ระบบการสื่อสารแบบ DIRECT SEQUENCE SPREAD SPECTRUM

DIRECT SEQUENCE SPREAD SPECTRUM COMMUNICATION SYSTEM

โดย

นายปธิกร อันชื่น

นายปิยะภูมิ ดีทองคำ

นายพลากร ปีติโภศล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

ระบบการสื่อสารแบบ DIRECT SEQUENCE SPREAD SPECTRUM
DIRECT SEQUENCE SPREAD SPECTRUM COMMUNICATION SYSTEM

โดย

นายปธิกร อันชื่น

นายปิยะภูมิ ดีทองคำ

นายพลากร ปีติโกศล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการสื่อสารแบบ DIRECT SEQUENCE SPREAD SPECTRUM

DIRECT SEQUENCE SPREAD SPECTRUM COMMUNICATION SYSTEM

โดย

นายปธิกร	อันชื่น	53010914
นายปิยะภูมิ	ดีทองคำ	53010998
นายพลากร	ปีติโกศล	53011074

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์ ดร.ศรวัฒน์ ชิวปรีชา

ผู้ช่วยศาสตราจารย์ อัครพล ตรีรัตน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2556

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบการสื่อสารแบบ DIRECT SEQUENCE SPREAD SPECTRUM

DIRECT SEQUENCE SPREAD SPECTRUM

ผู้จัดทำ

1. นายปธิกร อันชื่น 53010914
2. นายปิยะภูมิ ดีทองคำ 53010998
3. นายพลากร ปีติโกศล 53011074

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.ศรวัฒน์ ชิวปรีชา)

..... อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ อัครพล ตรีรัตน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงได้ด้วยดี ด้วยความเมตตาและความช่วยเหลือจาก ผู้ช่วยศาสตราจารย์ดร. ศรวัฒน์ ชิวปรีชา และผู้ช่วยศาสตราจารย์อัครพล ตรีรัตน์ ซึ่งเป็นอาจารย์ที่ปรึกษาโครงงาน ที่กรุณาให้คำแนะนำและคำปรึกษาให้กับคณะผู้จัดทำ ทางผู้จัดทำรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านอาจารย์ที่กรุณาและเอาใจใส่ต่อคณะผู้จัดทำ ขอกราบขอบพระคุณท่านอาจารย์เป็นอย่างสูงไว้ ณ โอกาสนี้

สุดท้ายนี้ ผู้จัดทำขอกราบขอบพระคุณ บิดา-มารดา และครอบครัวของคณะผู้จัดทำที่เป็นกำลังใจและให้การสนับสนุนในทุกๆเรื่อง ทำให้คณะผู้จัดทำสามารถทำโครงงานนี้สำเร็จลุล่วงด้วยดี

นายปธิกร อันชื่น
นายปิยะภูมิ ดีทองคำ
นายพลากร ปีติโกศล
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบการสื่อสารแบบไดเรกต์ซีเควนซ์สเปรดสเปกตรัม

DIRECT SEQUENCE SPREAD SPECTRUM COMMUNICATION SYSTEM

โดย นายปธิกร อันชื่น 53010914
นายปิยะภูมิ ดีทองคำ 53010998
นายพลากร ปีติโกศล 53011074

อาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์ ดร.ศรวัฒน์ ชิวปรีชา
ผู้ช่วยศาสตราจารย์ อัครพล ตรีรัตน์

บทคัดย่อ

โครงงานนี้นำเสนอระบบการสื่อสารแบบไดเรกต์ซีเควนซ์สเปรดสเปกตรัมซึ่งถูกใช้งานในระบบ CDMA โดยการทำให้สเปกตรัมของสัญญาณที่ต้องการส่งมีลักษณะคล้ายสัญญาณรบกวนที่มีความหนาแน่นของสเปกตรัมกำลังต่ำ วิธีการที่ใช้ในการทำให้สัญญาณมีลักษณะคล้ายสัญญาณรบกวนนั้นสามารถทำได้โดยใช้รหัสที่มีอัตราชิพที่สูงกว่ากว่าอัตราบิตข้อมูลมาคูณเข้ากับสัญญาณที่ต้องการส่ง ในการถอดรหัสสัญญาณที่ทำการแผ่นั้นทำได้โดยนำรหัสเดียวกันกับในฝั่งส่งมาคูณกลับโดยที่ทางด้านภาครับจะสามารถทำการกู้ข้อมูลเดิมมาได้อย่างถูกต้อง ผลการทดลองแสดงได้โดยการจำลองการทำงานบนโปรแกรม MATLAB SIMULINK นอกจากนั้นยังจะได้ทำการออกแบบสร้างต้นแบบทางฮาร์ดแวร์ของชุดสาธิตการทำงานระบบการสื่อสารแบบไดเรกต์ซีเควนซ์สเปรดสเปกตรัมบนอุปกรณ์ FPGA

ABSTRACT

This project presents a direct sequence spread spectrum communication system which is used in the CDMA. The transmitted signal spectrums are similar to the noise which has a low power spectral density. For the used method to make signal spectrum similar to noise that can be done by using the spreading code which has chip rate higher than data bit rate multiplied with the

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

send data. In the decoding is done by despreading with the same code sent in the transmitter where the receiver will be able to recover the original data. Experimental results are shown by simulations using MATLAB SIMULINK. Moreover, this project also developing the hardware prototype of demonstration test set of direct sequence spreading spectrum communication system on FPGA.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	IV
สารบัญรูป	IX
สารบัญตาราง	XVI
บทที่ 1 บทนำ	**1**
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงงาน	2
บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง	**3**
2.1 การทำสเพรดสเปกตรัมแบบไดเรกต์ซีเควนซ์ (DIRECT SEQUENCE SPREAD SPECTRUM)	3
2.2 ลำดับของชิฟรีจิสเตอร์ (SHIFT REGISTER)	4
2.3 ฟังก์ชันพีเอ็นออโตคอร์รีเลชัน (PN AUTOCORRELATION)	5
2.4 รหัสพีเอ็น	7
2.4.1 คุณสมบัติของ PN CODE	7
2.4.1.1 SHORT PN CODE	10
2.4.1.2 LONG PN CODE	10
2.5 วงจรภาคส่งสัญญาณ CDMA	11
2.6 วงจรภาครับของสัญญาณ CDMA	12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.7 คุณลักษณะเฉพาะของระบบ CDMA	14
2.7.1 SOFT CAPACITY	14
2.7.2 MULTIPATH RESOLUTION	14
2.7.3 SECTORIZATION FOR CAPACITY	15
2.7.4 NO GUARD TIME REQUIRED	15
2.8 ทฤษฎีการชักค่าตัวอย่าง (SAMPLING THEOREM)	16
2.9 การออกแบบวงจรดิจิตอลด้วยภาษา VHDL	20
2.9.1 ภาษา VHDL และส่วนประกอบต่าง ๆ ของภาษา	21
2.9.1.1 หน่วยการออกแบบเอนทิตี	21
2.9.1.2 หน่วยการออกแบบสถาปัตยกรรม	22
2.9.1.3 หน่วยการออกแบบแพ็กเก็ต	23
2.9.1.4 หน่วยการออกแบบโครงแบบ	24
2.9.2 การออกแบบวงจรดิจิตอลด้วยอุปกรณ์ FPGA	25
2.9.2.1 การออกแบบโดยใช้ภาษาอธิบายการทำงานของฮาร์ดแวร์	26
2.9.2.2 การจำลองการทำงานของวงจร (SIMULATION)	27
2.9.2.3 การสังเคราะห์วงจร	27
2.9.2.4 การแบ่งวงจร (PARTITIONING)	27
2.9.2.5 การวางอุปกรณ์ (PLACEMENT)	28
2.9.2.6 การเชื่อมต่อสัญญาณ (ROUTING)	28
2.9.2.7 การโปรแกรมอุปกรณ์ FPGA (CONFIGURATION)	28
2.10 วงจรแปลงสัญญาณเชิงอุปมานเป็นสัญญาณเชิงเลข (ANALOG TO DIGITAL CONVERTER)	29
2.11 วงจรแปลงสัญญาณเชิงเลขเป็นสัญญาณเชิงอุปมาน (DIGITAL TO ANALOG CONVERTER)	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
บทที่ 3	**การออกแบบและการจัดทำโครงงาน**	**31**
	3.1 การออกแบบวงจรด้วยโปรแกรม MATLAB	31
	3.2 การออกแบบวงจรด้วยภาษา VHDL	36
	3.2.1 การออกแบบวงจรภาคส่ง	36
	3.2.1.1 การออกแบบอุปกรณ์ในภาคส่ง	36
	3.2.2 การออกแบบวงจรภาครับ	40
	3.2.2.1 การออกแบบอุปกรณ์ในภาครับ	41
	3.3 การจัดเก็บผลการทดลอง	45
	3.3.1 การจำลองการทำงานผ่านโปรแกรมMATLAB	45
	3.3.2 การออกแบบสร้างต้นแบบทางฮาร์ดแวร์ของชุดสาธิตการทำงาน ที่ออกแบบด้วยภาษา VHDL ผ่านโปรแกรม MODELSIM 10.1B และ XILINX ISE 14.5	47
	3.3.3 การทดสอบวงจรกรองสัญญาณผ่านทาง FPGA	51
บทที่ 4	**ผลการทดลอง**	**60**
	4.1 ผลการทดลองด้วยโปรแกรม MATLAB	60
	4.1.1 ภาคส่ง	60
	4.1.1.1 สัญญาณข้อมูลที่ส่งจากผู้ใช้รายที่ 1	60
	4.1.1.2 สัญญาณข้อมูลที่ส่งจากผู้ใช้รายที่ 2	62
	4.1.1.3 สัญญาณข้อมูลที่ส่งจากผู้ใช้รายที่ 3	64
	4.1.1.4 การส่งข้อมูลของผู้ใช้ทั้ง 3 ราย ผ่านช่องสัญญาณ	66
	4.1.2 ภาครับ	67
	4.1.2.1 สัญญาณข้อมูลที่รับของผู้ใช้รายที่ 1	68
	4.1.2.2 สัญญาณข้อมูลที่รับของผู้ใช้รายที่ 2	71
	4.1.2.3 สัญญาณข้อมูลที่รับของผู้ใช้รายที่ 3	73
	4.2 ผลการทดลองด้วยภาษา VHDL ผ่านโปรแกรม MODELSIM 10.1B	75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
4.2.1 ภาคส่ง	75
4.2.1.1 วงจรหารความถี่ (FREQUENCY DIVIDER)	75
4.2.1.2 ชิฟต์รีจิสเตอร์ (SHIFT REGISTER)	76
4.2.1.3 วงจรแม็ปค่า (BPSK MAPPER)	77
4.2.1.4 วงจรคูณ (MULTIPLIER)	77
4.2.1.5 วงจรนับค่า (COUNTER)	78
4.2.1.6 วงจรกำเนิดสัญญาณพาหะ (DIGITAL OSCILLATOR)	79
4.2.1.7 วงจรรวมสัญญาณ (SUMMING)	80
4.2.2 ภาครับ	80
4.2.2.1 วงจรหารความถี่ (FREQUENCY DIVIDER)	81
4.2.2.2 ชิฟต์รีจิสเตอร์ (SHIFT REGISTER)	81
4.2.2.3 วงจรแม็ปค่า (BPSK MAPPER)	82
4.2.2.4 วงจรคูณ (MULTIPLIER)	83
4.2.2.5 วงจรนับค่า (COUNTER)	84
4.2.2.6 วงจรกำเนิดสัญญาณพาหะ (DIGITAL OSCILLATOR)	84
4.2.2.7 วงจรบวกสะสมค่า (ACCUMULATOR)	85
4.2.2.8 วงจรกำหนดขอบเขต (COUNTER ACCUMULATOR)	86
4.2.2.9 วงจรตัดสินใจบิต (DECISION)	87
4.2.3 ภาพรวมการทำงานของภาครับและภาคส่ง	88
4.3 ผลการทดลองการทำงานบนอุปกรณ์ FPGA	93

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
บทที่ 5	สรุปผลและข้อเสนอแนะ	108
	5.1 สรุปผลการทดลอง	108
	5.2 ข้อเสนอแนะ	108
บรรณานุกรม		109
ภาคผนวก ก	โปรแกรม MATLAB	110
ภาคผนวก ข	โปรแกรมภาษา VHDL	125

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่

2.1 ตัวอย่างของชิฟรีจิสเตอร์ที่มีการป้อนกลับแบบเชิงเส้น 4

2.2 ฟังก์ชันของพีเอ็นออโตคอรีเลชั่น (PN AUTOCORRELATION) 7

2.3 การสร้างรหัสพีเอ็นโดยอาศัยชิฟต์รีจิสเตอร์และ XOR GATE 8

2.4 รหัสพีเอ็น ที่มีสมการพหุนาม คือ $X^3 + X^2 + 1$ 9

2.5 AUTOCORRELATION ของรหัส PN CODE 9

2.6 วงจรภาคส่งของระบบ CDMA ที่ใช้การมอดูเลตแบบ BPSK 11

2.7 ตัวอย่างของสัญญาณในวงจรภาคส่งของระบบไดเรกต์ซีเควนซ์ที่ใช้ BPSK 12

2.8 วงจรภาครับของระบบไดเรกซ์ซีเควนซ์ที่ใช้การมอดูเลตแบบ BPSK 13

2.9 ตัวอย่างของสัญญาณในวงจรภาครับของระบบไดเรกต์ซีเควนซ์ที่ใช้ BPSK 14

2.10 สเปกตรัมของสัญญาณที่ถูกชักค่าตัวอย่าง 17

2.11 สเปกตรัมของสัญญาณที่เกี่ยวเนื่องกับรูปที่ 2.10 เมื่อใช้ช่วงเวลาในการสุ่มค่าสัญญาณต่างๆ 19

2.12 โครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี 21

2.13 โครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม 23

2.14 โครงสร้างโดยทั่วไปของส่วนการประกาศแพ็กเก็ต 24

2.15 โครงสร้างโดยทั่วไปของบอดี้แพ็กเก็ต 24

2.16 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ 25

2.17 ลักษณะของตัว FPGA 26

2.18 วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล 29

2.19 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก 30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

3.1 แบบจำลองของระบบไดเรกต์ซีเควนซ์สเปรดสเปกตรัม 31
3.2 แบบจำลองของภาคส่ง 32
3.3 บล็อกสัญญาณรบกวนเกาส์สีขาวแบบบวก (Additve White Gaussian Noise : AWGN) 33
3.4 แบบจำลองของภาครับ 33
3.5 โมเดลการจำลองของภาคส่งและภาครับ ของจำนวนผู้ใช้ 3 ราย 35
3.6 แผนผังการทำงานการออกแบบวงจรภาคส่งด้วยภาษา VHDL 36
3.7 วงจรรวมของภาคส่ง 37
3.8 วงจรหารความถี่ (Frequency divider) 38
3.9 ชิฟต์รีจิสเตอร์ (Shift Register) 38
3.10 วงจรแม็ปค่า (BPSK Mapper) 38
3.11 วงจรคูณ (Multiplier) 39
3.12 วงจรนับค่า (Counter) 39
3.13 วงจรกำเนิดสัญญาณพาหะ (Digital Oscillator) 39
3.14 วงจรรวมสัญญาณ (Summing) 40
3.15 แผนผังการทำงานการออกแบบวงจรภาครับด้วยภาษา VHDL 40
3.16 วงจรรวมของภาครับ 41
3.17 วงจรหารความถี่ (Frequency divider) 42
3.18 ชิฟต์รีจิสเตอร์ (Shift Register) 42
3.19 วงจรแม็ปค่า (BPSK Mapper) 42
3.20 วงจรคูณ (Multiplier) 43
3.21 วงจรนับค่า (Counter) 43
3.22 วงจรบวกสะสมค่า (Accumulator) 44
3.23 วงจรบัฟเฟอร์ (Buffer) 44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

3.24 วงจรกำหนดขอบเขต (Counter Accumulator) 44
3.25 วงจรตัดสินใจบิต (Decision) 45
3.26 การรันการจำลองการทำงานผ่านโปรแกรมแมทแลป 46
3.27 สัญญาณอินพุตจากสโคป 46
3.28 สัญญาณเอาต์พุตจากสโคป 47
3.29 การสร้าง VHDL Test Bench 48
3.30 การตั้งชื่อ VHDL Test Bench 48
3.31 การเลือกไฟล์ที่จะนำมาสร้าง VHDL Test Bench 49
3.32 หน้าต่างสรุปการสร้าง VHDL Test Bench 49
3.33 การเลือกเปิดหน้าต่างการจำลอง 50
3.34 การเลือกการจำลองผ่านโปรแกรม MODELSIM 10.1B 50
3.35 ผลการจำลองการทำงานของวงจรสร้างรหัสพีเอ็น (PN SEQUENCE GENERATOR) 51
3.36 การเลือกโปรแกรมกำหนดพอร์ต 51
3.37 โปรแกรม PLANAHEAD ที่เรียกขึ้นมา 52
3.38 ลักษณะโปรแกรม PLANAHEAD ที่เรียกขึ้นมา 52
3.39 การกำหนดพอร์ต 53
3.40 FPGA พร้อมทำการโปรแกรม 53
3.41 การ Implement Design 54
3.42 การเลือกเปิดโปรแกรม iMPACT 54
3.43 โปรแกรม iMPACT 55
3.44 การเลือก Boundary Scan 55
3.45 การเลือก Initialize Chain 56
3.46 หน้าต่าง Assign New Configuration File 56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

3.47 หน้าต่างเลือกไฟล์โปรแกรมที่จะนำมาเบิร์นลงบน FPGA 57
3.48 หน้าต่างกำหนดคุณสมบัติของอุปกรณ์ภายใน FPGA 57
3.49 หน้าต่างกำหนดอุปกรณ์ภายใน FPGA 58
3.50 การลบโปรแกรมที่อยู่ใน FPGA 58
3.51 การโปรแกรมวงจรลงบน FPGA สำเร็จ 59
3.52 FPGA ที่โปรแกรมวงจรสำเร็จ 59
4.1 การทำการสเปรดของข้อมูลผู้ใช้รายที่ 1 61
4.2 ข้อมูลที่ถูกมอดูเลตแบบ BPSK ของผู้ใช้รายที่ 1 62
4.3 การทำการสเปรดของข้อมูลผู้ใช้รายที่ 2 63
4.4 ข้อมูลที่ถูกมอดูเลตแบบ BPSK ของผู้ใช้รายที่ 2 64
4.5 การทำการสเปรดของข้อมูลผู้ใช้รายที่ 3 65
4.6 ข้อมูลที่ถูกมอดูเลตแบบ BPSK ของผู้ใช้รายที่ 3 66
4.7 สัญญาณก่อนการส่ง 67
4.8 การเปรียบเทียบการรับส่งข้อมูลของผู้ใช้แต่ละราย 68
4.9 การดีสเปรดและดีมอดูเลตของข้อมูลผู้ใช้รายที่ 1 69
4.10 การตัดสินใจบิตข้อมูลของผู้ใช้รายที่ 1 70
4.11 การดีสเปรดและดีมอดูเลตของข้อมูลผู้ใช้รายที่ 2 71
4.12 การตัดสินใจบิตข้อมูลของผู้ใช้รายที่ 2 72
4.13 การดีสเปรดและดีมอดูเลตของข้อมูลผู้ใช้รายที่ 3 73
4.14 การตัดสินใจบิตข้อมูลของผู้ใช้รายที่ 3 74
4.15 วงจรหารความถี่ (Frequency divider) 75
4.16 ผลการจำลองการทำงานของวงจรหารความถี่ 76
4.17 ชิฟต์รีจิสเตอร์ (Shift Register) 76
4.18 ผลการจำลองการทำงานของชิฟต์รีจิสเตอร์ 76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

รูปที่

4.19 วงจรแม็ปค่า (BPSK Mapper) 77
4.20 ผลการจำลองการทำงานของวงจรแม็ปค่า 77
4.21 วงจรคูณ (Multiplier) 77
4.22 ผลการจำลองการทำงานของวงจรคูณ 78
4.23 วงจรนับค่า (Counter) 78
4.24 ผลการจำลองการทำงานของวงจรนับค่า 79
4.25 วงจรกำเนิดสัญญาณพาหะ (Digital Oscillator) 79
4.26 ผลการจำลองการทำงานของวงจรกำเนิดสัญญาณพาหะ 79
4.27 วงจรรวมสัญญาณ (Summing) 80
4.28 ผลการจำลองการทำงานของวงจรรวมสัญญาณ 80
4.29 วงจรหารความถี่ (Frequency divider) 81
4.30 ผลการจำลองการทำงานของวงจรหารความถี่ 81
4.31 ชิฟต์รีจิสเตอร์ (Shift Register) 81
4.32 ผลการจำลองการทำงานของชิฟต์รีจิสเตอร์ 82
4.33 วงจรแม็ปค่า (BPSK Mapper) 82
4.34 ผลการจำลองการทำงานของวงจรแม็ปค่า 83
4.35 วงจรคูณ (Multiplier) 83
4.36 ผลการจำลองการทำงานของวงจรวงจรคูณ 83
4.37 วงจรนับค่า (COUNTER) 84
4.38 ผลการจำลองการทำงานของวงจรนับค่า 84
4.39 วงจรกำเนิดสัญญาณพาหะ (Digital Oscillator) 84
4.40 ผลการจำลองการทำงานของวงจรกำเนิดสัญญาณพาหะ 85
4.41 วงจรบวกสะสมค่า (Accumulator) 85
4.42 ผลการจำลองการทำงานของวงจรบวกสะสมค่า 86

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.43	วงจรกำหนดขอบเขต (COUNTER ACCUMULATOR)	86
4.44	ผลการจำลองการทำงานของวงจรกำหนดขอบเขต	86
4.45	วงจรตัดสินใจบิต (DECISION)	87
4.46	ผลการจำลองการทำงานของวงจรตัดสินใจบิต	87
4.47	ผลการจำลองการทำงานของภาคส่งของผู้ใช้รายที่ 1	88
4.48	ผลการจำลองการทำงานของภาคส่งของผู้ใช้รายที่ 2	89
4.49	ผลการจำลองการทำงานของภาคส่งของผู้ใช้รายที่ 3	90
4.50	ผลการจำลองการทำงานของภาครับของผู้ใช้รายที่ 1	91
4.51	ผลการจำลองการทำงานของภาครับของผู้ใช้รายที่ 2	92
4.52	ผลการจำลองการทำงานของภาครับของผู้ใช้รายที่ 3	93
4.53	สัญญาณไซน์เวฟที่ได้จากวงจรดิจิตอลออสซิลเลเตอร์	94
4.54	CH.1 สัญญาณข้อมูลของผู้ใช้รายที่ 1 CH.2 สัญญาณ PN CODE 1	94
4.55	CH.1 สัญญาณข้อมูลที่ถูกสเปรดจากรหัสพีเอ็นของผู้ใช้รายที่ 1 CH.2 สัญญาณ BPSK ของผู้ใช้รายที่ 1	95
4.56	CH.1 สัญญาณข้อมูลของผู้ใช้รายที่ 2 CH.2 สัญญาณ PN CODE 2	96
4.57	CH.1 สัญญาณข้อมูลที่ถูกสเปรดจากรหัสพีเอ็นของผู้ใช้รายที่ 2 CH.2 สัญญาณ BPSK ของผู้ใช้รายที่ 2	97
4.58	CH.1 สัญญาณข้อมูลของผู้ใช้รายที่ 3 CH.2 สัญญาณ PN CODE 3	98
4.59	CH.1 สัญญาณข้อมูลที่ถูกสเปรดจากรหัสพีเอ็นของผู้ใช้รายที่ 3 CH.2 สัญญาณ BPSK ของผู้ใช้รายที่ 3	99
4.60	สัญญาณรวมจากผู้ใช้ทั้ง 3 ราย	100
4.61	สัญญาณเปรียบเทียบระหว่างอินพุตกับเอาต์พุตของผู้ใช้รายที่ 1	101
4.62	สัญญาณเปรียบเทียบระหว่างอินพุตกับเอาต์พุตของผู้ใช้รายที่ 2	102
4.63	สัญญาณเปรียบเทียบระหว่างอินพุตกับเอาต์พุตของผู้ใช้รายที่ 3	103

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.64	กราฟความสัมพันธ์ระหว่าง BER กับ SNR ของผู้ใช้แต่ละราย	104
4.65	กราฟความสัมพันธ์ระหว่าง BER กับ SNRเฉลี่ยของผู้ใช้ทั้ง 3 ราย	105
4.66	สเปกตรัมของสัญญาณข้อมูลก่อนการสเปรดสเปกตรัม	106
4.67	สเปกตรัมของสัญญาณข้อมูลหลังการสเปรดสเปกตรัม	107

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตารางที่

3.1 การทำงานของบล็อกต่างๆ ในภาคส่ง 32
3.2 การทำงานของบล็อกต่างๆ ในภาครับ 34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันเทคโนโลยีที่สามารถตอบสนองความต้องการของมนุษย์ได้ในหลายๆด้าน เช่น บริการด้านเสียง (Voice) , การสื่อสารข้อมูล (Data) และสื่อผสม (Multimedia) นั่นคือ ระบบ CDMA (Code Division Multiple Access) ซึ่งระบบนี้ได้ถูกจัดให้เป็นเทคโนโลยีการสื่อสารไร้สายในยุคที่ 3 (3G; The Third Generation Wireless System) โดยความโดดเด่นของระบบ CDMA ที่มีมากกว่าระบบ FDMA (Frequency Division Multiple Access) และ ระบบ TDMA (Time Division Multiple Access) คือ การรองรับบริการได้เต็มรูปแบบ รวมทั้งยังรองรับจำนวนผู้ใช้ได้มากขึ้น และในขณะนี้ได้มีผู้สนใจระบบ CDMA ก็มีมากขึ้นทุกวันอีกทั้งยังมีการให้บริการแล้วในเกือบทุกพื้นที่

CDMA มีวิธีการแบ่งแต่ละสัญญาณด้วยรหัส ซึ่งผู้ใช้ในระบบแต่ละคนจะได้รับ code ที่แตกต่างกัน โดย code ที่ผู้ใช้แต่ละคนได้รับจะไม่ซ้ำกัน ขณะเดียวกันสถานีฐานก็มี code แยกต่างหาก เพื่อแยกว่ารับสัญญาณจากสถานีฐานไหน ซึ่งทำให้ผู้ใช้แต่ละคนได้รับข้อมูลในส่วนของตัวเองได้อย่างถูกต้องซึ่งเป็นเทคโนโลยีที่แก้ไขข้อจำกัดของการสื่อสารแบบไร้สายด้วยวิธีง่ายๆและมีประสิทธิภาพ

1.2 วัตถุประสงค์

1) เพื่อศึกษาการทำงานของเทคโนโลยีการสื่อสารไร้สายระบบ CDMA
2) เพื่อศึกษาการมัลติเพล็กซ์แบบแบ่งรหัส
3) เพื่อศึกษาและสร้างรูปแบบการใช้ช่องสัญญาณอย่างมีประสิทธิภาพในแบนด์วิดธ์ที่จำกัด
4) เพื่อศึกษาและสร้างกระบวนการในการทำดิจิตอลมอดูเลชั่น
5) เพื่อสร้างต้นแบบทางฮาร์ดแวร์ของชุดสาธิตการทำงานระบบการสื่อสารแบบไดเรกต์ซีเควนซ์สเปรดสเปกตรัม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตของปริญญานิพนธ์

1) ศึกษาการมัลติเพล็กซ์แบบแบ่งรหัส สำหรับการรับส่งข้อมูลผ่านการสื่อสาร

2) สามารถแก้ไขข้อจำกัดของการสื่อสารในเรื่องความจุของช่องสัญญาณให้สามารถใช้งานได้มากขึ้น

3) จำลองการรับส่งข้อมูลในระบบสื่อสารด้วยโปรแกรม MATLAB

4) สร้างฮาร์ดแวร์สำหรับรับส่งข้อมูล

5) ฝั่งรับสามารถรับข้อมูลได้อย่างถูกต้องมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 การทำสเปรดสเปกตรัมแบบไดเรกต์ซีเควนซ์ (Direct Sequence Spread Spectrum) [1]

ระบบสเปกตรัมแบบแผ่มีลักษณะพิเศษกว่าระบบอื่น คือการทำให้สเปกตรัมของสัญญาณที่ต้องการส่งมีลักษณะคล้ายสัญญาณรบกวนที่มีความหนาแน่นของสเปกตรัมกำลังต่ำ (Power spectral density) วิธีการหนึ่งที่ใช้ในการทำให้สัญญาณมีลักษณะคล้ายสัญญาณรบกวนนั้นสามารถทำได้โดยใช้รหัสที่มีอัตราชิพ (Chip rate) ที่สูงกว่ากว่าอัตราบิตข้อมูล (Bit rate) มาคูณเข้ากับสัญญาณที่ต้องการส่งซึ่งระบบสื่อสารชนิดนี้มีชื่อว่า Direct Sequence Spread Spectrum (DSSS) ในการถอดรหัสสัญญาณที่ทำการแผ่นั้นทำได้โดยนำรหัสเดียวกันกับในภาคส่งมาคูณกลับ โดยที่ทางด้านภาครับจะสามารถทำการกู้ข้อมูลเดิมมาได้อย่างถูกต้อง รหัสที่ใช้ส่งสัญญาณที่ถอดกลับมาจะมีลักษณะคล้ายสัญญาณรบกวน จากคุณสมบัตินี้จึงมีการนำมาใช้กับการสื่อสารที่ต้องการความปลอดภัยของข้อมูลสูง เช่น ระบบสื่อสารทางการทหาร เป็นต้น

การเลือกรหัสแผ่ที่เหมาะสมจัดว่าเป็นหัวใจสำคัญของการทำสเปรดสเปกตรัม โดยทั่วไปรหัสแผ่ที่ดีควรจะมีคุณสมบัติที่คล้ายคลึงกับสัญญาณรบกวนมากที่สุด และตัวอย่างของรหัสแผ่ที่ได้รับความสนใจมากก็คือ รหัส PN (Pseudorandom noise) ซึ่งมีการใช้งานในมาตรฐานระบบโทรศัพท์เคลื่อนที่ CDMA IS-95A การเลือกใช้รหัสแผ่ที่มีคุณสมบัติดังกล่าวทำให้คลื่นสัญญาณที่ส่งผ่านช่องสัญญาณมีคุณลักษณะที่ใกล้เคียงกับสัญญาณสุ่ม ดังนั้น การดักฟังจากผู้ที่ไม่ประสงค์ดีย่อมไม่เกิด เพราะหากไม่ทราบรหัสที่ใช้ในการแผ่แล้วก็จะไม่สามารถดึงบิตข้อมูลที่แท้จริงออกมาได้ ฉะนั้นจะมีเพียงภาครับที่ทราบรหัสแผ่ที่ใช้ในการสเปรดสเปกตรัม

ระบบสเปกตรัมแบบแผ่จะมีลักษณะดังต่อไปนี้ คือ

1. แบนด์วิดธ์ที่ใช้ในการส่งสัญญาณ จะต้องมากกว่าแบนด์วิดธ์ที่น้อยที่สุดที่ต้องการในการส่งสัญญาณข้อมูลข่าวสารอยู่มากๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

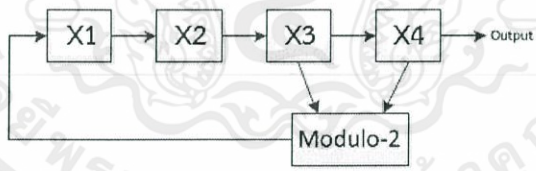
2. การกระจายแบนด์วิดธ์ของสัญญาณจะทำได้โดยการมอดูเลต (Modulate) สัญญาณข้อมูลข่าวสารกับสัญญาณสเปรดดิง (Spreading signal) หรือเรียกอีกอย่างหนึ่งว่า สัญญาณรหัส (Code) และสัญญาณรหัสจะต้องไม่ขึ้นกับสัญญาณข้อมูลข่าวสาร

3. ที่เครื่องรับจะสามารถทำการดึงสัญญาณกลับคืนมาโดยการใช้คอร์รีเลท (Correlate) ของสัญญาณที่รับได้กับสัญญาณรหัสที่ใช้ในการกระจายข้อมูลข่าวสารทางเครื่องส่ง

การมอดูเลตแบบเอฟเอ็ม (Frequency Modulation) พีซีเอ็ม (Pulse Code Modulation) เป็นการกระจายสัญญาณข้อมูลข่าวสารจริงแต่ก็ไม่เหมาะสมกับระบบสเปกตรัมแบบแผ่เพราะว่าไม่ได้สอดคล้องกับเงื่อนไขที่กล่าวมาข้างต้น

2.2 ลำดับของชิฟรีจิสเตอร์ (Shift register) [2]

พิจารณาชิฟรีจิสเตอร์ที่มีการป้อนกลับ (Feedback) อย่างเป็นเชิงเส้นดังรูปที่ 2.1 ซึ่งเป็นชิฟรีจิสเตอร์ 4 สเตจ (Stage) โดยจะทำหน้าที่เป็นตัวเก็บข้อมูลและเลื่อนข้อมูล ส่วนโมดูโล-2 (Modulo-2) จะทำหน้าที่บวกแล้วป้อนกลับไปเป็นอินพุตของชิฟรีจิสเตอร์ การทำงานของชิฟรีจิสเตอร์จะถูกควบคุมโดยสัญญาณนาฬิกา (Clock signal) โดยจะทำให้มีการเลื่อนข้อมูลจากซ้ายไปขวา โดยข้อมูลที่อยู่ที่สเตจ 3 จะถูกส่งไปให้สเตจ 4 แล้วข้อมูลในสเตจ 3 และสเตจ 4 จะถูกนำมาบวกกันโดยใช้โมดูโล-2 แล้วก็ถูกป้อนกลับไปที่สเตจ 1 เอาท์พุตก็คือการดึงข้อมูลออกมาจากสเตจ 4 [4]



รูปที่ 2.1 ตัวอย่างของชิฟรีจิสเตอร์ที่มีการป้อนกลับแบบเชิงเส้น

โดยสมมติให้ ขั้นแรกสเตจ 1 มีค่าเป็น 1 ส่วนสเตจอื่นๆ มีค่าเป็น 0 หมด ดังนั้นค่าภายในรีจิสเตอร์ (Register) จะมีค่าเป็น 1 0 0 0 โดยดูจากรูปแล้วเราจะได้ เอาต์พุตของชิฟรีจิสเตอร์ในแต่ละสเตจ จะมีค่าเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1000	0100	0010	1001	1100	0110	1011	0101
1010	1101	1110	1111	0111	0011	0001	1000

โดยค่าชิฟรีจิสเตอร์ในสเตจสุดท้ายจะมีค่าเป็น 1000 ซึ่งมีค่าเดียวกัน กับในสเตจแรก ซึ่งจะเห็นว่ารีจิสเตอร์ จะซ้ำค่าเดิมโดยใช้สัญญาณนาฬิกา 15 ลูก เอาต์พุตที่ดึงออกมาจากสเตจ 4 ในแต่ละสัญญาณนาฬิกาจะได้ดังนี้

0 0 0 1 0 0 1 1 0 1 0 1 1 1 1

โดยค่าที่อยู่ซ้ายสุดจะเป็นบิตแรกที่ได้ออกมา ต่อไปก็จะทำการตรวจสอบคุณสมบัติแรนดอมเนสตามที่ได้กล่าวไปแล้ว โดยคุณสมบัติแรกคือบาลานซ์ โดยพิจารณาเอาท์พุตจะเห็นว่ามีดิจิต 0 อยู่ 7 ตัว มีดิจิต 1 อยู่ 8 ตัว ซึ่งสอดคล้องตามคุณสมบัติเมื่อพิจารณาคุณสมบัติรันที่ดิจิตเป็น 0 จากเอาต์พุตจะเห็นว่ามีจำนวนรันอยู่ 4 ครั้ง โดย 1 ใน 2 มีความยาวเป็น 1 และ 1 ใน 4 มีความยาวเป็น 2 ซึ่งสอดคล้องตามคุณสมบัติ และรันที่มีดิจิตเป็น 1 ก็มีลักษณะเช่นเดียวกัน ชิฟรีจิสเตอร์จะให้ค่าดิจิตออกมาขึ้นกับจำนวนของรีจิสเตอร์ เงื่อนไขเริ่มแรก ค่าลำดับของเอาต์พุตนั้นสามารถแบ่งเป็นความยาวสูงสุด (Maximal Length) และความยาวไม่สูงสุด (No Maximal Length) โดยที่มีความยาวสุดจะพิจารณาได้จากจำนวนของชิฟรีจิสเตอร์ ดังนั้นการซ้ำของเอาต์พุตจะต้องใช้จำนวนสัญญาณนาฬิกา เป็นจำนวน ρ โดย

$$\rho = 2^n - 1 \tag{2.1}$$

ซึ่งจากตัวอย่างก็เป็นลำดับของความยาวสูงสุด (Maximal Length Sequence) ถ้าการซ้ำกันของเอาต์พุต มีเกิดขึ้นน้อยกว่า $\rho = 2^n - 1$ จะถือว่าเป็นความยาวไม่สูงสุด ส่วนคุณสมบัติของคอร์รีเลชันจะแสดงให้เห็นในหัวข้อต่อไป

2.3 ฟังก์ชันพีเอ็นออโตคอร์รีเลชัน (PN Autocorrelation) [3]

ฟังก์ชันพีเอ็นออโตคอร์รีเลชัน $R_x(\tau)$ ของรูปคลื่น $x(t)$ ที่มีคาบเวลา T_0 สามารถแสดงได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$R_x(\tau) = \frac{1}{K}\left(\frac{1}{T_0}\right)\int_{-T_0/2}^{T_0/2} x(t)x(1-\tau)dt \qquad -\infty < \tau < \infty \tag{2.2}$$

โดยที่

$$K = \frac{1}{T_0}\int_{-T_0/2}^{T_0/2} x(t)^2\,dt \tag{2.3}$$

โดยที่ $x(t)$ แสดงรูปคลื่นรหัสพีเอ็น (PN code) ซึ่งเราจะเรียกค่าแต่ละพัลส์ของรหัสพีเอ็นว่า ชิพ (chip) โดยจากการนอมอลไลซ์ (Normalize) ฟังก์ชันออโต้คอร์รีเลชันด้วย ρ ซึ่งพิจารณาใน 1 คาบของพีเอ็น

$$R_x(\tau) = \frac{1}{\rho} \text{ (จำนวนของอกรีเมนต์-จำนวนของดิสอกรีเมนต์)} \tag{2.4}$$

นอมอลไลซ์ฟังก์ชันออนโตคอร์รีเลชันที่เป็นความยาวสูงสุด $R_x(t)$ จะแสดงได้ดังรูปที่ 2.2 ซึ่งจะเห็นได้ว่า $R_x(t) = 1$ อย่างไรก็ตามสำหรับการเลื่อนที่เป็นวงกลม ระหว่าง $x(t)$ และ $x(t-x)$ เมื่อ $1 < \pi < \rho$ ค่าของออโตคอร์รีเลชันจะมีค่าเป็น $-1/\rho$ สำหรับค่า ρ ที่มีค่ามากๆจะเสมือนกับว่าไม่มีการคอร์รีเลทสำหรับการเลื่อน โดยพิจารณาคุณสมบัติข้อที่ 3 โดยการพิจารณาเอาต์พุตของชิพรีจิสเตอร์โดยให้แถวแรกเป็นเอาต์พุตของรีจิสเตอร์ และแถวล่างเป็นเอาต์พุตที่สุดท้ายเลื่อนไปทางซ้าย

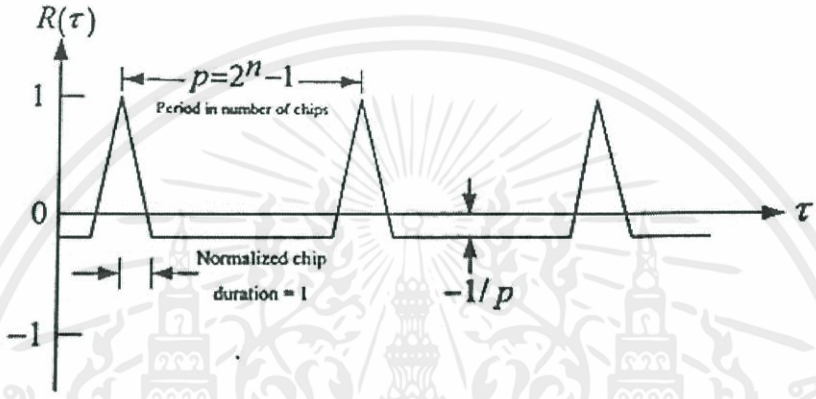
0	0	0	1	0	0	1	1	0	1	0	1	1	1	1
1	0	0	0	1	0	0	1	1	0	1	0	1	1	1
d	a	a	d	d	a	d	a	d	d	d	d	a	a	a

โดยให้อกรี (Agree) แทนด้วย a และดีสอกรี (Disagree) แทนด้วย d จากสมการที่ 2.4 ค่าของออโตคอร์รีเลชันของชิพจะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$R_x(\tau-1)=\frac{1}{\rho}(7-8)=-\frac{1}{15}$$

ซึ่งจะเห็นว่า ถ้าไม่มีการซิงโครไนซ์แล้วค่าของออโต้คอร์รีเลชันจะมีค่าเป็น $-1/\rho$ ซึ่งสอดคล้องกับคุณสมบัติที่ 3



รูปที่ 2.2 ฟังก์ชันของพีเอ็นออโตคอรีเลชั่น (PN autocorrelation)

2.4 รหัสพีเอ็น [4]

รหัสที่มีการใช้ในระบบ CDMA ได้แก่ รหัสพีเอ็นหรือที่ย่อมาจาก Pseudorandom Noise Code รหัสนี้ว่าเป็นรหัสที่มีลักษณะเหมือนสัญญาณรบกวน ทำให้ได้ชื่อส่วนหนึ่งว่า Noise ด้วย แต่เป็นสัญญาณรบกวนที่ถูกสร้างขึ้นด้วยกระบวนการสุ่มที่สามารถเขียนได้ทางคณิตศาสตร์ นั่นคือ ถึงแม้จะเป็นการสุ่มแต่รหัสจะวนกลับมาซ้ำเดิมเนื่องจากมีรูปแบบที่แน่นอนอธิบายได้ด้วยความสัมพันธ์ทางคณิตศาสตร์ ในช่วงต่อไปนี้ จะพิจารณาตัวอย่างรหัสพีเอ็นที่มีการนำมาใช้กัน ซึ่งเป็นรหัสพีเอ็นที่ประกอบด้วยบิต 0 หรือ 1 เท่านั้น

2.4.1 คุณสมบัติของรหัสพีเอ็น

1) Balance property เป็นคุณสมบัติซึ่งชุดรหัสพีเอ็นที่ถูกสร้างขึ้นมาจะมีจำนวนบิต 1 มากกว่า บิต 0 อยู่ 1 ค่า เสมอ

2) Run-length property เป็นคุณสมบัติซึ่งชุดรหัสพีเอ็นที่ถูกสร้างขึ้นมาในช่วงเวลาใดก็ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ครึ่งหนึ่งของชุดรหัส จะประกอบด้วยบิต 0 ที่ต่อเนื่องกัน (Consecutive 0s) หรือ บิต 1 ที่ต่อเนื่องกัน (Consecutive 1s) มีความยาวเป็น 1

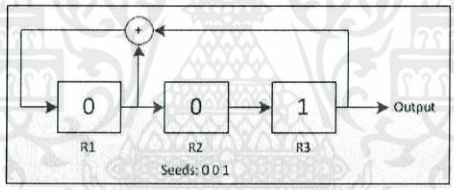
- หนึ่งในสี่ของชุดรหัส จะประกอบด้วยบิต 0 ที่ต่อเนื่องกัน หรือบิต 1 ที่ต่อเนื่องกันมีความยาวเป็น 2

- หนึ่งในแปดของชุดรหัส จะประกอบด้วยบิต 0 ที่ต่อเนื่องกัน หรือบิต 1 ที่ต่อเนื่องกันมีความยาวเป็น 3

- ต่อเนื่องเช่นนี้ไปเรื่อยๆ

3) Autocorrelation property กล่าวว่ารหัสชุดเดียวกันที่ถูกเลื่อนบิตออกไปจะมีความสัมพันธ์ (Autocorrelation) กับรหัสเดิมน้อย หรืออาจจะกล่าวได้ว่าไม่ให้ข้อมูลที่เกี่ยวกับรหัสเดิม

เพื่อให้เกิดความเข้าใจเกี่ยวกับรหัสพีเอ็นจะขอตัวอย่างขอยกตัวอย่าง โดยการสร้าง รหัสพีเอ็นที่ได้จากการอาศัยชิฟต์รีจิสเตอร์ XOR gate ดังรูปที่ 2.3

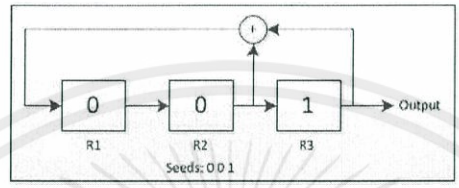


รูปที่ 2.3 การสร้างรหัสพีเอ็นโดยอาศัยชิฟต์รีจิสเตอร์และ XOR gate

จากรูปเครื่องหมาย แทนตัวชิฟต์รีจิสเตอร์ซึ่งมีบิตเริ่มต้น (Seed) เป็น 0 ในขณะที่เครื่องหมายแทน XOR gate ซึ่งให้ผลลัพธ์เป็นบิต 1 เมื่อ Input ทั้งสองต่างกันเท่านั้น ที่ด้าน Output Digit เป็นบิตของรหัสพีเอ็นที่ถูกสร้างขึ้น ซึ่งจะมีความยาวต่างๆกัน โดยการใช้ชิฟต์รีจิสเตอร์และ XOR gate ถ้าหากใช้ชิฟต์รีจิสเตอร์เป็นจำนวน N ตัว จะทำให้ได้ชุดรหัสที่มีความยาว (Length) เป็น $2^N - 1$ ซึ่งเป็น Maximum Length หรือเป็นรหัสที่ยาวที่สุดที่ยังไม่มีการวนซ้ำ ดังเช่นกรณีของตัวอย่างรหัสพีเอ็นที่ได้มีการใช้ชิฟต์รีจิสเตอร์ จำนวน 3 ชุด ทำให้ได้ชุดรหัสที่มี Length เป็น 7 หรือ $2^3 - 1$ นั่นเอง ดังนั้นหากเราต้องการรหัสพีเอ็นที่มีความยาวมาก ก็จะต้องใช้ชิฟต์รีจิสเตอร์เป็นจำนวนที่มากขึ้นด้วย

