

วิทยุติดตามตัวผ่านช่องสัญญาณเอฟเอ็ม เอสซีเอ

FM SCA PAGER



โดย

นางสาวนุชจรี จตุพรฤกษ์

นายพลาว์สถ์ มณีโชติปิติ

เลขหน้า.....
เลขทะเบียน.....42208
วัน, เดือน, ปี 15 พ.ค. 2545

b.....
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

b1190641x

วิทยุติดตามตัวผ่านช่องสัญญาณเอฟเอ็ม เอสซีเอ

FM SCA PAGER

โดย

นางสาวนุชจรี จตุพรฤกษ์ 40010402

นายพลาว์สิทธิ์ มณีโชติปิติ 40010497

อาจารย์ที่ปรึกษา

ศาสตราจารย์ ดร.วิวัฒน์ กิรานนท์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

ปริญญานิพนธ์ ปีการศึกษา 2543

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง วิทยุติดตามตัวผ่านช่องสัญญาณเอฟเอ็ม เอสซีเอ

FM SCA PAGER

ผู้จัดทำ น.ศ.นุชจรี จตุพรฤกษ์ 40010402

นายพลาว์ศักดิ์ มณีโชติปีติ 40010497


(.....) อาจารย์ที่ปรึกษา
ศ.ดร.วิวัฒน์ กิรานนท์



วิทยุติดตามตัวผ่านช่องสัญญาณเอฟเอ็ม เอสซีเอ
FM SCA PAGER

โดย	นางสาวนุชจรี จตุพรฤกษ์	40010402
	นายพลาวัสก์ มณีโชติพิติ	40010497

อาจารย์ที่ปรึกษา ศ.ดร.วิวัฒน์ กิรานนท์

บทคัดย่อ

โครงการนี้เป็นการประยุกต์ใช้งานการส่งข้อมูลระบบดิจิทัลผ่านทางความถี่วิทยุ โดยใช้ช่องสัญญาณเอฟเอ็ม เอสซีเอ (FM-SCA) ซึ่งมีแบนด์วิด (Bandwidth) ประมาณ 16 กิโลเฮิร์ต การมอดูเลตสัญญาณข่าวสารเข้ากับคลื่นพาห์ โดยวิธีการมอดูเลตแบบไบนารีเฟสชิฟติง (BPSK) ซึ่งมีความถี่คลื่นพาห์เท่ากับ 67.2 กิโลเฮิร์ต และสัญญาณข่าวสารทางค่านอินพุตมีอัตราการส่งเท่ากับ 4800 บิตต่อวินาที สัญญาณข่าวสารทางค่านอินพุตประกอบด้วย หมายเลขประจำเครื่องรับ และข่าวสารที่ต้องการส่ง ด้านเครื่องรับจะเลือกรับเฉพาะข่าวสารที่มีหมายเลขตรงกัน แล้วนำมาดีมอดูเลต ส่งออกแสดงหน้าจอแอลซีดี (LCD) ซึ่งควบคุมโดยไมโครคอนโทรลเลอร์ 8051 (Microcontroller)

ABSTRACT

This project is an application of digital data transmission by using the FM radio wave broadcasting. The FM-SCA channel is used for transmission channel which bandwidth is 16 kHz. The BPSK modulation technique is used to modulate the carrier and the digital baseband input. The carrier of BPSK is 67.2 kHz and bit rate of baseband input is 4800 bps. The digital baseband input is transmitted. Which consists of ID code (Number of Page) and message. The receiver will receive the only same ID code and demodulate. The demodulated message is shown on LCD which is controlled by MCS-51.

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 การมอดูเลชันเชิงความถี่ (Frequency modulation)	2
2.1.1 ลักษณะของคลื่นเอฟเอ็ม	2
2.1.2 ผลจากการเปลี่ยนแปลงขนาดของสัญญาณ	3
2.1.3 ผลจากการเปลี่ยนแปลงความถี่ของสัญญาณ	4
2.1.4 การวิเคราะห์สัญญาณเอฟเอ็ม	4
2.2 เครื่องส่ง FM	5
2.3 เครื่องรับ FM	7
2.4 เอฟเอ็ม เอสซีเอ (FM SCA)	8
2.4.1 การส่งและรับสัญญาณระบบ SCA	10
2.4.1.1 การส่ง	10
2.4.1.2 การรับ	11
2.4.2 การพิจารณาเกี่ยวกับการรบกวนในกรณีสถานีวิทยุ FM stereo ที่ส่ง SCA ด้วยการรบกวนข้างเคียง	11
2.4.3 การรบกวนกันเองภายในระบบ	12
2.5 การมอดูเลตสัญญาณดิจิทัล	13
2.5.1 เฟสชิฟต์อิง (PSK)	14
2.5.2 หลักการของไบนารี PSK โมดูเลเตอร์และดีโมดูเลเตอร์	17
2.5.3 โคอีเรนซ์ที่แตกชั้น	19
2.5.4 การผิดพลาดทางเฟส (phase ambiguity)	23
2.5.5 ชนิดของสัญญาณ	29
2.6 ทฤษฎีของเฟสล็อกกลูป	31
2.6.1 การออกแบบ เฟสล็อกกลูป	34
2.7 โมดูลแสดงผลแบบผลึกเหลว	35
2.7.1 โครงสร้างภายในของตัวควบคุมโมดูลแอลซีดี	35
2.7.2 โมดูลแอลซีดีขนาด 16 ตัวอักษร 1 บรรทัด (LCD 16×1)	37
2.7.3 คำสั่งควบคุมโมดูลแอลซีดี	37
2.7.4 การเขียนคำสั่งและข้อมูลให้แก่โมดูลแอลซีดี	40
2.7.5 จังหวะการทำงานของแอลซีดี โมดูล	41
บทที่ 3 การคำนวณและการสร้าง	42
3.1 การสร้างวงจรภาคส่ง	42

3.1.1	วงจรผลิตสัญญาณความถี่ 67.2 กิโลเฮิร์ตซ์	42
3.1.1.1	วงจรผลิตสัญญาณอ้างอิงความถี่ 4800 เฮิร์ตซ์	43
3.1.1.2	วงจรเฟสล็อกกลุ๊ป	44
3.1.2	วงจรเข้ารหัสสัญญาณเอ็นอาร์แซดแบบมาร์ค (Non Return to Zero – Mark)	45
3.1.3	วงจรคูณสัญญาณพาหะกับสัญญาณข่าวสาร	46
3.1.4	การออกแบบวงจรกรองความถี่แบนด์พาสฟิลเตอร์	46
3.1.5	วงจรเครื่องส่งเอฟเอ็ม	49
3.2	การสร้างวงจรภาคเครื่องรับ	50
3.2.1	วงจรกู้ความถี่คลื่นพาห้	51
3.2.2	วงจรยกกำลังสอง	51
3.2.3	วงจรเฟสล็อกกลุ๊ป	52
3.2.4	วงจรหารสอง	52
3.2.5	วงจรคูณสัญญาณคลื่นพาห้กับสัญญาณพีเอสเค	53
3.2.6	วงจรกู้สัญญาณนาฬิกา	54
3.2.7	วงจรถอดรหัสสัญญาณเอ็นอาร์แซด ไอ (NRZ-I) เป็นเอ็นอาร์แซด (NRZ)	56
3.2.8	วงจรเครื่องรับเอฟเอ็ม	58
3.2.9	ชุดแสดงผล	60
3.2.10	การออกแบบซอฟต์แวร์ที่ใช้แสดงผล	62
บทที่ 4	ผลการทดลอง	43
4.1	การทดลองวงจรภาคส่ง	65
4.1.1	วงจรเข้ารหัสสัญญาณเอ็นอาร์แซดเป็นเอ็นอาร์แซด ไอ	65
4.1.2	วงจรกำเนิดสัญญาณความถี่คลื่นพาห้ 67.2 กิโลเฮิร์ตซ์	65
4.1.3	วงจรคูณสัญญาณคลื่นพาห้และสัญญาณข่าวสาร	66
4.1.4	วงจรกรองความถี่ผ่าน 67.2 กิโลเฮิร์ตซ์	67
4.2	การทดลองวงจรภาครับ	69
4.2.1	วงจรกู้สัญญาณคลื่นพาห้	69
4.2.2	วงจรคูณสัญญาณพีเอสเคกับคลื่นพาห้	71
4.2.3	วงจรกู้สัญญาณนาฬิกา	72
4.2.4	วงจรถอดรหัสสัญญาณ	72
4.2.5	โปรแกรมส่งข้อมูล	75
บทที่ 5	สรุปและวิจารณ์	80
บรรณานุกรม		
กิตติกรรมประกาศ		

สารบัญรูปภาพ

	หน้า
รูปที่ 1.1 บล็อกไดอะแกรมของวงจรภาคส่ง	1
รูปที่ 1.2 บล็อกไดอะแกรมของวงจรภาครับ	1
รูปที่ 2.1 การแปรผันทางความถี่ของคลื่นพาห์โดยสัญญาณ	2
รูปที่ 2.2 รูปคลื่นเอฟเอ็มที่เกิดจากสัญญาณมีขนาดต่างๆกัน	3
รูปที่ 2.3 การเปลี่ยนแปลงความถี่เบี่ยงเบนโดยขนาดของสัญญาณ	3
รูปที่ 2.4 รูปคลื่นเอฟเอ็มที่เกิดจากสัญญาณมีความถี่ต่างๆกัน	4
รูปที่ 2.5 ผลของค่าเบี่ยงเบนต่อ FM/PM เทียบกับพารามิเตอร์ของอินเทลลิเจนซ์	5
รูปที่ 2.6 แผนผังเครื่องส่ง FM อย่างง่าย	6
รูปที่ 2.7 แผนผังเครื่องส่งกระจายเสียง FM แบบคูณความถี่ (มัลติพลาย)	6
รูปที่ 2.8 แผนผังของเครื่องรับ FM	7
รูปที่ 2.9 แสดงการใช้แถบความถี่ของสัญญาณในระบบ FM mono	8
รูปที่ 2.10 แสดงการใช้แถบความถี่ของสัญญาณในระบบ mono รวมกับ SCA	9
รูปที่ 2.11 แสดงการใช้แถบความถี่ของสัญญาณในระบบ FM stereo	9
รูปที่ 2.12 แสดงการใช้แถบความถี่ของสัญญาณในระบบ FM stereo รวมกับ SCA	9
รูปที่ 2.13 แสดง Block Diagram โดยสังเขป ของเครื่องส่งวิทยุกระจายเสียง ระบบ stereo phonic + SCA	10
รูปที่ 2.14 แสดง Block Diagram โดยสังเขปของเครื่องรับวิทยุ FM stereo ที่ สามารถเลือกรับสัญญาณ SCA ได้ด้วย	11
รูปที่ 2.15 แสดงความกว้างแถบความถี่ของคลื่นสัญญาณ สถานีวิทยุจุฬาฯ, ขสทป. และ ทอ.ที่ส่งระบบ SCA ด้วยทั้ง 3 สถานี	12
รูปที่ 2.16 แถบความถี่สัญญาณ SCA	13
รูปที่ 2.17 สัญญาณเบสแบนด์คิจิตอล	13
รูปที่ 2.18 เปรียบเทียบวิธีการมอดูเลตแบบคิจิตอลและอานาล็อก	14
รูปที่ 2.19 phase shift keying	14
รูปที่ 2.20 ไบนารี PSK	15
รูปที่ 2.21 สัญญาณคิจิตอล 2 บิต ที่จะส่งพร้อมกัน	15
รูปที่ 2.22 Quadri PSK	15
รูปที่ 2.23 8- เฟส PSK	16
รูปที่ 2.24 เปรียบเทียบอัตราการผิดพลาดของบิตของ PSK แบบต่างๆ	16
รูปที่ 2.25 การรับ - ส่ง สัญญาณคิจิตอล	17
รูปที่ 2.26 การ โมดูเลตสัญญาณคิจิตอล	17
รูปที่ 2.27 การเปลี่ยนแปลงทางเฟสของคลื่นพาหะ	18

รูปที่ 2.28 การสร้างสัญญาณขึ้นใหม่ทางด้านการรับ	18
รูปที่ 2.29 การรับสัญญาณ PSK ทางด้านการรับ	19
รูปที่ 2.30 สัญญาณ output หลังจากผ่าน LPF	19
รูปที่ 2.31 การพิจารณาหาค่าเป็น 1 หรือ เป็น 0 โดย sample holder	20
รูปที่ 2.32 รหัสผิดพลาดเนื่องจากเสียงรบกวน	20
รูปที่ 2.33 บล็อกไดอะแกรมของ coherent demodulator	21
รูปที่ 2.34 หลักการ carrier recovery circuit	21
รูปที่ 2.35 carrier recovery circuit และสัญญาณตามจุดต่างๆ	22
รูปที่ 2.36 การทำงานของ phase lock loop	22
รูปที่ 2.37 วงจร clock recovery	23
รูปที่ 2.38 การผิดพลาดทางเฟส	23
รูปที่ 2.39 คลื่นสัญญาณเมื่อผ่านวงจรหาความถี่	24
รูปที่ 2.40 เปรียบเทียบ output กรณี t_1 และ t_2 เป็นจุดเริ่ม	24
รูปที่ 2.41 สัญญาณดิจิทัลหลังผ่านการดีมอดูเลต	25
รูปที่ 2.42 differential coherent PSK	25
รูปที่ 2.43 วงจร differential encoder	26
รูปที่ 2.44 การส่งสัญญาณที่ differential encoder	26
รูปที่ 2.45 รหัสสัญญาณ ณ ตำแหน่งต่างๆ	27
รูปที่ 2.46 differential decoder	27
รูปที่ 2.47 ลำดับสัญญาณดิจิทัลด้านรับ	28
รูปที่ 2.48 การเกิดข้อผิดพลาดของบิต	28
รูปที่ 2.49 การรับส่งสัญญาณดิจิทัล	29
รูปที่ 2.50 แสดงการเข้ารหัสสัญญาณดิจิทัลในรูปแบบต่างๆ	30
รูปที่ 2.51 บล็อกไดอะแกรมของเฟสล็อกลูป	31
รูปที่ 2.52 วงจรเฟสดีเทกเตอร์แบบ I และการทำงาน	32
รูปที่ 2.53 วงจรเฟสดีเทกเตอร์แบบ II และการทำงาน	32
รูปที่ 2.54 ก) เฟสล็อกลูปแบบออร์เดอร์ลูปอันดับหนึ่งและ ข) เฟสล็อกลูปแบบออร์เดอร์ลูปอันดับ 2	34
รูปที่ 2.55 ไดอะแกรมการทำงานของโมดูลแอลซีดีแบบอักษร	35
รูปที่ 2.56 รูปร่างและการจัดขาโมดูลแอลซีดีแบบอักษร	36
รูปที่ 3.1 บล็อกไดอะแกรมของวงจรภาคส่ง	42
รูปที่ 3.2 บล็อกไดอะแกรมของวงจรผลิตสัญญาณความถี่ 67.2 กิโลเฮิร์ต	42
รูปที่ 3.3 บล็อกไดอะแกรมวงจรกำเนิดสัญญาณ 4800 เฮิร์ต	43
รูปที่ 3.4 วงจรกำเนิดสัญญาณความถี่ 4800 เฮิร์ต	43

รูปที่ 3.5 บล็อกไดอะแกรมวงจรเฟสล็อกคูล	44
รูปที่ 3.6 วงจรสร้างความถี่ 67.2 กิโลเฮิร์ต	44
รูปที่ 3.7 วงจรเข้ารหัสสัญญาณแบบไม่กลับศูนย์แบบมาร์ค	45
รูปที่ 3.8 วงจรคูณสัญญาณพาหะกับสัญญาณข่าวสาร	46
รูปที่ 3.9 วงจรกรองความถี่แบนด์พาสฟิลเตอร์	46
รูปที่ 3.10 แสดงวงจรมอดูเลตสัญญาณพีเอสเคทั้งหมด	48
รูปที่ 3.11 แสดงหลักการทำงานของวงจรมอดูเลตความถี่	49
รูปที่ 3.12 แสดงรูปวงจรเครื่องส่งเอฟเอ็ม	49
รูปที่ 3.13 แสดงบล็อกไดอะแกรมของวงจรภากรับ	50
รูปที่ 3.14 แสดงบล็อกไดอะแกรมวงจรกู้ความถี่คลื่นพาห้	51
รูปที่ 3.15 แสดงวงจรยกกำลังสองสัญญาณ	51
รูปที่ 3.16 แสดงวงจรเฟสล็อกคูลสำหรับลูบยกกำลังสอง	52
รูปที่ 3.17 แสดงวงจรหารสอง	52
รูปที่ 3.18 แสดงวงจรกู้สัญญาณคลื่นพาห้แบบลูบยกกำลังสอง	53
รูปที่ 3.19 แสดงวงจรคูณสัญญาณคลื่นพาห้กับสัญญาณพีเอสเค	53
รูปที่ 3.20 แสดงหลักการทำงานพื้นฐานของวงจรกู้สัญญาณนาฬิกาจากสัญญาณเอ็นอาร์แซด	54
รูปที่ 3.21 วงจรหาการเปลี่ยนแปลงของขอบโดยอาศัยการหน่วงเวลา	55
รูปที่ 3.22 แสดงหลักการทำงานของวงจรกู้สัญญาณนาฬิกา	55
รูปที่ 3.23 แสดงวงจรกู้สัญญาณนาฬิกาที่ใช้ในทางปฏิบัติ	56
รูปที่ 3.24 แสดงวงจรถอดรหัสสัญญาณเอ็นอาร์แซดไอเป็นเอ็นอาร์แซด	56
รูปที่ 3.25 แสดงวงจรดีมอดูเลตพีเอสเคทั้งหมด	58
รูปที่ 3.26 แสดงหลักการทำงานของวงจรดีมอดูเลตความถี่	58
รูปที่ 3.27 แสดงวงจรเครื่องรับเอฟเอ็ม	59
รูปที่ 3.28 แสดงบล็อกไดอะแกรมของชุดแสดงผล	60
รูปที่ 3.29 แสดงวงจรชุดแสดงผล	61
รูปที่ 3.30 แสดงฟลอว์ชาร์ตการเขียนโปรแกรมชุดแสดงผล	64
รูปที่ 4.1 แสดงผลของวงจรเข้ารหัสสัญญาณ โดยรูปกลางคือสัญญาณจากวงจรบีททดสอบ และรูปล่างคือเอาต์พุตจากวงจรเข้ารหัสสัญญาณเอ็นอาร์แซดเป็นเอ็นอาร์แซดไอ เทียบกับสัญญาณนาฬิกา	65
รูปที่ 4.2 สัญญาณคลื่นพาห้ความถี่ 67.2 กิโลเฮิร์ต (รูปบน) และสเปกตรัม (รูปล่าง)	66
รูปที่ 4.3 แสดงข้อมูลที่มอดูเลตกับคลื่นพาห้ความถี่ 67.2 กิโลเฮิร์ตซ์ (รูปล่าง) แบบบีพีเอสเค (BPSK) เปรียบเทียบกับคลื่นพาห้และข้อมูล	66
รูปที่ 4.4 แสดงสัญญาณบีพีเอสเค (รูปบน) เปรียบเทียบกับสัญญาณข้อมูล	67
รูปที่ 4.5 แสดงสัญญาณบีพีเอสเค (BPSK) (รูปบน) และสเปกตรัมของสัญญาณก่อนผ่านวงจรกรองความถี่ผ่าน	67

รูปที่ 4.6 สัญญาณบีทีเอสเคและสเปกตรัมของสัญญาณเมื่อผ่านวงจรกรองความถี่ผ่าน ความถี่ ศูนย์กลาง 67.2 กิโลเฮิร์ตซ์	68
รูปที่ 4.7 แสดงคุณลักษณะของวงจรกรองความถี่ผ่าน	68
รูปที่ 4.8 แสดงสัญญาณคลื่นพาห้ความถี่ 67.2 กิโลเฮิร์ตซ์ที่ถูกกลับคืนมาได้ทางด้านรับ	69
รูปที่ 4.9 แสดงเอาต์พุตของวงจรกรองความถี่ผ่านเพื่อให้ได้คลื่นพาห้เป็นรูปสัญญาณไซน์ (รูปล่าง)	69
รูปที่ 4.10 คลื่นพาห้ความถี่ 67.2 กิโลเฮิร์ตซ์ที่ถูกกลับคืนมาได้และสเปกตรัมของสัญญาณ	70
รูปที่ 4.11 สัญญาณพีเอสเคที่รับได้ทางด้านรับ (รูปบน) เทียบกับสัญญาณนาฬิกาที่ถูกกลับคืนมา ที่ด้านรับ	70
รูปที่ 4.12 ข้อมูลที่รับได้ที่ด้านรับเทียบกับสัญญาณนาฬิกา	71
รูปที่ 4.13 ข้อมูลที่รับได้ที่ด้านรับเมื่อผ่านวงจรกรองความถี่ผ่าน (รูปบน) เทียบกับสัญญาณนาฬิกา (รูปล่าง)	71
รูปที่ 4.14 สัญญาณนาฬิกาความถี่ 4.8 กิโลเฮิร์ตซ์ที่ถูกกลับคืนมาได้ทางด้านรับ (รูปบน) เทียบกับ สัญญาณนาฬิกาต้านเครื่องส่ง (รูปล่าง)	72
รูปที่ 4.15 สัญญาณข้อมูลหลังจากผ่านการถอดรหัสที่ด้านรับ (สัญญาณเอ็นอาร์แซด รูปล่าง) เปรียบเทียบกับสัญญาณเอ็นอาร์แซดไอที่รับได้ที่ด้านรับ	73
รูปที่ 4.16 สัญญาณนาฬิกาที่กู้ได้ที่ด้านรับ (รูปล่าง) เปรียบเทียบกับสัญญาณข้อมูลที่ถอดรหัสได้ (รูปบน)	73
รูปที่ 4.17 สัญญาณเอ็นอาร์แซดไอ (รูปกลาง) และสัญญาณเอ็นอาร์แซด (รูปล่าง) เทียบกับสัญญาณ นาฬิกา (รูปบน) ที่ด้านรับ	74
รูปที่ 4.18 ข้อมูลด้านส่ง (รูปที่ 3) และสัญญาณที่รับได้ที่ด้านรับ (รูปล่าง) เทียบกับสัญญาณนาฬิกา และสัญญาณ NRZ-I	74
รูปที่ 4.19 แสดงหน้าจอคอมพิวเตอร์เมื่อเริ่มสคริปต์โปรแกรม	75
รูปที่ 4.20 แสดงการเชื่อมต่อพอร์ท (Com port)	75
รูปที่ 4.21 แสดงการเซตบอว์ด์เรท (Baud rate)	76
รูปที่ 4.22 แสดงการเซตเพื่อทำการออนไลน์ (Online)	76
รูปที่ 4.23 แสดงข้อความที่ต้องการส่งที่ปรากฏบนโปรแกรม	77
รูปที่ 4.24 แสดงหน้าจอของโปรแกรมเมื่อทำการส่งข้อความ	77
รูปที่ 4.25 แสดงสัญญาณคลื่นพาห้ของวงจรเครื่องส่งเอฟเอ็มและสเปกตรัมของสัญญาณ	78
รูปที่ 4.26 แสดงสัญญาณเอาต์พุตของวงจรเครื่องส่งเอฟเอ็มเมื่อป้อนสัญญาณเบสแบนด์ความถี่ 1 กิโลเฮิร์ตซ์ และสเปกตรัมของสัญญาณ	78
รูปที่ 4.27 แสดงสัญญาณเอาต์พุตของวงจรเครื่องส่งเอฟเอ็มเมื่อป้อนสัญญาณเบสแบนด์ความถี่ 67.2 กิโลเฮิร์ตซ์ และสเปกตรัมของสัญญาณ	79

รูปที่ 4.28 รูปแสดงชิ้นงานเครื่องส่งเอฟเอ็ม เอสซีเอ เพจเจอร์	79
รูปที่ 4.29 รูปแสดงภายในชิ้นงานเครื่องส่ง	80
รูปที่ 4.30 รูปแสดงชิ้นงานเครื่องรับเอฟเอ็ม เอสซีเอ เพจเจอร์	80
รูปที่ 4.31 รูปแสดงภายในชิ้นงานเครื่องรับ	81
รูปที่ 4.32 แสดงการต่อรวมชิ้นงานทั้งหมดประกอบด้วยเครื่องรับ, เครื่องส่ง, และการส่งข้อความจาก คอมพิวเตอร์	81
ตาราง ก) คุณสมบัติของเฟสดีเทกเตอร์แบบ I และแบบ II	33
ตาราง ข) รายการของโวลเตจคอนโทรลอสซิลเลเตอร์ที่ใช้ในปัจจุบัน	33



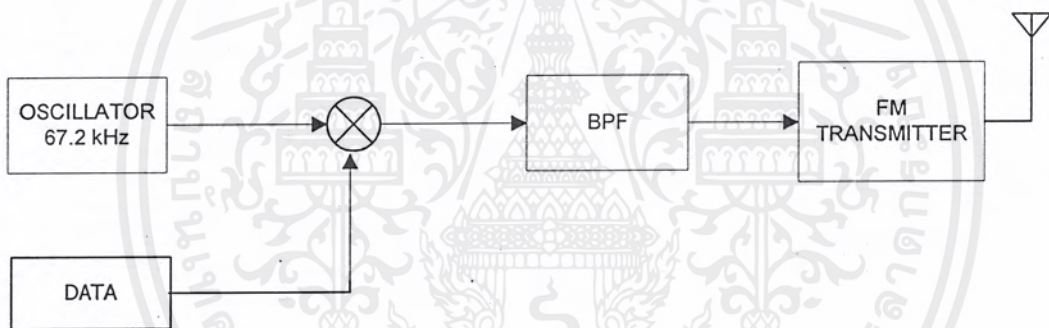
บทที่ 1

บทนำ

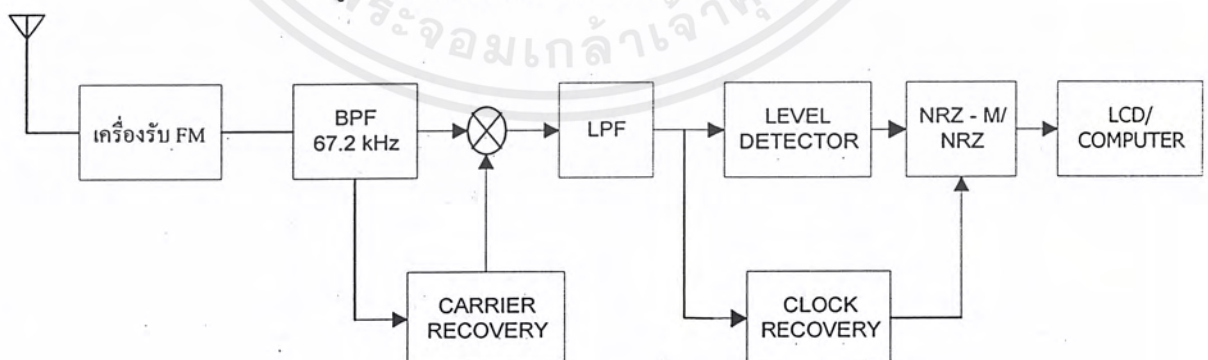
วิทยุติดตามตัวหรือเพจเจอร์เป็นอุปกรณ์สื่อสารแบบพกพาชนิดหนึ่ง โดยรูปแบบของการสื่อสารข้อมูลของวิทยุติดตามตัวจะเป็นการฝากส่งข่าวสารไว้ที่ศูนย์รับฝากข้อความ โดยที่ศูนย์รับฝากข้อความจะทำหน้าที่ส่งข่าวสารนั้นไปยังลูกข่ายที่ถูกระบุหมายเลขไว้

การส่งข้อความไปยังตัววิทยุติดตามตัวมีเทคนิคการส่งข้อความได้หลายวิธี ในโครงการนี้การส่งสัญญาณจะใช้การส่งสัญญาณไปยังช่องสัญญาณเอฟเอ็ม เอสซีเอโดยการนำข้อความต่างๆ มาทำการมอดูเลทกับคลื่นพาห์ (carrier) ที่ความถี่ 67.2 กิโลเฮิร์ต แล้วส่งออกอากาศด้วยความถี่คลื่นพาห์ของสัญญาณเอฟเอ็ม

ส่วนทางด้านเครื่องรับ จะเป็นเครื่องรับเอฟเอ็ม โดยจะแยกสัญญาณเอสซีเอออกมา เพื่อนำมาถอดรหัสเอาข้อมูลข่าวสาร และนำข้อมูลออกมาแสดงผลบนจอแอลซีดี (LCD) หรือเครื่องคอมพิวเตอร์



รูปที่ 1.1 บล็อกไดอะแกรมของวงจรภาคส่ง



รูปที่ 1.2 บล็อกไดอะแกรมของวงจรภาครับ

บทที่ 2

ทฤษฎีและหลักการ

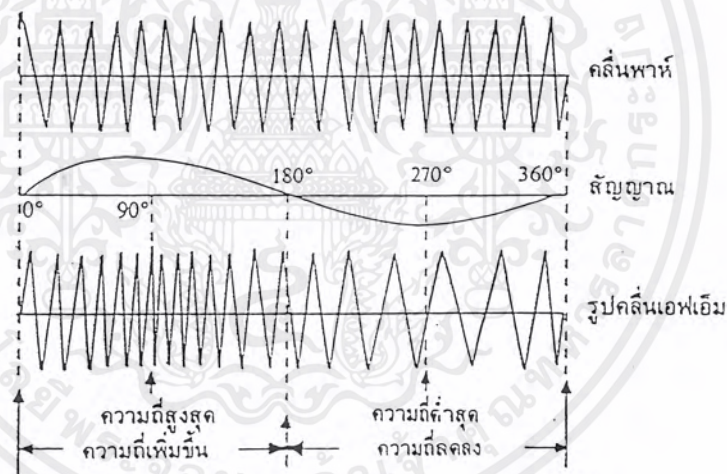
2.1 การมอดูเลชันเชิงความถี่ (Frequency modulation)

ในการมอดูเลชันเชิงขนาด (A.M.) มีจุดอ่อนอยู่ที่สัญญาณรบกวน (noise) ซึ่งไม่สามารถขจัดออกไปได้ แต่ในกรรมวิธีการมอดูเลชันเชิงความถี่ (F.M.) สามารถแก้ปัญหานี้ได้อย่างน้อยก็สามารถลดระดับการรบกวนให้น้อยลง จึงนิยมใช้กันแพร่หลายทั้งในระบบวิทยุสื่อสารและวิทยุกระจายเสียง

2.1.1 ลักษณะของคลื่นเอฟเอ็ม

การมอดูเลชันเชิงความถี่ เป็นกรรมวิธีรวมหรือฝากสัญญาณเข้ากับคลื่นพาห์อีกวิธีหนึ่ง ในลักษณะที่ความถี่ของคลื่นพาห์จะเปลี่ยนไปตามลักษณะของสัญญาณ ทั้งนี้ขนาดของคลื่นพาห์ยังคงที่ตลอดเวลา

รูปที่ 2.1 แสดงลักษณะของคลื่นพาห์ก่อนและหลังถูกมอดูเลตด้วยสัญญาณที่เป็นคลื่นรูปไซน์ รูปที่แสดงนี้เป็นเพียงตัวอย่างให้เข้าใจง่ายในทางปฏิบัติความถี่ของคลื่นพาห์กับสัญญาณจะต่างกันมากกว่าที่แสดงไว้ รวมทั้งสัญญาณปกคิจะเป็นคลื่นเชิงซ้อน แต่ผลที่ได้ก็เหมือนกับสัญญาณรูปไซน์



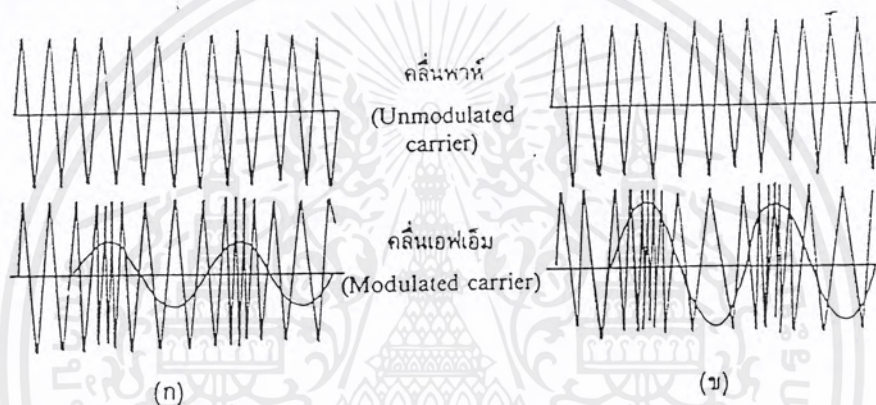
รูปที่ 2.1 การแปรผันทางความถี่ของคลื่นพาห์โดยสัญญาณ

ในช่วงมอดูเลชัน ณ เวลาใดๆก็ตาม ความถี่ของคลื่นพาห์จะขึ้นโดยตรงกับขนาดของสัญญาณในช่วงเริ่มต้นวัฏจักรของสัญญาณขนาดหรือโวลเตจจะต่ำ เป็นผลให้คลื่นพาห์มีความถี่เปลี่ยนแปลงเพียงเล็กน้อย ขณะที่สัญญาณมีขนาดเข้าใกล้ช่วงสุดท้ายทางบวก (positive peak) ความถี่ของคลื่นจะเปลี่ยนแปลงอย่างมากและจะมากที่สุด ณ จุด 90° ในวัฏจักรของสัญญาณในช่วงระหว่าง 90° และ 180° ขนาดของสัญญาณกลับคืนสู่ 0 โวลท์ ช่วงนี้ความถี่ของคลื่นพาห์ก็จะเลื่อนกลับเข้าสู่ค่าความถี่เดิม เหมือนกับตอนที่ไม่มีสัญญาณเข้ามามอดูเลต ถัดมาสัญญาณเข้าสู่ช่วง 270° ขนาดจะเริ่มสูงขึ้นอีกครั้ง แต่จะไปในทิศทางลบซึ่งหมายถึงความถี่ต่ำลงและต่ำสุด ณ จุด 270°

ในทางปฏิบัติจะไม่คำนึงว่า ในเครื่องจักรบวทของคลื่นรูปไซน์ของสัญญาณจะเป็นผลทำให้ความถี่ของคลื่นพาห์เพิ่มขึ้นหรือลดลง จุดสำคัญอยู่ที่ว่า ความถี่ของคลื่นพาห์จะเปลี่ยนเป็นสัดส่วนกับการเปลี่ยนแปลงทางขนาดของสัญญาณ

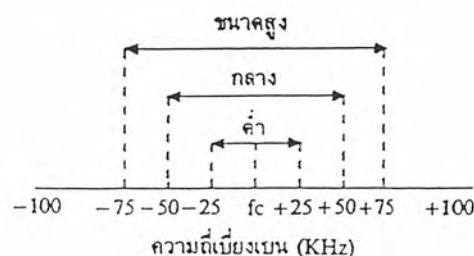
2.1.2 ผลจากการเปลี่ยนแปลงขนาดของสัญญาณ

ในคลื่นเอฟเอ็มนั้น ความถี่ของคลื่นพาห์ที่เปลี่ยนแปลงไปจากความถี่เดิม หรือความถี่ศูนย์กลางไปมากน้อยเท่าไร เรียกว่าความถี่เบี่ยงเบน (frequency deviation) โดยถูกกำหนดขนาดสูงสุดของสัญญาณ ถ้าขนาดสูงสุดเปลี่ยนแปลง ค่าความถี่ก็จะเปลี่ยนตามไปด้วย รูปที่ 2.2 แสดงถึงผลที่เกิดในคลื่นเอฟเอ็มเมื่อขนาดของสัญญาณสูงขึ้น แต่ความถี่ของมันยังคงที่อยู่ เพื่อเป็นการเปรียบเทียบจึงใช้รูปคลื่นของสัญญาณซ้อนอยู่บนรูปคลื่นเอฟเอ็ม ให้สังเกตความถี่เบี่ยงเบนเพิ่มขึ้นเมื่อสัญญาณมีขนาดเพิ่มขึ้นในรูปที่ 2.2 (ข) (เกิดขึ้นที่ขนาดสูงสุดของสัญญาณ)



รูปที่ 2.2 รูปคลื่นเอฟเอ็มที่เกิดจากสัญญาณมีขนาดต่างกัน

ตัวอย่างเช่น เมื่อคลื่นพาห์ถูกมอดูเลตด้วยสัญญาณความถี่ 3,000 Hz ที่มีขนาดต่ำๆ ความถี่เบี่ยงเบนของคลื่นเอฟเอ็มเป็น ± 25 kHz เมื่อขนาดของสัญญาณสูงขึ้นสองเท่า ความถี่เบี่ยงเบนจะเพิ่มขึ้นสองเท่าด้วยเป็น ± 50 kHz และเมื่อขนาดสูงขึ้นอีก ความถี่เบี่ยงเบนก็จะเพิ่มขึ้นอีกเป็น ± 75 kHz ทั้งหมดนี้ความถี่จะแกว่ง (frequency swing) ไป 50 kHz , 100 kHz และ 150 kHz ตามลำดับดังแสดงในรูปที่ 2.3

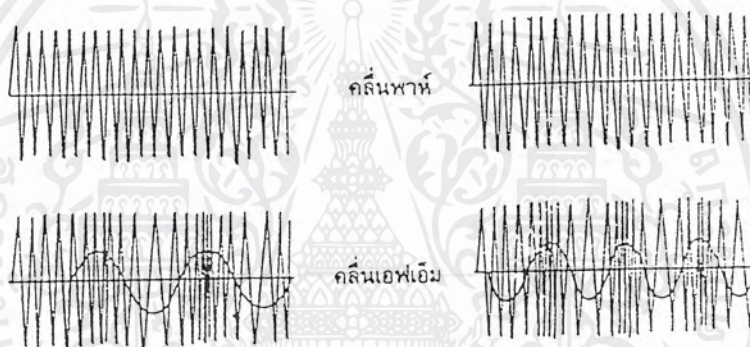


รูปที่ 2.3 การเปลี่ยนแปลงความถี่เบี่ยงเบนโดยขนาดของสัญญาณ

2.1.3 ผลจากการเปลี่ยนแปลงความถี่ของสัญญาณ

ความถี่ของสัญญาณเป็นตัวกำหนดอัตราที่ความถี่ของคลื่นพาห์เปลี่ยนแปลงไป ถ้าความถี่ของสัญญาณเพิ่มขึ้น อัตราการเปลี่ยนแปลงความถี่ของคลื่นพาห์ก็สูงตามไปด้วย รูปที่ 2.4 นี้ แสดงผลที่มีต่อคลื่นเอฟเอ็มเมื่อความถี่ของสัญญาณเพิ่มขึ้น แต่ขนาดสูงสุดยังคงที่อยู่ และเพื่อเป็นการเปรียบเทียบจึงแสดงรูปคลื่นของสัญญาณให้ซ้อนกับรูปคลื่นเอฟเอ็มในรูปที่ 2.4 (ข) อัตราการเปลี่ยนแปลงทางความถี่ของคลื่นเอฟเอ็มจะเร็วกว่าที่เกิดขึ้นในรูป (ก)

ตัวอย่างเช่นเมื่อคลื่นพาห์ถูกมอดูเลตด้วยสัญญาณความถี่ 1,000 Hz มีขนาดทำให้เกิดความถี่เบี่ยงเบนขนาดหนึ่ง สมมติ ± 25 kHz ด้วยอัตรา 1,000 ครั้งต่อวินาที เมื่อความถี่ของสัญญาณเพิ่มขึ้นเป็น 2,000 Hz แต่ขนาดยังคงเดิมความถี่ของคลื่นพาห์จะเบี่ยงเบนเท่าเดิมคือ ± 25 kHz แต่คราวนี้อัตราการเปลี่ยนแปลงความถี่เพิ่มเป็น 2,000 ครั้งต่อวินาที



รูปที่ 2.4 รูปคลื่นเอฟเอ็มที่เกิดจากสัญญาณมีความถี่ต่างกัน

2.1.4 การวิเคราะห์สัญญาณเอฟเอ็ม

การวิเคราะห์ทางคณิตศาสตร์ที่สมบูรณ์ของการมอดูเลชันทางมุม จะต้องใช้คณิตศาสตร์ชั้นสูงในจุดประสงค์ของเราแล้ว มันง่ายเพียงพอที่จะให้คำตอบ และสามารถพิจารณาได้ง่ายสำหรับการมอดูเลชันทางเฟส (PM)

สมการของความต่างศักย์ชั่วขณะหาได้จาก

$$e = A \sin (\omega_c t + m_p \sin \omega_i t) \quad (1)$$

โดย e = ความต่างศักย์ชั่วขณะ

A = ค่าพีคของคลื่นพาห์เริ่มต้น

ω_c = ความเร็วเชิงมุมของพาห์ ($2\pi f_c$)

m_p = ค่าเฟสขีฟสูงสุดอันเนื่องจากสัญญาณมอดูเลชัน (radians)

ω_i = ความเร็วเชิงมุมของสัญญาณมอดูเลชัน (อินเทลิเจนซ์) ($2\pi f_i$)

ค่าเฟสขีฟสูงสุดอันเนื่องมาจากสัญญาณในการมอดูเลชัน (m_f) จะถูกนิยามเป็นเสมือนเป็นดัชนีการมอดูเลชันของ PM

สมการต่อไปนี้จะกำหนดสูตรสำหรับ FM

$$e = c \sin (\omega_c t + m_f \sin \omega_i t) \quad (2)$$

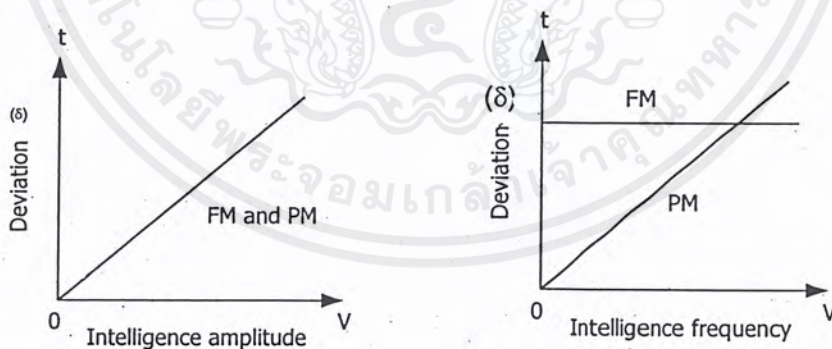
ในทุกเทอมของสมการ (2) กำหนดเช่นเดียวกับกับสมการ (1) ยกเว้นเทอมใหม่คือ m_f โดยจริงแล้ว สมการทั้งสองต่างก็เป็นเอกลักษณ์ซึ่งกัน เว้นแต่ในเทอม m_f ซึ่งเรียกว่า ดัชนีการมอดูเลชันของ FM จะมีค่าเท่ากับ

$$m_f = \text{FM modulation index} = \delta / f_i \quad (3)$$

โดย δ = ความถี่ขีฟสูงสุด (maximum frequency shift) อันเนื่องมาจากสัญญาณอินเทลิเจนซ์ (ค่าเบี่ยงเบน)

f_i = ความถี่ของสัญญาณอินเทลิเจนซ์ (สัญญาณมอดูเลชัน)

