รายงานฉบับสมบูรณ์

โครงการขอรับการส่งเสริมและสนับสนุนจากเงินกองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ

โครงการพัฒนาอุปกรณ์ถอดรหัสช่องสัญญาณมาตรฐาน 5G สำหรับสื่อการสอนและการ์ดเร่งความเร็วสำหรับโครงข่าย O-RAN

Development of the channel decoding in 5G standard for education tool and accelerator card for O-RAN

เวธิต ภาคย์พิสุทธิ์

กรกฎาคม 2567

กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ
(สำนักงาน กสทช.)

รายงานฉบับสมบูรณ์

ทุนส่งเสริมและสนับสนุนการวิจัยและพัฒนา
สัญญารับทุนเลขที่ D66-1-(2)-001

โครงการพัฒนาอุปกรณ์ถอดรหัสช่องสัญญาณมาตรฐาน 5G สำหรับสื่อการสอนและการ์ดเร่งความเร็วสำหรับโครงข่าย O-RAN

Development of the channel decoding in 5G standard for education tool and accelerator card for O-RAN

นักวิจัย

1. เวธิต ภาคย์พิสุทธิ์ หัวหน้าโครงการ
2. กฤษณะพงศ์ พันธ์ศรี นักวิจัยร่วม

ได้รับทุนอุดหนุนจาก
กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ
(สำนักงาน กสทช.)

กรกฎาคม 2567

บทสรุปผู้บริหาร

โครงการพัฒนาอุปกรณ์ถอดรหัสช่องสัญญาณมาตรฐาน 5G สำหรับสื่อการสอนและการ์ดเร่งความเร็ว
สำหรับโครงข่าย O-RAN
กรกฎาคม 2567

โครงการนี้ นักวิจัยมีเป้าหมาย 2 มิติ ได้แก่ มิติด้านการศึกษาวิศวกรรมโทรคมนาคม และมิติของการพัฒนาอุปกรณ์โทรคมนาคม โดยมิติด้านการศึกษานั้น ทีมวิจัยนี้ได้เล็งเห็นปัญหาสำคัญของระบบการศึกษาทั่วโลก ที่เทคโนโลยีการสื่อสารปัจจุบันมีความซับซ้อนมากกว่าเนื้อหาในวิชาเรียน ดังนั้น ทีมวิจัยจึงต้องการพัฒนาอุปกรณ์ที่ทันสมัยสามารถสนับสนุนการเรียนรู้ของนักศึกษาสาขาวิชาวิศวกรรมโทรคมนาคมหรือสาขาอื่น ๆ ที่เกี่ยวข้อง โดยอุปกรณ์ประกอบด้วยซอฟต์แวร์และฮาร์ดแวร์ ซึ่งซอฟต์แวร์สามารถแสดงให้เห็นถึงขั้นตอนและผลลัพธ์ของแต่ละขั้นตอนของระบบสื่อสารไร้สายมาตรฐาน 5G เพื่อให้นักศึกษามีความรู้ความเข้าใจ อีกทั้ง ซอฟต์แวร์ดังกล่าวสามารถเชื่อมต่อกับฮาร์ดแวร์ที่พัฒนาบนอุปกรณ์ FPGA เพื่อให้สามารถนำไปใช้ต่อยอดเชิงพาณิชย์ได้

นักวิจัยได้ดำเนินการจัดทำ “โครงการพัฒนาอุปกรณ์เข้ารหัสช่องสัญญาณมาตรฐาน 5G สำหรับสื่อการสอนและการต่อยอดเชิงพาณิชย์” โดยได้รับทุนสนับสนุนจากกองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมเพื่อประโยชน์สาธารณะ (กทปส.) ปี พ.ศ. 2563 อย่างไรก็ตาม โครงการก่อนหน้าจะมุ่งเน้นเฉพาะการพัฒนาซอฟต์แวร์และฮาร์ดแวร์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G เท่านั้น มิได้พัฒนาการถอดรหัสช่องสัญญาณ เนื่องจากองค์กร 3GPP ไม่ได้มีการกำหนดว่าการถอดรหัสจะต้องมีขั้นตอนอย่างไร

การพัฒนาการถอดรหัสช่องสัญญาณนั้น จะต้องใช้องค์ความรู้เฉพาะของผู้พัฒนา จึงเป็นที่มาของข้อเสนอโครงการนี้ ส่วนที่ 1 จะมุ่งเน้นการพัฒนาซอฟต์แวร์และฮาร์ดแวร์การถอดรหัสช่องสัญญาณมาตรฐาน 5G เมื่อพัฒนาส่วนดังกล่าวเสร็จสิ้น ผู้วิจัยจะดำเนินการ ส่วนที่ 2 พัฒนาชุดการเรียนรู้การถอดรหัสช่องสัญญาณในมาตรฐาน 5G และ ส่วนที่ 3 พัฒนาชุดทดสอบสมรรถนะการเข้ารหัสและถอดรหัสช่องสัญญาณในระบบสื่อสารไร้สาย โดยงานส่วนที่ 1 และ 2 นี้ จะทำให้ได้ซอฟต์แวร์และฮาร์ดแวร์ของรหัสช่องสัญญาณที่สมบูรณ์ สามารถนำไปใช้งานเชิงพาณิชย์และใช้เป็นสื่อการสอนในวิชาการสื่อสารไร้สายได้ นอกจากนี้ ส่วนที่ 4 จะมุ่งเน้นการพัฒนาฮาร์ดแวร์การเข้ารหัสและถอดรหัสช่องสัญญาณในรูปแบบของการ์ดเร่งความเร็ว FEC ซึ่งองค์กร O-RAN Alliance ได้กำหนดให้การ์ดเร่งเป็นส่วนหนึ่งในมาตรฐาน O-RAN

โครงการพัฒนาอุปกรณ์ถอดรหัสช่องสัญญาณมาตรฐาน 5G สำหรับสื่อการสอนและการ์ดเร่งความเร็วสำหรับโครงข่าย O-RAN

เวธิต ภาคย์พิสุทธิ์

กรกฎาคม 2567

เทคโนโลยีการสื่อสารไร้สายยุค 5G มีความสามารถในการส่งข้อมูลความเร็วสูง ความหน่วงต่ำ และรองรับการส่งข้อมูลปริมาณมากในพื้นที่จำกัด นอกจากนี้ เทคโนโลยี 5G เรียกร้องความน่าเชื่อถือในการส่งข้อมูลในระดับอัตราเฟรมผิดพลาดน้อยกว่า 10^{-5} ด้วยเหตุนี้ เทคโนโลยี 5G จึงใช้งานรหัสช่องสัญญาณที่มีประสิทธิภาพสูงเพื่อตอบสนองระดับความผิดพลาดดังกล่าว ปัจจุบัน การศึกษาในระดับอุดมศึกษาที่เกี่ยวข้องกับวิศวกรรมการสื่อสารยังมีข้อจำกัด เรื่องการจัดการเรียนการสอนที่สอดคล้องกับเทคโนโลยีสมัยใหม่ เช่น เทคโนโลยีรหัสช่องสัญญาณมาตรฐาน 5G ทำให้โครงการวิจัยนี้จึงนำเสนอการพัฒนาอุปกรณ์เข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G โดยมีเป้าหมายเพื่อใช้เป็นสื่อการสอนในด้านวิศวกรรมโทรคมนาคม อีกทั้งสามารถนำไปต่อยอดเชิงพาณิชย์ได้ โครงการจะประกอบไปด้วยการพัฒนาชุดซอฟต์แวร์และชุดอุปกรณ์ FPGA เพื่อใช้เข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G ซึ่งมีการออกแบบส่วนประสานกับผู้ใช้ให้มีความง่ายและเป็นมิตรต่อผู้ใช้งาน

Development of the channel decoding in 5G standard for education tool and accelerator card for O-RAN

Watid Phakphisut

July 2024

5 G wireless communication technology has the capability of high-speed data transmission, low latency, and supporting large data transfers in confined areas. Moreover, 5 G technology demands high reliability in data transmission with a frame error rate of less than 10^{-5}. Therefore, 5G technology employs highly efficient channel coding to meet these error rate requirements. Currently, higher education studies related to communication engineering still have limitations in teaching methods that align with modern technology, such as 5 G standard channel coding technology. This research project thus proposes the development of encoding and decoding devices for 5 G standard channels with the goal of being used as teaching tools in telecommunications engineering and having commercial application potential. The project will include the development of software and FPGA hardware kits to encode and decode 5G standard channels, designed with user-friendly interfaces.

สารบัญ

	หน้า
บทสรุปผู้บริหาร	**ก**
บทคัดย่อภาษาไทย	**ข**
บทคัดย่อภาษาอังกฤษ	**ค**
สารบัญตาราง	**ซ**
สารบัญภาพ	**ฌ**
บทที่ 1 บทนำ	**1**
1.1 ที่มา และความสำคัญของโครงการ	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของโครงการ	2
1.4 ประโยชน์ที่คาดว่าจะได้รับ	5
บทที่ 2 ทฤษฎี และงานวิจัยที่เกี่ยวข้อง	**6**
2.1 พื้นฐานสัญญาณสุ่มและช่องสัญญาณสื่อสาร	6
2.2 พื้นฐานรหัสช่องสัญญาณ	11
2.3 การมอดูเลชันและการดีมอดูเลชันมาตรฐาน 5G	24
2.4 การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G	32
2.5 กระบวนทำงานฟิสิคัลเลเยอร์ลำดับสูง กรณี PDSCH/PUSCH/PSSCH	37
2.6 กระบวนทำงานฟิสิคัลเลเยอร์ลำดับสูง กรณี PUCCH/PDCCH/PBCH/PSCCH/PSFCH/ PSBCH	53
2.7 โครงข่าย O-RAN และการ์ดเร่งความเร็ว	72
2.8 มาตรฐาน O-RAN Working Group 6 เรื่อง Acceleration Abstraction Layer General Aspects and Principles และ FEC Profiles	76
บทที่ 3 ระเบียบวิธีวิจัย	**102**
3.1 การออกแบบชุดการเรียนรู้	102
3.2 การออกแบบชุดทดสอบสมรรถนะ	123
3.3 การออกแบบการเชื่อมต่อระหว่างอุปกรณ์ FPGA กับคอมพิวเตอร์	130
3.4 การออกแบบการ์ดเร่งความเร็ว FEC	133
3.5 การออกแบบการเชื่อมต่อระหว่างการ์ดเร่งความเร็ว FEC กับคอมพิวเตอร์	134
3.6 รายละเอียดการสร้างข้อมูลอินพุตแบบสุ่ม	140
3.7 รายละเอียดการสร้างช่องสัญญาณสื่อสารแบบ AWGN	142
3.8 รายละเอียดการสร้างช่องสัญญาณสื่อสารแบบ Fading	147
3.9 รายละเอียดการสร้างการมอดูเลชันมาตรฐาน 5G	148
3.10 รายละเอียดการสร้างการดีมอดูเลชันมาตรฐาน 5G	152

3.11 รายละเอียดการสร้างการเข้ารหัสช่องสัญญาณมาตรฐาน 5G 163
3.12 รายละเอียดการสร้างการถอดรหัสช่องสัญญาณมาตรฐาน 5G 165

บทที่ 4 ผลการวิจัย และการวิจารณ์ผล 168

4.1 ผลการทดสอบการทำงานของซอฟต์แวร์ GUI 168
4.2 ผลการทดสอบการทำงานของซอฟต์แวร์ Module 251
4.3 ผลการทดสอบการทำงานของอุปกรณ์ FPGA 274
4.4 ผลการทดสอบการเชื่อมต่อระหว่างอุปกรณ์ FPGA กับคอมพิวเตอร์ 295
4.5 ผลการทดสอบการทำงานของการ์ดเร่งความเร็ว FEC 297
4.6 ผลการทดสอบการเชื่อมต่อระหว่างการ์ดเร่งความเร็ว FEC กับคอมพิวเตอร์ 299
4.7 ผลการทดสอบเว็บไซต์เผยแพร่ชุดการเรียนรู้และทดสอบสมรรถนะระบบสื่อสารมาตรฐาน 5G 300
4.8 ผลการเผยแพร่ชุดการเรียนรู้และทดสอบสมรรถนะระบบสื่อสารมาตรฐาน 5G ในมหาวิทยาลัย 308
4.9 ผลการเผยแพร่ข้อมูลการออกแบบการ์ดเร่งความเร็วในโครงข่าย O-RAN ในบริษัทเอกชน 314

บทที่ 5 สรุปผลการวิจัย และข้อเสนอแนะ 315

บรรณานุกรม 316

ภาคผนวก 320

ภาคผนวก ก เอกสารประกอบการเผยแพร่ในมหาวิทยาลัย ภาคทฤษฎี 321
ภาคผนวก ข เอกสารประกอบการเผยแพร่ในมหาวิทยาลัย ภาคปฏิบัติ 356
ภาคผนวก ค เอกสารประกอบการเผยแพร่ในบริษัทเอกชน 378
ภาคผนวก ง ผลการดำเนินงานฉบับย่อสำหรับตีพิมพ์ในวารสารสำนักงาน กสทช. 384

สารบัญรูป

รูปที่ 1.1 ตัวอย่าง ชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G 3
รูปที่ 1.2 ตัวอย่าง ชุดทดสอบสมรรถนะการเข้ารหัสและถอดรหัสช่องสัญญาณในระบบสื่อสารไร้สาย4
รูปที่ 1.3 ตัวอย่าง การ์ดเร่งความเร็ว FEC สำหรับรหัสช่องสัญญาณในโครงข่าย O-RAN 4
รูปที่ 1.4 ช่องทางการ download ชุดการเรียนรู้และชุดทดสอบ ผ่านทางเว็บไซต์..... 4
รูปที่ 2.1 ความน่าจะเป็นของค่าสุ่ม 0 และ 1 เมื่อถูกสุ่มมาจำนวนหลายค่า 6
รูปที่ 2.2 แผนภาพการกระจายสองมิติ (2D scatter plot) ของผลลัพธ์คู่เลขสุ่ม (u_i, u_{i+1}) ลำดับติดต่อกันจาก (ก) วงจร LFSR (ข) วงจร CTG [1]..... 7
รูปที่ 2.3 ความน่าจะเป็นของค่าสุ่มของการกระจายแบบปกติ..... 8
รูปที่ 2.4 แบบจำลองช่องสัญญาณรบกวนเกาส์เซียนขาวแบบบวก 9
รูปที่ 2.5 ฟังก์ชันความหนาแน่นสเปกตรัมแบบเกาส์เซียน..... 10
รูปที่ 2.6 ตัวอย่างเมทริกซ์พาริตีเช็ก 12
รูปที่ 2.7 กราฟแทนเนอร์ของเมทริกซ์พาริตีเช็ก 13
รูปที่ 2.8 เมทริกซ์ฐาน 1..... 14
รูปที่ 2.9 เมทริกซ์ฐาน 2..... 15
รูปที่ 2.10 ช่องสัญญาณที่เกิดจากกระบวนการเข้ารหัสและถอดรหัสโพลาร์ 20
รูปที่ 2.11 บิตข้อมูลส่งผ่านช่องสัญญาณดิบ 21
รูปที่ 2.12 ช่องสัญญาณดิบจำนวน N ช่อง 22
รูปที่ 2.13 การรวมช่องสัญญาณ W เพื่อสร้างช่องสัญญาณ W_2 23
รูปที่ 2.14 การรวมช่องสัญญาณ W_2 เพื่อสร้างช่องสัญญาณ W_4 23
รูปที่ 2.15 แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ π/2-BPSK..... 24
รูปที่ 2.16 แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ BPSK 25
รูปที่ 2.17 แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ QPSK..... 26
รูปที่ 2.18 แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ 16QAM 27
รูปที่ 2.19 แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ 64QAM 28
รูปที่ 2.20 แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ 256QAM 29
รูปที่ 2.21 แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ 1024QAM..... 30
รูปที่ 2.22 การแสดงเมทริกซ์พาริตีเช็คในรูปแบบกราฟแทนเนอร์ 33
รูปที่ 2.23 (ก) การส่งผ่านค่า LLR เข้ามาในกราฟแทนเนอร์..... 34
รูปที่ 2.24 ตัวอย่างการคำนวณค่า LLR ที่โหนดเช็ค 35
รูปที่ 2.25 แผนภาพบล็อกการเข้ารหัส CRC 38
รูปที่ 2.26 แผนภาพบล็อกการเลือกกราฟฐาน 39
รูปที่ 2.27 แผนภาพบล็อกการกำหนดค่าความยาวบล็อกรหัส..... 40
รูปที่ 2.28 แผนภาพบล็อกการคำนวณจำนวนบล็อกย่อย 40
รูปที่ 2.29 แผนภาพบล็อกการกำหนดจำนวนบิตข้อมูลของแต่ละบล็อกรหัส..... 41

รูปที่ 2.30 แผนภาพบล็อกการคำนวณอัตราการขยายขนาดของเมทริกซ์ฐาน 42
รูปที่ 2.31 แผนภาพบล็อกการแบ่งย่อยบล็อกข้อมูล..... 43
รูปที่ 2.32 แผนภาพบล็อกการเข้ารหัส CRC ของบล็อกรหัส 44
รูปที่ 2.33 แผนภาพบล็อกการกำหนดความยาวคำรหัส 45
รูปที่ 2.34 แผนภาพบล็อกการสร้างเมทริกซ์พาริตีเช็ก $\mathbf{H}$ 45
รูปที่ 2.35 แผนภาพบล็อกการเติมบิตข้อมูลในบล็อกรหัส 46
รูปที่ 2.36 แผนภาพบล็อกการเข้ารหัส LDPC..... 47
รูปที่ 2.37 แผนภาพบล็อกการกำหนดความยาวคำรหัสของบล็อกรหัส 48
รูปที่ 2.38 แผนภาพบล็อกการคำนวณข้อมูลการส่งกลุ่มของบล็อกรหัส ($CBGTI$)..... 49
รูปที่ 2.39 แผนภาพบล็อกการคำนวณความยาวข้อมูลเอาต์พุตของกระบวนการปรับอัตรารหัส..... 50
รูปที่ 2.40 แผนภาพบล็อกการเลือกบิตเพื่อส่ง..... 51
รูปที่ 2.41 แผนภาพบล็อกการแทรกสลับบิต..... 52
รูปที่ 2.42 แผนภาพบล็อกการต่อเรียงบล็อกรหัส..... 53
รูปที่ 2.43 แผนผังการทำงานการแบ่งย่อยบล็อกรหัส..... 54
รูปที่ 2.44 แผนผังการทำงานการเชื่อมบิต CRC ท้ายข้อมูล..... 55
รูปที่ 2.45 แผนผังการทำงานการสแครมบลิง CRC 56
รูปที่ 2.46 แผนผังการทำงานการแทรกสลับ CRC..... 57
รูปที่ 2.47 แผนผังการทำงานการสร้างลำดับการแทรกสลับ CRC..... 58
รูปที่ 2.48 แผนผังการทำงานการเข้ารหัส CRC 59
รูปที่ 2.49 แผนผังการทำงานการหาความยาวคำรหัสแม่..... 61
รูปที่ 2.50 แผนผังการทำงานการลำดับช่องสัญญาณย่อย..... 63
รูปที่ 2.51 แผนผังการทำงานการลำดับช่องสัญญาณย่อย (ต่อ) 64
รูปที่ 2.52 แผนผังการทำงานการคำนวณบิตพาริตี..... 65
รูปที่ 2.53 แผนผังการทำงานการเข้ารหัสโพลาร์ 66
รูปที่ 2.54 แผนผังการทำงานการแทรกสลับบล็อกย่อย 67
รูปที่ 2.55 แผนผังการทำงานการสร้างลำดับการแทรกสลับบล็อกย่อย..... 68
รูปที่ 2.56 แผนผังการทำงานการปรับอัตรารหัส 69
รูปที่ 2.57 แผนผังการทำงานการแทรกสลับบิตรหัส..... 70
รูปที่ 2.58 แผนผังการทำงานการสร้างลำดับการแทรกสลับบิตรหัส 71
รูปที่ 2.59 แผนผังการทำงานการต่อบล็อกรหัส..... 72
รูปที่ 3.1 โครงร่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G..... 102
รูปที่ 3.2 ชุดการเรียนรู้การมอดูเลชันและการดีมอดูเลชันมาตรฐาน 5G เชื่อมต่อกับซอฟต์แวร์ 107
รูปที่ 3.3 ชุดการเรียนรู้การมอดูเลชันและการดีมอดูเลชันมาตรฐาน 5G เชื่อมต่อกับอุปกรณ์ FPGA 107
รูปที่ 3.4 โครงร่างชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G..... 108
รูปที่ 3.5 ชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G เชื่อมต่อกับซอฟต์แวร์ 114

รูปที่ 3.6 ชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G เชื่อมต่อกับอุปกรณ์ FPGA 114
รูปที่ 3.7 โครงร่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G 115
รูปที่ 3.8 ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G เชื่อมต่อกับซอฟต์แวร์ 122
รูปที่ 3.9 ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G เชื่อมต่อกับอุปกรณ์ FPGA 122
รูปที่ 3.10 โครงร่างชุดทดสอบสมรรถนะการมอดูเลชันและดีมอดูเลชันในระบบสื่อสารไร้สาย 124
รูปที่ 3.11 ชุดทดสอบสมรรถนะการมอดูเลชันและดีมอดูเลชันในระบบสื่อสารไร้สาย เชื่อมต่อกับซอฟต์แวร์ 124
รูปที่ 3.12 ชุดทดสอบสมรรถนะการเข้ารหัสและถอดรหัสช่องสัญญาณในระบบสื่อสารไร้สาย เชื่อมต่อกับอุปกรณ์ FPGA 125
รูปที่ 3.13 โครงร่างชุดทดสอบสมรรถนะการเข้ารหัสและถอดรหัสในระบบสื่อสารไร้สาย 125
รูปที่ 3.14 ชุดทดสอบสมรรถนะการเข้ารหัสและถอดรหัสช่องสัญญาณในระบบสื่อสารไร้สาย เชื่อมต่อกับซอฟต์แวร์ 127
รูปที่ 3.15 ชุดทดสอบสมรรถนะการเข้ารหัสและถอดรหัสช่องสัญญาณในระบบสื่อสารไร้สาย เชื่อมต่อกับอุปกรณ์ FPGA 127
รูปที่ 3.16 โครงร่างกระบวนการทำงานชั้นฟิสิคัลเลเยอร์ลำดับสูงในระบบสื่อสารไร้สาย 128
รูปที่ 3.17 ชุดทดสอบสมรรถนะกระบวนการทำงานชั้นฟิสิคัลเลเยอร์ลำดับสูงในระบบสื่อสารไร้สาย เชื่อมต่อกับซอฟต์แวร์ 129
รูปที่ 3.18 ชุดทดสอบสมรรถนะกระบวนการทำงานชั้นฟิสิคัลเลเยอร์ลำดับสูงในระบบสื่อสารไร้สาย เชื่อมต่อกับอุปกรณ์ FPGA 130
รูปที่ 3.19 ชุดบิตข้อมูลการสื่อสารภายใต้โพรโทคอล UART 130
รูปที่ 3.20 การใช้งานการ์ดเร่งความเร็ว FEC แบบ PDSCH_FEC 133
รูปที่ 3.21 การใช้งานการ์ดเร่งความเร็ว FEC แบบ PUSCH_FEC 134
รูปที่ 3.22 การ์ดเร่งความเร็ว FEC แบบ AAL_PDSCH_FEC ด้วยอุปกรณ์ FPGA 135
รูปที่ 3.23 การเชื่อมต่อระหว่างการ์ดเร่งความเร็วกับระบบปฏิบัติการ 135
รูปที่ 3.24 TB CRC attachment 136
รูปที่ 3.25 CB segmentation + CB CRC attachment 136
รูปที่ 3.26 LDPC encoding 136
รูปที่ 3.27 Rate Matching 137
รูปที่ 3.28 CB concatenation 137
รูปที่ 3.29 การ์ดเร่งความเร็ว FEC แบบ AAL_PUSCH_FEC ด้วยอุปกรณ์ FPGA 138
รูปที่ 3.30 CB deconcatenation 138
รูปที่ 3.31 Rate dematching 139
รูปที่ 3.32 LDPC decoding 139
รูปที่ 3.33 CB CRC + CB desegmentation 140

รูปที่ 3.34 TB CRC check 140
รูปที่ 3.35 แผนภาพวงจร CTG 3 ส่วนประกอบที่ใช้งานบนอุปกรณ์ FPGA..... 141
รูปที่ 3.36 แผนภาพการสร้างสัญญาณรบกวนเกาส์เซียนขาวแบบบวกบนอุปกรณ์ FPGA 142
รูปที่ 3.37 ค่า MSE ของการคำนวณฟังก์ชันทางคณิตศาสตร์ประเมินค่าด้วยโปรแกรม MATLAB (ก) คำนวณลอการิทึมธรรมชาติด้วยการประมาณค่าด้วยพหุนาม (ข) คำนวณรากที่สองโดยใช้อัลกอริทึม CORDIC 145
รูปที่ 3.38 วงจรคำนวณฟังก์ชันคณิตศาสตร์โดยอัลกอริทึม CORDIC 146
รูปที่ 3.39 แผนภาพการออกแบบสัญญาณการเฟดบนอุปกรณ์ FPGA..... 147
รูปที่ 3.40 ตัวอย่างการสร้าง QPSK ด้วย FPGA..... 149
รูปที่ 3.41 ตัวอย่างการสร้าง 16QAM ด้วย FPGA 150
รูปที่ 3.42 ตัวอย่างการหาค่าในตารางสำหรับค่า μ_0 และ μ_1 ที่มีการมอดูเลชันแบบ 8PAM 154
รูปที่ 3.43 อัลกอริทึมการดีมอดูเลชันสัญญาณตามมาตรฐาน 5G บนอุปกรณ์ FPGA โดยมีตัวอย่าง LUT0 และ LUT1 ประกอบ 155
รูปที่ 3.44 โครงสร้างตัวถอดรหัส LDPC..... 165
รูปที่ 3.45 ส่วนหน่วยประมวลผล 1 ตัว..... 167
รูปที่ 4.1 ผลลัพธ์การสร้างข้อมูลอินพุตแบบสุ่มโดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G..... 168
รูปที่ 4.2 ผลลัพธ์การสร้างข้อมูลอินพุตแบบสุ่มโดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 169
รูปที่ 4.3 ผลลัพธ์การสร้างข้อมูลอินพุตแบบสุ่มโดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G 169
รูปที่ 4.4 กราฟของสัญญาณข้อมูลอินพุตแบบสุ่มที่สร้างโดยใช้ซอฟต์แวร์..... 170
รูปที่ 4.5 ผลลัพธ์การสร้างข้อมูลอินพุตแบบสุ่มโดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G..... 170
รูปที่ 4.6 ผลลัพธ์การสร้างข้อมูลอินพุตแบบสุ่มโดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 171
รูปที่ 4.7 ผลลัพธ์การสร้างข้อมูลอินพุตแบบสุ่มโดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G..... 171
รูปที่ 4.8 กราฟของสัญญาณข้อมูลอินพุตแบบสุ่มที่สร้างโดยใช้อุปกรณ์ FPGA 172
รูปที่ 4.9 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PUSCH โดยใช้ซอฟต์แวร์ บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 173
รูปที่ 4.10 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PDSCH โดยใช้ซอฟต์แวร์ บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 173
รูปที่ 4.11 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PUCCH โดยใช้ซอฟต์แวร์ บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 174
รูปที่ 4.12 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PDCCH โดยใช้ซอฟต์แวร์ บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 174

รูปที่ 4.13 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PBCH โดยใช้ซอฟต์แวร์ บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 175
รูปที่ 4.14 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PUSCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 175
รูปที่ 4.15 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PDSCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 176
รูปที่ 4.16 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PUCCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 176
รูปที่ 4.17 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PDCCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 177
รูปที่ 4.18 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PBCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 177
รูปที่ 4.19 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PUSCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 178
รูปที่ 4.20 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PDSCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 179
รูปที่ 4.21 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PUCCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 179
รูปที่ 4.22 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PDCCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 180
รูปที่ 4.23 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PBCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 180
รูปที่ 4.24 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PUSCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 181
รูปที่ 4.25 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PDSCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 181
รูปที่ 4.26 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PUCCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 182
รูปที่ 4.27 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PUCCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 182
รูปที่ 4.28 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PBCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 183
รูปที่ 4.29 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PUSCH โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G 184
รูปที่ 4.30 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PDSCH โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G 184

รูปที่ 4.31 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PUCCH โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 184
รูปที่ 4.32 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PDCCH โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 185
รูปที่ 4.33 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PBCH โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 185
รูปที่ 4.34 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PUSCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 186
รูปที่ 4.35 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PDSCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 186
รูปที่ 4.36 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PUCCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 187
รูปที่ 4.37 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PDCCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 187
รูปที่ 4.38 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PBCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 188
รูปที่ 4.39 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PUSCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 189
รูปที่ 4.40 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PDSCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 189
รูปที่ 4.41 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PUCCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 189
รูปที่ 4.42 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PDCCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 190
รูปที่ 4.43 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PBCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 190
รูปที่ 4.44 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PUSCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 191
รูปที่ 4.45 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PDSCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 191
รูปที่ 4.46 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PUCCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 192
รูปที่ 4.47 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PDCCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 192
รูปที่ 4.48 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PBCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 193

รูปที่ 4.49 ผลลัพธ์การสแครมมาตรฐาน 5G โดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 194
รูปที่ 4.50 กราฟของสัญญาณสแครมที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 194
รูปที่ 4.51 ผลลัพธ์การสแครมมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 195
รูปที่ 4.52 กราฟของสัญญาณที่ได้จากการสแครมที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G..... 195
รูปที่ 4.53 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ π/2-BPSK โดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G..... 196
รูปที่ 4.54 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ BPSK โดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G..... 197
รูปที่ 4.55 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ QPSK โดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G..... 197
รูปที่ 4.56 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ 16QAM โดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G..... 198
รูปที่ 4.57 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ 64QAM โดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G..... 198
รูปที่ 4.58 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ 256QAM โดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G..... 199
รูปที่ 4.59 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ 1024QAM โดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G..... 199
รูปที่ 4.60 กราฟของสัญญาณมอดูเลชันแบบ π/2-BPSK ที่สร้างโดยใช้ซอฟต์แวร์ บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G..... 200
รูปที่ 4.61 กราฟของสัญญาณมอดูเลชันแบบ BPSK ที่สร้างโดยใช้ซอฟต์แวร์ บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G..... 200
รูปที่ 4.62 กราฟของสัญญาณมอดูเลชันแบบ QPSK ที่สร้างโดยใช้ซอฟต์แวร์ บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G..... 201
รูปที่ 4.63 กราฟของสัญญาณมอดูเลชันแบบ 16QAM ที่สร้างโดยใช้ซอฟต์แวร์ บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G..... 201
รูปที่ 4.64 กราฟของสัญญาณมอดูเลชันแบบ 64QAM ที่สร้างโดยใช้ซอฟต์แวร์ บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G..... 202
รูปที่ 4.65 กราฟของสัญญาณมอดูเลชันแบบ 256QAM ที่สร้างโดยใช้ซอฟต์แวร์ บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G..... 202
รูปที่ 4.66 กราฟของสัญญาณมอดูเลชันแบบ 1024QAM ที่สร้างโดยใช้ซอฟต์แวร์ บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G..... 203

รูปที่ 4.67 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ $\pi/2$-BPSK ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 203
รูปที่ 4.68 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ BPSK ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 204
รูปที่ 4.69 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ QPSK ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 204
รูปที่ 4.70 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ 16QAM ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 205
รูปที่ 4.71 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ 64QAM ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 205
รูปที่ 4.72 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ 256QAM ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 206
รูปที่ 4.73 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ 1024QAM ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 206
รูปที่ 4.74 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ $\pi/2$-BPSK โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 207
รูปที่ 4.75 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ BPSK โดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 208
รูปที่ 4.76 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ QPSK โดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 208
รูปที่ 4.77 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ 16QAM โดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 209
รูปที่ 4.78 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ 64QAM โดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 209
รูปที่ 4.79 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ 256QAM โดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 210
รูปที่ 4.80 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ 1024QAM โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 210
รูปที่ 4.81 กราฟของสัญญาณมอดูเลชันแบบ $\pi/2$-BPSK ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 211
รูปที่ 4.82 กราฟของสัญญาณมอดูเลชันแบบ BPSK ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 211
รูปที่ 4.83 กราฟของสัญญาณมอดูเลชันแบบ QPSK ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 212
รูปที่ 4.84 กราฟของสัญญาณมอดูเลชันแบบ 16QAM ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 212

รูปที่ 4.85 กราฟของสัญญาณมอดูเลชันแบบ 64QAM ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 213
รูปที่ 4.86 กราฟของสัญญาณมอดูเลชันแบบ 256QAM ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 213
รูปที่ 4.87 กราฟของสัญญาณมอดูเลชันแบบ 1024QAM ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 214
รูปที่ 4.88 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ $\pi/2$-BPSK ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 214
รูปที่ 4.89 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ BPSK ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 215
รูปที่ 4.90 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ QPSK ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 215
รูปที่ 4.91 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ 16QAM ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 216
รูปที่ 4.92 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ 64QAM ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 216
รูปที่ 4.93 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ 256QAM ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 217
รูปที่ 4.94 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ 1024QAM ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 217
รูปที่ 4.95 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G โดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 218
รูปที่ 4.96 กราฟของสัญญาณมอดูเลชันที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 219
รูปที่ 4.97 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันที่สร้างโดยใช้ซอฟต์แวร์ บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 219
รูปที่ 4.98 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 220
รูปที่ 4.99 กราฟของสัญญาณมอดูเลชันที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 221
รูปที่ 4.100 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 221
รูปที่ 4.101 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G 222
รูปที่ 4.102 กราฟของสัญญาณมอดูเลชันที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G 223

รูปที่ 4.103 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G 223
รูปที่ 4.104 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G 224
รูปที่ 4.105 กราฟของสัญญาณมอดูเลชันที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G 225
รูปที่ 4.106 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G..... 225
รูปที่ 4.107 ผลลัพธ์ทดสอบผลของช่องสัญญาณสื่อสารโดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 227
รูปที่ 4.108 ผลลัพธ์ทดสอบผลของช่องสัญญาณสื่อสารโดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 227
รูปที่ 4.109 ผลลัพธ์ทดสอบผลของช่องสัญญาณสื่อสารโดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G 228
รูปที่ 4.110 กราฟของสัญญาณที่ผ่านช่องสัญญาณสื่อสารโดยใช้ซอฟต์แวร์..... 228
รูปที่ 4.111 แผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านช่องสัญญาณสื่อสารโดยใช้ซอฟต์แวร์ 229
รูปที่ 4.112 ผลลัพธ์ทดสอบผลของช่องสัญญาณสื่อสารโดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G..... 230
รูปที่ 4.113 ผลลัพธ์ทดสอบผลของช่องสัญญาณสื่อสารโดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G..... 230
รูปที่ 4.114 ผลลัพธ์ทดสอบผลของช่องสัญญาณสื่อสารโดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G 231
รูปที่ 4.115 กราฟของสัญญาณที่ผ่านช่องสัญญาณสื่อสารโดยใช้อุปกรณ์ FPGA บนหน้าต่าง..... 231
รูปที่ 4.116 แผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านช่องสัญญาณสื่อสารโดยใช้อุปกรณ์ FPGA 232
รูปที่ 4.117 ผลลัพธ์ของการดีมอดูเลชันโดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G 233
รูปที่ 4.118 ผลลัพธ์ของการดีมอดูเลชันโดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 233
รูปที่ 4.119 ผลลัพธ์การดีมอดูเลชันโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงาน ฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G..... 234
รูปที่ 4.120 กราฟของสัญญาณที่ผ่านการดีมอดูเลชันโดยใช้ซอฟต์แวร์..... 234
รูปที่ 4.121 ผลลัพธ์ของการดีมอดูเลชันโดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G..... 235
รูปที่ 4.122 ผลลัพธ์ของการดีมอดูเลชันโดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 236

รูปที่ 4.123 ผลลัพธ์การดีมอดูเลชันโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้ กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G 236
รูปที่ 4.124 กราฟของสัญญาณที่ผ่านการดีมอดูเลชันโดยใช้อุปกรณ์ FPGA 237
รูปที่ 4.125 ผลลัพธ์การดีสแครมมาตรฐาน 5G โดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G 238
รูปที่ 4.126 กราฟของสัญญาณดีสแครมที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G 238
รูปที่ 4.127 ผลลัพธ์การดีสแครมมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G 239
รูปที่ 4.128 กราฟของสัญญาณที่ได้จากการดีสแครมที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G 239
รูปที่ 4.129 ผลลัพธ์การถอดรหัสช่องสัญญาณมาตรฐาน 5G โดยใช้ซอฟต์แวร์ บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 240
รูปที่ 4.130 กราฟของสัญญาณที่ได้จากการถอดรหัสที่สร้างโดยใช้ซอฟต์แวร์ 241
รูปที่ 4.131 ผลลัพธ์การถอดรหัสช่องสัญญาณมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G 242
รูปที่ 4.132 กราฟของสัญญาณที่ได้จากการถอดรหัสที่สร้างโดยใช้อุปกรณ์ FPGA 242
รูปที่ 4.133 ผลลัพธ์การถอดรหัสช่องสัญญาณมาตรฐาน 5G โดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G 243
รูปที่ 4.134 กราฟของสัญญาณที่ได้จากการถอดรหัสที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่าง ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G 244
รูปที่ 4.135 ผลลัพธ์การถอดรหัสช่องสัญญาณมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G 245
รูปที่ 4.136 กราฟของสัญญาณที่ได้จากการถอดรหัสที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่าง ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G 245
รูปที่ 4.137 ผลการทดสอบอัตราบิตผิดพลาดของการมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G โดยใช้ซอฟต์แวร์ 246
รูปที่ 4.138 ผลการทดสอบอัตราบิตผิดพลาดของการมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA 247
รูปที่ 4.139 ผลการทดสอบอัตราบิตผิดพลาดของการเข้าและถอดรหัสมาตรฐาน 5G โดยใช้ซอฟต์แวร์ 248
รูปที่ 4.140 ผลการทดสอบอัตราบิตผิดพลาดของการเข้าและถอดรหัสมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA 249
รูปที่ 4.141 ผลการทดสอบอัตราบิตผิดพลาดของกระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G โดยใช้ซอฟต์แวร์ 250
รูปที่ 4.142 ผลการทดสอบอัตราบิตผิดพลาดของกระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA 250

รูปที่ 4.143 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เพื่อแสดงข้อมูลอินพุตแบบสุ่ม 253
รูปที่ 4.144 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เข้ารหัสสำหรับชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUCCH PDCCH และ PBCH 254
รูปที่ 4.145 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เข้ารหัสสำหรับชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUSCH และ PDSCH 254
รูปที่ 4.146 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เข้ารหัสสำหรับชุดการเรียนรู้กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUCCH 256
รูปที่ 4.147 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เข้ารหัสสำหรับชุดการเรียนรู้กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PDCCH 258
รูปที่ 4.148 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เข้ารหัสสำหรับชุดการเรียนรู้กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PBCH 259
รูปที่ 4.149 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เข้ารหัสสำหรับชุดการเรียนรู้กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUSCH 261
รูปที่ 4.150 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เพื่อแสดงผลการสแครม 262
รูปที่ 4.151 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เพื่อแสดงผลการดีสแครม 262
รูปที่ 4.152 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เพื่อแสดงการมอดูเลชันมาตรฐาน 5G . 263
รูปที่ 4.153 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เพื่อแสดงผลของช่องสัญญาณสื่อสาร .. 264
รูปที่ 4.154 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เพื่อแสดงผลการดีมอดูเลชันมาตรฐาน 5G 265
รูปที่ 4.155 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUCCH PDCCH และ PBCH 266
รูปที่ 4.156 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUSCH และ PDSCH 267
รูปที่ 4.157 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUCCH 268
รูปที่ 4.158 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PDCCH..... 271
รูปที่ 4.159 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PBCH 272
รูปที่ 4.160 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUSCH และ PDSCH 274
รูปที่ 4.161 ระบบของวงจรบนชิปเอฟพีจีเอ 278
รูปที่ 4.162 รายละเอียดระบบย่อย Serial receiver 279
รูปที่ 4.163 ผลการทดสอบ timing diagram ของระบบย่อย Serial receiver 279
รูปที่ 4.164 รายละเอียดระบบย่อย Receiver buffer 282
รูปที่ 4.165 ผลการทดสอบ timing diagram ของระบบย่อย Receiver buffer 283
รูปที่ 4.166 รายละเอียดระบบย่อย Simulator controller 283

รูปที่ 4.167 ผลการทดสอบ timing diagram ของระบบย่อย Simulator controller 284
รูปที่ 4.168 รายละเอียดระบบย่อย Random input generator 284
รูปที่ 4.169 ผลการทดสอบ timing diagram ของระบบย่อย Random input generator 285
รูปที่ 4.170 รายละเอียดระบบย่อย LDPC encoder 285
รูปที่ 4.171 ผลการทดสอบ timing diagram ของระบบย่อย LDPC encoder 286
รูปที่ 4.172 รายละเอียดระบบย่อย Polar encoder 286
รูปที่ 4.173 ผลการทดสอบ timing diagram ของระบบย่อย Polar encoder 287
รูปที่ 4.174 รายละเอียดระบบย่อย Modulator 287
รูปที่ 4.175 ผลการทดสอบ timing diagram ของระบบย่อย Modulator 288
รูปที่ 4.176 รายละเอียดระบบย่อย Noise generator 288
รูปที่ 4.177 ผลการทดสอบ timing diagram ของระบบย่อย Noise generator 289
รูปที่ 4.178 รายละเอียดระบบย่อย Demodulator 289
รูปที่ 4.179 ผลการทดสอบ timing diagram ของระบบย่อย Demodulator 290
รูปที่ 4.180 รายละเอียดระบบย่อย LDPC decoder 290
รูปที่ 4.181 ผลการทดสอบ timing diagram ของระบบย่อย LDPC decoder 291
รูปที่ 4.182 รายละเอียดระบบย่อย Polar decoder 292
รูปที่ 4.183 ผลการทดสอบ timing diagram ของระบบย่อย Polar decoder 292
รูปที่ 4.184 รายละเอียดระบบย่อย Transmitter buffer 293
รูปที่ 4.185 ผลการทดสอบ timing diagram ของระบบย่อย Transmitter buffer 293
รูปที่ 4.186 รายละเอียดระบบย่อย Serial transmitter 294
รูปที่ 4.187 ผลการทดสอบ timing diagram ของระบบย่อย Serial transmitter 295
รูปที่ 4.188 ชุดข้อมูลที่ส่งไปยังคอมพิวเตอร์ 296
รูปที่ 4.189 ชุดข้อมูลที่รับมาจากอุปกรณ์ FPGA 297
รูปที่ 4.190 ผลการทดสอบการ์ดเร่งความเร็ว FEC แบบ AAL_PDSCH_FEC 298
รูปที่ 4.191 ผลการทดสอบการ์ดเร่งความเร็ว FEC แบบ AAL_PUSCH_FEC 298
รูปที่ 4.192 ผลการทดสอบฮาร์ดแวร์หลังการจัดเตรียม 299
รูปที่ 4.193 ผลการทดสอบ IP-Core DMA/Bridge Subsystem for PCI Express ของ Xilinx 299
รูปที่ 4.194 ผลการทดสอบเว็บไซต์หน้าต่างหน้าแรก 300
รูปที่ 4.195 ผลการทดสอบเว็บไซต์ตัวเลือกเนื้อหาในแถบรหัสช่องสัญญาณ 302
รูปที่ 4.196 ผลการทดสอบเว็บไซต์หน้าต่างช่องสัญญาณคืออะไร 302
รูปที่ 4.197 ผลการทดสอบเว็บไซต์หน้าต่างการมอดูเลชัน 303
รูปที่ 4.198 ผลการทดสอบเว็บไซต์หน้าต่างช่องสัญญาณ AWGN 304
รูปที่ 4.199 ผลการทดสอบเว็บไซต์หน้าต่างชุดการเรียนรู้และทดสอบกรณีใช้งานบนคอมพิวเตอร์ 305
รูปที่ 4.200 ผลการทดสอบเว็บไซต์หน้าต่างชุดการเรียนรู้และทดสอบ กรณีใช้งานบนคอมพิวเตอร์ร่วมกับอุปกรณ์ FPGA 305
รูปที่ 4.201 ส่วนสำหรับดาวน์โหลดซอฟต์แวร์ GUI และเอกสารประกอบ บนหน้าต่างชุดการเรียนรู้และทดสอบ 306

รูปที่ 4.202 แสดงวิดีโอสาธิตการใช้งานซอฟต์แวร์ GUI บนหน้าต่างชุดการเรียนรู้และทดสอบ 306
รูปที่ 4.203 ผลการทดสอบเว็บไซต์หน้าต่างเกี่ยวกับเรา 307
รูปที่ 4.204 บรรยากาศการเผยแพร่ชุดการเรียนรู้ที่สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง 309
รูปที่ 4.205 บรรยากาศการเผยแพร่ชุดการเรียนรู้ที่มหาวิทยาลัยราชมงคลอีสาน วิทยาเขตนครราชสีมา 310
รูปที่ 4.206 บรรยากาศการเผยแพร่ชุดการเรียนรู้ที่มหาวิทยาลัยราชมงคลอีสาน วิทยาเขตขอนแก่น 311
รูปที่ 4.207 บรรยากาศการเผยแพร่ชุดการเรียนรู้ที่มหาวิทยาลัยขอนแก่น 312
รูปที่ 4.208 บรรยากาศการเผยแพร่ชุดการเรียนรู้ที่จุฬาลงกรณ์มหาวิทยาลัย 313

สารบัญตาราง

ตารางที่ 2.1 รายละเอียดเมทริกซ์ฐาน 15
ตารางที่ 2.2 ความสัมพันธ์ระหว่างเซตของการเลื่อนค่า (Shift-value set) และขนาดการยกระดับ (Z) สำหรับรหัส LDPC ที่ใช้ใน 5G NR [10] 16
ตารางที่ 2.3 รายละเอียดค่า $V_{i,j}$ สำหรับเมทริกซ์ฐาน 1 ของรหัส LDPC [10] 17
ตารางที่ 2.4 รายละเอียดค่า $V_{i,j}$ สำหรับเมทริกซ์ฐาน 2 ของรหัส LDPC [10] 19
ตารางที่ 3.1 ประเภทและความยาวของสัญญาณจำแนกระหว่างคอมพิวเตอร์และอุปกรณ์ FPGA 131
ตารางที่ 3.2 ค่าสัมประสิทธิ์สมการพหุนามของช่วงการประมาณ 7 ช่วง 144
ตารางที่ 3.3 ตารางค้นหาการมอดูเลชันสำหรับการมอดูเลชันแบบ $\pi/2$-BPSK 149
ตารางที่ 3.4 ตารางค้นหาการมอดูเลชันสำหรับการมอดูเลชันแบบ BPSK 149
ตารางที่ 3.5 ตารางค้นหาการมอดูเลชันสำหรับการมอดูเลชันแบบ QPSK (2PAM) 150
ตารางที่ 3.6 ตารางค้นหาการมอดูเลชันสำหรับการมอดูเลชันแบบ 16QAM (4PAM) 150
ตารางที่ 3.7 ตารางค้นหาการมอดูเลชันสำหรับการมอดูเลชันแบบ 64QAM (8PAM) 151
ตารางที่ 3.8 ตารางค้นหาการมอดูเลชันสำหรับการมอดูเลชันแบบ 256QAM (16PAM) 151
ตารางที่ 3.9 ตารางค้นหาการมอดูเลชันสำหรับการมอดูเลชันแบบ 1024QAM (32PAM) 151
ตารางที่ 3.10 ตารางค่าเฉลี่ยของบิต 0 ที่ใกล้ y ที่สุดสำหรับ $\pi/2$-BPSK 155
ตารางที่ 3.11 ตารางค่าเฉลี่ยของบิต 1 ที่ใกล้ y ที่สุดสำหรับ $\pi/2$-BPSK 155
ตารางที่ 3.12 ตารางค่าเฉลี่ยของบิต 0 ที่ใกล้ y ที่สุดสำหรับ BPSK 156
ตารางที่ 3.13 ตารางค่าเฉลี่ยของบิต 1 ที่ใกล้ y ที่สุดสำหรับ BPSK 156
ตารางที่ 3.14 ตารางค่าเฉลี่ยของบิต 0 ที่ใกล้ y ที่สุดสำหรับ QPSK (2PAM) 156
ตารางที่ 3.15 ตารางค่าเฉลี่ยของบิต 1 ที่ใกล้ y ที่สุดสำหรับ QPSK (2PAM) 156
ตารางที่ 3.16 ตารางค่าเฉลี่ยของบิต 0 ที่ใกล้ y ที่สุดสำหรับ 16QAM (4PAM) 156
ตารางที่ 3.17 ตารางค่าเฉลี่ยของบิต 1 ที่ใกล้ y ที่สุดสำหรับ 16QAM (4PAM) 157
ตารางที่ 3.18 ตารางค่าเฉลี่ยของบิต 0 ที่ใกล้ y ที่สุดสำหรับ 64QAM (8PAM) 157
ตารางที่ 3.19 ตารางค่าเฉลี่ยของบิต 1 ที่ใกล้ y ที่สุดสำหรับ 64QAM (8PAM) 157
ตารางที่ 3.20 ตารางค่าเฉลี่ยของบิต 0 ที่ใกล้ y ที่สุดสำหรับ 256QAM (16PAM) 158
ตารางที่ 3.21 ตารางค่าเฉลี่ยของบิต 1 ที่ใกล้ y ที่สุดสำหรับ 256QAM (16PAM) 159
ตารางที่ 3.22 ตารางค่าเฉลี่ยของบิต 0 ที่ใกล้ y ที่สุดสำหรับ 1024QAM (32PAM) 159
ตารางที่ 3.23 ตารางค่าเฉลี่ยของบิต 1 ที่ใกล้ y ที่สุดสำหรับ 1024QAM (32PAM) 161
ตารางที่ 3.24 ทรัพยากรของหน่วยความจำ 166
ตารางที่ 4.1 ซอฟต์แวร์ชุดคำสั่งของระบบย่อย 275
ตารางที่ 4.2 ประเภทและความยาวของสัญญาณจำแนกโดยระบบย่อย Receiver buffer 280

บทที่ 1
บทนำ

1.1 ที่มา และความสำคัญของโครงการ

ปัจจุบัน ประเทศไทยมีอุตสาหกรรมด้านการพัฒนาอุปกรณ์โทรคมนาคมอย่างจำกัด เนื่องจากอุปกรณ์โทรคมนาคมเกี่ยวข้องกับเทคโนโลยีที่มีความซับซ้อนสูง จำเป็นต้องใช้บุคคลากรที่มีความรู้ความสามารถระดับโลกเพื่อพัฒนาอุปกรณ์ให้สามารถแข่งขันกับต่างประเทศได้ จากปัญหาดังกล่าว ทำให้ บริษัทที่ดำเนินธุรกิจด้านโทรคมนาคมของประเทศไทย ส่วนใหญ่จะดำเนินธุรกิจในรูปแบบของการให้บริการลูกค้า ปราศจากการพัฒนาอุปกรณ์ของต้นเอง ซึ่งในอดีตที่ผ่านมา บริษัทของไทยสามารถดำเนินธุรกิจการให้บริการอย่างราบรื่น สามารถซื้ออุปกรณ์ของต่างประเทศและนำอุปกรณ์มาให้บริการแต่ลูกค้ารายย่อย อย่างไรก็ตาม สถานการณ์การดำเนินธุรกิจในปัจจุบันเปลี่ยนแปลงไปมาก การดำเนินธุรกิจด้านเทคโนโลยีเป็นแบบไม่มีพรมแดน บริษัทผู้ผลิตอุปกรณ์จากต่างประเทศต้องการมาดำเนินการให้บริการลูกค้าด้วยตัวเอง ไม่เช่นนั้น การซื้ออุปกรณ์จากต่างประเทศจะมีมูลค่าที่สูงมากขึ้นเนื่องจากบริษัทในประเทศไทยไม่สามารถหลีกเลี่ยงการให้งานอุปกรณ์ของบริษัทต่างชาติได้ ปัญหาดังกล่าว มิได้เกิดขึ้นแค่ในประเทศไทย แต่เกิดขึ้นทั่วโลก ทำให้ประเทศต่าง ๆ เช่น สหรัฐอเมริกา เริ่มมีการลงทุนกับอุตสาหกรรมที่เกี่ยวข้องกับเทคโนโลยีเพื่ออนาคตของประเทศ

จากปัญหาดังกล่าว ผู้วิจัยจึงมีแนวความคิดที่การนำความเชียวชาญด้านเทคโนโลยีการเข้ารหัสช่องสัญญาณมาพัฒนาบนอุปกรณ์ FPGA ที่สามารถต่อยอดเชิงพาณิชย์ได้ แม้ว่าผู้วิจัยสนใจพัฒนาอุปกรณ์ FPGA แต่ผู้วิจัยยังคงให้ความสำคัญกับการพัฒนาบุคคลากรและสร้างแรงจูงใจกับนักศึกษารุ่นใหม่ควบคู่ไปด้วย ทำให้ผู้วิจัยจึงดำเนินการจัดทำ “โครงการพัฒนาอุปกรณ์เข้ารหัสช่องสัญญาณมาตรฐาน 5G สำหรับสื่อการสอนและการต่อยอดเชิงพาณิชย์” โดยได้รับทุนสนับสนุนจากกองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคมเพื่อประโยชน์สาธารณะ (กทปส.) ปี พ.ศ. 2563 อย่างไรก็ตาม โครงการก่อนหน้าจะมุ่งเน้นเฉพาะการพัฒนาซอฟต์แวร์และฮาร์ดแวร์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G เท่านั้น มิได้พัฒนาการถอดรหัสช่องสัญญาณ เนื่องจากองค์กร 3GPP ไม่ได้มีการกำหนดว่าการถอดรหัสจะต้องมีขั้นตอนอย่างไร

การพัฒนาการถอดรหัสช่องสัญญาณนั้น จะต้องใช้องค์ความรู้เฉพาะของผู้พัฒนา จึงเป็นที่มาของข้อเสนอโครงการนี้ ส่วนที่ 1 จะมุ่งเน้นการพัฒนาซอฟต์แวร์และฮาร์ดแวร์การถอดรหัสช่องสัญญาณมาตรฐาน 5G เมื่อพัฒนาส่วนดังกล่าวเสร็จสิ้น ผู้วิจัยจะดำเนินการ ส่วนที่ 2 พัฒนาชุดการเรียนรู้การถอดรหัสช่องสัญญาณในมาตรฐาน 5G และ ส่วนที่ 3 พัฒนาชุดทดสอบสมรรถนะการเข้ารหัสและถอดรหัสช่องสัญญาณในระบบสื่อสารไร้สาย โดยงานส่วนที่ 1 และ 2 นี้ จะทำให้ได้ซอฟต์แวร์และฮาร์ดแวร์ของรหัสช่องสัญญาณที่สมบูรณ์ สามารถนำไปใช้งานเชิงพาณิชย์และใช้เป็นสื่อการสอนในวิชาการสื่อสารไร้สายได้ นอกจากนี้ ผู้วิจัยได้ศึกษามาตรฐาน O-RAN ซึ่งเป็นแนวทางการใช้ชุดฮาร์ดแวร์และซอฟต์แวร์ที่มีลักษณะแบบเปิด ซึ่งโอเปอร์เรเตอร์ทั่วโลกให้ความสนใจ

อย่างมาก เนื่องจากสามารถหลีกเลี่ยงการซื้ออุปกรณ์โทรคมนาคมจากบริษัทผู้ผลิตอุปกรณ์โทรคมนาคมได้ ทำให้ ส่วนที่ 4 จะมุ่งเน้นการพัฒนาฮาร์ดแวร์การเข้ารหัสและถอดรหัสช่องสัญญาณในรูปแบบของการ์ดเร่งความเร็ว FEC ซึ่งองค์กร O-RAN Alliance ได้กำหนดให้การ์ดเร่งเป็นส่วนหนึ่งในมาตรฐาน O-RAN

1.2 วัตถุประสงค์

1) เพื่อพัฒนาซอฟต์แวร์และฮาร์ดแวร์การถอดรหัสช่องสัญญาณมาตรฐาน 5G
2) เพื่อพัฒนาชุดการเรียนรู้การถอดรหัสช่องสัญญาณในมาตรฐาน 5G
3) เพื่อพัฒนาชุดทดสอบสมรรถนะการเข้ารหัสและถอดรหัสช่องสัญญาณในระบบสื่อสารไร้สาย
4) เพื่อพัฒนาการ์ดเร่งความเร็ว FEC สำหรับรหัสช่องสัญญาณในโครงข่าย O-RAN

1.3 ขอบเขตของโครงการ

โครงการพัฒนาอุปกรณ์ถอดรหัสช่องสัญญาณมาตรฐาน 5G สำหรับสื่อการสอนและการ์ดเร่งความเร็วสำหรับโครงข่าย O-RAN ประกอบไปด้วย 5 ส่วน

ส่วนที่ 1 พัฒนาการมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G ช่องสัญญาณสื่อสารแบบ AWGN Channel และ Fading Channel ด้วยภาษา Python อีกทั้ง ยังพัฒนาซอฟต์แวร์ดังกล่าวด้วยภาษา VHDL เพื่อให้ทำงานบนอุปกรณ์ประเภท FPGA ได้

ส่วนที่ 2 พัฒนาชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G และชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G ในลักษณะของซอฟต์แวร์ GUI เพื่อให้ผู้ใช้งานสามารถใช้งานได้สะดวก ดังรูปที่ 1.1 ผู้ใช้งานสามารถเห็นขั้นตอนการนำข้อมูลมาเข้ารหัสช่องสัญญาณมาตรฐาน 5G การมอดูเลชันมาตรฐาน 5G การเพิ่มสัญญาณรบกวน การดีมอดูเลชันและการถอดรหัสช่องสัญญาณมาตรฐาน 5G โดยผู้ใช้งานสามารถดูผลลัพธ์ ณ จุดต่าง ๆ รวมถึงสามารถเรียกดูแผนภาพคอนสเตลเลชัน (Constellation Diagrams) ของการมอดูเลชันมาตรฐาน 5G ได้โดยง่าย

ส่วนที่ 3 พัฒนาชุดทดสอบสมรรถนะการมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G และชุดทดสอบสมรรถนะการเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G ในระบบสื่อสารไร้สาย ผู้ใช้งานสามารถเลือกช่องสัญญาณสื่อสารแบบ AWGN Channel หรือ Fading Channel เพื่อทดลองหาสมรรถนะอัตราบิตผิดพลาด (Bit Error Rate) ณ จุดต่าง ๆ ของระบบได้ ดังรูปที่ 1.2

ส่วนที่ 4 พัฒนาการ์ดเร่งความเร็ว FEC ดังรูปที่ 1.3 ซึ่งองค์กร O-RAN Alliance ได้กำหนดให้การ์ดเร่งเป็นส่วนหนึ่งในมาตรฐาน O-RAN โดยลักษณะการใช้งานแบบ AAL_PDSCH_FEC Profile และ AAL_PUSCH_FEC Profile

ส่วนที่ 5 การเผยแพร่ชุดการเรียนรู้ ชุดทดสอบ และการ์ดเร่งความเร็ว FEC ให้แก่มหาวิทยาลัยต่าง ๆ เพื่อใช้ประกอบการเรียนในรายวิชาที่เกี่ยวข้อง และเผยแพร่ให้บริษัทเอกชนทราบถึงแนวทางการพัฒนาอุปกรณ์ FPGA สำหรับงานด้านโทรคมนาคม

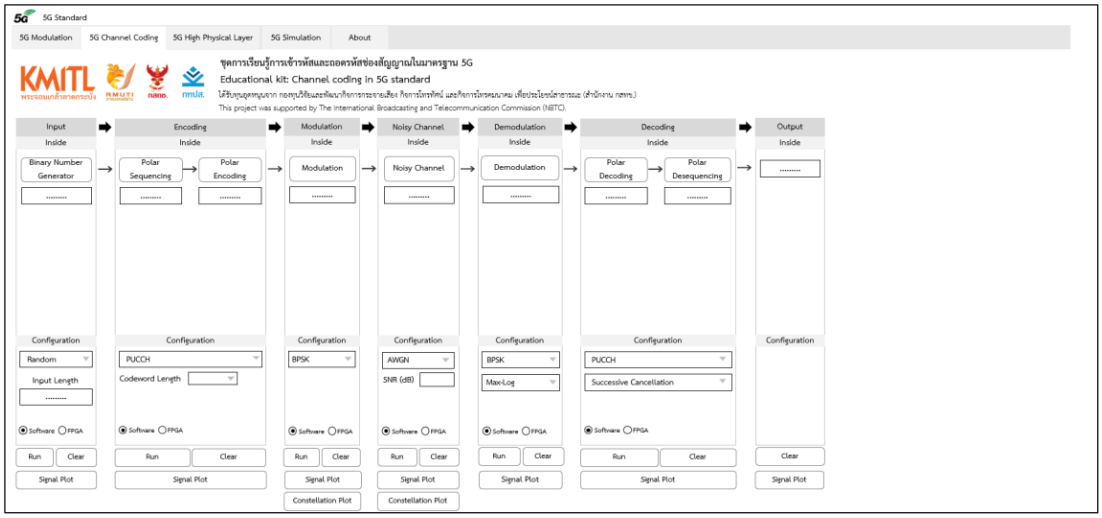
<u>หมายเหตุ</u> ชุดการเรียนรู้และชุดทดสอบ สามารถ download ได้ 3 ช่องทาง

1) Link: https://drive.google.com/file/d/16kuaMyXVoqCacankrETCwCCcPQMciTWV/view

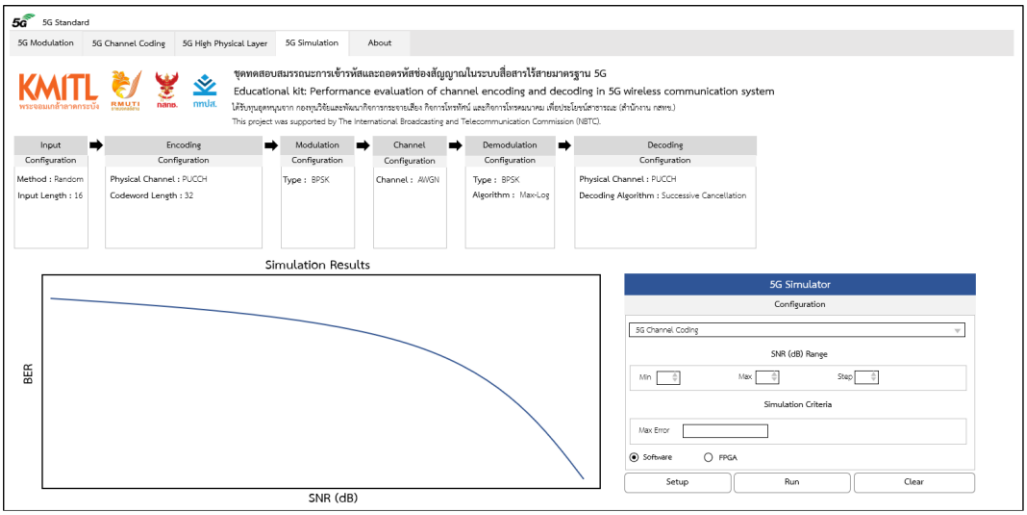
2) Website: https://www.channelcoding.com โดยเลือกหัวข้อ “ชุดการเรียนรู้และชุดทดสอบ” และ “ใช้งานบนคอมพิวเตอร์” ดังรูปที่ 1.4

3) QR code:

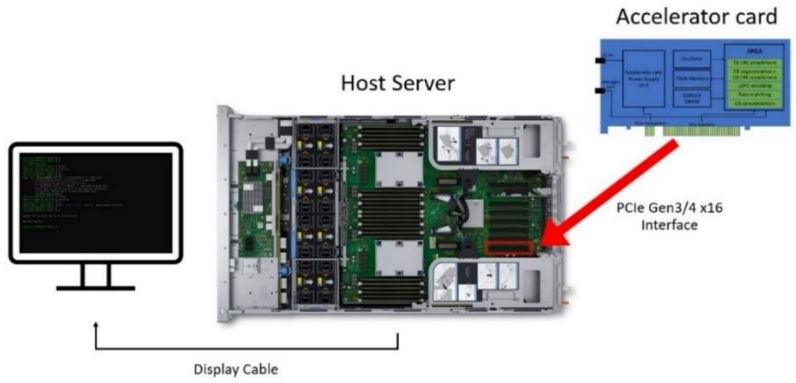
SCAN ME



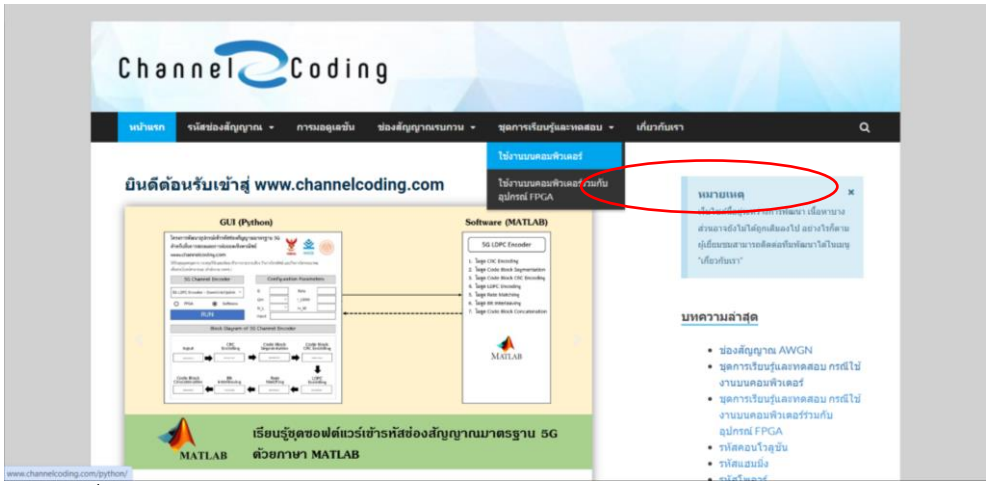
รูปที่ 1.1 ตัวอย่าง ชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G



รูปที่ 1.2 ตัวอย่าง ชุดทดสอบสมรรถนะการเข้ารหัสและถอดรหัสช่องสัญญาณในระบบสื่อสารไร้สาย



รูปที่ 1.3 ตัวอย่าง การ์ดเร่งความเร็ว FEC สำหรับรหัสช่องสัญญาณในโครงข่าย O-RAN



รูปที่ 1.4 ช่องทางการ download ชุดการเรียนรู้และชุดทดสอบ ผ่านทางเว็บไซต์

1.4 ประโยชน์ที่คาดว่าจะได้รับ

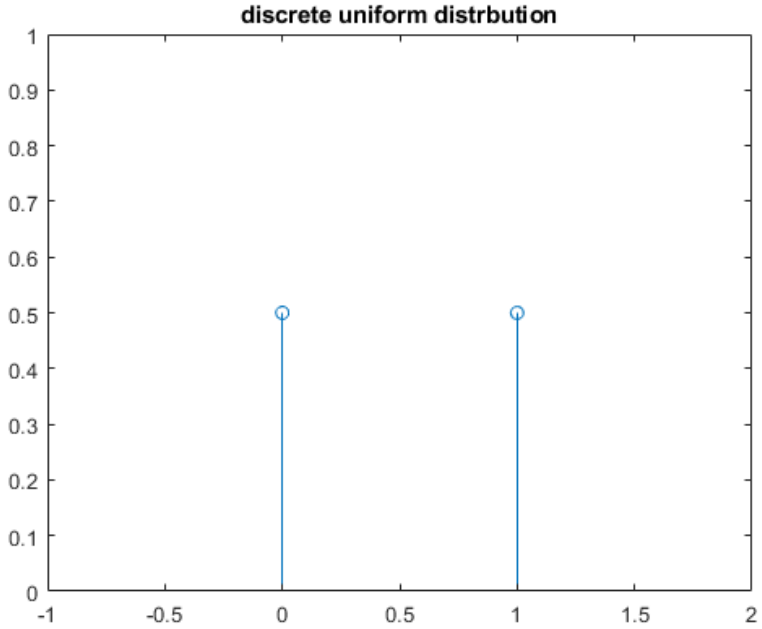
1) มหาวิทยาลัยที่จัดการเรียนการสอนด้านวิศวกรรมโทรคมนาคม
   ได้รับชุดเรียนรู้รหัสช่องสัญญาณในระบบสื่อสารไร้สายที่ทันสมัยและตรงตามมาตรฐาน 5G
2) ผู้ประกอบการด้านอุปกรณ์การสอน
   ได้เห็นแนวทางการพัฒนาอุปกรณ์การสอนที่สอดคล้องกับความต้องการของมหาวิทยาลัย
3) ผู้ประกอบการด้านอุปกรณ์โทรคมนาคม
   ได้เห็นแนวทางการพัฒนาการ์ดเร่งความเร็วที่ใช้งานในโครงข่าย O-RAN

บทที่ 2
ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

2.1 พื้นฐานสัญญาณสุ่มและช่องสัญญาณสื่อสาร

2.1.1 สัญญาณสุ่มที่มีการกระจายตัวแบบสม่ำเสมอ

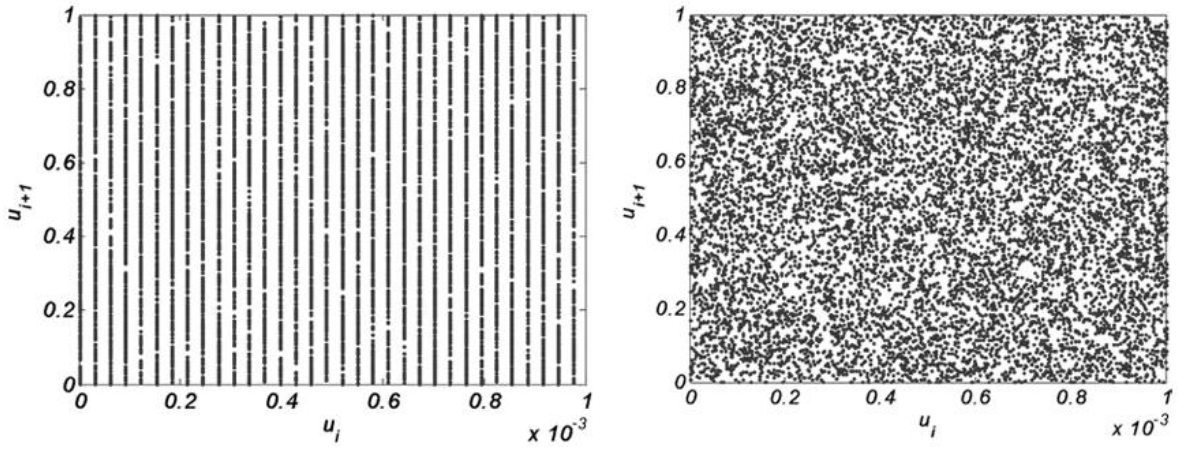
สัญญาณสุ่มนี้จัดเป็นข้อมูลที่มีการกระจายตัวแบบสม่ำเสมอ (Uniform Distribution) โดยความน่าจะเป็นของค่าสุ่มที่เป็นไปได้ทั้งหมดจะมีโอกาสถูกสุ่มขึ้นเท่ากัน ในกรณีนี้ค่าสุ่มที่เป็นไปได้คือ 0 และ 1 ซึ่งแสดงฟังก์ชันความหนาแน่นความน่าจะเป็นดังรูปที่ 2.1



รูปที่ 2.1 ความน่าจะเป็นของค่าสุ่ม 0 และ 1 เมื่อถูกสุ่มมาจำนวนหลายค่า

ข้อมูลที่มีการกระจายตัวแบบสม่ำเสมอจะถูกนำมาใช้งานเป็นข้อมูลอินพุตสำหรับระบบการสื่อสาร สำหรับการสร้างข้อมูลอินพุตแบบสุ่มที่มีการกระจายคงที่ สามารถทำได้ด้วยวงจร Shift Register เช่น วงจร Linear-Feedback Shift Register หรือ LFSR และวงจร Tausworthe generator การสร้างข้อมูลสุ่มที่ดี ข้อมูลที่สุ่มออกมาต้องมีความสุ่มเทียม (Pseudorandomness) สูง หรือมีค่าสหสัมพันธ์ (Cross-Correlation) ต่ำ กล่าวคือค่าที่สุ่มออกมาควรดูเหมือนมีความสุ่มสูง ทั้งที่ถูกสร้างออกมาจากกระบวนการ Deterministic และทำซ้ำได้ รูปที่ 2.2 เปรียบเทียบผลลัพธ์การสุ่มค่าของวงจร 2 ประเภท (ก) จากวงจร LFSR และ (ข) จากวงจร Tausworthe generator แบบรวม (Combined Tausworthe Generator) หรือ CTG ซึ่งจะสังเกตได้ว่าวงจร LFSR ให้ผลลัพธ์ที่ดูเป็นโครงสร้างมากกว่าวงจร CTG ซึ่งถือว่าไม่เหมาะสมสำหรับการสุ่ม ผลลัพธ์ที่เป็นโครงสร้างนี้บ่งบอกถึง

การมีสหสัมพันธ์ต่อกันหรือสามารถคาดเดาผลลัพธ์การสุ่มลำดับถัดไปได้ อีกหนึ่งตัวแปรที่สำคัญของการสร้างข้อมูลสุ่มคือขนาดของคาบ (Period) ซึ่งบ่งบอกถึงจำนวนค่าสุ่มที่จะไม่เกิดลำดับที่ซ้ำกัน สำหรับวงจร LFSR มักจะมีขนาดคาบเท่ากับสองยกกำลังความยาวของวงจรสุ่ม ขณะที่วงจร CTG

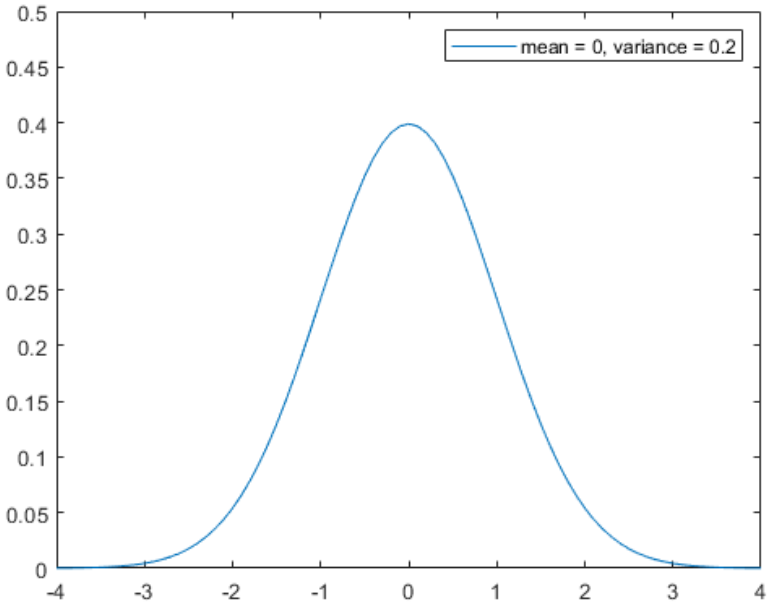


(ก) (ข)

รูปที่ 2.2 แผนภาพการกระจายสองมิติ (2D scatter plot) ของผลลัพธ์คู่เลขสุ่ม (u_i, u_{i+1}) ลำดับติดต่อกันจาก (ก) วงจร LFSR (ข) วงจร CTG [1]

2.1.2 สัญญาณสุ่มที่มีการกระจายตัวแบบสม่ำเสมอ

สัญญาณสุ่มนี้จัดเป็นข้อมูลที่มีการกระจายตัวแบบปกติ (Normal Distribution) หรือการกระจายเกาส์เซียน (Gaussian Distribution) โดยความน่าจะเป็นของค่าสุ่มที่เป็นไปได้ทั้งหมดจะมีโอกาสถูกสุ่มขึ้นไม่เท่ากัน โดยค่าที่มีโอกาสถูกสุ่มมากที่สุด จะเป็นค่าที่เทียบเท่ากับค่าเฉลี่ย (Mean) หรือ μ ของการกระจายตัวดังกล่าว และค่าข้างเคียงของค่าเฉลี่ยจะมีโอกาสถูกสุ่มลดลงไปเรื่อย ๆ จนเข้าใกล้ค่าศูนย์ ลักษณะการถูกสุ่มที่ลดลงของค่าข้างเคียงค่าเฉลี่ย จะถูกควบคุมด้วยค่าเบี่ยงเบนมาตรฐาน (Standard Deviation) หรือ σ ของการกระจายตัวดังกล่าวอีกเช่นกัน ซึ่งแสดงฟังก์ชันความหนาแน่นความน่าจะเป็นดัง



รูปที่ 2.3 ความน่าจะเป็นของค่าสุ่มของการกระจายแบบปกติ

ข้อมูลที่มีการกระจายตัวแบบปกติจะถูกนำมาใช้งานเป็นสัญญาณสุ่มสำหรับระบบการสื่อสารภายใต้ช่องสัญญาณรบกวนเกาส์เซียนขาวแบบบวก (Additive White Gaussian Noise) หรือ AWGN

2.1.3 ช่องสัญญาณรบกวนเกาส์เซียนขาวแบบบวก

ช่องสัญญาณรบกวนเกาส์เซียนขาวแบบบวกคือรูปแบบช่องสัญญาณพื้นฐานตามทฤษฎีข่าวสารเพื่ออธิบายถึงปรากฏการณ์ของกระบวนการสุ่มที่เกิดขึ้นในธรรมชาติ รวมถึงในระบบการสื่อสาร สัญญาณรบกวนประเภทนี้มักใช้ในการอธิบายสัญญาณพื้นหลัง โดยอาจเกิดขึ้นจากธรรมชาติหรืออุณหภูมิของอุปกรณ์ในระบบ เรียกว่าสัญญาณรบกวนเชิงความร้อนได้ และสัญญาณรบกวนที่เกิดขึ้นในช่องสัญญาณนี้คือสัญญาณรบกวนเกาส์เซียน (Gaussian noise) โดยคำศัพท์ดังกล่าวประกอบไปด้วยคำ 3 คำ ดังนี้

1) บวก (Additive) เนื่องจากสัญญาณประเภทนี้ถูกเพิ่มหรือบวกเข้าไปยังระบบหรือช่องสัญญาณ

2) ขาว (White) อ้างถึงความหนาแน่นกำลังเชิงสเปกตรัม (Power Spectral Density) ที่มีลักษณะสม่ำเสมอในโดเมนความถี่ภายในระบบหรือช่องสัญญาณ ซึ่งลักษณะต่าง ๆ จะถูกแทนด้วยชื่อสีและสีขาวจะมีลักษณะสม่ำเสมอ

3) เกาส์เซียน (Gaussian) อ้างอิงความหนาแน่น (Density) ที่มีลักษณะการกระจายปกติหรือการกระจายเกาส์เซียนในโดเมนเวลาภายในระบบหรือช่องสัญญาณ

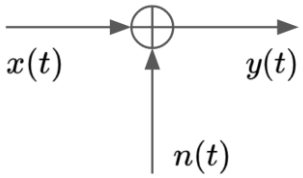
สำหรับช่องสัญญาณรบกวนดังกล่าวแสดงแบบจำลองดังรูปที่ 2.4 โดยกำหนดให้ $y(t)$ เป็นเอาต์พุตช่องสัญญาณ ซึ่งเกิดจากสัญญาณที่ส่ง $x(t)$ บวกกับสัญญาณรบกวนเกาส์เซียน $n(t)$ เขียนความสัมพันธ์ของสัญญาณต่าง ๆ ได้ดังนี้

$$y(t) = x(t) + n(t) \tag{2.1}$$

และสามารถแทนสัญญาณรบกวนเกาส์เซียนได้ดังนี้

$$n(t) \sim C\mathcal{N}(0, \mathbf{K}) \tag{2.2}$$

โดยที่สัญญาณรบกวนเกาส์เซียน $n(t)$ มีค่าเฉลี่ยเท่ากับ 0 และมีเมทริกซ์ความแปรปรวนร่วม (covariance matrix) เท่ากับ $\mathbf{K}$



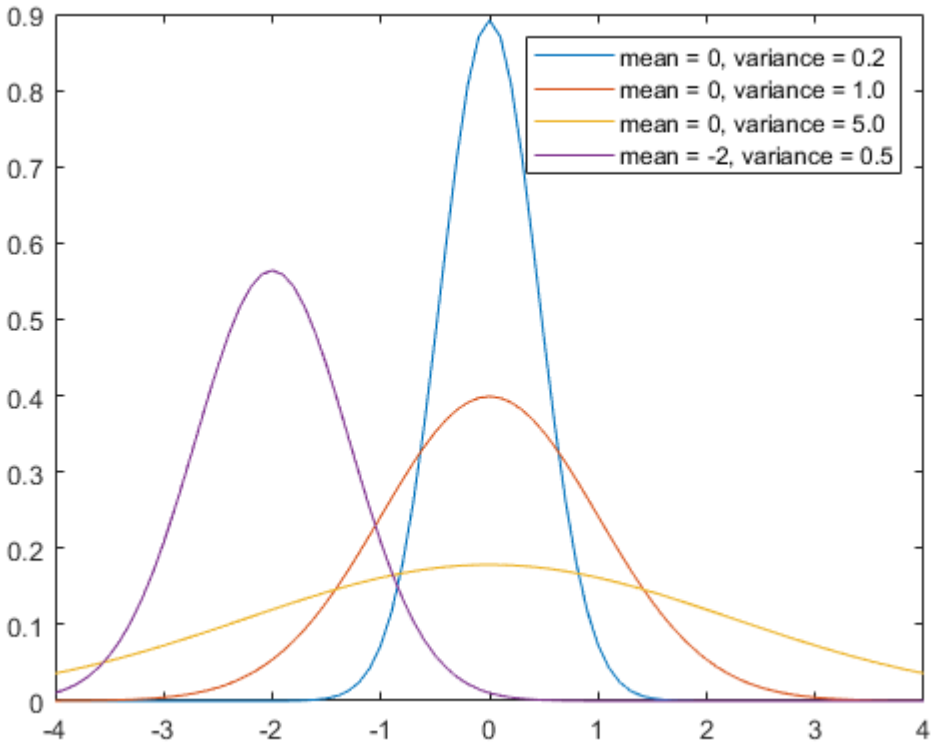
รูปที่ 2.4 แบบจำลองช่องสัญญาณรบกวนเกาส์เซียนขาวแบบบวก

หากกำหนดให้สัญญาณที่ส่งมีสัญญาณความถี่เดียวหรือมีฟังก์ชันความหนาแน่นสเปกตรัมแบบอิมพัลส์ (impulse) เมื่อสัญญาณดังกล่าวผ่านช่องสัญญาณรบกวนนี้ สัญญาณที่ส่งจะถูกบวกกับสัญญาณรบกวนเกาส์เซียนที่ช่วงเวลาต่าง ๆ ได้เอาต์พุตช่องสัญญาณที่มีการกระจายของฟังก์ชันความหนาแน่นสเปกตรัมแบบเกาส์เซียนเช่นเดียวกับสัญญาณรบกวนเกาส์เซียน แสดงดังรูปที่ 2.5 และเขียนสมการฟังก์ชันความหนาแน่นแบบเกาส์เซียนได้ดังนี้

$$f(y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y-\mu)^2}{2\sigma^2}} \tag{2.3}$$

โดย μ คือค่าเฉลี่ยของการกระจาย (distribution) σ^2 คือความแปรปรวน (variance) และ y คือตัวแปรสุ่ม ซึ่งสามารถแทนด้วยสัญญาณต่าง ๆ เช่น เอาต์พุตช่องสัญญาณ $y(t)$ สัญญาณรบกวนเกาส์เซียน ณ เวลาต่าง ๆ จะมีคุณสมบัติการกระจายที่อิสระและเหมือนกัน (independent and identically distributed) โดยความจุช่องสัญญาณ C ของช่องสัญญาณรบกวนเกาส์เซียนขาวแบบบวก

$$C = -\int_{-\infty}^{\infty} p(y) \log_2 p(y)\, dy - \frac{1}{2}\log_2\left(2\pi e\sigma^2\right) \tag{2.4}$$



รูปที่ 2.5 ฟังก์ชันความหนาแน่นสเปกตรัมแบบเกาส์เซียน

โดยที่ $p(y)=\frac{1}{\sqrt{8\pi\sigma^2}}\left(e^{-\frac{(y-1)^2}{2\sigma^2}}+e^{-\frac{(y+1)^2}{2\sigma^2}}\right)$

การจำลองช่องสัญญาณรบกวนเกาส์เซียนขาวแบบบวก สามารถทำได้โดยการสร้างสัญญาณรบกวนเกาส์เซียน เพื่อนำมาบวกกับสัญญาณที่ส่ง การสร้างสัญญาณรบกวนเกาส์เซียนสามารถทำได้โดยการสุ่มสัญญาณขึ้นมา โดยที่สัญญาณที่สุ่มทั้งหมดจะมีการกระจายเกาส์เซียนที่มีค่าเฉลี่ยเท่ากับ 0 และความแปรปรวนเท่ากับ 1 หรือ $n(t)\sim\mathcal{N}(0,1)$ แต่ความเป็นจริงแล้ว ความแปรปรวนของสัญญาณรบกวนเกาส์เซียนจะมีค่าขึ้นอยู่กับสภาพของช่องสัญญาณรบกวน เมื่อสภาพช่องสัญญาณเปลี่ยนไปจะทำให้สัญญาณรบกวนเกาส์เซียนเปลี่ยนแปลงไปเช่นเดียวกัน ซึ่งสามารถปรับค่าความแปรปรวนของได้โดย

$$C=-\int_{-\infty}^{\infty}p(y)\log_2 p(y)\,dy-\frac{1}{2}\log_2\left(2\pi e\sigma^2\right) \qquad (2.5)$$

2.2 พื้นฐานรหัสช่องสัญญาณ

รหัสช่องสัญญาณเป็นองค์ประกอบสำคัญในระบบสื่อสารดิจิทัลที่ขาดไม่ได้ เนื่องจากรหัสช่องสัญญาณทำให้การรับส่งข้อมูลดิจิทัลปราศจากความผิดพลาด เทคโนโลยีเครือข่ายโทรศัพท์เคลื่อนที่ยุคที่ 4 หรือ 4G ได้ประยุกต์ใช้รหัสช่องสัญญาณที่เรียกว่า turbo codes ใน data channel และ tail-biting convolutional codes (TBCC) ใน control channel [2] รหัสช่องสัญญาณ turbo codes เป็นรหัสช่องสัญญาณที่มีความน่าสนใจอย่างยิ่ง เนื่องจากเป็นรหัสช่องสัญญาณชนิดแรกที่มีสรรมถนะเข้าใกล้ทฤษฎีความจุช่องสัญญาณ ซึ่งเป็นเหตุผลสำคัญที่ทำให้ turbo code ถูกประยุกต์ใช้ในมาตรฐาน 3G และ 4G (รหัสช่องสัญญาณ turbo codes ถูกคิดค้นโดย Claude Berrou Alain Glavieux และ รศ.ดร.ปัญญา ฐิติมัชฌิมา [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 จะมีสรรมถนะเข้าใกล้ทฤษฎีความจุช่องสัญญาณเช่นเดียวกับ turbo codes แต่การพัฒนาวงจรเข้ารหัสและถอดรหัสของ 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 ซึ่งเป็นรหัสช่องสัญญาณที่สามารถพิสูจน์ได้ว่ามีสรรมถนะเข้าใกล้ทฤษฎีความจุช่องสัญญาณก็เป็นอีกตัวเลือกที่น่าสนใจสำหรับการประยุกต์ใช้ในมาตรฐาน 5G (รหัสช่องสัญญาณ polar codes ถูกคิดค้นโดย Erdal Arıkan [7] ในปี ค.ศ. 2009 ได้รับความสนใจจากนักวิจัยจำนวนมากเนื่องจากการวิธีการเข้ารหัส polar codes สามารถพิสูจน์ได้โดยตรงว่ามีสรรมถนะเข้าใกล้ทฤษฎีความจุช่องสัญญาณ ต่างจาก turbo codes และ LDPC codes ที่ไม่สามารถพิสูจน์ได้โดยตรงว่ามีสรรมถนะเข้าใกล้ทฤษฎีความจุช่องสัญญาณ

2.2.1 พื้นฐานรหัส LDPC

รหัสตรวจสอบภาวะหนาแน่นต่ำ หรือรหัส LDPC [4] เป็นรหัสแก้ไขความผิดพลาด (ECC: Error Correction Code) ประเภทหนึ่งซึ่งจัดเป็นรหัสบล็อกเชิงเส้น (Linear block code) โดยรหัสบล็อกเชิงเส้นแบบ (N, K) คือรหัสแก้ไขความผิดพลาดประเภทหนึ่งที่มีการแบ่งบิตข้อมูล (Message bit) ที่จะส่งออกเป็นบล็อกย่อย แต่ละบล็อกมีความยาว K บิต สามารถเขียนเป็นเวกเตอร์ได้ดังนี้

$\mathbf{m} = [m_1, m_2, ...m_K]$ บิตข้อมูลที่ถูกแบ่งเป็นบล็อกจะถูกนำไปเข้ารหัสโดยการเติมบิตพาริตี (Parity

$$\begin{aligned} c_1 + c_2 + c_3 + c_4 &= 0 \\ c_1 + c_5 + c_6 + c_7 &= 0 \\ c_2 + c_5 + c_8 + c_9 &= 0 \\ c_3 + c_6 + c_8 + c_{10} &= 0 \\ c_4 + c_7 + c_9 + c_{10} &= 0 \end{aligned} \qquad (2.6)$$

bit) หรือเขียนรูปเวกเตอร์เป็น $\mathbf{p} = [p_1, p_2, ...p_{N-K}]$ เข้าไปทำให้ได้คำรหัส (Codeword) ที่มีความยาว N บิต ซึ่งสามารถเขียนเป็นเวกเตอร์ได้ดังนี้ $\mathbf{c} = [c_1, c_2, ..., c_N]$ บิตพาริตีที่ถูกเพิ่มเข้าไปเป็นส่วนที่ช่วยให้ภาครับสามารถตรวจหาบิตข้อมูลที่มีความผิดพลาดได้ และถ้ามีบิตพาริตีมากพอก็อาจจะสามารถแก้ไขบิตที่ผิดพลาดให้ถูกต้องได้

ในการแบ่งบิตข้อมูลออกเป็นบล็อกย่อย ๆ นั้น ขนาดของบล็อกข้อมูลจะขึ้นอยู่กับแต่ละงานประยุกต์ ซึ่งสัดส่วนของบิตข้อมูลต่อบิตคำรหัสในแต่ละบล็อกจะเรียกว่าอัตรารหัส (Code rate) R ซึ่งนิยามโดย $R = K / N$ เมื่อ $0 \le R \le 1$ เสมอ

รหัสบล็อกเชิงเส้นแบบ (N, K) สามารถถูกกำหนดด้วยเมทริกซ์ตัวกำเนิด $\mathbf{G}$ ที่มีขนาด $K \times N$ โดยเมทริกซ์ตัวกำเนิดจะสัมพันธ์กับเมทริกซ์พาริตีเช็ก $\mathbf{H}$ ผ่านความสัมพันธ์ $\mathbf{GH}^T = \mathbf{0}$ และคำรหัสจะต้องสอดคล้องกับตามความสัมพันธ์ $\mathbf{Hc}^T = \mathbf{0}$ เสมอ นอกจากนี้สมาชิกในแนวแถวของเมทริกซ์ คือสมการพาริตีเช็ก (Parity check equation) ซึ่งเป็นตัวที่กำหนดความสัมพันธ์ของบิตข้อมูลในแต่ละคำรหัส

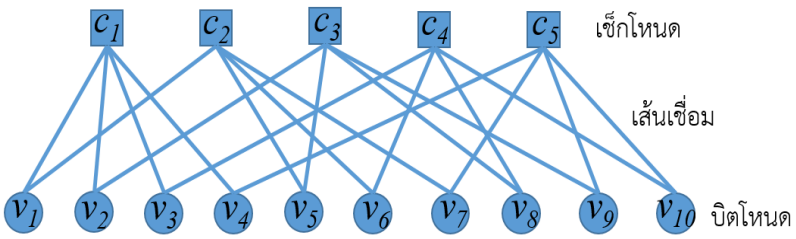
$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 \end{bmatrix}$$

รูปที่ 2.6 ตัวอย่างเมทริกซ์พาริตีเช็ก

ตัวอย่างเมทริกซ์พาริตีในรูปที่ 2.6 ซึ่งเป็นรหัสบล็อกเชิงเส้นแบบ (10, 5) พบว่าเมทริกซ์พาริตีเช็กมีทั้งหมด 5 แถว สามารถเขียนสมการพาริตีเช็กดังนี้

เมทริกซ์พาริตีเช็ก $\mathbf{H}$ ของรหัสบล็อกเชิงเส้นสามารถแสดงในรูปของกราฟแทนเนอร์ (Tanner graph) [8] $G(\mathbf{V},\mathbf{C},\mathbf{E})$ โดยที่กราฟแทนเนอร์เป็นกราฟแบบสองส่วน (Bipartite graph) ซึ่งประกอบด้วยเซตของโหนดตัวแปร $\mathbf{V}$ เมื่อ $\mathbf{V}=[v_1,v_2,...,v_N]$ คือเซตของโหนดตัวแปร (Variable node) และ $\mathbf{C}=[c_1,c_2,...,c_{N-K}]$ คือเซตของโหนดเช็ก (Check node) และเซตของเส้นเชื่อม (Edge) $\mathbf{E}$ ซึ่งเชื่อมต่อระหว่างโหนดตัวแปร v_j และโหนดเช็ก c_i สามารถเขียนแทนด้วย (v_j,c_i) โดยที่ $(v_j,c_i)\in\mathbf{E}$ ก็ต่อเมื่อ $h_{ji}\neq 0$ เมื่อ $h_{ji}\in\mathbf{H}$

จากตัวอย่างในรูปที่ 2.6 จะได้ว่าเซตของโหนดตัวแปรคือ $\mathbf{V}=[v_1,v_2,v_3,v_4,v_5,v_6,v_7,v_8,v_9,v_{10}]$ เซตของโหนดเช็กคือ $\mathbf{C}=[c_1,c_2,c_3,c_4,c_5]$ และเซตของเส้นเชื่อมคือ $E=[(v_1,c_1),(v_1,c_2),(v_2,c_1),(v_2,c_3),(v_3,c_1),(v_3,c_4),(v_4,c_1),(v_4,c_5),(v_5,c_2),(v_5,c_3),(v_6,c_2),(v_6,c_4),(v_7,c_2),(v_7,c_8),(v_8,c_3),(v_8,c_4),(v_9,c_3),(v_9,c_5),(v_{10},c_4),(v_{10},c_5)]$ กราฟแทนเนอร์ของเมทริกซ์พาริตีเช็ก $\mathbf{H}$ สามารถแสดงได้ดังรูปที่ 2.7



รูปที่ 2.7 กราฟแทนเนอร์ของเมทริกซ์พาริตีเช็ก

สำหรับรหัส LDPC ที่ใช้ในมาตรฐาน 5G จะเป็นรหัส LDPC แบบ PBRL [9] ซึ่งจะใช้เมทริกซ์ฐาน 2 แบบได้แก่ เมทริกซ์ฐาน 1 และเมทริกซ์ฐาน 2 เพื่อให้สามารถรองรับความยาวข้อมูลและอัตรารหัสที่หลากหลาย เมทริกซ์ฐานที่ใช้ในมาตรฐาน 5G แสดงรูปที่ 2.8 และรูปที่ 2.9 รหัส LDPC สำหรับช่องสัญญาณข้อมูลในมาตรฐาน release 15 เริ่มต้นจากการพิจารณาค่า MCS (Modulation and coding scheme) เพื่อให้ทราบอัตรารหัสที่ต้องใช้ในการเข้ารหัส แล้วทำการเลือกเมทริกซ์ฐานของรหัส LDPC ซึ่งเมทริกซ์ฐาน 1 ใช้สำหรับบล็อกข้อมูลขนาดใหญ่ $(44\leq K\leq 8448)$ และอัตรารหัสสูงในช่วง $(1/3\leq R\leq 8/9)$ ในขณะที่เมทริกซ์ฐาน 2 ใช้สำหรับบล็อกข้อมูลขนาดเล็ก $(20\leq K\leq 3840)$ และอัตรารหัสที่ต่ำในช่วง $(1/5\leq R\leq 10/13)$ โดยมีเงื่อนไขในการพิจารณาเลือกดังนี้

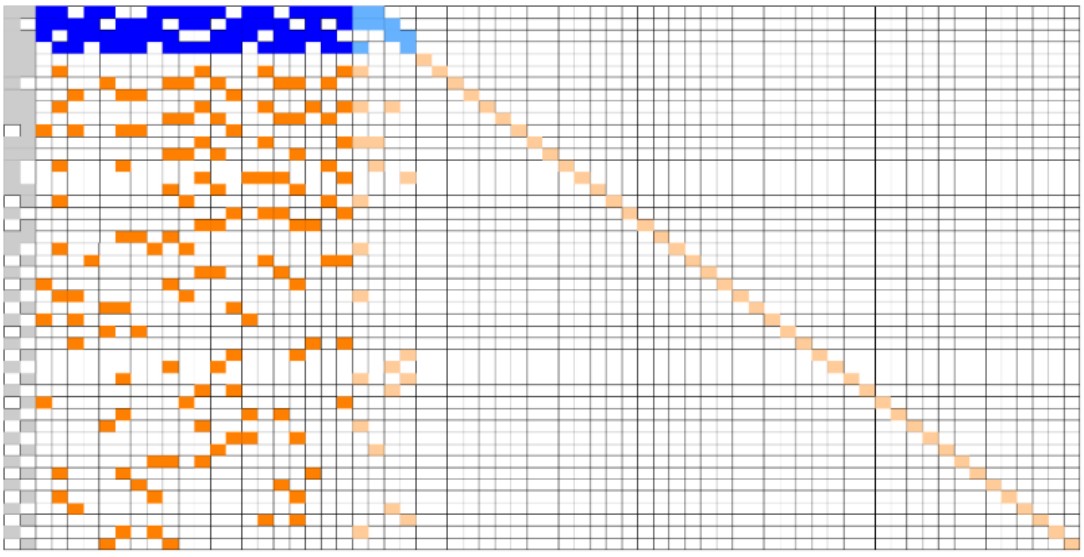
1) กรณีความยาวข้อมูลเท่ากับ $K\leq 292$ บิต ใช้เมทริกซ์ฐาน 2

2) กรณีอัตรารหัสเท่ากับ $R \le 0.67$ และความยาวข้อมูลเท่ากับ $K \le 3824$ บิต ใช้เมทริกซ์ฐาน 2

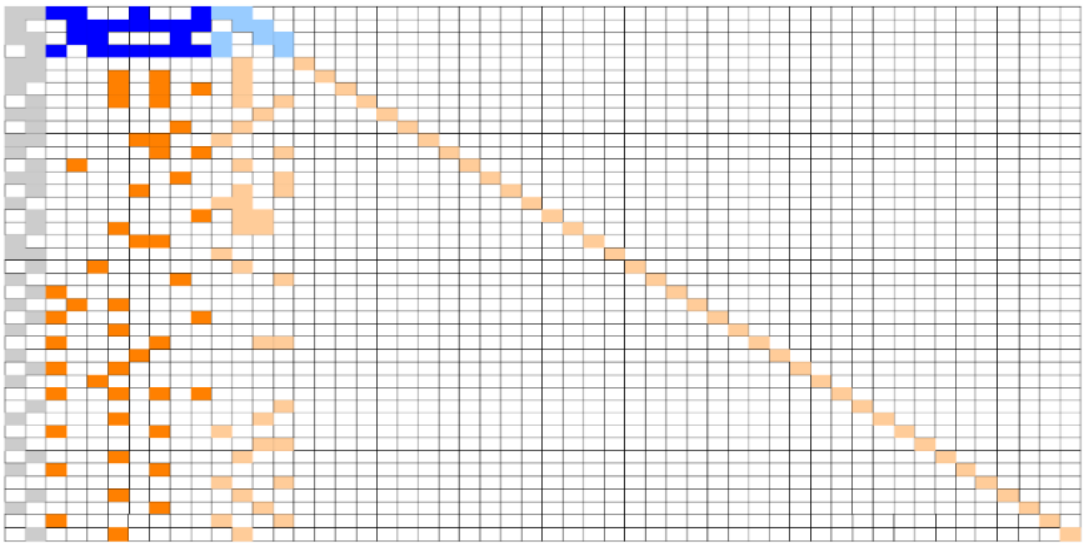
3) กรณีอัตรารหัสเท่ากับ $R \le 0.25$ ใช้เมทริกซ์ฐาน 2

4) กรณีอื่น ๆ ใช้ เมทริกซ์ฐาน 1

โครงสร้างของรหัส LDPC ในมาตรฐาน release 15 [10] หรือ มาตรฐาน 5G เฟส 1 แสดงได้โดยใช้เมทริกซ์ $\mathbf{H}$ เมทริกซ์ฐาน 1 ดังรูปที่ 2.8 และเมทริกซ์ฐาน 2 ดังรูปที่ 2.9 ซึ่งมีรายละเอียดของเมทริกซ์ฐานของทั้งเมทริกซ์ฐาน 1 และเมทริกซ์ฐาน 2 แสดงดังตารางที่ 2.1 โดยช่องสีขาวคือเมทริกซ์ศูนย์ขนาด $Z \times Z$ และช่องสีอื่นๆ คือเมทริกซ์ที่มีการสลับเปลี่ยนแบบเวียนหนุน (Circular permutation matrix) ขนาด $Z \times Z$ การเลือกค่า Z ต่ำสุดจากตารางที่ 2.2 โดยสอดคล้องกับเงื่อนไข $K_b Z \ge K'$ และ $K = 22Z$ ในกรณีเมทริกซ์ฐาน 1 และ $K = 10Z$ ในกรณีเมทริกซ์ฐาน 2 เมื่อ K_b คือความยาวบล็อกข้อมูล และ K' คือความยาวบล็อกข้อมูลที่มีการเติมบิตรหัส CRC (Cyclic redundancy check) ต่อท้าย เมื่อทราบค่า Z แล้วจึงสามารถสร้างเมทริกซ์ที่มีการสลับเปลี่ยนแบบเวียนหมุนได้จากการนำเมทริกซ์เอกลักษณ์มาดำเนินการหมุนแบบวนกลับ โดยจำนวนครั้งของการหมุนสามารถหาได้จาก $P_{i,j} = V_{i,j} \oplus Z$ โดยค่า $V_{i,j}$ หาได้จากตารางที่ 2.3 และตารางที่ 2.4 โดยขึ้นอยู่กับค่าดัชนีเซต (Set index) และเมทริกซ์ฐานของรหัส LDPC



รูปที่ 2.8 เมทริกซ์ฐาน 1



รูปที่ 2.9 เมทริกซ์ฐาน 2

ตารางที่ 2.1 รายละเอียดเมทริกซ์ฐาน

รายละเอียดของเมทริกซ์ฐาน	**เมทริกซ์ฐาน 1**	**เมทริกซ์ฐาน 2**
อัตรารหัส (R) ต่ำสุด	1/3	1/5
ขนาดของเมทริกซ์	46x68	42x52
หลักที่เกี่ยวข้องกับบิตข้อมูล	1-22	1-10
หลักที่เกี่ยวข้องกับบิตพาริตี	23-68	11-52
หลักที่เกี่ยวข้องกับบิต puncture	1-2	1-2

ตารางที่ 2.2 ความสัมพันธ์ระหว่างเซตของการเลื่อนค่า (Shift-value set) และขนาดการยกระดับ (Z) สำหรับรหัส LDPC ที่ใช้ใน 5G NR [10]

ดัชนีเซตของการเลื่อนค่า	ขนาดการยกระดับ (Z)
0	2, 4, 8, 16, 32, 64, 128, 256
1	3, 6, 12, 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, 140

ตารางที่ 2.3 รายละเอียดค่า $V_{i,j}$ สำหรับเมทริกซ์ฐาน 1 ของรหัส LDPC [10]

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

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

ตารางที่ 2.3 (ต่อ) รายละเอียดค่า $V_{i,j}$ สำหรับเมทริกซ์ฐาน 1 ของรหัส LDPC [10]

$\mathbf{H}_{BG}$		$V_{i,j}$								$\mathbf{H}_{BG}$		$V_{i,j}$							
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
	21	123	51	267	130	79	258	161	180		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	31	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	32	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
	28	0	0	0	0	0	0	0	0		24	121	287	217	30	162	83	20	211
7	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	33	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	34	0	127	230	187	82	197	60	4	161
8	0	112	307	295	33	54	348	172	181		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	35	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
	24	90	170	154	201	54	244	116	38		57	0	0	0	0	0	0	0	0
	30	0	0	0	0	0	0	0	0	36	0	37	210	259	222	216	135	6	11
9	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	37	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
	18	61	63	135	109	76	23	164	92		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	38	0	167	185	151	123	190	164	91	121
10	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	39	1	173	258	102	12	153	236	4	115
	14	58	354	311	103	201	267	70	84		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
11	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	40	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
	23	22	115	280	201	26	192	124	107	41	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
12	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	42	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
	34	0	0	0	0	0	0	0	0		64	0	0	0	0	0	0	0	0
13	0	177	313	39	81	231	311	52	220	43	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	44	0	139	80	234	84	18	79	4	191
14	0	206	142	78	14	0	22	1	124		7	157	78	227	4	0	244	6	211
	12	55	248	299	175	186	322	202	144		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	45	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
15	0	40	241	229	90	170	176	173	39		67	0	0	0	0	0	0	0	0

ตารางที่ 2.4 รายละเอียดค่า $V_{i,j}$ สำหรับเมทริกซ์ฐาน 2 ของรหัส LDPC [10]

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

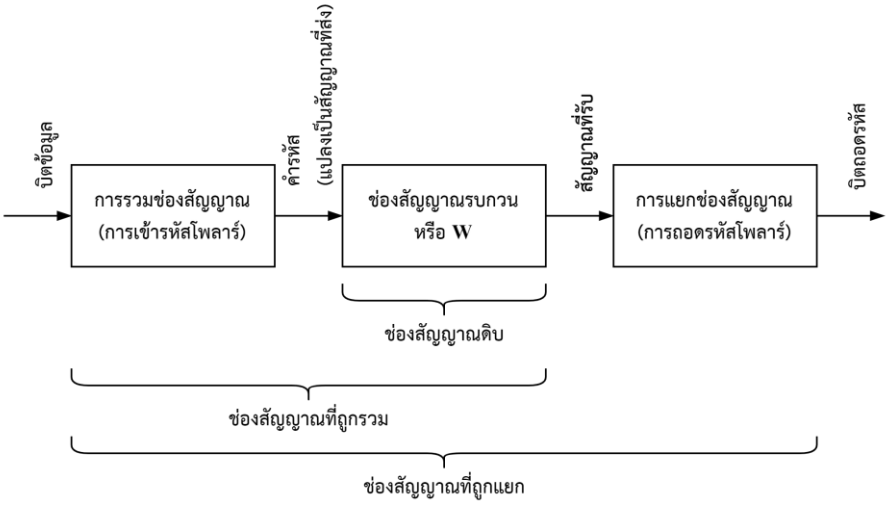
$\mathbf{H}_{\mathrm{BG}}$		$V_{i,j}$							
Row index	Column index	Set index i_{LS}							
i	j	0	1	2	3	4	5	6	7
16	26	0	0	0	0	0	0	0	0
17	1	254	158	0	48	120	134	57	196
	5	124	23	24	132	43	23	201	173
	11	114	9	109	206	65	62	142	195
	12	64	6	18	2	42	163	35	218
	27	0	0	0	0	0	0	0	0
18	0	220	186	0	68	17	173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

ตารางที่ 2.4 (ต่อ) รายละเอียดค่า $V_{i,j}$ สำหรับเมทริกซ์ฐาน 2 ของรหัส LDPC [10]

$\mathbf{H}_{BG}$		$V_{i,j}$								$\mathbf{H}_{BG}$		$V_{i,j}$							
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
	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	40	2	0	103	0	98	6	160	193	78
15	0	51	68	0	116	139	137	174	38		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
16	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
	12	242	64	143	97	8	165	176	202										

2.2.2 พื้นฐานรหัส Polar

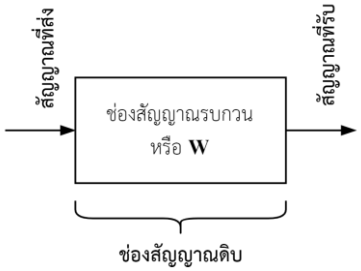
รหัสโพลาร์เป็นรหัสช่องสัญญาณที่พิสูจน์ถึงความสามารถของความจุที่เข้าใกล้ความจุช่องสัญญาณได้ ปัจจุบัน รหัสโพลาร์ถูกนำมาประยุกต์ใช้งานในระบบสื่อสารไร้สายยุคที่ 5 กรณีการสื่อสารในช่องสัญญาณควบคุม รหัสโพลาร์ประยุกต์ใช้ทฤษฎีการโพลาไรซ์ช่องสัญญาณ โดยจะทำการสังเคราะห์ช่องสัญญาณ ซึ่งประกอบไปด้วยกระบวนการรวมช่องสัญญาณ (channel combining) ที่เกิดขึ้นเมื่อทำกระบวนการเข้ารหัสโพลาร์ และการแยกช่องสัญญาณ (channel splitting) ที่จะเกิดขึ้นเมื่อดำเนินการถอดรหัสเสร็จสิ้น ความสัมพันธ์ของกระบวนการเข้ารหัสและถอดรหัสโพลาร์กับทฤษฎีการโพลาไรซ์ช่องสัญญาณสามารถแสดงได้ดังรูปที่ 2.10



รูปที่ 2.10 ช่องสัญญาณที่เกิดจากกระบวนการเข้ารหัสและถอดรหัสโพลาร์

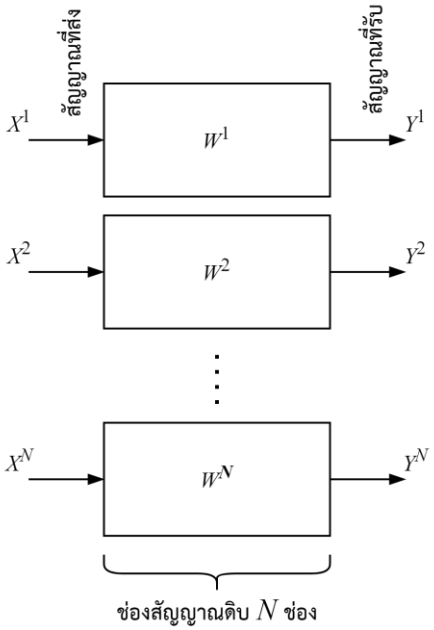
การโพลาไรซ์ช่องสัญญาณ (channel polarization) เป็นการทำให้ช่องสัญญาณรบกวนใด ๆ สามารถพิจารณาได้เป็นช่องสัญญาณรบกวนที่ประกอบไปด้วยช่องสัญญาณย่อยจำนวนมาก โดยช่องสัญญาณย่อยจะมีคุณสมบัติที่แตกต่างกัน ช่องสัญญาณย่อยบางช่องสัญญาณจะมีสัญญาณรบกวนต่ำ (noiseless subchannel) หรือมีค่าข่าวสารร่วมสูง และช่องสัญญาณย่อยบางช่องสัญญาณจะมีสัญญาณรบกวนสูง (noisy subchannel) หรือมีค่าข่าวสารร่วมต่ำ

กำหนดให้ช่องสัญญาณรบกวนแทนด้วยสัญลักษณ์ $W : X \to Y$ ในที่นี้จะเรียกช่องสัญญาณดังกล่าวว่า ช่องสัญญาณดิบ (raw channel) ดังรูปที่ 2.11 โดยสัญญาณ x ที่ถูกส่งผ่านช่องสัญญาณดิบแทนด้วยตัวแปรสุ่ม X และสัญญาณ y ที่ได้รับแทนด้วยตัวแปรสุ่ม Y กำหนดให้ความน่าจะเป็นการเปลี่ยนผ่าน (transition probability) ของช่องสัญญาณ W เขียนแทนด้วยสัญลักษณ์ $W\left(y|x\right)$



รูปที่ 2.11 บิตข้อมูลส่งผ่านช่องสัญญาณดิบ

สำหรับช่องสัญญาณดิบที่มีช่องสัญญาณจำนวน N ช่อง แสดงดังรูปที่ 2.12 ทั้งนี้ สามารถเขียนช่วยสัญญาณจำนวน N ช่อง ด้วยสัญลักษณ์ $W^N : X^N \to Y^N$ โดยสัญญาณ x_1^N จำนวน N ชุด ที่ถูกส่งผ่านช่องสัญญาณดิบ N ช่อง แทนด้วยตัวแปรสุ่ม X^N และสัญญาณ $\mathbf{y}_1^N$ จำนวน N ชุด ที่ได้รับแทนด้วยตัวแปรสุ่ม Y^N กำหนดให้ความน่าจะเป็นการเปลี่ยนผ่านของช่องสัญญาณ W^N เขียนแทนด้วยสัญลักษณ์ $W^N\left(y_1^N \middle| x_1^N\right)$



รูปที่ 2.12 ช่องสัญญาณดิบจำนวน N ช่อง

การรวมช่องสัญญาณดิบ W จำนวน N ช่องสัญญาณ สามารถดำเนินการในรูปแบบรีคลูซีฟ (recursive) โดยที่ N ต้องมีค่าเท่ากับสองยกกำลังด้วยจำนวนเต็มบวกใด ๆ โดยผลการรวมช่องสัญญาณจะสร้างช่องสัญญาณที่ถูกสังเคราะห์มาดังนี้

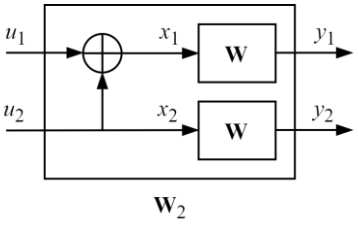
$$W_N : X^N \to Y^N \tag{2.7}$$

โดยสามารถเขียนความน่าจะเป็นการเปลี่ยนผ่านของช่องสัญญาณดิบ N ช่อง ได้โดย

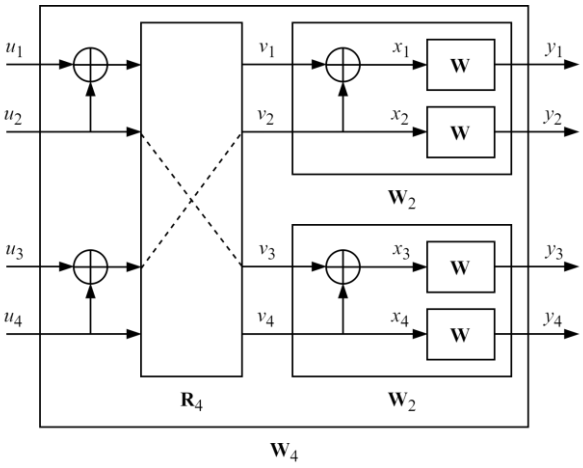
$$\begin{aligned} W_N\left(y_1^N \middle| u_1^N\right) &= W^N\left(y_1^N \middle| x_1^N\right) \\ &= W^N\left(y_1^N \middle| u_1^N G_N\right) \end{aligned} \tag{2.8}$$

จะสังเกตได้ว่าช่องสัญญาณ W_N ที่ได้จากการรวมช่องสัญญาณ W^N จะมีการนำสัญญาณ u_1^N มาดำเนินการกับเมทริกซ์ G_N ตัวอย่างเช่น การส่งข้อมูล u_1^N ผ่านช่องสัญญาณรบกวนในรูปที่ 2.13 ซึ่งเป็นการรวมช่องสัญญาณ W ที่เป็นอิสระกัน 2 ช่องเข้าด้วยกันเป็นช่องสัญญาณ W_2 กรณีนำช่องสัญญาณ W_2 จำนวน 2 ช่องสัญญาณรวมกันเพื่อสร้างช่องสัญญาณที่มีจำนวนอินพุตเท่ากับ 4 หรือ $N = 4$ จะได้ช่องสัญญาณ W_4 ดังรูปที่ 2.14 ทั้งนี้ จะต้องมีกระบวนการเรียงสับเปลี่ยน R_4 เพื่อรวมช่องสัญญาณ W_2 จำนวนช่องสัญญาณ

จากตัวอย่างการสร้างช่องสัญญาณ W_2 และ W_4 จะสังเกตได้ว่าโครงสร้างช่องสัญญาณ W_N ที่มีจำนวน $N > 2$ ขึ้นไป สามารถสร้างได้จากการรวมช่องสัญญาณ $W_{N/2}$ จำนวน 2 ช่องสัญญาณ โดยการแปลง u_N เป็น x_N สามารถแสดงด้วยเมทริกซ์กำเนิด G_N ร่วมกับเมทริกซ์การเรียงสับเปลี่ยน R_N



รูปที่ 2.13 การรวมช่องสัญญาณ W เพื่อสร้างช่องสัญญาณ W_2



รูปที่ 2.14 การรวมช่องสัญญาณ W_2 เพื่อสร้างช่องสัญญาณ W_4

การแยกช่องสัญญาณจะเป็นการพิจารณา ช่องสัญญาณ W_N ออกเป็นช่องสัญญาณย่อย $W_N^{(i)}$ จำนวน N ช่องสัญญาณ โดยช่องสัญญาณย่อย $W_N^{(i)}$ นิยามได้ดังนี้

$$W_N^{(i)} : X \to Y^N \times X^{i-1} \tag{2.9}$$

ทั้งนี้ สามารถเขียนความน่าจะเป็นการเปลี่ยนผ่านของช่องสัญญาณย่อยลำดับที่ i ได้ดังนี้

$$W_N^{(i)}\left(y_1^N, u_1^{i-1} \middle| u_i\right) \triangleq \sum_{u_{i+1}^N \in X^{N-i}} \frac{1}{2^{N-1}} W_N\left(y_1^N \middle| u_1^N\right) \tag{2.10}$$

ช่องสัญญาณ $W_N^{(i)}$ สามารถพิจารณาได้เป็นช่องสัญญาณใหม่ของบิตลำดับที่ i โดยการตัดสินใจบิตลำดับที่ i จะนำสัญญาณที่ได้รับจากช่องสัญญาณทั้งหมดและบิตข้อมูล u_1^{i-1} หรือบิตข้อมูลลำดับก่อนหน้าของบิตข้อมูลลำดับที่ i ทั้งหมด

2.3 การมอดูเลชันและการดีมอดูเลชันมาตรฐาน 5G

2.3.1 การมอดูเลชันมาตรฐาน 5G

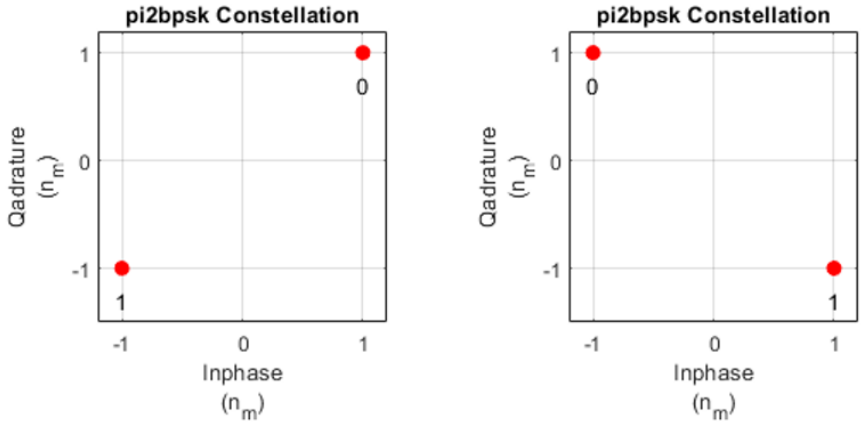
การมอดูเลชัน (Modulation) เป็นการแมปสัญญาณข่าวสารที่เป็นบิต ‘0’ หรือ ‘1’ ให้อยู่ในรูปของสัญญาณสัญญาณส่ง โดยในมาตรฐาน 5G กำหนดรูปแบบการมอดูเลชันไว้ 7 รูปแบบ [10] ดังนี้

2.3.1.1 การมอดูเลชันแบบ π/2-BPSK

กำหนดให้ $b(i)$ แทนบิตข้อมูลบิตที่ i ในสัญญาณข่าวสาร สัญลักษณ์เชิงซ้อน $d(i)$ ในสัญญาณส่งเมื่อมีการมอดูเลชันแบบ π/2-BPSK สามารถหาได้จาก

$$d(i)=\frac{e^{j\frac{\pi}{2}(i \bmod 2)}}{\sqrt{2}}\left[(1-2b(2i))+j(1-2b(2i))\right] \tag{2.11}$$

แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ π/2-BPSK แสดงดังรูปที่ 2.15 โดยหนึ่งจุดบนคอนสเตลเลชัน แทนหนึ่งสัญลักษณ์ ซึ่งในแต่ละสัญลักษณ์แทนบิตข้อมูล 1 บิต และ $n_m=\frac{1}{\sqrt{2}}$ แทนตัวปรับมาตรฐาน (Normalization Factor) สำหรับการมอดูเลชันแบบ π/2-BPSK



(ก) กรณีบิตคู่ (ข) กรณีบิตคี่

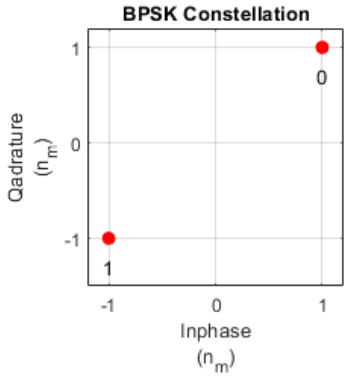
รูปที่ 2.15 แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ π/2-BPSK

2.3.1.2 การมอดูเลชันแบบ BPSK

กำหนดให้ $b(i)$ แทนบิตข้อมูลบิตที่ i ในสัญญาณข่าวสาร สัญลักษณ์เชิงซ้อน $d(i)$ ในสัญญาณส่งเมื่อมีการมอดูเลชันแบบ BPSK สามารถหาได้จาก

$$d(i) = \frac{1}{\sqrt{2}}\left[(1-2b(2i)) + j(1-2b(2i))\right] \tag{2.12}$$

แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ BPSK แสดงดังรูปที่ 2.16 โดยหนึ่งจุดบนคอนสเตลเลชัน แทนหนึ่งสัญลักษณ์ ซึ่งในแต่ละสัญลักษณ์แทนบิตข้อมูล 1 บิต และ $n_m = \frac{1}{\sqrt{2}}$ แทนตัวปรับมาตรฐาน (Normalization Factor) สำหรับการมอดูเลชันแบบ BPSK



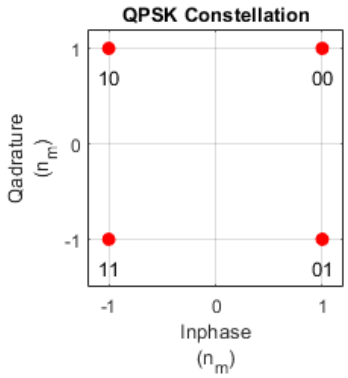
รูปที่ 2.16 แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ BPSK

2.3.1.3 การมอดูเลชันแบบ QPSK

กำหนดให้ $b(i)$ แทนบิตข้อมูลบิตที่ i ในสัญญาณข่าวสาร สัญลักษณ์เชิงซ้อน $d(i)$ ในสัญญาณส่งเมื่อมีการมอดูเลชันแบบ QPSK สามารถหาได้จาก

$$d(i) = \frac{1}{\sqrt{2}}\left[(1-2b(2i)) + j(1-2b(2i+1))\right] \tag{2.13}$$

แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ QPSK แสดงดังรูปที่รูปที่ 2.17 โดยหนึ่งจุดบนคอนสเตลเลชัน แทนหนึ่งสัญลักษณ์ ซึ่งในแต่ละสัญลักษณ์แทนบิตข้อมูล 2 บิต และ $n_m = \frac{1}{\sqrt{2}}$ แทนตัวปรับมาตรฐาน (Normalization Factor) สำหรับการมอดูเลชันแบบ QPSK



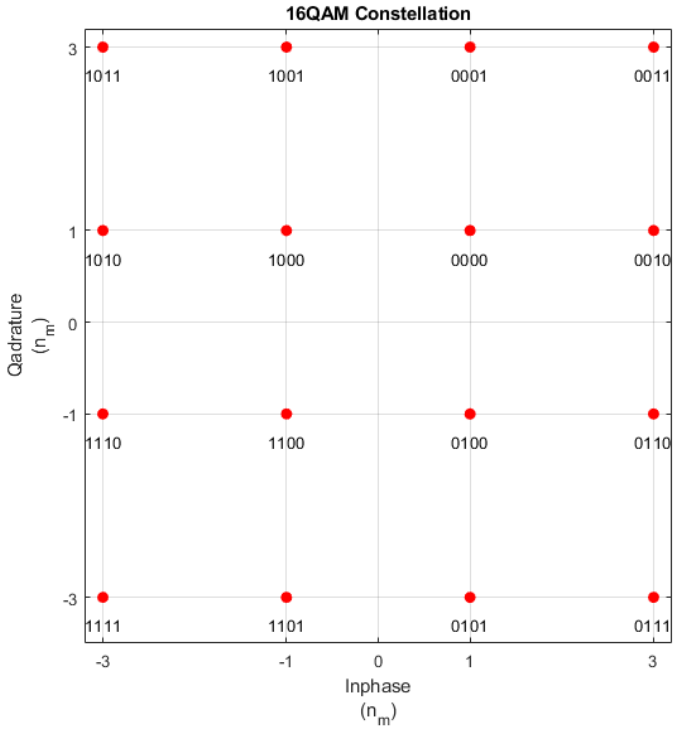
รูปที่ 2.17 แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ QPSK

2.3.1.4 การมอดูเลชันแบบ 16QAM

กำหนดให้ $b(i)$ แทนบิตข้อมูลบิตที่ i ในสัญญาณข่าวสาร สัญลักษณ์เชิงซ้อน $d(i)$ ในสัญญาณส่งเมื่อมีการมอดูเลชันแบบ 16QAM สามารถหาได้จาก

$$d(i)=\frac{1}{\sqrt{10}}\left\{(1-2b(4i))\left[2-(1-2b(4i+2))\right]+j(1-2b(4i+1))\left[2-(1-2b(4i+3))\right]\right\} \quad (2.14)$$

แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ 16QAM แสดงดังรูปที่ 2.18 โดยหนึ่งจุดบนคอนสเตลเลชันแทนหนึ่งสัญลักษณ์ ซึ่งในแต่ละสัญลักษณ์แทนบิตข้อมูล 4 บิต และ $n_m=\frac{1}{\sqrt{10}}$ แทนตัวปรับมาตรฐาน (Normalization Factor) สำหรับการมอดูเลชันแบบ 16QAM



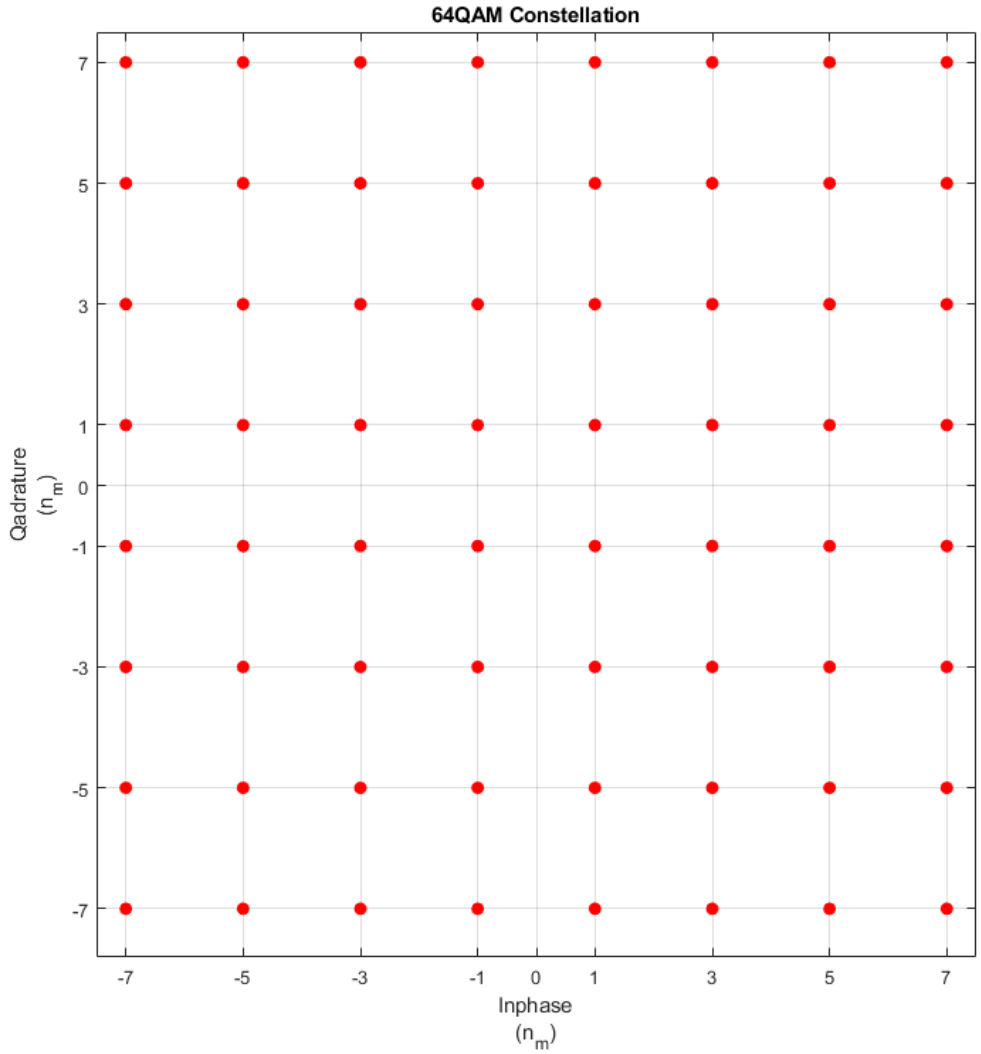
รูปที่ 2.18 แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ 16QAM

2.3.1.5 การมอดูเลชันแบบ 64QAM

กำหนดให้ $b(i)$ แทนบิตข้อมูลบิตที่ i ในสัญญาณข่าวสาร สัญลักษณ์เชิงซ้อน $d(i)$ ในสัญญาณส่งเมื่อมีการมอดูเลชันแบบ 64QAM สามารถหาได้จาก

$$\begin{aligned} d(i) = \frac{1}{\sqrt{42}}\Big\{&(1-2b(6i))\Big[4-(1-2b(6i+2))\Big[2-(1-2b(6i+4))\Big]\Big] \\ &+ j(1-2b(6i+1))\Big[4-(1-2b(6i+3))\Big[2-(1-2b(6i+5))\Big]\Big]\Big\} \end{aligned} \tag{2.15}$$

แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ 64QAM แสดงดังรูปที่ 2.19 โดยหนึ่งจุดบนคอนสเตลเลชันแทนหนึ่งสัญลักษณ์ ซึ่งในแต่ละสัญลักษณ์แทนบิตข้อมูล 6 บิต และ $n_m = \frac{1}{\sqrt{42}}$ แทนตัวปรับมาตรฐาน (Normalization Factor) สำหรับการมอดูเลชันแบบ 64QAM



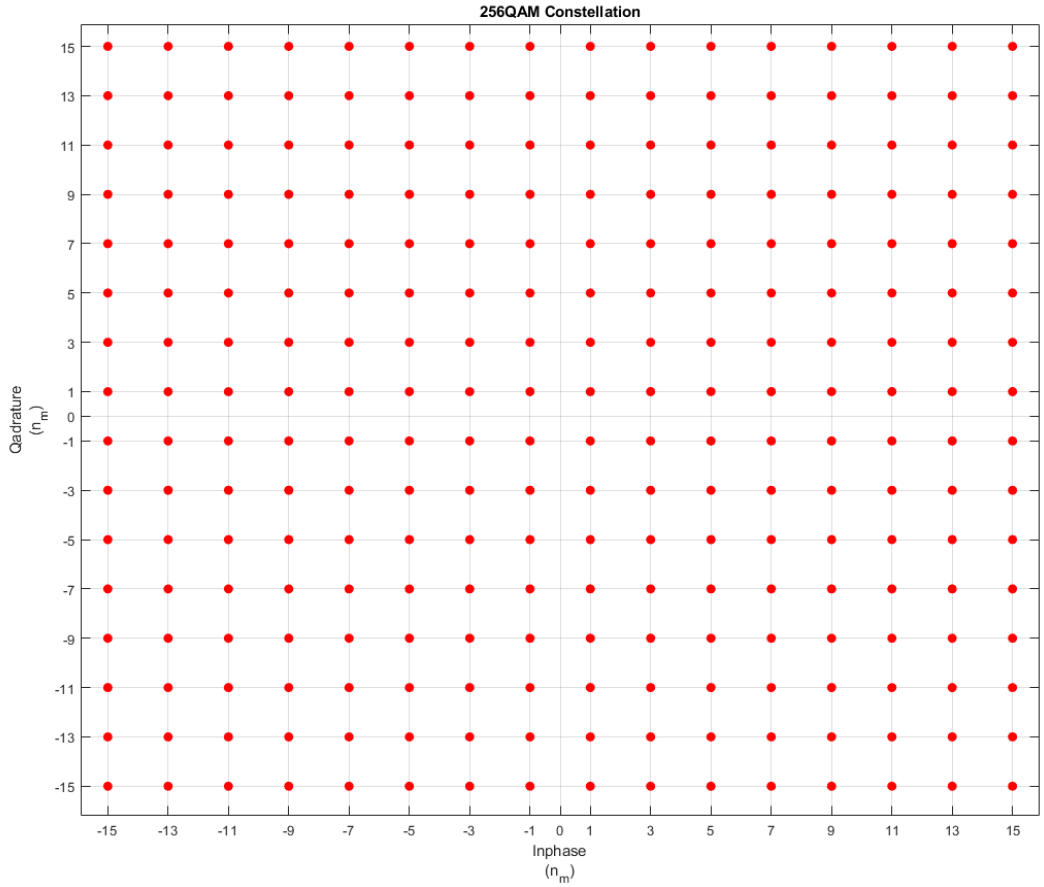
รูปที่ 2.19 แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ 64QAM

2.3.1.6 การมอดูเลชันแบบ 256QAM

กำหนดให้ $b(i)$ แทนบิตข้อมูลบิตที่ i ในสัญญาณข่าวสาร สัญลักษณ์เชิงซ้อน $d(i)$ ในสัญญาณส่งเมื่อมีการมอดูเลชันแบบ 256QAM สามารถหาได้จาก

$$d(i)=\frac{1}{\sqrt{170}}\Big\{(1-2b(8i))\Big[8-(1-2b(8i+2))\Big[4-(1-2b(8i+4))\Big[2-(1-2b(8i+6))\Big]\Big]\Big] + j(1-2b(8i+1))\Big[8-(1-2b(8i+3))\Big[4-(1-2b(8i+5))\Big[2-(1-2b(8i+7))\Big]\Big]\Big]\Big\} \quad (2.16)$$

แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ 256QAM แสดงดังรูปที่ 2.20 โดยหนึ่งจุดบนคอนสเตลเลชันแทนหนึ่งสัญลักษณ์ ซึ่งในแต่ละสัญลักษณ์แทนบิตข้อมูล 8 บิต และ $n_m = \frac{1}{\sqrt{170}}$ แทนตัวปรับมาตรฐาน (Normalization Factor) สำหรับการมอดูเลชันแบบ 256QAM



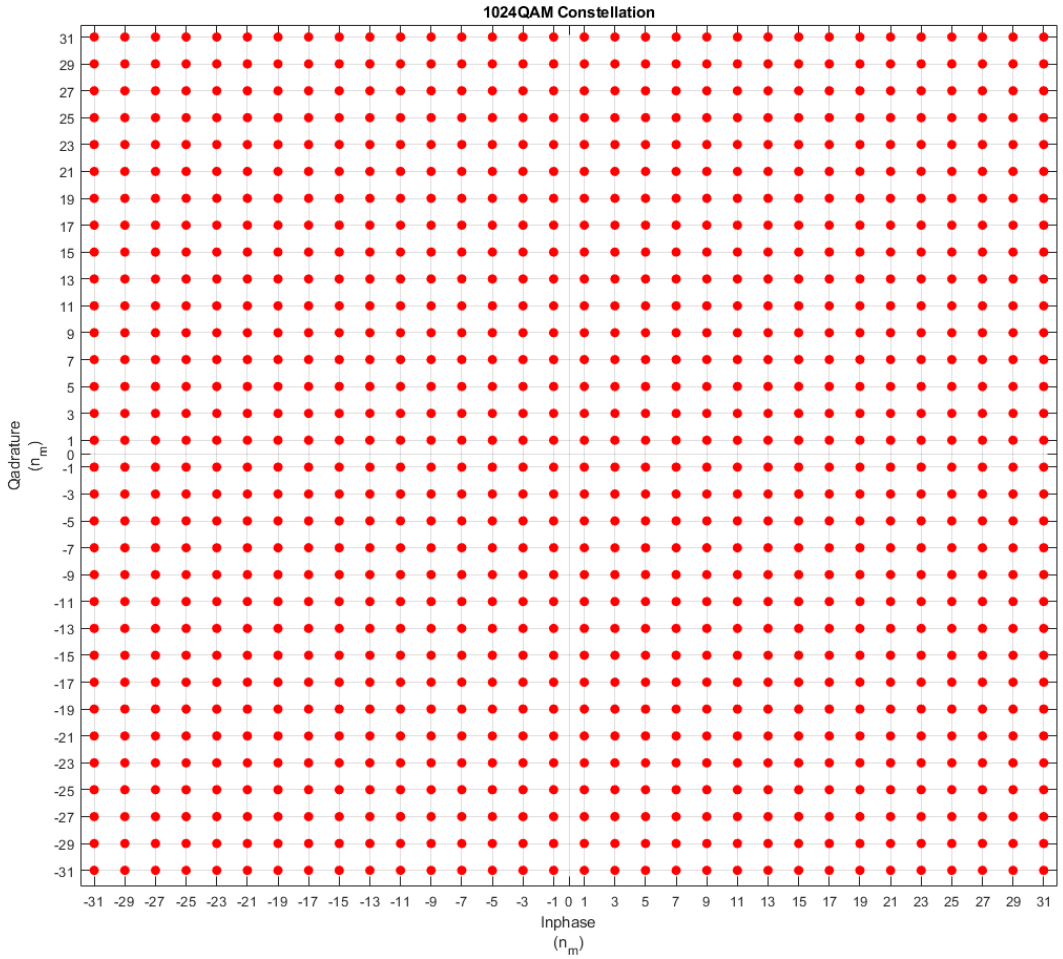
รูปที่ 2.20 แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ 256QAM

2.3.1.7 การมอดูเลชันแบบ 1024QAM

กำหนดให้ $b(i)$ แทนบิตข้อมูลบิตที่ i ในสัญญาณข่าวสาร สัญลักษณ์เชิงซ้อน $d(i)$ ในสัญญาณส่งเมื่อมีการมอดูเลชันแบบ 1024QAM สามารถหาได้จาก

$$d(i) = \frac{1}{\sqrt{682}}\Big\{(1-2b(10i))\Big[16-(1-2b(10i+2))\Big[8-(1-2b(10i+4))\Big[4-(1-2b(10i+6))\Big[2-(1-2b(10i+8))\Big]\Big]\Big]\Big] + j(1-2b(10i+1))\Big[16-(1-2b(10i+3))\Big[8-(1-2b(10i+5))\Big[4-(1-2b(10i+7))\Big[2-(1-2b(10i+9))\Big]\Big]\Big]\Big]\Big\} \quad (2.17)$$

แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ 1024QAM แสดงดังรูปที่ 2.21 โดยหนึ่งจุดบนคอนสเตลเลชันแทนหนึ่งสัญลักษณ์ ซึ่งในแต่ละสัญลักษณ์แทนบิตข้อมูล 10 บิต และ $n_m = \frac{1}{\sqrt{682}}$ แทนตัวปรับมาตรฐาน (Normalization Factor) สำหรับการมอดูเลชันแบบ 1024QAM



รูปที่ 2.21 แผนภาพคอนสเตลเลชันของการมอดูเลชันแบบ 1024QAM

2.3.2 การดีมอดูเลชันมาตรฐาน 5G

การดีมอดูเลชันจะสอดคล้องกับสัญญาณที่ถูกมอดูเลชันแต่ละแบบ สำหรับมาตรฐาน 5G เอาต์พุตที่ได้จากการดีมอดูเลชันจะถูกนำไปใช้สำหรับถอดรหัส LDPC (Low-density parity-check code) ซึ่งการถอดรหัส LDPC นั้นจำเป็นจะต้องใช้ค่า LLR (Log-likelihood ratio) $L(y)$ ซึ่งคำนวณได้จาก

$$L(y) = \log\left(\frac{p(y \mid x=1)}{p(y \mid x=0)}\right) \tag{2.18}$$

เมื่อ y คือเอาต์พุตช่องสัญญาณ x คือ อินพุตบิต และ $p(y|x)$ คือ ความน่าจะเป็นของเอาต์พุต y เมื่อทราบอินพุต x

จากสมการ (2.18) สามารถคำนวณค่า LLR สำหรับการมอดูเลชันแบบไบนารี เช่น BPSK และ $\pi/2$-BPSK ในกรณีที่มีการมอดูเลชันแบบนอนไบนารี เช่น QPSK 16QAM 64QAM 256QAM และ 1024QAM การคำนวณค่า LLR จะสามารถคำนวณได้ดังนี้

$$L_{b_i}(y) = \log\left(\frac{\sum_{s\in\{s_{b_1\ldots b_k}|b_i=1\}} p(y|b_i=s)}{\sum_{s\in\{s_{b_1\ldots b_k}|b_i=0\}} p(y|b_i=s)}\right) \tag{2.19}$$

เมื่อ b_i คือบิตที่ i ในสัญลักษณ์ s

ตัวอย่างของการคำนวณค่า LLR ของการมอดูเลชันแบบ 16QAM โดยจะมองเป็นการมอดูเลชันแบบ 4PAM (Pulse amplitude modulation) ของค่าจำนวนจริง และการมอดูเลชันแบบ 4PAM ของจำนวนจินตภาพ ดังนั้นการคำนวณค่า LLR ของ 16QAM (4 บิตต่อ 1 สัญลักษณ์) สามารถคำนวณได้จาก 4PAM (2 บิตต่อ 1 สัญลักษณ์) ของจำนวนจริงและ 4PAM (2 บิตต่อ 1 สัญลักษณ์) ของจำนวนจินตภาพรวมเป็น 4 บิตของ 16QAM การคำนวณค่า LLR บิตที่ 1 และบิตที่ 2 ของ 4PAM สามารถคำนวณได้จาก

$$L_{b_1}(y) = \log\left(\frac{p(y|b_1=s_{10})+p(y|b_1=s_{11})}{p(y|b_1=s_{00})+p(y|b_1=s_{01})}\right) \tag{2.20}$$

และ

$$L_{b_2}(y) = \log\left(\frac{p(y|b_2=s_{11})+p(y|b_2=s_{01})}{p(y|b_2=s_{00})+p(y|b_2=s_{10})}\right) \tag{2.21}$$

การคำนวณค่า LLR ของการมอดูเลชันแบบนอนไบนารีจากตัวอย่างจะพบว่าต้องใช้ผลรวมของความน่าจะเป็นที่บิตที่ i ของสัญลักษณ์เป็น 1 และ 0 ความน่าจะเป็นของสัญลักษณ์ที่อยู่ห่างจากค่า y จะมีค่าน้อยมาก ดังนั้นจึงสามารถประมาณค่า LLR จากค่าความน่าจะเป็นของสัญลักษณ์ที่ใกล้ค่า y ที่สุดได้ ซึ่งแนวคิดในการเลือกสัญลักษณ์ที่ใกล้ค่า y ที่สุดมาคำนวณ คือดูจากค่าความน่าจะเป็นของแต่ละสัญลักษณ์ สัญลักษณ์ที่ใกล้ค่า y มากที่สุดจะมีค่าความน่าจะเป็นมากที่สุดเรียกว่าการดีมอดูเลชันค่า LLR สูงสุด (Max-LLR demodulation) [11] ดังนั้นจากสมการ (2.20) จะได้

$$L_{b_i}(y) = \log\left(\frac{\max\limits_{s\in\{s_{b_1\ldots b_k}|b_i=1\}} p(y\,|\,b_i=s)}{\max\limits_{s\in\{s_{b_1\ldots b_k}|b_i=0\}} p(y\,|\,b_i=s)}\right) \tag{2.22}$$

2.4 การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G

2.4.1 การเข้ารหัส LDPC

กระบวนการเข้ารหัส LDPC ทำได้โดยการนำบิตข้อมูล $\mathbf{u}$ มาคูณเข้ากับเมทริกซ์ G ดังสมการที่ (2.23) โดยเมทริกซ์ G คือเมทริกซ์กำเนิดสำหรับการเข้ารหัสซึ่งแบ่งออกเป็น 2 ส่วน จากสมการ (2.24) จะเห็นว่าส่วนแรกคือเมทริกซ์เอกลักษณ์และส่วนที่ 2 คือเมทริกซ์พาริตี ตัวอย่างการเข้ารหัส LDPC เป็นดังสมการที่ (2.25)

$$\mathrm{c} = \mathrm{u}G \tag{2.23}$$

$$G = IP = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} \tag{2.24}$$

$$\mathrm{c} = \mathrm{u}G = \begin{bmatrix} 1 & 0 & 1 & 1 \end{bmatrix} G = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \tag{2.25}$$

อัตราส่วนของบิตข้อมูลกับจานวนบิตทั้งหมดในการหัสนั้นเรียกว่า “อัตรารหัส” (Code Rate: R) นิยามได้ดังสมการที่ (4)

2.4.2 การเข้ารหัส Polar

การเข้ารหัสโพลาร์สามารถดำเนินการทางคณิตศาสตร์ได้ดังนี้

$$\mathbf{x} = \mathbf{u}\mathbf{G}_n \tag{2.26}$$

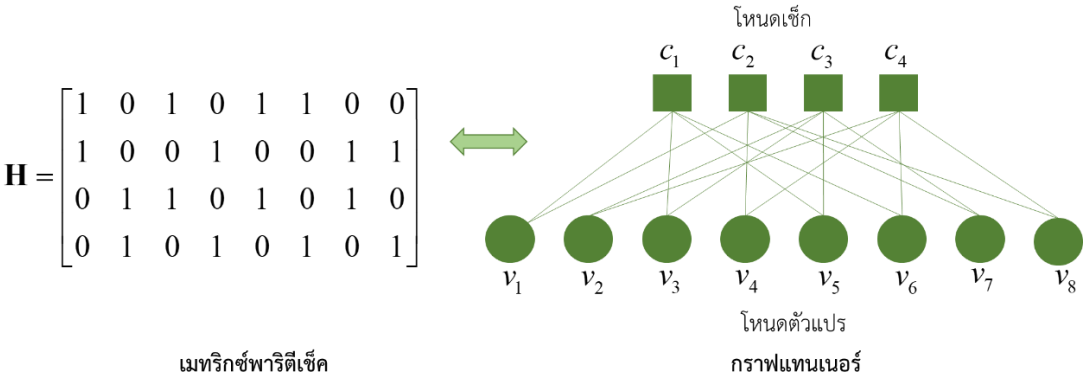
โดยที่ $\mathbf{x}$ คือคำรหัส $\mathbf{u}$ คือบิตแซ่แข็งและบิตข้อมูลที่เรียงตามลำดับความน่าเชื่อถือ ซึ่งจะกล่าวต่อไป $\mathbf{G}_n$ คือเมทริกซ์กำเนิด โดยที่ $\mathbf{G}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ และ $\mathbf{G}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$ และเวกเตอร์ $\mathbf{x}$ และ $\mathbf{u}$ มีความยาวได้เพียงสองยกกำลังใด ๆ เท่านั้น หรือ $N = 2^n$ อัตราส่วนระหว่างความยาวบิตข้อมูลและคำรหัสเขียนได้ดังนี้

$$R = K/N \tag{2.27}$$

เรียกว่าอัตรารหัส โดยที่ K คือความยาวบิตข้อมูล ดังนั้น ความยาวบิตแซ่แข็งจะเท่ากับ $N-K$

2.4.3 การถอดรดรหัสแบบการกระจายความเชื่อมั่น (BP: Belief-propagation decoder)

การถอดรหัส LDPC และรหัสโพลาร์ต้องใช้ค่า LLR ในการถอดรหัสซึ่งการคำนวณค่า LLR จากช่องสัญญาณได้อธิบายไปแล้วในหัวข้อการออกแบบการดีมอดูเลชันในมาตรฐาน 5G ซึ่งค่า LLR $L_{b_i}(y)$ จะถูกนำมาใช้ในการถอดรหัส LDPC และรหัสโพลาร์ เพื่อให้เห็นภาพการทำงานของการถอดรหัสอย่างชัดเจน กระบวนการถอดรหัส LDPC และรหัสโพลาร์ทั้งหมดจะแสดงในรูปกราฟแทนเนอร์ (Tanner graph) [7] อย่างที่กล่าวถึงไปแล้วว่ารหัส LDPC และรหัสโพลาร์สามารถแทนด้วยเมทริกซ์พาริตีเช็ค (Parity check matrix) $\mathbf{H}$ ซึ่งสามารถขียนในอยู่ในรูปของกราฟแทนเนอร์ได้ โดยที่และแถวของ $\mathbf{H}$ คือโหนดเช็ค (Check node) และแต่ละหลักของ $\mathbf{H}$ คือโหนดตัวแปร (Variable node) และเลข 1 ใน $\mathbf{H}$ แทนเส้นเชื่อมระหว่างโหนดเช็คและโหนดตัวแปร (Edge) ดังแสดงในรูปที่ 2.22



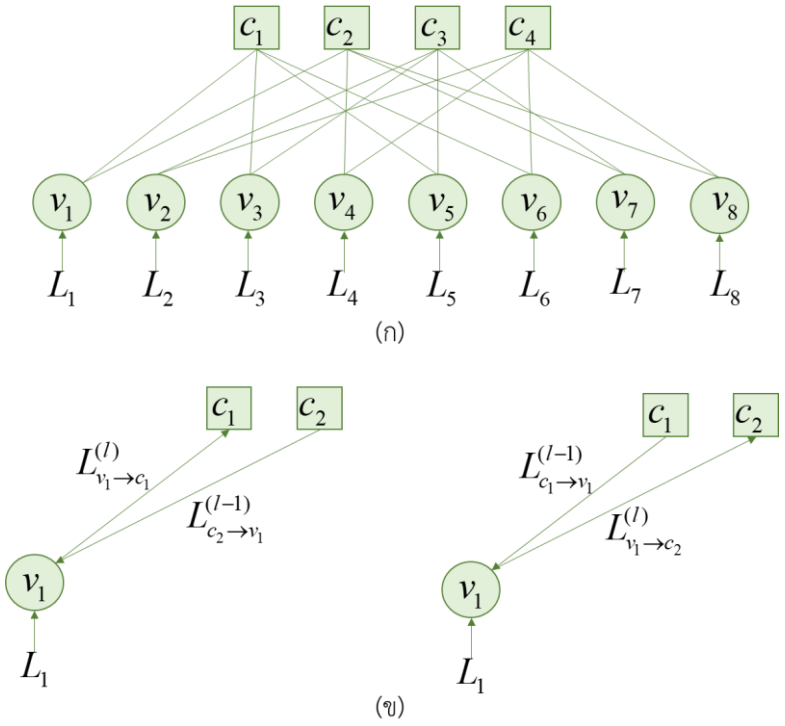
รูปที่ 2.22 การแสดงเมทริกซ์พาริตีเช็คในรูปแบบกราฟแทนเนอร์

ในขั้นตอนที่ 1 ค่า LLR $\mathbf{L} = \{L_1, L_2, ..., L_N\}$ จะถูกส่งมาคำนวณที่โหนดตัวแปรดังรูปที่ 2.23 (ก) เมื่อ N คือจำนวนโหนดตัวแปร กำหนดให้ $L^{(l)}_{v_j \to c_i}$ คือค่า LLR ที่ส่งจากโหนดตัวแปร v_j ไปยังโหนดเช็ค c_i ในการถอดรหัสรอบที่ l และ $L^{(l)}_{c_i \to v_j}$ คือค่า LLR ที่ส่งจากโหนดเช็ค c_i ไปยังโหนดตัวแปร v_j การคำนวณค่า LLR เอาต์พุตของโหนดตัวแปรจะได้จากผลรวมของค่า LLR อินพุตทั้งหมดที่เข้ามายังโหนดตัวแปรยกเว้นเส้นเชื่อมที่เป็นเอาต์พุตของโหนดตัวแปร และการคำนวณค่า LLR เอาต์พุตของโหนดตัวแปรจะต้องคำนวณทุกเส้นที่เชื่อมกับโหนดเช็ค ตัวอย่างการคำนวณแสดงในรูปที่ 2.23 (ข) การคำนวณ LLR เอาต์พุตของโหนดตัวแปร v_1 ซึ่งโหนดตัวแปรเชื่อมกับ 2 โหนดเช็คและ

มีอินพุต LLR จากช่องสัญญาณ การคำนวณค่า LLR เอาต์พุตจะต้องคำนวณ 2 ค่า ค่าแรกคือ $L^{(l)}_{v_1\to c_1}$ สามารถคำนวณได้จากผลรวมของ LLR อินพุตทั้งหมด ได้แก่ L_1 และ $L^{(l-1)}_{c_2\to v_1}$ ค่าที่ 2 คือ $L^{(l)}_{v_1\to c_2}$ สามารถคำนวณได้จากผลรวมของ LLR อินพุตทั้งหมด ได้แก่ L_1 และ $L^{(l-1)}_{c_1\to v_1}$ สำหรับโหนดตัวแปรอื่น ๆ ก็คำนวณในทำนองเดียวกัน สมการการคำนวณค่า LLR เอาต์พุตสามารถเขียนให้อยู่ในรูปทั่วไปได้เป็น

$$L^{(l)}_{v_j\to c_i} = L_j + \sum_{c_i'\in\mathcal{N}(v_j)\setminus c_i} L^{(l-1)}_{c_i'\to v_j} \qquad (2.28)$$

เมื่อ $\mathcal{N}(v_j)\setminus c_i$ คือเซตของโหนดเช็คทั้งหมดที่เชื่อมกับโหนดตัวแปร v_j ยกเว้นโหนดเช็ค c_i สำหรับการถอดรหัสในรอบที่ 1 ยังไม่มีค่า LLR จากโหนดเช็คดังนั้นจะได้ $L^{(l)}_{v_j\to c_i} = L_j$



รูปที่ 2.23 (ก) การส่งผ่านค่า LLR เข้ามาในกราฟแทนเนอร์

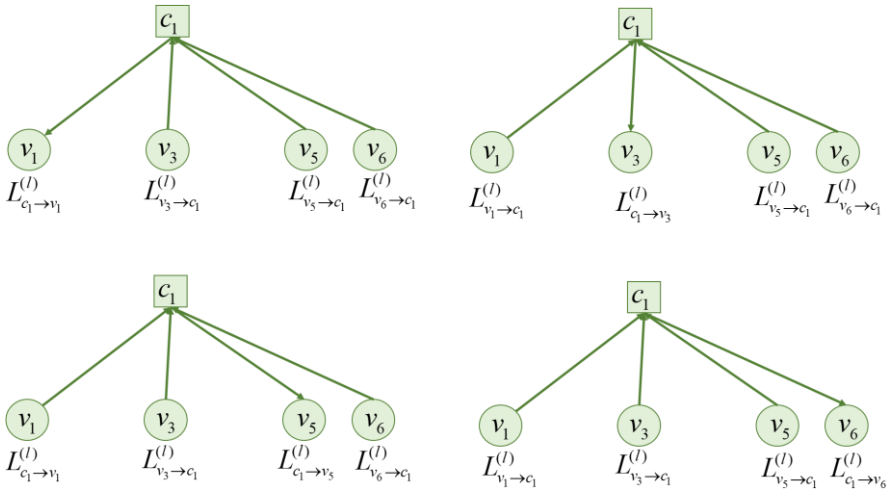
(ข) ตัวอย่างการคำนวณค่า LLR ที่โหนดตัวแปร

ขั้นตอนที่ 2 การคำนวณค่า LLR ที่โหนดเช็ค การคำนวณค่า LLR เอาต์พุตของโหนดเช็คจะได้จากการทำโอเปอเรชันของโหนดเช็คของค่า LLR อินพุตทั้งหมดที่เข้ามายังโหนดเช็คยกเว้นเส้นเชื่อมที่เป็นเอาต์พุตของโหนดเช็ค และการคำนวณค่า LLR เอาต์พุตของโหนดเช็คจะต้องคำนวณทุก

เส้นที่เชื่อมกับโหนดตัวแปร ตัวอย่างการคำนวณแสดงในรูปที่ 2.24 การคำนวณ LLR เอาต์พุตของโหนดเช็ค c_1 ซึ่งโหนดเช็คเชื่อมกับ 4 โหนดตัวแปร การคำนวณค่า LLR เอาต์พุตจะต้องคำนวณ 4 ค่า ค่าแรกคือ $L^{(l)}_{c_1\to v_1}$ สามารถคำนวณได้จากการทำโอเปอเรชันของโหนดเช็คของ LLR อินพุตทั้งหมด ได้แก่ $L^{(l)}_{v_3\to c_1}$ $L^{(l)}_{v_5\to c_1}$ และ $L^{(l)}_{v_6\to c_1}$ ค่าที่ 2 คือ $L^{(l)}_{c_1\to v_3}$ สามารถคำนวณได้จากการทำโอเปอเรชันของโหนดเช็คของ LLR อินพุตทั้งหมด ได้แก่ $L^{(l)}_{v_1\to c_1}$ $L^{(l)}_{v_5\to c_1}$ และ $L^{(l)}_{v_6\to c_1}$ ค่าที่ 3 คือ $L^{(l)}_{c_1\to v_5}$ สามารถคำนวณได้จากการทำโอเปอเรชันของโหนดเช็คของ LLR อินพุตทั้งหมด ได้แก่ $L^{(l)}_{v_1\to c_1}$ $L^{(l)}_{v_3\to c_1}$ และ $L^{(l)}_{v_6\to c_1}$ ค่าที่ 4 คือ $L^{(l)}_{c_1\to v_6}$ สามารถคำนวณได้จากการทำโอเปอเรชันของโหนดเช็คของ LLR อินพุตทั้งหมด ได้แก่ $L^{(l)}_{v_1\to c_1}$ $L^{(l)}_{v_3\to c_1}$ และ $L^{(l)}_{v_5\to c_1}$ สำหรับโหนดเช็คอื่น ๆ ก็คำนวณในทำนองเดียวกัน สมการการคำนวณค่า LLR เอาต์พุตสามารถเขียนให้อยู่ในรูปทั่วไปได้เป็น

$$L^{(l)}_{c_i\to v_j} = -2\tanh^{-1}\left(\prod_{v_j'\in\mathcal{M}(c_i)\setminus v_j}\tanh\left(L^{(l)}_{v_j'\to c_i}/2\right)\right) \tag{2.29}$$

เมื่อ $\mathcal{M}(c_i)\setminus v_j$ คือเซตของโหนดตัวแปรทั้งหมดที่เชื่อมกับโหนดเช็ค c_i ยกเว้นโหนดตัวแปร v_j



รูปที่ 2.24 ตัวอย่างการคำนวณค่า LLR ที่โหนดเช็ค

ขั้นตอนที่ 3 จะมีการคำนวณค่า LLR กลับไปกลับมาระหว่างโหนดตัวแปรและโหนดเช็คโดยใช้สมการที่ (2.28) และ (2.29) จนไม่พบบิตผิดพลาด หรือถึงจำนวนรอบการถอดรหัสที่ตั้งไว้ก็จะหยุดการถอดรหัส

