns 20 through 38
  0  1  1  1  1  1  0  1  0  0  1  1  1  1  0  1  0  1  0
Columns 39 through 57
  1  0  1  1  0  1  1  0  1  0  1  1  1  1  0  0  1  1  0
Columns 58 through 76
  0  1  1  1  0  0  1  1  0  1  0  0  1  0  1  0  1  1  0
Columns 77 through 95
  1  0  1  0  1  1  1  1  1  0  0  0  0  1  1  1  1  0  0

```

รูปที่ 4.96 เวกเตอร์ **g** (บางส่วน)

ผลลัพธ์ของระบบการเข้ารหัสโพลาร์สำหรับช่องสัญญาณ PUCCH ที่ความยาวบิตข้อมูลยาวจะสิ้นสุดที่กระบวนการนี้ โดยรูปที่ 4.96 เวกเตอร์ **g** จะเป็นผลลัพธ์บิตข้อมูลออกที่มีความยาว $G = 2176$ บิต ซึ่งเกิดจากการนำเวกเตอร์ **f** ทั้งสองมาต่อกัน

4.2.2.4 ผลการทดสอบชุดซอฟต์แวร์เข้ารหัสกรณีช่องสัญญาณ PUCCH ที่ความยาวบิตข้อมูลสั้น

สำหรับผลการทดสอบของช่องสัญญาณ PUCCH ที่ความยาวบิตข้อมูลสั้น หัวข้อในการเก็บผลจะถูกแบ่งออกตามจำนวนกระบวนการเข้ารหัสของช่องสัญญาณ PUCCH ที่ความยาวบิตข้อมูลสั้นซึ่งประกอบด้วย 7 กระบวนการ และรวมถึงผลลัพธ์ของกระบวนการโดยรวมดังนี้

- 1) กระบวนการทำงานโดยรวม
- 2) การเข้ารหัส CRC
- 3) การลำดับช่องสัญญาณย่อย
- 4) การคำนวณบิตพาริตีตรวจสอบ
- 5) การเข้ารหัสโพลาร์
- 6) การแทรกสลับบล็อกย่อย
- 7) การปรับอัตรารหัส
- 8) การแทรกสลับบิตรหัส

1) ผลการทดสอบกระบวนการทำงานโดยรวม

เมื่อผู้ใช้งานกรอกบิตข้อมูลขาเข้าหรือเวกเตอร์ **a** แสดงดังรูปที่ 4.97 ระบบการเข้ารหัสโพลาร์สำหรับช่องสัญญาณ PUCCH จะให้ผลลัพธ์บิตข้อมูลขาออกแสดงดังรูปที่ 4.98

```
informationBits =
  1  1  1  0  1  0  1  0  0  0  1  0  0  0  0  0
```

รูปที่ 4.97 บิตข้อมูลขาเข้า

```
concatenatedBits =
Columns 1 through 19
  1  0  1  1  0  1  0  0  0  0  1  0  1  1  0  0  1  0  1
Columns 20 through 38
  1  0  1  1  1  1  0  0  0  1  1  0  1  1  0  1  0  0  0
Columns 39 through 57
  1  1  0  0  1  0  0  1  0  1  0  0  0  0  1  0  0  0  0
Columns 58 through 72
  0  0  0  0  1  0  0  0  1  0  1  1  1  0  0
```

รูปที่ 4.98 บิตข้อมูลขาออก (บางส่วน)

โดยรูปที่ 4.97 บิตข้อมูลขาเข้าเวกเตอร์ **a** มีความยาว $A=16$ บิต โดยช่องสัญญาณ PUCCH สามารถใช้ความยาวบิตข้อมูลได้ตั้งแต่ 12 ถึง 1706 บิต และรูปที่ 4.98 ผลลัพธ์บิตข้อมูลขาออกมีความยาว $E=72$ บิต

2) ผลการทดสอบกระบวนการเข้ารหัส CRC

เวกเตอร์ **a** ความยาว A ดังรูปที่ 4.97 จะไม่ผ่านกระบวนการแบ่งย่อยบล็อกรหัสสำหรับช่องสัญญาณ PUCCH ที่ความยาวบิตข้อมูลสั้น โดยจะผ่านกระบวนการเข้ารหัส CRC และจะให้ผลลัพธ์เวกเตอร์ **c** ความยาว K แสดงดังรูปที่ 4.99

```
infoCRCBits =
Columns 1 through 19
  1  1  1  0  1  0  1  0  0  0  1  0  0  0  0  0  1  1  1
Columns 20 through 22
  1  0  0
```

รูปที่ 4.99 เวกเตอร์ **c**

โดยรูปที่ 4.99 เวกเตอร์ **c** มีความยาว $K = 22$ บิต ประกอบด้วยเวกเตอร์ **a** ความยาว $A = 16$ บิต และบิต CRC ความยาว $L = 6$ บิต

3) ผลการทดสอบกระบวนการลำดับช่องสัญญาณย่อย

เวกเตอร์ **c** จะไม่มีกระบวนการสแครมบลิง CRC และกระบวนการแทรกสลับ CRC สำหรับช่องสัญญาณ PUCCH โดยจะผ่านไปยังกระบวนการลำดับช่องสัญญาณย่อยและให้ผลลัพธ์เวกเตอร์ **u** แสดงดังรูปที่ 4.100

```

sourcewordBitsWOParity =
Columns 1 through 19
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
Columns 20 through 38
    0    0    0    0    1    0    0    0    1    0    1    0    1    0    0    0    0    0
Columns 39 through 57
    0    0    0    0    0    1    0    0    0    0    0    0    0    1    0    0    0    0
Columns 58 through 64
    0    1    1    1    1    0    0

```

รูปที่ 4.100 เวกเตอร์ **u** ที่ได้จากการลำดับช่องสัญญาณย่อย

โดยรูปที่ 4.100 เวกเตอร์ **u** มีความยาว $N = 64$ บิต โดยบิตในเวกเตอร์ **c** จะอยู่ภายในตำแหน่งของเวกเตอร์ **u** ร่วมกับบิตซ้ำซ้ำ

4) ผลการทดสอบกระบวนการคำนวณบิตพาริตีตรวจสอบ

สำหรับช่องสัญญาณ PUCCH ที่ความยาวบิตข้อมูลสั้น $A < 20$ บิต เวกเตอร์ **u** จะมีการคำนวณบิตพาริตีตรวจสอบ ได้ผลลัพธ์เป็นเวกเตอร์ **u** ที่ความยาวเดิม ดังรูปที่ 4.101

```

sourcewordBits =
Columns 1 through 19
  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
Columns 20 through 38
  0  0  0  0  1  0  0  0  1  0  1  0  1  0  0  0  0  0
Columns 39 through 57
  0  0  0  0  0  1  0  0  0  0  0  0  0  1  1  0  0  0
Columns 58 through 64
  0  1  1  1  1  0  0

```

รูปที่ 4.101 เวกเตอร์ **u** ที่ได้จากการคำนวณบิตพาริตีตรวจสอบ

โดยรูปที่ 4.101 เวกเตอร์ **u** มีความยาว $N = 64$ บิต โดยเวกเตอร์ **c** จะมีการคำนวณบิตพาริตีเพิ่มเติม ซึ่งบิตจะแตกต่างจากเวกเตอร์ **u** ในกระบวนการก่อนหน้า

5) ผลการทดสอบกระบวนการเข้ารหัสโพลาร์

เวกเตอร์ **u** จะผ่านกระบวนการเข้ารหัสโพลาร์ ซึ่งจะให้ผลลัพธ์เวกเตอร์ **d** แสดงดังรูปที่

4.102

```

motherCodewordBits =
Columns 1 through 19
  1  0  1  0  0  0  0  0  0  0  1  0  0  1  1  1  0  1  0
Columns 20 through 38
  1  0  0  0  0  1  1  0  1  0  1  1  1  1  0  0  1  1  1
Columns 39 through 57
  0  0  1  1  1  0  0  1  0  0  0  1  1  0  1  1  0  0  0
Columns 58 through 64
  0  0  1  0  1  0  0

```

รูปที่ 4.102 เวกเตอร์ **d**

โดยรูปที่ 4.102 เวกเตอร์ **d** มีความยาวเท่ากับเวกเตอร์ **u** ที่ $N = 64$ บิต

6) ผลการทดสอบกระบวนการแทรกสลับบิตเล็กน้อย

หลังกระบวนการเข้ารหัสโพลาร์ เวกเตอร์ **d** จะผ่านกระบวนการแทรกสลับบิตเล็กน้อยและให้ผลลัพธ์เวกเตอร์ **y** แสดงดังรูปที่ 4.103

```

subBlockIntlcodewordBits =
Columns 1 through 19
  1  0  1  0  0  0  0  0  0  0  0  1  0  0  1  1  1  0  1  1
Columns 20 through 38
  0  0  1  0  1  0  0  1  1  0  0  0  0  1  1  1  1  0  1
Columns 39 through 57
  1  0  0  1  0  1  1  1  0  0  0  1  1  0  1  1  0  0  0
Columns 58 through 64
  0  0  1  0  1  0  0

```

รูปที่ 4.103 เวกเตอร์ **y**

โดยรูปที่ 4.103 เวกเตอร์ **y** มีความยาวเท่ากับเวกเตอร์ **d** ที่ $N = 64$ บิต แต่จะมีลำดับบิตที่แตกต่างกันเนื่องเกิดการแทรกสลับ

7) ผลการทดสอบกระบวนการปรับอัตราหัส

เวกเตอร์ **y** จะผ่านกระบวนการปรับอัตราหัสและให้ผลลัพธ์เวกเตอร์ **e** แสดงดังรูปที่

4.104

```

codewordBits =
Columns 1 through 19
  1  0  1  0  0  0  0  0  0  0  0  1  0  0  1  1  1  0  1  1
Columns 20 through 38
  0  0  1  0  1  0  0  1  1  0  0  0  0  1  1  1  1  0  1
Columns 39 through 57
  1  0  0  1  0  1  1  1  0  0  0  1  1  0  1  1  0  0  0
Columns 58 through 72
  0  0  1  0  1  0  0  1  0  1  0  0  0  0  0

```

รูปที่ 4.104 เวกเตอร์ **e**

โดยรูปที่ 4.104 เวกเตอร์ **e** จะมีความยาว $E = 72$ บิต ซึ่งจะมีความยาวมากกว่าเวกเตอร์ **y** ที่ $N = 1024$ บิต เนื่องจากการปรับอัตราหัสแบบส่งซ้ำ

8) ผลการทดสอบกระบวนการแทรกสลับบิตรหัส

หลังกระบวนการปรับอัตราหัส เวกเตอร์ **e** จะผ่านกระบวนการแทรกสลับบิตรหัส ได้ผลลัพธ์เวกเตอร์ **f** แสดงดังรูปที่ 4.105

```

intlCodewordBits =
Columns 1 through 19
  1  0  1  1  0  1  0  0  0  0  1  0  1  1  0  0  1  0  1
Columns 20 through 38
  1  0  1  1  1  1  0  0  0  1  1  0  1  1  0  1  0  0  0
Columns 39 through 57
  1  1  0  0  1  0  0  1  0  1  0  0  0  0  1  0  0  0  0
Columns 58 through 72
  0  0  0  0  1  0  0  0  1  0  1  1  1  0  0

```

รูปที่ 4.105 เวกเตอร์ **f**

ผลลัพธ์ของระบบการเข้ารหัสโพลาร์สำหรับช่องสัญญาณ PUCCH ที่ความยาวบิตข้อมูลสั้นจะสิ้นสุดที่กระบวนการนี้ โดยรูปที่ 4.105 เวกเตอร์ **f** จะเป็นผลลัพธ์บิตข้อมูลขาออกที่มีความยาว $E = 72$ บิต ซึ่งจะมีความยาวเท่ากับกระบวนการก่อนหน้า แต่มีลำดับบิตที่ต่างกัน เนื่องจากเกิดการแทรกสลับ

4.2.3 การใช้งานชุดซอฟต์แวร์ PYTHON และกำหนดพารามิเตอร์เริ่มต้น

การใช้งานชุดซอฟต์แวร์ภาษา PYTHON สำหรับการเข้ารหัสช่องสัญญาณชนิด Polar Codes สามารถดาวน์โหลดได้ที่เว็บไซต์ www.channelcoding.com ได้ไฟล์ที่มีชื่อว่า Polar_Python ซึ่งประกอบไปด้วยชุดซอฟต์แวร์ต่าง ๆ ได้แก่ polarEncoder และ PolarEncoderModule โดยแต่ละชุดซอฟต์แวร์มีรายละเอียดดังนี้

4.2.3.1 polarEncoder

ชุดซอฟต์แวร์ polarEncoder เป็นชุดซอฟต์แวร์หลักสำหรับใช้ในการกำหนดพารามิเตอร์และเรียกใช้โมดูลย่อยในชุดซอฟต์แวร์ PolarEncoderModule ซึ่งประกอบด้วย 4 ส่วน ได้แก่ ส่วนการนำเข้าโมดูลย่อยสำหรับเรียกใช้งานในโปรแกรมหลัก ส่วนการกำหนดค่าพารามิเตอร์ ส่วนการตรวจสอบความถูกต้องของอินพุต และส่วนการเรียกใช้ซอฟต์แวร์ โดยแต่ละส่วนมีรายละเอียดดังนี้

- 1) ส่วนการนำเข้าโมดูลย่อยสำหรับเรียกใช้ในโปรแกรมหลักประกอบด้วยการนำเข้าฟังก์ชัน numpy sys และ PolarEncoderModule โดยใช้คำสั่ง import
- 2) ส่วนการกำหนดค่าพารามิเตอร์ ในส่วนนี้จะกำหนดค่าพารามิเตอร์ที่สำคัญสำหรับการเข้ารหัส ดังนี้

- ประเภทของช่องสัญญาณ (ChannelFlag) กำหนดตัวเลขเพื่อแทนประเภทของช่องสัญญาณ โดยที่ “1” แทนประเภทของช่องสัญญาณแบบ PBCH “2” แทนประเภทของช่องสัญญาณแบบ PDCCH และ “3” แทนประเภทของช่องสัญญาณแบบ PUCCH
- ความยาวรหัส (codewordLength) กำหนดความยาวรหัสที่ต้องการหลังจากการเข้ารหัส
- เวกเตอร์ข้อมูล (InformationBits) กำหนดเวกเตอร์ของบิตข้อมูลที่ต้องการเข้ารหัส
- เวกเตอร์ RNTI (rntiBits) กำหนดเวกเตอร์ของบิต RNTI

3) ส่วนการตรวจสอบความถูกต้องของอินพุต ในส่วนนี้จะทำการตรวจสอบบิตข้อมูลและพารามิเตอร์ที่กำหนดในส่วนการกำหนดพารามิเตอร์

4) ส่วนการเรียกใช้ซอฟต์แวร์ จะเรียกใช้โมดูลย่อยภายในชุดซอฟต์แวร์ PolarEncoderModule ซึ่งประกอบด้วย CodeBlockSegmentation CRCAttachment PolarSequencing PolarEncoding และ CodeBitInterleaver

```

import numpy as np
import sys
import PolarEncoderModule as pe

channelFlag = 1
codewordLength = 864
informationBits = [1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0]
rntiBits = np.zeros([16,1])

def InputValidation(channelFlag, codewordLength, informationBits, rntiBits):
    if channelFlag == 1:
        if len(informationBits) != 32:
            sys.exit('Information bits length for PBCH must be 32')
        if codewordLength != 864:
            sys.exit('codeword length for PBCH must be 864')
    elif channelFlag == 2:
        if len(informationBits) < 1 or len(informationBits) > 140:
            sys.exit('Information bits length for PDCCH must be within 1 to 140')
        if codewordLength < ((len(informationBits)+24) or codewordLength > 8192:
            sys.exit('codeword length for PDCCH must be within ' + str(len(informationBits)+24) + ' to 8192')
        if len(rntiBits) != 16:
            sys.exit('RNTI bits length for PDCCH must be 16')
    elif channelFlag == 3:
        if len(informationBits) < 12 or len(informationBits) > 1780:
            sys.exit('Information bits length for PUCCH must be within 12 to 1780')
        if len(informationBits) < 20:
            if codewordLength < ((len(informationBits)+6) or codewordLength > 8192:
                sys.exit('codeword length for PUCCH must be within ' + str(len(informationBits)+6) + ' to 8192')
            else:
                if codewordLength < ((len(informationBits)+11) or codewordLength > 8192:
                    sys.exit('codeword length for PUCCH must be within ' + str(len(informationBits)+11) + ' to 8192')
                if codewordLength < ((len(informationBits)+24) or codewordLength > 8192:
                    sys.exit('codeword length for PUCCH must be within ' + str(len(informationBits)+24) + ' to 8192')
        else:
            sys.exit('please select the correct channel type - 1 for PBCH - 2 for PDCCH - 3 for PUCCH')

InputValidation(channelFlag, codewordLength, informationBits, rntiBits)
segmentedBits = pe.CodeBlockSegmentation(channelFlag, informationBits, codewordLength)
infoCRBits, infoCRBitsWInt, infoCRBitsWScrambling = pe.CRCAttachment(channelFlag, segmentedBits, rntiBits)
sourceBits, sourceBitsPwly, subBlockInterleavingPattern, rateMatchingPattern, generatorMatrix, frozenPositions = pe.PolarSequencing(channelFlag, infoCRBits, codewordLength, len(informationBits))
codewordBits, subBlockInterleavingPattern, motherCodewordBits = pe.PolarEncoding(sourceBits, subBlockInterleavingPattern, rateMatchingPattern, generatorMatrix)
concatenatedBits, infoCodewordBits = pe.CodeBitInterleaver(channelFlag, codewordBits)

```

รูปที่ 4.106 ตัวอย่างการเรียกใช้งานซอฟต์แวร์ภาษา PYTHON และการกำหนดพารามิเตอร์

จากรูปที่ 4.106 แสดงการเรียกใช้งานซอฟต์แวร์ภาษา PYTHON และการกำหนดพารามิเตอร์โดยใช้โปรแกรม Pycharm ซึ่งกำหนดให้ใช้ช่องสัญญาณแบบ PBCH มีความยาวรหัส 864 บิต มีบิตข้อมูล 32 บิต คือ [1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0] และมีบิต rnti ที่เป็น 0 ทั้งหมดจำนวน 16 บิต ในการเข้ารหัสจะเริ่มต้นจากการเรียกใช้ฟังก์ชัน InputValidation ในส่วนการตรวจสอบความถูกต้องของอินพุต จากนั้นจึงเข้าสู่

ส่วนการเรียกใช้ซอฟต์แวร์ โดยจะทำการเรียกใช้โมดูลย่อย ได้แก่ CodeBlockSegmentation CRCAttachment PolarSequencing PolarEncoding และ CodeBitInterleaver ตามลำดับ

4.2.3.1 PolarEncoderModule

ชุดซอฟต์แวร์ PolarEncoderModule เป็นชุดซอฟต์แวร์ที่ประกอบด้วยโมดูลย่อยที่ใช้สำหรับการเข้ารหัสโพลาร์ โดยรายละเอียดของแต่ละโมดูลอธิบายได้ดังนี้

- 1) CodeBlockSegmentation เป็นโมดูลสำหรับการแบ่งบิตข้อมูลเป็นบล็อกย่อย โดยโมดูลนี้จะใช้สำหรับช่องสัญญาณแบบ PUCCH เท่านั้น
- 2) CRCAttachment เป็นโมดูลสำหรับเข้ารหัส CRC ซึ่งใช้สำหรับทุกช่องสัญญาณ
- 3) PolarSequencing เป็นโมดูลสำหรับการกำหนดตำแหน่งของบิตข้อมูลและบิตซ้ำแซ็งก่อนการเข้ารหัสโพลาร์ตามมาตรฐาน 5G ซึ่งใช้สำหรับทุกช่องสัญญาณ แต่ในช่องสัญญาณแบบ PUCCH จะมีการคำนวณตำแหน่งพาริตีเพิ่มเติม
- 4) PolarEncoding เป็นโมดูลการเข้ารหัสโพลาร์ ซึ่งใช้สำหรับทุกช่องสัญญาณ
- 5) CodeBitInterleaver เป็นโมดูลสำหรับสับเปลี่ยนบิตของคำรหัสที่ได้จากการเข้ารหัสโพลาร์ ซึ่งใช้สำหรับช่องสัญญาณ PUCCH เท่านั้น

4.2.4 ผลการทดสอบชุดซอฟต์แวร์ PYTHON

ในส่วนนี้จะทำการทดสอบชุดซอฟต์แวร์การเข้ารหัสช่องสัญญาณชนิด Polar Codes โดยแบ่งออก 3 ช่องสัญญาณ ได้แก่ PBCH PDCCH และ PUCCH ซึ่งกำหนดให้มีบิตข้อมูล 32 บิต คือ [1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0] และมีความยาวรหัส 864 บิต ผลการทดสอบของแต่ละช่องสัญญาณแสดงได้ดังนี้

4.2.4.1 ช่องสัญญาณ PBCH

จากบล็อกไดอะแกรมการเข้ารหัสของช่องสัญญาณในรูปที่ 2.13 จะเห็นว่าช่องสัญญาณ PBCH ประกอบด้วยการเข้ารหัส CRC การแทรกสลับ CRC การลำดับช่องสัญญาณย่อย การเข้ารหัสโพลาร์ การแทรกสลับบล็อกย่อย และการปรับอัตรารหัส ซึ่งผลลัพธ์ของแต่ละบล็อกแสดงดังรูปที่ 4.107 - 4.112

```
[[1 1 1 0 1 0 1 0 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 0 0 1 1 1 0  
0 0 0 1 1 0 0 1 1 0 0 0 1 0 1 0 0 1 0 0]]
```

รูปที่ 4.107 ผลลัพธ์การเข้ารหัส CRC สำหรับช่องสัญญาณ PBCH


```

[[0. 1. 0. 1. 0. 0. 0. 1. 1. 1. 1. 0. 1. 0. 1. 0. 1. 1. 1. 1. 0. 1. 0. 0.
0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0. 0. 1. 1. 1. 1. 1. 0. 1. 0. 0. 0. 0.
1. 1. 1. 1. 1. 1. 0. 1. 1. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0. 1. 0. 1. 1.
1. 0. 1. 1. 0. 0. 0. 0. 1. 0. 1. 0. 1. 1. 1. 0. 0. 0. 0. 1. 1. 0. 1. 0.
0. 0. 0. 0. 0. 1. 0. 0. 1. 1. 1. 0. 1. 0. 1. 0. 1. 0. 1. 1. 0. 0. 0.
0. 0. 1. 0. 0. 0. 0. 0. 1. 0. 1. 0. 1. 1. 1. 0. 1. 1. 1. 0. 1. 0. 1. 0.
0. 0. 0. 0. 1. 0. 1. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 1. 0. 0. 0.
1. 1. 0. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0. 1. 0. 1.
1. 1. 1. 1. 0. 1. 0. 0. 1. 0. 1. 1. 0. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 1.
0. 0. 0. 1. 1. 0. 1. 0. 1. 1. 1. 1. 1. 0. 1. 0. 1. 1. 1. 0. 1. 0. 1.
1. 0. 1. 0. 0. 1. 1. 1. 0. 0. 1. 0. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 1. 0.
0. 0. 1. 0. 1. 0. 0. 0. 0. 0. 1. 1. 1. 0. 0. 0. 1. 1. 1. 0. 0. 0. 1. 1. 0.
1. 0. 0. 1. 0. 1. 0. 0. 0. 0. 0. 1. 1. 1. 0. 0. 1. 1. 0. 0. 1. 1. 1. 0.
0. 1. 0. 0. 1. 1. 0. 0. 1. 0. 1. 1. 0. 1. 1. 1. 1. 0. 1. 1. 0. 0.
1. 0. 0. 1. 1. 1. 0. 1. 1. 1. 0. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 0. 0.
1. 0. 0. 1. 1. 1. 0. 1. 0. 0. 1. 1. 0. 0. 1. 1. 0. 0. 0. 1. 1. 0. 0. 0.
1. 0. 1. 1. 1. 0. 0. 1. 0. 1. 1. 0. 1. 0. 1. 1. 1. 1. 0. 1. 1. 0. 0.
1. 0. 0. 1. 1. 1. 0. 1. 0. 0. 1. 1. 0. 0. 1. 1. 0. 0. 0. 1. 1. 1. 0. 0.
1. 0. 0. 1. 1. 1. 0. 0. 0. 1. 1. 0. 0. 0. 1. 1. 1. 0. 0. 0. 1. 1. 1. 0.
1. 1. 0. 0. 1. 1. 1. 0. 1. 0. 1. 1. 1. 0. 0. 1. 1. 0. 0. 1. 0. 1. 0. 0.
1. 1. 1. 0. 1. 1. 0. 0.]]

```

รูปที่ 4.110 ผลลัพธ์การเข้ารหัสโพลาร์สำหรับช่องสัญญาณ PBCH

จากรูปที่ 4.110 แสดงผลลัพธ์การเข้ารหัสโพลาร์ โดยใช้โมดูล PolarEncoding ซึ่งผลลัพธ์ที่ได้จะเป็นคำรหัสโพลาร์ที่มีความยาวรหัส 512 บิต

```

[[0. 1. 0. 1. 0. 0. 0. 1. 1. 1. 1. 0. 1. 0. 1. 0. 1. 1. 1. 1. 0. 1. 0. 0.
0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0. 0. 1. 1. 1. 1. 1. 0. 1. 0. 0. 0. 0.
0. 0. 0. 0. 1. 0. 1. 1. 1. 0. 1. 1. 0. 0. 0. 0. 1. 1. 1. 1. 1. 0. 1.
1. 0. 0. 0. 0. 1. 0. 1. 1. 0. 1. 1. 1. 0. 0. 0. 0. 1. 1. 0. 1. 0.
0. 0. 1. 0. 0. 0. 0. 1. 0. 1. 1. 1. 0. 1. 1. 1. 0. 1. 1. 0. 1. 0. 1. 0.
0. 1. 1. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 1. 1. 0. 0. 0. 0. 0. 1. 0. 1. 1.
0. 1. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 1. 1. 1. 1. 0. 0. 0. 1. 1. 0. 0.
0. 0. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0. 1. 0.
1. 1. 0. 0. 1. 1. 1. 0. 0. 1. 0. 0. 1. 0. 0. 1. 1. 1. 1. 0. 1. 0. 0.
1. 0. 1. 1. 0. 0. 0. 0. 0. 0. 1. 1. 1. 0. 0. 0. 0. 1. 1. 1. 1. 1. 0. 0.
0. 1. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 1. 0. 1. 0. 0. 1. 1. 1. 0. 1.
1. 1. 0. 1. 0. 1. 1. 0. 1. 1. 1. 1. 1. 1. 0. 1. 0. 1. 1. 1. 0. 1. 1. 0.
0. 0. 1. 0. 0. 0. 0. 0. 1. 1. 0. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 0. 0.
1. 0. 0. 1. 1. 1. 0. 1. 0. 0. 1. 1. 0. 0. 1. 1. 0. 0. 0. 1. 1. 1. 0. 0.
1. 0. 0. 0. 1. 1. 0. 0. 0. 1. 1. 0. 1. 0. 1. 1. 0. 0. 0. 1. 1. 1. 0. 0.
1. 1. 0. 0. 1. 1. 1. 0. 1. 1. 1. 1. 0. 0. 0. 1. 1. 0. 0. 0. 1. 1. 0. 0.
1. 1. 0. 0. 1. 0. 0. 1. 0. 1. 1. 0. 0. 0. 1. 0. 1. 1. 0. 1. 0. 1. 1. 0.
1. 1. 1. 0. 1. 1. 0. 0.]]

```

รูปที่ 4.111 ผลลัพธ์การแทรกสลับบล็อกย่อยสำหรับช่องสัญญาณ PBCH

จากรูปที่ 4.111 แสดงผลลัพธ์การแทรกสลับบล็อกย่อย ซึ่งอยู่ในโมดูล PolarEncoding โดยรูปแบบการแทรกสลับจะได้จากการใช้โมดูล PolarSequencing และในรูปที่ 4.112 แสดงผลลัพธ์

การปรับอัตรารหัสที่ได้จากโมดูล PolarEncoding โดยจะได้คำรหัสที่มีความยาว 864 บิตตามที่กำหนด

```
[[0. 1. 0. 1. 0. 0. 0. 1. 1. 1. 1. 0. 1. 0. 1. 0. 1. 1. 1. 1. 0. 1. 0. 0. 0.
0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0. 0. 1. 1. 1. 1. 1. 0. 1. 0. 0. 0. 0.
0. 0. 0. 0. 1. 0. 1. 1. 1. 0. 1. 1. 0. 0. 0. 0. 1. 1. 1. 1. 1. 1. 0. 1.
1. 0. 0. 0. 0. 1. 0. 1. 1. 0. 1. 0. 1. 1. 1. 0. 0. 0. 0. 1. 1. 0. 1. 0.
0. 0. 0. 0. 0. 1. 0. 0. 1. 1. 1. 0. 1. 0. 1. 0. 1. 0. 1. 1. 0. 0. 0.
0. 1. 1. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 1. 1. 0. 0. 0. 0. 0. 0. 1. 0. 1. 1.
0. 1. 0. 0. 0. 0. 0. 1. 1. 0. 0. 0. 1. 1. 1. 1. 0. 0. 0. 1. 1. 0. 0.
0. 1. 0. 1. 1. 0. 0. 0. 1. 1. 0. 1. 0. 0. 0. 0. 1. 0. 0. 1. 0. 1. 0. 0.
0. 0. 0. 1. 1. 1. 0. 0. 0. 0. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0. 1. 0. 1.
1. 1. 0. 0. 1. 1. 1. 0. 0. 1. 0. 0. 1. 0. 0. 1. 1. 1. 1. 1. 0. 1. 0. 0.
1. 0. 1. 1. 0. 0. 0. 0. 0. 0. 1. 1. 1. 0. 0. 0. 0. 1. 1. 1. 1. 1. 0. 0.
0. 1. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 1. 0. 1. 0. 1. 0. 1. 1. 0. 1.
0. 0. 1. 1. 0. 0. 0. 1. 1. 0. 1. 1. 1. 1. 0. 0. 1. 1. 0. 1. 0. 0. 1. 1. 1.
0. 0. 1. 0. 0. 0. 0. 0. 1. 1. 0. 1. 0. 1. 1. 1. 1. 0. 1. 1. 0. 0.
1. 0. 0. 1. 1. 1. 0. 1. 0. 0. 1. 0. 0. 1. 1. 0. 0. 0. 1. 1. 1. 0. 0. 0.
1. 0. 0. 0. 1. 1. 0. 0. 0. 1. 1. 0. 1. 1. 1. 1. 0. 0. 0. 1. 1. 0. 0. 0. 1.
0. 1. 0. 0. 1. 0. 0. 1. 0. 1. 1. 0. 0. 0. 1. 0. 1. 1. 0. 1. 0. 1. 1. 0.
1. 1. 0. 0. 1. 1. 1. 0. 1. 0. 1. 1. 1. 0. 0. 1. 1. 0. 0. 1. 0. 1. 0. 1. 0.
1. 1. 0. 1. 1. 0. 0. 0. 1. 0. 1. 0. 0. 0. 1. 1. 1. 1. 0. 1. 1. 0. 0. 0.
1. 1. 1. 1. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1. 1. 0. 1. 1. 0. 1. 0. 0. 0.
1. 1. 1. 1. 1. 0. 0. 0. 0. 0. 0. 1. 0. 1. 1. 1. 0. 1. 1. 0. 0. 0. 0.
1. 1. 1. 1. 1. 0. 1. 1. 0. 0. 0. 1. 0. 0. 0. 1. 0. 1. 1. 0. 1. 0. 1. 1. 0.
0. 0. 0. 1. 1. 0. 1. 0. 0. 0. 0. 0. 0. 1. 0. 0. 0. 0. 1. 1. 0. 0. 0. 1. 1.
1. 0. 0. 0. 1. 1. 0. 0. 0. 1. 0. 1. 1. 0. 0. 0. 1. 1. 0. 1. 0. 0. 0. 0.
1. 1. 1. 1. 0. 1. 0. 0. 1. 0. 1. 1. 0. 0. 0. 0. 0. 1. 1. 1. 0. 0. 0. 1. 1.
0. 1. 1. 1. 1. 0. 0. 0. 1. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 1. 0. 1. 0.
1. 0. 0. 1. 1. 0. 1. 1. 1. 0. 1. 0. 1. 1. 0. 1. 1. 1. 1. 1. 1. 1. 0. 1.
0. 1. 1. 1. 0. 1. 0. 1. 0. 0. 1. 1. 0. 0. 1. 1. 0. 0. 1. 1. 0. 1. 1. 0. 0. 1.]])
```

รูปที่ 4.112 ผลลัพธ์การปรับอัตรารหัสสำหรับช่องสัญญาณ PBCH

4.2.4.2 ช่องสัญญาณ PDCCH

จากบล็อกไดอะแกรมการเข้ารหัสของช่องสัญญาณในรูปที่ 2.13 จะเห็นว่าช่องสัญญาณ PDCCH ประกอบด้วยการเข้ารหัส CRC การสแควมบลิง CRC การแทรกสลับ CRC การลำดับช่องสัญญาณย่อย การเข้ารหัสโพลาไร์ การแทรกสลับบล็อกย่อย และการปรับอัตรารหัส ซึ่งผลลัพธ์ของแต่ละบล็อกแสดงดังรูปที่ 4.113 – 4.119

```
[[1 1 1 0 1 0 1 0 0 0 1 0 0 0 0 0 1 1 1 0 1 0 1 0 0 0 1 0 0 0 0 0 1 0 1 1
0 0 1 0 1 1 0 0 0 1 0 1 0 1 0 0 1 1 0 1]]
```

รูปที่ 4.113 ผลลัพธ์การเข้ารหัส CRC สำหรับช่องสัญญาณ PDCCH

จากรูปที่ 4.116 แสดงผลลัพธ์การลำดับช่องสัญญาณย่อย โดยใช้โมดูล PolarSequencing ซึ่งหลังจากทำการแทรกสลับ CRC คำรหัสดังกล่าวจะถูกแทรกบิตแก้ไขให้มีความยาว 512 บิต โดยกำหนดตำแหน่งของบิตแก้ไขตามลำดับความน่าเชื่อถือของช่องสัญญาณย่อยที่กำหนดไว้ในมาตรฐาน 5G

```
[[0. 0. 1. 0. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0. 1. 0.
0. 1. 1. 1. 1. 1. 0. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 0. 0. 0. 0. 0. 0. 0. 1.
1. 1. 1. 0. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 1. 0. 0. 1. 1. 1. 1. 0. 1. 1.
0. 1. 0. 0. 1. 0. 1. 1. 0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 1. 0. 0. 1. 1. 1.
1. 1. 0. 1. 1. 0. 0. 0. 0. 0. 1. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0.
1. 0. 1. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 0. 0. 0. 0. 1. 0. 0. 0. 1.
1. 0. 1. 1. 1. 0. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 0. 0. 0. 0. 0. 1. 0.
1. 0. 0. 0. 0. 0. 1. 0. 0. 0. 1. 1. 1. 1. 0. 0. 0. 0. 1. 0. 0. 1. 0.
1. 0. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 1. 0. 0. 1. 1. 1. 0. 0. 1. 0. 0. 1.
1. 0. 1. 1. 1. 0. 1. 1. 1. 0. 1. 1. 0. 1. 1. 1. 0. 0. 0. 1. 0. 0. 1. 0.
1. 1. 0. 1. 1. 1. 0. 1. 0. 1. 1. 1. 0. 0. 0. 1. 1. 0. 1. 1. 0. 0. 0. 1.
0. 1. 0. 0. 1. 1. 1. 0. 0. 1. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 0. 1. 0. 1.
1. 1. 1. 0. 1. 0. 1. 1. 1. 1. 0. 1. 0. 1. 1. 1. 1. 1. 0. 1. 0. 1. 1.
1. 1. 1. 0. 1. 0. 0. 0. 0. 1. 1. 1. 0. 1. 1. 1. 0. 1. 1. 1. 0. 1. 1.
1. 1. 1. 0. 1. 1. 0. 1. 1. 1. 0. 1. 1. 1. 0. 1. 1. 0. 0. 0. 1. 1. 1. 0.
1. 0. 1. 1. 0. 0. 0. 1. 1. 0. 1. 1. 0. 0. 0. 1. 0. 1. 0. 0. 1. 1. 0. 1.
0. 0. 1. 0. 1. 1. 0. 1. 1. 1. 0. 1. 1. 1. 0. 1. 0. 1. 0. 1. 1. 0. 1. 1.
1. 0. 0. 0. 0. 1. 1. 1. 1. 1. 0. 1. 0. 1. 0. 0. 1. 1. 1. 0. 1. 1.
0. 0. 0. 1. 0. 1. 0. 0. 1. 1. 1. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0.
0. 1. 1. 1. 1. 0. 1. 1.]]
```

รูปที่ 4.117 ผลลัพธ์การเข้ารหัสโพลาร์สำหรับช่องสัญญาณ PDCCH

จากรูปที่ 4.117 แสดงผลลัพธ์การเข้ารหัสโพลาร์ โดยใช้โมดูล PolarEncoding ซึ่งผลลัพธ์ที่ได้จะเป็นคำรหัสโพลาร์ที่มีความยาวรหัส 512 บิต

```
[[0. 0. 1. 0. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 0. 1. 0. 0. 0. 0. 1. 0.
0. 1. 1. 1. 1. 1. 0. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 0. 0. 0. 0. 0. 0. 1.
0. 0. 0. 1. 0. 0. 1. 0. 0. 0. 0. 1. 1. 0. 0. 0. 0. 0. 1. 0. 0. 1. 0. 0. 1.
1. 1. 0. 1. 1. 0. 0. 0. 0. 0. 1. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 1. 0. 0.
1. 0. 1. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 1. 1. 0. 0. 0. 0. 1. 0. 0. 0. 1.
0. 0. 0. 1. 0. 0. 1. 0. 1. 1. 0. 1. 1. 1. 0. 1. 1. 0. 1. 1. 1. 0. 1.
0. 1. 1. 1. 1. 1. 0. 1. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 1. 0. 1. 0. 0. 1.
1. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 1. 1. 1. 1. 0. 0. 0. 0. 1. 0. 0. 1. 0.
1. 1. 0. 1. 0. 0. 1. 0. 1. 1. 0. 1. 1. 1. 1. 0. 1. 0. 0. 0. 0. 1. 0. 0.
0. 1. 0. 0. 1. 0. 1. 1. 0. 1. 0. 0. 1. 0. 0. 0. 1. 0. 0. 0. 0. 1. 1. 1.
1. 1. 1. 0. 0. 1. 1. 1. 0. 0. 1. 0. 0. 1. 1. 1. 0. 0. 1. 0. 1. 0. 1. 1.
1. 1. 1. 0. 1. 0. 1. 1. 0. 0. 1. 0. 0. 1. 1. 1. 0. 0. 1. 0. 0. 1. 0. 0.
1. 1. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 0. 0. 0. 1. 0. 1. 1. 1. 0. 1. 1.
1. 0. 1. 1. 0. 0. 0. 1. 1. 0. 1. 1. 0. 0. 0. 1. 0. 1. 0. 0. 1. 1. 0. 1.
1. 0. 1. 1. 0. 1. 1. 1. 1. 0. 0. 0. 0. 1. 1. 1. 0. 0. 1. 0. 1. 1. 0. 1.
0. 0. 0. 1. 0. 1. 0. 0. 1. 1. 1. 0. 1. 0. 0. 0. 1. 1. 1. 0. 0. 0. 1. 0.
0. 1. 1. 1. 1. 0. 1. 1.]]
```

รูปที่ 4.118 ผลลัพธ์การแทรกสลับบิตก้อยสำหรับช่องสัญญาณ PDCCH

จากรูปที่ 4.124 แสดงผลลัพธ์การเข้ารหัสโพลาร์ โดยใช้โมดูล PolarEncoding ซึ่งผลลัพธ์ที่จะเป็น คำรหัสโพลาร์สองบล็อกที่มีความยาวรหัสบล็อกละ 256 บิต

