

การพัฒนาซอฟต์แวร์และออกแบบส่วนประมวลสัญญาณ
ของระบบสายอากาศปรับตัว

DEVELOPMENT OF SOFTWARE AND SIGNAL PROCESSING UNIT
FOR AN ADAPTIVE ANTENNA



โดย

นางสาวเขมวไล ชีรสวรรณจักร

นางสาวศิริวรรณ ศิลป์ประกอบ

นางสาวส่องสี ไกรวิเศษ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

เลขที่.....
เลขทะเบียน... 42207
วัน, เดือน, ปี 15 พ.ค. 2545

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

611006122

การพัฒนาซอฟต์แวร์และออกแบบส่วนประมวลสัญญาณ
ของระบบสายอากาศปรับตัว

DEVELOPMENT OF SOFTWARE AND SIGNAL PROCESSING UNIT
FOR AN ADAPTIVE ANTENNA

โดย

นางสาวเขมวไล ชีรสวรรณจักร 40010096

นางสาวศิริวรรณ ศิลป์ประกอบ 40010779

นางสาวส้องสี ไกรวิเศษ 40010927

อาจารย์ที่ปรึกษา

รศ.ดร.โมไนย ไกรฤกษ์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2543

ปริญญานิพนธ์ปีการศึกษา 2543

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การพัฒนาซอฟต์แวร์และออกแบบส่วนประมวลสัญญาณของระบบสายอากาศปรับตัว

DEVELOPMENT OF SOFTWARE AND SIGNAL PROCESSING UNIT FOR
AN ADAPTIVE ANTENNA

ผู้จัดทำ

1. นางสาวเขมวไล ชีรสุวรรณจักร 40010096
2. นางสาวศิริวรรณ ศิลป์ประกอบ 40010779
3. นางสาวส้องสี ไกรวิเศษ 40010927

(รศ.ดร. โมไนย ไกรฤกษ์)

อาจารย์ที่ปรึกษา



การพัฒนาซอฟต์แวร์และออกแบบส่วนประมวลสัญญาณ
ของระบบสายอากาศปรับตัว
DEVELOPMENT OF SOFTWARE AND SIGNAL
PROCESSING UNIT FOR AN ADAPTIVE ANTENNA

โดย	นางสาวเขมวไล ชีรสุวรรณจักร	40010096
	นางสาวศิริวรรณ ศิลป์ประกอบ	40010779
	นางสาวส่องสี ไกรวิเศษ	40010927

อาจารย์ที่ปรึกษา รศ.ดร. โมไนย ไกรฤกษ์

บทคัดย่อ

โครงการนี้ เป็นการศึกษาหลักการทำงานของสายอากาศชนิดปรับตัว รวมทั้งศึกษาเกี่ยวกับอัลกอริทึมที่นำมาใช้กับสายอากาศชนิดนี้ จากนั้นจะพัฒนาซอฟต์แวร์ที่ใช้สำหรับการประมวลสัญญาณและออกแบบวงจรส่วนประมวลสัญญาณ โดยทำการทดลองที่ย่านความถี่ไมโครเวฟ

ABSTRACT

This project concerns with the algorithm study of an adaptive antenna. Then, the software for processing unit of the signal from antenna will be developed and implemented. This antenna will be designed and tested at the microwave frequency range.

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	
2.1 การกรองตำแหน่งและทิศทาง	2
2.2 การคำนวณค่าถ่วงน้ำหนักเพื่อรับสัญญาณ $p(t)$ และกำจัดสัญญาณรบกวน $n(t)$	5
2.3 รูปลักษณ์ของแถวลำดับองค์ประกอบปรับตัว	6
2.4 ตัวประมวลสัญญาณแบบปรับตัว	7
2.5 วิธีการหาคำตอบสมการซิมูลเทนเนียส	9
2.6 ความเป็นมาของอัลกอริทึมกำลังสองเฉลี่ยน้อยที่สุด	12
2.7 คอนเวอร์เจนซ์ของเวกเตอร์ถ่วงน้ำหนัก	13
2.8 วงจรวก	15
2.9 วงจรลบ	15
2.10 วงจรอินทิเกรต	16
2.11 วงจรเลื่อนเฟส	19
2.12 Successive Approximation ADC	21
2.13 การใช้งานของพอร์ตอนุกรม	25
2.14 การออกแบบวงจรลอจิกสำหรับหน่วยความจำและไอโอ	31
บทที่ 3 การออกแบบและการสร้าง	43
บทที่ 4 การทดลองและผลการทดลอง	49
บทที่ 5 บทสรุปและบทวิจารณ์	82
ภาคผนวก	
บรรณานุกรม	
กิตติกรรมประกาศ	

สารบัญรูปลูกภาพ

	หน้า
รูปที่ 2.1 แบบรูปเจาะจงทิศทางสำหรับสายอากาศแถวลำดับเชิงเส้น	3
(a) แถวลำดับอย่างง่าย	
(b) มีวงจรประวิงเวลา	
รูปที่ 2.2 แบบรูปเจาะจงทิศทางสำหรับสายอากาศแถวลำดับเชิงเส้น	4
(a) การถ่วงน้ำหนักเท่ากัน	
(b) การถ่วงน้ำหนักสำหรับวิธีการกำจัดสัญญาณรบกวน	
รูปที่ 2.3 รูปลักษณะของแถวลำดับสำหรับตัวอย่างการกำจัดสัญญาณรบกวน	5
รูปที่ 2.4 รูปลักษณะของแถวลำดับปรับตัวได้สำหรับการรับสัญญาณแถบแคบ	6
รูปที่ 2.5 รูปลักษณะของแถวลำดับปรับตัวได้สำหรับการรับสัญญาณแถบกว้าง	7
รูปที่ 2.6 องค์ประกอบปรับตัวได้มูลฐาน	7
รูปที่ 2.7 ผลรวมเชิงเส้นแบบปรับตัวได้	12
(a) แบบธรรมดา	
(b) แบบตัวกรองความถี่เชิงแนวขวาง	
รูปที่ 2.8 วงจรบวก	15
รูปที่ 2.9 วงจรลบ	16
รูปที่ 2.10 ลักษณะ โครงสร้างของวงจรอินทิเกรตแบบกลับเฟส	16
รูปที่ 2.11 ความสัมพันธ์ของสัญญาณอินพุต v_i และสัญญาณเอาต์พุต v_o	17
วงจรอินทิเกรตเป็นแบบกลับเฟส	
รูปที่ 2.12 วงจรอินทิเกรตในรูปโดเมนของความถี่	18
รูปที่ 2.13 วงจรความถี่ผ่านตลอดที่เลื่อนเฟสจาก 0° ถึง -180°	19
รูปที่ 2.14 วงจรความถี่ผ่านตลอดที่เลื่อนเฟสจาก 180° ถึง 0°	19
รูปที่ 2.15 วงจรที่แยกออกมาพิจารณาจากรูปที่ 2.13	20
รูปที่ 2.16 วงจรที่แยกออกมาพิจารณาจากรูปที่ 2.14	20
รูปที่ 2.17 บล็อกไดอะแกรมของ Successive Approximation ADC	21
รูปที่ 2.18 (a) แสดงแรงดันอ้างอิงของแอปโซลูตโหมด	23
(b) แสดงแรงดันอ้างอิงของเรกโซเมตริกโหมด	
รูปที่ 2.19 วงจรการปรับซีโรสเกล	24
รูปที่ 2.20 วงจรกำเนิดสัญญาณนาฬิกาภายใน ADC	24
รูปที่ 2.21 วงจรการแปลงสัญญาณอย่างต่อเนื่อง	25
รูปที่ 2.22 ผังการทำงานโหมด 1	29
รูปที่ 2.23 การต่อขา Input Enable	34
รูปที่ 2.24 การถอดรหัสเมื่ออินพุต C,B,A = 0,0,0	35

	หน้า
รูปที่ 2.25 เมื่อป้อน C,B,A = 0,0,1	35
รูปที่ 2.26 การเชื่อมต่อกับตัวกลางการส่งผ่าน	36
รูปที่ 2.27 RS-232-C Connector	37
รูปที่ 2.28 แผนภาพสถานะของ RS-232-C	40
รูปที่ 2.29 null modem	42
รูปที่ 3.1 บล็อกไดอะแกรมของระบบสายอากาศปรับตัวได้	43
รูปที่ 3.2 บล็อกไดอะแกรมของอัลกอริทึมกำลังสองเฉลี่ยน้อยที่สุดแบบอนาล็อก	44
รูปที่ 3.3 เครื่องมืออนาล็อกสำหรับอัลกอริทึมกำลังสองเฉลี่ยน้อยที่สุด ที่มีการปรับแก้ค่าถ่วงน้ำหนัก	44
รูปที่ 3.4 วงจรคูณที่ทำการออกแบบ	44
รูปที่ 3.5 วงจรอินทิเกรตที่ทำการออกแบบ	45
รูปที่ 3.6 วงจรบวกที่ทำการออกแบบ	45
รูปที่ 3.7 วงจรลบที่ทำการออกแบบ	45
รูปที่ 3.8 รูปวงจรรวม	46
รูปที่ 3.9 วงจรสมมติสัญญาณ	47
รูปที่ 3.10 บล็อกไดอะแกรมแสดงส่วนการออกแบบ	47
รูปที่ 3.11 บล็อกไดอะแกรมแสดงการออกแบบในส่วนซอฟต์แวร์	48
รูปที่ 4.1 บล็อกไดอะแกรมแสดงแบบจำลองการทำงานของวงจร	49
รูปที่ 4.2 สัญญาณอินพุต 1 ของแบบจำลองการทำงานของวงจร	50
รูปที่ 4.3 สัญญาณอินพุต 2 ของแบบจำลองการทำงานของวงจร	50
รูปที่ 4.4 สัญญาณอินพุต 3 ของแบบจำลองการทำงานของวงจร	51
รูปที่ 4.5 สัญญาณอินพุต 4 ของแบบจำลองการทำงานของวงจร	51
รูปที่ 4.6 สัญญาณเอาต์พุตและสัญญาณคลาดเคลื่อนเมื่อใช้เวลา 1 มิลลิวินาที	52
รูปที่ 4.7 สัญญาณเอาต์พุตและสัญญาณคลาดเคลื่อนเมื่อใช้เวลา 3 มิลลิวินาที	52
รูปที่ 4.8 สัญญาณเอาต์พุตและสัญญาณคลาดเคลื่อนเมื่อใช้เวลา 5 มิลลิวินาที	53
รูปที่ 4.9 สัญญาณเอาต์พุตและสัญญาณคลาดเคลื่อนเมื่อใช้เวลา 10 มิลลิวินาที	53
รูปที่ 4.10 ช่องสัญญาณที่ 1 แสดงสัญญาณอินพุตของวงจรบัฟเฟอร์ ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตของวงจรบัฟเฟอร์	54
รูปที่ 4.11 ช่องสัญญาณที่ 1 และช่องสัญญาณอ้างอิง 1 แสดงสัญญาณอินพุต ของวงจรบวก ช่องสัญญาณที่ 2 แสดงเอาต์พุตของวงจรบวก	54
รูปที่ 4.12 ช่องสัญญาณที่ 1 แสดงอินพุตของวงจรกลับเฟส ช่องสัญญาณที่ 2 แสดงเอาต์พุตของวงจรกลับเฟส	55

	หน้า
รูปที่ 4.13 ช่องสัญญาณที่ 1 และช่องสัญญาณอ้างอิง 1 แสดงสัญญาณอินพุต ของวงจรลบ (เฟสตรงกัน)	55
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตของวงจรลบ	
รูปที่ 4.14 ช่องสัญญาณที่ 1 และช่องสัญญาณอ้างอิง 1 แสดงสัญญาณอินพุต ของวงจรลบ (เฟสตรงกันข้าม)	56
รูปที่ 4.15 ช่องสัญญาณที่ 1 แสดงสัญญาณอินพุตของวงจรคูณ	56
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตของวงจรคูณ	
รูปที่ 4.16 ช่องสัญญาณที่ 1 แสดงสัญญาณอินพุตของวงจรอินทิเกรต	57
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตของวงจรอินทิเกรต	
รูปที่ 4.17 การจัดวางอุปกรณ์ในการเปรียบเทียบ	58
รูปที่ 4.18 ช่องสัญญาณที่ 1 แสดงสัญญาณที่ต้องการ	59
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตเมื่อผ่านวงจร	
รูปที่ 4.19 ช่องสัญญาณที่ 1 แสดงสัญญาณที่ต้องการ	59
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตเมื่อผ่านวงจร	
โดยใส่อินพุต 1 ช่องสัญญาณ	
รูปที่ 4.20 ช่องสัญญาณที่ 1 แสดงสัญญาณที่ต้องการ	60
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตเมื่อผ่านวงจร	
โดยใส่อินพุต 2 ช่องสัญญาณ	
รูปที่ 4.21 ช่องสัญญาณที่ 1 แสดงสัญญาณที่ต้องการ	60
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตเมื่อผ่านวงจร	
โดยใส่อินพุต 3 ช่องสัญญาณ	
รูปที่ 4.22 ช่องสัญญาณที่ 1 แสดงสัญญาณเอาต์พุต	61
ช่องสัญญาณที่ 2 แสดงสัญญาณค่าความคลาดเคลื่อน	
รูปที่ 4.23 การจัดวางอุปกรณ์การทดลองระบบสายอากาศปรับตัวได้	61
รูปที่ 4.24 สัญญาณอินพุตที่ 1 ของส่วนประมวลสัญญาณ	62
รูปที่ 4.25 สัญญาณอินพุตที่ 2 ของส่วนประมวลสัญญาณ	62
รูปที่ 4.26 สัญญาณอินพุตที่ 3 ของส่วนประมวลสัญญาณ	63
รูปที่ 4.27 สัญญาณอินพุตที่ 4 ของส่วนประมวลสัญญาณ	63
รูปที่ 4.28 สายอากาศส่งทำมุม 0 องศา กับสายอากาศรับ	64
ช่องสัญญาณที่ 1 แสดงสัญญาณเอาต์พุตเมื่อเริ่มต้น	
ช่องสัญญาณที่ 2 แสดงสัญญาณค่าความคลาดเคลื่อน	

รูปที่ 4.29 สายอากาศส่งทำมุม 0 องศา กับสายอากาศรับ	64
ช่องสัญญาณที่ 1 แสดงสัญญาณเอาต์พุตเมื่อผ่านการประมวลผลสัญญาณ	
ช่องสัญญาณที่ 2 แสดงสัญญาณค่าความคลาดเคลื่อน	
รูปที่ 4.30 สายอากาศส่งทำมุม 30 องศา กับสายอากาศรับ	65
ช่องสัญญาณที่ 1 แสดงสัญญาณเอาต์พุตเมื่อเริ่มต้น	
ช่องสัญญาณที่ 2 แสดงสัญญาณค่าความคลาดเคลื่อน	
รูปที่ 4.31 สายอากาศส่งทำมุม 30 องศา กับสายอากาศรับ	65
ช่องสัญญาณที่ 1 แสดงสัญญาณเอาต์พุตเมื่อผ่านการประมวลผลสัญญาณ	
ช่องสัญญาณที่ 2 แสดงสัญญาณค่าความคลาดเคลื่อน	
รูปที่ 4.32 สายอากาศส่งทำมุม -30 องศา กับสายอากาศรับ	66
ช่องสัญญาณที่ 1 แสดงสัญญาณเอาต์พุตเมื่อเริ่มต้น	
ช่องสัญญาณที่ 2 แสดงสัญญาณค่าความคลาดเคลื่อน	
รูปที่ 4.33 สายอากาศส่งทำมุม -30 องศา กับสายอากาศรับ	66
ช่องสัญญาณที่ 1 แสดงสัญญาณเอาต์พุตเมื่อผ่านการประมวลผลสัญญาณ	
ช่องสัญญาณที่ 2 แสดงสัญญาณค่าความคลาดเคลื่อน	
รูปที่ 4.34 การจัดวางอุปกรณ์การทดลองการทำงานของส่วนประมวลผลสัญญาณ	67
แบบอนาล็อกโดยการสมมติสัญญาณอินพุต	
รูปที่ 4.35 สัญญาณอินพุตที่ 1 ที่รับจากวงจรสมมติสัญญาณ	68
รูปที่ 4.36 สัญญาณอินพุตที่ 2 ที่รับจากวงจรสมมติสัญญาณ	68
รูปที่ 4.37 สัญญาณอินพุตที่ 3 ที่รับจากวงจรสมมติสัญญาณ	69
รูปที่ 4.38 สัญญาณอินพุตที่ 4 ที่รับจากวงจรสมมติสัญญาณ	69
รูปที่ 4.39 ช่องสัญญาณที่ 1 แสดงสัญญาณเอาต์พุต	70
ช่องสัญญาณที่ 2 แสดงสัญญาณค่าความคลาดเคลื่อน	
รูปที่ 4.40 ช่องสัญญาณที่ 1 แสดงสัญญาณอ้างอิง	70
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตจากส่วนประมวลผลสัญญาณ	
รูปที่ 4.41 สัญญาณอินพุตที่ 1 ที่ป้อนให้กับส่วนแปลงสัญญาณอนาล็อกเป็นดิจิทัล	71
ตัวที่ 1	
รูปที่ 4.42 สัญญาณอินพุตที่ 2 ที่ป้อนให้กับส่วนแปลงสัญญาณอนาล็อกเป็นดิจิทัล	71
ตัวที่ 2	
รูปที่ 4.43 สัญญาณอินพุตที่ 3 ที่ป้อนให้กับส่วนแปลงสัญญาณอนาล็อกเป็นดิจิทัล	72
ตัวที่ 3	
รูปที่ 4.44 สัญญาณอินพุตที่ 4 ที่ป้อนให้กับส่วนแปลงสัญญาณอนาล็อกเป็นดิจิทัล	72
ตัวที่ 4	

	หน้า
รูปที่ 4.45 สัญญาณเอาต์พุตที่ขาไอซี MAX 232 จากส่วนแปลงสัญญาณอนาล็อก เป็นดิจิตอลตัวที่ 1	73
รูปที่ 4.46 สัญญาณเอาต์พุตที่ขาไอซี MAX 232 จากส่วนแปลงสัญญาณอนาล็อก เป็นดิจิตอลตัวที่ 1	73
รูปที่ 4.47 สัญญาณเอาต์พุตที่ขาไอซี MAX 232 จากส่วนแปลงสัญญาณอนาล็อก เป็นดิจิตอลตัวที่ 1	74
รูปที่ 4.48 สัญญาณเอาต์พุตที่ขาไอซี MAX 232 จากส่วนแปลงสัญญาณอนาล็อก เป็นดิจิตอลตัวที่ 1	74
รูปที่ 4.49 สัญญาณเอาต์พุตที่ขาไอซี MAX 232 จากส่วนแปลงสัญญาณอนาล็อก เป็นดิจิตอลทั้ง 4 ตัวพร้อมกัน	75
รูปที่ 4.50 สัญญาณอินพุตของส่วนประมวลสัญญาณแบบดิจิตอลที่สมมติขึ้น	75
รูปที่ 4.51 สัญญาณเอาต์พุตและสัญญาณกลาดเคลื่อนที่ได้จากส่วนประมวลสัญญาณ	76
รูปที่ 4.52 สัญญาณที่ต้องการและสัญญาณเอาต์พุตจากส่วนประมวลสัญญาณ	76
รูปที่ 4.53 แผนการกรองของการกรองแบบปรับตัวได้	77
รูปที่ 4.54 การตอบสนองความถี่ของการกรองแบบปรับตัวได้	77
รูปที่ 4.55 การจัดวางอุปกรณ์การทดลองในส่วนของการประมวลสัญญาณ โดยใช้ซอฟต์แวร์	78
รูปที่ 4.56 สัญญาณอินพุตที่ 1 ที่รับจากช่องสัญญาณที่ 1	79
รูปที่ 4.57 สัญญาณอินพุตที่ 2 ที่รับจากช่องสัญญาณที่ 2	79
รูปที่ 4.58 สัญญาณอินพุตที่ 3 ที่รับจากช่องสัญญาณที่ 3	80
รูปที่ 4.59 สัญญาณอินพุตที่ 4 ที่รับจากช่องสัญญาณที่ 4	80
รูปที่ 4.60 การเปรียบเทียบรูปสัญญาณระหว่าง (a) สัญญาณอ้างอิง (b) สัญญาณเอาต์พุตที่ผ่านการประมวลผลแล้ว	81
รูปที่ 4.61 รูปขยายของรูปที่ 4.60 (a) สัญญาณอ้างอิง (b) สัญญาณเอาต์พุตที่ผ่านการประมวลผลแล้ว	81

สารบัญตาราง

	หน้า
ตารางที่ 2.1 แสดงการเลือกโหมดทำงานของ SCON	27
ตารางที่ 2.2 แสดงการเลือกค่า TH1 สำหรับ Baud rate ต่างๆ	30
ตารางที่ 2.3 แสดงค่าความจริงของ 2 to 4 line decoder	32
ตารางที่ 2.4 3 to 8 line decoder	32
ตารางที่ 2.5 ตารางค่าความจริงของ 74LS138	33
ตารางที่ 2.6 วงจรสับเปลี่ยน RS-449 และ RS-232-C	38
ตารางที่ 2.7 ขบวนการได้อัลของ RS-232-C	41



บทที่ 1

บทนำ

ด้วยความก้าวหน้าทางเทคโนโลยีในด้านต่างๆ เช่น อุปกรณ์อิเล็กทรอนิกส์ คอมพิวเตอร์ วิทยาศาสตร์ ปัจจุบันเหล่านี้เป็นส่วนสำคัญที่นำมาพัฒนาด้านการสื่อสารโทรคมนาคม สำหรับเทคโนโลยีทางด้านโทรคมนาคมนั้นได้ถูกพัฒนามาเรื่อยๆ อย่างต่อเนื่อง เริ่มต้นจากการสื่อสารที่ต้องอาศัยสายส่งสัญญาณ จนกระทั่งพัฒนามาเป็นการสื่อสารแบบเคลื่อนที่ หรือที่รู้จักกันโดยทั่วไปก็คือโทรศัพท์มือถือ แต่ด้วยข้อจำกัดต่างๆ เช่น การเดินทางของคลื่นซึ่งอาจเกิดการเบี่ยงเบน การสะท้อน ซึ่งเกิดจากคุณสมบัติของคลื่น ทำให้ประสิทธิภาพของการสื่อสารด้อยลงไป

ดังนั้นเพื่อเป็นการแก้ปัญหาของการสื่อสารแบบเคลื่อนที่ จึงได้มีการพัฒนาสายอากาศชนิดหนึ่งขึ้นมาที่เรียกว่าสายอากาศแบบปรับตัวได้ (Adaptive Antenna) ซึ่งมีคุณสมบัติที่สามารถดัดสัญญาณรบกวนที่มีความถี่เดียวกัน แต่เข้ามาในทิศทางที่ไม่ได้ต้องการ ให้น้อยที่สุดเท่าที่จะเป็นไปได้ เพราะในความเป็นจริงจะไม่สามารถควบคุมหรือรู้ล่วงหน้าถึงคุณลักษณะของสภาวะแวดล้อมในขณะที่ระบบกำลังทำงานได้ และเนื่องจากเป็นสายอากาศแบบแถวลำดับ (Array) ทำให้สามารถกำจัดสัญญาณรบกวนได้มาก และคุณสมบัติของสายอากาศแบบแถวลำดับอีกอย่างหนึ่งคือทำให้การชี้ทิศทางของลำคลื่นหลัก อยู่ด้านหน้าในทิศทางที่ต้องการ

สายอากาศแบบปรับตัวได้จะปรับเปลี่ยนตัวเอง ให้สามารถทำงานได้ไม่ว่าจะอยู่ในสภาวะแวดล้อมแบบใด โดยค่าถ่วงน้ำหนัก (weight) ของระบบจะเปลี่ยนไปตามเวลา และปรับตัวเองให้ลำคลื่นหลักอยู่ในทิศทางและแบนด์วิดท์ตามสัญญาณนำร่อง (pilot signal) เพื่อให้เอาต์พุตของระบบมีลักษณะที่ต้องการ แม้ว่าอินพุตจะมีลักษณะใดๆก็ตาม

ด้วยความสามารถของสายอากาศแบบปรับตัวได้ที่ค่อนข้างมีประสิทธิภาพดีกว่าสายอากาศโดยทั่วไป ดังนั้น สายอากาศแบบปรับตัวได้ จึงไม่ได้เป็นแค่เทคโนโลยีของสายอากาศเท่านั้น แต่จะรวมทั้งระบบ ดังนั้น การที่จะออกแบบสายอากาศชนิดนี้จำเป็นต้องมีความรู้ในหลายๆด้าน เช่น สายอากาศ, การแพร่กระจายคลื่น การประมวลผลสัญญาณแบบปรับตัวได้ เป็นต้น ซึ่งโครงการนี้เป็นการศึกษาขั้นพื้นฐานของสายอากาศแบบปรับตัวได้ เพื่อเป็นแนวทางในการพัฒนาต่อไป

เนื่องจากโครงการนี้เป็นการศึกษาที่ต้องอาศัยความรู้ในหลายๆด้าน จึงได้มีการแบ่งกลุ่มการทำงานเป็น 2 กลุ่ม ซึ่งมีการทำงานที่สัมพันธ์กัน คือ กลุ่มที่หนึ่ง ทำการศึกษาและออกแบบเกี่ยวกับระบบของสายอากาศ ส่วนกลุ่มที่สองจะศึกษาและออกแบบวงจรที่ใช้ในการประมวลผลสัญญาณ

รายงานฉบับนี้จะเป็นส่วนของกลุ่มที่สอง ดังนั้นเนื้อหาส่วนใหญ่จึงเป็นการศึกษาและออกแบบวงจรประมวลผลสัญญาณ แต่ก็มีเนื้อหาส่วนที่เกี่ยวข้องกับระบบสายอากาศด้วย ถ้าหากว่าผู้อ่านสนใจที่จะศึกษาเพิ่มเติมก็สามารถศึกษาได้จากโครงการการออกแบบสายอากาศอาร์เรย์เพื่อประยุกต์ใช้ทำสายอากาศชนิดปรับตัว (DESIGN OF PHASED ARRAY FOR THE APPLICATION OF ADAPTIVE ANTENNA)

บทที่ 2

ทฤษฎีและหลักการ

2.1 การกรองตำแหน่งและทิศทาง (Directional and spatial filtering)

ตัวอย่างของสายอากาศรับแถวลำดับเชิงเส้น (linear-array receiving antenna) แสดงดังรูปที่ 2.1(a) และ 2.1(b) สายอากาศดังรูปที่ 2.1(a) ประกอบด้วยสายอากาศไอโซทรอปิก (isotropic) 7 องค์ประกอบวางห่างกันเป็นระยะ $\lambda_0/2$ ในแนวเส้นตรง โดยที่ λ_0 เป็นความยาวคลื่นที่ความถี่กลาง f_0 ของแถวลำดับ สัญญาณที่รับได้จะถูกรวมกันเป็นสัญญาณเอาต์พุตของแถวลำดับ แบบรูปเจาะจงทิศทาง (directivity pattern) เช่นความไวการตอบสนองสัมพัทธ์ต่อสัญญาณจากหลายทิศทาง (the relative sensitivity of response to signal from various directions) จะถูกพล็อตบนระนาบในช่วงมุม $-\pi/2 < \theta < \pi/2$ สำหรับความถี่ f_0 แบบรูปนี้ จะสมมาตรในแนวแกนตั้งที่ $\theta = 0$ พุคสันใหญ่ (main lobe) จะอยู่ตรงกลางที่ $\theta = 0$ แอมพลิจูดของพูข้าง (side lobe) ที่มากที่สุดที่ $\theta = 24^\circ$ ถ้าความถี่กลางไม่ใช่ f_0 แบบรูปที่ได้จะต่างจากแบบรูปนี้

แถวลำดับที่เหมือนกันดังรูปที่ 2.1(b) แสดงให้เห็นว่าการประวิงเวลาของสัญญาณเอาต์พุตในแต่ละองค์ประกอบของแถวลำดับก่อนที่จะนำไปรวมกัน แบบรูปเจาะจงทิศทางลัพท์มีทิศทางของพุคสันใหญ่ที่มุม ψ โดยที่

$$\psi = \sin^{-1} \left(\frac{\lambda_0 \delta f_0}{d} \right) = \sin^{-1} \left(\frac{c \delta}{d} \right) \quad (2.1)$$

โดยที่

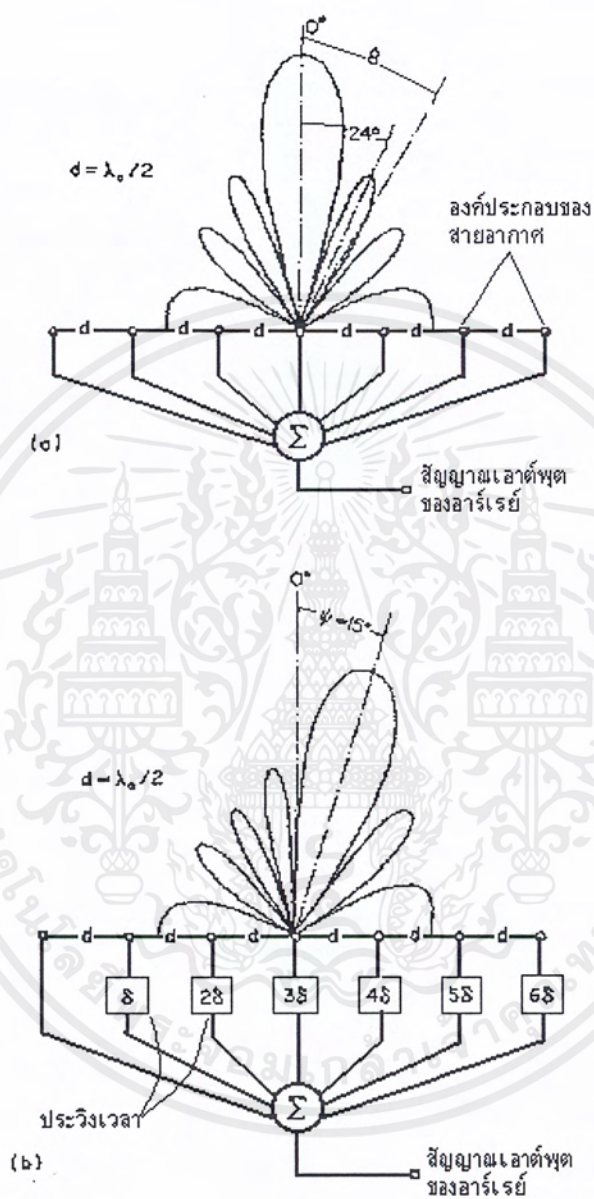
- f_0 คือความถี่ของสัญญาณที่รับได้
- λ_0 คือความยาวคลื่นที่ความถี่ f_0
- δ คือความแตกต่างของเวลาที่ประวิงระหว่างเอาต์พุตขององค์ประกอบข้างเคียง
- d คือระยะห่างระหว่างองค์ประกอบของสายอากาศ
- c คือความเร็วของการแพร่กระจายสัญญาณ เท่ากับ $\lambda_0 f_0$

ความไวที่มากที่สุดอยู่ที่มุม ψ เพราะสัญญาณที่รับได้จากแหล่งกำเนิดคลื่นระนาบจะตกกระทบที่มุมนี้ และถูกประวิงดังในรูปที่ 2.1(b) เพื่อให้ได้สัญญาณเอาต์พุตแรงที่สุด

รูปที่ 2.2(a) แสดงให้เห็นว่าการถ่วงน้ำหนักให้เอาต์พุตขององค์ประกอบของสายอากาศแต่ละตัวโดยแยกสัญญาณในแต่ละองค์ประกอบออกเป็น 2 ส่วน ส่วนหนึ่งเฟสยังคงเดิม อีกส่วนหนึ่งมีการเลื่อนเฟสไป 90° สัญญาณเอาต์พุตได้จากการรวมสัญญาณที่ถ่วงน้ำหนักแล้วเข้าด้วยกัน ถ้าน้ำหนักที่ถ่วงมีค่าเท่ากันหมด จะได้แบบรูปเจาะจงทิศทางที่สมมาตรกันเช่นเดียวกับในรูปที่ 2.1(a)

สมมติให้มีสัญญาณรบกวน (interference) แบบไซน์ซุซอิดัล (sinusoidal) ที่มีความถี่เท่ากับ f_0 ตกกระทบสายอากาศแถวลำดับดังรูปที่ 2.2(a) มุมที่สัญญาณรบกวนตกกระทบสายอากาศนี้ ตกอยู่บนพูข้างของสายอากาศพอดี ถ้าเปลี่ยนค่าถ่วงน้ำหนักมาเป็นดังในรูปที่ 2.2(b) จะทำให้แบบรูปเจาะจงทิศทางที่ความถี่ f_0 ดังในรูปที่ 2.2(b) จะเห็นว่าพุคสันใหญ่เกือบจะไม่เปลี่ยนแปลงแบบรูปจากในรูปที่ 2.1(a) และรูปที่ 2.2(b) เลย แต่พู

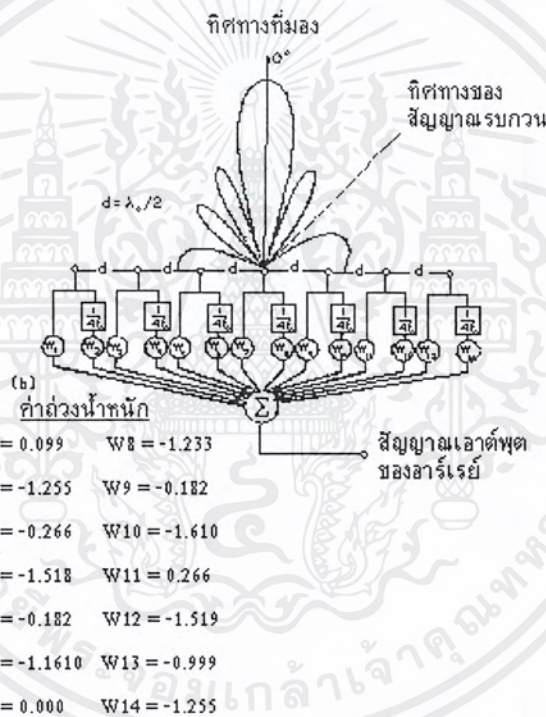
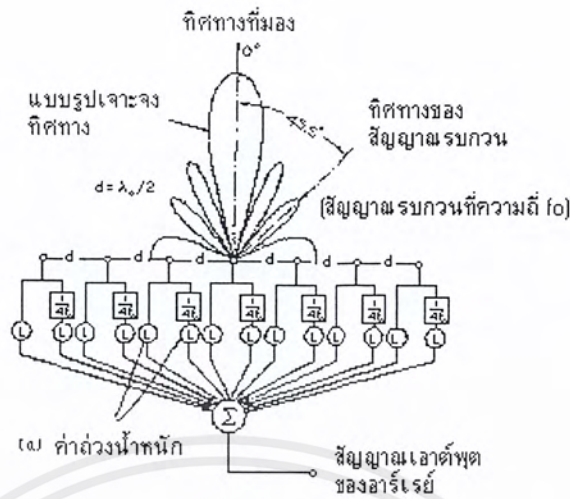
ข้างมีการเปลี่ยนแปลงแบบรูปเพื่อการจัดการรับสัญญาณรบกวนในรูปที่ 2.2(a) โดยมีการเปลี่ยนจุดที่สัญญาณเป็นศูนย์(null)ให้อยู่ในทิศทางที่สัญญาณรบกวนเข้ามา



รูปที่ 2.1 แบบรูปเจาะจงทิศทางสำหรับสายอากาศแถวลำดับเชิงเส้น

(a) แถวลำดับอย่างง่าย

(b) มีวงจรประวิงเวลา



รูปที่ 2.2 แบบรูปเจาะจงทิศทางสำหรับสายอากาศแถวลำดับเชิงเส้น

(a) การถ่วงน้ำหนักเท่ากัน

(b) การถ่วงน้ำหนักสำหรับวิธีการกำจัดสัญญาณรบกวน

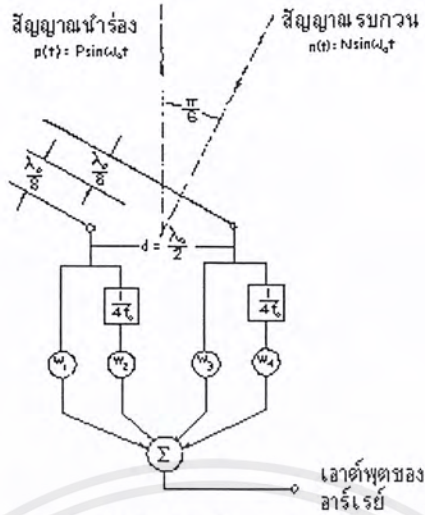
สมมติว่ามีสายอากาศแถวลำดับเชิงเส้นที่มีองค์ประกอบเป็นสายอากาศไอโซทรอปิก 2 องค์ประกอบที่วางองค์ประกอบห่างกันครึ่งหนึ่งของความยาวคลื่นที่สายอากาศรับ ให้สัญญาณเข้ามาในทิศทางที่ต้องการ (desired direction) ที่ $\theta = 0^\circ$ เรียกสัญญาณนี้ว่าสัญญาณนำร่อง (pilot signal) $p(t)$

$$p(t) = P \sin \omega_0 t \quad (2.2)$$

โดยที่ $\omega_0 = 2\pi f_0$ (2.3)

และให้สัญญาณอื่นที่เรียกว่าสัญญาณรบกวน $n(t)$ เข้ามาที่มุม $\theta = \pi/6$ เรเดียน

โดยที่ $n(t) = N \sin \omega_0 t$ (2.4)



รูปที่ 2.3 รูปลักษณะของแถวลำดับสำหรับตัวอย่างการกำจัดสัญญาณรบกวน

สัญญาณนำร่องและสัญญาณรบกวนมีความถี่เดียวกันคือ f_0 สมมติให้สัญญาณนำร่องและสัญญาณรบกวนมีเฟสตรงกัน ณ จุดตรงกลางระหว่างองค์ประกอบของสายอากาศ สัญญาณที่แต่ละองค์ประกอบรับได้จะแบ่งเป็น 2 ส่วนเพื่อถ่วงน้ำหนัก ส่วนแรกจะไม่ประวิงเวลาสัญญาณ อีกส่วนหนึ่งมีการประวิงเวลาของสัญญาณไป $1/(4f_0)$ สัญญาณเอาต์พุตของสายอากาศแถวลำดับจะได้อาจจากการรวมสัญญาณที่ถ่วงน้ำหนักทั้ง 4 สัญญาณรวมกัน

2.2 การคำนวณค่าถ่วงน้ำหนักเพื่อรับสัญญาณ $p(t)$ และกำจัดสัญญาณรบกวน $n(t)$

กำหนดให้ค่าถ่วงน้ำหนักไม่เป็นศูนย์ สัญญาณเอาต์พุตที่ได้จะอยู่ในรูปของ $A \sin(\omega_0 t + \phi)$ แต่อย่างไรก็ตาม สัญญาณเอาต์พุตของสายอากาศแถวลำดับจะต้องมีแอมพลิจูดและเฟสเป็นอิสระเพื่อที่จะกำจัดสัญญาณรบกวนได้

สัญญาณเอาต์พุตของสายอากาศแถวลำดับอันเนื่องมาจากสัญญาณนำร่องคือ

$$P[(w_1 + w_3) \sin \omega_0 t + (w_2 + w_4) \sin(\omega_0 t - \frac{\pi}{2})] \quad (2.5)$$

การที่จะทำให้สัญญาณเอาต์พุตเท่ากับสัญญาณเอาต์พุตที่ต้องการ $p(t)$ ดังสมการ (2.2) ได้นั้นจะต้องให้

$$w_1 + w_3 = 1 \quad \text{และ} \quad w_2 + w_4 = 0 \quad (2.6)$$

เวลาประวิงสัมพัทธ์ของสัญญาณรบกวนที่องค์ประกอบทั้งสองของสายอากาศหาได้จาก

$$\pm \left[\frac{1}{4f_0} \right] \sin \frac{\pi}{6} = \pm \left[\frac{1}{8f_0} \right] = \pm \frac{\lambda_0}{8c} \quad (2.7)$$

สัญญาณเอาต์พุตของสายอากาศแถวลำดับอันเนื่องมาจากสัญญาณรบกวนคือ

$$N[w_1 \sin(\omega_0 t - \frac{\pi}{4}) + w_2 \sin(\omega_0 t - \frac{3\pi}{4}) + w_3 \sin(\omega_0 t + \frac{\pi}{4}) + w_4 \sin(\omega_0 t - \frac{\pi}{4})] \quad (2.8)$$

การทำให้สัญญาณรบกวนมีขนาดเป็นศูนย์จะต้องทำให้

$$w_1 + w_4 = 0 \quad \text{และ} \quad w_2 - w_3 = 0 \quad (2.9)$$

โดยการแก้สมการ (2.6) และสมการ (2.9) จะได้คำตอบเป็น

$$w_1 = \frac{1}{2}, w_2 = \frac{1}{2}, w_3 = \frac{1}{2}, w_4 = -\frac{1}{2} \tag{2.10}$$

ค่าถ่วงน้ำหนักที่ได้ดังสมการ (2.10) ทำให้สายอากาศสามารถรับสัญญาณในทิศทางที่ต้องการได้โดยสามารถกำจัดสัญญาณรบกวนได้ด้วยแม้ว่าสัญญาณรบกวนนั้นจะมีความถี่เดียวกันกับสัญญาณที่ต้องการแต่ว่าสัญญาณรบกวนเดินทางมาคนละทิศกับสัญญาณที่ต้องการ

วิธีการดังกล่าวเหมาะสำหรับเมื่อมีจำนวนทิศทางของสัญญาณรบกวนน้อย เมื่อมีลักษณะของสัญญาณรบกวนรูปแบบเดียว (monochromatic) เมื่อทิศทางของสัญญาณรบกวนเป็นไปในแนวทางที่เรียกว่า *a priori* ตัวประมวลผล (processor) ในทางปฏิบัติไม่ต้องการทราบคุณสมบัติของสัญญาณรบกวน ดังนั้นเราจะใช้การประมวลผลแบบปรับตัว (adaptive processing) แทนซึ่งสามารถหาคำตอบของสมการเข้าไปเรื่อยๆ จนกว่าจะได้คำตอบที่มีค่าผิดพลาดระหว่างสัญญาณนำร่องกับสัญญาณเอาต์พุตน้อยที่สุด

2.3 รูปลักษณะของแถวลำดับองค์ประกอบปรับตัว(Configurations of adaptive arrays)

