

การออกแบบระบบสื่อสาร QPSK ด้วย SOFTWARE-DEFINED RADIO (SDR)
DESIGNING A QPSK COMMUNICATION SYSTEM WITH SOFTWARE-
DEFINED RADIO (SDR)



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบระบบสื่อสาร QPSK ด้วย SOFTWARE-DEFINED RADIO (SDR)
DESIGNING A QPSK COMMUNICATION SYSTEM WITH SOFTWARE-
DEFINED RADIO (SDR)



โดย

นาย กอปรกิตต์ กอบธัญกิจ

61010057

นาย ดิณฑ์ สุวรรณหมัด

61010387

อาจารย์ที่ปรึกษา

ผศ.ดร.สิรภพ ตู้ประกาย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2564

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2564


ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การออกแบบระบบสื่อสาร QPSK ด้วย SOFTWARE-DEFINED RADIO (SDR)
DESIGNING A QPSK COMMUNICATION SYSTEM WITH SOFTWARE-DEFINED
RADIO (SDR)

ผู้จัดทำ

1. นาย กอรปกิตต์ กอบธัญกิจ 61010057
2. นาย ดิณฑ์ สุวรรณหมัด 61010387


..... อาจารย์ที่ปรึกษา
(ผศ.ดร.สิรภพ ตู้ประกาย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เรื่องการออกแบบระบบสื่อสาร QPSK ด้วย Software-Defined Radio (SDR) เล่มนี้จัดทำขึ้นได้ด้วยความกรุณา และความช่วยเหลืออย่างสูง จากอาจารย์ที่ปรึกษา ผศ.ดร. สิริภพ ตู้ประกาย ที่ได้ให้คำปรึกษาแนะนำตลอดจนช่วยแก้ไขข้อผิดพลาดของพระคุณเป็นอย่างสูง

ขอขอบพระคุณผู้เกี่ยวข้องทุกท่านที่ได้กรุณาให้คำแนะนำ และอำนวยความสะดวกในการทำปริญญานิพนธ์รวมถึงที่ได้ให้กำลังใจ และมีส่วนช่วยให้ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปได้ด้วยดี

กอรปกิตต์ กอบธัญกิจ
ติณห์ สุวรรณหมัด
ผู้จัดทำ

การออกแบบระบบสื่อสาร QPSK ด้วย Software-defined Radio (SDR)

DESIGNING A QPSK COMMUNICATION SYSTEM WITH SOFTWARE-DEFINED RADIO (SDR)

โดย นาย กอปรกิตต์ กอบธัญกิจ 61010057

นาย ตินท์ สุวรรณหมัด 61010387

อาจารย์ที่ปรึกษา ผศ.ดร.สิรภพ ตู้ประกาย

บทคัดย่อ

ปริญญานิพนธ์เรื่องการออกแบบระบบสื่อสาร QPSK ด้วย Software-Defined Radio (SDR) นำเสนอการออกแบบระบบสื่อสารดิจิทัลด้วย SDR โดยใช้มาตรฐานการมอดูเลตแบบ QPSK ซึ่งระบบการสื่อสารแบบ SDR นี้ได้มีการพัฒนา และใช้กันแพร่หลายมากขึ้น ระบบการสื่อสารแบบ SDR เป็นระบบที่ใช้หน่วยประมวลผลที่เรียกว่า Field Programmable Gate Array (FPGA) ในการประมวลผลซอฟต์แวร์เพื่อควบคุมการทำงานในส่วนของ Digital Signal Processing (DSP) ภายในระบบสื่อสาร เช่น วงจรมิกเซอร์ (Mixer), วงจรฟิลเตอร์ (Filter), วงจรมอดูเลเตอร์ (Modulator) และวงจรดีมอดูเลเตอร์ (Demodulator) เป็นต้น โดยหน่วยประมวลผล FPGA นี้สามารถทำงานได้กับหลายภาษา ได้แก่ ภาษา VHSIC Hardware Description Language (VHDL) ทำให้ระบบการสื่อสารแบบ SDR นี้มีความยืดหยุ่น และง่ายต่อการปรับปรุง แก้ไข โดยปริญญานิพนธ์นี้ได้เลือกใช้แพลตฟอร์มที่เป็นฮาร์ดแวร์ และซอฟต์แวร์ซึ่งก็คือ USRP และ MATLAB เพื่อใช้ในการออกแบบระบบการสื่อสาร QPSK และใช้เป็นผลอ้างอิงในการ ออกแบบ และประยุกต์ใช้งานระบบสื่อสาร QPSK ด้วยภาษา VHDL บนบอร์ด FPGA เพื่อที่จะนำไปต่อยอดเป็นฮาร์ดแวร์ประเภท SDR ภายในภาคหน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSTRACT

This thesis “Designing a QPSK communication system with Software-Defined Radio (SDR)” presents a digital communication system with SDR using a QPSK modulation technique. An SDR communication system is developed and widely used. An SDR communications system is a system that uses a processing unit called a Field Programmable Gate Array (FPGA) to process software to control the Digital Signal Processing (DSP) within a communication system, such as a mixer, filter, modulator and demodulator etc. This FPGA processor can work with multiple languages, including VHSIC Hardware Description Language (VHDL). The SDR system makes a communication system design be flexible and high re-configurable. This thesis uses the SDR platform combined with hardware and software which is USRP B210 and MATLAB. The result from MATLAB is refer to develop and implement with VHDL design for FPGA board. To build our own SDR in future.

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	IV
สารบัญรูป	VIII
สารบัญตาราง	XII
บทที่ 1	
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของโครงการ	2
บทที่ 2	
ทฤษฎีและหลักการที่เกี่ยวข้อง	5
2.1 Software Defined Radio (SDR)	5
2.1.1 USRP B210	5
2.1.2 USRP E320	7
2.2 การสื่อสารทางดิจิทัล	8
2.2.1 Scramble	8
2.2.2 Barker Code	9
2.2.3 Finite Impulse Response Filter (FIR Filter)	10
2.2.4 Root Raised Cosine Filter (RRC Filter)	10
2.2.5 Roll Off Vector (β)	11
2.2.6 Interpolation	12
2.2.7 Sampling	12
2.2.8 Modulation	12

สารบัญ (ต่อ)

	หน้า
2.2.9 ผลกระทบในช่องสัญญาณ	14
2.2.10 Automatic Gain Control (AGC)	15
2.2.11 Receiver RRC (Root Raised Cosine)	15
2.2.12 Coarse Frequency Compensation	15
2.2.13 Symbol Synchronizer	18
2.2.14 Carrier Synchronizer	19
2.2.15 Preamble Detector	20
2.2.16 Frame Synchronizer	20
2.2.17 BER Data Decoding	20
2.2.18 QPSK Demodulator	20
2.2.19 Descrambler	21
2.3 พารามิเตอร์ที่ใช้วัด	21
2.3.1 อัตราความผิดพลาดบิต (BER)	22
2.3.2 กำลังของสัญญาณ	22
2.3.3 แบนด์วิท (Bandwidth)	22
2.3.4 อัตราส่วนของสัญญาณคลื่นพาห์ต่อสัญญาณรบกวน (C/N)	22
2.3.5 ขนาดเวกเตอร์ข้อผิดพลาด (EVM)	23
2.3.6 อัตราส่วนข้อผิดพลาดในการมอดูเลต (MER)	23
2.3.7 แผนภาพคอนสเทลเลชัน (Constellation diagram)	23
2.3.8 แผนภาพดวงตา (Eye diagram)	24
2.4 ซอฟต์แวร์ที่ใช้	25
2.4.1 MATLAB R2021b	25
2.4.2 Aldec active hdl	26
2.4.3 Vivado	27

สารบัญ (ต่อ)

	หน้า
2.5 ภาษา VHDL	28
2.5.1 ส่วนประกอบของ VHDL ประกอบด้วย 3 ส่วน	28
2.6 FPGA	28
2.6.1 IOB (Input/Output Block)	29
2.6.2 CLB (Configuration Logic Block)	29
2.6.3 Interconnected line	30
2.6.4 Switch matrix	30
2.7 ฮาร์ดแวร์	30
2.7.1 ZYBO Z7-20 Development Board	30
2.7.2 R&S RTM 2024 Oscilloscope	32
บทที่ 3 การออกแบบและการจัดทำปริญญาณิพนธ์	34
3.1 การออกแบบ	34
3.1.1 การออกแบบระบบสื่อสาร QPSK ด้วยซอฟต์แวร์ MATLAB	35
3.1.2 การออกแบบการทดสอบด้วย USRP B210	48
3.1.3 การออกแบบการทดสอบด้วย FPGA	51
3.2 เครื่องมือที่ใช้ในการทดลอง	64
3.2.1 ซอฟต์แวร์	64
3.2.2 ฮาร์ดแวร์	64
3.3 ขั้นตอนการทดลอง	72
3.3.1 การทดสอบด้วยซอฟต์แวร์ MATLAB	72
3.3.2 การทดลองระบบสื่อสาร QPSK ด้วย USRP B210	73
3.3.3 การทดลองระบบสื่อสาร QPSK ด้วย FPGA	77

สารบัญ (ต่อ)

	หน้า
บทที่ 4 ผลการทดลอง	86
4.1 ผลการทดสอบระบบสื่อสาร QPSK ด้วยซอฟต์แวร์ MATLAB	86
4.2 ผลการทดสอบระบบสื่อสาร QPSK ด้วย USRP B210	90
4.2.1 ผลการทดสอบระบบสื่อสาร QPSK ด้วย USRP B210 ผ่านสายโคแอกซ์เซียล	90
4.2.2 ผลการทดลองระบบสื่อสาร QPSK ด้วย USRP B210 ผ่านสายอากาศ	92
4.2.3 ผลการทดลองระบบสื่อสาร QPSK ด้วย USRP B210 ที่ symbol rate ต่างๆ	94
4.2.4 ผลการทดสอบระบบสื่อสาร M-PSK ด้วย USRP B210	97
4.3 ผลการทดสอบระบบสื่อสาร QPSK ด้วย FPGA	103
4.3.1 ผลของวงจร Scrambler	103
4.3.2 ผลของวงจร Polar NRZ-L	106
4.3.3 ผลของวงจร I/Q Separate	108
4.3.4 ผลของวงจร I/Q Pairing	111
4.3.5 ผลของวงจร QPSK Mapping	114
4.3.6 ผลของวงจร Tx RRC Filter	119
4.3.7 ผลรายงานจากซอฟต์แวร์ Vivado	126
บทที่ 5 สรุปผลและข้อเสนอแนะ	128
5.1 สรุปผล	128
5.2 ข้อเสนอแนะ	129
บรรณานุกรม	130

สารบัญรูป

รูปที่	หน้า
1.1 บล็อกไดอะแกรมการทดสอบระบบสื่อสาร QPSK ด้วยซอฟต์แวร์ MATLAB	2
1.2 แผนภาพการทดสอบด้วย USRP B210	3
1.3 บล็อกไดอะแกรมการทดสอบด้วย USRP B210	3
1.4 บล็อกไดอะแกรมการทดสอบระบบสื่อสาร QPSK ด้วย FPGA	4
2.1 บล็อกไดอะแกรมโครงสร้างภายในของ USRP B210	5
2.2 USRP E320	7
2.3 บล็อกไดอะแกรมโครงสร้างของ USRP E320	8
2.4 การทำงานสแครมเบิล	9
2.5 การซ้อนทับกันของข้อมูล	11
2.6 ขนาดของโหลบในโดเมนเวลา	11
2.7 ขนาดแบนด์วิธเพิ่มขึ้นในโดเมนความถี่	11
2.8 สเปกตรัมของสัญญาณ Baseband และ Bandpass	12
2.9 ผลตอบสนองความถี่	12
2.10 บล็อกไดอะแกรม QPSK Modulate	13
2.11 QPSK Constellation Diagram	14
2.12 ตัวอย่างวงจรควบคุมอัตราขยายตัวอัตโนมัติ	15
2.13 อัลกอริทึม Symbol Timing PLL	18
2.14 อัลกอริทึม Phase-Locked Loop (PLL)	19
2.15 โมดูลภายในบล็อก BER Data Decoding	20
2.16 QPSK Modulation	21
2.17 วงจรดีสแครมเบิล	21
2.18 แผนภาพดวงตา	24

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.19 ซอฟต์แวร์ Aldec Active HDL	26
2.20 ขั้นตอนการทำงานของ Aldec Active HDL	26
2.21 ซอฟต์แวร์ Vivado	27
2.22 ขั้นตอนการทำงานของ Aldec Active HDL	27
2.23 FPGA Architecture	29
2.24 Structure of Configuration Logic Block	30
2.25 ZYBO Z7-20 Development Board	31
2.26 Zynq APSoC (All Programmable System-on-Chip)	31
2.27 ลำดับของ Pin ในหนึ่ง Pmod	32
2.28 R&S RTM 2024 Oscilloscope	33
3.1 บล็อกไดอะแกรมแผนการทำงานทั้งหมด	34
3.2 บล็อกไดอะแกรมของระบบสื่อสาร QPSK	36
3.3 ระบบสื่อสาร QPSK ภาคส่ง ออกแบบด้วยซอฟต์แวร์ Simulink	36
3.4 บล็อก Scrambler ใน Simulink	37
3.5 การทำงานของสแครมเบลอร์	37
3.6 ข้อมูลที่ส่งข้อความ “Hello world 026\n” ผ่านสแครมเบลอร์	38
3.7 สร้างเฮตเตอร์ของเฟรมข้อมูลด้วยลำดับของรหัสบาร์คเกอร์	38
3.8 ข้อมูลบาร์คเกอร์โค้ดที่สร้างขึ้น	39
3.9 ข้อมูลบาร์คเกอร์โค้ดรวมกับข้อมูลอินพุตผ่านสแครมบลิง	39
3.10 บล็อก QPSK Modulator ใน Simulink	39
3.11 บล็อก Raised Cosine Transmit Filter ใน Simulink	40
3.12 ผลตอบสนองของตัวกรอง RRC filter	40
3.13 ภาพรวมของการส่งผ่านข้อมูลผ่านมอดูเลเตอร์ และฟิลเตอร์	41

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.14	42
3.15	43
3.16	43
3.17	44
3.18	45
3.19	45
3.20	46
3.21	46
3.22	47
3.23	47
3.24	48
3.25	48
3.26	49
3.27	49
3.28	50
3.29	51
3.30	51
3.31	52
3.32	53
3.33	54
3.34	55

สารบัญรูป (ต่อ)

รูปที่	หน้า	
3.35	บล็อก Clock Divider	56
3.36	บล็อก Parallel to Serial	57
3.37	บล็อก Barker_LUT	58
3.38	บล็อก Scrambler	58
3.39	บล็อก Poalar NRZ-L	59
3.40	บล็อก I/Q Separate	60
3.41	บล็อก I/Q Symbol	61
3.42	บล็อก QPSK Mapping	62
3.43	บล็อก RRC filter	63
3.44	แล็ปท็อป ASUS TUF Gaming A15 FA506II-AL012T	65
3.45	แล็ปท็อป HP Pavilion Laptop 15-cc000	65
3.46	USRP B210	66
3.47	สายอากาศ 2.4 GHz	66
3.48	สายโคแอกซ์เซียล	67
3.49	Coaxial Directional Coupler	67
3.50	Attenuator	68
3.51	Spectrum Analyzer	68
3.52	FPGA board (Zybo Z7-20)	69
3.53	Oscilloscope (R&S RTM 2024)	69
3.54	สาย USB B Micro	70
3.55	Logic probe และสายไฟสำหรับวัดสัญญาณดิจิทัล	70
3.56	Interface cable	71
3.57	Pin header ตัวผู้ 2 แถว	71

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.58 ข้อความ “Hello world 026\n” ที่แปลงข้อมูลจาก ASCII เป็น ไบนารี	72
3.59 สคริปต์แมทแลปที่ใช้ในการสร้างข้อมูลอินพุต	73
3.60 อุปกรณ์สำหรับการทดสอบระบบสื่อสาร QPSK ผ่านสายโคแอกซ์ เชียล	74
3.61 อุปกรณ์สำหรับการทดสอบระบบสื่อสาร QPSK ผ่านสายอากาศ	75
3.62 วิดีโอขนาด 100*100 pixels 1 fps	76
3.63 บล็อกไดอะแกรมการทดสอบระบบสื่อสาร M-PSK ภาควัดด้วย USRP B210	77
3.64 การออกแบบการทดลองด้วย FPGA	77
4.1 บล็อกไดอะแกรมระบบการสื่อสาร QPSK บนซอฟต์แวร์ Simulink	86
4.2 เปรียบเทียบสเปกตรัมที่ระบบสื่อสารภาควัดรับหลังผ่าน Rx RRC filter	87
4.3 ข้อความที่ถูกถอดรหัสที่ภาควัดรับของระบบสื่อสาร QPSK	87
4.4 กราฟ Constellation หลังผ่าน Rx RRC filter	88
4.5 กราฟ Constellation หลังผ่าน Symbol Synchronizer	89
4.6 กราฟ Constellation หลังผ่าน Carrier Synchronizer	89
4.7 สเปกตรัมของสัญญาณ QPSK รับผ่านสายโคแอกซ์ เชียล	90
4.8 หน้าต่าง VSA แสดงผลของสัญญาณ QPSK รับผ่านสายโคแอกซ์ เชียล	91
4.9 ผลของหน้าต่างของระบบสื่อสาร QPSK ภาควัดรับ รับผ่านสายโคแอกซ์ เชียล	91
4.10 สเปกตรัมของสัญญาณ QPSK รับผ่านสายอากาศ	92
4.11 หน้าต่าง VSA แสดงผลของสัญญาณ QPSK รับผ่านสายอากาศ	93
4.12 ผลของหน้าต่างของระบบสื่อสาร QPSK ภาควัดรับ รับผ่านสายอากาศ	93

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.13	สเปกตรัมของสัญญาณ QPSK ส่งที่ Symbol Rate เท่ากับ 0.25 MHz	95
4.14	หน้าต่าง VSA ของสัญญาณ QPSK ส่งที่ Symbol Rate เท่ากับ 0.25 MHz	96
4.15	สเปกตรัมของสัญญาณ QPSK ที่ Eb/No เท่ากับ 100 dB	97
4.16	หน้าต่าง VSA ของสัญญาณ QPSK ที่ Eb/No เท่ากับ 100 dB	97
4.17	สเปกตรัมของสัญญาณ QPSK ที่ Eb/No เท่ากับ 80 dB	98
4.18	หน้าต่าง VSA ของสัญญาณ QPSK ที่ Eb/No เท่ากับ 80 dB	99
4.19	สเปกตรัมของสัญญาณ 8-PSK ที่ Eb/No เท่ากับ 100 dB	100
4.20	หน้าต่าง VSA ของสัญญาณ 8-PSK ที่ Eb/No เท่ากับ 100 dB	100
4.21	สเปกตรัมของสัญญาณ 16-PSK ที่ Eb/No เท่ากับ 100 dB	101
4.22	หน้าต่าง VSA ของสัญญาณ 16-PSK ที่ Eb/No เท่ากับ 100 dB	102
4.23	กราฟความสัมพันธ์ระหว่าง C/N และ Eb/No (AWGN Channel) ของมอดูเลชันแบบ PSK	103
4.24	ผลการจำลองการทำงานของวงจร Scrambler บนซอฟต์แวร์ Aldec Active-HDL	104
4.25	ผลการจำลองการทำงานของวงจร Scrambler บนซอฟต์แวร์ Aldec Active-HDL	105
4.26	ผลของวงจร Scrambler จากบอร์ด FPGA วัดด้วยออสซิลโลสโคป	105
4.27	ผลจากการวัดสัญญาณของ Polar NRZ-L ด้วยซอฟต์แวร์ MATLAB	106
4.28	ผลการจำลองการทำงานของวงจร Polar NRZ-L บนซอฟต์แวร์ Aldec Active-HDL	107
4.29	ผลของวงจร Polar NRZ-L จากบอร์ด FPGA วัดด้วยออสซิลโลสโคป	108

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.30 ผลจากการวัดสัญญาณของ I/Q Separate ด้วยซอฟต์แวร์ MATLAB	109
4.31 ผลการจำลองการทำงานของวงจร I/Q Separate บนซอฟต์แวร์ Aldec Active-HDL	110
4.32 ผลของวงจร I/Q Separate จากบอร์ด FPGA วัดด้วยออสซิลโลสโคป	111
4.33 ถอดรหัสสัญญาณของวงจร I/Q Separate บนบอร์ด FPGA ด้วยออสซิลโลสโคป	111
4.34 ผลจากการวัดสัญญาณของ I/Q Pairing ด้วยซอฟต์แวร์ MATLAB	112
4.35 ผลการจำลองการทำงานของวงจร I/Q Pairing บนซอฟต์แวร์ Aldec Active-HDL	113
4.36 ผลของวงจร I/Q Pairing จากบอร์ด FPGA วัดด้วยออสซิลโลสโคป	114
4.37 ถอดรหัสสัญญาณของวงจร I/Q Pairing บนบอร์ด FPGA ด้วยออสซิลโลสโคป	114
4.38 ผลจากการวัดสัญญาณของ QPSK Mapping ด้วยซอฟต์แวร์ MATLAB	115
4.39 ผลของการจำลองการทำงานของวงจร QPSK Mapping บนซอฟต์แวร์ Aldec Active-HDL	116
4.40 ข้อมูลเอาต์พุต (Real) จากการจำลองการทำงานของวงจร QPSK Mapping บนซอฟต์แวร์ Aldec Active-HDL	116
4.41 ข้อมูลเอาต์พุต (Image) จากการจำลองการทำงานของวงจร QPSK Mapping บนซอฟต์แวร์ Aldec Active-HDL	117
4.42 ผลของวงจร QPSK Mapping จากบอร์ด FPGA วัดด้วยออสซิลโลสโคป	118

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.43 สัญญาณ Real ของวงจร QPSK Mapping จากบอร์ด FPGA ที่วัดโดย ออสซิลโลสโคป	118
4.44 สัญญาณ Image ของวงจร QPSK Mapping จากบอร์ด FPGA ที่วัด โดยออสซิลโลสโคป	118
4.45 ผลตอบสนองอิมพัลส์ของ RRC Filter ด้วยซอฟต์แวร์ MATLAB	119
4.46 ผลจากการวัดสัญญาณของ RRC Filter ด้วยซอฟต์แวร์ MATLAB	120
4.47 ผลตอบสนองอิมพัลส์ของ RRC Filter ด้วยซอฟต์แวร์ Aldec Active- HDL 8.3	120
4.48 ผลของสัญญาณ In-phase จาก RRC Filter ด้วยซอฟต์แวร์ Aldec Active-HDL 8.3	121
4.49 ผลของสัญญาณ Quadrature จาก RRC Filter ด้วยซอฟต์แวร์ Aldec Active-HDL 8.3	121
4.50 ผลตอบสนองอิมพัลส์ของวงจร RRC Filter จากบอร์ด FPGA วัดด้วย ออสซิลโลสโคป	122
4.51 พล็อตกราฟแสดงผลตอบสนองอิมพัลส์ของวงจร RRC Filter เทียบกับ ผลการจำลองจาก MATLAB	122
4.52 ผลของสัญญาณ In-phase ของวงจร RRC Filter จากบอร์ด FPGA ที่ วัดโดยออสซิลโลสโคป	123
4.53 ผลของการถอดรหัสสัญญาณ In-phase ของวงจร RRC Filter จาก บอร์ด FPGA ที่วัดโดยออสซิลโลสโคป	123
4.54 ผลของสัญญาณ Quadrature ของวงจร RRC Filter จากบอร์ด FPGA ที่วัดโดยออสซิลโลสโคป	124

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.55 ผลของการถอดรหัสสัญญาณ Quadrature ของวงจรร RRC Filter จากบอร์ด FPGA ที่วัดโดยออสซิลโลสโคป	124
4.56 ผลเปรียบเทียบสัญญาณ In-phase เอาต์พุตของระบบสื่อสาร QPSK ภาคส่ง ของสัญญาณ ระหว่าง FPGA และ MATLAB	125
4.57 ผลเปรียบเทียบสัญญาณ Quadrature เอาต์พุตของระบบสื่อสาร QPSK ภาคส่ง ของสัญญาณ ระหว่าง FPGA และ MATLAB	125
4.58 กราฟแสดงจำนวนการใช้ทรัพยากรภายในบอร์ด FPGA โดยด้านซ้าย เป็นผลหลังทำการ Synthesis วงจร และด้านขวาเป็นผลหลังการทำ Implementation	126
4.59 ผลการแสดงผลการใช้พลังงานภายในบอร์ด FPGA	126
4.60 ค่าแสดงผลทางเวลาหลังการสังเคราะห์วงจรลงบอร์ด FPGA ด้วย ความถี่ของสัญญาณนาฬิกาอินพุตเท่ากับ 125 MHz	127

สารบัญตาราง

ตาราง ที่	หน้า
2.1 ไฟ LED บ่งบอกสถานะการทำงานของ USRP B210	6
2.2 ลำดับของ Barker ที่ Length ต่างๆ	9
2.3 ชื่อ และชนิดของพอร์ตต่างๆ	32
3.1 แสดงค่าพารามิเตอร์ต่างๆ ที่ใช้สำหรับการออกแบบระบบสื่อสาร QPSK	34
3.2 เปรียบเทียบสเปกของแลปทอประหว่างเครื่องส่ง และเครื่องรับที่ใช้ใน การทดลอง	74
3.3 แมปชื่อสัญญาณอินพุต เอาต์พุตของบล็อก Scrambler กับพอร์ตของ Zybo Z7-20 และ logic analyzer	78
3.4 แมปชื่อสัญญาณอินพุต เอาต์พุตของบล็อก Polar NRZ-L กับพอร์ต ของ Zybo Z7-20 และ logic analyzer	79
3.5 แมปชื่อสัญญาณอินพุต เอาต์พุตของบล็อก I/Q Separate กับพอร์ต ของ Zybo Z7-20 และ logic analyzer	79
3.6 3.6 แมปชื่อสัญญาณอินพุต เอาต์พุตของบล็อก I/Q Pairing กับพอร์ต ของ Zybo Z7-20 และ logic analyzer	80
3.7 แมปชื่อสัญญาณอินพุต เอาต์พุตของบล็อก QPSK Mapping กับ พอร์ตของ Zybo Z7-20 และ logic analyzer	82
3.8 แมปชื่อสัญญาณอินพุต เอาต์พุตของบล็อก Tx RRC Filter กับพอร์ต ของ Zybo Z7-20 และ logic analyzer	84
4.1 ผลการทดลองระบบสื่อสาร QPSK ที่ Symbol Rate ต่างๆ	94
4.2 ค่า C/N ของระบบสื่อสาร QPSK, 8-PSK และ 16-PSK ด้วยค่า Eb/No ของช่องสัญญาณ AWGN ที่แตกต่างกัน	102

สารบัญตาราง

ตาราง ที่	หน้า
4.3 ผลของ Scrambler ด้วยซอฟต์แวร์ MATLAB	104
4.4 ผลของวงจร Scrambler ด้วยซอฟต์แวร์ Aldec Active-HDL 8.3	104
4.5 ผลของวงจร Scrambler จากบอร์ด FPGA	105
4.6 ผลของ Polar NRZ-L ด้วยซอฟต์แวร์ MATLAB	106
4.7 ผลของวงจร Polar NRZ-L ด้วยซอฟต์แวร์ Aldec Active-HDL 8.3	107
4.8 ผลของวงจร Polar NRZ-L จากบอร์ด FPGA	107
4.9 ผลของ I/Q Separate ด้วยซอฟต์แวร์ MATLAB	108
4.10 ผลของวงจร I/Q Separate ด้วยซอฟต์แวร์ Aldec Active-HDL 8.3	109
4.11 ผลของวงจร I/Q Separate จากบอร์ด FPGA	110
4.12 ผลของ I/Q Pairing ด้วยซอฟต์แวร์ MATLAB	112
4.13 ผลของวงจร I/Q Pairing ด้วยซอฟต์แวร์ Aldec Active-HDL 8.3	113
4.14 ผลของวงจร I/Q Pairing ด้วยซอฟต์แวร์ Aldec Active-HDL 8.3	113
4.15 ผลของ QPSK Mapping ด้วยซอฟต์แวร์ MATLAB	115
4.16 ผลของวงจร QPSK Mapping ด้วยซอฟต์แวร์ Aldec Active-HDL 8.3	116
4.17 ผลของวงจร QPSK Mapping จากบอร์ด FPGA	117

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ปัจจุบันเทคโนโลยีดิจิทัลมีการพัฒนาเป็นอย่างมาก รวมถึงการพัฒนาในด้านการสื่อสารวิทยุ [1] ประมาณปี.ศ.1980 องค์ประกอบของระบบการสื่อสารเริ่มถูกพัฒนาด้วยเทคโนโลยีดิจิทัลได้แก่ การประมวลผลสัญญาณเสียง (Audio Signal Processing), วงจรมอดูเลเตอร์ (Modulator), วงจรดีมอดูเลเตอร์ (Demodulator), วงจรฟิลเตอร์ (Filters) และวงจรมิกเซอร์ (Mixers) ซึ่งในปัจจุบันได้มีการพัฒนา และควบคุมกระบวนการภายในระบบสื่อสารดิจิทัลด้วยซอฟต์แวร์ ซึ่งเรียกระบบนี้ว่า Software Defined Radio (SDR) หรือ Software Radio

ระบบการสื่อสารแบบ SDR [1] เป็นระบบการสื่อสารที่ใช้ซอฟต์แวร์ในการควบคุมองค์ประกอบต่างๆ ภายในระบบสื่อสาร ทำให้ระบบการสื่อสารแบบ SDR นี้ มีความยืดหยุ่น ง่ายต่อการปรับเปลี่ยน และแก้ไขคุณสมบัติของการสื่อสาร โดยปราศจากฮาร์ดแวร์ ประโยชน์การใช้งานของ SDR อาทิเช่น การใช้เครื่องวิทยุเพียงตัวเดียวในการเข้าถึงเพื่อแก้ไขระบบเครื่องส่งหลายเครื่องที่มีช่วงใช้งานในย่านความถี่ที่แตกต่างกัน เป็นต้น SDR [2] นั้นมีอุปกรณ์ที่เป็นฮาร์ดแวร์สำหรับการค้าหลายยี่ห้อ เช่น Universal Software Radio Peripheral (USRP) ถูกผลิตโดย Ettus Research ทำหน้าที่เป็น RF Front-End สำหรับใช้งานนอกแบบการสื่อสารด้วยซอฟต์แวร์เช่นวิทยุสมัครเล่น,ระบบนำทางด้วยดาวเทียม และระบบโครงข่ายวิทยุ เป็นต้น โดยมีช่วงใช้งานตั้งแต่สัญญาณไฟตรงจนถึงสัญญาณที่มีความถี่ 6 GHz

ด้วยเหตุผลตามที่กล่าวมานี้ จึงเป็นสาเหตุให้โครงการนี้เลือกใช้ SDR ในการสร้างระบบการสื่อสาร QPSK โดยใช้แพลตฟอร์มที่เป็นฮาร์ดแวร์ และซอฟต์แวร์ซึ่งก็คือ USRP และ MATLAB และ Simulink โดยใช้ซอฟต์แวร์ MATLAB และ Simulink ในการออกแบบกระบวนการประมวลผลสัญญาณดิจิทัล (Digital Signal Processing : DSP) และประยุกต์ใช้อัลกอริทึมที่ได้ออกแบบกับ Field Programmable Gate Array (FPGA) ซึ่งอยู่ภายใน USRP เพื่อวัดผลการทำงานของระบบการสื่อสาร QPSK ตามอัลกอริทึมที่ได้ออกแบบไว้

1.2 วัตถุประสงค์

โครงการเรื่องการออกแบบระบบสื่อสาร QPSK ด้วย Software-Defined Radio (SDR) มีวัตถุประสงค์ดังนี้

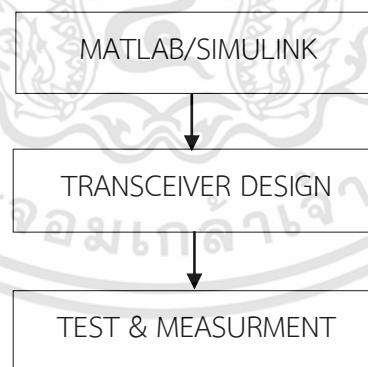
- ศึกษา และออกแบบการสื่อสารดิจิทัลด้วยเทคนิคการมอดูเลตแบบ QPSK โดยใช้ Software-Defined Radio (SDR) กับซอฟต์แวร์ MATLAB
- เพื่อประยุกต์ใช้โมดูลภายในระบบสื่อสารที่ได้ทำการออกแบบกับ Field Programmable Gate Array (FPGA)
- เพื่อศึกษาการใช้อุปกรณ์ และเครื่องมือวัด ทางด้านโทรคมนาคม

1.3 ขอบเขตของปริญญาโท

โครงการเรื่องการออกแบบระบบสื่อสาร QPSK ด้วย Software-Defined Radio (SDR) มีขอบเขต และขั้นตอนในการทดลอง 3 ส่วนดังนี้

1. การทดสอบระบบสื่อสาร QPSK ด้วยซอฟต์แวร์ MATLAB

การทดสอบในส่วนแรกเป็นการทดสอบระบบสื่อสารดิจิทัลที่ถูกออกแบบด้วยซอฟต์แวร์ MATLAB และ Simulink ทั้งภาครับ และภาคส่ง โดยจะจำลองการสื่อสาร และวัดผลผ่านซอฟต์แวร์ดังรูปที่ 1.1



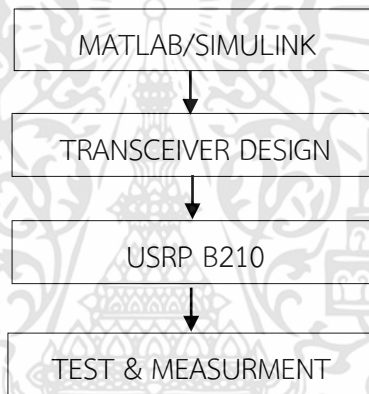
รูปที่ 1.1 บล็อกไดอะแกรมการทดสอบระบบสื่อสาร QPSK ด้วยซอฟต์แวร์ MATLAB

2. Test with USRP B210

การทดสอบในส่วนที่ 2 เป็นการทดสอบระบบการสื่อสารดิจิทัลที่ได้ออกแบบมาใน ส่วนแรก โดยทำการสื่อสารกันระหว่างแล็ปท็อป 2 เครื่องที่ทำหน้าที่เป็นภาคส่ง และภาครับ ผ่าน USRP B210 ซึ่งทำหน้าที่เป็น RF front-end ดังรูปที่ 1.2 และไดอะแกรมดังรูปที่ 1.3



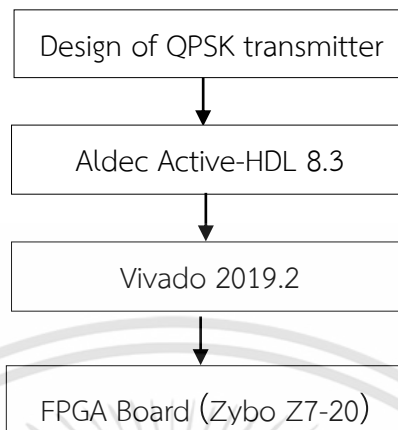
รูปที่ 1.2 แผนภาพการทดสอบด้วย USRP B210



รูปที่ 1.3 บล็อกไดอะแกรมการทดสอบด้วย USRP B210

3. การทดสอบระบบสื่อสาร QPSK ด้วย FPGA

การทดสอบในส่วนสุดท้ายจะเป็นการออกแบบระบบสื่อสาร QPSK ภาคส่ง ด้วยภาษา VHDL เพื่อสังเคราะห์เป็นวงจรดิจิทัลลงบนบอร์ด FPGA ซึ่งบอร์ดที่ใช้ก็คือ Zybo Z7-20 โดย ออกแบบระบบสื่อสาร QPSK ภาคส่งด้วยซอฟต์แวร์ Aldec Active-HDL 8.3 ก่อนนำไปสังเคราะห์ ลงบอร์ด FPGA ด้วยซอฟต์แวร์ Vivado 2019.2 ซึ่งมีบล็อกไดอะแกรมดังรูปที่ 1.4



รูปที่ 1.4 บล็อกไดอะแกรมการทดสอบระบบสื่อสาร QPSK ด้วย FPGA

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

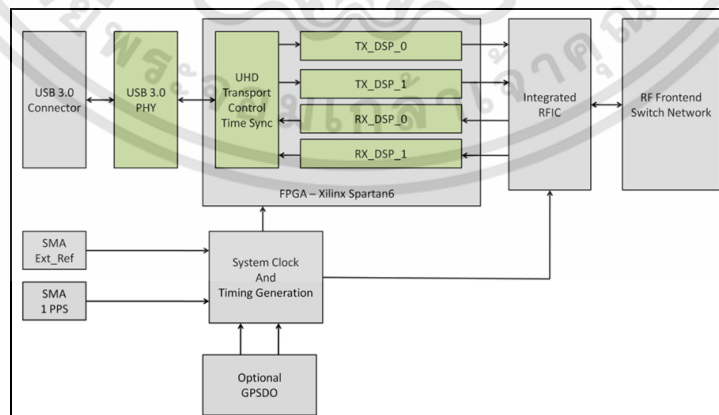
2.1 Software Defined Radio (SDR)

เป็นระบบการสื่อสารสัญญาณวิทยุซึ่งใช้ซอฟต์แวร์สำหรับการออกแบบและกำหนดการทำงานขององค์ประกอบต่าง เช่น วงจรรวมสัญญาณ, วงจรกรองสัญญาณ, วงจรมอดูเลเตอร์ และ วงจรดีมอดูเลเตอร์ เป็นต้น

Ettus Research [2] เป็นผู้นำแพลตฟอร์มด้าน SDR โดยได้ออกแบบผลิตภัณฑ์ประเภท SDR ฮาร์ดแวร์ คือ USRP โดย USRP นี้ทำหน้าที่เป็น rf front-end ของระบบสื่อสาร โดยจะประกอบไปด้วยหน่วยประมวลผล FPGA สำหรับทำหน้าที่ด้าน DSP และวงจรไมโครเวฟ ได้แก่ วงจรแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก หรือวงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล โดยแพลตฟอร์ม SDR นั้นได้รับความสนใจเป็นอย่างมากในการสื่อสารทางสัญญาณวิทยุ

2.1.1 USRP B210

USRP B210 [2] เป็น RF-frontend ซึ่งประกอบไปด้วย 2 ส่วนหลัก คือส่วนที่เป็น Processor เพื่อใช้ในการคำนวณด้าน Digital Signal Processing (DSP) โดยใช้ชิพ FPGA ของ Xilinx Spartan6 และในส่วนวงจรไมโครเวฟ ได้แก่ ADC และ DAC ช่วงความถี่ที่ใช้งานคือ 70 MHz – 6 GHz รองรับสาย USB 3.0 มี 2 พอร์ต Tx และ 2 พอร์ต Rx มี System Clock ในตัว ดังรูปที่ 2.1



รูปที่ 2.1 บล็อกไดอะแกรมโครงสร้างภายในของ USRP B210 [2]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1.1 สถานะไฟ LED ของ USRP B210

USRP B210 [2] มีไฟ LED สำหรับบ่งบอกสถานะการทำงาน ระหว่างใช้งานดังตารางที่ 2.1

ตารางที่ 2.1 ไฟ LED บ่งบอกสถานะการทำงานของ USRP B210

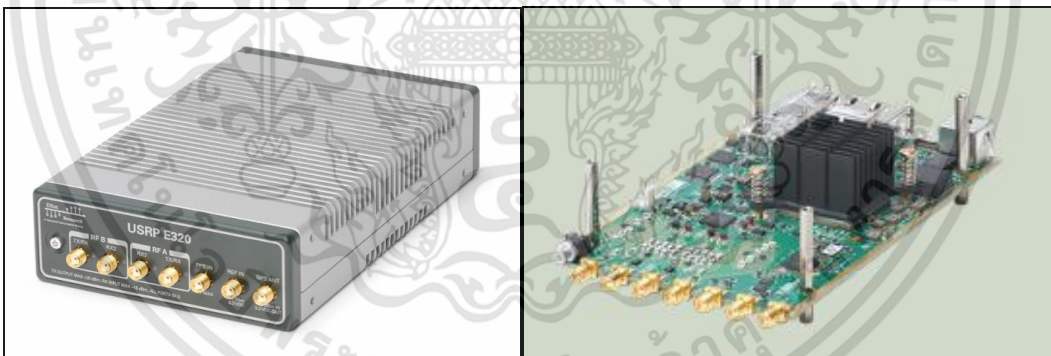
หมายเลข ID	คำอธิบาย	รายละเอียด
ED600	Power Indicator	ปิด = ไม่มีไฟเลี้ยง (before rev6, some rev6) / external power applied and not yet initialized (most rev6) น้ำเงิน = USB power applied (before rev6) / power supplied (USB) OR power supplied (external) (rev6) แดง = external power applied (before rev6) / external power applied and device initialized (rev6)
LED800	Channel 2 RX2 Activity	ปิด = ไม่มีไฟเลี้ยง เขียว = กำลังรับ
LED801	Channel 2 TX/RX Activity	ปิด = ไม่ได้ทำงาน เขียว = กำลังรับ แดง = กำลังส่ง ส้ม = สลับการทำงานระหว่างกำลังส่ง และกำลังรับ
LED802	Channel 1 TX/RX Activity	ปิด = ไม่ได้ทำงาน เขียว = กำลังรับ แดง = กำลังส่ง ส้ม = สลับการทำงานระหว่างกำลังส่ง และกำลังรับ
LED803	Channel 1 RX2 Activity	ปิด = ไม่มีไฟเลี้ยง เขียว = กำลังรับ
LED100	GPS lock indicator	ปิด = no lock เขียว = lock

จากตารางที่ 2.1 ระหว่าง USRP B210 ทำงาน หากไฟ LED กระพริบแสดงว่าเกิดสถานะ Underflow ขณะส่ง หรือเกิดสถานะ Overflow ขณะรับ บ่งบอกได้ว่าโฮสต์ไม่สามารถส่งหรือรับได้ด้วยการแซมเปิลที่ความถี่นี้

2.1.2 USRP E320

USRP E320 [2] รูปที่ 2.2 คือ SDR ที่ถูกพัฒนาโดย Ettus Research เป็น RF Front-End ที่สามารถนำมาประยุกต์ใช้งานได้หลากหลาย อาทิเช่น ใช้สำหรับวิเคราะห์สเปกตรัมของสัญญาณแบบ Real-Time, ใช้สำหรับจำลองระบบสื่อสารระหว่าง eNB และ UE หรือใช้สำหรับทำหน้าที่เป็นวิทยุสื่อสารบน UAV เป็นต้น

USRP E320 ใช้ 2x2 MIMO (2 RX และ 2 TX) ใช้งานครอบคลุมความถี่ตั้งแต่ 70 MHz ถึง 6 GHz แบนด์วิทที่สามารถใช้งานได้สูงสุดถึง 56 MHz มี Clock และ PPS Reference ภายในตัว และมี GPSDO (GPS Disciplined Oscillator) สำหรับการซิงโครไนซ์สัญญาณ GPS สามารถเชื่อมต่อกับ USRP E320 ผ่าน 1/10 GbE และ Aurora ด้วยพอร์ต SFP+, RJ45 สำหรับ 1 GbE, พอร์ต USB type A, Micro-USB (Serial Console, JTAG) และ GPIO (General Purpose Input/Output)

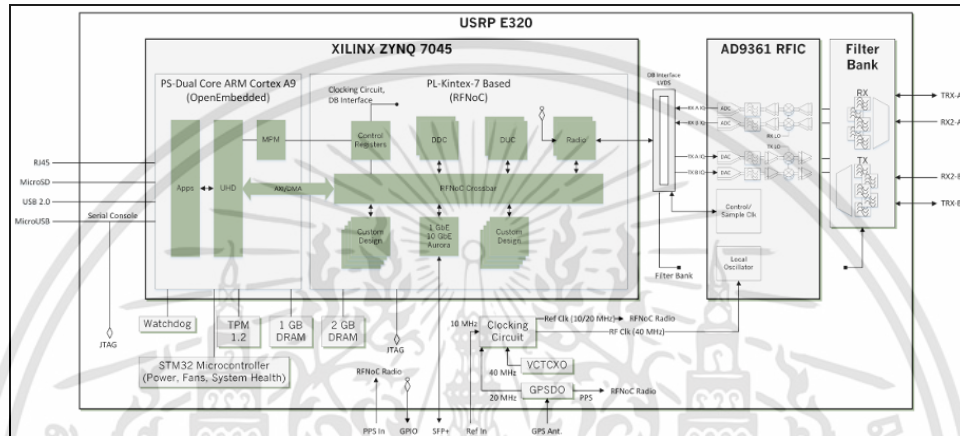


รูปที่ 2.2 USRP E320[4]

รูปที่ 2.3 คือบล็อกไดอะแกรมโครงสร้างภายในของ USRP E320 ซึ่งภายในประกอบด้วย 2 ส่วนหลักๆ ได้แก่ ส่วนที่เป็นหน่วยประมวลผลเบสแบนด์ซึ่งก็คือบอร์ด Xilinx Zynq 7045 SoC (System On Chip) และส่วนที่เป็น RFIC ซึ่งก็คือ IC AD9361

Xilinx Zynq 7045 SoC เป็น SoC ที่ส่วนประกอบ 2 ส่วนคือ PS (Processing Subsystem) – Dual Core ARM Cortex A9 ซึ่งเป็น Open Embedded และ PL (Programmable Logic) – Kintex-7 Based สำหรับการประมวลผลด้าน FPGA

AD9361 เป็น RFIC สำหรับทำหน้าที่เป็น DAC และ ADC ซึ่งเชื่อมต่อกับส่วนของบอร์ด SoC ด้วย LDVS Interface ก่อนส่ง – รับสัญญาณอนาล็อกด้วย 2x2 MIMO



รูปที่ 2.3 บล็อกไดอะแกรมโครงสร้างของ USRP E320 [5]