```
[[0. 0. 0. 1. 0. 1. 0. 0. 1. 1. 1. 0. 0. 1. 0. 0. 1. 1. 1. 0. 0. 1. 0. 0.
 1. 0. 1. 1. 1. 1. 1. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 1. 0. 0. 1. 1. 1. 0.
 0. 1. 0. 0. 1. 1. 1. 0. 1. 0. 1. 1. 1. 1. 1. 0. 0. 1. 0. 0. 0. 0. 0. 1.
 1. 1. 1. 0. 1. 0. 1. 1. 1. 0. 1. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 1. 1.
 1. 0. 1. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 1. 1. 0. 1. 0. 0. 0. 0. 0. 1.
 1. 1. 1. 0. 1. 0. 1. 1. 1. 1. 1. 0. 1. 0. 1. 1. 0. 1. 0. 1. 0. 0. 0. 0. 1.
 0. 0. 0. 1. 1. 0. 1. 1. 1. 0. 1. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 1. 1.
 1. 0. 1. 1. 0. 0. 0. 1. 1. 1. 1. 0. 1. 0. 1. 1. 0. 1. 0. 1. 0. 0. 0. 0. 1.
 1. 0. 1. 1. 1. 1. 1. 0. 0. 1. 0. 0. 1. 1. 1. 0. 0. 1. 0. 0. 1. 1. 1. 0.
 0. 0. 0. 1. 0. 1. 0. 0. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 1. 0. 0. 1. 0. 0.
 1. 1. 1. 0. 0. 1. 0. 0. 0. 0. 0. 1. 0. 1. 0. 0.]
[0. 0. 0. 1. 0. 1. 0. 0. 1. 1. 1. 0. 0. 1. 0. 0. 1. 1. 1. 0. 0. 1. 0. 0.
 1. 0. 1. 1. 1. 1. 1. 0. 0. 0. 0. 1. 0. 1. 0. 0. 0. 1. 0. 0. 1. 1. 1. 0.
 0. 1. 0. 0. 1. 1. 1. 0. 1. 0. 1. 1. 1. 1. 1. 0. 0. 1. 0. 0. 0. 0. 0. 1.
 1. 1. 1. 0. 1. 0. 1. 1. 1. 0. 1. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 1. 1.
 1. 0. 1. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 1. 1. 0. 1. 0. 1. 0. 0. 0. 0. 1.
 0. 0. 0. 1. 1. 0. 1. 1. 1. 0. 1. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 1. 1.
 1. 0. 1. 1. 0. 0. 0. 1. 1. 1. 1. 0. 1. 0. 1. 1. 0. 1. 0. 1. 0. 0. 0. 0. 1.
 0. 0. 0. 1. 1. 0. 1. 1. 1. 0. 1. 1. 0. 0. 0. 1. 0. 0. 0. 1. 1. 0. 1. 1.
 1. 0. 1. 1. 0. 0. 0. 1. 1. 1. 1. 0. 1. 0. 1. 1. 0. 1. 0. 1. 0. 0. 0. 0. 1.
 0. 0. 0. 1. 0. 1. 0. 0. 1. 0. 1. 1. 1. 1. 1. 0. 1. 1. 1. 0. 0. 1. 0. 0.
 1. 1. 1. 0. 0. 1. 0. 0. 0. 0. 0. 1. 0. 1. 0. 0.]
```

รูปที่ 4.125 ผลลัพธ์การแทรกสลับบล็อกย่อยสำหรับช่องสัญญาณ PUCCH

จากรูปที่ 4.125 แสดงผลลัพธ์การแทรกสลับบล็อกย่อย ซึ่งอยู่ในโมดูล PolarEncoding โดยในที่นี้จะได้ผลลัพธ์เหมือนรูปที่ 4.124 เนื่องจากทั้งสองบล็อกมีบิตเหมือนกันทั้งหมด

| | 0 | 1 |
|----|---------|---------|
| 0 | 0.00000 | 0.00000 |
| 1 | 0.00000 | 0.00000 |
| 2 | 0.00000 | 0.00000 |
| 3 | 1.00000 | 1.00000 |
| 4 | 0.00000 | 0.00000 |
| 5 | 1.00000 | 1.00000 |
| 6 | 0.00000 | 0.00000 |
| 7 | 0.00000 | 0.00000 |
| 8 | 1.00000 | 1.00000 |
| 9 | 1.00000 | 1.00000 |
| 10 | 1.00000 | 1.00000 |
| 11 | 0.00000 | 0.00000 |
| 12 | 0.00000 | 0.00000 |
| 13 | 1.00000 | 1.00000 |
| 14 | 0.00000 | 0.00000 |
| 15 | 0.00000 | 0.00000 |
| 16 | 1.00000 | 1.00000 |
| 17 | 1.00000 | 1.00000 |
| 18 | 1.00000 | 1.00000 |
| 19 | 0.00000 | 0.00000 |
| 20 | 0.00000 | 0.00000 |

รูปที่ 4.126 ผลลัพธ์การปรับอัตรารหัสสำหรับช่องสัญญาณ PUCCH

จากรูปที่ 4.126 แสดงผลลัพธ์การปรับอัตรารหัสที่ได้จากโมดูล PolarEncoding ซึ่งแต่ละบล็อกจะอยู่ใน แนวคอลัมน์ โดยแต่ละคอลัมน์จะมีความยาว 864 บิต

| | 0 | 1 |
|----|---------|---------|
| 0 | 0.00000 | 0.00000 |
| 1 | 0.00000 | 0.00000 |
| 2 | 1.00000 | 1.00000 |
| 3 | 0.00000 | 0.00000 |
| 4 | 0.00000 | 0.00000 |
| 5 | 0.00000 | 0.00000 |
| 6 | 1.00000 | 1.00000 |
| 7 | 1.00000 | 1.00000 |
| 8 | 1.00000 | 1.00000 |
| 9 | 0.00000 | 0.00000 |
| 10 | 1.00000 | 1.00000 |
| 11 | 1.00000 | 1.00000 |
| 12 | 1.00000 | 1.00000 |
| 13 | 1.00000 | 1.00000 |
| 14 | 1.00000 | 1.00000 |
| 15 | 1.00000 | 1.00000 |
| 16 | 0.00000 | 0.00000 |
| 17 | 0.00000 | 0.00000 |
| 18 | 1.00000 | 1.00000 |
| 19 | 0.00000 | 0.00000 |
| 20 | 0.00000 | 0.00000 |

รูปที่ 4.127 ผลลัพธ์การแทรกสลับบิตรหัสสำหรับช่องสัญญาณ PUCCH

จากรูปที่ 4.127 แสดงผลลัพธ์การแทรกสลับรหัสที่ได้จากโมดูล CodeBitInterleaver ซึ่งแต่ละบล็อก ที่จัดอยู่ในแนวคอลัมน์จะถูกแทรกสลับบิตตามมาตรฐาน 5G

| | 0 |
|----|---------|
| 0 | 0.00000 |
| 1 | 0.00000 |
| 2 | 1.00000 |
| 3 | 0.00000 |
| 4 | 0.00000 |
| 5 | 0.00000 |
| 6 | 1.00000 |
| 7 | 1.00000 |
| 8 | 1.00000 |
| 9 | 0.00000 |
| 10 | 1.00000 |
| 11 | 1.00000 |
| 12 | 1.00000 |
| 13 | 1.00000 |
| 14 | 1.00000 |
| 15 | 1.00000 |
| 16 | 0.00000 |
| 17 | 0.00000 |
| 18 | 1.00000 |
| 19 | 0.00000 |
| 20 | 0.00000 |

รูปที่ 4.128 ผลลัพธ์การต่อบล็อกรหัสสำหรับช่องสัญญาณ PUCCH

จากรูปที่ 4.128 แสดงผลลัพธ์การต่อรหัสที่ได้จากโมดูล CodeBitInterleaver บล็อกรหัสทั้งหมด จะถูกนำมาต่อกันเป็นคำรหัสที่มีความยาว 1728 บิต

4.2.5 วิธีการใช้งานวงจรเข้ารหัส FPGA สำหรับการเข้ารหัสช่องสัญญาณ

วิธีการใช้งานวงจรเข้ารหัสโพลาร์ ด้วยโปรแกรม Quartus Prime Lite Edition และโปรแกรม ModelSim-Altera มีขั้นตอนเหมือนหัวข้อที่ 4.2.1 ทุกประการ

4.2.6 ผลการทดสอบวงจรเข้ารหัสโพลาร์ตามมาตรฐาน 5G

การเก็บผลการทดสอบวงจรเข้ารหัสโพลาร์ตามมาตรฐาน 5G จะทำการเก็บผลรูปร่างสัญญาณและค่าบิตของสัญญาณ จากโปรแกรม ModelSim-Altera และแบ่งส่วนตามประเภทของช่องสัญญาณ ซึ่งประกอบไปด้วย 3 ช่องสัญญาณ ช่องสัญญาณ PBCH ช่องสัญญาณ PDCCH และช่องสัญญาณ PUCCH

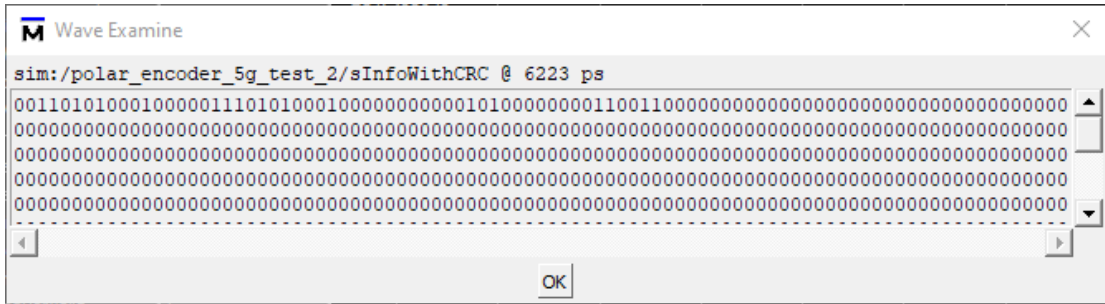
4.2.6.1 ผลการทดสอบวงจรเข้ารหัสกรณีช่องสัญญาณ PBCH

สำหรับผลการทดสอบของช่องสัญญาณ PBCH หัวข้อในการเก็บผลจะถูกแบ่งออกตามจำนวนกระบวนการเข้ารหัสของช่องสัญญาณ PBCH ซึ่งประกอบด้วย 6 กระบวนการ และรวมถึงผลลัพธ์ของกระบวนการโดยรวมดังนี้

- 1) กระบวนการทำงานโดยรวม
- 2) การเข้ารหัส CRC
- 3) การแทรกสลับ CRC
- 4) การลำดับช่องสัญญาณย่อย
- 5) การเข้ารหัสโพลาร์
- 6) การแทรกสลับบล็อกย่อย
- 7) การปรับอัตรารหัส

1) ผลการทดสอบกระบวนการทำงานโดยรวม

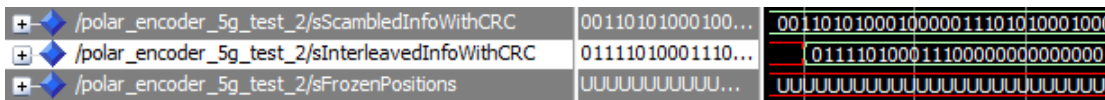
เมื่อวงจรเข้ารหัสที่ได้บิตข้อมูลเข้าเป็นรูปสัญญาณดังรูปที่ 4.129 และเลขบิตดังรูปที่ 4.130 วงจรเข้ารหัสจะให้ผลลัพธ์การเข้ารหัสบิตข้อมูลขาออกเป็นรูปสัญญาณดังรูปที่ 4.131 และเลขบิตดังรูปที่ 4.132



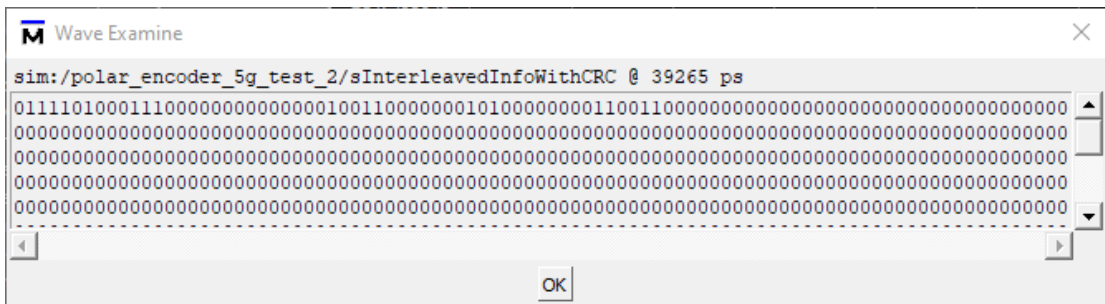
รูปที่ 4.134 บิต sInfoWithCRC (บางส่วน)

3) ผลการทดสอบกระบวนการแทรกสลับ CRC

หลังจากกระบวนการเข้ารหัส CRC ให้ผลลัพธ์สัญญาณดังรูปที่ 4.133 หรือแสดงเป็นเลขบิตดังรูปที่ 4.134 กระบวนการแทรกสลับ CRC จะให้ผลลัพธ์เป็นสัญญาณ sInterleavedInfoWithCRC ดังรูปที่ 4.135 หรือแสดงเป็นเลขบิตดังรูปที่ 4.136



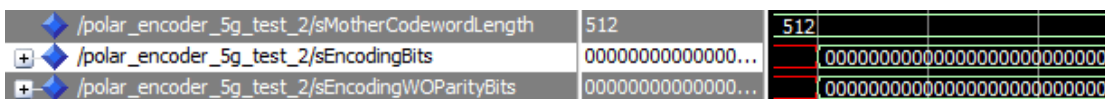
รูปที่ 4.135 สัญญาณ sInterleavedInfoWithCRC (บางส่วน)



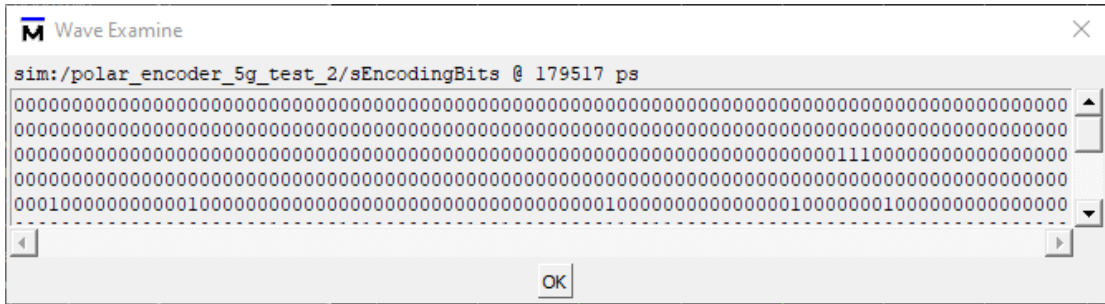
รูปที่ 4.136 บิต sInterleavedInfoWithCRC (บางส่วน)

4) ผลการทดสอบกระบวนการลำดับช่องสัญญาณย่อย

หลังจากกระบวนการแทรกสลับ CRC ให้ผลลัพธ์สัญญาณดังรูปที่ 4.135 หรือแสดงเป็นเลขบิตดังรูปที่ 4.136 กระบวนการลำดับช่องสัญญาณย่อยจะให้ผลลัพธ์เป็นสัญญาณ sEncodingBits ดังรูปที่ 4.137 หรือแสดงเป็นเลขบิตดังรูปที่ 4.138



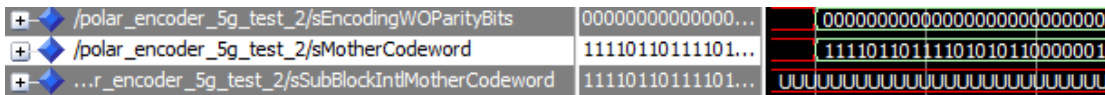
รูปที่ 4.137 สัญญาณ sEncodingBits (บางส่วน)



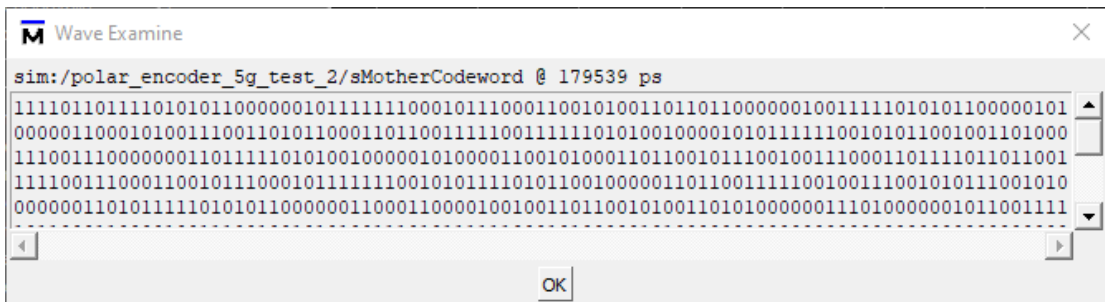
รูปที่ 4.138 บิต sEncodingBits (บางส่วน)

5) ผลการทดสอบกระบวนการเข้ารหัสโพลาร์

หลังจากกระบวนการลำดับช่องสัญญาณย่อยให้ผลลัพธ์สัญญาณดังรูปที่ 4.137 หรือแสดงเป็นเลขบิตดังรูปที่ 4.138 กระบวนการเข้ารหัสโพลาร์จะให้ผลลัพธ์เป็นสัญญาณ sMotherCodeword ดังรูปที่ 4.139 หรือแสดงเป็นเลขบิตดังรูปที่ 4.140



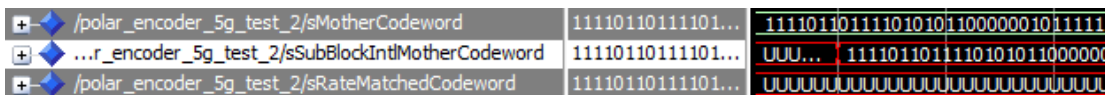
รูปที่ 4.139 สัญญาณ sMotherCodeword (บางส่วน)



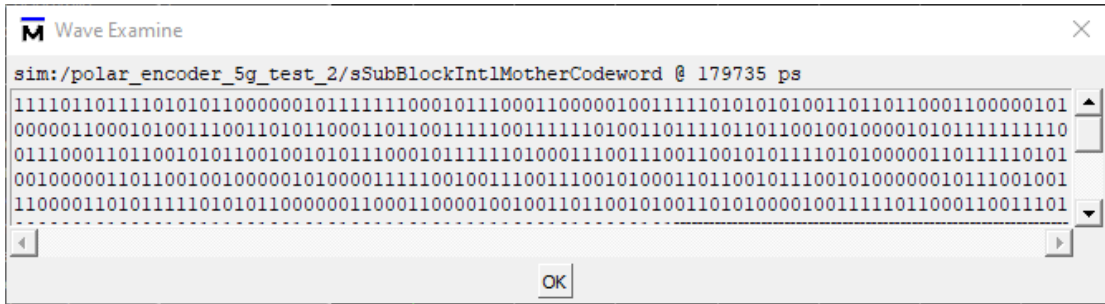
รูปที่ 4.140 บิต sMotherCodeword (บางส่วน)

6) ผลการทดสอบกระบวนการแทรกสลับบล็อกย่อย

หลังจากกระบวนการเข้ารหัสโพลาร์ให้ผลลัพธ์สัญญาณรูปที่ 4.139 หรือแสดงเป็นเลขบิตดังรูปที่ 4.140 กระบวนการแทรกสลับบล็อกย่อยจะให้ผลลัพธ์เป็นสัญญาณ sSubBlockIntlMotherCodeword ดังรูปที่ 4.141 หรือแสดงเป็นเลขบิตดังรูปที่ 4.142



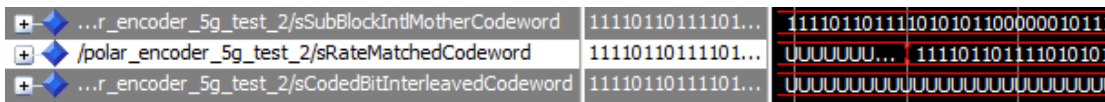
รูปที่ 4.141 สัญญาณ sSubBlockIntlMotherCodeword (บางส่วน)



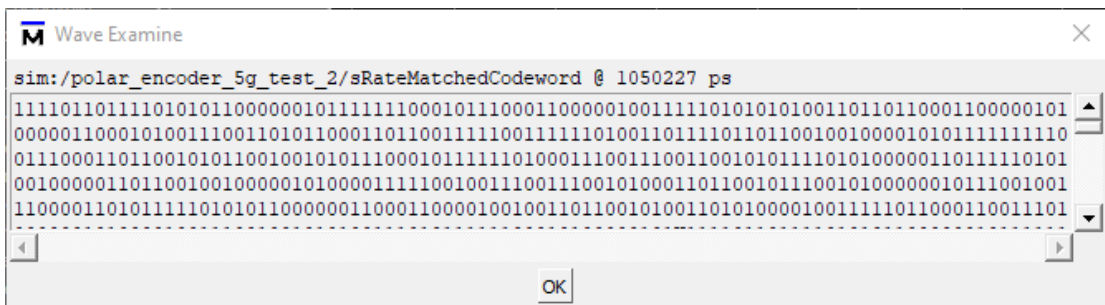
รูปที่ 4.142 บิต sSubBlockIntLMotherCodeword (บางส่วน)

7) ผลการทดสอบกระบวนการปรับอัตราหัส

หลังจากกระบวนการแทรกสลับบิตย่อยให้ผลลัพธ์ดังรูปที่ 4.141 หรือแสดงเป็นเลขบิตดังรูปที่ 4.142 กระบวนการปรับอัตราหัสจะให้ผลลัพธ์เป็นสัญญาณ sRateMatchedCodeword ดังรูปที่ 4.143 หรือแสดงเป็นเลขบิตดังรูปที่ 4.144



รูปที่ 4.143 สัญญาณ sRateMatchedCodeword (บางส่วน)



รูปที่ 4.144 บิต sRateMatchedCodeword (บางส่วน)

4.2.6.2 ผลการทดสอบวงจรเข้ารหัสกรณีช่องสัญญาณ PDCCH

สำหรับผลการทดสอบของช่องสัญญาณ PDCCH หัวข้อในการเก็บผลจะถูกแบ่งออกตามจำนวนกระบวนการเข้ารหัสของช่องสัญญาณ PDCCH ซึ่งประกอบด้วย 7 กระบวนการ และรวมถึงผลลัพธ์ของกระบวนการโดยรวมดังนี้

- 1) กระบวนการทำงานโดยรวม
- 2) การเข้ารหัส CRC
- 3) การสแครมบลิง CRC

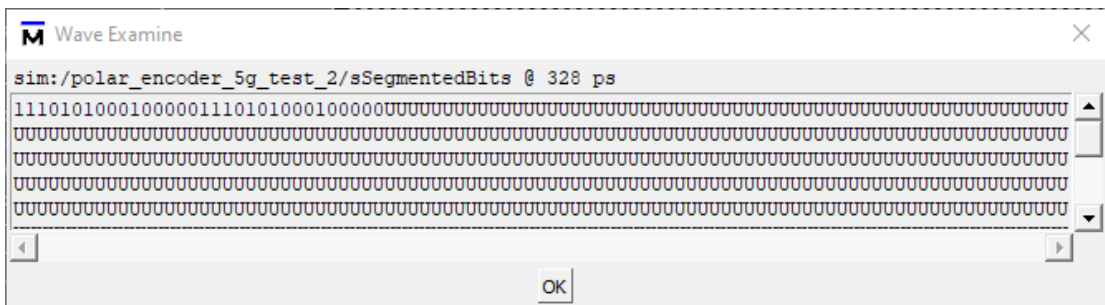
- 4) การแทรกสลับ CRC
- 5) การลำดับช่องสัญญาณย่อย
- 6) การเข้ารหัสโพลาไร
- 7) การแทรกสลับบล็อกย่อย
- 8) การปรับอัตรารหัส

1) ผลการทดสอบกระบวนการทำงานโดยรวม

เมื่อบริการเข้ารหัสที่ได้บิตข้อมูลขาเข้าเป็นรูปสัญญาณดังรูปที่ 4.145 และเลขบิตดังรูปที่ 4.146 วงจรเข้ารหัสจะให้ผลลัพธ์การเข้ารหัสบิตข้อมูลขาออกเป็นรูปสัญญาณดังรูปที่ 4.147 และเลขบิตดังรูปที่ 4.148

| | | | | | |
|--|-------------------|------|------------|------------|--------|
| /polar_encoder_5g_test_2/sBaseGraphSize | 0 | 0 | | | |
| /polar_encoder_5g_test_2/sSegmentedBits | 11101010001000... | U... | 1110101000 | 1000001110 | 101000 |
| /polar_encoder_5g_test_2/sCodeBlockSegmentationEn... | 0 | | | | |

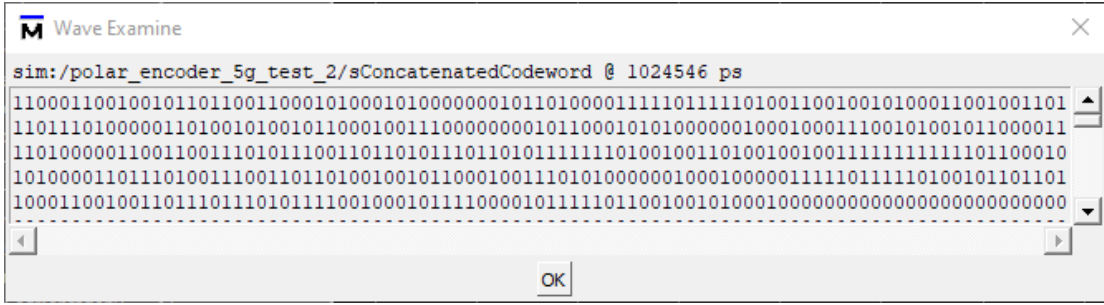
รูปที่ 4.145 สัญญาณบิตข้อมูลขาเข้า (บางส่วน)



รูปที่ 4.146 บิตข้อมูลขาเข้า (บางส่วน)

| | | | | |
|--|---------------|---------------------------------|--|--|
| /polar_encoder_5g_test_2/sCodedBitInterleavedCode... | UUUUUUUUUU... | 1100011001001011011001100010100 | | |
| /polar_encoder_5g_test_2/sConcatenatedCodeword | UUUUUUUUUU... | 110001100100101101100110001 | | |
| /polar_encoder_5g_test_2/sRXPackagingValid | 0 | | | |

รูปที่ 4.147 สัญญาณบิตข้อมูลขาออก (บางส่วน)



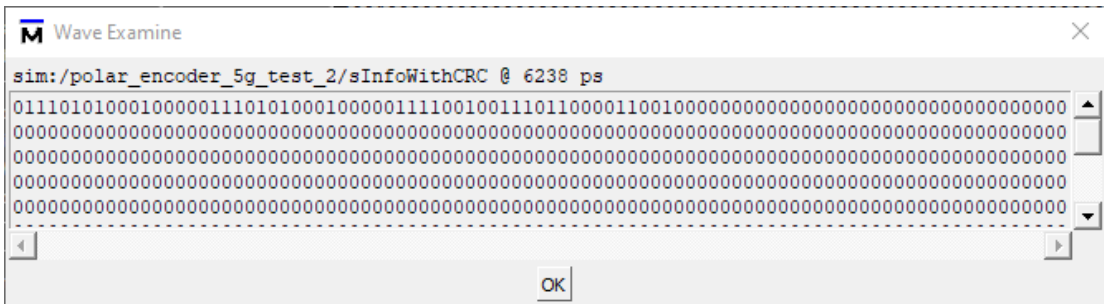
รูปที่ 4.148 บิตข้อมูลขาออก (บางส่วน)

2) ผลการทดสอบกระบวนการเข้ารหัส CRC

เมื่อวงจรเข้ารหัสรับสัญญาณบิตข้อมูลขาเข้าดังรูปที่ 4.145 หรือแสดงเป็นเลขบิตดังรูปที่ 4.146 กระบวนการเข้ารหัส CRC จะให้ผลลัพธ์เป็นสัญญาณ sInfoWithCRC ดังรูปที่ 4.149 หรือแสดงเป็นเลขบิตดังรูปที่ 4.150

| | | | | |
|--|-------------------|--|-----------------------------|--|
| /polar_encoder_5g_test_2/sCodeBlockSegmentationEn... | 0 | | | |
| + /polar_encoder_5g_test_2/sInfoWithCRC | 01110101000100... | | 011101010001000001110101000 | |
| + /polar_encoder_5g_test_2/sInfoWithCRCLength | 56 | | 56 | |

รูปที่ 4.149 สัญญาณ sInfoWithCRC (บางส่วน)



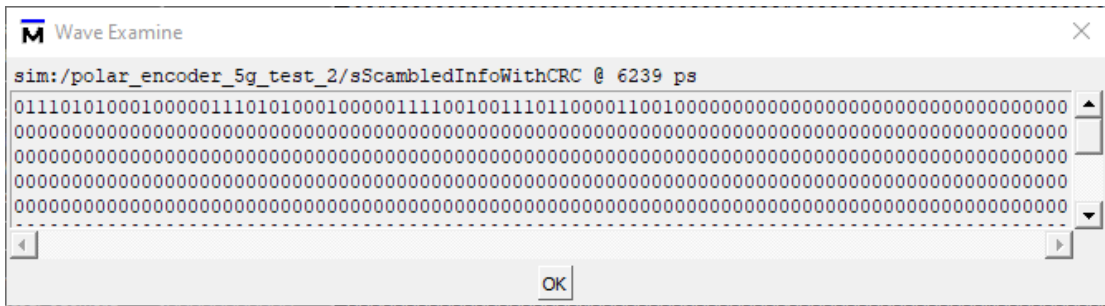
รูปที่ 4.150 บิต sInfoWithCRC (บางส่วน)

3) ผลการทดสอบกระบวนการสแครมbling CRC

หลังจากกระบวนการเข้ารหัส CRC ให้ผลลัพธ์สัญญาณดังรูปที่ 4.149 หรือแสดงเป็นเลขบิตดังรูปที่ 4.150 กระบวนการสแครมbling CRC จะให้ผลลัพธ์เป็นสัญญาณ sScambledInfoWithCRC ดังรูปที่ 4.151 หรือแสดงเป็นเลขบิตดังรูปที่ 4.152

| | | | | |
|--|-------------------|--|--------------------------------|--|
| + /polar_encoder_5g_test_2/sRNTIBits | 0000000000000000 | | 0000000000000000 | |
| + /polar_encoder_5g_test_2/sScambledInfoWithCRC | 01110101000100... | | 011101010001000001110101000 | |
| + /polar_encoder_5g_test_2/sInterleavedInfoWithCRC | 01111010001110... | | bbbbbbbbbbbbbbbbbbbbbbbbbbbbbb | |

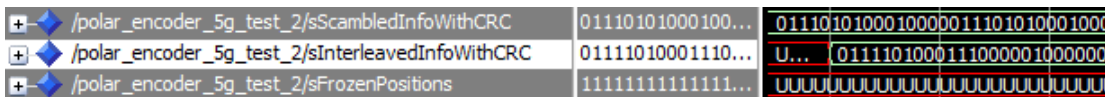
รูปที่ 4.151 สัญญาณ sScambledInfoWithCRC (บางส่วน)



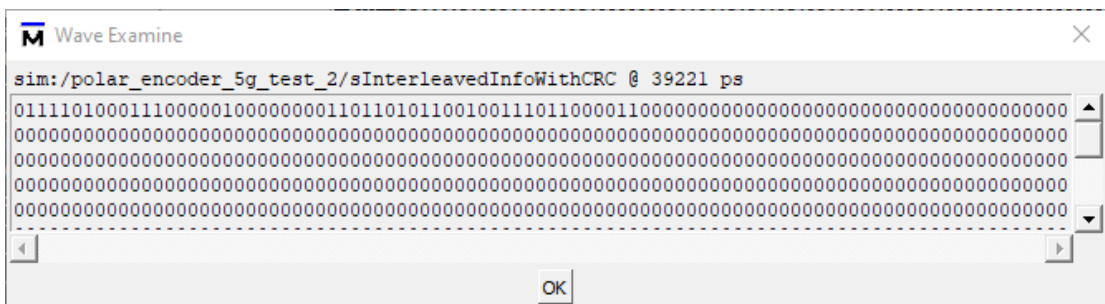
รูปที่ 4.152 บิต sScrambledInfoWithCRC (บางส่วน)

4) ผลการทดสอบกระบวนการแทรกสลับ CRC

หลังจากกระบวนการสแครมบลิง CRC ให้ผลลัพธ์สัญญาณดังรูปที่ 4.151 หรือแสดงเป็นเลขบิตดังรูปที่ 4.152 กระบวนการแทรกสลับ CRC จะให้ผลลัพธ์เป็นสัญญาณ sInterleavedInfoWithCRC ดังรูปที่ 4.153 หรือแสดงเป็นเลขบิตดังรูปที่ 4.154



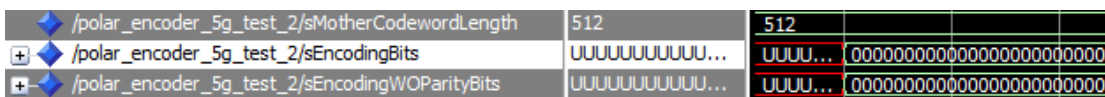
รูปที่ 4.153 สัญญาณ sInterleavedInfoWithCRC (บางส่วน)



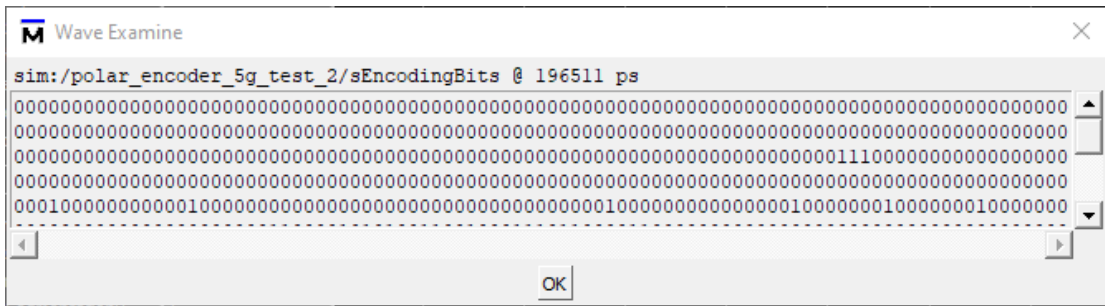
รูปที่ 4.154 บิต sInterleavedInfoWithCRC (บางส่วน)

5) ผลการทดสอบกระบวนการลำดับช่องสัญญาณย่อย

หลังจากกระบวนการแทรกสลับ CRC ให้ผลลัพธ์สัญญาณดังรูปที่ 4.153 หรือแสดงเป็นเลขบิตดังรูปที่ 4.154 กระบวนการลำดับช่องสัญญาณย่อยจะให้ผลลัพธ์เป็นสัญญาณ sEncodingBits ดังรูปที่ 4.155 หรือแสดงเป็นเลขบิตดังรูปที่ 4.156



รูปที่ 4.155 สัญญาณ sEncodingBits (บางส่วน)



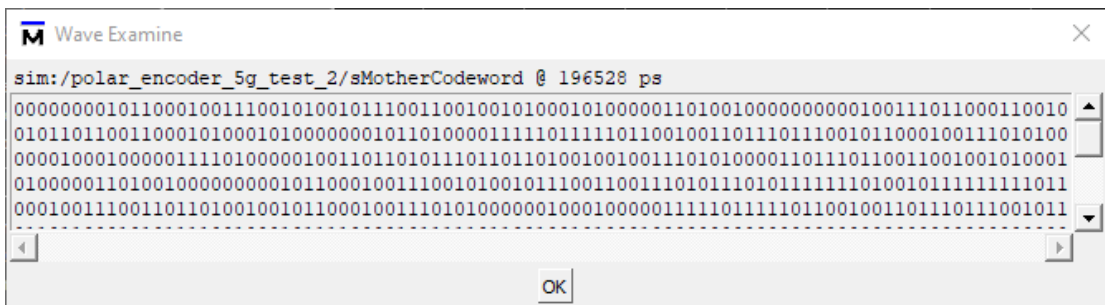
รูปที่ 4.156 บิต sEncodingBits (บางส่วน)

6) ผลการทดสอบกระบวนการเข้ารหัสโพลาร์

หลังจากกระบวนการลำดับช่องสัญญาณย่อยให้ผลลัพธ์สัญญาณดังรูปที่ 4.155 หรือแสดงเป็นเลขบิตดังรูปที่ 4.156 กระบวนการเข้ารหัสโพลาร์จะให้ผลลัพธ์เป็นสัญญาณ sMotherCodeword ดังรูปที่ 4.157 หรือแสดงเป็นเลขบิตดังรูปที่ 4.158



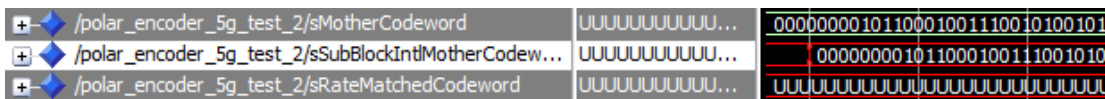
รูปที่ 4.157 สัญญาณ sMotherCodeword (บางส่วน)



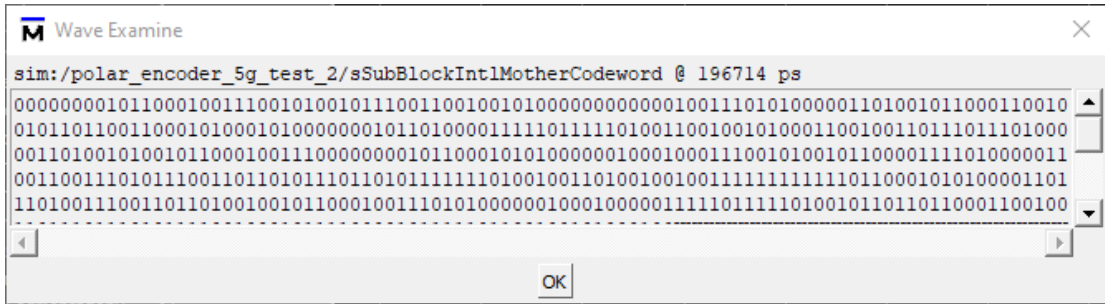
รูปที่ 4.158 บิต sMotherCodeword (บางส่วน)

7) ผลการทดสอบกระบวนการแทรกสลับบล็อกย่อย

หลังจากกระบวนการเข้ารหัสโพลาร์ให้ผลลัพธ์สัญญาณรูปที่ 4.157 หรือแสดงเป็นเลขบิตดังรูปที่ 4.158 กระบวนการแทรกสลับบล็อกย่อยจะให้ผลลัพธ์เป็นสัญญาณ sSubBlockIntlMotherCodeword ดังรูปที่ 4.159 หรือแสดงเป็นเลขบิตดังรูปที่ 4.160



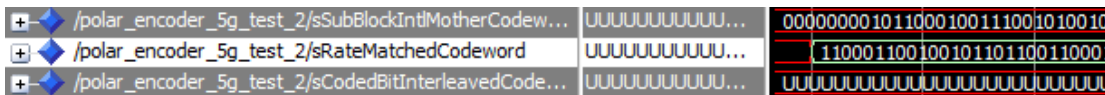
รูปที่ 4.159 สัญญาณ sSubBlockIntlMotherCodeword (บางส่วน)



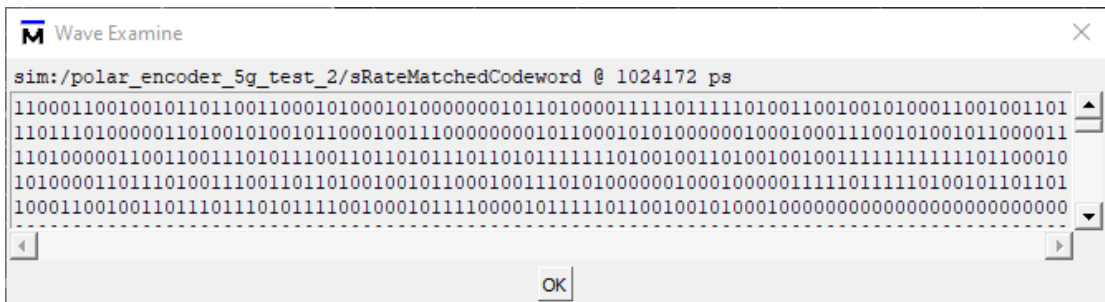
รูปที่ 4.160 บิต sSubBlockIntLMotherCodeword (บางส่วน)

8) ผลการทดสอบกระบวนการปรับอัตรารหัส

หลังจากกระบวนการแทรกสลับบิตก้อยยให้ผลลัพธ์ดังรูปที่ 4.159 หรือแสดงเป็นเลขบิตดังรูปที่ 4.160 กระบวนการปรับอัตรารหัสจะให้ผลลัพธ์เป็นสัญญาณ sRateMatchedCodeword ดังรูปที่ 4.161 หรือแสดงเป็นเลขบิตดังรูปที่ 4.162



รูปที่ 4.161 สัญญาณ sRateMatchedCodeword (บางส่วน)



รูปที่ 4.162 บิต sRateMatchedCodeword (บางส่วน)

4.2.6.3 ผลการทดสอบวงจรเข้ารหัสกรณีช่องสัญญาณ PUCCH ที่ความยาวบิต

ข้อมูลยาว

สำหรับผลการทดสอบของช่องสัญญาณ PUCCH ที่ความยาวบิตข้อมูลยาว หัวข้อในการเก็บผลจะถูกแบ่งออกตามจำนวนกระบวนการเข้ารหัสของช่องสัญญาณ PUCCH ที่ความยาวบิตข้อมูลยาว ซึ่งประกอบด้วย 8 กระบวนการ และรวมถึงผลลัพธ์ของกระบวนการโดยรวมดังนี้

- 1) กระบวนการทำงานโดยรวม
- 2) การแบ่งย่อยบิตกรหัส

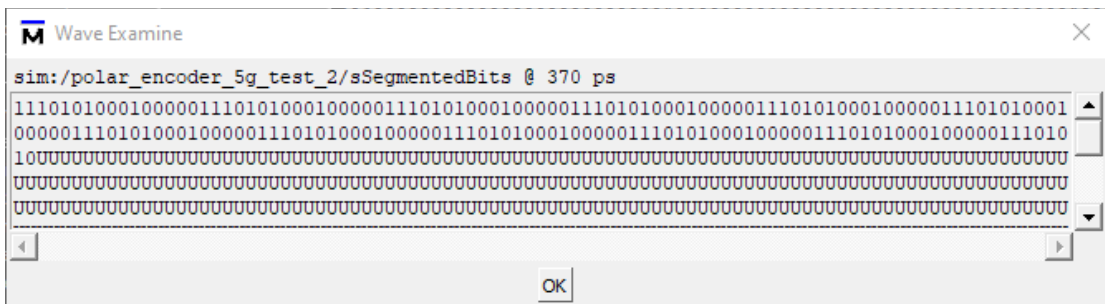
- 3) การเข้ารหัส CRC
- 4) การลำดับช่องสัญญาณย่อย
- 5) การเข้ารหัสโพลาไร
- 6) การแทรกสลับบล็อกย่อย
- 7) การปรับอัตรารหัส
- 8) การแทรกสลับบิตรหัส
- 9) การต่อบล็อกรหัส

1) ผลการทดสอบกระบวนการทำงานโดยรวม

เมื่อวงจรเข้ารหัสที่ได้บิตข้อมูลขาเข้าเป็นรูปสัญญาณดังรูปที่ 4.163 และเลขบิตดังรูปที่ 4.164 วงจรเข้ารหัสจะให้ผลลัพธ์การเข้ารหัสบิตข้อมูลขาออกเป็นรูปสัญญาณดังรูปที่ 4.165 และเลขบิตดังรูปที่ 4.166

| | | | | | |
|-----|---|-------------------|-----------------------------|--|--|
| ◆ | /polar_encoder_5g_test_2/sBaseGraphSize | 0 | 0 | | |
| + ◆ | /polar_encoder_5g_test_2/sSegmentedBits | 00100000111010... | 111010100010000011101010001 | | |
| ◆ | /polar_encoder_5g_test_2/sCodeBlockSegmentationE... | 1 | | | |

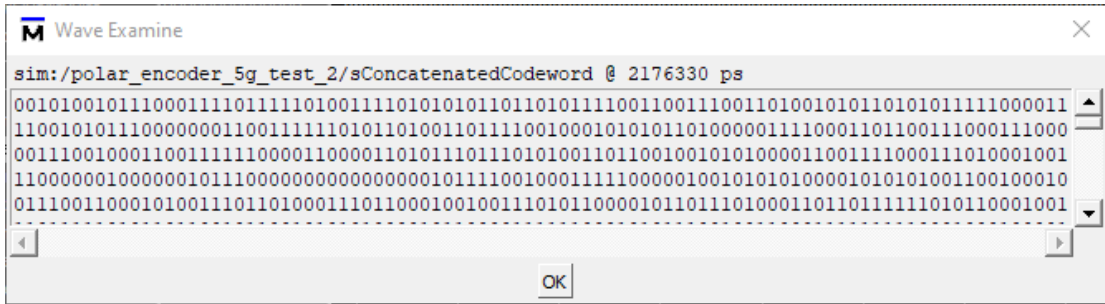
รูปที่ 4.163 สัญญาณบิตข้อมูลขาเข้า (บางส่วน)



