



เครื่องตรวจสอบสภาวะไฟฟ้า 3 เฟส
3 PHASE LINE MONITOR



โดย
นายคมสันต์ จักรอด 34162144
นายนพพร กลิ่นบุบผา 34162160
นายสมเกียรติ สุกภาวาว 34162174

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

สาขาคอมพิวเตอร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2535

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีกรนำไปใช้

032๗26

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

สาขา เทคโนโลยีคอมพิวเตอร์อุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องตรวจสอบสภาวะไฟฟ้า 3 เฟส

3 PHASE LINE MONITOR

ผู้จัดทำ

1. นายคมสันต์ จัศจรรย์ 34162144
2. นายนพพร กลิ่นบุบผา 34162160
3. นายสมเกียรติ สุภาถาวร 34162174



..... อาจารย์ที่ปรึกษา
(อาจารย์ ภาคกร หุตะสิงภาศ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032726

บทคัดย่อ

ปริญญาโทฉบับนี้ ได้เสนออุปกรณ์ควบคุมระบบไฟฟ้า 3 เฟส ครอบคลุมไมโครคอนโทรลเลอร์ MCS-48 เบอร์ CPU 8049 ซึ่งอุปกรณ์นี้ทำหน้าที่ในการตัดวงจรไฟฟ้า 3 เฟส เมื่อระบบไฟฟ้านี้มีแรงดันไฟฟ้าสูงสุดไม่เท่ากัน แรงดันไฟฟ้าไม่ครบ 3 เฟส หรือแรงดันไฟฟ้าเฟสใดเฟสหนึ่งขาดหายไป และมีการเรียงลำดับเฟสที่ไม่ถูกต้อง ครอบคลุมหลักการ ANALOG TO DIGITAL CONVERTER เปลี่ยนสัญญาณอนาล็อก (ANALOG) ที่มาจากระบบไฟฟ้า 3 เฟส ซึ่งผ่านการลดแรงดันมาแล้ว ทว่าการเปลี่ยนให้เป็นดิจิตอล (DIGITAL) แล้วผ่านเข้าไมโครคอนโทรลเลอร์ MCS-48 ครอบคลุมไมโครคอนโทรลเลอร์ MCS-48 จะทำการประมวลผลและสามารถควบคุมระบบการทางานได้ซึ่งอ่านโปรแกรมจาก EPROM ภายนอกแล้วแสดงผลด้วย SEVEN SEGMENT

Abstract

This thesis explains the accessories that can control a three-phase electrical system by using microcontroller MODEL MCS-48, which can be used with CPU 8049 in 3 phases. When the electrical system has the same voltage. If the electricity voltage is not in 3 phases or the phase is missing, there is a phase disorder. The convertor will change Analog to Digital signal from a three-phases electrical system, by decreasing voltage to Digital and to Microcontroller (MCS-48). Microcontroller MCS-48 will change to conclusion and it can control the programme by reading from outer EPROM and produce a Seven-Segment result.

สารบัญ

	หน้า
บทคัดย่อ	
Abstract	
บทนำ	I-1
บทที่ 1. MCS_48	
1.1 โครงสร้างภายนอก	1-3
1.2 โครงสร้างภายใน	4-4
1.3 หน่วยความจำ	4-5
1.4 ส่วนการติดต่ออุปกรณ์ภายนอก I/O	5-9
1.5 การอินเทอร์รัพท์	9-13
1.6 TIMER/COUNTER	14-15
1.7 ชุดคำสั่ง 8048	15-17
บทที่ 2. การแปลงสัญญาณอนาล็อกเป็นดิจิตอล	
2.1 ความหมายของค่า	18-21
2.2 ADC 0809	22-22
บทที่ 3. การทำงานของอุปกรณ์ควบคุมระบบไฟฟ้า 3 เฟส 50 Hz	23-26
บทที่ 4. TURBO 48	
4.1 ส่วนประกอบ ฮอฟท์แวร์	27-27
4.2 คุณสมบัติของ TURBO 48	27-29
4.3 การเริ่มใช้ TURBO 48	29-30
บทที่ 5. การตรวจการเรียงลำดับเฟส	
5.1 ส่วนประกอบการตรวจการเรียงลำดับเฟส	31-33
5.2 สรุปการทำงานของตรวจการเรียงลำดับเฟส	34-34
บทที่ 6. สรุป	
6.1 สรุปการทำงาน	35-35
6.2 ปัญหาที่เกิดขึ้น	35-35

กิตติกรรมประกาศ

เอกสารอ้างอิง

ภาคผนวก A โปรแกรมทดลองและวงจร
เอกสารฉบับเอกสารที่แนบมาหรือที่ปรึกษาการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นที่ยอมรับกันโดยทั่วไปว่า ไฟฟ้ามีความสำคัญต่อชีวิตประจำวันมาก และมีแนวโน้มที่จะเพิ่มขึ้นเรื่อย ๆ โดยเฉพาะอย่างยิ่งโรงงานอุตสาหกรรม จะขาดไฟฟ้าไม่ได้เลย เนื่องจากสามารถควบคุมได้ง่าย ควบคุมได้ด้วยระบบอัตโนมัติ และสามารถเปลี่ยนเป็นพลังงานรูปอื่นได้ง่าย ไม่ว่าจะเป็นพลังงานความร้อน พลังงานแสง หรือพลังงานกล โดยเฉพาะอย่างยิ่ง เครื่องจักรที่ใช้ในโรงงานอุตสาหกรรมปัจจุบันนี้ มาจากพลังงานไฟฟ้าเกือบทั้งหมด ดังนั้นระบบไฟฟ้าที่จ่ายให้กับเครื่องจักรทั้งหลายจะต้องมีความถูกต้อง ในเรื่องของแรงดันเฟส การเรียงลำดับเฟส และทุกเฟสต้องมีแรงดันครบ ถ้าไม่เป็นดังนี้ จะทำให้เครื่องจักรเกิดความเสียหายได้ง่าย ดังนั้นจึงต้องมีอุปกรณ์ควบคุมระบบไฟฟ้าที่มีความเที่ยงตรงสูง และมีความผิดพลาดน้อย ไว้คอยควบคุม

วัตถุประสงค์ของปริิญาานิพนธ์

จุดประสงค์ของปริิญาานิพนธ์นี้ คือ พัฒนาอุปกรณ์ควบคุมระบบไฟฟ้า 3 เฟส เนื่องจากก่อนหน้านี้ ได้มีการสร้างเครื่องควบคุมระบบไฟฟ้า แต่ใช้อุปกรณ์อิเล็กทรอนิกส์เป็นตัวควบคุม ซึ่งมีข้อเสียคือ เวลาใช้นาน อุณหภูมิที่สูงจะมีผลต่ออุปกรณ์อิเล็กทรอนิกส์ ทำให้ค่าที่ได้อาจไม่ตรงกับจุดประสงค์ในการวัด และมีความยุ่งยากในการต่อวงจรมาก นั่นคือถ้าต้องการให้ระบบควบคุมทำงานได้ตรงตามจุดประสงค์ ต้องใช้อุปกรณ์ที่มีประสิทธิภาพอย่างมาก ดังนั้นจึงได้ใช้ไมโครคอนโทรลเลอร์เป็นตัวประมวลผล ในการทำงาน โดยอาศัยหลักการแปลงสัญญาณจากอนาลอกให้เป็นดิจิตอล แล้วผ่านเข้าไมโครคอนโทรลเลอร์ประมวลผล

ขอบเขตของปริิญาานิพนธ์

ใช้ไมโครคอนโทรลเลอร์ควบคุมระบบไฟฟ้า โดยเมื่อแรงดันแต่ละสายของ 3 เฟส มีระดับแรงดันไม่เท่ากัน เฟสใดเฟสหนึ่งไม่มีแรงดัน และไม่มีการเรียงลำดับของเฟส เครื่องควบคุมจะทำการส่งสัญญาณไปตัดวงจร แต่เมื่อระบบไฟฟ้าเป็นปกติ เครื่องก็จะทำงานเป็นปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