2.2 การสื่อสารทางดิจิทัล

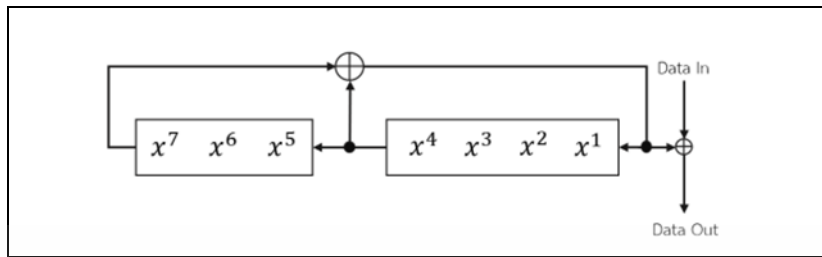
เป็นรูปแบบหนึ่งของการสื่อสารในโลกยุคปัจจุบัน มีเทคโนโลยีที่รองรับการสื่อสารเป็นจำนวนมากซึ่งลักษณะการสื่อสารทางดิจิทัลเป็นการส่งสัญญาณพัลส์ของข้อมูล และประมวลผลสัญญาณเพื่อให้สามารถส่ง และรับข้อมูลได้อย่างมีประสิทธิภาพ ตามมาตรฐานการใช้งานชนิดต่างๆ จึงได้ทำการออกแบบระบบการสื่อสารโดยมีองค์ประกอบของฟังก์ชันต่างๆ ดังต่อไปนี้

2.2.1 scramble

การสแครมเบิลเป็นวิธีการเข้ารหัสฝั่งส่งซึ่งในการถอดรหัสต้องใช้วงจรตีสแครมเบิลอร์ในการถอดรหัส เป็นการเข้ารหัสข้อมูลมีการส่งข้อมูลเป็นแบบ Serial-to-Serial โดยมีสมการพหุนามเป็นสมการการทำงานของระบบดังสมการที่ 2.1 โดยมีรูปแบบการทำงานดังรูปที่ 2.4 เพื่อให้ค่าความน่าจะเป็นของบิตที่ผ่านวงจรสแครมเบิลอร์มีค่าความน่าจะเป็นเท่ากันหรือไม่มีบิตข้อมูลเหมือนกันเรียงซ้ำต่อกันหลายบิต [6]

$$s(x) = x^7 + x^4 + 1 \quad (2.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 การทำงานสแควมเบิ้ล [6]

2.2.2 barker code

รหัสบาร์กเกอร์ถูกคิดค้นโดย R.H. Barker ในปี 1953 ในเทคโนโลยีด้านการสื่อสาร รหัสบาร์กเกอร์ หรือลำดับบาร์กเกอร์ เป็นลำดับจำกัดของค่าดิจิทัลพร้อมคุณสมบัติความสัมพันธ์อัตโนมัติในอุดมคติ ใช้เป็นรูปแบบการซิงโครไนซ์ระหว่างผู้ส่ง และผู้รับ Braker codes นิยมใช้สำหรับงานด้านเรดาร์ และงานด้านอื่นๆ เช่น 802.11b, แท็ก RFID, อัลตราซาวนด์ ฯลฯ

รหัส Barker หรือลำดับ Barker เป็นลำดับจำกัดของค่า N ที่ $+1$ และ -1 โดยหาค่า correlation (C_k) ได้ดังสมการที่ 2.2 [7]

$$c_k = \sum_{j=1}^{N-k} X_j X_{j+k} \quad (2.2)$$

ตารางที่ 2.2 ลำดับของ Barker ที่ Length ต่างๆ

Length	Codes	Sidelobe Level Ratio
2	+1 -1 +1 +1	-6 dB
3	+1 +1 -1	-9.5 dB
4	+1 +1 -1 +1 +1 +1 +1 -1	-12 dB
5	+1 +1 +1 -1 +1	-14 dB
7	+1 +1 +1 -1 -1 +1 -1	-16.9 dB
11	+1 +1 +1 -1 -1 -1 +1 -1 -1 +1 -1	-20.8 dB
13	+1 +1 +1 +1 +1 -1 -1 +1 +1 -1 +1 -1 +1	-22.3 dB

จากตารางที่ 2.2 จะเห็นได้ว่าจะการหาค่าความสัมพันธ์อัตโนมัติของลำดับ Barker ที่ Length แตกต่างกัน อัตราส่วนกำลังของไซด์โลบจะมีค่าที่ต่ำ

2.2.3 Finite Impulse Response Filter (FIR Filter)

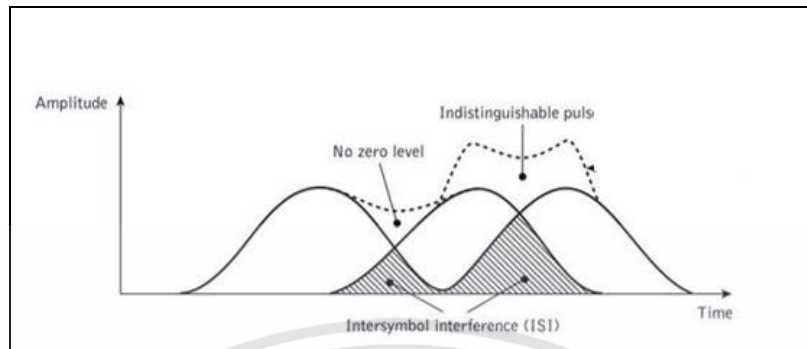
ตัวกรองมีผลตอบสนองต่ออิมพัลส์จำกัด ใช้ออกแบบและสร้างตัวกรองชนิดต่างๆ เช่น ตัวกรองความถี่ต่ำผ่าน (Low Pass Filter), ตัวกรองความถี่สูงผ่าน (High Pass Filter) เป็นต้น ความจำกัดอิมพัลส์ส่งผลให้สามารถเขียนตัวกรองได้โดยใช้สมการผลต่างสลับเนื่องดังสมการที่ 2.3 ที่มีสัมประสิทธิ์ a_k, b_m เป็นค่าคงที่ (Linear Constant-Coefficient Difference Equation) สัมประสิทธิ์เป็นค่าผ่านการคำนวณในการออกแบบตัวกรอง สมการที่ได้เป็นสมการในการสร้างวงจรดิจิทัล [8]

$$\sum_{k=0}^N a_k y[n-k] = \sum_{m=0}^M b_m [x-m] \quad (2.3)$$

2.2.4 Root Raised Cosine Filter (RRC Filter)

เป็นตัวกรองสัญญาณในการสื่อสารดิจิทัล ปรับสัญญาณให้มีรูปทรงเป็นไปตามผลตอบสนองอิมพัลส์โดยมีผลตอบสนองอิมพัลส์จำกัดโดยมีผลตอบสนองอิมพัลส์ดังสมการที่ 2.4 เพื่อลดการซ้อนทับกันของข้อมูล (Inter Symbol Interference : ISI) ดังรูปที่ 2.5 ระหว่างการส่งสัญญาณจะทำให้การส่งสัญญาณมีอัตราความผิดพลาด (Bit Error Rate) ที่เกิดขึ้นน้อยลง [9]

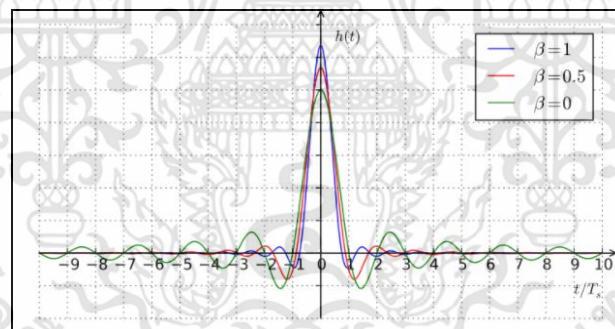
$$h(t) = \begin{cases} \frac{1}{T_s} \left(1 + \beta \left(\frac{4}{\pi} - 1 \right) \right), & t = 0 \\ \frac{\beta}{T_s \sqrt{2}} \left[1 + \frac{2}{\pi} \sin \left(\frac{\pi}{4\beta} \right) + \left(1 - \frac{2}{\pi} \right) \cos \left(\frac{\pi}{\beta} \right) \right], & t = \pm \frac{T_s}{4\beta} \\ \frac{1}{T_s} \frac{\sin \left[\pi \frac{t}{T_s} (1 - \beta) \right] + 4\beta \frac{t}{T_s} \cos \left[\pi \frac{t}{T_s} (1 + \beta) \right]}{\pi \frac{t}{T_s} \left[1 - \left(4\beta \frac{t}{T_s} \right) \right]}, & otherwise \end{cases} \quad (2.4)$$



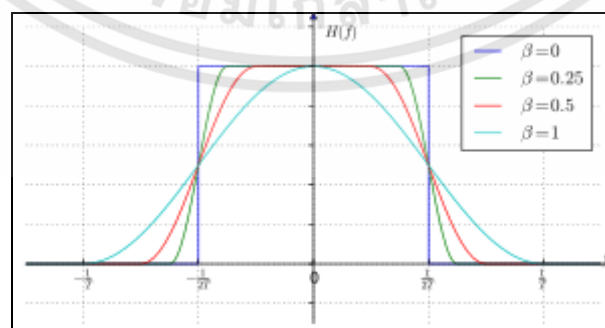
รูปที่ 2.5 การซ้อนทับกันของข้อมูล

2.2.5 Roll off vector (β)

เป็นค่าคงที่ในสมการ (β) อยู่ในช่วงระหว่าง 0-1 มีผลทำให้ขนาดของโพลบในโดเมนเวลา (Time Domain) มีขนาดลดลงเมื่อลดค่า β ลงดังรูปที่ 2.6 ช่วยป้องกันการรบกวนกันของข้อมูล และทำให้การแปลงสัญญาณอนาล็อกไปสัญญาณดิจิทัลทำได้ดีขึ้น แต่ทำให้มีขนาดแบนด์วิทเพิ่มขึ้นในโดเมนความถี่ (Frequency Domain) ดังรูปที่ 2.7 การใช้ค่าคงที่ขึ้นอยู่กับประเภทของการใช้งานในการส่งสัญญาณ [10]



รูปที่ 2.6 ขนาดของโพลบในโดเมนเวลา



รูปที่ 2.7 ขนาดแบนด์วิทเพิ่มขึ้นในโดเมนความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6 Interpolation

การประมาณค่าเป็นกระบวนการในการเพิ่มจุดข้อมูลใหม่ ภายในช่วงของชุดจุดข้อมูลที่ทราบค่า สามารถใช้ในการแก้ไขข้อมูลที่ขาดหายไป ปรับข้อมูลที่มีอยู่ให้ราบรื่น คาดการณ์ข้อมูลเป็นต้น [11]

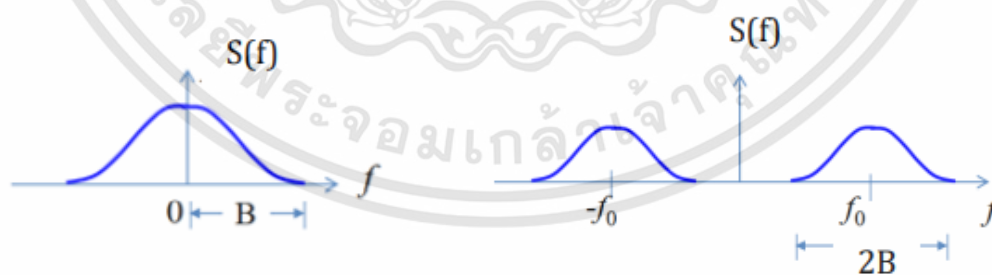
2.2.7 Sampling

การสุ่มตัวอย่างข้อมูลเป็นกระบวนการแปลงข้อมูลอนาล็อกเป็นข้อมูลดิจิทัลเพื่อส่งข้อมูลบนรูปแบบการสื่อสารแบบดิจิทัล โดยมีเงื่อนไขในการสุ่มตาม Nyquist–Shannon Sampling Theorem ว่าความถี่ในการสุ่มต้องมากกว่าสองเท่าของความถี่ข้อมูลดังสมการ 2.5 สามารถกู้คืนข้อมูลเดิมได้ [12]

$$f_s \geq 2f_m \quad (2.5)$$

2.2.8 Modulation

การมอดูเลชันเป็นกระบวนการการเปลี่ยนแปลงสัญญาณข้อมูลเพื่อให้ข้อมูลอยู่ในรูปแบบเหมาะสมในการส่งในสภาพแวดล้อมต่างๆ โดยการมอดูเลชันแบ่งเป็น 2 ประเภท คือ การมอดูเลชันแบบแถบฐาน (Baseband Modulation) โดยมีความถี่ศูนย์กลางเท่ากับ $f_c = 0$ ดังรูป 2.8 (ก) และการมอดูเลชันแบบแถบผ่าน (Bandpass Modulation) โดยมีความถี่ศูนย์กลางเท่ากับ $f_c = f_0$ ดังรูป 2.8 (ข) [13]



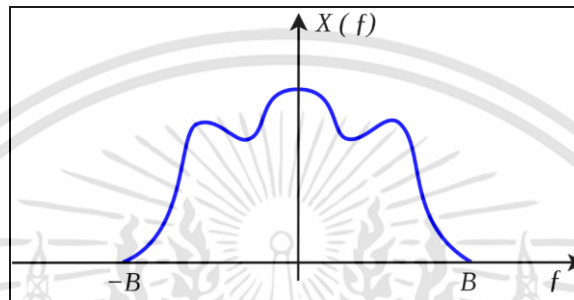
(ก) Baseband

(ข) Bandpass

รูปที่ 2.8 สเปกตรัมของสัญญาณ Baseband และ Bandpass

2.2.8.1 Baseband Modulation

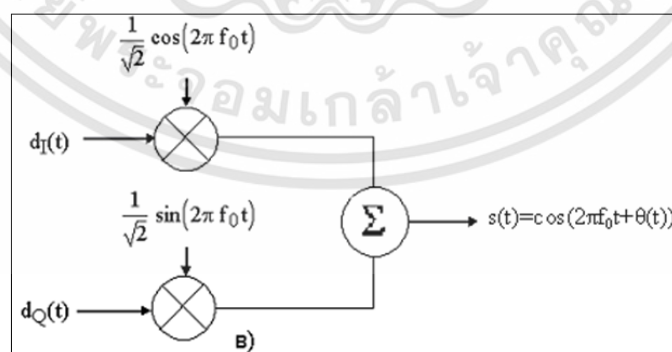
การมอดูเลตแถบฐานเป็นการส่งบิตข้อมูลให้เป็นแรงดันพัลส์ โดยพัลส์ที่เลือกใช้เป็นพัลส์สัญญาณต่อเนื่องที่มีความถี่ศูนย์กลางเท่ากับศูนย์เฮิรตซ์ ($f_c=0$) โดยมีผลตอบสนองของความถี่ดังรูปที่ 2.9



รูปที่ 2.9 ผลตอบสนองของความถี่

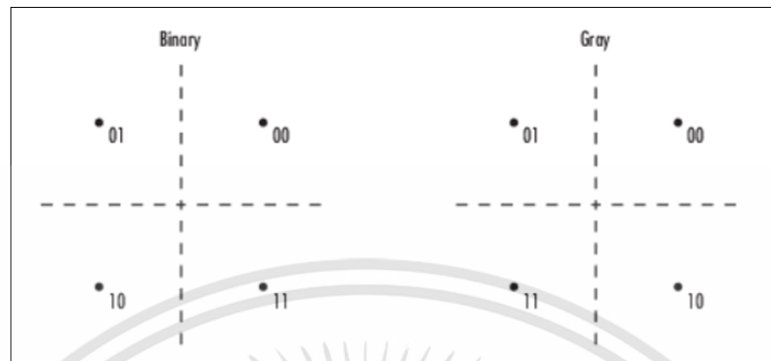
2.2.8.2 Quadrature Phase Shift Keying baseband

เป็นการมอดูเลชันเฟส (Phase Shift Keying) ที่ใช้เฟสต่างกันในการส่งข้อมูลแต่ละบิตเพื่อเป็นประโยชน์ในการส่ง และการรับข้อมูลโดยการมอดูเลชันแบบ QPSK เป็นการมอดูเลชันที่มีลักษณะการส่งครึ่งละสองบิตแบ่งเป็นบิตคู่ และบิตคี่ตามลักษณะดังรูป 2.10 ซึ่งเป็นประโยชน์ต่อการส่งข้อมูลเพื่อให้ส่งข้อมูลได้เร็วขึ้น และใช้แบนด์วิทได้อย่างมีประสิทธิภาพ และใช้ฟังก์ชันในการมอดูเลชันเป็นการมอดูเลชันแบบเกรย์โค้ดซึ่งดีต่ออัตราบิตผิดพลาดมีประสิทธิภาพดีกว่าการส่งแบบไบนารีเพราะเกรย์โค้ดทำให้บิตที่ใกล้เคียงกันมีค่าต่างกันเพียงบิตเดียวดังรูป 2.11 [14]



รูปที่ 2.10 บล็อกไดอะแกรม QPSK Modulate

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 QPSK Constellation Diagram

2.2.9 ผลกระทบในช่องสัญญาณ

2.2.9.1 สัญญาณรบกวนเกาส์สีขาวแบบบวก (Additive White Gaussian Noise, AWGN)

สัญญาณรบกวนเกาส์สีขาวแบบบวก (Additive White Gaussian Noise, AWGN) คือสัญญาณรบกวนที่มีลักษณะการกระจายตัวแบบเกาส์ที่มีค่าเฉลี่ยเป็นศูนย์ และมีค่าความแปรปรวนใดๆ [15]

2.2.9.2 การแทรกสัญญาณข้าม (Crosstalk)

การรบกวนกันของสัญญาณกันของสัญญาณ เกิดขึ้นขณะส่งสัญญาณในสายส่งสัญญาณด้วยความเร็วสูง ปรากฏการณ์นี้มักพบได้เมื่อใช้ส่งสัญญาณตั้งแต่สองคู่ขึ้นไป [16]

2.2.9.3 ความหน่วงเวลา (Time Delay)

การส่งสัญญาณผ่านตัวกลางชนิดต่างๆ ที่มีความแตกต่างกันจะส่งผลกระทบต่อประสิทธิภาพของสัญญาณหรือเกิดความหน่วงเกิดขึ้นเนื่องจากสภาพตัวกลางทำให้เวลาในการส่งสัญญาณคลาดเคลื่อน

2.2.9.4 ความคลาดเคลื่อนของสัญญาณ

การส่งสัญญาณผ่านตัวกลางทำให้สัญญาณเปลี่ยนแปลงหรือเกิดความคลาดเคลื่อนได้หลายอย่าง เช่น เฟสที่คลาดเคลื่อน, ความถี่ที่ต่างไปจากเดิม เป็นต้น

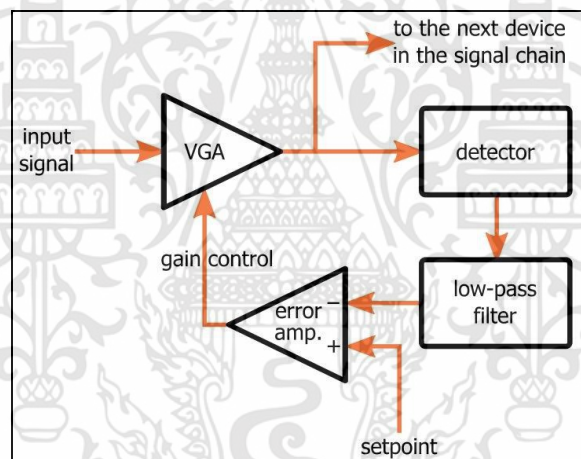
การสื่อสารดิจิทัลจะมีการเปลี่ยนแปลงสัญญาณในรูปแบบต่างๆ เพื่อให้การสื่อสารมีความถูกต้อง และปลอดภัย เช่น การเข้ารหัสข้อมูล, การมอดูเลชัน, การเพิ่มการสุ่มข้อมูล (Upsampling), การลดการสุ่มข้อมูล (Downsampling) และผ่านตัวกรองชนิดต่างๆ เป็นต้น เมื่อ

สัญญาณถูกส่งผ่านอากาศในรูปแบบคลื่นความถี่ทำให้สัญญาณเกิดสัญญาณรบกวนต่างๆ เมื่อฝั่งรับรับข้อมูลจะต้องแก้ไขปัญหาที่เกิดขึ้น เช่น Symbol Synchronizer และ Carrier Synchronizer จะแก้เรื่องการซิงค์กันของสัญญาณ เป็นต้น

ฝั่งรับข้อมูลจะต้องถอดรหัสหรือทำการถอดข้อมูลที่ถูกรับเข้ารหัสมาเพื่อให้ได้ข้อมูลที่ฝั่งส่ง ส่งมาในรูปแบบสมมาตร (Symmetry) กับฝั่งส่ง

2.2.10 Automatic Gain Control (AGC)

การควบคุมอัตราขยายอัตโนมัติโดยลักษณะของวงจรเป็นระบบแบบลูปปิด (Close Loop System) ที่สามารถปรับ อัตราขยายแรงดันโดยอัตโนมัติ ใช้วงจรแอมพลิฟายเออร์ (Amplifier) เป็นวงจรขยายสัญญาณ เพื่อให้แรงดันด้านออกมีค่าคงที่ ภายในย่านที่กำหนดโดยมีวงจรมีดังรูป 2.12



รูปที่ 2.12 ตัวอย่างวงจรควบคุมอัตราขยายตัวอัตโนมัติ [17]

2.2.11 Receiver RRC (Root Raise Cosine)

ฝั่งส่งใช้ RRC Filter เป็นตัวกรองสัญญาณผ่าน การส่งสัญญาณผ่านอากาศมีสัญญาณรบกวนเกิดขึ้น (Noise Signal) ใช้ RRC Filter เพื่อจำกัดขอบเขตของสัญญาณรบกวนใน ทำให้ได้สัญญาณที่ผ่านตัวกรองมีคุณภาพของสัญญาณดีขึ้น

2.2.12 Coarse Frequency Compensation

ขดเคียวความถี่ออฟเซตของคลื่นพาห์ (Carrier Frequency Offset) มีอัลกอริทึมในการประมาณดังนี้ [18]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.12.1 Correlation-Based

อัลกอริทึมการประมาณค่า Correlation-Based ใช้ในการประมาณความถี่ออฟเซต (Frequency Offset) ของสัญญาณ PSK และ PAM

ในการกำหนดความถี่ออฟเซต Δf อัลกอริทึมจะทำการประมาณค่าความน่าจะเป็นสูงสุด Maximum Likelihood (ML) ของ Complex-Valued Oscillation $e^{j2\pi\Delta f t}$ สัญญาณที่ได้แสดงด้วยสมการดังต่อไปนี้ [19]

$$r_k = e^{j2\pi\Delta f k T_s + \theta}, 1 \leq k \leq N \quad (2.6)$$

เมื่อ T_s คือ ช่วงเวลาของการสุ่มตัวอย่าง

θ คือ Unknown Random Phase

N คือ จำนวนของการสุ่มตัวอย่าง

การประมาณค่าความน่าจะเป็นสูงสุดของการชดเชยความถี่เท่ากับการหาค่าสูงสุดของฟังก์ชันความน่าจะเป็น

$$\Lambda_{\Delta f} \approx \left| \sum_{i=1}^N r_i e^{-j2\pi\Delta f i T_s} \right|^2 = \sum_{k=1}^N \sum_{m=1}^N r_k r_m^* e^{-j2\pi\Delta f T_s (k-m)} \quad (2.7)$$

จากนั้น ใช้การแปลงฟูเรียร์แบบไม่ต่อเนื่อง (Discrete Fourier Transform) ถ่วงน้ำหนักโดย Parabolic Windowing Function จะแสดงได้เป็น

$$\text{Im} \left\{ \sum_{k=1}^{N-1} k(N-k) R(k) e^{j2\pi\Delta f \wedge T_s} \right\} = 0 \quad (2.8)$$

โดยที่ $R(k)$ คือ Autocorrelation ที่ถูกประมาณแล้วของลำดับ r_k แสดงได้ดังนี้

$$R(k) \triangleq \frac{1}{N-k} \sum_{i=k+1}^N r_i r_{i-k}^*, 0 \leq k \leq N-1 \quad (2.9)$$

เมื่อ $k(N-k)$ คือ Parabolic Windowing Function $R(k)$ คือ การประมาณค่าอย่างต่ำของ Autocorrelation ของ r_k เมื่อ $k=0$ หรือ k เข้าใกล้ N ดังนั้น Windowing Function สามารถทำให้เป็น Rectangular Sequence 1's สำหรับ $k=1, 2, \dots, L$

โดยที่ $L \leq N-1$ ผลที่ได้ Nearest Integer Function คือค่าการประมาณค่าความน่าจะเป็นสูงสุด (ML) ที่ปรับแล้ว

$$\text{Im} \left\{ \sum_{k=1}^L R_k e^{-j2\pi\Delta f \hat{u}_k T_s} \right\} = 0 \quad (2.10)$$

ผลลัพธ์ของการประมาณ Δf แสดงได้ดังนี้

$$\hat{\Delta f} \cong \frac{f_{\text{samp}}}{\pi(L+1)} \arg \left\{ \sum_{k=1}^L R_k \right\} \quad (2.11)$$

f_{samp} ความถี่ในการสุ่มตัวอย่าง เป็นส่วนกลับของ T_s จำนวนขององค์ประกอบที่ใช้ในการคำนวณลำดับ Autocorrelation คำนวณได้จาก

$$L = \text{round} \left(\frac{f_{\text{samp}}}{f_{\text{max}}} \right) - 1 \quad (2.12)$$

โดยที่ f_{max} คือค่าคาดการณ์ความถี่สูงสุด และ Round คือ ฟังก์ชันจำนวนเต็มใกล้สุด (Nearest Integer Function) ค่าการประมาณความถี่ที่ขึ้นเมื่อ $L \geq 7$

และนำไปสู่ข้อเสนอที่ $f_{\text{max}} \leq \frac{f_{\text{samp}}}{4m}$

2.2.12.2 FFT-Based

อัลกอริทึมสามารถใช้ในการประมาณการชดเชยความถี่ออฟเซตสำหรับการมอดูเลตทุกแบบ มี 2 รูปแบบในการชดเชยความถี่แบบหยาบ

1) สำหรับการมอดูเลตแบบ BPSK, QPSK, 8PSK, PAM หรือ QAM อัลกอริทึมในการประมาณค่า Δf โดยใช้ periodogram ของกำลัง m^{th} ของสัญญาณที่รับได้ มีค่าดังนี้

$$\hat{\Delta f} \cong \frac{f_{\text{samp}}}{N \cdot m} \arg \max_f \left| \sum_{k=0}^{N-1} r^m(k) e^{-j2\pi k t / N} \right|, \left(\frac{-R_{\text{sym}}}{2} \leq f \leq \frac{R_{\text{sym}}}{2} \right) \quad (2.13)$$

โดยที่ m คือ Modulation Order
 $r(k)$ คือ Received Sequence
 R_{sym} คือ Symbol Rate
 N คือ จำนวนการสุ่มตัวอย่าง

อัลกอริทึมในการหาความถี่ที่เพิ่มค่าเฉลี่ยของเวลาที่กำลัง m^{th} ของสัญญาณที่รับได้คูณด้วยความถี่ต่างๆ ในช่วงของ $[-R_{\text{sym}}/2, R_{\text{sym}}/2]$ เนื่องจากรูปแบบของอัลกอริทึมนิยามโดยการแปลงฟูเรียร์แบบไม่ต่อเนื่อง ของ $r^m t$ การหาความถี่ที่เพิ่มค่าเฉลี่ยของเวลาจะเท่ากับเส้นที่มีค่าสูงสุดในสเปกตรัมของ $r^m t$ จำนวนจุดที่ FFT ต้องการคือ

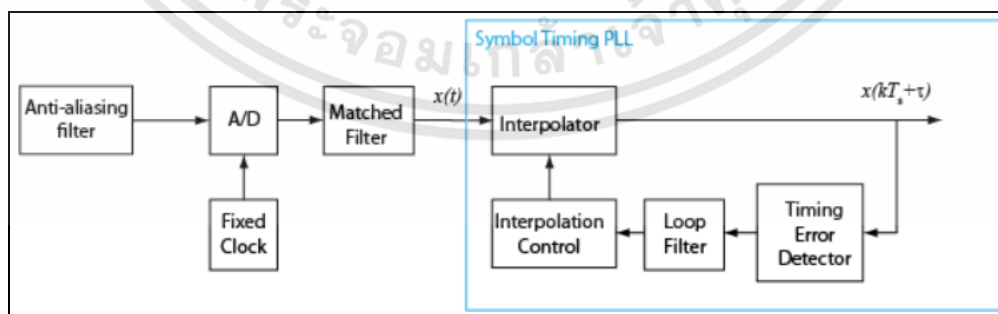
$$N = 2 \left\lceil \log_2 \left(\frac{f_{\text{samp}}}{f_r} \right) \right\rceil \quad (2.14)$$

โดยที่ f_r คือ Frequency Resolution ที่ต้องการ

2) สำหรับการมอดูเลตแบบ OQPSK อัลกอริทึมจะค้นหาสูงสุดของสเปกตรัมที่ ± 200 kHz ประมาณ Symbol Rate เทคนิคนี้หาค่าสูงสุดที่ต้องการเมื่อมีสัญญาณรบกวนจาก Spectral Content ที่ Baseband Frequency จากการ Filter

2.2.13 Symbol Synchronizer

อัลกอริทึมการซิงก์โครโนซ์เวลา Symbol ใช้อัลกอริทึมของ Phased Lock Loop (PLL) บล็อกไดอะแกรมนี้แสดงตัวอย่างของการซิงก์โครโนซ์เวลา [20]



รูปที่ 2.13 อัลกอริทึม Symbol Timing PLL

2.2.14 Carrier Synchronizer

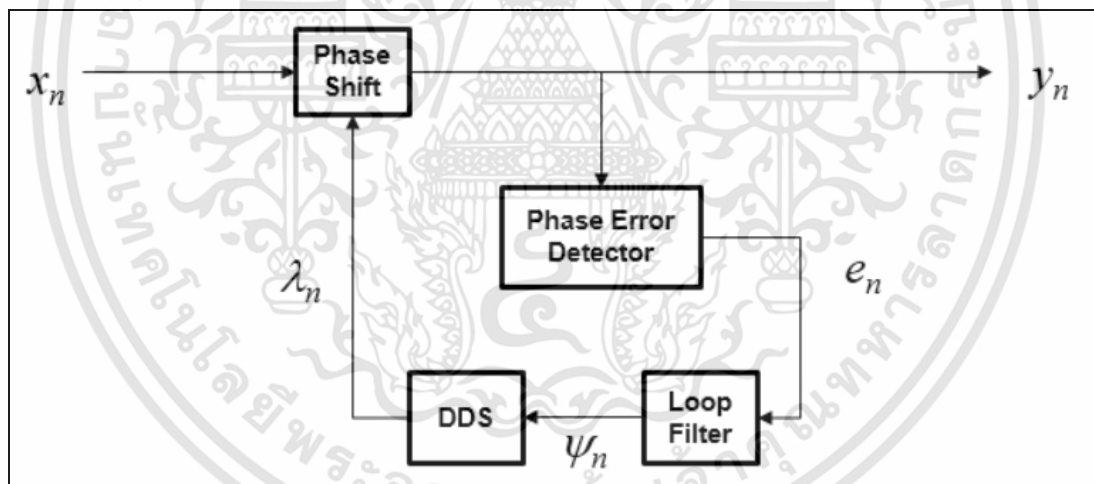
การชดเชย Carrier Phase และ Frequency Offset แบบ Closed-Loop โดยใช้หลักการ Phase-Locked Loop (PLL) [21]

y_n เอาต์พุตของการซิงโครไนซ์ x_n เป็น Frequency-Shifted ของสัญญาณอินพุตเชิงซ้อน และ n คือจำนวนการสุ่มตัวอย่าง เอาต์พุตของการซิงโครไนซ์แสดงได้ดังนี้

$$y_n = y_n e^{i\lambda_n} \quad (2.15)$$

โดยที่ λ_n คือเอาต์พุต Direct Digital Synthesizer (DDS) DDS เป็น Voltage-Controlled Oscillator และเป็นองค์ประกอบหลักของ PLL โดย DDS ทำหน้าที่เป็น Integration Filter ที่ควบคุมโดย

$$\psi_n = g_1 e_n + \psi_{n-1} \quad (2.16)$$



รูปที่ 2.14 อัลกอริทึม Phase-locked loop (PLL)

เพื่อแก้ความถี่ออฟเซต ขั้นแรกกำหนด Phase Error e_n โดย Phase Error ขึ้นอยู่กับรูปแบบของการมอดูเลต โดย Phase Error ของ QPSK Modulate คือ

$$e_n = \text{sgn Re } x_n \times \text{Im } x_n - \text{sgn Im } x_n \times \text{Re } x_n \quad (2.17)$$

2.2.15 Preamble Detector

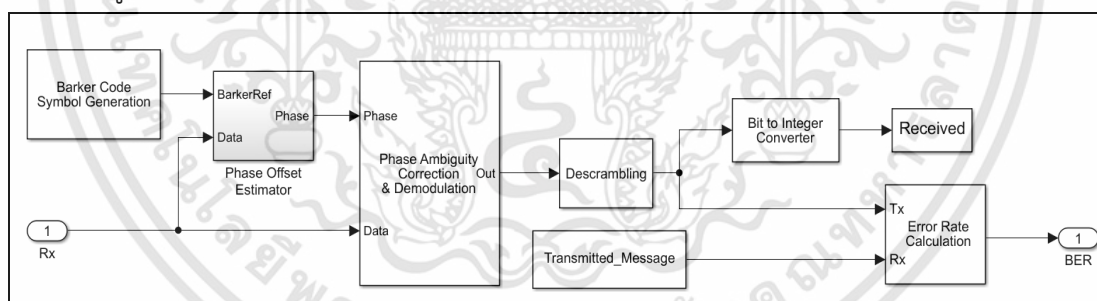
Bit Inputs เมื่อข้อมูลที่ป้อนเข้าไปประกอบด้วยบิตข้อมูล Preamble Detector จะใช้แพตเทิร์นในการแมตช์ Input Symbols เมื่อข้อมูลที่ป้อนเข้าไปประกอบด้วยสัญลักษณ์ (Symbol) Preamble Detector จะใช้อัลกอริทึม Cross-Correlation [22]

2.2.16 Frame Synchronizer

ในการสื่อสารโทรคมนาคม การซิงโครไนซ์เฟรมหรือการทำเฟรมเป็นกระบวนการที่ในขณะที่รับข้อมูลที่เป็นเฟรม จะมีการจัดเรียงสัญญาณเพื่อระบุตำแหน่งของข้อมูล เพื่อที่จะทำการแยกข้อมูลภายในเฟรมไปถอดรหัส หรือดำเนินการส่งสัญญาณใหม่ [23]

2.2.17 BER Data Decoding

ภายในบล็อก BER Data Decoding จะเป็นขั้นตอนการจัดการกับข้อมูลหลังทำการซิงโครไนซ์เฟรมสำเร็จ ข้อมูลที่ทำการถอด Preamble ออกแล้วจะทำการเปรียบเทียบเฟสของสัญญาณที่ได้รับได้ แล้วทำการดีโมดูเลตสัญญาณ QPSK และ Descrambling เพื่อให้ได้สัญญาณที่เป็นข้อมูลเดิมกลับคืนมา และเปรียบเทียบข้อมูลระหว่างภาครับ และภาคส่ง เพื่อหา Bit Error Rate (BER) ดังรูปที่ 2.15

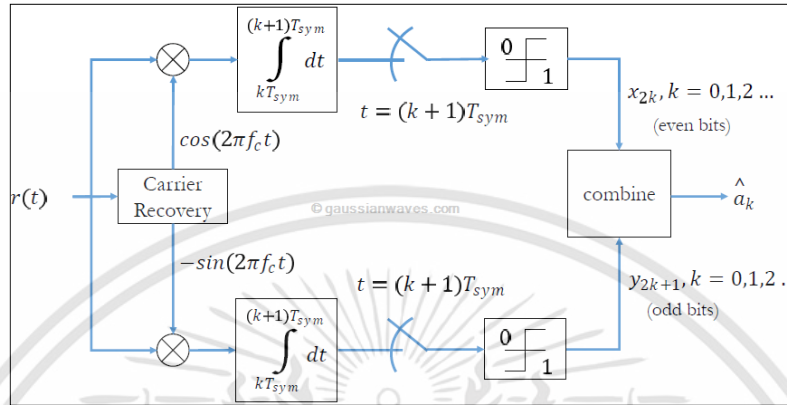


รูปที่ 2.15 โมดูลภายในบล็อก BER Data Decoding

2.2.18 QPSK Demodulator

เนื่องจากการส่งข้อมูลมีการมอดูเลชันแบบ QPSK จึงจำเป็นต้องใช้การดีมอดูเลชันแบบ QPSK เพื่อถอดข้อมูลที่ใช้ในการส่งออกมา การดีมอดูเลชันแบบ QPSK จะรับข้อมูลที่ส่งผ่านมาและใช้วิธีการ Correlator และการตัดสินใจเพื่อดึงข้อมูลกลับมาโดยจะได้ข้อมูลออกมาเป็นบิตคู่

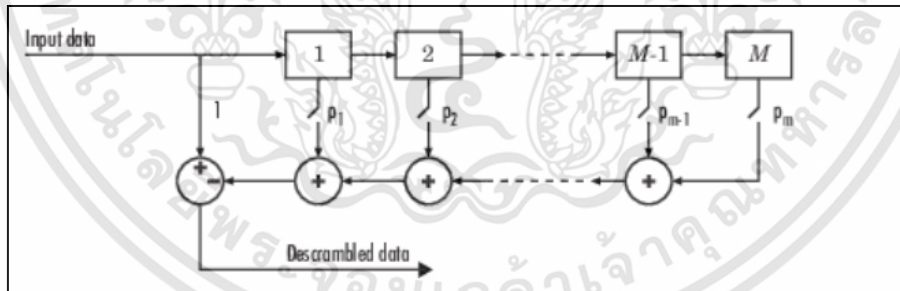
(Even Bits) และบิตคี่ (Odd Bits) เมื่อนำมารวมกันก็จะได้ข้อมูลชุดเดิมก่อนเข้า QPSK Modulation แสดงดังรูปที่ 2.16



รูปที่ 2.16 QPSK Modulation

2.2.19 Descrambler

จากการส่งข้อมูลการจัดเรียงข้อมูลโดยใช้สแครมเบิลในการช่วยจัดการข้อมูลจึงมีความจำเป็น ต้องมีตัวถอดรหัสจากการเข้ารหัสของสแครมเบิลคือตัวดีสแครมเบิลโดยใช้สมการพหุนามสมการเดียวกันในการถอดรหัสเพื่อที่จะได้ข้อมูลเดิมก่อนเข้ารหัสสแครมเบิลกลับมา โดยมีรูปแบบการถอดรหัสดังรูปที่ 2.17



รูปที่ 2.17 วงจรดีสแครมเบิล

2.3 พารามิเตอร์ที่ใช้วัด

การทดสอบระบบสื่อสาร QPSK ที่ได้ออกแบบนี้ ได้ทำการจัดเก็บผลการทดสอบด้วยค่าพารามิเตอร์ต่างๆ จากซอฟต์แวร์ และเครื่อง Spectrum Analyzer เพื่อใช้วิเคราะห์ และ ประเมินผลของระบบสื่อสาร โดยมีการจัดเก็บพารามิเตอร์ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1 อัตราความผิดพลาดบิต (BER)

อัตราความผิดพลาดบิตเป็นพารามิเตอร์ที่วัดประสิทธิภาพของช่องส่งข้อมูลโดยส่งข้อมูลจากจุดหนึ่งไปยังจุดหนึ่งผ่านช่องสัญญาณมีสายหรือไร้สายสามารถวัดอัตราบิตผิดพลาดได้ที่ปลายทาง โดยเป็นค่าอัตราส่วนของจำนวนบิตที่ภาครับตีความผิดพลาดต่อจำนวนบิตที่ถูกส่งมาทั้งหมดภายในช่วงเวลาหนึ่ง

$$\text{BER} = \frac{\text{Errors}}{\text{Total Number of Bits}} \quad (2.18)$$

2.3.2 กำลังของสัญญาณ

กำลังของสัญญาณเป็นค่าที่เกิดจากสัญญาณรูปแบบต่างๆ โดยสัญญาณที่มีความแตกต่างกันจะมีกำลังของสัญญาณแตกต่างกันไปด้วย โดยแบ่งเป็นสมการของสัญญาณต่อเนื่องกับสมการของสัญญาณไม่ต่อเนื่อง [24]

$$P_\infty = \lim_{T \rightarrow \infty} \frac{1}{2T} \int_{-T}^T |x(t)|^2 dt \quad (2.19)$$

$$P_\infty = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^N |x(n)|^2 \quad (2.20)$$

2.3.3 แบนด์วิท (Bandwidth)

เป็นการวัดความกว้างของสเปกตรัมของสัญญาณ สเปกตรัมของสัญญาณพัลส์จะแปรผกผันกับขนาดของพัลส์ในแต่ละคาบ โดยสามารถวัดประสิทธิภาพของแบนด์วิทได้จากอัตราส่วนระหว่างอัตราเร็วบิตต่อแบนด์วิทของพัลส์สัญญาณ [25]

$$\eta = \frac{R_b}{B} \left(\frac{\text{bps}}{\text{Hz}} \right) \quad (2.21)$$

2.3.4 อัตราส่วนของสัญญาณคลื่นพาท่ต่อสัญญาณรบกวน (C/N)

อัตราส่วนของสัญญาณคลื่นพาท่ต่อสัญญาณรบกวนเป็นพารามิเตอร์ที่ใช้วัดประสิทธิภาพของสัญญาณเมื่อสัญญาณถูกส่งโดยมีคลื่นพาท่ผ่านสัญญาณรบกวน [26]

$$\frac{C}{N} = P_{s \min} - 10 \log k T_0 B \quad (2.22)$$

2.3.5 ขนาดเวกเตอร์ข้อผิดพลาด (EVM)

ขนาดเวกเตอร์ข้อผิดพลาด หรือ Error Vector Magnitude หรือ EVM เป็นบ่งชี้คุณภาพของการทำงานของเครื่องส่ง และเครื่องรับสัญญาณวิทยุดิจิทัล ในการทดสอบนี้เป็นการมอดูเลตแบบ QPSK ซึ่งจะมี Constellation Point ดังรูปที่ 2.11 ในการส่งสัญญาณผ่านช่องสัญญาณ ทำให้ตำแหน่งของ Constellation Point คลาดเคลื่อน ซึ่ง EVM ก็คือค่าความแตกต่าง Vector ระหว่าง Transmitted Symbol Vector กับ Ideal Symbol Vector สามารถคำนวณได้จากสมการที่ 2.23

$$\text{EVM dB} = 20\log_{10}\left(\frac{\text{EVM \%}}{100 \%}\right) \quad (2.23)$$

2.3.6 อัตราส่วนข้อผิดพลาดในการมอดูเลต (MER)

อัตราส่วนข้อผิดพลาดในการมอดูเลต หรือ Modulation Error Ratio หรือ MER เป็นตัวชี้วัดประสิทธิภาพของเครื่องส่ง หรือเครื่องรับสัญญาณวิทยุดิจิทัล เช่นเดียวกันกับค่า EVM โดย MER คำนวณได้จากอัตราส่วนของค่าเฉลี่ยกำลังสองของกำลังของสัญญาณตั้งต้นส่วนด้วยกำลังของความผิดพลาดดังสมการ 2.24 และ 2.25

$$\text{MER(dB)} = 10\log_{10}\left(\frac{P_{\text{signal}}}{P_{\text{error}}}\right) \quad (2.24)$$

$$\text{MER(\%)} = \sqrt{\frac{P_{\text{signal}}}{P_{\text{error}}}} \times 100\% \quad (2.25)$$

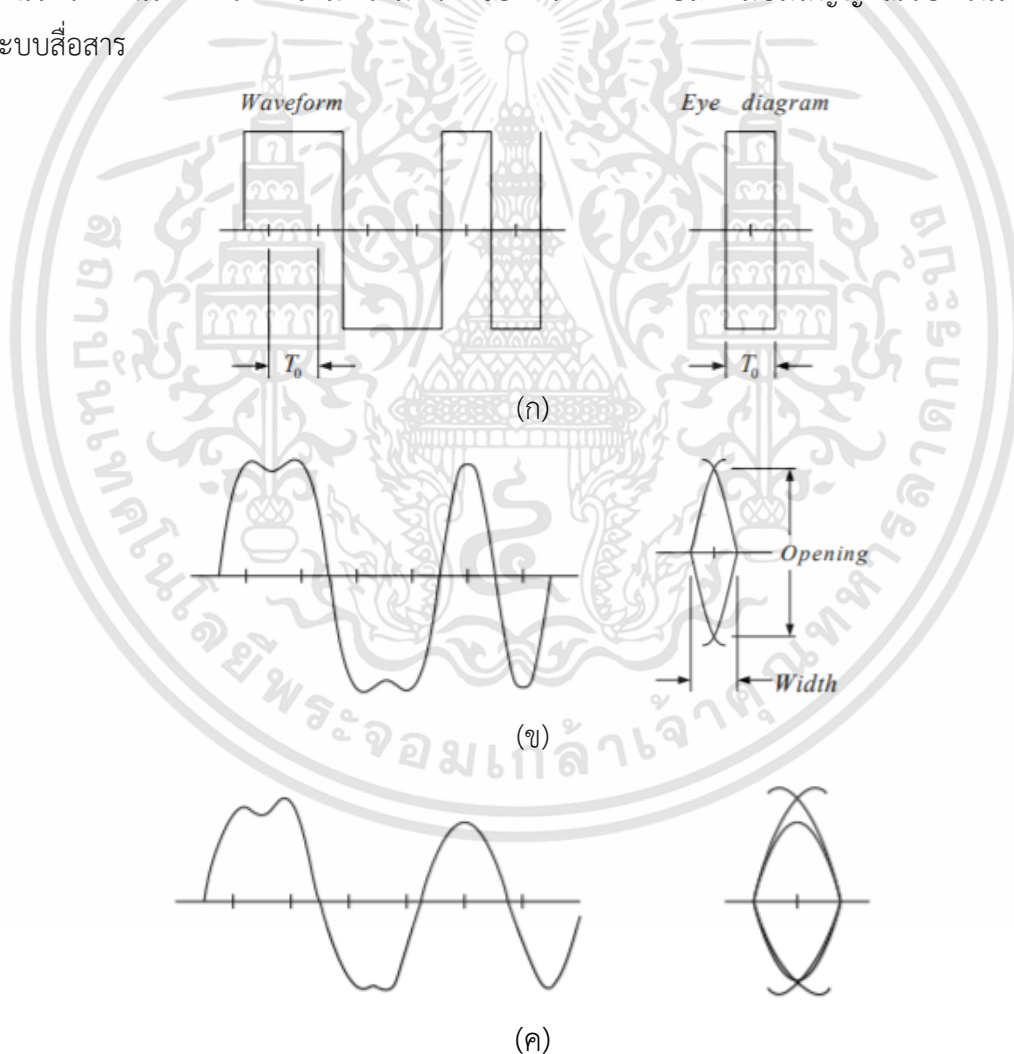
2.3.7 แผนภาพคอนสเทลเลชัน (Constellation Diagram)