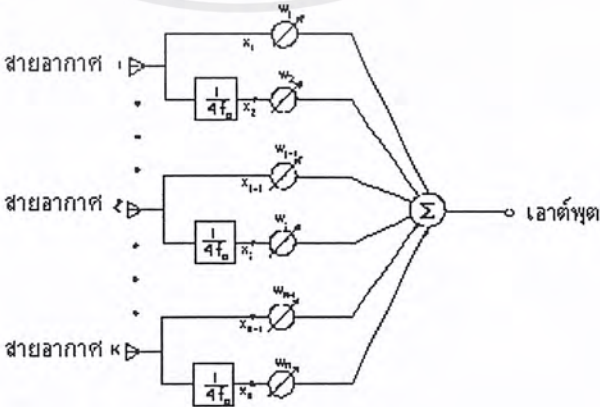
รูปลักษณะขององค์ประกอบปรับตัวสำหรับประมวลผลสัญญาณแถบแคบ (narrowband signal) แสดงดังรูปที่ 2.4 แต่ละองค์ประกอบของสายอากาศจะต่อกับตัวถ่วงน้ำหนักแบบปรับได้ (variable weight) และตัวประวิงเวลาหนึ่งในสี่ของคาบเวลา (a quarter – period time delay) ซึ่งต่อกับตัวถ่วงน้ำหนักแบบปรับได้ ถือว่าสัญญาณที่รับได้ทั้งองค์ประกอบของสายอากาศเป็นสัญญาณรูปแบบเดียว หรือสัญญาณแถบแคบ และจะถูกถ่วงน้ำหนักด้วยตัวประกอบอัตราขยายเชิงซ้อน(complex gain factor) $Ae^{j\phi}$

โดย $\phi = -\tan^{-1}(w_2/w_1)$ (2.11)

ซึ่งสามารถเลือกได้จากการตั้งค่าถ่วงน้ำหนักทั้งสองค่า และขนาด (magnitude) ของตัวประกอบอัตราขยายเชิงซ้อนคือ

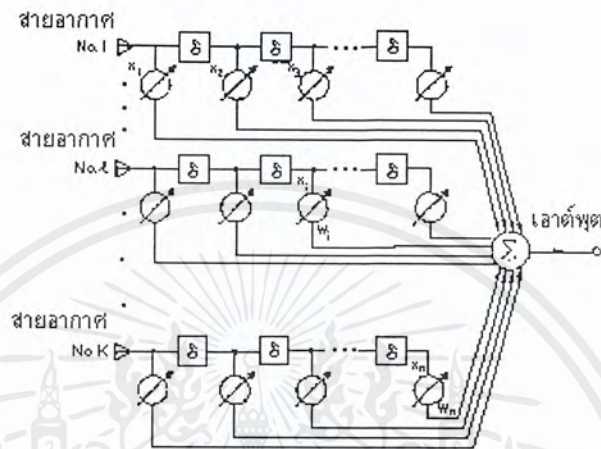
$$A = \sqrt{(w_1^2 + w_2^2)} \tag{2.12}$$

จากการที่สามารถกำหนดค่าถ่วงน้ำหนักได้ทำให้ค่าถ่วงน้ำหนักทั้งสองและตัวประวิงเวลาหนึ่งในสี่ของคาบเวลาสามารถประมวลผลแบบปรับตัวเชิงเส้นสำหรับสัญญาณแถบแคบที่รับได้ทีแต่ละองค์ประกอบของสายอากาศได้อย่างสมบูรณ์



รูปที่ 2.4 รูปลักษณะของแถวลำดับปรับตัวได้สำหรับการรับสัญญาณแถบแคบ

แถวลำดับเต็ม (full array) ในรูปที่ 2.4 แสดงให้เห็นถึงวิธีอย่างง่ายในการรวมสัญญาณจากแต่ละองค์ประกอบในโครงสร้างที่ปรับตัวแบบเชิงเส้นเมื่อสัญญาณที่รับได้และสัญญาณรบกวนเป็นสัญญาณแถบแคบ ควรตระหนักว่า จะได้รับสัญญาณแม้ว่าเวลาที่ประวิงจะไม่ทำให้เฟสเลื่อนไป $\pi/2$ ณ ความถี่กลาง f_0 การให้เฟสเลื่อนเข้าใกล้ $\pi/2$ จะช่วยให้ค่าถ่วงน้ำหนักมีค่าน้อยๆ แต่ก็ไม่จำเป็นมากนัก

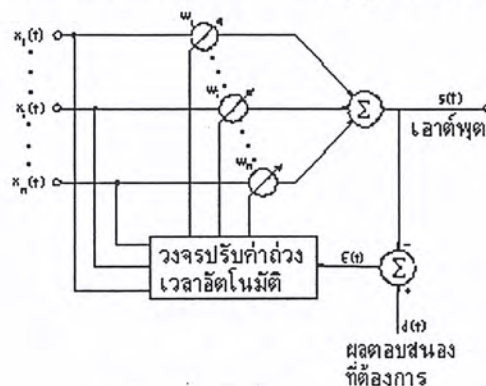


รูปที่ 2.5 รูปลักษณะของแถวลำดับแบบปรับตัวได้สำหรับการรับสัญญาณแถบกว้าง

ถ้าสนใจการรับสัญญาณในช่วงความถี่แถบกว้าง (wideband of frequencies) ตัวเลื่อนเฟสในรูปที่ 2.4 สามารถเปลี่ยนเป็นโครงข่ายสายประวิงเวลาแบบต่อแยก (tapped-delay-line network) ดังแสดงในรูปที่ 2.5 สายประวิงเวลาจะอนุญาตให้ปรับอัตราขยายและเฟสได้ตามต้องการ ถ้าระยะห่างของการต่อแยกใกล้พอ โครงข่ายนี้จะกลายเป็นวงจรกรองในอุดมคติที่อนุญาตให้ควบคุมอัตราการขยายและเฟสแต่ละความถี่ในย่านแถบผ่าน (passband) ได้อย่างสมบูรณ์

2.4 ตัวประมวลสัญญาณแบบปรับตัว (Adaptive Signal Processor)

เมื่อเลือกรูปแบบของโครงข่ายที่จะนำมาต่อที่แต่ละองค์ประกอบของสายอากาศได้แล้วตามรูปที่ 2.4 หรือรูปที่ 2.5 ขั้นตอนต่อไปคือการพัฒนาวิธีการปรับตัวซึ่งสามารถปรับได้อย่างอัตโนมัติโดยคุณค่าถ่วงน้ำหนักกับสัญญาณเพื่อให้ได้ตำแหน่งและความถี่ที่ต้องการในขั้นตอนของการปฏิบัติควรจะทำให้แถวลำดับมีอัตราการขยายในทิศทางที่ต้องการในขณะเดียวกันก็ต้องกำจัดสัญญาณรบกวนได้ด้วย



รูปที่ 2.6 องค์ประกอบปรับตัวได้มาตรฐาน

รูปที่ 2.6 แสดงรูปองค์ประกอบของตัวประมวลสัญญาณแบบปรับตัว ถ้าองค์ประกอบถูกรวมกับตัวควอนไทซ์สัญญาณเอาต์พุต มันจะรวมหน่วยตรรกะเริ่มเปลี่ยนการปรับตัว (adaptive threshold logic unit) องค์ประกอบดังกล่าวเรียกว่า Adaline หรือหน่วยตรรกะจุดเริ่มเปลี่ยน (threshold logic unit: TLU) การประยุกต์ใช้งานขององค์ประกอบจุดเริ่มเปลี่ยนการปรับตัวนั้นได้นำมาใช้ในการรู้จำแบบรูป (pattern recognition) และระบบควบคุมแบบปรับตัว (adaptive control system)

ในรูปที่ 2.6 นั้นสัญญาณอินพุต $x_1(t), \dots, x_i(t), \dots, x_n(t)$ เป็นสัญญาณที่เหมือนกันและถูกนำมาคูณกับค่าถ่วงน้ำหนัก $w_1, \dots, w_i, \dots, w_n$ ดังรูปที่ 2.4 และรูปที่ 2.5 เส้นที่แสดงเส้นทางการไหลของสัญญาณ (paths of signal flow) เส้นประแสดงฟังก์ชันที่สัมพันธ์กับการเปลี่ยนค่าการถ่วงน้ำหนักหรือวิธีการปรับตัว สัญญาณเอาต์พุต $s(t)$ ในรูปที่ 2.6 ได้จากผลรวมถ่วงน้ำหนักของสัญญาณ

$$s(t) = \sum_{i=1}^n x_i(t) w_i \quad (2.13)$$

โดยที่ n เป็นจำนวนของค่าถ่วงน้ำหนัก หรือเขียนในรูปเวกเตอร์ได้เป็น

$$s(t) = W^T X(t) \quad (2.14)$$

โดย W^T เป็นเวกเตอร์ค่าถ่วงน้ำหนักทรานสโพส

$$W = \begin{bmatrix} w_1 \\ \vdots \\ w_i \\ \vdots \\ w_n \end{bmatrix}$$

และเวกเตอร์สัญญาณอินพุตคือ

$$X(t) = \begin{bmatrix} x_1(t) \\ \vdots \\ x_i(t) \\ \vdots \\ x_n(t) \end{bmatrix}$$

สำหรับระบบดิจิทัลนั้นสัญญาณอินพุตจะเป็นรูปแบบข้อมูลที่สุ่มที่ไม่ต่อเนื่องทางเวลา (discrete-time sampled-data form) เขียนรูปเอาต์พุตได้เป็น

$$s(j) = W^T X(j) \quad (2.15)$$

เมื่อ j เป็นลำดับที่ของการสุ่มค่าสัญญาณ

ในขณะที่ทำการปรับตัวอยู่ สัญญาณที่ได้จากการตอบสนองที่ต้องการ (desired response) คือ $d(t)$ ถ้าเป็นสัญญาณต่อเนื่อง หรือคือ $d(j)$ ถ้าเป็นสัญญาณที่สุ่มค่า นั้นจะต้องสามารถป้อนให้กับองค์ประกอบปรับตัวได้ วิธีการรับสัญญาณเพื่อจะนำมาประมวลผลให้กับสายอากาศแถวลำดับปรับตัวได้จะกล่าวถึงในช่วงต่อไป

ความแตกต่างระหว่างการตอบสนองที่ต้องการกับการตอบสนองของสัญญาณเอาต์พุตทำให้เกิดเป็นสัญญาณผิดพลาด (error signal) $\mathcal{E}(j)$:

$$\mathcal{E}(j) = d(j) - W^T X(j) \quad (2.16)$$

สัญญาณนี้ถูกนำมาใช้เป็นสัญญาณควบคุมสำหรับวงจรปรับค่าถ่วงน้ำหนักของรูปที่ 2.6

2.5 วิธีการหาคำตอบสมการซิมูลเทนเนียส (Solving Simultaneous Equations)

จุดประสงค์หลักของการปรับตัวหรือการเปลี่ยนแปลงค่าถ่วงน้ำหนักก็เพื่อหาเซต(set)ของค่าถ่วงน้ำหนักที่ทำให้การตอบสนองเอาต์พุตขององค์ประกอบปรับตัวได้นั้นเท่ากับหรือใกล้เคียงกับการตอบสนองที่ต้องการมากที่สุด

ค่าความผิดพลาด(error) $\mathcal{E}(j)$ ในสมการ(2.16) สำหรับเวกเตอร์สัญญาณอินพุตที่แต่ละองค์ประกอบ $X(j)$ ควรจะมีค่าน้อยที่สุดเท่าที่จะเป็นไปได้

พิจารณาเซตจำกัด (finite set) ของสมการ simultaneous จึงเห็น

$$W^T X(1) = d(1)$$

$$W^T X(2) = d(2)$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$W^T X(j) = d(j)$$

$$\vdots$$

$$\vdots$$

$$\vdots$$

$$W^T X(N) = d(N)$$

(2.17)

โดยที่ N คือจำนวนเวกเตอร์ของสัญญาณอินพุต

มีสมการ N สมการสอดคล้องกับจำนวนของสัญญาณเอาต์พุตที่เกี่ยวข้องและมีตัวไม่ทราบค่า (unknown) จำนวน n ตัวและมีค่าถ่วงน้ำหนักจำนวน n ที่เป็นส่วนประกอบของ W

ถ้า N มีค่ามากกว่า n มากๆ จะพบว่าค่าตอบของเซตของสมการ N สมการที่มีค่าความผิดพลาดยกกำลังสองรวมกันได้น้อยที่สุดนั่นคือ

$$\sum_{j=1}^N \varepsilon^2(j) \tag{2.17}$$

เมื่อถือได้ว่าสัญญาณอินพุตเป็นตัวแปรพื้นสุ่มที่มีค่าคงที่ จะสนใจการหาเซตของค่าถ่วงน้ำหนักที่จะทำให้ค่าความผิดพลาดยกกำลังสองเฉลี่ยมีค่าน้อยที่สุดหรือค่าคาดหวัง (expected value) จำนวนดังกล่าวคิดจากค่าความผิดพลาดยกกำลังสอง นั่นคือ

$$E[\varepsilon^2(j)] = \overline{\varepsilon^2} \tag{2.18}$$

เซตของค่าถ่วงน้ำหนักที่ทำให้ค่าความผิดพลาดยกกำลังสองเฉลี่ยมีค่าน้อยที่สุดสามารถคำนวณได้จากการยกกำลังสองสมการ(2.16) ทั้งสองข้าง จะได้

$$\varepsilon^2(j) = d^2(j) + W^T X(j) X(j)^T W - 2 d(j) W^T X(j) \tag{2.19}$$

เมื่ออาศัยการใช้ค่าคาดหวังทั้ง 2 ข้าง ของสมการ ซึ่งค่าความผิดพลาดยกกำลังสองเฉลี่ย คือค่าคาดหวังของค่าความผิดพลาดยกกำลังสอง

$$\begin{aligned} E[\varepsilon^2(j)] &= E[d^2 + W^T X(j) X(j)^T W - 2 W^T d(j) X(j)] \\ &= E[d^2] + W^T \Phi_{(x,x)} W - 2 W^T \Phi_{(x,d)} \end{aligned} \tag{2.20}$$

โดยที่ $\Phi_{(x,x)} = E[X(j) X(j)^T] = E \begin{bmatrix} x_1 x_1 & x_1 x_2 & \dots & x_1 x_n \\ x_2 x_1 & & & x_2 x_n \\ \vdots & & & \vdots \\ x_n x_1 & & & x_n x_n \end{bmatrix}$ (2.21)

และ $\Phi_{(x,d)} = E[X(j) d(j)] = E \begin{bmatrix} x_1 d \\ x_2 d \\ \vdots \\ x_i d \\ \vdots \\ x_n d \end{bmatrix}$ (2.22)

เมตริกสมมาตร $\Phi(x, x)$ เป็นเมตริกสหสัมพันธ์ข้าม (cross correlation) และอัตสหสัมพันธ์ (auto correlate) ของสัญญาณอินพุตขององค์ประกอบปรับตัวได้ และเมตริกสหสัมพันธ์ $\Phi(x, d)$ เป็นเซตของสหสัมพันธ์ข้าม ระหว่าง อินพุต จำนวน n สัญญาณ และสัญญาณตอบสนองที่ออกแบบไว้

ค่าความผิดพลาดกำลังสองเฉลี่ย ที่กำหนดในสมการที่ (2.20) เป็นฟังก์ชันกำลังสองของค่าถ่วงน้ำหนัก ควรจะมีค่าน้อยลงเรื่อยๆ ในขณะที่ระบบปรับสถานะการทำงาน ซึ่งรูปกราฟของค่าความผิดพลาดยกกำลังสองเฉลี่ย ที่เป็นฟังก์ชันของจำนวนการทำซ้ำ เรียกว่า เส้นโค้งการเรียนรู้ (Learning curve) เนื่องจากเส้นโค้งนี้แสดงความสามารถของระบบในการทำงานตาม (track) สัญญาณผลตอบสนองที่ต้องการ ส่วนประกอบของเกรเดียนต์ของฟังก์ชันค่าความผิดพลาดกำลังสองเฉลี่ย เป็นอนุพันธ์ส่วนย่อยของค่าความผิดพลาดกำลังสองเฉลี่ยที่เกี่ยวข้องกับค่าถ่วงน้ำหนัก การหาอนุพันธ์(2.20) เกี่ยวข้องกับ W จะใช้เกรเดียนต์ $\nabla E[\varepsilon^2]$ ฟังก์ชันเชิงเส้นของค่าถ่วงน้ำหนัก

$$\nabla E[\varepsilon^2] = 2\Phi(x, x)W - 2\Phi(x, d) \quad (2.23)$$

เมื่อตัวเลือกของค่าถ่วงน้ำหนักได้ค่าดีที่สุด เกรเดียนต์จะเป็นศูนย์ จะได้ว่า

$$\begin{aligned} \Phi(x, x)W_{LMS} &= \Phi(x, d) \\ W_{LMS} &= \Phi^{-1}(x, x)\Phi(x, d) \end{aligned} \quad (2.24)$$

เวกเตอร์ถ่วงน้ำหนักที่ดีที่สุด W_{LMS} เป็นส่วนหนึ่งในค่าความผิดพลาดกำลังสองเฉลี่ยน้อยที่สุด สมการที่ (2.24) เป็นสมการWiener-Hopf และเป็นสมการสำหรับวงจรกรองกำลังสองน้อยที่สุดหลายช่องสัญญาณ Burg และ Claerbout ใช้ในกระบวนการของข้อมูลแถวลำดับไซสมิกแบบดิจิตอล (Digital seismic array data)

ทางหนึ่งที่จะหาเซตที่ดีที่สุดของค่าถ่วงน้ำหนักคือพิสูจน์สมการที่ (2.24) คำตอบนี้เป็นค่าที่ได้โดยตรงแบบธรรมดา แต่ก่อให้เกิดปัญหาการคำนวณที่ยาก เมื่อจำนวนของค่าถ่วงน้ำหนัก n มีมาก และเมื่ออัตราข้อมูลสูง ความจำเป็นในการกลับเมตริกซ์ $n \times n$ วิธีนี้อาจจะต้องการวัดแบบอัตสหสัมพันธ์ และสหสัมพันธ์ข้าม เป็นจำนวน $n(n+1)/2$ เพื่อที่จะทำได้องค์ประกอบของ $\Phi(x, x)$ ยิ่งกว่านั้น กระบวนการนี้ต้องการทำซ้ำอย่างต่อเนื่องในหลายสถานการณ์ที่เหมาะสม เมื่อสถิติของสัญญาณอินพุตเปลี่ยนไปอย่างช้าๆ คำตอบที่ไม่สมบูรณ์ของสมการที่ (2.24) เป็นไปได้ที่จะเหมาะสม เพราะความเป็นจริงที่การชักตัวอย่างทางสถิติแบบอนันต์ จะต้องการการประมาณค่าส่วนประกอบของเมตริกซ์สหสัมพันธ์ที่สมบูรณ์

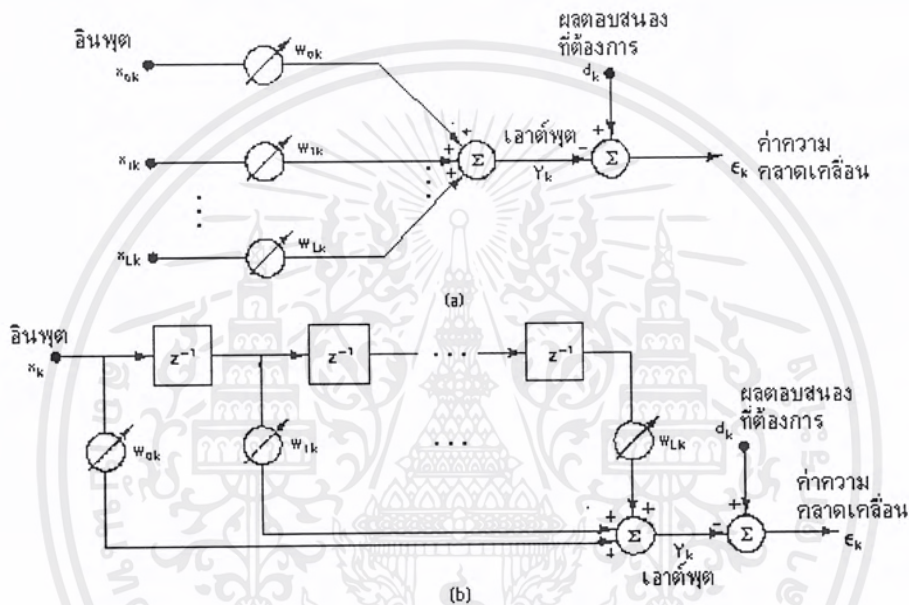
วิธีที่จะหาคำตอบของสมการที่ (2.24) โดยประมาณจะแสดงในหัวข้อต่อไป ความแน่นอนจะขึ้นอยู่กับขนาดของการชักตัวอย่างทางสถิติ ตั้งแต่การหาค่าถ่วงน้ำหนักขึ้นกับการวัดด้วยเวลาจำกัดของสัญญาณข้อมูลอินพุต วิธีนี้ไม่ต้องการการวัดที่แน่นอนของ ฟังก์ชันสหสัมพันธ์ หรือการกลับเมตริกซ์ วิธีนี้จะขึ้นอยู่กับเทคนิคการหาเกรเดียนต์ซึ่งประยุกต์เป็นฟังก์ชันค่าความผิดพลาดกำลังสองเฉลี่ย นั่นคือ อัลกอริทึมค่าความผิดพลาดกำลังสองเฉลี่ยน้อยที่สุด ซึ่งไม่ต้องการการยกกำลัง การเฉลี่ย หรือการหาอนุพันธ์เพื่อให้ได้เกรเดียนต์ของฟังก์ชันค่าความผิดพลาดกำลังสองเฉลี่ย

2.6 ความเป็นมาของ อัลกอริทึมกำลังสองเฉลี่ยน้อยที่สุด (LMS Algorithm)

Widrow [1] ได้ให้ความเห็นไว้ว่า อัลกอริทึมกำลังสองเฉลี่ยน้อยที่สุด เป็นวิธีที่ดีที่สุด สำหรับระบบที่มีโครงสร้างเป็น วงจรรวมแบบเชิงเส้น ผลรวมเชิงเส้นแบบปรับตัวได้ (Adaptive linear combiner) สามารถใช้ได้ 2 แบบ ขึ้นกับลักษณะของอินพุตที่เข้ามา ว่าเป็นแบบขนาน หรืออนุกรม (แล้วแต่จำนวนอินพุต) จากรูปที่ 2.7 เอาต์พุตที่ได้จากการรวมกันในแต่ละแบบคือ y_k และได้สมการเป็น

$$\varepsilon_k = d_k - X_k^T W_k \quad (2.25)$$

เมื่อ X_k เป็นเวกเตอร์ของอินพุต



รูปที่ 2.7 ผลรวมเชิงเส้นแบบปรับตัวได้

(a) แบบขนาน

(b) แบบตัวกรองความถี่เชิงแนวขวาง

หลักการของการเข้าสู่จุดต่ำสุดในฟังก์ชันประเมิณผล (performance function) ของวงจรรวมเชิงเส้น คือ ปรับค่าถ่วงน้ำหนักเพื่อให้ค่าความผิดพลาดยกกำลังสองเฉลี่ยมีค่าน้อยที่สุด โดยปรับค่าถ่วงน้ำหนักแต่ละขณะเวลาตามทิศทางตรงข้ามกับเกรเดียนต์ ของฟังก์ชันประเมิณผล

ในการพัฒนาอัลกอริทึมการปรับตัว ในวิธีก่อนจะทำการประมาณค่าเกรเดียนต์ของ $\xi = E[\varepsilon_k^2]$ โดย ใช้ความต่างระหว่างค่าเฉลี่ยเทอมสั้น (short-term averages) ของ ε_k^2 สำหรับการพัฒนา อัลกอริทึมกำลังสองเฉลี่ยน้อยที่สุด จะใช้ ε_k^2 ประมาณค่า ξ_k จากนั้นในแต่ละขั้นตอนจะมีค่าประมาณของเกรเดียนต์

$$\hat{\nabla}_k = \begin{bmatrix} \frac{\partial \varepsilon_k^2}{\partial w_0} \\ \vdots \\ \frac{\partial \varepsilon_k^2}{\partial w_L} \end{bmatrix} = 2\varepsilon_k \begin{bmatrix} \frac{\partial \varepsilon_k^2}{\partial w_0} \\ \vdots \\ \frac{\partial \varepsilon_k^2}{\partial w_L} \end{bmatrix} = -2\varepsilon_k X_k \quad (2.26)$$

ค่า ε_k ได้มาจากค่าถ่วงน้ำหนัก(Weight) ตามสมการที่ (2.25)

จากการประมาณค่าเกรเดียนต์ สามารถจะกำหนดชนิดการลดระดับค่าสูงสุด (steepest descent type) ของอแดปทีฟ อัลกอริทึม ได้ว่า

$$\begin{aligned} W_{k+1} &= W_k - \mu \hat{\nabla}_k \\ &= W_k + 2\mu \varepsilon_k X_k \end{aligned} \quad (2.27)$$

นี่คือ อัลกอริทึมกำลังสองเฉลี่ยน้อยที่สุด โดยที่ μ เป็นค่าคงที่การขยาย ที่กำหนดความเร็วและความสม่ำเสมอของการปรับตัวตั้งแต่ค่าถ่วงน้ำหนักเปลี่ยน ในแต่ละขั้นตอน ซึ่งขึ้นกับการประมาณค่า เกรเดียนต์ที่ไม่สมบูรณ์ ก่อให้เกิดสัญญาณรบกวน

จากสมการที่ (2.27) จะพบว่า อัลกอริทึมกำลังสองเฉลี่ยน้อยที่สุด สามารถทำได้โดยไม่ต้องอาศัยการยกกำลัง การเฉลี่ย หรือการหาอนุพันธ์ แต่อย่าง และมีคุณภาพแต่ละส่วนของเกรเดียนต์ เวกเตอร์ ได้จากข้อมูลเดี่ยว โดยไม่เกี่ยวกับค่าถ่วงน้ำหนัก การที่ไม่มีการเฉลี่ย ส่วนของเกรเดียนต์จะมีส่วนของสัญญาณรบกวนที่มาก แต่สัญญาณรบกวนจะลดลงตามเวลา จากขั้นตอนปรับตัว ที่ทำหน้าที่คล้ายวงจรกรองความถี่ต่ำผ่าน (Low pass filter : LPF)

2.7 คอนเวอร์เจนซ์ของเวกเตอร์ถ่วงน้ำหนัก

ส่วนแรกที่เกี่ยวข้องกับ อัลกอริทึมกำลังสองเฉลี่ยน้อยที่สุด คือคอนเวอร์เจนซ์ เพื่อให้ได้เวกเตอร์ถ่วงน้ำหนักที่ดีที่สุด เมื่อ $E[\varepsilon_k^2]$ มีขนาดเล็กที่สุด พิจารณา กำลังสองเฉลี่ยน้อยที่สุด คอนเวอร์เจนซ์ อันดับแรก ค่าประมาณของเกรเดียนต์ตามสมการที่ (2.26) มักจะแสดงว่าไม่ถูกไบแอส เมื่อเวกเตอร์ถ่วงน้ำหนักเป็นค่าคงที่ค่าที่ได้จากสมการที่ (2.26) โดยให้ W_k มีค่าเท่ากับ W

$$\begin{aligned} E[\hat{\nabla}_k] &= -2E[\varepsilon_k X_k] \\ &= -2E[d_k X_k - X_k X_k^T W] \\ &= 2(RW - P) = \nabla \end{aligned} \quad (2.28)$$

ในบรรทัดที่ 2 ของสมการที่ (2.28) นี้ มาจากสมการที่ (2.25) ที่ในความเป็นจริง ε_k จะเป็นปริมาณสเกลาร์ สามารถนำมากระจายในสมการได้ ส่วนในบรรทัดที่ 3 ของสมการ ได้มาจาก

$$\begin{aligned} R &= E[X_k X_k^T] \\ P &= E[d_k X_k] \end{aligned}$$

เมื่อค่ากลางของ $\hat{\nabla}_k$ เท่ากับเกรเดียนต์ที่ถูกต้อง (∇), $\hat{\nabla}_k$ จะเป็นค่าประมาณเมื่อไม่ถูกไบแอส

ค่าประมาณเกรเดียนต์ที่ไม่ถูกไบแอส สามารถทำให้ อัลกอริทึมกำลังสองเฉลี่ยน้อยที่สุด เป็น steepest-descent algorithm ที่ถูกต้อง อย่างน้อยในกรณีที่มีการกำหนดโดยประมาณค่า ∇ ในแต่ละขั้นจากสมการที่ (2.26) แต่ไม่เปลี่ยนแปลงค่าเวทจนกว่าหลายๆขั้นตอนจะเกิดขึ้น วิธีนี้จะทำให้ $\hat{\nabla}_k$ มีค่าเข้าใกล้ ∇_k

จากสมการที่ (2.27) เป็นไปได้ว่า เวกเตอร์ถ่วงน้ำหนัก W_k เป็นฟังก์ชันที่เกี่ยวกับอินพุตเวกเตอร์ $X_{k-1}, X_{k-2}, \dots, X_0$ ถ้าเราให้อินพุตเวกเตอร์ต่อเนื่อง เป็นอิสระไม่ขึ้นกับเวลา W_k จะไม่ขึ้นกับ X_k สำหรับขั้นตอนของอินพุตคงที่ ในลักษณะนี้ค่าที่ต้องการของเวกเตอร์ถ่วงน้ำหนัก $E[W_k]$ หลังจากผ่านการทำซ้ำอย่างเพียงพอแล้ว สามารถแสดงได้ดังนี้คือ $W^* = R^{-1}P$

ทำให้เป็นค่าที่ต้องการทั้ง 2 ด้านของสมการที่ (2.27) จะได้สมการเป็น

$$\begin{aligned} E[W_{k+1}] &= E[W_k] + 2\mu E[\varepsilon_k X_k] \\ &= E[W_k] + 2\mu (E[d_k X_k] - E[X_k X_k^T W_k]) \end{aligned} \quad (2.29)$$

จากเงื่อนไขที่กำหนดไว้ X_k และ W_k เป็นอิสระ จะได้ค่าดังสมการที่ (2.28) ค่าเวกเตอร์ถ่วงน้ำหนักที่ดีที่สุดจากสมการ $W^* = R^{-1}P$ จะสามารถเขียนสมการที่ (2.29) ได้เป็น

$$\begin{aligned} E[W_{k+1}] &= E[W_k] + 2\mu (P - RE[W_k]) \\ &= (I + 2\mu R)E[W_k] + 2\mu RW^* \end{aligned} \quad (2.30)$$

เมื่อได้รูปแบบที่ต้องการ ซึ่งจะถูกพิสูจน์โดยการเปลี่ยนเป็นระบบพิกัดแกนหลัก (Principle-axis coordinate system) โดยใช้ค่าที่ต้องการ ได้สมการเป็น

$$E[V'_k] = (I - 2\mu\Lambda)^k V'_0 \quad (2.31)$$

เมื่อ V' เป็นเวกเตอร์ถ่วงน้ำหนัก (W) ในระบบแกนหลัก Λ เป็นเมตริกซ์ของค่าเจาะจงเชิงเส้นทแยงมุม (diagonal eigenvalue matrix) ของ R และ V'_0 เป็นค่าเวกเตอร์ถ่วงน้ำหนักเริ่มต้น ในระบบนี้

ดังนั้นเมื่อ k เพิ่มขึ้นโดยไม่มีขอบเขต ค่าเวกเตอร์ถ่วงน้ำหนักที่ต้องการจากสมการที่ (2.31) จะเข้าใกล้คำตอบที่ดี (เช่นเป็น 0 ในระบบพิกัดแกนร่วม) ต่อเมื่อด้านขวาของสมการเข้าใกล้ 0 ซึ่งค่าคอนเวอร์เจนซ์จะยอมรับได้ก็ต่อเมื่อ

$$\frac{1}{\lambda_{\max}} > \mu > 0 \quad (2.32)$$

เมื่อ $\lambda_{\max}$ เป็นค่าเจาะจงที่ใหญ่ที่สุด นั่นคือ เป็นส่วนประกอบเส้นทแยงมุมที่ใหญ่ที่สุดในเมตริกซ์ Λ

จากสมการที่ (2.32) จะมีขอบเขตของ μ สำหรับคอนเวอร์เจนซ์ของเวกเตอร์ถ่วงน้ำหนัก เป็นเวกเตอร์ถ่วงน้ำหนักที่ดีที่สุด จากขอบเขตนี้ ความเร็วในการปรับตัว และสัญญาณในคำตอบของเวกเตอร์ถ่วงน้ำหนัก จะขึ้นกับขนาดของ μ โดยที่ $\lambda_{\max}$ จะไม่มากกว่า Trace of R ซึ่งเป็นผลรวมของส่วนประกอบของเส้นทแยงมุมของเมตริกซ์ R คือ

$$\begin{aligned} \lambda_{\max} &\leq tr[\Lambda] = \sum(\text{diagonal element of } \Lambda) \\ &= \sum(\text{diagonal element of } R) = tr[R] \end{aligned} \quad (2.33)$$

นอกเหนือจากนั้น ด้วยตัวกรองความถี่แบบปรับค่าได้เชิงแนวขวาง (transversal adaptive filter) ให้ $tr[R]$ เป็น $(L+1)E[x_k^2]$ หรือ $L+1$ คูณกับกำลังของสัญญาณที่เข้ามา ดังนั้นคอนเวอร์เจนซ์ของเวกเตอร์ถ่วงน้ำหนัก กำหนดโดย

$$\text{แบบธรรมดา} \quad 0 < \mu < \frac{1}{tr[R]}$$

$$\text{แบบตัวกรองความถี่เชิงแนวขวาง} \quad 0 < \mu < \frac{1}{(L+1)(\text{signal power})} \quad (2.34)$$

นี่คือข้อจำกัดอื่นของขอบเขตของ μ นอกจากสมการที่ (2.32) และสมการนี้สามารถประยุกต์ได้ง่าย เพราะส่วนประกอบของ R และกำลังของสัญญาณสามารถประมาณค่าได้ง่ายกว่า ค่าเจาะจงของ R

2.8 วงจรบวก

รูปที่ 2.8 แสดงวงจรออปแอมป์ที่ใช้ในการบวกรูปคลื่นอนาล็อกโดยที่ $v_2 = 0$ และวงจรทำงานในช่วงเชิงเส้น $v_1 = v_2 = 0$ และโดยที่ $i_1 = 0$ เราสามารถเขียนได้ว่า

$$\frac{v_{s1}}{R_1} + \frac{v_{s2}}{R_2} = -\frac{v_0}{R_3} \quad (2.35)$$

$$\text{หรือ} \quad v_0 = -\frac{R_3}{R_1} v_{s1} + \frac{R_3}{R_2} v_{s2} \quad (2.36)$$

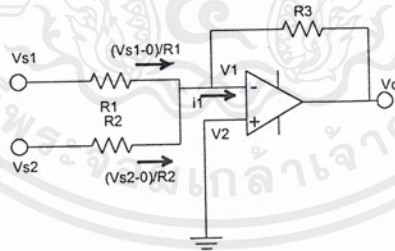
$$\text{ถ้า } R_1 = R_2 = R$$

$$v_0 = -\frac{R_3}{R} (v_{s1} + v_{s2}) \quad (2.37)$$

ดังนั้นแรงดันออก v_0 จะเท่ากับค่าคงตัว $-\frac{R_3}{R}$ คูณกับผลบวกของแรงดันเข้าทั้งสอง

$$\text{ถ้า } R_1 = R_2 = R_3$$

$$v_0 = -(v_{s1} + v_{s2}) \quad (2.38)$$



รูปที่ 2.8 วงจรบวก

2.9 วงจรลบ

ในวงจรลบนี้เรากำหนดให้ $R_3 = R_1$ และ $R_4 = R_2$ เพื่อความง่าย ในกรณีนี้แรงดันที่ขั้วเข้าทั้งสอง $v_1 = v_2 \neq 0$ ดังนั้นสมการของกระแสใน R_1 และ R_2 คือ

$$\frac{v_s - v_1}{R_1} = \frac{v_1 - v_0}{R_2} \quad (2.39)$$

และสมการของกระแสใน R_3 และ R_4 คือ

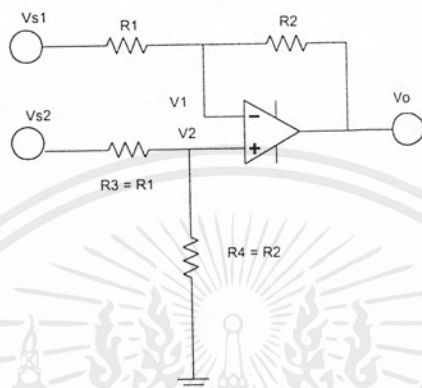
$$\frac{v_{s2} - v_1}{R_3} = \frac{v_1}{R_4} \quad (2.40)$$

แต่ $R_3 = R_1$, $R_4 = R_2$ จากสมการ เราได้

$$v_0 = \frac{R_2}{R_1} (v_{s2} - v_{s1}) \quad (2.41)$$

ถ้า $R_1 = R_2 = R$ จะได้

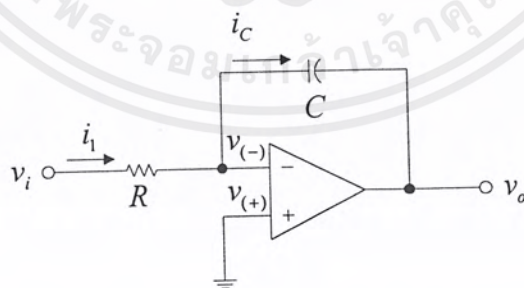
$$v_0 = v_{s2} - v_{s1} \quad (2.42)$$



รูปที่ 2.9 วงจรลบ

2.10 วงจรอินทิเกรต

วงจรรวมอินทิเกรตนั้น โดยปกติจะมีการออกแบบมาพร้อมกับตัวต้านทาน R และตัวเก็บประจุ C เพื่อต้องการทำให้ได้ค่ากระแสในการประจุ $R C$ มีค่าคงที่ ทำให้ได้วงจรรวมอินทิเกรต 2 ลักษณะ คือ วงจรรวมอินทิเกรตแบบไม่กลับเฟส (Non-inverting integrator) และวงจรรวมอินทิเกรตแบบกลับเฟส (Inverting integrator) แต่โดยทั่วไปจะมีการนิยมนำวงจรรวมอินทิเกรตแบบกลับเฟสมากกว่า เนื่องจากใช้งานง่าย และใช้อุปกรณ์ต่อภายนอกน้อยกว่า ดังนั้นจึงขอกล่าวถึงเฉพาะวงจรรวมอินทิเกรตแบบนี้นั่น



รูปที่ 2.10 ลักษณะ โครงสร้างของวงจรรวมอินทิเกรตแบบกลับเฟส

โครงสร้างของวงจรรวมอินทิเกรตแบบกลับเฟส ดังแสดงในรูปที่ 2.10 นี้ อาจจะเรียกอีกอย่างว่า วงจรมิลเลอร์อินทิเกรเตอร์ (Miller integrator) เมื่อให้กฎของเคอร์ชอฟ (Kirchoff's law) และลักษณะของออปแอมป์ที่ว่าอินพุตอิมพีแดนซ์มีค่าสูงมากจนสามารถประมาณได้ว่ากระแสไม่ไหลผ่านระหว่างขั้วอิน

พุดทั้งสองได้ จึงทำให้ได้ค่าของกระแส $I(+)$ เท่ากับศูนย์ และส่งผลให้กระแส $i_1 = i_c$ จากวงจรในรูปที่ 2.10 จะได้กระแส i_1 ดังนี้

$$i_1 = \frac{v_i}{R} \quad (2.43)$$

และ $v_0 + v_c = 0 \quad (2.44)$

เนื่องจาก $v_c = \frac{1}{C} \int_0^1 i_1 d\tau + V(0) \quad (2.45)$

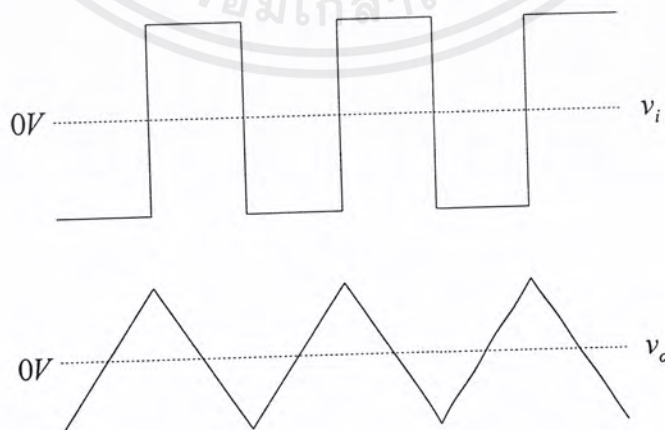
โดยที่ $V(0)$ คือค่าแรงดันเริ่มต้นในตัวเก็บประจุ

เมื่อแทนค่า i_1 และ v_c ลงในสมการ (2.44) แล้วจัดรูปสมการใหม่ จะได้ว่า

$$v_0 = -\left(\frac{1}{RC} \int_0^1 v_i d\tau + V(0) \right) \quad (2.46)$$

สมการ (2.46) นี้ แสดงให้เห็นว่า ค่าแรงดันเอาต์พุตของวงจรแปรผันตามค่าลบของอินทิกรัลสัญญาณอินพุต v_i และแปรผกผันกับค่า RC เครื่องหมายลบในสมการเอาต์พุต บอกให้ทราบว่าค่าแรงดันเอาต์พุต v_0 จะมีขั้ว (Polarity) ตรงกันข้ามกับแรงดันอินพุต ดังนั้นวงจรอินทิเกรตแบบนี้จึงได้ชื่อว่า วงจรอินทิเกรตแบบกลับเฟส

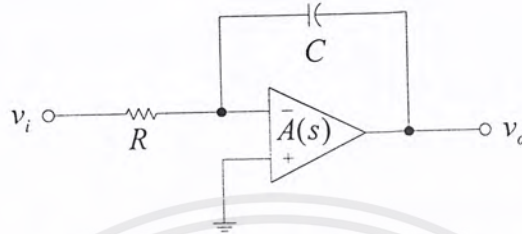
เมื่อทำการสมมติว่า v_i เป็นค่าคงที่ที่มีคาบเวลาหรือเป็นสัญญาณสี่เหลี่ยม และ R เป็นตัวต้านทานที่มีค่าคงตัว เมื่ออาศัยคุณสมบัติของออปแอมป์ที่ว่าอินพุตอิมพีแดนซ์มีค่าสูงมากจนสามารถประมาณได้ว่ากระแสไม่สามารถไหลผ่านระหว่างขั้วอินพุตทั้งสอง ดังนั้น i_1 จึงสามารถสมมติได้ว่ามีค่าคงที่ และเนื่องจากคุณสมบัติอินพุตอิมพีแดนซ์ของออปแอมป์ จึงประมาณได้ว่า ค่ากระแส i_1 ทั้งหมดไหลผ่าน C จึงทำให้อัตราการประจุและคายประจุของ C มีค่าคงที่ตามต้องการได้ เนื่องจากวงจรอินทิเกรตเป็นแบบกลับเฟส เอาต์พุตที่ได้จึงมีขั้วตรงกันข้ามกับแรงดันอินพุต ตัวอย่างความสัมพันธ์ของสัญญาณอินพุต v_i และสัญญาณเอาต์พุต v_0 มีลักษณะดังรูปที่ 2.11