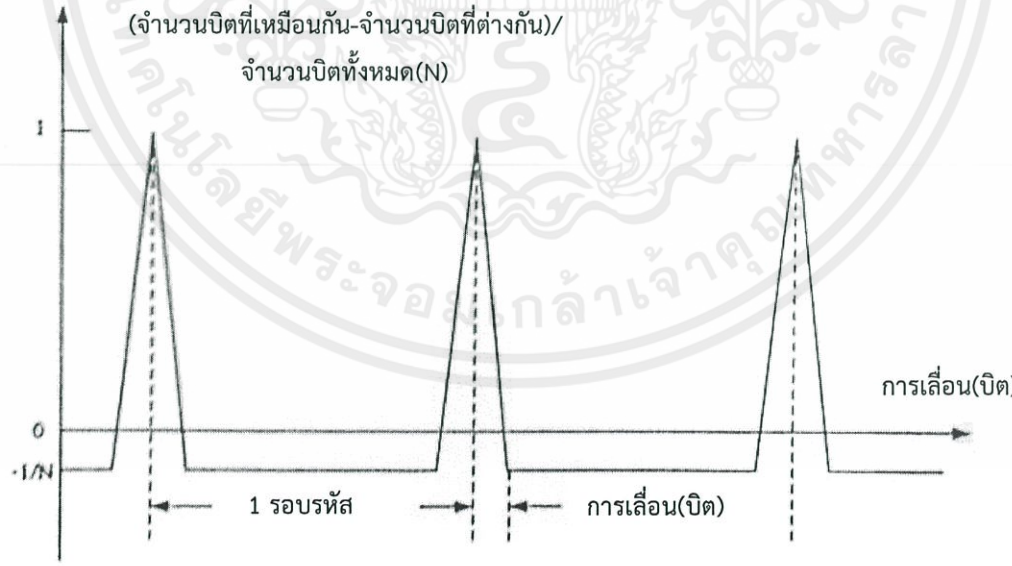
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.3 แสดงถึงลักษณะวงจรชิฟต์รีจิสเตอร์ที่ใช้สร้างรหัสพีเอ็นหรือกล่าวอีกนัยหนึ่งก็คือการระบุถึงชุดรหัสพีเอ็นซึ่งแทนด้วยชุดรหัสที่มีสมการพหุนามเป็น $X^3 + X + 1$ อีกตัวอย่างหนึ่งของชุดรหัสที่มี Polynomial เป็น $X^3 + X^2 + 1$ ซึ่งให้บิตรหัสเป็น 1 0 0 1 1 1 0 แสดงได้ดังรูปที่ 2.4



รูปที่ 2.4 รหัสพีเอ็น ที่มีสมการพหุนาม คือ $X^3 + X^2 + 1$

คุณสมบัติพิเศษของรหัสพีเอ็นก็คือคุณสมบัติในข้อที่ 3 คือ Autocorrelation property ที่กล่าวว่ารหัสชุดเดียวกันที่ถูกเลื่อนบิตออกไป จะมีความสัมพันธ์ (Autocorrelation) กับรหัสเดิมน้อย หรืออาจกล่าวได้ว่าไม่ให้ข้อมูลที่เกี่ยวกับรหัสเดิม ซึ่งสามารถอธิบายด้วยกราฟความสัมพันธ์ในรูปที่ 2.5



รูปที่ 2.5 Autocorrelation ของรหัส PN Code

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.5 จะพบว่าการเลื่อน (Shift) ออกไปเพียง 1 บิต ก็จะทำให้ค่า Autocorrelation ซึ่งในที่นี้แทนด้วยความสัมพันธ์ (จำนวนบิตที่เหมือนกัน-จำนวนบิตที่แตกต่างกัน / จำนวนบิตทั้งหมด (N)) มีค่าน้อยมาก นั่นคือ ไม่ให้ข้อมูลเกี่ยวกับชุดรหัสเลย ดังนั้นถึงแม้เราจะรู้ว่าข้อมูลที่ถูกส่งมาถูกเข้าด้วยรหัสพีเอ็นชุดใด แต่ถ้าหากเราไม่รู้ว่ารหัสเริ่มต้น ณ เวลาใด ก็จะไม่สามารถถอดรหัสออกมาได้ การถอดรหัสถึงแม้จะใช้รหัสพีเอ็นชุดที่ถูกต้อง แต่เลื่อนไปแม้แต่บิตเดียวก็จะไม่ได้ข้อมูลที่ถูกต้องกลับมาด้วยเหตุนี้รหัสรหัสพีเอ็นจึงถูกนำมาใช้กับระบบ CDMA เนื่องจากความปลอดภัยในการเข้ารหัสสูง

การใช้รหัสพีเอ็นอยู่สองแบบคือ Short PN Code และ Long PN Code รหัสทั้งสองจะมีความยาวต่างกันและถูกใช้เพื่อจุดประสงค์ที่แตกต่างกัน

2.4.1.1 Short PN Code

Short PN Code ถูกใช้เพื่อจุดประสงค์เพื่อใช้แยกแยะว่าเป็น Cell ใดและ Sector ใดและ Cell ในแต่ละ Cell และแต่ละ Sector จะใช้ Short PN code ชุดเดียวกัน แต่มีจุดเริ่มต้นที่เวลาต่างกัน หรือมีค่า Offset ต่างกันนั่นเอง โดยอาศัยคุณสมบัติ Autocorrelation ของรหัสพีเอ็นทำให้ด้านรับคือเครื่องโทรศัพท์เคลื่อนที่สามารถแยกแยะได้ว่ากำลังติดต่ออยู่กับที่ใดโดยไม่สับสนและรบกวนกัน ดังนั้นอาจกล่าวได้ว่ารหัสพีเอ็นที่ใช้เปรียบเสมือนกับหมายเลขประจำ Cell หรือ Sector (Cell/Sector ID) นั่นเอง

2.4.1.2 Long PN Code

Long PN Code ถูกใช้เพื่อวัตถุประสงค์รักษาความปลอดภัย (Privacy) เพื่อป้องกันการลักลอบดักฟังและนำข้อมูลไปถอดรหัสออกมา โดยอาศัยความยาวของรหัสที่ยาวมากจึงทำให้การลักลอบดักฟังเพื่อถอดรหัสไม่สามารถทำได้ในทางปฏิบัติสมการพหุนามของ Long PN Code คือ

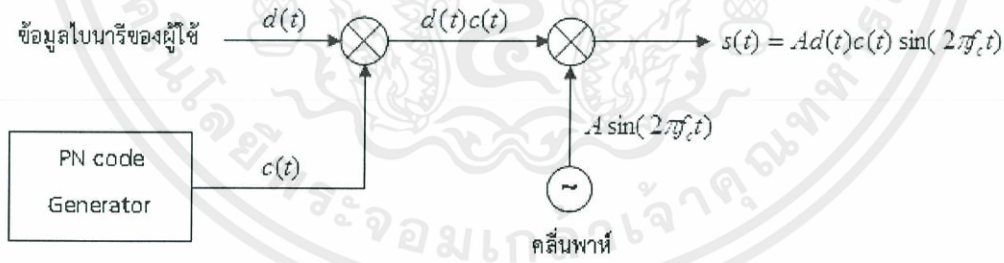
$$P_L(X) = X^{42} + X^{35} + X^{33} + X^{31} + X^{27} + X^{26} + X^{25} + X^{22} + X^{21} + X^{19} + X^{18} + X^{17} + X^{16} + X^{10} + X^7 + X^6 + X^5 + X^3 + X^2 + X + 1$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จาก Polynomial นี้จะพบว่า Long PN Code จะมีรหัสที่มีความยาวถึง $2^{42}-1=4.398\times10^{12}$ เนื่องจากอัตราการสร้างรหัสนี้มีความยาวเท่ากับ 1.228 Mbps ดังนั้นรหัสนี้จะมีการซ้ำทุกๆ 41.425 วัน ซึ่งทำให้การถอดรหัสไม่สามารถทำได้ในทางปฏิบัติ เพราะการที่จะดักฟังรหัสวนกลับมาซ้ำเพื่อแกะรูปแบบของรหัสนั้นต้องดักฟังนานถึง 41 วัน

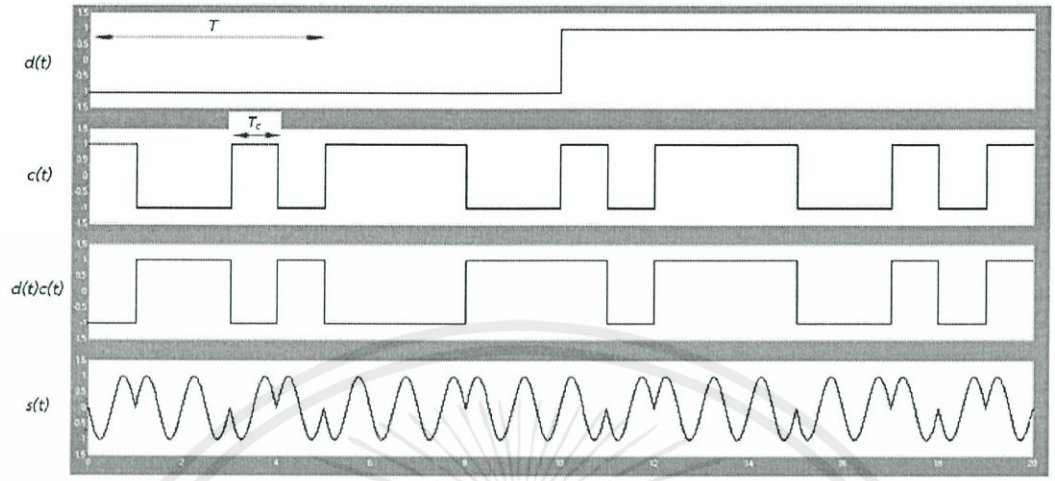
2.5 วงจรภาคส่งสัญญาณ CDMA [6]

โครงสร้างของวงจรภาคส่งสัญญาณ CDMA ของระบบที่ใช้การมอดูเลตแบบ BPSK (Binary Phase Shift Keying) ดังรูปที่ 2.6 แสดงรายละเอียดโครงสร้างของเครื่องส่งสัญญาณ CDMA ของผู้ใช้รายหนึ่ง ซึ่งในแผนภาพการทำงานประกอบด้วย 2 ส่วนสำคัญ คือ วงจรคูณสัญญาณข้อมูลไบนารี่กับรหัสแผ่ และวงจรมอดูเลเตอร์แบบ BPSK สัญญาณขาเข้า $d(t)$ ซึ่งมีอัตราของบิตข้อมูลที่ต่ำโดยให้สัญญาณนี้มีคาบเวลาเป็น T ถูกนำไปคูณกับรหัสแผ่ $c(t)$ ที่มีคาบเวลาของหนึ่งชิพเท่ากับ T_c ซึ่งสั้นกว่า T มาก ขั้นตอนนี้เรียกว่าการสเปรดสเปกตรัมของสัญญาณข้อมูลจากเดิมที่มีอัตราของบิตข้อมูลต่ำซึ่งใช้แบนด์วิดธ์ของช่องสัญญาณขนาดเล็กในการส่งข้อมูลที่มีอัตราชิพที่สูงขึ้นกว่าเดิมมาก กล่าวคือต้องใช้แบนด์วิดธ์ของช่องสัญญาณในการส่งที่ใหญ่ขึ้นมากด้วย จากนั้นก็นำสัญญาณ $d(t)c(t)$ ที่ได้นี้ไปทำการมอดูเลตแบบ BPSK โดยใช้คลื่นพาห์ความถี่สูง จากรูปที่ 2.7 แสดงตัวอย่างของสัญญาณที่ได้ในแต่ละขั้นตอน โดยในตัวอย่างนี้ได้กำหนดให้ $T=5T_c$ และคลื่นพาห์มีความถี่เท่ากับ $f_c=1/T_c$



รูปที่ 2.6 วงจรภาคส่งของระบบ CDMA ที่ใช้การมอดูเลตแบบ BPSK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 ตัวอย่างของสัญญาณในวงจรภาคส่งของระบบไดเรกต์ซีเควนซ์ที่ใช้ BPSK

2.6 วงจรภาครับของสัญญาณ CDMA

องค์ประกอบของภาครับระบบไดเรกต์ซีเควนซ์สามารถแสดงได้ดังรูปที่ 2.8 ซึ่งมีการใช้มอดูเลตแบบ BPSK วงจรนี้ใช้การดึงสัญญาณข้อมูล $d(t)$ ออกมาจากสัญญาณ RF ที่รับได้ ซึ่งโดยปกติแล้วสัญญาณนี้จะมีการประวิงเวลาเนื่องจากการแพร่สัญญาณ (propagation delay) ดังนั้นสัญญาณที่รับได้สามารถเขียนได้ดังสมการที่ 2.4

$$s(t-\tau) = d(t-\tau)c(t-\tau)A\sin(2\pi f_c(t-\tau)) \tag{2.4}$$

ซึ่งค่า τ เป็นค่าประวิงเวลา สามารถอธิบายได้ด้วยรูปที่ 2.4 โดยสัญญาณ $s(t-\tau)$ นี้ถูกนำไปคูณกับรหัสแผ่ชุดเดียวกันกับที่ใช้ในภาคส่ง โดยที่วงจรภาครับต้องสามารถหาค่าประมาณเวลาประวิงให้ใกล้เคียงกับค่า τ มากที่สุด ผลที่ได้จากการคูณของสัญญาณทั้งสอง คือ

$$c(t-\tau)s(t-\tau) = d(t-\tau)A\sin(2\pi f_c(t-\tau)) \tag{2.5}$$

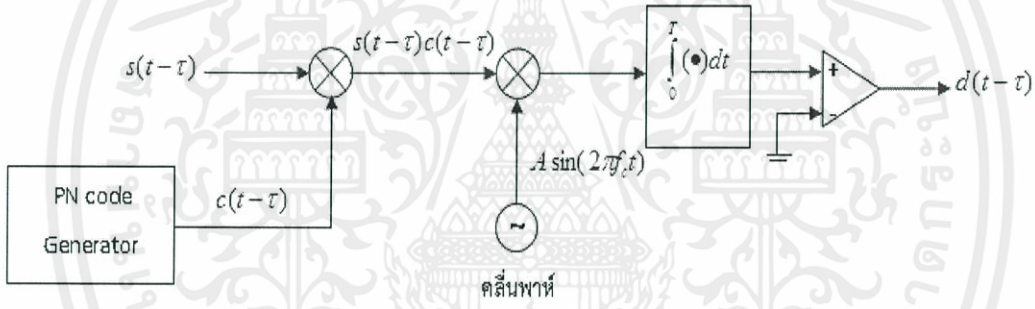
ทั้งนี้เพราะ $c(t-\tau)c(1-\tau) = 1$ ณ ที่เวลา t ใดๆ คือ การดีสเปรดของสัญญาณข้อมูลที่ต้องการออกมา จากนั้นเมื่อนำสัญญาณนี้ไปเข้าวงจรดีมอดูเลเตอร์แบบ BPSK ก็จะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณข้อมูลกลับคืนมา ในวงจรดีมอดูเลเตอร์ประกอบด้วย 2 ส่วนหลักคือ วงจรคูณและวงจรอินทิเกรเตอร์ สัญญาณที่ได้หลังจากผ่านวงจรคูณแล้วสามารถแสดงได้ ดังสมการที่ 2.6

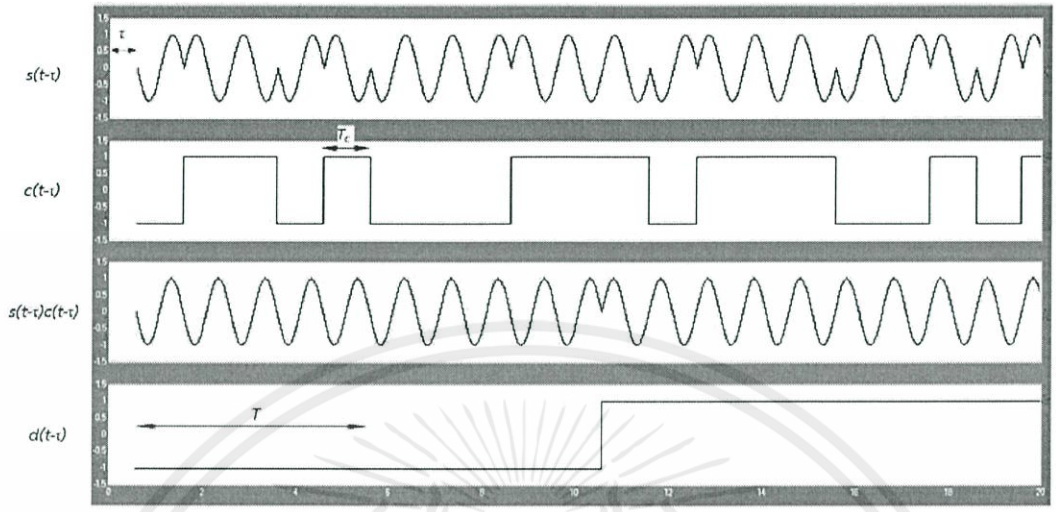
$$\begin{aligned} & d(t-\tau)A\sin(2\pi f_c(t-\tau))\sin(2\pi f_c(t-\tau)) \\ & = d(t-\tau)A\{1-\cos(4\pi f_c(t-\tau))\}/2 \end{aligned} \tag{2.6}$$

เมื่อนำสัญญาณเข้าวงจรอินทิเกรเตอร์ที่ทำการอินทิเกรตใหม่ทุกๆคาบเวลา T ซึ่งโดยปกติแล้วจะมีช่วงเวลายาวกว่าคาบเวลา T_c มาก ดังนั้น สัญญาณ $\sin(4\pi f_c(t-\tau))$ เมื่อถูกอินทิเกรตแล้วก็จะมีค่าใกล้ศูนย์มาก เพราะฉะนั้นสัญญาณที่เหลือจากวงจรอินทิเกรตจึงมีเพียงสัญญาณ $d(t-\tau)$ ซึ่งเมื่อนำมาเข้าวงจรเปรียบเทียบแล้วก็จะได้ข้อมูลเดิมที่มีค่าเป็น 0 และ 1 กลับคืนมา



รูปที่ 2.8 วงจรภาครับของระบบไดเรกซ์ซีเควนซ์ที่ใช้การมอดูเลตแบบ BPSK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 ตัวอย่างของสัญญาณในวงจรภาครับของระบบไดเรกต์ซีเควนซ์ที่ใช้ BPSK

2.7 คุณลักษณะเฉพาะของระบบ CDMA

ระบบ CDMA มีคุณสมบัติเฉพาะตัวที่ต่างไปจากระบบ TDMA หรือ FDMA ซึ่งสามารถสรุปแยกออกเป็นข้อได้ ดังนี้

2.7.1 Soft Capacity

CDMA ความจุถูกกำหนดโดยปริมาณสัญญาณแทรกสอด ดังนั้นจึงเรียกว่าเป็นระบบที่มีความจุแบบซอฟต์ (Soft Capacity) ซึ่งหมายความว่า ระบบสามารถรองรับผู้ใช้เพิ่มได้อีกเรื่อยๆ ตราบใดที่ปริมาณแทรกสอดในระบบยังไม่อยู่ในระดับที่ไม่สูงเกินไป แต่ในกรณีของ FDMA และ TDMA ความจุของระบบถูกกำหนดโดยตรงจากขนาดของสเปกตรัมแบนด์วิดธ์ที่มีอยู่ จะเป็นระบบที่มีความจุแบบฮาร์ด (Hard Capacity) ซึ่งหมายความว่าระบบไม่สามารถรองรับการร้องขอใช้งานของผู้ใช้รายใหม่เพิ่มเติมได้อีกเลย ถ้าหากช่องสัญญาณที่มีอยู่ได้ถูกใช้งานทั้งหมดแล้ว ทั้งนี้การรับผู้ใช้รายใหม่ของระบบ CDMA แต่ละครั้งจะส่งผลให้ระบบโดยรวมมีปริมาณสัญญาณแทรกสอดเพิ่มมากขึ้นทีละน้อย และทำให้คุณภาพการใช้งานของผู้ใช้ที่มีอยู่เดิมแย่ลงเป็นลำดับ

2.7.2 Multipath Resolution

ในระบบการส่งข้อมูลที่อัตราการส่งสูงขึ้น เช่น เกินกว่า 10 kbps มากๆ ภายใต้สภาพช่องสัญญาณที่มีปรากฏการณ์ของคลื่นพหุวิถี (Multipath) นั้น ระบบ FDMA และ TDMA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะได้รับผลกระทบอย่างรุนแรงจากปัญหาการแทรกสอดระหว่างสัญลักษณ์ (Intersymbol Interference) สืบเนื่องจากดีเลย์สเปรด (Delay Spread) ที่เกิดขึ้นดังนั้น จำเป็นที่ภาครับสัญญาณต้องอาศัยวงจรอีควอไลเซอร์ที่มีความซับซ้อนสูงในการลดผลกระทบที่เกิดขึ้น ในทางกลับกับวงจรภาครับของระบบ CDMA ไม่มีความจำเป็นต้องใช้วงจรอีควอไลเซอร์สำหรับการปรับแก้รูปสัญญาณ เพราะการทำสเปรดสเปกตรัมกับสัญญาณกลับช่วยทำให้วงจรภาครับสามารถแยกแยะคลื่นสัญญาณของแต่ละวิถีได้ดีขึ้นทั้งยังสามารถนำมารวมกันโดยใช้วงจรเครื่องรับ Rake เพื่อให้ได้เป็นสัญญาณที่มีคุณภาพมากขึ้นได้อีกด้วย ความสามารถในการนำสัญญาณจากแต่ละวิถีมาผนวกกันนั้นถือว่าเป็นการเพิ่มไดเวอร์ซิตีของระบบ นอกจากนี้ อุปกรณ์หลักของเครื่องรับประกอบด้วยวงจรคอร์รีเลเตอร์ (Correlator) ซึ่งทำหน้าที่ในการดีสเปรดสัญญาณกลับมา ซึ่งในทางปฏิบัติวงจรคอร์รีเลเตอร์จัดว่ามีความซับซ้อนน้อยกว่าวงจรอีควอไลเซอร์

2.7.3 Sectorization for Capacity

ในระบบ FDMA และ TDMA มีการแบ่งเซลล์ออกเป็นเซกเตอร์ย่อยเพื่อลดปริมาณของสัญญาณแทรกสอดช่องสัญญาณร่วม (Co-Channel Interference) ทำให้สัญญาณที่รับได้มีคุณภาพดีขึ้น และระบบสามารถนำความถี่ชุดเดิมมาใช้ซ้ำได้มากขึ้น ผลที่ได้คือในแต่ละเซลล์ระบบสามารถรองรับผู้ใช้ได้มากขึ้น อย่างไรก็ตามหากวิเคราะห์ในเชิงทราฟฟิกกลับพบกว่าการแบ่งเซกเตอร์ส่งผลเชิงลบต่อประสิทธิภาพ การใช้งานช่องสัญญาณ นั่นคือ โดยรวมระบบจะมีความจุลดลง และจะยิ่งลดลงเมื่อมีการแบ่งเซกเตอร์จำนวนมากขึ้น ฉะนั้นผลของการแบ่งเซกเตอร์ในระบบ FDMA หรือ TDMA จึงมีข้อดีและก่อให้เกิดผลเสียพร้อมกัน การแบ่งเซกเตอร์ในระบบ CDMA ช่วยทำให้สัญญาณแทรกสอดมีปริมาณลดลง ส่งผลให้ระบบได้ความจุเพิ่มขึ้น โดยหากมีการแบ่งเซกเตอร์ในระบบ CDMA จำมีแต่ให้ประโยชน์ โดยไม่ก่อให้เกิดผลข้างเคียงอื่นๆ แต่อย่างใด ดังนั้นการใช้ประโยชน์จากสเปกตรัมความถี่ที่มีอยู่อย่างจำกัดของระบบ CDMA จึงมีประสิทธิภาพที่ดีกว่าให้ขนาดความจุที่สูงกว่าและรองรับจำนวนผู้ใช้บริการได้มากกว่า

2.7.4 No Guard Time Required

ในระบบ TDMA มีความจำเป็นที่จะต้องเว้นช่องว่างที่เรียกว่า ช่องเวลาป้องกัน (Guard time) ไว้ขนาดหนึ่งเพื่อป้องกันมิให้ข้อมูลที่อยู่คนละไทม์สล็อตเกิดการทับกัน ส่วนระบบ CDMA นั้นไม่มีความจำเป็นที่ต้องมีช่วงเวลาป้องกันเลยเพราะฉะนั้นช่วงเวลาตรงนี้สามารถนำมาใช้เพิ่มประสิทธิภาพการส่งข้อมูลให้สูงขึ้นได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 ทฤษฎีการชักค่าตัวอย่าง (Sampling Theorem) [5]

สัญญาณที่เกิดขึ้นอย่างต่อเนื่องตลอดเวลาที่เรียกกันว่า สัญญาณอนาล็อก (Analog Signal) นั้น ถ้าค่าความหนาแน่นสเปกตรัมของมันเกิดอยู่ในแบนด์วิดธ์ที่จำกัดแล้ว เราจะพบว่าในโดเมนเวลานั้น สัญญาณอนาล็อกที่มีคุณสมบัติดังกล่าวจะมีข้อมูลที่มีความซ้ำซ้อนเกินความจำเป็นรวมอยู่ด้วยมากในระหว่างเวลาส่งสัญญาณนั้น ดังนั้นถ้าเราเลือกค่าชักตัวอย่าง (Sampling) ของสัญญาณนั้นที่ขณะเวลาต่างๆที่อยู่ห่างกันอย่างเหมาะสมแล้ว เพียงจากค่าตัวอย่างของสัญญาณที่ชักออกมาได้นี้ก็สามารถเก็บข้อมูลของสัญญาณทั้งหมดได้อย่างสมบูรณ์ ซึ่งกล่าวได้ว่าการเลือกชักค่าตัวอย่างสัญญาณที่เวลาที่เหมาะสมจะทำให้เราสามารถลดความซ้ำซ้อนหรือความฟุ่มเฟือยของการใช้ค่าสัญญาณลงได้ เพราะฉะนั้นเราจึงสามารถนำเอาช่วงเวลาที่ไม่ตรงกับขณะเวลาที่มีการชักค่าสัญญาณไปใช้ในการทำประโยชน์อย่างอื่นได้และเราสามารถใช้เพียงข้อมูลเท่าที่เราชักค่ามาได้ซึ่งขึ้นอยู่กับค่าตัวอย่างของสัญญาณอนาล็อกที่ชั่วขณะเวลาที่ทำการชักค่าสัญญาณนั้นไปจัดการประมวลผลตามต้องการได้สะดวก เนื่องจากค่าตัวอย่างที่ชักออกมานั้นเกิดที่เวลาห่างกันเป็นห้วงๆ ไม่ติดต่อกัน เราจึงเรียกค่าสัญญาณที่เกิดจากการชักตัวอย่างสัญญาณนี้ว่าสัญญาณดิสครีตทางเวลา หรือ นิยมเรียกสั้นๆว่า สัญญาณดิสครีต (Discrete Signal) ทฤษฎีสำคัญที่บอกความสัมพันธ์ในการกำหนดคาบเวลาสำหรับการชักค่าตัวอย่างสัญญาณอนาล็อกเพื่อสร้างสัญญาณดิสครีตที่มีข้อมูลสำหรับสัญญาณอนาล็อกเดิมอยู่ครบถ้วนคือ ทฤษฎีการชักตัวอย่าง (Sampling Theorem) ซึ่งมีใจความต่อไปนี้

ถ้าสัญญาณใดๆที่มีค่าฟังก์ชันมีความหนาแน่นสเปกตรัมอยู่ในช่วงความถี่ที่มีแบนด์วิดท์ไม่เกิน B เฮิรตซ์แล้ว การชักค่าตัวอย่างของสัญญาณนั้นที่ขณะเวลาที่ห่างกันอย่างสม่ำเสมอเป็นคาบไม่เกิน $\frac{1}{2B}$ วินาที จะทำให้สามารถกำหนดค่าสัญญาณอนาล็อกเดิมกลับคืนมาได้จากค่าตัวอย่างสัญญาณเหล่านี้ เมื่อลองพิจารณาเกี่ยวกับทฤษฎีการชักตัวอย่างนี้ให้ละเอียดขึ้นโดยการสมมติว่าสัญญาณอนาล็อก $f(t)$ เป็นสัญญาณที่มีความหนาแน่นเชิงสเปกตรัมอยู่ไม่เกินความถี่ B เฮิรตซ์ ดังแสดงในรูปที่ 2.10 ถ้าเราทำการชักค่าตัวอย่างของ $f(t)$ ที่ทุกขณะเวลาที่ห่างกัน T วินาที โดยกำหนดให้สัญญาณที่เกิดจากการชักค่าตัวอย่างของ $f(t)$ นี้คือ $f_s(t)$ แล้วจะทำการอธิบายในเชิงคณิตศาสตร์ได้ว่า

$$f_s(t) = f(t)\delta_T(t) \tag{2.7}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

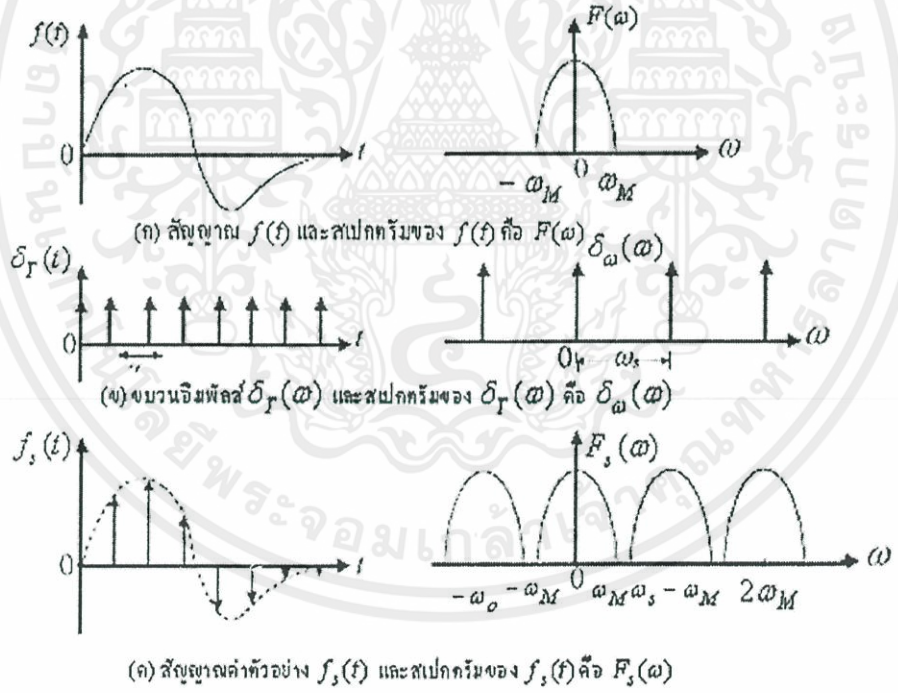
โดยในที่นี้ $\delta_T(t)$ คือขบวนอิมพัลส์ (Impulse Train) ที่มีการนิยามว่า

$$\delta_T(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \tag{2.8}$$

โดยอาศัยคุณสมบัติของฟังก์ชันอิมพัลส์ดังสมการที่ 2.7 ทำให้สามารถเขียนสมการที่ 2.8 ได้เป็น

$$f_s(t) = \sum_{n=-\infty}^{\infty} f(nT)\delta(t - nT) \tag{2.9}$$

สมการที่ 2.9 แสดงให้เห็นแน่นอนว่า $f_s(t)$ คือสัญญาณ $f(t)$ ที่มีค่าเพียงแต่ที่ขณะเวลา $t = nT, (n = 1,2,3,...)$ เท่านั้น



รูปที่ 2.10 สเปกตรัมของสัญญาณที่ถูกชักค่าตัวอย่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่การแปลงฟูเรียร์เบื้องต้นของขบวนอิมพัลส์มีรูปดังต่อไปนี้ คือ

$$\delta_T(t) \leftrightarrow \delta\omega_s(\omega)\omega_s \tag{2.10}$$

โดยที่ $\omega_s = \frac{2\pi}{T}$

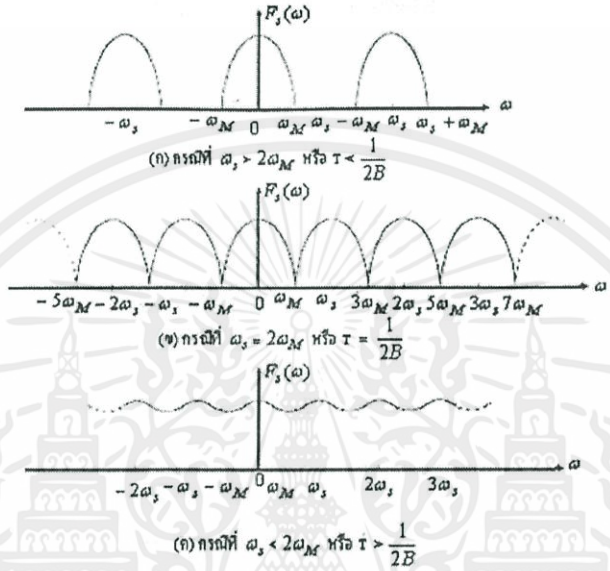
โดยอาศัยคุณสมบัติตามสมการที่ 2.9 ถ้ากำหนดให้ $f_s(t) \leftrightarrow F_s(\omega)$ และ $f(t) \leftrightarrow F(\omega)$ จะได้

$$\begin{aligned} F_s(\omega) &= \frac{1}{2\pi} F(\omega) \otimes \delta\omega_s(\omega)\omega_s \\ &= \frac{\omega_s}{2\pi} \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} F(z)\delta(\omega - n\omega_s - z)dz \\ &= \frac{1}{T} \sum_{n=-\infty}^{\infty} F(\omega - n\omega_s) \end{aligned} \tag{2.11}$$

จากสมการที่ 2.11 จะทำให้รู้ว่า $f_s(t)$ หรือสัญญาณดิสครีตที่เกิดจากการชักตัวอย่างของ $f(t)$ จะมีฟังก์ชันความหนาแน่นเชิงสเปกตรัม ที่มีรูปร่างเหมือนฟังก์ชันความหนาแน่นเชิงสเปกตรัมของสัญญาณ $f(t)$ แต่จะเกิดอยู่ซ้ำกันทุกคาบความถี่ ω_s ดังแสดงในรูปที่ 2.10 เมื่อ $f(t)$ มีองค์ประกอบของความถี่สูงสุด คือ B เฮิรตซ์ ค่าความถี่เชิงมุมที่ตรงกันกับค่าความถี่สูงสุดนี้จะมีค่า $\omega_M = 2\pi B$ เรเดียนต่อวินาที จะพบว่าถ้า $\omega_s > 2\omega_M$ หรือ $T < \frac{1}{2B}$ แล้ว $F_s(\omega)$ จะมีลักษณะเหมือนกับการเอา $F(\omega)$ มาเรียงกันห่างๆบนแกนความถี่ ω ดังแสดงในรูปที่ 2.11ก แต่เมื่อ $\omega_s = 2\omega_M$ หรือ $T = \frac{1}{2B}$ จะพบว่า $F_s(\omega)$ มีลักษณะเหมือนกับการเอา $F(\omega)$ มาเรียงชิดๆติดกันพอดีบนแกนความถี่ ω ดังแสดงในรูปที่ 2.11ข และถ้า $\omega_s < 2\omega_M$ หรือ $T > \frac{1}{2B}$ แล้วจะทำให้ $F_s(\omega)$ มีลักษณะที่เกิดจากผลรวมของ $F(\omega)$ ที่มาเรียงเหลื่อมทับกันดังแสดงในรูปที่ 2.11ค ซึ่งจะทำให้เกิดผลเทียบเท่ากับการพับกลับ (Fold Over) ขององค์ประกอบของสัญญาณที่มีความถี่สูงเกิน $\frac{\omega_s}{2}$ ซึ่งจะกลับมาทับกับองค์ประกอบของสัญญาณ $F_s(\omega)$ ที่มีความถี่ต่ำและรวมตัวกันจึงทำให้มีสเปกตรัมผิดเพี้ยนไปจากเดิม ปรากฏการณ์ที่เกิดการเหลื่อมทับ หรือ การพับกลับของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบของสเปกตรัมดังกล่าวนี้มีคำศัพท์เรียกเฉพาะว่า การเกิดการซ้อนทับของสเปกตรัม (Aliasing)



รูปที่ 2.11 สเปกตรัมของสัญญาณที่เกี่ยวเนื่องกับรูปที่ 2.10 เมื่อใช้ช่วงเวลาในการสุ่มค่าสัญญาณต่างๆ

จากรูปที่ 2.11 จะเห็นว่าหากคาบเวลาของการชักค่าตัวอย่างสัญญาณ T น้อยกว่า $\frac{1}{2B}$ วินาทีแล้วเราจะสามารถใช้วงจรกรองความถี่มาแยกเอาส่วนของสเปกตรัมเฉพาะส่วนที่มีความถี่ต่ำของ $F_s(\omega)$ ซึ่งเหมือนกับ $F(\omega)$ ออกมาได้โดยสะดวก เมื่อทำเช่นนี้ได้ก็เท่ากับว่าเราสามารถแยกหรือตรวจจับ(detect)เอา $f(t)$ จาก $f_s(t)$ ได้โดยใช้วงจรกรองความถี่เป็นอุปกรณ์ที่ใช้ช่วยแยกสัญญาณได้ แต่อย่างไรก็ตามในกรณีที่ T มากกว่า $\frac{1}{2B}$ แล้ว จะทำให้เกิดมีการซ้อนทับกันของส่วนประกอบความถี่ดังรูปที่ 2.11ค ทำให้เราไม่สามารถใช้วงจรกรองความถี่แยกเอา $F(\omega)$ จาก $F_s(\omega)$ ได้เลย การใช้รูปแบบทางคณิตศาสตร์มาช่วยวิเคราะห์ผลดังกล่าว ทำให้เราสามารถมองลึกลงไปได้ว่า ทำไมทฤษฎีการชักตัวอย่าง จึงกล่าวว่าคาบเวลาของการชักค่าตัวอย่างสัญญาณจึงต้องมีค่าไม่เกิน $\frac{1}{2B}$ วินาที และจะเห็นว่าในกรณีที่ $T=\frac{1}{2B}$ พอดี ฟังก์ชันความหนาแน่นเชิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สเปกตรัม $F_s(\omega)$ จะเกิดจาก $F(\omega)$ ที่มาเรียงชิดกันดังแสดงใน รูปที่ 2.11ข ทำให้เรารู้ว่าในกรณีนี้มีแต่วงจรกรองความถี่ในอุดมคติเท่านั้นที่จะสามารถแยกเอาแต่สเปกตรัมในช่วงความถี่ที่ต้องการ คือ $F(\omega)$ ออกมาได้ ปกติในทางปฏิบัติจะต้องมีการค่อยๆลดลงของค่าฟังก์ชันถ่ายโอนอยู่บริเวณใกล้จุดตัด(cutoff)ความถี่ ดังนั้นในทางปฏิบัติจึงมักจะเลือกใช้คาบเวลาการชักตัวอย่าง T ให้น้อยกว่า $\frac{1}{2B}$ เสมอ

ค่า $T=\frac{1}{2B}$ นี้มีความสำคัญในทางทฤษฎี เพราะมันเป็นค่าวิกฤตที่จะบอกว่าการชักค่าตัวอย่างสัญญาณนั้น จะมีผลกระทบต่อการแยกตรวจจับสัญญาณในภายหลังหรือไม่ ค่าคาบเวลา $T=\frac{1}{2B}$ นี้มีชื่อเฉพาะเรียกว่า ช่วงเวลาการชักตัวอย่างไนควิตซ์ หรือบางครั้งก็นิยมเรียกสั้นๆว่า ช่วงเวลาไนควิตซ์ (Nyquist interval) และค่าความถี่ $f=2B$ นี้มีชื่อเรียกว่าความถี่การชักตัวอย่างไนควิตซ์ (Nyquist sampling frequency) หรือบางครั้งนิยมเรียกสั้นๆว่า ความถี่ไนควิตซ์ หรือ อัตราไนควิตซ์ (Nyquist Rate) ของการชักค่าตัวอย่างสัญญาณ

2.9 การออกแบบวงจรดิจิตอลด้วยภาษา VHDL [7]

ในการออกแบบวงจรดิจิตอล (Digital Circuit) นั้นก้าวหน้าไปอย่างมาก โดยการใช้ภาษาบรรยายการทำงานของวงจร (Hardware Description Language: HDL) ซึ่งเป็นภาษาที่ใช้สำหรับออกแบบฮาร์ดแวร์โดยภาษาที่เป็นมาตรฐานสากล เช่น Verilog หรือ VHDL (VHSIC Hardware Description Language (VHSIC: Very High Speed Integrated Circuit)) หรือภาษาที่ไม่เป็นมาตรฐานเช่น AHDL (Altera Hardware Description Language) หรือ PHDL (Philips Hardware Description Language) เป็นต้น มาบรรยายการทำงานของ วงจรที่ได้ออกแบบไว้ ทำให้ลดความยุ่งยากในการนำเอาอุปกรณ์มาเชื่อมต่อให้เป็นวงจร รวมทั้งลดเวลาที่ใช้ในการออกแบบและทดสอบการทำงาน ซึ่งมีความแตกต่างเป็นอย่างมากเมื่อเปรียบเทียบกับการออกแบบในอดีตที่ผ่านมาคือผู้ออกแบบจะต้องนำเอาอุปกรณ์แต่ละตัวที่ทำการออกแบบไว้ มาทำการต่อทดลองในแผงวงจรจริงและทำการทดสอบวงจรเพื่อหาข้อผิดพลาด ซึ่งต้องใช้เวลานานกับการแก้ปัญหาแต่ละอย่างที่เกิดขึ้น แต่ในการออกแบบด้วยภาษา VHDL ผู้ออกแบบเพียงแต่เขียนซอสโค้ด (Source Code) บรรยายการทำงานของวงจร หลังจากนั้นก็ทำการคอมไพล์ (Compile) แล้วจำลองการทำงาน (Simulate) ดูว่าได้ฟังก์ชันการทำงานและไทม์มิ่ง (Timing) ตามที่ต้องการหรือไม่ จากนั้นก็นำซอสโค้ดที่ได้ไปทำการสังเคราะห์ด้วยโปรแกรมสังเคราะห์ (Synthesis Tool)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สุดท้ายนำวงจรที่ได้จากการสังเคราะห์ไปทำการแม็ป (Map) ลงไปยัง FPGA (Field Programmable Gate Array) เพื่อเป็นชิป (Chip) ต้นแบบสำหรับการนำไปทดสอบการทำงาน

2.9.1 ภาษา VHDL และส่วนประกอบต่าง ๆ ของภาษา

วิวัฒนาการของภาษา VHDL นั้นเริ่มต้นประมาณปี ค.ศ.1981 โดยที่กระทรวงกลาโหมสหรัฐอเมริกา หรือ DOD (Department of Defense) ได้ทำการพัฒนาโครงการที่มีชื่อว่า VHSIC ซึ่งเป็นการพัฒนาโปรแกรมซึ่งจัดเป็นภาษาระดับสูง เช่นเดียวกับภาษา C หรือ Pascal แต่สามารถบรรยายพฤติกรรมการทำงาน หรือโครงสร้างของวงจรได้ ทั้งนี้เพื่อให้สามารถออกแบบและสร้างวงจรรวมได้รวดเร็วขึ้น

ในการเขียนรูปแบบบรรยายระบบเชิงเลขในลักษณะของการออกแบบจากบนลงล่างจะต้องทำความเข้าใจในเรื่องของโครงสร้างและส่วนประกอบต่าง ๆ ของรูปแบบภาษา VHDL เสียก่อน ซึ่งส่วนประกอบที่สำคัญและเป็นพื้นฐานของการเขียนมี 4 หน่วยคือ

1) หน่วยการออกแบบเอนทิตี (Entity Design unit)
2) หน่วยการออกแบบสถาปัตยกรรม (Architecture Design unit)
3) หน่วยการออกแบบแพ็กเก็ต (Package Design unit)
4) หน่วยการออกแบบโครงแบบ (Configuration Design unit)

2.9.1.1 หน่วยการออกแบบเอนทิตี

หน่วยการออกแบบนี้เป็นส่วนที่ใช้สำหรับติดต่อระหว่างภายนอกกับรูปแบบที่เขียนขึ้นโดยเป็นการกำหนดจุดเชื่อมต่อของรูปแบบ กำหนดทิศทางการไหลของสัญญาณ และประเภทของค่าที่สามารถกำหนดให้กับสัญญาณตามจุดต่าง ๆ ของข้อมูลที่ไหลผ่านจุดต่อเหล่านั้น ซึ่งจะมีรูปแบบการเขียนดังรูปที่ 2.12

```
Entity component_name is
        Input and Output ports
        Physical and other parameter
```

รูปที่ 2.12 โครงสร้างโดยทั่วไปของหน่วยการออกแบบเอนทิตี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.12 ส่วนนี้จะขึ้นต้นด้วยคำว่า Entity และ is ระหว่างคำทั้งสองเป็นส่วนสำหรับชื่อรูปแบบที่ต้องการจะเขียน (component_name) หลังจากนั้นจะตามด้วยส่วนที่ใช้กำหนดช่องทางเข้าและออกของข้อมูล (Input Output) รวมทั้งพารามิเตอร์อื่น ๆ และที่สำคัญคือ หน่วยการออกแบบเอนทิตีจะต้องปิดท้ายด้วยคำว่า End และเครื่องหมายอัฒภาค (;) เสมอ

2.9.1.2 หน่วยการออกแบบสถาปัตยกรรม

หน่วยการออกแบบสถาปัตยกรรมคือส่วนที่ใช้เขียนบรรยายพฤติกรรมของรูปแบบในมุมมองของการจำลองการทำงานพฤติกรรมต่าง ๆ ที่บรรยายในส่วนนี้ขึ้นอยู่กับข้อมูลที่ผ่านเข้าและออกตรงช่องทาง ตลอดจนพารามิเตอร์ต่าง ๆ ที่กำหนดในหน่วยการออกแบบเอนทิตี