แผนภาพคอนสเทลเลชัน หรือ Constellation Diagram เป็นแผนภาพแสดงตำแหน่งของข้อมูลต่อ 1 Symbol ของสัญญาณที่ถูกมอดูเลต ซึ่งแสดงในรูปแบบ 2 มิติ โดย แกน I (In Phase) ซึ่งเหมือนแกน X และแกน Q (Quadrature Phase) ซึ่งเหมือนแกน Y

การสร้างแผนภาพคอนสเทลเลชันของสัญญาณนั้น จะเป็นการแสดงการมอดูเลตของสัญญาณบนคลื่นพาห์ (Carrier) โดยในการทดลองเป็นการมอดูเลตแบบ QPSK ดังนั้นแผนภาพคอนสเทลเลชันของสัญญาณ QPSK จึงมีลักษณะเป็น 4 จุด โดยมีเฟสต่างกันเท่ากับ $\pi/4$ ดังรูปที่ 2.11

2.3.8 แผนภาพดวงตา (Eye Diagram)

แผนภาพดวงตา หรือ Eye Diagram เป็นตัวชี้วัดคุณภาพของสัญญาณแบบเบื้องต้น สัญญาณที่ถูกส่งผ่านช่องสัญญาณจะถูกรบกวนด้วยสัญญาณรบกวนทำให้ส่งผลต่อแผนภาพดวงตา ลักษณะของแผนภาพดวงตาในอุดมคติจะมีลักษณะที่เป็นเส้นเรียบคม และดวงตาเปิดกว้าง จากรูป 2.18 (ก) เป็นการสร้างแผนภาพดวงตา โดยสัญญาณอินพุตเป็นสัญญาณพัลส์ ซึ่งแผนภาพดวงตาจะเกิดจากการนำสัญญาณพัลส์มาเรียงซ้อนทับกันเป็นแนวตั้ง ในช่วงเวลาเท่ากับ T_0 รูปที่ 2.18 (ข) เป็นการสร้างแผนภาพดวงตา โดยอินพุตเป็นสัญญาณที่ถูกมอดูเลต โดยไม่มีผลกระทบจากสัญญาณรบกวน และรูปที่ 2.18 (ค) เป็นแผนภาพดวงตาของสัญญาณที่ถูกรบกวนจากสัญญาณรบกวน จะเห็นได้ว่าแผนภาพดวงตาจะมีความกว้างของดวงตาที่แคบลง เมื่อมีสัญญาณรบกวนภายในระบบสื่อสาร



รูปที่ 2.18 แผนภาพดวงตา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ซอฟต์แวร์

2.4.1 MATLAB R2021b

เป็นซอฟต์แวร์ที่ใช้ในการคำนวณทางคณิตศาสตร์ และจำลองการทำงานของฟังก์ชันต่างๆ ทางวิศวกรรมศาสตร์

2.4.1.1 Simulink

เป็นฟังก์ชันหนึ่งในซอฟต์แวร์แมทแลปที่ใช้สร้างโครงสร้างการทำงานของระบบต่างๆ โดยสามารถเรียกใช้ฟังก์ชันใน Simulink Library แทนการคำนวณของฟังก์ชันต่างๆ ได้ และสามารถซิมูเลชันดูการทำงานของระบบที่ได้ออกแบบไว้ได้

2.4.1.2 Communications Toolbox

เป็นส่วนเสริมสำหรับซอฟต์แวร์แมทแลป โดยจัดเก็บฟังก์ชันการทำงานของอัลกอริทึมที่ใช้สำหรับการออกแบบระบบสื่อสาร อาทิเช่น การเข้ารหัส (Encode), การกรองสัญญาณ (Filter), การสแครมบลิง (Scrambling), การมอดูเลตสัญญาณ (Modulation) และการดีมอดูเลตสัญญาณ (Demodulation) เป็นต้น

2.4.1.3 Computer Vision Toolbox

เป็นส่วนเสริมของซอฟต์แวร์แมทแลป ใช้สำหรับทำงานอัลกอริทึมของการออกแบบ และทดสอบการแสดงผลบนคอมพิวเตอร์ อาทิเช่น โมเดล 3D, เสียง, รูปภาพ และวิดีโอ เป็นต้น

2.4.1.4 Communications Toolbox Support Package for USRP

Radio

เป็นส่วนเสริมของซอฟต์แวร์แมทแลป ใช้สำหรับติดตั้งเฟิร์มแวร์ ในการเชื่อมต่อระหว่างซอฟต์แวร์แมทแลป และฮาร์ดแวร์ SDR ที่เป็นผลิตภัณฑ์ประเภท USRP

2.4.1.5 HDL Coder

เป็นส่วนเสริมของซอฟต์แวร์แมทแลป ทำหน้าที่สำหรับแปลงสคริปต์แมทแลป หรือแบบบนซอฟต์แวร์ Simulink เป็นภาษาสำหรับใช้กับอุปกรณ์ประเภท FPGA ได้แก่ ภาษา VHDL และภาษา Verilog และยังสามารถทำงานร่วมกับซอฟต์แวร์ อาทิเช่น Vivado เพื่อโปรแกรมอุปกรณ์ประเภท FPGA ผ่านซอฟต์แวร์แมทแลป

2.4.1.6 HDL Coder Support Package for Xilinx Zynq Platform

เป็นอีกหนึ่งส่วนเสริมของซอฟต์แวร์เมทแลป ใช้สำหรับติดตั้งเฟิร์มแวร์ของหน่วยประมวลผลของอุปกรณ์ประเภท FPGA ในที่นี้คือแพลตฟอร์มของ Xilinx Zynq เพื่อให้ซอฟต์แวร์เมทแลปสามารถเชื่อมต่อกับอุปกรณ์ประเภท FPGA ได้

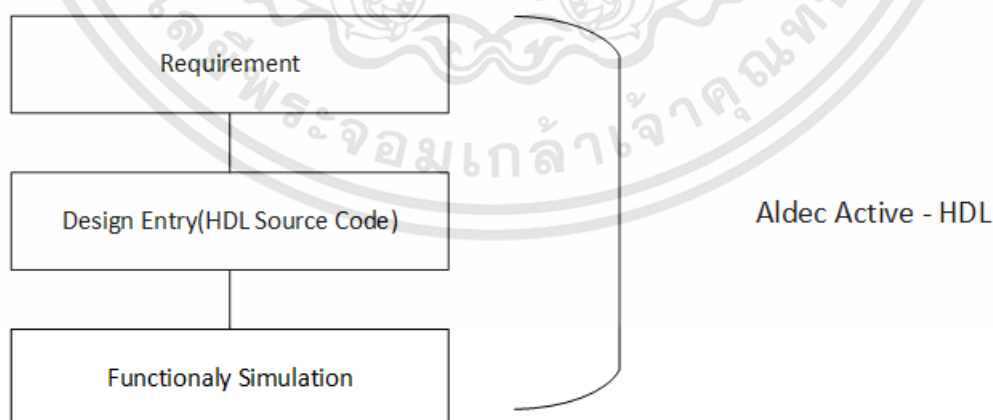
2.4.2 Aldec Active HDL

เป็นซอฟต์แวร์ที่ใช้ในการออกแบบ และดูการทำงานของวงจรดิจิทัล โดยใช้ภาษา VHSIC Hardware Description Language (VHDL) โดยแสดงดังรูปที่ 2.19

การใช้ Aldec Active HDL ในการออกแบบวงจรดิจิทัลหรือเขียนโค้ด VHDL มีลำดับขั้นตอนดังนี้ ขั้นตอนที่หนึ่ง requirement คือ ความต้องการในการออกแบบ ขั้นตอนที่สอง Design Entry คือ การเขียนโค้ด HDL ลงบนซอฟต์แวร์ และขั้นตอนสุดท้าย Functional Simulation คือ ดูการทำงานของ HDL โดยใช้การจำลองบนซอฟต์แวร์ (Simulation) ดังรูปที่ 2.20



รูปที่ 2.19 ซอฟต์แวร์ Aldec Active HDL

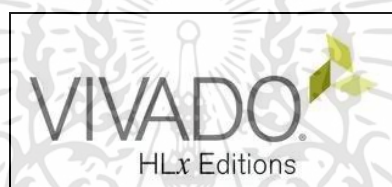


รูปที่ 2.20 ขั้นตอนการทำงานของ Aldec Active HDL

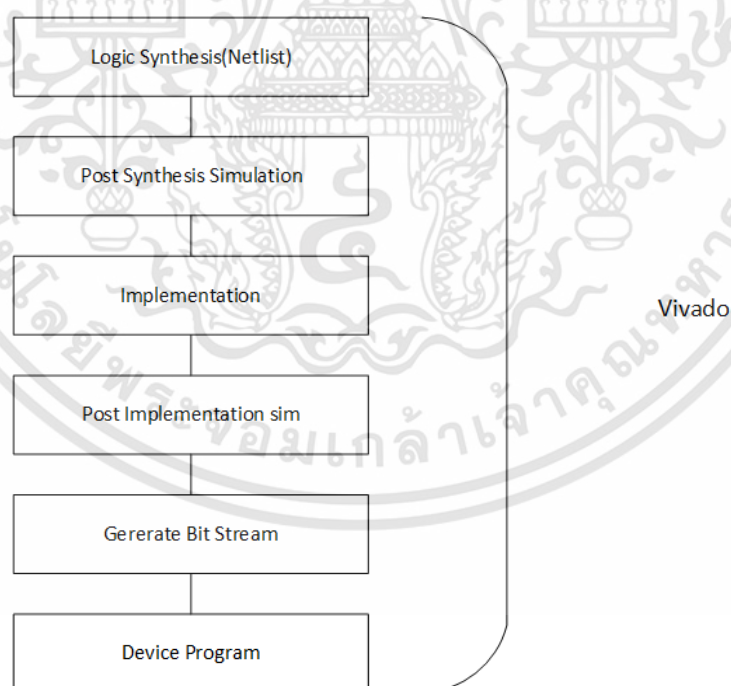
2.4.3 VIVADO

เป็นซอฟต์แวร์ที่ใช้ในการออกแบบหรือดูการทำงานของวงจรถิจริตอลที่ได้ออกแบบ และสามารถส่งข้อมูลที่ได้ออกแบบไว้ลงบอร์ด ที่ต้องการจะให้สร้างวงจรถิจริตอลที่ได้ออกแบบไว้โดยแสดงดังรูปที่ 2.21

การใช้ VIVADO ในการออกแบบวงจรมีลำดับขั้นตอนดังนี้ 1. Logic Synthesis (Netlist) 2. Post Synthesis Simulation 3. Implementation 4. Post Implementation sim 5. Generate Bit Stream คือ การสร้างไฟล์ .bit สำหรับลงฮาร์ดแวร์ และ 6. Device Program คือ การนำบิตสตรีมที่สร้างขึ้นลงบอร์ดเพื่อให้บอร์ดทำงาน แสดงดังรูปที่ 2.22



รูปที่ 2.21 ซอฟต์แวร์ Vivado



รูปที่ 2.22 ขั้นตอนการทำงานของ Aldec Active HDL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 ภาษา VHDL

VHDL (Very High Speed Integrated Circuit Hardware Description Language) เป็นภาษา HDL (Hardware Description Language) ซึ่งเป็นภาษาเฉพาะทางทางคอมพิวเตอร์ที่ใช้อธิบายโครงสร้าง และพฤติกรรมของวงจรรีเลย์ทรอนิกส์หรือวงจรรวมดิจิทัล (digital logic circuits) การเชื่อมต่อระหว่างกันของชิ้นส่วนอิเล็กทรอนิกส์เป็นรูปแบบเส้นทางต่างๆ ของวงจรรีเลย์ทรอนิกส์ ภาษาVHDL เป็นมาตรฐาน IEEE

VHDL ถูกเริ่มต้นพัฒนาในปี ค.ศ. 1981 กระทรวงกลาโหมของสหรัฐอเมริกาตั้งโครงการ VHSIC เพื่อสร้างโปรแกรมภาษาระดับสูงให้บรรยายพฤติกรรมของวงจรรวมดิจิทัล เพื่อให้ออกแบบวงจรรวมดิจิทัลได้อย่างรวดเร็ว ค.ศ. 1985 อนุญาตให้ถ่ายทอดเทคโนโลยีใช้ได้ทั่วไป ต่อมา IEEE ได้เข้ามาศึกษาจนถึงปี 1987 ได้เผยแพร่เป็นมาตรฐาน IEEE 1976-1987 ปัจจุบัน มาตรฐาน IEEE 1079-1993 หรือ VHDL 1993

2.5.1 ส่วนประกอบของ VHDL ประกอบด้วย 3 ส่วน

1) Library และ Package

ไลบรารี และแพคเกจเป็นฟังก์ชันต่าง ๆ ของตัวภาษา VHDL

2) Entity

Entity สำหรับบอกว่าอุปกรณ์ที่ออกแบบมี Input / Output อย่างไรบ้าง

3) Architecture

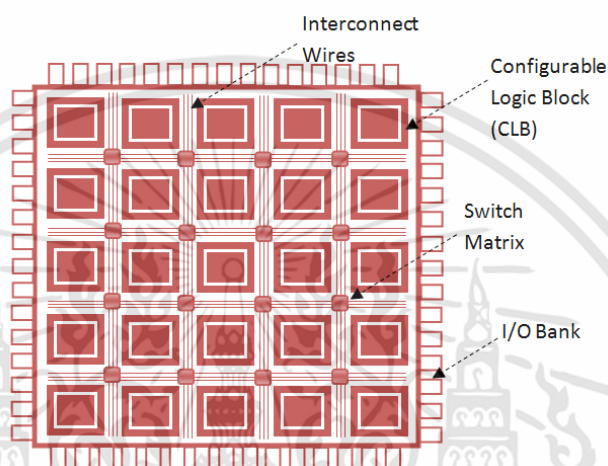
Architecture อธิบายพฤติกรรมของฮาร์ดแวร์ที่กำลังออกแบบ

2.6 FPGA

FPGA (Field Programable Gate Array) เป็นอุปกรณ์ลอจิกแบบโปรแกรมได้ โครงสร้างภายในของ FPGA สามารถโปรแกรมให้มีการทำงานเหมือนลอจิกเกตพื้นฐาน เช่น AND, OR, XOR, NOT หรือรวมกันหลายๆ ชนิด (Combinational logic) เพื่อให้วงจรทำหน้าที่ซับซ้อนยิ่งขึ้น เช่น decoder หรือฟังก์ชันทางคณิตศาสตร์ทั่วไป เป็นต้น และยังประกอบไปด้วยหน่วยความจำ ซึ่งอาจจะสร้างด้วยฟลิปฟล็อปอย่างง่าย (Sequential logic)

ในการออกแบบวงจรรีเลย์อิเล็กทรอนิกส์บนแผงวงจรรวมที่มีบอร์ด FPGA ในการออกแบบวงจรรีเลย์อิเล็กทรอนิกส์ ที่มี FPGA อยู่บนแผงวงจรรวมด้วยนั้น จะช่วยให้ผู้ออกแบบสามารถลดขนาดของแผงวงจรรวมทั้งสามารถออกแบบได้รวดเร็ว ไม่ต้องทดสอบรายละเอียด

ภายในให้เสร็จสมบูรณ์ก่อน ก็สามารถออกแบบแผงวงจรได้ เมื่อได้รับแผงวงจร และประกอบอุปกรณ์ต่างๆ เสร็จแล้ว จึงค่อยกำหนดหน้าที่การทำงานของ FPGA ได้ในภายหลัง โดย FPGA มีคุณสมบัติที่สำคัญ 3 อย่างด้วยกัน Flexible, Parallelism และ Reliability โดยโครงสร้างภายในของบอร์ด FPGA แสดงดังรูป 2.23



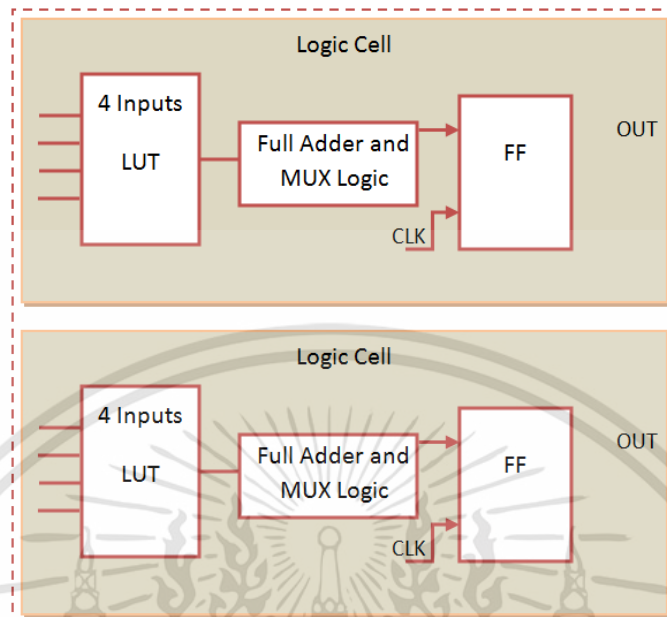
รูปที่ 2.23 FPGA Architecture

2.6.1 IOB (Input/Output Block)

เป็นส่วนประกอบที่เชื่อมต่อระหว่าง CLB และส่วนประกอบอื่นของชิพ FPGA

2.6.2 CLB (Configuration Logic Block)

เป็นทรัพยากรลอจิกแบบทำซ้ำพื้นฐานบน FPGA เมื่อเชื่อมโยงเข้าด้วยกันโดยการกำหนดเส้นทางทรัพยากร ส่วนประกอบใน CLB จะเรียกใช้ฟังก์ชันลอจิกที่ซับซ้อน ใช้ฟังก์ชันหน่วยความจำ และซิงก์โครไนซ์โค้ดบน FPGA ประกอบไปด้วย 1.Flip-Flop 2.Look-up Table(LUT) 3.Multiplexer ดังแสดงได้ดังรูปที่ 2.24



รูปที่ 2.24 Structure of Configuration Logic Block

2.6.3 Interconnected line

เป็นเส้นทางการเชื่อมต่อวงจรภายในบอร์ดระหว่าง CLB

2.6.4 Switch matrix

เมทริกซ์สวิตช์ถูกนำมาใช้เพื่อกำหนดเส้นทางสัญญาณจากช่องหนึ่งไปยังอีกช่องหนึ่งใน

ชิป FPGA

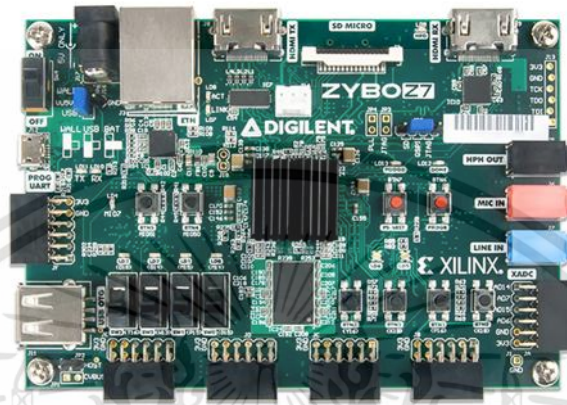
2.7 ฮาร์ดแวร์

2.7.1 ZYBO Z7-20 Development Board

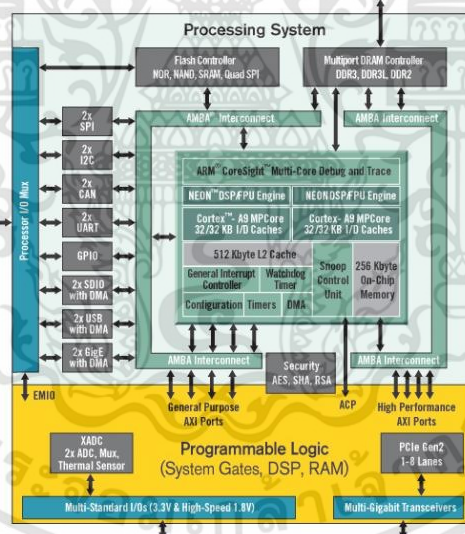
Zybo Z7 เป็นเครื่องมือที่ใช้ในระบบซอฟต์แวร์ฝังตัว (Embedded Software) และบอร์ดพัฒนาวงจรดิจิทัลที่สร้างขึ้นโดย Xilinx Zynq-7000 ซึ่งใช้สถาปัตยกรรม Xilinx All Programmable System-on-Chip (AP SoC) ที่ซึ่งรวมกันระหว่างโปรเซสเซอร์ ARM Cortex-A9 กับ Xilinx 7-series Field Programmable Gate Array (FPGA) โดยอุปกรณ์ล้อมรอบไปด้วยอุปกรณ์ต่อพ่วงมัลติมีเดียและการเชื่อมต่อที่หลากหลายเพื่อสร้างคอมพิวเตอร์บอร์ดที่แข็งแกร่ง รวมถึงมีพีเจอาร์ที่เป็น FPGA ดังแสดงดังรูป 2.25

Zynq APSoC ประกอบไปด้วย 2 ส่วนย่อยคือส่วนที่หนึ่ง Processing System (PS) เป็นชิพ ARM ที่ฝังอยู่ภายใน FPGA และส่วนที่สอง the Programmable Logic (PL) เป็นส่วนของ

logic ที่สามารถโปรแกรมได้บน FPGA ทำให้ชิพ FPGA สามารถรองรับการทำงาน High-end ที่ซับซ้อนได้ ภายในชิพเดียว โดยมีโครงสร้างดังรูป 2.26



รูป 2.25 ZYBO Z7-20 Development Board

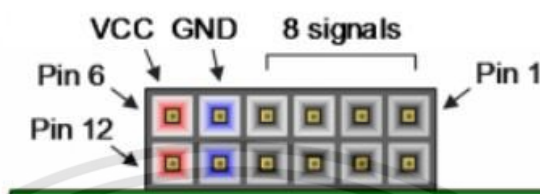


รูปที่ 2.26 Zynq APSoC (All Programmable System-on-Chip)

เนื่องจากบอร์ด Zybo Z7-20 เป็นบอร์ด FPGA สำหรับใช้ในการศึกษา และพัฒนางาน มีฟังก์ชันต่างๆ มากมาย ให้เลือกใช้ เช่น ปุ่ม, สวิตช์, ไฟ LED และพอร์ต I/O เป็นต้น ซึ่งเป็นส่วนประกอบที่ได้เลือกใช้ อย่างเช่นการตั้งค่านีเซตกับปุ่ม การแมปค่าเอาต์พุตกับพอร์ต I/O เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยในตารางที่ 2.3 นี้เป็นตารางแสดงชนิดของพอร์ต I/O โดยแต่ละพอร์ตจะประกอบไปด้วย 12 Pins ดังรูปที่ 2.27



รูป 2.27 ลำดับของ Pin ในหนึ่ง Pmod

ตารางที่ 2.3 ชื่อ และชนิดของพอร์ตต่างๆ

	Pmod JA	Pmod JB	Pmod JC	Pmod JD	Pmod JE	Pmod JF
Pmod Type	XADC	High-Speed	High-Speed	High-Speed	Standard	MIO
Pin 1	N15	V8	V15	T14	V12	MIO-13
Pin 2	L14	W8	W15	T15	W16	MIO-10
Pin 3	K16	U7	T11	P14	J15	MIO-11
Pin 4	K14	V7	T10	R14	H15	MIO-12
Pin 7	N16	Y7	W14	U14	V13	MIO-0
Pin 8	L15	Y6	Y14	U15	U17	MIO-9
Pin 9	J16	V6	T12	V17	T17	MIO-14
Pin 10	J14	W6	U12	V18	Y17	MIO-15

2.7.2 R&S RTM 2024 Oscilloscope

Oscilloscope เป็นเครื่องมือวัดคลื่นกระแสไฟฟ้า มีฟังก์ชันที่เป็นประโยชน์ในการวัดสัญญาณตัวอย่างเช่น Trigger หมายถึงการจับส่วนที่น่าสนใจของรูปคลื่นที่เกี่ยวข้อง การเลือกประเภททริกเกอร์ที่ถูกต้อง และการกำหนดค่าการตั้งค่าทริกเกอร์ทั้งหมดอย่างถูกต้องช่วยให้ตรวจค่าจับต่างๆ เหตุการณ์ในสัญญาณแอนะล็อกมดิจิตอล และสัญญาณลอจิกได้อย่างถูกต้อง Protocol

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Analysis หมายถึงการวิเคราะห์สัญญาณโดยใช้โปรโตคอล เช่น Parallel, SP, I²C เป็นต้น โดย
เครื่องมือแสดงดังรูป 2.28



รูป 2.28 R&S RTM 2024 Oscilloscope

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

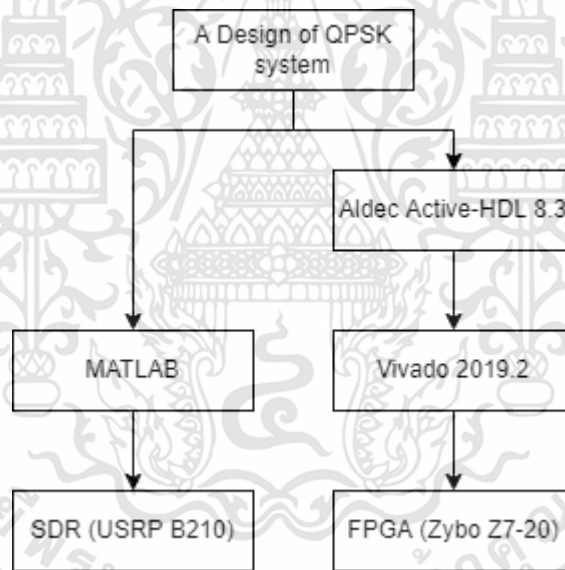
บทที่ 3

การออกแบบและการจัดทำปริญญาานิพนธ์

ปริญญาานิพนธ์เรื่องการออกแบบระบบสื่อสาร QPSK ด้วย Software-Defined Radio (SDR) มีอุปกรณ์ และขั้นตอนดำเนินงานดังนี้

3.1 การออกแบบ

ปริญญาานิพนธ์ เรื่องการออกแบบระบบสื่อสาร QPSK ด้วย Software-Defined Radio (SDR) นี้ จัดทำเพื่อศึกษากระบวนการในการออกแบบระบบสื่อสารด้วย SDR โดยมีขั้นตอนการออกแบบการทดสอบ 2 ส่วนคือ ส่วนที่จำลองด้วยซอฟต์แวร์ และส่วนที่ทดสอบด้วยฮาร์ดแวร์ ซึ่งมีแผนผังการทำงานดังรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมแผนการทำงานทั้งหมด

ตารางที่ 3.1 แสดงค่าพารามิเตอร์ต่างๆ ที่ใช้สำหรับการออกแบบระบบสื่อสาร QPSK

คุณสมบัติทั่วไปของระบบสื่อสาร (General properties of communication system)	
ความถี่ที่ใช้ทดลอง (frequency)	140 MHz (with coaxial cable), 4 GHz (with antenna)
อัตราเร็วข้อมูล (data rate)	2 Mbps

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.1 (ต่อ) แสดงค่าพารามิเตอร์ต่างๆ ที่ใช้สำหรับการออกแบบระบบสื่อสาร QPSK

คุณสมบัติด้านเทคนิคของระบบสื่อสาร (Technical properties of communication system)	
เทคนิคการมอดูเลชัน (modulation technique)	QPSK (with Gray's order)
สแครมเบลอร์ (Scrambler)	With polynomial = $x^7 + x^4 + 1$, Initial state = [0 0 0 0 0 0 0]
ฟิลเตอร์ (Filter)	RRC filter, Roll off factor = 0.35, 41 tabs

จากตารางที่ 3.1 เป็นตารางแสดงคุณสมบัติต่างๆ เพื่อใช้ในการออกแบบระบบการสื่อสาร QPSK นี้ โดยแบ่งเป็นค่าคุณสมบัติทั่วไปในการส่งสัญญาณของระบบสื่อสาร และการออกแบบเชิงเทคนิคของระบบสื่อสาร

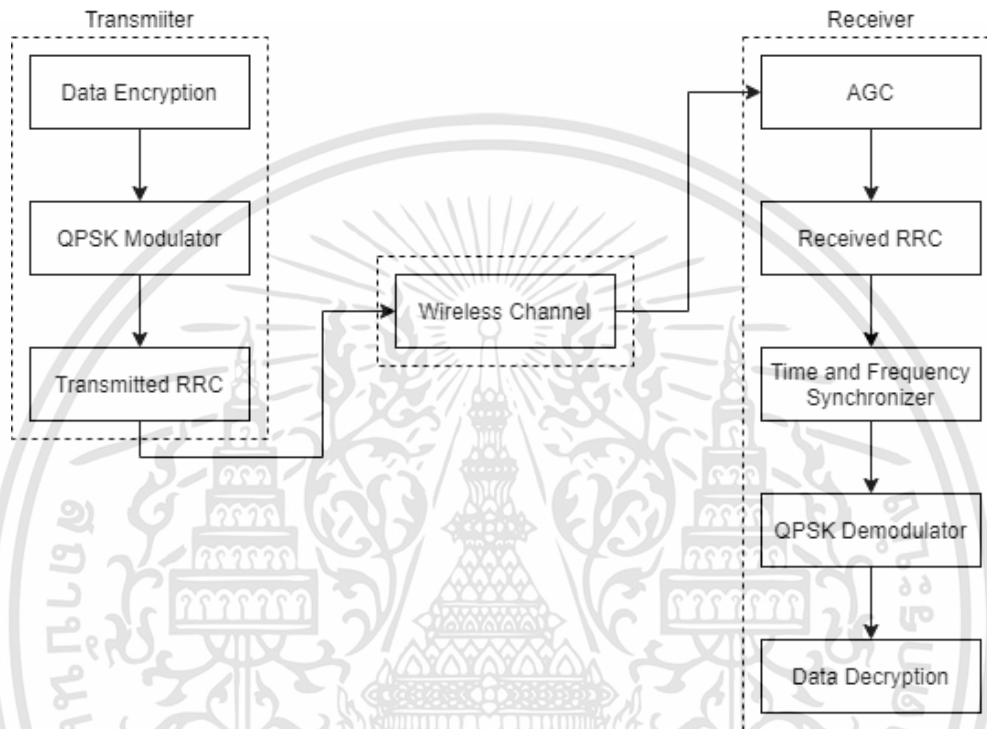
3.1.1 การออกแบบระบบสื่อสาร QPSK ด้วยซอฟต์แวร์ MATLAB

ในขั้นตอนนี้ได้ทำการออกแบบระบบสื่อสาร QPSK ด้วยซอฟต์แวร์ MATLAB โดยมีบล็อกไดอะแกรมดังรูปที่ 3.2 ซึ่งแบ่งออกเป็น 3 ส่วนหลักๆ ได้แก่ การออกแบบระบบสื่อสาร QPSK (Transmitter), การออกแบบระบบสื่อสาร QPSK ภาครับ (Receiver) และการออกแบบช่องสัญญาณการสื่อสาร (Channel)

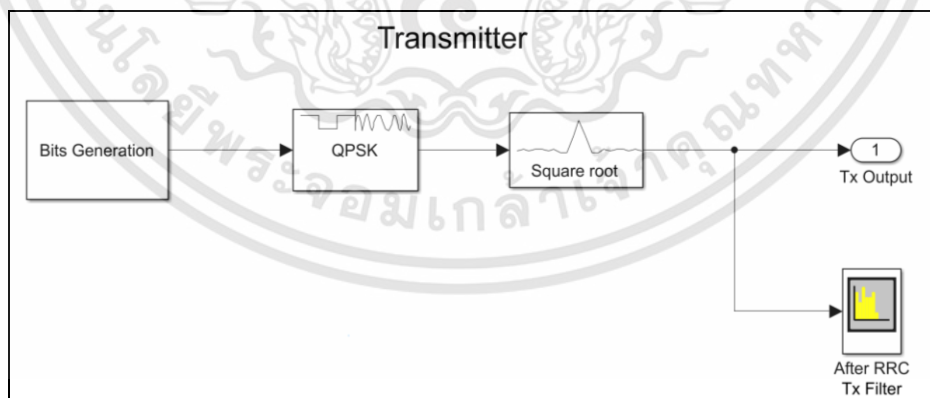
3.1.1.1 การออกแบบระบบสื่อสาร QPSK ภาครับ

โดยที่ฝั่งส่ง ข้อความอินพุตของระบบถูกสร้างขึ้นด้วยการเขียนสคริปต์แมทแลป ส่งผ่านมายังบล็อกภายใน Simulink ข้อความที่ถูกส่งมาถูกแปลงเป็นข้อมูลไบนารี และเข้าสู่กระบวนการ Encryption โดยจะทำการสเคลมบลิงข้อมูลอินพุต และทำการเพิ่มเฮดเดออร์ ไปยังส่วนต้นของข้อมูลเพื่อทำการส่งข้อมูลออกเป็นเฟรม

หลังจากผ่านการทำ Encryption แล้ว สัญญาณดิจิทัลจะถูกมอดูเลตแบบ QPSK ก่อนผ่านกระบวนการ Filter สัญญาณ และส่งไปยังส่วนถัดไป ดังรูปที่ 3.3



รูปที่ 3.2 บล็อกไดอะแกรมของระบบสื่อสาร QPSK



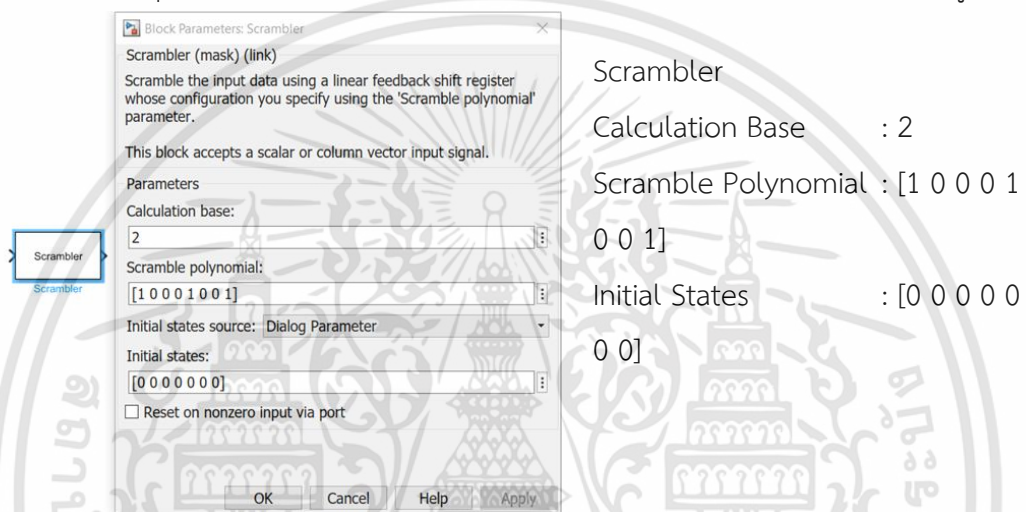
รูปที่ 3.3 ระบบสื่อสาร QPSK ภาดส่ง ออกแบบด้วยซอฟต์แวร์ Simulink

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบระบบสื่อสาร QPSK ภาคส่ง บนซอฟต์แวร์ Simulink ได้มีการตั้งค่าพารามิเตอร์ต่างๆ ไว้ดังต่อไปนี้

1) Scrambler

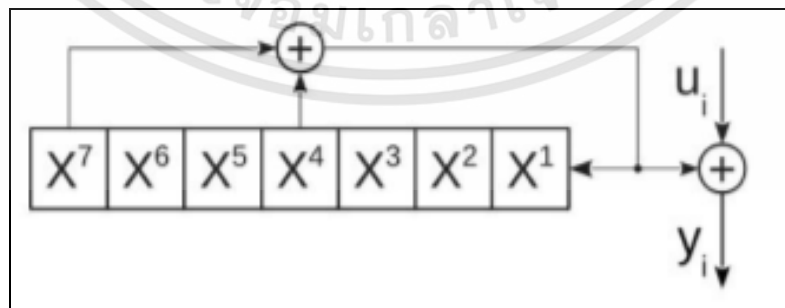
บล็อกภายในซอฟต์แวร์ Simulink ใช้สำหรับสลับข้อมูลเพื่อลดการรวมกลุ่มของบิต 0 และบิต 1 ของสัญญาณส่ง มีบล็อก และตั้งค่าพารามิเตอร์ดังรูปที่ 3.4



รูปที่ 3.4 บล็อก Scrambler ใน Simulink

จากการตั้งค่าในรูปที่ 3.4 ค่าโพลีโนเมียลของบล็อก Scrambler มีค่าเท่ากับ [1 0 0 0 1 0 0 1] ซึ่งมีลำดับดังสมการที่ 3.1 การทำงานของบล็อก Scrambler ด้วยค่าโพลีโนเมียลดังกล่าว จะมีหลักการทำงานดังรูปที่ 3.5

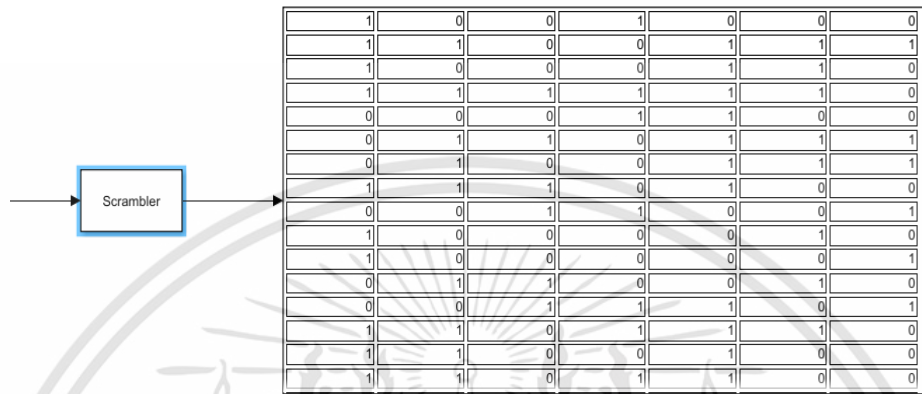
$$f(x) = x^7 + x^4 + 1 \quad (3.1)$$



รูปที่ 3.5 การทำงานของสแครมเบลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลอินพุตที่ผ่านบล็อกสแครมเบลอร์ จะได้ข้อมูลที่มีความน่าจะเป็นของการเฉลี่ยบิต 0 และบิต 1 ดังรูปที่ 3.6

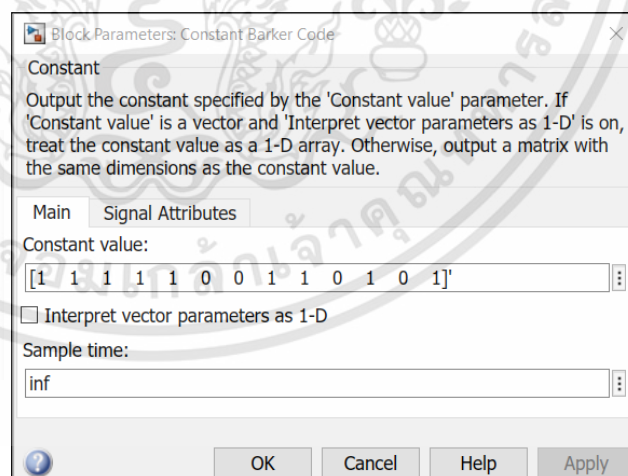


รูปที่ 3.6 ข้อมูลที่ส่งข้อความ “Hello world 026\n” ผ่านสแครมเบลอร์

2) Barker Code

ในการออกแบบเฮตเตอร์ของเฟรมข้อมูล ได้ใช้บล็อก Constant ในการสร้างค่าไบนารี ซึ่งเป็นค่าที่เรียงลำดับตามรหัสของบาร์คเกอร์ ความยาวเท่ากับ 13 บิต และแปลงเป็นข้อมูลแบบไม่มีขั้ว โดยมีการตั้งค่าพารามิเตอร์ดังรูปที่ 3.7

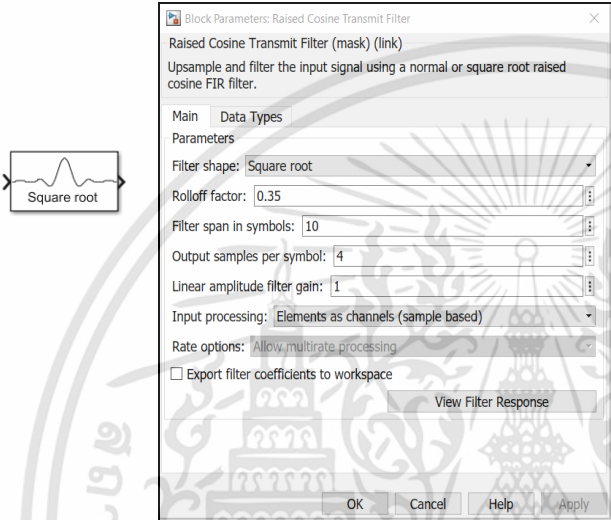
[1x13] 
Constant
Barker Code



รูปที่ 3.7 สร้างเฮตเตอร์ของเฟรมข้อมูลด้วยลำดับของรหัสบาร์คเกอร์

4) Raised Cosine Transmit Filter

บล็อก Raised Cosine Transmit Filter ทำหน้าที่อัปแซมเปิลและฟิลเตอร์สัญญาณอินพุต สามารถปรับใช้ได้ทั้ง RC ฟิลเตอร์ และ RRC ฟิลเตอร์ ดังรูปที่ 3.11 และได้ผลตอบสนองของฟิลเตอร์ดังรูปที่ 3.12



Block Parameters: Raised Cosine Transmit Filter

Raised Cosine Transmit Filter (mask) (link)
Upsample and filter the input signal using a normal or square root raised cosine FIR filter.