รูปที่ 2.11 ความสัมพันธ์ของสัญญาณอินพุต v_i และสัญญาณเอาต์พุต v_0
วงจรอินทิเกรตเป็นแบบกลับเฟส

จากรูปสัญญาณดังแสดงในรูปที่ 2.11 จะพบว่าสัญญาณทั้งสองมีจุดกึ่งกลางอยู่ที่ศูนย์โวลต์ แต่ในทางปฏิบัติการอินทิเกรตสัญญาณต่อเนื่องอย่างนี้ อาจจะต้องทำการชดเชยค่าต่างๆ ให้กับวงจรหลายอย่าง เช่น ต้องมี Breeding resistor มาช่วยระบายประจุที่ตกคร่อม C

นอกจากนี้ หากพิจารณาวงจรอินทิเกรตในโดเมนของความถี่ (Frequency domain) ตามที่แสดงในรูปที่ 2.12



รูปที่ 2.12 วงจรอินทิเกรต ในรูปโดเมนของความถี่
วงจรอินทิเกรตในรูปที่ 2.12 จะมีฟังก์ชันถ่ายโอน (Transfer function) ดังนี้

$$A(s) = \frac{GB}{s} \quad (2.47)$$

สำหรับวงจรอินทิเกรตทางอุดมคติ จะมีรูปแบบการตอบสนองเป็น

$$I(s) = \pm \frac{\omega_n}{s} \quad (2.48)$$

เมื่อ ω_n เป็นค่าเวลากลางที่ของวงจรอินทิเกรต

อย่างไรก็ตาม ผลคูณของอัตราขยายและแบนด์วิดท์ (Gain - Bandwidth product) ที่มีค่าจำกัดของแอมป์เป็นเหตุให้เกิดการตอบสนองที่เปลี่ยนไป โดยสามารถแสดงเป็นผลคูณของการตอบสนองทางอุดมคติ และฟังก์ชันค่าผิดพลาด (Error function) ดังนี้

$$\frac{v_o}{v_i}(s) = \pm I(s)E(s) \quad (2.49)$$

ฟังก์ชันค่าผิดพลาด $E(s)$ นี้จะขึ้นอยู่กับความถี่ โดยเป็นค่าที่แสดงถึงภาวะการเบี่ยงเบนไปจากการตอบสนองทางอุดมคติของวงจรอินทิเกรต การเบี่ยงเบนนี้สามารถวิเคราะห์ในรูปแบบของค่าผิดพลาดของอัตราขยาย (Gain error, $\varepsilon(s)$) (การเบี่ยงเบนจากหนึ่ง) และค่าผิดพลาดของเฟส (Phase error, $\phi(s)$) (การเบี่ยงเบนจาก 90°) โดยที่

$$\begin{aligned} \varepsilon(s) &= |E(s)| - 1 \\ \phi(s) &= \angle E(s) \end{aligned} \quad (2.50)$$

ค่าฟังก์ชันถ่ายโอนของวงจรอินทิเกรตทั่วไป ดังรูปที่ สามารถหาได้จาก

$$\frac{v_o}{v_i} = - \frac{GB\omega_0}{s(s + \omega_0 + GB)} \quad (2.51)$$

ดังนั้น เมื่อทำการนอร์มัลไลซ์ฟังก์ชันค่าผิดพลาด (Normalized error function) แล้ว จะได้

$$E(s) = \frac{GB_n}{s_n + GB_n + 1} \quad (2.52)$$

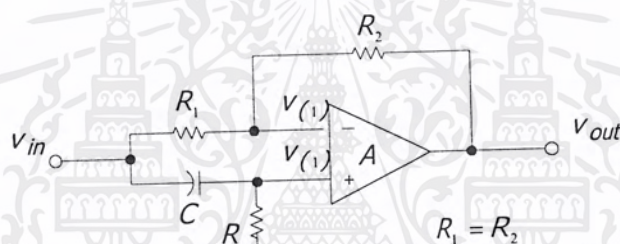
เมื่อ $s_n = s/\omega_0$ และ $GB_n = GB/\omega_0$ สมการ(2.52) แสดงให้เห็นว่าค่า Gain bandwidth product มีผลต่อฟังก์ชันค่าผิดพลาดของอัตราขยายและเฟสตามสมการ(2.50) ด้วย จะสังเกตได้อย่างชัดเจนว่าถ้ามีการใช้ออปแอมป์ที่มีค่า GB ต่ำในวงจรอินทิเกรต วงจรนี้จะมีค่าเบี่ยงเบนไปจากวงจรอินทิเกรตในทางอุดมคติมากนั่นเอง

ค่าตัวประกอบคุณภาพ(Quality factor, Q_I)ของวงจรอินทิเกรต สามารถหาได้จากสมการ (2.52) คือ

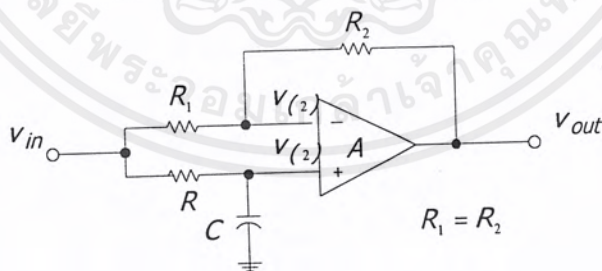
$$Q_I = -\frac{(\omega_0 + GB)}{\omega} \approx -\frac{GB}{\omega} = -|A(j\omega)| \quad (2.53)$$

จากสมการ (2.53) พบว่า ค่าตัวประกอบคุณภาพที่ความถี่ ω_0 มีค่าเท่ากับ GB_n โดย Q_I นี้เป็นค่าที่แสดงถึงความแม่นยำในการตอบสนองทางเฟสของวงจรอินทิเกรต ซึ่งในทางอุดมคติแล้ว ค่า Q_I นี้ควรมีค่าเป็นอนันต์ (Infinite) อย่างไรก็ตาม สมการ (2.53) เป็นตัวบ่งชี้ขีดจำกัดของค่า Q_I เนื่องมาจากผลคูณของ Gain-Bandwidth ที่มีค่าจำกัดของออปแอมป์

2.11 วงจรเลื่อนเฟส



รูปที่ 2.13 วงจรความถี่ผ่านตลอดที่เลื่อนเฟสจาก 0° ถึง -180°

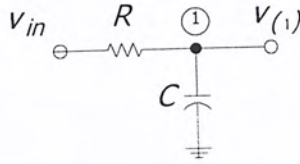


รูปที่ 2.14 วงจรความถี่ผ่านตลอดที่เลื่อนเฟสจาก 180° ถึง 0°

จากวงจรรูปที่ 2.13 และ 2.14 ประกอบด้วยความต้านทาน R_1, R_2 เป็นตัวกำหนดอัตราขยายของวงจรตัวต้านทาน R และตัวเก็บประจุ C จะเป็นตัวกำหนดเฟสที่ต้องการเลื่อน A คือโอเปอร์เรชันเนลแอมพลิไฟเออร์ ซึ่งสามารถนำมาวิเคราะห์ได้ดังต่อไปนี้

การวิเคราะห์วงจรความถี่ผ่านตลอดที่เลื่อนเฟสจาก 0° ถึง -180°

จากรูปที่ 2.13 สามารถนำมาเขียนแยกเป็นส่วนเพื่อให้พิจารณาง่าย



รูปที่ 2.15 วงจรที่แยกออกมาพิจารณาจากรูปที่ 2.13

ที่โนด 1 ใช้ทฤษฎีของเคอร์ชอฟ

$$V_1 = \frac{1}{RCs + 1} V_{in} \quad (2.54)$$

และส่วนที่เหลือของวงจรสามารถหาสมการได้เป็น

$$\frac{V_{out} - V_{(1)}}{R_2} = \frac{V_{(1)} - V_{in}}{R_1} \quad (2.55)$$

กำหนดให้ และนำสมการที่ (2.54) แทนค่าลงในสมการ (2.55) จะได้ว่า

$$\begin{aligned} \frac{(RCs + 1)V_{out} - V_{in}}{(RCs + 1)} &= \frac{V_{in} - (RCs + 1)V_{in}}{(RCs + 1)} \\ (1 + RCs)V_{out} &= (1 - RCs)V_{in} \end{aligned}$$

ทำการแก้สมการ จะสามารถหาฟังก์ชันถ่ายโอน (Transfer function) ได้เป็น

$$\frac{V_{out}}{V_{in}} = \frac{1 - RCs}{1 + RCs} \quad (2.56)$$

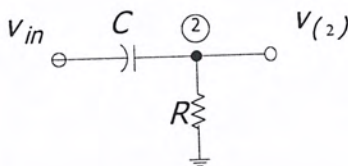
จากสมการที่ (2.56) กำหนดให้ จะสามารถหาสมการเลื่อนเฟสได้เป็น

$$\theta = -2 \tan^{-1}(\omega T) \quad (2.57)$$

โดยที่ เฟสที่เลื่อน เมื่อพิจารณาจากสมการที่ (2.57) จะพบว่าสามารถเลื่อนเฟสจาก 0° ถึง -180° และอัตราขยายของวงจรจะมีค่าเป็นหนึ่งตลอดย่านความถี่ ตามสมการที่ (2.56)

การวิเคราะห์ห้วงจรความถี่ผ่านตลอดที่เลื่อนเฟสจาก 180° ถึง 0°

จากรูปที่ 2.14 สามารถนำมาเขียนแยกส่วนเพื่อให้พิจารณาง่ายขึ้น



รูปที่ 2.16 วงจรที่แยกออกมาพิจารณาจากรูปที่ 2.14

ที่โนด 2 ใช้ทฤษฎีของเคอร์ชอฟ

$$V_{(2)} = \frac{RCs}{RCs + 1} V_{in} \quad (2.58)$$

และส่วนที่เหลือของสมการ สามารถหาสมการได้เป็น

$$\frac{V_{out} - V_{(2)}}{R_2} = \frac{V_{(2)} - V_{in}}{R_1} \quad (2.59)$$

กำหนดให้ และนำสมการที่ (2.58) แทนค่าลงในสมการที่ (2.59) ได้เป็น

$$\begin{aligned} V_{out} - \left(\frac{RCs}{RCs + 1} \right) V_{in} &= \left(\frac{RCs}{RCs + 1} \right) V_{in} - V_{in} \\ (1 + RCs) V_{out} &= (RCs - 1) V_{in} \end{aligned}$$

ทำการแก้สมการ จะสามารถหาฟังก์ชันถ่ายโอนได้เป็น

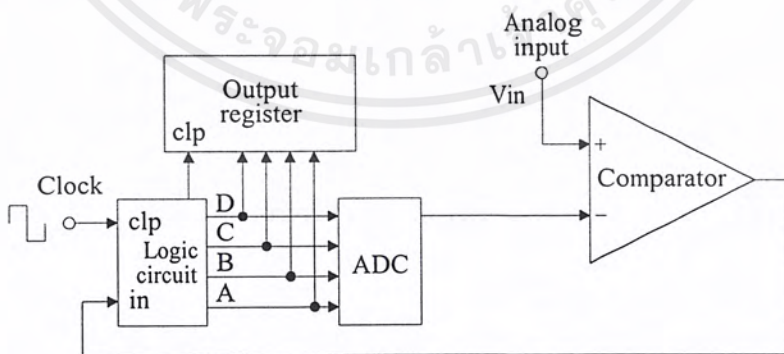
$$\frac{V_{out}}{V_{in}} = \frac{RCs - 1}{RCs + 1} \quad (2.60)$$

สมการที่ (2.59) กำหนดให้ $T = RC, S = j\omega$ จะสามารถหาสมการเคลื่อนได้เป็น

$$\theta = \pi - 2 \tan^{-1}(\omega T) \quad (2.61)$$

2.12 Successive Approximation ADC

ADC ชนิดนี้เป็น ADC ที่นิยมใช้กันอย่างแพร่หลายเพราะมีความละเอียด, ความเร็ว, ความถูกต้องและราคาต่ำ ซึ่งสัญญาณดิจิทัลได้มาจากการประมาณค่าจากแรงดันทางด้านอินพุตซึ่งในการแปลงจะใช้เวลาน้อยกว่า ADC ชนิด Counter และค่าที่ได้จะเป็นอิสระจากค่าแรงอินพุต อย่างไรก็ตาม ADC ชนิดนี้จะประกอบไปด้วยวงจรควบคุมที่ซับซ้อนกว่า ADC ชนิดอื่น แสดงบล็อกไดอะแกรมของ Successive Approximation ADC ชนิด 4 บิต ดังรูปที่ 2.17



รูปที่ 2.17 บล็อกไดอะแกรมของ Successive Approximation ADC

ADC ชนิดนี้จะใช้รีจิสเตอร์แทนเคาน์เตอร์ เพื่อเตรียมสัญญาณอินพุตสำหรับ ADC ซึ่งค่าไบนารีของรีจิสเตอร์ ทำได้ดังนี้

1. จำนวนไบนารีถูกป้อนเข้าไปยังรีจิสเตอร์ โดยเซตให้ MSB เป็น 1 และบิตอื่นเป็น 0 ค่าเอาต์พุตของ ADC จะเท่ากับน้ำหนักของ MSB ถ้าค่าเอาต์พุตมากกว่า V_{in} ที่คอมพาราเตอร์ จะได้ค่า Low ผลที่ได้จะทำให้ MSB ถูกรีเซตให้เป็น 0 แต่ถ้าค่านี้น้อยกว่า V_{in} MSB จะยังคงเป็น 1

2. บิตที่มีนัยสำคัญตัวถัดไปถูกเซตเป็น 1 และเอาต์พุตของคอมพาราเตอร์จะเหมือนกับข้อแรกถ้าค่าที่ได้ใหม่มากกว่า V_{in} ค่าที่มีนัยสำคัญนี้จะถูกรีเซตเป็น 0 และอีกอันหนึ่งก็คือถ้าค่าใหม่ที่ได้มีน้อยกว่า V_{in} บิตนี้จะยังคงเป็น 1

3. ลำดับความต่อเนื่องนี้ใช้กับทุกบิตในจำนวนของไบนารีในรีจิสเตอร์ โดยขั้นตอนดังกล่าวใช้สัญญาณนาฬิกา 1 ลูกต่อบิต

4. หลังจากทุกบิตถูกทดสอบแล้วเอาต์พุตของวงจรจะจัดไบนารี 1 ไปปรากฏที่รีจิสเตอร์ เพื่อนำไปเป็นสัญญาณเอาต์พุตดิจิทัล

การเปรียบเทียบค่าเวลาสูงสุดในการแปลงสัญญาณของ eight-bit digital-ramp ADC ไปเป็น eight-bit successive-approximation ADC ทั้งสองชนิดจะใช้สัญญาณนาฬิกา 1 MHz เราสามารถแสดงค่าความแตกต่างของเวลาสูงสุดที่ใช้ในการแปลงสัญญาณ digital-ramp ADC ได้ดังนี้

$$2^N - 1 \times 1\mu s = 2^8 - 1 \times 1s = 255\mu s \quad (2.62)$$

เมื่อ N คือจำนวนของบิต

Successive Approximation ADC ค่าความเร็วสูงสุดในการแปลงสัญญาณของจำนวน bit-time เราสามารถคำนวณคาบเวลาของสัญญาณนาฬิกาได้ดังนี้

$$8 \times 1\mu s = 8\mu s$$

เห็นได้ชัดแล้วว่า Successive Approximation ADC เหมาะที่จะใช้งานเป็น High speed

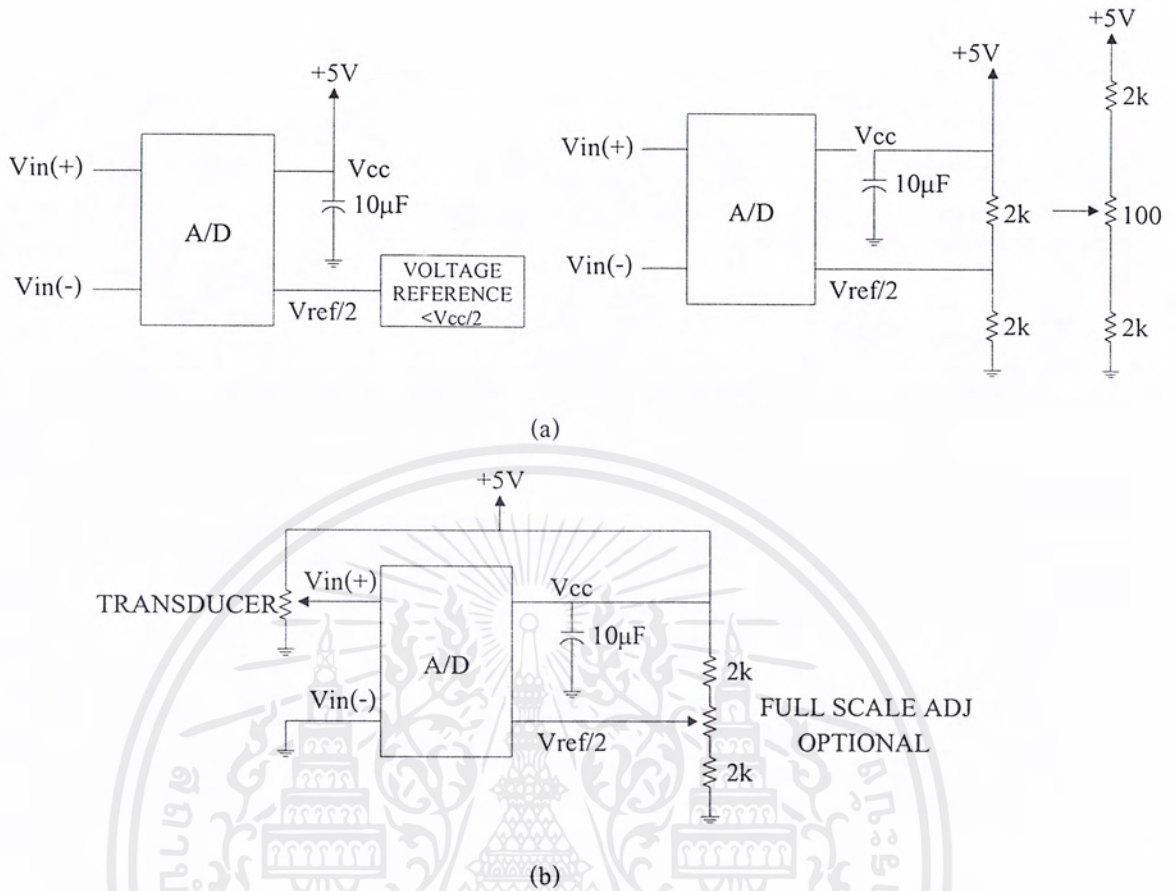
2.12.1 หน้าที่ของ ADC

หลังจากทุกบิตถูกทดสอบและกำหนดรหัสเป็นไบนารี 8 บิต ที่มีค่าตรงกับแรงดันอินพุตแล้วจะถูกส่งไปยังเอาต์พุต ที่อินพุตของ WR ก็จะเริ่มแปลงสัญญาณของขบวนการพัลส์ที่เข้ามาถ้า CS มีสถานะ Low เมื่อมีการเปลี่ยนสัญญาณที่ WR หรือ CS Successive Approximation Register (SAR) ก็จะถูกระตุ้นให้เลื่อนข้อมูลเข้าไปโหมดการทำงาน ADDC มีการทำงานอยู่ 2 โหมด ดังนี้

1. แอปโซลูตโหมด (Absolute Mode)
2. เรโซเมตริกโหมด (Ratiometric Mode)

แอปโซลูตโหมด แสดงในรูปที่ 2.18(ก) ทั้งค่าความถูกต้องและค่าคงที่ของอุณหภูมิทั้งสองอย่างนี้เป็นองค์ประกอบสำคัญในการแปลงสัญญาณให้ถูกต้อง

เรโซเมตริกโหมด แสดงในรูปที่ 2.18(ข) แรงดันอ้างอิงจะเป็นองค์ประกอบของซอสทรานสดิวเซอร์ (Source transducer) เอาต์พุตของ ADC ดังนั้น $V_{ref}/2$ จะถูกตัดออกที่รหัสดิจิทัลตัวสุดท้าย ค่าความผิดพลาดเนื่องจากปรับแรงดันอ้างอิงไม่ถูกต้องจะมีผลต่อการแปลงสัญญาณ

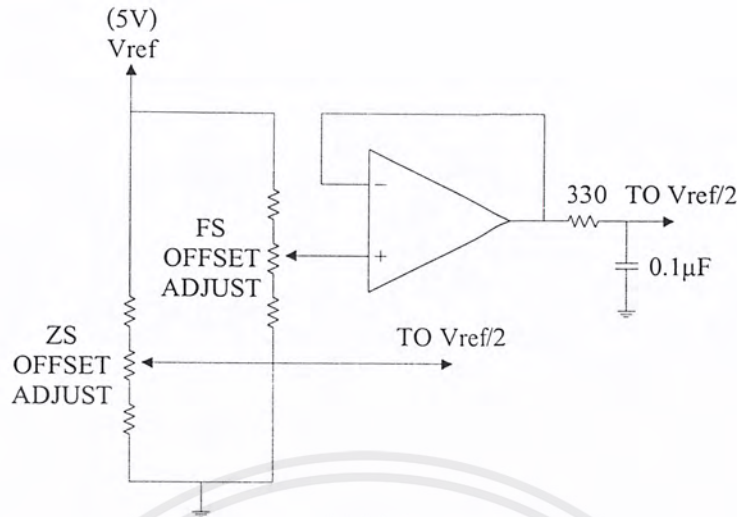


รูปที่ 2.18 (a) แสดงแรงดันอ้างอิงของแอปโซลูตโหมด

(b) แสดงแรงดันอ้างอิงของเรโซเมตริกโหมด

2.12.2 ความผิดพลาดและการปรับระยะช่วงกว้างของอินพุต

มีหลายอย่างที่ทำให้ข้อมูลจากการแปลงสัญญาณเกิดผิดพลาด ซึ่งบางอย่างก็สามารถกำจัดได้แม้ว่าความผิดพลาดที่เกิดโดยธรรมชาติเช่น รีเลทีฟแอคคูเรซี (relative accuracy) ส่วนนี้ไม่สามารถกำจัดได้ แต่ความผิดพลาดที่เป็นฟูลสเกล (Full Scale) และซีโรสเกลออฟเซตเออเรอร์ (Zero Scale Offset Error) สามารถกำจัดได้ง่ายดังรูปที่ 2.19

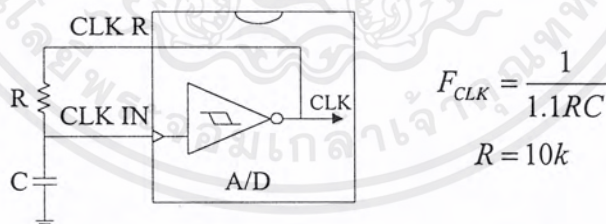


รูปที่ 2.19 วงจรการปรับซีโรสเกล

ซีโรสเกลเออเรอร์ ของADC เป็นค่าความแตกต่างของศักย์ไฟฟ้าระหว่างค่าอุดมคติ $\frac{1}{2}$ LSB (9.8mV สำหรับ $V_{ref}/2$ เท่ากับ 2.5V) แรงดันอินพุตที่ป้อนเข้ามามีผลให้การเปลี่ยนรหัสที่เอาต์พุตจาก 0000 0000 เป็น 0000 0001 ถ้าค่าต่ำสุดทางค่านินพุตไม่ได้เป็นศักย์ไฟฟ้ากราวด์จะเกิดซีโรสเกลขึ้น

2.12.3 การเลือกสัญญาณนาฬิกา

สัญญาณนาฬิกาสำหรับ ADC นั้นได้มาจากสัญญาณนาฬิกาภายนอกหรือ สัญญาณนาฬิกาภายในตัวมันเอง สัญญาณนาฬิกาภายนอกสามารถทำได้โดยต่อความต้านทานและตัวเก็บประจุ ดังรูปที่ 2.20 การเลือกตัวเก็บประจุควรหลีกเลี่ยงค่ามากๆ ในที่นี้ควรมีค่าน้อยกว่า 50 pF



รูปที่ 2.20 วงจรกำเนิดสัญญาณนาฬิกาภายใน ADC

2.12.4 การเริ่มต้นใหม่ในระหว่างการแปลงสัญญาณ

ในขบวนการแปลงสัญญาณสามารถหยุดขบวนการนั้นได้และเริ่มใหม่โดยให้ CS และ WR เป็น Low ข้อมูลเอาต์พุตก็จะไม่ถูกแลตช์ (Latch) ถ้าขบวนการแปลงสัญญาณไม่ครบขั้นตอนข้อมูลเดิมก่อนหน้านั้นจะยังคงอยู่จนกระทั่งขั้นตอนการแปลงสัญญาณใหม่จะสมบูรณ์

2.12.5 การแปลงสัญญาณต่อเนื่อง

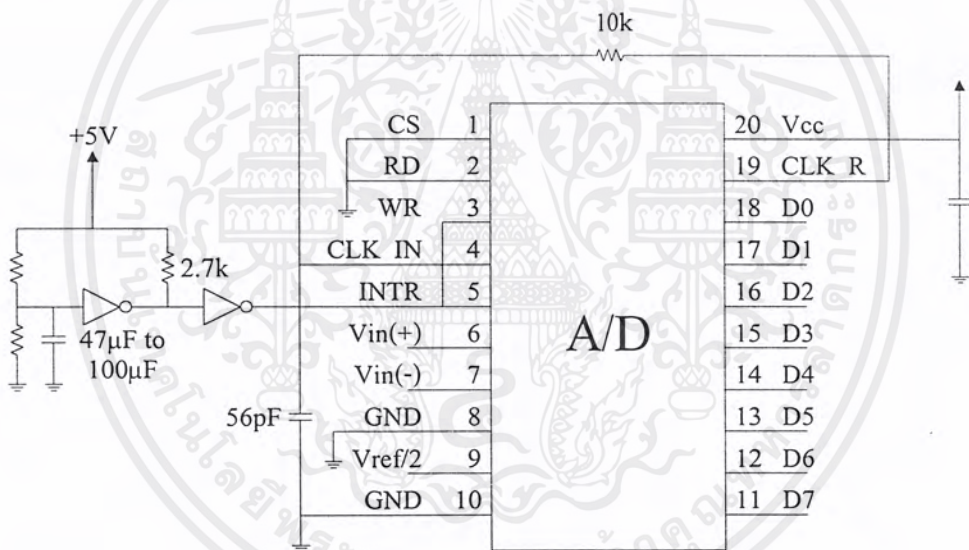
ใช้เพื่อเปลี่ยนสัญญาณอินพุต การแปลงสัญญาณต่อเนื่องโดยให้ขา CS และ RD ต่อลงกราวด์ และขา INTR ต่อกับ WR การต่อควรจะเป็นระดับลอจิกต่ำชั่วขณะก่อนที่จะจ่ายไฟเข้าวงจรดังรูปที่ 2.21

2.12.6 แหล่งจ่ายไฟ

น๊อยส์สไปค์ (Noise spike) ของแรงดัน V_{cc} สามารถทำให้การแปลงสัญญาณเกิดผิดพลาดได้ ในที่นี่จะใช้ตัวเก็บประจุ $1\ \mu\text{F}$ หรือมากกว่าแล้วแต่สมควร ต่อที่ขั้ว V_{cc} การแยกจ่ายไฟระหว่างวงจรภายนอกกับวงจร ADC จะช่วยลดปัญหานี้ได้

2.12.7 การเชื่อมต่อสายและข้อควรระวังในการต่อ

การใช้ไวร์แรปซ็อกเก็ต (Wire Wrap Socket) และการเชื่อมโยงสายไม่ควรทำสำหรับ ADC ควรใช้ซ็อกเก็ต (Socket) ใส่ลงบนแผ่นปริ้นท์ (PCBs) สายสัญญาณและสายนำสัญญาณควรจัดให้อยู่ในกลุ่มเดียวกัน เพราะว่าอาจจะทำให้เกิดฮัม (Hum) และน๊อยส์ (Noise) ได้ อนาล็อกกราวด์ (Analog Ground) และ ดิจิตอลกราวด์ (Digital Ground) ควรแยกจากกัน ตัวเก็บประจุที่ใช้บายพาส V_{cc} และตัวเก็บประจุที่ใช้เป็น Self-Clocking ควรต่อที่ดิจิตอลกราวด์ ตัวเก็บประจุที่ต่อบายพาส $V_{ref}/2$ และตัวเก็บประจุที่กรองสัญญาณอินพุต ควรต่อที่อนาล็อกกราวด์



รูปที่ 2.21 วงจรการแปลงสัญญาณอย่างต่อเนื่อง

2.13 การใช้งานของพอร์ตอนุกรม

การสื่อสารข้อมูลอนุกรมเป็นการรับหรือส่งข้อมูลในลักษณะกลุ่มของบิตคราวละหนึ่งบิตเรียงลำดับเรื่อยไปจนถึงสิ้นสุดการสื่อสารแบบนี้จะมีข้อแตกต่างจากการสื่อสารแบบขนานเป็นอย่างมากเนื่องจากข้อมูลมีการโอนย้ายมาพร้อมกันจึงมีความจำเป็นต้องใช้จำนวนเส้นสัญญาณมากขึ้นตามจำนวนบิตของข้อมูลด้วยในขณะที่การสื่อสารแบบอนุกรมนั้นต้องการเส้นสัญญาณเพียงสองหรือสามเส้นเท่านั้น ดังนั้น การสื่อสารแบบขนานจึงไม่เหมาะสมในการสื่อสารกับอุปกรณ์ภายนอกเป็นระยะทางไกลๆ เพราะจะทำให้สิ้นเปลืองค่าใช้จ่ายมาก

2.13.1 ความเร็วของการสื่อสารข้อมูลอนุกรม

เนื่องจากการสื่อสารแบบอนุกรมเป็นการรับ/ส่ง ข้อมูล ในลักษณะกลุ่มของบิตข้อมูล (Bit Stream) ดังนั้นจึงต้องให้ความสนใจในการพิจารณาถึงเรื่องอัตราความเร็วในการรับ/ส่งบิตเหล่านี้ เป็นลำดับแรก โดยทั่วไปมักได้แก่ 110,150,300,1200,2400,4800,9600,19200 บอด ข้อมูลทั้ง 8 บิตนี้หากว่าถูกส่งออกมาด้วยอัตรา 2400 บอด จะใช้เวลาในการส่งข้อมูลหนึ่งบิตมีค่าเท่ากับ $1/2400$ หรือ $460 \mu\text{S}$ และใช้เวลาในการส่งข้อมูลทั้ง 8 บิตมีค่าเท่ากับ (8×416) หรือ $3328 \mu\text{S}$

2.13.2 รูปแบบของการส่งข้อมูลอนุกรม

การสื่อสารอนุกรมแบบ อะซิงโครนัส จะใช้การแปลงข้อมูลขนานให้เป็นอนุกรมแล้วเพิ่มเติมบิตบางอย่างรวมไปกับการส่งข้อมูลจริงได้แก่

1. บิตเริ่มต้น (Start Bit)

บิตเริ่มต้นมีหน้าที่สำหรับการบ่งบอกให้ทราบถึงตำแหน่งจุดเริ่มต้นก่อนบิตข้อมูล ตามปกติแล้วค่าของบิตเริ่มต้นจะเป็นระดับลอจิกต่ำ

2. บิตแสดงสถานะความเป็นเลขคู่หรือเลขคี่

บิตนี้มีหน้าที่เพื่อการตรวจสอบความถูกต้องของข้อมูลโดยทั่วไปมักจะเรียกว่า บิตพาริตีและจะนำไปต่อท้ายข้อมูล ค่าของบิตนี้ขึ้นอยู่กับจำนวนค่าของบิตที่เป็นหนึ่ง ซึ่งจะเป็นได้สองลักษณะคือ พาริตีคู่ (Even Parity) หรือพาริตีคี่ (Odd Parity) ตัวอย่างเช่นระบบที่ติดต่อกับโดยระบุว่าจะใช้พาริตีคี่ ทางด้านส่งจะนำค่าข้อมูลที่จะส่งมาพิจารณาหา จำนวนของบิตที่มีค่าเป็น 1 เป็นเลขจำนวนคี่ ค่าของพาริตีคี่จะมีค่าเป็น 1 การพิจารณาทางด้านรับเป็นการตรวจสอบจำนวนบิตที่มีค่าเป็น 1 ของข้อมูลที่ได้รับมาทั้งหมดรวมทั้งบิตพาริตี ถ้ามีค่าเป็นเลขจำนวนคู่แสดงว่า เกิดการผิดพลาดของข้อมูลขึ้นเป็นต้น

3. บิตสุดท้าย (Stop Bit)

บิตสุดท้ายเป็นบิตที่เพิ่มขึ้นเพื่อระบุถึงขอบเขตการสิ้นสุดของกลุ่มบิตข้อมูล บิตสุดท้ายนี้สามารถโปรแกรมได้ คือ 1 บิต 1 บิต และ 2 บิต ดังนั้นกรณีของการส่งข้อมูล 8 บิตหากข้อมูลถูกส่งออกไปด้วยอัตรา 2400 บอดเวลาโดยรวมในการส่งข้อมูล 1 ไบต์ จะมีค่าเป็น $(112 \times 416) \mu\text{S}$ หรือ 4.99 ms

2.13.3 การส่งข้อมูลอนุกรมของ 8051

พอร์ตอนุกรมของ 8051 มีโครงสร้างการทำงานในแบบที่เรียกว่า ฟูลดูเพล็กซ์ (Full Duplex) ในการรับและส่งข้อมูลได้ในเวลาเดียวกัน โดยทางวงจรของตัวส่ง (Transmitter) ประกอบไปด้วยข้อมูลออกไปยังพอร์ตอนุกรม ทางขาสัญญาณ TxD (พอร์ต 3.1) ส่วนวงจรด้านตัวรับ(Receiver) ประกอบด้วย SBUF เช่นเดียวกันสัญญาณข้อมูลอนุกรม ที่รับเข้ามาทางขาสัญญาณ RxD (พอร์ต 3.0)

พอร์ตอนุกรมของ 8051 สามารถโปรแกรมการทำงานได้หลายโหมดด้วยกันโดยเลือกที่บิต SM0 และ SM1 ซึ่งอยู่ในรีจิสเตอร์ควบคุม SCON การทำงานทั้ง 4 โหมดของพอร์ตอนุกรมมีดังนี้

โหมด 0 : ใช้รับส่งข้อมูล 8 บิต โดยการส่งจะเลื่อนออกทีละบิต โดยส่งบิต D0 ออกไปก่อนทางขา

RxD และไม่มีการส่ง start bit แต่จะส่ง shift clock ทางขา TxD ความเร็วทำของ CPU CLOCK

โหมด 1 : ใช้สำหรับการเชื่อมต่ออนุกรมแบบ UART (Universal Asynchronous Receiver/Transmitter) โดยส่ง

- แบบ 10 บิต ข้อมูล 8 บิต 1 start bit และ 1 stop bit และสามารถเปลี่ยนแปลงอัตราความเร็วในการส่งข้อมูลได้ โดยขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โพล์ของ Timer1
- โหมด 2 : ใช้สำหรับการเชื่อมต่ออนุกรมแบบ UART โดยการใช้กลุ่มข้อมูลแบบ 11 บิต และกำหนดอัตราความเร็วในการส่งข้อมูลเท่ากับ 1/32 และ 1/64 ของ CPU CLOCK โดยโปรแกรมที่บิต SMOD ใน PCON
- โหมด 3 : ใช้สำหรับการเชื่อมต่ออนุกรมแบบ UART โดยการใช้กลุ่มข้อมูลแบบ 11 บิต และสามารถเปลี่ยนแปลงอัตราความเร็วในการส่งข้อมูลได้ โดยควบคุมที่บิต SMOD และอัตราโอเวอร์โพล์ของ Timer1 นอกจากนี้โหมด 2 และ 3 ยังมีการดำเนินการอีกแบบหนึ่ง โดยสามารถนำมาใช้ประโยชน์ในการสื่อสารข้อมูลแบบที่มีไมโครโปรเซสเซอร์หลายตัวทำงานร่วมกันได้ซึ่งมีชื่อเรียกว่า Multi Processor Mode

ตารางที่ 2.1 แสดงการเลือกโหมดการทำงานของ SCON

SM0,SM1 บิตเลือกโหมดการทำงาน

SM0	SM1	โหมด	การทำงาน
0	0	0	ทำงานเป็น shift register อัตราเร็วในการรับหรือส่งข้อมูลเท่ากับ 1/12 ของความถี่ออสซิลเลเตอร์
0	1	1	8 bit UART อัตราเร็วในการรับหรือส่งข้อมูลกำหนดเองได้
1	0	2	9 bit UART อัตราเร็วในการรับหรือส่งข้อมูล = 1/32 หรือ 1/64 ของความถี่ออสซิลเลเตอร์ ขึ้นกับบิต SMOD ใน PCON
1	1	3	9 bit UART อัตราเร็วในการรับส่งข้อมูลกำหนดเองได้

SM2 บิตเลือกการทำงานแบบ

- 1 : เลือก Multiprocessor ใช้ได้กับ โหมด 2,3
- 0 : เลือก Single Processor ใช้ได้กับทุกโหมด
- (เมื่อเลือกการทำงานรับข้อมูลแบบ Multiprocessor Mode แล้ว เมื่อข้อมูลบิตที่รับได้มีค่าเป็น 1 RI จะเซ็ททันที)

REN บิตควบคุมให้รับหรือไม่รับข้อมูล

- 1 : ให้รับข้อมูลได้
- 0 : ห้ามรับข้อมูล

TB8 (Transmit bit D8) ข้อมูลบิตที่ 9 ที่จะส่งออกไปในโหมด 2,3 ให้ใส่ในบิต TB8

RB8 (Receiver bit D8) ข้อมูลบิตที่ 9 ที่จะรับเข้ามาเก็บในบิตนี้ ข้อมูลบิตที่ 9 ก็คือค่าใน TB8 ทางด้านส่งนั่นเอง (ข้อมูลบิตที่ 9 คือค่าใน TB8 หรือ RB8 นั่นเอง)

TI บิต TI จะเป็น 1 เมื่อสิ้นสุดการส่งข้อมูล 1 ไบต์

RI บิต RI จะเป็น 1 เมื่อรับข้อมูลเสร็จ 1 ไบต์ (บิต RI, TI ผู้เขียนโปรแกรมจะต้องเคลียร์เอง)

2.13.4 การอินเตอร์รัปต์ของพอร์ตสื่อสารอนุกรม

เนื่องจากการส่งหรือรับ ข้อมูลอนุกรมจะส่งทีละไบต์ 8051 จึงได้กำหนดให้บิตหรือแฟล็กสถานะที่จัดรวมอยู่ในรีจิสเตอร์ SCON เช่น แฟล็ก TI ซึ่งจะมีค่าเป็น 1 เมื่อข้อมูลได้ทำการส่งออกไปภายนอกเสร็จสิ้นแล้ว และแฟล็ก RI ซึ่งจะมีค่าเป็น 1 เพื่อให้รู้ว่าได้รับข้อมูลผ่านเข้ามาพอร์ตอนุกรม เสร็จแล้วเมื่อแฟล็ก RI, TI นี้มีค่าเป็น 1 จะมีผลทำให้เกิดการอินเตอร์รัปต์ขึ้น ดังนั้นภายในโปรแกรมรับหรือส่งข้อมูลจะต้องทำการตรวจสอบจากสถานะของแฟล็กเหล่านั้นเองว่าเป็นการส่งหรือรับข้อมูล

2.13.5 กระบวนการรับและการส่งข้อมูลอนุกรมของ 8051

การส่งข้อมูลออกทางพอร์ตอนุกรมของ 8051 จะเริ่มต้นขึ้นภายหลังเมื่อมีการเขียนข้อมูลลงใน SBUF ข้อมูลนี้จะถูกเลื่อนทีละบิต และส่งสัญญาณออกไปภายนอกโดยอัตโนมัติเมื่อข้อมูลเหล่านี้ได้ส่งออกครบถ้วนแล้วจะทำให้ค่าของแฟล็ก TI ให้เป็น 1 เพื่อแจ้งให้ทราบว่าขณะนี้ SBUF ว่างและพร้อมที่จะส่งข้อมูลไบต์ต่อไปแล้วในกรณีที่ผู้ใช้เขียนข้อมูลใหม่ลงในรีจิสเตอร์ SBUF โดยไม่รอให้แฟล็ก TI มีค่าเป็น 1 ก่อนจะมีผลทำให้ข้อมูลที่ส่งไปผิดพลาดได้