ในหน่วยนี้มีรูปแบบการเขียนดังรูปที่ 2.13 ส่วนของหน่วยการออกแบบสถาปัตยกรรมเริ่มต้นด้วยคำว่า Architecture และตามด้วยชื่อ (identifier) สิ่งที่ต้องกำหนดลงไปได้แก่ สิ่งที่แสดงให้เห็นว่า Architecture นั้นใช้บรรยายหน่วยการออกแบบเอนทิตีใด ๆ (of <entity design unit> is) ส่วนที่อยู่ระหว่าง Architecture และ Begin เป็นพื้นที่ส่วนประกาศหน่วยของสถาปัตยกรรมกำหนด (Architecture declaration area) ที่เป็นส่วนเผื่อเลือก (Option) ในบริเวณนี้สามารถใช้เขียนประกาศ กำหนดค่าต่าง ๆ ที่จะนำไปใช้ภายในสถาปัตยกรรมนั้นได้ อาทิเช่น ประเภท (type) ต่าง ๆ (ตัวอย่างเช่น bit, bit_vector), สัญญาณ (signal), ค่าคงที่ (constant), โปรแกรมย่อย (ได้แก่ function และ procedure) และอุปกรณ์ (component) ส่วนที่ใช้บรรยายความสำพันธ์ระหว่างข้อมูลที่ไหลเข้าและไหลออกของรูปแบบ (สัญญาณที่กำหนดในชุดคำสั่ง port) นั้นจะถูกบรรยายในบริเวณเนื้อที่ระหว่างคำว่า Begin กับ End ของหน่วยการออกแบบสถาปัตยกรรม และนอกจากนั้นชุด คำสั่งทุกคำสั่งที่อยู่ภายในบริเวณนี้จะเป็นชุดคำสั่งแบบแข่งขนาน (Concurrent statement) เท่านั้นคือทุก ๆ statement จะทำงานพร้อมกันลำดับก่อนหลังจะไม่มีผลต่อการทำงานของรูปแบบ หน่วยการออกแบบสถาปัตยกรรมจะต้องปิดท้ายคำสั่ง End และชื่อของสถาปัตยกรรมนั้น ๆ โดยทั่วไปสามารถเขียนได้ในลักษณะต่าง ๆ ดังนี้

1) ลักษณะการไหลของข้อมูล (Dataflow style)
2) ลักษณะพฤติกรรม (Behavioral style)
3) ลักษณะโครงสร้าง (Structural style)
4) ลักษณะผสม (Mixed Model style)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Architecture  identifier of component_name is
              [declaration]
        Begin
              Specification of the functionality
              of the component in terms of its
              input lines and as influenced by
              physical and other parameters
End [identifier];
```

รูปที่ 2.13 โครงสร้างโดยทั่วไปของหน่วยการออกแบบสถาปัตยกรรม

2.9.1.3 หน่วยการออกแบบแพ็กเก็ต

ข้อมูลต่าง ๆ ตลอดจนโปรแกรมย่อย ที่เป็นประโยชน์ต่อการเขียนรูปแบบบรรยายระบบเชิงเลขสามารถเก็บไว้ในส่วนของแพ็กเก็ตได้ และข้อมูลเหล่านี้สามารถเรียกไปใช้ได้ โดยหน่วยการออกแบบเอนทิตี หน่วยการออกแบบสถาปัตยกรรม หรือจากหน่วยการออกแบบแพ็กเก็ตอื่น ๆ โดยปกติแล้วแพ็กเก็ตจะแบ่งออกเป็น 2 ส่วน คือ การประกาศแพ็กเก็ต (Package Body) เนื่องจากแพ็กเก็ตถูกสร้างขึ้นเป็นส่วนแยกต่างหากออกจากรูปแบบที่กำลังเขียนอยู่ ฉะนั้นการที่จะนำแพ็กเก็ตไปใช้นั้นจะต้องมีการเชื่อมโยงหรืออ้างอิงเสียก่อนซึ่งในภาษา VHDL สามารถกระทำได้ด้วยชุดคำสั่ง USE

Package Declaration ส่วนที่มีความสำคัญที่สุดของแพ็กเก็ต (ถ้ามองในแง่ การนำไปใช้จากภายนอก) ได้แก่ส่วนการประกาศแพ็กเก็ต เพราะจะเป็นส่วนที่กำหนดชื่อของสิ่งที่ ประกาศอยู่ภายในแพ็กเก็ตสำหรับนำไปใช้ภายนอกตัวของแพ็กเก็ตเอง สิ่งใดๆที่ถูกประกาศไว้ใน ส่วนของบอดี้แพ็กเก็ต แต่ไม่ได้ถูกประกาศไว้ในส่วนการประกาศแพ็กเก็ตจะไม่สามารถถูกนำค่า และพฤติกรรมไปใช้ส่วนนอกได้ซึ่งสามารถเปรียบเทียบได้กับสิ่งที่ประกาศไว้ในส่วนของการประกาศเอนทิตี คือจุดเชื่อมต่อหรือพอร์ทที่มีหน้าที่ติดต่อกับโลกภายนอก ฉะนั้นโดยทั่วไปแล้วแพ็กเก็ต สามารถสร้างขึ้นได้โดยไม่จำเป็นต้องมีส่วนบอดี้ดังแสดงในรูปที่ 2.14 และยังสามารถถูกนำไปใช้จากรูปแบบภายนอกได้ เช่น ใช้สำหรับประกาศชนิด (Type) หรือสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เช่นเดียวกันกับส่วนบอดี้แพ็กเก็ตที่ไม่จำเป็นต้อง มีส่วนของการประกาศแพ็กเก็ต แต่แพ็กเก็ตนั้นจะไม่สามารถถูกนำไปใช้จากรูปแบบอื่นได้

```
Package package_name is
        Package_declaration_part
End package_name;
```

รูปที่ 2.14 โครงสร้างโดยทั่วไปของส่วนการประกาศแพ็กเก็ต

บอดี้แพ็กเก็ต ดังรูปที่ 2.15 เป็นโครงสร้างที่ประกอบด้วยคำสั่งต่าง ๆ ในรูปของคำสั่งลำดับ (Sequence) ที่ใช้บรรยายฟังก์ชันการทำงานของโปรแกรมย่อย (Subprogram) ทั้งหลาย ที่ชื่อของโปรแกรมย่อยนั้น ๆ ที่ถูกประกาศไปในส่วนของการประกาศแพ็กเก็ตแล้วจะถูกเก็บไว้ในส่วนบอดี้แพ็กเก็ต ทั้งนี้รวมทั้งการกำหนดค่าคงที่ต่าง ๆ อันได้แก่ ค่าคงที่ที่ถูกประกาศชื่อก่อนในส่วนของการประกาศแพ็กเก็ต แต่ถูกกำหนดค่าในส่วนของบอดี้แพ็กเก็ต ฉะนั้นส่วนบอดี้แพ็กเก็ตจึงไม่จำเป็นต้องมี ถ้าในส่วนของการประกาศแพ็กเก็ตไม่มีการประกาศชื่อที่เป็นโปรแกรมย่อยหรือค่าคงที่

```
Package body package_name is
        declaration part
End package_name;
```

รูปที่ 2.15 โครงสร้างโดยทั่วไปของบอดี้แพ็กเก็ต

2.9.1.4 หน่วยการออกแบบโครงแบบ

ดังที่ทราบกันแล้วว่ารูปแบบหนึ่งของระบบเชิงเลขไม่ว่าจะเป็นอะไร จะมีหน่วยการออกแบบเอนทิตีได้เพียงหนึ่งเดียวเท่านั้น แต่หน่วยการออกแบบเอนทิตีหนึ่งหน่วยนี้อาจจะมีสถาปัตยกรรมที่เป็นหน่วยรองได้หลายหน่วย ดังนั้นจะต้องมีหน่วยการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกแบบโครงแบบมาเพื่อกำหนดการใช้โครงแบบ (Configuration) ประกอบเอนทิตีกับหน่วยการออกแบบสถาปัตยกรรมหน่วยไหนเข้าด้วยกัน มีรูปแบบการเขียนดังรูปที่ 2.16