ขั้นตอนที่ 4 จะเป็นการคำนวณค่า LLR ที่โหนดตัวแปรเนื่องจากในการถอดรหัสรอบสุดท้ายจะไม่มีการส่งค่า LLR ไปยังโหนดเช็คดังนั้นค่า LLR สุดท้าย L_{v_j} จะเป็นผลรวมของ LLR อินพุตทั้งหมด นั่นคือ

$$L_{v_j} = L_j + \sum_{c_i \in \mathcal{N}(v_j)} L_{c_i \to v_j}^{(l_{\max}-1)} \tag{2.30}$$

เมื่อ $\mathcal{N}(v_j)$ คือเซตของโหนดเช็คทั้งหมดที่เชื่อมกับโหนดตัวแปร v_j

ขั้นตอนที่ 5 บิตข้อมูล $\hat{x}_j$ จะถูกประมาณจากค่า LLR สุดท้าย L_{v_j} ดังนี้

$$\hat{x}_j = \begin{cases} 0, & \text{if } L_{v_j} < 0 \\ 1, & \text{otherwise} \end{cases} \tag{2.31}$$

2.4.4 การถอดรหัสแบบผลรวมค่าต่ำสุด (MS: Min-sum decoder)

การถอดรหัสแบบผลรวมค่าต่ำสุดจะมีขั้นตอนการถอดรหัสแบบเดียวกันกับการถอดรหัสแบบการกระจายความเชื่อมั่น แต่จะมีความแตกต่างกันของโอเปอเรชันโหนดเช็ค โดยการถอดรหัสแบบผลรวมค่าต่ำสุดจะคำนวณค่า LLR เอาต์พุตของโหนดเช็คจากค่า LLR ต่ำสุดของอินพุตทั้งหมด นั่นคือ

$$L_{c_i \to v_j}^{(l)} = -\text{sign}\left(\prod_{v_j' \in \mathcal{M}(c_i) \setminus v_j} L_{v_j' \to c_i}^{(l)} \right) \min_{v_j' \in \mathcal{M}(c_i) \setminus v_j} \left(| L_{v_j' \to c_i}^{(l)} | \right) \tag{2.32}$$

เมื่อ

$$\text{sign}(x) = \begin{cases} -1, & \text{if } x < 0 \\ 1, & \text{otherwise} \end{cases} \tag{2.33}$$

2.4.5 การถอดรหัสแบบผลรวมค่าต่ำสุดที่ถูกนอมัลไลซ์ (NMS: Normalized min-sum decoder)

การถอดรหัสแบบผลรวมค่าต่ำสุดที่ถูกนอมัลไลซ์จะมีขั้นตอนการถอดรหัสแบบเดียวกันกับการถอดรหัสแบบการกระจายความเชื่อมั่น แต่จะมีความแตกต่างกันของโอเปอเรชันโหนดเช็ค สมการคำนวณค่า LLR เอาต์พุตของโหนดเช็คจะคล้ายกับการถอดรหัสแบบผลรวมค่าต่ำสุด แต่จะมีการคูณค่าคงที่ α เข้าไปเพื่อนอมัลไลซ์ค่า LLR ต่ำสุดของอินพุตทั้งหมด นั่นคือ

$$L_{c_i \to v_j}^{(l)} = -\text{sign}\left(\prod_{v_j' \in \mathcal{M}(c_i) \setminus v_j} L_{v_j' \to c_i}^{(l)} \right) \min_{v_j' \in \mathcal{M}(c_i) \setminus v_j} \left(| L_{v_j' \to c_i}^{(l)} | \right) \alpha \qquad (2.34)$$

เมื่อ $0 < \alpha < 1$ สำหรับค่า α จะต้องมีการออปติไมซ์ให้เหมาะกับการถอดรหัส

2.4.6 การถอดรหัสแบบผลรวมค่าต่ำสุดที่ถูกออฟเซต (OMS: Offset min-sum decoder)

การถอดรหัสแบบผลรวมค่าต่ำสุดที่ถูกออฟเซตจะมีขั้นตอนการถอดรหัสแบบเดียวกันกับการถอดรหัสแบบการกระจายความเชื่อมั่น แต่จะมีความแตกต่างกันของโอเปอเรชันโหนดเช็ค สมการคำนวณค่า LLR เอาต์พุตของโหนดเช็คจะคล้ายกับการถอดรหัสแบบผลรวมค่าต่ำสุด แต่จะมีการลบค่าคงที่ β เข้าไปเพื่อออฟเซตค่า LLR ต่ำสุดของอินพุตทั้งหมด นั่นคือ

$$L_{c_i \to v_j}^{(l)} = -\text{sign}\left(\prod_{v_j' \in \mathcal{M}(c_i) \setminus v_j} L_{v_j' \to c_i}^{(l)} \right) \max\left(\min_{v_j' \in \mathcal{M}(c_i) \setminus v_j} \left(| L_{v_j' \to c_i}^{(l)} | \right) - \beta, 0 \right) \qquad (2.35)$$

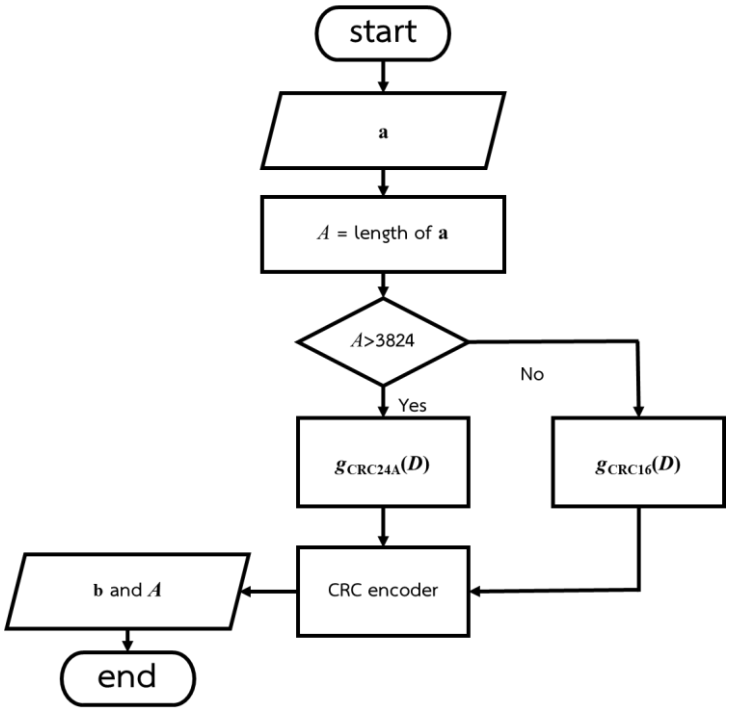
เมื่อ $0 < \beta < 1$ สำหรับค่า β จะต้องมีการออปติไมซ์ให้เหมาะกับการถอดรหัส

2.5 กระบวนทำงานฟิสิคัลเลเยอร์ลำดับสูง กรณี PDSCH/PUSCH/PSSCH

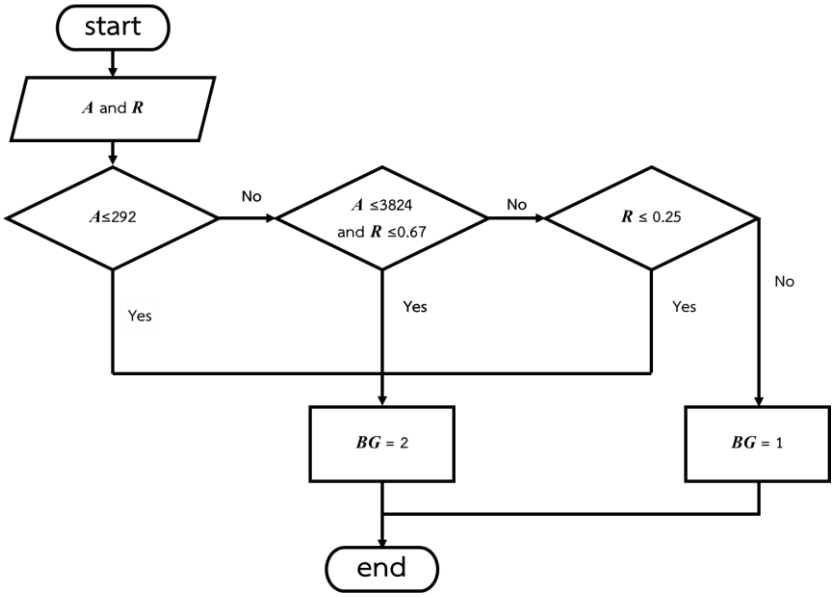
กระบวนการทำงานทำงานฟิสิคัลเลเยอร์ลำดับสูง กรณี PDSCH/PUSCH/PSSCH ประกอบด้วยโมดูลที่สำคัญทั้งหมด 7 ส่วน ภายในโมดูลหลักจะมีโมดูลย่อยที่ทำหน้าที่ประมวลผลค่าพารามิเตอร์ที่สำคัญอยู่ภายใน รายละเอียดดังนี้

2.5.1 การเข้ารหัส CRC

ในส่วนของโมดูลหลักการเข้ารหัส CRC ประกอบด้วยโมดูลย่อย 2 โมดูลได้แก่ การเข้ารหัส CRC และการเลือกกราฟฐาน (Base graph) รูปที่ 2.25 แสดงแผนภาพการทำงานของซอฟต์แวร์ เริ่มต้นจากรับบิตข้อมูล **a** ในรูปแบบเวกเตอร์ แล้วทำการคำนวณค่าความยาวของข้อมูล (A) หลังจากนั้นจะมีการตัดสินใจว่าจะเลือกใช้โพลิโนเมียลความยาว 24 ($g_{CRC24A}(D)$) หรือ 16 บิต ($g_{CRC16A}(D)$) โดยพิจารณาจากความยาวข้อมูล (A) ที่เข้ามา ถ้ามากกว่า 3824 บิต จะใช้โพลิโนเมียลความยาว 24 บิต แต่ถ้าน้อยกว่าหรือเท่ากับ 3824 บิต จะใช้โพลิโนเมียลความยาว 16 บิต ถัดมาก็จะมีการเข้ารหัส CRC โดยใช้โพลิโนเมียลที่ตรงตามเงื่อนไขคำรหัสที่ได้จะแทนด้วยเวกเตอร์ **b**



รูปที่ 2.25 แผนภาพบล็อกการเข้ารหัส CRC

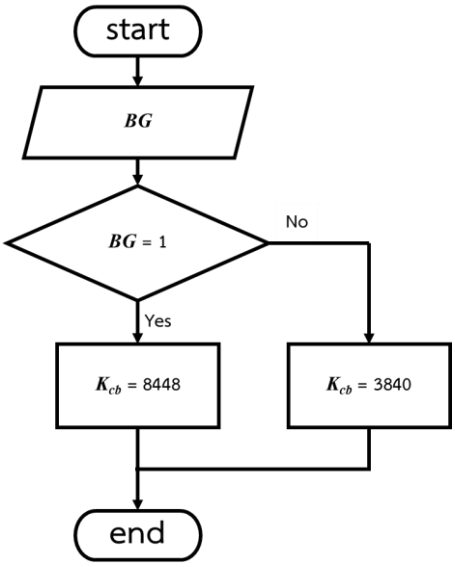


รูปที่ 2.26 แผนภาพบล็อกการเลือกกราฟฐาน

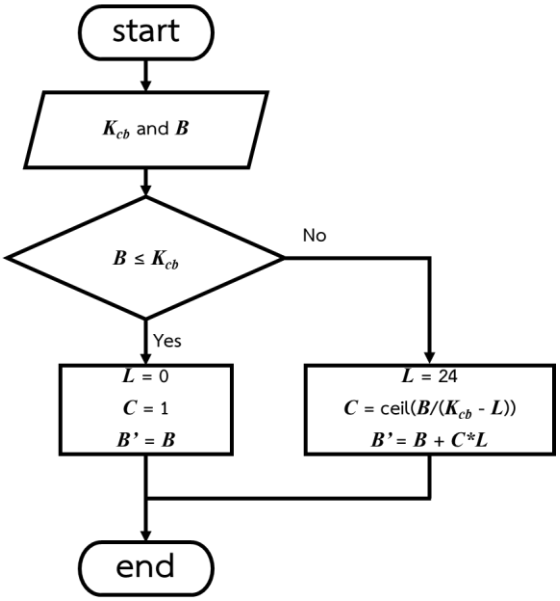
รูปที่ 2.26 แสดงแผนภาพการทำงานของซอฟต์แวร์ เริ่มจากรับค่าความยาวข้อมูล (A) ที่ได้จากแผนภาพในรูปที่ 1 และรับค่าอัตรารหัส (R) หลังจากนั้นจะตัดสินใจเลือกกราฟฐาน (BG) โดยพิจารณาจากความยาวข้อมูลและอัตรารหัสโดยมีเงื่อนไขดังนี้ ถ้าความยาวข้อมูลน้อยกว่าหรือเท่ากับ 292 บิต หรือ ความยาวข้อมูลน้อยกว่า 3824 บิต และอัตรารหัสน้อยกว่าหรือเท่ากับ 0.67 หรือ อัตรารหัสน้อยกว่าหรือเท่ากับ 0.25 จะเลือกใช้กราฟฐาน 2 ส่วนกรณีอื่นจะเลือกใช้กราฟฐาน 1

2.5.2 การแบ่งย่อยบล็อกข้อมูล

โมดูลหลักการแบ่งย่อยปรกอบด้วย 5 โมดูลย่อยได้แก่ การกำหนดค่าความยาวบล็อกรหัส การคำนวณจำนวนบล็อกย่อย การกำหนดจำนวนบิตข้อมูลของแต่ละบล็อกรหัส การคำนวณอัตรา การขยายขนาดของเมทริกซ์ฐาน และการแบ่งย่อยบล็อกข้อมูล รูปที่ 2.27 แสดงแผนภาพบล็อกการกำหนดค่าความยาวบล็อกรหัส การทำงานเริ่มจากรับค่ากราฟฐาน (BG) ที่ได้จากรูปที่ 2.26 หลังจากนั้นจะกำหนดจำนวนบิตข้อมูลในแต่ละบล็อกรหัส (K_{cb}) โดยพิจารณาจากกราฟฐาน (BG) ถ้ากราฟฐานเท่ากับ 1 จะกำหนด K_{cb} เป็น 8448 บิต ถ้าเป็นกราฟฐานเป็น 2 จะกำหนด K_{cb} เป็น 3840 บิต



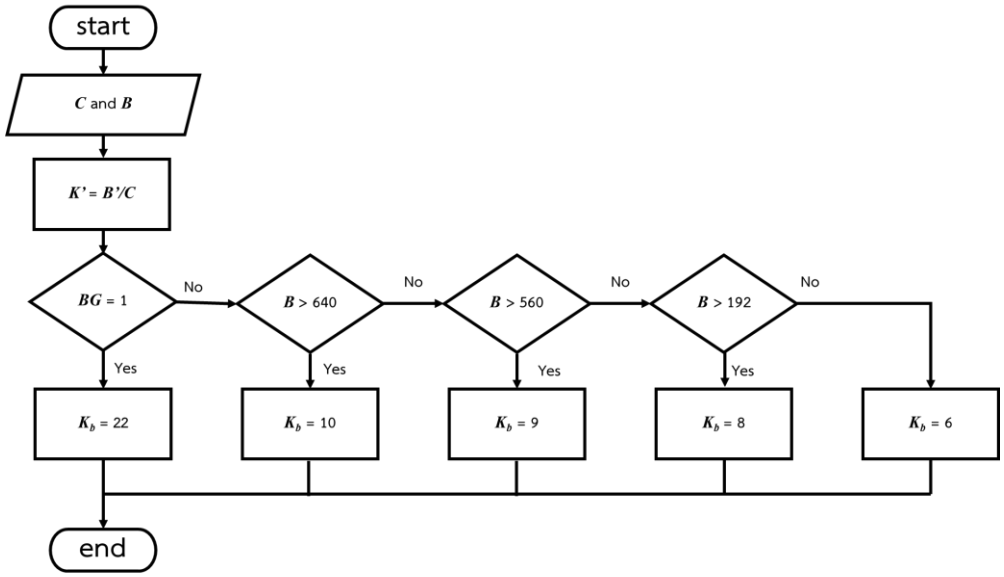
รูปที่ 2.27 แผนภาพบล็อกการกำหนดค่าความยาวบล็อกรหัส



รูปที่ 2.28 แผนภาพบล็อกการคำนวณจำนวนบล็อกย่อย

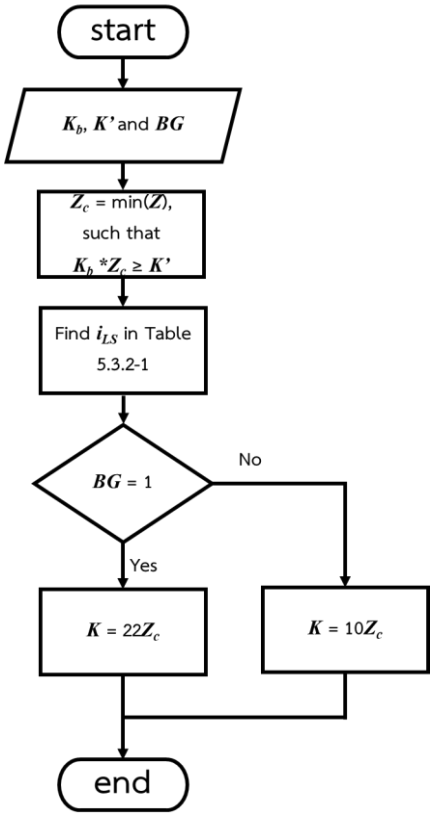
รูปที่ 2.28 แสดงแผนภาพบล็อกการคำนวณจำนวนบล็อกย่อย การทำงานเริ่มจากรับค่าความยาวคำรหัส (B) ที่ได้จากรูปที่ 2.25 และจำนวนบิตข้อมูลในแต่ละบล็อกรหัส (K_{cb}) ที่ได้จากรูปที่ 2.27 ถ้าความยาวคำรหัส (B) น้อยกว่าหรือเท่ากับจำนวนบิตข้อมูลในแต่ละบล็อกรหัส (K_{cb}) ที่กำหนดไว้ จำนวนบล็อกรหัส (C) จะมีบล็อกรหัสเพียง 1 บล็อก จำนวนบิตพาริตีของ CRC (L) จะ

เท่ากับ 0 บิต และจำนวนบิตข้อมูลทั้งหมด (B') จะเท่าความยาวคำรหัส (B) แต่ถ้าความยาวคำรหัส (B) มากกว่าจำนวนบิตข้อมูลในแต่ละบล็อกรหัส (K_{cb}) ที่กำหนดไว้ จำนวนบล็อกรหัส (C) จะถูกคำนวณโดย $\lceil B/(K_{cb}-L) \rceil$ จำนวนบิตพาริตีของ CRC (L) จะเท่ากับ 24 บิต ส่งผลให้จำนวนบิตข้อมูลทั้งหมด (B') จะเท่ากับ ($B+C*L$)



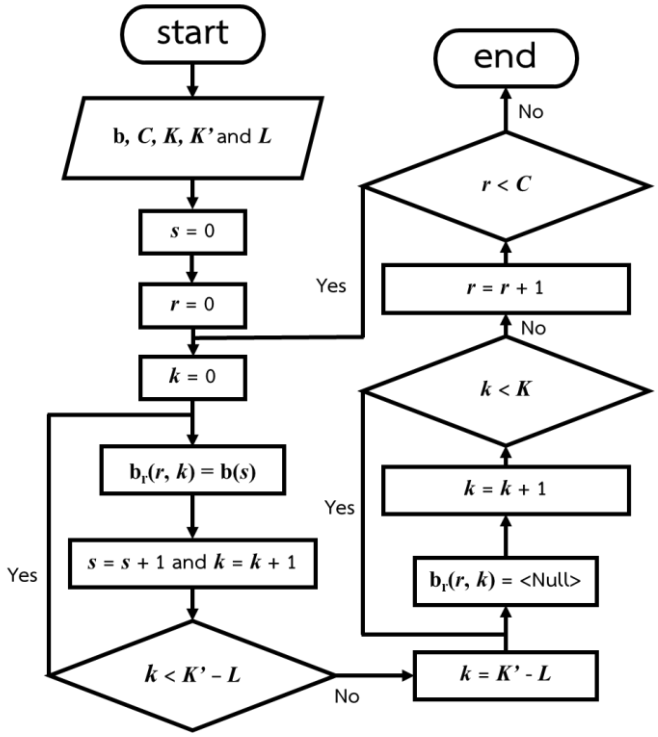
รูปที่ 2.29 แผนภาพบล็อกการกำหนดจำนวนบิตข้อมูลของแต่ละบล็อกรหัส

รูปที่ 2.29 แสดงแผนภาพบล็อกการกำหนดจำนวนบิตข้อมูลของแต่ละบล็อกรหัส การทำงานเริ่มจากรับค่าความยาวคำรหัส (B) ที่ได้จากรูปที่ 2.25 และจำนวนบล็อกรหัส (C) ที่ได้จากรูปที่ 2.28 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 2.30 แผนภาพบล็อกการคำนวณอัตราการขยายขนาดของเมทริกซ์ฐาน

รูปที่ 2.30 แสดงแผนภาพบล็อกการคำนวณอัตราการขยายขนาดของเมทริกซ์ฐาน การทำงานเริ่มจากรับค่าจำนวนบิตข้อมูลที่ต้องการในแต่ละบล็อกรหัส (K_b) จำนวนบิตข้อมูลจริงแต่ละบล็อกรหัส (K') จากรูปที่ 2.29 และกราฟฐาน (BG) ที่ได้จากรูปที่ 2.26 ถัดมาคำนวณหาค่าอัตราการขยายขนาดของเมทริกซ์ฐาน (Z_c) โดยได้จากการหาค่าต่ำสุดของอัตราการขยายขนาดของเมทริกซ์ฐาน (Z) ในตารางที่ 2.2 โดยต้องสอดคล้องกับเงื่อนไขบังคับ $K_b \cdot Z_c \geq K'$ หลังจากนั้นตรวจสอบค่า i_{LS} ของอัตราการขยายขนาดของเมทริกซ์ฐาน (Z_c) ที่ได้ ถัดมาจะพิจารณาค่าความยาวบิตข้อมูลหลังการขยายขนาดเมทริกซ์ฐาน (K) ถ้าเลือกใช้กราฟฐาน 1 ค่าความยาวบิตข้อมูลหลังการขยายขนาดเมทริกซ์ฐานคือ $22Z_c$ บิต แต่ถ้าเลือกใช้การฟฐาน 2 ค่าความยาวบิตข้อมูลหลังการขยายขนาดเมทริกซ์ฐาน (K) คือ $10Z_c$ บิต



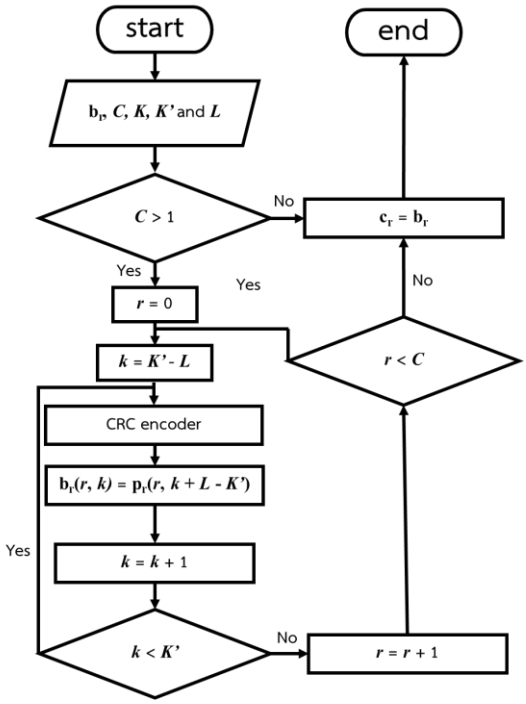
รูปที่ 2.31 แผนภาพบล็อกการแบ่งย่อยบล็อกข้อมูล

รูปที่ 2.31 แสดงแผนภาพบล็อกการแบ่งย่อยบล็อกข้อมูล การทำงานเริ่มจากรับค่าเวกเตอร์คำรหัส $\mathbf{b}$ จากโมดูลหลักการเข้ารหัส CRC จำนวนบล็อกรหัส (C) ความยาวบิตข้อมูลหลังการขยายขนาดเมทริกซ์ฐาน (K) จำนวนบิตข้อมูลจริงแต่ละบล็อกรหัส (K') และจำนวนบิตพาริตีของ CRC (L) จากรูปที่ 2.28 หลังจากนั้นจะเป็นการนำบิตข้อมูลในเวกเตอร์ $\mathbf{b}$ ไปเก็บในแต่ละบล็อกรหัสย่อย $\mathbf{b}_r$ ในตำแหน่งที่ 0 ถึง $K'-L-1$ ส่วนในตำแหน่งที่ $K'-L$ ถึง K จะเป็นบิตว่าง <NULL>

2.5.3 การเข้ารหัส CRC ของบล็อกรหัส

โมดูลการเข้ารหัส CRC ครั้งที่ 2 จะเป็นการเข้ารหัส CRC ให้กับบล็อกรหัส รูปที่ 2.32 แสดงแผนภาพบล็อกการเข้ารหัส CRC ของบล็อกรหัส การทำงานเริ่มจากรับค่าเวกเตอร์บล็อกรหัส $\mathbf{b}_r$ จำนวนบล็อกรหัส (C) ความยาวบิตข้อมูลหลังการขยายขนาดเมทริกซ์ฐาน (K) จำนวนบิตข้อมูลจริงแต่ละบล็อกรหัส (K') และจำนวนบิตพาริตีของ CRC (L) ถัดมาจะพิจารณาเงื่อนไขการเข้ารหัส CRC ของบล็อกรหัสดังนี้ ถ้าบล็อกรหัสย่อยมีเพียง 1 บล็อกจะไม่มีการเข้ารหัส CRC และเวกเตอร์บล็อกรหัสอันใหม่ $\mathbf{c}_r$ จะกำหนดให้เท่ากับ $\mathbf{b}_r$ ในกรณีที่จำนวนบล็อกรหัสย่อยมีมากกว่า 1 บล็อกจะทำการเข้ารหัส CRC ให้กับบล็อกรหัสทีละบล็อกจะได้เวกเตอร์ $\mathbf{p}_r$ หลังจากนั้นจะนำบิตพาริ

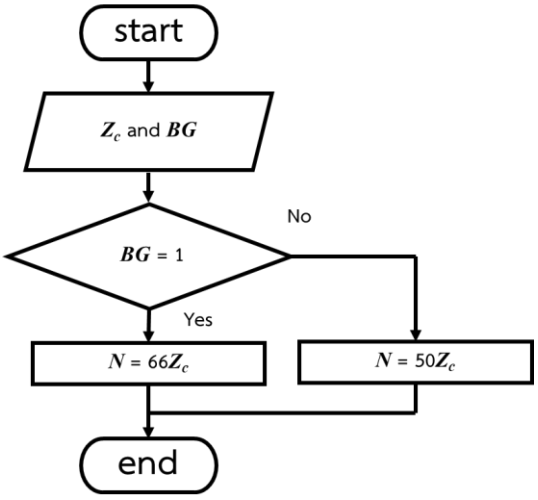
ตีมาในตำแหน่ง $K'-L$ ถึง K' เติมให้กับเวกเตอร์ $\mathbf{b}_r$ ในตำแหน่งที่ $K'-L$ ถึง K' และในขั้นตอนสุดท้ายเวกเตอร์บล็อกรหัสอันใหม่ $\mathbf{c}_r$ จะกำหนดให้เท่ากับ $\mathbf{b}_r$



รูปที่ 2.32 แผนภาพบล็อกการเข้ารหัส CRC ของบล็อกรหัส

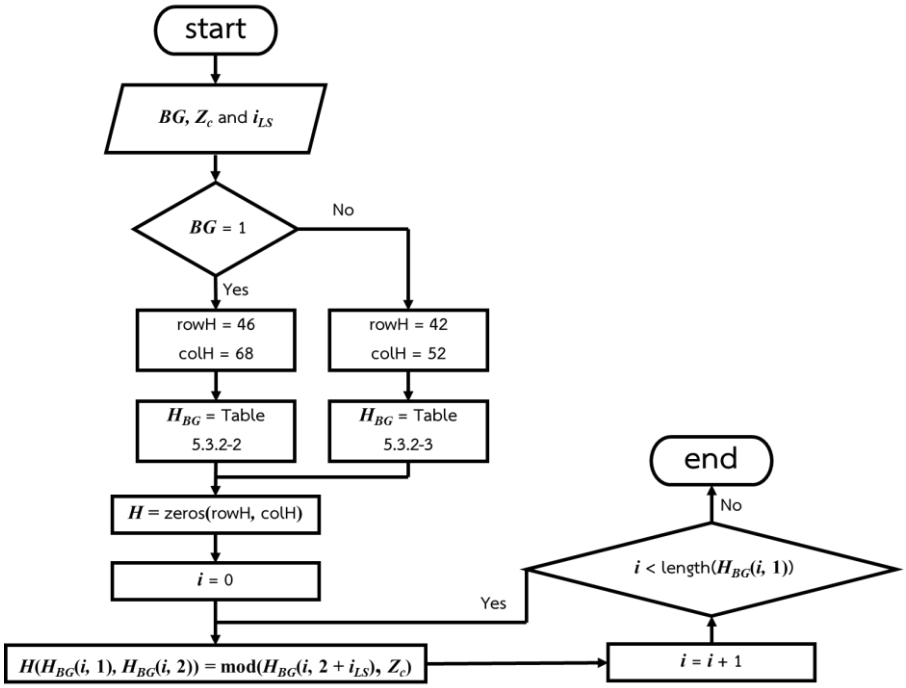
2.5.4 การเข้ารหัส LDPC

โมดูลหลักการเข้ารหัส LDPC ประกอบไปด้วย 4 โมดูลย่อยได้แก่ การกำหนดความยาวคำรหัส การสร้างเมทริกซ์พาริตีเช็ก H การเติมบิตข้อมูลในบล็อกรหัส และ การเข้ารหัส LDPC



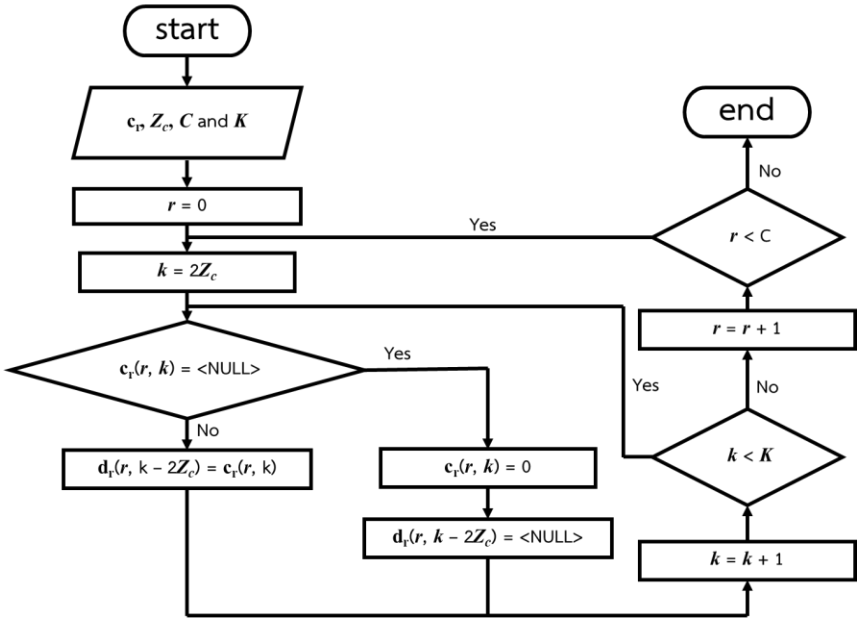
รูปที่ 2.33 แผนภาพบล็อกการกำหนดความยาวคำรหัส

รูปที่ 2.33 แสดงแผนภาพบล็อกการกำหนดความยาวคำรหัส การทำงานเริ่มจากรับค่า Z_c และ BG หลังจากนั้นจะพิจารณาเงื่อนไขดังนี้ ถ้าเลือกใช้กราฟฐาน 1 ความยาวคำรหัส N จะเท่ากับ $66Z_c$ บิต ถ้าเลือกใช้กราฟฐาน 2 ความยาวคำรหัส N จะเท่ากับ $50Z_c$ บิต

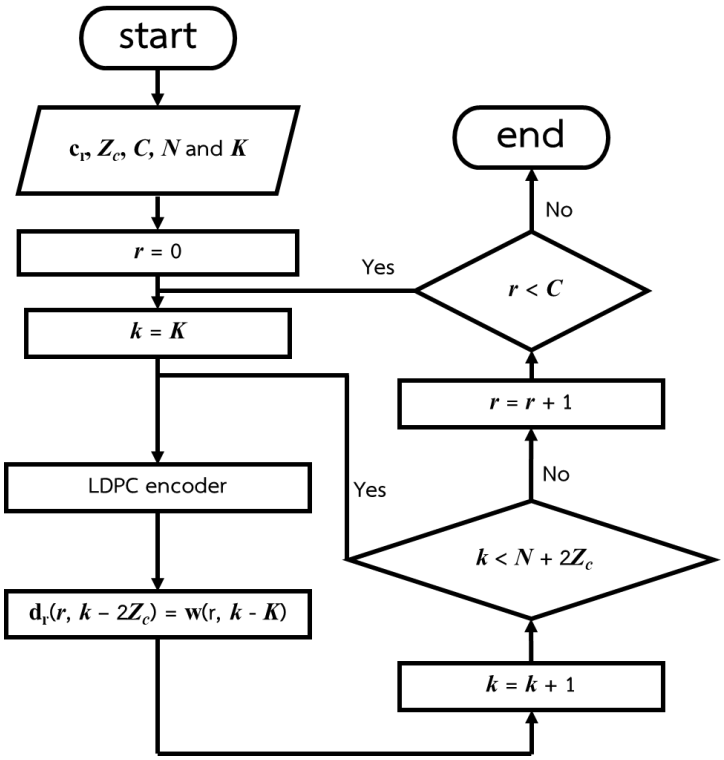


รูปที่ 2.34 แผนภาพบล็อกการสร้างเมทริกซ์พาริตีเช็ก $\mathbf{H}$

รูปที่ 2.34 แสดงแผนภาพบล็อกการสร้างเมทริกซ์พาริตีเช็ก $\mathbf{H}$ การทำงานเริ่มจากรับค่า BG Z_c และ i_{LS} ถัดมาเป็นการกำหนดขนาดขอมเมทริกซ์ $\mathbf{H}$ และการเลือกใช้ตารางการสร้าง $\mathbf{H}$ โดยพิจารณาจากเงื่อนไขดังนี้ ถ้าเลือกใช้กราฟฐาน 1 ขนาดของเมทริกซ์ $\mathbf{H}$ คือ 46×68 และเลือกใช้ค่าในตารางที่ 2.3 ถ้าเลือกใช้กราฟฐาน 2 ขนาดของเมทริกซ์ $\mathbf{H}$ คือ 42×52 และเลือกใช้ค่าในตารางที่ 2.4 หลังจากนั้นจะสร้างเมทริกซ์ $\mathbf{H}$ ให้มีค่า 0 ทั้งหมดโดยมีขนาดที่กำหนดตามเงื่อนไขข้างต้น หลังจากนั้นทำการเติมค่าของเมทริกซ์ในตำแหน่งที่กำหนดในตาราง โดยค่าจำนวนเต็มในตารางจะสอดคล้องกับ Z_c รูปที่ 2.35 แสดงแผนภาพบล็อกการเติมบิตข้อมูลในบล็อกรหัส การทำงานเริ่มจากรับค่าเวกเตอร์ $\mathbf{c}_r$ Z_c C และ K ถัดมาเป็นการเติมบิตข้อมูล $\mathbf{c}_r$ ลงไปในเวกเตอร์ $\mathbf{d}_r$ ในตำแหน่งที่ $2Z_c$ ถึง K เนื่องจากข้อมูลใน 2 หลักแรกจะไม่ถูกส่งออกไป ขั้นตอนถัดมาจะเป็นการพิจารณาค่าในเวกเตอร์ $\mathbf{c}_r$ ถ้าบิตในตำแหน่งดังกล่าวเป็นบิตว่าง <NULL> จะเติมบิต 0 ลงไปแทน



รูปที่ 2.35 แผนภาพบล็อกการเติมบิตข้อมูลในบล็อกรหัส

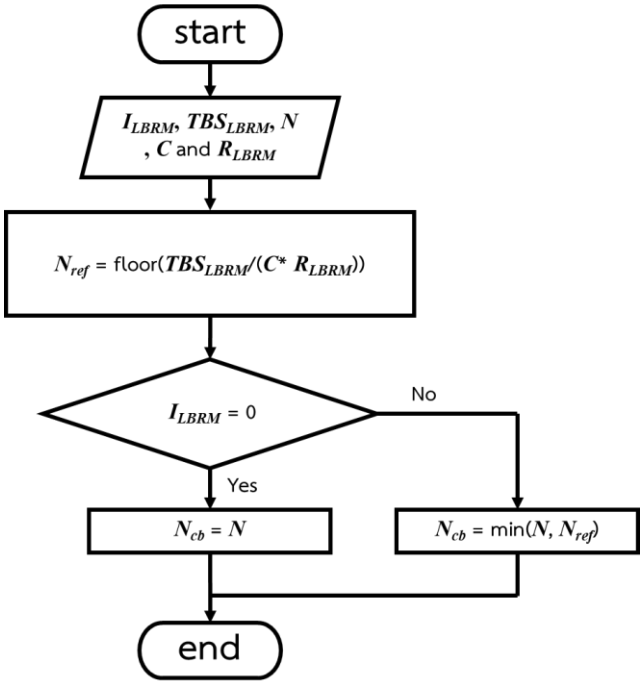


รูปที่ 2.36 แผนภาพบล็อกการเข้ารหัส LDPC

รูปที่ 2.36 แสดงแผนภาพบล็อกการเข้ารหัส LDPC การทำงานเริ่มจากรับค่าเวกเตอร์ $\mathbf{c}_r$ Z_c C N และ K ถัดมาเป็นกระบวนการเข้ารหัส LDPC แต่ละบล็อกรหัสจะได้คำรหัส $\mathbf{w}$ และนำบิตพาริตีในเวกเตอร์ $\mathbf{w}$ เติมในเวกเตอร์ในตำแหน่งที่ K ถึง $N+2Z_c$

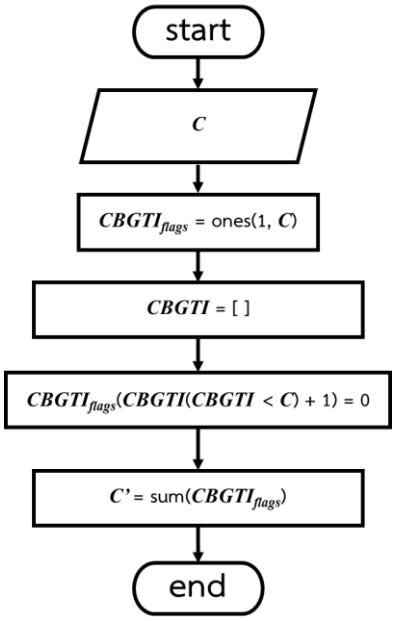
2.5.5 การปรับอัตรารหัส

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



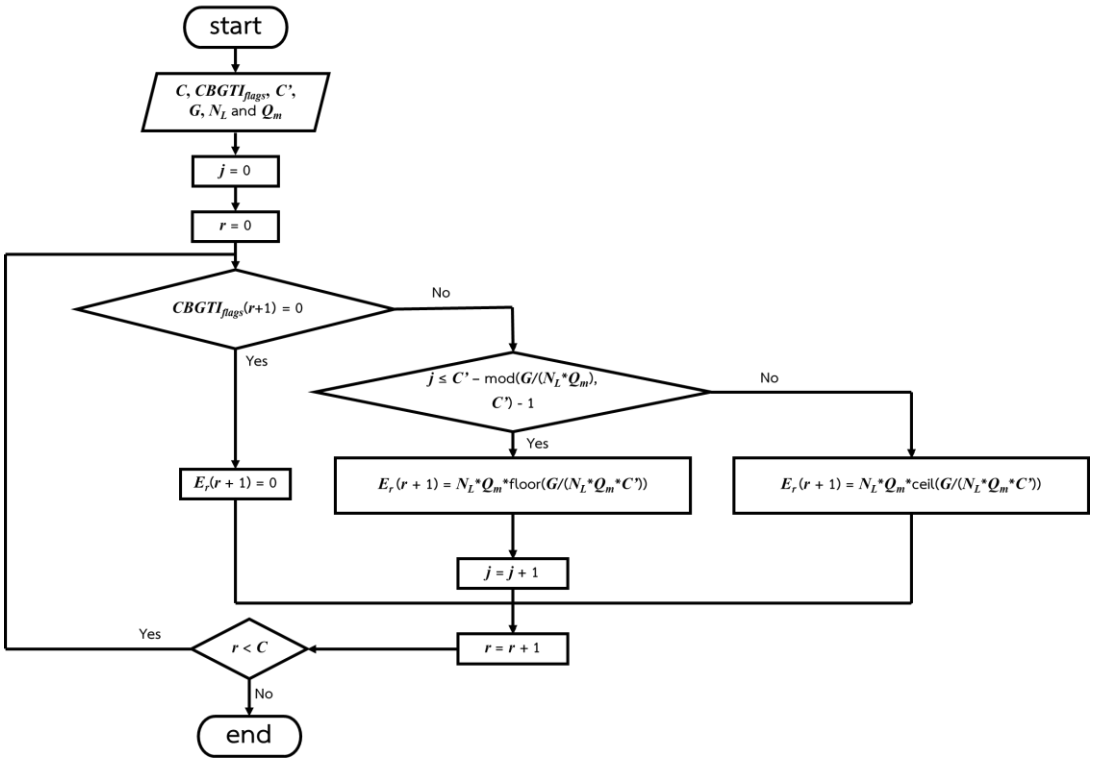
รูปที่ 2.37 แผนภาพบล็อกการกำหนดความยาวคำรหัสของบล็อกรหัส

รูปที่ 2.37 แสดงแผนภาพบล็อกการกำหนดความยาวคำรหัสของบล็อกรหัส การทำงานเริ่มจากรับค่า I_{LBRM} TBS_{LBRM} N C และ R_{LBRM} ถัดมาคำนวณ N_{ref} หลังจากนั้นพิจารณาเงื่อนไขดรรชนีของบัฟเฟอร์ที่ถูกจำกัด 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}



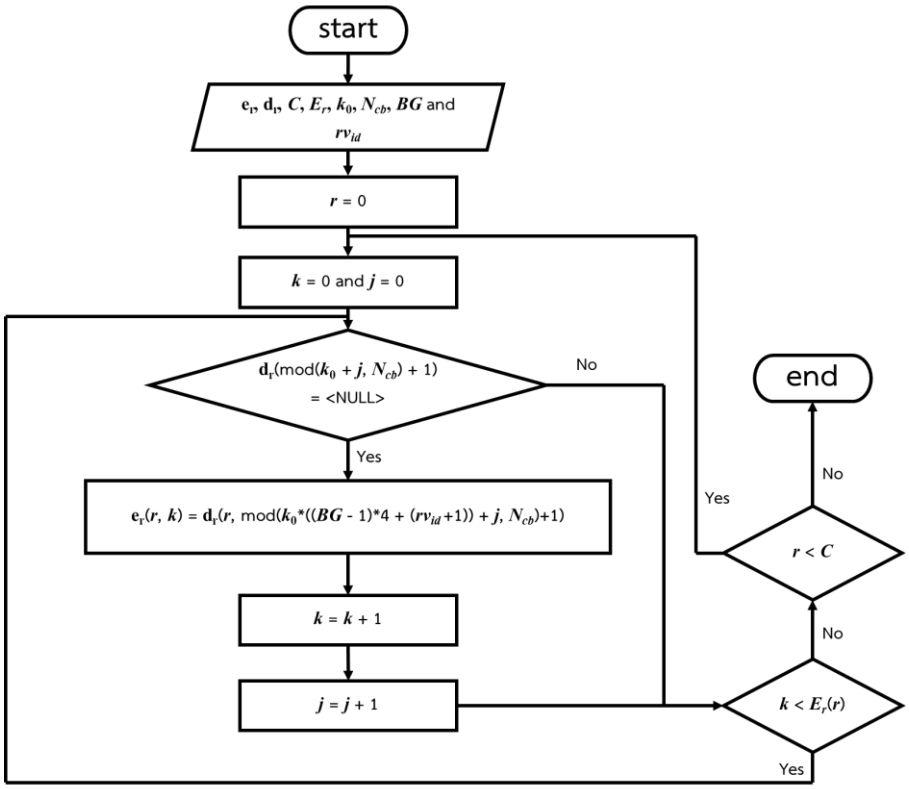
รูปที่ 2.38 แผนภาพบล็อกการคำนวณข้อมูลการส่งกลุ่มของบล็อกรหัส ($CBGTI$)

รูปที่ 2.38 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 2.39 แผนภาพบล็อกการคำนวณความยาวข้อมูลเอาต์พุตของกระบวนการปรับอัตรารหัส

รูปที่ 2.39 แสดงแผนภาพบล็อกการคำนวณความยาวข้อมูลเอาต์พุตของกระบวนการปรับอัตรารหัส การทำงานเริ่มจากรับค่า 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 $j \le C' - \mathrm{mod}(G / N_L \cdot Q_m, C') - 1$ ถ้าเป็นไปตามเงื่อนไขจะคำนวณค่า E_r ตามสมการที่ (11) ถ้าไม่เป็นไปตามเงื่อนไขจะคำนวณค่า E_r ตามสมการที่ (12)

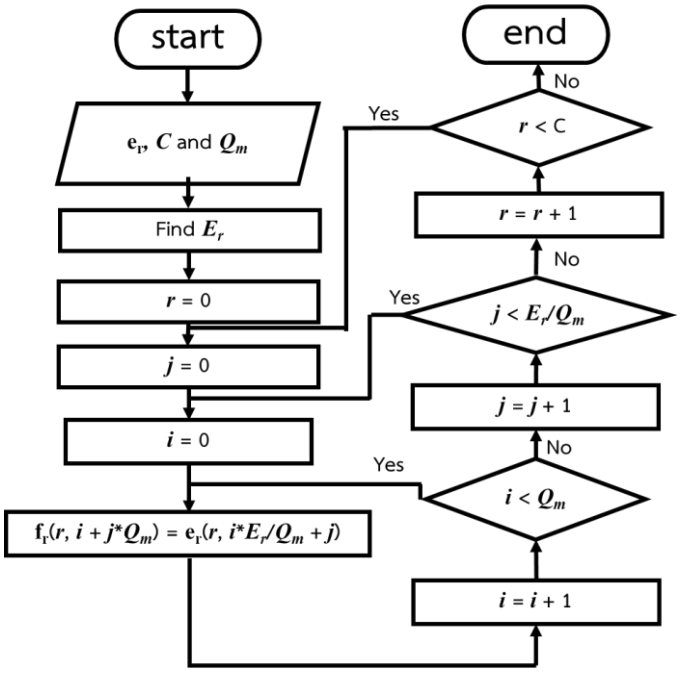


รูปที่ 2.40 แผนภาพบล็อกการเลือกบิตเพื่อส่ง

รูปที่ 2.40 แสดงแผนภาพบล็อกการเลือกบิตเพื่อส่ง การทำงานเริ่มจากรับค่า $\mathbf{e}_r$ $\mathbf{d}_r$ C E_r k_0 N_{cb} BG และ rv_{id} ถัดมา พิจารณาเงื่อนไขดังนี้ ถ้าเวกเตอร์ $\mathbf{d}_r$ เป็นบิตว่าง <NULL> จะนำค่าบิตในตำแหน่ง $\mathrm{mod}(k_0(4(BG-1)+rv_{id}+1+j, N_{cb}))$ มาใส่ในเวกเตอร์ $\mathbf{e}_r$ ถ้าเวกเตอร์ $\mathbf{d}_r$ ไม่ใช่บิตว่าง <NULL> จะไม่ทำอะไรกับข้อมูล

2.5.6 การแทรกสลับบิต

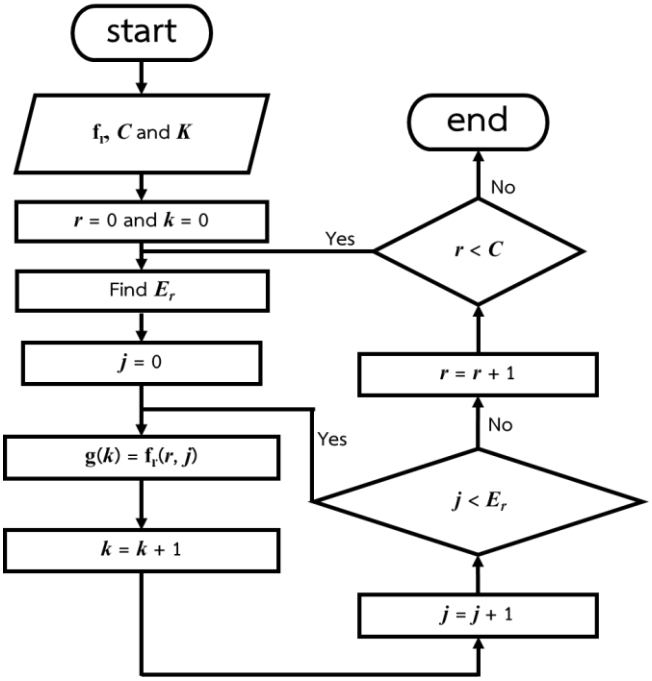
รูปที่ 2.41 แสดงแผนภาพบล็อกการแทรกสลับบิต การทำงานเริ่มจากรับค่าเวกเตอร์ $\mathbf{e}_r$ C และ Q_m ถัดมาคำนวณความยาวข้อมูล E_r หลังจากนั้นจะแทรกสลับบิตข้อมูลโดยนำบิตข้อมูลของ $\mathbf{e}_r$ ในตำแหน่ง $i*E_r/Q_m + j$ ไปใส่ในเวกเตอร์ $\mathbf{f}_r$ ในตำแหน่ง $i + j*Q_m$



รูปที่ 2.41 แผนภาพบล็อกการแทรกสลับบิต

2.5.7 การต่อเรียงบล็อกรหัส

รูปที่ 2.42 แสดงแผนภาพบล็อกการต่อเรียงบล็อกรหัส การทำงานเริ่มจากรับค่าเวกเตอร์ $\mathbf{f}_r$ C และ K ถัดมาคำนวณค่าความยาวบล็อกรหัส E_r หลังจากนั้นบล็อกรหัสย่อยจะถูกนำไปรียงต่อกันในเวกเตอร์ $\mathbf{g}$ ตามลำดับ



รูปที่ 2.42 แผนภาพบล็อกการต่อเรียงบล็อกรหัส

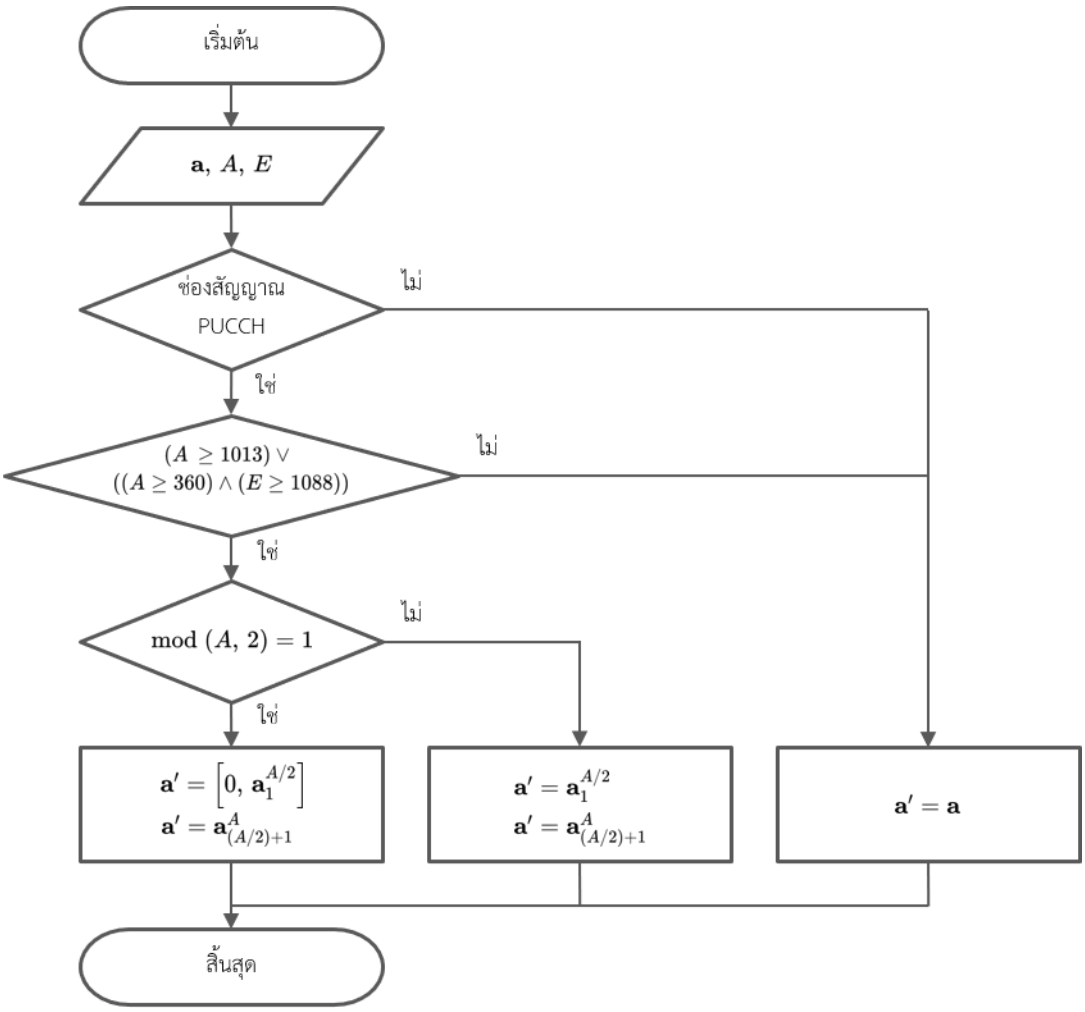
2.6 กระบวนทำงานฟิสิคัลเลเยอร์ลำดับสูง กรณี PUCCH/PDCCH/PBCH/PSCCH/PSFCH/ PSBCH

กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูง กรณี PUCCH/PDCH/PBCH/PSCCH/PSFCH/ PSBCH บนอุปกรณ์ FPGA จะถูกแบ่งเป็นโมดูลหลัก 5 โมดูล ซึ่งจะประกอบไปด้วยโมดูลย่อยทั้งหมด 10 โมดูล อธิบายรายละเอียดของซอฟต์แวร์แต่ละโมดูลย่อยในโมดูลหลักภายใต้หัวข้อย่อยดังต่อไปนี้

1) การแบ่งย่อยบล็อกรหัส

โมดูลการแบ่งย่อยบล็อกรหัส ภายใต้โมดูลหลัก CodeBlockSegmentation

การแบ่งย่อยบล็อกรหัสแบ่งข้อมูลขาเข้าเป็นสองส่วน โดยเวกเตอร์ ความยาว A จะถูกแบ่งครึ่งออกเป็นสองเวกเตอร์ $\mathbf{a}'$ ความยาว A' โมดูลจะทำงานเฉพาะช่องสัญญาณ PUCCH ในกรณีที่ $A \geq 1013 \vee \left(A \geq 360 \wedge E \geq 1088\right)$ สำหรับกรณีอื่นและช่องสัญญาณ PBCH และ PDCCH โมดูลจะไม่ทำงานและแทนเวกเตอร์ $\mathbf{a}' = \mathbf{a}$ ที่ความยาว $A' = A$ แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 2.43



รูปที่ 2.43 แผนผังการทำงานการแบ่งย่อยบล็อกรหัส

2) การเข้ารหัส การสแครมบลิง และการแทรกสลับ CRC

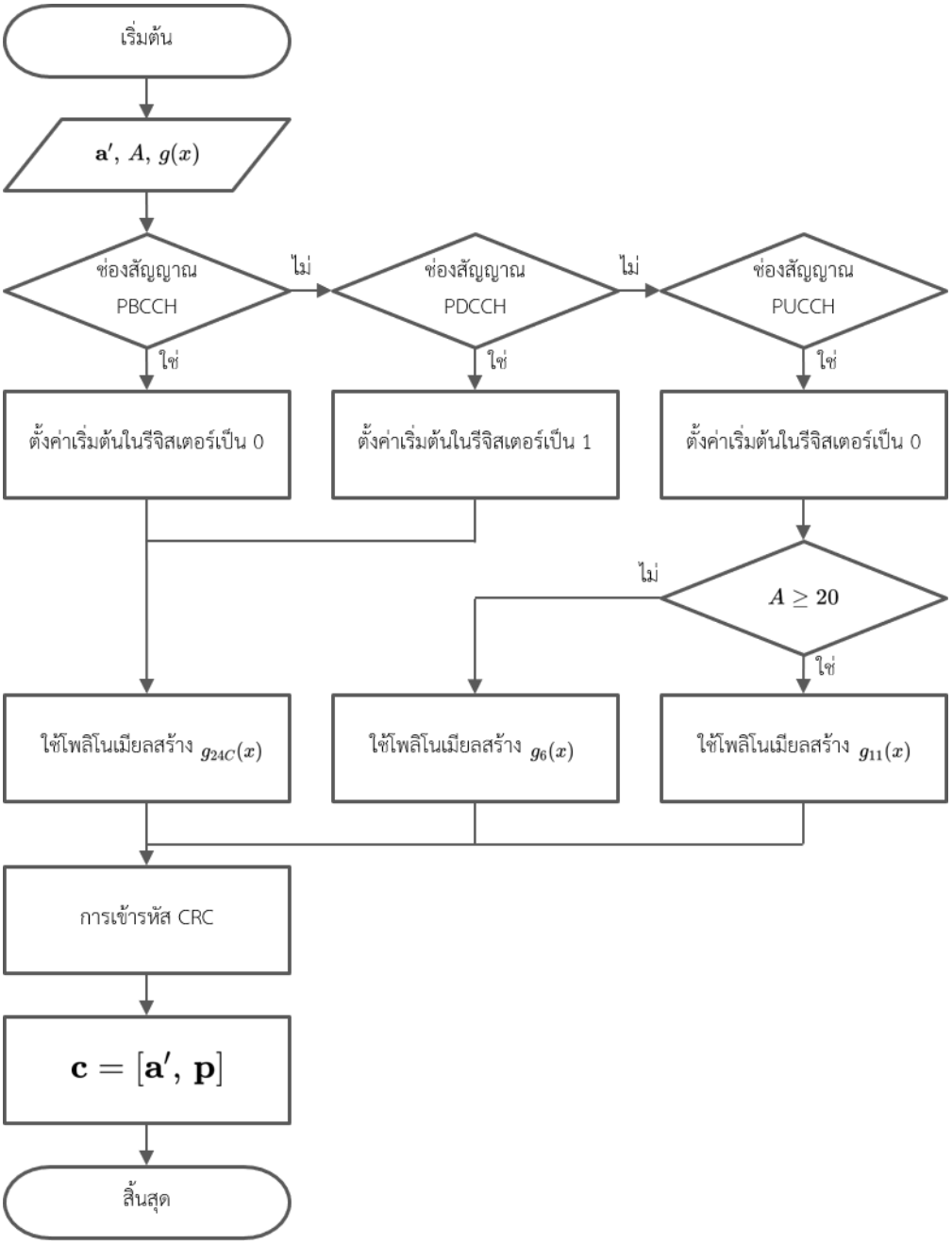
โมดูลการเข้ารหัส การสแครมบลิง และการแทรกสลับ CRC ภายใต้โมดูลย่อย CRCAttachment

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 2.44

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

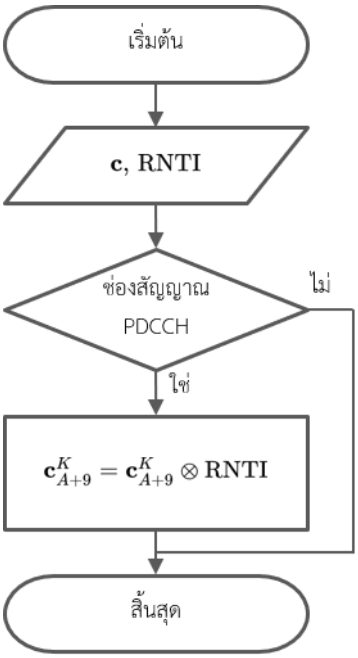
สำหรับช่องสัญญาณ PUCCH หากความยาว $A \geq 20$ จะเลือกใช้โพลิโนเมียลสร้าง $g_{11}(x)$ ที่มีความยาว $L = 11$ และหากความยาว $A \leq 19$ จะเลือกใช้โพลิโนเมียลสร้าง $g_6(x)$ ที่มีความยาว $L = 6$



รูปที่ 2.44 แผนผังการทำงานการเชื่อมบิต CRC ท้ายข้อมูล

2.2) การสแครมบลิง CRC

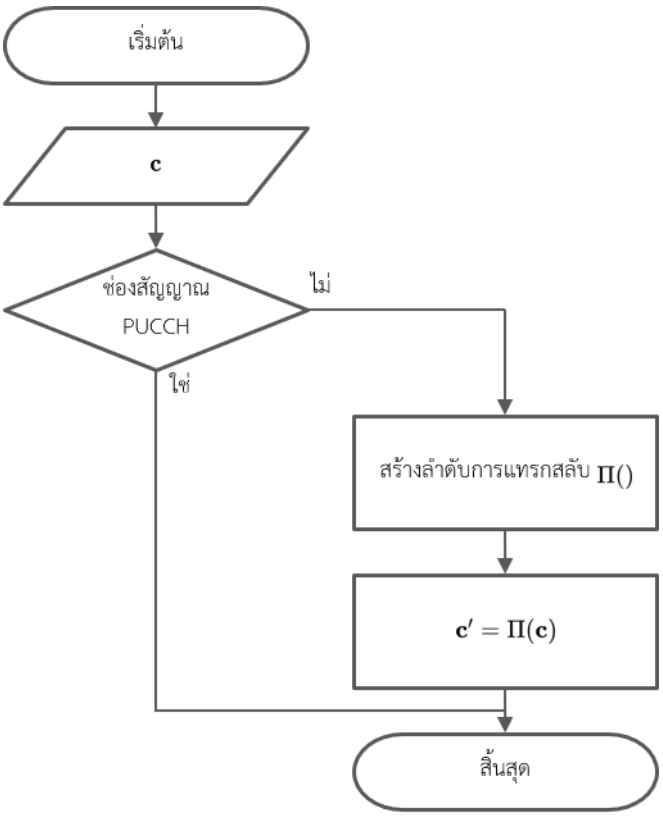
การสแครมบลิง CRC จะนำบิต RNTI ความยาว 16 บิตมาทำกระบวนการเอ็กซ์คลูซีฟออร์กับบิต CRC 16 บิตสุดท้ายหรือเวกเตอร์ $\mathbf{c}_{A'+9}^{K}$ แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 2.45



รูปที่ 2.45 แผนผังการทำงานการสแครมบลิง CRC

2.3) การแทรกสลับ CRC

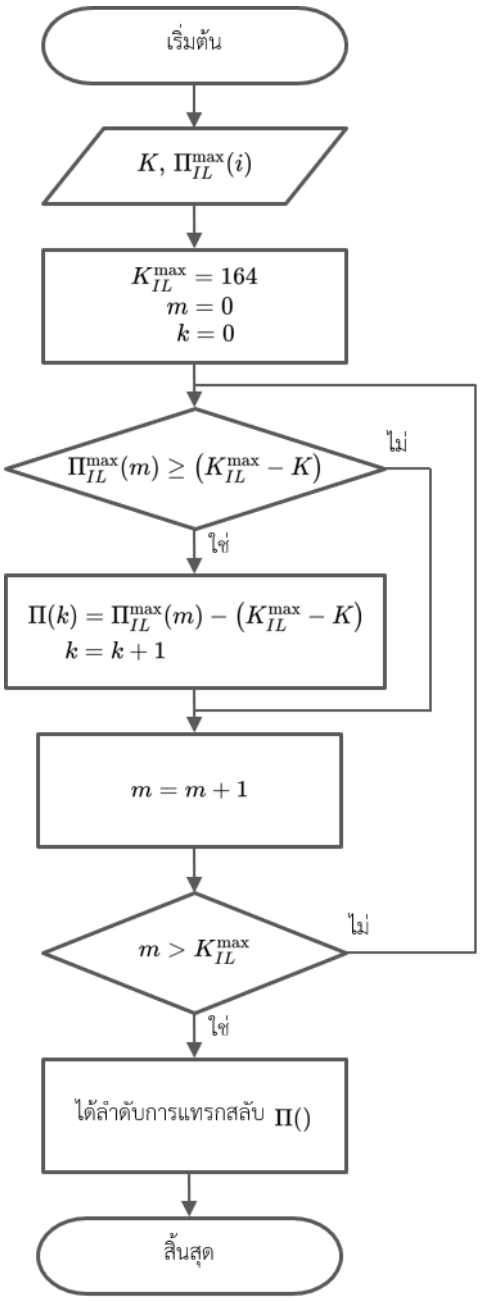
การแทรกสลับ CRC จะนำเวกเตอร์ $\mathbf{c}$ ที่ผ่านการเข้ารหัส CRC หรือการสแครมบลิงมาแทรกสลับตำแหน่งบิต ได้ผลลัพธ์เป็นเวกเตอร์ $\mathbf{c}'$ ที่มีความยาว K ที่มีความยาวเท่ากับเวกเตอร์ $\mathbf{c}$ แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 2.46 โมดูลจะทำงานเฉพาะช่องสัญญาณ PBCH และ PDCCH ที่มีการเข้ารหัส CRC ด้วยโพลิโนเมียลสร้าง $g_{24C}(x)$ สำหรับช่องสัญญาณ PUCCH โมดูลจะไม่ทำงาน



รูปที่ 2.46 แผนผังการทำงานการแทรกสลับ CRC

2.4) การสร้างลำดับการแทรกสลับ CRC

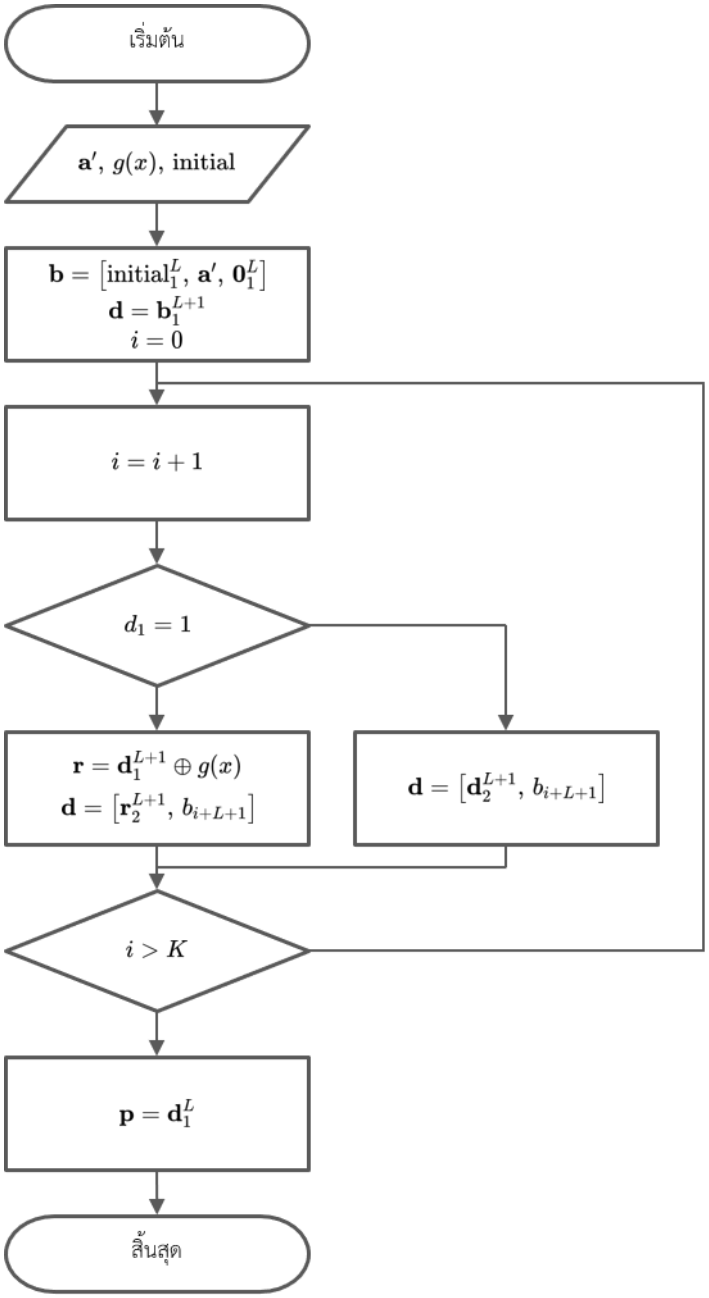
ลำดับการแทรกสลับ CRC ถูกสร้างโดยอัลกอริทึมที่ได้ระบุไว้ตามมาตรฐานร่วมกับลำดับการแทรกสลับ แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 2.47



รูปที่ 2.47 แผนผังการทำงานการสร้างลำดับการแทรกสลับ CRC

2.5) การเข้ารหัส CRC

การเข้ารหัส CRC จะเข้ารหัส CRC แก่เวกเตอร์ $\mathbf{a}'$ จากกระบวนการก่อน โดยโมดูลจะเลือกโพลิโนเมียลสร้างในการเข้ารหัส CRC จามเงื่อนไขช่องสัญญาณและความยาว A การเข้ารหัส CRC สามารถดำเนินการได้โดยการใช้งาน cyclic shift register ที่มีการตั้งค่าเริ่มต้นในรีจิสเตอร์เป็น 0 ทั้งหมดยกเว้นช่องสัญญาณ PDCCH ที่ตั้งค่าเริ่มต้นในรีจิสเตอร์เป็น 1 แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 2.48



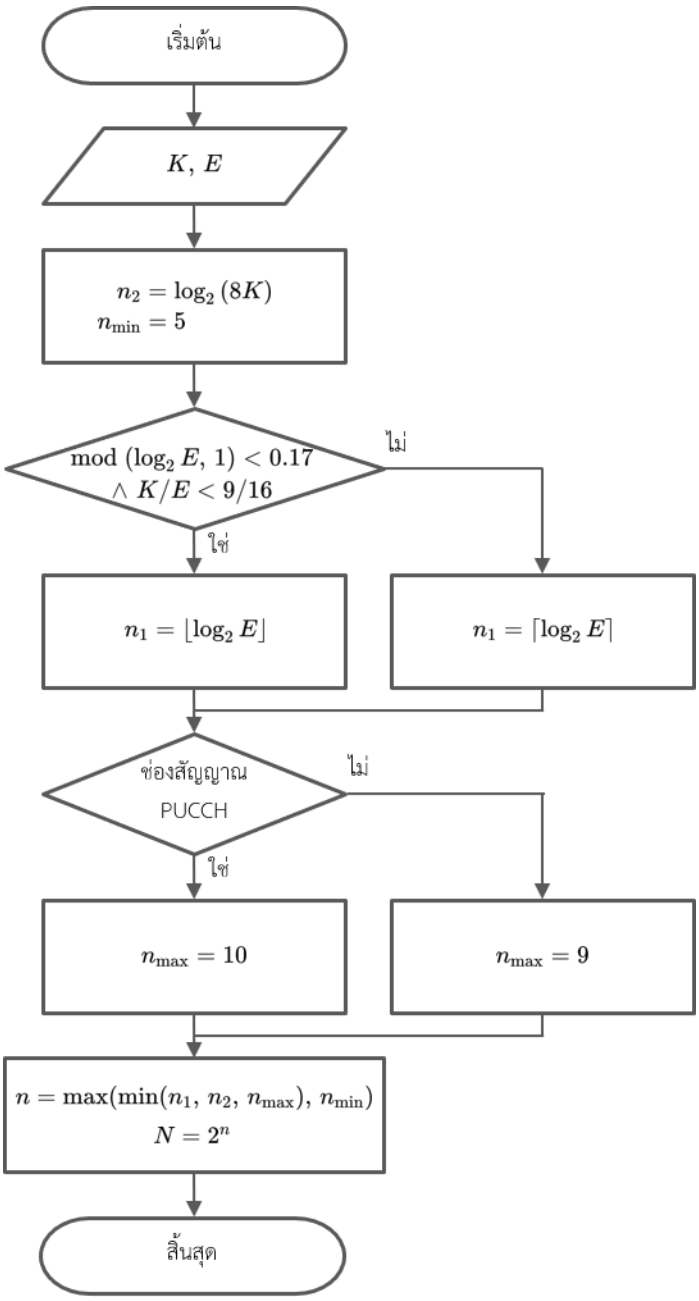
รูปที่ 2.48 แผนผังการทำงานการเข้ารหัส CRC

3) การลำดับช่องสัญญาณย่อยและการคำนวณบิตพาริตี

โมดูลการลำดับช่องสัญญาณย่อยและการคำนวณบิตพาริตี ภายใต้โมดูลหลัก PolarSequencing

3.1) การหาความยาวคำรหัสแม่

ก่อนการลำดับช่องสัญญาณย่อย ระบบจำเป็นต้องหาความยาวคำรหัสแม่ N สำหรับการลำดับช่องสัญญาณย่อยและการเข้ารหัสโพลาร์ โดย N ต้องมีค่าเท่ากับสองกำลังใด ๆ การคำนวณความยาวคำรหัสแม่จะนำเวกเตอร์ $\mathbf{c}$ ที่มีความยาว K และความยาวคำรหัส E มาพิจารณาเพื่อให้ได้ความยาวคำรหัสแม่ที่เหมาะสม แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 2.49



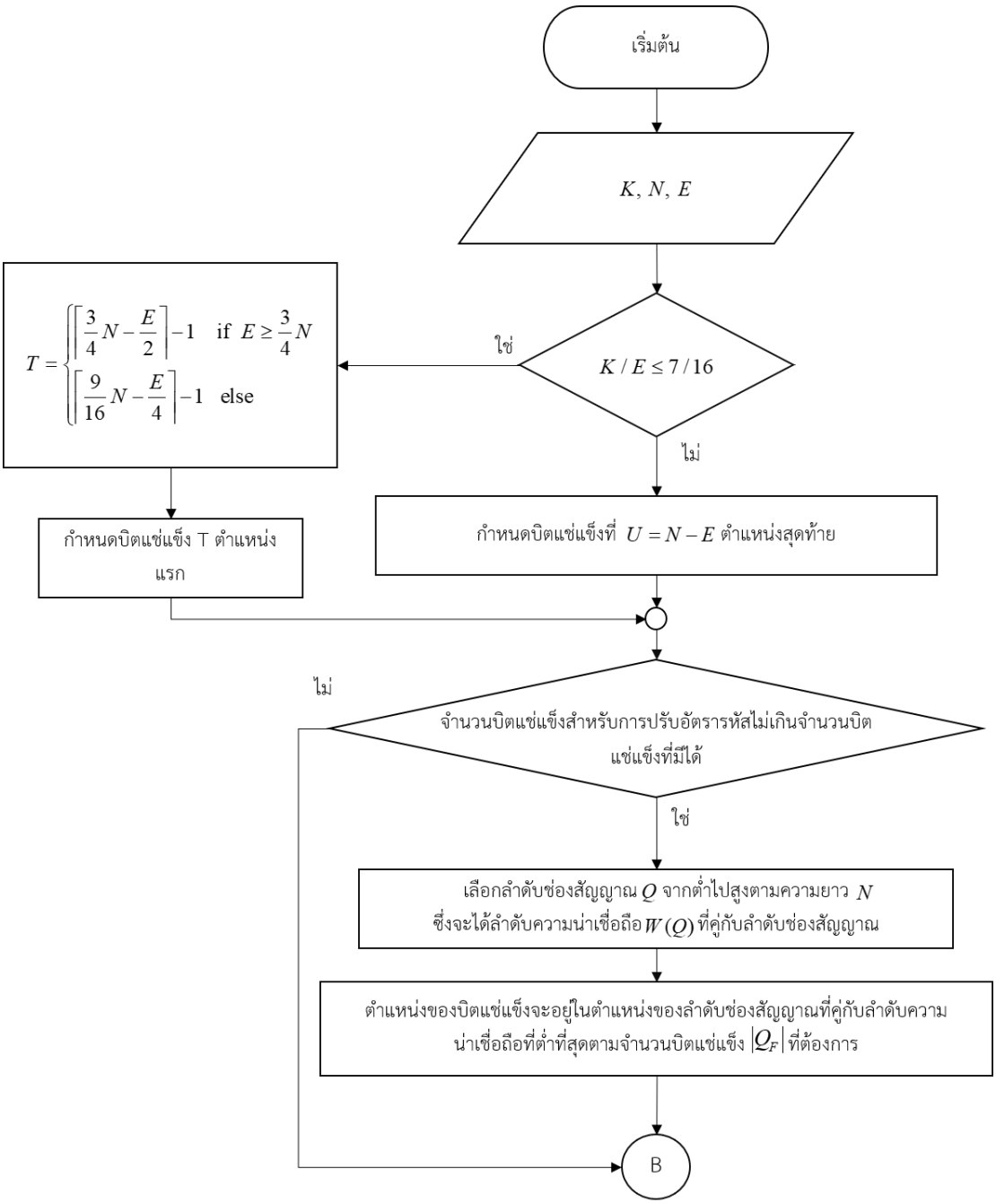
รูปที่ 2.49 แผนผังการทำงานการหาความยาวคำรหัสแม่

3.2) การลำดับช่องสัญญาณย่อย

การลำดับช่องสัญญาณย่อย ระบบจะค้นหาตำแหน่งบิตแซ่แข็งของรหัสโพลาร์ตามเงื่อนไขต่าง ๆ แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 2.50 และสามารถสรุปได้ดังนี้

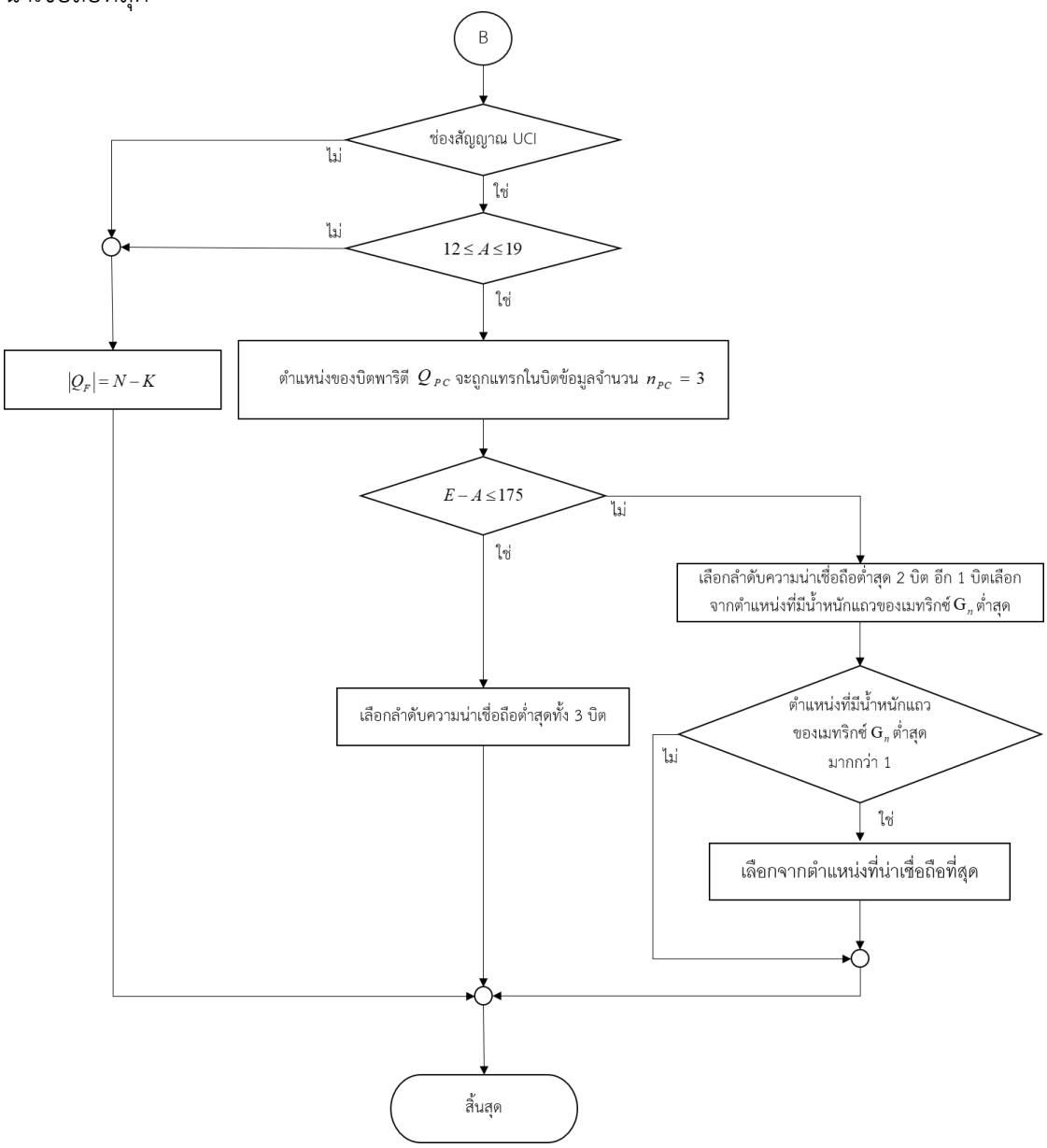
- เลือกตำแหน่งบิตแซ่แข็งสำหรับการปรับอัตรารหัส หาก $K/E \leq 7/16$ จะปรับอัตรารหัสด้วยการพังก์เจอร์ ซึ่งจะกำหนดบิตแซ่แข็งที่ T ตำแหน่งแรก ในกรณีอื่นจะปรับอัตรารหัสด้วยการช็อตเทน ซึ่งจะกำหนดบิตแซ่แข็งที่ $U = N - E$ ตำแหน่งสุดท้าย T

- หากจำนวนบิตแซ่แข็งสำหรับการปรับอัตรารหัสยังไม่เกินจำนวนบิตแซ่แข็งที่มีได้ จะเลือกบิตแซ่แข็งเพิ่มเติมตามลำดับความน่าเชื่อถือช่องสัญญาณตามมาตรฐาน



รูปที่ 2.50 แผนผังการทำงานการลำดับช่องสัญญาณย่อย

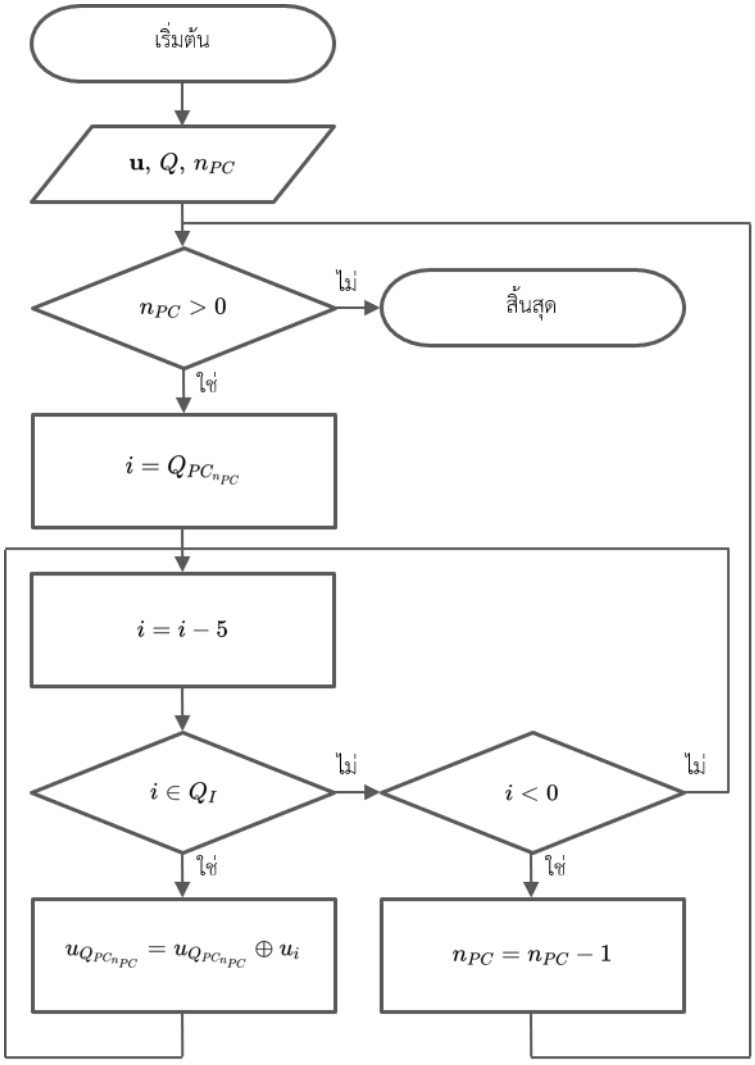
จากนั้นสำหรับช่องสัญญาณ PUCCH ที่ความยาว $A \le 19$ ระบบจะเติมบิตพาริตีเข้าในจำนวน $n_{PC} = 3$ ตามแผนภาพการทำงานแสดงได้ดังรูปที่ 2.51 โดยในกรณี $E - A \le 175$ บิตพาริตีทั้ง 3 บิตจะถูกเลือกจากลำดับความน่าเชื่อถือที่ต่ำที่สุด ส่วนในกรณี $E - A > 175$ $E - A > 175$ บิตพาริตี 2 บิตจะถูกเลือกจากลำดับความน่าเชื่อถือที่ต่ำที่สุดและบิตพาริตีสุดท้ายจะถูกเลือกโดยตำแหน่งที่มีน้ำหนักแถวของเมทริกซ์สร้าง $\mathbf{G}$ ต่ำที่สุดในตำแหน่งที่ไม่ใช่ตำแหน่งบิตแช่แข็ง $Q \notin Q_F$ $Q \notin Q_F$ หากมีตำแหน่งของน้ำหนักแถวมากกว่าหนึ่ง จะเลือกตำแหน่งน้ำหนักแถวต่ำสุดที่มีความน่าเชื่อถือที่สุด



รูปที่ 2.51 แผนผังการทำงานการลำดับช่องสัญญาณย่อย (ต่อ)

3.3) การคำนวณบิตพาริตี

การคำนวณบิตพาริตีจะหาคำนวณค่าของบิตพาริตีหากกระบวนการก่อนหน้ามีการจองพื้นที่ให้บิตพาริตีจำนวน n_{PC} บิต ซึ่งโมดูลจะทำงานเฉพาะช่องสัญญาณ PUCCH ในกรณีที่ $A \leq 19$ สำหรับกรณีอื่นและช่องสัญญาณ PBCH และ PDCCH โมดูลจะไม่ทำงาน โดยบิตพาริตีที่อยู่ในตำแหน่ง Q_{PC} จะมีความสัมพันธ์กับบิตก่อนหน้าถัดไปทีละ 5 ตำแหน่งเรื่อย ๆ ไปยังจุดเริ่มต้น และจะมีความสัมพันธ์เฉพาะบิตข้อมูลที่อยู่ในตำแหน่ง Q_I โดยเฉพาะ แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 2.52



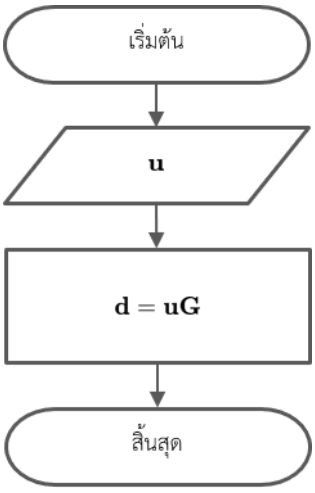
รูปที่ 2.52 แผนผังการทำงานการคำนวณบิตพาริตี

4) การเข้ารหัสโพลาร์ การแทรกสลับบล็อกย่อย และการปรับอัตรารหัส

โมดูลการเข้ารหัสโพลาร์ การแทรกสลับบล็อกย่อย และการปรับอัตรารหัส ภายใต้โมดูลหลัก PolarEncoding

4.1) การเข้ารหัสโพลาร์

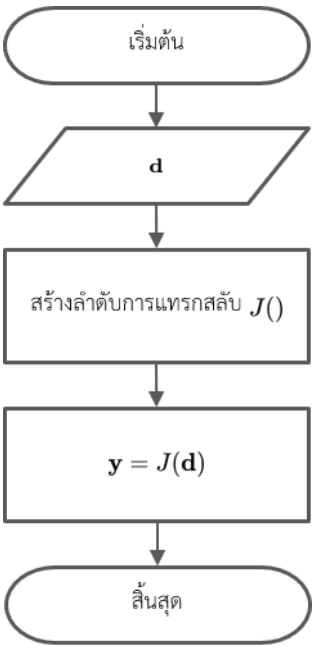
การเข้ารหัสโพลาร์จะทำการเข้ารหัสเวกเตอร์ $\mathbf{u}$ ที่มีความยาว N ที่ผ่านกระบวนการก่อนหน้า โดย N ต้องมีค่าเท่ากับสองกำลังใด ๆ ได้ผลลัพธ์เป็นเวกเตอร์ $\mathbf{d}$ ที่มีความยาว N เท่ากับเวกเตอร์ $\mathbf{u}$ แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 2.53



รูปที่ 2.53 แผนผังการทำงานการเข้ารหัสโพลาร์

4.2) การแทรกสลับบล็อกย่อย

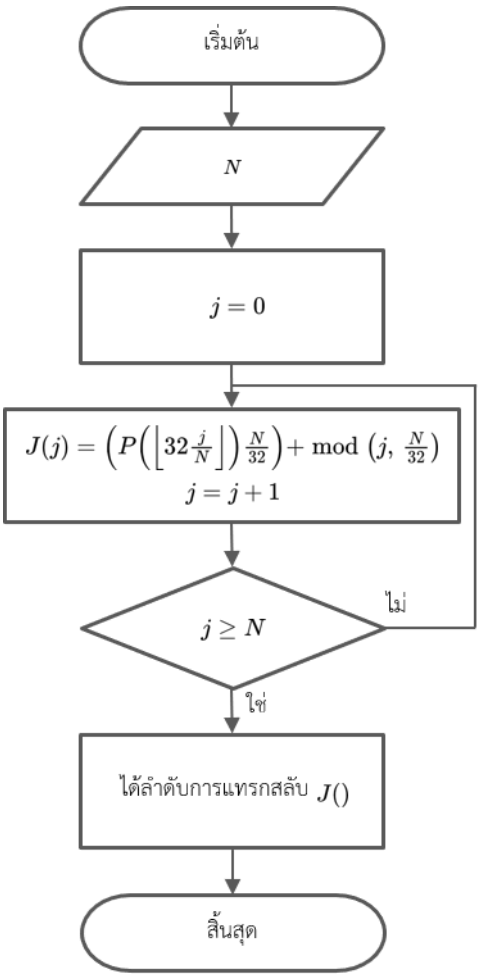
การแทรกสลับบล็อกย่อยจะนำเวกเตอร์ $\mathbf{d}$ ที่ผ่านการเข้ารหัสโพลาร์มาแทรกสลับตำแหน่งบิต ได้ผลลัพธ์เป็นเวกเตอร์ $\mathbf{y} = J(\mathbf{d})$ ที่มีความยาว N ที่มีความยาวเท่ากับเวกเตอร์ $\mathbf{d}$ แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 2.54



รูปที่ 2.54 แผนผังการทำงานการแทรกสลับบล็อกย่อย

4.3) การสร้างลำดับการแทรกสลับบล็อกย่อย

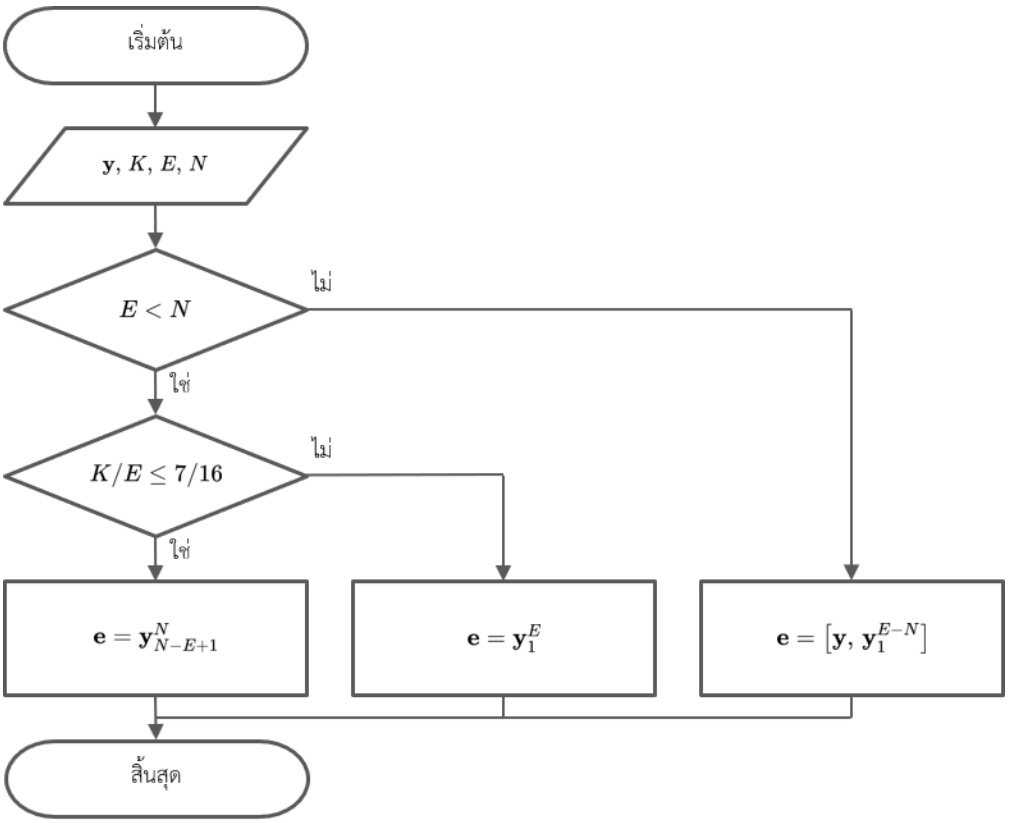
ลำดับการแทรกสลับบล็อกย่อยถูกสร้างโดยอัลกอริทึมที่ได้ระบุไว้ตามมาตรฐานร่วมกับลำดับการแทรกสลับ ได้ผลลัพธ์เป็นลำดับการแทรกสลับบล็อกย่อย $J(\ )$ แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 2.55 ลำดับการแทรกสลับบล็อกย่อยจะถูกคำนวณไว้ก่อนภายในกระบวนการลำดับช่องสัญญาณย่อยเพื่อใช้ในการกำหนดตำแหน่งบิตแซ่แข็งสำหรับการปรับอัตรารหัสแบบพังก์เชอร์



รูปที่ 2.55 แผนผังการทำงานการสร้างลำดับการแทรกสลับบล็อกย่อย

4.4) การปรับอัตรารหัส

ระบบจะปรับความยาวคำรหัสแม่ให้มีความยาวเท่ากับความยาวคำรหัสที่ผู้ใช้งานต้องการ โดยเวกเตอร์ $\mathbf{y}$ ที่ความยาว N จากกระบวนการก่อนหน้าจะถูกปรับความยาวให้เท่ากับ E หากกรณี $E < N \wedge K/E \leq 7/16$ เวกเตอร์ $\mathbf{y}$ จะถูกพังก์เชอร์ หากกรณี $E < N \wedge K/E > 7/16$ เวกเตอร์ $\mathbf{y}$ จะถูกช็อตเทน และหากกรณี $E > N$ เวกเตอร์ $\mathbf{y}$ จะถูกส่งซ้ำให้มีความยาวมากขึ้น โดยจะผลลัพธ์เป็นเวกเตอร์ $\mathbf{e}$ ที่มีความยาว E แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 2.56



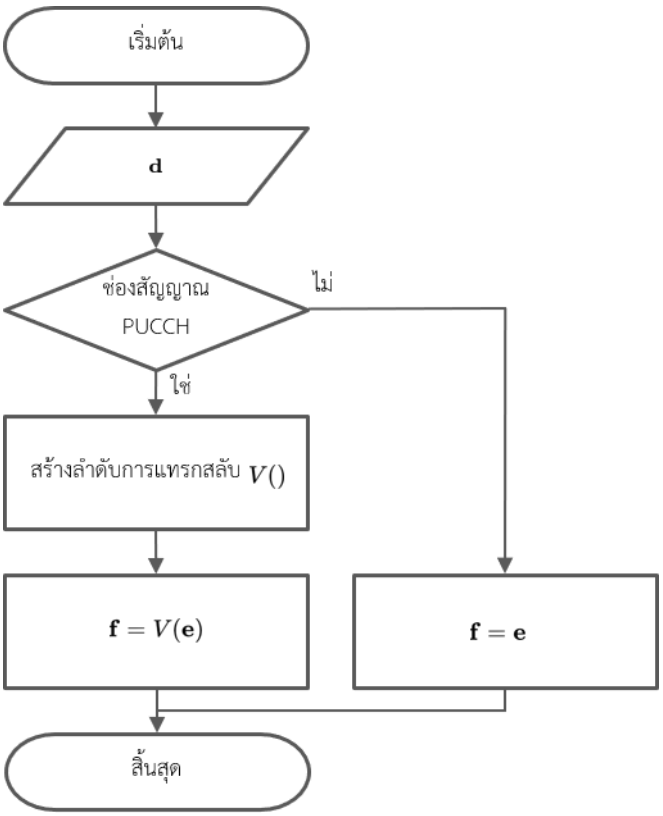
รูปที่ 2.56 แผนผังการทำงานการปรับอัตรารหัส

5) การแทรกสลับบิตรหัสและการต่อบล็อกรหัส

โมดูลการแทรกสลับบิตรหัสและการต่อบล็อกรหัส ภายใต้โมดูลหลัก CodedBitInterleaver

5.1) การแทรกสลับบิตรหัส

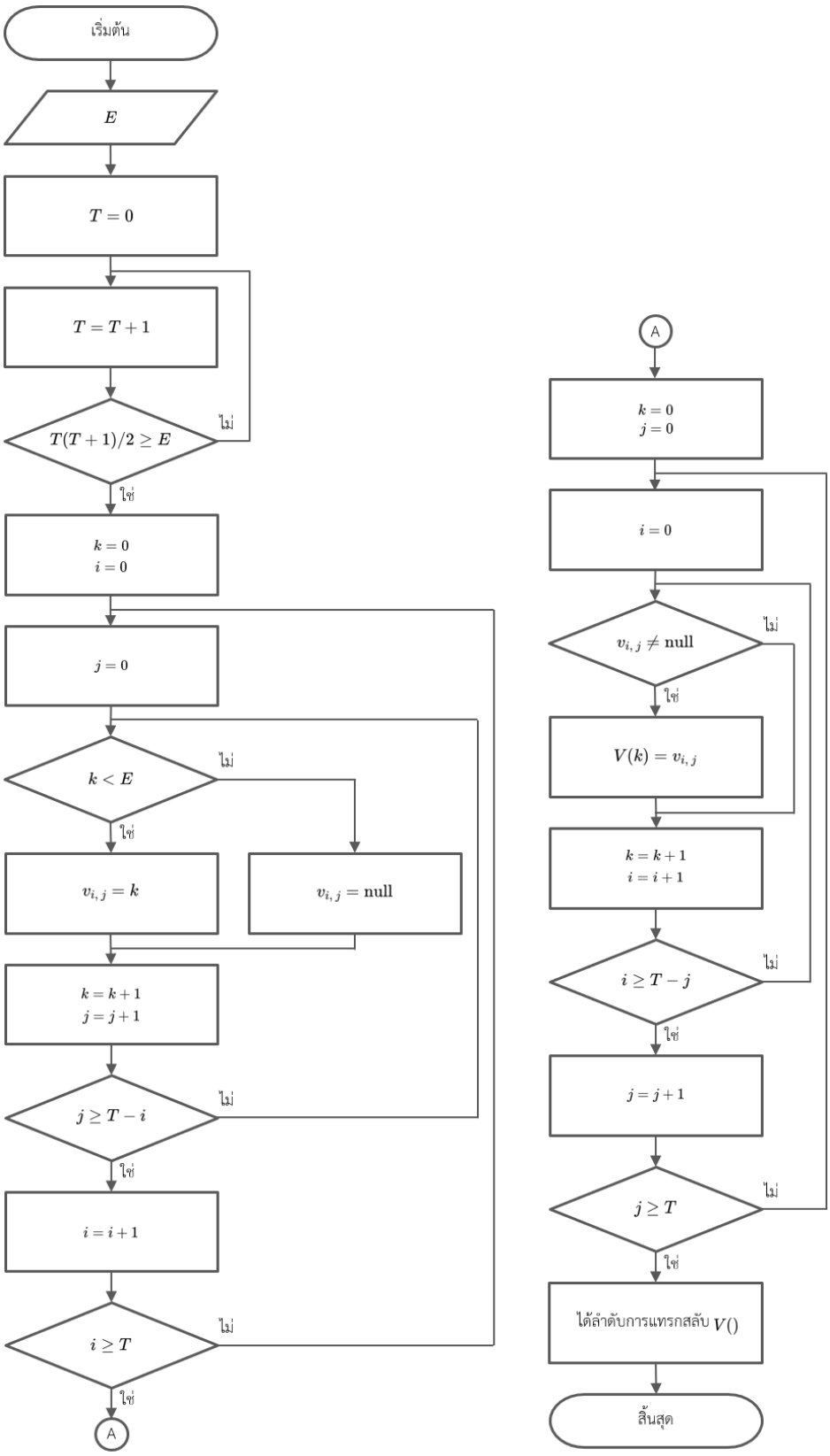
การแทรกสลับบิตรหัสจะนำเวกเตอร์ $\mathbf{e}$ ที่ผ่านการปรับอัตรารหัสมาแทรกสลับตำแหน่งบิต ได้ผลลัพธ์เป็นเวกเตอร์ $\mathbf{f} = V(\mathbf{e})$ ที่มีความยาว E ที่มีความยาวเท่ากับเวกเตอร์ $\mathbf{e}$ แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 2.57 โมดูลจะทำงานเฉพาะช่องสัญญาณ PUCCH สำหรับช่องสัญญาณ PBCH และ PDCCH โมดูลจะไม่ทำงานและได้ผลลัพธ์เวกเตอร์ $\mathbf{f} = \mathbf{e}$



รูปที่ 2.57 แผนผังการทำงานการแทรกสลับบิตรหัส

5.2) การสร้างลำดับการแทรกสลับบิตรหัส

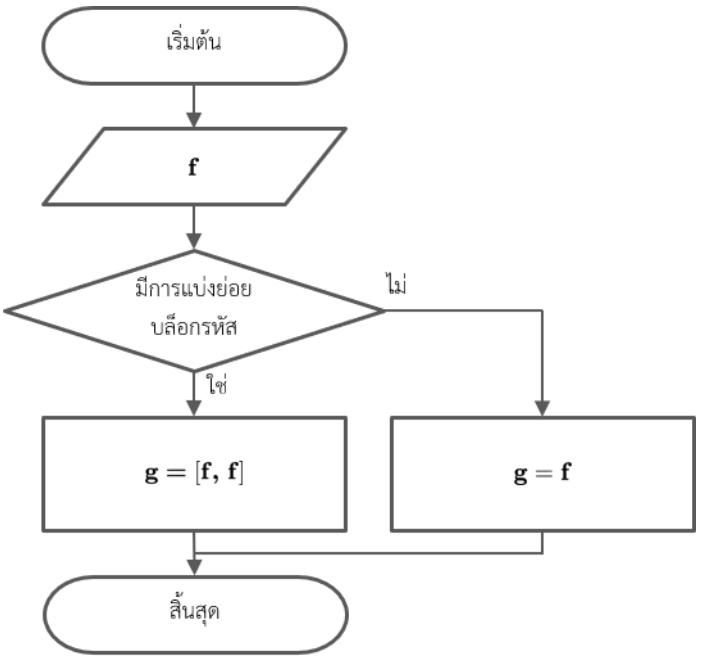
ลำดับการแทรกสลับบิตรหัสถูกสร้างโดยอัลกอริทึมที่ได้ระบุไว้ตามมาตรฐานร่วมกับลำดับการแทรกสลับที่คำนวณตามอัลกอริทึมดังรูปที่ 2.58 ได้ผลลัพธ์เป็นลำดับการแทรกสลับบิตรหัส $V(\ )$ แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 2.58



รูปที่ 2.58 แผนผังการทำงานการสร้างลำดับการแทรกสลับบิตรหัส

5.3) การต่อบล็อกรหัส

หากกระบวนแรกระบบได้มีการแบ่งย่อยบล็อกรหัส โมดูลการต่อบล็อกรหัสจะทำงาน โดยจะนำเวกเตอร์ $\mathbf{f}$ ทั้งสองเวกเตอร์ที่มีการแบ่งย่อยบล็อกรหัสก่อนหน้ามาต่อกันตามลำดับเดิม ได้ผลลัพธ์เป็นเวกเตอร์ $\mathbf{g}$ ที่ความยาว $2E$ หากมีการแบ่งย่อยบล็อกรหัสก่อนหน้าและ E หากไม่มีการแบ่งย่อยบล็อกรหัสก่อนหน้า แผนภาพการทำงานสามารถแสดงได้ดังรูปที่ 2.59

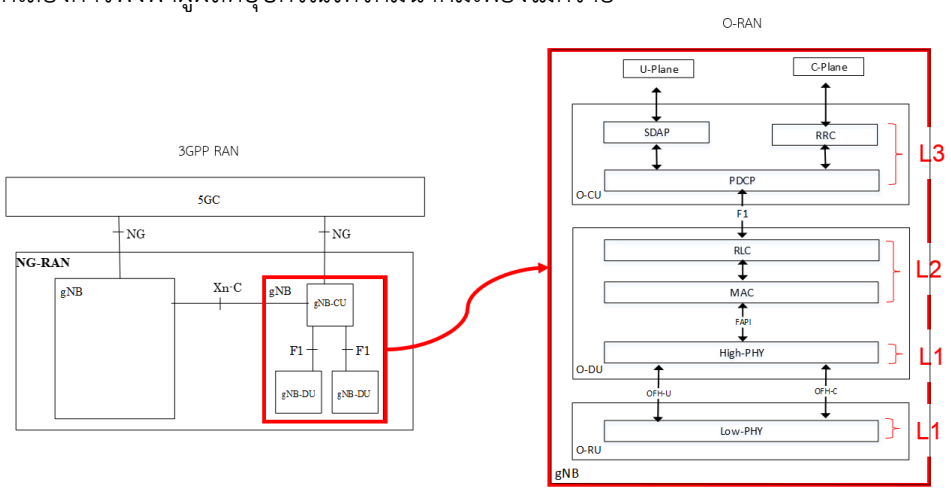


รูปที่ 2.59 แผนผังการทำงานการต่อบล็อกรหัส

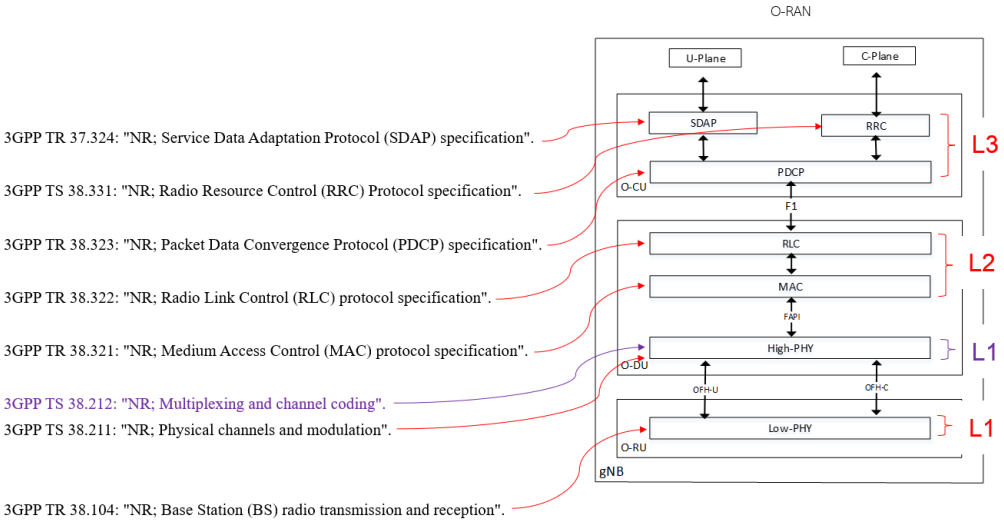
2.7 โครงข่าย O-RAN และการ์ดเร่งความเร็ว

ปัจจุบัน โครงข่ายโทรศัพท์มือถือสามารถแบ่งได้เป็น โครงข่ายการเข้าถึงด้วยคลื่นวิทยุ (Radio Access Network: RAN) และโครงข่ายแกนหลัก (Core Network: CN) ดังรูปที่ 2.59 ซึ่งรายละเอียดของโครงข่ายจะถูกพัฒนาตามมาตรฐานของ 3GPP ทั้งนี้ โครงข่าย 3GPP RAN สามารถแบ่งได้เป็น Central unit (CU) และ Distributed unit (DU) ซึ่งในโครงข่าย 5G จะเรียกว่า gNB-CU และ gNB-DU ตามลำดับ อย่างไรก็ตาม 3GPP ไม่ได้มีการกำหนดมาตรฐานการเชื่อมต่อระหว่าง CU และ DU อีกทั้งไม่ได้ระบบวิธีการแบ่ง RAN ออกเป็น CU และ DU ไว้ชัดเจน ทำให้ ผู้ผลิตอุปกรณ์โทรคมนาคมต่างออกแบบวิธีการเชื่อมต่อส่วนประกอบต่าง ๆ โดยใช้มาตรฐานของตนเอง จึงเป็นที่มาของ O-RAN ที่ทำการแยก 3GPP RAN ออกเป็น 3 ส่วน ได้แก่ O-RU O-DU และ O-CU มีการระบุมาตรฐานการเชื่อมต่อส่วนต่าง ๆ ไว้อย่างชัดเจน ได้แก่ OFH-U OFH-C FAPI และ F1 ดังรูปที่ 2.60 โดยฟังก์ชันการทำงานภายใน O-RU O-DU และ O-CU ยังคงอ้างอิงมาตรฐานของ 3GPP ดังรูปที่ 2.61 โดยจะพบว่า O-DU จะพัฒนาโดยอ้างอิงมาตรฐานของ 3GPP จำนวน 2 ฉบับ ได้แก่ 3GPP TS 38.212: “NR; Multiplexing and channel coding” และ 3GPP TS 38.211: “NR; Physical

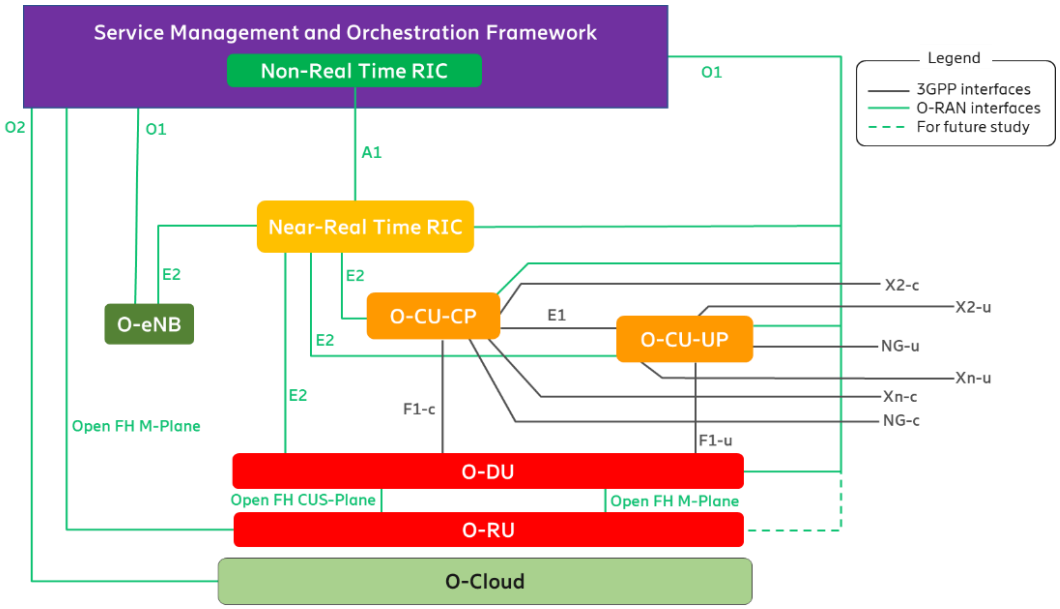
channels and modulation” ซึ่งเอกสาร 3GPP TS 38.212 เป็นเอกสารที่อธิบายวิธีการเข้ารหัสช่องสัญญาณที่ผู้วิจัยใช้อ้างอิงในการพัฒนาซอฟต์แวร์และฮาร์ดแวร์ของโครงการวิจัยก่อนหน้า นอกจากการกำหนดรายละเอียดภายใน O-CU O-DU O-RU และมาตรฐานการเชื่อมต่อ OFH-U OFH-C FAPI และ F1 องค์กร O-RAN ได้นำเสนอการพัฒนาระบบควบคุม O-CU O-DU และ O-RU โดยใช้ Non-Real Time RIC และ Near-Real Time RIC ซึ่งทำให้ 3GPP RAN สามารถทำงานโดยใช้ซอฟต์แวร์แบบเปิด เพื่อหลีกเลี่ยงการใช้ฮาร์ดแวร์ของผู้ผลิตอุปกรณ์โทรคมนาคมเพียงไม่กี่ราย รูปที่ 2.62 แสดงส่วนประกอบของ O-RAN แบบละเอียด [12] นอกจากนี้ ยังแสดงมาตรฐานการเชื่อมต่อระหว่างส่วนประกอบของ O-RAN ที่ผู้ผลิตอุปกรณ์สามารถนำไปพัฒนาอุปกรณ์ของตนเอง การกำหนดส่วนประกอบและมาตรฐานการเชื่อมต่อที่ชัดเจนนี้ จะทำให้เกิดบริษัทใหม่ๆ ที่เน้นวิจัยและพัฒนาอุปกรณ์เฉพาะส่วน โดยยังคงสามารถนำอุปกรณ์เฉพาะส่วนของบริษัทต่างๆ มาทำงานร่วมกันได้ หลีกเลี่ยงการพึงพาผู้ผลิตอุปกรณ์โทรคมนาคมเพียงไม่กี่ราย



รูปที่ 2.59 การเปรียบเทียบระหว่าง 3GPP RAN และ O-RAN [12]



รูปที่ 2.60 มาตรฐาน 3GPP ที่ใช้ในส่วนประกอบของ O-RAN

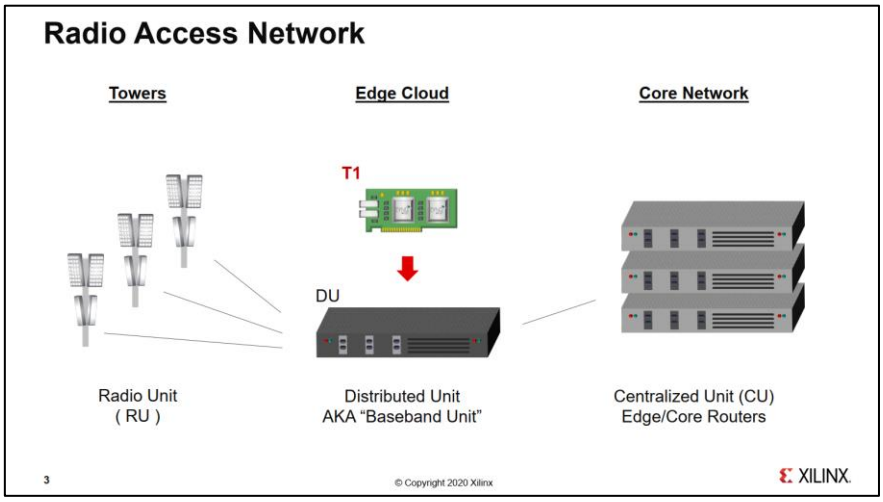


รูปที่ 2.61 ส่วนประกอบของ O-RAN แบบละเอียด [12]

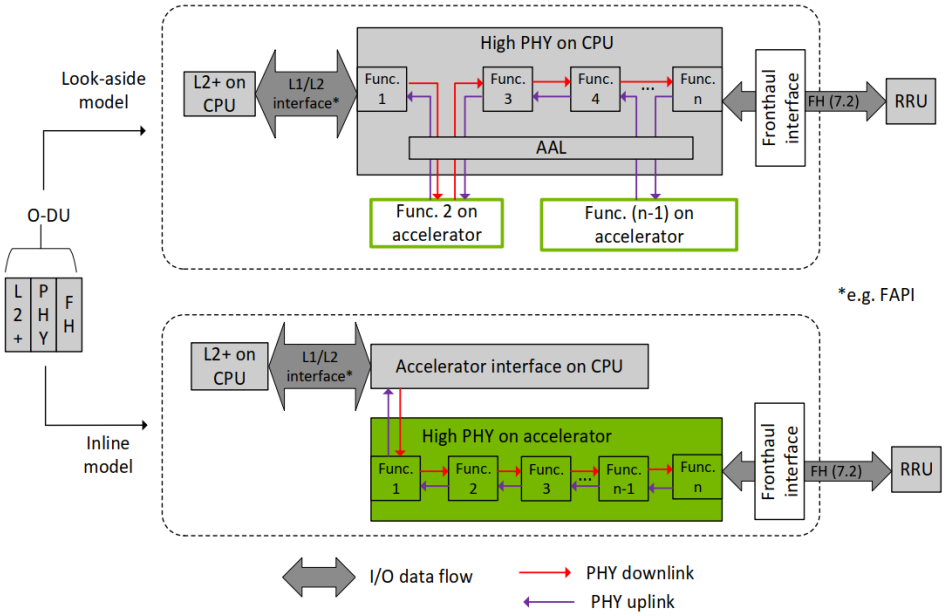
นอกจาก O-RAN จะกำหนดมาตรฐานการเชื่อมต่อระหว่างส่วนประกอบต่าง ๆ ของ RAN ทางกลุ่ม O-RAN ยังมีความประสงค์ที่จะพัฒนาซอฟต์แวร์โอเพนซอร์ซ (open source software) ที่สามารถทำงานบนเครื่องคอมพิวเตอร์แบบเซิร์ฟเวอร์ (server) ทั่วไปได้ เพื่อให้ผู้บริการโทรศัพท์เคลื่อนที่ต่าง ๆ สามารถนำซอฟต์แวร์โอเพนซอร์ซไปพัฒนาต่อยอดตามวัตถุประสงค์ที่ต้องการ ทั้งนี้ จะต้องใช้อุปกรณ์ที่มีคุณสมบัติดังรูปที่ 4 โดยจะพบว่าส่วนประกอบ Near-Real Time RIC O-CU-UP และ O-CU-UP สามารถทำงานบนเครื่องเซิร์ฟเวอร์มาตรฐานได้ แต่ส่วนประกอบ O-DU จำเป็นจะต้องใช้การ์ดเร่งความเร็ว (Accelerator Card) เพิ่มเติมดังรูปที่ 2.63 โดยเอกสารของ O-RAN Working Group 6 ได้กำหนดแนวทางการนำการ์ดเร่งความเร็วมาใช้ใน O-DU จำนวน 2 รูปแบบ ได้แก่ Look-aside model และ Inline model ดังรูปที่ 2.64 โดยการใช้งานการ์ดเร่งความเร็วแบบ Look-aside model หมายถึงการนำการ์ดเร่งความเร็วมาช่วยคำนวณการทำงานของ O-DU บางฟังก์ชัน เช่น FEC สำหรับการใช้งานการ์ดเร่งความเร็วแบบ Inline model จะเป็นการนำการ์ดตัวเร่งมาช่วยทุกส่วนตั้งแต่การรับสัญญาณจาก RRU ไปจนถึงการฟังก์ชัน O-DU Layer-2

Cloud/ HW features	Near-RT RIC	O-CU-CP	O-CU-UP	O-DU	O-RU
Standard Cloud Infrastructure (CI) & General Purpose CPU	✓	✓			
CI + high speed UP support. Acceleration optional			✓		
CI + high speed UP, acceleration for O-DU				✓	
CI + high speed UP, acceleration for O-RU					✓

รูปที่ 2.62 ฮาร์ดแวร์ที่ใช้สำหรับ O-RAN [12]



รูปที่ 2.63 การส่วนประกอบของ DU ที่มีการประยุกต์ใช้งานการ์ดเร่งความเร็ว (Accelerator Card)



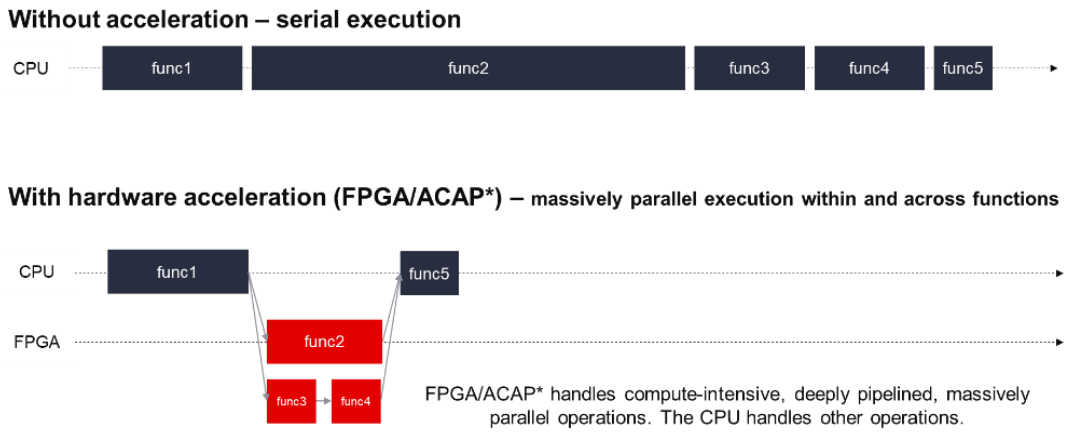
รูปที่ 2.64 รูปแบบการใช้งานการ์ดเร่งความเร็ว (Accelerator Card) กับ O-DU [12]

2.8 มาตรฐาน O-RAN Working Group 6 เรื่อง Acceleration Abstraction Layer General Aspects and Principles และ FEC Profiles

2.8.1 หมวดทั่วไป

1) ฮาร์ดแวร์สำหรับเร่งความเร็วการประมวลผล

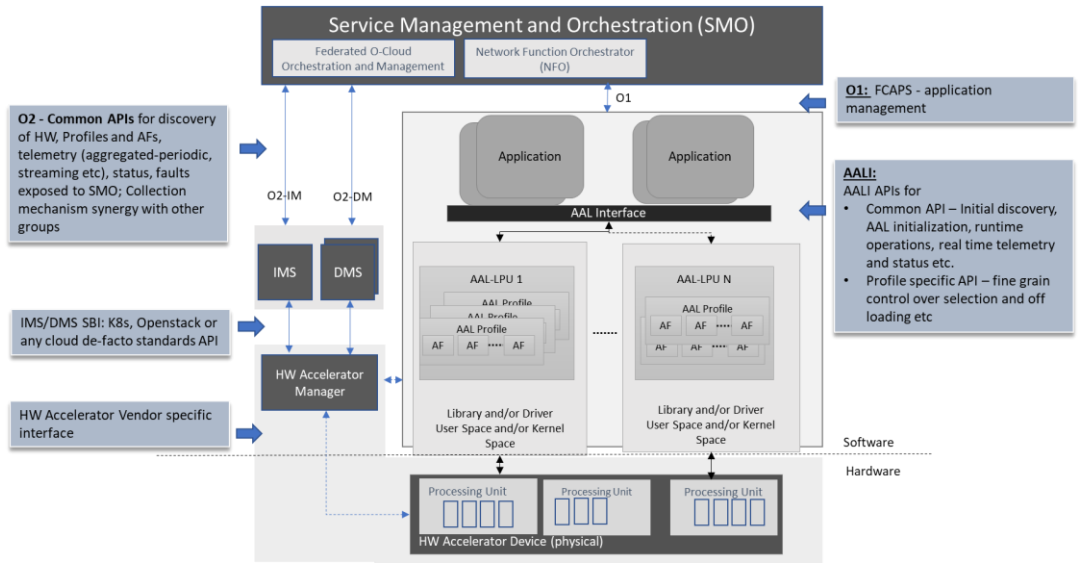
CPU เป็นฮาร์ดแวร์ซึ่งสามารถประมวลผลแบบใด ๆ แต่มีข้อจำกัดซึ่งเกิดจากลักษณะการทำงานของ CPU ที่ทำงานแบบอนุกรม ซึ่งส่งผลให้ CPU ประมวลผลได้ช้าหากมีข้อมูลจำนวนมาก ฮาร์ดแวร์ประเภทอื่นเช่น GPU FPGA ASIC ซึ่งออกแบบมาเพื่อการประมวลแบบขนานจึงถูกนำมาใช้แทนในบางบริบท มาตรฐาน O-RAN ได้นำฮาร์ดแวร์ดังกล่าวมาใช้เพื่อเร่งความเร็วแทน CPU เพื่อให้สามารถทำงานแบบขนานได้ ดังแสดงในรูปที่ 2.65



รูปที่ 2.65 การทำงานของระบบที่มีฮาร์ดแวร์เร่งความเร็วเทียบกับระบบที่มีแต่ CPU

2) โครงสร้างของ Acceleration Abstraction Layer (AAL)

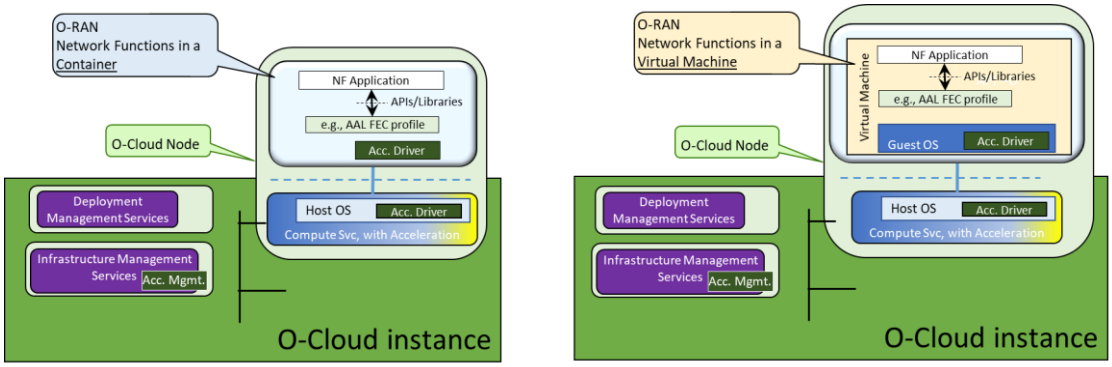
โครงสร้าง AAL ประกอบไปด้วยส่วนเชื่อมต่อระหว่างแอปพลิเคชันกับฮาร์ดแวร์สำหรับเร่งความเร็วการประมวลผล เพื่อเป็นการรองรับความหลากหลายทางฮาร์ดแวร์ ซอฟท์แวร์และการใช้งานในรูปแบบต่าง ๆ จึงได้มีการกำหนดรูปแบบของ AAL ที่หลากหลายเพื่อให้เหมาะกับการใช้งาน ภาพรวมของโครงสร้าง AAL เป็นดังรูปที่ 2.66



รูปที่ 2.66 ภาพรวมของ AAL

2.1) AAL บน Virtual Machine

ข้อมูลจำเพาะของ AAL ได้นิยามถึงส่วนต่อประสานและโปรไฟล์ใด ๆ ส่วนต่อประสานแอปพลิเคชัน AAL-C จะถูกใช้ในฟังก์ชันของโครงข่าย O-RAN Cloudified สำหรับการเข้าถึงซึ่งครอบคลุมทั้งฮาร์ดแวร์และซอฟท์แวร์ดังแสดงในรูปที่ 2.67



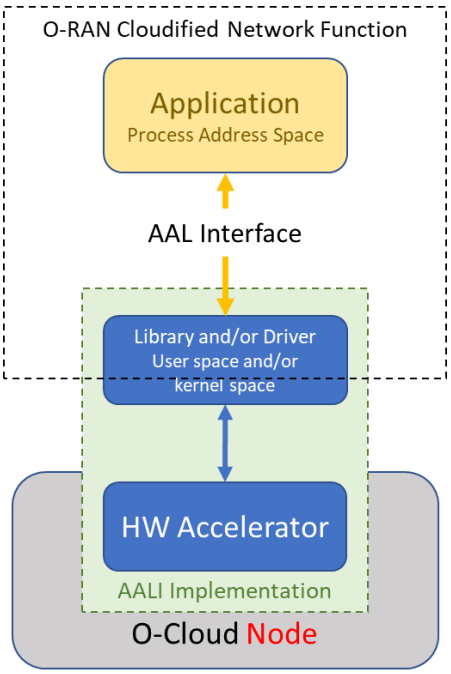
รูปที่ 2.67 AAL บน Virtual Machine

3) ลักษณะจำเพาะของ AAL

AAL ต้องมีความยืดหยุ่นสำหรับการใช้งานซอฟท์แวร์ที่หลากหลายบน CPU ซึ่งมีฮาร์ดแวร์เร่งความเร็วอยู่ในระบบและยืดหยุ่นสำหรับการเพิ่มหรือลดขนาดโครงข่ายด้วยซอฟท์แวร์ชุดเดิม

4) ขอบเขตของ AAL

AAL ประกอบไปด้วย API โมเดลข้อมูลข่าวสาร การดำเนินการต่าง ๆ และระบบของอินพุตเอาต์พุตที่ถูกใช้โดย แอปพลิเคชันเพื่อสร้างส่วนประสานถึง AAL ดังในกรอบเส้นประแสดงในรูปที่ 2.68



รูปที่ 2.68 ขอบเขตของ AAL

5) หลักการทั่วไปสำหรับส่วนต่อประสาน

ส่วนต่อประสาน AAL จะต้องมีคุณสมบัติดังนี้

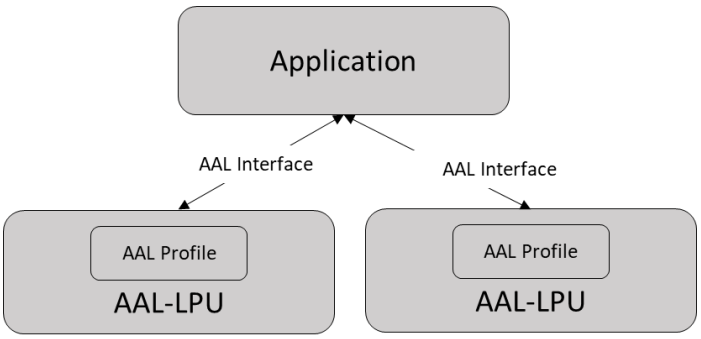
5.1) มีความยืดหยุ่นสำหรับการใช้ฟังก์ชันต่าง ๆ

5.2) มีฮาร์ดแวร์สำหรับการใช้งานเป็นของตัวเอง

5.3) มีระบบ Interrupt and Poll

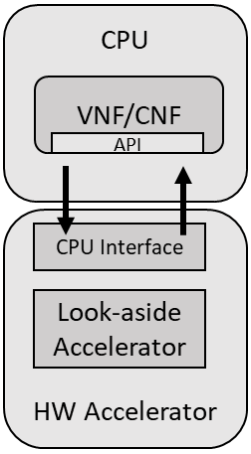
5.4) รับรู้ได้ว่าได้รับอินพุตจากที่ใด เพื่อไปกระทำการใด ๆ ต่อ

5.5) รองรับ AAL-LPU หลายชุดดังรูปที่ 2.69



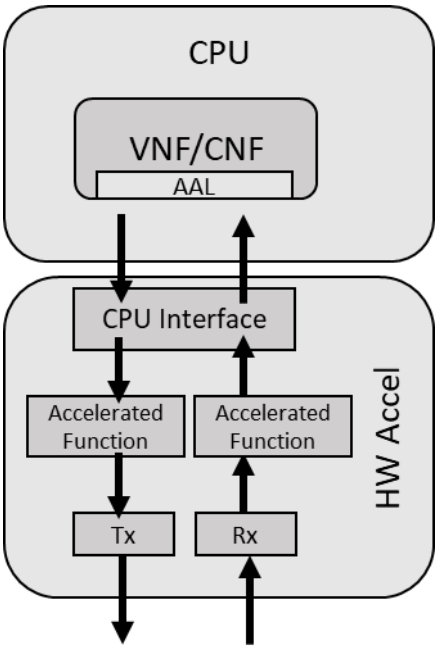
รูปที่ 2.69 AAL-LPU หลายชุดที่เชื่อมต่อกับแอปพลิเคชัน

5.6) รองรับโครงสร้างแบบ Look-aside ดังรูปที่ 2.70



รูปที่ 2.70 โครงสร้างแบบ Look-aside

5.7) รองรับโครงสร้างแบบ Inline ดังรูปที่ 2.71



รูปที่ 2.71 โครงสร้างแบบ Inline

5.8) ทำงานแบบขนานในแต่ละส่วน

5.9) ส่วนประสานของผู้ใช้และส่วนประสานของตัวควบคุมแยกจากกัน

5.10) รับอินพุตได้ทุกขนาดตามโปรโตคอล

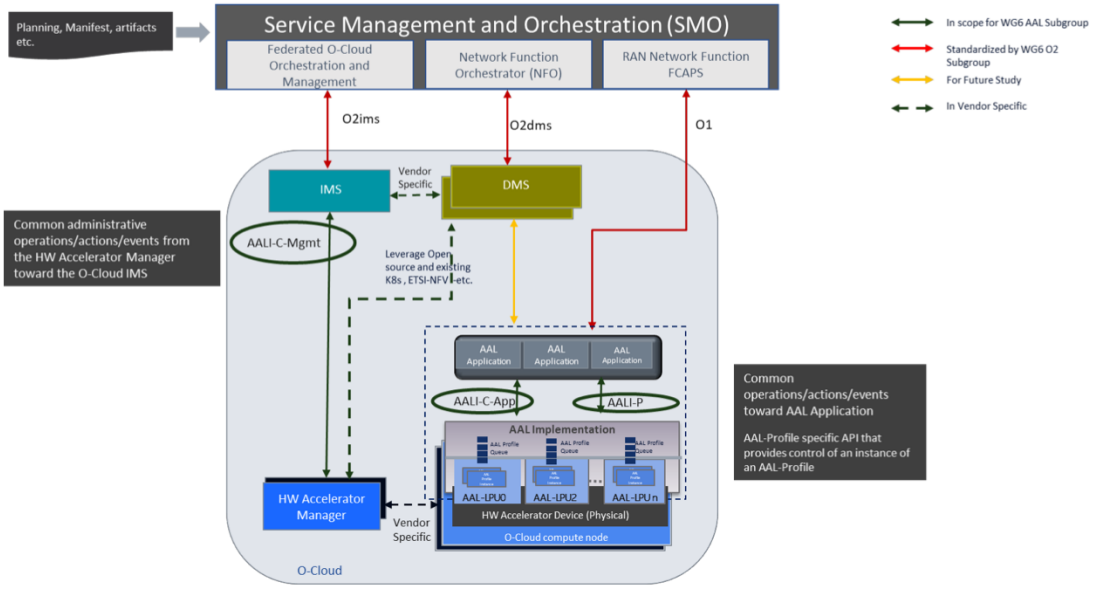
5.11) รับและส่งข้อมูลได้หลากหลายรูปแบบ

5.12) มีชื่อ API เฉพาะตัว

2.8.2 ตัวควบคุมฮาร์ดแวร์เร่งความเร็วและส่วนต่อประสาน AAL

1) หน้าที่ของตัวควบคุมฮาร์ดแวร์เร่งความเร็ว

ตัวควบคุมฮาร์ดแวร์เร่งความเร็วมีหน้าที่ตรวจสอบการทำงานของฮาร์ดแวร์บน O-Cloud ในทุกด้านซึ่งประกอบด้วย อายุขัย ความผิดปกติทางฮาร์ดแวร์ สถานะ ประสิทธิภาพ การตั้งค่า การอัปเกรด และการจัดการความผิดพลาด ตัวควบคุมฮาร์ดแวร์เร่งความเร็วนี้เป็นจุดสิ้นสุดของส่วนต่อประสาน AALI-C-Mgmt ดังรูปที่ 2.72



รูปที่ 2.72 ตัวควบคุมฮาร์ดแวร์เร่งความเร็วในระบบ O-Cloud

2.8.3 AAL-LPU

AAL-LPU เป็นส่วนประมวลผลระหว่างแอปพลิเคชันกับฮาร์ดแวร์เร่งความเร็วซึ่งมีคุณสมบัติดังนี้

1) มีการเก็บสถิติการเรียกใช้

2) มีหน่วยความจำแยกเป็นของตัวเอง

3) มีการตั้งค่าการประมวลผลให้รองรับโปรไฟล์ AAL ที่หลากหลาย

2.8.4 โปรไฟล์ของ AAL

2.8.4.1 O-DU AAL

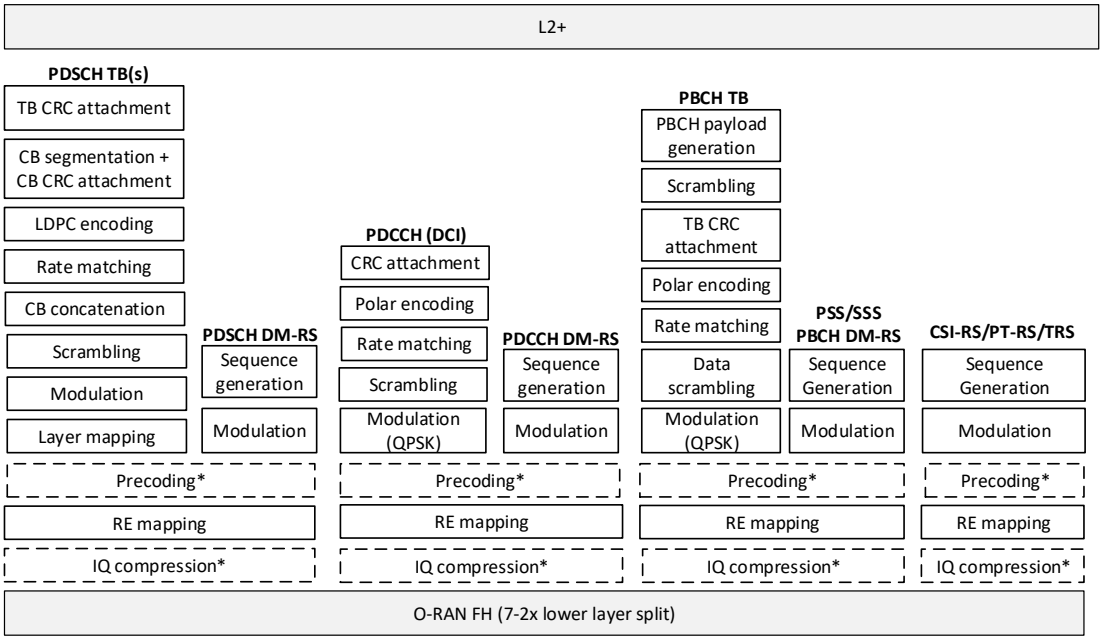
โปรไฟล์ O-DU AAL คือเซ็ตของฟังก์ชันในลำดับชั้นโปรโตคอลของ O-DU โดยฟังก์ชันดังกล่าวอาจกระจายอยู่ต่างชั้นโปรโตคอลกัน แต่มาตรฐาน O-RAN จะเน้นฟังก์ชันที่อยู่บนลำดับชั้น PHY เป็นหลัก

1) ลำดับชั้นโปรโตคอลของ O-DU

บล็อกกระบวนการประมวลผลในลำดับชั้นโปรโตคอลของ O-DU จะเป็นไปตามกระบวนการตามมาตรฐานที่ 3GPP กำหนดสำหรับการสื่อสาร 5G

1.1) ลำดับชั้นโปรโตคอลของ O-DU สำหรับช่องสัญญาณขาลง

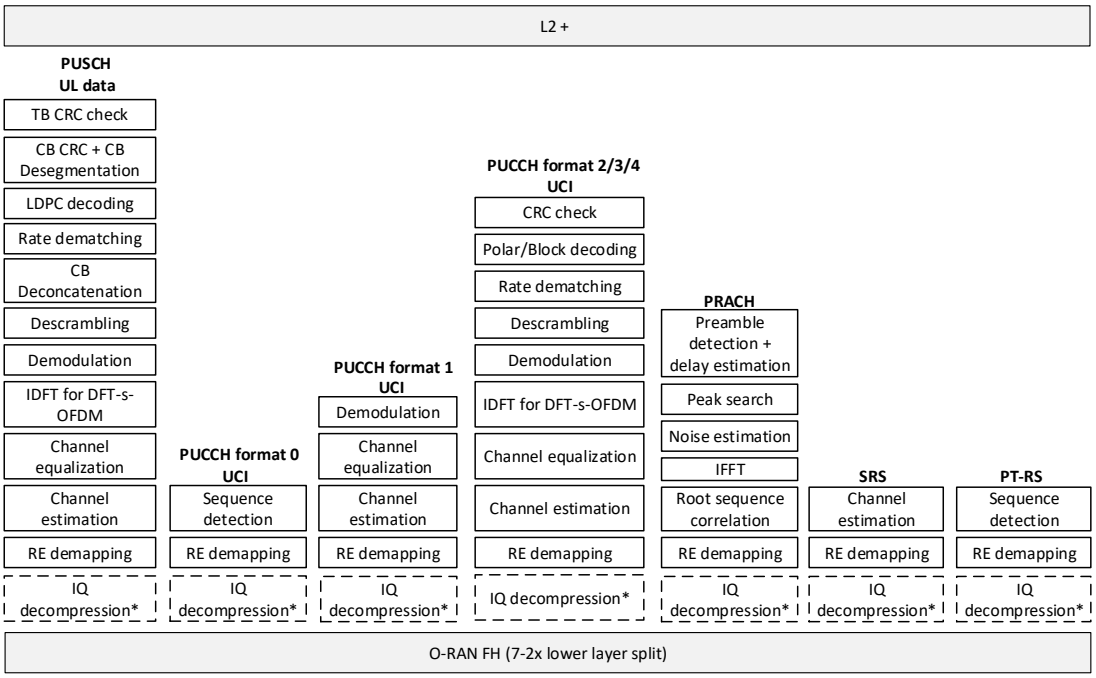
ลำดับชั้นโปรโตคอลของ O-DU สำหรับช่องสัญญาณขาลงเป็นดังรูปที่ 2.73



รูปที่ 2.73 ลำดับชั้นโปรโตคอลของ O-DU สำหรับช่องสัญญาณขาลง

1.2) ลำดับชั้นโปรโตคอลของ O-DU สำหรับช่องสัญญาณขาขึ้น

ลำดับชั้นโปรโตคอลของ O-DU สำหรับช่องสัญญาณขาขึ้นเป็นดังรูปที่ 2.74



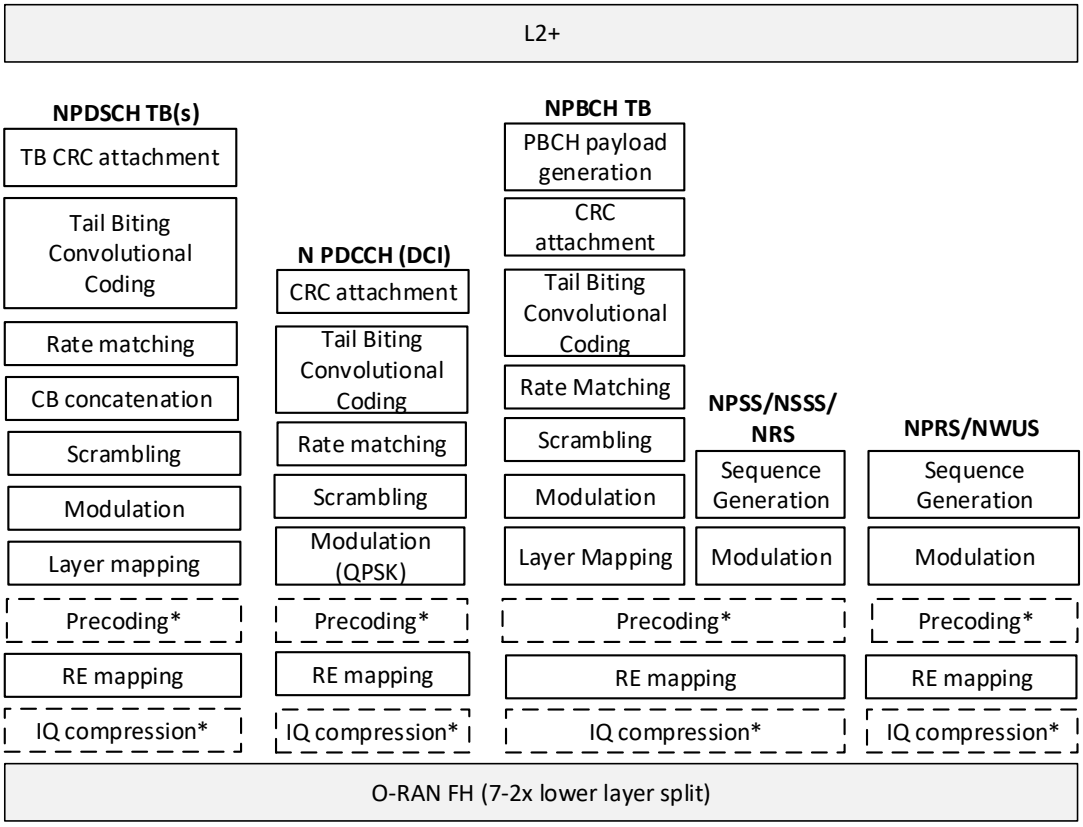
รูปที่ 2.74 ลำดับชั้นโปรโตคอลของ O-DU สำหรับช่องสัญญาณขาขึ้น

2.8.4.2 ลำดับชั้นโปรโตคอลของ O-DU (mMTC)

บล็อกกระบวนการประมวลผลในลำดับชั้นโปรโตคอลของ O-DU (mMTC) จะเป็นไปตามกระบวนการตามมาตรฐานที่ 3GPP กำหนดสำหรับการสื่อสาร 4G และ 5G

1) ลำดับชั้นโปรโตคอลของ O-DU (mMTC) สำหรับช่องสัญญาณขาลง

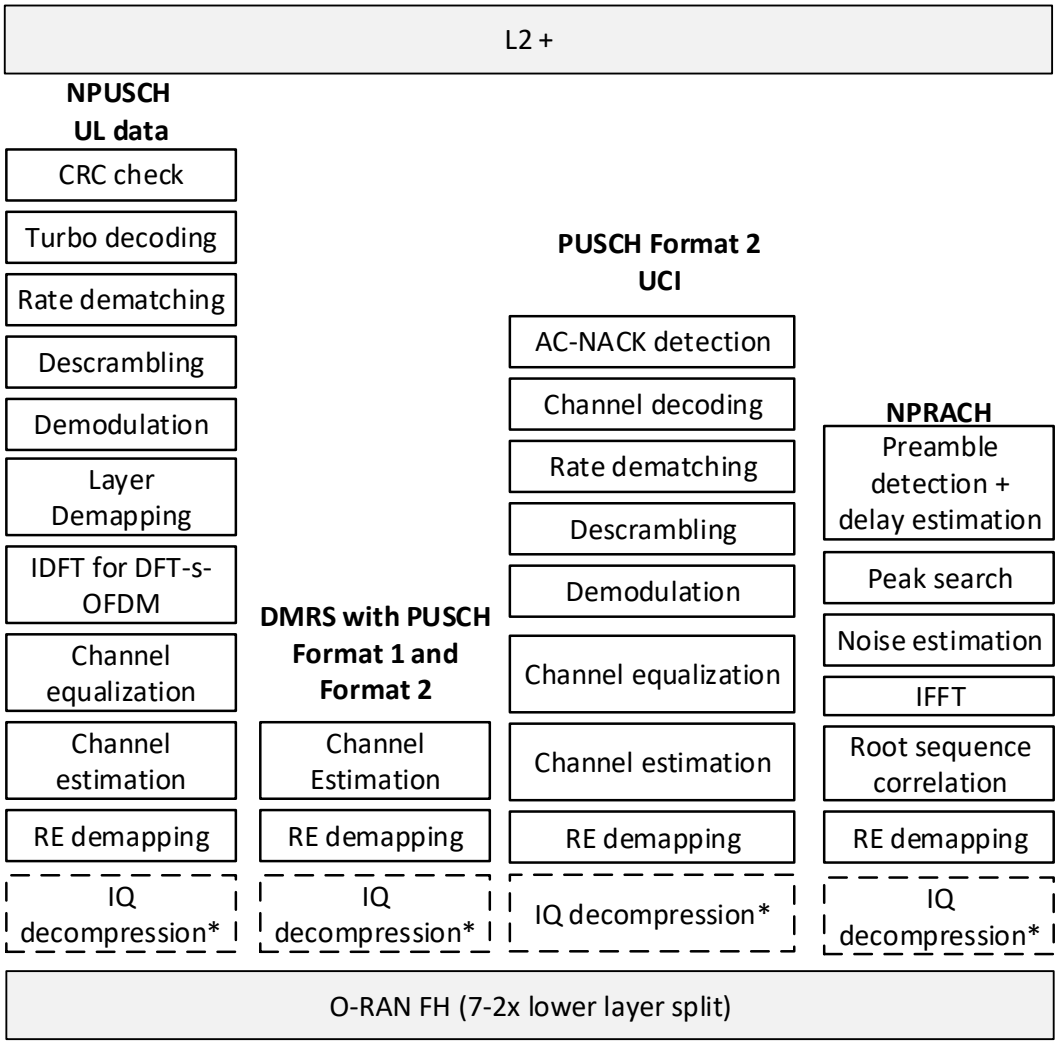
ลำดับชั้นโปรโตคอลของ O-DU (mMTC) สำหรับช่องสัญญาณขาลงเป็นดังรูปที่ 2.75



รูปที่ 2.75 ลำดับชั้นโปรโตคอลของ O-DU (mMTC) สำหรับช่องสัญญาณขาลง

2) ลำดับชั้นโปรโตคอลของ O-DU (mMTC) สำหรับช่องสัญญาณขาขึ้น

ลำดับชั้นโปรโตคอลของ O-DU (mMTC) สำหรับช่องสัญญาณขาขึ้นเป็นดังรูปที่ 2.76

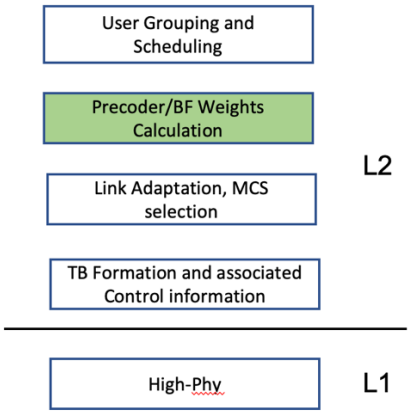


รูปที่ 2.76 ลำดับชั้นโปรโตคอลของ O-DU (mMTC) สำหรับช่องสัญญาณขาขึ้น

2.8.4.3 โปรไฟล์ของ O-DU AAL

1) AAL_MU-MIMO_PRECODER_WEIGHTS_CALC

โปรไฟล์ AAL_MU-MIMO_PRECODER_WEIGHTS_CALC เป็นดังบล็อกสีเขียวในรูปที่ 2.77

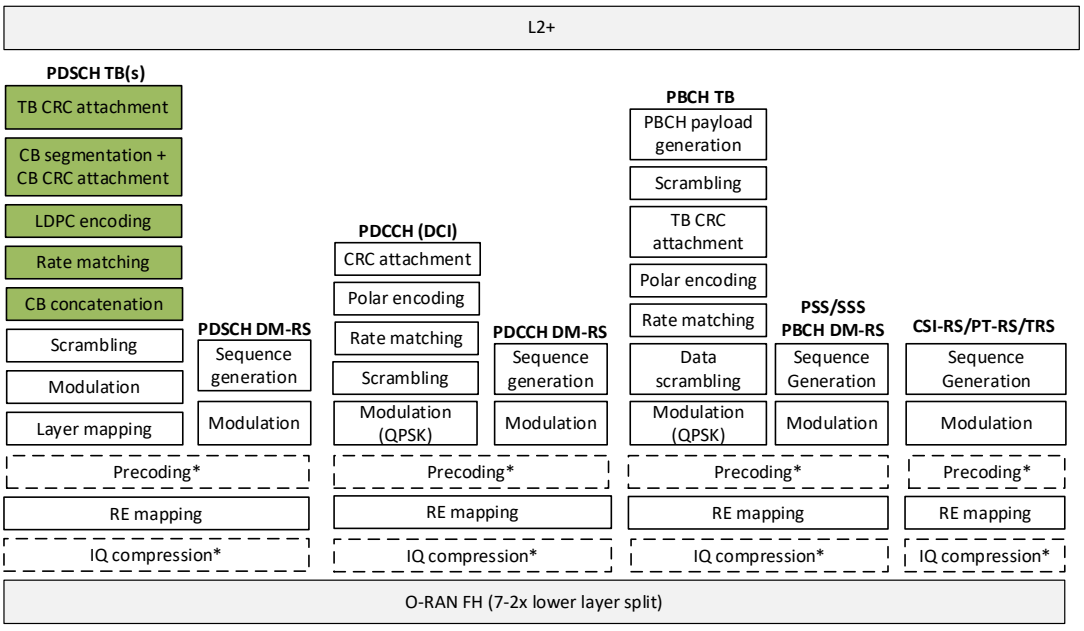


รูปที่ 2.77 โปรไฟล์ AAL_MU-MIMO_PRECODER_WEIGHTS_CALC

2.8.4.4 โปรไฟล์ของ O-DU AAL สำหรับช่องสัญญาณขาลง

1) AAL_PDSCH_FEC

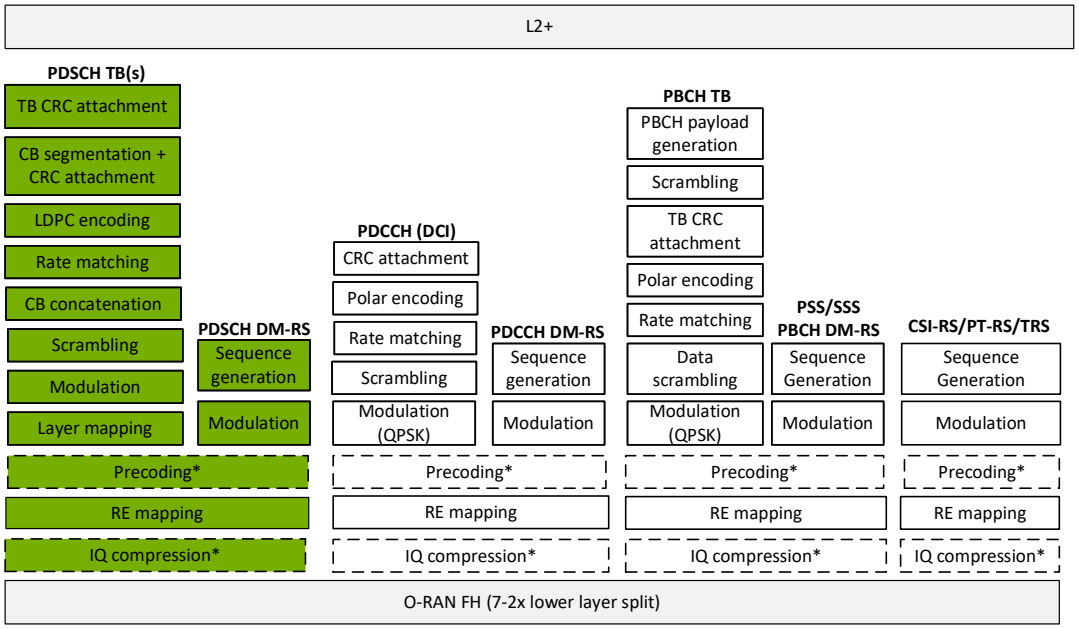
โปรไฟล์ AAL_PDSCH_FEC เป็นดังบล็อกสีเขียวในรูปที่ 2.78



รูปที่ 2.78 โปรไฟล์ AAL_PDSCH_FEC

2) AAL_PDSCH_HIGH-PHY

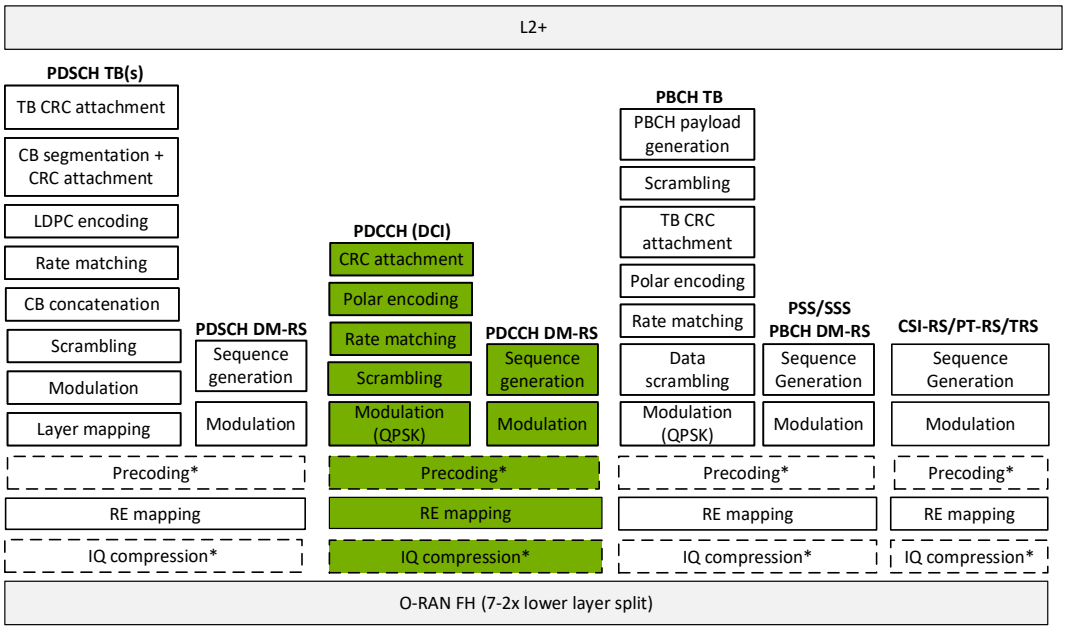
โปรไฟล์ AAL_PDSCH_HIGH-PHY เป็นดังบล็อกสีเขียวในรูปที่ 2.79