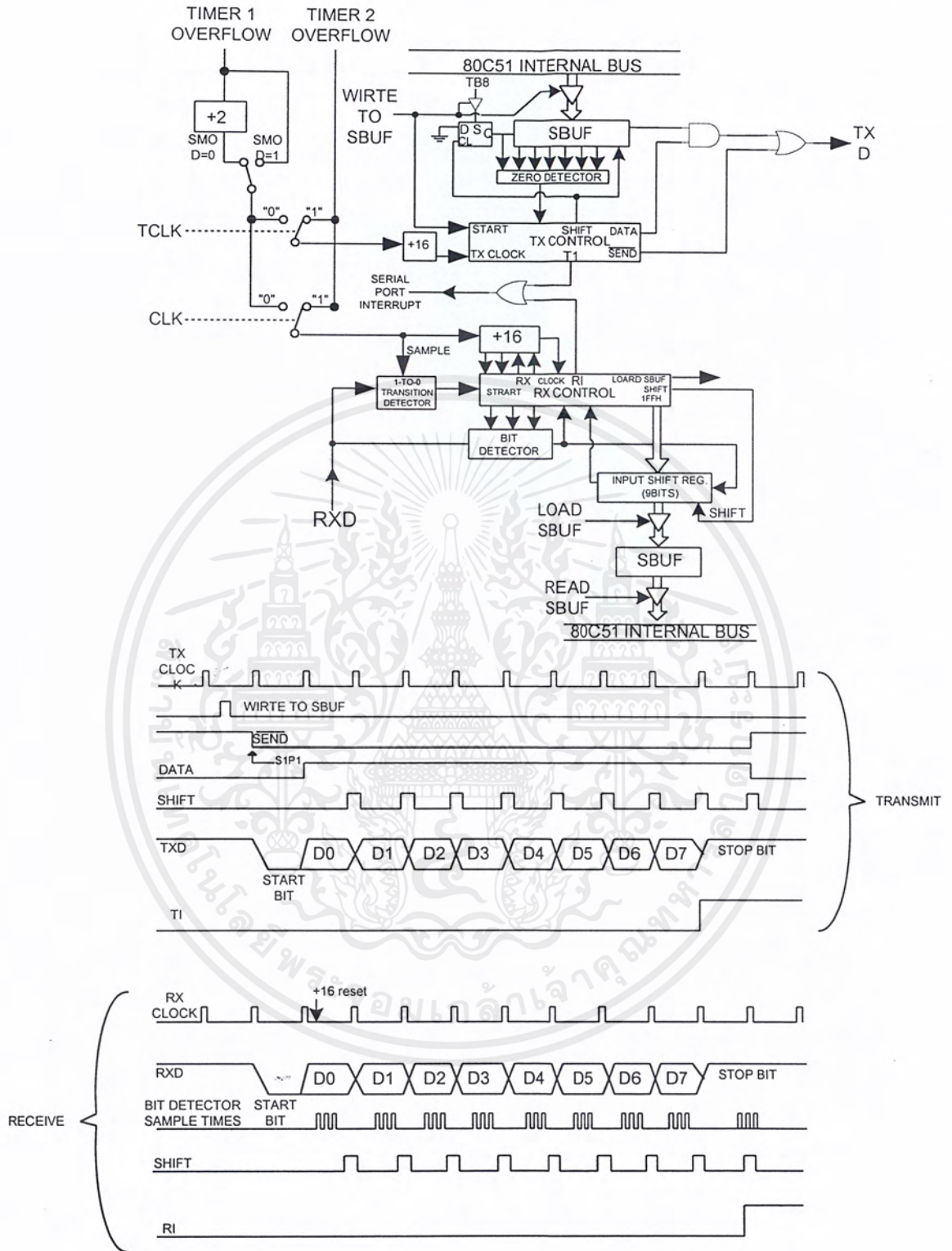
สำหรับการรับข้อมูลจากพอร์ตอนุกรมจะต้องเริ่มต้นโดยการกำหนดเซตค่าดังนี้ REN(Receiver Enable) ให้มีค่าเป็น 1 ก่อน หลังจากนั้นเมื่อมีการส่งข้อมูลภายนอกถูกส่งเข้ามายัง 8051 ทีละบิตจนครบ และเมื่อบิต SBUF และแฟล็ก RI ก็จะมีค่าเป็น 1 (ถูกเซต) หลังจากนั้นก็จะเกิดการอินเตอร์รัปต์ขึ้น

2.13.6 พอร์ตอนุกรม (โหมด 0)

การทำงานของพอร์ตอนุกรม (โหมด 0) เป็นการรับและส่งข้อมูลอนุกรมจำนวน 8 บิตโดยใช้เพียงขาสัญญาณ RxD เท่านั้น (ขานี้ใช้งาน 2 หน้าที่ใช้ส่งและรับข้อมูล) ส่วนขา TxD จะนำไปใช้เพื่อเป็นขาสัญญาณนาฬิกาในการให้จังหวะ การเลื่อนข้อมูลกับวงจร เลื่อนบิต ภายนอกสำหรับอัตราความเร็ว จะถูกกำหนดไว้คงที่ที่ค่า 1/12 ของความถี่ออสซิลเลเตอร์ ในโหมด 0 เมื่อมีการรับและส่งข้อมูล 1 ไบต์ โดยสัญญาณนาฬิกาในการเลื่อนบิตนี้จะเกิดขึ้นภายในตัวของ 8051 เอง เนื่องจากโหมดนี้ไม่มีการส่ง start bit และ stop bit ดังนั้นจึงจำเป็นต้องส่งสัญญาณ shift clock ออกไป เพื่อใช้ synchronise ระหว่างฝ่ายรับและฝ่ายส่งโดยจะใช้ขา TxD ส่วนการรับข้อมูลจะรับข้อมูลเข้าทางขา RxD และรับ shift clock เข้าทางขา TxD ถ้าความถี่ออสซิลเลเตอร์มีค่าเท่ากับ 12MHz ก็จะสามารถส่งได้ถึง 1 ล้านบิตต่อวินาที (โหมด 0) เป็นโหมดที่ส่งข้อมูลได้เร็วที่สุด

2.13.7 การรับส่งอนุกรม (โหมด 1) ส่งข้อมูล 8 บิต, 1 start bit, 1 stop bit

การทำงานในโหมด 1 เป็นการสื่อสารข้อมูลอนุกรมจำนวน 10 บิต ประกอบไปด้วยบิตเริ่มต้น 1 บิต ข้อมูลจำนวน 8 บิต และบิตสุดท้าย 1 บิต ดังแสดงในรูปที่ 2.18 โดยข้อมูลจะถูกส่งออกทาง TxD และรับเข้าทางขาสัญญาณ RxD ในส่วนของข้อมูล 8 บิต ที่ได้รับหรือทำการส่งออก จะเป็นรีจิสเตอร์ SCON สำหรับอัตราเร็วในการส่งข้อมูลของ(โหมด 1) นั้น สามารถกำหนดเลือกได้จาก Timer 1 ฝั่งเวลาและการทำงานดังแสดงในรูปที่ 2.22



รูปที่ 2.22 ฟังก์การทำงานโหมด 1

ดังได้กล่าวแล้วว่าการส่งข้อมูลอนุกรม (โหมด 1) สามารถเปลี่ยนแปลงความเร็วได้โดยใช้ Timer 1 ทำหน้าที่เป็นตัวกำเนิดอัตราการส่งข้อมูล และใช้แฟล็กที่เกิดขึ้นจากการโอเวอร์โฟลว์ ของ Timer 1 โดยโปรแกรม Timer 1 ทำงานใน(โหมด 2) 8 bit automatic reload

ความถี่อัตราบอดขึ้นกับ
$$= \frac{(2)^{SMOD}}{(32)} \times \text{อัตราโอเวอร์โฟลว์ของ Timer 1} \tag{2.63}$$

หรือความถี่อัตราบอด
$$= \left| \frac{(2)^{SMOD}}{(32)} \right| \times \left| \frac{Fosc}{12 \times (256 - TH1)} \right| \tag{2.64}$$

โดย SMOD เป็นค่าของบิตในรีจิสเตอร์ PCON (มีค่าเป็น 0 หรือ 1)
ค่าภายในรีจิสเตอร์ TH1 ซึ่งใช้เป็นค่าสำหรับโหลดซ้ำ

ตารางที่ 2.2 แสดงการเลือกค่า TH1 สำหรับ Baud rate ต่างๆ

Baud Rate	Fosc	SMOD	TIMER 1		
			C/T	MODE	Reload Value
(MODE 0) Max : 1 MHz	12 MHz	X	X	X	X
(MODE 2) Max : 375 KHz	12 MHz	1	X	X	X
(MODE 2) Min : 187.5 KHz	12 MHz	0	X	X	X
MODE 1, 3 : 62.5 K	12 MHz	1	0	2	FFH
19.2 K	11.059 MHz	1	0	2	FDH
9.6 K	11.059 MHz	0	0	2	FDH
4.8 K	11.059 MHz	0	0	2	FAH
2.4 K	11.059 MHz	0	0	2	F4H
1.2 K	11.059 MHz	0	0	2	E8H
137.5	11.059 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEDH

2.13.8 การรับ/ส่งข้อมูลอนุกรม (โหมด2)

โหมดนี้ใช้ทั้งหมด 11 บิต โดยแบ่งเป็น start bit , 9 data bit และ stop bit ผู้ใช้สามารถกำหนดค่าเองได้ว่าจะส่งค่าอะไรออกไป โดยจะต้องนำไปใส่ไว้ในบิต TB8 ในรีจิสเตอร์ SCON ส่วนมากผู้ใช้นักจะนำบิตนี้มาใช้เป็น Parity bit โดยโหลดค่ามาจาก Parity flag ใน PSW ส่วนทางด้านรับบิตที่ 9 จะถูกนำมาเก็บไว้ในบิตรับข้อมูลขึ้นกับความถี่ออสซิลเลเตอร์ของ CPU และค่า SMOD ซึ่งอยู่ในบิต 7 ใน PCON

$$\text{MODE 2 BAUD RATES} = \frac{(2^{SMOD})fosc}{64}$$

(2.65)

ถ้า CPU ที่ 12 MHz และ SMOD มีค่า 0 และ 1

$$\text{เมื่อ SMOD} = 0 \text{ จะได้} = \frac{2^1(12)(10^6)}{64} = 187,500 \text{ บอด}$$

$$\text{เมื่อ SMOD} = 1 \text{ จะได้} = \frac{2^1(12)(10^6)}{64} = 375,500 \text{ บอด}$$

2.13.9 การรับส่งข้อมูลอนุกรมโหมด 3

เหมือนกับ (โหมด 2) ทุกอย่างยกเว้นความเร็วในการรับส่งข้อมูลจะขึ้นอยู่กับอัตราโอเวอร์โพล์ของ Timer1 หรือ Timer2 โดยมีฟังก์การทำงาน และการเปลี่ยนแปลงความเร็วได้จากตารางที่ 2.1 หรือคำนวณสูตรดังนี้เมื่อกำหนดให้ Timer 1 ทำงานโหมด 2 Auto Reload

$$\text{MODE 1,3} = \frac{(2^{SMOD}(f_{ocs}))}{(32)12[256 - TH1]} \text{ บอด} \quad (2.66)$$

$$\text{หรือ} = \frac{K(OSCfreq)}{(32)12[256 - TH1]} \text{ บอด} \quad (2.67)$$

ถ้า SMOD = 0 จะได้ K = 1

SMOD = 1 จะได้ K = 2

เมื่อต้องการหา TH1 จะได้

$$TH1 = 256 - \frac{K(OSCfreq)}{384(Baudrate)} \quad (2.68)$$

เมื่อ SMOD = 0

$$\begin{aligned} \text{ถ้าต้องการ 1200 บอด จะได้} &= 256 - \frac{1(OSCfreq)}{384(1200)} \\ &= 256 - \frac{11059000}{460800} \\ &= (256 - 24) = 232_{10} \end{aligned}$$

2.14 การออกแบบวงจรถอดรหัสสำหรับหน่วยความจำและไอโอ

วงจรถอดรหัสตำแหน่งมีความจำเป็นอย่างยิ่งในระบบไมโครโปรเซสเซอร์ เพราะเป็นตัวใช้เลือกชิปให้ทำงานทีละตัว มิฉะนั้นจะทำให้เกิดการชนกันของข้อมูล เนื่องจากชิปแต่ละตัวจะถูกต้องเชื่อมโยงเข้ากับบัสเดียวกันเช่น Address Bus และ Data Bus

2.14.1 วงจรถอดรหัสสำหรับหน่วยความจำ (Memory Decoder)

วงจรถอดรหัสหมายถึงวงจรที่ให้เอาต์พุตออกมาตามฟังก์ชันของอินพุตเช่น

2 to 4 line decoder

3 to 8 line decoder

4 to 16 line decoder

2.14.2 2 to 4 line decoder หมายถึงมีอินพุต 2 line และมีเอาต์พุต 4 line มีตารางความจริงตามตาราง

ที่ 2.3

ตารางที่ 2.3 แสดงค่าความจริงของ 2 to 4 line decoder

อินพุต		เอาต์พุต			
B	A	Y ₀	Y ₁	Y ₂	Y ₃
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

จากตารางที่ 2.3 สามารถอธิบายได้ดังนี้

อินพุตมี 2 ขา คือ A และ B โดยที่ A คือบิต LSB และ B คือบิต MSB

ถ้า B และ A มีค่าเป็น 0,0 ตามลำดับ หมายถึงเลือก Y₀

ถ้า B และ A มีค่าเป็น 0,1 ตามลำดับ หมายถึงเลือก Y₁

เอาต์พุตตามตารางที่ 1 นี้ แยกที่ฟลที่ Low (0)

2.14.3 2 to 8 line decoder หมายถึง มีอินพุต 3 line และมีเอาต์พุต 8 line ดังแสดงในตารางที่ 2.4

ตารางที่ 2.4 3 to 8 line decoder

อินพุต			เอาต์พุต							
C	B	A	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇
0	0	0	0	1	1	1	1	1	1	1
0	0	1	1	0	1	1	1	1	1	1
0	1	0	1	1	0	1	1	1	1	1
0	1	1	1	1	1	0	1	1	1	1
1	0	0	1	1	1	1	0	1	1	1
1	0	1	1	1	1	1	1	0	1	1
1	1	0	1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1	1	1	0

จากตารางที่ 2.4 บรรทัดที่ 1 จะพบว่า

เมื่ออินพุต C, B, A มีค่า 0,0,0 จะได้ 0 ที่ Y_0 นอกนั้นเป็น 1 หมด
เมื่ออินพุต C, B, A มีค่า 0,0,1 จะได้ 0 ที่ Y_1 นอกนั้นเป็น 1 หมด
เมื่ออินพุต C, B, A มีค่า 0,1,0 จะได้ 0 ที่ Y_2 นอกนั้นเป็น 1 หมด

3 to 8 line decoder with enable input

Decoder แบบนี้จะมีขาอินพุต 2 ขาคือ

Input Select

Input Enable

Input Select หมายถึง ขาอินพุตที่ใช้เลือกเอาต์พุต เช่น ขา C, B, A

Input Enable หมายถึง ขาอินพุตที่ใช้ควบคุมให้มีการถอดรหัสหรือไม่มีการถอดรหัสมีในเบอร์

74LS138 ตารางค่าความจริงตารางที่ 2.5

ตารางที่ 2.5 ตารางค่าความจริงของ 74LS138

Input Enable		Input Select			เอาต์พุต							
G1	G2*	C	B	A	Y_0	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7
X	1	X	X	X	1	1	1	1	1	1	1	1
0	X	X	X	X	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	1	1	1	1	1	1
1	0	0	0	1	1	0	1	1	1	1	1	1
1	0	0	1	0	1	1	0	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1	1	1
1	0	1	0	0	1	1	1	1	0	1	1	1
1	0	1	0	1	1	1	1	1	1	0	1	1
1	0	1	1	0	1	1	1	1	1	1	0	1
1	0	1	1	1	1	1	1	1	1	1	1	0

การทำงานของ 74LS138 สามารถอธิบายได้ดังนี้

ถ้าต้องการให้มีการถอดรหัสได้ต้องป้อนอินพุตที่ขา C, B, A โดยขา A จะเป็นบิต LSB ขา C จะเป็นบิต MSB

ส่วนขา G1 ต้องป้อน High

G2 ต้องป้อน Low

ถ้าต้องการถอดรหัส ก็ต้องป้อนอินพุตที่ขา C, B, A โดยขา A จะเป็นบิต LSB ขา C จะเป็นบิต MSB

เมื่ออินพุต C, B, A มีค่า 0,0,0 จะได้ 0 ที่ Y_0 นอกนั้นเป็น 1 หมด
เมื่ออินพุต C, B, A มีค่า 0,0,1 จะได้ 0 ที่ Y_1 นอกนั้นเป็น 1 หมด
เมื่ออินพุต C, B, A มีค่า 0,1,0 จะได้ 0 ที่ Y_2 นอกนั้นเป็น 1 หมด

2.14.4 การออกแบบวงจรถอดรหัส

ก่อนอื่นจะต้องศึกษาถึงคำสั่งของ MCS-51 ที่มีผลให้สถานะที่ขาแอดเดรสมีการเปลี่ยนแปลงอย่างไร เช่น คำสั่ง `MOVX@DPTR,A`

คำสั่ง `MOVX@DPTR,A` มีผลทำให้ไมโครคอนโทรลเลอร์ส่งค่าต่าง ๆ ออกมา 2 ค่าใหญ่ ๆ คือ

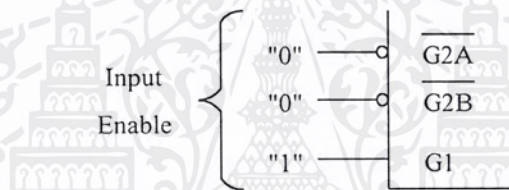
- 1) ข้อมูลในรีจิสเตอร์ A (DATA) จะส่งออกมาทาง DATA BUS
- 2) ขาแอดเดรสทั้ง 16 เส้น จะมีสถานะดังค่าใน DPTR

เช่น $A = 08H$, $DPTR = 0000H$

มีผลการทำงานดังนี้ DATA BUS จะปรากฏค่า 08H

แอดเดรสบัสจะปรากฏ 0000H (A_{15} ถึง A_0 มีค่าเป็น 0 ทั้งหมด)

การออกแบบวงจรถอดรหัสของหน่วยความจำ จำเป็นต้องศึกษา IC ที่จะนำมาถอดรหัสตามจุดประสงค์เบอร์ที่นิยมก็คือ 74LS138 (3 to 8 Line Decoder with Enable input) มีตารางค่าความจริงตามตารางที่ 2.5



รูปที่ 2.23 การต่อขา Input Enable

จากตารางที่ 2.5 จะพบว่าทางด้านอินพุตมี 2 ชุด คือ

1. Input Select มีขา A, B, C ใช้เลือกเอาต์พุต Y_0 ถึง Y_7 ,
2. Input Enable มีขา G1 และ G2 ใช้ควบคุมให้ถอดรหัสหรือห้ามการถอดรหัสโดย

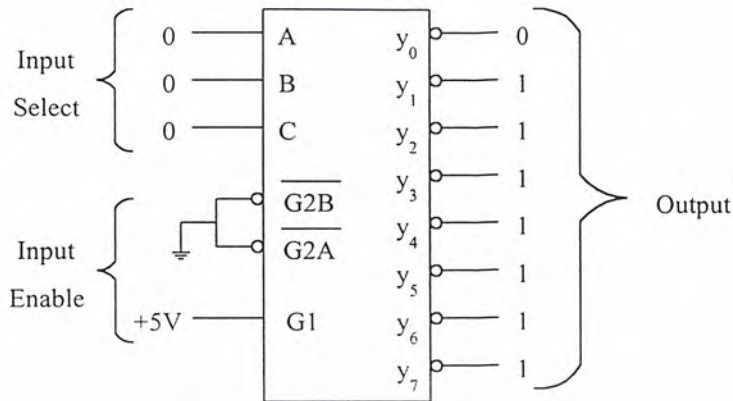
G1 ต้องการ High

G2 ต้องการ Low

(G2 ก็คือขา G2A และ G2B)

เอาต์พุตจะมีขา Y_0 ถึง Y_7 โดยจะแยกที่ที่ Low เพียงเอาต์พุตเดียว

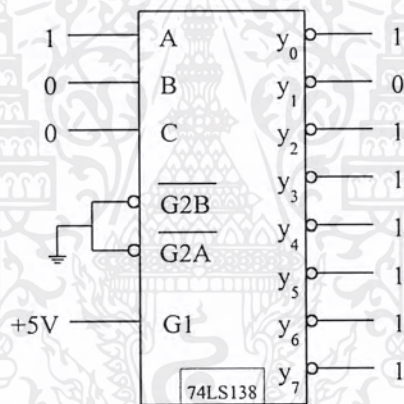
ถ้าอินพุตที่เข้ามาเป็น Low ที่ A, B, C ก็จะได้ $Y_0 = Low$ ที่เหลือจะเป็น High ดังรูปที่ 2.24



รูปที่ 2.24 การถอดรหัสเมื่ออินพุต C, B, A = 0,0,0

เมื่ออินพุต C, B, A มีค่าเป็น 0,0,1 จะได้เอาต์พุตเป็น Low ที่ Y_1 ส่วนขาที่เหลือจะเป็น High ดังรูปที่

2.25

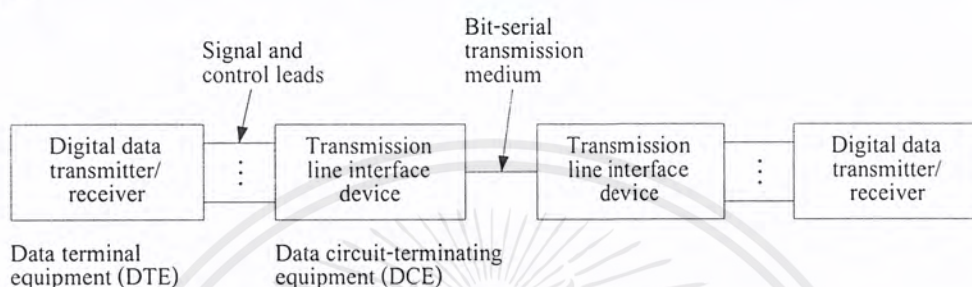


รูปที่ 2.25 เมื่ออินพุต C, B, A = 0,0,1

2.14.5 การเชื่อมต่อ (INTERFACING)

อุปกรณ์ประมวลผลข้อมูลดิจิทัลจำนวนมาก จะมีความสามารถในการส่งผ่านข้อมูลที่จำกัดค่าหนึ่ง ซึ่งอุปกรณ์เหล่านี้จะให้กำเนิดเพียงสัญญาณดิจิทัลเท่านั้น และสัญญาณดิจิทัลเหล่านี้ส่วนใหญ่จะอยู่ในรูป NRZ-L ระยะในการส่งข้อมูลที่อยู่ในช่วงที่จำกัด ดังนั้นจึงเป็นเรื่องที่ยากที่จะมีการติดต่อส่งผ่านข้อมูลไปในตัวกลางได้โดยตรง สถานภาพที่รู้จักกันโดยทั่วไป แสดงไว้ในรูป 2.26 ซึ่งเป็นอุปกรณ์ที่เราากำลังกล่าวถึงอยู่ ซึ่งรวมทั้งเทอร์มินอลและคอมพิวเตอร์ นั้นนิยมเรียกกันว่า data terminal equipment (DTE) DTE จะใช้ระบบการส่งผ่านข้อมูลโดยอาศัย data circuit terminal equipment (DCE) เป็นตัวช่วยตัวอย่างที่ผ่านมาก็ได้แก่โมเดม ในด้านหนึ่ง DCE จะรับผิดชอบต่อการส่งและรับบิตข้อมูล ที่เวลาใด ๆ บนตัวกลางการส่งผ่านในอีกด้านหนึ่ง DCE จะต้องมีการติดต่อรับส่งข้อมูลกับ DTE โดยทั่วไปแล้วก็จะเป็นทั้งสัญญาณข้อมูลและสัญญาณการควบคุมที่จะมีการแลกเปลี่ยน ซึ่งการแลกเปลี่ยนจะกระทำกันโดยอาศัยกลุ่มของสายที่เรียกว่า interchange circuit สำหรับกระบวนการทำงานนี้ต้องการการประสานงานกันอย่างดี DCE ทั้งสองตัวจะต้องเข้าใจซึ่งกันและกัน

เป็นอย่างดี นั่นคือเครื่องรับของแต่ละฝ่ายจะต้องใช้รหัสชนิดเดียวกันกับเครื่องส่ง เพื่อให้การทำงานดังกล่าวสำเร็จลงได้ DTE-DCE ทุก ๆ คู่จะถูกออกแบบให้มีการเชื่อมประสานที่เข้ากันได้ และจะต้องสามารถติดต่อซึ่งกันและกันได้อย่างมีประสิทธิภาพเพื่อความสะดวก จึงต้องมีการกำหนดมาตรฐานของการเชื่อมต่อขึ้นมาระหว่าง DTE กับ DCE ในปัจจุบันมาตรฐานเหล่านี้เป็นที่รู้จักกันในชื่อ โปรโตคอล ระดับกายภาพ (Physical layer protocol) ซึ่งครอบคลุมเลเยอร์ 1 ของ OSI โมเดล ในหัวข้อนี้เราจะได้กล่าวถึงคุณสมบัติและหน้าที่ของอินเตอร์เฟส



รูปที่ 2.26 การเชื่อมต่อกับตัวกลางการส่งผ่าน

การเชื่อมต่อมีคุณสมบัติที่สำคัญอยู่ 4 ประการคือ

- Mechanical
- Electrical
- Functional
- Procedural

- คุณสมบัติทางกลไก จะเกี่ยวข้องกับการเชื่อมต่อทางกายภาพอย่างแท้จริงของ DTE กับ DCE ตัวอย่างเช่น ตัวนำสัญญาณและการควบคุมจะมัดรวมกันเป็นสายเคเบิลและต่อกันด้วยตัวต่อตัวผู้หรือตัวเมีย ที่แต่ละขั้ว DTE และ DCE จะต้องมีตัวต่อที่ตรงข้ามกันที่ขั้วหนึ่งของสาย

- คุณสมบัติทางไฟฟ้า จะต้องมีการกำหนดระดับศักย์ไฟฟ้า และเวลาของการเปลี่ยนระดับศักย์ไฟฟ้าซึ่งทั้ง DTE และ DCE จะต้องใช้รหัสเดียวกัน จะต้องใช้ระดับศักย์ไฟฟ้าที่มีความหมายเดียวกันและจะต้องใช้เวลาระหว่างสัญญาณแต่ละตัวให้เท่ากัน ซึ่งคุณสมบัติเหล่านี้ทำให้เราสามารถหาค่าอัตราการส่งข้อมูลและกำหนดระยะเวลาในการส่งได้

- คุณสมบัติของการทำงาน เป็นการกำหนดหน้าที่ที่จะต้องปฏิบัติ โดยการกำหนดความหมายของสายการแลกเปลี่ยนข้อมูลให้มีหน้าที่เท่า ๆ กัน หน้าที่ของสายต่าง ๆ จะสามารถแบ่งออกได้เป็นประเภท สายข้อมูล สายควบคุม สายเวลา และสายดิน

- หลักการขั้นตอนการดำเนินงาน กำหนดให้เป็นลำดับของเหตุการณ์ในการส่งผ่านข้อมูลซึ่งเป็นหน้าที่ของคุณสมบัติของฟังก์ชันนอลของการเชื่อมต่อ ตัวอย่างต่อไปนี้จะทำให้เราเข้าใจจุดนี้ได้ดีขึ้น

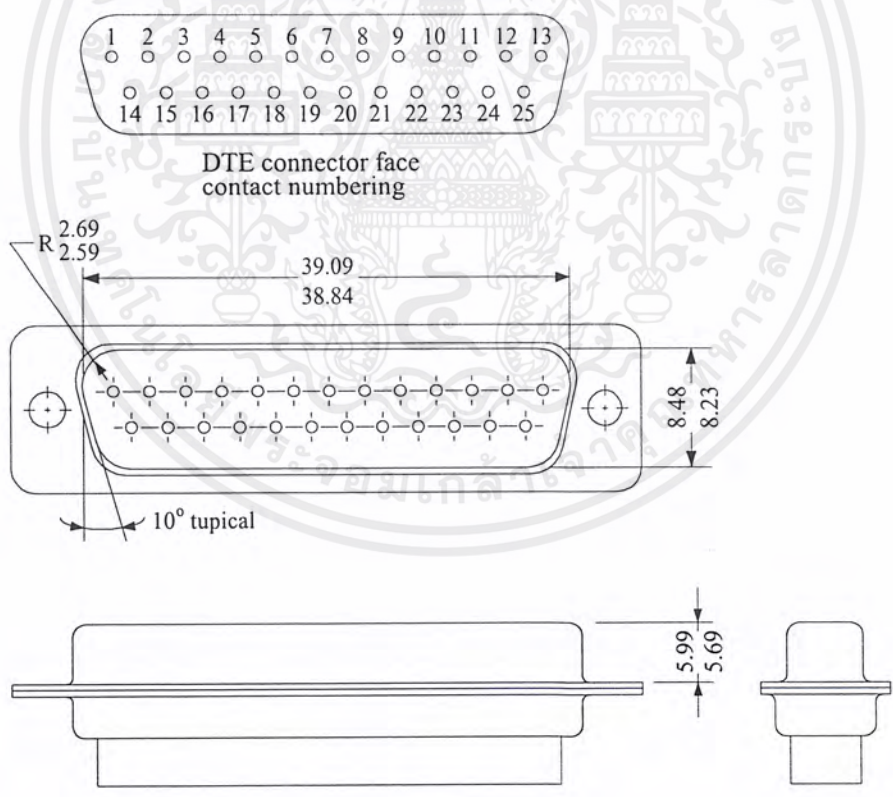
มาตรฐานต่าง ๆ ในการเชื่อมต่อที่ใช้กันอยู่มีมากมาย เช่น RS-232-C, RS-449/422-A/423-A และ X.21 สำหรับในหัวข้อนี้จะแสดงมาตรฐาน RS-232-C เท่านั้น

2.14.6 RS-232-C

มาตรฐานการเชื่อมต่อที่เป็นที่รู้จักกันแพร่หลายอันหนึ่งก็คือ RS-232-C ซึ่งใช้ในการเชื่อมต่อ DTE เข้ากับโมเด็มของ voice-grade เพื่อใช้ในระบบโทรศัพท์ ซึ่งก็ได้รับความนิยมอย่างมากในการทำหน้าที่เชื่อมต่อ กับระบบอื่น ๆ ด้วย

ข้อกำหนดทางกลไกของ RS-232-C อธิบายได้ดังในรูป 2.23 ซึ่งเรียกกันว่าตัวต่อ 25 ขา โดยจัดแบ่งขาออกเป็นกลุ่ม ๆ ซึ่งในทางทฤษฎีกำหนดว่าควรมีสาย 25 เส้นต่อถึงกันระหว่าง DTE กับ DCE แต่ในทางปฏิบัติอาจมีเพียงสายไม่กี่เส้นเท่านั้นที่ถูกลำมาต่อกันจริง ๆ

คุณสมบัติทางไฟฟ้า กำหนดว่าสัญญาณระหว่าง DTE กับ DCE ใช้สัญญาณดิจิทัลส่งผ่านสายตาม ข้อตกลงที่กำหนดขึ้นคือว่า จะต้องต่อกราวด์ร่วมกัน, ศักย์ไฟฟ้าที่เป็นลบมากกว่า -3 V จะถูกแปลเป็นไบนารี 1 และศักย์ไฟฟ้าที่มากกว่า +3 V จะถูกแปลให้เป็น 0 การเชื่อมต่อจะสัมพันธ์กับอัตราสัญญาณที่น้อยกว่า 20 kbps และระยะทางน้อยกว่า 15 เมตร สำหรับค่าระยะทางและอัตราการส่งที่สูงกว่านี้ก็สามารถทำได้ด้วยการ ออกแบบที่ดี แต่มันก็เป็นเหตุผลที่เขายึดถือลงไปว่าข้อจำกัดดังกล่าวสามารถประยุกต์ใช้ได้ทั้งในการใช้งาน ในทางปฏิบัติและในทางทฤษฎี



รูปที่ 2.27 RS-232-C connector

ตารางที่ 2.6 ได้รวบรวมข้อกำหนดคุณสมบัติหน้าที่ของสายที่สำคัญเอาไว้ มีสายข้อมูล 1 เส้นในแต่ละทิศทาง ดังนั้นกระบวนการแลกเปลี่ยนแบบ full duplex จึงสามารถทำได้ สายกราวด์มีไว้เพื่อกราวด์เครื่อง

เพื่อแยกการป้องกันและอีกเส้นหนึ่งใช้เป็นสายกราวด์รวมของสัญญาณ ดังนั้นการส่งผ่านแบบนี้จึงเป็นแบบไม่สมดุลโดยมีสายแอกทีฟเพียงสายเดียว สัญญาณเวลาจะกำหนดพัลส์นาฬิกาสำหรับการส่งผ่านแบบซิงโครนัส เมื่อ DCE ส่งข้อมูลไปในสาย BB มันก็จะส่งค่าการเปลี่ยนแปลงสถานะ 1→0 และ 0→1 ลงไปบนสาย DD ด้วย เวลาการเปลี่ยนสถานะจะกำหนดอยู่ที่กึ่งกลางขององค์ประกอบสัญญาณบน BB แต่ละตัว เมื่อ DTE ส่งข้อมูลอาจจะเป็น DTE หรือ DCE ที่เป็นตัวกำหนดพัลส์เวลาซึ่งขึ้นอยู่กับสภาพแวดล้อม เรื่องเกี่ยวกับสัญญาณควบคุมได้อธิบายไว้ในข้อกำหนดของขบวนการรับ-ส่ง ซึ่งจะได้แสดงตัวอย่างกันต่อไป

ตารางที่ 2.6 วงจรสับเปลี่ยน RS-449 และ RS-232-C

RS-449		RS-232-C	
SG	Signal Ground	AA	Protective Ground
SC	Send Common	AB	Signal Ground
RC	Receive Common		
IS	Terminal in Service		
IC	Incoming Call	CE	Ring Indicator
TR	Terminal Ready	CD	Data Terminal Ready
DM	Data Mode	CC	Data Set Ready
SD	Send Data	BA	Transmitted Data
RD	Receive Data	BB	Received Data
TT	Terminal Timing	DA	Transmitter Signal Element Timing (DTE source)
ST	Send Timing	DB	Transmitter Signal Element Timing (DCE source)
RT	Receive Timing	DD	Receiver Signal Element Timing
RS	Request to Send	CA	Request to Send
CS	Clear to Send	CB	Clear to Send
RR	Receiver Ready	CF	Received Line Signal Detector
SQ	Signal Quality	CG	Signal Quality Detector
NS	New Signal		
SF	Select Frequency		
SR	Signaling Rate Selector	CH	Data Signal Rate Selector (DTE source)
SI	Signaling Rate Indicator	CI	Data Signal Rate Selector (DCE source)
SSD	Secondary Send Data	SBA	Secondary Transmitted Data
SRD	Secondary Receive Data	SBB	Secondary Received Data
SRS	Secondary Request to Send	SCA	Secondary Request to Send
SCS	Secondary Clear to Send	SCB	Secondary Clear to Send

SRR	Secondary Receiver Ready	SCF	Secondary Received Line Signal Detector
LL	Local Loopback		
RL	Remote Loopback		
TM	Test Mode		
		Pins 9 and 10 Test Function	
SS	Select Standby		
SB	Standby Indicator		

Category I circuits.

ตัวอย่างแรกคือกรณีการส่งผ่านแบบอะซิงโครนัสของโมเด็มที่ใช้สายเช่าซึ่งเป็นการต่ออุปกรณ์แบบจุดต่อจุด การใช้โมเด็มเหมือนกับ DCE ในกรณีนี้ มีความจำเป็นในการใช้สายต่าง ๆ เพียงดังต่อไปนี้คือ

- Signal Ground (AB)
- Transmitted data (BA)
- Received data (BB)
- Requested to Send (CA)
- Clear to send (CB)
- Data set ready (CC)
- Carrier detect (CF)

ตามสายแรกมีหน้าที่ต่อไปนี้คือเมื่อ DTE พร้อมที่จะส่งข้อมูลมันจะยืนยันด้วย สัญญาณ Request-to-send โมเด็มก็จะตอบสนอง เมื่อมันพร้อมด้วย Clear-to-send โดยที่เป็นการแสดงว่าข้อมูลจะถูกส่งไปในสาย BA ซึ่งถ้าการเตรียมการส่งเป็นแบบ half-duplex แล้ว Request-to-send จะแสดงในโมเด็มด้วย สาย Data-set-Ready จะถูกยืนยัน เมื่อโมเด็มพร้อมที่จะทำงานการแสดงดังกล่าวนี้ควรจะถูกยืนยันก่อนที่ DTE จะพยายามร้องขอด้วยสัญญาณ Request to send และสุดท้าย Carrier Detect จะเป็นตัวแสดงให้รู้ว่าโมเด็มที่อยู่ไกลออกไปกำลังส่งข้อมูล นั่นแสดงว่ามันไม่จำเป็นต้องใช้สายสัญญาณเวลา เพราะการส่งนี้เป็นแบบอะซิงโครนัส

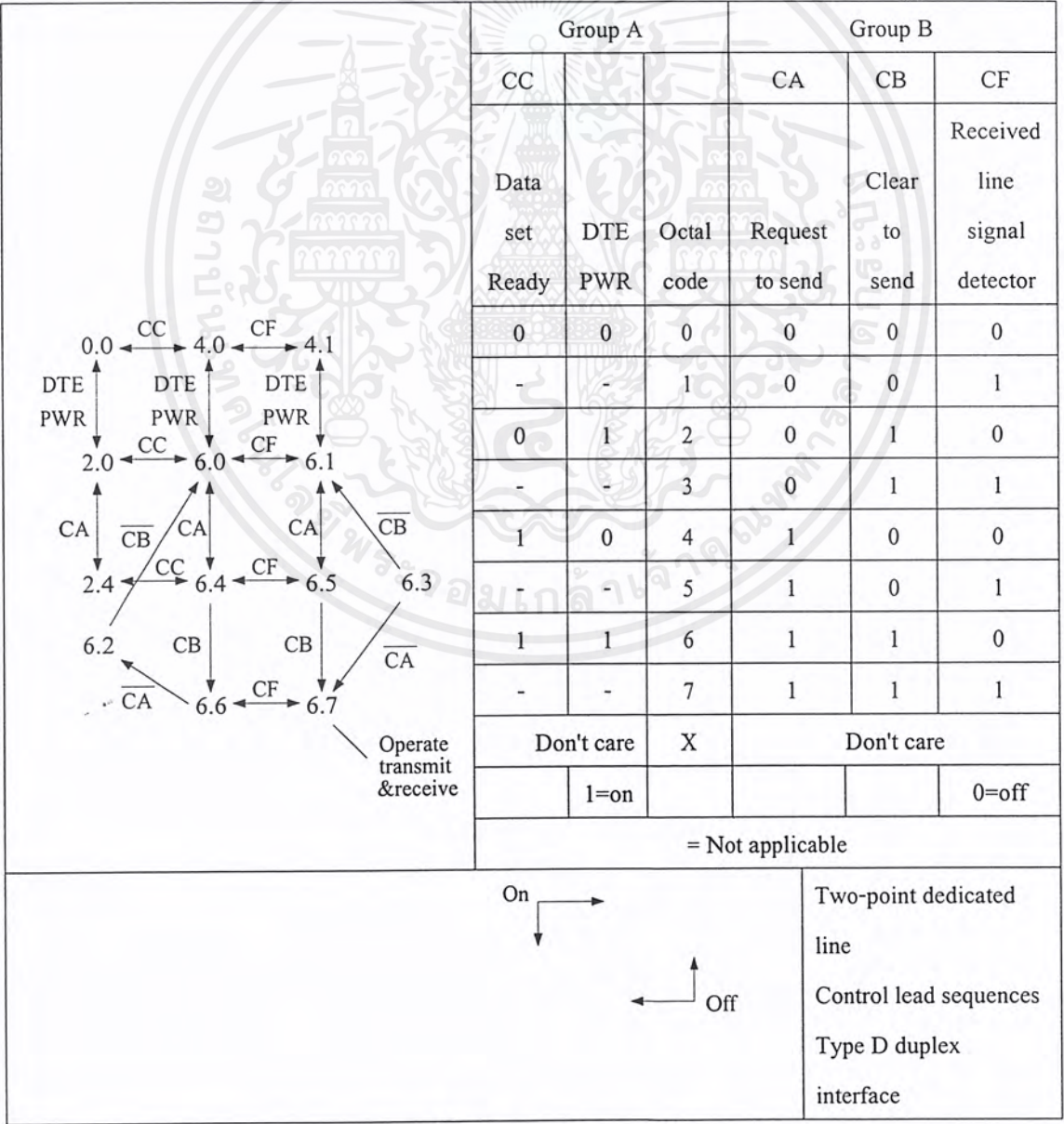
รูปที่ 2.28 แสดงแผนภูมิสถานะภาพของขบวนนี้ A ได้แบ่งสายควบคุมหลักออกเป็น 2 กลุ่ม กลุ่ม A (CC, CD, CE) จะเกี่ยวข้องกับการเลือกและความพร้อมของอุปกรณ์ในการทำงาน และกลุ่ม B (CA, CB, CF) เกี่ยวข้องกับการเตรียมอุปกรณ์การสื่อสารเพื่อส่งและรับข้อมูล สถานะของ ON และ OFF ของทั้ง 2 กลุ่มที่จะถูกเข้ารหัสโดยใช้ค่า 8 bit เป็นตัวบ่งชี้ การเข้ารหัสนี้รวมทั้งการ ON power ในขบวนการนี้ด้วย

สำหรับสายต่าง ๆ ก็ได้อธิบายมาพอสมควรแล้วในเรื่องสายโมเด็มอิสระที่ต่อแบบจุดต่อจุด แต่สำหรับสายที่เพิ่มขึ้นมาถูกกำหนดขึ้นเมื่อเราใช้โครงข่ายโทรศัพท์ในการส่งผ่านข้อมูล ปัจจุบันตัวเริ่มต้นของการติดต่อจะต้องเรียกไปยังสถานีปลายทาง ตัวแสดงที่เพิ่มขึ้นมา 2 ตัวที่ต้องการมีดังนี้

- Data terminal ready (CD)
- Ring Indicator (CE)

โดยการเพิ่มตัวแสดงสองตัวนี้ทำให้ระบบโมเด็ม DTE สามารถใช้สายโทรศัพท์ อุปมาเหมือนกับใช้ ในการส่งผ่านเสียงได้ตารางที่ 2.7 แสดงขั้นตอนที่เกี่ยวข้องกับขบวนการเรียกและรับ เมื่อการเรียกเกิดขึ้น ทั้ง แบบเรียกโดยคนหรือแบบอัตโนมัติ ระบบโทรศัพท์จะส่งสัญญาณกระดิ่งไป เครื่องรับโทรศัพท์จะตอบสนอง ด้วยการส่งเสียงกริ่งออกมา แต่โมเด็มจะตอบสนองด้วยการยืนยันด้วย Data terminal ready สำหรับคนก็จะได้ ยินเสียงของอีกฝ่ายหนึ่งถ้าไม่ได้ยินก็วางหู แต่ DTE ฟัง Carrier Detect ซึ่งจะเกิดขึ้นโดยโมเด็มเมื่อ Carrier ถูก ส่งออกมา และถ้าสายนี้ไม่ถูกยืนยันสาย Data terminal ready ก็จะถูกดับไปซึ่งเหตุการณ์นี้เกิดขึ้นได้ในกรณีที่มี คนหมุนเบอร์โทรศัพท์เข้ามาผิดเบอร์

นอกจากนี้แล้วถ้าระยะทางระหว่างคอมพิวเตอร์สั้น ๆ เราอาจจะส่งผ่าน RS-232-C กันได้โดยตรงโดยไม่ผ่านโมเด็มซึ่งเราเรียกว่าเป็นวิธีการใช้ "null modem" ซึ่งก็จะต้องมีการต่อสายสัญญาณต่าง ๆ ดังแสดงไว้ใน รูปที่ 2.29



รูปที่ 2.28 แผนภาพสถานะของ RS-232-C

ตารางที่ 2.7 ขบวนการได้อัลของ RS-232-C

Establish Connection

1. TA sets Data Terminal Ready ON, to indicate to the modem that it is ready to make a call.
2. MA opens the phone circuit and dials number. Number may be already stored in modem, or it is conveyed to the modem from TA via Transmitted Data pin.
3. MB detects ring on phone line and set Ring Indicator ON, to inform TB that a call has arrived.
4. TB sets Data Terminal Ready ON, to accept the call.
5. MB "answer" the call by transmitting a carrier signal to MA over the phone line. It also sets Data set Ready ON, to indicate to TB that it is ready to communicate.
6. MA sets both Data Set Ready and Carrier Detect ON, to indicate to TA that a circuit has been established and that it is ready to communicate.
7. For full-duplex operation, MA also transmits a carrier signal to MB MB then sets Carrier Detect ON.