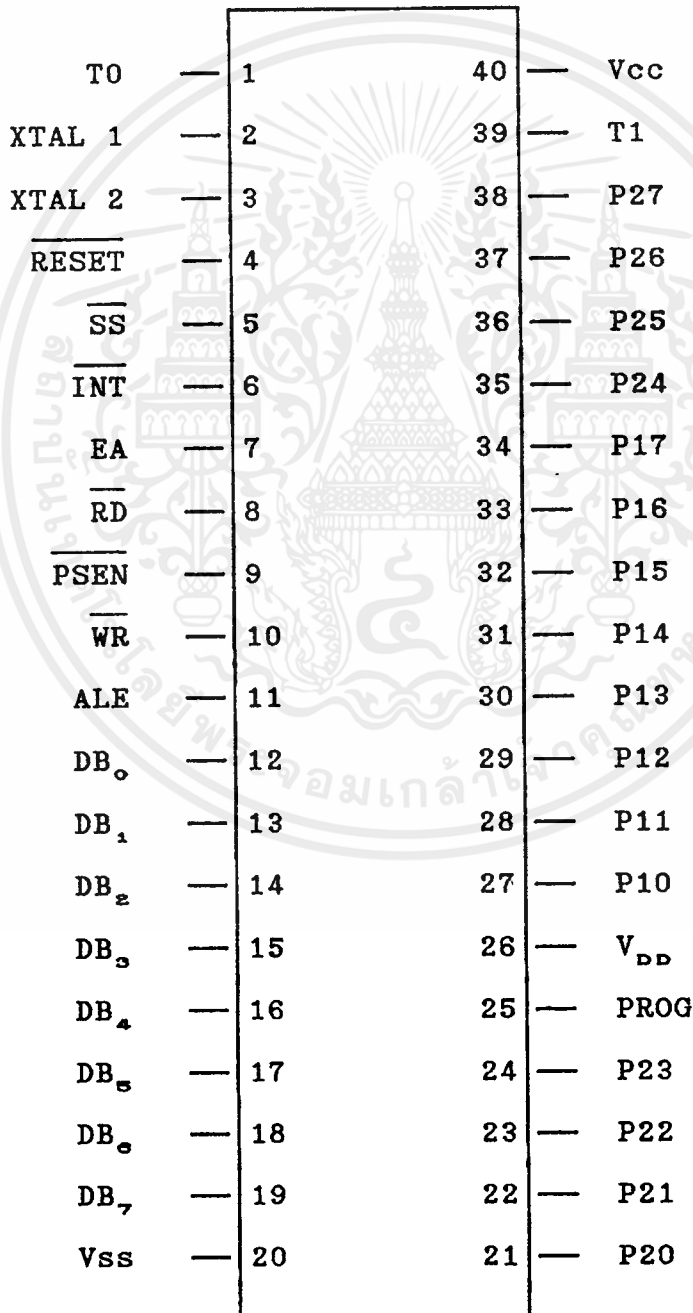
บทที่ 1

MCS-48

โครงสร้างของ MCS-48 แบ่งเป็น 2 ลักษณะ

1. โครงสร้างภายนอก
2. โครงสร้างภายใน

โครงสร้างภายนอก



ขา 1	TO	เป็นขาทดสอบสัญญาณเข้า หรือเป็นขาเอาต์พุตสัญญาณ CLOCK ของระบบ
ขา 2	XTAL 1	สำหรับต่อขาข้างหนึ่งของคริสตัลภายนอก เพื่อใช้ขอสซิสเลเตอร์ภายนอก หรือแหล่งกำเนิดสัญญาณจากภายนอกต่อเข้าขานี้
ขา 3	XTAL 2	สำหรับต่อขาอีกข้างหนึ่งของคริสตัล
ขา 4	$\overline{\text{RESET}}$	เป็นสัญญาณอินพุตแอกทีฟ"0" ใช้ RESET ระบบในตัว 8049
ขา 5	$\overline{\text{SS}}$	จะใช้ร่วมกับ ALE เพื่อทดสอบคำสั่งทีละคำสั่ง (single step)
ขา 6	$\overline{\text{INT}}$	เป็นขาอินพุตรับสัญญาณอินเคอร์รัพต์ภายนอก
ขา 7	EA	เมื่อเป็นสถานะตรรกสูง จะเป็นการคิดต่อรับรหัสข้อมูล จากภายนอกของชิปเกิลชิปตัวนี้ และควบคุมให้ตัว PROGRAM COUNTER แพรทช์ รหัสจากโปรแกรมภายนอกทั้งหมด
ขา 8	$\overline{\text{RD}}$	จะเป็นสัญญาณควบคุมการรับ หรือการอ่านข้อมูลจากภายนอก ผ่านเข้าขาข้อมูลการอ่าน และสัญญาณนี้จะสลับครบทุกครั้งที่มีการอ่านข้อมูลไม่ว่าจะอ่านรหัสจากภายนอกหรือภายใน
ขา 9	$\overline{\text{PSEN}}$	เป็นสัญญาณ PROGRAM STORE ENABLE สัญญาณควบคุมการอ่านรหัสคำสั่งทุกคำสั่งจากหน่วยความจำภายนอก และสัญญาณนี้จะแอกทีฟต่ำ เมื่อมีการแพร่ชข้อมูลโปรแกรมจากภายนอกเท่านั้น
ขา 10	$\overline{\text{WR}}$	เป็นสัญญาณควบคุมการส่ง หรือเขียนข้อมูลออกภายนอกผ่านบัสข้อมูล และสัญญาณนี้จะสลับครบออกทุกครั้ง ที่มีการเขียนข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา 11 ALE เป็นสัญญาณ ADDRESS LATCH ENABLE สัญญาณนี้จะเกิดทุกวัฏจักรของการแพชในแต่ละครั้ง และใช้ประโยชน์ในการส่งสัญญาณนาฬิกาเอาท์พุทด้วยการใช้สัญญาณขอบขาลงของสัญญาณนี้เป็นการสรีทรบ แลทซ์แอดเดรสข้อมูล จากภายนอกในหน่วยความจำโปรแกรม
- ขา 12-19 D₀-D₇ เป็น BIDIRECTIONAL PORT ทาหน้าที่ต่างวดังนี้
1. เป็นที่ส่งข้อมูลเข้าหรือออก ในขณะที่ซึ่งค้ด้วยสัญญาณ RD หรือ WR สรีทรบ และสามารถทาหน้าที่เป็นพอร์ต สำหรับแลทซ์ข้อมูล
 2. ใช้เป็นตัวส่งข้อมูลตัวนับโปรแกรมบิทอันค้บค้าระหว่างการแพชโปรแกรมจากภายนอก และรับรหัสคำสั่งภายใต้การส่งสัญญาณควบคุม PSEN
 3. ใช้เป็นตัวส่งข้อมูลทั้งแอดเดรส และข้อมูล ในการทางานกับ EPROM
- ขา 20 Vss เป็นสายดินของวงจร
- ขา 21-24 P₂₀-P₂₃ ใช้เป็นตัวส่งค่าอันค้บสูง ของแอดเดรสตัวนับโปรแกรม และใช้ในการติดค้กับหน่วยความจำภายนอก
- ขา 35-38 P₂₄-P₂₇ และใช้เป็น 4 บิทเอโรบัส สำหรับการขยายออก เพื่อใช้กับพอร์ตเอโรของ 8243 และอีก 4 บิทอันค้บสูงจะใช้เป็นพอร์ต
- ขา 25 PROG ใช้เป็นสัญญาณเอาท์พุทสรีทรบ สำหรับการควบคุมการใช้รวมกับการขยายพอร์ตเอโรกับแรม 8243 และใช้บ้อนโปรแกรมพัลส์ ขนาด 18 V เข้าขานี้สำหรับการเขียนโปรแกรมเข้า 8748/8749

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา 26 VDD ต่อกับแบตเตอรี่ขนาดเล็ก เพื่อเลี้ยงแรมภายใน ไม่ให้ข้อมูลสูญหายเมื่อไม่มีไฟเลี้ยง Vcc
- ขา 27-34 P10-P17 เป็นพอร์ตขนาด 8 บิต แบบ QUASI-BIDIRECTIONAL พอร์ต 1.
- ขา 39 T1 เป็นขาอินพุตทดสอบ หรือเป็นอินพุตของ CLOCK ที่ป้อนให้กับ TIMER/COUNTER ภายใน 8049
- ขา 40 Vcc จ่ายแรงดันไฟ 5 โวลต์เข้าที่ขานี้ ในขณะที่ทำงานปกติ และขณะที่อัปเดตโปรแกรมเข้า 8748/8749

โครงสร้างภายใน

1. หน่วยคำนวณคณิตศาสตร์ (ALU)

ตัว ALU มีความสามารถที่จะทำงานตามหน้าที่ต่าง ๆ ดังนี้

- บวกด้วยตัวทดหรือปราศจากตัวทด (ADD WITH OR WITHOUT CARRY)
- ทางานทางตรรก (AND ,OR, EXCLUSIVE OR)
- การเพิ่มหรือลดค่าหนึ่งค่า (INCREMENT/DECREMENT)
- การแปลงกลับค่าบิต (BIT COMPONENT)
- วนบิตทางซ้ายหรือขวา (LEFT OR RIGHT ROTATE)
- การสลับค่า nibble (SWEP NIBBLE)
- การปรับค่าเป็น BCD DECIMAL (BCD DECIMAL ADJUST)

2. แอ็คคิวมิวเลเตอร์

จะเป็นตัวรีจิสเตอร์ที่สำคัญที่สุดของ MCS-49 ในตัวทรานเซสเซอร์จะทำหน้าที่เป็นตัวรับข้อมูล จากแหล่งข้อมูลที่ส่งมา และเป็นตัวส่งผลลัพธ์ให้กับรีจิสเตอร์ตัวรับ (DESTINATION REGISTER) ในการทำงานของ ALU การคิดต่อกับอุปกรณ์ภายนอก และคิดต่อกับหน่วยความจำ จะต้องผ่านแอ็คคิวมิวเลเตอร์เสมอ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. แพลกตัวทค (CARRY FLAG)

เป็นตัวบอกสถานะการทงานของ MCS-49 ว่าทงานเกินจำนวนบิตที่มีอยู่หรือไม่ถ้าเกินแพลกตัวทคนี้จะถูก เช็ค

4. ตัวถอดรหัส (INSTRUCTION REGISTER AND DECODER)

รับคำสั่งจาก ROM แล้ว decode สร้าง OUTPUT ควบคุม ALU

หน่วยความจำ (MEMORY)

1. หน่วยความจำโปรแกรม (PROGRAM MEMORY -ROM)
2. หน่วยความจำข้อมูล (RAM)

เบอร์	หน่วยความจำภายใน		ROM สำรอง
	โปรแกรม ROM	ข้อมูล RAM	
8050 AH	4K x 8	256 x 8	มี
8049 AH	2K x 8	128 x 8	มี
8048 AH	1K x 8	64 x 8	มี
8040 AHL	ไม่มี	256 x 8	มี
8039 AHL	ไม่มี	128 x 8	มี
8035 AHL	ไม่มี	64 x 8	มี
8749 H	2K x 8 (EPROM)	128 x 8	ไม่มี
8748 H	1K x 8 (EPROM)	64 x 8	ไม่มี

ตารางที่ 2 รายละเอียดต่าง ๆ ของตระกูล MCS-49

หน่วยความจำโปรแกรม

ตำแหน่งที่สำคัญของหน่วยความจำส่วนนี้ คือ

1. ADDRESS \$0000 เป็นจุดเริ่มต้นของโปรแกรมหลังจากการรีเซต
2. ADDRESS \$0003 เป็นจุดเริ่มต้นของการ INTERRUPT SERVICE ROUTINE (ISR) ของการอินเทอร์รัพท์ จากภายนอกที่เข้ามาทางขา INT
3. ADDRESS \$0007 เป็นจุดเริ่มต้นของ ISR ของการอินเทอร์รัพท์ จากตัว TIMER/COUNTER ที่อยู่ภายใน 8049
4. ADDRESS \$0100 เป็นจุดเริ่มต้นของ USER PROGRAM แต่อาจจะไม่จำเป็นต้องเริ่มที่นี้เสมอไป
5. ADDRESS \$0300-\$03FF เป็นข้อมูลค่าคงที่ต่าง ๆ จำนวน 256 ไบต์

หน่วยความจำข้อมูล

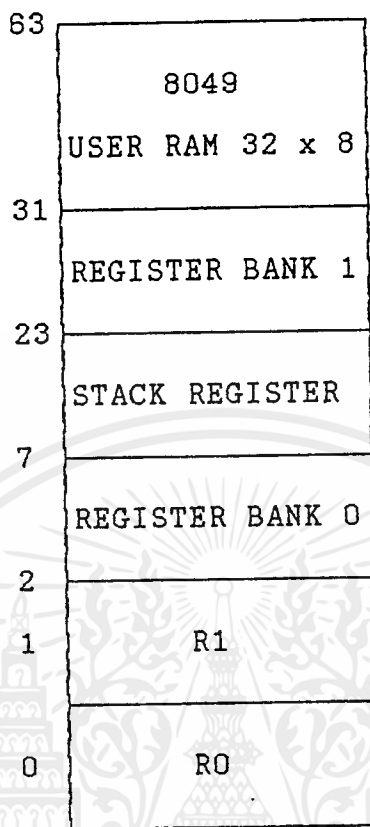
สำหรับหน่วยความจำข้อมูลภายในตัว MCS-48 นั้นมีด้วยกัน 128 ไบต์ ซึ่งหน่วยความจำทั้งหมดนี้จะแบ่งออกเป็น 2 ส่วน

1. ส่วนรีจิสเตอร์
2. ส่วนของแอสติก

ส่วนรีจิสเตอร์ ใ้แก่ หน่วยความจำแอดเดรส 00-07 ซึ่งเราเรียกว่า รีจิสเตอร์แบงก์ 1 (BANK 1) แต่ละแบงก์จะมีรีจิสเตอร์อยู่ 8 ตัว ซึ่งมีชื่อเรียกว่า R0-R7 ในการใช้งานรีจิสเตอร์เหล่านี้ เราจะสามารถใช้งานได้ทีละแบงก์เท่านั้น โดยคำสั่ง SEL RB₀ หรือ SEL RB₁ ในการเลือกแบงก์ของรีจิสเตอร์

รีจิสเตอร์ทั้ง 8 ตัวนี้เป็นรีจิสเตอร์สำหรับใช้งานทั่วไป มีคำสั่งต่าง ๆ ที่กระทำร่วมกับรีจิสเตอร์เหล่านี้ครบ แต่จะมีรีจิสเตอร์ R₀-R₁ สองตัวเท่านั้นที่มีหน้าที่พิเศษ คือสามารถใช้เป็นตัวอ้างถึงหน่วยความจำข้อมูลได้ โดยค่าแอดเดรสจะมีขนาดเพียง 8 บิตเท่านั้น เป็นการจำกัดหน่วยความจำข้อมูลให้มีเพียง 256 ไบต์ ซึ่งจะคลุมพื้นที่ส่วนที่เป็นรีจิสเตอร์ และส่วนที่เป็นแอสติกด้วย ทำให้สามารถติดต่อกับรีจิสเตอร์ หรือแอสติก ในลักษณะที่เป็นหน่วยความจำข้อมูลได้วิธีหนึ่ง การติดต่อยังวิธีนี้ส่วนใหญ่มักจะใช้กับแอสติกมากกว่าส่วนอื่น เพราะว่าคำสั่งต่าง ๆ ของ 8049 ที่เกี่ยวข้องกับแอสติก โดยตรงมีเพียงคำสั่ง CALL กับคำสั่ง RET (RETR) เท่านั้น การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADDRESS



รูป 2. แสดงการแบ่งหน่วยความจำข้อมูลไปใช้งาน

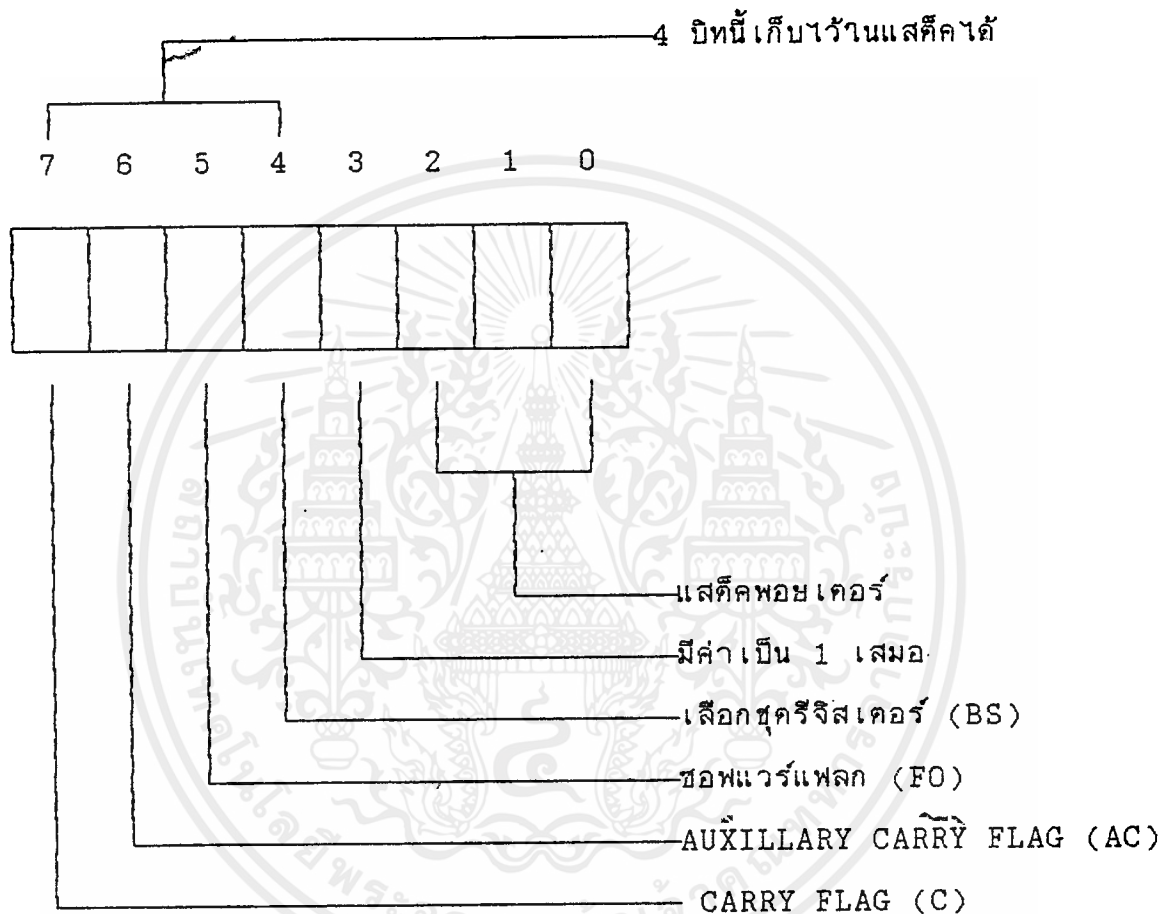
ส่วนของแอสติก ประกอบด้วยแอสติกทั้งหมด 8 ชั้น แต่ละชั้นประกอบด้วยหน่วยความจำข้อมูล 2 ไบท์ เพื่อให้ความเพียงพอกับการเก็บค่าแอดเดรสและเก็บสถานะของโปรแกรมการที่มีแอสติกเพียง 8 ชั้นนี้ทำให้มีข้อจำกัดในการเรียกโปรแกรมย่อย (SUBROUTINE) ซ้อน ๆ กันได้ไม่เกิน แอสติกพอยเตอร์ ก็จะย้อนกลับไปที่แอสติกชั้นแรกใหม่