Main Data Types

Parameters

Filter shape: Square root

Rolloff factor: 0.35

Filter span in symbols: 10

Output samples per symbol: 4

Linear amplitude filter gain: 1

Input processing: Elements as channels (sample based)

Rate options: Allow multirate processing

Export filter coefficients to workspace

View Filter Response

OK Cancel Help Apply

Raised Root Cosine Tx Filter (RRC Tx)

Filter Shape : Square Root

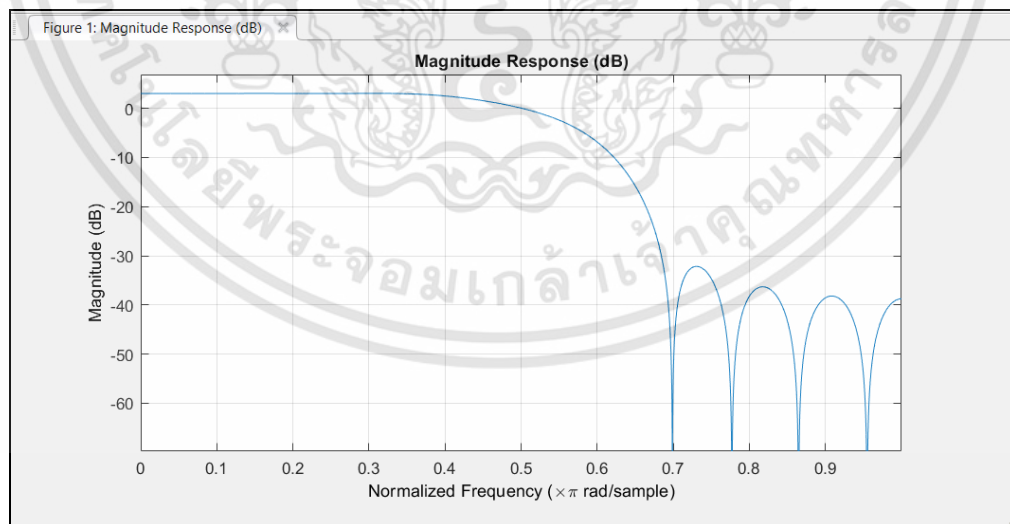
Roll Off Factor : 0.35

Filter Span in Symbols : 10

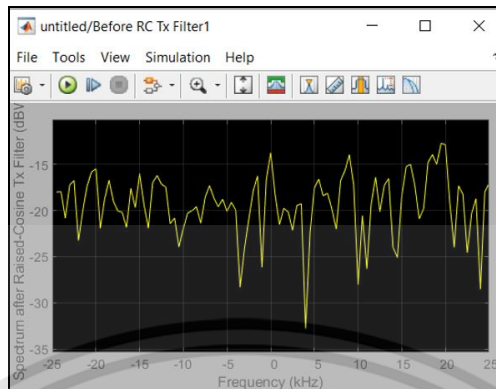
Output Samples per Symbol : 4

Linear Amplitude Filter Gain : 1

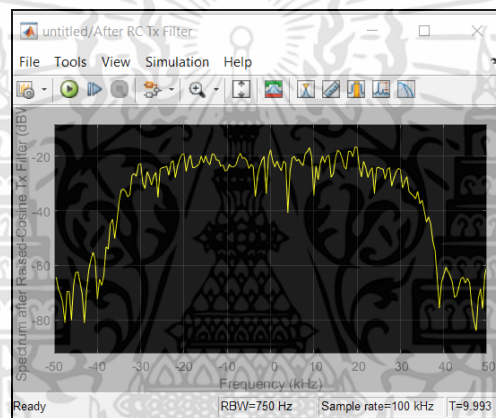
รูปที่ 3.11 บล็อก Raised Cosine Transmit Filter ใน Simulink



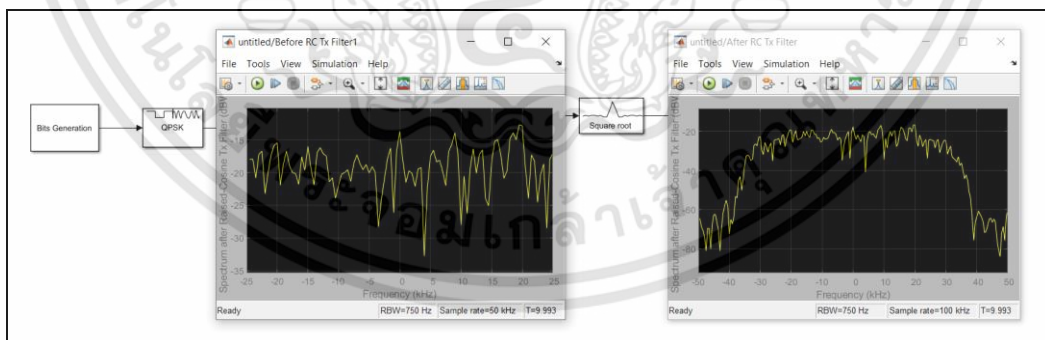
รูปที่ 3.12 ผลตอบสนองของตัวกรอง RRC filter



(ก)



(ข)



(ค)

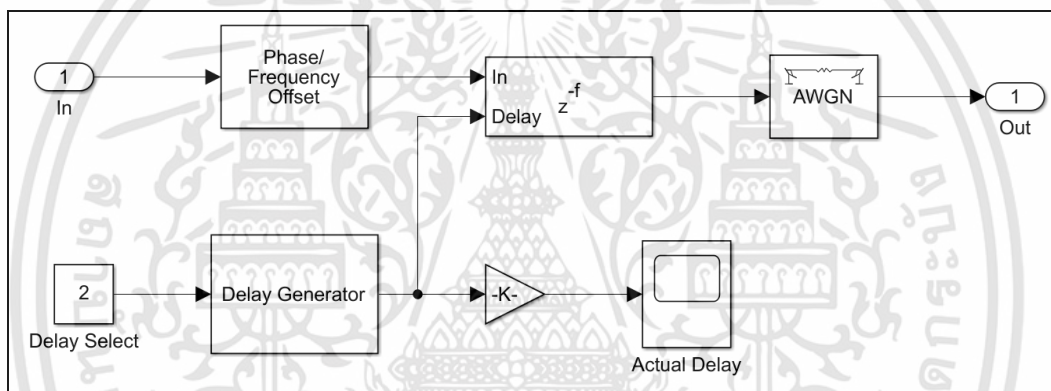
รูปที่ 3.13 ภาพรวมของการส่งผ่านข้อมูลผ่านมอดูเลเตอร์ และฟิลเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.13 (ก) เมื่อใส่ชุดข้อมูลที่ได้จาก QPSK Modulation โดยแสดงสเปกตรัมได้ดังรูปที่ 3.13 (ก) ผ่านตัวกรองจะได้ข้อมูลดังรูปที่ 3.13 (ข) จะเห็นเป็นข้อมูลที่ตอบสนองต่อผลตอบสนองตัวกรองของฟิลเตอร์โดยแสดงภาพรวมของการทำงานได้ดังรูปที่ 3.13 (ค)

3.1.1.2 การออกแบบช่องสัญญาณไร้สาย

ข้อมูลที่ถูกส่งจากภาคส่งจะผ่านบล็อกที่ออกแบบสำหรับการจำลองช่องสัญญาณไร้สายดังรูปที่ 3.14 เป็นการจำลองผลกระทบต่อสัญญาณส่ง ซึ่งเกิดจากสิ่งแวดล้อมภายในตัวกลาง เช่น สัญญาณรบกวน, Time Delay และ Frequency Offset

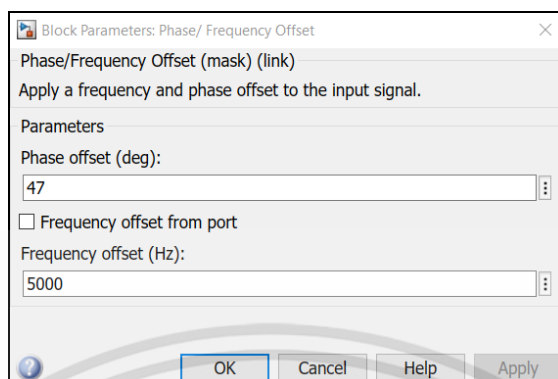


รูปที่ 3.14 ออกแบบช่องสัญญาณไร้สายด้วยซอฟต์แวร์ Simulink

การออกแบบผลกระทบจากการสื่อสารผ่านช่องสัญญาณบนซอฟต์แวร์ Simulink ได้มีการตั้งค่าพารามิเตอร์ไว้ดังต่อไปนี้

1) Phase and Frequency Offset

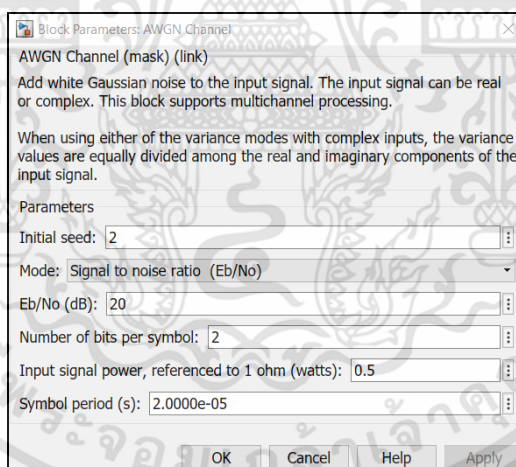
บล็อก Phase and Frequency Offset ทำหน้าที่เลื่อนเฟส และความถี่ของสัญญาณอินพุต ตามที่ตั้งค่าไว้ ดังรูปที่ 3.15



รูปที่ 3.15 บล็อก Phase and Frequency Offset ใน Simulink

2) AWGN Channel

บล็อก AWGN Channel ใช้สำหรับจำลองการสื่อสารผ่านช่องสัญญาณ โดยสร้างสัญญาณรบกวนเกาส์สีขาวแบบบวกเข้าไปในระบบการสื่อสาร โดยมีการตั้งค่าพารามิเตอร์ ดังรูปที่ 3.16

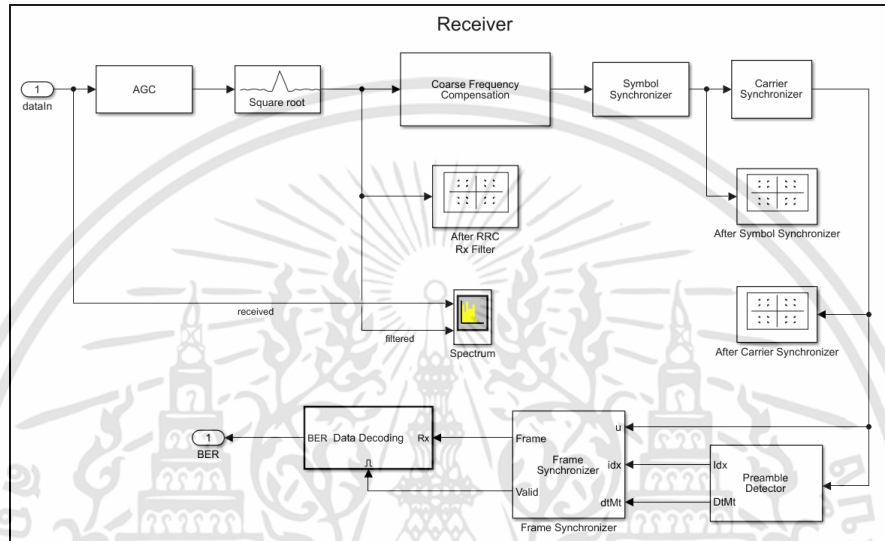


รูปที่ 3.16 บล็อก AWGN Channel ใน Simulink

3.1.1.3 การออกแบบระบบสื่อสาร QPSK ภาครับ

ในส่วนของการออกแบบภาครับรูปที่ 3.17 สัญญาณที่ได้จะผ่าน AGC เพื่อปรับเกนของสัญญาณอัตโนมัติ ตามค่า Threshold ที่ตั้งไว้ และจะทำการกรอง Filter ภาครับ ส่ง

ต่อไปยังส่วนที่ใช้สำหรับซิงโครไนซ์สัญญาณ ทั้งในส่วนของเวลา ความถี่ และการชดเชยเฟส ก่อนผ่านกระบวนการตีมอดูเลตสัญญาณ QPSK ข้อมูลที่ได้จากการตีมอดูเลตแล้วจะถูกทำกระบวนการ Description ข้อมูล ซึ่งเป็นการถอดส่วนหัวของเฟรมข้อมูลออก และทำการ Descrambler เพื่อให้ได้ข้อมูลดั้งเดิม



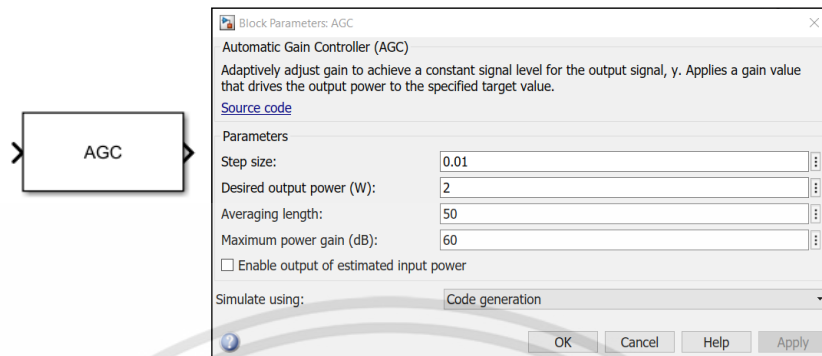
รูปที่ 3.17 ระบบสื่อสาร QPSK ภาครับ ออกแบบด้วยซอฟต์แวร์ Simulink

โดยการออกแบบระบบสื่อสาร QPSK ภาครับบนซอฟต์แวร์ Simulink มีการตั้งค่าพารามิเตอร์ไว้ดังต่อไปนี้

1) Automatic Gain Control (AGC)

บล็อก AGC ใน Simulink ใช้สำหรับปรับเกนของสัญญาณอินพุต เพื่อให้เกนของสัญญาณเอาต์พุตตามค่าที่ต้องการ มีพารามิเตอร์ตามรูปที่ 3.18

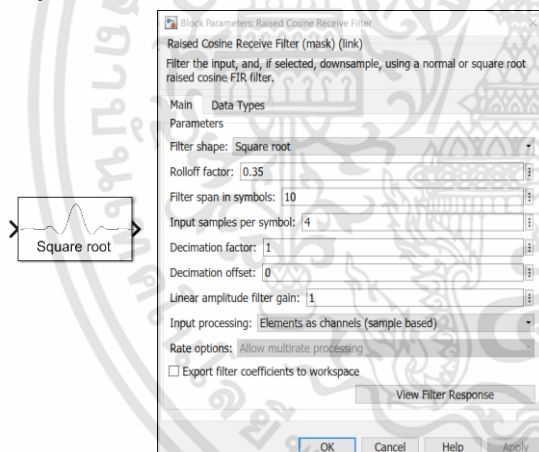
- Step Size : 0.01
- Desired Output Power (W) : 2
- Averaging Length : 50
- Maximum Power Gain (dB) : 60



รูปที่ 3.18 บล็อก AGC ใน Simulink

2) Raised Cosine Receive Filter

บล็อก Raised Cosine Receive Filter ใน Simulink ใช้สำหรับ คำนวณแอมพลิจูดของสัญญาณที่ได้รับในภาครับ และกรองสัญญาณที่ถูกฟิลเตอร์ ซึ่งได้ตั้งค่าพารามิเตอร์ไว้ ดังรูปที่ 3.19

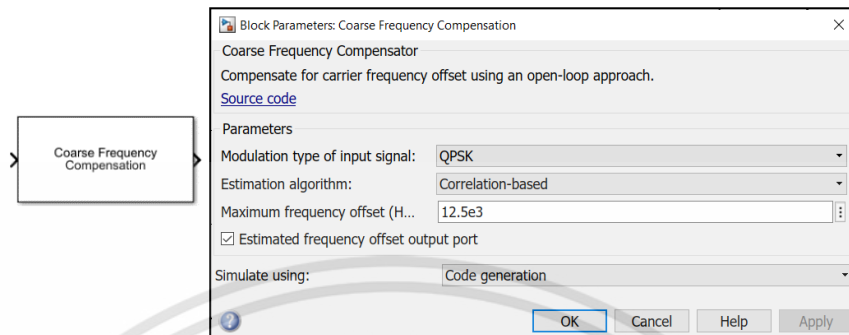


- Raised Root Cosine Rx Filter (RRC Rx)
- Filter Shape : Square Root
- Roll Off Factor : 0.35
- Filter Span in Symbols : 10
- Input Samples per Symbol : 4
- Decimation Factor : 1
- Decimation Offset : 0
- Linear Amplitude Filter Gain : 1

รูปที่ 3.19 บล็อก Raised Cosine Receive Filter ใน Simulink

3) Coarse Frequency Compensation

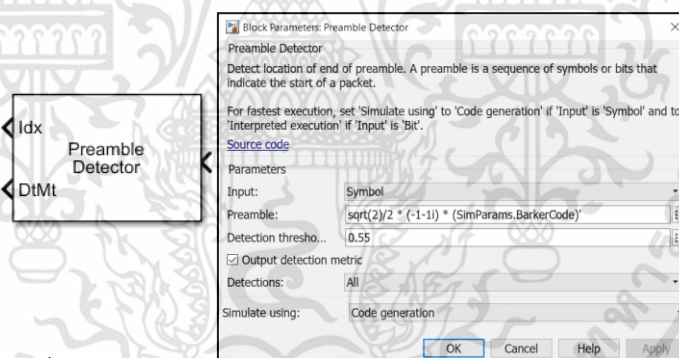
บล็อก Coarse Frequency Compensation มีหน้าที่ชดเชย ความถี่ออฟเซตของสัญญาณแควร์รีที่ภาครับ สำหรับการมอดูเลตแบบ BPSK, QPSK, OQPSK, 8-FSK, PAM และ QAM ดังรูปที่ 3.20



รูปที่ 3.20 บล็อก Coarse Frequency Compensation ใน Simulink

4) Preamble Detector

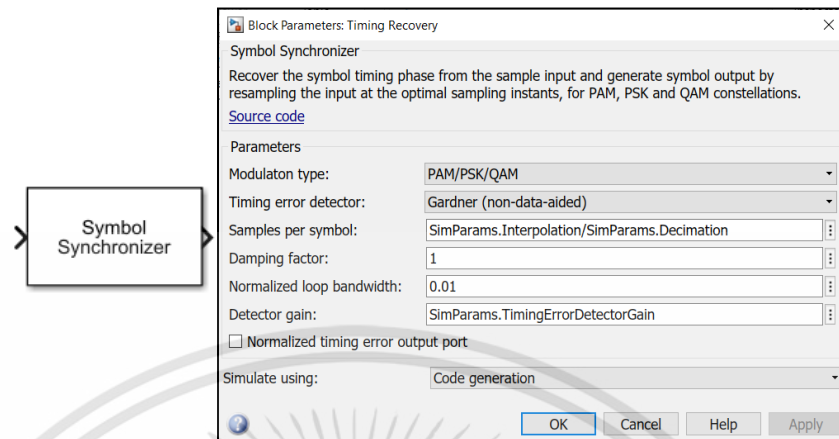
บล็อก Preamble Detector ใช้สำหรับทำหน้าที่ในการตรวจหาพรีเอมเบิลของเฟรมข้อมูล เพื่อใช้สำหรับระบุส่วนต้นของเฟรมข้อมูลในระบบสื่อสารภาครับ ดังรูปที่ 3.21



รูปที่ 3.21 บล็อก Preamble Detector ใน Simulink

5) Symbol Synchronizer

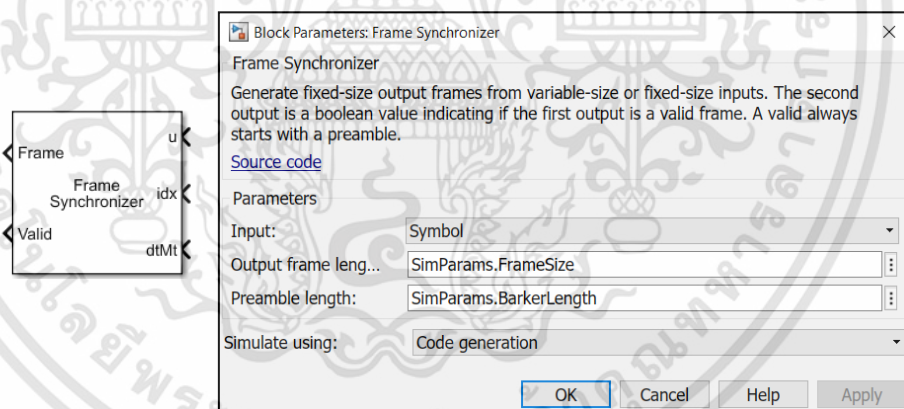
บล็อก Symbol Synchronizer ทำหน้าที่แก้ไขความผิดพลาดที่เกิดจากการคลาดเคลื่อนของเวลา ของสัญญาณที่มอดูเลตแบบ PAM, PSK, QAM, และ OQPSK ในระบบสื่อสารภาครับ ระบบการซิงโครไนซ์ Symbol ใช้พื้นฐานของ PPL อัลกอริทึม ดังรูปที่ 3.22



รูปที่ 3.22 บล็อก Symbol Synchronizer ใน Simulink

6) Frame Synchronizer

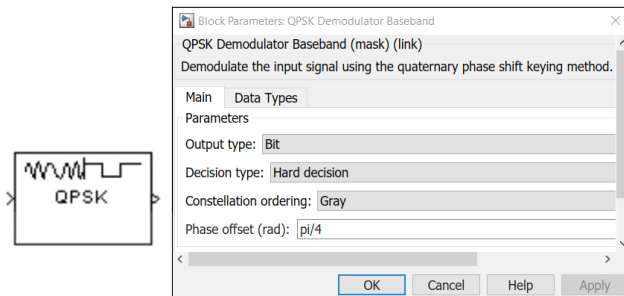
รูปที่ 3.23 บล็อก Frame Synchronizer ใน Simulink ทำหน้าที่ระบุความยาวของเฟรม และความยาวของพรีแอมเบิล เฟรมเอาต์พุตจะถูกนำหน้าด้วยพรีแอมเบิลเสมอ



รูปที่ 3.23 บล็อก Frame Synchronizer ใน Simulink

7) QPSK Demodulation

เป็นการตีמודูเลชันจากการมอดูเลชันแบบ QPSK โดยใช้การ Correlator และการตัดสินใจในการดึงข้อมูลกลับมาโดยจะได้ข้อมูลออกมาเป็นบิตคู่ และบิตคู่เมื่อนำมารวมกันก็จะได้ข้อมูลชุดเดิมก่อนเข้า QPSK โดยมีการตั้งค่าพารามิเตอร์ดังรูปที่ 3.24



QPSK Demodulator

Output Type : Bit

Decision Type : Hard Decision

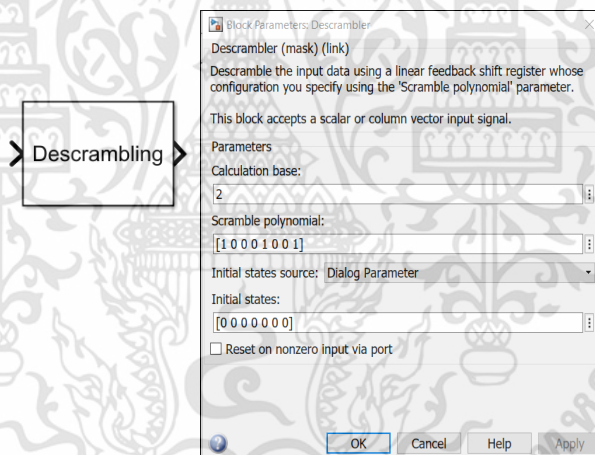
Constellation Ordering: Gray

Phase Offset : $\pi/4$

รูปที่ 3.24 บล็อก QPSK Demodulator ใน Simulink

8) Descrambler

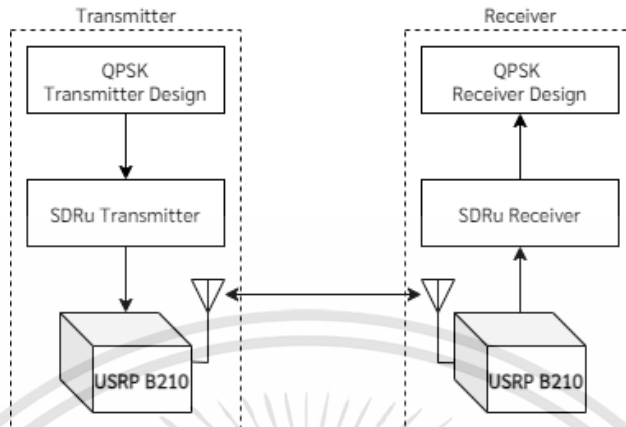
บล็อก Descrambler ใช้สำหรับตีสแครมเบิลข้อมูลในระบบสื่อสารภาครับเพื่อให้ได้ข้อมูลดั้งเดิม ดังรูปที่ 3.25



รูปที่ 3.25 บล็อก Descrambler ใน Simulink

3.1.2 การออกแบบการทดสอบด้วย USRP B210

ขั้นตอนนี้ใช้แบบการสื่อสาร QPSK ที่ได้ทำการออกแบบไว้ในส่วนการวัดผล Simulation ซึ่งการสื่อสารจากฝั่งส่งไปยังฝั่งรับ สัญญาณจะถูกส่งจาก USRP B210 ฝั่งส่งผ่านสายอากาศ หรือสายโคแอกซ์เซียม ไปยัง USRP B210 ฝั่งรับดังรูปที่ 3.26



รูปที่ 3.26 แผนภาพการทดสอบด้วย USRP B210

ระบบสื่อสาร QPSK ที่ได้ออกแบบไว้บนซอฟต์แวร์ MATLAB และ Simulink เชื่อมต่อกับ USRP B210 ผ่านบล็อกที่ชื่อว่า “SDRu Transmitter” และ “SDRu Receiver” ดังรูปที่ 3.27 และรูปที่ 3.28

SDRu Transmitter

Center Frequency (Hz): 140 MHz

Gain (dB) : 60 dB

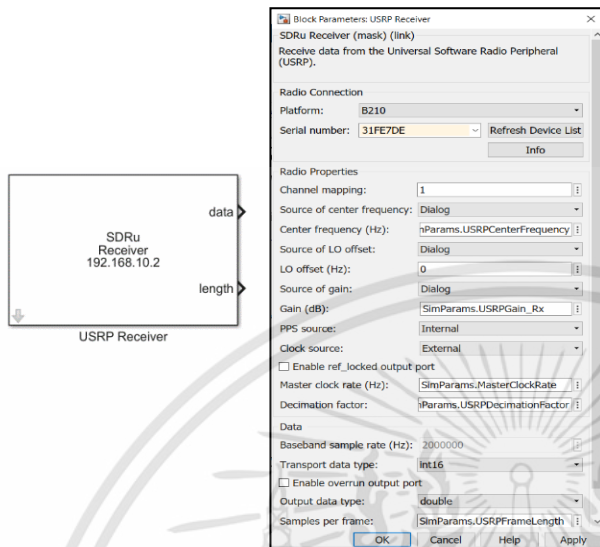
Clock Source : Internal

Master Clock Rate (Hz): 20 MHz

Interpolation Factor : 10

รูปที่ 3.27 บล็อก SDRu Transmitter ใน Simulink

จากรูปที่ 3.27 แสดงการตั้งค่าภายในบล็อก SDRu Transmitter โดยจะเลือกแพลตฟอร์มของ SDR เป็น B210 ส่งที่ความถี่ RF เท่ากับ 140 MHz เกนฝั่งส่งเท่ากับ 60 dB และใช้สัญญาณนาฬิกาภายในที่ความถี่ 20 MHz



SDRu Receiver

Center Frequency (Hz): 140 MHz

Gain (dB) : 40 dB

Clock Source : Internal

Master Clock Rate (Hz): 20 MHz

Decimation Factor : 10

รูปที่ 3.28 บล็อก SDRu Receiver ใน Simulink

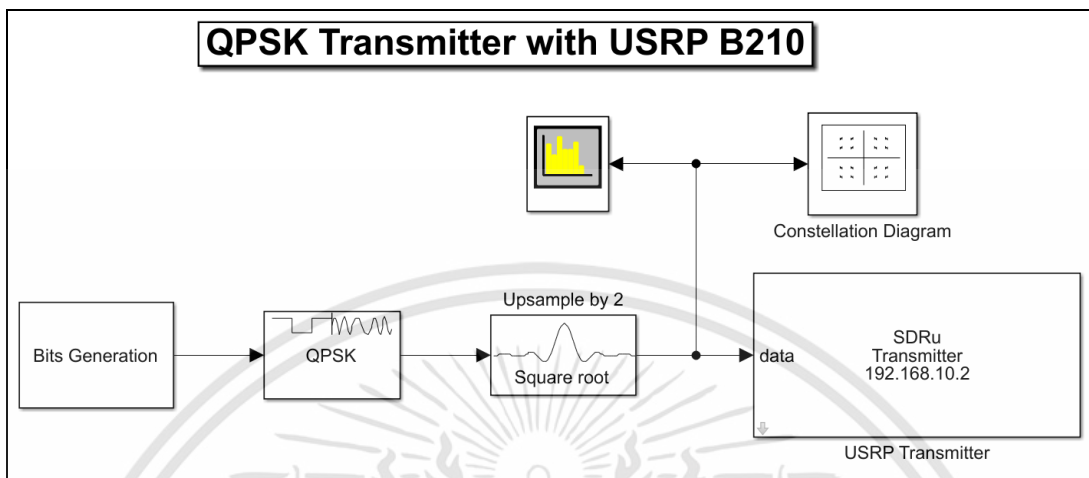
จากรูปที่ 3.28 แสดงการตั้งค่าภายในบล็อก SDRu Receiver โดยแพลตฟอร์มของ SDR คือ B210 รับที่ความถี่ RF เท่ากับ 140 MHz เกนฝั่งรับเท่ากับ 40 dB และใช้สัญญาณนาฬิกา อยู่ที่ความถี่ 20 MHz

3.1.2.1 การออกแบบระบบสื่อสาร QPSK ภาคส่ง กับ USRP B210

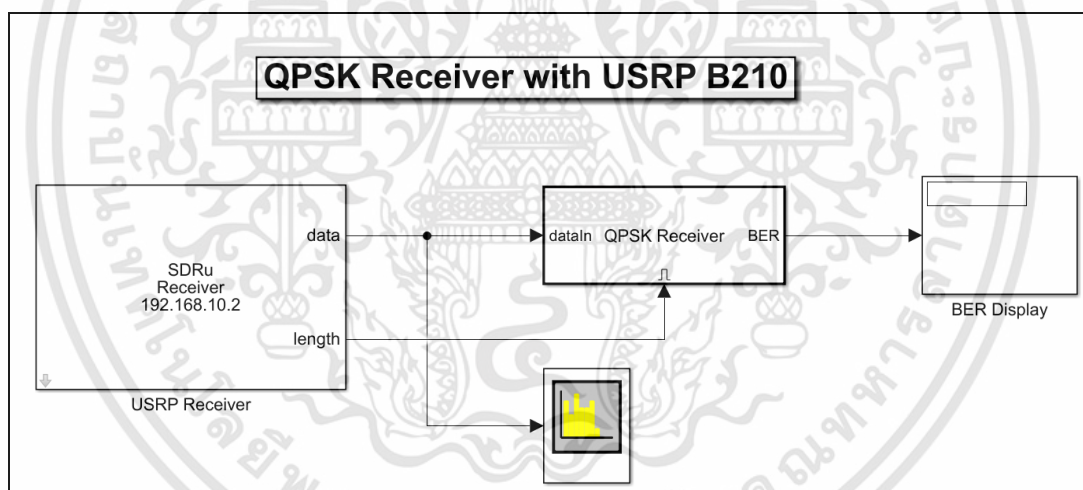
ระบบสื่อสาร QPSK ภาคส่งในรูปที่ 3.29 ข้อมูลอินพุตจะถูกสร้างขึ้นมาด้วยซอฟต์แวร์ MATLAB โดยจะผ่านกระบวนการจัดการสัญญาณทางดิจิทัล ก่อนเชื่อมต่อกับบล็อก SDRu Transmitter ซึ่งจะเป็นการเชื่อมต่อแลกเปลี่ยนภาคส่งกับฮาร์ดแวร์ USRP B210 ภาคส่ง

3.1.2.2 การออกแบบระบบสื่อสาร QPSK ภาครับ กับ USRP B210

ระบบสื่อสาร QPSK ภาครับในรูปที่ 3.30 USRP B210 ภาครับ จะรับสัญญาณส่งไปประมวลผลบนซอฟต์แวร์ MATLAB และ Simulink ของแลกเปลี่ยนเครื่องรับโดยเชื่อมต่อกับฮาร์ดแวร์ USRP B210 ภาครับกับแลกเปลี่ยนภาครับด้วยบล็อก SDRu Receiver



รูปที่ 3.29 QPSK Transmitter with USRP B210



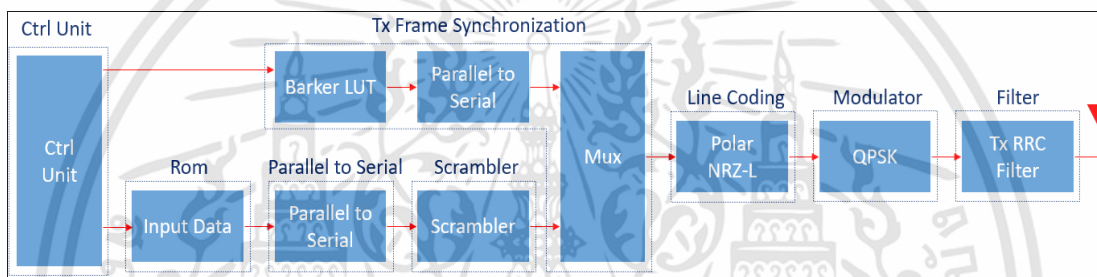
รูปที่ 3.30 QPSK Receiver with USRP B210

3.1.3 การออกแบบการทดสอบด้วย FPGA

ในขั้นตอนนี้ได้ทำการออกแบบระบบสื่อสาร QPSK จากวงจรดิจิทัล ด้วยภาษา VHDL บนซอฟต์แวร์ Aldec Active-HDL 8.3 โดยออกแบบระบบสื่อสาร QPSK ตามแบบที่ได้ทำการออกแบบไว้ด้วยซอฟต์แวร์ MATLAB เพื่อนำโค้ดที่ได้จากการออกแบบไปโปรแกรมลงบอร์ด

FPGA ซึ่งบอร์ด FPGA ที่ใช้ในการทดลองนี้คือบอร์ด Zybo Z7-20 โดยมีบล็อกไดอะแกรมการทำงานดังรูปที่ 1.4

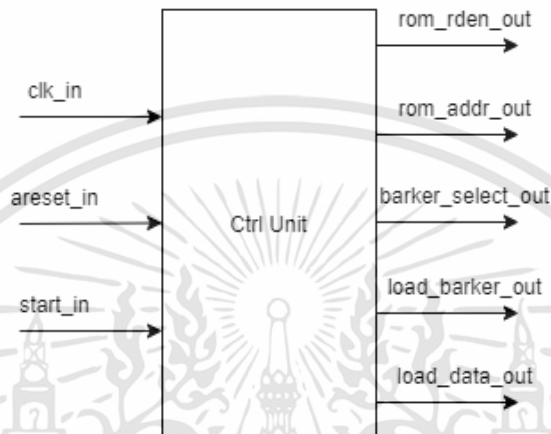
โดยในรูปที่ 3.31 คือบล็อกไดอะแกรมที่ใช้ในการออกแบบระบบสื่อสาร QPSK ภาคส่งด้วยภาษา VHDL ซึ่งประกอบไปด้วยโมดูลภายในระบบสื่อสาร ได้แก่ Scrambler, Tx Frame Synchronization, Polar NRZ-L, QPSK Modulator และ Tx RRC Filter และโมดูลอื่นๆ ได้แก่ Control Unit, Rom, Parallel to Serial, D-Flipflop, Multiplexer และ Clock Divider ซึ่งมีการออกแบบสัญญาณอินพุต และสัญญาณเอาต์พุตไว้ดังต่อไปนี้



รูปที่ 3.31 บล็อกไดอะแกรมระบบสื่อสาร QPSK ภาคส่ง สำหรับการทดสอบด้วย FPGA

3.3.3.1 Control Unit

Control Unit – เพื่อควบคุมสัญญาณ และการทำงานของโมดูลต่างๆ ภายในระบบมีสัญญาณอินพุต เอาต์พุตดังรูปที่ 3.33



รูปที่ 3.33 บล็อก Control Unit

Input

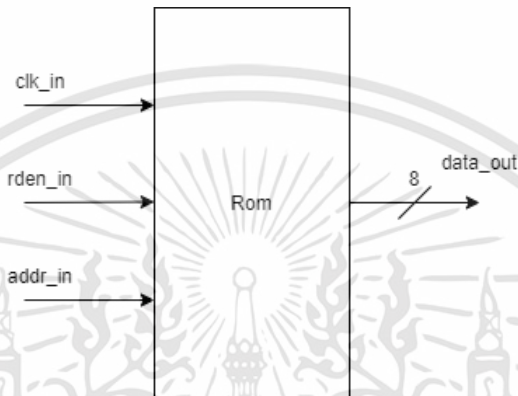
- clk_in
- areset_in (Asynchronous reset)
- start_in (ระบบเริ่มทำงานเมื่อ Start_in เป็นลอจิก 1)

Output

- rom_rden_out (เมื่อลอจิกเป็น 1 อ่านข้อมูลจาก rom)
- rom_addr_out (เพื่อเลือกแอดเดรสใน rom)
- barker_select_out (เพื่อเลือกความยาวของลำดับบาร์คเกอร์)
- load_barker_out (เพื่อส่งข้อมูลจากโมดูล Barker LUT)
- load_data_out (เพื่อส่งข้อมูลจาก rom)

3.3.3.2 ROM

เพื่อเป็นเครื่องมือที่เก็บข้อมูลที่จะเป็นข้อมูลขาเข้าของระบบไว้โดยมีสัญญาณจาก Control Unit เป็นตัวอ่านค่าภายในรอม ภายในรอมมีสัญญาณอินพุต และเอาต์พุตดังรูปที่ 3.34



รูปที่ 3.34 บล็อก Rom

Input

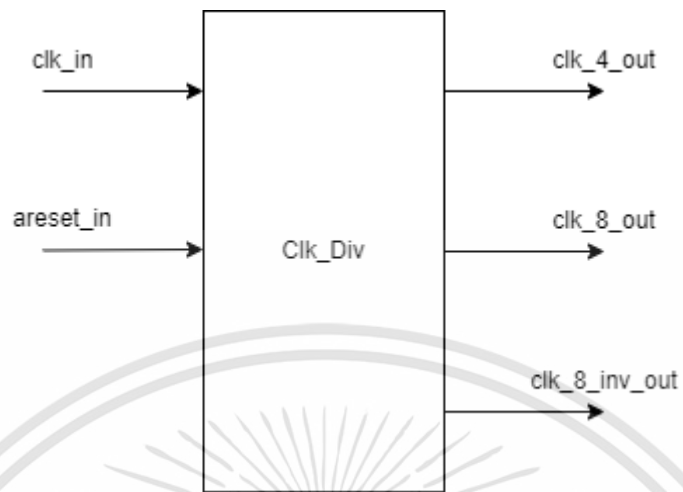
- clk_in
- rden_in (เมื่อลอจิกเป็น 1 อ่านข้อมูลจากรอม)
- add_in (เพื่อเลือกตัวแอดเดรสในรอม)

Output

- data_out (ข้อมูลที่อ่านจากรอมมี 8 บิต)

3.3.3.3 Clock Divider

เป็นวงจรที่ทำหน้าที่ลดความถี่ของความถี่คล็อกในระบบเพื่อให้ระบบมีสัดส่วนของการทำงานตามคล็อกที่ได้ออกแบบไว้โดยภายในระบบมีสัญญาณอินพุต และเอาต์พุตดังรูป 3.35



รูปที่ 3.35 บล็อก Clock Divider

Input

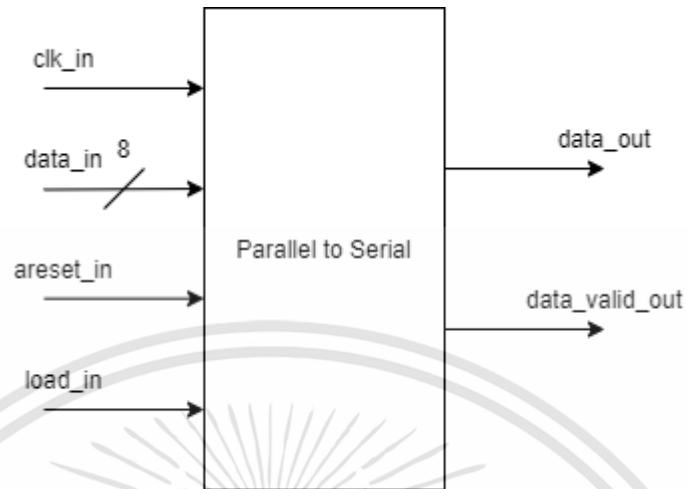
- clk_in
- areset_in (Asynchronous reset)

Output

- clk_4_out (หารความถี่ลง 4 เท่า)
- clk_8_out (หารความถี่ลง 8 เท่า)
- clk_8_inv_out (อินเวอร์สความถี่หาร 8 เท่า)

3.3.3.4 Parallel to Serial

ข้อมูลที่ได้จากรวมเป็นข้อมูลขนาน 8 บิตจึงต้องแปลงข้อมูลเป็นอนุกรม เพื่อที่จะทำงานกับบล็อกต่อไปได้ โดยภายในระบบมีสัญญาณอินพุต และเอาต์พุตดังรูป 3.36



รูปที่ 3.36 บล็อก Parallel to Serial

Input

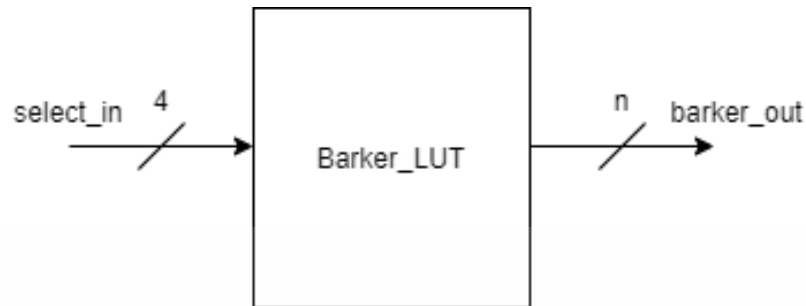
- clk_in
- data_in (8 bits)
- areset_in (Asynchronous reset)
- load_in (High = รับค่าอินพุต, Low = ส่งค่าเอาต์พุต)

Output

- data_out
- data_valid_out (เมื่อข้อมูลออกจากระบบลอจิกจะเป็น 1)

3.3.3.5 Barker LUT

ทำหน้าที่เพิ่มส่วนของข้อมูลบาร์คเกอร์ให้กับระบบเพื่อฝั่งรับจะได้สามารถถอดรหัสได้ โดยภายในระบบมีสัญญาณอินพุต และเอาต์พุตดังรูป 3.37



รูปที่ 3.37 บล็อก Barker_LUT

Input

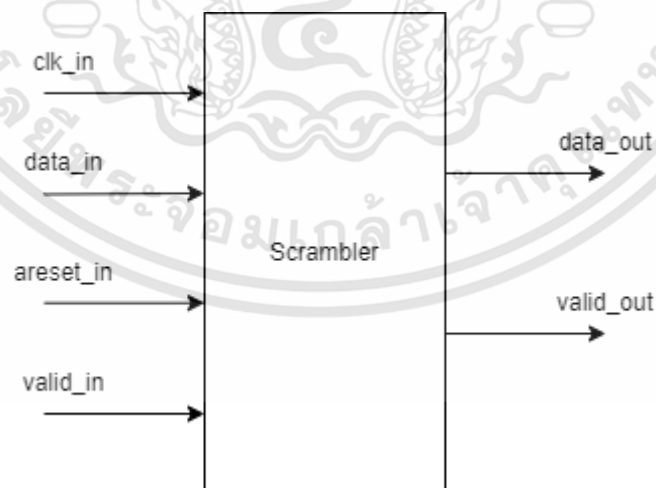
- select_in (เลือกความยาวของตัวบาร์คเกอร์)

Output

- Barker_out (4 bits)

3.3.3.6 Frame Based Scrambler

เพื่อทำหน้าที่ในการสแกนเบิตข้อมูลเพื่อให้ความน่าจะเป็นของข้อมูลบิต 0 กับบิต 1 มีความน่าจะเป็นใกล้เคียง โดยภายในระบบมีสัญญาณอินพุต และเอาต์พุตดังรูป 3.38



รูปที่ 3.38 บล็อก Scrambler

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Input

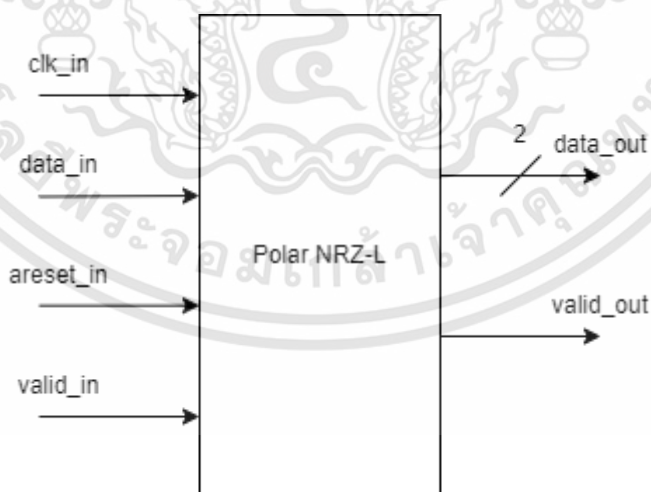
- clk_in
- data_in
- areset_in (Asynchronous reset)
- valid_in (วงจรสแกนเบิลจะทำงานเมื่อค่าเป็น 1)

Output

- data_out
- valid_out (ค่าเอาต์พุตของสแกนเบิลอยู่ในช่วงที่เป็น 1)

3.3.3.7 Polar NRZ-L

วงจร Polar NRZ-L เป็นวงจรที่ทำหน้าที่ในการทำให้ข้อมูลมีขั้ว และลดไฟตรงของระบบ โดยภายในระบบมีสัญญาณอินพุต และเอาต์พุตดังรูป 3.39



รูปที่ 3.39 บล็อก Polar NRZ-L

Input

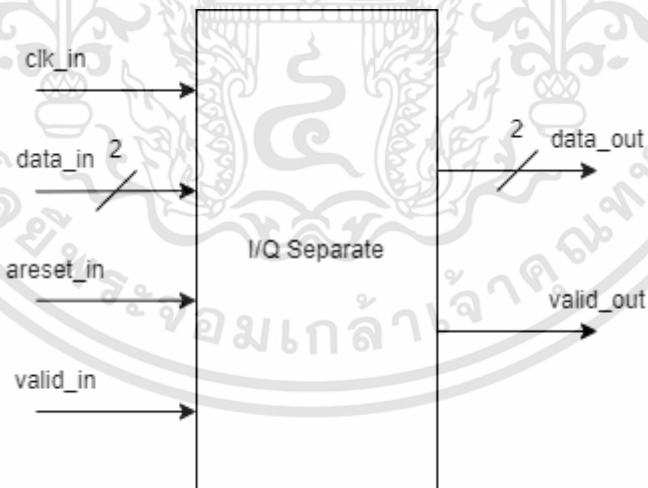
- clk_in
- data_in
- areset_in
- valid_in (ระบบทำงานเมื่อลอจิกเป็น 1)

Output

- data_out (2 bits)
- valid_out (เพื่อระบุสัญญาณเอาต์พุต)

3.3.3.8 I/Q Separate

I/Q Separate เป็นตัวแยกข้อมูลระหว่างบิตคู่ และบิตคี่ โดยภายในระบบมีสัญญาณอินพุต และเอาต์พุตดังรูป 3.40



รูปที่ 3.40 บล็อก I/Q Separate

Input

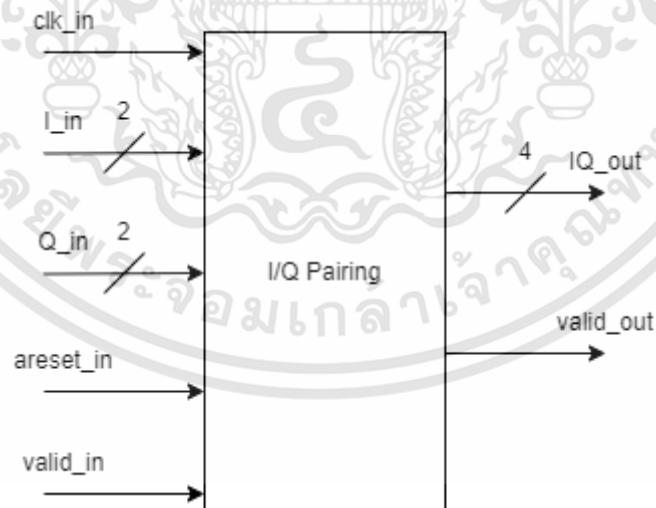
- clk_in
- data_in (NRZ 2 bits)
- areset_in (Asynchronous reset)
- valid_in (จะทำงานเมื่อเป็นลอจิก 1)

Output

- data_out (2 bits)
- valid_out (จะได้เอาต์พุตของระบบเมื่อลอจิกเป็น 1)

3.3.3.9 I/Q Pairing

ทำหน้าที่จับคู่บิตคู่ บิตคู่เป็น 1 สัญลักษณ์ เพื่อที่จะใช้กับบล็อก QPSK Mapping โดยภายในระบบมีสัญญาณอินพุต และเอาต์พุตดังรูป 3.41



รูปที่ 3.41 บล็อก I/Q Symbol

Input

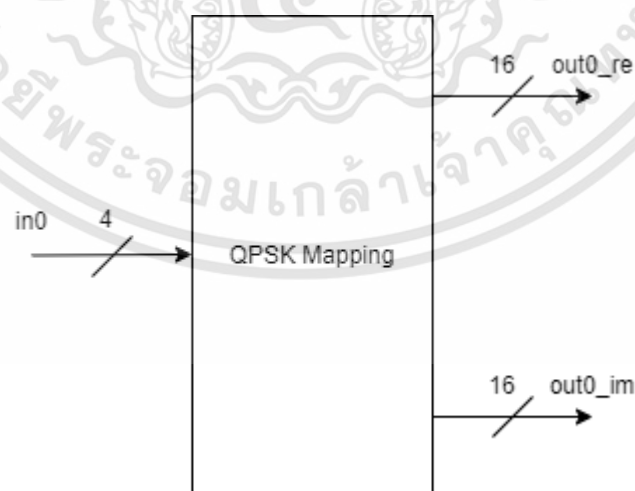
- clk_in
- I_in (2 bits)
- Q_in (2 bits)
- areset_in (Asynchronous reset)
- valid_in (ระบบจะทำงานเมื่อลอจิกเป็น 1)