รูปที่ 2.79 โปรไฟล์ AAL_PDSCH_HIGH-PHY

3) AAL_PDCCH_HIGH-PHY

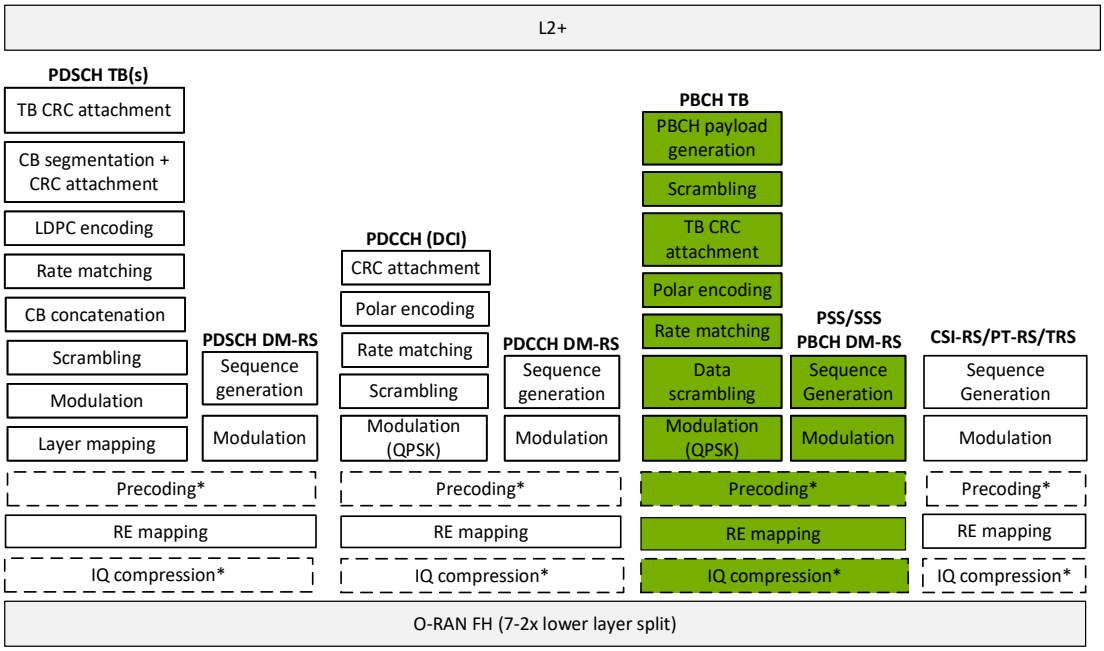
โปรไฟล์ AAL_PDCCH_HIGH-PHY เป็นดังบล็อกสีเขียวในรูปที่ 2.80



รูปที่ 2.80 โปรไฟล์ AAL_PDCCH_HIGH-PHY

4) AAL_PBCH_HIGH-PHY

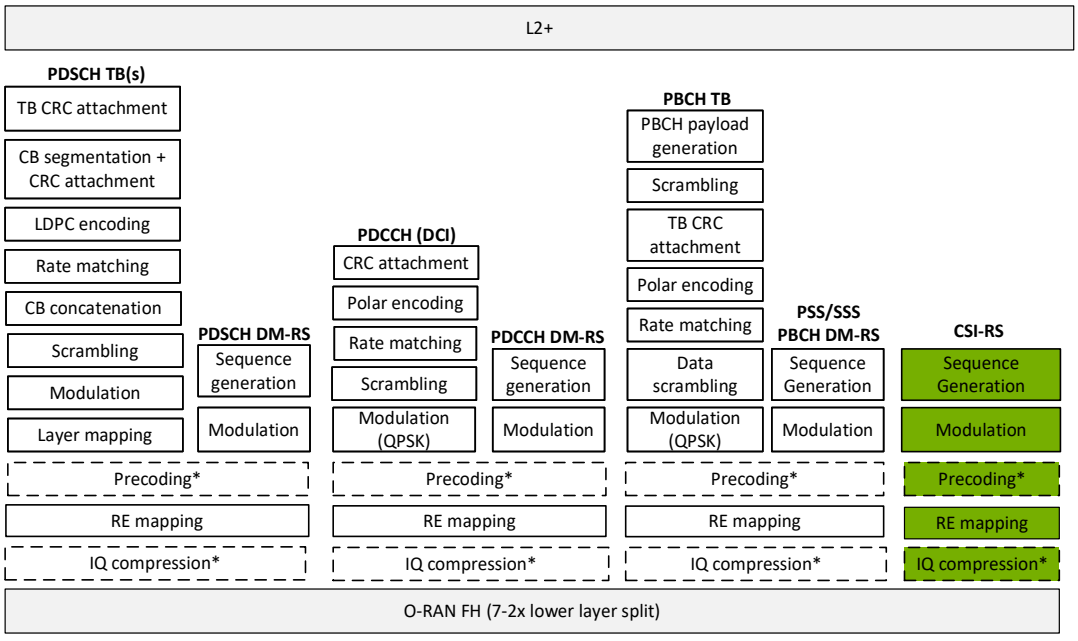
โปรไฟล์ AAL_PBCH_HIGH-PHY เป็นดังบล็อกสีเขียวในรูปที่ 2.81



รูปที่ 2.81 โปรไฟล์ AAL_PBCH_HIGH-PHY

5) AAL_CSI-RS_HIGH-PHY

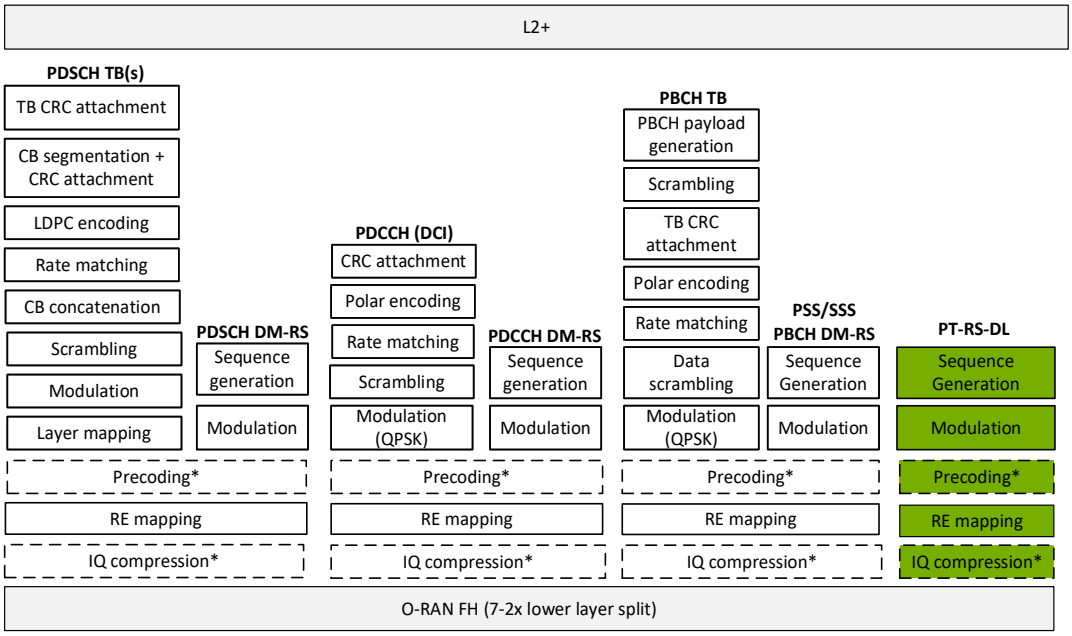
โปรไฟล์ AAL_CSI-RS_HIGH-PHY เป็นดังบล็อกสีเขียวในรูปที่ 2.82



รูปที่ 2.82 โปรไฟล์ AAL_CSI-RS_HIGH-PHY

6) AAL_PT-RS-DL_HIGH-PHY

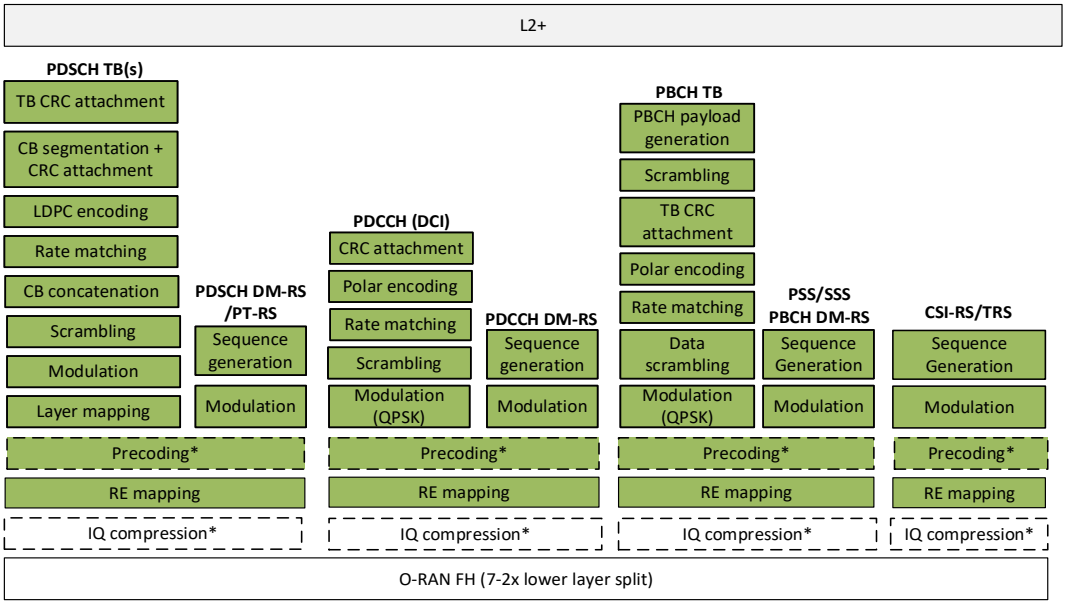
โปรไฟล์ AAL_PT-RS-DL_HIGH-PHY เป็นดังบล็อกสีเขียวในรูปที่ 2.83



รูปที่ 2.83 โปรไฟล์ AAL_PT-RS-DL_HIGH-PHY

7) AAL_DOWNLINK_HIGH-PHY

โปรไฟล์ AAL_DOWNLINK_HIGH-PHY เป็นดังบล็อกสีเขียวในรูปที่ 2.84

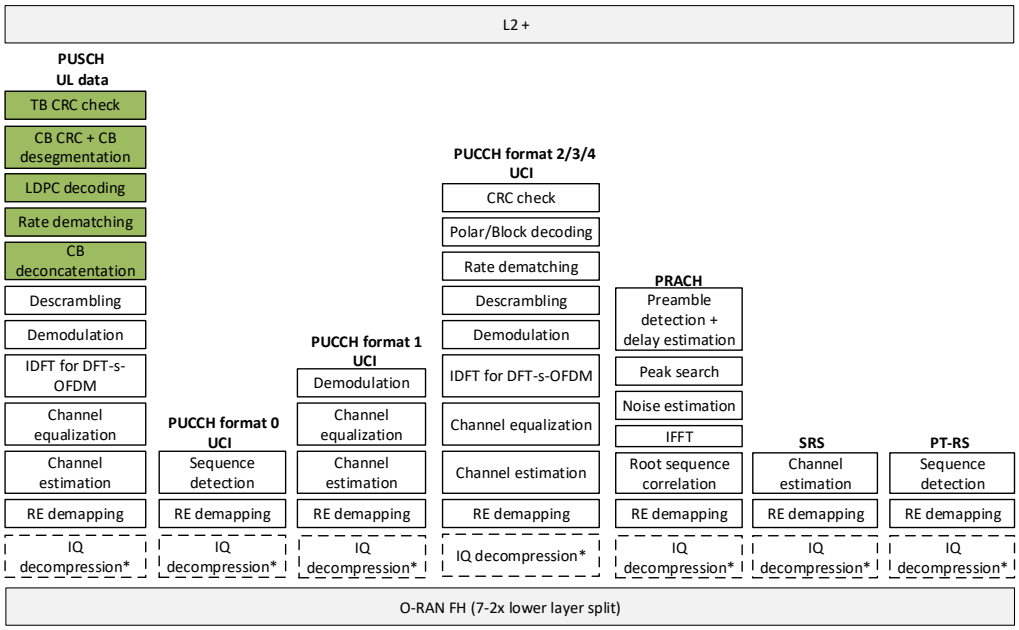


รูปที่ 2.84 โปรไฟล์ AAL_DOWNLINK_HIGH-PHY

2.8.4.5 โปรไฟล์ของ O-DU AAL สำหรับช่องสัญญาณขาขึ้น

1) AAL_PUSCH_FEC

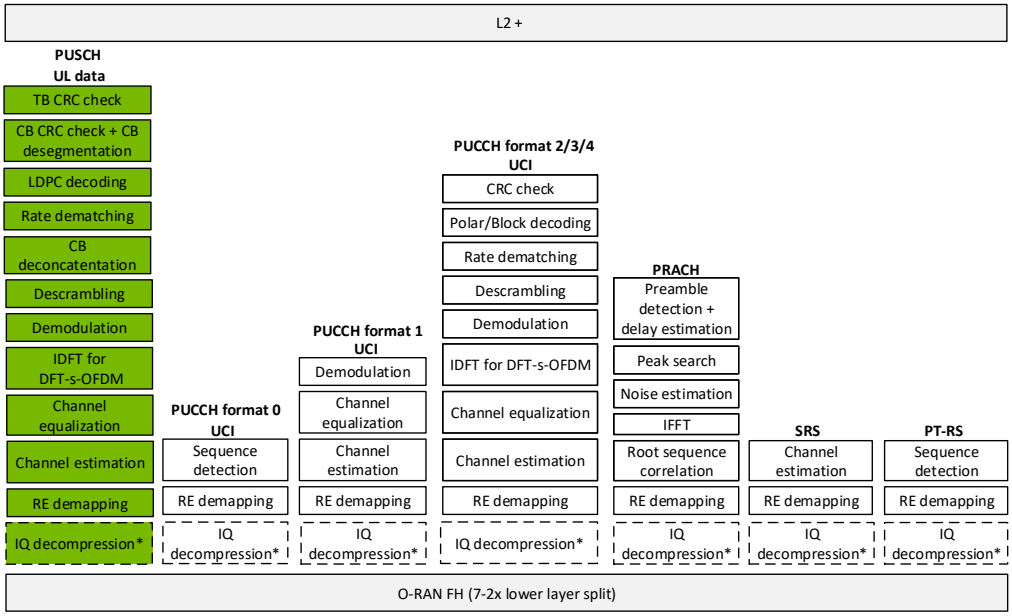
โปรไฟล์ AAL_PUSCH_FEC เป็นดังบล็อกสีเขียวในรูปที่ 2.85



รูปที่ 2.85 โปรไฟล์ AAL_PUSCH_FEC

2) AAL_PUSCH_HIGH-PHY

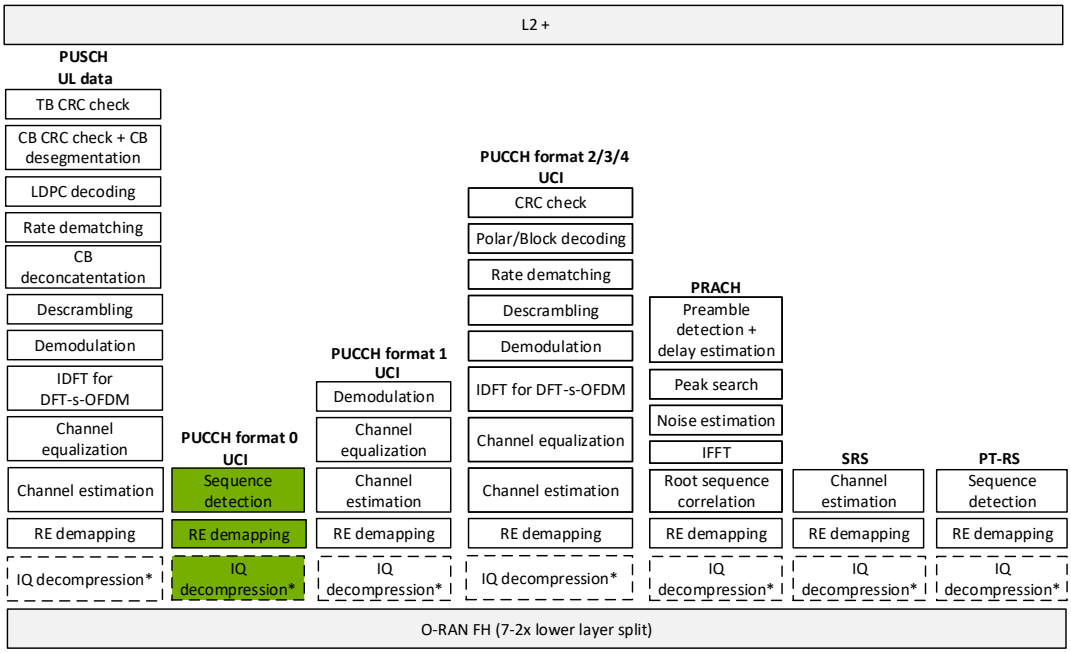
โปรไฟล์ AAL_PUSCH_HIGH-PHY เป็นดังบล็อกสีเขียวในรูปที่ 2.86



รูปที่ 2.86 โปรไฟล์ AAL_PUSCH_HIGH-PHY

3) AAL_PUCCH_HIGH-PHY (format 0)

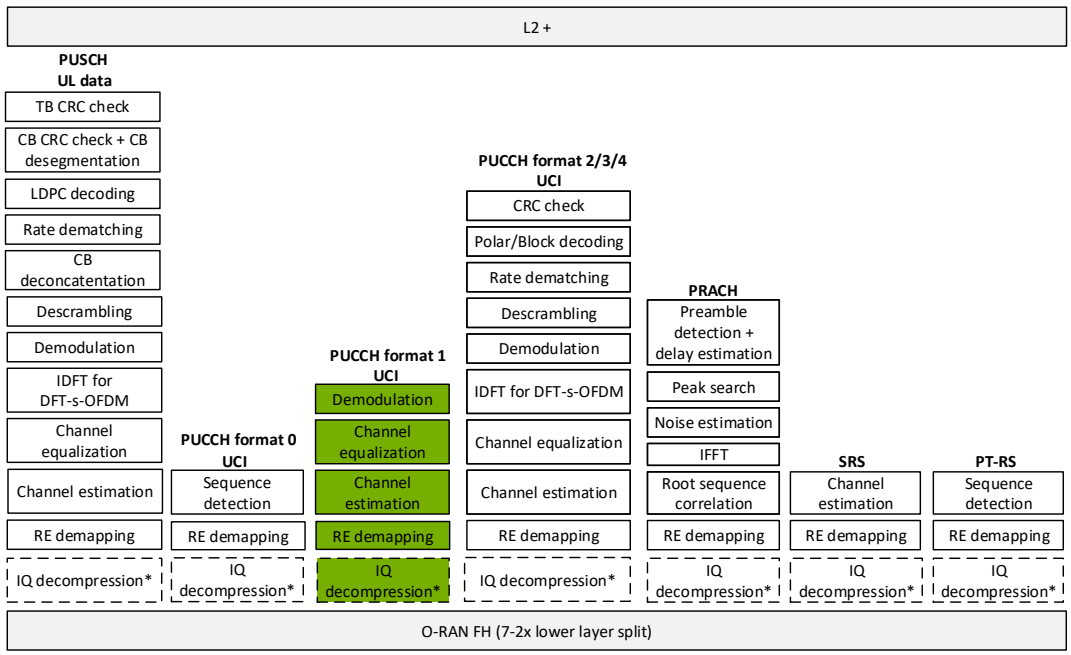
โปรไฟล์ AAL_PUCCH_HIGH-PHY (format 0) เป็นดังบล็อกสีเขียวในรูปที่ 2.87



รูปที่ 2.87 โปรไฟล์ AAL_PUCCH_HIGH-PHY (format 0)

4) AAL_PUCCH_HIGH-PHY (format 1)

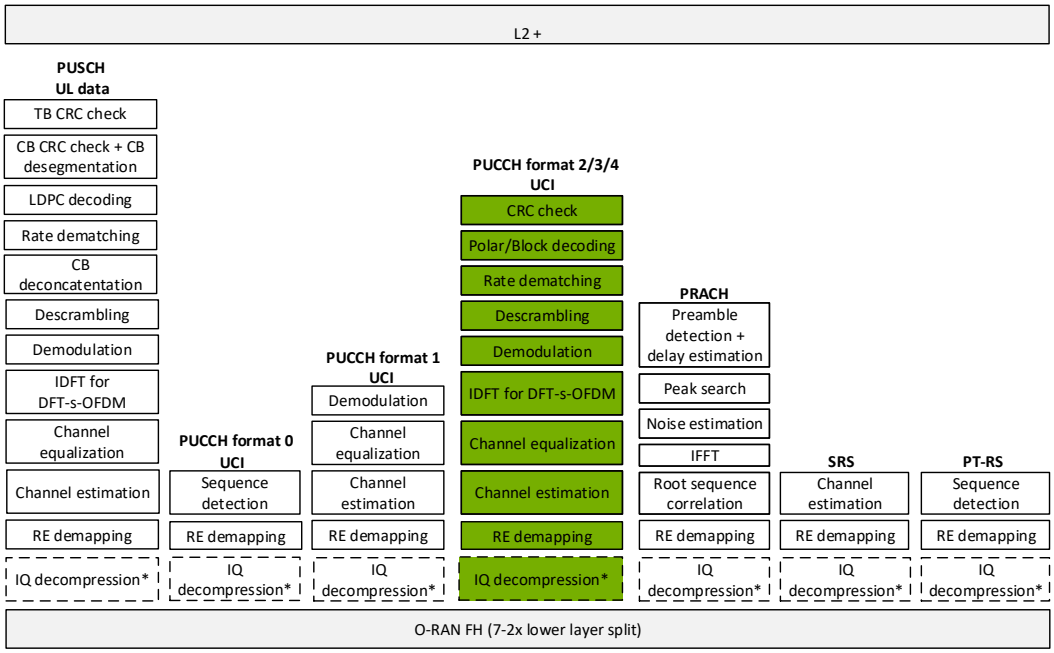
โปรไฟล์ AAL_PUCCH_HIGH-PHY (format 1) เป็นดังบล็อกสีเขียวในรูปที่ 2.88



รูปที่ 2.88 โปรไฟล์ AAL_PUCCH_HIGH-PHY (format 1)

5) AAL_PUCCH_HIGH-PHY (format 2/3/4)

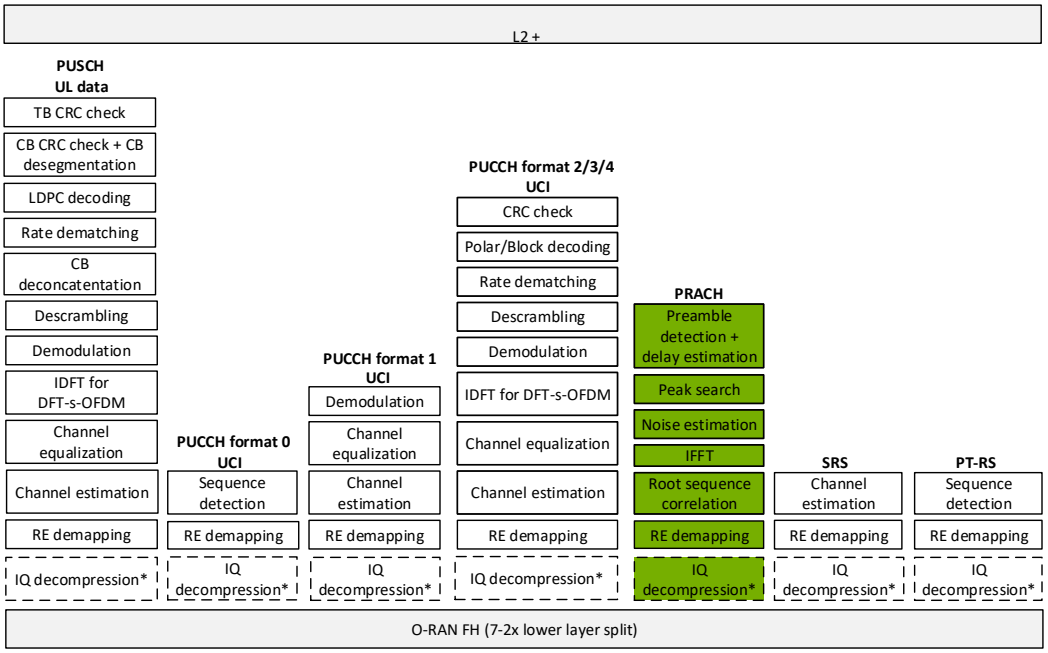
โปรไฟล์ AAL_PUCCH_HIGH-PHY (format 2/3/4) เป็นดังบล็อกสีเขียวในรูปที่ 2.89



รูปที่ 2.89 โปรไฟล์ AAL_PUCCH_HIGH-PHY (format 2/3/4)

6) AAL_PRACH_HIGH-PHY

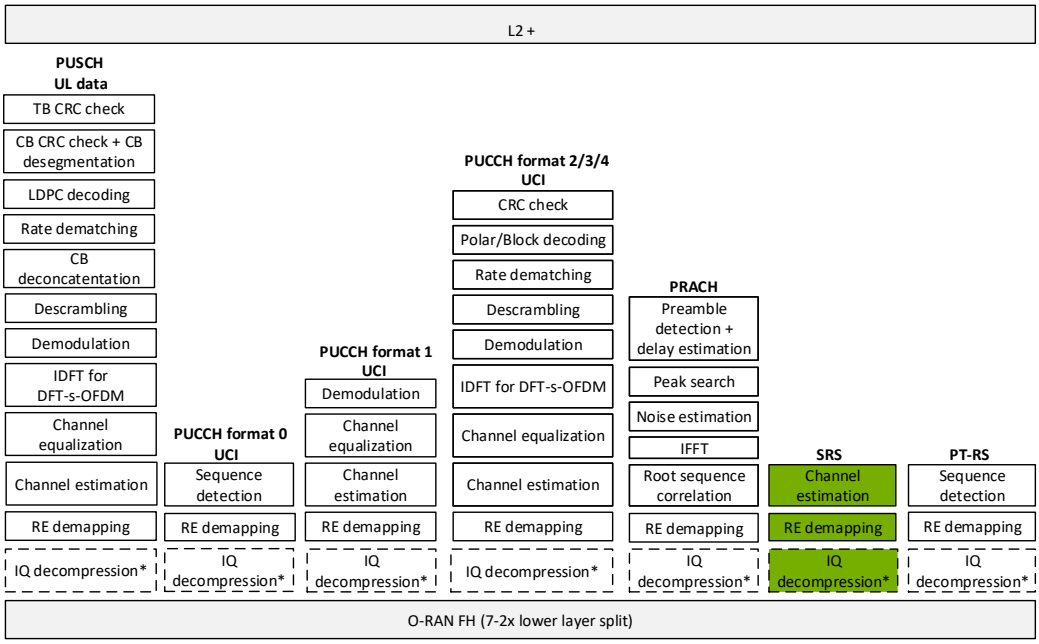
โปรไฟล์ AAL_PRACH_HIGH-PHY เป็นดังบล็อกสีเขียวในรูปที่ 2.90



รูปที่ 2.90 AAL_PRACH_HIGH-PHY

7) AAL_SRS_HIGH-PHY

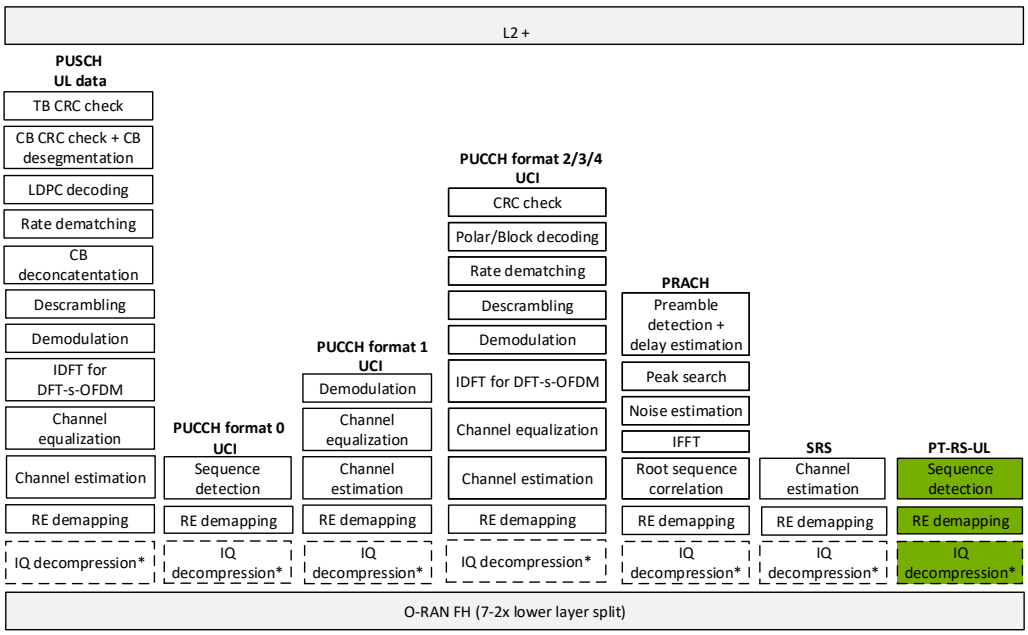
โปรไฟล์ AAL_SRS_HIGH-PHY เป็นดังบล็อกสีเขียวในรูปที่ 2.91



รูปที่ 2.91 โปรไฟล์ AAL_SRS_HIGH-PHY

8) AAL_PT-RS-UL_HIGH-PHY

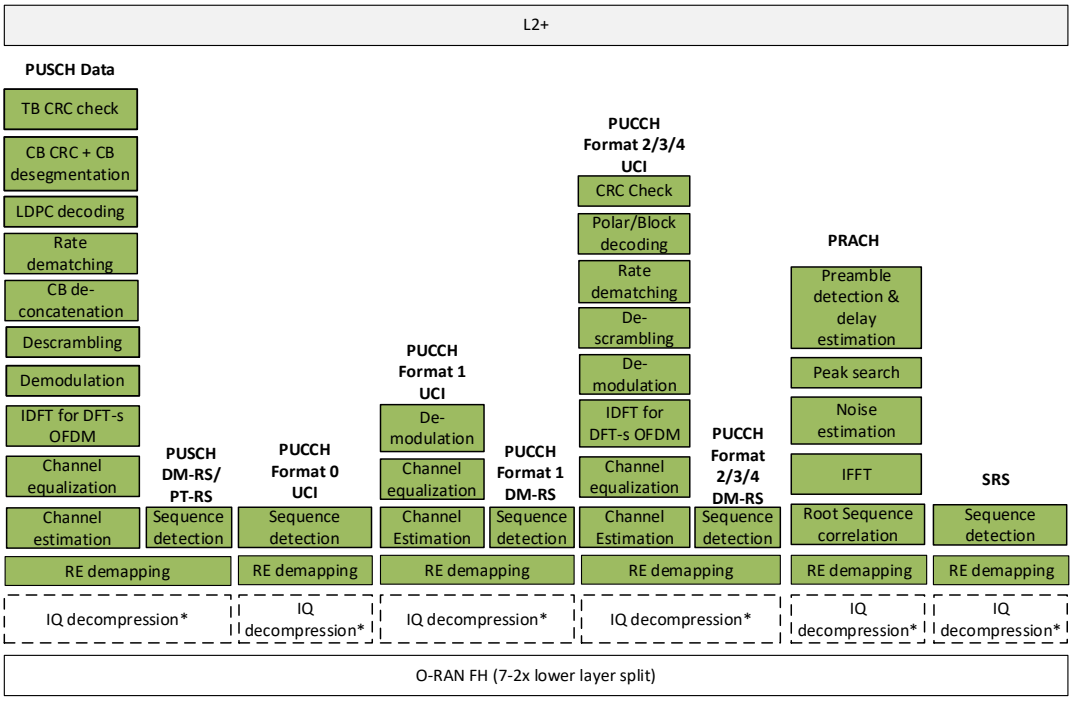
โปรไฟล์ AAL_PT-RS-UL_HIGH-PHY เป็นดังบล็อกสีเขียวในรูปที่ 2.92



รูปที่ 2.92 โปรไฟล์ AAL_PT-RS-UL_HIGH-PHY

9) AAL_UPLINK_HIGH-PHY

โปรไฟล์ AAL_UPLINK_HIGH-PHY เป็นดังบล็อกสีเขียวในรูปที่ 2.93

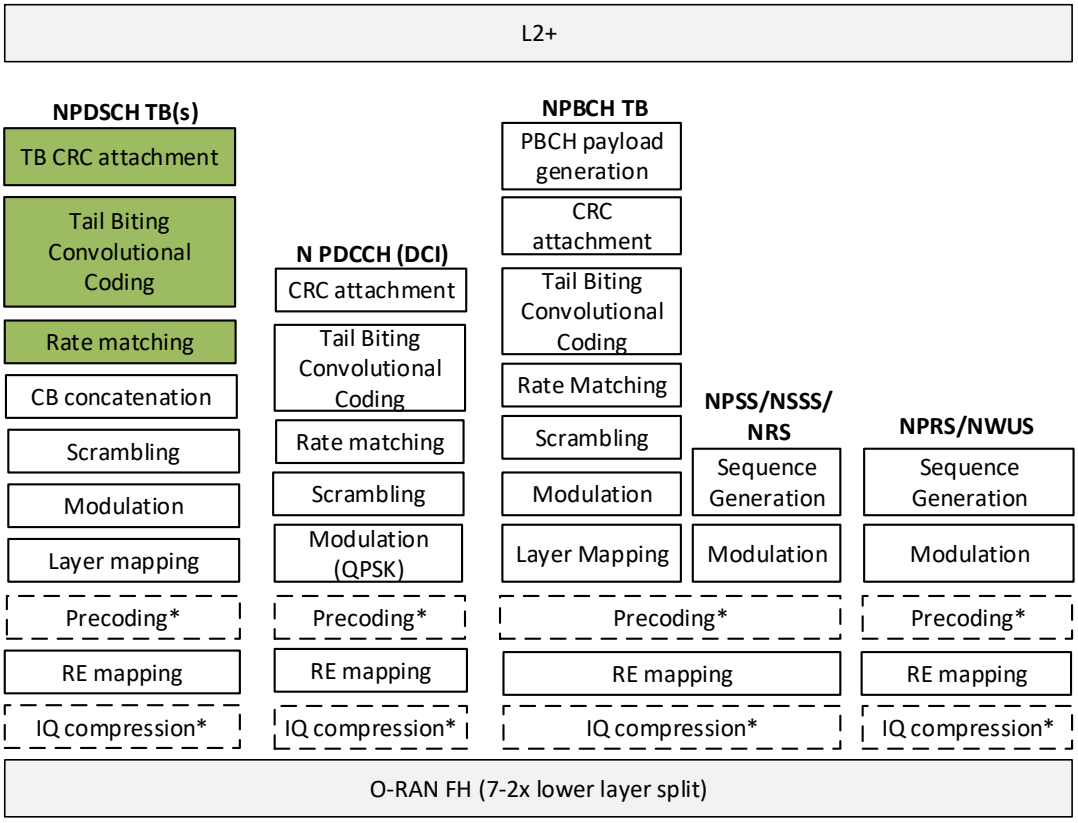


รูปที่ 2.93 โปรไฟล์ AAL_UPLINK_HIGH-PHY

2.8.4.6 โปรไฟล์ของ O-DU AAL (mMTC)

1) AAL_NPDSCH_FEC

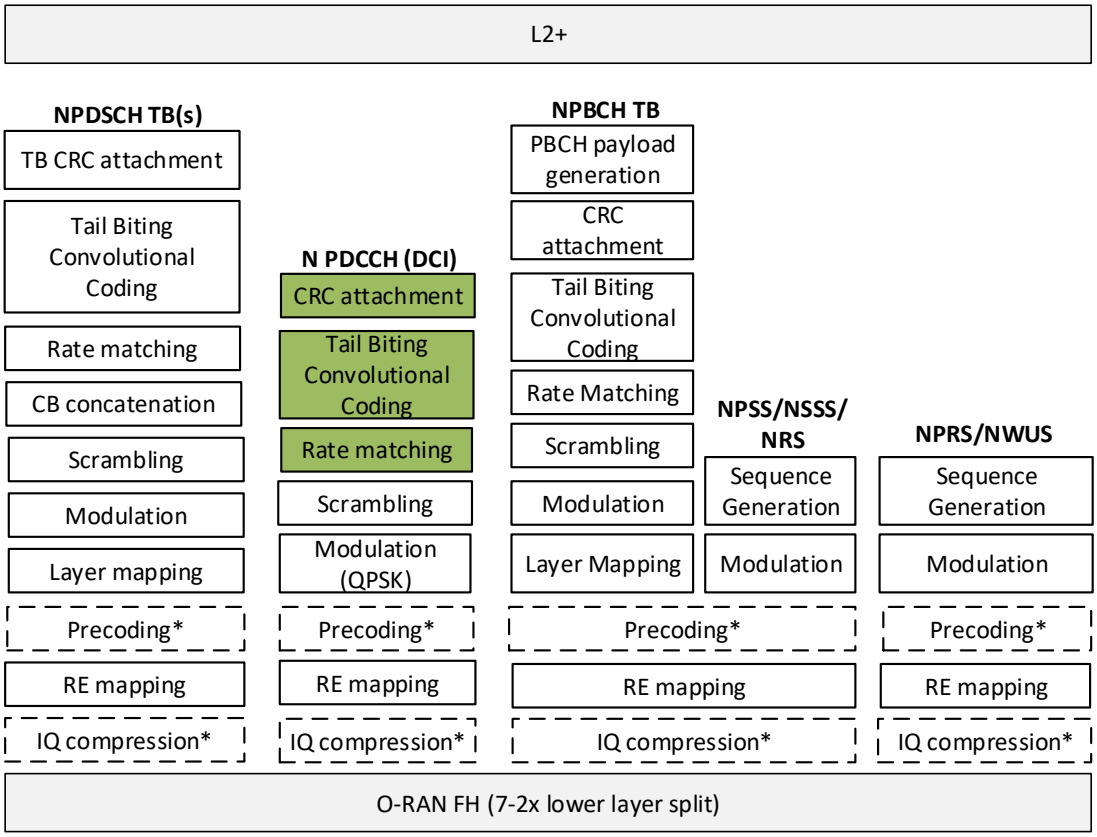
โปรไฟล์ AAL_NPDSCH_FEC เป็นดังบล็อกสีเขียวในรูปที่ 2.94



รูปที่ 2.94 โปรไฟล์ AAL_NPDSCH_FEC

2) AAL_NPDCCH_FEC

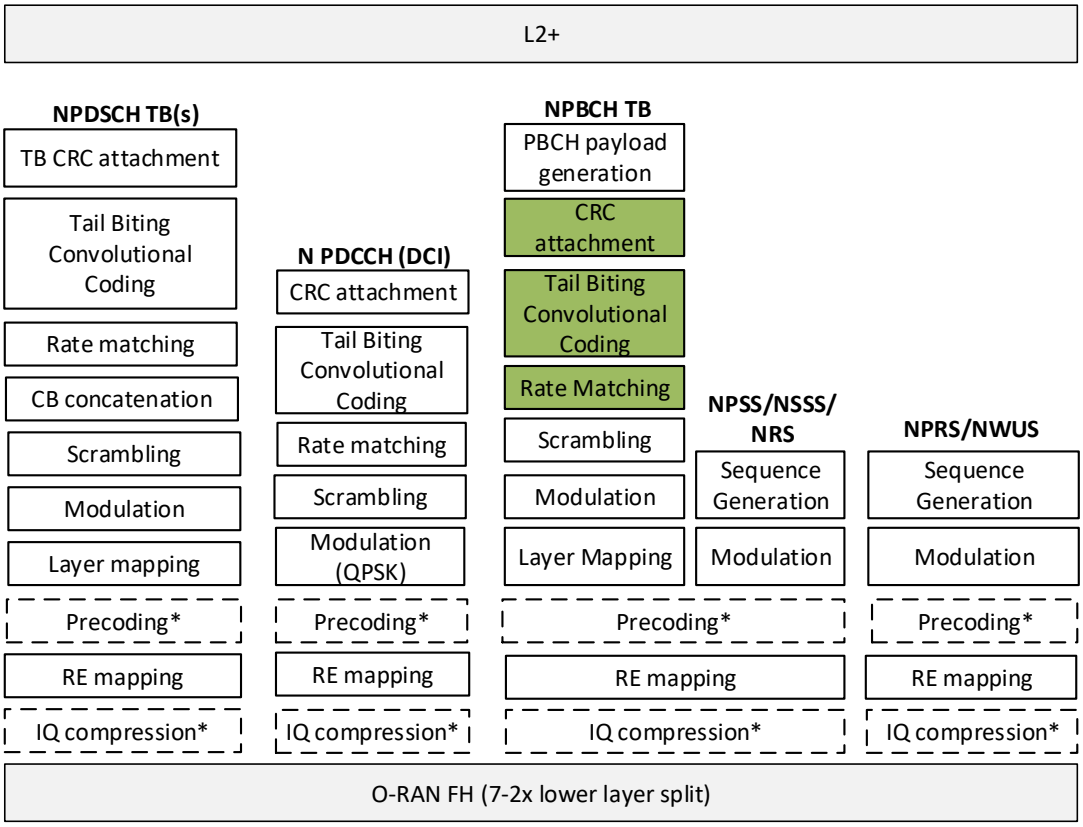
โปรไฟล์ AAL_NPDCCH_FEC เป็นดังบล็อกสีเขียวในรูปที่ 2.95



รูปที่ 2.95 โปรไฟล์ AAL_NPDCCH_FEC

3) AAL_NPBCH_FEC

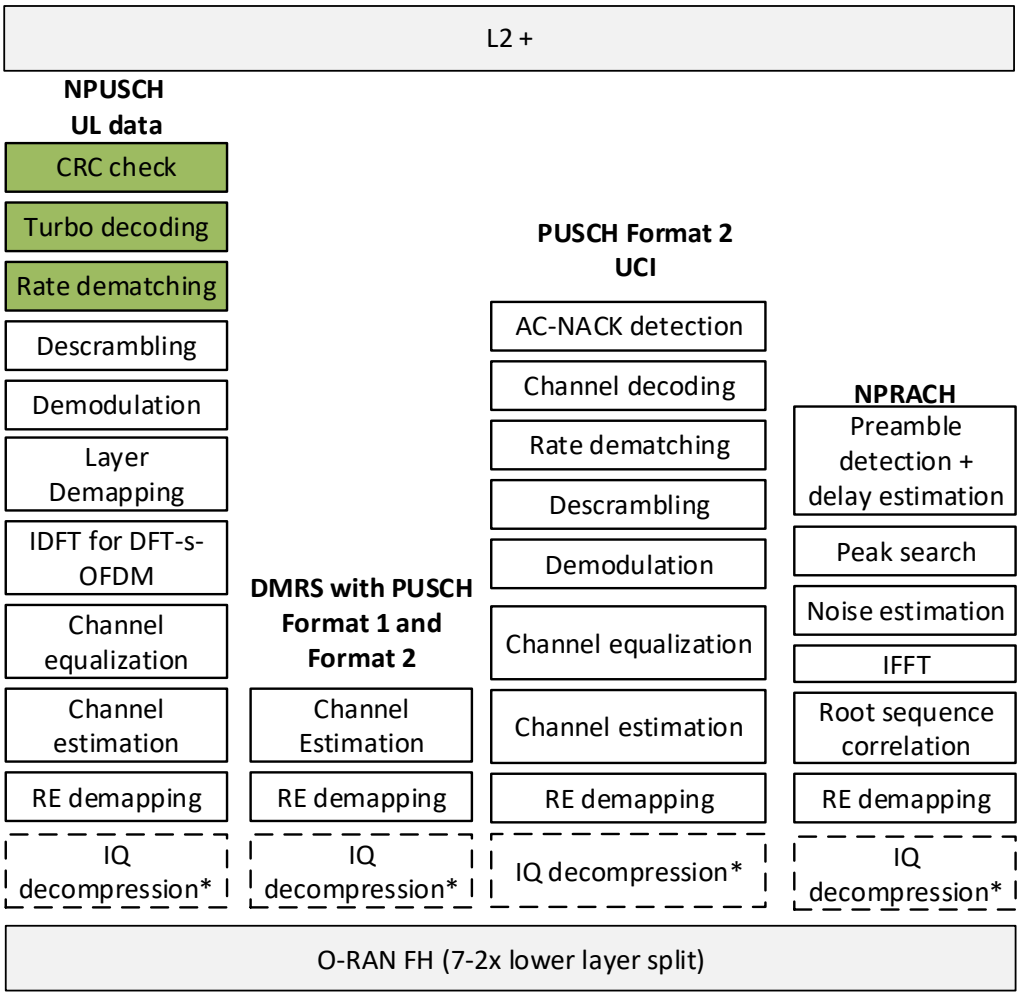
โปรไฟล์ AAL_NPBCH_FEC เป็นดังบล็อกสีเขียวในรูปที่ 2.96



รูปที่ 2.96 โปรไฟล์ AAL_NPBCH_FEC

4) AAL_NPUSCH_FEC

โปรไฟล์ AAL_NPUSCH_FEC เป็นดังบล็อกสีเขียวในรูปที่ 2.97



รูปที่ 2.97 โปรไฟล์ AAL_NPUSCH_FEC

2.8.5 FEC Profile

2.8.5.1 วัตถุประสงค์

เอกสารนี้จะแจกแจงรายละเอียดข้อมูลจำเพาะของส่วนต่อประสาน AAL สำหรับโปรไฟล์ AAL_PUSCH_FEC and AAL_PDSCH_FEC ดังรูปที่ X และ X ตามลำดับ

2.8.5.2 การจัดการและการตั้งค่า AALi

1) การจัดการและการตั้งค่ามีลำดับดังนี้

1.1) นับจำนวน AAL-LPU

1.2) นับ AAL-LPU ที่พร้อมใช้งาน

1.3) จัดเตรียมคิว AAL-LPU

1.4) จัดการปรับแต่งคิว AAL-LPU

2) การจัดการคิว AAL-LPU

เป็นการจัดการคิวให้รองรับการใช้งานร่วมกับ AALi ดังนี้

2.1) เริ่มการทำงาน AAL-LPU

2.2) เริ่มคิว AAL-LPU

2.3) หยุดคิว AAL-LPU

2.4) หยุดการทำงาน AAL-LPU

2.5) ปิดการใช้งาน AAL-LPU

3) การเก็บสถิติ

เป็นการจัดเก็บสถิติของความผิดพลาดในการจัดการคิวของ AAL-LPU ซึ่งต้องมีส่วนต่อประสานสำหรับการรีเซ็ตสถิติดังกล่าวด้วย

2.8.6 คุณลักษณะของโปรไฟล์ AAL FEC

กระบวนการเข้ารหัสของทั้งสองโปรไฟล์จะเป็นไปตามมาตรฐาน 3GPP TS 38.212 และ 38.214

2.8.6.1 O-DU AAL PDSCH FEC

สรุปคุณลักษณะของ O-DU AAL PDSCH FEC ได้ดังนี้

1) สามารถหยุดการทำงานจากการร้องขอภายนอกได้

2) การเลือกชุด CRC ที่ถูกต้อง

3) รองรับการกระจัดกระจายของบัฟเฟอร์อินพุต

4) เรียงต่อเอาต์พุตที่ไม่เต็มไบต์ได้

5) มีการเลือกโหมด CB หรือ TB

พารามิเตอร์หลักของ O-DU AAL PDSCH FEC ที่ต้องใช้สำหรับ AALi เป็นดังนี้

1) ข้อมูลอินพุต

2) อัตรารหัส

3) ชุด CRC ที่ถูกต้อง

4) Rate matching redundancy version, [10]

5) กราฟฐานของ LDPC, [10]

6) Zc [10]

7) Ncb, [10]

8) Qm, {2,4,6,8}. [10]

9) n_filler = K – K’ [10]

10) การเลือกโหมด CB หรือ TB

พารามิเตอร์ O-DU AAL PDSCH FEC สำหรับโหมด CB เป็นดังนี้

1) E, [10]

พารามิเตอร์ O-DU AAL PDSCH FEC สำหรับโหมด TB เป็นดังนี้

1) Ea, r < cab. [10]

2) Eb, r >= cab. [10]

3) จำนวน CB ทั้งหมด

4) ดัชนีบล็อกแรกของ mbuf data โดยมีค่าตั้งต้นเป็น 0

5) cab, จำนวนของ CB ที่ใช้ Ea ก่อนเปลี่ยนเป็น Eb, [0:255]

6) max_cbg, จำนวน CB ต่อ TB {1,2,4,6,8}, [13]

7) ข้อมูลแบบกลุ่มก้อนของ CB [13]

สรุปคุณลักษณะ O-DU AAL PUSCH FEC ได้ดังนี้

1) มีขนาดของ LLR เป็นบิตซึ่งเป็นแบบ 2’complement

2) มีค่า LLR ซึ่งสามารถแปลงเป็นเลขฐานสิบ

3) มีชุด CRC ที่ถูกต้อง

4) สามารถดรอปข้อมูลจากการตรวจสอบ CRC

5) สามารถเลือกการใช้ HARQ

6) สามารถเลือกเอาต์พุตละเอียด

7) สามารถเลือกจำนวนรอบการถอดรหัส

8) รองรับการกระจัดกระจายของบัฟเฟอร์อินพุต

9) เรียงต่อเอาต์พุตที่ไม่เต็มไบต์ได้

10) การบีบอัดของ HARQ การบีบอัดจะเป็น 6 บิตต่อ 1 ไบต์

11) การบีบอัดของ LLR การบีบอัดจะเป็น 6 บิต

12) รองรับการรับอินพุต HARQ จากหน่วยความจำภายในของ AAL-LPU

13) รองรับการส่งเอาต์พุต HARQ ไปยังหน่วยความจำภายในของ AAL-LPU

14) สามารถเพิ่มบิต LLR filler ไปในบัฟเฟอร์หน่วยความจำของ HARQ

15) มีการเลือกโหมด CB หรือ TB

พารามิเตอร์หลักของ O-DU AAL PUSCH FEC ที่ต้องใช้สำหรับ AALi เป็นดังนี้

1) อินพุตสำหรับบัฟเฟอร์หมุนวนเสมือน แทนด้วย 1 ค่า LLR ต่อบิตของ CB หรือ TB

2) บัฟเฟอร์สำหรับการตัดสินใจหยาบขนาด K’ สำหรับแต่ละ CB

3) Rate matching redundancy version, [10]

4) จำนวนรอบการถอดรหัส

5) กราฟฐานของ LDPC [10]

6) Zc, [10]

7) Ncb, [10]

8) Qm, [10]

9) n_filler = K-K', [10]

11) การเลือกโหมด CB หรือ TB

พารามิเตอร์ O-DU AAL PUSCH FEC สำหรับโหมด CB เป็นดังนี้

1) ลำดับเอาต์พุตหลังการปรับอัตรารหัสเป็นบิตหรือ LLR [13]

พารามิเตอร์ O-DU AAL PUSCH FEC สำหรับโหมด TB เป็นดังนี้

1) Ea, r < cab. [10]

2) Eb, r >= cab. [10]

3) จำนวน CB ทั้งหมด

4) ดัชนีบล็อกแรกของ mbuf data โดยมีค่าตั้งต้นเป็น 0

5) cab, จำนวนของ CB ที่ใช้ Ea ก่อนเปลี่ยนเป็น Eb, [0:255]

6) max_cbg, จำนวน CB ต่อ TB {1,2,4,6,8}, [13]

7) ข้อมูลแบบกลุ่มก้อนของ CB [13]

8) ข้อมูลใน CB ที่ผิดพลาดจากการตรวจสอบ CRC

2.8.6.2 นิยามพารามิเตอร์ของ API

นิยามของส่วนต่อประสานและพารามิเตอร์ของ API ภายใต้ DPDK เป็นดังตารางที่ 2.5 และ 2.6 ตามลำดับ

ตารางที่ 2.5 นิยามส่วนต่อประสานของ API

AAL FEC	AAL FEC DPDK API
AAL-LPU	Device ID
AAL Queue	Queue ID
AAL_PDSCH_FEC	RTE_BBDEV_OP_LDPC_ENC
AAL_PUSCH_FEC	RTE_BBDEV_OP_LDPC_DEC

ตารางที่ 2.6 นิยามพารามิเตอร์ของ API

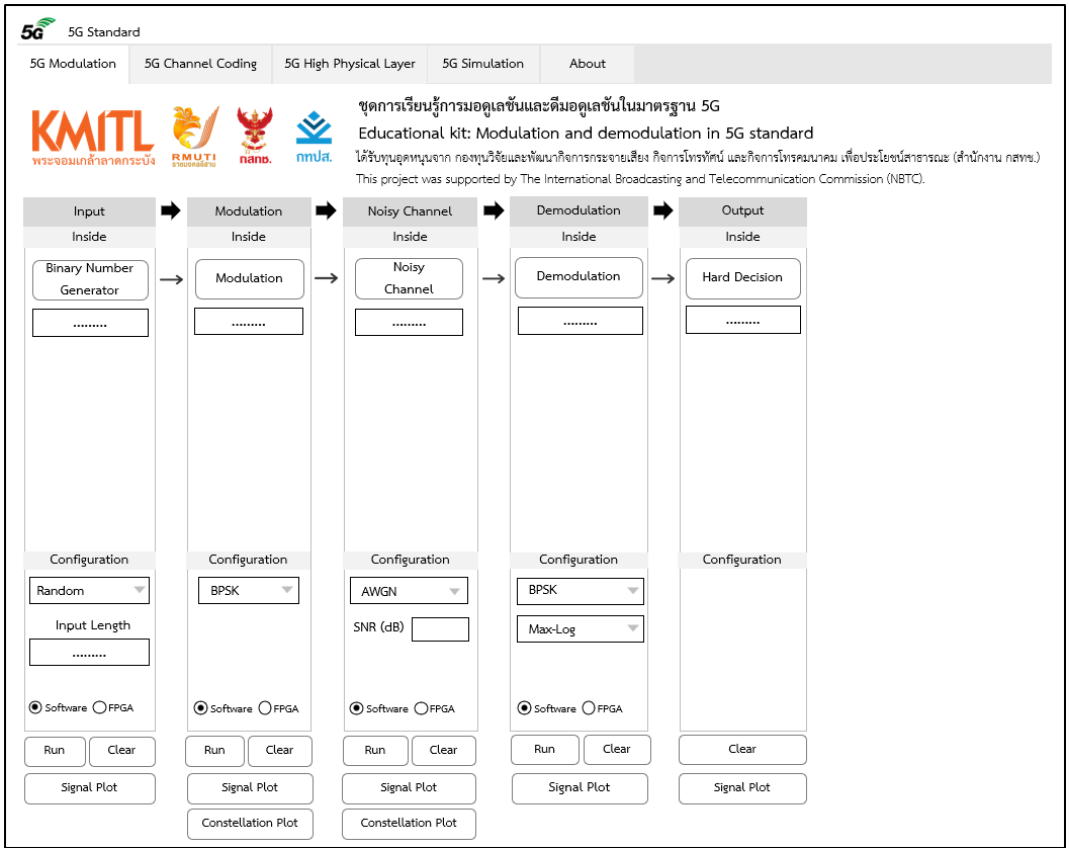
พารามิเตอร์	**DPDK rte_bbdev_op_ldpc_enc**
ข้อมูลอินพุต	struct rte_bbdev_op_data input
อัตรารหัส	struct rte_bbdev_op_data output
ชุด CRC ที่ถูกต้อง	/** Flags from rte_bbdev_op_ldpcenc_flag_bitmasks */ uint32_t op_flags;
Rate matching redundancy version [9]	uint8_t rv_index;
กราฟฐานของ LDPC [10]	uint8_t basegraph;
Zc, [10]	uint16_t z_c;
Ncb, [10]	uint16_t n_cb;
Qm, {2,4,6,8}. [10]	uint8_t q_m;
n_filler = K – K' [10]	uint16_t n_filler;
การเลือกโหมด CB หรือ TB	/** [0 - TB : 1 - CB] */ uint8_t code_block_mode;

บทที่ 3
ระเบียบวิธีวิจัย

3.1 การออกแบบชุดการเรียนรู้

3.1.1 การออกแบบชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G

โครงการนี้ออกแบบซอฟต์แวร์ GUI ชุดการเรียนรู้การมอดูเลชันและการดีมอดูเลชันมาตรฐาน 5G โดยใช้ภาษา Python เพื่อให้ผู้ใช้งานสามารถเรียนรู้กระบวนการมอดูเลชันและดีมอดูเลชันได้โดยง่าย ซึ่งผู้ใช้งานจะสามารถเห็นกระบวนการทำงานตั้งแต่การสร้างข้อมูลแบบสุ่มหรือนำเข้าข้อมูลอินพุต การมอดูเลชัน การส่งผ่านช่องสัญญาณรบกวน และการดีมอดูเลชัน นอกจากนี้ผู้ใช้งานสามารถดูผลลัพธ์ ณ จุดต่าง ๆ รวมถึงสามารถเรียกดูกราฟผลลัพธ์ของสัญญาณและแผนภาพคอนสเตลเลชันของการมอดูเลชันได้ โครงร่างการออกแบบชุดการเรียนรู้การมอดูเลชันและการดีมอดูเลชันมาตรฐาน 5G แสดงดังรูปที่ 3.1



รูปที่ 3.1 โครงร่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G

จาก รูปที่ 3.1 ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G ประกอบด้วย 5 องค์ประกอบ ได้แก่ อินพุต (Input) การมอดูเลชัน (Modulation) ช่องสัญญาณรบกวน (Noisy Channel) การดีมอดูเลชัน (Demodulation) และเอาต์พุต (Output) โดยแต่ละองค์ประกอบมีรายละเอียดดังนี้

1) อินพุต เป็นส่วนสร้างและแสดงผลอินพุต ซึ่งประกอบด้วยส่วนย่อย 4 ส่วน ดังนี้

1.1) ส่วนแสดงผลภายใน (Inside) ประกอบด้วย

- บล็อกไดอะแกรม (Block Diagram) สำหรับแสดงบล็อกอินพุต
- กล่องข้อความ (Text Box) สำหรับแสดงอินพุต

1.2) ส่วนการกำหนดค่า (Configuration) ประกอบด้วย

- กล่องตัวเลือก (Combo Box) สำหรับเลือกวิธีการป้อนอินพุต ผู้ใช้สามารถเลือกวิธีการป้อนอินพุตได้ 3 วิธี ได้แก่ การสร้างอินพุตแบบสุ่ม (Random) การป้อนอินพุตด้วยตนเอง (Fill-in) และการอัปโหลดรูปภาพ (Upload Image)
- กล่องป้อนข้อความ (Entry Box) สำหรับป้อนความยาวของอินพุตหรือป้อนอินพุต กรณีที่ผู้ใช้เลือกการสร้างอินพุตแบบสุ่มผู้ใช้สามารถป้อนความยาวของอินพุตตามต้องการได้ กรณีที่ผู้ใช้เลือกการป้อนอินพุตด้วยตัวเองผู้ใช้จะต้องป้อนอินพุตลงในกล่องข้อความ และกรณีที่ผู้ใช้เลือกการอัปโหลดรูปภาพ ระบบจะแปลงรูปภาพเป็นบิตข้อมูลแบบไบนารีและกำหนดความยาวตามบิตข้อมูลของรูปภาพในกล่องข้อความโดยอัตโนมัติ
- ปุ่มวิทยุ (Radio Button) สำหรับเลือกช่องทางการสร้างอินพุต ผู้ใช้สามารถเลือกช่องทางการสร้างอินพุตได้ 2 ช่องทาง ได้แก่ การสร้างโดยใช้ซอฟต์แวร์ (Software) และการสร้างโดยใช้อุปกรณ์ FPGA

1.3) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Run” สำหรับดำเนินการสร้างอินพุต
- ปุ่ม “Clear” สำหรับล้างค่าอินพุต

1.4) ส่วนแสดงกราฟ ประกอบด้วย

- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของสัญญาณอินพุต

2) การมอดูเลชัน เป็นส่วนการมอดูเลชันสัญญาณตามมาตรฐาน 5G และแสดงผลลัพธ์ของการมอดูเลชัน ซึ่งประกอบด้วยส่วนย่อย 4 ส่วน ดังนี้

2.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- บล็อกไดอะแกรม สำหรับแสดงบล็อกการมอดูเลชันสัญญาณ
- กล่องข้อความ สำหรับแสดงผลลัพธ์ที่ได้จากการมอดูเลชันสัญญาณ

2.2) ส่วนการกำหนดค่า ประกอบด้วย

- กล่องตัวเลือก สำหรับเลือกรูปแบบการมอดูเลชันสัญญาณมาตรฐาน 5G ผู้ใช้สามารถเลือกรูปแบบการมอดูเลชันได้ 7 รูปแบบ ได้แก่ BPSK, π/2-BPSK, QPSK, 16QAM, 64QAM, 256QAM และ 1024QAM
- ปุ่มวิทยุ สำหรับเลือกช่องทางการมอดูเลชันสัญญาณ ผู้ใช้สามารถเลือกช่องทางการมอดูเลชันสัญญาณได้ 2 ช่องทาง ได้แก่ การมอดูเลชันโดยใช้ซอฟต์แวร์ และใช้อุปกรณ์ FPGA

2.3) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Run” สำหรับดำเนินการมอดูเลชันสัญญาณ
- ปุ่ม “Clear” สำหรับล้างค่าผลลัพธ์ของการมอดูเลชันสัญญาณ

2.4) ส่วนแสดงกราฟ ประกอบด้วย

- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของสัญญาณมอดูเลชัน
- ปุ่ม “Constellation Plot” สำหรับแสดงแผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชัน

3) ช่องสัญญาณ เป็นส่วนสร้างช่องสัญญาณรบกวนและแสดงผลลัพธ์ของสัญญาณมอดูเลชันที่ผ่านช่องสัญญาณรบกวน ซึ่งประกอบด้วยส่วนย่อย 4 ส่วน ดังนี้

3.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- บล็อกไดอะแกรม สำหรับแสดงบล็อกช่องสัญญาณรบกวน
- กล่องข้อความ สำหรับแสดงผลลัพธ์สัญญาณมอดูเลชันที่ผ่านช่องสัญญาณรบกวน

3.2) ส่วนการกำหนดค่า ประกอบด้วย

- กล่องตัวเลือก สำหรับเลือกรูปแบบสัญญาณรบกวน ผู้ใช้สามารถเลือกรูปแบบของช่องสัญญาณรบกวนได้ 2 รูปแบบ ได้แก่ AWGN Channel และ Fading Channel
- กล่องป้อนข้อความ สำหรับป้อนค่าอัตราส่วนของสัญญาณต่อสัญญาณรบกวน (Signal-to-Noise Ratio: SNR)
- ปุ่มวิทยุ สำหรับเลือกช่องทางการสร้างสัญญาณรบกวนและรวมสัญญาณมอดูเลชันเข้ากับสัญญาณรบกวน ผู้ใช้สามารถเลือกช่องทางการสร้างสัญญาณรบกวนและรวมสัญญาณมอดูเลชันได้ 2 ช่องทาง ได้แก่ การสร้างโดยใช้ซอฟต์แวร์ และการสร้างโดยใช้อุปกรณ์ FPGA

3.3) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Run” สำหรับดำเนินการสร้างสัญญาณรบกวนและรวมสัญญาณมอดูเลชันเข้ากับสัญญาณรบกวน
- ปุ่ม “Clear” สำหรับล้างค่าผลลัพธ์สัญญาณมอดูเลชันที่ผ่านช่องสัญญาณรบกวน

3.4) ส่วนแสดงกราฟ ประกอบด้วย

- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของสัญญาณมอดูเลชันที่ผ่านช่องสัญญาณรบกวน
- ปุ่ม “Constellation Plot” สำหรับแสดงแผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันที่ผ่านช่องสัญญาณรบกวน

4) การดีมอดูเลชัน เป็นส่วนการดีมอดูเลชันสัญญาณตามมาตรฐาน 5G และแสดงผลลัพธ์ของการดีมอดูเลชันสัญญาณ ซึ่งประกอบด้วยส่วนย่อย 4 ส่วน ดังนี้

4.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- บล็อกไดอะแกรม สำหรับแสดงบล็อกการดีมอดูเลชันสัญญาณ
- กล่องข้อความ สำหรับแสดงผลลัพธ์ที่ได้จากการดีมอดูเลชันสัญญาณ

4.2) ส่วนการกำหนดค่า ประกอบด้วย

- กล่องตัวเลือก สำหรับเลือกรูปแบบการดีมอดูเลชันสัญญาณมาตรฐาน 5G ผู้ใช้สามารถเลือกรูปแบบการดีมอดูเลชันได้ 7 รูปแบบ ได้แก่ BPSK, π/2-BPSK, QPSK, 16QAM, 64QAM, 256QAM และ 1024QAM
- กล่องตัวเลือก สำหรับเลือกอัลกอริทึมการดีมอดูเลชันสัญญาณ ผู้ใช้สามารถเลือกอัลกอริทึมการดีมอดูเลชันได้ 2 อัลกอริทึม ได้แก่ Mag-Log และ LUT
- ปุ่มวิทยุ สำหรับเลือกช่องทางการดีมอดูเลชันสัญญาณ ผู้ใช้สามารถเลือกช่องทางการดีมอดูเลชันสัญญาณได้ 2 ช่องทาง ได้แก่ การดีมอดูเลชันโดยใช้ซอฟต์แวร์ และการดีมอดูเลชันโดยใช้อุปกรณ์ FPGA

4.3) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Run” สำหรับดำเนินการเข้าดีมอดูเลชันสัญญาณ
- ปุ่ม “Clear” สำหรับล้างค่าผลลัพธ์ของการดีมอดูเลชันสัญญาณ

4.4) ส่วนแสดงกราฟ ประกอบด้วย

- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของผลลัพธ์ที่ได้จากการดีมอดูเลชันสัญญาณ

5) เอาต์พุต เป็นแสดงผลลัพธ์ของการดีมอดูเลชัน ซึ่งประกอบด้วยส่วนย่อย 3 ส่วน ดังนี้

5.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- บล็อกไดอะแกรม สำหรับแสดงบล็อกเอาต์พุต

- กล่องข้อความ สำหรับแสดงผลลัพธ์ที่ได้จากการตัดสินใจบิต (Hard Decision)

5.2) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Clear” สำหรับล้างค่าของผลลัพธ์

5.3) ส่วนแสดงกราฟ ประกอบด้วย

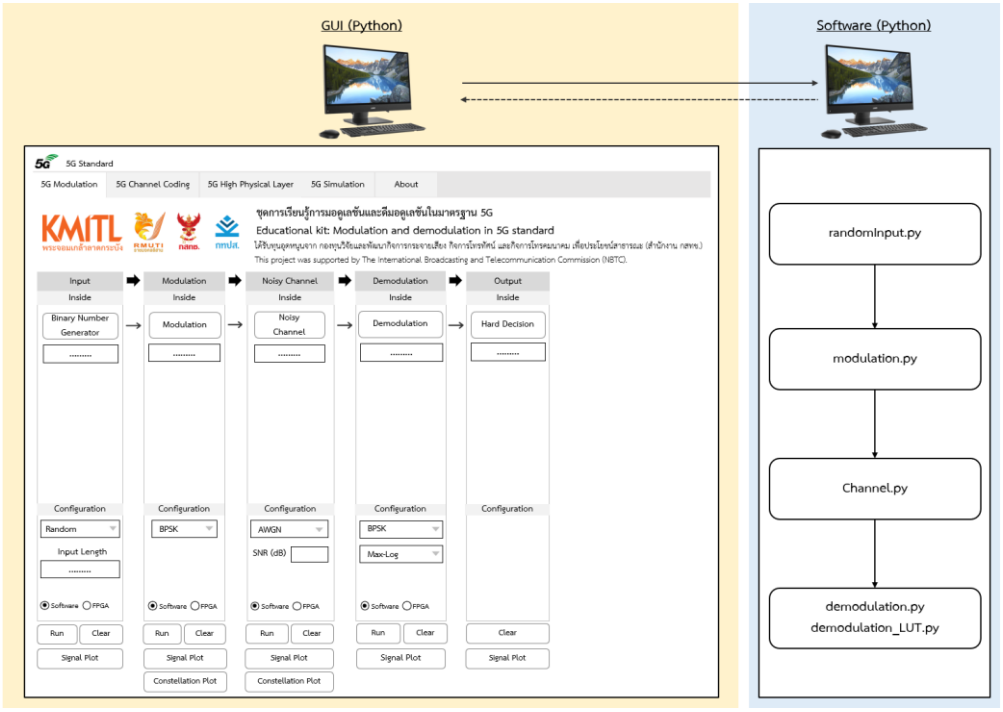
- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของผลลัพธ์

นอกจากนี้ ผู้ใช้สามารถเลือกการทำงานของแต่ละองค์ประกอบด้วยซอฟต์แวร์ หรืออุปกรณ์ FPGA ได้โดยการเลือกปุ่มวิทยุที่อยู่ในส่วนการกำหนดค่า สำหรับการทำงานด้วยซอฟต์แวร์แต่ละองค์ประกอบจะเรียกใช้ Module การทำงานภาษา Python ที่สอดคล้องกัน ดังแสดงในรูปที่ 3.2 โดยแต่ละองค์ประกอบจะเรียกใช้ Module ดังตารางที่ 3.1

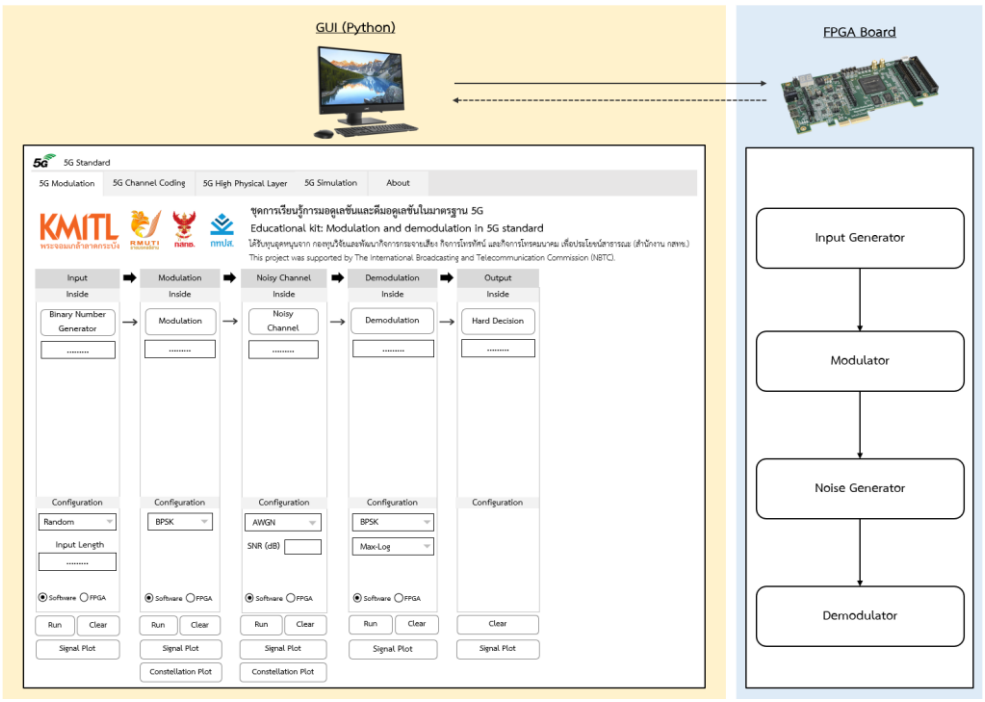
ตารางที่ 3.1 การเรียกใช้ซอฟต์แวร์ Module ของแต่ละองค์ประกอบของชุดการเรียนรู้การมอดูเลชันและการดีมอดูเลชันมาตรฐาน 5G

องค์ประกอบ	**Module**
อินพุต (Input)	randomInput.py
การมอดูเลชัน (Modulation)	modulation.py
ช่องสัญญาณ (Noisy Channel)	Channel.py
การดีมอดูเลชัน (Demodulation)	deodulation.py demodulation_LUT.py

สำหรับการทำงานด้วยอุปกรณ์ FPGA แต่ละองค์ประกอบจะเรียกใช้ Module การทำงานบนอุปกรณ์ FPGA ที่สอดคล้องกัน โดยส่งข้อมูลการกำหนดค่าบน GUI ไปยังอุปกรณ์ FPGA และรับเอาต์พุตผ่าน Serial Port ดังแสดงในรูปที่ 3.3 ทั้งนี้ ผู้ใช้สามารถเลือกการทำงานด้วยซอฟต์แวร์ในบางองค์ประกอบและเลือกการทำงานด้วยอุปกรณ์ FPGA ในบางองค์ประกอบได้



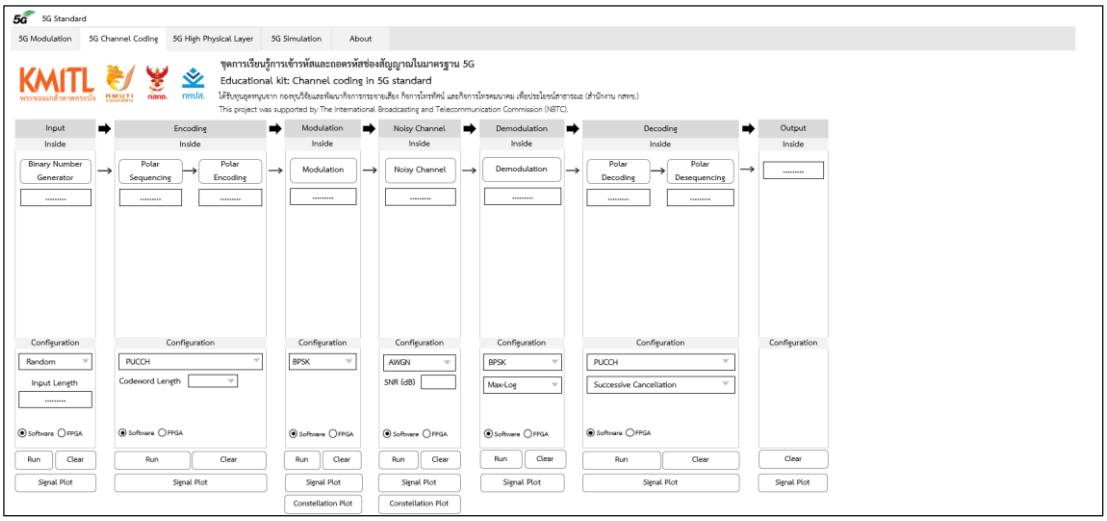
รูปที่ 3.2 ชุดการเรียนรู้การมอดูเลชันและการดีมอดูเลชันมาตรฐาน 5G เชื่อมต่อกับซอฟต์แวร์



รูปที่ 3.3 ชุดการเรียนรู้การมอดูเลชันและการดีมอดูเลชันมาตรฐาน 5G เชื่อมต่อกับอุปกรณ์ FPGA

3.1.2 การออกแบบชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G

โครงการนี้ออกแบบซอฟต์แวร์ GUI ชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G โดยใช้ภาษา Python เพื่อให้ผู้ใช้งานสามารถเรียนรู้กระบวนการเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G ได้โดยง่าย ซึ่งผู้ใช้งานจะสามารถเห็นกระบวนการทำงานตั้งแต่การสร้างข้อมูลแบบสุ่มหรือนำเข้าข้อมูลอินพุต การเข้ารหัสช่องสัญญาณ การมอดูเลชัน การส่งผ่านช่องสัญญาณรบกวน การดีมอดูเลชัน และการถอดรหัสช่องสัญญาณ นอกจากนี้ผู้ใช้งานสามารถดูผลลัพธ์ ณ จุดต่าง ๆ รวมถึงสามารถเรียกดูกราฟผลลัพธ์ของสัญญาณและแผนภาพคอนสเตลเลชันของการมอดูเลชันได้ โครงร่างการออกแบบชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G แสดงดังรูปที่ 3.4



รูปที่ 3.4 โครงร่างชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G

จากรูปที่ 3.4 ชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G ประกอบด้วย 7 องค์ประกอบ ได้แก่ อินพุต (Input) การเข้ารหัสช่องสัญญาณ (Encoding) การมอดูเลชัน (Modulation) ช่องสัญญาณ (Noisy Channel) การดีมอดูเลชัน (Demodulation) การถอดรหัสช่องสัญญาณ (Decoding) และเอาต์พุต (Output) โดยแต่ละองค์ประกอบมีรายละเอียดดังนี้

1) อินพุต เป็นส่วนสร้างและแสดงผลอินพุต ซึ่งประกอบด้วยส่วนย่อย 4 ส่วน ดังนี้

1.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- บล็อกไดอะแกรม สำหรับแสดงบล็อกอินพุต
- กล่องข้อความ สำหรับแสดงอินพุต

1.2) ส่วนการกำหนดค่า ประกอบด้วย

- กล่องตัวเลือก สำหรับเลือกวิธีการป้อนอินพุต ผู้ใช้สามารถเลือกวิธีการป้อนอินพุตได้ 3 วิธี ได้แก่ การสร้างอินพุตแบบสุ่ม การป้อนอินพุตด้วยตนเอง และการอัปโหลดรูปภาพ

- กล่องป้อนข้อความ สำหรับป้อนความยาวของอินพุตหรือป้อนอินพุต กรณีที่ผู้ใช้เลือกการสร้างอินพุตแบบสุ่มผู้ใช้สามารถป้อนความยาวของอินพุตตามต้องการได้ กรณีที่ผู้ใช้เลือกการป้อนอินพุตด้วยตัวเองผู้ใช้จะต้องป้อนอินพุตลงในกล่องข้อความ และกรณีที่ผู้ใช้เลือกการอัปโหลดรูปภาพ ระบบจะแปลงรูปภาพเป็นบิตข้อมูลแบบไบนารีและกำหนดความยาวตามบิตข้อมูลของรูปภาพในกล่องข้อความโดยอัตโนมัติ
- ปุ่มวิทยุ สำหรับเลือกช่องทางการสร้างอินพุต ผู้ใช้สามารถเลือกช่องทางการสร้างอินพุตได้ 2 ช่องทาง ได้แก่ การสร้างโดยใช้ซอฟต์แวร์ และการสร้างโดยใช้อุปกรณ์ FPGA

1.3) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Run” สำหรับดำเนินการสร้างอินพุต
- ปุ่ม “Clear” สำหรับล้างค่าอินพุต

1.4) ส่วนแสดงกราฟ ประกอบด้วย

- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของสัญญาณอินพุต

2) การเข้ารหัสช่องสัญญาณ เป็นส่วนการเข้ารหัสช่องสัญญาณตามมาตรฐาน 5G และแสดงผลลัพธ์ของการเข้ารหัสช่องสัญญาณ ซึ่งประกอบด้วยส่วนย่อย 4 ส่วน ดังนี้

2.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- บล็อกไดอะแกรม สำหรับแสดงกระบวนการเข้ารหัสช่องสัญญาณ
- กล่องข้อความ สำหรับแสดงผลลัพธ์ที่ได้ในแต่ละขั้นตอนของกระบวนการเข้ารหัสช่องสัญญาณ

2.2) ส่วนการกำหนดค่า ประกอบด้วย

- กล่องตัวเลือก สำหรับเลือกช่องสัญญาณกายภาพ (Physical Channel) ผู้ใช้สามารถเลือกช่องสัญญาณกายภาพได้ 5 ช่องสัญญาณ ได้แก่ Physical Uplink Share Channel (PUSCH), Physical Uplink Control Channel (PUCCH), Physical Broadcast Channel (PBCH), Physical Downlink Share Channel (PDSCH) และ Physical Downlink Control Channel (PDCCH)
- กล่องตัวเลือก สำหรับเลือกพารามิเตอร์ในการเข้ารหัสช่องสัญญาณ สำหรับกรณี PUSCH และ PDSCH จะเป็นการเลือกกราฟฐาน (Base Graph) และสำหรับกรณี PUCCH PDCCH และ PBCH จะเป็นการเลือกความยาวของคำรหัส (Codeword Length)
- ปุ่มวิทยุ สำหรับเลือกช่องทางการเข้ารหัสช่องสัญญาณ ผู้ใช้สามารถเลือกช่องทางการเข้ารหัสช่องสัญญาณได้ 2 ช่องทาง ได้แก่ การเข้ารหัสโดยใช้ซอฟต์แวร์ และการเข้ารหัสโดยใช้อุปกรณ์ FPGA

2.3) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Run” สำหรับดำเนินการเข้ารหัสช่องสัญญาณ
- ปุ่ม “Clear” สำหรับล้างค่าผลลัพธ์ของการเข้ารหัสช่องสัญญาณ

2.4) ส่วนแสดงกราฟ ประกอบด้วย

- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของผลลัพธ์ที่ได้จากการเข้ารหัสช่องสัญญาณ

3) การมอดูเลชัน เป็นส่วนการมอดูเลชันสัญญาณตามมาตรฐาน 5G และแสดงผลลัพธ์ของการมอดูเลชัน ซึ่งประกอบด้วยส่วนย่อย 4 ส่วน ดังนี้

3.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- บล็อกไดอะแกรม สำหรับแสดงบล็อกการมอดูเลชันสัญญาณ
- กล่องข้อความ สำหรับแสดงผลลัพธ์ที่ได้จากการมอดูเลชันสัญญาณ

3.2) ส่วนการกำหนดค่า ประกอบด้วย

- กล่องตัวเลือก สำหรับเลือกรูปแบบการมอดูเลชันสัญญาณมาตรฐาน 5G ผู้ใช้สามารถเลือกรูปแบบการมอดูเลชันได้ 7 รูปแบบ ได้แก่ BPSK, π/2-BPSK, QPSK, 16QAM, 64QAM, 256QAM และ 1024QAM อย่างไรก็ตาม ตัวเลือกของรูปแบบการมอดูเลชันจะเปลี่ยนแปลงตามช่องสัญญาณกายภาพที่เลือกในส่วนการกำหนดค่าของการเข้ารหัสช่องสัญญาณ
- ปุ่มวิทยุ สำหรับเลือกช่องทางการมอดูเลชันสัญญาณ ผู้ใช้สามารถเลือกช่องทางการมอดูเลชันสัญญาณได้ 2 ช่องทาง ได้แก่ การมอดูเลชันโดยใช้ซอฟต์แวร์ และการมอดูเลชันโดยใช้อุปกรณ์ FPGA

3.3) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Run” สำหรับดำเนินการมอดูเลชันสัญญาณ
- ปุ่ม “Clear” สำหรับล้างค่าผลลัพธ์ของการมอดูเลชันสัญญาณ

3.4) ส่วนแสดงกราฟ ประกอบด้วย

- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของสัญญาณมอดูเลชัน
- ปุ่ม “Constellation Plot” สำหรับแสดงแผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชัน

4) ช่องสัญญาณ เป็นส่วนสร้างช่องสัญญาณรบกวนและแสดงผลลัพธ์ของสัญญาณมอดูเลชันที่ผ่านช่องสัญญาณรบกวน ซึ่งประกอบด้วยส่วนย่อย 4 ส่วน ดังนี้

4.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- บล็อกไดอะแกรม สำหรับแสดงบล็อกช่องสัญญาณรบกวน

- กล่องข้อความ สำหรับแสดงผลลัพธ์สัญญาณมอดูเลชันที่ผ่านช่องสัญญาณรบกวน

4.2) ส่วนการกำหนดค่า ประกอบด้วย

- กล่องตัวเลือก สำหรับเลือกรูปแบบสัญญาณรบกวน ผู้ใช้สามารถเลือกรูปแบบของช่องสัญญาณรบกวนได้ 2 รูปแบบ ได้แก่ AWGN Channel และ Fading Channel
- กล่องป้อนข้อความ สำหรับป้อนค่าอัตราส่วนของสัญญาณต่อสัญญาณรบกวน
- ปุ่มวิทยุ สำหรับเลือกช่องทางการสร้างสัญญาณรบกวนและรวมสัญญาณมอดูเลชันเข้ากับสัญญาณรบกวน ผู้ใช้สามารถเลือกช่องทางการสร้างสัญญาณรบกวนและรวมสัญญาณมอดูเลชันได้ 2 ช่องทาง ได้แก่ การสร้างโดยใช้ซอฟต์แวร์ และการสร้างโดยใช้ FPGA

4.3) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Run” สำหรับดำเนินการสร้างสัญญาณรบกวนและรวมสัญญาณมอดูเลชันเข้ากับสัญญาณรบกวน
- ปุ่ม “Clear” สำหรับล้างค่าผลลัพธ์สัญญาณมอดูเลชันที่ผ่านช่องสัญญาณรบกวน

4.4) ส่วนแสดงกราฟ ประกอบด้วย

- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของสัญญาณมอดูเลชันที่ผ่านช่องสัญญาณรบกวน
- ปุ่ม “Constellation Plot” สำหรับแสดงแผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันที่ผ่านช่องสัญญาณรบกวน

5) การดีมอดูเลชัน เป็นส่วนการดีมอดูเลชันสัญญาณตามมาตรฐาน 5G และแสดงผลลัพธ์ของการดีมอดูเลชันสัญญาณ ซึ่งประกอบด้วยส่วนย่อย 4 ส่วน ดังนี้

5.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- บล็อกไดอะแกรม สำหรับแสดงบล็อกการดีมอดูเลชันสัญญาณ
- กล่องข้อความ สำหรับแสดงผลลัพธ์ที่ได้จากการดีมอดูเลชันสัญญาณ

5.2) ส่วนการกำหนดค่า ประกอบด้วย

- กล่องตัวเลือก สำหรับเลือกรูปแบบการดีมอดูเลชันสัญญาณมาตรฐาน 5G ผู้ใช้สามารถเลือกรูปแบบการดีมอดูเลชันได้ 7 รูปแบบ ได้แก่ BPSK, π/2-BPSK, QPSK, 16QAM, 64QAM, 256QAM และ 1024QAM อย่างไรก็ตาม ตัวเลือกของรูปแบบการดีมอดูเลชันจะเปลี่ยนแปลงตามช่องสัญญาณกายภาพที่เลือกในส่วนการกำหนดค่าของการเข้ารหัสช่องสัญญาณ และสอดคล้องกับการมอดูเลชัน

- กล่องตัวเลือก สำหรับเลือกอัลกอริทึมการดีมอดูเลชันสัญญาณ ผู้ใช้สามารถเลือกอัลกอริทึมการดีมอดูเลชันได้ 2 อัลกอริทึม ได้แก่ Mag-Log และ LUT
- ปุ่มวิทยุ สำหรับเลือกช่องทางการดีมอดูเลชันสัญญาณ ผู้ใช้สามารถเลือกช่องทางการดีมอดูเลชันสัญญาณได้ 2 ช่องทาง ได้แก่ การดีมอดูเลชันโดยใช้ซอฟต์แวร์ และการดีมอดูเลชันโดยใช้อุปกรณ์ FPGA

5.3) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Run” สำหรับดำเนินการดีมอดูเลชันสัญญาณ
- ปุ่ม “Clear” สำหรับล้างค่าผลลัพธ์ของการดีมอดูเลชันสัญญาณ

5.4) ส่วนแสดงกราฟ ประกอบด้วย

- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของผลลัพธ์ที่ได้จากการดีมอดูเลชันสัญญาณ

6) การถอดรหัสช่องสัญญาณ เป็นส่วนการถอดรหัสช่องสัญญาณตามมาตรฐาน 5G และแสดงผลลัพธ์ของการถอดรหัสช่องสัญญาณ ซึ่งประกอบด้วยส่วนย่อย 4 ส่วน ดังนี้

6.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- บล็อกไดอะแกรม สำหรับแสดงกระบวนการถอดรหัสช่องสัญญาณ
- กล่องข้อความ สำหรับแสดงผลลัพธ์ที่ได้ในแต่ละขั้นตอนของกระบวนการถอดรหัสช่องสัญญาณ

6.2) ส่วนการกำหนดค่า ประกอบด้วย

- กล่องตัวเลือก สำหรับเลือกช่องสัญญาณกายภาพ ผู้ใช้สามารถเลือกช่องสัญญาณกายภาพได้ 5 ช่องสัญญาณ PUSCH, PUCCH, PBCH, PDSCH และ PDCCH อย่างไรก็ตามช่องสัญญาณกายภาพที่เลือกต้องสอดคล้องกับช่องสัญญาณกายภาพที่เลือกในการเข้ารหัสช่องสัญญาณ
- กล่องตัวเลือก สำหรับเลือกอัลกอริทึมการถอดรหัสช่องสัญญาณ โดยในกรณี PUSCH และ PDSCH ผู้ใช้สามารถเลือกอัลกอริทึมการถอดรหัสได้ 4 อัลกอริทึม ได้แก่ Belief Propagation, Min-Sum, Normalized Min-Sum และ Offset Min-Sum และในกรณี PUCCH, PDCCH และ PBCH ผู้ใช้สามารถเลือกอัลกอริทึมการถอดรหัสได้ 3 อัลกอริทึม ได้แก่ Successive Cancellation, Successive Cancellation List และ Belief Propagation
- กล่องป้อนข้อความ สำหรับป้อนพารามิเตอร์ในการถอดรหัสช่องสัญญาณ
- ปุ่มวิทยุ สำหรับเลือกช่องทางการดีมอดูเลชันสัญญาณ ผู้ใช้สามารถเลือกช่องทางการดีมอดูเลชันสัญญาณได้ 2 ช่องทาง ได้แก่ การดีมอดูเลชันโดยใช้ซอฟต์แวร์ และการดีมอดูเลชันโดยใช้อุปกรณ์ FPGA

6.3) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Run” สำหรับดำเนินการถอดรหัสช่องสัญญาณ

- ปุ่ม “Clear” สำหรับล้างค่าผลลัพธ์ของการถอดรหัสช่องสัญญาณ

6.4) ส่วนแสดงกราฟ ประกอบด้วย

- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของผลลัพธ์ที่ได้จากการถอดรหัสช่องสัญญาณ

7) เอาต์พุต เป็นแสดงผลลัพธ์ของการถอดรหัสช่องสัญญาณ ซึ่งประกอบด้วยส่วนย่อย 3 ส่วน ดังนี้

7.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- กล่องข้อความ สำหรับแสดงผลลัพธ์

7.2) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Clear” สำหรับล้างค่าของผลลัพธ์

7.3) ส่วนแสดงกราฟ ประกอบด้วย

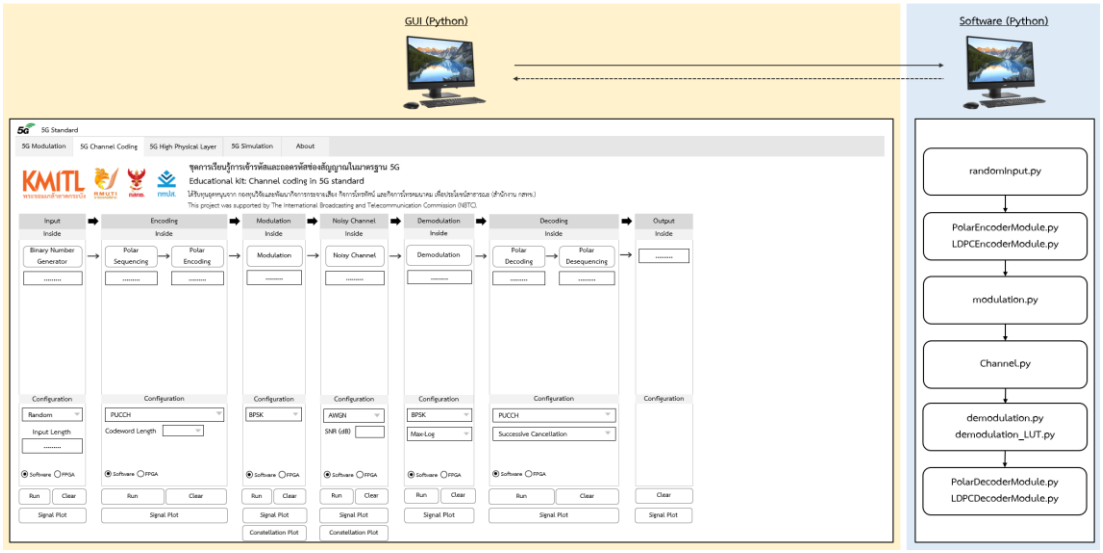
- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของผลลัพธ์

นอกจากนี้ ผู้ใช้สามารถเลือกการทำงานของแต่ละองค์ประกอบด้วยซอฟต์แวร์ หรืออุปกรณ์ FPGA ได้ โดยการเลือกปุ่มวิทยุที่อยู่ในส่วนการกำหนดค่า สำหรับการทำงานด้วยซอฟต์แวร์แต่ละองค์ประกอบจะเรียกใช้ Module การทำงานภาษา Python ที่สอดคล้องกัน ดังแสดงในรูปที่ 3.5 โดยแต่ละองค์ประกอบจะเรียกใช้ Module ดังตารางที่ 3.2

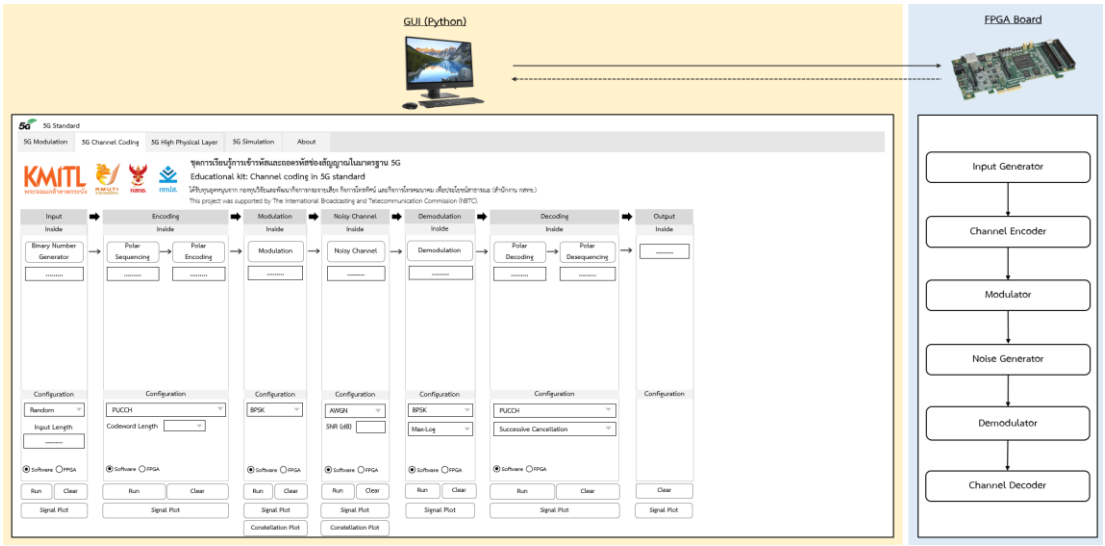
ตารางที่ 3.2 การเรียกใช้ซอฟต์แวร์ Module ของแต่ละองค์ประกอบของชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G

องค์ประกอบ	Module
อินพุต (Input)	randomInput.py
การเข้ารหัสช่องสัญญาณ (Encoding)	PolarEncoderModule.py LDPCEncoderModule.py
การมอดูเลชัน (Modulation)	modulation.py
ช่องสัญญาณ (Noisy Channel)	Channel.py
การถอดรหัสช่องสัญญาณ (Decoding)	PolarDecoderModule.py LDPCDecoderModule.py
การดีมอดูเลชัน (Demodulation)	deodulation.py demodulation_LUT.py

สำหรับการทำงานด้วยอุปกรณ์ FPGA แต่ละองค์ประกอบจะเรียกใช้ Module การทำงานบนอุปกรณ์ FPGA ที่สอดคล้องกัน โดยส่งข้อมูลการกำหนดค่าบน GUI ไปยังอุปกรณ์ FPGA และรับเอาต์พุตผ่าน Serial Port ดังแสดงในรูปที่ 3.6 ทั้งนี้ ผู้ใช้สามารถเลือกการทำงานด้วยซอฟต์แวร์ในบางองค์ประกอบและเลือกการทำงานด้วยอุปกรณ์ FPGA ในบางองค์ประกอบได้



รูปที่ 3.5 ชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G เชื่อมต่อกับซอฟต์แวร์

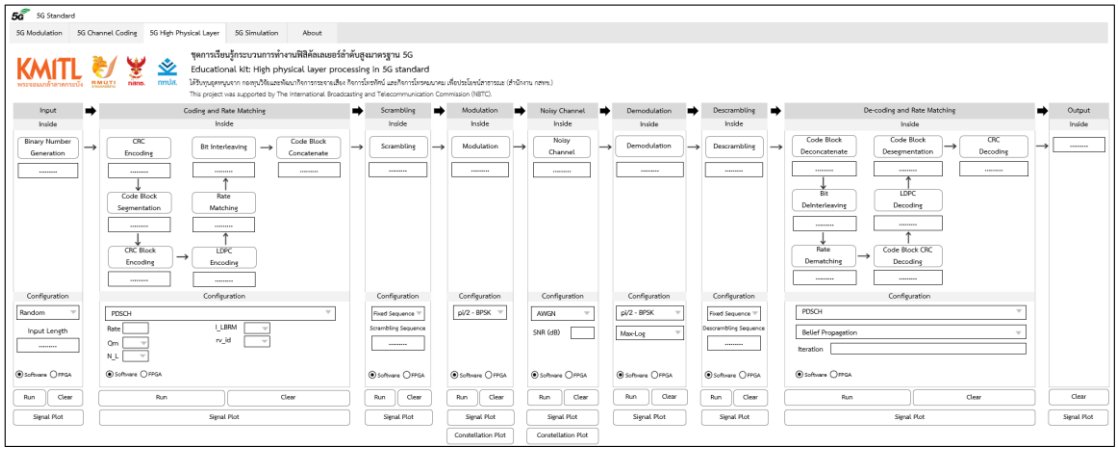


รูปที่ 3.6 ชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G
เชื่อมต่อกับอุปกรณ์ FPGA

3.1.3 การออกแบบชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G

โครงการนี้ออกแบบซอฟต์แวร์ GUI ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G โดยใช้ภาษา Python เพื่อให้ผู้ใช้งานสามารถเรียนรู้กระบวนทำงานของฟิสิคัลเยอร์ลำดับสูง (High Physical Layer Processing) ตามมาตรฐาน 5G ได้โดยง่าย ซึ่งผู้ใช้งานจะสามารถเห็นกระบวนการทำงานตั้งแต่การสร้างข้อมูลแบบสุ่มหรือนำเข้าข้อมูลอินพุต การเข้ารหัสและปรับอัตรารหัส การสแครม (Scrambling) การมอดูเลชัน การส่งผ่านช่องสัญญาณรบกวน การดีมอดูเลชัน

การดีสแครม (Descrambling) และการถอดรหัสและปรับอัตรารหัส นอกจากนี้ผู้ใช้งานสามารถดูผลลัพธ์ ณ จุดต่าง ๆ รวมถึงสามารถเรียกดูกราฟผลลัพธ์ของสัญญาณและแผนภาพคอนสเตลเลชันของการมอดูเลชันได้ โครงร่างการออกแบบชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G แสดงดังรูปที่ 3.7



รูปที่ 3.7 โครงร่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G

จากรูปที่ 3.7 ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G ประกอบด้วย 9 องค์ประกอบ ได้แก่ อินพุต (Input) การเข้ารหัสและปรับอัตรารหัส (Coding and Rate Matching) การสแครม (Scrambling) การมอดูเลชัน (Modulation) ช่องสัญญาณ (Noisy Channel) การดีมอดูเลชัน (Demodulation) การดีสแครม (Descrambling) การถอดรหัสและปรับอัตรารหัส (De-coding and Rate Matching) และเอาต์พุต (Output) โดยแต่ละองค์ประกอบมีรายละเอียดดังนี้

1) อินพุต เป็นส่วนสร้างและแสดงผลอินพุต ซึ่งประกอบด้วยส่วนย่อย 4 ส่วน ดังนี้

1.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- บล็อกไดอะแกรม สำหรับแสดงบล็อกอินพุต
- กล่องข้อความ สำหรับแสดงอินพุต

1.2) ส่วนการกำหนดค่า ประกอบด้วย

- กล่องตัวเลือก สำหรับเลือกวิธีการป้อนอินพุต ผู้ใช้สามารถเลือกวิธีการป้อนอินพุตได้ 3 วิธี ได้แก่ การสร้างอินพุตแบบสุ่ม การป้อนอินพุตด้วยตนเอง และการอัปโหลดรูปภาพ
- กล่องป้อนข้อความ สำหรับป้อนความยาวของอินพุตหรือป้อนอินพุต กรณีที่ผู้ใช้เลือกการสร้างอินพุตแบบสุ่มผู้ใช้สามารถป้อนความยาวของอินพุตตามต้องการได้ กรณีที่ผู้ใช้เลือกการป้อนอินพุตด้วยตัวเองผู้ใช้จะต้องป้อนอินพุตลงในกล่องข้อความ และกรณีที่ผู้ใช้เลือกการอัปโหลดรูปภาพ ระบบจะแปลงรูปภาพเป็นบิต

ข้อมูลแบบไบนารีและกำหนดความยาวตามบิตข้อมูลของรูปภาพในกล่องข้อความโดยอัตโนมัติ

- ปุ่มวิทยุ สำหรับเลือกช่องทางการสร้างอินพุต ผู้ใช้สามารถเลือกช่องทางการสร้างอินพุตได้ 2 ช่องทาง ได้แก่ การสร้างโดยใช้ซอฟต์แวร์ และการสร้างโดยใช้อุปกรณ์ FPGA

1.3) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Run” สำหรับดำเนินการสร้างอินพุต
- ปุ่ม “Clear” สำหรับล้างค่าอินพุต

1.4) ส่วนแสดงกราฟ ประกอบด้วย

- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของสัญญาณอินพุต

2) การเข้ารหัสและปรับอัตรารหัส เป็นส่วนการเข้ารหัสช่องและปรับอัตรารหัสตามมาตรฐาน 5G และแสดงผลลัพธ์ของแต่ละขั้นตอนในกระบวนการเข้ารหัสและปรับอัตรารหัส ซึ่งประกอบด้วยส่วนย่อย 4 ส่วน ดังนี้

2.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- บล็อกไดอะแกรม สำหรับแสดงกระบวนการเข้ารหัสและปรับอัตรารหัส
- กล่องข้อความ สำหรับแสดงผลลัพธ์ที่ได้ในแต่ละขั้นตอนของกระบวนการเข้ารหัสและปรับอัตรารหัส

2.2) ส่วนการกำหนดค่า ประกอบด้วย

- กล่องตัวเลือก สำหรับเลือกช่องสัญญาณกายภาพ ผู้ใช้สามารถเลือกช่องสัญญาณกายภาพได้ 5 ช่องสัญญาณ ได้แก่ PUSCH, PUCCH, PBCH, PDSCH และ PDCCH
- กล่องป้อนข้อความและกล่องตัวเลือก สำหรับเลือกพารามิเตอร์ในการเข้ารหัสและปรับอัตรารหัส สำหรับกรณี PUSCH และ PDSCH จะต้องป้อนหรือเลือกพารามิเตอร์ 5 ค่า ได้แก่ อัตรารหัส (Rate) จำนวนบิตต่อสัญลักษณ์ (Qm) จำนวนชั้นหรือพอร์ต (N_L) ข้อจำกัดดรรชนีของบัฟเฟอร์ (I_LBRM) และลำดับของส่วนซ้ำซ้อน (rv_id) สำหรับกรณี PUCCH, PDCCH และ PBCH จะต้องป้อนความยาวรหัสหลังการพังค์เจอร์ (Punctured Codeword Length) อย่างไรก็ตาม ในกรณี PBCH ความยาวหลังการพังค์เจอร์จะถูกกำหนดไว้ตายตัวที่ 864 บิต
- ปุ่มวิทยุ สำหรับเลือกช่องทางการเข้ารหัสและปรับอัตรารหัส ผู้ใช้สามารถเลือกช่องทางการเข้ารหัสเข้ารหัสและปรับอัตรารหัสได้ 2 ช่องทาง ได้แก่ การเข้ารหัสโดยใช้ซอฟต์แวร์ และการเข้ารหัสโดยใช้อุปกรณ์ FPGA

2.3) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Run” สำหรับดำเนินการเข้ารหัสและปรับอัตรารหัส
- ปุ่ม “Clear” สำหรับล้างค่าผลลัพธ์ของการเข้ารหัสและปรับอัตรารหัส

2.4) ส่วนแสดงกราฟ ประกอบด้วย

- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของผลลัพธ์ที่ได้จากการเข้ารหัสและปรับอัตรารหัส

3) การสแครม เป็นส่วนการรวมสัญญาณที่ได้จากการเข้ารหัสและปรับอัตรารหัสเข้ากับลำดับการสแครม (Scrambling Sequence) แบบมอดูโลสอง (Modulo-2) และแสดงผลลัพธ์ของการสแครม ซึ่งประกอบด้วยส่วนย่อย 4 ส่วน ดังนี้

3.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- บล็อกไดอะแกรม สำหรับแสดงบล็อกการสแครม
- กล่องข้อความ สำหรับแสดงผลลัพธ์ที่ได้จากการสแครม

3.2) ส่วนการกำหนดค่า ประกอบด้วย

- กล่องตัวเลือก สำหรับเลือกรูปแบบการสร้างลำดับการสแครม ผู้ใช้สามารถเลือกรูปแบบการสร้างลำดับการสแครมได้ 3 รูปแบบ ได้แก่ การใช้ลำดับแบบตายตัว (Fixed Sequence) การสร้างลำดับแบบสุ่มเทียม (Pseudo-Random Sequence) และการป้อนลำดับด้วยตนเอง (Fill-in) อย่างไรก็ตาม ลำดับที่สร้างขึ้นหรือผู้ใช้ป้อนจะมีความยาวเท่ากับเอาต์พุตที่ได้จากการเข้ารหัสและปรับอัตรารหัส
- ปุ่มวิทยุ สำหรับเลือกช่องทางการสแครม ผู้ใช้สามารถเลือกช่องทางการสแครมได้ 2 ช่องทาง ได้แก่ การการสแครมโดยใช้ซอฟต์แวร์ และการสแครมโดยใช้อุปกรณ์ FPGA

3.3) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Run” สำหรับดำเนินการสแครม
- ปุ่ม “Clear” สำหรับล้างค่าผลลัพธ์ของการสแครม

3.4) ส่วนแสดงกราฟ ประกอบด้วย

- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของผลลัพธ์ที่ได้จากการสแครม

4) การมอดูเลชัน เป็นส่วนการมอดูเลชันสัญญาณตามมาตรฐาน 5G และแสดงผลลัพธ์ของการมอดูเลชัน ซึ่งประกอบด้วยส่วนย่อย 4 ส่วน ดังนี้

4.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- บล็อกไดอะแกรม สำหรับแสดงบล็อกการมอดูเลชันสัญญาณ
- กล่องข้อความ สำหรับแสดงผลลัพธ์ที่ได้จากการมอดูเลชันสัญญาณ

4.2) ส่วนการกำหนดค่า ประกอบด้วย

- กล่องตัวเลือก สำหรับเลือกรูปแบบการมอดูเลชันสัญญาณมาตรฐาน 5G ผู้ใช้สามารถเลือกรูปแบบการมอดูเลชันได้ 7 รูปแบบ ได้แก่ BPSK, π/2-BPSK, QPSK, 16QAM, 64QAM, 256QAM และ 1024QAM อย่างไรก็ตาม ตัวเลือกของ

รูปแบบการมอดูเลชันจะเปลี่ยนแปลงตามช่องสัญญาณกายภาพที่เลือกในส่วนการกำหนดค่าของการเข้ารหัสช่องและปรับอัตรารหัส

- ปุ่มวิทยุ สำหรับเลือกช่องทางการมอดูเลชันสัญญาณ ผู้ใช้สามารถเลือกช่องทางการมอดูเลชันสัญญาณได้ 2 ช่องทาง ได้แก่ การมอดูเลชันโดยใช้ซอฟต์แวร์ และการมอดูเลชันโดยใช้อุปกรณ์ FPGA

4.3) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Run” สำหรับดำเนินการมอดูเลชันสัญญาณ
- ปุ่ม “Clear” สำหรับล้างค่าผลลัพธ์ของการมอดูเลชันสัญญาณ

4.4) ส่วนแสดงกราฟ ประกอบด้วย

- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของสัญญาณมอดูเลชัน
- ปุ่ม “Constellation Plot” สำหรับแสดงแผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชัน

5) ช่องสัญญาณ เป็นส่วนสร้างช่องสัญญาณรบกวนและแสดงผลลัพธ์ของสัญญาณมอดูเลชันที่ผ่านช่องสัญญาณรบกวน ซึ่งประกอบด้วยส่วนย่อย 4 ส่วน ดังนี้

5.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- บล็อกไดอะแกรม สำหรับแสดงบล็อกช่องสัญญาณรบกวน
- กล่องข้อความ สำหรับแสดงผลลัพธ์สัญญาณมอดูเลชันที่ผ่านช่องสัญญาณรบกวน

5.2) ส่วนการกำหนดค่า ประกอบด้วย

- กล่องตัวเลือก สำหรับเลือกรูปแบบสัญญาณรบกวน ผู้ใช้สามารถเลือกรูปแบบของช่องสัญญาณรบกวนได้ 2 รูปแบบ ได้แก่ AWGN Channel และ Fading Channel
- กล่องป้อนข้อความ สำหรับป้อนค่าอัตราส่วนของสัญญาณต่อสัญญาณรบกวน

5.3) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Run” สำหรับดำเนินการสร้างสัญญาณรบกวนและรวมสัญญาณมอดูเลชันเข้ากับสัญญาณรบกวน
- ปุ่ม “Clear” สำหรับล้างค่าผลลัพธ์สัญญาณมอดูเลชันที่ผ่านช่องสัญญาณรบกวน
- ปุ่มวิทยุ สำหรับเลือกช่องทางการสร้างสัญญาณรบกวนและรวมสัญญาณมอดูเลชันเข้ากับสัญญาณรบกวน ผู้ใช้สามารถเลือกช่องทางการสร้างสัญญาณรบกวนและรวมสัญญาณมอดูเลชันได้ 2 ช่องทาง ได้แก่ การสร้างโดยใช้ซอฟต์แวร์ และการสร้างโดยใช้อุปกรณ์ FPGA

5.4) ส่วนแสดงกราฟ ประกอบด้วย

- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของสัญญาณมอดูเลชันที่ผ่านช่องสัญญาณรบกวน
- ปุ่ม “Constellation Plot” สำหรับแสดงแผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันที่ผ่านช่องสัญญาณรบกวน

6) การดีมอดูเลชัน เป็นส่วนการดีมอดูเลชันสัญญาณตามมาตรฐาน 5G และแสดงผลลัพธ์ของการดีมอดูเลชันสัญญาณ ซึ่งประกอบด้วยส่วนย่อย 4 ส่วน ดังนี้

6.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- บล็อกไดอะแกรม สำหรับแสดงบล็อกการดีมอดูเลชันสัญญาณ
- กล่องข้อความ สำหรับแสดงผลลัพธ์ที่ได้จากการดีมอดูเลชันสัญญาณ

6.2) ส่วนการกำหนดค่า ประกอบด้วย

- กล่องตัวเลือก สำหรับเลือกรูปแบบการดีมอดูเลชันสัญญาณมาตรฐาน 5G ผู้ใช้สามารถเลือกรูปแบบการดีมอดูเลชันได้ 7 รูปแบบ ได้แก่ BPSK, π/2-BPSK, QPSK, 16QAM, 64QAM, 256QAM และ 1024QAM อย่างไรก็ตาม ตัวเลือกของรูปแบบการดีมอดูเลชันจะเปลี่ยนแปลงตามช่องสัญญาณกายภาพที่เลือกในส่วนการกำหนดค่าของการเข้ารหัสและปรับอัตรารหัส และสอดคล้องกับการมอดูเลชัน
- กล่องตัวเลือก สำหรับเลือกอัลกอริทึมการดีมอดูเลชันสัญญาณ ผู้ใช้สามารถเลือกอัลกอริทึมการดีมอดูเลชันได้ 2 อัลกอริทึม ได้แก่ Mag-Log และ LUT
- ปุ่มวิทยุ สำหรับเลือกช่องทางการดีมอดูเลชันสัญญาณ ผู้ใช้สามารถเลือกช่องทางการดีมอดูเลชันสัญญาณได้ 2 ช่องทาง ได้แก่ การดีมอดูเลชันโดยใช้ซอฟต์แวร์ และการดีมอดูเลชันโดยใช้อุปกรณ์ FPGA

6.3) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Run” สำหรับดำเนินการดีมอดูเลชันสัญญาณ
- ปุ่ม “Clear” สำหรับล้างค่าผลลัพธ์ของการดีมอดูเลชันสัญญาณ

6.4) ส่วนแสดงกราฟ ประกอบด้วย

- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของผลลัพธ์ที่ได้จากการดีมอดูเลชันสัญญาณ

7) การดีสแครม เป็นส่วนการรวมสัญญาณที่ได้จากการดีมอดูเลชันเข้ากับลำดับการดีสแครม (Descrambling Sequence) แบบมอดูโลสอง (Modulo-2) และแสดงผลลัพธ์ของการสแครม ซึ่งประกอบด้วยส่วนย่อย 4 ส่วน ดังนี้

7.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- บล็อกไดอะแกรม สำหรับแสดงบล็อกการดีสแครม
- กล่องข้อความ สำหรับแสดงผลลัพธ์ที่ได้จากการดีสแครม

7.2) ส่วนการกำหนดค่า ประกอบด้วย

- กล่องตัวเลือก สำหรับเลือกรูปแบบการสร้างลำดับการดีสแครม ผู้ใช้สามารถเลือกรูปแบบการสร้างลำดับการสแครมได้ 3 รูปแบบ ได้แก่ การใช้ลำดับแบบตายตัว การสร้างลำดับแบบสุ่มเทียม และการป้อนลำดับด้วยตนเอง อย่างไรก็ตามลำดับการดีสแครมต้องตรงกับลำดับการสแครม
- ปุ่มวิทยุ สำหรับเลือกช่องทางการดีสแครม ผู้ใช้สามารถเลือกช่องทางการดีสแครมได้ 2 ช่องทาง ได้แก่ การการดีสแครมโดยใช้ซอฟต์แวร์ และการดีสแครมโดยใช้อุปกรณ์ FPGA

7.3) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Run” สำหรับดำเนินการดีสแครม
- ปุ่ม “Clear” สำหรับล้างค่าผลลัพธ์ของการดีสแครม

7.4) ส่วนแสดงกราฟ ประกอบด้วย

- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของผลลัพธ์ที่ได้จากการดีสแครม

8) การถอดรหัสและปรับอัตรารหัส เป็นส่วนการถอดรหัสและปรับอัตรารหัสตามมาตรฐาน 5G และแสดงผลลัพธ์ของแต่ละขั้นตอนในกระบวนการถอดรหัสและปรับอัตรารหัส ซึ่งประกอบด้วยส่วนย่อย 4 ส่วน ดังนี้

8.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- บล็อกไดอะแกรม สำหรับแสดงกระบวนการถอดรหัสและปรับอัตรารหัส
- กล่องข้อความ สำหรับแสดงผลลัพธ์ที่ได้ในแต่ละขั้นตอนของกระบวนการถอดรหัสและปรับอัตรารหัส

8.2) ส่วนการกำหนดค่า ประกอบด้วย

- กล่องตัวเลือก สำหรับเลือกช่องสัญญาณกายภาพ ผู้ใช้สามารถเลือกช่องสัญญาณกายภาพได้ 5 ช่องสัญญาณ PUSCH, PUCCH, PBCH, PDSCH และ PDCCH อย่างไรก็ตามช่องสัญญาณกายภาพที่เลือกต้องสอดคล้องกับช่องสัญญาณกายภาพที่เลือกในการเข้ารหัสและปรับอัตรารหัส
- กล่องตัวเลือก สำหรับเลือกอัลกอริทึมการถอดรหัสและปรับอัตรารหัส โดยในกรณี PUSCH และ PDSCH ผู้ใช้สามารถเลือกอัลกอริทึมการถอดรหัสได้ 4 อัลกอริทึม ได้แก่ Belief Propagation, Min-Sum, Normalized Min-Sum และ Offset Min-Sum และในกรณี PUCCH, PDCCH และ PBCH ผู้ใช้สามารถเลือกอัลกอริทึมการถอดรหัสได้ 3 อัลกอริทึม ได้แก่ Successive Cancellation, Successive Cancellation List และ Belief Propagation
- กล่องป้อนข้อความและกล่องเพิ่มจำนวน สำหรับป้อนพารามิเตอร์ในการถอดรหัสและปรับอัตรารหัส

- ปุ่มวิทยุ สำหรับเลือกช่องทางการดีมอดูเลชันสัญญาณ ผู้ใช้สามารถเลือกช่องทางการดีมอดูเลชันสัญญาณได้ 2 ช่องทาง ได้แก่ การดีมอดูเลชันโดยใช้ซอฟต์แวร์ และการดีมอดูเลชันโดยใช้อุปกรณ์ FPGA

8.3) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Run” สำหรับดำเนินการถอดรหัสและปรับอัตรารหัส
- ปุ่ม “Clear” สำหรับล้างค่าผลลัพธ์ของการถอดรหัสและปรับอัตรารหัส

8.4) ส่วนแสดงกราฟ ประกอบด้วย

- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของผลลัพธ์ที่ได้จากการถอดรหัสและปรับอัตรารหัส

9) เอาต์พุต เป็นแสดงผลลัพธ์ของการถอดรหัสและปรับอัตรารหัส ซึ่งประกอบด้วยส่วนย่อย 3 ส่วน ดังนี้

9.1) ส่วนบล็อกแสดงผลภายใน ประกอบด้วย

- กล่องข้อความ สำหรับแสดงผลลัพธ์

9.2) ส่วนควบคุม ประกอบด้วย

- ปุ่ม “Clear” สำหรับล้างค่าของผลลัพธ์

9.3) ส่วนแสดงกราฟ ประกอบด้วย

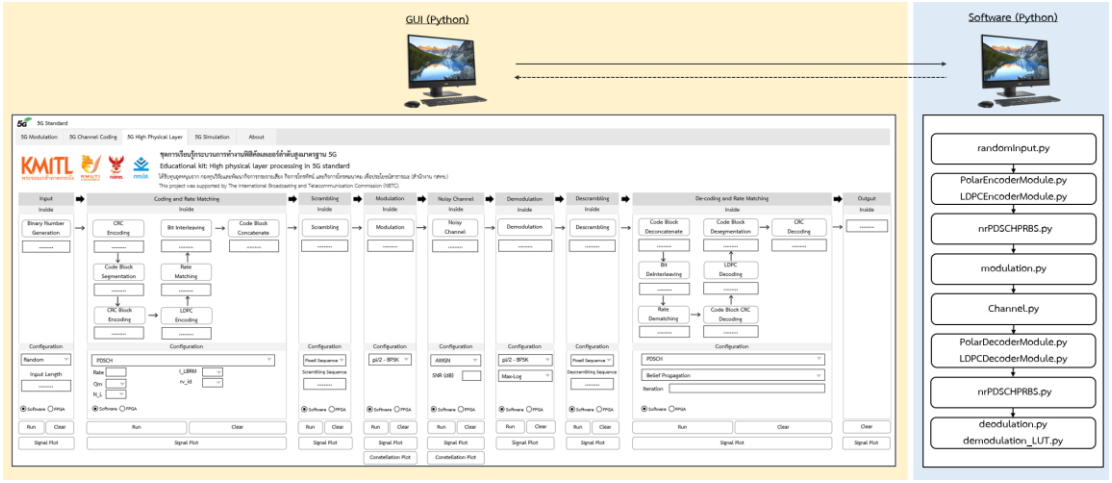
- ปุ่ม “Signal Plot” สำหรับแสดงกราฟของผลลัพธ์

นอกจากนี้ ผู้ใช้สามารถเลือกการทำงานของแต่ละองค์ประกอบด้วยซอฟต์แวร์ หรืออุปกรณ์ FPGA ได้โดยการเลือกปุ่มวิทยุที่อยู่ในส่วนการกำหนดค่า สำหรับการทำงานด้วยซอฟต์แวร์แต่ละองค์ประกอบจะเรียกใช้ Module การทำงานภาษา Python ที่สอดคล้องกัน ดังแสดงในรูปที่ 3.8 โดยแต่ละองค์ประกอบจะเรียกใช้ Module ดังตารางที่ 3.3

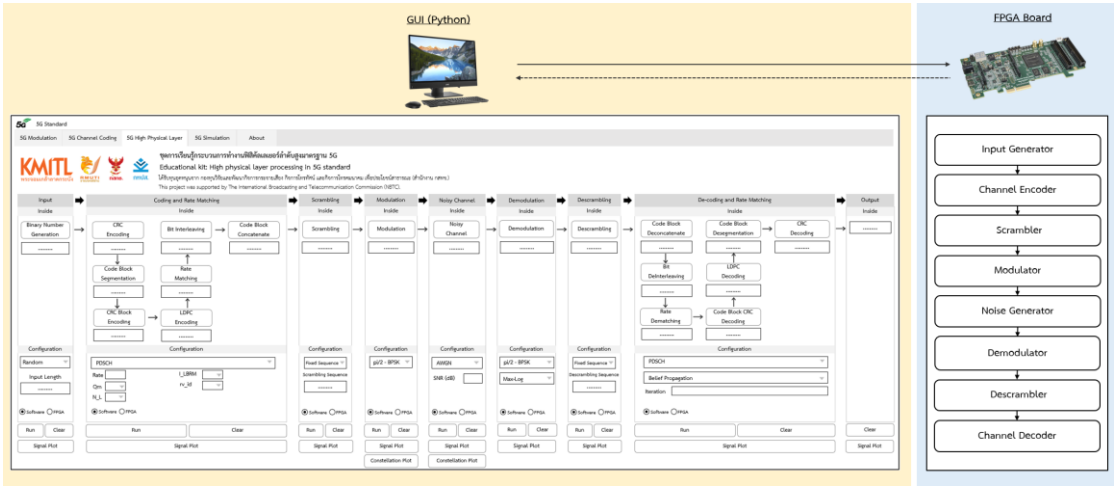
ตารางที่ 3.3 การเรียกใช้ซอฟต์แวร์ Module ของแต่ละองค์ประกอบของชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G

องค์ประกอบ	Module
อินพุต (Input)	randomInput.py
การเข้ารหัสและปรับอัตรารหัส (Coding and Rate Matching)	PolarEncoderModule.py LDPCEncoderModule.py
การสแครม (Scrambling)	nrPDSCHPRBS.py
การมอดูเลชัน (Modulation)	modulation.py
ช่องสัญญาณ (Noisy Channel)	Channel.py
การถอดรหัสและปรับอัตรารหัส (De-coding and Rate Matching)	PolarDecoderModule.py LDPCDecoderModule.py
การดีสแครม (Descrambling)	nrPDSCHPRBS.py
การดีมอดูเลชัน (Demodulation)	deodulation.py demodulation_LUT.py

สำหรับการทำงานด้วยอุปกรณ์ FPGA แต่ละองค์ประกอบจะเรียกใช้ Module การทำงานบนอุปกรณ์ FPGA ที่สอดคล้องกัน โดยส่งข้อมูลการกำหนดค่าบน GUI ไปยังอุปกรณ์ FPGA และรับเอาต์พุตผ่าน Serial Port ดังแสดงในรูปที่ 3.9 ทั้งนี้ ผู้ใช้สามารถเลือกการทำงานด้วยซอฟต์แวร์ในบางองค์ประกอบและเลือกการทำงานด้วยอุปกรณ์ FPGA ในบางองค์ประกอบได้



รูปที่ 3.8 ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G เชื่อมต่อกับซอฟต์แวร์



รูปที่ 3.9 ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G เชื่อมต่อกับอุปกรณ์ FPGA

3.2 การออกแบบชุดทดสอบสมรรถนะ

3.2.1 การออกแบบชุดทดสอบสมรรถนะการมอดูเลชันและดีมอดูเลชันในระบบสื่อสารไร้สาย

โครงการนี้ออกแบบชุดทดสอบสมรรถนะการมอดูเลชันและดีมอดูเลชันในระบบสื่อสารไร้สาย โดยผู้ใช้งานสามารถเลือกช่องสัญญาณสื่อสารแบบ AWGN Channel หรือ Fading Channel เพื่อทดลองหาสมรรถนะอัตราบิตผิดพลาด ณ จุดต่าง ๆ ของระบบได้ ซึ่งการออกแบบชุดทดสอบสมรรถนะการมอดูเลชันและดีมอดูเลชันในระบบสื่อสารไร้สายประกอบด้วย 3 ส่วน ได้แก่ ส่วนแสดงพารามิเตอร์ของการมอดูเลชันและดีมอดูเลชัน ส่วนกำหนดค่าการทดสอบสมรรถนะ และส่วนแสดงผลสมรรถนะ ดังแสดงในรูปที่ 3.10 โดยแต่ละส่วนมีรายละเอียดดังนี้

1) ส่วนแสดงพารามิเตอร์ของการมอดูเลชันและดีมอดูเลชัน เป็นส่วนแสดงพารามิเตอร์ที่กำหนดไว้ในหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G

2) ส่วนกำหนดค่าการทดสอบสมรรถนะ เป็นส่วนควบคุมและกำหนดค่าพารามิเตอร์สำหรับการทดสอบสมรรถนะ ซึ่งประกอบด้วย 2 ส่วนย่อย ดังนี้

2.1) ส่วนการกำหนดค่า ประกอบด้วย

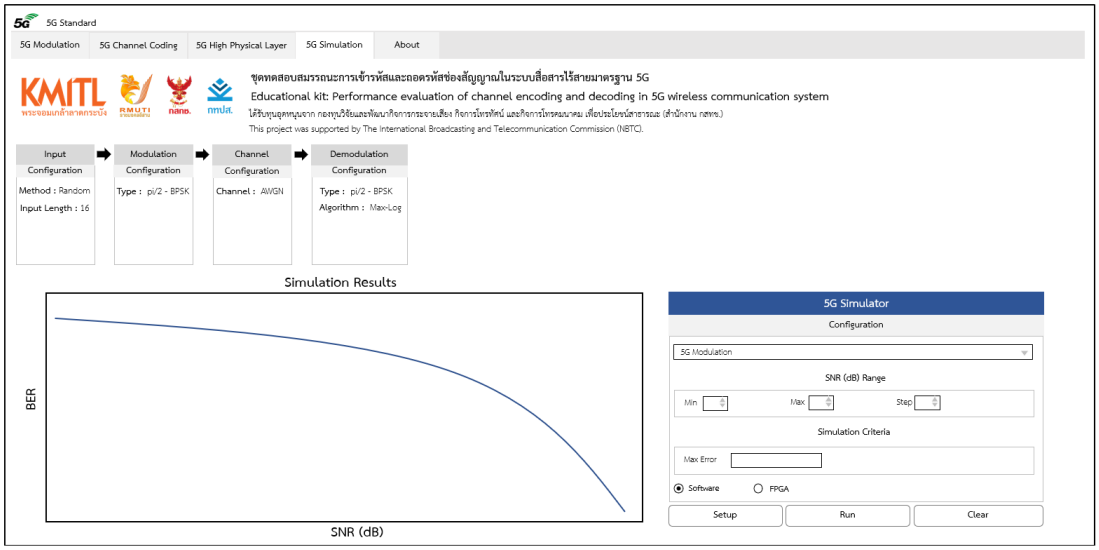
- กล่องตัวเลือก สำหรับเลือกหน้าต่างที่ต้องการดึงพารามิเตอร์มาใช้ในการทดสอบสมรรถนะ ผู้ใช้สามารถเลือกพารามิเตอร์ได้จาก 3 หน้าต่าง ได้แก่ หน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G (5G Modulation) หน้าต่างชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G (5G Channel coding) และหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G (5G High Physical Layer)
- กล่องเพิ่มจำนวน (Spin Box) สำหรับกำหนดค่าต่ำสุด (Min) ค่าสูงสุด (Max) และขั้นการเพิ่มค่า (Step) ของ SNR
- กล่องป้อนข้อความ สำหรับป้อนค่าจำนวนบล็อกผิดพลาดสะสมสูงสุด (Max Error)
- ปุ่มวิทยุ สำหรับเลือกช่องทางการทดสอบสมรรถนะ ผู้ใช้สามารถเลือกช่องทางการทดสอบสมรรถนะได้ 2 ช่องทาง ได้แก่ ทดสอบสมรรถนะโดยใช้ซอฟต์แวร์ และทดสอบสมรรถนะโดยใช้อุปกรณ์ FPGA

2.2) ส่วนควบคุม ประกอบด้วย

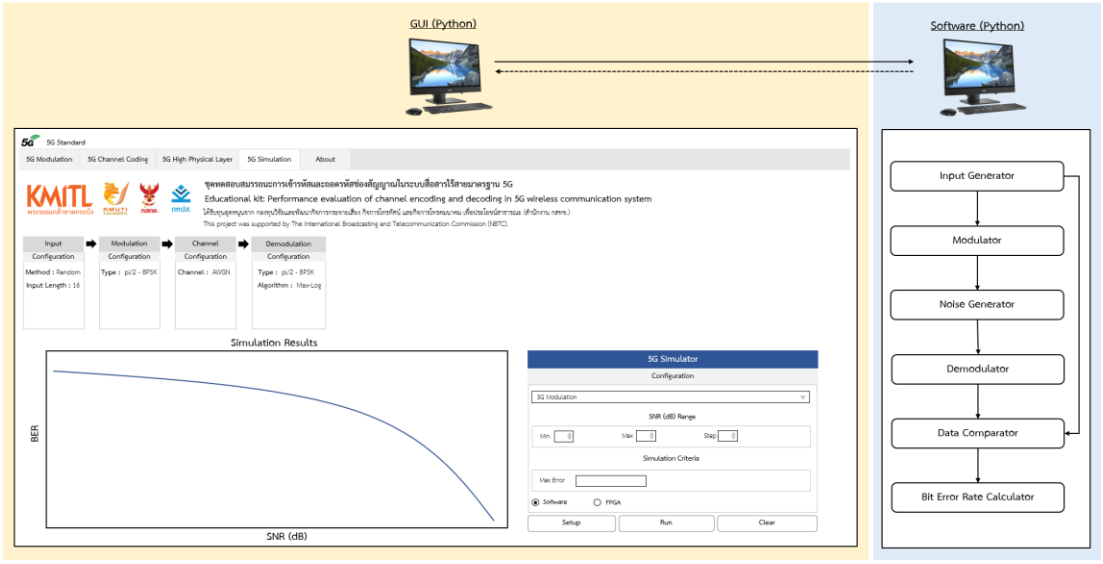
- ปุ่ม “Setup” สำหรับเรียกค่าพารามิเตอร์การมอดูเลชันและดีมอดูเลชันจากหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G
- ปุ่ม “Run” สำหรับดำเนินทดสอบสมรรถนะ
- ปุ่ม “Clear” สำหรับล้างค่าพารามิเตอร์และผลการทดสอบสมรรถนะ

3) ส่วนแสดงผลสมรรถนะ เป็นส่วนแสดงผลการทดสอบสมรรถนะในรูปแบบกราฟของอัตราบิตผิดพลาด นอกจากนี้ ผู้ใช้สามารถเลือกทดสอบสมรรถนะการมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G ด้วยซอฟต์แวร์ หรืออุปกรณ์ FPGA ได้โดยการเลือกปุ่มวิทยุที่อยู่ในส่วนการกำหนดค่า สำหรับการทำงานด้วยซอฟต์แวร์แต่ละองค์ประกอบจะเรียกใช้ Module การทำงานภาษา Python ภายในคอมพิวเตอร์ ซึ่งประกอบด้วย Module ตัวสร้างอินพุต Module ตัวมอดูเลชัน Module ตัวสร้าง

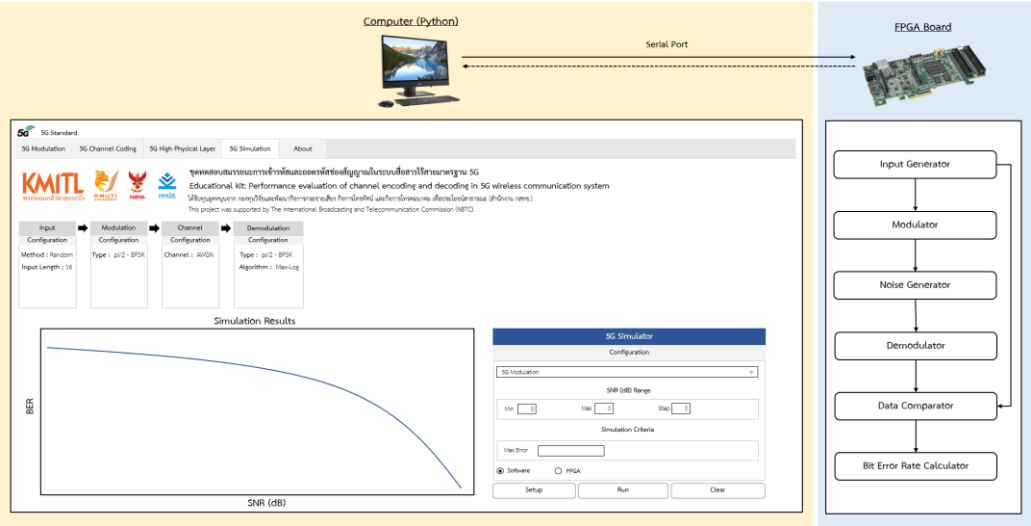
สัญญาณรบกวน Module ตัวดีมอดูเลชัน Module ตัวเปรียบเทียบข้อมูล (Data Comparator) และ Module ตัวคำนวณอัตราบิตผิดพลาด (Bit Error Rate Calculator) ดังแสดงในรูปที่ 3.11 สำหรับการทำงานด้วยอุปกรณ์ FPGA หน้าต่าง GUI จะเรียกใช้ Module การทำงานบนอุปกรณ์ FPGA โดยส่งข้อมูลการกำหนดค่าและรับค่าเอาต์พุตผ่าน Serial Port ดังแสดงในรูปที่ 3.12



รูปที่ 3.10 โครงร่างชุดทดสอบสมรรถนะการมอดูเลชันและดีมอดูเลชันในระบบสื่อสารไร้สาย



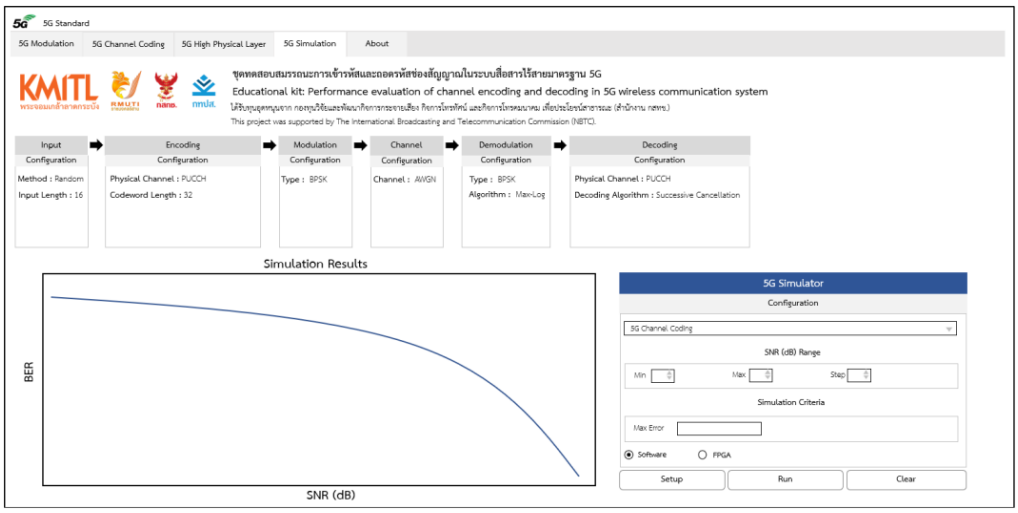
รูปที่ 3.11 ชุดทดสอบสมรรถนะการมอดูเลชันและดีมอดูเลชันในระบบสื่อสารไร้สาย เชื่อมต่อกับซอฟต์แวร์



รูปที่ 3.12 ชุดทดสอบสมรรถนะการเข้ารหัสและถอดรหัสช่องสัญญาณในระบบสื่อสารไร้สาย เชื่อมต่อกับอุปกรณ์ FPGA

3.2.2 การออกแบบชุดทดสอบสมรรถนะการเข้ารหัสและถอดรหัสช่องสัญญาณในระบบสื่อสารไร้สาย

โครงการนี้ออกแบบชุดทดสอบสมรรถนะการเข้ารหัสและถอดรหัสช่องสัญญาณในระบบสื่อสารไร้สาย โดยผู้ใช้งานสามารถเลือกช่องสัญญาณสื่อสารแบบ AWGN Channel หรือ Fading Channel เพื่อทดลองหาสมรรถนะอัตราบิตผิดพลาด ณ จุดต่าง ๆ ของระบบได้ ซึ่งการออกแบบชุดทดสอบสมรรถนะการเข้ารหัสและถอดรหัสช่องสัญญาณในระบบสื่อสารไร้สาย ประกอบด้วย 3 ส่วน ได้แก่ ส่วนแสดงพารามิเตอร์ของการเข้ารหัสและถอดรหัส ส่วนกำหนดค่าการทดสอบสมรรถนะ และส่วนแสดงผลสมรรถนะ ดังแสดงในรูปที่ 3.13 โดยแต่ละส่วนมีรายละเอียดดังนี้



รูปที่ 3.13 โครงร่างชุดทดสอบสมรรถนะการเข้ารหัสและถอดรหัสในระบบสื่อสารไร้สาย

1) ส่วนแสดงพารามิเตอร์ของการเข้ารหัสและถอดรหัส เป็นส่วนแสดงพารามิเตอร์ที่กำหนดไว้ในหน้าต่างชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G

2) ส่วนกำหนดค่าการทดสอบสมรรถนะ เป็นส่วนควบคุมและกำหนดค่าพารามิเตอร์สำหรับการทดสอบสมรรถนะ ซึ่งประกอบด้วย 2 ส่วนย่อย ดังนี้

2.1) ส่วนการกำหนดค่า ประกอบด้วย

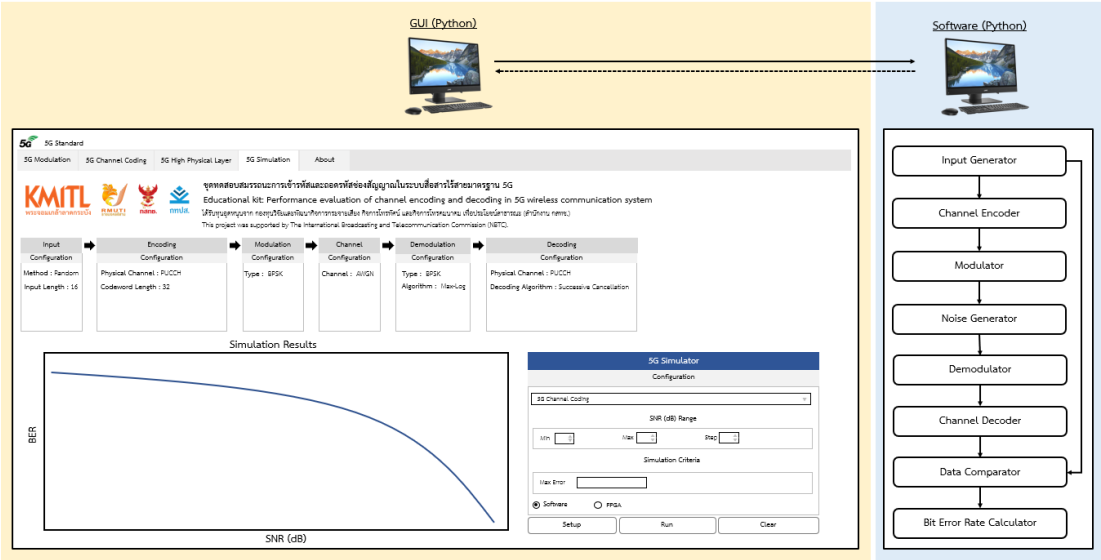
- กล่องตัวเลือก สำหรับเลือกหน้าต่างที่ต้องการดึงพารามิเตอร์มาใช้ในการทดสอบสมรรถนะ ผู้ใช้สามารถเลือกพารามิเตอร์ได้จาก 3 หน้าต่าง ได้แก่ หน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G หน้าต่างชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G และหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G
- กล่องเพิ่มจำนวน สำหรับกำหนดค่าต่ำสุด ค่าสูงสุด และขั้นการเพิ่มค่าของ SNR
- กล่องป้อนข้อความ สำหรับป้อนค่าจำนวนบิตผิดพลาดสะสมสูงสุด
- ปุ่มวิทยุ สำหรับเลือกช่องทางการทดสอบสมรรถนะ ผู้ใช้สามารถเลือกช่องทางการทดสอบสมรรถนะได้ 2 ช่องทาง ได้แก่ ทดสอบสมรรถนะโดยใช้ซอฟต์แวร์ และทดสอบสมรรถนะโดยใช้อุปกรณ์ FPGA

2.2) ส่วนควบคุม ประกอบด้วย

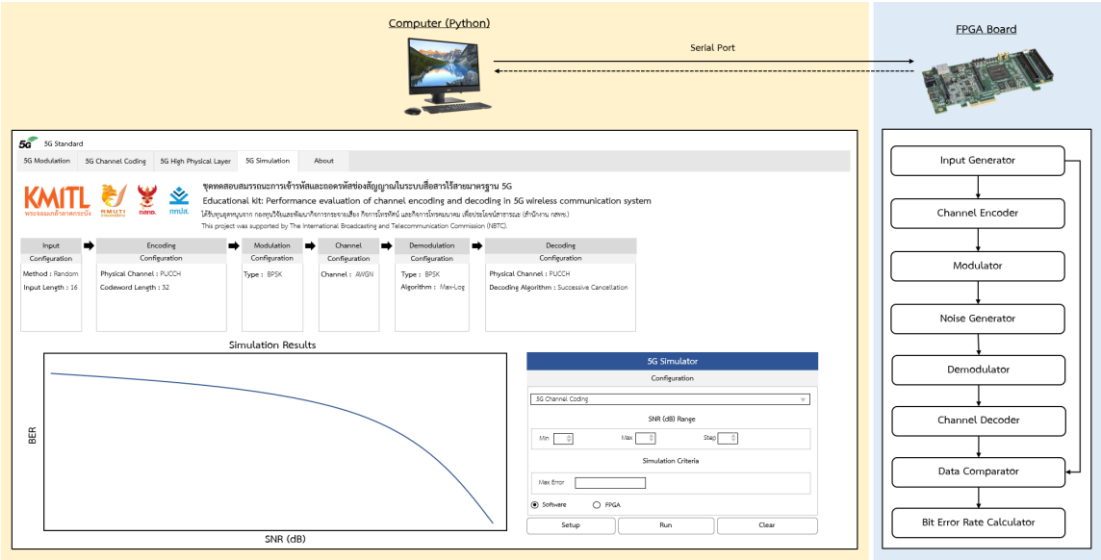
- ปุ่ม “Setup” สำหรับเรียกค่าพารามิเตอร์การเข้ารหัสและถอดรหัสจากหน้าต่างชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G
- ปุ่ม “Run” สำหรับดำเนินทดสอบสมรรถนะ
- ปุ่ม “Clear” สำหรับล้างค่าพารามิเตอร์และผลการทดสอบสมรรถนะ

3) ส่วนแสดงผลสมรรถนะ เป็นส่วนแสดงผลการทดสอบสมรรถนะในรูปแบบกราฟของอัตราบิตผิดพลาด

นอกจากนี้ ผู้ใช้สามารถเลือกทดสอบสมรรถนะการเข้ารหัสและถอดรหัสช่องสัญญาณในระบบสื่อสารไร้สายด้วยซอฟต์แวร์ หรืออุปกรณ์ FPGA ได้โดยการเลือกปุ่มวิทยุที่อยู่ในส่วนการกำหนดค่า สำหรับการทำงานด้วยซอฟต์แวร์แต่ละองค์ประกอบจะเรียกใช้ Module การทำงานภาษา Python ภายในคอมพิวเตอร์ ซึ่งประกอบด้วย Module ตัวสร้างอินพุต Module ตัวเข้ารหัสช่องสัญญาณ Module ตัวมอดูเลชัน Module ตัวสร้างสัญญาณรบกวน Module ตัวดีมอดูเลชัน Module ตัวถอดรหัส Module ตัวเปรียบเทียบข้อมูล และ Module ตัวคำนวณอัตราบิตผิดพลาด ดังแสดงในรูปที่ 2 สำหรับการทำงานด้วยอุปกรณ์ FPGA หน้าต่าง GUI จะเรียกใช้ Module การทำงานบนอุปกรณ์ FPGA โดยส่งข้อมูลการกำหนดค่าและรับค่าเอาต์พุตผ่าน Serial Port ดังแสดงในรูปที่ 3.14



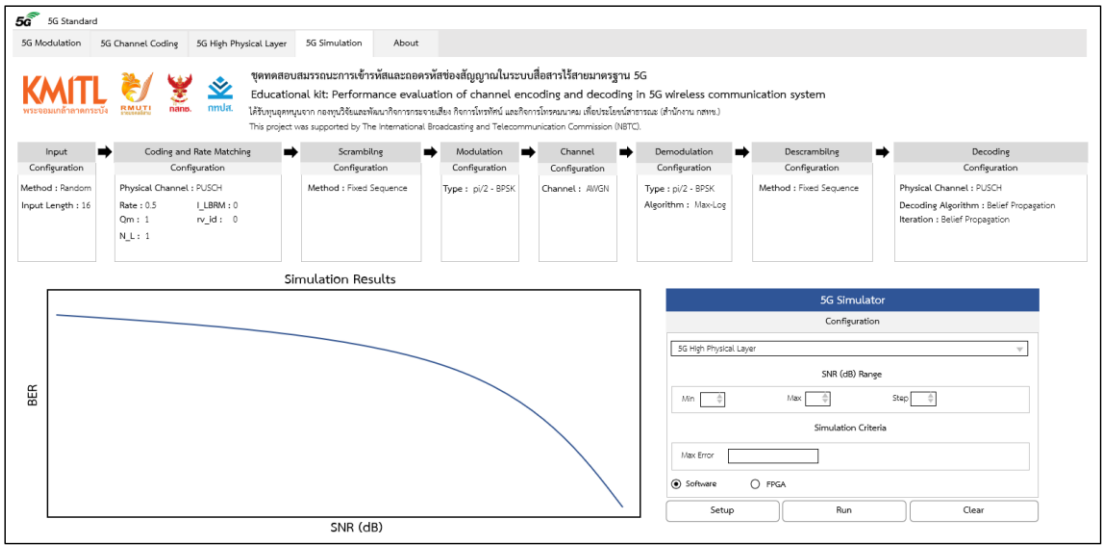
รูปที่ 3.14 ชุดทดสอบสมรรถนะการเข้ารหัสและถอดรหัสช่องสัญญาณในระบบสื่อสารไร้สายเชื่อมต่อกับซอฟต์แวร์



รูปที่ 3.15 ชุดทดสอบสมรรถนะการเข้ารหัสและถอดรหัสช่องสัญญาณในระบบสื่อสารไร้สายเชื่อมต่อกับอุปกรณ์ FPGA

3.2.3 การออกแบบชุดทดสอบสมรรถนะกระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงในระบบสื่อสารไร้สาย

โครงการนี้ออกแบบชุดทดสอบสมรรถนะกระบวนการทำงานชั้นฟิสิคัลเลเยอร์ลำดับสูงในระบบสื่อสารไร้สาย โดยผู้ใช้งานสามารถเลือกช่องสัญญาณสื่อสารแบบ AWGN Channel หรือ Fading Channel เพื่อทดลองหาสมรรถนะอัตราบิตผิดพลาด ณ จุดต่าง ๆ ของระบบได้ ซึ่งการออกแบบชุดทดสอบสมรรถนะกระบวนการทำงานชั้นฟิสิคัลเลเยอร์ลำดับสูงในระบบสื่อสารไร้สายประกอบด้วย 3 ส่วน ได้แก่ ส่วนแสดงพารามิเตอร์ของกระบวนการทำงานชั้นฟิสิคัลเลเยอร์ลำดับสูง ส่วนกำหนดค่าการทดสอบสมรรถนะ และส่วนแสดงผลสมรรถนะ ดังแสดงในรูปที่ 3.16 โดยแต่ละส่วนมีรายละเอียดดังนี้



รูปที่ 3.16 โครงร่างกระบวนการทำงานชั้นฟิสิคัลเลเยอร์ลำดับสูงในระบบสื่อสารไร้สาย

1) ส่วนแสดงพารามิเตอร์ของกระบวนการทำงานชั้นฟิสิคัลเลเยอร์ลำดับสูง เป็นส่วนแสดงพารามิเตอร์ที่กำหนดไว้ในหน้าต่างชุดการเรียนรู้กระบวนการทำงานชั้นฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G

2) ส่วนกำหนดค่าการทดสอบสมรรถนะ เป็นส่วนควบคุมและกำหนดค่าพารามิเตอร์สำหรับการทดสอบสมรรถนะ ซึ่งประกอบด้วย 2 ส่วนย่อย ดังนี้

2.1) ส่วนการกำหนดค่า ประกอบด้วย

- กล่องตัวเลือก สำหรับเลือกหน้าต่างที่ต้องการดึงพารามิเตอร์มาใช้ในการทดสอบสมรรถนะ ผู้ใช้สามารถเลือกพารามิเตอร์ได้จาก 3 หน้าต่าง ได้แก่ หน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G หน้าต่างชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G และหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G
- กล่องเพิ่มจำนวน สำหรับกำหนดค่าต่ำสุด ค่าสูงสุด และขั้นการเพิ่มค่าของ SNR
- กล่องป้อนข้อความ สำหรับป้อนค่าจำนวนบิตผิดพลาดสะสมสูงสุด

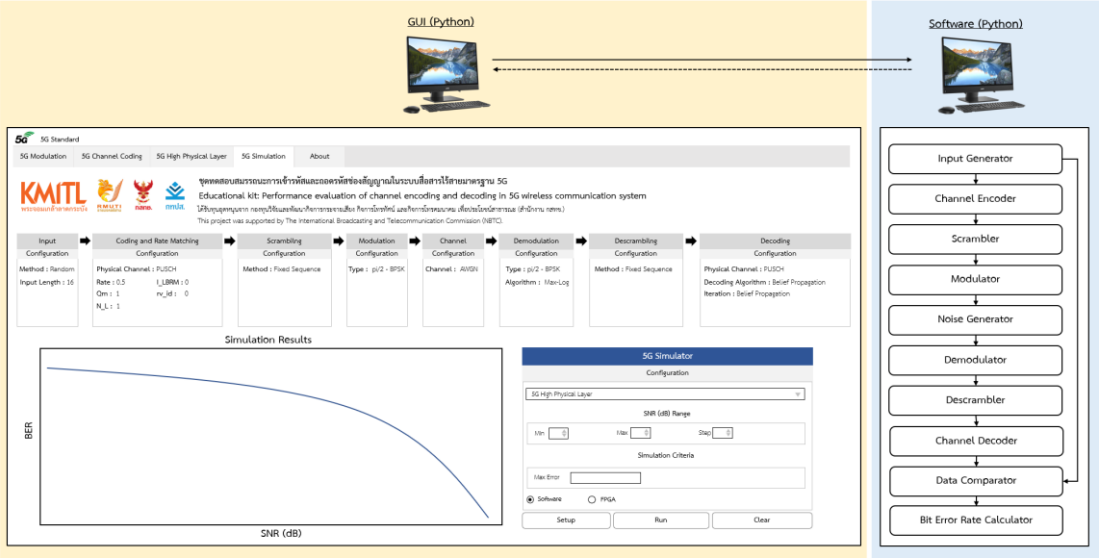
- ปุ่มวิทยุ สำหรับเลือกช่องทางการทดสอบสมรรถนะ ผู้ใช้สามารถเลือกช่องทางการทดสอบสมรรถนะได้ 2 ช่องทาง ได้แก่ ทดสอบสมรรถนะโดยใช้ซอฟต์แวร์ และทดสอบสมรรถนะโดยใช้อุปกรณ์ FPGA

2.2) ส่วนควบคุม ประกอบด้วย

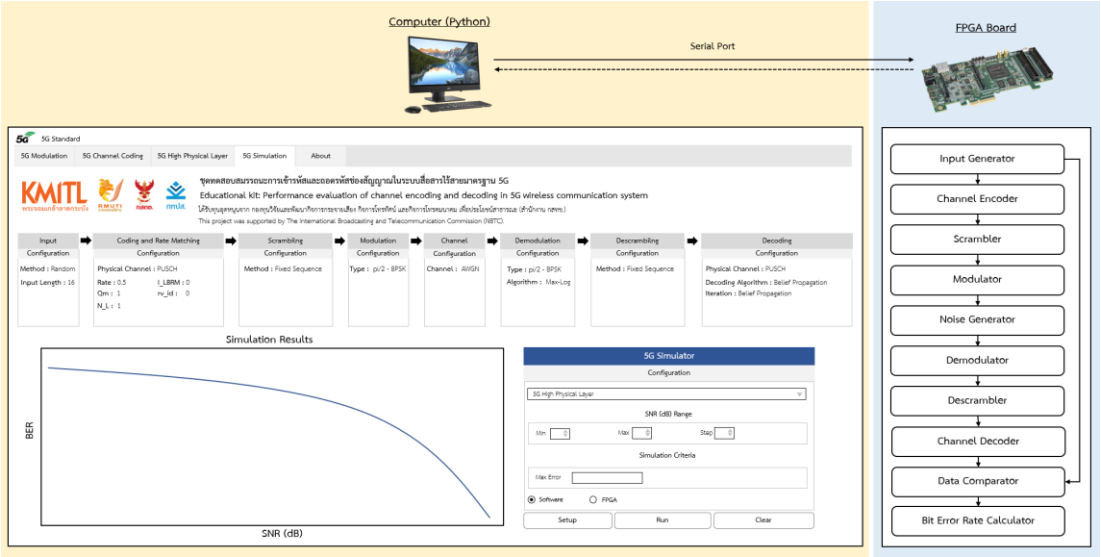
- ปุ่ม “Setup” สำหรับเรียกค่าพารามิเตอร์ของกระบวนการทำงานชั้นฟิสิคัลเลเยอร์ลำดับสูง จากหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G
- ปุ่ม “Run” สำหรับดำเนินทดสอบสมรรถนะ
- ปุ่ม “Clear” สำหรับล้างค่าพารามิเตอร์และผลการทดสอบสมรรถนะ

3) ส่วนแสดงผลสมรรถนะ เป็นส่วนแสดงผลการทดสอบสมรรถนะในรูปแบบกราฟของอัตราบิตผิดพลาด

นอกจากนี้ ผู้ใช้สามารถเลือกทดสอบสมรรถนะกระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงในระบบสื่อสารไร้สายด้วยซอฟต์แวร์ หรืออุปกรณ์ FPGA ได้โดยการเลือกปุ่มวิทยุที่อยู่ในส่วนการกำหนดค่า สำหรับการทำงานด้วยซอฟต์แวร์แต่ละองค์ประกอบจะเรียกใช้ Module การทำงานภาษา Python ภายในคอมพิวเตอร์ ซึ่งประกอบด้วย Module ตัวสร้างอินพุต Module ตัวเข้ารหัสและปรับอัตรารหัส Module ตัวสแครม Module ตัวมอดูเลชัน Module ตัวสร้างสัญญาณรบกวน Module ตัวดีมอดูเลชัน Module ตัวดีสแครม Module ตัวถอดรหัส Module ตัวเปรียบเทียบข้อมูล และ Module ตัวคำนวณอัตราบิตผิดพลาด ดังแสดงในรูปที่ 3.17 สำหรับการทำงานด้วยอุปกรณ์ FPGA หน้าต่าง GUI จะเรียกใช้ Module การทำงานบนอุปกรณ์ FPGA โดยส่งข้อมูลการกำหนดค่าและรับค่าเอาต์พุตผ่าน Serial Port ดังแสดงในรูปที่ 3.18



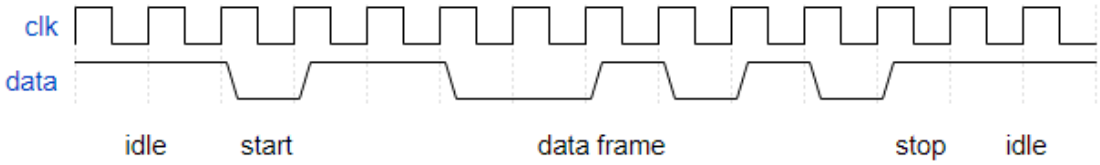
รูปที่ 3.17 ชุดทดสอบสมรรถนะกระบวนการทำงานชั้นฟิสิคัลเลเยอร์ลำดับสูงในระบบสื่อสารไร้สายเชื่อมต่อกับซอฟต์แวร์



รูปที่ 3.18 ชุดทดสอบสมรรถนะกระบวนการทำงานชั้นฟิสิคัลเลเยอร์ลำดับสูงในระบบสื่อสารไร้สายเชื่อมต่อกับอุปกรณ์ FPGA

3.3 การออกแบบการเชื่อมต่อระหว่างอุปกรณ์ FPGA กับคอมพิวเตอร์

ชุดการเรียนรู้และชุดทดสอบสมรรถนะถูกออกแบบให้สามารถประมวลผลกระบวนการของระบบสื่อสารมาตรฐาน 5G ได้ทั้งบนคอมพิวเตอร์ที่โปรแกรมทั้งสองทำงานและส่งข้อมูลไปประมวลผลบนอุปกรณ์ FPGA การส่งข้อมูลไปประมวลผลบนอุปกรณ์ FPGA จากคอมพิวเตอร์ จะส่งผ่านช่องทางข้อมูลอนุกรมหรือ serial port ของระหว่างทั้งคอมพิวเตอร์และอุปกรณ์ FPGA ภายใต้โพรโทคอล universal asynchronous receiver-transmitter หรือ UART โดยข้อมูลชุดละ 8 บิต บิต start และบิต stop จะสื่อสารระหว่างสองอุปกรณ์ตามโพรโทคอล UART ดังรูปที่ 3.19



รูปที่ 3.19 ชุดบิตข้อมูลการสื่อสารภายใต้โพรโทคอล UART

เพื่อให้การสื่อสารเข้าใจทั้งสองอุปกรณ์ อุปกรณ์ทั้งทั้งสองสามารถแยกประเภทสัญญาณได้จากชุดข้อมูลส่วนหัว (header) ที่มีความยาว 8 บิตทั้งหมด ประเภทและความยาวของสัญญาณที่จำแนก สามารถสรุปได้ดังตารางที่ 3.1

ตารางที่ 3.1 ประเภทและความยาวของสัญญาณจำแนกระหว่างคอมพิวเตอร์และอุปกรณ์ FPGA

ประเภทสัญญาณ	ส่วนหัว	ความยาวข้อมูล
ส่วนหัวทั่วไป		
Modulation flag	00010001	-
Channel coding flag	00010010	-
Simulation flag	00011000	16
ส่วนหัวสำหรับระบบย่อย Simulator controller ภายใต้ Simulator		
snrValue	00011010	4080
snrNumber	00011011	8
maxBlockError	00011100	4080
maxBlockLength	00011101	32
ส่วนหัวสำหรับระบบย่อย Random input generator ภายใต้ Simulator		
uniformBitLength	00100001	16
ส่วนหัวสำหรับระบบย่อย LDPC encoder ภายใต้ Channel encoder และ Simulator		
Rate	00110001	8
liftingSize	00110010	16
I_LBRM	00110011	8
N_L	00110100	8
rv_id	00110101	8
infoLength	00110110	16
infoBits	00110111	8448
channelType	01000101	8
modScheme	01010001	8
ส่วนหัวสำหรับระบบย่อย Polar encoder ภายใต้ Channel encoder และ Simulator		
infoBits	01000001	1712
infoLength	01000010	16
codewordLength	01000011	16
rntiBits	01000100	16
channelType	01000101	8
ส่วนหัวสำหรับระบบย่อย Modulator ภายใต้ Simulator		
modScheme	01010001	8
modBitLength	01010010	16
modBitInput	01010011	1024
ส่วนหัวสำหรับระบบย่อย Noise generator ภายใต้ Simulator		
channelAndSNR	00100011	24
sqrtVar	00100100	16
gaussianSymbolLength	00100101	16
gaussianValueInput	00100110	32768

ประเภทสัญญาณ	ส่วนหัว	ความยาวข้อมูล
complexChannel Coefficient	00100111	16384
inverseComplex ChannelCoefficient	00101000	16384
ส่วนหัวสำหรับระบบย่อย Demodulator ภายใต้ Simulator		
modScheme	01010001	8
channelAndSNR	00100011	24
inverseSqrtVar	01010101	16
demodEquation	01010110	8
demodSymbolLength	01010111	16
demodValueInput	01011000	32768
ส่วนหัวสำหรับระบบย่อย LDPC decoder ภายใต้ Channel decoder และ Simulator		
channelType	01000101	8
decoderConfig	00111000	8
iteration	00111001	8
offsetAndScale	00111010	32
processorNumber	00111011	8
liftingSize	00111101	16
ldpcLLRSymbolLength	00111110	16
ldpcLLRInput	00111111	8192
ส่วนหัวสำหรับระบบย่อย Polar decoder ภายใต้ Channel decoder และ Simulator		
rntiBits	01000100	16
channelType	01000101	8
offsetAndScale	00111010	32
decoderConfig	01000111	8
crcPolynomial	01001000	24
frozenPosition	01001010	2048
llrSymbolLength	01001011	16
llrInput	01001100	8192