```
Configuration identifier of entity_name is
        Configuration_declarative_part
End;
```

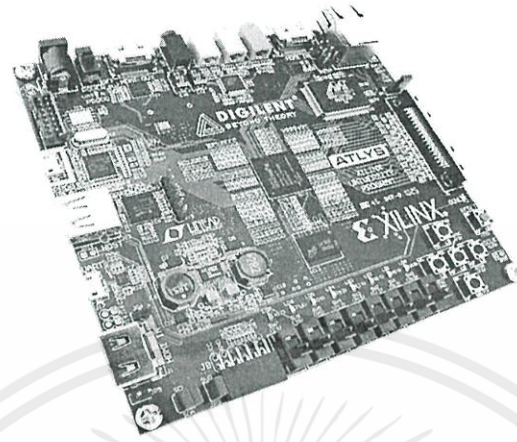
รูปที่ 2.16 โครงสร้างโดยทั่วไปของหน่วยการออกแบบโครงแบบ

2.9.2 การออกแบบวงจรดิจิตอลด้วยอุปกรณ์ FPGA

อุปกรณ์ FPGA (Field Programmable Gate Array) เป็นอุปกรณ์ที่ใช้ในการโปรแกรมวงจรที่ได้ออกแบบลงไปเพื่อให้อุปกรณ์ FPGA มีฟังก์ชันการทำงานตามที่ออกแบบไว้ ในการทำ FPGA ซึ่งเป็นวิธีการออกแบบ IC (Integrated Circuit) แบบ Semicustom อีกวิธีหนึ่ง เมื่อเทียบกับการทำ ASICs (Application Specific Integrated Circuit) แล้วนั้นก็มีทั้งข้อดีและข้อเสียคือการทำ FPGA จะมีข้อจำกัดในด้านขนาดของวงจรเพราะภายในอุปกรณ์ FPGA จะมีจำนวนเกท (Gate) ให้ใช้จำนวนจำกัด และการทำ FPGA ก็เหมาะสำหรับการทำผลิตภัณฑ์ต้นแบบหรือเพื่อผลิต ในปริมาณต่ำส่วนข้อดีของการทำ FPGA ก็คือระยะเวลาที่ใช้ในการทำตั้งแต่เขียนรหัส (Code) อธิบายฮาร์ดแวร์จนกระทั่งดาวน์โหลด (Download) นั้นน้อยกว่าการทำ ASIC มาก และการตรวจสอบหรือแก้ไขการออกแบบก็ทำได้สะดวก

การทำ FPGA ในปัจจุบันมีประสิทธิภาพและความสะดวกมากขึ้น ทั้งนี้ก็เนื่องจากทาง บริษัทผู้ผลิตอุปกรณ์ FPGA ได้เพิ่มความสามารถของอุปกรณ์ FPGA โดยเพิ่มจำนวนองค์ประกอบ ภายในหรือปรับปรุงโครงสร้างสถาปัตยกรรมภายใน และยังได้เพิ่มประสิทธิภาพของซอฟต์แวร์ที่ใช้ ทำ PPR (Partitioning, Placement and Routing) สำหรับอุปกรณ์นั้น ๆ ด้วย ในการใช้งาน นั้นอุปกรณ์ FPGA สามารถไปประยุกต์ใช้งานได้ เช่น การประมวลผลสัญญาณเชิงเลข (DSP: Digital Signal Processing) การออกแบบไมโครคอนโทรลเลอร์เป็นต้น ลักษณะของตัว FPGA แสดงได้ดังรูปที่ 2.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 ลักษณะของตัว FPGA

2.9.2.1 การออกแบบโดยใช้ภาษาอธิบายการทำงานของฮาร์ดแวร์

ในการออกแบบวงจรเชิงเลขนั้นทำได้โดยการวาดวงจรหรือใช้ภาษาอธิบายฮาร์ดแวร์ ในขั้นตอนนี้เป็นขั้นตอนที่ไม่แตกต่างกันระหว่างการออกแบบด้วย FPGA และ ASIC ในกรณีที่ใช้ภาษาอธิบายฮาร์ดแวร์ แต่ในกรณีที่ออกแบบโดยวิธีการวาดวงจรจะแตกต่างกันโดยที่การทำวิธีนี้จะต้องคำนึงถึงเทคโนโลยีที่จะใช้ซึ่งแต่ละเทคโนโลยีก็มีความแตกต่างกันไป จะเห็นได้ว่าการออกแบบโดยใช้ภาษาอธิบายฮาร์ดแวร์ทำได้สะดวกกว่า เพราะการทำด้วยวิธีนี้ไม่ต้องคำนึงถึงเทคโนโลยีที่จะใช้ (Technology independence) และที่สำคัญการออกแบบด้วยวิธีนี้สามารถที่จะแก้ไขโมเดล (Model) หรือเปลี่ยนแปลงเทคโนโลยีได้สะดวกกว่าเพราะไม่ต้องวาดวงจรใหม่ นั่นคือการออกแบบโดยใช้ภาษาอธิบายฮาร์ดแวร์จะทำให้โมเดลที่ได้ไม่ขึ้นกับเทคโนโลยี

ในการเขียนโค้ดสิ่งที่ต้องคำนึงถึงคือเขียนอย่างไรจึงจะสามารถสังเคราะห์เป็นวงจรได้และให้คุณสมบัติของวงจรตามที่กำหนดเพราะลักษณะการเขียนโค้ดจะมีผลโดยตรงกับวงจรที่ได้เนื่องจากในการสังเคราะห์วงจรนั้นซอฟต์แวร์สังเคราะห์วงจร (Synthesis Tools) จะทำการสังเคราะห์ตามโค้ดที่เขียน ถ้าอธิบายการทำงานของวงจรเดียวกันแต่เขียนโค้ดในลักษณะที่ต่างกัน เมื่อสังเคราะห์แล้วจะได้วงจรที่ต่างกันและจากวงจรที่ต่างกัน เมื่อนำไปทำต้นแบบด้วย FPGA หรือการทำ ASIC แล้วจะได้ไอซีที่มีคุณสมบัติต่างกันทั้งในด้านของขนาดหรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเร็ว (Area and time) ส่วนการเขียนโค้ดลักษณะใดเพื่อให้ได้ผลลัพธ์ที่ดีที่สุดนั้นก็ขึ้นอยู่กับประสบการณ์ในการออกแบบ

2.9.2.2 การจำลองการทำงานของวงจร (Simulation)

ขั้นตอนนี้เป็นขั้นตอนที่สำคัญเพราะเป็นขั้นตอนที่ใช้ตรวจสอบฟังก์ชันการทำงานของวงจรว่าถูกต้องหรือไม่มีข้อผิดพลาดตรงไหน เพื่อที่จะได้ทำการแก้ไขให้ถูกต้อง ในขั้นตอนนี้จะใช้ซอฟต์แวร์สำหรับทำการจำลองการทำงานของวงจร เช่น V-System และ ModelSim ของบริษัท Model Technology

2.9.2.3 การสังเคราะห์วงจร

ในขั้นตอนนี้จะใช้ซอฟต์แวร์สังเคราะห์วงจร (Synthesis Tools) ทำการสังเคราะห์โค้ดเพื่อให้ได้เป็นวงจรขึ้นมา แต่ต้องตรวจสอบด้วยว่าซอฟต์แวร์นั้น ๆ สนับสนุนเทคโนโลยี FPGA (FPGA Library) ที่ต้องการใช้หรือไม่ โดย FPGA ที่นิยมใช้งานเช่น ของบริษัท Xilinx และบริษัท Altera ซอฟต์แวร์สังเคราะห์วงจรจะแปลงโค้ดและทำการออปติไมซ์ (Optimization) เพื่อให้ได้วงจรตามเทคโนโลยีที่เลือกใช้นอกจากนี้ยังสามารถกำหนดข้อบังคับ สำหรับวงจรได้เช่น ข้อบังคับเรื่องของเวลา (Time Constraints) หรือข้อบังคับในเรื่องของพื้นที่ ซึ่งข้อบังคับเหล่านี้จะถูกนำไป ใช้ในขั้นตอนออปติไมซ์เพื่อให้วงจรที่ได้เป็นไปตามที่กำหนด ส่วน สำคัญในการออปติไมซ์คือการเทียบ (Mapping) วงจรให้เข้ากับเทคโนโลยีที่ใช้เพื่อให้ได้วงจรที่ เหมาะสมกับโครงสร้างสถาปัตยกรรมภายในอุปกรณ์ FPGA ในกรณีของ Xilinx จะเทียบโดยใช้วิธี LUT (Look Up Table) เมื่อทำการสังเคราะห์วงจรเสร็จแล้วซอฟต์แวร์สังเคราะห์วงจรก็จะมีการ รายงานผลว่าวงจรที่ออกแบบไปนั้นเป็นอย่างไร เช่น มีความหน่วง (Delay) เท่าไร ใช้ทรัพยากรต่าง ๆ ใน FPGA อะไรบ้าง เป็นต้น

2.9.2.4 การแบ่งวงจร (Partitioning)

ขั้นตอนนี้เป็นการแบ่งวงจรที่ได้จากการสังเคราะห์ให้เป็นหน่วยย่อย ๆ สำหรับลงใน CLBs, IOBs หรือองค์ประกอบอื่น ๆ ภายในอุปกรณ์ FPGA สำหรับเกณฑ์ที่ใช้ในการแบ่งคือให้แต่ละส่วนที่จะแยกออกจากกันมีจำนวนสัญญาณที่เชื่อมต่อระหว่างกันน้อยที่สุดเท่าที่จะทำได้ เพื่อช่วยลดความหนาแน่นในตอนทำการเชื่อมต่อสัญญาณ (Routing) ในขั้นตอนนี้จะใช้ซอฟต์แวร์ทำ โดยซอฟต์แวร์จะเทียบส่วนประกอบของวงจร เช่น เกท (Gate), ฟลิปฟลอป (Flipflop) ลงในทรัพยากรต่าง ๆ ที่มีอยู่ในอุปกรณ์ FPGA (CLBs, IOBs, BUFT และ edge

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

decoder) หลังจากทำขั้นตอนนี้เสร็จแล้วสามารถที่ทราบว่าวงจรใช้จำนวนทรัพยากรภายในอุปกรณ์ FPGA ไปเท่าไร ส่วนซอฟต์แวร์ที่ใช้ในขั้นตอนนี้ขึ้นอยู่กับตัว FPGA ที่ใช้งานเช่น FPGA ของบริษัท Xilinx จะใช้ Xilinx Foundation Series 2.1i ซึ่งซอฟต์แวร์ตัวนี้จะรวมเอาซอฟต์แวร์ย่อยอื่น ๆ อีก เพื่อให้การทำ PPR (Partitioning, Placement and Routing) เป็นไปอย่างต่อเนื่อง ส่วน FPGA ของบริษัท Altera จะใช้ Altera MAX+II

2.9.2.5 การวางอุปกรณ์ (Placement)

ขั้นตอนนี้เป็นการเลือกทำเลที่ตั้งของแต่ละส่วนของวงจรที่ผ่านการแบ่งวงจร (Partitioning) มาแล้วว่าควรจะอยู่ในตำแหน่งใดในอุปกรณ์ FPGA เพื่อให้ได้ผลลัพธ์ที่ดีที่สุด เช่น วงจรส่วนไหนควรอยู่ใกล้กันเพื่อจะได้ค้นหาเส้นทาง (Route) ได้ง่ายหรือช่วยลดความหน่วง จะเห็นได้ว่าตำแหน่งภายในอุปกรณ์ FPGA นั้นมีความสำคัญเพราะถ้าจัดวางวงจรลงในตำแหน่งที่ไม่เหมาะสมแล้วจะทำให้ความหน่วงเพิ่มขึ้นหรือตัว Router ทำการค้นหาเส้นทางสัญญาณได้ไม่หมด

2.9.2.6 การเชื่อมต่อสัญญาณ (Routing)

ในขั้นตอนนี้เป็นการเชื่อมต่อสัญญาณระหว่างองค์ประกอบต่าง ๆ ภายในอุปกรณ์ FPGA เช่น ระหว่าง CLBs หรือระหว่าง CLBs กับ IOBs ขั้นตอนนี้จะทำต่อเนื่องจากการวางอุปกรณ์ ในกรณีที่ทำการวางอุปกรณ์ไว้ไม่ดีซอฟต์แวร์ ก็จะทำการเชื่อมต่อสัญญาณได้ไม่หมดหรือเกิดความหน่วงเกินค่าที่กำหนดในข้อบังคับโดยสามารถทำขั้นตอนนี้ได้โดยใช้ซอฟต์แวร์ เช่นกัน หรือจะทำการเชื่อมต่อสัญญาณด้วยตัวเอง (Manual Layout) ก็ได้ แต่ทางที่ดีควรใช้ซอฟต์แวร์ทำดีกว่าโดยให้ทำการค้นหาเส้นทางหลายๆ ครั้งเพื่อหาครั้งที่ดีที่สุด นอกจากนั้นการกำหนดข้อบังคับทางเวลา (Time Constraints) จะช่วยให้ผลที่ได้จากการทำการเชื่อมต่อสัญญาณดีขึ้นได้

2.9.2.7 การโปรแกรมอุปกรณ์ FPGA (Configuration)

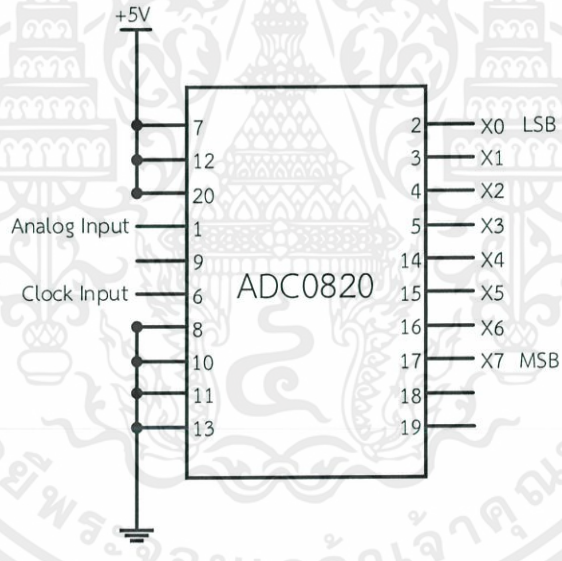
หลังจากที่วงจรผ่านขั้นตอนต่าง ๆ จนกระทั่งผ่านการทำ PPR (Partitioning, Placement and Routing) แล้วนั้น ถึงตอนนี้ก็สามารถที่จะดาวน์โหลด (Download) ลงในอุปกรณ์ FPGA ได้แล้ว ในการดาวน์โหลดนี้ก่อนอื่นต้องแปลงแบบวงจรรวมที่ได้ให้เป็นข้อมูลวงจร (Configuration data) ซึ่งอยู่ในรูปบิตสตรีม (Bit-stream) ก่อนแล้วจึงดาวน์โหลดลงไปเพื่อให้อุปกรณ์ FPGA มีฟังก์ชันการทำงานตามวงจรที่ออกแบบไว้จากที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายมาทั้งหมดจะเห็นได้ว่าการออกแบบเพื่อทำ FPGA นั้น ทำได้สะดวกกว่าการทำ ASIC มากเพราะใช้เวลาน้อยกว่ามาก ส่วนสำคัญที่ใช้ในการทำ FPGA คือ ซอฟต์แวร์ที่ใช้ตั้งแต่การเขียนโค้ดอธิบายฮาร์ดแวร์จนกระทั่งดาวน์โหลดลงในอุปกรณ์ FPGA ซึ่งซอฟต์แวร์ที่ใช้ต้องเป็นซอฟต์แวร์ที่ใช้งานต่อเนื่องกัน

2.10 วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล (Analog to Digital Converter)

วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอลเป็นอุปกรณ์ที่แปลงสัญญาณอนาล็อกให้เป็นสัญญาณดิจิตอล ในปริญญานิพนธ์นี้จะเลือกใช้ไอซี ADC0820 ซึ่งเมื่อแปลงข้อมูลอนาล็อกแล้วจะได้ข้อมูลเป็นดิจิตอล 8 บิต โดยมีวงจรแสดงดังรูปที่ 2.18

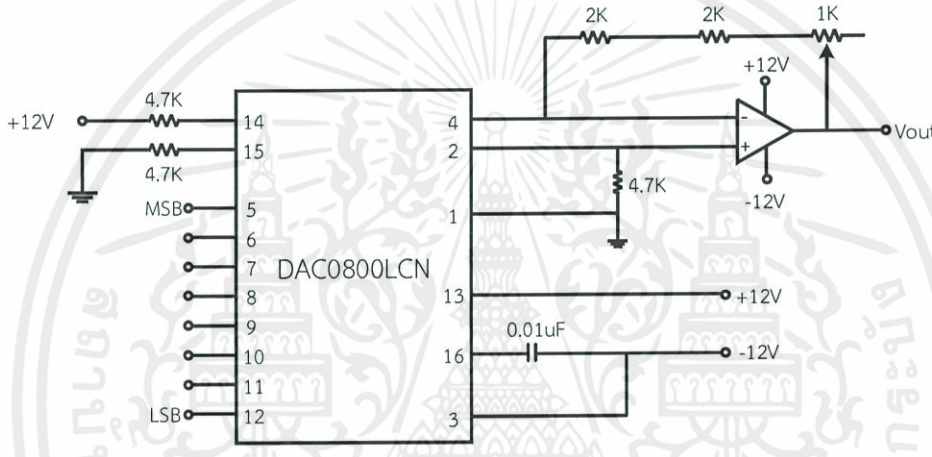


รูปที่ 2.18 วงจรแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิตอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.11 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก (Digital to Analog Converter)

วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อกนเป็นอุปกรณ์ที่แปลงสัญญาณดิจิตอลให้เป็นสัญญาณอนาล็อก ในปริญญานิพนธ์นี้จะเลือกใช้ไอซี DAC0800LCN ซึ่งจะรับข้อมูลดิจิตอลจากฟิลเตอร์แบงก์มาเพียง 8 บิตบนเพื่อนำมาแปลงเป็นสัญญาณอนาล็อก โดยมีวงจรแสดงดังรูปที่ 2.19



รูปที่ 2.19 วงจรแปลงสัญญาณดิจิตอลเป็นสัญญาณอนาล็อก

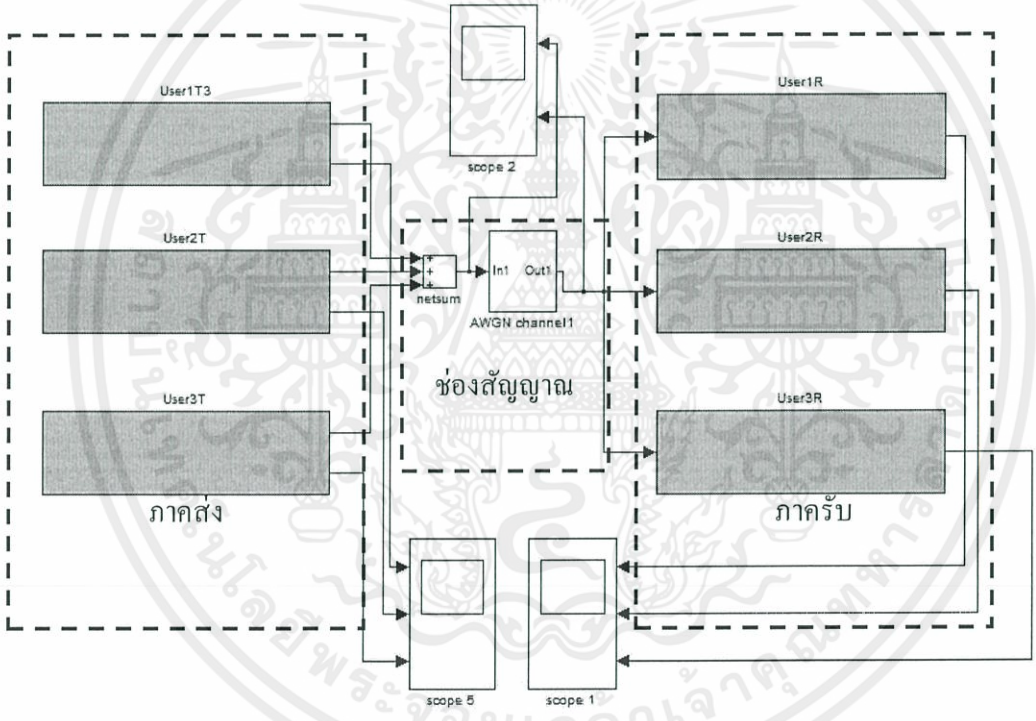
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการจัดทำโครงงาน

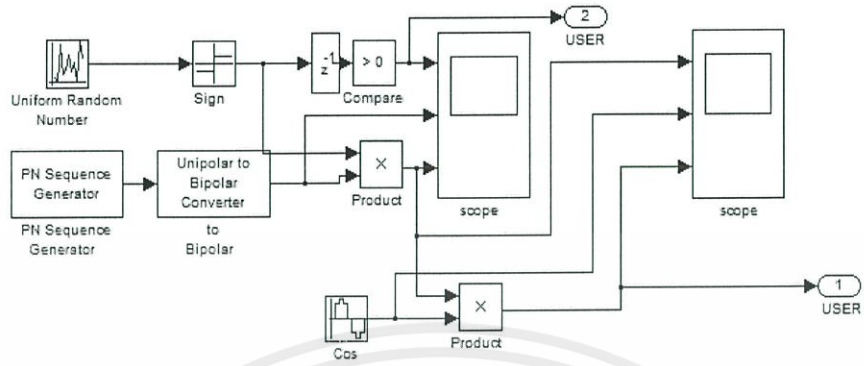
3.1 การออกแบบวงจรด้วยโปรแกรม MATLAB

ในหัวข้อนี้จะกล่าวถึงการจำลองการทำงานของระบบสื่อสารแบบไดเรกต์ซีเควนซ์สเปรดสเปกตรัมซึ่งแสดงให้เห็นในรูปที่ 3.1 โดยใช้การออกแบบการรับส่งข้อมูลผ่านการจำลองด้วยโปรแกรม MATLAB



รูปที่ 3.1 แบบจำลองของระบบไดเรกต์ซีเควนซ์สเปรดสเปกตรัม

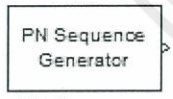
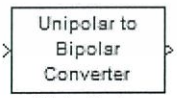
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แบบจำลองของภาคส่ง

จากรูปที่ 3.2 ได้แสดงถึงโมเดลไดเรกต์ซีเควนซ์สเปรดสเปกตรัมของภาคส่งด้วยการจำลองผ่านโปรแกรม MATLAB และการทำงานของบล็อกต่างๆ ในโมเดล ดังตารางที่ 3.1

ตารางที่ 3.1 การทำงานของบล็อกต่างๆ ในภาคส่ง

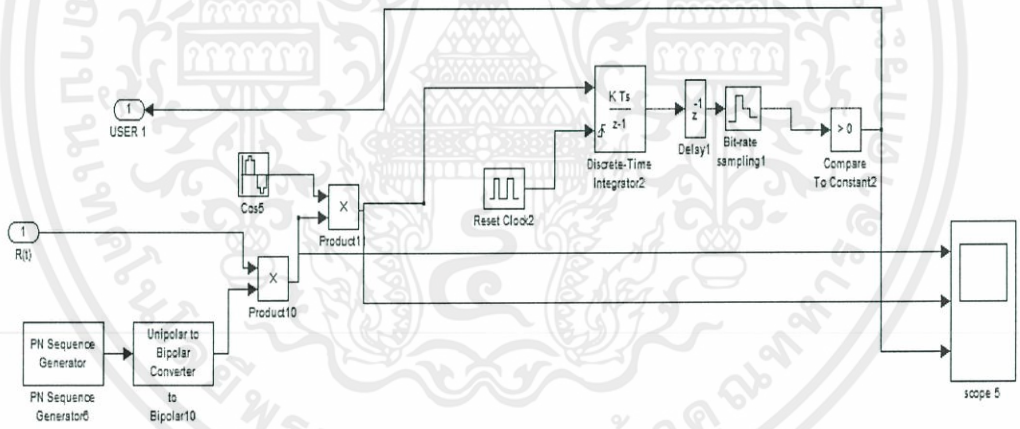
บล็อค	ชื่อ	การทำงาน
Uniform Random Number	Uniform Random Number	เป็นการสร้างสัญญาณจากการสุ่มค่า
Sign	Sign	เพื่อแปลงสัญญาณบิตข้อมูลที่สุ่มจาก Uniform Random Number ให้มีค่าเท่ากับ 1 และ -1
 PN Sequence Generator	PN Sequence Generator	การสร้างรหัสพีเอ็น ซึ่งในแต่ละผู้ใช้จะถูกกำหนดค่าเริ่มต้นของรหัสไม่เหมือนกันเพื่อเป็นการแบ่งแยกข้อมูลของแต่ละผู้ใช้
 Unipolar to Bipolar Converter	Unipolar to Bipolar Converter	แปลงสัญญาณบิตที่ได้จาก PN Sequence Generator ให้มีค่าเท่ากับ 1 และ -1
Carrier	Carrier	การสร้างสัญญาณคลื่นพาห์เพื่อนำไปคูณกับรหัสพีเอ็น ทำให้ได้สัญญาณที่อยู่ในรูปของสัญญาณ Binary Phase Shift Keying (BPSK signal)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 3.1 แสดงถึงการทำงานของบล็อกต่างๆ ในภาคส่ง ที่ใช้ในระบบไดเรกต์ซีเควนซ์สเปรดสเปกตรัมจากโปรแกรม MATLAB

รูปที่ 3.3 บล็อกสัญญาณรบกวนเกาส์สีขาวแบบบวก (Additve White Gaussian Noise : AWGN)

จากรูปที่ 3.3 แสดงถึงบล็อกของสัญญาณรบกวนเกาส์สีขาวแบบบวกของระบบการสื่อสารไดเรกต์ซีเควนซ์สเปรดสเปกตรัม โดยในกรณีศึกษาจะทำการปรับค่า ความแปรปรวนให้มีค่าเท่ากับ 0.01

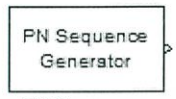
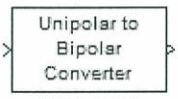


รูปที่ 3.4 แบบจำลองของภาครับ

จากรูปที่ 3.4 ได้แสดงถึงโมเดลไดเรกต์ซีเควนซ์สเปรดสเปกตรัมของภาคส่งด้วยการจำลองผ่านโปรแกรม MATLAB และการทำงานของบล็อกต่างๆ ในโมเดล ดังตารางที่ 3.2

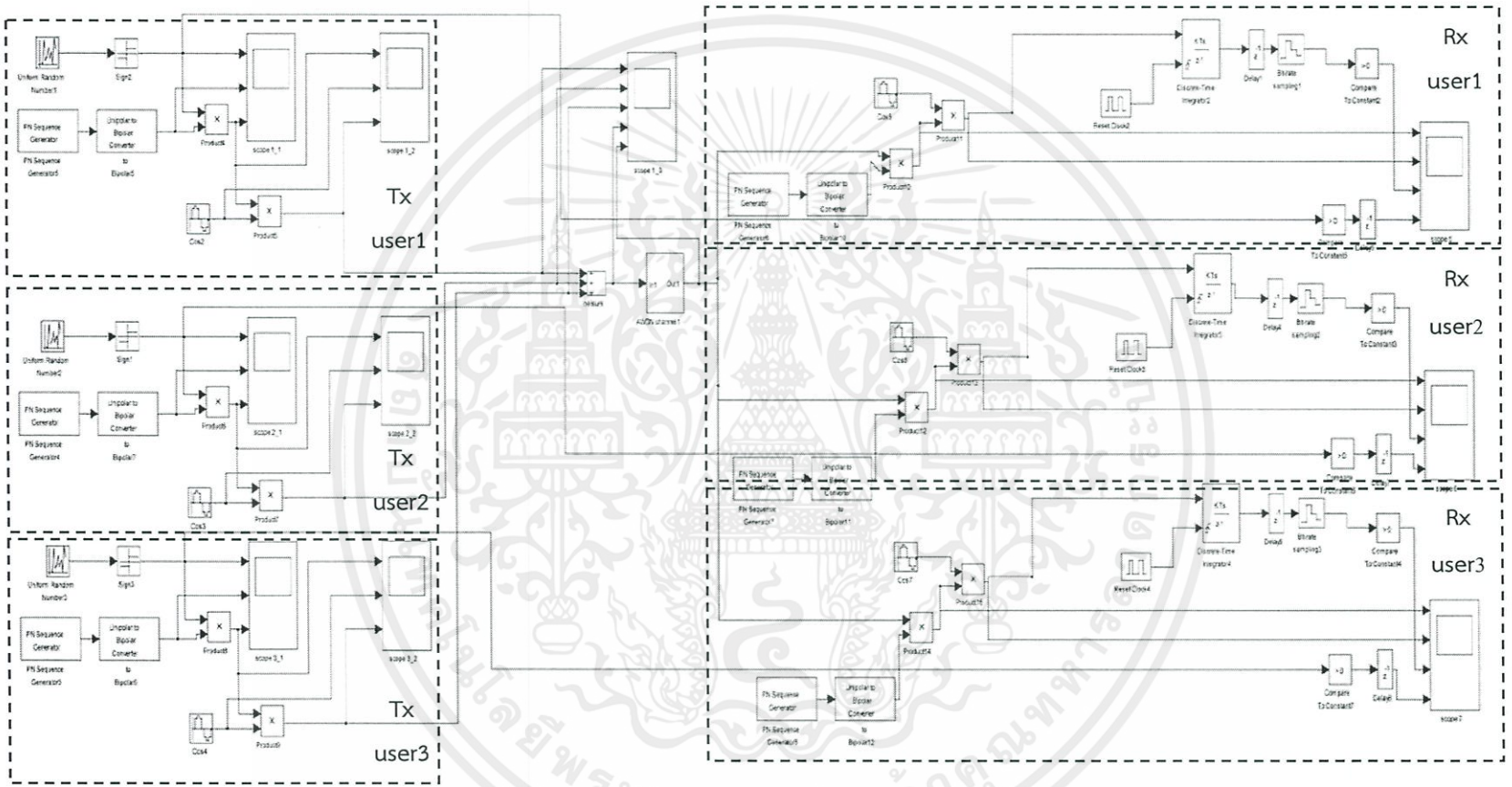
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.2 การทำงานของบล็อกต่างๆ ในภาครับ

บล็อค	ชื่อ	การทำงาน
 PN Sequence Generator	PN Sequence Generator	การกำเนิด PN sequence โดยใช้รหัสชุดเดียวกันกับภาคส่งที่ใช้ในการสเปรดสัญญาณ
 Unipolar to Bipolar Converter	Unipolar to Bipolar Converter	การแปลงค่าสัญญาณจาก 1 และ 0 ให้เป็นสัญญาณสองขั้ว (1 และ –1)
Discrete-Time Integrator	Discrete-Time Integrator	ค่าที่ได้รับจากการอินทีเกรตพื้นที่ใต้กราฟของสัญญาณหลังจากการดีมอดูเลต ซึ่งค่าที่ได้รับสำหรับช่วงเวลาแต่ละบิตเป็นการรวมพลังงานของค่าที่ได้รับในแต่ละบิต
Bit-rate sampling	Bit-rate sampling and Compare to zero	กระบวนการจัดระดับสัญญาณ เพื่อให้ได้สัญญาณกลับคืนมาอย่างถูกต้อง
Carrier	Carrier	สร้างสัญญาณเพื่อการดีมอดูเลต หลังจากกระบวนการดีสเปรดสัญญาณ

จากตารางที่ 3.2 แสดงถึงการทำงานของบล็อกต่างๆ ในภาครับ ที่ใช้ในระบบไดเรกต์ซีเควนซ์สเปรดสเปกตรัมจากโปรแกรม MATLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



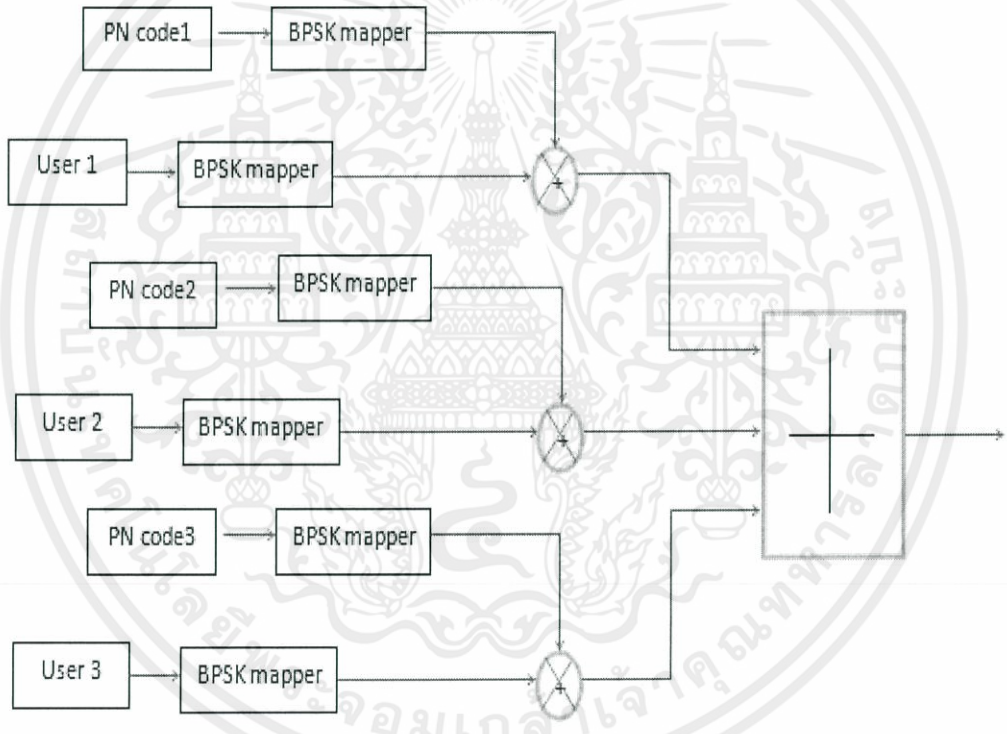
รูปที่ 3.5 โมเดลการจำลองของภาคส่งและภาครับ ของจำนวนผู้ใช้ 3 ราย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

3.2 การออกแบบวงจรด้วยภาษา VHDL

3.2.1 การออกแบบวงจรภาคส่ง

การออกแบบวงจรในภาคส่งนั้นได้ออกแบบให้สัญญาณข้อมูลมีการรับข้อมูลเข้ามาแล้วจะนำข้อมูลมาทำการเข้ารหัสสัญญาณด้วยรหัสพีเอ็น ซึ่งจะได้จากการสร้างด้วย Linear feedback shift register (3 stage) ที่สามารถรองรับผู้ใช้ได้พร้อมกันมากสุดได้ถึง 7 ราย แต่ในการทดลองนี้จะออกแบบให้มีผู้ใช้เพียงแค่ 3 ราย และข้อมูลในการส่งแต่ละผู้ใช้ก็จะมีรหัสพีเอ็นที่แตกต่างกันออกไป ซึ่งทำให้ภาครับสามารถรับข้อมูลจากภาคส่งได้อย่างถูกต้อง แสดงดังรูปที่ 3.6



รูปที่ 3.6 แผนผังการทำงานการออกแบบวงจรภาคส่งด้วยภาษา VHDL

3.2.1.1 การออกแบบอุปกรณ์ในภาคส่ง

ในการออกแบบโครงสร้างของภาคส่งแสดงได้ ดังรูปที่ 3.7 ซึ่งวงจรดังกล่าวสามารถแบ่งออกเป็นส่วนย่อยๆได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) วงจรหารความถี่ (Frequency divider)

2) ชิฟต์รีจิสเตอร์ (Shift Register)

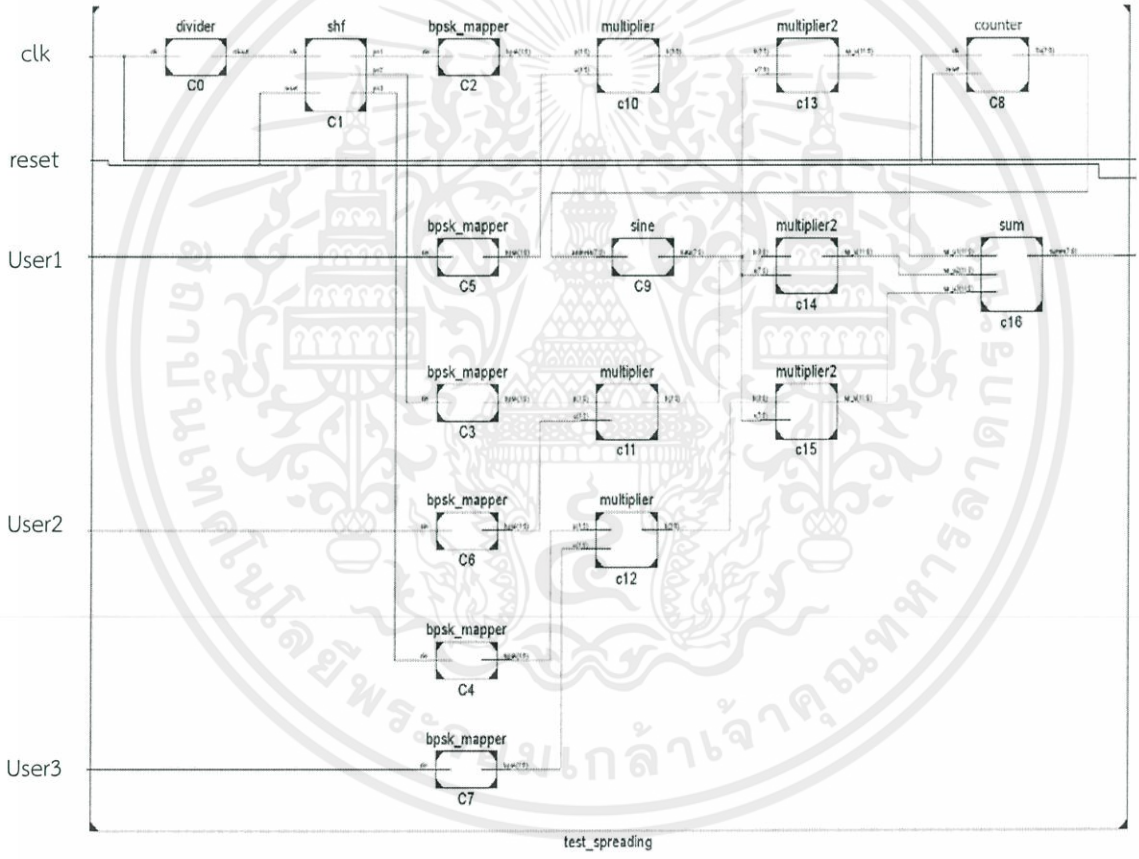
3) วงจรแม็ปค่า (BPSK Mapper)

4) วงจรคูณ (Multiplier)

5) วงจรนับค่า (Counter)

6) วงจรกำเนิดสัญญาณพาหะ (Digital Oscillator)

7) วงจรรวมสัญญาณ (Summing)



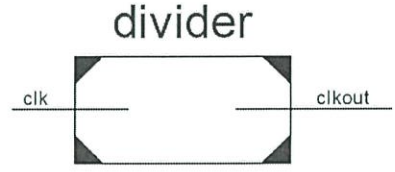
รูปที่ 3.7 วงจรรวมของภาคส่ง

1) วงจรหารความถี่ (Frequency divider)

วงจรหารความถี่สำหรับใช้ในการสร้างรหัสพีเอ็น จะทำการสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ทั้งนี้ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

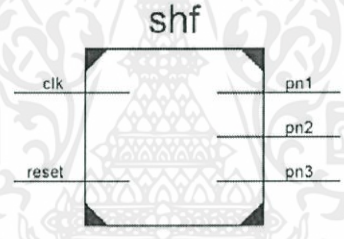
ด้วยภาษา VHDL โดยออกแบบให้เป็น counter ดังรูปที่ 3.8



รูปที่ 3.8 วงจรหารความถี่ (Frequency divider)

2) ชิฟต์รีจิสเตอร์ (Shift Register)

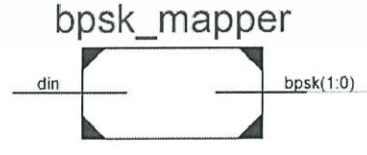
วงจรสร้างสัญญาณรบกวนเทียมนำไปคูณกับสัญญาณข้อมูลเพื่อเป็นการสเปรด โดยการใช้ LSFR (Linear feedback shift register) โดยใช้ทั้งหมด 3 stage ซึ่งทำการสร้างรหัสพีเอ็น ดังรูปที่ 3.9



รูปที่ 3.9 ชิฟต์รีจิสเตอร์ (Shift Register)

3) วงจรแม็ปค่า (BPSK Mapper)

เมื่อวงจรได้รับข้อมูลซึ่งอยู่ในรูปแบบข้อมูลดิจิตอลเข้ามาเป็น 1 และ 0 ซึ่งวงจรนี้จะทำการปรับค่าของข้อมูลเป็น 1 และ -1 ดังรูปที่ 3.10

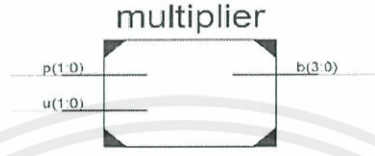


รูปที่ 3.10 วงจรแม็ปค่า (BPSK Mapper)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) วงจรคูณ (Multiplier)

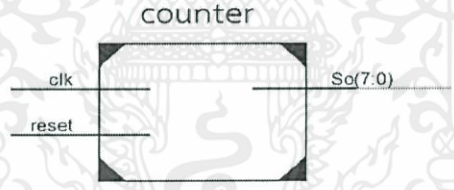
วงจรนี้ทำหน้าที่คูณสัญญาณระหว่างสัญญาณที่ได้จากการแม็ปค่ากับสัญญาณพาหะที่ได้จากเครื่องกำเนิดสัญญาณ ดังรูปที่ 3.11



รูปที่ 3.11 วงจรคูณ (Multiplier)

5) วงจรนับค่า (Counter)

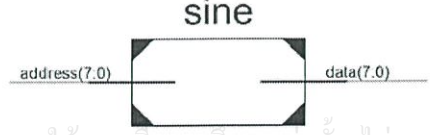
วงจรนับค่าทำหน้าที่ใช้เก็บเลขฐานสองที่แทนจำนวนพัลส์ของสัญญาณนาฬิกาที่ป้อนเข้าอินพุต สัญญาณนาฬิกาทำให้ฟลิปฟลอปมีการเปลี่ยนสถานะ ลอจิกที่เอาต์พุต ซึ่งบอกจำนวนพัลส์ของสัญญาณนาฬิกาที่ป้อนเข้าวงจร ดังรูปที่ 3.12



รูปที่ 3.12 วงจรนับค่า (Counter)

6) วงจรกำเนิดสัญญาณพาหะ (Digital Oscillator)

วงจรกำเนิดสัญญาณพาหะ ทำหน้าที่สร้างสัญญาณพาหะ ดังรูปที่ 3.13

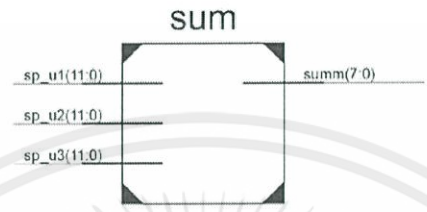


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.13 วงจรกำเนิดสัญญาณพาหะ (Digital Oscillator)

7) วงจรรวมสัญญาณ (Summing)

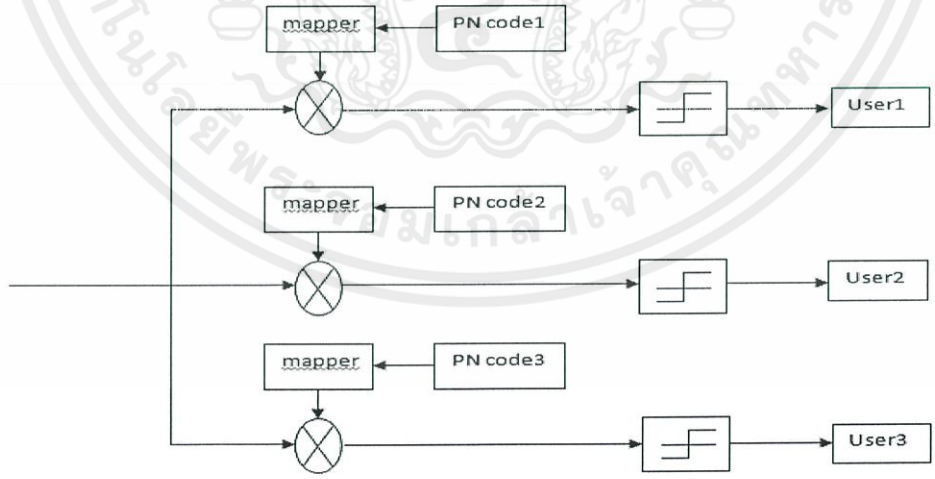
วงจรรวมสัญญาณทำหน้าที่รวมสัญญาณของผู้ใช้แต่ละรายที่ได้หลังจากการสเปรดสเปกตรัมแล้วส่งผ่านช่องสัญญาณ ดังรูปที่ 3.14



รูปที่ 3.14 วงจรรวมสัญญาณ (Summing)

3.2.2 การออกแบบวงจรภาครับ

การออกแบบวงจรในภาครับนั้น เราจะทำการถอดรหัสด้วยการดีสเปรด (Despreading) โดยการนำข้อมูลที่ได้คูณกับรหัสพีเอ็นของผู้ใช้แต่ละราย ซึ่งกำหนดให้ รหัสพีเอ็นในภาครับของผู้ใช้แต่ละราย จะต้องเหมือนกันกับรหัสพีเอ็นในภาคส่ง เพื่อจำแนกข้อมูลออกจากสัญญาณรวมออกเป็นข้อมูลของผู้ใช้ในแต่ละราย



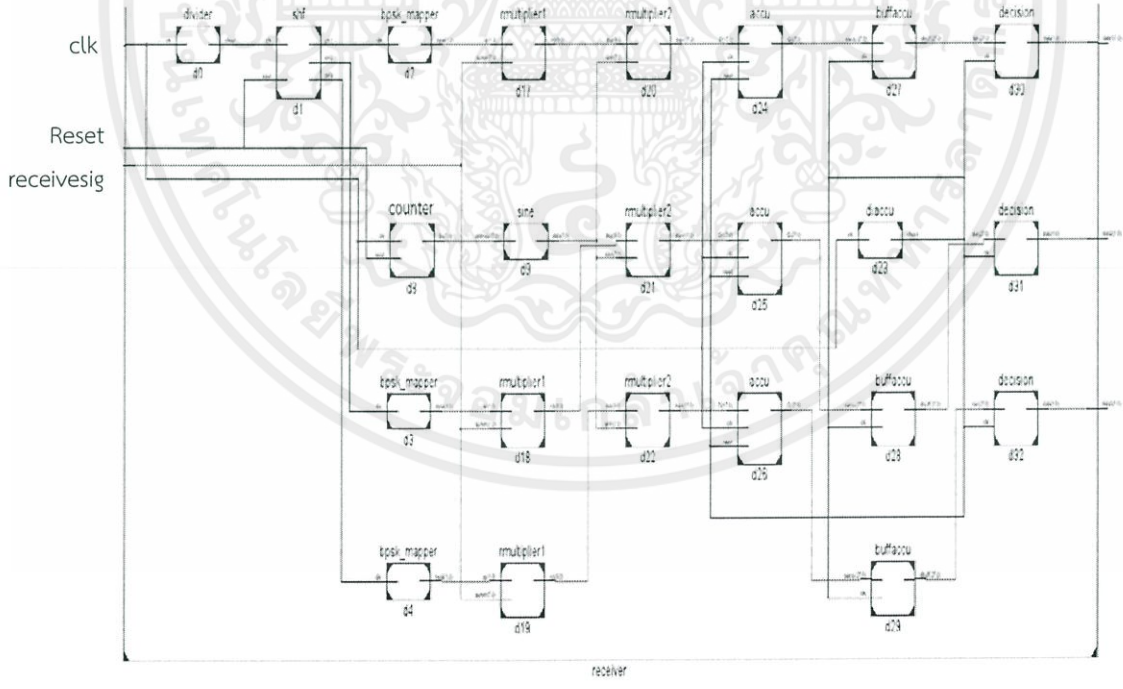
รูปที่ 3.15 แผนผังการทำงานการออกแบบวงจรภาครับด้วยภาษา VHDL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2.1 การออกแบบอุปกรณ์ในภาครับ

ในการออกแบบโครงสร้างของภาครับแสดงได้ ดังรูปที่ 3.16 ซึ่งวงจรดังกล่าวสามารถแบ่งออกเป็นส่วนย่อยๆได้ดังนี้

1) วงจรหารความถี่ (Frequency divider)
2) ชิฟต์รีจิสเตอร์ (Shift Register)
3) วงจรแม็ปค่า (BPSK Mapper)
4) วงจรคูณ (Multiplier)
5) วงจรนับค่า (Counter)
6) วงจรบวกสะสมค่า (Accumulator)
7) วงจรบัฟเฟอร์ (Buffer)
8) วงจรกำหนดขอบเขต (Counter Accumulator)
9) วงจรตัดสินใจบิต (Decision)

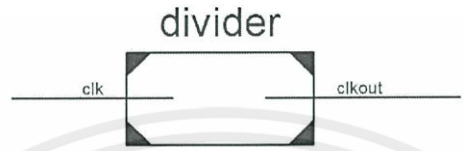


รูปที่ 3.16 วงจรรวมของภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) วงจรหารความถี่ (Frequency divider)

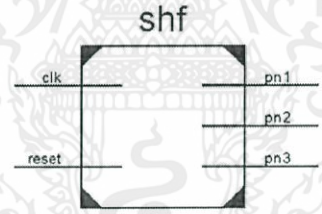
วงจรหารความถี่สำหรับใช้ในการสร้างรหัสพีเอ็น จะทำการสร้างด้วยภาษา VHDL โดยออกแบบให้เป็น counter ดังรูปที่ 3.17



รูปที่ 3.17 วงจรหารความถี่ (Frequency divider)

2) ชิฟต์รีจิสเตอร์ (Shift Register)

วงจรสร้างสัญญาณรบกวนเทียมนำไปคูณกับสัญญาณข้อมูลเพื่อเป็นการสเปรด โดยการใช้ LSFR (Linear feedback shift register) โดยใช้ทั้งหมด 3 stage ซึ่งทำการสร้างรหัสพีเอ็น ดังรูปที่ 3.18