Output

- IQ_out (4 bits)
- valid_out (เพื่อระบุสัญญาณเอาต์พุต)

3.3.3.10 QPSK Mapping

เป็นตัวมอดูเลชันโดยการแทนค่าข้อมูลขาเข้าด้วยข้อมูลที่เป็นมาตรฐานมอดูเลชันแบบ QPSK Mapping โดยภายในระบบมีสัญญาณอินพุต และเอาต์พุตดังรูป 3.42



รูป 3.42 บล็อก QPSK Mapping

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Input

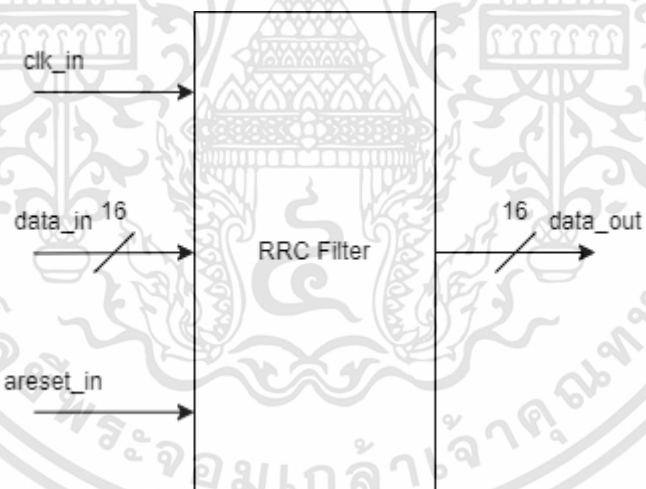
- in0 (4 bits)

Output

- out0_re (16 bits)
- out0_im (16 bits)

3.3.3.11 RRC filter

เป็นฟิลเตอร์ RRC 41 tabs เพื่อกรองสัญญาณอินพุตตามค่าสัมประสิทธิ์ (Coefficients) โดยภายในระบบมีสัญญาณอินพุต และเอาต์พุตดังรูป 3.43



รูปที่ 3.43 บล็อก RRC filter

Input

- clk_in
- data_in (16 bits)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- areset_in (Asynchronous reset)

Output

- data_out (16 bits)

3.2 เครื่องมือที่ใช้ในการทดลอง

เครื่องมือที่ใช้ในการทดลองระบบสื่อสารนี้ประกอบไปด้วย 2 ส่วนคือ ซอฟต์แวร์ และ ฮาร์ดแวร์

3.2.1 ซอฟต์แวร์

3.2.1.1 ซอฟต์แวร์ MATLAB R2020b ติดตั้งส่วนเสริมดังนี้

- 1) Simulink
- 2) Communications Toolbox
- 3) Computer Vision Toolbox
- 4) Communications Toolbox Support Package for USRP Radio
- 5) HDL Coder
- 6) HDL Coder Support Package for Xilinx Zynq Platform

3.2.1.2 ซอฟต์แวร์ Aldec Active HDL 8.3

3.2.1.3 ซอฟต์แวร์ Vivado 2019.2

3.2.2 ฮาร์ดแวร์

3.2.2.1 แลปท็อป ASUS TUF Gaming A15 FA506II-AL012T

รูปที่ 3.44 แลปท็อป ASUS TUF Gaming A15 FA506II-AL012T ใช้สำหรับรันซอฟต์แวร์แมทแลป ใช้หน่วยประมวลผล AMD Ryzen 5 4600H ใช้หน่วยประมวลผลกราฟิก NVIDIA GeForce GTX 1650 Ti (4GB GDDR5) และแรม 8 GB DDR4



รูปที่ 3.44 แลปทอป ASUS TUF Gaming A15 FA506II-AL012T

3.2.2.2 แลปทอป HP Pavilion Laptop 15-cc000

รูปที่ 3.45 แลปทอป HP Pavilion Laptop 15-cc000 ใช้สำหรับรันซอฟต์แวร์แมทแลป ใช้หน่วยประมวลผล Intel(R) Core(TM) i7-7500U ใช้หน่วยประมวลผลกราฟิก NVIDIA GeForce 940MX และแรม 4 GB



รูปที่ 3.45 แลปทอป HP Pavilion Laptop 15-cc000

3.2.2.3 USRP B210 2 เครื่อง

ฮาร์ดแวร์ประเภท SDR ผลิตโดย Ettus Research ทำหน้าที่เป็น RF Front-End ของระบบสื่อสาร QPSK ที่ได้ออกแบบ โดยมีหน่วยประมวลผล FPGA ของ Xilinx Spartan 6 ดังรูปที่ 3.46



รูปที่ 3.46 USRP B210

3.2.2.4 สายอากาศ 2.4 GHz 2 อัน

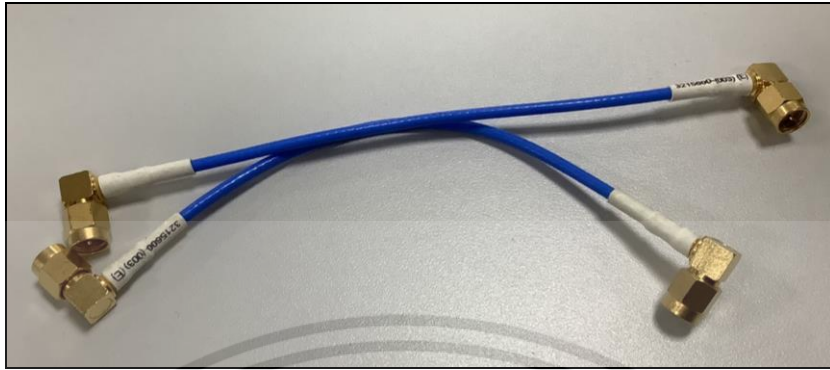
สายอากาศความถี่ 2.4 GHz ใช้สำหรับแพร่กระจายสัญญาณวิทยุ เพื่อใช้สื่อสารแบบไร้สาย ลักษณะการแพร่กระจาย (Radiation Pattern) เป็นแบบ Omnidirectional ดังรูปที่ 3.47



รูปที่ 3.47 สายอากาศ 2.4 GHz

3.2.2.5 สายโคแอกซ์เซียล 2 เส้น

สายโคแอกซ์เซียลหัวต่อ SMA ตัวผู้แบบโค้ง ใช้สำหรับนำสัญญาณ ในการสื่อสารผ่านสายดังรูปที่ 3.48



รูปที่ 3.48 สายโคแอกซ์เชิยล

3.2.2.6 Coaxial Directional Coupler 1 อัน

Coaxial Directional Coupler ใช้สำหรับคัปปลิงสัญญาณ เพื่อทำการวัดสัญญาณ QPSK ระหว่างทำการสื่อสาร โดยสัญญาณจากฝั่งส่งต่อเข้าที่หัวต่อ IN สัญญาณไปฝั่งรับต่อที่หัวต่อ OUT และสัญญาณที่ต้องการนำไปวัดกับเครื่อง Spectrum Analyzer ต่อเข้าที่หัวต่อ CPL ดังรูปที่ 3.49



รูปที่ 3.49 Coaxial Directional Coupler

3.2.2.7 Attenuator 10 dB และ 30 dB อย่างละ 1 อัน

Attenuator หรือตัวลดทอนสัญญาณ ใช้สำหรับลดทอนกำลังของสัญญาณ ก่อนที่สัญญาณจะเข้าสู่ตัวอุปกรณ์ เพื่อป้องกันความเสียหายจากความแรงของกำลังงานได้ ดังรูปที่ 3.50



รูปที่ 3.50 Attenuator

3.2.2.8 Spectrum Analyzer 1 เครื่อง

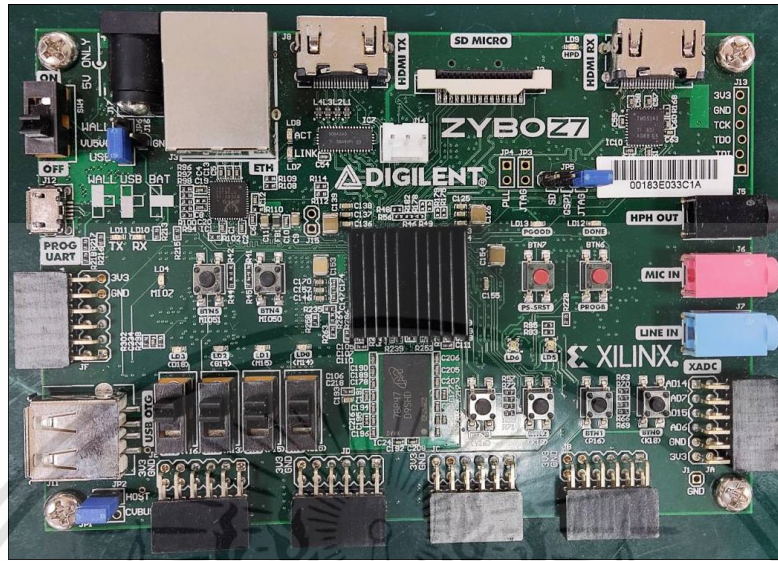
Spectrum Analyzer เป็นเครื่องมือวัด ใช้สำหรับทำการวัดค่าพารามิเตอร์ต่างๆ ของสัญญาณที่ต้องการวัด โดยมีฟังก์ชันต่างๆ เช่น กราฟโดเมนเวลา, กราฟสเปกตรัม, กราฟ Constellation และ Eye Diagram เป็นต้น ดังรูปที่ 3.51



รูปที่ 3.51 Spectrum Analyzer

3.2.2.9 ZYBO Z7-20 Development Board 1 บอร์ด

บอร์ด FPGA ของ Xilinx ใช้สำหรับการทดสอบระบบสื่อสาร QPSK โดยสัญญาณเอาต์พุตที่ได้จะทำการเชื่อมต่อกับพอร์ต I/O เพื่อทำการวัดผลสัญญาณทางดิจิทัล โดยมีลักษณะดังรูปที่ 3.52



รูปที่ 3.52 FPGA board (Zybo Z7-20)

3.2.2.10 Oscilloscope (R&S RTM 2024) 1 เครื่อง

สโคปสำหรับวัดสัญญาณไฟฟ้า สามารถวัดได้ทั้งสัญญาณอนาล็อก และ ดิจิตอล โดยความถี่ของสัญญาณที่วัดได้สูงสุดเท่ากับ 200 MHz ด้วยอัตราแซมเปิล 5 GSa/s ซึ่ง สามารถวัดสัญญาณดิจิตอลได้สูงสุด 16 แชลแนล ดังรูปที่ 3.53



รูปที่ 3.53 Oscilloscope (R&S RTM 2024)

3.2.2.11 USB B Micro cable 1 เส้น

สาย Micro B สำหรับเชื่อมต่อกับบอร์ด FPGA กับพอร์ต USB A กับแล็ปท็อป ดังรูปที่ 3.54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.54 สาย USB B Micro

3.2.2.12 Logic Probe (RT-ZL04) 2 อัน

โพรบที่ใช้สำหรับวัดสัญญาณไฟฟ้าที่เป็นสัญญาณทางดิจิทัล โดยแต่ละอันสามารถวัดได้มากที่สุด 8 แชนแนล ดังรูปที่ 3.55

3.2.2.13 สายไฟ (เส้นสีขาว) 16 เส้น

ใช้สำหรับต่อเข้ากับ logic probe และพอร์ต I/O ของบอร์ด FPGA เพื่อวัดสัญญาณ ดังรูปที่ 3.55

3.2.2.14 สายกราวด์ (เส้นสีดำ) 2 เส้น

ใช้สำหรับต่อสายกราวด์ระหว่าง logic probe และบอร์ด FPGA ดังรูปที่ 3.55



รูปที่ 3.55 Logic probe และสายไฟสำหรับวัดสัญญาณดิจิทัล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.15 Interface cable 2 เส้น

ใช้สำหรับเชื่อมต่อ oscilloscope กับ logic probe ด้วยหัว HDMI ดังรูปที่

3.56



รูปที่ 3.56 Interface cable

3.2.2.16 Pin header (ตัวผู้ 2 แถว 2.54 มม.) 2 อัน

ใช้สำหรับเชื่อมต่อสายไฟกับพอร์ต I/O บนบอร์ด FPGA ดังรูปที่ 3.57



รูปที่ 3.57 Pin header ตัวผู้ 2 แถว

3.3 ขั้นตอนการทดลอง

ปริญญานิพนธ์เรื่องการออกแบบระบบสื่อสาร QPSK ด้วย Software-Defined Radio (SDR) ได้ทำการทดลอง ไว้ดังต่อไปนี้

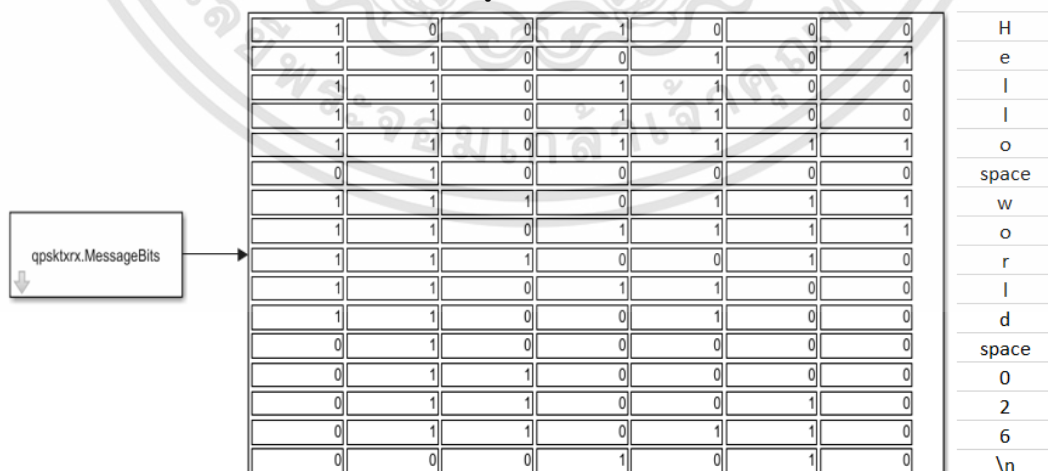
3.3.1 การทดสอบด้วยซอฟต์แวร์ MATLAB

ขั้นตอนการทดสอบระบบสื่อสาร QPSK ด้วยการทำ Simulation ได้ทำการทดสอบ โดยการออกแบบระบบสื่อสาร QPSK ทั้งภาคส่ง และภาครับ รวมถึงออกแบบช่องสัญญาณบนแลป ทอปหนึ่งเครื่องคือ ASUS TUF Gaming A15 FA506II-AL012T ไว้ดังหัวข้อที่ 3.1.1 ด้วยซอฟต์แวร์ MATLAB และ Simulink

โดยข้อความอินพุตของระบบถูกสร้างด้วยสคริปต์แมทแลป และทำการรันระบบสื่อสาร QPSK บนซอฟต์แวร์ Simulink เพื่อส่งข้อความดังกล่าวจากระบบสื่อสารภาคส่ง ไปยังภาครับ เพื่อถอดรหัสข้อความ และวัดผลของระบบสื่อสาร QPSK ที่ได้ออกแบบด้วยค่า EVM, MER และค่า BER

3.3.1.1 การสร้างข้อมูลอินพุต

ข้อมูลอินพุตที่ป้อนในระบบภาคส่งเป็นข้อความ “Hello world” ซึ่งข้อความนี้มีความยาวคือ 11 ตัวอักษร โดยข้อความนี้จะส่งเป็นจำนวน 20 ข้อความต่อ 1 เฟรม และใส่หมายเลขกำกับหลังข้อความ พร้อมคำสั่งขึ้นบรรทัดใหม่จะได้เป็น “Hello world ###\n” ซึ่ง ### คือหมายเลขระบุลำดับของข้อความ และ \n คือคำสั่งขึ้นบรรทัดใหม่ (LF) ซึ่งรวมกันได้เป็น 16 ตัวอักษร ดังนั้นใน 1 เฟรม จะมีจำนวนบิตอยู่เท่ากับ $16 \times 7 \times 20$ (แปลง ASCII เป็นไบนารี)



รูปที่ 3.58 ข้อความ “Hello world 026\n” ที่แปลงข้อมูลจาก ASCII เป็นไบนารี

รูปที่ 3.58 ตัวอย่างข้อมูลไบนารีของข้อความ “Hello world 026\n” โดย
 วัดเอาต์พุตจากการป้อนข้อความลงในบล็อก Constant ซึ่งอยู่ในซอฟต์แวร์ Simulink และรูปที่
 3.59 สคริปต์แมทแลปที่ใช้ในการสร้างข้อมูลอินพุต

```

1 SimParams.Message = 'Hello world';
2 SimParams.MessageLength = length(SimParams.Message) + 5; % 'Hello world 000\n'...
3 SimParams.NumberOfMessage = 20; % Number of messages in a frame
4
5 SimParams.PayloadLength = SimParams.NumberOfMessage * SimParams.MessageLength * 7; % 7 bits per characters
6
7 msgSet = zeros(SimParams.NumberOfMessage * SimParams.MessageLength, 1);
8 for msgCnt = 0 : SimParams.NumberOfMessage-1
9     msgSet(msgCnt * SimParams.MessageLength + (1 : SimParams.MessageLength)) = ...
10    sprintf('%s %03d\n', SimParams.Message, msgCnt);
11 end
12 bits = de2bi(msgSet, 7, 'left-msb');
13 SimParams.MessageBits_1 = bits(:);
14 SimParams.MessageBits=SimParams.MessageBits_1;
  
```

รูปที่ 3.59 สคริปต์แมทแลปที่ใช้ในการสร้างข้อมูลอินพุต

3.3.2 การทดลองระบบสื่อสาร QPSK ด้วย USRP B210

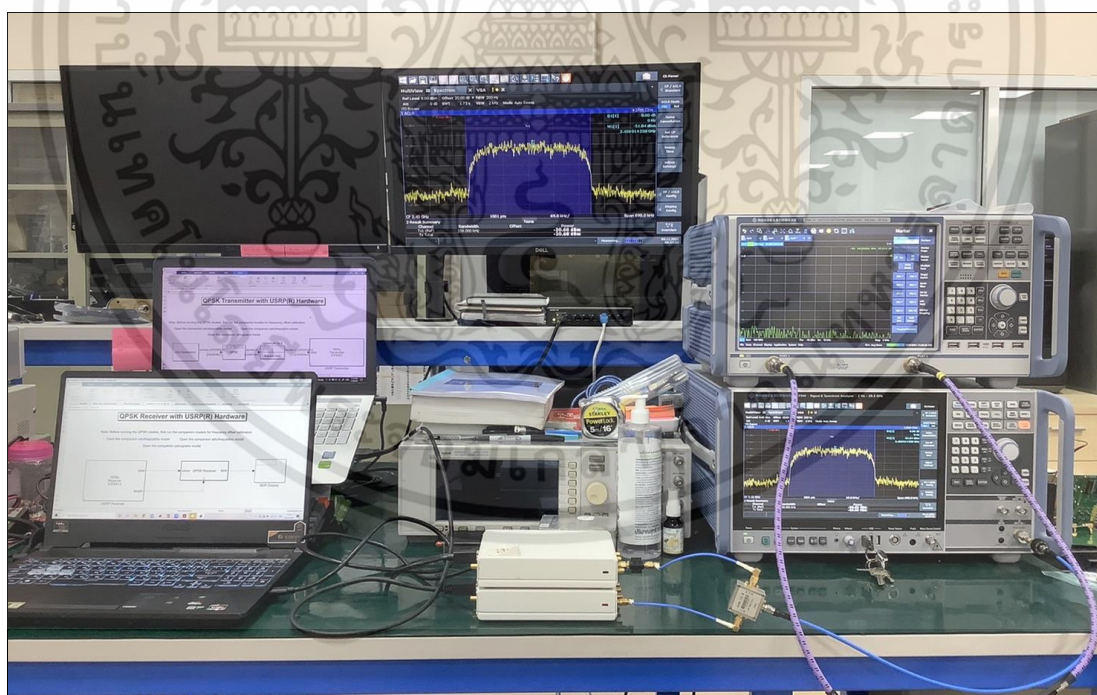
การทดสอบในส่วนที่ 2 นี้ได้ออกแบบระบบสื่อสาร QPSK โดยอ้างอิงองค์ประกอบ
 ของระบบสื่อสารตามที่ได้ออกแบบไว้ในส่วนที่ 1 โดยจะจำลองอัลกอริทึมในภาคส่ง และภาครับ บน
 แพลตฟอร์มเครื่อง โดยมีส่วนดังตารางที่ 3.2 ซึ่งสัญญาณดิจิทัลที่ฝั่งส่งจะผ่านกระบวนการ
 จัดการสัญญาณทางดิจิทัล ตามอัลกอริทึมที่ได้ออกแบบไว้ และถูกส่งผ่านสาย USB 3.0 เข้าสู่ RF
 Front-End ซึ่งก็คือ USRP B210 เพื่อทำการแปลงสัญญาณจากดิจิทัลเป็นอนาล็อก (Digital to
 Analog) และแปลงความถี่ IF เป็นความถี่ RF (Up Converter) ก่อนแพร่กระจายสัญญาณผ่าน
 สายอากาศ ไปยัง USRP B210 เครื่องรับ ซึ่งจะทำการกระบวนการตรงกันข้ามกับ USRP B210 ที่ฝั่งส่ง
 ซึ่งจะทำการแปลงความถี่ RF เป็นความถี่ IF (Down Converter) และแปลงสัญญาณอนาล็อกเป็น
 ดิจิทัล (Analog to Digital) เพื่อส่งสัญญาณดิจิทัลไปประมวลผล และถอดรหัสข้อความ

3.3.2.1 ทดลองระบบสื่อสาร QPSK ผ่านสายโคแอกซ์เชียล

การทดลองในส่วนนี้ ออกแบบเพื่อทดสอบระบบสื่อสาร QPSK ที่ได้
 ออกแบบไว้ด้วยซอฟต์แวร์ MATLAB และ Simulink โดยจำลองการสื่อสารผ่าน USRP B210 2
 เครื่อง ที่จำลองเป็นภาครับ และภาคส่ง และทำการวัดผลของสัญญาณ QPSK โดยส่งข้อความ
 Hello World โดยทำการติดตั้งอุปกรณ์ไว้ดังรูปที่ 3.60

ตารางที่ 3.2 เปรียบเทียบสเปกของแลปท็อประหว่างเครื่องส่ง และเครื่องรับที่ใช้ในการทดลอง

	เครื่องส่ง	เครื่องรับ
รุ่น	ASUS TUF Gaming A15 FA506II-AL012T	HP Pavilion Laptop 15 - cc000
หน่วยประมวลผล (CPU)	AMD Ryzen 5 4600H	Intel(R) Core(TM) i7-7500U CPU
หน่วยประมวลผลกราฟิกส์ (GPU)	NVIDIA GeForce GTX 1650 Ti (4GB GDDR5)	NVIDIA GeForce 940MX
Memory Size (RAM)	8 GB DDR4	4 GB



รูปที่ 3.60 อุปกรณ์สำหรับการทดสอบระบบสื่อสาร QPSK ผ่านสายโคแอกซ์เชียล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.60 แลปทอป HP Pavilion Laptop 15-cc000 (ด้านบน) ถูกใช้สำหรับรันระบบสื่อสาร QPSK ภาคส่ง และแลปทอป ASUS TUF Gaming A15 FA506II-AL012T (ด้านล่าง) ถูกใช้สำหรับรันระบบสื่อสาร QPSK ภาครับ

แลปทอปภาคส่งจะส่งข้อมูลดิจิทัลไปยัง USRP B210 ภาคส่ง ซึ่ง DAC ภายใน USRP B210 จะแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก และทำการมิกซ์ความถี่ โดยส่งออกไปที่ $f_c = 140 \text{ MHz}$ ผ่านสายโคแอกซ์เชียลไปยัง USRP B210 เครื่องรับซึ่งถูกลดเกนด้วย attenuator 10 dB ก่อน ADC ภายใน USRP B210 เครื่องรับ จะทำการแปลงสัญญาณอนาล็อกเป็นดิจิทัล และส่งไปยังแลปทอปที่ทำงานอัลกอริทึมของระบบสื่อสาร QPSK ภาครับ

ระหว่างทำการสื่อสารระหว่าง USRP B210 2 เครื่อง ผ่านสายโคแอกซ์เชียล ได้ทำการต่อกับ Directional Coupler และต่อ Attenuator 30 dB ที่พอร์ตของเครื่อง Spectrum Analyzer เพื่อทำการวัดพารามิเตอร์ต่างๆ ของสัญญาณ QPSK

3.3.2.2 ทดลองระบบสื่อสาร QPSK ผ่านสายอากาศ

ขั้นตอนนี้จะทดลองระบบสื่อสาร QPSK ผ่านสายอากาศ โดยปรับให้ USRP B210 ส่งสัญญาณออกที่ความถี่เท่ากับ 4 GHz ซึ่งยังเป็นช่วงความถี่ที่สายอากาศสามารถใช้งานได้ เนื่องจากที่ความถี่ 2.4 GHz อาจมีสัญญาณ Wi-Fi รบกวนภายในบริเวณที่ใช้ทดสอบเยอะ ซึ่งจัดเตรียมอุปกรณ์ไว้ดังรูปที่ 3.61



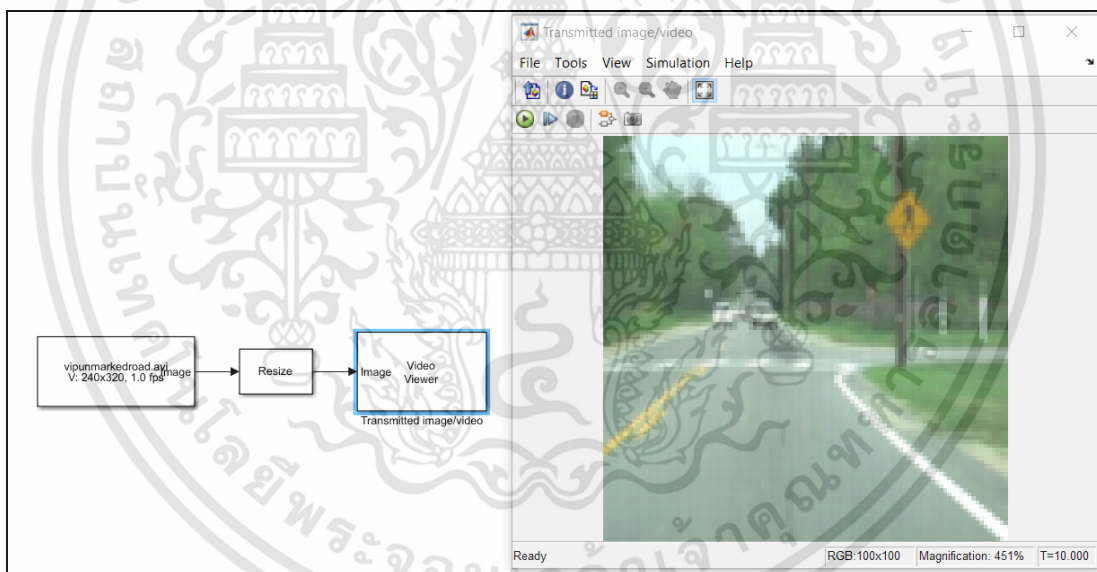
รูปที่ 3.61 อุปกรณ์สำหรับการทดสอบระบบสื่อสาร QPSK ผ่านสายอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2.3 ทดลองระบบสื่อสาร QPSK ด้วย Symbol Rate ต่างๆ

ขั้นตอนนี้เป็น การทดสอบระบบสื่อสาร QPSK ด้วยการส่งสัญญาณวิดีโอ ขนาด 100*100 pixels Frame Rate เท่ากับ 1 fps ดังรูปที่ 3.62 ด้วยค่า Symbol Rate ที่แตกต่างกัน เพื่อหาค่า Symbol Rate สูงสุดโดยที่ระบบสื่อสารสามารถทำงานได้ ค่า Symbol Rate และค่า Master Clock Rate สามารถตั้งค่า USRP B210 ผ่านบล็อก SDRu Transmitter และ SDRu Receiver โดยค่าดังกล่าว จะสัมพันธ์กับค่า Interpolation ของ USRP B210 ฝั่งส่ง และค่า Decimation ของ USRP B210 ฝั่งรับ ดังสมการที่ 3.2

$$\text{Interpolation, Decimation} = \frac{\text{Master clock rate}}{\text{Symbol rate} \times 2} \quad (3.2)$$

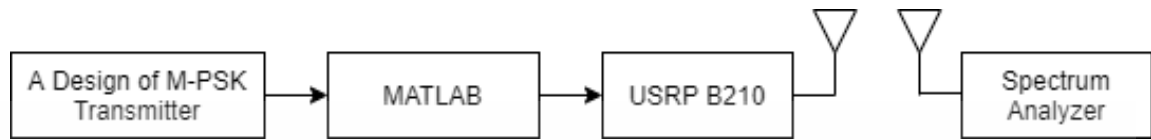


รูปที่ 3.62 วิดีโอขนาด 100*100 pixels 1 fps

3.3.2.4 ทดสอบระบบสื่อสาร M-PSK ด้วย USRP B210

ขั้นตอนนี้ได้ทำการทดสอบระบบสื่อสารภาคส่งด้วยการมอดูเลตแบบ QPSK, 8-PSK และ 16-PSK เพื่อทำการเปรียบเทียบผลคุณภาพของสัญญาณที่มีต่อสัญญาณรบกวน ซึ่งก็คือ

C/N และทำการเปรียบเทียบประสิทธิภาพของแบนด์วิทในการส่งสัญญาณ โดยมีการออกแบบบล็อกไดอะแกรมสำหรับการทดลองดังรูปที่ 3.63

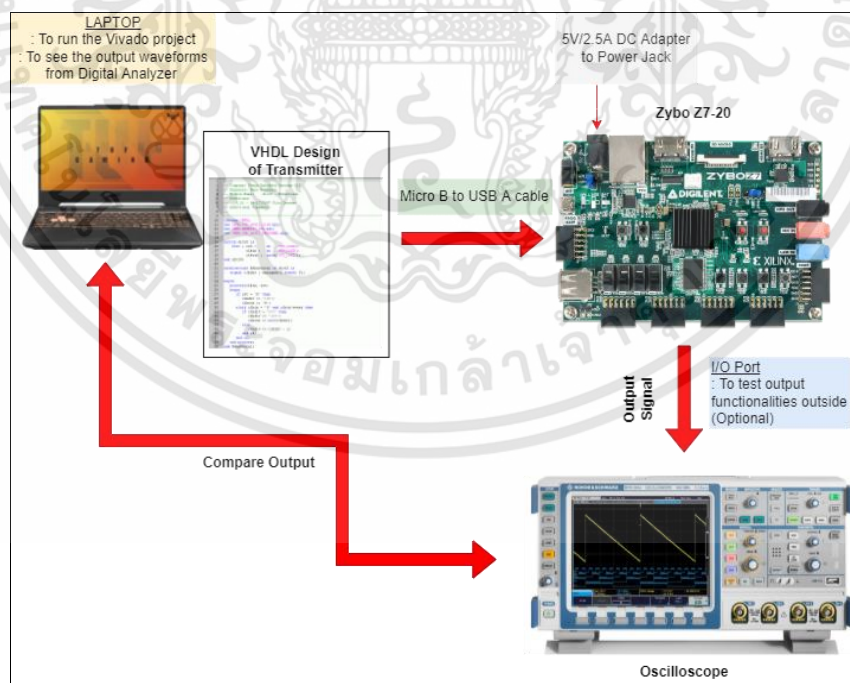


รูปที่ 3.63 บล็อกไดอะแกรมการทดสอบระบบสื่อสาร M-PSK ภาดส่งด้วย USRP B210

3.3.3 การทดลองระบบสื่อสาร QPSK ด้วย FPGA

การทดลองระบบสื่อสาร QPSK ด้วย FPGA มีขั้นตอนการออกแบบการทดลองโดยนำผลที่ได้จากการจำลองจากการออกแบบระบบสื่อสาร QPSK ภาดส่งจากซอฟต์แวร์ MATLAB และ Aldec Active-HDL มาเปรียบเทียบกับผลของสัญญาณดิจิทัลจากบอร์ด FPGA ที่วัดด้วยออสซิลโลสโคป โดยมีขั้นตอนการทดสอบดังรูปที่ 3.64

สัญญาณดิจิทัลเอาต์พุตต่างๆ จากบอร์ด FPGA จะถูกวัดผลด้วยออสซิลโลสโคป โดยมีการแมปค่าอินพุต และเอาต์พุตตามตำแหน่งของแชลแนลใน logic probe กับตำแหน่งของพินในบอร์ด FPGA ตามตารางที่ 2.3 เพื่อวัดผลการทดลองไว้ดังต่อไปนี้



รูปที่ 3.64 การออกแบบการทดลองด้วย FPGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3.1 Scrambler

การวัดผลการทดลองของบล็อก Scrambler มีการตั้งค่าพอร์ตกับสัญญาณอินพุต และเอาต์พุต ดังรูปที่ 3.3

ตารางที่ 3.3 แผนที่ชื่อสัญญาณอินพุต เอาต์พุตของบล็อก Scrambler กับพอร์ตของ Zybo Z7-20 และ logic analyzer โดยป้อนความถี่ของสัญญาณนาฬิกาเท่ากับ 15.625 MHz

Scrambler	Zybo Z7-20	Logic Analyzer	
Input	clk_in	K17	-
	areset_in	Y16 (Button)	-
	start_in	T16 (Switch)	-
	valid_in	W13 (Switch)	-
Output	clk_out_0	JB-V8	Pin0
	valid_in_0	JB-W8	Pin1
	start_in_0	JB-U7	Pin2
	data_in_0	JB-V7	Pin3
	data_out	JB-Y7	Pin4
	valid_out	JB-Y6	Pin5

3.3.3.2 Polar NRZ-L

การวัดผลการทดลองของบล็อก Polar NRZ-L มีการตั้งค่าพอร์ตกับสัญญาณอินพุต และเอาต์พุต ดังรูปที่ 3.4 โดยป้อนความถี่ของสัญญาณนาฬิกาเท่ากับ 15.625 MHz

ตารางที่ 3.4 แผนที่สัญญาณอินพุต เอาต์พุตของบล็อก Polar NRZ-L กับพอร์ตของ Zybo Z7-20 และ logic analyzer

Polar NRZ-L		Zybo Z7-20	Logic Analyzer
Input	clk_in	K17	-
	areset_in	Y16 (Button)	-
	start_in	T16 (Switch)	-
	valid_in	W13 (Switch)	-
Output	clk_in_0	JB-V8	Pin0
	valid_in_0	JB-W8	Pin1
	start_in_0	JB-U7	Pin2
	data_in_0	JB-V7	Pin3
	data_out(1)	JB-Y7	Pin4
	data_out(0)	JB-Y6	Pin5
	valid_out	JB-V6	Pin6

3.3.3.3 I/Q Separate

การวัดผลการทดลองของบล็อก I/Q Separate มีการตั้งค่าพอร์ตกับสัญญาณอินพุต และเอาต์พุต ดังรูปที่ 3.5 โดยป้อนความถี่ของสัญญาณนาฬิกาเท่ากับ 7.8125 MHz

ตารางที่ 3.5 แผนที่สัญญาณอินพุต เอาต์พุตของบล็อก I/Q Separate กับพอร์ตของ Zybo Z7-20 และ logic analyzer

I/Q Separate		Zybo Z7-20	Logic Analyzer
Input	clk_in	K17	-
	areset_in	Y16 (Button)	-

ตารางที่ 3.5 (ต่อ) แมปชื่อสัญญาณอินพุต เอาต์พุตของบล็อก I/Q Separate กับพอร์ตของ Zybo Z7-20 และ logic analyzer

I/Q Separate		Zybo Z7-20	Logic Analyzer
Input	start_in	T16 (Switch)	-
	valid_in	W13 (Switch)	-
Output	clk_out_0	JB-V8	Pin0
	valid_in_0	JB-W8	Pin1
	start_in_0	JB-U7	Pin2
	data_in_0	JB-V7	Pin3
	data_in_1	JB-Y7	Pin4
	data_out(0)	JB-Y6	Pin5
	data_out(1)	JB-V6	Pin6
	valid_out	JB-W6	Pin7

3.3.3.4 I/Q Pairing

การวัดผลการทดลองของบล็อก I/Q Pairing มีการตั้งค่าพอร์ตกับสัญญาณอินพุต และเอาต์พุต ดังรูปที่ 3.6 โดยป้อนความถี่ของสัญญาณนาฬิกาเท่ากับ 15.625 MHz

ตารางที่ 3.6 แมปชื่อสัญญาณอินพุต เอาต์พุตของบล็อก I/Q Pairing กับพอร์ตของ Zybo Z7-20 และ logic analyzer

I/Q Pairing		Zybo Z7-20	Logic Analyzer
Input	clk_in	K17	-
	areset_in	Y16 (Button)	-
	start_in	T16 (Switch)	-

ตารางที่ 3.6 (ต่อ) แมปชื่อสัญญาณอินพุต เอาต์พุตของบล็อก I/Q Pairing กับพอร์ตของ Zybo Z7-20 และ logic analyzer

I/Q Pairing		Zybo Z7-20	Logic Analyzer
Input	valid_in	W13 (Switch)	-
Output	clk_out_0	JB-V8	Pin0
	valid_in_0	JB-W8	Pin1
	start_in_0	JB-U7	Pin2
	i_in_0	JB-V7	Pin3
	i_in_1	JB-Y7	Pin4
	q_in_0	JB-Y6	Pin5
	q_in_1	JB-V6	Pin6
	lq_out(0)	JC-V15	Pin7
	iq_out(1)	JC-W15	Pin8
	lq_out(2)	JC-T11	Pin9
	iq_out(3)	JC-T10	Pin10
	valid_out	JC-W14	Pin11

3.3.3.5 QPSK Mapping

การวัดผลการทดลองของบล็อก QPSK Mapping มีการตั้งค่าพอร์ตกับสัญญาณอินพุต และเอาต์พุต ดังรูปที่ 3.7 โดยป้อนความถี่ของสัญญาณนาฬิกาเท่ากับ 15.625 MHz

ตารางที่ 3.7 แมปชื่อสัญญาณอินพุต เอาต์พุตของบล็อก QPSK Mapping กับพอร์ตของ Zybo Z7-20 และ logic analyzer

QPSK Mapping		Zybo Z7-20	Logic Analyzer
Input	clk_in	K17	-
	areset_in	Y16 (Button)	-
	start_in	T16 (Switch)	-
Output	clk_in_0	JB-V8	Pin 0
	start_in_0	JB-W8	Pin 1
	data_in_0	JB-U7	Pin 2
	data_in_1	JB-V7	Pin 3
	real_out(0)	JC-V15	Pin 4
	real_out(1)	JC-W15	Pin 5
	real_out(2)	JC-T11	Pin 6
	real_out(3)	JC-T10	Pin 7
	real_out(4)	JC-W14	Pin 8
	real_out(5)	JC-Y14	Pin 9
	real_out(6)	JC-T12	Pin 10
	real_out(7)	JC-U12	Pin 11
	real_out(8)	JD-T14	Pin 12
	real_out(9)	JD-T15	Pin 13
real_out(10)	JD-P14	Pin 14	
real_out(11)	JD-R14	Pin 15	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.7 (ต่อ) แมปชื่อสัญญาณอินพุต เอาต์พุตของบล็อก QPSK Mapping กับ พอร์ตของ Zybo Z7-20 และ logic analyzer

QPSK Mapping	Zybo Z7-20	Logic Analyzer	
Output	real_out(12)	JD-U14	Pin 16
	real_out(13)	JD-U15	Pin 17
	real_out(14)	JD-V17	Pin 18
	real_out(15)	JD-V18	Pin 19
	imag_out(0)	JC-V15	Pin 20
	imag_out(1)	JC-W15	Pin 21
	imag_out(2)	JC-T11	Pin 22
	imag_out(3)	JC-T10	Pin 23
	imag_out(4)	JC-W14	Pin 24
	imag_out(5)	JC-Y14	Pin 25
	imag_out(6)	JC-T12	Pin 26
	imag_out(7)	JC-U12	Pin 27
	imag_out(8)	JD-T14	Pin 28
	imag_out(9)	JD-T15	Pin 29
	imag_out(10)	JD-P14	Pin 30
	imag_out(11)	JD-R14	Pin 31
	imag_out(12)	JD-U14	Pin 32
imag_out(13)	JD-U15	Pin 33	
imag_out(14)	JD-V17	Pin 34	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.7 (ต่อ) แมปชื่อสัญญาณอินพุต เอาต์พุตของบล็อก QPSK Mapping กับพอร์ตของ Zybo Z7-20 และ logic analyzer

QPSK Mapping		Zybo Z7-20	Logic Analyzer
Output	imag_out(15)	JD-V18	Pin 35

3.3.3.6 Tx RRC Filter

การวัดผลการทดลองของบล็อก Tx RRC Filter มีการตั้งค่าพอร์ตกับสัญญาณอินพุต และเอาต์พุต ดังรูปที่ 3.8 โดยป้อนความถี่ของสัญญาณนาฬิกาเท่ากับ 31.25 MHz

ตารางที่ 3.8 แมปชื่อสัญญาณอินพุต เอาต์พุตของบล็อก Tx RRC Filter กับพอร์ตของ Zybo Z7-20 และ logic analyzer

Tx RRC Filter		Zybo Z7-20	Logic Analyzer
Input	clk_in	K17	-
	areset_in	Y16 (Button)	-
	start_in	T16 (Switch)	-
Output	clk_in_0	JA-N15	Pin 0
	start_in_0	JA-L14	Pin 1
	data_in_0	JB-V8	Pin 2
	data_in_1	JB-W8	Pin 3
	data_in_2	JB-U7	Pin 4
	data_in_3	JB-V7	Pin 5
	data_in_4	JB-Y7	Pin 6
	data_in_5	JB-Y6	Pin 7
	data_in_6	JB-V6	Pin 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.8 (ต่อ) แมปชื่อสัญญาณอินพุต เอาต์พุตของบล็อก Tx RRC Filter กับพอร์ตของ Zybo Z7-20 และ logic analyzer

Tx RRC Filter		Zybo Z7-20	Logic Analyzer
Output	data_in_7	JB-W6	Pin 9
	data_in_8	JC-V15	Pin 10
	data_in_9	JC-W15	Pin 11
	data_in_10	JC-T11	Pin 12
	data_in_11	JC-T10	Pin 13
	data_in_12	JC-W14	Pin 14
	data_in_13	JC-Y14	Pin15
	data_in_14	JC-T12	Pin16
	data_in_15	JC-U12	Pin17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

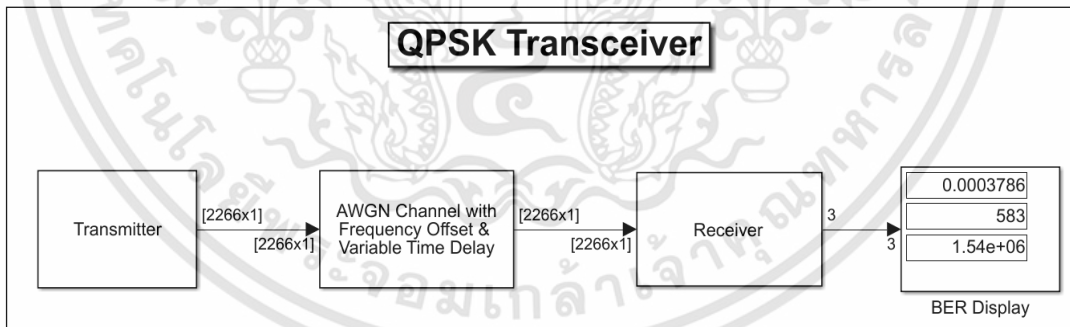
ผลการทดลอง

ปฏิญานินทร์เรื่องการออกแบบระบบสื่อสาร QPSK ด้วย Software-Defined Radio (SDR) มีผลการทดลองดังนี้

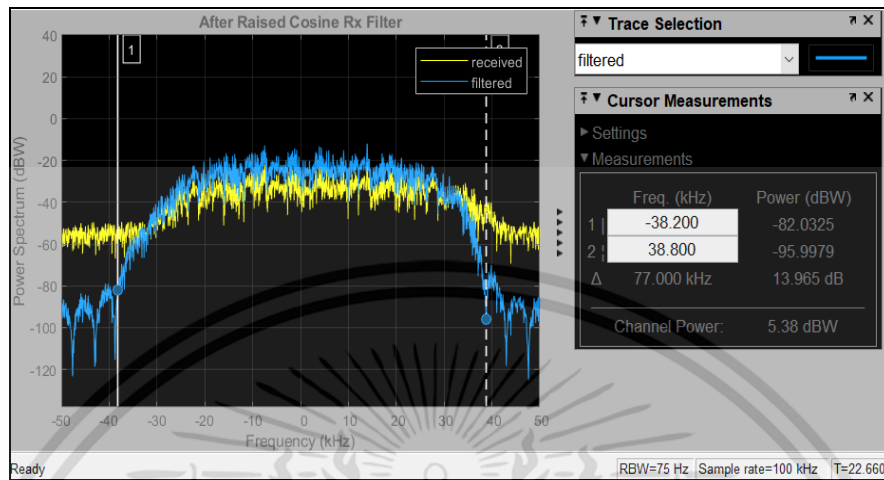
4.1 ผลการทดสอบระบบสื่อสาร QPSK ด้วยซอฟต์แวร์ MATLAB

ระบบสื่อสารภาคส่ง และภาครับ QPSK มีบล็อกไดอะแกรมดังภาพที่ 4.1 โดยข้อความ อินพุตจะถูกส่งผ่าน Transmitter เข้าสู่บล็อก AWGN Channel with Frequency Offset & Variable Time Delay ซึ่งทำหน้าที่เปรียบเสมือนการจำลองการสื่อสารผ่านช่องสัญญาณไร้สาย อาทิเช่น การสร้างสัญญาณรบกวนเข้าไปภายในระบบ ซึ่งที่นี้ก็คือ สัญญาณรบกวนเกาส์สีขาวแบบ บวก (Additive White Gaussian Noise, AWGN), การจำลองผลของการเลื่อนความถี่ และเฟสของ สัญญาณ และการสร้างดีเลย์ของสัญญาณ และจึงเข้าสู่บล็อก Receiver ที่ทำหน้าที่เป็นภาครับ