รูปที่ 4.164 บิตข้อมูลขาเข้า (บางส่วน)

| | | | | | |
|-----|--|-------------------|-------------------------------|--|--|
| + ◆ | /polar_encoder_5g_test_2/sCodedBitInterleavedCode... | 10111000011100... | 10111000011100100110111001001 | | |
| + ◆ | /polar_encoder_5g_test_2/sConcatenatedCodeword | 00101001011100... | 0010100101110001111011111010 | | |
| ◆ | /polar_encoder_5g_test_2/sRXPackagingValid | 0 | | | |

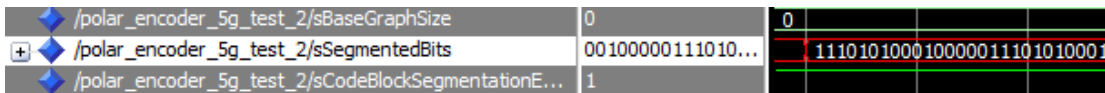
รูปที่ 4.165 สัญญาณบิตข้อมูลขาออก (บางส่วน)



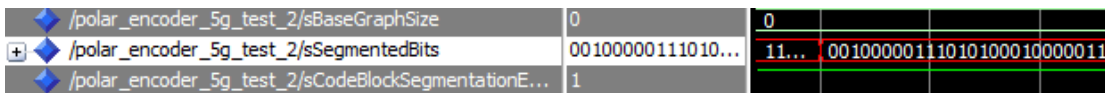
รูปที่ 4.166 บิตข้อมูลขาออก (บางส่วน)

2) ผลการทดสอบกระบวนการแบ่งย่อยบล็อกรหัส

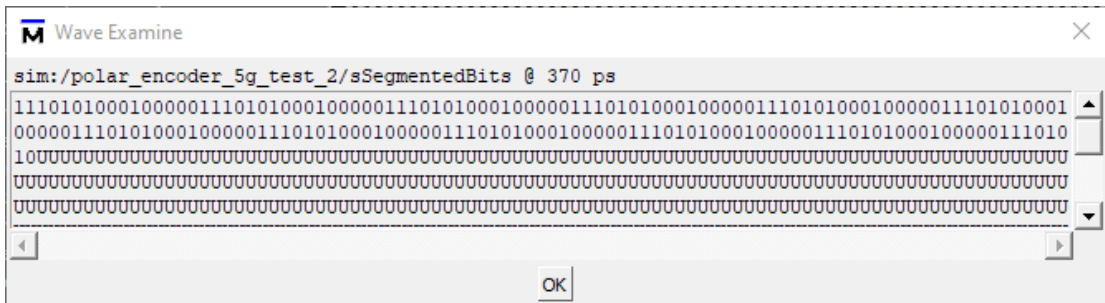
เมื่อวงจรเข้ารหัสรับสัญญาณบิตข้อมูลขาเข้าดังรูปที่ 4.165 หรือแสดงเป็นเลขบิตดังรูปที่ 4.166 กระบวนการแบ่งย่อยบล็อกรหัสจะให้ผลลัพธ์เป็นสัญญาณ sSegmentedBits ดังรูปที่ 4.167 และ 4.168 หรือแสดงเป็นเลขบิตดังรูปที่ 4.169 และ 4.170



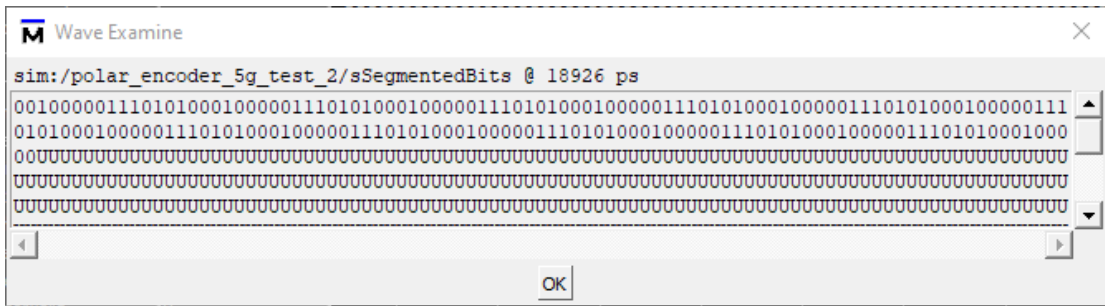
รูปที่ 4.167 สัญญาณ sSegmentedBits บล็อกแรก (บางส่วน)



รูปที่ 4.168 สัญญาณ sSegmentedBits บล็อกท้าย (บางส่วน)



รูปที่ 4.169 บิต sSegmentedBits บล็อกแรก (บางส่วน)



รูปที่ 4.170 บิต sSegmentedBits บล็อกท้าย (บางส่วน)

3) ผลการทดสอบกระบวนการเข้ารหัส CRC

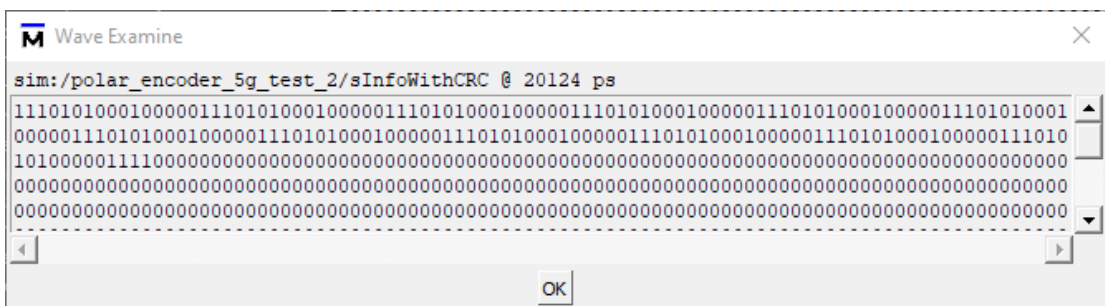
หลังจากกระบวนการแบ่งย่อยบล็อกรหัสให้ผลลัพธ์สัญญาณดังรูปที่ 4.167 หรือแสดงเป็นเลขบิตดังรูปที่ 4.169 กระบวนการเข้ารหัส CRC จะให้ผลลัพธ์เป็นสัญญาณ sInfoWithCRC ดังรูปที่ 4.171 และ 4.172 หรือแสดงเป็นเลขบิตดังรูปที่ 4.173 และ 4.174

| | | | | | |
|---|-------------------|--------|----------------------------|--|--|
| /polar_encoder_5g_test_2/sCodeBlockSegmentationE... | 1 | | | | |
| + /polar_encoder_5g_test_2/sInfoWithCRC | 00100000111010... | UUU... | 11101010001000001110101000 | | |
| /polar_encoder_5g_test_2/sInfoWithCRCLength | 195 | 195 | | | |

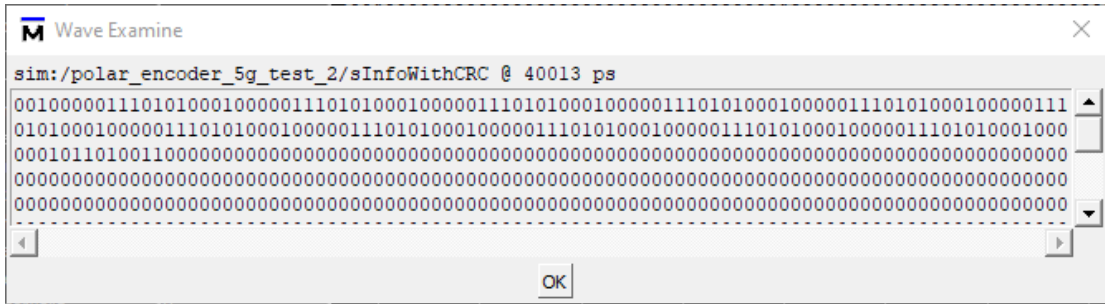
รูปที่ 4.171 สัญญาณ sInfoWithCRC บล็อกแรก (บางส่วน)

| | | | | | |
|---|-------------------|-----|----------------------------|--|--|
| /polar_encoder_5g_test_2/sCodeBlockSegmentationE... | 1 | | | | |
| + /polar_encoder_5g_test_2/sInfoWithCRC | 00100000111010... | | 00100000111010100010000011 | | |
| /polar_encoder_5g_test_2/sInfoWithCRCLength | 195 | 195 | | | |

รูปที่ 4.172 สัญญาณ sInfoWithCRC บล็อกท้าย (บางส่วน)



รูปที่ 4.173 บิต sInfoWithCRC บล็อกแรก (บางส่วน)



รูปที่ 4.174 บิต sInfoWithCRC บล็อกท้าย (บางส่วน)

4) ผลการทดสอบกระบวนการการลำดับของสัญญาณย่อย

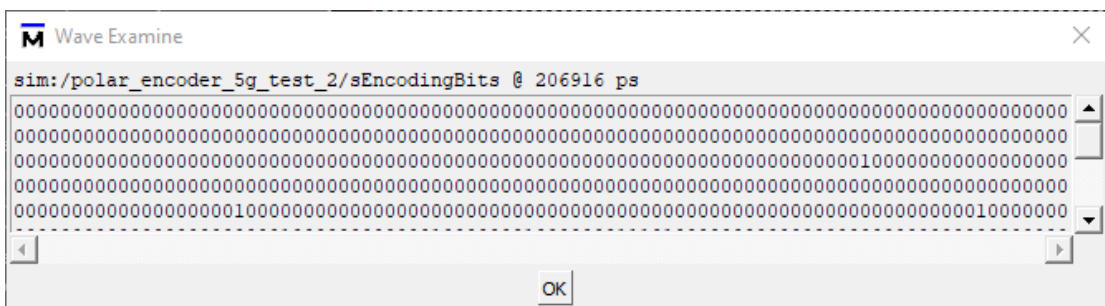
หลังจากกระบวนการเข้ารหัส CRC ให้ผลลัพธ์สัญญาณดังรูปที่ 4.171 และ 4.172 หรือแสดงเป็นเลขบิตดังรูปที่ 4.173 และ 4.174 กระบวนการลำดับของสัญญาณย่อยจะให้ผลลัพธ์เป็นสัญญาณ sEncodingBits ดังรูปที่ 4.175 และ 4.176 หรือแสดงเป็นเลขบิตดังรูปที่ 4.177 และ 4.178

| | | | | |
|--|-------------------|----------------------------------|--|--|
| /polar_encoder_5g_test_2/sMotherCodewordLength | 1024 | 1024 | | |
| + /polar_encoder_5g_test_2/sEncodingBits | 00000000000000... | 00000000000000000000000000000000 | | |
| + /polar_encoder_5g_test_2/sEncodingWOParityBits | 00000000000000... | 00000000000000000000000000000000 | | |

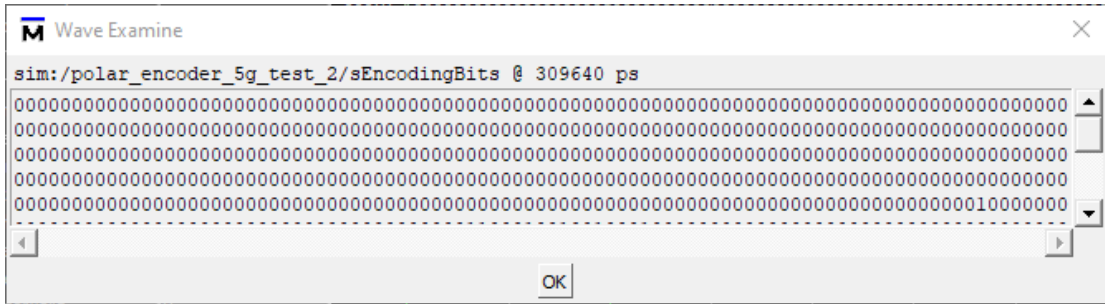
รูปที่ 4.175 สัญญาณ sEncodingBits บล็อกแรก (บางส่วน)

| | | | | |
|--|-------------------|---|--|--|
| /polar_encoder_5g_test_2/sMotherCodewordLength | 1024 | 1024 | | |
| + /polar_encoder_5g_test_2/sEncodingBits | 00000000000000... | 000... 00000000000000000000000000000000 | | |
| + /polar_encoder_5g_test_2/sEncodingWOParityBits | 00000000000000... | 000... 00000000000000000000000000000000 | | |

รูปที่ 4.176 สัญญาณ sEncodingBits บล็อกท้าย (บางส่วน)



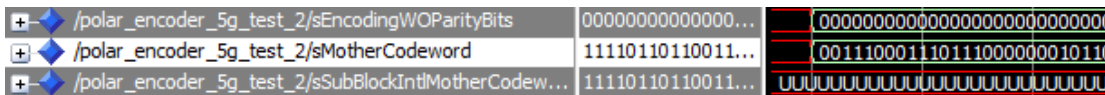
รูปที่ 4.177 บิต sEncodingBits บล็อกแรก (บางส่วน)



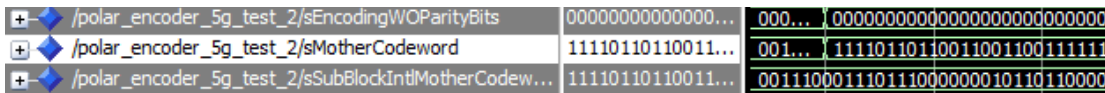
รูปที่ 4.178 บิต sEncodingBits บล็อกท้าย (บางส่วน)

5) ผลการทดสอบกระบวนการเข้ารหัสโพลาร์

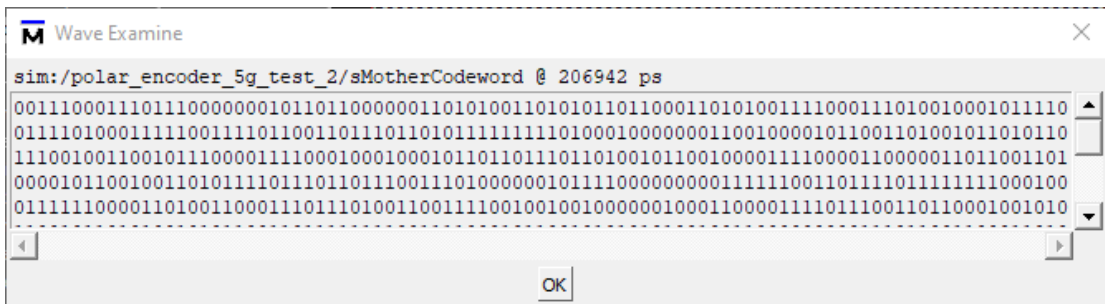
หลังจากกระบวนการลำดับช่องสัญญาณย่อยให้ผลลัพธ์สัญญาณดังรูปที่ 4.175 และ 4.176 หรือแสดงเป็นเลขบิตดังรูปที่ 4.177 และ 4.178 กระบวนการเข้ารหัสโพลาร์จะให้ผลลัพธ์เป็นสัญญาณ sMotherCodeword ดังรูปที่ 4.179 และ 4.180 หรือแสดงเป็นเลขบิตดังรูปที่ 4.181 และ 4.182



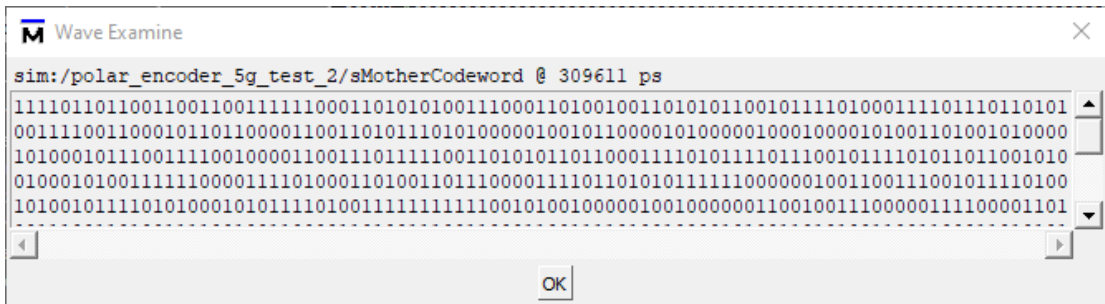
รูปที่ 4.179 สัญญาณ sMotherCodeword บล็อกแรก (บางส่วน)



รูปที่ 4.180 สัญญาณ sMotherCodeword บล็อกท้าย (บางส่วน)



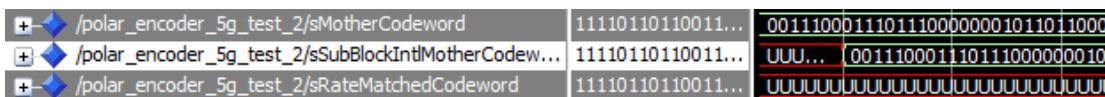
รูปที่ 4.181 บิต sMotherCodeword บล็อกแรก (บางส่วน)



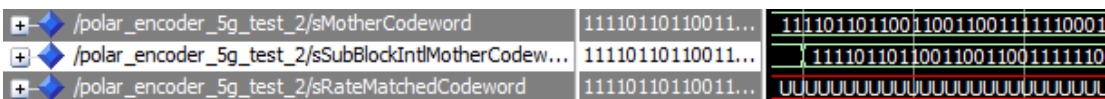
รูปที่ 4.182 บิต sMotherCodeword บล็อกท้าย (บางส่วน)

6) ผลการทดสอบกระบวนการแทรกสลับบล็อกย่อย

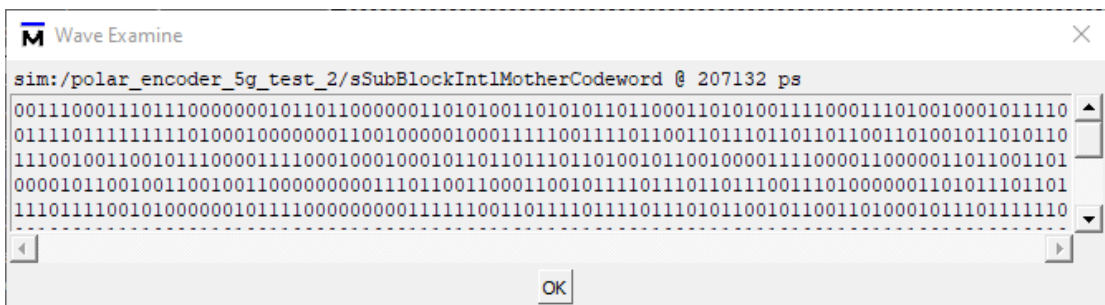
หลังจากกระบวนการเข้ารหัสโพลาร์ให้ผลลัพธ์สัญญาณรูปที่ 4.179 และ 4.180 หรือแสดงเป็นเลขบิตดังรูปที่ 4.181 และ 4.182 กระบวนการแทรกสลับบล็อกย่อยจะให้ผลลัพธ์เป็นสัญญาณ sSubBlockIntlMotherCodeword ดังรูปที่ 4.183 และ 4.184 หรือแสดงเป็นเลขบิตดังรูปที่ 4.185 และ 4.186



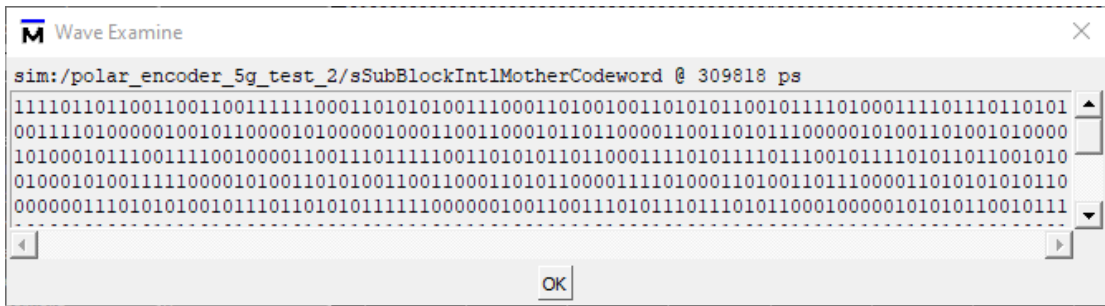
รูปที่ 4.183 สัญญาณ sSubBlockIntlMotherCodeword บล็อกแรก (บางส่วน)



รูปที่ 4.184 สัญญาณ sSubBlockIntlMotherCodeword บล็อกท้าย (บางส่วน)



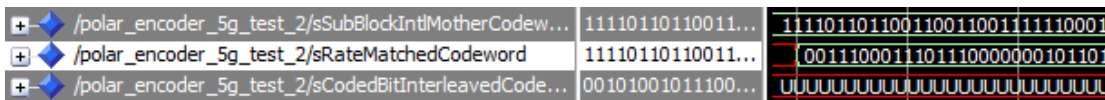
รูปที่ 4.185 บิต sSubBlockIntlMotherCodeword บล็อกแรก (บางส่วน)



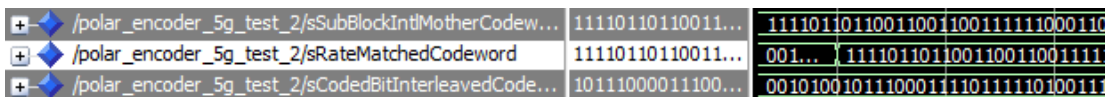
รูปที่ 4.186 บิต sSubBlockIntLMotherCodeword บล็อกท้าย (บางส่วน)

7) ผลการทดสอบกระบวนการปรับอัตราหัส

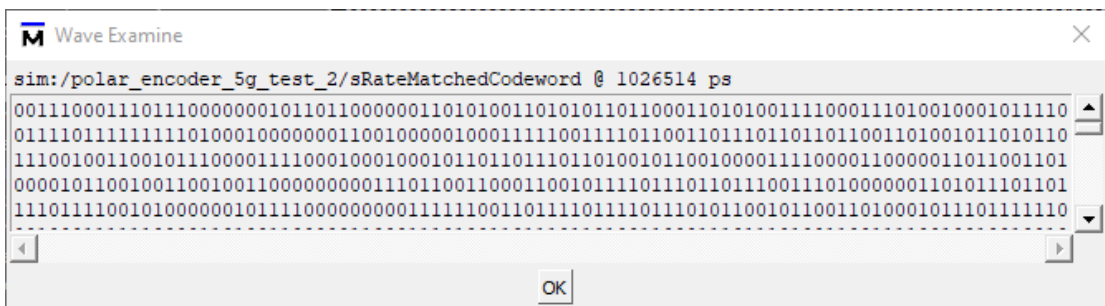
หลังจากกระบวนการแทรกสลับบล็อกย่อยให้ผลลัพธ์ดังรูปที่ 4.183 และ 4.184 หรือแสดงเป็นเลขบิตดังรูปที่ 4.185 และ 4.186 กระบวนการปรับอัตราหัสจะให้ผลลัพธ์เป็นสัญญาณ sRateMatchedCodeword ดังรูปที่ 4.187 และ 4.188 หรือแสดงเป็นเลขบิตดังรูปที่ 4.189 และ 4.190



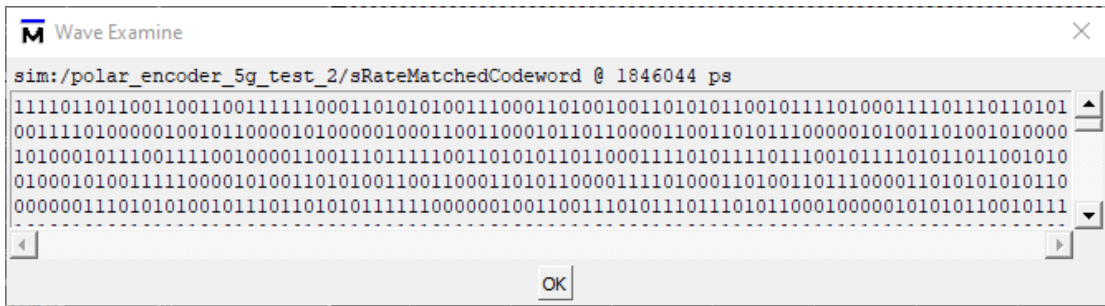
รูปที่ 4.187 สัญญาณ sRateMatchedCodeword บล็อกแรก (บางส่วน)



รูปที่ 4.188 สัญญาณ sRateMatchedCodeword บล็อกท้าย (บางส่วน)



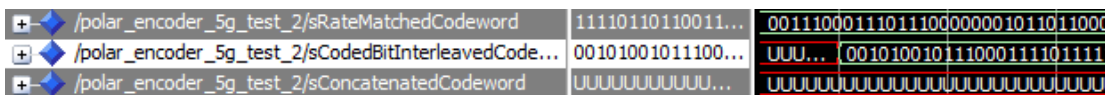
รูปที่ 4.189 บิต sRateMatchedCodeword บล็อกแรก (บางส่วน)



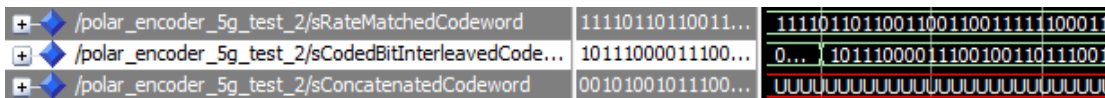
รูปที่ 4.190 บิต sRateMatchedCodeword บล็อกท้าย (บางส่วน)

8) ผลการทดสอบกระบวนการแทรกสลับบิตรหัส

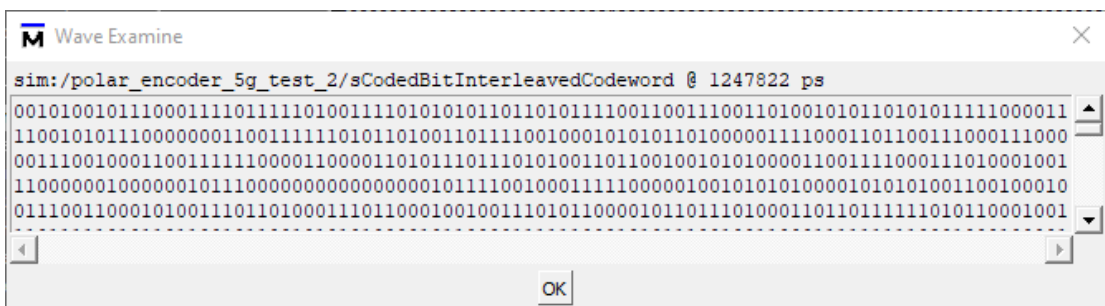
หลังจากกระบวนการปรับอัตรารหัสให้ผลลัพธ์ดังรูปที่ 4.187 และ 4.188 และหรือแสดงเป็นเลขบิตดังรูปที่ 4.189 และ 4.190 และกระบวนการแทรกสลับบิตรหัสจะให้ผลลัพธ์เป็นสัญญาณ sCodedBitInterleavedCodeword ดังรูปที่ 4.191 และ 4.192 หรือแสดงเป็นเลขบิตดังรูปที่ 4.193 และ 4.194



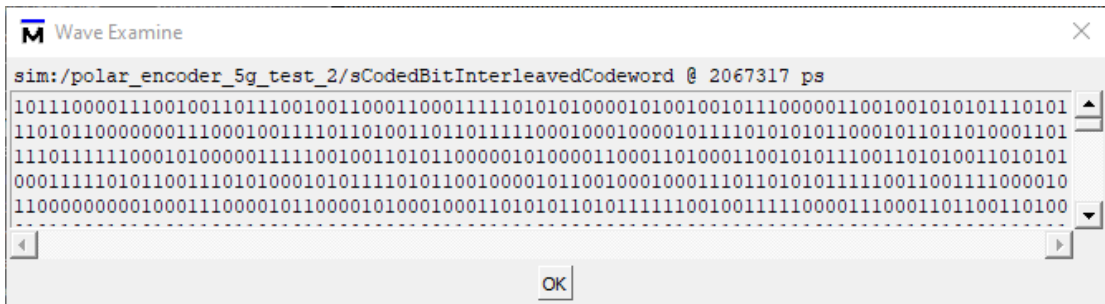
รูปที่ 4.191 สัญญาณ sCodedBitInterleavedCodeword บล็อกแรก (บางส่วน)



รูปที่ 4.192 สัญญาณ sCodedBitInterleavedCodeword บล็อกท้าย (บางส่วน)



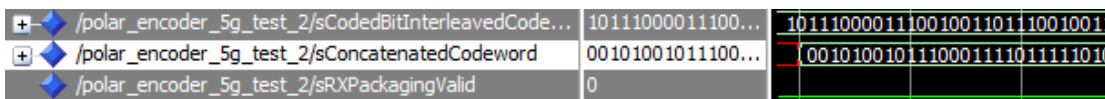
รูปที่ 4.193 บิต sCodedBitInterleavedCodeword บล็อกแรก (บางส่วน)



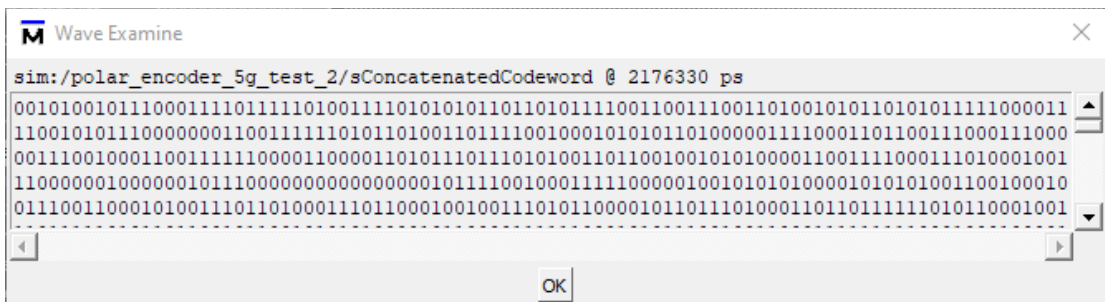
รูปที่ 4.194 บิต sCodedBitInterleavedCodeword บล็อกท้าย (บางส่วน)

9) ผลการทดสอบกระบวนการต่อบล็อกย่อย

หลังจากกระบวนการแทรกสลับบิตรหัสให้ผลลัพธ์ดังรูปที่ 4.191 และ 4.192 หรือแสดงเป็นเลขบิตดังรูปที่ 4.193 และ 4.194 กระบวนการต่อบล็อกย่อยจะให้ผลลัพธ์เป็นสัญญาณ sConcatenatedCodeword ดังรูปที่ 4.195 หรือแสดงเป็นเลขบิตดังรูปที่ 4.196



รูปที่ 4.195 สัญญาณ sConcatenatedCodeword (บางส่วน)

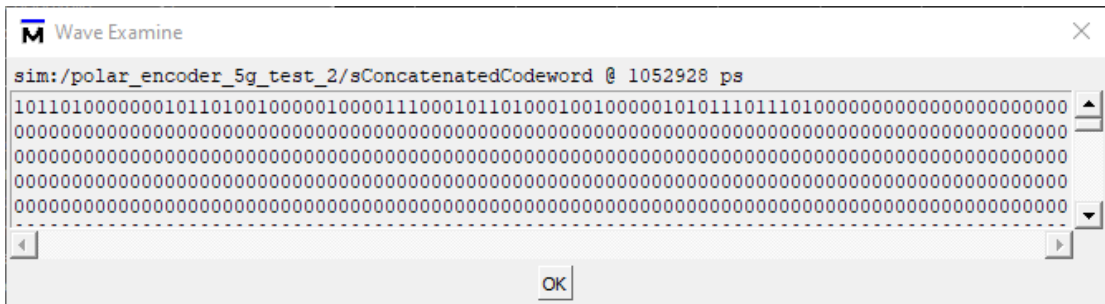


รูปที่ 4.196 บิต sConcatenatedCodeword (บางส่วน)

6.3.2.4 ผลการทดสอบวงจรเข้ารหัสกรณีช่องสัญญาณ PUCCH ที่ความยาวบิตข้อมูลสั้น

สำหรับผลการทดสอบของช่องสัญญาณ PUCCH ที่ความยาวบิตข้อมูลสั้น หัวข้อในการเก็บผลจะถูกแบ่งออกตามจำนวนกระบวนการเข้ารหัสของช่องสัญญาณ PUCCH ที่ความยาวบิตข้อมูลสั้น ซึ่งประกอบด้วย 7 กระบวนการ และรวมถึงผลลัพธ์ของกระบวนการโดยรวมดังนี้

- 1) กระบวนการทำงานโดยรวม
- 2) การเข้ารหัส CRC



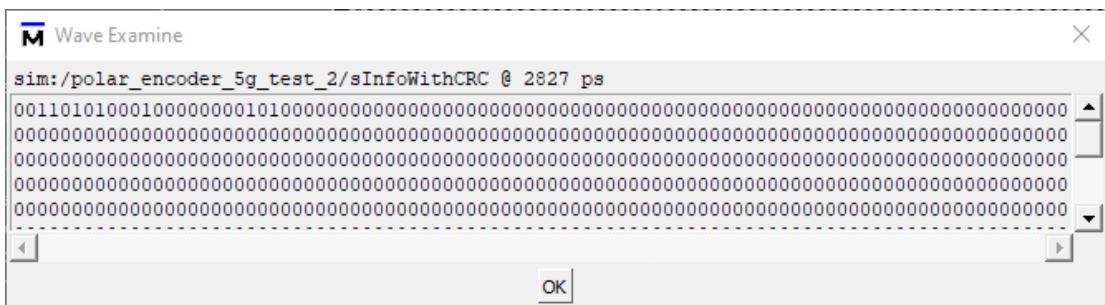
รูปที่ 4.200 บิตข้อมูลขาออก (บางส่วน)

2) ผลการทดสอบกระบวนการเข้ารหัส CRC

เมื่อวงจรเข้ารหัสรับสัญญาณบิตข้อมูลขาเข้าดังรูปที่ 4.197 หรือแสดงเป็นเลขบิตดังรูปที่ 4.198 กระบวนการเข้ารหัส CRC จะให้ผลลัพธ์เป็นสัญญาณ sInfoWithCRC ดังรูปที่ 4.201 หรือแสดงเป็นเลขบิตดังรูปที่ 4.202

| | | | | |
|--|-------------------|----------------------|----------|--|
| /polar_encoder_5g_test_2/sCodeBlockSegmentationEn... | 0 | | | |
| /polar_encoder_5g_test_2/sInfoWithCRC | 00110101000100... | 00110101000100000000 | 10100000 | |
| /polar_encoder_5g_test_2/sInfoWithCRCLength | 22 | 22 | | |

รูปที่ 4.201 สัญญาณ sInfoWithCRC (บางส่วน)



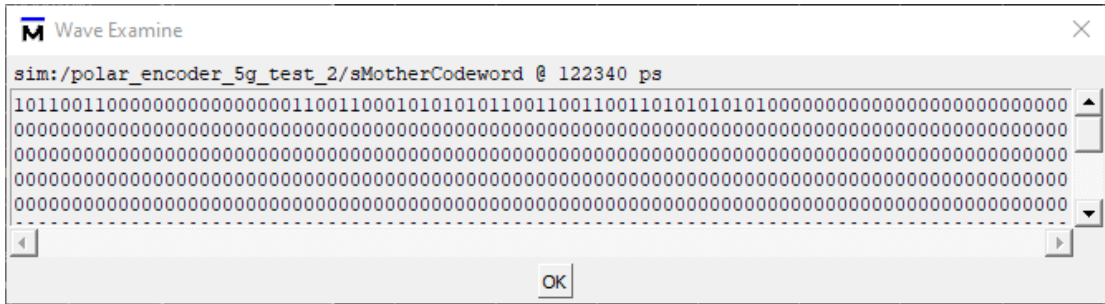
รูปที่ 4.202 บิต sInfoWithCRC (บางส่วน)

3) ผลการทดสอบกระบวนการลำดับช่องสัญญาณย่อย

หลังจากกระบวนการเข้ารหัส CRC ให้ผลลัพธ์สัญญาณดังรูปที่ 4.201 หรือแสดงเป็นเลขบิตดังรูปที่ 4.202 กระบวนการลำดับช่องสัญญาณย่อยจะให้ผลลัพธ์เป็นสัญญาณ sEncodingWOParityBits ดังรูปที่ 4.203 หรือแสดงเป็นเลขบิตดังรูปที่ 4.204

| | | | |
|--|-------------------|------|-------------------------|
| /polar_encoder_5g_test_2/sEncodingBits | 00000000000000... | U... | 0000000000000000000000 |
| /polar_encoder_5g_test_2/sEncodingWOParityBits | 00000000000000... | U... | 0000000000000000000000 |
| /polar_encoder_5g_test_2/sMotherCodeword | 10110011000000... | U... | 10110011000000000000001 |

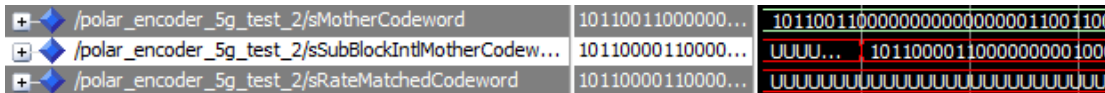
รูปที่ 4.203 สัญญาณ sEncodingWOParityBits (บางส่วน)



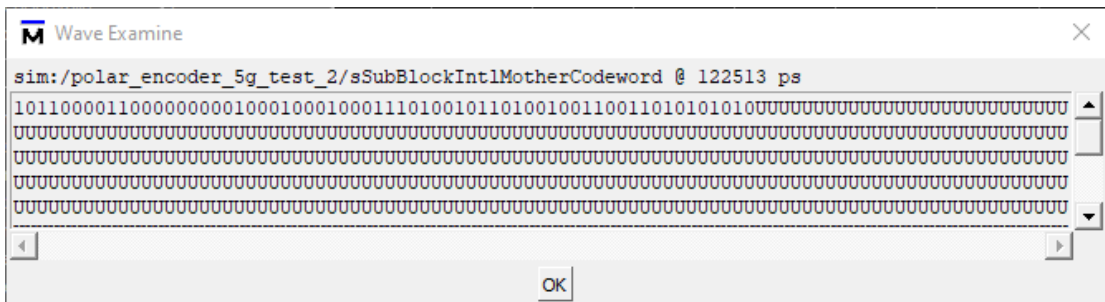
รูปที่ 4.208 บิต sMotherCodeword (บางส่วน)

6) ผลการทดสอบกระบวนการแทรกสลับบล็อกย่อย

หลังจากกระบวนการเข้ารหัสโพลาร์ให้ผลลัพธ์สัญญาณดังรูปที่ 4.207 หรือแสดงเป็นเลขบิตดังรูปที่ 4.208 กระบวนการแทรกสลับบล็อกย่อยจะให้ผลลัพธ์เป็นสัญญาณ sSubBlockIntlCodeword ดังรูปที่ 4.209 หรือแสดงเป็นเลขบิตดังรูปที่ 4.210



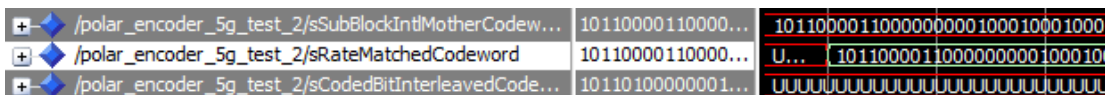
รูปที่ 4.209 สัญญาณ sSubBlockIntlCodeword (บางส่วน)



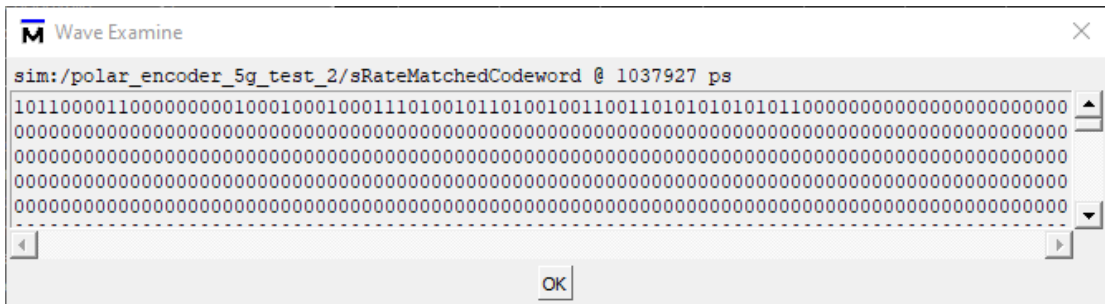
รูปที่ 4.210 บิต sSubBlockIntlCodeword (บางส่วน)

7) ผลการทดสอบกระบวนการปรับอัตราหัส

หลังจากกระบวนการแทรกสลับบล็อกย่อยให้ผลลัพธ์สัญญาณดังรูปที่ 4.209 หรือแสดงเป็นเลขบิตดังรูปที่ 4.210 กระบวนการปรับอัตราหัสจะให้ผลลัพธ์เป็นสัญญาณ sRateMatchedCodeword ดังรูปที่ 4.211 หรือแสดงเป็นเลขบิตดังรูปที่ 4.212



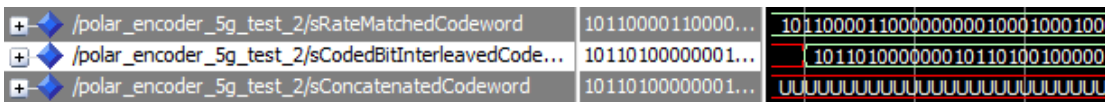
รูปที่ 4.211 สัญญาณ sRateMatchedCodeword (บางส่วน)



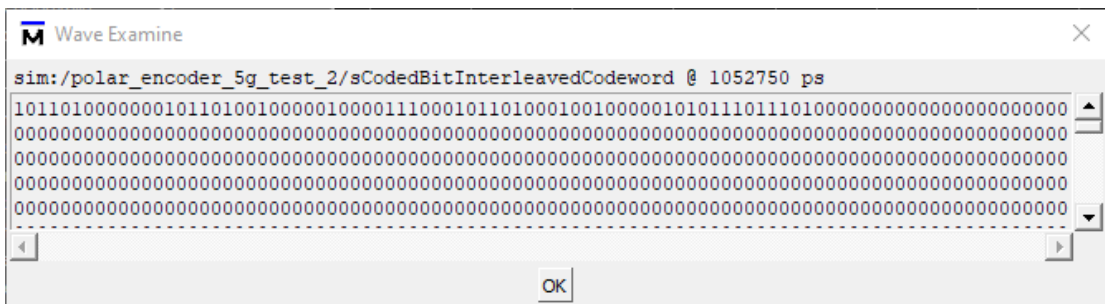
รูปที่ 4.212 บิต sRateMatchedCodeword (บางส่วน)

8) ผลการทดสอบกระบวนการแทรกสลับบิตรหัส

หลังจากกระบวนการแทรกสลับบิตเรียบร้อยแล้วให้ผลลัพธ์สัญญาณดังรูปที่ 4.211 หรือแสดงเป็นเลขบิตดังรูปที่ 4.212 กระบวนการปรับอัตรารหัสจะให้ผลลัพธ์เป็นสัญญาณ sCodedBitInterleaveCodeword ดังรูปที่ 4.213 หรือแสดงเป็นเลขบิตดังรูปที่ 4.214



รูปที่ 4.213 สัญญาณ sCodedBitInterleaveCodeword (บางส่วน)



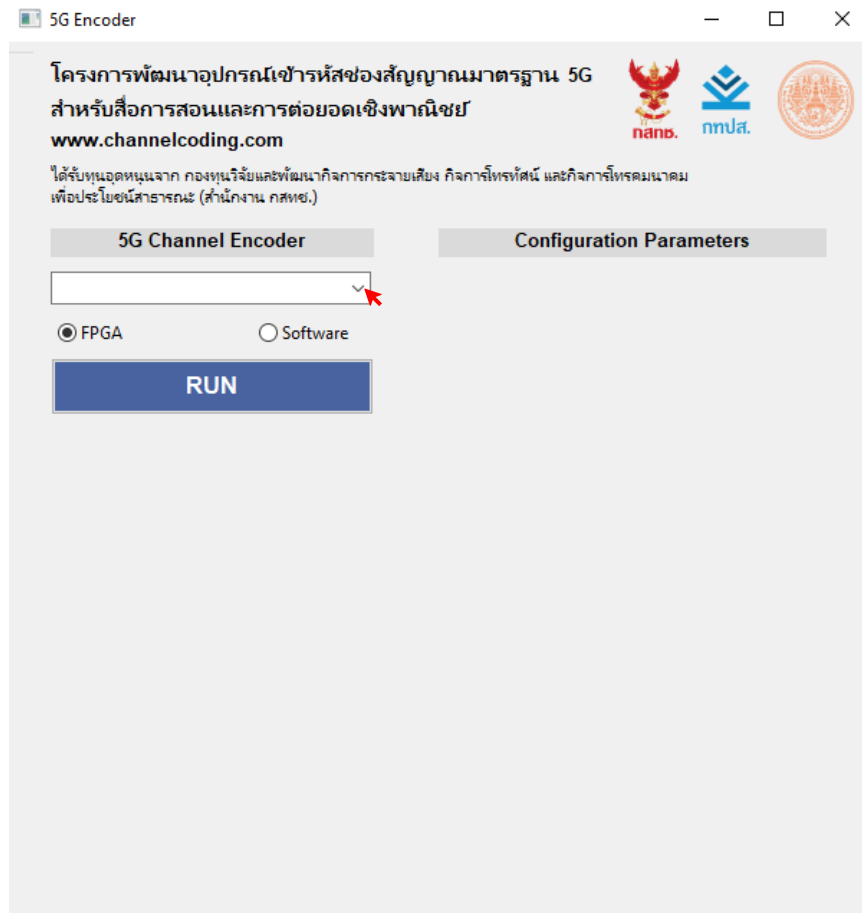
รูปที่ 4.214 บิต sCodedBitInterleaveCodeword (บางส่วน)

4.3 ผลการทดสอบชุดซอฟต์แวร์การเรียนรู้การสื่อสารดิจิทัลที่มีการเข้ารหัสช่องสัญญาณ

มาตรฐาน 5G

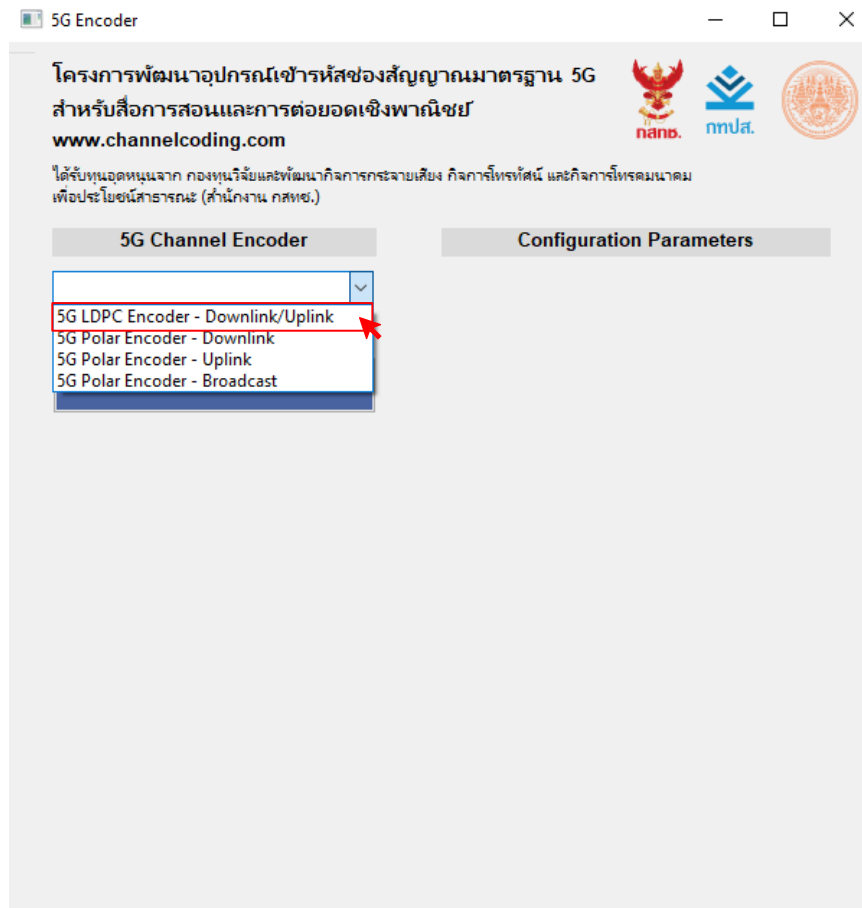
4.3.1 ส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อควบคุมวงจรเข้ารหัสช่องสัญญาณชนิด LDPC Codes

จากการออกแบบและสร้างส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อควบคุมวงจรเข้ารหัสช่องสัญญาณชนิด LDPC Codes โดยการใช้การเขียนโปรแกรมภาษาไพธอน ผลการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้เมื่อเริ่มต้นใช้งานแสดงดังรูปที่ 4.215



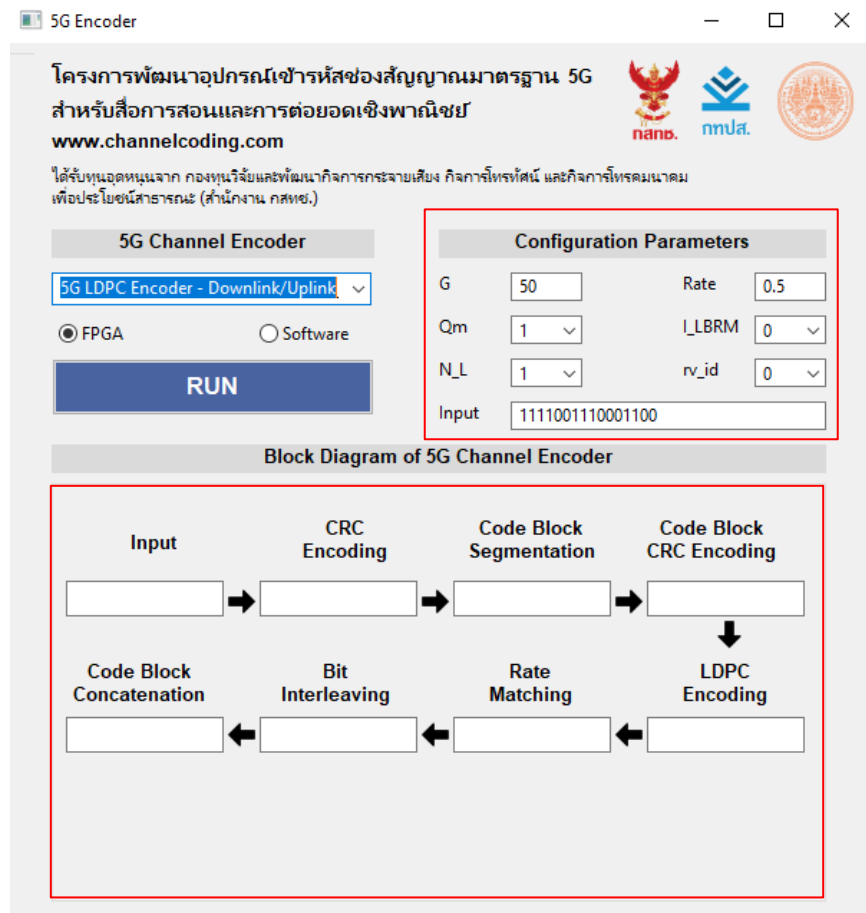
รูปที่ 4.215 หน้าแรกของส่วนต่อประสานกราฟิกกับผู้ใช้

จากรูปที่ 4.215 ผู้ใช้สามารถเลือกชนิดของตัวเข้ารหัสได้โดยกดตรงลูกศรเพื่อให้เห็นรายการของตัวเข้ารหัสทั้งหมดในมาตรฐาน 5G ซึ่งในกรณีที่ต้องการเข้ารหัสช่องสัญญาณชนิด LDPC Codes กดเลือกที่ “5G LDPC Encoder – Downlink/Uplink” ดังแสดงในรูปที่ 4.216



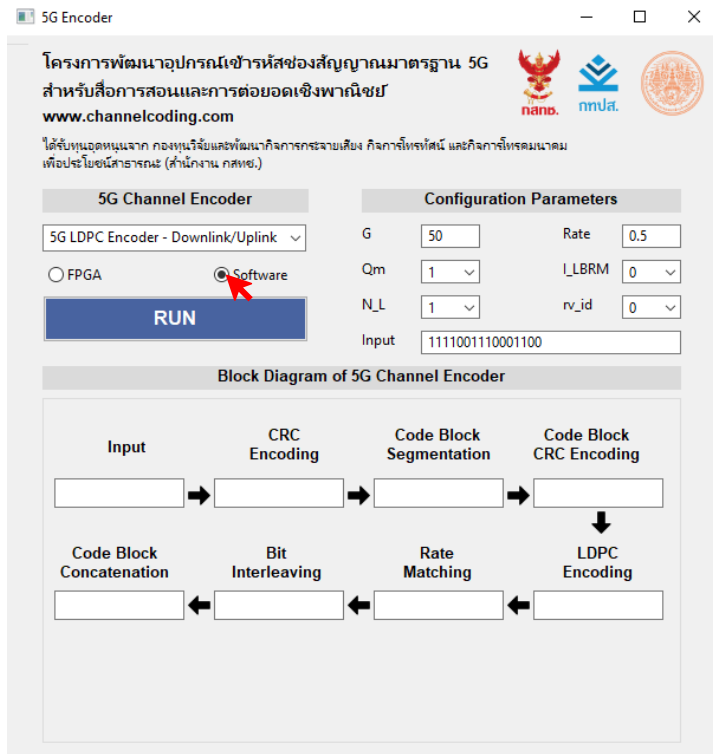
รูปที่ 4.216 การเลือกชนิดของตัวเข้ารหัสกรณีรหัสช่องสัญญาณชนิด LDPC Codes

หลังจากที่ผู้ใช้เลือกชนิดของตัวเข้ารหัส หน้าต่างของส่วนต่อประสานกราฟิกกับผู้ใช้จะแสดงรายการพารามิเตอร์ที่ผู้ใช้จำเป็นต้องป้อนและอินพุตทางด้านขวา ซึ่งโปรแกรมจะกำหนดค่าเริ่มต้นของทุกพารามิเตอร์และกำหนดอินพุตแบบสุ่มให้อัตโนมัติ โดยผู้ใช้สามารถปรับเปลี่ยนค่าดังกล่าวได้ตามต้องการ นอกจากนี้ยังแสดงบล็อกไดอะแกรมการเข้ารหัสตามมาตรฐาน 5G บริเวณด้านล่าง ผลการทดสอบสำหรับกรณีการเข้ารหัสของสัญญาณโดยใช้ “5G – LDPC Encoder – Downlink/Uplink” แสดงดังรูปที่ 4.217

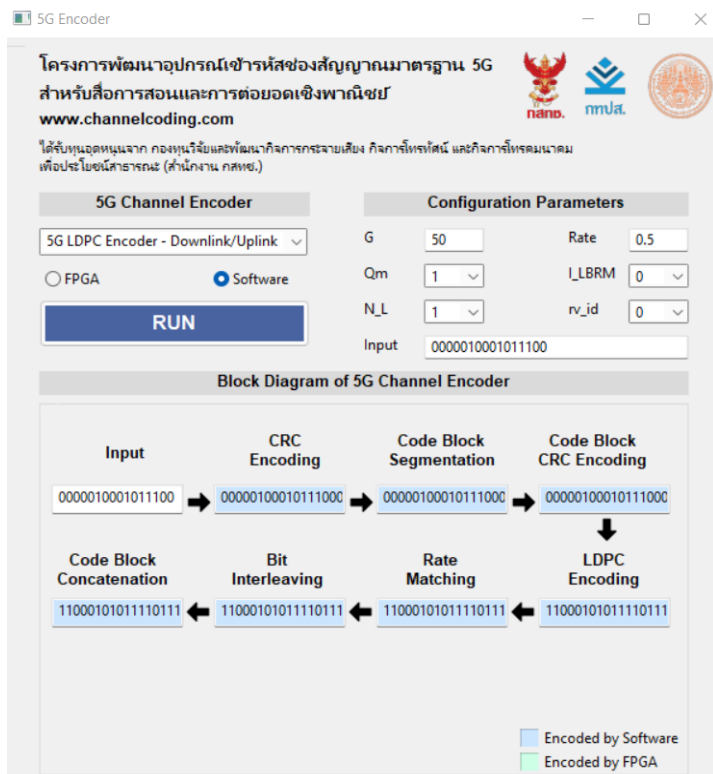


รูปที่ 4.217 หน้าต่างแสดงรายการพารามิเตอร์กรณีเข้ารหัสช่องสัญญาณ โดยใช้ 5G LDPC Encoder – Downlink/Uplink

หลังจากที่ผู้ใช้ป้อนพารามิเตอร์และอินพุตเสร็จสิ้น ในกรณีที่ต้องการเข้ารหัสโดยใช้ชุดซอฟต์แวร์ ผู้ใช้จะต้องกดเลือก “Software” ก่อนกดปุ่ม “RUN” ดังแสดงในรูปที่ 4.218 และหลังจากกดปุ่ม “RUN” โปรแกรมจะทำการเข้ารหัสโดยใช้ชุดซอฟต์แวร์การเข้ารหัสตามมาตรฐาน 5G และแสดงผลลัพธ์ของแต่ละขั้นตอนในบล็อกไดอะแกรม ดังแสดงในรูปที่ 4.219



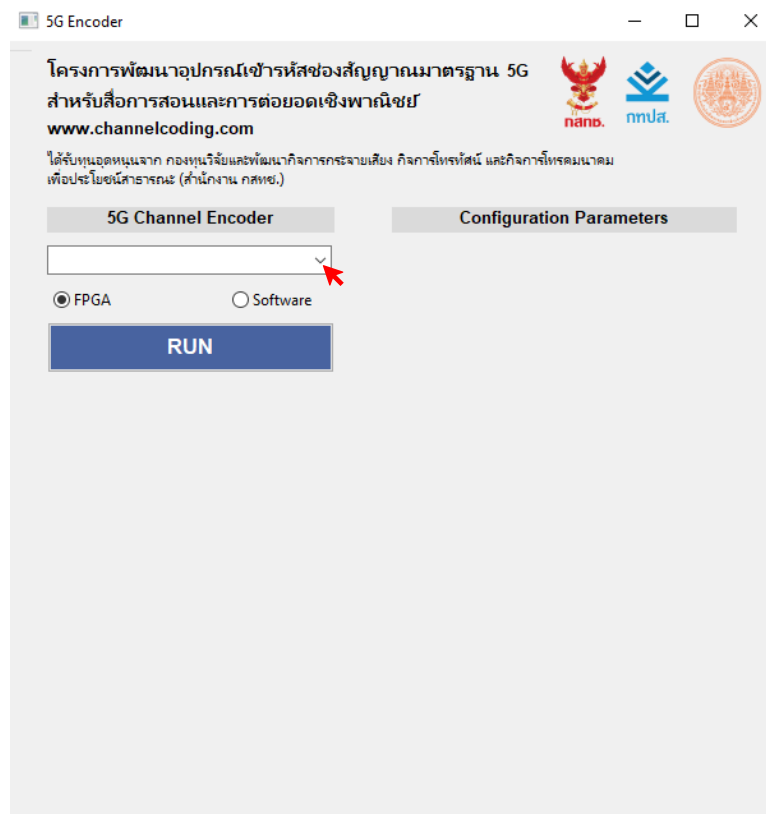
รูปที่ 4.218 การเลือกวิธีการเข้ารหัส



รูปที่ 4.219 ผลลัพธ์การเข้ารหัสช่องสัญญาณโดยใช้ 5G – LDPC Encoder Downlink/Uplink

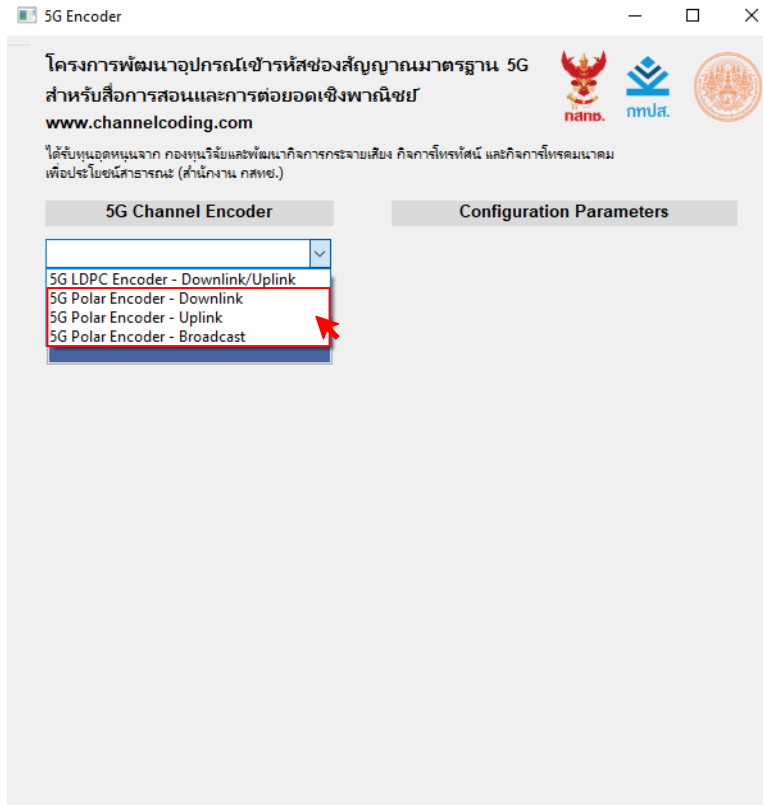
4.3.2 ส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อควบคุมวงจรเข้ารหัสช่องสัญญาณชนิด Polar Codes

จากการออกแบบและสร้างส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อควบคุมวงจรเข้ารหัสช่องสัญญาณชนิด Polar Codes โดยใช้การเขียนโปรแกรมภาษาไพธอน ผลการทดสอบส่วนต่อประสานกราฟิกกับผู้ใช้เมื่อเริ่มต้นใช้งานแสดงดังรูปที่ 4.220



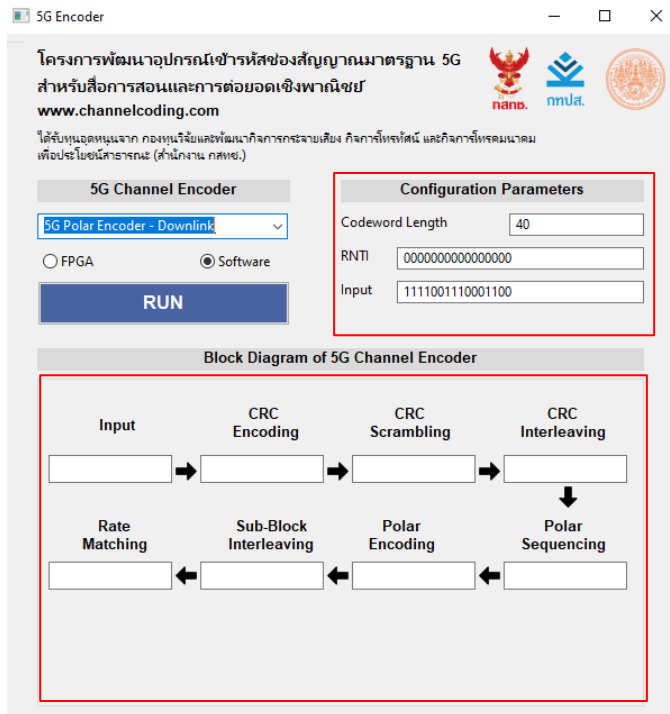
รูปที่ 4.220 หน้าแรกของส่วนต่อประสานกราฟิกกับผู้ใช้

จากรูปที่ 4.220 ผู้ใช้สามารถเลือกชนิดของตัวเข้ารหัสได้โดยกดตรงลูกศรเพื่อให้เห็นรายการของตัวเข้ารหัสทั้งหมดในมาตรฐาน 5G ซึ่งในกรณีที่ต้องการเข้ารหัสช่องสัญญาณชนิด Polar Codes จะแบ่งออกเป็น 3 รูปแบบ ได้แก่ “5G Polar Encoder – Downlink” “5G Polar Encoder - Uplink” และ “5G Polar Encoder - Broadcast” ดังแสดงในรูปที่ 4.221

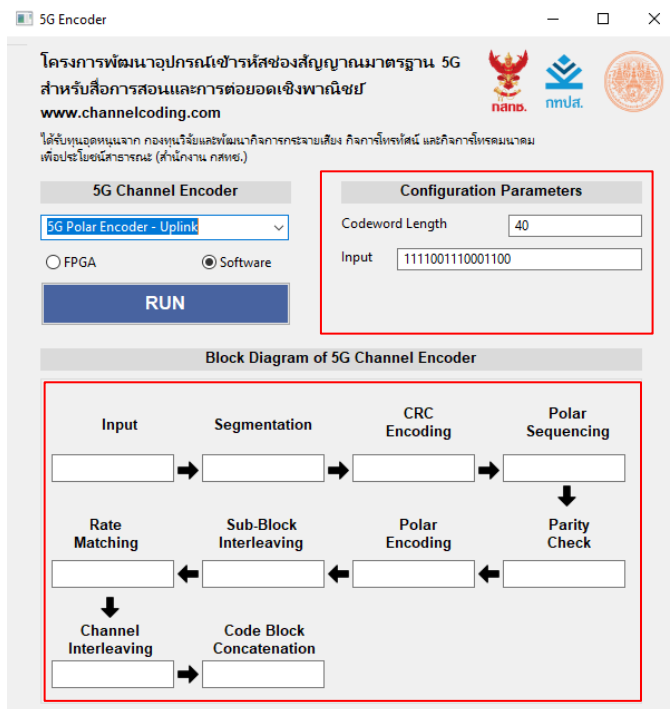


รูปที่ 4.221 การเลือกชนิดของตัวเข้ารหัสกรณีรหัสช่องสัญญาณชนิด Polar Codes

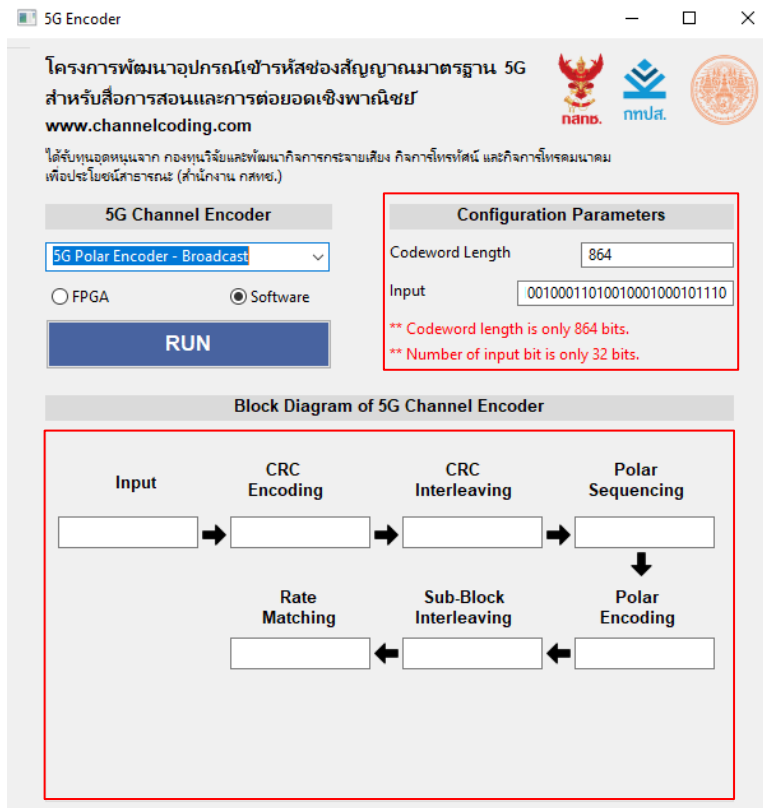
หลังจากที่ผู้ใช้เลือกชนิดของตัวเข้ารหัส หน้าต่างของส่วนต่อประสานกราฟิกกับผู้ใช้จะแสดงรายการพารามิเตอร์ที่ผู้ใช้จำเป็นต้องป้อนและอินพุตทางด้านขวา ซึ่งโปรแกรมจะกำหนดค่าเริ่มต้นของทุกพารามิเตอร์และกำหนดอินพุตแบบสุ่มให้อัตโนมัติ โดยผู้ใช้สามารถปรับแก้ค่าดังกล่าวได้ตามต้องการ นอกจากนี้ยังแสดงบล็อกไดอะแกรมการเข้ารหัสตามมาตรฐาน 5G บริเวณด้านล่าง ผลการทดสอบสำหรับกรณีการเข้ารหัสช่องสัญญาณโดยใช้ “5G Polar Encoder – Downlink” “5G Polar Encoder - Uplink” และ “5G Polar Encoder - Broadcast” แสดงดังรูปที่ 4.222 – 4.224 ตามลำดับ



รูปที่ 4.222 หน้าต่างแสดงรายการพารามิเตอร์กรณีการเข้ารหัสช่องสัญญาณ โดยใช้ 5G Polar Encoder – Downlink

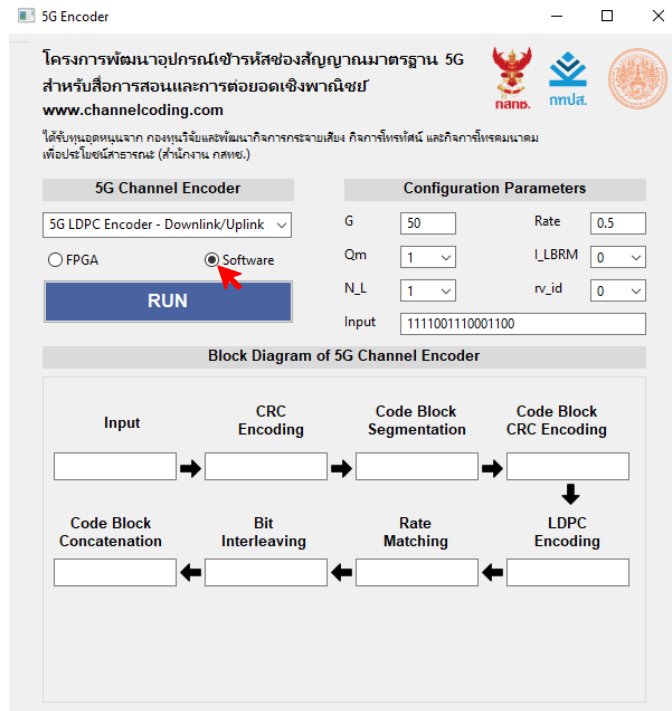