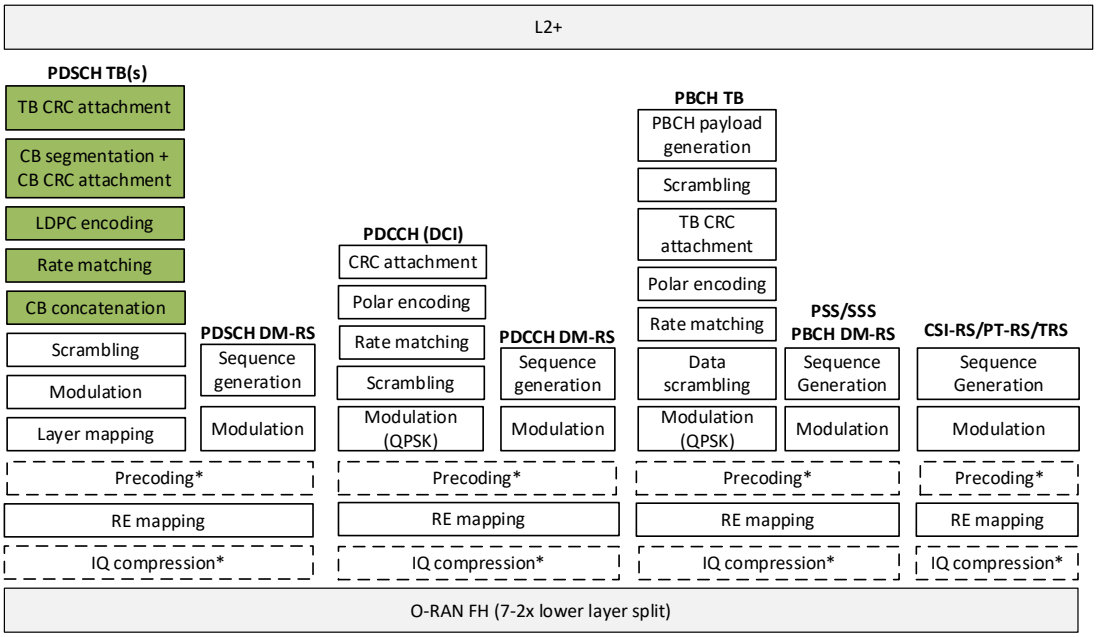
3.4 การออกแบบการ์ดเร่งความเร็ว FEC

กระบวนการชั้นฟิสิคัลเลเยอร์ลำดับสูงและลำดับต่ำ สำหรับช่องสัญญาณ Downlink แสดงในรูปที่ 3.20 และสำหรับช่องสัญญาณ Uplink แสดงในรูปที่ 3.21 ซึ่งมาตรฐาน O-RAN ได้กำหนดให้มีการใช้งานการ์ดเร่งความเร็ว FEC แบบ PDSCH_FEC โดยการ์ดเร่งความเร็วจะทำกระบวณการต่อไปนี้

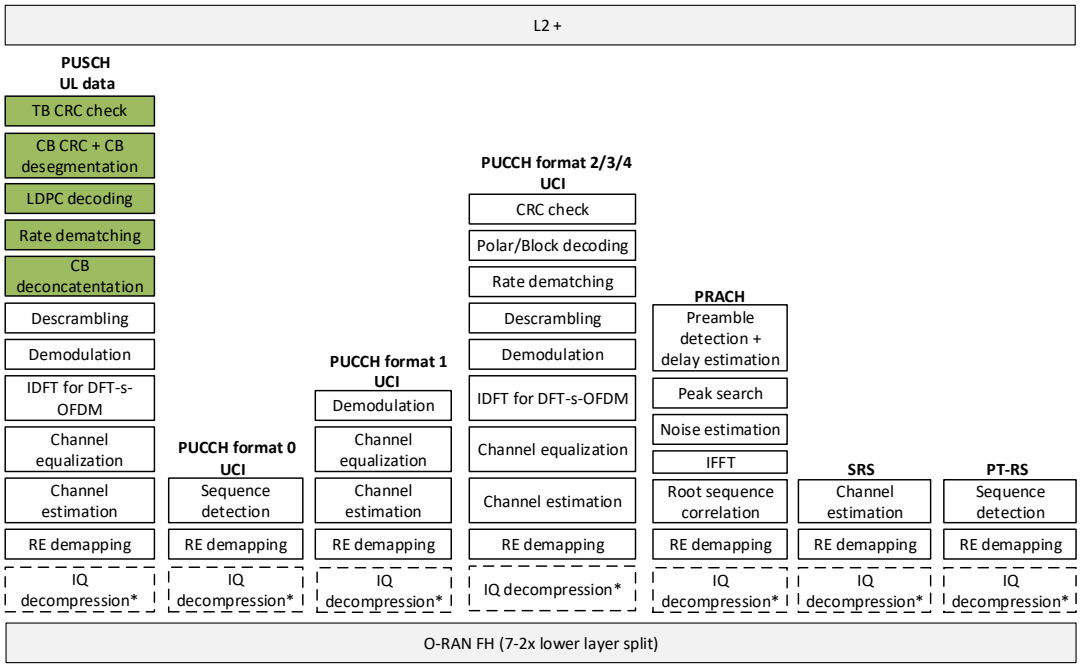
1. TB CRV attachment
2. CB segmentation + CB CRV attachment
3. LDPC encoding
4. Rate matching
5. CB concatenation

นอกจากนี้ มาตรฐาน O-RAN ได้กำหนดให้มีการใช้งานการ์ดเร่งความเร็ว FEC แบบ PUSCH_FEC โดยการ์ดเร่งความเร็วจะทำกระบวณการต่อไปนี้

1. TB CRC check
2. CB CRC + CB desegmentation
3. LDPC decoding
4. Rate dematching
5. CB deconcatentation



รูปที่ 3.20 การใช้งานการ์ดเร่งความเร็ว FEC แบบ PDSCH_FEC

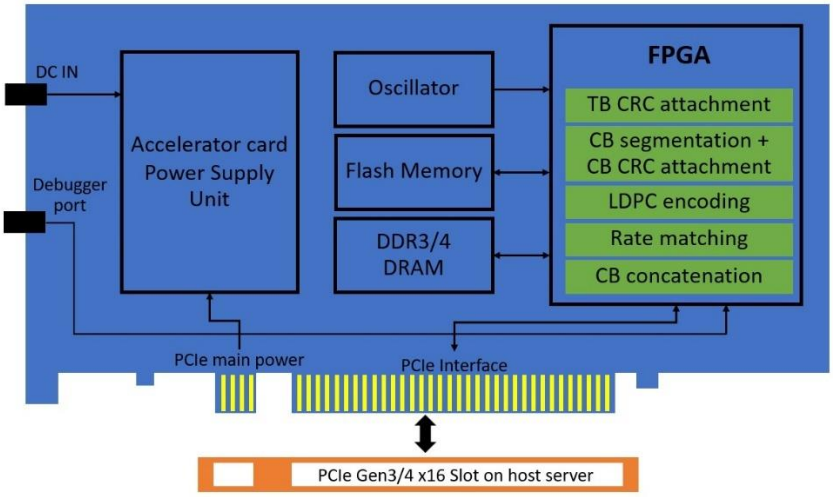


รูปที่ 3.21 การใช้งานการ์ดเร่งความเร็ว FEC แบบ PUSCH_FEC

3.5 การออกแบบการเชื่อมต่อระหว่างการ์ดเร่งความเร็ว FEC กับคอมพิวเตอร์

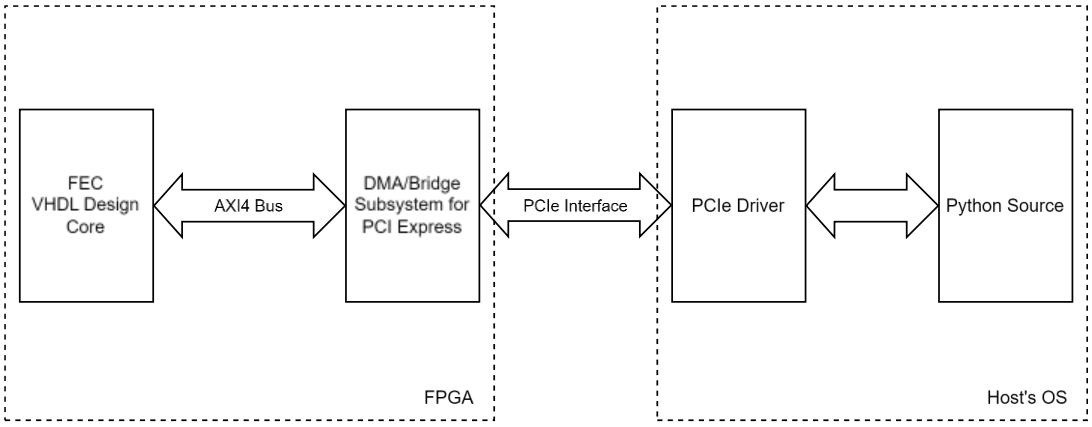
3.5.1 การออกแบบการ์ดเร่งความเร็ว FEC แบบ AAL_PDSCH_FEC ด้วยอุปกรณ์ FPGA

การออกแบบการ์ดเร่งความเร็ว FEC แบบ AAL_PDSCH_FEC ด้วยอุปกรณ์ FPGA จะเป็นการนำซอฟต์แวร์ที่พัฒนาขึ้นด้วยภาษา VHDL ที่ใช้งานในชุดการเรียนรู้และทดสอบสมรรถนะ มาพัฒนาในรูปแบบของอุปกรณ์ FPGA ที่มีส่วนต่อประสาน PCIe ดังรูปที่ 3.22 โครงการนี้ใช้ AMD Virtex UltraScale+ FPGA VCU118 Evaluation Kit เป็น FPGA สำหรับการออกแบบและทดสอบการ์ดเร่งความเร็วเนื่องจากอุปกรณ์ดังกล่าวมีส่วนต่อประสาน PCIe พร้อมซอฟท์แวร์ต่าง ๆ ที่จำเป็นสำหรับการรับส่งข้อมูลผ่านส่วนต่อประสาน PCIe



รูปที่ 3.22 การ์ดเร่งความเร็ว FEC แบบ AAL_PDSCH_FEC ด้วยอุปกรณ์ FPGA

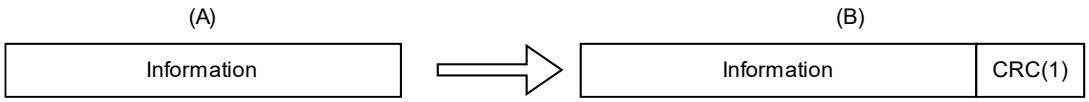
โดยทำการออกแบบระบบให้สามารถเชื่อมกับตัว IP-Core DMA/Bridge Subsystem for PCI Express ของ Xilinx ซึ่งเป็นโมดูลที่จัดการเฟรมข้อมูลสำหรับการรับส่งข้อมูลผ่าน PCIe จากนั้นจึงทำการใช้โปรแกรมภาษา Python เพื่อเรียก Driver ของ AMD ซึ่งเป็นโปรแกรมภาษา C เมื่อเรียกใช้ Driver ดังกล่าวก็จะสามารถทำการทดสอบการรับส่งข้อมูลระหว่าง FPGA และคอมพิวเตอร์ผ่านส่วนต่อประสาน PCIe ได้ดังแสดงในรูปที่ 3.23



รูปที่ 3.23 การเชื่อมต่อระหว่างการ์ดเร่งความเร็วกับระบบปฏิบัติการ

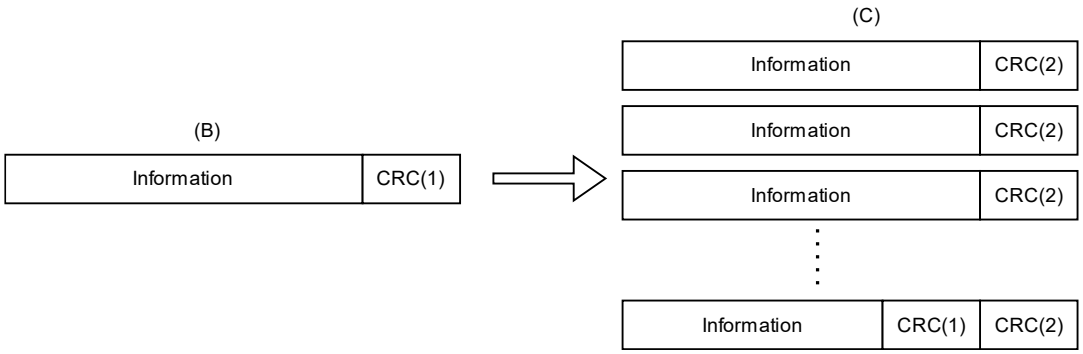
เมื่อการ์ดเร่งความเร็ว FEC แบบ AAL_PDSCH_FEC ได้รับข้อมูลจากระบบปฏิบัติการจะดำเนินการขั้นตอนต่อไปนี้

1) TB CRC attachment เป็นกระบวนการเข้ารหัส CRC ด้วยโพลิโนเมียลตามมาตรฐาน 5G ดังรูปที่ 3.24



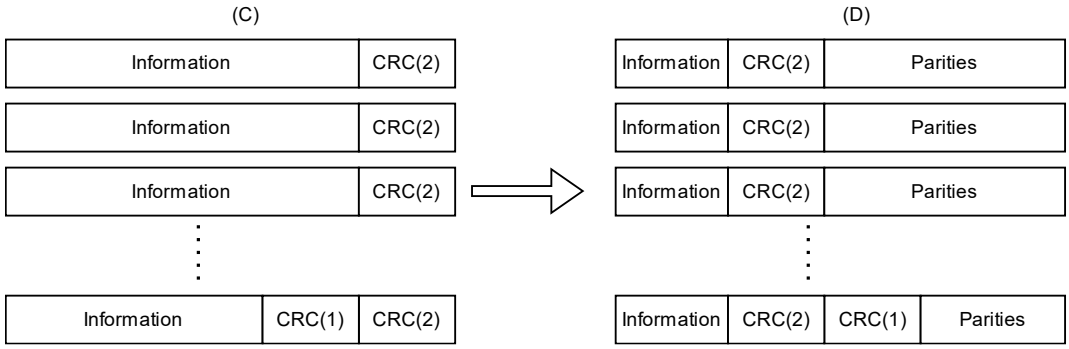
รูปที่ 3.24 TB CRC attachment

2) CB segmentation + CB CRC attachment เป็นกระบวนการตัดข้อมูลออกเป็นบล็อกย่อย ๆ และเข้ารหัส CRC ด้วยโพลิโนเมียลตามมาตรฐาน 5G ซ้ำอีกรอบให้แต่ละบล็อกดังรูปที่ 3.25



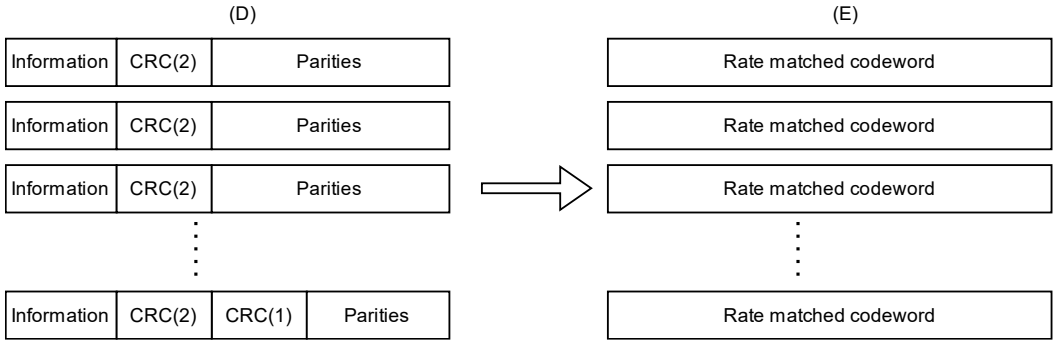
รูปที่ 3.25 CB segmentation + CB CRC attachment

3) LDPC encoding เป็นกระบวนการเข้ารหัส LDPC ตามมาตรฐาน 5G ดังรูปที่ 3.26



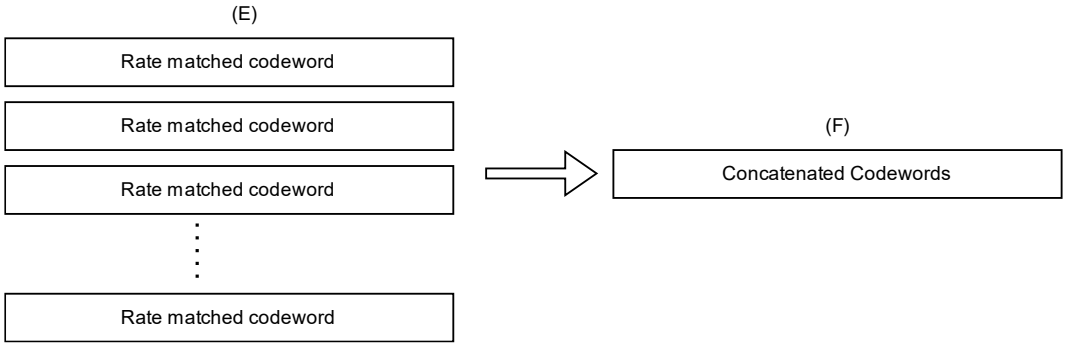
รูปที่ 3.26 LDPC encoding

4) Rate Matching เป็นกระบวนการปรับอัตรารหัสตามมาตรฐาน 5G ดังรูปที่ 3.27



รูปที่ 3.27 Rate Matching

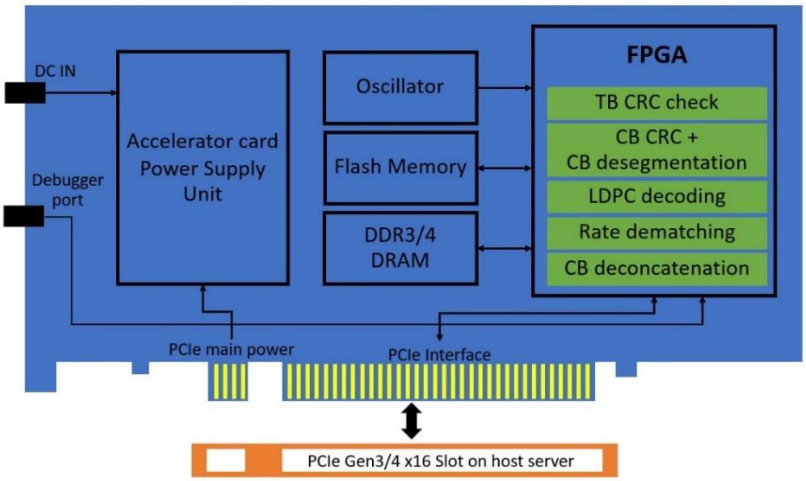
5) CB concatenation เป็นกระบวนการที่นำบล็อกรหัสมาต่อกันเพื่อส่งต่อไปยังกระบวนการอื่นดังรูปที่ 3.28



รูปที่ 3.28 CB concatenation

3.5.2 การออกแบบการ์ดเร่งความเร็ว FEC แบบ AAL_PUSCH_FEC ด้วยอุปกรณ์ FPGA

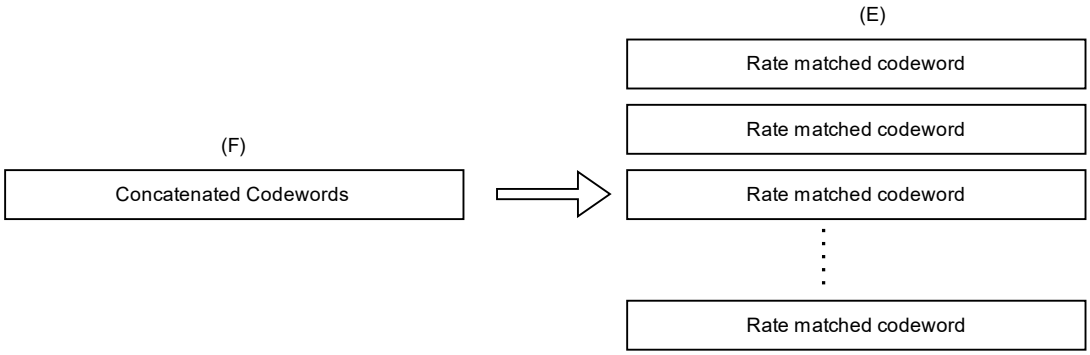
การออกแบบการ์ดเร่งความเร็ว FEC แบบ AAL_PUSCH_FEC ด้วยอุปกรณ์ FPGA จะทำลักษณะเดียวกับการออกแบบการ์ดเร่งความเร็ว FEC แบบ AAL_PDSCH_FEC ในหัวข้อก่อนหน้า เพียงแต่เปลี่ยนกระบวนการภายในให้สอดคล้องกับที่มาตรฐานกำหนด ดังรูปที่ 3.29



รูปที่ 3.29 การ์ดเร่งความเร็ว FEC แบบ AAL_PUSCH_FEC ด้วยอุปกรณ์ FPGA

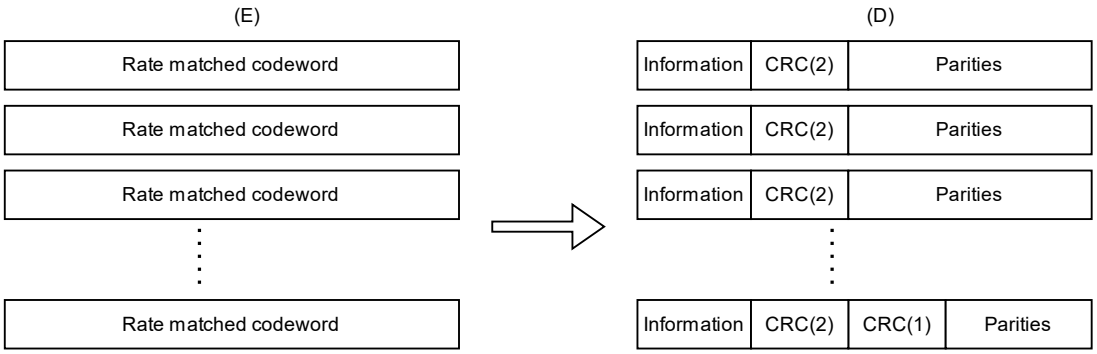
ทำให้เมื่อการ์ดเร่งความเร็ว FEC แบบ AAL_PUSCH_FEC ได้รับข้อมูลจากระบบปฏิบัติการ จะดำเนินการขั้นตอนต่อไปนี้

1) CB deconcatenation เป็นกระบวนการในตัดบล็อกคำรหัสจากช่องสัญญาณออกเป็นบล็อกคำรหัสย่อยดังรูปที่ 3.30



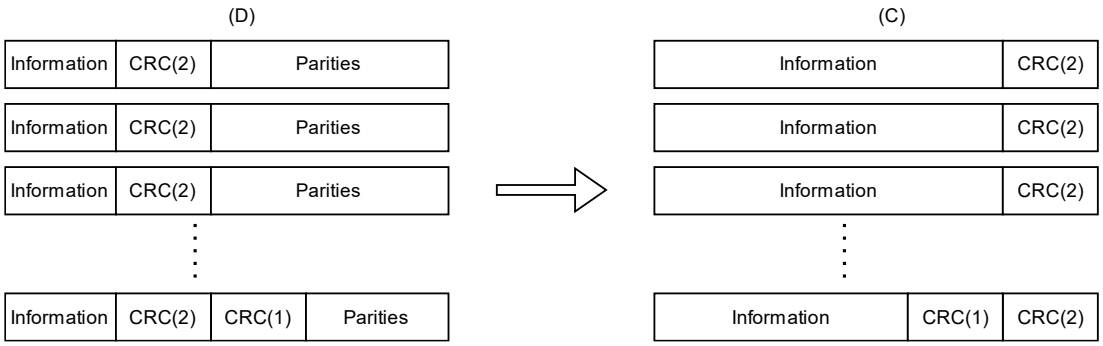
รูปที่ 3.30 CB deconcatenation

2) Rate dematching เป็นกระบวนการในการปรับขนาดคำรหัสจากช่องสัญญาณให้มีขนาดก่อนจะถูกปรับอัตรารหัสดังรูปที่ 3.31



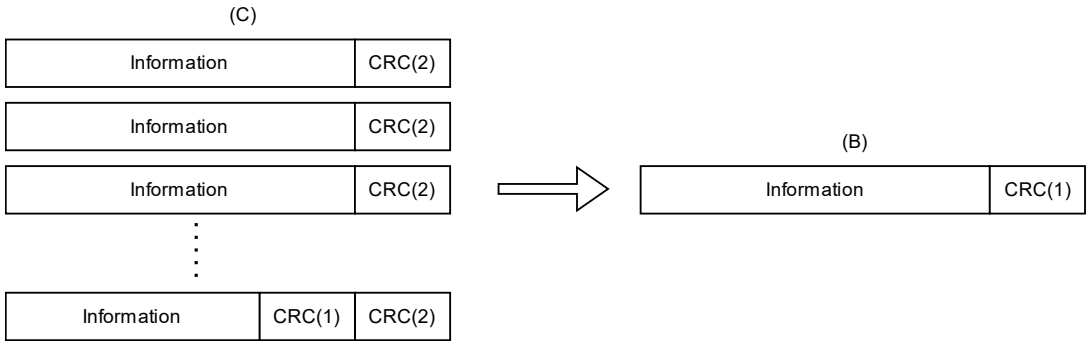
รูปที่ 3.31 Rate dematching

3) LDPC decoding เป็นกระบวนการถอดรหัส LDPC ตามมาตรฐาน 5G ดังรูปที่ 3.32



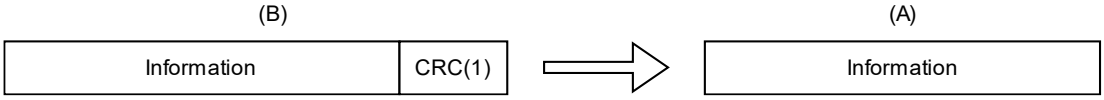
รูปที่ 3.32 LDPC decoding

4) CB CRC + CB desegmentation เป็นกระบวนการถอดรหัส CRC ของแต่ละบล็อกรหัสและนำบล็อกรหัสดังกล่าวกลับมาต่อกันเป็นบล็อกรหัสยาวดังรูปที่ 3.33



รูปที่ 3.33 CB CRC + CB desegmentation

5) TB CRC check เป็นกระบวนการถอดรหัส CRC ของบล็อกรหัสหลักซึ่งเป็นกระบวนการสุดท้ายของการถอดรหัสดังรูปที่ 3.34



รูปที่ 3.34 TB CRC check

3.6 รายละเอียดการสร้างข้อมูลอินพุตแบบสุ่ม

สำหรับการออกแบบวงจรสุ่มข้อมูลอินพุตบนอุปกรณ์ FPGA ได้เลือกใช้วงจร CTG 3 ส่วนประกอบ ความยาว 32 บิต มีขนาดคาบเท่ากับ 2^{88} [1] แสดงอัลกอริทึมดังรูปที่ 3.35 ข้อมูลสุ่มขนาด 32 บิตนี้จะถูกนำไปใช้กับระบบการสุ่มข้อมูลอินพุต อีกทั้งยังถูกนำไปใช้กับระบบการออกแบบสัญญาณรบกวนแบบเกาสเซียนเช่นเดียวกัน โดยสำหรับระบบการสุ่มข้อมูลข้อมูลอินพุตข้อมูลที่สุ่มได้จะถูกใช้งานเป็นข้อมูลอินพุตโดยตรง ขณะที่ระบบการออกแบบสัญญาณรบกวนแบบเกาส์เซียนจะถูกนำไปใช้เป็นอินพุตค่าทศนิยมขนาด [0,1)

อัลกอริทึม 1 วงจร CTG 3 ส่วนประกอบ

กำหนดค่าเริ่มต้น s_1, s_2, s_3

$$b_1 = \left(\left(\left(s_1 \ll 13\right) \wedge s_1\right) \gg 19\right)$$

..

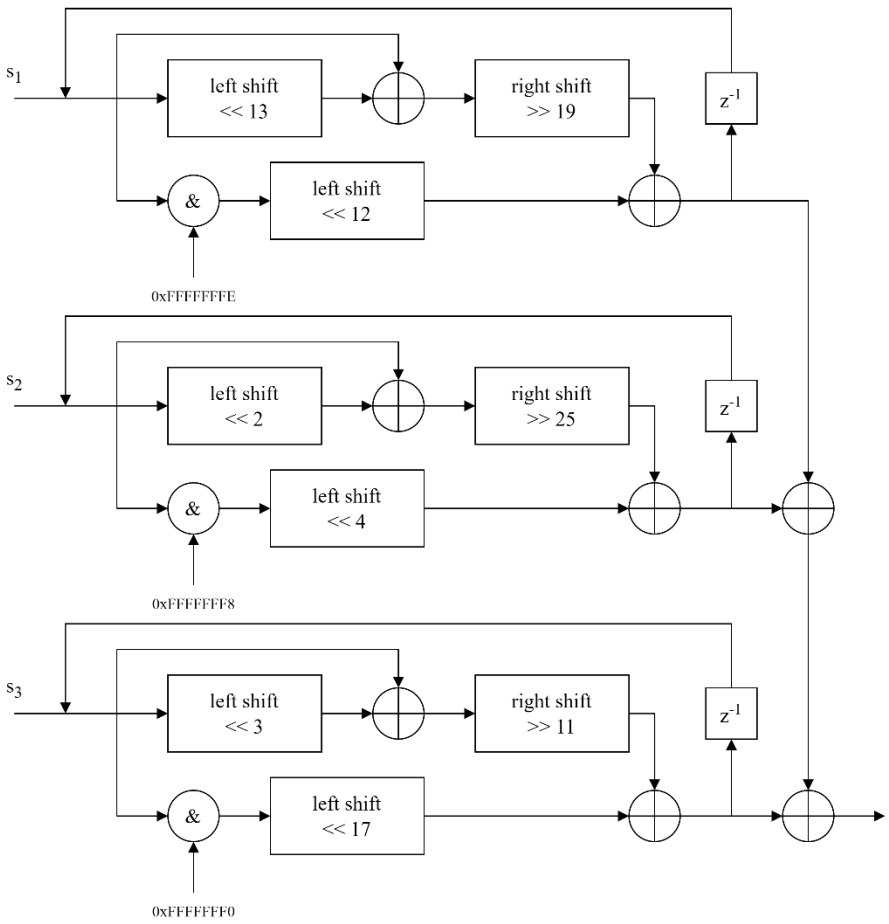
$$b_2 = \left(\left(\left(s_2 \ll 2\right) \wedge s_2\right) \gg 25\right)$$

$$s_2 = \left(\left(\left(s_2 \,\&\, 0\text{xFFFFFFF8}\right) \ll 4\right) \wedge b_2\right)$$

$$b_3 = \left(\left(\left(s_3 \ll 3\right) \wedge s_3\right) \gg 11\right)$$

$$s_3 = \left(\left(\left(s_3 \,\&\, 0\text{xFFFFFFF0}\right) \ll 17\right) \wedge b_3\right)$$

ผลลัพธ์เท่ากับ $s_1 \wedge s_2 \wedge s_3$



รูปที่ 3.35 แผนภาพวงจร CTG 3 ส่วนประกอบที่ใช้งานบนอุปกรณ์ FPGA

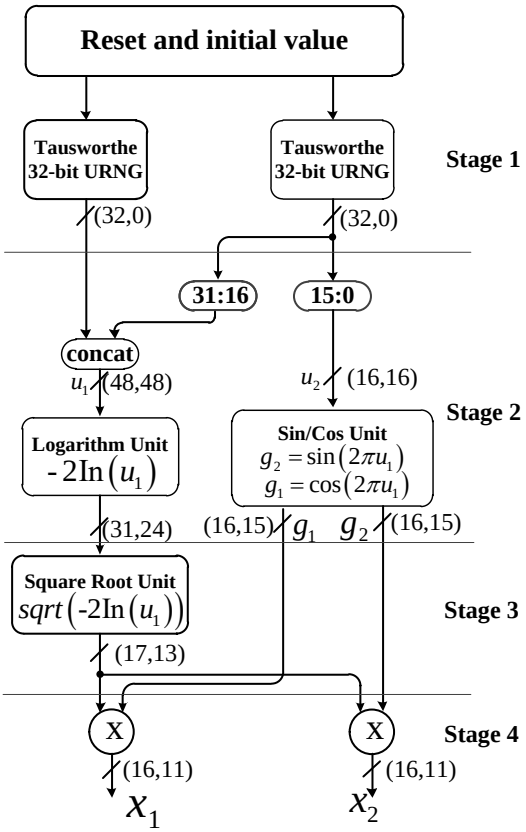
3.7 รายละเอียดการสร้างช่องสัญญาณสื่อสารแบบ AWGN

การสุ่มสัญญาณรบกวนเกาส์เซียนสามารถทำได้โดยวิธีการสุ่มเลข หนึ่งในนั้นคือการแปลงบ็อกซ์-มูลเลอร์ (Box-Muller transform) วิธีการดังกล่าวจะให้ผลลัพธ์เป็นคู่เลขสุ่มที่เป็นอิสระ มีมาตรฐาน และมีการกระจายเกาส์เซียนที่มีค่าเฉลี่ยเท่ากับ 0 และความแปรปรวนเท่ากับ 1 ที่มีช่วง [-1, 1] โดยที่วิธีการดังกล่าวต้องรับค่าคู่อินพุตที่มีการกระจายแบบสม่ำเสมอ (uniform distribution) ที่มีช่วง [0, 1) เขียนสมการได้ดังนี้

$$x_0 = \sqrt{-2\ln u_0}\cos\left(2\pi u_1\right) \tag{3.1}$$

$$z_1 = \sqrt{-2\ln u_0}\sin\left(2\pi u_1\right) \tag{3.2}$$

การออกแบบวงจรสร้างสัญญาณรบกวนเกาส์เซียนขาวแบบบวกบนอุปกรณ์ FPGA ยังได้อาศัยการแปลงบ็อกซ์-มูลเลอร์ ดังรูปที่ 3.36 ประกอบไปด้วยขั้นตอนและฟังก์ชันทางคณิตศาสตร์ดังนี้



รูปที่ 3.36 แผนภาพการสร้างสัญญาณรบกวนเกาส์เซียนขาวแบบบวกบนอุปกรณ์ FPGA

1) การสุ่มเลขทศนิยม [0, 1) ที่มีการกระจายแบบสม่ำเสมอ

2) การคำนวณฟังก์ชันลอการิทึมธรรมชาติ (natural logarithm)

3) การคำนวณค่ารากที่สอง (square root)

4) การคำนวณฟังก์ชันโคซายน์ (cosine function)

โดยกำหนดให้เอาต์พุตของการคำนวณฟังก์ชันลอการิทึมธรรมชาติ การคำนวณค่ารากที่สอง การคำนวณฟังก์ชันโคซายน์ และการคำนวณฟังก์ชันซายน์ ดังสมการที่ 3.3 3.4 3.5 และ 3.6 ตามลำดับ

$$f_0(u_0) = -\ln(u_0) \tag{3.3}$$

$$g_0(u_0) = \sqrt{f_0(u_0)} \tag{3.4}$$

$$g_1(u_1) = \sqrt{2}\cos(2\pi u_1) \tag{3.5}$$

$$g_2(u_1) = \sqrt{2}\sin(2\pi u_1) \tag{3.6}$$

โดยที่ u_0 และ u_2 คือเลขสุ่มทศนิยม [0, 1) ที่มีการกระจายแบบสม่ำเสมอ ผลลัพธ์ของการสร้างจะได้สัญญาณรบกวนแบบเกาส์เซียนขาวแบบบวก x_1 และ x_2 ที่มีค่าเฉลี่ยเท่ากับ 0 และค่าความแปรปรวนเท่ากับ 1 หรือ $N(0,1)$ ดังนี้

$$x_1 = q_0(u_1)q_1(u_2) \tag{3.7}$$

$$x_2 = q_0(u_1)q_2(u_2) \tag{3.8}$$

รายละเอียดของขั้นตอนการคำนวณและฟังก์ชันทางคณิตศาสตร์ สามารถกล่าวได้ดังต่อไปนี้

การสุ่มเลขทศนิยมที่มีการกระจายแบบสม่ำเสมอ

เลขสุ่มทศนิยมที่มีการกระจายแบบสม่ำเสมอเป็นขั้นตอนแรกในการสร้างสัญญาณรบกวน โดยได้เลือกใช้วงจร CTG ความยาว 32 บิต ขนาดคาบ 2^{88} จำนวน 2 วงจร เช่นเดียวกับวงจรสุ่มข้อมูลอินพุต แสดงดังรูปที่ 3.35 จะให้เอาต์พุตเป็นค่าไบนารีความยาว 32 บิต ซึ่งสามารถมองเป็นค่าทศยนิยมช่วง [0, 1) ในระบบเลขฐานสิบ โดยมองเป็นค่า fixed point ที่มีความละเอียด fi(32, 32)

การคำนวณฟังก์ชันลอการิทึมธรรมชาติ

การคำนวณฟังก์ชันลอการิทึมธรรมชาติ จะทำการคำนวณด้วยวิธีการประมาณค่าด้วยพหุนาม (polynomial approximation) โดยได้เลือกใช้พหุนามดีกรี 4 ตามสมการต่อไปนี้

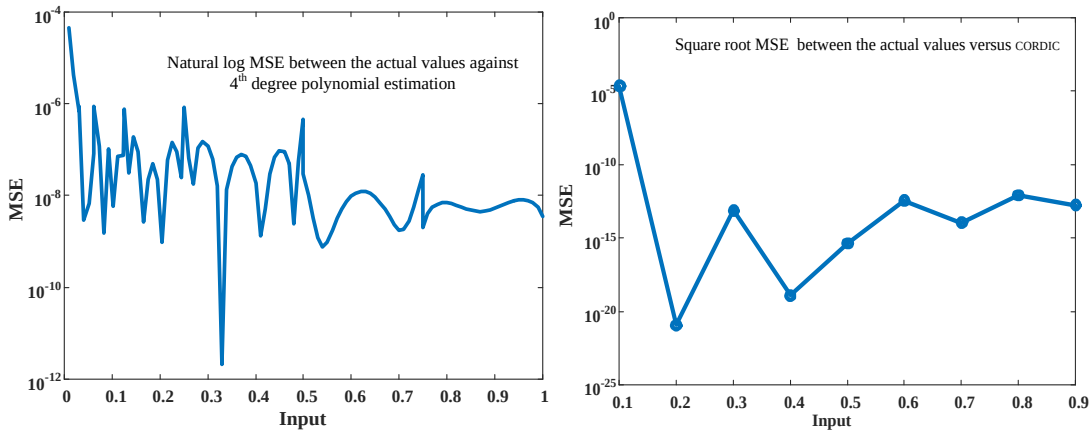
$$p_i(x) = a_i x^4 + b_i x^2 + c_i x + d_i \quad (3.9)$$

โดยที่ a c และ d คือค่าสัมประสิทธิ์พหุนาม (polynomial coefficients)

การประมาณฟังก์ชันลอการิทึมธรรมชาติจะทำการแบ่งช่วงของฟังก์ชันเป็นทั้งหมด 7 ช่วง โดยกำหนดให้ $i = [1,\ldots,7]$ แต่ละช่วงจะมีค่าสัมประสิทธิ์ของสมการพหุนามที่แตกต่างกัน แสดงดังตารางที่ 3.2 โดยค่าสัมประสิทธิ์เหล่าจะถูกบันทึกไว้ในหน่วยความจำบนอุปกรณ์ FPGA ผลของการแบ่งฟังก์ชันเป็น 7 ส่วนทำให้ความคลาดเคลื่อนของการประมาณฟังก์ชันลอการิทึมธรรมชาติในรูปของค่าเฉลี่ยของผลต่างยกกำลังสอง (mean squared error) หรือ MSE เป็นที่ยอมรับได้ ซึ่งมีค่าต่ำกว่า 10^{-6} แสดงความคลาดเคลื่อนดังรูปที่ 3.37

ตารางที่ 3.2 ค่าสัมประสิทธิ์สมการพหุนามของช่วงการประมาณ 7 ช่วง

	ค่าสัมประสิทธิ์ a_i	ค่าสัมประสิทธิ์ b_i	ค่าสัมประสิทธิ์ c_i	ค่าสัมประสิทธิ์ d_i
$i=1$	0x5ABE0000000000	0xF7C7B333333333	0x00514A3D70A3D7	0xFFFD1339C0EBEE
$i=2$	0x06D74CCCCCCCCD	0xFE92251EB851EC	0x002169FBE76C8B	0xFFFD86F9DB22D1
$i=3$	0x00DA8B851EB852	0xFFA4A3D70A3D71	0x0010B28F5C28F6	0xFFFDDFC504816F
$i=4$	0x001B5083126E98	0xFFE9299999999A	0x00085916872B02	0xFFFE387FCB923A
$i=5$	0x06D404EA4A8C15	0xF494FDF3B645A2	0x0859096BB98C7E	0xFD227525460AA6
$i=6$	0x01671DE69AD42C	0xFC113A92A30553	0x04DF34D6A161E5	0xFDADCC63F14120
$i=7$	0x008122FAD6CB53	0xFE045A1CAC0831	0x0374538EF34D6A	0xFE062B6AE7D567



(ก) (ข)

รูปที่ 3.37 ค่า MSE ของการคำนวณฟังก์ชันทางคณิตศาสตร์ประเมินค่าด้วยโปรแกรม MATLAB
(ก) คำนวณลอการิทึมธรรมชาติด้วยการประมาณค่าด้วยพหุนาม
(ข) คำนวณรากที่สองโดยใช้อัลกอริทึม CORDIC

การคำนวณรากที่สอง

การคำนวณรากที่สองจะใช้การคำนวณตามอัลกอริทึม CORDIC ในโหมดเวกเตอร์ไฮเปอร์โบลิก (Hyperbolic vectoring) อัลกอริทึม CORDIC ถือเป็นอัลกอริทึมวนซ้ำ (iterative) ใช้ในการประมาณค่าของฟังก์ชันทางคณิตศาสตร์ที่เป็นมิตรต่อฮาร์ดแวร์เนื่องจากการคำนวณใช้ตัวดำเนินการพื้นฐาน เช่น ตารางค้นหาสำหรับการคูณค่าคงที่ การเลื่อน และการบวก เท่านั้น การคำนวณอัลกอริทึม CORDIC ในโหมดเวกเตอร์ไฮเปอร์โบลิก จะให้ผลลัพธ์ดังสามสมการต่อไปนี้

$$x_{i+1} = x_i + y_i d_i 2^{-i} \tag{3.10}$$

$$y_{i+1} = y_i + x_i d_i 2^{-i} \tag{3.11}$$

$$z_{i+1} = z_i - d_i \tanh^{-1}\left(2^{-i}\right) \tag{3.12}$$

โดยที่ i คือจำนวนรอบการคำนวณและ

$$d_i = \begin{cases} +1 & y_i < 0 \\ -1 & \text{อื่น ๆ} \end{cases} \tag{3.13}$$

จากนั้นเมื่อจำนวนรอบการคำนวณ N ลู่เข้าอนันต์ สมการที่ (3.10) – (3.12) จะให้ผลลัพธ์การประมาณดังต่อไปนี้

$$x_N \approx A_N \sqrt{x_0^2 - y_0^2} \tag{3.14}$$

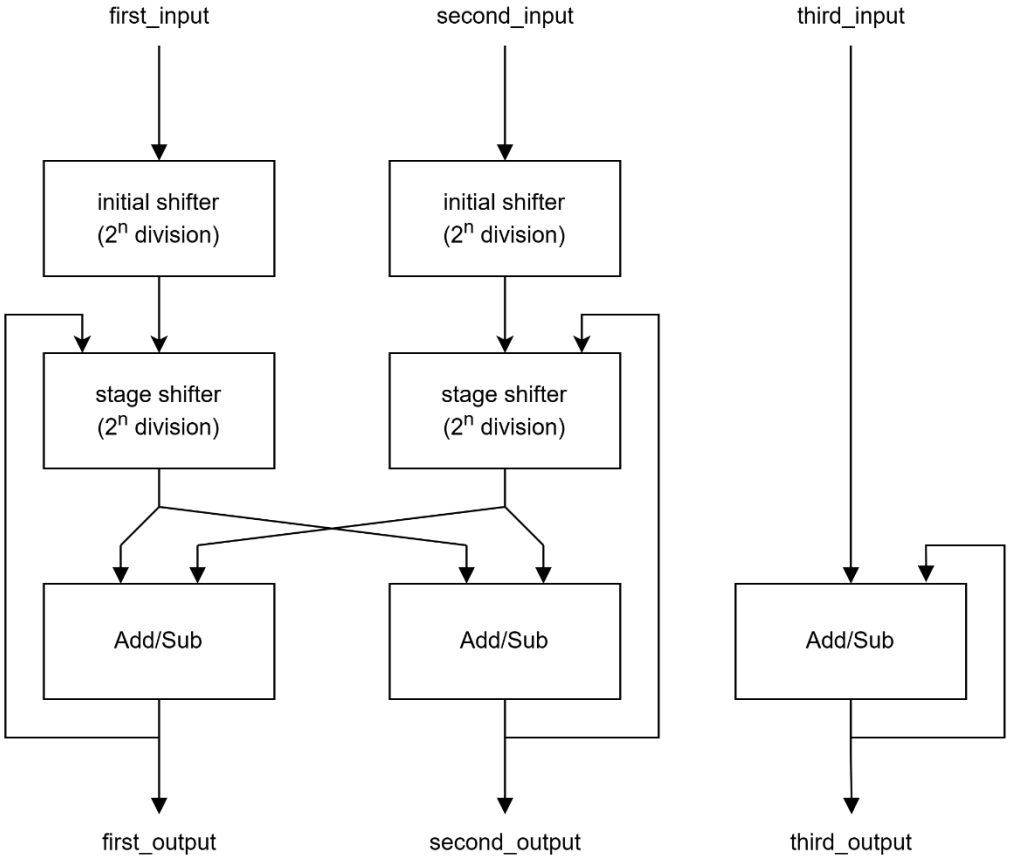
$$y_N \approx 0 \tag{3.15}$$

$$z_N \approx z_0 + \tanh^{-1}\left(\frac{y_0}{x_0}\right) \tag{3.16}$$

โดยที่ค่าคงที่ A_N สามารถทำการคำนวณไว้ก่อนหน้าแล้วเก็บไว้ในหน่วยความจำบนอุปกรณ์ FPGA ได้ดังนี้

$$A_N = \prod_{i=1}^{N} \sqrt{1-2^{-2i}} \tag{3.17}$$

จะสังเกตได้ว่าสมการที่ (3.14) ให้ผลลัพธ์การประมาณเป็นฟังก์ชันรากที่สอง ซึ่งจะใช้ความสัมพันธ์ตามสมการดังกล่าวในการคำนวณค่ารากที่สองบนอุปกรณ์ FPGA สำหรับจำนวนรอบ N ในการคำนวณอัลกอริทึม CORDIC จะใช้อยู่ที่ประมาณ 11 รอบ ซึ่งจะให้ค่าความคลาดเคลื่อนที่ยอมรับได้ประมาณ 10^{-12} แสดงดังรูปที่ 3.38 วงจรการคำนวณรากที่สองนี้ใช้อินพุตเป็นค่า fixed point ที่ความละเอียด fi(32, 28) และให้เอาต์พุตเป็นค่า fixed point ที่ความละเอียด fi(17, 13)



รูปที่ 3.38 วงจรคำนวณฟังก์ชันคณิตศาสตร์โดยอัลกอริทึม CORDIC

การคำนวณฟังก์ชันโคซายน์

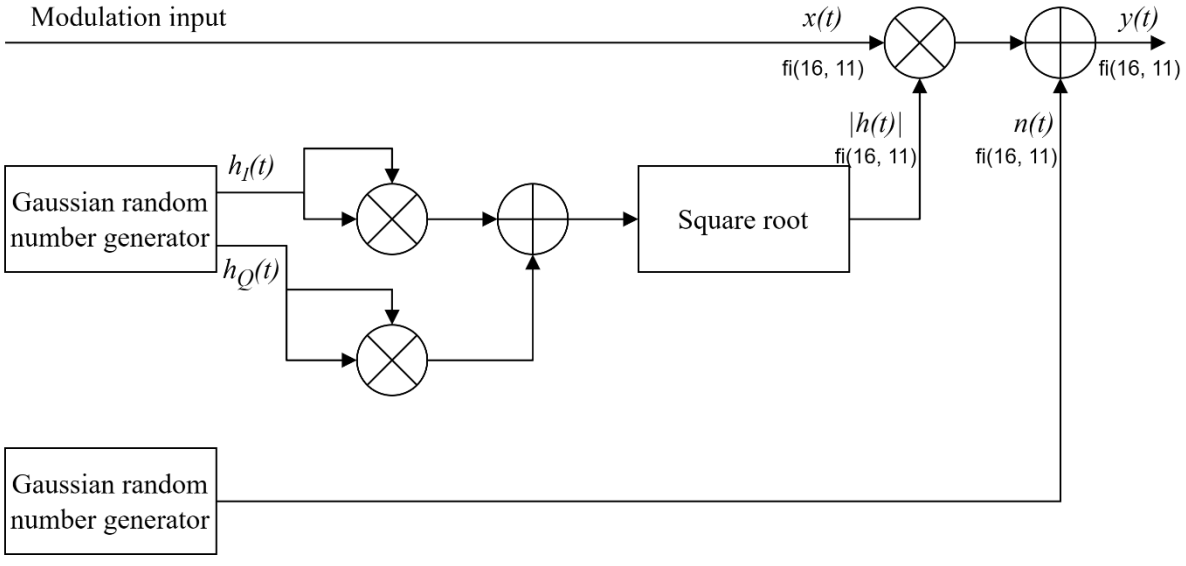
การสร้างสัญญาณโคซายน์และรวมถึงซายน์ จะทำการเก็บค่าคงที่จำนวน 1 ใน 4 ส่วนของฟังก์ชันโคซายน์เต็มคาบ โดยจะคำนวณค่าล่วงหน้าก่อนหน้าโดยโปรแกรม MATLAB เนื่องจากฟังก์ชันโคซายน์และซายน์มีความสมมาตร จึงทำให้เรียกค่าของฟังก์ชันโคซายน์ได้เต็มคาบขณะที่เก็บค่าไว้เพียง 1 ใน 4 ของคาบได้ ค่าที่ถูกคำนวณไว้ก่อนหน้าจะสามารถเข้าถึงได้โดยการใช้ตารางค้นหา ซึ่งตารางค้นหาสามารถลดการใช้ตรรกะได้หรือคำนวณที่ซับซ้อนลงได้มาก เมื่อเทียบกับการสร้างวงจรคำนวณที่ซับซ้อน

3.8 รายละเอียดการสร้างช่องสัญญาณสื่อสารแบบ Fading

การออกแบบสัญญาณการเฟดบนอุปกรณ์ FPGA จะออกแบบระบบตามสมการที่ (3.18) แสดงดังรูปที่ 3.39 ค่าสัมประสิทธิ์ช่องสัญญาณ $h(t)$ ที่ใช้งานจะอยู่ในรูปของค่าสัมบูรณ์ (absolute) ซึ่งสามารถคำนวณได้จากค่าสัมประสิทธิ์ช่องสัญญาณเชิงซ้อนแกนจริงและแกนจินตภาพ ดังสมการต่อไปนี้

$$|h(t)| = \sqrt{h_I^2(t) + h_Q^2(t)} \qquad (3.18)$$

จากสมการดังกล่าวมี 2 ขั้นตอนหลักในการคำนวณค่าสัมบูรณ์ค่าสัมประสิทธิ์ช่องสัญญาณเชิงซ้อน ประกอบด้วยการยกกำลังสองและการคำนวณรากที่สอง



รูปที่ 3.39 แผนภาพการออกแบบสัญญาณการเฟดบนอุปกรณ์ FPGA

การยกกำลังสอง

สำหรับการยกกำลังสองบนอุปกรณ์ FPGA สามารถทำได้โดยการคูณค่าที่ต้องการยกกำลังด้วยค่านั้นเอง โดยผลลัพธ์การคูณบนอุปกรณ์ FPGA จะสร้างผลลัพธ์ที่มีความยาวเท่ากับความยาวตัวคูณพจน์ที่หนึ่งและพจน์ที่สองบวกกัน

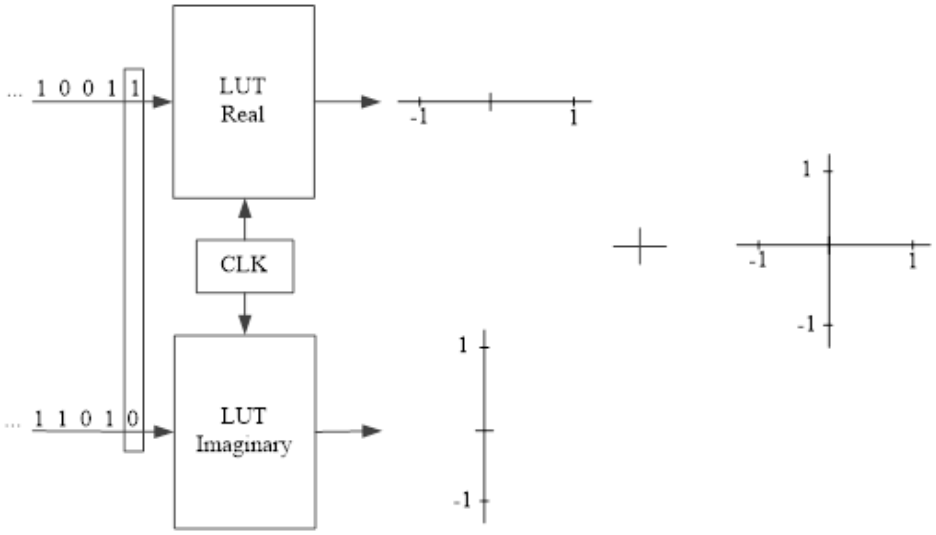
การคำนวณรากที่สอง

เช่นเดียวกับการคำนวณรากที่สองในระบบการออกแบบสัญญาณรบกวนเกาส์เซียนขาวแบบบวก การคำนวณรากที่สองในระบบนี้ใช้งานวงจร CORDIC ในโหมดเวกเตอร์ไฮเปอร์โบลิก (Hyperbolic vectoring) โดยมีข้อแตกต่างกับวงจรในการออกแบบสัญญาณรบกวนเกาส์เซียนขาวแบบบวก โดยที่ใช้งานอินพุตและเอาต์พุตเป็นค่า fixed point ที่ความละเอียด fi(16, 11) ทั้งคู่

3.9 รายละเอียดการสร้างการมอดูเลชันมาตรฐาน 5G

ระบบการพัฒนาการมอดูเลตมาตรฐาน 5G บนอุปกรณ์ FPGA เป็นการสร้างวงจรการประมวลผลสัญญาณดิจิทัล (Digital signal processing: DSP) เป็นหลัก ทำให้การมอดูเลตสัญญาณตามมาตรฐาน 5G สามารถทำได้โดยการบวก ลบ และ คูณ ทั้งนี้ไม่นิยมการหารเนื่องจากวงจรจะมีความซับซ้อนสูง จึงมักใช้วิธีการอื่น เช่น การคำนวณค่าผลหารและบันทึกผลลัพธ์ไว้ใน ROM หรือ RAM ของอุปกรณ์ FPGA

การมอดูเลชันแบบ BPSK (Binary phase shift keying) การมอดูเลชันแบบ $\pi/2$-BPSK การมอดูเลชันแบบ QPSK (Quadrature phase shift keying) การมอดูเลชันแบบ 16QAM (Quadrature amplitude modulation) การมอดูเลชันแบบ 64QAM การมอดูเลชันแบบ 256QAM และการมอดูเลชันแบบ 1024QAM มีลักษณะคล้ายกัน โดยสามารถพิจารณาเป็นการมอดูเลชันแบบ PAM (Pulse amplitude modulation) ที่มีค่าบนแกนจริง (In-phase) และแกนจินตภาพ (Quadrature-phase) แตกต่างกันที่จำนวนระดับของ PAM และค่านอมัลไลซ์ ดังนั้น การสร้างวงจรมอดูเลตมาตรฐาน 5G จะเป็นเพียงการสร้าง look-up table หรือ LUT เพื่อเก็บค่าการคูณและการหารของค่าในแกนจริงและแกนจินตภาพไว้ เนื่องจากค่าในแกนจริงและแกนจินตภาพเป็นค่าเดียวกัน ดังนั้นจริงสามารถลดตารางให้เหลือเพียงตารางของค่าในแกนจริงเท่านั้น



รูปที่ 3.40 ตัวอย่างการสร้าง QPSK ด้วย FPGA

ตัวอย่างการสร้างสัญญาณแบบ QPSK สำหรับเครื่องส่ง จะทำได้โดยการจำคู่บิตข้อมูลทีละ 2 บิต ต่อหนึ่งสัญลักษณ์ ทำให้สัญญาณที่เป็นไปได้ทั้งหมดเท่า 4 สัญญาณดังรูปที่ 3.40 ทั้งนี้ บิตข้อมูล 2 บิต ที่ต้องการจับคู่ถูกใช้เป็นอินพุตของ LUT ส่วนสัญญาณเอาต์พุตประกอบไปด้วยสัญญาณ I (In-phase) (หรือค่าในแกนจริง) และ Q (Quadrature-phase) (หรือค่าในแกนจินตภาพ) โดยสัญญาณมอดูเลชันจะถูกเลื่อนออกมาทุกสัญญาณนาฬิกา ตารางที่ 3.3 – 3.5 แสดงอินพุตและเอาต์พุตของ LUT สำหรับการมอดูเลชันแบบ $\pi/2$-BPSK การมอดูเลชันแบบ BPSK และการมอดูเลชันแบบ QPSK

ตารางที่ 3.3 ตารางค้นหาการมอดูเลชันสำหรับการมอดูเลชันแบบ $\pi/2$-BPSK

LUT		
Input	‘1’	‘0’
Output	0.7071	-0.7071

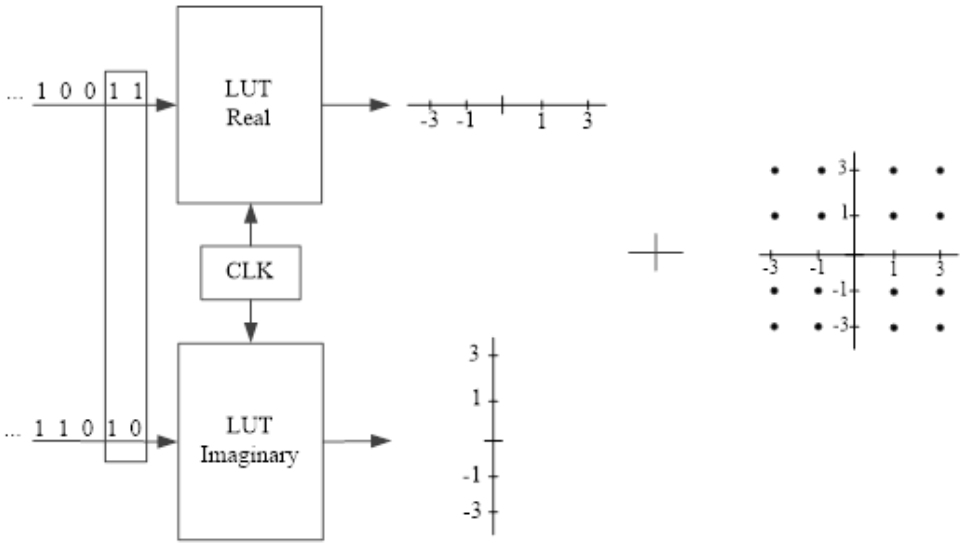
ตารางที่ 3.4 ตารางค้นหาการมอดูเลชันสำหรับการมอดูเลชันแบบ BPSK

LUT		
Input	‘1’	‘0’
Output	-0.7071	0.7071

ตารางที่ 3.5 ตารางค้นหาการมอดูเลชันสำหรับการมอดูเลชันแบบ QPSK (2PAM)

LUT		
Input	‘1’	‘0’
Output	-0.7071	0.7071

ตัวอย่างการสร้างสัญญาณแบบ 16QAM สำหรับเครื่องส่ง จะทำได้โดยการจำคู่บิตข้อมูลทีละ 4 บิต โดย 2 บิตเป็นค่าในแกนจริง ในรูปตัวอย่างคือ ‘11’ และ 2 บิตเป็นค่าในแกนจินตภาพในรูปตัวอย่างคือ ‘10’ ดังนั้นจะได้สัญญาณ 16QAM ที่ดึงมาจากตารางที่ 3.6 คือ -0.9487 - 0.3162j โดยตารางที่ 3.6 – 3.9 แสดงอินพุตและเอาต์พุตของ LUT สำหรับการมอดูเลชันแบบ 16QAM การมอดูเลชันแบบ 64QAM การมอดูเลชันแบบ 256QAM และการมอดูเลชันแบบ 1024QAM



รูปที่ 3.41 ตัวอย่างการสร้าง 16QAM ด้วย FPGA

ตารางที่ 3.6 ตารางค้นหาการมอดูเลชันสำหรับการมอดูเลชันแบบ 16QAM (4PAM)

LUT				
Input	‘11’	‘10’	‘00’	‘01’
Output	-0.9487	-0.3162	0.3162	0.9487

ตารางที่ 3.7 ตารางค้นหาการมอดูเลชันสำหรับการมอดูเลชันแบบ 64QAM (8PAM)

LUT				
Input	‘111’	‘110’	‘100’	‘101’
Output	-1.0801	-0.7715	-0.4629	-0.1543
Input	‘001’	‘000’	‘010’	‘011’
Output	0.1543	0.46291	0.77152	1.08012

ตารางที่ 3.8 ตารางค้นหาการมอดูเลชันสำหรับการมอดูเลชันแบบ 256QAM (16PAM)

LUT				
Input	‘1111’	‘1110’	‘1100’	‘1101’
Output	-1.1504	-0.9971	-0.8437	-0.6903
Input	‘1001’	‘1000’	‘1010’	‘1011’
Output	-0.5369	-0.3835	-0.2301	-0.0767
Input	‘0011’	‘0010’	‘0000’	‘0001’
Output	0.0767	0.23009	0.38348	0.53688
Input	‘0101’	‘0100’	‘0110’	‘0111’
Output	0.69027	0.84366	0.99705	1.15045

ตารางที่ 3.9 ตารางค้นหาการมอดูเลชันสำหรับการมอดูเลชันแบบ 1024QAM (32PAM)

LUT				
Input	‘11111’	‘11110’	‘11100’	‘11101’
Output	-1.1871	-1.1105	-1.0339	-0.9573
Input	‘11001’	‘11000’	‘11010’	‘11011’
Output	-0.8807	-0.8041	-0.7275	-0.651
Input	‘10011’	‘10010’	‘10000’	‘10001’
Output	-0.5744	-0.4978	-0.4212	-0.3446
Input	‘10101’	‘10100’	‘10110’	‘10111’
Output	-0.268	-0.1915	-0.1149	-0.0383
Symbol	‘00111’	‘00110’	‘00100’	‘00101’
value	0.03829	0.11488	0.19146	0.26804

Input	‘00001’	‘00000’	‘00010’	‘00011’
Output	0.34463	0.42121	0.4978	0.57438
Input	‘01011’	‘01010’	‘01000’	‘01001’
Output	0.65096	0.72755	0.80413	0.88072
Input	‘01101’	‘01100’	‘01110’	‘01111’
Output	0.9573	1.03388	1.11047	1.18705

3.10 รายละเอียดการสร้างการดีมอดูเลชันมาตรฐาน 5G

การออกแบบการดีมอดูเลตมาตรฐาน 5G บนอุปกรณ์ FPGA ไม่มีข้อกำหนดไว้ ให้ขึ้นอยู่กับผู้ผลิตแต่ละราย และไม่มีข้อกำหนดอัลกอริทึมไว้ด้วยเช่นกัน แต่มีข้อกำหนดทางเทคนิคไว้ตายตัว ไม่สามารถเปลี่ยนแปลงได้ เช่น การถอดรหัส LDPC และ Polar code ต้องมีขนาดของ บล็อก (block) เท่าไร และวางไว้ตรงไหนตามข้อกำหนด ตามที่ได้กล่าวมาข้างต้น ไม่สามารถเปลี่ยนแปลงได้ อย่างไรก็ดี สำหรับการคำนวณด้วย FPGA ควรพิจารณา Logic blocks และ DSP blocks เพียงพอต่อการประมวลดังกล่าวหรือไม่ และอัลกอริทึมสำหรับการดีมอดูเลชันสามารถคำนวณได้หรือไม่ การออกแบบจะเริ่มจากจากสมการที่ (2.22) ถ้ากำหนดให้ช่องสัญญาณเป็นเป็นแบบ AWGN (Additive white Gaussian noise) จะได้

$$
\begin{aligned}
L_{b_i}(y) &= \log\left(\frac{\max\limits_{s\in\{s_{b_1\ldots b_k}|b_i=1\}}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\{-\frac{(y-s)^2}{2\sigma^2}\}}{\max\limits_{s\in\{s_{b_1\ldots b_k}|b_i=0\}}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\{-\frac{(y-s)^2}{2\sigma^2}\}}\right)\\
&= \log\left(\frac{\max\limits_{s\in\{s_{b_1\ldots b_k}|b_i=1\}}\exp\{-\frac{(y-s)^2}{2\sigma^2}\}}{\max\limits_{s\in\{s_{b_1\ldots b_k}|b_i=0\}}\exp\{-\frac{(y-s)^2}{2\sigma^2}\}}\right)\\
&= \log\left(\max_{s\in\{s_{b_1\ldots b_k}|b_i=1\}}\exp\{-\frac{(y-s)^2}{2\sigma^2}\}\right)\\
&\quad -\log\left(\max_{s\in\{s_{b_1\ldots b_k}|b_i=0\}}\exp\{-\frac{(y-s)^2}{2\sigma^2}\}\right)\\
&= \max_{s\in\{s_{b_1\ldots b_k}|b_i=1\}}\log\left(\exp\{-\frac{(y-s)^2}{2\sigma^2}\}\right)\\
&\quad -\max_{s\in\{s_{b_1\ldots b_k}|b_i=0\}}\log\left(\exp\{-\frac{(y-s)^2}{2\sigma^2}\}\right)\\
&= \max_{s\in\{s_{b_1\ldots b_k}|b_i=1\}}\left(-\frac{(y-s)^2}{2\sigma^2}\right)-\max_{s\in\{s_{b_1\ldots b_k}|b_i=0\}}\left(-\frac{(y-s)^2}{2\sigma^2}\right)
\end{aligned}
$$

$$= \frac{1}{2\sigma^2}\left(\max_{s\in\{s_{b_1\ldots b_k}|b_i=1\}} \left(-(y-s)^2\right) - \max_{s\in\{s_{b_1\ldots b_k}|b_i=0\}} \left(-(y-s)^2\right) \right)$$

$$= \frac{1}{2\sigma^2}\left(- \min_{s\in\{s_{b_1\ldots b_k}|b_i=1\}} \left((y-s)^2\right) + \min_{s\in\{s_{b_1\ldots b_k}|b_i=0\}} \left((y-s)^2\right) \right)$$

$$= \frac{1}{2\sigma^2}\left(\min_{s\in\{s_{b_1\ldots b_k}|b_i=0\}} \left((y-s)^2\right) - \min_{s\in\{s_{b_1\ldots b_k}|b_i=1\}} \left((y-s)^2\right) \right) \quad (3.19)$$

จากสมการที่ (3.19) จะต้องมีการออปติไมซ์ (Optimization) หาค่าต่ำสุดของ $(y-s)^2$ ดังนั้นจะกำหนดให้ μ_0 และ μ_1 เป็นค่าต่ำสุดของ $\min_{s\in\{s_{b_1\ldots b_k}|b_i=0\}} \left((y-s)^2\right)$ และ $\min_{s\in\{s_{b_1\ldots b_k}|b_i=1\}} \left((y-s)^2\right)$ สามารถเขียนได้เป็น

$$\mu_0 = \min_{s\in\{s_{b_1\ldots b_k}|b_i=0\}} \left((y-s)^2\right) \quad (3.20)$$

$$\mu_1 = \min_{s\in\{s_{b_1\ldots b_k}|b_i=1\}} \left((y-s)^2\right) \quad (3.21)$$

ดังนั้นจากสมการที่ (3.19) จะสามารถเขียนใหม่ได้เป็น

$$L_{b_i}(y) = \frac{1}{2\sigma^2}\left((y-\mu_0)^2 - (y-\mu_1)^2\right)$$

$$= \frac{1}{2\sigma^2}\left(y^2 - 2y\mu_0 + (\mu_0)^2 - (y^2 - 2y\mu_1 + (\mu_1)^2)\right)$$

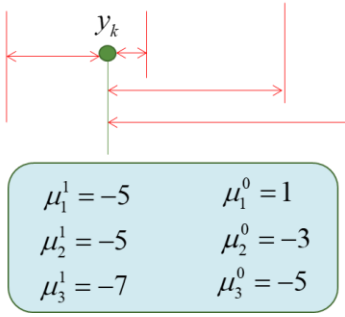
$$= \frac{1}{2\sigma^2}\left(2y(\mu_1-\mu_0) + (\mu_0)^2 - (\mu_1)^2\right) \quad (3.22)$$

ดังนั้น การประมาณค่า LLR ด้วยวิธีการดีมอดูเลชันค่า LLR สูงสุดด้วยสมการที่ (3.22) ใช้สำหรับค่าจำนวนจริงและจำนวนจินตภาพของการมอดูเลชันแบบต่าง ๆ ในมาตรฐาน 5G ที่ส่งผ่านช่องสัญญาณ AWGN เท่านั้น

จากสมการที่ (3.22) จะเห็นได้ว่ามีค่า μ_0 และ μ_1 ที่จะต้องออปติไมซ์หาค่าต่ำสุดจากสมการที่ (3.20) และ (3.21) เราจะหาค่าดังกล่าวเป็นตารางเก็บไว้เพื่อลดความซับซ้อนและเวลาในการคำนวณค่า วิธีการออกแบบตารางสามารถทำได้โดยการแบ่งช่วงค่า y ตัวอย่างเช่น การมอดูเลชันแบบ 64QAM จะพิจารณาเป็นการมอดูเลชันแบบ 8PAM ของแกนจำนวนจริง และการมอดูเลชันแบบ 8PAM ของแกนจำนวนจินตภาพ ในที่นี้เราจะยกตัวอย่างเพียง 1 แกน การมอดูเลชันแบบ 8PAM จะมีการแบ่งค่า y เป็น 8 ช่วง แต่ละช่วงประกอบด้วย 3 บิต และมีค่าเฉลี่ยดังแสดงในรูปที่ 3.42 สมมุติให้ค่า y ตกในช่วงของสัญลักษณ์ 110 ค่าเฉลี่ยของสัญลักษณ์ที่มีบิตที่ 1 เป็น 1 μ_1^1 ที่ใกล้ค่า y ที่สุดคือ -5 หรือก็ค่าค่าเฉลี่ยของช่วงที่ค่า y อยู่ ค่าเฉลี่ยของสัญลักษณ์ที่มีบิตที่ 1 เป็น 0 μ_1^0 ที่ใกล้ค่า y ที่สุดคือ 1 ค่าเฉลี่ยของสัญลักษณ์ที่มีบิตที่ 1 เป็น 0 μ_1^0 ที่ใกล้ค่า y ที่สุดคือ 1 ค่าเฉลี่ยของสัญลักษณ์ที่มีบิตที่ 2 เป็น 1 μ_2^1 ที่ใกล้ค่า y ที่สุดคือ -5 ค่าเฉลี่ยของสัญลักษณ์ที่มีบิตที่

2 เป็น 0 μ_2^0 ที่ใกล้ค่า y ที่สุดคือ -3 ค่าเฉลี่ยของสัญลักษณ์ที่มีบิตที่ 3 เป็น 1 μ_3^1 ที่ใกล้ค่า y ที่สุดคือ -7 และค่าเฉลี่ยของสัญลักษณ์ที่มีบิตที่ 3 เป็น 0 μ_3^0 ที่ใกล้ค่า y ที่สุดคือ -5

Symbols	111	110	100	101	001	000	010
means	−7	−5	−3	−1	1	3	5



รูปที่ 3.42 ตัวอย่างการหาค่าในตารางสำหรับค่า μ_0 และ μ_1 ที่มีการมอดูเลชันแบบ 8PAM

จากตัวอย่างวิธีการออกแบบตารางด้านบนเราจะหาค่าในตารางทั้งหมดโดยสมมุติให้ค่า y อยู่ในแต่ละช่วงแล้วหาค่าจนครบทุกช่วง ดังนั้นจะได้ตารางดังตารางที่ 3.18 และตารางที่ 3.19 จากสมการที่ (3.22) เมื่อแทนค่า μ_0 และ μ_1 ด้วยตารางจะได้ดังนี้

$$L(i) = \frac{1}{2\sigma^2}\Big(2y(n_m)\big(LUT_1(i,F(y)) - LUT_0(i,F(y))\big) + \big(LUT_0(i,F(y))\big)^2 - \big(LUT_0(i,F(y))\big)^2\Big) \tag{3.23}$$

เมื่อ $n_m \in \{\sqrt{2}, \sqrt{10}, \sqrt{42}, \sqrt{170}, \sqrt{682}\}$ i คือลำดับบิตในสัญลักษณ์ และ $F(y)$ คือฟังก์ชันในการคำนวณอินเด็กซ์ของตารางจากค่า y คำนวณได้จาก

$$F(y) = saturate\left(round(y\frac{n_m}{2} - 0.5)\right) \tag{3.24}$$

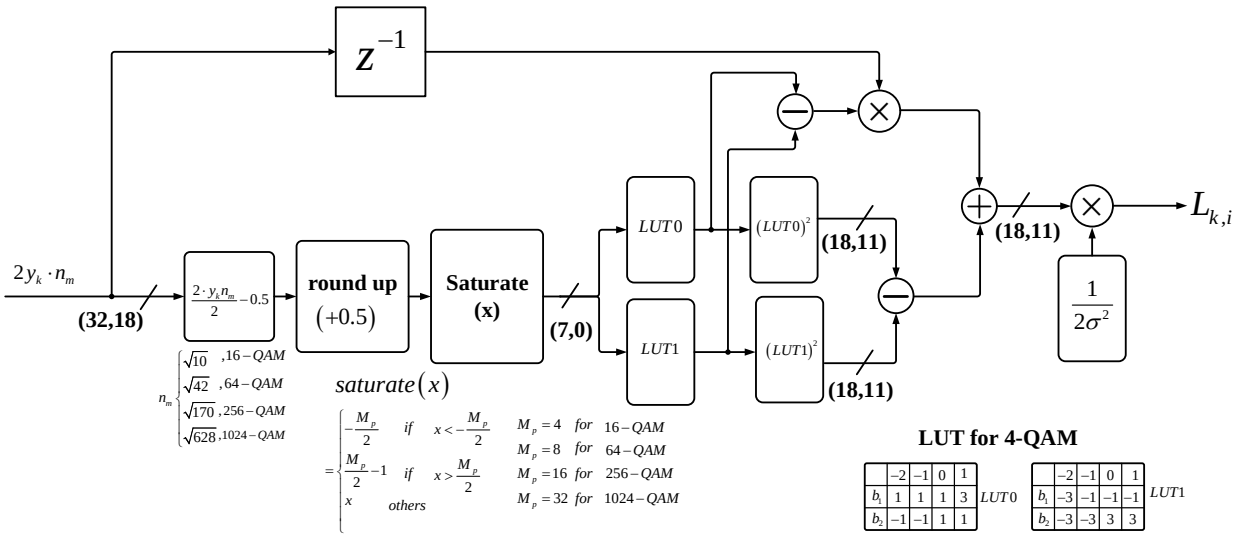
เมื่อ

$$saturate(x) = \begin{cases} -M/2, & if\ x < -M/2 \\ M/2 - 1, & if\ x > M/2 - 1 \\ x, & otherwise. \end{cases} \tag{3.25}$$

และ M คือ M-QAM

การคำนวณด้วย FPGA สามารถออกแบบได้ดังรูปที่ 3.43 เริ่มต้นทำการปัดเศษขึ้น (Round up) ตั้งขอบเขต (Saturate) จากนั้น นำสัญญาณที่ได้เพื่อหาค่าตาม LUT โดยที่ตัวอย่าง

LUT สำหรับ QPSK โดยการคำนวณกำหนดให้ขนาดบิตสุดท้ายที่ใช้คือ 18 บิต ประกอบไปด้วยส่วนของทศนิยม 11 บิต และส่วนของจำนวนเต็ม 7 บิต



รูปที่ 3.43 อัลกอริทึมการดีมอดูเลชันสัญญาณตามมาตรฐาน 5G บนอุปกรณ์ FPGA โดยมีตัวอย่าง LUT0 และ LUT1 ประกอบ

สำหรับการมอดูเลชันในมาตรฐาน 5G กำหนดให้ใช้งานการมอดูเลชัน 7 รูปแบบได้แก่การมอดูเลชันแบบ $\pi/2$-BPSK การมอดูเลชันแบบ BPSK การมอดูเลชันแบบ QPSK การมอดูเลชันแบบ 16QAM การมอดูเลชันแบบ 64QAM การมอดูเลชันแบบ 256QAM และการมอดูเลชันแบบ 1024QAM ซึ่งมีตาราง LUT_0 และ LUT_1 ดังนี้

1) การมอดูเลชันแบบ $\pi/2$-BPSK

ตารางที่ 3.10 ตารางค่าเฉลี่ยของบิต 0 ที่ใกล้ y ที่สุดสำหรับ $\pi/2$-BPSK

LUT_0			
Input		$-\infty < y \le 0$	$0 < y \le \infty$
Output	บิตที่ 1	-1	1

ตารางที่ 3.11 ตารางค่าเฉลี่ยของบิต 1 ที่ใกล้ y ที่สุดสำหรับ $\pi/2$-BPSK

LUT_1			
Input		$-\infty < y \le 0$	$0 < y \le \infty$
Output	บิตที่ 1	-1	1

2) การมอดูเลชันแบบ BPSK

ตารางที่ 3.12 ตารางค่าเฉลี่ยของบิต 0 ที่ใกล้ y ที่สุดสำหรับ BPSK

LUT_0			
Input		$-\infty < y \le 0$	$0 < y \le \infty$
Output	บิตที่ 1	-1	1

ตารางที่ 3.13 ตารางค่าเฉลี่ยของบิต 1 ที่ใกล้ y ที่สุดสำหรับ BPSK

LUT_1			
Input		$-\infty < y \le 0$	$0 < y \le \infty$
Output	บิตที่ 1	-1	1

3) การมอดูเลชันแบบ QPSK

ตารางที่ 3.14 ตารางค่าเฉลี่ยของบิต 0 ที่ใกล้ y ที่สุดสำหรับ QPSK (2PAM)

LUT_0			
Input		$-\infty < y \le 0$	$0 < y \le \infty$
Output	บิตที่ 1	-1	1

ตารางที่ 3.15 ตารางค่าเฉลี่ยของบิต 1 ที่ใกล้ y ที่สุดสำหรับ QPSK (2PAM)

LUT_1			
Input		$-\infty < y \le 0$	$0 < y \le \infty$
Output	บิตที่ 1	-1	1

4) การมอดูเลชันแบบ 16QAM

ตารางที่ 3.16 ตารางค่าเฉลี่ยของบิต 0 ที่ใกล้ y ที่สุดสำหรับ 16QAM (4PAM)

LUT_0					
Input		$-\infty < y \le -2$	$-2 < y \le 0$	$0 < y \le 2$	$2 < y \le \infty$
Output	บิตที่ 1	1	1	1	3
	บิตที่ 2	-1	-1	1	1

ตารางที่ 3.17 ตารางค่าเฉลี่ยของบิต 1 ที่ใกล้ y ที่สุดสำหรับ 16QAM (4PAM)

LUT_1					
Input		$-\infty < y \le -2$	$-2 < y \le 0$	$0 < y \le 2$	$2 < y \le \infty$
Output	บิตที่ 1	-3	-1	-1	-1
	บิตที่ 2	-3	-3	3	3

5) การมอดูเลชันแบบ 64QAM

ตารางที่ 3.18 ตารางค่าเฉลี่ยของบิต 0 ที่ใกล้ y ที่สุดสำหรับ 64QAM (8PAM)

LUT_1					
Input		$-\infty < y \le -6$	$-6 < y \le -4$	$-4 < y \le -2$	$-2 < y \le 0$
Output	บิตที่ 1	1	1	1	1
	บิตที่ 2	-3	-3	-3	-1
	บิตที่ 3	-5	-5	-3	-3
Input		$0 < y \le 2$	$2 < y \le 4$	$4 < y \le 6$	$6 < y \le \infty$
Output	บิตที่ 1	1	3	5	7
	บิตที่ 2	1	3	3	3
	บิตที่ 3	3	3	5	5

ตารางที่ 3.19 ตารางค่าเฉลี่ยของบิต 1 ที่ใกล้ y ที่สุดสำหรับ 64QAM (8PAM)

LUT_1					
Input		$-\infty < y \le -6$	$-6 < y \le -4$	$-4 < y \le -2$	$-2 < y \le 0$
Output	บิตที่ 1	-7	-5	-3	-1
	บิตที่ 2	-7	-5	-5	-5
	บิตที่ 3	-7	-7	-1	-1
Input		$0 < y \le 2$	$2 < y \le 4$	$4 < y \le 6$	$6 < y \le \infty$
Output	บิตที่ 1	-1	-1	-1	-1
	บิตที่ 2	5	5	5	7
	บิตที่ 3	1	1	7	7

6) การมอดูเลชันแบบ 256QAM

ตารางที่ 3.20 ตารางค่าเฉลี่ยของบิต 0 ที่ใกล้ y ที่สุดสำหรับ 256QAM (16PAM)

LUT_0					
Input		$-\infty < y \le -14$	$-14 < y \le -12$	$-12 < y \le -10$	$-10 < y \le -8$
Output	1	1	1	1	-9
	-7	-7	-7	-7	-9
	-11	-11	-11	-9	-13
	-13	-13	-11	-11	-9
Input		$-8 < y \le -6$	$-6 < y \le -4$	$-4 < y \le -2$	$-2 < y \le 0$
Output	1	1	1	1	-1
	-7	-5	-3	-1	-9
	-7	-5	-5	-5	-1
	-5	-5	-3	-3	-1
Input		$0 < y \le 2$	$2 < y \le 4$	$4 < y \le 6$	$6 < y \le 8$
Output	1	3	5	7	-1
	1	3	5	7	9
	5	5	5	7	3
	3	3	5	5	7
Input		$8 < y \le 10$	$10 < y \le 12$	$12 < y \le 14$	$14 < y \le \infty$
Output	9	11	13	15	-1
	7	7	7	7	15
	9	11	11	11	15
	11	11	13	13	15

ตารางที่ 3.21 ตารางค่าเฉลี่ยของบิต 1 ที่ใกล้ y ที่สุดสำหรับ 256QAM (16PAM)

LUT_0					
Input		$-\infty < y \le -14$	$-14 < y \le -12$	$-12 < y \le -10$	$-10 < y \le -8$
Output	บิตที่ 1	-15	-13	-11	-9
	บิตที่ 2	-15	-13	-11	-9
	บิตที่ 3	-15	-13	-13	-13
	บิตที่ 4	-15	-15	-9	-9
Input		$-8 < y \le -6$	$-6 < y \le -4$	$-4 < y \le -2$	$-2 < y \le 0$
Output	บิตที่ 1	-7	-5	-3	-1
	บิตที่ 2	-9	-9	-9	-9
	บิตที่ 3	-3	-3	-3	-1
	บิตที่ 4	-7	-7	-1	-1
Input		$0 < y \le 2$	$2 < y \le 4$	$4 < y \le 6$	$6 < y \le 8$
Output	บิตที่ 1	-1	-1	-1	-1
	บิตที่ 2	9	9	9	9
	บิตที่ 3	1	3	3	3
	บิตที่ 4	1	1	7	7
Input		$8 < y \le 10$	$10 < y \le 12$	$12 < y \le 14$	$14 < y \le \infty$
Output	บิตที่ 1	-1	-1	-1	-1
	บิตที่ 2	9	11	13	15
	บิตที่ 3	13	13	13	15
	บิตที่ 4	9	9	15	15

7) การมอดูเลชันแบบ 1024QAM

ตารางที่ 3.22 ตารางค่าเฉลี่ยของบิต 0 ที่ใกล้ y ที่สุดสำหรับ 1024QAM (32PAM)

LUT_0					
Input		$-\infty < y \le -30$	$-30 < y \le -28$	$-28 < y \le -26$	$-26 < y \le -24$
Output	บิตที่ 1	1	1	1	1
	บิตที่ 2	-15	-15	-15	-15
	บิตที่ 3	-23	-23	-23	-23

	บิตที่ 4	-27	-27	-27	-25
	บิตที่ 5	-29	-29	-27	-27
Input		$-24 < y \leq -22$	$-22 < y \leq -20$	$-20 < y \leq -18$	$-18 < y \leq -16$
Output	บิตที่ 1	1	1	1	1
	บิตที่ 2	-15	-15	-15	-15
	บิตที่ 3	-23	-21	-19	-17
	บิตที่ 4	-23	-21	-21	-21
	บิตที่ 5	-21	-21	-19	-19
Input		$-16 < y \leq -14$	$-14 < y \leq -12$	$-12 < y \leq -10$	$-10 < y \leq -8$
Output	บิตที่ 1	1	1	1	1
	บิตที่ 2	-15	-13	-11	-9
	บิตที่ 3	-15	-13	-11	-9
	บิตที่ 4	-11	-11	-11	-9
	บิตที่ 5	-13	-13	-11	-11

Input		$-8 < y \leq -6$	$-6 < y \leq -4$	$-4 < y \leq -2$	$-2 < y \leq 0$
Output	บิตที่ 1	1	1	1	1
	บิตที่ 2	-7	-5	-3	-1
	บิตที่ 3	-9	-9	-9	-9
	บิตที่ 4	-7	-5	-5	-5
	บิตที่ 5	-5	-5	-3	-3
Input		$0 < y \leq 2$	$2 < y \leq 4$	$4 < y \leq 6$	$6 < y \leq 8$
Output	บิตที่ 1	1	3	5	7
	บิตที่ 2	1	3	5	7
	บิตที่ 3	9	9	9	9
	บิตที่ 4	5	5	5	7
	บิตที่ 5	3	3	5	5
Input		$8 < y \leq 10$	$10 < y \leq 12$	$12 < y \leq 14$	$14 < y \leq 16$
Output	บิตที่ 1	9	11	13	15

	บิตที่ 2	9	11	13	15
	บิตที่ 3	9	11	13	15
	บิตที่ 4	9	11	11	11
	บิตที่ 5	11	11	13	13
Input		$16 < y \le 18$	$18 < y \le 20$	$20 < y \le 22$	$22 < y \le 24$
Output	บิตที่ 1	17	19	21	23
	บิตที่ 2	15	15	15	15
	บิตที่ 3	17	19	21	23
	บิตที่ 4	21	21	21	23
	บิตที่ 5	19	19	21	21
Input		$24 < y \le 26$	$26 < y \le 28$	$28 < y \le 30$	$30 < y \le \infty$
Output	บิตที่ 1	25	27	29	31
	บิตที่ 2	15	15	15	15
	บิตที่ 3	23	23	23	23
	บิตที่ 4	25	27	27	27
	บิตที่ 5	27	27	29	29

ตารางที่ 3.23 ตารางค่าเฉลี่ยของบิต 1 ที่ใกล้ y ที่สุดสำหรับ 1024QAM (32PAM)

LUT_1					
Input		$-\infty < y \le -30$	$-30 < y \le -28$	$-28 < y \le -26$	$-26 < y \le -24$
Output	บิตที่ 1	-31	-29	-27	-25
	บิตที่ 2	-31	-29	-27	-25
	บิตที่ 3	-31	-29	-27	-25
	บิตที่ 4	-31	-29	-29	-29
	บิตที่ 5	-31	-31	-25	-25
Input		$-24 < y \le -22$	$-22 < y \le -20$	$-20 < y \le -18$	$-18 < y \le -16$
Output	บิตที่ 1	-23	-21	-19	-17
	บิตที่ 2	-23	-21	-19	-17
	บิตที่ 3	-25	-25	-25	-25
	บิตที่ 4	-19	-19	-19	-17

	บิตที่ 5	-23	-23	-17	-17
Input		$-16 < y \leq -14$	$-14 < y \leq -12$	$-12 < y \leq -10$	$-10 < y \leq -8$
Output	บิตที่ 1	-15	-13	-11	-9
	บิตที่ 2	-17	-17	-17	-17
	บิตที่ 3	-7	-7	-7	-7
	บิตที่ 4	-15	-13	-13	-13
	บิตที่ 5	-15	-15	-9	-9
Input		$-8 < y \leq -6$	$-6 < y \leq -4$	$-4 < y \leq -2$	$-2 < y \leq 0$
Output	บิตที่ 1	-7	-5	-3	-1
	บิตที่ 2	-17	-17	-17	-17
	บิตที่ 3	-7	-5	-3	-1
	บิตที่ 4	-3	-3	-3	-1
	บิตที่ 5	-7	-7	-1	-1
Input		$0 < y \leq 2$	$2 < y \leq 4$	$4 < y \leq 6$	$6 < y \leq 8$
Output	บิตที่ 1	-1	-1	-1	-1
	บิตที่ 2	17	17	17	17
	บิตที่ 3	1	3	5	7
	บิตที่ 4	1	3	3	3
	บิตที่ 5	1	1	7	7

Input		$8 < y \leq 10$	$10 < y \leq 12$	$12 < y \leq 14$	$14 < y \leq 16$
Output	บิตที่ 1	-1	-1	-1	-1
	บิตที่ 2	17	17	17	17
	บิตที่ 3	7	7	7	7
	บิตที่ 4	13	13	13	15
	บิตที่ 5	9	9	15	15
Input		$16 < y \leq 18$	$18 < y \leq 20$	$20 < y \leq 22$	$22 < y \leq 24$
Output	บิตที่ 1	-1	-1	-1	-1
	บิตที่ 2	17	19	21	23

	บิตที่ 3	25	25	25	25
	บิตที่ 4	17	19	19	19
	บิตที่ 5	17	17	23	23
Input		$24 < y \leq 26$	$26 < y \leq 28$	$28 < y \leq 30$	$30 < y \leq \infty$
Output	บิตที่ 1	-1	-1	-1	-1
	บิตที่ 2	25	27	29	31
	บิตที่ 3	25	27	29	31
	บิตที่ 4	29	29	29	31
	บิตที่ 5	25	25	31	31

3.11 รายละเอียดการสร้างการเข้ารหัสช่องสัญญาณมาตรฐาน 5G

3.11.1 การออกแบบการเข้ารหัส LDPC บนอุปกรณ์ FPGA

การออกแบบการเข้ารหัส LDPC บนอุปกรณ์จะทำตามกระบวนการเข้ารหัสข้างต้น ซึ่งจะเห็นได้ว่ากระบวนการต่าง ๆ จะอาศัยการคูณและการบวกเป็นหลัก สำหรับบนอุปกรณ์ FPGA การบวกจะทำได้ง่ายและไม่เปลืองทรัพยากรด้วยการใช้ลอจิก XOR แทนการบวกแบบมอดูโล 2 ส่วนวงจรคูณจะเป็นส่วนที่ใช้ทรัพยากรค่อนข้างมากซึ่งสามารถหลีกเลี่ยงการใช้งานวงจรคูณได้จากคุณสมบัติของ QC-LDPC ซึ่งเป็นคุณลักษณะของเมทริกซ์ของมาตรฐาน 5G ดังนี้

$$H = \begin{bmatrix} Q(P_{1,1}) & Q(P_{1,2}) & \cdots & Q(P_{1,n_b}) \\ Q(P_{2,1}) & Q(P_{2,2}) & \cdots & Q(P_{2,n_b}) \\ \vdots & \vdots & \ddots & \vdots \\ Q(P_{m_b,1}) & Q(P_{m_b,2}) & \cdots & Q(P_{m_b,n_b}) \end{bmatrix} \tag{3.26}$$

โดยเมทริกซ์ $Q(P_{i,j})$ เป็นเมทริกซ์เอกลักษณ์ที่ถูกหมุนวนไปทางด้านขวาจำนวน $P_{i,j}$ ครั้งโดย $P_{i,j}$ ตัวอย่างของ $Q(P_{i,j})$ เป็นดังนี้

$$Q(0)=\begin{pmatrix}1&0&0&\cdots&0\\0&1&0&\cdots&0\\\vdots&\vdots&\ddots&\vdots&\\0&0&0&\cdots&0\\0&0&0&\cdots&1\end{pmatrix}$$
$$Q(1)=\begin{pmatrix}0&1&0&\cdots&0\\0&0&1&\cdots&0\\\vdots&\vdots&\ddots&\vdots&\\0&0&0&\cdots&1\\1&0&0&\cdots&0\end{pmatrix} \quad (3.27)$$

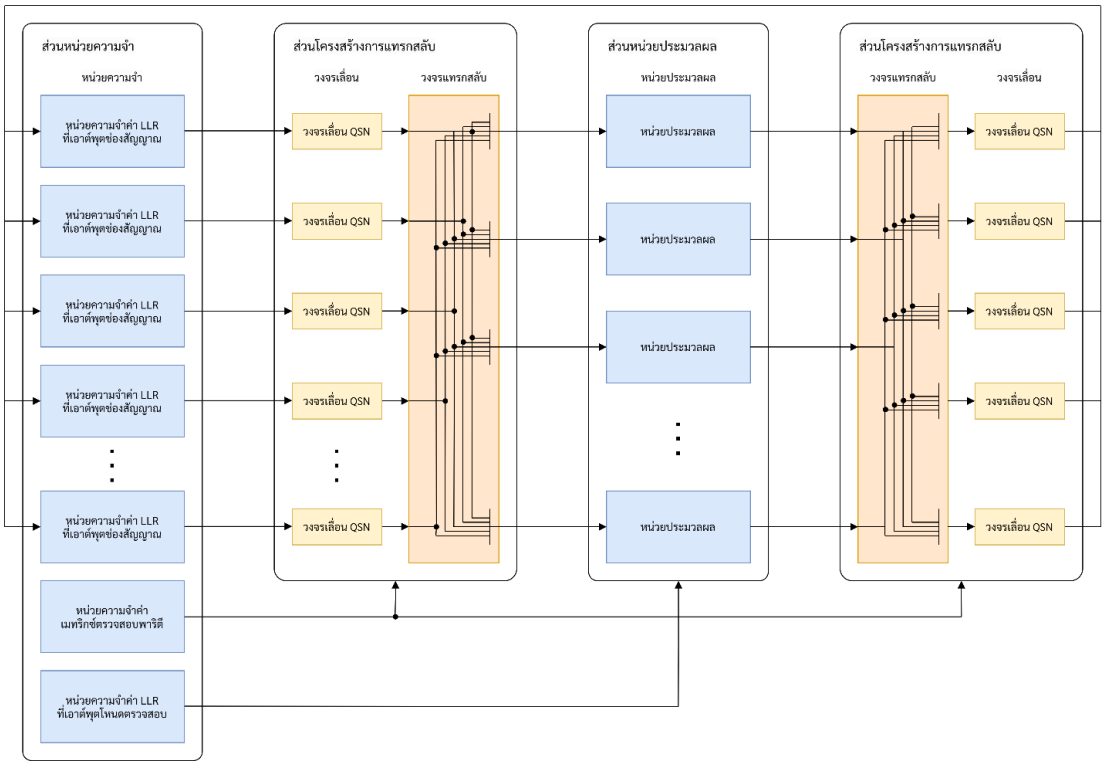
หากทำการคูณเมทริกซ์หลักใด ๆ เข้ากับเมทริกซ์ข้างต้น จะได้ผลลัพธ์ดังนี้

$$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}\times\begin{bmatrix}1\\1\\0\\0\end{bmatrix}=\begin{bmatrix}1\\1\\0\\0\end{bmatrix}$$
$$\begin{bmatrix}0&1&0&0\\0&0&1&0\\0&0&0&1\\1&0&0&0\end{bmatrix}\times\begin{bmatrix}1\\1\\0\\0\end{bmatrix}=\begin{bmatrix}1\\0\\0\\1\end{bmatrix} \quad (3.28)$$
$$\begin{bmatrix}0&0&1&0\\0&0&0&1\\1&0&0&0\\0&1&0&0\end{bmatrix}\times\begin{bmatrix}1\\1\\0\\0\end{bmatrix}=\begin{bmatrix}0\\0\\1\\1\end{bmatrix}$$

จะเห็นว่าผลลัพธ์ของการคูณเปรียบเสมือนเป็นเพียงการเลื่อนค่าของอินพุตซึ่งสามารถใช้วงจรเลื่อนค่าแทนวงจรคูณได้ วงจรเลื่อนค่าบนอุปกรณ์ FPGA มีจำนวนมากกว่าวงจรคูณหลายเท่า ดังนั้นการใช้งานในลักษณะนี้จึงเหมาะกว่าในแง่ของการใช้งานทรัพยากร

3.12 รายละเอียดการสร้างการถอดรหัสช่องสัญญาณมาตรฐาน 5G

สำหรับการถอดรหัส LDPC บนชิปเอฟพีจีเอจะอาศัยอัลกอริทึมการถอดรหัสความเชื่อมั่นแบบเลเยอร์ (Layered Belief Propagation) โดยที่โครงสร้างจะประกอบไปด้วยโหนด 2 ประเภท โหนดตัวแปร และโหนดตรวจสอบ โดยแต่ละโหนดจะมีการคำนวณตามสมการที่ 2.25 และ 2.26 ตามลำดับ โครงสร้างการถอดรหัสถูกออกแบบให้มีความคล้ายคลึงกับตัวถอดรหัสใน [14] โดยมีรูปโครงสร้างตามรูปที่ 3.44 โดยแบ่งโครงสร้างออกเป็นได้ทั้งหมด 3 ภาคส่วน ได้แก่ ส่วนหน่วยความจำ ส่วนโครงสร้างการแทรกสลับ และส่วนหน่วยประมวลผล



รูปที่ 3.44 โครงสร้างตัวถอดรหัส LDPC

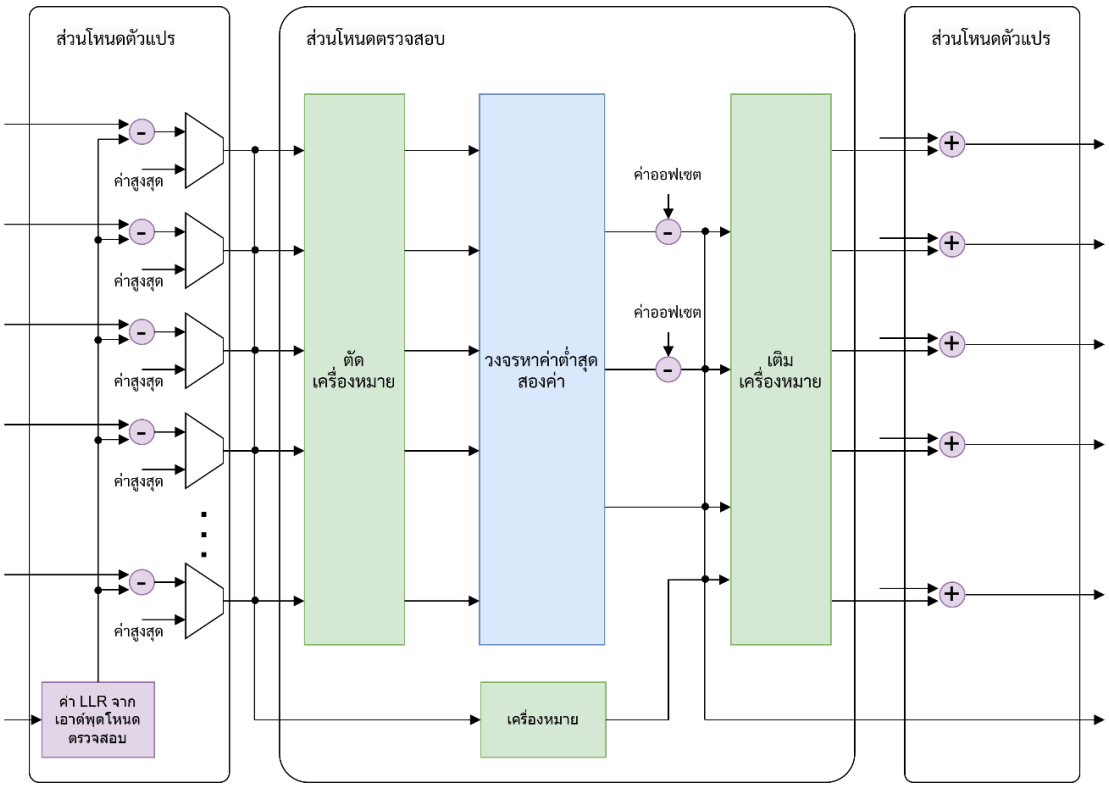
(1) ส่วนหน่วยความจำ จะประกอบไปด้วยหน่วยความจำ 3 ประเภท สำหรับการเก็บค่าเมทริกซ์ตรวจสอบพาริตี การเก็บค่า LLR ที่เอาต์พุตช่องสัญญาณ และการเก็บค่า LLR ที่เอาต์พุตโหนดตรวจสอบ เนื่องด้วยเป็นการถอดรหัสแบบเลเยอร์ ทำให้หน่วยความจำที่จำเป็นในการเก็บค่าข้อความภายในโครงสร้างตัวถอดรหัสมีเพียงค่า 3 ประเภทนี้ ดังนั้น หน่วยความจำดังกล่าวจะใช้งานบล็อกแรมภายในชิปเอฟพีจีเอทั้งหมด โดยจำนวนบล็อกแรมที่ใช้งานแก่หน่วยความจำประเภทต่าง ๆ แสดงดังตารางที่ 3.24

ตารางที่ 3.24 ทรัพยากรของหน่วยความจำ

ประเภทหน่วยความจำ	จำนวนบล็อกแรมที่ใช้งาน	ขนาดของบล็อกแรม
ค่าเมทริกซ์ตรวจสอบพาริตี	1	(จำนวนบล็อกแถว x (จำนวนบล็อกหลัก x (log_2(ขนาดการยกเมทริกซ์) + 1)))
ค่า LLR ที่เอาต์พุตช่องสัญญาณ	จำนวนบล็อกแถวของเมทริกซ์ตรวจสอบพาริตี	ขนาดการยกเมทริกซ์ x (จำนวนบิตต่อค่า LLR + 1)
ค่า LLR ที่เอาต์พุตโหนดตรวจสอบ	จำนวนหน่วยประมวลผล	(2 x (จำนวนบิตต่อค่า LLR – 1)) + log_2(จำนวนบล็อกแถว x ขนาดการยกเมทริกซ์) + (จำนวนบล็อกหลัก + ขนาดการยกเมทริกซ์)

(2) ส่วนโครงสร้างการแทรกสลับ เป็นโครงสร้างที่ขั้นอยู่ระหว่างส่วนหน่วยความจำและส่วนหน่วยประมวลผล ส่วนดังกล่าวทำหน้าที่จัดเรียงข้อมูลจากส่วนหน่วยความจำไปยังส่วนหน่วยประมวลผล โครงสร้างส่วนนี้จะประกอบไปด้วยวงจรย่อยอีก 2 ส่วน ได้แก่ วงจรเลื่อน QSN และวงจรแทรกสลับ เนื่องด้วยข้อจำกัดของบล็อกแรมที่สามารถเข้าถึงแรมได้เพียง 1 address ต่อ 1 สัญญาณนาฬิกา จึงทำให้หน่วยความจำ 1 address ต้องเก็บค่า LLR เท่ากับจำนวนหน่วยประมวลผล รวมถึงใช้จำนวนบล็อกแรมจำนวนบล็อกแถวของเมทริกซ์ตรวจสอบพาริตี ทำให้สามารถคำนวณโหนดตรวจสอบในลักษณะขนานกันเป็นจำนวนเท่ากับจำนวนหน่วยประมวลผลภายใน 1 สัญญาณนาฬิกาได้ ทั้งนี้ข้อมูลที่ถูกอ่านออกมาจากหน่วยความจำจะยังไม่ถูกจัดเรียง จึงต้องอาศัยส่วนโครงสร้างการแทรกสลับในการจัดเรียงข้อมูลเพื่อส่งเข้าสู่ส่วนหน่วยประมวลผล

(3) ส่วนหน่วยประมวลผล เป็นส่วนที่คำนวณค่าภายในการถอดรหัส ตัวถอดรหัสนี้ประกอบด้วยโหนดตรวจสอบและโหนดตัวแปร โหนดตรวจสอบใช้อัลกอริทึม Min-sum ภายการถอดรหัสแบบเลเยอร์ แสดงโครงสร้างหน่วยประมวลผล 1 ตัวได้ดังรูปที่ 3.45 โดยแสดงให้เห็นว่าภายการถอดรหัสแบบเลเยอร์ ค่า LLR ที่เอาต์พุตช่องสัญญาณก่อนที่จะคำนวณโหนดตรวจสอบจะต้องทำการลบกับค่า LLR ที่เอาต์พุตโหนดตรวจสอบเก่าก่อน ซึ่งจะได้ค่า LLR ที่เอาต์พุตโหนดตัวแปร จากนั้นจะทำการหาค่าที่ต่ำสุดสองค่าสำหรับการถอดรหัสด้วยอัลกอริทึม Min-sum ซึ่งจะได้ค่า LLR ที่เอาต์พุตโหนดตรวจสอบใหม่ สุดท้ายค่า LLR ที่เอาต์พุตโหนดตรวจสอบใหม่จะบวกกับค่า LLR ที่เอาต์พุตโหนดตัวแปร ได้เป็นค่า LLR ที่เอาต์พุตช่องสัญญาณใหม่



รูปที่ 3.45 ส่วนหน่วยประมวลผล 1 ตัว

บทที่ 4
ผลการวิจัย และการวิจารณ์ผล

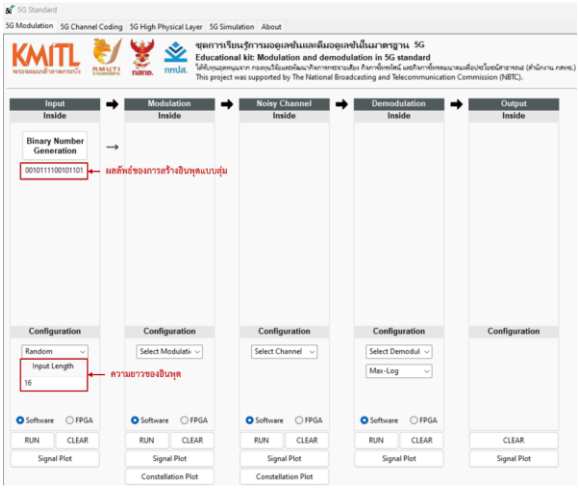
4.1 ผลการทดสอบการทำงานของซอฟต์แวร์ GUI

4.1.1 ผลการทดสอบซอฟต์แวร์ GUI เพื่อแสดงข้อมูลอินพุตแบบสุ่ม

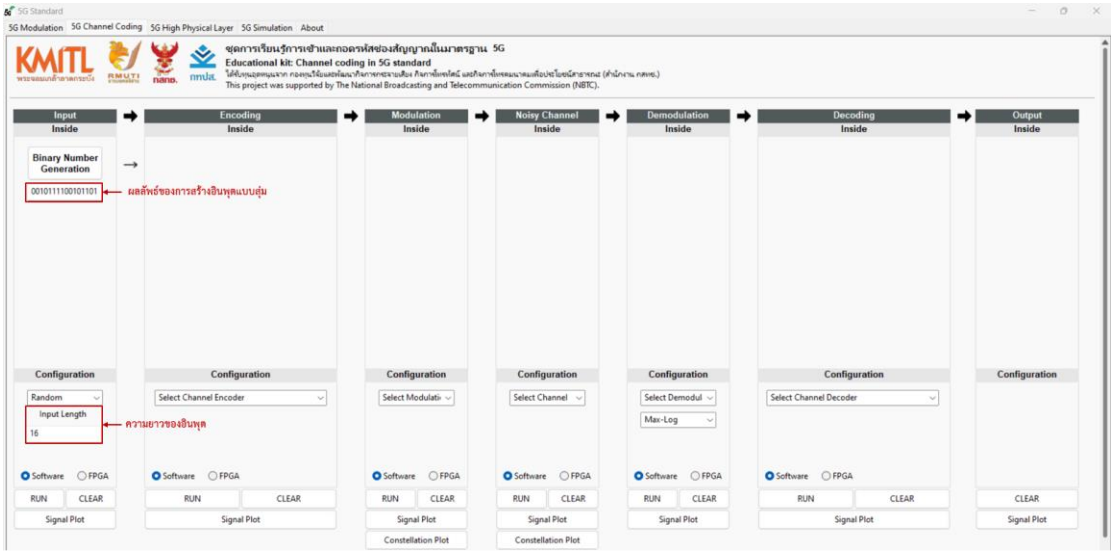
เมื่อผู้ใช้เปิดซอฟต์แวร์ GUI ที่หน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันในมาตรฐาน 5G หรือหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G หรือหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูง คอลัมน์ที่ 1 จะแสดงส่วนสำหรับสร้างข้อมูลอินพุต โดยผู้ใช้สามารถเลือกสร้างข้อมูลอินพุตแบบสุ่ม หรืออัปโหลดรูปภาพ หรือป้อนข้อมูลอินพุตด้วยตนเอง สำหรับกรณีการสร้างข้อมูลอินพุตแบบสุ่ม ผู้ใช้สามารถกำหนดความยาวอินพุต และเลือกสร้างข้อมูลอินพุตโดยใช้ซอฟต์แวร์หรืออุปกรณ์ FPGA

- กรณีใช้ซอฟต์แวร์สร้างข้อมูลอินพุตแบบสุ่ม

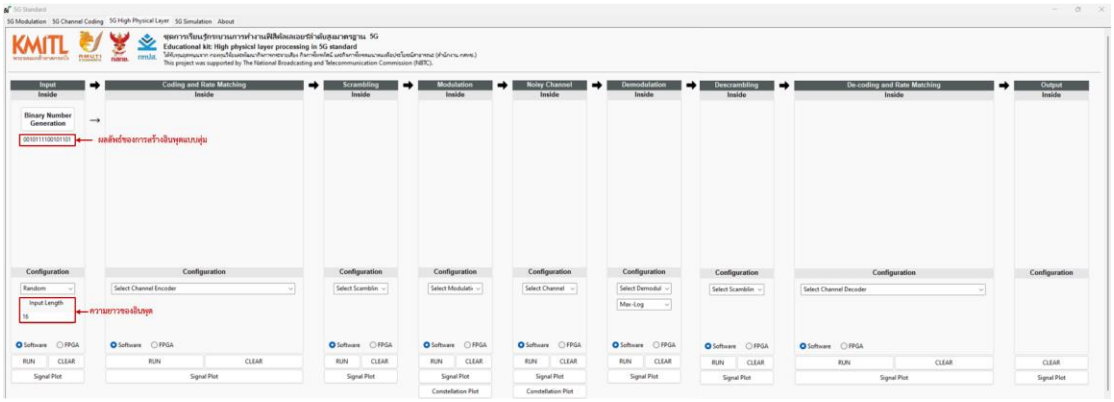
การทดสอบสร้างข้อมูลอินพุตแบบสุ่มโดยใช้ซอฟต์แวร์จะกำหนดให้ข้อมูลอินพุตมีความยาว 16 บิต จากนั้นเมื่อกดปุ่ม “RUN” ซอฟต์แวร์ GUI จะเรียกใช้ซอฟต์แวร์ Module randomInput.py เพื่อสร้างบิตข้อมูลอินพุตและแสดงผลลัพธ์ที่ได้ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน (Inside) ดังแสดงในรูปที่ 4.1 - รูปที่ 4.3 โดยผลลัพธ์ดังกล่าวจะแสดงอยู่ในรูปของบิตข้อมูลแบบไบนารี นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟของสัญญาณอินพุตได้โดยกดปุ่ม “Signal Plot” ตัวอย่างกราฟของสัญญาณอินพุตที่สร้างโดยใช้ซอฟต์แวร์แสดงดังรูปที่ 4.4



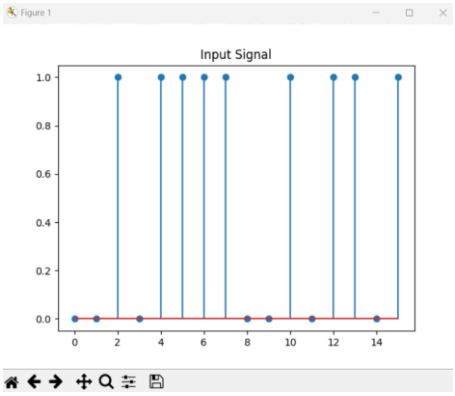
รูปที่ 4.1 ผลลัพธ์การสร้างข้อมูลอินพุตแบบสุ่มโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



รูปที่ 4.2 ผลลัพธ์การสร้างข้อมูลอินพุตแบบสุ่มโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



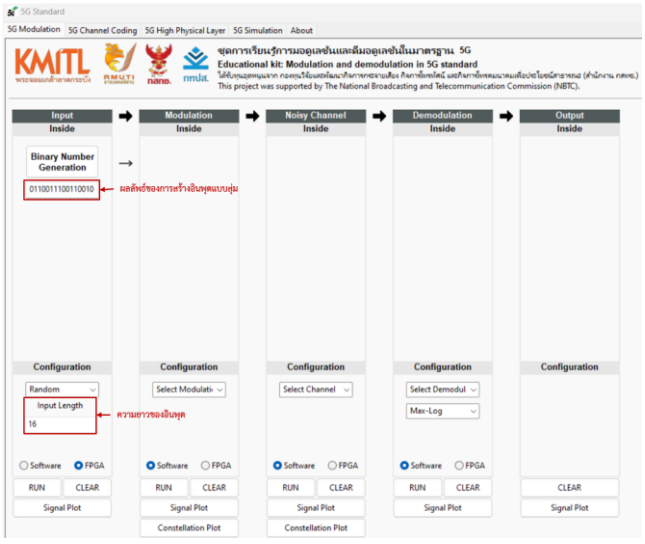
รูปที่ 4.3 ผลลัพธ์การสร้างข้อมูลอินพุตแบบสุ่มโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G



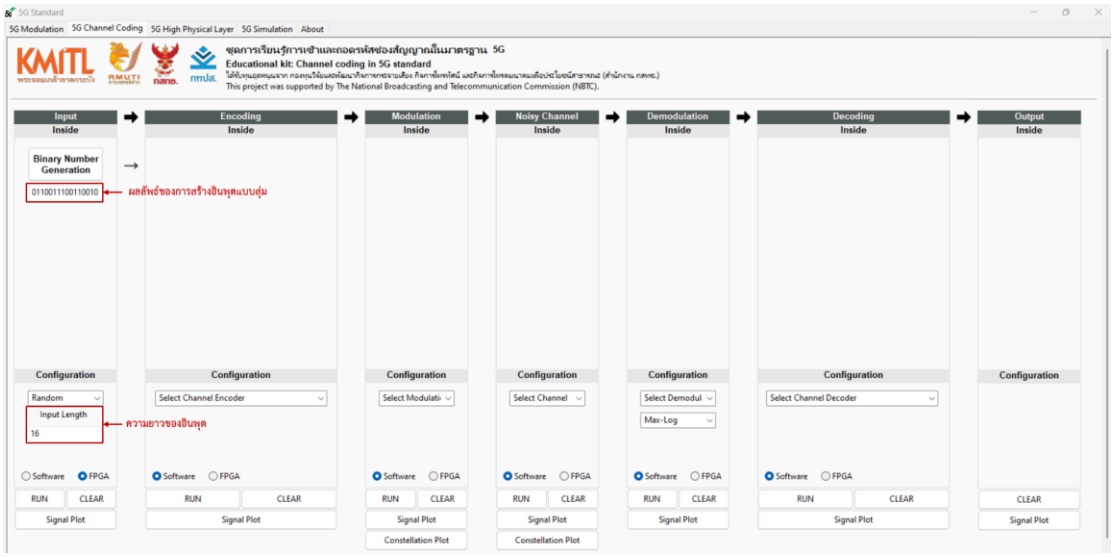
รูปที่ 4.4 กราฟของสัญญาณข้อมูลอินพุตแบบสุ่มที่สร้างโดยใช้ซอฟต์แวร์

- กรณีใช้อุปกรณ์ FPGA สร้างข้อมูลอินพุตแบบสุ่ม

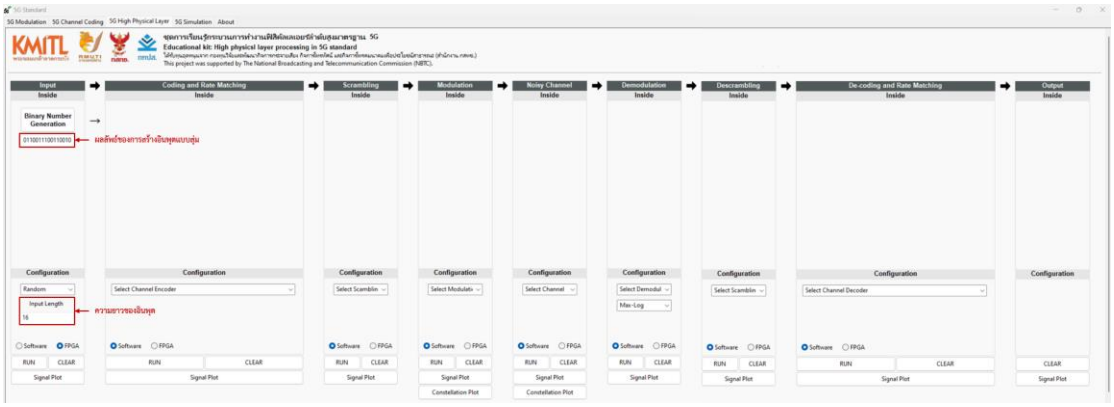
การทดสอบสร้างข้อมูลอินพุตแบบสุ่มโดยใช้อุปกรณ์ FPGA จะกำหนดให้อินพุตมีความยาว 16 บิต เมื่อกดปุ่ม “RUN” ซอฟต์แวร์ GUI จะส่งค่าข้อมูลที่กำหนดความยาวของบิตผ่าน Serial Port ไปยังอุปกรณ์ FPGA จากนั้นอุปกรณ์ FPGA จะดำเนินการสร้างบิตข้อมูลอินพุตแบบสุ่มและส่งผลลัพธ์ผ่าน Serial Port กลับมาแสดงบน GUI ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.5 - รูปที่ 4.7 โดยผลลัพธ์ดังกล่าวจะแสดงอยู่ในรูปของบิตข้อมูลแบบไบนารี นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟของสัญญาณอินพุตได้โดยกดปุ่ม “Signal Plot” ตัวอย่างกราฟของสัญญาณอินพุตที่สร้างโดยใช้อุปกรณ์ FPGA แสดงดังรูปที่ 4.8



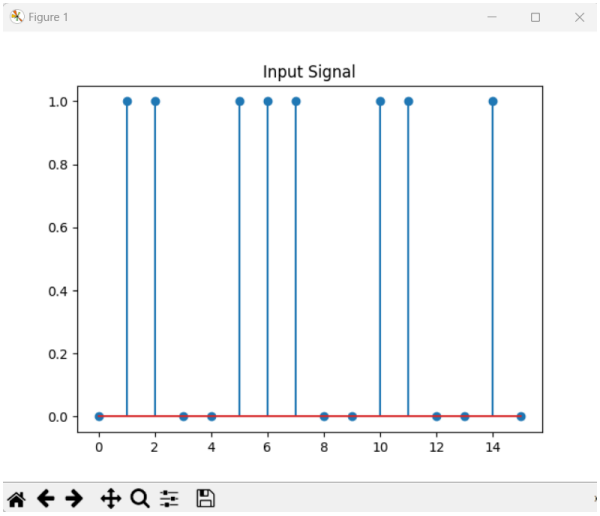
รูปที่ 4.5 ผลลัพธ์การสร้างข้อมูลอินพุตแบบสุ่มโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



รูปที่ 4.6 ผลลัพธ์การสร้างข้อมูลอินพุตแบบสุ่มโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



รูปที่ 4.7 ผลลัพธ์การสร้างข้อมูลอินพุตแบบสุ่มโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G



รูปที่ 4.8 กราฟของสัญญาณข้อมูลอินพุตแบบสุ่มที่สร้างโดยใช้อุปกรณ์ FPGA

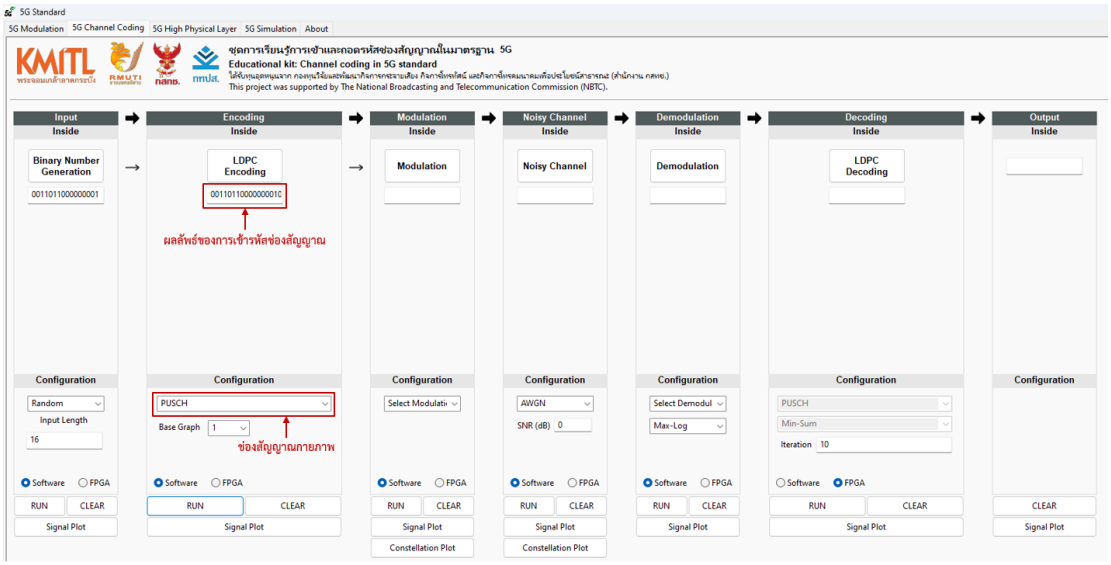
4.1.2 ผลการทดสอบ GUI เพื่อแสดงผลการเข้ารหัสช่องสัญญาณมาตรฐาน 5G

การเข้ารหัสช่องสัญญาณมาตรฐาน 5G แสดงอยู่ในคอลัมน์ที่ 2 ของหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G และหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G โดยผู้ใช้สามารถเลือกช่องสัญญาณภาพได้ทั้งหมด 5 ช่องสัญญาณ ได้แก่ PUSCH, PDSCH, PUCCH, PDCCH และ PBCH นอกจากนี้ ผู้ใช้สามารถเลือกทดสอบการเข้ารหัสช่องสัญญาณโดยใช้ซอฟต์แวร์หรืออุปกรณ์ FPGA

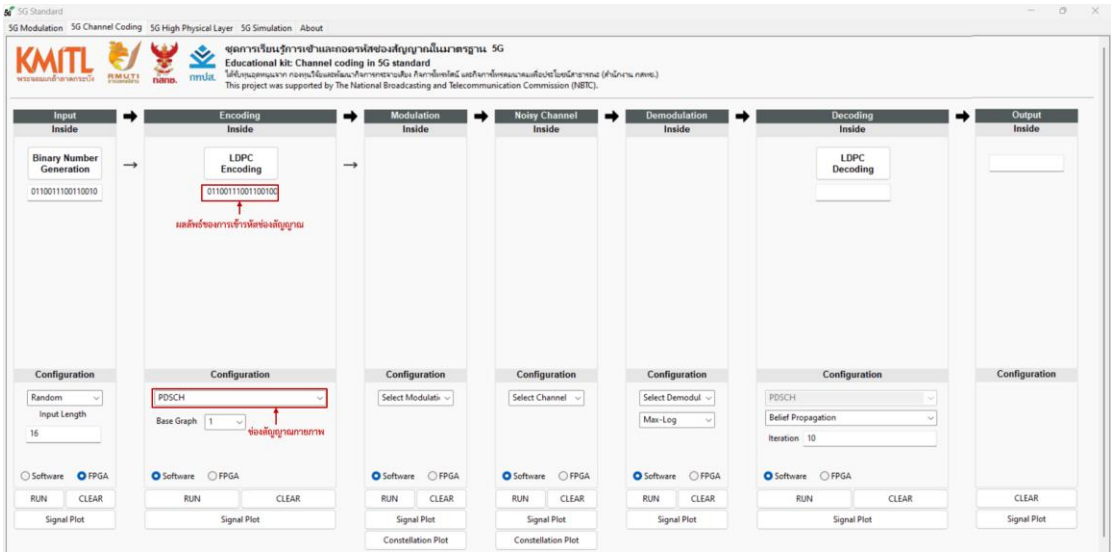
4.1.2.1 ผลการทดสอบ GUI สำหรับชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G

- กรณีใช้ซอฟต์แวร์เข้ารหัสช่องสัญญาณ

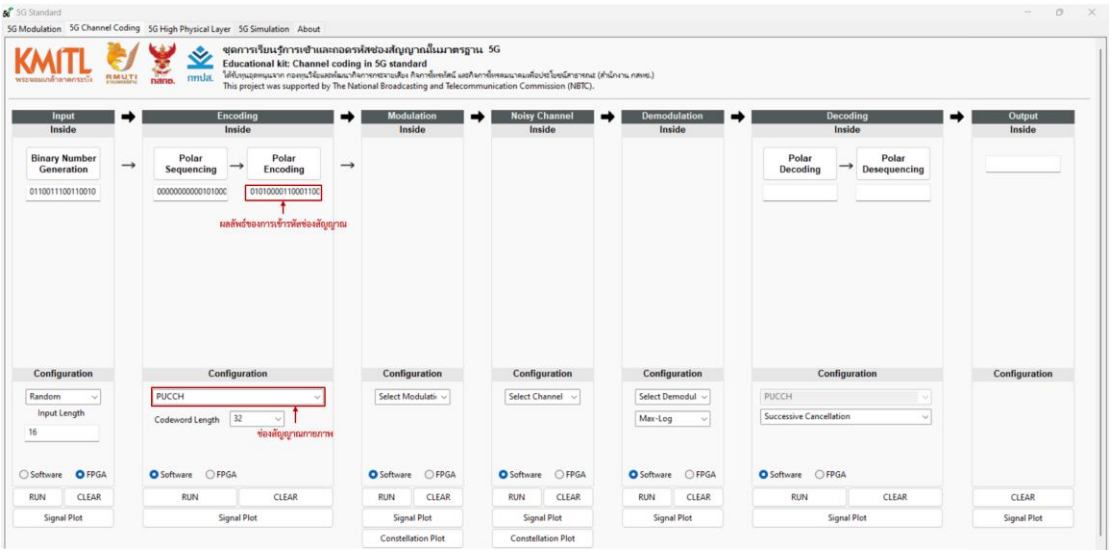
การทดสอบการเข้ารหัสช่องสัญญาณโดยใช้ซอฟต์แวร์จะกำหนดให้ข้อมูลอินพุตมีความยาว 16 บิตซึ่งสร้างจากการทดสอบข้อมูลอินพุตแบบสุ่มโดยใช้ซอฟต์แวร์ และเลือกช่องสัญญาณกายภาพ จากนั้นเมื่อกดปุ่ม “RUN” กรณีเลือกช่องสัญญาณกายภาพชนิด PUSCH และ PDSCH ซอฟแวร์ GUI จะเรียกใช้ซอฟแวร์ Module LDPCEncoderModule.py เพื่อดำเนินการเข้ารหัสช่องสัญญาณ และกรณีที่เลือกช่องสัญญาณ PUCCH, PDCCH และ PBCH ซอฟต์แวร์ GUI จะเรียกใช้ซอฟต์แวร์ Module PolarEncoderModule.py เพื่อดำเนินการเข้ารหัสช่องสัญญาณ จากนั้นจะแสดงผลลัพธ์ที่ได้ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.9 - รูปที่ 4.13 โดยผลลัพธ์ดังกล่าวจะอยู่ในรูปของบิตคำรหัสแบบไบนารี นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟของสัญญาณที่ได้จากการเข้ารหัสได้โดยกดปุ่ม “Signal Plot” ตัวอย่างกราฟของสัญญาณที่ได้จากการเข้ารหัสแสดงดังรูปที่ 4.14 - รูปที่ 4.18



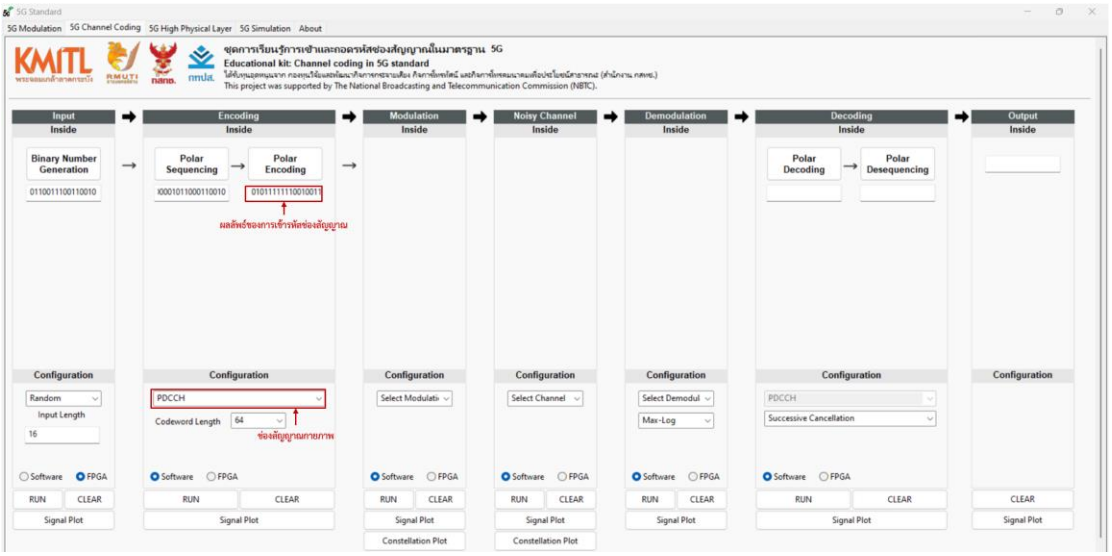
รูปที่ 4.9 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PUSCH โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



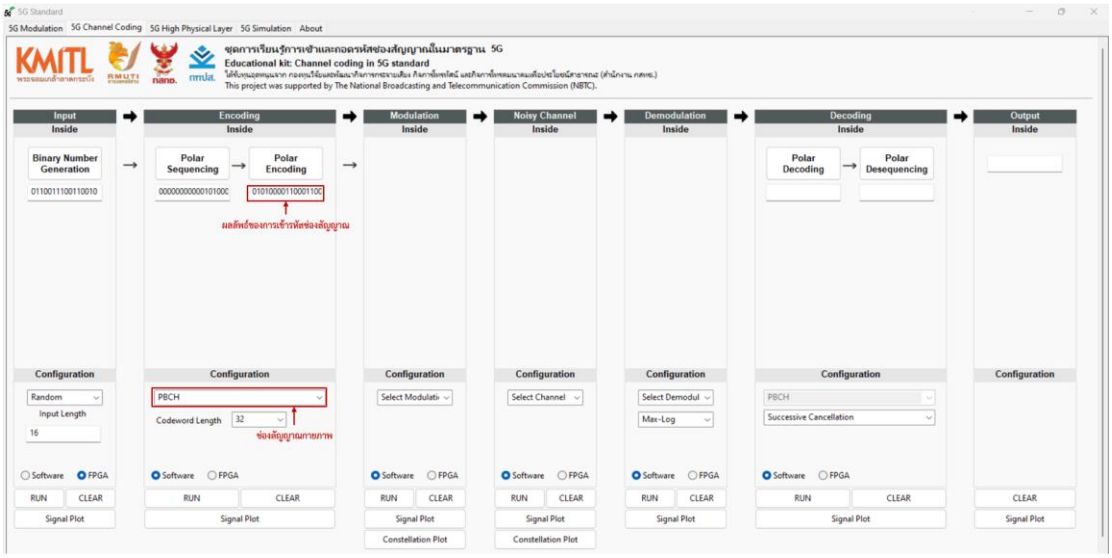
รูปที่ 4.10 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PDSCH โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



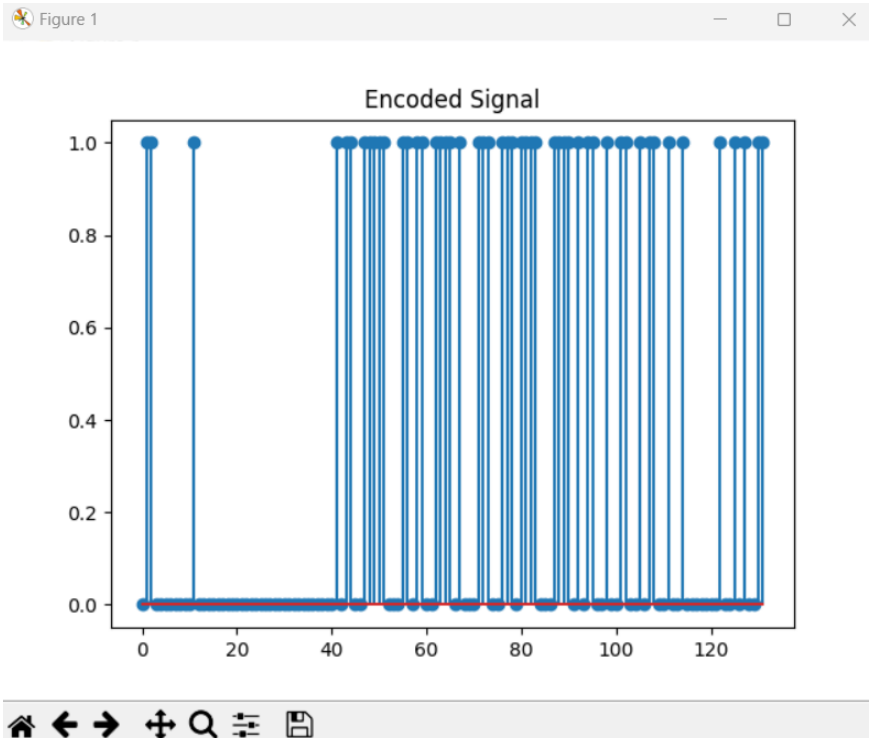
รูปที่ 4.11 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PUCCH โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



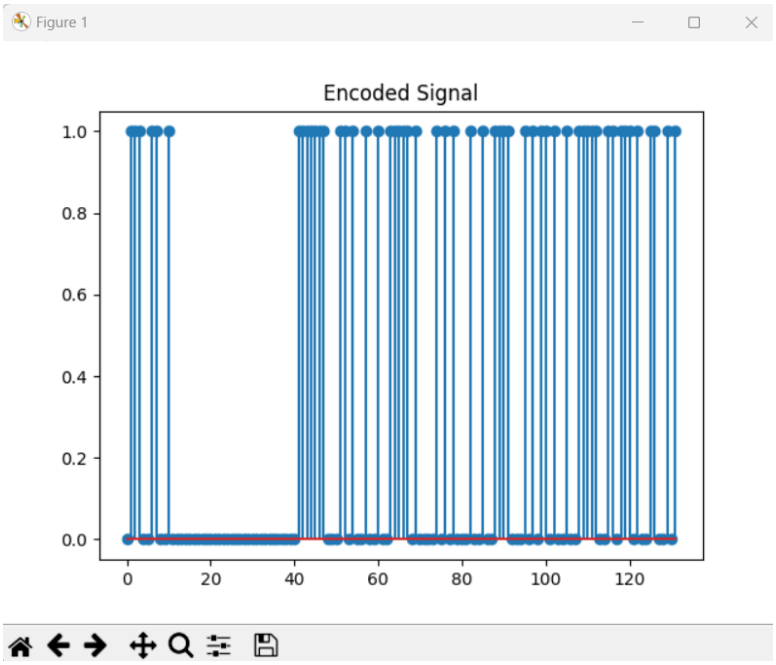
รูปที่ 4.12 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PDCCH โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



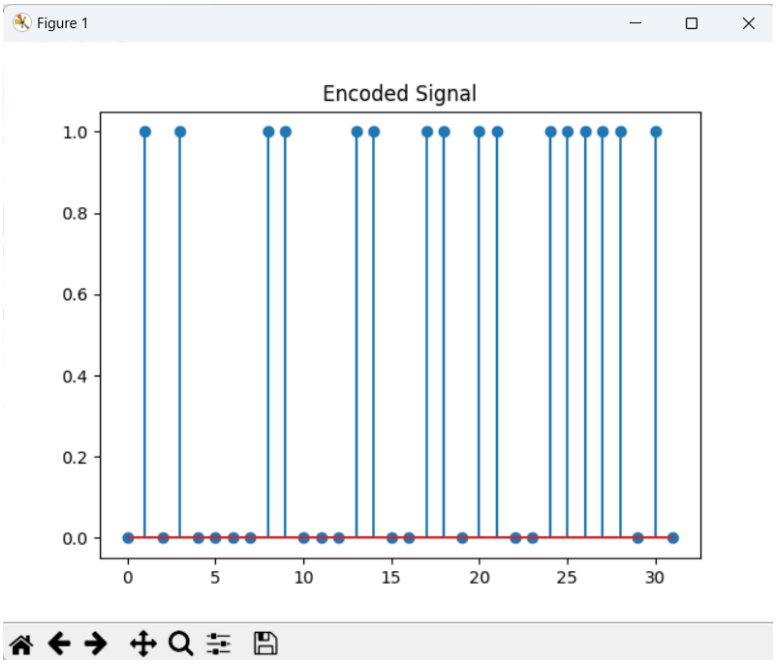
รูปที่ 4.13 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PBCH โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



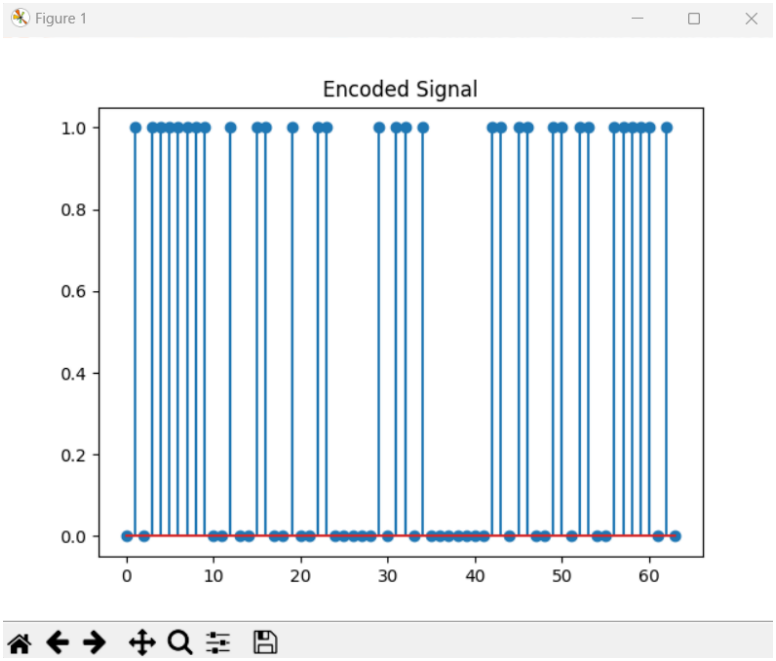
รูปที่ 4.14 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PUSCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



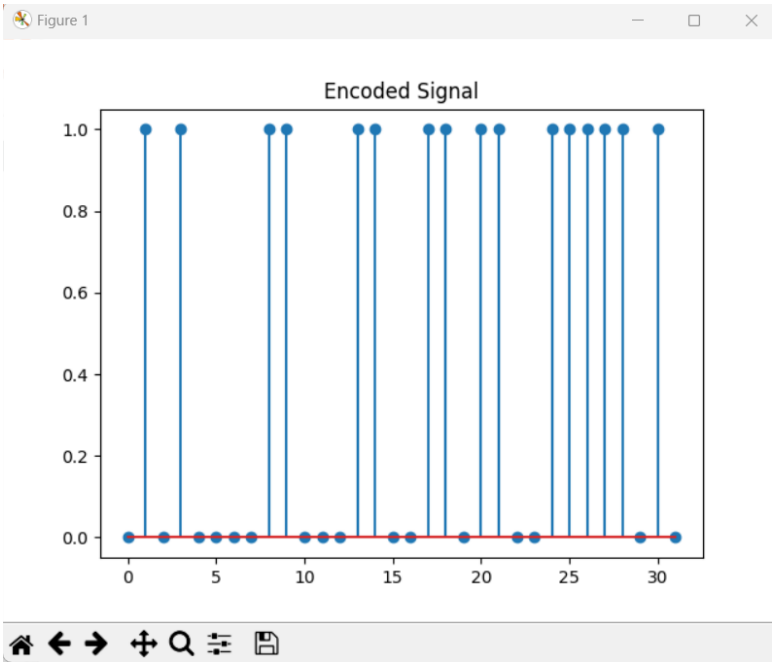
รูปที่ 4.15 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PDSCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



รูปที่ 4.16 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PUCCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



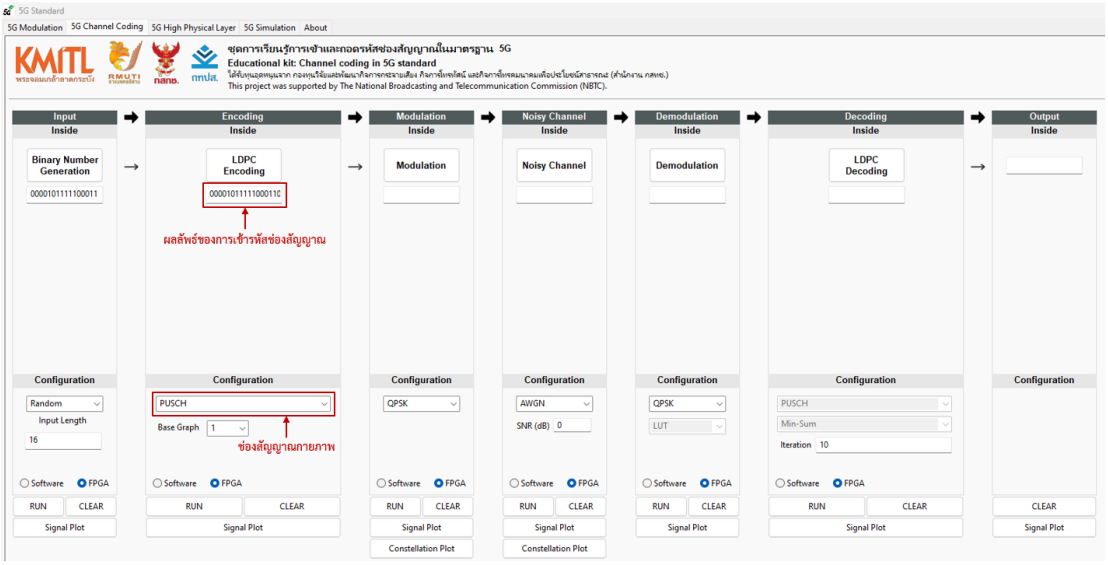
รูปที่ 4.17 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PDCCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



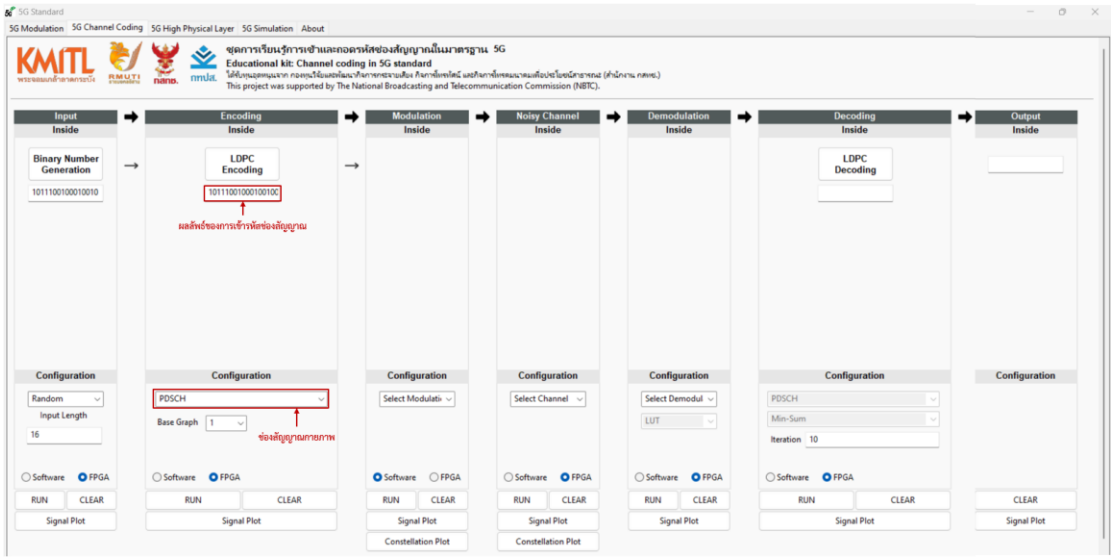
รูปที่ 4.18 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PBCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G

- กรณีใช้อุปกรณ์ FPGA เข้ารหัสช่องสัญญาณ

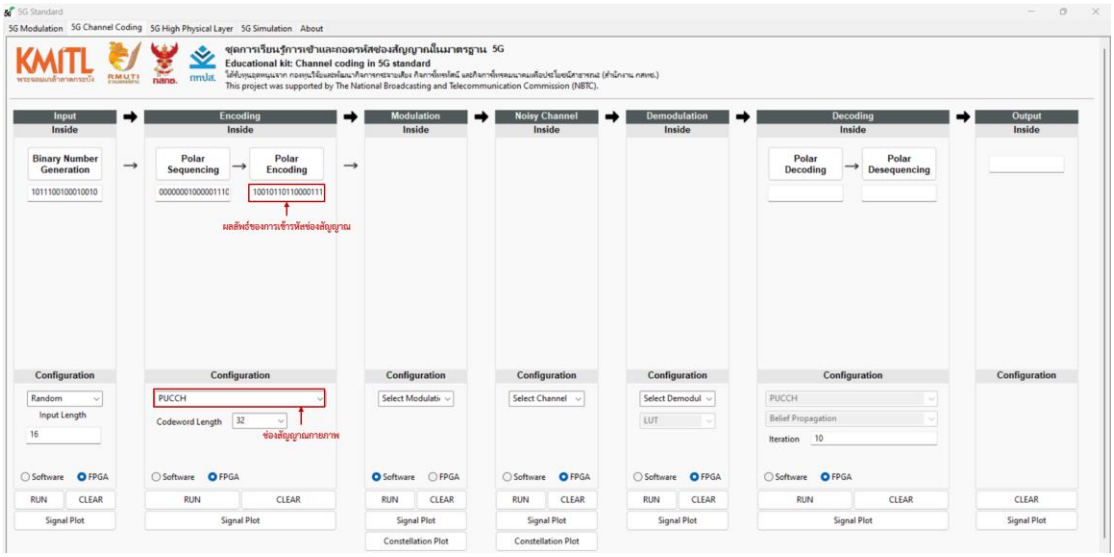
การทดสอบการเข้ารหัสช่องสัญญาณโดยใช้อุปกรณ์ FPGA จะกำหนดให้อินพุตมีความยาว 16 บิต ซึ่งสร้างจากการทดสอบข้อมูลอินพุตแบบสุ่มโดยใช้ FPGA และเลือกช่องสัญญาณกายภาพ จากนั้นเมื่อกดปุ่ม “RUN” กรณีเลือกช่องสัญญาณกายภาพชนิด PUSCH และ PDSCH ซอฟต์แวร์ GUI จะส่งบิตข้อมูลอินพุตและกราฟฐานผ่าน Serial Port ไปยังอุปกรณ์ FPGA และกรณีช่องสัญญาณ PUCCH, PDCCH และ PBCH ซอฟต์แวร์ GUI จะส่งบิตข้อมูลอินพุตและความยาวรหัสหลังการพังค์เจอร์ จากนั้นอุปกรณ์ FPGA จะดำเนินการเข้ารหัสช่องสัญญาณและส่งผลลัพธ์ผ่าน Serial Port กลับมาแสดงบน GUI ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.19 - รูปที่ 4.23 โดยผลลัพธ์ดังกล่าวจะแสดงอยู่ในรูปของบิตคำรหัสแบบไบนารี นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟของสัญญาณที่ได้จากการเข้ารหัสได้โดยกดปุ่ม “Signal Plot” ตัวอย่างกราฟของสัญญาณที่ได้จากการเข้ารหัสแสดงดังรูปที่ 4.24 - รูปที่ 4.28



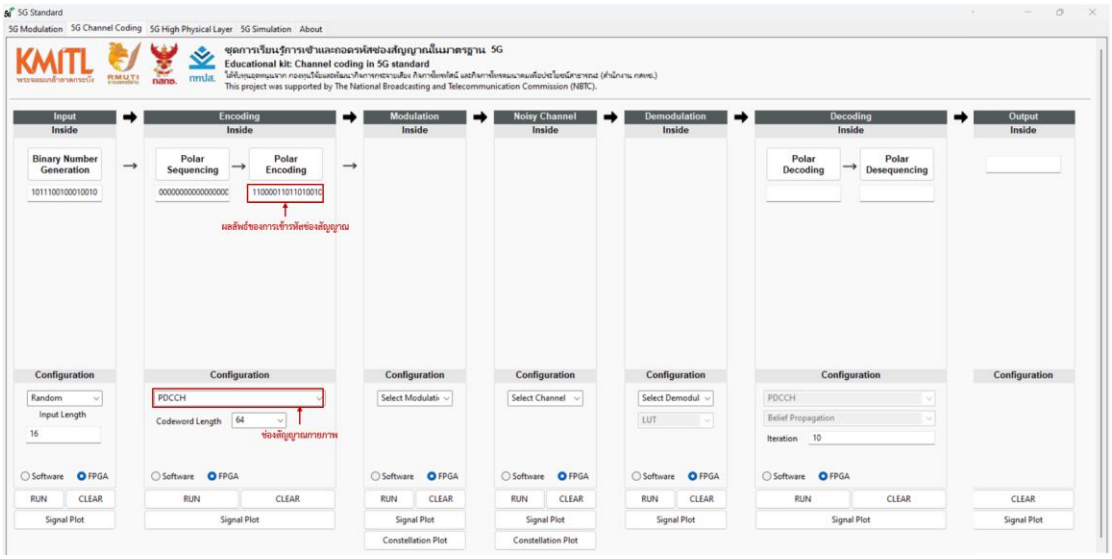
รูปที่ 4.19 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PUSCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



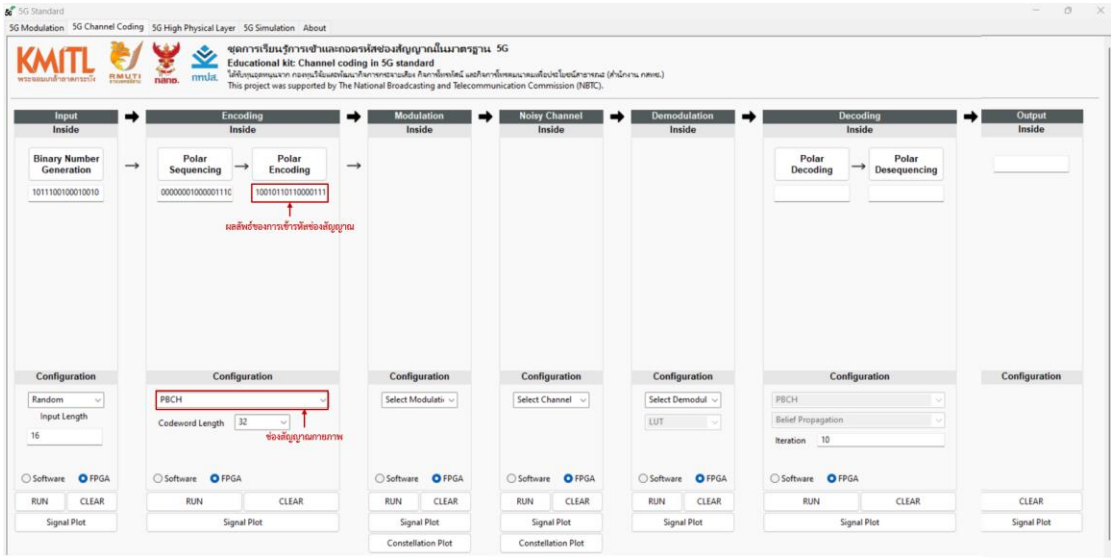
รูปที่ 4.20 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PDSCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



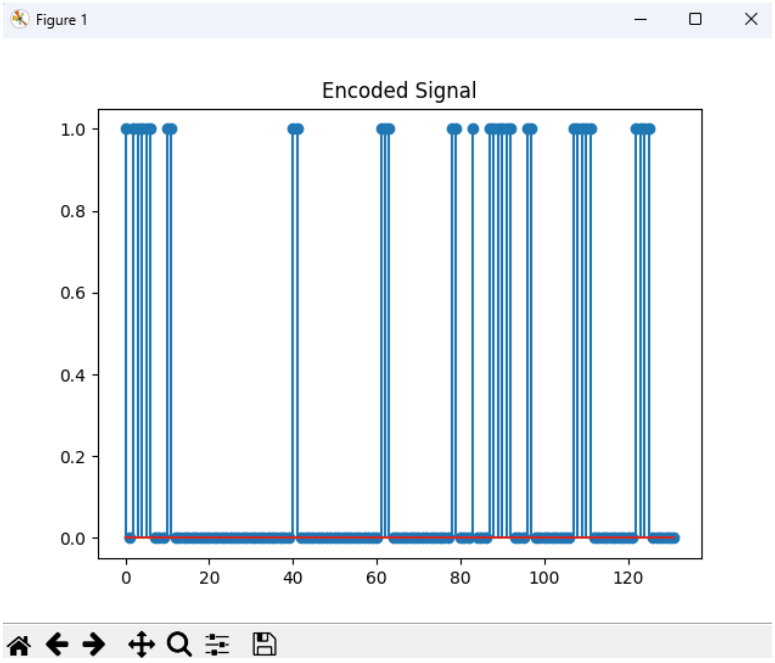
รูปที่ 4.21 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PUCCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



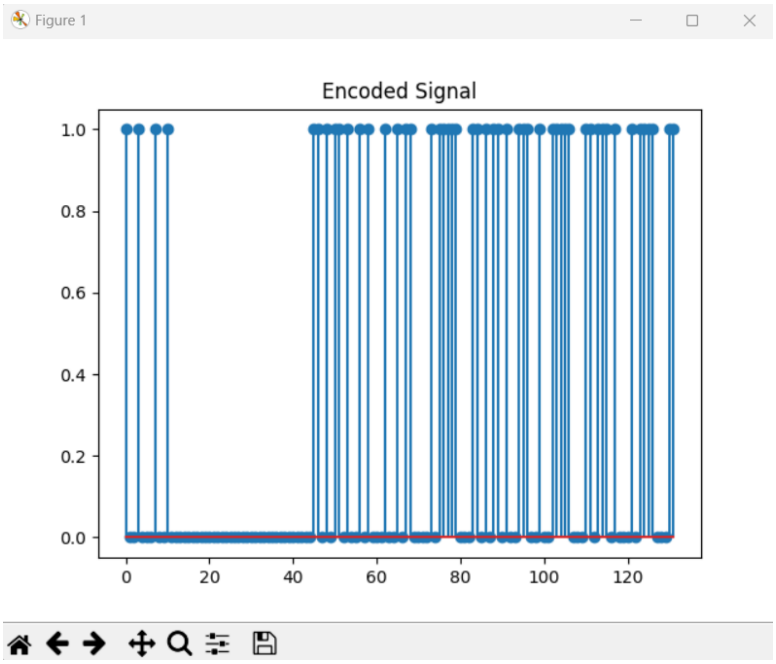
รูปที่ 4.22 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PDCCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



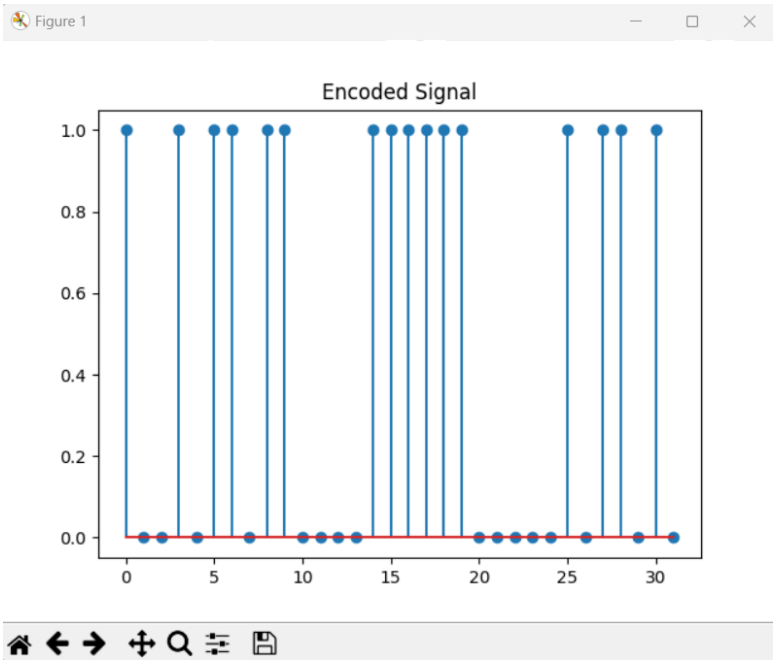
รูปที่ 4.23 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PBCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



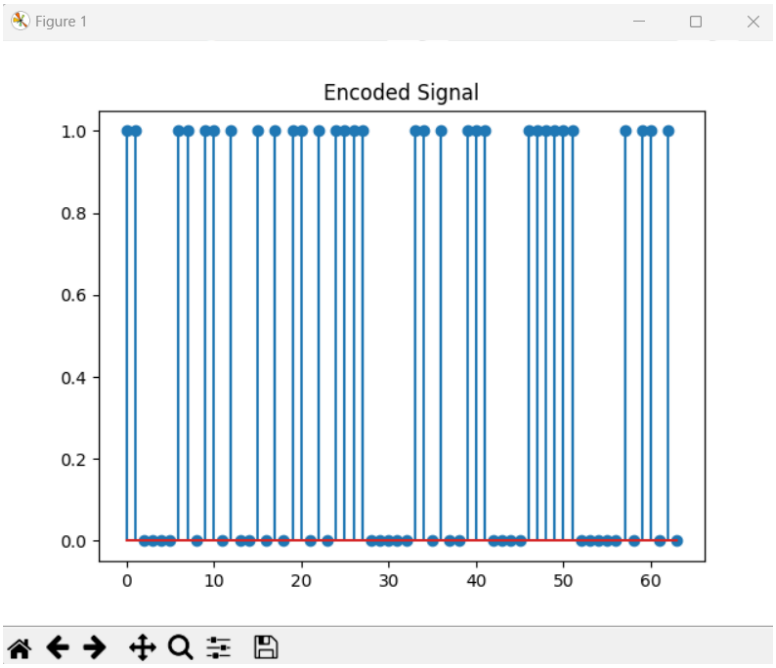
รูปที่ 4.24 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PUSCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