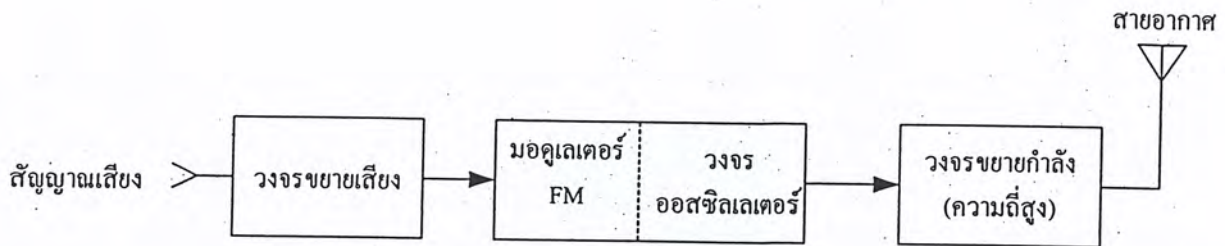
เมื่อเปรียบเทียบสมการ (1) และสมการ (2) แล้ว จะสังเกตเห็นจุดแตกต่างระหว่าง PM และ FM สมการของ PM แสดงถึงเฟสของพาหะเปลี่ยนแปลงกับแอมพลิจูดของสัญญาณมอดูเลชัน และใน FM เฟสพาหะจะหาได้โดยอัตราส่วนของแอมพลิจูดของสัญญาณอินเทลิเจนซ์ (δ) ต่อความถี่อินเทลิเจนซ์ (f_i) ด้วยเหตุนี้ FM จะไม่ไวกับความถี่สัญญาณมอดูเลชัน แต่ PM นั้นจะไวต่อความถี่สัญญาณมอดูเลชัน ความต่างกันของทั้งสองระบบนี้ยากที่จะกล่าวถึง โดยจริงแล้วหากสัญญาณอินเทลิเจนซ์ถูกรวมเข้า และทำให้เกิดการมอดูเลต เฟสกับพาหะ อันเป็นการสร้างสัญญาณ FM ขึ้น วิธีการนี้จะเป็นวิธีที่ใช้ในระบบ Armstrong indirect FM ใน FM นั้น จำนวนของค่าเบี่ยงเบนที่เกิดขึ้น ไม่ได้ขึ้นกับความถี่อินเทลิเจนซ์ เสมือนกันกับ PM จำนวนของค่าเบี่ยงเบนเป็นสัดส่วนกับแอมพลิจูดสัญญาณอินเทลิเจนซ์ทั้งในระบบ PM และ FM สภาพเหล่านี้ได้แสดงไว้ดังรูปต่อไปนี้



รูปที่ 2.5 ผลของค่าเบี่ยงเบนต่อ FM/PM เทียบกับพารามิเตอร์ของอินเทลิเจนซ์

2.2 เครื่องส่ง FM

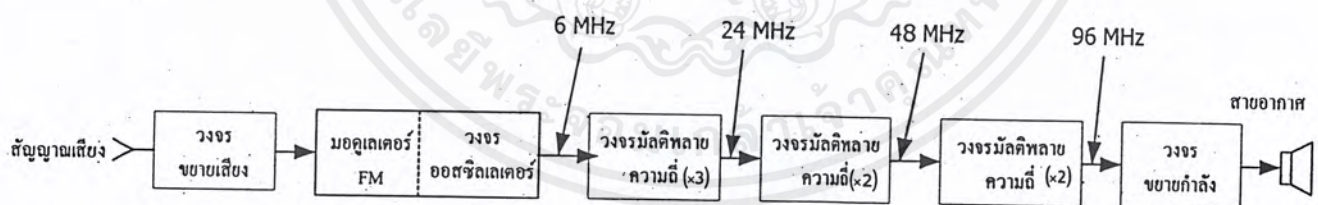
จากแผนผังของเครื่องส่ง FM (ในรูปที่ 2.6) สัญญาณเสียงผ่านการขยายแล้วป้อนสู่มอดูเลเตอร์ วงจรมอดูเลเตอร์นี้จะทำการเปลี่ยนความถี่ของออสซิลเลเตอร์ โดยมีช่วงความถี่เบี่ยงเบนและอัตราการเบี่ยงเบนขึ้นอยู่กับแอมพลิจูดของสัญญาณเสียงตามลำดับ พาหะ FM ที่ถูกมอดูเลตแล้วจะถูกขยายโดยภาควิทยุกำลังสุดท้ายป้อนสู่สายอากาศเพื่อส่งออกอากาศต่อไป



รูปที่ 2.6 แผนผังเครื่องส่ง FM อย่างง่าย

เครื่องส่งที่กล่าวมาข้างต้นอาจเกิดปัญหาเมื่อเราต้องการส่งออกอากาศที่ความถี่สูงๆ เช่น เครื่องส่งกระจายเสียง FM (ซึ่งมีความถี่อยู่ระหว่าง 88 ถึง 108 เมกะเฮิรตซ์) ทำงานที่ความถี่สูง ทำให้ยากต่อการควบคุมให้ความถี่คงที่ นอกจากนี้การควบคุมการเบี่ยงเบนความถี่ก็ทำได้ยากขึ้นด้วย วิธีแก้ปัญหาดังกล่าวสามารถทำได้หลายวิธีแตกต่างกันออกไป

ในรูปที่ 2.7 แสดงการใช้ความถี่ของออสซิลเลเตอร์ 8 เมกะเฮิรตซ์ และมัลติพลาย (หรือคูณ) ความถี่ขึ้นไปเป็น 96 เมกะเฮิรตซ์ การคูณความถี่นี้ทำได้โดยวงจรมัลติพลาย หลักการของวงจรมัลติพลายก็คือ ใช้คุณสมบัติ ความไม่ลิเนียร์ของวงจรขยาย ซึ่งจะทำให้เกิดสัญญาณฮาร์มอนิกออกมาจำนวนมากจากนั้นวงจรแพคเกจที่ output จะจับเอาเฉพาะความถี่ฮาร์มอนิกที่ต้องการไปใช้ประโยชน์ โดยทั่วไปวงจรมัลติพลายมักเป็นชนิดคูณ 2 (เรียก ดับเบิลเลอร์ หรือ doubler) หรือชนิดคูณสาม (เรียกทริปเปิลเลอร์ หรือ tripler) ในที่นี้เราใช้วงจรคูณ 3 จำนวน 1 วงจร และวงจรคูณ 2 จำนวน 2 วงจร นั่นคือ $3 \times 2 \times 2 = 12$ เท่า ฉะนั้นความถี่เอาต์พุตจะเป็น $8 \text{ เมกะเฮิรตซ์} \times 12 \text{ เท่า} = 96 \text{ เมกะเฮิรตซ์}$



รูปที่ 2.7 แผนผังเครื่องส่งกระจายเสียง FM แบบคูณความถี่ (มัลติพลาย)

ช่วงความถี่เบี่ยงเบนของสัญญาณวิทยุกระจายเสียง FM เท่ากับ $\pm 75 \text{ KHz}$ ฉะนั้น output จะต้องมีความถี่เบี่ยงเบนไปเท่ากับค่านี้นี้เมื่อสัญญาณเสียงมอดูเลต (แบบ FM) อย่างไรก็ตามการมัลติพลายความถี่จะทำให้ปริมาณความถี่เบี่ยงเบนถูกคูณให้กว้างขึ้นไปด้วย เช่น ออสซิลเลเตอร์ 8 MHz เบี่ยงเบนอยู่ระหว่าง 7.9 ถึง 8.1 MHz ($\pm 0.1 \text{ MHz}$) เมื่อคูณ 12 เท่า พหุจะมีค่าถี่กลางเป็น 96 MHz และเบี่ยงเบนอยู่ระหว่าง 94.8 ถึง 97.2 MHz ($\pm 12 \text{ MHz}$) ดังนั้นถ้าหากเราต้องการให้ความถี่เบี่ยงเบนเป็น $\pm 75 \text{ MHz}$ ที่ output ความถี่ออสซิลเลเตอร์จะเบี่ยงเบนไปเท่ากับ $\pm 75 \div 12 = \pm 6.25 \text{ KHz}$

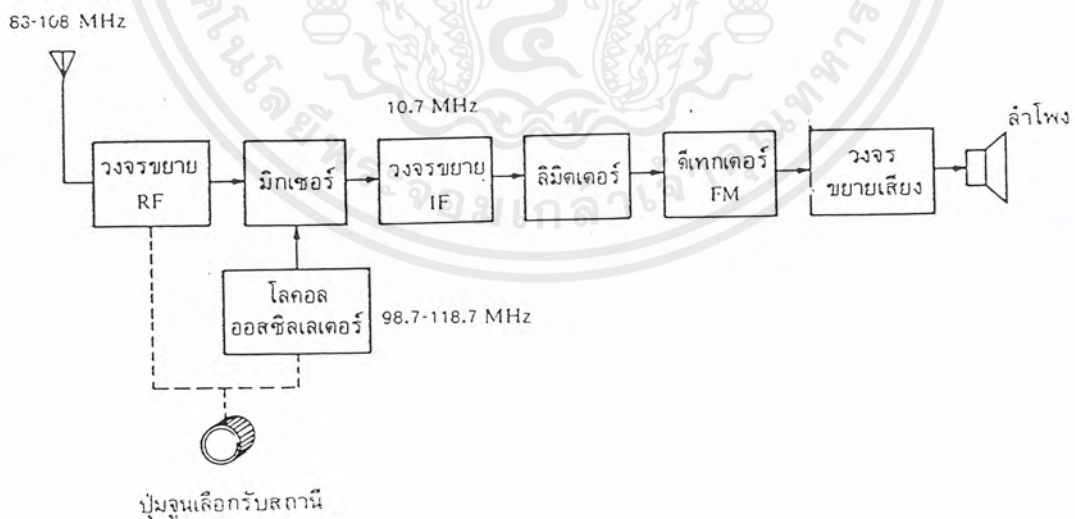
ข้อดีอีกประการหนึ่งของระบบ FM ก็คือวงจรขยายกำลัง (power amplifier หรือ PA) สามารถทำงานในคลาสร C ซึ่งมีประสิทธิภาพสูงกว่า ทั้งนี้เพราะแอมพลิจูดของสัญญาณ FM ไม่มีผลทำให้ข่าวสารเพี้ยน (แม้จะเกิดการขลิบยอดสัญญาณ) ข่าวสารนั้นอยู่ในช่วงความเปลี่ยนแปลงความถี่ของสัญญาณ FM เท่านั้น

2.3 เครื่องรับ FM

แผนผังของเครื่องรับ FM มีความคล้ายคลึงกับเครื่องรับ AM อย่างมาก จะแตกต่างกันก็แต่เฉพาะขบวนการดีเทกเท่านั้น สำหรับความถี่ IF มักจะใช้ 10.7 MHz เพื่อกำจัดสัญญาณเงาและเพื่อให้แบนด์วิดท์ของเครื่องรับต้องมีค่า 150 KHz เป็นอย่างน้อยปกติมักจะเผื่อให้กว้างอีกเล็กน้อยเป็น 180 ถึง 200 KHz

สมมติว่าเราจูนเครื่องรับไว้ที่ 100 MHz ลูกบิดหน้าปัด จะเลื่อนไปตรงกับความถี่ 100 MHz (บนหน้าปัด) วงจรขยาย RF จะจูนไว้ที่ความถี่ 100 MHz ส่วนโลคอลออสซิลเลเตอร์จะจูนไว้ที่ความถี่ 110.7 MHz เมื่อผ่านกรรมวิธีเฮตเทอโรไดน์ในวงจรมิกเซอร์ ผลต่างความถี่จะปรากฏที่ input ของวงจรขยาย IF เท่ากับ $110.7 \text{ MHz} - 100 \text{ MHz} = 10.7 \text{ MHz}$ สัญญาณที่ความถี่ IF นี้จะถูกขยายและกำจัดแบนด์วิดท์ให้กว้างเพียงพอที่จะรับสัญญาณ FM และแคบเพียงพอที่จะกำจัดสัญญาณที่ไม่ต้องการอื่นๆ ทิ้งไป

ถ้าพาหะ FM ที่ส่งจากเครื่องส่งมีความถี่เบี่ยงเบนเท่ากับ $\pm 50 \text{ KHz}$ (โดยความถี่ FM เท่ากับ 100 MHz คงเดิม โลคอลออสซิลเลเตอร์คงเดิม และ IF คงเดิม) สัญญาณ IF จะมีความถี่เบี่ยงเบนเท่ากับ $\pm 50 \text{ KHz}$ ด้วย ฉะนั้นสัญญาณที่มอดูเลตมาบนพาหะจะยังอยู่ในสัญญาณ IF โดยไม่มีความเพี้ยน แม้ว่าความถี่สัญญาณ FM จะลดทอนจาก 100 MHz เหลือ 10.7 MHz



รูปที่ 2.8 แผนผังของเครื่องรับ FM

2.4 เอฟเอ็ม เอสซีเอ (FM SCA)

SCA (Subsidiary Communications Authorization) หมายถึง การส่งกระจายในระบบ FM ซึ่งสอดแทรกข่าวสาร หรือ รายการที่นอกเหนือไปจากรายการปกติอีกหนึ่ง หรือหลายรายการควบคู่ไปพร้อมกับการส่งกระจายเสียงของระบบ FM โดยไม่ก่อให้เกิดการรบกวนระหว่างข่าวสาร หรือรายการซึ่งกันและกัน รายการของ SCA อาจเป็นสัญญาณเสียงพูดหรือดนตรี เช่นเดียวกับ รายการปกติหรืออาจเป็นรหัสตัวเลขหรือสัญญาณอื่นใดก็ได้

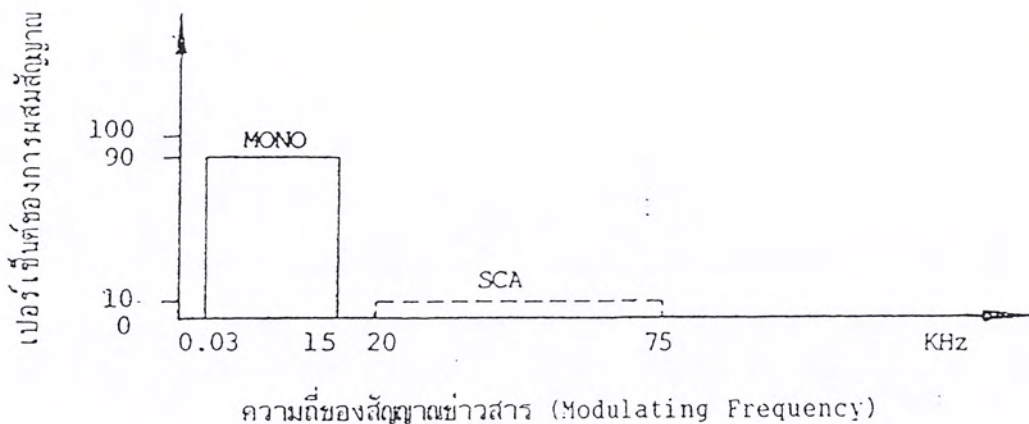
- การส่งกระจายเสียงในระบบ FM mono จะใช้ความกว้างแถบคลื่นประมาณ 108 KHz
- การส่งกระจายเสียงในระบบ FM stereo multiplex จะใช้ความกว้างแถบคลื่น ประมาณ 256 KHz
- การส่งกระจายเสียงในระบบ FM stereo multiplex รวมกับ SCA จะใช้ความกว้างแถบคลื่น ประมาณ 300 KHz

สำหรับสัญญาณ SCA ที่จะนำไปผสมกับสัญญาณในระบบ FM นั้น จะต้องมีความ amplitude และ bandwidth อยู่ในขอบเขตจำกัด เพื่อป้องกันมิให้เกิดการรบกวนกับรายการ FM ในภาคปกติและสัญญาณของสถานีที่ใช้ความถี่ข้างเคียง การรับฟังรายการของ SCA เครื่องรับวิทยุ จะต้องมีความเลือกสัญญาณ SCA ด้วย จึงสามารถรับฟังรายการ SCA ได้เช่นเดียวกับเครื่องรับวิทยุระบบ FM mono ที่ไม่สามารถแยกสัญญาณในระบบ stereo multiplex ได้เนื่องจาก ไม่มีอุปกรณ์ในการแยกสัญญาณ stereo multiplex

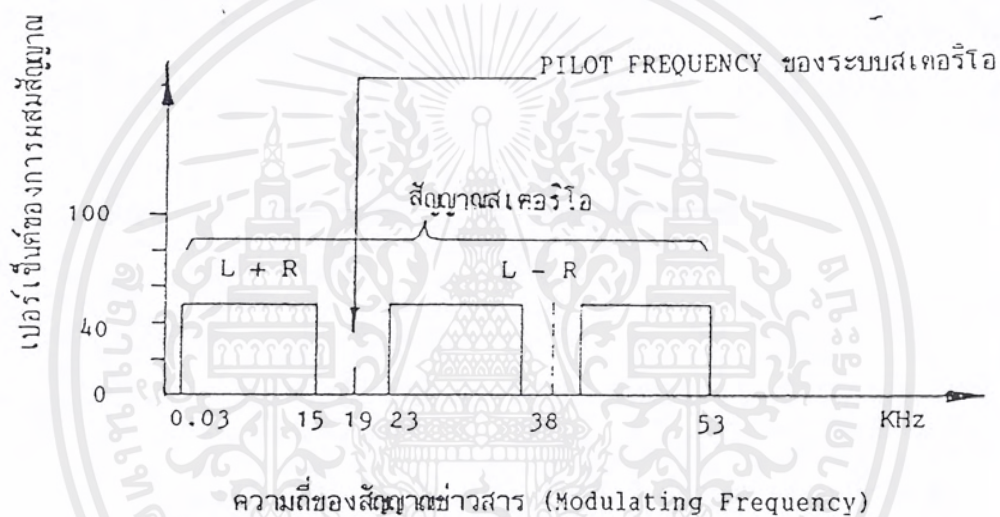
ในรูปที่ 2.9, 2.10, 2.11 และ 2.12 แสดงถึงความแตกต่างของแถบความถี่ที่ใช้ส่งกระจายเสียงในระบบ FM mono , FM stereo multiplex , FM mono บวกกับ SCA และ FM stereo multiplex บวกกับ SCA



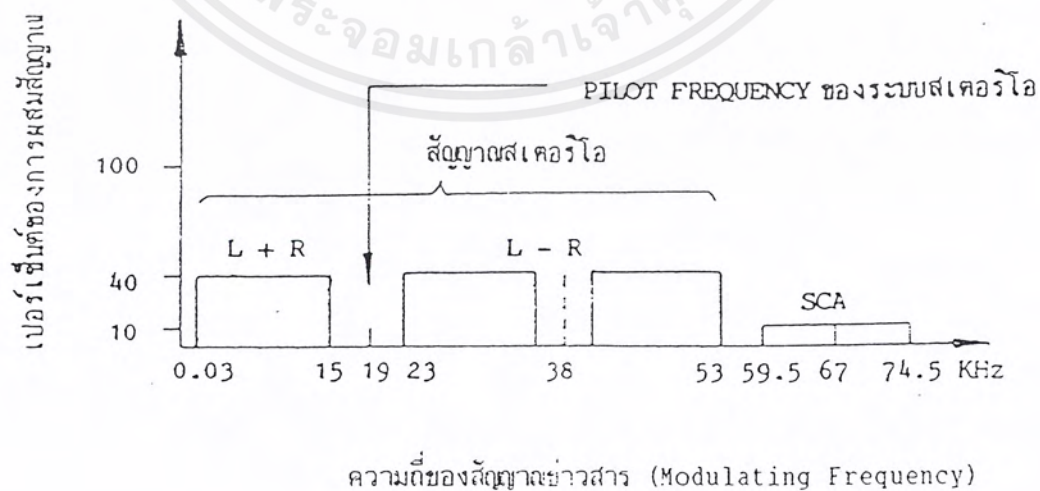
รูปที่ 2.9 แสดงการใช้แถบความถี่ของสัญญาณในระบบ FM mono



รูปที่ 2.10 แสดงการใช้แถบความถี่ของสัญญาณในระบบ mono รวมกับ SCA



รูปที่ 2.11 แสดงการใช้แถบความถี่ของสัญญาณในระบบ FM stereo



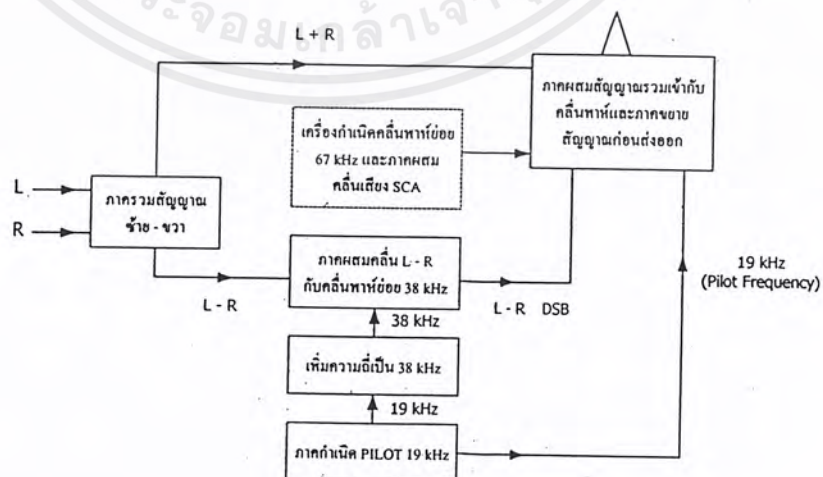
รูปที่ 2.12 แสดงการใช้แถบความถี่ของสัญญาณในระบบ FM stereo รวมกับ SCA

ในรูปที่ 2.9 จะเห็นได้ว่า ระบบ FM mono ความถี่ของสัญญาณข่าวสาร (modulating frequency) มีความกว้างของแถบคลื่นไม่เกิน 15 KHz รูปที่ 2.10 แสดงระบบ FM stereo ซึ่งจะใช้ความกว้างของแถบความถี่เพิ่มขึ้นเป็น 53 KHz โดยมีคลื่นพาห่อย่อย (sub-carrier) ความถี่ 38 KHz เป็นตัวทำให้เกิดการแยกสัญญาณออกเป็นข้างซ้ายและขวา โดยคลื่นพาห่อย่อยนี้ ได้มาจากความถี่ 19 KHz ซึ่งกำหนดขึ้นภายในเครื่องส่งความถี่ 19 KHz นี้ ยังใช้เป็นความถี่นำ (pilot frequency) เพื่อให้ทางเครื่องรับทราบว่าเป็นระบบ stereo multiplex และใช้ในการแยกสัญญาณข้างซ้ายและสัญญาณข้างขวาในเครื่องรับอีกครั้งหนึ่ง

2.4.1 การส่งและรับสัญญาณระบบ SCA

2.4.1.1 การส่ง การส่งวิทยุกระจายเสียงระบบ FM มีอยู่ 2 แบบ คือ monophonic และ stereo - phonic สำหรับสัญญาณของ SCA นั้น สามารถสอดแทรกได้ทั้งสองระบบ แต่จะแตกต่างกันตรงขนาดความถี่ของคลื่นพาห่นั้น ในระบบ mono สามารถสอดแทรกจำนวนคลื่นพาห่อย่อยของ SCA ได้ มากกว่ากรณีของ ระบบ stereo ในระบบ FM stereo โดยปกติ จะสอดแทรกคลื่นพาห่อย่อยของ SCA ได้เพียงหนึ่งหรือสองช่องสัญญาณเท่านั้น หากเกินกว่านี้ จะทำให้ความกว้างของแถบความถี่ของคลื่นวิทยุเกินไป อาจก่อให้เกิดการรบกวนต่อช่องสัญญาณของสถานีที่ใช้ความถี่วิทยุข้างเคียงได้ เมื่อได้ทำการสอดแทรกสัญญาณ SCA ร่วมไปกับสัญญาณกระจายเสียงหลักนั้นจะต้องจำกัดขนาดและความแรงของสัญญาณทั้งสัญญาณกระจายเสียงหลัก และสัญญาณ SCA ด้วย เพื่อป้องกันไม่ให้เกิดการรบกวนซึ่งกันและกัน

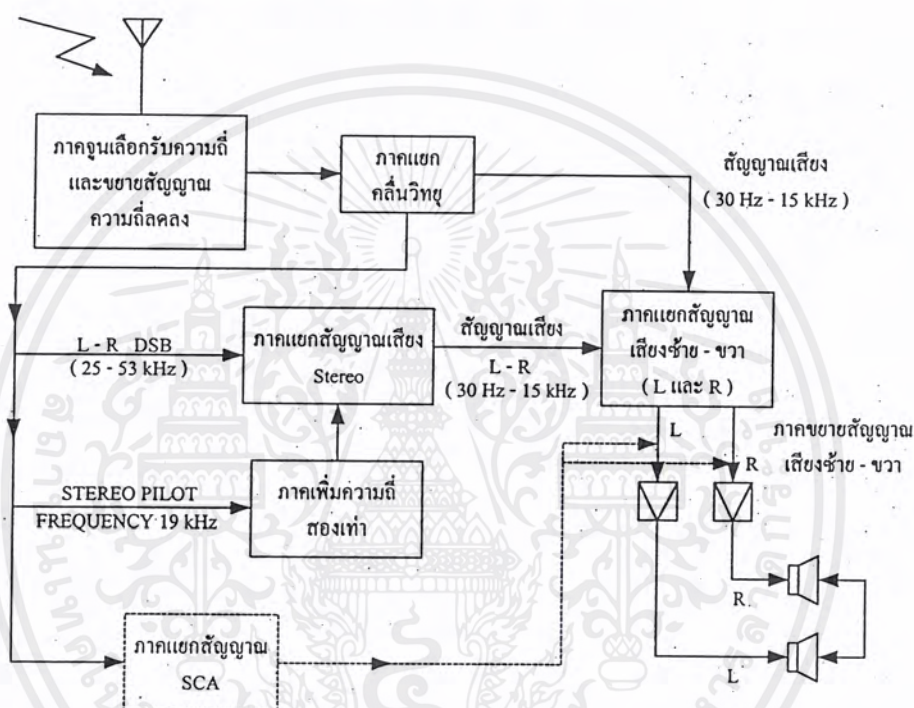
ในการส่งกระจายเสียงระบบ FM แบบ monophonic , stereo phonic , mono phonic + SCA และ stereo phonic + SCA จะใช้อุปกรณ์ในการส่งที่แตกต่างกัน โดยระบบที่มี SCA รวมอยู่ด้วย จะมีอุปกรณ์ SCA generator เพิ่มเข้ามาให้เห็นใน Block Diagram ของระบบ stereo phonic + SCA ดังรูปที่ 2.13 Block ที่แสดงด้วยเส้นประ แสดงถึงส่วนของอุปกรณ์ที่เพิ่มเข้ามาจากระบบการส่ง FM stereo ปกติ รายการที่ส่งโดยระบบ SCA จะถูกผสมคลื่นด้วยคลื่นพาห่อย่อยขนาดความถี่ 67 KHz แล้วส่งไปพร้อมกับสัญญาณ stereo ซึ่งเป็นรายการกระจายเสียง FM เข้าไปผสมคลื่นพาห่หลัก (main carrier) ในเครื่องส่ง แล้วจึงแพร่กระจายคลื่นออกทางสายอากาศ



รูปที่ 2.13 แสดง block diagram โดยสังเขป ของเครื่องส่งวิทยุกระจายเสียงระบบ stereo phonic + SCA

2.4.1.2 การรับ กรณีการรับสัญญาณของสถานีวิทยุกระจายเสียงระบบ FM ก็เช่นเดียวกัน จะมี การรับอยู่สองแบบ คือ monophonic และ stereo phonic หากเครื่องรับวิทยุต้องการรับฟังสัญญาณเสียงใน ระบบ SCA ให้ได้นั้นจะต้องมีอุปกรณ์แยกสัญญาณ SCA จึงจะสามารถรับฟังรายการหรือข่าวสารของ ระบบ SCA ได้ ดังแสดงให้เห็นด้วย Block Diagram ของเครื่องรับวิทยุซึ่งมีอุปกรณ์แยกสัญญาณ SCA ใน รูปที่ 2.14

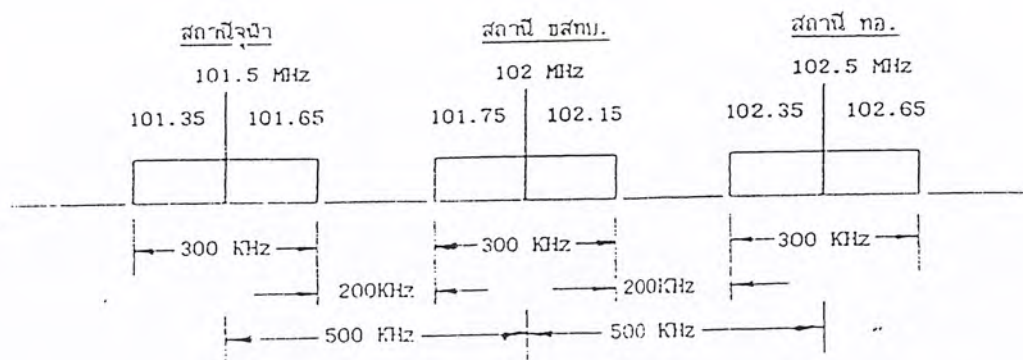
ในเครื่องรับระบบ SCA สัญญาณ SCA ซึ่งรวมกับสัญญาณ stereo จะถูกแยกโดยภาคแยก สัญญาณ SCA ดังแสดงใน Block เส้นประ ภาคนี้จะแยกคลื่นพาห่อย่อย (67 KHz) ออกให้เหลือเพียง สัญญาณข่าวสารในรายการ SCA แล้วขยายออกที่ลำโพงของเครื่องรับ



รูปที่ 2.14 แสดง Block Diagram โดยสังเขปของเครื่องรับวิทยุ FM stereo ที่สามารถเลือกรับ สัญญาณ SCA ได้ด้วย

2.4.2 การพิจารณาเกี่ยวกับการรบกวนในกรณีสถานีวิทยุ FM stereo ที่ส่ง SCA ด้วยการรบกวน ข้างเคียง

สถานีวิทยุกระจายเสียงระบบ FM ในกรุงเทพมหานคร (39 สถานี) ทุกสถานีจะมีความถี่ข้าง เคียงห่างกัน 500 KHz สถานี FM stereo ที่ส่งรายการ SCA ด้วย จะมีความกว้างแถบความถี่หรือ Bandwidth ประมาณ 300 KHz ($2 \times 75 + 2 \times 74.5 = 299$ KHz) และสถานีซึ่งออกอากาศในระบบ FM stereo ที่ไม่มีการส่ง SCA จะใช้ความแถบความถี่ประมาณ 256 KHz ($2 \times 75 + 2 \times 53 = 256$ KHz) จากตัว เลขดังกล่าวจะเห็นว่า จะไม่เกิดการรบกวนกันระหว่างสถานีที่อยู่ข้างเคียง ตัวอย่างเช่น สมมติว่า สถานี วิทยุ ขสทบ. , สถานีวิทยุ ทอ. ส่งคลื่นระบบ SCA อาจใช้การพิจารณาข้างต้น แสดงความกว้างของแถบ ความถี่ของคลื่นทั้ง 3 สถานีขณะส่ง SCA ด้วยว่าจะไม่เกิดการรบกวนกัน ดังแสดงในรูปที่ 7



รูปที่ 2.15 แสดงความกว้างแถบความถี่ของคลื่นสัญญาณ สถานีวิทยุจุฬา, ขสททบ. และ ทอ.

ที่ส่งระบบ SCA ด้วยทั้ง 3 สถานี

ในกรณีของสถานีวิทยุในต่างจังหวัด ปัญหาการรบกวนระหว่างสถานีที่ใช้ความถี่ข้างเคียงยังไม่ต้องพิจารณาถึงเลย เนื่องจากในแผนความถี่วิทยุ FM ของประเทศ ได้กำหนดให้สถานีในต่างจังหวัดใช้ความถี่ห่างกันอย่างน้อย 750 KHz

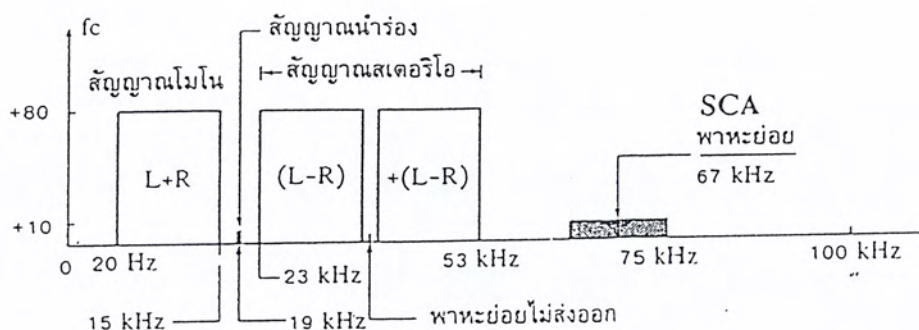
2.4.3 การรบกวนกันเองภายในระบบ

โดยที่การส่งคลื่นระบบ SCA หมายถึง การผสมคลื่นอีกชุดหนึ่ง (หรือหลายชุด) เพิ่มจากการผสมคลื่น FM ตามปกติในช่องความถี่เดียวกัน คลื่นผสมดังกล่าว จึงมีช่วงความถี่อยู่ไม่ห่างกันนัก ดังนั้น โอกาสที่จะเกิดการรบกวนกันเองในสถานีส่งระหว่างคลื่นความถี่หลักกับคลื่น SCA (crosstalk) จึงอาจเกิดขึ้นได้หากไม่มีการควบคุมที่ดีพอ โดยเฉพาะอย่างยิ่ง กรณีที่คลื่นความถี่หลักเป็นชนิด FM stereo ซึ่งต้องใช้ความกว้างแถบความถี่ถึง 30 - 53 KHz การรบกวนกันเองจึงยิ่งเป็นไปได้มาก

เกี่ยวกับเรื่องนี้ ในกรณีของประเทศสหรัฐอเมริกา FCC (Federal Communication commission) ซึ่งเป็นหน่วยงานกำหนด และ ควบคุมการใช้วิทยุของหน่วยงานเอกชนของประเทศได้ กำหนดไว้ว่า ในกรณีของระบบ FM mono ความถี่ของคลื่นผสมในระบบ SCA ที่ใช้ จะอยู่ในระหว่าง 20 - 75 KHz โดยไม่จำกัดว่าจะใช้ชุดคลื่นผสม SCA ที่ชุด (ที่รายการ) และผลรวมของความแรงในการผสมคลื่น SCA จะต้องสูงไม่เกิน 30 % และตัวสัญญาณรบกวนของ SCA ที่เข้าไปในช่องความถี่ของสัญญาณ mono จะต้องมีความแรงของสัญญาณ mono ไม่น้อยกว่า 60 dB

กรณีของสถานีวิทยุระบบ FM stereo ความถี่ของคลื่นผสม SCA จะจำกัดอยู่ในระหว่าง 53 - 75 KHz ความแรงในการผสมคลื่น SCA จะต้องไม่เกิน 10 % (ของผลรวมคลื่นผสมทั้งหมดของสถานี) และสัญญาณทั้งหมดของ SCA ที่เลขเข้าไปในช่วงสัญญาณ stereo จะต้องมีความแรงต่ำกว่าสัญญาณ stereo ไม่น้อยกว่า 60 dB (เช่นเดียวกับกรณีของสถานี mono)

เพื่อป้องกันปัญหาการรบกวนกันเองดังกล่าว สถานีส่งที่มีการส่งคลื่นระบบ SCA ด้วย จะต้องรับผิดชอบในการดำเนินการทางเทคนิค ให้เป็นไปตามที่หน่วยงานรับผิดชอบของแต่ละประเทศได้กำหนดไว้หรือไม่ให้เกิดการรบกวนดังกล่าวขึ้น โดยอาจถือเอาตามรายงานของคณะกรรมการที่ปรึกษาการวิทยุระหว่างประเทศ (CCIR) หรือถือปฏิบัติตามก่อนหน้าในประเทศอื่นในกรณีที่ประเทศนั้นๆ ยังไม่มีการกำหนดในส่วนที่เกี่ยวกับ SCA ไว้



รูปที่ 2.16 แถบความถี่สัญญาณ SCA

2.5 การมอดูเลตสัญญาณดิจิทัล

การส่งสัญญาณดิจิทัล เช่น สัญญาณ PCM โดยคลื่นวิทยุไมโครเวฟ จำเป็นที่จะต้องเปลี่ยนสัญญาณดิจิทัลไปเป็นสัญญาณในย่านความถี่วิทยุ เทคนิคในการมอดูเลตสัญญาณดิจิทัลนี้มี 3 แบบด้วยกันคือ

1. Amplitude Shift Keying (ASK)
2. Frequency Shift Keying (FSK)
3. Phase Shift Keying (PSK)

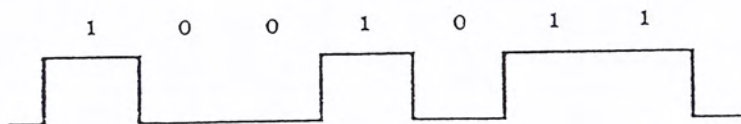
คลื่นพาหะในย่านความถี่วิทยุ หรือไมโครเวฟสามารถกำหนดได้โดย

$$\text{คลื่นพาหะ} = A \cos(2\pi f_c t + \sigma)$$

ในที่นี้

A	:	แอมพลิจูดของคลื่นพาหะ
f_c	:	ความถี่คลื่นพาหะ
σ	:	initial phase

สัญญาณดิจิทัลเบสแบนด์เป็นคลื่นรูปสี่เหลี่ยมแสดงรหัสไบนารี 1 และ 0 ในการมอดูเลตสัญญาณดิจิทัลหนึ่งในระบบพารามิเตอร์คือ แอมพลิจูด ความถี่ หรือ เฟสของคลื่นพาหะจะเปลี่ยนไปตามสถานะ 1 หรือ 0 ของสัญญาณเบสแบนด์



รูปที่ 2.17 สัญญาณเบสแบนด์ดิจิทัล

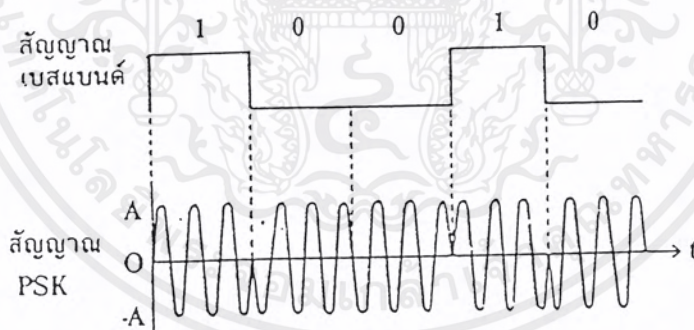
การมอดูเลตแบบดิจิตอล	การมอดูเลตแบบอนาล็อก
A S K Amplitude Shift Keying	AM
F S K Frequency Shift Keying	FM
P S K Phase Shift Keying	PM

รูปที่ 2.18 เปรียบเทียบวิธีการมอดูเลตแบบดิจิตอลและอนาล็อก

ในที่นี้เราจะมาพิจารณาในกรณีของ PSK ซึ่งได้นำหลักการนี้มาใช้ในโครงงานนี้ กรณีของ PSK แอมพลิจูดและความถี่จะคงที่แต่ initial phase จะต่างกันสำหรับสัญญาณที่เป็น 1 หรือ 0 คือเฟสของคลื่นพาหะ $A \cos(2\pi f_c t + \sigma)$ จะเปลี่ยนไปตามสถานะของสัญญาณ เช่น เฟส σ เป็น 0 กรณีสัญญาณมีสถานะเป็น 1 และเฟส σ เป็น π กรณีสัญญาณมีสถานะเป็น 0

$$S(t) = A \cos 2\pi f_c t \quad : \quad \text{กรณีสัญญาณเป็น 1}$$

$$S(t) = A \cos(2\pi f_c t + \pi) \quad : \quad \text{กรณีสัญญาณเป็น 0}$$

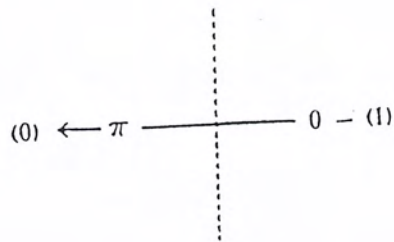


รูปที่ 2.19 phase shift keying

2.5.1 เฟสชิฟคีย์อิง (PSK)

โดยหลักการแล้วเฟสเริ่มแรกของคลื่นพาหะมีจำนวนมาก จำนวนของเฟสเริ่มแรกที่ใช้ใน PSK สามารถที่จะเพิ่มขึ้นได้ ถ้าเราสามารถแยกแยะสัญญาณทางด้านรับออกได้อย่างถูกต้อง

เนื่องจากระบบดิจิตอลใช้รหัสไบนารี สัญญาณดิจิตอลสามารถส่งได้โดยใช้ 2 initial phase เรียกว่า ไบนารี PSK เฟสเป็น 0 สำหรับรหัส 1 และเฟสเป็น π สำหรับรหัส 0



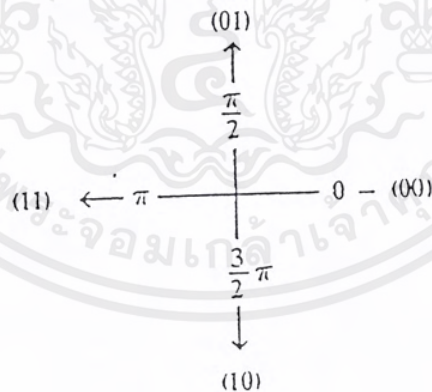
รูปที่ 2.20 ไบนารี PSK

ลองมาพิจารณาการส่งสัญญาณดิจิทัลพร้อมกัน 2 บิตในเวลาเดียวกัน ในกรณีนี้จะมีการรวมรหัส 4 ตัว ด้วยกันคือ 00, 01, 10 และ 11 โดยใช้รหัสที่รวมกันนี้เทียบกับเฟส 4 เฟส สัญญาณดิจิทัล 2 บิตจึงสามารถส่งพร้อมกันได้

1st bit	0	0	1	1
2nd bit	0	1	0	1

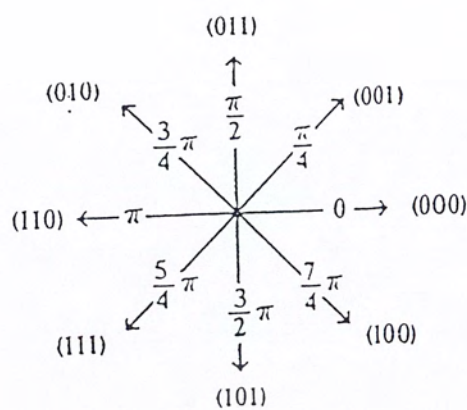
รูปที่ 2.21 สัญญาณดิจิทัล 2 บิต ที่จะส่งพร้อมกัน

PSK ที่ใช้ 4 เฟสเริ่มแรกเรียกว่า quadri PSK เฟส 0 สำหรับรหัส 00, เฟส $\pi/2$ สำหรับรหัส 01, เฟส π สำหรับรหัส 11 และ เฟส $3\pi/4$ สำหรับรหัส 10



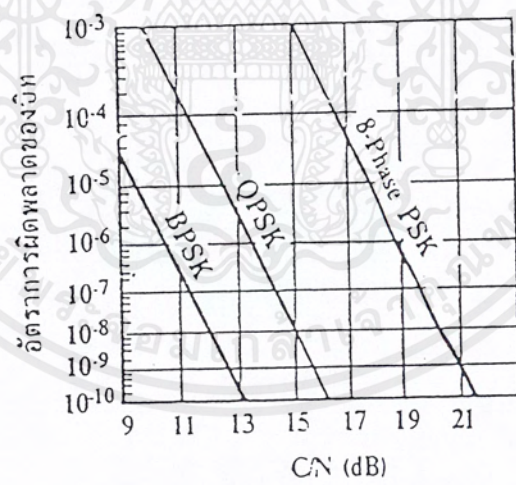
รูปที่ 2.22 Quadri PSK

เมื่อต้องการจะส่งสัญญาณ 3 บิตไปพร้อมๆกัน จะมี $2^3 = 8$ combination ของรหัสสัญญาณ 8 - เฟส PSK จะต้องใช้เพื่อการนี้ เมื่อจำนวนเฟสเพิ่มขึ้น จำนวนบิตที่สามารถส่งแต่ละครั้งก็มาก แต่กรณีนี้คุณภาพของสัญญาณทางด้านรับจะต่ำลง