รูปที่ 3.18 ชิฟต์รีจิสเตอร์ (Shift Register)

3) วงจรแม็ปค่า (BPSK Mapper)

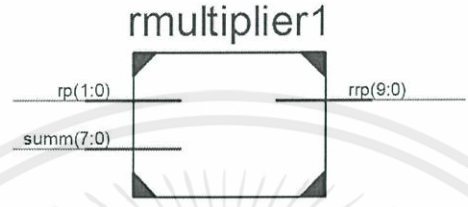
เมื่อวงจรได้รับข้อมูลซึ่งอยู่ในรูปแบบข้อมูลดิจิตอลเข้ามาเป็น 1 และ 0 ซึ่งวงจรนี้จะทำการปรับค่าของข้อมูลเป็น 1 และ -1 ดังรูปที่ 3.19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.19 วงจรแม็ปค่า (BPSK Mapper)

4) วงจรคูณ (Multiplier)

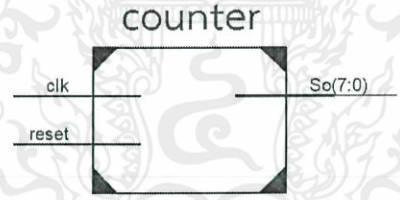
วงจรนี้ทำหน้าที่คูณสัญญาณระหว่างสัญญาณที่ได้จากการแม็ปค่ากับสัญญาณพาหะที่ได้จากเครื่องกำเนิดสัญญาณ ดังรูปที่ 3.20



รูปที่ 3.20 วงจรคูณ (Multiplier)

5) วงจรนับค่า (Counter)

วงจรนับค่าทำหน้าที่ใช้เก็บเลขฐานสองที่แทนจำนวนพัลส์ของสัญญาณนาฬิกาที่ป้อนเข้าอินพุต สัญญาณนาฬิกาทำให้ฟลิปฟลอปมีการเปลี่ยนสถานะ ลอจิกที่เอาต์พุต ซึ่งบอกจำนวนพัลส์ของสัญญาณนาฬิกาที่ป้อนเข้าวงจร ดังรูปที่ 3.21

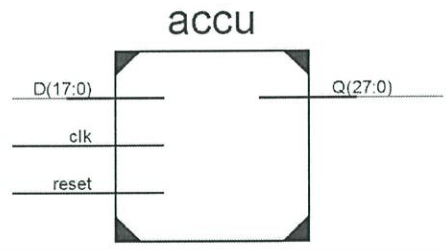


รูปที่ 3.21 วงจรนับค่า (Counter)

6) วงจรบวกสะสมค่า (Accumulator)

วงจรบวกสะสมค่าทำหน้าที่รับข้อมูล 18 บิต เข้ามาทำการบวกสะสมค่าจนครบ 500 ชุดแล้วจึงส่งผลบวกของข้อมูลทั้ง 500 ชุด ออกไปพร้อมทั้งล้างข้อมูลภายในเพื่อทำการรับข้อมูลในชุดถัดไป โดยอาศัยสัญญาณนาฬิกาในการควบคุมการล้างข้อมูล เนื่องจากการล้างค่าภายในข้อมูลต้องทำงานภายในจังหวะก่อนสัญญาณข้อมูลชุดต่อไปจะเข้ามา ดังนั้นสัญญาณนาฬิกาที่ใช้ควบคุมจึงจำจำเป็นต้องไวกว่าสัญญาณในการนับชุดข้อมูลแต่ละชุด ดังรูปที่ 3.22

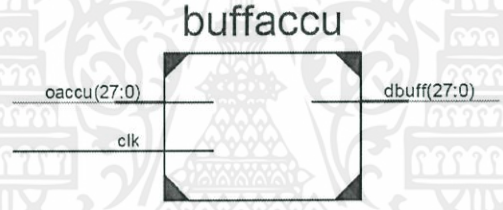
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.22 วงจรบวกสะสมค่า (Accumulator)

7) วงจรบัฟเฟอร์ (Buffer)

วงจรบัฟเฟอร์ทำหน้าที่คงค่าสัญญาณที่ได้รับมา เพื่อให้ข้อมูลที่ได้รับมาไม่เปลี่ยนแปลง ดังรูปที่ 3.23



รูปที่ 3.23 วงจรบัฟเฟอร์ (Buffer)

8) วงจรกำหนดขอบเขต (Counter Accumulator)

วงจรนี้ทำหน้าที่กำหนดขอบเขตของการบวกสะสมค่าแล้วนำไปตัดสินใจบิตที่ถูกต้อง ดังรูปที่ 3.24

รูปที่ 3.24 วงจรกำหนดขอบเขต (Counter Accumulator)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9) วงจรตัดสินใจบิต (Decision)

วงจรนี้ทำหน้าที่ในการตัดสินค่าในการกู้สัญญาณกลับว่าจะออกเป็นสัญญาณลอจิก 0 หรือ 1 โดยตัดสินว่าค่าระดับแรงดันเสมือนมีค่ามากกว่า 0 หรือน้อยกว่า 0 ถ้ามากกว่า 0 ให้แสดงว่าเป็น 1 ถ้าน้อยกว่า 0 ให้แสดงค่าเป็น 0 โดยพิจารณาจากบิต MSB ซึ่งเป็นบิตเครื่องหมาย (Sign bit) โดยถ้าหากเป็น 0 จะหมายความว่าข้อมูลมีค่ามากกว่า 0 จะทำการส่งข้อมูลลอจิก 1 ออกไป และหากเป็น 1 จะหมายความว่า ข้อมูลมีค่าน้อยกว่า 0 ก็จะทำการส่งข้อมูลลอจิก 0 ออกไป ดังรูปที่ 3.25

รูปที่ 3.25 วงจรตัดสินใจบิต (Decision)

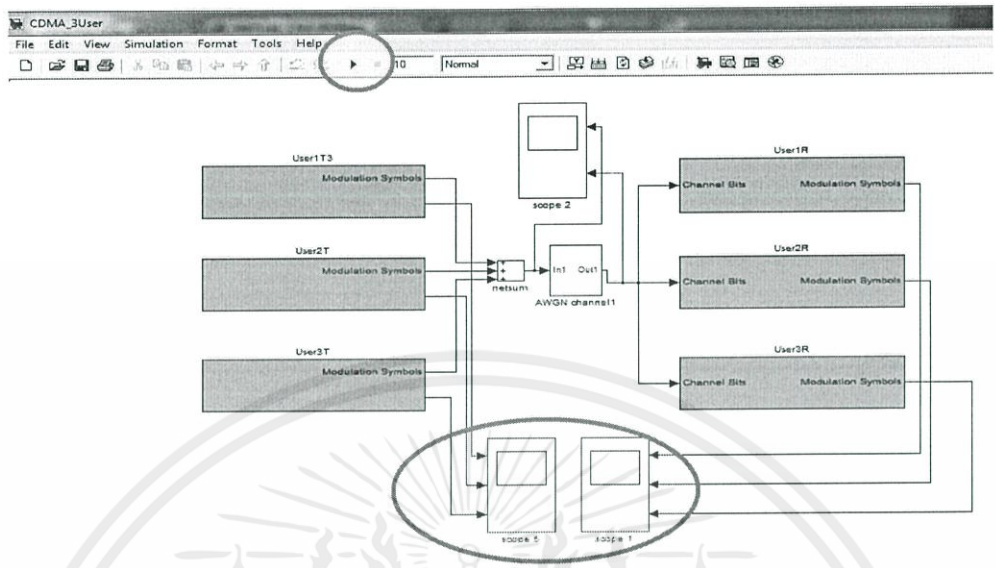
3.3 การจัดเก็บผลการทดลอง

ในการทดลองระบบการสื่อสารแบบไดเรกต์ซีเควนซ์สเปรดสเปกตรัม จะแบ่งออกเป็น 2 ส่วน คือการทดลองโดยใช้การจำลองการทำงานผ่านโปรแกรม MATLAB และการออกแบบสร้างต้นแบบทางฮาร์ดแวร์ของชุดสาธิตการทำงาน ที่ออกแบบด้วยภาษา VHDL ผ่านโปรแกรม ModelSim 10.1b และ XILINX ISE 14.5

3.3.1 การจำลองการทำงานผ่านโปรแกรม MATLAB

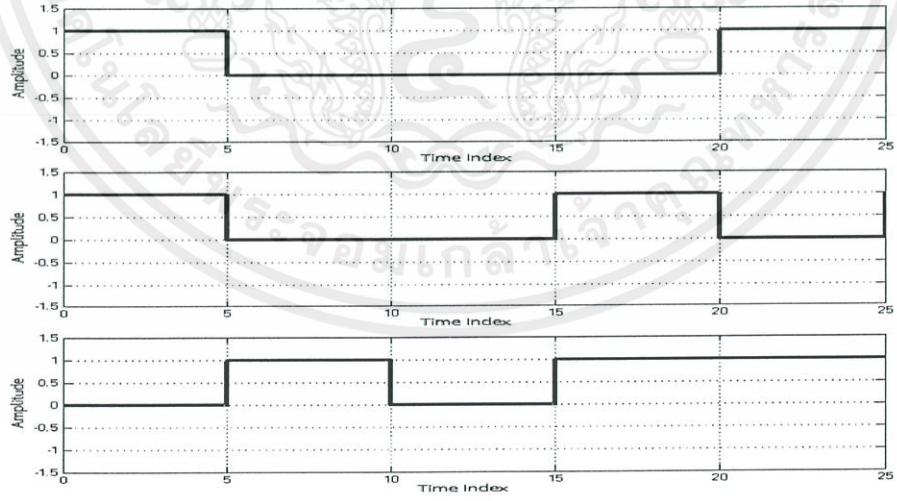
การจัดเก็บผลการทดลองโดยการจำลองระบบรับส่งข้อมูลตามที่ได้ออกแบบไว้ดังหัวข้อที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



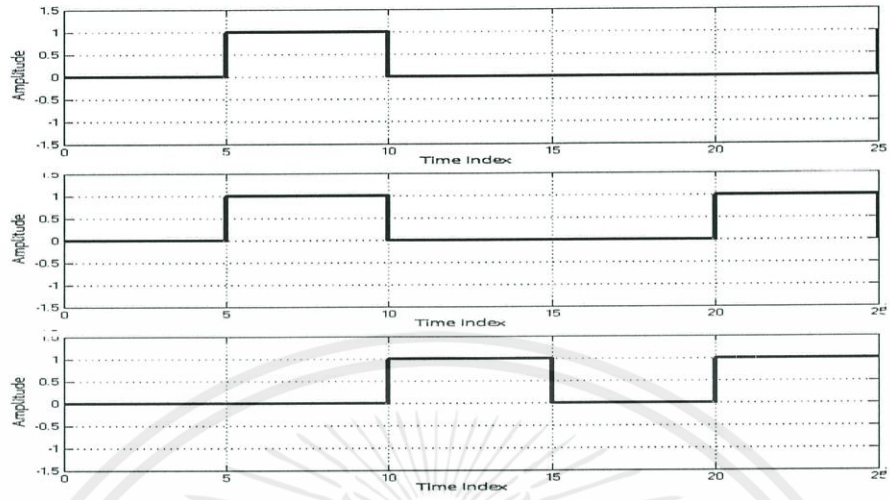
รูปที่ 3.26 การรันการจำลองการทำงานผ่านโปรแกรมMATLAB

จากรูปที่ 3.26 เป็นรูปที่แสดงถึงการเริ่มต้นของการจัดเก็บผลการทดลองด้วยการรันโปรแกรม และทำการเลือกสโคป แล้วเปรียบเทียบสัญญาณอินพุตของภาคส่งกับสัญญาณเอาท์พุตของภาครับ ดังรูปที่ 3.27 และ 3.28



รูปที่ 3.27 สัญญาณอินพุตจากสโคป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.28 สัญญาณเอาต์พุตจากสโคป

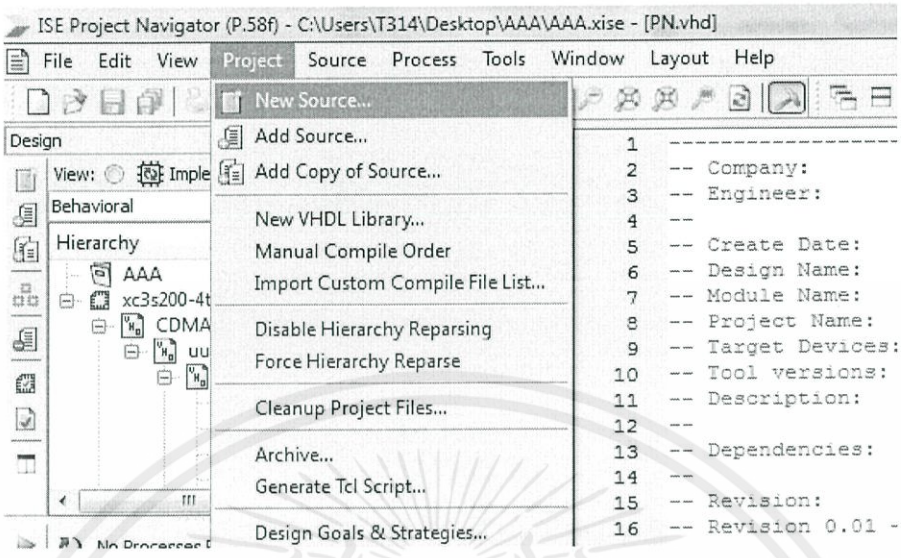
3.3.2 การออกแบบสร้างต้นแบบทางฮาร์ดแวร์ของชุดสาธิตการทำงาน ที่ออกแบบด้วยภาษา VHDL ผ่านโปรแกรม MODELSIM 10.1b และ XILINX ISE 14.5

ทำการทดลองระบบในข้อ 3.3.1 นำลักษณะการทำงานในหัวข้อ 3.1และ 3.2 มาทำการออกแบบสร้างต้นแบบทางฮาร์ดแวร์ของชุดสาธิตการทำงานด้วยภาษา VHDL และจำลองการทำงานในโปรแกรม XILINX ISE 14.5 ร่วมกับโปรแกรม MODELSIM 10.1b โดยมีขั้นตอนดังนี้

ในขั้นตอนนี้ขอยกตัวอย่างการทำงานของอุปกรณ์สร้างรหัสพีเอ็น (PN Sequence Generator)

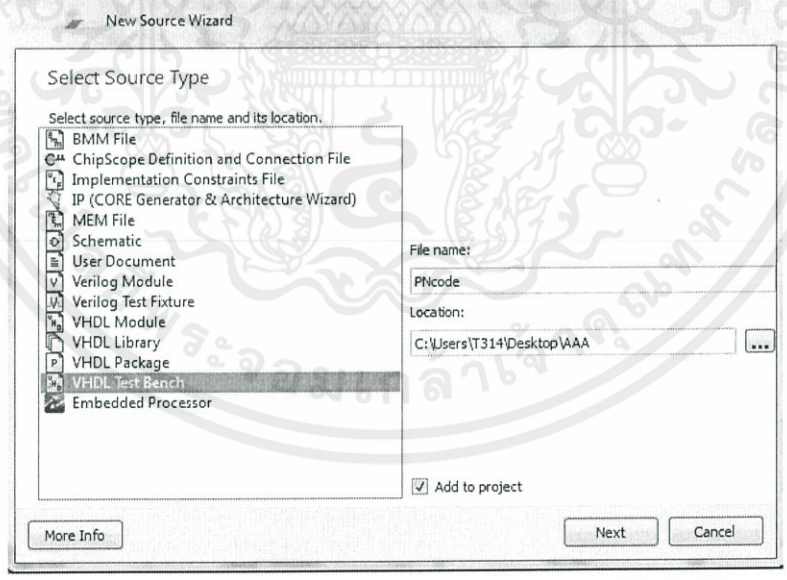
1) การสร้าง VHDL Test Bench โดยเริ่มจากคลิกที่เมนู Project > New Source... ดังรูปที่ 3.29

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.29 การสร้าง VHDL Test Bench

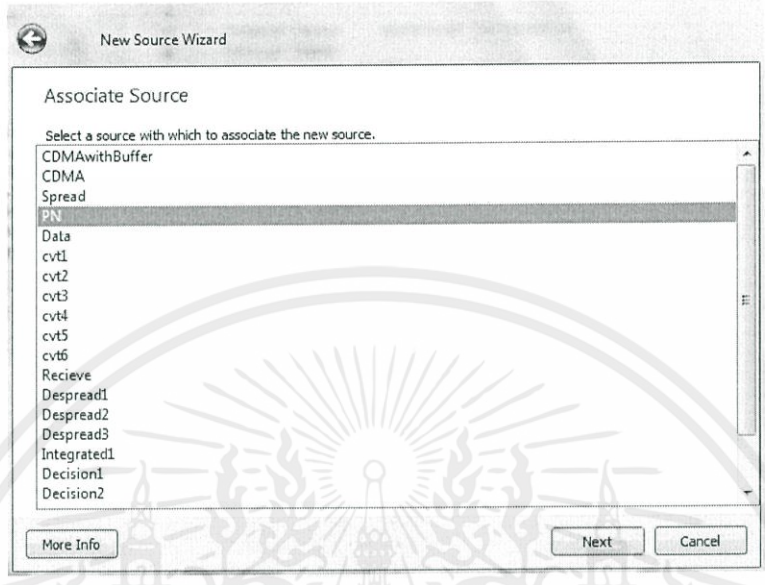
2) ในหน้าต่างเลือกไฟล์ให้เลือก VHDL Test Bench และทำการตั้งชื่อไฟล์ แล้วกดปุ่ม Next ดังรูปที่ 3.30



รูปที่ 3.30 การตั้งชื่อ VHDL Test Bench

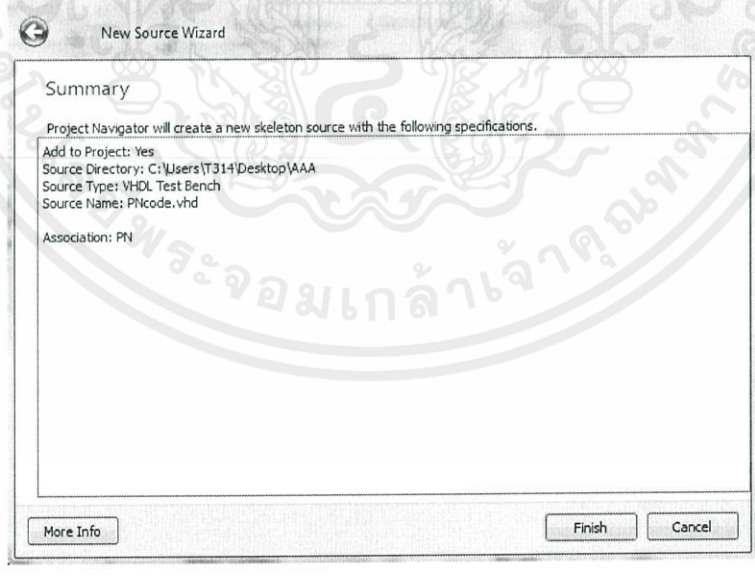
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) เลือกไฟล์ที่จะนำมาทำการทดลองจากไฟล์ .vhd ที่สร้างไว้ ดังรูปที่ 3.31



รูปที่ 3.31 การเลือกไฟล์ที่จะนำมาสร้าง VHDL Test Bench

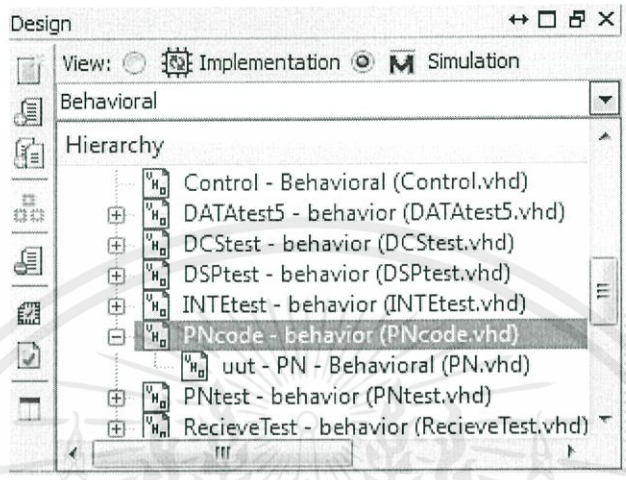
4) จากนั้นกด Finish ตามรูปที่ 3.32



รูปที่ 3.32 หน้าต่างสรุปการสร้าง VHDL Test Bench

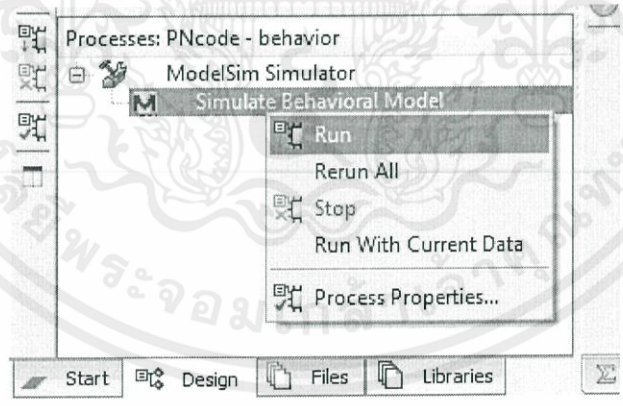
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) เลือกหัวข้อ View เป็น Simulation แล้วเลือก Test Bench ที่ต้องการในหน้าต่าง Hierarchy ดังรูปที่ 3.33



รูปที่ 3.33 การเลือกเปิดหน้าต่างการจำลอง

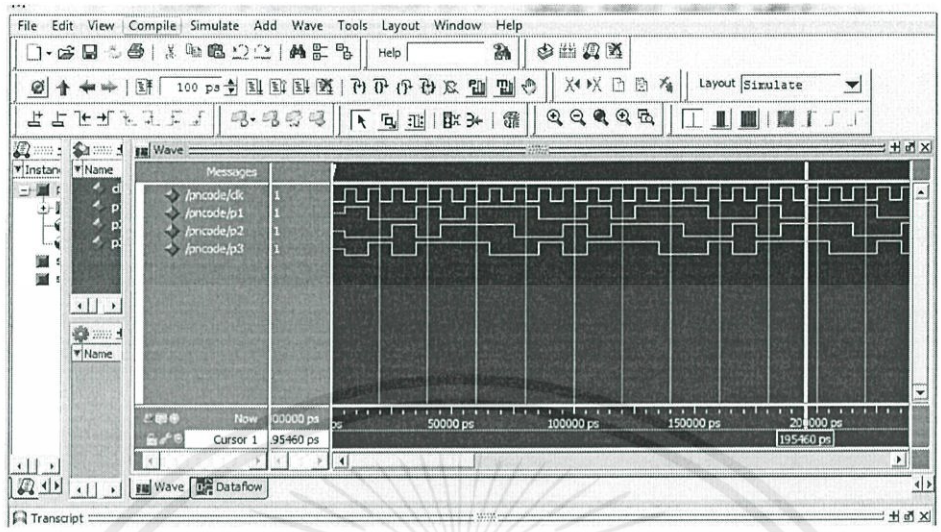
6) ในหน้าต่าง Processes ให้คลิกขวาที่ Simulate Behavioral Model แล้วกด Run ดังรูปที่ 3.34



รูปที่ 3.34 การเลือกการจำลองผ่านโปรแกรม MODELSIM 10.1b

ซึ่งโปรแกรม ISE จะทำการเรียกโปรแกรม MODELSIM 10.1b ขึ้นมาโดยอัตโนมัติ จากนั้นหน้าต่างโปรแกรม MODELSIM จะเปิดขึ้นพร้อมทั้งแสดงผลการจำลอง (Simulation Results) ด้วยรูปที่ 3.35

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



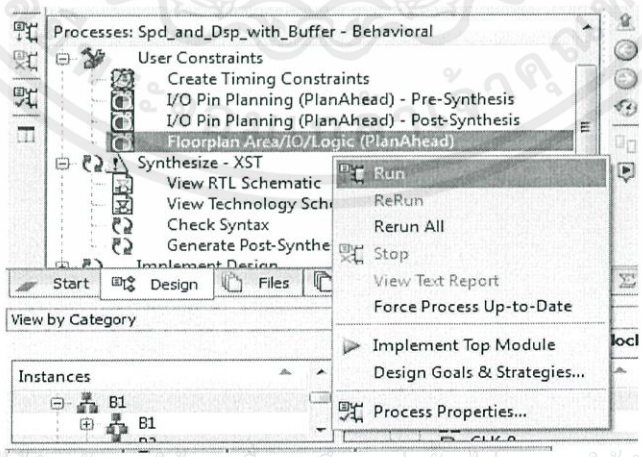
รูปที่ 3.35 ผลการจำลองการทำงานของวงจรสร้างรหัสพีเอ็น (PN Sequence Generator)

3.3.3 การทดสอบวงจรกรองสัญญาณผ่านทาง FPGA

หลังจากบรรยายการจำลองการรับส่งข้อมูลด้วยภาษา VHDL และทำการจำลองอุปกรณ์แต่ละชนิด ก็จะเข้าสู่การสร้างการจำลองระบบรับส่งข้อมูลบน FPGA เพื่อทำเป็นฮาร์ดแวร์ โดยลักษณะของการจำลองระบบรับส่งข้อมูลที่ได้จากการนำอุปกรณ์ย่อยมารวมกันแล้ว

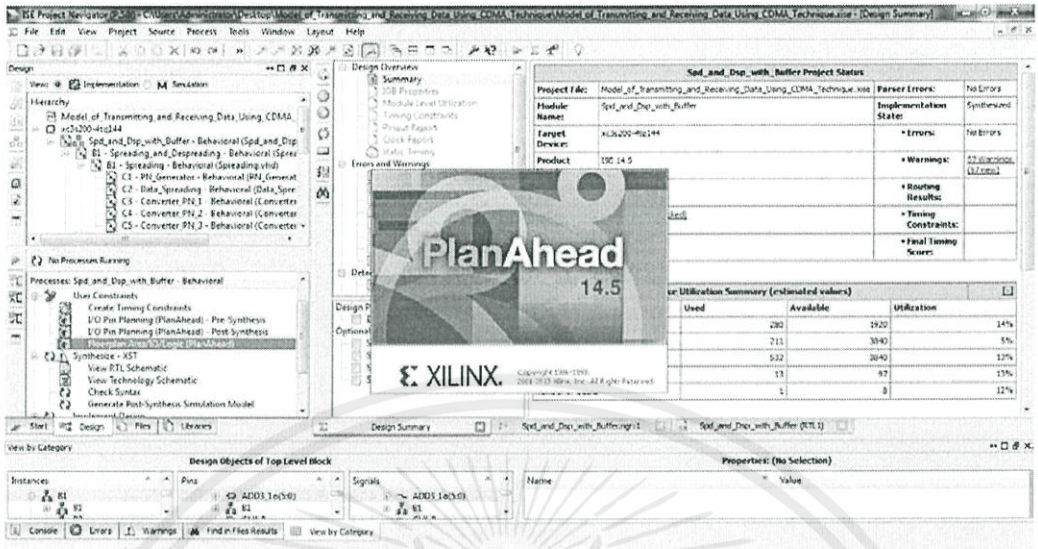
ในการทดสอบการจำลองระบบรับส่งข้อมูลผ่านทาง FPGA มีขั้นตอนดังนี้

1) เริ่มทำการกำหนดพอร์ตอินพุตและเอาท์พุตก่อน โดยคลิกเลือกที่ Floorplan Area/IO/Logic ดังแสดงในรูปที่ 3.37 เพื่อเรียกโปรแกรม PlanAhead



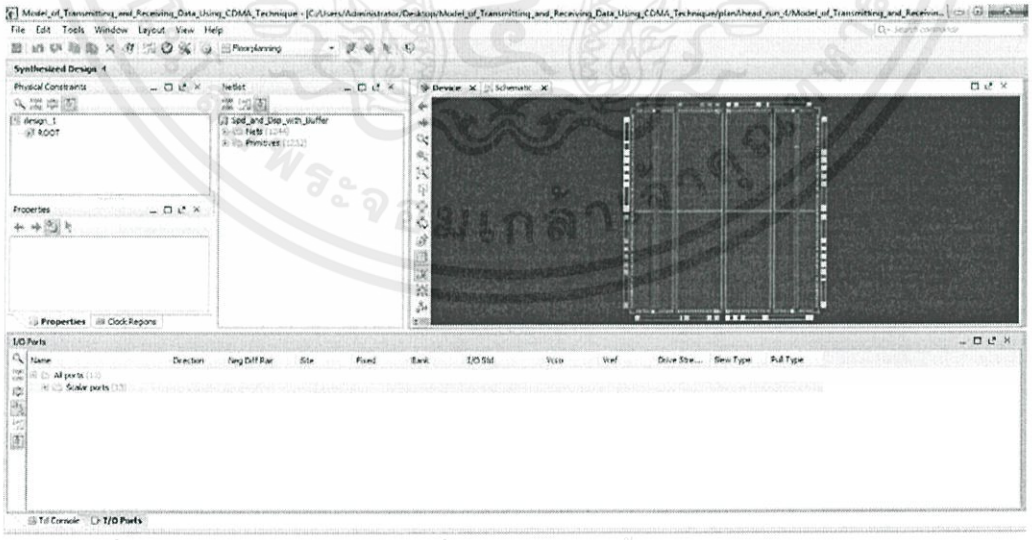
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.36 การเลือกโปรแกรมกำหนดพอร์ต



รูปที่ 3.37 โปรแกรม PlanAhead ที่เรียกขึ้นมา

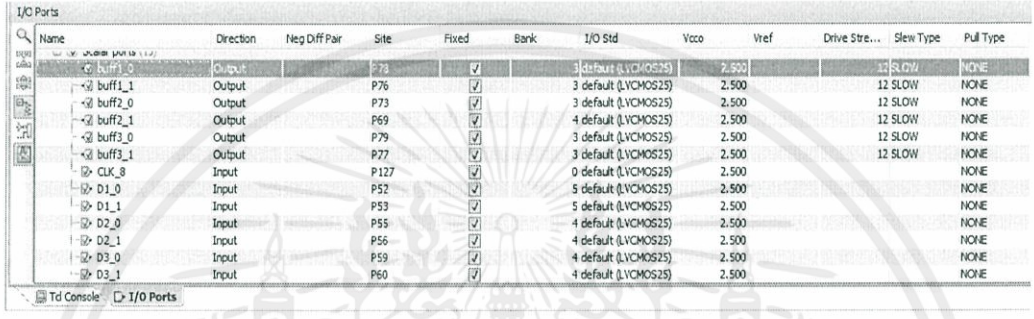
2) เมื่อทำการเปิดโปรแกรม PlanAhead จะพบหน้าต่างชื่อ I/O Ports อยู่บริเวณส่วนล่างของโปรแกรมจะเป็นบริเวณที่ใช้กำหนดพอร์ตต่างๆ ดังรูปที่ 3.39 โดยจะพบว่าพอร์ตที่ต้องทำการกำหนดมีอยู่ 13 พอร์ต คือพอร์ตอินพุต 6 พอร์ต ได้แก่ D1_0 , D1_1 , D2_0 , D2_1 , D3_0 และ D3_1 พอร์ตเอาท์พุต 6 พอร์ต ได้แก่ buff1_0 , buff1_1 , buff2_0 , buff2_1 , buff3_0 และ buff3_1 พอร์ตสัญญาณนาฬิกา 1 พอร์ต คือ CLK_8



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

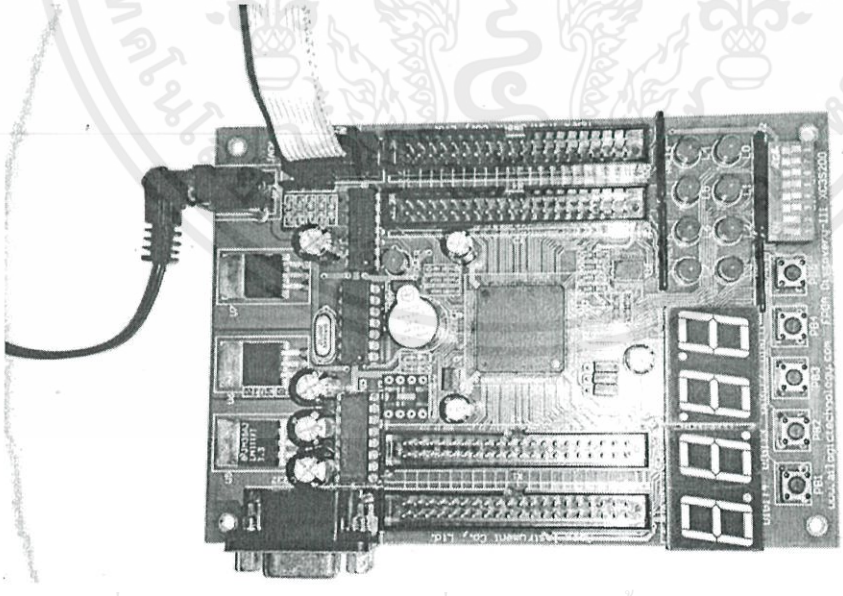
รูปที่ 3.38 ลักษณะโปรแกรม PlanAhead ที่เรียกขึ้นมา

3) ทำการกำหนดพอร์ตอินพุตข้อมูล 8 บิตได้แก่พอร์ต D1_0 , D1_1 , D2_0 , D2_1 , D3_0 และ D3_1 เข้าที่พอร์ต P52, P53, P55, P56, P59 และ P60 ตามลำดับ กำหนดพอร์ตสัญญาณนาฬิกา CLK_8 เข้าที่พอร์ต P127 และกำหนดพอร์ตเอาท์พุตได้แก่พอร์ต buff1_0 , buff1_1 , buff2_0 , buff2_1 , buff3_0 และ buff3_1 ออกที่พอร์ต P78, P76, P73, P69, P79 และ P77 ตามลำดับ โดยกำหนดลงในคอลัมน์ชื่อ site ดังรูปที่ 3.40



รูปที่ 3.39 การกำหนดพอร์ต

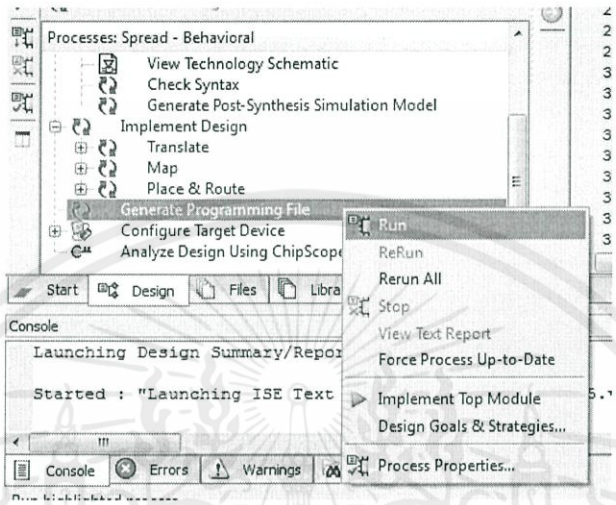
4) เริ่มต้นทำการโปรแกรมลงบน FPGA จะต้องต่อสาย JTAC และ ไฟเลี้ยงให้กับ FPGA ดังรูปที่ 3.41



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

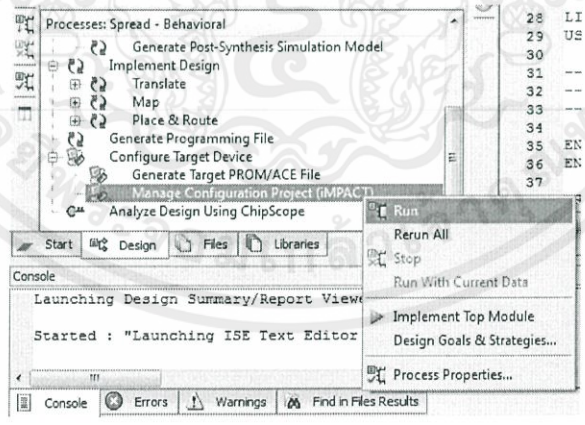
รูปที่ 3.40 FPGA พร้อมทำการโปรแกรม

5) คลิกที่ Generate Programming File เพื่อทำการ Implement Design ดังรูปที่ 3.41



รูปที่ 3.41 การ Implement Design

6) จากนั้นคลิกที่ Manage Configuration Project (iMPACT) ดังรูปที่ 3.42 เพื่อเปิดโปรแกรม iMPACT ขึ้นมา ดังรูปที่ 3.43

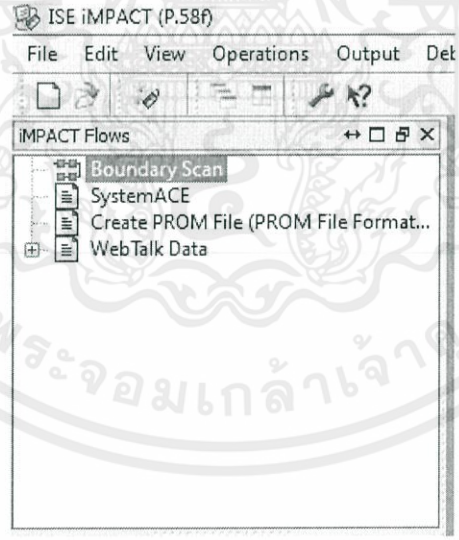


รูปที่ 3.42 การเลือกเปิดโปรแกรม iMPACT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

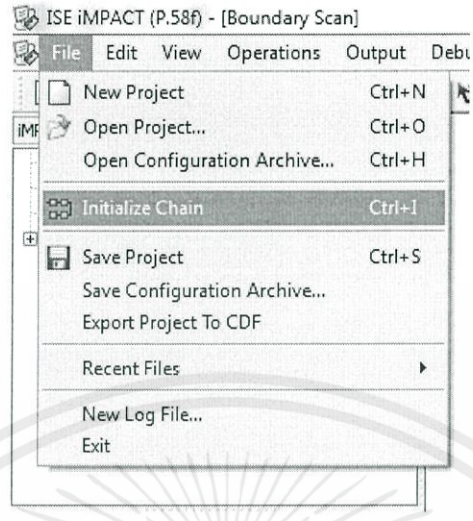
รูปที่ 3.43 โปรแกรม iMPACT

7) จากนั้นเลือกที่ Boundary Scan ดังรูปที่ 3.44 แล้วคลิกที่เมนู File เลือก Initialize Chain ดังรูปที่ 3.45



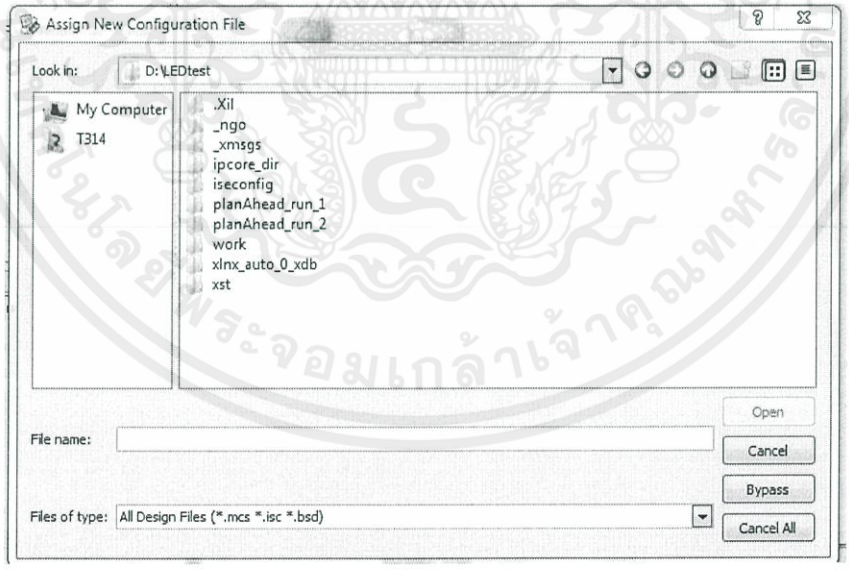
รูปที่ 3.44 การเลือก Boundary Scan

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



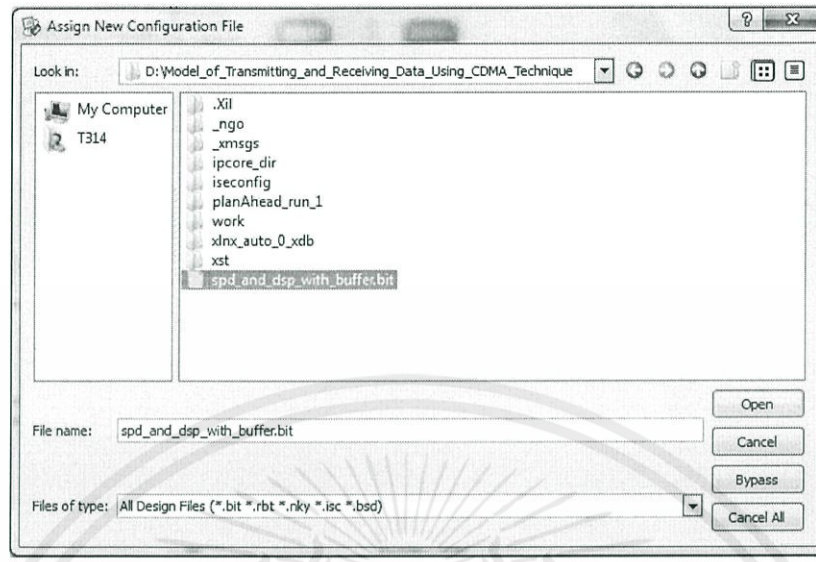
รูปที่ 3.45 การเลือก Initialize Chain

8) หลังจากนั้นจะได้หน้าต่างดังรูปที่ 3.46 จากนั้นกดที่ปุ่ม Bypass จะได้หน้าต่างให้กดเลือกไฟล์นามสกุล .bit ของโครงงานนี้ ชื่อไฟล์ spd_and_dsp_with_buffer.bit ที่จะนำมาเบิร์นลงบน FPGA หลังจากนั้นกด open ดังรูปที่ 3.47



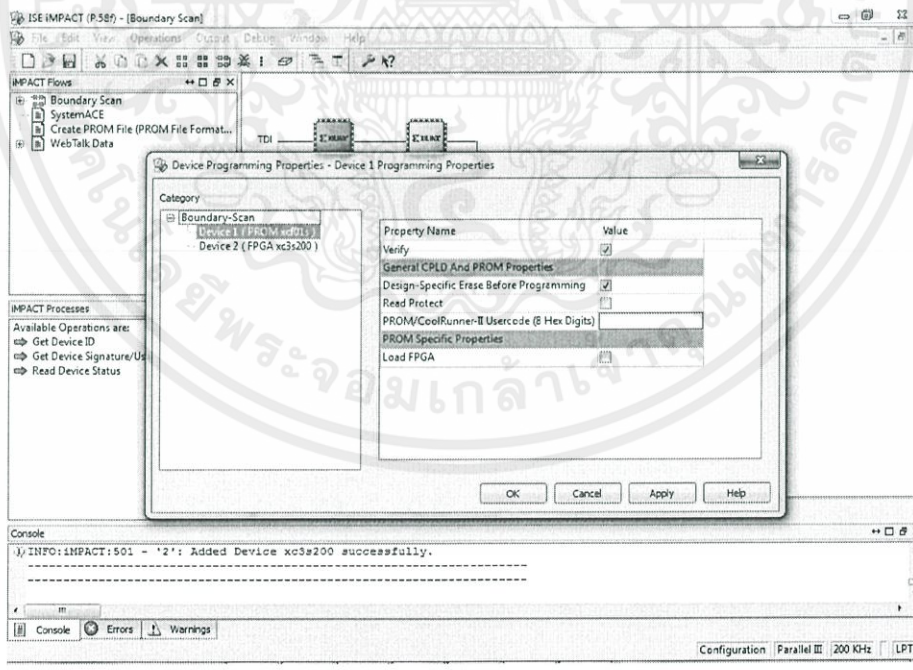
รูปที่ 3.46 หน้าต่าง Assign New Configuration File

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



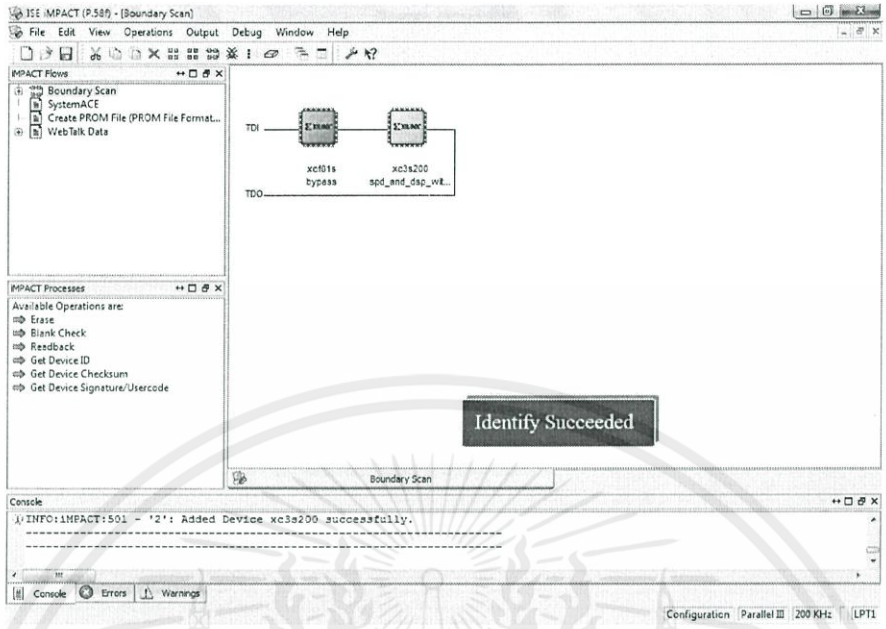
รูปที่ 3.47 หน้าต่างเลือกไฟล์โปรแกรมที่จะนำมาเบิร์นลงบน FPGA

9) จะได้หน้าต่างกำหนดคุณสมบัติของอุปกรณ์ภายใน FPGA ดังรูปที่ 3.48 ให้กด OK จะปรากฏดังรูปที่ 3.49 นั้น หมายถึงการกำหนดคุณสมบัติของอุปกรณ์สำเร็จ



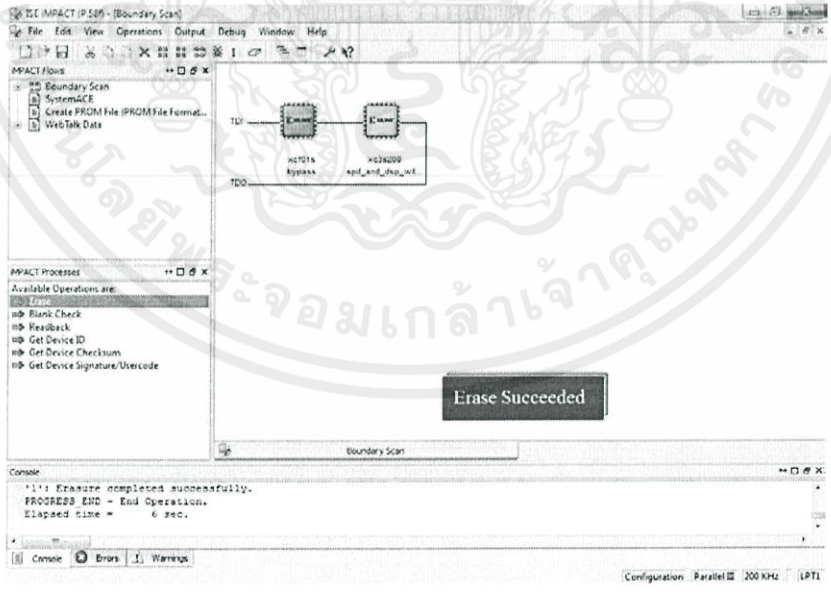
รูปที่ 3.48 หน้าต่างกำหนดคุณสมบัติของอุปกรณ์ภายใน FPGA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.49 หน้าต่างกำหนดอุปกรณ์ภายใน FPGA

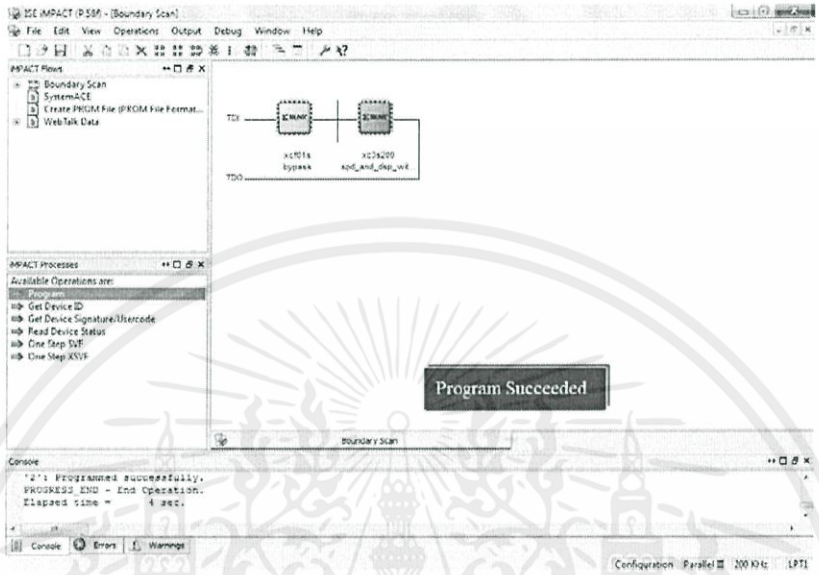
10) ให้กดเลือกอุปกรณ์ทางด้านซ้ายมือ แล้วกด Erase เพื่อทำการลบโปรแกรมที่อยู่ใน FPGA ก่อน ดังแสดงในรูปที่ 3.51



รูปที่ 3.50 การลบโปรแกรมที่อยู่ใน FPGA

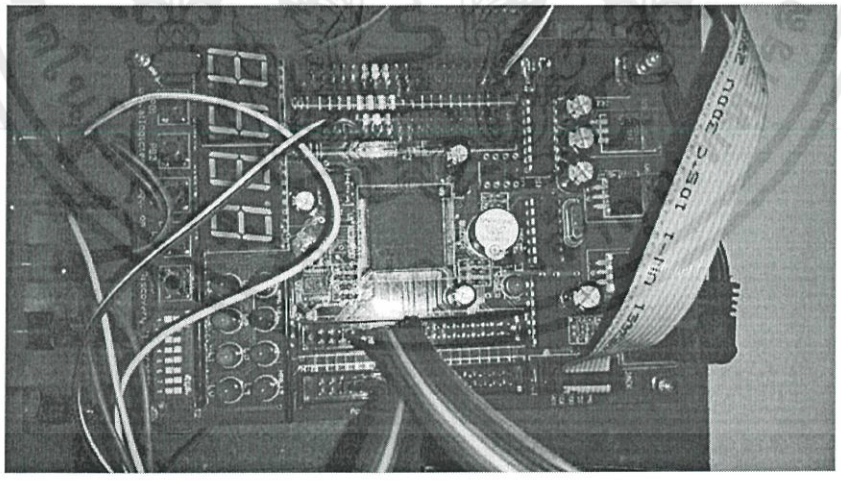
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11) หลังจากนั้นให้กดเลือกอุปกรณ์ทางด้านขวา เพื่อทำการเบิร์นโปรแกรม โดยกดที่ Program ดังรูปที่ 3.51



รูปที่ 3.51 การโปรแกรมวงจรลงบน FPGA สำเร็จ

12) หลังจากการโปรแกรมวงจรสำเร็จ แสดงดังรูปที่ 3.52



รูปที่ 3.52 FPGA ที่โปรแกรมวงจรสำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 ผลการทดลองด้วยโปรแกรม MATLAB

จากการออกแบบส่วนของเครื่องส่งและเครื่องรับ แบบสเปรดสเปกตรัม ในโปรแกรม MATLAB ได้ทำการออกแบบแต่ละส่วนให้ทำงานตามที่ต้องการโดยใช้ฟังก์ชันของการจำลอง โดยการจำลองผลการทำงานของส่วนต่างๆของเครื่องรับเครื่องส่ง

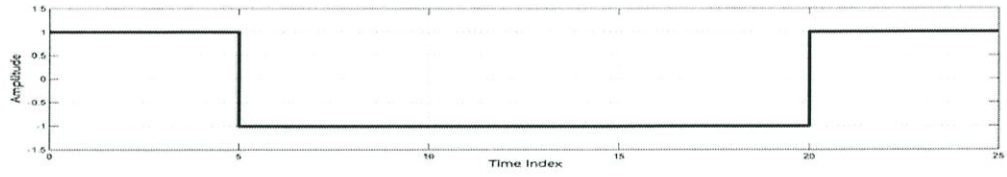
4.1.1 ภาคส่ง

จากการออกแบบส่วนของเครื่องส่งแบบสเปรดสเปกตรัมโดยการจำลองรูปแบบในโปรแกรม MATLAB ได้ทำการกำหนดผู้ใช้ในการส่งข้อมูลทั้งหมด 3 ราย โดยการให้ผู้ใช้แต่ละรายเป็นอิสระต่อกัน และจะใช้รหัสพีเอ็นในการสเปรดสเปกตรัม ด้วยการนำรหัสพีเอ็นดังกล่าวที่มีจำนวนอัตราบิตที่สูงกว่าอัตราของการส่งข้อมูลของผู้ใช้แต่ละราย ซึ่งผู้ใช้แต่ละราย จะมีรหัสพีเอ็นที่แตกต่างกัน เพื่อเป็นการจำแนกข้อมูลที่ส่งในแต่ละรายออกจากกัน

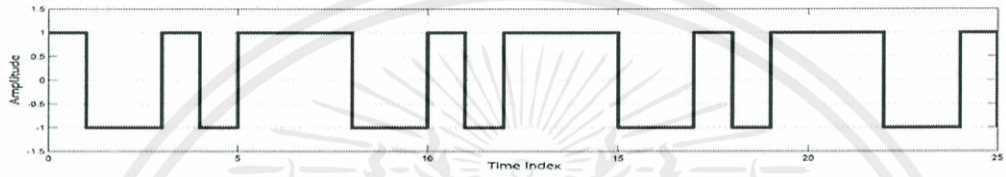
4.1.1.1 สัญญาณข้อมูลที่ส่งจากผู้ใช้รายที่ 1

ในส่วนของการส่งสัญญาณของผู้ใช้รายที่ 1 โดยจะทำการจำลองบิตข้อมูลด้วยการสุ่มบิต 0 หรือ บิต 1 เพื่อส่งข้อมูลออกด้วยอัตราความเร็วค่าหนึ่ง แล้วทำการแม็ปค่าข้อมูลจาก บิตที่เท่ากับ 0 ให้มีค่าเท่ากับ -1 และค่าของข้อมูลที่เป็น 1 ให้มีค่าเป็น 1 เช่นเดิม ซึ่งหลังจากการแม็ปค่าของข้อมูลแล้ว จะนำข้อมูลที่ทำการแม็ปมาคูณกับรหัสพีเอ็นของผู้ใช้รายที่ 1 ที่มีอัตราเร็วในการส่ง เป็นจำนวนเท่าของอัตราเร็วในการส่งบิตข้อมูล จากรูปที่ 4.1 จะกำหนดให้รหัสพีเอ็นของผู้ใช้รายที่ 1 ที่มีอัตราเร็วในการส่ง 5 เท่าของอัตราเร็วในการส่งบิตข้อมูลของ ผู้ใช้รายที่ 1 ซึ่งผลจากการคูณข้อมูลกับรหัสพีเอ็นของผู้ใช้รายที่ 1 จะแสดงได้ดังรูปที่ 4.1

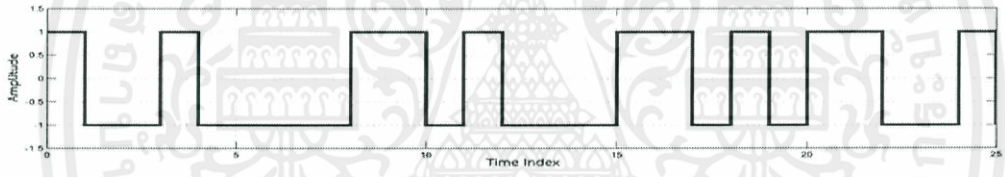
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) สัญญาณข้อมูลในการสุ่มของผู้ใช้รายที่ 1



ข) รหัสพีเอ็นของผู้ใช้รายที่ 1

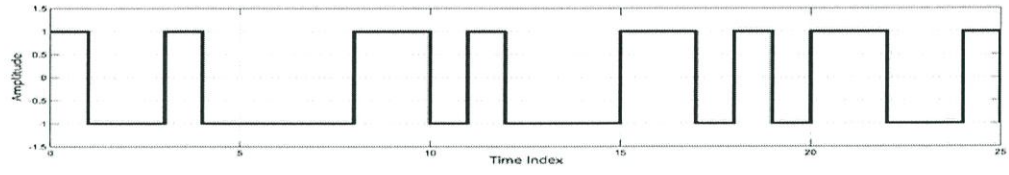


ค) ข้อมูลที่ถูกสเปรดจากรหัสพีเอ็นของผู้ใช้รายที่ 1

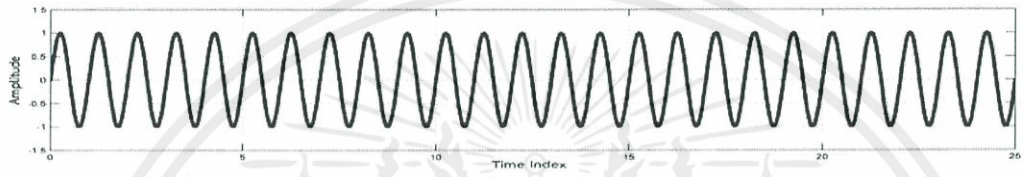
รูปที่ 4.1 การทำการสเปรดของข้อมูลผู้ใช้รายที่ 1

ซึ่งจะเห็นได้ว่ารหัสพีเอ็น ของผู้ใช้รายที่ 1 เมื่อทำการคูณกับสัญญาณข้อมูลแล้ว จะทำให้ข้อมูลในการส่งอยู่ในรูปแบบของการสเปรด ซึ่งหลักจากนั้นจะนำข้อมูลที่ถูกสเปรดไปคูณกับคลื่นพาหะ ซึ่งกล่าวได้ว่าเป็นการมอดูเลตในรูปแบบของ Binary Phase Shift Keying (BPSK) แล้วถูกส่งออกไปยังช่องสัญญาณต่อไป ซึ่งจะแสดงได้ดังรูปที่ 4.2

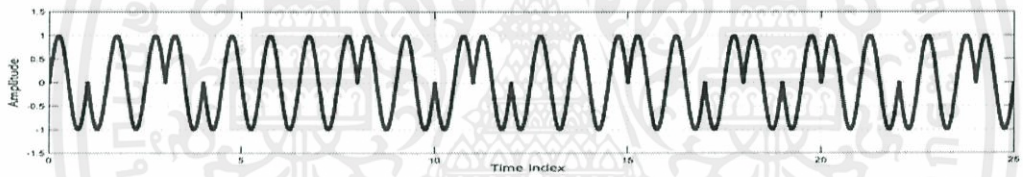
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) ข้อมูลที่ถูกสเปรดจากรหัสพีเอ็นของผู้ใช้รายที่ 1



ข) สัญญาณพาหะ



ค) สัญญาณที่ถูกมอดูเลต แบบ BPSK ของผู้ใช้รายที่ 1

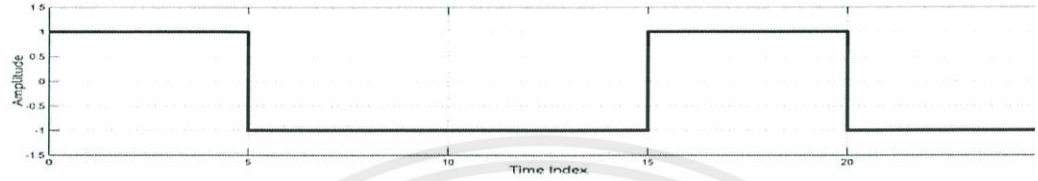
รูปที่ 4.2 ข้อมูลที่ถูกมอดูเลตแบบ BPSK ของผู้ใช้รายที่ 1

4.1.1.2 สัญญาณข้อมูลที่ส่งจากผู้ใช้รายที่ 2

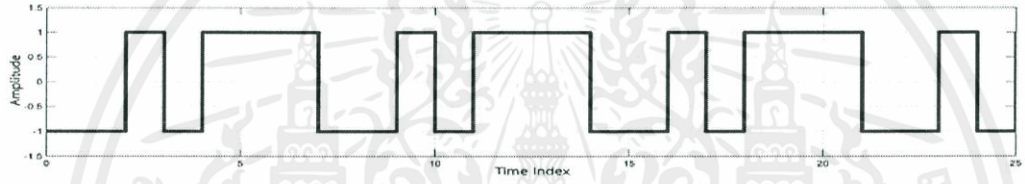
ในส่วนของการส่งสัญญาณของผู้ใช้รายที่ 2 โดยจะทำการจำลองบิตข้อมูลด้วยการสุ่มบิต 0 หรือ บิต 1 เพื่อส่งข้อมูลออกด้วยอัตราความเร็วค่าหนึ่ง แล้วทำการแม็ปค่าข้อมูลจาก บิตที่เท่ากับ 0 ให้มีค่าเท่ากับ -1 และค่าของข้อมูลที่เป็น 1 ให้มีค่าเป็น 1 เช่นเดิม ซึ่งหลังจากการแม็ปค่าของข้อมูลแล้ว จะนำข้อมูลที่ทำการแม็ปมาคูณกับรหัสพีเอ็นของผู้ใช้รายที่ 2 ที่มีอัตราเร็วในการส่ง เป็นจำนวนเท่าของอัตราเร็วในการส่งบิตข้อมูล จากรูปที่ 4.3 จะกำหนดให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

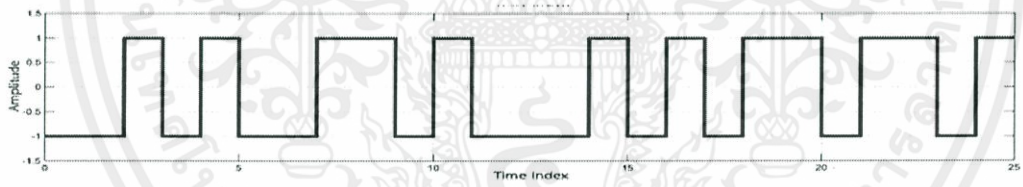
รหัสพีเอ็นของผู้ใช้รายที่ 2 ที่มีอัตราเร็วในการส่ง 5 เท่าของอัตราเร็วในการส่งบิตข้อมูลของ ผู้ใช้ราย ที่ 2 ซึ่งผลจาการคูณข้อมูลกับรหัสพีเอ็นของผู้ใช้รายที่ 2 จะแสดงได้ดังรูปที่ 4.3



ก) สัญญาณข้อมูลในการสุ่มของผู้ใช้รายที่ 2



ข) รหัสพีเอ็นของผู้ใช้รายที่ 2

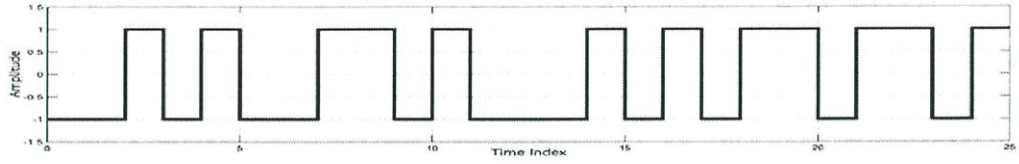


ค) ข้อมูลที่ถูกสเปรดจากรหัสพีเอ็นของผู้ใช้รายที่ 2

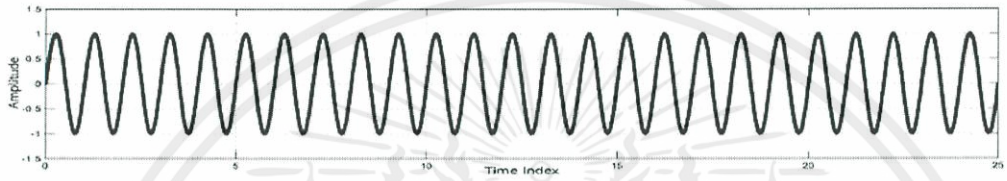
รูปที่ 4.3 การทำการสเปรดของข้อมูลผู้ใช้รายที่ 2

ซึ่งจะเห็นได้ว่ารหัสพีเอ็น ของผู้ใช้รายที่ 2 เมื่อทำการคูณกับสัญญาณข้อมูลแล้ว จะทำให้ข้อมูลในการส่งอยู่ในรูปแบบของการสเปรด ซึ่งหลักจากนั้นจะนำข้อมูลที่ถูกสเปรดไปคูณกับคลื่นพาหะ ซึ่งกล่าวได้ว่าเป็นการมอดูเลตในรูปแบบของ Binary Phase Shift Keying (BPSK) แล้วถูกส่งออกไปยังช่องสัญญาณต่อไป ซึ่งจะแสดงได้ดังรูปที่ 4.4

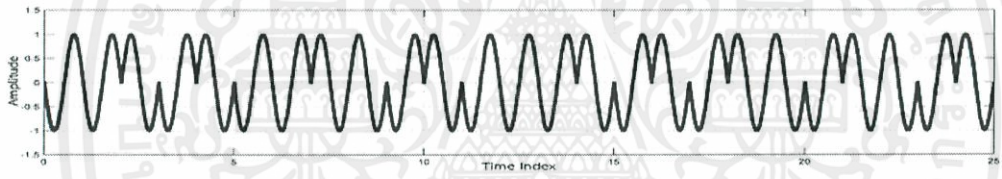
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) ข้อมูลที่ถูกสเปรดจากรหัสพีเอ็นของผู้ใช้รายที่ 2



ข) สัญญาณพาหะ



ค) สัญญาณที่ถูกมอดูเลต แบบ BPSK ของผู้ใช้รายที่ 2

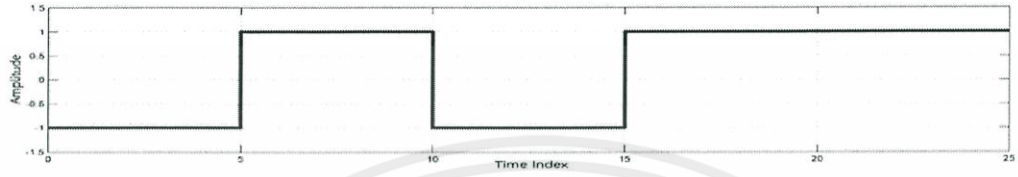
รูปที่ 4.4 ข้อมูลที่ถูกมอดูเลตแบบ BPSK ของผู้ใช้รายที่2

4.1.1.3 สัญญาณข้อมูลที่ส่งจากผู้ใช้รายที่ 3

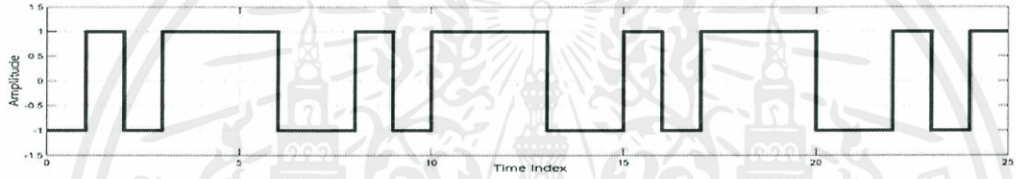
ในส่วนของการส่งสัญญาณของผู้ใช้รายที่ 3 โดยจะทำการจำลองบิตข้อมูลด้วยการสุ่มบิต 0 หรือ บิต 1 เพื่อส่งข้อมูลออกด้วยอัตราความเร็วค่าหนึ่ง แล้วทำการแม็ปค่าข้อมูลจาก บิตที่เท่ากับ 0 ให้มีค่าเท่ากับ -1 และค่าของข้อมูลที่เป็น 1 ให้มีค่าเป็น 1 เช่นเดิม ซึ่งหลังจากการแม็ปค่าของข้อมูลแล้ว จะนำข้อมูลที่ทำการแม็ปมาคูณรหัสพีเอ็นของผู้ใช้รายที่ 3 ที่มีอัตราเร็วในการส่ง เป็นจำนวนเท่าของอัตราเร็วในการส่งบิตข้อมูล จากรูปที่ 4.5 จะกำหนดให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสพีเอ็นของผู้ใช้รายที่ 3 ที่มีอัตราเร็วในการส่ง 5 เท่าของอัตราเร็วในการส่งบิตข้อมูลของ ผู้ใช้ราย ที่ 3 ซึ่งผลจาการคูณข้อมูลกับรหัสพีเอ็นของผู้ใช้รายที่ 3 จะแสดงได้ดังรูปที่ 4.5



ก) สัญญาณข้อมูลในการสุ่มของผู้ใช้รายที่ 3



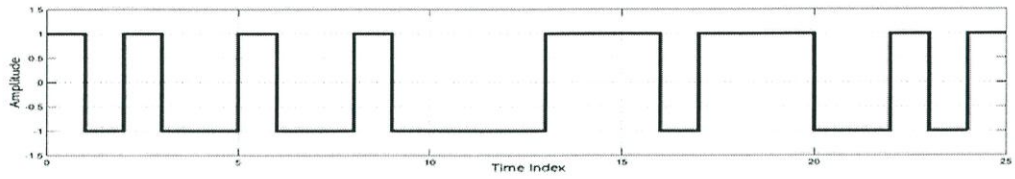
ข) รหัสพีเอ็นของผู้ใช้รายที่ 3

ข) ข้อมูลที่ถูกสเปรดจากรหัสพีเอ็นของผู้ใช้รายที่ 3

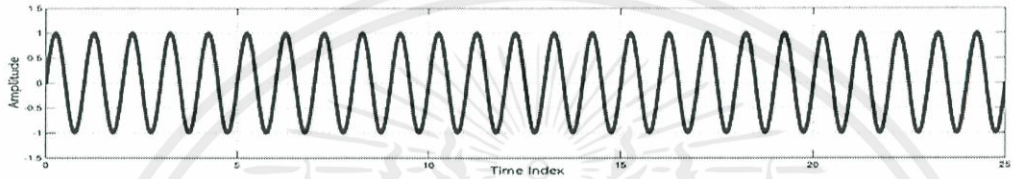
รูปที่ 4.5 การทำการสเปรดของข้อมูลผู้ใช้รายที่ 3

ซึ่งจะเห็นได้ว่ารหัสพีเอ็น ของผู้ใช้รายที่ 3 เมื่อทำการคูณกับสัญญาณข้อมูลแล้ว จะทำให้ข้อมูลในการส่งอยู่ในรูปแบบของการสเปรด ซึ่งหลักจากนั้นจะนำข้อมูลที่ถูกสเปรดไปคูณกับคลื่นพาหะ ซึ่งกล่าวได้ว่าเป็นการมอดูเลตในรูปแบบของ Binary Phase Shift Keying (BPSK) แล้วถูกส่งออกไปยังช่องสัญญาณต่อไป ซึ่งจะแสดงได้ดังรูปที่ 4.6

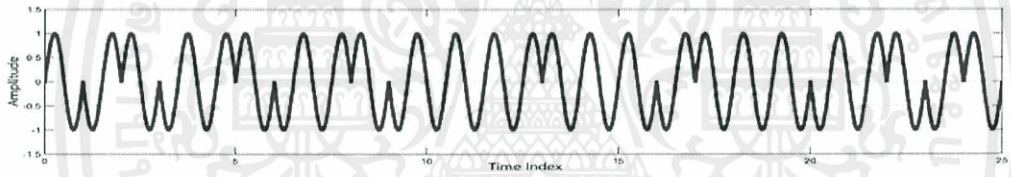
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) ข้อมูลที่ถูกสเปรดจากรหัสพีเอ็นของผู้ใช้รายที่ 3



ข) สัญญาณพาหะ



ค) สัญญาณที่ถูกมอดูเลต แบบ BPSK ของผู้ใช้รายที่ 3

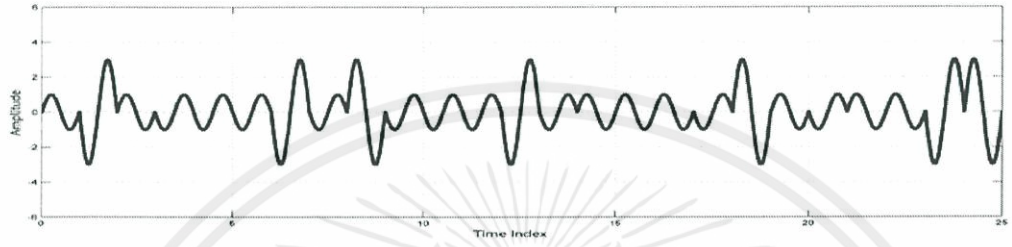
รูปที่ 4.6 ข้อมูลที่ถูกมอดูเลตแบบ BPSK ของผู้ใช้รายที่ 3

4.1.1.4 การส่งข้อมูลของผู้ใช้ทั้ง 3 ราย ผ่านช่องสัญญาณ

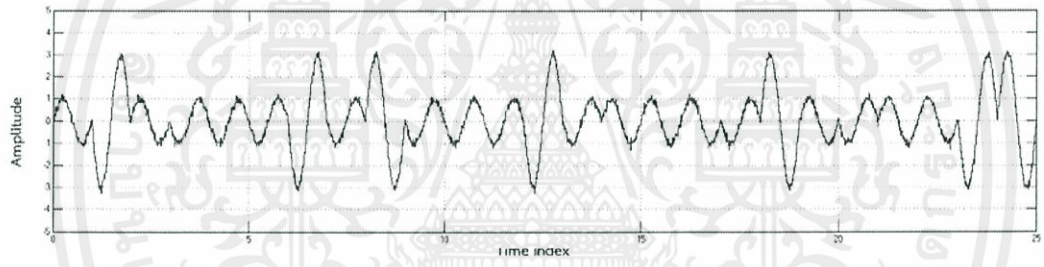
จากการจำลองการส่งข้อมูลของผู้ใช้แต่ละราย ผ่านโปรแกรม MATLAB ซึ่งจะได้สัญญาณของผู้ใช้แต่ละรายที่ใช้ในการส่ง จะถูกมอดูเลตในรูปแบบของ BPSK แล้วทำการรวมสัญญาณของแต่ละผู้ใช้ในอากาศ ซึ่งในแต่ละผู้ใช้ จะมีลักษณะเฉพาะของข้อมูลโดยการถูกกำหนดจากรหัสพีเอ็นของผู้ใช้แต่ละราย และข้อมูลหลังจากที่ถูกการมอดูเลต ของผู้ใช้แต่ละรายจะถูกรวมสัญญาณในอากาศและส่งผ่านช่องสัญญาณที่มีการรบกวนจากสัญญาณรบกวน ซึ่งจากการจำลองด้วยโปรแกรม MATLAB ได้กำหนดให้ข้อมูลที่ถูกส่งมาถูกรบกวนจากสัญญาณรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เกาส์เซียนสีขาว (Additive White Gaussian Noise, AWGN) และจากในบล็อก ได้ทำการกำหนดให้ค่าความแปรปรวน มีค่าเท่ากับ 0.01 เพื่อทำให้สัญญาณที่อยู่ในช่องสัญญาณมีรูปร่างผิดเพี้ยนไปจากเดิม แสดงได้ดังรูปที่ 4.7