รูปที่ 4.2 แสดงสเปกตรัมของสัญญาณที่ภาครับ สเปกตรัมเส้นสีเหลืองคือสเปกตรัมที่ ระบบสื่อสารภาครับ รับผิดชอบ ส่วนสเปกตรัมเส้นสีฟ้าเกิดจากสัญญาณที่ผ่าน RRC ฟิเตอร์ที่ภาครับ ซึ่งมีแบนด์วิธเท่ากับ 77 kHz

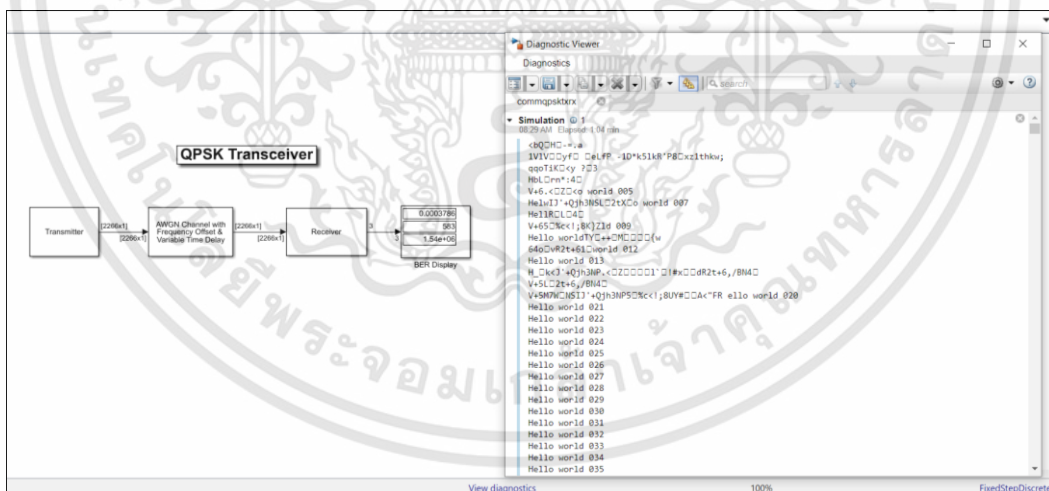


รูปที่ 4.1 บล็อกไดอะแกรมระบบการสื่อสาร QPSK บนซอฟต์แวร์ Simulink



รูปที่ 4.2 เปรียบเทียบสเปกตรัมที่ระบบสื่อสารภาครับหลังจากผ่าน Rx RRC Filter

จากรูปที่ 4.1 เมื่อทำการรันซอฟต์แวร์เพื่อทดสอบระบบสื่อสาร QPSK ได้ออกแบบไว้ จะสามารถแสดงอัตราความผิดพลาดบิต (BER) ของข้อมูลที่ส่ง และรับได้เท่ากับ $1.54e+06$ และสามารถแสดงข้อความที่รับได้ดังรูปที่ 4.3

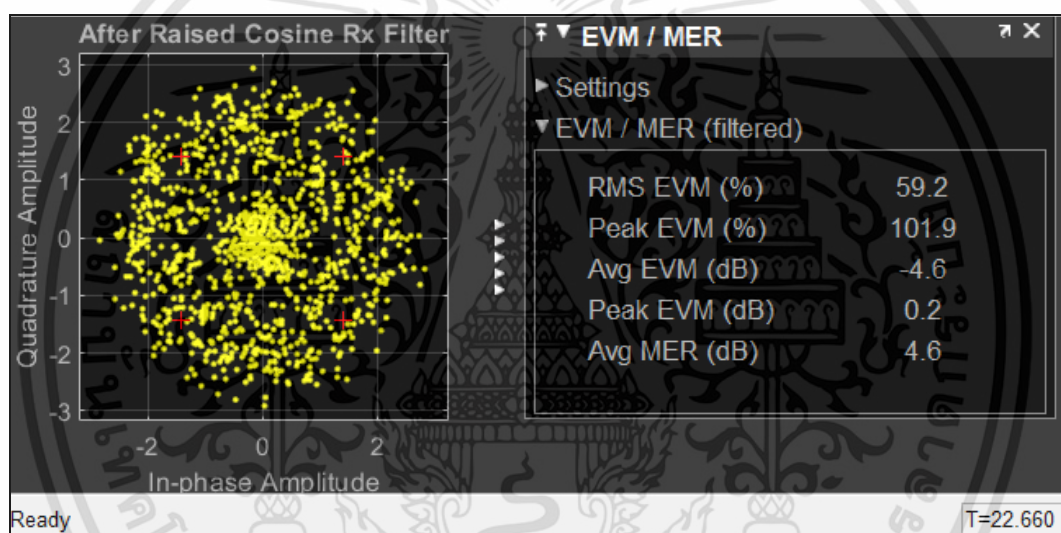


รูปที่ 4.3 ข้อความที่ถูกถอดรหัสที่ภาครับของระบบสื่อสาร QPSK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

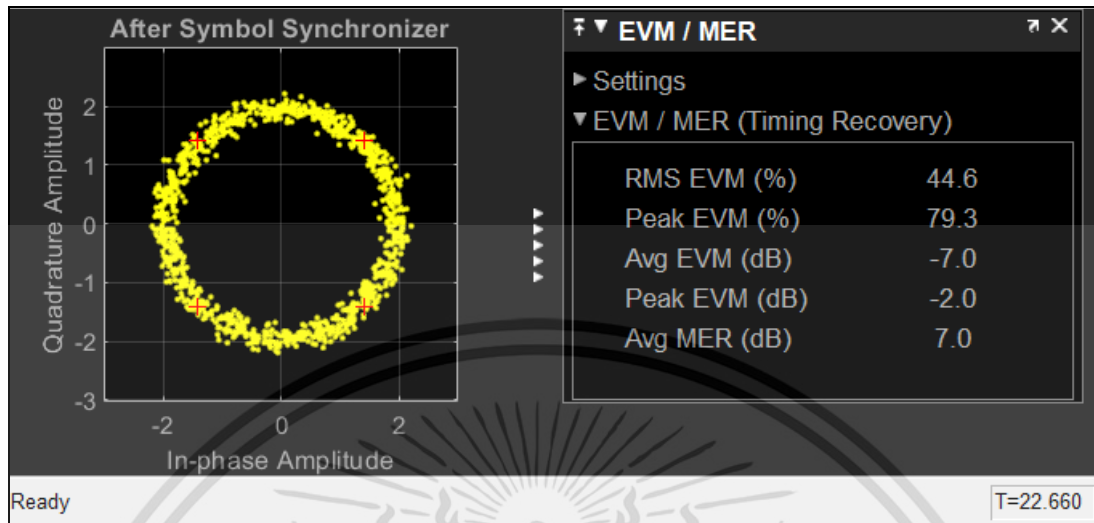
จากรูปที่ 4.3 จะเห็นได้ว่าข้อความที่ได้รับในช่วงแรกมีการผิดเพี้ยนไปจากเดิม แต่ความสามารถของระบบสื่อสารภาครับได้กู้คืนข้อมูลกลับคืนมา ทำให้ได้ข้อความที่ถูกต้องตามข้อความฝั่งส่ง ซึ่งสามารถตรวจสอบพฤติกรรมของการทำงานอัลกอริทึมการซิงโครไนซ์ข้อมูล โดยรูปที่ 4.4 เป็นกราฟ Constellation ของสัญญาณที่รับได้หลังผ่าน Rx RRC filter

จากรูปที่ 4.4 จะเห็นได้ว่ามีค่า EVM ที่สูง และค่า MER ที่ต่ำ ซึ่งค่า EVM เฉลี่ยเท่ากับ -4.6 dB ค่า EVM สูงสุดเท่ากับ 0.2 dB และค่า MER เฉลี่ย เท่ากับ 4.6 dB ซึ่งสำหรับค่า MER ในการมอดูเลตแบบ QPSK ถือว่ามีค่าต่ำมาก ดังนั้นภายในระบบจึงมีการซิงโครไนซ์เพื่อให้ได้สัญญาณที่มีข้อมูลที่ถูกต้องกลับคืนมา

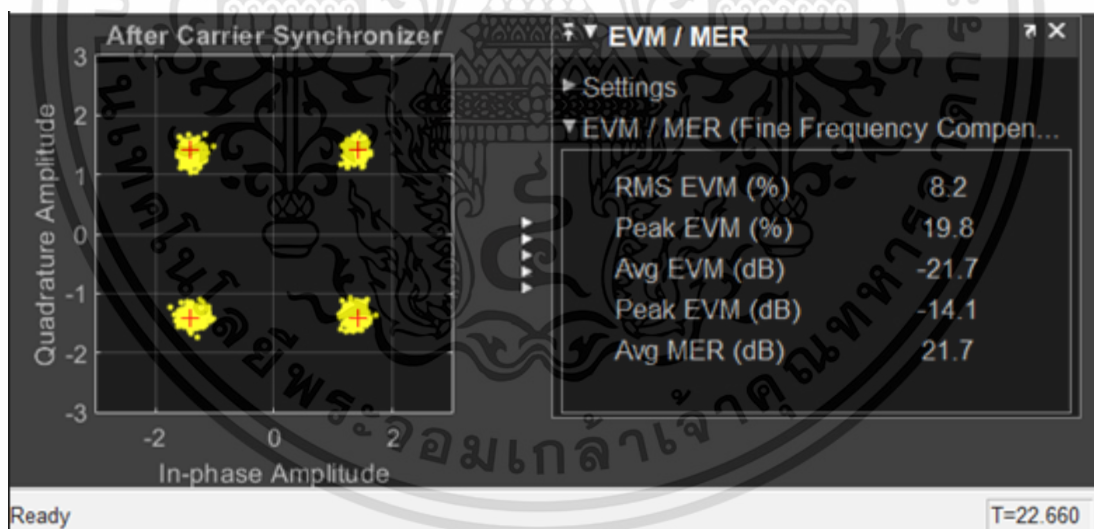


รูปที่ 4.4 กราฟ Constellation หลังผ่าน Rx RRC filter

รูปที่ 4.5 และรูปที่ 4.6 เป็นการกู้คืนความผิดพลาดของสัญญาณข้อมูลโดยผ่าน Synchronizer ซึ่งรูปที่ 4.5 เป็นกราฟ Constellation หลังผ่าน Symbol Synchronizer และรูปที่ 4.6 เป็นกราฟ Constellation หลังผ่าน Carrier Synchronizer จะเห็นได้ว่าค่า EVM หลังจากการกู้คืนสัญญาณจะมีค่าที่ลดลง และค่า MER มีค่าที่เพิ่มขึ้น



รูปที่ 4.5 กราฟ Constellation หลังผ่าน Symbol Synchronizer



รูปที่ 4.6 กราฟ Constellation หลังผ่าน Carrier Synchronizer

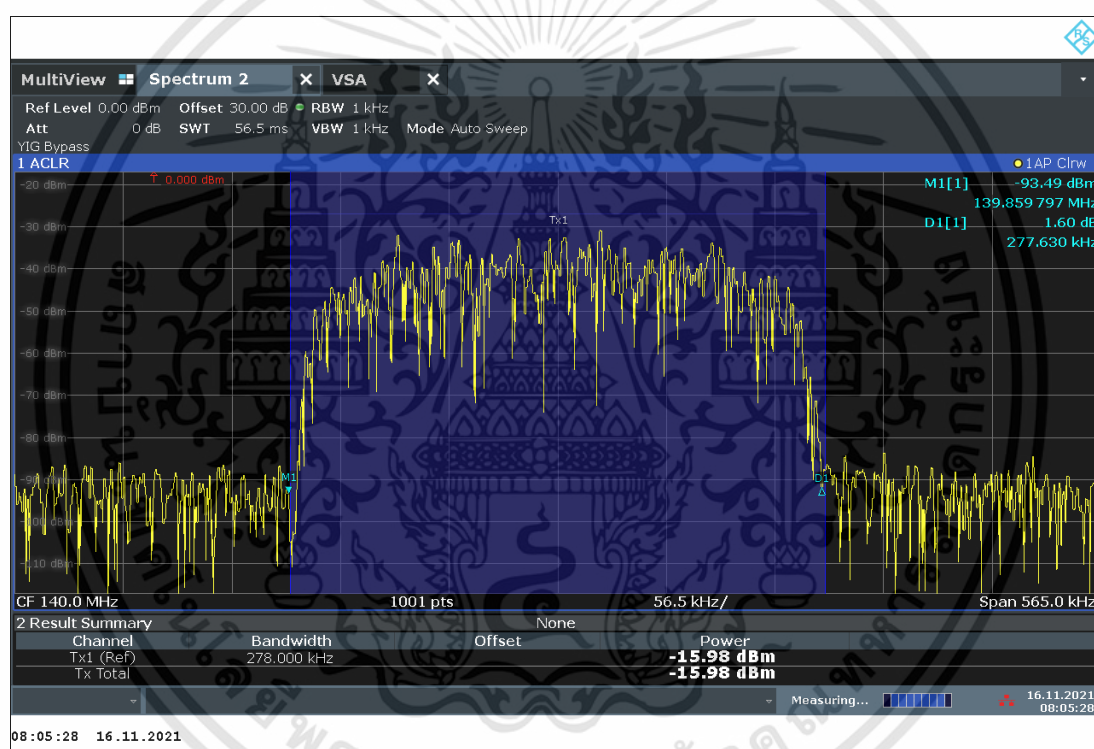
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดสอบระบบสื่อสาร QPSK ด้วย USRP B210

ผลการทดสอบระบบสื่อสาร QPSK ด้วย USRP B210 ได้แบ่งการเก็บผลการทดสอบเป็น 2 ส่วน ได้แก่ การทดสอบระบบสื่อสารผ่านสายโคแอกซ์เชียล และการทดสอบระบบสื่อสารผ่านสายอากาศ

4.2.1 ผลการทดสอบระบบสื่อสาร QPSK ด้วย USRP B210 ผ่านสายโคแอกซ์เชียล

เช็ล

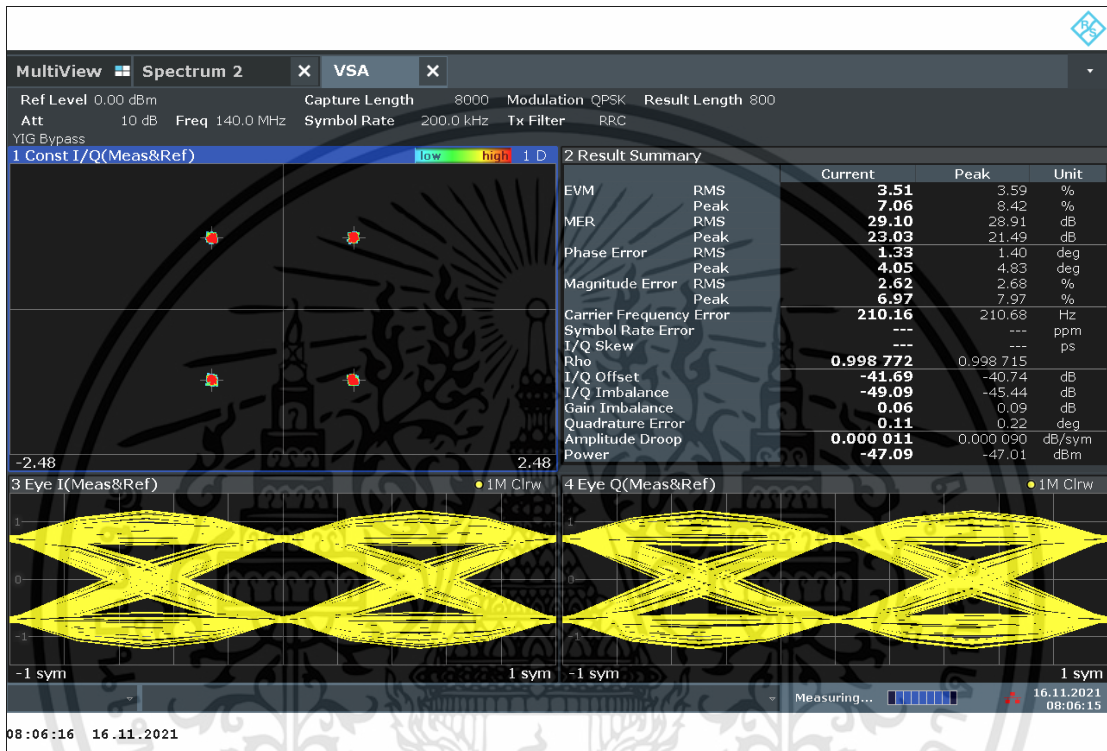


รูปที่ 4.7 สเปกตรัมของสัญญาณ QPSK รับผ่านสายโคแอกซ์เชียล

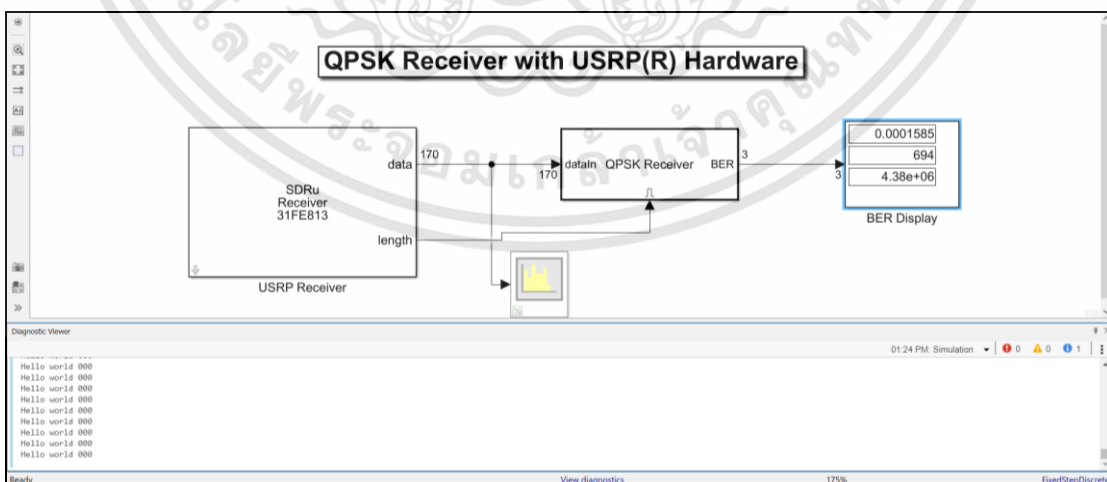
จากรูปที่ 4.7 เป็นกราฟสเปกตรัมของสัญญาณ QPSK ที่ส่งจาก USRP B210 เครื่องส่งไปยัง USRP B210 เครื่องรับ โดยวัดที่ $f_c = 140 \text{ MHz}$ แบนด์วิทที่วัดได้เท่ากับ 278 kHz ค่า C/N เท่ากับ 70 dBm กำลังของสัญญาณเท่ากับ -15.98 dBm

จากรูปที่ 4.8 เป็นหน้าต่าง VSA ของ Spectrum Analyzer ซึ่งได้แสดงผลต่างๆ ได้แก่ กราฟ Constellation, กราฟแสดงผลต่างๆ, กราฟ Eye diagram ของสัญญาณ I และ

สัญญาณ Q จะเห็นได้ว่าสัญญาณ QPSK ที่วัดได้มีค่า EVM (RMS) เท่ากับ 3.51% และค่า MER (RMS) เท่ากับ 29.10 dB Eye Diagram ของสัญญาณ I และสัญญาณ Q จะสามารถสังเกตเห็นว่ามีลักษณะของดวงตาที่เปิดกว้าง แต่ยังมีผลกระทบจากสัญญาณรบกวน



รูปที่ 4.8 หน้าต่าง VSA แสดงผลของสัญญาณ QPSK รับผ่านสายโคแอกซ์เซียม



รูปที่ 4.9 ผลของหน้าจอของระบบสื่อสาร QPSK ภาครับ รับผ่านสายโคแอกซ์เซียม

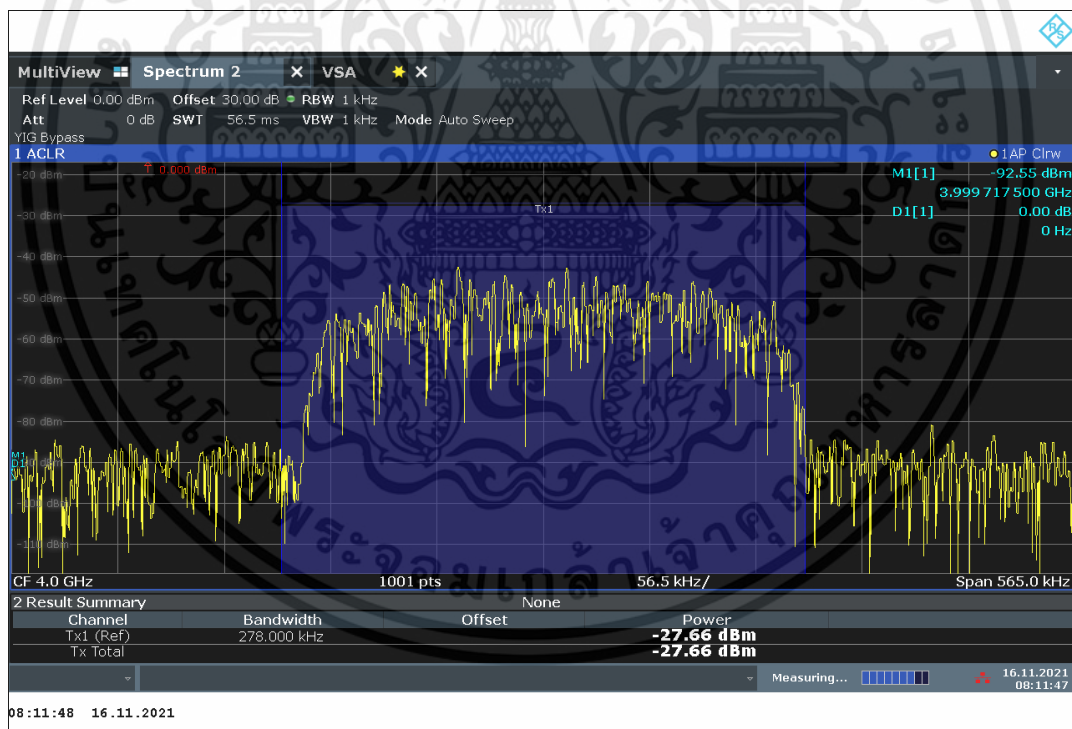
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.9 จะเห็นได้ว่าที่ระบบสื่อสาร QPSK ภาครับสามารถถอดรหัสข้อความ “Hello World” ออกมาได้ และสามารถวัดค่า BER ได้เท่ากับ 4.38×10^{-6}

4.2.2 ผลการทดลองระบบสื่อสาร QPSK ด้วย USRP B210 ผ่านสายอากาศ

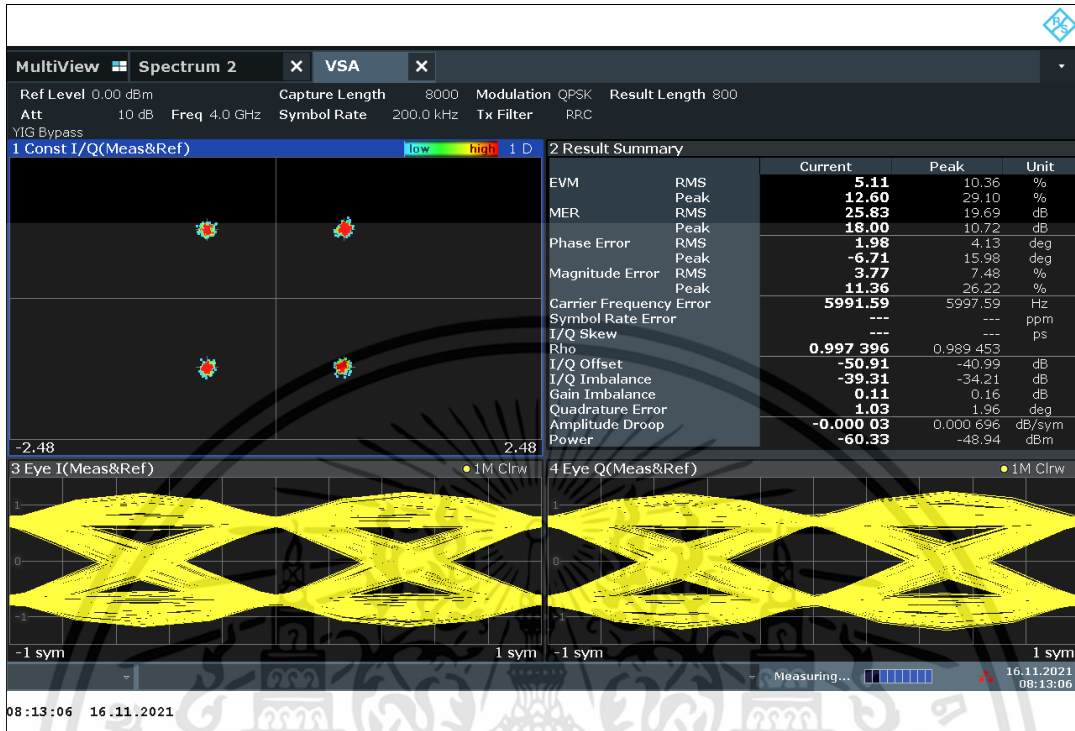
จากรูปที่ 4.10 แสดงสเปกตรัมของสัญญาณ QPSK ที่รับด้วยสายอากาศ ที่ $f_c = 4 \text{ GHz}$ แบนด์วิธที่รับได้เท่ากับ 278 kHz ค่า C/N เท่ากับ 60 dBm กำลังของสัญญาณเท่ากับ -27.66 dBm

จะเห็นได้ว่าเมื่อส่งสัญญาณผ่านสายอากาศอัตราส่วนระหว่างกำลังของสัญญาณแคเรียร์ และกำลังของสัญญาณรบกวน มีค่าต่ำลง กำลังของเครื่องรับที่รับได้ก็มีค่าต่ำกว่าส่งผ่านสายโคแอกซ์เชียวด้วยเช่นกัน มีผลมาจากการส่งสัญญาณไร้สายถูกรบกวนได้ง่ายกว่าการส่งผ่านสายนำสัญญาณ



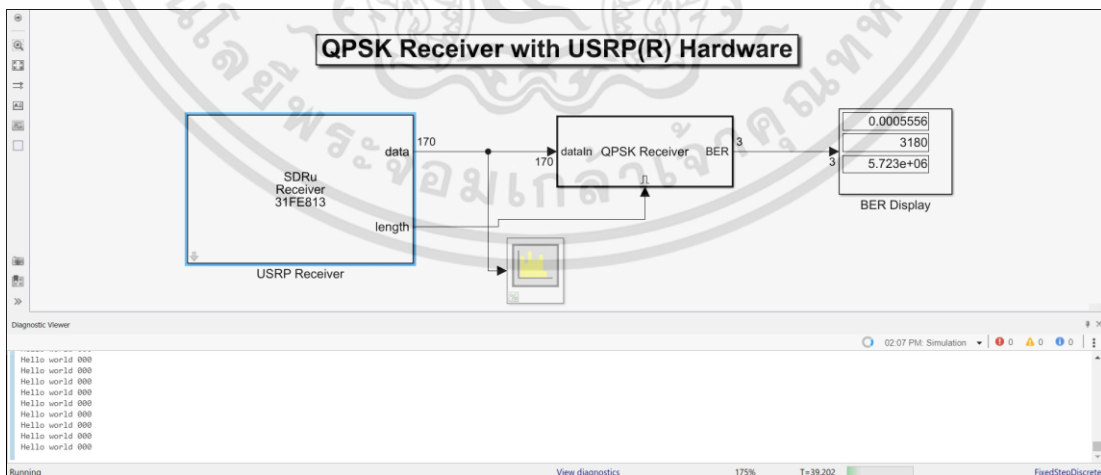
รูปที่ 4.10 สเปกตรัมของสัญญาณ QPSK รับผ่านสายอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 หน้าต่าง VSA แสดงผลของสัญญาณ QPSK รับผ่านสายอากาศ

รูปที่ 4.11 จะเห็นได้ว่า Eye Diagram ของสัญญาณ I และสัญญาณ Q มีลักษณะของดวงตาที่เปิดกว้าง แต่ยังมีผลกระทบจากสัญญาณรบกวน ซึ่งเมื่อวัดค่า EVM (RMS) จะได้เท่ากับ 5.11% และ MER (RMS) เท่ากับ 25.83 dB ซึ่งค่า EVM มีค่าที่สูงขึ้น และค่า MER มีค่าที่ต่ำกว่าการส่งผ่านสายนำสัญญาณ เนื่องจากสัญญาณถูกรบกวนมากขึ้น เมื่อสื่อสารแบบไร้สาย



รูปที่ 4.12 ผลของหน้าจอของระบบสื่อสาร QPSK ภาครับ รับผ่านสายอากาศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.12 จะเห็นได้ว่าที่ระบบสื่อสาร QPSK ภาครับสามารถถอดรหัสข้อความ “Hello World” ออกมาได้ และสามารถวัดค่า BER ได้เท่ากับ 5.72×10^{-6}

4.2.3 ผลการทดลองระบบสื่อสาร QPSK ด้วย USRP B210 ที่ Symbol Rate ต่างๆ

ผลจากการทดลองในขั้นตอนนี้ ได้จากการส่งข้อมูลที่เป็นไฟล์วิดีโอขนาด 100×100 pixels ที่ Frame Rate เท่ากับ 1 fps โดยทำการปรับค่า Symbol Rate และ Master Clock Rate ของ USRP B210 ดังตารางที่ 4.1 ได้ผลดังนี้

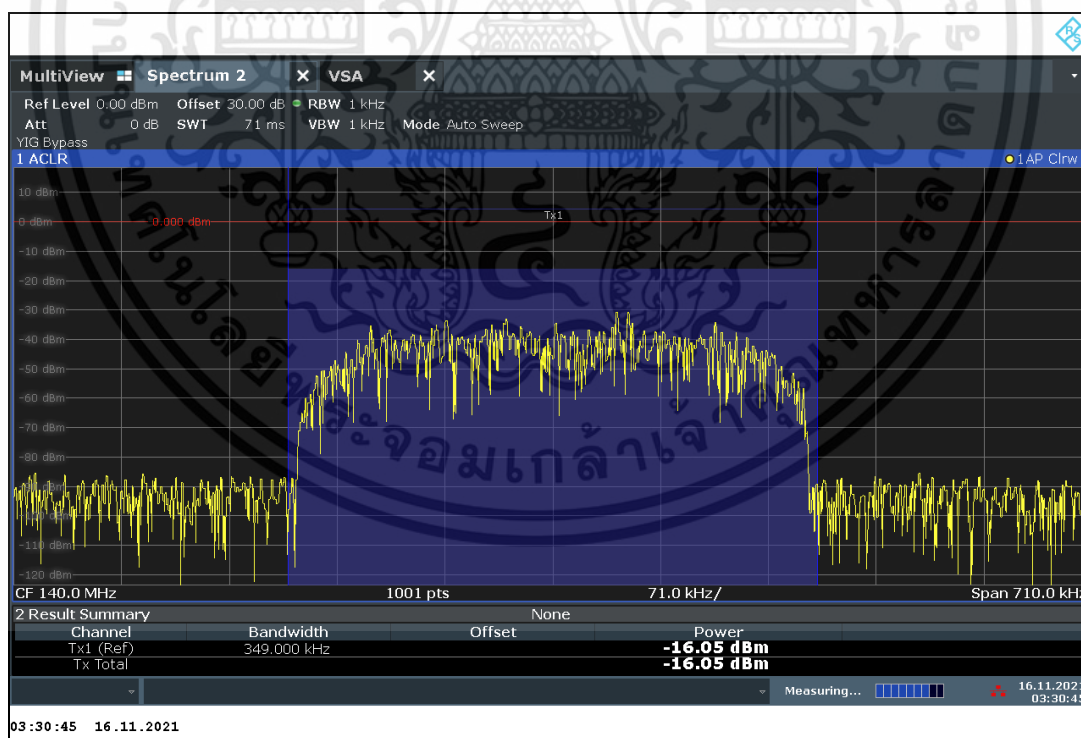
ตารางที่ 4.1 ผลการทดลองระบบสื่อสาร QPSK ที่ symbol rate ต่างๆ

Test case	Symbol rate (MHz)	Master clock rate (MHz)	USRP rate (MHz)	Interpolation factor	Decimation factor	Tx LED status	Rx LED status
1	2.500	5	5	1	1	Blink	Blink
2	1.250	5	2.5	2	2	Blink	Blink
3	0.625	5	1.25	4	4	Blink	Blink
4	0.500	5	1	5	5	Blink	Blink
5	0.250	5	0.5	10	10	Red	Green
6	0.200	5	0.4	12.5	12.5	Blink	Blink
7	1.000	10	2	5	5	Blink	Blink
8	0.625	10	1.25	8	8	Blink	Blink
9	0.500	10	1	10	10	Blink	Blink
10	0.250	10	0.5	20	20	Red	Green
11	0.200	10	0.4	25	25	Red	Green
12	2.000	20	4	5	5	Blink	Blink
13	1.000	20	2	10	10	Blink	Blink
14	0.625	20	1.25	16	16	Blink	Blink
15	0.500	20	1	20	20	Blink	Blink

ตารางที่ 4.1 (ต่อ) ผลการทดลองระบบสื่อสาร QPSK ที่ symbol rate ต่างๆ

Test case	Symbol rate (MHz)	Master clock rate (MHz)	USRP rate (MHz)	Interpolation factor	Decimation factor	Tx LED status	Rx LED status
16	0.250	20	0.5	40	40	Red	Green
17	0.200	20	0.4	50	50	Red	Green
18	0.500	50	1	50	50	Blink	Blink
19	0.250	50	0.5	100	100	Blink	Blink
20	0.560	56	1.12	50	50	Blink	Blink

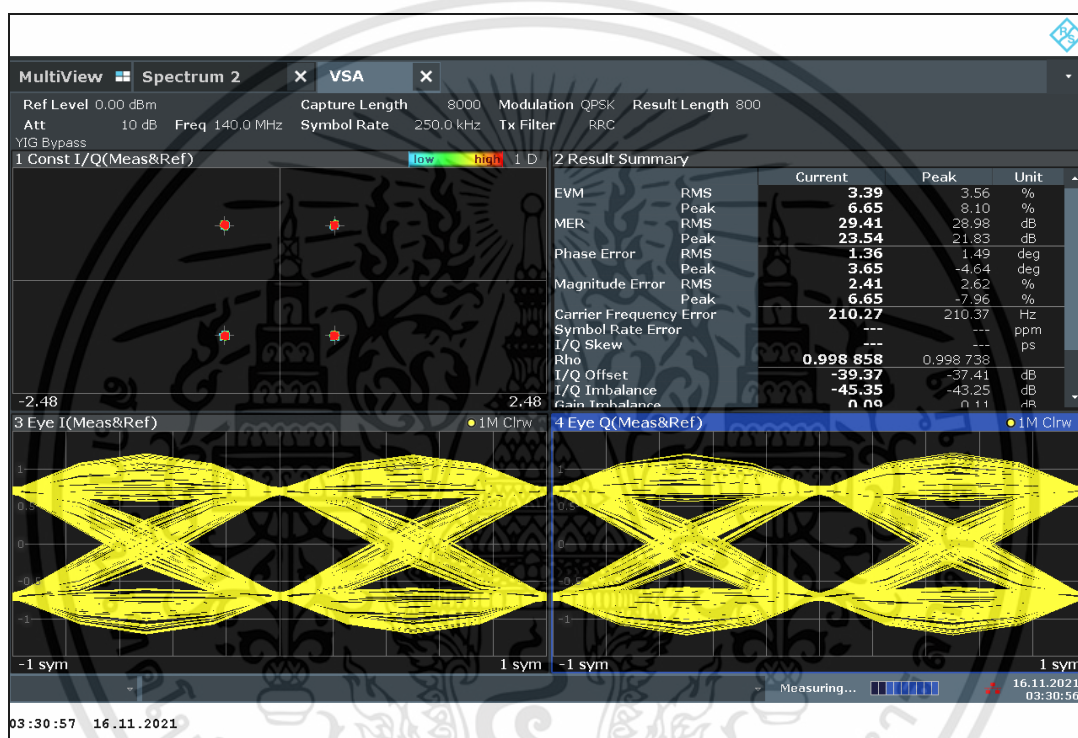
จากตารางที่ 4.1 ค่า Symbol Rate และ Master Clock Rate ที่กำหนดค่าในสคริปต์แมทแลป จะถูกคำนวณเป็นค่า Interpolation ของ USRP B210 เครื่องส่ง และค่า Decimation ของ USRP B210 ในเครื่องรับดังสมการที่ (4.1)



รูปที่ 4.13 สเปกตรัมของสัญญาณ QPSK ส่งที่ Symbol Rate เท่ากับ 0.25 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อทดสอบระบบสื่อสาร QPSK ด้วยการส่งข้อมูลที่ Symbol Rate ต่างๆ นี้พบว่า การส่งข้อมูลด้วย Symbol Rate และใช้ Master Clock Rate ที่เหมาะสม ส่งผลต่อระบบสื่อสารด้วย สังเกตได้จากสถานะของไฟ LED ของ USRP B210 ดังตารางที่ 2.1 แลaptop เครื่องส่ง ASUS TUF Gaming A15 FA506II-AL012T สามารถส่งข้อมูลที่ Symbol Rate สูงสุดเท่ากับ 0.25 MHz ด้วย Master Clock Rate สูงสุดเท่ากับ 20 MHz แสดงผลดังรูปที่ 4.13 และรูปที่ 4.14

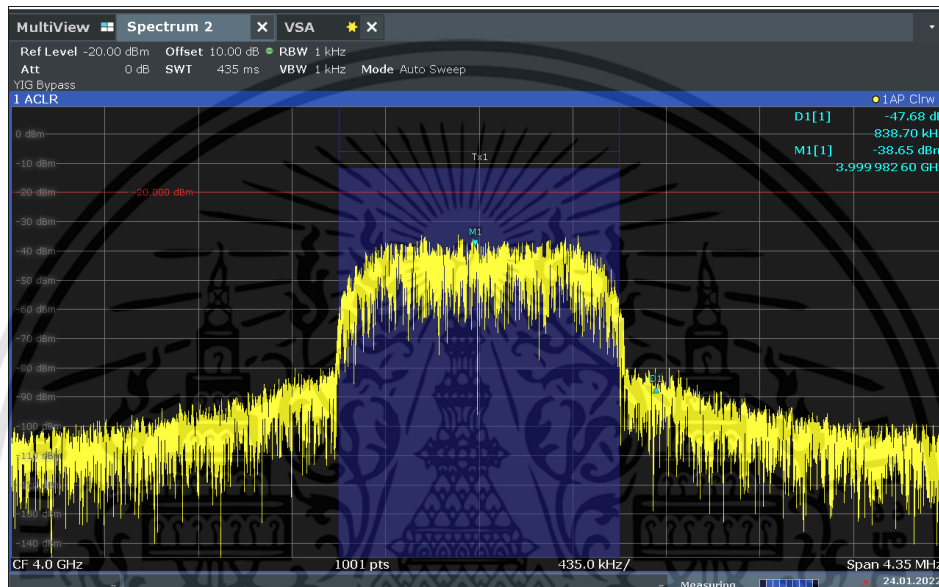


รูปที่ 4.14 หน้าต่าง VSA ของสัญญาณ QPSK ส่งที่ Symbol Rate เท่ากับ 0.25 MHz

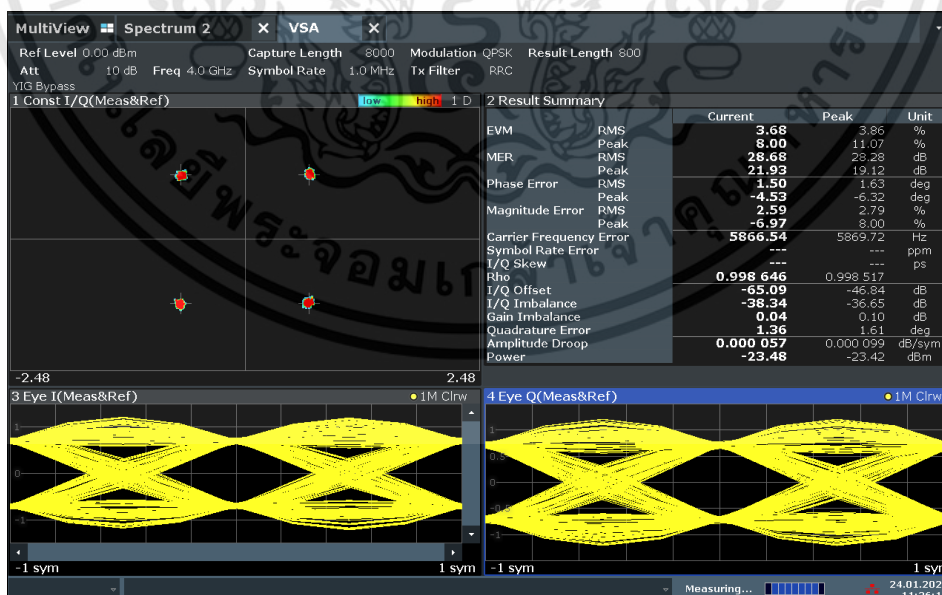
จากรูปที่ 4.13 สามารถวัดแบนด์วิธได้เท่ากับ 349 kHz ค่า C/N เท่ากับ 60 dBm และกำลังของสัญญาณเท่ากับ -16.05 dBm รูปที่ 4.14 จะเห็นได้ว่า Eye Diagram ของสัญญาณ I และสัญญาณ Q มีลักษณะของดวงตาที่เปิดกว้าง แต่ยังมีผลกระทบจากสัญญาณรบกวน Constellation ของสัญญาณ QPSK ที่ถูกต้องวัดได้ที่ Symbol Rate เท่ากับ 0.25 MHz โดยมีค่า EVM (RMS) เท่ากับ 3.39% และค่า MER (RMS) เท่ากับ 29.41 dB

4.2.4 ผลการทดสอบระบบสื่อสาร M-PSK ด้วย USRP B210

จากการทดสอบในขั้นตอนนี้ได้ทำการเพิ่มช่องสัญญาณรบกวนเข้าไปภายในระบบสื่อสารที่ได้ออกแบบ และทำการปรับค่า Eb/No ของบล็อก AWGN Channel ดังรูปที่ 7 เพื่อทดสอบค่า C/N ที่เปลี่ยนไปของสัญญาณที่วัดได้จากเครื่องวัดสเปกตรัม โดยตั้งค่าเกณฑ์ของ SDR เท่ากับ 60 dB ส่งที่ความถี่เท่ากับ 4 GHz ผ่านสายอากาศ



รูปที่ 4.15 สเปกตรัมของสัญญาณ QPSK ที่ Eb/No เท่ากับ 100 dB

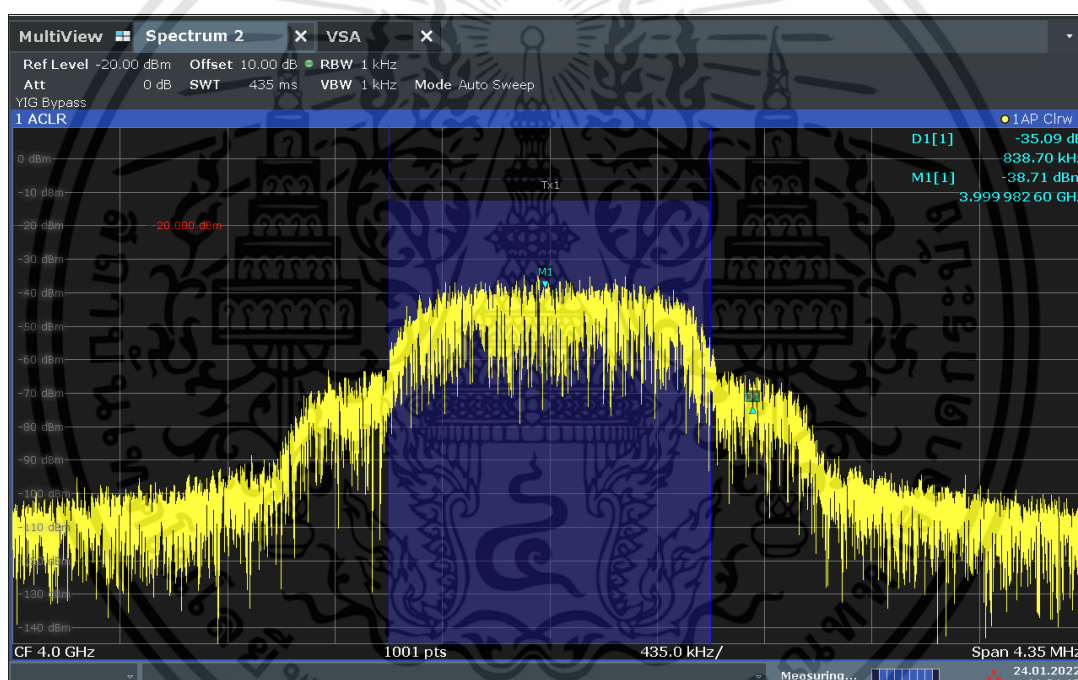


รูปที่ 4.16 หน้าต่าง VSA ของสัญญาณ QPSK ที่ Eb/No เท่ากับ 100 dB

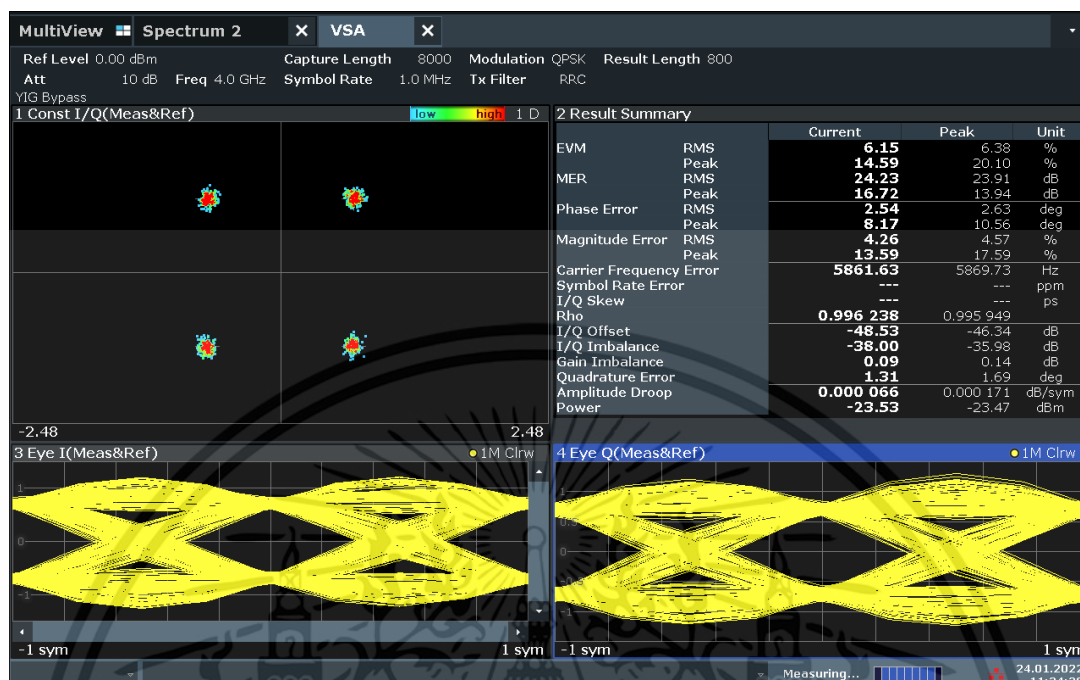
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.15 คือสเปกตรัมของสัญญาณ QPSK ที่ถูกเพิ่มสัญญาณรบกวนโดยตั้งค่า E_b/N_0 ของช่องสัญญาณ AWGN เท่ากับ 100 dB ซึ่งสามารถวัดแบนด์วิธของสเปกตรัมได้เท่ากับ 1.3 GHz และวัดค่า C/N ได้เท่ากับ 47.68 dB