โดยปกติแล้วแอสติกพอยเตอร์จะชี้ที่ S_0 เป็นแอสติกชั้นแรก เมื่อมีการ CALL หรือการอินเตอร์รัพท์ ก็จะเก็บค่าแอดเดรส และสถานะของโปรแกรม (PROGRAM STATUS WORD) ลงในแอสติกแล้วเพิ่มค่าแอสติกพอยเตอร์ให้ชี้ไปที่ S_1

และเมื่อเอ็กซีคิวต์คำสั่ง RET (RETR) ก็จะลดค่าแอสติกพอยเตอร์ลงไปที่ S_0 แล้วนำค่าแอดเดรส กับสถานะของโปรแกรม ที่เก็บไว้ในแอสติก S_0 กับคืนที่เดิม

PROGRAM STATUS WORD (PSW)

PSW เป็นรีจิสเตอร์ตัวหนึ่ง ที่ทำหน้าที่เก็บค่าของแอสตัสคอยเคอร์ และสถานะของโปรแกรม (สถานะของโปรแกรม ก็คือ ค่าของแฟลกต่าง ๆ) แต่ละบิตของ PSW



รูป 3 แสดงความหมายของแต่ละบิตใน PSW

- C หมายถึง แฟลกตัวทดที่ได้ จากการบวก หรือโดยการเลื่อนบิต
- AC หมายถึง แฟลกตัวทดที่ได้ จากการบวกบิตที่ 3 ของตัวเลข BCD จะเข้าทำงานร่วมกับคำสั่ง DA (DECIMAL ADJUST)
- FO หมายถึง ซอฟต์แวร์แฟลกสามารถใช้งานร่วมได้ทั่วประเทศ คำสั่งที่มีผลต่อแฟลกนี้ได้แก่ คำสั่ง CLR FO และ CPL FO
- BS หมายถึง แฟลกที่แสดงการเลือกแบ่งค้ของรีจิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



นอกจากแฟล็ก FO แล้วยังมีซอฟต์แวร์แฟล็ก F1 อีกตัวหนึ่ง ที่มีลักษณะเช่นเดียวกับแฟล็ก FO ต่างกันเพียงแต่ว่าแฟล็ก F1 ไม่ได้เก็บไว้ใน PSW ด้วย

บิต 0- บิต 2 ของ PSW จะเก็บค่าของแอสต์คพอยเตอร์เอาไว้ เมื่อมีการเรียกโปรแกรมย่อยไม่ว่าจะโดยการ CALL หรือการเรียกอินเตอร์รัพท์ จะมีการเก็บค่าแอดเดรสจากโปรแกรมเคาน์เตอร์ และค่าของ PSW ลงในแอสต์ค สำหรับการ RETURN นั้น จะแบ่งออกเป็นสองแบบ คือ คำสั่ง RET ใช้สำหรับออกจากโปรแกรมย่อย จะเอาเฉพาะค่าแอดเดรสใส่คืนให้แก่โปรแกรมเคาน์เตอร์ เท่านั้น

ส่วนคำสั่ง RETR นั้นใช้สำหรับออกจาก INTERRUPT SERVICE ROUTINE (ISR) ซึ่งจะนำเอาทั้งค่าแอดเดรส และสถานะของโปรแกรม ใส่กลับคืนที่เดิม นอกจากนี้หน่วยความจำข้อมูลภายในตัว 8049 แล้ว เรายังสามารถต่อขยายหน่วยความจำแรมภายนอกได้อีก 256 ไบท์ สำหรับการติดต่อกับหน่วยความจำข้อมูลภายนอกจะยังคงใช้รีจิสเตอร์ R_0 และ R_1 ในการอ้างถึงแอดเดรสเช่นเดิม จะต่างกันตรงที่มีคำสั่งในการติดต่อกับหน่วยความจำข้อมูลแยกจากกันคือ

ใช้คำสั่ง MOV ในการติดต่อกับหน่วยความจำข้อมูลภายใน

ใช้คำสั่ง MOVX ในการติดต่อกับหน่วยความจำข้อมูลภายนอก

ส่วนการติดต่ออุปกรณ์ภายนอก INPUT/OUTPUT

1. พอร์ท (PORT) ภายในตัว MCS-49 จะมีพอร์ทคล้ายกัน 2 พอร์ท คือพอร์ท 1 และพอร์ท 2 ที่มีขนาด 8 บิต พอร์ททั้ง 2 เป็นพอร์ทชนิด QUASI-BIDIRECTIONAL เนื่องจากโครงสร้างของวงจรของพอร์ท แต่ละเส้นจะทำงานเป็นอินพุทและเอาต์พุท ด้วยลักษณะข้อมูลเอาต์พุท จะเป็น STATIC LATCH พอร์ท 1 และ พอร์ท 2 นั้นมีลักษณะพิเศษสามารถเป็นทั้งอินพุทและเอาต์พุท เมื่อเราเอาต์พุทค่าตัวเลขออกมาด้วยคำสั่ง OUT ค่าตัวเลขนั้นจะถูกแลทช์ค้างอยู่ที่ พอร์ทนั้น ถ้ามีข้อมูลใหม่เข้ามาก็จะทับข้อมูลเก่าไป ส่วนการใช้เป็นอินพุทพอร์ทนั้น มีวิธีการเพิ่มขึ้นมาอีกนิด คือ เมื่อต้องการให้บิตใดเป็นอินพุท เราก็ต้องเอาต์พุทลอจิกหนึ่งออกมาที่บิตนั้น จากนั้นการอ่านค่าพอร์ทเข้าไป วงจรภายนอกสามารถที่จะดึงลอจิกที่บิตนั้น ให้กลายเป็นลอจิกศูนย์ ดังนั้นการตรวจสอบว่าวงจรภายนอกส่งสัญญาณอินพุทเข้ามาหรือไม่ ก็ทำได้โดยการตรวจสอบลอจิกศูนย์ที่อินพุทบิตนั้นเอง ส่วนบิตใดที่เราเอาต์พุทลอจิกศูนย์ ออกไปนั้น เมื่ออ่านกลับก็ยังคงเป็นลอจิกศูนย์อยู่ดี วงจรภายนอกไม่สามารถดึงให้กลายเป็นลอจิกหนึ่งได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อแตกต่างระหว่าง พอร์ต 1 และ พอร์ต 2 คือ 4 บิตล่างของพอร์ต 2 (P20-P23) ถูกนำไปใช้งานได้สองลักษณะ คือมีการมัลติเพล็กซ์ ค่าแอดเดรส 4 บิตสูงของโปรแกรมเคาน์เตอร์ (A8-A11) กับค่าของพอร์ต ดังนั้นในการใช้งานส่วนใหญ่แล้วจะมีความสะดวกที่จะใช้ P20-P23 เป็นเอาต์พุตพอร์ต มากกว่าเป็นอินพุตพอร์ต โดยการใช้ตัวแลทช์ขนาด 4 บิต เช่น 74LS175 มาแลทช์ค่าเอาต์พุตไว้ได้

คำสั่งอินพุต เอาต์พุต ที่สามารถเข้ากับพอร์ตนี้ ได้แก่

IN A, Pn

และ OUT Pn, A เมื่อ n เป็นเบอร์พอร์ต

2. บัส (BUS) จัดว่าเป็นพอร์ตหนึ่งเหมือนกัน แต่ลักษณะของคาค่าบัสไม่เหมือนกับพอร์ต 1 และ พอร์ต 2 เท่าใดนัก โดยปกติคาค่าบัสจะทำหน้าที่ส่งแอดเดรส และรับส่งข้อมูล สลับกันแบบมัลติเพล็กซ์ ตลอดเวลา

การใช้งานคาค่าบัสในฐานะเป็นพอร์ตนั้น จะเอาบัสเป็นพอร์ตได้ก็คือเมื่อไม่มีการต่อหน่วยความจำภายนอกเลย และคำสั่งที่ใช้คือ INS A, BUS และ OUTL BUS, A เป็นต้น แต่โดยทั่วไปแล้วเรามักจะต่อหน่วยความจำ ภายนอกกับ 8049 เสมอ การใช้งานคาค่าบัสเป็นพอร์ต จึงไม่ค่อยมีให้เห็น

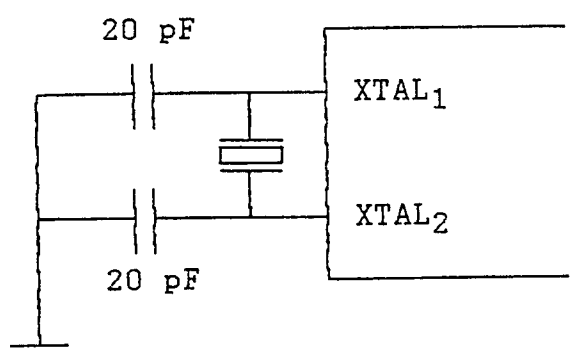
นอกจากลักษณะพิเศษของพอร์ต และคำสั่งอินพุต เอาต์พุตธรรมดาแล้ว 8049 ยังมีคำสั่งอีกสองแบบ ที่ใช้ติดต่อกับพอร์ต คือ

คำสั่ง ANL PORT, DATA เป็นการนำค่าตัวเลข DATA 16 บิต AND กับข้อมูลในพอร์ต

คำสั่ง ORL PORT, DATA เป็นการนำค่าตัวเลข DATA 16 บิต OR กับข้อมูลในพอร์ต

สัญญาณนาฬิกาของระบบและสัญญาณควบคุม

สัญญาณนาฬิกานี้อาจจะสร้างจาก วงจรออสซิลเลเตอร์ภายนอก แล้วบ่อนำให้แก 8049 ทางขา XTAL₁ แต่ภายใน 8049 เอง ก็มีวงจรออสซิลเลเตอร์อยู่ภายในแล้ว เพียงแต่ต่อคริสตอล และตัวเก็บประจุอีก 2 ตัว ดังรูป 4 ก็จะทำให้มีสัญญาณนาฬิกาใช้เช่นกัน สำหรับความถี่ของสัญญาณนาฬิกา นี้ ขึ้นกับความถี่ของคริสตอล จะใช้คริสตอลขนาด 4 MHz-11 MHz กับไอซี 8039 และ 8049



รูป 6 แสดงการต่อวงจรภายนอกเพื่อสร้างฐานเวลา

สัญญาณนาฬิกาที่ได้จากคริสตอล (OSC) ถูกหารความถี่ด้วย 3 ได้เป็นสัญญาณของระบบ (CLK) เราหาสัญญาณ CLK นี้ หารความถี่ด้วย 5 จะได้เป็นสัญญาณ ALE ซึ่งเป็นบล็อกไดอะแกรมของระบบสัญญาณนาฬิกาภายใน 8048

นอกจากสัญญาณนาฬิกาที่ใช้เป็นหลักฐานเวลาของระบบแล้ว 8048 ก็ยังต้องการสัญญาณจากภายนอก เพื่อมาควบคุมการทำงาน เช่น สัญญาณ $\overline{\text{RESET}}$, $\overline{\text{EA}}$, $\overline{\text{INT}}$, $\overline{\text{SS}}$ เป็นต้น และในทางตรงกันข้าม ก็ต้องส่งสัญญาณควบคุม ไปยังอุปกรณ์ภายนอก เช่นกัน สัญญาณเหล่านั้น ได้แก่ $\overline{\text{ALE}}$, $\overline{\text{PSEN}}$, $\overline{\text{WR}}$, $\overline{\text{RD}}$ เป็นต้น เราจะมาทำความรู้จักกับสัญญาณควบคุมเหล่านี้กันเลย โดยเริ่มจาก