ก) สัญญาณข้อมูลของผู้ใช้ทั้ง 3 ราย ที่ถูกรวมในช่องสัญญาณ



ข) สัญญาณข้อมูลของผู้ใช้ทั้ง 3 ราย ที่ถูกรวมในช่องสัญญาณและ
ถูกรบกวนด้วย สัญญาณรบกวนเกาส์เซียนสีขาว

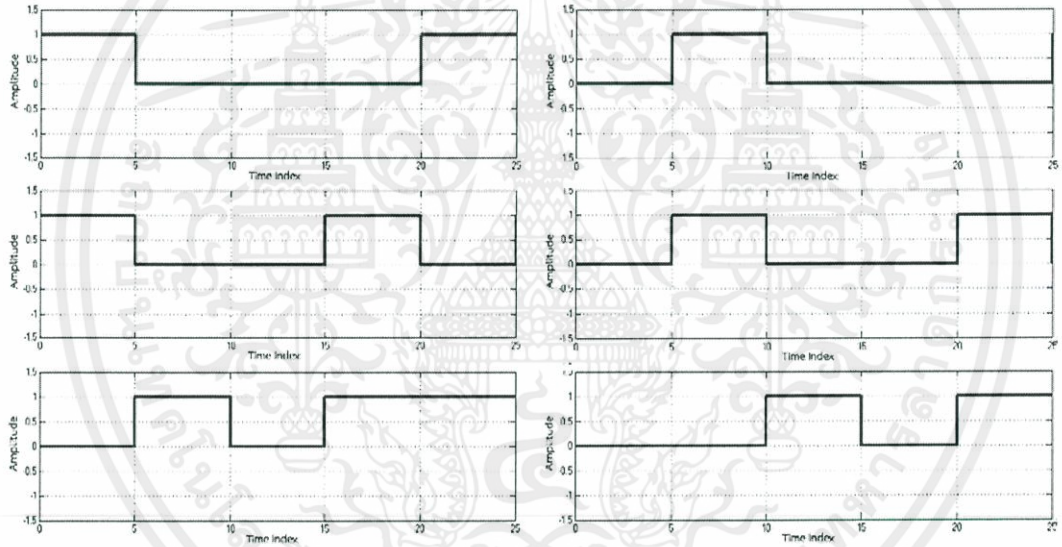
รูปที่ 4.7 สัญญาณก่อนการส่ง

4.1.2 ภาครับ

จากการออกแบบในส่วนของเครื่องรับที่ใช้ คอร์รีเลเตอร์ (Correlator) โดยการจำลองรูปแบบด้วยโปรแกรม MATLAB ได้ทำการรับข้อมูลจากช่องสัญญาณที่ถูกรวมกันของผู้ใช้ทั้ง 3 ราย โดยผ่านช่องสัญญาณ ทำให้มีรูปแบบของสัญญาณที่ผิดเพี้ยนไปจากเดิม โดยจะต้องทำการออกแบบในส่วนของเครื่องรับ เพื่อให้รับข้อมูลกลับคืนมาได้อย่างถูกต้อง และเครื่องรับของผู้ใช้แต่ละราย จะต้องมีรหัสพีเอ็นที่เหมือนกันกับรหัสพีเอ็นของภาคส่ง ซึ่งจะนำรหัสพีเอ็นดังกล่าว ของแต่ละ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผู้ใช้ ไปคูณกับสัญญาณรวม เพื่อทำการดีสเปรดแล้วจำแนกสัญญาณออกมา หลังจากนั้นก็นำไปเข้าสู่กระบวนการบวกสะสมค่า และทำการตัดสินใจค่าของข้อมูล ซึ่งข้อมูลที่ได้จะอยู่ในรูปของ บิต 0 หรือ บิต 1 โดยค่าที่ได้จากการบวกสะสมค่านั้น ถ้ามีค่ามากกว่า 0 จะถูกตัดสินใจให้บิตข้อมูลมีค่าเป็น 1 และถ้าค่าที่ได้จากการบวกสะสมค่า มีค่าน้อยกว่า 0 จะถูกตัดสินใจให้บิตข้อมูลมีค่าเป็น 0 โดยเครื่องรับที่ดีควรจะรับสัญญาณข้อมูลได้ตรงกับข้อมูลที่ถูกส่งมา ซึ่งจากการจำลองในการส่งข้อมูลและรับข้อมูล จะเห็นว่าข้อมูลจะถูกเลื่อนไป 1 บิตในวงจรของภาครับ และภาครับสามารถจำแนกข้อมูลของผู้ใช้แต่ละราย จากช่องสัญญาณรวมที่มีสัญญาณรบกวนเกาส์เซียน แล้วสามารถตัดสินใจให้มีค่าตรงกับบิตข้อมูลที่ส่งมา ซึ่งจะแสดงการรับข้อมูลของผู้ใช้แต่ละราย ดังรูปที่ 4.8 โดยเป็นการเปรียบเทียบการรับส่งข้อมูลของผู้ใช้แต่ละราย

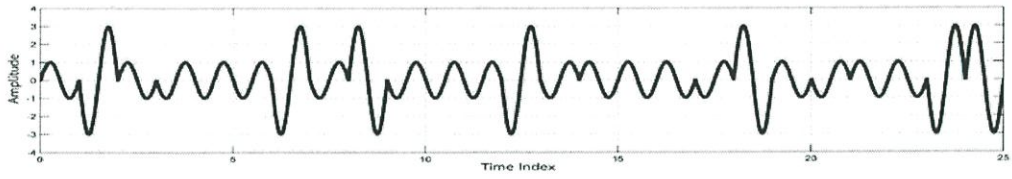


รูปที่ 4.8 การเปรียบเทียบการรับส่งข้อมูลของผู้ใช้แต่ละราย

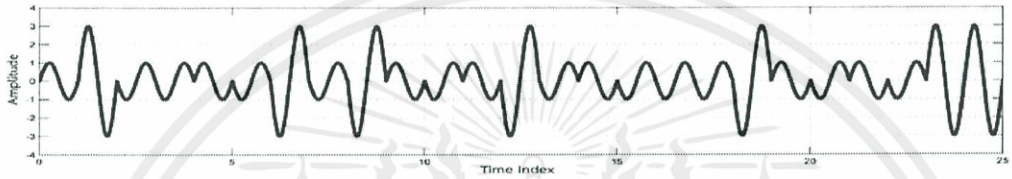
4.1.2.1 สัญญาณข้อมูลที่รับของผู้ใช้รายที่ 1

ในส่วนของภาครับ การรับสัญญาณข้อมูลของ ผู้ใช้รายที่ 1 จะทำการรับสัญญาณข้อมูลจากแล้วทำการคูณกับรหัสพีเอ็นของผู้ใช้รายที่ 1 ซึ่งจะเป็นรหัสเดียวกันกับภาคส่ง เพื่อจำแนกสัญญาณและดีสเปรด จากนั้นก็เข้าสู่ขั้นตอนของการบวกสะสมค่าและทำการตัดสินใจบิตข้อมูล ซึ่งในการรับสัญญาณนี้จะอธิบายได้ดังรูปที่ 4.9

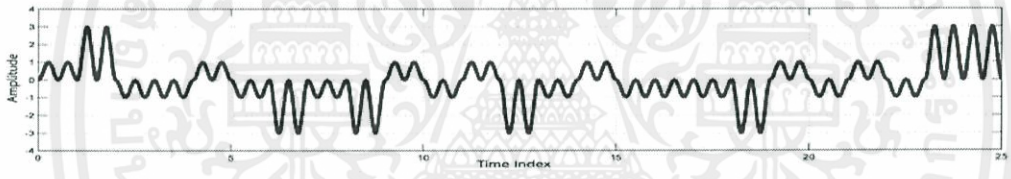
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) สัญญาณข้อมูลที่ได้รับจากช่องสัญญาณของผู้ใช้รายที่ 1



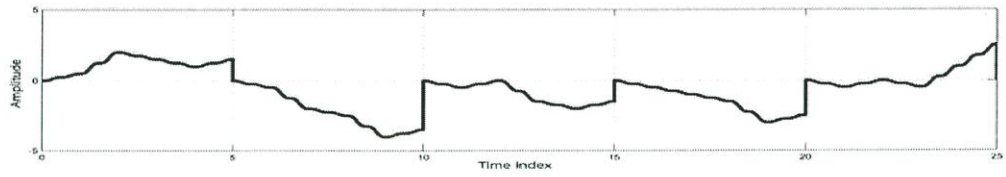
ข) สัญญาณข้อมูลที่ได้รับคูณกับรหัสพีเอ็นของผู้ใช้รายที่ 1



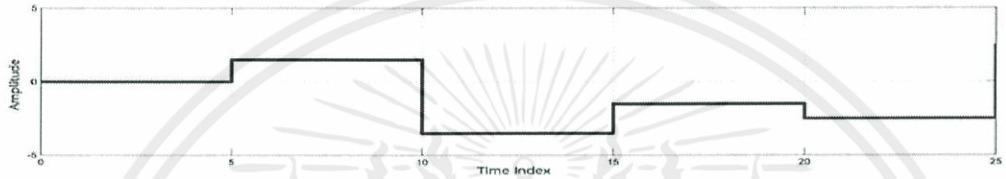
ค) การดีมอดูเลตสัญญาณข้อมูลของผู้ใช้รายที่ 1

รูปที่ 4.9 การดีสเปรดและดีมอดูเลตของข้อมูลผู้ใช้รายที่ 1

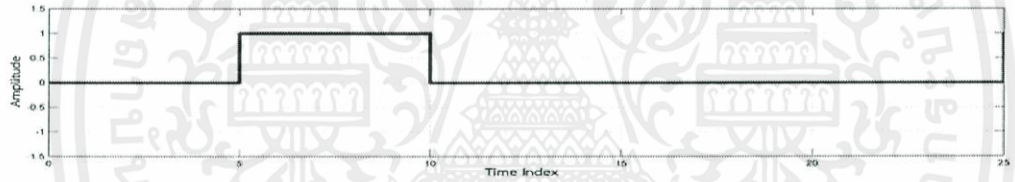
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) การบวกสะสมค่าของสัญญาณผู้ใช้รายที่ 1



ก) สัญญาณจากการจัดระดับค่าสัญญาณของผู้ใช้รายที่ 1



ข) สัญญาณข้อมูลหลังจากที่ผ่านขั้นตอนการตัดสินใจบิตของผู้ใช้รายที่ 1

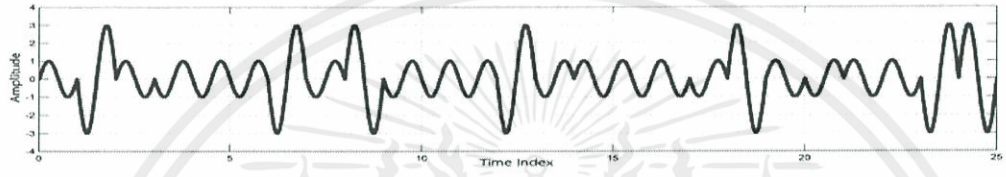
รูปที่ 4.10 การติดสินใจบิตข้อมูลของผู้ใช้รายที่ 1

จากรูปที่ 4.10 ก แสดงถึงค่าที่ได้รับจากการอินทีเกรตพื้นที่ใต้กราฟของสัญญาณข้อมูลที่ถูกคูณด้วยคลื่นพาหะ จะเห็นว่าค่าที่ได้รับสำหรับช่วงเวลาแต่ละบิต เป็นการรวมพลังงานของค่าที่ได้รับในแต่ละบิต ส่วนรูปที่ 4.10 ข คือสัญญาณที่ถูกจัดระดับค่าของสัญญาณ และรูปที่ 4.10 ค แสดงถึงบิตที่ได้จากการตัดสินใจค่า ซึ่งเป็นการตัดสินใจว่าแต่ละบิตที่ส่งมาคือบิตใด

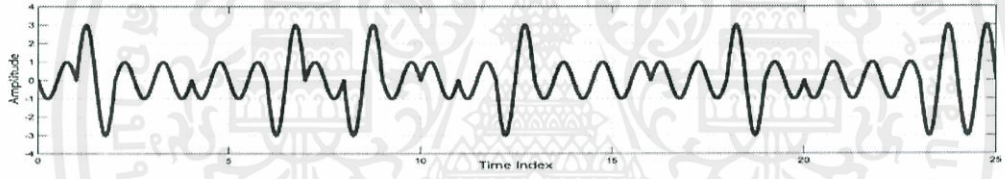
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2.2 สัญญาณข้อมูลที่รับของผู้ใช้รายที่ 2

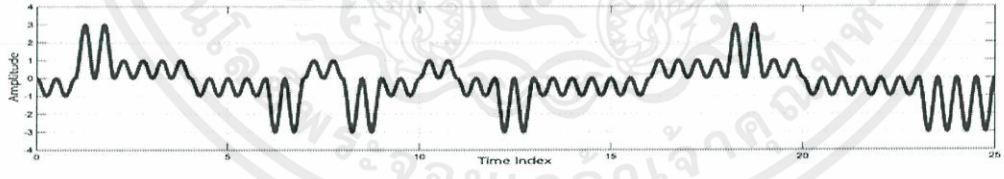
ในส่วนของภาครับ การรับสัญญาณข้อมูลของ ผู้ใช้รายที่ 2 จะทำการรับสัญญาณข้อมูลจากแล้วทำการคูณกับรหัสพีเอ็นของผู้ใช้รายที่ 2 ซึ่งจะเป็นรหัสเดียวกันกับภาคส่ง เพื่อจำแนกสัญญาณและดีสเปรด จากนั้นก็เข้าสู่ขั้นตอนของการบวกสะสมค่าและทำการตัดสินใจบิตข้อมูล ซึ่งในการรับสัญญาณนี้จะอธิบายได้ดังรูปที่ 4.11



ก) สัญญาณข้อมูลที่ได้รับจากช่องสัญญาณของผู้ใช้รายที่ 2



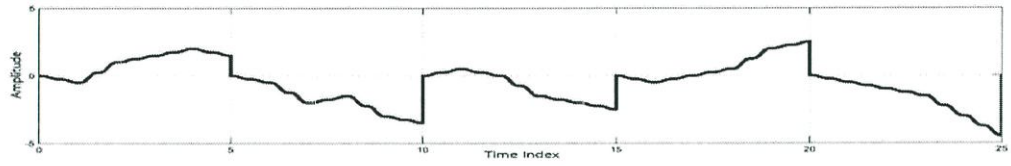
ข) สัญญาณข้อมูลที่ได้รับคูณกับรหัสพีเอ็นของผู้ใช้รายที่ 2



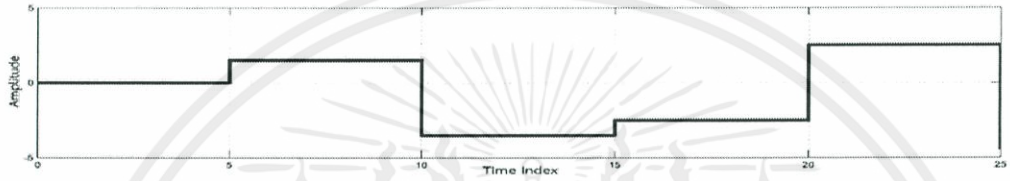
ค) การดีมอดูเลตสัญญาณข้อมูลของผู้ใช้รายที่ 2

รูปที่ 4.11 การดีสเปรดและดีมอดูเลตของข้อมูลผู้ใช้รายที่ 2

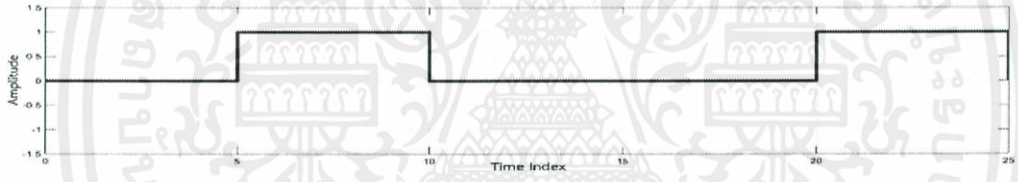
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) การบวกสะสมค่าของสัญญาณผู้ใช้รายที่ 2



ข) สัญญาณจากการจัดระดับค่าสัญญาณของผู้ใช้รายที่ 2



ค) สัญญาณข้อมูลหลังจากที่ผ่านขั้นตอนการตัดสินใจบิตของผู้ใช้รายที่ 2

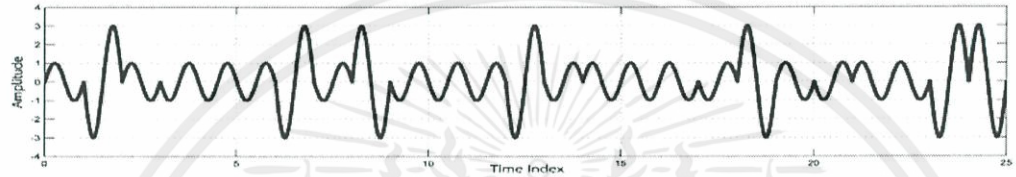
รูปที่ 4.12 การติดสินใจบิตข้อมูลของผู้ใช้รายที่ 2

จากรูปที่ 4.12 ก แสดงถึงค่าที่ได้รับจากการอินทีเกรตพื้นที่ใต้กราฟของสัญญาณข้อมูลที่ถูกคูณด้วยคลื่นพาหะ จะเห็นว่าค่าที่ได้รับสำหรับช่วงเวลาแต่ละบิต เป็นการรวมพลังงานของค่าที่ได้รับในแต่ละบิต ส่วนรูปที่ 4.12 ข คือสัญญาณที่ถูกจัดระดับค่าของสัญญาณ และรูปที่ 4.12 ค แสดงถึงบิตที่ได้จากการตัดสินใจค่า ซึ่งเป็นการตัดสินใจแต่ละบิตที่ส่งมาคือบิตใด

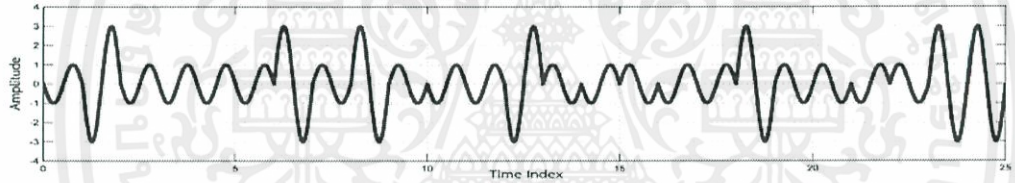
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2.3 สัญญาณข้อมูลที่รับของผู้ใช้รายที่ 3

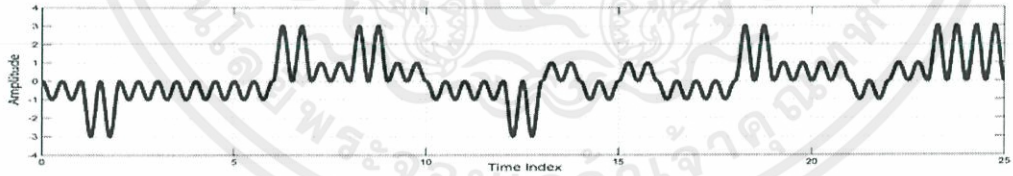
ในส่วนของภาครับ การรับสัญญาณข้อมูลของ ผู้ใช้รายที่ 3 จะทำการรับสัญญาณข้อมูลจากแล้วทำการคูณกับรหัสพีเอ็นของผู้ใช้รายที่ 3 ซึ่งจะเป็นรหัสเดียวกันกับภาคส่ง เพื่อจำแนกสัญญาณและดีสเปรด จากนั้นก็เข้าสู่ขั้นตอนของการบวกสะสมค่าและทำการตัดสินใจบิตข้อมูล ซึ่งในการรับสัญญาณนี้จะอธิบายได้ดังรูปที่ 4.13



ก) สัญญาณข้อมูลที่ได้รับจากช่องสัญญาณของผู้ใช้รายที่ 3



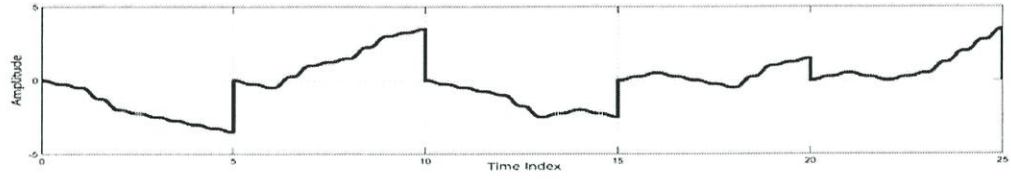
ข) สัญญาณข้อมูลที่ได้รับคูณกับรหัสพีเอ็นของผู้ใช้รายที่ 3



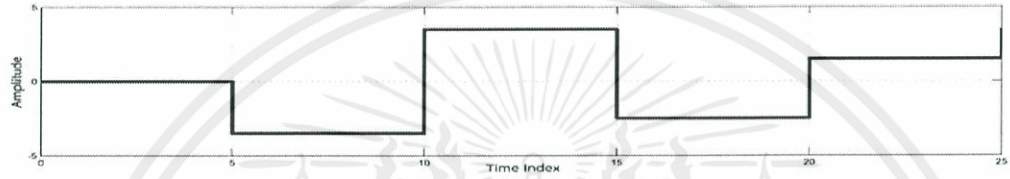
ค) การดีมอดูเลตสัญญาณข้อมูลของผู้ใช้รายที่ 3

รูปที่ 4.13 การดีสเปรดและดีมอดูเลตของข้อมูลผู้ใช้รายที่ 3

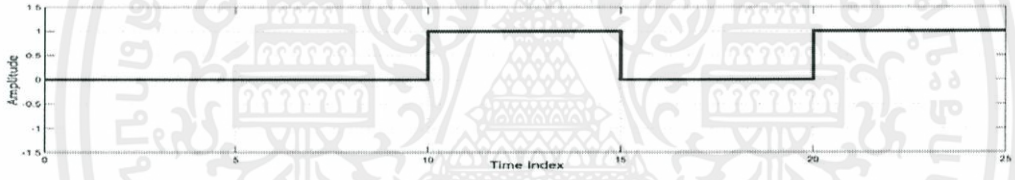
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ก) การบวกสะสมค่าของสัญญาณผู้ใช้รายที่ 3



ข) สัญญาณจากการจัดระดับค่าสัญญาณของผู้ใช้รายที่ 3



ค) สัญญาณข้อมูลหลังจากที่ผ่านขั้นตอนการตัดสินใจบิตของผู้ใช้รายที่ 3

รูปที่ 4.14 การติดสินใจบิตข้อมูลของผู้ใช้รายที่ 3

จากรูปที่ 4.14 ก แสดงถึงค่าที่ได้รับจากการอินทีเกรตพื้นที่ใต้กราฟของสัญญาณข้อมูลที่ถูกคูณด้วยคลื่นพาหะ จะเห็นว่าค่าที่ได้รับสำหรับช่วงเวลาแต่ละบิต เป็นการรวมพลังงานของค่าที่ได้รับในแต่ละบิต ส่วนรูปที่ 4.14 ข คือสัญญาณที่ถูกจัดระดับค่าของสัญญาณ และรูปที่ 4.14 ค แสดงถึงบิตที่ได้จากการตัดสินใจค่า ซึ่งเป็นการตัดสินใจว่าแต่ละบิตที่ส่งมาคือบิตใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

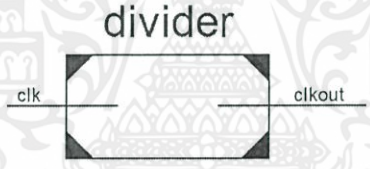
4.2 ผลการทดลองด้วยภาษา VHDL ผ่านโปรแกรม MODELSIM 10.1b

จากการออกแบบส่วนของเครื่องส่งและเครื่องรับแบบสเปรดสเปกตรัม ได้ทำการออกแบบแต่ละส่วนให้ทำงานตามที่ต้องการโดยใช้ภาษา VHDL ในการเขียนชุดคำสั่ง รวมถึงใช้การออกแบบผ่านทางโปรแกรม XILINX ISE 14.5 แล้วทำการจำลองผลการทำงานด้วยโปรแกรม MODELSIM 10.1b เพื่อจำลองการทำงานส่วนต่างๆ ของเครื่องรับและเครื่องส่ง ด้วย FPGA เบอร์ XC3S200F

4.2.1 ภาคส่ง

จากบทที่ 3 ได้กล่าวถึงการออกแบบการทดลอง ส่วนในบทนี้จะกล่าวถึงผลการทดลองของวงจรในส่วนต่างๆของภาคส่งที่ใช้ในระบบการจำลองการสื่อสารแบบไดเรกต์ซีเควนซ์สเปรดสเปกตรัม ซึ่งแสดงผลได้ดังนี้

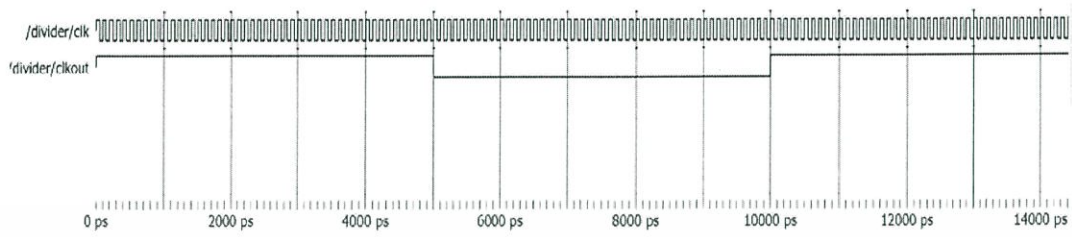
4.2.1.1 วงจรหารความถี่ (Frequency divider)



รูปที่ 4.15 วงจรหารความถี่ (Frequency divider)

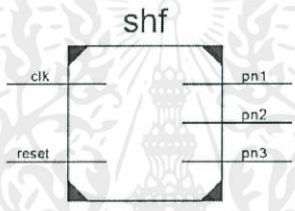
วงจรหารความถี่สัญญาณนาฬิกาที่ป้อนเข้าไปเพื่อลดความถี่ไปใช้ควบคุมจังหวะของวงจรต่างๆที่ความเร็วไม่เท่ากันซึ่งสามารถจำลองการทำงาน โดยสัญญาณนาฬิกา clk ที่ได้รับเข้ามาจะถูกหารความถี่ออกเป็นจำนวน 100 เท่า ตามผลการจำลองการทำงานเป็นสัญญาณ clkout ดังรูปที่ 4.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



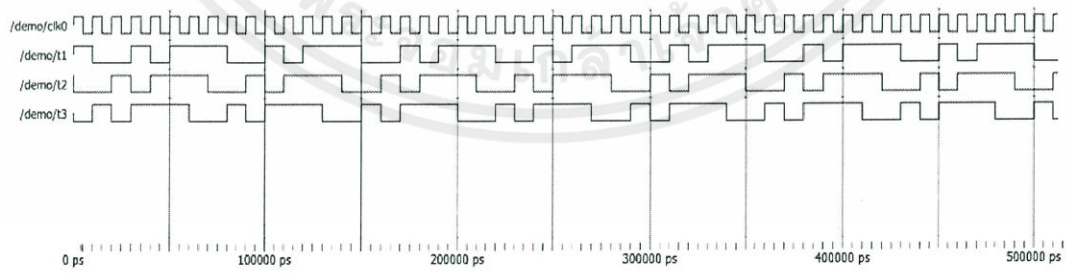
รูปที่ 4.16 ผลการจำลองการทำงานของวงจรหารความถี่

4.2.1.2 ชิฟต์รีจิสเตอร์ (Shift Register)



รูปที่ 4.17 ชิฟต์รีจิสเตอร์ (Shift Register)

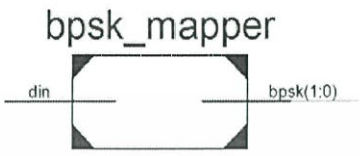
เป็นวงจรกำเนิดชุดรหัส PN (Pseudorandom noise) เพื่อที่จะนำไปสเปรดสัญญาณข้อมูลเพื่อให้สัญญาณข้อมูลมีลักษณะใกล้เคียงสัญญาณรบกวน โดยสัญญาณนาฬิกา clk จะให้จังหวะในการกำเนิดสัญญาณรหัสพีเอ็น ออกมา 3 ชุดสำหรับผู้ใช้จำนวน 3 ราย ซึ่งก็คือ สัญญาณ t1 , t2 และ t3 ดังรูปที่ 4.18



รูปที่ 4.18 ผลการจำลองการทำงานของชิฟต์รีจิสเตอร์

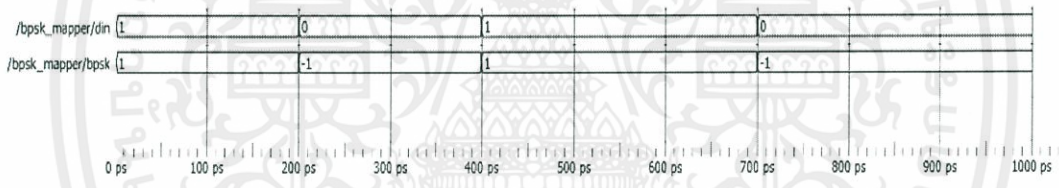
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.3 วงจรแม็ปค่า (BPSK Mapper)



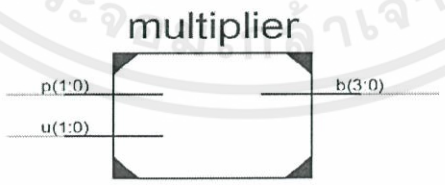
รูปที่ 4.19 วงจรแม็ปค่า (BPSK Mapper)

วงจรปรับระดับค่าสัญญานจากสัญญาณดิจิตอลที่มีค่า 1 กับ 0 ปรับเป็น 1 กับ -1 โดยสัญญาณ din หากมีค่า 1 จะได้สัญญาณ bpsk มีค่าเป็น 1 หากสัญญาณ din มีค่า 0 จะได้สัญญาณ bpsk มีค่าเป็น -1 ซึ่งโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงานได้ ดังรูปที่ 4.20



รูปที่ 4.20 ผลการจำลองการทำงานของวงจรแม็ปค่า

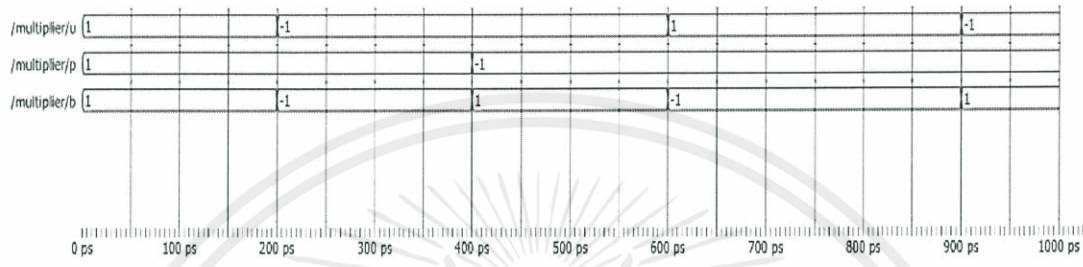
4.2.1.4 วงจรคูณ (Multiplier)



รูปที่ 4.21 วงจรคูณ (Multiplier)

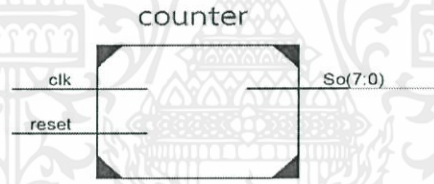
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นวงจรที่ออกแบบมาเพื่อนำสัญญาณมาคูณกันแล้วทำการสเปรดสัญญาณและมอดูเลต โดยสัญญาณ u และ p เป็นสัญญาณที่รับเข้ามาซึ่งจะนำมาคูณกันก็จะได้สัญญาณ b ซึ่งโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงานได้ดังรูปที่ 4.22



รูปที่ 4.22 ผลการจำลองการทำงานของวงจรคูณ

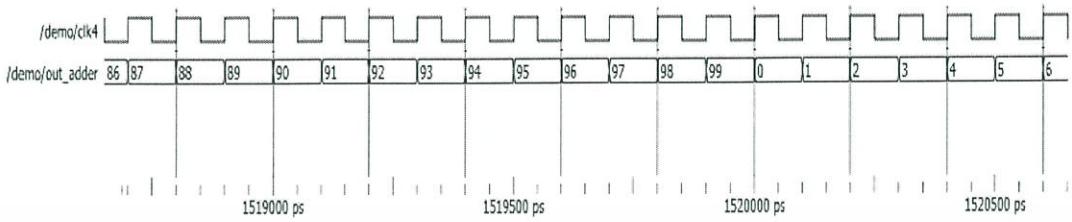
4.2.1.5 วงจรนับค่า (Counter)



รูปที่ 4.23 วงจรนับค่า (Counter)

วงจรนับค่านับเพื่อที่จะสร้างสัญญาณไซน์ออกมาแล้วนำมาเป็นสัญญาณพาหะในการมอดูเลต โดยสัญญาณนาฬิกาจะนำมาใช้ควบคุมจังหวะในการนับค่า 0 ถึง 99 เพื่อที่จะนำไปใช้เป็นค่าในการป้อนเข้าวงจรเก็บค่าสัญญาณพาหะ จากโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงานได้ดังรูปที่ 4.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

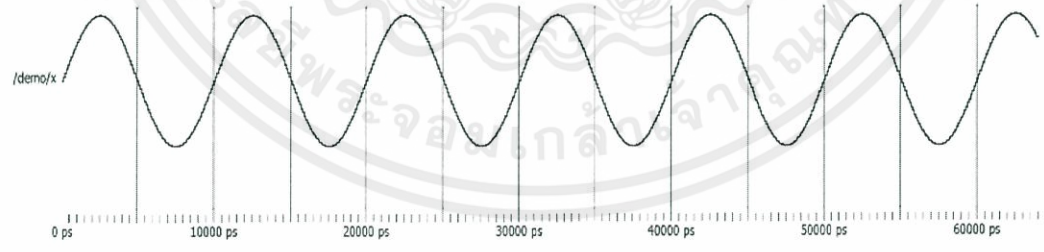


รูปที่ 4.24 ผลการจำลองการทำงานของวงจรนับค่า

4.2.1.6 วงจรกำเนิดสัญญาณพาหะ (Digital Oscillator)

รูปที่ 4.25 วงจรกำเนิดสัญญาณพาหะ (Digital Oscillator)

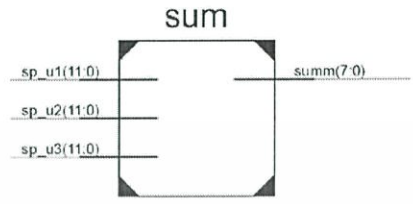
เป็นวงจรที่เก็บค่าที่ตำแหน่งต่างๆในสัญญาณไซน์โดยจะใช้วงจรนับค่าในการเลือกค่าในวงจรกำเนิดสัญญาณพาหะออกมาเพื่อจะนำไปใช้ในการมอดูเลต จากโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงานได้ดังรูปที่ 4.26



รูปที่ 4.26 ผลการจำลองการทำงานของวงจรกำเนิดสัญญาณพาหะ

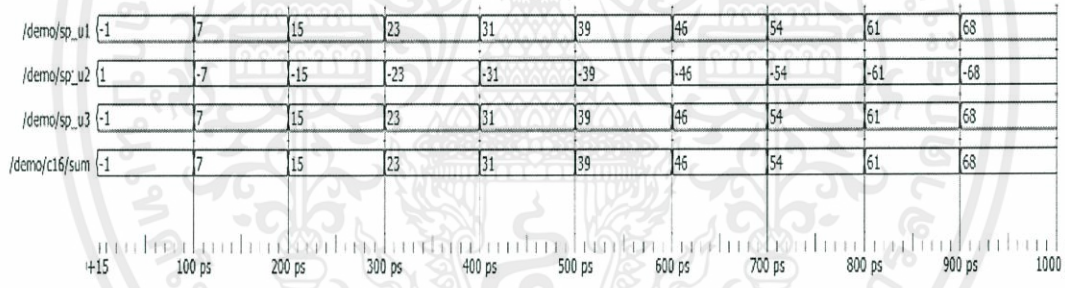
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.1.7 วงจรรวมสัญญาณ (Summing)



รูปที่ 4.27 วงจรรวมสัญญาณ (Summing)

เป็นวงจรรวมสัญญาณที่ใช้รวมสัญญาณของผู้ใช้ทุกรายเพื่อที่จะสามารถส่งรวมกันไปได้ในสัญญาณเดียวกันไปยังวงจรภาครับ โดยจากผลการจำลองการทำงาน จะเป็นการบวกกันระหว่างสัญญาณ sp_u1 , sp_u2 และ sp_u3 จะได้สัญญาณ sum ซึ่งจากโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงานได้ดังรูปที่ 4.28



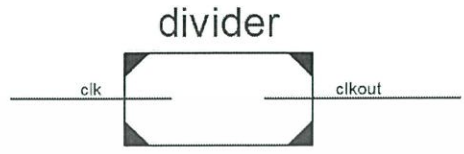
รูปที่ 4.28 ผลการจำลองการทำงานของวงจรรวมสัญญาณ

4.2.2 ภาครับ

จากบทที่ 3 ได้กล่าวถึงการออกแบบการทดลอง ส่วนในบทนี้จะกล่าวถึงผลการทดลองของวงจรในส่วนต่างๆของภาครับที่ใช้ในระบบการจำลองการสื่อสารแบบไดเรกต์ซีเควนซ์สเปรดสเปกตรัม ซึ่งแสดงผลได้ดังนี้

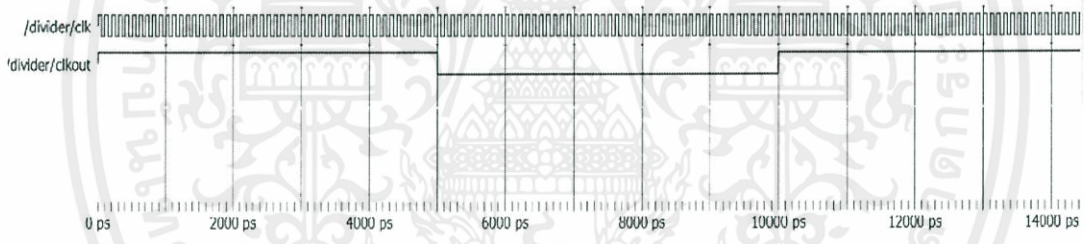
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.1 วงจรหารความถี่ (Frequency divider)



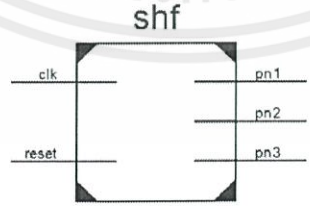
รูปที่ 4.29 วงจรหารความถี่ (Frequency divider)

วงจรหารความถี่สัญญาณนาฬิกาที่ป้อนเข้าไปเพื่อลดความถี่ไปใช้ควบคุมจังหวะของวงจรต่างๆที่ความเร็วไม่เท่ากันซึ่ง โดยสัญญาณนาฬิกา clk ที่ได้รับเข้ามาจะถูกหารความถี่ออกมาก 100 เท่า ตามผลการจำลองการทำงานเป็นสัญญาณ clkout สามารถจำลองการทำงานได้ดังรูปที่ 4.30



รูปที่ 4.30 ผลการจำลองการทำงานของวงจรหารความถี่

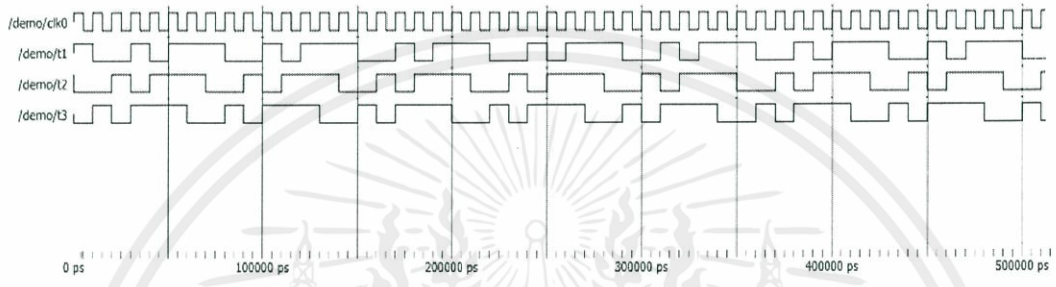
4.2.2.2 ชิฟต์รีจิสเตอร์ (Shift Register)



รูปที่ 4.31 ชิฟต์รีจิสเตอร์ (Shift Register)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นวงจรกำเนิดชุดรหัส PN (Pseudorandom noise) เพื่อที่จะนำไปดีสเปรดสัญญาณข้อมูลเพื่อให้สัญญาณข้อมูลมีลักษณะใกล้เคียงสัญญาณรบกวน โดยสัญญาณนาฬิกา clk จะให้จังหวะในการกำเนิดสัญญาณรหัส PN ออกมา 3 ชุดสำหรับผู้ใช้จำนวน 3 ราย ซึ่งก็คือ สัญญาณ t1 , t2 และ t3 ซึ่งสามารถจำลองการทำงานได้ดังรูปที่ 4.32



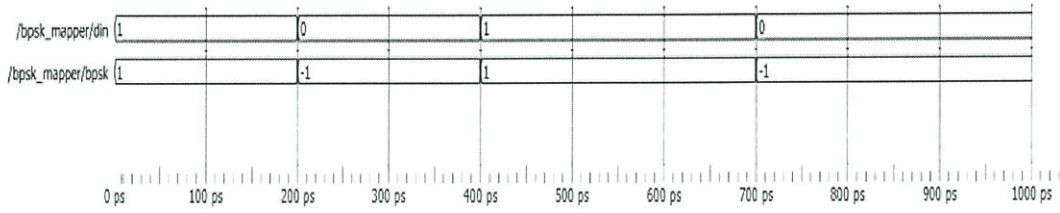
รูปที่ 4.32 ผลการจำลองการทำงานของชิฟต์รีจิสเตอร์

4.2.2.3 วงจรแม็ปค่า (BPSK Mapper)

รูปที่ 4.33 วงจรแม็ปค่า (BPSK Mapper)

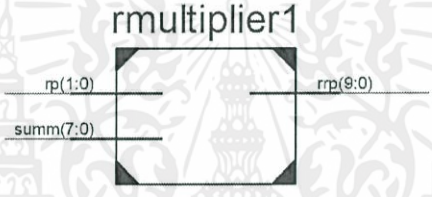
วงจรปรับระดับค่าสัญญานจากสัญญาณดิจิตอลที่มีค่า 1 กับ 0 ปรับเป็น 1 กับ -1 โดยสัญญาณ din หากมีค่า 1 จะได้สัญญาณ bpsk มีค่าเป็น 1 หากสัญญาณ din มีค่า 0 จะได้สัญญาณ bpsk มีค่าเป็น -1 ซึ่งโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงานได้ดังรูปที่ 4.34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



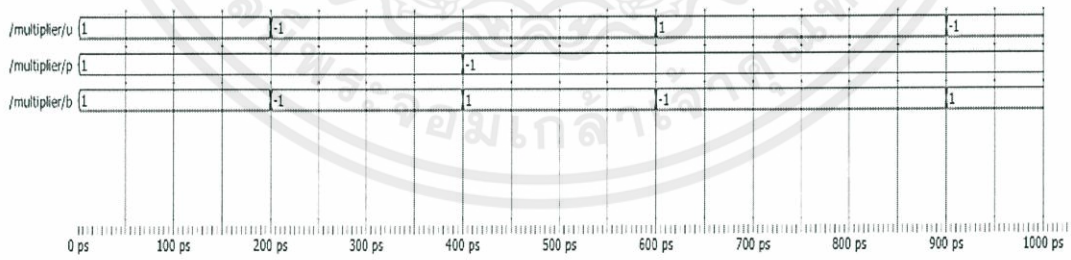
รูปที่ 4.34 ผลการจำลองการทำงานของวงจรแม็ปค่า

4.2.2.4 วงจรคูณ (Multiplier)



รูปที่ 4.35 วงจรคูณ (Multiplier)

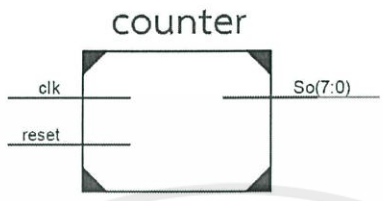
เป็นวงจรที่ออกแบบมาเพื่อนำสัญญาณมาคูณกันแล้วทำการดีสเปรดสัญญาณและดีมอดูเลต โดยสัญญาณ u และ p เป็นสัญญาณที่รับเข้ามาซึ่งจะนำมาคูณกันก็จะได้สัญญาณ b ซึ่งโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงานได้ดังรูปที่ 4.36



รูปที่ 4.36 ผลการจำลองการทำงานของวงจรวงจรคูณ

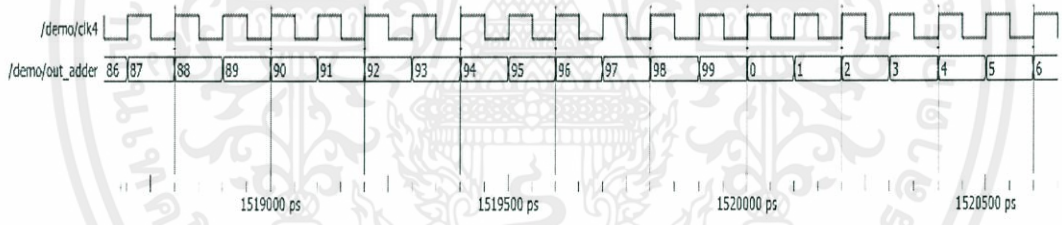
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.5 วงจรนับค่า (Counter)



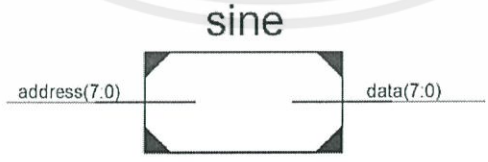
รูปที่ 4.37 วงจรนับค่า (Counter)

วงจรนับค่านับเพื่อที่จะสร้างสัญญาณไซน์ออกมาแล้วนำมาเป็นสัญญาณพาหะในการดีมอดูเลต โดยสัญญาณนาฬิกาจะนำมาใช้ควบคุมจังหวะในการนับค่า 0 ถึง 99 เพื่อที่จะนำไปใช้เป็นค่าในการป้อนเข้าวงจรเก็บค่าสัญญาณพาหะ จากโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงานได้ดังรูปที่ 4.38



รูปที่ 4.38 ผลการจำลองการทำงานของวงจรนับค่า

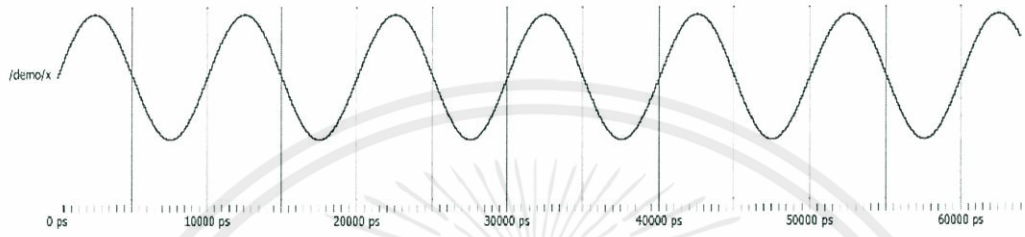
4.2.2.6 วงจรกำเนิดสัญญาณพาหะ (Digital Oscillator)



รูปที่ 4.39 วงจรกำเนิดสัญญาณพาหะ (Digital Oscillator)

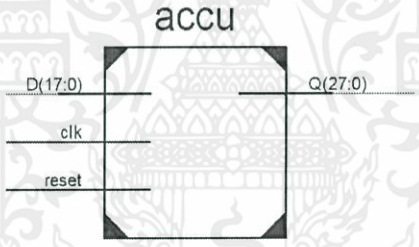
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นวงจรที่เก็บค่าที่ตำแหน่งต่างๆในสัญญาณไซน์โดยจะใช้วงจรนับค่าในการเลือกค่าในวงจรกำเนิดสัญญาณพาหะออกมา จากโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงานได้ดังรูปที่ 4.40



รูปที่ 4.40 ผลการจำลองการทำงานของวงจรกำเนิดสัญญาณพาหะ

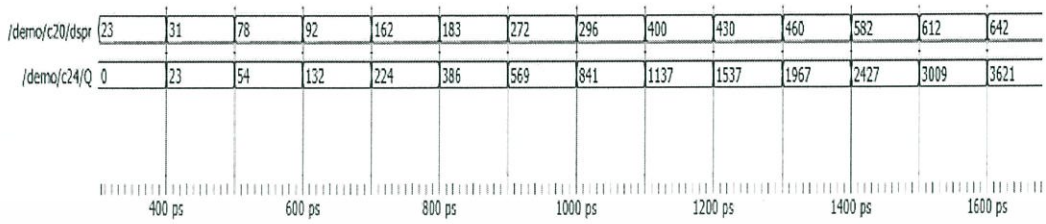
4.2.2.7 วงจรบวกสะสมค่า (Accumulator)



รูปที่ 4.41 วงจรบวกสะสมค่า (Accumulator)

เป็นวงจรที่ใช้ในการบวกสะสมค่าขึ้นไปเรื่อยๆ จนกว่าจะได้รับสัญญาณรีเซ็ต วงจรก็จะเริ่มนับค่าใหม่โดยเริ่มตั้งแต่ 0 โดยสัญญาณ dspr คือ สัญญาณที่ได้รับเข้ามาเพื่อที่จะทำการบวกสะสมค่าไปเรื่อยๆ จนกว่าจะมีการรีเซ็ต ซึ่งก็คือ ค่าสัญญาณ Q จากโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงานได้ดังรูปที่ 4.42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

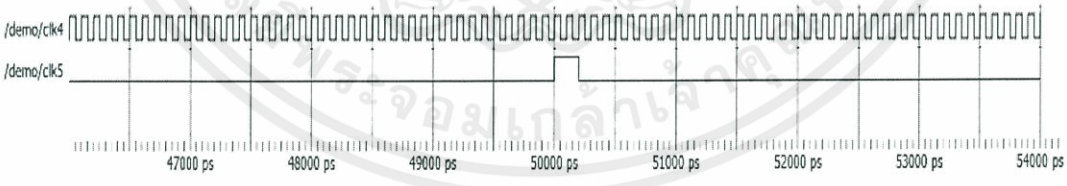


รูปที่ 4.42 ผลการจำลองการทำงานของวงจรบวกสะสมค่า

4.2.2.8 วงจรกำหนดขอบเขต (Counter Accumulator)

รูปที่ 4.43 วงจรกำหนดขอบเขต (Counter Accumulator)

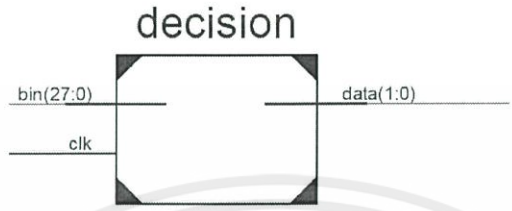
วงจรนับค่านับเพื่อที่จะกำหนดช่วงในการบวกสะสมค่าสัญญาณภายหลังการดีมอดูเลต ซึ่งจะอยู่ในช่วงหนึ่งคาบบิต โดยที่สัญญาณ clk5 คือ สัญญาณที่นำไปรีเซ็ตวงจรสะสมค่า จากโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงานได้ดังรูปที่ 4.44



รูปที่ 4.44 ผลการจำลองการทำงานของวงจรกำหนดขอบเขต

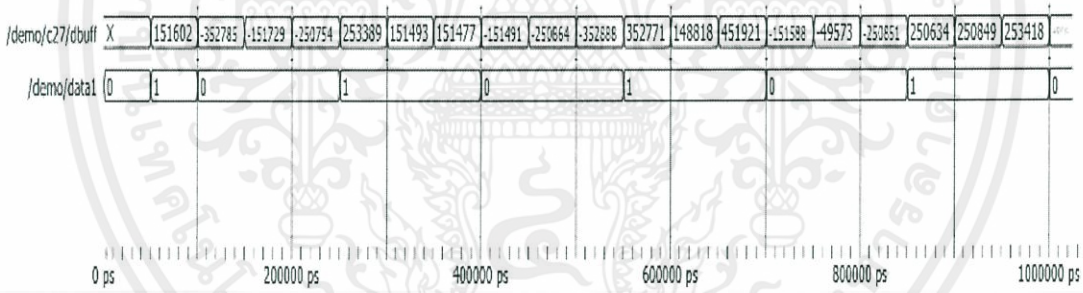
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.2.9 วงจรตัดสินใจบิต (Decision)



รูปที่ 4.45 วงจรตัดสินใจบิต (Decision)

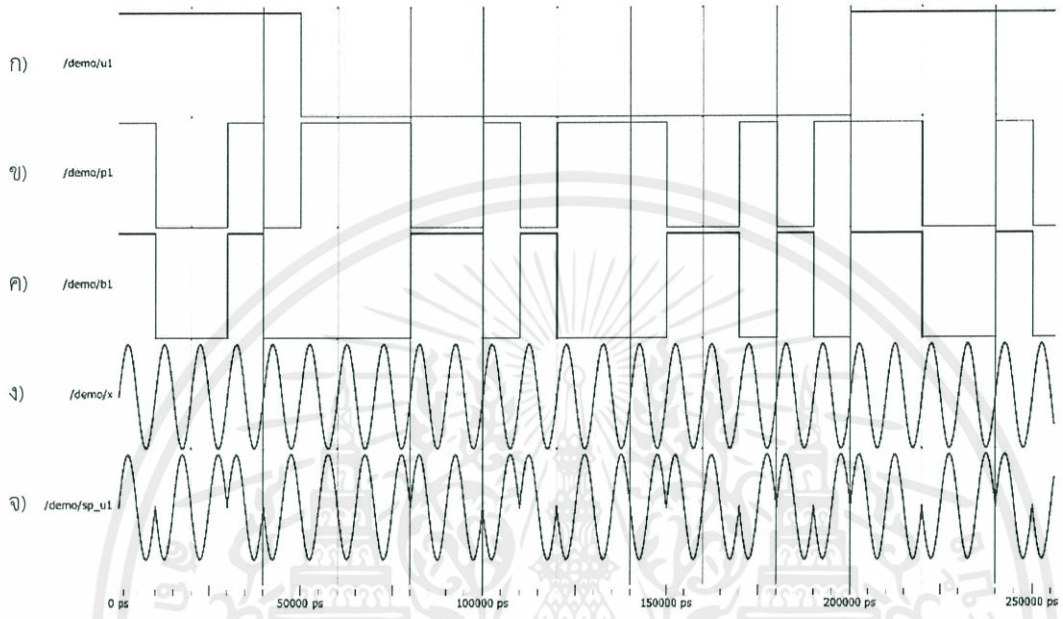
เป็นวงจรที่ใช้ในการตัดสินใจค่าที่ได้รับเข้ามาถ้ามีค่ามากกว่า 0 จะตัดสินใจเป็นบิต 1 นอกจากนั้นจะตัดสินใจให้เป็นบิต 0 โดยตัดสินใจค่าสัญญาณ dbuff ที่ได้รับเข้ามา ถ้ามีค่ามากกว่า 0 จะตัดสินใจสัญญาณ data เป็นบิต 1 นอกจากนั้นจะตัดสินใจสัญญาณ data ให้เป็นบิต จากโปรแกรมที่เขียนขึ้นสามารถจำลองการทำงานได้ดังรูปที่ 4.46



รูปที่ 4.46 ผลการจำลองการทำงานของวงจรตัดสินใจบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

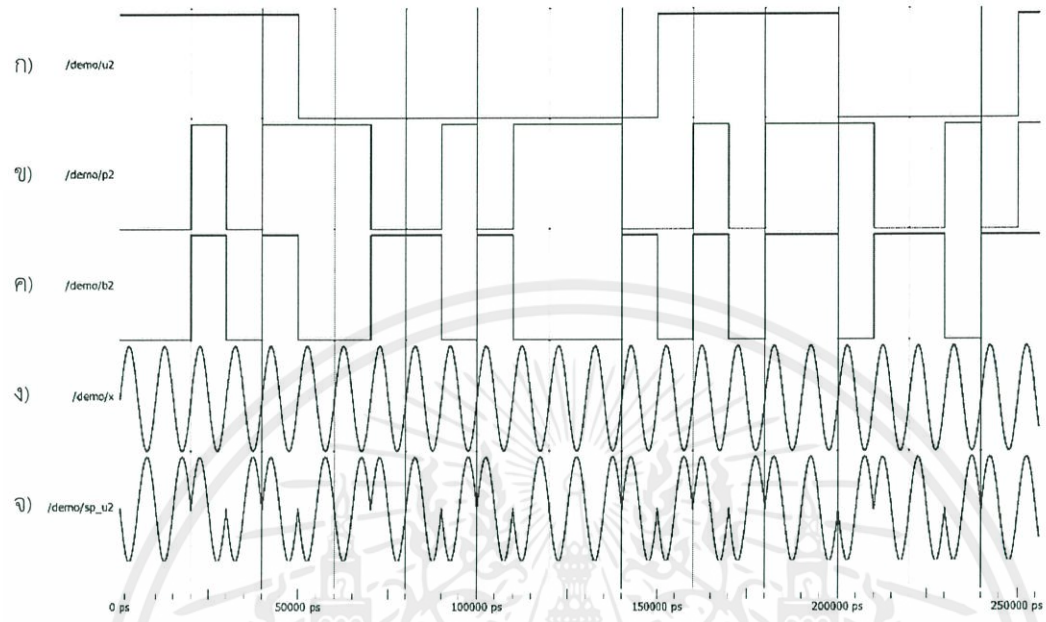
4.2.3 ภาพรวมการทำงานของทั้งภาครับและภาคส่ง



รูปที่ 4.47 ผลการจำลองการทำงานของภาคส่งของผู้ใช้รายที่ 1

จากรูปที่ 4.47 รูป ก แสดงถึงข้อมูลของผู้ใช้รายที่ 1 รูป ข แสดงถึงรหัสพีเอ็นของผู้ใช้รายที่ 1 รูป ค แสดงถึงข้อมูลที่ถูกสเปรดจากรหัสพีเอ็นของผู้ใช้รายที่ 1 ซึ่งจะเห็นได้ว่ารหัสพีเอ็นของผู้ใช้รายที่ 1 เมื่อทำการคูณกับสัญญาณข้อมูลแล้ว จะทำให้ข้อมูลในการส่งอยู่ในรูปแบบของการสเปรด ซึ่งหลังจากนั้นจะนำข้อมูลที่ถูกสเปรดไปคูณกับคลื่นพาหะดังรูป ง จะกล่าวได้ว่าเป็นการมอดูเลตในรูปแบบของ Binary Phase Shift Keying (BPSK) ของผู้ใช้รายที่ 1 แล้วนำไปรวมกับสัญญาณของผู้ใช้รายอื่นก่อนถูกส่งออกไปยังภาครับดังรูป จ