รูปที่ 2.23 8-เฟส PSK

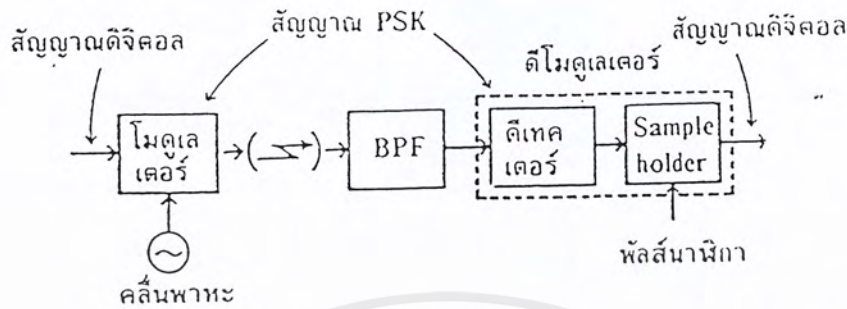
จากรูปต่อไปนี้จะเห็นว่า อัตราการผิดพลาดของบิตจะลดลง ถ้าเพิ่มจำนวนของเฟสมากขึ้น ใน การรับส่งสัญญาณ



รูปที่ 2.24 เปรียบเทียบอัตราการผิดพลาดของบิตของ PSK แบบต่างๆ

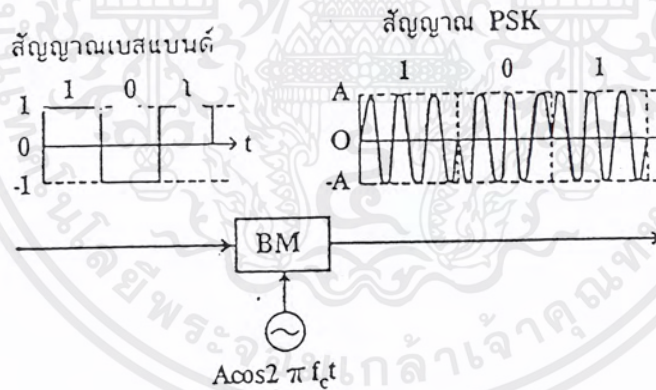
2.5.2 หลักการของไบนารี PSK โมดูเลเตอร์และดีโมดูเลเตอร์

ทางด้านส่งมีโมดูเลเตอร์ ซึ่งเปลี่ยนสัญญาณดิจิทัลเป็นสัญญาณ PSK ทางด้านรับมี band - pass filter และดีโมดูเลเตอร์ ซึ่งจะเปลี่ยนสัญญาณ PSK เป็นสัญญาณดิจิทัลอันเดิม ดีโมดูเลเตอร์ประกอบด้วย ดีเทคเตอร์ และ sample holder



รูปที่ 2.25 การรับ - ส่ง สัญญาณดิจิทัล

โดยทั่วไปแล้วการโมดูเลต PSK จะใช้ balanced modulator สมมติว่าสัญญาณดิจิทัลเบสแบนด์ อยู่ในรูปคลื่นสี่เหลี่ยมมีแอมพลิจูดเท่ากับ 1 สำหรับรหัส 1 และมีแอมพลิจูดเท่ากับ -1 สำหรับรหัส 0 เมื่อ คลื่นรูปสี่เหลี่ยมนี้ถูกผสมโดยคลื่นแคเรียร์ $A \cos 2\pi f_c t$ ผ่านวงจร balanced modulator สัญญาณ PSK ที่ ได้แสดงดังในรูป



รูปที่ 2.26 การ โมดูเลตสัญญาณดิจิทัล

สัญญาณ PSK สำหรับรหัส 1 จะเป็น $S(t) = 1 \times A \cos 2\pi f_c t = A \cos 2\pi f_c t$ ส่วนสัญญาณ PSK สำหรับรหัส 0 จะเป็น $S(t) = -1 \times A \cos 2\pi f_c t = -A \cos 2\pi f_c t$ เราสามารถเขียน $-A \cos 2\pi f_c t = A \cos (2\pi f_c t + \pi)$ ตามหลักการนี้แสดงว่าสำหรับรหัส 0 เฟสของคลื่นแคเรียร์จะเปลี่ยนไปเท่ากับ π ซึ่งจะเห็นได้ชัดเจนดังในรูป

กรณี 1

$$S(t) = 1 \times A \cos 2\pi f_c t$$

$$= A \cos 2\pi f_c t$$

กรณี 2

$$S(t) = -1 \times A \cos 2\pi f_c t$$

$$= -A \cos 2\pi f_c t$$

$$= A \cos (2\pi f_c t + \pi)$$

กรณี 1

$$S(t) = 1 \times A \cos 2\pi f_c t$$

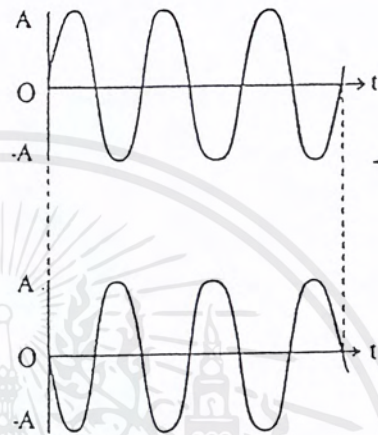
$$= A \cos 2\pi f_c t$$

กรณี 0

$$S(t) = -1 \times A \cos 2\pi f_c t$$

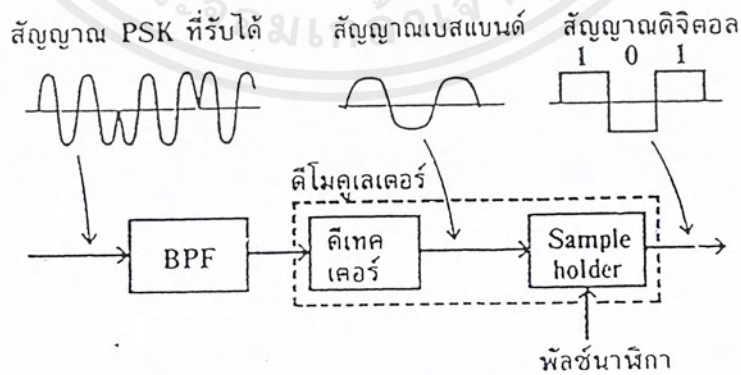
$$= -A \cos 2\pi f_c t$$

$$= A \cos (2\pi f_c t + \pi)$$



รูปที่ 2.27 การเปลี่ยนแปลงทางเฟสของคลื่นพาหะ

Band-pass filter ทางด้านรับรับสัญญาณ ซึ่งมีเสียงรบกวนปนอยู่ด้วยจะถูกดีเทคเตอร์ที่ติดอูเลเตอร์, ดีเทคเตอร์จะเอาสัญญาณเบสแบนด์ออกมาโดยคัดแคเรียร์คอมโพเนนต์ ซึ่งรวมอยู่ในสัญญาณ PSK ออกไป สัญญาณเบสแบนด์จะไม่เป็นรูปสี่เหลี่ยมที่สมบูรณ์นัก เนื่องจากการกำหนดแถบความกว้างโดยตัวกรองความถี่ และเนื่องจากอิทธิพลของเสียงรบกวน sample holder จะสร้างสัญญาณดิจิทัลเดิมขึ้นมาใหม่ผ่านการพิจารณา polarity บวก หรือ ลบ ของสัญญาณเบสแบนด์

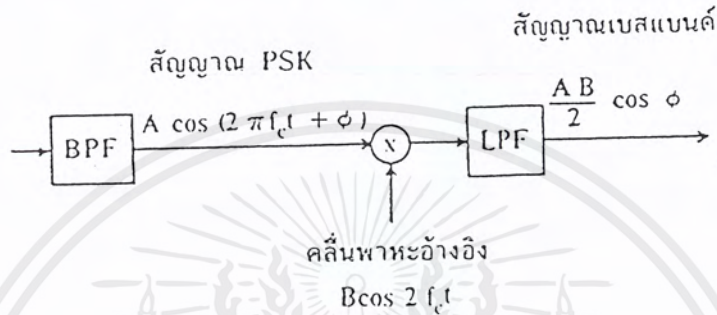


รูปที่ 2.28 การสร้างสัญญาณขึ้นมาใหม่ทางด้านรับ

2.5.3 โคอีเรนทีเทกชัน

การดีโมดูเลตสัญญาณ PSK ทางด้านรับเป็นแบบ coherent detection ซึ่งมีหลักการดังนี้ สัญญาณ PSK ที่รับได้ขั้นแรกจะผ่านวงจร multiplier ซึ่งจะคูณสัญญาณ PSK ที่รับได้ด้วย reference carrier สำหรับฮาร์โมนิก ซึ่งรวมอยู่ในสัญญาณ output จะถูกคัดออกไปโดยผ่าน low pass filter และได้สัญญาณเบสแบนด์ออกมาทางด้านทางออก

ถ้าสัญญาณ PSK ที่รับเข้ามาคือ 0 คลื่นพาหะอ้างอิงมีไว้เพื่อการซิงโครไนซ์กับคลื่นพาหะทางด้านส่งคือ $B \cos 2 f_c t$ สัญญาณทางออกหลังจากผ่าน LPF แล้วสามารถกำหนดได้เป็น $AB/2 \cos \tau$

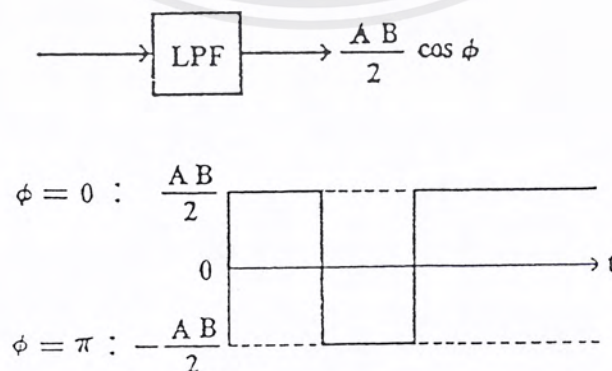


รูปที่ 2.29 การรับสัญญาณ PSK ทางด้านรับ

ทั้งนี้เนื่องจาก

$$\begin{aligned} & A \cos(2\pi f_c t + \tau) \cdot B \cos 2\pi f_c t \\ &= AB/2 \{ \cos(2\pi f_c t + \tau + 2\pi f_c t) + \cos(2\pi f_c t + \tau - 2\pi f_c t) \} \\ &= AB/2 \{ \cos(4\pi f_c t + \tau) + \cos \tau \} \end{aligned}$$

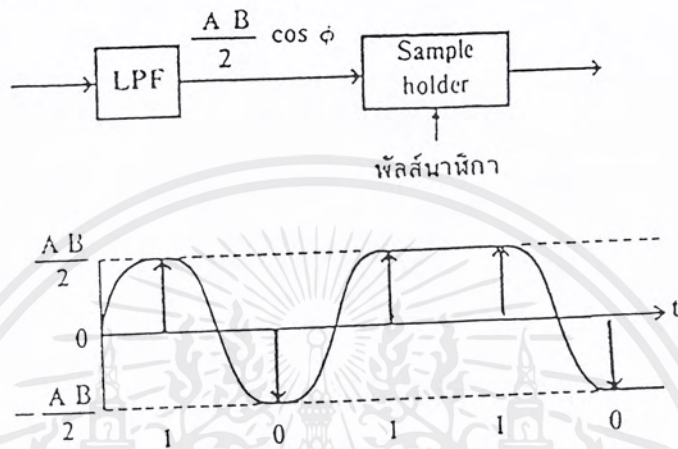
ซึ่ง $AB/2 \cos(4\pi f_c t + \tau)$ นี้คือฮาร์โมนิก โดยมีความถี่เป็น 2 เท่าของความถี่คลื่นพาหะเดิม ซึ่งจะถูกคัดออกไปเมื่อผ่าน low pass filter ดังนั้น ทางด้านทางออกจึงมีเพียง $AB/2 \cos \tau$ เท่านั้น $AB/2 \cos \tau$ นี้จะได้เป็น $AB/2$ เมื่อ $\tau = 0$ จะได้เป็น $-AB/2$ เมื่อ $\tau = \pi$ ดังในรูป



รูปที่ 2.30 สัญญาณ output หลังจากผ่าน LPF

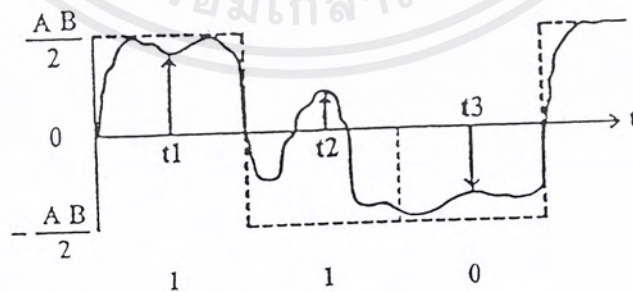
สัญญาณ output ในรูปข้างบน ในทางปฏิบัติแล้วจะไม่เป็นรูปสี่เหลี่ยมโดยสมบูรณ์ ทั้งนี้เนื่องจากอิทธิพลของ band limitation และเสียงรบกวน

ต่อจากนี้ สัญญาณเบสแบนด์นี้จะถูกส่งไปยังวงจร sample holder ซึ่งวงจรจะทำการสุ่มตัวอย่างสัญญาณเบสแบนด์ด้วย clock pulse ที่จุดกึ่งกลางของแต่ละรหัสสัญญาณ และกำหนดว่ารหัสนั้นเป็น 1 หรือ เป็น 0 โดยการตัดสินใจหรือพิจารณาจาก polarity ของสัญญาณ sample นั้นแล้วทำการสร้างสัญญาณดิจิตอลดั้งเดิมขึ้นมาใหม่ ในรูปนี้รหัสเป็น 1 สำหรับสัญญาณที่มีแอมพลิจูดเป็นบวก และรหัสเป็น 0 สำหรับสัญญาณที่มีแอมพลิจูดเป็นลบ



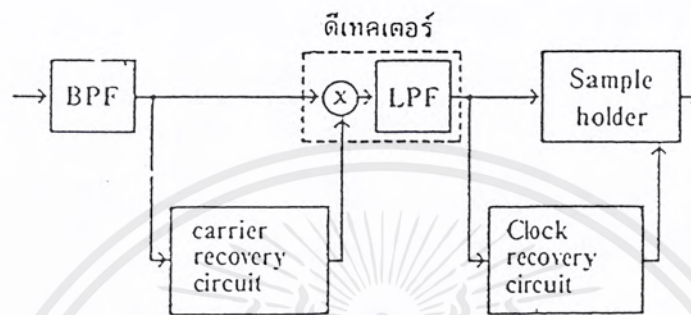
รูปที่ 2.31 การพิจารณารหัสเป็น 1 หรือ เป็น 0 โดย sample holder

เมื่อสัญญาณมีการผิดเพี้ยนเนื่องจากอิทธิพลของเสียงรบกวน ถ้ามีการผิดเพี้ยนน้อย polarity ของสัญญาณก็ไม่เปลี่ยนแปลง เช่นที่ t_1 และ t_3 รหัสที่ได้ออกมาก็ถูกต้อง แต่ถ้าพิจารณาที่ t_2 polarity ของสัญญาณถูกเปลี่ยนไปในทางตรงกันข้ามเนื่องจากเกิดการผิดเพี้ยนมาก รหัสที่ได้ออกมาก็จะผิดไปจากเดิม bit error ก็เกิดขึ้น



รูปที่ 2.32 รหัสผิดเพี้ยนเนื่องจากเสียงรบกวน

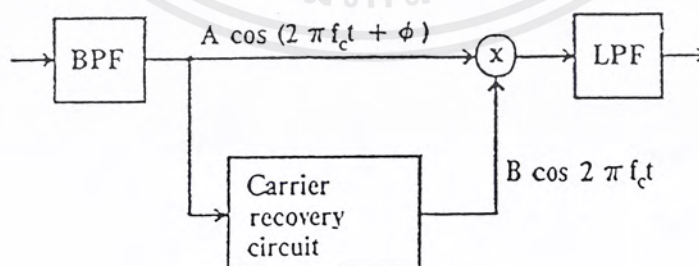
รูปต่อไปนี้เป็นบล็อกไดอะแกรม coherent demodulator มี BPF ดีเทคเตอร์ sample holder carrier recovery circuit และ clock recovery circuit วงจร carrier recovery circuit จะสร้างคลื่นพาหะอ้างอิงขึ้นมาใหม่ โดยการทวีคูณสัญญาณ PSK ที่รับมาได้ ส่วน clock recovery circuit จะสร้างสัญญาณ clock ขึ้นมาใหม่เพื่อใช้สำหรับ sample holding ในวิธีการ coherent detection นี้ ทั้งคลื่นพาหะอ้างอิง และสัญญาณ clock จะถูกสร้างขึ้นใหม่จากสัญญาณ PSK ที่รับเข้ามา



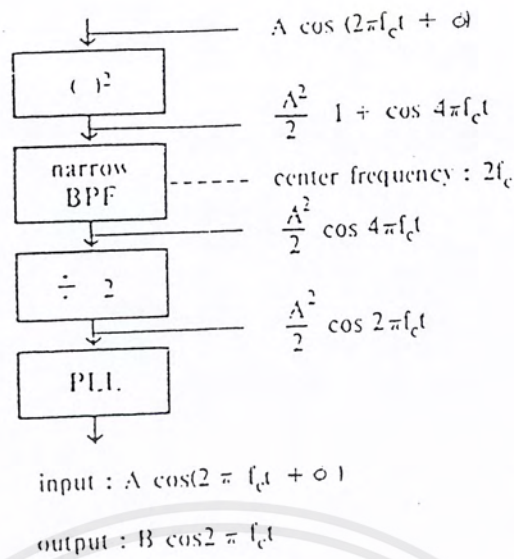
รูปที่ 2.33 บล็อกไดอะแกรมของ coherent demodulator

คลื่นพาหะอ้างอิงที่ใช้ใน coherent detection นี้ เพื่อการซิงโครไนซ์กับคลื่นพาหะอ้างอิงทางด้านส่ง เพื่อจุดประสงค์นี้คลื่นพาหะอ้างอิงถูกสร้างขึ้นใหม่จากสัญญาณ PSK ที่รับเข้ามา สัญญาณ input ที่เข้าวงจร carrier recovery circuit สามารถกำหนดโดย $A \cos(2\pi f_c t + \tau)$ สัญญาณ output ก็คือคลื่นพาหะอ้างอิงกำหนดโดย $B \cos 2\pi f_c t$

วงจร carrier recovery ประกอบด้วย squarer, narrowband pass filter โดยมีความถี่กึ่งกลางที่ $2f_c$ วงจร 1/2 frequency divider และวงจร phase lock loop

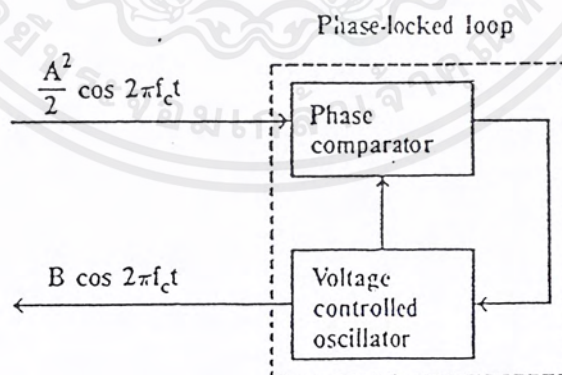


รูปที่ 2.34 หลักการ carrier recovery circuit



รูปที่ 2.35 carrier recovery circuit และสัญญาณตามจุดต่างๆ

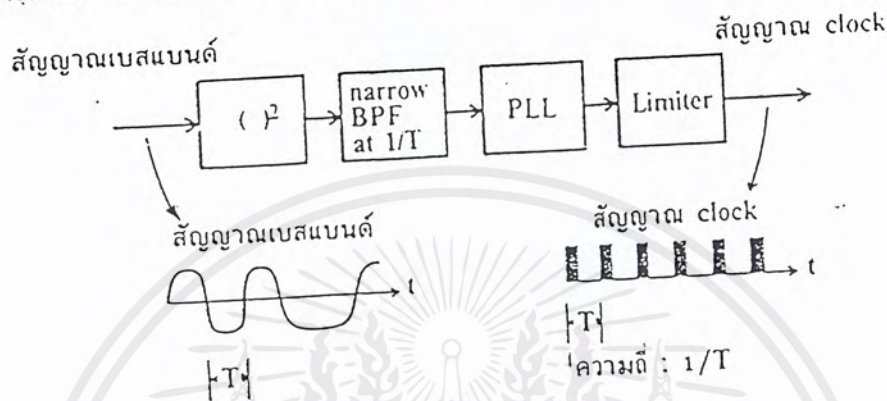
สัญญาณ $\{ (A^2/2) \cos 2\pi f_c t \}$ ที่ออกจากวงจร divider ปกติจะมีเสียงรบกวนรวมอยู่ด้วย เพื่อที่จะตัดเสียงรบกวนออก และได้คลื่นพาหะอ้างอิงปราศจากการผิดเพี้ยน จำเป็นต้องใช้วงจร phase lock loop ช่วย วงจร phase lock loop นี้ประกอบด้วย phase comparator และ voltage controlled oscillator (VCO) ที่วงจร phase comparator สัญญาณ output ของวงจร divider จะถูกเปรียบเทียบอย่างต่อเนื่องกับ output ของ VCO เมื่อมีความต่างทางเฟสระหว่างสัญญาณทั้งสองนี้ สัญญาณ output ของ VCO จะถูกควบคุมอย่างอัตโนมัติในการปรับเฟสของสัญญาณทั้งสอง โดยวิธีการนี้คลื่นพาหะอ้างอิง $B \cos 2\pi f_c t$ ซึ่งไม่มีเสียงรบกวนรวมอยู่ด้วย สามารถที่จะสร้างขึ้นใหม่



รูปที่ 2.36 การทำงานของ phase lock loop

Clock pulse ที่จำเป็นสำหรับ sample holding ก็ถูกสร้างขึ้นใหม่จากสัญญาณ PSK ที่รับได้เช่นกัน เพื่อจุดประสงค์นี้ ต้องใช้วงจร clock recovery สัญญาณเบสแบนด์ที่ได้ที่ output ของดีเทกเตอร์ โดยทั่วไป จะใช้เป็นสัญญาณ input ให้กับวงจร clock recovery

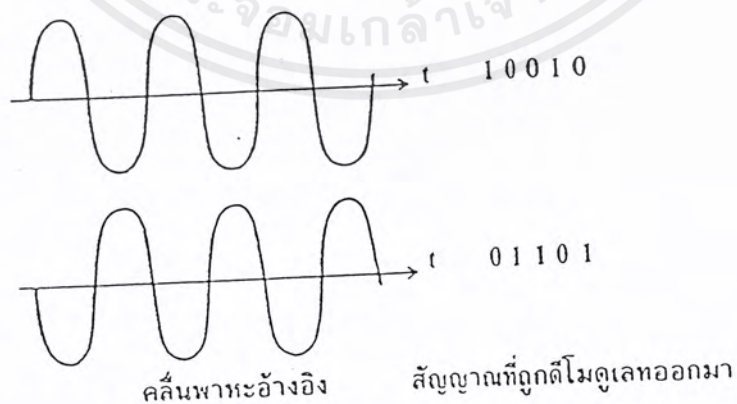
มีอยู่หลายวิธีในการสร้างสัญญาณ clock ขึ้นใหม่ วงจรข้างล่างนี้ เป็นตัวอย่างของวงจรหนึ่ง ซึ่ง ประกอบด้วย square, narrow BPF, PLL และลิมิตเตอร์ สำหรับ narrow BPF ที่ใช้ในวงจรนี้ มีความถี่กึ่งกลางของ $1/T$ ซึ่ง T คือ ความกว้างของรหัสตัวหนึ่งๆ ดังนั้น $1/T$ คือ ความถี่ของสัญญาณ clock ที่ได้ โดยผ่านสัญญาณเบสแบนด์ที่ได้รับที่ output ของดีเทกเตอร์ผ่านวงจรนี้ เราจะได้สัญญาณ clock ที่ความถี่ $1/T$



รูปที่ 2.37 วงจร clock recovery

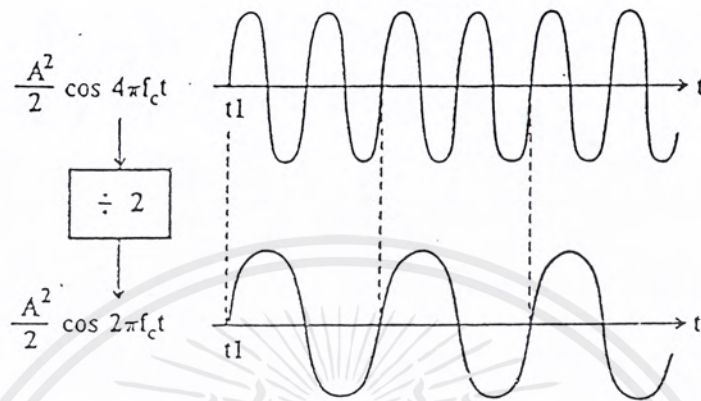
2.5.4 การผิดพลาดทางเฟส (phase ambiguity)

คลื่นพาหะอ้างอิงที่ใช้ใน coherent detection มีความจำเป็นเพื่อล็อกเฟสกับคลื่นพาหะทางด้านการส่ง แต่อย่างไรก็ตาม ปกติแล้วเป็นไปไม่ได้ที่จะทำให้คลื่นพาหะสองชนิดนี้มีการล็อกเฟสซึ่งกันและกันอย่างสมบูรณ์ ในไมนารี PSK ความต่างเฟสเท่ากับ อาจมีสาเหตุจากความต่างระหว่างคลื่นทั้งสอง แต่เป็นไปไม่ได้ที่จะรู้ความต่างนี้ที่ทางด้านรับ ความไม่สามารถที่จะ detect ความต่างเฟสนี้เป็นผลทำให้เกิดการผิดพลาดทางเฟส เมื่อคลื่นพาหะอ้างอิงเกิดการ out of phase เท่ากับ เครื่องหมายของบิตที่อยู่ในการดีมอดูเลต สัญญาณดิจิตอล จะเกิดการเปลี่ยนกลับจาก 1 เป็น 0 และจาก 0 เป็น 1



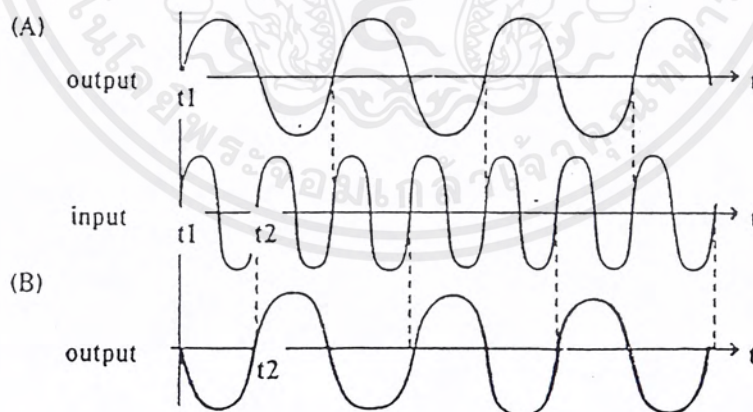
รูปที่ 2.38 การผิดพลาดทางเฟส

การผิดพลาดทางเฟสมีสาเหตุมาจาก frequency divider ในวงจร carrier recovery ในการหาความถี่สัญญาณ $\cos 4\pi f_c t$ ถูกหารความถี่ด้วย 2 จะได้ $\cos 2\pi f_c t$ รูปต่อไปนี้แสดงรูปคลื่นดั้งเดิม และรูปคลื่นทาง output ที่ได้หลังการผ่านวงจรหารความถี่ด้วย 2 ซึ่งมีจุดเริ่มแรกที่ t_1 โดยการกำหนดจุดเริ่มแรกที่ต่างกันเราจะได้รูปคลื่น output ที่ต่างกัน



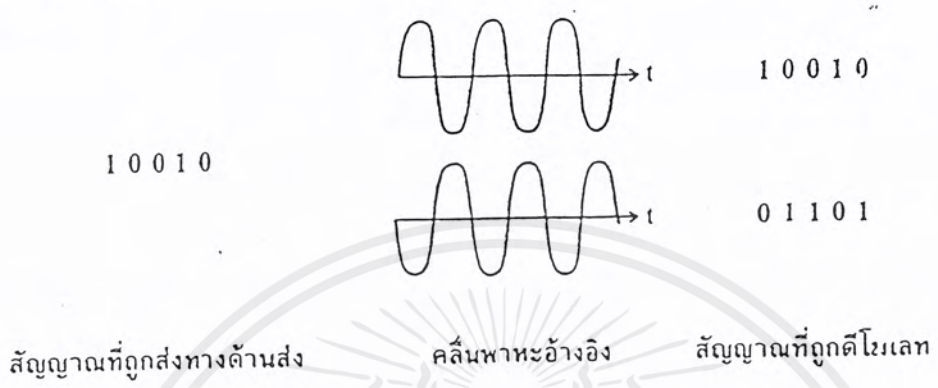
รูปที่ 2.39 คลื่นสัญญาณเมื่อผ่านวงจรหารความถี่

รูปข้างล่างนี้เป็น input และ output ของวงจรหารความถี่ซึ่งใช้ในวงจร carrier recovery ซึ่ง output (A) ได้จากกรณี t_1 เป็นจุดเริ่มส่วน output (B) ได้จากกรณีให้ t_2 เป็นจุดเริ่ม จะเห็นว่า output ทั้งสองกรณีกลับเฟสกัน



รูปที่ 2.40 เปรียบเทียบ output กรณี t_1 และ t_2 เป็นจุดเริ่ม

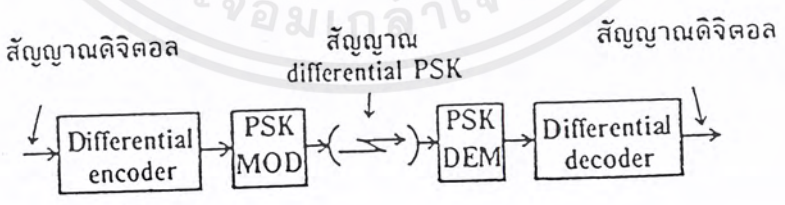
เนื่องจากรูปคลื่น output 2 แบบ สามารถทำให้เกิดขึ้นที่ output ของวงจรหาความถี่ ดังนั้นเรา
ต้องมาพิจารณารูปคลื่นของคลื่นพาหะอ้างอิง 2 แบบ แบบหนึ่งเฟสตรงกับคลื่นพาหะทางด้านส่ง และอีก
แบบหนึ่งต่างเฟส π กับคลื่นพาหะ ตามปกติแล้วแบบใดแบบหนึ่งของคลื่นพาหะอ้างอิงใน 2 แบบนี้ ใช้
ในการคิ่มอดูเลตอยู่แล้ว อย่างไรก็ตามไม่มีการกำหนดแน่นอนว่าแบบไหนที่ใช้สำหรับการคิ่มอดูเลต และ
ดังนั้นเราจึงไม่สามารถกำหนดได้ว่าสัญญาณดิจิตอลที่ได้รับหลังจากผ่านการคิ่มอดูเลตเป็นแบบปกติ หรือ
เป็นแบบกลับเฟส



รูปที่ 2.41 สัญญาณดิจิตอลหลังผ่านการคิ่มอดูเลต

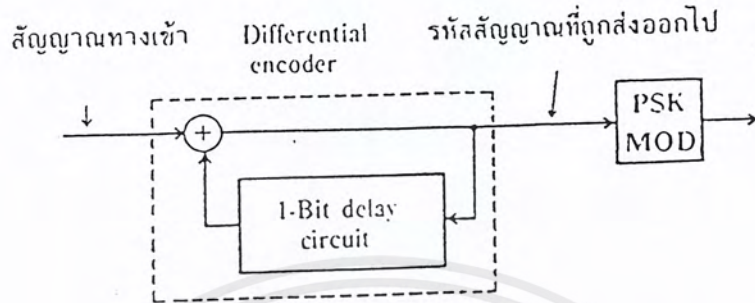
วิธีการที่จะแก้ปัญหาคือการผิดพลาดทางเฟสคือ differential coherent PSK โดยวิธีการนี้ ขั้วสารใน
สัญญาณดิจิตอลที่จะส่งจะถูกทำให้เฟสของสัญญาณ PSK เปลี่ยนไปคือ ไม่ให้เป็น initial phase ของ
สัญญาณ PSK

จากรูปสัญญาณดิจิตอลที่จะส่งทางด้านส่งจะถูก differential encoder ก่อนที่จะทำการมอดูเลต
แบบ PSK เพื่อประสงค์นี้ differential encoder จะอยู่ข้างหน้าวงจร PSK modulator สำหรับทางด้านรับ
สัญญาณ differential PSK จะถูกคิ่มอดูเลตแบบ coherent demodulate และป้อนเข้าวงจร differential
decoder เพื่อสร้างสัญญาณดิจิตอลดั้งเดิมขึ้นมาใหม่



รูปที่ 2.42 differential coherent PSK

Differential encoder ที่ใช้ประกอบด้วย Exclusive - OR gate และ one bit delay circuit รหัสสัญญาณจะถูกส่งไปหนึ่งบิตก่อน และถูกบวกกับแต่ละบิตโดย Exclusive -OR gate สร้างรหัสขึ้นมาเพื่อจะส่งไป รหัสที่ส่งไปนี้ถูกผ่าน PSK modulator และในเวลาเดียวกัน จะเข้าวงจร one bit delay อีกครั้งเพื่อบวกกับสัญญาณดิจิทัล input ตัวต่อไป

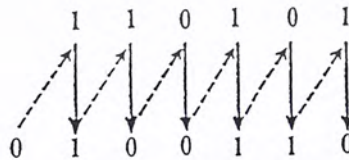


$\oplus$: EXCLUSIVE - OR gate

รูปที่ 2.43 วงจร differential encoder

สมมติว่าเรามีสัญญาณที่จะส่งเป็น 110101 โลกิก ต่อไปนี้จะถูกนำมาใช้สำหรับ Exclusive - OR gate คือ $0 + 0 = 0$, $0 + 1 = 1$, $1 + 0 = 1$ และ $1 + 1 = 0$ ขั้นแรกให้ค่าเริ่มแรกของ one bit delay circuit เป็น 0 และให้ 0 นี้เป็นรหัสแรกที่จะส่งไป บวกบิตแรกของสัญญาณดั้งเดิมคือ "1" เข้ากับรหัสแรกที่จะส่งคือ "0" นั่นคือ $0 + 1 = 1$ ตอนนี้ให้เรากำหนดรหัส "1" นี้เป็นรหัสที่สองที่จะส่ง บวกบิตที่สองคือ "1" ของสัญญาณดั้งเดิมเข้ากับรหัสตัวที่สองที่จะส่ง นั่นคือ $1 + 1 = 0$ ให้เรากำหนดรหัส "0" นี้เป็นรหัสที่สามที่จะส่ง โดยการดำเนินขั้นตอนซ้ำกันไปอย่างนี้เรื่อยๆ ลำดับของรหัสที่จะต้องส่งจะเป็น 0100110 หลังจากขบวนการ differential encoding จะเห็นได้ว่า รหัสที่จะส่งหลังจากทำการ differential encoding จะเหมือนกับรหัสที่ส่งไปก่อนหนึ่งบิต เมื่อสัญญาณ input เป็น "0" และจะกลับกันเมื่อสัญญาณ input เป็น "1"

รหัสสัญญาณเดิม



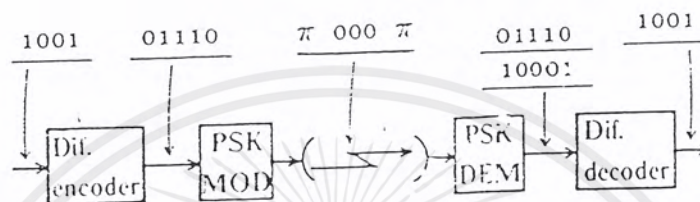
รหัสสัญญาณที่ถูกส่งออกไป

where $0 + 0 = 0$
 $0 + 1 = 1$
 $1 + 0 = 1$
 $1 + 1 = 0$

รูปที่ 2.44 การส่งสัญญาณที่ differential encoder

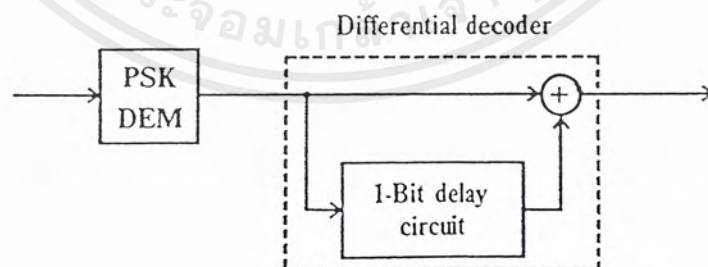
ลำดับสัญญาณ differential encoder คือ PSK มอดูเลตโดยการทำให้ initial phase π แทนรหัส 0 และ initial phase 0 แทนรหัส 1 แล้วส่งออกไป ส่วนทางด้านรับ สัญญาณ differential PSK ถูกตีเทคแบบ coherent เพื่อสร้างสัญญาณดิจิทัลที่ส่งขึ้นมาใหม่ สัญญาณดิจิทัลที่ได้โดยผ่าน coherent detection นี้ อาจจะเปลี่ยนจาก 1 เป็น 0 หรือ จาก 0 เป็น 1 เนื่องจาก phase ambiguity อย่างไรก็ตามใน differential coherent PSK ลำดับของสัญญาณที่ได้รับหลังจากการ differential encoding จะพร้อมกันกับลำดับสัญญาณดั้งเดิมเสมอ โดยปราศจากผลจาก phase ambiguity

การผลิตพลาตทางเฟส



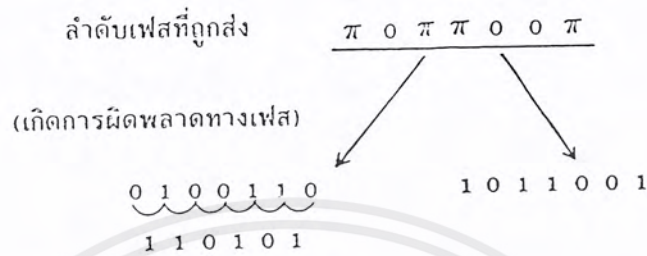
รูปที่ 2.45 รหัสสัญญาณ ณ ตำแหน่งต่างๆ

Differential decoder ประกอบด้วยวงจร one bit delay และ Exclusive – OR gate สัญญาณดิจิทัลที่ถูกมอดูเลตผ่านขบวนการ coherent detection ถูกป้อนเข้า Exclusive – OR gate และในขณะเดียวกันก็ป้อนเข้าวงจร one bit delay ใน Exclusive – OR gate สัญญาณดิจิทัล input และสัญญาณ output ของวงจร one bit delay ถูกบวกเข้าด้วยกัน นั่นคือ โดยการบวก 2 บิตข้างเคียงในลำดับสัญญาณดิจิทัลที่ได้มอดูเลตโดย Exclusive – OR gate การเปลี่ยนแปลงในทุกๆ 2 บิตข้างเคียงจะถูกตรวจสอบตลอดเวลา



รูปที่ 2.46 differential decoder

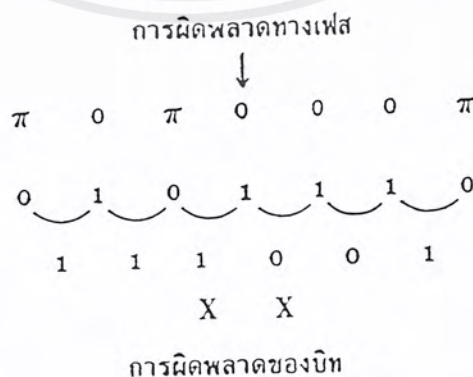
สมมติว่าลำดับเฟสที่ส่งของสัญญาณ PSK เป็น $\pi \ 0 \ \pi \ \pi \ 0 \ 0 \ \pi$ ทางด้านรับหนึ่งในสองลำดับสัญญาณดิจิทัล อาจจะได้รับที่ output ของ demodulator เนื่องจาก phase ambiguity นั่นคือ 0100110 หรือ 1011001 ตอนนี้ให้เรา decode 0100110 Exclusive – OR gate บวก 2 บิตข้างเคียง โดยการบวกบิตแรกกับบิตที่สอง จะได้ $0 + 1 = 1$ บวกบิตที่สอง กับบิตที่สามจะได้ $1 + 0 = 1$ และการทำอย่างนี้เรื่อยๆ ไปจะได้ 110101



รูปที่ 2.47 ลำดับสัญญาณดิจิทัลด้านรับ

โดยการให้สัญญาณผ่าน differential encoder ทางด้านส่ง ลำดับสัญญาณดั้งเดิมถูกเปลี่ยนไป เพื่อให้ information ยังคงมีอยู่ แม้มีการเปลี่ยนแปลงใน 2 บิตข้างเคียงของลำดับรหัสส่ง ส่วนทางด้านรับ ลำดับรหัสที่ถูกดีมอดูเลตถูกป้อนเข้าไปใน differential decoder เพื่อเอาสัญญาณดั้งเดิมออกมา โดยการตรวจสอบการเปลี่ยนแปลงใน 2 บิตข้างเคียง ในลักษณะนี้ information ยังคงมีอยู่ดังเดิม แม้มีการเปลี่ยนแปลงของ initial phase ในสัญญาณที่มอดูเลตแบบ PSK เนื่องจาก information ถูกทำให้คงอยู่ ถึงแม้ว่าถ้าคลื่นพาหะอ้างอิงที่ใช้สำหรับ coherent detection ต่างเฟสไป ก็ตาม และดังนั้นอิทธิพลของ phase ambiguity สามารถที่จะแก้ปัญหาก็ได้

Differential coherent PSK มีข้อดีตรงที่ว่า ไม่มีผลจากการที่เกิด phase ambiguity แต่มีข้อเสียเกี่ยวกับ bit error คือเมื่อ initial phase อันหนึ่งของสัญญาณ differential PSK ถูกการดีมอดูเลตอย่างไม่ถูกต้อง จะเป็นสาเหตุให้เกิด bit error คู่หนึ่ง เพราะว่าสัญญาณดั้งเดิม ถูกสร้างขึ้นใหม่จาก 2 บิตข้างเคียง จากการดีมอดูเลต