สัญญาณควบคุมเอาต์พุต ได้แก่สัญญาณ $\overline{\text{ALE}}$, $\overline{\text{PSEN}}$, $\overline{\text{WR}}$ และ $\overline{\text{RD}}$

ALE (ADDRESS LATCH ENABLE) เป็นสัญญาณนาฬิกาที่ออกมาทุก ๆ แมชชีนไซเคิล (MACHINE CYCLE) หรือทุก ๆ สัญญาณ CLK 5 ลูกตามทีกล่าวก่อนแล้ว เราจะใช้ขอบขาลงของสัญญาณ ALE ในการแลทช์ค่าของแอดเดรสไบต์ค่า ที่มีติเฟลิกซ์มาจากดาต้าบัส

$\overline{\text{PSEN}}$ (PROGRAM STORE ANABLE) เป็นสัญญาณที่แอกทีฟศูนย์และจะแอกทีฟเฉพาะเมื่อ 8048 เพทซ์ คำสั่งจากหน่วยความจำโปรแกรมภายนอกเท่านั้น ในการต่อหน่วยความจำโปรแกรมภายนอกเข้ากับ 8048 นั้น เราจำเป็นต้องแลทช์ค่าแอดเดรส ที่มีติเฟลิกซ์มาจากดาต้าบัสเสียก่อน โดยที่ใช้สัญญาณ ALE เป็น STROBE (ไอซีที่เราานิยมมาใช้แลทช์ค่าแอดเดรสนี้เป็นไอซีเบอร์ 74LS373) ค่าแอดเดรส 8 บิตที่ได้จากการแลทช์ เมื่อนามารวมกับแอดเดรส 4 บิตสูง ($\text{A}_8\text{-A}_{11}$) ซึ่งได้จากการพอร์ต 2 P20-P23 ก็จะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้าไม่ว่ากรณีใด ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอดเดรสขนาด 12 บิต ที่จะใช้อ้างถึงหน่วยความจำโปรแกรมได้ครบ 4 KB จากนั้น เราใช้สัญญาณ $\overline{\text{PSEN}}$ เป็น STROBE ในการเพชคำสั่งจากหน่วยความจำเข้ามายัง 8048 ได้ $\overline{\text{WR}}$ และ $\overline{\text{RD}}$ เป็นสัญญาณที่ ACTIVE LOW แต่จะแอกทีฟเมื่อ 8048 มีการติดต่อหน่วยความจำข้อมูลภายนอก หรือเมื่อเราจะใช้คำสั่งปรับสมათาหน้าที่เป็นพอร์ท เราก็ใช้สัญญาณ WR หรือ RD ที่แอกทีฟมาเป็น STROBE ในการเขียนหรืออ่านข้อมูลกับทางบัสพอร์ทได้ด้วย

จากที่กล่าวมาแล้ว พอจะสรุปเรื่องราวของสัญญาณ $\overline{\text{PSEN}}$, $\overline{\text{WR}}$, $\overline{\text{RD}}$ ได้ว่า สัญญาณ STROBE ทั้ง 3 นี้จะสามารถแอกทีฟได้เพียงครั้งละ 1 สัญญาณเท่านั้น เช่น ในช่วงการเพชคำสั่ง สัญญาณ $\overline{\text{PSEN}}$ จะแอกทีฟส่วน $\overline{\text{WR}}$ และ $\overline{\text{RD}}$ จะไม่แอกทีฟ เป็นต้น

สัญญาณ	สัญญาณจะแอกทีฟเมื่อ
RD	MOVX A,@R,INS A,BUS
WR	MOVX @R,A,OUTL BUS,A
PSEN	การเพชคำสั่งจากภายนอก MOVX,MOVDP3

สัญญาณควบคุมอินพุท ได้แก่ สัญญาณ RESET, SS, EA และสัญญาณ INT

รายละเอียดของสัญญาณต่าง ๆ มีดังนี้

RESET เป็นสัญญาณอินพุทที่บังคับให้ CPU เริ่มทำงานใหม่ ภายหลังการบ่อนลอจิคศูนย์ที่ขาอินพุทนี้ในการใช้งานทั่วไปแล้ว จะมีสวิทช์กดติด บล้อยดับ ทาหน้าทีส่งสัญญาณรีเซท เมื่อ 8048 ถูกรีเซทจะทาให้

1. โปรแกรมเคอร์เนลเตอร์และแอสคิพอยเตอร์ ถูกรีเซทเป็นศูนย์
2. เลือกหน่วยความจำ BANK 0 และ REGISTER BANK 0
3. อินเทอร์รัพท์ ถูกคิสเอเบิ้ล

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการเขียนขึ้นเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใด ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. แพลก F_0 และ F_1 และ 1 เทเมอร์แพลก ถูกเคลียร์

5. T_0 จะกลายเป็นขาอินพุท

\overline{SS}

(SINGLE STEP) เป็นสัญญาณอินพุท ที่สามารถควบคุมให้ 8048 ทำงานที่ละคำสั่ง ความจริงที่เราต้องการคือ ถ้าเราส่งสัญญาณ ALE ก็จะสามารถบังคับให้ 8048 หยุดการทำงานของมันได้ครบเท่าที่ สัญญาณ \overline{SS} ยังคงแอกทีฟอยู่ และในระหว่างที่ 8048 นี้หยุดทำงาน ตัวมันเองจะแสดงค่าแอกเคสออกมาทางคาตาบัส และ 4 บิตล่างของพอร์ท 2 (P_{20} - P_{23}) จากลักษณะนี้เอง จึงเป็นการอำนวยความสะดวกแก่ผู้ใช้ในการติดตามการทำงานของ 8048 ว่าเป็นไปตามโปรแกรมที่เราต้องการหรือไม่

EA

(EXTERNAL ACCESS) ตามที่กล่าวมาแล้วว่า 8048 นั้นแบ่งหน่วยความจำโปรแกรมออกเป็น ส่วนภายนอกและส่วนภายใน 8048 เองการให้ลอจิกแก่ขา EA เป็นการบอกให้ 8048 รับโปรแกรมในหน่วยความจำส่วนใด ถ้า EA มีลอจิกหนึ่ง 8048 จะรับโปรแกรมในหน่วยความจำภายนอกทั้งหมดแต่ถ้าให้ EA มีลอจิก 0 จะเป็นการสั่ง ให้รับโปรแกรมจากหน่วยความจำภายใน และอาจจะรับโปรแกรมต่อมาถึงหน่วยความจำภายนอกที่เราต่อเพิ่มก็ได้

การอินเทอร์รัพท์

อินเทอร์รัพท์เป็นกระบวนการอย่างหนึ่งที่ ระบบไมโครคอมพิวเตอร์ทั้งหลายจำเป็นต้องมีไว้แม้แต่ระบบเล็ก ๆ อย่าง 8048 เองก็ยังมีระบบอินเทอร์รัพท์ 2 อย่างด้วยกันคือ

1. อินเทอร์รัพท์จากอุปกรณ์ภายนอก ที่ส่งสัญญาณอินเทอร์รัพท์ออกมาทางขา INT
2. อินเทอร์รัพท์จากภายใน เป็นการอินเทอร์รัพท์โดย เทเมอร์ เคาน์เตอร์ที่อยู่ภายใน 8048 เมื่อ 8048 ถูกอินเทอร์รัพท์ไม่ว่าจะเป็นอุปกรณ์ชนิดใดก็ตาม 8048 จะตรวจสอบสถานะ การอินเทอร์รัพท์ของตัวเองเสียก่อน ว่าถูกอินเทอร์รัพท์แล้วละก็ 8048 จะหยุดทำงานที่คำสั่งที่อยู่เดิม และก้าวไปตอบรับอุปกรณ์ที่ส่งสัญญาณอินเทอร์รัพท์มา โดยการเก็บค่าของโปรแกรม (PSW) เข้าไปไว้ในแอสติก และกระโดดไปยัง ISR ซึ่งมีอยู่สองแห่งด้วยกันตามประเภทของ

อินเทอร์รัพท์ การขออินเทอร์รัพท์โดยอุปกรณ์ภายนอก ด้วยการส่งสัญญาณมาทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใด ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หา INT นั้น จะทำให้ 8048 กระโดดไปยัง ISR ที่ตำแหน่ง \$0003 แต่เมื่อ 8048 ถูกอินเทอร์รัพท์ด้วย ไทเมอร์\เคาน์เตอร์ ที่ตำแหน่งแอดเดรส \$0007 ในขณะที่ 8048 กำลังรันโปรแกรมใน ISR อยู่ นั้น จะดีสเอเบิลอินเตอร์รัพท์โดยอัตโนมัติ จึงทำให้ไม่สามารถมีการเรียกอินเตอร์รัพท์ซ้ำได้อีก จนกระทั่งเมื่อ 8048 พบกับคำสั่ง RETR การรีนาเบิลอินเตอร์รัพท์ก็คือการยอมให้ 8048 รับรู้อินเตอร์รัพท์ต่าง ๆ ได้ เราสามารถสั่งการรีนาเบิลได้โดยตรง ด้วยคำสั่ง ENI นอกจากนี้ 8048 ก็ยังสามารถรีนาเบิลตัวมันเองโดยอัตโนมัติ เมื่อกระทำคำสั่ง RETR ด้วยเช่นกัน

การดีสเอเบิลอินเตอร์รัพท์ ทำได้ 3 วิธีคือ

1. ใช้คำสั่ง DIS I ในการดีสเอเบิลโดยตรง
2. โดยการรีเซต 8048 จะทำให้อินเตอร์รัพท์ ถูกดีสเอเบิลโดยอัตโนมัติ
3. ในขณะที่ 8048 กำลังอยู่ในโปรแกรม ISR 8048 จะถูกดีสเอเบิลอินเตอร์รัพท์เช่นกัน เพื่อป้องกันการอินเตอร์รัพท์ซ้ำซ้อน

ไทเมอร์\เคาน์เตอร์ (TIMER\COUNTER)

ลักษณะของตัวเคาน์เตอร์ เป็นแบบนับขึ้น มีขนาด 8 บิต สามารถตั้งโปรแกรมให้เริ่มนับที่ค่าใดค่าหนึ่งก็ได้โดยค่าตัวเลขเริ่มต้น จะถูกเก็บไว้ในรีจิสเตอร์ที่มีชื่อว่า T เมื่อเราสั่งให้ไทเมอร์\เคาน์เตอร์ เริ่มทำงาน โดยใช้คำสั่ง START ค่าในรีจิสเตอร์ T จะถูกโหลดเข้าไปในไทเมอร์\เคาน์เตอร์ (T\CNT) แล้วเริ่มการนับสัญญาณ และเพิ่มค่าใน T\CNT เรื่อย ๆ จนเมื่อนับจากค่า \$FF ไปเป็น \$00 T\CNT ก็จะส่งสัญญาณอินเตอร์รัพท์ภายในไปยัง ซีพียู และขณะเดียวกัน ก็จะเซตค่าไทเมอร์แฟลคให้เป็นลอจิกหนึ่ง หลังจากที T\CNT นับถึง \$00 แล้วก็ตาม แต่ T\CNT ก็จะไม่หยุดจนกว่า ซีพียูจะพบกับคำสั่ง STOP หรือซีพียูจะถูกรีเซตใหม่

การทำงานของ T\CNT แบ่งออกเป็นสองแบบด้วยกัน คือ

1. ไทเมอร์ โหมด
2. เคาน์เตอร์ โหมด

- SEL MB₀, SEL MB₁ คำสั่ง เลือกแบ่งค้ของหน่วยความจา
- ENTO CLK คำสั่ง เอาท์พุทสัญญาณนาฬิกาออกมาทางขา T₀
- NOP คำสั่ง NO OPERATION

กลุ่มคำสั่ง เคลื่อนย้ายข้อมูล ๖ค้แก้

- MOV คำสั่ง เคลื่อนย้ายข้อมูลระหว่างรีจิสเตอร์ หรือ หน่วยความจาข้อมูลภายในกับแอสคิวมูลเตอร์
- MOVX คำสั่ง เคลื่อนย้ายข้อมูลระหว่างแอสคิวมูลเตอร์กับ หน่วยความจาข้อมูลภายนอก
- MOVP คำสั่งอ่านข้อมูลจากหน่วยความจาโปรแกรม ที่อยู่ ในเพจเดียวกัน
- XCH คำสั่ง แลก เปลี่ยนข้อมูลระหว่างแอสคิวมูลเตอร์กับ รีจิสเตอร์ หรือ กับหน่วยความจาข้อมูลภายใน
- XCHD คำสั่ง แลก เปลี่ยนข้อมูลระหว่างแอสคิวมูลเตอร์กับ หน่วยความจาข้อมูลภายใน โดยแลก เปลี่ยนเฉพาะ ข้อมูล 4 บิตล่างเท่านั้น