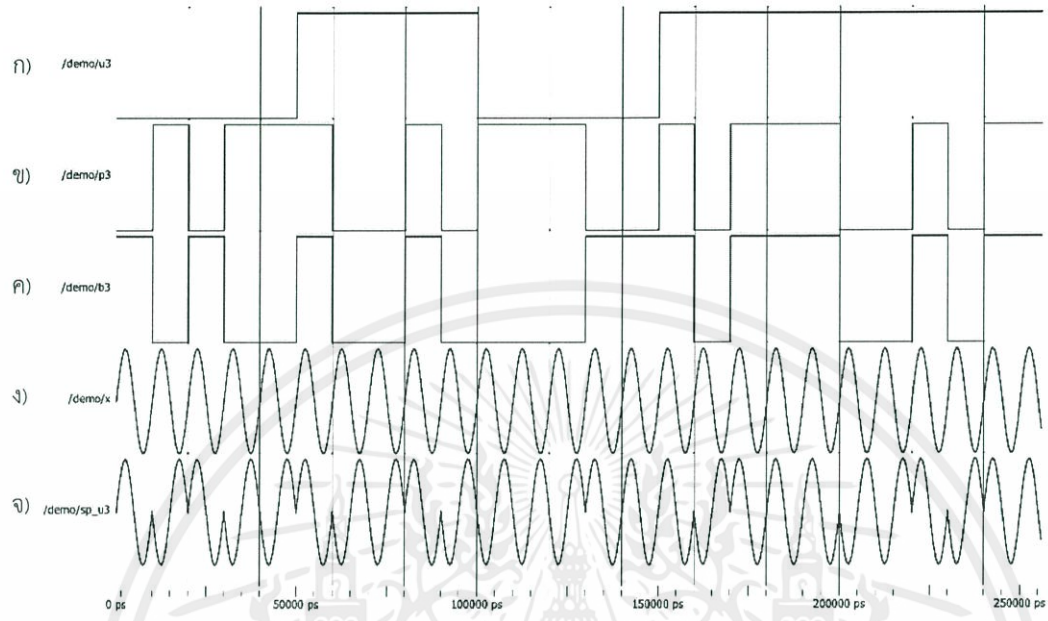
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.48 ผลการจำลองการทำงานของภาคส่งของผู้ใช้รายที่ 2

จากรูปที่ 4.48 รูป ก แสดงถึงข้อมูลของผู้ใช้รายที่ 2 รูป ข แสดงถึงรหัสพีเอ็นของผู้ใช้รายที่ 2 รูป ค แสดงถึงข้อมูลที่ถูกสเปรดจากรหัสพีเอ็นของผู้ใช้รายที่ 2 ซึ่งจะเห็นได้ว่ารหัสพีเอ็นของผู้ใช้รายที่ 2 เมื่อทำการคูณกับสัญญาณข้อมูลแล้ว จะทำให้ข้อมูลในการส่งอยู่ในรูปแบบของการสเปรด ซึ่งหลังจากนั้นจะนำข้อมูลที่ถูกสเปรดไปคูณกับคลื่นพาหะดังรูป ง จะกล่าวได้ว่าเป็นการมอดูเลตในรูปแบบของ Binary Phase Shift Keying (BPSK) ของผู้ใช้รายที่ 2 แล้วนำไปรวมกับสัญญาณของผู้ใช้รายอื่นก่อนถูกส่งออกไปยังภาครับดังรูป จ

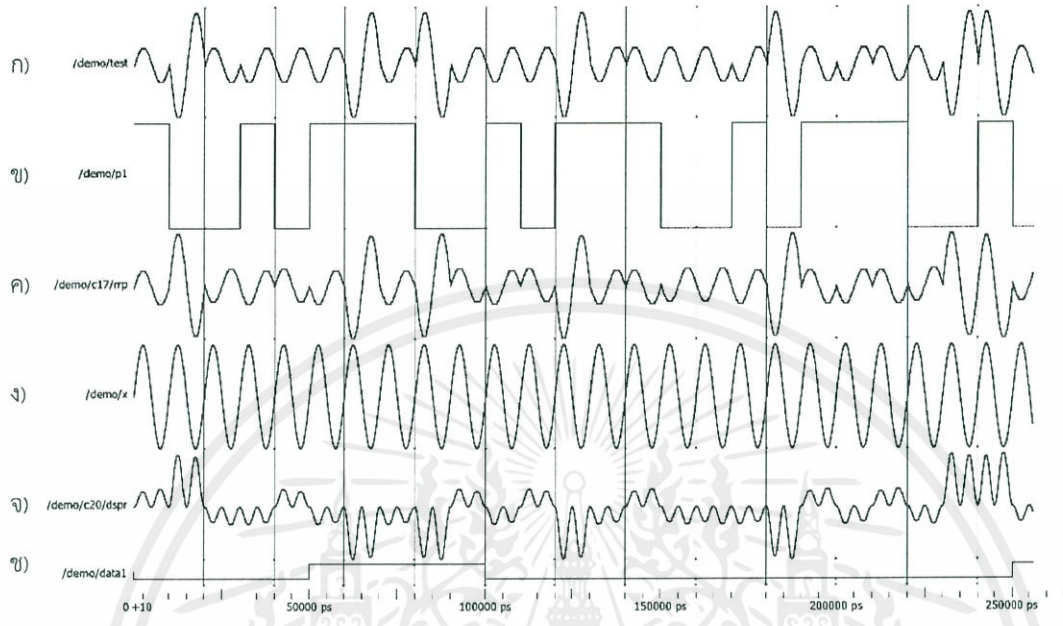
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.49 ผลการจำลองการทำงานของภาคส่งของผู้ใช้รายที่ 3

จากรูปที่ 4.49 รูป ก แสดงถึงข้อมูลของผู้ใช้รายที่ 3 รูป ข แสดงถึงรหัสพีเอ็นของผู้ใช้รายที่ 3 รูป ค แสดงถึงข้อมูลที่ถูกสเปรดจากรหัสพีเอ็นของผู้ใช้รายที่ 3 ซึ่งจะเห็นได้ว่ารหัสพีเอ็นของผู้ใช้รายที่ 3 เมื่อทำการคูณกับสัญญาณข้อมูลแล้ว จะทำให้ข้อมูลในการส่งอยู่ในรูปแบบของการสเปรด ซึ่งหลังจากนั้นจะนำข้อมูลที่ถูกสเปรดไปคูณกับคลื่นพาหะดังรูป ง จะกล่าวได้ว่าเป็นการมอดูเลตในรูปแบบของ Binary Phase Shift Keying (BPSK) ของผู้ใช้รายที่ 3 แล้วนำไปรวมกับสัญญาณของผู้ใช้รายอื่นก่อนถูกส่งออกไปยังภาครับดังรูป จ

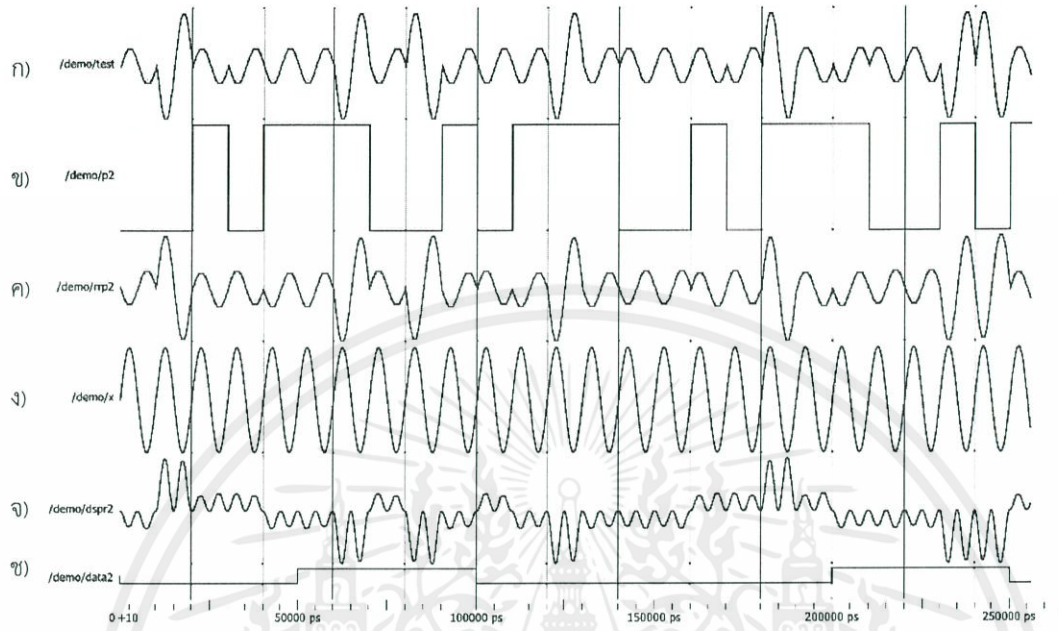
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.50 ผลการจำลองการทำงานของภาครับของผู้ใช้รายที่ 1

จากรูปที่ 4.50 รูป ก แสดงถึงสัญญาณข้อมูลที่ได้รับจากภาคส่งของผู้ใช้ ส่วนรูป ข แสดงถึงรหัสพีเอ็นของผู้ใช้รายที่ 1 รูป ค แสดงถึงสัญญาณข้อมูลที่ได้รับคูณกับรหัสพีเอ็นของผู้ใช้รายที่ 1 รูป ง แสดงถึงคลื่นพาหะ และ รูป จ แสดงถึงการดีมอดูเลตสัญญาณข้อมูลของผู้ใช้รายที่ 1

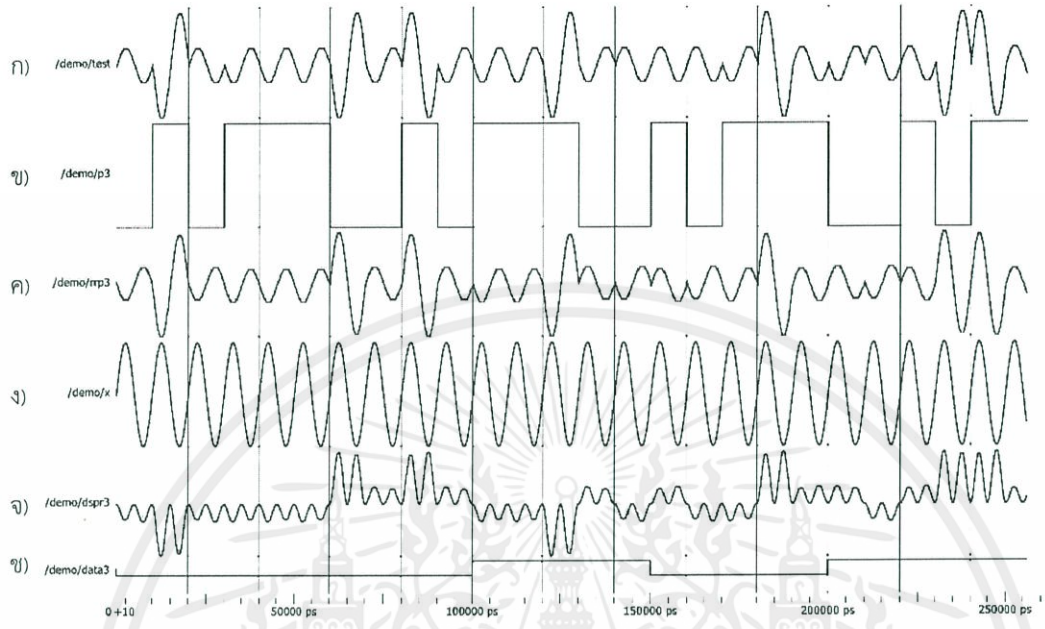
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.51 ผลการจำลองการทำงานของภาครับของผู้ใช้รายที่ 2

จากรูปที่ 4.51 รูป ก แสดงถึงสัญญาณข้อมูลที่ได้รับจากภาคส่งของผู้ใช้ ส่วนรูป ข แสดงถึงรหัสพีเอ็นของผู้ใช้รายที่ 2 รูป ค แสดงถึงสัญญาณข้อมูลที่ได้รับคูณกับรหัสพีเอ็นของผู้ใช้รายที่ 2 รูป ง แสดงถึงคลื่นพาหะ และ รูป จ แสดงถึงการดีมอดูเลชั่นสัญญาณข้อมูลของผู้ใช้รายที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



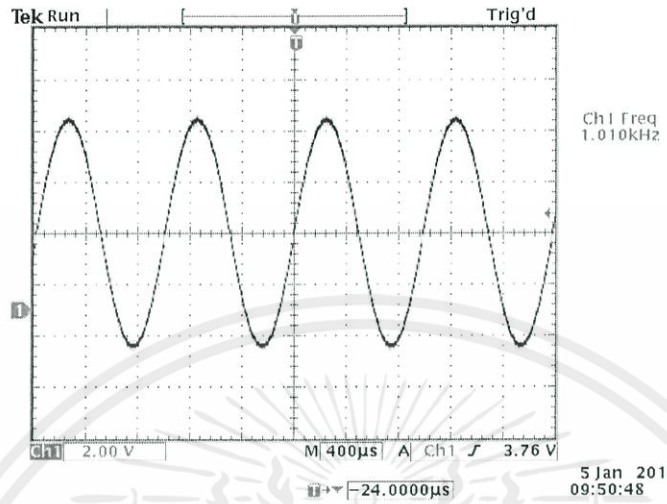
รูปที่ 4.52 ผลการจำลองการทำงานของภาครับของผู้ใช้รายที่ 3

จากรูปที่ 4.52 รูป ก แสดงถึงสัญญาณข้อมูลที่ได้รับจากภาคส่งของผู้ใช้ ส่วนรูป ข แสดงถึงรหัสพีเอ็นของผู้ใช้รายที่ 3 รูป ค แสดงถึงสัญญาณข้อมูลที่ได้รับคูณกับรหัสพีเอ็นของผู้ใช้รายที่ 3 รูป ง แสดงถึงคลื่นพาหะ และ รูป จ แสดงถึงการดีมอดูเลชั่นสัญญาณข้อมูลของผู้ใช้รายที่ 3

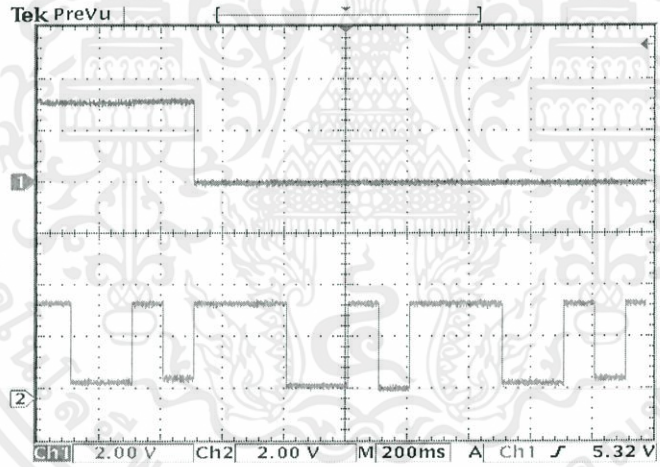
4.3 ผลการทดลองการทำงานบนอุปกรณ์ FPGA

จากรูปที่ 4.53 เป็นการวัดสัญญาณจากการสร้างสัญญาณไซน์เวฟของวงจรดิจิตอลโดยใช้บอร์ด FPGA ผ่าน DAC (Digital analog converter) ซึ่งมี 100 sample time โดยในการทดลองวัดสัญญาณใช้ความถี่ของสัญญาณนาฬิกา 1 KHz จึงได้สัญญาณไซน์เวฟที่มีความถี่ 10Hz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



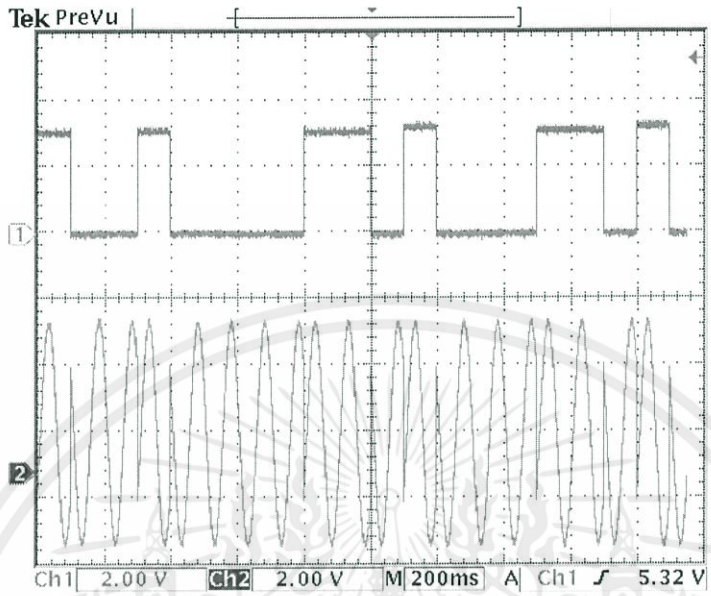
รูปที่ 4.53 สัญญาณไซน์เวฟที่ได้จากวงจรดิจิตอลออสซิลเลเตอร์



รูปที่ 4.54 Ch.1 สัญญาณข้อมูลของผู้ใช้รายที่ 1
Ch.2 สัญญาณ PN code 1

จากรูปที่ 4.54 Ch.1 แสดงถึงข้อมูลของผู้ใช้รายที่ 1 ส่วน Ch.2 จะแสดงถึงรหัสพีเอ็น ของผู้ใช้รายที่ 1 ซึ่งสร้างมาจากวงจร Linear Feedbaack Shift Register 3 stage

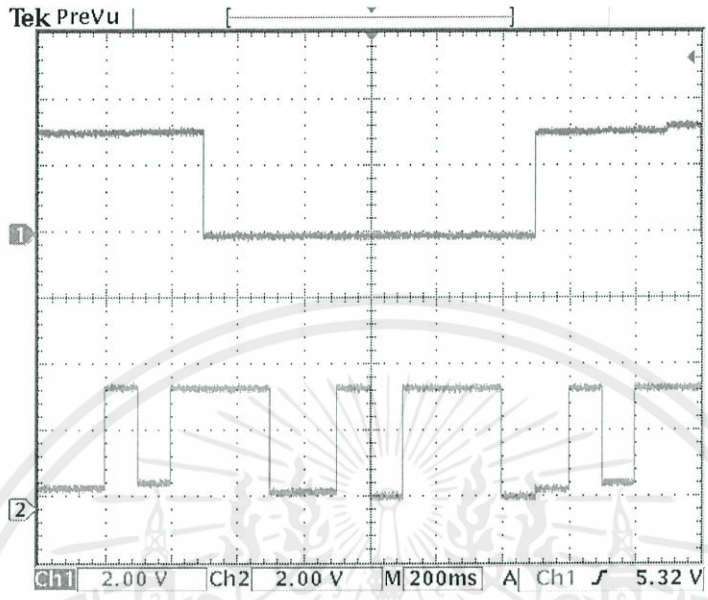
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.55 Ch.1 สัญญาณข้อมูลที่ถูกสเปรดจากรหัสพีเอ็นของผู้ใช้รายที่ 1
Ch.2 สัญญาณ BPSK ของผู้ใช้รายที่ 1

รูปที่ 4.55 Ch.1 แสดงถึงข้อมูลที่ถูกสเปรดจากรหัสพีเอ็นของผู้ใช้รายที่ 1 ซึ่งจะเห็นได้ว่ารหัสพีเอ็น ของผู้ใช้รายที่ 1 เมื่อทำการคูณกับสัญญาณข้อมูลแล้ว จะทำให้ข้อมูลในการส่งอยู่ในรูปแบบของการสเปรด ซึ่งหลังจากนั้นจะนำข้อมูลที่ถูกสเปรดไปคูณกับคลื่นพาหะ จะกล่าวได้ว่าเป็นการมอดูเลตในรูปแบบของ Binary Phase Shift Keying (BPSK) ของผู้ใช้รายที่ 1

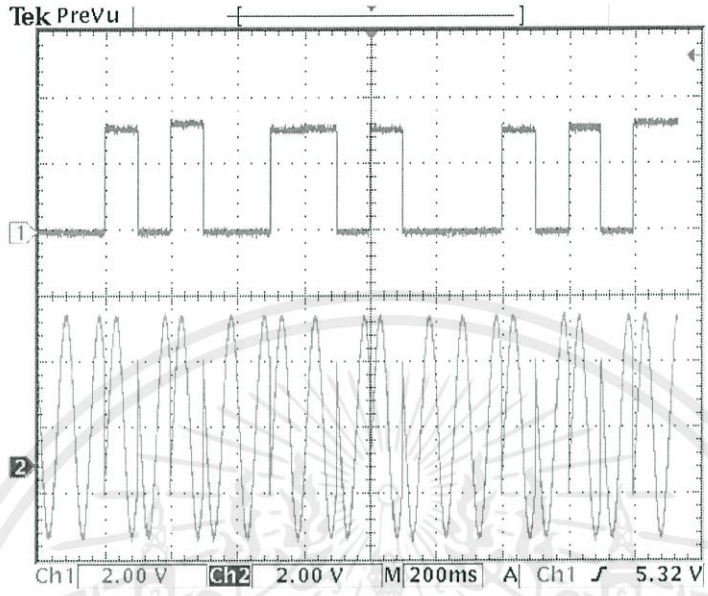
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.56 Ch.1 สัญญาณข้อมูลของผู้ใช้รายที่ 2
Ch.2 สัญญาณ PN code 2

จากรูปที่ 4.56 Ch.1 แสดงถึงข้อมูลของผู้ใช้รายที่ 2 ส่วน Ch.2 จะแสดงถึงรหัสพีเอ็น ของผู้ใช้รายที่ 2 ซึ่งสร้างมาจากวงจร Linear Feedbaack Shift Register 3 stages

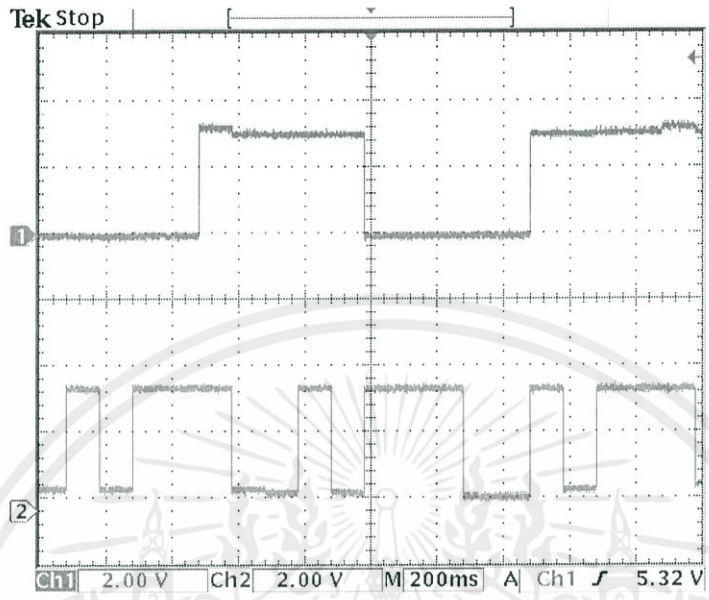
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.57 Ch.1 สัญญาณข้อมูลที่ถูกสเปรดจากรหัสพีเอ็นของผู้ใช้รายที่ 2
Ch.2 สัญญาณ BPSK ของผู้ใช้รายที่ 2

รูปที่ 4.57 Ch.1 แสดงถึงข้อมูลที่ถูกสเปรดจากรหัสพีเอ็นของผู้ใช้รายที่ 2 ซึ่งจะเห็นได้ว่ารหัสพีเอ็น ของผู้ใช้รายที่ 2 เมื่อทำการคูณกับสัญญาณข้อมูลแล้ว จะทำให้ข้อมูลในการส่งอยู่ในรูปแบบของการสเปรด ซึ่งหลังจากนั้นจะนำข้อมูลที่ถูกสเปรดไปคูณกับคลื่นพาหะ จะกล่าวได้ว่าเป็นการมอดูเลตในรูปแบบของ Binary Phase Shift Keying (BPSK) ของผู้ใช้รายที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

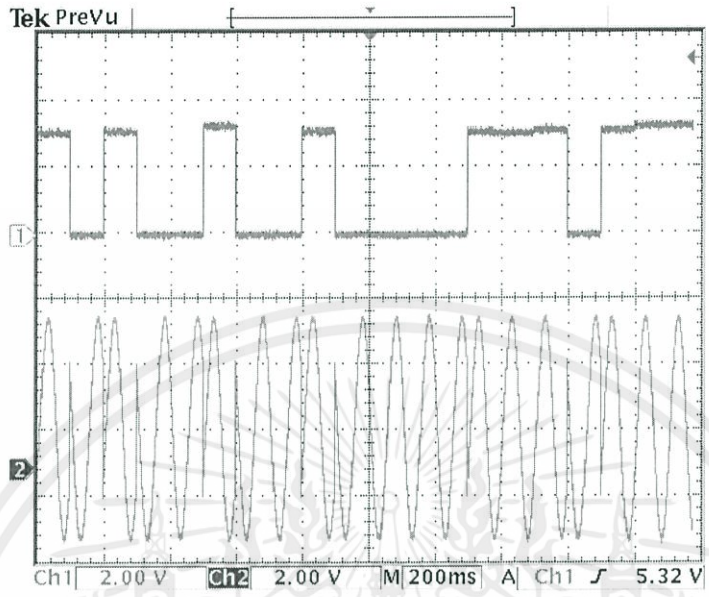


รูปที่ 4.58 Ch.1 สัญญาณข้อมูลของผู้ใช้รายที่ 3

Ch.2 สัญญาณ PN code 3

จากรูปที่ 4.58 Ch.1 แสดงถึงข้อมูลของผู้ใช้รายที่ 3 ส่วน Ch.2 จะแสดงถึงรหัสพีเอ็น ของผู้ใช้รายที่ 3 ซึ่งสร้างมาจากวงจร Linear Feedback Shift Register 3 stages

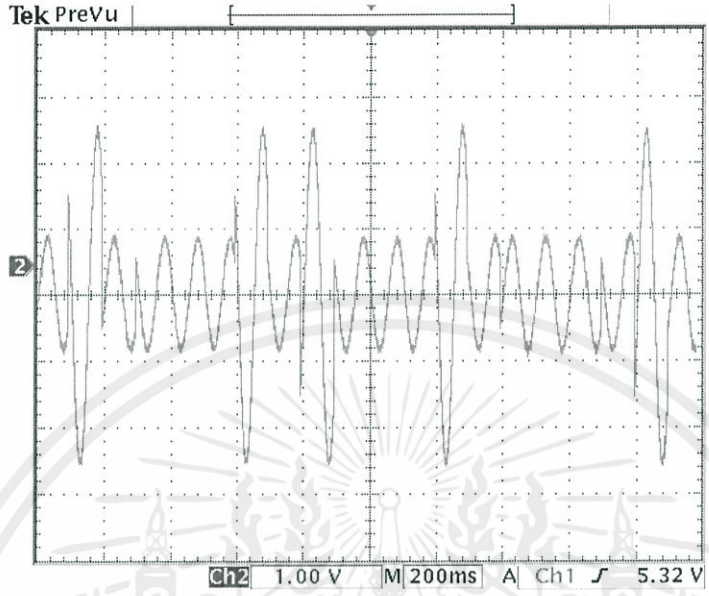
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.59 Ch.1 สัญญาณข้อมูลที่ถูกสเปรดจากรหัสพีเอ็นของผู้ใช้รายที่ 3
Ch.2 สัญญาณ BPSK ของผู้ใช้รายที่ 3

รูปที่ 4.59 Ch.1 แสดงถึงข้อมูลที่ถูกสเปรดจากรหัสพีเอ็นของผู้ใช้รายที่ 3 ซึ่งจะเห็นได้ว่ารหัสพีเอ็น ของผู้ใช้รายที่ 3 เมื่อทำการคูณกับสัญญาณข้อมูลแล้ว จะทำให้ข้อมูลในการส่งอยู่ในรูปแบบของการสเปรด ซึ่งหลังจากนั้นจะนำข้อมูลที่ถูกสเปรดไปคูณกับคลื่นพาหะ จะกล่าวได้ว่าเป็นการมอดูเลตในรูปแบบของ Binary Phase Shift Keying (BPSK) ของผู้ใช้รายที่ 3

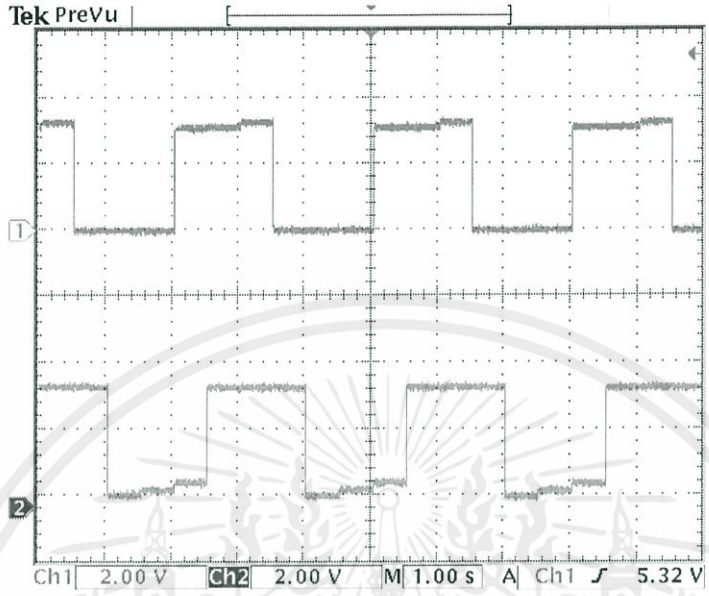
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.60 สัญญาณรวมจากผู้ใช้ทั้ง 3 ราย

จากรูปที่ 4.60 แสดงถึงสัญญาณ BPSK ของผู้ใช้ทั้ง 3 ราย ที่ได้มาจากวงจรรวมสัญญาณ

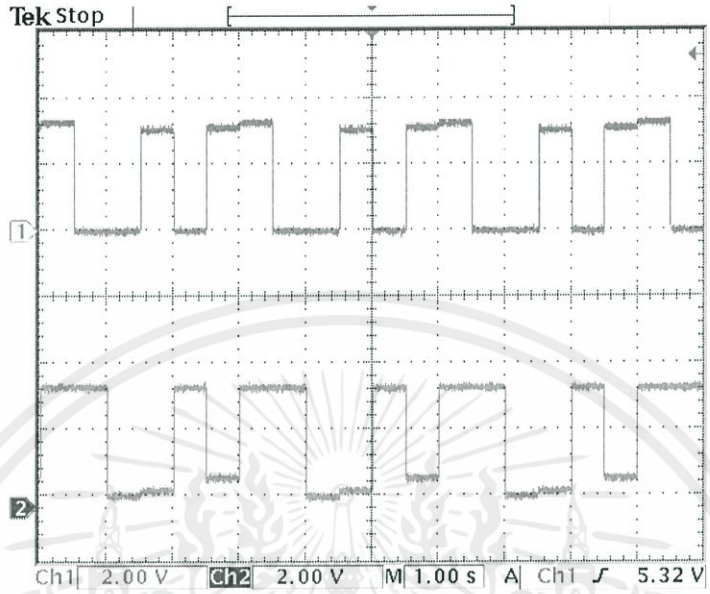
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.61 สัญญาณเปรียบเทียบระหว่างอินพุตกับเอาต์พุตของผู้ใช้รายที่ 1

จากรูปที่ 4.61 เป็นการวัดสัญญาณดิจิตอลจากบอร์ด FPGA ซึ่งเป็นสัญญาณอินพุตของผู้ใช้รายที่ 1 เปรียบเทียบกับสัญญาณเอาต์พุตของผู้ใช้รายที่ 1 ซึ่งจะเห็นว่าได้สัญญาณข้อมูลตามเดิม แต่มีการหน่วงเวลาเพิ่มขึ้น

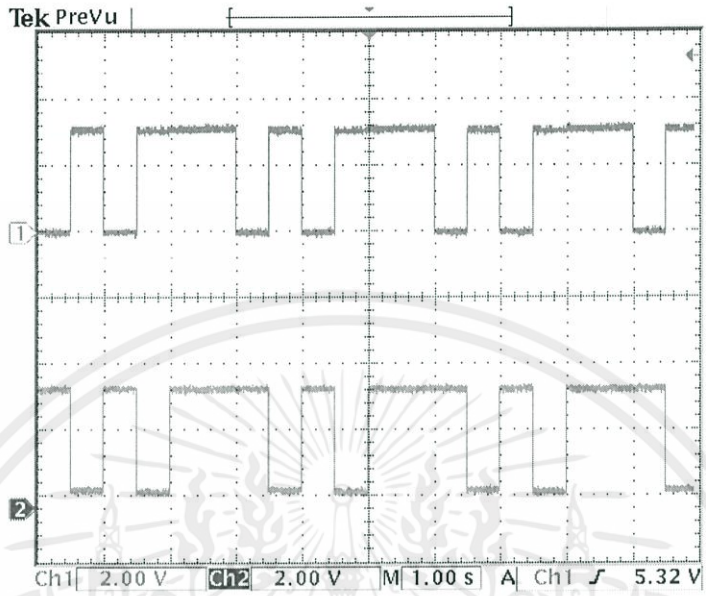
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.62 สัญญาณเปรียบเทียบระหว่างอินพุตกับเอาต์พุตของผู้ใช้รายที่ 2

จากรูปที่ 4.62 เป็นการวัดสัญญาณดิจิตอลจากบอร์ด FPGA ซึ่งเป็นสัญญาณอินพุตของผู้ใช้รายที่ 2 เปรียบเทียบกับสัญญาณเอาต์พุตของผู้ใช้รายที่ 2 ซึ่งจะเห็นว่าได้สัญญาณข้อมูลตามเดิม แต่มีการหน่วงเวลาเพิ่มขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

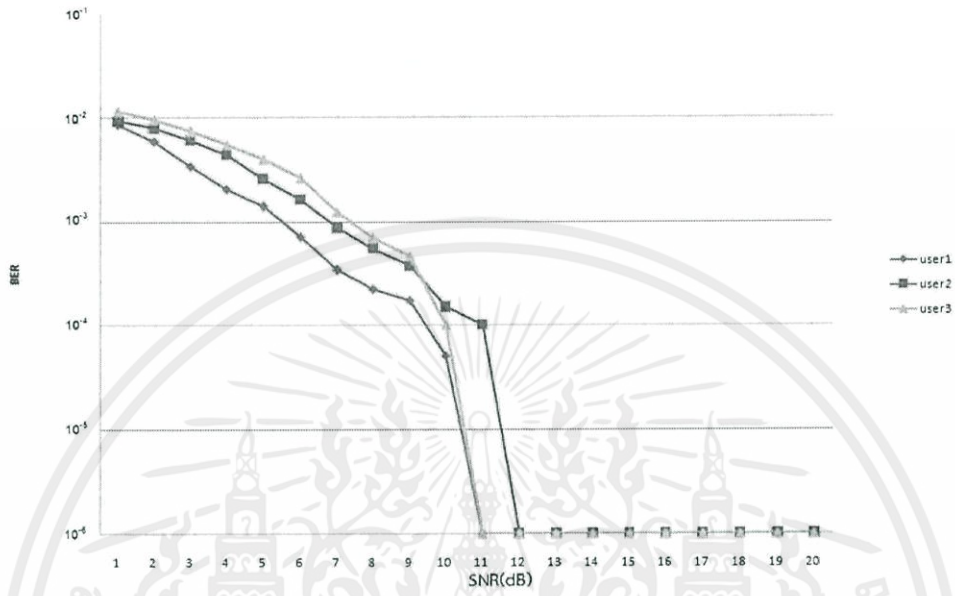


รูปที่ 4.63 สัญญาณเปรียบเทียบระหว่างอินพุตกับเอาต์พุตของผู้ใช้รายที่ 3

จากรูปที่ 4.63 เป็นการวัดสัญญาณดิจิตอลจากบอร์ด FPGA ซึ่งเป็นสัญญาณอินพุตของผู้ใช้รายที่ 3 เปรียบเทียบกับสัญญาณเอาต์พุตของผู้ใช้รายที่ 3 ซึ่งจะเห็นว่าได้สัญญาณข้อมูลตามเดิม แต่มีการหน่วงเวลาเพิ่มขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

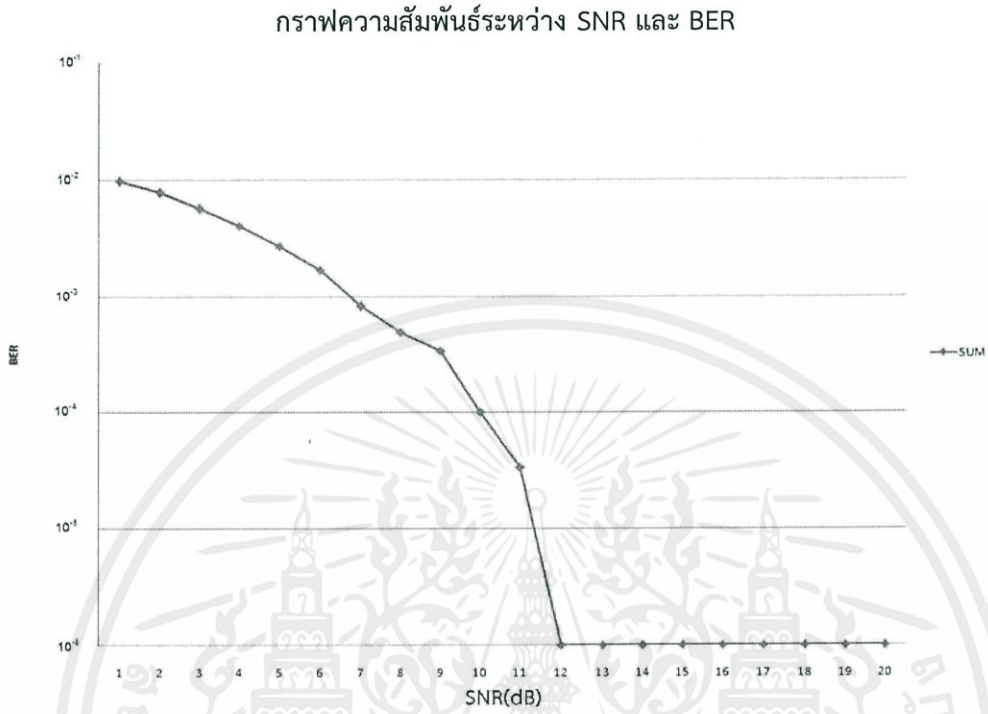
กราฟความสัมพันธ์ระหว่าง SNR และ BER



รูปที่ 4.64 กราฟความสัมพันธ์ระหว่าง BER กับ SNR ของผู้ใช้แต่ละราย

จากรูปที่ 4.64 เป็นการแสดงการเปรียบเทียบค่าระหว่าง Bit Error Rate กับ Signal to Noise Ratio ของผู้ใช้แต่ละรายโดยที่ค่า SNR ที่อยู่ในแกน x ที่เพิ่มขึ้น จะส่งผลให้ค่า BER ในแกน y มีค่าลดลง

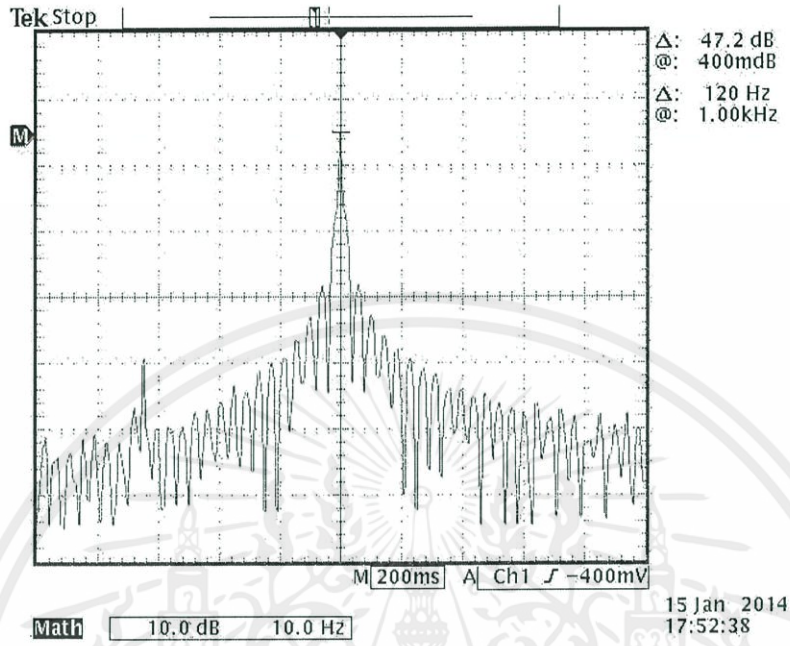
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.65 กราฟความสัมพันธ์ระหว่าง BER กับ SNR เฉลี่ยของผู้ใช้ทั้ง 3 ราย

จากรูปที่ 4.65 เป็นการแสดงการเปรียบเทียบค่าระหว่าง Bit Error Rate กับ Signal to Noise Ratio เฉลี่ยของผู้ใช้ทั้ง 3 ราย โดยที่ค่า SNR ที่อยู่ในแกน x ที่เพิ่มขึ้น จะส่งผลให้ค่า BER ในแกน y มีค่าลดลง

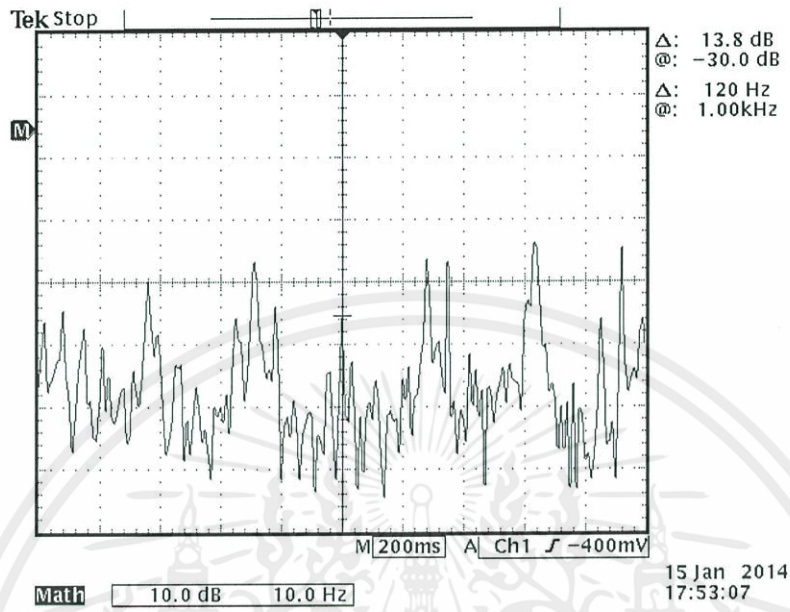
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.66 สเปกตรัมของสัญญาณข้อมูลก่อนการสเปรดสเปกตรัม

จากรูปที่ 4.66 แสดงถึงสเปกตรัมของสัญญาณข้อมูลเดิมซึ่งมีย่านความถี่แบนด์แคบ และจากรูปที่ 4.77 แสดงถึงสเปกตรัมของข้อมูลซึ่งถูกทำให้ย่านความถี่เกิดการแผ่กระจายคล้ายกับสัญญาณรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.67 สเปกตรัมของสัญญาณข้อมูลหลังการสเปรดสเปกตรัม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผลการทดลอง

การออกแบบการทำงานของชุดสาธิตวงจรเครื่องรับส่งสเปรดสเปกตรัม ทำให้มีความเข้าใจและสามารถนำความรู้มาประยุกต์ใช้ในการออกแบบระบบการสื่อสารดิจิตอลโดยใช้ทฤษฎีการสเปรดสเปกตรัม ซึ่งการออกแบบแบ่งออกเป็น 2 ส่วน คือ

ทางด้านซอฟต์แวร์ ได้ออกแบบการจำลองการทำงานการสื่อสารระบบสเปรดสเปกตรัมด้วยโปรแกรม MATLAB

ทางด้านฮาร์ดแวร์ ได้ใช้อุปกรณ์ FPGA ในการสร้างชุดสาธิตวงจรเครื่องรับส่งสเปรดสเปกตรัม ทางด้านภาคส่งได้ทำการสเปรดสัญญาณและการมอดูเลต ก่อนส่งไปยังภาครับซึ่งประกอบด้วยการดีสเปรดสัญญาณ การดีมอดูเลตและวงจรการตัดสินใจข้อมูล ซึ่งการออกแบบได้ใช้ต้นแบบเดียวกันกับการออกแบบการจำลองการทำงานการสื่อสารระบบสเปรดสเปกตรัมด้วยโปรแกรม MATLAB

5.2 ข้อเสนอแนะ

ในการออกแบบการทำงานของชุดสาธิตวงจรดิจิตอลด้วย FPGA โดยใช้ภาษา VHDL ต้องคำนึงถึงจังหวะต่างๆในการดำเนินการในวงจร ซึ่งต้องออกแบบโดยการใช้สัญญาณควบคุมการทำงานของทุกส่วนให้ถูกจังหวะเพื่อที่จะสามารถรับข้อมูลได้อย่างถูกต้องมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] John G. Proakis. *Digital Communications*. 3rd ed. McGraw-Hill, 1995.

[2] Proaki, J. G. and Salehi, M., *Communication Systems Engineering*, 2nd

[3] Sklar, B., *Digital Communications: Fundamentals and Applications*, 2nd Edition, Prentice Hall PTR, 2001.

[4] S.G. Glisic and B. Vucetic, *Spread Spectrum CDMA Systems for Wireless Communications*, Artech House, 1997.

[5] H. Simon, *Communication Systems*, 4th ed., John Wiley&Sons, 2001.

[6] ลัญฉกร วุฒิสิทธิกุลกิจ. *เทคโนโลยีการสื่อสารไร้สาย CDMA (CDMA TECHNOLOGY)* กรุงเทพฯ : สำนักพิมพ์แห่งจุฬาลงกรณ์มหาวิทยาลัย, 2533.

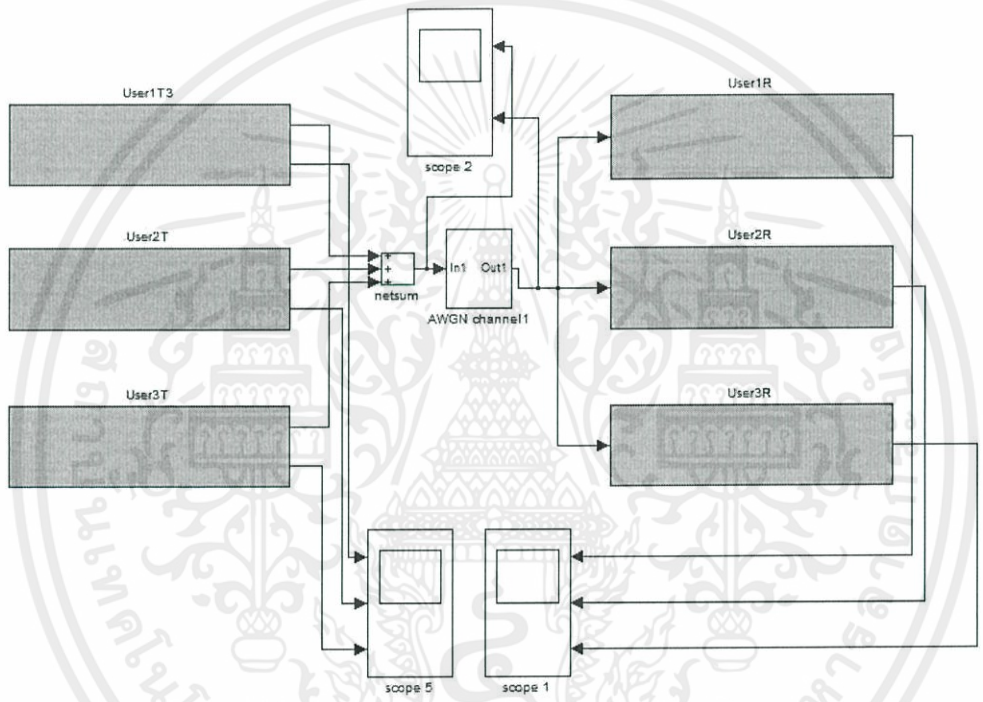
[7] ชำนาญ ปัญญาใส. *ภาษา VHDL สำหรับการออกแบบวงจรดิจิตอล*. กรุงเทพฯ : ซีเอ็ดยูเคชั่น, 2547.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

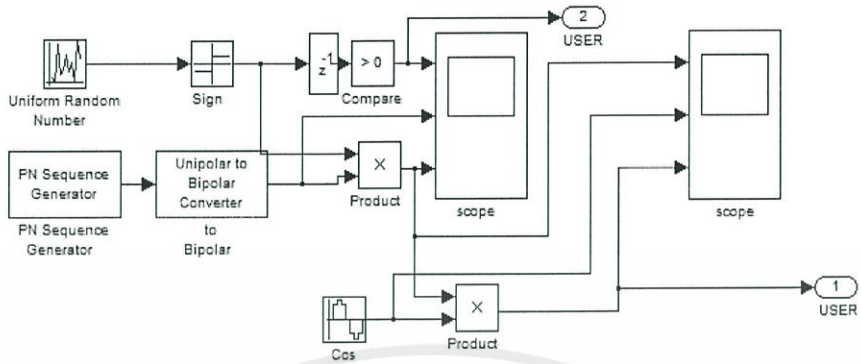
โปรแกรม MATLAB

การจำลองการทำงานของระบบสื่อสารแบบไดเรกต์ซีเควนซ์สเปรดสเปกตรัมซึ่งแสดงให้เห็นในรูปที่ 1ก โดยใช้การออกแบบการรับส่งข้อมูลผ่านการจำลองด้วยโปรแกรม MATLAB

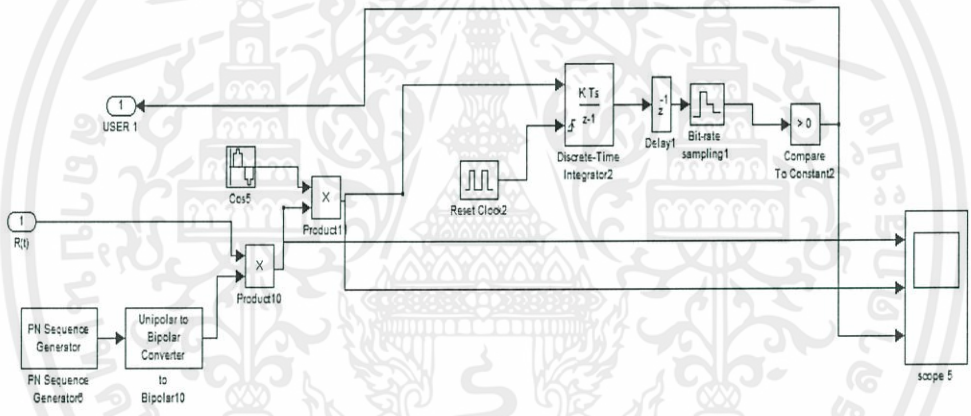


รูปที่ 1ก แบบจำลองของระบบไดเรกต์ซีเควนซ์สเปรดสเปกตรัม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2ก แบบจำลองของภาคส่ง

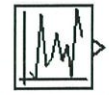


รูปที่ 3ก แบบจำลองของภาครับ

จากรูปที่ 2ก ได้แสดงถึงโมเดลไดเรกต์ซีเควนซ์สเปรดสเปกตรัมของภาคส่งและจากรูปที่ 3ก ได้แสดงถึงโมเดลไดเรกต์ซีเควนซ์สเปรดสเปกตรัมของภาครับด้วยการจำลองผ่านโปรแกรม MATLAB ในภาคผนวกนี้จะกล่าวถึงการเซ็ทค่าพารามิเตอร์ของบล็อกต่างๆ ที่ใช้ในระบบ

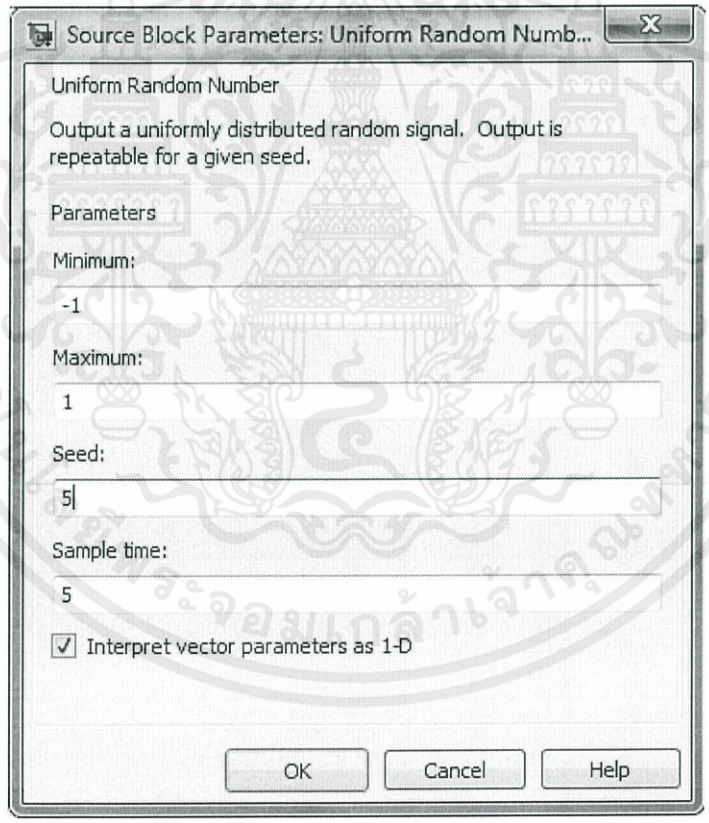
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Uniform Random Number



รูปที่ 4ก บล็อก Uniform Random Number

จากรูปที่ 4ก แสดงถึงบล็อก Uniform Random Number ซึ่งเป็นการสร้างสัญญาณจากการสุ่มค่าโดยลำดับการสุ่ม



รูปที่ 5ก การตั้งค่าพารามิเตอร์ของบล็อก Uniform Random Number

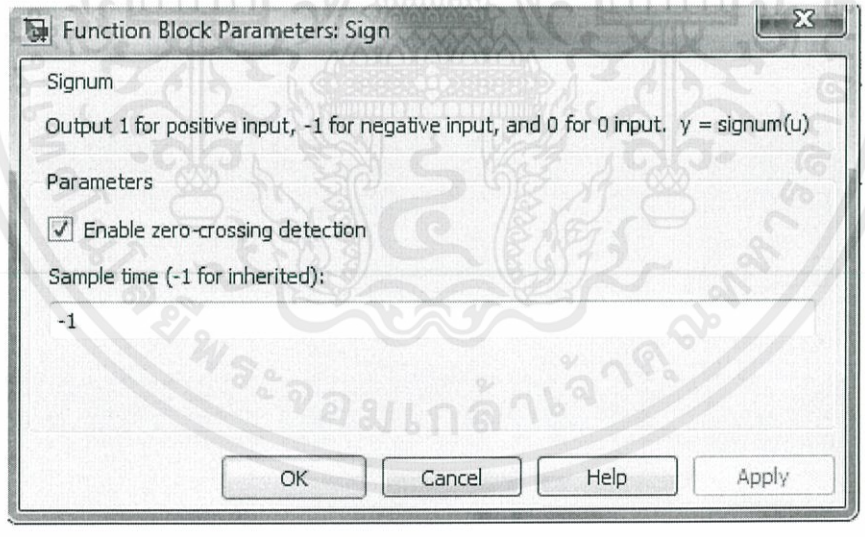
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 5ก จะแสดงถึงการตั้งค่าพารามิเตอร์ของบล็อก Uniform Random Number โดยกำหนดให้ Minimum เท่ากับ -1 , Maximum เท่ากับ +1 , Seed เท่ากับ 5 และ Sample Time เท่ากับ 5

Sign

รูปที่ 6ก บล็อก Sign

จากรูปที่ 6ก แสดงถึงการแปลงสัญญาณบิตข้อมูลที่สุ่มจาก Uniform Random Number ให้มีค่าเท่ากับ 1 และ -1

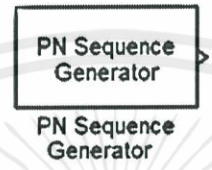


รูปที่ 7ก การตั้งค่าพารามิเตอร์ของบล็อก Sign

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 7ก แสดงถึงการตั้งค่าพารามิเตอร์ของบล็อก Sign โดยกำหนดให้ Sample time มีค่าเท่ากับ -1

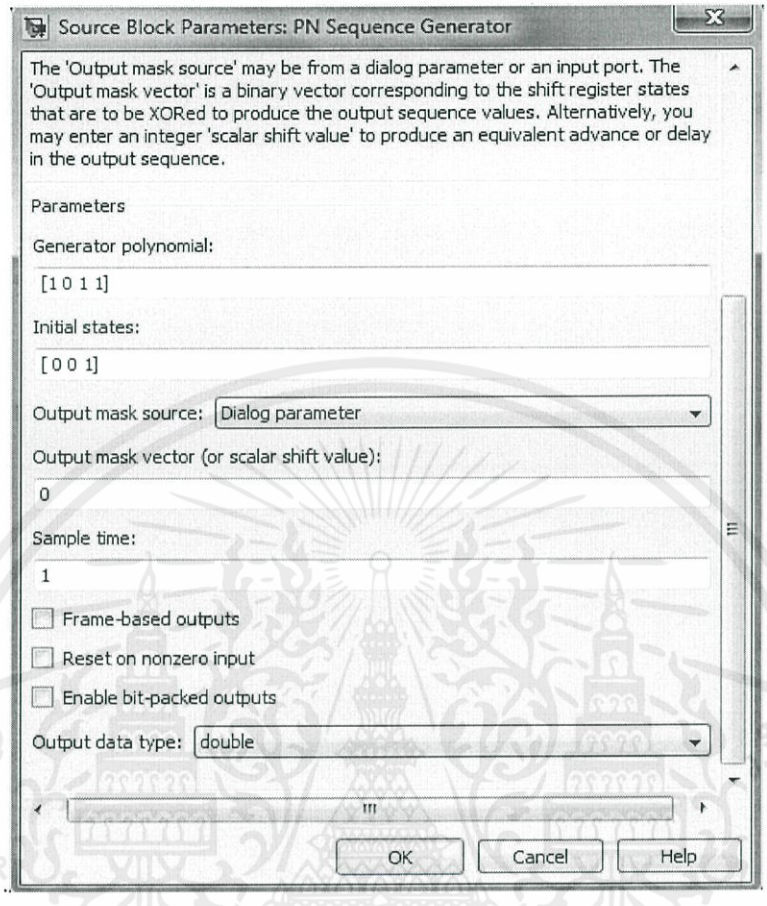
PN Sequence Generator



รูปที่ 8ก บล็อก PN Sequence Generator

จากรูปที่ 8ก แสดงถึงการสร้างรหัสพีเอ็น ซึ่งในแต่ละผู้ใช้จะถูกกำหนดค่าเริ่มต้นของรหัสไม่เหมือนกันเพื่อเป็นการแบ่งแยกข้อมูลของแต่ละผู้ใช้ โดยผู้ใช้แต่ละรายจะปรับค่า Initial states ที่แตกต่างกัน ซึ่งค่า Initial states ทั้งในภาครับและภาคส่งจะต้องเป็นค่าเดียวกันและค่า Generator polynomial จะต้องเหมือนกันทั้งระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

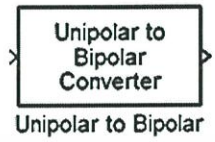


รูปที่ 9ก การตั้งค่าพารามิเตอร์ของบล็อก PN Sequence Generator

จากรูปที่ 9ก แสดงถึงการตั้งค่าพารามิเตอร์ของบล็อก PN Sequence Generator โดยกำหนดให้ Generator polynomial เท่ากับ 1 0 1 1 และ Initial states เท่ากับ 0 0 1

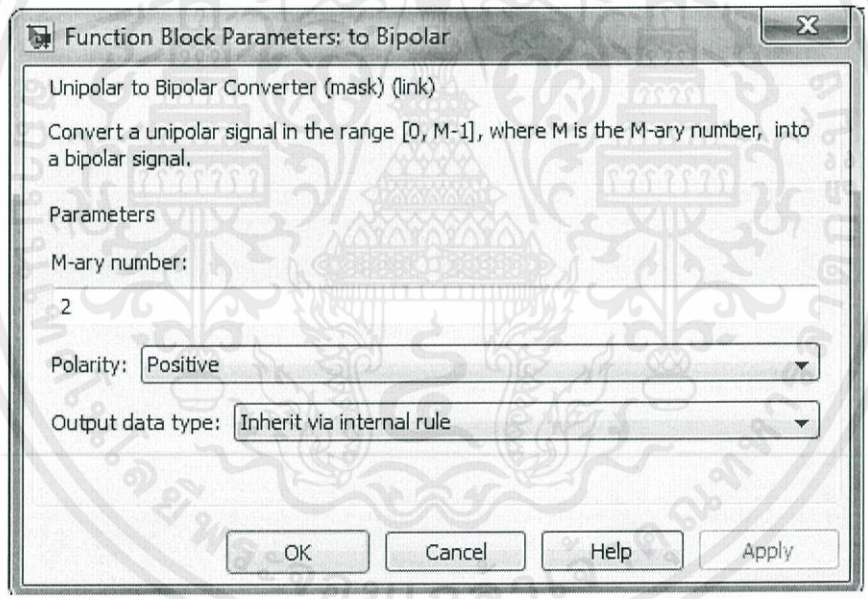
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Unipolar to Bipolar Converter



รูปที่ 10ก บล็อก Unipolar to Bipolar Converter

จากรูปที่ 10ก แสดงถึงการแปลงสัญญาณบิตที่ได้จาก PN Sequence Generator ให้มีค่าเท่ากับ 1 และ -1



รูปที่ 11ก การตั้งค่าพารามิเตอร์ของบล็อก Unipolar to Bipolar Converter

จากรูปที่ 11ก แสดงถึงการตั้งค่าพารามิเตอร์ของบล็อก Unipolar to Bipolar Converter โดยกำหนดให้ M-ary number เท่ากับ 2

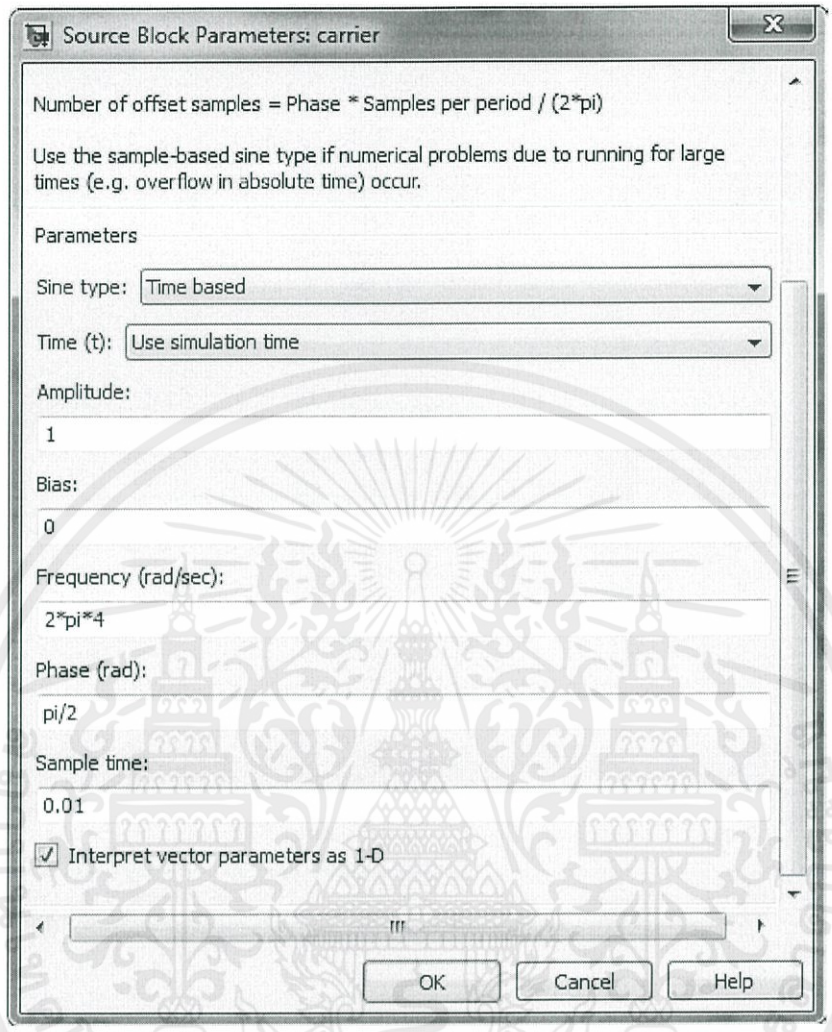
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Carrier

รูปที่ 12ก บล็อก Carrier

จากรูปที่ 12ก แสดงถึงการสร้างสัญญาณคลื่นพาหะเพื่อนำไปคูณกับรหัสพีเอ็น ทำให้ได้สัญญาณที่อยู่ในรูปแบบของสัญญาณ Binary Phase Shift Keying (BPSK signal) ซึ่งค่า Frequency Phase และ Sample time ต้องมีค่าเท่ากันของผู้ใช้ทุกรายทั้งในภาคส่งและภาครับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 13ก การตั้งค่าพารามิเตอร์ของบล็อก Carrier

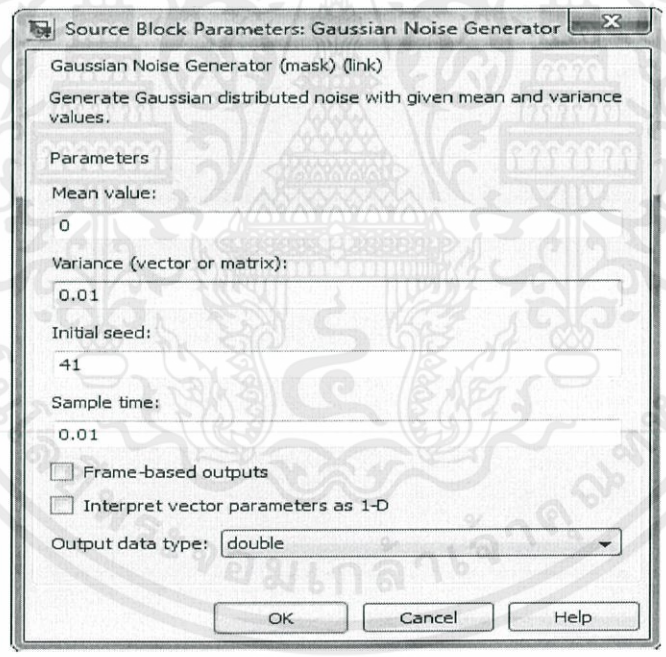
จากรูปที่ 13ก แสดงถึงการตั้งค่าพารามิเตอร์ของบล็อก Carrier โดยกำหนดให้ Amplitude เท่ากับ 1, Bias เท่ากับ 0, Frequency เท่ากับ 2*pi*4, Phase เท่ากับ pi/2 และ Sample time เท่ากับ 0.01

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Gaussian Noise Generator

รูปที่ 14ก บล็อก Gaussian Noise Generator

จากรูปที่ 14ก แสดงถึงบล็อกของสัญญาณรบกวนเกาส์สีขาวแบบบวกของระบบการสื่อสารไดเรกต์ซีเควนซ์สเปรดสเปกตรัม

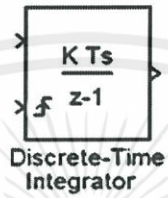


รูปที่ 15ก การตั้งค่าพารามิเตอร์ของบล็อก Gaussian Noise Generator

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 15ก แสดงถึงการตั้งค่าพารามิเตอร์ของบล็อก Gaussian Noise Generator โดยในกรณีศึกษาจะทำการปรับค่า Variance ให้มีค่าเท่ากับ 0.01

Discrete-Time Integrator

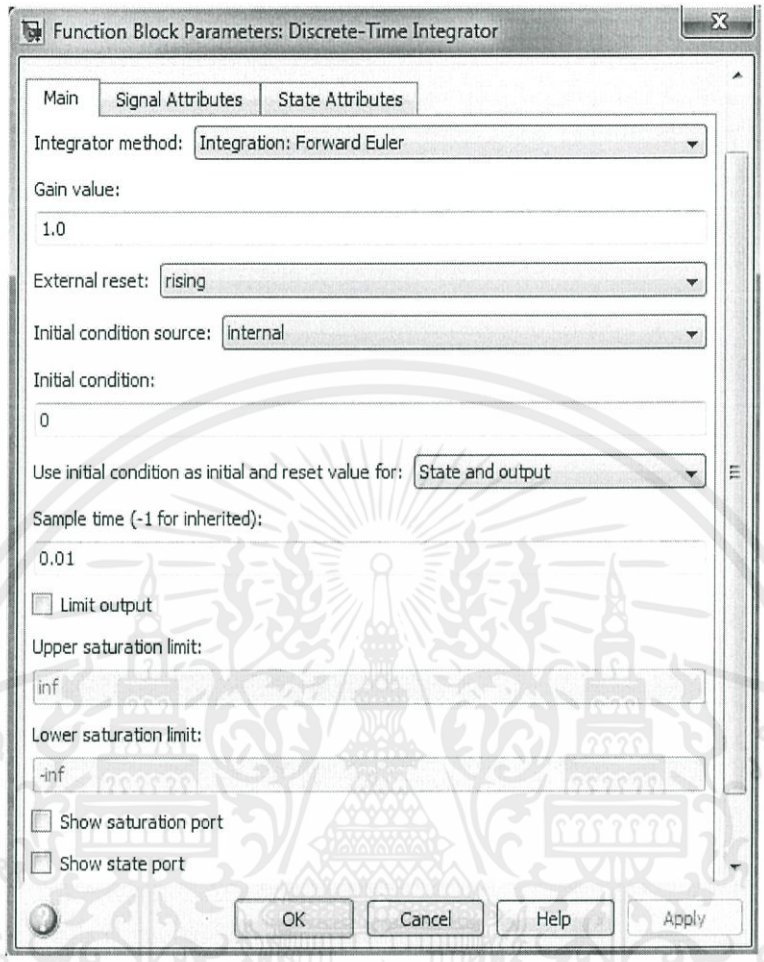


รูปที่ 16ก บล็อก Discrete-Time Integrator

จากรูปที่ 16ก แสดงถึงบล็อก Discrete-Time Integrator โดยค่าที่ได้รับจากการอินทีเกรตพื้นที่ใต้กราฟของสัญญาณหลังจากการดีมอดูเลต ซึ่งค่าที่ได้รับสำหรับช่วงเวลาแต่ละบิตเป็นการรวมพลังงานของค่าที่ได้รับในแต่ละบิต

การเซ็ทค่าพารามิเตอร์ของบล็อก Discrete-Time Integrator จะต้องกำหนดให้ Gain value เท่ากับ 1.0, Initial condition เท่ากับ 0 และ Sample time เท่ากับ 0.01 ดังรูปที่ 17ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



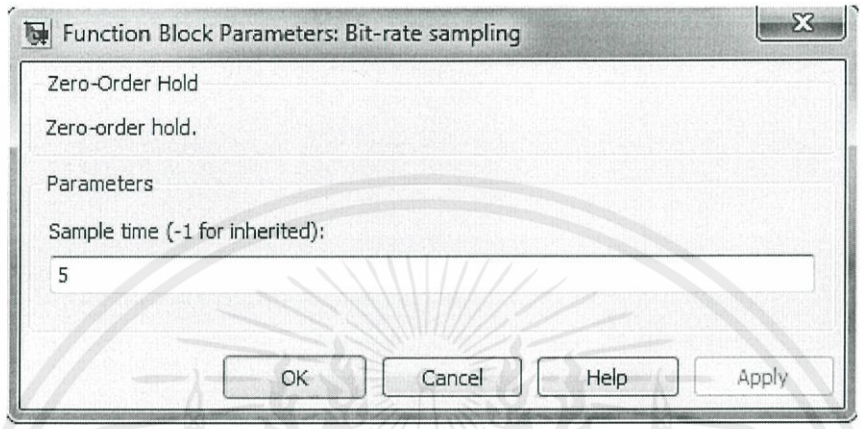
รูปที่ 17ก การตั้งค่าพารามิเตอร์ของบล็อก Discrete-Time Integrator

Bit-rate sampling and Compare to zero

รูปที่ 18ก บล็อก Bit-rate sampling and Compare to zero

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 18ก แสดงถึงบล็อก Bit-rate sampling and Compare to zero เป็นกระบวนการจัดระดับสัญญาณ เพื่อให้ได้สัญญาณกลับคืนมาอย่างถูกต้อง



รูปที่ 19ก การตั้งค่าพารามิเตอร์ของบล็อก Bit-rate sampling and Compare to zero

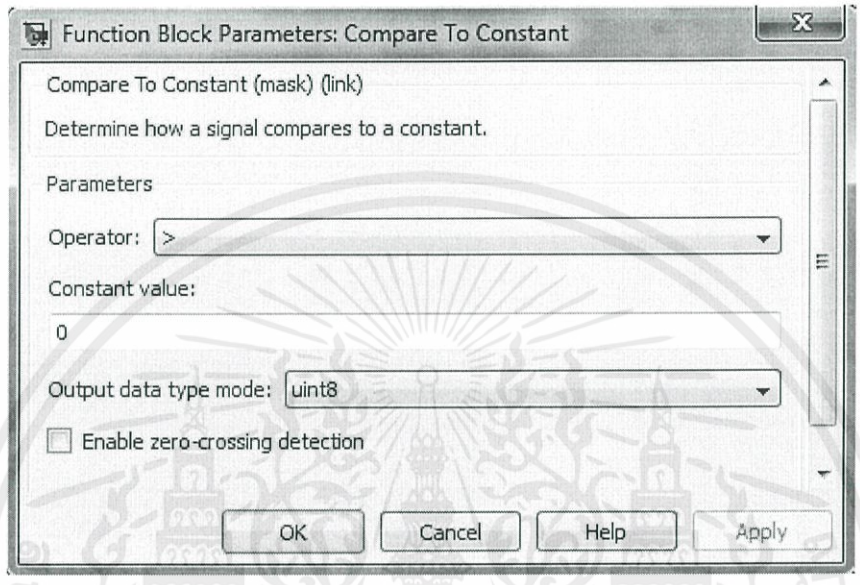
จากรูปที่ 19ก แสดงถึงการตั้งค่าพารามิเตอร์ของบล็อก Bit-rate sampling and Compare to zero โดยกำหนดให้ Sample time มีค่าเท่ากับ 5

Compare to Constant

รูปที่ 20ก บล็อก Compare to Constant

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 20ก แสดงถึงบล็อก Compare to Constant ซึ่งเป็นการปรับค่าสัญญาณจากบิตที่เป็น 1 ให้เป็น 1 และจากบิตที่เป็น -1 ให้เป็น 0



รูปที่ 21ก การเซ็ทค่าพารามิเตอร์ของบล็อก Compare to Constant

จากรูปที่ 21ก แสดงถึงการเซ็ทค่าพารามิเตอร์ของบล็อก Compare to Constant โดยกำหนดให้ Operator เท่ากับ >, Constant value เท่ากับ 0 และ Output data type mode เท่ากับ uint8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมสร้างสัญญาณไซน์