รูปที่ 4.223 หน้าต่างแสดงรายการพารามิเตอร์กรณีการเข้ารหัสช่องสัญญาณ โดยใช้ 5G Polar Encoder – Uplink

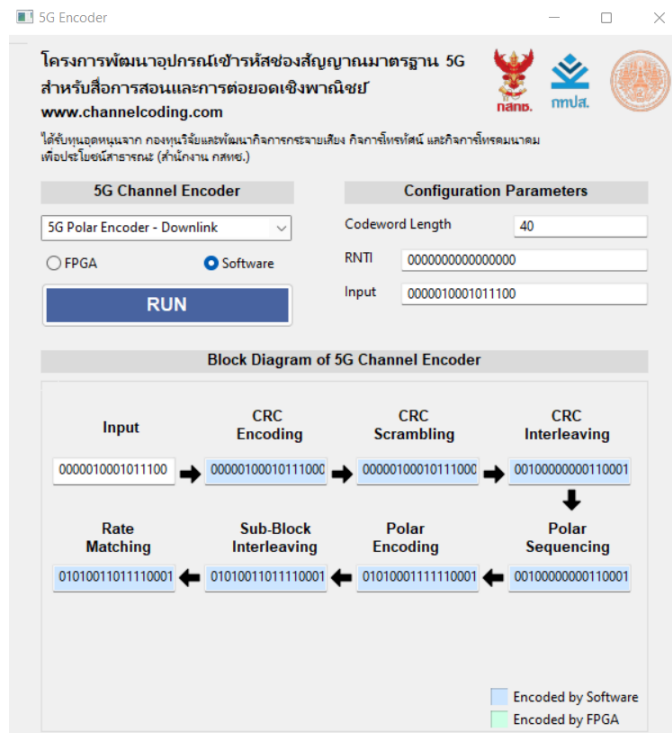


รูปที่ 4.224 หน้าต่างแสดงรายการพารามิเตอร์กรณีการเข้ารหัสของสัญญาณ โดยใช้ 5G Polar Encoder – Broadcast

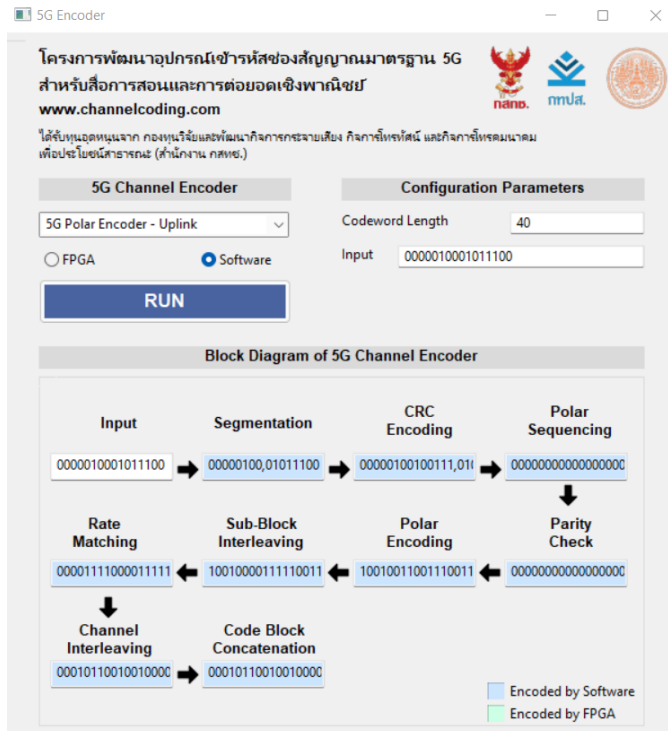
หลังจากที่ผู้ใช้ป้อนพารามิเตอร์และอินพุตเสร็จสิ้น ในกรณีที่ต้องการเข้ารหัสโดยใช้ชุดซอฟต์แวร์ ผู้ใช้จะต้องกดเลือก “Software” ก่อนกดปุ่ม “RUN” ดังแสดงในรูปที่ 4.225 และหลังจากกดปุ่ม “RUN” โปรแกรมจะทำการเข้ารหัสโดยใช้ชุดซอฟต์แวร์การเข้ารหัสตามมาตรฐาน 5G และแสดงผลลัพธ์ของแต่ละขั้นตอนในบล็อกไดอะแกรม ดังแสดงในรูปที่ 4.226 – 4.228



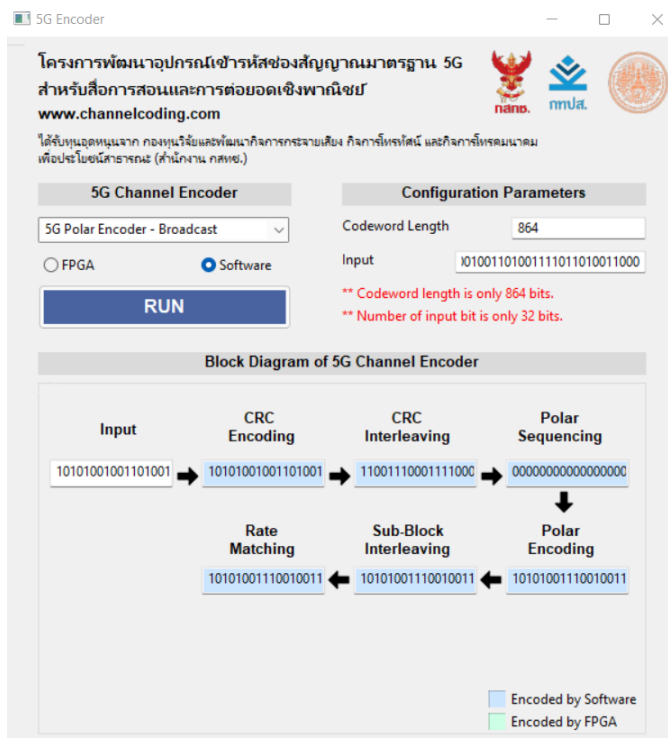
รูปที่ 4.225 การเลือกวิธีการเข้ารหัส



รูปที่ 4.226 ผลลัพธ์การเข้ารหัสช่องสัญญาณโดยใช้ 5G Polar Encoder – Downlink



รูปที่ 4.227 ผลลัพธ์การเข้ารหัสช่องสัญญาณโดยใช้ 5G Polar Encoder – Uplink

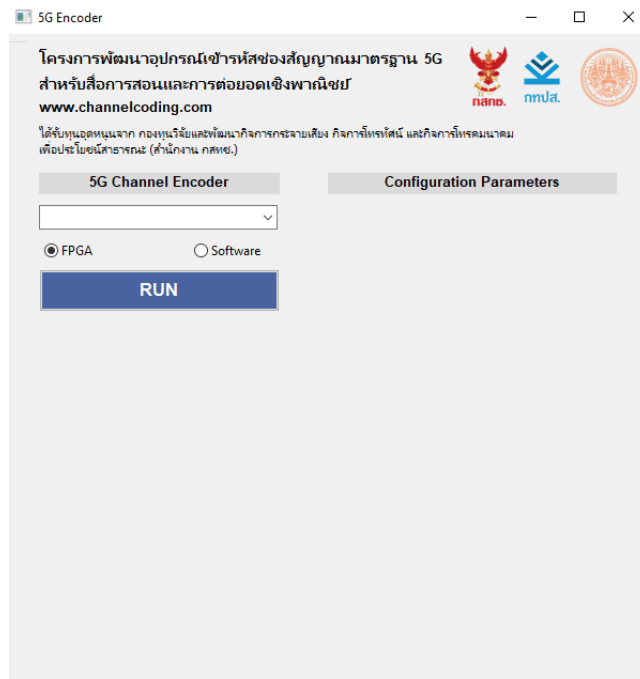


รูปที่ 4.228 ผลลัพธ์การเข้ารหัสช่องสัญญาณโดยใช้ 5G Polar Encoder – Broadcast

4.4 ผลการทดสอบชุดวงจรการเรียนรู้การสื่อสารดิจิทัลที่มีการเข้ารหัสช่องสัญญาณ มาตรฐาน 5G

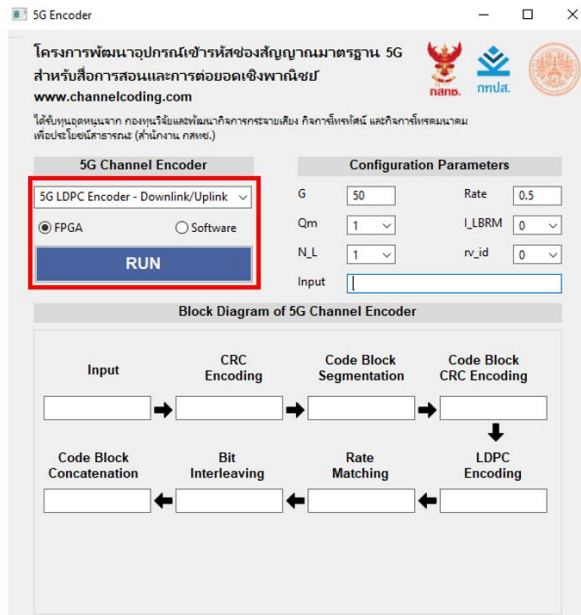
4.4.1 ชุดวงจรเข้ารหัสร่วมกับส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อการเข้ารหัสช่องสัญญาณ ชนิด LDPC Codes

จากการออกแบบและสร้างชุดวงจรเข้ารหัสร่วมกับส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อเชื่อมต่อข้อมูลระหว่างผู้ใช้งานผ่านส่วนต่อประสานกราฟิกกับผู้ใช้กับวงจรเข้ารหัสช่องสัญญาณชนิดรหัส LDPC ผลการทดสอบการแสดงผลลัพธ์ของวงจรเข้ารหัสร่วมกับส่วนต่อประสานกราฟิกกับผู้ใช้ ผู้ใช้งานจะเริ่มเลือกวิธีการเข้ารหัส เลือกตัวเข้ารหัส และกรอกพารามิเตอร์และอินพุต ลงบนส่วนต่อประสานกราฟิกกับผู้ใช้ดังรูปที่ 4.229



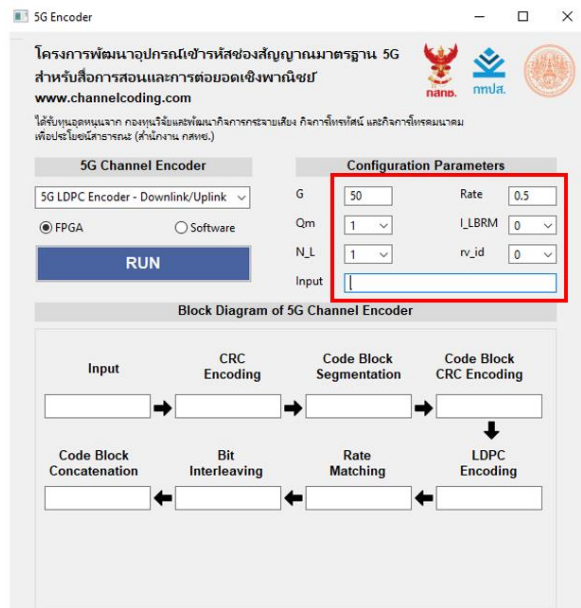
รูปที่ 4.229 หน้าแรกของส่วนต่อประสานกราฟิกกับผู้ใช้

จากรูปที่ 4.229 ผู้ใช้งานสามารถเลือกวิธีการเข้ารหัสชนิดต่าง ๆ ได้จากเมนู drop down และเลือกตัวเข้ารหัสได้จากปุ่ม radio หัวข้อนี้เลือกวิธีการเข้ารหัส “5G LDPC Encoder - Downlink/Uplink” และเลือกตัวเข้ารหัส “FPGA” หลังจากเลือกวิธีการเข้ารหัสที่ต้องการ ส่วนต่อประสานกราฟิกกับผู้ใช้จะแสดงส่วนการรับค่าพารามิเตอร์และอินพุต และส่วนการแสดงผลที่สอดคล้องกับรหัสช่องสัญญาณที่ถูกเลือก ดังรูปที่ 4.230



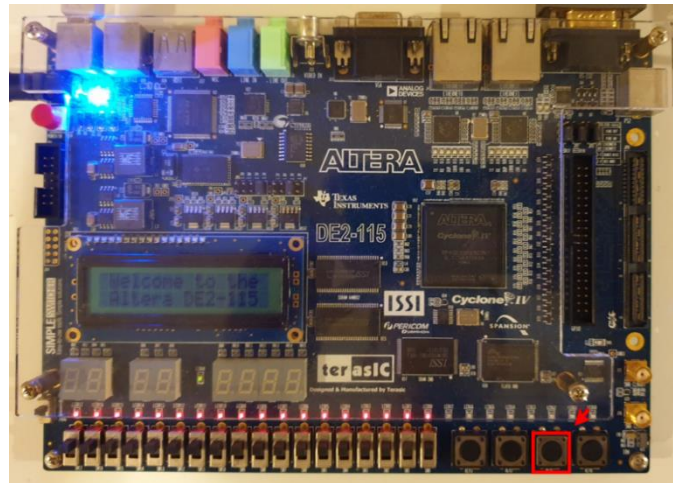
รูปที่ 4.230 ผู้ใช้งานเลือกวิธีการเข้ารหัส “5G LDPC Encoder - Downlink/Uplink” และเลือกตัวเข้ารหัส “FPGA”

จากรูปที่ 4.230 ผู้ใช้งานสามารถกรอกค่าพารามิเตอร์และอินพุตที่ต้องการเข้ารหัสในส่วนการรับค่าพารามิเตอร์และอินพุต หัวข้อนี้จะกรอกค่าพารามิเตอร์และอินพุตดังรูปที่ 4.231

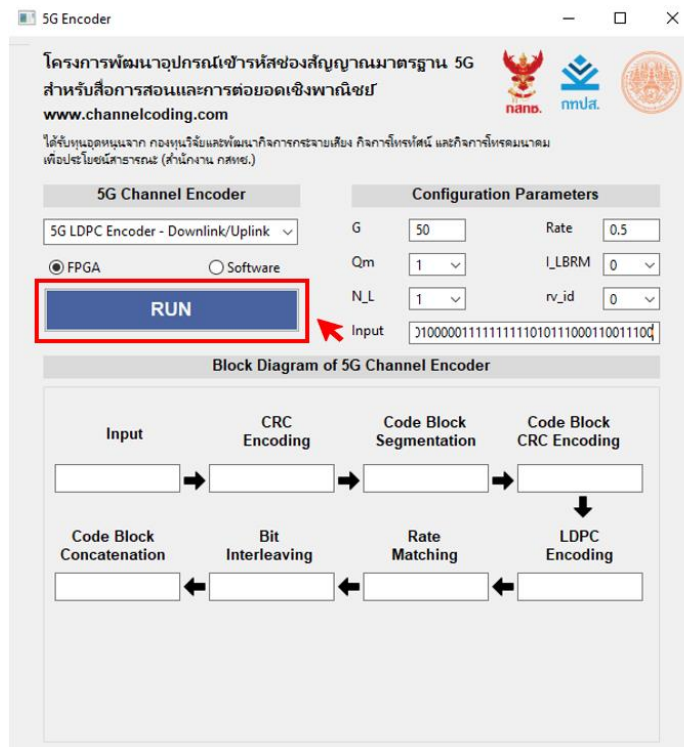


รูปที่ 4.231 ผู้ใช้งานกรอกค่าพารามิเตอร์และอินพุตในส่วนการรับค่าพารามิเตอร์และอินพุตสำหรับรหัส LDPC

จากรูปที่ 4.231 ก่อนการกดปุ่ม “RUN” เพื่อส่งข้อมูลจากคอมพิวเตอร์ไปยังวงจรเข้ารหัส FPGA วงจรเข้ารหัสต้องทำการล้างข้อมูลที่ค้างในวงจรเข้ารหัสเสมอ โดยการกดปุ่ม “Key1” แสดงดังรูปที่ 4.232 จากนั้นผู้ใช้งานสามารถส่งข้อมูลจากคอมพิวเตอร์ไปยังวงจรเข้ารหัส FPGA ได้โดยการกดปุ่ม “RUN” ส่วนต่อประสานกราฟิกกับผู้ใช้ ดังรูปที่ 4.233

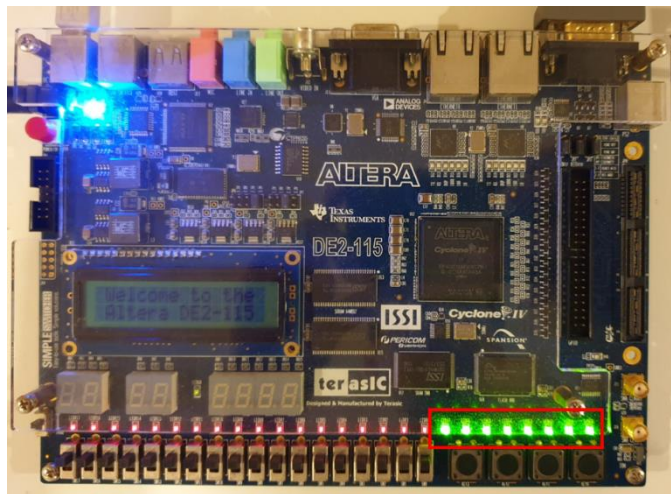


รูปที่ 4.232 การกดปุ่ม “Key1” เพื่อล้างค่าที่ค้างในวงจรเข้ารหัส FPGA



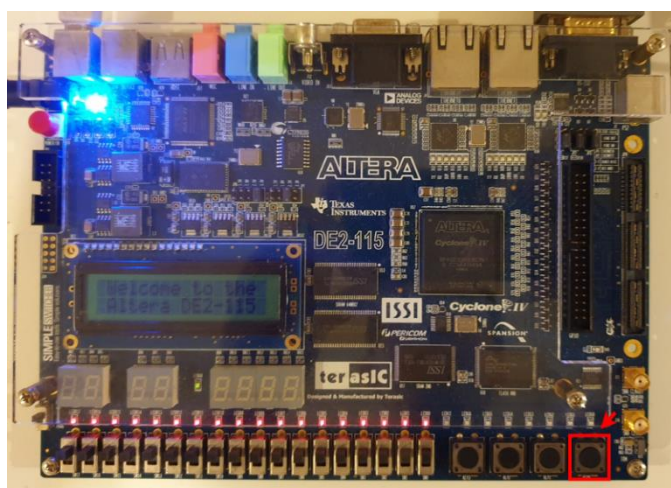
รูปที่ 4.233 การกดปุ่ม “RUN” เพื่อส่งข้อมูลจากคอมพิวเตอร์ไปยังวงจรเข้ารหัส FPGA

จากรูปที่ 4.233 หลังการกดปุ่ม “RUN” ข้อมูลจะถูกส่งไปยังวงจรเข้ารหัส FPGA และทำการเข้ารหัส LDPC เมื่อทำการเข้ารหัสเสร็จสิ้น วงจรจะแสดงสีบนหลอด LED จำนวน 8 ดวงทางขวา ดังรูปที่ 4.234



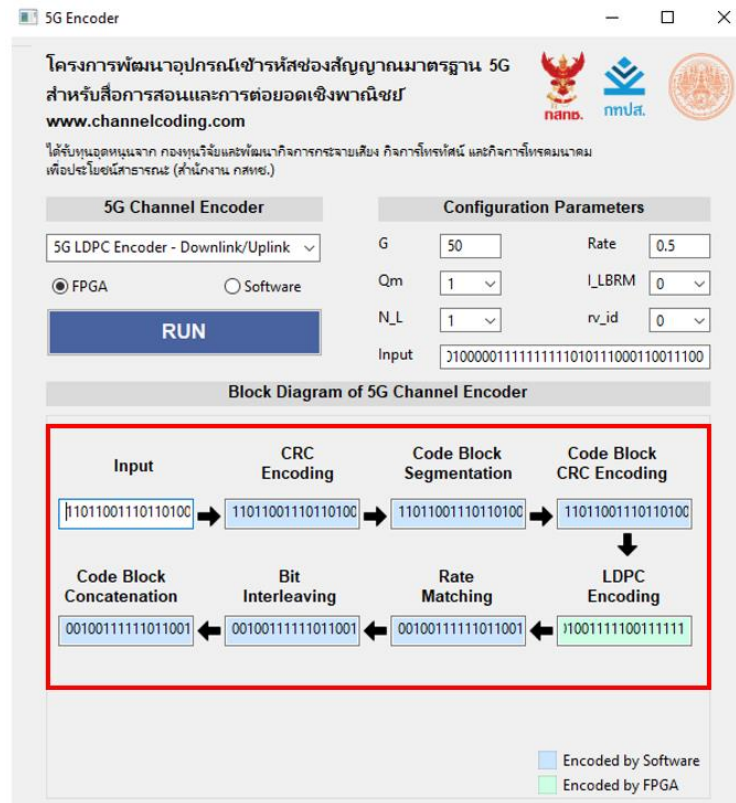
รูปที่ 4.234 หลอด LED 8 ดวงทางขวาติด เพื่อบ่งบอกถึงการเข้ารหัสในวงจรเข้ารหัส FPGA เสร็จสิ้น

จากรูปที่ 4.234 จะถือว่าวงจรเข้ารหัส FPGA ทำการเข้ารหัสเสร็จสิ้น ผู้ใช้งานจะสามารถกดปุ่ม “Key0” เพื่อส่งผลลัพธ์การเข้ารหัส LDPC จากวงจรเข้ารหัส FPGA ไปยังคอมพิวเตอร์ได้ ดังรูปที่ 4.235



รูปที่ 4.235 การกดปุ่ม “Key0” เพื่อส่งผลลัพธ์จากวงจรเข้ารหัส FPGA ไปยังคอมพิวเตอร์

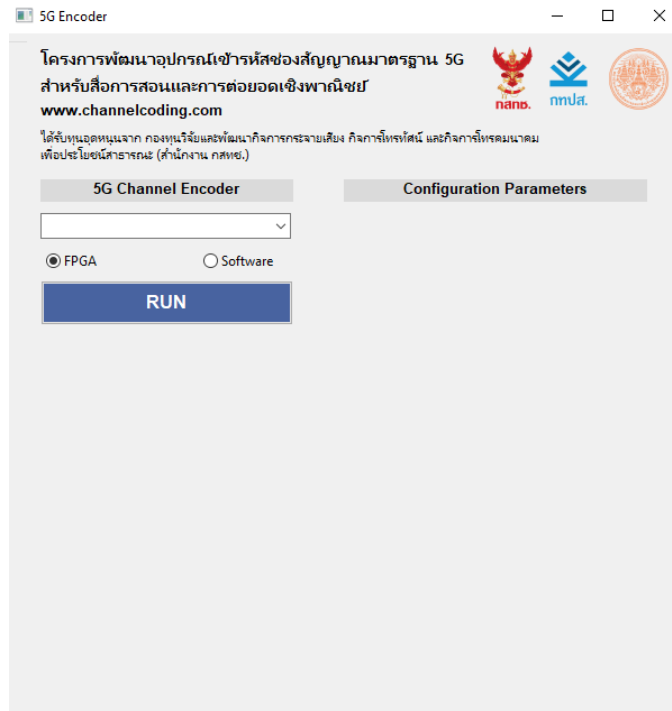
จากรูปที่ 4.235 หลังจากการกดปุ่ม “Key0” วงจรเข้ารหัส FPGA จะส่งผลลัพธ์การเข้ารหัส LDPC กลับไปยังคอมพิวเตอร์รูปแบบซีเรียล ส่วนต่อประสานกราฟิกกับผู้ใช้บนคอมพิวเตอร์จะประมวลผลข้อมูลที่ได้รับมาและแสดงผลการเข้ารหัส LDPC ของแต่ละกระบวนการเข้ารหัสบน ส่วนการแสดงผลดังรูปที่ 4.236



รูปที่ 4.236 ส่วนต่อประสานกราฟิกกับผู้ใช้แสดงผลลัพธ์ของแต่ละกระบวนการเข้ารหัส LDPC

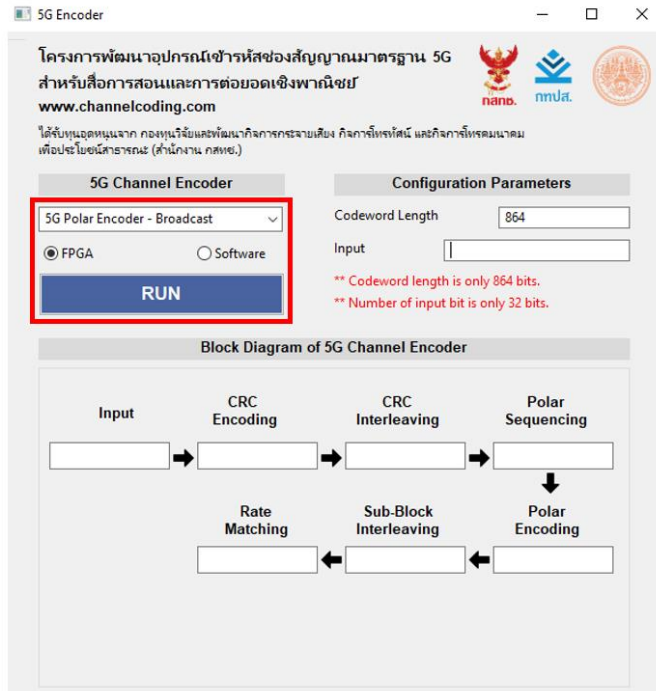
4.4.2 ชุดวงจรรหัสร่วมกับส่วนต่อประสานกราฟิกกับผู้ใช้เพื่อการเข้ารหัสช่องสัญญาณชนิด Polar Codes

จากการออกแบบและสร้างชุดวงจรรหัสร่วมกับส่วนต่อประสานกราฟิกกับผู้ใช้ เพื่อเชื่อมต่อข้อมูลระหว่างผู้ใช้งานผ่านส่วนต่อประสานกราฟิกกับผู้ใช้กับวงจรรหัสช่องสัญญาณชนิดรหัส Polar ผลการทดสอบการแสดงผลลัพธ์ของวงจรรหัสร่วมกับส่วนต่อประสานกราฟิกกับผู้ใช้ ผู้ใช้งานจะเริ่มเลือกวิธีการเข้ารหัส เลือกตัวเข้ารหัส และกรอกพารามิเตอร์และอินพุต ลงบนส่วนต่อประสานกราฟิกกับผู้ใช้ดังรูปที่ 4.237

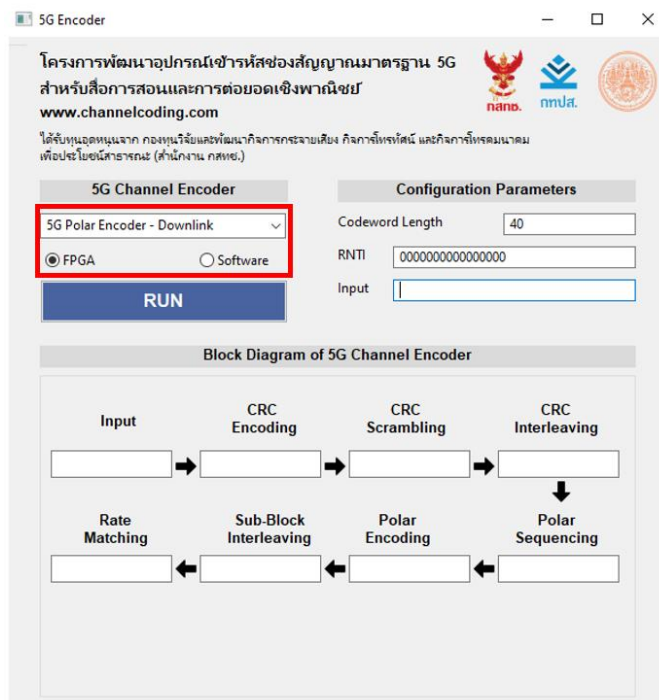


รูปที่ 4.237 หน้าแรกของส่วนต่อประสานกราฟิกกับผู้ใช้

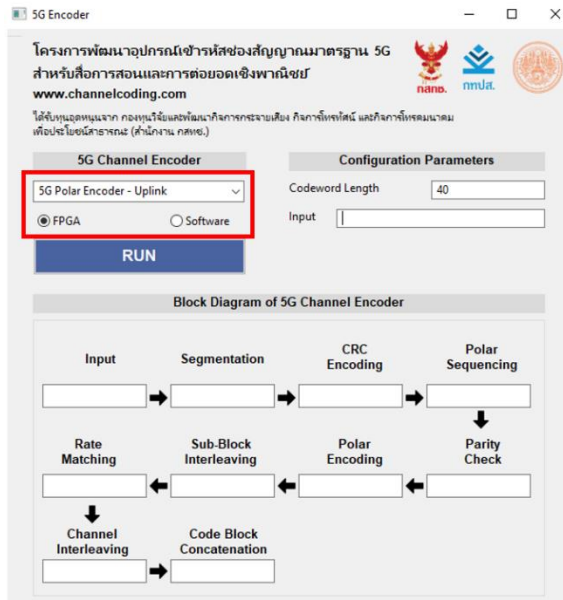
จากรูปที่ 4.237 ผู้ใช้งานสามารถเลือกวิธีการเข้ารหัสชนิดต่าง ๆ ได้จากเมนู drop down และเลือกตัวเข้ารหัสได้จากปุ่ม radio หัวข้อนี้เลือกวิธีการเข้ารหัสทั้ง 3 ชนิด “5G Polar Encoder - Broadcast” “5G Polar Encoder - Downlink” และ “5G Polar Encoder - Uplink” และเลือกตัวเข้ารหัส “FPGA” หลังจากเลือกวิธีการเข้ารหัสที่ต้องการ ส่วนต่อประสานกราฟิกกับผู้ใช้จะแสดงส่วนการรับค่าพารามิเตอร์และอินพุต และส่วนการแสดงผลที่สอดคล้องกับรหัสช่องสัญญาณที่ถูกเลือก แสดงดังรูปที่ 4.238 – 4.240



รูปที่ 4.238 ผู้ใช้งานเลือกวิธีการเข้ารหัส “5G Polar Encoder - Broadcast”
 และเลือกตัวเข้ารหัส “FPGA”

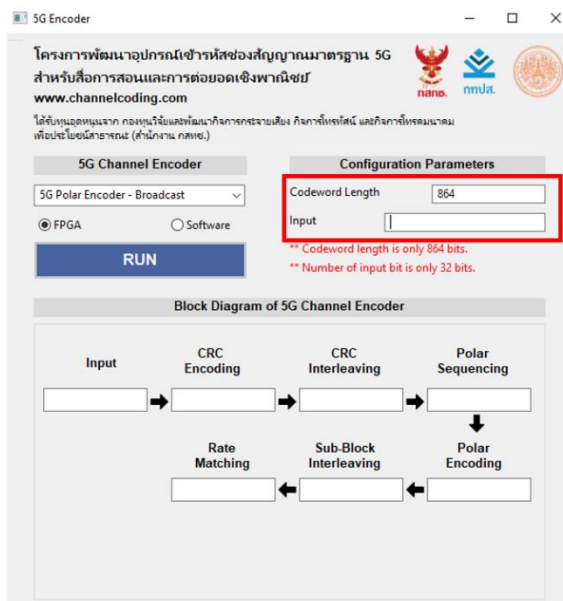


รูปที่ 4.239 ผู้ใช้งานเลือกวิธีการเข้ารหัส “5G Polar Encoder - Downlink”
 และเลือกตัวเข้ารหัส “FPGA”

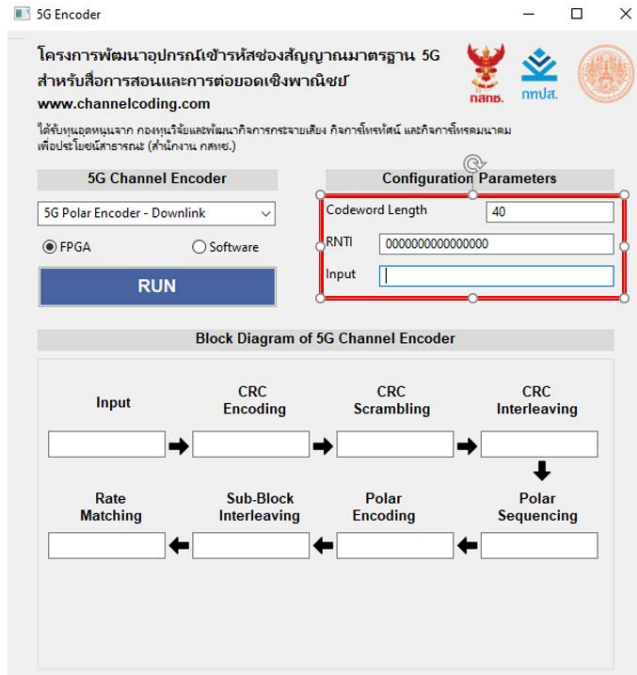


รูปที่ 4.240 ผู้ใช้งานเลือกวิธีการเข้ารหัส “5G Polar Encoder - Uplink” และเลือกตัวเข้ารหัส “FPGA”

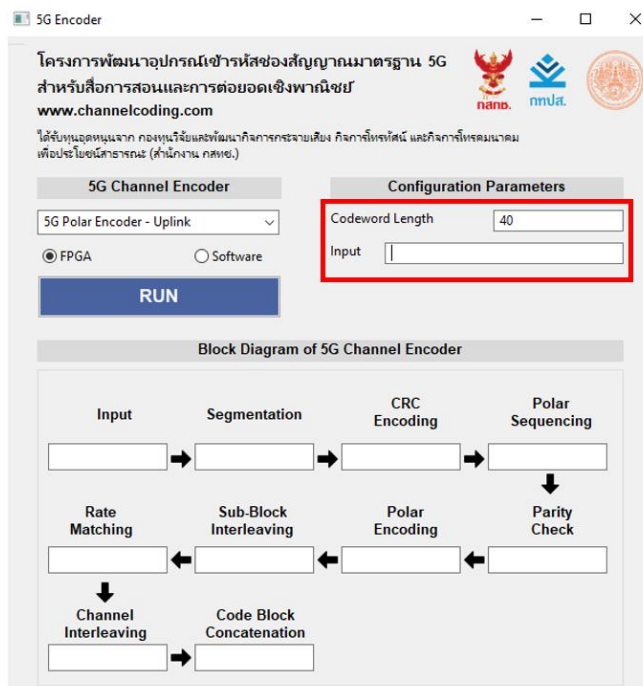
จากรูปที่ 4.238 – 4.240 ผู้ใช้งานสามารถรอกค่าพารามิเตอร์และอินพุตที่ต้องการเข้ารหัส ในส่วนการรับค่าพารามิเตอร์และอินพุต หัวข้อนี้จะกรอกค่าพารามิเตอร์และอินพุตดังรูปที่ 4.241 – 4.243



รูปที่ 4.241 ผู้ใช้งานกรอกค่าพารามิเตอร์และอินพุตในส่วนการรับค่าพารามิเตอร์ และอินพุต สำหรับรหัส Polar ช่องสัญญาณ Broadcast

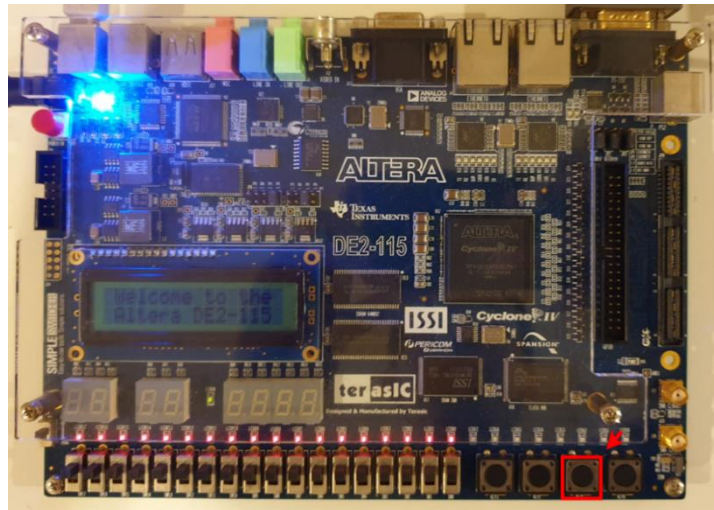


รูปที่ 4.242 ผู้ใช้งานกรอกค่าพารามิเตอร์และอินพุตในส่วนการรับค่าพารามิเตอร์ และอินพุต สำหรับรหัส Polar ช่องสัญญาณ Downlink

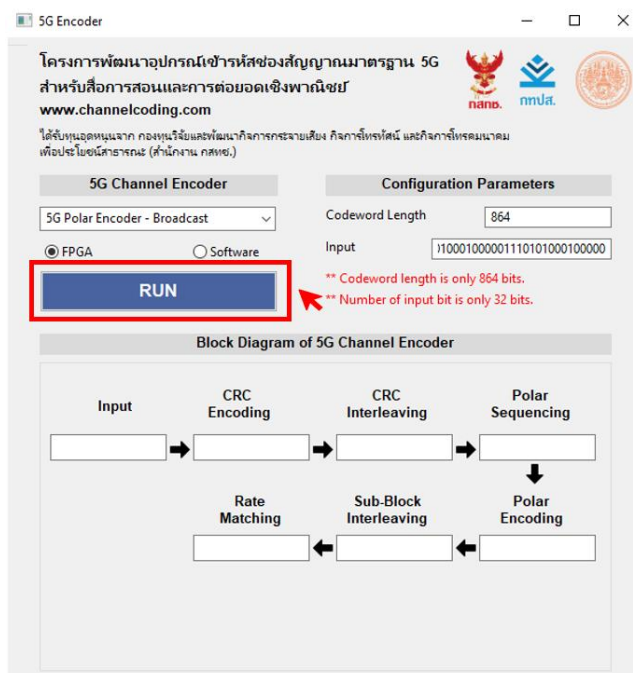


รูปที่ 4.243 ผู้ใช้งานกรอกค่าพารามิเตอร์และอินพุตในส่วนการรับค่าพารามิเตอร์ และอินพุต สำหรับรหัส Polar ช่องสัญญาณ Uplink

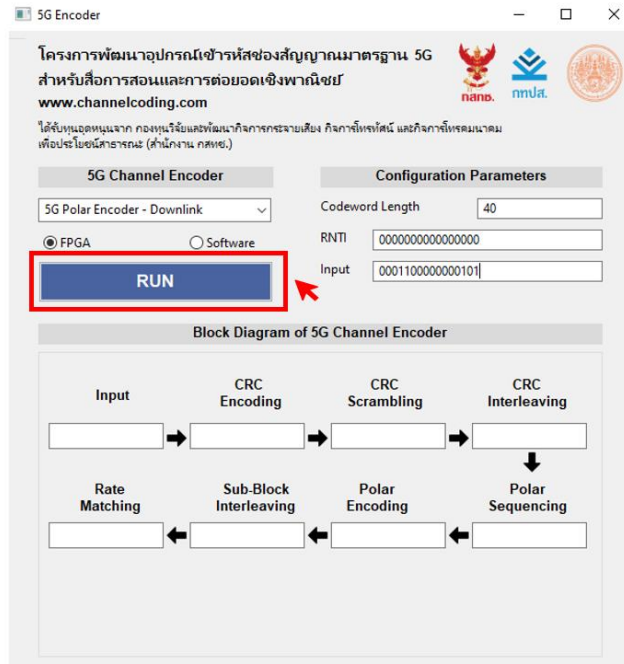
จากรูปที่ 4.241 - 4.243 ก่อนการกดปุ่ม “RUN” เพื่อส่งข้อมูลจากคอมพิวเตอร์ไปยังวงจร
 เข้ารหัส FPGA วงจรเข้ารหัสต้องทำการล้างข้อมูลที่ค้างในวงจรเข้ารหัสเสมอ โดยการกดปุ่ม “Key1”
 แสดงดังรูปที่ 4.244 จากนั้นผู้ใช้งานสามารถส่งข้อมูลจากคอมพิวเตอร์ไปยังวงจรเข้ารหัส FPGA ได้
 โดยการกดปุ่ม “RUN” ส่วนต่อประสานกราฟิกกับผู้ใช้ ดังรูปที่ 4.245 – 4.247



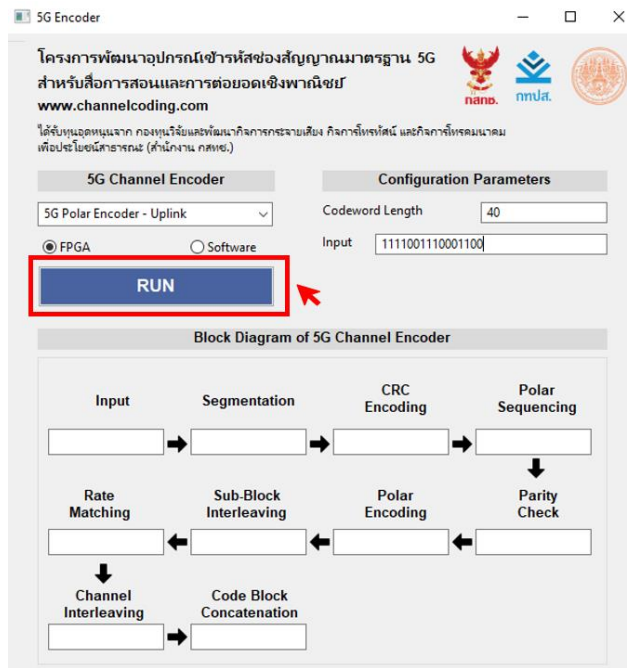
รูปที่ 4.244 การกดปุ่ม “Key1” เพื่อล้างค่าที่ค้างในวงจรเข้ารหัส FPGA



รูปที่ 4.245 การกดปุ่ม “RUN” เพื่อส่งข้อมูลจากคอมพิวเตอร์ไปยังวงจรเข้ารหัส FPGA
 สำหรับรหัส Polar ช่องสัญญาณ Broadcast

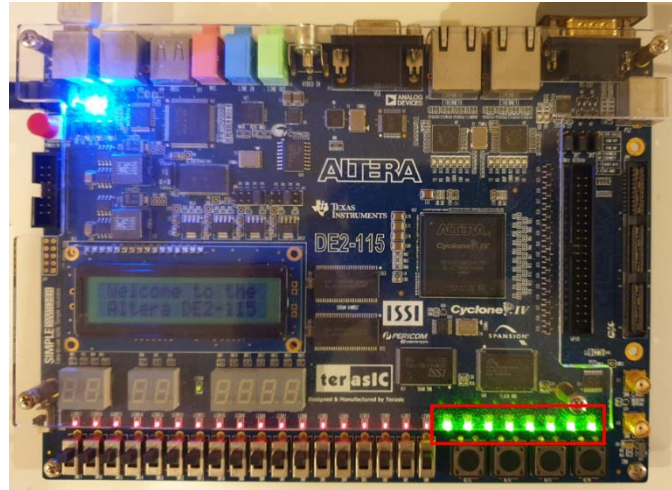


รูปที่ 4.246 การกดปุ่ม “RUN” เพื่อส่งข้อมูลจากคอมพิวเตอร์ไปยังวงจรเข้ารหัส FPGA สำหรับรหัส Polar ช่องสัญญาณ Downlink



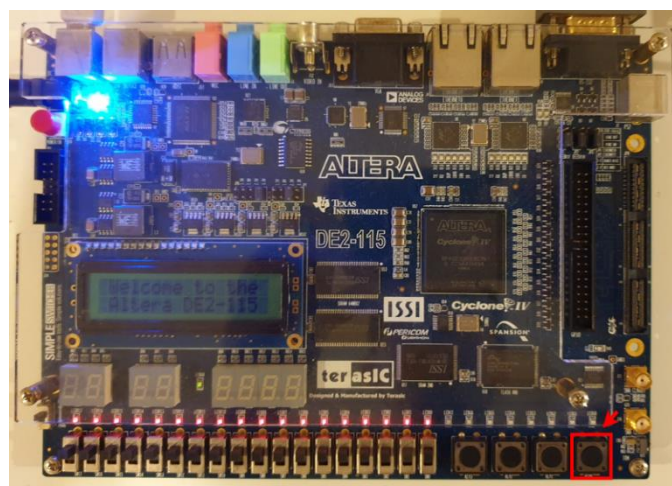
รูปที่ 4.247 การกดปุ่ม “RUN” เพื่อส่งข้อมูลจากคอมพิวเตอร์ไปยังวงจรเข้ารหัส FPGA สำหรับรหัส Polar ช่องสัญญาณ Uplink

จากรูปที่ 4.245 – 4.247 หลังการกดปุ่ม “RUN” ข้อมูลจะถูกส่งไปยังวงจรเข้ารหัส FPGA และทำการเข้ารหัส Polar สำหรับช่องสัญญาณต่าง ๆ เมื่อทำการเข้ารหัสเสร็จสิ้น วงจรจะแสดงสีบนหลอด LED จำนวน 8 ดวงทางขวา ดังรูปที่ 4.248



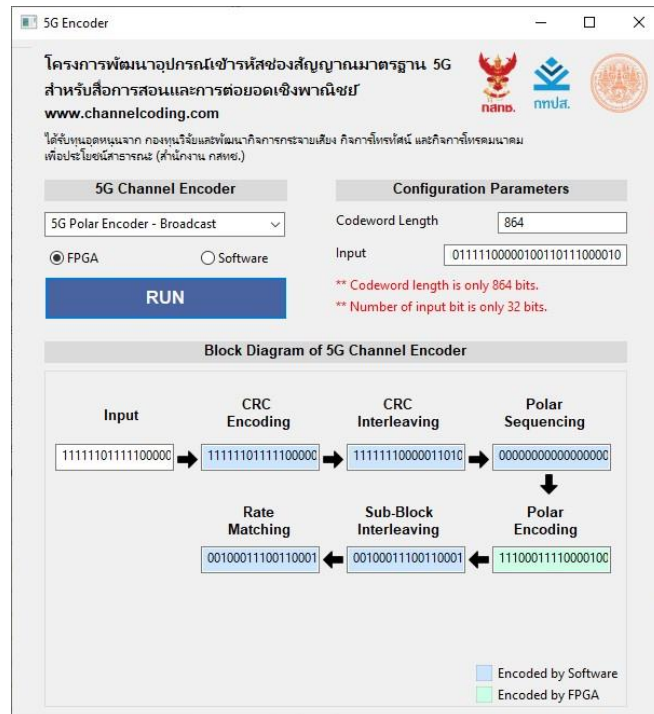
รูปที่ 4.248 หลอด LED 8 ดวงทางขวาติด เพื่อบ่งบอกถึงการเข้ารหัสในวงจรเข้ารหัส FPGA เสร็จสิ้น

จากรูปที่ 4.248 จะถือว่าวงจรเข้ารหัส FPGA ทำการเข้ารหัสเสร็จสิ้น ผู้ใช้งานจะสามารถกดปุ่ม “Key0” เพื่อส่งผลลัพธ์การเข้ารหัส Polar ช่องสัญญาณต่าง ๆ จากวงจรเข้ารหัส FPGA ไปยังคอมพิวเตอร์ได้ ดังรูปที่ 4.249

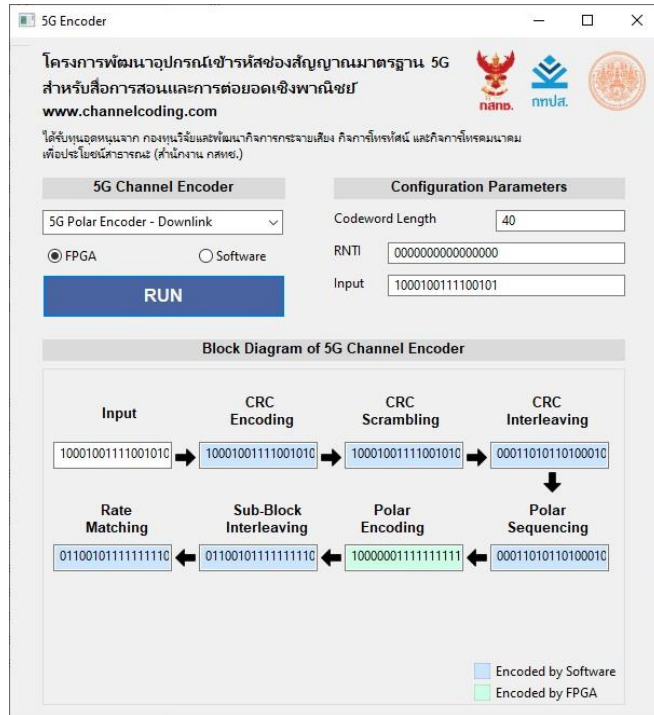


รูปที่ 4.249 การกดปุ่ม “Key0” เพื่อส่งผลลัพธ์จากวงจรเข้ารหัส FPGA ไปยังคอมพิวเตอร์

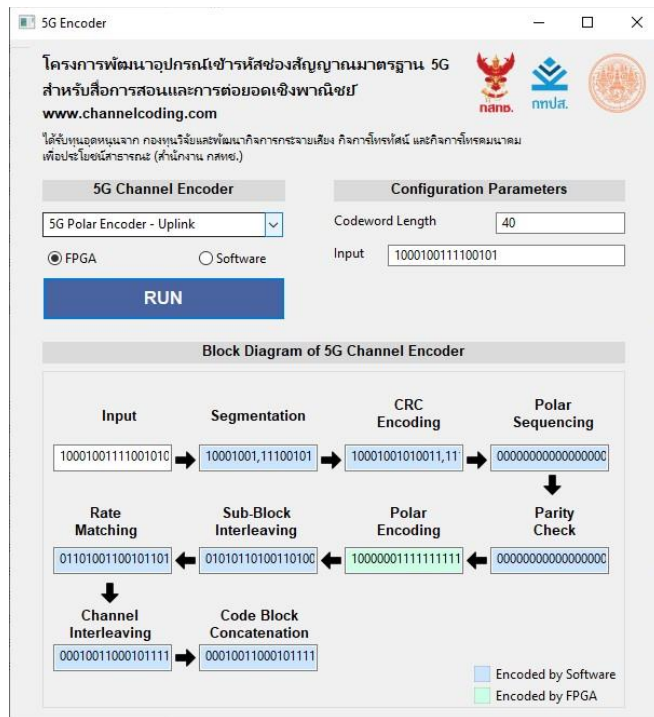
จากรูปที่ 4.249 หลังจากการกดปุ่ม “Key0” วงจรเข้ารหัส FPGA จะส่งผลลัพธ์การเข้ารหัส Polar ช่องสัญญาณต่าง ๆ กลับไปยังคอมพิวเตอร์รูปแบบซีเรียล ส่วนต่อประสานกราฟิกกับผู้ใช้บนคอมพิวเตอร์จะประมวลผลข้อมูลที่ได้รับมาและแสดงผลการเข้ารหัส Polar ช่องสัญญาณต่าง ๆ ของแต่ละกระบวนการเข้ารหัสบนส่วนการแสดงผลดังรูปที่ 4.250 – 4.252



รูปที่ 4.250 ส่วนต่อประสานกราฟิกกับผู้ใช้แสดงผลลัพธ์ของแต่ละกระบวนการเข้ารหัส Polar ช่องสัญญาณ Broadcast



รูปที่ 4.251 ส่วนต่อประสานกราฟิกกับผู้ใช้แสดงผลลัพธ์ของแต่ละกระบวนการเข้ารหัส Polar ช่องสัญญาณ Downlink



รูปที่ 4.252 ส่วนต่อประสานกราฟิกกับผู้ใช้แสดงผลลัพธ์ของแต่ละกระบวนการเข้ารหัส Polar ช่องสัญญาณ Uplink

บทที่ 5

สรุปผลการวิจัย และข้อเสนอแนะ

โครงการวิจัยนี้มีเป้าหมาย 2 ด้านได้แก่ 1) มิติด้านการศึกษาวิศวกรรมโทรคมนาคม และ 2) มิติด้านการพัฒนาอุปกรณ์โทรคมนาคมต้นแบบ โดยสรุปผลการวิจัยและข้อเสนอแนะของแต่ละเป้าหมายมีดังต่อไปนี้

1) มิติด้านการศึกษาวิศวกรรมโทรคมนาคม

ทีมวิจัยมีความประสงค์ที่จะพัฒนาอุปกรณ์การศึกษาที่สอดคล้องกับเทคโนโลยีสื่อสารในปัจจุบัน โดยทีมวิจัยได้ศึกษามาตรฐาน 3GPP TS 38.212 ซึ่งกำหนดกระบวนการเข้ารหัสที่ใช้ในระบบสื่อสารยุค 5G จากนั้น ทีมวิจัยได้พัฒนาซอฟต์แวร์เข้ารหัสด้วยภาษา python และ matlab (รายชื่อซอฟต์แวร์แสดงในภาคผนวก ก และ ข) เพื่อให้ นักศึกษาสามารถเลือกศึกษาภาษาที่ตนเองถนัด นอกจากนี้ ทีมวิจัยได้พัฒนาซอฟต์แวร์ประเภท GUI (รายละเอียดการใช้งานอยู่ในเอกสารแนบชื่อคู่มือการใช้งาน) เพื่อให้ นักศึกษาสามารถเห็นผลลัพธ์ของการเข้ารหัสได้อย่างง่าย

ทีมวิจัยได้พัฒนาเว็บไซต์ www.channelcoding.com เพื่อเผยแพร่เนื้อหาความรู้ด้านการเข้ารหัสของสัญญาณ รายละเอียดของมาตรฐาน 3GPP TS 38.212 และซอฟต์แวร์เข้ารหัสที่ทีมวิจัยได้พัฒนาขึ้น นอกจากนี้ ทีมวิจัยได้นำซอฟต์แวร์ไปใช้ประกอบการสอนในหลักสูตรที่เกี่ยวข้องกับวิศวกรรมโทรคมนาคม เช่น หลักสูตรวิศวกรรมโทรคมนาคม ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เป็นต้น (รายละเอียดอยู่ในภาคผนวก ง)

ทีมวิจัยคาดหวังว่าซอฟต์แวร์ที่พัฒนาขึ้นจะสามารถทำให้นักศึกษาไทยที่ศึกษาในด้านวิศวกรรมโทรคมนาคมมีความรู้ความเข้าใจเทคโนโลยีที่ใช้อยู่ในปัจจุบัน และก่อให้เกิดแรงผลักดันที่ให้นักศึกษาอยากที่จะพัฒนาเทคโนโลยีด้านวิศวกรรมโทรคมนาคมขึ้นมาใช้งานเอง

2) มิติของการพัฒนาอุปกรณ์โทรคมนาคมต้นแบบ

ทีมวิจัยมีความประสงค์ที่จะพัฒนาอุปกรณ์เข้ารหัสที่สามารถนำไปต่อเชิงพาณิชย์ได้ ทีมวิจัยจึงนำข้อกำหนดการเข้ารหัสมาตรฐาน 3GPP TS 38.212 มาพัฒนาบน chip ประเภท FPGA โดยทีมวิจัยได้เลือกใช้ภาษา VHDL (รายชื่อวงจรแสดงในภาคผนวก ค) ทั้งนี้ ภาษา VHDL ที่พัฒนาขึ้นในโครงการได้เผยแพร่ในเว็บไซต์ www.channelcoding.com เช่นกัน

ทีมวิจัยได้เผยแพร่การพัฒนาการเข้ารหัส 5G บน chip ประเภท FPGA ให้บริษัทไทยรับทราบถึงโอกาสและแนวทางการพัฒนาอุปกรณ์โทรคมนาคมต้นแบบ นอกจากนี้ ทีมวิจัยได้พัฒนา GUI (รายละเอียดการใช้งานอยู่ในเอกสารแนบชื่อคู่มือการใช้งาน) ให้เชื่อมกับอุปกรณ์ที่มี chip ประเภท FPGA เพื่อให้บริษัทและนักเรียกได้เห็นว่าคุณสมบัติสามารถทำงานได้จริง

ทีมวิจัยคาดหวังว่าต้นแบบวงจรเข้ารหัสมาตรฐาน 3GPP TS 38.212 มาพัฒนาบน chip ประเภท FPGA นี้จะถูกนำไปพัฒนาต่อยอดเป็น application ต่างๆ ตัวอย่างเช่น การพัฒนาอุปกรณ์ IoT ที่ใช้ระบบสื่อสาร 5G และการพัฒนา accelerator card ใน O-RAN เป็นต้น

บรรณานุกรม

- [1] 3rd Generation Partnership Project (3GPP), “Multiplexing and channel coding,” 3GPP 38.212 V.15.2.0, Jul 2018.
- [2] Saroch and Nontawat. “การใช้วิธี CRC (Cyclic Redundancy Checksum)” <http://www.geoci.ties.ws/boonsuwanno1995/page4>.
- [3] Supakit. “เทคนิคการตรวจสอบความถูกต้องของข้อมูล” http://home.npru.ac.th/supakit/Slide_71_22702/Error%20Detection.pdf.
- [4] 3GPP TS 38.214: “NR; Multiplexing and channel coding”, Release 15.
- [5] รศ.ดร.ปิยะ โควิพันธ์ ทวีวัฒน์. “การสื่อสารดิจิทัล การเข้ารหัสช่องสัญญาณ.” <http://home.npru.ac.th/piya/DigitalComm/file/Lec1415.pdf>.
- [6] กานต์ ศรีรัชตบุรณ์. “การออกแบบเมทริกซ์พาริตีเชิงขงรหัสแอลดีพีซี” วิทยานิพนธ์ปริญญาวิศวกรรมศาสตรมหาบัณฑิต, สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์, จุฬาลงกรณ์มหาวิทยาลัย, 2557.
- [7] Getsthiew. “ประวัติการสื่อสารไทยด้วยเทคโนโลยี HSPA” http://tookit-tt.blogspot.com/2008/08/hspa-high-speed-packet-access_12.html.
- [8] คณะเจ้าหน้าที่บริษัท อสมท จำกัด. “ระบบสื่อสารดิจิทัลสำหรับงาน Broadcasting” http://dtv.mcot.net/techno_one.php?dateone=1241680100.
- [9] ผศ.ดร.เวจิต ภาคย์พิสุทธิ์, รศ.ดร.ลัญจกร วุฒิสิริกุลกิจ, ดร.พิสิฐ วินิชชานันท์, ศ.ดร.ปิยะ โควิพันธ์ ทวีวัฒน์. รหัสช่องสัญญาณในระบบสื่อสารไร้สายยุค 5G.
- [10] J. H. Bae, A. Abotabl, H. P. Lin, K. B. song, and J. Lee. “An overview of channel coding for 5G NR cellular communications.” Cambridge University, 2019.
- [11] V. Bioglio, C. Condo and I. Land, “Design of Polar Codes in 5G New Radio,” in IEEE Communications Surveys & Tutorials, pp. 1-1, Jan 2020.

ภาคผนวก ก

ซอฟต์แวร์เข้ารหัสช่องสัญญาณด้วย MATLAB

ซอฟต์แวร์เข้ารหัสช่องสัญญาณสามารถเข้าไปดาวน์โหลดได้ที่เว็บไซต์ www.channelcoding.com โดยเข้าไปที่หน้าหลัก จากนั้นกด ชุดการเรียนรู้ > ชุดซอฟต์แวร์เข้ารหัสช่องสัญญาณ 5G > ซอฟต์แวร์เข้ารหัสด้วยภาษา MATLAB จากนั้นดาวน์โหลดชุดซอฟต์แวร์ โดยซอฟต์แวร์ทั้งหมดสามารถสรุปได้ตามตารางที่ 1

ตารางที่ 1 ซอฟต์แวร์เข้ารหัสช่องสัญญาณด้วย MATLAB

| รหัส ช่องสัญญาณ | ซอฟต์แวร์ | หน้าที่ |
|--------------------|--------------------------|---|
| LDPC codes | BitInterleaving.m | สลับตำแหน่งบิตข้อมูล |
| | CodeBlockConcatenation.m | ต่อเรียงบล็อกรหัส |
| | CRCEncoder0.m | เข้ารหัส CRC ทั้งหมด |
| | CRCEncoder1.m | เข้ารหัส CRC ข้อมูลและค่านวน ค่าพารามิเตอร์ |
| | CRCEncoder2.m | เข้ารหัส CRC บล็อกรหัสและค่านวน ค่าพารามิเตอร์ |
| | encodeLDPC5G.m | เข้ารหัส LDPC |
| | LDPCEncoding.m | เข้ารหัส LDPC และค่านวนค่าพารามิเตอร์ |
| | matrix5G.m | สร้างเมทริกซ์ H ของ 5G แบบที่ 1 |
| | matrix5GH.m | สร้างเมทริกซ์ H ของ 5G แบบที่ 2 |
| | RateMatching.m | เลือกบิตเพื่อส่งและค่านวนค่าพารามิเตอร์ |

| รหัส ช่องสัญญาณ | ซอฟต์แวร์ | หน้าที่ |
|--------------------|-------------------------|--|
| | Sementation.m | แบ่งข้อมูลเป็นบล็อกรหัสย่อยและคำนวณค่าพารามิเตอร์ |
| | test_5G_Transmission.m | ตัวอย่างการเรียกใช้ซอฟต์แวร์ และกำหนดค่าพารามิเตอร์ |
| Polar codes | CodeBlockSegmentation.m | กระบวนการแบ่งย่อยบล็อกรหัส |
| | CodedBitInterleaver.m | กระบวนการแทรกสลับบิตรหัสและการต่อบล็อกรหัส |