กลุ่มคำสั่ง เคลื่อนย้ายข้อมูล ๖ค้แก้

- IN & OULT คำสั่งอ่านข้อมูลหรือ เขียนข้อมูลระหว่าง แอสคิวมูลเตอร์กับ พอร์ท 1 หรือ พอร์ท 2
- INS & OULT คำสั่งอ่านข้อมูลหรือ เขียนข้อมูลระหว่าง แอสคิวมูลเตอร์กับ คาต้าบัส
- ANL & ORL คำสั่ง เกี่ยวกับการ AND หรือการ OR ข้อมูล กับ ข้อมูลเติมในพอร์ท 1 หรือพอร์ท 2 หรือ คาต้าบัส

กลุ่มคำสั่ง เกี่ยวกับการแอสคิวมูลเตอร์ ๖ค้แก้

- ADD & ADDC คำสั่ง เกี่ยวกับการบวกตัวเลข แบบไม่คิคตัวทคและ แบบที่คิคตัวทคด้วย
- ANL, ORL, XRL, CPL, CLR คำสั่ง เกี่ยวกับการกระทำทางลอจิก คีอ การ AND, OR, EX-CLUSIVE OR, COMPLEMENT, CLEAR
- INC & DEC คำสั่ง เพิ่มหรือลดค่าของแอสคิวมูลเตอร์ หรือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานรีจิสเตอร์ หรือหน่วยความจาข้อมูลภายใน การค้
ไม่ว่ากรณีใด ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- DA คำสั่งปรับค่าตัวเลขฐานสิบ DECIMAL ADJUST
- SWAP คำสั่งสลับค่าข้อมูลในแอสคิวเลเตอร์ ระหว่าง 4 บิตล่างกับ 4 บิตบน

กลุ่มคำสั่งเกี่ยวกับ เทเมอร์\เคาน์เตอร์ ได้แก่

- MOV คำสั่งเคลื่อนย้ายข้อมูล ระหว่างแอสคิวเลเตอร์ กับรีจิสเตอร์ T
- START & STOP คำสั่งเริ่ม หรือ หยุดการทำงานของ เทเมอร์\เคาน์เตอร์
- EN TCNTI & DIS TCNTI คำสั่งอื่นาเปิดเทเมอร์อินเตอร์รัพท์

กลุ่มคำสั่ง เกี่ยวกับการกระโดด ได้แก่

- JMP คำสั่งกระโดด
- JMPP คำสั่งกระโดดทางอ้อม ภายในเพจเดียวกัน
- JTO, JNTO, JT1, JNT1, JN1 คำสั่งตรวจสอบขาอินพุตทดสอบ T₀, T₁, INT
- JC, JNC, JFO, JF1 คำสั่งตรวจสอบแฟลคต่าง ๆ เช่น แฟลคตัวทดสอบแวร์แฟลค
- JZ, JNC คำสั่งทดสอบค่าในแอสคิวเลเตอร์ว่าเป็นศูนย์หรือไม่
- DJNZ คำสั่งเกี่ยวกับการวนลูป ุคยการลดค่ารีจิสเตอร์ลงหนึ่งแล้วตรวจสอบว่าเป็นศูนย์หรือไม่ ถ้าค่าในรีจิสเตอร์ไม่เท่ากับศูนย์ ก็จะกระโดดไป
- JB คำสั่งตรวจสอบลอจิกหนึ่ง ของบิตใดบิตหนึ่ง ในแอสคิวเลเตอร์

กลุ่มคำสั่ง เกี่ยวกับโปรแกรมย่อย ได้แก่

- CALL คำสั่ง เรียกโปรแกรมย่อย
- RET, RETR คำสั่ง RETURN

กลุ่มคำสั่ง เกี่ยวกับแฟลค ได้แก่

- CLR, CPL คำสั่ง CLEAR และ COMPLEMENT แฟลคตัวทดสอบและซอฟต์แวร์แฟลค F₀, F₁

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

การแปลงสัญญาณอนาล็อกเป็นดิจิทัล

(ANALOG TO DIGITAL CONVERSION)

ในการควบคุมระบบไฟฟ้า สัญญาณที่รับมายังเครื่องควบคุม จะถือว่าเป็นสัญญาณอนาล็อก (ANALOG SIGNAL) แต่ในการประมวลผลของเครื่องควบคุมไมโครโปรเซสเซอร์ สามารถกระทำได้เฉพาะสัญญาณดิจิทัลเท่านั้น ด้วยเหตุผลดังกล่าว จึงจำเป็นต้องแปลงสัญญาณอนาล็อกเป็นดิจิทัล

จึงต้องส่งข้อมูลให้เครื่องควบคุมไปทำการประมวลผลต่อไป ส่วนของวงจรที่ทำหน้าที่เปลี่ยนสัญญาณลักษณะนี้เรียกว่า ANALOG TO DIGITAL CONVERTER (ADC)

ความหมายของคำต่อไปนี้

1. RESOLUTION

RESOLUTION ของ A/D คิดเป็นขนาดหรือจำนวนของบิตของสัญญาณดิจิทัล ที่ถูกเปลี่ยนจากสัญญาณอนาล็อก แต่ละ STEP ของ LSB จะเป็นฟังก์ชันของ RESOLUTION และในการใช้งานสิ่งสำคัญก็คือ A/D ของตัวนั้นต้องมีค่าสูง RESOLUTION สูง

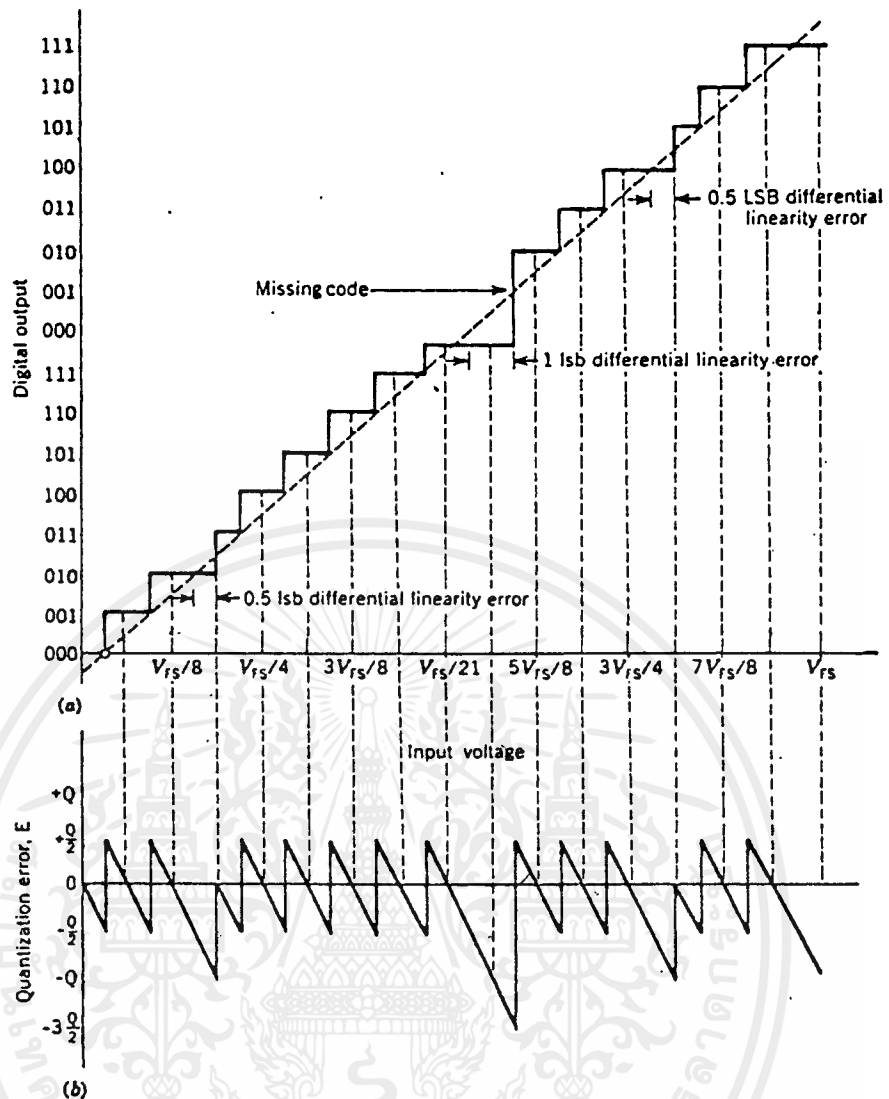
2. LINEARITY

ความเบี่ยงเบนสูงสุดจากเส้นตรง ระหว่างปลายทั้งสองข้างของเส้น Transfer Curve ซึ่งจะเป็นฟังก์ชันกับขนาดของ LSB ค่า Linearity ของคอนเวอร์เตอร์ที่ดีเป็น $1/2$ LSB สำหรับค่า LINEARITY ERROR แสดงได้ด้วย DIFFERENTIAL NON-LINEARITY และ INTEGRAL NONLINEARITY

DIFFERENTIAL NON-LINEARITY เป็นการวัดความเป็นเชิงเส้น ของสถานะสัญญาณดิจิทัล state หนึ่งไปยังอีก state ถัดไป ขนาดของ step จาก state หนึ่งไปยังอีก state ถัดไปอาจจะเริ่มจากค่า $1/2$ ไป $3/2$... เป็นขนาด 1 LSB สำหรับ DIFFERENTIAL LINEARITY จะกำหนดที่ค่า $\pm 1/2$ LSB จากรูป

สำหรับขนาดคอนเวอร์ชันขนาด n bit ค่าการแปลงระหว่างสัญญาณอนาล็อกไปเป็นดิจิทัล ควรเป็นปริมาณที่สอดคล้องตรงกันจริง ๆ ของอินพุตกับเอาต์พุต ถ้าค่าการวัดเบี่ยงเบนไปจากค่า IDEAL ก็ถือว่าเป็น DIFFERENTIAL NON-LINEARITY

ไม่ทราบว่าใครได้ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 1 DIFFERENTIAL LINEARITY

ค่า DIFFERENTIAL NON-LINEARITY นี้เป็นข้อกำหนดที่สำคัญยิ่งเพราะว่า ถ้ามี error จากค่านี้มากกว่า 1 LSB ก็สามารถหาบรู๊ปรากการผล MONOTONIC ของ D/A คอนเวอร์เตอร์ทำให้เกิดความผิดพลาดในการแปลงค่าทั้งหมดที่ออกมาที่เอาต์พุต

INTERGREL NON-LINEARITY ค่า DIFFERENTIAL NON-LINEARITY เราทราบว่าคือการผิดพลาดในแต่ละ STEP แต่ถ้าเป็น INTEGRAL NON-LINEARITY คือคือการเบี่ยงเบนทั้งหมดที่ถูกเปลี่ยนมา

3. ACCURACY

แบ่งเป็น ABSOLUTE ACCURACY หรือ RELATIVE ACCURACY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE ACCURACY คือ ค่าแตกต่างระหว่าง แรงดันเอาต์พุต ที่ได้จากแรงดันอินพุตทางทฤษฎี กับแรงดันเอาต์พุตที่ได้จากการบ่อนแรงดันอินพุตจริง ๆ ประกอบด้วย GAIN ERROR, ZERO ERROR NON-LINEARITY รวมทั้งพวกสัญญาณรบกวน ดูรูป 2

RELATIVE ACCURACY คือการวัดเป็น % PPM หรือ ทศนิยมของ LSB เป็นค่าที่เบี่ยงเบนไปจากค่าจริงโดยสัมพันธ์ กับค่า Full scale ของ Transfer characteristic หลังจากได้รับการปรับแต่งแล้ว

4. CONVERSION TIME

เป็นช่วงเวลาที่ A/D ใช้จ่ายในการเปลี่ยนสัญญาณอนาล็อก เป็นดิจิทัลจนเสร็จในแต่ละครั้ง ซึ่งในคอนเวอร์เตอร์ความเร็วสูง ช่วงเวลาการเปลี่ยนควรมีน้อยมาก ๆ สำหรับคอนเวอร์เตอร์ขนาด 8 บิต จะใช้เวลาในการเปลี่ยนเพียง 1 คาบ สัญญาณนาฬิกาเท่านั้น

5. EXTERNAL INPUT IMPEDANCE

ค่าอิมพีแดนซ์ของ SOURCE หรือ CIRCUIT ตลอดจน DEVICE ภายนอกที่จะถูกนำมาต่อกับ OUTPUT ของ A/D หรือ D/A ซึ่งอาจจะกำหนดกระแสที่เอาต์พุตขณะใช้งานควรมีค่าสูง

EXTERNAL OUTPUT IMPEDANCE ค่าอิมพีแดนซ์ทางเอาต์พุตของ SOURCE ที่จะนำมาบ่อนเข้า ที่อินพุต ของ A/D หรือ D/A