```
close all;
clear all;
clc;
%----- generate signal -------
Am = 1;
fm = 10;
Fs =1000;
N = 0:99;
t = N./Fs;
x = Am.*sin(2*pi*fm*t);
 figure(1)
 stem(t,x);xlabel('time in sec');ylabel('amplitude');title('input
signal');grid;
xx = dec2bin(127*x+256,8);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

โปรแกรมภาษา VHDL

1. โปรแกรมวงจรหารความถี่

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;
entity divider is
        port (clk: in std_logic;
                clkout : out std_logic);
end;

architecture rtl of divider is
                signal count : INTEGER RANGE 0 TO 100 := 0;
                signal co : INTEGER RANGE 0 TO 100 := 0;
begin
        process(clk) --,reset
                begin
                if (clk'event AND clk = '1' )THEN
                                if (count < 98) THEN
                                        if (co = 0) THEN
                                                clkout <= '1'     ;
                                                co <= co + 1;
                                    else
                                        count <= count + 1;
                                        clkout <= '0';
                                        end if;
                                else

                                        --clkout <= '0'   ;
                                        count <= 0;
                                        co <= 0;

                                end if;

                end if;
end process;
end rtl;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. โปรแกรมวงจรชิฟรีจิสเตอร์

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;
entity shf is
        port (reset,clk : in std_logic;  --,init
                        --init_val : in std_logic_vector(2 downto 0);
                        pn1 : out std_logic);

end;

architecture rtl of shf is
                        signal init_reg : std_logic_vector(2 downto 0) := "011";

begin
                        process(clk,reset) --,init
                        begin

                        if reset = '1' then
                                init_reg <= "011";
                                        elsif clk'event and clk = '1' then
                                        init_reg(2) <= init_reg(1) xor init_reg(0);
                                        init_reg(1) <= init_reg(2);
                                        init_reg(0) <= init_reg(1);

                                        end if;

                        end process;
                pn1 <= init_reg(0);
                pn2 <= init_reg(1);
                pn3 <= init_reg(2);


end rtl;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. โปรแกรมวงจรแมปค่า

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;

entity bpsk_mapper is
        port(din : in std_logic;
                bpsk : out std_logic_vector(1 downto 0));
end;

architecture rtl of bpsk_mapper is
begin
        process(din)
        begin
                case din is
                        when '0' => bpsk <="11";
                        when others => bpsk <= "01";
                end case;
        end process;
end;
```

4. โปรแกรมวงคูณสัญญาณ

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
USE ieee.std_logic_arith.ALL;
USE ieee.std_logic_signed.ALL;
entity multiplier is
        port(   u,p     : in std_logic_vector(1 downto 0);

                        c               : out  integer);
end multiplier;

architecture Behavioral of multiplier is
SIGNAL a1, b1 : integer;
SIGNAL c1 : integer;
begin
        a1 <= (conv_integer(u));
        b1 <= (conv_integer(p));
        c <= a1*b1;

        --b      <=      u*p;

end Behavioral;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. วงจรนับค่า

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
entity adder is
   Port (-- a : in  STD_LOGIC_VECTOR (7 downto 0);
                        reset,clk: in  STD_LOGIC;
         So : out  STD_LOGIC_VECTOR (7 downto 0));
end adder;

architecture Behavioral of adder is
begin
        process(clk,reset)--(clk)
                variable a : std_logic_vector (7 downto 0) := "11111111";
                variable b : std_logic_vector (7 downto 0) := "00000001";
                variable e,d : std_logic_vector (7 downto 0) := "11111111";
                variable c : std_logic := '1';

        begin
                if reset = '1' then
                    so<= (others =>'0');
                    a:=e;
                    elsif a="01100011" then
                        a:=e;

                        elsif clk'event and clk = '1' then
                        a := a+b;
                        So<=a(7 downto 0);

                 end if;
        --end if;
        end process;

end Behavioral;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. วงจรกำเนิดสัญญาณพาหะ

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_SIGNED.ALL;
entity sine is
                Port ( address : in  STD_LOGIC_VECTOR (7 downto 0);
        data : out  STD_LOGIC_VECTOR (7 downto 0));
end sine;
architecture Behavioral of sine is
        type vector_array is array (0 to 99) of std_logic_vector (7 downto 0);
        constant memory : vector_array :=("10000000",
"10000011",
"10000111",
"10001011",
"10001111",
"10010011",
"10010111",
"10011010",
        .
        .
        .
        .
"10011110",
"10100001",
"01000010",
"01000010",
"01000100",
"01010010",
"01111100");

begin
        data <= memory(conv_integer(address));
end Behavioral;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. วงจรบวกสัญญาณ

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;

entity sum is
        port(   sp_u1,sp_u2,sp_u3       : in std_logic_vector(11 downto 0);
                        summ                                    : out std_logic_vector(7 downto
0));
end sum;

architecture Behavioral of sum is
        signal sum : std_logic_vector(11 downto 0);
begin
        sum <= sp_u1+sp_u2+sp_u3;
        summ <= sum(11 downto 4);

end Behavioral;
```

8. วงจรบวกสะสมค่า

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;

entity accu is
  port(clk, reset : in  std_logic;
       D : in std_logic_vector(17 downto 0);
       Q : out std_logic_vector(27 downto 0));
end accu;
architecture rtl of accu is
  signal tmp: std_logic_vector(27 downto 0);
  begin
    process (Clk, reset)
      begin
       if (reset='1') then
         tmp <= "000000000000000000000000000"+D+d;
       elsif (clk'event and clk='1') then
         tmp <= tmp + D;
       end if;
    end process;
    Q <= tmp;
end rtl;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

9. วงจรบัฟเฟอร์

```
library IEEE;
use ieee.std_logic_1164.all;
entity buffaccu is
        port (clk : in std_logic;
                        oaccu : in std_logic_vector(27 downto 0);
                        dbuff : out std_logic_vector(27 downto 0));
end;

architecture rtl of buffaccu is

begin
                        process(clk)
                        begin
                                if (clk'event AND clk = '1' )THEN
                                dbuff <= oaccu;
                        end if;


        end process;

                end rtl;
```

10. วงจรตัดสินใจบิต

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;
entity decision is
        port(clk : in std_logic;
                        bin : in std_logic_vector(27 downto 0);
                data : out std_logic);
end;
architecture rtl of decision is
begin
        process(bin,clk)
        begin
                case (bin(27)) is
                        when '0' => data <='1';
                        when others => data <= '0';
                end case;
        end process;
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. โปรแกรมการทำงานของวงจร Direct Sequence Spread Spectrum

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_signed.all;

entity demo is
        port(--usere1,usere2,usere3 : in std_logic;
                        reset,clk4: in std_logic;
                        reseto,clk4o: out std_logic;
                        test : out std_logic_vector(7 downto 0);
                        testr : in std_logic_vector(7 downto 0);
                        adata1,adata2,adata3 : out std_logic;
                        data1,data2,data3 : out std_logic);
end;

architecture rtl of demo is

component divider500
        port (clk: in std_logic;  --,reset
                        clkout : out std_logic);
end component;
component user1 is
        port (reset,clk : in std_logic;
                        user1 : out std_logic);
end component;
component user2 is
        port (reset,clk : in std_logic;
                        user2 : out std_logic);
end component;
component user3 is
        port (reset,clk : in std_logic;
                        user3 : out std_logic);
end component;
---------------------------component SHF-----------------------------------
component shf
        port(reset,clk: in std_logic;
                        pn1 : out std_logic);
end component;
---------------------------component SHF-----------------------------------
component shf2
        port(reset,clk: in std_logic;
                        pn2 : out std_logic);
end component;
---------------------------component SHF-----------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
component shf3
        port(reset,clk: in std_logic;
                        pn3 : out std_logic);
end component;

--------------------------component divider-----------------------------------
component divider
        port(clk: in std_logic;
                        clkout : out std_logic);
end component;
--------------------------component BPSK_mapper-------------------------
component bpsk_mapper
        port(din : in std_logic;
                        bpsk : out std_logic_vector(1 downto 0));
end component;
--------------------------component adder-------------------------
component adder
        Port ( reset,clk: in  STD_LOGIC;
       So : out  STD_LOGIC_VECTOR (7 downto 0));
end component;

--------------------------component sine-------------------------
component sine
        Port ( address : in  STD_LOGIC_VECTOR (7 downto 0);
        data : out  STD_LOGIC_VECTOR (7 downto 0));
end component;
--------------------------component multiplier-------------------------
component multiplier
        port(    u,p        : in std_logic_vector(1 downto 0);
                        b                : out std_logic_vector(3 downto 0));
end component;

--------------------------component multiplier2-------------------------
component multiplier2
        port(    x          : in std_logic_vector(7 downto 0);
                        b          : in std_logic_vector(3 downto 0);
                        sp_u               : out std_logic_vector(11 downto 0));
end component;

--------------------------component sum-------------------------
component sum
        port(    sp_u1,sp_u2,sp_u3        : in std_logic_vector(11 downto 0);
                        summ                                        : out std_logic_vector(7 downto
0));
end component;
```

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
-----------------------------------------------------------component recieve-------------------------------
--------------------
component rmultiplier1
        port(    summ   : in std_logic_vector(7 downto 0);
                        rp   : in std_logic_vector(1 downto 0);
                        rrp              : out std_logic_vector(9 downto 0));
end component;

component rmultiplier2
        port(    dsp    : in std_logic_vector(9 downto 0);
                        sinn   : in std_logic_vector(7 downto 0);
                        dspr             : out std_logic_vector(17 downto 0));
end component;
component diaccu
        port (clk : in std_logic;
                        clkout : out std_logic);
end component;
component accu
  port(clk, reset : in  std_logic;
      d : in std_logic_vector(17 downto 0);
      q : out std_logic_vector(27 downto 0));
end component;
component buffaccu
        port (clk : in std_logic;
                        oaccu : in std_logic_vector(27 downto 0);
                        dbuff : out std_logic_vector(27 downto 0));
end component;
component decision
        port(clk : in std_logic;
                bin : in std_logic_vector(27 downto 0);
                data : out std_logic);
end component;

---------------------------------signaltran-------------------------------
signal clk0,t1,t2,t3,clk5,clk500,userr1,userr2,userr3: std_logic;
signal p1,p2,p3 : std_logic_vector(1 downto 0);
signal u1,u2,u3 : std_logic_vector(1 downto 0);
signal out_adder,summ : std_logic_vector(7 downto 0);
signal x : std_logic_vector(7 downto 0);
signal b1,b2,b3 : std_logic_vector(3 downto 0);
signal sp_u1,sp_u2,sp_u3 : std_logic_vector (11 downto 0);
---------------------------------signalreceive----------------------------
signal rrp1,rrp2,rrp3 : std_logic_vector(9 downto 0);
signal dspr1,dspr2,dspr3 : std_logic_vector(17 downto 0);
```

เอกสารนี้ [illegible] นั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
signal q1,q2,q3,bq1,bq2,bq3 : std_logic_vector(27 downto 0);

begin
-------------------------------------------------------------trans-----------------------------------------
        c001 : divider500 port map (clk4,clk500);
        c002 : user1 port map (reset,clk500,userr1);
        c003 : user2 port map (reset,clk500,userr2);
        c004 : user3 port map (reset,clk500,userr3);
        C0: divider port map (clk4,clk0);
        C101: shf port map (reset,clk0,t1); --_vhdl  --,init,init_val
        C102: shf2 port map (reset,clk0,t2);
        C103: shf3 port map (reset,clk0,t3);
        C2: bpsk_mapper port map(t1,p1);  --bpsk pn code1
        C3: bpsk_mapper port map(t2,p2);              --bpsk pn code2
        C4: bpsk_mapper port map(t3,p3);              --bpsk pn code3
        C5: bpsk_mapper port map(userr1,u1); --bpsk user1
        C6: bpsk_mapper port map(userr2,u2);  --bpsk user2
        C7: bpsk_mapper port map(userr3,u3);  --bpsk user3
        C8: adder port map(reset,clk4,out_adder);
        C9: sine port map(out_adder,x);
        c10: multiplier port map(u1,p1,b1);
        c11: multiplier port map(u2,p2,b2);
        c12: multiplier port map(u3,p3,b3);
        c13: multiplier2 port map(x,b1,sp_u1);
        c14: multiplier2 port map(x,b2,sp_u2);
        c15: multiplier2 port map(x,b3,sp_u3);
        c16: sum port map(sp_u1,sp_u2,sp_u3,test);
-------------------------------------------------------------receive---------------------------------------
        c17: rmultiplier1 port map(testr,p1,rrp1);
        c18: rmultiplier1 port map(testr,p2,rrp2);
        c19: rmultiplier1 port map(testr,p3,rrp3);
        c20: rmultiplier2 port map(rrp1,x,dspr1);
        c21: rmultiplier2 port map(rrp2,x,dspr2);
        c22: rmultiplier2 port map(rrp3,x,dspr3);
        c23: diaccu port map(clk4,clk5);
        c24: accu port map(clk4,clk5,dspr1,q1);
        c25: accu port map(clk4,clk5,dspr2,q2);
        c26: accu port map(clk4,clk5,dspr3,q3);
        c27: buffaccu port map(clk5,q1,bq1);
        c28: buffaccu port map(clk5,q2,bq2);
        c29: buffaccu port map(clk5,q3,bq3);
c30: decision port map(clk5,bq1,data1);
c31: decision port map(clk5,bq2,data2);
c32: decision port map(clk5,bq3,data3);
--test <= x;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
clk4o <= clk4;
reseto <= reset;
adata1 <= userr1;
adata2 <= userr2;
adata3 <= userr3;
end rtl;
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้