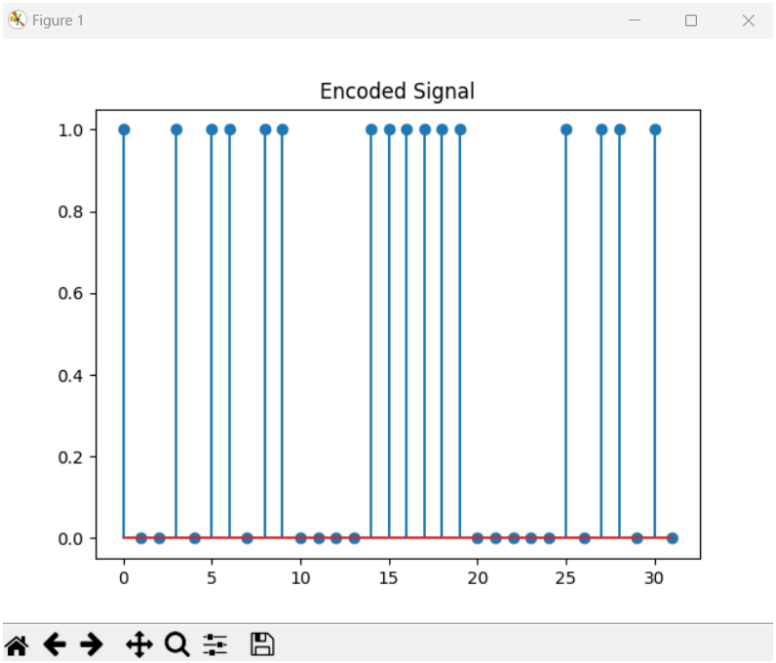
รูปที่ 4.25 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PDSCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



รูปที่ 4.26 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PUCCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



รูปที่ 4.27 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PUCCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G

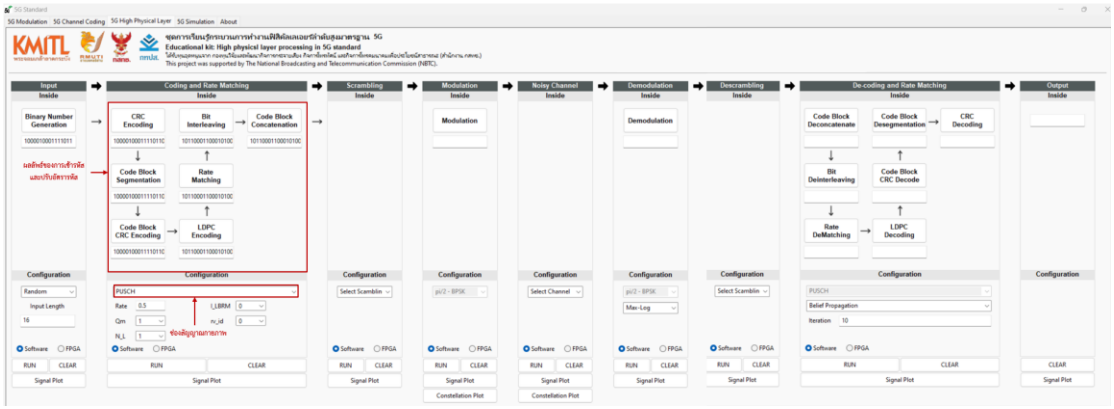


รูปที่ 4.28 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PBCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G

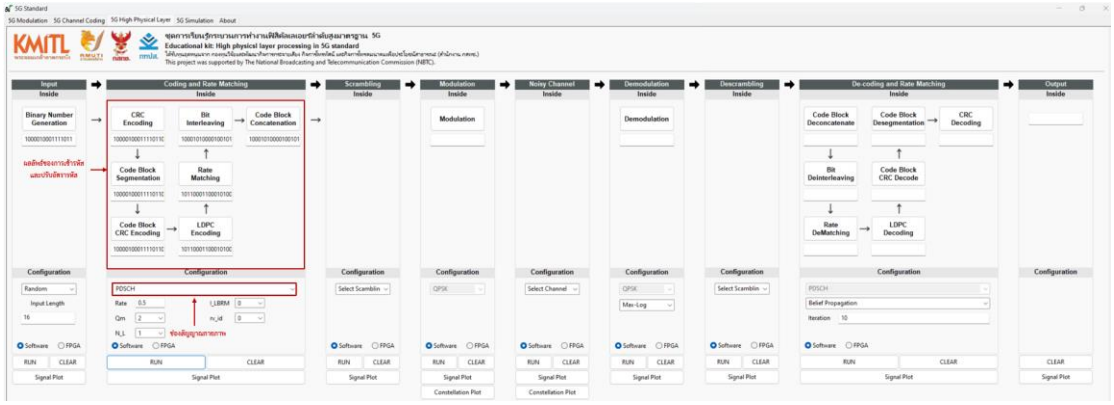
4.1.2.2 ผลการทดสอบ GUI สำหรับชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G

- กรณีใช้ซอฟต์แวร์เข้ารหัสและปรับอัตรารหัส

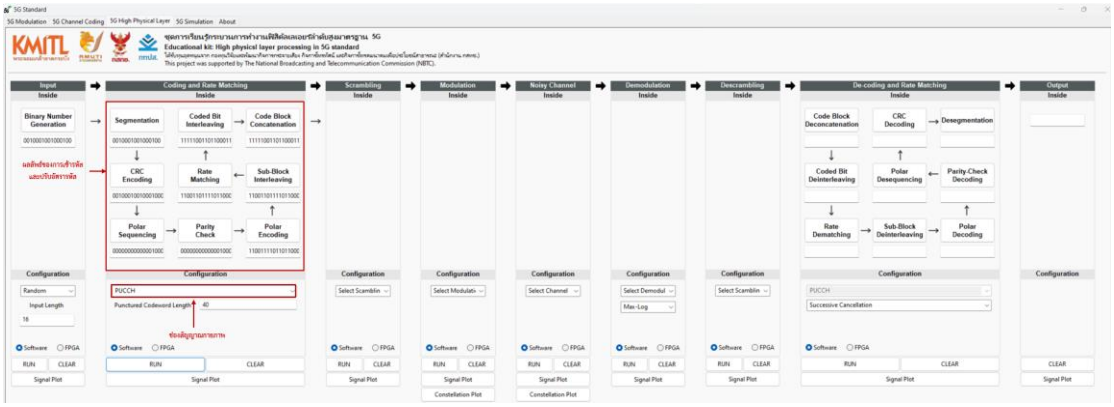
การทดสอบการเข้ารหัสและปรับอัตรารหัสโดยใช้ซอฟต์แวร์จะกำหนดให้ข้อมูลอินพุตมีความยาว 16 บิตซึ่งสร้างจากการทดสอบข้อมูลอินพุตแบบสุ่มโดยใช้ซอฟต์แวร์ และเลือกช่องสัญญาณกายภาพ จากนั้นเมื่อกดปุ่ม “RUN” กรณีเลือกช่องสัญญาณกายภาพชนิด PUSCH และ PDSCH ซอฟแวร์ GUI จะเรียกใช้ซอฟแวร์ Module LDPCEncoderModule.py เพื่อดำเนินการเข้ารหัสและปรับอัตรารหัส และกรณีที่เลือกช่องสัญญาณ PUCCH, PDCCH และ PBCH ซอฟต์แวร์ GUI จะเรียกใช้ซอฟต์แวร์ Module PolarEncoderModule.py เพื่อดำเนินการเข้ารหัสและปรับอัตรารหัส จากนั้นแสดงผลลัพธ์ที่ได้ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.29 - รูปที่ 4.33 โดยผลลัพธ์ดังกล่าวจะอยู่ในรูปของบิตคำรหัสแบบไบนารี นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟของสัญญาณที่ได้จากการเข้ารหัสได้โดยกดปุ่ม “Signal Plot” ตัวอย่างกราฟของสัญญาณที่ได้จากการเข้ารหัสแสดงดังรูปที่ 4.34 - รูปที่ 4.38



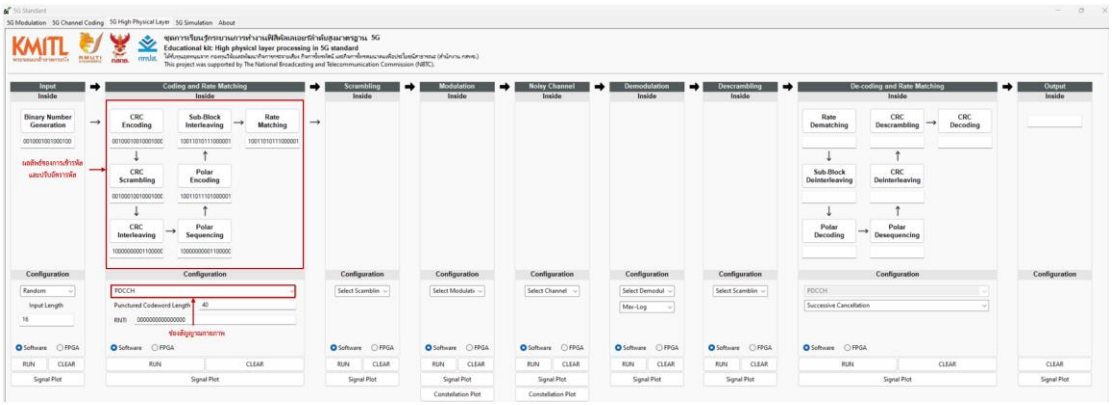
รูปที่ 4.29 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PUSCH โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



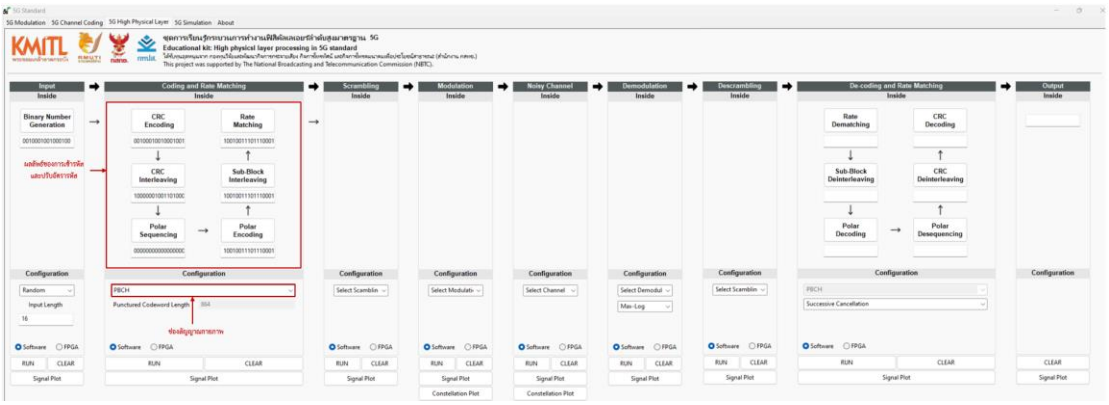
รูปที่ 4.30 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PDSCH โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



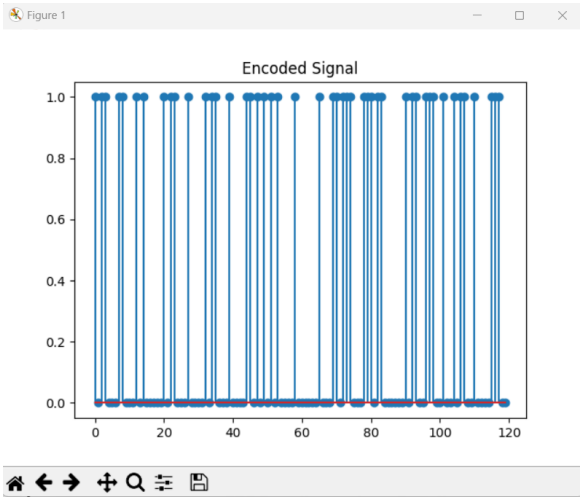
รูปที่ 4.31 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PUCCH โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



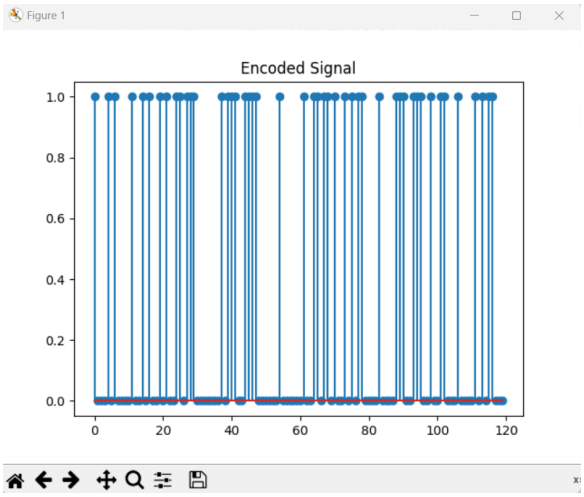
รูปที่ 4.32 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PDCCH โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



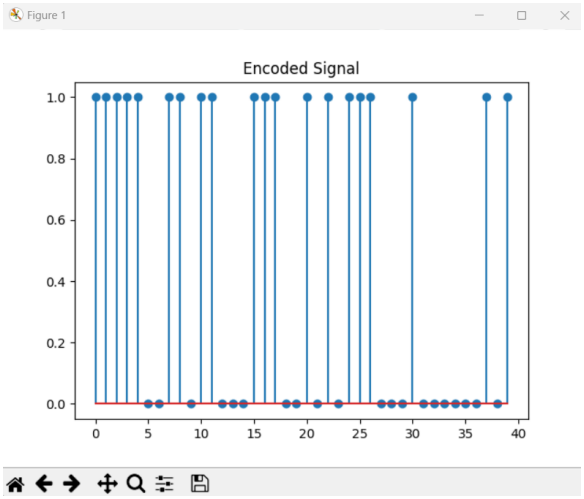
รูปที่ 4.33 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PBCH โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



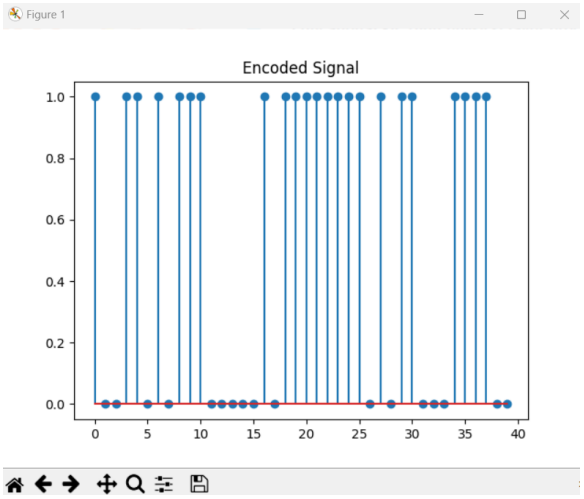
รูปที่ 4.34 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PUSCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



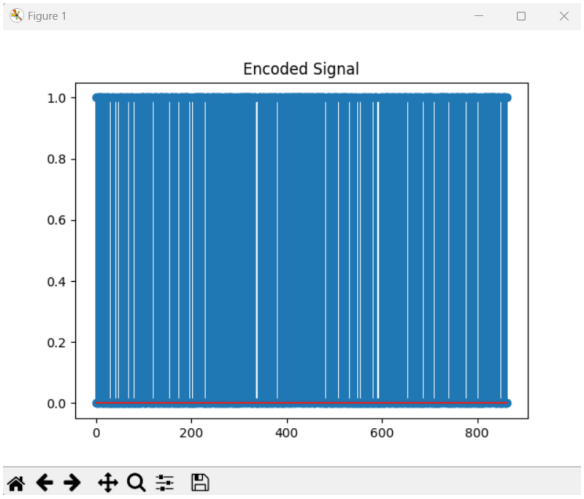
รูปที่ 4.35 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PDSCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



รูปที่ 4.36 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PUCCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



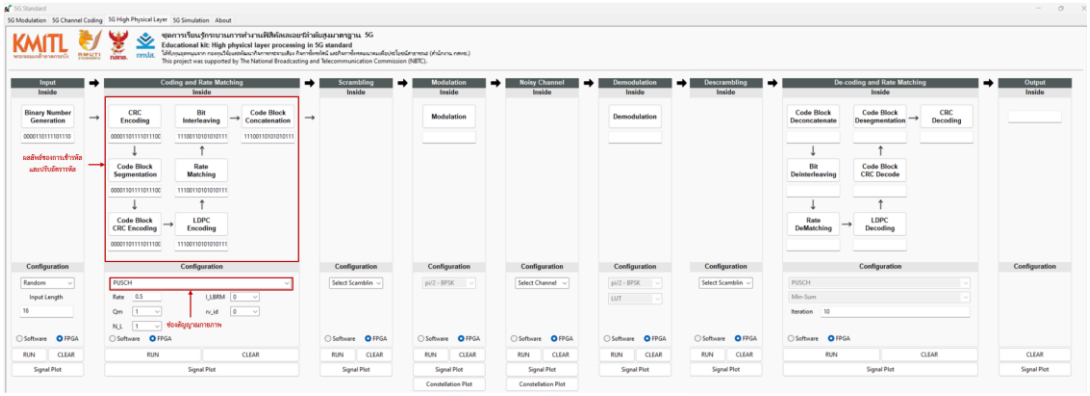
รูปที่ 4.37 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PDCCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



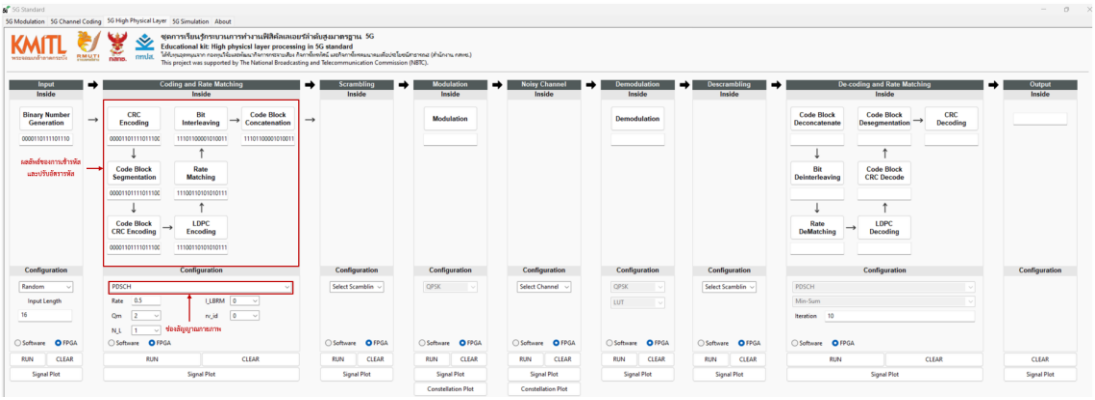
รูปที่ 4.38 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PBCH ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G

- กรณีใช้อุปกรณ์ FPGA เข้ารหัสและปรับอัตรารหัส

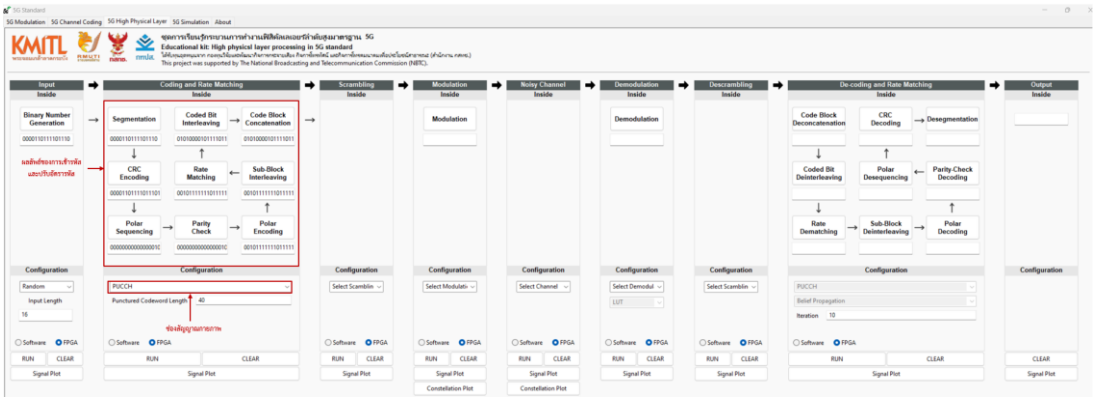
การทดสอบการเข้ารหัสและปรับอัตรารหัสโดยใช้อุปกรณ์ FPGA จะกำหนดให้อินพุตมีความยาว 16 บิต ซึ่งสร้างจากการทดสอบข้อมูลอินพุตแบบสุ่มโดยใช้ FPGA และเลือกช่องสัญญาณกายภาพ เมื่อกดปุ่ม “RUN” กรณีเลือกช่องสัญญาณกายภาพชนิด PUSCH และ PDSCH ซอฟต์แวร์ GUI จะส่งบิตข้อมูลอินพุตและกราฟฐานผ่าน Serial Port ไปยังอุปกรณ์ FPGA และกรณีช่องสัญญาณ PUCCH, PDCCH และ PBCH ซอฟต์แวร์ GUI จะส่งบิตข้อมูลอินพุตและความยาวรหัสหลังการพังค์เจอร์ จากนั้นอุปกรณ์ FPGA จะดำเนินการเข้ารหัสช่องสัญญาณและส่งผลลัพธ์ผ่าน Serial Port กลับมาแสดงบน GUI ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.39 - รูปที่ 4.43 โดยผลลัพธ์ดังกล่าวจะแสดงอยู่ในรูปของบิตคำรหัสแบบไบนารี นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟของสัญญาณที่ได้จากการเข้ารหัสได้โดยกดปุ่ม “Signal Plot” ตัวอย่างกราฟของสัญญาณที่ได้จากการเข้ารหัสแสดงดังรูปที่ 4.44 - รูปที่ 4.48



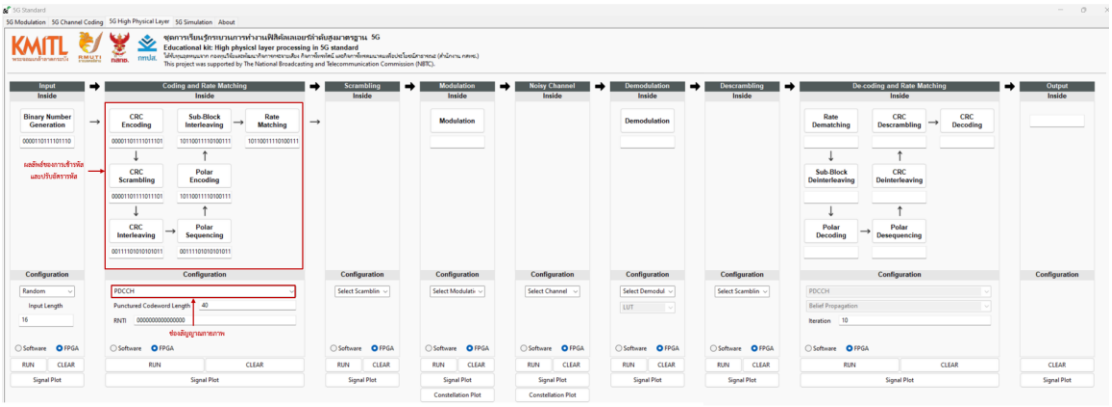
รูปที่ 4.39 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PUSCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



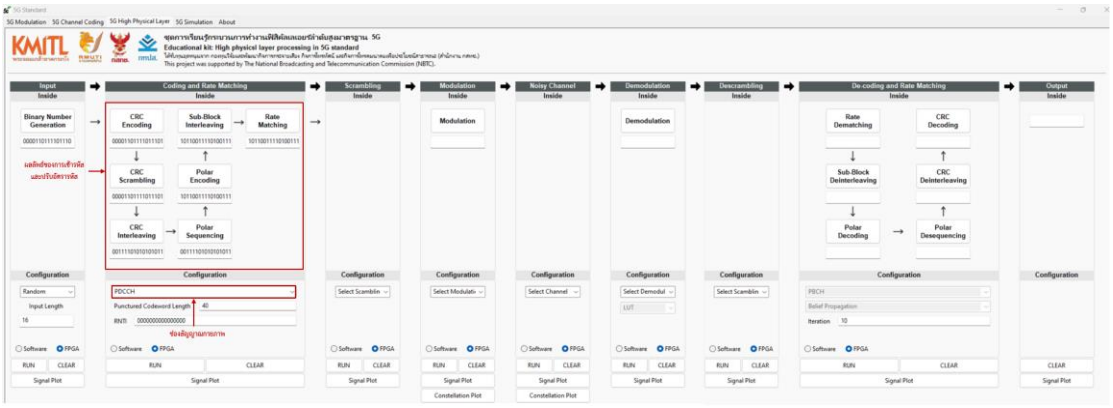
รูปที่ 4.40 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PDSCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



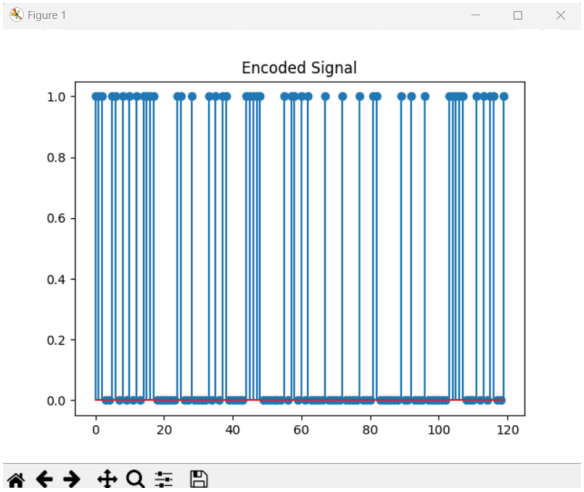
รูปที่ 4.41 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PUCCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



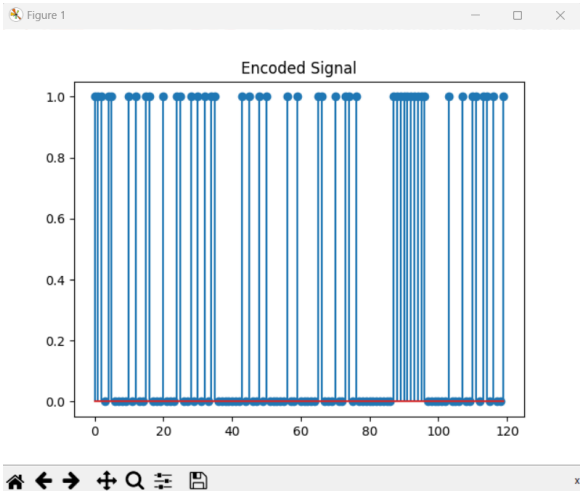
รูปที่ 4.42 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PDCCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



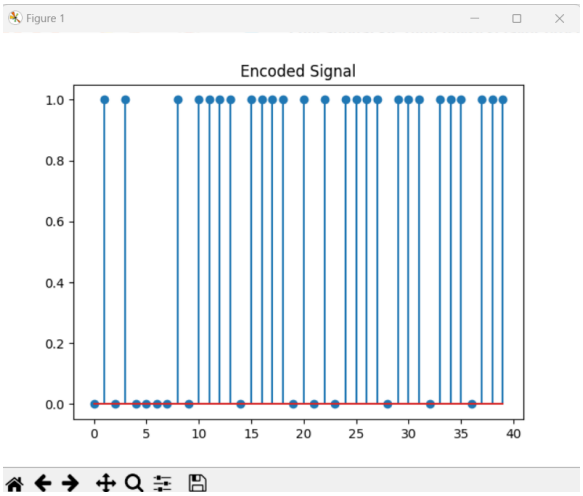
รูปที่ 4.43 ผลลัพธ์การเข้ารหัสช่องสัญญาณมาตรฐาน 5G ชนิด PBCH โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



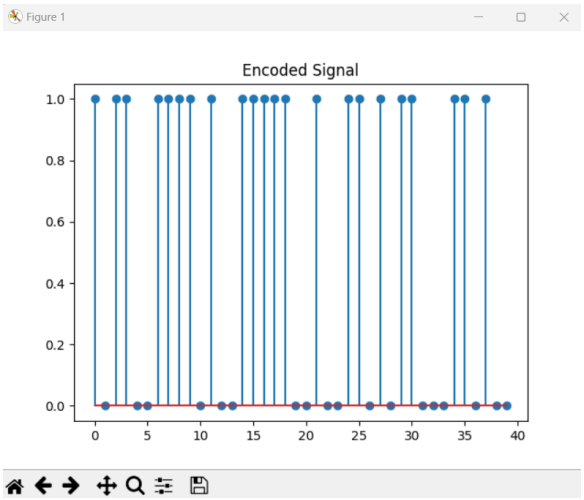
รูปที่ 4.44 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PUSCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



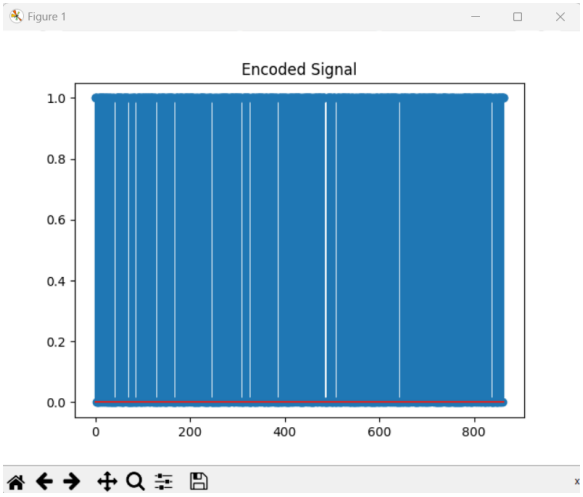
รูปที่ 4.45 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PDSCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



รูปที่ 4.46 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PUCCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



รูปที่ 4.47 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PDCCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



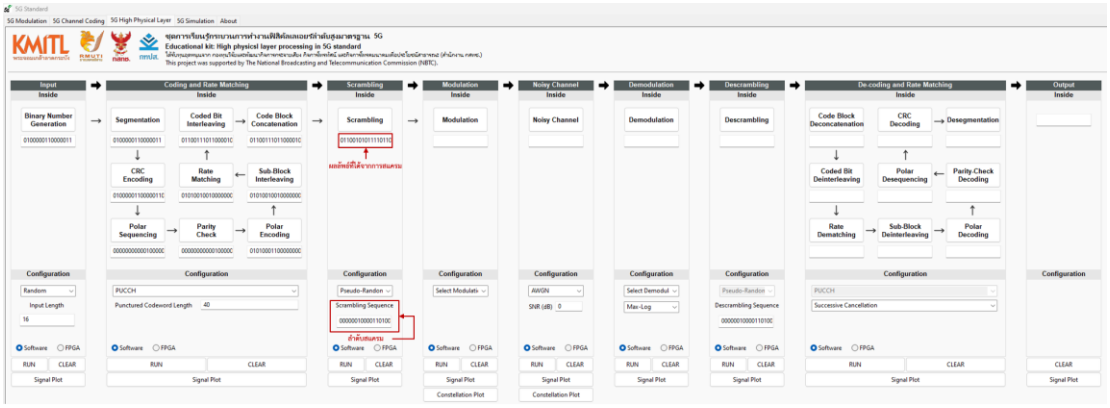
รูปที่ 4.48 กราฟของสัญญาณที่ได้จากการเข้ารหัสช่องสัญญาณชนิด PBCH ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G

4.1.3 ผลการทดสอบ GUI เพื่อแสดงผลการสแครมมาตรฐาน 5G

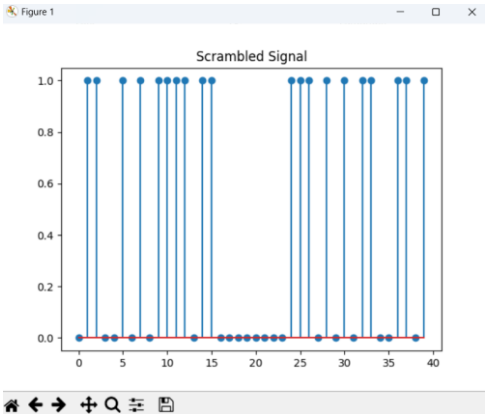
การสแครมมาตรฐาน 5G แสดงอยู่ในคอลัมน์ที่ 3 ของหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G โดยผู้ใช้สามารถเลือกรูปแบบการสร้างลำดับสแครมได้ทั้งหมด 3 รูปแบบ ได้แก่ การใช้ลำดับแบบตายตัว การสร้างลำดับแบบสุ่มเทียม และการป้อนลำดับด้วยตนเอง นอกจากนี้ ผู้ใช้สามารถเลือกทดสอบการสแครมโดยใช้ซอฟต์แวร์หรืออุปกรณ์ FPGA อย่างไรก็ตาม สำหรับการสแครมโดยใช้อุปกรณ์ FPGA จะสามารถสร้างลำดับการสแครมโดยใช้ลำดับแบบตายตัวเท่านั้น

- กรณีใช้ซอฟต์แวร์สแครม

การทดสอบการสแครมโดยใช้ซอฟต์แวร์จะกำหนดให้ความยาวของลำดับการสแครมเท่ากับความยาวของเอาต์พุตที่ได้จากการเข้ารหัสและปรับอัตรารหัส และเลือกการสร้างลำดับแบบสุ่มเทียม จากนั้นเมื่อกดปุ่ม “RUN” ซอฟแวร์ GUI จะเรียกใช้ซอฟแวร์ Module nrPDSCHPRBS.py เพื่อดำเนินการสร้างลำดับการสแครมและดำเนินการสแครม จากนั้นแสดงผลลัพธ์ที่ได้ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.49 โดยผลลัพธ์ดังกล่าวจะอยู่ในรูปของบิตแบบไบนารี นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟของสัญญาณสแครมได้โดยกดปุ่ม “ Signal Plot” ตัวอย่างกราฟของสัญญาณสแครมแสดงดังรูปที่ 4.50



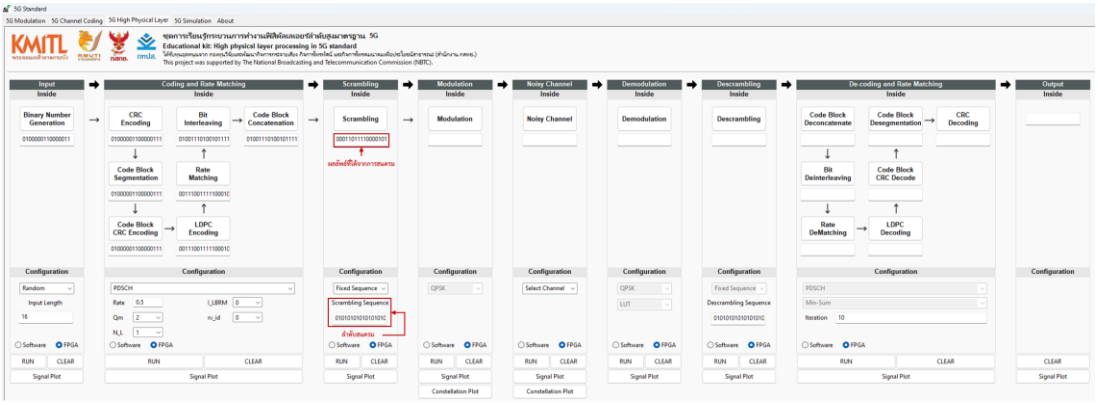
รูปที่ 4.49 ผลลัพธ์การสแครมมาตรฐาน 5G โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



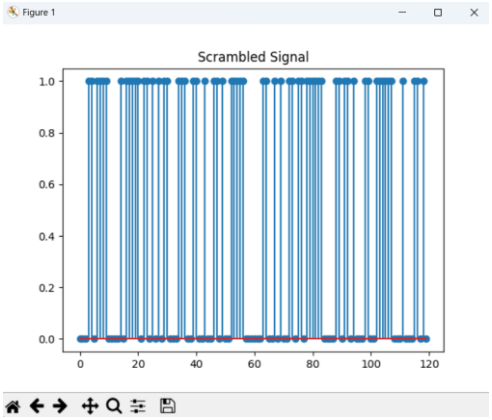
รูปที่ 4.50 กราฟของสัญญาณสแครมที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G

- กรณีใช้อุปกรณ์ FPGA สแครม

การทดสอบการสแครมโดยใช้อุปกรณ์ FPGA จะกำหนดให้ความยาวของลำดับการสแครมเท่ากับความยาวของเอาต์พุตที่ได้จากการเข้ารหัสและปรับอัตรารหัส และเลือกใช้ลำดับสแครมแบบตายตัว เมื่อกดปุ่ม “RUN” ซอฟต์แวร์ GUI จะส่งความยาวเอาต์พุตและรูปแบบการสร้างลำดับสแครม ผ่าน Serial Port ไปยังอุปกรณ์ FPGA จากนั้นอุปกรณ์ FPGA จะดำเนินการสร้างลำดับสแครมและดำเนินการสแครม จากนั้นส่งผลลัพธ์กลับมาแสดงบน GUI ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.51 โดยผลลัพธ์ดังกล่าวจะแสดงอยู่ในรูปของบิตคำรหัสแบบไบนารี นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟของสัญญาณที่ได้จากการสแครมได้โดยกดปุ่ม “Signal Plot” ตัวอย่างกราฟของสัญญาณที่ได้จากการเข้ารหัสแสดงดังรูปที่ 4.52



รูปที่ 4.51 ผลลัพธ์การสแครมมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



รูปที่ 4.52 กราฟของสัญญาณที่ได้จากการสแครมที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G

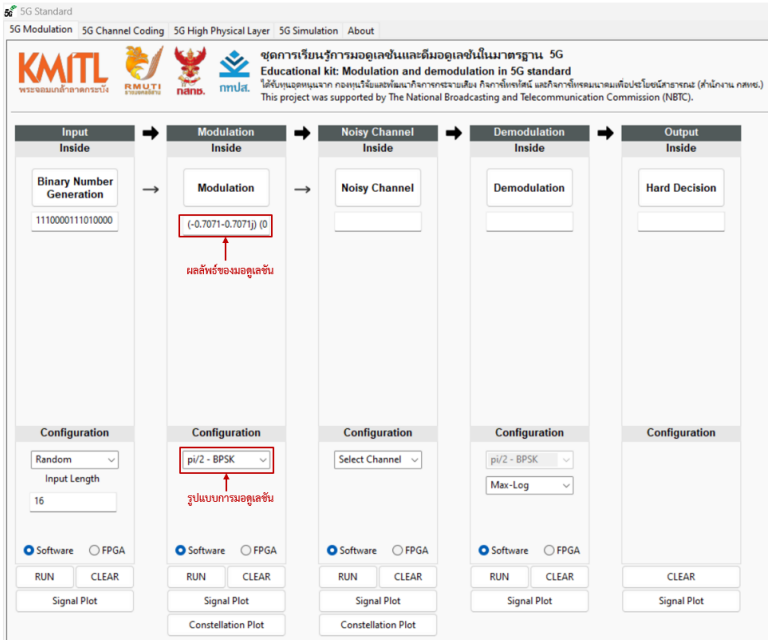
4.1.4 ผลการทดสอบ GUI เพื่อแสดงผลการมอดูเลชันมาตรฐาน 5G

การมอดูเลชันมาตรฐาน 5G แสดงอยู่ในคอลัมน์ที่ 2 ของหน้าชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G คอลัมน์ที่ 3 ของหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G และคอลัมน์ที่ 4 ของหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G โดยผู้ใช้สามารถเลือกรูปแบบการมอดูเลชันได้ทั้งหมด 7 รูปแบบ ได้แก่ BPSK, $\pi/2$-BPSK, QPSK, 16QAM, 64QAM, 256QAM และ 1024QAM อย่างไรก็ตาม สำหรับหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G และชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงจะตัวเลือกรูปแบบการมอดูเลชันจะสอดคล้องกับช่องสัญญาณกายภาพที่อยู่ในส่วนการเข้ารหัสช่องสัญญาณ นอกจากนี้ ผู้ใช้สามารถเลือกทดสอบการมอดูเลชันโดยใช้ซอฟต์แวร์หรืออุปกรณ์ FPGA

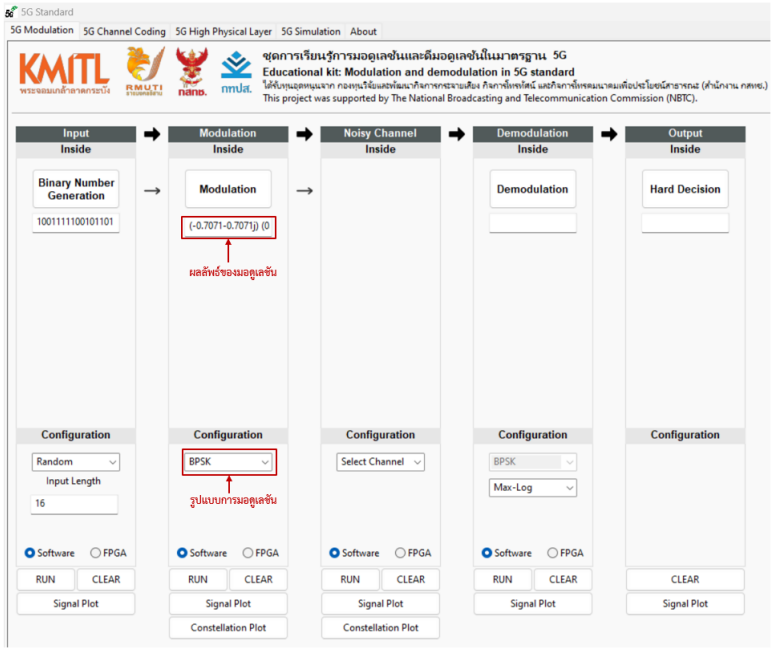
4.1.4.1 ผลการทดสอบ GUI สำหรับชุดการเรียนรู้มอดูเลชันและดีมอดูเลชันมาตรฐาน 5G

- กรณีใช้ซอฟต์แวร์ทดสอบการมอดูเลชัน

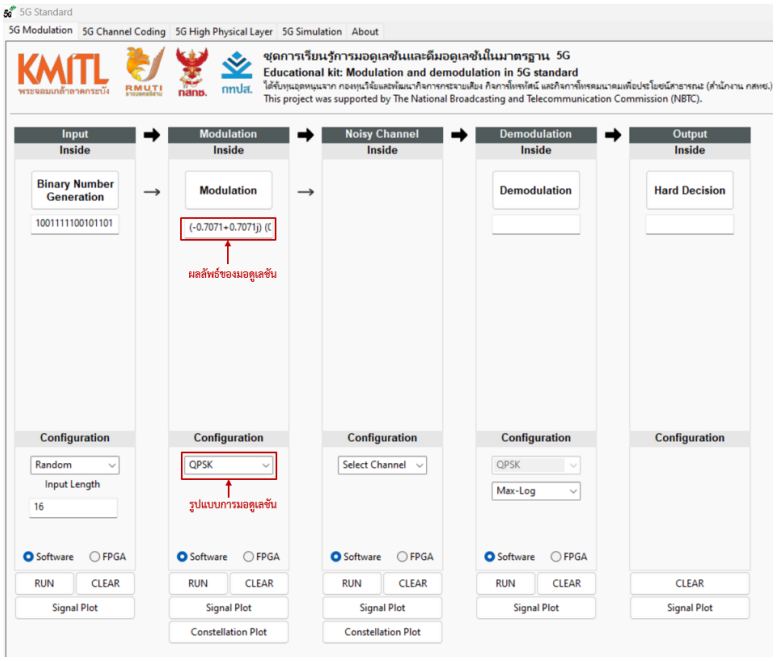
การทดสอบการมอดูเลชันโดยใช้ซอฟต์แวร์จะกำหนดให้ข้อมูลอินพุตมีความยาว 16 บิต ซึ่งได้จากการทดสอบสร้างข้อมูลอินพุตแบบสุ่มโดยใช้ซอฟต์แวร์ และเลือกการมอดูเลชัน จากนั้นเมื่อกดปุ่ม “RUN” ซอฟต์แวร์ GUI จะเรียกใช้ซอฟต์แวร์ Module modulation.py เพื่อดำเนินการมอดูเลชันและแสดงผลลัพธ์ที่ได้ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.53 - รูปที่ 4.59 โดยผลลัพธ์ดังกล่าวจะแสดงอยู่ในรูปของตัวเลขเชิงซ้อน (Complex Number) นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟและแผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านการมอดูเลชันได้โดยกดปุ่ม “Signal Plot” และ “Constellation Plot” ตามลำดับ ตัวอย่างกราฟและแผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านการมอดูเลชันซึ่งทดสอบโดยใช้ซอฟต์แวร์ แสดงดังรูปที่ 4.60 - รูปที่ 4.66 และรูปที่ 4.67 - รูปที่ 4.73 ตามลำดับ



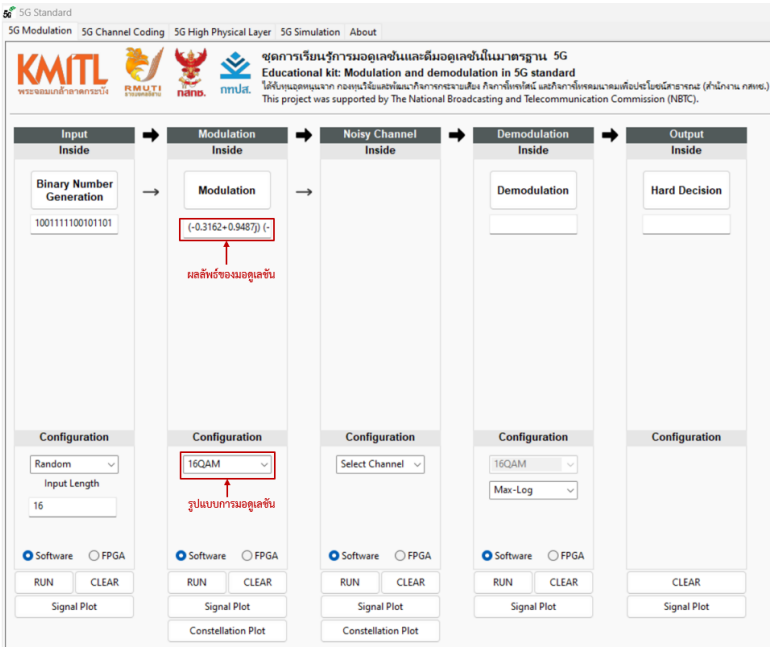
รูปที่ 4.53 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ $\pi/2$-BPSK โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



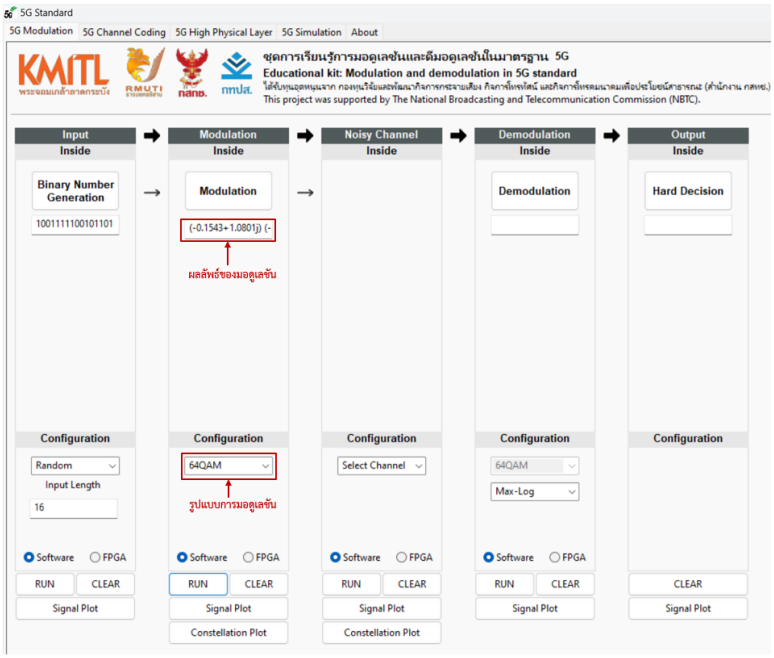
รูปที่ 4.54 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ BPSK โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



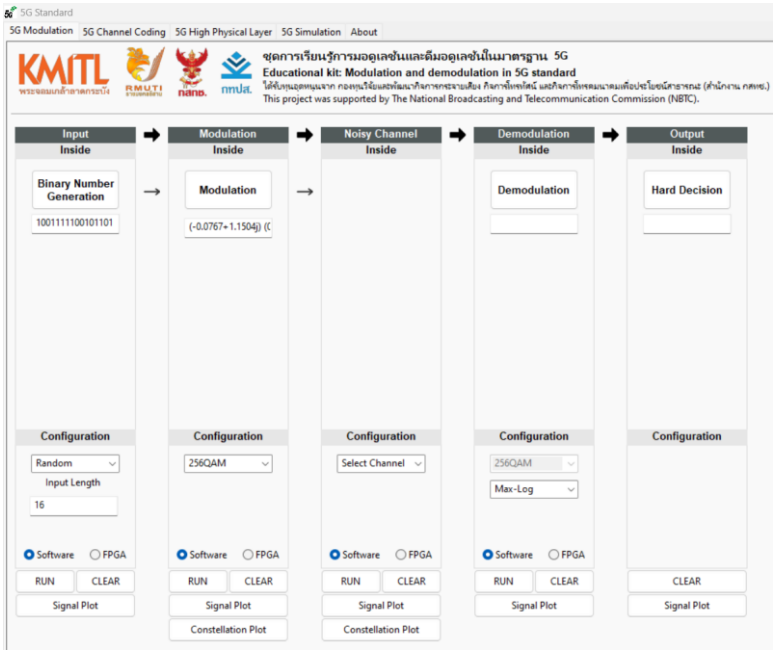
รูปที่ 4.55 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ QPSK โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



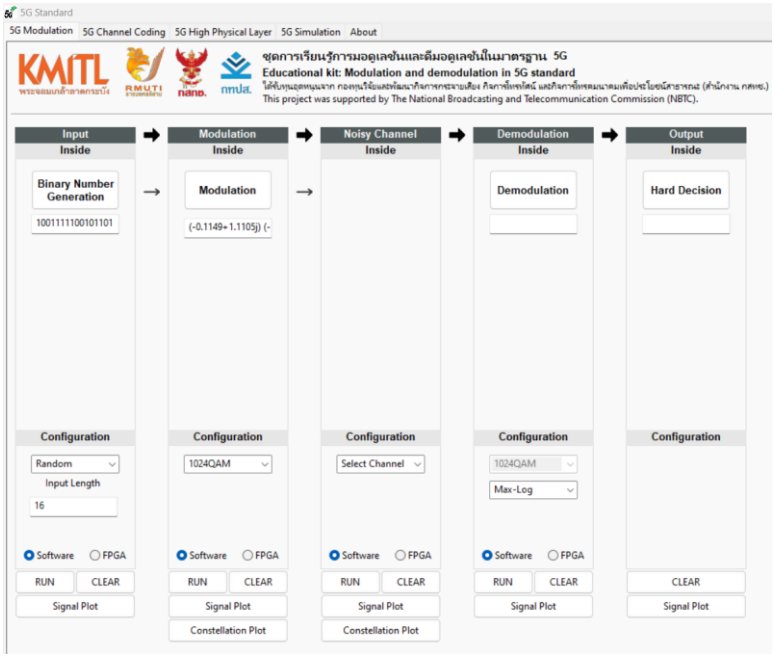
รูปที่ 4.56 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ 16QAM โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



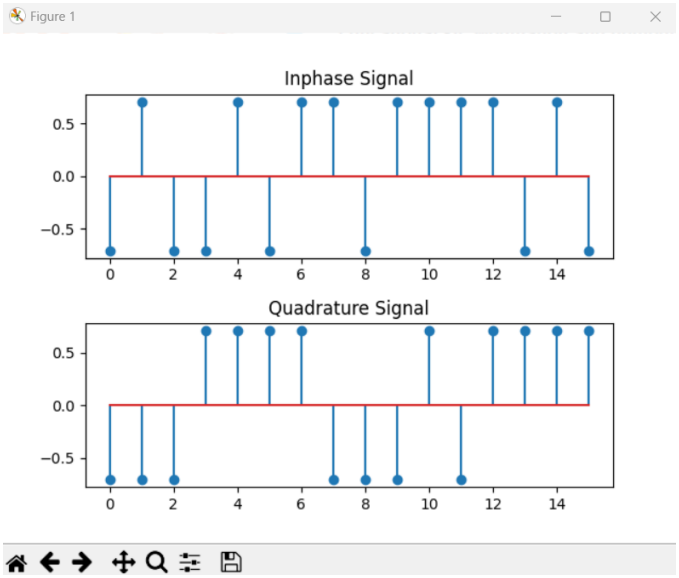
รูปที่ 4.57 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ 64QAM โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



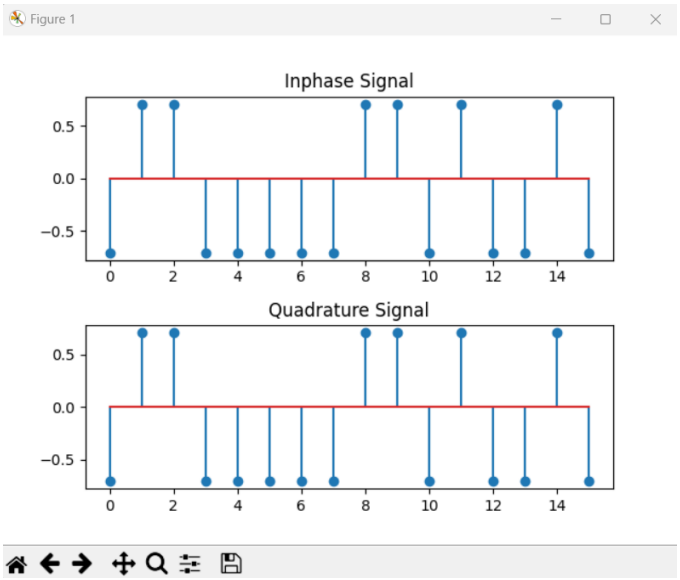
รูปที่ 4.58 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ 256QAM โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



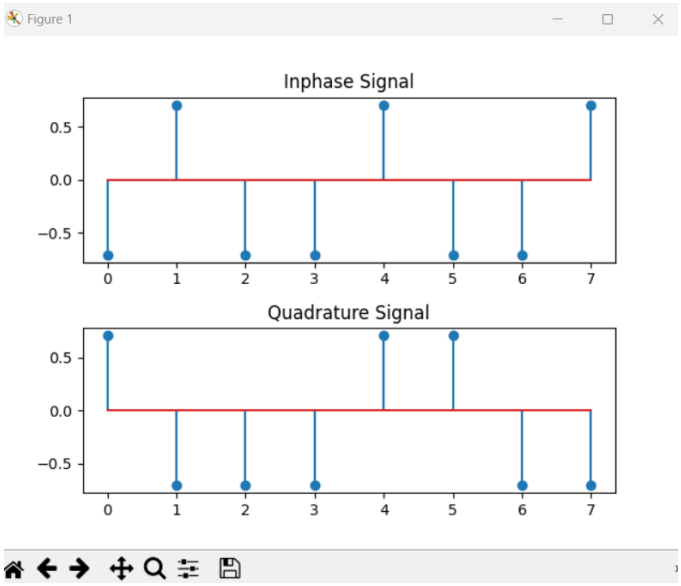
รูปที่ 4.59 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ 1024QAM โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



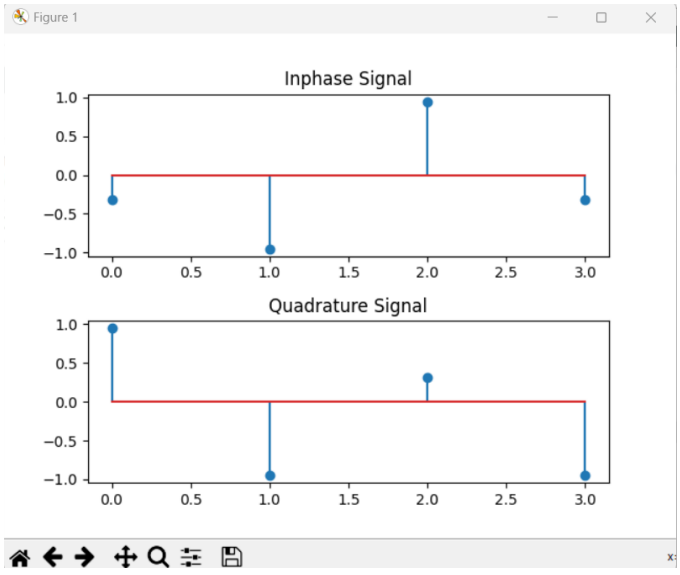
รูปที่ 4.60 กราฟของสัญญาณมอดูเลชันแบบ π /2-BPSK ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



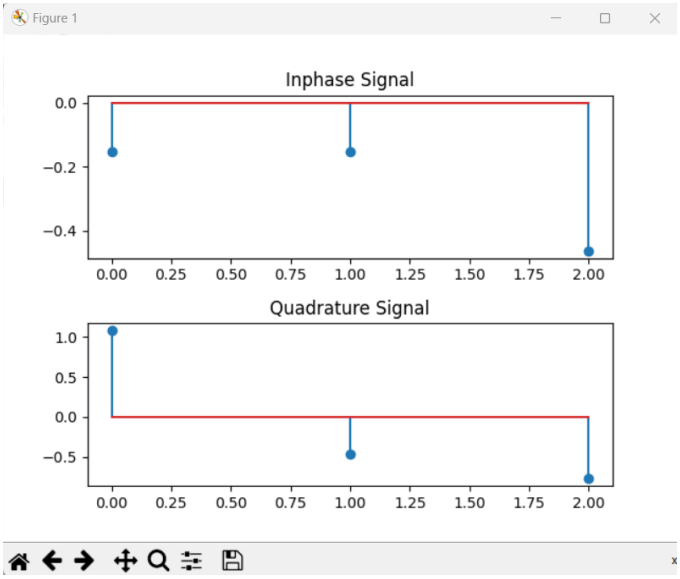
รูปที่ 4.61 กราฟของสัญญาณมอดูเลชันแบบ BPSK ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



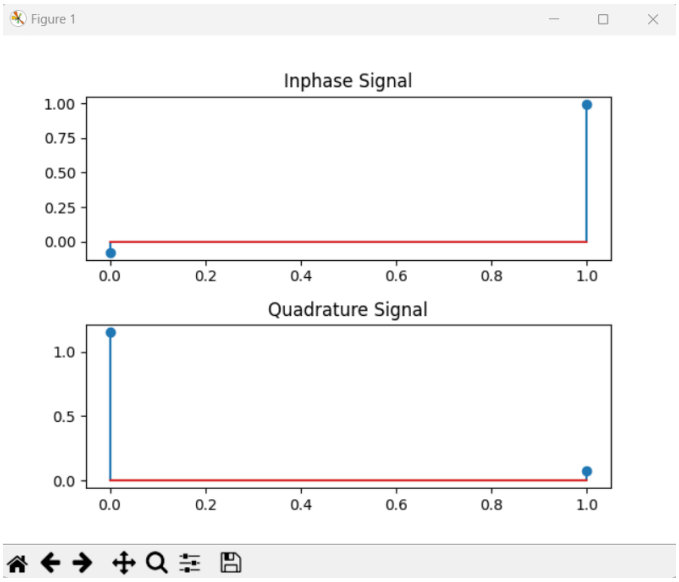
รูปที่ 4.62 กราฟของสัญญาณมอดูเลชันแบบ QPSK ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



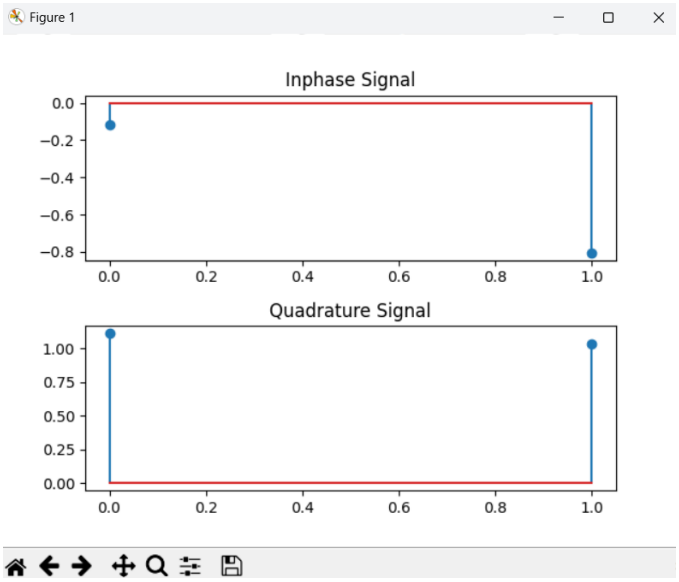
รูปที่ 4.63 กราฟของสัญญาณมอดูเลชันแบบ 16QAM ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



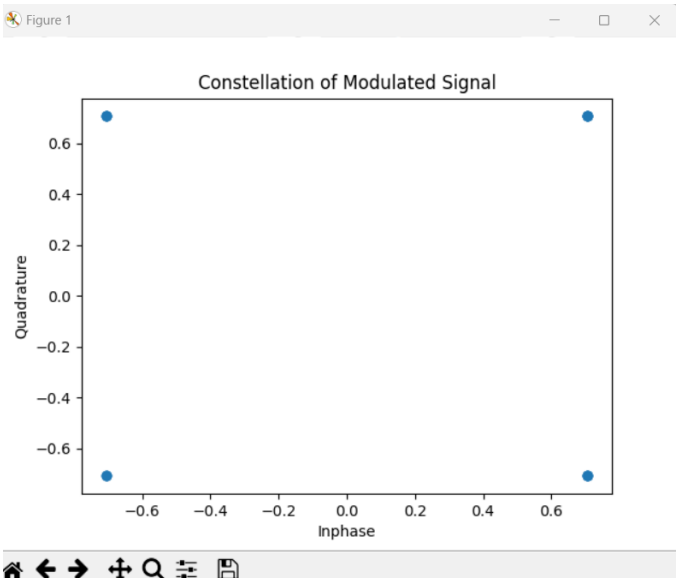
รูปที่ 4.64 กราฟของสัญญาณมอดูเลชันแบบ 64QAM ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



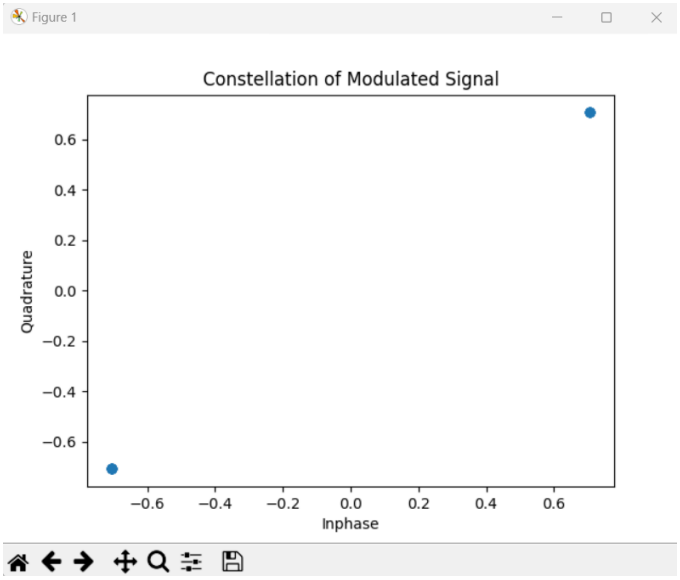
รูปที่ 4.65 กราฟของสัญญาณมอดูเลชันแบบ 256QAM ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



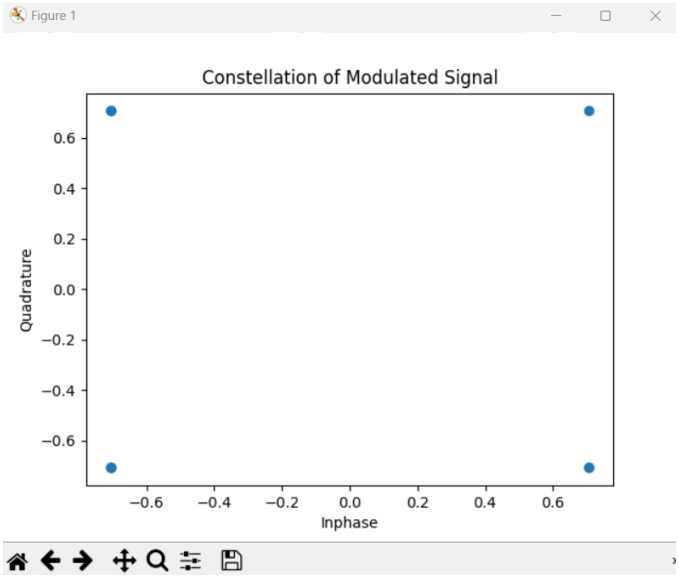
รูปที่ 4.66 กราฟของสัญญาณมอดูเลชันแบบ 1024QAM ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



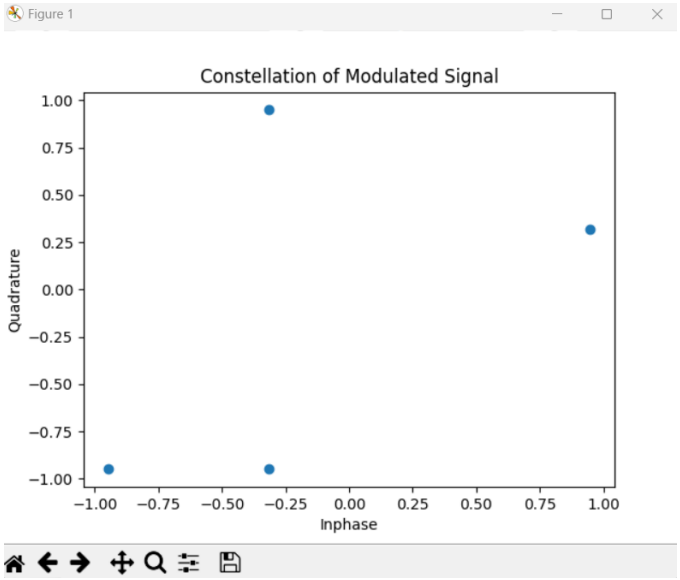
รูปที่ 4.67 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ π/2-BPSK ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



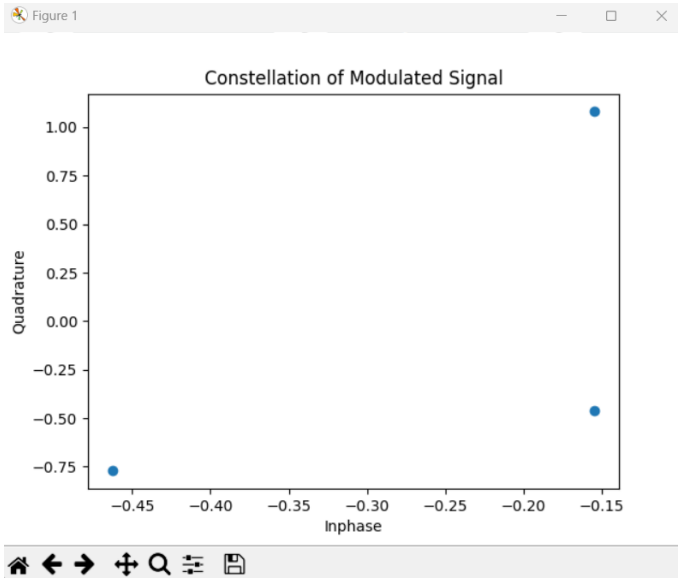
รูปที่ 4.68 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ BPSK ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



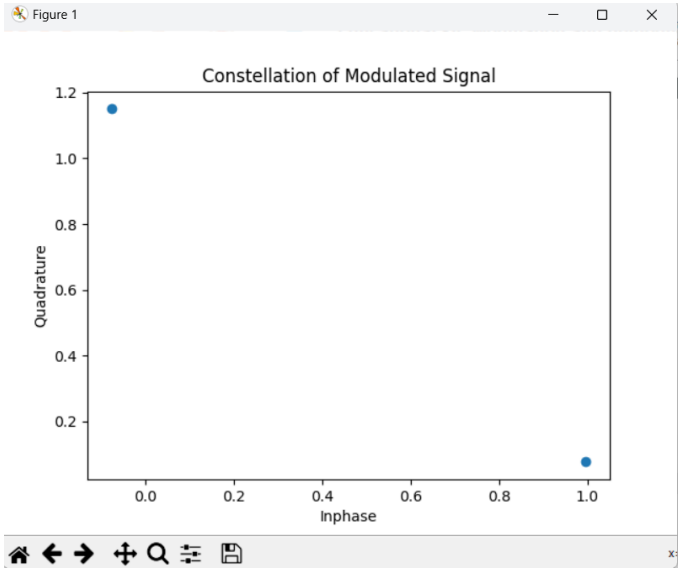
รูปที่ 4.69 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ QPSK ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



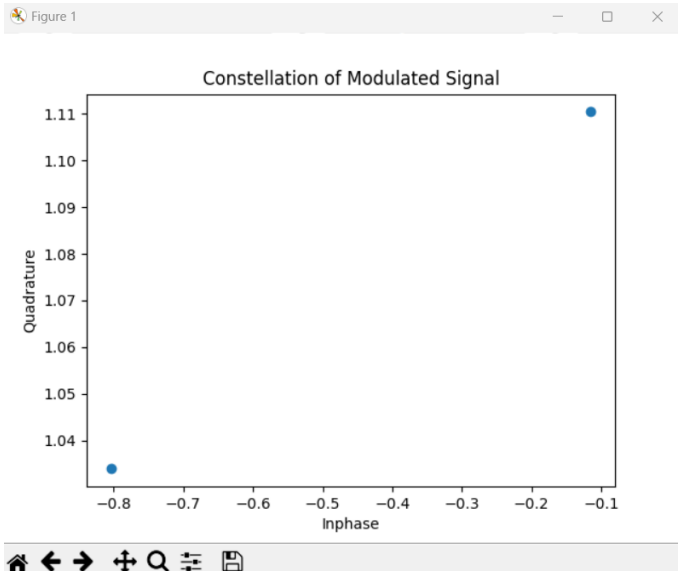
รูปที่ 4.70 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ 16QAM ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



รูปที่ 4.71 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ 64QAM ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



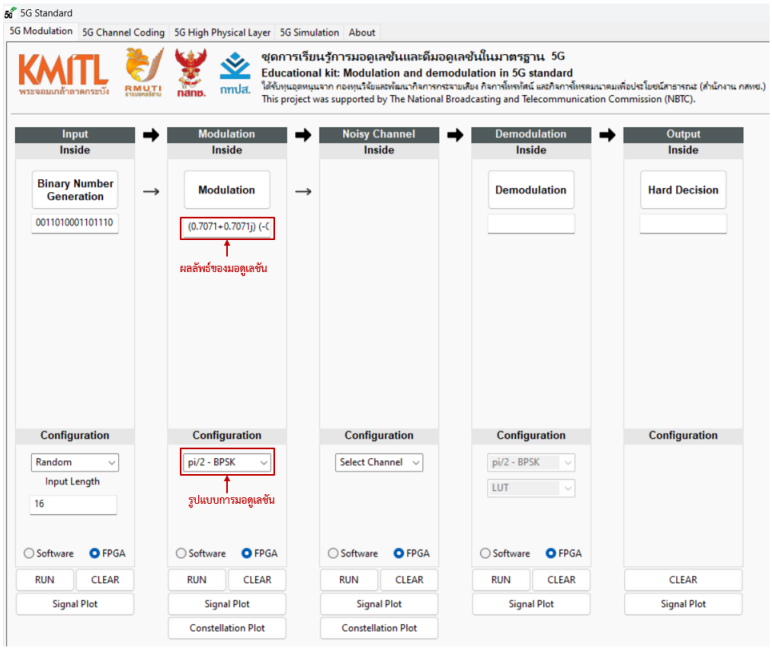
รูปที่ 4.72 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ 256QAM ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



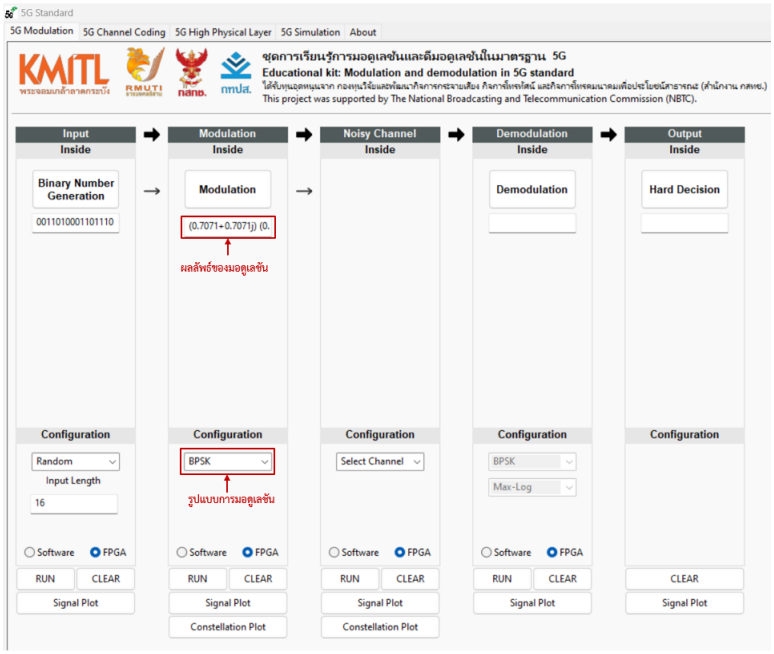
รูปที่ 4.73 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ 1024QAM ที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G

- กรณีใช้อุปกรณ์ FPGA ทดสอบการมอดูเลชัน

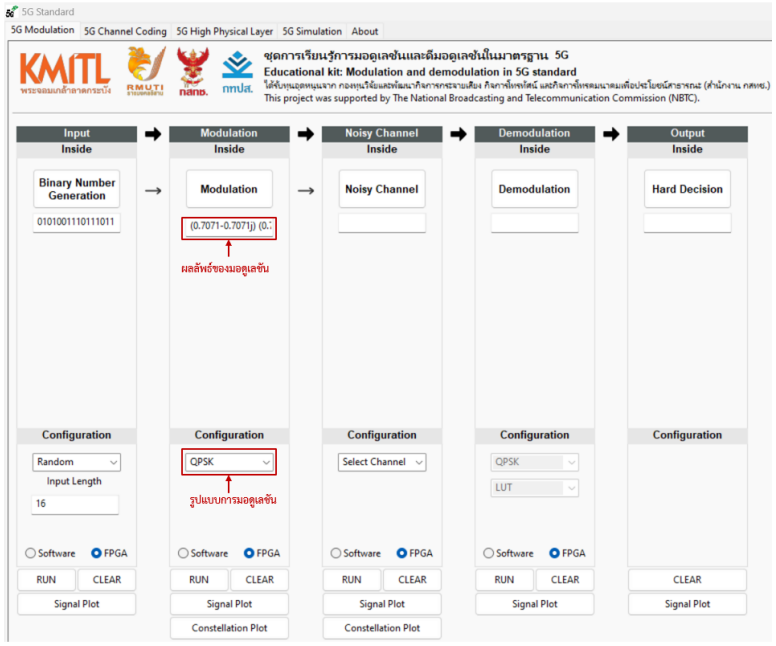
การทดสอบการมอดูเลชันโดยใช้อุปกรณ์ FPGA จะกำหนดให้อินพุตมีความยาว 16 บิต ซึ่งได้จากการทดสอบสร้างข้อมูลอินพุตแบบสุ่มโดยใช้อุปกรณ์ FPGA และเลือกการมอดูเลชัน เมื่อกดปุ่ม “RUN” ซอฟต์แวร์ GUI จะส่งข้อมูลอินพุตและรูปแบบการมอดูเลชันผ่าน Serial Port ไปยังอุปกรณ์ FPGA จากนั้นอุปกรณ์ FPGA จะดำเนินการมอดูเลชันและส่งผลลัพธ์ผ่าน Serial Port กลับมาแสดงบน GUI ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.74 - รูปที่ 4.80 โดยผลลัพธ์ดังกล่าวจะแสดงอยู่ในรูปของตัวเลขเชิงซ้อน นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟและแผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านการมอดูเลชันได้โดยกดปุ่ม “Signal Plot” และ “Constellation Plot” ตามลำดับ ตัวอย่างกราฟและแผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านการมอดูเลชันซึ่งทดสอบโดยใช้อุปกรณ์ FPGA แสดงดังรูปที่ 4.81 - รูปที่ 4.87 และ รูปที่ 4.88 - รูปที่ 4.94 ตามลำดับ



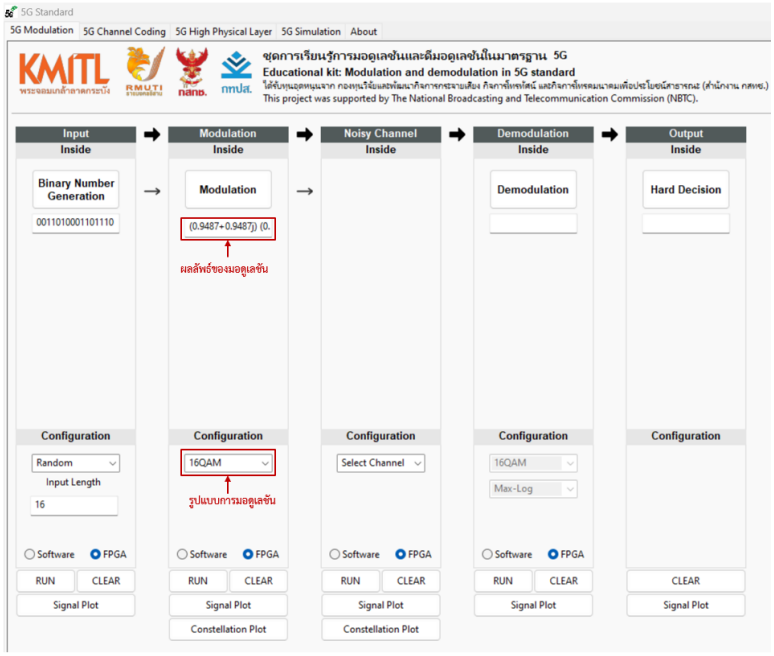
รูปที่ 4.74 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ $\pi/2$-BPSK โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



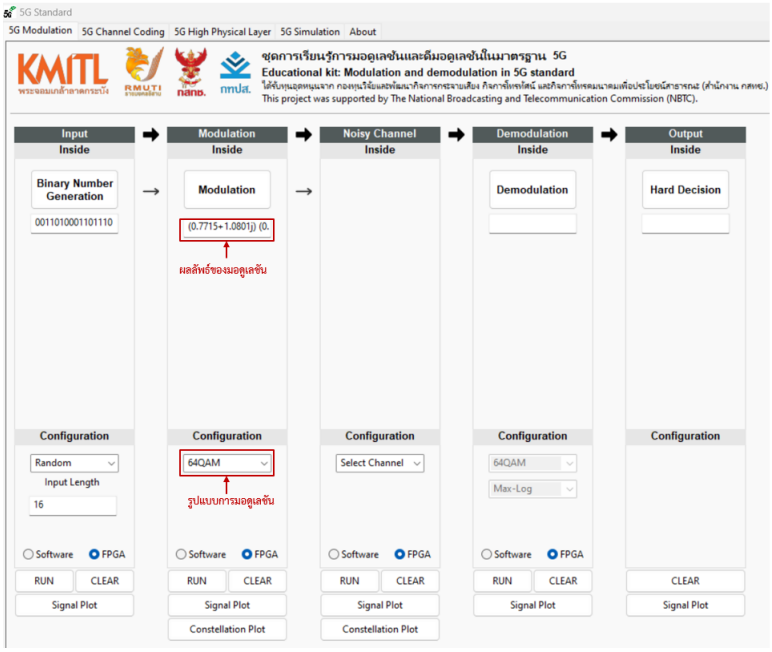
รูปที่ 4.75 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ BPSK โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



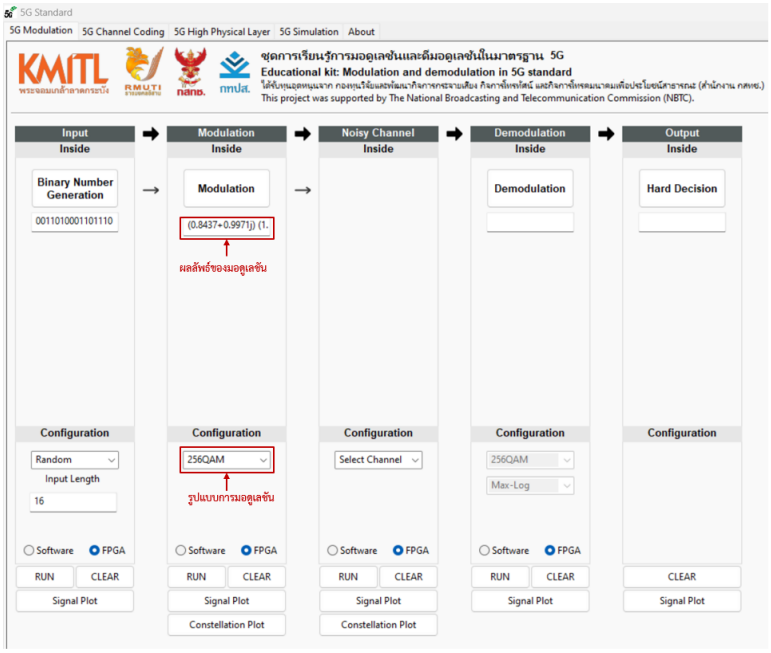
รูปที่ 4.76 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ QPSK โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



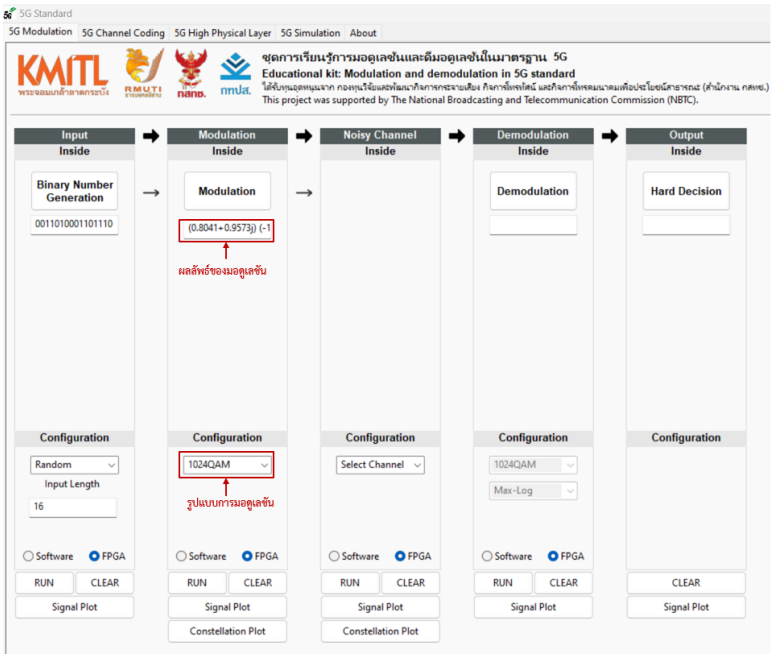
รูปที่ 4.77 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ 16QAM โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



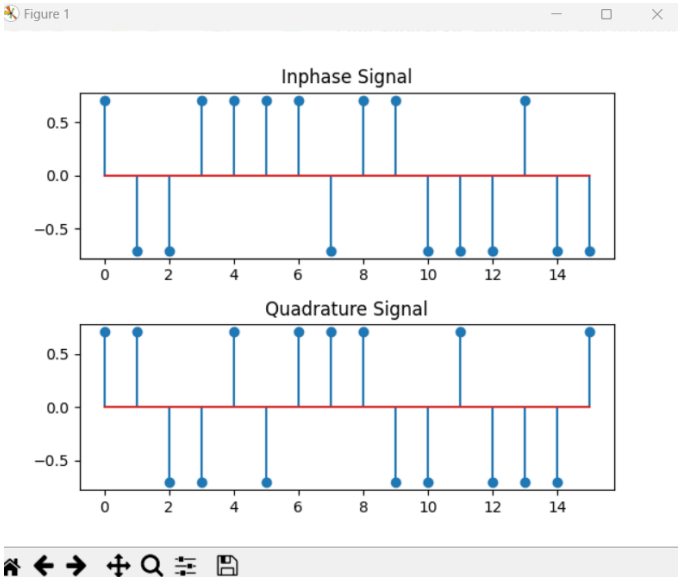
รูปที่ 4.78 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ 64QAM โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



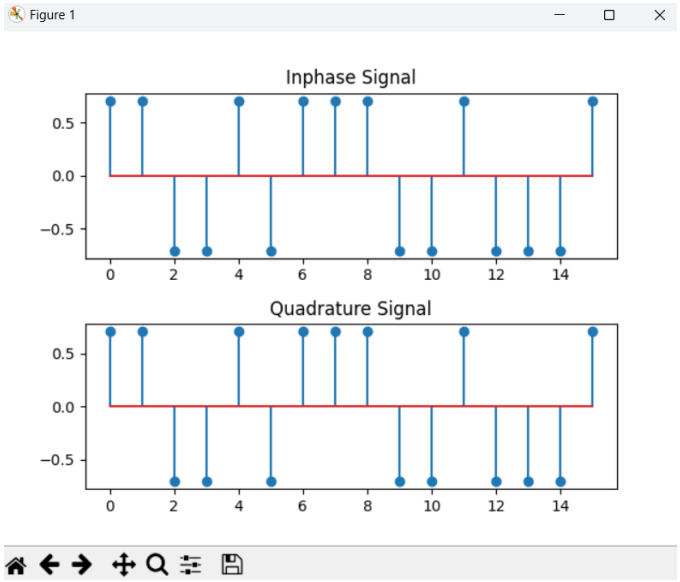
รูปที่ 4.79 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ 256QAM โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



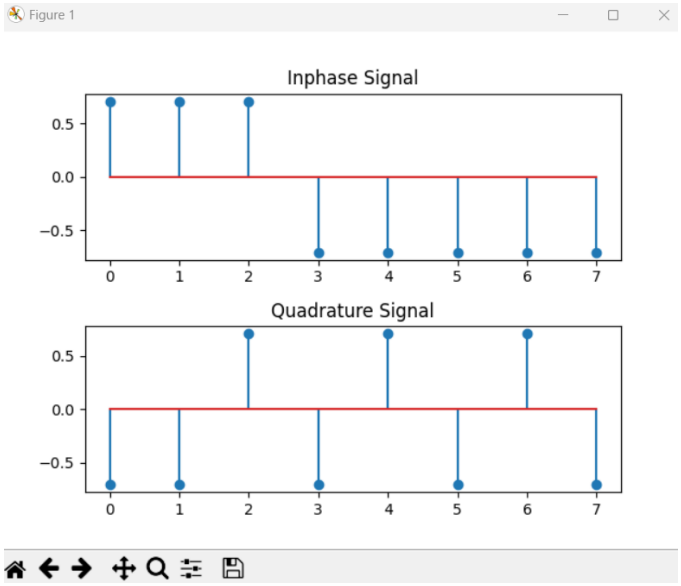
รูปที่ 4.80 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G แบบ 1024QAM โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



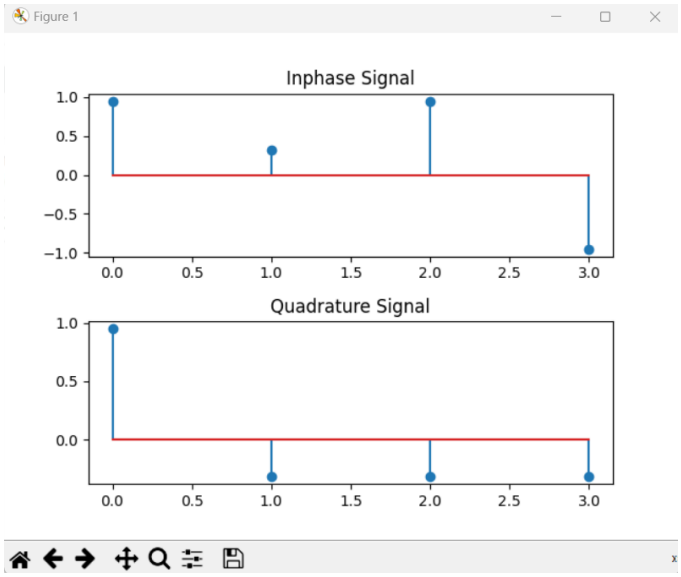
รูปที่ 4.81 กราฟของสัญญาณมอดูเลชันแบบ π/2-BPSK ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



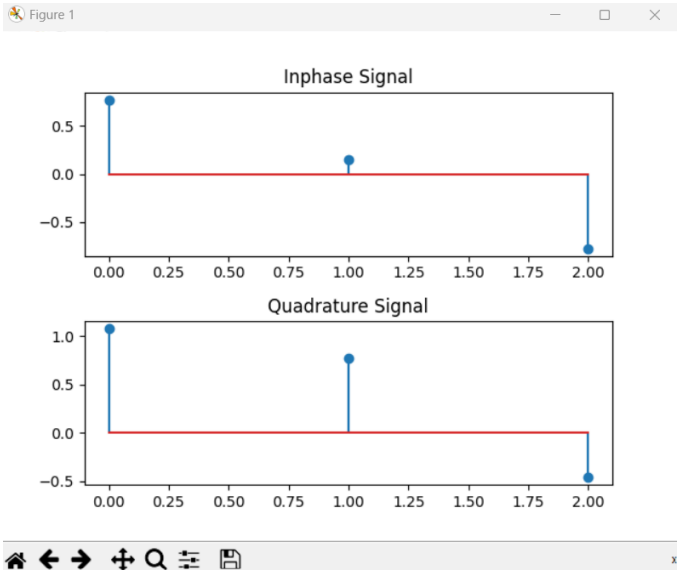
รูปที่ 4.82 กราฟของสัญญาณมอดูเลชันแบบ BPSK ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



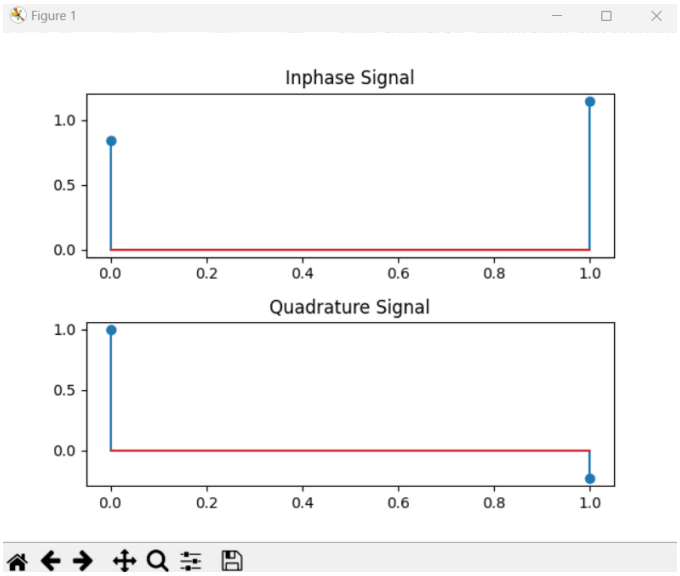
รูปที่ 4.83 กราฟของสัญญาณมอดูเลชันแบบ QPSK ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



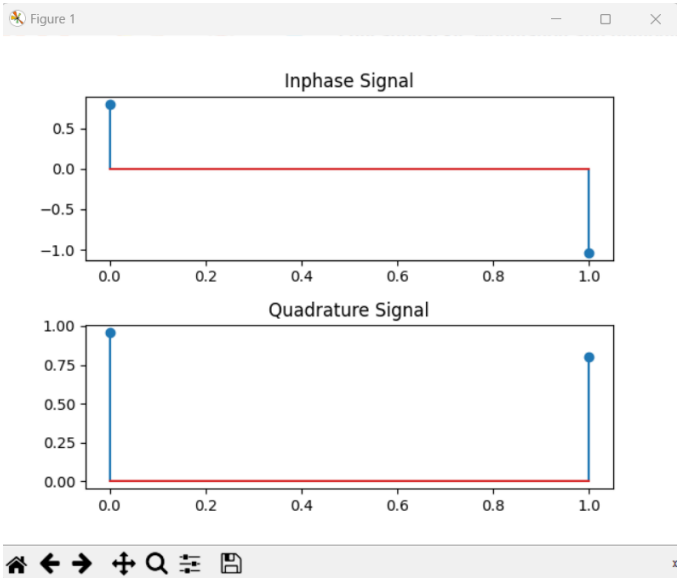
รูปที่ 4.84 กราฟของสัญญาณมอดูเลชันแบบ 16QAM ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



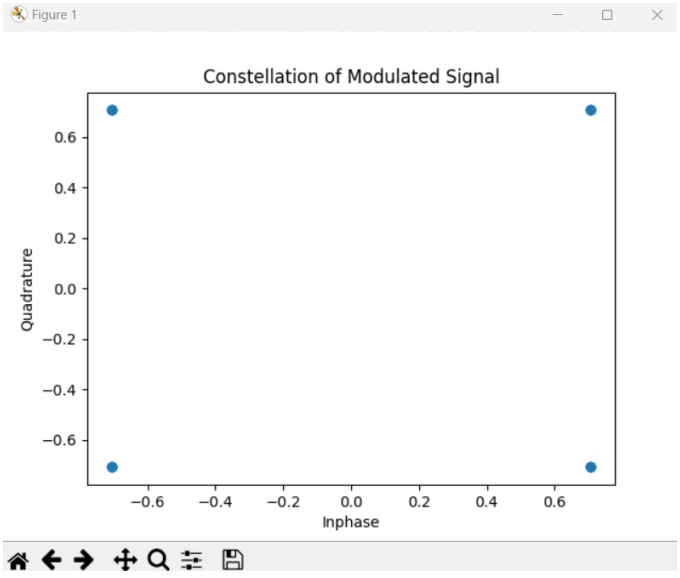
รูปที่ 4.85 กราฟของสัญญาณมอดูเลชันแบบ 64QAM ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



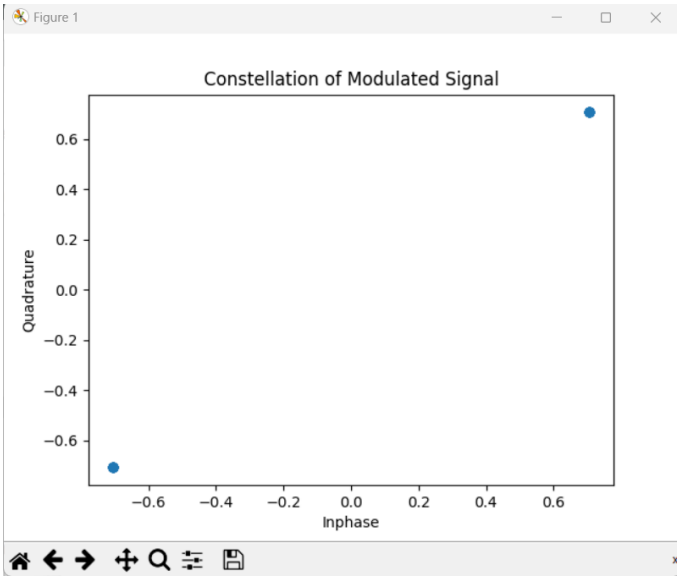
รูปที่ 4.86 กราฟของสัญญาณมอดูเลชันแบบ 256QAM ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



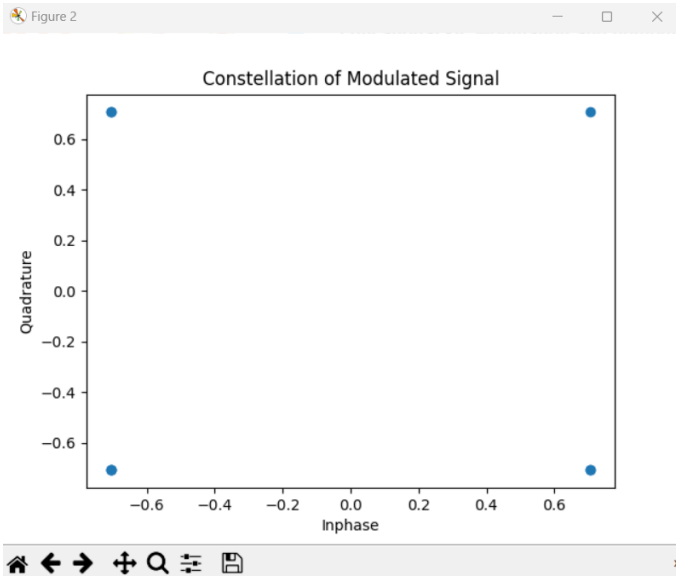
รูปที่ 4.87 กราฟของสัญญาณมอดูเลชันแบบ 1024QAM ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



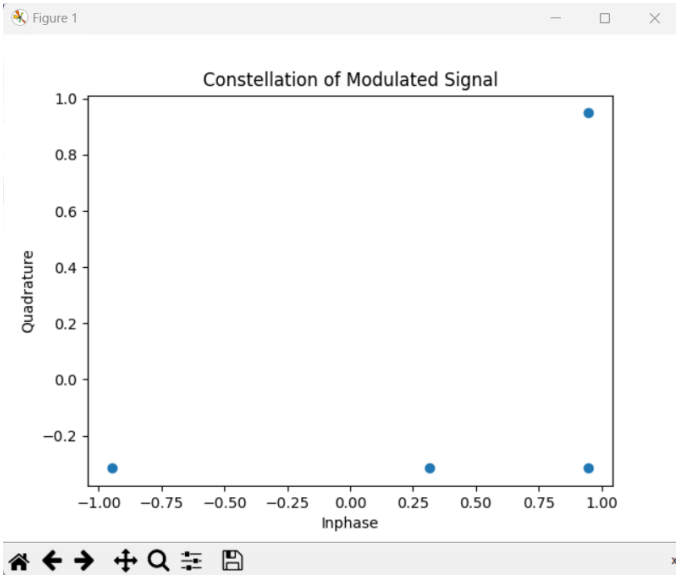
รูปที่ 4.88 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ $\pi/2$-BPSK ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



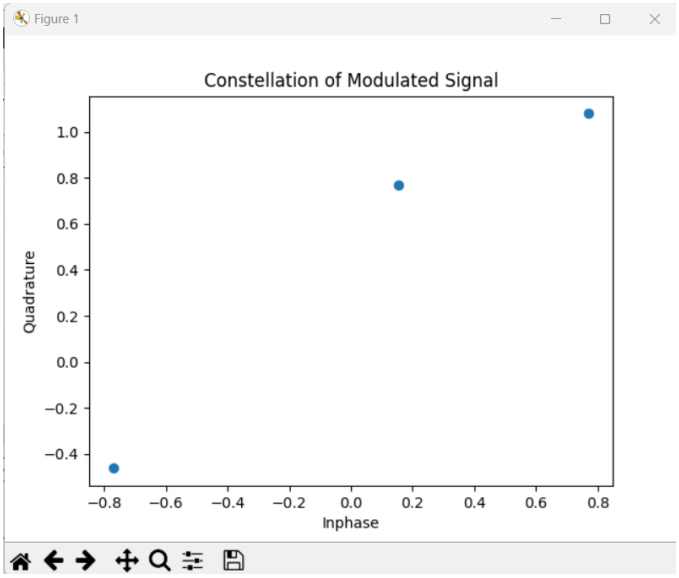
รูปที่ 4.89 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ BPSK ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



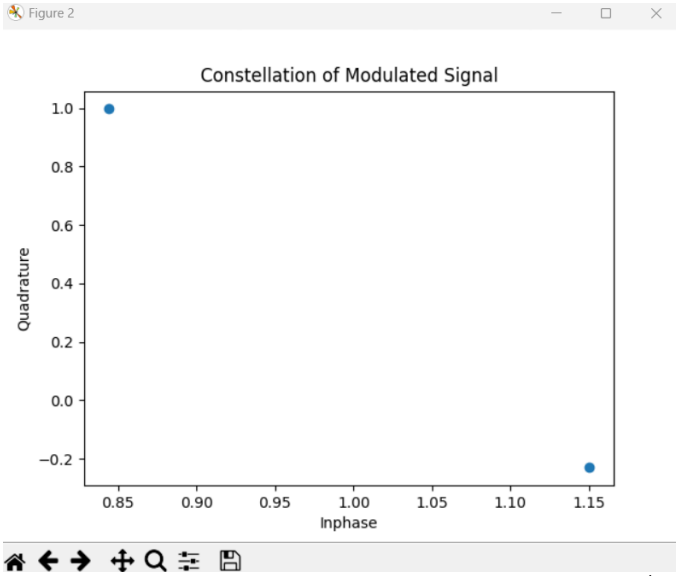
รูปที่ 4.90 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ QPSK ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



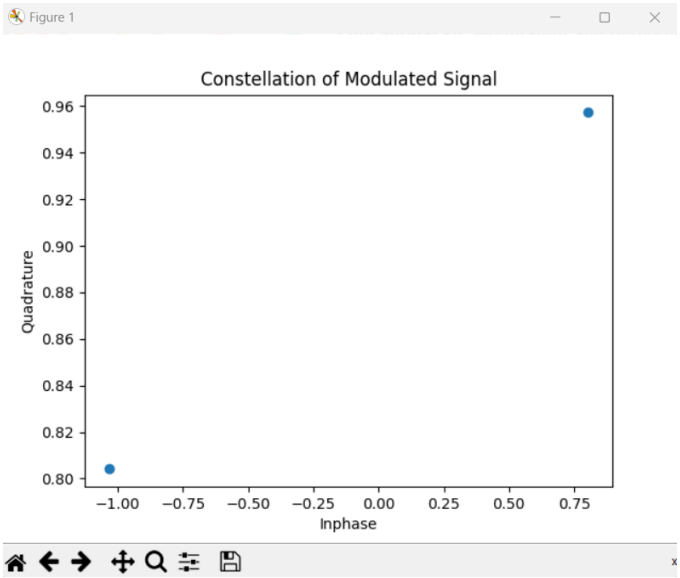
รูปที่ 4.91 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ 16QAM ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



รูปที่ 4.92 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ 64QAM ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



รูปที่ 4.93 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ 256QAM ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G

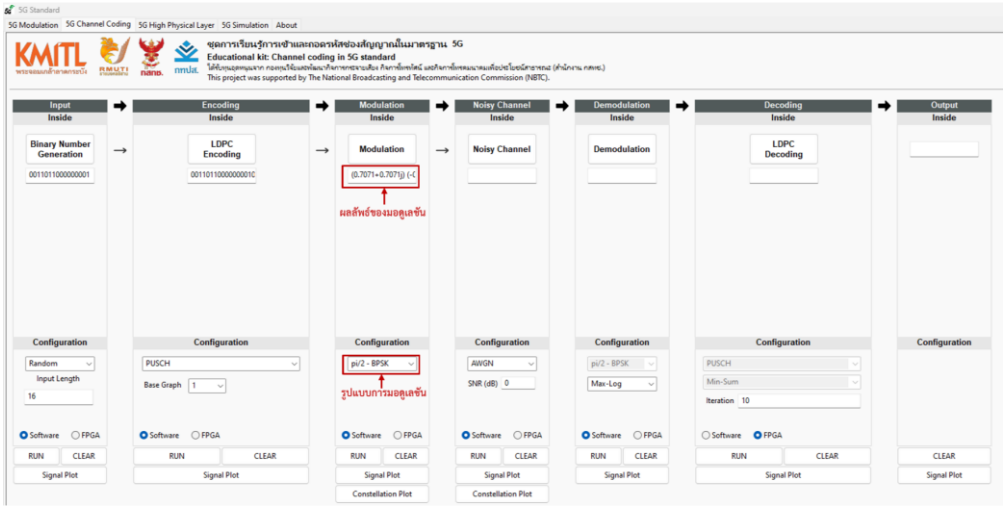


รูปที่ 4.94 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันแบบ 1024QAM ที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G

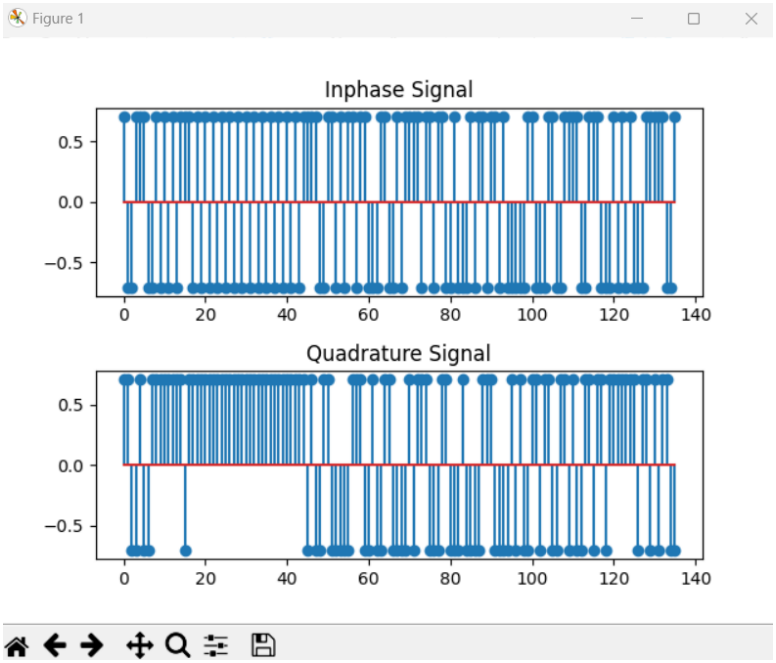
4.1.4.2 ผลการทดสอบ GUI สำหรับชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G

- กรณีใช้ซอฟต์แวร์ทดสอบการมอดูเลชัน

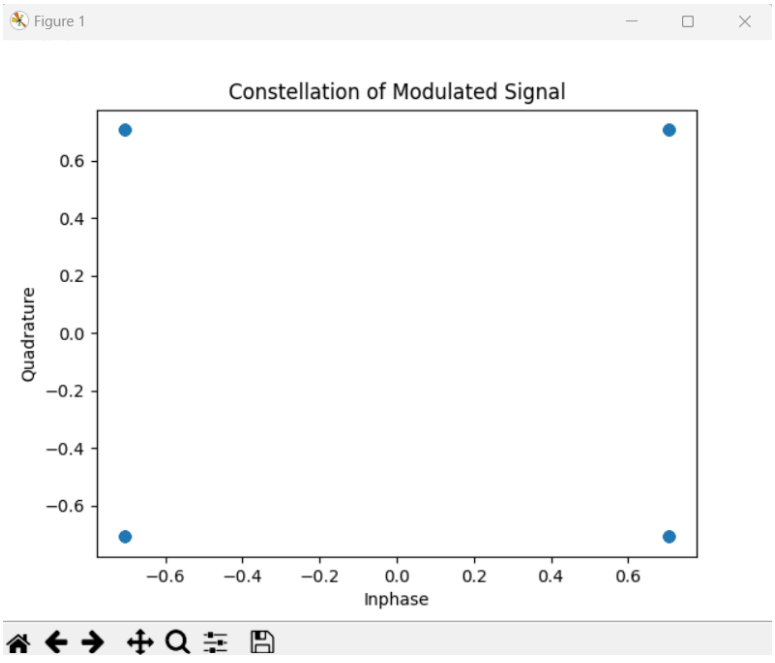
การทดสอบการมอดูเลชันโดยใช้ซอฟต์แวร์จะกำหนดให้รับอินพุตเป็นคำรหัสที่ได้จากส่วนการเข้ารหัสมาตรฐาน 5G และเลือกการมอดูเลชันแบบ $\pi/2$ - BPSK จากนั้นเมื่อกดปุ่ม “RUN” ซอฟต์แวร์ GUI จะเรียกใช้ซอฟต์แวร์ Module modulation.py เพื่อดำเนินการมอดูเลชันและแสดงผลลัพธ์ที่ได้ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.95 โดยผลลัพธ์ดังกล่าวจะแสดงอยู่ในรูปของตัวเลขเชิงซ้อน นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟและแผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านการมอดูเลชันได้โดยกดปุ่ม “Signal Plot” และ “Constellation Plot” ตามลำดับ ตัวอย่างกราฟและแผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านการมอดูเลชันซึ่งทดสอบโดยใช้ซอฟต์แวร์ แสดงดังรูปที่ 4.96 และรูปที่ 4.97 ตามลำดับ



รูปที่ 4.95 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



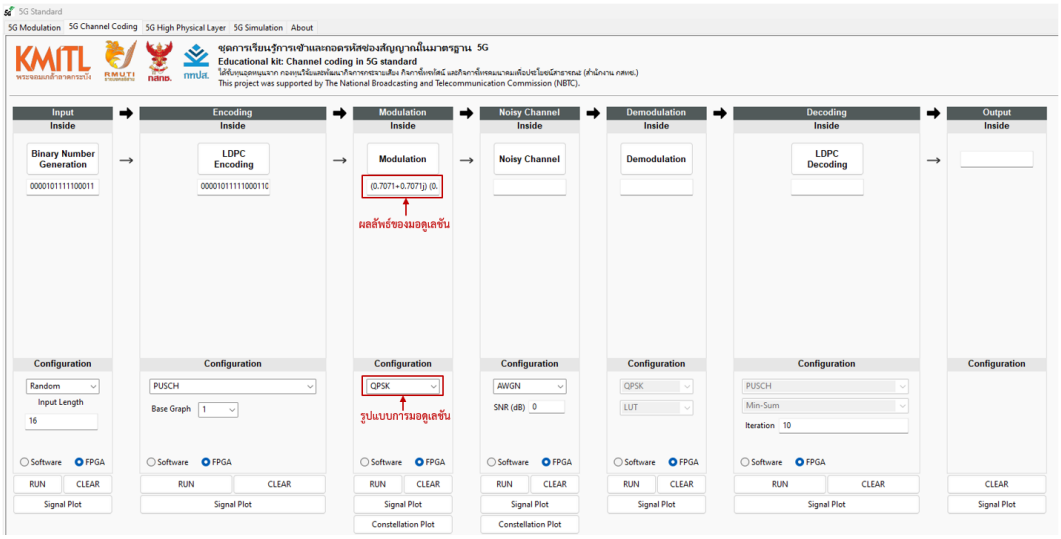
รูปที่ 4.96 กราฟของสัญญาณมอดูเลชันที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



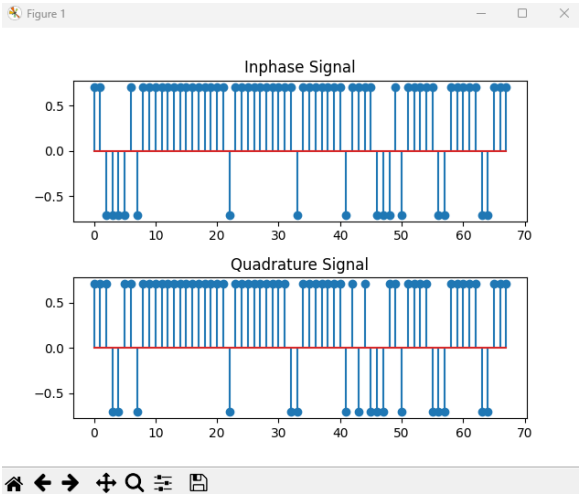
รูปที่ 4.97 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G

- กรณีใช้อุปกรณ์ FPGA ทดสอบการมอดูเลชัน

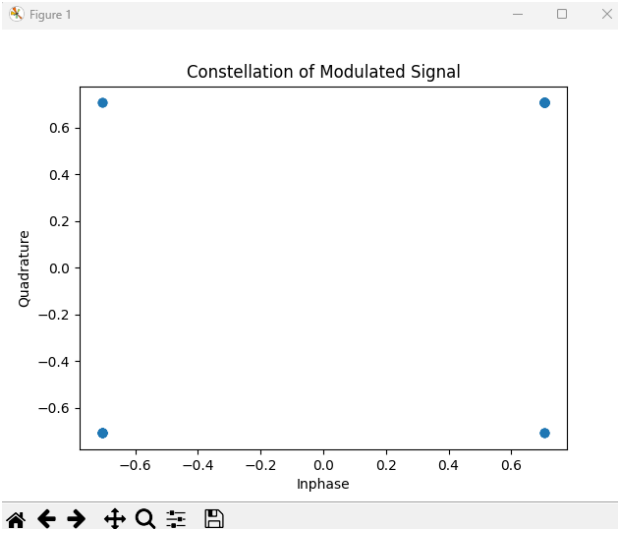
การทดสอบการมอดูเลชันโดยใช้อุปกรณ์ FPGA จะกำหนดให้รับอินพุตเป็นคำรหัสที่ได้จากส่วนการเข้ารหัสมาตรฐาน 5G และเลือกการมอดูเลชันแบบ QPSK เมื่อกดปุ่ม “RUN” ซอฟต์แวร์ GUI จะส่งข้อมูลอินพุตและรูปแบบการมอดูเลชันผ่าน Serial Port ไปยังอุปกรณ์ FPGA จากนั้นอุปกรณ์ FPGA จะดำเนินการมอดูเลชันและส่งผลลัพธ์ผ่าน Serial Port กลับมาแสดงบน GUI ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.98 โดยผลลัพธ์ดังกล่าวจะแสดงอยู่ในรูปของตัวเลขเชิงซ้อน นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟและแผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านการมอดูเลชันได้โดยกดปุ่ม “Signal Plot” และ “Constellation Plot” ตามลำดับ ตัวอย่างกราฟและแผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านการมอดูเลชันซึ่งทดสอบโดยใช้อุปกรณ์ FPGA แสดงดังรูปที่ 4.99 และรูปที่ 4.100 ตามลำดับ



รูปที่ 4.98 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



รูปที่ 4.99 กราฟของสัญญาณมอดูเลชันที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G

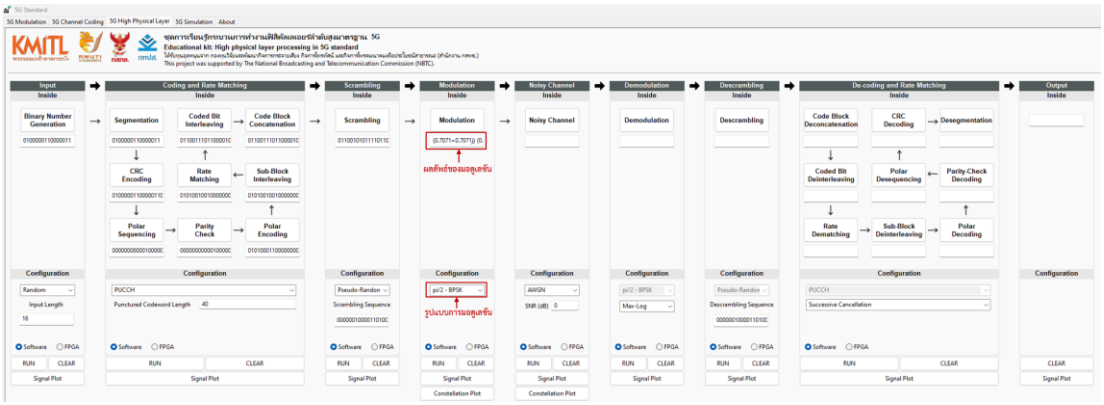


รูปที่ 4.100 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G

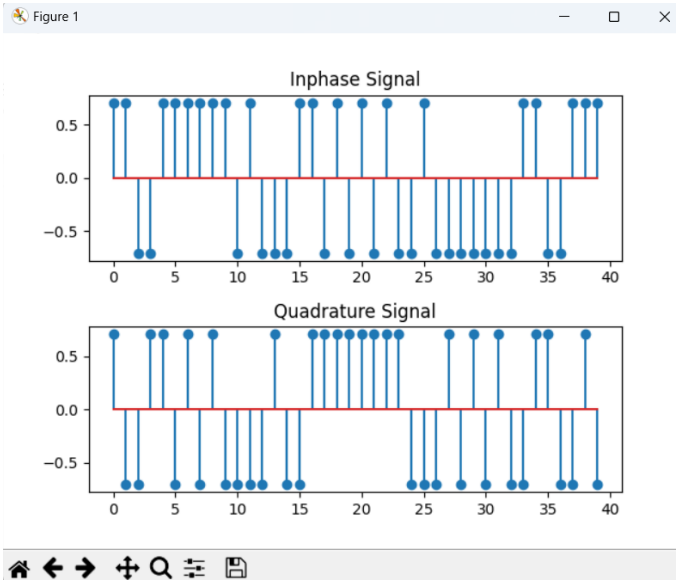
4.1.4.3 ผลการทดสอบ GUI สำหรับชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G

- กรณีใช้ซอฟต์แวร์ทดสอบการมอดูเลชัน

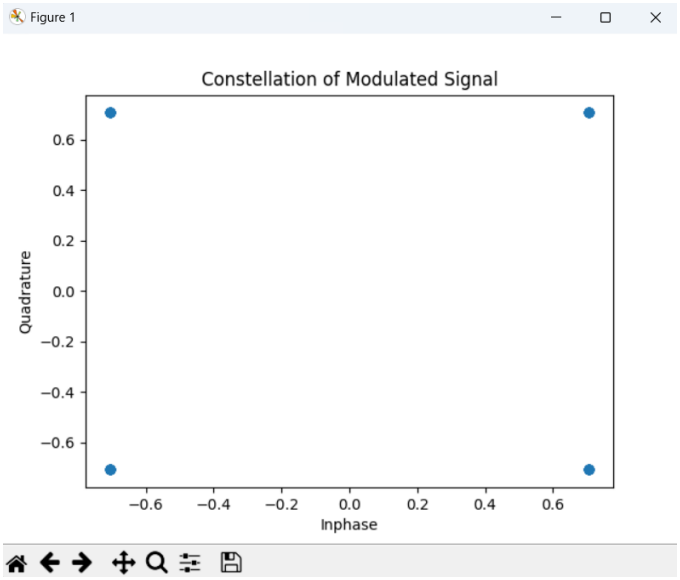
การทดสอบการมอดูเลชันโดยใช้ซอฟต์แวร์จะกำหนดให้รับอินพุตเป็นสัญญาณสแครมซึ่งได้จากส่วนการสแครมมาตรฐาน 5G และเลือกการมอดูเลชันแบบ $\pi/2$ - BPSK จากนั้นเมื่อกดปุ่ม “RUN” ซอฟต์แวร์ GUI จะเรียกใช้ซอฟต์แวร์ Module modulation.py เพื่อดำเนินการมอดูเลชันและแสดงผลลัพธ์ที่ได้ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูป รูปที่ 4.101 โดยผลลัพธ์ดังกล่าวจะแสดงอยู่ในรูปของตัวเลขเชิงซ้อน นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟและแผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านการมอดูเลชันได้โดยกดปุ่ม “Signal Plot” และ “Constellation Plot” ตามลำดับ ตัวอย่างกราฟและแผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านการมอดูเลชันซึ่งทดสอบโดยใช้ซอฟต์แวร์ แสดงดังรูปที่ 4.101รูปที่ 4.102 และรูปที่ 4.103 ตามลำดับ



รูปที่ 4.101 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G



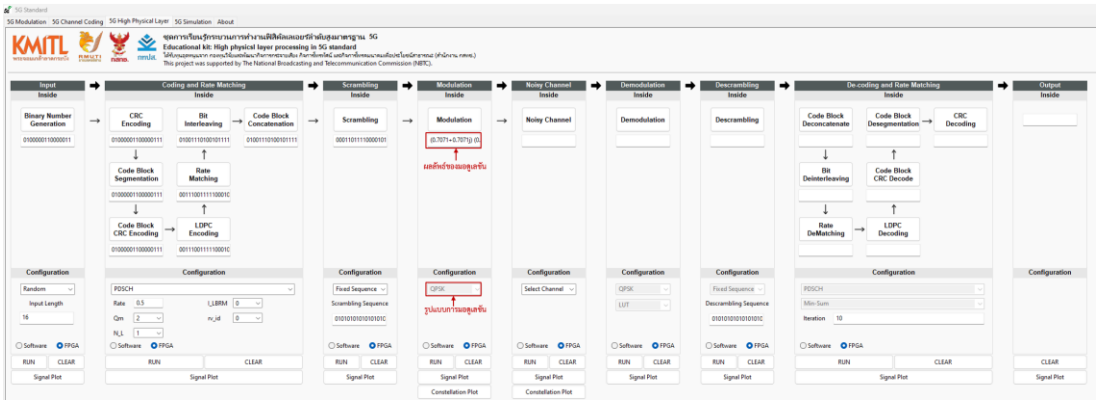
รูปที่ 4.102 กราฟของสัญญาณมอดูเลชันที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G



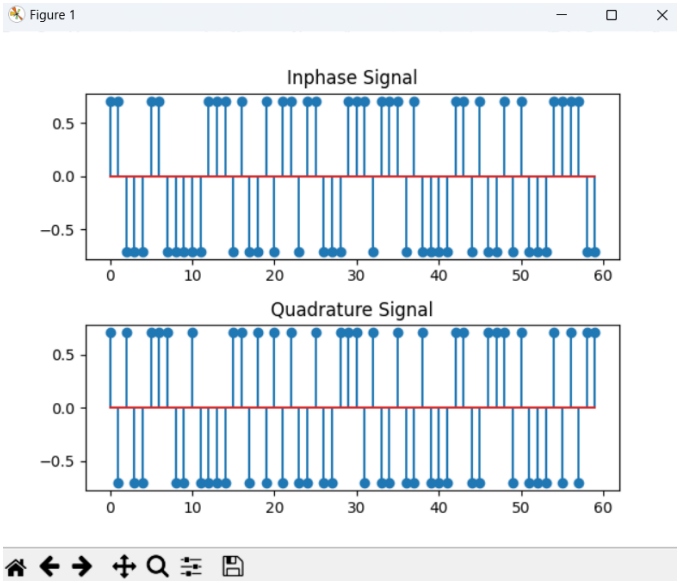
รูปที่ 4.103 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G

- กรณีใช้อุปกรณ์ FPGA ทดสอบการมอดูเลชัน

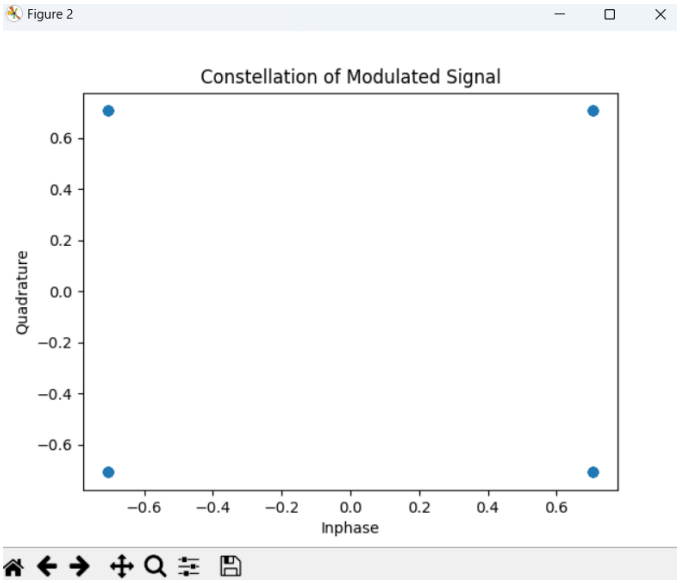
การทดสอบการมอดูเลชันโดยใช้อุปกรณ์ FPGA จะกำหนดให้รับอินพุตเป็นสัญญาณสแครม ซึ่งได้จากส่วนการสแครมมาตรฐาน 5G และเลือกการมอดูเลชันแบบ QPSK เมื่อกดปุ่ม “RUN” ซอฟต์แวร์ GUI จะส่งข้อมูลอินพุตและรูปแบบการมอดูเลชันผ่าน Serial Port ไปยังอุปกรณ์ FPGA จากนั้นอุปกรณ์ FPGA จะดำเนินการมอดูเลชันและส่งผลลัพธ์ผ่าน Serial Port กลับมาแสดงบน GUI ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.104 โดยผลลัพธ์ดังกล่าวจะแสดงอยู่ในรูปของตัวเลขเชิงซ้อน นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟและแผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านการมอดูเลชันได้โดยกดปุ่ม “Signal Plot” และ “Constellation Plot” ตามลำดับ ตัวอย่างกราฟและแผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านการมอดูเลชันซึ่งทดสอบโดยใช้อุปกรณ์ FPGA แสดงดังรูปที่รูปที่ 4.105 และรูปที่ 4.106 ตามลำดับ



รูปที่ 4.104 ผลลัพธ์การมอดูเลชันมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G



รูปที่ 4.105 กราฟของสัญญาณมอดูเลชันที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G



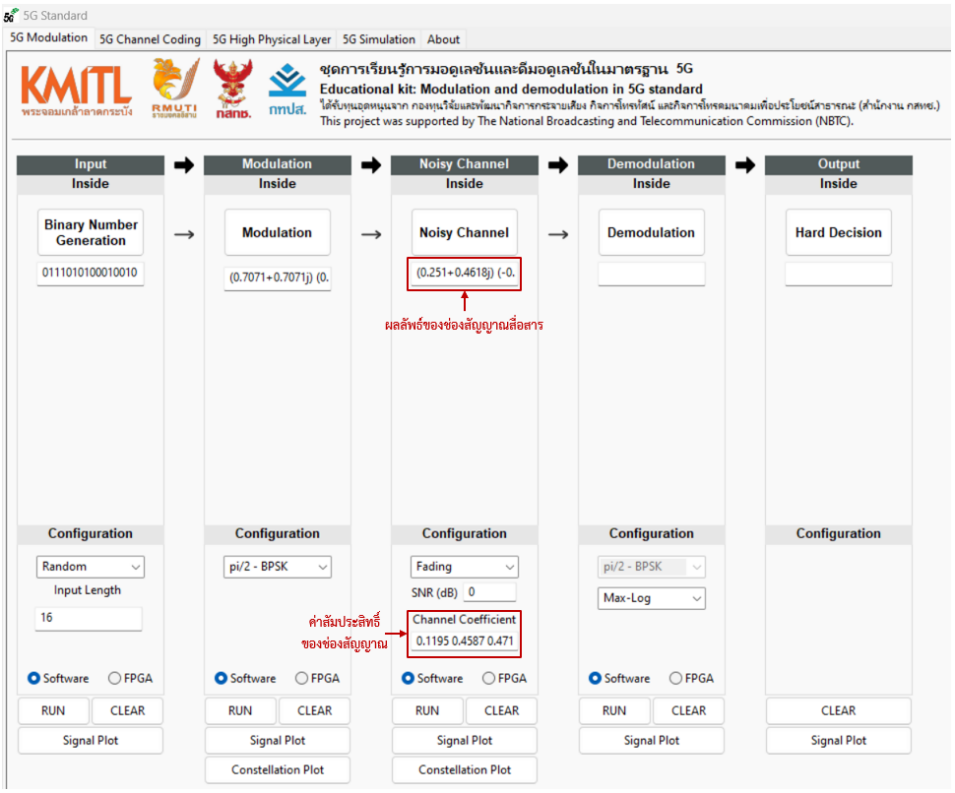
รูปที่ 4.106 แผนภาพคอนสเตลเลชันของสัญญาณมอดูเลชันที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G

4.1.5 ผลการทดสอบ GUI เพื่อแสดงผลของช่องสัญญาณสื่อสาร

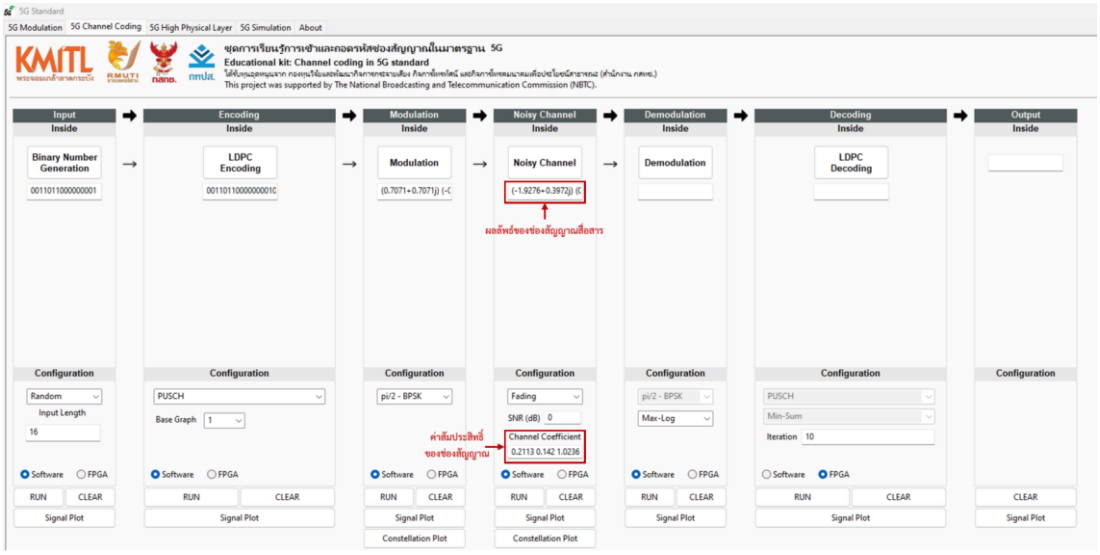
ช่องสัญญาณสื่อสารแสดงอยู่ในคอลัมน์ที่ 3 ของหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G คอลัมน์ที่ 4 ของหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G และคอลัมน์ที่ 5 ของหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G โดยผู้ใช้สามารถเลือกรูปแบบของช่องสัญญาณได้ทั้งหมด 2 รูปแบบ ได้แก่ AWGN Channel และ Fading Channel นอกจากนี้ ผู้ใช้สามารถเลือกทดสอบผลของช่องสัญญาณสื่อสารโดยใช้ซอฟต์แวร์หรืออุปกรณ์ FPGA

- กรณีใช้ซอฟต์แวร์ทดสอบผลของช่องสัญญาณสื่อสาร

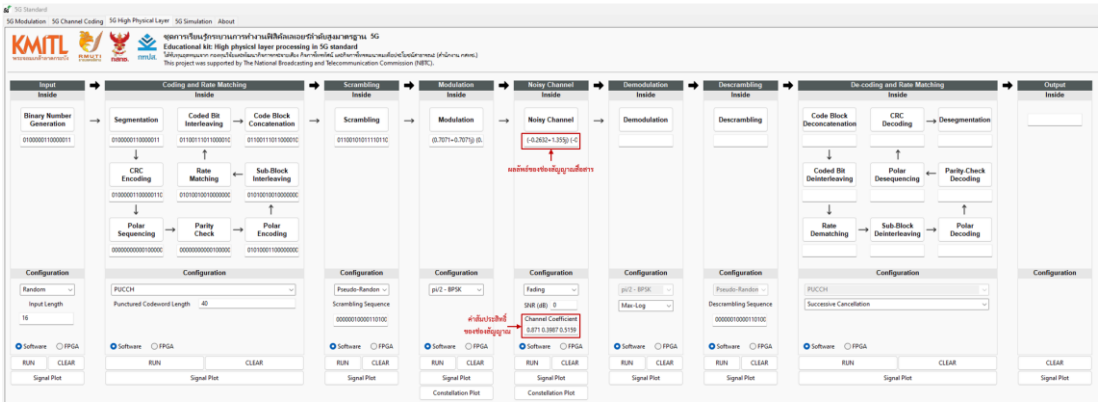
การทดสอบผลของช่องสัญญาณสื่อสารโดยใช้ซอฟต์แวร์จะกำหนดให้อินพุตเป็นสัญญาณที่ผ่านการมอดูเลชันแบบ $\pi/2$ - BPSK และเลือกช่องสัญญาณสื่อสารแบบ Fading Channel ที่ค่า SNR 0 dB จากนั้นเมื่อกดปุ่ม “RUN” ซอฟต์แวร์ GUI จะเรียกใช้ซอฟต์แวร์ Module Channel.py เพื่อดำเนินการสร้างค่าสัมประสิทธิ์ของช่องสัญญาณ (Channel Coefficient) และสัญญาณรบกวน (Noise) จากนั้นจึงรวมผลกระทบของช่องสัญญาณดังกล่าวเข้ากับสัญญาณมอดูเลชันและแสดงผลลัพธ์ที่ได้ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.107 - รูปที่ 4.109 โดยผลลัพธ์ดังกล่าวจะแสดงอยู่ในรูปของตัวเลขเชิงซ้อน นอกจากนี้ผู้ใช้สามารถเรียกดู กราฟและแผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านช่องสัญญาณสื่อสารได้โดยกดปุ่ม “Signal Plot” และ “Constellation Plot” ตัวอย่างกราฟและแผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านช่องสัญญาณสื่อสารซึ่งทดสอบโดยใช้ซอฟต์แวร์ แสดงดังรูปที่ 4.110 และรูปที่ 4.111 ตามลำดับ



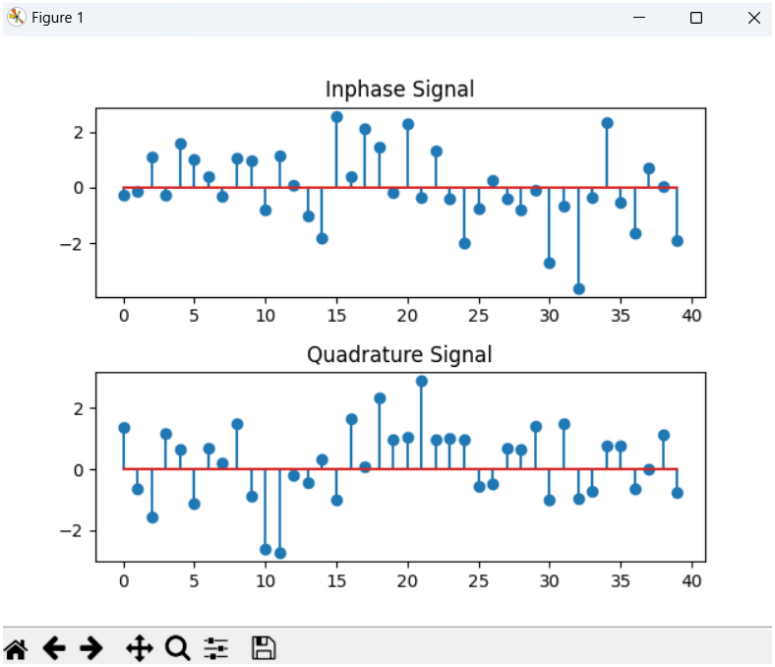
รูปที่ 4.107 ผลลัพธ์ทดสอบผลของช่องสัญญาณสื่อสารโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



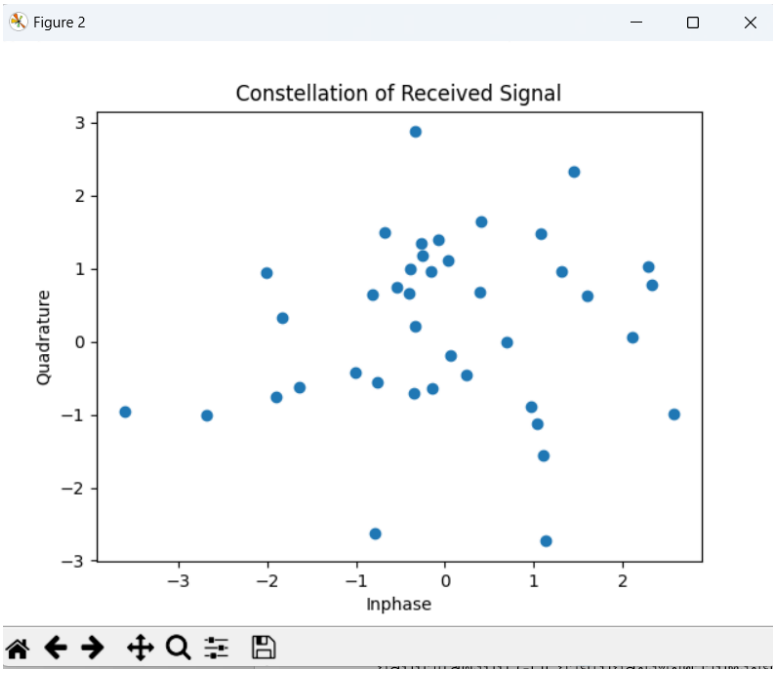
รูปที่ 4.108 ผลลัพธ์ทดสอบผลของช่องสัญญาณสื่อสารโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



รูปที่ 4.109 ผลลัพธ์ทดสอบผลของช่องสัญญาณสื่อสารโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G



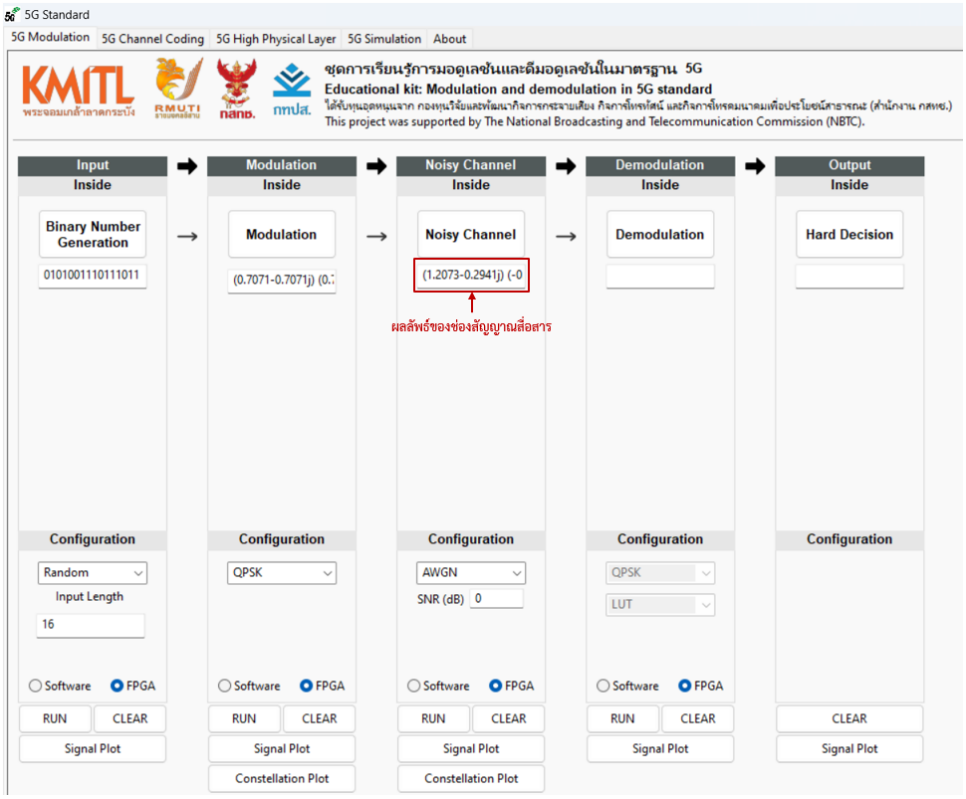
รูปที่ 4.110 กราฟของสัญญาณที่ผ่านช่องสัญญาณสื่อสารโดยใช้ซอฟต์แวร์



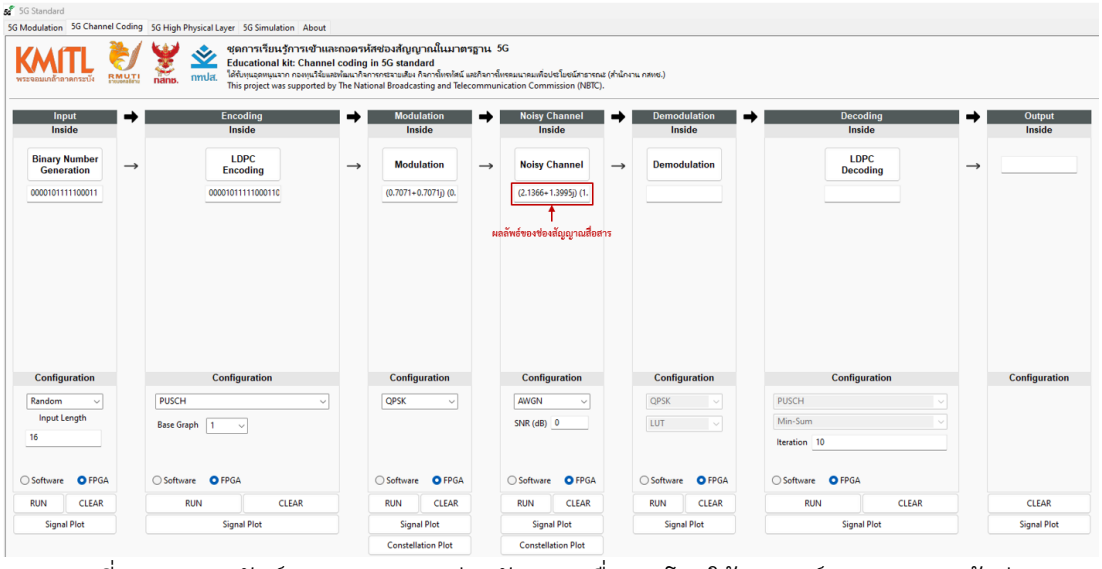
รูปที่ 4.111 แผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านช่องสัญญาณสื่อสารโดยใช้ซอฟต์แวร์

- กรณีใช้อุปกรณ์ FPGA ทดสอบการมอดูเลชัน

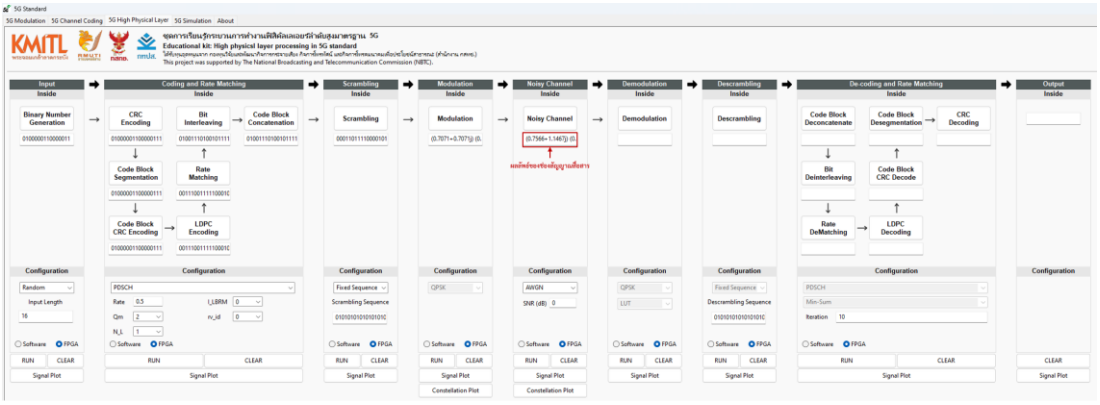
การทดสอบผลของช่องสัญญาณสื่อสารโดยใช้อุปกรณ์ FPGA จะกำหนดให้นพุตเป็นสัญญาณที่ผ่านการมอดูเลชันแบบ QPSK และเลือกช่องสัญญาณสื่อสารแบบ AWGN Channel เมื่อกดปุ่ม “RUN” ซอฟต์แวร์ GUI จะส่งสัญญาณมอดูเลชันและรูปแบบของช่องสัญญาณสื่อสารผ่าน Serial Port ไปยังอุปกรณ์ FPGA จากนั้นอุปกรณ์ FPGA จะดำเนินการรวมผลกระทบของช่องสัญญาณเข้ากับสัญญาณมอดูเลชันและส่งผลลัพธ์ผ่าน Serial Port กลับมาแสดงบน GUI ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.112 - รูปที่ 4.114 โดยผลลัพธ์ดังกล่าวจะแสดงอยู่ในรูปของตัวเลขเชิงซ้อน นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟและแผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านการมอดูเลชันได้โดยกดปุ่ม “Signal Plot” และ “Constellation Plot” ตามลำดับ ตัวอย่างกราฟและแผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านการมอดูเลชันซึ่งทดสอบโดยใช้อุปกรณ์ FPGA แสดงดังรูปที่ 4.115 และรูปที่ 4.116 ตามลำดับ



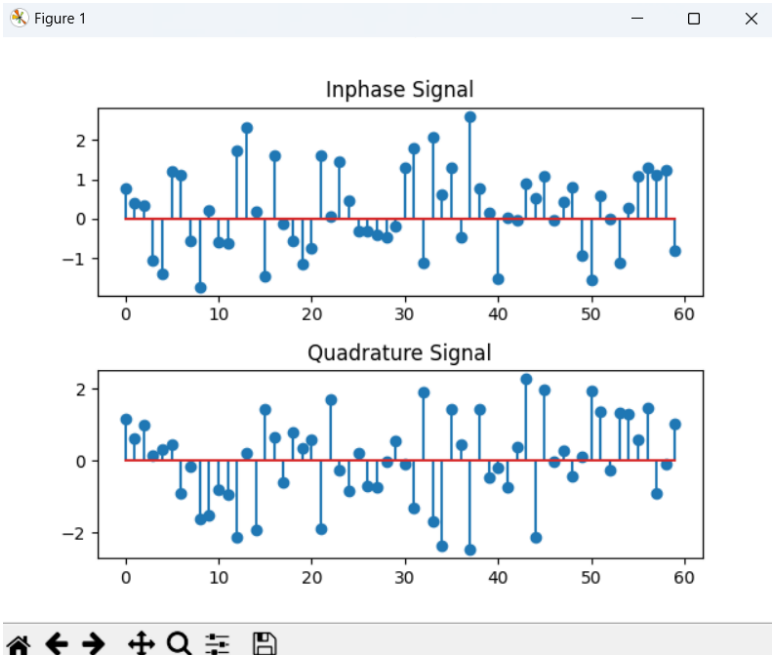
รูปที่ 4.112 ผลลัพธ์ทดสอบผลของช่องสัญญาณสื่อสารโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



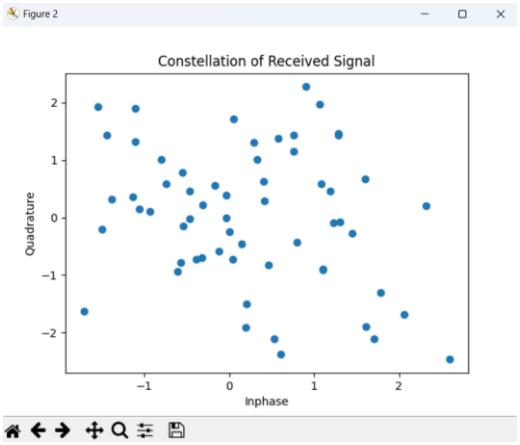
รูปที่ 4.113 ผลลัพธ์ทดสอบผลของช่องสัญญาณสื่อสารโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



รูปที่ 4.114 ผลลัพธ์ทดสอบผลของช่องสัญญาณสื่อสารโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G



รูปที่ 4.115 กราฟของสัญญาณที่ผ่านช่องสัญญาณสื่อสารโดยใช้อุปกรณ์ FPGA บนหน้าต่าง



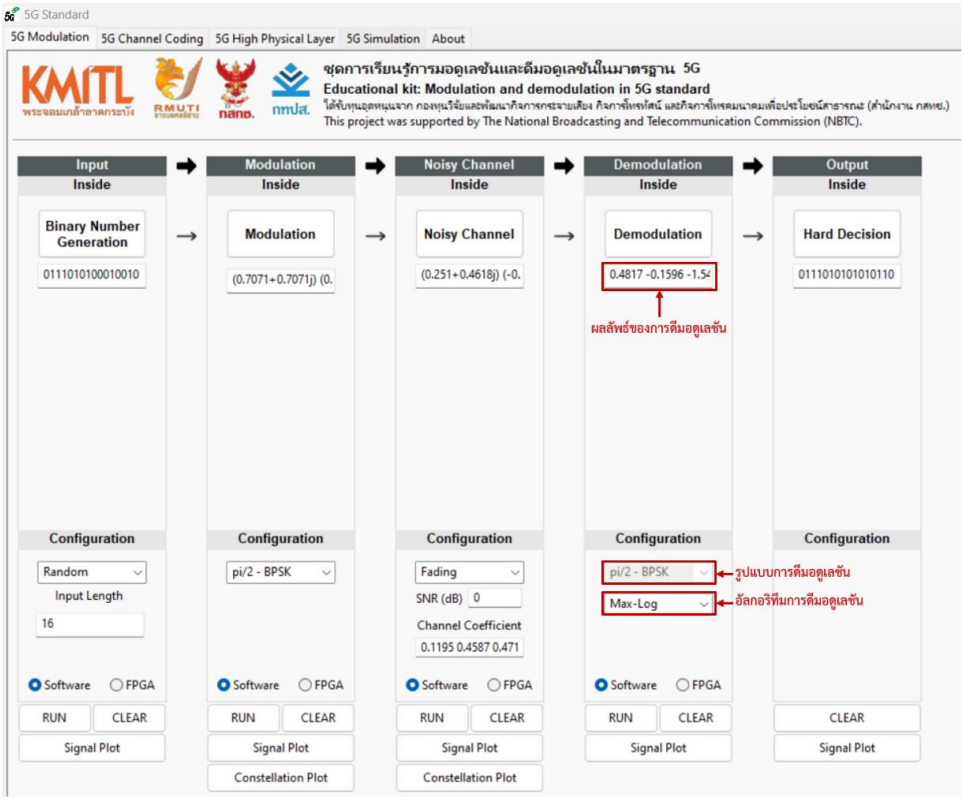
รูปที่ 4.116 แผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านช่องสัญญาณสื่อสารโดยใช้อุปกรณ์ FPGA

4.1.6 ผลการทดสอบ GUI เพื่อแสดงผลการดีมอดูเลชันมาตรฐาน 5G

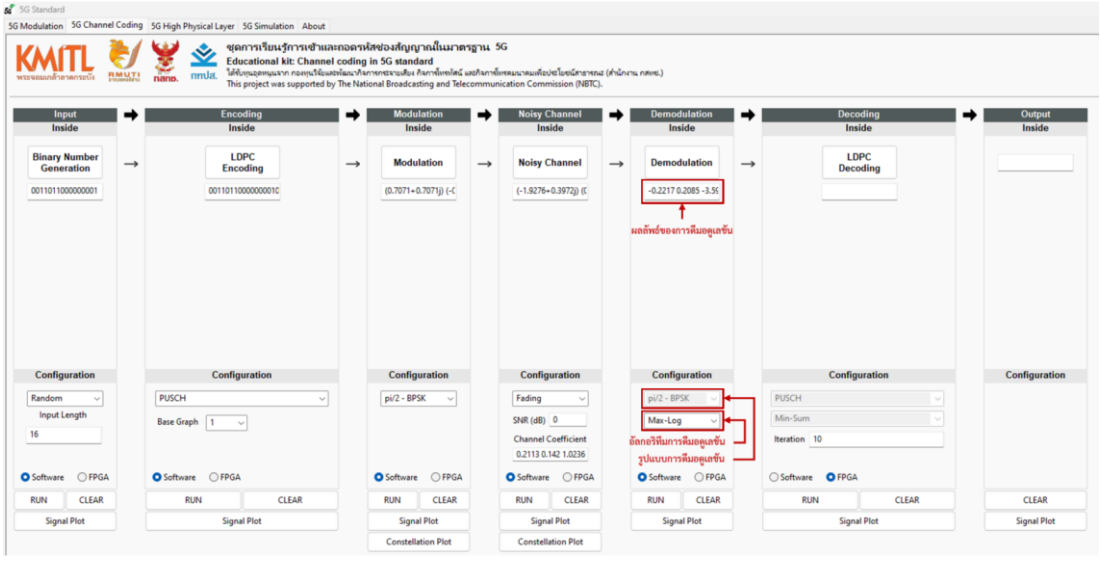
การดีมอดูเลชันมาตรฐาน 5G แสดงอยู่ในคอลัมน์ที่ 4 ของหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G คอลัมน์ที่ 5 ของหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G และคอลัมน์ที่ 6 ของหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G โดยผู้ใช้สามารถเลือกรูปแบบการดีมอดูเลชันได้ทั้งหมด 7 รูปแบบ ได้แก่ BPSK, π/2-BPSK, QPSK, 16QAM, 64QAM, 256QAM และ 1024QAM ทั้งนี้รูปแบบการดีมอดูเลชันจะต้องสอดคล้องกับรูปแบบการมอดูเลชัน นอกจากนี้ ผู้ใช้สามารถเลือกอัลกอริทึมการดีมอดูเลชันได้ 2 อัลกอริทึม ได้แก่ Max-log และ LUT รวมทั้งสามารถเลือกทดสอบผลการดีมอดูเลชันโดยใช้ซอฟต์แวร์หรืออุปกรณ์ FPGA

- กรณีใช้ซอฟต์แวร์ทดสอบการดีมอดูเลชัน

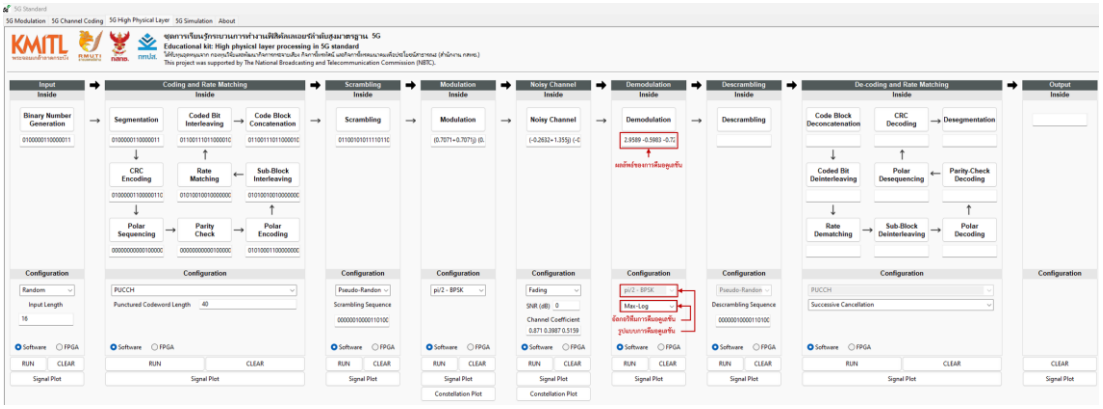
การทดสอบผลการดีมอดูเลชันโดยใช้ซอฟต์แวร์จะกำหนดให้อินพุตเป็นข้อมูลที่ผ่านช่องสัญญาณสื่อสารและกำหนดรูปแบบการดีมอดูเลชันเป็นแบบ π/2-BPSK เช่นเดียวกับการมอดูเลชันและเลือกอัลกอริทึมการดีมอดูเลชันแบบ Max-Log จากนั้นเมื่อกดปุ่ม “RUN” ซอฟต์แวร์ GUI จะเรียกใช้ซอฟต์แวร์ Module demodulation.py เพื่อดำเนินการดีมอดูเลชันสัญญาณ และแสดงผลลัพธ์ที่ได้ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.117 - รูปที่ 4.119 โดยผลลัพธ์ดังกล่าวจะแสดงอยู่ในรูปของตัวเลขจำนวนจริงซึ่งเป็นค่าอัตราส่วนความควรจะเป็นแบบล็อก (Log-Likelihood Ratio: LLR) นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟของสัญญาณที่ผ่านการดีมอดูเลชันได้โดยกดปุ่ม “Signal Plot” ตัวอย่างกราฟของสัญญาณที่ผ่านการดีมอดูเลชันซึ่งทดสอบโดยใช้ซอฟต์แวร์ แสดงดังรูปที่ 4.120



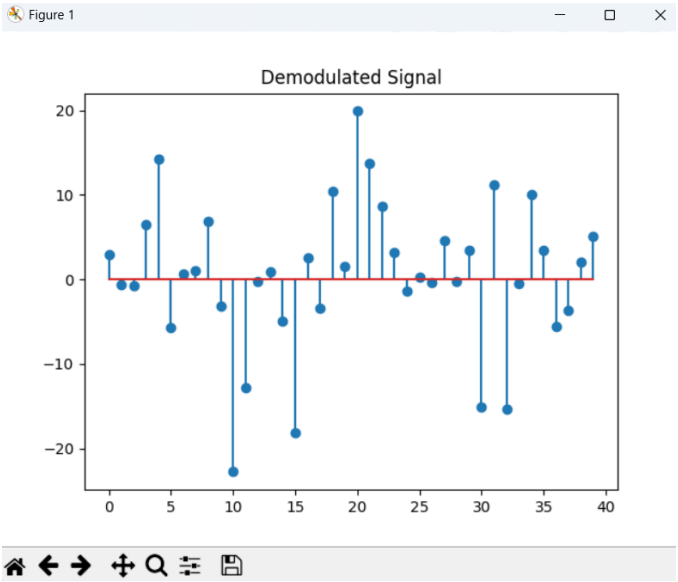
รูปที่ 4.117 ผลลัพธ์ของการดีมอดูเลชันโดยใช้ซอฟต์แวร์บนหน้าต่าง
ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



รูปที่ 4.118 ผลลัพธ์ของการดีมอดูเลชันโดยใช้ซอฟต์แวร์บนหน้าต่าง
ชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



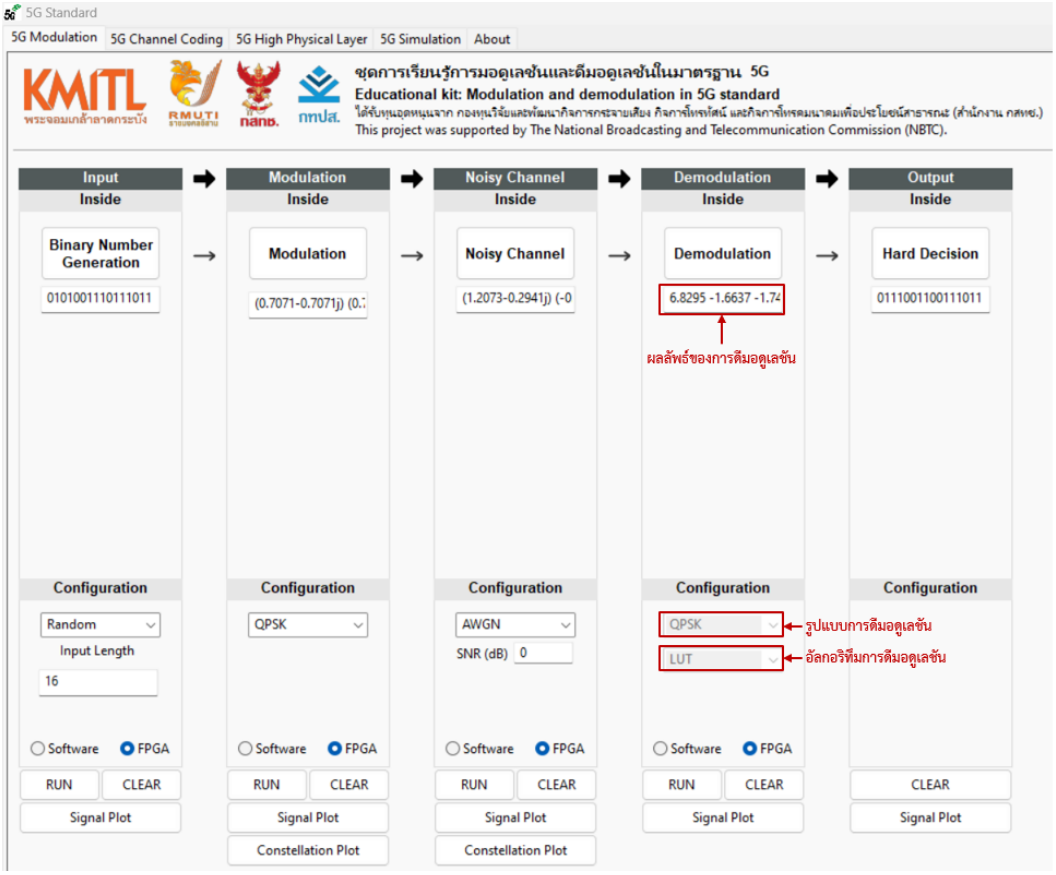
รูปที่ 4.119 ผลลัพธ์การดีมอดูเลชันโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G