| | CRCAttachment.m | กระบวนการเข้ารหัส การสแครมบลิง และการแทรกสลับ CRC |
| | polarEncoder.m | ไฟล์ชุดคำสั่งเพื่อกำหนดพารามิเตอร์และเรียกใช้โมดูลฟังก์ชัน |
| | PolarEncoding.m | กระบวนการเข้ารหัสโพลาร์ การแทรกสลับบล็อกย่อย และการปรับอัตรารหัส |
| | PolarSequencing.m | กระบวนการลำดับช่องสัญญาณย่อยและการคำนวณบิตพาริตี |

ภาคผนวก ข

ซอฟต์แวร์เข้ารหัสช่องสัญญาณด้วย PYTHON

ซอฟต์แวร์เข้ารหัสช่องสัญญาณสามารถเข้าไปดาวน์โหลดได้ที่เว็บไซต์ www.channelcoding.com โดยเข้าไปที่หน้าหลัก จากนั้นกด ชุดการเรียนรู้ > ชุดซอฟต์แวร์เข้ารหัสช่องสัญญาณ 5G > ซอฟต์แวร์เข้ารหัสด้วยภาษา PYTHON จากนั้นดาวน์โหลดชุดซอฟต์แวร์ โดยซอฟต์แวร์ทั้งหมดสามารถสรุปได้ตามตารางที่ 2

ตารางที่ 2 ซอฟต์แวร์เข้ารหัสช่องสัญญาณด้วย PYTHON

| รหัสช่องสัญญาณ | ซอฟต์แวร์ | หน้าที่ |
|----------------|---------------------------|--|
| LDPC codes | BitInterleaving.py | สลับตำแหน่งบิตข้อมูล |
| | CodeBlockConcatenation.py | ต่อเรียงบล็อกรหัส |
| | CRCEncoder0.py | เข้ารหัส CRC ทั่วไป |
| | CRCEncoder1.py | เข้ารหัส CRC ข้อมูลและค่านวนค่าพารามิเตอร์ |
| | CRCEncoder2.py | เข้ารหัส CRC บล็อกรหัสและค่านวนค่าพารามิเตอร์ |
| | encodeLDPC5G.py | เข้ารหัส LDPC |
| | LDPCEncoding.py | เข้ารหัส LDPC และค่านวนค่าพารามิเตอร์ |
| | matrix5GH.py | สร้างเมทริกซ์ H ของ 5G แบบที่ 2 |
| | RateMatching.py | เลือกบิตเพื่อส่งและค่านวนค่าพารามิเตอร์ |
| | Segmentation.py | แบ่งข้อมูลเป็นบล็อกรหัสย่อยและค่านวนค่าพารามิเตอร์ |

| รหัส ช่องสัญญาณ | ซอฟต์แวร์ | หน้าที่ |
|--------------------|--------------------------|--|
| | UserDataChannelCoding.py | ตัวอย่างการเรียกใช้ซอฟต์แวร์ และกำหนดค่าพารามิเตอร์ |
| Polar codes | CodeBlockSegmentation.py | กระบวนการแบ่งย่อยบล็อกรหัส |
| | CodedBitInterleaver.py | กระบวนการแทรกสลับบิตรหัสและการต่อบล็อกรหัส |
| | CRCAttachment.py | กระบวนการเข้ารหัส การสแครมบลิง และการแทรกสลับ CRC |
| | polarEncoder.py | ไฟล์ชุดคำสั่งเพื่อกำหนดพารามิเตอร์และเรียกใช้โมดูลฟังก์ชัน |
| | PolarEncoding.py | กระบวนการเข้ารหัสโพลาร์ การแทรกสลับบล็อกย่อย และการปรับอัตรารหัส |
| | PolarSequencing.py | กระบวนการลำดับช่องสัญญาณย่อยและการคำนวณบิตพาริตี |

ภาคผนวก ค

รายละเอียดของอุปกรณ์ FPGA รุ่น Altera DE2-115

วงจรเข้ารหัสเข้ารหัสช่องสัญญาณสามารถดาวน์โหลดได้ที่เว็บไซต์ www.channelcoding.com โดยเข้าไปที่หน้าหลัก จากนั้นกด ชุดการเรียนรู้ > อุปกรณ์เข้ารหัสช่องสัญญาณบน FPGA > วงจรเข้ารหัสด้วยภาษา VHDL จากนั้นดาวน์โหลดไฟล์ VHDL โดยไฟล์ทั้งหมดสามารถสรุปได้ตามตารางที่ 3 โดยไฟล์เตอร์ชื่อ bg1 สำหรับมาตรฐาน Base Graph 1 และ bg2 สำหรับมาตรฐาน Base Graph 2

ตารางที่ 3 วงจรเข้ารหัสช่องสัญญาณบนบอร์ด FPGA

| รหัสช่องสัญญาณ | ชื่อไฟล์ VHDL | หน้าที่ |
|----------------|---------------------|---|
| LDPC codes | TxSerial.vhdl | ส่งบิตข้อมูลไปภายนอกด้วยโปรโตคอล UART |
| | RxSerial.vhdl | รับบิตข้อมูลจากภายนอกด้วยโปรโตคอล UART |
| | Demux.vhdl | จัดเรียงบิตข้อมูลขาเข้า |
| | Mux.vhdl | จัดเรียงบิตข้อมูลขาออก |
| | crc_attachment.vhdl | เข้ารหัส CRC และเข้ารหัส CRC ให้บล็อกข้อมูลย่อย |
| | seg.vhdl | แบ่งข้อมูลเป็นบล็อกย่อย |
| | new_circulant.vhdl | เอนติตี้เพื่อเชื่อมไฟล์ที่เกี่ยวข้องกับการเข้ารหัส LDPC |
| | data_cyclic.vhdl | คำนวณผลรวมในกระบวนการเข้ารหัส LDPC |
| | parity.vhdl | คำนวณบิตพาริตีในกระบวนการเข้ารหัส LDPC |

| รหัสช่องสัญญาณ | ชื่อไฟล์ VHDL | หน้าที่ |
|---------------------------|--------------------------------------|--|
| | circulant.vhdl | หมุนวนข้อมูลไปทางซ้าย |
| | circulantr.vhdl | หมุนวนข้อมูลไปทางขวา |
| | BG1.vhdl | เอนติตีเพื่อเชื่อมต่อไฟล์ทั้งหมดสำหรับกราฟ ฐาน 1 และกราฟฐาน 2 |
| | BG2.vhdl | |
| | rate_matching.vhdl | ปรับอัตรารหัส |
| | bit_interleaver.vhdl | แทรกสลับสลับบิตข้อมูลของบล็อกข้อมูล |
| | concatenation.vhdl | นำบล็อกข้อมูลย่อยมาเรียงต่อและส่งออก |
| Polar codes | code_block_concatenation.vhdl | ต่อบล็อกรหัส |
| | code_block_segmentation.vhdl | แยกบล็อกรหัส |
| | coded_bit_interleaver.vhdl | แทรกสลับคำรหัส |
| | crc_attachment.vhdl | คำนวณบิต CRC และต่อท้ายบิตข้อมูล |
| | demux.vhdl | จัดเรียงบิตข้อมูลขาเข้า |
| | mux.vhdl | จัดเรียงบิตข้อมูลขาออก |
| | polar_encoder_5G.vhdl | เอนติตีเพื่อเชื่อมต่อไฟล์ทั้งหมด |
| | polar_encoder_N_2.vhdl | โครงสร้างเข้ารหัสโพลาร์ขนาด 2 บิต |
| | polar_encoder_N_4.vhdl | โครงสร้างเข้ารหัสโพลาร์ขนาด 4 บิต |
| | polar_encoder_N_8.vhdl | โครงสร้างเข้ารหัสโพลาร์ขนาด 8 บิต |
| | polar_encoder_N_16.vhdl | โครงสร้างเข้ารหัสโพลาร์ขนาด 16 บิต |
| | polar_encoder_N_32.vhdl | โครงสร้างเข้ารหัสโพลาร์ขนาด 32 บิต |
| | polar_encoder_N_64.vhdl | โครงสร้างเข้ารหัสโพลาร์ขนาด 64 บิต |
| | polar_encoder_N_128.vhdl | โครงสร้างเข้ารหัสโพลาร์ขนาด 128 บิต |
| | polar_encoder_N_256.vhdl | โครงสร้างเข้ารหัสโพลาร์ขนาด 256 บิต |
| polar_encoder_N_512.vhdl | โครงสร้างเข้ารหัสโพลาร์ขนาด 512 บิต | |
| polar_encoder_N_1024.vhdl | โครงสร้างเข้ารหัสโพลาร์ขนาด 1024 บิต | |

| รหัสช่องสัญญาณ | ชื่อไฟล์ VHDL | หน้าที่ |
|----------------|-------------------------------------|---|
| | polar_encoder.vhdl | เรียงบิตแช่แข็ง บิตข้อมูล และบิต CRC และทำการเข้ารหัสโพลาไรซ์ |
| | polar_sequencing.vhdl | คำนวณตำแหน่งบิตแช่แข็ง บิตข้อมูล บิต CRC และรวมถึงบิตพาริตี |
| | rate_matching.vhdl | ปรับอัตรารหัส |
| | Rxserial.vhdl | รับบิตข้อมูลจากภายนอกด้วยโปรโตคอล UART |
| | sub_block_deinterleaver_N_32.vhdl | โครงสร้างแทรกสลับบล็อกย่อยย้อนกลับ ขนาด 32 บิต |
| | sub_block_deinterleaver_N_64.vhdl | โครงสร้างแทรกสลับบล็อกย่อยย้อนกลับ ขนาด 64 บิต |
| | sub_block_deinterleaver_N_128.vhdl | โครงสร้างแทรกสลับบล็อกย่อยย้อนกลับ ขนาด 128 บิต |
| | sub_block_deinterleaver_N_256.vhdl | โครงสร้างแทรกสลับบล็อกย่อยย้อนกลับ ขนาด 256 บิต |
| | sub_block_deinterleaver_N_512.vhdl | โครงสร้างแทรกสลับบล็อกย่อยย้อนกลับ ขนาด 512 บิต |
| | sub_block_deinterleaver_N_1024.vhdl | โครงสร้างแทรกสลับบล็อกย่อยย้อนกลับ ขนาด 1024 บิต |
| | sub_block_deinterleaver.vhdl | แทรกสลับบล็อกย่อยย้อนกลับ |
| | sub_block_interleaver_N_32.vhdl | โครงสร้างแทรกสลับบล็อกย่อยขนาด 32 บิต |
| | sub_block_interleaver_N_64.vhdl | โครงสร้างแทรกสลับบล็อกย่อยขนาด 64 บิต |

| รหัสช่องสัญญาณ | ชื่อไฟล์ VHDL | หน้าที่ |
|----------------|-----------------------------------|--|
| | sub_block_interleaver_N_128.vhdl | โครงสร้างแทรกสลับบล็อกย่อยขนาด 128 บิต |
| | sub_block_interleaver_N_256.vhdl | โครงสร้างแทรกสลับบล็อกย่อยขนาด 256 บิต |
| | sub_block_interleaver_N_512.vhdl | โครงสร้างแทรกสลับบล็อกย่อยขนาด 512 บิต |
| | sub_block_interleaver_N_1024.vhdl | โครงสร้างแทรกสลับบล็อกย่อยขนาด 1024 บิต |
| | sub_block_interleaver.vhdl | แทรกสลับบล็อกย่อย |
| | Txserial.vhdl | ส่งบิตข้อมูลไปภายนอกด้วยโปรโตคอล UART |

ภาคผนวก ง

การเผยแพร่ผลผลิตของโครงการ

ภายหลังจากที่มิวิจัยได้ทำการพัฒนาชุดซอฟต์แวร์และอุปกรณ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ที่มิวิจัยได้มีการเผยแพร่ผลผลิตของโครงการในรูปแบบเว็บไซต์ และการนำเสนอแก่ภาคอุตสาหกรรมในรูปแบบการประชุมออนไลน์ ซึ่งมีรายละเอียดโดยสังเขป ดังนี้

1. การพัฒนาเว็บไซต์เผยแพร่ความรู้การเข้ารหัสช่องสัญญาณมาตรฐาน 5G

ที่มิวิจัยได้พัฒนาเว็บไซต์ www.channelcoding.com เพื่อนำเสนอการใช้งาน “รหัสช่องสัญญาณ” ในมาตรฐานการสื่อสารต่าง ๆ เช่น มาตรฐาน 5G (3GPP NR) และ Wi-Fi (IEEE 802.11) นอกจากนี้ ที่มิวิจัยได้แสดงการพัฒนาชุดซอฟต์แวร์ต้นแบบ (ภาคผนวก ก. หัวข้อที่ 1) และอุปกรณ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G บนบอร์ด FPGA (ภาคผนวก ก. หัวข้อที่ 2) รวมถึงชุดการเรียนรู้การเข้ารหัสช่องสัญญาณ (ภาคผนวก ก. หัวข้อที่ 3) ในเว็บไซต์ประกอบไปด้วย 6 เมนูหลักดังต่อไปนี้

1.1 หน้าแรก

หน้าแรกแสดงคำอธิบายเกี่ยวกับเนื้อหาเว็บไซต์ในภาพรวม เช่น จุดประสงค์ในการพัฒนาเว็บไซต์ กลุ่มผู้ใช้งานที่เหมาะสมกับเว็บไซต์ ช่องทางติดต่อผู้พัฒนาเว็บไซต์เพื่อซักถามปัญหาการใช้งานชุดการเรียนรู้ และปัญหาการใช้งานซอฟต์แวร์ต่าง ๆ

ยินดีต้อนรับเข้าสู่ www.channelcoding.com

ทำความรู้จักกับ [CHANNEL CODING]
สำหรับมาตรฐานการสื่อสารยุคใหม่

หมายเหตุ
เว็บไซต์นี้อยู่ระหว่างการพัฒนา เนื้อหาบางส่วนอาจยังไม่ได้ออกเต็มลงไป อย่างไรก็ตามผู้เยี่ยมชมสามารถติดตามพัฒนาได้ในเมนู "เกี่ยวกับเรา"

บทความล่าสุด

- GUI แสดงผลลัพธ์ของชุดซอฟต์แวร์
- GUI แสดงผลลัพธ์ของชุดอุปกรณ์
- รหัสคอนโวลูชัน
- รหัสแฮมมิง
- รหัสโพลาร์
- รหัสแอลดีพีซี
- รหัสเทอร์โบ

เว็บไซต์นี้แสดงอะไร ? **เว็บไซต์นี้เหมาะกับใคร ?** **ติดต่อพวกเขาได้ที่ไหน ?**

เว็บไซต์นี้นำเสนอการใช้งาน "รหัสช่องสัญญาณ" ในมาตรฐานการสื่อสารต่าง ๆ ไม่ว่าจะเป็น มาตรฐาน 5G (3GPP NR) และ Wi-Fi (IEEE 802.11) นอกจากนี้ เว็บไซต์ยังแสดงการพัฒนาชุดอุปกรณ์เข้ารหัสช่องสัญญาณมาตรฐาน 5G บนบอร์ด FPGA และชุดซอฟต์แวร์การเรียนรู้การเข้ารหัสช่องสัญญาณด้วยภาษา MATLAB และ Python หากมีผู้ใช้งานมีข้อสงสัยในเรื่องต่างๆ สามารถตั้งกระทู้คำถามในกระดานสนทนา โดยจะมีผู้เชี่ยวชาญเข้ามาตอบคำถามโดยเร็วที่สุด

เมนูหน้าแรกยังแสดงความเคลื่อนไหวล่าสุดของเว็บไซต์ เช่น บทความล่าสุด บทความแนะนำ และความเห็นล่าสุด เป็นต้น เพื่อให้ผู้เข้าชมได้รับข่าวสารอัปเดตใหม่บนเว็บไซต์อยู่เสมอ

บทความล่าสุด

- GUI แสดงผลผลลัพธ์ของชุดซอฟต์แวร์
- GUI แสดงผลผลลัพธ์ของชุดอุปกรณ์
- รหัสคอนโวลูชัน
- รหัสแฮมมิง
- รหัสโพลาร์
- รหัสแอลดีพีซี
- รหัสเทอร์โบ

บทความแนะนำ

- ชุดทดสอบ (2)
- ทฤษฎีรหัสช่องสัญญาณ (6)
- มาตรฐาน 5G (5)

กระทู้ล่าสุด

ธันข ศรีสุภา บน การสื่อสารไร้สายยุคที่ 5

นอกจากนี้ เมนูหน้าแรกยังแสดงผู้ให้การสนับสนุนการจัดทำโครงการซึ่งได้แก่ กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมเพื่อประโยชน์สาธารณะ รวมถึงแสดงเว็บไซต์ที่เกี่ยวข้องเพื่อใช้ประกอบความเข้าใจเนื้อหาภายในเว็บไซต์นี้

สนับสนุนโดย



เว็บไซต์ที่เกี่ยวข้อง



ข่าวสาร ความรู้ เกี่ยวกับเทคโนโลยีแห่งอนาคต การสื่อสารไร้สายยุค 5G ที่จะเข้ามาเป็นส่วนหนึ่งในชีวิตประจำวันในรอบ ๆ ด้าน ไม่เพียงแต่โทรศัพท์มือถือ แต่เป็นทุกสิ่งที่อยู่รอบตัว ซึ่งสามารถติดตามความเคลื่อนไหวของเทคโนโลยี 5G ในประเทศไทยได้ที่ 5G-Thailand.org

เว็บไซต์นี้จัดทำขึ้นภายใต้ โครงการพัฒนาความร่วมมือระดับประเทศของการสื่อสารไร้สายในยุค 5G สนับสนุนโดย กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมเพื่อประโยชน์สาธารณะ

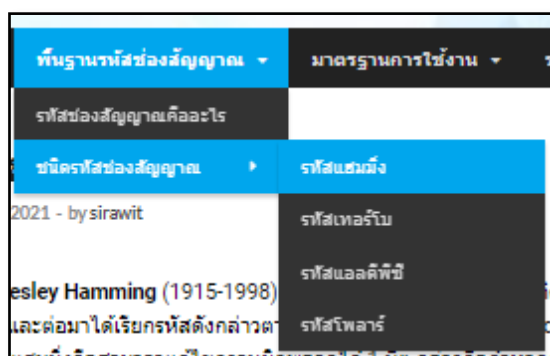
Powered by WordPress and HiMag.

1.2 พื้นฐานรหัสช่องสัญญาณ

เมนูนี้บรรยายถึงความรู้พื้นฐานของรหัสช่องสัญญาณ โดยทีมวิจัยได้ให้คำจำกัดความว่าเหตุใดรหัสช่องสัญญาณจึงจำเป็นในระบบสื่อสารดิจิทัล เมนูนี้ประกอบไปด้วยเมนูย่อยที่กล่าวถึง “ความหมายของรหัสช่องสัญญาณ” รวมถึงคำพารามิเตอร์ เทคนิคการเข้ารหัส และการถอดรหัสของรหัสช่องสัญญาณแต่ละชนิด



ในเมนูย่อย “ชนิดรหัสช่องสัญญาณ” ผู้เข้าชมสามารถเยี่ยมชมรายละเอียดเชิงลึกของรหัสช่องสัญญาณแต่ละชนิด เช่น



- รหัสแฮมมิง (Hamming) ซึ่งเป็นรหัสช่องสัญญาณชนิดแรกที่เกิดขึ้นในปี 1950

รหัสแฮมมิง

ณภาพที่ 18, 2021 - by sirawit

Richard Wesley Hamming (1915-1998) นักคณิตศาสตร์ชาวอเมริกันได้คิดค้นรหัสช่องสัญญาณชนิดแรกขึ้นมาในปี 1950 และต่อมาได้เรียกรหัสดังกล่าวตามชื่อผู้คิดค้นว่า "Hamming code" หรือ "รหัสแฮมมิง" หลักการเบื้องต้นของรหัสแฮมมิงคือสามารถแก้ไขความผิดพลาดได้ 1 บิต กล่าวคือถ้าหากคำรหัสรับได้มีความผิดพลาดไม่เกิน 1 บิต รหัสแฮมมิงจะสามารถระบุตำแหน่งที่มีความผิดพลาดบิตได้อย่างถูกต้อง



รูปที่ 1 Richard Wesley Hamming ผู้คิดค้นรหัสแฮมมิงในปี 1950



รูปที่ 2 Richard Wesley Hamming กับรางวัลของเขามี 1988 ในการพัฒนารหัสแฮมมิง

- รหัสเทอร์โบ (Turbo) ซึ่งเป็นรหัสช่องสัญญาณที่ถูกคิดค้นโดยหนึ่งในนักวิจัยชาวไทย

รหัสเทอร์โบ

ณภาพที่ 18, 2021 - by sirawit

รหัสเทอร์โบ (1993) เป็นรหัสช่องสัญญาณที่ถูกคิดค้นโดยนักวิจัย 3 ท่านคือ Claude Berrou, Alain Glavieux และ Punya Thitimajshima ซึ่งหรือคือ ดร. ปิญญา ฐิติวัฒน์ชมา อดีตอาจารย์พิเศษที่รศมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งเป็นนักวิจัยชาวไทยนั่นเอง



Claude Berrou



Alain Glavieux



Punya Thitimajshima

จุดเด่นของรหัสเทอร์โบ ทำงานได้ดีแม้ว่าช่องสัญญาณมีค่า SNR ต่ำ รวมถึงสามารถแก้ไขข้อผิดพลาดได้ดี และมีสมรรถนะเข้าใกล้ขีดจำกัดของแชนนอน โดยอาศัยกระบวนการเข้ารหัสและถอดรหัส ที่ไม่ซับซ้อนรหัสเทอร์โบสามารถประยุกต์ใช้ในหลายๆ แอปพลิเคชัน เช่น ระบบโทรศัพท์เคลื่อนที่ยุคที่สาม (3G) และสี่ (4G) ซึ่งได้ป็นรหัสเทอร์โบมาใช้เป็นมาตรฐานสำหรับการติดต่อสื่อสารระหว่าง สถานีฐานและเครื่องโทรศัพท์เคลื่อนที่ เป็นต้น

- รหัสแอลดีพีซี (LDPC) ซึ่งเป็นรหัสช่องสัญญาณยอติติที่ถูกใช้อย่างแพร่หลาย ๆ ในแอปพลิเคชันปัจจุบัน รหัส LDPC ได้ถูกนำไปใช้สำหรับการสื่อสารไร้สายยุค 5G ซึ่งทีมวิจัยได้อธิบายรายละเอียดการออกแบบ และข้อกำหนดเพิ่มเติมในเมนูถัดไป

รหัสแอลดีพีซี

คุณภาพวันที่ 18, 2021 - by sirawit

รหัสพาริตีความหนาแน่นต่ำ (Low-Density Parity-Check codes) หรือที่นิยมเรียกโดยย่อว่า รหัสแอลดีพีซี จัดเป็นรหัสแก้ไขความผิดพลาดข้างหน้า (Forward Error Correcting codes) ประเภทหนึ่งที่มีประสิทธิภาพสูงมาก เพราะสามารถให้สมรรถนะการแก้ไขความผิดพลาดที่เข้าใกล้ขีดจำกัดของเชนนอน (Shannon's limit) ได้ ซึ่งหมายความว่า ระบบสามารถไขปริศนาจากช่องสัญญาณสื่อสารได้ถึงศักยภาพสูงสุดเท่าที่เป็นไปได้ รหัสแอลดีพีซีได้รับการคิดค้นขึ้นโดย Robert Gallager โดยเป็นส่วนหนึ่งของงานวิทยานิพนธ์ปริญญาเอกแห่งสถาบัน Massachusetts Institute of Technology (MIT) ใน พ.ศ. 1960



รูปที่ 1 Robert Gallager ผู้คิดค้นรหัสแอลดีพีซีในปี ค.ศ. 1960



รูปที่ 2 Robert Gallager รับรางวัลผู้คิดค้นรหัสแอลดีพีซี จากองค์กร IEEE ในปี ค.ศ. 1990

- รหัสโพลาร์ (Polar) ซึ่งเป็นรหัสช่องสัญญาณใหม่ล่าสุดที่เพิ่งถูกคิดค้นในปี 2009 ซึ่งรหัส Polar ก็ได้ถูกนำไปใช้สำหรับการสื่อสารไร้สายยุค 5G เช่นเดียวกับรหัส LDPC

รหัสโพลาร์

คุณภาพวันที่ 18, 2021 - by sirawit

รหัสโพลาร์ ถูกคิดค้นในปี พ.ศ. 2552 โดยเออร์ดัล อริกาน (Erdal Arıkan) ซึ่งเป็นรหัสช่องสัญญาณชนิดแรกที่สามารถพิสูจน์ได้โดยตรงว่าเข้าใกล้ทฤษฎีความจุของสัญญาณของเชนนอน (Shannon's limit) ต่างจากรหัสช่องสัญญาณชนิดอื่นที่ต้องใช้การสุ่มทางอ้อม



รูปที่ 1 Erdal Arıkan ผู้คิดค้นรหัสโพลาร์ในปี พ.ศ. 2552



รูปที่ 2 Erdal Arıkan รับรางวัลในฐานะบิดาแห่งรหัสโพลาร์จากบริษัท Huawei

"I am honored to be here today receiving this award," Dr. Erdal Arıkan said to more than a hundred of his fellow scientists and engineers. "As engineers, there is no greater reward than seeing our ideas turn into reality."



Erdal Arıkan
Father of polar code

1.3 มาตรฐานการใช้งาน

เมนูนี้แสดงถึงการประยุกต์ใช้รหัสช่องสัญญาณในมาตรฐานการสื่อสารต่าง ๆ เช่น มาตรฐาน 5G (3GPP NR) และ Wi-Fi (IEEE 802.11) ดังมีรายละเอียดต่อไปนี้

1.3.1) มาตรฐาน 5G (3GPP NR) อธิบายถึงมาตรฐานการสื่อสารไร้สายยุค 5G ด้วยมาตรฐาน 3GPP NR (New Radio) ในเมนู “มาตรฐาน 5G (3GPP NR)” ประกอบไปด้วย 5 เมนูย่อย ดังนี้



- การสื่อสารไร้สายยุคที่ 5 อธิบายถึงการพัฒนาเทคโนโลยีการสื่อสารไร้สายตั้งแต่ยุคเริ่มต้น 1G จนถึง 5G

การสื่อสารไร้สายยุคที่ 5

กุมภาพันธ์ 18, 2021 - by sirawit

ในอดีตมนุษย์ได้มีการคิดค้นวิธีการสื่อสารเพื่อแลกเปลี่ยนข้อมูลข่าวสารระหว่างกัน จากการติดต่อสื่อสารระหว่างกันในกลุ่มชนเดียวกัน พัฒนาการสื่อสารข้ามกลุ่มชนชนข้ามประเทศ และทั่วโลก ตามลำดับ จะเห็นได้จากอดีต ปัจจุบันมนุษย์ได้มีความพยายามคิดค้นวิธีการติดต่อสื่อสารเพื่อให้เกิดความสะดวกและรวดเร็วเพิ่มมากขึ้น