จากรูปที่ 4.16 คือหน้าต่าง VSA ของสัญญาณ QPSK ที่ E_b/N_0 เท่ากับ 100 dB ซึ่งสามารถวัด Symbol Rate ของสัญญาณได้เท่ากับ 1 MHz ซึ่งผลของกราฟ Constellation และ Eye Diagram แสดงให้เห็นได้ว่ามีผลกระทบของสัญญาณรบกวนเล็กน้อย ซึ่งค่า EVM(RMS) เท่ากับ 3.68% และค่า MER(RMS) เท่ากับ 28.68 dB



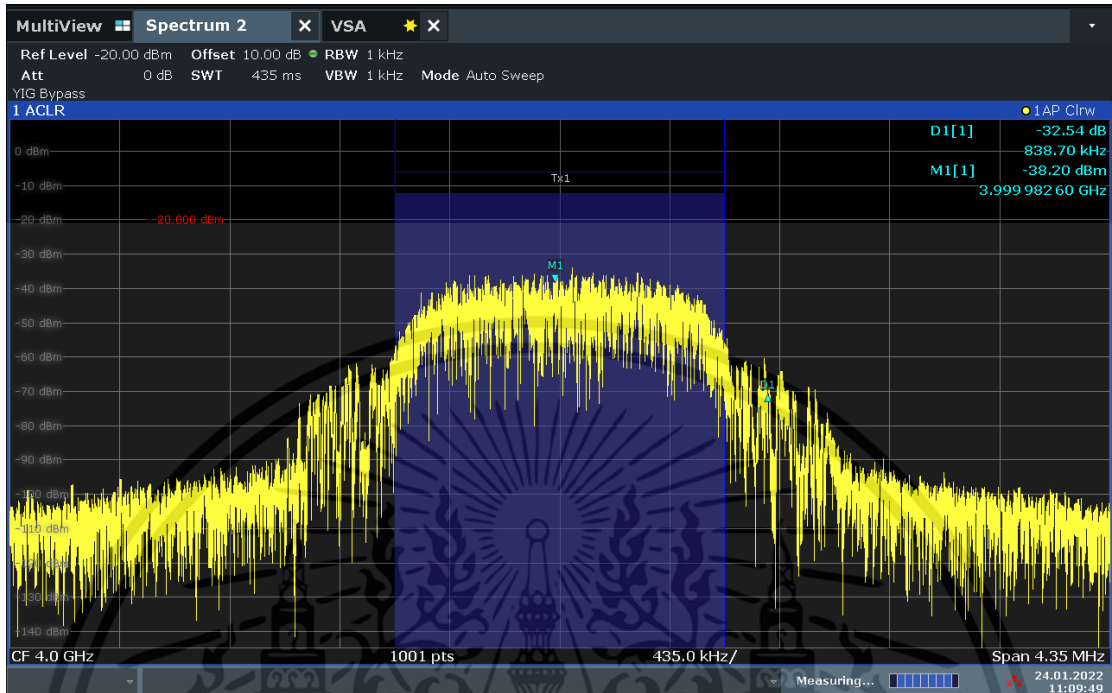
รูปที่ 4.17 สเปกตรัมของสัญญาณ QPSK ที่ E_b/N_0 เท่ากับ 80 dB



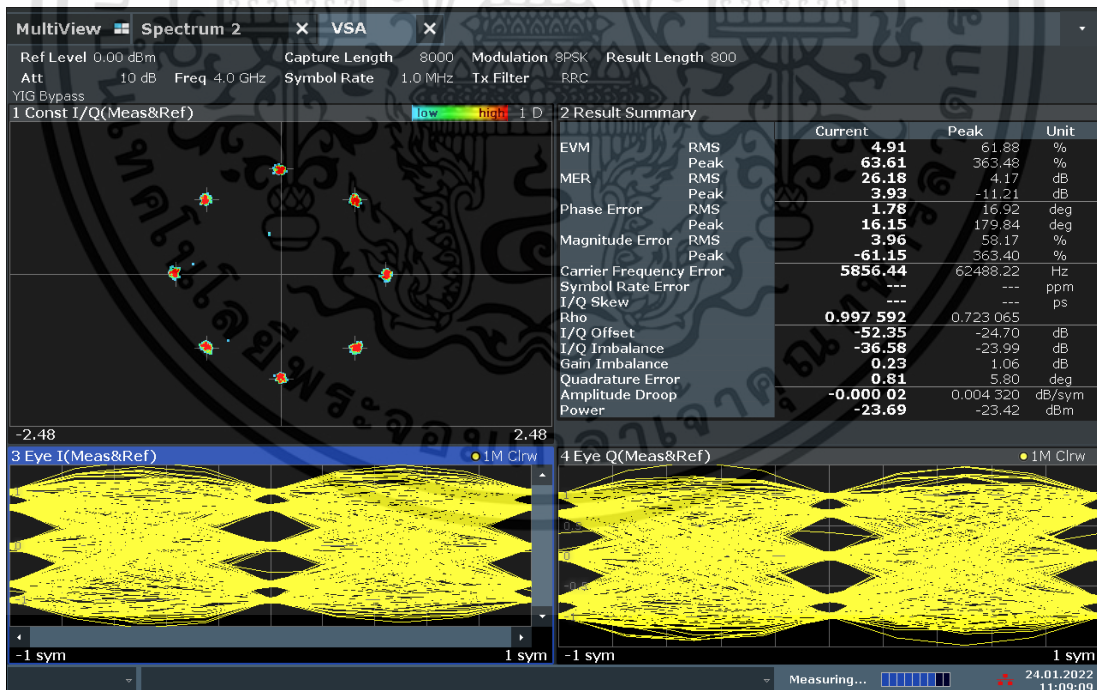
รูปที่ 4.18 หน้าต่าง VSA ของสัญญาณ QPSK ที่ E_b/N_0 เท่ากับ 80 dB

จากรูปที่ 4.17 คือสเปกตรัมของสัญญาณ QPSK ที่ถูกเพิ่มสัญญาณรบกวนโดยตั้งค่า E_b/N_0 ของช่องสัญญาณ AWGN เท่ากับ 80 dB ซึ่งสามารถวัดแบนด์วิธของสเปกตรัมได้เท่ากับ 1.3 GHz แต่เนื่องจากระบบนี้มีกำลังของสัญญาณรบกวนที่มากขึ้น ทำให้เกิดโลบข้าง (Side Lobe) ของสัญญาณรบกวนขึ้นมา ค่า C/N ที่วัดได้จากโลบหลัก (Main Lobe) เท่ากับ 35.09 dB ซึ่ง C/N มีค่าลดลงจากการตั้งค่า E_b/N_0 เท่ากับ 100 dB

จากรูปที่ 4.18 คือหน้าต่าง VSA ของสัญญาณ QPSK ที่ E_b/N_0 เท่ากับ 80 dB ซึ่งสามารถวัด Symbol Rate ของสัญญาณได้เท่ากับ 1 MHz ซึ่งผลของกราฟ Constellation และ Eye Diagram แสดงให้เห็นได้ว่ามีผลกระทบของสัญญาณรบกวนเพิ่มมากขึ้น ซึ่งค่า EVM(RMS) เท่ากับ 6.15% และค่า MER(RMS) เท่ากับ 24.23 dB



รูปที่ 4.19 สเปกตรัมของสัญญาณ 8-PSK ที่ Eb/No เท่ากับ 100 dB



รูปที่ 4.20 หน้าต่าง VSA ของสัญญาณ 8-PSK ที่ Eb/No เท่ากับ 100 dB

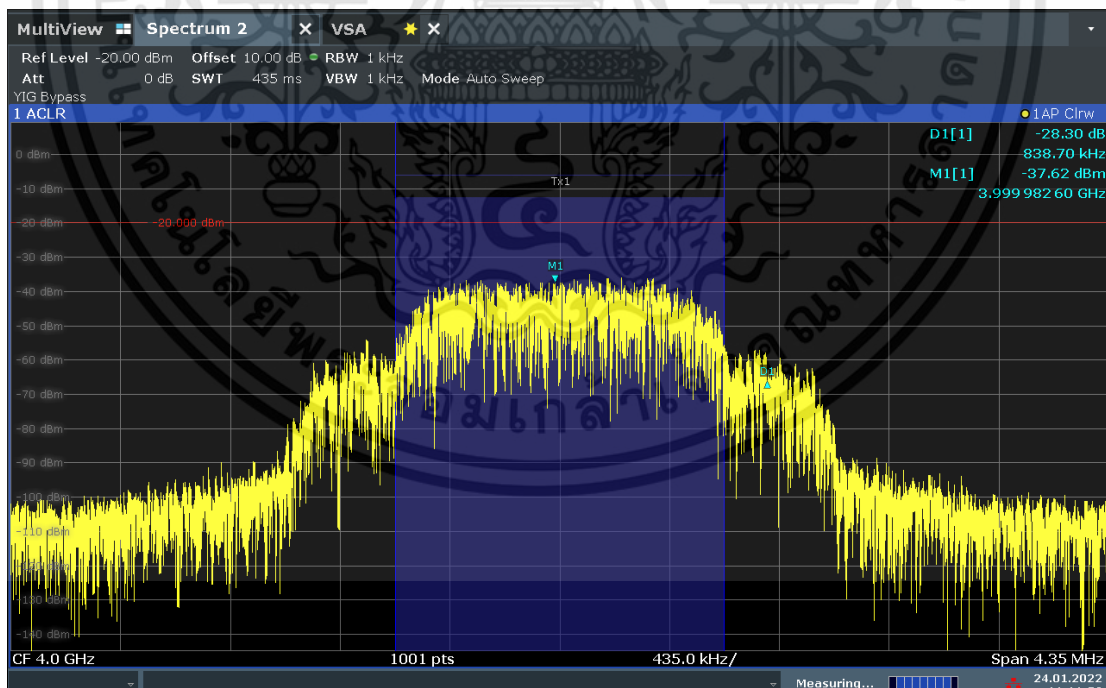
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.19 คือสเปกตรัมของสัญญาณ 8-PSK ที่ถูกเพิ่มสัญญาณรบกวนโดยตั้งค่า Eb/No ของช่องสัญญาณ AWGN เท่ากับ 100 dB ซึ่งสามารถวัดแบนด์วิธของสเปกตรัมได้เท่ากับ 1.3 GHz และวัดค่า C/N ได้เท่ากับ 32.54 dB

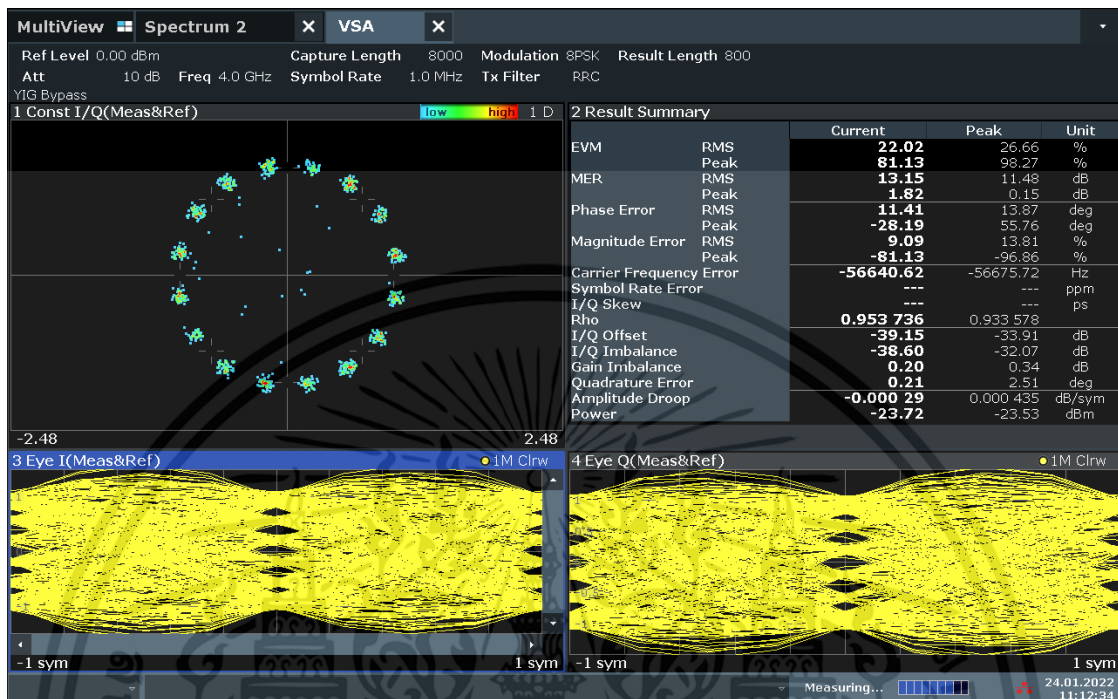
จากรูปที่ 4.20 คือหน้าต่าง VSA ของสัญญาณ 8-PSK ที่ Eb/No เท่ากับ 100 dB ซึ่งสามารถวัด Symbol Rate ของสัญญาณได้เท่ากับ 1 MHz ซึ่งผลของกราฟ Constellation และ Eye Diagram แสดงให้เห็นได้ว่ามีผลกระทบของสัญญาณรบกวนเล็กน้อย ซึ่งค่า EVM(RMS) เท่ากับ 4.91% และค่า MER(RMS) เท่ากับ 26.18 dB

จากรูปที่ 4.21 คือสเปกตรัมของสัญญาณ 16-PSK ที่ถูกเพิ่มสัญญาณรบกวนโดยตั้งค่า Eb/No ของช่องสัญญาณ AWGN เท่ากับ 100 dB ซึ่งสามารถวัดแบนด์วิธของสเปกตรัมได้เท่ากับ 1.3 GHz และวัดค่า C/N ได้เท่ากับ 28.30 dB

และจากรูปที่ 4.22 คือหน้าต่าง VSA ของสัญญาณ 16-PSK ที่ Eb/No เท่ากับ 100 dB ซึ่งสามารถวัด Symbol Rate ของสัญญาณได้เท่ากับ 1 MHz ซึ่งผลของกราฟ Constellation และ Eye Diagram แสดงให้เห็นได้ว่ามีผลกระทบของสัญญาณรบกวนเล็กน้อย ซึ่งค่า EVM(RMS) เท่ากับ 22.02% และค่า MER(RMS) เท่ากับ 13.15 dB



รูปที่ 4.21 สเปกตรัมของสัญญาณ 16-PSK ที่ Eb/No เท่ากับ 100 dB



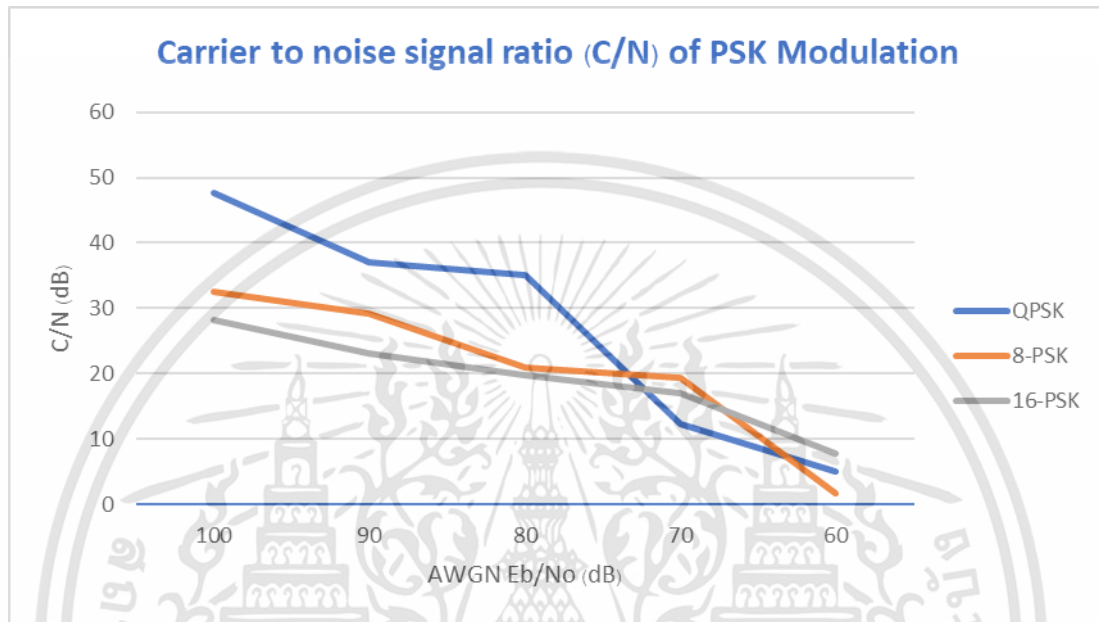
รูปที่ 4.22 หน้าต่าง VSA ของสัญญาณ 16-PSK ที่ Eb/No เท่ากับ 100 dB

ตารางที่ 4.2 ค่า C/N ของระบบสื่อสาร QPSK, 8-PSK และ 16-PSK ด้วยค่า Eb/No ของช่องสัญญาณ AWGN ที่แตกต่างกัน

AWGN Eb/No (dB)	C/N (dB)		
	QPSK	8-PSK	16-PSK
100	47.68	32.54	28.30
90	37.04	29.27	23.12
80	35.09	21.04	19.72
70	12.36	19.46	16.99
60	5.14	1.74	7.72

จากตารางที่ 4.2 สามารถเก็บผลการทดลองได้ดังนี้ โดยค่า Eb/No ของช่องสัญญาณ AWGN ที่ปรับได้ต่ำที่สุดเท่ากับ 60 dB โดยจะเห็นได้ว่าลำดับของการมอดูเลตแบบ PSK ที่มากขึ้น

สัญญาณก็จะถูกรบกวนจากสัญญาณรบกวนได้ง่ายขึ้นเช่นเดียวกัน โดยสามารถสังเกตได้จากค่า C/N ในรูปที่ 4.23



รูปที่ 4.23 กราฟความสัมพันธ์ระหว่าง C/N และ Eb/No (AWGN Channel) ของมอดูเลชันแบบ PSK

4.3 ผลการทดสอบระบบสื่อสาร QPSK ด้วย FPGA

ผลการทดสอบระบบสื่อสาร QPSK ด้วย FPGA ได้แบ่งการเก็บผลการทดสอบเป็น 2 ส่วน ได้แก่ ผลการจำลองด้วยซอฟต์แวร์ Aldec Active-HDL 8.3 และในส่วนของฮาร์ดแวร์โดยวัดสัญญาณดิจิทัลด้วยออสซิลโลสโคป โดยมีการเก็บผลของบล็อกต่างๆ ภายในระบบสื่อสาร QPSK ภาคส่ง และนำผลที่วัดได้ทั้งจากการจำลองด้วยซอฟต์แวร์ และผลจากบอร์ด FPGA มาเปรียบเทียบกับผลการจำลองจากซอฟต์แวร์ MATLAB ซึ่งมีขั้นตอนดังต่อไปนี้

4.3.1 ผลของวงจร Scrambler

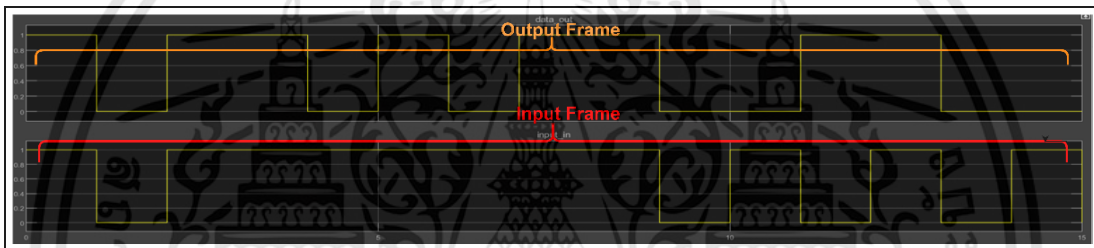
จากการออกแบบวงจร Scrambler ด้วยภาษา VHDL ได้ทำการเปรียบเทียบผลกับซอฟต์แวร์ MATLAB โดยป้อนอินพุตเดียวกันเข้าไปในระบบ

4.3.1.1 ผลการจำลองของ Scrambler ด้วย MATLAB

ข้อมูลอินพุตที่ป้อนให้กับ Scrambler ในซอฟต์แวร์ MATLAB สามารถดูช่วงของข้อมูลได้จาก Input Frame และสัญญาณเอาต์พุตที่ได้จาก Output Frame ดังรูปที่ 4.24 โดยตารางที่ 4.3 แสดงข้อมูลไบนารีที่อ่านได้จากการทำงานของบล็อก Scrambler นี้

ตารางที่ 4.3 ผลของ Scrambler ด้วยซอฟต์แวร์ MATLAB

Input Frame (data_in)	[0 1 1 1 1 1 1 1 0 1 0 1 0 1 1]
Output Frame (data_out)	[1 0 1 1 0 1 0 1 1 0 0 1 1 0 0]



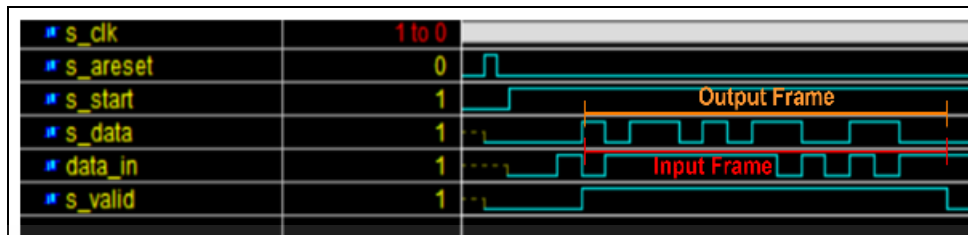
รูปที่ 4.24 ผลจากการวัดสัญญาณของ Scrambler ด้วยซอฟต์แวร์ MATLAB

4.3.3.2 ผลการจำลองของ Scrambler ด้วยภาษา VHDL

ผลจากการจำลองการทำงานของโค้ด VHDL ที่ได้ออกแบบไว้ของวงจร สแกนเบลอร์ ด้วยซอฟต์แวร์ Aldec Active-HDL 8.3 โดยมีสัญญาณอินพุตคือ สัญญาณ data_in และสัญญาณเอาต์พุตคือ สัญญาณ s_data โดยสามารถดูช่วงของข้อมูลได้จาก Input Frame และสัญญาณเอาต์พุตที่ได้จาก Output Frame ดังรูปที่ 4.25 โดยตารางที่ 4.4 แสดงข้อมูลไบนารีที่อ่านได้จากการทำงานของวงจร Scrambler นี้

ตารางที่ 4.4 ผลของวงจร Scrambler ด้วยซอฟต์แวร์ Aldec Active-HDL 8.3

Input Frame (data_in)	[0 1 1 1 1 1 1 1 0 1 0 1 0 1 1]
Output Frame (s_data)	[1 0 1 1 0 1 0 1 1 0 0 1 1 0 0]



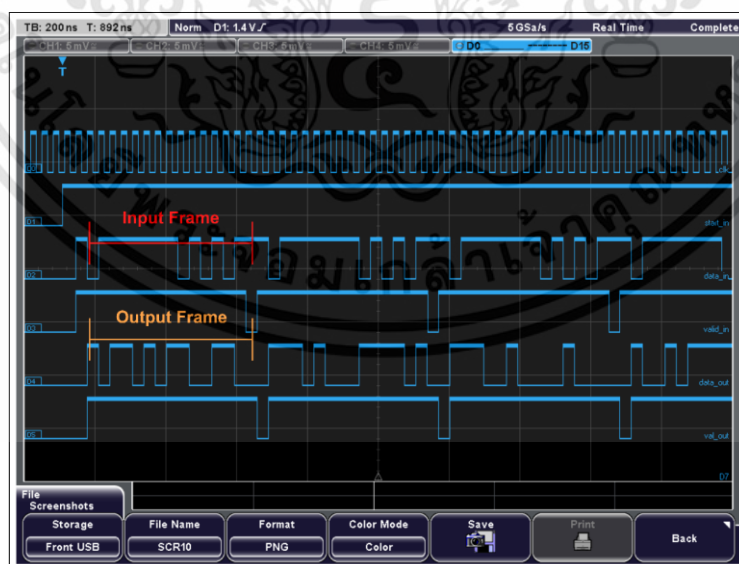
รูปที่ 4.25 ผลการจำลองการทำงานของวงจร Scrambler บนซอฟต์แวร์ Aldec Active-HDL

4.3.3.3 ผลของวงจร Scrambler จากบอร์ด FPGA

วงจร Scrambler ที่ได้ออกแบบด้วยภาษา VHDL จะถูกนำมาสังเคราะห์ลงบอร์ด FPGA (Zybo Z7-20) ด้วยซอฟต์แวร์ Vivado 2019.2 โดยมีสัญญาณอินพุตคือ สัญญาณ data_in และสัญญาณเอาต์พุตคือ สัญญาณ s_data โดยสามารถดูช่วงของข้อมูลได้จาก Input Frame และสัญญาณเอาต์พุตที่ได้จาก Output Frame ดังรูปที่ 4.26 โดยตารางที่ 4.5 แสดงข้อมูลไบนารีที่อ่านได้จากการทำงานของวงจร Scrambler นี้

ตารางที่ 4.5 ผลของวงจร Scrambler จากบอร์ด FPGA

Input Frame (data_in)	[0 1 1 1 1 1 1 0 1 0 1 0 1 1]
Output Frame (data_out)	[1 0 1 1 0 1 0 1 1 0 0 1 1 0 0]



รูปที่ 4.26 ผลของวงจร Scrambler จากบอร์ด FPGA วัดด้วยออสซิลโลสโคป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 ผลของวงจร Polar NRZ-L

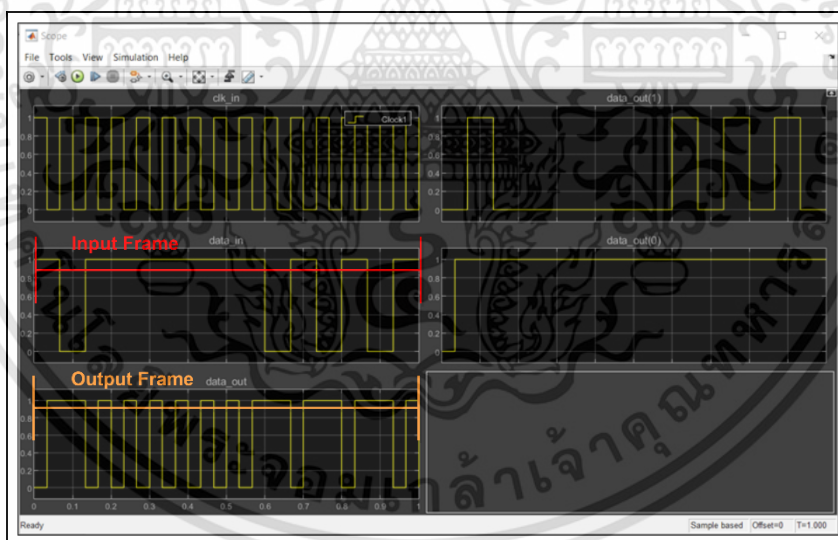
จากการออกแบบวงจร Polar NRZ-L ด้วยภาษา VHDL ได้ทำการเปรียบเทียบผลกับซอฟต์แวร์ MATLAB โดยป้อนอินพุตเดียวกันเข้าไปในระบบ

4.3.2.1 ผลการจำลองของ Polar NRZ-L ด้วย MATLAB

ข้อมูลอินพุตที่ป้อนให้กับ Polar NRZ-L ในซอฟต์แวร์ MATLAB สามารถดูช่วงของข้อมูลได้จาก Input Frame และสัญญาณเอาต์พุตที่ได้จาก Output Frame ดังรูปที่ 4.27 โดยตารางที่ 4.6 แสดงข้อมูลไบนารีที่อ่านได้จากการทำงานของบล็อก Polar NRZ-L นี้

ตารางที่ 4.6 ผลของ Polar NRZ-L ด้วยซอฟต์แวร์ MATLAB

Input Frame (data_in)	[1 0 1 1 1 1 1 1 0 1 0 1 0 1]
Output Frame (data_out)	[01 11 01 01 01 01 01 01 01 11 01 11 01 11 01]



รูปที่ 4.27 ผลจากการวัดสัญญาณของ Polar NRZ-L ด้วยซอฟต์แวร์ MATLAB

4.3.2.2 ผลการจำลองของ Polar NRZ-L ด้วยภาษา VHDL

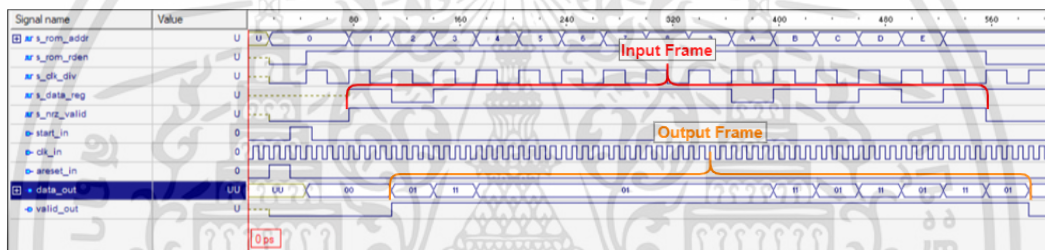
ผลจากการจำลองการทำงานของโค้ด VHDL ที่ได้ออกแบบไว้ของวงจร สแกรมเบลอร์ ด้วยซอฟต์แวร์ Aldec Active-HDL 8.3 โดยมีสัญญาณอินพุตคือ สัญญาณ data_in และ

สัญญาณเอาต์พุตคือ สัญญาณ s_data โดยสามารถดูช่วงของข้อมูลได้จาก Input Frame และสัญญาณเอาต์พุตที่ได้จาก Output Frame ดังรูปที่ 4.28 โดยตารางที่ 4.7 แสดงข้อมูลไบนารีที่อ่านได้จากการทำงานของวงจรร Polar NRZ-L นี้

ตารางที่ 4.7 ผลของวงจรร Polar NRZ-L ด้วยซอฟต์แวร์ Aldec Active-HDL

8.3

Input Frame (s_data_reg)	[1 0 1 1 1 1 1 1 1 0 1 0 1 0 1]
Output Frame (data_out)	[01 11 01 01 01 01 01 01 01 11 01 11 01 11 01 11 01]



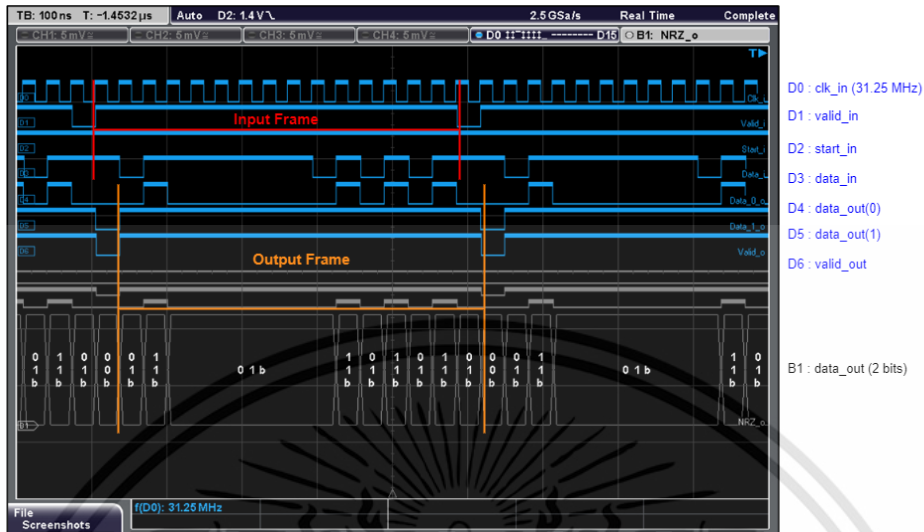
รูปที่ 4.28 ผลการจำลองการทำงานของวงจรร Polar NRZ-L บนซอฟต์แวร์ Aldec Active-HDL

4.3.2.3 ผลของวงจรร Polar NRZ-L จากบอร์ด FPGA

วงจรร Polar NRZ-L ที่ได้ออกแบบด้วยภาษา VHDL จะถูกนำมาสังเคราะห์ลงบอร์ด FPGA (Zybo Z7-20) ด้วยซอฟต์แวร์ Vivado 2019.2 โดยมีสัญญาณอินพุตคือ สัญญาณ data_in และสัญญาณเอาต์พุตคือ สัญญาณ s_data โดยสามารถดูช่วงของข้อมูลได้จาก Input Frame และสัญญาณเอาต์พุตที่ได้จาก Output Frame ดังรูปที่ 4.29 โดยตารางที่ 4.8 แสดงข้อมูลไบนารีที่อ่านได้จากการทำงานของวงจรร Polar NRZ-L นี้

ตารางที่ 4.8 ผลของวงจรร Polar NRZ-L จากบอร์ด FPGA

Input Frame (data_in)	[1 0 1 1 1 1 1 1 1 0 1 0 1 0 1]
Output Frame (data_out)	[01 11 01 01 01 01 01 01 01 11 01 11 01 11 01 11 01]



รูปที่ 4.29 ผลของวงจร Polar NRZ-L จากบอร์ด FPGA วัดด้วยออสซิลโลสโคป

4.3.1 ผลของวงจร I/Q Separate

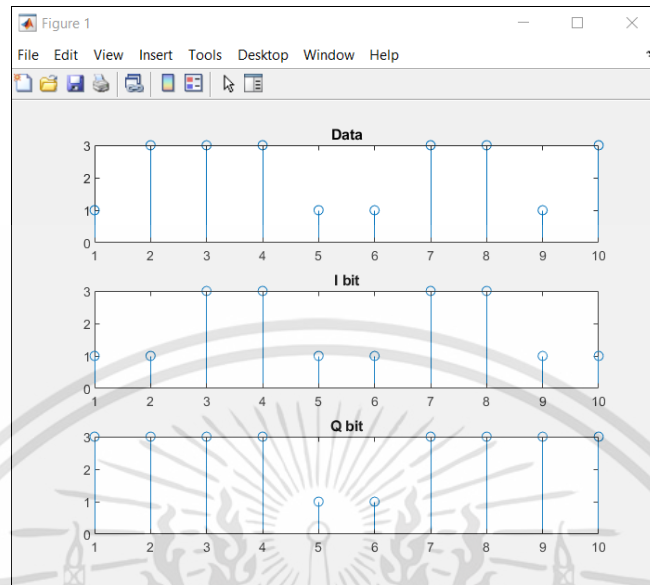
จากการออกแบบวงจร I/Q Separate ด้วยภาษา VHDL ได้ทำการเปรียบเทียบผลกับซอฟต์แวร์ MATLAB โดยป้อนอินพุตเดียวกันเข้าไปในระบบ

4.3.1.1 ผลการจำลองของ I/Q Separate ด้วย MATLAB

ข้อมูลอินพุตที่ป้อนให้กับ I/Q Separate ในซอฟต์แวร์ MATLAB สามารถดูช่วงของข้อมูลได้จาก Input Frame และสัญญาณเอาต์พุตที่ได้จาก Output Frame ดังรูปที่ 4.30 โดยตารางที่ 4.9 แสดงข้อมูลเลขฐาน 16 ที่อ่านได้จากการทำงานของบล็อก I/Q Separate นี้

ตารางที่ 4.9 ผลของ I/Q Separate ด้วยซอฟต์แวร์ MATLAB

Input Frame (data_in)	[1 3 3 3 1 1 3 3 1 3]
Output Frame (data_out)	[1 3 1 3 1]
Output Frame (data_out)	[3 3 1 3 3]



รูปที่ 4.30 ผลจากการวัดสัญญาณของ I/Q Separate ด้วยซอฟต์แวร์ MATLAB

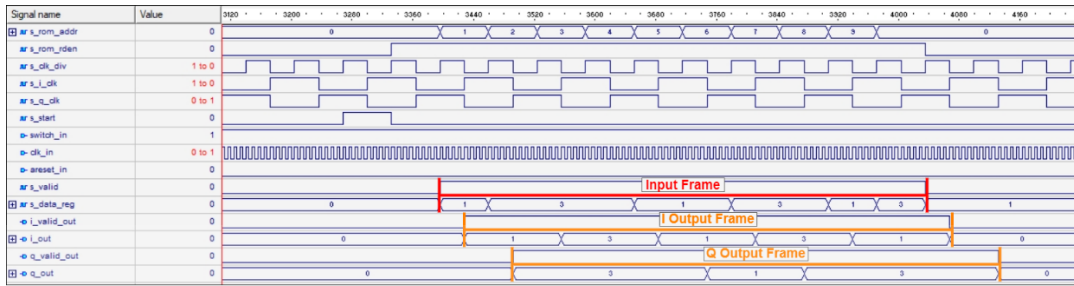
4.3.3.2 ผลการจำลองของ I/Q Separate ด้วยภาษา VHDL

ผลจากการจำลองการทำงานของโค้ด VHDL ที่ได้ออกแบบไว้ของวงจร สแกนเบลอร์ ด้วยซอฟต์แวร์ Aldec Active-HDL 8.3 โดยมีสัญญาณอินพุตคือ สัญญาณ s_data_reg และสัญญาณเอาต์พุตคือ สัญญาณ i_out และ q_out โดยสามารถดูช่วงของข้อมูลได้จาก Input Frame และสัญญาณเอาต์พุตที่ได้จาก Output Frame ดังรูปที่ 4.31 โดยตารางที่ 4.10 แสดงข้อมูลเลขฐาน 16 ที่อ่านได้จากการทำงานของวงจร I/Q Separate นี้

ตารางที่ 4.10 ผลของวงจร I/Q Separate ด้วยซอฟต์แวร์ Aldec Active-HDL

8.3

Input Frame (s_data_reg)	[1 3 3 3 1 1 3 3 1 3]
Output Frame (i_out)	[1 3 1 3 1]
Output Frame (q_out)	[3 3 1 3 3]



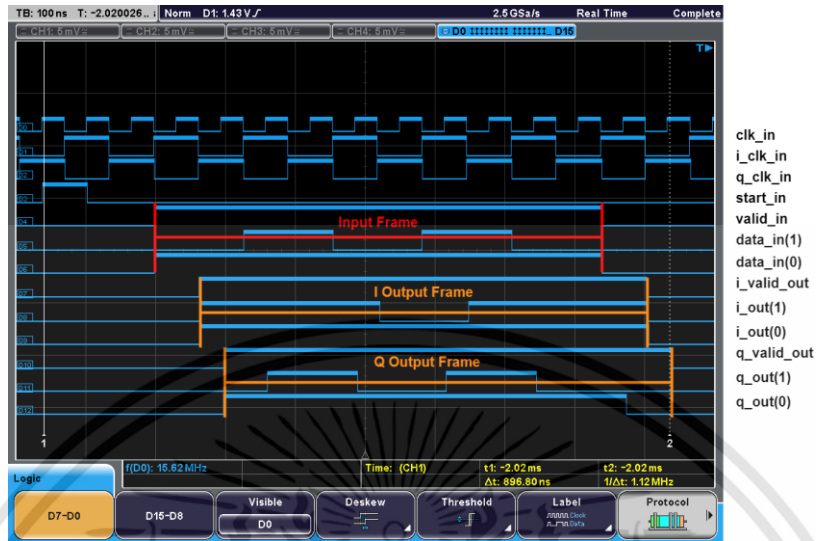
รูปที่ 4.31 ผลการจำลองการทำงานของวงจร I/Q Separate บนซอฟต์แวร์ Aldec Active-HDL

4.3.3.3 ผลของวงจร I/Q Separate จากบอร์ด FPGA

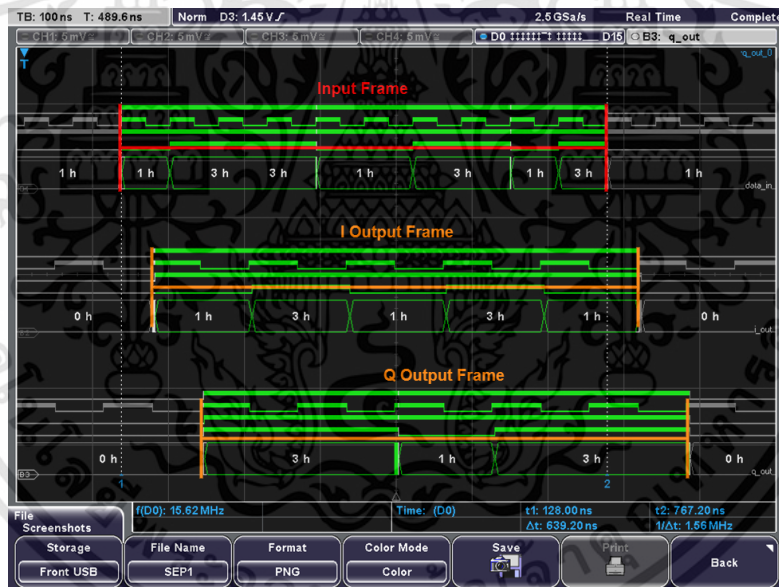
วงจร I/Q Separate ที่ได้ออกแบบด้วยภาษา VHDL จะถูกนำมาสังเคราะห์ลงบอร์ด FPGA (Zybo Z7-20) ด้วยซอฟต์แวร์ Vivado 2019.2 โดยมีสัญญาณอินพุตคือ สัญญาณ data_in และสัญญาณเอาต์พุตคือ สัญญาณ i_out และ q_out โดยสามารถดูช่วงของข้อมูลได้จาก Input Frame และสัญญาณเอาต์พุตที่ได้จาก Output Frame ดังรูปที่ 4.32 และ 4.33 โดยตารางที่ 4.11 แสดงข้อมูลเลขฐาน 16 ที่อ่านได้จากการทำงานของวงจร I/Q Separate นี้

ตารางที่ 4.11 ผลของวงจร I/Q Separate จากบอร์ด FPGA

Input Frame (data_in)	[1 3 3 3 1 1 3 3 1 3]
Output Frame (i_out)	[1 3 1 3 1]
Output Frame (q_out)	[3 3 1 3 3]



รูปที่ 4.32 ผลของวงจร I/Q Separate จากบอร์ด FPGA วัดด้วยออสซิลโลสโคป



รูปที่ 4.33 ถอดรหัสสัญญาณของวงจร I/Q Separate บนบอร์ด FPGA ด้วยออสซิลโลสโคป

4.3.4 ผลของวงจร I/Q Pairing

จากการออกแบบวงจร I/Q Pairing ด้วยภาษา VHDL ได้ทำการเปรียบเทียบผลกับซอฟต์แวร์ MATLAB โดยป้อนอินพุตเดียวกันเข้าไปในระบบ

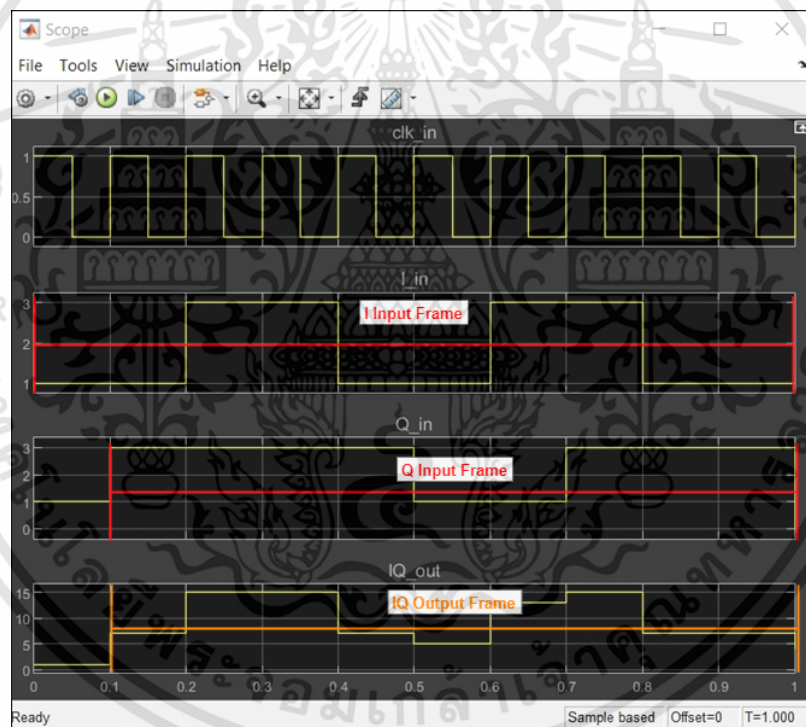
4.3.4.1 ผลการจำลองของ I/Q Pairing ด้วย MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลอินพุตที่ป้อนให้กับ I/Q Pairing ในซอฟต์แวร์ MATLAB สามารถดูช่วงของข้อมูลได้จาก Input Frame และสัญญาณเอาต์พุตที่ได้จาก Output Frame ดังรูปที่ 4.34 โดยตารางที่ 4.12 แสดงข้อมูลเลขฐาน 16 ที่อ่านได้จากการทำงานของบล็อก I/Q Pairing นี้

ตารางที่ 4.12 ผลของ I/Q Pairing ด้วยซอฟต์แวร์ MATLAB

Input Frame (I_in)	[1 1 3 3 1 1 3 3 1]
Input Frame (Q_in)	[3 3 3 3 1 1 3 3 3]
Output Frame (IQ_out)	[7 F F 7 5 D F 7 7]