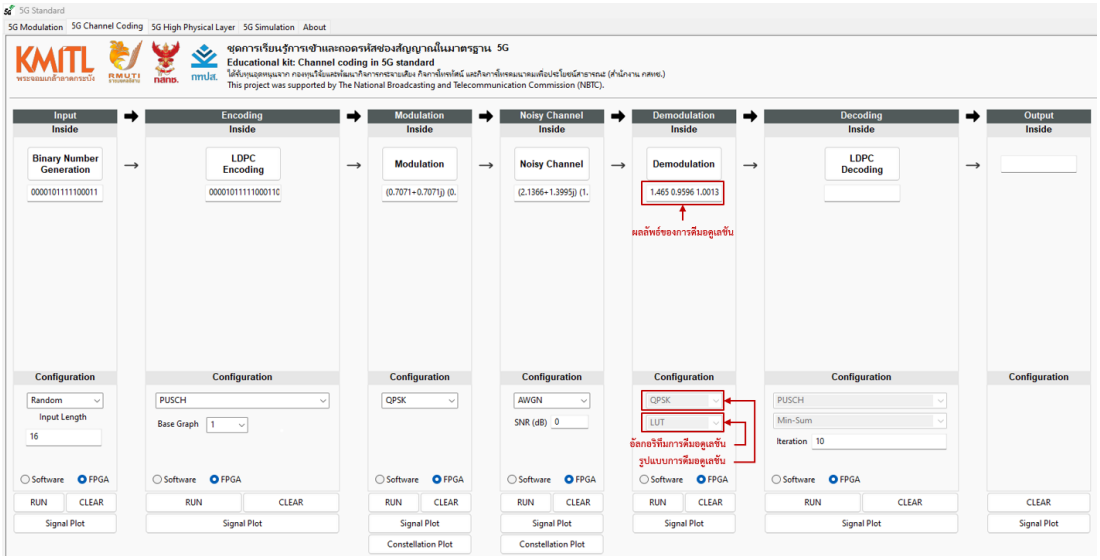
รูปที่ 4.120 กราฟของสัญญาณที่ผ่านการดีมอดูเลชันโดยใช้ซอฟต์แวร์

- กรณีใช้อุปกรณ์ FPGA ทดสอบการดีมอดูเลชัน

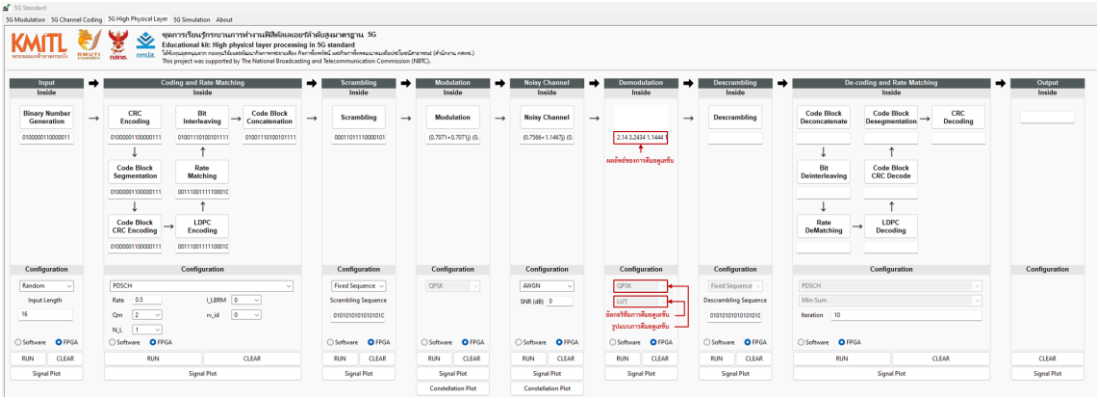
การทดสอบผลของช่องสัญญาณสื่อสารโดยใช้อุปกรณ์ FPGA จะกำหนดให้อินพุตเป็นข้อมูลที่ผ่านช่องสัญญาณสื่อสารและกำหนดรูปแบบการดีมอดูเลชันเป็นแบบ เช่นเดียวกับการมอดูเลชัน และเนื่องจากความซับซ้อนของอัลกอริทึม Max-Log กรณีการใช้อุปกรณ์ FPGA จะใช้อัลกอริทึม LUT ในการดีมอดูเลชันเท่านั้น จากนั้นเมื่อกดปุ่ม “RUN” ซอฟต์แวร์ GUI จะส่งสัญญาณที่ผ่านช่องสัญญาณสื่อสารและรูปแบบการดีมอดูเลชันผ่าน Serial Port ไปยังอุปกรณ์ FPGA ซึ่งอุปกรณ์ FPGA จะดำเนินการดีมอดูเลชัน และส่งผลลัพธ์ผ่าน Serial Port กลับมาแสดงบน GUI ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.121 - รูปที่ 4.123 โดยผลลัพธ์ดังกล่าวจะแสดงอยู่ในรูปของตัวเลขจำนวนจริงซึ่งเป็นค่า LLR นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟของสัญญาณที่ผ่านการดีมอดูเลชันได้โดยกดปุ่ม “Signal Plot” ตัวอย่างกราฟของสัญญาณที่ผ่านการดีมอดูเลชันซึ่งทดสอบโดยใช้อุปกรณ์ FPGA แสดงดังรูปที่ 4.124



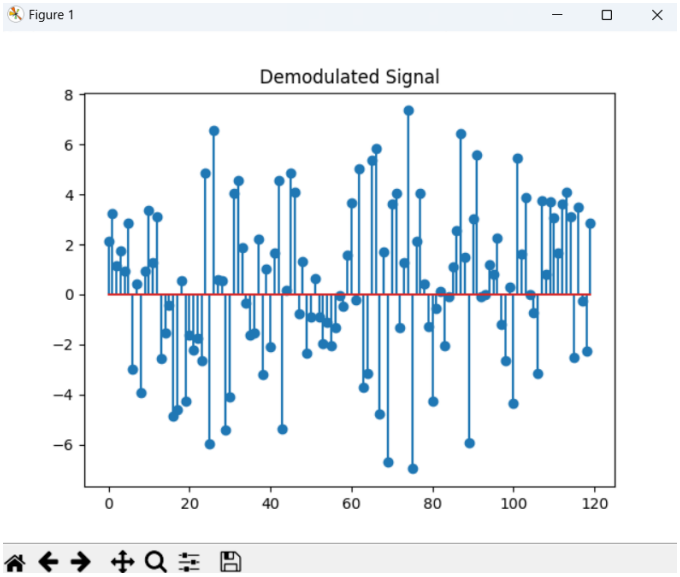
รูปที่ 4.121 ผลลัพธ์ของการดีมอดูเลชันโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G



รูปที่ 4.122 ผลลัพธ์ของการดีมอดูเลชันโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



รูปที่ 4.123 ผลลัพธ์การดีมอดูเลชันโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G



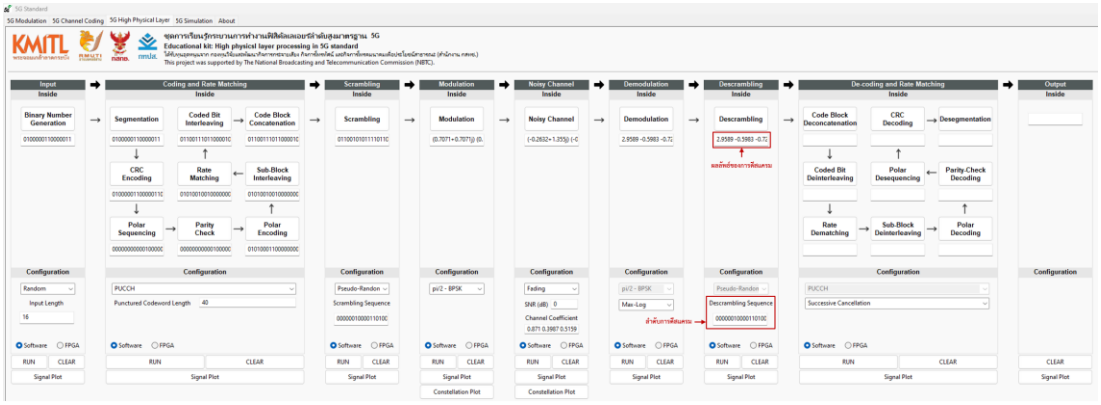
รูปที่ 4.124 กราฟของสัญญาณที่ผ่านการดีมอดูเลชันโดยใช้อุปกรณ์ FPGA

4.1.7 ผลการทดสอบ GUI เพื่อแสดงผลการดีสแครมมาตรฐาน 5G

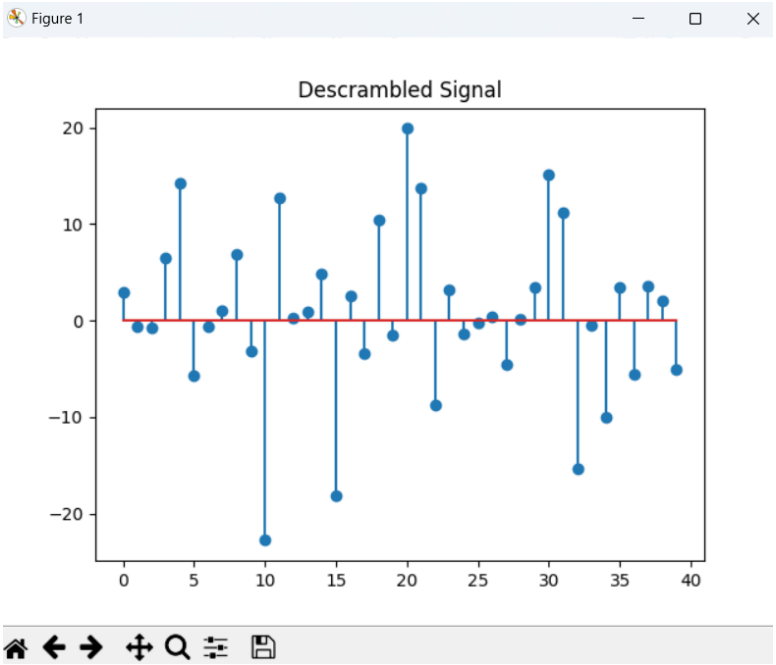
การดีสแครมมาตรฐาน 5G แสดงอยู่ในคอลัมน์ที่ 7 ของหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G โดยผู้ใช้สามารถเลือกรูปแบบการสร้างลำดับดีสแครมได้ทั้งหมด 3 รูปแบบ ได้แก่ การใช้ลำดับแบบตายตัว การสร้างลำดับแบบสุ่มเทียม และการป้อนลำดับด้วยตนเอง นอกจากนี้ ผู้ใช้สามารถเลือกทดสอบการดีสแครมโดยใช้ซอฟต์แวร์หรืออุปกรณ์ FPGA อย่างไรก็ตาม รูปแบบการสร้างลำดับดีสแครมและการลำดับดีสแครมจะต้องตรงกับการสแครม และการดีสแครมโดยใช้อุปกรณ์ FPGA จะสามารถสร้างลำดับการดีสแครมโดยใช้ลำดับแบบตายตัวเท่านั้น

- กรณีใช้ซอฟต์แวร์ดีสแครม

การทดสอบการดีสแครมโดยใช้ซอฟต์แวร์จะกำหนดให้อินพุตเป็นสัญญาณดีมอดูเลชัน และเลือกการสร้างลำดับแบบสุ่มเทียมตามการสแครม จากนั้นเมื่อกดปุ่ม “RUN” ซอฟแวร์ GUI จะเรียกใช้ซอฟแวร์ Module nrPDSCHPRBS.py เพื่อดำเนินการดีสแครม จากนั้นแสดงผลลัพธ์ที่ได้ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.125 โดยผลลัพธ์ดังกล่าวจะอยู่ในรูปของตัวเลขจำนวนจริง นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟของสัญญาณสแครมได้โดยกดปุ่ม “Signal Plot” ตัวอย่างกราฟของสัญญาณสแครมแสดงดังรูปที่ 4.126



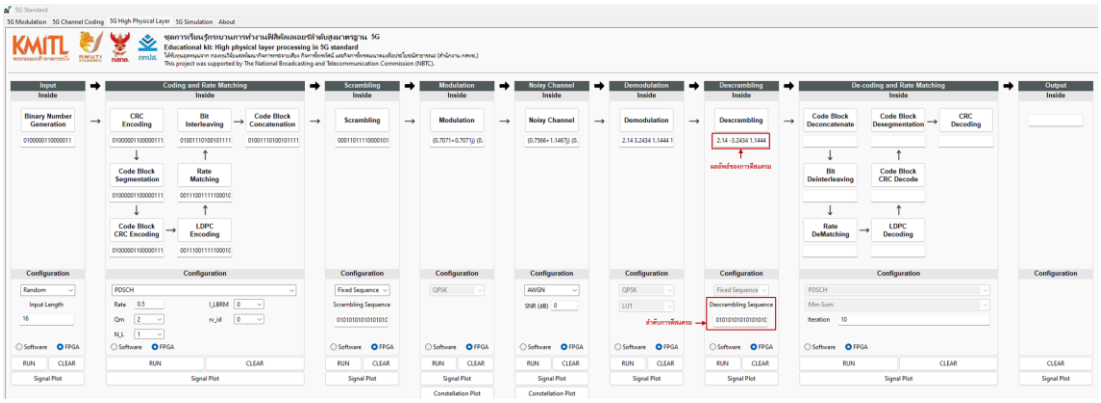
รูปที่ 4.125 ผลลัพธ์การดีสแครมมาตรฐาน 5G โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



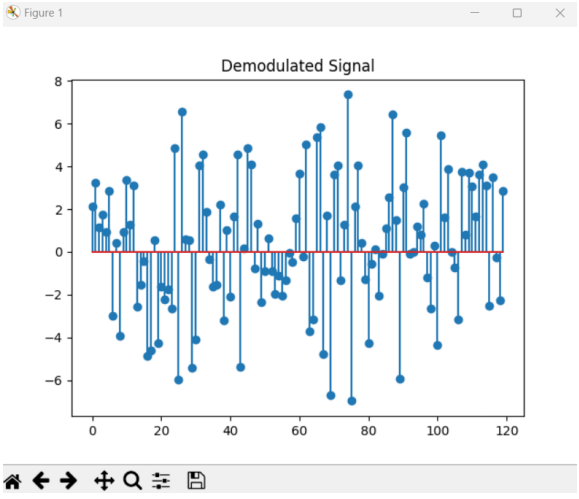
รูปที่ 4.126 กราฟของสัญญาณดีสแครมที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G

- กรณีใช้อุปกรณ์ FPGA ดีสแครม

การทดสอบการดีสแครมโดยใช้อุปกรณ์ FPGA จะกำหนดให้อินพุตเป็นสัญญาณดีมอดูเลชันและเลือกใช้ลำดับสแครมแบบตายตัว เมื่อกดปุ่ม “RUN” ซอฟต์แวร์ GUI จะส่งสัญญาณดีมอดูเลชันผ่าน Serial Port ไปยังอุปกรณ์ FPGA จากนั้นอุปกรณ์ FPGA จะดำเนินการดีสแครม จากนั้นส่งผลลัพธ์กลับมาแสดงบน GUI ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.127 โดยผลลัพธ์ดังกล่าวจะแสดงอยู่ในรูปของตัวเลขจำนวนจริง นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟของสัญญาณที่ได้จากการดีสแครมได้โดยกดปุ่ม “Signal Plot” ตัวอย่างกราฟของสัญญาณที่ได้จากการเข้ารหัสแสดงดังรูปที่ 4.128



รูปที่ 4.127 ผลลัพธ์การดีสแครมมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



รูปที่ 4.128 กราฟของสัญญาณที่ได้จากการดีสแครมที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G

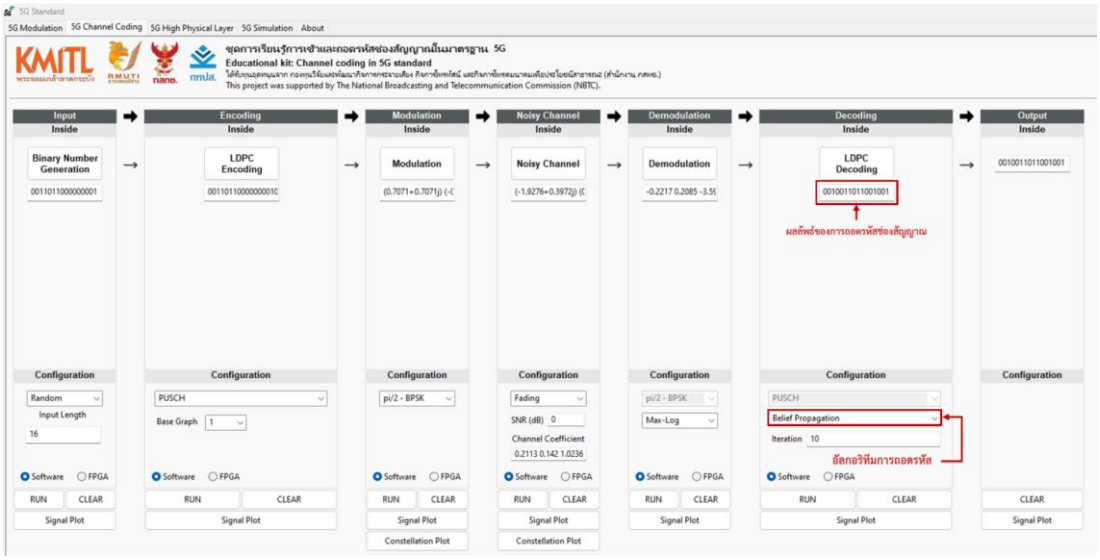
4.1.8 ผลการทดสอบ GUI เพื่อแสดงผลการถอดรหัสช่องสัญญาณมาตรฐาน 5G

การถอดรหัสช่องสัญญาณมาตรฐาน 5G แสดงอยู่ในคอลัมน์ที่ 6 ของหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G และคอลัมน์ที่ 8 ของหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G โดยผู้ใช้สามารถเลือกช่องสัญญาณภาพได้ทั้งหมด 5 ช่องสัญญาณ ได้แก่ PUSCH, PDSCH, PUCCH, PDCCH และ PBCH นอกจากนี้ ผู้ใช้สามารถเลือกอัลกอริทึมการถอดรหัสได้ และสามารถเลือกทดสอบการถอดรหัสช่องสัญญาณโดยใช้ซอฟต์แวร์หรืออุปกรณ์ FPGA อย่างไรก็ตาม รูปแบบของช่องสัญญาณกายภาพในการถอดรหัสจะต้องตรงกับการเข้ารหัส

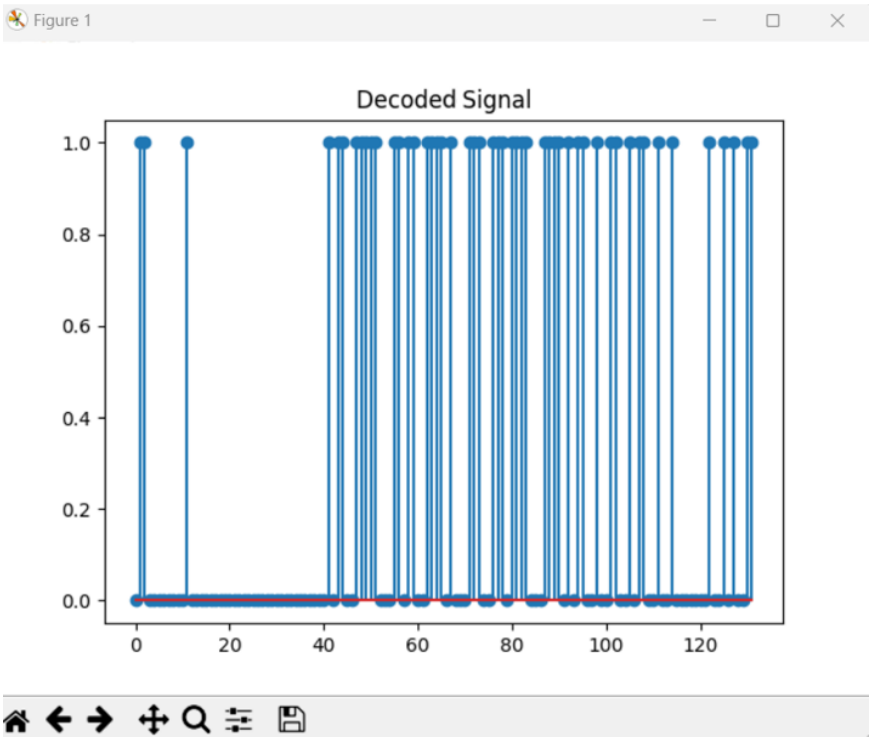
4.1.8.1 ผลการทดสอบ GUI สำหรับชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G

- กรณีใช้ซอฟต์แวร์ถอดรหัสช่องสัญญาณ

การทดสอบการถอดรหัสช่องสัญญาณโดยใช้ซอฟต์แวร์จะกำหนดให้อินพุตเป็นสัญญาณดีมอดูเลชันและเลือกช่องสัญญาณกายภาพแบบ PUSCH ตามการเข้ารหัส รวมทั้งเลือกอัลกอริทึมการถอดรหัสแบบ Belief Propagation ที่มี Iteration = 10 จากนั้นเมื่อกดปุ่ม “RUN” ซอฟแวร์ GUI จะเรียกใช้ซอฟแวร์ Module LDPCDecoderModule.py เพื่อดำเนินการถอดรหัสช่องสัญญาณและแสดงผลลัพธ์ที่ได้ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.129 โดยผลลัพธ์ดังกล่าวจะอยู่ในรูปของบิตข้อมูลแบบไบนารี นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟของสัญญาณที่ได้จากการถอดรหัสได้โดยกดปุ่ม “Signal Plot” ตัวอย่างกราฟของสัญญาณที่ได้จากการถอดรหัสแสดงดังรูปที่ 4.130



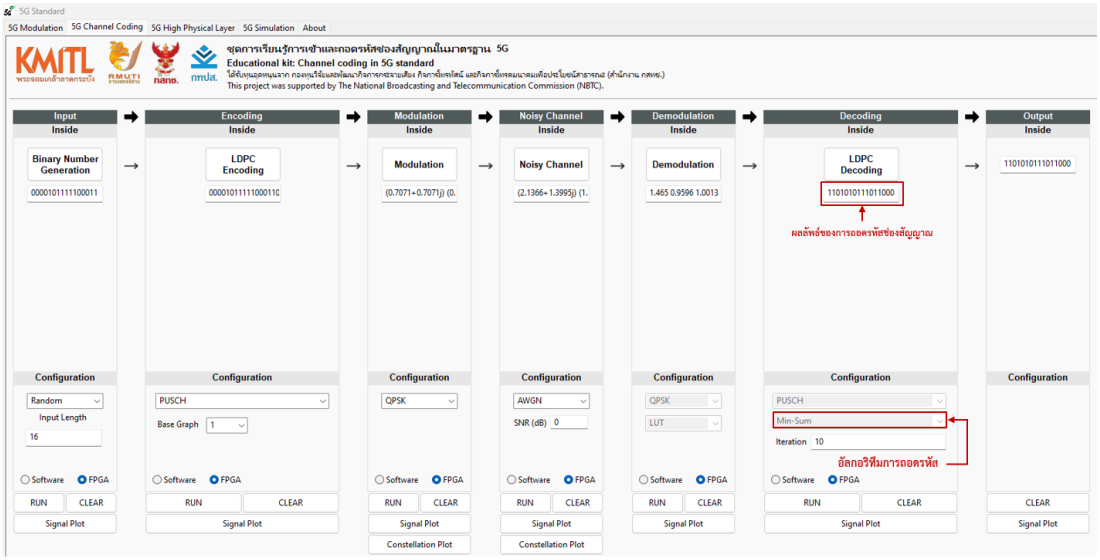
รูปที่ 4.129 ผลลัพธ์การถอดรหัสช่องสัญญาณมาตรฐาน 5G โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G



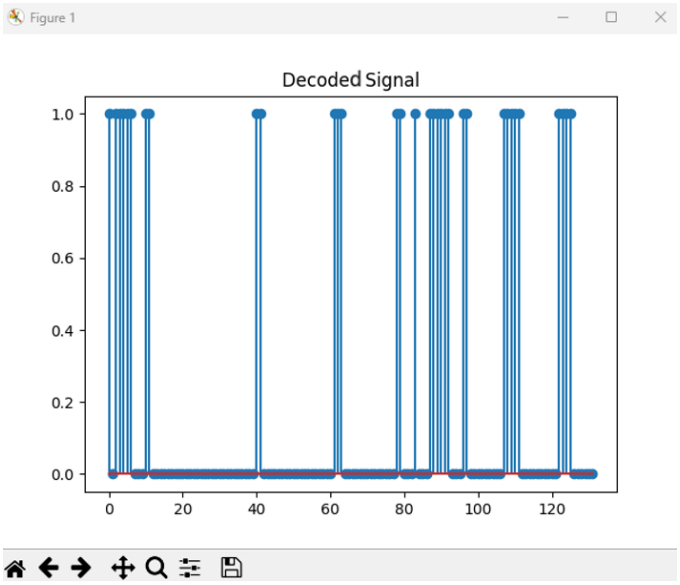
รูปที่ 4.130 กราฟของสัญญาณที่ได้จากการถอดรหัสที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G

- กรณีใช้อุปกรณ์ FPGA ถอดรหัสช่องสัญญาณ

การทดสอบการถอดรหัสช่องสัญญาณโดยใช้อุปกรณ์ FPGA จะกำหนดให้อินพุตเป็นสัญญาณดีมอดูเลชัน และเลือกช่องสัญญาณกายภาพแบบ PDSCH ตามการเข้ารหัส เนื่องจากความซับซ้อนของอัลกอริทึมการถอดรหัส กรณีการใช้อุปกรณ์ FPGA จะใช้อัลกอริทึม Min-Sum ในการถอดรหัสเท่านั้น โดยกำหนดให้มี Iteration = 10 เมื่อกดปุ่ม “RUN” ซอฟต์แวร์ GUI จะส่งสัญญาณดีมอดูเลชัน อัลกอริทึมการถอดรหัส และพารามิเตอร์การถอดรหัสผ่าน Serial Port ไปยังอุปกรณ์ FPGA จากนั้นอุปกรณ์ FPGA จะดำเนินการถอดรหัสช่องสัญญาณและส่งผลลัพธ์ผ่าน Serial Port กลับมาแสดงบน GUI ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.131 โดยผลลัพธ์ดังกล่าวจะแสดงอยู่ในรูปของบิตข้อมูลแบบไบนารี นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟของสัญญาณที่ได้จากการถอดรหัสได้โดยกดปุ่ม “Signal Plot” ตัวอย่างกราฟของสัญญาณที่ได้จากการถอดรหัสแสดงดังรูปที่ 4.132



รูปที่ 4.131 ผลลัพธ์การถอดรหัสช่องสัญญาณมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G

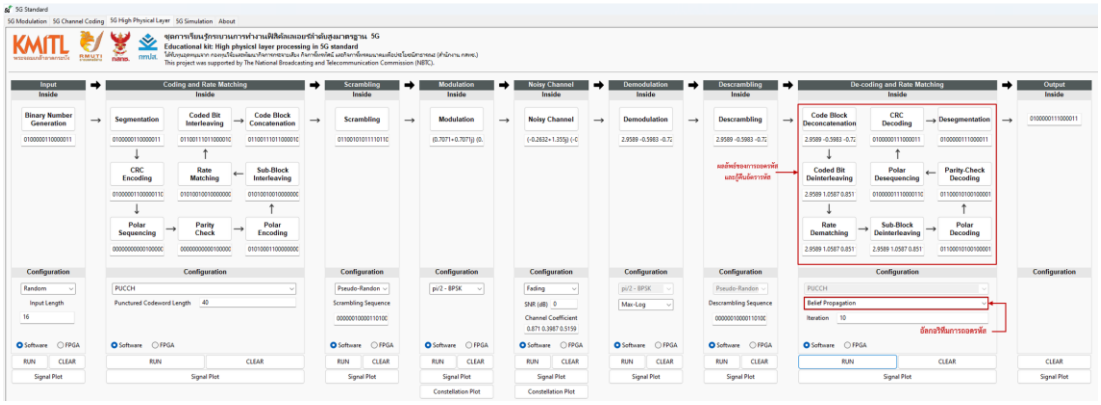


รูปที่ 4.132 กราฟของสัญญาณที่ได้จากการถอดรหัสที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G

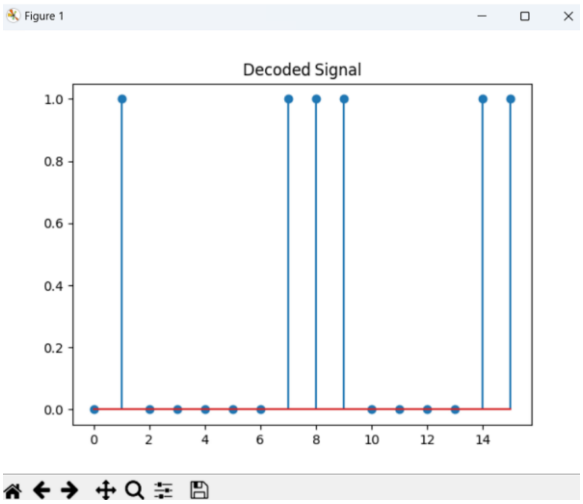
4.1.8.2 ผลการทดสอบ GUI สำหรับชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G

- กรณีใช้ซอฟต์แวร์ถอดรหัสและกู้คืนอัตรารหัส

การทดสอบการถอดรหัสและกู้คืนอัตรารหัสโดยใช้ซอฟต์แวร์จะกำหนดให้อินพุตเป็นสัญญาณดีสแครม และช่องสัญญาณกายภาพแบบ PUCCH ตามการเข้ารหัส รวมทั้งเลือกอัลกอริทึมการถอดรหัสแบบ Belief Propagation ที่มี Iteration = 10 จากนั้นเมื่อกดปุ่ม “RUN” ซอฟแวร์ GUI จะเรียกใช้ซอฟแวร์ Module PolarDecoderModule.py เพื่อดำเนินการถอดรหัสและกู้คืนอัตรารหัส จากนั้นแสดงผลลัพธ์ที่ได้ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.133 โดยผลลัพธ์ดังกล่าวจะอยู่ในรูปของบิตข้อมูลแบบไบนารี นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟของสัญญาณที่ได้จากการถอดรหัสได้โดยกดปุ่ม “Signal Plot” ตัวอย่างกราฟของสัญญาณที่ได้จากการถอดรหัสแสดงดังรูปที่ 4.134



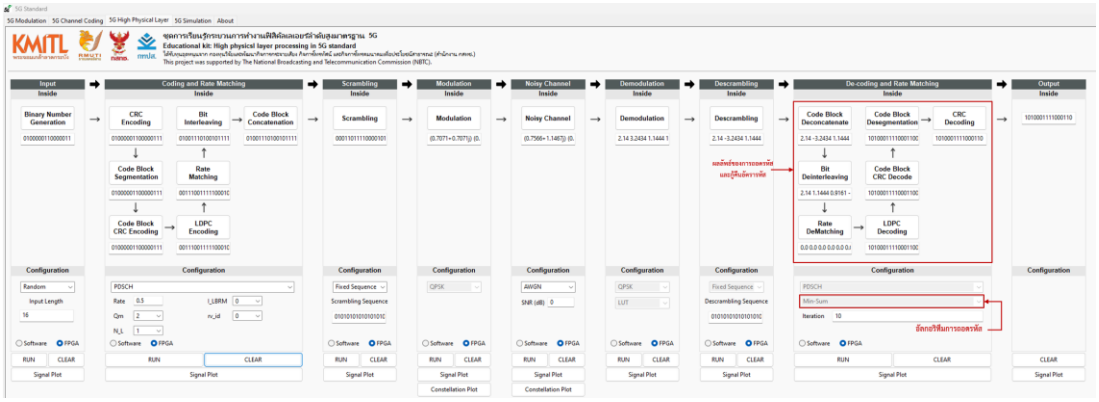
รูปที่ 4.133 ผลลัพธ์การถอดรหัสช่องสัญญาณมาตรฐาน 5G โดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



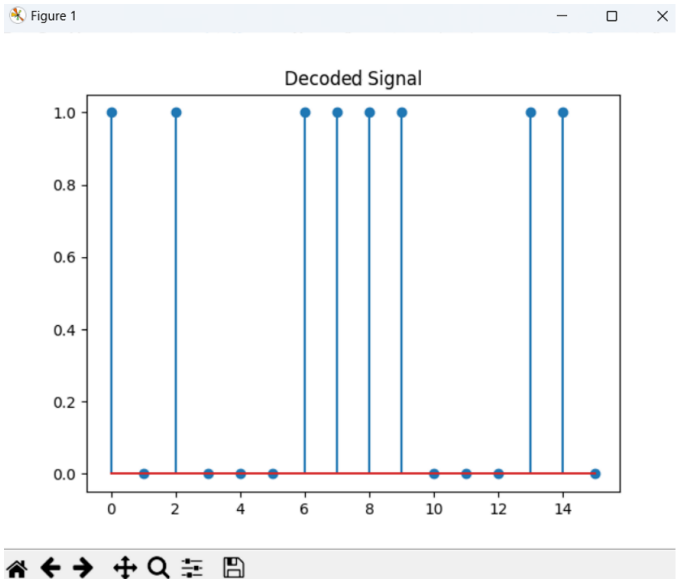
รูปที่ 4.134 กราฟของสัญญาณที่ได้จากการถอดรหัสที่สร้างโดยใช้ซอฟต์แวร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G

- กรณีใช้อุปกรณ์ FPGA ถอดรหัสและปรับอัตรารหัส

การทดสอบการถอดรหัสและกู้คืนอัตรารหัสโดยใช้อุปกรณ์ FPGA จะกำหนดให้อินพุตเป็นสัญญาณดีสแครม และเลือกช่องสัญญาณกายภาพแบบ PDSCH ตามการถอดรหัส เนื่องจากความซับซ้อนของอัลกอริทึมการถอดรหัส กรณีการใช้อุปกรณ์ FPGA จะใช้อัลกอรึทึม Min-Sum ในการถอดรหัสเท่านั้น โดยกำหนดให้มี Iteration = 10 เมื่อกดปุ่ม “RUN” ซอฟต์แวร์ GUI จะส่งสัญญาณดีสแครม อัลกอริทึมการถอดรหัส และพารามิเตอร์การถอดรหัสผ่าน Serial Port ไปยังอุปกรณ์ FPGA จากนั้นอุปกรณ์ FPGA จะดำเนินการถอดรหัสช่องสัญญาณและส่งผลลัพธ์ผ่าน Serial Port กลับมาแสดงบน GUI ภายในกล่องข้อความซึ่งอยู่ในส่วนบล็อกแสดงผลภายใน ดังแสดงในรูปที่ 4.135 โดยผลลัพธ์ดังกล่าวจะแสดงอยู่ในรูปของบิตข้อมูลแบบไบนารี นอกจากนี้ผู้ใช้สามารถเรียกดูกราฟของสัญญาณที่ได้จากการถอดรหัสได้โดยกดปุ่ม “Signal Plot” ตัวอย่างกราฟของสัญญาณที่ได้จากการถอดรหัสแสดงดังรูปที่ 4.136



รูปที่ 4.135 ผลลัพธ์การถอดรหัสช่องสัญญาณมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G



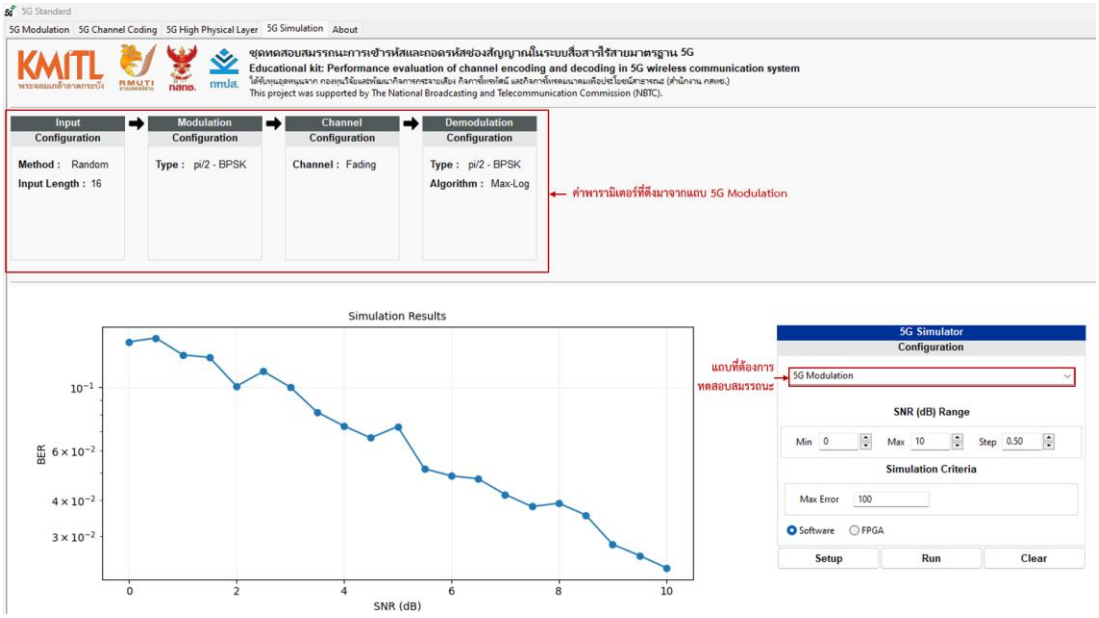
รูปที่ 4.136 กราฟของสัญญาณที่ได้จากการถอดรหัสที่สร้างโดยใช้อุปกรณ์ FPGA บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์มาตรฐาน 5G

4.1.9 ผลการทดสอบ GUI เพื่อทดสอบอัตราบิตผิดพลาดการมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G

เมื่อผู้ใช้กำหนดค่าพารามิเตอร์บนหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G เสร็จสิ้น ผู้ใช้สามารถเลือกแถบ 5G Simulation เพื่อทดสอบอัตราบิตผิดพลาดของการมอดูเลชัน โดยเลือก 5G Modulation ภายในกล่องตัวเลือกของส่วนการกำหนดค่า และกดปุ่ม “Setup” จากนั้นซอฟต์แวร์ GUI จะแสดงข้อมูลพารามิเตอร์ที่กำหนดไว้ในหน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันในมาตรฐาน 5G นอกจากนี้ ผู้ใช้สามารถกำหนดช่วงของค่า SNR และจำนวนบล็อกผิดพลาดสูงสุดที่ต้องการทดสอบสมรรถนะ รวมทั้งสามารถเลือกทดสอบอัตราบิตผิดพลาดโดยใช้ซอฟต์แวร์หรืออุปกรณ์ FPGA

- กรณีใช้ซอฟต์แวร์ทดสอบอัตราบิตผิดพลาดของการมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G

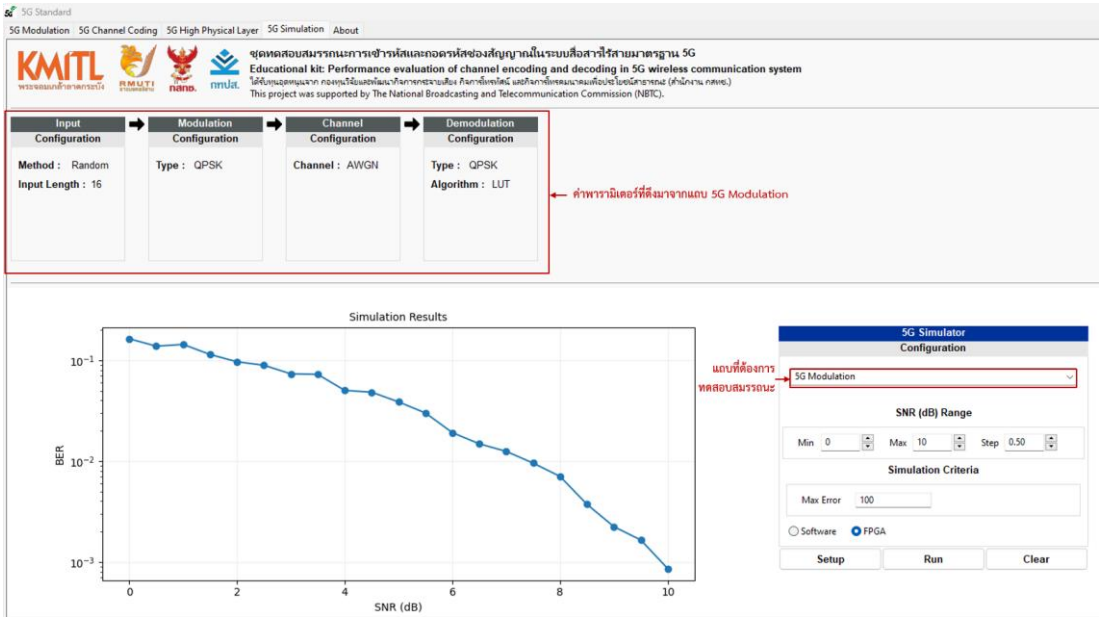
ผลการทดสอบอัตราบิตผิดพลาดของการมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G โดยใช้ซอฟต์แวร์ซึ่งกำหนดให้ข้อมูลอินพุตความยาว 16 บิต ถูกมอดูเลชันแบบ $\pi/2$ - BPSK ส่งผ่านช่องสัญญาณแบบ Fading Channel และทางฝั่งรับใช้อัลกอริทึมการดีมอดูเลชันแบบ Max-Log แสดงดังรูปที่ 4.137



รูปที่ 4.137 ผลการทดสอบอัตราบิตผิดพลาดของการมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G โดยใช้ซอฟต์แวร์

- กรณีใช้อุปกรณ์ FPGA ทดสอบอัตราบิตผิดพลาดของการมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G

ผลการทดสอบอัตราบิตผิดพลาดของการมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA ซึ่งกำหนดให้ข้อมูลอินพุตความยาว 16 บิต ถูกมอดูเลชันแบบ QPSK ส่งผ่านช่องสัญญาณแบบ AWGN Channel และทางฝั่งรับใช้อัลกอริทึมการดีมอดูเลชันแบบ LUT แสดงดังรูปที่ 4.138



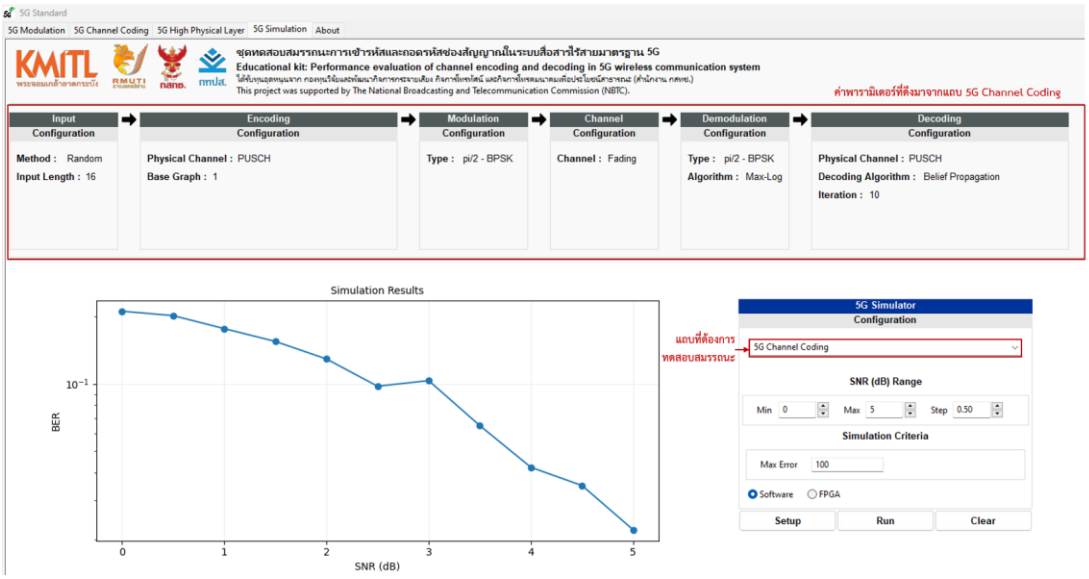
รูปที่ 4.138 ผลการทดสอบอัตราบิตผิดพลาดของการมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA

4.1.10 ผลการทดสอบ GUI เพื่อทดสอบอัตราบิตผิดพลาดการเข้าและถอดรหัสมาตรฐาน 5G

เมื่อผู้ใช้กำหนดค่าพารามิเตอร์บนหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G เสร็จสิ้น ผู้ใช้สามารถเลือกแถบ 5G Simulation เพื่อทดสอบอัตราบิตผิดพลาดของการมอดูเลชัน โดยเลือก 5G Channel Coding ภายในกล่องตัวเลือกของส่วนการกำหนดค่า และกดปุ่ม “Setup” จากนั้นซอฟต์แวร์ GUI จะแสดงข้อมูลพารามิเตอร์ที่กำหนดไว้ในหน้าต่างชุดการเรียนรู้การเข้าและถอดรหัสมาตรฐาน 5G นอกจากนี้ ผู้ใช้สามารถกำหนดช่วงของค่า SNR และจำนวนบล็อกผิดพลาดสูงสุดที่ต้องการทดสอบสมรรถนะ รวมทั้งสามารถเลือกทดสอบอัตราบิตผิดพลาดโดยใช้ซอฟต์แวร์หรืออุปกรณ์ FPGA

- กรณีใช้ซอฟต์แวร์ทดสอบอัตราบิตผิดพลาดของการเข้าและถอดรหัสมาตรฐาน 5G

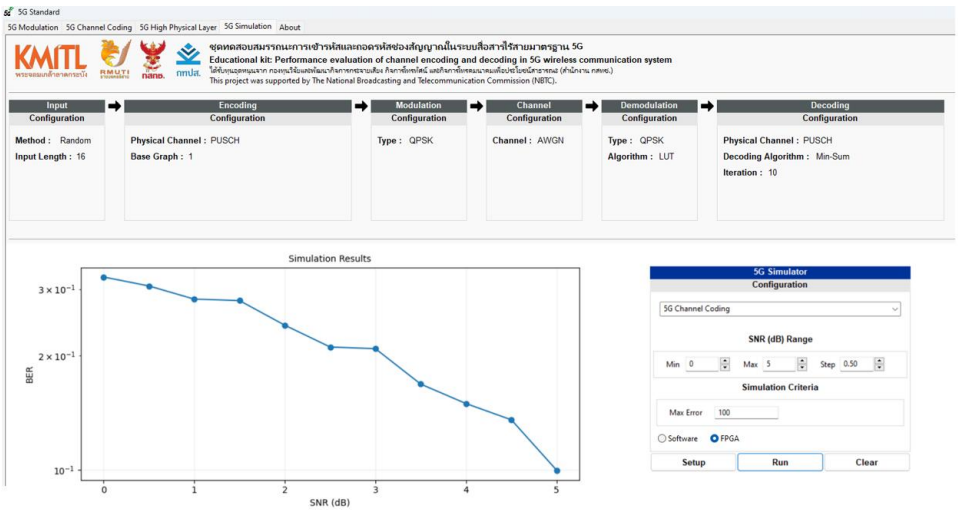
ผลการทดสอบอัตราบิตผิดพลาดของการเข้าและถอดรหัสมาตรฐาน 5G โดยใช้ซอฟต์แวร์ซึ่งกำหนดให้ข้อมูลอินพุตความยาว 16 บิต ถูกเข้ารหัสโดยใช้ช่องสัญญาณกายภาพ PUSCH ที่มีกราฟฐาน 1 จากนั้นมอดูเลชันแบบ π/2 - BPSK และส่งผ่านช่องสัญญาณแบบ Fading Channel สำหรับทางฝั่งรับใช้อัลกอริทึมการดีมอดูเลชันแบบ Max-Log และใช้อัลกอริทึมการถอดรหัสแบบ Belief Propagation ที่มี Iteration = 10 แสดงดังรูปที่ 4.139



รูปที่ 4.139 ผลการทดสอบอัตราบิตผิดพลาดของการเข้าและถอดรหัสมาตรฐาน 5G โดยใช้ซอฟต์แวร์

- กรณีใช้อุปกรณ์ FPGA ทดสอบอัตราบิตผิดพลาดของการเข้าและถอดรหัสมาตรฐาน 5G

ผลการทดสอบอัตราบิตผิดพลาดของการมอดูเลชันมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA ซึ่งกำหนดให้ข้อมูลอินพุตความยาว 16 บิต ถูกเข้ารหัสโดยใช้ช่องสัญญาณกายภาพ PUSCH ที่มีกราฟฐาน 1 จากนั้นมอดูเลชันแบบ QPSK และส่งผ่านช่องสัญญาณแบบ AWGN Channel สำหรับทางฝั่งรับใช้อัลกอริทึมการดีมอดูเลชันแบบ LUT และใช้อัลกอริทึมการถอดรหัสแบบ Min-Sum ที่มี Iteration = 10 แสดงดังรูปที่ 4.140



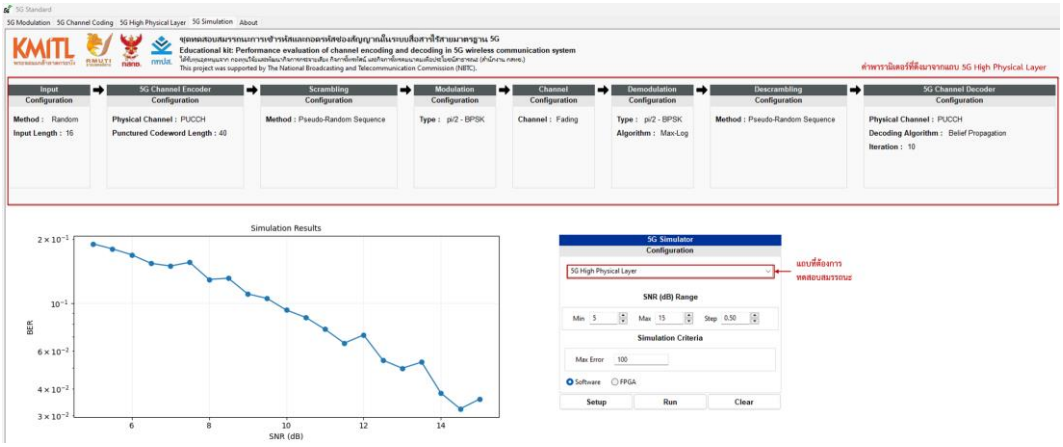
รูปที่ 4.140 ผลการทดสอบอัตราบิตผิดพลาดของการเข้าและถอดรหัสมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA

4.1.11 ผลการทดสอบ GUI เพื่อทดสอบอัตราบิตผิดพลาดกระบวนทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G

เมื่อผู้ใช้กำหนดค่าพารามิเตอร์บนหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G เสร็จสิ้น ผู้ใช้สามารถเลือกแถบ 5G Simulation เพื่อทดสอบอัตราบิตผิดพลาดของการมอดูเลชัน โดยเลือก 5G High Physical Layer ภายในกล่องตัวเลือกของส่วนการกำหนดค่า และกดปุ่ม “Setup” จากนั้นซอฟต์แวร์ GUI จะแสดงข้อมูลพารามิเตอร์ที่กำหนดไว้ในหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G นอกจากนี้ ผู้ใช้สามารถกำหนดช่วงของค่า SNR และจำนวนบล็อกผิดพลาดสูงสุดที่ต้องการทดสอบสมรรถนะ รวมทั้งสามารถเลือกทดสอบอัตราบิตผิดพลาดโดยใช้ซอฟต์แวร์หรืออุปกรณ์ FPGA

- กรณีใช้ซอฟต์แวร์ทดสอบอัตราบิตผิดพลาดของกระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G

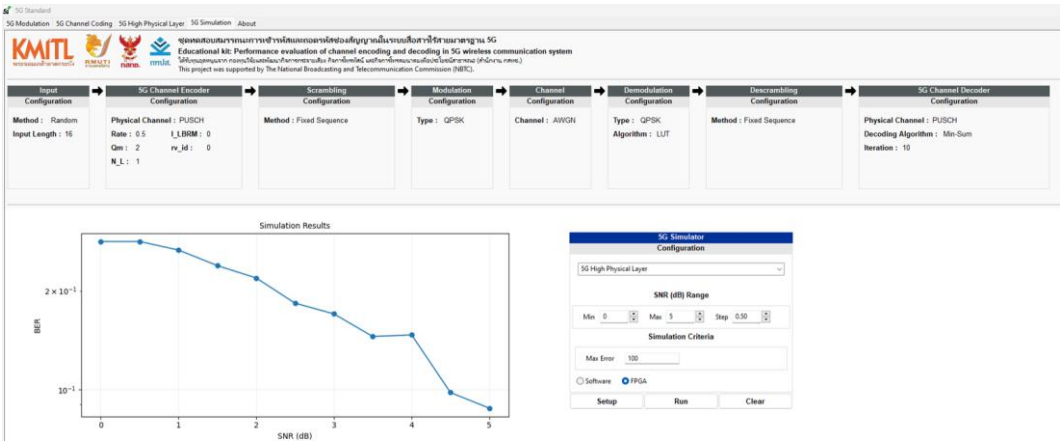
ผลการทดสอบอัตราบิตผิดพลาดของกระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G โดยใช้ซอฟต์แวร์ซึ่งกำหนดให้ข้อมูลอินพุตความยาว 16 บิต ถูกเข้ารหัสโดยใช้ช่องสัญญาณกายภาพ PUCCH ที่มีความยาวหลังการพังค์เจอร์ 40 บิต ดำเนินการสแครมโดยใช้ลำดับสแครมแบบสุ่มเทียม จากนั้นมอดูเลชันแบบ $\pi/2$ - BPSK และส่งผ่านช่องสัญญาณแบบ Fading Channel สำหรับทางฝั่งรับใช้อัลกอริทึมการดีมอดูเลชันแบบ Max-Log และใช้อัลกอริทึมการถอดรหัสแบบ Belief Propagation ที่มี Iteration = 10 แสดงดังรูปที่ 4.141



รูปที่ 4.141 ผลการทดสอบอัตราบิตผิดพลาดของกระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G โดยใช้ซอฟต์แวร์

- กรณีใช้อุปกรณ์ FPGA ทดสอบอัตราบิตผิดพลาดของกระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงรหัสมาตรฐาน 5G

ผลการทดสอบอัตราบิตผิดพลาดของกระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA ซึ่งกำหนดให้ข้อมูลอินพุตความยาว 16 บิต ถูกเข้ารหัสโดยใช้ช่องสัญญาณกายภาพ PUSCH ที่มีความยาวหลังการพังค์เจอร์ 40 บิต ดำเนินการสแครมโดยใช้ลำดับสแครมแบบตายตัว จากนั้นมอดูเลชันแบบ QPSK ส่งผ่านช่องสัญญาณแบบ AWGN Channel สำหรับทางฝั่งรับใช้อัลกอริทึมการดีมอดูเลชันแบบ LUT และใช้อัลกอริทึมการถอดรหัสแบบ Min-Sum ที่มี Iteration = 10 แสดงดังรูปที่ 4.142



รูปที่ 4.142 ผลการทดสอบอัตราบิตผิดพลาดของกระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G โดยใช้อุปกรณ์ FPGA

4.2 ผลการทดสอบการทำงานของซอฟต์แวร์ Module

4.2.1 รายชื่อซอฟต์แวร์ Module

โครงการนี้สร้างซอฟต์แวร์ Module ภาษา Python สำหรับชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G ชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G และชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G รวมทั้งสิ้น 9 Module ดังตารางที่ 4.1 ซึ่งซอฟต์แวร์ Module เหล่านี้จะทำงานร่วมกับซอฟต์แวร์ GUI

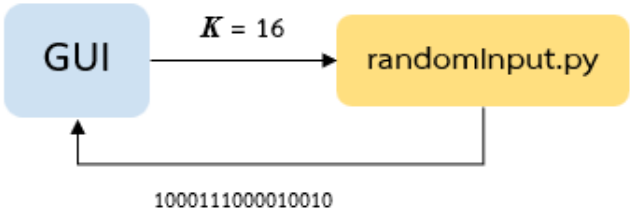
ตารางที่ 4.1 ซอฟต์แวร์ Module สำหรับซอฟต์แวร์ GUI

ซอฟต์แวร์ Module	ซอฟต์แวร์ Module ย่อย	อินพุต	เอาต์พุต
randomInput.py	-	1. จำนวนบิตอินพุต	1. บิตข้อมูลแบบไบนารี
PolarEncoderModule.py	CodeBlockSegmentation CRCInterleaver CRCAttachment CRCScrambling PolarSequencing ParityCheckEncoding PolarEncoding SubBlockInterleaver RateMatching CodedBitInterleaver CodeBlockConcatenation	1. ชนิดของช่องสัญญาณกายภาพ 2. ความยาวรหัส 3. บิตข้อมูลแบบไบนารี	1. คำรหัสแบบไบนารี
LDPCEncoderModule.py	CRCEncoder1 Segmentation CRCEncoder2 LDPCEncoder RateMatching BitInterleaving CodeBlockConcatenation	1. ชนิดของช่องสัญญาณกายภาพ 2. พารามิเตอร์การเข้ารหัส 3. บิตข้อมูลแบบไบนารี	1. คำรหัสแบบไบนารี
nrPDSCHPRBS.py	Scrambling Descrambling	1. คำรหัสแบบไบนารีหรือค่า LLR	1. สัญญาณที่ถูกสแครมหรือดีสแครม
modulation.py	pi2BPSK BPSK QPSK 16QAM 64QAM 256QAM 1024QAM	1. รูปแบบการมอดูเลชัน 2. บิตข้อมูลแบบไบนารีหรือคำรหัสแบบบารี หรือสัญญาณที่ถูกสแครม	1. สัญญาณมอดูเลชัน
Channel.py	AWGN	1. รูปแบบช่องสัญญาณ	1. สัญญาณที่ผ่านช่อง

ซอฟต์แวร์ Module	ซอฟต์แวร์ Module ย่อย	อินพุต	เอาต์พุต
	Fading	2. สัญญาณมอดูเลชัน 3. ค่า SNR	2. สัญญาณสื่อสาร
demodulation.py demodulation_LUT.py	pi2BPSK BPSK QPSK 16QAM 64QAM 256QAM 1024QAM	1. อัลกอริทึมการดีมอดูเลชัน 2. รูปแบบการดีมอดูเลชัน 3. สัญญาณที่ผ่านช่องสัญญาณสื่อสาร	1. ค่า LLR
PolarDecoderModule.py	TraverseIndexesCalculation CodedBitDeInterleaver PrePolarDecoder SCBased5GDecoder PolarBPDecoder	1. ชนิดของช่องสัญญาณกายภาพ 2. ค่า LLR หรือสัญญาณที่ได้จากการดีสแครม 3. อัลกอริทึมการถอดรหัส 4. พารามิเตอร์การถอดรหัส	1. บิตข้อมูลเอาต์พุตแบบไบนารี
LDPCDecoderModule.py	DeCodeBlockConcatenation DeBitInterleaving DeRateMatching LDPCDecoding CRCDecoder2 DeSegmentation CRCDecoder1	1. ชนิดของช่องสัญญาณกายภาพ 2. ค่า LLR หรือสัญญาณที่ได้จากการดีสแครม 3. อัลกอริทึมการถอดรหัส 4. พารามิเตอร์การถอดรหัส	1. บิตข้อมูลเอาต์พุตแบบไบนารี

4.2.2 ผลการทดสอบซอฟต์แวร์ Module เพื่อแสดงผลข้อมูลอินพุตแบบสุ่ม

ซอฟต์แวร์ Module สร้างข้อมูลอินพุตแบบสุ่มจะใช้กับชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G ชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G และชุดทดสอบสมรรถนะในระบบสื่อสารไร้สาย เมื่อซอฟต์แวร์ GUI ได้รับคำสั่งให้สร้างข้อมูลอินพุตแบบสุ่มจะทำการเรียกใช้ซอฟต์แวร์ randomInput.py โดยมีอินพุตคือความยาวของบิตข้อมูล จากนั้นซอฟต์แวร์ randomInput.py จะดำเนินการสร้างข้อมูลอินพุตแบบสุ่มในรูปแบบของไบนารี และส่งกลับไปแสดงผลยังซอฟต์แวร์ GUI ดังแสดงในรูปที่ 4.143



รูปที่ 4.143 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เพื่อแสดงข้อมูลอินพุตแบบสุ่ม

จากรูปที่ 4.143 แสดงตัวอย่างผลลัพธ์การสร้างข้อมูลอินพุตแบบสุ่ม โดยกำหนดให้มีความยาวของอินพุต 16 บิต ดังนั้น ผลลัพธ์ที่ได้จึงบิตข้อมูลแบบไบนารีที่มีความยาว 16 บิต ทั้งนี้ เนื่องจากเป็นการสร้างข้อมูลแบบสุ่ม ในการเรียกใช้งานซอฟต์แวร์ Module แต่ละครั้งจึงอาจให้ผลลัพธ์ที่แตกต่างกัน

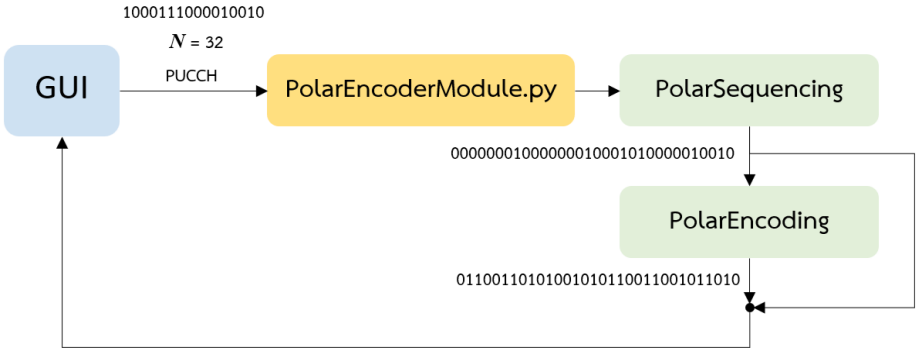
4.2.3 ผลการทดสอบซอฟต์แวร์ Module เพื่อแสดงผลการเข้ารหัสช่องสัญญาณมาตรฐาน 5G

ซอฟต์แวร์ Module เข้ารหัสช่องสัญญาณมาตรฐาน 5G จะใช้กับชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G และชุดทดสอบสมรรถนะในระบบสื่อสารไร้สาย ซึ่งจะประกอบด้วย 2 Module ได้แก่ PolarEncoderModule.py สำหรับการเข้ารหัสโพลาร์ และ LDPCEncoderModule.py สำหรับการเข้ารหัส LDPC อย่างไรก็ตาม ชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G และชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G จะมีการเรียกใช้ Module ย่อยที่แตกต่างกัน ดังนี้

4.2.3.1 ผลการทดสอบซอฟต์แวร์ Module สำหรับชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G

กรณีช่องสัญญาณกายภาพ PUCCH PDCCH และ PBCH เมื่อซอฟต์แวร์ GUI ได้รับคำสั่งให้ดำเนินการเข้ารหัสจะทำการเรียกใช้ซอฟต์แวร์ PolarEncoderModule.py โดยมีอินพุตคือชนิดของช่องสัญญาณกายภาพ ความยาวรหัส และบิตข้อมูลแบบไบนารี จากนั้นเรียกใช้ซอฟต์แวร์ Module

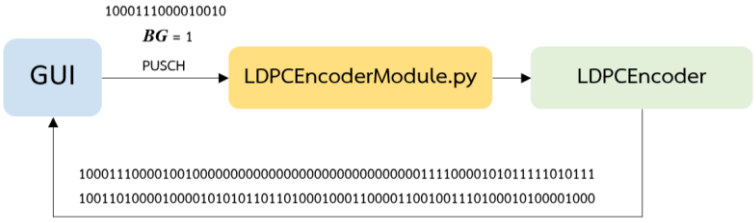
ย่อย PolarSequencing และ PolarEncoding เพื่อดำเนินการแทรกบิตแช่แข็ง (Frozen Bit) และดำเนินการเข้ารหัสโพลาร์ ตามลำดับ เมื่อกระบวนการเข้ารหัสเสร็จสิ้นซอฟต์แวร์ Module ย่อยทั้งสองจะส่งผลลัพธ์กลับไปแสดงผลยังซอฟต์แวร์ GUI ดังแสดงในรูปที่ 4.144



รูปที่ 4.144 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เข้ารหัสสำหรับชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUCCH PDCCH และ PBCH

จากรูปที่ 4.144 แสดงตัวอย่างผลลัพธ์การเข้ารหัสกรณีช่องสัญญาณ PUCCH โดยกำหนดให้มีความยาวรหัส 32 บิต และมีบิตอินพุตคือ ‘1000111000010010’ ผลลัพธ์ที่ได้จาก Module ย่อย PolarSequencing เป็นบิตข้อมูลแบบใบนารีที่มีการแทรกบิตแช่แข็งให้มีความยาว 32 บิต และผลลัพธ์ที่ได้จาก Module ย่อย PolarEncoding เป็นคำรหัสโพลาร์ที่มีความยาว 32 บิต สำหรับช่องสัญญาณกายภาพ PDCCH และ PBCH จะให้ผลลัพธ์เช่นเดียวกับกรณีช่องสัญญาณกายภาพ PUCCH เนื่องจากในมาตรฐานกำหนดให้ใช้ Module เดียวกัน

กรณีช่องสัญญาณกายภาพ PUSCH และ PDSCH เมื่อซอฟต์แวร์ GUI ได้รับคำสั่งให้ดำเนินการเข้ารหัสจะทำการเรียกใช้ซอฟต์แวร์ LDPCEncoderModule.py โดยมีอินพุตคือชนิดของช่องสัญญาณการภาพ ความยาวรหัส และบิตข้อมูลแบบไบนารี จากนั้นเรียกใช้ซอฟต์แวร์ Module ย่อย LDPCEncoder เพื่อดำเนินการเข้ารหัส LDPC เมื่อกระบวนการเข้ารหัสเสร็จสิ้นซอฟต์แวร์ Module ย่อยจะส่งผลลัพธ์กลับไปแสดงผลยังซอฟต์แวร์ GUI ดังแสดงในรูปที่ 4.145



รูปที่ 4.145 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เข้ารหัสสำหรับชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUSCH และ PDSCH

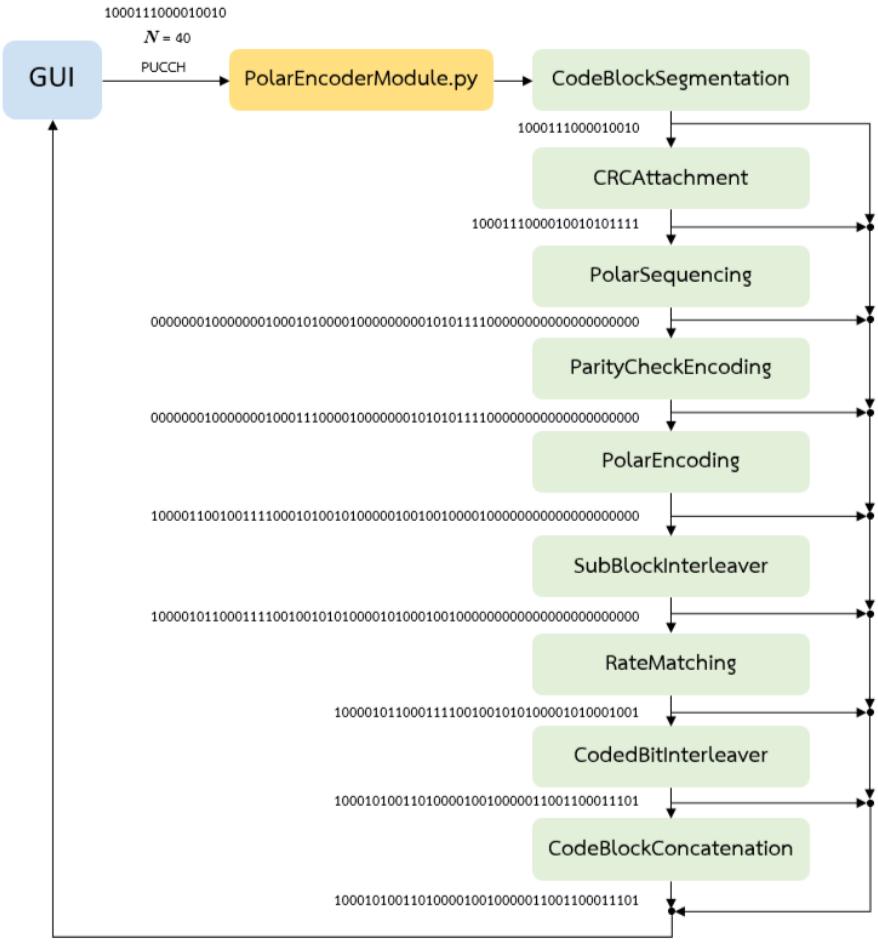
จากรูปที่ 4.145 แสดงตัวอย่างผลลัพธ์การเข้ารหัสกรณีช่องสัญญาณ PUSCH โดยกำหนดให้ใช้กราฟฐาน 1 และมีบิตอินพุตคือ ‘1000111000010010’ ผลลัพธ์จาก Module ย่อย LDPCEncoder เป็นคำรหัส LDPC ที่มีความยาว 136 บิต สำหรับช่องสัญญาณกายภาพ PDSCH จะให้ผลลัพธ์เช่นเดียวกับกรณีช่องสัญญาณกายภาพ PUSCH เนื่องจากในมาตรฐานกำหนดให้ใช้ Module เดียวกัน

4.2.3.2 ผลการทดสอบซอฟต์แวร์ Module สำหรับชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G

กรณีช่องสัญญาณกายภาพ PUCCH เมื่อซอฟต์แวร์ GUI ได้รับคำสั่งให้ดำเนินการเข้ารหัสและปรับอัตรารหัสจะทำการเรียกใช้ซอฟต์แวร์ PolarEncoderModule.py โดยมีอินพุตคือชนิดของช่องสัญญาณกายภาพ ความยาวรหัสหลังการพังค์เจอร์ และบิตข้อมูลแบบไบนารี จากนั้นเรียกใช้ซอฟต์แวร์ Module ย่อยดังนี้

1) CodeBlockSegmentation เพื่อดำเนินการแบ่งย่อยบล็อกข้อมูล
2) CRCAttachment เพื่อดำเนินการเข้ารหัส CRC
3) PolarSequencing เพื่อดำเนินการแทรกบิตแช่แข็ง
4) ParityCheckEncoding เพื่อดำเนินการเข้ารหัสพาริตี
5) PolarEncoding เพื่อดำเนินการเข้ารหัสโพลาร์
6) SubBlockInterleaver เพื่อดำเนินการแทรกสลับบล็อกย่อย
7) RateMatching เพื่อดำเนินการปรับอัตรารหัส
8) CodedBitInterleaver เพื่อดำเนินการแทรกสลับบิตที่ถูกเข้ารหัส
9) CodeBlockConcatenation เพื่อดำเนินการต่อบล็อกคำรหัส

เมื่อกระบวนการเข้ารหัสและปรับอัตรารหัสเสร็จสิ้นซอฟต์แวร์ Module ย่อยทั้งหมดจะส่งผลลัพธ์กลับไปแสดงผลยังซอฟต์แวร์ GUI ดังแสดงในรูปที่ 4.146



รูปที่ 4.146 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เข้ารหัสสำหรับชุดการเรียนรู้กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUCCH

จากรูปที่ 4.146 แสดงตัวอย่างผลลัพธ์การเข้ารหัสกรณีช่องสัญญาณ PUCCH โดยกำหนดให้ใช้ความยาวรหัสหลังการพังค์เจอร์ 40 บิต และมีบิตอินพุตคือ ‘1000111000010010’ ผลลัพธ์จาก Module ย่อยต่าง ๆ มีรายละเอียดดังนี้

1) CodeBlockSegmentation ผลลัพธ์เป็นบิตข้อมูลแบบไบนารีที่มีความยาว 16 บิต จำนวน 1 บล็อก เนื่องจากกรณีนี้ไม่มีการแบ่งย่อยบล็อกข้อมูล

2) CRCAttachment ผลลัพธ์เป็นคำรหัส CRC แบบไบนารีที่มีความยาว 22 บิต ซึ่งประกอบด้วยบิตข้อมูล 16 บิต และบิต CRC จำนวน 6 บิต

3) PolarSequencing ผลลัพธ์เป็นคำรหัส CRC ที่มีการแทรกบิตแซ่แข็งให้มีความยาว 64 บิต

4) ParityCheckEncoding ผลลัพธ์เป็นคำรหัสพาริตีที่มีการแทนที่บิตพาริตีในตำแหน่งของบิตแซ่แข็งบางตำแหน่ง

5) PolarEncoding ผลลัพธ์เป็นคำรหัสโพลาร์ที่มีความยาว 64 บิต

6) SubBlockInterleaver ผลลัพธ์เป็นคำรหัสโพลาร์ที่มีการสลับตำแหน่ง

7) RateMatching ผลลัพธ์เป็นคำรหัสโพลาร์ที่ถูกปรับอัตรารหัสให้มีความยาวรหัส 40 บิต

8) CodedBitInterleaver ผลลัพธ์เป็นคำรหัสโพลาร์ความยาว 40 บิต ที่มีการแทรกสลับบิต

9) CodeBlockConcatenation ผลลัพธ์เป็นการรวมกันของคำรหัสโพลาร์ในแต่ละบล็อก ซึ่งในที่นี้มีเพียง 1 บล็อก

กรณีช่องสัญญาณกายภาพ PDCCH เมื่อซอฟต์แวร์ GUI ได้รับคำสั่งให้ดำเนินการเข้ารหัสและปรับอัตรารหัสจะทำการเรียกใช้ซอฟต์แวร์ PolarEncoderModule.py โดยมีอินพุตคือชนิดของช่องสัญญาณกายภาพ ความยาวรหัสหลังการพังค์เจอร์ บิต RNTI และบิตข้อมูลแบบไบนารี จากนั้นเรียกใช้ซอฟต์แวร์ Module ย่อยดังนี้

1) CRCAttachment เพื่อดำเนินการเข้ารหัส CRC

2) CRCScrambling เพื่อดำเนินการสแครมคำรหัส CRC เข้ากับบิต RNTI

3) CRCInterleaver เพื่อดำเนินการแทรกสลับคำรหัส CRC

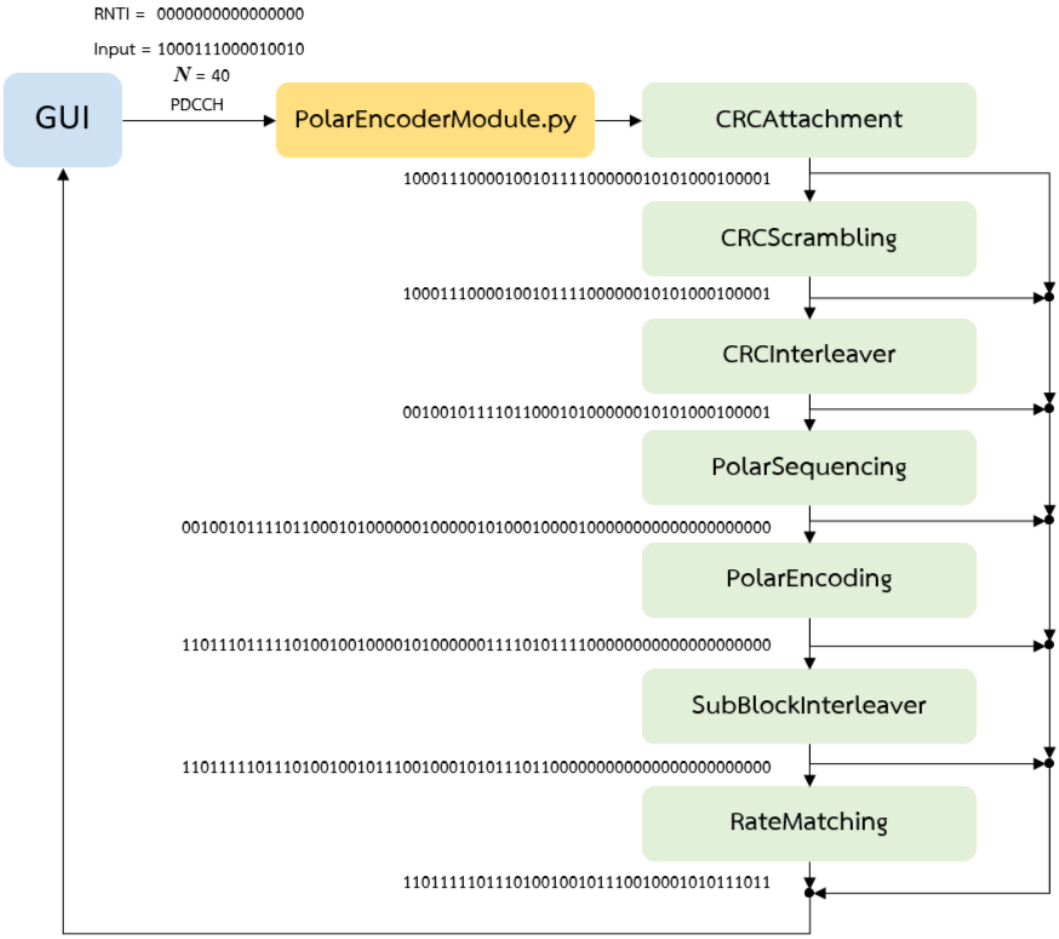
4) PolarSequencing เพื่อดำเนินการแทรกบิตแช่แข็ง

5) PolarEncoding เพื่อดำเนินการเข้ารหัสโพลาร์

6) SubBlockInterleaver เพื่อดำเนินการแทรกสลับบล็อกย่อย

7) RateMatching เพื่อดำเนินการปรับอัตรารหัส

เมื่อกระบวนการเข้ารหัสและปรับอัตรารหัสเสร็จสิ้นซอฟต์แวร์ Module ย่อยทั้งหมดจะส่งผลลัพธ์กลับไปแสดงผลยังซอฟต์แวร์ GUI ดังแสดงในรูปที่ 4.147



รูปที่ 4.147 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เข้ารหัสสำหรับชุดการเรียนรู้ กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PDCCH

จากรูปที่ 4.147 แสดงตัวอย่างผลลัพธ์การเข้ารหัสกรณีช่องสัญญาณ PDCCH โดยกำหนดให้ใช้ความยาวรหัสหลังการพังค์เจอร์ 40 บิต บิต RNTI คือ ‘0000000000000000’ และมีบิตอินพุตคือ ‘1000111000010010’ ผลลัพธ์จาก Module ย่อยต่าง ๆ มีรายละเอียดดังนี้

1) CRCAttachment ผลลัพธ์เป็นคำรหัส CRC แบบไบนารีที่มีความยาว 40 บิต ซึ่งประกอบด้วยบิตข้อมูล 16 บิต และบิต CRC จำนวน 24 บิต

2) CRCScrambling ผลลัพธ์เป็นคำรหัส CRC ที่ถูกสแครมเข้ากับบิต RNTI ในที่นี้บิต RNTI เป็น 0 ทั้งหมด ดังนั้นผลลัพธ์ที่ได้จากการสแครม CRC จะเท่ากับผลลัพธ์ของการเข้ารหัส CRC

3) CRCInterleaver ผลลัพธ์เป็นคำรหัสสแครม CRC ที่มีการแทรกสลับบิต CRC ให้อยู่ในระหว่างบิตข้อมูล

4) PolarSequencing ผลลัพธ์เป็นคำรหัส Interleaved CRC ที่มีการแทรกบิตแช่แข็งให้มีความยาว 64 บิต

5) PolarEncoding ผลลัพธ์เป็นคำรหัสโพลาร์ที่มีความยาว 64 บิต

6) SubBlockInterleaver ผลลัพธ์เป็นคำรหัสโพลาร์ที่มีการสลับตำแหน่ง

7) RateMatching ผลลัพธ์เป็นคำรหัสโพลาร์ที่ถูกปรับอัตรารหัสให้มีความยาวรหัส 40 บิต

กรณีช่องสัญญาณกายภาพ PBCH เมื่อซอฟต์แวร์ GUI ได้รับคำสั่งให้ดำเนินการเข้ารหัสและปรับอัตรารหัสจะทำการเรียกใช้ซอฟต์แวร์ PolarEncoderModule.py โดยมีอินพุตคือชนิดของช่องสัญญาณกายภาพ ความยาวรหัสหลังการพังค์เจอร์ และบิตข้อมูลแบบไบนารี จากนั้นเรียกใช้ซอฟต์แวร์ Module ย่อยดังนี้

1) CRCAttachment เพื่อดำเนินการเข้ารหัส CRC

2) CRCInterleaver เพื่อดำเนินการแทรกสลับคำรหัส CRC

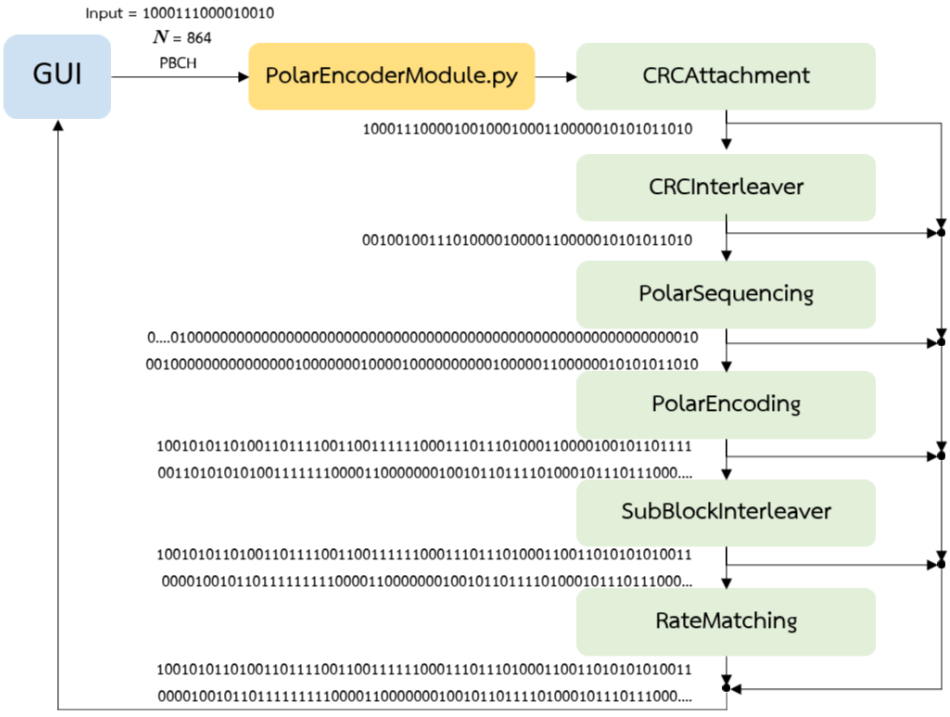
3) PolarSequencing เพื่อดำเนินการแทรกบิตแช่แข็ง

4) PolarEncoding เพื่อดำเนินการเข้ารหัสโพลาร์

5) SubBlockInterleaver เพื่อดำเนินการแทรกสลับบล็อกย่อย

6) RateMatching เพื่อดำเนินการปรับอัตรารหัส

เมื่อกระบวนการเข้ารหัสและปรับอัตรารหัสเสร็จสิ้นซอฟต์แวร์ Module ย่อยทั้งหมดจะส่งผลลัพธ์กลับไปแสดงผลยังซอฟต์แวร์ GUI ดังแสดงในรูปที่ 4.148



รูปที่ 4.148 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เข้ารหัสสำหรับชุดการเรียนรู้กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PBCH

จากรูปที่ 4.148 แสดงตัวอย่างผลลัพธ์การเข้ารหัสกรณีช่องสัญญาณ PBCH โดยความยาวรหัสหลังการพังค์เจอร์ถูกกำหนดไว้ตายตัวเป็น 864 บิต และมีบิตอินพุตคือ ‘1000111000010010’ ผลลัพธ์จาก Module ย่อยต่าง ๆ มีรายละเอียดดังนี้

1) CRCAttachment ผลลัพธ์เป็นคำรหัส CRC แบบไบนารีที่มีความยาว 40 บิต ซึ่งประกอบด้วยบิตข้อมูล 16 บิต และบิต CRC จำนวน 24 บิต

2) CRCInterleaver ผลลัพธ์เป็นคำรหัส CRC ที่มีการแทรกสลับบิต CRC ให้อยู่ในระหว่างบิตข้อมูล

3) PolarSequencing ผลลัพธ์เป็นคำรหัส Interleaved CRC ที่มีการแทรกบิตแช่แข็งให้มีความยาว 512 บิต

4) PolarEncoding ผลลัพธ์เป็นคำรหัสโพลาร์ที่มีความยาว 512 บิต

5) SubBlockInterleaver ผลลัพธ์เป็นคำรหัสโพลาร์ที่มีการสลับตำแหน่ง

6) RateMatching ผลลัพธ์เป็นคำรหัสโพลาร์ที่ถูกปรับอัตรารหัสให้มีความยาวรหัส 864 บิต
หมายเหตุ: เนื่องจากผลลัพธ์ของบาง Module ย่อยมีความยาวมาก ในที่นี้จึงแสดงบิตของผลลัพธ์เพียงบางส่วน

กรณีช่องสัญญาณกายภาพ PUSCH และ PDSCH เมื่อซอฟต์แวร์ GUI ได้รับคำสั่งให้ดำเนินการเข้ารหัสและปรับอัตรารหัสจะทำการเรียกใช้ซอฟต์แวร์ LDPCEncoderModule.py โดยมีอินพุตคือชนิดของช่องสัญญาณกายภาพ พารามิเตอร์การเข้ารหัส และบิตข้อมูลแบบไบนารี จากนั้นเรียกใช้ซอฟต์แวร์ Module ย่อยดังนี้

1) CRCEncoder1 เพื่อดำเนินการเข้ารหัส CRC ของบิตข้อมูลอินพุต

2) Segmentation เพื่อดำเนินการแบ่งย่อยบล็อกข้อมูล

3) CRCEncoder2 เพื่อดำเนินการเข้ารหัส CRC ของบล็อกข้อมูลย่อย

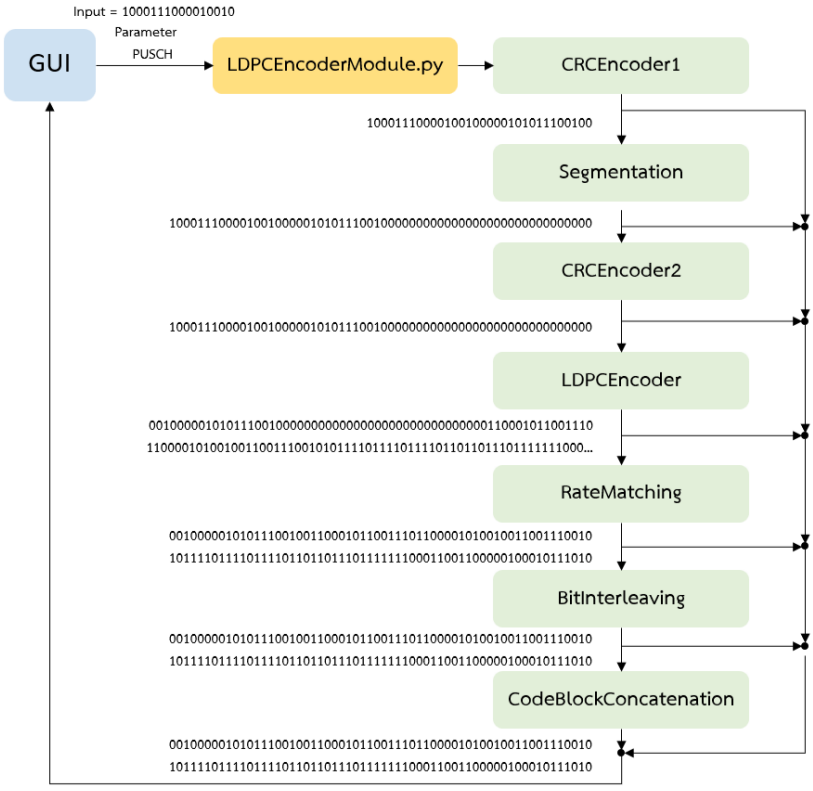
4) LDPCEncoder เพื่อดำเนินการเข้ารหัส LDPC

5) RateMatching เพื่อดำเนินการปรับอัตรารหัส

5) BitInterleaving เพื่อดำเนินการแทรกสลับบิตในคำรหัส LDPC

6) CodeBlockConcatenation เพื่อดำเนินการต่อบล็อกคำรหัส

เมื่อกระบวนการเข้ารหัสและปรับอัตรารหัสเสร็จสิ้นซอฟต์แวร์ Module ย่อยทั้งหมดจะส่งผลลัพธ์กลับไปแสดงผลยังซอฟต์แวร์ GUI ดังแสดงในรูปที่ 4.149



รูปที่ 4.149 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เข้ารหัสสำหรับชุดการเรียนรู้กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUSCH

จากรูปที่ 4.149 แสดงตัวอย่างผลลัพธ์การเข้ารหัสกรณีช่องสัญญาณ PUSCH โดยกำหนดพารามิเตอร์ Rate = 0.5, Qm = 1, N_L = 1, I_LBRM = 0, rv_id = 0 และมีบิตอินพุตคือ‘100011 1000010010’ ผลลัพธ์จาก Module ย่อยต่าง ๆ มีรายละเอียดดังนี้

1) CRCEncoder1 ผลลัพธ์เป็นคำรหัส CRC แบบไบนารีที่มีความยาว 32 บิต ซึ่งประกอบด้วยบิตข้อมูล 16 บิต และบิต CRC จำนวน 16 บิต

2) Segmentation ผลลัพธ์เป็นคำรหัส CRC ที่มีความยาว 32 บิต จำนวน 1 บล็อก เนื่องจากกรณีนี้ไม่มีการแบ่งย่อยบล็อกข้อมูล

3) CRCEncoder2 ผลลัพธ์เป็นคำรหัส CRC ที่มีความยาว 32 บิต จำนวน 1 บล็อก เช่นเดียวกับเอาต์พุตของซอฟต์แวร์ Module Segmentation เนื่องจากกรณีที่ไม่มีการแบ่งย่อยบล็อกข้อมูลจะไม่มีการเข้ารหัส CRC ครั้งที่สอง

4) LDPCEncoder ผลลัพธ์เป็นคำ LDPC ที่มีความยาว 659 บิต

5) RateMatching ผลลัพธ์เป็นคำรหัส LDPC ที่ถูกปรับอัตรารหัสให้มีความยาว 120 บิต

6) CodeBlockConcatenation ผลลัพธ์เป็นการรวมกันของคำรหัส LDPC ในแต่ละบล็อก ซึ่งในที่นี้มีเพียง 1 บล็อก

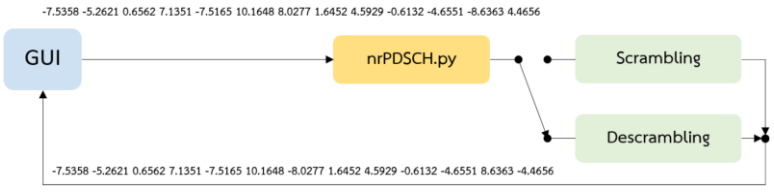
หมายเหตุ: เนื่องจากผลลัพธ์ของบาง Module ย่อยมีความยาวมาก ในที่นี้จึงแสดงบิตของผลลัพธ์เพียงบางส่วน และสำหรับช่องสัญญาณกายภาพ PDSCH จะให้ผลลัพธ์เช่นเดียวกับกรณีช่องสัญญาณกายภาพ PUSCH เนื่องจากในมาตรฐานกำหนดให้ใช้ Module เดียวกัน

4.2.4 ผลการทดสอบซอฟต์แวร์ Module เพื่อแสดงผลการสแครมหรือดีสแครมมาตรฐาน 5G

ซอฟต์แวร์ Module ดำเนินการสแครมหรือดีสแครมโดยสร้างลำดับแบบสุ่มเทียมจะใช้กับชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G และชุดทดสอบสมรรถนะในระบบสื่อสารไร้สาย เมื่อซอฟต์แวร์ GUI ได้รับคำสั่งให้ดำเนินการสแครมหรือดีสแครมโดยสร้างลำดับแบบสุ่มเทียมจะทำการเรียกใช้ซอฟต์แวร์ nrPDSCHPRBS.py โดยมีอินพุตคือคำรหัสแบบไบนารี (สำหรับกรณีการสแครม) หรือค่า LLR (สำหรับกรณีการดีสแครม) จากนั้นซอฟต์แวร์ nrPDSCHPRBS.py จะดำเนินการลำดับการสแครมรูปแบบของไบนารี และนำไปดำเนินการสแครมกับคำรหัส หรือนำไปดีสแครมกับค่า LLR ที่ได้จากการดีมอดูเลชัน เมื่อกระบวนการดีสแครมเสร็จสิ้นผลลัพธ์จะถูกส่งกลับไปแสดงผลยังซอฟต์แวร์ GUI ดังแสดงในรูปที่ 4.150 - รูปที่ 4.151

รูปที่ 4.150 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เพื่อแสดงผลการสแครม

จากรูปที่ 4.150 แสดงตัวอย่างผลลัพธ์การสแครมโดยสร้างลำดับแบบสุ่มเทียม ซึ่งกำหนดให้คำรหัสที่เป็นอินพุตของ Module คือ ‘10001010011010000100100000110011000111101’ ดังนั้น ผลลัพธ์ที่ได้จะเป็นคำรหัสสแครมที่มีความยาว 40 บิต ซึ่งเท่ากับความยาวของคำรหัสอินพุต

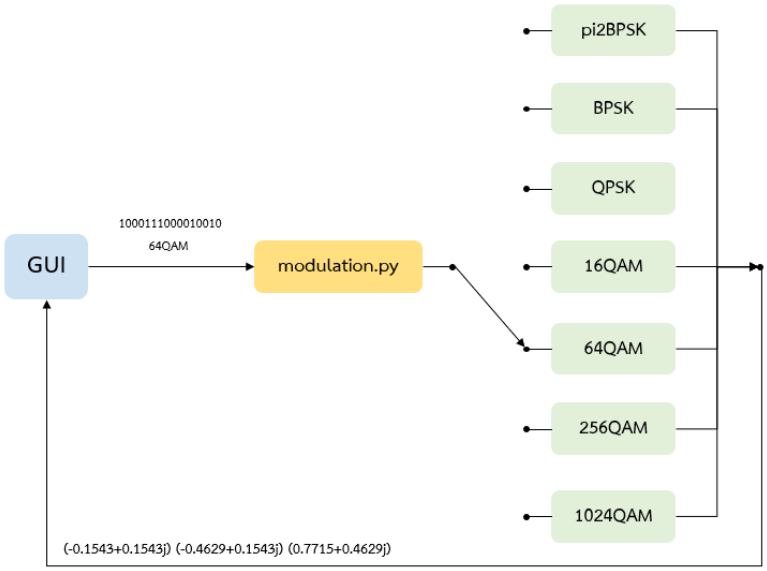


รูปที่ 4.151 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เพื่อแสดงผลการดีสแครม

จากรูปที่ 4.151 แสดงตัวอย่างผลลัพธ์การดีสแครมโดยสร้างลำดับแบบสุ่มเทียม ซึ่งกำหนดให้ค่า LLR ที่เป็นอินพุตของ Module คือ ‘-7.5358 -5.2621 0.6562 7.1351 -7.5165 10.1648 8.0277 1.6452 4.5929 -0.6132 -4.6551 -8.6363 4.4656’ ดังนั้น ผลลัพธ์ที่ได้จะเป็นค่า LLR ที่ถูกดีสแครมที่มีความยาว 13 บิต ซึ่งเท่ากับความยาวของค่า LLR อินพุต

4.2.5 ผลการทดสอบซอฟต์แวร์ Module เพื่อแสดงผลการมอดูเลชันมาตรฐาน 5G

ซอฟต์แวร์ Module มอดูเลชันจะใช้กับชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G ชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G และชุดทดสอบสมรรถนะในระบบสื่อสารไร้สาย เมื่อซอฟต์แวร์ GUI ได้รับคำสั่งให้ดำเนินการมอดูเลชันมาตรฐาน 5G จะทำการเรียกใช้ซอฟต์แวร์ modulation.py โดยมีอินพุตคือรูปแบบการมอดูเลชันและข้อมูลอินพุต หรือคำรหัส หรือสัญญาณสแครม จากนั้นซอฟต์แวร์ modulation.py จะดำเนินการมอดูเลชันตามรูปแบบการมอดูเลชันที่กำหนดโดยเรียกใช้งานซอฟต์แวร์ Module ย่อยของรูปแบบการมอดูเลชันนั้น ๆ และส่งผลลัพธ์ของการมอดูเลชันกลับไปยังซอฟต์แวร์ GUI ในรูปแบบของตัวเลขเชิงซ้อนซึ่งแสดงค่าของแต่ละสัญลักษณ์ (Symbol) ดังแสดงในรูปที่ 4.152



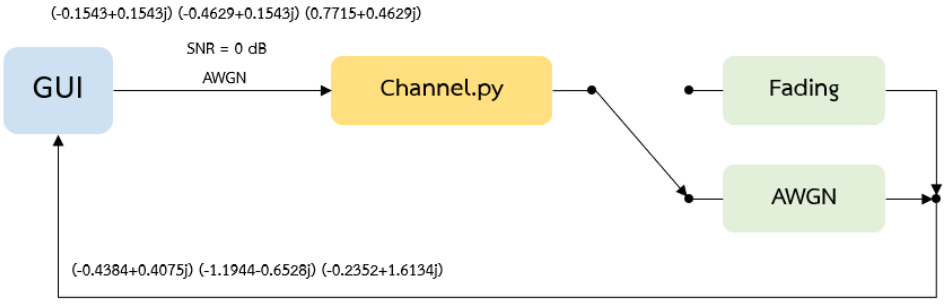
รูปที่ 4.152 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เพื่อแสดงการมอดูเลชันมาตรฐาน 5G

จากรูปที่ 4.152 แสดงผลลัพธ์การมอดูเลชันมาตรฐาน 5G โดยกำหนดให้บิตอินพุตคือ “100111000010010” และกำหนดรูปแบบการมอดูเลชันแบบ 64QAM ดังนั้น ผลลัพธ์ที่ได้จึงเป็นตัวเลขเชิงซ้อนซึ่งแสดงค่าของสัญลักษณ์จำนวน 3 สัญลักษณ์

4.2.6 ผลการทดสอบซอฟต์แวร์ Module เพื่อแสดงผลของช่องสัญญาณสื่อสาร

ซอฟต์แวร์ Module ช่องสัญญาณสื่อสารจะใช้กับชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G ชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G และชุดทดสอบสมรรถนะในระบบสื่อสารไร้สาย เมื่อซอฟต์แวร์ GUI ได้รับคำสั่งให้ดำเนินการทดสอบผลของช่องสัญญาณสื่อสาร จะทำการเรียกใช้ซอฟต์แวร์ Channel.py โดยมีอินพุตคือรูปแบบของช่องสัญญาณ ค่า SNR และสัญญาณมอดูเลชัน

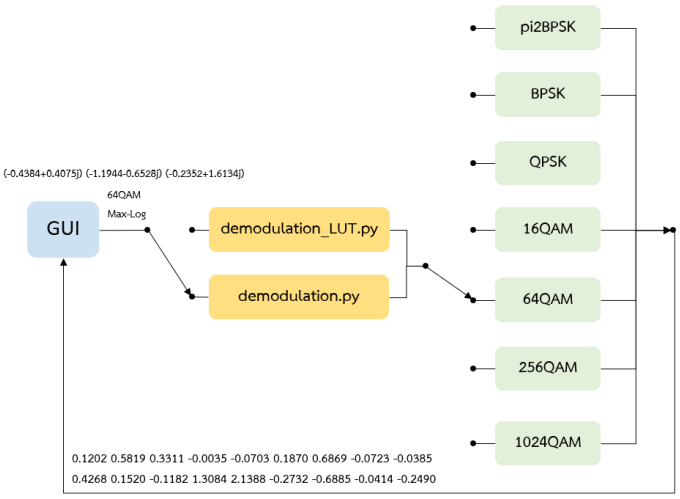
จากนั้นซอฟต์แวร์ Channel.py ดำเนินการรวมผลกระทบของช่องสัญญาณเข้ากับสัญญาณมอดูเลชัน โดยเรียกใช้ซอฟต์แวร์ Module ย่อยของช่องสัญญาณที่กำหนด และส่งผลลัพธ์กลับไปยังซอฟต์แวร์ GUI ในรูปแบบของตัวเลขเชิงซ้อนซึ่งแทนค่าของสัญลักษณ์ที่ได้รับผลกระทบจากช่องสัญญาณสื่อสาร รูปที่ 4.153 แสดงผลลัพธ์ของผลกระทบของช่องสัญญาณสื่อสาร โดยกำหนดให้สัญญาณมอดูเลตคือ ‘(-0.1543+0.1543j) (-0.4629+0.1543j) (0.7715+0.4629j)’ ส่งผ่านช่องสัญญาณแบบ AWGN Channel ที่มีค่า SNR = 0 dB ดังนั้น ผลลัพธ์ที่ได้จึงเป็นตัวเลขเชิงซ้อนซึ่งแทนค่าของสัญลักษณ์ที่ได้รับผลกระทบจากช่องสัญญาณสื่อสาร



รูปที่ 4.153 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เพื่อแสดงผลของช่องสัญญาณสื่อสาร

4.2.7 ผลการทดสอบซอฟต์แวร์ Module เพื่อแสดงผลการดีมอดูเลชันมาตรฐาน 5G

ซอฟต์แวร์ Module ดีมอดูเลชันจะใช้กับชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G ชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G และชุดทดสอบสมรรถนะในระบบสื่อสารไร้สาย เมื่อซอฟต์แวร์ GUI ได้รับคำสั่งให้ดำเนินการดีมอดูเลชันมาตรฐาน 5G โดยที่ผู้ใช้เลือกอัลกอริทึมการดีมอดูเลชันแบบ Max-Log ซอฟต์แวร์ GUI จะเรียกใช้ซอฟต์แวร์ demodulation.py แต่หากผู้ใช้เลือกอัลกอริทึมการดีมอดูเลชันแบบ LUT ซอฟต์แวร์ GUI จะเรียกใช้ซอฟต์แวร์ demodulation_LUT.py โดยทั้งสองกรณีมีอินพุตคือรูปแบบของการดีมอดูเลชัน และสัญญาณที่ผ่านช่องสัญญาณสื่อสาร จากนั้น ซอฟต์แวร์ demodulation.py หรือ demodulation_LUT.py จะดำเนินการดีมอดูเลชันสัญญาณโดยเรียกใช้ซอฟต์แวร์ Module ย่อยของรูปแบบการดีมอดูเลชันนั้น ๆ และส่งผลลัพธ์ของการดีมอดูเลชันกลับไปยังซอฟต์แวร์ GUI ในรูปแบบของตัวเลขจำนวนจริงซึ่งแทนค่า LLR ของบิตที่ได้จากการดีมอดูเลชัน รูปที่ 4.154 แสดงผลลัพธ์ของการดีมอดูเลชันมาตรฐาน 5G เมื่อสัญญาณที่ผ่านช่องสัญญาณสื่อสาร คือ “(-0.4384+0.4075j) (-1.1944-0.6528j) (-0.2352+1.6134j)” โดยกำหนดให้ใช้อัลกอริทึมการดีมอดูเลชันแบบ Max-log และรูปแบบการดีมอดูเลชันแบบ 64QAM ดังนั้น ผลลัพธ์ที่ได้จึงเป็นตัวเลขจำนวนจริงจำนวน 16 ค่าซึ่งเป็นค่า LLR ของบิตที่ได้จากการดีมอดูเลชัน



รูปที่ 4.154 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module เพื่อแสดงผลการดีมอดูเลชันมาตรฐาน 5G

4.2.8 ผลการทดสอบซอฟต์แวร์ Module เพื่อแสดงผลการถอดช่องสัญญาณรหัสมาตรฐาน 5G

ซอฟต์แวร์ Module ถอดรหัสช่องสัญญาณมาตรฐาน 5G จะใช้กับชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G ชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G และชุดทดสอบสมรรถนะในระบบสื่อสารไร้สาย ซึ่งจะประกอบด้วย 2 Module ได้แก่ PolarDecoderModule.py สำหรับการถอดรหัสโพลาร์ และ LDPCDecoderModule.py สำหรับการเข้ารหัส LDPC อย่างไรก็ตาม ชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G และชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G จะมีการเรียกใช้ Module ย่อยที่แตกต่างกัน ดังนี้

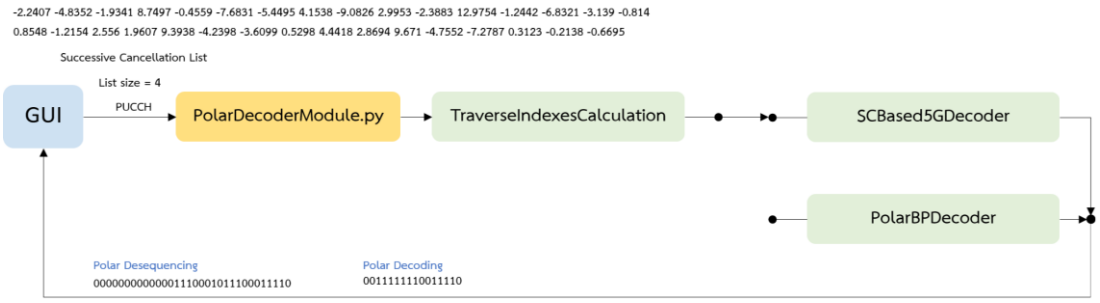
4.2.8.1 ผลการทดสอบซอฟต์แวร์ Module สำหรับชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G

กรณีช่องสัญญาณกายภาพ PUCCH PDCCH และ PBCH เมื่อซอฟต์แวร์ GUI ได้รับคำสั่งให้ดำเนินการถอดรหัสจะทำการเรียกใช้ซอฟต์แวร์ PolarDecoderModule.py โดยมีอินพุตคือชนิดของช่องสัญญาณกายภาพ อัลกอริทึมการถอดรหัส พารามิเตอร์การถอดรหัส และค่า LLR ที่ได้จากการดีมอดูเลชัน จากนั้นเรียกใช้ซอฟต์แวร์ Module ย่อย TraverseIndexesCalculation สำหรับคำนวณดัชนีการถอดรหัส และสำหรับกรณีที่ผู้ใช้เลือกอัลกอริทึมการเข้ารหัสแบบ Successive Cancellation หรือ Successive Cancellation List ซอฟต์แวร์ GUI จะเรียกใช้ Module ย่อย SCBased5GDecoder เพื่อดำเนินการถอดรหัสและดึงบิตข้อมูลที่ถอดรหัสได้ออกจากบิตแซ่แข็ง แต่สำหรับกรณีที่ผู้ใช้เลือกอัลกอริทึมการถอดรหัสแบบ Belief Propagation ซอฟต์แวร์ GUI จะเรียกใช้ Module ย่อย PolarBPDecoder เพื่อดำเนินการถอดรหัสและดึงบิตข้อมูลที่ถอดรหัสได้ออกจากบิตแซ่แข็ง จากนั้นจะส่งผลลัพธ์บิตที่ถอดรหัสได้ (Coded Bit) และบิตข้อมูลที่ถอดรหัสได้ (Information Coded Bit) กลับไปแสดงผลยังซอฟต์แวร์ GUI ดังแสดงในรูปที่ 4.155 แผนภาพการ

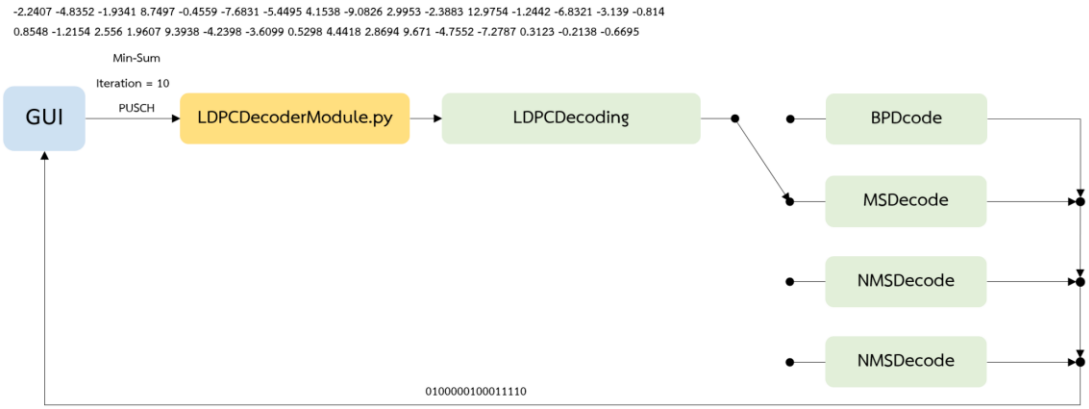
เรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUCCH PDCCH และ PBCH

จากรูปที่ 4.155 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUCCH PDCCH และ PBCH แสดงตัวอย่างผลลัพธ์การถอดรหัสกรณีช่องสัญญาณ PUCCH โดยกำหนดให้ใช้อัลกอริทึมการถอดรหัสแบบ Successive Cancellation List ที่มี List Size = 4 ผลลัพธ์ที่ได้จากถอดรหัสจะประกอบด้วย 2 ส่วน ได้แก่ บิตที่ถอดรหัสได้ '00000000000001110001011100011110' ซึ่งจะแสดงในส่วน Polar Decoding บนซอฟต์แวร์ GUI และบิตข้อมูลที่ถอดรหัสได้ '0011111110011110' ซึ่งจะแสดงในส่วน Polar Desequencing บนซอฟต์แวร์ GUI สำหรับช่องสัญญาณกายภาพ PDCCH และ PBCH จะให้ผลลัพธ์เช่นเดียวกับกรณีช่องสัญญาณกายภาพ PUCCH เนื่องจากในมาตรฐานกำหนดให้ใช้ Module เดียวกัน

กรณีช่องสัญญาณกายภาพ PUSCH และ PDSCH เมื่อซอฟต์แวร์ GUI ได้รับคำสั่งให้ดำเนินการถอดรหัสจะทำการเรียกใช้ซอฟต์แวร์ LDPCDecoderModule.py โดยมีอินพุตคือชนิดของช่องสัญญาณกายภาพ อัลกอริทึมการถอดรหัส พารามิเตอร์การถอดรหัส และค่า LLR จากนั้นเรียกใช้ซอฟต์แวร์ Module ย่อย LDPCDecoding เพื่อดำเนินการถอดรหัส LDPC เมื่อกระบวนการเข้ารหัสเสร็จสิ้นซอฟต์แวร์ Module ย่อยจะส่งผลลัพธ์กลับไปแสดงผลยังซอฟต์แวร์ GUI ดังแสดงในรูปที่ 4.156 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUSCH และ PDSCH



รูปที่ 4.155 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUCCH PDCCH และ PBCH



รูปที่ 4.156 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUSCH และ PDSCH

จากรูปที่ 4.156 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้การเข้าและถอดรหัสช่องสัญญาณมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUSCH และ PDSCH แสดงตัวอย่างผลลัพธ์การถอดรหัสกรณีช่องสัญญาณ PUSCH โดยกำหนดให้ใช้อัลกอลิทึมการถอดรหัสแบบ Min-Sum ที่มี Iteration = 10 ผลลัพธ์ที่ได้จาก Module ย่อย LDPCDecoding กรณี MSDecode เป็นบิตข้อมูล ‘0100000100011110’ ซึ่งจะแสดงบนซอฟต์แวร์ GUI

4.2.8.2 ผลการทดสอบซอฟต์แวร์ Module สำหรับชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G

กรณีช่องสัญญาณกายภาพ PUCCH เมื่อซอฟต์แวร์ GUI ได้รับคำสั่งให้ดำเนินการถอดรหัสและกู้คืนอัตรารหัสจะทำการเรียกใช้ซอฟต์แวร์ PolarDecoderModule.py โดยมีอินพุตคือชนิดของช่องสัญญาณกายภาพ อัลกอริทึมการถอดรหัส พารามิเตอร์การถอดรหัส และสัญญาณดีสแครม จากนั้นเรียกใช้ซอฟต์แวร์ Module ย่อย ดังนี้

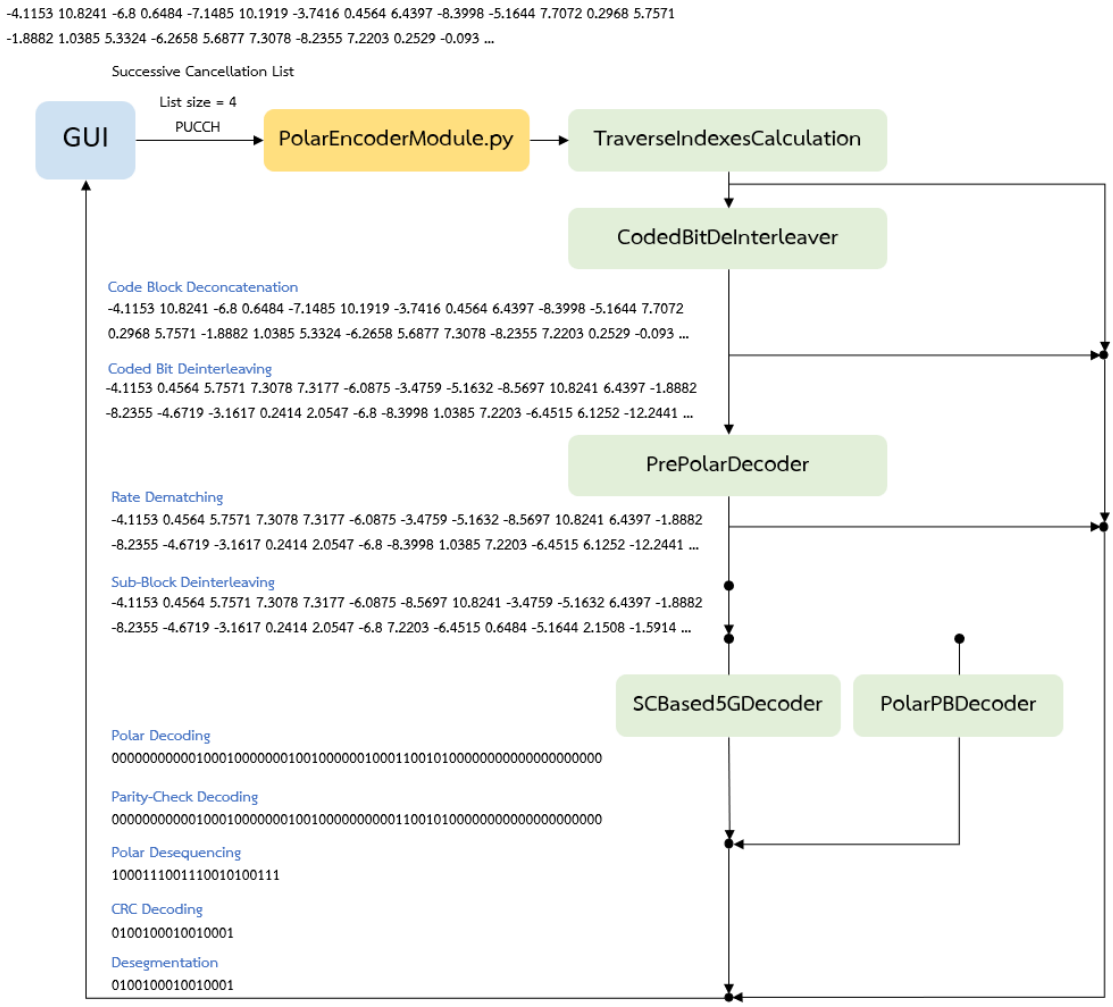
1) TraverseIndexesCalculation สำหรับคำนวณดัชนีการถอดรหัส

2) CodedBitDeInterleaver สำหรับดำเนินการแบ่งบล็อกย่อยและแทรกสลับบิตที่ถูกเข้ารหัส

3) PrePolarDecoder สำหรับดำเนินการกู้คืนอัตรารหัสและแทรกสลับบล็อกย่อย

4) SCBased5GDecoder สำหรับดำเนินการถอดรหัสโพลาร์ ถอดรหัสพาริตี นำบิตแช่แข็งออกจากบิตที่ถอดรหัส ถอดรหัส CRC และต่อบล็อกข้อมูล (สำหรับกรณีที่ผู้ใช้เลือกการถอดรหัสแบบ Successive Cancellation หรือ Successive Cancellation List)

5) PolarBPDecoder สำหรับดำเนินการถอดรหัสโพลาร์ ถอดรหัสพาริตี นำแช่แข็งออกจากบิตที่ถอดรหัส ถอดรหัส CRC และต่อบล็อกข้อมูล (สำหรับกรณีที่ผู้ใช้เลือกการถอดรหัสแบบ Belief Propagation)



รูปที่ 4.157 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUCCH

เมื่อกระบวนการถอดรหัสและกู้คืนอัตรารหัสเสร็จสิ้นซอฟต์แวร์ Module ย่อยทั้งหมดจะส่งผลลัพธ์กลับไปแสดงผลยังซอฟต์แวร์ GUI รูปที่ 4.157 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUCCH แสดงตัวอย่างผลลัพธ์การถอดรหัสกรณีช่องสัญญาณ PUCCH โดยกำหนดให้ใช้อัลกอริทึมการถอดรหัสแบบ Successive Cancellation List ที่มี List Size = 4 ผลลัพธ์จาก Module ย่อยต่าง ๆ มีรายละเอียดดังนี้

1) TraverseIndexesCalculation ผลลัพธ์เป็นค่าดัชนีการถอดรหัส

2) CodedBitDeInterleaver ผลลัพธ์ประกอบด้วย 2 ส่วน ได้แก่ ค่า LLR ที่ถูกแบ่งเป็นบล็อกย่อยซึ่งจะแสดงในส่วน Code Block Deconcatenation บนซอฟต์แวร์ GUI และค่า LLR ที่ถูกแทรกสลับ ซึ่งจะแสดงในส่วน Coded Bit Deinterleaving

3) PrePolarDecoder ผลลัพธ์ประกอบด้วย 2 ส่วน ได้แก่ ค่า LLR ที่มีการกู้คืนอัตรารหัส ซึ่งจะแสดงในส่วน Rate Dematching บนซอฟต์แวร์ GUI และค่า LLR ที่ถูกแทรกสลับบล็อกย่อย ซึ่งจะแสดงในส่วน Sub-Block Deinterleaving บนซอฟต์แวร์ GUI

4) SCBased5GDecoder ผลลัพธ์ประกอบด้วย 5 ส่วน ดังนี้

- บิตที่ได้จากการถอดรหัสโพลาร์ ซึ่งอยู่ในรูปแบบไบนารี โดยจะแสดงในส่วน Polar Decoding บนซอฟแวร์ GUI
- บิตที่ได้จากการถอดรหัสพาริตี ซึ่งอยู่ในรูปแบบไบนารี โดยจะแสดงในส่วน Parity-Check Decoding บนซอฟแวร์ GUI
- บิตที่ได้จากการนำบิตแซ่แข็งออก ซึ่งอยู่ในรูปแบบไบนารี โดยจะแสดงในส่วน Polar Desequencing บนซอฟแวร์ GUI
- บิตข้อมูลที่ได้จากการถอดรหัส CRC ซึ่งอยู่ในรูปแบบไบนารี โดยจะแสดงในส่วน CRC Decoding บนซอฟแวร์ GUI
- บิตข้อมูลทั้งหมดที่นำบิตข้อมูลแต่ละบล็อกมาต่อรวมกัน ซึ่งอยู่ในรูปแบบไบนารี โดยจะแสดงในส่วน Desegmentation บนซอฟแวร์ GUI อย่างไรก็ตามในกรณีนี้มีบล็อกของบิตข้อมูลเพียง 1 บล็อก

หมายเหตุ: เนื่องจากผลลัพธ์ของบาง Module ย่อยมีความยาวมาก ในที่นี้จึงแสดงผลลัพธ์เพียงบางส่วน

กรณีช่องสัญญาณกายภาพ PDCCH เมื่อซอฟต์แวร์ GUI ได้รับคำสั่งให้ดำเนินการถอดรหัสและกู้คืนอัตรารหัสจะทำการเรียกใช้ซอฟต์แวร์ PolarDecoderModule.py โดยมีอินพุตคือชนิดของช่องสัญญาณกายภาพ อัลกอริทึมการถอดรหัส พารามิเตอร์การถอดรหัส และสัญญาณดีสแครม จากนั้นเรียกใช้ซอฟต์แวร์ Module ย่อย ดังนี้

1) TraverseIndexesCalculation สำหรับคำนวณดัชนีการถอดรหัส

2) PrePolarDecoder สำหรับดำเนินการกู้คืนอัตรารหัสและแทรกสลับบล็อกย่อย

3) SCBased5GDecoder สำหรับดำเนินการถอดรหัสโพลาร์ นำบิตแซ่แข็งออกจากบิตที่ถอดรหัสได้ แทรกสลับ CRC ดีสแครม CRC และถอดรหัส CRC (สำหรับกรณีที่ผู้ใช้เลือกการถอดรหัสแบบ Successive Cancellation หรือ Successive Cancellation List)

4) PolarBPDecoder สำหรับดำเนินการถอดรหัสโพลาร์ นำบิตแซ่แข็งออกจากบิตที่ถอดรหัสได้ แทรกสลับ CRC ดีสแครม CRC และถอดรหัส CRC (สำหรับกรณีที่ผู้ใช้เลือกการถอดรหัสแบบ Belief Propagation)

เมื่อกระบวนการถอดรหัสและกู้คืนอัตรารหัสเสร็จสิ้นซอฟต์แวร์ Module ย่อยทั้งหมดจะส่งผลลัพธ์กลับไปแสดงผลยังซอฟต์แวร์ GUI รูปที่ 4.158 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PDCCH แสดงตัวอย่างผลลัพธ์การถอดรหัสกรณีช่องสัญญาณ PDCCH โดยกำหนดให้ใช้

อัลกอริทึมการถอดรหัสแบบ Belief Propagation ที่มี Iteration = 10 ผลลัพธ์จาก Module ย่อยต่าง ๆ มีรายละเอียดดังนี้

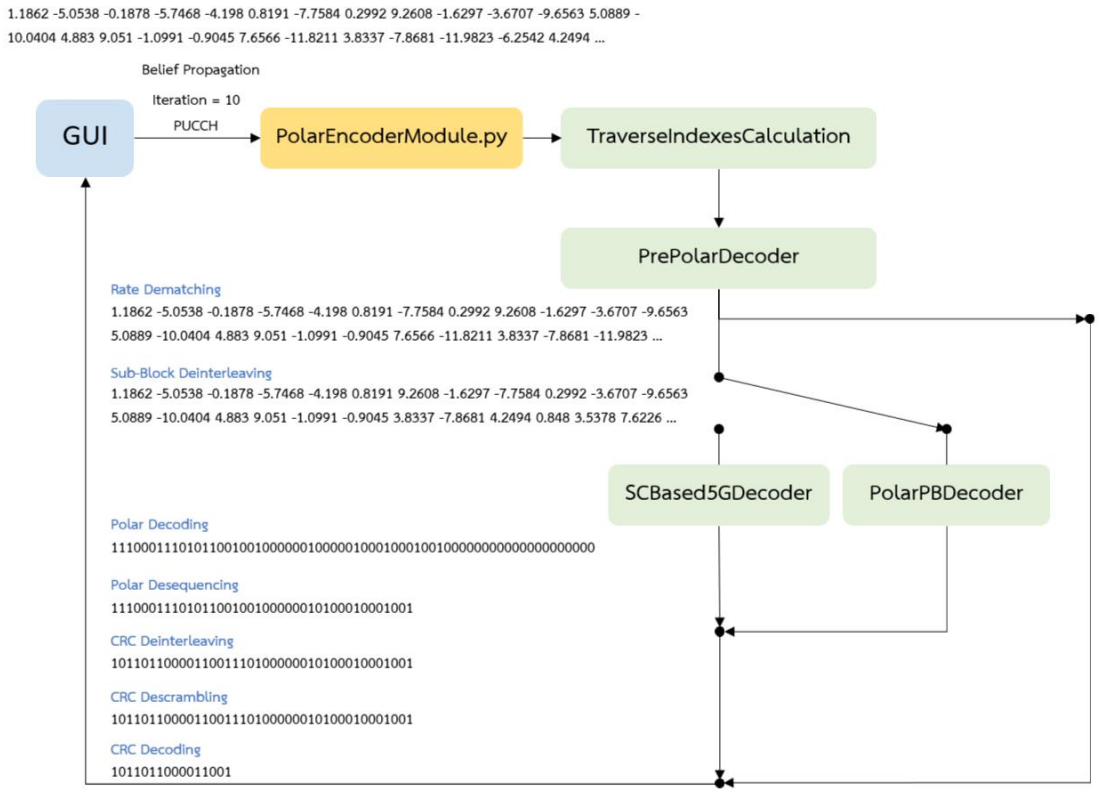
1) TraverseIndexesCalculation ผลลัพธ์เป็นค่าดัชนีการถอดรหัส

2) PrePolarDecoder ผลลัพธ์ประกอบด้วย 2 ส่วน ได้แก่ ค่า LLR ที่มีการกู้คืนอัตรารหัส ซึ่งจะแสดงในส่วน Rate Dematching บนซอฟต์แวร์ GUI และค่า LLR ที่ถูกแทรกสลับบล็อกย่อย ซึ่งจะแสดงในส่วน Sub-Block Deinterleaving บนซอฟต์แวร์ GUI

3) PolarBPDecoder ผลลัพธ์ประกอบด้วย 5 ส่วน ดังนี้

- บิตที่ได้จากการถอดรหัสโพลาร์ ซึ่งอยู่ในรูปแบบไบนารี โดยจะแสดงในส่วน Polar Decoding บนซอฟแวร์ GUI
- บิตที่ได้จากการนำบิตแซ่แข็งออก ซึ่งอยู่ในรูปแบบไบนารี โดยจะแสดงในส่วน Polar Desequencing บนซอฟแวร์ GUI
- บิตที่ได้จากการแทรกสลับ CRC ซึ่งอยู่ในรูปแบบไบนารี โดยจะแสดงในส่วน CRC Deinterleaving บนซอฟแวร์ GUI
- บิตที่ได้จากการดีสแครม CRC ซึ่งอยู่ในรูปแบบไบนารี โดยจะแสดงในส่วน CRC Descrambling บนซอฟแวร์ GUI
- บิตข้อมูลที่ได้จากการถอดรหัส CRC ซึ่งอยู่ในรูปแบบไบนารี โดยจะแสดงในส่วน CRC Decoding บนซอฟแวร์ GUI

หมายเหตุ: เนื่องจากผลลัพธ์ของบาง Module ย่อยมีความยาวมาก ในที่นี้จึงแสดงผลลัพธ์เพียงบางส่วน



รูปที่ 4.158 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PDCCH

กรณีช่องสัญญาณกายภาพ PBCH เมื่อซอฟต์แวร์ GUI ได้รับคำสั่งให้ดำเนินการถอดรหัสและกู้คืนอัตรารหัสจะทำการเรียกใช้ซอฟต์แวร์ PolarDecoderModule.py โดยมีอินพุตคือชนิดของช่องสัญญาณกายภาพ อัลกอริทึมการถอดรหัส พารามิเตอร์การถอดรหัส และสัญญาณดีสแครม จากนั้นเรียกใช้ซอฟต์แวร์ Module ย่อย ดังนี้

1) TraverseIndexesCalculation สำหรับคำนวณดัชนีการถอดรหัส

2) PrePolarDecoder สำหรับดำเนินการกู้คืนอัตรารหัสและแทรกสลับบล็อกย่อย

3) SCBased5GDecoder สำหรับดำเนินการถอดรหัสโพลาร์ นำบิตแซ่แข็งออกจากบิตที่ถอดรหัสได้ แทรกสลับ CRC และถอดรหัส CRC (สำหรับกรณีที่ผู้ใช้เลือกการถอดรหัสแบบ Successive Cancellation หรือ Successive Cancellation List)

4) PolarBPDecoder สำหรับดำเนินการถอดรหัสโพลาร์ นำบิตแซ่แข็งออกจากบิตที่ถอดรหัสได้ แทรกสลับ CRC และถอดรหัส CRC (สำหรับกรณีที่ผู้ใช้เลือกการถอดรหัสแบบ Belief Propagation)

เมื่อกระบวนการถอดรหัสและกู้คืนอัตรารหัสเสร็จสิ้นซอฟต์แวร์ Module ย่อยทั้งหมดจะส่งผลลัพธ์กลับไปแสดงผลยังซอฟต์แวร์ GUI รูปที่ 4.159 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PBCH แสดงตัวอย่างผลลัพธ์การ

ถอดรหัสกรณีช่องสัญญาณ PBCH โดยกำหนดให้ใช้อัลกอริทึมการถอดรหัสแบบ Successive Cancellation ผลลัพธ์จาก Module ย่อยต่าง ๆ มีรายละเอียดดังนี้

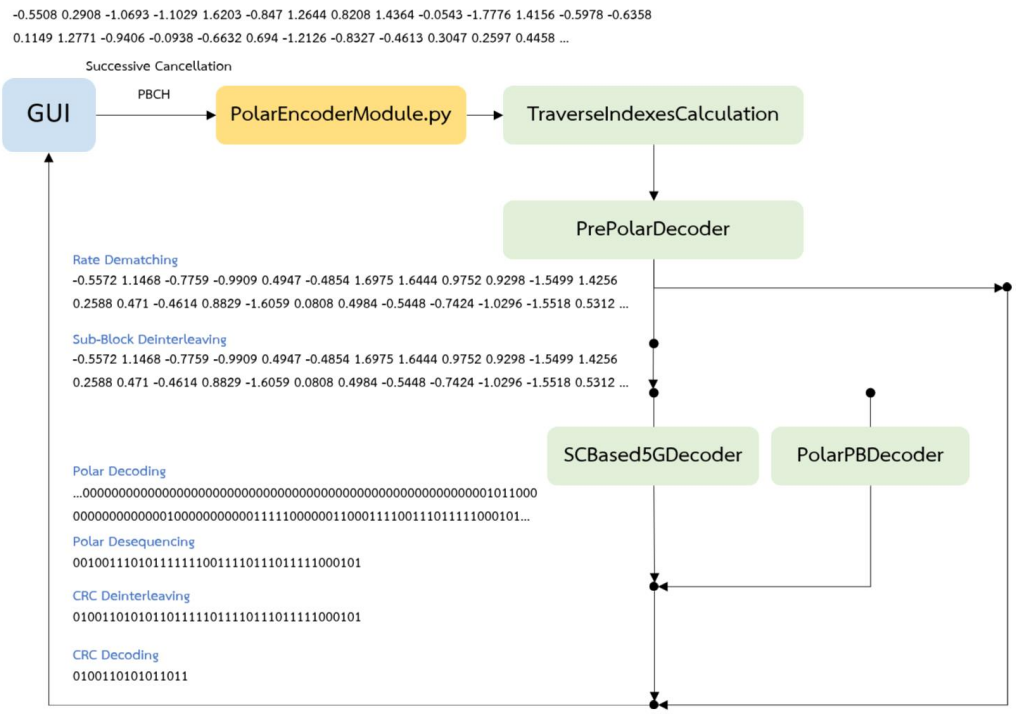
1) TraverseIndexesCalculation ผลลัพธ์เป็นค่าดัชนีการถอดรหัส

2) PrePolarDecoder ผลลัพธ์ประกอบด้วย 2 ส่วน ได้แก่ ค่า LLR ที่มีการกู้คืนอัตรารหัส ซึ่งจะแสดงในส่วน Rate Dematching บนซอฟต์แวร์ GUI และค่า LLR ที่ถูกแทรกสลับบล็อกย่อย ซึ่งจะแสดงในส่วน Sub-Block Deinterleaving บนซอฟต์แวร์ GUI

3) PolarBPDecoder ผลลัพธ์ประกอบด้วย 4 ส่วน ดังนี้

- บิตที่ได้จากการถอดรหัสโพลาร์ ซึ่งอยู่ในรูปแบบไบนารี โดยจะแสดงในส่วน Polar Decoding บนซอฟแวร์ GUI
- บิตที่ได้จากการนำบิตแซ่แข็งออก ซึ่งอยู่ในรูปแบบไบนารี โดยจะแสดงในส่วน Polar Desequencing บนซอฟแวร์ GUI
- บิตที่ได้จากการแทรกสลับ CRC ซึ่งอยู่ในรูปแบบไบนารี โดยจะแสดงในส่วน CRC Deinterleaving บนซอฟแวร์ GUI
- บิตข้อมูลที่ได้จากการถอดรหัส CRC ซึ่งอยู่ในรูปแบบไบนารี โดยจะแสดงในส่วน CRC Decoding บนซอฟแวร์ GUI

หมายเหตุ: เนื่องจากผลลัพธ์ของบาง Module ย่อยมีความยาวมาก ในที่นี้จึงแสดงผลลัพธ์เพียงบางส่วน



รูปที่ 4.159 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PBCH

กรณีช่องสัญญาณกายภาพ PUSCH และ PDSCH เมื่อซอฟต์แวร์ GUI ได้รับคำสั่งให้ดำเนินการถอดรหัสและกู้คืนอัตรารหัสจะทำการเรียกใช้ซอฟต์แวร์ LDPCDecoderModule.py โดยมีอินพุตคือชนิดของช่องสัญญาณกายภาพ อัลกอริทึมการถอดรหัส พารามิเตอร์การถอดรหัส และสัญญาณดีสแครม จากนั้นเรียกใช้ซอฟต์แวร์ Module ย่อยดังนี้

1) DeCodeBlockConcatenation เพื่อดำเนินการแบ่งค่า LLR ออกเป็นบล็อกย่อย

2) DeBitInterleaving เพื่อดำเนินการแทรกสลับค่า LLR

3) DeRateMatching เพื่อดำเนินการกู้คืนอัตรารหัส

4) LDPCDecoding เพื่อดำเนินการถอดรหัส LDPC

5) CRCDecoder2 เพื่อดำเนินการถอดรหัส CRC ของแต่ละบล็อกย่อย

5) DeSegmentation เพื่อดำเนินการรวมบิตที่ได้จากการถอดรหัส CRC ในแต่ละบล็อกย่อย

6) CRCDecoder1 เพื่อดำเนินการถอดรหัส CRC ของบล็อกข้อมูลรวม

เมื่อกระบวนการถอดรหัสและกู้คืนอัตรารหัสเสร็จสิ้นซอฟต์แวร์ Module ย่อยทั้งหมดจะส่งผลลัพธ์กลับไปแสดงผลยังซอฟต์แวร์ GUI จากรูปที่ 4.160 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUSCH และ PDSCHแสดงตัวอย่างผลลัพธ์การถอดรหัสกรณีช่องสัญญาณ PUSCH โดยกำหนดให้ใช้อัลกอริทึมการถอดรหัสแบบ Belief Propagation ที่มี Iteration = 10 ผลลัพธ์จาก Module ย่อยต่าง ๆ มีรายละเอียดดังนี้

1) DeCodeBlockConcatenation ผลลัพธ์เป็นค่า LLR จำนวน 1 บล็อก เนื่องจากในกรณีนี้ไม่มีการแบ่งย่อย

2) DeBitInterleaving ผลลัพธ์เป็นค่า LLR ที่มีการแทรกสลับตำแหน่ง

3) DeRateMatching ผลลัพธ์เป็นค่า LLR ที่มีการกู้คืนอัตรารหัส

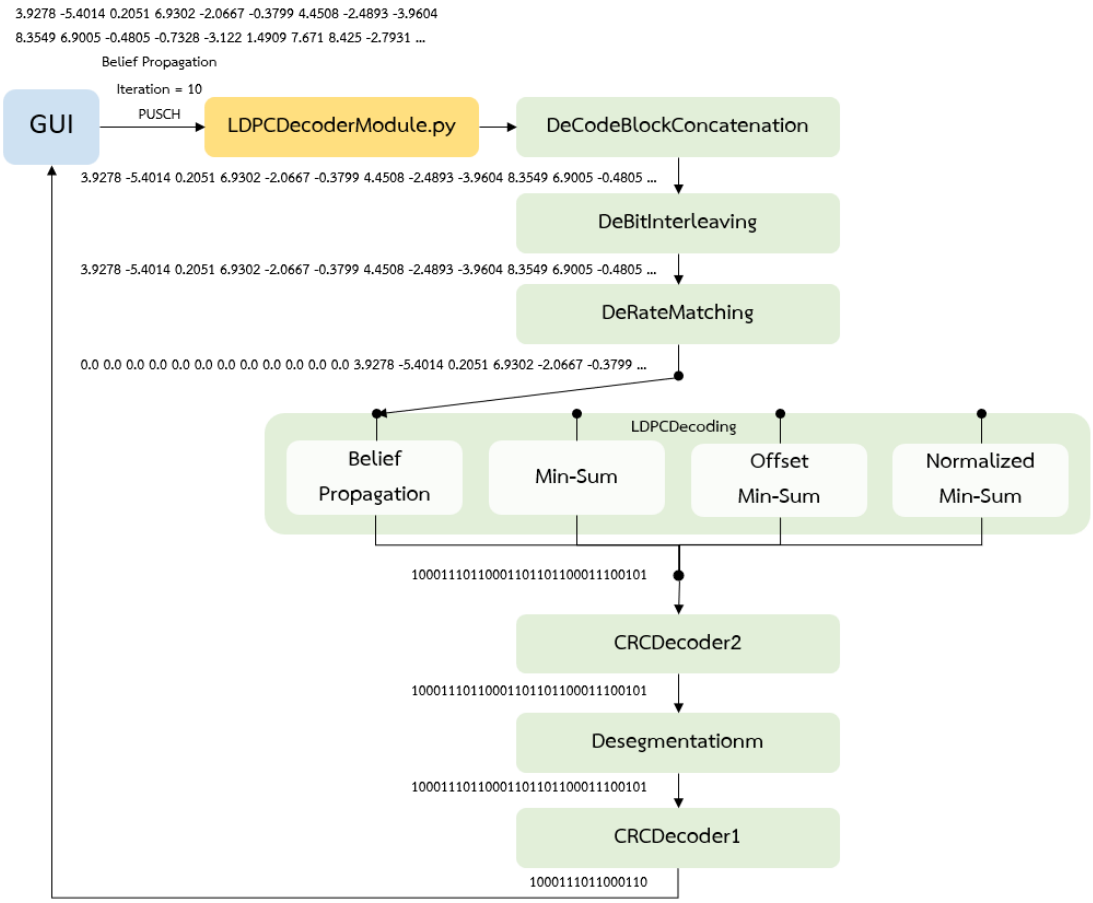
4) LDPCDecoding ผลลัพธ์เป็นบิตแบบไบนารีที่ได้จากการถอดรหัส LDPC

5) CRCDecoder2 ผลลัพธ์เป็นบิตที่ได้จากการถอดรหัส CRC ในแต่ละบล็อกย่อย อย่างไรก็ตามเนื่องจากในที่นี้ไม่มีการแบ่งบล็อกย่อยจึงไม่มีการถอดรหัส CRC ในขั้นตอนนี้

5) DeSegmentation ผลลัพธ์เป็นการนำบิตที่ได้จากการถอดรหัส CRC ในแต่ละบล็อกย่อยมาต่อกัน

6) CRCDecoder1 ผลลัพธ์เป็นบิตข้อมูลที่ได้จากการถอดรหัส CRC ของบล็อกข้อมูลรวม

หมายเหตุ: เนื่องจากผลลัพธ์ของบาง Module ย่อยมีความยาวมาก ในที่นี้จึงแสดงผลลัพธ์เพียงบางส่วน



รูปที่ 4.160 แผนภาพการเรียกใช้งานซอฟต์แวร์ Module ถอดรหัสสำหรับชุดการเรียนรู้กระบวนการฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G กรณีช่องสัญญาณกายภาพ PUSCH และ PDSCH

4.3 ผลการทดสอบการทำงานของอุปกรณ์ FPGA

4.3.1 รายชื่อซอฟต์แวร์สำหรับการสร้างวงจรบนชิปเอฟพีจีเอ

โครงการนี้ได้เขียนซอฟต์แวร์สำหรับการสร้างวงจรบนชิปเอฟพีจีเอ ภายใต้ภาษา VHDL ที่ใช้งานร่วมกับซอฟต์แวร์ GUI ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G รวมทั้งสิ้น 9 ระบบย่อย โดยแสดง ซึ่งซอฟต์แวร์เหล่านี้จะถูกสังเคราะห์ เชื่อมโยง และโปรแกรมไปยังชิปเอฟพีจีเอ และชิปเอฟพีจีเอจะถูกเชื่อมเข้ากับคอมพิวเตอร์ผ่านการสื่อสารแบบ serial เพื่อทำงานร่วมกับซอฟต์แวร์ GUI ชุดการเรียนรู้

ตารางที่ 4.1 ซอฟต์แวร์ชุดคำสั่งของระบบย่อย

ซอฟต์แวร์ชุดคำสั่งหลัก	ซอฟต์แวร์ชุดคำสั่งประกอบ	**ระบบย่อย**
top_entity_sim.vhdl	simulator_package.vhdl RxSerial.vhdl rx_buffer.vhdl simulator.vhdl tx_buffer.vhdl TxSerial.vhd memory_interface.vhdl	ชุดการเรียนรู้ส่วนประมวลผลบนอุปกรณ์ FPGA
RxSerial.vhd	-	Serial receiver
rx_buffer.vhdl	-	Receiver buffer
simulator.vhdl	sim_controller.vhdl uniform_random_input_generator.vhdl channel_encoder.vhdl digital_modulation.vhdl gaussian_random_input_generator.vhdl digital_demodulation.vhdl channel_decoder.vhdl	-
sim_controller.vhdl	-	Simulator controller
uniform_random_input_generator.vhdl	combined_Tausworthe_88.vhdl	Random input generator
channel_encoder.vhdl	ldpc_encoder.vhdl polar_encoder.vhdl	channel encoder (ประกอบด้วยรหัส LDPC และ Polar)
digital_modulation.vhdl	-	Modulator
gaussian_random_input_generator.vhdl	noisegen_top.vhdl combined_Tausworthe_88.vhdl LUTsincos.vhdl LUTcosonefour.vhdl logCal_top.vhdl CofROM.vhdl xinPropare.vhdl polyCal.vhdl	Noise generator

ซอฟต์แวร์ชุดคำสั่งหลัก	ซอฟต์แวร์ชุดคำสั่งประกอบ	**ระบบย่อย**
	sqrtCal.vhdl cordic_calculation.vhdl cordic_equation.vhdl cordic_stage_0.vhdl cordic_stage_1.vhdl cordic_stage_2.vhdl cordic_stage_3.vhdl cordic_stage_4.vhdl cordic_stage_5.vhdl cordic_stage_6.vhdl cordic_stage_7.vhdl cordic_stage_8.vhdl cordic_stage_9.vhdl cordic_stage_10.vhdl cordic_stage_11.vhdl cordic_stage_12.vhdl Div_VHDL_control.vhdl Div_lut.vhdl	
digital_demodulation.vhdl	LLR_BPSK.vhdl LLR_p2BPSK.vhdl LLR_QPSK.vhdl LLR_16QAM.vhdl LUT16QAM_cal.vhdl LUT_16QAM.vhdl LLR_64QAM.vhdl LUT64QAM_cal.vhdl LUT_64QAM.vhdl LLR_256QAM.vhdl LUT256QAM_cal.vhdl LUT_256QAM.vhdl LLR_1024QAM.vhdl LUT1024QAM_cal.vhdl LUT_1024QAM.vhdl	Demodulator

ซอฟต์แวร์ชุดคำสั่งหลัก	ซอฟต์แวร์ชุดคำสั่งประกอบ	**ระบบย่อย**
channel_decoder.vhdl	belief_propagation_layered.vhdl belief_propagation_packaged.vhdl memory_controller.vhdl parity_check_matrix_memory_generator_hdl.vhdl app_memory_generator_hdl.vhdl check_to_var_memory_generator_hdl.vhdl data_arrangement.vhdl barrel_shifter_QSN_bypass.vhdl barrel_shifter_QSN.vhdl barrel_shifter_QSN_reverse.vhdl node_processing_unit.vhdl c2v_subtraction.vhdl c2v_addition.vhdl sign_comparison.vhdl minima_structure.vhdl minima_structure_64.vhdl minima_structure_32.vhdl minima_structure_16.vhdl minima_structure_8.vhdl minima_structure_4.vhdl comparator_1_multiplexer_1.vhdl comparator_1_multiplexer_2.vhdl approximate_min_sum.vhdl	channel decoder
tx_buffer.vhdl	-	Transmitter buffer
TxSerial.vhd	-	Serial transmitter

4.3.2 ระบบของวงจรบนชิปเอฟพีจีเอ

ระบบของวงจรบนชิปเอฟพีจีเอประกอบด้วย 10 ระบบย่อย แสดงดังรูปที่ 4.161 ประกอบด้วย

1) Serial receiver สำหรับการรับข้อมูลด้วยโพรโทคอล UART จากคอมพิวเตอร์

2) Receiver buffer สำหรับการพักข้อมูลที่รับจากคอมพิวเตอร์ เพื่อส่งต่อไปยังส่วนต่าง ๆ ในวงจร รวมถึงแรม

3) Simulator controller สำหรับการควบคุมโหมดการจำลองสมรรถนะและโหมดแสดงระบบ

4) Random input generator สำหรับการสร้างข้อมูลอินพุตแบบสุ่ม เพื่อใช้ร่วมกับระบบอื่น ๆ

5) LDPC encoder สำหรับการเข้ารหัส LDPC ตามมาตรฐาน 5G

6) Polar encoder สำหรับการเข้ารหัสโพลาร์ตามมาตรฐาน 5G

7) Modulator สำหรับการมอดูเลตข้อมูลตามมาตรฐาน 5G

8) Noise generator สำหรับการบวกสัญญาณรบกวนทั้งแบบ AWGN และเฟดดิงตามมาตรฐาน 5G

9) Demodulator สำหรับการดีมอดูเลตข้อมูลตามมาตรฐาน 5G

10) LDPC decoder สำหรับการถอดรหัส LDPC ตามมาตรฐาน 5G

11) Polar decoder สำหรับการถอดรหัสโพลาร์ตามมาตรฐาน 5G

12) Transmitter buffer สำหรับการพักข้อมูลจากระบบอื่น ๆ รวมถึงแรม เพื่อส่งต่อไปยังคอมพิวเตอร์

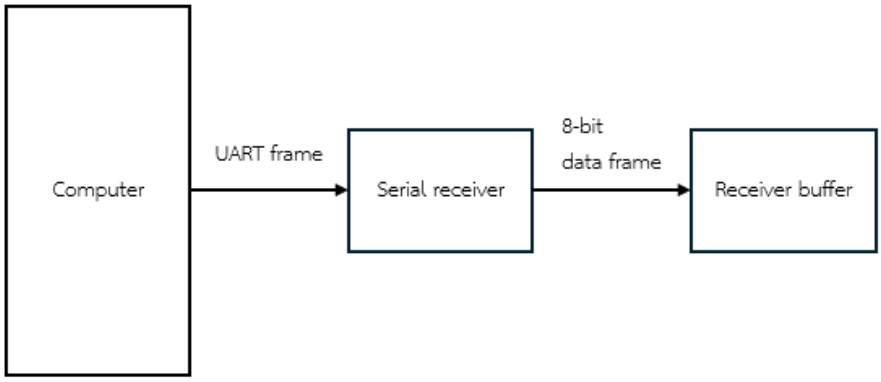
13) Serial transmitter สำหรับการส่งข้อมูลด้วยโพรโทคอล UART ไปยังคอมพิวเตอร์

ระบบย่อยจากรายการดังกล่าวจะเป็นระบบย่อยที่ถูกแบ่งตามแผนภาพระบบโดยรวม ซึ่งในทางปฏิบัติระบบย่อยเหล่านี้สามารถอยู่ภายใต้ระบบใด ๆ ได้อีก เช่น ระบบย่อย LDPC encoder และ Polar encoder จะอยู่ภายใต้ Channel encoder ระบบย่อย LDPC decoder และ Polar decoder จะอยู่ภายใต้ Channel decoder และระบบย่อย Simulator controller Random input generator Channel encoder Modulator Noise generator Demodulator และ Channel decoder จะอยู่ภายใต้ Simulator อีกต่อหนึ่ง

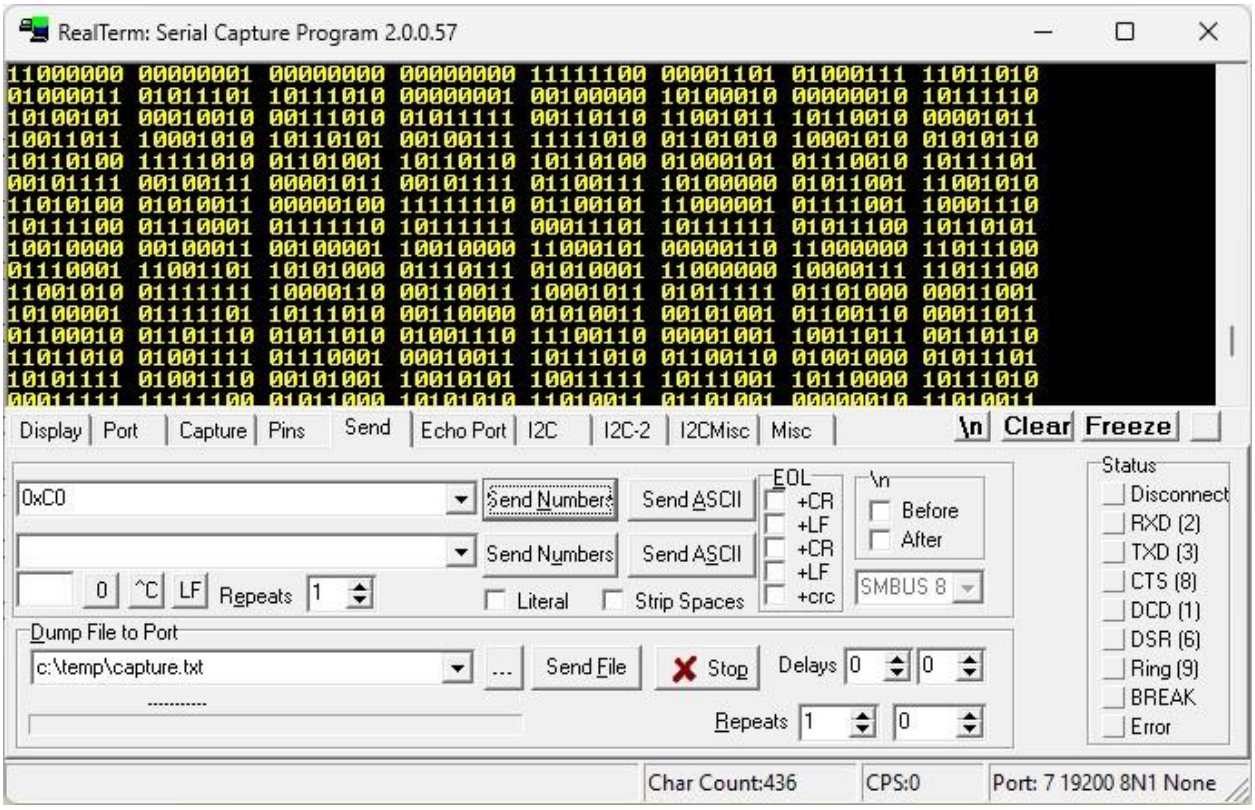
รูปที่ 4.161 ระบบของวงจรบนชิปเอฟพีจีเอ

4.3.3 ผลการทดสอบระบบย่อย Serial receiver

ระบบย่อย Serial receiver ทำหน้าที่รับข้อมูลด้วยโพรโทคอล UART จากคอมพิวเตอร์ โดยคอมพิวเตอร์จะส่งข้อมูลทีละ 8 บิตต่อเวลาอย่างอนุกรมมายังชิปเอฟพีจีเอ แสดงรายละเอียดระบบย่อย Serial receiver ดังรูปที่ 4.162 โดยผลการทดสอบระบบย่อย Serial receiver แสดงดังรูปที่ 4.163



รูปที่ 4.162 รายละเอียดระบบย่อย Serial receiver



รูปที่ 4.163 ผลการทดสอบ timing diagram ของระบบย่อย Serial receiver

จากรูปที่ 4.163 แสดงผลการทดสอบการรับข้อมูลจากคอมพิวเตอร์มายังระบบย่อย Serial receiver และส่งกลับมาแสดงผลบนโปรแกรม terminal บนคอมพิวเตอร์ เพื่อตรวจสอบว่าค่าที่รับเข้าไปยังระบบย่อยมีความถูกต้อง ผลการทดสอบดังกล่าวอาจแสดงผลการทดสอบร่วมกันของระบบย่อย Serial receiver และ Serial transmitter

4.3.4 ผลการทดสอบระบบย่อย Receiver buffer

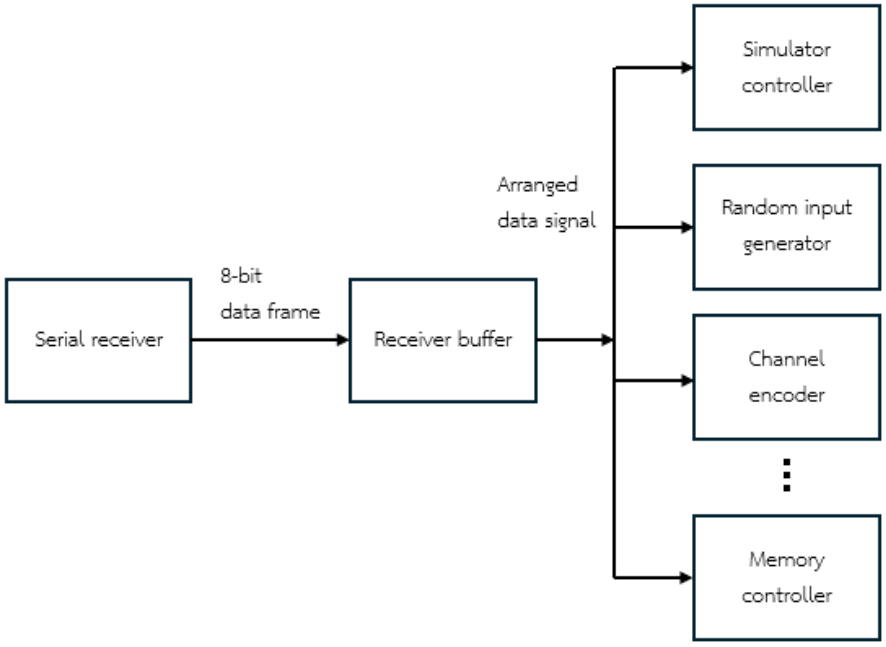
ระบบย่อย Receiver buffer ทำหน้าที่พักข้อมูลที่รับจากคอมพิวเตอร์ เพื่อส่งต่อไปยังส่วนต่าง ๆ ในวงจร รวมถึงแรม โดยข้อมูลชุดละ 8 บิตจากระบบย่อย Serial receiver จะถูกนำมาจัดเรียงเป็นสัญญาณที่มีความยาวขนาดต่าง ๆ ตามที่กำหนดไว้ โดยสามารถแยกประเภทสัญญาณได้จากชุดข้อมูลส่วนหัว (header) ที่มีความยาว 8 บิตทั้งหมด ประเภทและความยาวของสัญญาณจำแนกโดยระบบย่อย Receiver buffer สามารถสรุปได้ดังตารางที่ 4.2 โดยจะแสดงรายละเอียดระบบย่อย Receiver buffer ดังรูปที่ 4.164 และแสดงผลการทดสอบระบบย่อย Receiver buffer ดังรูปที่ 4.165

ตารางที่ 4.2 ประเภทและความยาวของสัญญาณจำแนกโดยระบบย่อย Receiver buffer

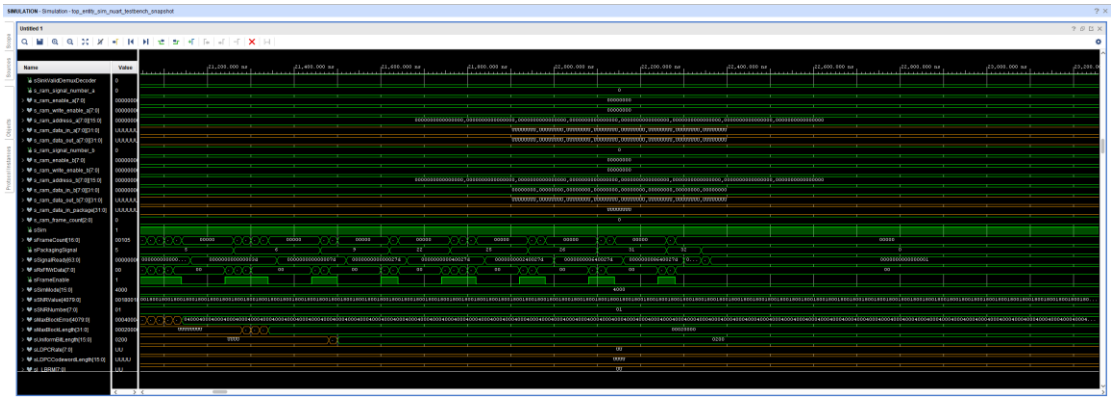
ประเภทสัญญาณ	ส่วนหัว	ความยาวข้อมูล
ส่วนหัวทั่วไป		
Modulation flag	00010001	-
Channel coding flag	00010010	-
Simulation flag	00011000	16
ส่วนหัวสำหรับระบบย่อย Simulator controller ภายใต้ Simulator		
snrValue	00011010	4080
snrNumber	00011011	8
maxBlockError	00011100	4080
maxBlockLength	00011101	32
ส่วนหัวสำหรับระบบย่อย Random input generator ภายใต้ Simulator		
uniformBitLength	00100001	16
ส่วนหัวสำหรับระบบย่อย LDPC encoder ภายใต้ Channel encoder และ Simulator		
Rate	00110001	8
liftingSize	00110010	16
I_LBRM	00110011	8
N_L	00110100	8
rv_id	00110101	8
infoLength	00110110	16

ประเภทสัญญาณ	ส่วนหัว	ความยาวข้อมูล
infoBits	00110111	8448
channelType	01000101	8
modScheme	01010001	8
ส่วนหัวสำหรับระบบย่อย Polar encoder ภายใต้ Channel encoder และ Simulator		
infoBits	01000001	1712
infoLength	01000010	16
codewordLength	01000011	16
rntiBits	01000100	16
channelType	01000101	8
ส่วนหัวสำหรับระบบย่อย Modulator ภายใต้ Simulator		
modScheme	01010001	8
modBitLength	01010010	16
modBitInput	01010011	1024
ส่วนหัวสำหรับระบบย่อย Noise generator ภายใต้ Simulator		
channelAndSNR	00100011	24
sqrtVar	00100100	16
gaussianSymbolLength	00100101	16
gaussianValueInput	00100110	32768
complexChannel Coefficient	00100111	16384
inverseComplex ChannelCoefficient	00101000	16384
ส่วนหัวสำหรับระบบย่อย Demodulator ภายใต้ Simulator		
modScheme	01010001	8
channelAndSNR	00100011	24
inverseSqrtVar	01010101	16
demodEquation	01010110	8
demodSymbolLength	01010111	16
demodValueInput	01011000	32768
ส่วนหัวสำหรับระบบย่อย LDPC decoder ภายใต้ Channel decoder และ Simulator		
channelType	01000101	8
decoderConfig	00111000	8
iteration	00111001	8
offsetAndScale	00111010	32
processorNumber	00111011	8
liftingSize	00111101	16
ldpcLLRSymbolLength	00111110	16
ldpcLLRInput	00111111	8192

ประเภทสัญญาณ	ส่วนหัว	ความยาวข้อมูล
ส่วนหัวสำหรับระบบย่อย Polar decoder ภายใต้ Channel decoder และ Simulator		
rntiBits	01000100	16
channelType	01000101	8
offsetAndScale	00111010	32
decoderConfig	01000111	8
crcPolynomial	01001000	24
frozenPosition	01001010	2048
llrSymbolLength	01001011	16
llrInput	01001100	8192