จนกระทั่งในปัจจุบันโลกเราได้ก้าวเข้าสู่ยุคทองของการสื่อสารไร้สาย ซึ่งสามารถติดต่อสื่อสารถึงกันได้ทุกที่ทั่วโลก การสื่อสารไร้สายทำให้เกิดบริการเช่นการสื่อสารระยะไกลซึ่งเป็นไปได้เลยหรือทำไม่ได้ในทางปฏิบัติที่จะดำเนินการแบบใช้สาย ตัวอย่างที่ชัดเจนที่สุดคือโทรศัพท์มือถือที่เข้ามาแทนที่โทรศัพท์บ้าน โดยหากสังเกตจากจำนวนยอดผู้ใช้งานโทรศัพท์มือถือในประเทศไทยฮอตเดดล่าสุด ณ ต้นเดือน มีนาคม 2562 มีจำนวนถึงกว่า 90 ล้านเลขหมาย ฉะนั้นจะเห็นได้ว่าการสื่อสารอยู่ใกล้ชิดกับการใช้ชีวิตประจำวันของเรา จากจุดเริ่มต้นเพียงเพื่อการสื่อสารด้วยเสียง ในปัจจุบันเราสามารถดูวิดีโอและเล่นเกมออนไลน์แบบคมชัดสูงได้อย่างสะดวกสบาย เทคโนโลยีการสื่อสารไร้สายได้พัฒนาอย่างต่อเนื่องทั้งในด้าน คุณภาพ ราคา และความเร็ว

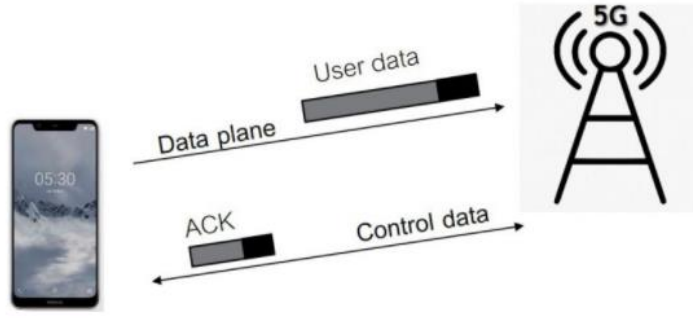


- ประเภทรหัสช่องสัญญาณในมาตรฐาน 5G อธิบายถึงข้อเรียกร้องในด้านความน่าเชื่อถือ และความถูกต้องในการส่งข้อมูลสำหรับการสื่อสารไร้สายยุค 5G รวมถึงการจำแนกประเภทข้อมูลที่มีการรับ และส่ง ระหว่างโทรศัพท์มือถือ และสถานีฐาน

ประเภทของรหัสช่องสัญญาณในมาตรฐาน 5G

กุมภาพันธ์ 18, 2021 - bysirawit

ในส่วนของเทคโนโลยี 5G การสื่อสารระหว่างสถานีฐานและโทรศัพท์เคลื่อนที่จะประกอบไปด้วยข้อมูลอยู่ 2 ชนิด ชนิดแรกคือ User data ซึ่งเป็นข้อมูลที่แท้จริงของผู้ใช้งานที่พยายามส่งไปให้ผู้รับปลายทาง ชนิดที่สองคือ Control data ซึ่งเป็นข้อมูลที่โทรศัพท์เคลื่อนที่ส่งข้อความตอบกลับไปยังสถานีฐานว่าได้รับข้อมูลถูกต้องหรือไม่ หรือสภาพสัญญาณบริเวณนั้นเป็นเช่นไร



- ความเป็นมาของมาตรฐานการเข้ารหัส 5G อธิบายถึงองค์กร 3rd Generation Partnership Project (3GPP) ที่มีหน้าที่ในการประชุมหารือเพื่อเลือกเฟ้นรหัสช่องสัญญาณที่สามารถตอบสนองข้อเรียกร้องในด้านความน่าเชื่อถือ และความถูกต้องของข้อมูลได้มากที่สุด

ความเป็นมาของมาตรฐานการเข้ารหัส 5G

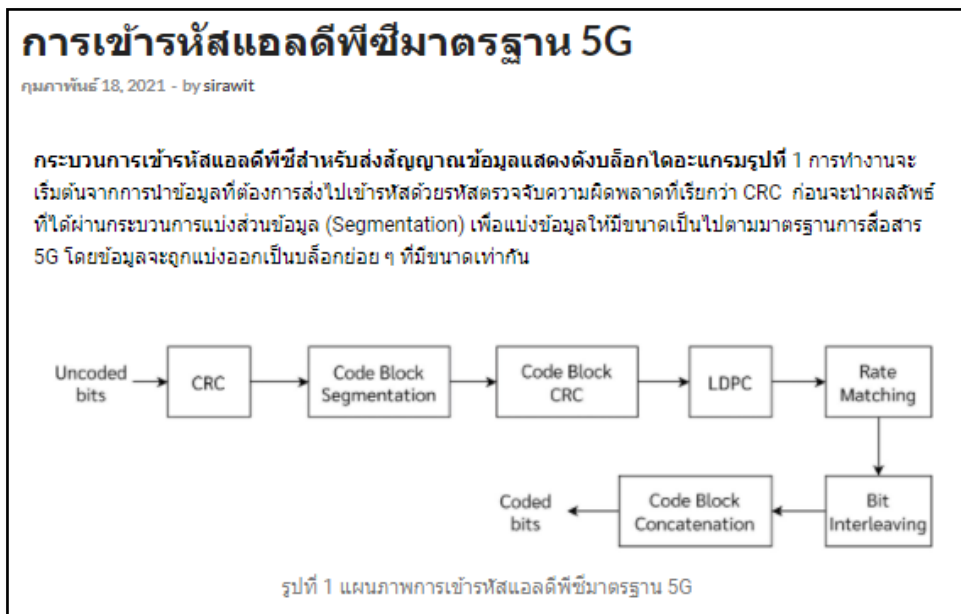
กุมภาพันธ์ 18, 2021 - bysirawit

The 3rd Generation Partnership Project หรือ องค์กร 3GPP คือหน่วยงานตามข้อตกลงความร่วมมือในการพัฒนารายละเอียดด้านมาตรฐานทางเทคนิค (Standards Development Organizations; SDOs) และการจัดทำรายงานต่างๆ ระหว่างองค์กรพัฒนามาตรฐานกับหน่วยงานอื่นๆ ที่เกี่ยวข้องในเครือข่ายระบบโทรศัพท์เคลื่อนที่ที่ใช้กันทั่วโลก องค์กร 3GPP เป็นการรวมกลุ่มอุตสาหกรรมโทรคมนาคมยักษ์ใหญ่ของโลก เช่น

- สมาคมอุตสาหกรรมวิทยุและธุรกิจญี่ปุ่น (Association of Radio Industries and Businesses : ARIB)
- สหพันธ์เพื่อทางออกอุตสาหกรรมโทรคมนาคมสหรัฐอเมริกา (Automatic terminal information service : ATIS)
- สมาคมมาตรฐานการสื่อสารจีน (China Communications Standards Association : CCSA)
- สถาบันมาตรฐานโทรคมนาคมยุโรป (European Telecommunication Standards Institute : ETSI)
- สมาคมเทคโนโลยีโทรคมนาคมเกาหลี (Telecommunications Technology Association : TTA)
- คณะกรรมการเทคโนโลยีโทรคมนาคมญี่ปุ่น (Telecommunication Technology Committee : TTC)



- การเข้ารหัสแวลดีพีซีมาตรฐาน 5G สรุปผลลัพธ์การประชุมของ 3GPP เพื่อเลือกเฟ้นรหัสช่องสัญญาณมาใช้ในการสื่อสารไร้สายยุค 5G ซึ่งได้แก่ “รหัส LDPC” ที่ถูกใช้ในช่องสัญญาณข้อมูล (Data channel)

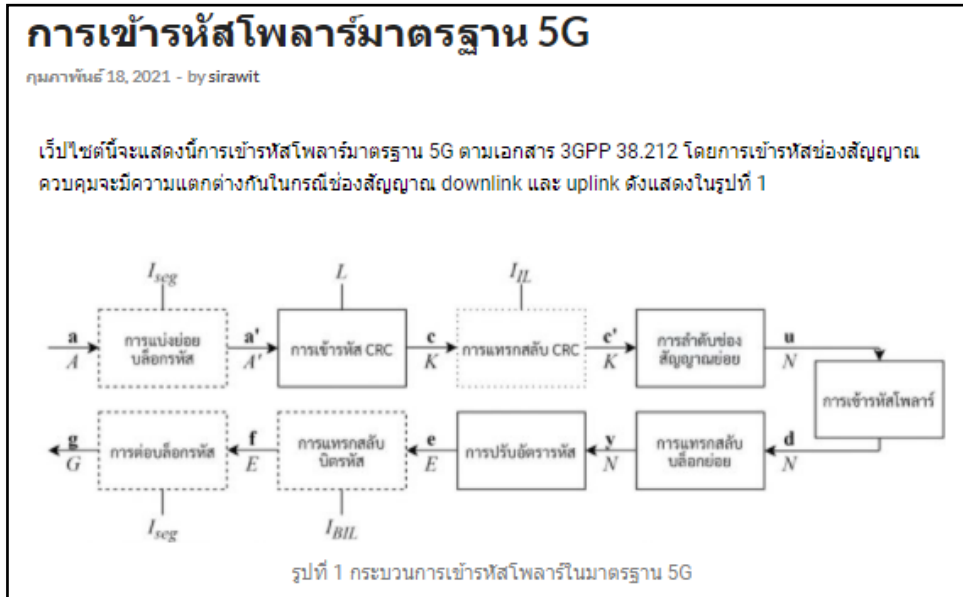


โดยผู้รับชมสามารถศึกษารายละเอียดเชิงลึกของแต่ละบล็อกข้อมูลในแถบด้านล่างของเว็บไซต์

รายละเอียดเพิ่มเติม

| |
|---|
| ▶ การเข้ารหัส CRC (Cyclic Redundancy Check) |
| ▶ การแบ่งส่วนข้อมูล (Segmentation) |
| ▶ วงจรเข้ารหัสแวลดีพีซี |
| ▶ การปรับอัตรารหัส (Rate Matching) |
| ▶ การแทรกสลับบิต (Bit Interleaving) |
| ▶ การต่อเรียงบล็อกรหัส (Code Block Concatenation) |

- การเข้ารหัสโพลาร์มาตรฐาน 5G ให้รายละเอียดเชิงลึกเกี่ยวกับการใช้รหัส Polar ในการสื่อสารไร้สายยุค 5G สำหรับช่องสัญญาณควบคุม (Control channel)



โดยผู้รับชมสามารถศึกษารายละเอียดเชิงลึกของแต่ละบล็อกข้อมูลในแถบด้านล่างของเว็บไซต์

| รายละเอียดเพิ่มเติม |
|--------------------------|
| ▶ การแบ่งย่อยบล็อกรหัส |
| ▶ การเข้ารหัส CRC |
| ▶ การสแครมบลิง CRC |
| ▶ การแทรกสลับ CRC |
| ▶ การลำดับช่องสัญญาณย่อย |
| ▶ การคำนวณเบิตพาร์ตี |
| ▶ การเข้ารหัสโพลาไร |
| ▶ การแทรกสลับบล็อกย่อย |
| ▶ การปรับอัตรารหัส |
| ▶ การแทรกสลับบิตรหัส |
| ▶ การต่อบล็อกรหัส |

1.3.2) มาตรฐาน Wi-Fi (IEEE 802.11) อธิบายถึงมาตรฐานการสื่อสารของ Wi-Fi ด้วยมาตรฐาน IEEE 802.11 ที่ถูกกำหนดขึ้นโดยสถาบันวิชาชีพวิศวกรไฟฟ้าและอิเล็กทรอนิกส์ (Institute of Electrical and Electronics Engineers: IEEE) ซึ่งประกอบไปด้วย 4 เมนูย่อยดังนี้



- ความเป็นมามาตรฐาน IEEE 802.11 อธิบายถึงกำเนิดของมาตรฐาน IEEE 802.11 รวมถึงอุปกรณ์การเชื่อมต่อระบบเครือข่ายไร้สาย (Wireless Local Area Network: WLAN) ภายใต้มาตรฐานดังกล่าว

ความเป็นมามาตรฐาน IEEE 802.11

IEEE 802.11 คือมาตรฐานการทำงานของระบบเครือข่ายไร้สายที่ถูกกำหนดขึ้นโดย "สถาบันวิชาชีพวิศวกรไฟฟ้าและอิเล็กทรอนิกส์ (Institute of Electrical and Electronics Engineers: IEEE)" ซึ่งเป็นสถาบันระดับนานาชาติที่ไม่วางผลกำไร ทำหน้าที่ดูแลเกี่ยวกับไฟฟ้าและคอมพิวเตอร์ โดยมาตรฐาน IEEE 802.11 นั้นถูกนำมาใช้เพื่อการเชื่อมต่ออุปกรณ์ไร้สายเข้าด้วยกัน ซึ่งโดยปกติแล้วการเชื่อมต่อระบบเครือข่ายไร้สายจะต้องใช้อุปกรณ์สองชิ้นที่สำคัญคือ

- **แอคเซสพอยต์ (Wireless Access Point)** เป็นตัวกลางที่ช่วยในการเชื่อมต่อระหว่าง ตัวรับ-ส่ง สัญญาณของผู้ใช้กับเราเตอร์ผ่านทางสายนำสัญญาณ เช่น สายแลน หรือสายโทรศัพท์ ADSL หรือ สายใยแก้วนำแสง เป็นต้น
- **ตัวรับ-ส่ง สัญญาณไร้สาย (Wireless Adapter)** เป็นอุปกรณ์ทำหน้าที่ รับ-ส่ง สัญญาณระหว่างตัวรับส่งแต่ละตัวด้วยกัน หรือระหว่างตัวลูกข่ายกับแอคเซสพอยต์





Wireless access point



Wireless Adapter

- วิวัฒนาการของมาตรฐาน IEEE 802.11 อธิบายถึงมาตรฐานที่ถูกพัฒนาต่อยอดจากมาตรฐาน IEEE 802.11 ดั้งเดิม โดยกำหนดเป็นตัวอักษรภาษาอังกฤษต่อท้ายคือ a, b, g, n, ac และ ax ตามลำดับ ซึ่งแต่ละมาตรฐานมีคุณสมบัติด้านความเร็ว และคลื่นความถี่ที่ใช้ แตกต่างกัน

วิวัฒนาการของมาตรฐาน IEEE 802.11

อุปกรณ์ที่รองรับมาตรฐาน IEEE 802.11 ต้องมีความสามารถในการรับส่งข้อมูลความเร็วขั้นต่ำ 1 และ 2 Mbps ด้วยอินฟราเรด หรือคลื่นวิทยุที่ความถี่ 2.4 GHz และมีกลไก WEP (Wired Equivalent Privacy) ซึ่งเป็นทางเลือกสำหรับสร้างความปลอดภัยให้กับเครือข่าย WLAN อย่างไรก็ตาม มาตรฐาน IEEE 802.11 ในเวอร์ชันแรกนั้น มีประสิทธิภาพค่อนข้างต่ำ อีกทั้งไม่มีการรองรับหลักการการจัดการบริหารแบนด์วิดท์หรือช่องทางของระบบเครือข่าย (Quality of Service : QoS) นอกจากนี้ กลไกรักษาความปลอดภัยที่ใช้อยู่ยังมีช่องโหว่อยู่มาก ด้วยเหตุนี้ องค์กร IEEE จึงได้จัดตั้งคณะทำงานขึ้นมาเพื่อทำการปรับปรุงมาตรฐานเพิ่มเติมให้มีศักยภาพสูงขึ้น หลังจากนั้น จึงถือกำเนิดเทคโนโลยี IEEE 802.11 ในเวอร์ชันพัฒนาหลายมาตรฐานที่เป็นที่รู้จักกันดังนี้

| IEEE Standard | 802.11a | 802.11b | 802.11g | 802.11n | 802.11ac | 802.11ax |
|-------------------|---------|---------|---------|----------------|----------------|----------------|
| Release | 1999 | 1999 | 2003 | 2009 | 2014 | 2019 |
| Frequency | 5Ghz | 2.4Ghz | 2.4Ghz | 2.4Ghz 5Ghz | 2.4Ghz 5Ghz | 2.4Ghz 5Ghz |
| Maximum Data Rate | 54Mbps | 11Mbps | 54Mbps | 600Mbps | 1.3Gbps | 10-12Gbps |

Wi-Fi evolution overview by release date, frequency and maximum data rate

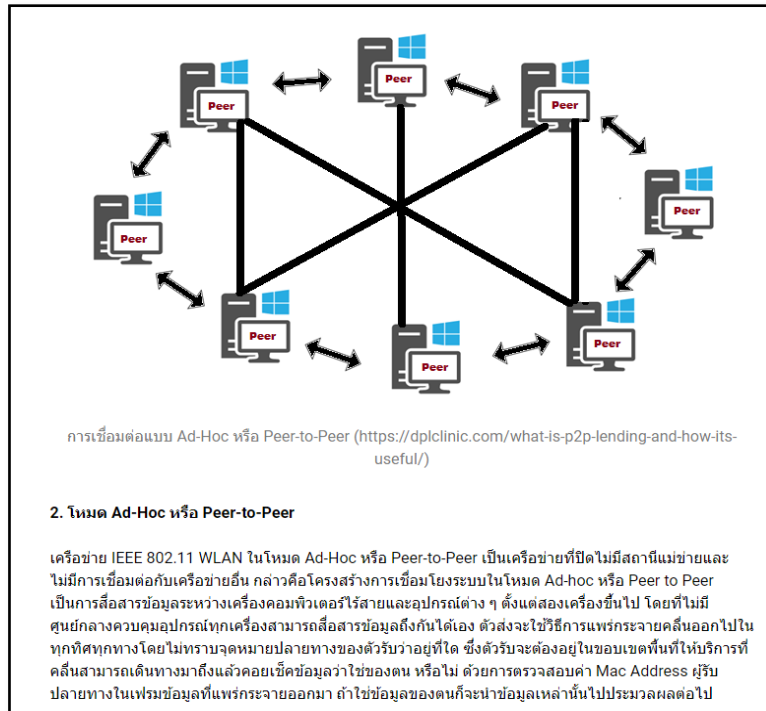
- ลักษณะการเชื่อมต่อมาตรฐาน IEEE 802.11 ให้รายละเอียดเกี่ยวกับลักษณะการเชื่อมต่อของอุปกรณ์ภายในเครือข่าย WLAN ซึ่งประกอบด้วย 2 ลักษณะ คือ โหมด Infrastructure และโหมด Ad-Hoc หรือ Peer-to-Peer

การเชื่อมต่อแบบกลุ่มโครงสร้าง (Infrastructure) (<http://www.aicomputer.co.th>)

1. โหมด Infrastructure

หรือเรียกอีกชื่อหนึ่งว่า "ระบบเครือข่ายไร้สายแบบ Client / server" โดยทั่วไปอุปกรณ์ไร้สายในมาตรฐาน IEEE 802.11 จะเชื่อมต่อกันในโหมดโครงสร้างพื้นฐาน (Infrastructure) ซึ่งเป็นโหมดที่อนุญาตให้อุปกรณ์ภายในระบบเครือข่ายไร้สาย (WLAN) สามารถเชื่อมต่อกับเครือข่ายอื่นได้ การเชื่อมต่อ WLAN ในโหมด Infrastructure นั้นประกอบไปด้วยอุปกรณ์ 2 ประเภทได้แก่

- สถานีผู้ใช้ (Client Station) ซึ่งก็คืออุปกรณ์คอมพิวเตอร์ (Desktop, Laptop, หรือ PDA ต่างๆ) ที่มีอุปกรณ์ Client Adapter เพื่อใช้รับส่งข้อมูลผ่าน IEEE 802.11 WLAN และ
- สถานีแม่ข่าย (Access Point) ซึ่งทำหน้าที่เชื่อมต่อเชื่อมสถานีผู้ใช้เข้ากับเครือข่าย

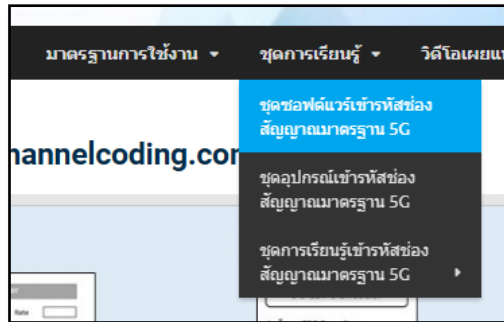


- การเข้ารหัสแอสสมิตรีมาตรฐาน IEEE 802.11 แสดงการใช้งานรหัสแอสสมิตรีสำหรับระบบเครือข่ายไร้สาย (WLAN) ภายใต้มาตรฐาน IEEE 802.11 (ซึ่งยังอยู่ในระหว่างการจัดทำ)
-



1.4 ชุดการเรียนรู้

ในเมนูชุดการเรียนรู้แสดงการพัฒนาชุดทดสอบการเข้ารหัสช่องสัญญาณในมาตรฐานการสื่อสารไร้สาย 5G โดยเมนูนี้เหมาะสำหรับ นักเรียน/นักศึกษา หรือผู้ที่สนใจเกี่ยวกับการใช้งานรหัสช่องสัญญาณเพื่อเพิ่มประสิทธิภาพและความน่าเชื่อถือในการส่งข้อมูลดิจิทัล เมนูนี้ประกอบไปด้วย 3 เมนูย่อยคือ



1.4.1) การพัฒนาชุดซอฟต์แวร์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ผู้ใช้งานสามารถศึกษารายละเอียด และวิธีการพัฒนาชุดซอฟต์แวร์ของกระบวนการเข้ารหัส LDPC และ Polar ด้วยภาษา MATLAB และ Python

ชุดซอฟต์แวร์เข้ารหัสช่องสัญญาณมาตรฐาน 5G

ทีมวิจัยเลือกใช้ภาษา [MATLAB](#) และ [Python](#) ในการพัฒนาชุดซอฟต์แวร์สำหรับเข้ารหัสช่องสัญญาณมาตรฐาน 5G ประกอบด้วย 4 ส่วน ได้แก่

- 1) **5G LDPC Encoder** – Downlink/Uplink เป็นตัวเข้ารหัส LDPC ที่ใช้งานในช่องสัญญาณข้อมูล (Data Channel) แบบ Downlink และ Uplink
- 2) **5G Polar Encoder** – Downlink เป็นตัวเข้ารหัส Polar ที่ใช้งานในช่องสัญญาณควบคุม (Control Channel) แบบ Downlink
- 3) **5G Polar Encoder** – Uplink เป็นตัวเข้ารหัส Polar ที่ใช้งานในช่องสัญญาณควบคุม (Control Channel) แบบ Downlink
- 4) **5G Polar Encoder** – Broadcast เป็นตัวเข้ารหัส Polar ที่ใช้งานในช่องสัญญาณควบคุม (Control Channel) แบบ Broadcast







นอกจากนี้ ผู้รับชมสามารถดาวน์โหลดชุดซอฟต์แวร์ที่พัฒนาด้วยภาษา MATLAB และ Python ไปใช้งานได้โดยไม่มีค่าใช้จ่าย

[ดาวน์โหลด] ชุดซอฟต์แวร์การเข้ารหัสของสัญญาณมาตรฐาน 5G

พัฒนาด้วยภาษา MATLAB และ Python เพื่อจำลองกระบวนการเข้ารหัสข้อมูลด้วยรหัสของสัญญาณ สำหรับมาตรฐานการสื่อสารไร้สายยุค 5G



โหลด: รหัส LDPC
[*MATLAB code]



โหลด: รหัส Polar
[*MATLAB code]



โหลด: รหัส LDPC
[*Python code]



โหลด: รหัส Polar
[*Python code]

1.4.2) การพัฒนาชุดอุปกรณ์การเข้ารหัสของสัญญาณมาตรฐาน 5G ผู้ใช้งานสามารถศึกษารายละเอียด และวิธีการพัฒนาชุดซอฟต์แวร์ของกระบวนการเข้ารหัส LDPC และ Polar สำหรับอุปกรณ์ FPGA ด้วยภาษา VHDL

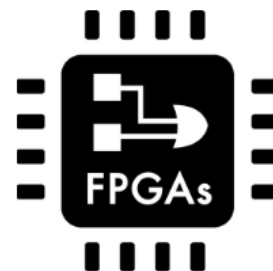
ชุดอุปกรณ์เข้ารหัสของสัญญาณมาตรฐาน 5G

ทีมวิจัยพัฒนาชุดอุปกรณ์เข้ารหัสของสัญญาณมาตรฐาน 5G โดยใช้ภาษา VHSIC Hardware Description Language (VHDL) สำหรับบอร์ด FPGA ซึ่งประกอบด้วย 4 ส่วน ได้แก่

- 1) 5G LDPC Encoder – Downlink/Uplink เป็นตัวเข้ารหัส LDPC ที่ใช้งานในช่องสัญญาณข้อมูล (Data Channel) แบบ Downlink และ Uplink
- 2) 5G Polar Encoder – Downlink เป็นตัวเข้ารหัส Polar ที่ใช้งานในช่องสัญญาณควบคุม (Control Channel) แบบ Downlink
- 3) 5G Polar Encoder – Uplink เป็นตัวเข้ารหัส Polar ที่ใช้งานในช่องสัญญาณควบคุม (Control Channel) แบบ Downlink
- 4) 5G Polar Encoder – Broadcast เป็นตัวเข้ารหัส Polar ที่ใช้งานในช่องสัญญาณควบคุม (Control Channel) แบบ Broadcast

VHDL

Hardware Description Language



นอกจากนี้ ผู้รับชมสามารถดาวน์โหลดซอฟต์แวร์ที่พัฒนาด้วยภาษา VHDL สำหรับอุปกรณ์ FPGA ไปใช้งานได้โดยไม่มีค่าใช้จ่าย

[ดาวน์โหลด] ชุดอุปกรณ์การเข้ารหัสของสัญญาณมาตรฐาน 5G

พัฒนาด้วยภาษา VHDL เพื่อจำลองกระบวนการเข้ารหัสข้อมูลด้วยรหัสของสัญญาณมาตรฐานการสื่อสารไร้สายยุค 5G บนอุปกรณ์ FPGA



โหลด: รหัส LDPC
[*VHDL code]



โหลด: รหัส Polar
[*VHDL code]

1.4.3) การพัฒนาชุดการเรียนรู้การเข้ารหัสของสัญญาณมาตรฐาน 5G ผู้ใช้งานสามารถศึกษารายละเอียด ของชุดการเรียนรู้ในรูปแบบส่วนประสานผู้ใช้งาน (GUI) ซึ่งรับผลลัพธ์การเข้ารหัสจากชุดซอฟต์แวร์ (หัวข้อ 1.4.1) และชุดอุปกรณ์ FPGA (หัวข้อ 1.4.2)

GUI แสดงผลลัพธ์ของชุดซอฟต์แวร์

มีนาคม 14, 2021 - by sirawit

ทีมวิจัยเลือกใช้ภาษา Python เพื่อสร้างส่วนต่อประสานกับผู้ใช้ (Graphic User Interface: GUI) เพื่อแสดงผลการเข้ารหัสของสัญญาณจากชุดซอฟต์แวร์ที่พัฒนาด้วยภาษา Python ที่มีการวิจัยออกแบบหน้าต่าง GUI เพื่อให้ผู้ใช้สามารถใช้งานได้โดยง่ายตาย

GUI (Python)

โครงการศึกษาอุปกรณ์เข้ารหัสสัญญาณมาตรฐาน 5G สำหรับการใช้งานและการสอนเชิงพาณิชย์
www.channelcoding.com

วัตถุประสงค์ของโครงการวิจัยคือการออกแบบและพัฒนาระบบเข้ารหัสสัญญาณมาตรฐาน 5G สำหรับการใช้งาน (สำหรับงานวิจัย)

SG Channel Encoder

SG LDPC Encoder - DVB-S2/S2-X

FPGA Software

RUN

Configuration Parameters


Q: Rate:

Qm: LDM:

PL: H2M:

Input

Block Diagram of SG Channel Encoder



Software (Python)

SG LDPC Encoder

1. LDPC CRC Encoding
2. LDPC Code Block Segmentation
3. LDPC Code Block CRC Encoding
4. LDPC LDPC Encoding
5. LDPC Rate Matching
6. LDPC Bit Interleaving
7. LDPC Code Block Concatenation

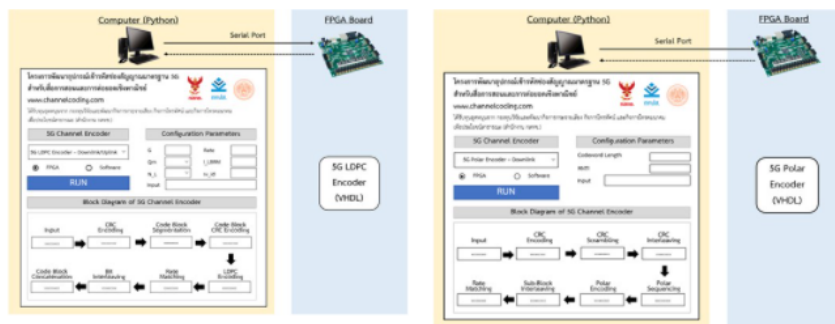
python

รูปที่ 1 GUI จากชุดซอฟต์แวร์ Python

GUI แสดงผลลัพธ์ของชุดอุปกรณ์

มีนาคม 14, 2021 - by sirawit

ในส่วนนี้ทีมวิจัยได้พัฒนาส่วนต่อประสานกับผู้ใช้ (GUI) เพื่อแสดง **ผลลัพธ์การเข้ารหัสช่องสัญญาณ จากอุปกรณ์ FPGA ที่พัฒนาด้วยภาษา VHDL** ทีมวิจัยได้ออกแบบหน้าต่าง GUI คล้ายคลึงกับการแสดงผลจากชุดซอฟต์แวร์ MATLAB ในกรณีที่ผู้ใช้เลือกการเข้ารหัสแบบ 5G LDPC Encoder - Downlink/Uplink แสดงได้ดังรูปที่ 1 (การทดสอบในส่วนนี้คือการทดสอบผลลัพธ์การเข้ารหัส LDPC ในช่องสัญญาณข้อมูล) บอร์ด FPGA จะใช้โมดูล 5G LDPC Encoder ที่เขียนด้วยภาษา VHDL ในการเข้ารหัสข้อมูลที่กำหนด หลังจากเข้ารหัสเสร็จสิ้น บอร์ด FPGA จะส่งผลลัพธ์ที่เกิดขึ้นในแต่ละขั้นตอนของกระบวนการเข้ารหัสผ่าน Serial Port กลับไปยังคอมพิวเตอร์ และแสดงผลลัพธ์แก่ผู้ใช้งานบน GUI ในทำนองเดียวกันผู้ใช้งานสามารถเลือกการเข้ารหัสแบบ 5G Polar Encoder เพื่อทดสอบผลลัพธ์การเข้ารหัส Polar ในรูปแบบ Downlink Uplink และส่วน Broadcast ดังแสดงดังแสดงในรูปที่ 2 3 และ 4 ตามลำดับ



รูปที่ 1 GUI จากชุดอุปกรณ์ FPGA ส่วน 5G LDPC Encoder - Downlink/Uplink

รูปที่ 2 GUI จากชุดอุปกรณ์ FPGA ส่วน 5G Polar

ในหน้าดังกล่าวยังแสดงวิดีโอสาธิตการใช้งาน GUI เบื้องต้นทีละขั้นตอน



นอกจากนี้ ผู้รับชมสามารถดาวน์โหลดชุดซอฟต์แวร์ GUI เพื่อแสดงผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ไปใช้งานได้โดยไม่มีค่าใช้จ่าย

[ดาวน์โหลด] GUI แสดงการเข้ารหัสช่องสัญญาณมาตรฐาน 5G

พัฒนาด้วยภาษา Python เพื่อแสดงผลสัฟท์แวร์กระบวนการเข้ารหัสข้อมูลด้วยรหัสช่องสัญญาณ สำหรับมาตรฐานการสื่อสารไร้สายยุค 5G



โหลด: รหัส GUI
[*Python code]

1.5 วิดีโอเผยแพร่ความรู้

ในเมนูนี้แสดงวิดีโอที่อธิบายถึงพื้นฐานรหัสช่องสัญญาณที่ครอบคลุมเนื้อหาสำคัญ เช่น ประเภทของรหัสช่องสัญญาณ กระบวนการเข้ารหัสช่องสัญญาณ และพารามิเตอร์สำคัญที่บ่งบอกถึงคุณลักษณะของรหัสช่องสัญญาณ เป็นต้น นอกจากนี้ ยังแสดงวิธีการใช้งานซอฟต์แวร์การเข้ารหัสช่องสัญญาณ LDPC และ Polar ที่พัฒนาขึ้นด้วยภาษา MATLAB และ Python รวมถึงการใช้งานวงจรเข้ารหัส LDPC และ Polar ที่พัฒนาขึ้นด้วยภาษา VHDL ตามลำดับ เมนูนี้เหมาะสำหรับใช้เพื่อประกอบการเรียนของนักศึกษา หรือผู้ที่สนใจ

1.6 เกี่ยวกับเรา

ในเมนูสุดท้ายนี้แสดงข้อความขอบคุณผู้ให้การสนับสนุนโครงการ ได้แก่ กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมเพื่อประโยชน์สาธารณะ

เกี่ยวกับเรา

เว็บไซต์นี้จัดทำขึ้นมาภายใต้ โครงการพัฒนาความพร้อมระดับประเทศของการสื่อสารไร้สายในยุค 5G