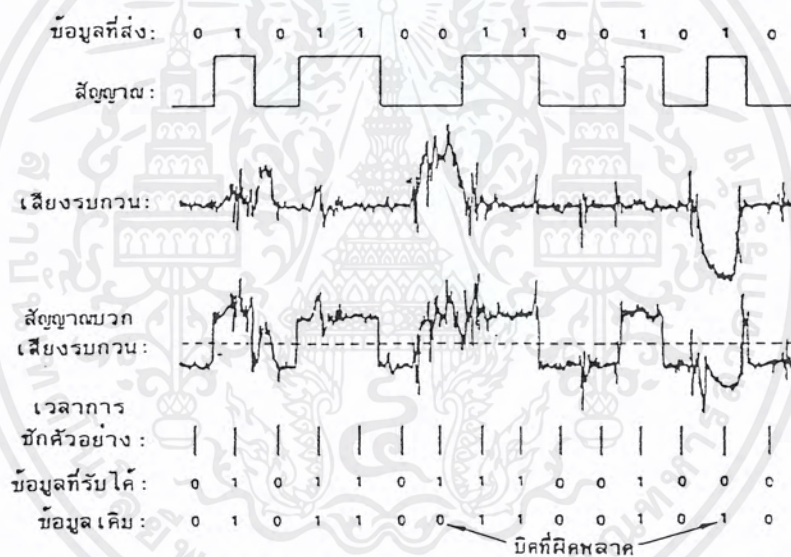
รูปที่ 2.48 การเกิดข้อผิดพลาดของบิต

2.5.5 ชนิดของสัญญาณ

สัญญาณที่ใช้ในการส่งข่าวสาร มีตั้งแต่รูปคลื่นไซน์ชอยคัลบริสุทธิ์เพียงความถี่เดียว สัญญาณคอมเพล็กซ์ที่ประกอบด้วยรูปคลื่นไซน์ชอยคัลหลายความถี่มารวมกันจนถึงรูปพัลส์ เป็นต้น ในบางกรณีอาจมีการจัดรูปพัลส์ให้อยู่ในรูปที่จะลดความเพี้ยน (distortion) ต่างๆ ที่อาจเกิดขึ้นในวงจรสื่อสารเมื่อการตอบสนอง (response) ของวงจรนั้นไม่คมพอที่จะให้พัลส์ผ่านได้โดยไม่เกิดความเพี้ยน ตัวอย่างเช่นพัลส์ชนิด เรสโคไซน์ (raised cosine) เป็นต้น

การเลือกสัญญาณที่มีความถี่หรือมีรูปร่างใดนั้น สามารถทำได้โดยให้ผ่านเครื่องกรองซึ่งจะกรองความถี่ที่ไม่ต้องการออกและเปลี่ยนแปลงรูปร่างของสัญญาณตามต้องการได้ สัญญาณที่มีค่าของขนาด (amplitude) เป็นค่าต่อเนื่อง เราเรียกว่าสัญญาณแอนาล็อก ส่วนสัญญาณที่มีขนาดเปลี่ยนแปลงเป็นค่าของตัวเลขลงตัว เราเรียกว่าสัญญาณดิจิทัล

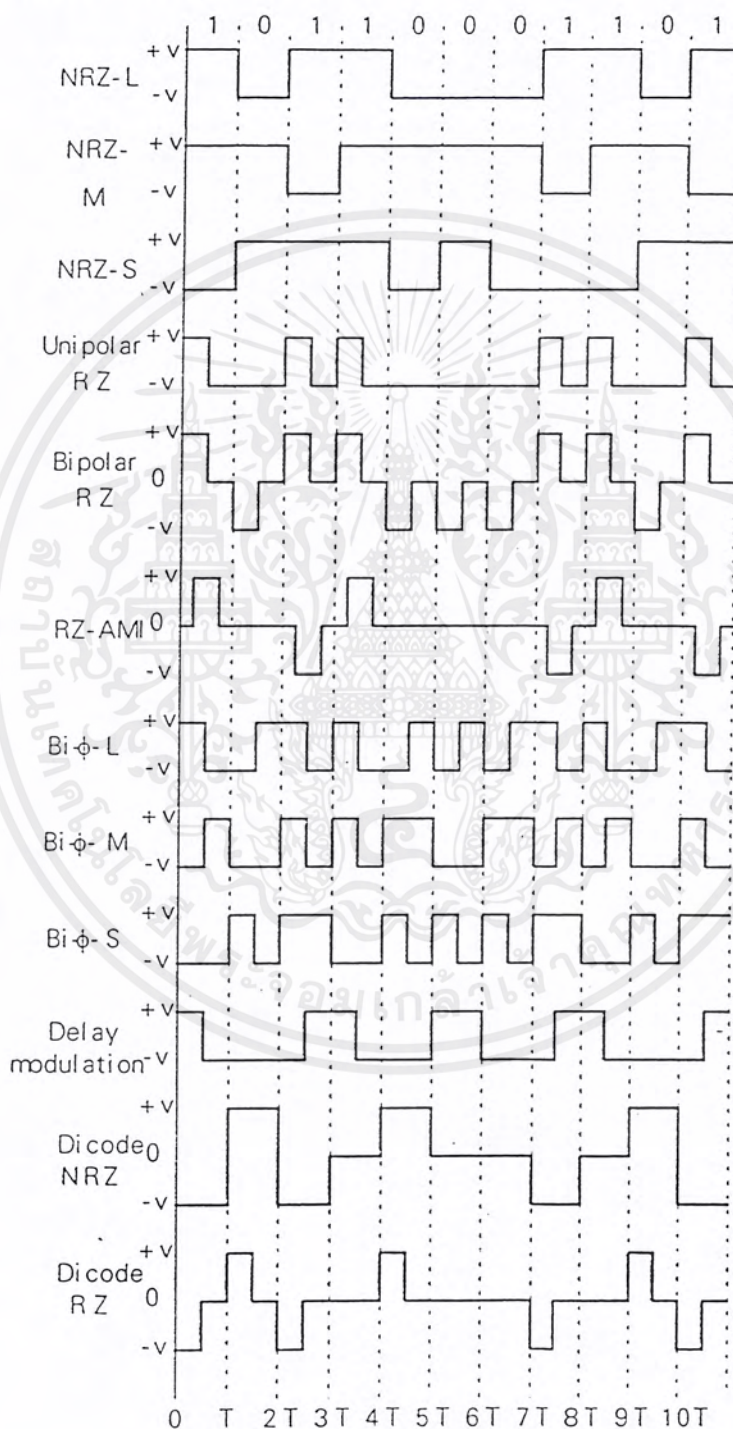
ปัจจุบันนี้ การสื่อสารนิยมใช้สัญญาณดิจิทัลแทนสัญญาณแอนาล็อกมากขึ้นทั้งนี้เพราะสัญญาณดิจิทัลทนต่อการถูกรบกวนโดยเสียงรบกวนได้ดีกว่าระบบแอนาล็อกดังรูป



รูปที่ 2.49 การรับส่งสัญญาณดิจิทัล

สัญญาณที่เป็นแอนาล็อกเมื่อถูกเสียงรบกวนแล้ว ทางภาครับนอกจากจะต้องแยกสัญญาณออกจากเสียงรบกวนแล้ว ยังจำเป็นต้องมีขนาดและความถี่เหมือนเดิมทุกประการ จึงไม่เกิดความเพี้ยนขึ้น ส่วนสัญญาณที่เป็นดิจิทัล แม้จะถูกเสียงรบกวนเนื่องจากเราต้องการทราบเพียงขนาดของพัลส์ว่าเป็น “ 1 ” หรือ “ 0 ” เท่านั้น จึงสร้างพัลส์เป็น “ 1 ” หรือ “ 0 ” ได้ง่าย โดยเทียบกับค่าจุดเริ่มเปลี่ยน (threshold) เท่านั้น ความผิดพลาดจะเกิดขึ้นเฉพาะกรณีที่เสียงรบกวนสามารถเปลี่ยนขนาดของสัญญาณดิจิทัลจาก “ 1 ” เป็น “ 0 ” หรือจาก “ 0 ” เป็น “ 1 ” เท่านั้นจึงมีโอกาสเกิดความผิดพลาดน้อยลง

สัญญาณดิจิทัลที่เป็นพัลส์อาจมีระดับขนาดอะไรก็ได้ไม่จำเป็นต้องเป็น “1” และ “0” เสมอไป แต่ถ้าเป็น “1” และ “0” เรียกว่าสัญญาณเลขฐานสองหรือสัญญาณไบนารี (binary signal) ซึ่งมีทั้งแบบเอ็นอาร์แซด (nonreturn-to-zero : NRZ) และแบบอาร์แซด (return-to-zero : RZ) นอกจากนี้ยังมีพัลส์ไบโพลาร์ ที่แทน “1” ด้วย “+1” หรือ “-1” สลับกันและแทน “0” ด้วย “0” นอกจากนี้ยังมีพัลส์แบบอื่นๆอีกมากที่ใช้ในการสื่อสารดังในรูป

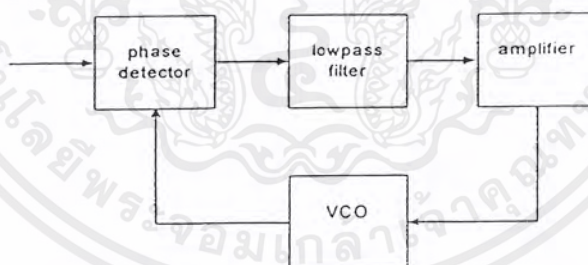


รูปที่ 2.50 แสดงการเข้ารหัสสัญญาณดิจิทัลในรูปแบบต่างๆ

ในโครงงานนี้สัญญาณดิจิทัลแบบสแตนด์บายจะกล่าวถึงสัญญาณแบบไม่กลับสู่ศูนย์แบบระดับ (Non return to zero-level : NRZ-L ซึ่งต่อไปจะใช้คำว่าเอ็นอาร์แซดแทน) เท่านั้น โครงงานนี้ใช้การเข้ารหัสแบบไม่กลับสู่ศูนย์แบบอินเวอร์ส (Non return to zero-invert : NRZ-I ซึ่งต่อไปจะใช้คำว่า เอ็นอาร์แซดไอแทน) เพื่อป้องกันการกลับเฟสที่ภากรับ หลักการเบื้องต้นที่สำคัญของการเข้ารหัสแบบ NRZ-I คือ การใช้การเปลี่ยนระดับ (transition) ของสัญญาณเป็นข้อมูลแทนการใช้ระดับแรงดันแทนข้อมูล โดยสัญญาณ output จะเปลี่ยนแปลงระดับทุกครั้งที่มีข้อมูล input มีระดับลอจิก “ 1 ” ซึ่งเรียกการเข้ารหัสแบบนี้ว่าสัญญาณเอ็นอาร์แซดแบบมาร์ค (Non return to zero-mark : NRZ-M) หรือจะให้สัญญาณเอาต์พุตเปลี่ยนแปลงระดับทุกครั้งที่มีสัญญาณอินพุตมีระดับลอจิกเป็น “ 0 ” ซึ่งเรียกการเข้ารหัสแบบนี้ว่าสัญญาณเอ็นอาร์แซดแบบสเปซ (Non return to zero-space : NRZ-S)

2.6 ทฤษฎีของเฟสล็อกคัล

เฟสล็อกคัลมีองค์ประกอบพื้นฐาน คือ เฟสดีเทคเตอร์ (Phase Detector : PD) โวลเตจคอนโทรลลอซซิลเลเตอร์ (Voltage – controlled oscillator : VCO) และวงจรกรองความถี่ต่ำผ่าน (LPF) ที่ต่อกันเป็นวงจรดังแสดงไว้ในรูปที่ 2.51 การทำงานของเฟสล็อกคัล อาจจะเริ่มที่เฟสดีเทคเตอร์ซึ่งเป็นส่วนทำการเปรียบเทียบความถี่สัญญาณเข้า (f_{in}) (ซึ่งเราจะถือว่าเป็นความถี่อ้างอิง : f_{ref}) กับความถี่ของสัญญาณที่ได้จากวงจรโวลเตจคอนโทรลลอซซิลเลเตอร์ (f_{VCO}) ซึ่งจะได้เอาต์พุตอยู่ในรูปของค่าความแตกต่างทางเฟส ซึ่งจะถูกรอง (และขยายสัญญาณ) เป็นสัญญาณไฟตรงเพื่อนำไปทำการควบคุมการผลิตความถี่ของโวลเตจคอนโทรลลอซซิลเลเตอร์ จนกระทั่งโวลเตจคอนโทรลลอซซิลเลเตอร์ ทำการล๊อคกับสัญญาณเข้า นั่นคือ f_{VCO} เท่ากับ f_{in} ด้วยค่าความแตกต่างทางเฟสคงที่ค่าหนึ่ง

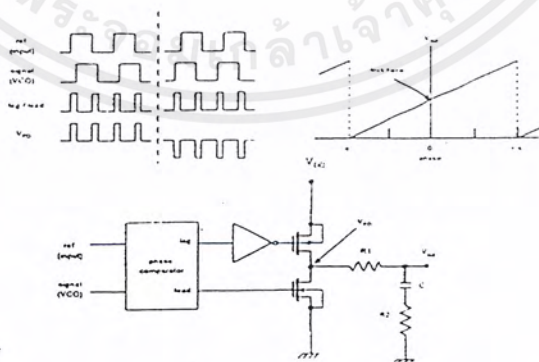


รูปที่ 2.51 บล็อกไดอะแกรมของเฟสล็อกคัล

เฟสดีเทคเตอร์ (PD) จะมีใช้กันสองแบบ คือ แบบ I และ แบบ II เฟสดีเทคเตอร์ แบบ I จะถูกออกแบบมาเพื่อทำการตรวจจับคลื่นสี่เหลี่ยม (square wave) ต่าง ๆ ทั้งทางอนาล็อกหรือทางดิจิทัล ในขณะที่ เฟสดีเทคเตอร์ แบบ II จะเพื่อทำการตรวจจับการเปลี่ยนแปลงสถานะ หรือขอบของสัญญาณ ตัวอย่างของวงจรรวมที่มี เฟสดีเทคเตอร์ แบบ I คือ 565 และ เฟสดีเทคเตอร์ แบบ II คือ TTL4044 ส่วน CMOS4046 จะมีเฟสดีเทคเตอร์ทั้งสองแบบอยู่ในตัว วงจร เฟสดีเทคเตอร์ แบบ I แบบง่าย ๆ ก็คือเอ็กคลูซีฟออร์เกต (Exclusive – OR gate) กับวงจรกรองความถี่ต่ำผ่าน ดังแสดงไว้ในรูปที่ 2.52 ซึ่งความสัมพันธ์

(duty cycle) 50% ได้แสดงไว้ทางขวามือของรูปดังกล่าว ซึ่งจะเห็นได้ว่า เฟสดีเทคเตอร์ แบบ I นี้ มีความสัมพันธ์เป็นเชิงเส้น สำหรับเฟสอินพุทในช่วงหนึ่ง ($0-\pi$) นี้ จะผลิตพัลส์ (pulse) บวกหรือลบ ขึ้นอยู่กับ การนำ (lead) หรือตาม (lag) ของสัญญาณโวลเตจคอนโทรลอสซิลเลเตอร์ เมื่อเปรียบเทียบกับสัญญาณอ้างอิง ความกว้างของพัลส์เหล่านี้จะเท่ากับช่วงเวลาระหว่างขอบสัญญาณอินพุททั้งสอง เฟสดีเทคเตอร์ และจะเป็นตัวกำหนดขนาดของแรงดันที่ป้อนให้ โวลเตจคอนโทรลอสซิลเลเตอร์ ข้อดีของเฟสดีเทคเตอร์ ชนิดนี้ คือการมีเอาต์พุทที่ไม่ขึ้นกับควิตซ์ไซเคิลของสัญญาณอินพุทและการไม่มีกระเพื่อมที่เอาต์พุท ส่วนเฟสดีเทคเตอร์ แบบ II จะทำการตรวจจับขอบของสัญญาณที่เข้ามายังตัวมัน ดังแสดงไว้ในรูปที่ 2.53 วงจรชนิดเฟสดีเทคเตอร์แบบ I และแบบ II เมื่อเฟสของสัญญาณอินพุททั้งสองทำการล๊อคกันแล้วเฟสดีเทคเตอร์ทั้งสองแบบได้เปรียบเทียบคุณสมบัติอยู่ตาราง ก)

รูปที่ 2.52 วงจรเฟสดีเทคเตอร์แบบ I และการทำงาน



รูปที่ 2.53 วงจรเฟสดีเทคเตอร์แบบ II และการทำงาน

ตาราง ก) คุณสมบัติของ เฟสดีเทกเตอร์ แบบ I และแบบ II

	เฟสดีเทกเตอร์ แบบ I	เฟสดีเทกเตอร์ แบบ II
Input duty cycle	50% optimum	-
Lock กับ harmonic	lock	ไม่ lock
Noise rejection	ดี	ไม่ดี
Residual ripple ที่ $2f_{in}$	มาก	น้อย
Lock range	ตลอดย่าน โวลเตจคอนโทรล ออสซิลเลเตอร์ range	ตลอดย่าน โวลเตจคอนโทรล ออสซิลเลเตอร์ range
Capture range	< Lock range	เท่ากับ Lock range
ความถี่ที่ไม่มีการ lock	f_{center}	f_{min}

โวลเตจคอนโทรลออสซิลเลเตอร์ เป็นส่วนที่จำเป็นสำหรับเฟสล็อกลูป ถือเป็นออสซิลเลเตอร์ ซึ่งผลิตความถี่ที่สามารถควบคุมได้จากเอาต์พุต ของเฟสดีเทกเตอร์หรือในทางทฤษฎีแล้ว โวลเตจคอนโทรลออสซิลเลเตอร์ก็คือวงจรรวมอินทิเกรเตอร์ (integrator) นั่นเอง โวลเตจคอนโทรลออสซิลเลเตอร์ ที่ใช้ในเฟสล็อกลูปไม่จำเป็นจะต้องมีความสัมพันธ์เป็นเชิงเส้นกับแรงดันที่มาควบคุมมากนัก ในกรณีทั่วไป แต่ถ้าเป็นมีความไม่เป็นเชิงเส้นมากเกินไปแล้ว ค่าลูปเกน (loop gain) ก็จะเปลี่ยนแปลงไปตามความถี่ของสัญญาณด้วย ฉะนั้นจึงจำเป็นต้องคำนึงถึงประสิทธิภาพของลูป (loop) ด้วย ตาราง ก) เป็นตาราง แสดงรายการของโวลเตจคอนโทรลออสซิลเลเตอร์ที่ใช้ในปัจจุบัน

ตาราง ข) รายการของ โวลเตจคอนโทรลออสซิลเลเตอร์ ที่ใช้ในปัจจุบัน

Type	Family	f_{max}	Output
566	Linear	1 MHz	Square, triangle
2206	Linear	0.5 MHz	Square, triangle, sine
2207	Linear	0.5 MHz	Square, triangle
4024	TTL	25 MHz	TTL
4046	CMOS	1 MHz	CMOS
8038	Linear	0.1 MHz	Square, triangle, sine
74LS124	TTL	20 MHz	TTL
74S124	TTL	60 MHz	TTL
74S324	TTL	20 MHz	TTL

2.6.1 การออกแบบ เฟสล็อกลูป

เฟสดีเทคเตอร์ ซึ่งให้ความผิดพลาดของสัญญาณ (error signal) ทางเฟสระหว่างสัญญาณอ้างอิงกับสัญญาณจากโวลเตจคอนโทรลลอซซิติเลเตอร์ และโวลเตจคอนโทรลลอซซิติเลเตอร์เป็นตัวแปลงความผิดพลาดของสัญญาณให้เป็นความถี่ ลักษณะเช่นนี้จะคล้ายกับวงจรขยายของออปแอมป์ที่มีความเสถียรภาพเมื่อมีการป้อนกลับด้วยเกนค่าหนึ่ง ต่างกันก็เพียงแต่ลักษณะของสัญญาณในออปแอมป์จะเป็นสัญญาณประเภทเดียวกันที่อินพุต และฟีดแบค (feedback Voltage) แต่ใน เฟสล็อกลูป จะมีสัญญาณเฟสเป็นอินพุตและความถี่เป็นส่วนป้อนกลับ เนื่องจากเฟสคืออินทิกรัล (integral) ของความถี่ ดังนั้นฟีดแบค ลูป (feedback loop) จะมีเฟสชิฟ 90° แลกกิง (lagging) เพื่อเป็นการรักษาความเสถียรภาพของฟีดแบค จึงไม่ควรเพิ่มแลกกิงเฟสชิฟคอมโพเนนต์ (lagging phase shift components) เข้าไปในส่วนฟีดแบคอีก วงจรดังกล่าวจะเป็นเฟสล็อกลูป แบบออร์เดอร์ลูปอันดับ 1 (first – order loop : FOL) ซึ่งแสดงไว้ในรูปที่ 2.54 ก) ถึงแม้ว่าแบบออร์เดอร์ลูปอันดับ 1 จะมีโครงสร้างแบบง่าย ๆ แต่มันก็มีข้อเสียคือ ความถี่ที่ผลิตจาก โวลเตจคอนโทรลลอซซิติเลเตอร์ (และการดีเทค (detect) ของเฟส) จะมีความคลาดเคลื่อนมาก ปัญหานี้สามารถลดไปได้โดยเพิ่ม วงจรกรองความถี่ต่ำผ่านลงไปในลูป แล้ววงจรจะเป็นแบบออร์เดอร์ลูปอันดับ 2 (second – order loop : SOL) ซึ่งแสดงไว้ในรูปที่ 2.54 ข) นอกจากนี้แล้วยังช่วยลดแคปเจอร์เรจ (capture range) และเพิ่มแคปเจอร์ไทม์ (capture time) โดยเฉพาะอย่างยิ่งถ้าใช้เฟสดีเทคเตอร์แบบ II แล้ว การล็อกของเฟสจะเกิดขึ้นเมื่อค่าความแตกต่างทางเฟสเป็นศูนย์จริง ๆ ดังนั้นแบบออร์เดอร์ลูปอันดับ 2 จึงเป็นที่นิยมใช้ในงานทั่ว ๆ ไป



รูปที่ 2.54 ก) เฟสล็อกลูปแบบออร์เดอร์ลูปอันดับหนึ่งและ ข) เฟสล็อกลูปแบบออร์เดอร์ลูปอันดับ 2

บัฟเฟอร์อินพุตเอาต์พุต เป็นส่วนที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอก เพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม

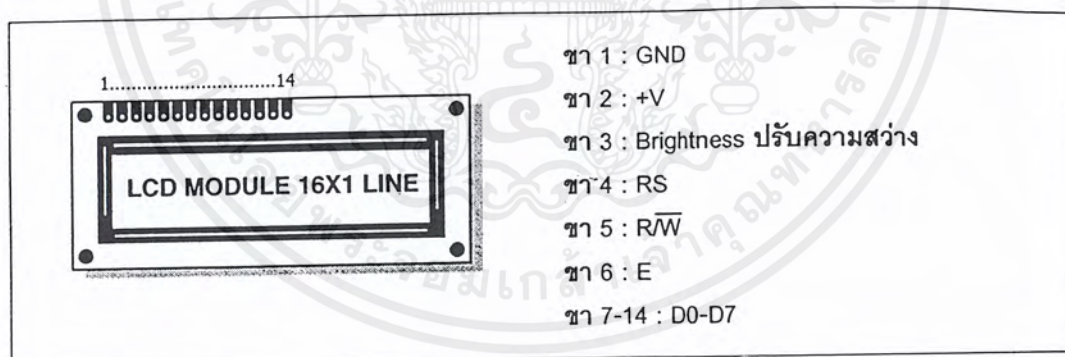
รีจิสเตอร์คำสั่ง (Instruction Register : IR) เป็นรีจิสเตอร์ใช้รับข้อมูลคำสั่งจากอุปกรณ์ภายนอกเพื่อนำไปควบคุมการแสดงผล

รีจิสเตอร์ข้อมูล (Data Register : DR) เป็นรีจิสเตอร์ใช้รับข้อมูลจากอุปกรณ์ภายนอกเพื่อถ่ายทอดไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลแสดงผล หรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในแรมเก็บตัวอักษร

แรมเก็บข้อมูลแสดงผล (Display Data Ram : DDRAM) เป็นหน่วยความจำแรมทำหน้าที่เก็บข้อมูลที่เข้ามาจากรีจิสเตอร์ DR ตัวควบคุมจะนำข้อมูลใน DDRAM นี้ไปเปิดตาราง (Look up - table) ของตัวอักษรที่เก็บไว้ในหน่วยความจำรวมและแรมเก็บตัวอักษร เพื่อนำไปแสดงที่ตัวแสดงผล

รวมเก็บตัวอักษร (Character Generator ROM : CGROM) เป็นหน่วยความจำรวมที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่สามารถอ่านออกไปแสดงที่ตัวแสดงผลได้ มีขนาด 7,200 บิต โดยจะถูกอ่านด้วยค่าของข้อมูลใน DDRAM

แรมเก็บตัวอักษร (Character Generator RAM : CGRAM) เป็นหน่วยความจำแรมที่ใช้เก็บอักษรที่มีการสร้างเพิ่มเติมขึ้นใหม่ ในกรณีที่ตัวอักษรใน CGROM ไม่เพียงพอ มีขนาด 512 บิต การเขียนและอ่านค่าไปใช้นั้นทำได้เช่นเดียวกับ CGROM คือ เขียนข้อมูลลงใน DDRAM แล้วตัวควบคุมจะมาอ่านค่าจาก CGRAM เอง



รูปที่ 2.56 รูปร่างและการจัดขาโมดูล LCD แบบอักษร

แฟล็ก BUSY เป็นส่วนที่ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่า ตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งหรือไม่ ดังนั้นก่อนการส่งข้อมูลหรือคำสั่งมายังตัวควบคุมต้องตรวจสอบสถานะของแฟล็ก BUSY นี้เสียก่อน

2.7.2 โมดูล LCD ขนาด 16 ตัวอักษร 1 บรรทัด (LCD 16×1)

สำหรับโมดูล LCD ที่ยกมาใช้ในการเรียนรู้ในการทดลอง เป็นขนาด 16 ตัวอักษร 1 บรรทัด เนื่องจากราคาถูก ง่าย และเป็นโมดูล LCD ที่มีโครงสร้างเป็นมาตรฐาน มีผู้ผลิตหลายราย และมีการระบุเบอร์แตกต่างกันออกไปตามผู้ผลิต อาทิ LM020L ของฮิตาชิ, DMC – 16117A ของคอปเท็กซ์ (Optrex) เป็นต้น แต่อย่างไรก็ตาม คอนโทรลเลอร์ที่ใช้คือเบอร์เดียวกันนั้นคือเบอร์ HD44750 ของฮิตาชิ

โมดูล LCD ขนาด 16×1 มีขาต่อใช้งานทั้งสิ้น 14 ขา มีการจัดขาตั้งในรูปแบบที่ 2.56 สำหรับรายละเอียดการทำงานของแต่ละขามีดังนี้

V_{SS} (ขา 1) : ต่อกราวด์

V_{DD} (ขา 2) : ต่อไฟเลี้ยง + 5 โวลต์

V_O (ขา 3) : เป็นขาอินพุตรับแรงดันเพื่อปรับความเข้มของการแสดงผล

RS (ขา 4) : เป็นขาอินพุตใช้ในการแยกชนิดของข้อมูลที่ต้องการประมวลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจิสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจิสเตอร์ DR โดยถ้าขานี้เป็น “0” ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าขานี้เป็น “1” ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล

$R/\overline{W}$ (ขา 5) : เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับ LCD ถ้าเป็น “0” เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น “1” จะเป็นการอ่านข้อมูล

E (ขา 6) : เป็นขาอินพุตเปิด LCD ให้ทำงาน

D0 – D7 (ขา 7 – 14) : เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่าง LCD กับอุปกรณ์ภายนอก ขนาด 8 บิต

2.7.3 คำสั่งควบคุมโมดูล LCD

ในการเขียนคำสั่งลงในตัวควบคุม แน่นอนว่าต้องกำหนดให้ขา RS และ $R/\overline{W}$ เป็น “0” แล้วเขียนคำสั่งตามไป คำสั่งควบคุมโมดูล LCD ของชิปควบคุม HD44780 ที่สำคัญมี 10 คำสั่งดังนี้

1. คำสั่งเคลียร์ตัวแสดงผล (clear display)

มีข้อมูลคำสั่งเป็น 01H เป็นคำสั่งที่ใช้เขียนข้อมูลช่องว่าง หรือ space เข้าไปใน DDRAM ทั้งหมด เมื่อตัวควบคุมเอ็กซิกิวต์คำสั่งนี้ จะทำการกำหนดแอดเดรสของ DDRAM เป็น 0 เคอร์เซอร์จะกลับไปอยู่ที่ตำแหน่งซ้ายมือสุดของจอแสดงผล แล้วเซตบิต I/O (ซึ่งจะกล่าวถึงภายหลัง) ให้เป็น “1”

2. คำสั่ง return home

ต้องกำหนดให้บิต 1 ของข้อมูลเป็น “1” เป็นคำสั่งให้เคอร์เซอร์เคลื่อนที่กลับไปยังตำแหน่งซ้ายสุดของจอแสดงผล แต่ข้อมูลบนจอแสดงผลไม่เปลี่ยนแปลง นั่นคือ ข้อมูลคำสั่งของคำสั่งนี้จะเป็น 02H หรือ 03H ก็ได้

3. คำสั่งเลือกโหมดการป้อนข้อมูล (Entry mode Set)

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	0	1	I/D	S

บิต S เป็นบิตที่ใช้ในการกำหนดลักษณะของการแสดงผล เมื่อมีการป้อนข้อมูล ถ้าหากบิต S เป็น “1” เมื่อเกิดข้อมูลใหม่บนจอแสดงผล ตัวเคอร์เซอร์จะอยู่กับที่ แต่ตัวอักษรข้อมูลเดิมจะถูกดันไปทางซ้าย แต่ถ้าหากบิตนี้เป็น “0” เมื่อเกิดข้อมูลใหม่ ตัวเคอร์เซอร์จะเลื่อนไปทางขวามือ

บิต I/D เป็นบิตที่ใช้ในการกำหนดว่า เมื่อเขียนหรืออ่านข้อมูลแล้ว ทำให้แอดเดรสของ DDRAM เพิ่มขึ้นหรือลดลงหนึ่งแอดเดรส โดยถ้าบิตนี้เป็น “1” แอดเดรสของ DDRAM จะเพิ่มขึ้น แต่ถ้าเป็น “0” แอดเดรสจะลดลง

ดังนั้น ข้อมูลคำสั่งที่เกิดขึ้นสำหรับคำสั่งนี้ได้แก่ 04H – 07H (4 ข้อมูลคำสั่ง) และที่ใช้บ่อยคือ 06H หมายถึง กำหนดให้เมื่อเกิดข้อมูลใหม่ เคอร์เซอร์จะเลื่อนไปทางขวามือ และแอดเดรสของ DDRAM เพิ่มขึ้น

4. คำสั่งควบคุมการแสดงผล

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	0	1	D	C	B

บิต D ใช้ควบคุมการเปิดปิดจอแสดงผล ถ้าบิตนี้เป็น “1” จะเป็นการเปิดจอแสดงผล ถ้าเป็น “0” จะเป็นการปิดจอแสดงผล

บิต C ใช้ควบคุมการแสดงผลตัวเคอร์เซอร์บนจอแสดงผล ถ้าต้องการให้มีเคอร์เซอร์แสดงผลบนจอแสดงผล ต้องกำหนดให้บิตนี้เป็น “1” ถ้ากำหนดให้เป็น “0” จะเป็นการปิดเคอร์เซอร์ หรือไม่แสดงเคอร์เซอร์

บิต B ใช้ควบคุมการกะพริบของเคอร์เซอร์ ถ้าบิตนี้เป็น “1” เคอร์เซอร์จะกะพริบ

ดังนั้นจะมีข้อมูลคำสั่งได้ตั้งแต่ 08H – 0FH (8 รูปแบบคำสั่ง) ที่ใช้บ่อยคือ 0CH เป็นการสั่งให้เปิดจอแสดงผล แต่ไม่แสดงเคอร์เซอร์ และ 0FH เป็นการสั่งให้เปิดจอแสดงผล แสดงเคอร์เซอร์ และสั่งให้เคอร์เซอร์กะพริบ

5. คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	0	1	S/C	R/L	*	*

การควบคุมการเลื่อนเคอร์เซอร์และตัวอักษรบนจอแสดงผลขึ้นอยู่กับการกำหนดบิต S/C และ R/L ซึ่งสามารถสรุปได้ดังนี้

S/C	R/L	ลักษณะการเลื่อน	ข้อมูลคำสั่ง
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย	10H – 13H
0	1	เลื่อนเคอร์เซอร์ไปทางขวา	14H – 17H
1	0	เลื่อนตัวอักษรใหม่ไปทางซ้าย	18H – 1BH
1	1	เลื่อนตัวอักษรใหม่ไปทางขวา	1C – 1FH

6. คำสั่งกำหนดฟังก์ชันการทำงาน

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิต 7	บิต 6	บิต 5	บิต 4	บิต 3	บิต 2	บิต 1	บิต 0
0	0	1	DL	N	F	*	*

บิต D/L ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูล ถ้าบิตนี้เป็น “ 0 ” จะเป็นการติดต่อแบบ 4 บิต แต่ถ้าเป็น “ 1 ” จะเป็นแบบ 8 บิต

บิต N ใช้กำหนดจำนวนบรรทัดของการแสดงผล ถ้าเป็น “ 0 ” จะแสดงผล 1 บรรทัด ถ้าเป็น “ 1 ” จะแสดงผล 2 บรรทัด ในกรณีที่จอแสดงผลสามารถแสดงได้มากกว่า 2 บรรทัด และต้องการให้แสดงผลมากกว่า 2 บรรทัด ก็กำหนดบิต N นี้ให้เป็น “ 1 ”

บิต F ใช้เลือกความละเอียดของตัวอักษรในการแสดงผล ถ้าบิตนี้เป็น “ 0 ” จะเป็นการแสดงผลแบบ 5×7 จุด และถ้าเป็น “ 1 ” จะแสดงผลเป็นแบบ 5×10 จุด

ข้อมูลคำสั่งที่ใช้บ่อยคือ 38H เป็นการกำหนดให้โมดูล LCD ทำงานในแบบ 8 บิต แสดงผล 2 บรรทัด และเลือกความละเอียดเป็น 5×7 จุด

จุดที่น่าสังเกตคือ โมดูล LCD แบบ 16 ตัวอักษร 1 บรรทัด แม้จะมีบรรทัดการแสดงผลเพียง 1 บรรทัด แต่จะต้องกำหนด N ให้เป็น “ 1 ” เนื่องจากแอดเดรสของ DDRAM แบ่งเป็น 2 ช่วง คือ 00H และ 40H

7. คำสั่งเลือกแอดเดรสของ CGRAM

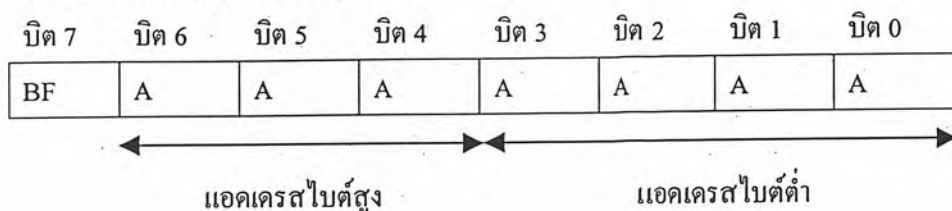
เมื่อต้องการกำหนดแอดเดรสของ CGRAM ต้องกำหนดให้บิต 7 เป็น “ 0 ” บิต 6 เป็น “ 1 ” ส่วนอีก 6 บิตที่เหลือจะแทนด้วยค่าแอดเดรสของ CGRAM จะต้องทำการกำหนดแอดเดรสด้วยคำสั่งนี้ก่อนที่จะอ่านหรือเขียนข้อมูลให้ CGRAM โดยแอดเดรสของ CGRAM อยู่ระหว่าง 00H – 3FH

8. คำสั่งเลือกแอดเดรสของ DDRAM

ใช้ในการเลือกแอดเดรสของ DDRAM ก่อนที่จะทำการอ่านหรือเขียนข้อมูล โดยบิตที่ 7 ต้องเป็น “ 1 ” และข้อมูลอีก 7 บิตที่เหลือจะเป็นค่าแอดเดรสของ DDRAM ซึ่งแอดเดรสของ DDRAM จะอยู่ระหว่าง 8CH – 0FFH ทั้งนี้จำนวนแอดเดรสนี้ขึ้นกับการกำหนดสถานะที่บิต N ด้วย หากบิต N เป็น “ 0 ” แอดเดรสของ DDRAM จะอยู่ระหว่าง 80H – 0CFH และถ้าบิต N เป็น “ 1 ” แอดเดรสของ DDRAM จะมี 2 ช่วงคือ 8CH – 87H และ 0C0H – 0C7H

9. คำสั่งอ่านแฟลก BUSY และแอดเดรส

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้



เป็นคำสั่งที่ใช้อ่านแฟลก BUSY (BF) โดยแฟลคนี้จะเป็นตัวบอกสถานะของตัวควบคุม LCD ว่าพร้อมจะรับข้อมูลอยู่หรือไม่ ถ้าหากบิต BF เป็น “ 0 ” แสดงว่าตัวควบคุม LCD พร้อมรับข้อมูลหรือคำสั่ง แต่ถ้าเป็น “ 1 ” แสดงว่า ขณะนี้ตัวควบคุม LCD ยังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผลข้อมูลอยู่ ยังไม่พร้อมรับข้อมูลหรือคำสั่ง

เมื่อต้องการอ่านแฟลกต้องกำหนดให้ขา $R/\overline{W}$ เป็น “ 1 ” ด้วย แต่สัญญาณที่ RS ยังต้องเป็น “ 0 ” อยู่ เพราะข้อมูลนี้เป็นข้อมูลคำสั่ง

นอกจากนี้ ยังใช้เป็นคำสั่งอ่านข้อมูลแอดเดรสของ CGRAM และ DDRAM ด้วย โดยบิต 0 – บิต 6 เป็นค่าข้อมูลของแอดเดรสที่ต้องการอ่าน

2.7.4 การเขียนคำสั่งและข้อมูลให้แก่โมดูล LCD

ในการเขียนข้อมูลเพื่อควบคุมโมดูล LCD แสดงผลตามที่ผู้ใช้งานต้องการ ต้องส่งคำสั่ง (instruction) แล้วกำหนดโหมดการทำงานให้แก่โมดูล LCD ก่อน จากนั้นจึงค่อยส่งข้อมูล (data) ที่ต้องการแสดงผลเนื่องจากบัสข้อมูลของโมดูล LCD มี 8 เส้น คือ D0 – D7 และใช้เป็นทางผ่านของทั้งคำสั่งและข้อมูล ดังนั้น ในการส่งคำสั่งและข้อมูลจึงต้องอาศัยการกำหนดสัญญาณลอจิกที่ขา RS ถ้าหากที่ขา RS ได้ลอจิก “ 0 ” หมายความว่า ข้อมูลที่ป้อนให้แก่โมดูล LCD ขณะนั้นเป็นคำสั่ง ในทางตรงข้าม หากขา RS ได้รับลอจิก “ 1 ” ข้อมูลที่ป้อนให้ขณะนั้นเป็นข้อมูลที่ใช้ในการแสดงผล

เมื่อต้องการเขียนหรืออ่านข้อมูลใน CGRAM และ DDRAM เริ่มต้นต้องกำหนดแอดเดรสที่ต้องการอ่านหรือเขียนก่อน โดยใช้คำสั่งเลือกแอดเดรส จากนั้นกำหนดให้ขา RS เป็น “ 1 ” เพื่อแจ้งให้ตัวควบคุมภายในโมดูล LCD ทราบว่าข้อมูลที่ปรากฏต่อไปนี้เป็นข้อมูลปกติไม่ใช่คำสั่ง

ในกรณีที่ต้องการอ่านข้อมูลต้องกำหนดให้ขา $R/\overline{W}$ เป็น “ 1 ” ข้อมูลขนาด 8 บิต (หรือ 4 บิต) ก็ปรากฏบนบัสข้อมูล โดยข้อมูลที่อ่านออกมาได้จะเป็นข้อมูลจากแอดเดรสของ CGRAM หรือ DDRAM ตามต้องการ

ในกรณีที่ต้องการเขียนข้อมูล เมื่อกำหนดแอดเดรสและป้อนลอจิก “ 1 ” ให้ขา RS แล้ว แล้วต้องกำหนดให้ขา $R/\overline{W}$ เป็น “ 0 ” ข้อมูลที่อยู่บนบัสข้อมูลจะถูกเขียนลงในรีจิสเตอร์ DR จากนั้นจึงถ่ายทอดลงใน DDRAM ต่อไป

2.7.5 จังหวะการทำงานของ LCD โมดูล

ในการติดต่อกับโมดูลของ LCD จะต้องมีการหน่วงเวลาหลังจากที่ทำการส่งรหัสคำสั่งหรือข้อมูล เนื่องจากต้องรอให้คอนโทรลเลอร์ภายใน LCD โมดูล แปลความหมายของรหัสคำสั่งและทำงานตามคำสั่งให้เรียบร้อยก่อน จากนั้นจึงจะรับข้อมูลหรือดำเนินการต่อไป

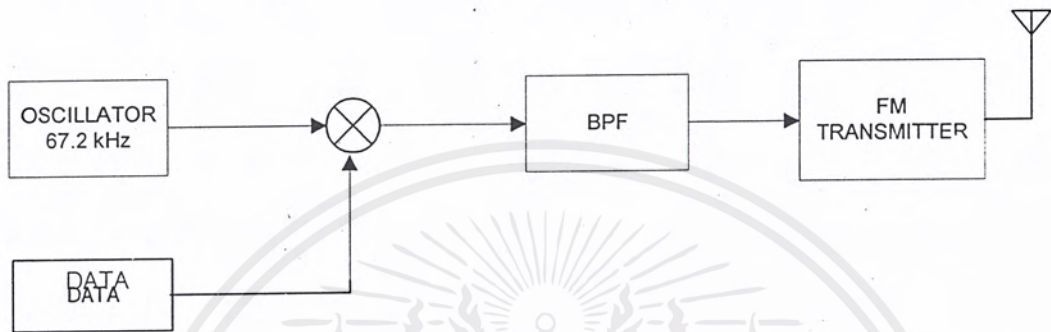
ดังนั้น ในการใช้งานโมดูล LCD ผู้เขียนโปรแกรมต้องมีโปรแกรมเพื่อหน่วงเวลารอให้โมดูล LCD พร้อมทำงานด้วย โดยเมื่อเริ่มจ่ายไฟให้แก่โมดูล LCD ต้องรอประมาณ 10 มิลลิวินาที เพื่อให้โมดูล LCD ทำการเตรียมความพร้อมหรืออินิเชียล (initial) หลังจากนั้นก็จะกำหนดลอจิกให้แก่ขา RS ของโมดูล LCD แล้วต้องหน่วงเวลาอีกประมาณ 2 มิลลิวินาทีเพื่อให้คอนโทรลเลอร์ใน LCD โมดูลแปลความหมายของลอจิกที่ขา RS ว่า ข้อมูลต่อไปที่จะได้รับนั้นเป็นรหัสคำสั่งหรือเป็นข้อมูลที่ต้องการแสดงผล จากนั้นจะเป็นการส่งข้อมูลมารอที่บัสข้อมูล D0-D7 (กรณีทำงานในโหมด 8 บิต) ขั้นตอนต่อไปจะเป็นการส่งสัญญาณพัลส์ไปที่ขา E เพื่ออีนาเบิลโมดูล LCD ให้รับข้อมูลจากบัสข้อมูลเข้าไป โดยพัลส์ที่ป้อนเข้าที่ขา E ของโมดูล LCD ต้องเป็นพัลส์ขอบขาขึ้น จากนั้นทำการหน่วงเวลา 2 มิลลิวินาที

ทั้งหมดที่กล่าวมาคือขั้นตอนและจังหวะในการทำงาน 1 รอบของโมดูล LCD จะเห็นได้ว่ามีโปรแกรมย่อยที่สำคัญอยู่ 3 โปรแกรมย่อยคือ โปรแกรมอินิเชียล LCD, โปรแกรมหน่วงเวลา และโปรแกรมย่อยการส่งพัลส์เพื่ออีนาเบิลโมดูล LCD