รูปที่ 4.164 รายละเอียดระบบย่อย Receiver buffer

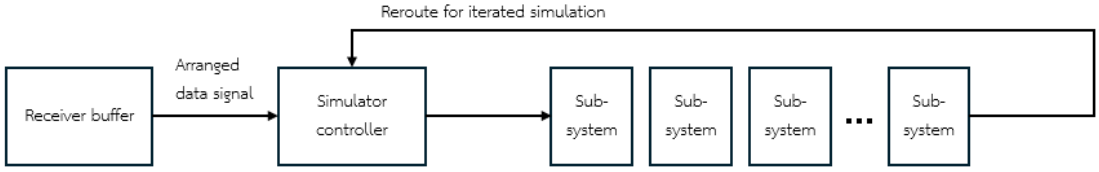


รูปที่ 4.165 ผลการทดสอบ timing diagram ของระบบย่อย Receiver buffer

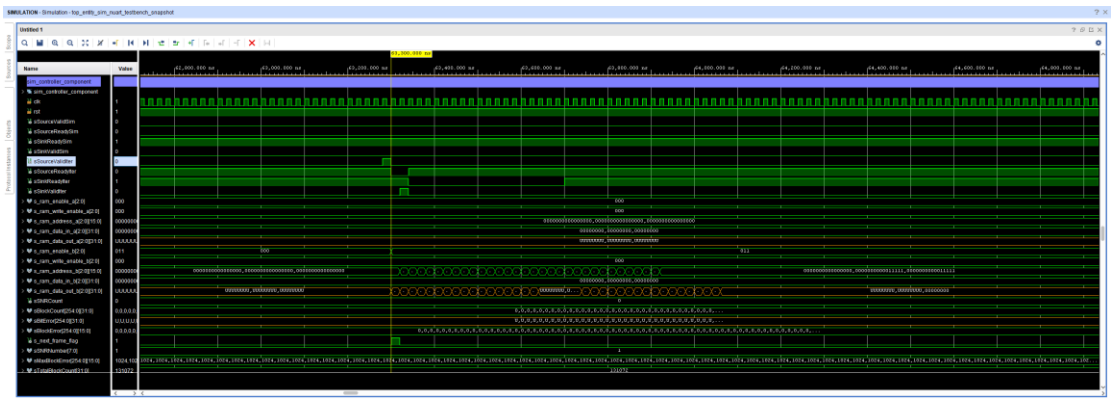
จากรูปที่ 4.165 แสดงผลการทดสอบการพักและจัดเรียงข้อมูลจากระบบย่อย Serial receiver เพื่อส่งต่อไปยังระบบย่อยอื่น ๆ และแรม โดยข้อมูลหนึ่งประเภทจะถูกเริ่มตรวจจับจากข้อมูลส่วนหัวชุดแรก จากนั้นตามด้วยชุดข้อมูลประเภทนั้นตามจำนวนบิตหรือเฟรมที่กำหนดไว้ตามตารางก่อนหน้า เมื่อข้อมูลถูกจัดเรียงเป็นสัญญาณตามความยาวที่กำหนดเสร็จสิ้น สัญญาณเหล่านี้จะถูกส่งต่อไปยังระบบย่อยถัดไป หากสัญญาณนั้นมีความยาวมากเป็นพิเศษจะถูกจัดเก็บในบล็อกแรมแทน

4.3.5 ผลการทดสอบระบบย่อย Simulator controller

ระบบย่อย Simulator controller สำหรับการควบคุมโหมดการจำลองสมรรถนะและโหมดแสดงระบบ โดยระบบย่อยนี้จะรับคำสั่งจากคอมพิวเตอร์ สำหรับโหมดการจำลองสมรรถนะ สัญญาณของระบบย่อยต่าง ๆ จะถูกโยงให้มีการทำงานเป็นวงรอบ เพื่อทำการประมาณสมรรถนะของระบบตามวิธี Monte Carlo โดยการจำลองการทำงานของระบบซ้ำหลายรอบ โหมดการจำลองสมรรถนะจะถูกใช้งานร่วมกับชุดทดสอบสมรรถนะในโปรแกรมชุดการเรียนรู้บนคอมพิวเตอร์ สำหรับโหมดแสดงระบบ สัญญาณของระบบย่อยภายใต้ simulator จะถูกเชื่อมระหว่างระบบย่อย Receiver buffer และ Transmitter buffer แทนที่การเรียงต่อกับระบบย่อยอื่น ๆ เนื่องจากจำเป็นต้องรับส่งสัญญาณภายในระบบย่อยระหว่างโปรแกรมชุดการเรียนรู้บนคอมพิวเตอร์ โดยจะแสดงรายละเอียดระบบย่อย Simulator controller ดังรูปที่ 4.166 และแสดงผลการทดสอบระบบย่อย Simulator controller ดังรูปที่ 4.167



รูปที่ 4.166 รายละเอียดระบบย่อย Simulator controller

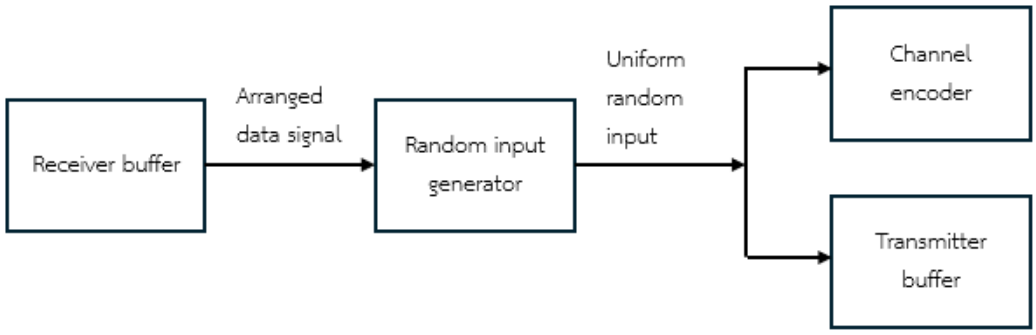


รูปที่ 4.167 ผลการทดสอบ timing diagram ของระบบย่อย Simulator controller

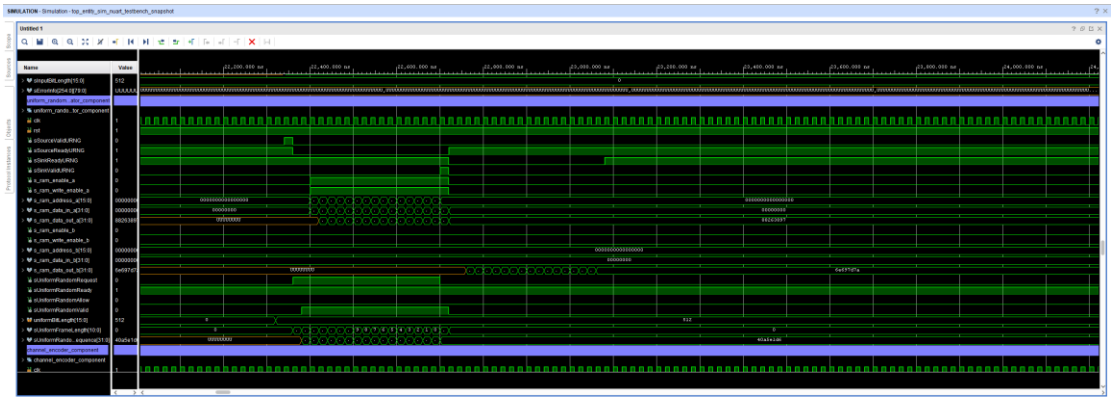
จากรูปที่ 4.167 แสดงผลการทดสอบการควบคุมโหมดการจำลองสมรรถนะและโหมดแสดงระบบ โดยหากเลือกโหมดการจำลองสมรรถนะ สัญญาณภายในระบบย่อย Simulator controller จะมีการใช้งานสำหรับการจำลองสมรรถนะ เช่น การเก็บจำนวนรอบ นับจำนวนความผิดพลาด เป็นต้น หากเลือกโหมดแสดงระบบ ระบบย่อยภายใต้ simulator จะถูกเชื่อมระหว่างระบบย่อย Receiver buffer และ Transmitter buffer ซึ่งจะทำให้สัญญาณภายในระบบย่อย Simulator controller ส่วนใหญ่จะไม่ถูกใช้งาน

4.3.6 ผลการทดสอบระบบย่อย Random input generator

ระบบย่อย Random input generator สำหรับการสร้างข้อมูลอินพุตแบบสุ่ม เพื่อใช้ร่วมกับระบบอื่น ๆ โดยระบบย่อยนี้จะรับความยาวบิตอินพุตแบบสุ่ม เพื่อที่จะสร้างสัญญาณสุ่มที่มีการกระจายแบบเอกรูป โดยระบบย่อยจะเรียกใช้งานวงจร combine Tausworthe generator ที่สามารถสร้างตัวเลขที่มีการกระจายแบบเอกรูปได้ โดยจะนับจำนวนชุดตัวเลขที่ถูกสร้างจากวงจรดังกล่าวตามค่าความยาวบิตอินพุตแบบสุ่ม ชุดข้อมูลที่ถูกสร้างจะถูกเก็บไว้ในแรม โดยจะแสดงรายละเอียดระบบย่อย Random input generator ดังรูปที่ 4.168 และแสดงผลการทดสอบระบบย่อย Random input generator ดังรูปที่ 4.169



รูปที่ 4.168 รายละเอียดระบบย่อย Random input generator

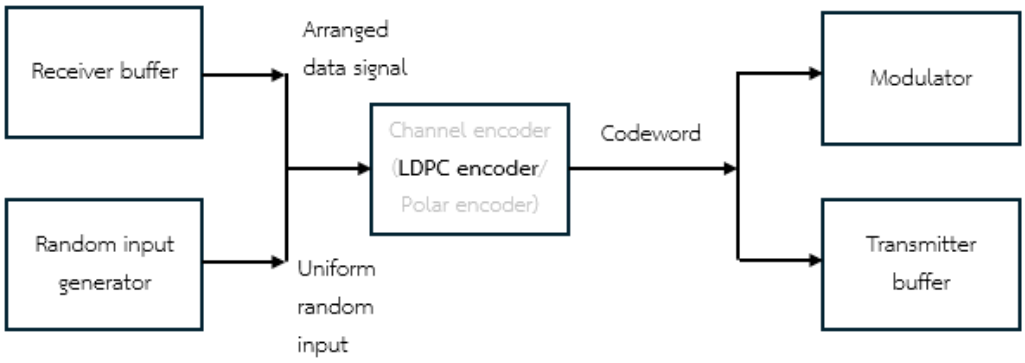


รูปที่ 4.169 ผลการทดสอบ timing diagram ของระบบย่อย Random input generator

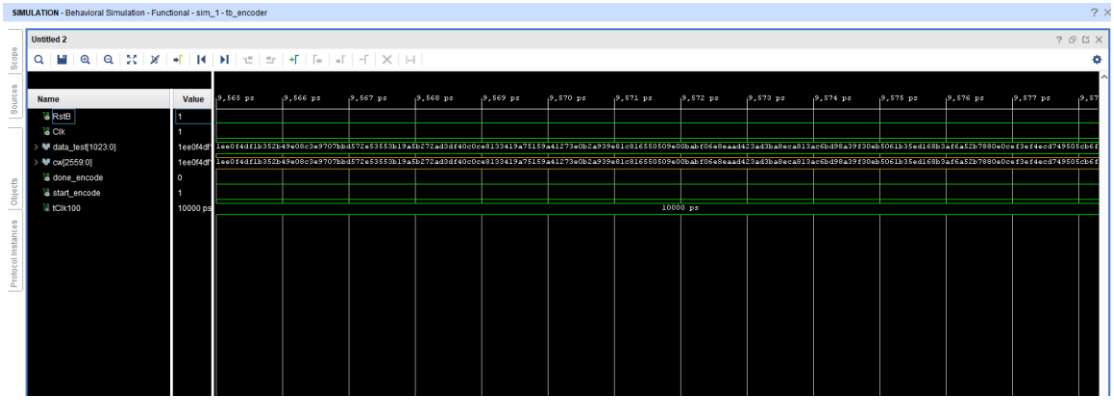
จากรูปที่ 4.169 แสดงผลการทดสอบการสร้างข้อมูลอินพุตแบบสุ่ม โดยระบบย่อยจะรับความยาวบิตอินพุตแบบสุ่มไว้ในสัญญาณ sUniformFrameLength จากนั้นสัญญาณนี้จะถูกลดค่าเพื่อนับจำนวนชุดข้อมูลที่จะถูกสร้างขึ้น ชุดข้อมูลที่ถูกสร้างจะเป็นสัญญาณ sUniformFrameLength และถูกเก็บเข้าแรมทันที

4.3.7 ผลการทดสอบระบบย่อย LDPC encoder

ระบบย่อย LDPC encoder สำหรับการเข้ารหัส LDPC ตามมาตรฐาน 5G ระบบย่อยนี้จะทำการเข้ารหัส LDPC ชุดข้อมูลตามมาตรฐาน 5G โดยระบบย่อยจะรับบิตอินพุตจากแรมและพารามิเตอร์อื่น ๆ จากระบบย่อย Receiver buffer เพื่อทำการเข้ารหัส LDPC ผลลัพธ์ที่ได้ เรียก คำรหัส จะถูกจัดเก็บในแรม โดยจะแสดงรายละเอียดระบบย่อย LDPC encoder ดังรูปที่ 4.170 และแสดงผลการทดสอบระบบย่อย LDPC encoder ดังรูปที่ 4.171



รูปที่ 4.170 รายละเอียดระบบย่อย LDPC encoder

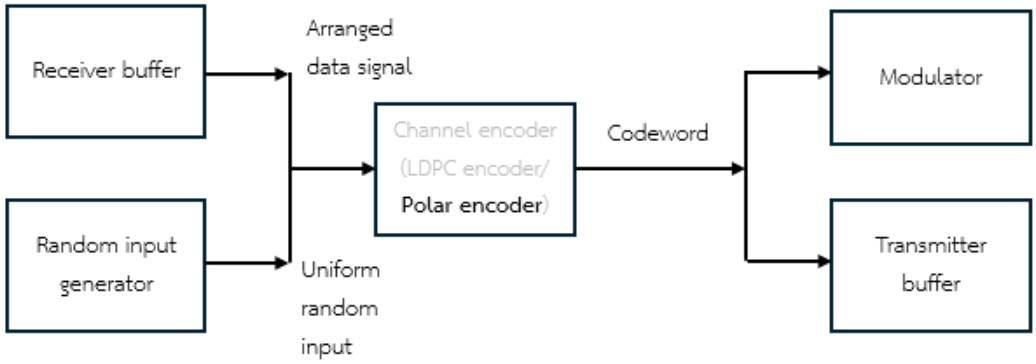


รูปที่ 4.171 ผลการทดสอบ timing diagram ของระบบย่อย LDPC encoder

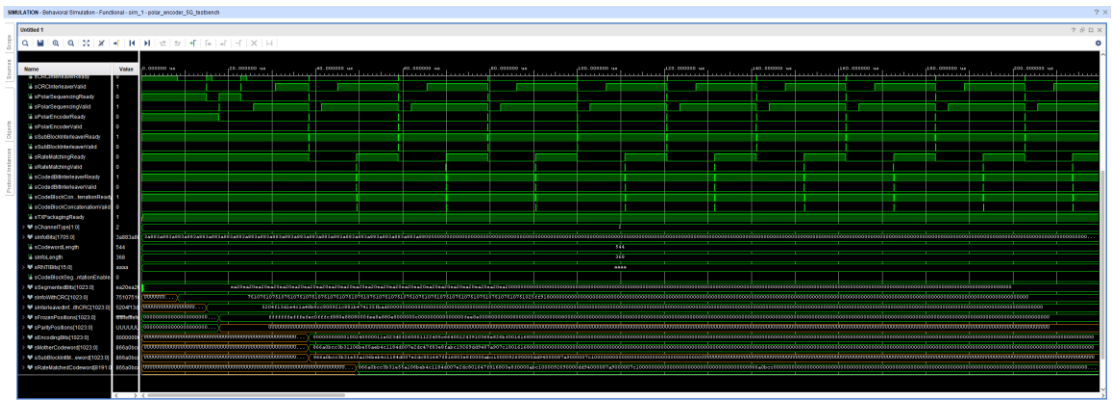
จากรูปที่ 4.171 แสดงผลการทดสอบการเข้ารหัส LDPC ตามมาตรฐาน 5G โดยระบบย่อยจะทำการเข้ารหัสจากบิตข้อมูลที่มีสัญญาณชื่อ rto_top_circulant ส่งไปยังวงจรหมุนภายในระบบย่อย data_cyclic และวงจรคำนวณบิตพาริตีภายในระบบย่อย parity ในลำดับถัดไปตามโครงสร้างเมทริกซ์ตรวจสอบพาริตีมาตรฐาน 5G ผลลัพธ์การคำนวณบิตพาริตีจะถูกส่งออกมาจากวงจรคำนวณบิตพาริตีในระบบย่อย parity เพื่อถูกนำไปเก็บไว้ในแรม

4.3.8 ผลการทดสอบระบบย่อย Polar encoder

ระบบย่อย Polar encoder สำหรับการเข้ารหัสโพลาร์ตามมาตรฐาน 5G ระบบย่อยนี้จะทำการเข้ารหัสโพลาร์ชุดข้อมูลตามมาตรฐาน 5G โดยระบบย่อยจะรับบิตอินพุตจากแรมและพารามิเตอร์อื่น ๆ จากระบบย่อย Receiver buffer เพื่อทำการเข้ารหัสโพลาร์ ผลลัพธ์ที่ได้ เรียก คำรหัส จะถูกจัดเก็บในแรม โดยจะแสดงรายละเอียดระบบย่อย Polar encoder ดังรูปที่ 4.172 และแสดงผลการทดสอบระบบย่อย Polar encoder ดังรูปที่ 4.172



รูปที่ 4.172 รายละเอียดระบบย่อย Polar encoder

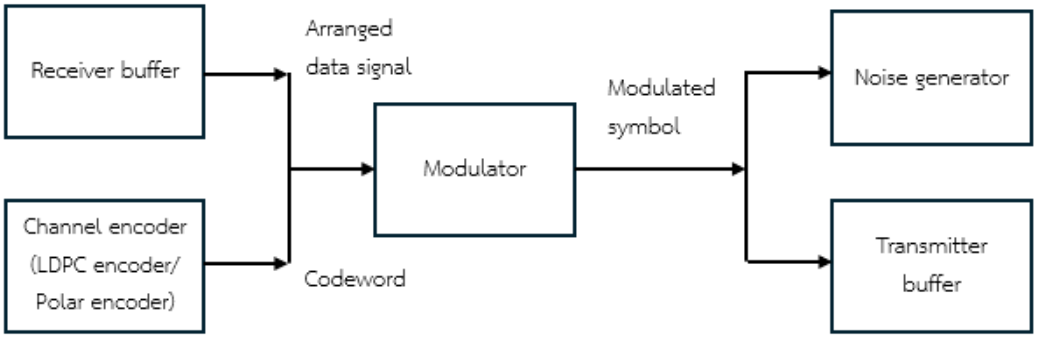


รูปที่ 4.173 ผลการทดสอบ timing diagram ของระบบย่อย Polar encoder

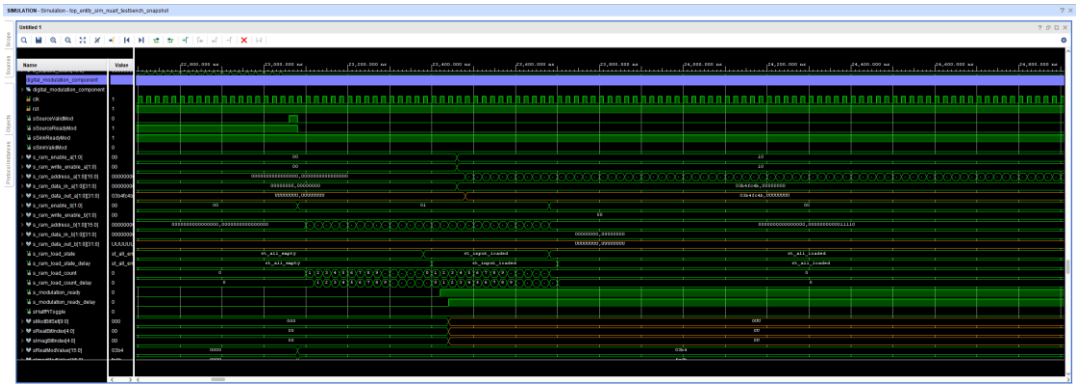
จากรูปที่ 4.173 แสดงผลการทดสอบการเข้ารหัสโพลาร์ตามมาตรฐาน 5G โดยระบบย่อยจะทำการหาตำแหน่งบิตข้อมูลและบิตแช่แข็งตามมาตรฐาน 5G และเก็บข้อมูลตำแหน่งไว้ในสัญญาณชื่อ sFrozenPosition ภายในระบบย่อย polar_sequencing จากนั้นระบบจะทำการเข้ารหัสจากบิตข้อมูลที่มีสัญญาณชื่อ sInfoWithCRC ที่ถูกเรียงตำแหน่งบิตข้อมูลและบิตแช่แข็งแล้วภายในระบบย่อย polar_encoder ผลลัพธ์ที่ได้ เรียก คำรหัส จะถูกจัดเก็บในแรม

4.3.9 ผลการทดสอบระบบย่อย Modulator

ระบบย่อย Modulator สำหรับการมอดูเลตข้อมูลตามมาตรฐาน 5G โดยระบบย่อยจะรับบิตอินพุตจากแรมและพารามิเตอร์อื่น ๆ จากระบบย่อย Receiver buffer เพื่อทำการมอดูเลต การมอดูเลตภายใต้มาตรฐาน 5G มีเลือกกว่า 7 รูปแบบ ขึ้นอยู่กับเงื่อนไขชุดข้อมูล ช่องสัญญาณการสื่อสาร และตัวแปรอื่น ผลลัพธ์การมอดูเลตจะถูกจัดเก็บเข้าแรม โดยจะแสดงรายละเอียดระบบย่อย Modulator ดังรูปที่ 4.174 และแสดงผลการทดสอบระบบย่อย Modulator ดังรูปที่ 4.175



รูปที่ 4.174 รายละเอียดระบบย่อย Modulator

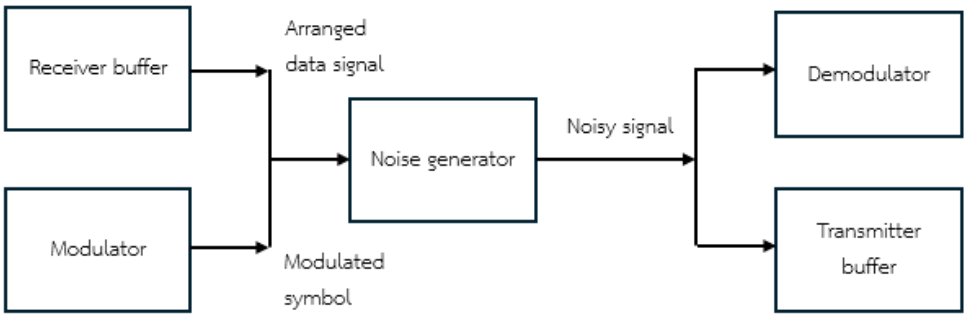


รูปที่ 4.175 ผลการทดสอบ timing diagram ของระบบย่อย Modulator

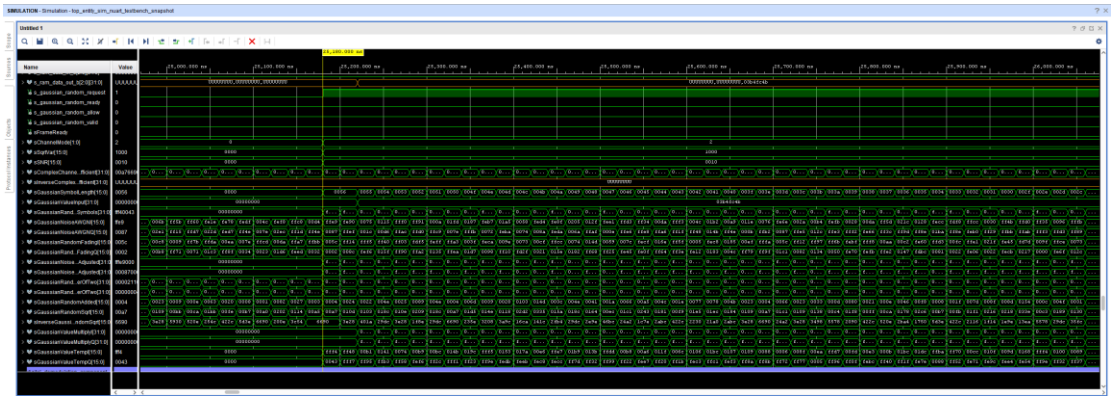
จากรูปที่ 4.175 แสดงผลการทดสอบการมอดูเลตข้อมูลตามมาตรฐาน 5G โดยระบบย่อยจะนำข้อมูลจากแรมแบ่งย่อยเป็นกลุ่มบิตที่มีความยาวตามประเภทการมอดูเลตที่ใช้จากสัญญาณ sModScheme จากนั้นจะใช้ตารางค้นหาในการโยงค่าชุดบิตจากสัญญาณ sModBitSet ไปเป็นค่าเชิงซ้อนในสัญญาณ sRealModValue และ sImagModValue เป็นผลลัพธ์การมอดูเลต ผลลัพธ์เหล่านี้จะถูกเก็บในแรม

4.3.10 ผลการทดสอบระบบย่อย Noise generator

ระบบย่อย Noise generator สำหรับการบวกสัญญาณรบกวนทั้งแบบ AWGN และเฟดดิงตามมาตรฐาน 5G โดยระบบย่อยจะทำการสร้างสัญญาณรบกวนขึ้นจากวงจร Box-Muller เพื่อสร้างค่าเลขสุ่มที่มีการกระจายแบบเกาส์เซียน โดยเลขสุ่มนี้จะมีค่าเฉลี่ยที่ 0 และความแปรปรวนเท่ากับ 1 ค่าเลขสุ่มดังกล่าวจะสามารถนำไปสร้างเป็นสัญญาณรบกวนเกาส์เซียนขาวบวกได้ รวมถึงค่าสัมประสิทธิ์ช่องสัญญาณของช่องสัญญาณเฟดดิงเรย์ลีย์ ค่าสัมประสิทธิ์ช่องสัญญาณจะถูกคูณเข้าไปกับสัญญาณของชุดข้อมูล รวมถึงสัญญาณรบกวนที่ถูกสร้างจะถูกบวกเข้าไปกับสัญญาณของชุดข้อมูล โดยจะแสดงรายละเอียดระบบย่อย Noise generator ดังรูปที่ 4.176 และแสดงผลการทดสอบระบบย่อย Noise generator ดังรูปที่ 4.177



รูปที่ 4.176 รายละเอียดระบบย่อย Noise generator

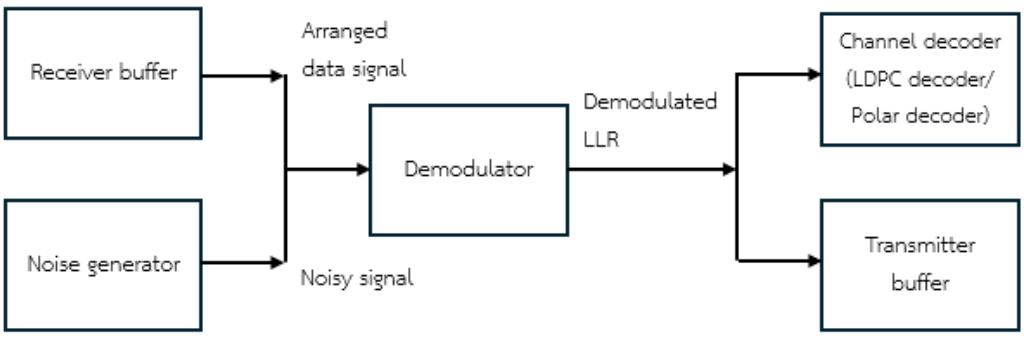


รูปที่ 4.177 ผลการทดสอบ timing diagram ของระบบย่อย Noise generator

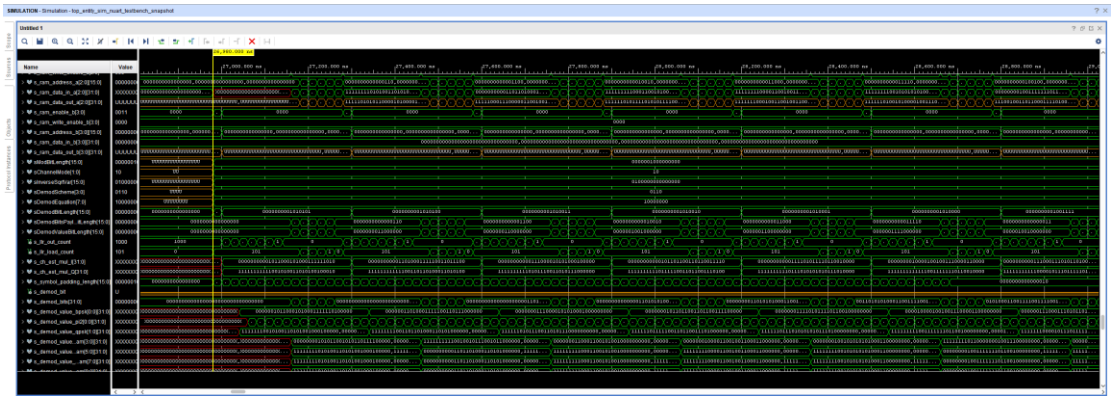
จากรูปที่ 4.177 แสดงผลการทดสอบการบวกสัญญาณรบกวนทั้งแบบ AWGN และเฟดดิง สัญญาณรบกวนที่ถูกสร้างขึ้นจะถูกสัญญาณไว้ภายในสัญญาณ sGaussianNoiseIVarAdjusted และ sGaussianNoiseQVar Adjusted ค่าสัมประสิทธิ์ช่องสัญญาณจะถูกสร้างขึ้นและถูกเก็บไว้ภายในสัญญาณ sGaussianRandomSqrt สัญญาณดังกล่าวจะถูกบวกและคูณไปยังสัญญาณอินพุตได้ผลลัพธ์เป็นสัญญาณที่เกิดการรบกวน sGaussian RandomInputBits สัญญาณดังกล่าวจะถูกจัดเก็บไว้ในแรม

4.3.11 ผลการทดสอบระบบย่อย Demodulator

ระบบย่อย Demodulator สำหรับการดีมอดูเลตข้อมูลตามมาตรฐาน 5G โดยระบบย่อยจะรับบิตอินพุตจากแรมและพารามิเตอร์อื่น ๆ จากระบบย่อย Receiver buffer เพื่อทำการดีมอดูเลต การดีมอดูเลตภายใต้มาตรฐาน 5G มีเลือกกว่า 7 รูปแบบ สอดคล้องกับการมอดูเลต ผลลัพธ์การดีมอดูเลตจะถูกจัดเก็บเข้าแรม โดยจะแสดงรายละเอียดระบบย่อย Demodulator ดังรูปที่ 4.178 และแสดงผลการทดสอบระบบย่อย Demodulator ดังรูปที่ 4.149



รูปที่ 4.178 รายละเอียดระบบย่อย Demodulator

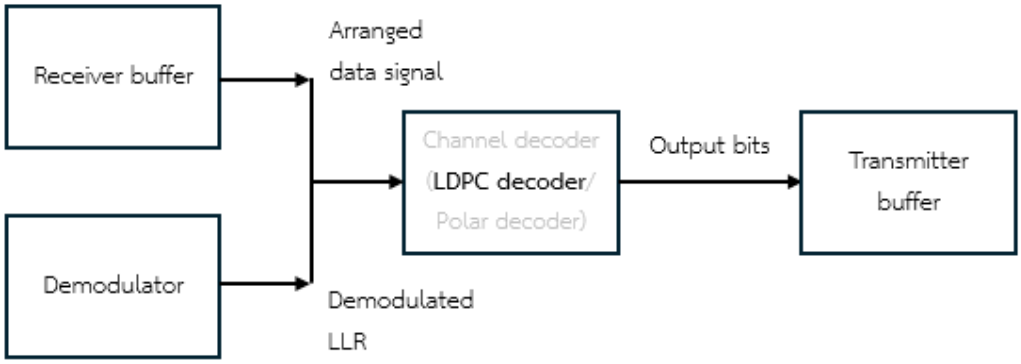


รูปที่ 4.179 ผลการทดสอบ timing diagram ของระบบย่อย Demodulator

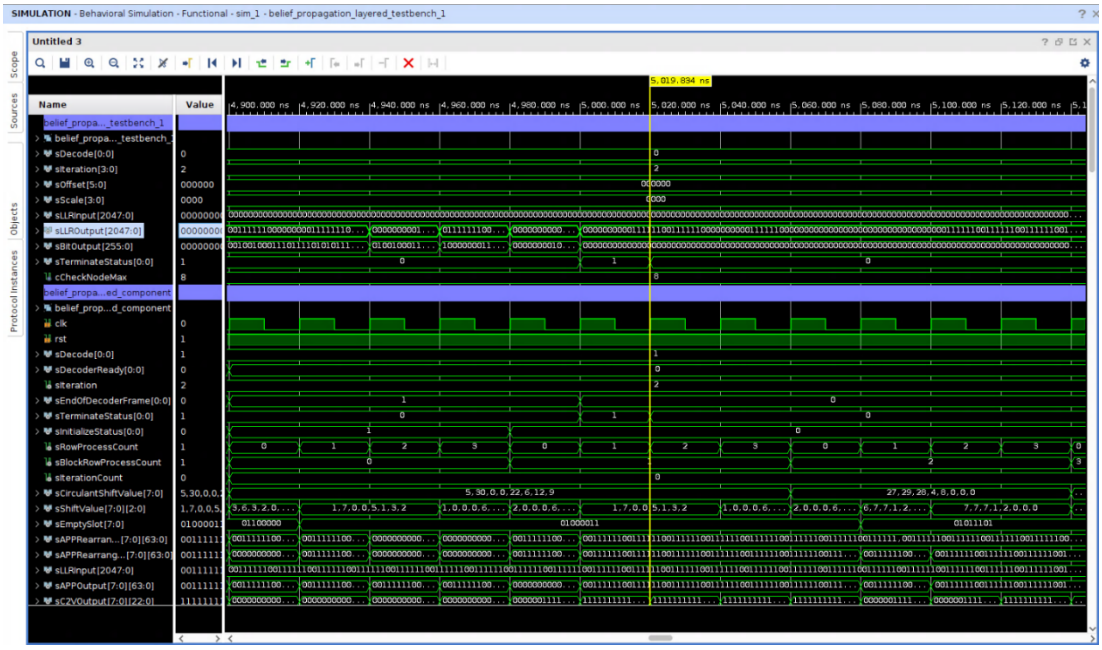
จากรูปที่ 4.179 แสดงผลการทดสอบการดีมอดูเลตข้อมูลตามมาตรฐาน 5G โดยระบบย่อยจะนำข้อมูลจากแรมเป็นกลุ่มสัญลักษณ์ จากนั้นจะใช้ตารางค้นหาในการโยงค่าเชิงซ้อนในสัญญาณ rDemodValueInput ไปเป็นค่า LLR ภายใต้สัญญาณ rDemodValue เป็นผลลัพธ์การดีมอดูเลต ผลลัพธ์เหล่านี้จะถูกเก็บในแรม

4.3.12 ผลการทดสอบระบบย่อย LDPC decoder และ Polar decoder

ระบบย่อย LDPC decoder สำหรับการถอดรหัส LDPC ตามมาตรฐาน 5G ระบบย่อยนี้จะทำการถอดรหัส LDPC ชุดข้อมูลตามมาตรฐาน 5G โดยระบบย่อยจะรับบิตอินพุตจากแรมและพารามิเตอร์อื่น ๆ จากระบบย่อย Receiver buffer เพื่อทำการถอดรหัส LDPC ผลลัพธ์ที่ได้ บิตข้อมูลเดิม จะถูกจัดเก็บในแรม โดยจะแสดงรายละเอียดระบบย่อย LDPC decoder ดังรูปที่ 4.180 และแสดงผลการทดสอบระบบย่อย LDPC decoder ดังรูปที่ 4.181



รูปที่ 4.180 รายละเอียดระบบย่อย LDPC decoder

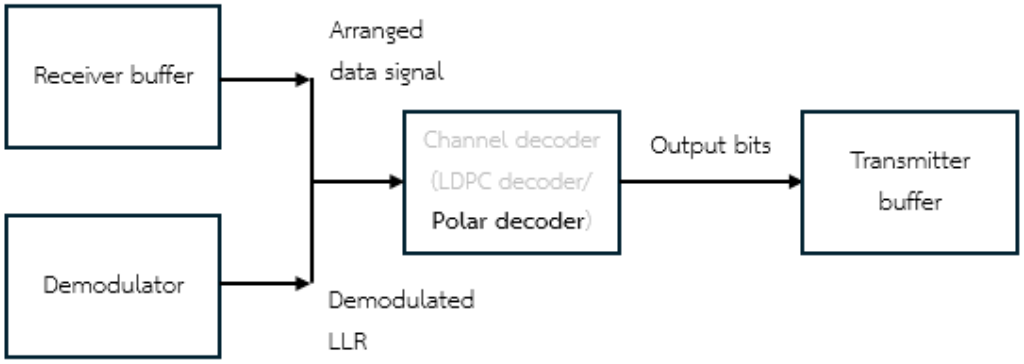


รูปที่ 4.181 ผลการทดสอบ timing diagram ของระบบย่อย LDPC decoder

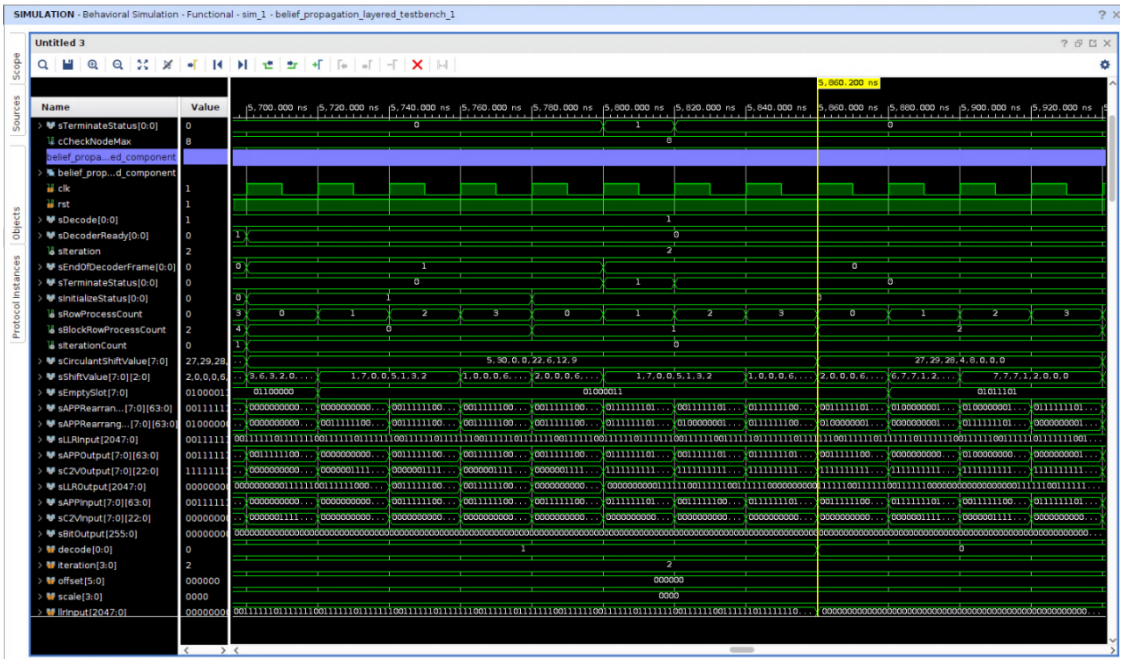
จากรูปที่ 4.181 แสดงผลการทดสอบการถอดรหัส LDPC ตามมาตรฐาน 5G โดยระบบย่อยจะทำการถอดรหัสจากค่า LLR ที่มีสัญญาณชื่อ sLLRInput ส่งไปยังวงจรโหนดตรวจสอบและวงจรโหนดตัวแปรตามโครงสร้างเมทริกซ์ตรวจสอบพาริตีมาตรฐาน 5G ผลลัพธ์การถอดรหัสจะถูกส่งออกมายังสัญญาณ sLLROutput เพื่อถูกนำไปเก็บไว้ในแรม

4.3.13 ผลการทดสอบระบบย่อย Polar decoder

ระบบย่อย Polar decoder สำหรับการถอดรหัสโพลาร์ โดยระบบย่อยจะรับบิตอินพุตจากแรมและพารามิเตอร์อื่น ๆ จากระบบย่อย Receiver buffer เพื่อทำการถอดรหัสโพลาร์ ผลลัพธ์ที่ได้บิตข้อมูลเดิม จะถูกจัดเก็บในแรม โดยจะแสดงรายละเอียดระบบย่อย Polar decoder ดังรูปที่ 4.182 และแสดงผลการทดสอบระบบย่อย Polar decoder ดังรูปที่ 4.183



รูปที่ 4.182 รายละเอียดระบบย่อย Polar decoder

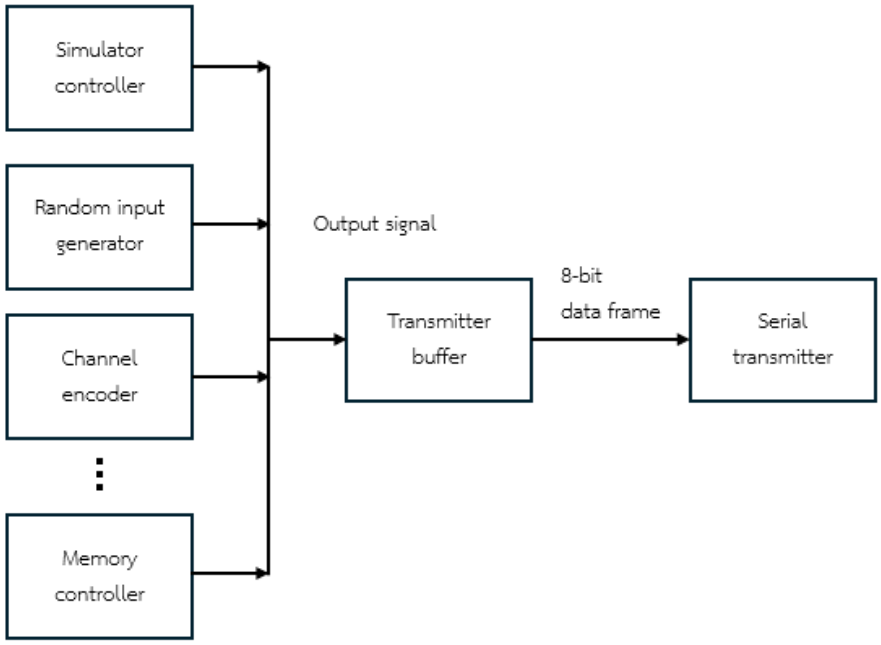


รูปที่ 4.183 ผลการทดสอบ timing diagram ของระบบย่อย Polar decoder

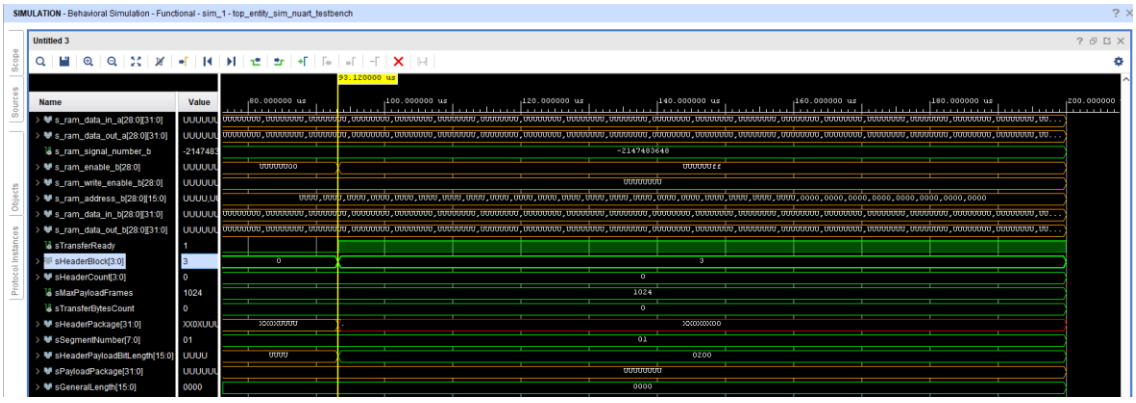
จากรูปที่ 4.183 แสดงผลการทดสอบการถอดรหัสโพลาร์ตามมาตรฐาน 5G โดยระบบย่อยจะทำการถอดรหัสจากค่า LLR ที่มีสัญญาณชื่อ sLLRInput ส่งไปยังวงจรโหนดตรวจสอบและวงจรโหนดตัวแปรตามโครงสร้างเมทริกซ์ตรวจสอบพาริตีของรหัสโพลาร์ ที่ถูกสร้างขึ้นมาโดยเฉพาะ ผลลัพธ์การถอดรหัสจะถูกส่งออกมายังสัญญาณ sLLROutput เพื่อถูกนำไปเก็บไว้ในแรม สำหรับการถอดรหัส LDPC และรหัสโพลาร์สามารถใช้งานวงจรถอดรหัสร่วมกันได้ จึงมีชื่อสัญญาณที่คล้ายคลึงกัน

4.3.14 ผลการทดสอบระบบย่อย Transmitter buffer

ระบบย่อย Transmitter buffer ทำหน้าที่พักข้อมูลที่รับจากส่วนต่าง ๆ ในวงจร รวมถึงแรม เพื่อส่งต่อไปยังคอมพิวเตอร์ โดยสัญญาณข้อมูลจากระบบย่อยต่าง ๆ จะถูกนำมาจัดเรียงเป็นชุดข้อมูล ความยาว 8 บิต โดยสามารถแยกประเภทชุดข้อมูลได้จากชุดข้อมูลส่วนหัว (header) ที่มีความยาว 8 บิตท โดยจะแสดงรายละเอียดระบบย่อย Transmitter buffer ดังรูปที่ 4.184 และแสดงผลการทดสอบระบบย่อย Transmitter buffer ดังรูปที่ 4.185



รูปที่ 4.184 รายละเอียดระบบย่อย Transmitter buffer

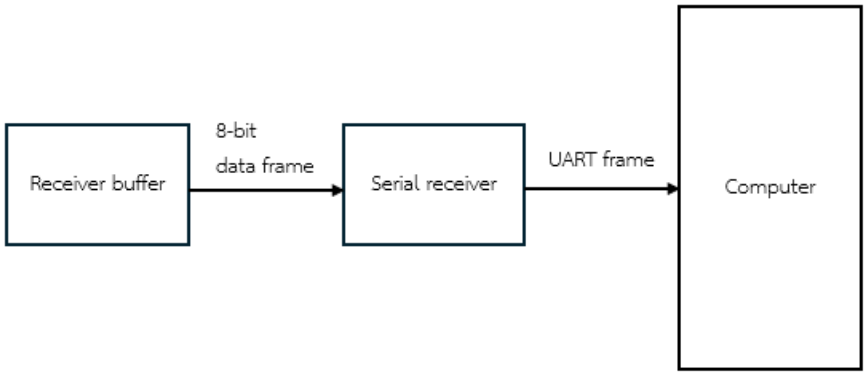


รูปที่ 4.185 ผลการทดสอบ timing diagram ของระบบย่อย Transmitter buffer

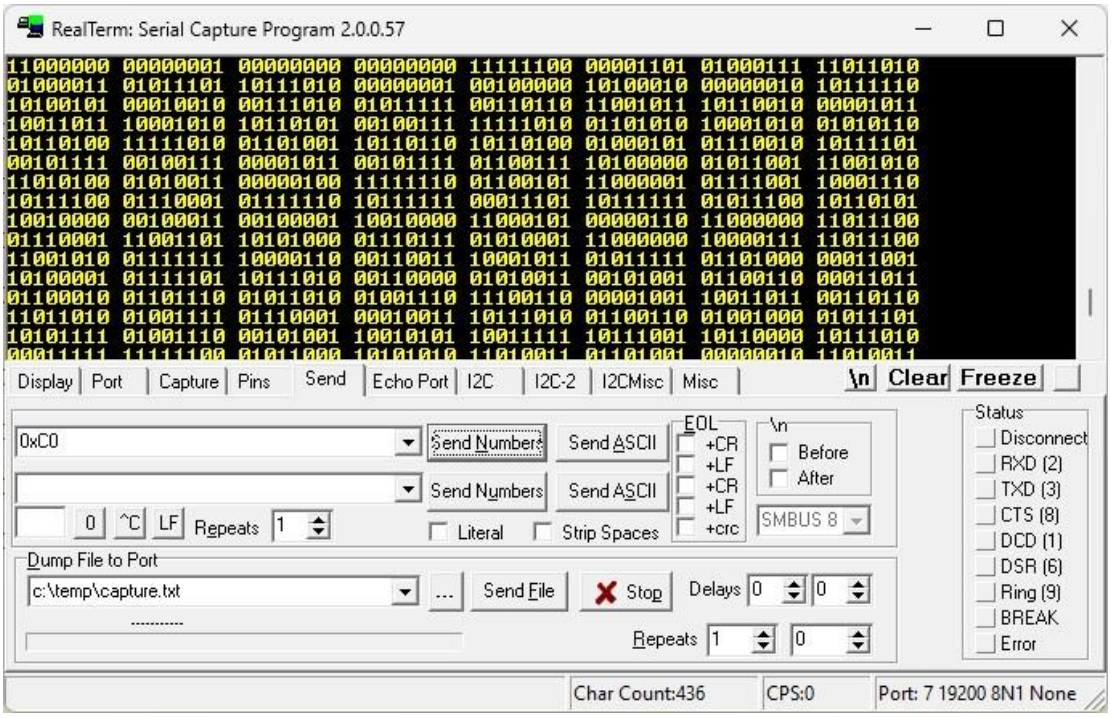
จากรูปที่ 4.185 แสดงผลการทดสอบสำหรับการพักข้อมูลจากระบบอื่น ๆ รวมถึงแรม เพื่อส่งต่อไปยังคอมพิวเตอร์ โดยสัญญาณประเภทหนึ่งจะถูกแบ่งชุดข้อมูลความยาว 8 บิต และถูกส่งไปยังคอมพิวเตอร์ การส่งชุดข้อมูลไปยังคอมพิวเตอร์จะทำการส่งชุดข้อมูลส่วนหัวก่อนตามด้วยความยาวของสัญญาณทั้งหมด จากนั้นชุดข้อมูลที่ถูกแบ่งความยาว 8 บิต จะถูกส่งต่อไปตามหลัง

4.3.15 ผลการทดสอบระบบย่อย Serial transmitter

ระบบย่อย Serial transmitter ทำหน้าที่ส่งข้อมูลด้วยโพรโทคอล UART ไปยังคอมพิวเตอร์ โดยชิปเอฟพีจีเอ จะส่งข้อมูลทีละ 8 บิตต่อเวลาอย่างอนุกรมมายังคอมพิวเตอร์ แสดงรายละเอียดระบบย่อย Serial transmitter ดังรูปที่ 4.186 โดยผลการทดสอบระบบย่อย Serial transmitter แสดงดังรูปที่ 4.187



รูปที่ 4.186 รายละเอียดระบบย่อย Serial transmitter



รูปที่ 4.187 ผลการทดสอบ timing diagram ของระบบย่อย Serial transmitter

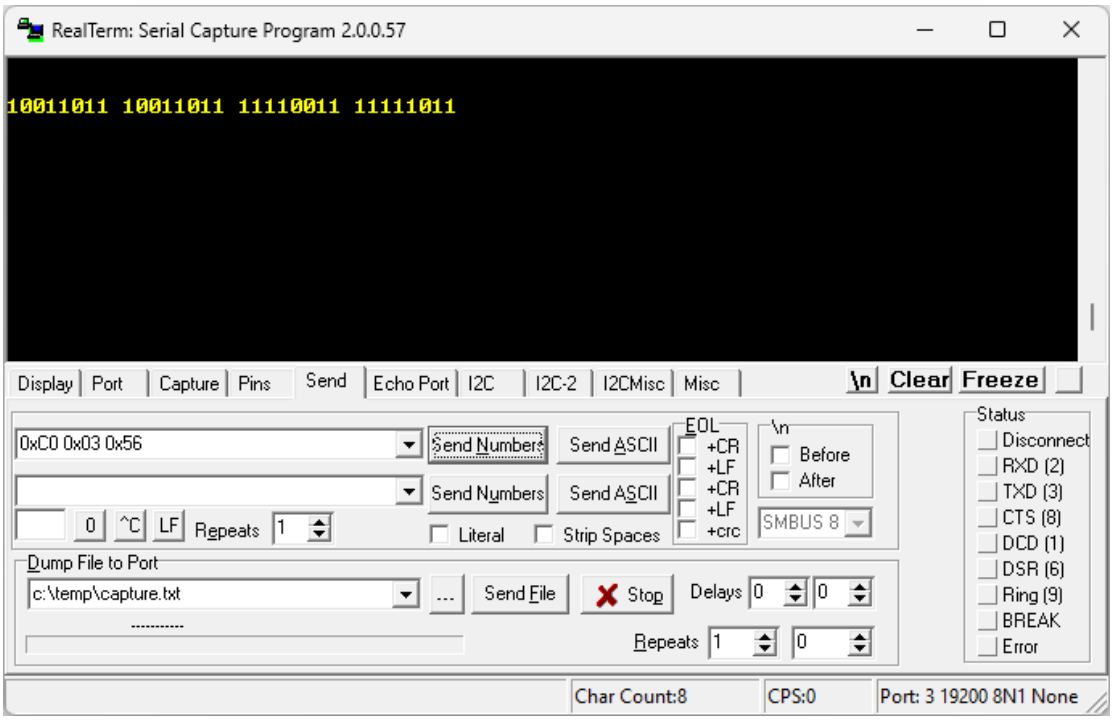
จากรูปที่ 4.187 แสดงผลการทดสอบการส่งข้อมูลจากระบบย่อย Serial receiver มายังคอมพิวเตอร์ และถูกแสดงผลบนโปรแกรม terminal บนคอมพิวเตอร์ เพื่อตรวจสอบว่าค่าที่รับเข้าไปยังระบบย่อยมีความถูกต้อง ผลการทดสอบดังกล่าวอาจแสดงผลการทดสอบร่วมกันของระบบย่อย Serial receiver และ Serial transmitter

4.4 ผลการทดสอบการเชื่อมต่อระหว่างอุปกรณ์ FPGA กับคอมพิวเตอร์

4.4.1 ผลการทดสอบการส่งข้อมูลจาก Computer ไปยังอุปกรณ์ FPGA

การทดสอบการส่งข้อมูลจาก Computer ไปยังอุปกรณ์ FPGA จะต้องส่งชุดบิตส่วนหัวก่อน จากนั้นตามด้วยชุดบิตข้อมูล และส่งชุดบิตส่วนหัวและชุดบิตข้อมูลให้ครบจำนวนประเภทข้อมูลของแต่ละโหมดหรือโมดูลของการใช้งาน ในตัวอย่างแสดงดังรูปที่ 4.188 ได้ส่งข้อมูลสำหรับโมดูล random number generator ที่ประกอบไปด้วยส่วนหัวสำหรับโมดูลนี้ 0xC0 และชุดบิตข้อมูลที่บ่งบอกความยาวของบิตที่จะถูกสุ่มขึ้นมาเท่ากับ 0x03 0x56 ซึ่งเท่ากับความยาว 854 บิต ชุดบิตข้อมูลทั้งหมดจะถูกส่งผ่าน serial port ผ่านโพรโทคอล UART ไปยังอุปกรณ์ FPGA เพื่อให้อุปกรณ์ FPGA อ่านค่าและทำงานตามโปรแกรมที่ใส่ไว้ในอุปกรณ์ หลังจากที่อุปกรณ์ FPGA ได้รับค่าจากคอมพิวเตอร์จะทำการอ่านค่าดังกล่าวทีละ 8 บิต โดยที่ 8 บิตแรกจะถูกตรวจจับสำหรับชุดบิตส่วนหัว เมื่อเจอชุดบิตส่วนหัวที่ถูกต้องตามโปรแกรม ค่า 8 บิตถัดไปจะถูกเก็บไว้ตามประเภทของส่วนหัว โดยจะทำการนับชุดบิตข้อมูลละ 8 บิต เป็นจำนวนชุดตามที่ตั้งค่าไว้ในโปรแกรม เมื่อเก็บค่าชุดบิตข้อมูลตามจำนวน

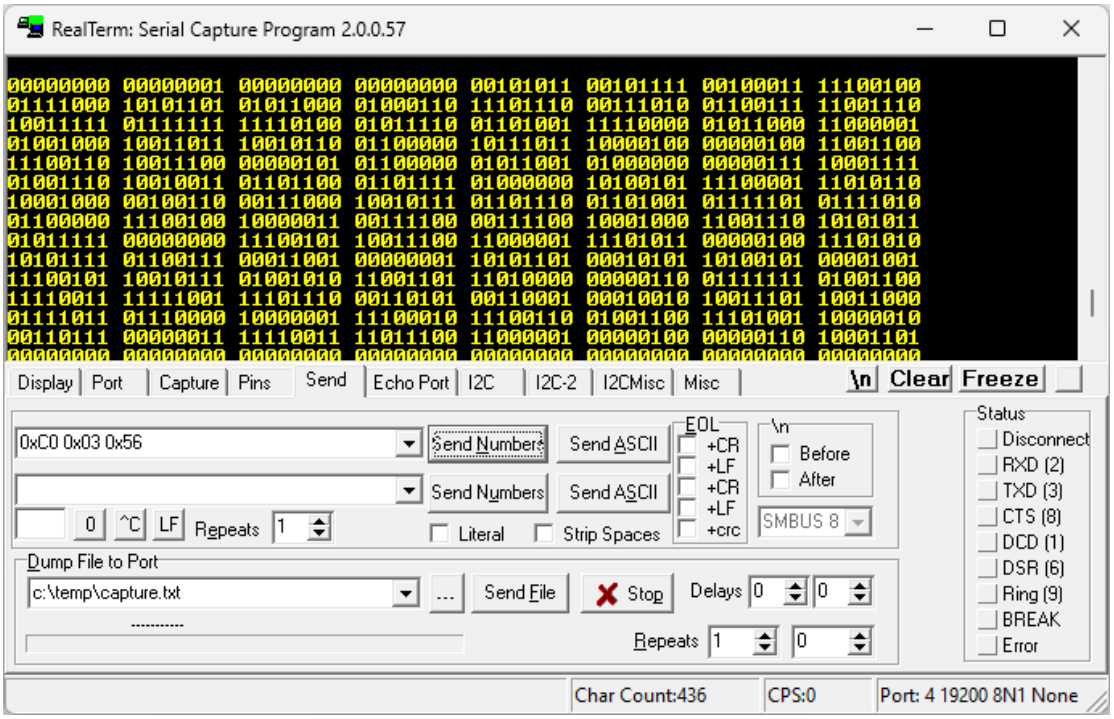
ที่ถูกต้อง อุปกรณ์ FPGA จะทำการตรวจจับส่วนหัวประเภทอื่นต่อไป เมื่อชุดบิตข้อมูลทั้งหมดถูกจัดเก็บทั้งหมดครบตามจำนวนประเภทของแต่ละโหมดหรือโมดูลแล้ว บิตข้อมูลเหล่านี้จะถูกส่งต่อไปยังส่วนประมวลผลถัดไปภายในอุปกรณ์ FPGA รูปที่ 4.188 ชุดบิตข้อมูลทั้งหมดได้ถูกส่งผ่าน serial port แบบ enhanced ซึ่งมีหน้าที่ในการสะท้อน (echo) ค่าที่ส่งกลับมายังคอมพิวเตอร์ และสามารถชุดบิตข้อมูลที่ถูกสะท้อนได้บนโปรแกรมคอมพิวเตอร์



รูปที่ 4.188 ชุดข้อมูลที่ส่งไปยังคอมพิวเตอร์

4.4.2 ผลการทดสอบการส่งข้อมูลจากอุปกรณ์ FPGA ไปยัง Computer

การทดสอบการส่งข้อมูลจากอุปกรณ์ FPGA ไปยัง Computer จะต้องส่งชุดบิตส่วนหัวก่อน จากนั้นตามด้วยจำนวนส่วน ความยาวของชุดบิตข้อมูล และสุดท้ายตามด้วยชุดบิตข้อูมล จำนวนส่วนสามารถบ่งบอกคอมพิวเตอร์ได้ว่าส่วนหัวดังกล่าว จะมีการส่งชุดบิตข้อมูลเป็นจำนวนกี่ครั้ง ประเภทของชุดบิตข้อมูลจะขึ้นอยู่กับแต่ละโหมดหรือโมดูลของการใช้งาน ในตัวอย่างแสดงดังรูปที่ 4.189 คอมพิวเตอร์ที่รับข้อมูลจากอุปกรณ์ FPGA จากโมดูล random number generator ที่ประกอบด้วยส่วนหัว จำนวนส่วน ความยาวชุดบิตข้อมูล และชุดบิตข้อมูล เมื่อชุดบิตข้อมูลทั้งหมดถุกส่งไปยังคอมพิวเตอร์ คอมพิวเตอร์จะรับชุดบิตข้อมูลทั้งหมดและทำการหั่นชุดบิตออกเป็นส่วน เพื่อตรวจจับชุดบิตส่วนหัว จำนวนส่วน ความยาวชุดบิตข้อมูล และบิตข้อมูล

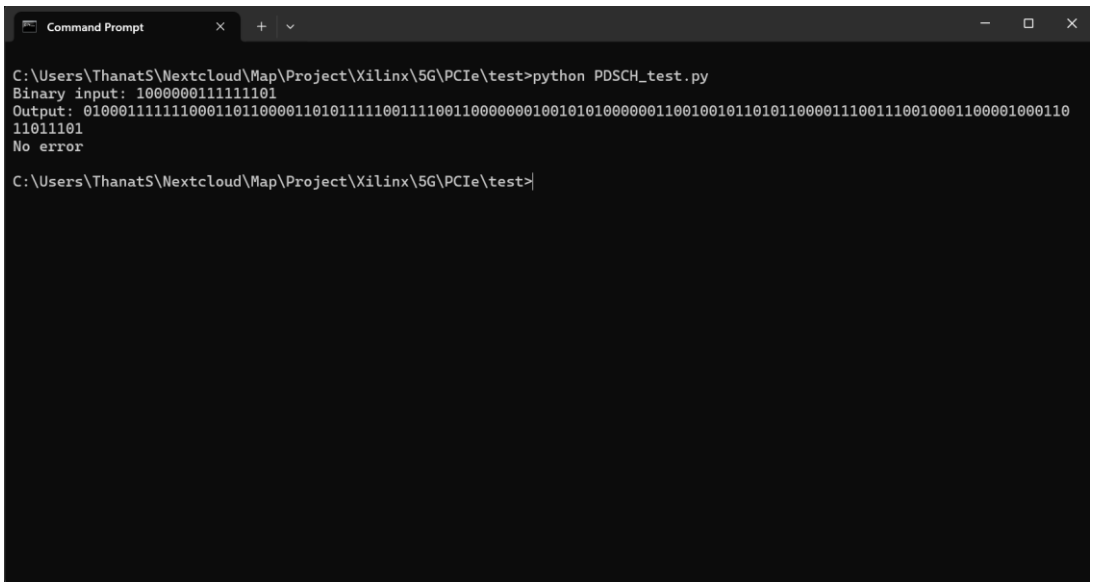


รูปที่ 4.189 ชุดข้อมูลที่รับมาจากอุปกรณ์ FPGA

4.5 ผลการทดสอบการทำงานของการ์ดเร่งความเร็ว FEC

4.5.1 ทดสอบการทำงานแบบ AAL_PDSCH

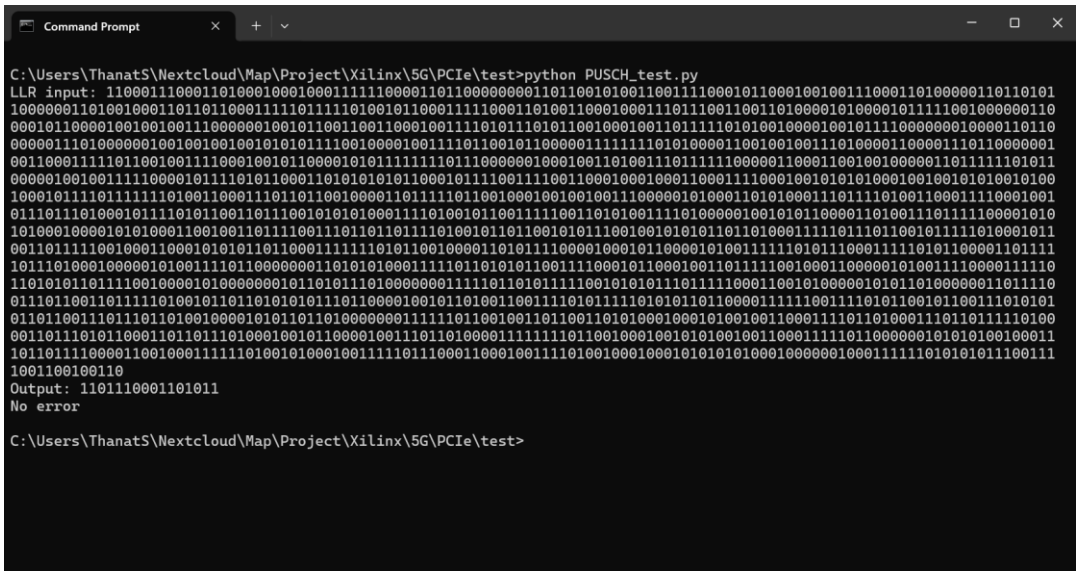
ทำการทดสอบโดยใช้โปรแกรมภาษา Python ให้เรียกใช้ XDMA เพื่อรับส่งข้อมูลสำหรับ AAL_PDSCH ผลการทดสอบเป็นดังรูปที่ 4.190 ซึ่งพบว่าค่าถูกต้องเมื่อเทียบกับโปรแกรม GUI



รูปที่ 4.190 ผลการทดสอบการ์ดเร่งความเร็ว FEC แบบ AAL_PDSCH_FEC

4.5.2 ทดสอบการทำงานแบบ AAL_PUSCH

ทำการทดสอบโดยใช้โปรแกรมภาษา Python ให้เรียกใช้ Driver XDMA เพื่อรับส่งข้อมูลสำหรับ AAL_PUSCH ผลการทดสอบเป็นดังรูปที่ 4.191 ซึ่งพบว่าค่าถูกต้องเมื่อเทียบกับโปรแกรม GUI



รูปที่ 4.191 ผลการทดสอบการ์ดเร่งความเร็ว FEC แบบ AAL_PUSCH_FEC

4.6 ผลการทดสอบการเชื่อมต่อระหว่างการ์ดเร่งความเร็ว FEC กับคอมพิวเตอร์

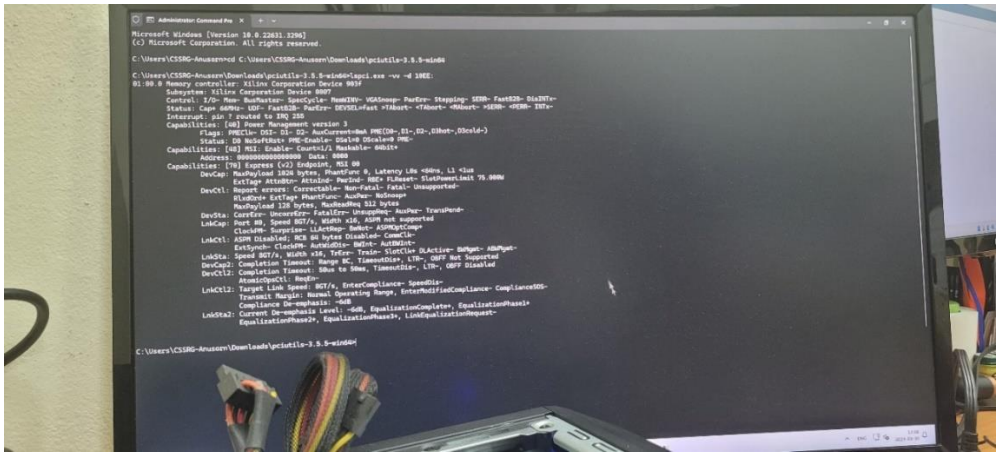
4.6.1 การจัดเตรียมฮาร์ดแวร์

ทำการตั้งค่าอุปกรณ์ FPGA ตามคู่มือกำหนดสำหรับการใช้งาน PCIe แล้วจึงนำอุปกรณ์เสียบลงบนเครื่องคอมพิวเตอร์ดังรูปที่ 4.192 และทำการตรวจสอบว่าอุปกรณ์ยังสามารถเปิดใช้งานตามปกติ

รูปที่ 4.192 ผลการทดสอบฮาร์ดแวร์หลังการจัดเตรียม

4.6.2 ทดสอบ IP-Core DMA/Bridge Subsystem for PCI Express ของ Xilinx

ทดสอบ IP-Core DMA/Bridge Subsystem for PCI Express โดยการทำตามตัวอย่างในคู่มือของ Xilinx ได้ผลดังรูปที่ 4.193 ซึ่งหมายความว่าระบบปฏิบัติการได้ตรวจพบอุปกรณ์ FPGA ตัวนี้เป็นหนึ่งในอุปกรณ์ต่อพ่วง PCIe



รูปที่ 4.193 ผลการทดสอบ IP-Core DMA/Bridge Subsystem for PCI Express ของ Xilinx

4.7 ผลการทดสอบเว็บไซต์เผยแพร่ชุดการเรียนรู้และทดสอบสมรรถนะระบบสื่อสารมาตรฐาน 5G

โครงการนี้สร้างเว็บไซต์เพื่อเผยแพร่ชุดการเรียนรู้และทดสอบสมรรถนะระบบสื่อสาร โดยสามารถเข้าถึงได้ที่ https://www.channelcoding.com ภายในเว็บไซต์ประกอบด้วย 6 แถบหลัก ได้แก่ หน้าแรก รหัสช่องสัญญาณ การมอดูเลชัน ช่องสัญญาณรบกวน ชุดการเรียนรู้และทดสอบ และเกี่ยวกับเรา โดยแต่ละแถบมีรายละเอียดดังนี้

4.7.1 แถบหน้าแรก

แถบหน้าแรก เป็นแถบที่ประกอบด้วยหน้าต่างหน้าแรกซึ่งแสดงข้อมูลแนะนำเว็บไซต์และผู้สนับสนุน เพื่อใช้ผู้ใช้เห็นภาพรวมของเว็บไซต์ นอกจากนี้ยังสามารถเข้าถึงบทความที่อัพเดทล่าสุดและบทความแนะนำจากแถบด้านขวาของหน้าต่าง ผลการทดสอบเว็บไซต์หน้าต่างหน้าแรกแสดงดังรูปที่ 4.194

รูปที่ 4.194 ผลการทดสอบเว็บไซต์หน้าต่างหน้าแรก

4.7.2 แถบรหัสช่องสัญญาณ

แถบรหัสช่องสัญญาณ เป็นแถบที่ประกอบด้วยเนื้อหา 4 เรื่อง ดังแสดงในรูปที่ 4.195 โดยแต่ละเรื่องจะประกอบด้วยหน้าต่างแสดงรายละเอียดเนื้อหาของเรื่องย่อยดังตารางที่ 4.2 และตัวอย่างหน้าต่างเนื้อหาเรื่อง “ช่องสัญญาณคืออะไร” แสดงรูปที่ 4.196

ตารางที่ 4.2 รายการเนื้อหาภายในแถบรหัสช่องสัญญาณ

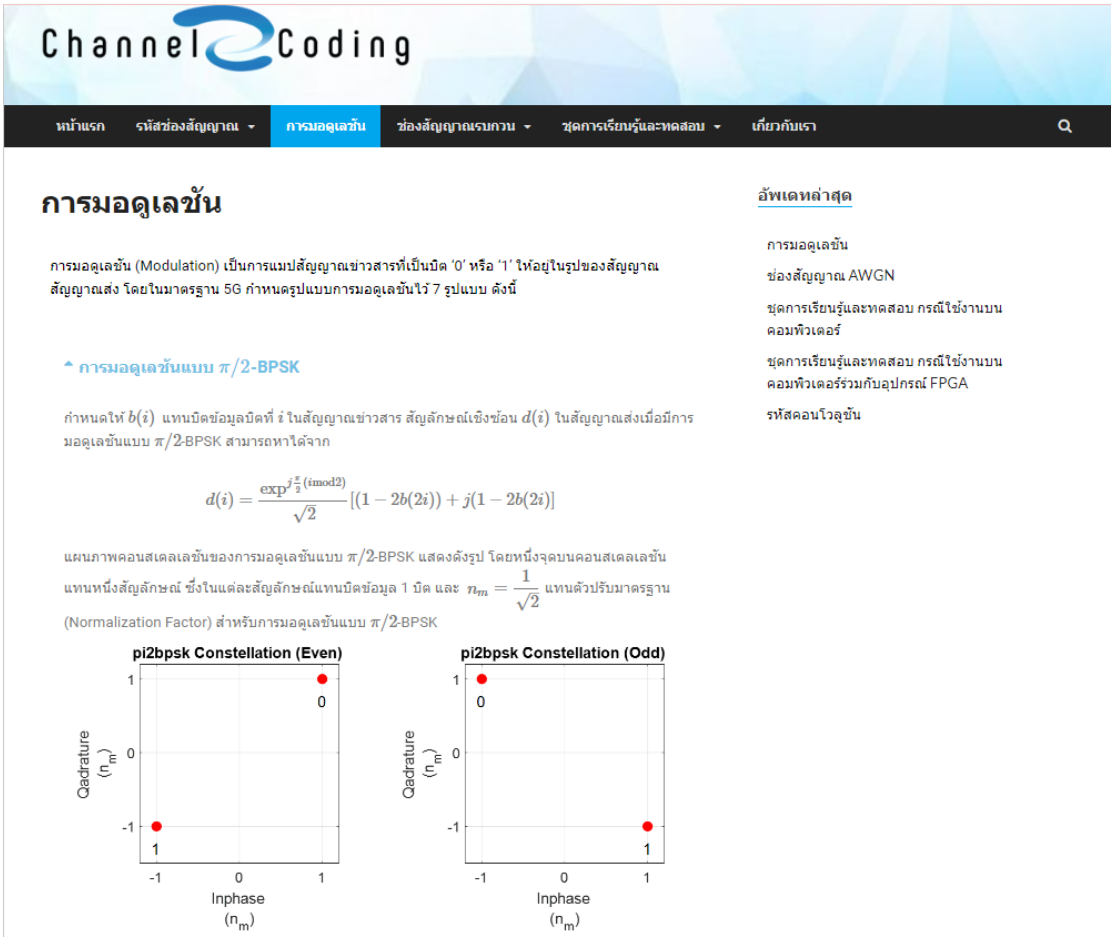
ชื่อเรื่อง	รายการเรื่องย่อย
รหัสช่องสัญญาณคืออะไร	รหัสช่องสัญญาณคืออะไร
รหัสช่องสัญญาณที่ได้ความนิยม	รหัสแฮมมิ่ง รหัสเทอร์โบ รหัส LDPC รหัสโพลาร์
มาตรฐาน 5G (3GPP NR)	การสื่อสารไร้สายยุคที่ 5 ประเภทของรหัสช่องสัญญาณในมาตรฐาน 5G ความเป็นมาของมาตรฐานการเข้ารหัส 5G การเข้ารหัสแอลดีพีมาตรฐาน 5G การเข้ารหัสโพลาร์มาตรฐาน 5G
มาตรฐาน Wi-Fi (IEEE 802.11)	ความเป็นมามาตรฐาน IEEE 802.11 วิวัฒนาการของมาตรฐาน IEEE 802.11 ลักษณะการเชื่อมต่อของอุปกรณ์ IEEE 802.11 การเข้ารหัส LDPC IEEE 802.11

รูปที่ 4.195 ผลการทดสอบเว็บไซต์ตัวเลือกเนื้อหาในแถบรหัสช่องสัญญาณ

รูปที่ 4.196 ผลการทดสอบเว็บไซต์หน้าต่างช่องสัญญาณคืออะไร

4.7.3 แถบการมอดูเลชัน

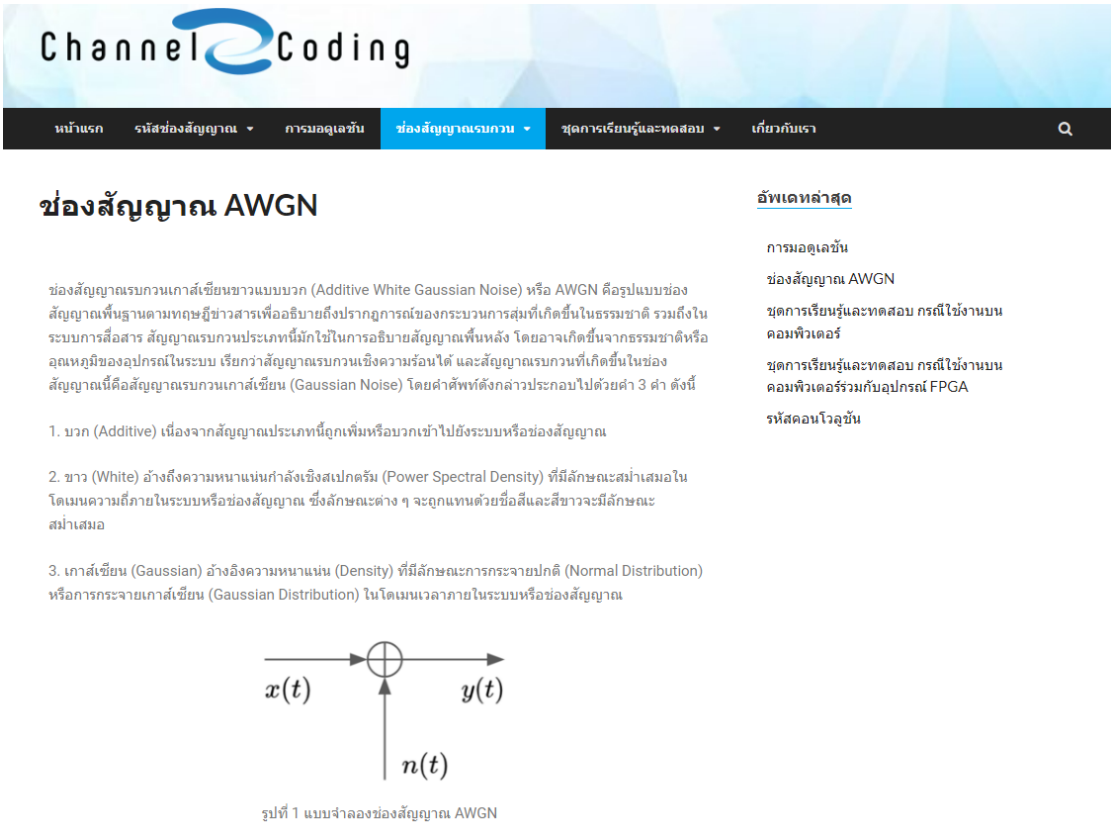
แถบการมอดูเลชัน เป็นแถบแสดงเนื้อหาเกี่ยวกับการมอดูเลชันมาตรฐาน 5G ซึ่งประกอบด้วย 7 รูปแบบ ได้แก่ π/2-BPSK, BPSK, QPSK, 16QAM, 64QAM, 256QAM และ 1024QAM โดยเนื้อหาแต่ละรูปแบบการมอดูเลชันจะแสดงแบบ Toggle ซึ่งผู้ใช้สามารถย่อหรือขยายเพื่อเลือกดูรูปแบบการมอดูเลชันที่สนใจได้ ตัวอย่างผลการทดสอบเว็บไซต์หน้าต่างการมอดูเลชันแสดงดังรูปที่ 4.197



รูปที่ 4.197 ผลการทดสอบเว็บไซต์หน้าต่างการมอดูเลชัน

4.7.4 แถบช่องสัญญาณรบกวน

แถบช่องสัญญาณรบกวน เป็นแถบแสดงเนื้อหาเกี่ยวกับช่องสัญญาณ AWGN ซึ่งประกอบไปด้วยนิยามของช่องสัญญาณ AWGN สมการที่สำคัญ และรูปภาพประกอบ ตัวอย่างผลการทดสอบเว็บไซต์หน้าต่างช่องสัญญาณ AWGN แสดงดังรูปที่ 4.198



รูปที่ 4.198 ผลการทดสอบเว็บไซต์หน้าต่างช่องสัญญาณ AWGN

4.7.5 แถบชุดการเรียนรู้และทดสอบ

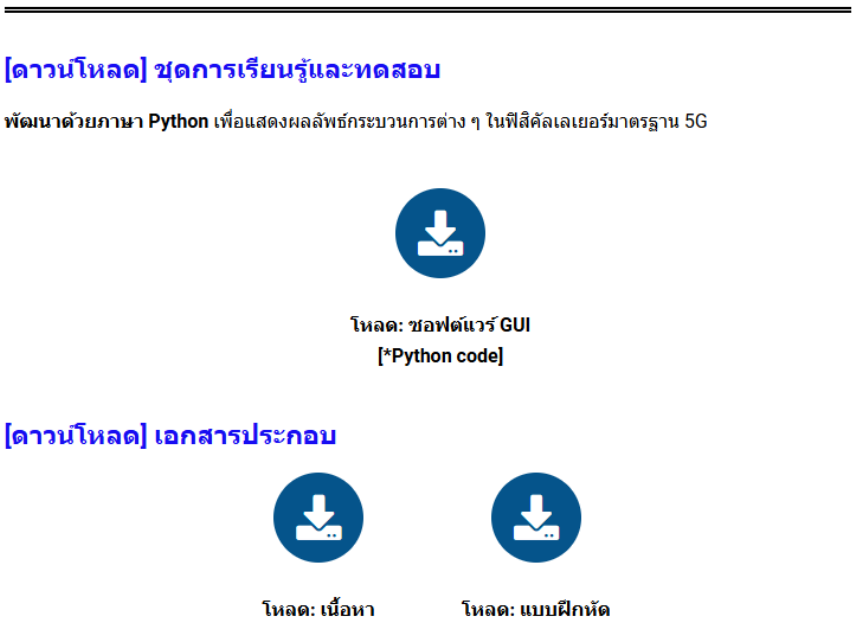
แถบชุดการเรียนรู้และทดสอบ เป็นประกอบด้วย 2 หน้าต่าง ได้แก่ หน้าต่างกรณีใช้งานบนคอมพิวเตอร์ และใช้งานบนคอมพิวเตอร์ร่วมกับอุปกรณ์ FPGA ดังแสดงในรูปที่ 4.199 - รูปที่ 4.200 โดยทั้งสองหน้าต่างจะแสดงตัวอย่างการเรียกใช้งาน Module พร้อมทั้งอธิบายองค์ประกอบของซอฟต์แวร์ GUI โดยผู้ใช้สามารถดาวน์โหลดซอฟต์แวร์ GUI เนื้อหาความรู้ และแบบฝึกหัด บริเวณส่วนท้ายของหน้าต่างเว็บไซต์ ดังแสดงในรูปที่ 4.201 นอกจากนี้บนต่างเว็บไซต์ยังมีวิดีโอแสดงตัวอย่างการใช้งานซอฟต์แวร์ GUI ดังแสดงในรูปที่ 4.202

รูปที่ 1 หน้าต่าง GUI แถบ 5G Modulation ที่เรียกใช้ซอฟต์แวร์บนคอมพิวเตอร์

รูปที่ 4.199 ผลการทดสอบเว็บไซต์หน้าต่างชุดการเรียนรู้และทดสอบกรณีใช้งานบนคอมพิวเตอร์

รูปที่ 1 หน้าต่าง GUI แถบ 5G Modulation ที่เรียกใช้ซอฟต์แวร์บน FPGA

รูปที่ 4.200 ผลการทดสอบเว็บไซต์หน้าต่างชุดการเรียนรู้และทดสอบ กรณีใช้งานบนคอมพิวเตอร์ร่วมกับอุปกรณ์ FPGA

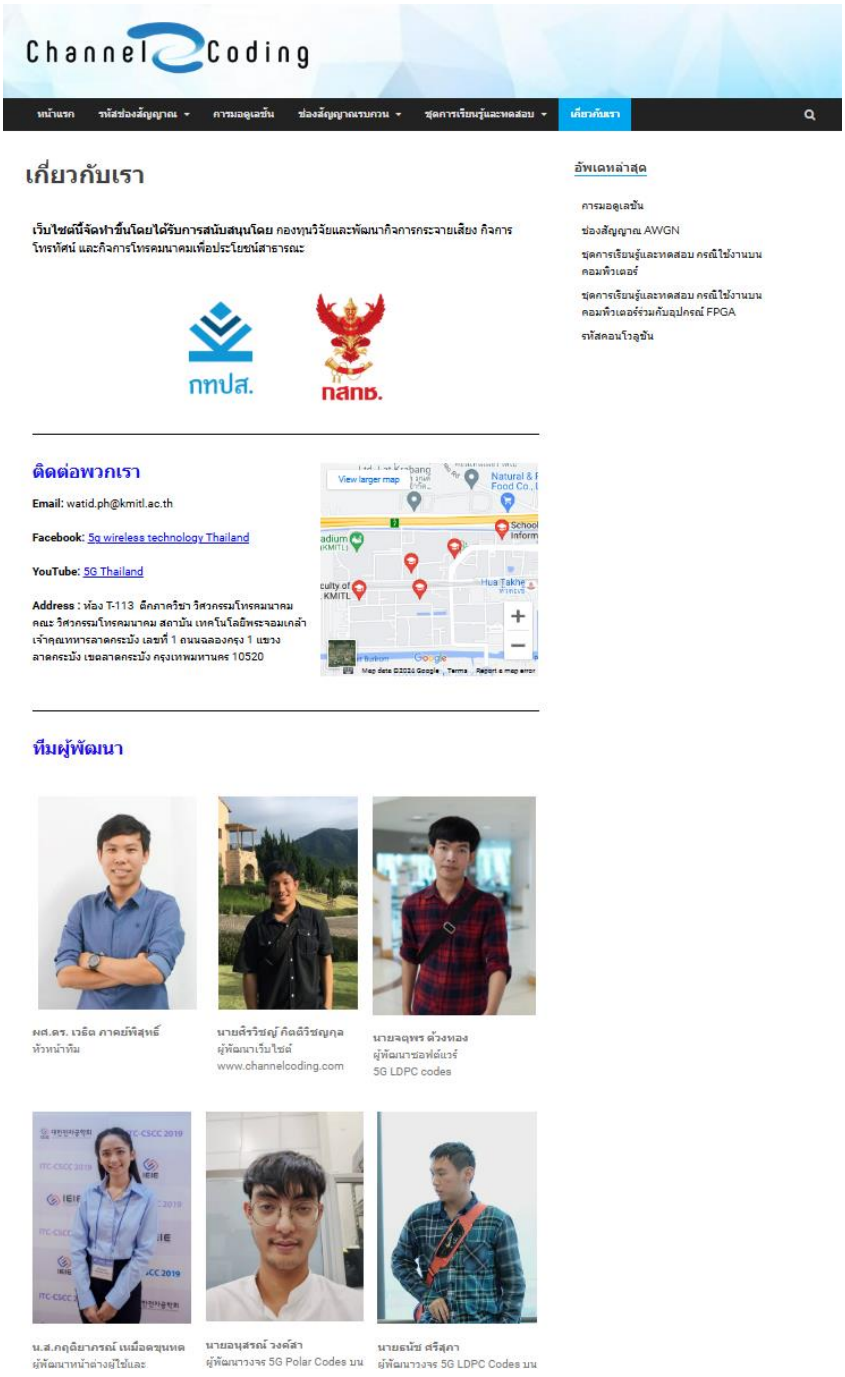


รูปที่ 4.201 ส่วนสำหรับดาวน์โหลดซอฟต์แวร์ GUI และเอกสารประกอบ
บนหน้าต่างชุดการเรียนรู้และทดสอบ

รูปที่ 4.202 แสดงวิดีโอสาธิตการใช้งานซอฟต์แวร์ GUI บนหน้าต่างชุดการเรียนรู้และทดสอบ

4.7.5 แถบเกี่ยวกับเรา

แถบเกี่ยวกับเรา เป็นแถบที่ประกอบด้วยหน้าต่างเกี่ยวกับเรา ซึ่งแสดงข้อมูลผู้สนับสนุนโครงการ ข้อมูลติดต่อและข้อมูลความเชี่ยวชาญของทีมผู้พัฒนาโครงการ ดังแสดงในรูปที่ 4.203



รูปที่ 4.203 ผลการทดสอบเว็บไซต์หน้าต่างเกี่ยวกับเรา

4.8 ผลการเผยแพร่ชุดการเรียนรู้และทดสอบสมรรถนะระบบสื่อสารมาตรฐาน 5G ในมหาวิทยาลัย

4.8.1 ภาพรวมของการเผยแพร่ชุดการเรียนรู้และทดสอบสมรรถนะในมหาวิทยาลัย

การเผยแพร่ชุดการเรียนรู้และทดสอบสมรรถนะในมหาวิทยาลัยเพื่อให้นักศึกษาได้เรียนรู้และทดสอบได้ถูกจัดขึ้นที่มหาวิทยาลัย 5 แห่ง ได้แก่

- สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- มหาวิทยาลัยราชมงคลอีสาน วิทยาเขตนครราชสีมา
- มหาวิทยาลัยราชมงคลอีสาน วิทยาเขตขอนแก่น
- มหาวิทยาลัยขอนแก่น
- จุฬาลงกรณ์มหาวิทยาลัย

การเผยแพร่ชุดการเรียนรู้มี 2 ส่วนคือการให้ความรู้พื้นฐานที่เกี่ยวข้องกับระบบสื่อสารในมาตรฐาน 5G และการทดสอบใช้งานซอฟต์แวร์ชุดการเรียนรู้ระบบสื่อสารมาตรฐาน 5G ในส่วนของการให้ความรู้พื้นฐานได้มีการจัดทำเอกสารให้ความรู้ และมีแบบฝึกหัดให้นักศึกษาได้ทดสอบความเข้าใจในความรู้พื้นฐานดังแสดงในหัวข้อถัดไป

การถ่ายทอดความรู้พื้นฐานบรรยายโดยวิทยากร 4 ท่านได้ แก่ รศ.ดร.เวธิต ภาคย์พิสุทธิ์ นางสาวกฤติยาภรณ์ เหมือดขุนทด นายจตุพร ด้วงทอง และนายอนุสรณ์ วงค์ษา ในการบรรยายในช่วงแรกโดย รศ.ดร.เวธิต ภาคย์พิสุทธิ์ ซึ่งมีเนื้อหาในการบรรยายที่เกี่ยวข้องกับภาพรวมของมาตรฐานสื่อสาร 5G เพื่อปูให้นักศึกษาที่เข้าฟังการบรรยายได้เห็นภาพรวมของการสื่อสารมาตรฐาน 5G การบรรยายในช่วงที่ 2 โดยนางสาวกฤติยาภรณ์ เหมือดขุนทด ซึ่งมีเนื้อหาในการบรรยายที่เกี่ยวข้องกับระบบสื่อสาร 5G ในส่วนของการมอดูเลชันมาตรฐาน 5G และช่องสัญญาณในระบบสื่อสารได้แก่ AWGN (Adaptive white Gaussian noise) และช่องสัญญาณการจางหาย การบรรยายในช่วงที่ 3 โดยนายจตุพร ด้วงทอง ซึ่งมีเนื้อหาในการบรรยายเกี่ยวข้องกับรหัสช่องสัญญาณ ได้แก่ รหัส LDPC (Low-density parity-check) ที่ถูกใช้งานในมาตรฐาน 5G เพื่อปกป้องข้อมูลผู้ใช้จากความผิดพลาดที่เกิดจากสัญญาณรบกวน การบรรยายในช่วงสุดท้ายโดยนายอนุสรณ์ วงค์ษา ซึ่งมีเนื้อหาบรรยายเกี่ยวกับรหัสช่องสัญญาณที่เป็นรหัสโพลาร์ (Polar) ซึ่งถูกใช้เพื่อป้องกันสัญญาณควบคุมจากสัญญาณรบกวน หลังจากจบการบรรยายนักศึกษาได้ลองใช้ชุดซอฟต์แวร์การเรียนรู้โดยให้มีการปรับค่าพารามิเตอร์ต่างๆ เพื่อดูผลลัพธ์ซึ่งเป็นการนำความรู้ที่ได้จากการบรรยายมาใช้ ทำให้นักศึกษามีความเข้าใจมากยิ่งขึ้นและทำให้มีความสนุกในการเรียนรู้

4.8.2 สรุปการเผยแพร่ชุดการเรียนรู้และทดสอบสมรรถนะในมหาวิทยาลัย

1) สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

วิชา Telecommunications Laboratory

วันที่จัดอบรม

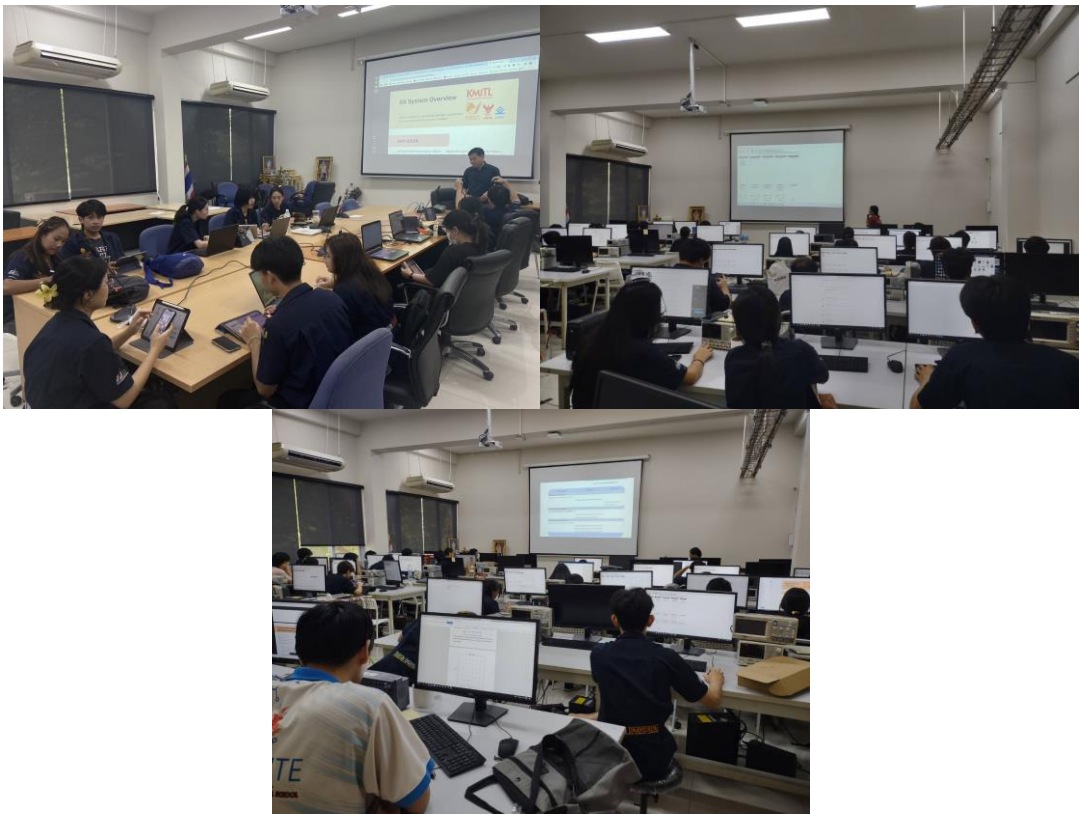
ครั้งที่ 1: 5 กุมภาพันธ์ 2567

ครั้งที่ 2: 12 กุมภาพันธ์ 2567

ครั้งที่ 3: 13 มีนาคม 2567

จำนวนนักศึกษาที่เข้าร่วม 96 คน

ภาพบรรยากาศการเผยแพร่ชุดการเรียนรู้และทดสอบสมรรถนะแสดงดังรูปที่ 4.204



รูปที่ 4.204 บรรยากาศการเผยแพร่ชุดการเรียนรู้ที่สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

2) มหาวิทยาลัยราชมงคลอีสาน วิทยาเขตนครราชสีมา

วิชา Industrial Data Communications

วันที่จัดอบรม 16 กุมภาพันธ์ 2567

จำนวนนักศึกษาที่เข้าร่วม 50 คน

ภาพบรรยากาศการเผยแพร่ชุดการเรียนรู้และทดสอบสมรรถนะแสดงดังรูปที่ 4.205

รูปที่ 4.205 บรรยากาศการเผยแพร่ชุดการเรียนรู้ที่มหาวิทยาลัยราชมงคลอีสาน วิทยาเขตนครราชสีมา

3) มหาวิทยาลัยราชมงคลอีสาน วิทยาเขตขอนแก่น

วิชา Data Communication

วันที่จัดอบรม 19 กุมภาพันธ์ 2567

จำนวนนักศึกษาที่เข้าร่วม 15 คน

ภาพบรรยากาศการเผยแพร่ชุดการเรียนรู้และทดสอบสมรรถนะแสดงดังรูปที่ 4.206

รูปที่ 4.206 บรรยากาศการเผยแพร่ชุดการเรียนรู้ที่มหาวิทยาลัยราชมงคลอีสาน วิทยาเขตขอนแก่น

4) มหาวิทยาลัยขอนแก่น

วิชา Forward Error Correcting Coding

วันที่จัดอบรม 20 กุมภาพันธ์ 2567

จำนวนนักศึกษาที่เข้าร่วม 20 คน

ภาพบรรยากาศการเผยแพร่ชุดการเรียนรู้และทดสอบสมรรถนะแสดงดังรูปที่ 4.207

รูปที่ 4.207 บรรยากาศการเผยแพร่ชุดการเรียนรู้ที่มหาวิทยาลัยขอนแก่น

5) จุฬาลงกรณ์มหาวิทยาลัย

วิชา Digital Communications

วันที่จัดอบรม 18 มีนาคม 2567

จำนวนนักศึกษาที่เข้าร่วม 8 คน

ภาพบรรยากาศการเผยแพร่ชุดการเรียนรู้และทดสอบสมรรถนะแสดงดังรูปที่ 4.208

รูปที่ 4.208 บรรยากาศการเผยแพร่ชุดการเรียนรู้ที่จุฬาลงกรณ์มหาวิทยาลัย

4.9 ผลการเผยแพร่ข้อมูลการออกแบบการ์ดเร่งความเร็วในโครงข่าย O-RAN ในบริษัทเอกชน

นอกจากการเผยแพร่ผลผลิตของโครงการวิจัยให้แก่อาจารย์ในมหาวิทยาลัยต่าง ๆ เพื่อนำไปใช้เป็นสื่อสารการสอนแก่นักศึกษา โครงการวิจัยนี้ ยังได้ติดต่อบริษัทเอกชนที่มีการดำเนินธุรกิจเกี่ยวข้องกับเทคโนโลยีหรือผลผลิตของโครงการวิจัย เพื่อทำให้ทราบถึงการพัฒนาอุปกรณ์ FPGA ที่ได้รับการสนับสนุนจากหน่วยงาน กสทช. โดยทำการติดต่อบริษัทจำนวน 2 บริษัท ได้แก่

1. บริษัท พาซเวรี่ จำกัด ปัจจุบันดำเนินธุรกิจด้านการออกแบบระบบประมวลผลสัญญาณให้แก่บริษัทผู้ผลิตชิปที่ประเทศสหรัฐอเมริกา โดยระบบประมวลผลสัญญาณดังกล่าวจะถูกนำไปใช้ในอุปกรณ์บันทึกข้อมูลดิจิทัล เช่น อุปกรณ์จำพวก hard disk drive และ solid-state drive นอกจากระบบประมวลผลสัญญาณแล้ว บริษัท พาซเวรี่ จำกัด ยังมีประสบการณ์ในการออกแบบรหัสช่องสัญญาณเช่น LDPC codes ด้วยเช่นกัน อย่างไรก็ตาม ทางบริษัทมิได้ออกแบบรหัสช่องสัญญาณหรือระบบประมวลผลสัญญาณของอุปกรณ์สื่อสารไร้สาย 5G ทั้งนี้ จากการหารือ ผู้วิจัยได้รับคำแนะนำที่สามารถนำมาประยุกต์ใช้ในการออกแบบอุปกรณ์สื่อสารไร้สาย 5G ได้ รวมถึงได้รับการสนับสนุนในเชิงเทคนิคเพื่อผลักดันงานวิจัยหรือการพัฒนาอุปกรณ์ FPGA ในอนาคต เพื่อให้สามารถแข่งขันกับบริษัทเทคโนโลยีของต่างประเทศได้

2. บริษัท ดีไซน์ เกทเวย์ อินเตอร์เนชั่นแนล จำกัด ปัจจุบันดำเนินธุรกิจด้านการออกแบบอุปกรณ์ FPGA ให้แก่บริษัทต่างประเทศ ทั้งนี้ บริษัทจะมุ่งเน้นการออกแบบกระบวนการอ่านและเขียนหน่วยความจำเท่านั้น มิได้มีการออกแบบที่เกี่ยวข้องกับเทคโนโลยีสื่อสาร 5G อย่างไรก็ตาม จากการที่ได้นำเสนอผลผลิตเบื้องต้นของโครงการ ทาง บริษัท ดีไซน์ เกทเวย์ อินเตอร์เนชั่นแนล มีความสนใจที่สนับสนุนและส่งเสริมการพัฒนาอุปกรณ์ FPGA ในมหาวิทยาลัยเพื่อผลิตบุคลากรเข้าสู่ตลาดแรงงานไทย ซึ่งปัจจุบันประเทศไทยมีบุคลากรที่เชี่ยวชาญ FPGA อย่างจำกัด แตกต่างจากประเทศที่มีเทคโนโลยีชั้นสูง นอกจากนี้ ทางบริษัทมีความยินดีที่จะช่วยสอบถามกับลูกค้าต่างประเทศของบริษัท เพื่อสอบถามความต้องการของผลผลิตที่โครงการนี้จัดทำขึ้น

บทที่ 5
สรุปผลการวิจัย และข้อเสนอแนะ

โครงการวิจัยนี้มีเป้าหมาย 2 ด้านได้แก่ 1) มิติด้านการศึกษาวิศวกรรมโทรคมนาคม และ 2) มิติด้านการพัฒนาอุปกรณ์โทรคมนาคมต้นแบบ โดยสรุปผลการวิจัยและข้อเสนอแนะของแต่ละเป้าหมายมีดังต่อไปนี้

1) มิติด้านการศึกษาวิศวกรรมโทรคมนาคม

ทีมวิจัยมีความประสงค์ที่จะพัฒนาอุปกรณ์การศึกษาที่สอดคล้องกับเทคโนโลยีสื่อสารในปัจจุบัน โดยทีมวิจัยได้ศึกษามาตรฐาน 3GPP TS 38.212 ซึ่งกำหนดกระบวนการเข้ารหัสช่องสัญญาณและมาตรฐาน 3GPP TS 38.211 ซึ่งกำหนดการมอดูเลชันที่ใช้ในการสื่อสารไร้สายยุค 5G โดยทีมวิจัยได้พัฒนาซอฟต์แวร์ที่สอดคล้องกับมาตรฐานดังกล่าวด้วยภาษา python เพื่อให้นักศึกษามีความรู้ความเข้าใจอย่างลึกซึ้ง นอกจากนี้ ทีมวิจัยได้พัฒนาซอฟต์แวร์ประเภท GUI เพื่ออธิบายเทคนิคการมอดูเลชัน การเข้ารหัสช่องสัญญาณ และการส่งข้อมูลผ่านช่องสัญญาสื่อสารไร้สาย โดยนักศึกษาสามารถทำการเรียนรู้ขั้นตอนต่าง ๆ ของการสื่อสารมาตรฐาน 5G อีกทั้ง สามารถกำหนดอินพุตและทดสอบหาเอาต์พุตของขั้นตอนต่าง ๆ ได้โดยง่าย ทีมวิจัยได้จัดทำเอกสารที่มีเนื้อหาทฤษฎีและปฏิบัติ (รายละเอียดอยู่ในภาคผนวก ก และ ข) เพื่อใช้ในการจัดการเรียนการสอนในมหาวิทยาลัย ปัจจุบัน ซอฟต์แวร์และเอกสารนี้ถูกใช้ในการจัดการเรียนการสอนของหลักสูตรวิศวกรรมโทรคมนาคม เช่น สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง มหาวิทยาลัยเทคโนโลยีราชมงคลอีสาน มหาวิทยาลัยขอนแก่น จุฬาลงกรณ์มหาวิทยาลัย เป็นต้น

ทีมวิจัยได้พัฒนาเว็บไซต์ https://www.channelcoding.com เพื่อเผยแพร่เนื้อหาความรู้ด้านรหัสช่องสัญญาณและซอฟต์แวร์เข้ารหัสที่ทีมวิจัยได้พัฒนาขึ้น อีกทั้งเผยแพร่เอกสารต่าง ๆ ทีมวิจัยคาดหวังว่าซอฟต์แวร์ที่พัฒนาขึ้นจะสามารถทำให้นักศึกษาไทยที่ศึกษาในด้านวิศวกรรมโทรคมนาคมมีความรู้ความเข้าใจเทคโนโลยีที่ใช้อยู่ในปัจจุบัน และก่อให้เกิดแรกผลักดันที่ทำให้นักศึกษาอยากที่จะพัฒนาเทคโนโลยีด้านวิศวกรรมโทรคมนาคมขึ้นมาใช้งานเอง

2) มิติของการพัฒนาอุปกรณ์โทรคมนาคมต้นแบบ

ทีมวิจัยมีความประสงค์ที่จะพัฒนาอุปกรณ์เข้ารหัสและถอดรหัสช่องสัญญาณ อุปกรณ์มอดูเลชัน อุปกรณ์สร้างสัญญาณรบกวน ที่สามารถนำไปต่อเชิงพาณิชย์ได้ ทีมวิจัยได้ศึกษามาตรฐาน 3GPP TS 38.212 และ 3GPP TS 38.211 เพื่ออ้างอิงการออกแบบอุปกรณ์เช่นเดียวกับการออกแบบซอฟต์แวร์ ทีมวิจัยได้พัฒนาบนอุปกรณ์ประเภท FPGA โดยทีมวิจัยได้เลือกใช้ภาษา VHDL โดยอุปกรณ์ประเภท FPGA นี้สามารถเชื่อมต่อกับซอฟต์แวร์ประเภท GUI เพื่อแสดงผลลัพธ์การทำงานของอุปกรณ์ประเภท FPGA ได้ เพื่อให้บริษัทและนักเรียนได้เห็นว่าอุปกรณ์สามารถทำงานได้จริง นอกจากนี้ ทีมวิจัยได้เผยแพร่การพัฒนาอุปกรณ์ประเภท FPGA ให้บริษัทไทยจำนวน 2 ราย เพื่อได้รับทราบถึงโอกาสและแนวทางการพัฒนาอุปกรณ์โทรคมนาคมต้นแบบ ทีมวิจัยคาดหวังว่าต้นแบบวงจรเข้ารหัสมาตรฐาน 3GPP TS 38.212 และ 3GPP TS 38.211 มาพัฒนาบนอุปกรณ์ประเภท FPGA นี้จะถูกนำไปพัฒนาต่อยอดและนำไปสู่การสร้างอุตสาหกรรมใหม่ของประเทศไทย

บรรณานุกรม

[1] P. L'Ecuyer, "Maximally Equidistributed Combined Tausworthe Generators," *Mathematics of Computation,* vol. 65, no. 213, pp. 203-213, 1996.

[2] ETSI, "LTE; evolved universal terrestrial radio access (E-UTRA);," European Telecommunications Standards Inst., 2014.

[3] A. G. a. P. T. C. Berrou, "Near Shannon limit error-correcting coding and decoding : turbo-codes," in *In International Conference on*, Geneva, 1993.

[4] R. G. Gallager, Low-Density Parity-Check Codes, USA: MIT Press, 1963.

[5] IEEE, "IEEE Standard for Information technology Telecommunications and information exchange between systems Local and metropolitan area networks Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amend," IEEE Std 802.11n-2009, 2009.

[6] ETSI, "Digital Video Broadcasting (DVB), Second Generation Framing Structure, Channel Coding and Modulation Systems for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Application," 2004.

[7] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory,* vol. 55, pp. 3051-3073, 2009.

[8] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans on Inform Theory,* Vols. IT-27, no. 5, pp. 533-547, 1981.

[9] D. D. a. R. W. T.-Y. Chen, "Protograph-based Raptor-like LDPC codes with low thresholds," in *Proc. IEEE ICC*, 2012.

[10] 3rd Generation Partnership Project (3GPP), "Multiplexing and channel coding," 3GPP TS38.212 V.15.2.0, 2018.

[11] AMD, "Soft-Decision QAM Demodulator Design on AI Engine (XAPP1388)," 20 04 2023. [Online]. Available: https://docs.amd.com/r/en-US/xapp1388-sd-qam-demod.

[12] O.-R. Alliance, O-RAN Alliance, “O-RAN Acceleration Abstraction Layer General Aspects and Principles,” O-RAN.WG6.AAL-GAnP.0-v04.00, 2022., 2022.

[13] 3GPP, 3rd Generation Partnership Project (3GPP), "Physical layer procedures for data, " 3GPP TS 38.214 V16.2.0, 2020, 2020.

[14] S. a. P. S. a. J. B. a. P. I.-C. Lee, "Multi-Mode QC-LDPC Decoding Architecture With Novel Memory Access Scheduling for 5G New-Radio Standard," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 69, no. 5, pp. 2032-2048, 2022.

[15] 3rd Generation Partnership Project (3GPP), "Physical channel and modulation," 3GPP TS38.211 V.17.4.0, 2022.

[1] P. L'Ecuyer, "Maximally Equidistributed Combined Tausworthe Generators," *Mathematics of Computation,* vol. 65, no. 213, pp. 203-213, 1996.

[2] ETSI, "LTE; evolved universal terrestrial radio access (E-UTRA);," European Telecommunications Standards Inst., 2014.

[3] A. G. a. P. T. C. Berrou, "Near Shannon limit error-correcting coding and decoding : turbo-codes," in *In International Conference on*, Geneva, 1993.

[4] R. G. Gallager, Low-Density Parity-Check Codes, USA: MIT Press, 1963.

[5] IEEE, "IEEE Standard for Information technology Telecommunications and information exchange between systems Local and metropolitan area networks Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amend," IEEE Std 802.11n-2009, 2009.

[6] ETSI, "Digital Video Broadcasting (DVB), Second Generation Framing Structure, Channel Coding and Modulation Systems for Broadcasting, Interactive Services, News Gathering and Other Broadband Satellite Application," 2004.

[7] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *IEEE Trans. Inf. Theory,* vol. 55, pp. 3051-3073, 2009.

[8] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans on Inform Theory,* Vols. IT-27, no. 5, pp. 533-547, 1981.

[9] D. D. a. R. W. T.-Y. Chen, "Protograph-based Raptor-like LDPC codes with low thresholds," in *Proc. IEEE ICC*, 2012.

[10] 3rd Generation Partnership Project (3GPP), "Multiplexing and channel coding," 3GPP TS38.212 V.15.2.0, 2018.

[11] AMD, "Soft-Decision QAM Demodulator Design on AI Engine (XAPP1388)," 20 04 2023. [Online]. Available: https://docs.amd.com/r/en-US/xapp1388-sd-qam-demod.

[12] O.-R. Alliance, O-RAN Alliance, “O-RAN Acceleration Abstraction Layer General Aspects and Principles,” O-RAN.WG6.AAL-GAnP.0-v04.00, 2022., 2022.

[13] 3GPP, 3rd Generation Partnership Project (3GPP), "Physical layer procedures for data, " 3GPP TS 38.214 V16.2.0, 2020, 2020.

[14] S. a. P. S. a. J. B. a. P. I.-C. Lee, "Multi-Mode QC-LDPC Decoding Architecture With Novel Memory Access Scheduling for 5G New-Radio Standard," *IEEE Transactions on Circuits and Systems I: Regular Papers,* vol. 69, no. 5, pp. 2032-2048, 2022.

[15] 3rd Generation Partnership Project (3GPP), "Physical channel and modulation," 3GPP TS38.211 V.17.4.0, 2022.

[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 โควินท์ทวีวัฒน์. “การสื่อสารดิจิทัล การเข้ารหัสช่องสัญญาณ.” http://home.npru.ac.th/ piya/DigitalComm/file/Lec1415.pdf.

[6] กานต์ ศรีรัชตบูรณ์. “การออกแบบเมทริกซ์พาริตีเช็กของรหัสแอลดีพีซี” วิทยานิพนธ์ปริญญาวิศวกรรมศาสตร์มหาบัณฑิต, สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์, จุฬาลงกรณ์มหาวิทยาลัย, 2557.

[7] Getsthiew. “ปฏิวัติการสื่อสารไทยด้วยเทคโนโลยี HSPA” http://tooktik-tt.blogspot.com/2008/08 /hspa-high-speed-packet-access_12.html.

[8] คณะเจ้าหน้าที่บริษัท อสมท จำกัด. “ระบบสื่อสารดิจิตอลสำหรับงาน Broadcasting” http://dtv.mcot .net/techno_one.php?dateone=1241680100.

[9] ผศ.ดร.เวธิต ภาคย์พิสุทธิ์, รศ.ดร.ลัญฉกร วุฒิสิทธิกุลกิจ, ดร.พิสิฐ วินิชชานันท์, ศ.ดร.ปิยะ โควินท์ 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

ภาคผนวก ก
เอกสารประกอบการเผยแพร่ในมหาวิทยาลัย ภาคทฤษฎี

เอกสารดังต่อไปนี้คือเอกสารที่ใช้ในการเผยแพร่ชุดการเรียนรู้และการถ่ายทอดความรู้พื้นฐาน ในส่วนการให้ความรู้พื้นฐานที่เกี่ยวข้องกับระบบสื่อสารในมาตรฐาน 5G

5G System Overview

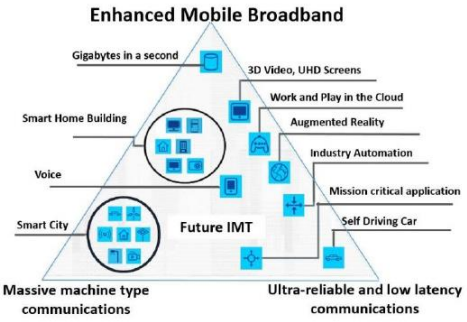
กฤติยาภรณ์ เหมือดขุนทด จตุพร ด้วงทอง ธนัช ศรีสุภา อนุสรณ์ วงค์ษา
รศ.ดร.กฤษณะพงศ์ พันธ์ศรี และ รศ.ดร.เวธิต ภาคย์พิสุทธิ์

IMT-2020

IMT-2020 คือวิสัยทัศน์และมาตรฐานการสื่อสารไร้สายในยุค ค.ศ.2020 กำหนดโดยสหภาพโทรคมนาคมระหว่างประเทศ (international telecommunication union: ITU) ภายใต้หน่วยงานที่เกี่ยวกับการสื่อสารวิทยุ (ITU radiocommunication sector: ITU-R)

วิสัยทัศน์ของ ITU-R ที่มีต่อการใช้งานและขีดความสามารถของการสื่อสารไร้สายได้ถูกเผยออกมาในเอกสาร **IMT Vision** ช่วงปี ค.ศ. 2015 และมีการคัดเลือกเทคโนโลยีที่ใช้กำหนดรายละเอียดมาตรฐาน IMT-2020 ในเอกสาร **IMT-2020 specifications** ช่วงปี ค.ศ. 2020

วิสัยทัศน์ IMT-2020 ถูกคิดค้นเพื่อรองรับการใช้งานเทคโนโลยีสื่อสารไร้สาย 3 ประเภท ดังรูปที่ 1



รูปที่ 1 รูปแบบการใช้งานจากกรณีการใช้งานทั้ง 3 ประเภท

1) enhanced Mobile BroadBand (eMBB)

เน้นการใช้งานกับข้อมูลความเร็วสูง ข้อมูลขนาดใหญ่ เช่น การสตรีมมิงวิดีโอ 4K หรือ VR/AR (https://youtu.be/v4VqgWSTH08)

2) massive Machine-Type Communications (mMTC)

เน้นการใช้งานกับอุปกรณ์จำนวนมาก เช่น อุปกรณ์ IoT ในบ้านหรือเมืองอัจฉริยะ (https://youtu.be/lBYEFd5J4Z4)+96

3) Ultra-Reliable and Low Latency Communications (URLLC)

เน้นการใช้งานกับงานที่ต้องการความน่าเชื่อถือสูง เช่น ทางการแพทย์ หรือระบบอัตโนมัติในโรงงานอุตสาหกรรมหรือบนท้องถนน (https://youtu.be/-1JGktixWBo)

เพื่อให้สอดคล้องกับการใช้งานทั้ง 3 ประเภท IMT-2020 จึงได้มีการกำหนด 8 ขีดความสามารถ ที่เพิ่มขึ้นจากการสื่อสารยุคก่อนหน้ามีดังนี้

1) ความเร็วสูงสุด (peak data rate) ที่ 20 Gbit/s

2) ความเร็วประสบการณ์ผู้ใช้งาน (user experienced data rate) ที่ 100 Mbit/s

3) ประสิทธิภาพสเปกตรัม (spectrum efficiency) ที่ 3 เท่า จากยุคก่อน

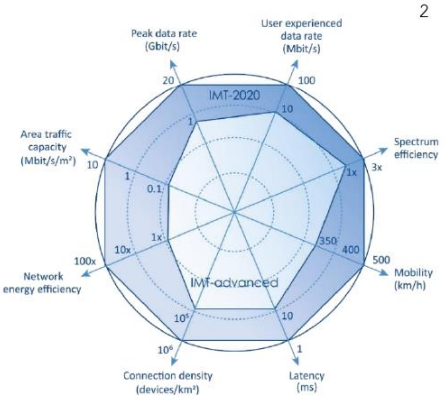
4) ความคล่องตัว (mobility) ที่ 500 km/h

5) ความหน่วง (latency) ที่ 1 ms

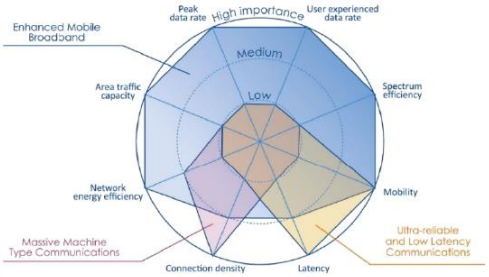
6) ความหนาแน่นการเชื่อมต่อ (connection density) ที่ 106 devices/km^2

7) ประสิทธิภาพพลังงานโครงข่าย (network energy efficiency) ที่ 100 เท่า จากยุคก่อน

8) ความจุพื้นที่การจลาจร (area traffic capacity) ที่ 10 Mbit/s/m^2



รูปที่ 2 เปรียบเทียบขีดความสามารถของ IMT-Advanced (4G) และ IMT-2020 (5G) ตามวิสัยทัศน์ ITU-R M.2083



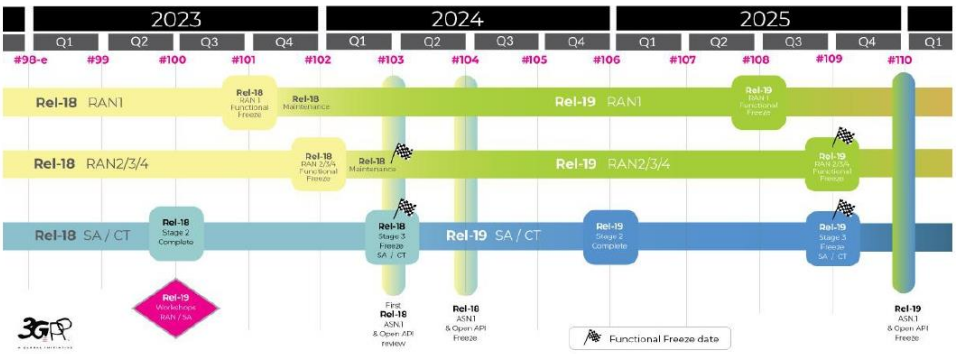
รูปที่ 3 เปรียบเทียบขีดความสามารถของการใช้งาน eMBB mMTC และ URLLC

5G NR Standard (3GPP)

5G New Radio (NR) คือมาตรฐานการสื่อสารไร้สายที่ได้รับคัดเลือกให้เป็นเทคโนโลยี IMT-2020 สำหรับโทรศัพท์มือถือ 5G NR เป็นมาตรฐานเชิงเทคนิคที่รวบรวมวิธีการทางวิศวกรรมต่าง ๆ เพื่อให้ได้การสื่อสารไร้สายที่สอดคล้องกับความต้องการของ IMT-2020 โดย 5G NR ถูกกำหนดโดยองค์กร 3rd Generation Partnership Project (3GPP) ที่รวบรวมบริษัทโทรคมนาคมและบริษัทเทคโนโลยีที่เกี่ยวข้องจากทั่วโลก รวมทั้งหน่วยงานภาครัฐจากประเทศต่าง ๆ

ความสามารถของการสื่อสารยุค 5G ที่แตกต่างจาก 4G คือการสื่อสารที่รวดเร็วขึ้น รองรับปริมาณที่มากขึ้น มีเสถียรภาพและความปลอดภัย เป้าหมายของการสื่อสารยุค 5G จึงสอดคล้องกับการใช้งานทั้ง 3 รูปแบบทั้ง eMBB mMTC และ URLLC ตามวิสัยทัศน์ของ IMT-2020

สำหรับองค์กร 3GPP การออกมาตรฐานเชิงเทคนิคของมาตรฐานการสื่อสารไร้สาย จะแบ่งช่วงเป็น release โดยมาตรฐาน 5G NR ดังกล่าวจะถูกจัดอยู่ใน release 15 ซึ่ง ณ ปัจจุบันมีแผนการออก release ใหม่ดังนี้

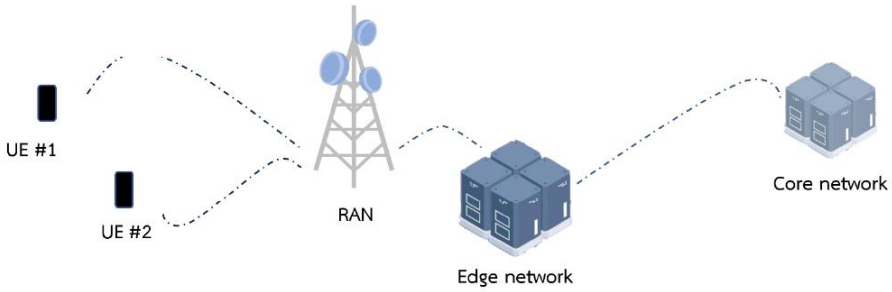


รูปที่ 4 แผนการออกมาตรฐานขององค์กร 3GPP

5G NR System Architecture

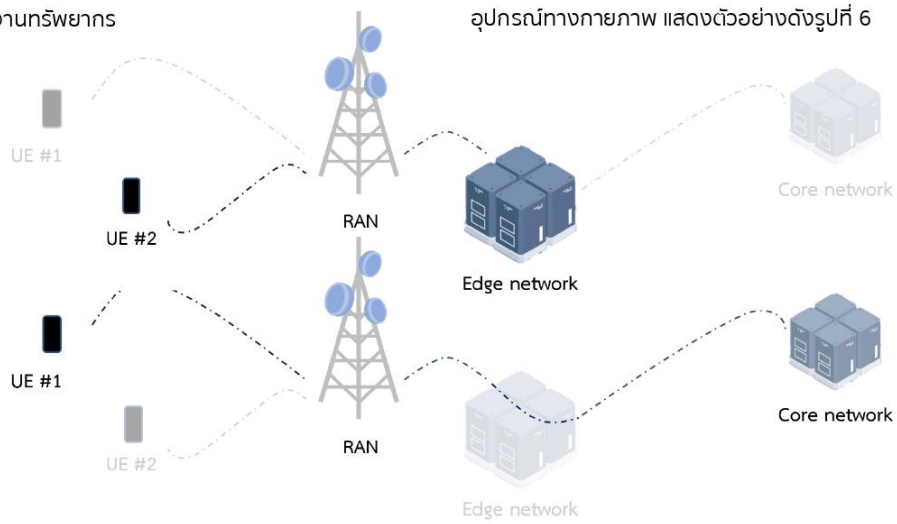
สถาปัตยกรรมของระบบสื่อสารไร้สายยุคที่ 5 มีส่วนประกอบ 3 ส่วน คือ

1) โครงข่ายหลัก	2) โครงข่ายการเข้าถึงวิทยุ	3) อุปกรณ์ผู้ใช้งาน
(core network) หรือ CN	(radio access network) หรือ RAN	(user equipment) หรือ UE



รูปที่ 5 อุปกรณ์ในโครงสร้างพื้นฐานทางกายภาพ

โครงสร้างพื้นฐานทางกายภาพ (physical infrastructure) จะประกอบไปด้วยอุปกรณ์ทั้ง 3 ส่วน แสดงดังรูปที่ 5 แต่ทางปฏิบัติสามารถพิจารณาโครงสร้างพื้นฐานทางลอจิก (logical infrastructure) เพื่อความยืดหยุ่นและเพิ่มประสิทธิภาพการใช้งานทรัพยากร แนวคิดโครงสร้างพื้นฐานทางลอจิก จะมองการสื่อสาร 1 ช่องสัญญาณเป็นสไลด์ (network slice) อุปกรณ์ในโครงสร้างพื้นฐานทางกายภาพสามารถรับรองการสื่อสารได้มากกว่า 1 ช่องสัญญาณหรือหลายสไลด์ โดยที่ 1 สไลด์ไม่จำเป็นต้องใช้งานทุกอุปกรณ์ทางกายภาพ แสดงตัวอย่างดังรูปที่ 6



รูปที่ 6 การสื่อสารของ UE โดยใช้แนวคิดโครงสร้างพื้นฐานทางลอจิก

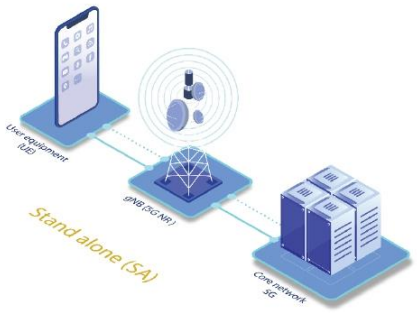
5G NR and LTE Internetworking

อุปกรณ์ในมาตรฐาน 5G นอกจากจะเชื่อมต่อกับ CN ของมาตรฐาน 5G แล้ว ยังต้องสามารถเชื่อมต่อกับ CN ของมาตรฐาน 4G ด้วย ซึ่งรู้จักในนามของสถาปัตยกรรมร่วม

สถาปัตยกรรมร่วม (non-standalone: NSA) จะเกิดขึ้นในระยะแรกของการเปลี่ยนผ่านจากเทคโนโลยี 4G ไปเป็น 5G โดยนำเอา RAN และ UE ของมาตรฐาน 5G ไปใช้ร่วมกับ CN และ RAN ของมาตรฐาน 4G

ขณะที่**สถาปัตยกรรมเดี่ยว (standalone: SA)** จะใช้งาน CN RAN และ UE ตามมาตรฐาน 5G ล่าสุดทั้งหมดเพื่อให้ได้ความสามารถตามขีดจำกัดของวิสัยทัศน์ IMT-2020

ก) สถาปัตยกรรมร่วม



ข) สถาปัตยกรรมเดี่ยว

รูปที่ 7 การเชื่อมต่อระหว่างอุปกรณ์ 5G NR และ LTE

5G Physical-layer Processing

กฤติยาภรณ์ เหมือดขุนทด จตุพร ด้วงทอง ธนัช ศรีสุภา อนุสรณ์ วงค์ษา
รศ.ดร.กฤษณะพงศ์ พันธ์ศรี และ รศ.ดร.เวธิต ภาคย์พิสุทธิ์

5G NR Radio Protocol Stack

ระบบการสื่อสารทุกระบบมักมีแบบจำลองการสื่อสารที่บ่งบอกถึงกระบวนการทำงานของระบบ ต้นแบบของแบบจำลองการสื่อสารอาจทราบกันในชื่อ open systems interconnection หรือ OSI model ซึ่งอธิบายถึงกระบวนการสื่อสารของระบบ OSI model ประกอบด้วย 7 ชั้นอย่างนามธรรม

แต่ระบบการสื่อสารที่ถูกใช้งานจริง ไม่จำเป็นต้องจำแนกชั้นตามระบบตาม OSI model เช่น ชุดโพรโทคอลอินเทอร์เน็ตหรือ TCP/IP ที่นิยมใช้ในระบบคอมพิวเตอร์จะประกอบไปด้วย 5 ชั้น

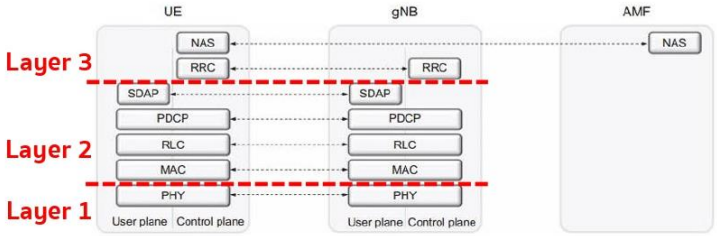
อย่างไรก็ตามแบบจำลองทั้ง 2 มีความคล้ายคลึงกันและอาจเชื่อมโยงโดยคร่าวได้ โดยแต่ละชั้นของแบบจำลองการสื่อสารก็จะมีการไหลผ่านของข้อมูลตามมาตรฐานของระบบการสื่อสารนั้น ๆ เพื่อให้เกิดการสื่อสารอย่างสมบูรณ์และน่าเชื่อถือ

7) application	
6) presentation	
5) session	5) application
4) transport	4) transport
3) network	3) network
2) data link	2) data link
1) physical	1) physical
OSI model	**TCP/IP protocol stack**

สำหรับมาตรฐาน 5G แบบจำลองการสื่อสารแต่ละชั้นของ UE และ RAN (gNB) จะถูกพูดถึงเพียงชั้นที่ 1 ถึงชั้นที่ 3 และจำแนกรูปแบบข้อมูลการสื่อสาร 2 ประเภท

1) user plane สำหรับรับส่งข้อมูลของผู้ใช้งาน เช่น แชทข้อความ ไลฟ์วิดีโอ หรือไฟล์เอกสาร ของผู้ใช้งาน

2) control plane สำหรับรับส่งข้อมูลที่ใช้ในการควบคุมและจำเป็นต่อการสื่อสาร ที่ไม่เกี่ยวข้องกับผู้ใช้งาน เช่น สภาพช่องสัญญาณการสื่อสาร การมอดูเลต อัตรารหัส เวลา และความถี่ในการสื่อสาร



รูปที่ 8 แบบจำลองการสื่อสารในมาตรฐาน 5G

1) non-access-stratum (NAS) ติดต่อสื่อสารระหว่าง UE และฟังก์ชัน AMF ใน core network มีหน้าที่ รวมทั้งการรับรองความถูกต้อง (authentication) ความปลอดภัย (security) การดำเนินการในโหมด idle รูปแบบต่าง ๆ และยังทำหน้าที่กำหนดหมายเลข IP ให้แก่ UE

2) radio resource control (RRC) รับส่งข้อมูลที่ใช้ในการควบคุม โดยข้อความ RRC จะถูกส่งให้ UE โดยใช้ signaling radio bearers

3) service data application protocol (SDAP) รับผิดชอบการจัดการ QoS ไปยัง radio bearer โพรโทคอลชั้นดังกล่าวไม่มีใน LTE

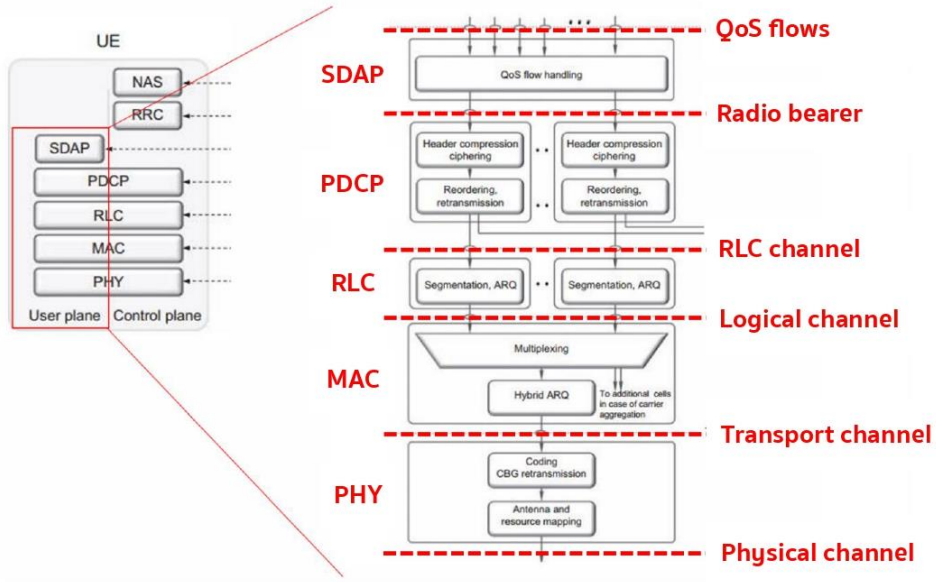
4) packet data convergence protocol (PDCP) ทำหน้าที่บีบอัด IP header ciphering และ integrity protection และยังทำหน้าที่ส่งซ้ำ (retransmission) การจัดส่งตามลำดับ (in-sequence delivery) และกำจัดการเชื่อมต่อซ้ำในกรณี handover

5) radio-link control (RLC) ทำหน้าที่ แบ่งส่วนข้อมูล (segmentation) และจัดการการส่งซ้ำ (retransmission)

6) medium-access control (MAC) ทำหน้าที่ มัลติเพล็กซ์ช่องสัญญาณลอจิก (logical channel) การส่งซ้ำ (retransmission) แบบ HARQ และจัดการฟังก์ชันการ schedule

7) physical layer (PHY) ทำหน้าที่ เข้าและถอดรหัส มอดูเลต/ดีมอดูเลต การเชื่อมโยงสายอากาศหลายสาย และฟังก์ชันที่เกี่ยวข้องกับชั้นกายภาพ (physical layer) ทั่วไป

โดยหากขยายภาพแบบจำลองการสื่อสาร จากรูปที่ 8 จะพบเห็นรายละเอียดกระบวนการทำงานของแบบจำลองการสื่อสาร รวมถึงช่องสัญญาณประเภทต่าง ๆ ที่ขั้นระหว่างกระบวนการดังรูปที่ 9



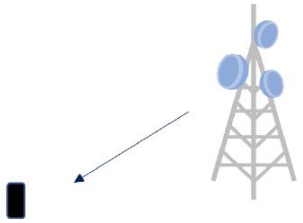
รูปที่ 9 รายละเอียดกระบวนการทำงานของแบบจำลองการสื่อสารของ user plane

Physical/Transport/Logical Channel

อุปกรณ์ RAN และ UE จะมีช่องสัญญาณการสื่อสารมากมาย เงื่อนไขการแบ่งประเภทช่องสัญญาณอาจเป็นได้ทั้ง**ทิศทางการสื่อสาร**หรือ**ประเภทของข้อมูล**

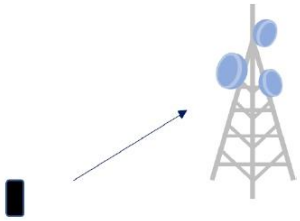
ทิศทางการสื่อสารสามารถแบ่งได้เป็น 3 ช่องสัญญาณ ดังรูปที่ 10 11 และ 12

ช่องสัญญาณขาลง (downlink)
เป็นช่องสัญญาณการสื่อสารจาก RAN ไปยัง UE



รูปที่ 10 ช่องสัญญาณขาลง

ช่องสัญญาณขาขึ้น (uplink)
เป็นช่องสัญญาณการสื่อสารจาก UE ไปยัง RAN



รูปที่ 11 ช่องสัญญาณขาขึ้น

ช่องสัญญาณข้างเคียง (sidelink)
เป็นช่องสัญญาณการสื่อสารระหว่าง UE และ UE

รูปที่ 12 ช่องสัญญาณข้างเคียง

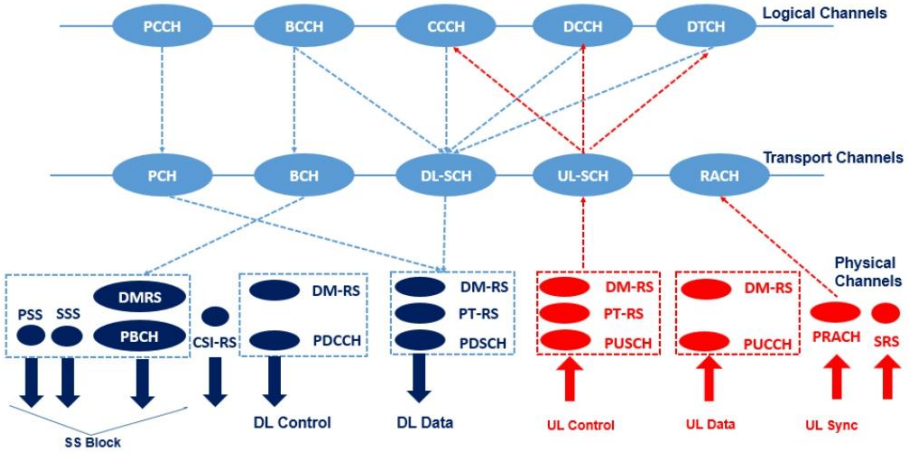
นอกจากแบ่งช่องสัญญาณตามทิศทางการสื่อสารแล้ว ยังสามารถแบ่งได้ตามลักษณะอื่นได้ โดย**ช่องสัญญาณกายภาพ (physical channel)** ดังต่อไปนี้

ช่องสัญญาณกายภาพ

ช่องสัญญาณขาขึ้น (uplink)
physical uplink shared channel (PUSCH)
physical uplink control channel (PUCCH)
physical random access channel (PRACH)

ช่องสัญญาณขาลง (downlink)
physical downlink shared channel (PDSCH)
physical downlink control channel (PDCCH)
physical broadcast channel (PBCH)

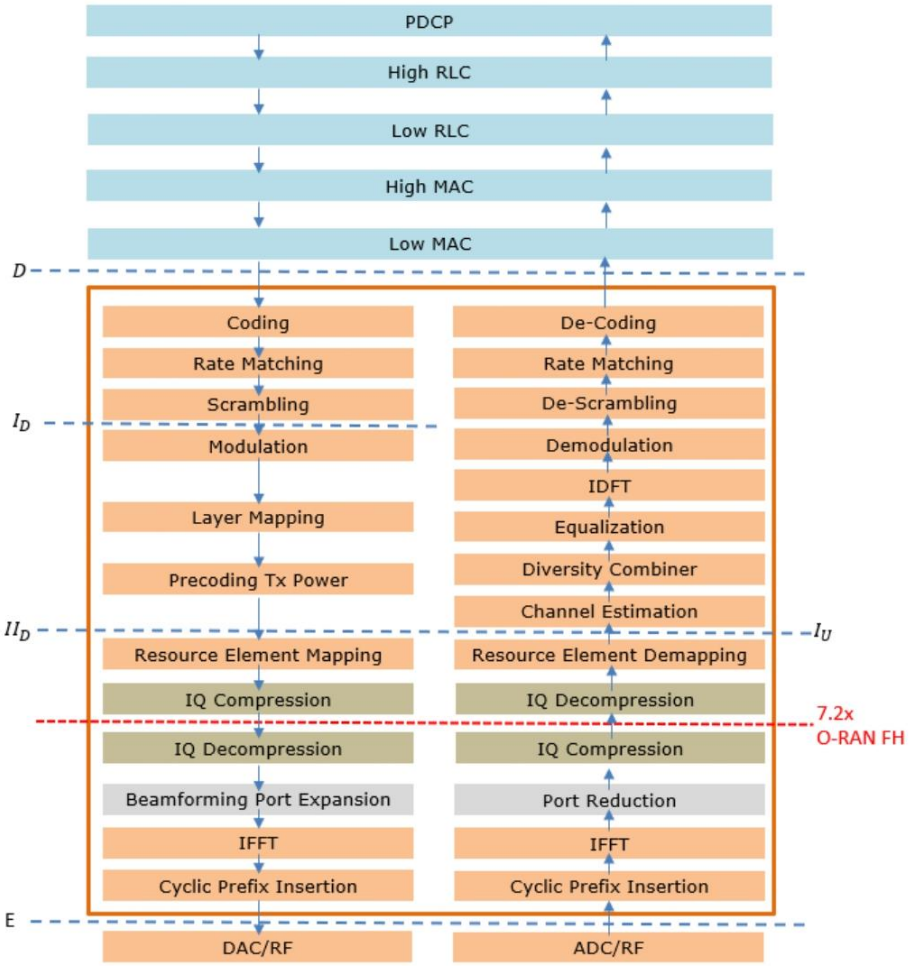
ช่องสัญญาณข้างเคียง (sidelink)
physical sidelink shared channel (PSSCH)
physical sidelink control channel (PSCCH)
physical sidelink broadcast channel (PSBCH)
physical sidelink feedback channel (PSFCH)



รูปที่ 13 ช่องสัญญาณกายภาพเชื่อมต่อกับช่องสัญญาณอื่น

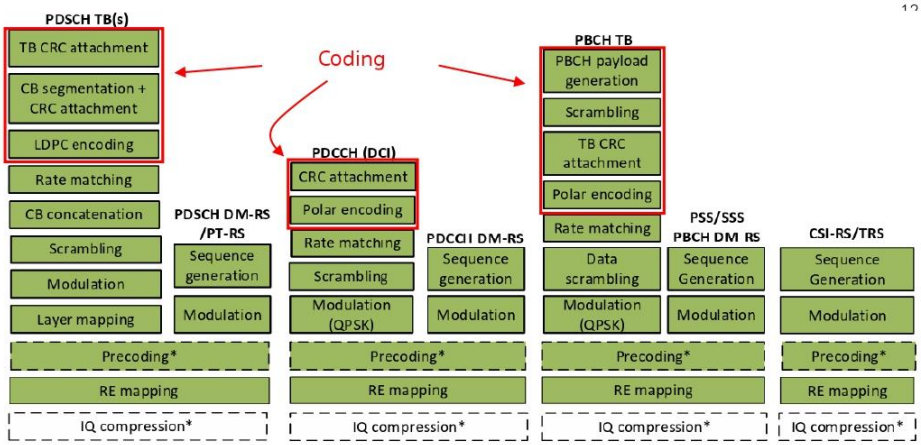
รูปที่ 13 ได้แสดงถึงการโยง (mapping) ช่องสัญญาณระหว่างช่องสัญญาณกายภาพไปยังช่องสัญญาณทรานสปอร์ต (transport channel) รวมถึงช่องสัญญาณลอจิก (logical channel) ช่องสัญญาณเหล่านี้จะอยู่ระหว่างชั้นแบบจำลองการสื่อสารของมาตรฐานการสื่อสารไร้สาย 5G

5G Physical-layer Processing

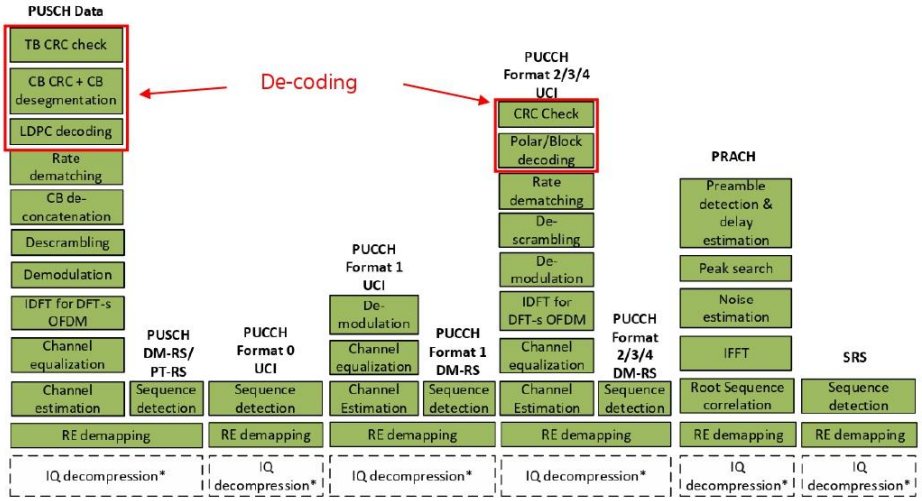


รูปที่ 14 กระบวนการทำงานโดยคร่าวภายในแบบจำลองชั้นกายภาพ (physical layer) ของมาตรฐาน 5G

รูปที่ 14 แสดงถึงกระบวนการทำงานโดยคร่าวภายในแบบจำลองชั้นกายภาพ ที่เชื่อมต่อกับสัญญาณอนาล็อกสู่สายอากาศด้านล่างและแบบจำลองชั้น MAC ด้านบน โดยโดยจะพบกระบวนการ coding rate matching scrambling และ modulation ซึ่งอาจเรียกว่าชั้นกายภาพส่วนบน (high-PHY) ชั้นกายภาพส่วนบนมีรายละเอียดและกระบวนการทำงานของแต่ละช่องสัญญาณกายภาพ ได้แก่ PDSCH PDCCH PBCH PUSCH PUCCH และ PRACH ดังรูปต่อไปนี้



รูปที่ 15 กระบวนการทำงานภายในแบบจำลองชั้นกายภาพของที่เกี่ยวข้องกับช่องสัญญาณขาลง เมื่อพิจารณาที่อุปกรณ์ RAN



รูปที่ 16 กระบวนการทำงานภายในแบบจำลองชั้นกายภาพของที่เกี่ยวข้องกับช่องสัญญาณขาขึ้น เมื่อพิจารณาที่อุปกรณ์ RAN

5G Channel Coding

การสื่อสารดิจิตอลโดยทั่วไปล้วนต้องการสื่อสารอย่างมีประสิทธิภาพหรือปราศจากข้อผิดพลาด แต่อย่างไรก็ตาม เนื่องจากในสภาพอากาศมีสิ่งรบกวนจำนวนมาก เช่น การรบกวนจากฝน สัญญาณรบกวนแบบจางหาย รวมถึงความร้อนจากวงจรอิเล็กทรอนิกส์ ดังนั้นการที่จะทำให้การสื่อสารของเราปราศจากข้อผิดพลาดเลยจึงเป็นไปได้ยาก

รหัสช่องสัญญาณ (channel coding)

หรือรหัสแก้ไขความผิดพลาด เป็นองค์ประกอบสำคัญในระบบสื่อสารดิจิทัลที่ขาดไม่ได้ เนื่องจากรหัสช่องสัญญาณทำให้การรับส่งข้อมูลดิจิทัลปราศจากบิตผิด รหัสช่องสัญญาณจะทำการเพิ่มบิตตรวจสอบ (parity bits) ไปกับข้อมูลผู้ใช้งานเพื่อช่วยแก้ไขความผิดพลาดของข้อมูล

รหัสช่องสัญญาณในมาตรฐาน 5G

1) รหัสพาริตีเช็คความหนาแน่นต่ำ (low-density parity-check code: LDPC)

2) รหัสโพลาร์ (polar code)

ถูกใช้งานในประเภทของช่องสัญญาณตามลักษณะข่าวสารและทิศทางการสื่อสารดังนี้

รูปที่ 17 ตัวอย่างการใช้งานและไม่ใช้งานรหัสช่องสัญญาณ

รหัสแอลดีพีซี	รหัสโพลาร์
ช่องสัญญาณขาขึ้น (uplink)	
physical uplink shared channel (PUSCH)	physical uplink control channel (PUCCH)
ช่องสัญญาณขาลง (downlink)	
physical downlink shared channel (PDSCH)	physical downlink control channel (PDCCH) physical broadcast channel (PBCH)
ช่องสัญญาณข้างเคียง (sidelink)	
physical sidelink shared channel (PSSCH)	physical sidelink control channel (PSCCH) physical sidelink broadcast channel (PSBCH) physical sidelink feedback channel (PSFCH)

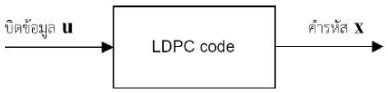
LDPC Code

รหัสบล็อกเชิงเส้น
คือรหัสที่ทำการเข้าหรือถอดรหัสหลายอักษร (alphabet) ต่อเวลา

อัตราส่วนของบิตข้อมูลกับจำนวนบิตทั้งหมดในคำรหัสนั้นเรียกว่า **อัตรารหัส** (Code Rate: R) นิยามได้ดังสมการที่ 1

$$R = \frac{K}{N} \quad (1)$$

รหัสแอลดีพีซี (low-density parity-check: LDPC) จะถูกกำหนดโดยเมทริกซ์พาริตีเช็ค (parity check matrix: $\mathbf{H}$) ซึ่งสัมพันธ์กับเมทริกซ์ตัวกำเนิด (generator matrix: $\mathbf{G}$) มีหน้าที่ในการสร้างคำรหัสจากบิตข้อมูลและเมทริกซ์ $\mathbf{H}$ มีหน้าที่ในการถอดรหัสจากบิตรหัสที่ได้รับจากช่องสัญญาณ



รูปที่ 18 แผนภาพบล็อกการเข้ารหัสแอลดีพีซี

เมทริกซ์กำเนิด
เมทริกซ์ $\mathbf{G}$ จะเข้ารหัสบิตข้อมูล ซึ่งกระบวนการเข้ารหัสทำได้โดยการนำบิตข้อมูล $\mathbf{u}$ มาคูณกับเมทริกซ์ $\mathbf{G}$ ดังสมการที่ 2 และตัวอย่างดังสมการที่ 3

$$\mathbf{x} = \mathbf{uG} \quad (2)$$

$$\mathbf{x} = \mathbf{uG} = \begin{bmatrix} 1 & 0 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 1 & 0 \end{bmatrix} \quad (3)$$

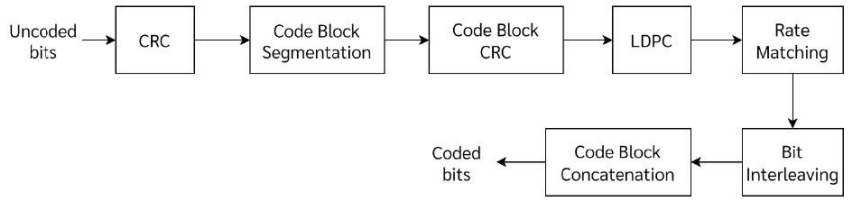
เมทริกซ์ตรวจสอบพาริตี
จากสมการที่ 4 จะเห็นว่าทั้งเมทริกซ์ $\mathbf{H}$ และ $\mathbf{G}$ เมื่อนำมาคูณกันแล้วจะได้เวกเตอร์ 0 ดังนั้นเมทริกซ์ $\mathbf{H}$ มีความสัมพันธ์กับคำรหัสดังสมการที่ 5

$$\mathbf{HG}^T = \mathbf{0} \quad (4)$$

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

5G PDSCH/PUSCH/PSSCH Encoding

รหัสแอลดีพีซีเป็นรหัส่องสัญญาณที่ถูกใช้งานกับข้อมูลบน user plane กระบวนการเข้ารหัสแอลดีพีซีตามมาตรฐาน 5G สำหรับช่องสัญญาณ PDSCH PUSCH และ PSSCH ประกอบด้วย 7 กระบวนการ ดังรูปที่ 19



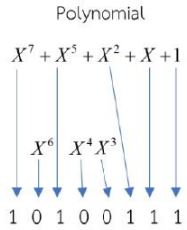
รูปที่ 19 แผนภาพการเข้ารหัสแอลดีพีซีช่องสัญญาณ PDSCH PUSCH และ PSSCH มาตรฐาน 5G

1) การตรวจสอบด้วยส่วนซ้ำซ้อนแบบวน
(cyclic redundancy check: CRC)

การเข้ารหัส CRC เป็นกระบวนการตรวจสอบความผิดพลาดของข้อมูล หลักการของ CRC จะใช้โพลีโนเมียล (polynomial) ใช้หารกับบิตข้อมูล เศษจากการหารจะถูกนำไปต่อท้ายบิตข้อมูล

การถอดรหัส CRC บิตคำรหัสจะถูกนำมาหารกับโพลีโนเมียลอีกครั้ง หากได้เศษเป็นศูนย์ จะถือว่าไม่มีข้อมูลผิดพลาด นอกจากนั้นถือว่าเกิดความผิดพลาด

สำหรับมาตารฐาน 5G ช่องสัญญาณ PDSCH PUSCH และ PSSCH จะใช้โพลีโนเมียลอยู่ 2 ดังนี้



รูปที่ 20 การหาบิตจากโพลีโนเมียลสำหรับการหาร

รูปที่ 21 บิตข้อมูลหลังการเข้ารหัส CRC

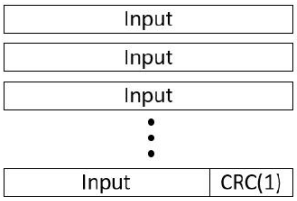
$$g_{CRC24A}(X) = X^{24} + X^{23} + X^{18} + X^{17} + X^{14} + X^{11} + X^{10} + X^7 + X^6 + X^5 + X^4 + X^3 + X + 1 \quad (10)$$

$$g_{CRC16}(X) = X^{16} + X^{12} + X^5 + 1 \quad (11)$$

โดยที่ $g_{CRC16}(X)$ ใช้กับข้อมูลที่มีความยาวน้อยกว่าหรือเท่ากับ 3824 บิต และ $g_{CRC24A}(X)$ สำหรับอื่น ๆ

2) Code Block Segmentation

เป็นการแบ่งส่วนข้อมูลเป็นส่วนย่อย เมื่อข้อมูลมีขนาดมากเกินไป ดังรูปที่ 22 สำหรับรหัสแอลดีพีซีกราฟฐาน 1 การแบ่งส่วนเมื่อความยาวข้อมูลมากกว่า 8448 บิต สำหรับรหัสแอลดีพีซีกราฟฐาน 2 การแบ่งส่วนเมื่อความยาวข้อมูลมากกว่า 3840 บิต



รูปที่ 22 โครงสร้างบล็อกอินพุตจากการแบ่งส่วนข้อมูล

3) Code Block CRC

การเข้ารหัส CRC ของบล็อกย่อย (Code Block CRC) เป็นกระบวนการเข้ารหัส CRC อีกครั้งหนึ่งด้วยโพลีโนเมียลดังสมการที่ 12 ให้กับแต่ละบล็อกหลังจากการแบ่งส่วนข้อมูลพร้อมเติมบิตว่าง (Null) ให้แต่ละบล็อกมีขนาดที่สามารถนำไปเข้ารหัสได้

$$g_{\mathrm{CRC24B}}(X) = X^{24} + X^{23} + X^{6} + X^{5} + X + 1 \quad (12)$$

Input		CRC(2)	Null
Input		CRC(2)	Null
Input		CRC(2)	Null
⋮			
Input	CRC(1)	CRC(2)	Null

รูปที่ 23 โครงสร้างแต่ละบล็อกของอินพุตจากการแบ่งส่วนข้อมูลและเข้ารหัส CRC

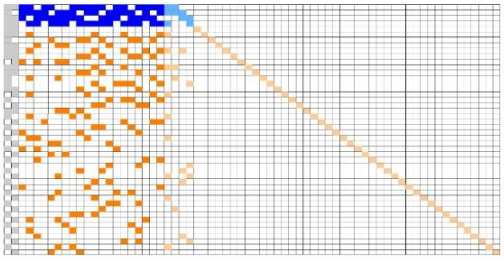
4) LDPC Encoder

เมทริกซ์ตรวจสอบพาริตีของการเข้ารหัสแอลดีพีซีตามมาตรฐาน 5G มีอยู่สองแบบคือกราฟฐาน 2 และกราฟฐาน 1 ดังรูปที่ 24 และรูปที่ 25 ตามลำดับ

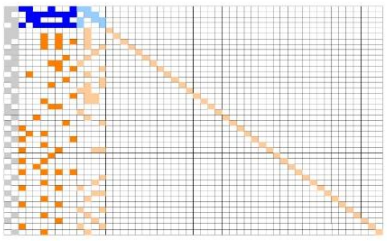
โดยใช้กราฟฐาน 2 ก็ต่อเมื่อ

$(A <= 292) \mid (A <= 3824 \ \& \ R <= 0.67) \mid (R <= 0.25)$

และใช้กราฟฐาน 1 ในกรณีอื่น โดย A คือความยาวข้อมูลและ R คืออัตรารหัส



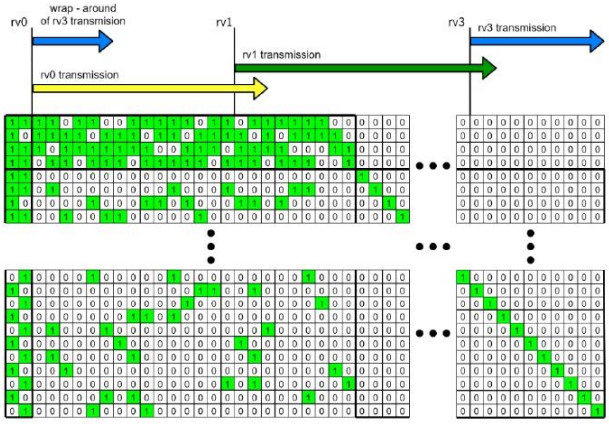
รูปที่ 24 กราฟฐาน 1



รูปที่ 25 กราฟฐาน 2

5) Rate Matching

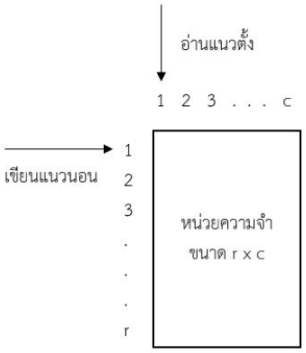
คือกระบวนการปรับขนาดคำรหัสให้มีขนาดเท่ากับขนาดบล็อกการขนส่ง การปรับอัตรารหัสสำหรับรหัส LDPC จะมีการกำหนดค่า redundancy version (rv) ซึ่งจะมีความสอดคล้องกับตำแหน่งคอลัมน์ของกราฟฐาน โดยกราฟฐานจะถูกแบ่งออกเป็น 4 ส่วนโดยไม่รวมบิต puncture ใน 2 คอลัมน์แรกดังรูปที่ 26



รูปที่ 26 การปรับอัตราสำหรับรหัส LDPC ตามมาตรฐาน 5G

6) Bit Interleaving

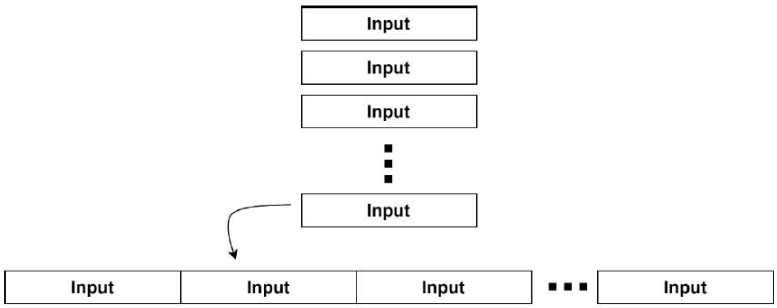
คือการป้องกันความผิดพลาดของข้อมูลที่ส่งไปบนช่องสัญญาณด้วยกระบวนการสลับบิตหรือไบต์ข้อมูลที่ภาคส่งดังรูปที่ 27 หากมีสัญญาณรบกวนเกิดขึ้นบนช่องสัญญาณเป็นผลให้ข้อมูลที่ส่งนั้นมีความผิดพลาดติดกัน เมื่อทำกระบวนการแทรกสลับบิตกลับ (De-Interleaving) จะทำให้บิตผิดพลาดที่ติดกันมีการกระจายตัวออก