สนับสนุนโดย กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมเพื่อประโยชน์สาธารณะ



อีกทั้งยังแสดงข้อมูล และช่องทางในการติดต่อสอบถามทีมวิจัย เกี่ยวกับปัญหาการใช้งานชุดซอฟต์แวร์และอุปกรณ์รวมถึงปัญหาอื่น ๆ ได้

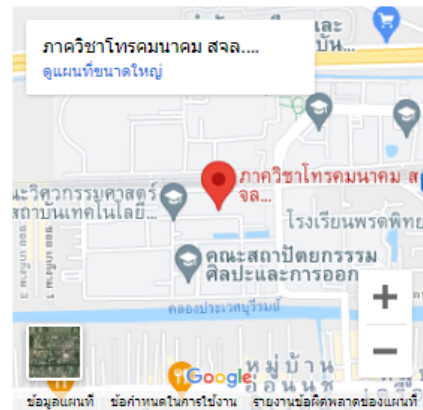
ติดต่อพวกเรา

Email: cssrg.ccr@gmail.com

Facebook: [5g_wireless_technology_Thailand](#)

YouTube: [5G Thailand](#)

Address : ห้อง T-113 ตึกภาควิชา วิศวกรรมโทรคมนาคม คณะ วิศวกรรมโทรคมนาคม สถาบัน เทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เลขที่ 1 ถนนฉลองกรุง 1 แขวงลาดกระบัง เขตลาดกระบัง กรุงเทพมหานคร 10520

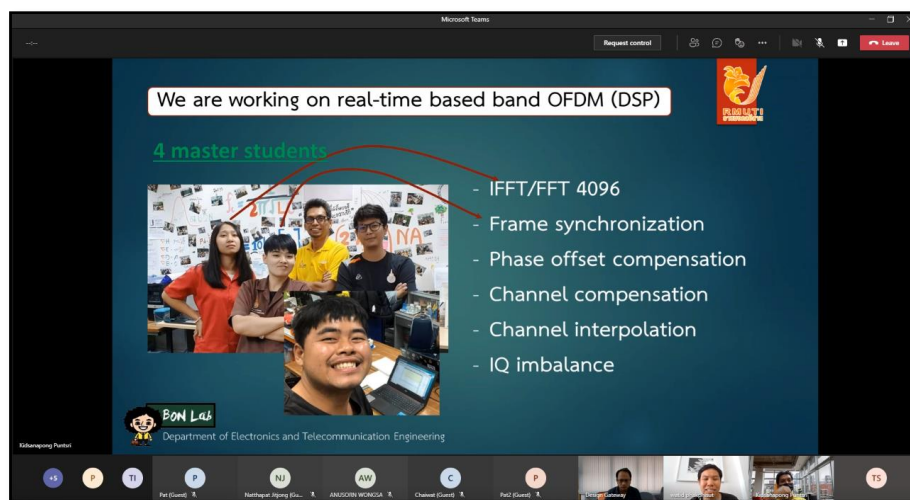


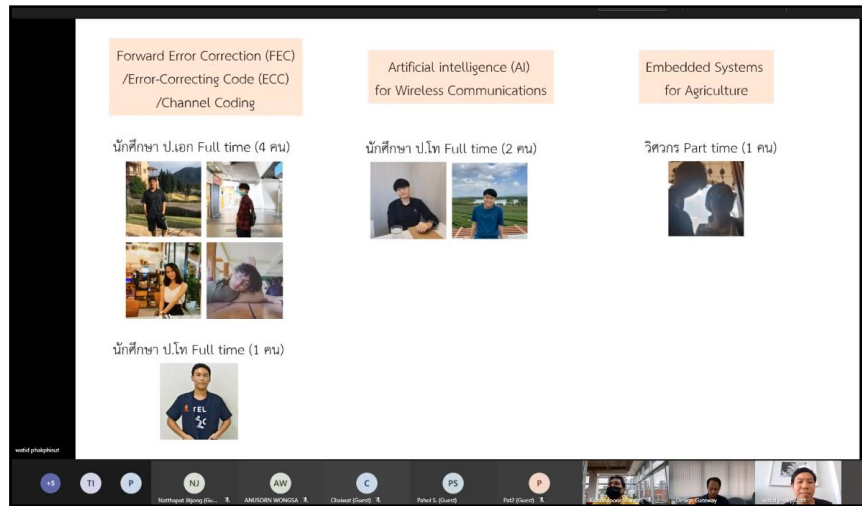
2. การนำเสนอวงจรเข้ารหัสช่องสัญญาณมาตรฐาน 5G บนบอร์ด FPGA แก่ภาคอุตสาหกรรม

ผลผลิตของโครงการพัฒนารหัส LDPC และ Polar สำหรับมาตรฐาน 5G บนบอร์ด FPGA ได้ถูกนำเสนอและแลกเปลี่ยนความรู้กับบริษัท Design Gateway ในวันที่ 29 กรกฎาคม 2564 ผ่านโปรแกรม Microsoft Team มีเนื้อหาโดยสรุปดังนี้

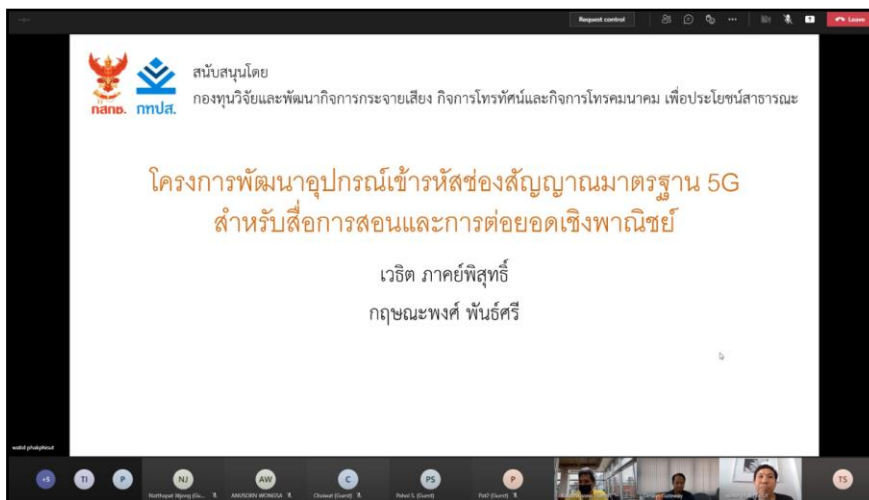


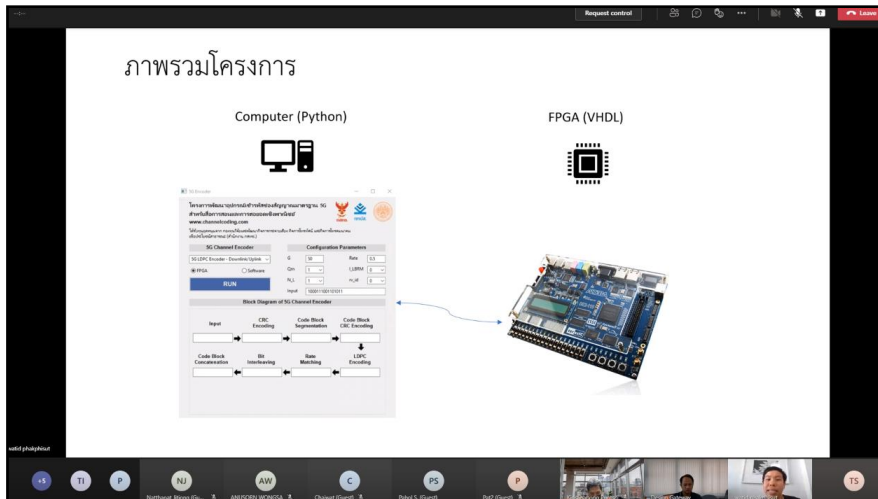
ในช่วงต้นการประชุมได้มีการแนะนำสมาชิกของทีมวิจัยที่มาจาก มหาวิทยาลัยราชภัฏวชิราขันธ์ วิทยาเขตขอนแก่น และสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ซึ่งทั้งสองทีมจะมีหน้าที่ในการพัฒนารหัสช่องสัญญาณ LDPC และ Polar บนบอร์ด FPGA รวมถึงการจัดทำเว็บไซต์ และวิดีโอเพื่อเป็นสื่อการเรียนการสอนของนักศึกษาทั้งสองสถาบัน



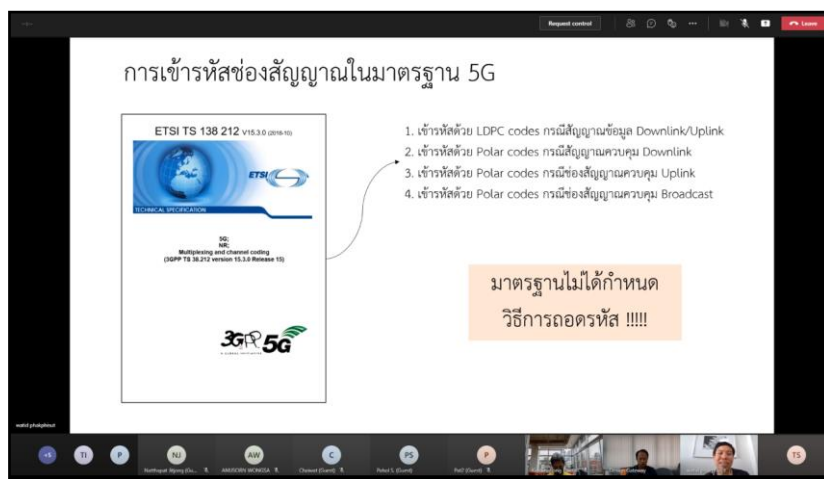


หลังจากนั้น อ.เวธิต ได้มีการบรรยายถึงหัวข้อ จุดประสงค์ และภาพรวมของโครงการพัฒนาอุปกรณ์เข้ารหัสช่องสัญญาณมาตรฐาน 5G สำหรับสื่อการสอนและการต่อยอดเชิงพาณิชย์ ที่ได้รับทุนสนับสนุนจากกองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ

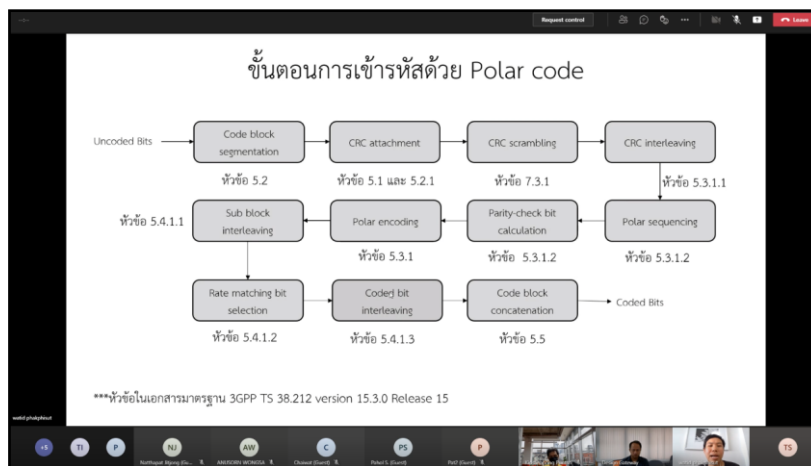
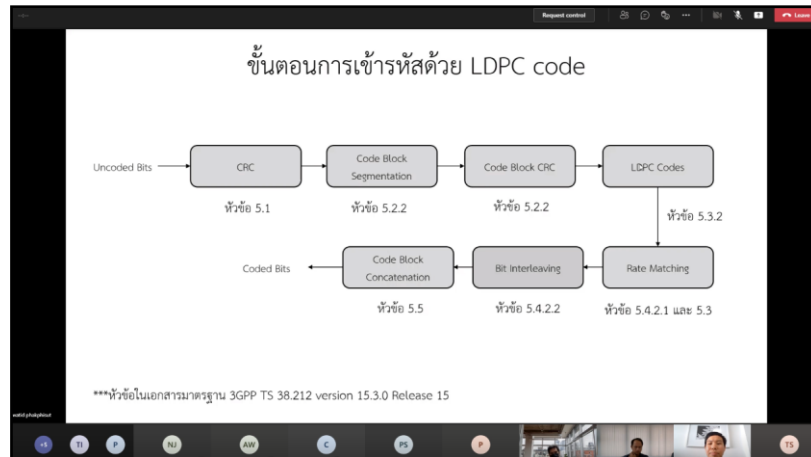




จากนั้น ได้มีการบรรยายถึงมาตรฐานการสื่อสารไร้สายยุค 5G และรหัสช่องสัญญาณที่ถูกนำมาใช้ในการแก้ไขความผิดพลาดในการส่งข้อมูลผู้ใช้ (Data channel) ซึ่งได้แก่รหัส LDPC และสัญญาณควบคุม (Control channel) ได้แก่ รหัส Polar ซึ่งโครงการนี้ ได้พัฒนากระบวนการเข้ารหัสทั้งสองตัวบนวงจร FPGA และแสดงผลผ่านส่วนต่อประสานผู้ใช้ (GUI)



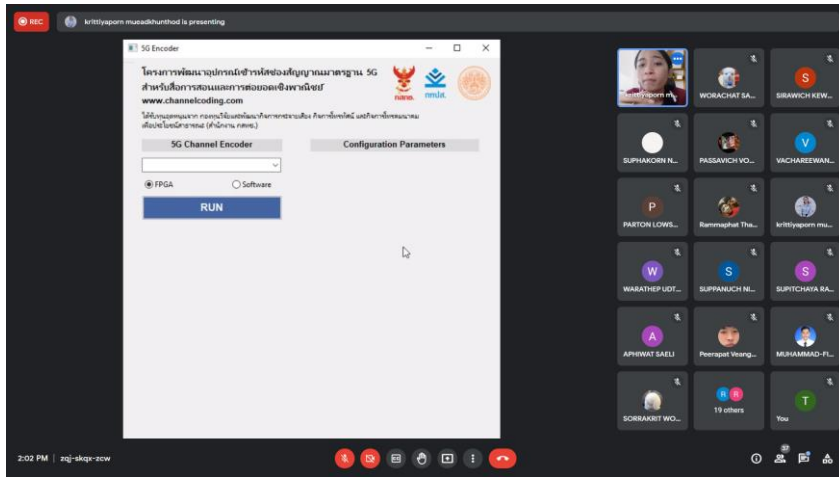
ในช่วงท้าย มีการอธิบายถึงบล็อกไดอะแกรมของกระบวนการเข้ารหัส LDPC และ Polar ซึ่งอ้างอิงมาตรฐาน 3GPP TS 38.212 version 15.3.0 Release 15 ซึ่งทีมวิจัยจะพัฒนาด้วยภาษา VHDL



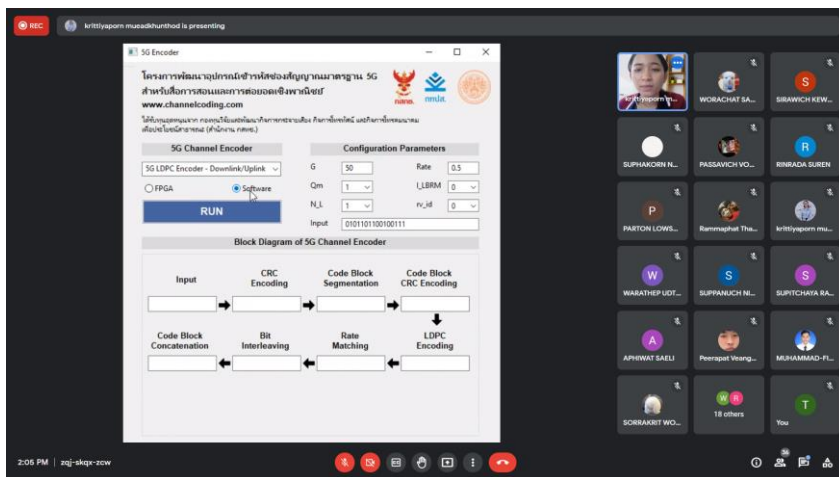
3. การนำเสนอชุดการเรียนรู้แก่มหาวิทยาลัย

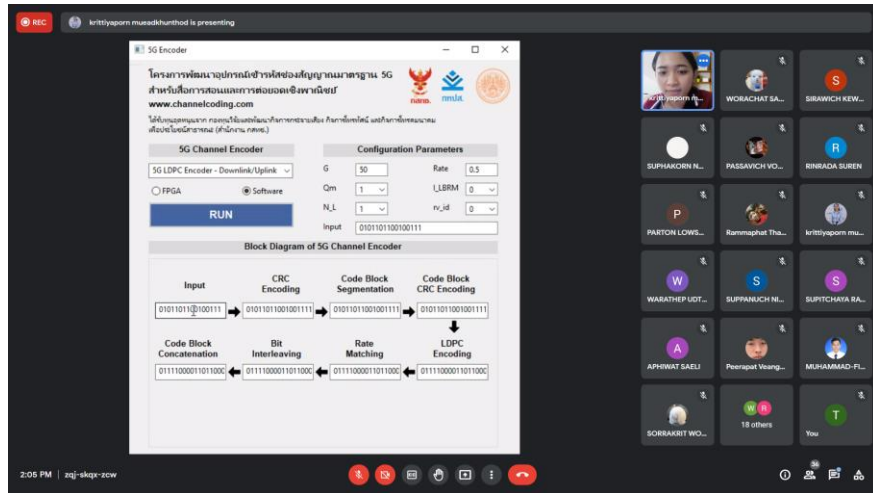
3.1 การใช้ชุดการเรียนรู้สำหรับการเรียนการสอนในหลักสูตรวิศวกรรมโทรคมนาคมครั้งที่ 1

การนำเสนอชุดการเรียนรู้สำหรับการเรียนรู้รหัส LDPC และ Polar สำหรับมาตรฐาน 5G บนบอร์ด FPGA ได้ถูกนำไปใช้สอนในห้องเรียนออนไลน์ในซึ่งมีนักศึกษาที่เข้าเรียนทั้งสิ้นจำนวน 100 คน ในวันที่ 15 พฤศจิกายน 2564 ช่วงเวลา 9.00 น. ถึง 12.00 น. และ 13.00 น. ถึง 15.00 น. ผ่านโปรแกรม Google meet มีเนื้อหาโดยสรุปดังนี้

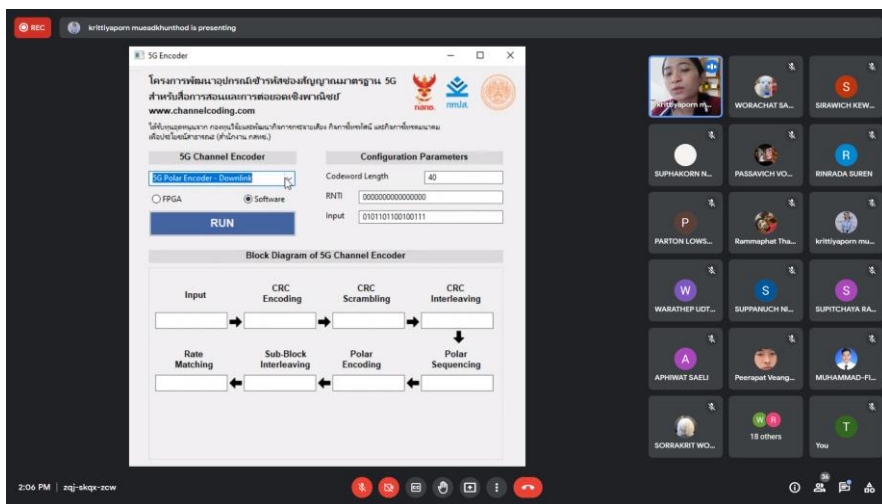


การใช้งานชุดการเรียนรู้การเข้ารหัสแก้ไขความผิดพลาดในมาตรฐาน 5G จะสามารถเลือกได้ว่าต้องการการเรียนรู้การเข้ารหัสใน FPGA หรือใน software ในรูปด้านล่างจะเป็นตัวอย่างการเลือกรู้รหัสแก้ไขความผิดพลาดใน Software หลังจากนั้นก็จะมีการเลือกรหัสแก้ไขความผิดพลาดที่ต้องการศึกษา ในตัวอย่างจะเลือกเป็นรหัส LDPC ที่ใช้ในการเข้ารหัสข้อมูลผู้ใช้ โดยจะถูกใช้ทั้งในช่องสัญญาณ Uplink และ Downlink สิ่งที่ใช้ต้องทำถัดมาคือกำหนดค่าพารามิเตอร์ต่าง ๆ ให้กับโปรแกรม เช่น G, Qm, N_L, Rate, L_BRM, rv_id และ input

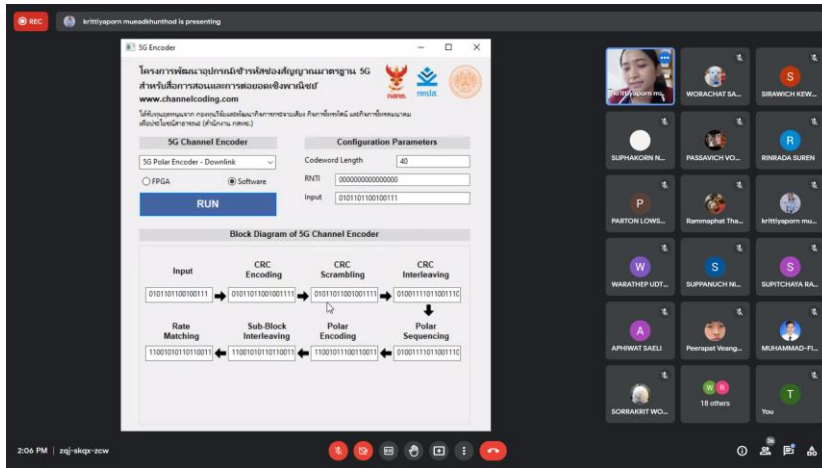




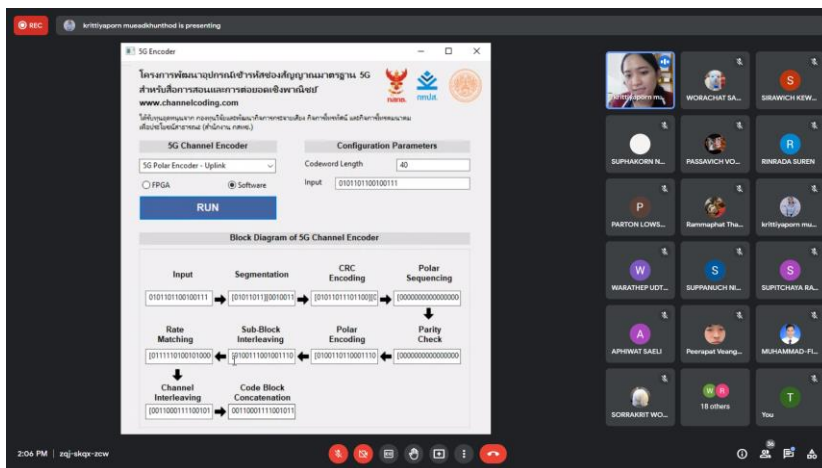
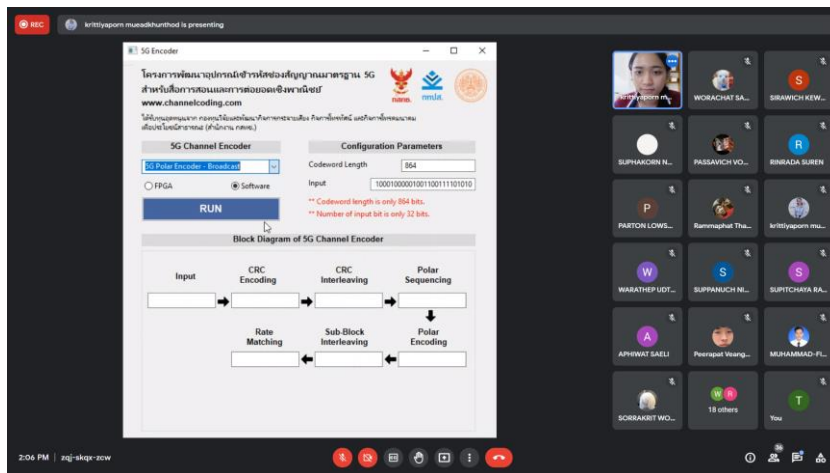
หลังจากเติมค่าพารามิเตอร์แล้วก็กดปุ่มรัน บิตข้อมูลที่ผ่านแต่ละกระบวนการจะแสดงในแต่ละช่องด้านล่าง ผู้ใช้สามารถปรับค่าพารามิเตอร์เพื่อดูการเปลี่ยนแปลงได้ตามต้องการ



ถัดมาเป็นตัวอย่างการเข้ารหัส Polar ในช่องสัญญาณควบคุม โดยเลือกใช้การเข้ารหัสใน Software ในตัวอย่างจะเลือกเป็นช่องสัญญาณ Downlink หลังจากนั้นก็เติมพารามิเตอร์ที่สำคัญ ได้แก่ Codeword length, RNTI และ input หลังจากกดรันข้อมูลที่ผ่านแต่ละกระบวนการจะแสดงในช่อง ผู้ใช้งานสามารถปรับค่าแล้วดูการเปลี่ยนแปลงของข้อมูลได้ตามต้องการ



นอกจากนี้ยังสามารถเลือกคืกขารหัส Polar สำหรับช่องสัญญาณ Broadcast และ ช่องสัญญาณ uplink ได้ดังรูปด้านล่าง



3.2 การอบรมมาตรฐาน 5G สำหรับการเรียนการสอนในหลักสูตรวิศวกรรมโทรคมนาคมครั้งที่ 2

การเปิดอบรมแบบออนไลน์ในหัวข้อการใช้งานชุดการเรียนรู้สำหรับการเรียนรู้รหัส LDPC และ Polar สำหรับมาตรฐาน 5G บนบอร์ด FPGA นักศึกษาที่เข้าร่วมทั้งในมหาวิทยาลัยพระจอมเกล้าพระนครเหนือและมหาวิทยาลัยราชภัฏวชิรเวศน์วิทยาเขตขอนแก่น ในวันที่ 15 พฤศจิกายน 2564 ช่วงเวลา 9.00 น. ถึง 12.00 น. ผ่านโปรแกรม Microsoft team มีเนื้อหาโดยสรุปดังนี้

Basic of LDPC code (1)

$R = K/N$

Low-density parity-check (LDPC) code

parity-check code

จำนวนบิตข้อมูลจำนวน 5 บิต $K=5$ → เพื่อการส่งข้อมูลด้วยอัตรา $R = 0.5$ → Parity bits (unknown bits) $N=10$ codeword

ในทางคณิตศาสตร์ Parity bits จะมีความสัมพันธ์กับใน codeword เรียกว่า check equation

ตัวอย่างเช่น

$$p_1 + p_2 + p_3 + p_4 = 0$$

$$p_1 + p_5 + m_1 + m_2 = 0$$

$$p_2 + p_5 + m_3 + m_4 = 0$$

$$p_3 + m_1 + m_3 + m_5 = 0$$

$$p_4 + m_2 + m_4 + m_5 = 0$$

ตรวจสอบ check equation จะทำให้ได้ parity bit

เราสามารถแปลงพจน์เหล่านี้ที่อยู่ในรูปของการเขียนเป็นเมทริกซ์ได้

| | | | | | | | | | | | |
|-------|---|---|---|---|---|---|---|---|---|---|---|
| p_1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| p_2 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| p_3 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | 0 |
| p_4 | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 1 |
| m_1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| m_2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| m_3 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| m_4 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| m_5 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 |

Parity check matrix (H)

การอบรมเริ่มต้นด้วยการอธิบายหลักการพื้นฐานของการเข้ารหัส LDPC และรายละเอียดของรหัส LDPC ที่ถูกนำไปใช้ในมาตรฐาน 5G นอกจากนี้ยังอธิบายถึงกระบวนการต่าง ๆ ในการเข้ารหัสเพื่อให้ผู้ใช้งานมีความเข้าใจเพื่อก่อนที่จะใช้งานชุดการเรียนรู้สำหรับการเรียนรู้รหัส LDPC สำหรับมาตรฐาน 5G

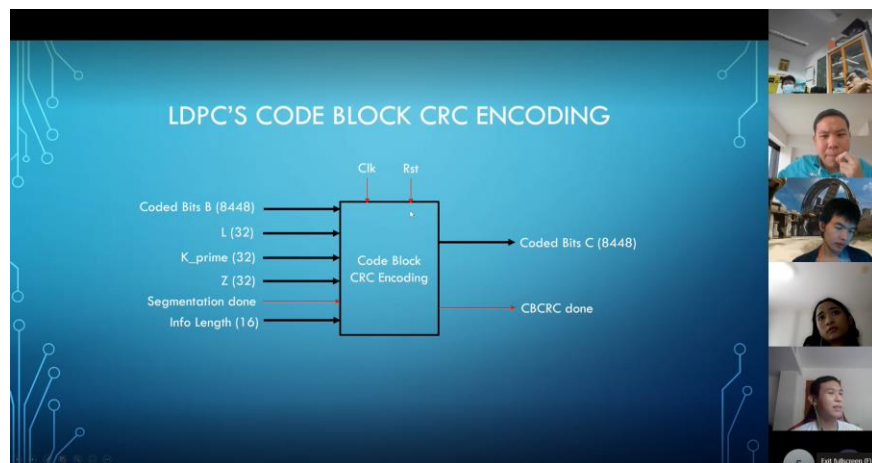
การเข้ารหัสโพลาร์

- ทำงานที่ทุกช่องสัญญาณ

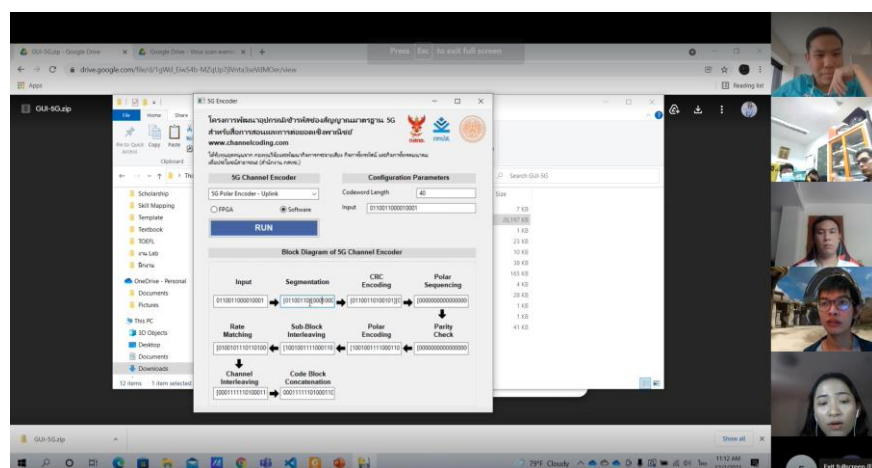
- โครงสร้างการเข้ารหัสโพลาร์ คล้ายโครงสร้าง butterfly ในการทำ fast Fourier transform (FFT)

เข้ารหัสโพลาร์

ถัดมาเป็นการอธิบายพื้นฐานของรหัส Polar และรหัส Polar ในมาตรฐาน 5G เพื่อให้มีความรู้ความเข้าใจในการเข้ารหัส Polar ก่อนที่จะใช้งานชุดการเรียนรู้รหัส Polar สำหรับมาตรฐาน 5G



นอกจากนี้ยังมีการอธิบายการอิมพลีเมนต์การเข้ารหัสแก้ไขความผิดพลาดทั้งรหัส LDPC และรหัส Polar ลงในบอร์ด FPGA



สุดท้ายเป็นการสาธิตการใช้งานชุดการเรียนรู้รหัสแก้ไขความผิดพลาดในมาตรฐาน 5G โดยสามารถปรับค่าพารามิเตอร์ต่าง ๆ ตามที่ได้มีการปูพื้นฐานมาตั้งแต่เริ่มต้นการอบรม

ภาคผนวก จ

ข้อเสนอแนะแนวทางการนำชุดการเรียนรู้และอุปกรณ์เข้ารหัส ช่องสัญญาณไปต่อยอด

1. ข้อเสนอแนะแนวทางการนำชุดซอฟต์แวร์การเรียนรู้การเข้ารหัสช่องสัญญาณในมาตรฐาน 5G เป็นสื่อในการสอนด้านวิศวกรรมโทรคมนาคม

ในการเรียนการสอนด้านวิศวกรรมโดยเฉพาะวิศวกรรมโทรคมนาคมที่เกี่ยวข้องโดยตรงกับการสื่อสารไร้สายในยุคที่ 5 หรือ 5G ยังไม่มีชุดการเรียนรู้ที่เป็นแบบสำเร็จรูปสามารถใช้งานง่าย ดังนั้นชุดการเรียนรู้การเข้ารหัสช่องสัญญาณในมาตรฐาน 5G นี้จะช่วยให้ นักศึกษามีความรู้ความเข้าใจและเห็นภาพมากขึ้น นอกจากนี้ นักศึกษายังสามารถปรับค่าพารามิเตอร์ต่างๆ เพื่อดูการเปลี่ยนแปลงของบิตข้อมูลในกระบวนการต่าง ๆ ชุดการเรียนรู้นี้เหมาะสำหรับใช้ภายหลังการเรียนเนื้อหาเกี่ยวกับการเข้ารหัส LDPC หรือรหัส Polar เนื่องจากผู้เรียนจะมีความเข้าใจเกี่ยวกับพื้นฐานการตั้งค่าพารามิเตอร์แล้วนำไปประยุกต์เพื่อต่อยอดการศึกษาต่อไปได้

2. ข้อเสนอแนะแนวทางการนำชุดซอฟต์แวร์การเรียนรู้การเข้ารหัสช่องสัญญาณในมาตรฐาน 5G ไปต่อยอดเชิงพาณิชย์

สำหรับการนำชุดการเรียนรู้การเข้ารหัสช่องสัญญาณในมาตรฐาน 5G ไปต่อยอดในเชิงพาณิชย์ ในส่วนของ GUI อาจจะต้องมีการปรับปรุงให้ชุดการเรียนรู้มีให้รูปลักษณะดูทันสมัยเพื่อให้ความน่าใช้งานมากขึ้นเพื่อที่จะสามารถนำไปใช้ในเชิงพาณิชย์ได้ แต่สิ่งที่สำคัญในโครงการนี้ที่น่าจะนำไปต่อยอดในเชิงพาณิชย์ได้คือ FPGA เนื่องจากการอิมพลีเมนต์ระบบลงใน FPGA สามารถนำไปเป็นโปรโตไทป์ในการออกแบบชิปเพื่อผลิตต่อไปได้

ภาคผนวก ฉ
รายงานผลการดำเนินงานฉบับย่อสำหรับตีพิมพ์ในวารสารสำนักงาน
กสทช.

การพัฒนาอุปกรณ์เข้ารหัสช่องสัญญาณมาตรฐาน 5G สำหรับสื่อการสอนและการต่อยอดเชิงพาณิชย์

เวริต ภาคย์พิสุทธิ, ศิริวิชญ์ กิตติวิชญกุล, จตุพร ดั่งทอง, กฤติยาภรณ์ เหมือดขุนทด, อนุสรณ์ วงศ์ษา และ ธนัช ศรีสุภา

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทคัดย่อ

เทคโนโลยีการสื่อสารยุค 5G มีความสามารถในการส่งข้อมูลความเร็วสูง ความหน่วงต่ำ และรองรับการส่งข้อมูลปริมาณมากในพื้นที่จำกัด นอกจากนี้ เทคโนโลยี 5G เรียกร้องความน่าเชื่อถือในการส่งข้อมูลในระดับอัตราเฟรมผิดพลาดน้อยกว่า 10^{-5} ด้วยเหตุนี้ เทคโนโลยี 5G จึงใช้งานรหัสช่องสัญญาณที่มีประสิทธิภาพสูงเพื่อตอบสนองระดับความผิดพลาดดังกล่าว ในปัจจุบัน การศึกษาในระดับอุดมศึกษาในสาขาที่เกี่ยวข้องกับวิศวกรรมการสื่อสารยังมีข้อจำกัดในเรื่องทดสอบการเข้ารหัสช่องสัญญาณในระบบสื่อสารยุคสมัยใหม่ เนื่องจากอุปกรณ์ที่เกี่ยวข้องมีราคาแพง บทความนี้จึงนำเสนอการพัฒนาอุปกรณ์เข้ารหัสช่องสัญญาณมาตรฐาน 5G โดยมีเป้าหมายเพื่อใช้เป็นสื่อการสอนในด้านวิศวกรรมโทรคมนาคม ประกอบไปด้วยการพัฒนาชุดซอฟต์แวร์และชุดอุปกรณ์ลอจิกแบบโปรแกรมได้ (FPGA) ในการเข้ารหัสช่องสัญญาณมาตรฐาน 5G ซึ่งมีการออกแบบส่วนประสานกับผู้ใช้ (GUI) ให้มีความง่ายและเป็นมิตรต่อผู้ใช้งาน

คำสำคัญ: เทคโนโลยี 5G, รหัสช่องสัญญาณ, ส่วนประสานผู้ใช้, อุปกรณ์ลอจิกแบบโปรแกรมได้

Abstract

Fifth Generation (5G) communication technology allows for high-speed data transmission, low latency, and high-volume data transmission in restricted area. Moreover, it requires high-reliability transmission with a frame error rate (FER) of less than 10^{-5} . As a results, the channel coding with high efficiency has been used to achieve a target reliability. Nowadays, higher education in the field of communication engineering is limited in testing channel coding for the modern communication systems, due to the high cost of the associated equipment. In this article, we then develop equipment for channel -

encoding process in accordance with the 5G communication standard, with the intention of using it as a teaching tool in communication engineering. Our developed equipment includes a software package as well as hardware in the form of a field-programmable gate array (FPGA). In addition, we develop a user-friendly graphical user interface (GUI) to display the encoding results.

Keywords: 5G communication, Channel coding, GUI, FPGA

1. ที่มาและความสำคัญ

เทคโนโลยีการสื่อสารไร้สายยุค 5G เป็นวิวัฒนาการครั้งสำคัญของระบบสื่อสารและโทรคมนาคมของโลก เนื่องจากไม่ได้ถูกจำกัดแค่เพียงโทรศัพท์มือถือดังเช่นเทคโนโลยี 1G – 4G ที่ผ่านมา แต่จะรวมถึงการสื่อสารระหว่างอุปกรณ์ต่างๆ เช่น อุปกรณ์อินเทอร์เน็ตของสรรพสิ่ง (Internet of Things, IoT) [1] ซึ่งเป็นแรงผลักดันไปสู่มิติใหม่ของมนุษย์ในการดำรงชีวิต และการทำงาน โดยรูปแบบการใช้งาน (Use case) ที่จะเกิดขึ้นในยุค 5G เช่น การรับชมวิดีโอสตรีมมิ่งคมชัดสูง การใช้งานยานพาหนะไร้คนขับ การเชื่อมต่ออุปกรณ์ IoT หลายพันล้านตัวเข้ากับโครงสร้างพื้นฐาน เป็นต้น เนื่องด้วยปริมาณข้อมูลเพิ่มมากขึ้นจากการใช้งานอุปกรณ์ IoT ภายใต้อุปกรณ์ 5G ดังนั้น นอกเหนือจากการส่งข้อมูลด้วยความเร็วสูง และความหน่วงต่ำแล้ว ความน่าเชื่อถือในการส่งข้อมูลก็เป็นอีกหนึ่งปัจจัยที่จะต้องมีการพิจารณาด้วย สหภาพโทรคมนาคมระหว่างประเทศ (ITU) ได้กำหนดความต้องการด้านความน่าเชื่อถือในการสื่อสารยุค 5G ว่าต้องมีอัตราเฟรมผิดพลาดน้อยกว่า 10^{-5} [2] เทคโนโลยีของการสื่อสารยุค 5G ที่สามารถตอบสนองระดับความผิดพลาดดังกล่าวคือการใช้งานเทคโนโลยีรหัสช่องสัญญาณ (Channel coding) โดยการสื่อสารยุค 5G ใช้งานรหัสช่องสัญญาณในชนิด รหัสพรีดีเร็กความหนาแน่นต่ำ หรือรหัสแอลดีพีซี [3] และรหัสโพลาร์ [4] อย่างไรก็ตาม รหัส

ช่องสัญญาณทั้งสองชนิดมีความซับซ้อนในการคำนวณทางคณิตศาสตร์สูง ซึ่งอาจเป็นการยากในการทำความเข้าใจอย่างลึกซึ้งในช่วงเวลาสั้น ๆ เมื่อถูกนำไปบรรจุในบทเรียนของวิชาวิศวกรรมโทรคมนาคม

ทีมวิจัยได้สังเกตเห็นปัญหาสำคัญของระบบการศึกษาทั่วโลกที่เทคโนโลยีที่ใช้ในระบบสื่อสารปัจจุบัน เช่น เทคโนโลยีการเข้ารหัสช่องสัญญาณ มีความซับซ้อนมากกว่าเนื้อหาในวิชาเรียน ปัญหาดังกล่าวเกิดจากอุปกรณ์การสอนสมัยใหม่ที่มีราคาแพง และต้องนำเข้าจากต่างประเทศ ดังนั้น ทีมวิจัยจึงนำเสนอการพัฒนาชุดซอฟต์แวร์ และชุดอุปกรณ์ FPGA เพื่อเข้ารหัสช่องสัญญาณตามมาตรฐานการสื่อสารยุค 5G โดยได้พัฒนากระบวนการเข้ารหัสรหัสแวลด์พีซีเพื่อใช้ในช่องสัญญาณข้อมูล (Data channel) และรหัสโพลาร์เพื่อใช้ในช่องสัญญาณควบคุม (Control channel) นอกจากนี้ มีการออกแบบส่วนประสานกับผู้ใช้ (GUI) เพื่อแสดงผลลัพธ์ผลลัพธ์การเข้ารหัสช่องสัญญาณทั้งสองแก่ผู้ใช้งาน ชุดอุปกรณ์ FPGA สำหรับเข้ารหัสแวลด์พีซีและรหัสโพลาร์ และ GUI ถูกใช้เป็นที่ประกอบการสอนสำหรับนักศึกษาภาควิชาวิศวกรรมโทรคมนาคม

2. รหัสช่องสัญญาณสำหรับการสื่อสารยุค

5G

2.1 พื้นฐานรหัสช่องสัญญาณ

รหัสช่องสัญญาณเป็นเทคโนโลยีสำคัญที่ขาดไม่ได้ในระบบสื่อสารดิจิทัล เนื่องจากทำให้การรับส่งข้อมูลดิจิทัลปราศจากความผิดพลาด เทคโนโลยีการสื่อสารไร้สายยุค 4G ที่ผ่านมา ใช้เข้ารหัสช่องสัญญาณในชนิดรหัสเทอร์โบ (Turbo codes) [5] สำหรับช่องสัญญาณข้อมูล และรหัสคอนโวลูชัน (Convolutional codes) [6] สำหรับช่องสัญญาณควบคุม รหัสเทอร์โบเป็นรหัสช่องสัญญาณที่มีความน่าสนใจอย่างมาก เนื่องจากมีสมรรถนะเข้าใกล้ทฤษฎีความจุช่องสัญญาณ [7] ซึ่งเป็นเหตุผลที่ทำให้รหัสเทอร์โบถูกประยุกต์ใช้ในมาตรฐาน 4G จากการค้นพบรหัสเทอร์โบทำให้ให้นักวิจัยทำการศึกษารหัสช่องสัญญาณอื่น ๆ ที่มีสมรรถนะเทียบเท่าหรือดีกว่ารหัสเทอร์โบจนได้พบว่ารหัสแวลด์พีซี (LDPC codes) [3] มีสมรรถนะเข้าใกล้ทฤษฎีความจุช่องสัญญาณเช่นเดียวกับรหัสเทอร์โบ อีกทั้งยังมีความซับซ้อนในการเข้ารหัสและถอดรหัสต่ำกว่า ปัจจุบันรหัสแวลด์

พีซีถูกนำมาใช้งานในระบบสื่อสารมาตรฐานต่าง ๆ อย่างกว้างขวาง เช่น ระบบเครือข่ายคอมพิวเตอร์ไร้สายมาตรฐาน (IEEE 802.11) [8] ระบบโทรทัศน์ดิจิทัลมาตรฐาน (DVB-S2 และ DVB-T2) [9] นอกจากนี้ รหัสโพลาร์ (Polar codes) [4] ซึ่งเป็นรหัสช่องสัญญาณชนิดใหม่ที่เพิ่งถูกคิดค้นไม่นานมานี้ ก็ได้รับการพิสูจน์ว่ามีสมรรถนะเข้าใกล้ทฤษฎีความจุช่องสัญญาณ เช่นเดียวกับรหัสเทอร์โบ และรหัสแวลด์พีซี

2.2 ชื่อเรียกหรือรหัสช่องสัญญาณสำหรับมาตรฐาน 5G

สหภาพโทรคมนาคมระหว่างประเทศ (ITU) ได้แบ่งรูปแบบการใช้งานเครือข่ายโทรศัพท์เคลื่อนที่ยุค 5G ไว้ 3 ประเภท ได้แก่ 1) enhanced mobile broadband (eMBB) ซึ่งมุ่งเน้นการใช้งานในลักษณะที่ต้องการส่งข้อมูลความเร็วสูง เช่น การรับชมวิดีโอความละเอียด 4K หรือ การสื่อสารเรียลไทม์ 2) ultra-reliable and low latency communications (URLLC) ซึ่งมุ่งเน้นส่งข้อมูลที่มีความเสถียรมาก และมีความหน่วงต่ำ เช่น การรับส่งข้อมูลระหว่างรถขับเคลื่อนอัตโนมัติที่มีการหน่วงเวลาต่ำเพื่อตอบสนองเหตุการณ์ต่าง ๆ และ 3) massive machine type communications (mMTC) ซึ่งมุ่งเน้นเชื่อมต่ออุปกรณ์จำนวนมากในบริเวณพื้นที่จำกัด เช่น การรับส่งข้อมูลของอุปกรณ์ IoT