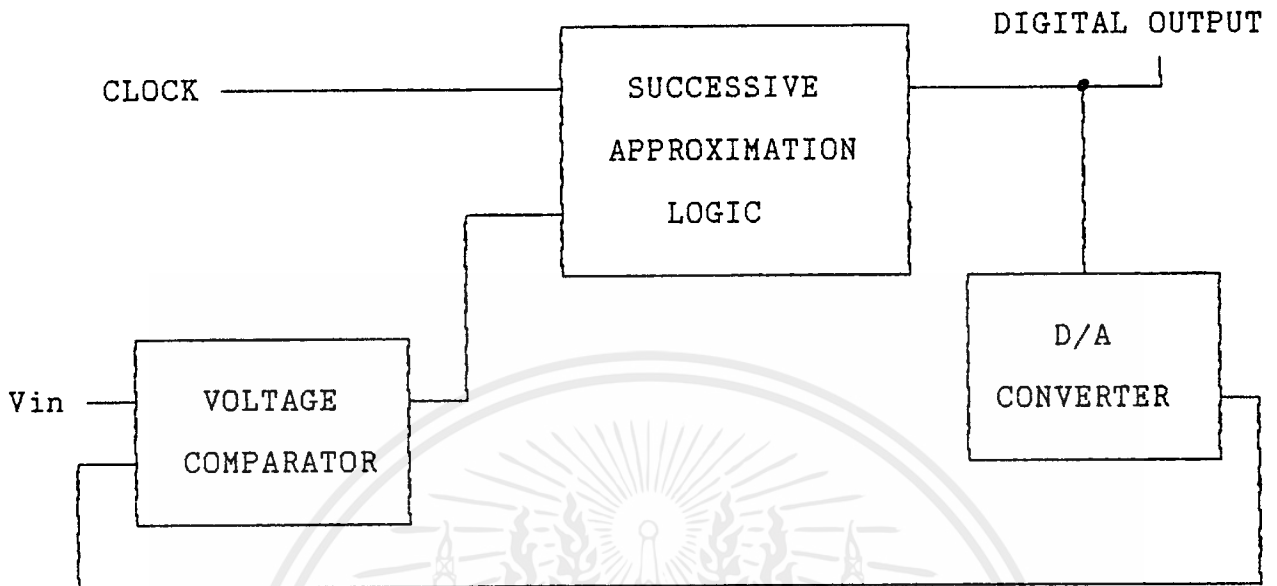
RESOLUTION (STEP SIZE) คือการเปลี่ยนแปลงค่าสุดของสัญญาณอนาล็อก จากระดับแรงดันหนึ่ง เมื่อสัญญาณอินพุตเปลี่ยนแปลงไป

การแปลงสัญญาณ ADC มี 5 ชนิดคือ

1. วงจร ADC แบบเคาน์เตอร์-แรมพ์ (COUNTER RAMP ADC)
2. วงจรเปรียบเทียบแรงดัน (VOLTAGE COMPARATOR)
3. วงจรลิเนียร์-แรมพ์ (LINEAR RAMP ADC)
4. วงจร SUCCESSION APPROXIMATION ADC
5. วงจร INTEGRATING A/D CONVERTER (DUAL-SLOP A/D CONVERTER)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจร SUCCESSIVE APPROXIMATE A/D



รูป ผังโคะแกรมของวงจร SUCCESSIVE APPORXIMATE A/D

การทำงานตามรูปแผนภูมิในรูป อธิบายได้ดังนี้ สมมุติว่าแรงดันอนาล็อกที่อินพุตเท่ากับ 7 โวลต์ วงจร SUCCESSIVE APPORXIMATION LOGIC จะต้องมีบิต MSB เป็นลอจิก "1" หากได้รหัสเบนารี 1000 บิตผ่านวงจร D/A ไปยังอินพุตของวงจร เปรียบเทียบแรงดัน ซึ่งจะเปรียบเทียบว่าแรงดันเปรียบเทียบ ที่บิตกลับสูงหรือต่ำกว่าแรงดัน ที่อินพุต ถ้าแรงดันเปรียบเทียบสูงกว่า วงจร SUCCESSIVE APPROXIMATION LOGIC ก็จะเคลียร์บิต MSB หรือบิต 2^3 ให้เป็นลอจิก 0 แล้วเซตบิตถัดไปคือบิตที่ 2^2 ให้เป็นลอจิก 1 ผลที่ได้คือรหัส 0100 ซึ่งจะเปลี่ยนเป็นแรงดันบิตกลับไปเปรียบเทียบกับแรงดันอินพุตอีก

คราวนี้ ถ้าแรงดันเปรียบเทียบต่ำกว่า แรงดันอินพุตวงจรลอจิกก็จะเซตบิตถัดไป คือบิตที่ 2^1 ให้เป็นลอจิก 1 ผลที่ได้คือรหัส 0110 แล้วนำไปเปรียบใหม่ ถ้าแรงดันเปรียบเทียบยังต่ำกว่าแรงดันที่อินพุตอีก วงจรลอจิกก็จะเซตบิตถัดไปคือบิตที่ 2^0 ให้เป็นลอจิก 1 ผลที่ได้คือรหัส 0111 เมื่อนำไปเปรียบเทียบกับแรงดันที่อินพุต บวกกว่า เท่ากัน วงจรเปรียบเทียบก็จะควบคุมให้วงจร SUCCESSIVE APPORXIMATION หยุดการทำงาน ข้อมูลรหัสเบนารี ที่อยู่ในวงจรลอจิกควบคุมก็คือ 0111 ซึ่งจะเป็นรหัสเบนารีแทนค่าแรงดันอนาล็อก ที่อินพุต 7 โวลต์นั่นเอง

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อดีของวงจร A/D แบบนี้ก็คือ จะใช้เวลาในการเปลี่ยนสัญญาณเร็วมาก เพราะไม่ต้องใช้วิธีนับเรียงลำดับ มาเรื่อยๆ เหมือนกับวงจร A/D แบบอื่น ๆ จึงเป็นที่นิยมมาใช้อย่างกว้างขวาง

ADC 0809

ADC 0809 เป็นอุปกรณ์ที่แปลงอนาลอกเป็นดิจิทัล 8 บิต โดยมีอินพุต 8 CHANNEL แบบ MULTIPLEX ใช้งานหลักการ SUCCESSIVE APPROXIMATION ซึ่งประกอบด้วย HIGH IMPEDANCE CHOPPER STABILIZED COMPARATOR 256R VOLTAGE DIVIDER WITH ANALOG SWITCH TOCC AND SUCCESSIVE APPROXIMATION REGISTER ในการปรับค่า A/D จะปรับได้ตั้งแต่ 0 ถึง FULLSCALE ซึ่งง่ายต่อการเข้ากับไมโครโปรเซสเซอร์ ที่ใช้โหมด Latch และ DECODED MULTIPLEXER ADDRESS INPUT และการ Latch TTL-STATE OUTPUT ในการออกแบบ A/D ชนิดนี้ ได้รวมคุณสมบัติที่ดีหลายอย่าง เช่น มีความเร็วสูงในการ CONVERT HIGH ACCURACY

การใช้งาน

ADC ชนิดนี้เป็นแบบ SUCCESSIVE APPROXIMATION REGISTER (SAR) เมื่อเริ่มมาจะถูก START ด้วย PULSE (START CONVERSION) ระดับสูง และสัญญาณ อินเทอร์รัพท์จะออกจากขา END OF CONVERSION (EOC) เพื่อให้ไมโครโปรเซสเซอร์ ส่งสัญญาณ START มา START ในการทำงานขั้นตอน ส่วนสำคัญของ A/D ชนิดนี้คือส่วน COMPARATOR ซึ่งจะเป็นส่วนที่ควบคุมให้ได้ค่า ACCURACY ที่ดี ในการ CONVERT

บทที่ 3

การทำงานของอุปกรณ์ควบคุมระบบไฟฟ้า 3-เฟส 50 Hz

การทำงานของระบบควบคุมระบบไฟฟ้า 3 เฟส 50 Hz ด้วยไมโครคอนโทรลเลอร์ เป็นวงจรที่ควบคุมระบบไฟฟ้า เมื่อระบบไฟฟ้า 3 เฟส มีอาการผิดปกติคือ

1. แรงดันระหว่างสาย ไม่เท่ากัน
2. เฟสใดเฟสหนึ่งขาดหายไป (OUT OF PHASE)
3. การเรียงลำดับเฟส (PHASE SEQUENCE)

เมื่อมีอาการผิดปกติดังกล่าว ไมโครคอนโทรลเลอร์จะควบคุมให้รีเลย์ ตัดกระแสที่เข้าไปยังขดลวดของ magnetic contractor แต่ระบบไฟฟ้าไม่มีอาการผิดปกติ รีเลย์ก็จะต่อ ให้กระแสไฟฟ้าผ่าน magnetic contractor ทำงานตามปกติ

ขั้นตอนการทำงาน

1. หม้อแปลงลดแรงดันไฟฟ้า

สายแต่ละสายจากจุดต่อจะต่อเข้า Line ของหม้อแปลงแต่ละตัว คือมีแรงดันเข้า 220 โวลต์ และแรงดันออก 12 โวลต์ สายแต่ละสาย ที่ออกจากหม้อแปลง จะผ่าน ไดโอด เพื่อป้องกันการไหลย้อนกลับของกระแส แล้วทำการรวมสายที่ออกจาก ไดโอด เข้าด้วยกัน เพื่อป้องกันเมื่อเฟสใดเฟสหนึ่ง ไม่มีแรงดัน แต่ก็ยังมีไฟในวงจรแหล่งจ่ายไฟ และขั้วเอาต์พุตที่เหลือของหม้อแปลง ต่อเข้าอินพุต Multiplex ของ A/D

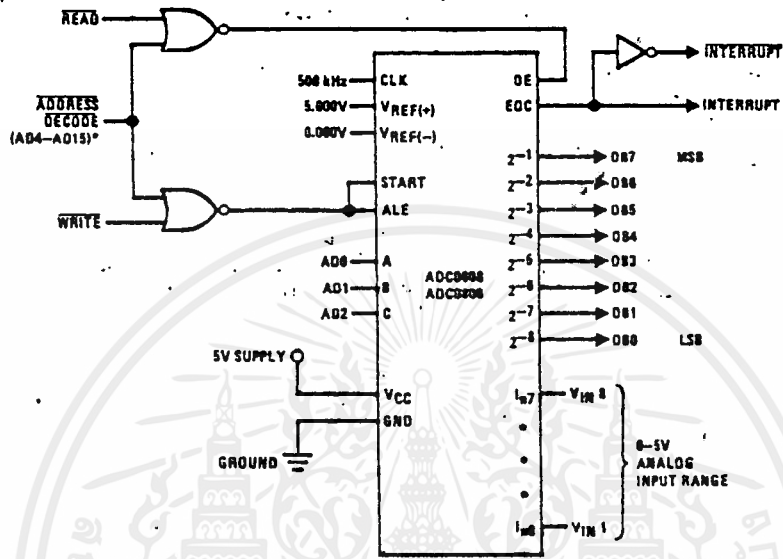
2. วงจรแหล่งจ่ายไฟ

ใช้ IC#LM7805 เป็น IC ขนาด 500 mA ทำงานที่อุณหภูมิตั้งแต่ 0-70 °C ความถี่ตั้งแต่ 10 Hz-100 KHz โดยมีแรงดัน 7.5-20 โวลต์ ช่วงความแตกต่างแรงดันเอาต์พุต 4.8-5.2 โวลต์

แรงดันทางอินพุตของ LM#7805 ได้จากวงจร Rectifier โดยมีค่าประมาณ 18 Vdc มี Capacitor 4700 μ F ทำหน้าที่กรองกระแส และ Capacitor ค่า 0.01 μ F ทำหน้าที่ bypass high frequency ค่าแรงดันเอาต์พุตมีค่าประมาณ 5 โวลต์ โดยมี Capacitor 10 μ F เป็นตัวกรองกระแสทางเอาต์พุตนี้ จะทำหน้าที่จ่ายแรงดันให้กับ IC#ADC0809 IC#MCS-48 IC ที่ใช้ทุกตัวในวงจร และ Seven Segment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใด ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. ตัว เปลี่ยนสัญญาณอนาล็อก เป็นสัญญาณดิจิทัล



IC#ADC0809 ทาหน้าที่แปลงสัญญาณอนาล็อก ใหัเป็นสัญญาณดิจิทัล ที่มีอินพุต 8 แชนแนล แบบมัลติเพล็กซ์ ช่วงเวลาในการ เปลี่ยนสัญญาณ 100 ๕SEC 8 บิตเอาต์พุต ให้ความถี่นาฬิกา 50 KHz

เมื่อเราทาการต่อ ADC0809 เข้ากับไมโครคอนโทรลเลอร์ การควบคุม ADC นั้นตัวไมโครคอนโทรลเลอร์จะเป็นตัวควบคุม โดยการควบคุมนั้นจะควบคุมการเลือก แชนแนล พร้อมทั้งส่งสัญญาณควบคุมการทำงานใหักับ ADC และรอรับสัญญาณอินเตอร์รัพท์ แล้วทาการตอบสนองการอินเตอร์รัพท์ โดยการอ่านค่าสัญญาณเอาต์พุตของ ADC ที่เป็น คิจิตอล เข้ามาทางบัสข้อมูล แล้วนำมาประมวลผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