บทที่ 3

การคำนวณและการสร้าง

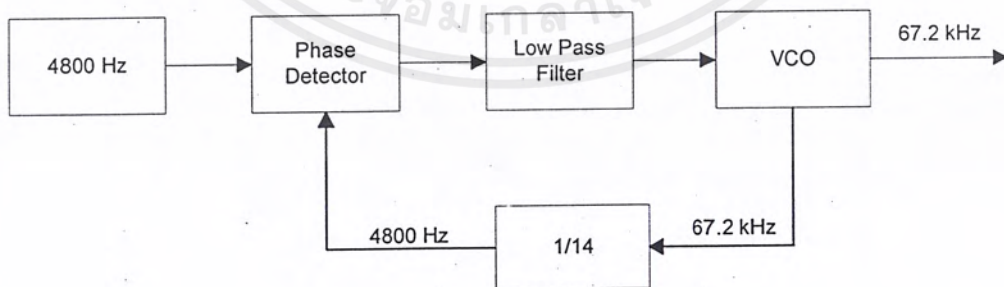
3.1 การสร้างวงจรภาคส่ง



รูปที่ 3.1 บล็อกไดอะแกรมของวงจรภาคส่ง

การทำงานของภาคส่งสัญญาณอินพุตจะถูกมอดูเลตกับสัญญาณคลื่นพาห์ สำหรับสัญญาณคลื่นพาห์นั้นใช้ความถี่ 67.2 กิโลเฮิร์ต เพราะว่าความถี่กึ่งกลางของช่องสัญญาณเอฟเอ็ม เอสซีเอ (FM SCA) โดยการคูณสัญญาณข่าวสารเข้ากับสัญญาณคลื่นพาห์ เพื่อกำจัดสัญญาณคลื่นพาห์ไม่ให้ปรากฏที่ด้านเอาต์พุต หลังจากนั้นนำสัญญาณไปผ่านวงจรแบนด์พาสฟิลเตอร์ซึ่งมีความถี่กึ่งกลางที่ 67.2 กิโลเฮิร์ตและมีแบนด์วิธประมาณ 11.2 กิโลเฮิร์ต หลังจากนั้นนำสัญญาณไปมอดูเลตทางความถี่เพื่อส่งออกอากาศต่อไป

3.1.1 วงจรผลิตสัญญาณความถี่ 67.2 กิโลเฮิร์ต



รูปที่ 3.2 บล็อกไดอะแกรมของวงจรผลิตสัญญาณความถี่ 67.2 กิโลเฮิร์ต

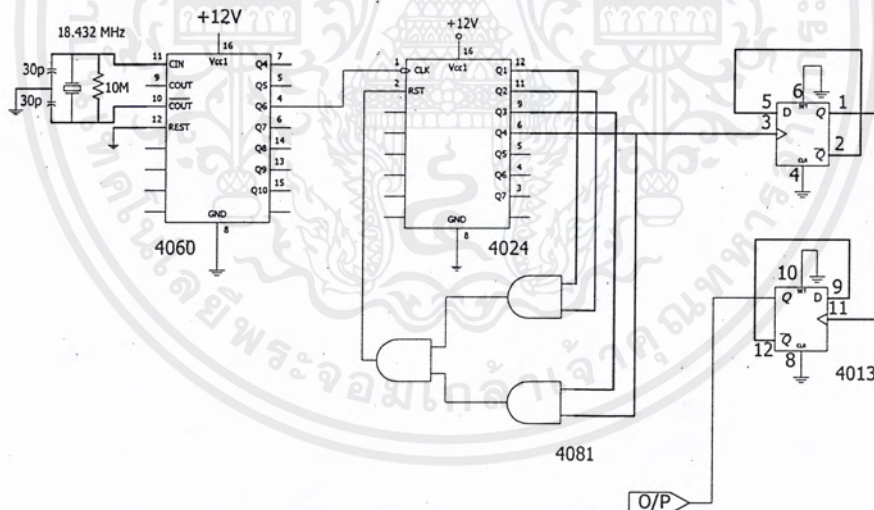
วงจรผลิตสัญญาณความถี่ 67.2 กิโลเฮิร์ต ประกอบด้วยวงจรผลิตสัญญาณความถี่ 4800 เฮิร์ต ซึ่งใช้เป็นความถี่อ้างอิง และวงจรเฟสล็อกกลุ๊ป

3.1.1.1 วงจรผลิตสัญญาณอ้างอิงความถี่ 4800 เฮิร์ต



รูปที่ 3.3 บล็อกไดอะแกรมวงจรกำเนิดสัญญาณ 4800 เฮิร์ต

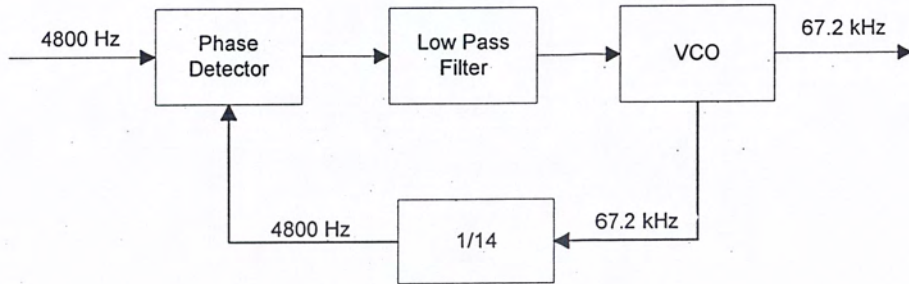
ใช้คริสตัลอสซิลเลเตอร์ที่ผลิตความถี่ 18.432 เมกะเฮิร์ต มาหารความถี่จนได้สัญญาณที่มีความถี่ 4800 เฮิร์ตตามต้องการ



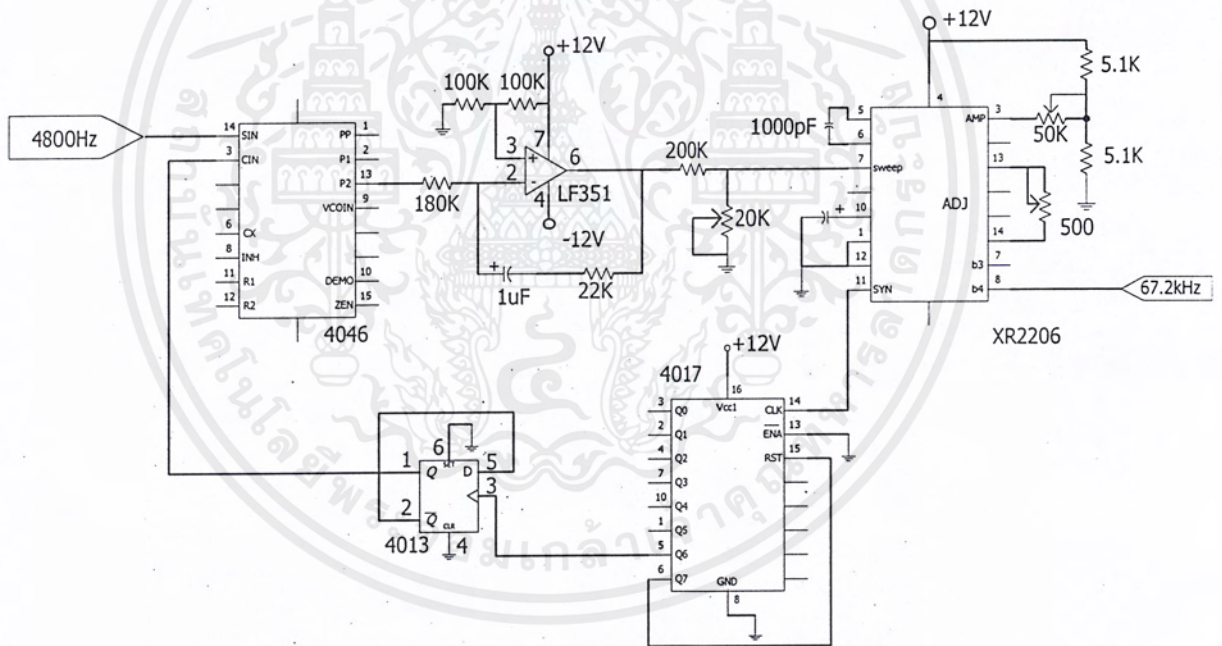
รูปที่ 3.4 วงจรกำเนิดสัญญาณความถี่ 4800 เฮิร์ต

จากรูปที่ 3.4 ความถี่ 18.432 เมกะเฮิร์ตจะเข้าไปยังขา 10 และ 11 ของวงจรรวม 4060 ซึ่งทำหน้าที่เป็นวงจรหาร 64 จากนั้นผ่านวงจรหาร 15 ซึ่งประกอบด้วยวงจรรวม 4024 และแอนด์เกตจำนวนหนึ่ง ได้สัญญาณเอาต์พุตที่ขา 5 ของวงจรรวม 4024 นำสัญญาณที่ได้ไปหาร 4 ซึ่งประกอบด้วยวงจรดีฟลิปฟล็อป 2 ตัว จะได้เอาต์พุตความถี่ 4800 เฮิร์ตซึ่งจะนำไปใช้เป็นสัญญาณอ้างอิงสำหรับวงจรเฟสล็อกกลุ๊ปต่อไป

3.1.1.2 วงจรเฟสล็อกกลุ่



รูปที่ 3.5 บล็อกไดอะแกรมวงจรเฟสล็อกกลุ่

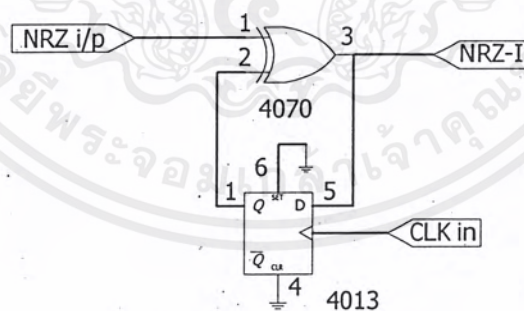


รูปที่ 3.6 วงจรสร้างความถี่ 67.2 กิโลเฮิรท์

สัญญาณความถี่อ้างอิง 4800 เฮิรท์ และสัญญาณที่ผลิตโดยวงจร โวลเตจคอนโทรลอสซิลเลเตอร์ (Voltage Control Oscillator) ที่ผ่านวงจรหารแล้วจะถูกนำมาเปรียบเทียบกับทางเฟส โดยวงจรรวม 4046 ซึ่งทำหน้าที่เป็นเฟสดีเทคเตอร์ เอาท์พุทที่ได้เมื่อผ่านวงจรกรองความถี่ต่ำจะได้สัญญาณเป็นไฟกระแสตรงที่จะไปควบคุมความถี่เอาท์พุทของวงจรรวม XR2206 ซึ่งทำหน้าที่เป็นวงจร โวลเตจคอนโทรลอสซิลเลเตอร์ ซึ่งเป็นวงจรที่ความถี่ทางด้านเอาท์พุทจะเปลี่ยนแปลงตามแรงดันที่เข้ามา เพื่อผลิตสัญญาณความถี่ 67.2 กิโลเฮิรท์ตามต้องการ

3.1.2 วงจรเข้ารหัสสัญญาณเอ็นอาร์แซดแบบมาร์ค (Non Return to Zero – Mark)

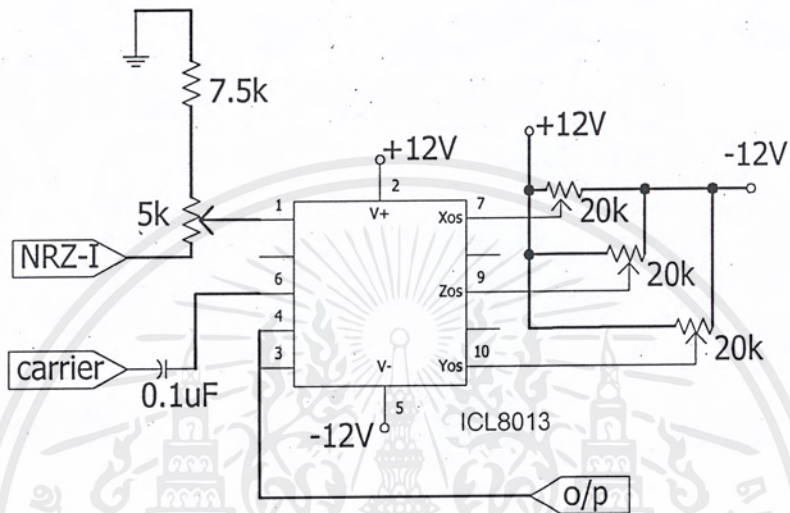
สำหรับสาเหตุที่ต้องเข้ารหัสสัญญาณเบสแบนด์ (Baseband) จากการเข้ารหัสแบบเอ็นอาร์แซด (Non Return to Zero) เป็นการเข้ารหัสแบบเอ็นอาร์แซดแบบอินเวอร์ท (Non Return to Zero – Invert) ก็เพื่อป้องกันการสับสนทางเฟส (Phase ambiguity) ซึ่งเป็นผลจากการที่คลื่นพาหะจากวงจรกำเนิดสัญญาณผิดจาก $\cos(\omega_c t)$ เป็น $-\cos(\omega_c t)$ ผลจะทำให้การตีความข้อมูลผิดไปจากความเป็นจริง แต่ถ้าหากใช้การเข้ารหัสแบบเอ็นอาร์แซดแบบอินเวอร์ท ซึ่งใช้การเปลี่ยนระดับของสัญญาณแทนการส่ง “0” หรือ “1” จะสามารถแก้ปัญหาเนื่องจากการสับสนทางเฟสได้ โดยได้เลือกใช้การเข้ารหัสแบบเอ็นอาร์แซดแบบมาร์ค (Non Return to Zero – Mark) ซึ่งจะมีการเปลี่ยนระดับสัญญาณทุกครั้งที่มีลอจิก (Logic) “1” เข้ามา



รูปที่ 3.7 วงจรเข้ารหัสสัญญาณแบบเอ็นอาร์แซดแบบมาร์ค

3.1.3 วงจรคูณสัญญาณพาหะกับสัญญาณข่าวสาร

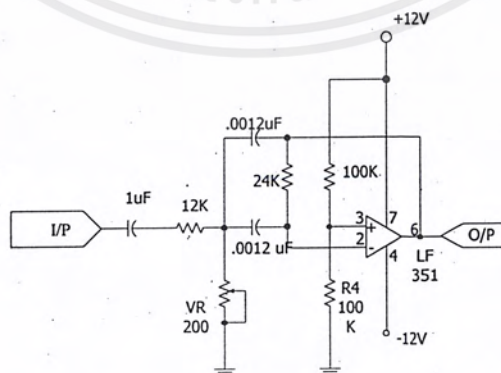
ใช้วงจรรวม ICL8013 โดยสัญญาณข่าวสารจะถูกป้อนเข้าที่ขา 1 ของไอซีและสัญญาณคลื่นพาหะป้อนเข้าที่ขา 6 ของวงจร ได้สัญญาณเอาต์พุตที่ขา 4 ของวงจรจะเป็นสัญญาณพีเอมแอสเค โดยทำการต่อวงจรดังรูป



รูปที่ 3.8 วงจรคูณสัญญาณพาหะกับสัญญาณข่าวสาร

3.1.4 การออกแบบวงจรกรองความถี่แบบคัตออฟฟิลเตอร์

สำหรับวงจรแบบคัตออฟฟิลเตอร์ จะเลือกใช้วงจรมัลติเพิล ฟีดแบค ออร์เดอร์ 2 (Multiple feedback order 2) ที่มีการจัดวงจรดังรูป



รูปที่ 3.9 วงจรกรองความถี่แบบคัตออฟฟิลเตอร์

ถ้าให้ Q เป็นค่า Quality factor ของวงจร
 ω_0 เป็นความถี่ศูนย์กลาง
 H เป็นเกนของวงจร

ทรานเฟอร์ฟังก์ชันของวงจรแสดงได้คือ

$$\frac{V_o(S)}{V_i(S)} = \frac{H(\omega/Q)S}{S^2 + (\omega_0/Q)S + \omega_0^2}$$

$$\omega_0^2 = \frac{1}{R_3 C_1 C_2} \left(\frac{1}{R_1} + \frac{1}{R_2} \right)$$

$$Q = \frac{1}{\sqrt{\frac{1}{R_3} \left(\frac{1}{R_1} + \frac{1}{R_2} \right) \left(\sqrt{\frac{C_1}{C_2}} + \sqrt{\frac{C_2}{C_1}} \right)}}$$

$$H = \frac{-R_3 C_3}{R_1 (C_1 + C_2)}$$

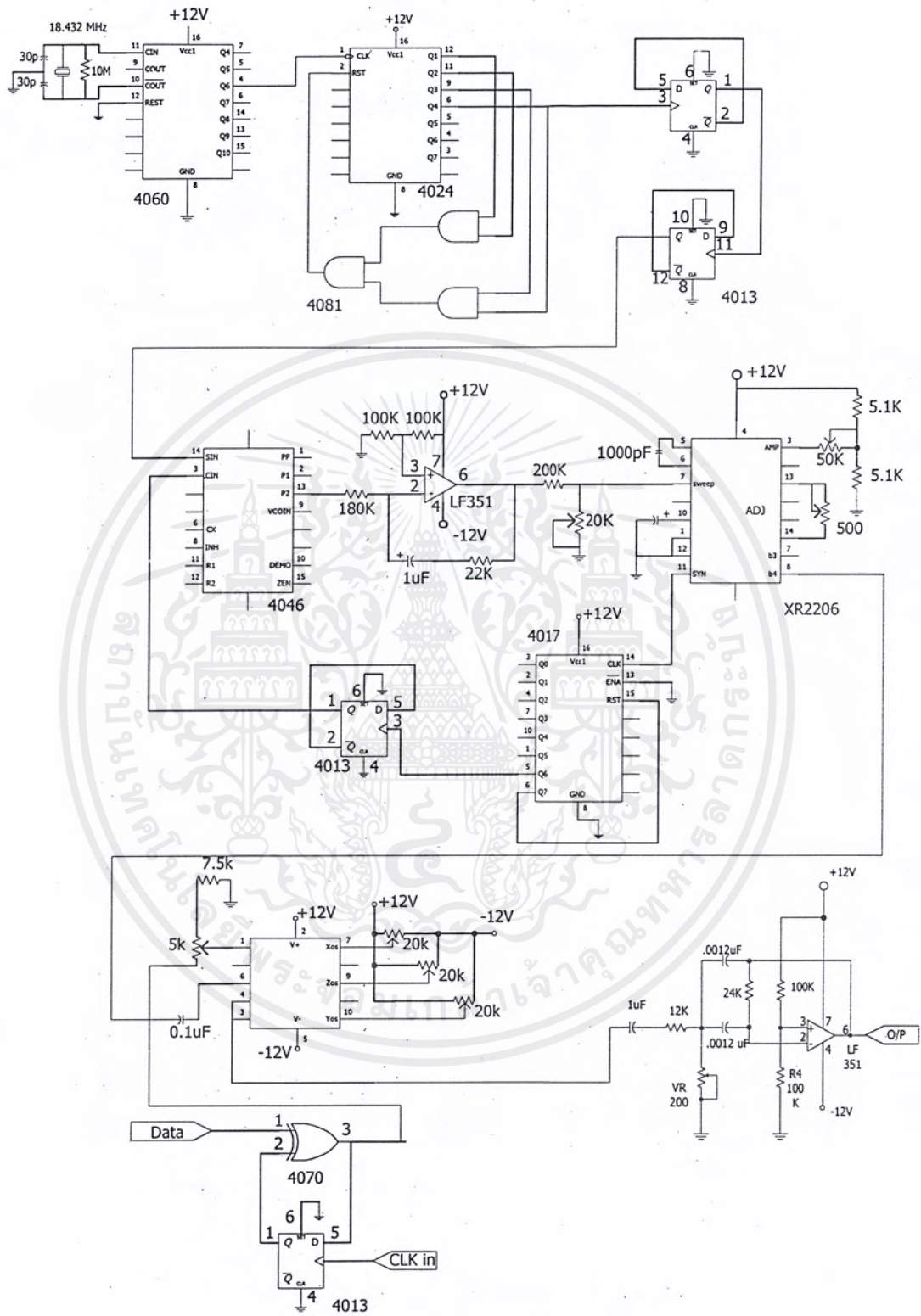
สำหรับการออกแบบจะกำหนดค่า Q , ω_0 , H และกำหนดค่า C_1, C_2 โดยให้ $C_1 = C_2$ ค่า R_1, R_2 และ R_3 หาได้จาก

$$R_1 = \frac{Q}{|H|\omega_0 C_1}$$

$$R_2 = \frac{Q}{(2Q^2 - |H|)\omega_0 C_1}$$

$$R_3 = \frac{2Q}{\omega_0 C_1}$$

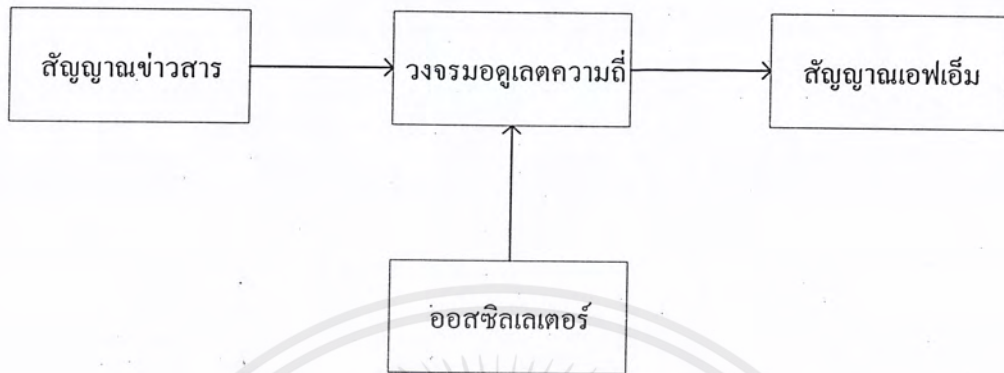
ความถี่ที่ต้องการคือ 67.2 กิโลเฮิรต์ กำหนดอัตราขยายที่ความถี่ศูนย์กลางมีค่าเท่ากับ 1 สำหรับขนาดของแบนด์วิธต้องให้มีขนาดมากกว่า 2 เท่าของความถี่พหุของสัญญาณเบสแบนด์ ดังนั้นค่า Q ของวงจรจึงต้องน้อยกว่า 7 ในการออกแบบเลือกใช้ค่า Q มีขนาดเท่ากับ 6 โดยขนาดของแบนด์วิธจะประมาณเท่ากับ 11.2 กิโลเฮิรต์ ค่าความต้านทานและค่าตัวเก็บประจุของวงจรหาได้จากสมการข้างบน



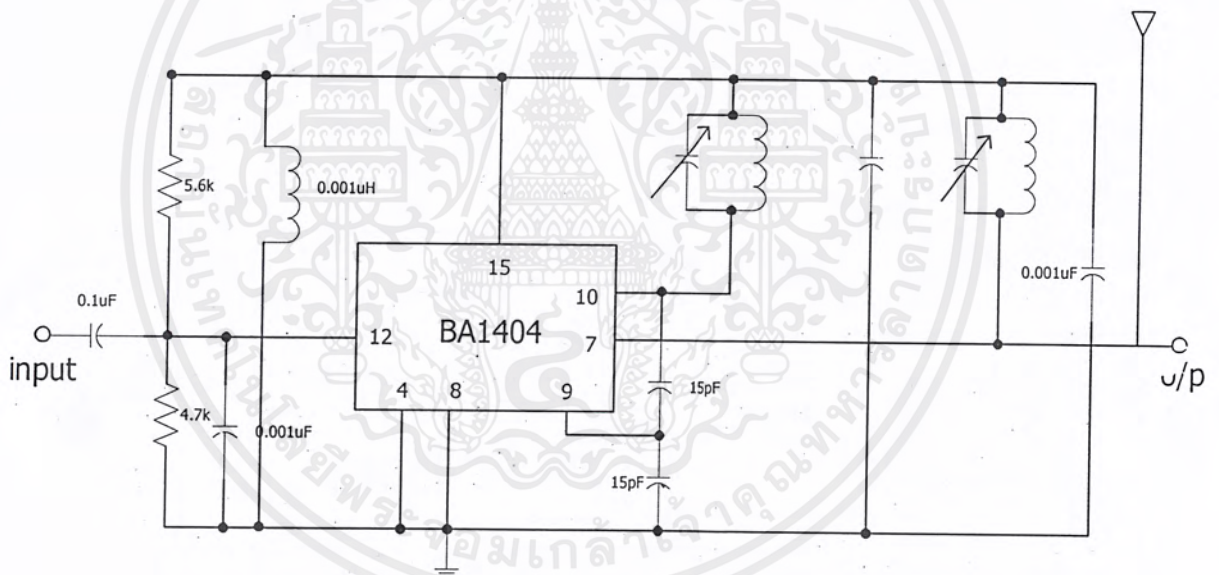
รูปที่ 3.10 แสดงวงจรมอดูลเลขสัญญาณพีเอสเคทั้งหมด

3.1.5 วงจรเครื่องส่งเอฟเอ็ม

หลักการทำงานของวงจรมอดูเลตความถี่



รูปที่ 3.11 แสดงหลักการทำงานของวงจรมอดูเลตความถี่



รูปที่ 3.12 แสดงรูปวงจรเครื่องส่งเอฟเอ็ม

วงจรเครื่องส่งเอฟเอ็มสร้างขึ้นจากวงจรรวมเบอร์ BA1404 ซึ่งเป็นไอซีที่ทำหน้าที่ในการมอดูเลตความถี่ที่สามารถมอดูเลตได้ทั้งระบบเอฟเอ็มโมโนและเอฟเอ็มสเตอริโอ

จากรูป สัญญาณข่าวสารจะเป็นอินพุตของไอซี (ขา 12) ซึ่งจะนำไปมอดูเลตความถี่กับคลื่นพาห์ที่กำหนดมาจากออสซิลเลเตอร์ภายในตัวไอซี ซึ่งสามารถจูนความถี่ของออสซิลเลเตอร์ได้จากการเปลี่ยนแปลงค่าตัวเก็บประจุหรือตัวเหนี่ยวนำที่ต่ออยู่กับขา 10 ของไอซี สัญญาณเอฟเอ็มที่เอาต์พุตของไอซี (ขา 7) จะผ่านวงจรเรโซแนนซ์อีกชุดหนึ่งคือชุดตัวเหนี่ยวนำและตัวเก็บประจุที่ต่ออยู่กับขา 7 ของไอซี เพื่อจูนความถี่ของสัญญาณเอฟเอ็มเพื่อส่งต่อไปยังวงจรอื่น

การคำนวณความถี่เรโซแนนซ์สามารถคำนวณได้จากสมการ

$$f_c = \frac{1}{2\pi\sqrt{LC}}$$

กำหนดให้ค่าความถี่กลางของสัญญาณเอฟเอ็มของเครื่องควบคุมมีค่าเป็น 49 เมกกะเฮิร์ต ค่าความเหนี่ยวนำ (L) มีค่า 1 ไมโครเฮนรี

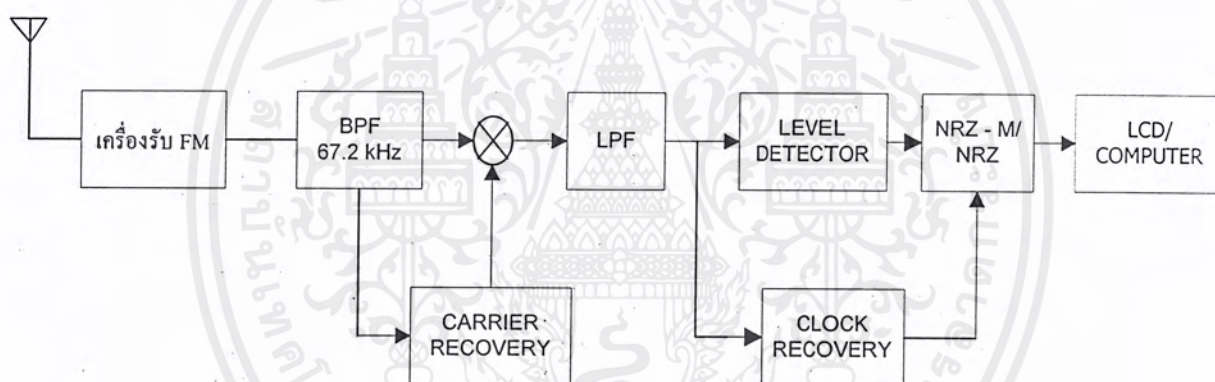
แทนค่าในสมการจะได้

$$49 \times 10^6 = \frac{1}{2\pi\sqrt{(1 \times 10^{-6})C}}$$

จะได้ค่าตัวเก็บประจุเป็น 10.55 พิโคฟารัด

ดังนั้นจึงใช้ค่าตัวเหนี่ยวนำ 1 ไมโครเฮนรีและตัวเก็บประจุ 10 พิโคฟารัด (ปรับค่าได้ตั้งแต่ 0-10 พิโคฟารัด) และด้วยวิธีการเดียวกันจะได้ค่าตัวเหนี่ยวนำที่ต่อที่ขา 7 เป็น 1 ไมโครเฮนรีและตัวเก็บประจุมีค่าเป็น 10 พิโคฟารัด (ปรับค่าได้)

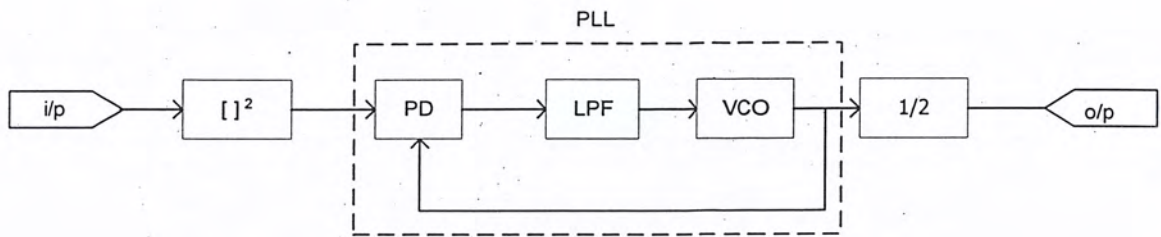
3.2 การสร้างวงจรภาคเครื่องรับ



รูปที่ 3.13 แสดงบล็อกไดอะแกรมของวงจรภาครับ

การทำงานของภาคเครื่องรับ วงจรเครื่องรับเอฟเอ็มจะรับเอาสัญญาณที่ด้านส่งมา โดยวงจรกรองความถี่ผ่าน 67.2 กิโลเฮิร์ตจะกรองเอาสัญญาณข่าวสารที่ด้านส่งส่งมาด้วยความถี่คลื่นพาห์ 67.2 กิโลเฮิร์ต สัญญาณแอมพลิจูดของวงจรกรองความถี่ผ่านจะนำไปเป็นสัญญาณอินพุตของวงจรกู้ความถี่คลื่นพาห์ เพื่อกู้สัญญาณคลื่นพาห์ที่ซึ่งโครโมโซมกับด้านส่งที่ด้านรับ สัญญาณพีเอสเคที่รับได้จะนำไปคูณกับสัญญาณคลื่นพาห์ที่ถูกกลับคืนมาได้ เพื่อให้ได้ข่าวสารกลับคืนมา เนื่องจากด้านส่งมีการเข้ารหัสสัญญาณเอ็นอาร์แซดเป็นเอ็นอาร์แซดไอ ดังนั้นที่ด้านรับจึงต้องมีการถอดรหัสสัญญาณเพื่อให้ได้ข่าวสารที่ด้านส่งกลับคืนมา นำข่าวสารที่ผ่านการถอดรหัสแล้วแสดงผลที่หน้าจอแอลซีดี

3.2.1 วงจรกู้ความถี่คลื่นพาห้

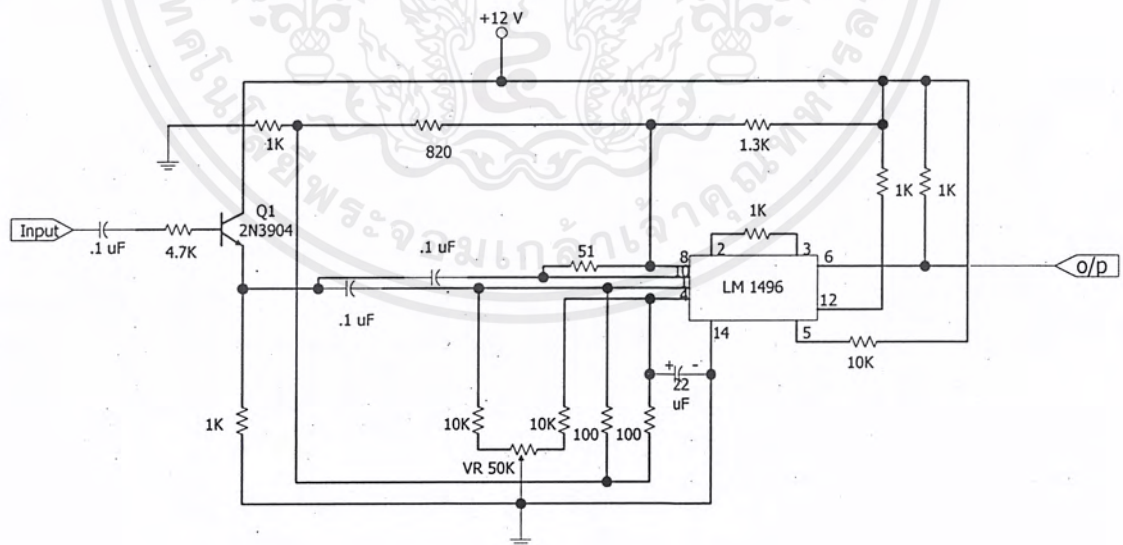


รูปที่ 3.14 แสดงบล็อกไดอะแกรมวงจรกู้ความถี่คลื่นพาห้

การทำงานของลูกบาศก์กำลังสองเมื่อรับสัญญาณอินพุตพีเอสเคซึ่งมีเฟส $\cos \omega_c t$ หรือ $-\cos \omega_c t$ ขึ้นอยู่กับสัญญาณเบสแบนด์ วงจรยกกำลังสองทำหน้าที่ยกกำลังสองของสัญญาณคลื่นพาห้โดยเอาต์พุตของวงจรยกกำลังสองจะได้ฮาร์โมนิกที่ 2 ของคลื่นพาห้ หรือ $\cos 2\omega_c t$ ซึ่งจะใช้เป็นสัญญาณอ้างอิงสำหรับวงจรเฟสล็อกkup วงจรเฟสล็อกkup จะล็อกเฉพาะสัญญาณ $\cos 2\omega_c t$ ซึ่งเป็นอินพุตของวงจรหารสอง เมื่อหารความถี่แล้วจะได้สัญญาณเป็น $\cos \omega_c t$ สำหรับเป็นสัญญาณอ้างอิงในการดีมอดูเลตสัญญาณพีเอสเคต่อไป

3.2.2 วงจรยกกำลังสอง

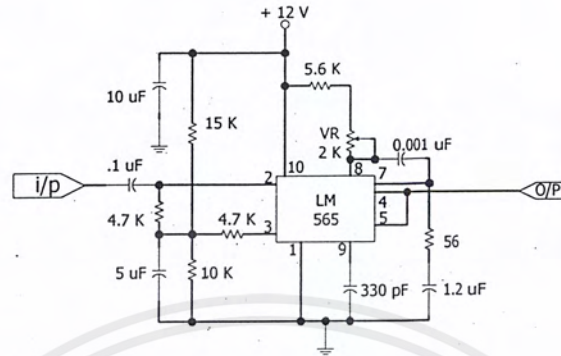
ใช้วงจรรวม LM1496 ซึ่งเป็นวงจรรวมสำหรับคูณสัญญาณ โดยสัญญาณอินพุตทั้งสองของวงจรต่อเข้ากับสัญญาณที่ต้องการยกกำลังสอง



รูปที่ 3.15 แสดงวงจรยกกำลังสองสัญญาณ

3.2.3 วงจรเฟสล็อกคูล

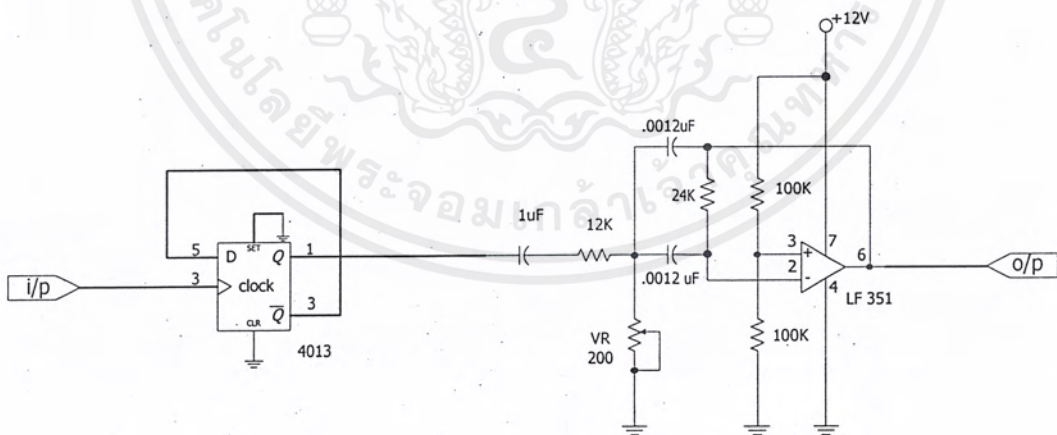
ทำหน้าที่ล็อกสัญญาณความถี่ 2 เท่าของสัญญาณคลื่นพาห์ 67.2 กิโลเฮิร์ต หรือ $\cos 2\omega_c t$ โดยใช้ วงจรรวม LM565 ดังรูป



รูปที่ 3.16 แสดงวงจรเฟสล็อกคูลสำหรับรูปยกกำลังสอง

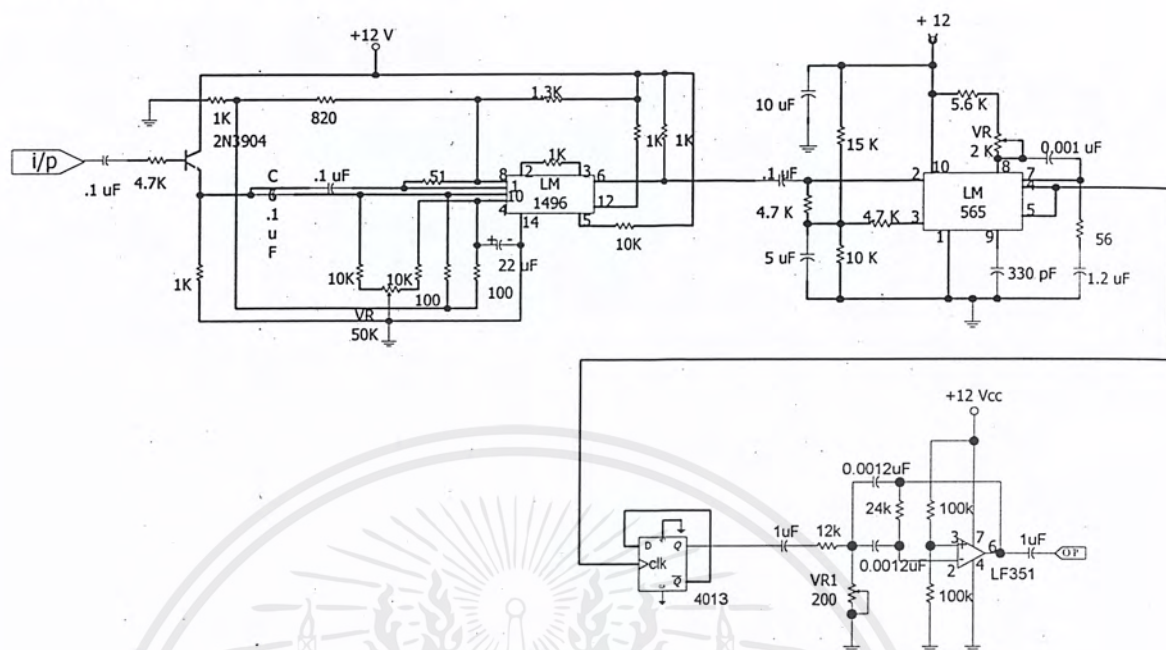
3.2.4 วงจรหารสอง

ใช้วงจรรวม 4013 ซึ่งมีดีฟลิป ฟลอปอยู่ภายในสำหรับเป็นวงจรหารสอง เอาต์พุตของสัญญาณ จากดี ฟลิปฟลอปจะได้สัญญาณความถี่ 67.2 กิโลเฮิร์ต เป็นสัญญาณรูปสี่เหลี่ยมซึ่งใช้เป็นอินพุตของวงจรกรองความถี่ผ่าน 67.2 กิโลเฮิร์ตที่มีแบนด์วิธแคบมากเพื่อให้ได้เอาต์พุตเป็นสัญญาณคลื่นพาห์ที่เป็นรูปสัญญาณไซน์เพื่อนำไปใช้สำหรับวงจรคุณภาพถัดไป โดยแสดงวงจรได้ดังรูป