ดังนั้น การออกแบบมาตรฐานของรหัสช่องสัญญาณสำหรับใช้งานในการสื่อสารยุค 5G ต้องมีการกำหนดชื่อเรียกหรือพื้นฐานเพื่อรองรับรูปแบบการใช้งานทั้งสามด้านดังนี้

- รหัสช่องสัญญาณต้องมีสมรรถนะในการแก้ไขบิดเบือนผิดพลาดที่เชื่อมโยง เพื่อตอบสนองระดับอัตราเฟรมผิดพลาดที่น้อยกว่า 10^{-6} ดังนั้น รหัสช่องสัญญาณที่ใช้ต้องมีสมรรถนะเข้าใกล้ทฤษฎีความจุช่องสัญญาณ
- รหัสช่องสัญญาณต้องมีความซับซ้อนฮาร์ดแวร์ที่ยอมรับได้ในทางปฏิบัติ เนื่องจากความซับซ้อนฮาร์ดแวร์ส่งผลกระทบต่อระยะเวลาใช้งานแบตเตอรี่ในอุปกรณ์ IoT
- รหัสช่องสัญญาณต้องมีวงจรถอดรหัสที่สามารถทำงานแบบขนาน (Parallel processing) โดยจะส่งผลให้ความหน่วงต่ำลง ซึ่งช่วยให้การสื่อสารเป็นไปอย่างรวดเร็ว
- รหัสช่องสัญญาณต้องรองรับการทำงาน IR-HARQ (Hybrid ARQ with Incremental Redundancy) [10] ใน

ที่นี้ เมื่อภาครับรู้ว่าข้อมูลที่ได้รับมีความผิดพลาดเกิดขึ้น ภาครับจะทำการส่งการแจ้งเตือนกลับไปภาคส่ง โดยภาคส่งจะมีการเข้ารหัสข้อมูลเดิมด้วยอัตรารหัสที่ต่ำลง เพื่อเพิ่มจำนวนบิตพาริตีให้มากขึ้น และทำการส่งเฉพาะบิตพาริตีอันใหม่มาที่ภาครับเท่านั้น เพื่อลดความซ้ำซ้อนในการส่งข้อมูลเดิมซ้ำ

- รหัสช่องสัญญาณต้องมีกระบวนการเข้ารหัสที่สามารถรองรับความยาวคำรหัสใด ๆ อีกทั้งอัตรารหัสต้องมีความยืดหยุ่นโดยปรับเปลี่ยนได้ตามสถานการณ์ของความผิดพลาดที่เกิดขึ้น

2.3 บทสรุปการเลือกใช้รหัสช่องสัญญาณสำหรับมาตรฐาน 5G

องค์กร 3GPP ได้จัดตั้งการประชุมมีจุดประสงค์เพื่อออกแบบมาตรฐานรหัสช่องสัญญาณสำหรับการสื่อสารยุค 5G โดยเริ่มศึกษาอัตราหัสช่องสัญญาณสำหรับมาตรฐาน 5G ในการประชุม RAN WG1 Meeting #84bis และเสร็จสิ้นในการประชุม RAN WG1 Meeting #91 ได้ข้อสรุปว่าจะมีการนำรหัสแวลดีพีซีมาใช้สำหรับช่องสัญญาณข้อมูล เนื่องจากรหัสแวลดีพีซีมีประสิทธิภาพของปริมาณการส่งผ่านข้อมูล (Throughput) ต่อพื้นที่ และปริมาณการส่งผ่านข้อมูลสูงสุดที่สามารถใช้งานสูงกว่ารหัสเทอร์โบ อีกทั้งรหัสแวลดีพีซียังสามารถลดความซับซ้อนและความหน่วงเวลาจากการประมวลผลแบบขนานในการถอดรหัส นอกจากนี้ จะมีการใช้งานรหัสโพลาร์ในช่องสัญญาณควบคุม เนื่องจากรหัสดังกล่าวสามารถใช้งานร่วมการตรวจสอบความผิดพลาดแบบ CRC (Cyclic Redundancy Check) และมีการถอดรหัสแบบ Successive Cancellation List (SCL) [11] ซึ่ง มีประสิทธิภาพในการถอดรหัสดีกว่ารหัสเทอร์โบโดยเฉพาะเมื่อใช้สำหรับการส่งข้อมูลที่มีความยาวไม่เกิน 250 บิต ซึ่งเป็นขนาดข้อมูลที่เพียงพอสำหรับการส่งข้อมูลในช่องสัญญาณควบคุม

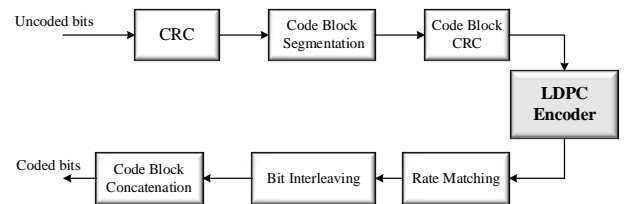
3. รายละเอียดการเข้ารหัสช่องสัญญาณ

มาตรฐาน 5G

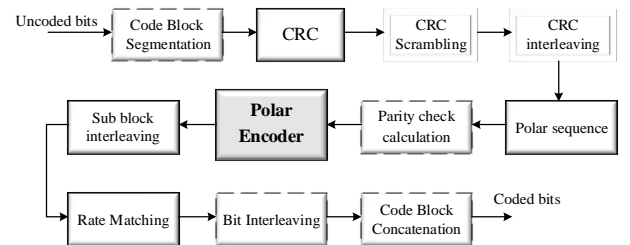
3.1 วงจรเข้ารหัสแวลดีพีซี

ทีมวิจัยได้พัฒนาการเข้ารหัสแวลดีพีซีมาตรฐาน 5G ตามเอกสาร 3GPP 38.212 [12] รูปที่ 1 แสดง

บล็อกไดอะแกรมของกระบวนการเข้ารหัสช่องสัญญาณในช่องสัญญาณข้อมูล เริ่มต้นจากการนำข้อมูลไปเข้ารหัสตรวจสอบความผิดพลาด CRC ซึ่งนิยมใช้ในการรับส่งข้อมูลที่มีการเข้าจังหวะ (Synchronous) และการสื่อสารที่ต้องการความรวดเร็วในการตรวจสอบความผิดพลาดข้อมูล ผลลัพธ์ถูกนำเข้าสู่กระบวนการแบ่งส่วนข้อมูล (Segmentation) เพื่อแบ่งข้อมูลให้มีขนาดเป็นไปตามมาตรฐาน 5G จากนั้นบล็อกข้อมูลจะถูกนำไปเข้ารหัส CRC ซ้ำอีกครั้ง แล้วนำผลลัพธ์ที่ได้ไปเข้ารหัสแวลดีพีซีด้วยการเติมบิตพาริตีเพื่อให้สามารถแก้ไขความผิดพลาดที่ภาครับได้อย่างถูกต้อง ผลลัพธ์ที่ได้จากการเข้ารหัสแวลดีพีซีเรียกว่าคำรหัส (Codeword) โดยคำรหัสจะถูกส่งต่อไปยังวงจรปรับอัตรารหัส (Rate matching) เพื่อปรับความยาวให้เหมาะสมกับการส่งตามมาตรฐาน 5G จากนั้น วงจรแทรกสลับบิต (Bit interleaving) ถูกใช้เพื่อแทรกสลับบิตข้อมูลในคำรหัสเพื่อป้องกันความผิดพลาดแบบติดกัน (Burst error) และจะมีการต่อเรียงบล็อกรหัส (Concatenation) เพื่อส่งต่อไปมอดูเลตและส่งข้อมูลออกไป



รูปที่ 1 กระบวนการเข้ารหัสช่องสัญญาณในช่องสัญญาณข้อมูล



รูปที่ 2 กระบวนการเข้ารหัสช่องสัญญาณในช่องสัญญาณควบคุม

3.2 วงจรเข้ารหัสโพลาร์

ในส่วนนี้ ทีมวิจัยพัฒนาการเข้ารหัสโพลาร์มาตรฐาน 5G ตามเอกสาร 3GPP 38.212 [12] ดังที่ทราบกันว่ารหัสโพลาร์ถูกใช้งานในส่วนช่องสัญญาณควบคุม อย่างไรก็ตาม การใช้งานรหัสโพลาร์สำหรับมาตรฐาน 5G จะแตกต่างจาก

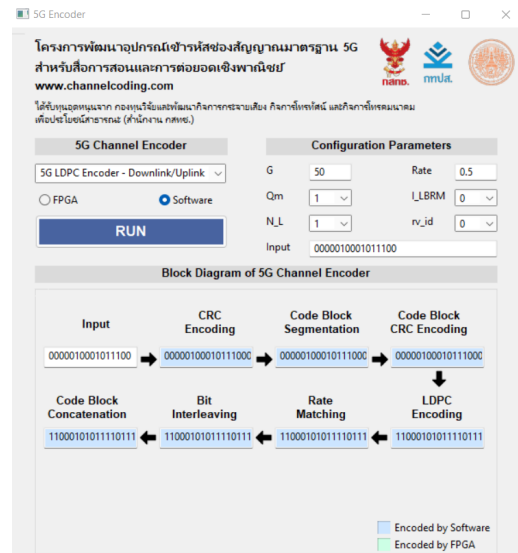
รหัสแอสกีที่ชี้ในแง่ที่ว่ากระบวนการทำงานจะแตกต่างกัน สำหรับการส่งในรูปแบบ ดาวน์ลิงก์ (Downlink) อัปลิงก์ (Uplink) และบรอดแคสต์ (Broadcast) รูปที่ 2 แสดงบล็อกไดอะแกรมของกระบวนการเข้ารหัสช่องสัญญาณในช่องสัญญาณควบคุม หมายความว่าเส้นที่บอกใช้งานในทุกช่องสัญญาณ เส้นประจะถูกใช้งานในเฉพาะช่องสัญญาณอัปลิงก์ และเส้นขาวถูกใช้งานในเฉพาะช่องสัญญาณดาวน์ลิงก์ แต่บล็อก CRC interleaving ทำงานในช่องสัญญาณบรอดแคสต์ด้วย

กระบวนการเข้ารหัสช่องสัญญาณในช่องสัญญาณควบคุมสามารถแบ่งออกเป็น 11 กระบวนการย่อย ดังนี้ 1) กระบวนการแบ่งส่วนข้อมูล ซึ่งใช้สำหรับลดความซับซ้อนของกระบวนการเข้ารหัสเมื่อบิตข้อมูลมีความยาวมาก 2) กระบวนการเข้ารหัส CRC 3) กระบวนการสแครมเบิล CRC (CRC Scrambling) ซึ่งใช้ในการตรวจหาแบบอำพราง (Blind detection) เพื่อระบุตัวตนของอุปกรณ์ผู้ใช้งาน 4) กระบวนการแทรกสลับ CRC (CRC Interleaving) ซึ่งใช้สำหรับลดความซับซ้อนของการถอดรหัสในขั้นตอนการทำการเลิกก่อนกำหนด (Early termination) 5) กระบวนการลำดับช่องสัญญาณย่อย (Polar sequence) ซึ่งเป็นการค้นหาตำแหน่งบิตแช่แข็ง (Frozen bits) และเรียงบิตข้อมูลที่ได้รับมาจากกระบวนการก่อนหน้านี้ โดยตำแหน่งของบิตแช่แข็งจะถูกเลือกให้เหมาะสมเพื่อสมรรถนะการแก้ไขความผิดพลาดที่ดีที่สุด 6) กระบวนการคำนวณบิตพาริตี (Parity check calculation) ซึ่งเป็นการเติมบิตพาริตีในกรณีที่มีบิตข้อมูลที่มีความยาวคี่ 7) กระบวนการเข้ารหัสโพลาร์ (Polar encoder) 8) กระบวนการแทรกสลับบล็อกย่อย (Sub block interleaving) ซึ่งจะแบ่งคำรหัสออกเป็น 32 ส่วน และแทรกสลับตามมาตรฐาน 5G เพื่อให้มั่นใจว่าการปรับอัตรารหัสในกระบวนการถัดไปจะทำการตัดบิตในตำแหน่งตรงกับบิตแช่แข็ง 9) กระบวนการปรับอัตรารหัส ซึ่งจะทำการปรับความยาวคำรหัสตามความยาวที่ต้องการ 10) กระบวนการแทรกสลับบิตรหัส และ 11) กระบวนการเรียงการต่อบล็อกรหัส

4. การพัฒนาชุดซอฟต์แวร์เข้ารหัสช่องสัญญาณมาตรฐาน 5G

4.1 ส่วนประกอบชุดซอฟต์แวร์เข้ารหัสช่องสัญญาณ

จากภาพรวมการพัฒนารหัสช่องสัญญาณมาตรฐาน 5G ที่ได้กล่าวถึงในหัวข้อที่แล้ว ทีมวิจัยได้พัฒนาชุดซอฟต์แวร์เพื่อเข้ารหัสช่องสัญญาณมาตรฐาน 5G ทีมวิจัยใช้ภาษา MATLAB และ Python ในการพัฒนาชุดซอฟต์แวร์สำหรับเข้ารหัสช่องสัญญาณประกอบด้วย 4 ส่วน ได้แก่



รูปที่ 3 GUI แสดงผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G โดยใช้ชุดซอฟต์แวร์ กรณียเลือกการเข้ารหัสแบบ (5G LDPC Encoder – Downlink/Uplink)

- (5G LDPC Encoder – Downlink/Uplink) เป็นตัวเข้ารหัสแอสกีที่ใช้งานในช่องสัญญาณข้อมูล ในรูปแบบดาวน์ลิงก์ และอัปลิงก์
- (5G Polar Encoder – Downlink) เป็นตัวเข้ารหัสโพลาร์ที่ใช้งานในช่องสัญญาณควบคุม ในรูปแบบดาวน์ลิงก์
- (5G Polar Encoder – Uplink) เป็นตัวเข้ารหัสโพลาร์ที่ใช้งานในช่องสัญญาณควบคุม ในรูปแบบอัปลิงก์
- (5G Polar Encoder – Broadcast) เป็นตัวเข้ารหัสโพลาร์ที่ใช้งานในช่องสัญญาณควบคุม ในรูปแบบบรอดแคสต์

การพัฒนาชุดซอฟต์แวร์เข้ารหัสแอสกีที่มาตรฐาน 5G ครอบคลุมการทำงานทั้งหมด 7 ส่วน ตามที่ได้อธิบายในหัวข้อ 3.1 และการพัฒนาชุดซอฟต์แวร์เข้ารหัสโพลาร์ ครอบคลุมการทำงานทั้งหมด 11 ส่วน ตามที่ได้อธิบายในหัวข้อ 3.2

4.2 ส่วนประสานผู้ใช้

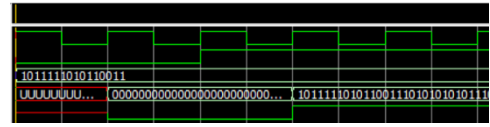
ทีมวิจัยออกแบบ GUI เพื่อแสดงผลการทำงานของรหัสช่องสัญญาณจากชุดซอฟต์แวร์ โดย GUI ถูกออกแบบให้ผู้ใช้สามารถใช้งานได้โดยง่ายโดยซึ่งพิจารณาได้จากรูปที่ 3 ผู้ใช้สามารถเลือกรูปแบบการเข้ารหัสที่ต้องการในหัวข้อ 5G Channel Encoder (กรอบเขียว) และป้อนค่าพารามิเตอร์ที่จำเป็นในหัวข้อ Configuration parameters (กรอบน้ำเงิน) กรณีเลือกการเข้ารหัสแบบ (5G LDPC Encoder – Downlink/Uplink) ข้อมูลอินพุตที่ป้อนลงไปจะถูกนำไปเข้าสู่กระบวนการทำงานของโมดูลต่างๆ ที่ได้กล่าวถึงในหัวข้อ 3.1 ผลลัพธ์ที่ได้ของแต่ละโมดูลจะแสดงแก่ผู้ใช้ในหัวข้อ Block Diagram of 5G Channel Encoder (กรอบแดง) นอกจากนี้ผู้ใช้งานยังสามารถเลือกรูปแบบการเข้ารหัส 5G Polar Encoder ในรูปแบบ คานาลิงก์ อับลิงก์ และบรอดแคสต์ ได้เช่นกัน ซึ่งข้อมูลอินพุตจะถูกนำไปเข้าสู่กระบวนการทำงานของโมดูลต่างๆ ที่ได้กล่าวถึงในหัวข้อ 3.2 จากนั้น จะแสดงผลแก่ผู้ใช้งานบน GUI

5. การพัฒนาชุดอุปกรณ์เข้ารหัส

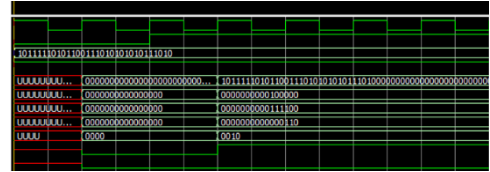
ช่องสัญญาณมาตรฐาน 5G

5.1 ส่วนประกอบวงจรเข้ารหัสช่องสัญญาณ

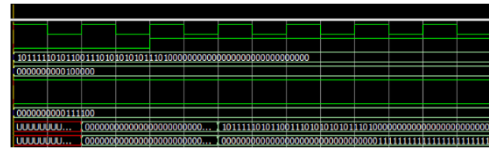
ทีมวิจัยได้พัฒนาชุดอุปกรณ์เข้ารหัสช่องสัญญาณบน FPGA ตามมาตรฐาน 5G ซึ่งประกอบด้วย 4 ส่วนเช่นเดียวกับการพัฒนาชุดซอฟต์แวร์การเข้ารหัสช่องสัญญาณ ได้แก่ (5G LDPC Encoder – Downlink/Uplink) (5G Polar Encoder – Downlink) (5G Polar Encoder – Uplink) และ (5G Polar Encoder – Broadcast) ทีมวิจัยเลือกใช้บอร์ด FPGA รุ่น Altera DE2-115 ซึ่งประกอบไปด้วยหลายอินเทอร์เฟซเพื่อรองรับการใช้งานได้อย่างหลากหลาย รายละเอียดคุณสมบัติและส่วนประกอบบน FPGA รุ่น Altera DE2-115 สามารถดูได้จากเอกสารข้อมูล [13] การพัฒนางจรเข้ารหัสแอลดีพีมาตรฐาน 5G ครอบคลุมการทำงานตามที่ได้อธิบายในหัวข้อ 3.1 และการพัฒนางจรเข้ารหัสโพลาไรมาตรฐาน 5G ครอบคลุมการทำงานตามที่ได้อธิบายในหัวข้อ 3.2



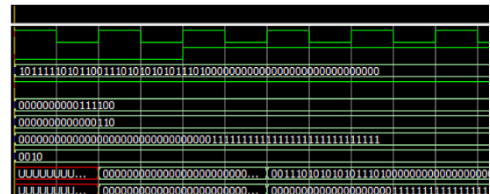
(ก)



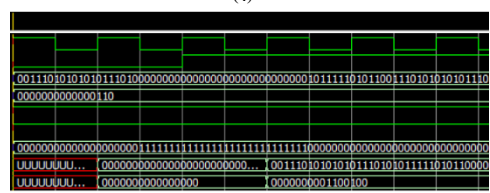
(ข)



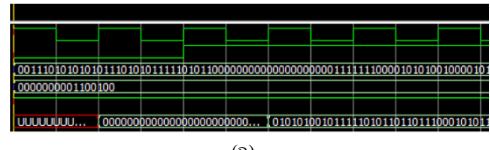
(ค)



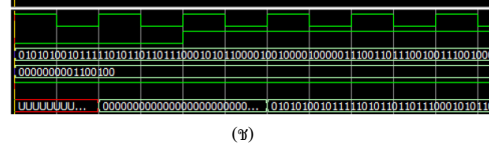
(ง)



(จ)



(ฉ)

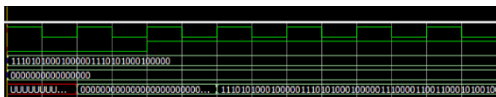


(ช)

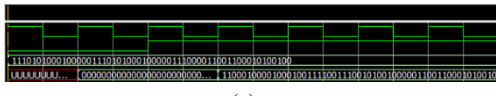
รูปที่ 4 ผลลัพธ์การเข้ารหัสในช่องสัญญาณข้อมูลเมื่อผ่าน (ก) วงจรเข้ารหัส CRC (ข) วงจรแบ่งย่อยบล็อกข้อมูล (ค) วงจรเข้ารหัส CRC ในครั้งที่สอง (ง) วงจรเข้ารหัสแอลดีพีซี (จ) วงจรปรับอัตราหัส (ฉ) วงจรแทรกสลับบิต และ (ช) วงจรต่อเรียงบล็อกรหัส

5.2 ผลการทดสอบวงจรเข้ารหัส

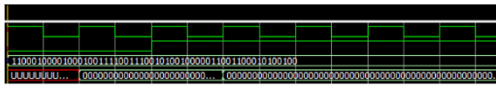
ผลการทดสอบใช้งานวงจรเข้ารหัสแอลดีพีซีและโพลาร์ด้วยวงจร FPGA ถูกแสดงบนโปรแกรม Quartus Prime Lite Edition และโปรแกรม ModelSim-Altera ในลำดับแรกจะพิจารณาการเข้ารหัสแอลดีพีซีสำหรับช่องสัญญาณข้อมูลซึ่งประกอบด้วยบล็อกย่อยดังแสดงในรูปที่ 1 ดังนั้น ส่วนนี้จะทำการแสดงผลพัลส์ของแต่ละบล็อกย่อยเมื่อกำหนดค่าได้ทำการป้อนข้อมูลอินพุตจำนวน 16 บิตคือ 101111010110011 ซึ่งผลลัพธ์ของแต่ละบล็อกย่อยแสดงในรูปที่ 4 (ก) – (ข)



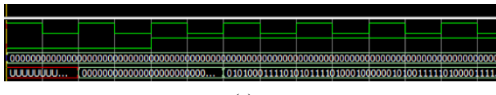
(ก)



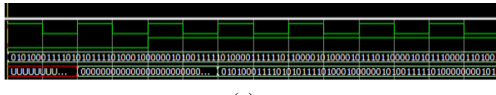
(ข)



(ค)



(ง)



(จ)

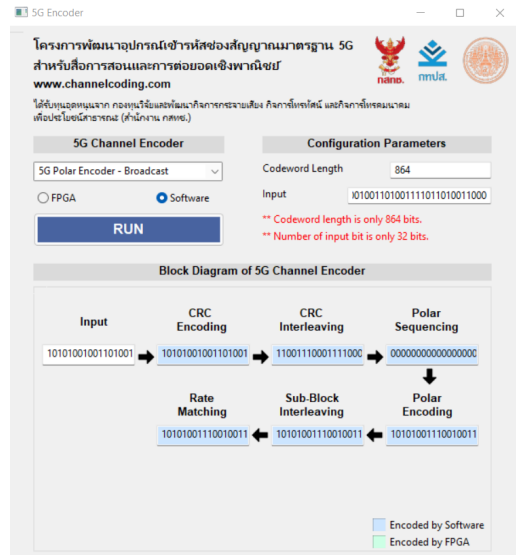


(ฉ)

รูปที่ 5 ผลลัพธ์การเข้ารหัสในช่องสัญญาณควบคุม (บรอดแคสต์) เมื่อผ่าน (ก) วงจรเข้ารหัส CRC (ข) วงจรการแทรกสลับ CRC (ค) วงจรลำดับช่องสัญญาณย่อย (ง) วงจรเข้ารหัสโพลาร์ (จ) แทรกสลับบล็อกย่อย และ (ฉ) วงจรปรับอัตรารหัส

ลำดับถัดไปจะพิจารณาการเข้ารหัสโพลาร์สำหรับช่องสัญญาณควบคุม อย่างไรก็ตาม การเข้ารหัสโพลาร์สำหรับช่องสัญญาณควบคุมสามารถแบ่งออกเป็น 3 รูปแบบคือ ดาวนลิงก์ อัปลิงก์ และบรอดแคสต์ ในส่วนนี้จะแสดงผลเฉพาะบรอดแคสต์เท่านั้น เนื่องด้วยกระบวนการเข้ารหัสโพลาร์สำหรับช่องสัญญาณควบคุมประกอบไปด้วย

บล็อกย่อยที่แสดงในรูปที่ 2 ในส่วนนี้ ทำการแสดงผลพัลส์การทำงานของแต่ละบล็อกย่อยเมื่อกำหนดค่ามีการป้อนข้อมูลอินพุตจำนวน 32 บิตคือ 11101010001000001110101000100000 ซึ่งผลลัพธ์ของแต่ละบล็อกย่อยแสดงในรูปที่ 5 (ก) – (ฉ)



รูปที่ 6 GUI แสดงผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G โดยใช้วงจรเข้ารหัส FPGA กรณีเลือกการเข้ารหัสแบบ (5G Polar Encoder – Broadcast)

5.3 ส่วนประสานผู้ใช้

ทีมวิจัยได้พัฒนา GUI เพื่อแสดงผลพัลส์การเข้ารหัสช่องสัญญาณจากวงจรเข้ารหัสบน FPGA โดยทีมวิจัยได้ออกแบบหน้าต่าง GUI ให้คล้ายคลึงกับการแสดงผลพัลส์จากชุดซอฟต์แวร์ กรณีที่ผู้ใช้เลือกการเข้ารหัสแบบ (5G Polar Encoder – Broadcast) บอร์ด FPGA จะใช้โมดูล 5G LDPC Encoder เพื่อทำการเข้ารหัสข้อมูลที่กำหนด ตัวอย่างผลลัพธ์การเข้ารหัสแต่ละบล็อกในส่วน (5G Polar Encoder – Broadcast) ถูกกล่าวถึงในหัวข้อ 5.2 โดย GUI จะแสดงผลพัลส์การเข้ารหัสท้ายสุดผ่านโปรโตคอล universal asynchronous receiver-transmitter (UART) และแสดงแก่ผู้ใช้งานแสดงดังรูปที่ 6 นอกจากนี้ ผู้ใช้งานสามารถเลือกการเข้ารหัสรูปแบบอื่น ๆ เช่น (5G LDPC Encoder – Downlink/Uplink) (5G Polar Encoder – Downlink) และ (5G Polar Encoder – Uplink) ได้เช่นกัน

6. สรุปผลการพัฒนา

ในบทความนี้ ทีมวิจัยได้พัฒนาชุดซอฟต์แวร์และอุปกรณ์ FPGA สำหรับเข้ารหัสแอลดีพีซีและรหัสโพลาร์สำหรับมาตรฐาน 5G นอกจากนี้ ได้มีการออกแบบ GUI เพื่อแสดงผลการทำงานของรหัสในรูปแบบต่าง ๆ เช่น (5G LDPC Encoder – Downlink/Uplink) (5G Polar Encoder – Downlink) (5G Polar Encoder – Uplink) และ (5G Polar Encoder – Broadcast) แก่ผู้ใช้งาน ชุดอุปกรณ์ FPGA สำหรับเข้ารหัสแอลดีพีซีและรหัสโพลาร์ถูกใช้เป็นการสอนสำหรับนักศึกษาภาควิชาวิศวกรรมโทรคมนาคม

เอกสารอ้างอิง

- [1] L. Chettri and R. Bera, "A Comprehensive Survey on Internet of Things (IoT) Toward 5G Wireless Systems," in *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 16-32, Jan. 2020.
- [2] M. Bennis, M. Debbah and H. V. Poor, "Ultrareliable and Low-Latency Wireless Communication: Tail, Risk, and Scale," in *Proceedings of the IEEE*, vol. 106, no. 10, pp. 1834-1853, Oct. 2018.
- [3] R. Gallager, "Low-density parity-check codes," in *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21-28, January 1962.
- [4] E. Arıkan, "Channel Polarization: A Method for Constructing Capacity-Achieving Codes for Symmetric Binary-Input Memoryless Channels," in *IEEE Transactions on Information Theory*, vol. 55, no. 7, pp. 3051-3073, July 2009.
- [5] C. Berrou, A. Glavieux and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," *Proceedings of ICC '93 - IEEE International Conference on Communications*, 1993, pp. 1064-1070 vol.2.
- [6] P. Elias. Coding for noisy channels. *IRE Convention Records*, 3(4):37-46, 1955.
- [7] C. E. Shannon, "A mathematical theory of communication", *Bell Syst. Tech. J.*, vol. 27, pp. 379-423, Oct. 1948.
- [8] IEEE P802.11n/TM-2009, "IEEE Standard for Information Technology Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," 2001.
- [9] Digital Video Broadcasting (DVB). Second generation framing structure, channel coding and modulation systems for broadcasting, interactive services, news gathering and other broadband satellite applications (DVB-S2). *European Standard (Telecommunications series) ETSI EN 302 307 V1.2.1 (2009-08)*, 2009.
- [10] H. Saber and I. Marsland, "An Incremental Redundancy Hybrid ARQ Scheme via Puncturing and Extending of Polar Codes," in *IEEE Transactions on Communications*, vol. 63, no. 11, pp. 3964-3973, Nov. 2015.
- [11] Tal and A. Vardy, "List decoding of polar codes," 2011 *IEEE International Symposium on Information Theory Proceedings*, 2011, pp. 1-5.
- [12] Specification # 38.212 - 3GPP [Online], Source: <https://portal.3gpp.org/desktopmodules/Specifications/SpecificationDetails.aspx?specificationId=3214>.
- [13] Altera DE2-115 User Manual – Intel [Online], Source: https://www.intel.com/content/dam/altera-www/global/en_US/portal/dsn/42/doc-us-dsnbk-42-1404062209-de2-115-user-manual.pdf.



กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ
(สำนักงาน กสทช.)