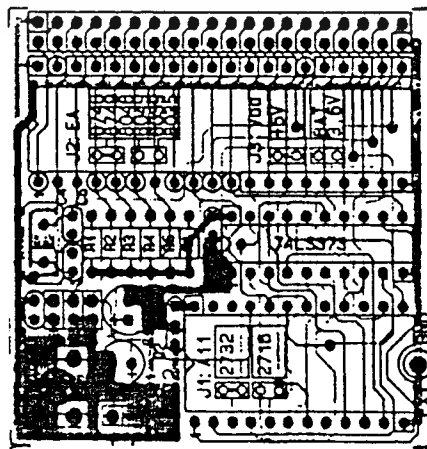
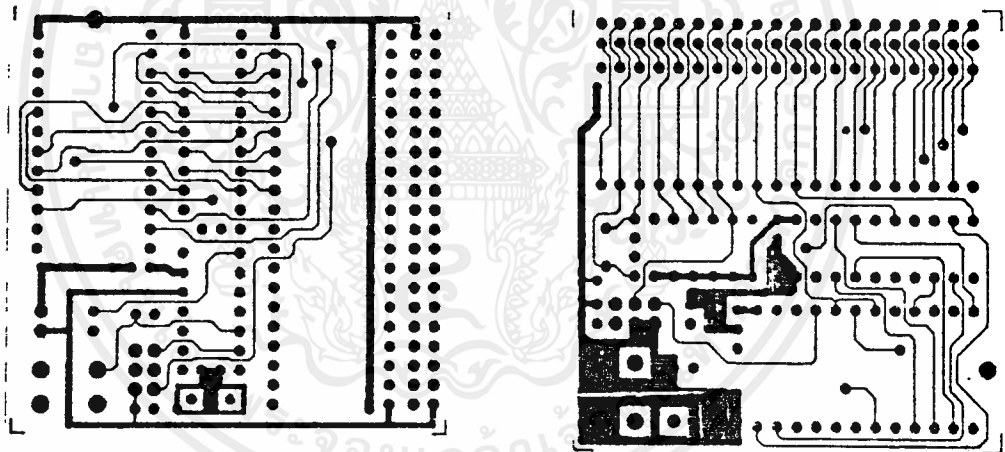
4. รีเลย์ (RELAY)

จะเป็นตัวตัดวงจรเมื่อไฟฟ้าเกิดปัญหา รีเลย์รีเลย์มีคุณสมบัติ แรงดันคอยล์ 12 โวลต์ หน้าสัมผัสทนกระแสได้ 5 แอมป์ 220-240 Vac ซึ่งรีเลย์นี้จะควบคุมแมคเน็ตคัล คอนแท็คเตอร์อีกทีหนึ่ง ในระบบไฟฟ้า

5. ส่วนแสดงผล (Display) มี 2 แบบคือ

1. Seven Segment จะเป็นการแสดงค่า จากการประมวลผลของ ไมโครคอนโทรลเลอร์ เมื่อค่าที่เ้าจากการประมวลผลถูกต้องคือ ระบบไฟฟ้าไม่ผิดปกติ จะแสดงค่าแรงดันของ Line นั้น
2. เ้าแสดงผลเมื่อแรงดันไม่เท่ากัน เฟสผิดเฟสหนึ่งหาย หรือเฟสไม่ Sequence ก็จะมี LED เป็นตัวแสดงของแต่ละตัว

6. IC#MCS-48

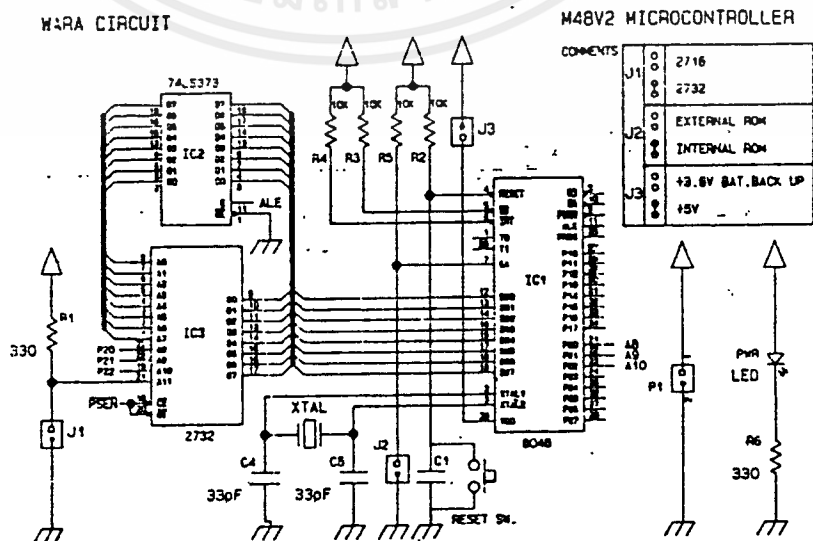


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราใช้ไมโครคอนโทรลเลอร์ชิพเดี่ยว ในการควบคุมนั้นคือใช้ SINGLE SHIP เบอร์ 8049 เป็นตัวเบรเซสเซอร์ โดยต่อคริสตอลให้โดยตรง 6 MHz แล้วใช้หน่วยความจำ โดยใช้การขยายหน่วยความจำภายนอก โดยขยายออกไปอีก 2 K คือใช้ IC 2716 โดยใช้ IC# 74LS373 เป็นตัวแลทซ์ค่าของแอดเดรส โดยใช้สัญญาณจากขา ALE บอกรหัสการแลทซ์ ให้แลทซ์ให้แอดเดรสบัส (ADDRESS BUS) แยกออกจากคาคิวบัส โดยให้เป็นแอดเดรสบัส A₀-A₇ ส่วนแอดเดรสบัส A₈-A₁₁ นั้นได้จาก 4 บิตล่าง ของพอร์ท 2 ของ 8049 คือขา P₂₀-P₂₃

ส่วนสัญญาณ PSEN จาก 8049 ไปต่อกับ IC2716 เป็นสัญญาณบอกให้รู้ว่า 8049 กำลังเพทคำสั่งจากหน่วยความจำเบรแกรมภายนอก ซึ่งสัญญาณนี้เป็นตัว ENABLE หน่วยความจำภายนอก

สำหรับพอร์ท 1 ของ 8049 คือ P₁₀-P₁₇ ใช้ในการขับให้กับ SEVEN SEGMENT เพื่อทำการจุด SEVEN SEGMENT โดยจ่ายให้แต่ละบิตเท่ากับ 0 จะเป็นการจุดให้สว่าง P₂₀-P₂₂ ใช้ช่วยในการ SCAN และเปลี่ยนค่าสภาวะในการแสดงผลของ T₀ หรือ T₁ ก็จะเป็น 0 ตัวใดตัวหนึ่งทำให้สภาวะได้ว่า คีย์คกดเข้ามาได้จาก T₀ หรือ T₁ ซึ่งในวงจรนี้ 74LS138 จะทำการ SCAN KEY พร้อมกับการ SCAN ให้ SEVEN SEGMENT ด้วย โดยไม่ขับให้ทรานซิสเตอร์อีกครึ่งหนึ่ง ส่วนในการใช้หน่วยความจำแรม จะใช้แรมในตัวของ 8049 ซึ่งมีอยู่ในตัวแล้ว



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

TURBO 48

ส่วนประกอบของซอฟต์แวร์

TURBO 48 นี้เป็นซอฟต์แวร์รวมคือมี EDITOR, ASSEMBLER และ DEBUG ในตัวเดียวกัน การใช้งานเป็นแบบตระกูล turbo ของบอร์แลนด์ คือมีลักษณะเป็นพลูดาวน์เมนู แต่เมนูไม่มากเหมือนกับ turbo ของ บอร์แลนด์ ซึ่งอาจทำให้สับสนได้ ดังนั้นจึงเป็นซอฟต์แวร์ที่ใช้งานง่ายมาก ๆ

- EDITOR เป็นส่วนที่ทำหน้าที่สร้างและแก้ไขโปรแกรม ทำให้มีความสะดวก เนื่องจากสามารถที่จะทำการแทรกข้อความ หรือลบข้อความที่ผิดได้ การใช้งานของ EDITOR เป็นลักษณะของ full screen editor
- ASSEMBLER เป็นส่วนที่ทำหน้าที่แปลคำสั่งจาก source program (ภาษาแอสเซมบลี) ไปเป็น object program (ภาษาเครื่อง) โดย object program เมื่อแปลแล้วจะถูกนำไปเก็บไว้ในส่วนของ DEBUG .
- DEBUG เป็นส่วนที่ทำหน้าที่ตรวจสอบข้อบกพร่องต่าง ๆ ของโปรแกรม โดยสามารถ RUN ได้ทีละคำสั่ง และสามารถแสดงค่า รีจิสเตอร์, พอร์ต อินพุต-เอาต์พุต รวมทั้งโปรแกรมนี้จะใช้หลักการ software emulator

คุณสมบัติของ TURBO 48

- ทำงานบนเครื่อง IBM PC หรือเครื่อง compatible ที่มีหน่วยความจำอย่างน้อย 256 KB ใช้ได้กับจอภาพทุกชนิด
- ทำงานภายใต้ DOS 2.0 ขึ้นไป
- โปรแกรมเขียนด้วยภาษาแอสเซมบลี ของ 8088\8086

ความสามารถของ EDITOR

- เป็น full screen edit
- source file เป็นรหัส ASCII ยาวไม่เกิน 65536 ตัวอักษร (64 กิโลไบท์)
- ความยาวของคอลัมน์ได้ไม่เกิน 255 ตัวอักษร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
- การใช้คีย์ควบคุมและเลือกฟังก์ชันต่าง ๆ เหมือนกับของโปรแกรม SIDEKICK
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสามารถของ ASSEMBLER

- มี Pseudo DB (Define Byte), DW (Define Word), DS (Define Storage), ORG (Origin), END (End program)
- ในการคำนวณ ำตัวเลขเป็นแบบ Integer ซึ่งมีค่าได้ตั้งแต่ 0-65535 ในฐานสิบ หรือ 0-1111111111111111B ในฐานสอง หรือ 0-FFFFH ในฐานสิบหก หรือ 0-1777770 ในฐานแปด (octal) และสามารถรับรหัส ASCII ได้ด้วย เช่น 'ABCABC\$_+-*123'
- OPERATOR ในการคำนวณ
 - + (addition)
 - (subtraction)
 - * (multiplication)
 - / (division)
 - MOD (modulus)
 - AND (logical AND)
 - OR (logical OR)
 - XOR (logical Exclusive OR)
 - EQ (comparision equal)
 - GE (comparision Greater than or Equal)
 - GT (comparision Greater than)
 - LE (comparision Less than or Equal)
 - LT (comparision Less than)
 - () (parenthesis)
- สามารถรับ label ได้ยาว 10 ตัวอักษร ำยจะต้องถูกต้องตามหลักการตั้งชื่อ ด้วยคือตัวอักษรตัวแรกจะต้องเป็นตัวอักษรเท่านั้น ตัวต่อไปเป็นตัวเลขหรือตัวอักษรก็ได้ ทั้งหมดไม่เกิน 10 ตัว และไม่มีเครื่องหมายใด ๆ ยกเว้น "_"
- สามารถรับ label ได้ 255 ตัว
- ความยาวของ source file ในแต่ละบรรทัดได้มากที่สุด 80 ตัวอักษร

ข้อบังคับบางประการของ ASSEMBLER

- label จะต้องมีคำด้วยเครื่องหมาย colon (:) เสมอ เช่น

ADDRESS:MOV A,#DATA

- variable (หรือตัวแปร) จะต้องมีคำด้วยเครื่องหมาย (:) เช่นกัน เช่น

DATA:EQU127Q

ความสามารถของ DEBUG

DEBUG นี้เป็นการจำลองคำสั่งของ 8048 ทุกคำสั่ง สามารถรันและทดสอบโปรแกรมได้ใน DEBUG มีหน่วยความจำที่เก็บโปรแกรมได้ทั้งหมด 4 KB หน่วยความจำข้อมูลภายใน 256 ไบต์ และหน่วยความจำข้อมูลภายนอก 256 ไบต์ 8048 มีเพียง 64 ไบต์ ทั้งนี้เนื่องจากในตระกูล 8048 นั้นเบอร์ 8050 และเบอร์ 8040 มีหน่วยความจำข้อมูลภายใน 256 ไบต์

พอร์ทของ 8048 ใน DEBUG นี้สามารถนำมาแสดงและแก้ไขได้ส่วนรีจิสเตอร์ต่าง ๆ มีดังนี้คือรีจิสเตอร์ A, รีจิสเตอร์ R₀-R₇, รีจิสเตอร์ T (TIME), STACK และโปรแกรมเคาน์เตอร์ (PC)

แฟลก หรือ PSW (program status word) มีดังนี้คือ แบนด์รีจิสเตอร์ (BS), แฟลก O(FO), แฟลกทดช่วย(AC), แฟลกตัวทด(CY) และแฟลก 1(F1) ทัตเมอร์ฟิลิปฟลอป(TF) และอินเทอร์รัพท์ฟิลิปฟลอป(IF) นอกจากนี้ DEBUG ยังจำลองการทำงานที่เกี่ยวกับ timer/counter ด้วยโดยเมื่อพบคำสั่ง STRT T จะนำค่าในรีจิสเตอร์ T เข้าไปใน T/CNT และจะนับจน T/CNT นับจากค่า 0FFH ไปเป็น 0H แล้ว T/CNT จะส่งสัญญาณอินเทอร์รัพท์ โปรแกรมจะกระโดดไปที่แอดเดรส 007H ทันที