รูปที่ 3.17 แสดงวงจรหารสอง

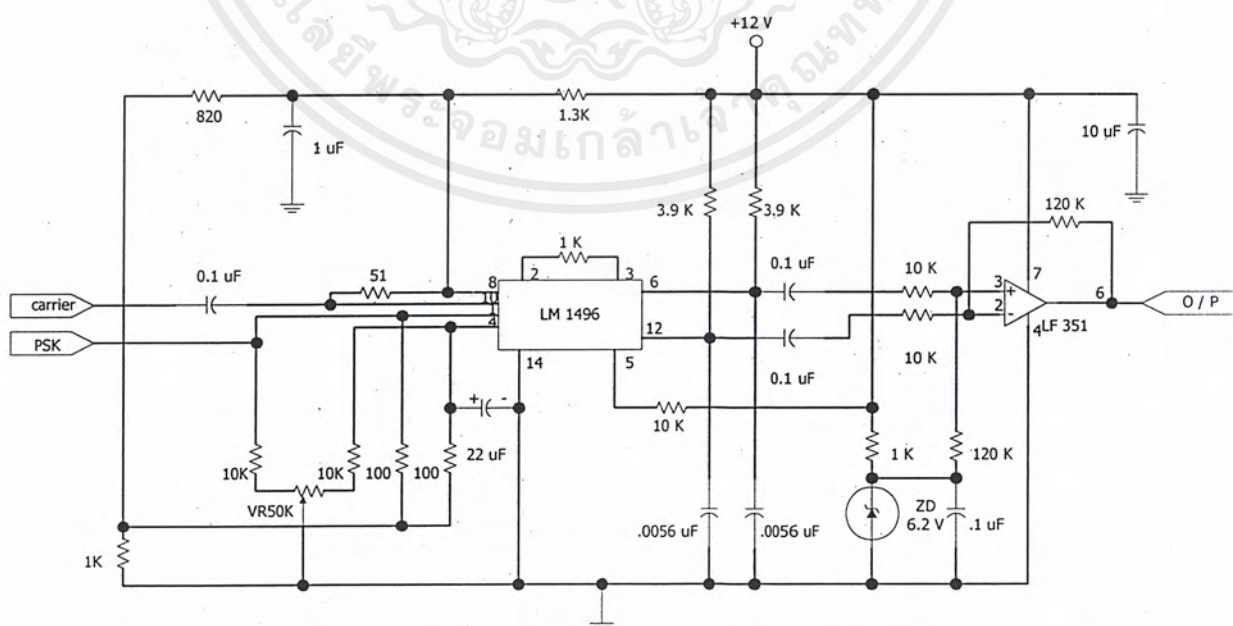
วงจรสัญญาณคลื่นพาห้ทั้งหมดแสดงได้ดังรูป



รูปที่ 3.18 แสดงวงจรสัญญาณคลื่นพาห้แบบลูบยกกำลังสอง

3.2.5 วงจรคูณสัญญาณคลื่นพาห้กับสัญญาณพีเอสเค

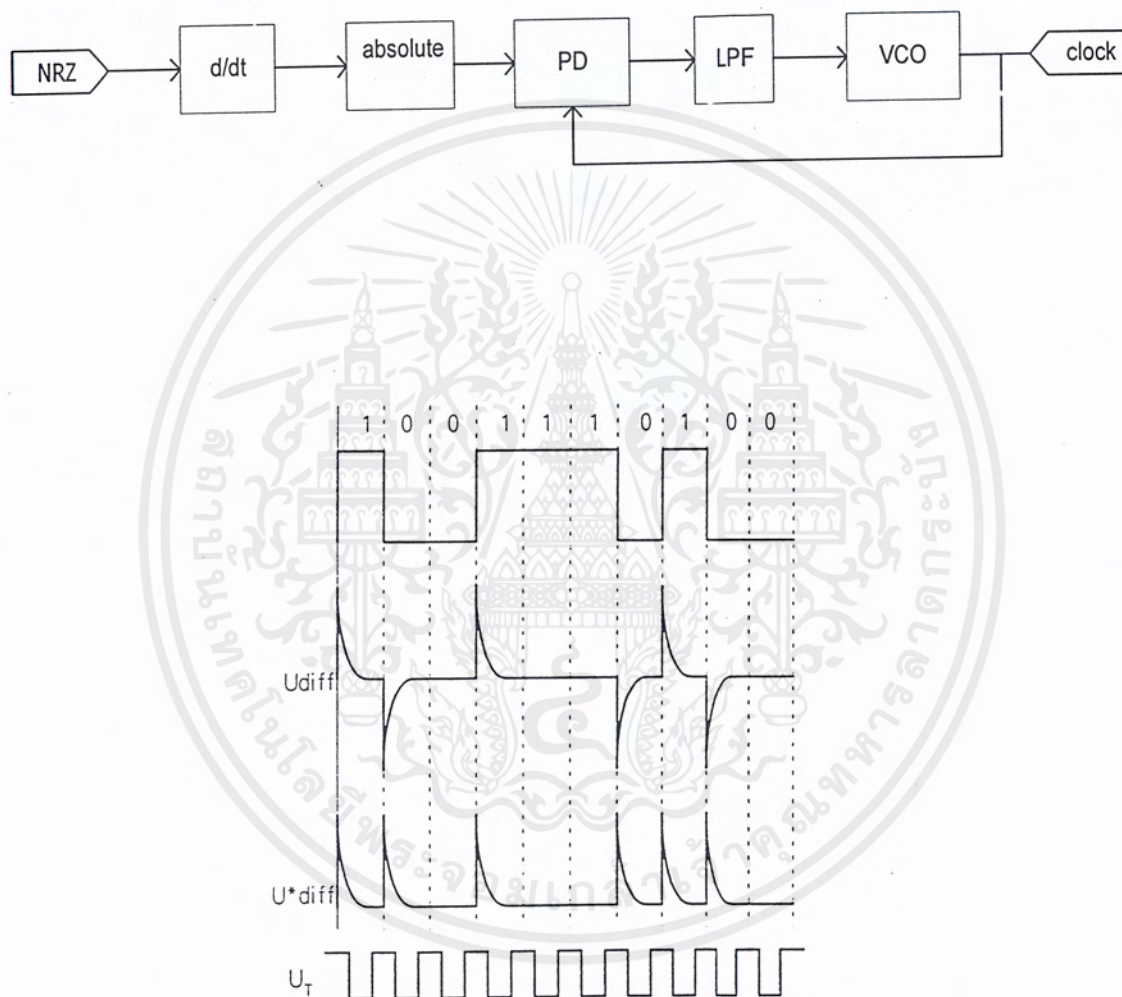
วงจรคูณสัญญาณคลื่นพาห้กับสัญญาณพีเอสเคนี้ใช้วงจรรวม LM1496 เช่นเดียวกับวงจรยกกำลังสอง โดยป้อนสัญญาณอินพุตที่ขา 1 เป็นสัญญาณพีเอสเค และที่ขา 10 ป้อนคลื่นพาห้จากวงจรคูณคลื่นพาห้ สัญญาณความถี่สูงจะถูกคัปปลิงลงกราวด์ เอาต์พุตของวงจรคูณเป็นสัญญาณเบสแบนด์ที่จะนำไปเป็นอินพุตของวงจรสัญญาณนาฬิกาและเพื่อข้อมูลที่ส่งมาจากเครื่องส่ง โดยมีวงจรดังรูป



รูปที่ 3.19 แสดงวงจรคูณสัญญาณคลื่นพาห้กับสัญญาณพีเอสเค

3.2.6 วงจรกู้สัญญาณนาฬิกา

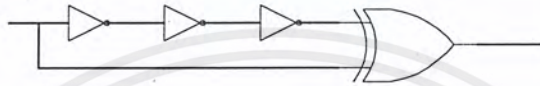
วงจรกู้สัญญาณนาฬิกาทำหน้าที่สร้างสัญญาณนาฬิกาที่ภาครับที่มีความถี่ 4.800 กิโลเฮิร์ต โดยที่สัญญาณนาฬิกาที่ภาครับซึ่งโครโมสกับสัญญาณนาฬิกาของภาคส่ง ซึ่งสามารถกู้สัญญาณนาฬิกาได้โดยอาศัยหลักการดังรูป



รูปที่ 3.20 แสดงหลักการพื้นฐานของวงจรกู้สัญญาณนาฬิกาจากสัญญาณเอ็นอาร์แซด (NRZ)

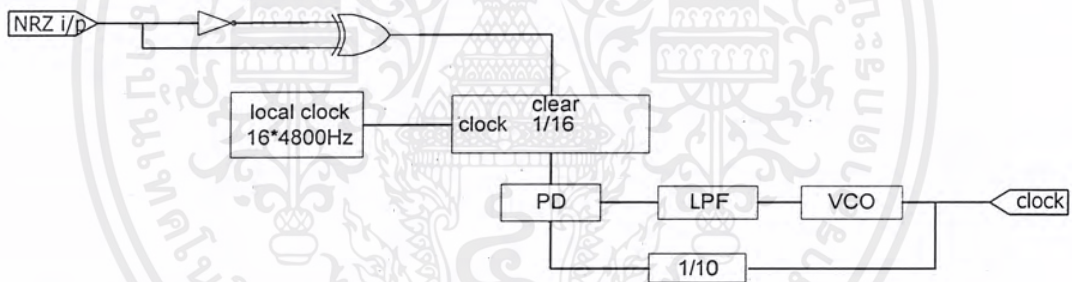
สัญญาณที่ผ่านการดีมอดูเลทแล้วจะเป็นอินพุทของวงจรดิฟเฟอเรนทิเอท โดยสัญญาณที่มีค่าคงที่เมื่อ ดิฟแล้วจะได้เป็นค่าความชันเป็นบวกหรือลบขึ้นอยู่กับสัญญาณที่เข้ามา ซึ่งจะใช้เป็นอินพุทของวงจรค่าตัวคูณ เอาต์พุทของวงจรค่าตัวคูณนี้จะมีส่วนประกอบของสัญญาณนาฬิกาปะปนอยู่ เมื่อนำไปใช้เป็นสัญญาณอ้างอิงสำหรับวงจรเฟสล็อกแล้ว เอาต์พุทของวงจรเฟสล็อกจะเป็นสัญญาณนาฬิกาที่ต้องการ

ในทางปฏิบัติวงจรดิฟเฟอเรนทิเอทและวงจรค่าตัวคูณนั้น สามารถทำได้โดยใช้วงจรการเปลี่ยนแปลงของขอบสัญญาณโดยอาศัยการหน่วงเวลาของนอเทท ซึ่งจะทำให้เกิดพัลส์แคบๆ ตรงช่วงขอบการเปลี่ยนแปลงของสัญญาณ สามารถทำได้โดยใช้วงจรดิจิตอลแสดงได้ดังรูป



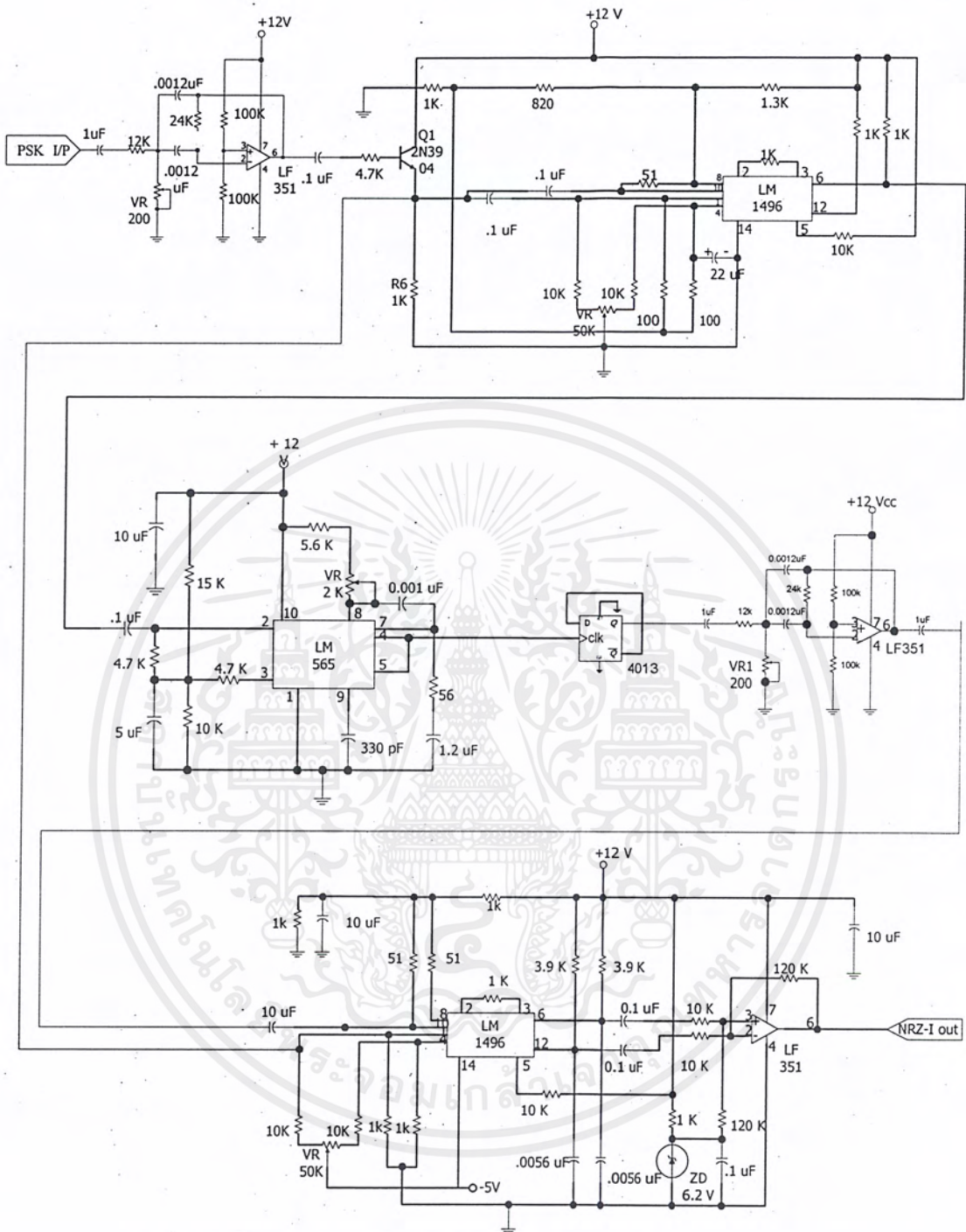
รูปที่ 3.21 วงจรการเปลี่ยนแปลงของขอบโดยอาศัยการหน่วงเวลา

โดยวงจรกู้สัญญาณนาฬิกาที่ภาครับใช้หลักการดังรูป

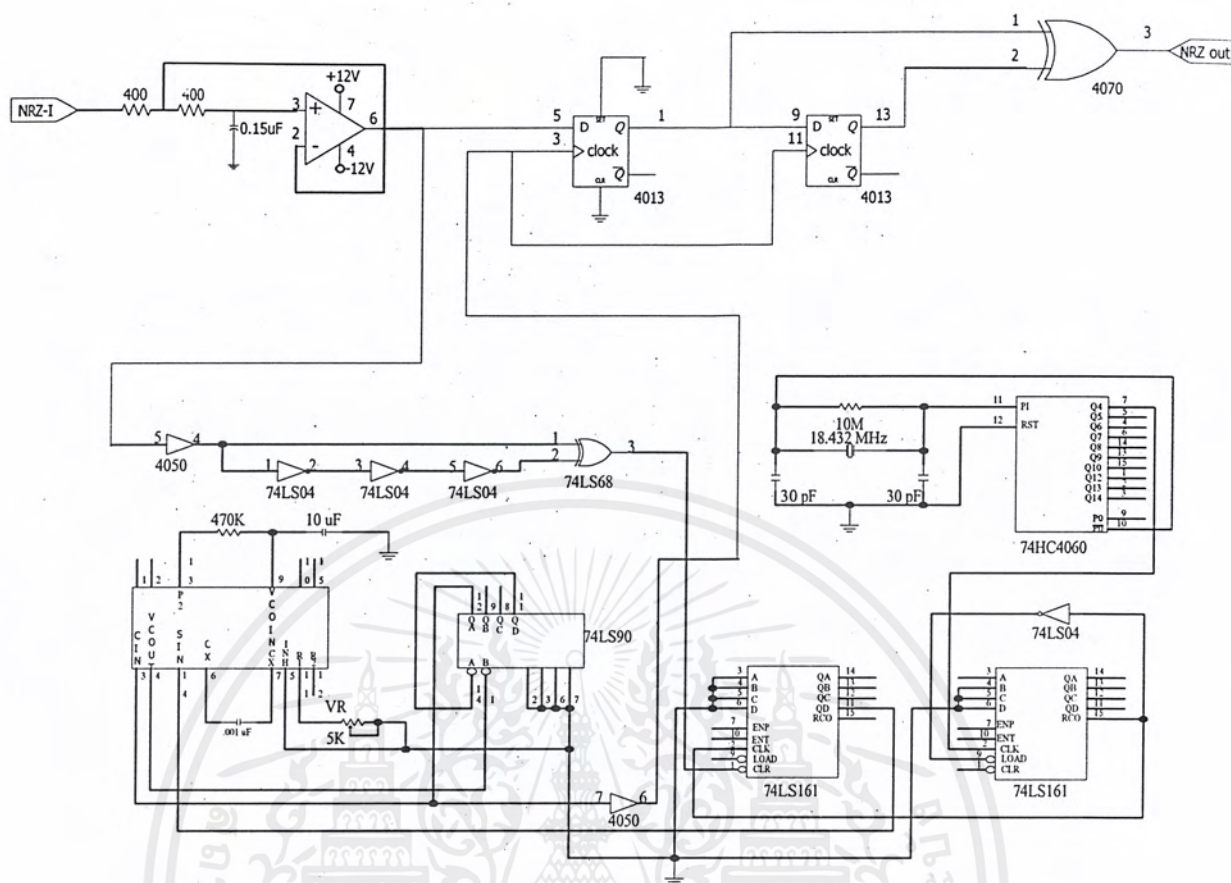


รูปที่ 3.22 แสดงหลักการของวงจรกู้สัญญาณนาฬิกา

จากรูปวงจรกู้สัญญาณนาฬิกาประกอบด้วยวงจรสร้างสัญญาณความถี่ 16 เท่าของ 4800 เฮิร์ต, วงจรการเปลี่ยนแปลงของขอบของสัญญาณ ซึ่งเอาต์พุทของวงจรหาขอบนี้จะเป็นพัลส์แคบๆ ที่เกิดขึ้นทุกๆ การเปลี่ยนแปลงของสัญญาณเอ็นอาร์แซด โดยสัญญาณพัลส์นี้จะเป็นสัญญาณเคลียร์ให้กับวงจรหาร 16 โดยที่อินพุทของวงจรเป็นสัญญาณที่มีความถี่เป็น 16 เท่าของความถี่นาฬิกา ดังนั้นเอาต์พุทของวงจรหาร 16 จะเป็นสัญญาณที่มีความถี่เท่ากับสัญญาณนาฬิกาที่ซึ่งโครโมโซมกับสัญญาณเบสแบนด์ที่รับเข้ามา แต่ยังมี jitter) ดังนั้นจึงใช้วงจรเฟสล็อกซึ่งทำหน้าที่คล้ายวงจรกรองความถี่ผ่านทำการสร้างเฉพาะความถี่นาฬิกาที่ไม่มีการสั่นผ่านเป็นสัญญาณนาฬิกาเอาต์พุท ซึ่งวงจรแสดงได้ดังรูป



รูปที่ 3.25 (ก)

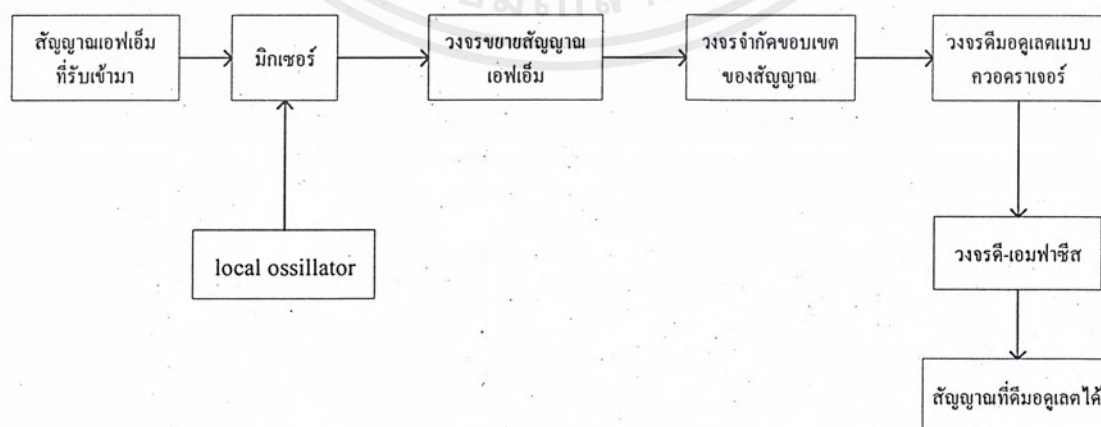


รูปที่ 3.25 (จ)

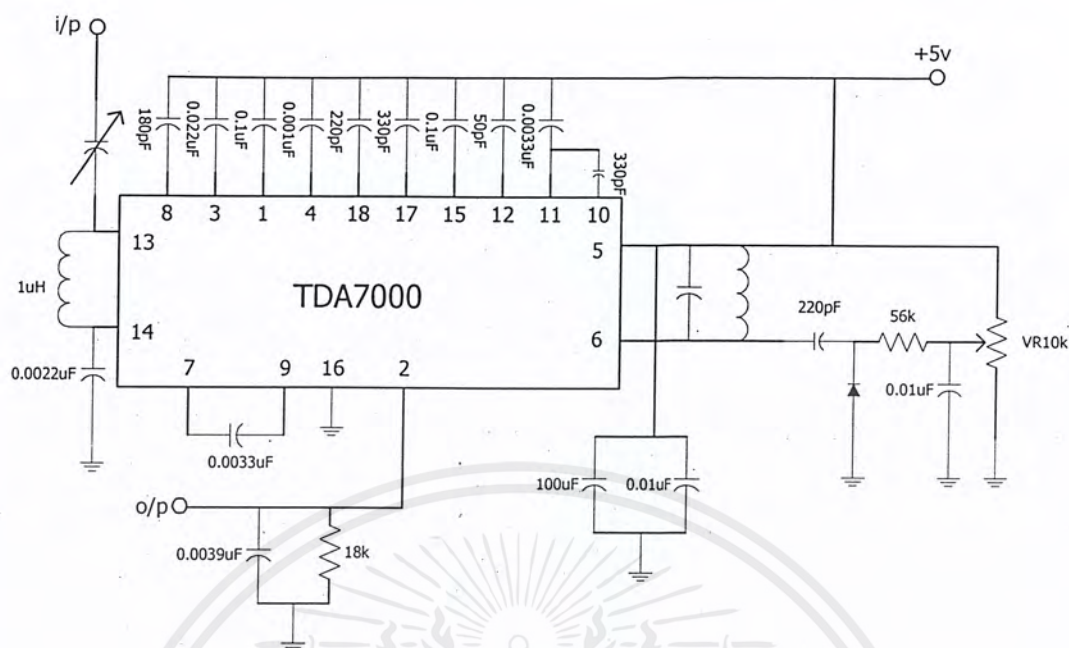
รูปที่ 3.25 แสดงวงจรคิมอดูเลทพีเอสเคทั้งหมด

3.2.8 วงจรเครื่องรับเอฟเอ็ม

หลักการทำงานของวงจรดีมอดูเลตความถี่



รูปที่ 3.26 แสดงหลักการทำงานของวงจรดีมอดูเลตความถี่

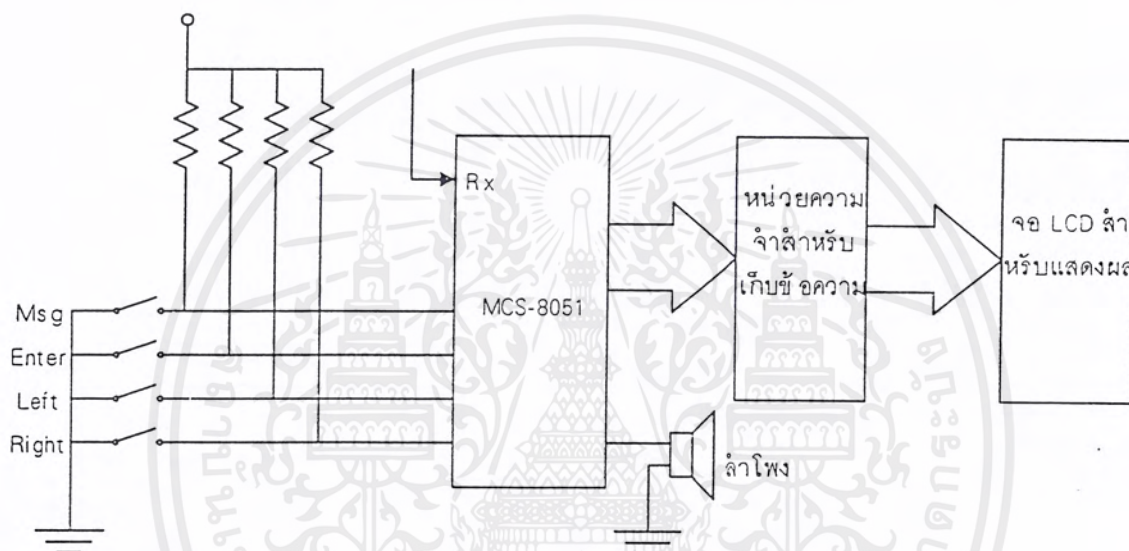


รูปที่ 3.27 แสดงรูปวงจรเครื่องรับเอฟเอ็ม

วงจรรับสัญญาณเอฟเอ็มที่ใช้ในโครงงานนี้สร้างขึ้นจากไอซีสำเร็จรูปเบอร์ TDA7000 ซึ่งเป็นไอซีที่ทำหน้าที่ในการคิ่มอดูเลตความถี่ที่ใช้หลักการของซูเปอร์เฮเทอโรไดน์ โดยสัญญาณที่คิ่มอดูเลตได้จะเป็นสัญญาณแบบโมโน จากรูป สัญญาณเอฟเอ็มที่รับเข้ามาจะเป็นสัญญาณอินพุตของไอซี (ขา13) ซึ่งจะนำไปทำการเฮเทอโรไดน์กับคลื่นพาห์เฉพาะของวงจรคิ่มอดูเลตความถี่ที่กำเนิดจากออสซิลเลเตอร์ที่อยู่ภายในตัวไอซี โดยสามารถจะจูนความถี่ดังกล่าวดังกล่าวนี้จากการปรับตัวด้านทานปรับค่าได้ การปรับตัวด้านทานปรับค่าได้นี้จะเป็นการเปลี่ยนแปลงค่าแรงดันย้อนกลับที่ตกคร่อมตัววาร์คาปเตอร์ไดโอด ทำให้ค่าความจุของวงจรจูนความถี่ของออสซิลเลเตอร์เฉพาะเปลี่ยนแปลงไป ซึ่งเป็นผลให้ความถี่ของคลื่นพาห์เฉพาะเปลี่ยนแปลงไปด้วย สัญญาณเอาต์พุตของมิกเซอร์จะเป็นสัญญาณเอฟเอ็มที่มีความถี่กลางลดลงมา เป็นความถี่ปานกลางซึ่งมีค่า 70 กิโลเฮิรตซ์ จากนั้นจะถูกนำไปขยายด้วยวงจรขยายสัญญาณเอฟเอ็มความถี่ปานกลาง แล้วจึงนำสัญญาณที่ขยายออกมาได้ไปผ่านวงจรจำกัดขอบเขตสัญญาณ (Limiter) เพื่อขจัดสัญญาณรบกวนอันเนื่องมาจากการเปลี่ยนแปลงแอมพลิจูด หลังจากที่สัญญาณเอฟเอ็มดังกล่าวผ่านวงจรจำกัดขอบเขตของสัญญาณแล้วจึงถูกนำเข้าทำการคิ่มอดูเลตแบบควอดราเจอร์เพื่อแยกสัญญาณที่มอดูเลตออกจากคลื่นพาห์ และเพื่อที่จะทำการลดระดับแอมพลิจูดของสัญญาณที่คิ่มอดูเลตได้ในย่านความถี่สูง (ในอัตราที่สอดคล้องกับที่กระทำที่วงจรพรี-เอมฟาซิสของเครื่องส่ง) จึงต้องนำสัญญาณที่คิ่มอดูเลตได้นี้ไปผ่านวงจรดี-เอมฟาซิสก่อน โดยสัญญาณเอาต์พุตของวงจรดี-เอมฟาซิสที่ได้จะกลายเป็นสัญญาณเอาต์พุตของไอซีออกมาที่ขา 2

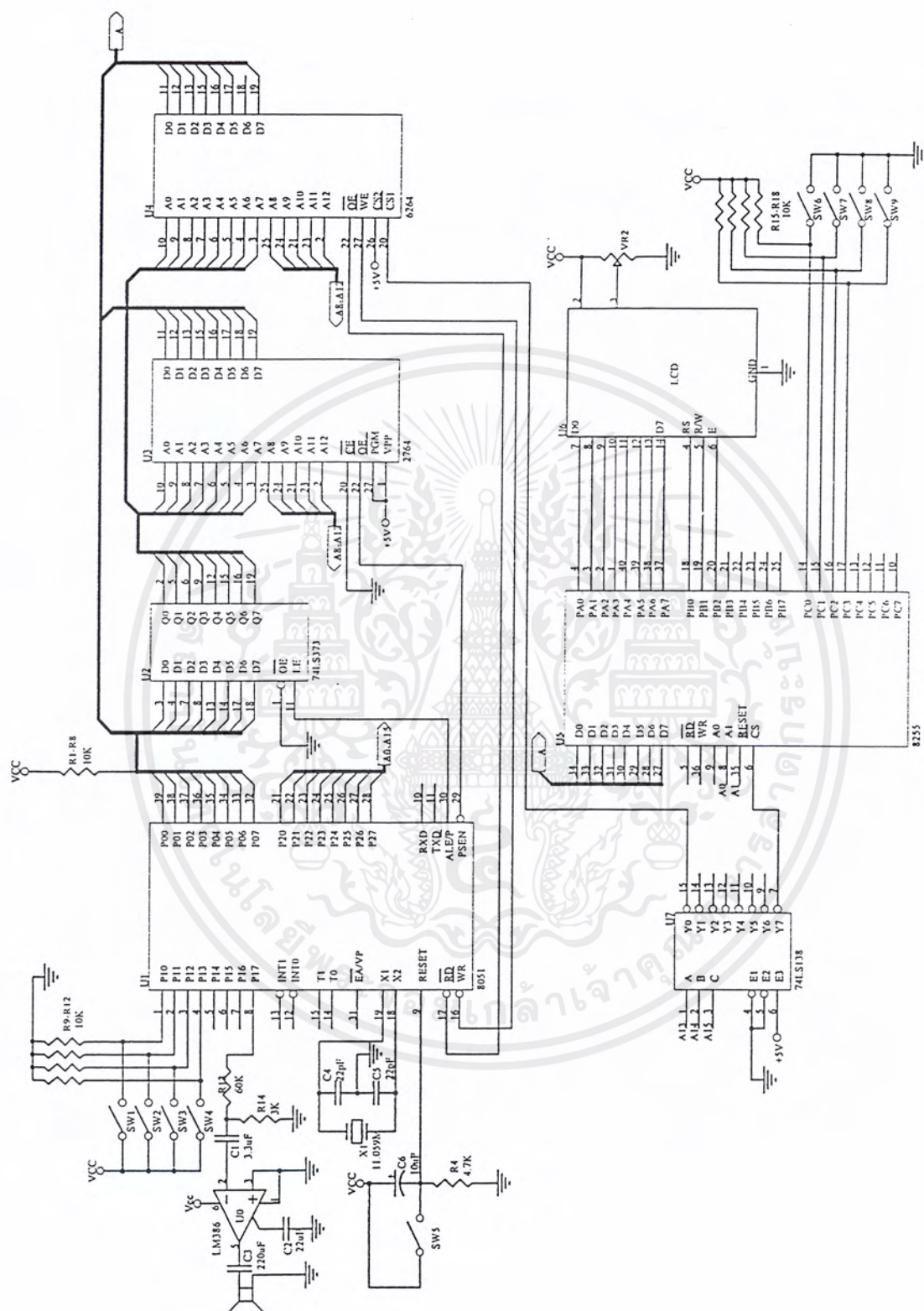
3.2.9 ชุดแสดงผล

สำหรับชุดแสดงผลเลือกใช้ไมโครคอนโทรลเลอร์ MCS-51 เป็นตัวควบคุมการแสดงผล โดยต่อร่วมกับแอลซีดี โมดูล(LCD module) สามารถแสดงตัวอักษรภาษาอังกฤษได้ทุกตัวอักษร สำหรับการติดต่อกับผู้ใช้ นั้น ผู้ใช้สามารถกดปุ่มเลือกดูข้อความที่เก็บอยู่ในหน่วยความจำ (Memory) ได้ และเมื่อมีข้อความใหม่เข้ามา ก็จะมีเสียงเตือนบอกให้รู้ว่าขณะนี้ข้อความเข้ามาแล้ว สำหรับบล็อกไดอะแกรมผลแสดงได้ดังรูปที่ 3.28



รูปที่ 3.28 แสดงบล็อกไดอะแกรมของชุดแสดงผล

8051 จะรับข้อความที่ถูกส่งมาทางขา Rx และนำไปเก็บไว้ในหน่วยความจำ เมื่อเก็บเสร็จแล้วก็จะส่งเสียงเตือนให้ผู้ใช้ทราบว่าขณะนี้ข้อความเข้ามาแล้ว ผู้ใช้สามารถดูข้อความได้โดยการกดปุ่มเลือกข้อความที่ต้องการจะดู สำหรับวงจรชุดแสดงผลโดยการใช้ไมโครคอนโทรลเลอร์ MCS-51 แสดงได้ดังรูปที่ 3.29

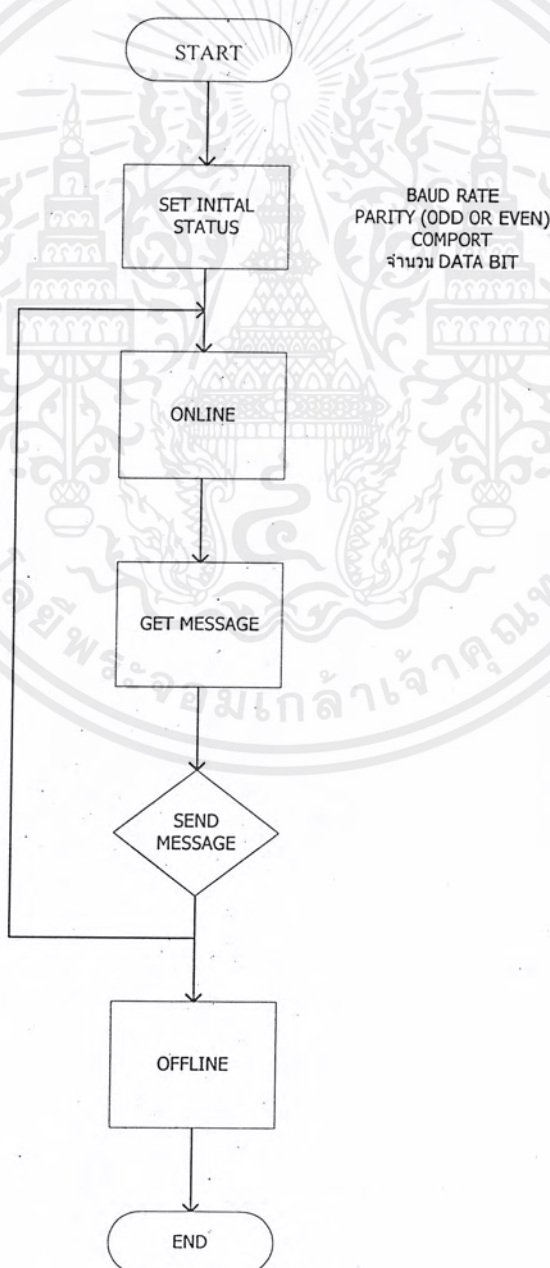


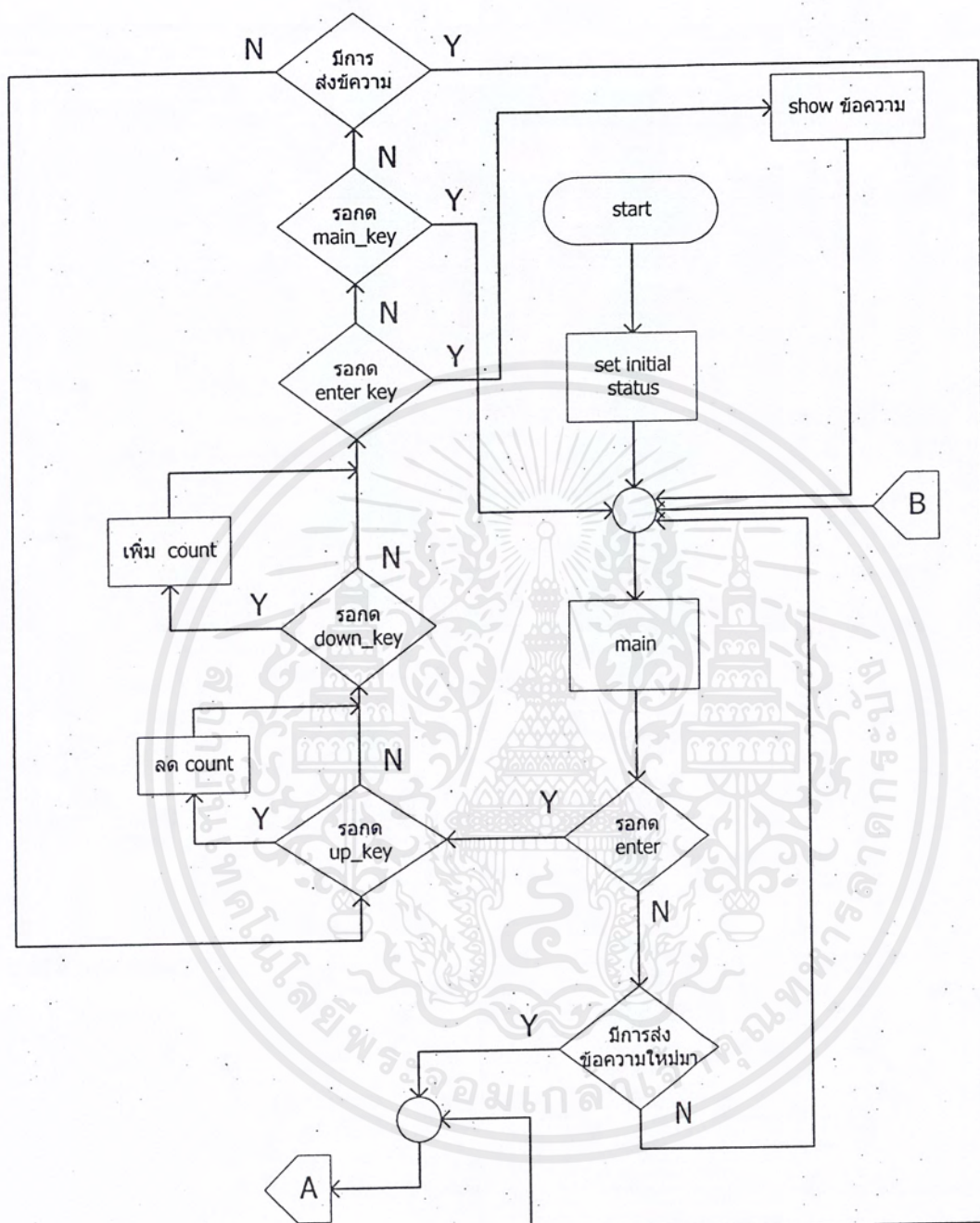
รูปที่ 3.29 แสดงวงจรชุดแอสคองผล

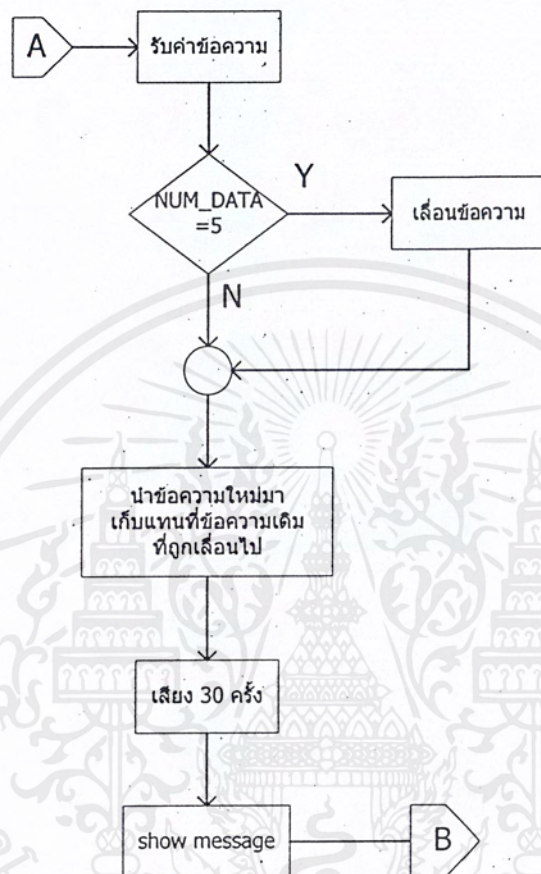
3.2.10 การออกแบบซอฟต์แวร์ที่ใช้แสดงผล

หลักการคือเมื่อมีสัญญาณอะซิงโครนัสจากภาคคิมอูลุท พีเอสเค เข้ามายังขา Rx ของ MCS-51 ก็จะเกิดการอินเตอร์พรัทขึ้น MCS-51 ก็จะทำงานอย่างอื่นชั่วขณะแล้วมารับสัญญาณที่เข้ามายังขา RX ไปเก็บไว้ในหน่วยความจำข้อมูล เมื่อ MCS-51 รับสัญญาณได้เสร็จเรียบร้อยแล้วก็จะทำเสียงบี๊ๆ ขึ้นมา สำหรับความถี่เสียงที่สร้างขึ้นมานั้นมีความถี่ประมาณ 2 กิโลเฮิรตซ์ โดยอาศัยการหน่วงเวลาและการคอมพลิเมนต์ สัญญาณเอาต์พุตที่ขา P1.7 ซึ่งต่ออยู่กับไอซี LM386 ซึ่งทำหน้าที่ขยายเสียงและจำดังไปจนกว่าผู้ใช้จะกดสวิทช์ หรือถ้าหากว่าไม่มีการกดสวิทช์ ก็จะถูกเซตให้ดังไป 30 ครั้งจึงหยุด ผู้ใช้สามารถที่จะเรียกดูข้อความที่ส่งมาได้โดยการกดปุ่มเลือกจะดูข้อความใด ในโครงงานนี้ได้ออกแบบเอาไว้ให้แต่ละข้อความยาว 80 ตัวอักษรและเก็บข้อความไว้ได้ 3 ข้อความ

สำหรับการออกแบบซอฟต์แวร์นี้ สามารถแสดงเป็นโฟลว์ชาร์ต (Flow chart) ได้ดังนี้







รูปที่ 3.30 แสดงโฟลว์ชาร์ตของการเขียนโปรแกรมชุดแสดงผล

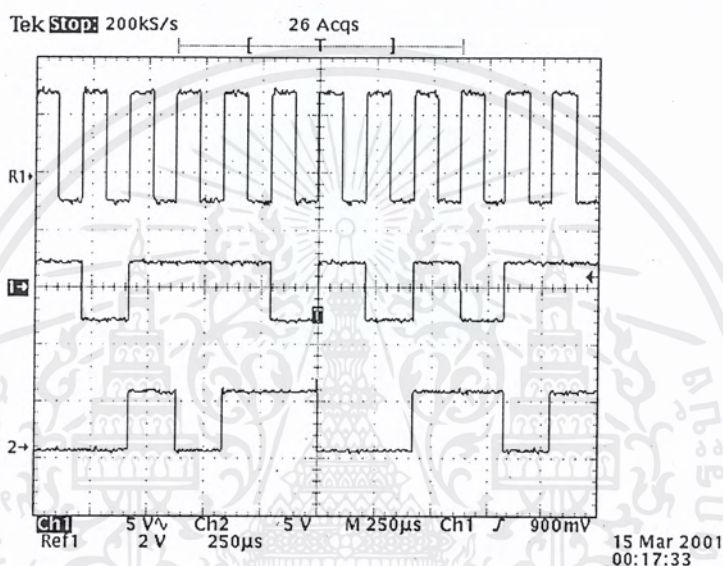
บทที่ 4

ผลการทดลอง

4.1 การทดลองวงจรภาคส่ง

4.1.1 วงจรเข้ารหัสสัญญาณเอ็นอาร์แชนด์เป็นเอ็นอาร์แชนด์ไอ

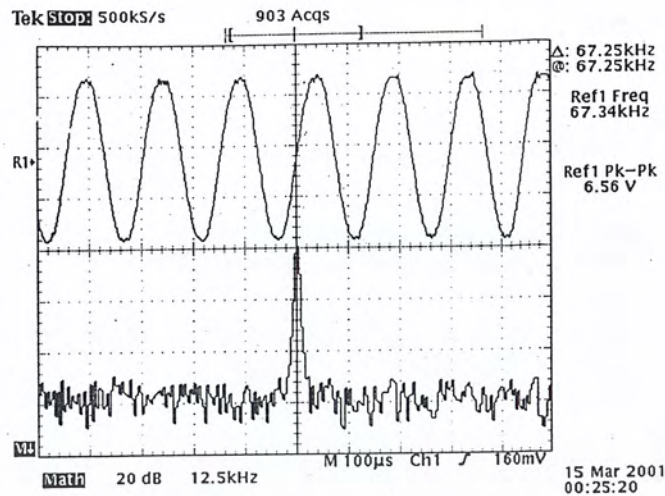
เมื่อต่อวงจรดังรูปที่ วัดเอาต์พุตที่ขา 3 ของวงจรรวม 4070 เมื่อป้อนสัญญาณเอ็นอาร์แชนด์เข้าที่ขา 1 จะได้เอาต์พุตดังรูป



รูปที่ 4.1 แสดงผลของวงจรเข้ารหัสสัญญาณ โดยรูปกลางคือสัญญาณจากวงจรบิตทดสอบ และรูปล่างคือเอาต์พุตจากวงจรเข้ารหัสสัญญาณเอ็นอาร์แชนด์เป็นเอ็นอาร์แชนด์ไอ เทียบกับสัญญาณนาฬิกา