รูปที่ 27 การแทรกสลับบิตสำหรับ LDPC ตามมาตรฐาน 5G

7) Code Block Concatenation

คือการต่อเรียงบล็อกรหัสเอาต์พุตที่ได้จากการแทรกสลับบิต

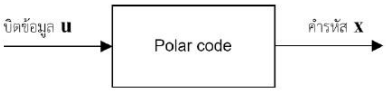


รูปที่ 28 การต่อเรียงบล็อกรหัสสำหรับ LDPC ตามมาตรฐาน 5G

Polar Code

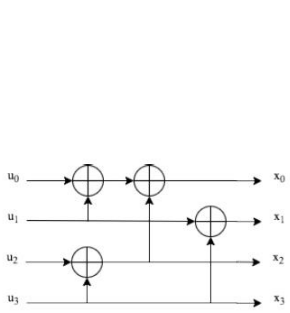
รหัสโพลาร์เป็นรหัสช่องสัญญาณประเภทบล็อก (block code) การเข้ารหัสโพลาร์สามารถดำเนินการทางคณิตศาสตร์ดังสมการที่ 13

$$\mathbf{x} = \mathbf{u}\mathbf{G}_n \qquad (13)$$

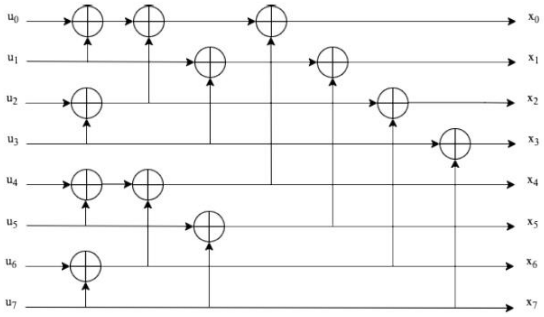


รูปที่ 29 แผนภาพบล็อกการเข้ารหัสโพลาร์

โดยที่ $\mathbf{x}$ คือคำรหัส $\mathbf{u}$ คือบิตแซ่แข็งและบิตข้อมูลที่เรียงตามลำดับความน่าเชื่อถือ $\mathbf{G}_n$ คือเมทริกซ์กำเนิด โดยที่ $\mathbf{G}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ และ $\mathbf{G}_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}$ และเวกเตอร์ $\mathbf{x}$ และ $\mathbf{u}$ มีความยาวได้เพียงสองยกกำลังใด ๆ เท่านั้น หรือ $N = 2^n$ K คือความยาวบิตข้อมูล ดังนั้น ความยาวบิตแซ่แข็งจะเท่ากับ $N-K$ การเข้ารหัสโพลาร์แสดงเป็น**โครงสร้างการทำเอ็กคลูซีฟออร์**และเทียบเป็น**การคูณทางคณิตศาสตร์**ดังต่อไปนี้



ก) รหัสโพลาร์ขนาด $N = 4$



ข) รหัสโพลาร์ขนาด $N = 8$

รูปที่ 30 โครงสร้างการเข้ารหัสโพลาร์ขนาดต่าง ๆ

$$\begin{bmatrix} x_0 x_1 x_2 x_3 \end{bmatrix} = \begin{bmatrix} u_0 u_1 u_2 u_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \qquad (15)$$

$$\begin{bmatrix} x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7 \end{bmatrix} = \begin{bmatrix} u_0 u_1 u_2 u_3 u_4 u_5 u_6 u_7 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \qquad (16)$$

การสร้างเมทริกซ์กำเนิดขนาดต่าง ๆ ทำได้ดังความสัมพันธ์ต่อไปนี้

$$[\mathbf{G}_N]=\begin{bmatrix}\mathbf{G}_{N/2} & \mathbf{0}\\ \mathbf{G}_{N/2} & \mathbf{G}_{N/2}\end{bmatrix} \tag{17}$$

ตัวอย่างการสร้างเมทริกซ์กำเนิดความยาว $N=4$ จะสามารถทำได้ดังต่อไปนี้

$$[\mathbf{G}_N]=\begin{bmatrix}\mathbf{G}_{N/2} & \mathbf{0}\\ \mathbf{G}_{N/2} & \mathbf{G}_{N/2}\end{bmatrix}$$

$$[\mathbf{G}_4]=\begin{bmatrix}\mathbf{G}_{4/2} & \mathbf{0}\\ \mathbf{G}_{4/2} & \mathbf{G}_{4/2}\end{bmatrix}=\begin{bmatrix}\mathbf{G}_2 & \mathbf{0}\\ \mathbf{G}_2 & \mathbf{G}_2\end{bmatrix}=\begin{bmatrix}1&0&0&0\\1&1&0&0\\1&0&1&0\\1&1&1&1\end{bmatrix}$$

ตัวอย่างการเข้ารหัสความยาว $N=8$ กำหนดบิตข้อมูลคือ 1010_2 ความยาว $K=4$ ที่อัตรารหัส $R=1/2$ และให้บิตข้อมูลอยู่ที่ตำแหน่ง $u_3u_5u_6u_7$ จะสามารถเข้ารหัสได้ดังสมการต่อไปนี้

$$\begin{aligned}[x_0x_1x_2x_3x_4x_5x_6x_7]&=[u_0u_1u_2u_3u_4u_5u_6u_7][\mathbf{G}_8]\\&=[00010010]\begin{bmatrix}1&0&0&0&0&0&0&0\\1&1&0&0&0&0&0&0\\1&0&1&0&0&0&0&0\\1&1&1&1&0&0&0&0\\1&0&0&0&1&0&0&0\\1&1&0&0&1&1&0&0\\1&0&1&0&1&0&1&0\\1&1&1&1&1&1&1&1\end{bmatrix}\\&=[01011010]\end{aligned}$$

โดยบิตแซ่แข็งความยาว $N-K=4$ จะถูกแทนด้วยบิต 0000_2 ที่ตำแหน่ง $u_0u_1u_2u_4$

รูปที่ 31 ตำแหน่งบิตแซ่แข็ง (สีฟ้า) และบิตข้อมูล (สีแดง) ในเวกเตอร์ $\mathbf{u}$

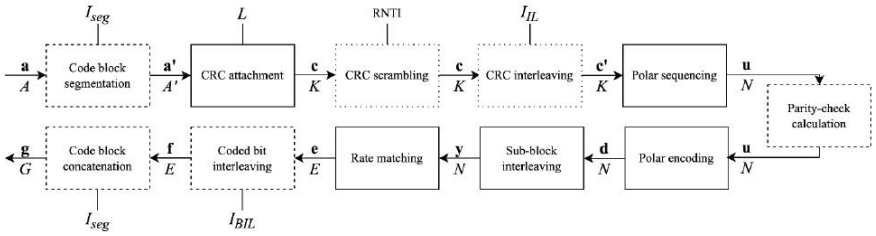
รูปที่ 31 แสดงถึงตัวอย่างการเรียงบิตแซ่แข็งและบิตข้อมูลในเวกเตอร์ $\mathbf{u}$ การเรียงดังกล่าวถือ**มีความสำคัญต่อสมรรถนะการแก้ไขความผิดพลาด**ของรหัสโพลาร์อย่างมาก รหัสโพลาร์จะเลือกวางบิตข้อมูลความยาว K บิต ในตำแหน่งที่มีความน่าเชื่อถือมากที่สุดและตำแหน่งที่เหลือจะถูกวางด้วยบิตแซ่แข็ง โดยที่บิตแซ่แข็งเป็นบิตที่ทราบในทางส่งเข้ารหัสและถอดรหัส ปกติจะแทนด้วยบิต 0 ทั้งหมด

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

5G PUCCH/PDCCH/PBCH/ PSCCH/PSFCH/PSBCH Encoding

กระบวนการเข้ารหัสสำหรับช่องสัญญาณ PDCCH PUCCH และ PBCH ประกอบด้วย 11 กระบวนการ โดยแต่ละกระบวนการจะถูกใช้งานต่างกันตามช่องสัญญาณการสื่อสาร ช่องสัญญาณที่ใช้งานรหัสโพลาร์จะจับกลุ่มเป็นช่องสัญญาณ 3 ประเภท uplink downlink และ broadcast โดยมีความสัมพันธ์กับช่องสัญญาณทางกายภาพดังนี้

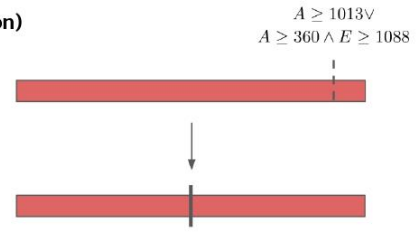
รหัสแอลดีพีซี	รหัสโพลาร์		
	uplink	downlink	broadcast
PUSCH	PUCCH	PDCCH	PBCH
PDSCH		PSCCH	PSBCH
PSSCH		PSFCH	



รูปที่ 32 กระบวนการเข้ารหัสสำหรับช่องสัญญาณ PDCCH PUCCH และ PBCH โดยขั้นตอนในกล่องเส้นทึบจะถูกใช้งานในทุกช่องสัญญาณ กล่องเส้นประขีดจะถูกใช้งานเฉพาะช่องสัญญาณ uplink และกล่องเส้นประจุดใช้งานเฉพาะช่องสัญญาณ broadcast และ downlink

1) การแบ่งย่อยบล็อกรหัส (Code block segmentation) (เฉพาะช่องสัญญาณ PUCCH)

การแบ่งย่อยบล็อกรหัสเป็นการแบ่งข้อมูลเพย์โหลด a ออกเป็น 2 ส่วน โดยจะทำงานเมื่อ $A \geq 1013$ หรือ $A \geq 360 \wedge E \geq 1088$ เข้าเงื่อนไข



รูปที่ 32 การแบ่งย่อยบล็อกรหัส

2) การเข้ารหัส CRC (CRC attachment)
(ทุกช่องสัญญาณ)

การเข้ารหัส CRC จะทำการคำนวณบิต CRC จำนวน L บิต จากนั้นบิต CRC ที่คำนวณได้จะต่อท้ายกับเวกเตอร์ $\mathbf{a}$ ความยาว A (หรือ $\mathbf{a}'$ ความยาว A' ในกรณีที่ $I_{seg}=1$) โดยจะได้เวกเตอร์ $\mathbf{c}$ ที่มีความยาว $K=A+L$ (หรือ $\mathbf{c}'$ ความยาว $K=A'+L$ ในกรณีที่ $I_{seg}=1$) การคำนวณบิต CRC ในมาตรฐาน 5G จะใช้พหุนามกำเนิด (generator polynomial) 3 ตัว ประกอบด้วยพหุนามสำหรับรหัสโพลาร์ดังนี้

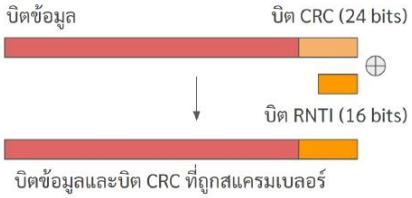
$$g_{CRC6}(x)=x^6+x^5+1$$
$$g_{CRC11}(x)=x^{11}+x^{10}+x^9+x^5+1$$
$$g_{CRC24C}(x)=x^{24}+x^{23}+x^{21}+x^{20}+x^{17}+x^{15}+x^{13}+x^{12}+x^8+x^4+x^2+x+1$$

รหัสโพลาร์ที่มีรหัส CRC สามารถเพิ่มสมรรถนะของรหัสโพลาร์ได้อย่างมีนัยสำคัญ โดยการถอดรหัสโพลาร์ จะทำงานร่วมกับรหัส CRC โดยจะช่วยเลือกบิตจากการถอดรหัส ช่วยลดความผิดพลาดของตัวถอดรหัสได้อย่างมาก

รูปที่ 33 การเข้ารหัส CRC

3) การสแครมบลิง CRC (CRC scrambling)
(เฉพาะช่องสัญญาณ PDCCH PSCCH PSFCH)

เวกเตอร์ $\mathbf{c}$ จากกระบวนการก่อนจะถูกสแครมบลิงโดยดำเนินการเอ็กซ์คลูซีฟออร์บิต RNTI กับบิต CRC 16 บิตสุดท้าย เพื่อให้มีคุณสมบัติ (blind detection หรือ blind decoding)

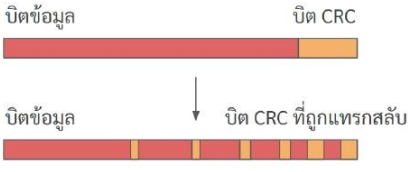


รูปที่ 34 การสแครมบลิง CRC

ในการส่งข้อมูลควบคุมผ่านช่องสัญญาณ downlink อุปกรณ์ฝั่ง RAN จะไม่มีการใส่ข้อมูลส่วนหัว (header) เพื่อระบุตัวตนของอุปกรณ์ผู้ใช้งาน แต่จะมีการสแครมบลิงด้วยรหัสเฉพาะตัว ซึ่งเป็นรหัสเฉพาะตัวของอุปกรณ์ที่ต้องการสื่อสาร เมื่ออุปกรณ์ RAN จะส่งข้อมูลไปยังอุปกรณ์ผู้ใช้งาน อุปกรณ์ของผู้ใช้งานจะทำการถอดรหัสข้อมูลแบบ blind detection การสแครมบลิงจะทำให้ข้อมูลที่ไม่ตรงกับอุปกรณ์ที่ต้องการสื่อสารมีโอกาสถอดรหัสผิดสูง

4) การแทรกสลับ CRC (CRC interleaving)
(เฉพาะช่องสัญญาณ PDCCH
PSCCH PSFCH PBCH PSBCH)

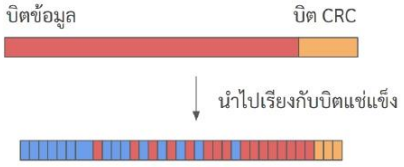
เวกเตอร์ $\mathbf{c}$ ที่ผ่านกระบวนก่อนหน้าจะถูกแทรกสลับตามลำดับ (3GPP, 2017, ดังตารางที่ 5.3.1.1-1) เพื่อลดระยะเวลาในการถอดรหัสด้วยเทคนิค early termination ซึ่งจะหยุดการถอดรหัสระหว่างทางหากพบความผิดพลาดระหว่างการถอดรหัส



รูปที่ 35 การแทรกสลับ CRC

5) การลำดับช่องสัญญาณย่อย (Polar sequencing)
(ทุกช่องสัญญาณ)

ระบบจะสร้างเวกเตอร์ $\mathbf{u}$ ความยาว N บิต โดยการเลือกตำแหน่งของบิตแซ่แข็งและบิตข้อมูล บิตข้อมูลหรือเวกเตอร์ $\mathbf{c}$ (หรือ $\mathbf{c}'$) จากกระบวนการก่อนจะถูกวางไว้ในตำแหน่งบิตข้อมูล ส่วนตำแหน่งบิตแซ่แข็งจะถูกกำหนดค่าเป็นบิต 0

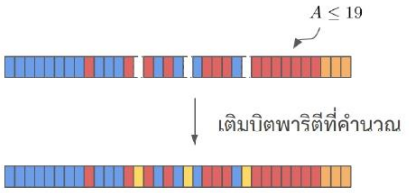


รูปที่ 36 การลำดับช่องสัญญาณย่อย

การเลือกตำแหน่งของบิตแซ่แข็งจะสอดคล้องกับเงื่อนการปรับอัตรารหัสและลำดับความน่าเชื่อถือช่องสัญญาณตามลำดับ (3GPP, 2017, ดังตารางที่ 5.3.1.2-1) โดยลำดับความน่าเชื่อถือของช่องสัญญาณจะเป็นลำดับคงที่ ข้อดีของลำดับความน่าเชื่อถือช่องสัญญาณที่เป็นลำดับคงที่จะช่วยลดความซับซ้อนในการเข้ารหัสโพลาร์ได้เป็นอย่างมาก

6) การคำนวณบิตพาริตี (Parity-check calculation)
(เฉพาะช่องสัญญาณ PUCCH)

เลือกตำแหน่งของบิตพาริตีจำนวน 3 บิตในกระบวนการลำดับช่องสัญญาณย่อย การคำนวณบิตพาริตีสามารถทำได้โดยใช้ชิปรีจิสเตอร์วนซ้ำขนาด 5 บิต โดยตั้งค่าบิตเริ่มต้นเป็นบิต 0 โดยบิตพาริตีจะคำนวณโดยการเอ็กซ์คลูซีฟออร์กับบิตลำดับหน้าเว้นครั้งละ 5 บิต โดยทำกระบวนการเอ็กซ์คลูซีฟออร์เฉพาะบิตข้อมูลซึ่งไม่รวมบิตพาริตีลำดับก่อนและไม่รวมบิตแซ่แข็ง

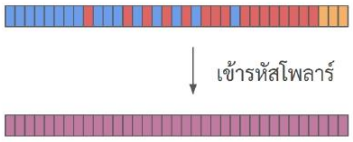


รูปที่ 37 การคำนวณบิตพาริตี

7) การเข้ารหัสโพลาร์ (Polar encoding)
(ทุกช่องสัญญาณ)

การเข้ารหัสโพลาร์สามารถดำเนินการทางคณิตศาสตร์โดยสมการ

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

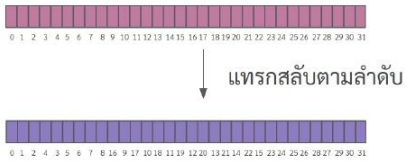


รูปที่ 38 การเข้ารหัสโพลาร์

โดยที่เมทริกซ์กำเนิด (generator matrix) แทนด้วย $\mathbf{G}_N = \mathbf{G}_2^{\otimes n}$ เป็นผลคูณโครเน็กเกอร์ (Kronecker product) และ $\mathbf{G}_2 = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$ มีอินพุตคือ $\mathbf{u}$ ความยาว N บิต เมื่อดำเนินการเข้ารหัสจะได้ผลลัพธ์เป็นคำรหัสแม่ $\mathbf{d}$ ความยาว N

8) การแทรกสลับบล็อกย่อย (Sub-block interleaving)
(ทุกช่องสัญญาณ)

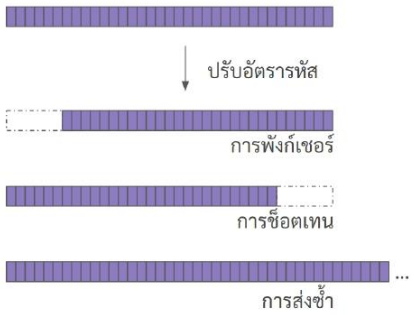
เวกเตอร์ **d** ความยาว N บิต จากกระบวนการก่อนหน้าจะถูกแบ่งเป็น 32 บล็อกย่อย แต่ละบล็อกย่อยมีความยาวเท่ากับ $N/32$ บิต จากนั้นจะแทรกสลับบล็อกย่อยตามลำดับ



รูปที่ 39 การแทรกสลับบล็อกย่อย

9) การปรับอัตรารหัส (Rate matching)
(ทุกช่องสัญญาณ)

การปรับอัตรารหัสถูกกำหนดโดยคำรหัส **e** ที่มีความยาว E บิต ซึ่งเป็นความยาวที่ถูกกำหนดจากการสื่อสารชั้นถัดไป กระบวนการนี้จะทำการตัดบิตออกจากเวกเตอร์ **y** ในกรณีที่ใช้การปรับอัตรารหัสด้วยวิธีการพังก์เชอร์หรือการช็อตเทน หรือส่งบิตเวกเตอร์ **y** ซ้ำในกรณีที่ใช้การส่งซ้ำ รายละเอียดของการปรับอัตรารหัสทั้ง 3 รูปแบบ โดยมีรายละเอียดดังนี้

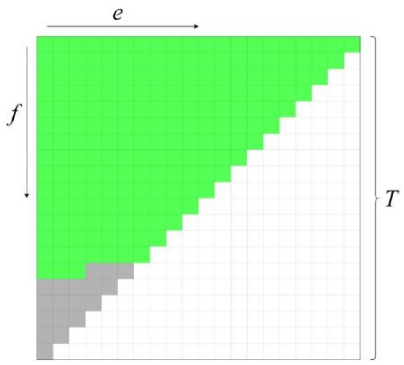


รูปที่ 40 การปรับอัตรารหัส

1) การพังก์เจอร์ (puncturing) จะทำเมื่อ $E < N$ และ $K/E \leq 7/16$ โดยจะไม่ส่งบิต $U = N - E$ แรก
2) การช็อตเทน (shortening) จะทำเมื่อ $E < N$ และ $K/E > 7/16$ โดยจะไม่ส่งบิต $U = N - E$ สุดท้าย
3) การส่งซ้ำ (repetition) จะทำเมื่อ $E > N$ โดยจะส่งบิต $U = N - E$ แรกซ้ำ

10) การแทรกสลับบิตรหัส (Coded bit interleaving)
(เฉพาะช่องสัญญาณ PUCCH)

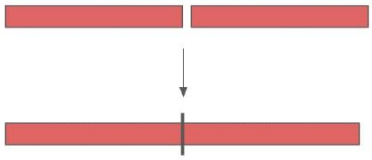
การแทรกสลับแบบสามเหลี่ยมขั้นบันไดสามารถลดอัตราบล็อกผิดพลาดในการมอดูเลชันลำดับสูง เวกเตอร์ **e** แต่ละบิตจะถูกเขียนลงในโครงสร้างสามเหลี่ยมสามเหลี่ยมควำ่ จากซ้ายไปขวาและมีทิศทางจากบนลงล่าง จากนั้นจะสร้างคำรหัสที่ผ่านการแทรกสลับบิตรหัส **f** โดยการอ่านค่าจากโครงสร้างสามเหลี่ยมจากทิศทางบนลงล่างและมีทิศทางจากซ้ายไปขวา ส่วนที่เว้าของโครงสร้างสามเหลี่ยมสามารถแทนค่าด้วย null โดยที่เวกเตอร์ **e** และ **f** มีขนาดเท่ากัน



รูปที่ 41 การแทรกสลับบิตรหัส

11) การต่อบล็อกรหัส (Code block concatenation)
(เฉพาะช่องสัญญาณ PUCCH)

หากกระบวนการข้างต้นมีการแบ่งบล็อกรหัสหรือมีเงื่อนไข $\left(A \geq 360 \wedge E \geq 1088\right) \vee A \geq 1013$ และ $I_{seg} = 1$ เวกเตอร์ $\mathbf{f}$ จำนวนสองเวกเตอร์จะถูกต่อกันตามลำดับเดิมและส่งไปยังการสื่อสารชั้นถัดไป



รูปที่ 42 การต่อบล็อกรหัส

5G Modulation

การมอดูเลต (Modulation) เป็นการแปลงสัญญาณข่าวสารที่เป็นบิต '0' หรือ '1' ให้อยู่ในรูปของสัญญาณส่ง โดยในมาตรฐาน 5G ได้กำหนดรูปแบบการมอดูเลตไว้ 7 แบบ

นอกจากนี้ รูปแบบการมอดูเลชันที่สามารถใช้งานได้จะแตกต่างกันตามช่องสัญญาณกายภาพ

การมอดูเลต	จำนวนบิตต่อสัญลักษณ์ (Qm)
BPSK	1
π/2 – BPSK	1
QPSK	2
16QAM	4
64QAM	6
256QAM	8
1024QAM	10

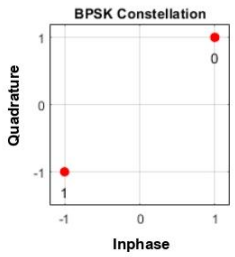
ช่องสัญญาณกายภาพ	รูปแบบการมอดูเลต						
	BPSK	π/2- BPSK	QPSK	16QAM	64QAM	256QAM	1024QAM
PDSCH			✓	✓	✓	✓	✓
PDCCH			✓				
PUSCH		✓	✓	✓	✓		
PUCCH	✓	✓	✓				
PBCH			✓				

1) การมอดูเลตแบบ BPSK

กำหนดให้ b_i แทนบิตข้อมูลบิตที่ i ในสัญญาณข่าวสาร สัญลักษณ์เชิงซ้อน s_i ในสัญญาณส่งเมื่อมีการมอดูเลตแบบ BPSK สามารถหาได้จากสมการ

$$s_i = \frac{1}{\sqrt{2}}\left[(1-2b_i)+j(1-2b_i)\right] \quad (19)$$

แผนภาพคอนสเตลเลชันของการมอดูเลตแบบ BPSK แสดงดังรูปที่ 43 โดยหนึ่งจุดบนคอนสเตลเลชัน แทนหนึ่งสัญลักษณ์ ซึ่งในแต่ละสัญลักษณ์แทนบิตข้อมูล 1 บิต และ $n_m = \frac{1}{\sqrt{2}}$ แทนตัวปรับมาตรฐาน (Normalization Factor) สำหรับการมอดูเลตแบบ BPSK



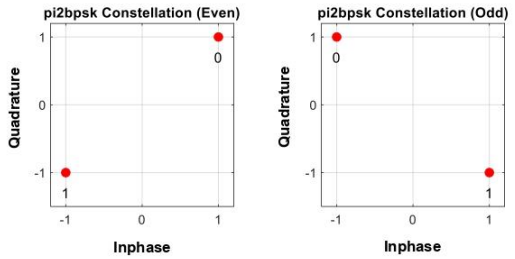
รูปที่ 43 แผนภาพคอนสเตลเลชันของการมอดูเลตแบบ BPSK กรณีไม่มีตัวปรับมาตรฐาน

2) การมอดูเลตแบบ $\pi/2$ – BPSK

กำหนดให้ b_i แทนบิตข้อมูลบิตที่ i ในสัญญาณข่าวสาร สัญลักษณ์เชิงซ้อน s_i ในสัญญาณส่งเมื่อมีการมอดูเลตแบบ $\pi/2$ - BPSK สามารถหาได้จากสมการ

$$s_i = \frac{e^{j\frac{\pi}{2}(i \bmod 2)}}{\sqrt{2}}\left[(1-2b_i)+j(1-2b_i)\right] \quad ((20)$$

แผนภาพคอนสเตลเลชันของการมอดูเลตแบบ $\pi/2$ - BPSK แสดงดังรูปที่ 44 โดยหนึ่งจุดบนคอนสเตลเลชันแทนหนึ่งสัญลักษณ์ ซึ่งในแต่ละสัญลักษณ์แทนบิตข้อมูล 1 บิต และ $n_m = \frac{1}{\sqrt{2}}$ แทนตัวปรับมาตรฐานสำหรับการมอดูเลตแบบ $\pi/2$ - BPSK



(ก) กรณีบิตคู่ (ข) กรณีบิตคี่

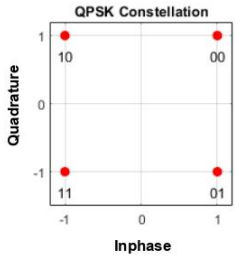
รูปที่ 44 แผนภาพคอนสเตลเลชันของการมอดูเลตแบบ $\pi/2$ – BPSK กรณีไม่มีตัวปรับมาตรฐาน

3) การมอดูเลตแบบ QPSK

กำหนดให้ b_i แทนบิตข้อมูลบิตที่ i ในสัญญาณข่าวสาร สัญลักษณ์เชิงซ้อน s_i ในสัญญาณส่งเมื่อมีการมอดูเลตแบบ QPSK สามารถหาได้จากสมการ

$$s_i = \frac{1}{\sqrt{2}}\left[(1-2b_{2i})+j(1-2b_{2i+1})\right] \quad (21)$$

แผนภาพคอนสเตลเลชันของการมอดูเลตแบบ QPSK แสดงดังรูปที่ 45 โดยหนึ่งจุดบนคอนสเตลเลชัน แทนหนึ่งสัญลักษณ์ ซึ่งในแต่ละสัญลักษณ์แทนบิตข้อมูล 2 บิต และ $n_m = \frac{1}{\sqrt{2}}$ แทนตัวปรับมาตรฐานสำหรับการมอดูเลตแบบ QPSK



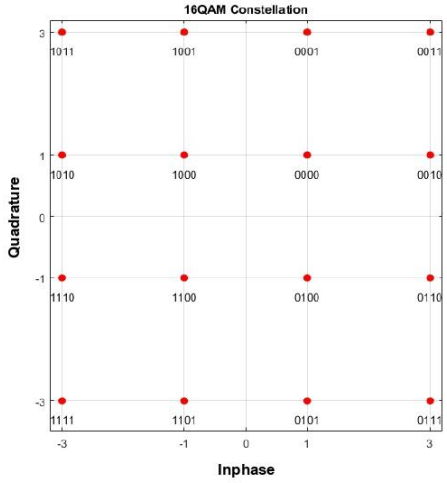
รูปที่ 45 แผนภาพคอนสเตลเลชันของการมอดูเลตแบบ QPSK กรณีไม่มีตัวปรับมาตรฐาน

4) การมอดูเลตแบบ 16QAM

กำหนดให้ b_i แทนบิตข้อมูลบิตที่ i ในสัญญาณข่าวสาร สัญลักษณ์เชิงซ้อน s_i ในสัญญาณส่งเมื่อมีการมอดูเลตแบบ 16QAM สามารถหาได้จากสมการ

$$s_i = \frac{1}{\sqrt{10}}\left\{(1-2b_{4i})\left[2-(1-2b_{4i+2})\right]+j(1-2b_{4i+1})\left[2-(1-2b_{4i+3})\right]\right\} \quad (22)$$

แผนภาพคอนสเตลเลชันของการมอดูเลตแบบ 16QAM แสดงดังรูปที่ 46 โดยหนึ่งจุดบนคอนสเตลเลชันแทนหนึ่งสัญลักษณ์ ซึ่งในแต่ละสัญลักษณ์แทนบิตข้อมูล 4 บิต และ $n_m = \frac{1}{\sqrt{10}}$ แทนตัวปรับมาตรฐานสำหรับการมอดูเลตแบบ 16QAM



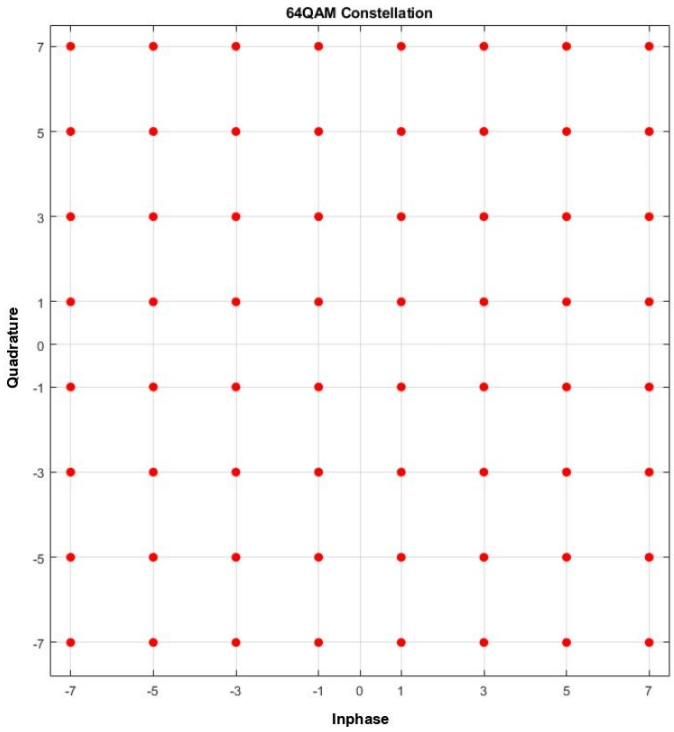
รูปที่ 46 แผนภาพคอนสเตลเลชันของการมอดูเลตแบบ 16QAM กรณีไม่มีตัวปรับมาตรฐาน

5) การมอดูเลตแบบ 64QAM

กำหนดให้ b_i แทนบิตข้อมูลบิตที่ i ในสัญญาณข่าวสาร สัญลักษณ์เชิงซ้อน s_i ในสัญญาณส่งเมื่อมีการมอดูเลตแบบ 64QAM สามารถหาได้จากสมการ

$$\begin{aligned} s_i = \frac{1}{\sqrt{42}}\{&(1-2b_{6i})\left[4-(1-2b_{6i+2})\left[2-(1-2b_{6i+4})\right]\right] \\ &+j(1-2b_{6i+1})\left[4-(1-2b_{6i+3})\left[2-(1-2b_{6i+5})\right]\right]\} \end{aligned} \quad (23)$$

แผนภาพคอนสเตลเลชันของการมอดูเลตแบบ 64QAM แสดงดังรูปที่ 47 โดยหนึ่งจุดบนคอนสเตลเลชันแทนหนึ่งสัญลักษณ์ ซึ่งในแต่ละสัญลักษณ์แทนบิตข้อมูล 6 บิต และ $n_m = \frac{1}{\sqrt{42}}$ แทนตัวปรับมาตรฐานสำหรับการมอดูเลตแบบ 64QAM



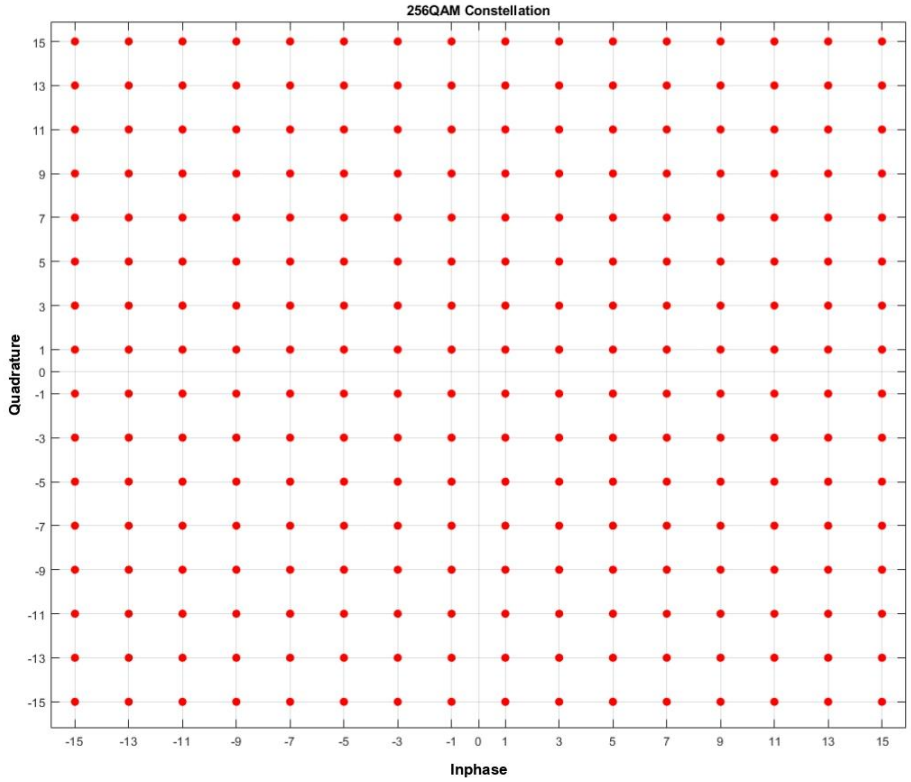
รูปที่ 47 แผนภาพคอนสเตลเลชันของการมอดูเลตแบบ 64QAM กรณีไม่มีตัวปรับมาตรฐาน

6) การมอดูเลตแบบ 256QAM

กำหนดให้ b_i แทนบิตข้อมูลบิตที่ i ในสัญญาณข่าวสาร สัญลักษณ์เชิงซ้อน s_i ในสัญญาณส่งเมื่อมีการมอดูเลตแบบ 256QAM สามารถหาได้จากสมการ

$$s_i = \frac{1}{\sqrt{170}}\Big\{(1-2b_{8i})\Big[8-(1-2b_{8i+2})\Big[4-(1-2b_{8i+4})\Big[2-(1-2b_{8i+6})\Big]\Big]\Big] + j(1-2b_{8i+1})\Big[8-(1-2b_{8i+3})\Big[4-(1-2b_{8i+5})\Big[2-(1-2b_{8i+7})\Big]\Big]\Big]\Big\} \quad (24)$$

แผนภาพคอนสเตลเลชันของการมอดูเลตแบบ 256QAM แสดงดังรูปที่ 48 โดยหนึ่งจุดบนคอนสเตลเลชันแทนหนึ่งสัญลักษณ์ ซึ่งในแต่ละสัญลักษณ์แทนบิตข้อมูล 8 บิต และ $n_m = \frac{1}{\sqrt{170}}$ แทนตัวปรับมาตรฐานสำหรับการมอดูเลตแบบ 256QAM



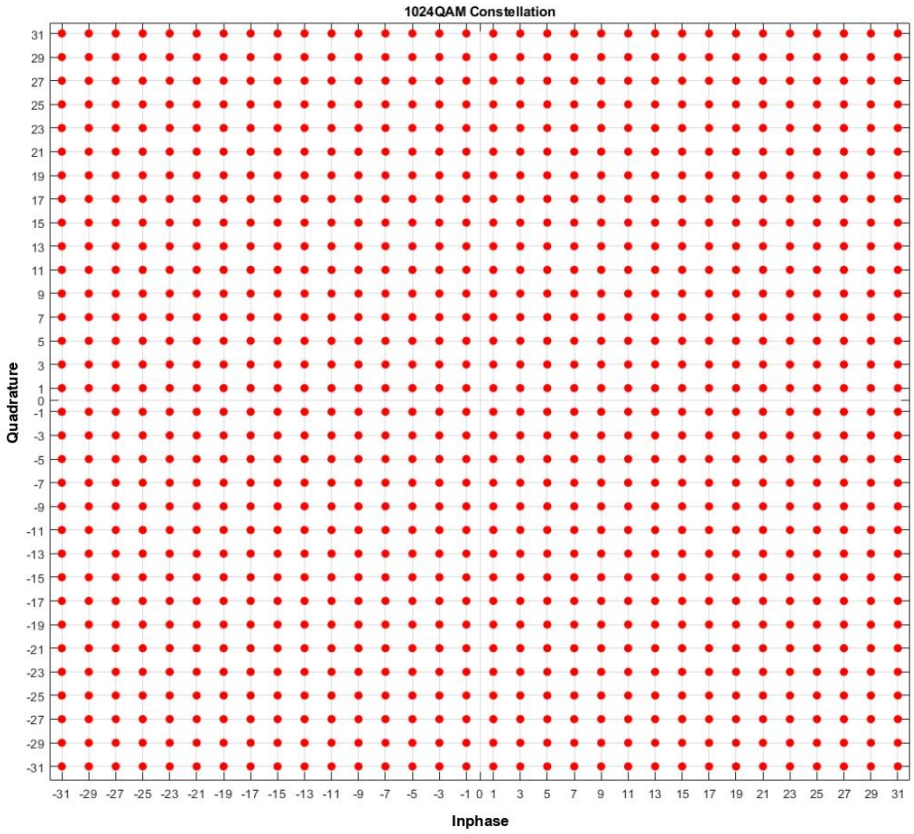
รูปที่ 48 แผนภาพคอนสเตลเลชันของการมอดูเลตแบบ 256QAM กรณีไม่มีตัวปรับมาตรฐาน

7) การมอดูเลตแบบ 1024QAM

กำหนดให้ b_i แทนบิตข้อมูลบิตที่ i ในสัญญาณข่าวสาร สัญลักษณ์เชิงซ้อน s_i ในสัญญาณส่งเมื่อมีการมอดูเลตแบบ 1024QAM สามารถหาได้จากสมการ

$$s_i = \frac{1}{\sqrt{682}}\Big\{(1-2b_{10i})\Big[16-(1-2b_{10i+2})\Big[8-(1-2b_{10i+4})\Big[4-(1-2b_{10i+6})\Big[2-(1-2b_{10i+8})\Big]\Big]\Big]\Big] + j(1-2b_{10i+1})\Big[16-(1-2b_{10i+3})\Big[8-(1-2b_{10i+5})\Big[4-(1-2b_{10i+7})\Big[2-(1-2b_{10i+9})\Big]\Big]\Big]\Big]\Big\} \quad (25)$$

แผนภาพคอนสเตลเลชันของการมอดูเลตแบบ 1024QAM แสดงดังรูปที่ 49 โดยหนึ่งจุดบนคอนสเตลเลชันแทนหนึ่งสัญลักษณ์ ซึ่งในแต่ละสัญลักษณ์แทนบิตข้อมูล 10 บิต และ $n_m = \frac{1}{\sqrt{682}}$ แทนตัวปรับมาตรฐานสำหรับการมอดูเลตแบบ 1024QAM



รูปที่ 49 แผนภาพคอนสเตลเลชันของการมอดูเลตแบบ 1024QAM กรณีไม่มีตัวปรับมาตรฐาน

Noisy Channel

กฤติยาภรณ์ เหมือดขุนทด จตุพร ด้วงทอง ธนัช ศรีสุภา อนุสรณ์ วงค์ษา
รศ.ดร.กฤษณะพงศ์ พันธ์ศรี และ รศ.ดร.เวธิต ภาคย์พิสุทธิ์

AWGN Channel

1) ช่องสัญญาณรบกวนเกาส์เซียนขาวบวก

ช่องสัญญาณรบกวนเกาส์เซียนขาวบวก (Additive White Gaussian Noise: AWGN) คือรูปแบบช่องสัญญาณพื้นฐาน เพื่ออธิบายถึงปรากฏการณ์ของกระบวนการบวกของสัญญาณสุ่มที่เกิดขึ้นในธรรมชาติ รวมถึงในระบบการสื่อสาร สัญญาณรบกวนประเภทนี้มักใช้ในการอธิบายสัญญาณที่อาจเกิดขึ้นจากธรรมชาติหรืออุณหภูมิของอุปกรณ์ เรียกว่าสัญญาณรบกวนเชิงความร้อน

สำหรับช่องสัญญาณรบกวนดังกล่าวแสดงแบบจำลองดังรูปที่ 50 โดยกำหนดให้ $y(t)$ เป็นเอาต์พุตช่องสัญญาณ ซึ่งเกิดจากสัญญาณที่ส่ง $x(t)$ บวกกับสัญญาณรบกวนเกาส์เซียน $n(t)$ เขียนความสัมพันธ์ของสัญญาณต่าง ๆ ได้ดังนี้

และสามารถแทนสัญญาณรบกวนเกาส์เซียนได้ดังนี้

โดยที่สัญญาณรบกวนเกาส์เซียน $n(t)$ มีค่าเฉลี่ยเท่ากับ 0 และมีความแปรปรวนเท่ากับ σ^2

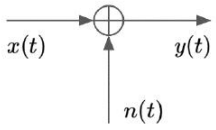
1) บวก (Additive) เนื่องจากสัญญาณประเภทนี้ถูกเพิ่มหรือบวกเข้าไปยังระบบหรือช่องสัญญาณ

2) ขาว (White) อ้างถึงความหนาแน่นกำลังเชิงสเปกตรัม (Power Spectral Density) ที่มีลักษณะสม่ำเสมอในโดเมนความถี่ภายในระบบหรือช่องสัญญาณ ซึ่งลักษณะต่าง ๆ จะถูกแทนด้วยชื่อสีและสีขาวจะมีลักษณะสม่ำเสมอ

3) เกาส์เซียน (Gaussian) อ้างอิงความหนาแน่น (Density) ที่มีลักษณะการกระจายปกติ (Normal Distribution) หรือการกระจายเกาส์เซียน (Gaussian Distribution) ในโดเมนเวลาภายในระบบหรือช่องสัญญาณ

$$y(t) = x(t) + n(t) \quad (26)$$

$$n(t) \sim \mathcal{N}(0, \sigma^2) \quad (27)$$

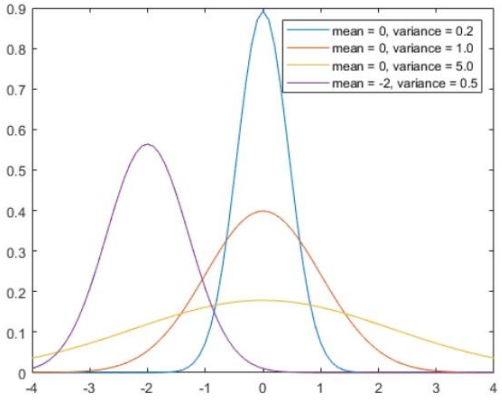


รูปที่ 50 แบบจำลองช่องสัญญาณรบกวนเกาส์เซียนขาวแบบบวก

หากกำหนดให้สัญญาณที่ส่งมีสัญญาณความถี่เดียวหรือมีฟังก์ชันความหนาแน่นสเปกตรัมแบบอิมพัลส์ (Impulse) เมื่อสัญญาณดังกล่าวผ่านช่องสัญญาณรบกวนนี้ สัญญาณที่ส่งจะถูกบวกกับสัญญาณรบกวนเกาส์เซียนที่ช่วงเวลาต่าง ๆ ทำให้ได้เอาต์พุตช่องสัญญาณที่มีการกระจายของฟังก์ชันความหนาแน่นสเปกตรัมแบบเกาส์เซียนเช่นเดียวกับสัญญาณรบกวนเกาส์เซียน แสดงดังรูปที่ 51 และเขียนสมการฟังก์ชันความหนาแน่นแบบเกาส์เซียนได้ดังนี้

$$f(y)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(y-\mu)^2}{2\sigma^2}} \tag{28}$$

โดย μ คือค่าเฉลี่ยของการกระจาย (Distribution) σ^2 คือความแปรปรวน (Variance) และ y คือตัวแปรสุ่ม ซึ่งสามารถแทนด้วยสัญญาณต่าง ๆ เช่น เอาต์พุตช่องสัญญาณ $y(t)$ สัญญาณรบกวนเกาส์เซียน ณ เวลาต่าง ๆ ซึ่งมีคุณสมบัติการกระจายที่อิสระและเหมือนกัน (Independent and Identically Distributed)

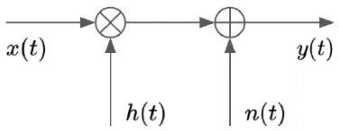


รูปที่ 51 ฟังก์ชันความหนาแน่นสเปกตรัมแบบเกาส์เซียน

Fading Channel

2) ช่องสัญญาณการเฟด

ช่องสัญญาณการเฟด (Fading Channel) คือรูปแบบช่องสัญญาณพื้นฐานที่อธิบายเหตุการณ์การเฟด (Fading) ของสัญญาณ โดยการเฟดคือการขยาย-หดของขนาดสัญญาณ เช่น สัญญาณถูกส่งจากสถานีฐานไปยังอุปกรณ์เคลื่อนที่ในพื้นที่หนึ่งจะถูกเฟดทั้งขนาด (Amplitude) และเฟส (Phase) ซึ่งอาจเกิดขึ้นจากการกระเจิงของสัญญาณจากสิ่งต่าง ๆ เช่น ต้นไม้ เสา หรือกำแพง แบบจำลองช่องสัญญาณการเฟดทั่วไปแสดงดังรูปที่ 51 ช่องสัญญาณการเฟดมีแบบจำลองช่องสัญญาณอยู่หลากหลายรูปแบบจากตัวแปรที่ส่งผลต่อการเฟด หนึ่งในพื้นฐานแบบจำลองช่องสัญญาณการเฟด คือ ช่องสัญญาณการเฟดแบบเรย์ลีห์ (Rayleigh Fading Channel)



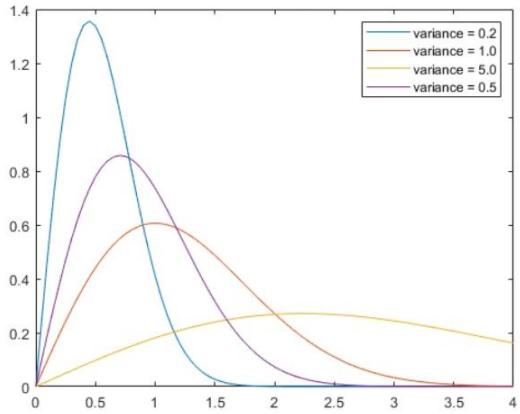
รูปที่ 51 แบบจำลองช่องสัญญาณการเฟดทั่วไป

ช่องสัญญาณการเฟดแบบเรย์ลีห์ คือแบบจำลองช่องสัญญาณที่สัญญาณส่งเกิดการขยาย-หดทางขนาดและการเฟด ตามการกระจายแบบเรย์ลีห์ ช่องสัญญาณดังกล่าวเหมาะสำหรับการจำลองช่องสัญญาณภายใต้สภาพแวดล้อมของชั้นบรรยากาศโทรโพสเพียร์ ไอโอโนสเพียร์ รวมถึงสภาพชุมชนเมือง นอกจากนี้ยังเหมาะกับแบบจำลองที่ไม่มีการแพร่กระจายสัญญาณทางตรง กล่าวคือสัญญาณที่ได้รับอาจมาจากการสะท้อนหรือถูกลดทอนทั้งหมด แบบจำลองของช่องสัญญาณการเฟดแบบเรย์ลีห์แสดงดังรูปที่ 51 โดยกำหนดให้ $y(t)$ เป็นเอาต์พุตช่องสัญญาณ ซึ่งเกิดจากสัญญาณที่ส่ง $x(t)$ ผ่านช่องสัญญาณการเฟดที่มีคุณสมบัติช่องสัญญาณแทนด้วย $h(t)$ ซึ่งจะถูกคูณเข้าไปกับสัญญาณที่ส่ง และรวมกับสัญญาณรบกวนเกาส์เซียน $n(t)$ ซึ่งสามารถเขียนความสัมพันธ์ของสัญญาณต่าง ๆ ได้ดังนี้

$$y(t) = h(t)x(t) + n(t) \tag{30}$$

โดยที่ $h(t)$ คือค่าสัมประสิทธิ์ช่องสัญญาณเชิงซ้อน (Complex Channel Coefficient) ซึ่งประกอบไปด้วยสัญญาณรบกวนเกาส์เซียนที่เป็นอิสระกัน 2 ชุด ที่มีค่าจำนวนจริงและค่าจำนวนเชิงซ้อนแทนด้วย h_I และ h_Q ที่ทั้งคู่มีค่าเฉลี่ยเท่ากับ 0 และความแปรปรวนเท่ากับ σ_h^2 ตามลำดับ และหากแทนขนาดสัญญาณที่มีการกระจายเกาส์เซียนทั้ง 2 ด้วย T_I และ T_Q ตามลำดับ ขนาดของ $h = h_I + jh_Q$ จะได้เป็น $|h| = \sqrt{h_I^2 + h_Q^2}$ ซึ่งสามารถแสดงฟังก์ชันความหนาแน่นแบบเรย์ลีห์ของ $|h|$ ดังรูปที่ 52 และเขียนสมการได้โดย

$$p(|h|) = \begin{cases} \dfrac{h}{\sigma_h^2} e^{-\frac{h^2}{2\sigma_h^2}}, & u \geq 0 \\ 0, & u < 0 \end{cases} \quad (31)$$



รูปที่ 52 ฟังก์ชันความหนาแน่นสเปกตรัมแบบเรย์ลีห์

ภาคผนวก ข
เอกสารประกอบการเผยแพร่ในมหาวิทยาลัย ภาคปฏิบัติ

เอกสารดังต่อไปนี้คือเอกสารที่ใช้ในการเผยแพร่ชุดการเรียนรู้และการถ่ายทอดความรู้พื้นฐานในส่วนการทดสอบใช้งานซอฟต์แวร์ชุดการเรียนรู้ระบบสื่อสารมาตรฐาน 5G

5G Physical-layer Processing

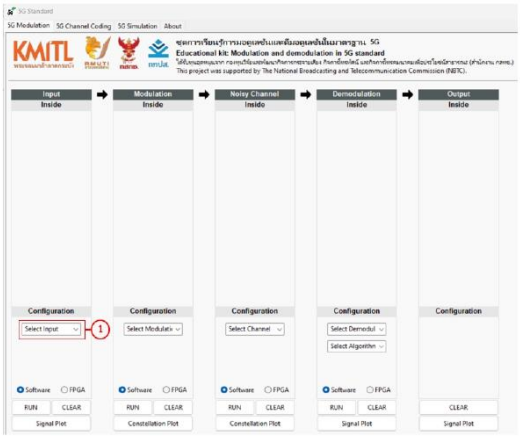
กฤติยาภรณ์ เหมือดขุนทด จตุพร ด้วงทอง ธนัช ศรีสุภา อนุสรณ์ วงค์ษา
รศ.ดร.กฤษณะพงศ์ พันธ์ศรี และ รศ.ดร.เวธิต ภาคย์พิสุทธิ์

การทดลองที่ 1 การมอดูเลชัน

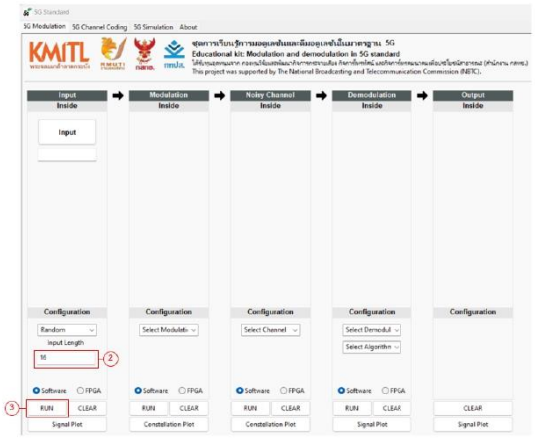
<u>ส่วนที่ 1</u> การสร้างบิตข้อมูลอินพุต

วิธีการทดลอง

1) สร้างข้อมูลอินพุตแบบสุ่ม โดยคลิก Select Input และเลือก Random

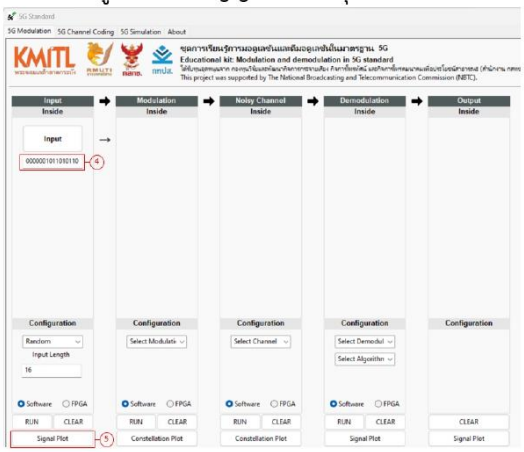


2) ป้อนความยาวบิตอินพุตที่ต้องการ
3) กดปุ่ม RUN เพื่อดำเนินการสร้างอินพุต



4) ผลลัพธ์จะแสดงภายในส่วน Inside โดยผู้ใช้สามารถเลื่อนดูผลลัพธ์ทั้งหมด หรือคัดลอกข้อมูล

5) กดปุ่ม Signal Plot เพื่อดูกราฟของสัญญาณบิตอินพุต



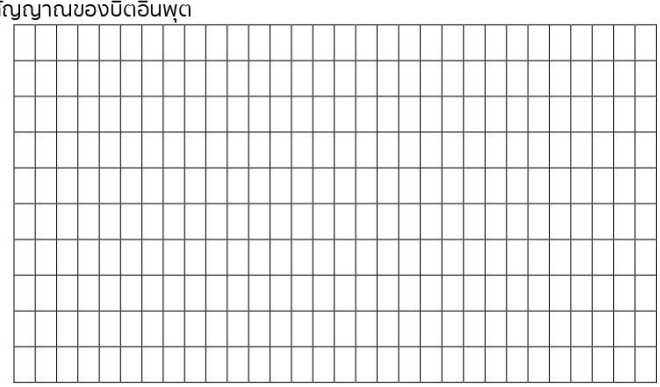
การทดลองและบันทึกผล

ให้นักศึกษาทดลองสร้างบิตข้อมูลอินพุตขนาด 16 บิต และวาดกราฟสัญญาณของบิตอินพุต

บิตอินพุต

.....

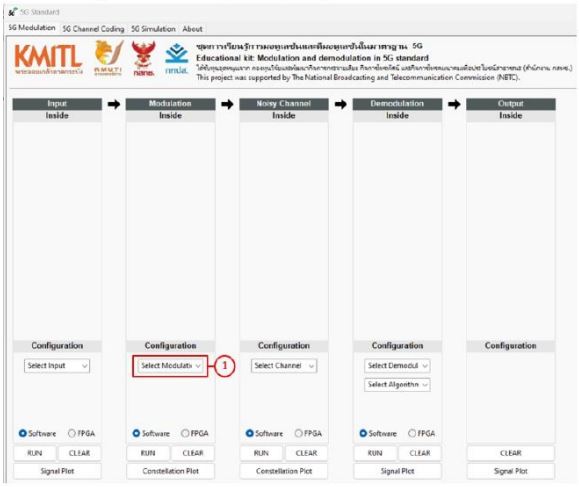
กราฟสัญญาณของบิตอินพุต



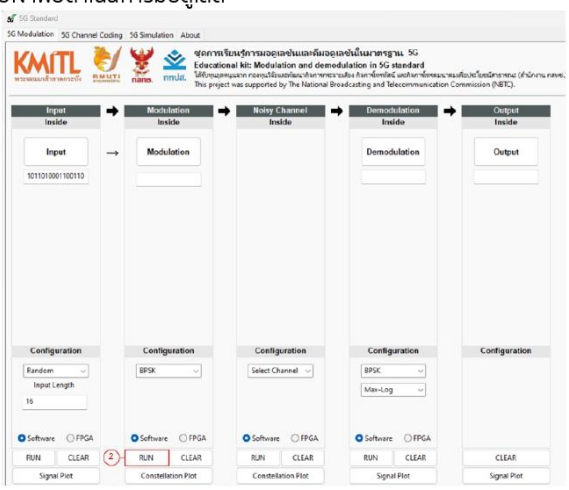
<u>ส่วนที่ 2</u> การมอดูเลต

วิธีการทดลอง

1) เลือกรูปแบบการมอดูเลต โดยคลิก Select Modulati

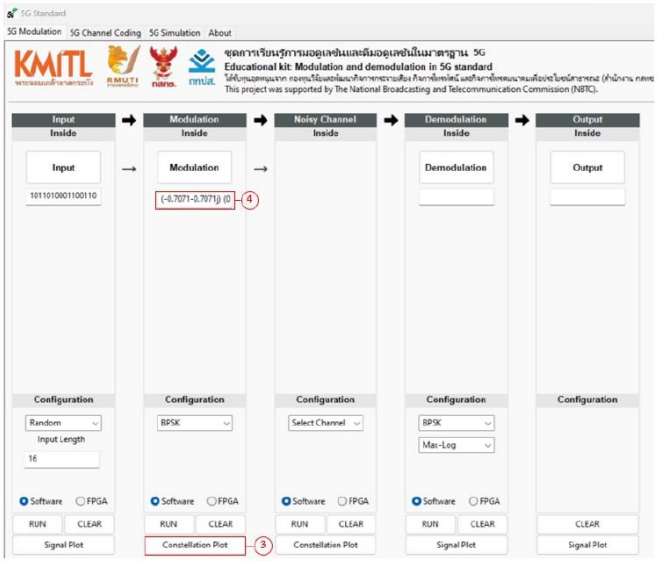


2) กดปุ่ม RUN เพื่อดำเนินการมอดูเลต



3) ผลลัพธ์จะแสดงภายในส่วน Inside โดยผู้ใช้สามารถเลื่อนดูผลลัพธ์ทั้งหมด หรือคัดลอกข้อมูล

4) กดปุ่ม Constellation Plot เพื่อดูแผนภาพคอนสเตลเลชันของสัญญาณมอดูเลต



การทดลองและบันทึกผล

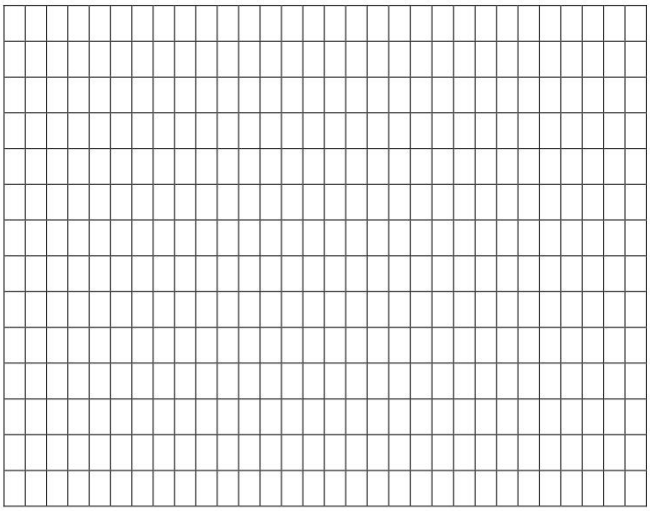
ให้นักศึกษาทดลองมอดูเลตสัญญาณของบิตข้อมูลอินพุตขนาด 16 บิต โดยกำหนดรูปแบบการมอดูเลตแบบ $\pi/2$ - BPSK และวาดแผนภาพคอนสเตลเลชันของการมอดูเลต

สัญลักษณ์ที่ได้จากการมอดูเลต

.....

.....

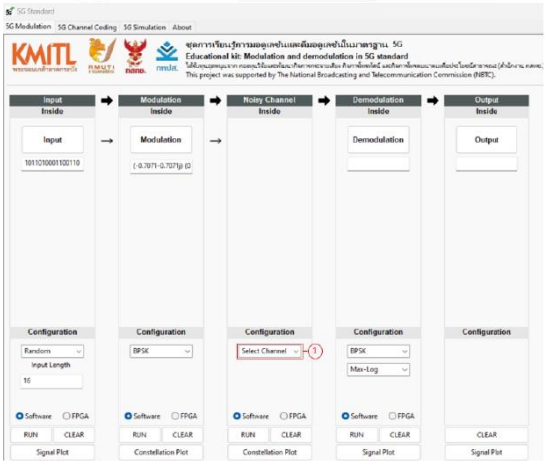
แผนภาพคอนสเตลเลชัน



<u>ส่วนที่ 3</u> ช่องสัญญาณรบกวน

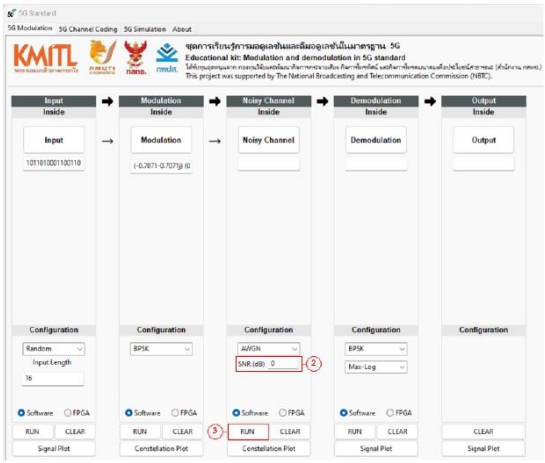
วิธีการทดลอง

1) เลือกรูปแบบช่องสัญญาณรบกวน โดย Select Channel



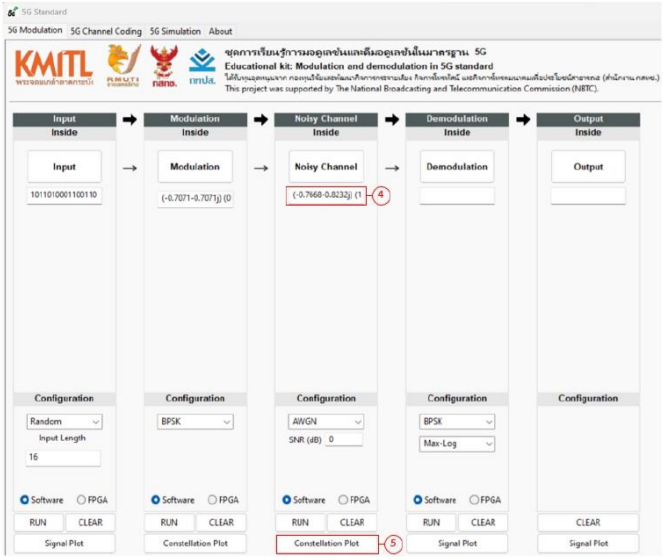
2) กำหนดค่า SNR ที่ต้องการ

3) กดปุ่ม RUN เพื่อดำเนินการสร้างสัญญาณรบกวน และรวมผลกระทบของสัญญาณรบกวนเข้ากับสัญญาณมอดูเลต



4) ผลลัพธ์จะแสดงภายในส่วน Inside โดยผู้ใช้สามารถเลื่อนดูผลลัพธ์ทั้งหมด หรือคัดลอกข้อมูล

5) กดปุ่ม Constellation Plot เพื่อดูแผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านช่องสัญญาณรบกวน



การทดลองและบันทึกผล

ให้นักศึกษาทดลองผลกระทบของช่องสัญญาณแบบ Fading ที่มีบิตข้อมูลอินพุตขนาด 16 บิต และมีรูปแบบการมอดูเลตแบบ $\pi/2$ - BPSK โดยกำหนดค่า SNR = -5 dB และ SNR = 5 dB พร้อมทั้งวาดแผนภาพคอนสเตลเลชันของสัญญาณที่ผ่านช่องสัญญาณรบกวนทั้ง 2 กรณี

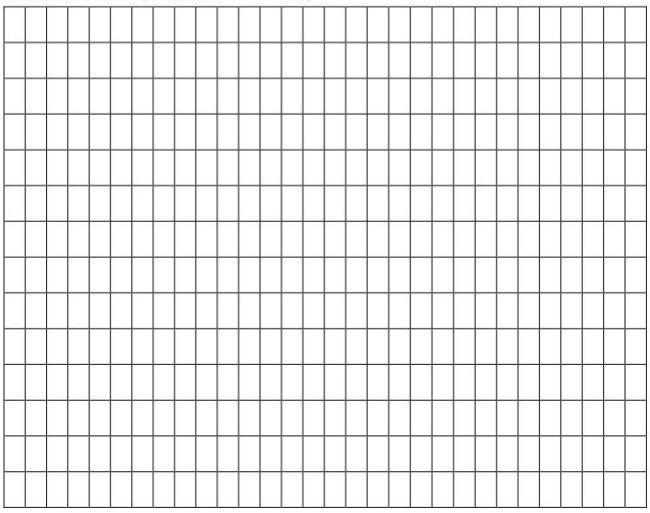
สัญญาณที่ผ่านช่องสัญญาณรบกวน (กรณี SNR = -5 dB)

.....

.....

.....

แผนภาพคอนสเตลเลชัน (กรณี SNR = -5 dB)

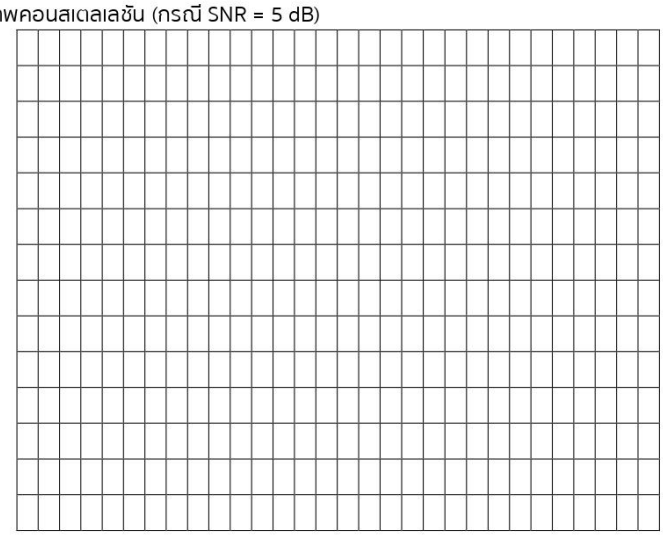


สัญญาณที่ผ่านช่องสัญญาณรบกวน (กรณี SNR = 5 dB)

.....

.....

แผนภาพคอนสเตลเลชัน (กรณี SNR = 5 dB)

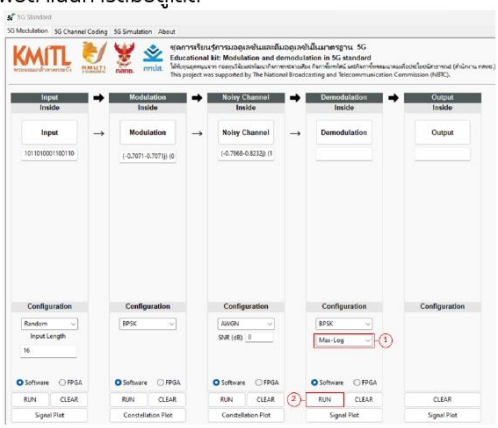


<u>ส่วนที่ 4</u> การดีมอดูเลต

วิธีการทดลอง

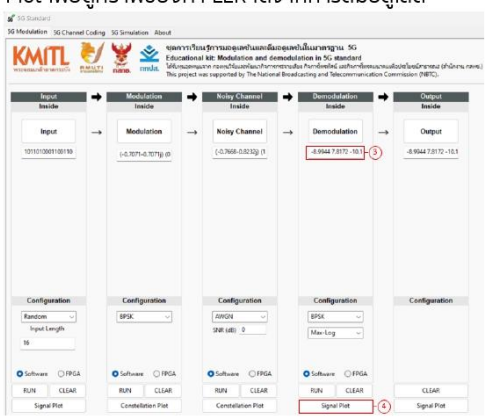
1) เลือกอัลกอริทึมการดีมอดูเลต

2) กดปุ่ม RUN เพื่อดำเนินการดีมอดูเลต



3) ผลลัพธ์จะแสดงภายในส่วน Inside โดยผู้ใช้สามารถเลื่อนดูผลลัพธ์ทั้งหมด หรือคัดลอกข้อมูล

4) กดปุ่ม Signal Plot เพื่อดูกราฟของค่า LLR ได้จากการดีมอดูเลต



การทดลองและบันทึกผล

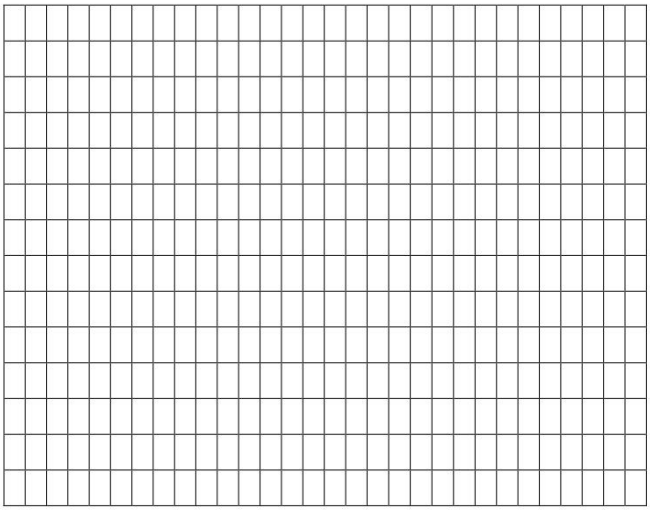
ให้นักศึกษาทดลองการดีมอดูเลตสัญญาณ ที่มีบิตข้อมูลอินพุตขนาด 16 บิต และมีรูปแบบการมอดูเลตแบบ $\pi/2$ - BPSK โดยส่งผ่านช่องสัญญาณแบบ Fading ที่มีค่า SNR = 5dB และกำหนดให้ใช้อัลกอริทึมการดีมอดูเลตแบบ Mag-Log พร้อมทั้งวาดกราฟของค่า LLR ที่จากการดีมอดูเลต

ค่า LLR ที่ได้จากการดีมอดูเลต

.....

.....

กราฟของค่า LLR



การทดลองที่ 2 รหัสช่องสัญญาณ

<u>ส่วนที่ 1</u> การสร้างบิตข้อมูลอินพุต

วิธีการทดลอง

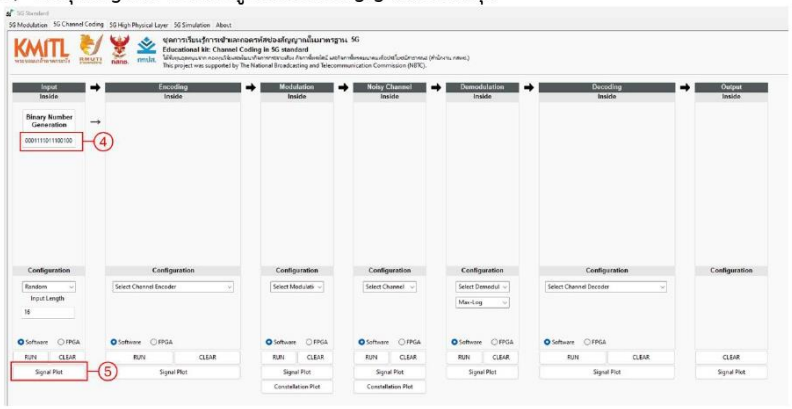
1) สร้างข้อมูลอินพุตแบบสุ่ม โดยคลิก Select Input และเลือก Random

2) ป้อนความยาวบิตอินพุตที่ต้องการ

3) กดปุ่ม RUN เพื่อดำเนินการสร้างอินพุต

4) ผลลัพธ์จะแสดงภายในส่วน Inside โดยผู้ใช้สามารถเลื่อนดูผลลัพธ์ทั้งหมด หรือคัดลอกข้อมูล

5) กดปุ่ม Signal Plot เพื่อดูกราฟของสัญญาณบิตอินพุต



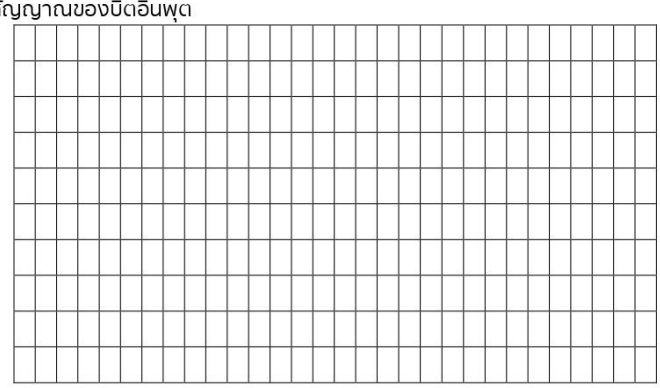
การทดลองและบันทึกผล

ให้นักศึกษาทดลองสร้างบิตข้อมูลอินพุตขนาด 16 บิต และวาดกราฟสัญญาณของบิตอินพุต

บิตอินพุต

.....

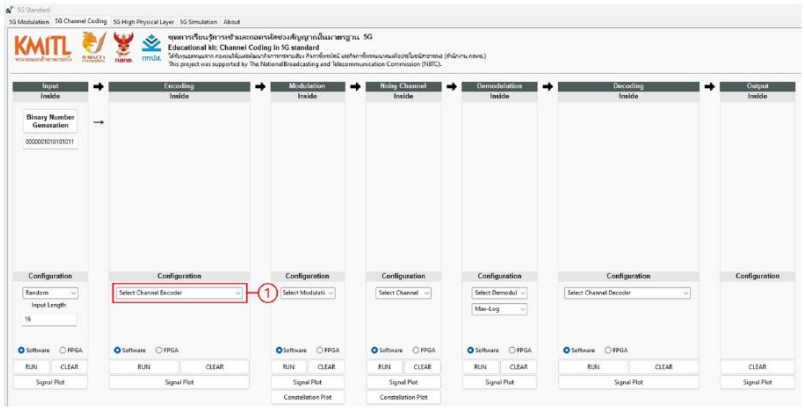
กราฟสัญญาณของบิตอินพุต



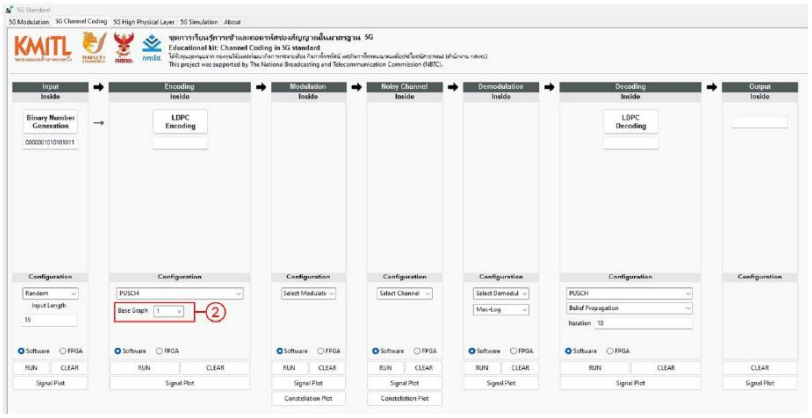
<u>ส่วนที่ 2</u> การเข้ารหัสแอลดีพีซี

วิธีการทดลอง

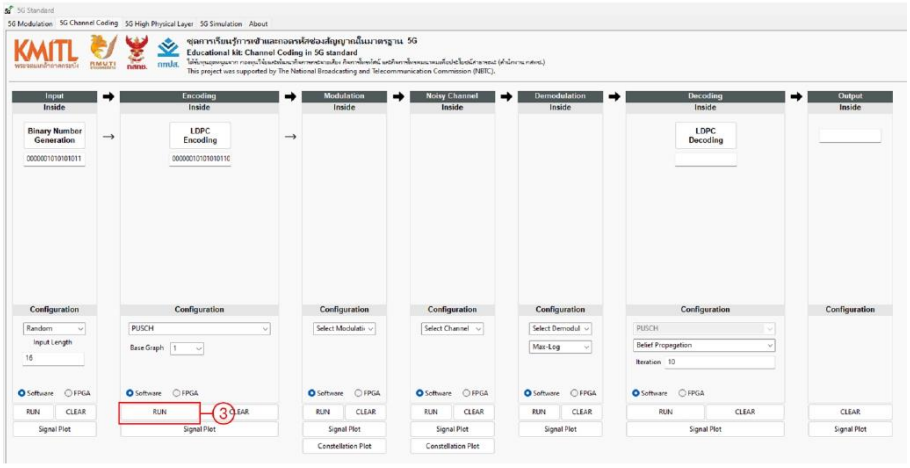
1) เลือกรูปแบบช่องสัญญาณ โดยคลิก Select Channel Encoder สำหรับรหัสแอลดีพีซีจะอยู่ในช่องสัญญาณ PUSCH และ PDSCH เท่านั้น



2) เลือก base graph

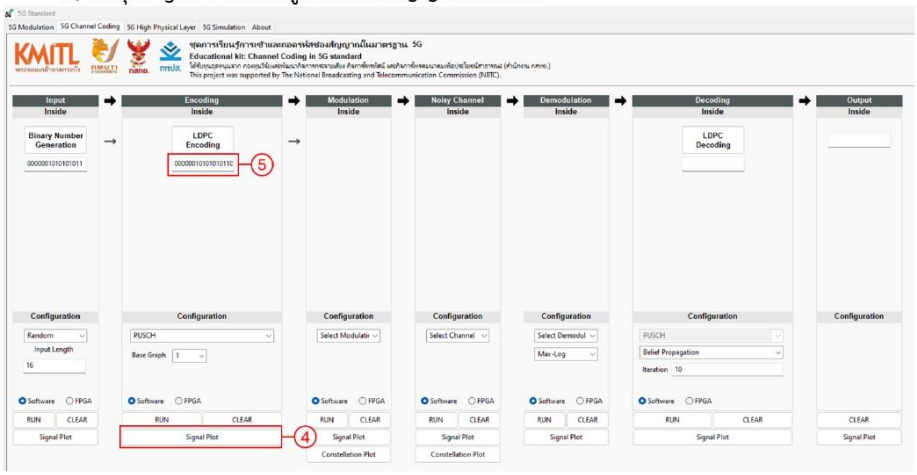


3) กดปุ่ม RUN เพื่อดำเนินการเข้ารหัส



5) ผลลัพธ์จะแสดงภายในส่วน Inside โดยผู้ใช้สามารถเลื่อนดูผลลัพธ์ทั้งหมด หรือคัดลอกข้อมูล

6) กดปุ่ม Signal Plot เพื่อดูกราฟของสัญญาณที่เข้ารหัส



การทดลองและบันทึกผล

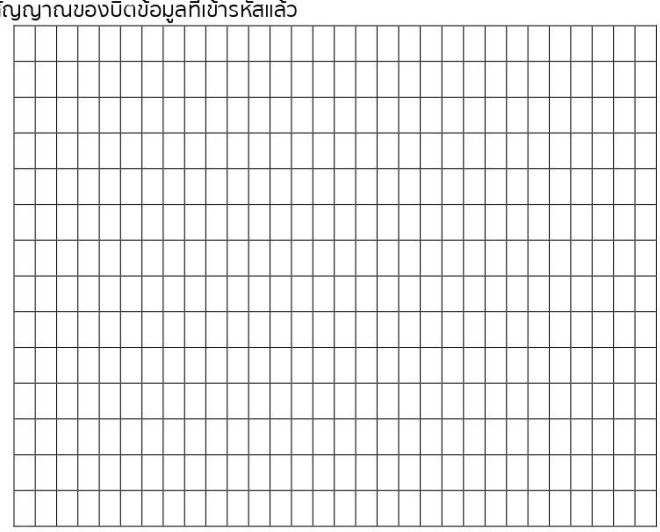
ให้นักศึกษาทดลองเข้ารหัสบิตข้อมูลอินพุตขนาด 16 บิต โดยกำหนดรูปแบบช่องสัญญาณแบบ PUSCH และใช้ Base graph 1 ให้วาดสัญญาณที่ถูกเข้ารหัสแล้ว

บิตข้อมูลก่อนเข้ารหัสและบิตข้อมูลหลังเข้ารหัส

.....

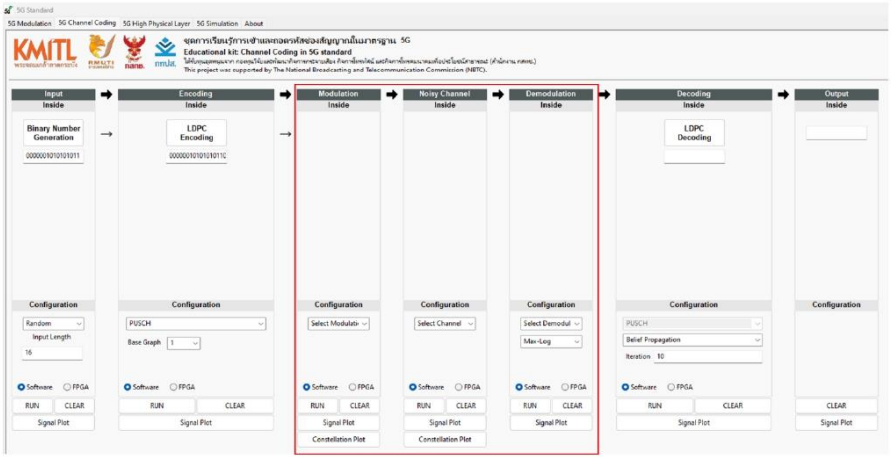
.....

กราฟสัญญาณของบิตข้อมูลที่เข้ารหัสแล้ว



<u>ส่วนที่ 3</u> การมอดูเลต ช่องสัญญาณ และการดีมอดูเลต

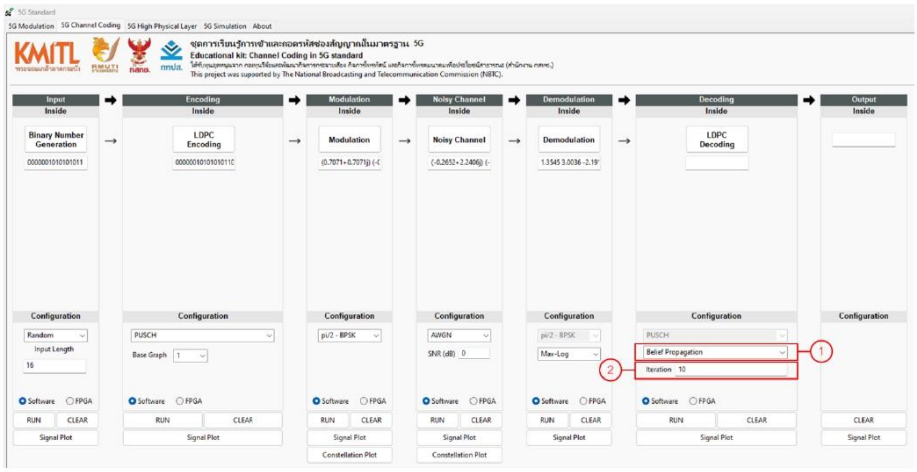
วิธีการทดลองเช่นเดียวกันกับในแบบฝึกหัดที่ 1



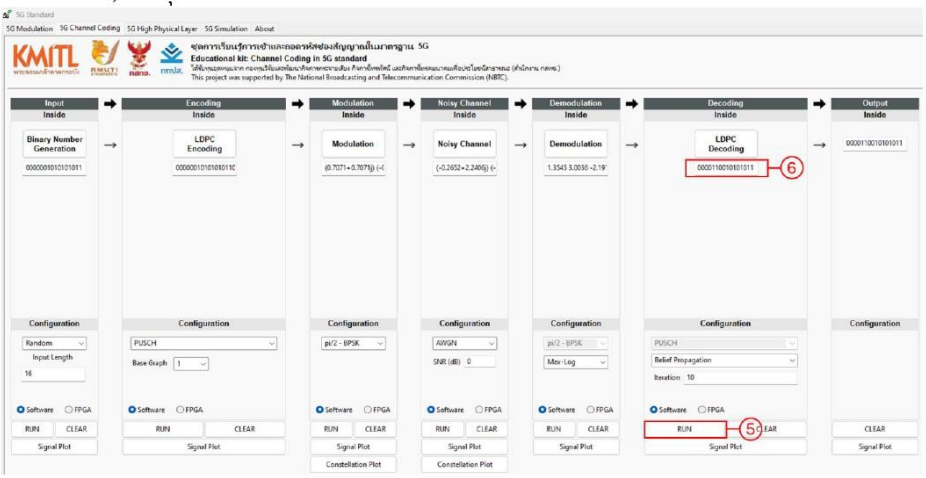
<u>ส่วนที่ 4</u> การถอดรหัสแอลดีพีซี

วิธีการทดลอง

1) เลือกอัลกอริทึมการถอดรหัส และกำหนดจำนวนรอบการถอดรหัส

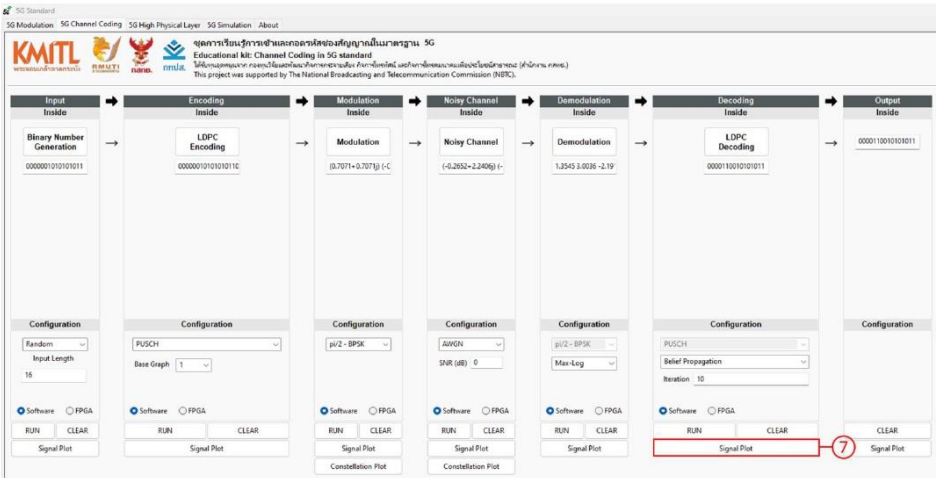


2) กดปุ่ม RUN เพื่อดำเนินการถอดรหัส



3) ผลลัพธ์จะแสดงภายในส่วน Inside โดยผู้ใช้สามารถเลื่อนดูผลลัพธ์ทั้งหมด หรือคัดลอกข้อมูล

4) กดปุ่ม Signal Plot เพื่อดูกราฟของบิตข้อมูลที่ได้จากการถอดรหัส



การทดลองและบันทึกผล

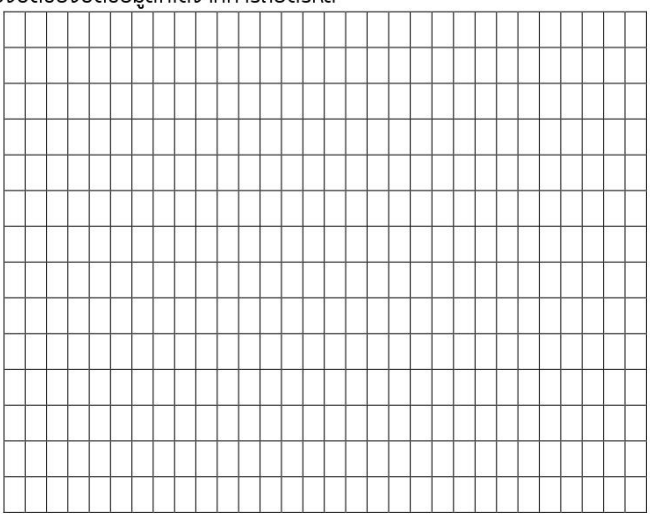
ให้นักศึกษาทดลองการถอดรหัสบิตข้อมูลอินพุตขนาด 16 บิต และมีรูปแบบช่องสัญญาณ PDSCH ใช้การมอดูเลตแบบ $\pi/2$ - BPSK โดยส่งผ่านช่องสัญญาณแบบ Fading ที่มีค่า SNR = 5dB และกำหนดให้ใช้อัลกอริทึมการดีมอดูเลตแบบ Mag-Log และถอดรหัสด้วยอัลกอริทึม Min-sum จำนวน 20 รอบ พร้อมทั้งวาดกราฟของบิตข้อมูลที่ได้จากการถอดรหัส

บิตข้อมูลที่ได้จากการถอดรหัส

.....

.....

กราฟของบิตของบิตข้อมูลที่ได้จากการถอดรหัส



แบบฝึกหัด

1) จงเข้ารหัส CRC ของชุดข้อมูลข่าวสาร $\mathbf{u}$ ที่เป็นเลขฐานสองจำนวน 10 บิตซึ่งถูกแปลงจากเลขฐานสิบของรหัสนักศึกษา 3 ตัวท้ายของนักศึกษา ตัวอย่างเช่น $620102222_{10} = 0011011110_2$ และใช้โพลีโนเมียลที่ถูกต้องตามมาตรฐานการเข้ารหัส LDPC ของมาตรฐาน 5G

2) เลขฐาน 2 จำนวน 10 บิตที่ได้จากข้อ 1. นำมาตรวจสอบคำตอบในโปรแกรมเข้ารหัสแอลดีพีซีตามมาตรฐาน 5G โดยกำหนดพารามิเตอร์ดังนี้ $R = 0.5$

2.1 เปรียบเทียบความถูกต้องของคำตอบที่ได้จากข้อ 1. และคำตอบที่ได้จากโปรแกรมการเข้ารหัส CRC

2.2 เปรียบเทียบข้อมูลก่อนเข้ารหัสและหลังเข้ารหัส CRC ว่ามีการเติมบิต Redundancy กี่บิตที่ตำแหน่งใดบ้าง

2.3 จากโปรแกรมพิจารณาว่าข้อมูลถูกเข้ารหัส CRC ครั้งที่ 2 หรือไม่เพราะเหตุใด

2.4 จากโปรแกรมพิจารณาว่าข้อมูลก่อนเข้ารหัสแอลดีพีซีมีความยาวกี่บิต มีการเติมบิต 0 เข้าไปในบล็อกข้อมูลหรือไม่ (ถ้ามีจำนวนกี่บิต) การเข้ารหัสรหัสใช้กราฟฐานใด

2.5 คำรหัสที่ได้จากการเข้ารหัสมีจำนวนกี่บิต พิจารณาว่าตำแหน่งของบิตข้อมูลอยู่ตำแหน่งใด และบิตพาริตีอยู่ในตำแหน่งใด

3) เข้ารหัสโพลาร์ความยาวคำรหัส $N = 16$ บิต ที่อัตรารหัส $R = 1$ โดยนำรหัสนักศึกษา 4 หลักท้ายของตนเอง (มองเป็นเลขฐานสิบหก) แปลงเป็นบิตข้อมูล (เลขฐานสอง) เช่น รหัส 6201137_{16} ได้เป็น 0001001101111000_2

4) ช่วยเช็คคำตอบโปรแกรมเข้ารหัสโพลาร์มาตรฐาน 5G หน่อย โดยลองนำบิตก่อนหน้าบล็อกการเข้ารหัสโพลาร์ไปทำการเข้ารหัส โดยให้เลือกการตั้งค่าดังนี้

- Physical Downlink Control Channel (PDCCH)
- บิตข้อมูลบล็อก ความยาว 16 บิต จากรหัสนักศึกษา 4 หลักท้าย เหมือนกับแบบฝึกหัดก่อนหน้า
- ความยาวคำรหัส 40 บิต
- บิต RNTI คือ 1010101010101010_2

โดยผลลัพธ์ควรได้คำรหัสแม่ของรหัสโพลาร์ความยาว $N = 64$ บิต หากโปรแกรมถูกต้อง โดยนำผลลัพธ์บิตความยาว $N = 64$ บิตนั้นเป็นคำตอบ

5) การมอดูเลตแบบ pi/2-BPSK ต่างจาก BPSK อย่างไร

.....

.....

.....

6) การมอดูเลตแบบ QPSK ต่างจาก 16-QAM อย่างไร

.....

.....

.....

7) กรอกข้อมูลในตารางให้สมบูรณ์ (ตารางการมอดูเลตในมาตรฐาน 5G)

การมอดูเลต	จำนวนบิตต่อสัญลักษณ์ (bits per symbol)
pi/2-BPSK	
QPSK	
16QAM	
64QAM	
256QAM	
1024QAM	

8) อธิบายความแตกต่างระหว่างช่องสัญญาณเกาส์เซียนและช่องสัญญาณเฟดดิง

.....

.....

.....

9) เพราะเหตุใดเราจึงใช้ช่องสัญญาณเฟดดิงแทนช่องสัญญาณการสื่อสารไร้สาย

.....

.....

.....

ภาคผนวก ค
เอกสารประกอบการเผยแพร่ในบริษัทเอกชน

เอกสารดังต่อไปนี้คือเอกสารที่ใช้ในการเผยแพร่แก่บริษัทเอกชนทั้ง 2 ดังที่กล่าวในหัวข้อที่ 4.9 โดยเนื้อหาที่เผยแพร่กล่าวถึงการใช้งานทรัพยากรของอุปกรณ์ FPGA ในแต่ละโมดูล

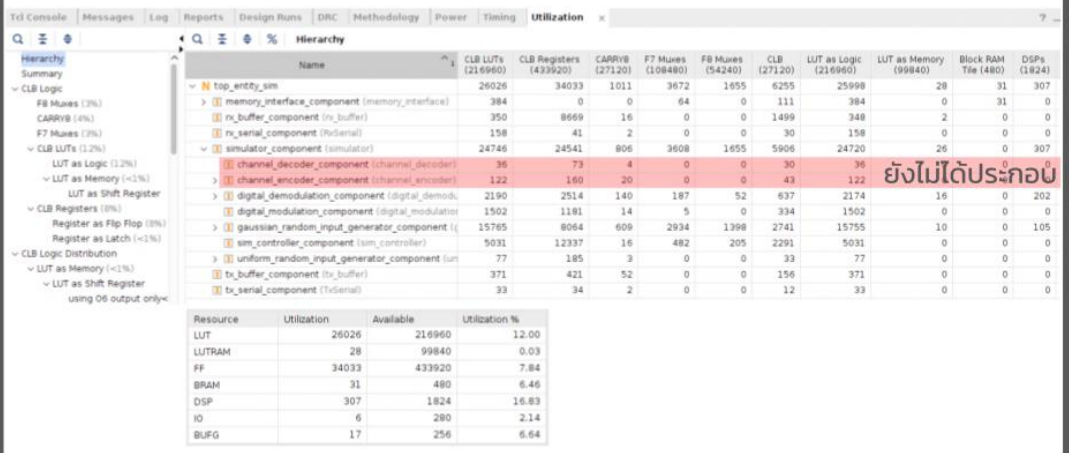
FPGA model

Board : AMD Kintex UltraScale+ FPGA KCU116 Evaluation Kit

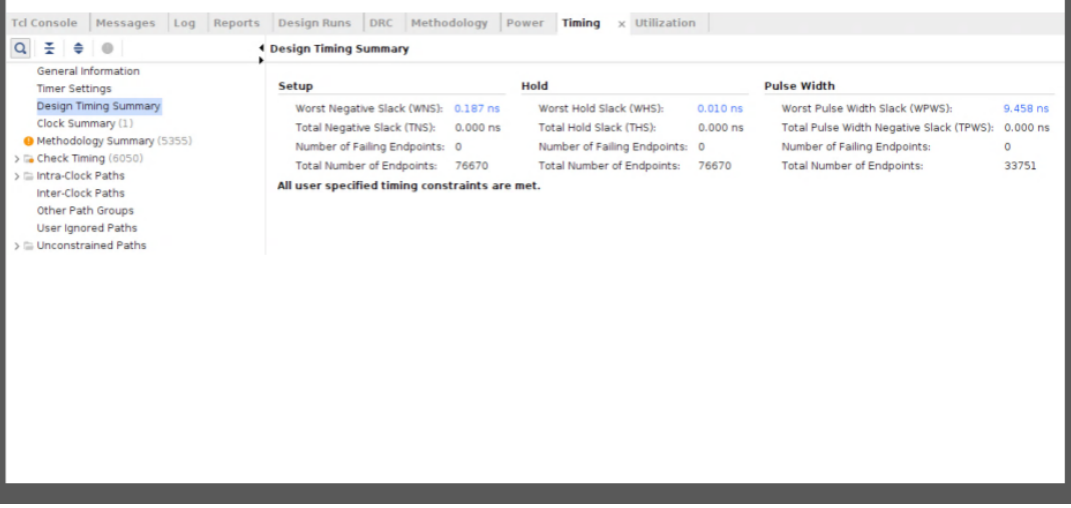
Chipset : xcku5p-ffvb676-2-e

Strategy : Vivado synthesis/implementation default

Simulator utillization



Simulator timing



Simulator

Implementation results

Resource element
- CLB LUT : 26,026
- CLB Register : 34,033

Frequency
- Setting constraint :
  - Clk period : 20 ns
  - Setup WNS : 0.187 ns
  - Hold WNS : 0.010 ns

Ports

```
entity top_entity_sim is
    port(
        rstSys                  : in std_logic;
        -- ClkSys               : in std_logic;
        clk125p                 : in std_logic;
        clk125n                 : in std_logic;

        button                  : in std_logic;
        -- buttonArray          : in std_logic_vector(3 downto 0);
        -- switchArray          : in std_logic_vector(7 downto 0);
        -- ledArray             : out std_logic_vector(7 downto 0);

        TxFfEmpty               : in std_logic;
        SerDataIn               : in std_logic;

        SerDataOut              : out std_logic;
        RxFfWrEn                : out std_logic
    );
end entity top_entity_sim;
```

Pseudorandom number generator

Synthesis results

Resource element
CLB LUT : 86
CLB Register : 192

Frequency
Setting constraint :
Clk period : 8 ns
Setup WNS : 6.289 ns
Hold WNS : -0.070 ns

Ports

```
entity uniform_random_input_generator is
    port(
        rst                             : in std_logic;
        clk                             : in std_logic;
        sim                             : in std_logic;

        sourceValidURNG                 : in std_logic;
        sourceReadyURNG                 : out std_logic;

        sinkReadyURNG                   : in std_logic;
        sinkValidURNG                   : out std_logic;

        ram_enable_a                    : out std_logic;
        ram_write_enable_a              : out std_logic;
        ram_address_a                   : out std_logic_vector(c_max_memory_address_depth_log - 1 downto 0);
        ram_data_in_a                   : out std_logic_vector(c_memory_data_width - 1 downto 0);
        ram_data_out_a                  : in std_logic_vector(c_memory_data_width - 1 downto 0);

        ram_enable_b                    : out std_logic;
        ram_write_enable_b              : out std_logic;
        ram_address_b                   : out std_logic_vector(c_max_memory_address_depth_log - 1 downto 0);
        ram_data_in_b                   : out std_logic_vector(c_memory_data_width - 1 downto 0);
        ram_data_out_b                  : in std_logic_vector(c_memory_data_width - 1 downto 0);

        uniformBitLength                : in std_logic_vector(15 downto 0)
        -- uniformRandomInputBits          : out std_logic_vector(8447 downto 0)
    );
end entity uniform_random_input_generator;
```

5G LDPC encoder

Synthesis results

Resource element
CLB LUT : 149,288
CLB Register : 60,884

Frequency
Setting constraint :
Clk period : 10 ns
Setup WNS : 0.044 ns
Hold WNS : -0.068 ns

Ports

```
entity encoder is
generic(    BITS    :   integer := 384;
            STAGES  :   integer := 9);
port(   Clk         :   in  std_logic;
        Rst         :   in  std_logic;
        start_encode:   in std_logic;
        z           :   in  integer range 2 to 384;
        hg          :   in  std_logic;
        i_LS        :   in  std_logic_vector(2 downto 0);
        data_0      :   in  std_logic_vector(383 downto 0);
        data_1      :   in  std_logic_vector(383 downto 0);
        data_2      :   in  std_logic_vector(383 downto 0);
        data_3      :   in  std_logic_vector(383 downto 0);
        data_4      :   in  std_logic_vector(383 downto 0);
        data_5      :   in  std_logic_vector(383 downto 0);
        data_6      :   in  std_logic_vector(383 downto 0);
        data_7      :   in  std_logic_vector(383 downto 0);
        data_8      :   in  std_logic_vector(383 downto 0);
        data_9      :   in  std_logic_vector(383 downto 0);
        data_10     :   in  std_logic_vector(383 downto 0);
        data_11     :   in  std_logic_vector(383 downto 0);
        data_12     :   in  std_logic_vector(383 downto 0);
        data_13     :   in  std_logic_vector(383 downto 0);
        data_14     :   in  std_logic_vector(383 downto 0);
        data_15     :   in  std_logic_vector(383 downto 0);
        data_16     :   in  std_logic_vector(383 downto 0);
        data_17     :   in  std_logic_vector(383 downto 0);
        data_18     :   in  std_logic_vector(383 downto 0);
        data_19     :   in  std_logic_vector(383 downto 0);
        data_20     :   in  std_logic_vector(383 downto 0);
        data_21     :   in  std_logic_vector(383 downto 0);
        parity_done : out std_logic;
        parity : out std_logic_vector(17663 downto 0)
    );
end entity encoder;
```

5G modulator

Synthesis results

Resource element
CLB LUT : 1520
CLB Register : 1178

Frequency
Setting constraint :
Clk period : 8 ns
Setup WNS : 6.104 ns
Hold WNS : -0.070 ns

Ports

```
entity digital_modulation is
    port(
        rst                         : in std_logic;
        clk                         : in std_logic;
        sim                         : in std_logic;

        sourceValidMod              : in std_logic;
        sourceReadyMod              : out std_logic;

        sinkReadyMod                : in std_logic;
        sinkValidMod                : out std_logic;

        ram_enable_a                : out std_logic_vector(1 downto 0);
        ram_write_enable_a          : out std_logic_vector(1 downto 0);
        ram_address_a               : out t_memory_address_depth_set(1 downto 0);
        ram_data_in_a               : out t_memory_data_width_set(1 downto 0);
        ram_data_out_a              : in t_memory_data_width_set(1 downto 0);

        ram_enable_b                : out std_logic_vector(1 downto 0);
        ram_write_enable_b          : out std_logic_vector(1 downto 0);
        ram_address_b               : out t_memory_address_depth_set(1 downto 0);
        ram_data_in_b               : out t_memory_data_width_set(1 downto 0);
        ram_data_out_b              : in t_memory_data_width_set(1 downto 0);

        modScheme                   : in std_logic_vector(3 downto 0);
        modBitLength                : in std_logic_vector(15 downto 0);
        -- modBitInput                  : in std_logic_vector(1023 downto 0);

        modValueBitLength           : out std_logic_vector(15 downto 0)
        -- modValue                 : out std_logic_vector(32767 downto 0)
    );
end entity digital_modulation;
```

Noisy channel generator

Synthesis results

Resource element
CLB LUT : 15,895
CLB Register : 7,974

Frequency
Setting constraint :
Clk period : 8 ns
Setup WNS : 3.631 ns
Hold WNS : -0.081 ns

Ports

```
entity gaussian_random_input_generator is
    port(
        rst                         : in std_logic;
        clk                         : in std_logic;
        sim                         : in std_logic;

        sourceValidGRNG             : in std_logic;
        sourceReadyGRNG             : out std_logic;

        sinkReadyGRNG               : in std_logic;
        sinkValidGRNG               : out std_logic;

        ram_enable_a                : out std_logic_vector(3 downto 0);
        ram_write_enable_a          : out std_logic_vector(3 downto 0);
        ram_address_a               : out t_memory_address_depth_set(3 downto 0);
        ram_data_in_a               : out t_memory_data_width_set(3 downto 0);
        ram_data_out_a              : in t_memory_data_width_set(3 downto 0);

        ram_enable_b                : out std_logic_vector(2 downto 0);
        ram_write_enable_b          : out std_logic_vector(2 downto 0);
        ram_address_b               : out t_memory_address_depth_set(2 downto 0);
        ram_data_in_b               : out t_memory_data_width_set(2 downto 0);
        ram_data_out_b              : in t_memory_data_width_set(2 downto 0);

        channelMode                 : in std_logic_vector(1 downto 0);
        sqrtVar                     : in std_logic_vector(15 downto 0);
        snr                         : in std_logic_vector(15 downto 0);
        gaussianSymbolLength        : in std_logic_vector(15 downto 0)
        -- gaussianValueInput           : in std_logic_vector(32767 downto 0);

        -- gaussianRandomSymbols        : out std_logic_vector(32767 downto 0);
        -- complexChannelCoefficient    : out std_logic_vector(16383 downto 0)
    );
end entity gaussian_random_input_generator;
```

5G demodulator

Synthesis results

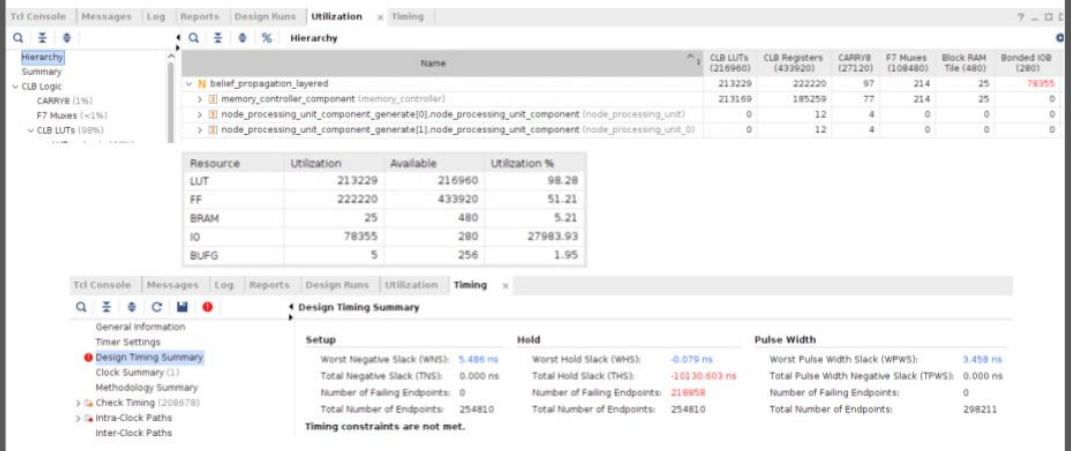
Resource element
CLB LUT : 2,236
CLB Register : 2,465

Frequency
Setting constraint :
Clk period : 8 ns
Setup WNS : 4.855 ns
Hold WNS : -0.070 ns

Ports

```
entity digital_demodulation is
    generic(
        data_width                  : integer := 32
    );
    port(
        rst                         : in std_logic;
        clk                         : in std_logic;
        sim                         : in std_logic;

        sourceValidDemod            : in std_logic;
        sourceReadyDemod            : out std_logic;

        sinkReadyDemod              : in std_logic;
        sinkValidDemod              : out std_logic;

        ram_enable_a                : out std_logic_vector(2 downto 0);
        ram_write_enable_a          : out std_logic_vector(2 downto 0);
        ram_address_a               : out t_memory_address_depth_set(2 downto 0);
        ram_data_in_a               : out t_memory_data_width_set(2 downto 0);
        ram_data_out_a              : in t_memory_data_width_set(2 downto 0);

        ram_enable_b                : out std_logic_vector(3 downto 0);
        ram_write_enable_b          : out std_logic_vector(3 downto 0);
        ram_address_b               : out t_memory_address_depth_set(3 downto 0);
        ram_data_in_b               : out t_memory_data_width_set(3 downto 0);
        ram_data_out_b              : in t_memory_data_width_set(3 downto 0);

        -- complexChannelCoefficient      : in std_logic_vector(16383 downto 0); --(5,11)
        modBitLength                : in std_logic_vector(15 downto 0);
        channelMode                 : in std_logic_vector(1 downto 0);
        InversubpTVar               : in std_logic_vector(15 downto 0);
        demodScheme                 : in std_logic_vector(3 downto 0);
        demodEquation               : in std_logic_vector(7 downto 0);
        demodSymbolLength           : in std_logic_vector(15 downto 0);
        -- demodValueInput             : in std_logic_vector(32767 downto 0);

        demodValueBitLength         : out std_logic_vector(15 downto 0)
        -- demodBitValue                  : out std_logic_vector(1823 downto 0);
        -- demodValue                     : out std_logic_vector(32767 downto 0);
        -- inverseComplexChannelCoefficient : out std_logic_vector(16383 downto 0)
    );
end entity digital_demodulation;
```

LDPC decoder utillization and timing



LDPC decoder

Synthesis results

Resource element
CLB LUT : 213,229
CLB Register : 222,220

Frequency
Setting constraint :
Clk period : 8 ns
Setup WNS : 5.486 ns
Hold WNS : -0.079 ns

Ports

ภาคผนวก ง
ผลการดำเนินงานฉบับย่อสำหรับตีพิมพ์ในวารสารสำนักงาน กสทช.

1. บทนำ

ปัจจุบัน ประเทศไทยมีอุตสาหกรรมด้านการพัฒนาอุปกรณ์โทรคมนาคมอย่างจำกัด เนื่องจากเทคโนโลยีที่มีความซับซ้อนและต้องการบุคลากรที่มีความสามารถสูง ซึ่งทำให้บริษัทโทรคมนาคมในประเทศต้องพึ่งพาอุปกรณ์จากต่างประเทศ อย่างไรก็ตาม สถานการณ์ปัจจุบันที่บริษัทผู้ผลิตอุปกรณ์จากต่างประเทศเริ่มเข้ามาให้บริการเอง ทำให้การพัฒนาอุปกรณ์ในประเทศมีความสำคัญมากขึ้น จากปัญหาดังกล่าว โครงการ "พัฒนาอุปกรณ์เข้ารหัสช่องสัญญาณมาตรฐาน 5G สำหรับสื่อการสอนและการต่อยอดเชิงพาณิชย์" จึงถูกจัดตั้งขึ้น โดยเน้นการพัฒนาซอฟต์แวร์และฮาร์ดแวร์การถอดรหัสช่องสัญญาณมาตรฐาน 5G และการพัฒนาชุดการเรียนรู้การถอดรหัสช่องสัญญาณในมาตรฐาน 5G รวมถึงการ์ดเร่งความเร็ว FEC สำหรับโครงข่าย O-RAN

โครงการนี้มีวัตถุประสงค์ที่จะพัฒนาซอฟต์แวร์และฮาร์ดแวร์การถอดรหัสช่องสัญญาณมาตรฐาน 5G พัฒนาชุดการเรียนรู้การถอดรหัสช่องสัญญาณในมาตรฐาน 5G พัฒนาชุดทดสอบสมรรถนะการเข้ารหัสและถอดรหัสช่องสัญญาณในระบบสื่อสารไร้สาย และพัฒนาการ์ดเร่งความเร็ว FEC สำหรับรหัสช่องสัญญาณในโครงข่าย O-RAN

โครงการนี้ประกอบด้วย 5 ส่วน คือ การพัฒนาการมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G, ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G, ชุดทดสอบสมรรถนะการมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G, การพัฒนาการ์ดเร่งความเร็ว FEC และการเผยแพร่ชุดการเรียนรู้และการ์ดเร่งความเร็ว FEC ให้แก่มหาวิทยาลัยและบริษัทเอกชน ประโยชน์ที่คาดว่าจะได้รับจากโครงการนี้คือ มหาวิทยาลัยที่จัดการเรียนการสอนด้านวิศวกรรมโทรคมนาคมจะได้รับชุดการเรียนรู้ที่ทันสมัยและตรงตามมาตรฐาน 5G ผู้ประกอบการด้านอุปกรณ์การสอนจะได้เห็นแนวทางการพัฒนาอุปกรณ์การสอนที่สอดคล้องกับความต้องการของมหาวิทยาลัย และผู้ประกอบการด้านอุปกรณ์โทรคมนาคมจะได้เห็นแนวทางการพัฒนาการ์ดเร่งความเร็วที่ใช้งานในโครงข่าย O-RAN

2. ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

2.1 พื้นฐานสัญญาณสุ่มและช่องสัญญาณสื่อสาร

การแจกแจงแบบสม่ำเสมอ (Uniform Distribution) คือการแจกแจงที่ทุกค่าภายในช่วงที่กำหนดมีความน่าจะเป็นเท่ากัน การแจกแจงแบบสม่ำเสมอมีการใช้งานในหลายด้าน เช่น การสร้างเลขสุ่มและการจำลองเหตุการณ์สุ่มในงานวิจัยและคอมพิวเตอร์ การแจกแจงแบบไม่สม่ำเสมอ (Nonuniform Distribution) คือการแจกแจงที่ค่าความน่าจะเป็นของผลลัพธ์ต่าง ๆ ภายในช่วงที่กำหนดไม่เท่ากัน โดยแต่ละค่าอาจมีความน่าจะเป็นที่แตกต่างกันไปตามลักษณะของการแจกแจง เช่น การแจกแจงแบบปกติ (Normal Distribution) ที่ค่ากลางมีความน่าจะเป็นสูงกว่าค่าที่อยู่ห่างจากค่ากลาง

2.2 พื้นฐานรหัสช่องสัญญาณ

รหัส LDPC (Low-Density Parity-Check) คือรหัสแก้ไขความผิดพลาดประเภทรหัสบล็อกเชิงเส้นที่เมทริกซ์พาริตีเช็กมีจำนวน ‘1’ น้อยมาก ๆ และความสามารถในการแก้ไขข้อผิดพลาดที่มีประสิทธิภาพสูง สามารถเข้าใกล้ความจุของช่องสัญญาณได้อย่างมาก LDPC ใช้อัลกอริธึมการถอดรหัสแบบผ่านข้อความ (Message Passing) ซึ่งทำให้สามารถลดจำนวนข้อผิดพลาดได้ดี จึงถูกนำไปใช้ในเทคโนโลยีสื่อสารและการจัดเก็บข้อมูลหลากหลายประเภท เช่น Wi-Fi, DVB-S2 และ 5G

รหัส Polar เป็นรหัสแก้ไขข้อผิดพลาดที่มีสมรรถนะเข้าใกล้ขีดจำกัดของช่องสัญญาณ (Channel Capacity) ตามทฤษฎีของ Shannon โดยใช้กระบวนการ Polarization เพื่อแยกช่องสัญญาณออกเป็นช่องสัญญาณที่มีความน่าเชื่อถือสูงและต่ำ ข้อมูลจะถูกส่งผ่านช่องสัญญาณที่มีความน่าเชื่อถือสูงเท่านั้น ทำให้มีประสิทธิภาพสูง รหัส Polar ได้รับการใช้งานในมาตรฐานการสื่อสารไร้สาย 5G เนื่องจากมีโครงสร้างที่ง่ายและมีประสิทธิภาพในการถอดรหัสสูง

2.3 การมอดูเลชันมาตรฐาน 5G

ในมาตรฐาน 5G การมอดูเลชัน (Modulation) ถูกใช้เพื่อแปลงข้อมูลดิจิทัลเป็นสัญญาณที่สามารถส่งผ่านช่องสัญญาณไร้สายได้ เทคนิคการมอดูเลชันที่ใช้ใน 5G ได้แก่ BPSK (Binary Phase Shift Keying), **π**/2-BPSK, QPSK (Quadrature Phase Shift Keying), 16-QAM (Quadrature Amplitude Modulation), 64-QAM, 256-QAM และ 1024-QAM ซึ่งช่วยเพิ่มอัตราการส่งข้อมูลโดยการเพิ่มจำนวนบิตที่สามารถส่งผ่านได้ต่อสัญลักษณ์ โดยแต่ละเทคนิคมีการเทรดออฟระหว่างความซับซ้อนในการส่ง และความทนทานต่อสัญญาณรบกวนเพื่อให้เหมาะสมกับสภาพแวดล้อมและความต้องการในการสื่อสาร

2.4 กระบวนทำงานฟิสิคัลเลเยอร์ลำดับสูง

กระบวนการทำงานทำงานฟิสิคัลเลเยอร์ลำดับสูง กรณี PDSCH/PUSCH/PSSCH ประกอบด้วยโมดูลที่สำคัญทั้งหมด 7 ส่วน ได้แก่ 1) การเข้ารหัส CRC, 2) การแบ่งย่อยบล็อกข้อมูล, 3) การเข้ารหัส CRC ของบล็อกรหัส, 4) การเข้ารหัส LDPC, 5) การปรับอัตรารหัส, 6) การแทรกสลับบิต และ 7) การต่อเรียงบล็อกรหัส

กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูง กรณี PUCCH/PDCH/PBCH/PSCCH/PSFCH/PSBCH บนอุปกรณ์ FPGA จะถูกแบ่งเป็นโมดูลหลัก 5 โมดูล ซึ่งจะประกอบไปด้วยโมดูลย่อยทั้งหมด 10 โมดูล ได้แก่ 1) การแบ่งย่อยบล็อกรหัส, 2) การเข้ารหัส การสแครมบลิง และการแทรกสลับ CRC, 3) การลำดับช่องสัญญาณย่อยและการคำนวณบิตพาริตี, 4) การเข้ารหัสโพลาร์ การแทรกสลับบล็อกย่อย และการปรับอัตรารหัส และ 5) การแทรกสลับบิตรหัสและการต่อบล็อกรหัส

3. ระเบียนวิธีวิจัย

3.1 การออกแบบชุดการเรียนรู้

โครงการนี้ออกแบบซอฟต์แวร์ GUI ชุดการเรียนรู้การมอดูเลชันและการดีมอดูเลชันมาตรฐาน 5G โดยใช้ภาษา Python เพื่อให้ผู้ใช้งานสามารถเรียนรู้กระบวนการมอดูเลชันและ ดีมอดูเลชันได้โดยง่าย ซึ่งผู้ใช้งานจะสามารถเห็นกระบวนการทำงานตั้งแต่การสร้างข้อมูลแบบสุ่มหรือ

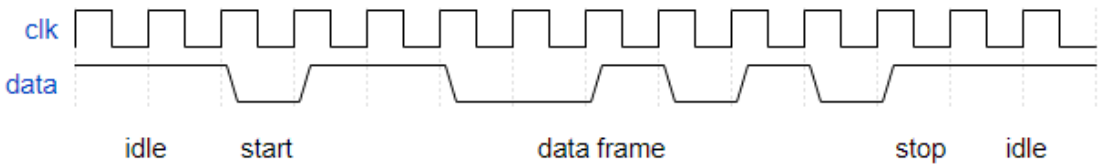
นำเข้าข้อมูลอินพุต การมอดูเลชัน การส่งผ่านช่องสัญญาณรบกวน และการดีมอดูเลชัน นอกจากนี้ ผู้ใช้งานสามารถดูผลลัพธ์ ณ จุดต่าง ๆ รวมถึงสามารถเรียกดูกราฟผลลัพธ์ของสัญญาณและแผนภาพคอนสเตลเลชันของการมอดูเลชันได้

3.2 การออกแบบชุดทดสอบสมรรถนะ

โครงการนี้ออกแบบชุดทดสอบสมรรถนะการมอดูเลชันและดีมอดูเลชันในระบบสื่อสารไร้สาย โดยผู้ใช้งานสามารถเลือกช่องสัญญาณสื่อสารแบบ AWGN Channel หรือ Fading Channel เพื่อทดลองหาสมรรถนะอัตราบิตผิดพลาด ณ จุดต่าง ๆ ของระบบได้ ซึ่งการออกแบบชุดทดสอบสมรรถนะการมอดูเลชันและดีมอดูเลชันในระบบสื่อสารไร้สายประกอบด้วย 3 ส่วน ได้แก่ ส่วนแสดงพารามิเตอร์ของการมอดูเลชันและดีมอดูเลชัน ส่วนกำหนดค่าการทดสอบสมรรถนะ และส่วนแสดงผลสมรรถนะ

3.3 การออกแบบการเชื่อมต่อระหว่างอุปกรณ์ FPGA กับคอมพิวเตอร์

ชุดการเรียนรู้และชุดทดสอบสมรรถนะได้รับการออกแบบให้ประมวลผลระบบสื่อสารมาตรฐาน 5G ทั้งบนคอมพิวเตอร์และอุปกรณ์ FPGA โดยการส่งข้อมูลจากคอมพิวเตอร์ไปยัง FPGA ผ่านช่องทางข้อมูลอนุกรม (serial port) ตามโพรโทคอล UART ข้อมูลจะถูกส่งในชุดละ 8 บิต พร้อมบิตเริ่มต้น (start bit) และบิตหยุด (stop bit) ตามรูปที่ 1



รูปที่ 1 ชุดบิตข้อมูลการสื่อสารภายใต้โพรโทคอล UART

3.4 การออกแบบการ์ดเร่งความเร็ว FEC

มาตรฐาน O-RAN กำหนดการใช้งานการ์ดเร่งความเร็ว FEC แบบ PDSCH_FEC สำหรับ Downlink โดยทำกระบวนการดังนี้:

1) TB CRV attachment
2) CB segmentation + CB CRV attachment
3) LDPC encoding
4) Rate matching
5) CB concatenation

สำหรับ Uplink มาตรฐาน O-RAN กำหนดการใช้งานการ์ดเร่งความเร็ว FEC แบบ PUSCH_FEC โดยทำกระบวนการดังนี้:

1) TB CRC check
2) CB CRC + CB desegmentation
3) LDPC decoding

4) Rate dematching
5) CB deconcatenation

3.5 การออกแบบการเชื่อมต่อระหว่างการ์ดเร่งความเร็ว FEC กับคอมพิวเตอร์

การออกแบบการ์ดเร่งความเร็ว FEC แบบ AAL_PDSCH_FEC ด้วย FPGA ใช้ภาษา VHDL พัฒนาบนอุปกรณ์ AMD Virtex UltraScale+ FPGA VCU118 Evaluation Kit ซึ่งมีส่วนต่อประสาน PCIe การออกแบบเชื่อมกับ IP-Core DMA/Bridge Subsystem for PCI Express ของ Xilinx เพื่อจัดการเฟรมข้อมูลผ่าน PCIe โปรแกรมภาษา Python ใช้เรียก Driver ของ AMD ที่เขียนด้วยภาษา C ทำให้สามารถทดสอบการรับส่งข้อมูลระหว่าง FPGA และคอมพิวเตอร์ผ่าน PCIe

การออกแบบการ์ดเร่งความเร็ว FEC แบบ AAL_PUSCH_FEC ด้วย FPGA ทำในลักษณะเดียวกับการ์ด AAL_PDSCH_FEC ที่กล่าวก่อนหน้า โดยปรับกระบวนการภายในให้ตรงตามมาตรฐาน เมื่อการ์ด AAL_PUSCH_FEC ได้รับข้อมูลจากระบบปฏิบัติการ จะดำเนินการตามขั้นตอนที่กำหนด

3.6 รายละเอียดการสร้างข้อมูลอินพุตแบบสุ่ม

สำหรับการออกแบบวงจรสุ่มข้อมูลอินพุตบนอุปกรณ์ FPGA ได้เลือกใช้วงจร CTG 3 ส่วนประกอบ ความยาว 32 บิต มีขนาดคาบเท่ากับ 2^{88} ข้อมูลสุ่มขนาด 32 บิตนี้จะถูกนำไปใช้กับระบบการสุ่มข้อมูลอินพุต อีกทั้งยังถูกนำไปใช้กับระบบการออกแบบสัญญาณรบกวนแบบเกาสเซียนเช่นเดียวกัน โดยสำหรับระบบการสุ่มข้อมูลข้อมูลอินพุตข้อมูลที่สุ่มได้จะถูกใช้งานเป็นข้อมูลอินพุตโดยตรง ขณะที่ระบบการออกแบบสัญญาณรบกวนแบบเกาส์เซียนจะถูกนำไปใช้เป็นอินพุตค่าทศนิยมขนาด [0,1)

3.7 รายละเอียดการสร้างช่องสัญญาณสื่อสารแบบ AWGN

การสุ่มสัญญาณรบกวนเกาส์เซียนด้วยการแปลงบ็อกซ์-มูลเลอร์ (Box-Muller transform) จะให้ผลลัพธ์เป็นคู่เลขสุ่มที่มีการกระจายเกาส์เซียน มีค่าเฉลี่ย 0 และความแปรปรวน 1 ในช่วง [-1, 1] โดยใช้คู่อินพุตที่มีการกระจายแบบสม่ำเสมอในช่วง [0, 1)

การออกแบบวงจรสร้างสัญญาณรบกวนเกาส์เซียนขาวบน FPGA ใช้การแปลงบ็อกซ์-มูลเลอร์ มีขั้นตอนดังนี้:

1) การสุ่มเลขทศนิยม [0, 1) ที่มีการกระจายแบบสม่ำเสมอ
2) การคำนวณฟังก์ชันลอการิทึมธรรมชาติด้วยการประมาณค่าด้วยพหุนามดีกรี 4
3) การคำนวณค่ารากที่สองด้วยอัลกอริทึม CORDIC ในโหมดเวกเตอร์ไฮเปอร์โบลิก
4) การคำนวณฟังก์ชันโคซายน์และซายน์โดยใช้ตารางค้นหา (lookup table) ที่คำนวณล่วงหน้าด้วย MATLAB
5) ผลลัพธ์ที่ได้คือสัญญาณรบกวนเกาส์เซียนขาวที่มีค่าเฉลี่ย 0 และค่าความแปรปรวน 1

3.8 รายละเอียดการสร้างช่องสัญญาณสื่อสารแบบ Fading

การออกแบบสัญญาณการเฟดบนอุปกรณ์ FPGA ค่าสัมประสิทธิ์ช่องสัญญาณเป็นค่าสัมบูรณ์คำนวณจากค่าสัมประสิทธิ์เชิงซ้อนแกนจริงและแกนจินตภาพ มี 2 ขั้นตอนหลัก:

การยกกำลังสอง: ทำได้โดยการคูณค่าที่ต้องการยกกำลังด้วยตัวมันเอง ผลลัพธ์จะมีความยาวเท่ากับความยาวของตัวคูณสองพจน์รวมกัน

การคำนวณรากที่สอง: ใช้วงจร CORDIC ในโหมดเวกเตอร์ไฮเปอร์โบลิก (Hyperbolic vectoring) โดยใช้อินพุตและเอาต์พุตเป็นค่า fixed point ความละเอียด fi(16, 11)

3.9 รายละเอียดการสร้างการมอดูเลชันมาตรฐาน 5G

ระบบการพัฒนาการมอดูเลตมาตรฐาน 5G บนอุปกรณ์ FPGA เป็นการสร้างวงจรการประมวลผลสัญญาณดิจิทัล (Digital signal processing: DSP) เป็นหลัก ทำให้การมอดูเลตสัญญาณตามมาตรฐาน 5G สามารถทำได้โดยการบวก ลบ และ คูณ ทั้งนี้ไม่นิยมการหารเนื่องจากวงจรจะมีความซับซ้อนสูง จึงมักใช้วิธีการอื่น เช่น การคำนวณค่าผลหารและบันทึกผลลัพธ์ไว้ใน ROM หรือ RAM ของอุปกรณ์ FPGA

การมอดูเลชันแบบ BPSK (Binary phase shift keying) การมอดูเลชันแบบ $\pi/2$-BPSK การมอดูเลชันแบบ QPSK (Quadrature phase shift keying) การมอดูเลชันแบบ 16QAM (Quadrature amplitude modulation) การมอดูเลชันแบบ 64QAM การมอดูเลชันแบบ 256QAM และการมอดูเลชันแบบ 1024QAM มีลักษณะคล้ายกัน โดยสามารถพิจารณาเป็นการมอดูเลชันแบบ PAM (Pulse amplitude modulation) ที่มีค่าบนแกนจริง (In-phase) และแกนจินตภาพ (Quadrature-phase) แตกต่างกันที่จำนวนระดับของ PAM และค่านอมัลไลซ์ ดังนั้น การสร้างวงจรมอดูเลตมาตรฐาน 5G จะเป็นเพียงการสร้าง look-up table หรือ LUT เพื่อเก็บค่าการคูณและการหารของค่าในแกนจริงและแกนจินตภาพไว้ เนื่องจากค่าในแกนจริงและแกนจินตภาพเป็นค่าเดียวกัน ดังนั้นจริงสามารถลดตารางให้เหลือเพียงตารางของค่าในแกนจริงเท่านั้น

3.10 รายละเอียดการสร้างการดีมอดูเลชันมาตรฐาน 5G

การออกแบบการดีมอดูเลตมาตรฐาน 5G บนอุปกรณ์ FPGA สามารถคำนวณได้จาก

$$L_{b_i}(y) = \frac{1}{2\sigma^2}\left(2y(\mu_1 - \mu_0) + (\mu_0)^2 - (\mu_1)^2\right)$$

เมื่อ μ_1 และ μ_0 คือค่าในตารางของบิต ‘0’ และบิต ‘1’

3.11 รายละเอียดการสร้างการเข้ารหัสช่องสัญญาณมาตรฐาน 5G

การออกแบบการเข้ารหัส LDPC บนอุปกรณ์ FPGA ใช้การคูณและการบวกเป็นหลัก โดยการบวกใช้ลอจิก XOR แทนการบวกแบบมอดูโล 2 ซึ่งประหยัดทรัพยากร ส่วนการคูณเมทริกซ์ถูกแทนที่ด้วยการเลื่อนค่าตามคุณสมบัติของ QC-LDPC ในมาตรฐาน 5G การเลื่อนค่าใช้วงจรเลื่อนค่าที่มีอยู่มากกว่าวงจรคูณบน FPGA จึงประหยัดทรัพยากรมากกว่า

การเข้ารหัสโพลาร์อาศัยการดำเนินการแบบต่อเนื่อง (recursive) โดยใช้การดำเนินการแบบพื้นฐาน (Basic operations) เช่น การ XOR และการคำนวณค่าใหม่ตามขั้นตอนการเข้ารหัสและใช้เครือข่ายการแปลงสัญญาณ (Butterfly network) เพื่อดำเนินการคำนวณที่ซับซ้อนให้เป็นการดำเนินการที่ง่ายและรวดเร็ว ค่าฟังก์ชัน F และ G เป็นฟังก์ชันพื้นฐานในการเข้ารหัสโพลาร์ โดยฟังก์ชันเหล่านี้ถูกคำนวณในรูปแบบที่เหมาะสมกับการประมวลผลแบบขนานบน FPGA และใช้ความสามารถในการประมวลผลแบบขนานของ FPGA เพื่อเพิ่มประสิทธิภาพในการคำนวณ โดยสามารถแบ่งงานออกเป็นหลายส่วนและดำเนินการพร้อมกัน

3.12 รายละเอียดการสร้างการถอดรหัสช่องสัญญาณมาตรฐาน 5G

การออกแบบการถอดรหัส LDPC บนชิป FPGA ใช้วิธีการถอดรหัสแบบเลเยอร์ (Layered Belief Propagation) โดยรวมโครงสร้างเป็นไปตามรูปแบบของตัวถอดรหัส LDPC ที่มีอยู่ในวรรณกรรมที่เกี่ยวข้อง โดยมีส่วนประกอบหลัก ได้แก่:

ส่วนหน่วยความจำ: ประกอบด้วยหน่วยความจำที่ใช้เก็บค่าเมทริกซ์ตรวจสอบพาริตี, ค่า LLR ที่เอาต์พุตช่องสัญญาณ, และค่า LLR ที่เอาต์พุตโหนดตรวจสอบ เพื่อการถอดรหัสแบบเลเยอร์ ส่วนนี้ใช้ทรัพยากรแรมในชิป FPGA โดยอย่างคุ้มค่า

ส่วนโครงสร้างการแทรกสลับ: เป็นโครงสร้างที่ตั้งอยู่ระหว่างหน่วยความจำและหน่วยประมวลผล ใช้ในการจัดเรียงข้อมูลจากหน่วยความจำไปยังหน่วยประมวลผล ด้วยวงจรย่อย QSN และวงจรแทรกสลับ เพื่อให้การทำงานเป็นไปอย่างมีประสิทธิภาพ

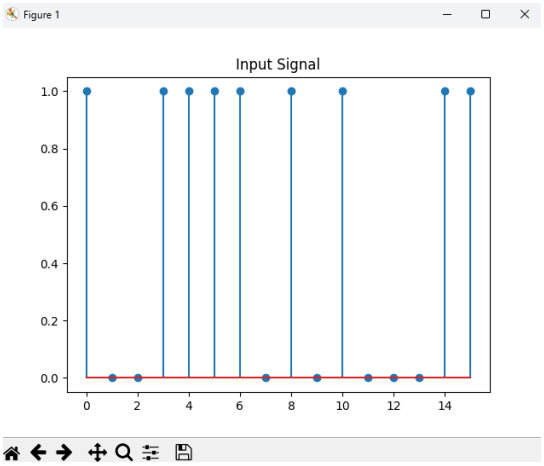
ส่วนหน่วยประมวลผล: ประกอบด้วยโหนดตรวจสอบและโหนดตัวแปร โดยใช้อัลกอริทึม Min-sum ในการถอดรหัส โดยคำนวณค่า LLR ที่เอาต์พุตโหนดตรวจสอบใหม่ และค่า LLR ที่เอาต์พุตโหนดตัวแปร เพื่อให้ได้ค่า LLR ที่เอาต์พุตช่องสัญญาณใหม่

การถอดรหัส LDPC บน FPGA ต้องออกแบบอย่างรอบคอบเพื่อให้การทำงานเป็นไปอย่างมีประสิทธิภาพและใช้ทรัพยากรในชิปอย่างคุ้มค่า

4. ผลการวิจัยและการวิจารณ์ผล

4.1 ผลการทดสอบการทำงานของซอฟต์แวร์ GUI

เมื่อผู้ใช้เปิดซอฟต์แวร์ GUI ที่หน้าต่างชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันในมาตรฐาน 5G หรือหน้าต่างชุดการเรียนรู้การเข้ารหัสและถอดรหัสมาตรฐาน 5G หรือหน้าต่างชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูง คอลัมน์ที่ 1 จะแสดงส่วนสำหรับสร้างข้อมูลอินพุต ผู้ใช้สามารถเลือกสร้างข้อมูลอินพุตแบบสุ่ม อัปโหลดรูปภาพ หรือป้อนข้อมูลอินพุตด้วยตนเอง ผลลัพธ์จะปรากฏในรูปบิตข้อมูลไบนารีและสามารถแสดงกราฟของสัญญาณอินพุตตามตัวอย่างในรูปที่ 2



รูปที่ 2 กราฟของสัญญาณข้อมูลอินพุตแบบสุ่มที่สร้างโดยใช้ซอฟต์แวร์

4.2 ผลการทดสอบการทำงานของซอฟต์แวร์ Module

โครงการนี้สร้างซอฟต์แวร์ Module ภาษา Python สำหรับชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G ชุดการเรียนรู้การเข้ารหัสและถอดรหัสช่องสัญญาณมาตรฐาน 5G และชุดการเรียนรู้กระบวนการทำงานฟิสิคัลเลเยอร์ลำดับสูงมาตรฐาน 5G รวมทั้งสิ้น 9 Module ดังตารางที่ 1 ซึ่งซอฟต์แวร์ Module เหล่านี้จะทำงานร่วมกับซอฟต์แวร์ GUI

ตารางที่ 1 ซอฟต์แวร์ Module สำหรับซอฟต์แวร์ GUI

ซอฟต์แวร์ Module	ซอฟต์แวร์ Module ย่อย	อินพุต	เอาต์พุต
randomInput.py	-	1. จำนวนบิตอินพุต	1. บิตข้อมูลแบบไบนารี
PolarEncoderModule.py	CodeBlockSegmentation CRCInterleaver CRCAttachment CRCScrambling PolarSequencing ParityCheckEncoding PolarEncoding SubBlockInterleaver RateMatching CodedBitInterleaver CodeBlockConcatenation	1. ชนิดของช่องสัญญาณกายภาพ 2. ความยาวรหัส 3. บิตข้อมูลแบบไบนารี	1. คำรหัสแบบไบนารี
LDPCEncoderModule.py	CRCEncoder1 Segmentation CRCEncoder2 LDPCEncoder RateMatching	1. ชนิดของช่องสัญญาณกายภาพ 2. พารามิเตอร์การเข้ารหัส 3. บิตข้อมูลแบบไบนารี	1. คำรหัสแบบไบนารี

ซอฟต์แวร์ Module	ซอฟต์แวร์ Module ย่อย	อินพุต	เอาต์พุต
	BitInterleaving CodeBlockConcatenation		
nrPDSCHPRBS.py	Scrambling Descrambling	1. คำรหัสแบบไบนารี หรือค่า LLR	1. สัญญาณที่ถูกสแครม หรือดีสแครม
modulation.py	pi2BPSK BPSK QPSK 16QAM 64QAM 256QAM 1024QAM	1. รูปแบบการมอดูเลชัน 2. บิตข้อมูลแบบไบนารี หรือคำรหัสแบบบารี หรือสัญญาณที่ถูกสแครม	1. สัญญาณมอดูเลชัน
Channel.py	AWGN Fading	1. รูปแบบช่องสัญญาณ 2. สัญญาณมอดูเลชัน 3. ค่า SNR	1. สัญญาณที่ผ่านช่อง 2. สัญญาณสื่อสาร
demodulation.py demodulation_LUT.py	pi2BPSK BPSK QPSK 16QAM 64QAM 256QAM 1024QAM	1. อัลกอริทึมการดีมอดูเลชัน 2. รูปแบบการดีมอดูเลชัน 3. สัญญาณที่ผ่านช่องสัญญาณสื่อสาร	1. ค่า LLR
PolarDecoderModule.py	TraverseIndexesCalculation CodedBitDeInterleaver PrePolarDecoder SCBased5GDecoder PolarBPDecoder	1. ชนิดของช่องสัญญาณกายภาพ 2. ค่า LLR หรือสัญญาณที่ได้จากการดีสแครม 3. อัลกอริทึมการถอดรหัส 4. พารามิเตอร์การถอดรหัส	1. บิตข้อมูลเอาต์พุตแบบไบนารี
LDPCDecoderModule.py	DeCodeBlockConcatenation DeBitInterleaving DeRateMatching LDPCDecoding CRCDecoder2 DeSegmentation CRCDecoder1	1. ชนิดของช่องสัญญาณกายภาพ 2. ค่า LLR หรือสัญญาณที่ได้จากการดีสแครม 3. อัลกอริทึมการถอดรหัส 4. พารามิเตอร์การถอดรหัส	1. บิตข้อมูลเอาต์พุตแบบไบนารี

4.3 ผลการทดสอบการทำงานของอุปกรณ์ FPGA

โครงการนี้ได้เขียนซอฟต์แวร์สำหรับการสร้างวงจรบนชิปเอฟพีจีเอ ภายใต้ภาษา VHDL ที่ใช้งานร่วมกับซอฟต์แวร์ GUI ชุดการเรียนรู้การมอดูเลชันและดีมอดูเลชันมาตรฐาน 5G รวมทั้งสิ้น 9 ระบบย่อย โดยแสดง ซึ่งซอฟต์แวร์เหล่านี้จะถูกสังเคราะห์ เชื่อมโยง และโปรแกรมไปยังชิปเอฟพีจีเอ และชิปเอฟพีจีเอจะถูกเชื่อมเข้ากับคอมพิวเตอร์ผ่านการสื่อสารแบบ serial เพื่อทำงานร่วมกับซอฟต์แวร์ GUI ชุดการเรียนรู้

ตารางที่ 2 ซอฟต์แวร์ชุดคำสั่งของระบบย่อย

ซอฟต์แวร์ชุดคำสั่งหลัก	ซอฟต์แวร์ชุดคำสั่งประกอบ	**ระบบย่อย**
top_entity_sim.vhdl	simulator_package.vhdl RxSerial.vhdl rx_buffer.vhdl simulator.vhdl tx_buffer.vhdl TxSerial.vhd memory_interface.vhdl	ชุดการเรียนรู้ส่วนประมวลผลบนอุปกรณ์ FPGA
RxSerial.vhd	-	Serial receiver
rx_buffer.vhdl	-	Receiver buffer
simulator.vhdl	sim_controller.vhdl uniform_random_input_generator.vhdl channel_encoder.vhdl digital_modulation.vhdl gaussian_random_input_generator.vhdl digital_demodulation.vhdl channel_decoder.vhdl	-
sim_controller.vhdl	-	Simulator controller
uniform_random_input_generator.vhdl	combined_Tausworthe_88.vhdl	Random input generator
channel_encoder.vhdl	ldpc_encoder.vhdl polar_encoder.vhdl	channel encoder (ประกอบด้วยรหัส LDPC และ Polar)
digital_modulation.vhdl	-	Modulator
gaussian_random_input_generator.vhdl	noisegen_top.vhdl	Noise generator

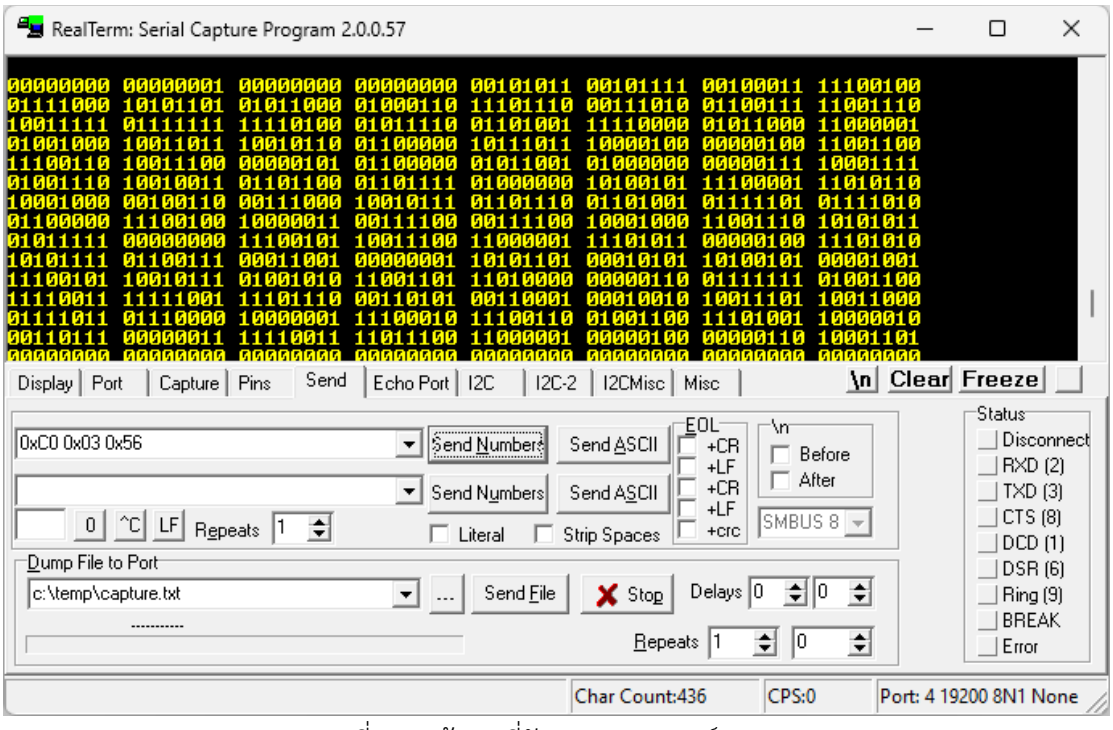
ซอฟต์แวร์ชุดคำสั่งหลัก	ซอฟต์แวร์ชุดคำสั่งประกอบ	**ระบบย่อย**
	combined_Tausworthe_88.vhdl LUTsincos.vhdl LUTcosonefour.vhdl logCal_top.vhdl CofROM.vhdl xinPropare.vhdl polyCal.vhdl sqrtCal.vhdl cordic_calculation.vhdl cordic_equation.vhdl cordic_stage_0.vhdl cordic_stage_1.vhdl cordic_stage_2.vhdl cordic_stage_3.vhdl cordic_stage_4.vhdl cordic_stage_5.vhdl cordic_stage_6.vhdl cordic_stage_7.vhdl cordic_stage_8.vhdl cordic_stage_9.vhdl cordic_stage_10.vhdl cordic_stage_11.vhdl cordic_stage_12.vhdl Div_VHDL_control.vhdl Div_lut.vhdl	
digital_demodulation.vhdl	LLR_BPSK.vhdl LLR_p2BPSK.vhdl LLR_QPSK.vhdl LLR_16QAM.vhdl LUT16QAM_cal.vhdl LUT_16QAM.vhdl LLR_64QAM.vhdl LUT64QAM_cal.vhdl	Demodulator

ซอฟต์แวร์ชุดคำสั่งหลัก	ซอฟต์แวร์ชุดคำสั่งประกอบ	**ระบบย่อย**
	LUT_64QAM.vhdl LLR_256QAM.vhdl LUT256QAM_cal.vhdl LUT_256QAM.vhdl LLR_1024QAM.vhdl LUT1024QAM_cal.vhdl LUT_1024QAM.vhdl	
channel_decoder.vhdl	belief_propagation_layered.vhdl belief_propagation_packaged.vhdl memory_controller.vhdl parity_check_matrix_memory_generator_hdl.vhdl app_memory_generator_hdl.vhdl check_to_var_memory_generator_hdl.vhdl data_arrangement.vhdl barrel_shifter_QSN_bypass.vhdl barrel_shifter_QSN.vhdl barrel_shifter_QSN_reverse.vhdl node_processing_unit.vhdl c2v_subtraction.vhdl c2v_addition.vhdl sign_comparison.vhdl minima_structure.vhdl minima_structure_64.vhdl minima_structure_32.vhdl minima_structure_16.vhdl minima_structure_8.vhdl minima_structure_4.vhdl	channel decoder

ซอฟต์แวร์ชุดคำสั่งหลัก	ซอฟต์แวร์ชุดคำสั่งประกอบ	**ระบบย่อย**
	comparator_1_multiplexer_1.vhdl comparator_1_multiplexer_2.vhdl approximate_min_sum.vhdl	
tx_buffer.vhdl	-	Transmitter buffer
TxSerial.vhd	-	Serial transmitter

4.4 ผลการทดสอบการเชื่อมต่อระหว่างอุปกรณ์ FPGA กับคอมพิวเตอร์

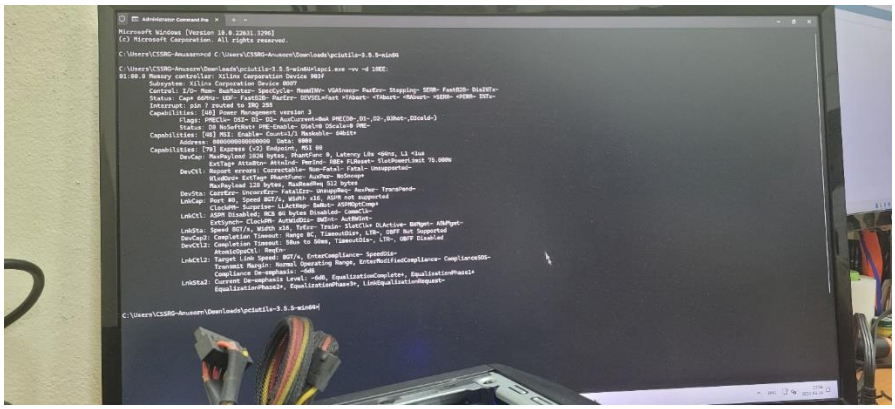
การทดสอบการส่งข้อมูลจาก Computer ไปยังอุปกรณ์ FPGA จะต้องส่งชุดบิตส่วนหัวก่อน จากนั้นตามด้วยชุดบิตข้อมูล และส่งชุดบิตส่วนหัวและชุดบิตข้อมูลให้ครบจำนวนประเภทข้อมูลของแต่ละโหมดหรือโมดูลของการใช้งาน นอกจากนั้น การทดสอบการส่งข้อมูลจากอุปกรณ์ FPGA ไปยัง Computer จะต้องส่งชุดบิตส่วนหัวก่อน จากนั้นตามด้วยจำนวนส่วน ความยาวของชุดบิตข้อมูล และสุดท้ายตามด้วยชุดบิตข้อมูล จำนวนส่วนสามารถบ่งบอกคอมพิวเตอร์ได้ว่าส่วนหัวดังกล่าว จะมีการส่งชุดบิตข้อมูลเป็นจำนวนกี่ครั้ง ประเภทของชุดบิตข้อมูลจะขึ้นอยู่กับแต่ละโหมดหรือโมดูลของการใช้งาน ผลลัพธ์การสื่อสารแสดงดังรูปต่อไปนี้



รูปที่ 3 ชุดข้อมูลที่รับมาจากอุปกรณ์ FPGA

4.5 ผลการทดสอบการทำงานของการ์ดเร่งความเร็ว FEC

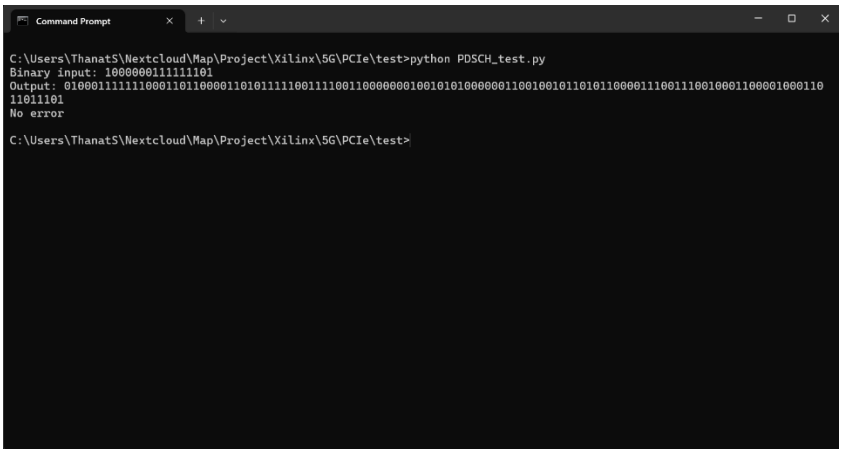
การจัดเตรียมฮาร์ดแวร์: ทำการตั้งค่าอุปกรณ์ FPGA ตามคู่มือกำหนดสำหรับการใช้งาน PCIe แล้วจึงนำอุปกรณ์เสียบลงบนเครื่องคอมพิวเตอร์



รูปที่ 4 ผลการทดสอบ IP-Core DMA/Bridge Subsystem for PCI Express ของ Xilinx

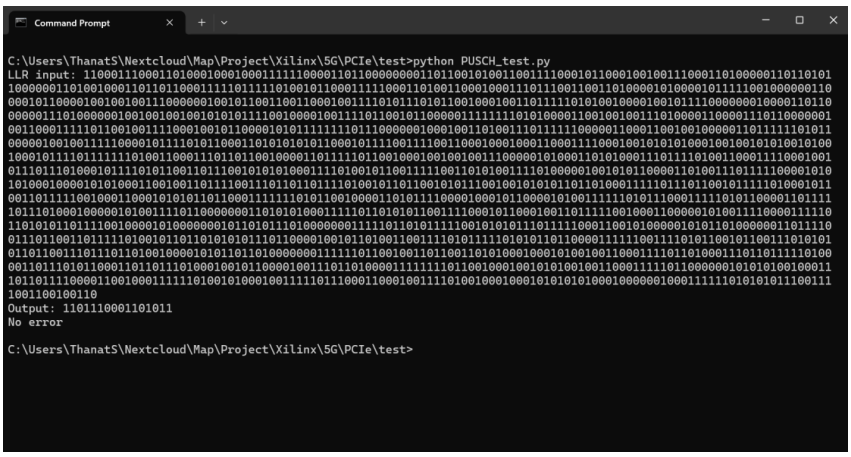
ทดสอบ IP-Core DMA/Bridge Subsystem for PCI Express ตามคู่มือของ Xilinx ผลการทดสอบแสดงในรูปที่ 4 แสดงว่าระบบปฏิบัติการตรวจพบอุปกรณ์ FPGA เป็นหนึ่งในอุปกรณ์ต่อพ่วง PCIe

4.6 ผลการทดสอบการเชื่อมต่อระหว่างการ์ดเร่งความเร็ว FEC กับคอมพิวเตอร์



รูปที่ 5 ผลการทดสอบการ์ดเร่งความเร็ว FEC แบบ AAL_PDSCH_FEC

ทดสอบโดยใช้โปรแกรม Python เรียกใช้ XDMA เพื่อรับส่งข้อมูลสำหรับ AAL_PDSCH ผลการทดสอบแสดงในรูปที่ 5 พบว่าค่าถูกต้องเมื่อเทียบกับโปรแกรม GUI



รูปที่ 6 ผลการทดสอบการ์ดเร่งความเร็ว FEC แบบ AAL_PUSCH_FEC

ทดสอบโดยใช้โปรแกรม Python เรียกใช้ Driver XDMA เพื่อรับส่งข้อมูลสำหรับ AAL_PUSCH ผลการทดสอบแสดงในรูปที่ 6 พบว่าค่าถูกต้องเมื่อเทียบกับโปรแกรม GUI

4.7 ผลการทดสอบเว็บไซต์เผยแพร่ชุดการเรียนรู้และทดสอบสมรรถนะระบบสื่อสารมาตรฐาน 5G

โครงการนี้สร้างเว็บไซต์เพื่อเผยแพร่ชุดการเรียนรู้และทดสอบสมรรถนะระบบสื่อสาร โดยสามารถเข้าถึงได้ที่ https://www.channelcoding.com ภายในเว็บไซต์ประกอบด้วย 6 แถบหลัก ได้แก่ หน้าแรก รหัสช่องสัญญาณ การมอดูเลชัน ช่องสัญญาณรบกวน ชุดการเรียนรู้และทดสอบ และเกี่ยวกับเรา โดยแต่ละแถบมีรายละเอียดดังนี้

แถบหน้าแรก เป็นแถบที่ประกอบด้วยหน้าต่างหน้าแรกซึ่งแสดงข้อมูลแนะนำเว็บไซต์ และผู้สนับสนุน เพื่อให้ผู้ใช้เห็นภาพรวมของเว็บไซต์ นอกจากนี้ยังสามารถเข้าถึงบทความที่อัพเดทล่าสุดและบทความแนะนำจากแถบด้านขวาของหน้าต่าง ผลการทดสอบเว็บไซต์หน้าต่างหน้าแรกแสดงดังรูปที่ 7

รูปที่ 7 ผลการทดสอบเว็บไซต์หน้าต่างหน้าแรก

แถบรหัสช่องสัญญาณ เป็นแถบที่ประกอบด้วยเนื้อหา 4 เรื่อง โดยแต่ละเรื่องจะประกอบด้วยหน้าต่างแสดงรายละเอียดเนื้อหาของเรื่องย่อยดังตารางที่ 2

ตารางที่ 2 รายการเนื้อหาภายในแถบรหัสช่องสัญญาณ

ชื่อเรื่อง	รายการเรื่องย่อย
รหัสช่องสัญญาณคืออะไร	รหัสช่องสัญญาณคืออะไร
รหัสช่องสัญญาณที่ได้ความนิยม	รหัสแฮมมิ่ง รหัสเทอร์โบ รหัส LDPC รหัสโพลาร์
มาตรฐาน 5G (3GPP NR)	การสื่อสารไร้สายยุคที่ 5 ประเภทของรหัสช่องสัญญาณในมาตรฐาน 5G ความเป็นมาของมาตรฐานการเข้ารหัส 5G การเข้ารหัสแอลดีพีมาตรฐาน 5G การเข้ารหัสโพลาร์มาตรฐาน 5G
มาตรฐาน Wi-Fi (IEEE 802.11)	ความเป็นมามาตรฐาน IEEE 802.11 วิวัฒนาการของมาตรฐาน IEEE 802.11 ลักษณะการเชื่อมต่อของอุปกรณ์ IEEE 802.11 การเข้ารหัส LDPC IEEE 802.11

4.8 ผลการเผยแพร่ชุดการเรียนรู้และทดสอบสมรรถนะระบบสื่อสารมาตรฐาน 5G ในมหาวิทยาลัย

การเผยแพร่ชุดการเรียนรู้และทดสอบสมรรถนะในมหาวิทยาลัย 5 แห่ง ได้แก่ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, มหาวิทยาลัยราชมงคลอีสาน วิทยาเขตนครราชสีมา, มหาวิทยาลัยราชมงคลอีสาน วิทยาเขตขอนแก่น, มหาวิทยาลัยขอนแก่น, และจุฬาลงกรณ์มหาวิทยาลัย โดยมีวัตถุประสงค์เพื่อให้นักศึกษาได้เรียนรู้และทดสอบการใช้งานซอฟต์แวร์ชุดการเรียนรู้ระบบสื่อสารมาตรฐาน 5G

การเผยแพร่แบ่งเป็น 2 ส่วนคือ1) การให้ความรู้พื้นฐานเกี่ยวกับระบบสื่อสารในมาตรฐาน 5G โดยมีเอกสารและแบบฝึกหัดให้นักศึกษาทดสอบความเข้าใจและ 2) การทดสอบใช้งานซอฟต์แวร์ชุดการเรียนรู้ระบบสื่อสารมาตรฐาน 5G

การบรรยายแบ่งออกเป็น 4 ช่วงคือ การบรรยายภาพรวมของมาตรฐานสื่อสาร 5G, การบรรยายเรื่องการมอดูเลชันมาตรฐาน 5G และช่องสัญญาณ AWGN และช่องสัญญาณการจางหาย, การบรรยายเกี่ยวกับรหัสช่องสัญญาณ LDPC ในมาตรฐาน 5G และบรรยายเรื่องรหัสช่องสัญญาณโพลาร์ที่ใช้ป้องกันสัญญาณควบคุม หลังการบรรยาย นักศึกษาได้ทดสอบการใช้งานซอฟต์แวร์และปรับค่าพารามิเตอร์ต่าง ๆ เพื่อเสริมสร้างความเข้าใจและเพิ่มความสนุกในการเรียนรู้

5. บทสรุป

โครงการวิจัยนี้มุ่งเน้นใน 2 ด้านหลักคือการศึกษาในวิศวกรรมโทรคมนาคมและการพัฒนาอุปกรณ์โทรคมนาคมต้นแบบ สำหรับด้านการศึกษาวิศวกรรมโทรคมนาคม ทีมวิจัยพัฒนาซอฟต์แวร์ซึ่งเข้ากันได้กับมาตรฐานการสื่อสาร 5G ทำให้นักศึกษาได้เรียนรู้การทำงานของมอดูเลชันและการเข้ารหัสช่องสัญญาณอย่างละเอียด ซอฟต์แวร์นี้ถูกนำไปใช้ในการจัดการเรียนการสอนที่หลากหลายมหาวิทยาลัยทั่วประเทศไทยผ่านเว็บไซต์ https://www.channelcoding.com และสำหรับด้านการพัฒนาอุปกรณ์โทรคมนาคมต้นแบบ ทีมวิจัยพัฒนาอุปกรณ์เข้ารหัสและถอดรหัสช่องสัญญาณบน FPGA โดยใช้ VHDL เพื่อแสดงให้บริษัทเห็นภาพรวมของการทำงานได้จริง โครงการนี้เป็นแรงผลักดันสู่การพัฒนาอุตสาหกรรมโทรคมนาคมในประเทศไทย

ผลลัพธ์จากโครงการนี้คาดว่าจะช่วยสร้างนักศึกษาที่มีความรู้และความเข้าใจในเทคโนโลยีปัจจุบันและสร้างแรงผลักดันให้นักศึกษาและบริษัทท้องถิ่นสนใจที่จะพัฒนาเทคโนโลยีโทรคมนาคมต่อไป

กองทุนวิจัยและพัฒนากิจการกระจายเสียง กิจการโทรทัศน์ และกิจการโทรคมนาคม เพื่อประโยชน์สาธารณะ
(สำนักงาน กสทช.)