Send Data

8. When TA wants to send data, it sets Request to Send ON.
9. MA responds by setting Clear to Send ON.
10. TA transmit data one bit at time to MA over the Transmitted Data pin. MA modulates its carrier signal (e.g. ASK) to transmit the data to MB.
11. MB receives the modulated signal from MA, demodulates to recover the data, and sends the data to TB over the Received Data pin.

Break Connection

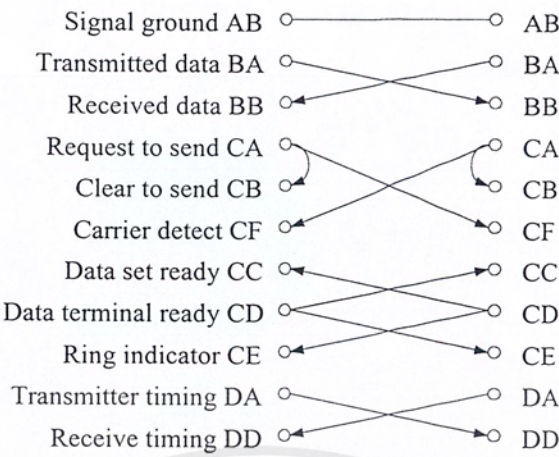
12. When TA is finished, it sets Request to Send OFF.
13. MA sets Clear to Send OFF and drops the carrier (hangs up)

TA = DTE A

MA = Modem A

TB = DTE B

MB = Modem B



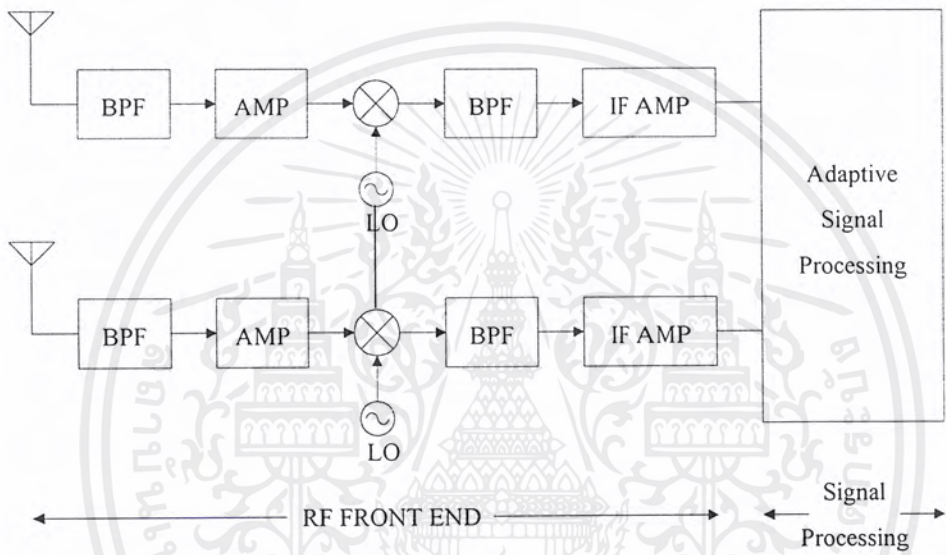
รูปที่ 2.29 null modem



บทที่ 3

การออกแบบและการสร้าง

เนื่องจากสายอากาศแบบปรับตัวได้จะมีความสามารถในการกดสัญญาณรบกวนและปรับค่าคลื่นให้อยู่ในทิศทางของสัญญาณที่แรงที่สุดเสมอ ด้วยเหตุนี้ การทำงานของสายอากาศแบบปรับตัวได้ จึงต้องมีการทำงานที่เป็นระบบหลายระบบทำงานร่วมกัน ดังแสดงในรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมของระบบสายอากาศปรับตัวได้

สำหรับในส่วนของสายอากาศนั้น จะประกอบไปด้วยสายอากาศชนิดแถวลำดับ และเครื่องแปลงผันความถี่ลง (Down converter)

แต่เนื่องจากโครงงานนี้เน้นการศึกษาและออกแบบการประมวลสัญญาณ ดังนั้นจึงนำมาเฉพาะส่วนการประมวลสัญญาณเบสแบนด์

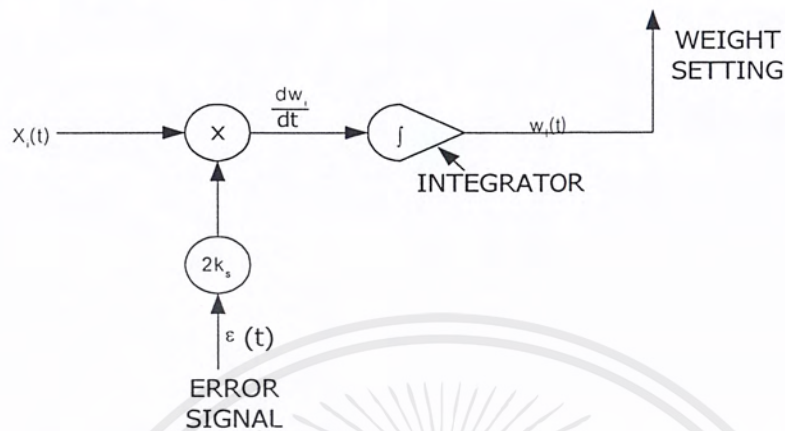
สำหรับการออกแบบวงจรเพื่อประมวลสัญญาณนี้ จะใช้อัลกอริทึมกำลังสองเฉลี่ยน้อยที่สุด (LMS Algorithm) เป็นแนวทางในการออกแบบวงจร

โดยกระบวนการของอัลกอริทึมกำลังสองเฉลี่ยน้อยที่สุด มีค่าถ่วงน้ำหนัก (Weight) ที่เหมาะสม สายอากาศแต่ละตัวจะมีค่าถ่วงน้ำหนักเฉพาะ ทำให้ผลจากกระบวนการนี้ในสายอากาศแต่ละตัวแตกต่างกันไป เนื่องจากค่าถ่วงน้ำหนักนี้ไม่ขึ้นอยู่กับสัญญาณอินพุตที่มาจากสายอากาศแต่ละตัว

จากทฤษฎีในบทที่ 2 ได้สมการของกระบวนการนี้คือ

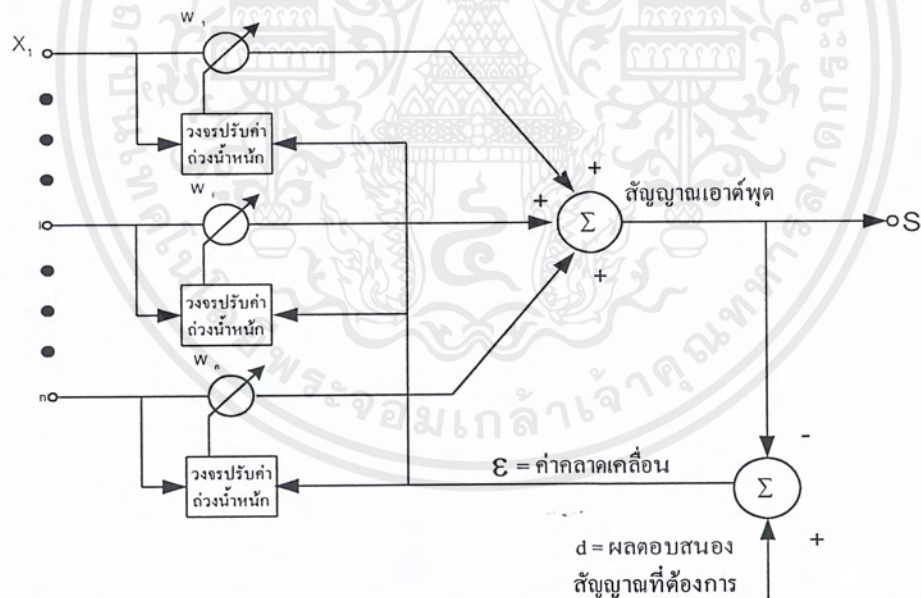
$$W(t) = -2k_s \int_{\epsilon}(\xi) X(t) d\xi$$

และนำมาออกแบบเป็นวงจรที่เหมาะสม



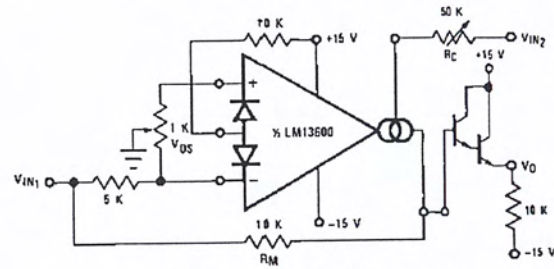
รูปที่ 3.2 บล็อกไดอะแกรมของอัลกอริทึมกำลังสองเฉลี่ยน้อยที่สุดแบบอนาล็อก

จากวงจรหาค่าถ่วงน้ำหนัก เมื่อนำมาเข้าสู่กระบวนการประมวลสัญญาณในระบบสายอากาศแบบปรับตัวได้ จะได้ดังรูปที่ 3.3



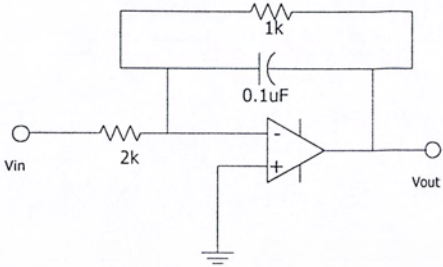
รูปที่ 3.3 เครื่องมืออนาล็อกสำหรับอัลกอริทึมกำลังสองเฉลี่ยน้อยที่สุดที่มีการปรับแก้ค่าถ่วงน้ำหนัก

- การออกแบบวงจร



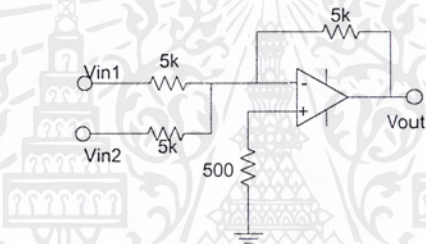
รูปที่ 3.4 วงจรที่ทำกรออกแบบ

- การออกแบบวงจรอินทิเกรต



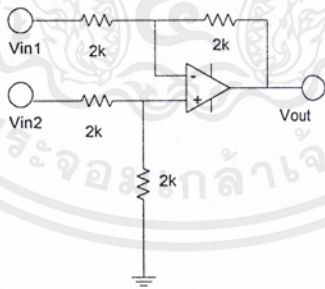
รูปที่ 3.5 วงจรอินทิเกรตที่ทำการออกแบบ

- การออกแบบวงจรบวก

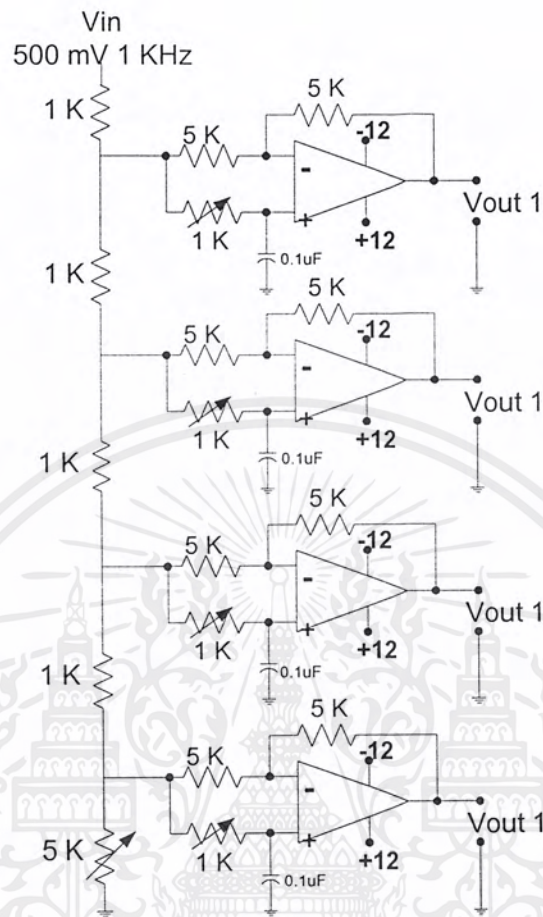


รูปที่ 3.6 วงจรบวกที่ทำการออกแบบ

- การออกแบบวงจรลบ



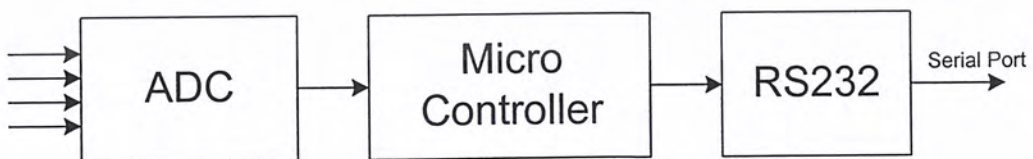
รูปที่ 3.7 วงจรลบที่ทำการออกแบบ



รูปที่ 3.9 วงจรสมมุติสัญญาณ

- การออกแบบในส่วนอุปกรณ์ต่อรวม

จุดประสงค์ของการออกแบบในส่วนนี้ เนื่องจากว่าในกระบวนการทำงานของระบบสายอากาศแบบปรับตัวได้ ที่ต้องอาศัยการประมวลสัญญาณโดยใช้ซอฟต์แวร์จะต้องมีการส่งสัญญาณที่เป็นสัญญาณต่อเนื่องมาจากสายอากาศแบบแถวลำดับ เข้าสู่คอมพิวเตอร์เพื่อประมวลผล



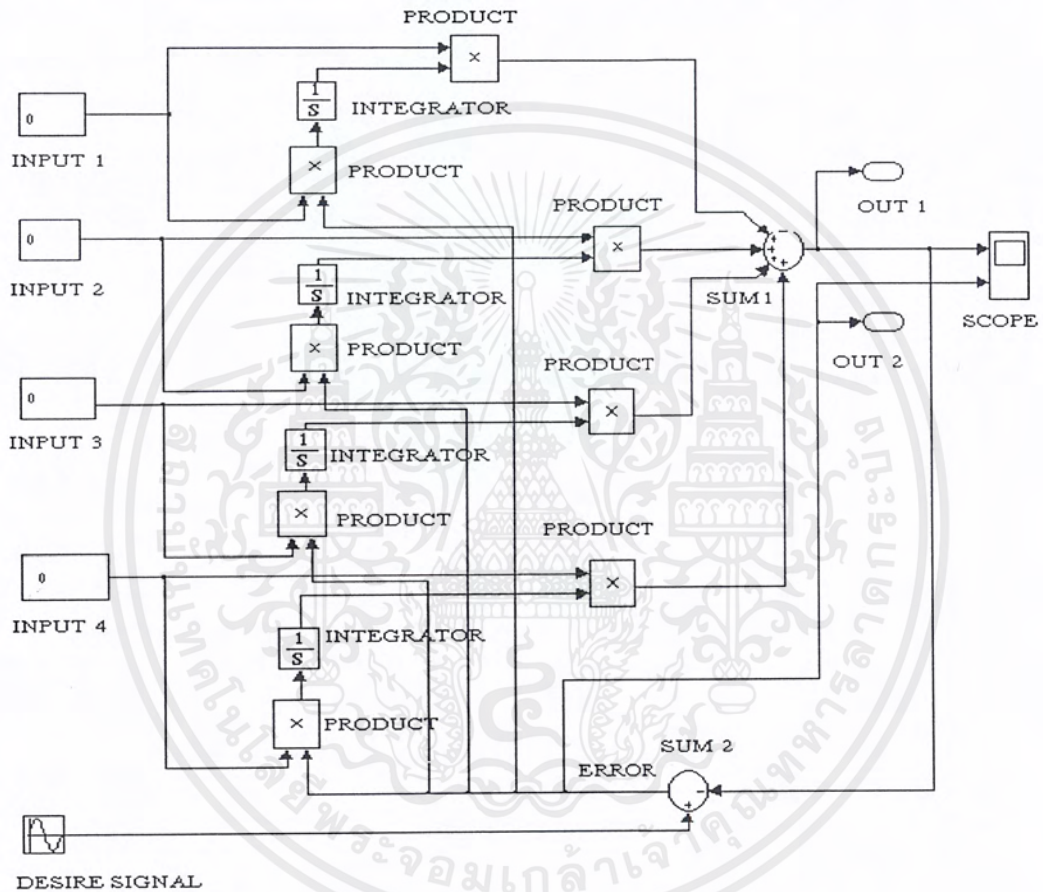
รูปที่ 3. 10 บล็อกไดอะแกรมแสดงส่วนการออกแบบ

หลักการทำงานของส่วนที่ออกแบบนี้ ไมโครคอนโทรลเลอร์จะควบคุมการทำงานของตัวแปลงอนาลอกเป็นดิจิทัล โดยทำงานในลักษณะสลับกันทำงานไปเรื่อยๆ โดยการทำงานของ ADC จะถูกเลือกโดย I/P Decoder เบอร์ 74LS138

เมื่อ ADC แต่ละตัวทำงานและส่งข้อมูลเข้ามา ไมโครคอนโทรลเลอร์จะส่งผ่านทางพอร์ตอนุกรมไปผ่าน MAX 232 เพื่อส่งข้อมูลที่ได้อ่านไปยังพอร์ตอนุกรมของคอมพิวเตอร์ต่อไป

- ส่วนของการออกแบบในส่วนของการประมวลสัญญาณดิจิทัล

ใช้โปรแกรมแมทแลบมาใช้ในการปฏิบัติงาน เนื่องจากความสะดวกของการนำมาใช้งาน และการแสดงผลสามารถแสดงผลผ่านจอคอมพิวเตอร์ได้ หลักการทำงานคือ นำสัญญาณที่รับเข้ามาจากพอร์ตอนุกรมไปเป็นอินพุตในส่วน of toolbox ที่ได้ออกแบบตามหลักการของอัลกอริทึมกำลังสองเฉลี่ยน้อยที่สุดเหมือนกับเป็นเวลาจริง

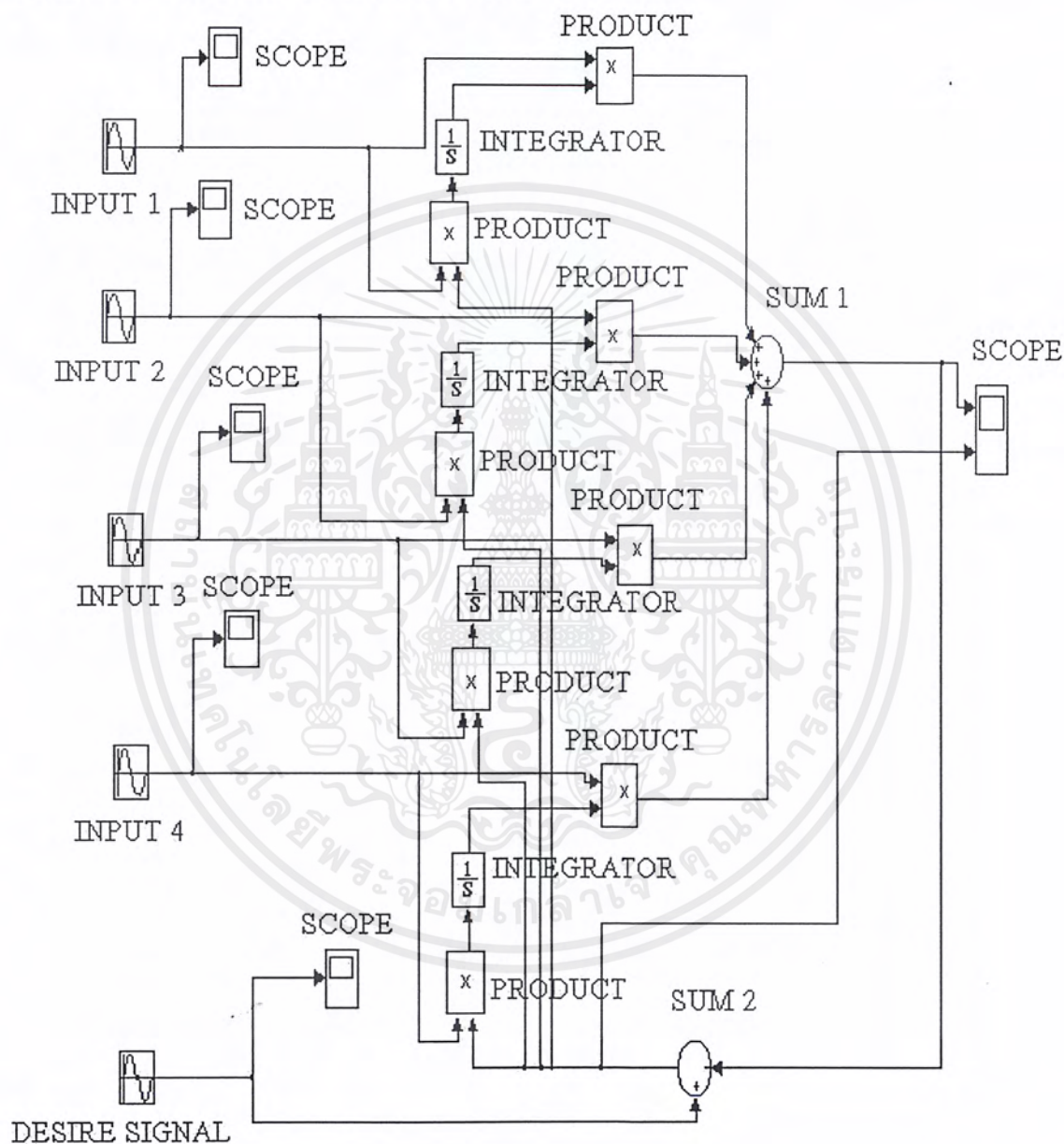


รูปที่ 3.11 บล็อกไดอะแกรมแสดงการออกแบบในส่วนซอฟต์แวร์

บทที่ 4

การทดลองและผลการทดลอง

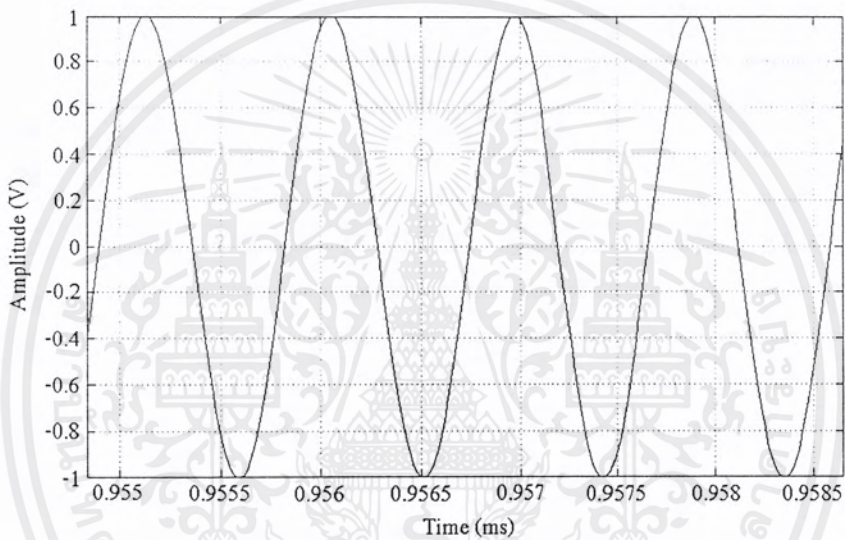
จากวงจรที่ได้ออกแบบไว้ ได้ทำการจำลองการทำงานโดยใช้โปรแกรมซิมูลิงค์ (Simulink) ในแมทแล็บ (Matlab) ซึ่งแสดงการทำงานดังบล็อกไดอะแกรม



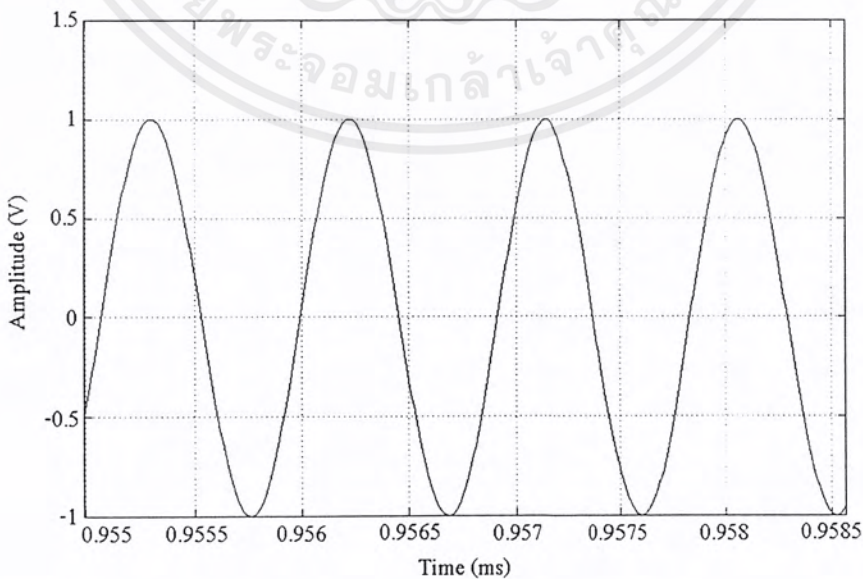
รูปที่ 4.1 บล็อกไดอะแกรมแสดงแบบจำลองการทำงานของวงจร

ผลจากการจำลองการทำงานของวงจรที่เวลาต่างๆ ในหน่วยมิลลิวินาที ซึ่งกราฟจะแสดงให้เห็นถึงการเปลี่ยนแปลงของสัญญาณอินพุต เอาต์พุตค่าความคลาดเคลื่อนซึ่งมีค่าแอมพลิจูดเปลี่ยนไปตามเวลาที่เปลี่ยนไป

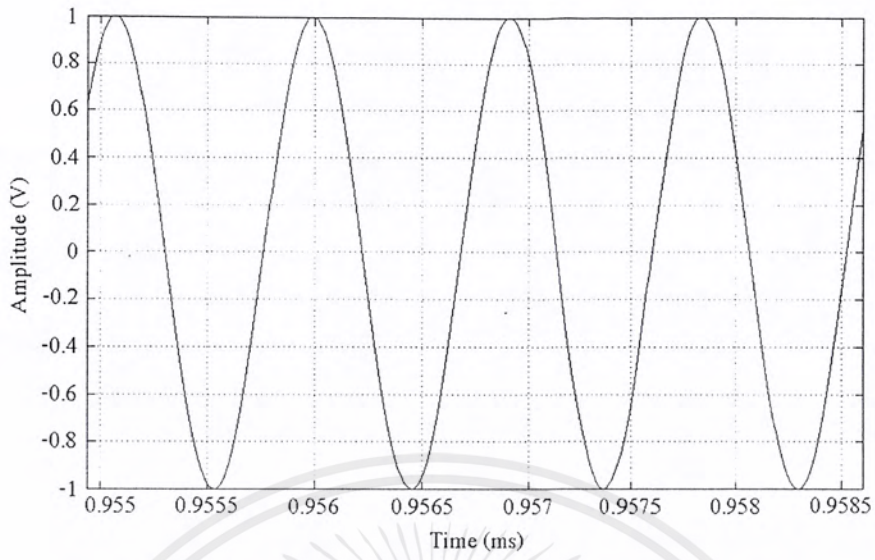
จากแบบจำลองการทำงานนี้ กำหนดให้อินพุตมีทั้งหมด 4 อินพุต ซึ่งแต่ละสัญญาณจะมีความถี่เท่ากัน แต่มีเฟสที่แตกต่างกันออกไปตามค่าของเวลาประวิง (Delay time) ซึ่งเกิดจากการสมมติให้สัญญาณรบกวนที่รับเข้ามานั้นเข้าในทิสทำมุมกับสัญญาณนำร่อง



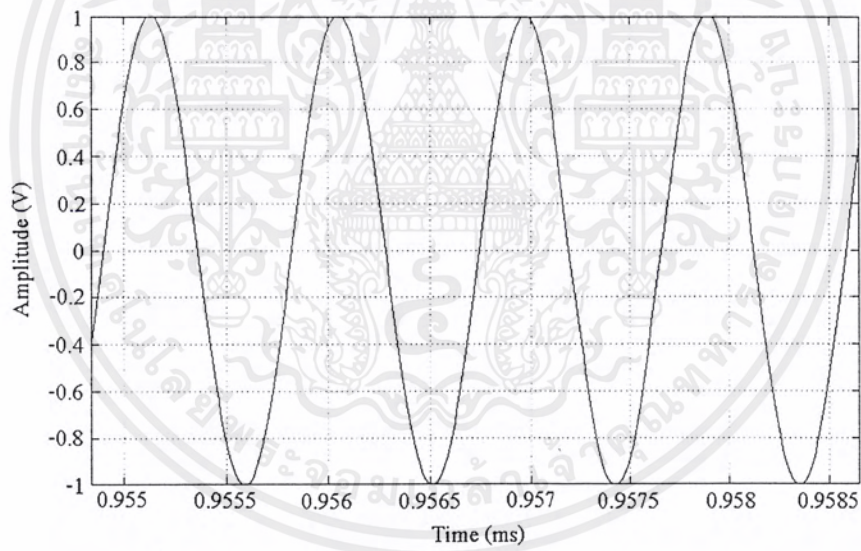
รูปที่ 4.2 สัญญาณอินพุต 1 ของแบบจำลองการทำงานของวงจร



รูปที่ 4.3 สัญญาณอินพุต 2 ของแบบจำลองการทำงานของวงจร



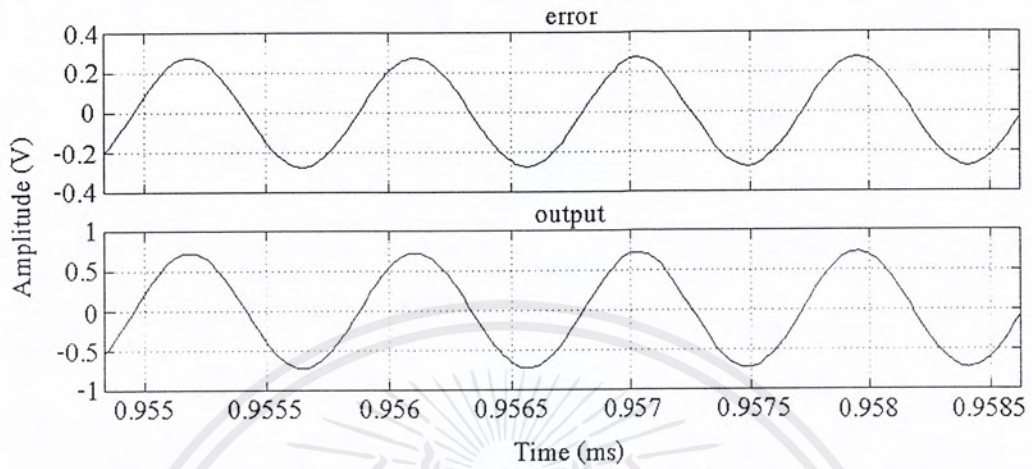
รูปที่ 4.4 สัญญาณอินพุต 3 ของแบบจำลองการทำงานของวงจร



รูปที่ 4.5 สัญญาณอินพุต 4 ของแบบจำลองการทำงานของวงจร

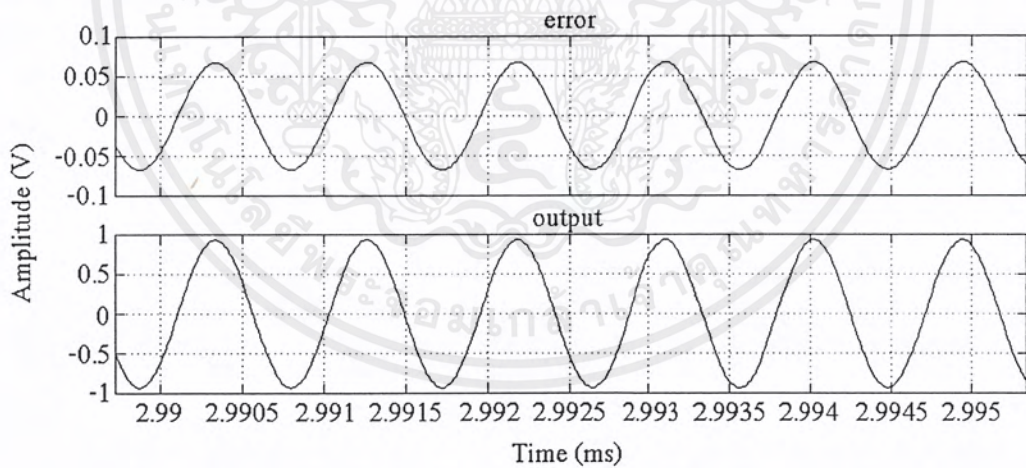
กราฟแสดงสัญญาณเอาต์พุตและสัญญาณคลาดเคลื่อนที่แสดงในช่วงเวลาที่ต่างกัน

ช่วงเวลาที่ 1 ใช้เวลา 1 มิลลิวินาที



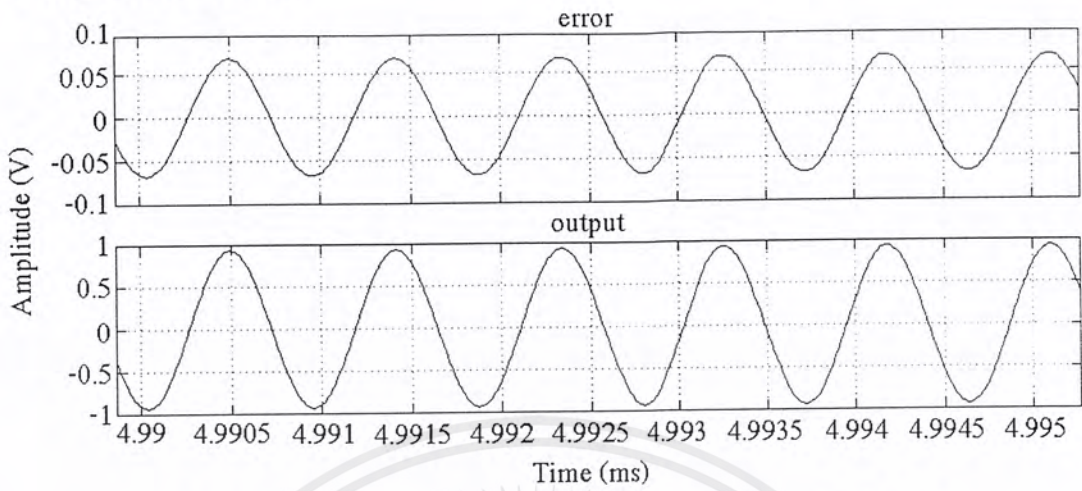
รูปที่ 4.6 สัญญาณเอาต์พุตและสัญญาณคลาดเคลื่อนเมื่อใช้เวลา 1 มิลลิวินาที

ช่วงเวลาที่ 2 ใช้เวลา 3 มิลลิวินาที



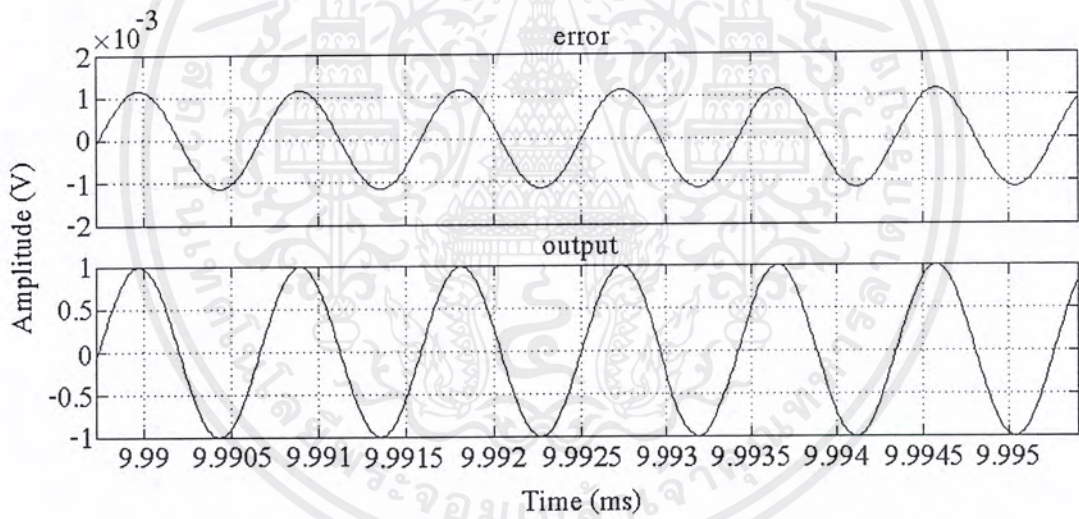
รูปที่ 4.7 สัญญาณเอาต์พุตและสัญญาณคลาดเคลื่อนเมื่อใช้เวลา 3 มิลลิวินาที

ช่วงเวลา 3 ใช้เวลา 5 มิลลิวินาที



รูปที่ 4.8 สัญญาณเอาต์พุตและสัญญาณคลาดเคลื่อนเมื่อใช้เวลา 5 มิลลิวินาที

ช่วงเวลา 4 ใช้เวลา 10 มิลลิวินาที

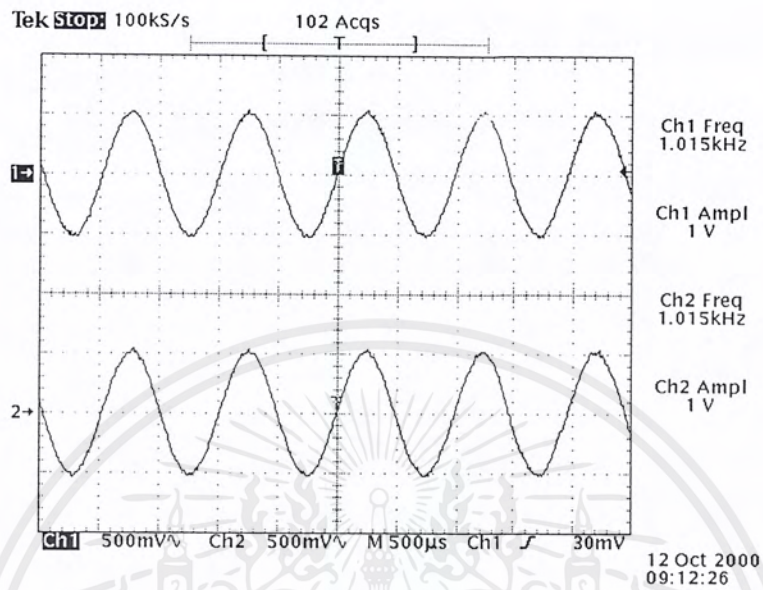


รูปที่ 4.9 สัญญาณเอาต์พุตและสัญญาณคลาดเคลื่อนเมื่อเวลาใช้เวลา 10 มิลลิวินาที

ที่เวลาเพิ่มมากขึ้น จะพบว่าขนาดของสัญญาณจะเปลี่ยนไป และที่น่าสังเกตอีกประการคือ ขนาดของสัญญาณคลาดเคลื่อนจะลดลงตามเวลาเช่นกัน สัญญาณเอาต์พุตจะมีขนาดของสัญญาณและเฟสเหมือนกับสัญญาณที่ต้องการ

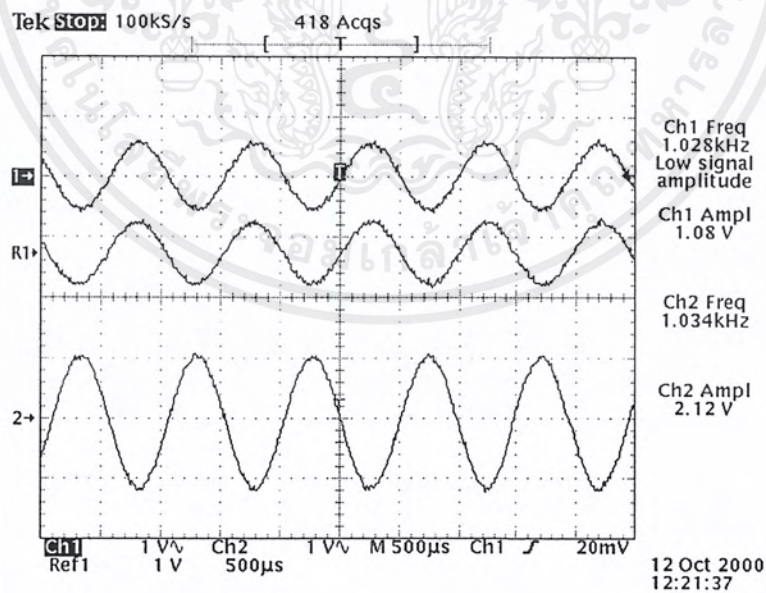
จากวงจรที่ได้ทำการออกแบบและทำการวัดเพื่อตรวจสอบการทำงาน ได้ผลดังนี้

วงจรบัฟเฟอร์



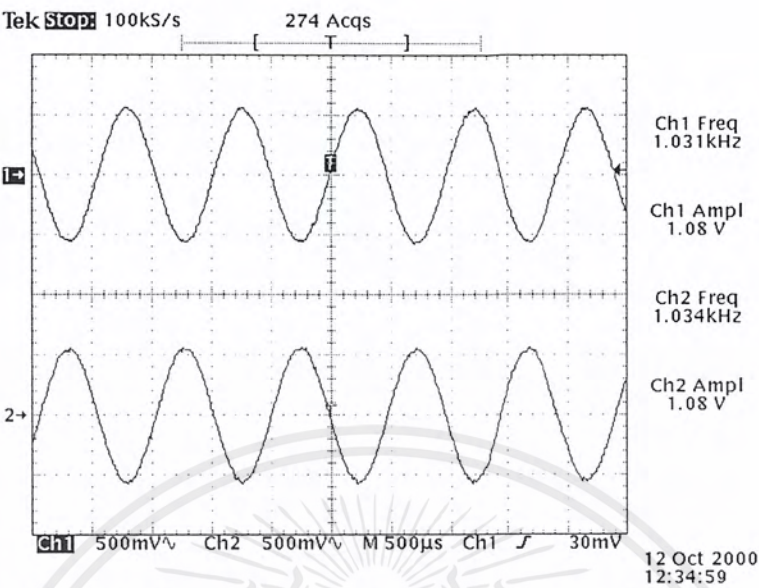
รูปที่ 4.10 ช่องสัญญาณที่ 1 แสดงสัญญาณอินพุตของวงจรบัฟเฟอร์
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตของวงจรบัฟเฟอร์

วงจรบวก



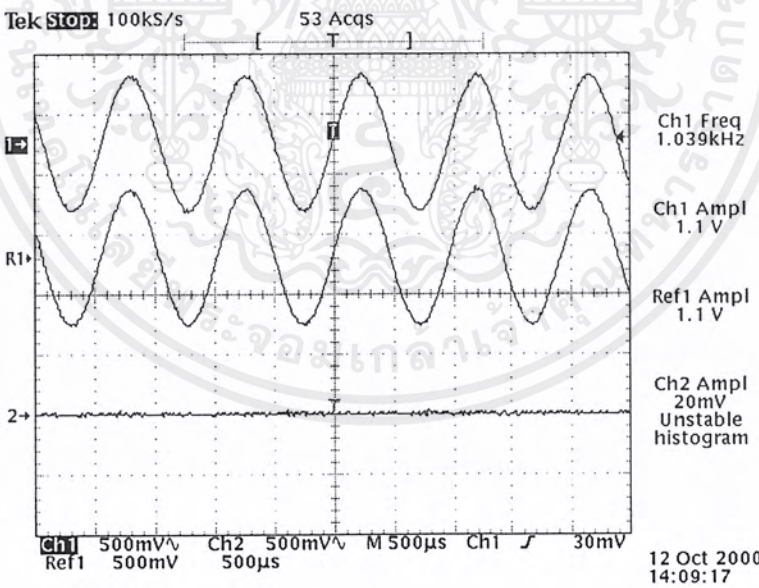
รูปที่ 4.11 ช่องสัญญาณที่ 1 และช่องสัญญาณอ้างอิง 1 แสดงสัญญาณอินพุตของวงจรบวก
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตของวงจรบวก

- วงจรกลับเฟส



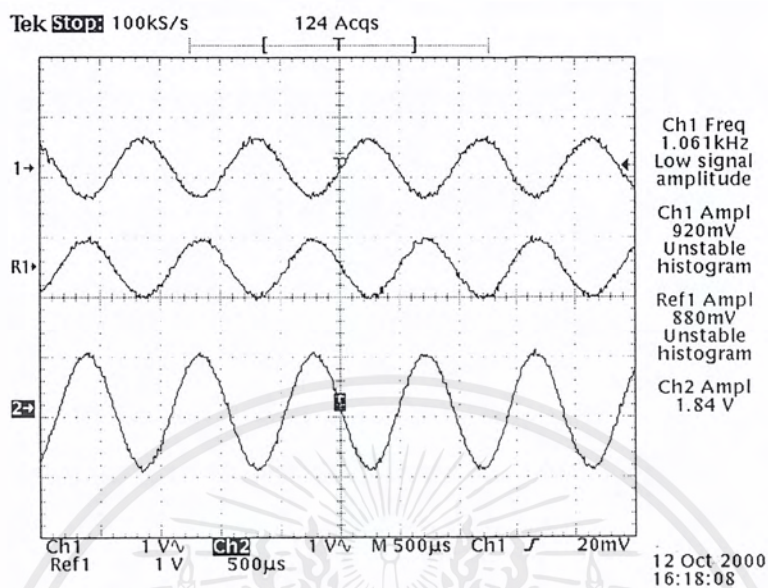
รูปที่ 4.12 ช่องสัญญาณที่ 1 แสดงสัญญาณอินพุตของวงจรกลับเฟส
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตของวงจรกลับเฟส

วงจรลบ (เมื่อสัญญาณมีขนาดเท่ากันและเฟสตรงกัน)



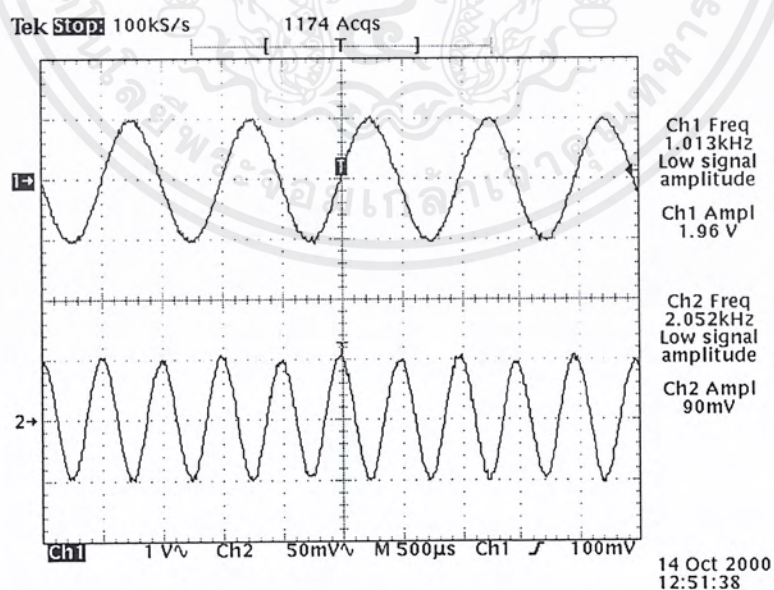
รูปที่ 4.13 ช่องสัญญาณที่ 1 และช่องสัญญาณอ้างอิง 1 แสดงสัญญาณอินพุตของวงจรลบ
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตของวงจรลบ

วงจรลบ (เมื่อสัญญาณมีขนาดเท่ากัน แต่เฟสตรงกันข้าม)



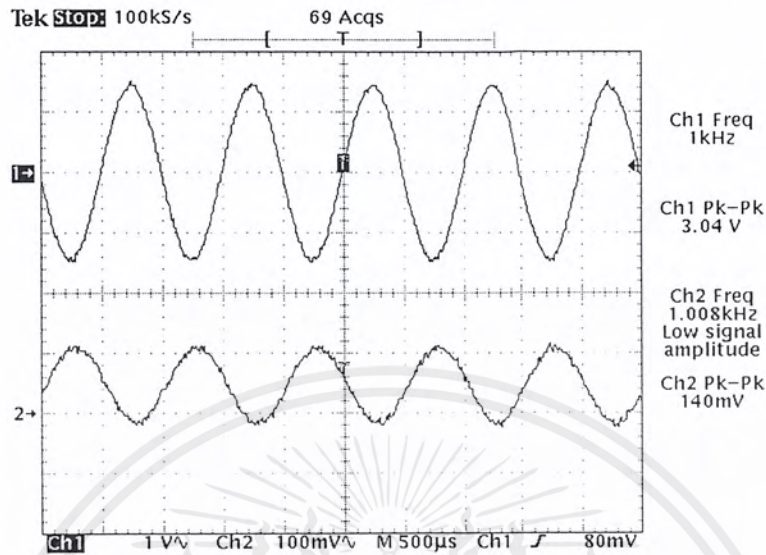
รูปที่ 4.14 ช่องสัญญาณที่ 1 และช่องสัญญาณอ้างอิง 1 แสดงสัญญาณอินพุตของวงจรลบ
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตของวงจรลบ

วงจรคูณ



รูปที่ 4.15 ช่องสัญญาณที่ 1 แสดงสัญญาณอินพุตของวงจรคูณ
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตของวงจรคูณ

วงจรอินทิเกรต



รูปที่ 4.16 ช่องสัญญาณที่ 1 แสดงสัญญาณอินพุตของวงจรอินทิเกรต
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตของวงจรอินทิเกรต

จากการออกแบบวงจรส่วนประมวลสัญญาณที่รับสัญญาณอินพุตจำนวน 4 อินพุตเข้ามา แล้วทำการปรับแต่งสัญญาณแต่ละตัวให้มีขนาดและเฟสตรงกัน เพื่อจะนำทุกสัญญาณมารวมกันเป็นสัญญาณเอาต์พุตที่มีขนาดและเฟสเหมือนกันสัญญาณที่ต้องการซึ่งป้อนให้กับวงจรไว้เป็นสัญญาณอ้างอิง

สำหรับการออกแบบการทดลองจะแบ่งการทดลองออกเป็น 2 ขั้นตอนด้วยกันคือ

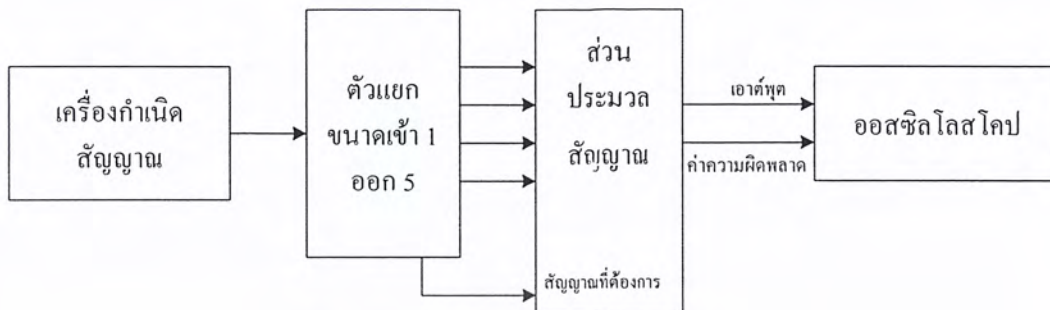
1. การปรับเทียบ (Calibrate) การทำงานของส่วนประมวลสัญญาณที่อยู่ในสถานะที่ไม่มีสัญญาณรบกวนใดๆ
2. การทดลองกับระบบสายอากาศทั้งระบบ

การปรับเทียบ

อุปกรณ์

- | | |
|-----------------------------------|-----------|
| 1. ออสซิลโลสโคป | 1 เครื่อง |
| 2. เครื่องกำเนิดสัญญาณ | 1 เครื่อง |
| 3. ตัวแยกกำลังงานขนาดเข้า 1 ออก 5 | 1 ตัว |
| 4. สายส่งสัญญาณ | 8 เส้น |
| 5. วงจรส่วนประมวลสัญญาณ | |

การจัดวางอุปกรณ์



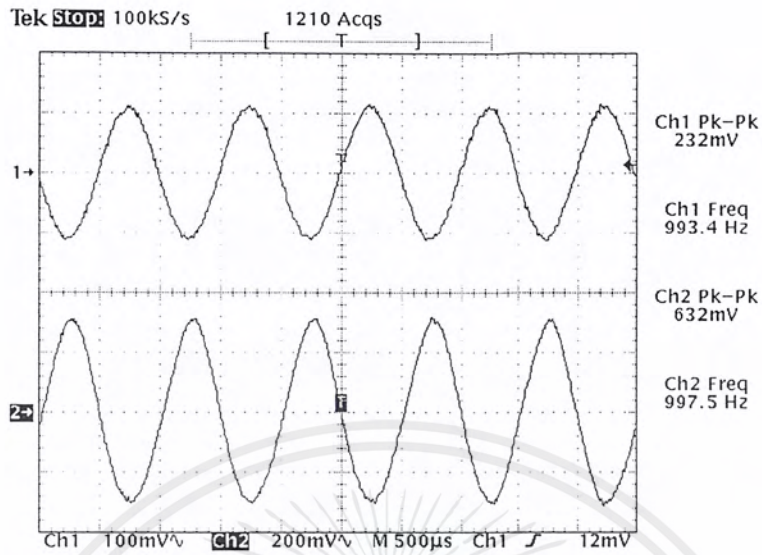
รูปที่ 4.17 การจัดวางอุปกรณ์ในการเปรียบเทียบ

1. นำสายส่งสัญญาณต่อจากเครื่องกำเนิดสัญญาณเข้าตัวแยกกำลังงาน เพื่อให้ได้สัญญาณที่มีขนาดและความถี่เท่ากันจำนวน 5 ตัว
2. นำสายส่งสัญญาณเชื่อมเอาต์พุตจากตัวแยกทั้ง 5 เอาต์พุตไปเชื่อมต่อเข้ากับส่วนประมวลสัญญาณ โดยให้ 4 ตัวแรกเป็นสัญญาณอินพุตของส่วนประมวลสัญญาณ และที่เหลืออีก 1 ตัวเป็นสัญญาณอ้างอิง
3. หลังจากที่เชื่อมต่อเสร็จแล้ว นำสายส่งสัญญาณเชื่อมเอาต์พุตจากส่วนประมวลสัญญาณและค่าความผิดพลาดไปแสดงผลในช่องสัญญาณที่ 1 และช่องสัญญาณที่ 2 ของออสซิลโลสโคป ตามลำดับ

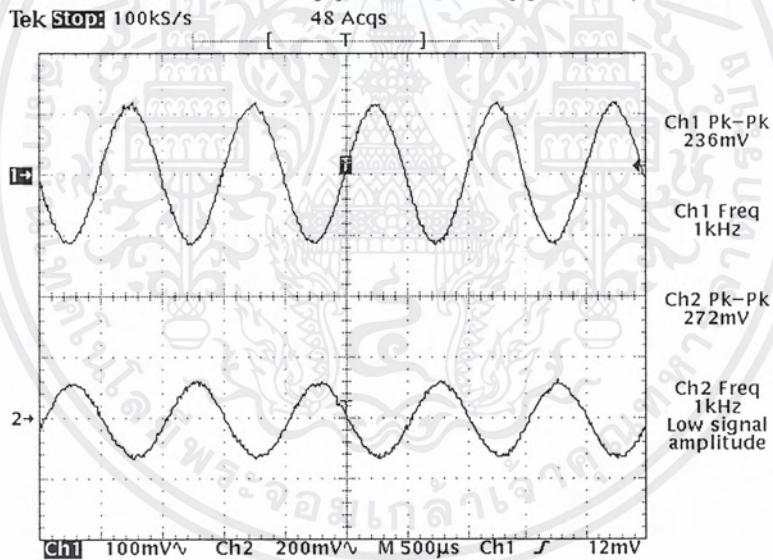
การทดลอง

1. กำหนดสัญญาณที่เครื่องกำเนิดสัญญาณให้มีความถี่ 1 กิโลเฮิร์ตซ์ ขนาด 500 มิลลิโวลต์
2. ดูสัญญาณเอาต์พุตที่แสดงในออสซิลโลสโคปแล้วสังเกตว่ารูปสัญญาณที่ได้มีลักษณะเหมือนกับสัญญาณอ้างอิงหรือไม่ ถ้ารูปสัญญาณที่ได้เหมือนกับสัญญาณอ้างอิงแสดงว่า วงจรส่วนประมวลสัญญาณพร้อมที่จะทำงานแล้ว แต่ถ้ารูปสัญญาณที่ได้ไม่เหมือนกับสัญญาณอ้างอิง ต้องกลับไปปรับค่าความต้านทานปรับค่าได้ขนาด 1 กิโลโอห์ม ในวงจรส่วนประมวลสัญญาณ จนได้รูปสัญญาณเอาต์พุตเป็นลักษณะที่คล้ายกับสัญญาณอ้างอิงมากที่สุด
3. วัดค่าความคลาดเคลื่อนในช่องสัญญาณที่สองแล้วบันทึกค่า

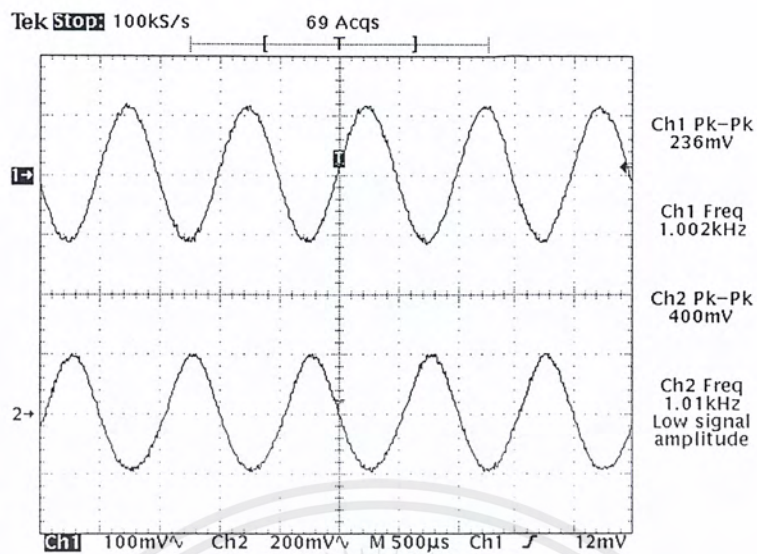
ผลการทดลอง



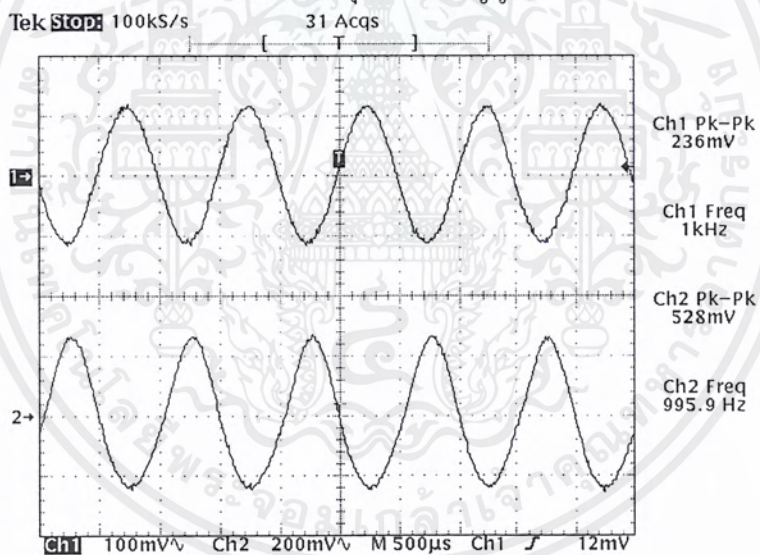
รูปที่ 4.18 ช่องสัญญาณที่ 1 แสดงสัญญาณที่ต้องการ
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตเมื่อผ่านวงจร



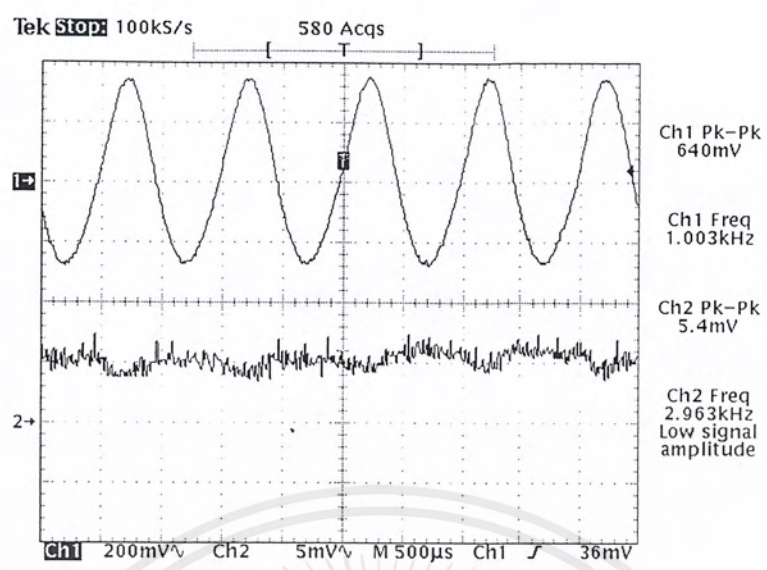
รูปที่ 4.19 ช่องสัญญาณที่ 1 แสดงสัญญาณที่ต้องการ
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตเมื่อผ่านวงจร
โดยใส่อินพุต 1 ช่องสัญญาณ



รูปที่ 4.20 ช่องสัญญาณที่ 1 แสดงสัญญาณที่ต้องการ
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตเมื่อผ่านวงจร
โดยใส่อินพุต 2 ช่องสัญญาณ



รูปที่ 4.21 ช่องสัญญาณที่ 1 แสดงสัญญาณที่ต้องการ
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตเมื่อผ่านวงจร
โดยใส่อินพุต 3 ช่องสัญญาณ

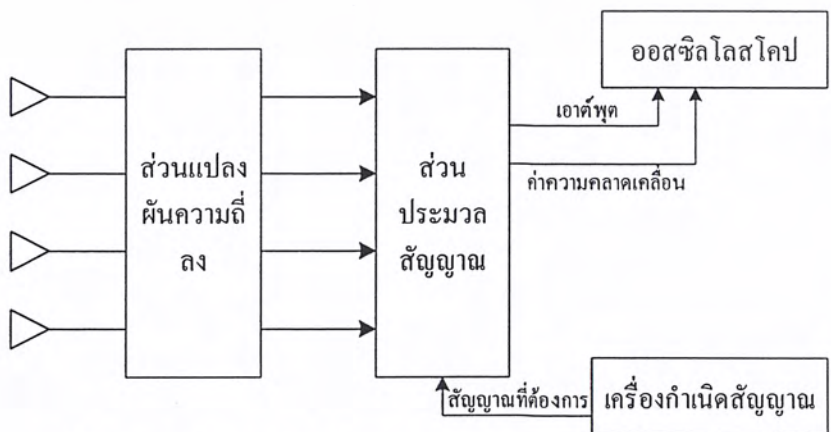


รูปที่ 4.22 ช่องสัญญาณที่ 1 แสดงสัญญาณเอาต์พุต
ช่องสัญญาณที่ 2 แสดงสัญญาณคลาดเคลื่อน

การทดลองระบบสายอากาศปรับตัวได้
อุปกรณ์

- | | |
|-------------------------|-----------|
| 1. ออสซิลโลสโคป | 1 เครื่อง |
| 2. เครื่องกำเนิดสัญญาณ | 1 เครื่อง |
| 3. สายส่งสัญญาณ | 7 เส้น |
| 4. สายอากาศแถวลำดับ | |
| 5. ส่วนแปลงผันความถี่ลง | |
| 6. ส่วนประมวลสัญญาณ | |

การจัดวางอุปกรณ์

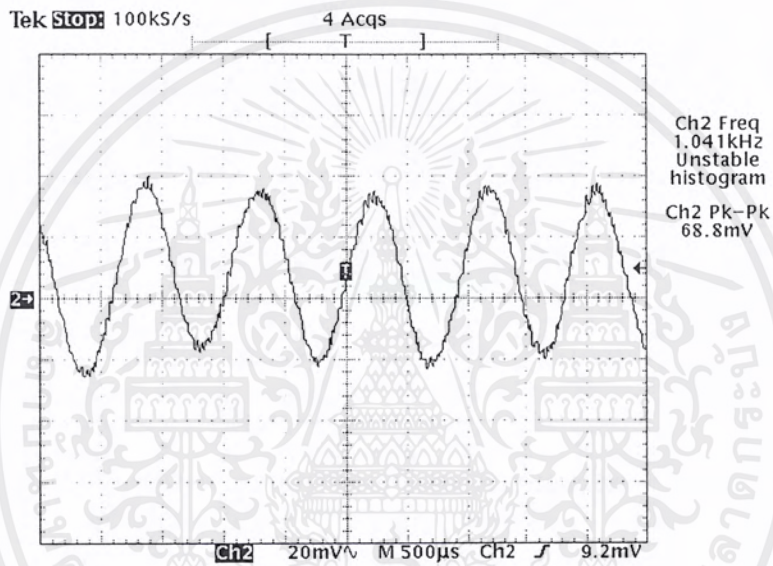


รูปที่ 4.23 การจัดวางอุปกรณ์การทดลองระบบสายอากาศปรับตัวได้

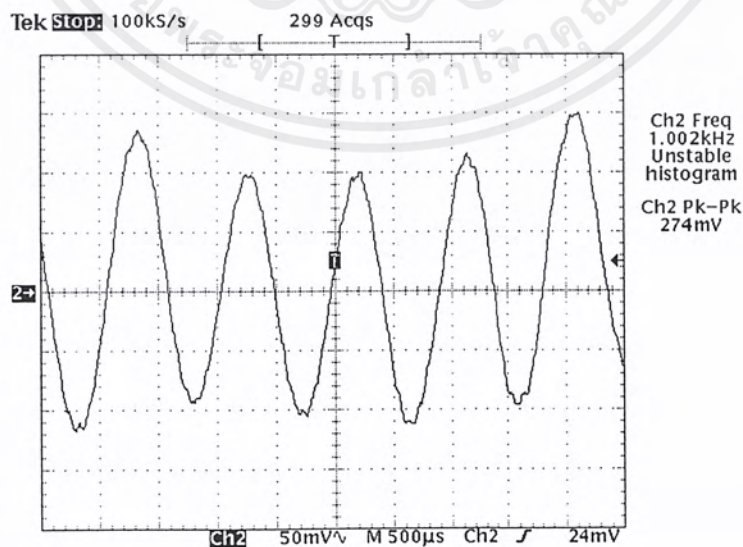
การทดลอง

1. รับสัญญาณ IF จากสายอากาศเข้าสู่ส่วนแปลงผันความถี่ลง เอาต์พุตจากส่วนแปลงผันความถี่ลงจะเป็นอินพุตของส่วนประมวลสัญญาณ
2. ป้อนสัญญาณอ้างอิงโดยให้สายส่งสัญญาณรับสัญญาณจากเครื่องกำเนิดสัญญาณ
3. นำสายส่งสัญญาณเชื่อมต่อจากส่วนเอาต์พุตและค่าความคาดเคลื่อนของส่วนประมวลสัญญาณไปแสดงผลยังออสซิลโลสโคปทางช่องสัญญาณที่ 1 และ 2 ตามลำดับ

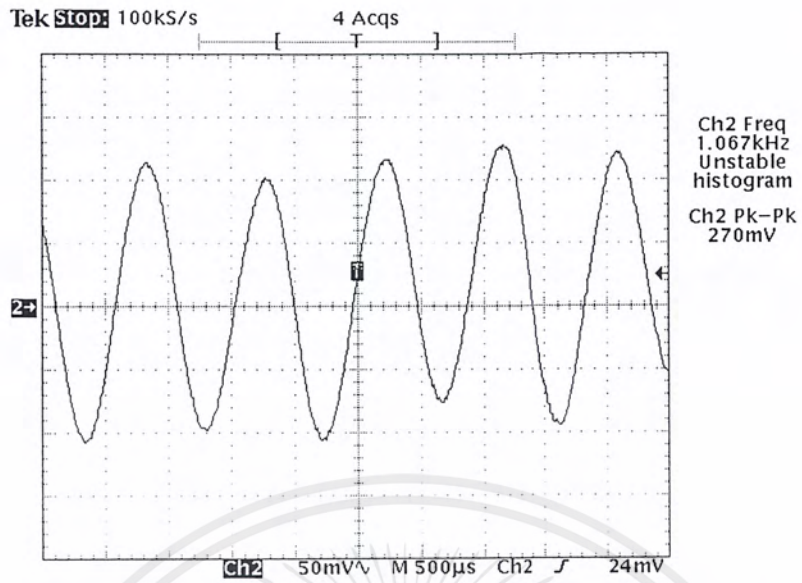
ผลการทดลอง



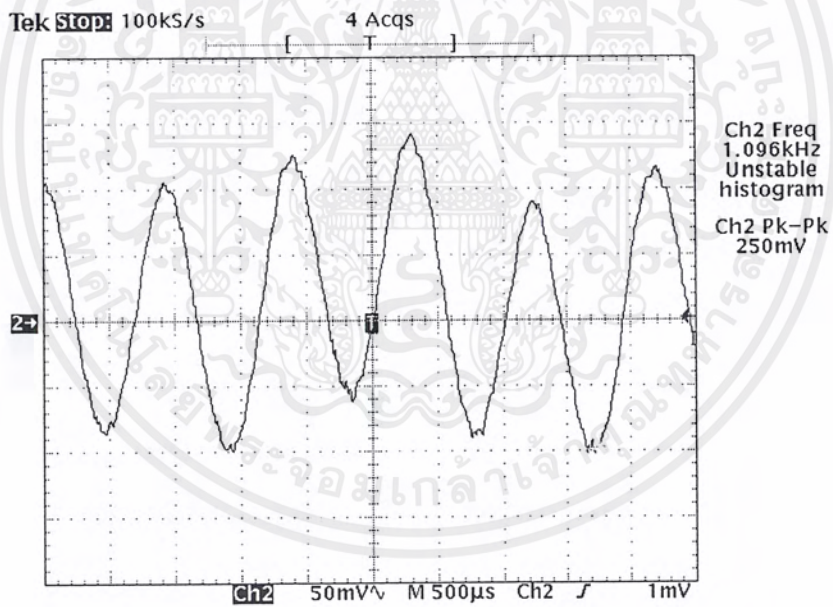
รูปที่ 4.24 สัญญาณอินพุตที่ 1 ของส่วนประมวลสัญญาณ



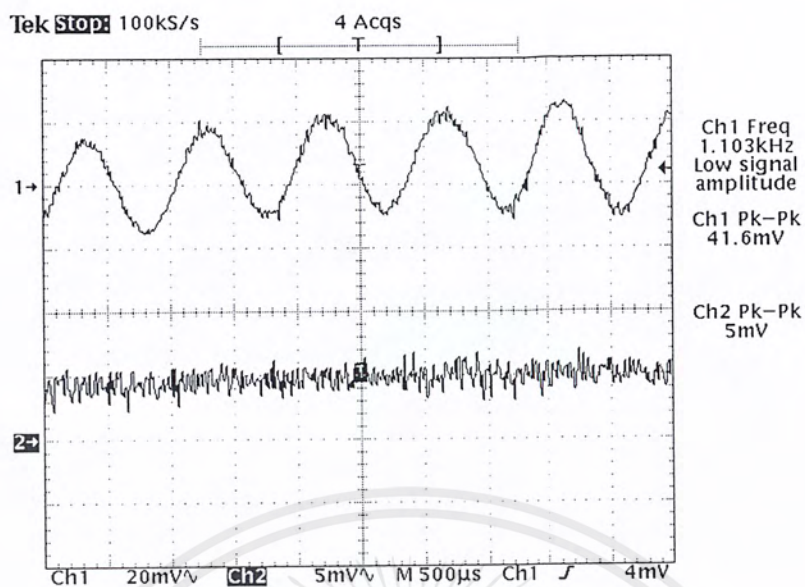
รูปที่ 4.25 สัญญาณอินพุตที่ 2 ของส่วนประมวลสัญญาณ



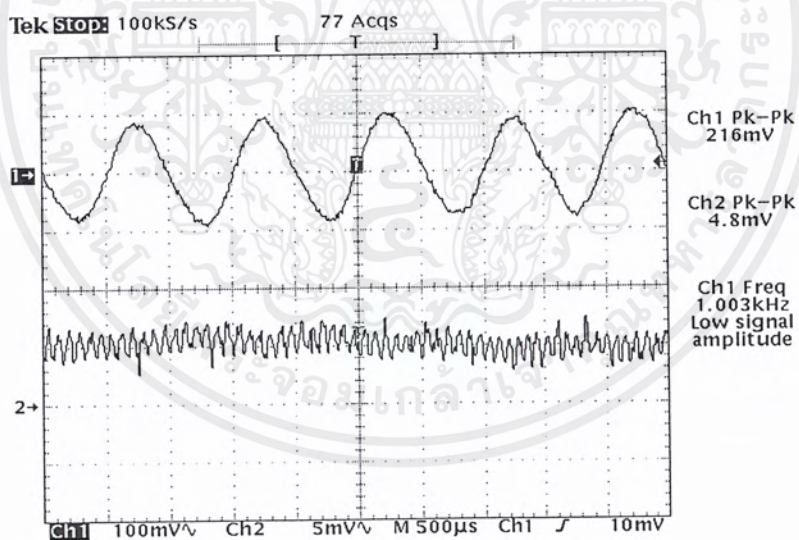
รูปที่ 4.26 สัญญาณอินพุตที่ 3 ของส่วนประมวลสัญญาณ



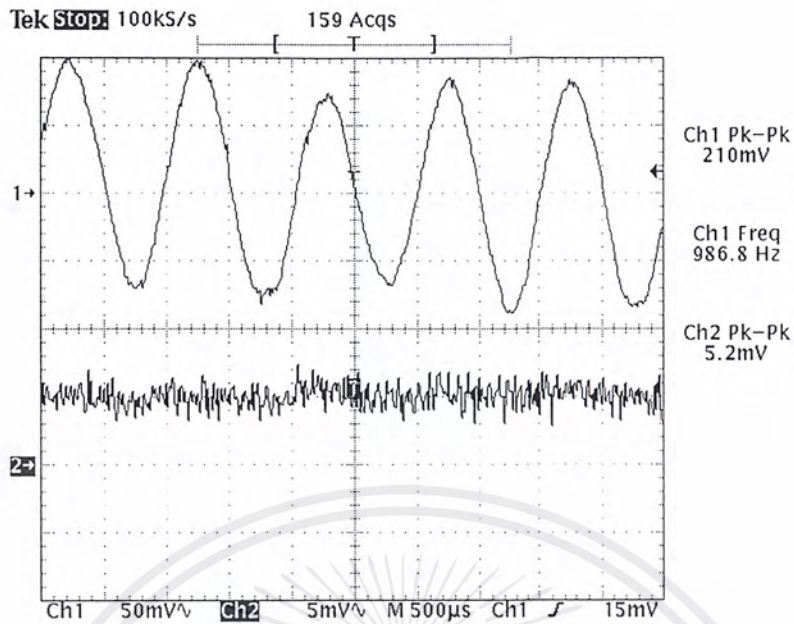
รูปที่ 4.27 สัญญาณอินพุตที่ 4 ของส่วนประมวลสัญญาณ



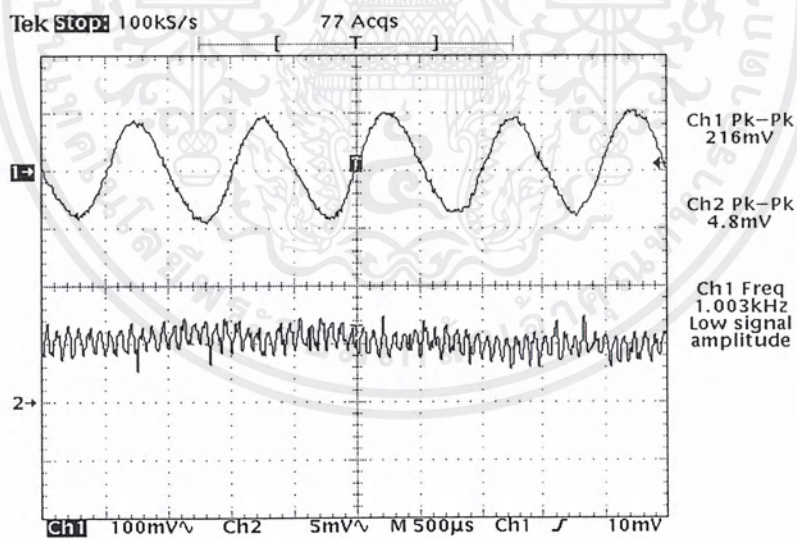
รูปที่ 4.28 สายอากาศส่งทำมุม 0 องศา กับสายอากาศรับ
ช่องสัญญาณที่ 1 แสดงสัญญาณเอาต์พุตเมื่อเริ่มต้น
ช่องสัญญาณที่ 2 แสดงสัญญาณค่าความคลาดเคลื่อน



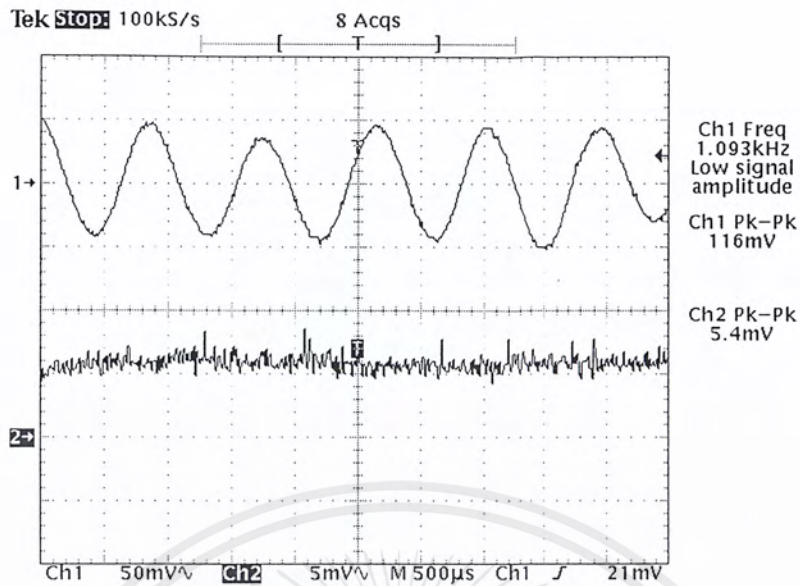
รูปที่ 4.29 สายอากาศส่งทำมุม 0 องศา กับสายอากาศรับ
ช่องสัญญาณที่ 1 แสดงสัญญาณเอาต์พุตเมื่อผ่านการประมวลผลสัญญาณ
ช่องสัญญาณที่ 2 แสดงสัญญาณค่าความคลาดเคลื่อน



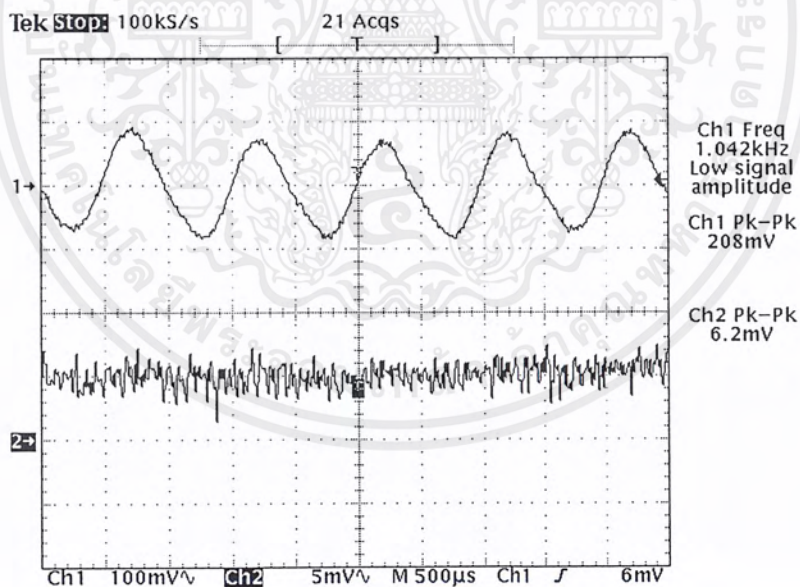
รูปที่ 4.30 สายอากาศส่งทำมุม 30 องศา กับสายอากาศรับ
ช่องสัญญาณที่ 1 แสดงสัญญาณเอาต์พุตเมื่อเริ่มต้น
ช่องสัญญาณที่ 2 แสดงสัญญาณค่าความคาดเคลื่อน



รูปที่ 4.31 สายอากาศส่งทำมุม 30 องศา กับสายอากาศรับ
ช่องสัญญาณที่ 1 แสดงสัญญาณเอาต์พุตเมื่อผ่านการประมวลผลสัญญาณ
ช่องสัญญาณที่ 2 แสดงสัญญาณค่าความคาดเคลื่อน



รูปที่ 4.32 สายอากาศส่งทำมุม -30 องศา กับสายอากาศรับ
ช่องสัญญาณที่ 1 แสดงสัญญาณเอาต์พุตเมื่อเริ่มต้น
ช่องสัญญาณที่ 2 แสดงสัญญาณค่าความคลาดเคลื่อน



รูปที่ 4.33 สายอากาศส่งทำมุม -30 องศา กับสายอากาศรับ
ช่องสัญญาณที่ 1 แสดงสัญญาณเอาต์พุตเมื่อผ่านการประมวลผลสัญญาณ
ช่องสัญญาณที่ 2 แสดงสัญญาณค่าความคลาดเคลื่อน

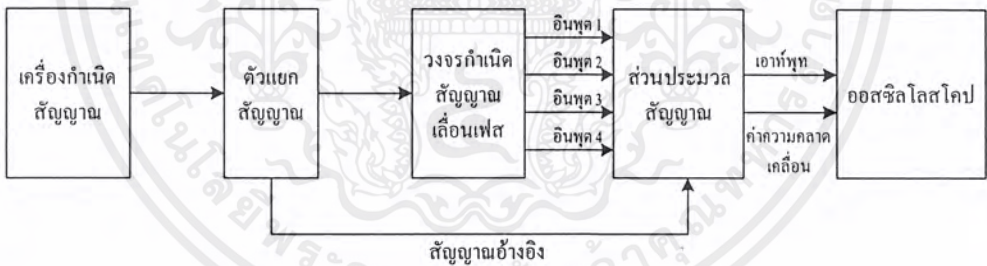
การทดลองการทำงานของส่วนประมวลสัญญาณแบบอนาลอกโดยวงจรสมมุติสัญญาณ

จากการออกแบบการทดลองทั้งระบบของสายอากาศแบบปรับตัวได้ มีปัญหาในการไม่เสถียรของสายอากาศรับทำให้ไม่สามารถเห็นการทำงานของส่วนประมวลผลสัญญาณได้อย่างชัดเจน ดังนั้นจึงได้ทำการออกแบบการทดลองใหม่เพื่อให้สามารถเห็นกระบวนการทำงานได้อย่างชัดเจน เนื่องจากว่าในโครงการนี้สัญญาณที่จะนำมาประมวลผลเป็นสัญญาณเบสแบนด์ความถี่ 1 กิโลเฮิร์ตซ์ แอมพลิจูด 500 มิลลิโวลต์ ซึ่งเป็นสัญญาณที่ผ่านการคิมอดูเลตโดย ส่วนแปลงผันความถี่ลงมาแล้ว ดังนั้นในการออกแบบการทดลองใหม่จึงได้ทำการสมมุติสัญญาณขึ้น 4 สัญญาณ โดยทั้งสี่สัญญาณนี้จะมีเฟสและแอมพลิจูดที่แตกต่างกัน แต่มีความถี่เดียวกัน ซึ่งสัญญาณที่สร้างขึ้นนี้ยึดตามความน่าจะเป็นของการเดินทางของสัญญาณจากด้านส่งมายังด้านรับ ซึ่งจะต้องมีการสะท้อนหรือเกิดการเลี้ยวเบนตามคุณสมบัติของคลื่น ดังนั้นสัญญาณที่เข้ามายังสายอากาศรับจึงมีเฟสที่แตกต่างกัน และแอมพลิจูดต่ำ ดังสัญญาณที่ได้ทำการสมมุติขึ้น

อุปกรณ์

- 1. เครื่องกำเนิดสัญญาณ 1 เครื่อง
- 2. ออสซิลโลสโคป 1 เครื่อง
- 3. ส่วนประมวลผลสัญญาณ
- 4. วงจรสมมุติสัญญาณอินพุต
- 5. ตัวแยกกำลังงาน