4.1.2 วงจรกำเนิดสัญญาณคลื่นพาห้ความถี่ 67.2 กิโลเฮิร์ต

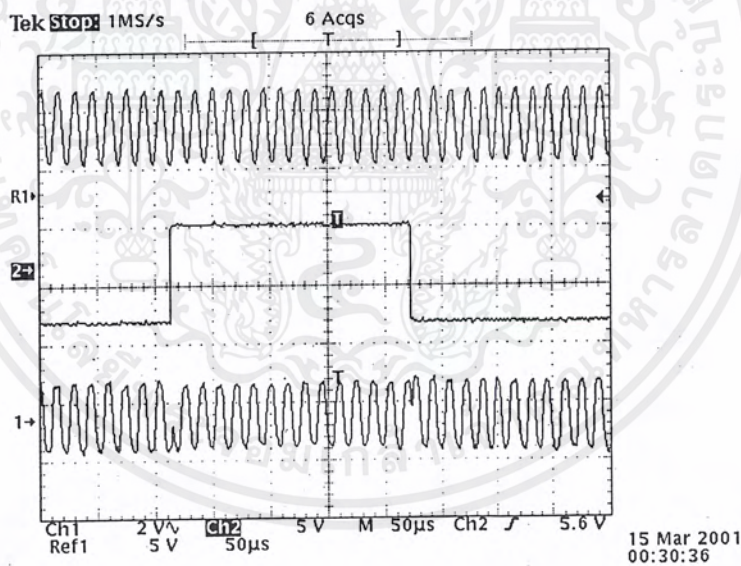
สัญญาณความถี่ 4800 เฮิร์ตจะใช้เป็นสัญญาณอ้างอิงของวงจรเฟสล็อกคัลลูป วงจรรวม XR2206 ทำการปรับค่าความดันทานปรับค่าได้ที่ขา 7 ของวงจรรวม XR2206 เพื่อให้ได้เอาต์พุตของที่ขา 2 เป็นสัญญาณไซน์ความถี่ 67.2 กิโลเฮิร์ต ส่วนสัญญาณรูปสี่เหลี่ยมความถี่ 67.2 กิโลเฮิร์ตที่ขา 11 ของวงจรรวมจะถูกนำไปหารความถี่เพื่อให้ได้สัญญาณความถี่ 4800 เฮิร์ต ซึ่งใช้เป็นสัญญาณเปรียบเทียบกับที่วงจรรวม



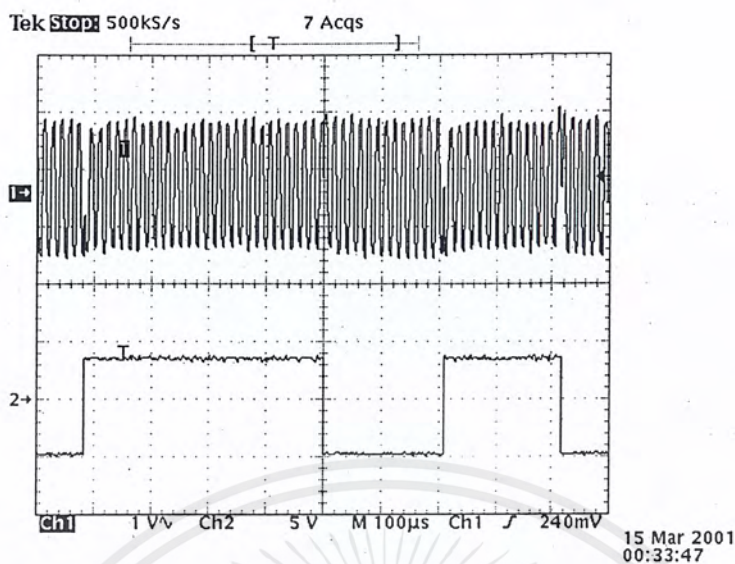
รูปที่ 4.2 สัญญาณคลื่นพาห้ความถี่ 67.2 กิโลเฮิร์ตซ์ (รูปบน) และสเปกตรัม (รูปล่าง)

4.1.3 วงจรคุณลักษณะคลื่นพาห้กับสัญญาณข่าวสาร

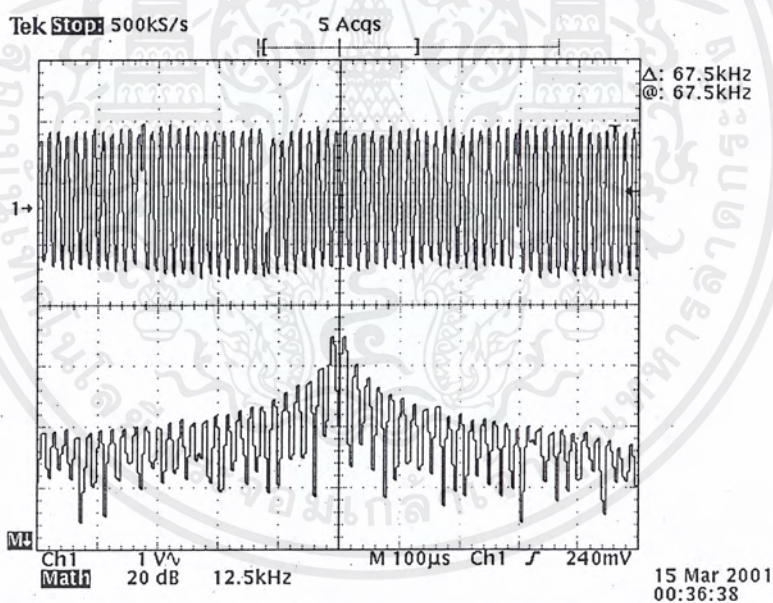
นำสัญญาณข่าวสารคูณกับสัญญาณคลื่นพาห้ โดยการป้อนสัญญาณตามรูปที่ 3.8 วัตถุประสงค์พหุที่ขา 4 ของวงจรรวม ICL8013 ได้สัญญาณดังรูป



รูปที่ 4.3 แสดงข้อมูลที่มีมอดูเลทกับคลื่นพาห้ความถี่ 67.2 กิโลเฮิร์ตซ์ (รูปล่าง) แบบบีพีเอสเค (BPSK) เปรียบเทียบกับคลื่นพาห้และข้อมูล



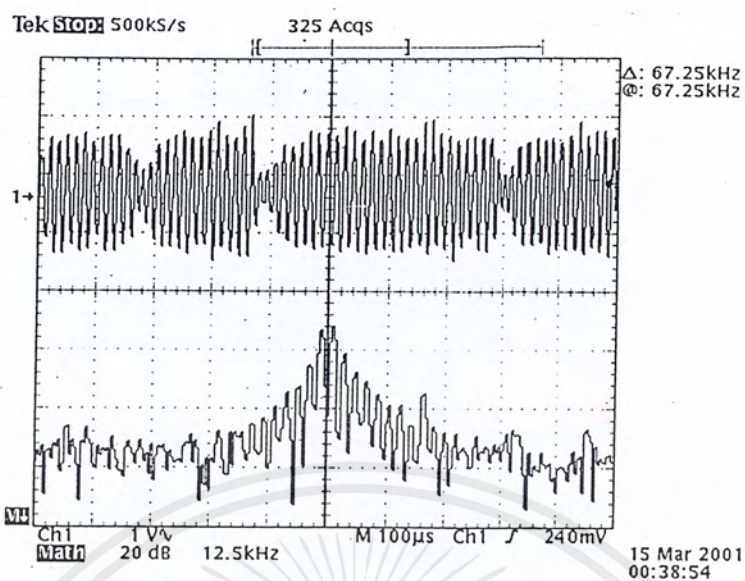
รูปที่ 4.4 แสดงสัญญาณบีพีเอสเค (รูปบน) เปรียบเทียบกับสัญญาณข้อมูล



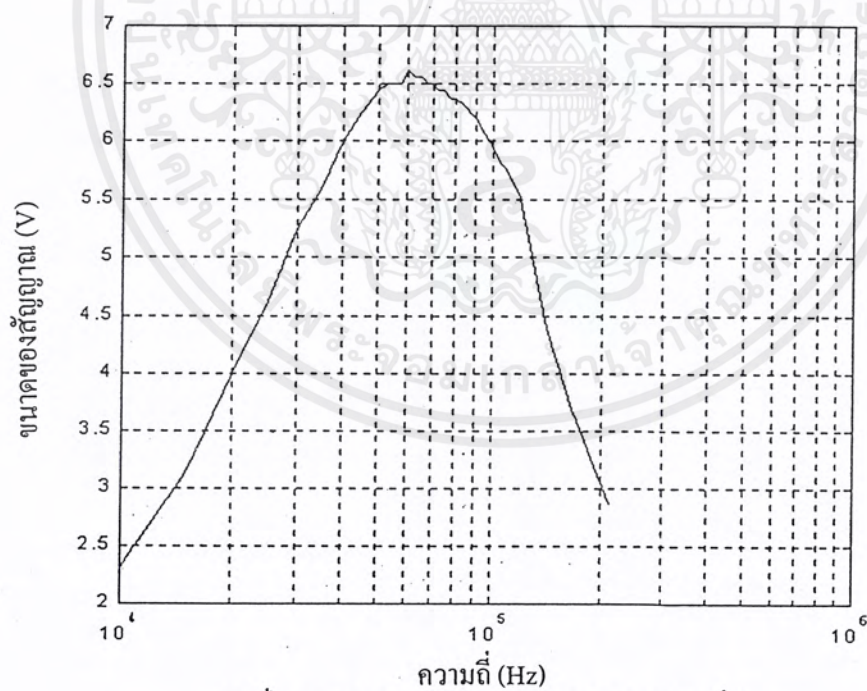
รูปที่ 4.5 แสดงสัญญาณบีพีเอสเค (BPSK) (รูปบน) และสเปกตรัมของสัญญาณก่อนผ่านวงจรกรองความถี่ผ่าน

4.1.4 วงจรกรองความถี่ผ่าน 67.2 กิโลเฮิร์ต

เมื่อป้องกันสัญญาณข่าวสารที่ส่งไปรบกวนช่องสัญญาณปกติของช่องสัญญาณเอฟเอ็มจึงทำการกรองสัญญาณโดยใช้วงจรกรองความถี่ผ่านความถี่ศูนย์กลาง 67.2 กิโลเฮิร์ต แบนด์วิธ 11.2 กิโลเฮิร์ต ต่อวงจรดังรูปที่ 3.9 วัดเอาต์พุตของสัญญาณได้ดังรูป



รูปที่ 4.6 สัญญาณบีฟีสเค และสเปกตรัมของสัญญาณเมื่อผ่านวงจรกรองความถี่ผ่านความถี่ศูนย์กลาง 67.2 กิโลเฮิร์ต

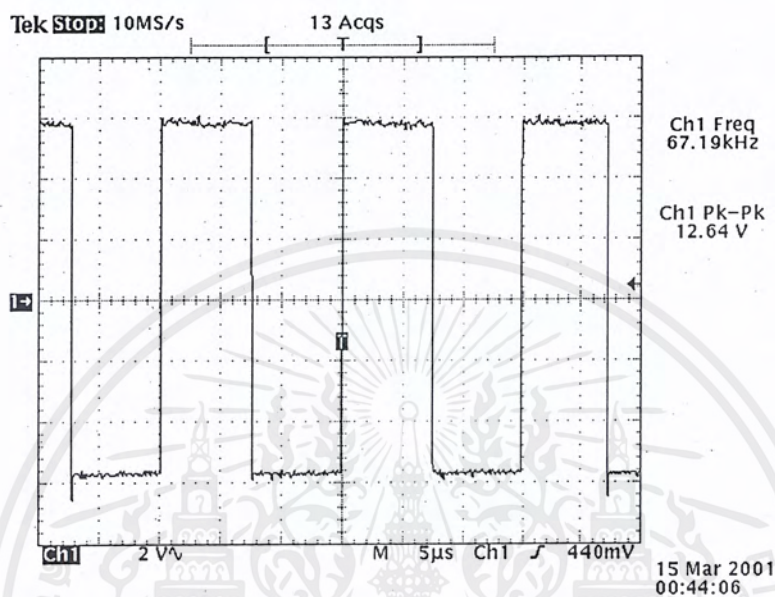


รูปที่ 4.7 แสดงคุณลักษณะของวงจรกรองความถี่ผ่าน

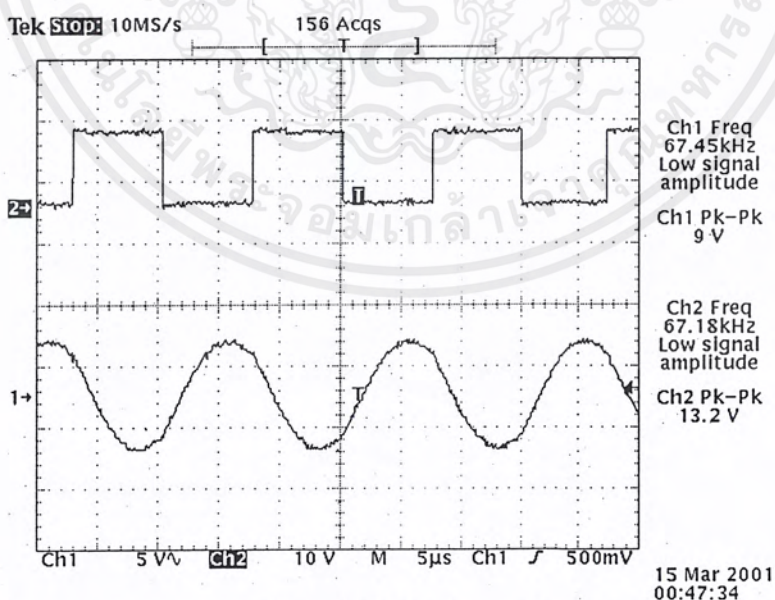
4.2 การทดลองวงจรภาครับ

4.2.1 วงจรกู้สัญญาณคลื่นพาห์

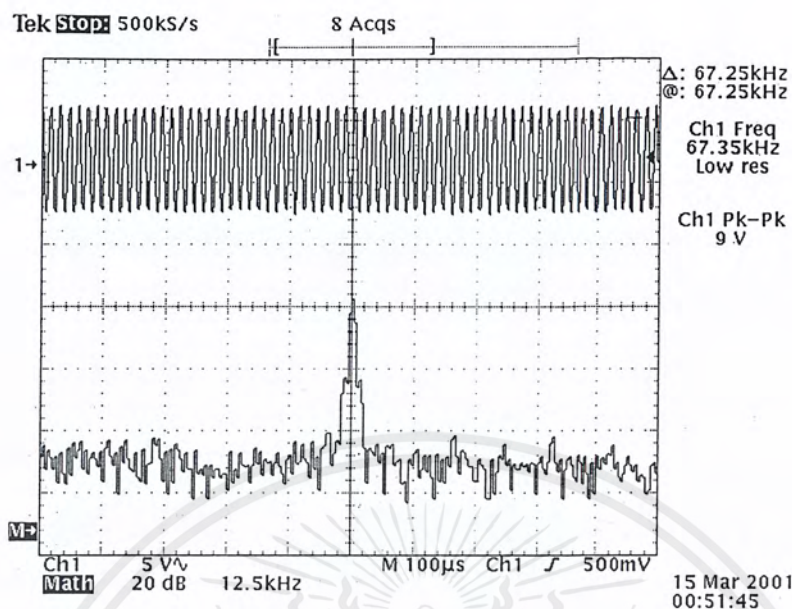
สัญญาณพีเอสเคที่รับได้ที่ด้านรับจะเป็นอินพุตของวงจรสแควริงลูป (Squaring loop) เพื่อกู้คลื่นพาห์กลับคืนมา ผลการทดลองเป็นดังรูป



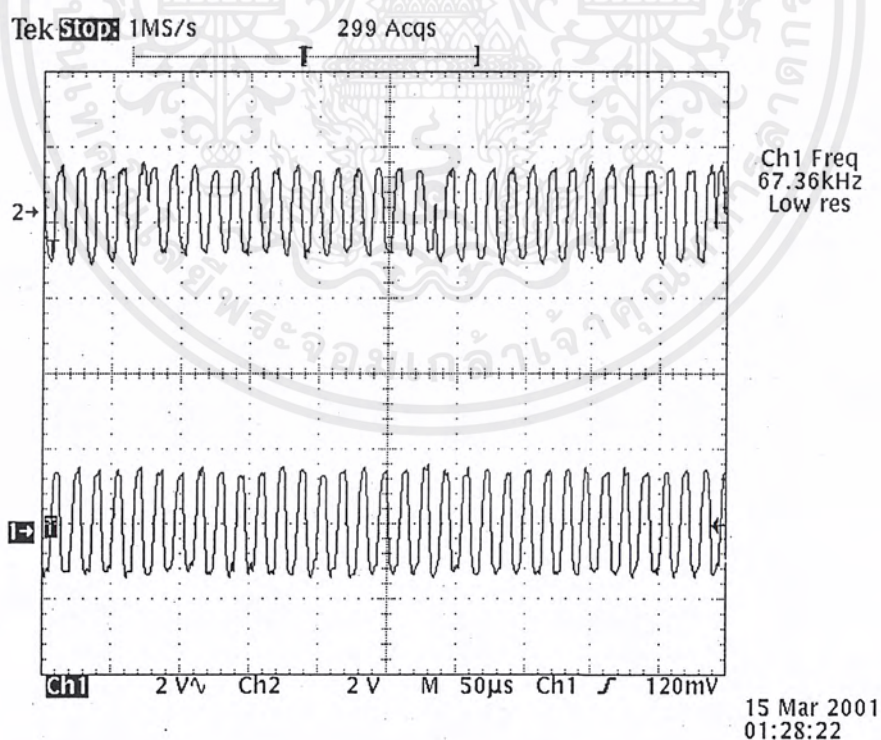
รูปที่ 4.8 แสดงสัญญาณคลื่นพาห์ความถี่ 67.2 กิโลเฮิร์ตที่ถูกกลับคืนมาได้ทางด้านรับ



รูปที่ 4.9 แสดงเอาต์พุตของวงจรกรองความถี่ผ่านเพื่อให้ได้คลื่นพาห์เป็นรูปสัญญาณไซน์ (รูปล่าง)



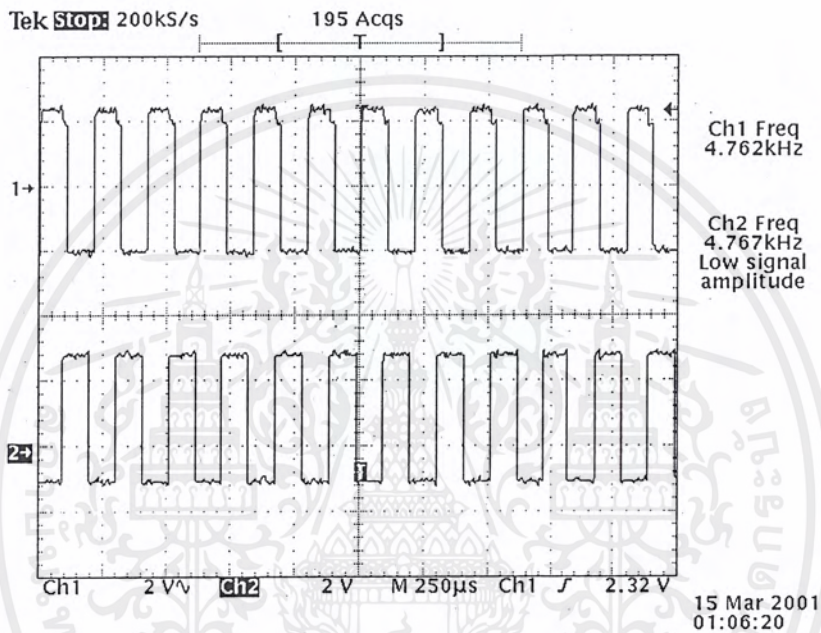
รูปที่ 4.10 คลื่นพาห์ความถี่ 67.2 กิโลเฮิร์ตที่ถูกกลับคืนมาได้และสเปกตรัมของสัญญาณ



รูปที่ 4.11 สัญญาณพีเอสเคทีรับได้ที่ด้านรับ (รูปบน) เทียบกับสัญญาณนาฬิกาที่ถูกกลับคืนได้ที่ด้านรับ

4.2.3 วงจรกู้สัญญาณนาฬิกา

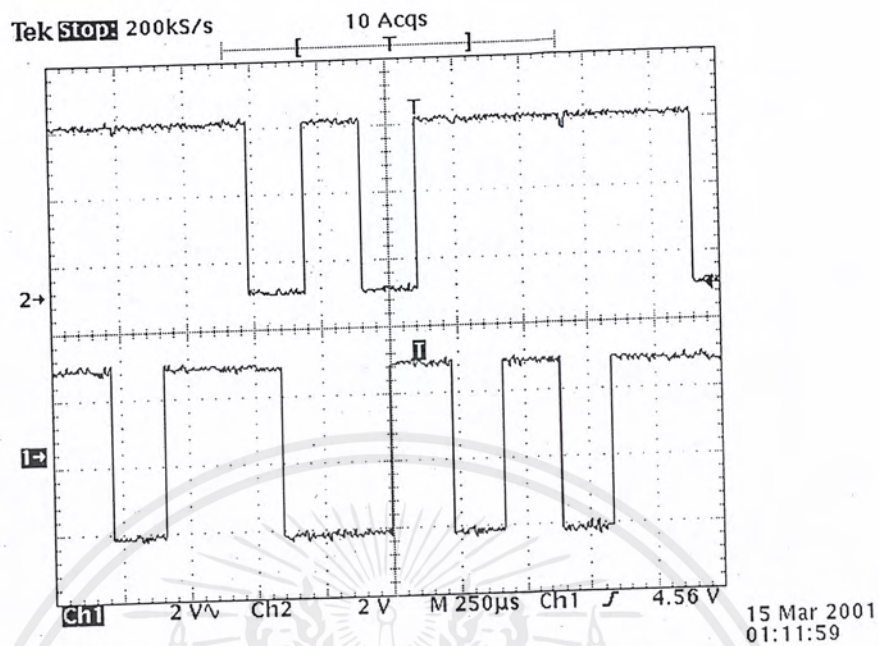
เนื่องจากทางด้านการส่งทำการเข้ารหัสสัญญาณเป็นเอ็นอาร์แซดให้ เพื่อให้ด้านรับสามารถรับสัญญาณที่ด้านส่งส่งมาได้จำเป็นต้องมีกู้สัญญาณนาฬิกากลับคืนมาเพื่อนำไปใช้ในวงจรถอดรหัส โดยนำสัญญาณข่าวสารที่กลับคืนมาได้ทางด้านรับมาเป็นสัญญาณอินพุตของวงจรหาการเปลี่ยนแปลงของขอบสัญญาณ วัดเอาต์พุตที่ขา 3 ของวงจรรวม 4046 ได้สัญญาณนาฬิกาที่ถูกกลับคืนมาได้ดังรูป



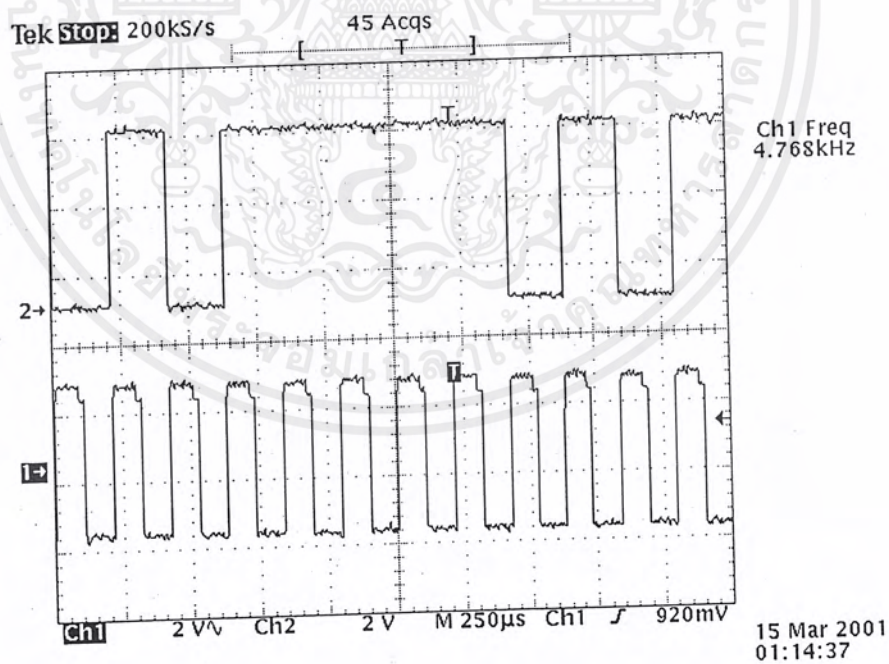
รูปที่ 4.14 สัญญาณนาฬิกาความถี่ 4.8 กิโลเฮิร์ตที่ถูกกลับคืนมาได้ทางด้านรับ (รูปบน) เทียบกับสัญญาณนาฬิกา ด้านเครื่องส่ง (รูปล่าง)

4.2.4 วงจรถอดรหัสสัญญาณ

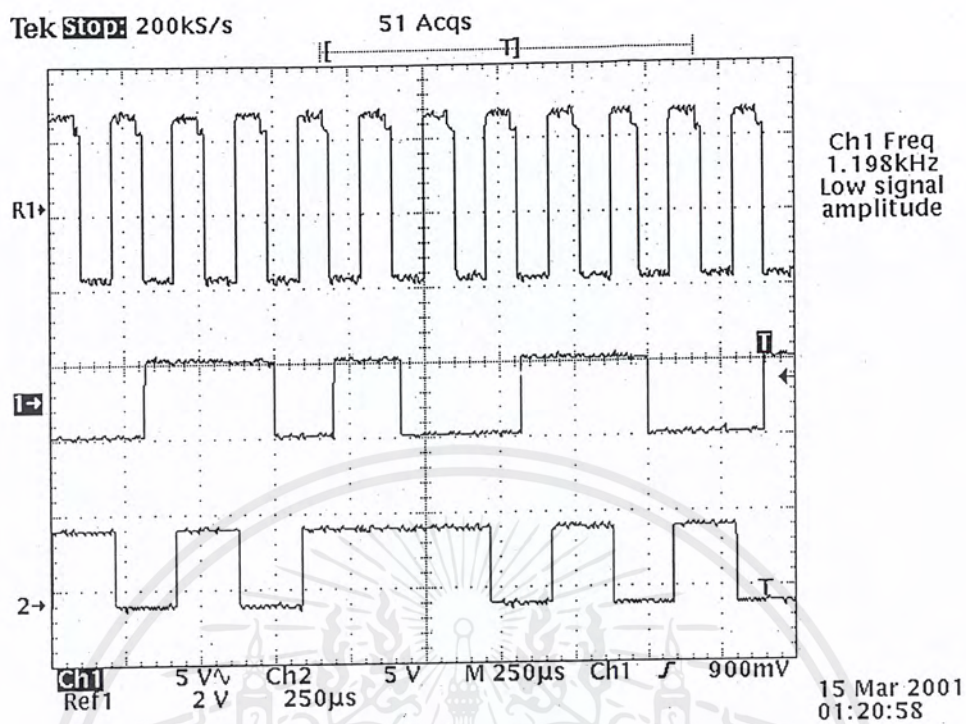
เมื่อป้อนสัญญาณข่าวสารที่ถูกกลับคืนมาได้เป็นอินพุตของวงจรถอดรหัสสัญญาณดังรูปที่ 3.24 ได้สัญญาณเอาต์พุตของวงจรดังรูป



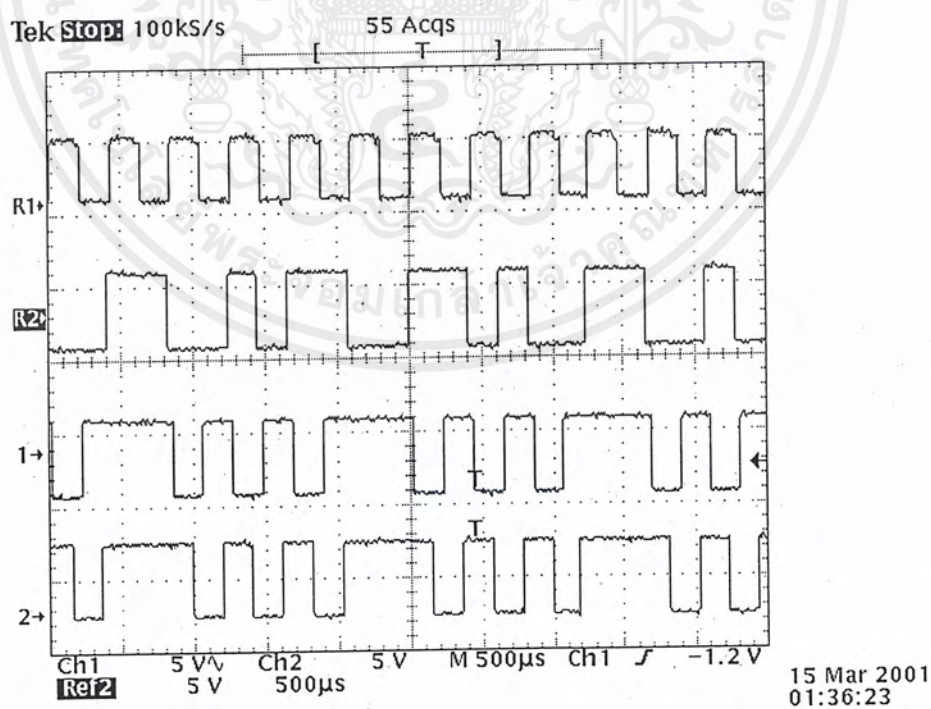
รูปที่ 4.15 สัญญาณข้อมูลหลังจากผ่านการถอดรหัสที่ด้านรับ (สัญญาณเอ็นอาร์แซด รูปล่าง) เปรียบเทียบกับสัญญาณเอ็นอาร์แซดไอที่รับได้ที่ด้านรับ



รูปที่ 4.16 สัญญาณนาฬิกาที่กู้ได้ที่ด้านรับ (รูปล่าง) เปรียบเทียบกับสัญญาณข้อมูลที่ถอดรหัสได้ (รูปบน)



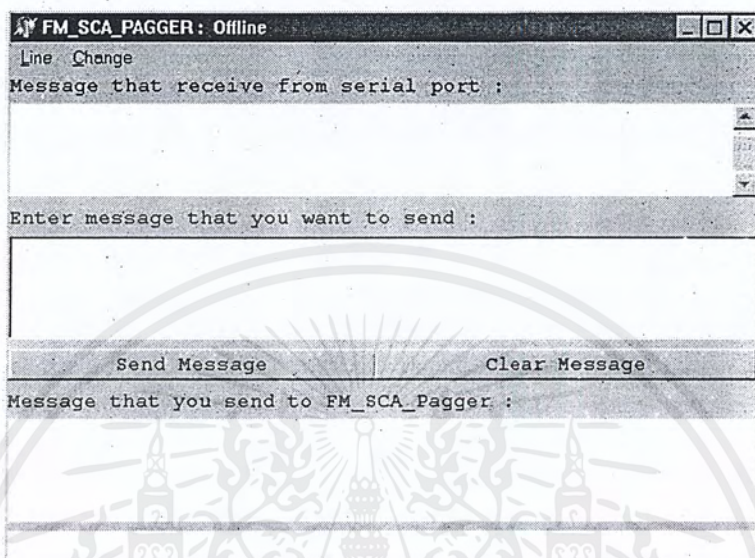
รูปที่ 4.17 สัญญาณเอนอาร์เชดไอ (รูปกลาง) และสัญญาณเอนอาร์เชด (รูปล่าง) เทียบกับสัญญาณนาฬิกา (รูปบน) ที่ด้านรับ



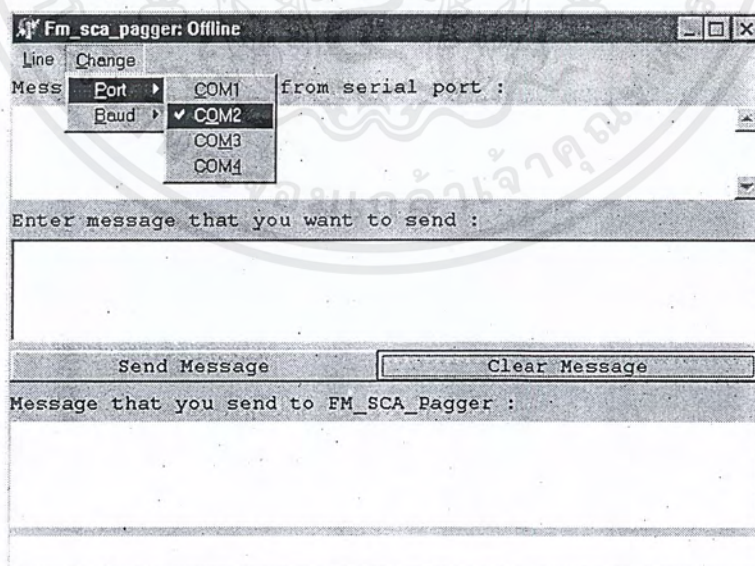
รูปที่ 4.18 ข้อมูลด้านส่ง (รูปที่ 3) และสัญญาณที่รับได้ที่ด้านรับ (รูปล่าง) เทียบกับสัญญาณนาฬิกาและสัญญาณ NRZ-I

4.2.5 โปรแกรมส่งข้อมูล

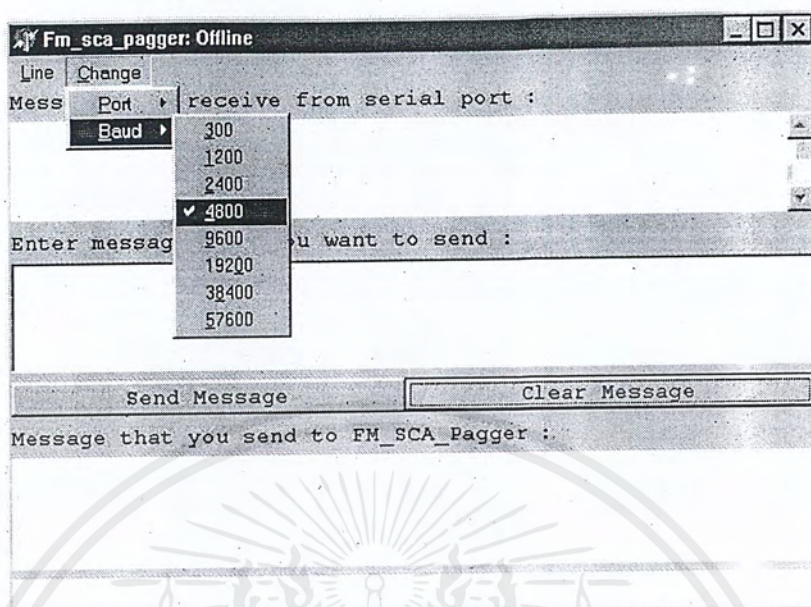
โดยในการส่งข้อความไปยังเครื่องเพจเจอร์จะทำการป้อนข้อความที่ต้องการส่งลงในโปรแกรมที่ได้จัดทำขึ้น ซึ่งสามารถแสดงหน้าจอของโปรแกรมได้ดังนี้



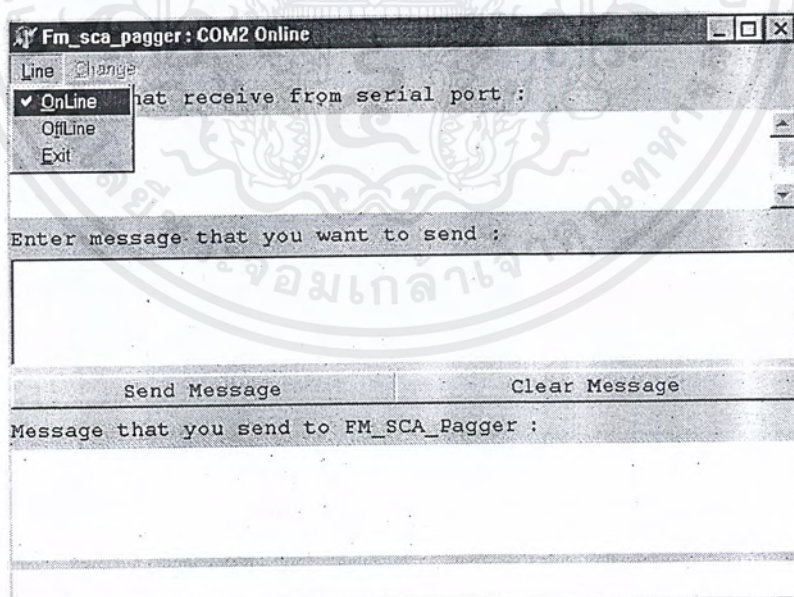
รูปที่ 4.19 แสดงหน้าจอคอมพิวเตอร์เมื่อเริ่มสาร์ทโปรแกรม



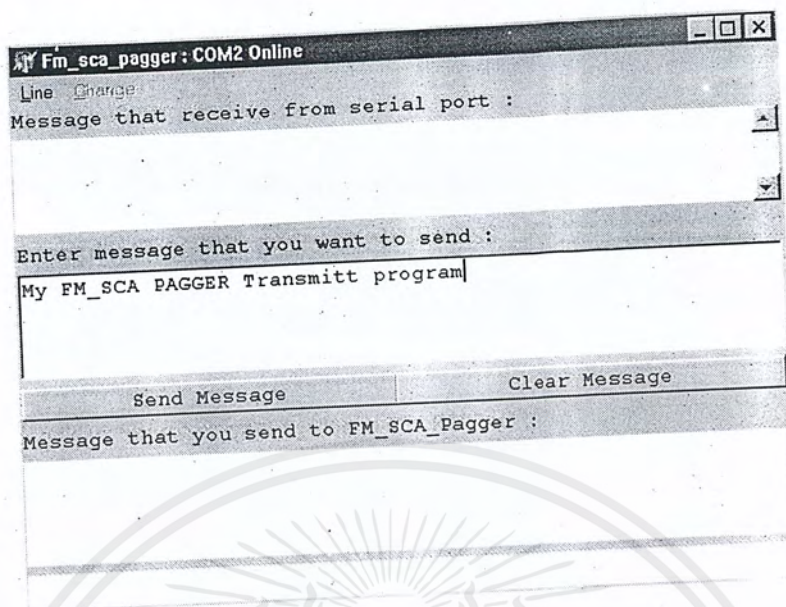
รูปที่ 4.20 แสดงการเซ็คคอมพอร์ท (Com port)



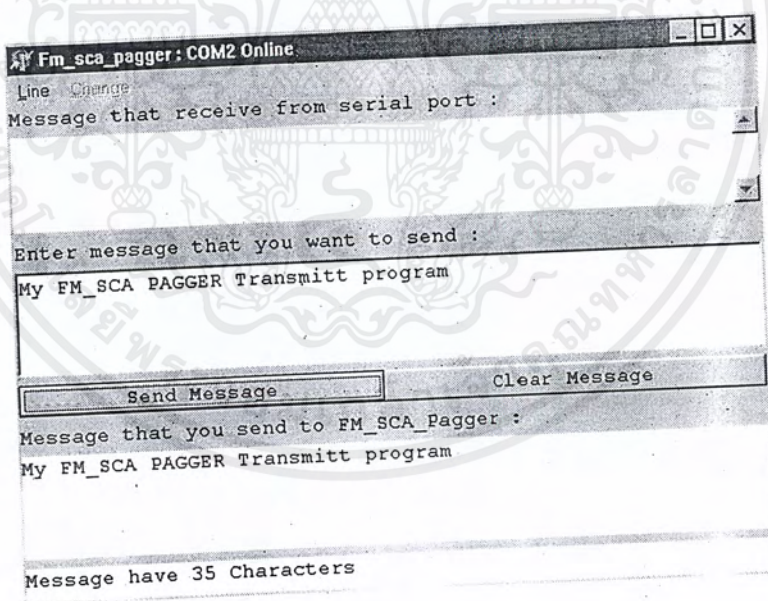
รูปที่ 4.21 แสดงการเซตบอว์ดเรท (Baud rate)



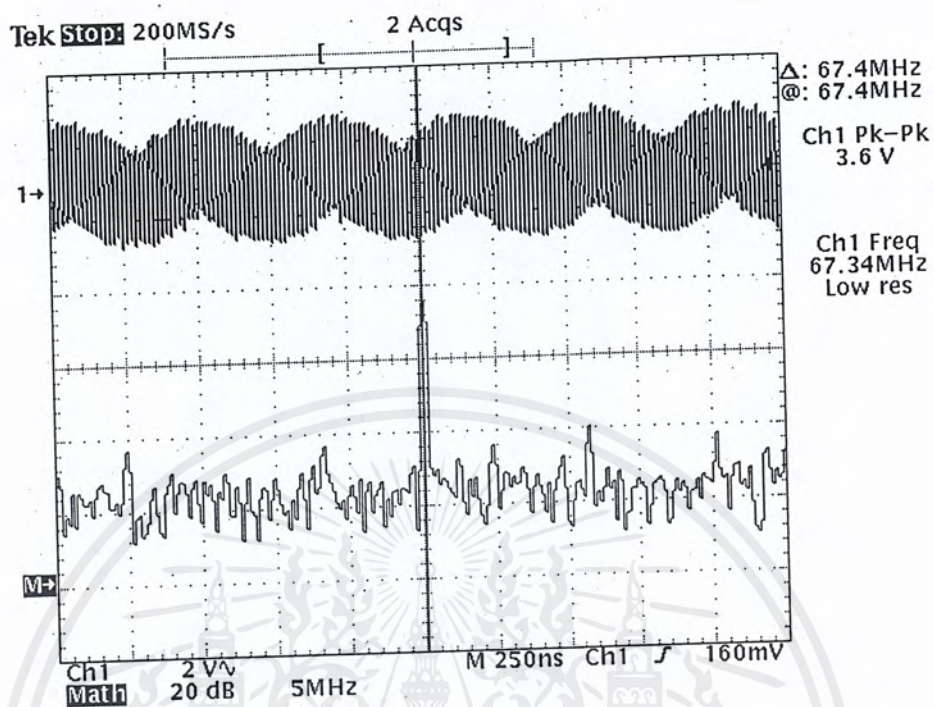
รูปที่ 4.22 แสดงการเซตเพื่อทำการออนไลน์ (Online)



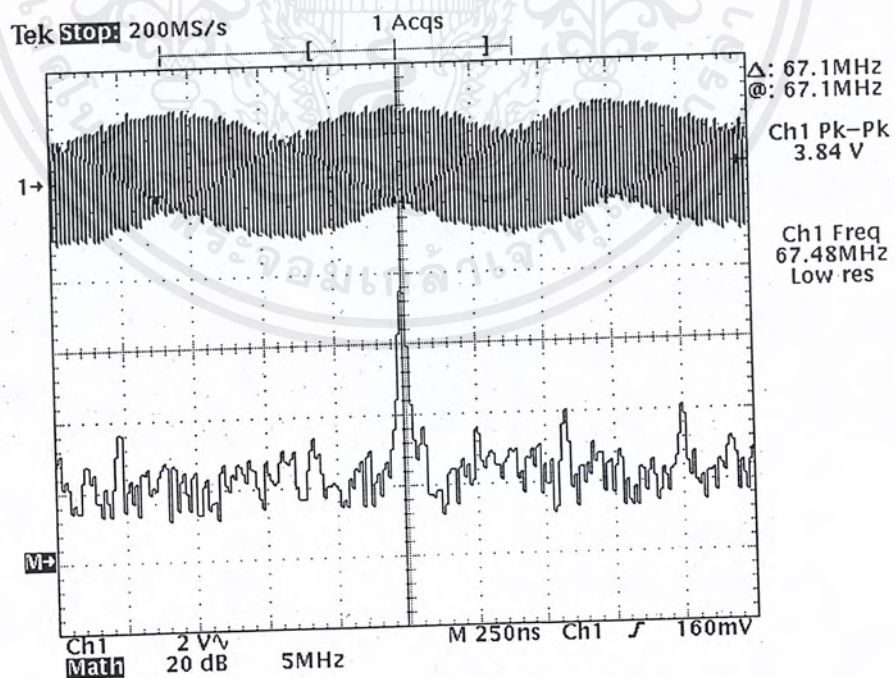
รูปที่ 4.23 แสดงข้อความที่ต้องการส่งที่ปรากฏบนโปรแกรม