รูปที่ 4.34 ผลจากการวัดสัญญาณของ I/Q Pairing ด้วยซอฟต์แวร์ MATLAB

4.3.4.2 ผลการจำลองของ I/Q Pairing ด้วยภาษา VHDL

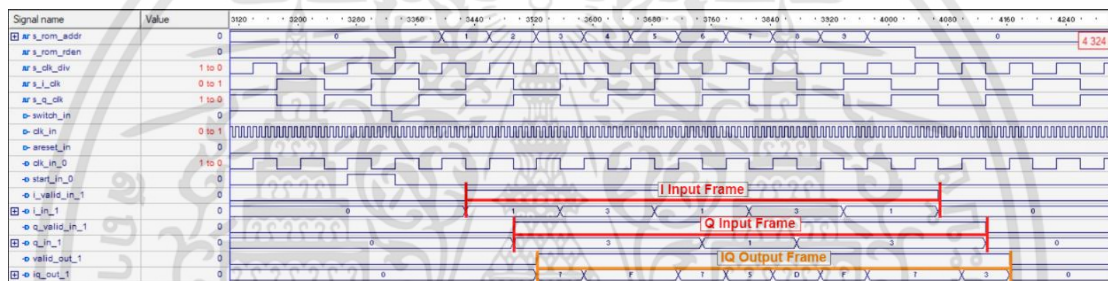
ผลจากการจำลองการทำงานของโค้ด VHDL ที่ได้ออกแบบไว้ของวงจร สแกรมเบลอร์ ด้วยซอฟต์แวร์ Aldec Active-HDL 8.3 โดยมีสัญญาณอินพุตคือ สัญญาณ i_in_1 และ q_in_1 และสัญญาณเอาต์พุตคือ สัญญาณ iq_out_1 โดยสามารถดูช่วงของข้อมูลได้จาก Input

Frame และสัญญาณเอาต์พุตที่ได้จาก Output Frame ดังรูปที่ 4.35 โดยตารางที่ 4.13 แสดงข้อมูลเลขฐาน 16 ที่อ่านได้จากการทำงานของวงจรร I/Q Pairing นี้

ตารางที่ 4.13 ผลของวงจรร I/Q Pairing ด้วยซอฟต์แวร์ Aldec Active-HDL

8.3

Input Frame (i_in_1)	[1 1 3 3 1 1 3 3 1]
Input Frame (q_in_1)	[3 3 3 3 1 1 3 3 3]
Output Frame (iq_out_1)	[7 F F 7 5 D F 7 7]



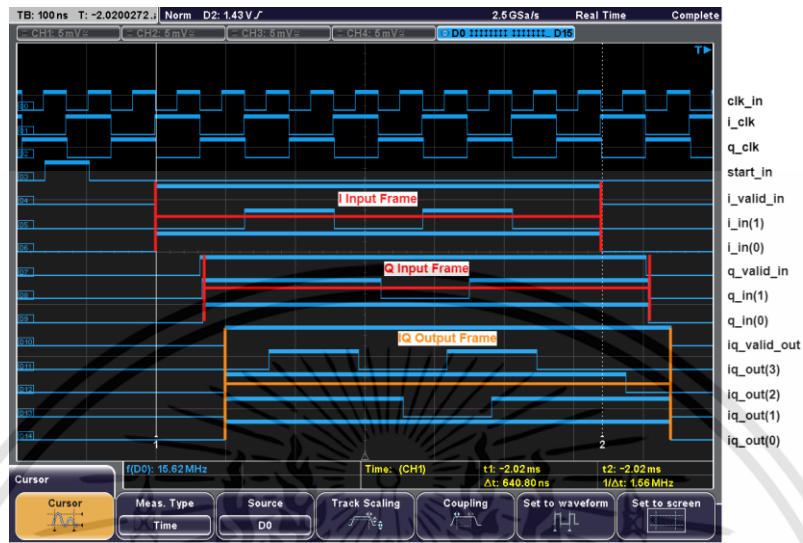
รูปที่ 4.35 ผลการจำลองการทำงานของวงจรร I/Q Pairing บนซอฟต์แวร์ Aldec Active-HDL

4.3.4.3 ผลของวงจรร I/Q Pairing จากบอร์ด FPGA

วงจรร I/Q Pairing ที่ได้ออกแบบด้วยภาษา VHDL จะถูกนำมาสังเคราะห์ลงบอร์ด FPGA (Zybo Z7-20) ด้วยซอฟต์แวร์ Vivado 2019.2 โดยมีสัญญาณอินพุตคือ สัญญาณ i_out และ q_out และสัญญาณเอาต์พุตคือ สัญญาณ iq_out โดยสามารถดูช่วงของข้อมูลได้จาก Input Frame และสัญญาณเอาต์พุตที่ได้จาก Output Frame ดังรูปที่ 4.36 และ 4.37 โดยตารางที่ 4.14 แสดงข้อมูลเลขฐาน 16 ที่อ่านได้จากการทำงานของวงจรร I/Q Pairing นี้

ตารางที่ 4.14 ผลของวงจรร I/Q Pairing จากบอร์ด FPGA

Input Frame (i_in)	[1 1 3 3 1 1 3 3 1]
Input Frame (q_in)	[3 3 3 3 1 1 3 3 3]
Output Frame (iq_out)	[7 F F 7 5 D F 7 7]



รูปที่ 4.36 ผลของวงจร I/Q Pairing จากบอร์ด FPGA วัดด้วยออสซิลโลสโคป



รูปที่ 4.37 ถอดรหัสสัญญาณของวงจร I/Q Pairing บนบอร์ด FPGA ด้วยออสซิลโลสโคป

4.3.5 ผลของวงจร QPSK Mapping

จากการออกแบบวงจร QPSK Mapping ด้วยภาษา VHDL ได้ทำการเปรียบเทียบ ผลกับซอฟต์แวร์ MATLAB โดยป้อนอินพุตเดียวกันเข้าไปในระบบ

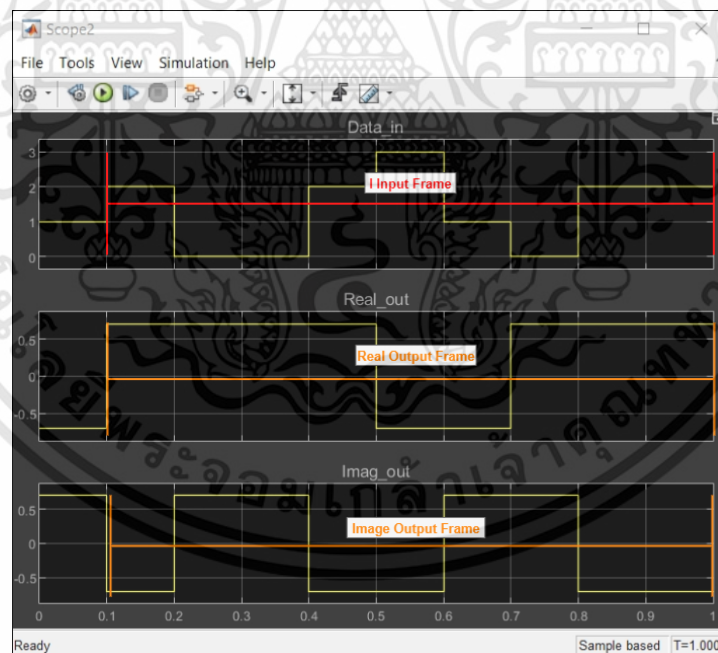
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.5.1 ผลการจำลองของ QPSK Mapping ด้วย MATLAB

ข้อมูลอินพุตที่ป้อนให้กับ QPSK Mapping ในซอฟต์แวร์ MATLAB สามารถดูช่วงของข้อมูลได้จาก Input Frame และสัญญาณเอาต์พุตที่ได้จาก Output Frame ดังรูปที่ 4.38 โดยตารางที่ 4.15 แสดงข้อมูล sign fixed-point ที่อ่านได้จากการทำงานของบล็อก QPSK Mapping นี้

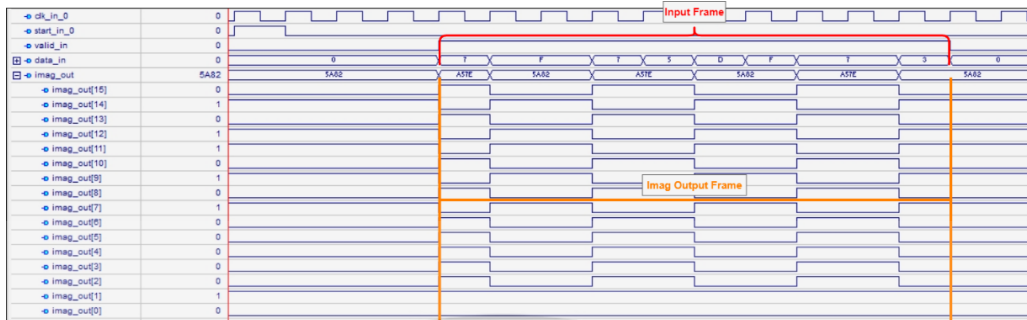
ตารางที่ 4.15 ผลของ QPSK Mapping ด้วยซอฟต์แวร์ MATLAB

Input Frame (Data_in)	[7 F F 7 5 D F 7 7]
Input Frame (Real_out)	[0.7071 0.7071 0.7071 0.7071 -0.7071 -0.7071 0.7071 0.7071 0.7071]
Output Frame (Imag_out)	[-0.7071 0.7071 0.7071 -0.7071 -0.7071 0.7071 0.7071 -0.7071 -0.7071]



รูปที่ 4.38 ผลจากการวัดสัญญาณของ QPSK Mapping ด้วยซอฟต์แวร์ MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



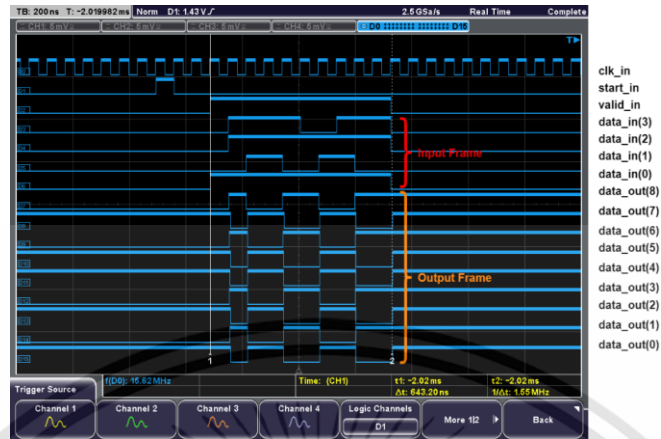
รูปที่ 4.41 ข้อมูลเอาต์พุต (Image) จากการจำลองการทำงานของวงจร QPSK Mapping บนซอฟต์แวร์ Aldec Active-HDL

4.3.5.3 ผลของวงจร QPSK Mapping จากบอร์ด FPGA

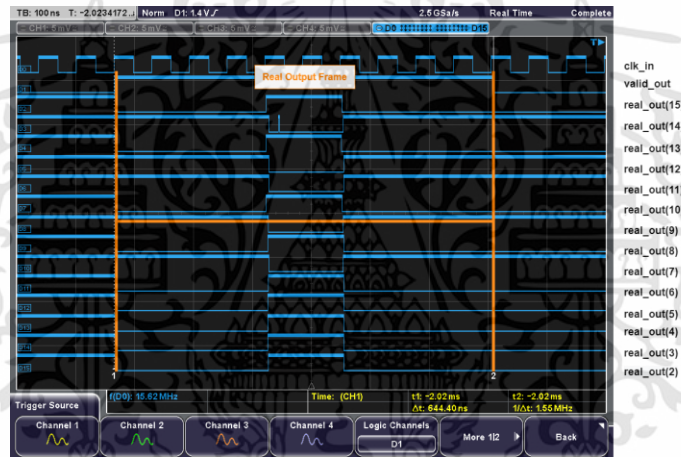
วงจร QPSK Mapping ที่ได้ออกแบบด้วยภาษา VHDL จะถูกนำมาสังเคราะห์ลงบอร์ด FPGA (Zybo Z7-20) ด้วยซอฟต์แวร์ Vivado 2019.2 โดยมีสัญญาณอินพุตคือสัญญาณ data_in และสัญญาณเอาต์พุตคือ สัญญาณ s_data โดยสามารถดูช่วงของข้อมูลได้จาก Input Frame และสัญญาณเอาต์พุตที่ได้จาก Output Frame ดังรูปที่ 4.42-4.44 โดยตารางที่ 4.17 แสดงข้อมูล sign fixed-point ที่อ่านได้จากการทำงานของวงจร QPSK Mapping นี้

ตารางที่ 4.17 ผลของวงจร QPSK Mapping จากบอร์ด FPGA

Input Frame (data_in)	[7 F F 7 5 D F 7 7]
Input Frame (real_out)	[0.7071 0.7071 0.7071 0.7071 -0.7071 -0.7071 0.7071 0.7071 0.7071]
Output Frame (imag_out)	[-0.7071 0.7071 0.7071 -0.7071 -0.7071 0.7071 0.7071 -0.7071 -0.7071]



รูปที่ 4.42 ผลของวงจร QPSK Mapping จากบอร์ด FPGA วัดด้วยออสซิลโลสโคป



รูปที่ 4.43 สัญญาณ Real ของวงจร QPSK Mapping จากบอร์ด FPGA ที่วัดโดยออสซิลโลสโคป



รูปที่ 4.44 สัญญาณ Image ของวงจร QPSK Mapping จากบอร์ด FPGA ที่วัดโดยออสซิลโลสโคป

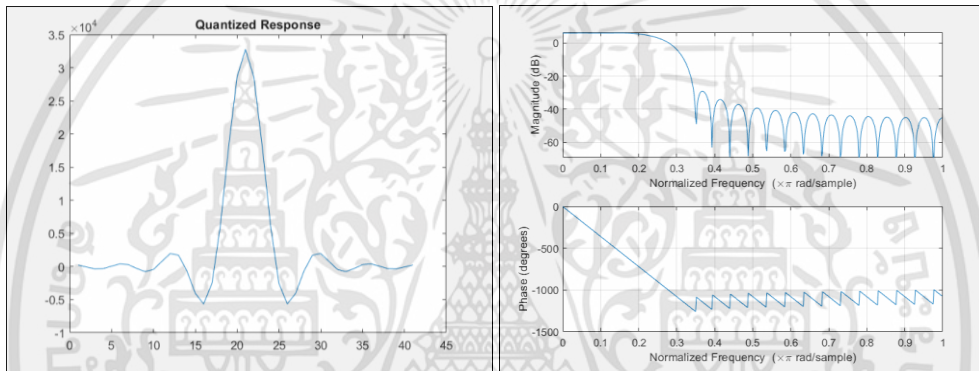
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.6 ผลของวงจร Tx RRC Filter

จากการออกแบบวงจร RRC Filter ด้วยภาษา VHDL ได้ทำการเปรียบเทียบผลกับซอฟต์แวร์ MATLAB โดยป้อนอินพุตเดียวกันเข้าไปในระบบ

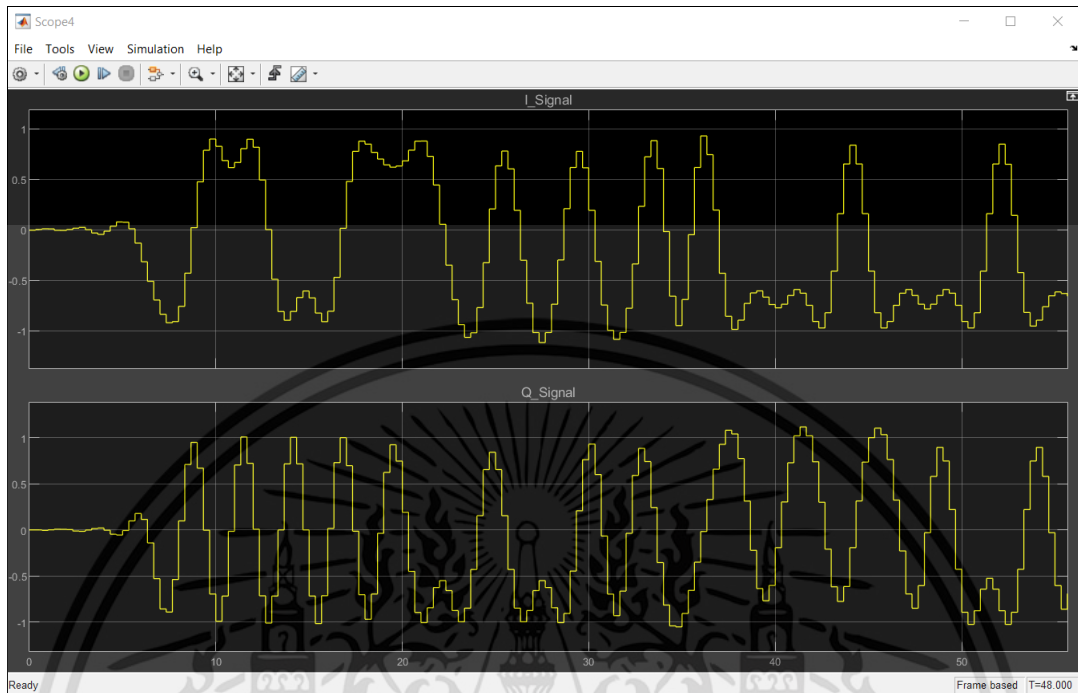
4.3.6.1 ผลการจำลองของ RRC Filter ด้วย MATLAB

ข้อมูลอินพุตที่ป้อนให้กับ RRC Filter ในซอฟต์แวร์ MATLAB โดยสร้างฟิลเตอร์ด้วยค่า Roll off = 0.35 โดย interpolate = 4 ซึ่งจะได้ฟิลเตอร์ 41 taps ขึ้นมา เก็บผลโดยการป้อนสัญญาณอินพุตเป็นอิมพัลส์เข้าไป และพล็อตกราฟเพื่อดูผลตอบสนองอิมพัลส์ของระบบ ดังรูปที่ 4.45



รูปที่ 4.45 ผลตอบสนองอิมพัลส์ของ RRC Filter ด้วยซอฟต์แวร์ MATLAB

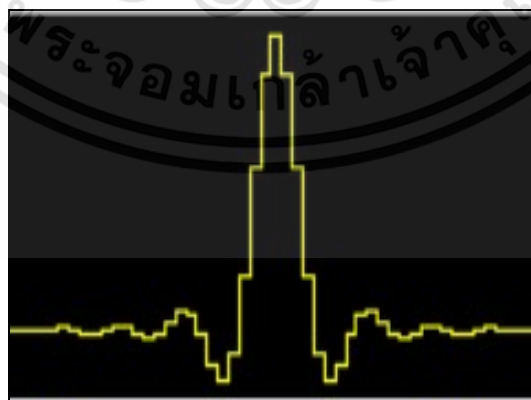
และอีกกรณีเป็นการทดสอบระบบการสื่อสาร QPSK ภาคส่ง โดยป้อนข้อมูลอินพุตเป็นสตรีมเข้าไปในระบบเพื่อดูผลของสัญญาณเอาต์พุตหลังผ่าน RRC Filter โดยมีสัญญาณเอาต์พุต 2 สัญญาณ ได้แก่ สัญญาณ In-phase (I_Signal) และสัญญาณ Quadrature (Q_Signal) ดังรูปที่ 4.46



รูปที่ 4.46 ผลจากการวัดสัญญาณของ RRC Filter ด้วยซอฟต์แวร์ MATLAB

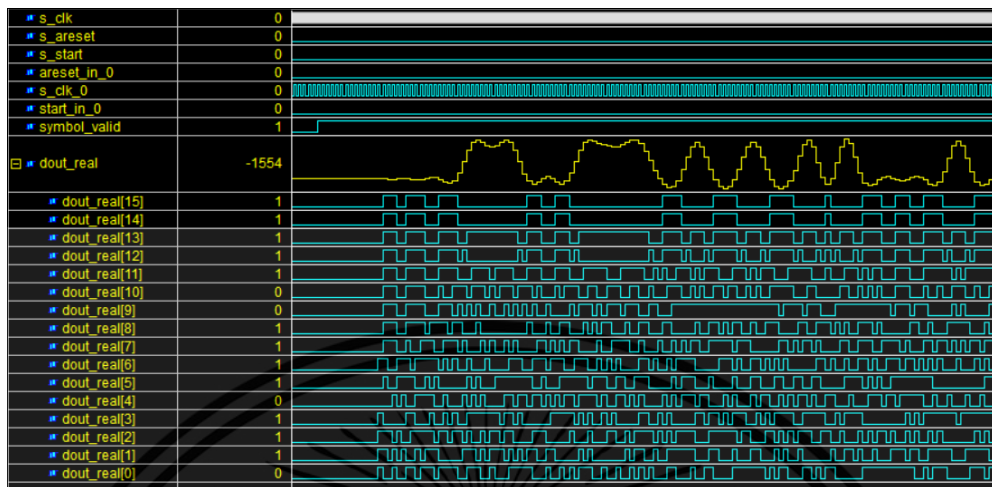
4.3.6.2 ผลการจำลองของ RRC Filter ด้วยภาษา VHDL

จากการจำลองผลของวงจร RRC Filter ด้วยภาษา VHDL บนซอฟต์แวร์ Aldec Active-HDL 8.3 ได้ผลลัพธ์จากการป้อนอินพุตเป็นสัญญาณอิมพัลส์ดังรูปที่ 4.47 และผลลัพธ์จากการทดสอบระบบสื่อสาร QPSK ภาคส่ง ด้วยการป้อนอินพุตเป็นข้อมูลต่อเนื่องดังรูปที่ 4.48 และ 4.49

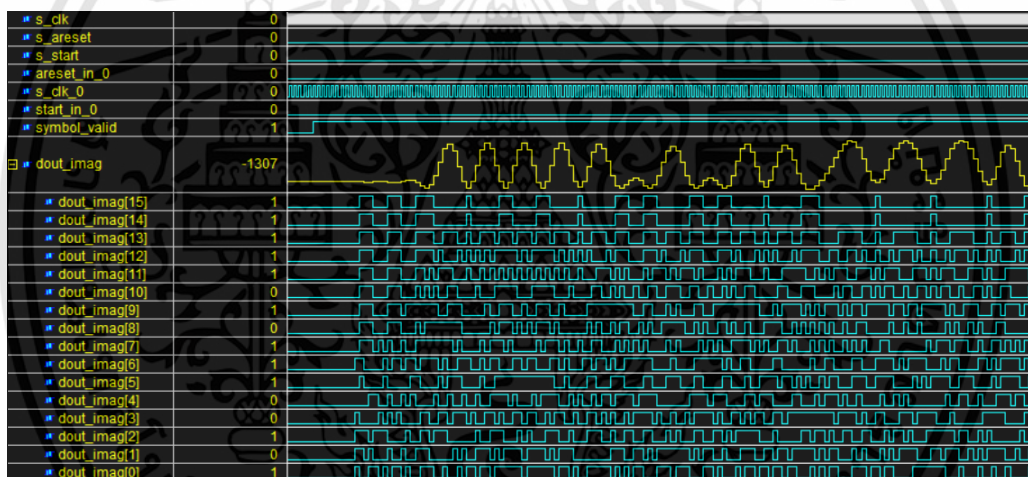


รูปที่ 4.47 ผลตอบสนองอิมพัลส์ของ RRC Filter ด้วยซอฟต์แวร์ Aldec Active-HDL 8.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



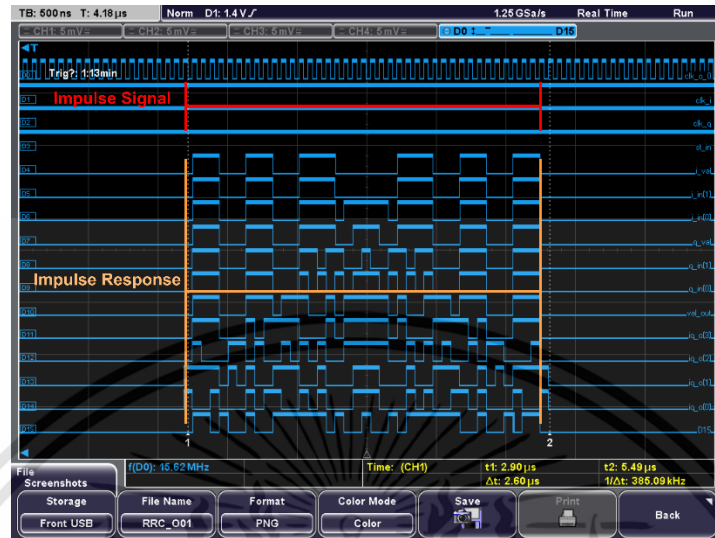
รูปที่ 4.48 ผลของสัญญาณ In-phase จาก RRC Filter ด้วยซอฟต์แวร์ Aldec Active-HDL 8.3



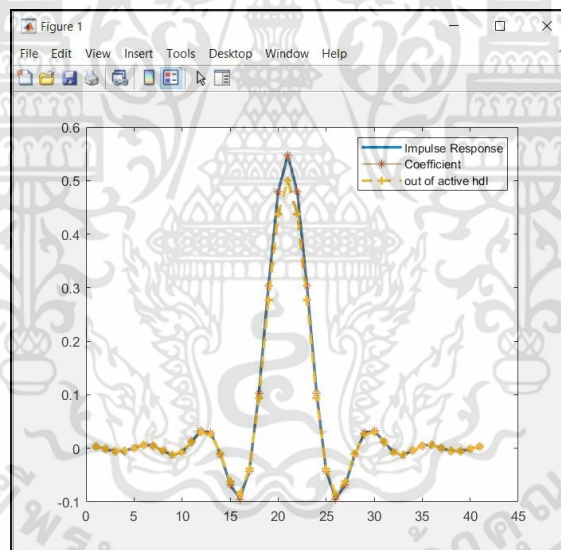
รูปที่ 4.49 ผลของสัญญาณ Quadrature จาก RRC Filter ด้วยซอฟต์แวร์ Aldec Active-HDL 8.3

4.3.6.3 ผลของวงจร RRC Filter จากบอร์ด FPGA

วงจร RRC Filter ที่ได้ออกแบบด้วยภาษา VHDL จะถูกนำมาสังเคราะห์ลงบอร์ด FPGA (Zybo Z7-20) ด้วยซอฟต์แวร์ Vivado 2019.2 โดยมีผลลัพธ์จากการบ้อนอินพุตเป็นสัญญาณอิมพัลส์ดังรูปที่ 4.50



รูปที่ 4.50 ผลตอบสนองอิมพัลส์ของวงจร RRC Filter จากบอร์ด FPGA วัดด้วยออสซิลโลสโคป

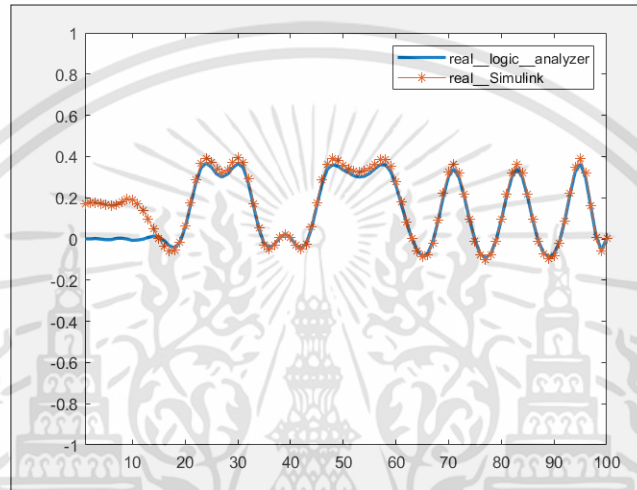


รูปที่ 4.51 พล็อตกราฟแสดงผลตอบสนองอิมพัลส์ของวงจร RRC Filter เทียบกับผลการจำลองจาก MATLAB

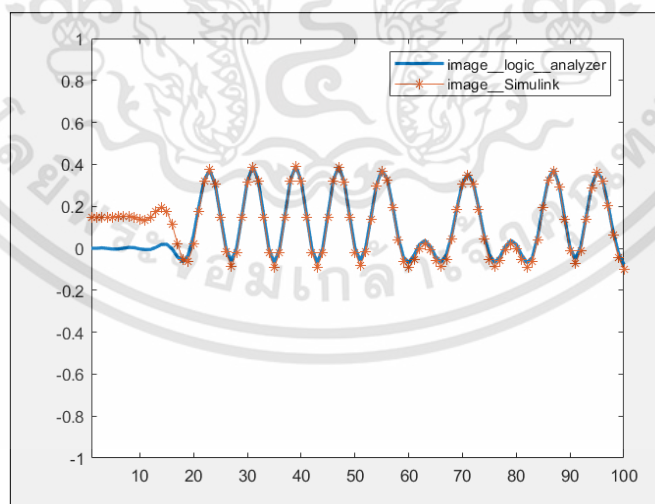
จากรูปที่ 4.51 จะเห็นได้ว่าผลตอบสนองอิมพัลส์ที่วัดได้จาก FPGA กับ MATLAB มีรูปร่างใกล้เคียงกัน ซึ่งอาจแตกต่างกันเล็กน้อยเนื่องจากวงจร RRC Filter ที่ออกแบบบน FPGA มีการตัดจำนวนบิตของสัญญาณเอาต์พุตออกเพื่อแก้ปัญหาการโอเวอร์โฟล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.56 พล็อตกราฟแสดงผลเปรียบเทียบระหว่างผลของสัญญาณ In-phase ที่วัดได้จาก FPGA กับผลที่วัดได้จากซอฟต์แวร์ MATLAB และรูปที่ 4.57 พล็อตกราฟแสดงผลเปรียบเทียบระหว่างผลของสัญญาณ Quadrature ที่วัดได้จาก FPGA กับผลที่วัดได้จากซอฟต์แวร์ MATLAB จะเห็นได้ว่าผลจากการออกแบบระบบสื่อสาร QPSK ภาคส่ง ด้วย MATLAB และ FPGA มีผลใกล้เคียงกัน หลังจากเฟรมของสัญญาณเอาต์พุตออกมา



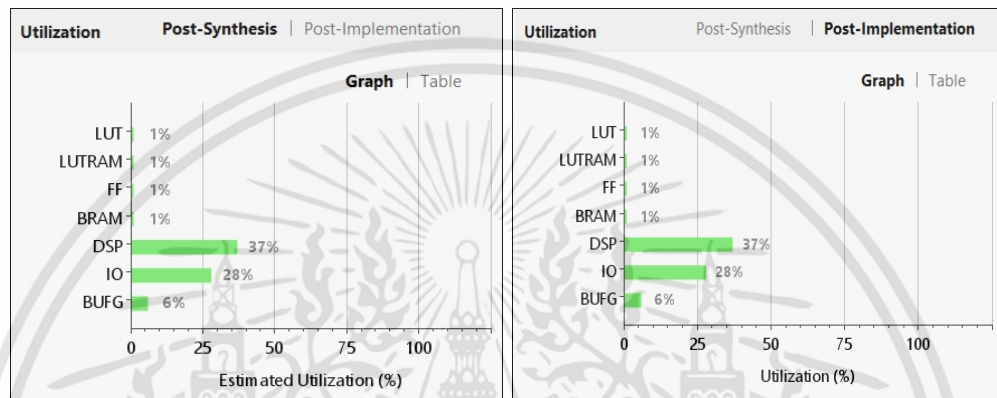
รูปที่ 4.56 ผลเปรียบเทียบสัญญาณ In-phase เอาต์พุตของระบบสื่อสาร QPSK ภาคส่ง ของสัญญาณ ระหว่าง FPGA และ MATLAB



รูปที่ 4.57 ผลเปรียบเทียบสัญญาณ Quadrature เอาต์พุตของระบบสื่อสาร QPSK ภาคส่ง ของสัญญาณ ระหว่าง FPGA และ MATLAB

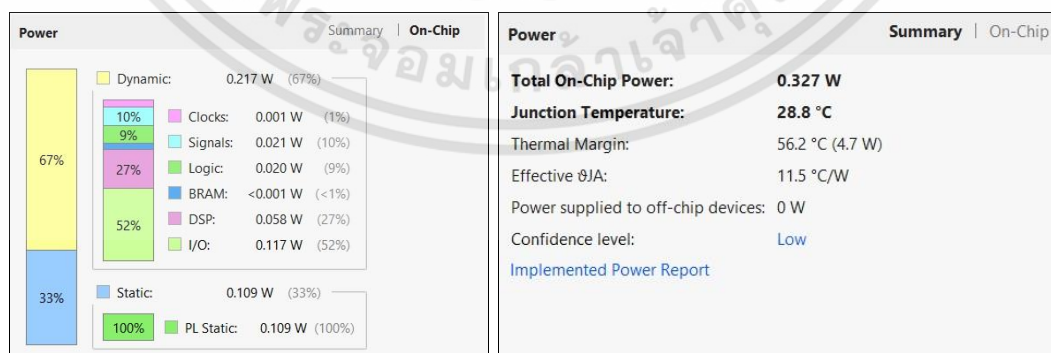
4.3.7 ผลรายงานจากแวย์ Vivado

จากการสังเคราะห์วงจรที่ได้ออกแบบระบบสื่อสาร QPSK ภาคส่ง ด้วยภาษา VHDL บนซอฟต์แวร์ Vivado 2019.2 สามารถดูผลการรายงานการใช้ทรัพยากรบนบอร์ด FPGA และผลแสดงค่าการทำงานต่างๆ ของบอร์ดได้



รูปที่ 4.58 กราฟแสดงจำนวนการใช้ทรัพยากรภายในบอร์ด FPGA โดยด้านซ้ายเป็นผลหลังทำการ Synthesis วงจร และด้านขวาเป็นผลหลังการทำ Implementation

จากรูปที่ 4.58 จะเห็นได้ว่าโค้ด VHDL ที่ได้ออกแบบระบบสื่อสาร QPSK ภาคส่ง ที่ได้นำไปสังเคราะห์เป็นวงจรดิจิทัลลงบอร์ด FPGA มีจำนวนการใช้ DSP slices สูงสุด 37 เปอร์เซ็นต์ เนื่องจากวงจร RRC Filter 41 tabs ที่เป็น Direct form มีการใช้การคำนวณค่อนข้างมาก ลองลงมาเป็นพอร์ต I/O 28 เปอร์เซ็นต์ และ Buffer ภายใน 6 เปอร์เซ็นต์ ส่วนที่เหลือเป็น LUT, Ram ของ LUT, Flipflop และ BRAM อย่างละ 1 เปอร์เซ็นต์



รูปที่ 4.59 ผลการแสดงผลการใช้พลังงานภายในบอร์ด FPGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.59 จะแสดงผลของพลังงานที่บอร์ด FPGA ใช้งาน เช่นพลังงานที่ชิป FPGA ใช้ เท่ากับ 0.327 W อุณหภูมิที่เกิดขึ้นเท่ากับ 28.8 เซลเซียส เป็นต้น

และจากรูปที่ 4.60 เป็นผลรายงานจำนวนเวลาที่ใช้ในการประมวลผลการทำงานของบอร์ด FPGA ใน 1 Cycle โดยนับจากความถี่ของสัญญาณนาฬิกาแรกเริ่มที่ป้อนเข้าไปในระบบ ซึ่งใช้เท่ากับความถี่สูงสุดที่บอร์ด FPGA (Zybo Z7-20) นี้ทำงานได้คือ 125 MHz ก่อนผ่านวงจรหารความถี่เพื่อนำไปใช้กับวงจรอื่นภายในระบบที่มีความซับซ้อนกว่า

Setup	Hold	Pulse Width
Worst Negative Slack (WNS): 6.315 ns	Worst Hold Slack (WHS): 0.118 ns	Worst Pulse Width Slack (WPWS): 3.500 ns
Total Negative Slack (TNS): 0.000 ns	Total Hold Slack (THS): 0.000 ns	Total Pulse Width Negative Slack (TPWS): 0.000 ns
Number of Failing Endpoints: 0	Number of Failing Endpoints: 0	Number of Failing Endpoints: 0
Total Number of Endpoints: 6	Total Number of Endpoints: 6	Total Number of Endpoints: 7

All user specified timing constraints are met.

รูปที่ 4.60 ค่าแสดงผลทางเวลาหลังการสังเคราะห์วงจรลงบอร์ด FPGA ด้วยความถี่ของสัญญาณนาฬิกาอินพุตเท่ากับ 125 MHz

จากรูปที่ 4.60 ค่าที่สำคัญก็คือ Worst Negative Slack (WNS) แสดงให้เห็นว่ามีค่าเป็นบวกเท่ากับ 6.315 ns แสดงว่าสามารถสังเคราะห์ระบบสื่อสาร QPSK ภาคส่งที่ความถี่ของสัญญาณนาฬิกาอินพุตเท่ากับ 125 MHz ได้

บทที่ 5

สรุปผลและข้อเสนอแนะ

ปริญญานิพนธ์เรื่อง การออกแบบระบบสื่อสาร QPSK ด้วย Software-Defined Radio (SDR) มีการสรุปผล และข้อเสนอแนะดังนี้

5.1 สรุปผล

จากผลการทดสอบระบบสื่อสาร QPSK ด้วย Software-Defined Radio (SDR) จะเห็นได้ว่าผลจากการทดลองในส่วนที่หนึ่ง ผลการ simulation ระบบสื่อสาร QPSK ที่ออกแบบด้วยซอฟต์แวร์ MATLAB และ Simulink นั้นสามารถทำงานได้จริง ซึ่งผลการทำงานของบล็อกต่างๆ บนซอฟต์แวร์ อาทิเช่น สแควมเบลอร์, ดีสแควมเบลอร์, RRC ฟิลเตอร์, QPSK มอดูเลเตอร์ และ QPSK ดีมอดูเลเตอร์ เป็นต้น เป็นไปตามทฤษฎีของการออกแบบระบบสื่อสาร ระบบสื่อสาร QPSK ภาครับสามารถซิงค์ไครโนสสัญญาณ และแก้ความผิดพลาดของสัญญาณที่ได้รับผลกระทบจากช่องสัญญาณได้ ทำให้การออกแบบระบบสื่อสารแบบ SDR นี้ มีความน่าเชื่อถือ

จากผลการทดสอบระบบสื่อสาร QPSK ด้วยซอฟต์แวร์ MATLAB สามารถแสดงได้ว่าระบบสื่อสาร QPSK ที่ได้ออกแบบสามารถใช้งานได้จริง การทดลองในส่วนที่สอง จึงเป็นการปรับใช้ระบบสื่อสาร QPSK ที่ได้ออกแบบบนซอฟต์แวร์ MATLAB และ Simulink กับฮาร์ดแวร์ประเภท SDR ซึ่งก็คือ USRP B210 เพื่อทำการทดสอบระบบสื่อสารจริงระหว่างแลปท็อป 2 เครื่อง ที่ทำหน้าที่เป็นเครื่องส่ง และเครื่องรับ ผลจากการทดลองในส่วนที่สองจะเห็นได้ว่า เครื่องรับสามารถรับสัญญาณจากเครื่องส่ง และถอดรหัสออกมาเป็นข้อมูลได้ถูกต้อง โดยพารามิเตอร์ที่ใช้ในการทดสอบได้แก่ค่า BER, แบนด์วิท, C/N, กำลังของสัญญาณ, EVM, MER, Constellation Diagram และ Eye Diagram ซึ่ง Symbol Rate สูงสุดที่แลปท็อปเครื่องส่งสามารถทำงานได้เท่ากับ 1 MHz ด้วย Master Clock Rate สูงสุดเท่ากับ 20 MHz ผลลัพธ์ที่วัดได้ จะเห็นได้ว่าลักษณะของกราฟ Constellation มี Constellation Point ของสัญญาณตามการมอดูเลตแบบ QPSK และ Eye Diagram มีลักษณะกราฟดวงตาที่เปิดกว้าง ทนทานต่อสัญญาณรบกวน แบนด์วิทเท่ากับ 1.3 GHz, C/N เท่ากับ 47.68 dB, กำลังของสัญญาณเท่ากับ -16.05 dBm, EVM (RMS) เท่ากับ 3.68% และค่า MER (RMS) เท่ากับ 28.68 dB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และจากการทดสอบระบบสื่อสาร QPSK ด้วย FPGA จะเห็นได้ว่าระบบสื่อสาร QPSK ภาคส่งที่ออกแบบด้วยภาษา VHDL มีความน่าเชื่อถือ เนื่องจากผลลัพธ์ของสัญญาณดิจิทัลที่นำไปเปรียบเทียบกับผลจากซอฟต์แวร์แมปแลป โดยบ่อนอินพุตเดียวกัน ให้ผลลัพธ์ทางสัญญาณที่คล้ายเคียงกัน ซึ่งสามารถนำไปพัฒนาต่อยอดเพื่อสร้าง SDR ขึ้นมาเองได้

การสร้างระบบสื่อสารด้วยแพลตฟอร์ม SDR นี้ จึงเหมาะสมแก่การศึกษา พัฒนา และประยุกต์ใช้เพื่อออกแบบระบบสื่อสาร เพราะระบบสื่อสารแบบ SDR สามารถทำงานได้จริง มีความน่าเชื่อถือ และปรับแต่งแก้ไขได้ง่าย

5.2 ข้อเสนอแนะ

การสร้างระบบสื่อสารด้วยแพลตฟอร์ม SDR นี้ จึงเหมาะสมแก่การศึกษา พัฒนา และประยุกต์ใช้เพื่อออกแบบระบบสื่อสาร เพราะระบบสื่อสารแบบ SDR สามารถทำงานได้จริง มีความน่าเชื่อถือ และปรับแต่งแก้ไขได้ง่าย

จากการทดสอบการออกแบบระบบสื่อสาร QPSK ด้วย SDR บนซอฟต์แวร์ MATLAB และ Simulink จะเห็นได้ว่าระบบสื่อสารที่ได้ออกแบบด้วยแลปทอป ASUS TUF Gaming A15 FA506II-AL012T สามารถส่งข้อมูลได้ด้วย Symbol Rate สูงสุด เท่ากับ 1 MHz ซึ่งเป็นผลเนื่องจากประสิทธิภาพของระบบประมวลผลบนแลปทอปที่ใช้รันระบบสื่อสาร QPSK ดังนั้นการออกแบบระบบสื่อสารด้วย SDR ก็จะถูกจำกัดตามสเปกของระบบประมวลผล ดังนั้นในการออกแบบระบบสื่อสารด้วย SDR ในปัจจุบัน จึงอาศัย FPGA ในการประมวลผลทางสัญญาณดิจิทัล ซึ่งจะทำให้ระบบสื่อสารทำงานได้รวดเร็ว และมีประสิทธิภาพ

บรรณานุกรม

- [1] Jusnaini Muslimin, Ani Liza Asnawi, Ahmad Fadzil Ismail และ Ahmad Zamani Jusoh. “SDR-based Transceiver of Digital Communication System using USRP and GNU.” Electrical and Computer Department, Kulliyah of Engineering, International Islamic University Malaysia (IIUM), 2016.
- [2] Ettus Research™. “USRP X Series.”
<https://www.ettus.com/products>
- [3] Ettus Research. “USRP B200 and B210 hardware”.
https://www.ettus.com/wp-content/uploads/2019/01/b200-b210_spec_sheet.pdf
- [4] Ettus Research. “USRP E320 board”.
<https://www.ettus.com/all-products/usrp-e320-board/>
- [5] Ettus Research. “USRP E320 board”.
<https://www.ettus.com/all-products/usrp-e320/>
- [6] “Scrambler”.
<https://en.wikipedia.org/wiki/Scrambler>
- [7] radartutorial.eu.” Barker Codes”.
<https://www.radartutorial.eu/08.transmitters/Barker%20Code.en.html>
- [8] sciencedirect.” FIR Filters”.
<https://www.sciencedirect.com/topics/engineering/fir-filters>
- [9] rfwireless-world.” Raised Cosine Filter vs Root Raised Cosine Filter-difference between Raised Cosine and Root Raised Cosine Filters”.
<https://www.rfwireless-world.com/Terminology/Raised-Cosine-Filter-vs-Root-Raised-Cosine-Filter.html>
- [10] “Root-raised-cosine filter”.
https://en.wikipedia.org/wiki/Root-raised-cosine_filter

- [11] The Editors of Encyclopaedia Britannica. “The Editors of Encyclopaedia Britannica”
<https://www.britannica.com/science/interpolation>
- [12] Siemens. “Digital Signal Processing: Sampling Rates, Bandwidth, Spectral Lines, and more...”
<https://community.sw.siemens.com/s/article/digital-signal-processing-sampling-rates-bandwidth-spectral-lines-and-more>.
- [13] Leon W. Couch II. Digital and Analog Communication Systems. 8th edition. Pearson / Prentice. Chapter 6
- [14] electronics.” What is Bit Error Rate: BER tutorial”.
<https://www.electronics-notes.com/articles/radio/bit-error-rate-ber/what-is-ber-definition-tutorial.php>
- [15] “Additive white Gaussian noise”
https://en.wikipedia.org/wiki/Additive_white_Gaussian_noise
- [16] “Crosstalk”
<https://en.wikipedia.org/wiki/Crosstalk>
- [17] allaboutcircuits.” “Understanding Automatic Gain Control”
<https://www.allaboutcircuits.com/technical-articles/understanding-automatic-gain-control/>
- [18] “Coarse Frequency Compensator”
<https://www.mathworks.com/help/comm/ref/coarsefrequencycompensator.html>
- [19] “comm.CoarseFrequencyCompensator”
<https://www.mathworks.com/help/comm/ref/comm.coarsefrequencycompensator-system-object.html>
- [20] “comm.SymbolSynchronizer”

<https://www.mathworks.com/help/comm/ref/comm.symbolsynchronizer-system-object.html>

[21] “comm.CarrierSynchronizer”

<https://www.mathworks.com/help/comm/ref/comm.carriersynchronizer-system-object.html>

[22] “Preamble Detector”

<https://www.mathworks.com/help/comm/ref/preambledetector.html>

[23] “Frame synchronization”

https://en.wikipedia.org/wiki/Frame_synchronization

[24] newbedev.” definition of power signals and energy signals”.

<https://newbedev.com/definition-of-power-signals-and-energy-signals>

[25] ศ.ดร.พรชัย ทรัพย์นิธิ. การสื่อสารดิจิทัล เล่มที่ 1 ฉบับปรับปรุงครั้งที่ 3.พิมพ์ ครั้งที่ 1: 1 กรกฎาคม พ.ศ. 2563.ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สจล.

[26] “อัตราส่วนของผู้ให้บริการต่อสัญญาณรบกวน - Carrier-to-noise ratio Wikipedia site:isecosmetic.com”

https://isecosmetic.com/wiki/Carrier-to-noise_ratio