การจัดวางอุปกรณ์



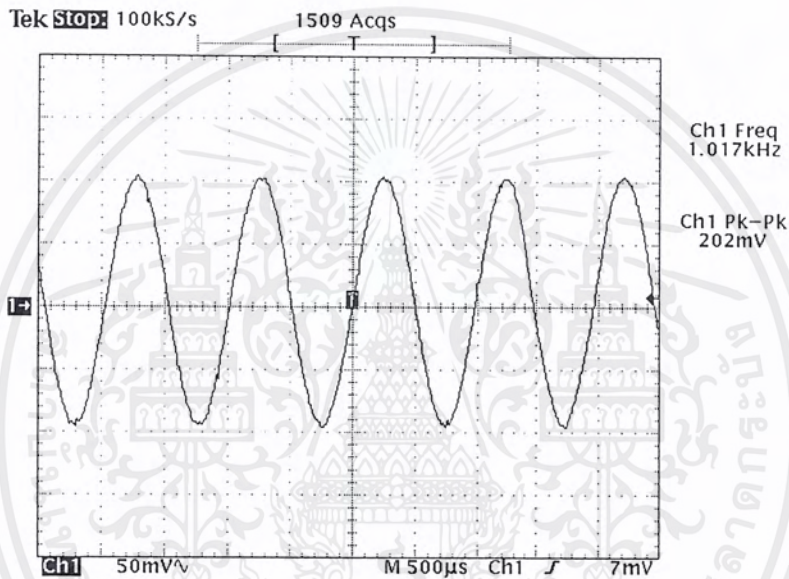
รูปที่ 4.34 การจัดวางอุปกรณ์การทดลองการทำงานของส่วนประมวลสัญญาณแบบอนาลอก โดยการใช้สมมุติสัญญาณอินพุต

การทดลอง

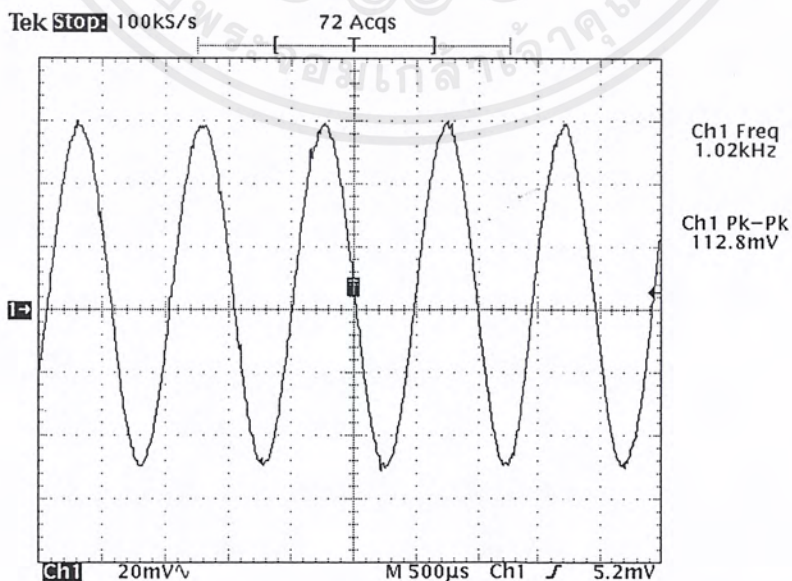
- 1. จัดวางอุปกรณ์ดังภาพ
- 2. ป้อนสัญญาณจากเครื่องกำเนิดสัญญาณ ให้มีความถี่ 1 กิโลเฮิร์ตซ์ และมีแอมพลิจูด 500 มิลลิโวลต์ ให้กับตัวแยกแบบ เข้า 1 ออก 2
- 3. นำสัญญาณ เอาต์พุตจากตัวแยกทั้งสองสัญญาณไปป้อนให้กับ วงจร สมมุติสัญญาณ และ นำไปป้อนเป็นสัญญาณอ้างอิงให้กับส่วนประมวลผลสัญญาณ ตามลำดับ
- 4. เมื่อเริ่มป้อนสัญญาณให้กับส่วนประมวลผลให้ทำการวัดสัญญาณเอาต์พุต
- 5. ทำการบันทึกผลโดยบันทึกค่าแอมพลิจูดของเอาต์พุต เทียบกับค่า ความคลาดเคลื่อน

6. สังเกตการเปลี่ยนแปลงของรูปสัญญาณและขนาดของสัญญาณเมื่อเวลาผ่านไป และทำการบันทึกเวลาที่ใช้ในการปรับเปลี่ยนสัญญาณตั้งแต่เริ่มป้อนสัญญาณให้กับส่วนประมวลผล
7. ทำการบันทึกผลเปรียบเทียบระหว่างสัญญาณเอาต์พุตที่ผ่านการประมวลผลแล้วกับสัญญาณอ้างอิง
8. ทำการบันทึกผลเปรียบเทียบระหว่างสัญญาณเอาต์พุตที่ผ่านการประมวลผลแล้วกับค่าความคลาดเคลื่อน

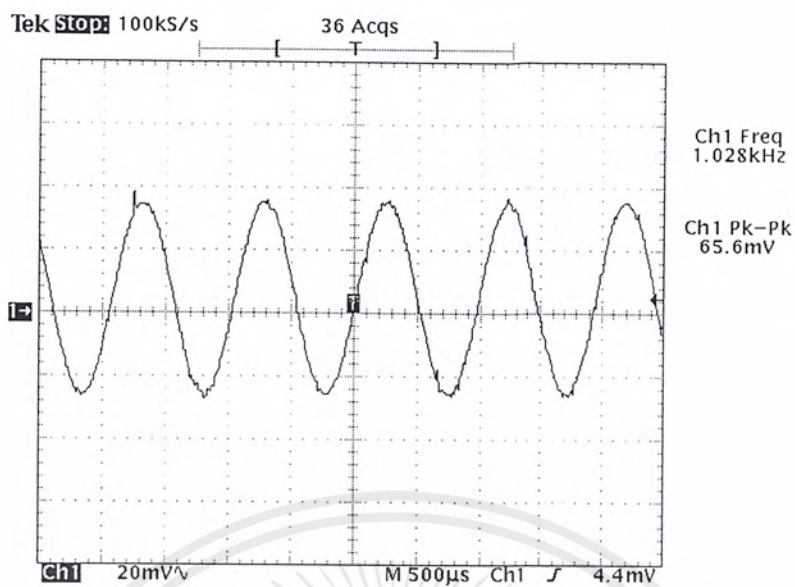
ผลการทดลอง



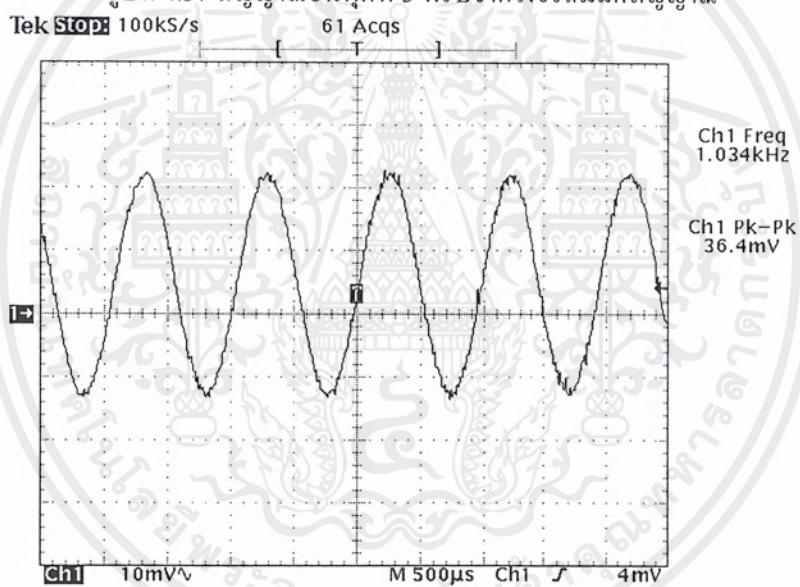
รูปที่ 4.35 สัญญาณอินพุตที่ 1 ที่รับจากวงจรสมมติสัญญาณ



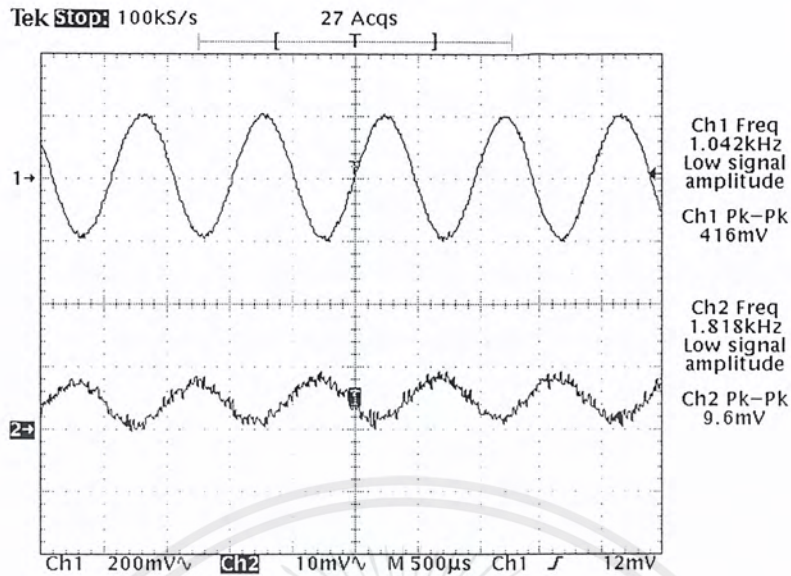
รูปที่ 4.36 สัญญาณอินพุตที่ 2 ที่รับจากวงจรสมมติสัญญาณ



รูปที่ 4.37 สัญญาณอินพุตที่ 3 ที่รับจากวงจรสมมติสัญญาณ

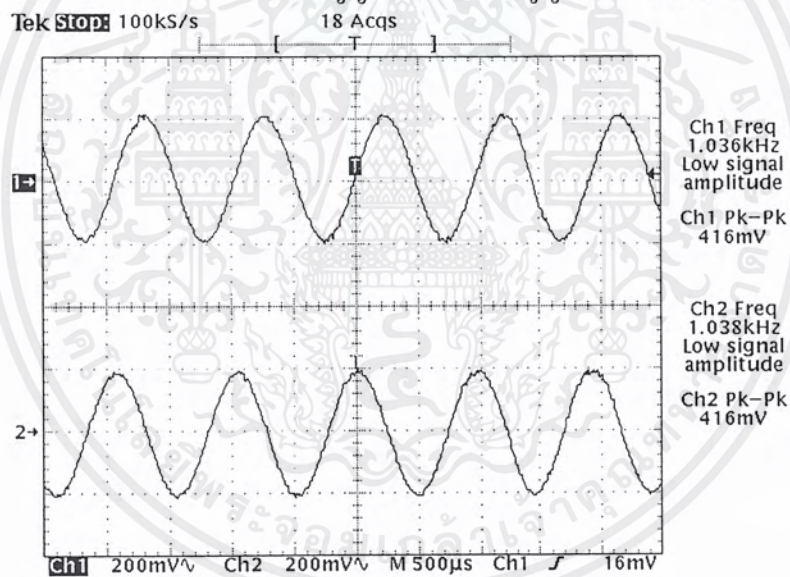


รูปที่ 4.38 สัญญาณอินพุตที่ 4 ที่รับจากวงจรสมมติสัญญาณ



รูปที่ 4.39 ช่องสัญญาณที่ 1 แสดงสัญญาณเอาต์พุต

ช่องสัญญาณที่ 2 แสดงสัญญาณค่าความคลาดเคลื่อน

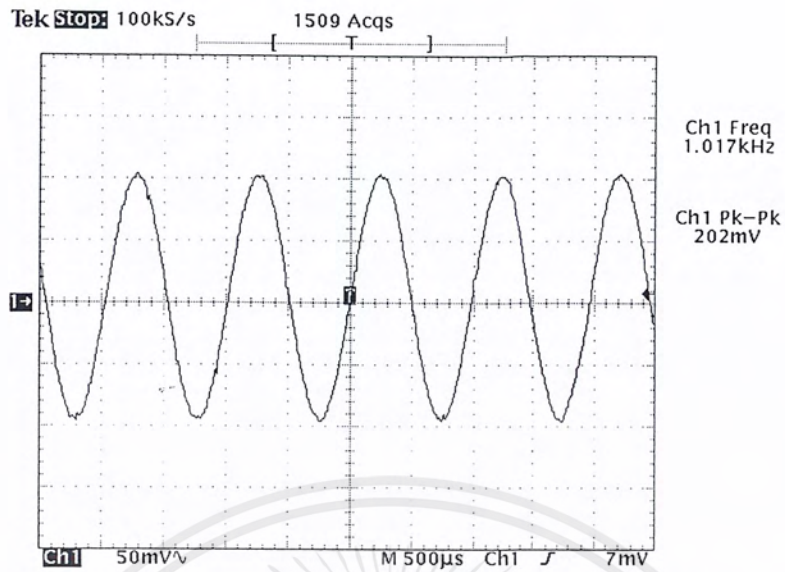


รูปที่ 4.40 ช่องสัญญาณที่ 1 แสดงสัญญาณอ้างอิง

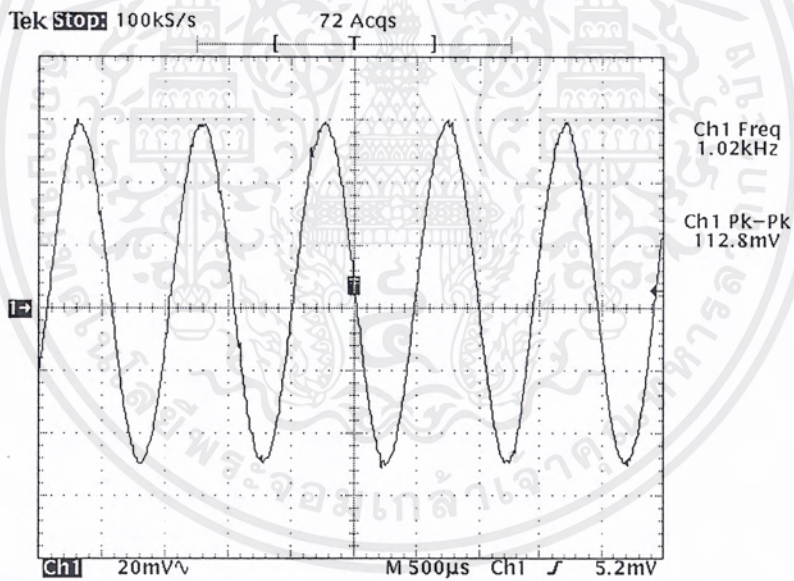
ช่องสัญญาณที่ 2 แสดงสัญญาณเอาต์พุตจากส่วนประมวลสัญญาณ

การทดลองส่วนแปลงสัญญาณที่รับจากสายอากาศเพื่อให้สามารถนำไปประมวลผลโดยคอมพิวเตอร์

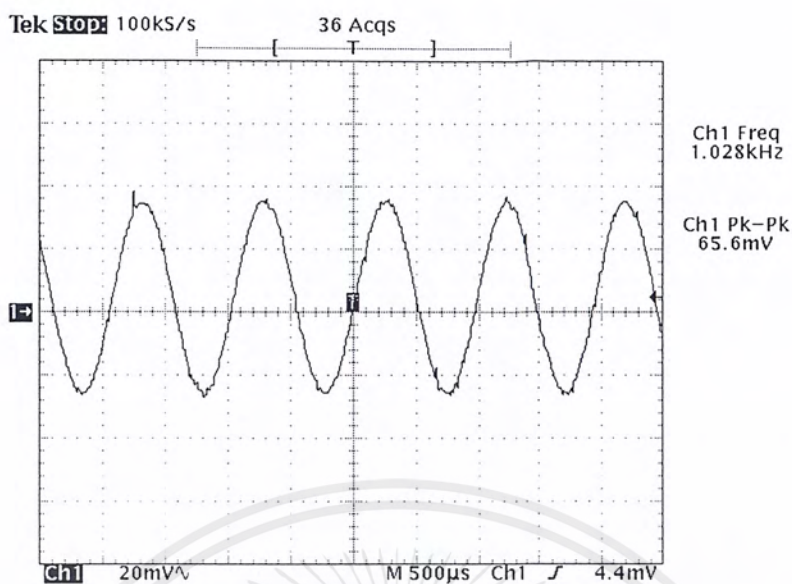
สำหรับการทดลองในส่วนนี้เราได้ทำการสมมุติสัญญาณอินพุตขึ้นมาเหมือนกับการทดลองในส่วนต่างๆ ที่ผ่านมา แต่สัญญาณที่อินพุตที่สมมุติขึ้นมานี้จะต้องมีคุณสมบัติใกล้เคียงกับสัญญาณที่รับเข้ามาจริงจากสายอากาศ จากเงื่อนไขข้อนี้เราได้ทำการออกแบบวงจรสมมุติสัญญาณขึ้นมาเพื่อทำการทดลองแล้ว ดังนั้นสัญญาณอินพุตที่ป้อนให้กับอุปกรณ์ส่วนนี้ประกอบด้วย 4 อินพุตดังนี้



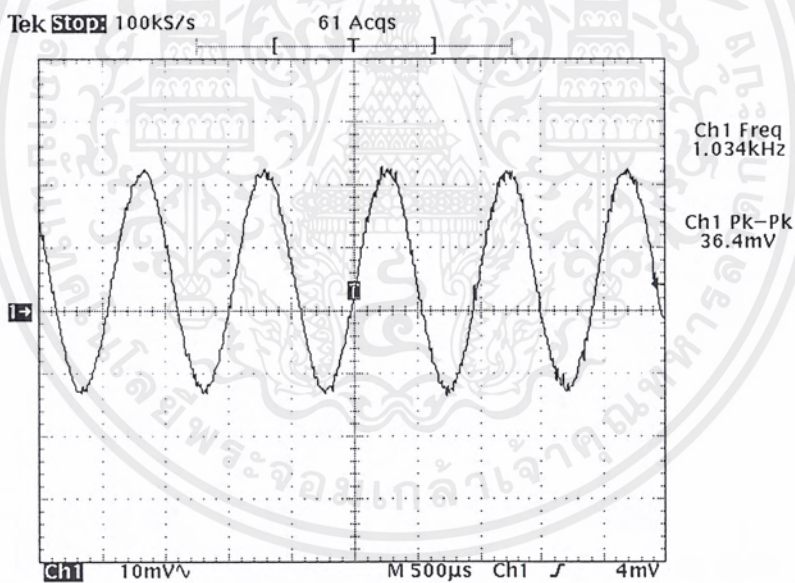
รูปที่ 4.41 สัญญาณอินพุตที่ 1 ที่ป้อนให้กับส่วนแปลงสัญญาณอนาลอกเป็นดิจิตอลตัวที่ 1



รูปที่ 4.42 สัญญาณอินพุตที่ 2 ที่ป้อนให้กับส่วนแปลงสัญญาณอนาลอกเป็นดิจิตอลตัวที่ 2

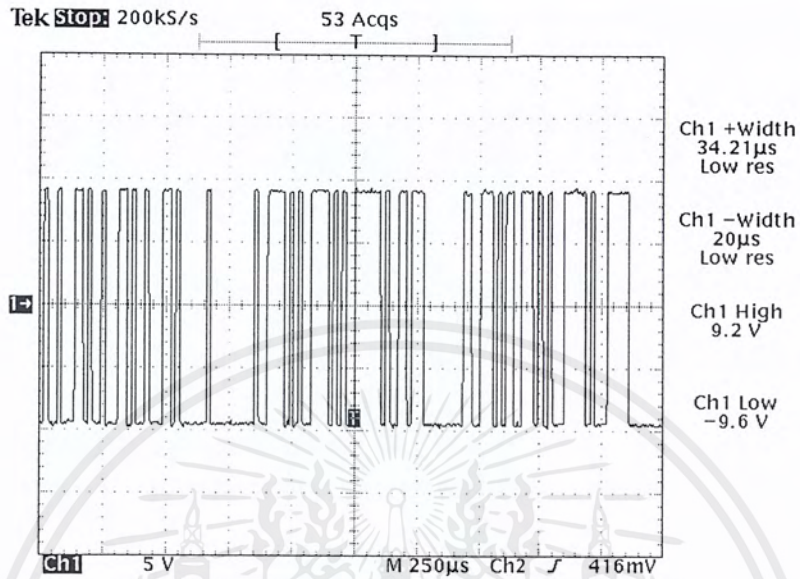


รูปที่ 4.43 สัญญาณอินพุตที่ 3 ที่ป้อนให้กับส่วนแปลงสัญญาณอนาลอกเป็นดิจิตอลตัวที่ 3

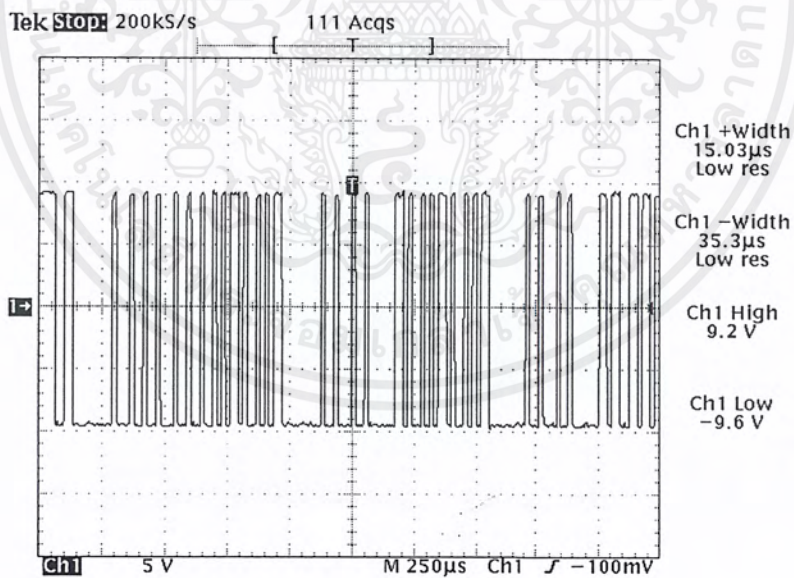


รูปที่ 4.44 สัญญาณอินพุตที่ 4 ที่ป้อนให้กับส่วนแปลงสัญญาณอนาลอกเป็นดิจิตอลตัวที่ 4

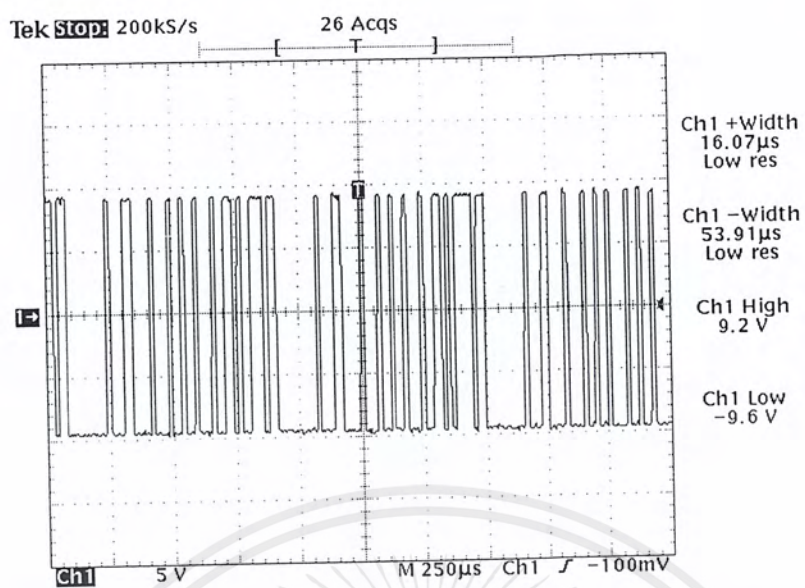
จากการป้อนสัญญาณอินพุตทั้งสี่ให้กับส่วนที่แปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัลทำให้สามารถวัดสัญญาณเอาต์พุตที่ขาของตัวไอซี MAX 232 ได้ดังนี้



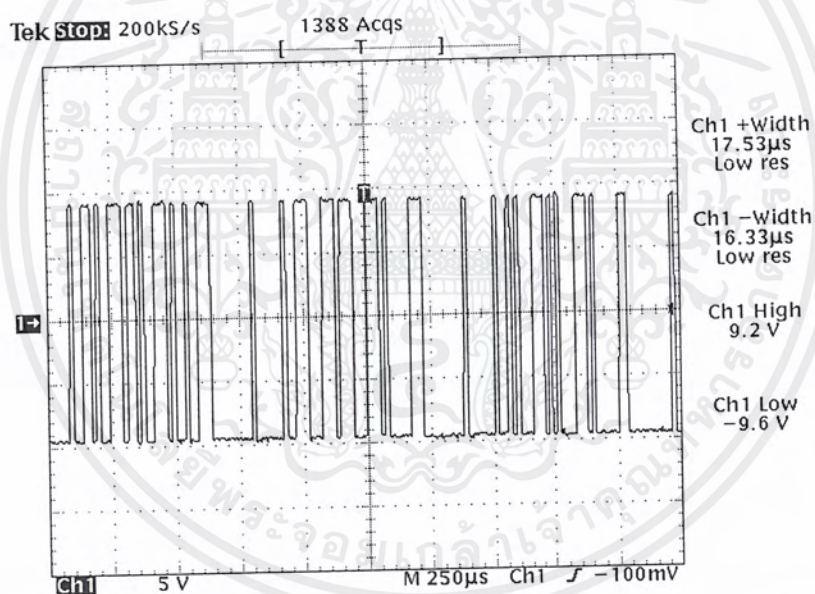
รูปที่ 4.45 สัญญาณเอาต์พุตที่ขาไอซี MAX 232 จากส่วนแปลงสัญญาณอนาลอกเป็นดิจิทัลตัวที่ 1



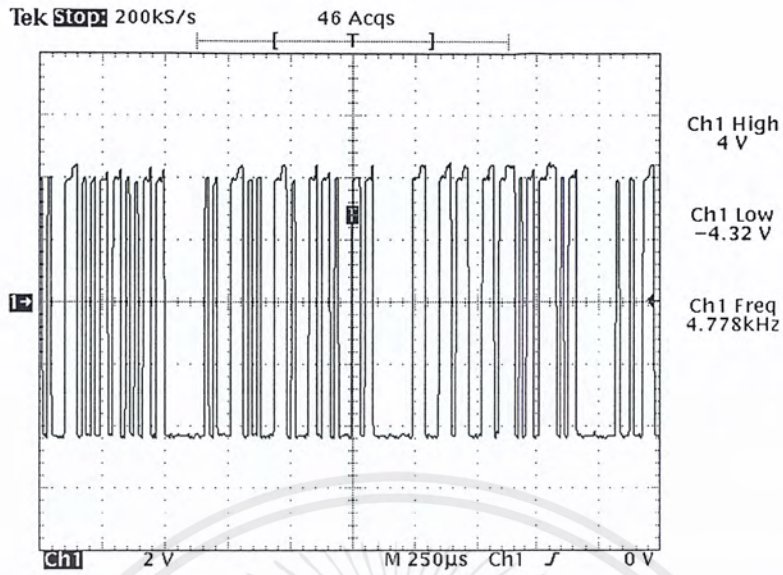
รูปที่ 4.46 สัญญาณเอาต์พุตที่ขาไอซี MAX 232 จากส่วนแปลงสัญญาณอนาลอกเป็นดิจิทัลตัวที่ 2



รูปที่ 4.47 สัญญาณเอาต์พุตที่ขาไอซี MAX 232 จากส่วนแปลงสัญญาณอนาลอกเป็นดิจิทัลตัวที่ 3



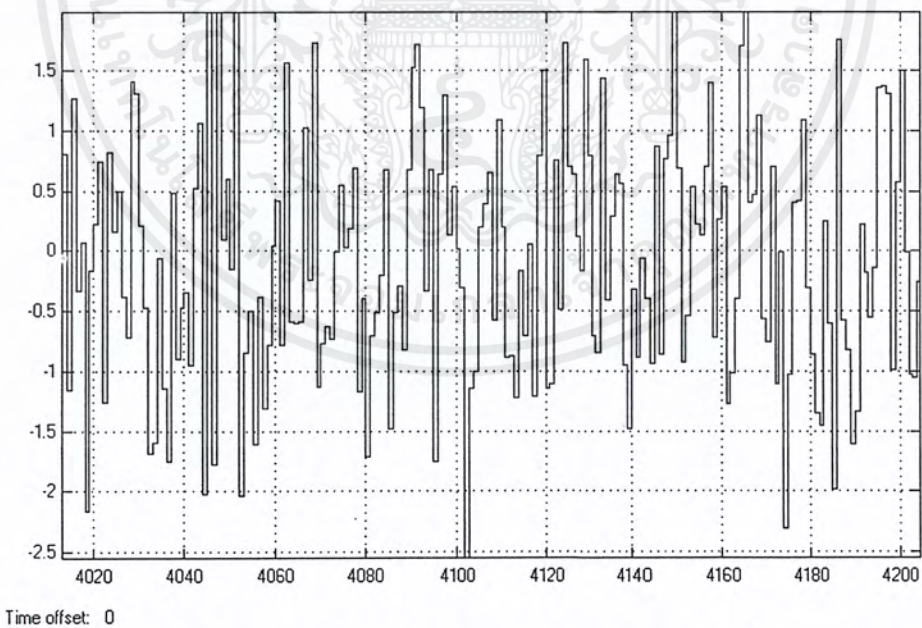
รูปที่ 4.48 สัญญาณเอาต์พุตที่ขาไอซี MAX 232 จากส่วนแปลงสัญญาณอนาลอกเป็นดิจิทัลตัวที่ 4



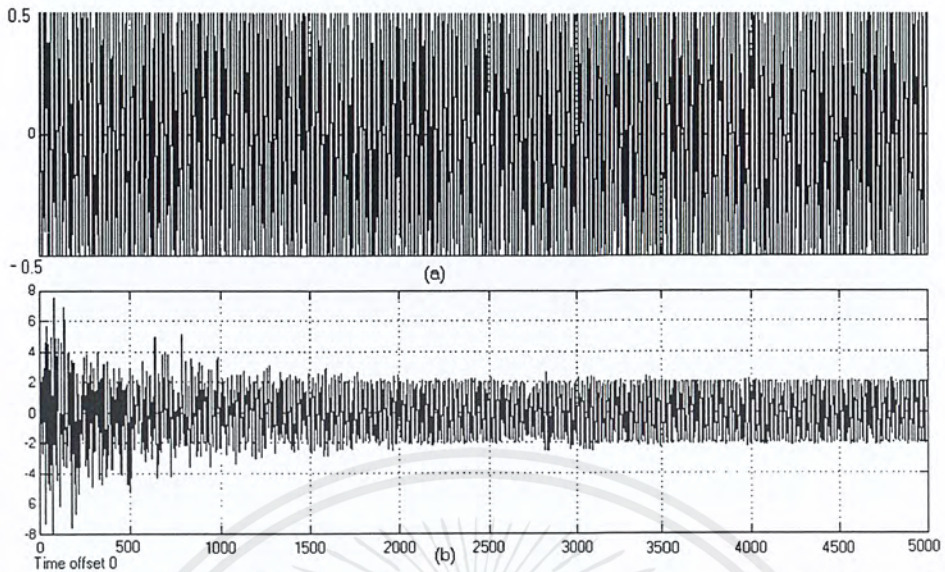
รูปที่ 4.49 สัญญาณเอาต์พุตที่ขาไอซี MAX 232 จากส่วนแปลงสัญญาณอนาล็อกเป็นดิจิทัลทั้ง 4 ตัวพร้อมกัน

การทดลองส่วนประมวลสัญญาณแบบดิจิทัลโดยการสมมติสัญญาณ

การทดลองในส่วนนี้ จะเป็นการทดลองส่วนประมวลสัญญาณแบบดิจิทัล โดยอาศัยการสมมติสัญญาณขึ้น ซึ่งสัญญาณที่สมมติขึ้นนี้จะมีลักษณะเป็นสัญญาณที่ต้องการ ร่วมกับสัญญาณรบกวน ส่งเข้าเป็นอินพุตของส่วนประมวลสัญญาณ



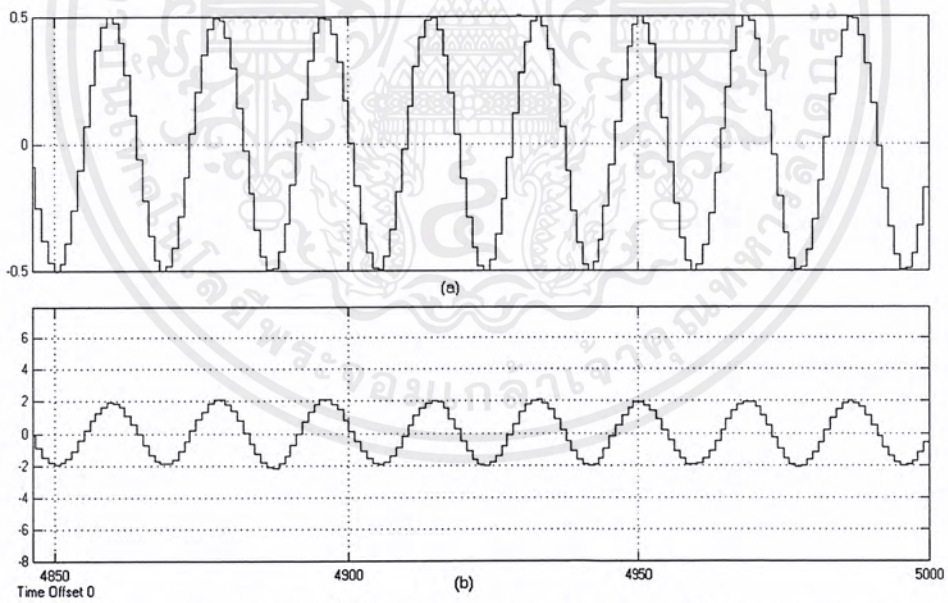
รูปที่ 4.50 สัญญาณอินพุตของส่วนประมวลสัญญาณแบบดิจิทัลที่สมมติขึ้น



รูปที่ 4.51 แสดงสัญญาณที่ได้จากการจำลองการทำงาน โดยแสดงช่วงเวลาดังแต่เริ่มต้นจนเสร็จสิ้น

(a) สัญญาณอ้างอิง

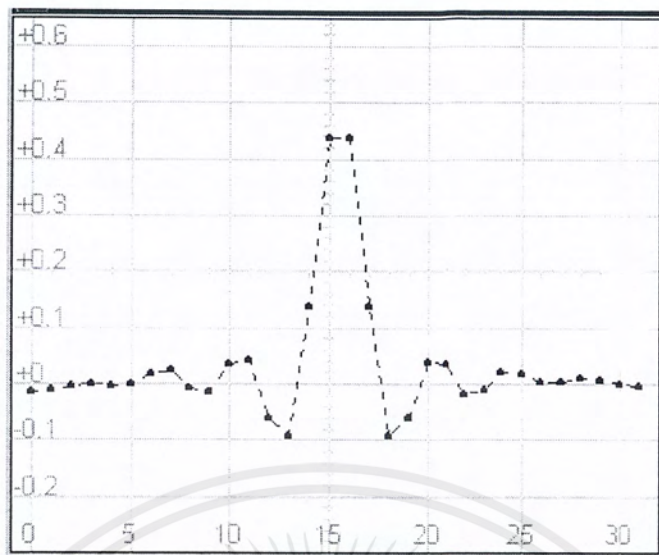
(b) สัญญาณคลาดเคลื่อนที่ได้จากส่วนประมวลสัญญาณ



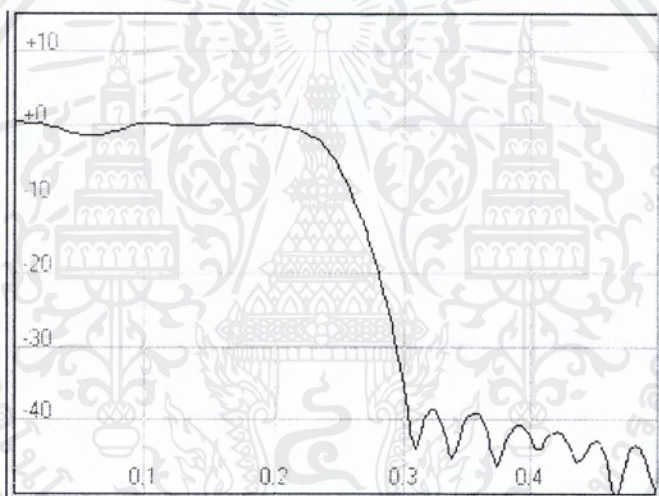
รูปที่ 4.52 แสดงสัญญาณที่ได้จากการจำลองการทำงานโดยใช้โปรแกรมเมทแลบ

(a) แสดงสัญญาณที่ต้องการ

(b) สัญญาณเอาต์พุตจากส่วนประมวลสัญญาณ



รูปที่ 4.53 แถบการการกรองของการกรองแบบปรับตัวได้

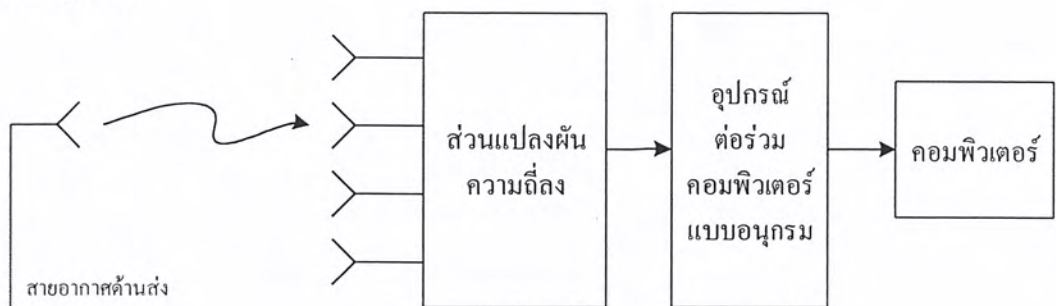


รูปที่ 4.54 การตอบสนองความถี่ของการกรองแบบปรับตัวได้

การทดลองในส่วนของการประมวลสัญญาณโดยใช้ซอฟต์แวร์ในการประมวลผลจากอัลกอริทึมที่ออกแบบอุปกรณ์ที่ใช้ในการทดลอง

1. ส่วนอุปกรณ์ต่อร่วมคอมพิวเตอร์แบบอนุกรม
2. สายอากาศด้านรับแบบแถวลำดับ
3. สายอากาศด้านส่งแบบท่อนำคลื่นปลายเปิด 1 ด้าน
4. คอมพิวเตอร์

การจัดวางอุปกรณ์



รูปที่ 4.55 การจัดวางอุปกรณ์การทดลองในส่วนของการประมวลสัญญาณโดยใช้ซอฟต์แวร์

การทดลอง

1. ส่งสัญญาณทางด้านส่งด้วยความถี่พาหะ 10.6 กิโลเฮิร์ตซ์ มอดูเลตกับสัญญาณความถี่ 1 กิโลเฮิร์ตซ์
2. สายอากาศแถวลำดับด้านรับ รับสัญญาณมาผ่านส่วนแปลงผันความถี่ลง แล้วผ่านเข้าสู่อุปกรณ์ต่อร่วมกับคอมพิวเตอร์แบบอนุกรมซึ่งจะทำการชักตัวอย่างข้อมูลจากสัญญาณอินพุตที่รับเข้ามาทั้ง 4 ช่องสัญญาณส่งเข้าคอมพิวเตอร์ โดยผ่านทางพอร์ตอนุกรม
3. ส่วนของซอฟต์แวร์ จะทำการประมวลสัญญาณอินพุตที่รับเข้ามาแล้วแสดงลักษณะของเอาต์พุตที่ได้

ผลการทดลอง

จากการทดลองเมื่อเราส่งผ่านค่าสัญญาณอินพุตผ่านเข้าไปในคอมพิวเตอร์เพื่อทำการประมวลผลโดยหลักการที่ได้ออกแบบได้ จะได้สัญญาณเอาต์พุตออกมา โดยจะแสดงที่ส่วนสโคปของอัลกอริทึมที่ใช้ในการประมวลผล

จากการทดลองได้กำหนดค่าเริ่มต้นต่างๆ ดังนี้

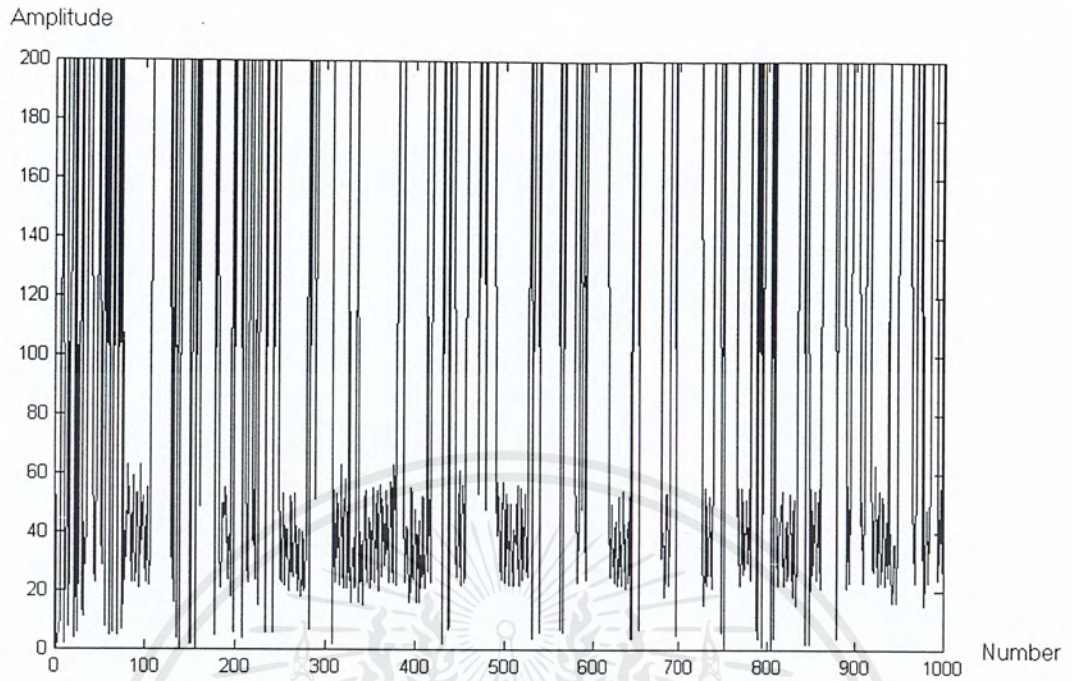
กำหนดค่าสัมประสิทธิ์การลู่เข้าเป็น 0.01

กำหนดการทำซ้ำ 50 ครั้ง

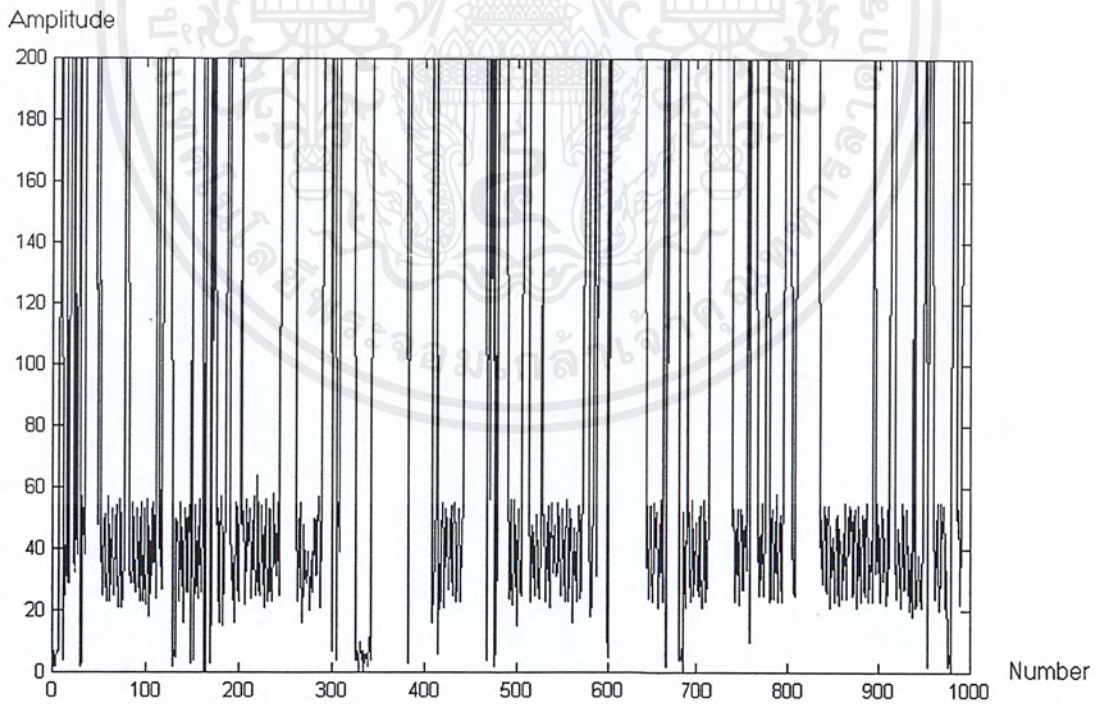
กำหนดเวลาทำงาน ตั้งแต่ 0 มิลลิวินาทีจนกระทั่ง 100 มิลลิวินาทีการทำงาน