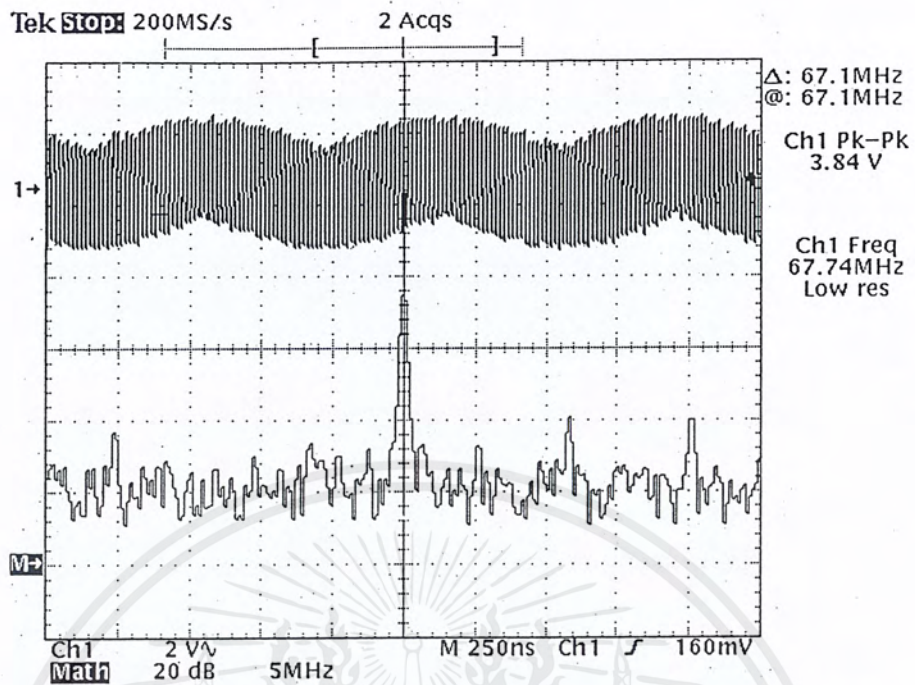
รูปที่ 4.24 แสดงหน้าจอของโปรแกรมเมื่อทำการส่งข้อความ



รูปที่ 4.25 แสดงสัญญาณคลื่นพาห์ของวงจรเครื่องส่งเอฟเอ็มและสเปกตรัมของสัญญาณ



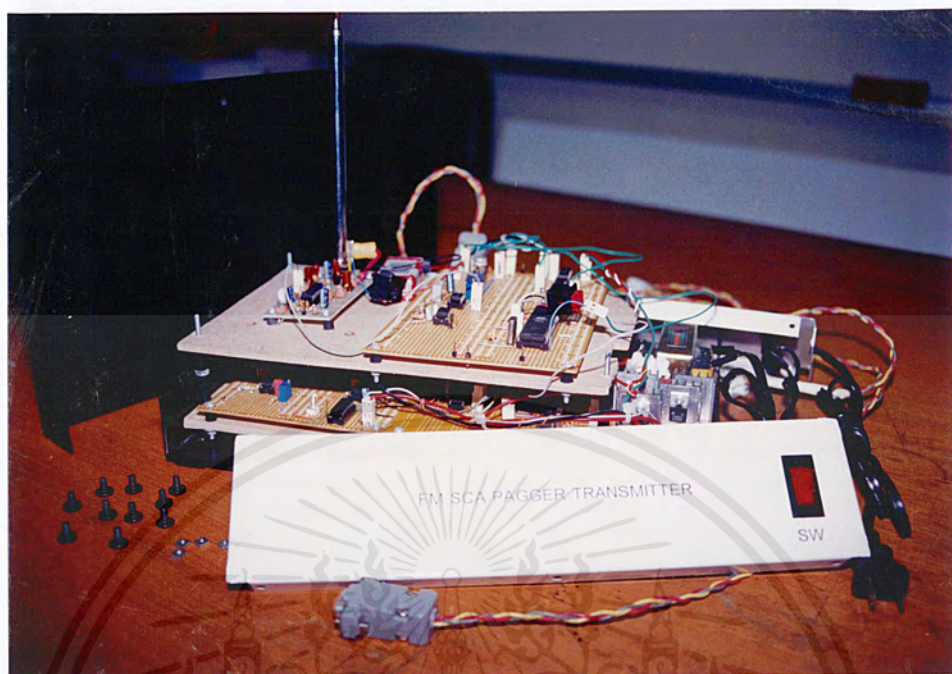
รูปที่ 4.26 แสดงสัญญาณเอาต์พุตของวงจรเครื่องส่งเอฟเอ็มเมื่อป้อนสัญญาณเบสแบนด์ความถี่ 1 กิโลเฮิร์ตซ์ และสเปกตรัมของสัญญาณ



รูปที่ 4.27 แสดงสัญญาณเอาต์พุตของวงจรเครื่องส่งเอฟเอ็มเมื่อป้อนสัญญาณเบสแบนด์ความถี่ 67.2 กิโลเฮิร์ตซ์ และสเปกตรัมของสัญญาณ



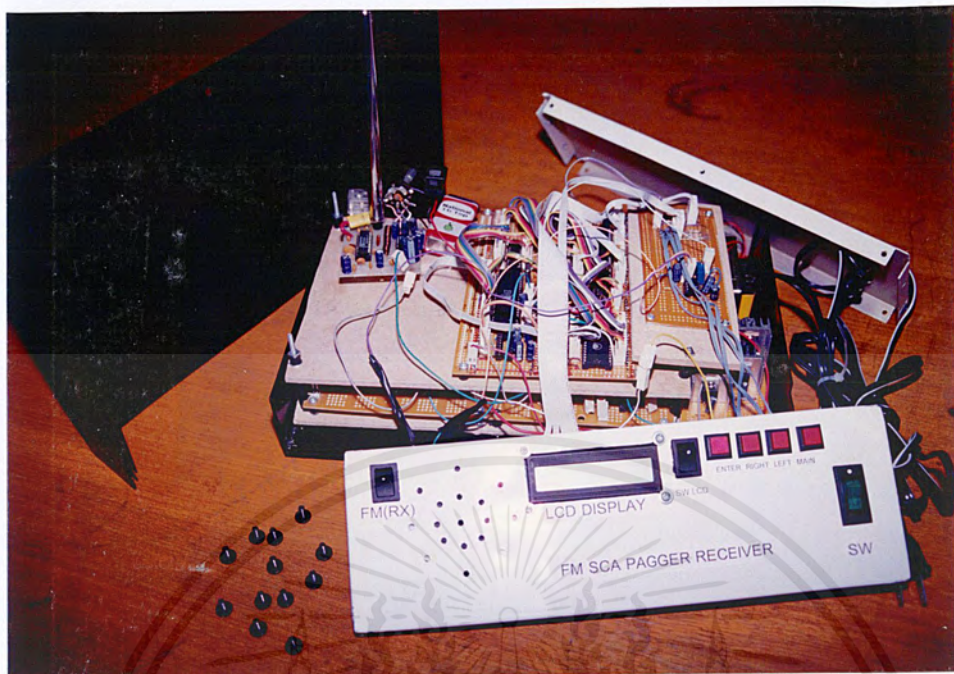
รูปที่ 4.28 รูปแสดงชิ้นงานเครื่องส่งเอฟเอ็ม เอสซีเอ เพจเจอร์



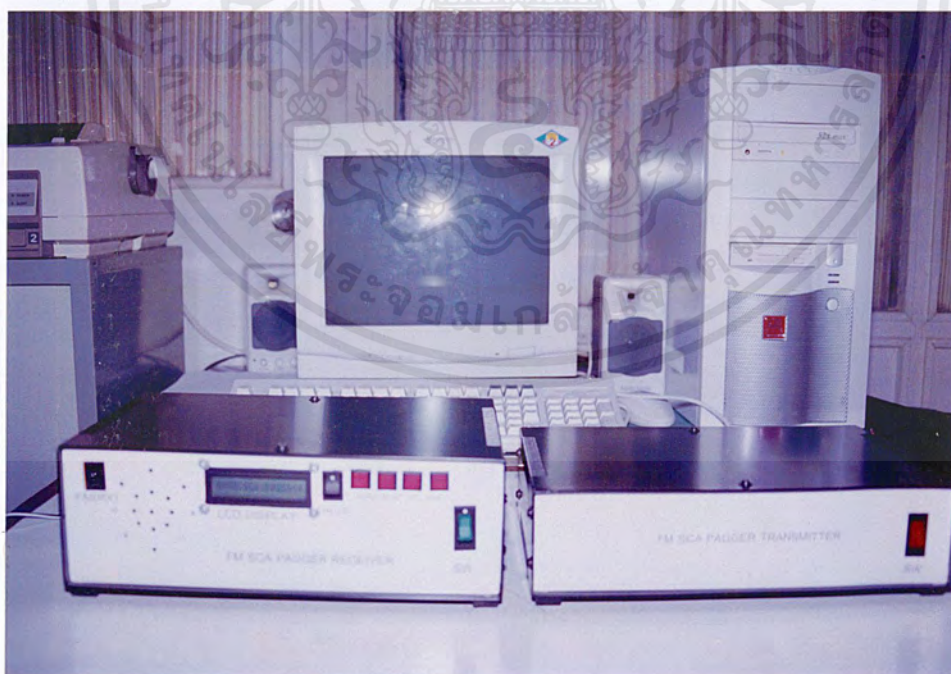
รูปที่ 4.29 รูปแสดงภายในชิ้นงานเครื่องส่ง



รูปที่ 4.30 รูปแสดงชิ้นงานเครื่องรับเอฟเอ็ม เอสซีเอ เพจเจอร์



รูปที่ 4.31 รูปแสดงภายในชิ้นงานเครื่องรับ



รูปที่ 4.32 แสดงการต่อรวมชิ้นงานทั้งหมดประกอบด้วยเครื่องรับ, เครื่องส่ง, และการส่งข้อความจากคอมพิวเตอร์

บทที่ 5

สรุปและวิจารณ์

โครงการ “วิทยุติดตามตัวโดยผ่านช่องสัญญาณเอฟเอ็ม เอสซีเอ” เป็นการประยุกต์การส่งข้อมูล โดยฝากข้อมูลไปกับคลื่นวิทยุเอฟเอ็มที่แพร่กระจายเสียง โดยใช้ช่องสัญญาณเอสซีเอในการฝากส่งข้อมูล จากการศึกษาและทำการทดลองสามารถสรุปผลได้ดังนี้

จากการทดลองของวงจรภาคส่งพบว่า

1. วงจรผลิตรคลื่นพาห์ความถี่ 67.2 กิโลเฮิร์ตซ์

ใช้วงจรเฟสล็อกกลูป ซึ่งใช้ไอซี 4046 ทำหน้าที่เป็นเฟสล็อกเตอร์ โดยสัญญาณที่ใช้เป็นสัญญาณอ้างอิงต้องมีเสถียรภาพทางความถี่ที่ดีจึงใช้วงจรผลิตรความถี่โดยใช้คริสตอลออสซิลเลเตอร์ความถี่สูง นำมาหารความถี่ให้ได้สัญญาณอ้างอิงความถี่ 4800 เฮิร์ตซ์ ป้อนเป็นสัญญาณอินพุตที่ขา 14 ของไอซี 4046 เอาต์พุตของไอซี 4046 เป็นแรงดันที่ใช้ควบคุมไอซี XR2206 ซึ่งใช้เป็นวงจรโวลเตจคอนโทรลลอสซิลเลเตอร์ โดยทำการปรับความต้านทานปรับค่าได้ที่ต่อที่ขา 7 ของไอซี XR2206 เพื่อให้ได้ความถี่เอาต์พุตที่ขา 2 เป็นรูปสัญญาณไซน์ความถี่ 67.2 กิโลเฮิร์ตซ์ ส่วนเอาต์พุตที่ขา 11 จะถูกนำไปหารความถี่เพื่อใช้เป็นสัญญาณเปรียบเทียบกับไอซี 4046 จากการทดลองพบว่าสัญญาณที่ได้มีเสถียรภาพดี

2. วงจรคูณสัญญาณข่าวสารกับคลื่นพาห์

เนื่องจากคลื่นพาห์จากวงจรผลิตรคลื่นพาห์มีสัญญาณไฟดิซียูเมื่อนำมาป้อนเป็นสัญญาณอินพุตของวงจรคูณทำให้เกิดความผิดพลาดขึ้น จำเป็นต้องตัวเก็บประจุเพื่อคัปปลิงสัญญาณไฟดิซียู สำหรับไอซี ICL8013 เมื่อนำมาต่อเป็นวงจรคูณจะเป็นต้องมีการศึกษาการทำงานจากคู่มือไอซีก่อนเพื่อทำการเซตค่าความต้านทานปรับค่าได้ขนาด 20k ที่ต่อกับไอซีเพื่อให้ได้สัญญาณเอาต์พุตที่ดีที่สุด

3. วงจรกรองความถี่ผ่าน

เนื่องจากการส่งข่าวสารเป็นการส่งในช่องสัญญาณเอฟเอ็ม เอสซีเอซึ่งเป็นการฝากส่งข่าวสารไปกับช่องสัญญาณปกติ ซึ่งอาจไปรบกวนช่องสัญญาณปกติได้ จึงจำเป็นต้องจำกัดแบนด์วิธของสัญญาณข่าวสารที่ส่งออกไป โดยใช้วงจรกรองความถี่ผ่าน ซึ่งจากผลการทดลองพบว่าวงจรสามารถกรองสัญญาณให้มีแบนด์วิธอยู่ในช่วงที่กำหนดได้ดี

4. วงจรเข้ารหัสสัญญาณเอ็นอาร์แซด (NRZ) เป็นเอ็นอาร์แซดไอ (NRZ-I)

เพื่อป้องกันการสับสนทางเฟสที่อาจเกิดขึ้นที่ด้านรับเนื่องจากขบวนในการกู้คลื่นพาห์ ความต่างเฟสของคลื่นพาห์ด้านส่งและด้านรับนี้เป็นผลทำให้การดีเทกต์สัญญาณที่ด้านรับผิดพลาด ซึ่งการผิดพลาดทางเฟสเป็นผลมาจากวงจรหารความถี่ในวงจรกู้สัญญาณคลื่นพาห์ ดังนั้นจึงต้องมีแก้ไขโดยใช้ดิฟเฟอเรนเชียล โคฮีเรนท์ พีเอสเค (differential coherent PSK) โดยวิธีการนี้ข่าวสารในสัญญาณดิจิทัลที่จะส่งจะถูกทำให้เฟสของสัญญาณพีเอสเคเปลี่ยนไปคือไม่ให้เป็นอินนิเชียลของสัญญาณพีเอสเค ซึ่งจากผลการทดลองวงจรเข้ารหัสสัญญาณสามารถทำงานได้ดี ไม่มีข้อผิดพลาด

5. เครื่องส่งเอฟเอ็ม

เนื่องจากวงจรเครื่องส่งใช้ไอซี BA1404 ซึ่งเป็นไอซีสำเร็จรูปในการมอดูเลตสัญญาณเอฟเอ็ม มีผลทำให้สัญญาณข่าวสารที่นำมามอดูเลต จะต้องมียกระดับสัญญาณในระดับที่ต่ำกว่า 500 mV และหากสัญญาณข่าวสารที่นำมามอดูเลตมีความแรงมากเกินไป จะทำให้ทางเครื่องรับตีมอดูเลตสัญญาณออกมาผิดพลาดได้ เนื่องจากค่าความถี่เบี่ยงเบนของสัญญาณคลื่นพาห้มากเกินไป และกำลังส่งของเครื่องส่งเอฟเอ็มค่อนข้างต่ำ ทำให้ส่งสัญญาณได้ไม่ไกลนัก การปรับจูนเพื่อให้ได้สัญญาณเอาต์พุตที่แรงที่สุดค่อนข้างลำบาก และเมื่อเปิดเครื่องใหม่ ความแรงของสัญญาณเอาต์พุตก็เปลี่ยนแปลงไปด้วย ทำให้ต้องมีการปรับจูนบ่อย

การทดลองวงจรภาครับ

1. วงจรเครื่องรับเอฟเอ็ม

การจูนรับสัญญาณทำได้ลำบาก ในการทดลองใช้ลำโพงต่อที่เอาต์พุตของวงจรเครื่องรับเพื่อทดสอบ โดยป้อนสัญญาณจากฟังก์ชันเจเนอเรเตอร์ที่ทางด้านส่ง เมื่อเปลี่ยนความถี่ที่ด้านส่งจะได้ยินสัญญาณเสียงที่ลำโพงเปลี่ยนไป ทำการปรับจูนได้ยินสัญญาณที่ดีที่สุด แต่มีปัญหาคือเมื่อเปิดเครื่องใหม่ ความถี่ที่ตั้งไว้จะคลาดเคลื่อนไป อาจมีปัญหาเนื่องมาจากตัวเหนี่ยวนำในวงจร บางครั้งต้องลองบิดหรือหัดตัวเหนี่ยวนำที่ค่ออยู่เพื่อให้รับสัญญาณได้ดีที่สุด จากผลการทดลองสัญญาณที่รับได้จากเครื่องรับมีระดับสัญญาณต่ำมาก หรือบางครั้งรับสัญญาณจากช่องกระจายเสียงเอฟเอ็ม ทำให้ไม่สามารถนำสัญญาณที่รับได้ไปทำการกู้ข่าวสารกลับคืนมาได้

2. วงจรกรองความถี่ผ่าน 67.2 กิโลเฮิร์ตซ์

สามารถทำงานได้ดีตามที่ออกแบบไว้

3. วงจรกู้ความถี่คลื่นพาห้

จากการทดลองพบว่าวงจรกู้สัญญาณคลื่นพาห้สามารถกู้สัญญาณคลื่นพาห้กลับมาได้ โดยสัญญาณคลื่นพาห้ที่กู้ได้นี้ซึ่งโครโมสโคปกับคลื่นพาห้ทางด้านส่ง แต่เนื่องจากสัญญาณเอาต์พุตของวงจรหารสองเป็นรูปสัญญาณสี่เหลี่ยม จึงจำเป็นต้องไปผ่านวงจรกรองความถี่ผ่านที่มีแบนด์วิดธ์แคบๆ เพื่อให้ได้สัญญาณรูปไซน์ ซึ่งพบว่าวงจรกรองความถี่ผ่านสามารถกรองได้สัญญาณเป็นรูปไซน์ตามที่ต้องการ

4. วงจรคูณสัญญาณพีเอสแควกับคลื่นพาห้

จากการทดลองพบว่าเอาต์พุตของสัญญาณมีองค์ประกอบของฮาร์โมนิกรวมอยู่ในสัญญาณเอาต์พุต ซึ่งสามารถกำจัดได้โดยใช้วงจรกรองความถี่ต่ำผ่าน และได้สัญญาณเบสแบนด์ออกมาทางด้านเอาต์พุตของวงจรกรองความถี่ต่ำผ่าน

5. วงจรกู้สัญญาณนาฬิกา

วงจรกู้สัญญาณนาฬิกาสามารถกู้สัญญาณนาฬิกาความถี่ 4800 เฮิร์ตซ์ได้ดี โดยสัญญาณนาฬิกาที่กู้ได้จะมีเฟสต่างกับสัญญาณนาฬิกาที่ด้านส่ง 180 องศา ซึ่งสามารถนำไปใช้เป็นสัญญาณนาฬิกาในวงจรถอดรหัสได้เนื่องจากสัญญาณมีการเข้ารหัสเป็นเอ็นอาร์แซดไอ

6. วงจรถอดรหัสสัญญาณเอ็นอาร์แซดไอเป็นเอ็นอาร์แซด

เนื่องจากด้านส่งมีการเข้ารหัสสัญญาณ ดังนั้นที่ด้านรับต้องมีการถอดรหัสสัญญาณเพื่อให้ได้สัญญาณข่าวสารจริงที่ส่งมา จากการทดลองพบว่าวงจรถอดรหัสสามารถทำงานได้ดี

7. อุปกรณ์แสดงผล

สามารถเก็บข้อความได้ 5 ข้อความ เมื่อทำการทดสอบส่งข้อมูลจากคอมพิวเตอร์สามารถทำงานได้ดีตามที่ตั้งไว้ โดยเมื่อสัญญาณที่ด้านอินพุตหากมีการเปลี่ยนระดับสัญญาณเครื่องจะคิดเลขสัญญาณเป็นสตาร์ทบิท และทำให้ลำโพงส่งเสียงเพื่อบอกให้ผู้ใช้ทราบว่าข้อความส่งเข้ามา

โดยการทดลองของวงจรในส่วนเครื่องรับเมื่อไม่ได้ทำการส่งออกอากาศจริงพบว่า วงจรทางด้านเครื่องรับสามารถทำงานได้ถูกต้อง แต่เมื่อทำการส่งสัญญาณทางเครื่องส่งเอฟเอ็มปรากฏว่าสัญญาณที่รับได้ ไม่สามารถทำให้วงจรเฟสล็อกถูกล็อกในวงจรได้ ความถี่คลื่นพาห์ทำงานได้ เนื่องจากรับสัญญาณที่มีขนาดต่ำเกินไป ทดลองปรับแกนของวงจรกรองความถี่ผ่านให้เป็น 2 เท่า แต่ก็ยังมีปัญหาเรื่องสัญญาณที่รับเข้ามาได้ผิดพลาด มีสัญญาณรบกวนมาก แต่ในช่วงกลางคืนสามารถรับสัญญาณได้ดีกว่า วงจรเฟสล็อกถูกล็อกสามารถทำงานได้ แต่สัญญาณข้อมูลที่ถูกกลับคืนมาได้ยังไม่ถูกต้อง ทำให้ไม่สามารถแสดงข้อความที่ด้านส่งส่งมาได้

แนวทางปรับปรุง

เพิ่มกำลังส่งของเครื่องส่งเอฟเอ็ม หรือเพิ่มวงจรโลว์นอยส์แอมพลิไฟร์ที่ด้านรับเพื่อรับสัญญาณได้ดีขึ้น และจากการศึกษาพบว่า การส่งสัญญาณแบบเอฟเอ็มจะมียังวงจรทางด้านเครื่องรับและเครื่องส่งที่ง่ายกว่านี้ เนื่องจากส่งสัญญาณโดยให้ความถี่ที่ส่งข้อมูล "0" กับ "1" เปลี่ยนแปลงไป จึงไม่จำเป็นต้องมียังวงจรความถี่คลื่นพาห์และไม่ต้องกู้สัญญาณนาฬิกากลับคืนที่ด้านรับ ใช้เพียงวงจรเฟสล็อกก็ยังสามารถกู้สัญญาณข่าวสารกลับคืนมาได้ และวงจรที่ด้านส่งก็ไม่ยุ่งยาก (เนื่องจากส่งสัญญาณข่าวสารความเร็วไม่มากนักไม่จำเป็นต้องส่งสัญญาณแบบพีเอสเค)

บรรณานุกรม

ณรงค์ เหมกรณ์, การสื่อสารดาวเทียม . สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
กรุงเทพมหานคร , 2533

บรรเจิด ดันติกัลยาภรณ์, เครื่องรับส่ง . กรุงเทพมหานคร : เครื่องรับส่ง , 2533

ประสิทธิ์ ประพัฒน์มงคลการ, หลักการระบบสื่อสาร . ซีเอ็ดยูเคชั่น

นรินทร์ เนาวประทีป , ความรู้เบื้องต้นทางวิทยุกระจายเสียงและวิทยุโทรทัศน์ . ฟิสิกส์เซนเตอร์

บุญชาติ เนติศักดิ์, ทฤษฎีและปฏิบัติเครื่องรับวิทยุ AM/FM ซีเอ็ดยูเคชั่น

ประดิษฐ์ วัชรพิบูลย์, ตำราชุดวิศวกรรมศาสตร์เครื่องส่งวิทยุและโทรทัศน์ . สถาบันเทคโนโลยีพระจอม
เกล้าเจ้าคุณทหารลาดกระบัง . กรุงเทพมหานคร





```

unit Fm_sca_pagger_unit;

interface

uses
  Display,
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics,
  Controls,
  Forms, Dialogs, Menus,
  wsc, ExtCtrls, StdCtrls, jpeg, ImgList;
type
  TFm_sca_pagger = class(TForm)
    MainMenu: TMainMenu;
    menuLine: TMenuItem;
    menuOnLine: TMenuItem;
    menuOffline: TMenuItem;
    menuExit: TMenuItem;
    menuChange: TMenuItem;
    menuPort: TMenuItem;
    menuBaud: TMenuItem;
    menuCOM1: TMenuItem;
    menuCOM2: TMenuItem;
    menuCOM3: TMenuItem;
    menuCOM4: TMenuItem;
    menu300: TMenuItem;
    menu1200: TMenuItem;
    menu2400: TMenuItem;
    menu4800: TMenuItem;
    menu9600: TMenuItem;
    menu19200: TMenuItem;
    menu38400: TMenuItem;
    menu57600: TMenuItem;
    Timer: TTimer;
    MemoMain: TMemo;
    Memo1: TMemo;
    Button1: TButton;
    Memo2: TMemo;
    Label1: TLabel;
    Label2: TLabel;
    Button2: TButton;
    Label3: TLabel;
    Edit1: TEdit;
    procedure FormCreate(Sender: TObject);
    procedure menuOnLineClick(Sender: TObject);
    procedure menuOfflineClick(Sender: TObject);
    procedure menuCOM1Click(Sender: TObject);
    procedure menuCOM2Click(Sender: TObject);
    procedure menuCOM3Click(Sender: TObject);
    procedure menuCOM4Click(Sender: TObject);
    procedure menuExitClick(Sender: TObject);
    procedure menu300Click(Sender: TObject);
    procedure menu1200Click(Sender: TObject);
    procedure menu2400Click(Sender: TObject);
    procedure menu4800Click(Sender: TObject);
    procedure menu9600Click(Sender: TObject);
    procedure menu19200Click(Sender: TObject);
    procedure menu38400Click(Sender: TObject);
    procedure menu57600Click(Sender: TObject);
    procedure TimerTimer(Sender: TObject);
    procedure Button1Click(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  end;

```

```

// procedure MyKeyPress(Sender: TObject; var Key: Char);

private
  { Private declarations }
  Port : Integer;
  Baud : Integer;
  Parity : Integer;
  DataBits : Integer;
  StopBits : Integer;

public
  { Public declarations }
end ;

var
  Fm_sca_pagger: Tfm_sca_pagger;

implementation

{$R *.DFM}

procedure Tfm_sca_pagger.FormCreate(Sender: TObject);
begin
  (* initialize parameters *)
  Port := COM1;
  Baud := Baud19200;
  Parity := NoParity;
  DataBits := WordLength8;
  StopBits := OneStopBit;
  (* initialize menu settings *)
  menuOffLine.Checked := true;
  menuCOM1.Checked := true;
  menu19200.Checked := true;
  memo1.Clear;
  memo2.Clear;
  memoMain.Clear;
  edit1.Clear;
end;

procedure Tfm_sca_pagger.menuOnLineClick(Sender: TObject);
var
  Code : Integer;
begin
  (* initialize WSC *)
  Code := SioReset(Port, 2048, 512);
  if Code < 0 then
    begin
      DisplayString(Fm_sca_pagger.MemoMain, Format('Error %d: Cannot
reset port', [Code]));
      exit
    end;
  (* update menu settings *)
  Fm_sca_pagger.Caption := 'Fm_sca_pagger : COM' + Chr($31+Port) + '
Online';
  menuOnLine.Checked := true;
  menuOffLine.Checked := false;
  menuChange.Enabled := false;
  SioBaud(Port, Baud);
  SioParms(Port, Parity, StopBits, DataBits);
  SioDTR(Port, 'S');
  SioRTS(Port, 'S');

```

```

MemoMain.SetFocus
end;

procedure TFm_sca_pagger.menuOfflineClick(Sender: TObject);
begin
    Fm_sca_pagger.Caption := 'Fm_sca_pagger: Offline';
    DisplayString(Fm_sca_pagger.MemoMain, 'Shutting down COM port');
    menuOnLine.Checked := false;
    menuOffline.Checked := true;
    menuChange.Enabled := true;
    SioDone(Port)
end;

procedure TFm_sca_pagger.menuCOM1Click(Sender: TObject);
begin
    menuCOM1.Checked := true;
    menuCOM2.Checked := false;
    menuCOM3.Checked := false;
    menuCOM4.Checked := false;
    Port := COM1
end;

procedure TFm_sca_pagger.menuCOM2Click(Sender: TObject);
begin
    menuCOM1.Checked := false;
    menuCOM2.Checked := true;
    menuCOM3.Checked := false;
    menuCOM4.Checked := false;
    Port := COM2
end;

procedure TFm_sca_pagger.menuCOM3Click(Sender: TObject);
begin
    menuCOM1.Checked := false;
    menuCOM2.Checked := false;
    menuCOM3.Checked := true;
    menuCOM4.Checked := false;
    Port := COM3
end;

procedure TFm_sca_pagger.menuCOM4Click(Sender: TObject);
begin
    menuCOM1.Checked := false;
    menuCOM2.Checked := false;
    menuCOM3.Checked := false;
    menuCOM4.Checked := true;
    Port := COM4
end;

procedure TFm_sca_pagger.menuExitClick(Sender: TObject);
begin
    SioDone(Port);
    Application.Terminate;
end;

procedure TFm_sca_pagger.menu300Click(Sender: TObject);
begin
    menu300.Checked := true;
    menu1200.Checked := false;
    menu2400.Checked := false;
    menu4800.Checked := false;

```

```
- menu9600.Checked := false;  
  menu19200.Checked := false;  
  menu38400.Checked := false;  
  menu57600.Checked := false;  
  Baud := Baud300  
end;
```

```
procedure TFm_sca_pagger.menu1200Click(Sender: TObject);  
begin  
  menu300.Checked := false;  
  menu1200.Checked := true;  
  menu2400.Checked := false;  
  menu4800.Checked := false;  
  menu9600.Checked := false;  
  menu19200.Checked := false;  
  menu38400.Checked := false;  
  menu57600.Checked := false;  
  Baud := Baud1200  
end;
```

```
procedure TFm_sca_pagger.menu2400Click(Sender: TObject);  
begin  
  menu300.Checked := false;  
  menu1200.Checked := false;  
  menu2400.Checked := true;  
  menu4800.Checked := false;  
  menu9600.Checked := false;  
  menu19200.Checked := false;  
  menu38400.Checked := false;  
  menu57600.Checked := false;  
  Baud := Baud2400  
end;
```

```
procedure TFm_sca_pagger.menu4800Click(Sender: TObject);  
begin  
  menu300.Checked := false;  
  menu1200.Checked := false;  
  menu2400.Checked := false;  
  menu4800.Checked := true;  
  menu9600.Checked := false;  
  menu19200.Checked := false;  
  menu38400.Checked := false;  
  menu57600.Checked := false;  
  Baud := Baud4800  
end;
```

```
procedure TFm_sca_pagger.menu9600Click(Sender: TObject);  
begin  
  menu300.Checked := false;  
  menu1200.Checked := false;  
  menu2400.Checked := false;  
  menu4800.Checked := false;  
  menu9600.Checked := true;  
  menu19200.Checked := false;  
  menu38400.Checked := false;  
  menu57600.Checked := false;  
  Baud := Baud9600  
end;
```

```
procedure TFm_sca_pagger.menu19200Click(Sender: TObject);  
begin
```

```

    menu300.Checked := false;
    menu1200.Checked := false;
    menu2400.Checked := false;
    menu4800.Checked := false;
    menu9600.Checked := false;
    menu19200.Checked := true;
    menu38400.Checked := false;
    menu57600.Checked := false;
    Baud := Baud19200
end;

procedure TFM_sca_pagger.menu38400Click(Sender: TObject);
begin
    menu300.Checked := false;
    menu1200.Checked := false;
    menu2400.Checked := false;
    menu4800.Checked := false;
    menu9600.Checked := false;
    menu19200.Checked := false;
    menu38400.Checked := true;
    menu57600.Checked := false;
    Baud := Baud38400
end;

procedure TFM_sca_pagger.menu57600Click(Sender: TObject);
begin
    menu300.Checked := false;
    menu1200.Checked := false;
    menu2400.Checked := false;
    menu4800.Checked := false;
    menu9600.Checked := false;
    menu19200.Checked := false;
    menu38400.Checked := false;
    menu57600.Checked := true;
    Baud := Baud57600
end;

procedure TFM_sca_pagger.TimerTimer(Sender: TObject);
var
    I, Code : Integer;
    S : String;
    CharCount : Integer;
begin
    {Gather all incoming}
    CharCount := 0;
    S := '';
    for I := 1 to 128 do
    begin
        Code := SioGetc(Port);
        if Code < 0 then break;
        {ignore CR}
        if Chr(Code) <> Chr(13) then
        begin
            {got character (other than CR)}
            Inc(CharCount);
            if Chr(Code) = Chr(10) then break;
            S := S + Chr(Code);
        end
    end; {for}
    {display}
    if CharCount > 0 then

```

```

begin
    DisplayString(Fm_sca_pagger.MemoMain,S);
    if Chr(Code) = Chr(10) then DisplayChar(Fm_sca_pagger.MemoMain,Chr
(10));
end
end;
{
procedure TEasy.MyKeyPress(Sender: TObject; var Key: Char);
begin
    SioPutc(Port,Key);
end;
}
procedure TFm_sca_pagger.Button1Click(Sender: TObject);
var
    i:integer;
begin
    i:=0;
    memo2.Clear;
    for i := 1 to Length(memol.Text) do
        begin
            SioPutc(Port,memol.Text[i]);
            memo2.Text := memo2.Text + memol.Text[i];
        end;
        edit1.Text := 'Message have '+IntToStr(i)+' Characters';
    end;

procedure TFm_sca_pagger.Button2Click(Sender: TObject);
begin
    memol.Clear;
    memo2.Clear;
    memoMain.Clear;
    edit1.Clear;
end;

end.

unit Display;

interface

uses
    WSC,
    SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics,
    Controls,
    Forms, Dialogs, Menus,
    ExtCtrls, StdCtrls;

procedure DisplayChar(Memo: TMemo; TheChar : Char);
procedure DisplayString(Memo: TMemo; Text : String);
procedure DisplayLine(Memo: TMemo; Text : String);
procedure DisplayError(Memo: TMemo; Code : Integer);

implementation

{Display character}

procedure DisplayChar(Memo: TMemo; TheChar : Char);
var
    Count      : Integer;
    TheString  : String;

```

```

begin
  if TheChar <> Chr(13) then
    {ignore CR}
    begin
      if TheChar = Chr(10) then
        begin
          {create new line}
          Memo.Lines.Add('');
        end
      else
        begin
          {printable chars only}
          if (TheChar<' ') or (TheChar>'~') then TheChar := ' ';
          {append char}
          Count := Memo.Lines.Count;
          TheString := Memo.Lines.Strings[Count-1] + TheChar;
          Memo.Lines.Strings[Count-1] := TheString;
        end
      end
    end;

  {Display string}
  procedure DisplayString(Memo: TMemo; Text : String);
  var
    S : String;
    I : Integer;
    Len : Integer;
    Count : Integer;
    TheChar : Char;
  begin
    Len := Length(Text);
    Count := Memo.Lines.Count;
    S := '';
    for I := 1 to Len do
      begin
        TheChar := Text[I];
        if TheChar <> Chr(13) then
          {ignore CR}
          begin
            if TheChar = Chr(10) then
              begin {is linefeed}
                {update current line}
                if Length(S) > 0 Then
                  begin
                    S := Memo.Lines.Strings[Count-1] + S;
                    Memo.Lines.Strings[Count-1] := S;
                  end;
                {create new line}
                Memo.Lines.Add('');
                Inc(Count); S := '';
              end
            else {is NOT linefeed}
              begin
                {printable chars only}
                if (TheChar<' ') or (TheChar>'~') then TheChar := ' ';
                {append char}
                S := S + TheChar;
              end
            end
          end
        end
      end;
    end; {end-for}
  end;

```

```

if Length(S) > 0 Then
begin
    {update string}
    S := Memo.Lines.Strings[Count-1] + S;
    Memo.Lines.Strings[Count-1] := S;
end
end;

{Display String & CRLF}

procedure DisplayLine(Memo: TMemo; Text : String);
begin
    DisplayString(Memo, Text);
    DisplayChar(Memo, Chr(10))
end;

(* Display error text *)

procedure DisplayError(Memo: TMemo; Code : Integer);
var
    Text : String;
    {$IFDEF WIN32}
    Ptr : PChar;
    {$ENDIF}
begin
    if Code < 0 then
    begin
        case Code of
            WSC_IE_BADID: Text := 'Bad port ID';
            WSC_IE_OPEN: Text := 'Cannot open port';
            WSC_IE_NOPEN: Text := 'Port already open';
            WSC_IE_MEMORY: Text := 'Cannot allocate memory';
            WSC_IE_DEFAULT: Text := 'Error in default parameters';
            WSC_IE_HARDWARE: Text := 'Hardware error';
            WSC_IE_BYTESIZE: Text := 'Unsupported byte size';
            WSC_IE_BAUDRATE: Text := 'Unsupported baud rate';
            WSC_RANGE: Text := 'Parameter out of range';
            WSC_ABORTED: Text := 'Shareware version corrupted';
            {$IFDEF WIN32}
            WSC_WIN32ERR:
            begin
                GetMem(Ptr, 120);
                Code := SioWinError(Ptr, 120);
                DisplayLine(Memo, StrPas(Ptr));
                Text := Format(' Win32 error %d.', [Code]);
                FreeMem(Ptr, 120);
            end;
            {$ENDIF}
            else Text := 'Unknown error';
        end;
        DisplayLine(Memo, Text);
    end
end;

(* initialization *)
begin
end.

unit wsc;

interface

```

```

const
  (* COMM Ports *)
  COM1 = 0;
  COM2 = 1;
  COM3 = 2;
  COM4 = 3;
  COM5 = 4;
  COM6 = 5;
  COM7 = 6;
  COM8 = 7;
  COM9 = 8;
  COM10 = 9;
  COM11 = 10;
  COM12 = 11;
  COM13 = 12;
  COM14 = 13;
  COM15 = 14;
  COM16 = 15;
  (* Baud Rate Codes *)
  Baud110 = 0;
  Baud300 = 1;
  Baud1200 = 2;
  Baud2400 = 3;
  Baud4800 = 4;
  Baud9600 = 5;
  Baud19200 = 6;
  Baud38400 = 7;
  Baud57600 = 8;
  Baud115200 = 9;
  (* Parity Codes *)
  NoParity = 0;
  OddParity = 1;
  EvenParity = 2;
  MarkParity = 3;
  SpaceParity = 4;
  (* Stop Bit Codes *)
  OneStopBit = 0;
  One5StopBits = 1;
  TwoStopBits = 2;
  (* Word Length Codes *)
  WordLength5 = 5;
  WordLength6 = 6;
  WordLength7 = 7;
  WordLength8 = 8;
  (* WSC errors *)
  WSC_NO_DATA = -100;
  WSC_RANGE = -101;
  WSC_ABORTED = -102;
  WSC_WIN32ERR = -103;
  (* Windows comm errors *)
  WSC_IE_BADID = -1;
  WSC_IE_OPEN = -2;
  WSC_IE_NOPEN = -3;
  WSC_IE_MEMORY = -4;
  WSC_IE_DEFAULT = -5;
  WSC_IE_HARDWARE = -10;
  WSC_IE_BYTESIZE = -11;
  WSC_IE_BAUDRATE = -12;
  (* SioGetError masks *)
  WSC_RXOVER = $0001;

```

```

WSC_OVERRUN      = $0002;
WSC_PARITY        = $0004;
WSC_FRAME         = $0008;
WSC_BREAK         = $0010;
WSC_TXFULL        = $0100;
(* commands *)
ASSERT_BREAK     = 'A';
CANCEL_BREAK     = 'C';
DETECT_BREAK     = 'D';
SET_LINE         = 'S';
CLEAR_LINE       = 'C';
READ_LINE        = 'R';

function SioBaud(Port, BaudCode : Integer) : Integer; stdcall
function SioBrkSig(Port : Integer; Cmd : Char) : Integer; stdcall
function SioCTS(Port : Integer) : Integer; stdcall
function SioDCD(Port : Integer) : Integer; stdcall
function SioDebug(Cmd : Integer) : Integer; stdcall
function SioDone(Port : Integer) : Integer; stdcall
function SioDSR(Port : Integer) : Integer; stdcall
function SioDTR(Port : Integer; Cmd : Char) : Integer; stdcall
function SioEvent(Port : Integer; Mask : Word) : Integer; stdcall
function SioFlow(Port : Integer; Cmd : Char) : Integer; stdcall
function SioGetc(Port : Integer) : Integer; stdcall
function SioGets(Port : Integer; Buffer : PChar; Size : Word) :
Integer; stdcall
function SioInfo(Cmd : Char) : Integer; stdcall
function SioParms(Port, ParityCode, StopBitsCode, DataBitsCode :
Integer) : Integer; stdcall
function SioPutc(Port : Integer; Ch : Char) : Integer; stdcall
function SioPuts(Port : Integer; Buffer : PChar; Size : Word) :
Integer; stdcall
function SioRead(Port : Integer; Reg : Integer) : Integer; stdcall
function SioReset(Port, RxQueueSize, TxQueueSize : Integer) : Integer;
stdcall
function SioRI(Port : Integer) : Integer; stdcall
function SioRTS(Port : Integer; Cmd : Char) : Integer; stdcall
function SioRxClear(Port : Integer) : Integer; stdcall
function SioRxQue(Port : Integer) : Integer; stdcall
function SioStatus(Port : Integer; Mask : Word) : Integer; stdcall
function SioTimer : LongInt; stdcall;
function SioTxClear(Port : Integer) : Integer; stdcall
function SioTxQue(Port : Integer) : Integer; stdcall
function SioUnGetc(Port : Integer; Ch : Char) : Integer; stdcall
function SioWinError(Buffer : PChar; Size : Word) : Integer; stdcall

```

implementation

```

function SioBaud; external 'WSC32.DLL';
function SioBrkSig; external 'WSC32.DLL';
function SioCTS; external 'WSC32.DLL';
function SioDCD; external 'WSC32.DLL';
function SioDebug; external 'WSC32.DLL';
function SioDone; external 'WSC32.DLL';
function SioDSR; external 'WSC32.DLL';
function SioDTR; external 'WSC32.DLL';
function SioEvent; external 'WSC32.DLL';
function SioFlow; external 'WSC32.DLL';
function SioGetc; external 'WSC32.DLL';
function SioGets; external 'WSC32.DLL';

```

```
function SioInfo; external 'WSC32.DLL';  
function SioParms; external 'WSC32.DLL';  
function SioPutc; external 'WSC32.DLL';  
function SioPuts; external 'WSC32.DLL';  
function SioRead; external 'WSC32.DLL';  
function SioReset; external 'WSC32.DLL';  
function SioRI; external 'WSC32.DLL';  
function SioRTS; external 'WSC32.DLL';  
function SioRxClear; external 'WSC32.DLL';  
function SioRxQue; external 'WSC32.DLL';  
function SioStatus; external 'WSC32.DLL';  
function SioTimer; external 'WSC32.DLL';  
function SioTxClear; external 'WSC32.DLL';  
function SioTxQue; external 'WSC32.DLL';  
function SioUnGetc; external 'WSC32.DLL';  
function SioWinError; external 'WSC32.DLL';  
end.
```