กำหนดค่าถ่วงน้ำหนักเริ่มต้นเป็น 0

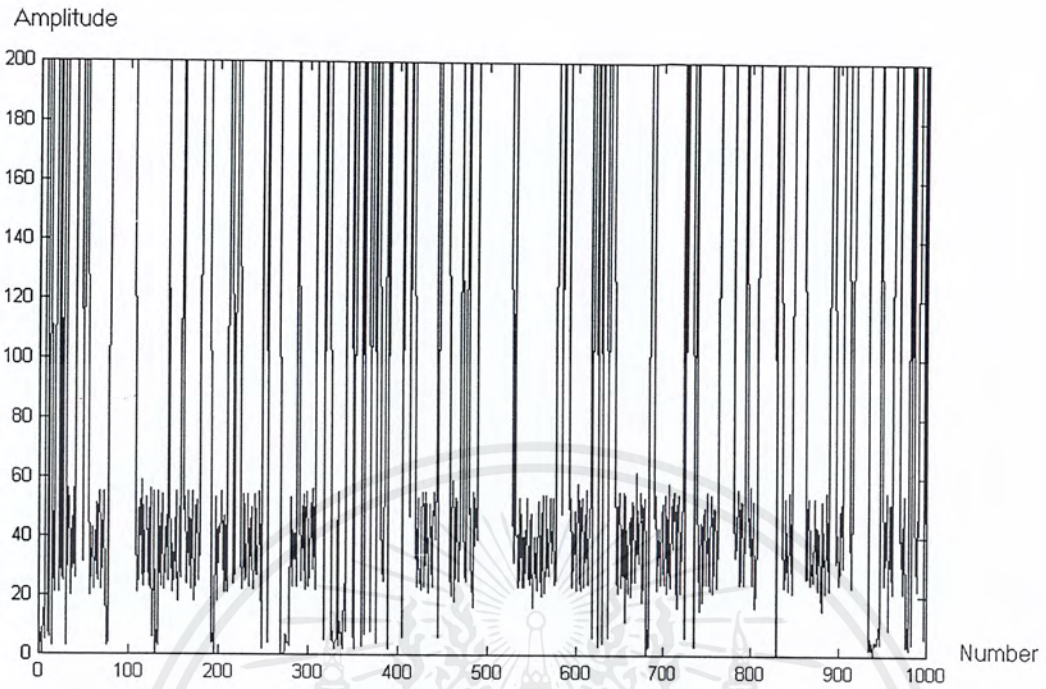
จากการทดลองได้ผลการทดลองดังนี้



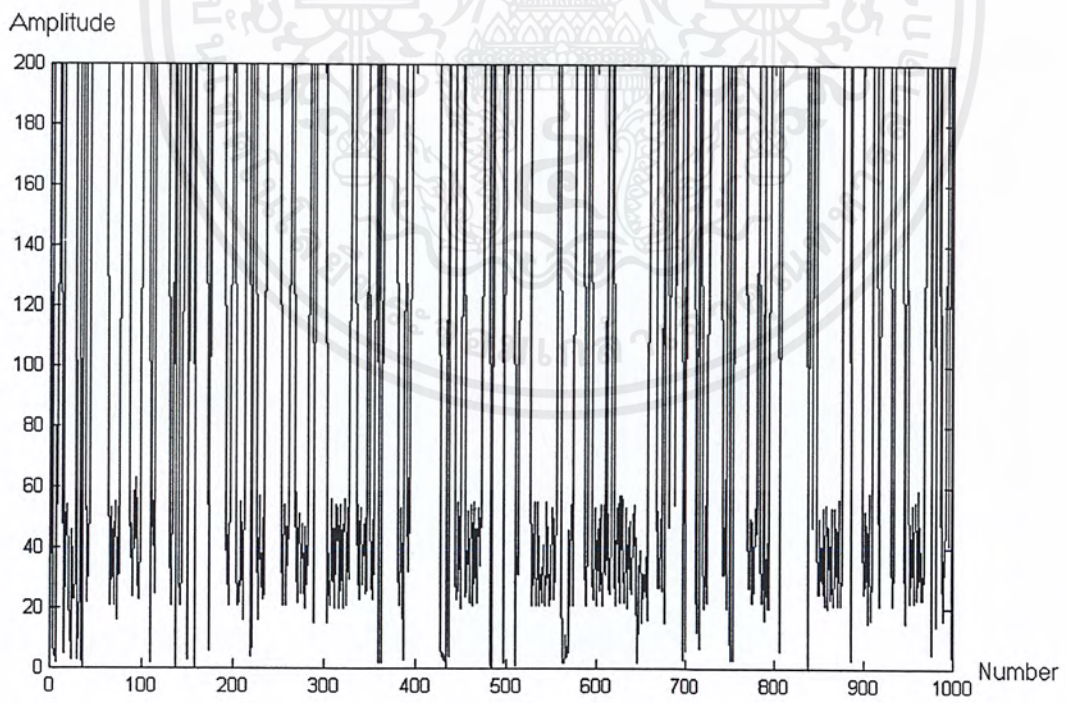
รูปที่ 4.56 แสดงสัญญาณอินพุตที่ 1 ที่รับจากช่องสัญญาณที่ 1



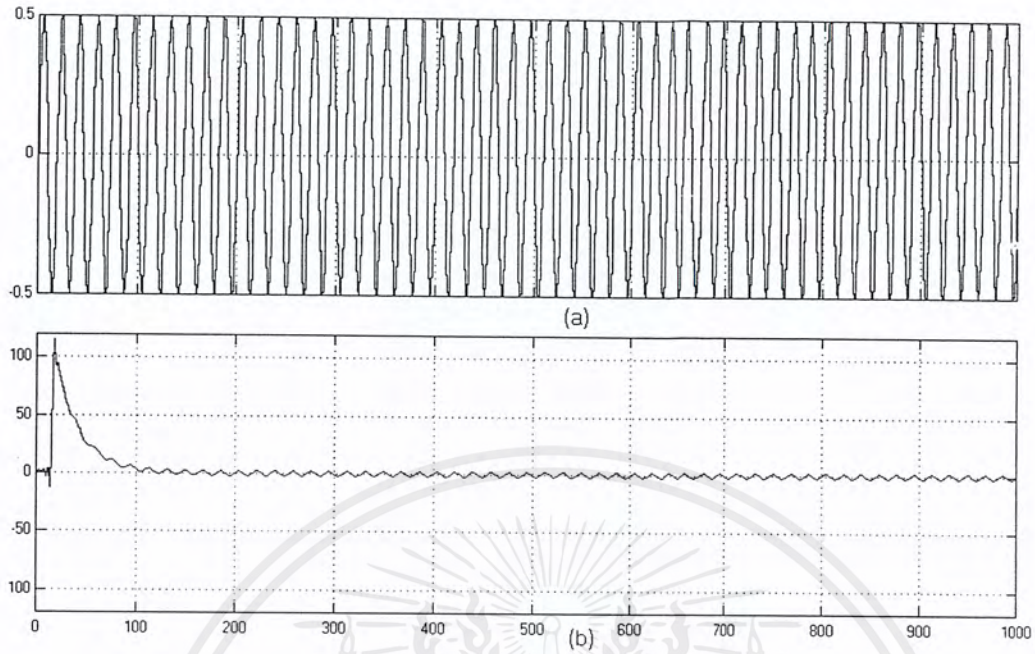
รูปที่ 4.57 แสดงสัญญาณอินพุตที่ 2 ที่รับจากช่องสัญญาณที่ 2



รูปที่ 4.58 แสดงสัญญาณอินพุตที่ 3 ที่รับจากช่องสัญญาณที่ 3



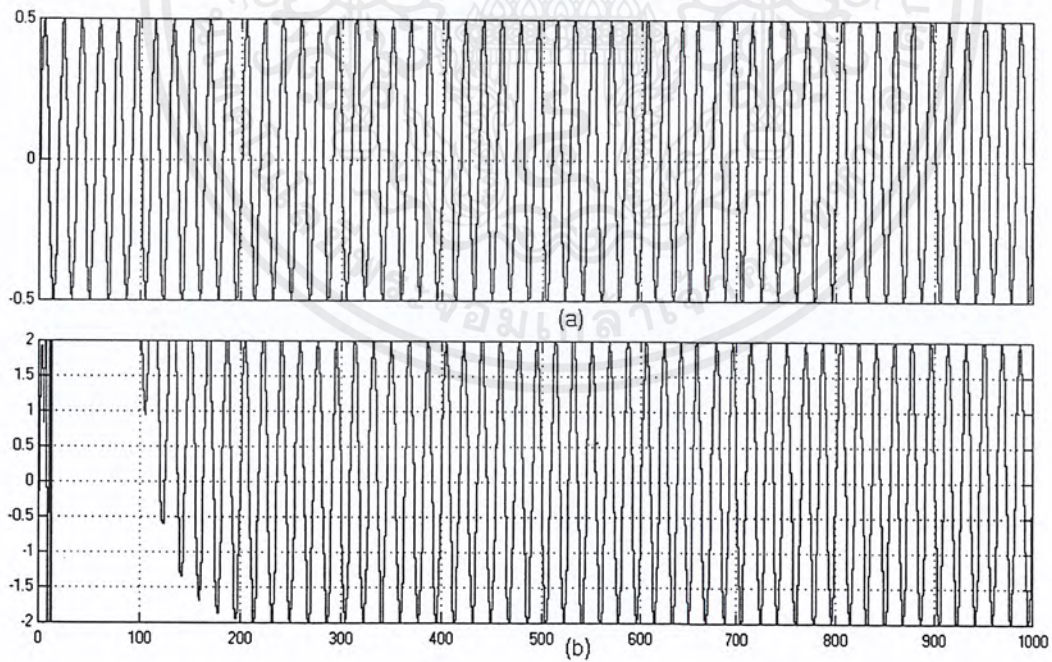
รูปที่ 4.59 แสดงสัญญาณอินพุตที่ 4 ที่รับจากช่องสัญญาณที่ 4



รูปที่ 4.60 แสดงการเปรียบเทียบรูปสัญญาณระหว่าง

(a) สัญญาณอ้างอิง

(b) สัญญาณเอาต์พุตที่ผ่านการประมวลผลแล้ว



รูปที่ 4.61 รูปขยายของรูปที่ 4.60

(a) สัญญาณอ้างอิง

(b) สัญญาณเอาต์พุตที่ผ่านการประมวลผลแล้ว

บทที่ 5

บทสรุปและบทวิจารณ์

จากการศึกษาและออกแบบวงจรรวมทั้งการพัฒนาซอฟต์แวร์ส่วนประมวลผลสัญญาณชนิดปรับตัวได้ ซึ่งขั้นตอนการทำงานนั้นสามารถแบ่งเป็น 3 ส่วนตามลำดับดังนี้

1. การจำลองการทำงานของวงจรจริงโดยใช้โปรแกรมซิมูลิค์ ของเมทแลบ

จากผลทางคอมพิวเตอร์ทำให้ทราบว่าการที่ได้ออกแบบไปนั้นมีความกระบวนการทำงานเป็นอย่างไร และจะเห็นได้ว่า วงจรที่ออกแบบสามารถปรับความแรงของสัญญาณเอาต์พุต จนกระทั่งได้เท่ากับขนาดของสัญญาณที่ต้องการ และมีขนาดเท่าเดิมไปตลอด แสดงว่าความแรงของสัญญาณคงที่ ถึงแม้ว่าสัญญาณอินพุตที่เข้ามาจะเป็นสัญญาณที่มีเฟสต่างกัน ซึ่งโดยทั่วไปแล้วการที่สัญญาณที่เข้ามาในลักษณะนี้จะถือได้ว่าเป็นสัญญาณรบกวน จากผลการจำลองรูปแบบการทำงานจะเห็นได้ว่า สัญญาณดังกล่าวจะถูกกดลงไปเหลือไว้เฉพาะสัญญาณเอาต์พุตที่ต้องการเท่านั้น

2. ในส่วนของการสร้างวงจรประมวลผลสัญญาณอนาลอก ซึ่งวงจรที่สร้างขึ้นมาได้จำลองรูปแบบมาจากวงจรที่ได้ทำการทดลองการทำงานโดยโปรแกรมซิมูลิค์แล้ว

จากผลการทดลอง เห็นได้ว่าวงจรสามารถทำการปรับแต่งสัญญาณอินพุตที่รับเข้ามาให้มีรูปสัญญาณและขนาดเหมือนกับสัญญาณที่ต้องการ แต่ด้วยข้อจำกัดทางด้านอุปกรณ์ ทำให้การทำงานของวงจรไม่เสถียรเท่าที่ควร หมายความว่า ช่วงเวลาที่ทำการปรับแต่งสัญญาณเสร็จแล้ว สามารถแสดงรูปสัญญาณที่ปรับได้ไม่นานพอที่จะเห็นได้ชัดเจน

3. ในส่วนของการพัฒนาซอฟต์แวร์เพื่อใช้ในการประมวลผลสัญญาณโดยระบบดิจิทัลสำหรับในส่วนนี้มีอยู่ 3 ขั้นตอนย่อยๆ ที่ต้องทำการศึกษา และควบคุมการทำงานโดยใช้โปรแกรม

3.1 ส่วนที่ 1 เป็นส่วนที่ใช้ในการแปลงสัญญาณอนาลอกให้เป็นสัญญาณดิจิทัลแล้วทำการส่งข้อมูลเข้าคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม

3.2 ส่วนที่ 2 เป็นส่วนของโปรแกรมที่ทำการเชื่อมโยงระหว่างข้อมูลที่ได้รับมาจากพอร์ตอนุกรมของคอมพิวเตอร์กับโปรแกรมแอปพลิเคชันที่ใช้ในการประมวลผล ซึ่งในโครงงานนี้ใช้ Visual Basic ในการควบคุม

3.3 ส่วนที่ 3 เป็นส่วนของโปรแกรมแอปพลิเคชันเพื่อใช้ในการประมวลผลสัญญาณที่ส่งผ่านเข้ามา ซึ่งกระบวนการทำงานของโปรแกรมในส่วนนี้จะทำงานตามหลักการของอัลกอริทึมกำลังสองเฉลี่ยน้อยที่สุด ในโครงงานนี้เลือกใช้โปรแกรมเมทแลบ เนื่องจากมีอุปกรณ์เครื่องมือที่อำนวยความสะดวกในการทำงานมากกว่า และเป็นโปรแกรมที่รองรับการทำงานทางวิศวกรรมเช่นการประมวลผลแบบเวลาจริง

แต่เนื่องจากในส่วนของการพัฒนาซอฟต์แวร์นี้ ยังไม่พร้อมที่จะทำงานร่วมกันกับระบบการทำงานของ

สายอากาศ ดังนั้นจึงทำการสมมติข้อมูลป้อนให้กับซอฟต์แวร์แทนข้อมูลจริง

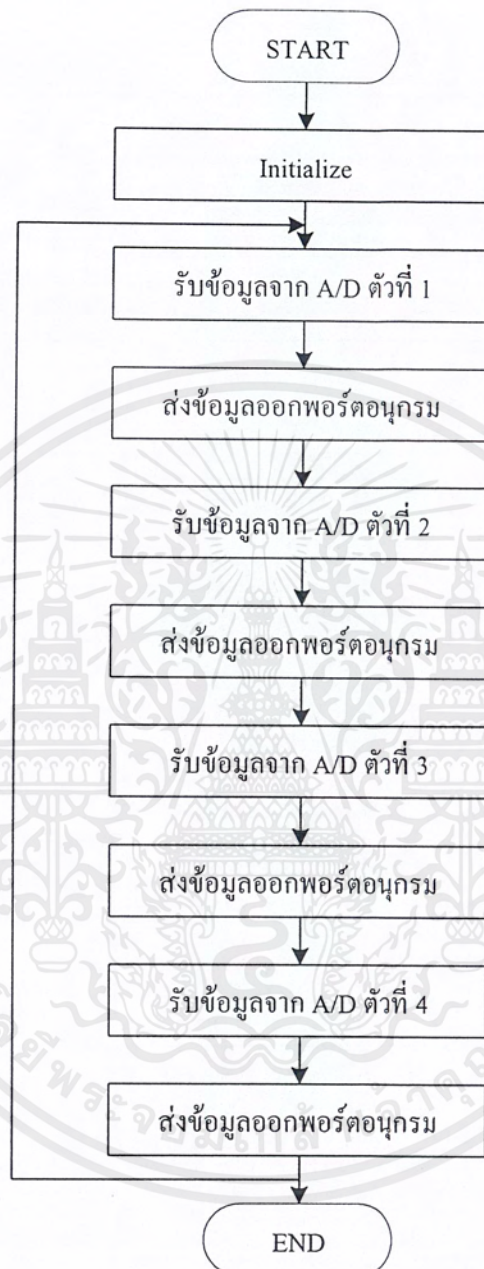
ปัญหาในการทดลอง

1. เนื่องจากว่าระบบของสายอากาศแบบปรับตัวได้นี้ ตัวสายอากาศของระบบใช้สายอากาศแบบท่อนำคลื่นปลายเปิด 1 ด้าน ที่มีก้านไดโอดในตัว เพื่อเป็นออสซิลเลเตอร์ในตัว และด้วยความไม่เสถียรของก้านไดโอด ทำให้รูปสัญญาณที่ได้ไม่คงที่ เพราะเกิดการไม่ล็อกกันของสัญญาณที่รับเข้ามาโดยสายอากาศแถวลำดับ
 2. วงจรที่รับสัญญาณจากสายอากาศมีแบนด์วิดท์แคบ ดังนั้นการดีเทกต์สัญญาณเพื่อให้ได้สัญญาณเบสแบนด์ที่สอดคล้องตามกับสัญญาณพาห์ จึงทำได้ในช่วงเวลาเท่านั้น อันเนื่องมาจากสาเหตุในข้อ 1
 3. ในส่วนของวงจรประมวลสัญญาณ เนื่องจากว่าวงจรที่สร้างส่วนใหญ่เป็นอุปกรณ์ที่มีความไวสูงทางด้านอุณหภูมิ ดังนั้นเมื่อทำงานเป็นเวลานาน จึงทำให้สัญญาณที่ได้เกิดการผิดเพี้ยนได้
 4. สำหรับปัญหาที่เกิดขึ้นจากโครงการนี้ในด้านซอฟต์แวร์มีอยู่สองส่วนด้วยกันคือ
 - 4.1 ฮาร์ดแวร์ที่ได้ทดลอง ยังไม่มีประสิทธิภาพเพียงพอ สำหรับรับสัญญาณเข้าสู่คอมพิวเตอร์ ทำให้การสร้างสัญญาณมีความผิดเพี้ยนผลที่ได้จึงไม่ถูกต้อง ดังนั้นการพัฒนาต่อไปควรคำนึงถึงอุปกรณ์สำหรับออกแบบระบบฮาร์ดแวร์ให้เหมาะสม
 - 4.2 ในการส่งข้อมูลระหว่างโปรแกรมแมทแลบและ Visual Basic ไม่สามารถส่งข้อมูลได้อย่างต่อเนื่อง และอัลกอริทึม ในการส่งที่ใช้ในการทดลองไม่สามารถทำให้สมบูรณ์ได้ นั่นคือ การส่งข้อมูลแบบเมตริกซ์ระหว่างโปรแกรมเป็นข้อจำกัดอย่างหนึ่งจึงควรส่งข้อมูลแบบเวกเตอร์แทน
- เนื่องจากว่าโครงการนี้เป็นการศึกษาสายอากาศปรับตัวได้ในระดับพื้นฐาน และด้วยความหวังที่ว่า การศึกษาครั้งนี้จะเป็นการนำไปสู่การกระตุ้นในคนที่ศึกษาอยู่ในปีถัดไปได้มีโอกาสเรียนรู้และนำไปพัฒนาให้ดียิ่งๆ ขึ้นไป

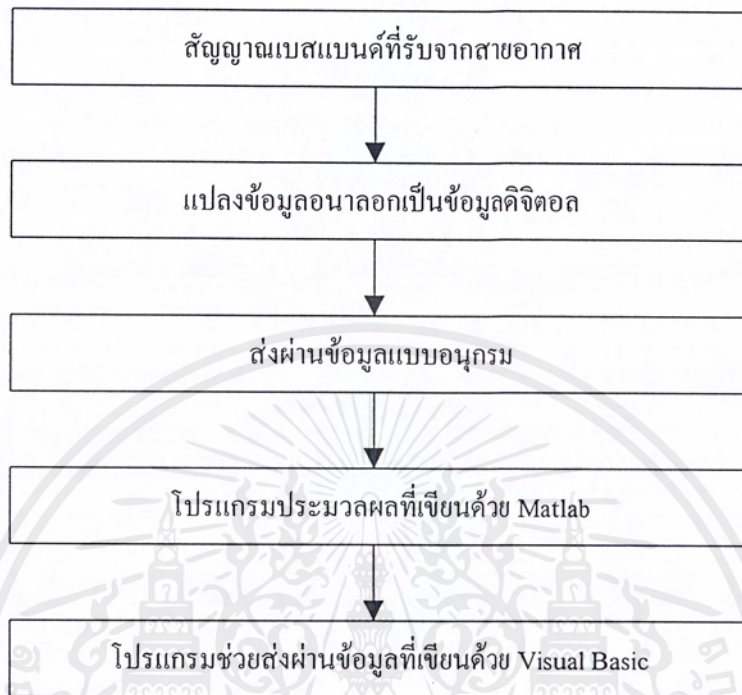
ภาคผนวก



บล็อกไดอะแกรมแสดงการเชื่อมต่อข้อมูลและส่งออกข้อมูลดิจิทัลเข้าคอมพิวเตอร์



บล็อกไดอะแกรมแสดงการประมวลผลด้วยซอฟต์แวร์



โปรแกรมประมวลผลสัญญาณโดยใช้แมทแลบ

```
lmsdemo32;  
l = 0;  
while l < 1000  
    l = l+1  
    x(l)=1  
    d=AntValue(1,1:1:1);  
    z=AntValue(2,1:1:1);  
    A=AntValue(3,1:1:1);  
    b=AntValue(4,1:1:1);  
    set_param('lmsdemo32/Constant','value',num2str(AntValue(1,x(1))));  
    set_param('lmsdemo32/Constant1','value',num2str(AntValue(2,x(1))));  
    set_param('lmsdemo32/Constant2','value',num2str(AntValue(3,x(1))));  
    set_param('lmsdemo32/Constant3','value',num2str(AntValue(4,x(1))));  
    [t,xl,y] = sim('lmsdemo32');  
end  
figure;  
plot(d);  
figure;  
plot(z);  
figure;  
plot(A);  
figure;  
plot(b);  
figure;  
plot(y);
```



โปรแกรมควบคุมการแปลงข้อมูลจากอนาล็อกเป็นข้อมูลดิจิทัลโดยใช้ภาษาAssembly

; THIS PROGRAM IS FOR 11.0592MHz OSCILLATOR

```
ORG 0000H
MOV A,#00H
MOV R0,#00H
MOV R1,#00H
MOV R2,#00H
MOV R3,#00H
MOV PCON,#80H      ;SMOD=1
MOV SCON,#40H      ;8bit UART baud rate variable
MOV TMOD,#20H      ;8bit auto reload use timer1
MOV TH1,#0FFH      ;baud 57600bps
SETB TR1           ;control timer1 to work
MOV P0,#0FFH       ;CLEAR DATA BUS
MOV P1,#0FFH       ;TO BE INPUT PORT
CLR P1.1           ;CHIP SELECT
NOP
NOP
NOP
NOP
NOP
MAIN: CLR P1.0      ;A2D READ ANALOG
NOP
NOP
NOP
NOP
MOV P2,#00H        ;READ BUFFER_0
MOV A,P0            ;MOVE BUFFER VALUE TO ACC
MOV SBUF,A          ;SEND TO SERIAL PORT
JNB TI,$            ;WAIT TILL DONE
CLR TI              ;CLEAR FLAG
MOV P2,#01H        ;READ BUFFER_1
MOV A,P0            ;MOVE BUFFER VALUE TO ACC
MOV SBUF,A          ;SEND TO SERIAL PORT
JNB TI,$            ;WAIT TILL DONE
CLR TI              ;CLEAR FLAG
MOV P2,#02H        ;READ BUFFER_2
MOV A,P0            ;MOVE BUFFER VALUE TO ACC
MOV SBUF,A          ;SEND TO SERIAL PORT
JNB TI,$            ;WAIT TILL DONE
CLR TI              ;CLEAR FLAG
MOV P2,#03H        ;READ BUFFER_3
MOV A,P0            ;MOVE BUFFER VALUE TO ACC
MOV SBUF,A          ;SEND TO SERIAL PORT
JNB TI,$            ;WAIT TILL DONE
CLR TI              ;CLEAR FLAG
SETB P1.0          ;UNABLE READ ANALOG
MOV P2,#0FFH
SJMP MAIN
END
```

โปรแกรมอินเทอร์เฟซและรับข้อมูลผ่านพอร์ตอนุกรมโดยใช้ Visual Basic

```
Option Explicit
Const MaxData = 32766
Const MaxZero = 32766
Dim AntValue(4, MaxData) As Double
Dim Ch1Value(MaxData) As Byte
Dim Ch2Value(MaxData) As Byte
Dim Ch3Value(MaxData) As Byte
Dim Ch4Value(MaxData) As Byte
Dim AntNumber As Integer
Dim DataCount As Integer
Dim Chk As Boolean
Dim MatlabCtrl As Object

Private Sub CreateMFile()
    Dim ch, Value As Integer
    Dim tmp As String
    Open "D:\Matlab\work\mfile.m" For Output As #1
    Print #1, "AntValue = ["
    For ch = 1 To 4
        tmp = "["
        'For Value = 0 To DataCount
        For Value = 1 To 1000
            tmp = tmp & Str(AntValue(ch, Value)) & ",..."
            Print #1, tmp
            tmp = ""
        Next Value
        If ch < 4 Then
            tmp = tmp & "], "
            Print #1, tmp
        Else
            tmp = tmp & "]]"
            Print #1, tmp
        End If
    Next ch
    Close #1
    MsgBox "Creating M-File for MATLAB in mfile.m is completed."
End Sub

Private Sub ReloadComm()
    If MSComm1.PortOpen Then
        MSComm1.PortOpen = False
    End If
    MSComm1.CommPort = 1
    MSComm1.InputLen = 8
    MSComm1.Settings = "57600,n,8,1"
    MSComm1.InputMode = comInputModeText
    MSComm1.PortOpen = True
End Sub

Private Sub StartReceive(dCnt As Integer)
    Dim i, ch, Value As Integer
    'StatusBar.Max = dCnt
    StatusBar.Max = 1000
    'For Value = 1 To dCnt
    For Value = 1 To 1000
        StatusBar.Value = Value
        For ch = 1 To 4
            If (ch <> 1) Or (Value <> 1) Then
```

```

i = 0
Do While MSComm1.InBufferCount = 0
    If i > MaxZero Then
        MsgBox "Timeout please try again later."
        StartButton.Enabled = True
        ProcessButton.Enabled = False
        ClearButton.Enabled = False
        Exit Sub
    End If
    i = i + 1
Loop
AntValue(ch, Value) = AscB(MSComm1.Input)

'If Mscomm1.input is binary, make sure to convert them to
decimal

Select Case ch Mod 4
    Case 1: ListCH1.AddItem AntValue(ch, Value)
    Case 2: ListCH2.AddItem AntValue(ch, Value)
    Case 3: ListCH3.AddItem AntValue(ch, Value)
    Case 0: ListCH4.AddItem AntValue(ch, Value)
End Select
End If
Next ch
Next Value
MsgBox ("Received and logging completed")
StartButton.Enabled = True
ProcessButton.Enabled = True
End Sub

Private Sub ClearButton_Click()
    Dim ch, Value As Integer
    For ch = 1 To 4
        For Value = 0 To MaxData
            AntValue(ch, Value) = 0
        Next Value
    Next ch
    ListCH1.Clear
    ListCH2.Clear
    ListCH3.Clear
    ListCH4.Clear
    MsgBox "All data cleared"
    ProcessButton.Enabled = False
    StartButton.Enabled = True
    ClearButton.Enabled = False
End Sub

Private Sub ExitButton_Click()
    If MsgBox("Do you really want to exit?", 68, "Question") = 6 Then
        End
        MSComm1.PortOpen = False
    Else
        StartButton.Enabled = True
        ReloadComm
    End If
End Sub

Private Sub Form_Load()
    ReloadComm
    ProcessButton.Enabled = False
    StartButton.Enabled = True

```

```

        ClearButton.Enabled = False
        Set MatlabCtrl = CreateObject("Matlab.Application")
    End Sub

Private Sub ProcessButton_Click()
    CreateMFile

    Set MatlabCtrl = CreateObject("matlab.application")
    MatlabCtrl.Execute ("mfile;")
    'MatlabCtrl.Execute ("AntValue = AntValue(:,2:503);")
    MatlabCtrl.Execute ("loop32")

    'ProcessButton.Enabled = False
    'MsgBox "Process completed"
End Sub

Private Sub StartButton_Click()
    ReloadComm
    StartButton.Enabled = False
    Dim Cnt, DataCount As Integer
    Dim tmp As Byte
    DataCount = Val(InputBox("How many data per channel ? <Max " & Str
(MaxData) & "> "))
    Cnt = 0
    tmp = AscB(MSComm1.Input)
    Do While tmp = 0
        Cnt = Cnt + 1
        If Cnt > MaxZero Then
            MsgBox "Timeout, please try again later."
            StartButton.Enabled = True
            ProcessButton.Enabled = False
            ClearButton.Enabled = False
            ReloadComm
            Exit Sub
        End If

        tmp = AscB(MSComm1.Input)
    Loop
    AntValue(1, 1) = tmp
    ListCH1.AddItem AntValue(1, 1)
    Call StartReceive(DataCount)

End SubOption Explicit
Const MaxData = 32766
Const MaxZero = 32766
Dim AntValue(4, MaxData) As Double
Dim Ch1Value(MaxData) As Byte
Dim Ch2Value(MaxData) As Byte
Dim Ch3Value(MaxData) As Byte
Dim Ch4Value(MaxData) As Byte
Dim AntNumber As Integer
Dim DataCount As Integer
Dim Chk As Boolean
Dim MatlabCtrl As Object

Private Sub CreateMFile()
    Dim ch, Value As Integer
    Dim tmp As String
    Open "D:\Matlab\work\mfile.m" For Output As #1
    Print #1, "AntValue = ["
    For ch = 1 To 4

```

```

    tmp = "["
    'For Value = 0 To DataCount
    For Value = 1 To 1000
    tmp = tmp & Str(AntValue(ch, Value)) & ",..."
        Print #1, tmp
        tmp = ""
    Next Value
    If ch < 4 Then
        tmp = tmp & "], "
        Print #1, tmp
    Else
        tmp = tmp & "]] "
        Print #1, tmp
    End If
Next ch
Close #1
MsgBox "Creating M-File for MATLAB in mfile.m is completed."

```

End Sub

```

Private Sub ReloadComm()
    If MSComm1.PortOpen Then
        MSComm1.PortOpen = False
    End If
    MSComm1.CommPort = 1
    MSComm1.InputLen = 8
    MSComm1.Settings = "57600,n,8,1"
    MSComm1.InputMode = comInputModeText
    MSComm1.PortOpen = True
End Sub

```

```

Private Sub StartReceive(dCnt As Integer)
    Dim i, ch, Value As Integer
    'StatusBar.Max = dCnt
    StatusBar.Max = 1000
    'For Value = 1 To dCnt
    For Value = 1 To 1000
        StatusBar.Value = Value
        For ch = 1 To 4
            If (ch <> 1) Or (Value <> 1) Then
                i = 0
                Do While MSComm1.InBufferCount = 0
                    If i > MaxZero Then
                        MsgBox "Timeout please try again later."
                        StartButton.Enabled = True
                        ProcessButton.Enabled = False
                        ClearButton.Enabled = False
                        Exit Sub
                    End If
                    i = i + 1
                Loop
                AntValue(ch, Value) = AscB(MSComm1.Input)
            End If
        Next ch
    Next Value
End Sub

```

'If Mscmm1.input is binary, make sure to convert them to decimal

```

Select Case ch Mod 4
    Case 1: ListCH1.AddItem AntValue(ch, Value)
    Case 2: ListCH2.AddItem AntValue(ch, Value)
    Case 3: ListCH3.AddItem AntValue(ch, Value)
    Case 0: ListCH4.AddItem AntValue(ch, Value)
End Select

```

```

        End Select
    End If
    Next ch
    Next Value
    MsgBox ("Received and logging completed")
    StartButton.Enabled = True
    ProcessButton.Enabled = True
End Sub

Private Sub ClearButton_Click()
    Dim ch, Value As Integer
    For ch = 1 To 4
        For Value = 0 To MaxData
            AntValue(ch, Value) = 0
        Next Value
    Next ch
    ListCH1.Clear
    ListCH2.Clear
    ListCH3.Clear
    ListCH4.Clear
    MsgBox "All data cleared"
    ProcessButton.Enabled = False
    StartButton.Enabled = True
    ClearButton.Enabled = False
End Sub

Private Sub ExitButton_Click()
    If MsgBox("Do you really want to exit?", 68, "Question") = 6 Then
        End
        MSComm1.PortOpen = False
    Else
        StartButton.Enabled = True
        ReloadComm
    End If
End Sub

Private Sub Form_Load()
    ReloadComm
    ProcessButton.Enabled = False
    StartButton.Enabled = True

    ClearButton.Enabled = False
    Set MatlabCtrl = CreateObject("Matlab.Application")
End Sub

Private Sub ProcessButton_Click()
    CreateMFile

    Set MatlabCtrl = CreateObject("matlab.application")
    MatlabCtrl.Execute ("mfile;")
    'MatlabCtrl.Execute ("AntValue = AntValue(:,2:503);")
    MatlabCtrl.Execute ("loop32")

    'ProcessButton.Enabled = False
    'MsgBox "Process completed"
End Sub

Private Sub StartButton_Click()
    ReloadComm
    StartButton.Enabled = False
    Dim Cnt, DataCount As Integer
    Dim tmp As Byte

```

```
DataCount = Val(InputBox("How many data per channel ? <Max " & Str  
(MaxData) & "> "))
```

```
Cnt = 0
```

```
tmp = AscB(MSComm1.Input)
```

```
Do While tmp = 0
```

```
    Cnt = Cnt + 1
```

```
    If Cnt > MaxZero Then
```

```
        MsgBox "Timeout, please try again later."
```

```
        StartButton.Enabled = True
```

```
        ProcessButton.Enabled = False
```

```
        ClearButton.Enabled = False
```

```
        ReloadComm
```

```
        Exit Sub
```

```
    End If
```

```
        tmp = AscB(MSComm1.Input)
```

```
Loop
```

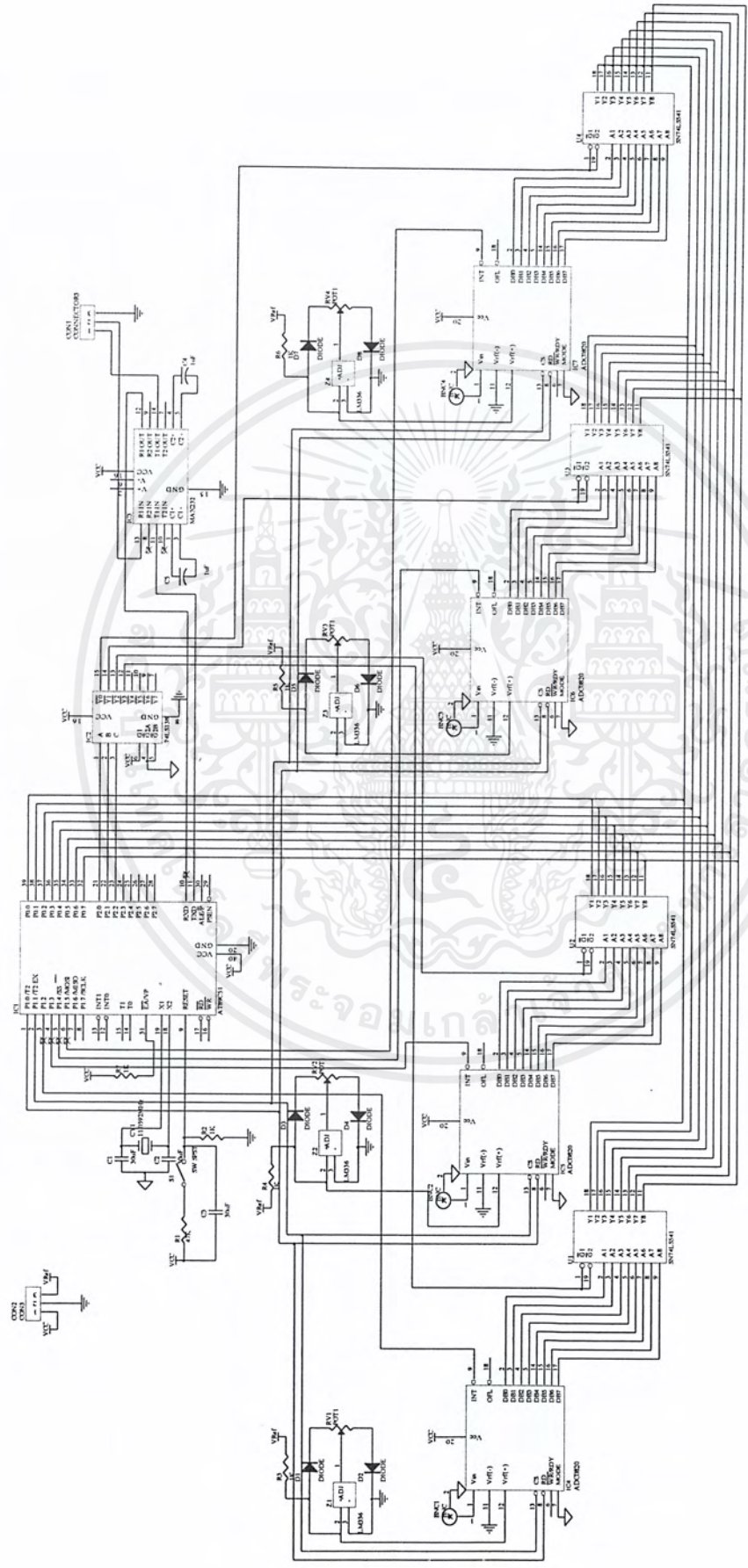
```
AntValue(1, 1) = tmp
```

```
ListCH1.AddItem AntValue(1, 1)
```

```
Call StartReceive(DataCount)
```

```
End Sub
```





วงจรแปลงข้อมูลขนาดออกเป็นข้อมูลดิจิทัล

บรรณานุกรม

- [1] ปราโมทย์ วาดเขียน และ วิวัฒน์ กิรานนท์, “พื้นฐานการสื่อสารข้อมูล,” ภาควิชาโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, กรุงเทพฯ, 2536.
- [2] มงคล เชนครินทร์ และชาติ ศรีไพพรรณ, “อิเล็กทรอนิกส์พื้นฐาน,” บริษัทซีเอ็ดยูเคชั่น จำกัด, กรุงเทพฯ, 2532.
- [3] Comton R.T., “ADAPTIVE ANTENNA : Concepts and performance,” Prentice-Hall Inc., New Jersey, 1988.
- [4] Farhang-Boroujeny B., “ADAPTIVE FILTERS : Theory and applications,” John Wiley., Chichester, 1998
- [5] Garret T. Okamoto., “SMART ANTENNA SYSTEMS AND WIRELESS LANS,” Kluwer Academic., Boston, 1999
- [6] Widrow B. and Stearns Samuel D. ., “ADAPTIVE SIGNAL PROCESSING,” Prentice-Hall Inc., New Jersey, 1985.
- [7] Widrow B., “ADAPTIVE ANTENNA SYSTEM,” Proceeding of the IEEE, Vol. 55, No.12, December 1967: 2143-2150.

กิตติกรรมประกาศ

รายงานฉบับนี้คงไม่สามารถสำเร็จลุล่วงไปได้ ถ้าหากไม่ได้รับความช่วยเหลือและคำแนะนำที่เป็นประโยชน์อย่างมากในการดำเนินงานจากบุคคลหลายๆ ท่าน ดังนี้

รศ.ดร.โมไนย ไกรฤกษ์

อาจารย์ที่ปรึกษา

ดร. ทองทอด วานิชศรี

รศ. ณรงค์ เหมกรณ์

อ. ชูวงศ์ พงศ์เจริญพาณิชย์

อาจารย์ที่ปรึกษา

นายชาญชัย งามโสภา

นายอานัติ เหมือนชู

นายไพศาล งามจรรยาภรณ์

นางสาววัลย์ลิกา บัวสมบุญ

นางสาวมิ่งขวัญ ธนะชัยพันธ์

นายสุทธิ แก้วสองศรี