การเริ่มใช้ TURBO 48

เมื่อจะเริ่มรัน TURBO 48 ให้เรียกไฟล์ T48.EXE ซอฟต์แวร์จะเริ่มทำงาน โดยแสดงที่จอภาพจากนั้นให้เลื่อนเมนูไปที่ FILE โดยใช้คีย์ลูกศรซ้าย-ขวา จะเห็นเมนู FILE และมีเมนูย่อยให้เลือกใช้ ให้เลื่อนคีย์ลูกศรขึ้นลงไปที่ NEW FILE แล้วกดคีย์ ENTER แล้ว TURBO 48 จะถามชื่อไฟล์ที่ต้องการแก้ไขหากมีไฟล์อยู่แล้วให้กดคีย์ ENTER ได้เลข TURBO 48 จะแสดงชื่อไฟล์ต่าง ๆ ให้เลือก ซึ่งจะเลือกได้โดยเลื่อนลูกศรเมื่อตรงกับชื่อไฟล์ที่ต้องการแล้วก็ให้กดคีย์ ENTER จากนั้น TURBO 48 จะโหลดโปรแกรมลงไปที่ EDITOR จากนั้นก็เลื่อนเมนูลงไปที่ EDIT แล้วกด ENTER จึงจะเข้าไปที่ EDITOR

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้คีย์ต่าง ๆ ของ EDITOR จะเหมือนกับของโปรแกรม SIDEKICK ในขณะที่อยู่ใน EDITOR เมื่อต้องการย้ายฟังก์ชันต่าง ๆ ใน FILE เมนู ให้ใช้คีย์ฟังก์ชัน (F1-F10) ความที่แสดงในบรรทัดล่างสุด และเมื่อทำการเขียนหรือแก้ไขโปรแกรมเรียบร้อยแล้วจะออกจาก EDITOR ให้ไปใช้คีย์ Esc เพื่อกลับเข้าเมนู

การ ASSEMBLER

เมื่อต้องการแปลโปรแกรมที่เขียนโดย EDITOR ให้เป็นภาษาแอสเซมบลี ให้เลือกประเภทของไฟล์ที่ต้องการ ซึ่งมี 3 ประเภท คือ

1. HEX เป็นไฟล์ที่มีรหัส ASCII อยู่ในเลขฐาน 16 และมีนามสกุลเป็น .HEX
2. LIST เป็นไฟล์ที่ใช้สำหรับพิมพ์ หรือตรวจสอบ และมีนามสกุลเป็น .LST
3. BIN เป็นไฟล์ภาษาเครื่อง ซึ่งสามารถนำไฟล์นี้ ไปโปรแกรม EPROM ได้โดยตรง และมีนามสกุลเป็น .BIN

ในการเลือกประเภทของไฟล์ให้เลื่อนเมนูขึ้น-ลง ไปยังประเภทที่ต้องการแล้ว กด ENTER ซึ่งเป็นลักษณะที่อกเกิดสวิทช์ คือ กด ENTER 1 ครั้ง ON และกดอีกครั้ง OFF ON หมายถึงต้องการให้ SAVE ไฟล์ที่ต้องการให้ลงแผ่นดิสก์ ส่วน OFF หมายถึงไม่ต้อง SAVE

เมื่อเลือกไฟล์ที่ต้องการได้แล้ว ให้เลื่อนเมนูขึ้นไปตำแหน่ง Assembler แล้วกดคีย์ ENTER โปรแกรมจะทำงานแปลทันที หาก source โปรแกรมมีที่ผิดโปรแกรมจะหยุด แปลทันที และบอก เหตุที่ผิดหรือมีบรรทัดที่ผิด โดยแสดงไว้ที่ช่อง message เมื่อกดคีย์ใด ๆ โปรแกรมจะวิ่งเข้าไปใน EDITOR และเลื่อน cursor ไปยังบรรทัดที่ผิด

เมื่อเข้า DEBUG

เมื่อเขียนโปรแกรม และแปลเรียบร้อยแล้ว แต่ยังไม่ทราบว่าโปรแกรมนั้นผิดส่วนไหนบ้าง การตรวจโปรแกรมต้องเข้าไปตรวจสอบใน DEBUG แล้วเลื่อนเมนูไปที่ DEBUG แล้วกดคีย์ ENTER จะเข้าไปในโปรแกรม DEBUG ทันที โดยจะมีเครื่องหมายแสดงความพร้อม ("____") เมื่อเริ่มเข้าให้กด " ? " และกด ENTER โปรแกรมจะแสดงเมนู HELP ของคำสั่งต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การตรวจการเรียงลำดับเฟส

สรุปขั้นตอนการตรวจการเรียงลำดับเฟส

1. ยึดเฟสใดเฟสหนึ่งเป็นเฟสอ้างอิง
2. เฟสที่เหลืออีก 2 เฟส เฟสหนึ่งเลื่อนเฟสให้ล่าช้าลงไป 120 องศา และอีกเฟสหนึ่งเลื่อนให้นำหน้าไป 120 องศา
3. นำแรงดันทั้ง 3 เฟสมารวมเข้าด้วยกัน

หลักการของการตรวจการเรียงลำดับเฟสก็คือ การนำเอาวงจรอิเล็กทรอนิกส์ไปทำงานแทนส่วนต่าง ๆ ทั้ง 3 ขั้นตอนแล้วนำผลที่ได้ไปควบคุมการทำงานของสวิตช์ให้เป็นไปตามที่ต้องการและส่งผลไปยังภาคแสดงผล

ส่วนประกอบของการตรวจการเรียงลำดับเฟส

1. วงจรลดแรงดันไฟฟ้า (VOLTAGE DIVIDER)

เนื่องจากการตรวจการเรียงลำดับเฟสประกอบไปด้วยวงจรอิเล็กทรอนิกส์ซึ่งใช้กับแรงดันไฟฟ้าค่าต่ำ ๆ แต่แรงดันไฟฟ้าที่ใช้ในโรงงานอุตสาหกรรมนั้นเป็นระบบ 3 เฟส 380/220 ซึ่งไม่อาจนำไปใช้กับวงจรอิเล็กทรอนิกส์โดยตรงได้ต้องลดแรงดันไฟฟ้าให้ต่ำลงเสียก่อนโดยใช่วงจรลดแรงดันไฟฟ้า สัญญาณไฟฟ้า ที่ออกจาก วงจรลดแรงดันไฟฟ้า จะมีขนาดลดลงโดยที่แรงดันไฟฟ้าสูงสุดเหลือประมาณ 3-4 โวลต์ ส่วนเฟสและรูปร่างของสัญญาณไฟฟ้ายังคงเหมือนเดิม

2. วงจรเลื่อนเฟสล่าช้า 120 องศา (PHASE SHIFTER 120° LAGGING)

เป็นวงจรที่ทำหน้าที่เลื่อนเฟสให้ล่าช้าไปจากเดิม 120 องศา โดยที่รูปร่างและขนาดของสัญญาณยังคงเหมือนเดิม

3. วงจรเลื่อนเฟสนำหน้า 120 องศา (PHASE SHIFTER 120° LEADING)

การทำงานคล้ายกับวงจรเลื่อนเฟสล่าช้าต่างกันที่ว่าวงจรเลื่อนเฟสล่าช้า จะมีสัญญาณเอาต์พุทล่าช้าสัญญาณอินพุทแต่วงจรเลื่อนเฟสนำหน้าจะมีสัญญาณนำหน้าสัญญาณอินพุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. วงจรรวมสัญญาณ(SUMMING AMPLIFIER)

วงจรนี้มีอินพุตอยู่ 3 อินพุต และมีเอาต์พุตอยู่เอาต์พุตเดียว โดยอินพุตที่ 1 จะต่อรับสัญญาณจากวงจรเลื่อนเฟสหน้า อินพุตที่ 2 จะต่อกับวงจรเลื่อนเฟสล้าหลังและอีกอินพุตหนึ่งจะต่อกับสัญญาณที่ใช้เป็นเฟสอ้างอิง สัญญาณที่เข้าสู่อินพุตทั้ง 3 อินพุตนี้จะถูกรวมเข้ากัน โดยวงจรรวมสัญญาณเป็นสัญญาณเดียวและสัญญาณนี้จะถูกขยายในที่นี้ให้อัตราขยายเท่ากับ 1

5. วงจรเปรียบเทียบ(COMPARATOR)

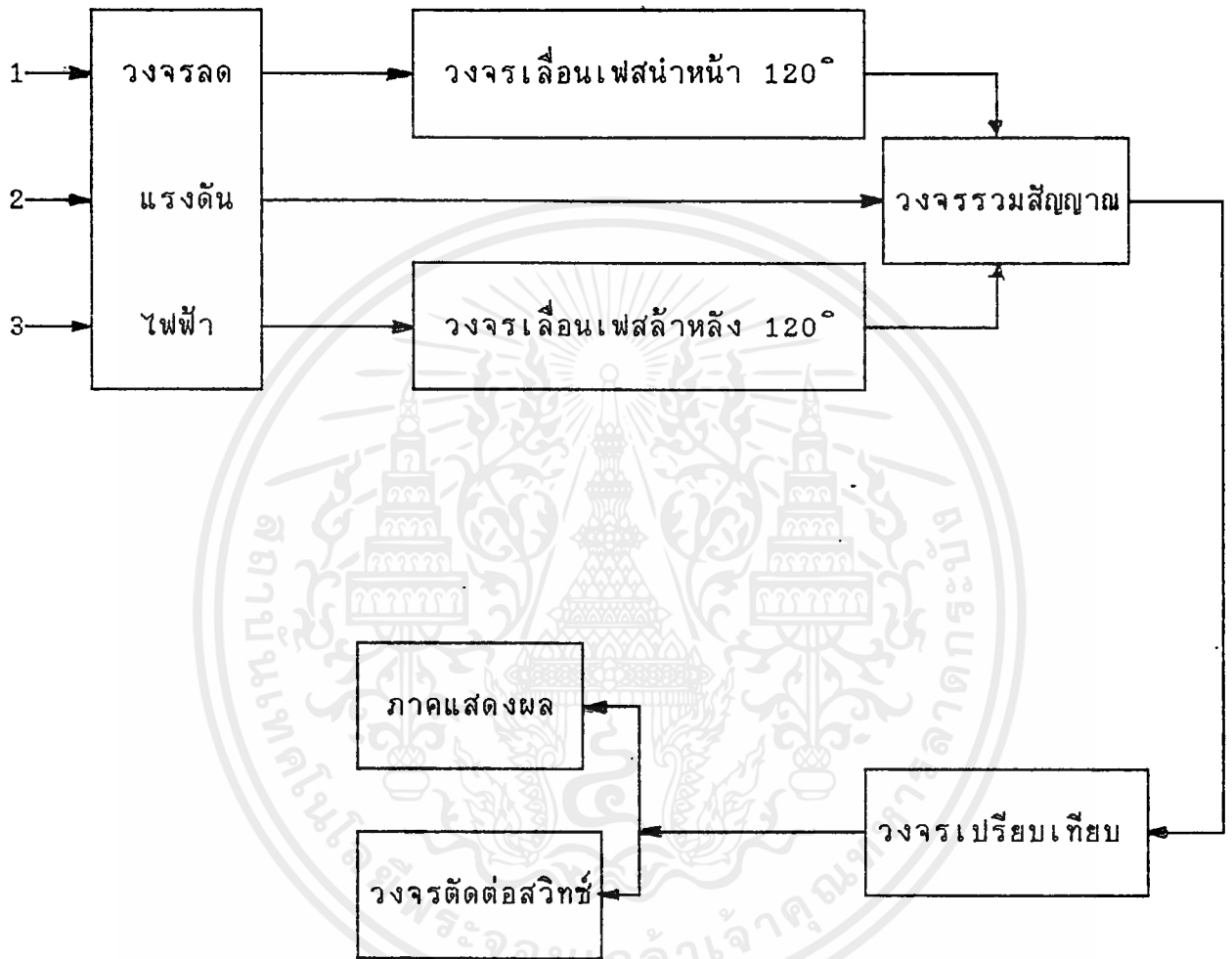
วงจรเปรียบเทียบเป็นวงจรที่ประกอบด้วยศักดาไฟฟ้าอ้างอิง ที่ตั้งที่ค่าหนึ่งวงจรนี้จะทำหน้าที่นำเอาสัญญาณอินพุตมาเปรียบเทียบกับศักดาไฟฟ้าอ้างอิง ถ้าสัญญาณไฟฟ้าอินพุตมีค่าสูงกว่าหรือต่ำกว่าศักดาไฟฟ้าอ้างอิง จะทำให้สัญญาณเอาต์พุตมีค่าเป็น 0

6. วงจรตัดต่อสวิทช์(SWITCH CONTROLLER)

เป็นวงจรที่รับสัญญาณจากวงจรเปรียบเทียบ เมื่อสัญญาณเอาต์พุตจากวงจรเปรียบเทียบเป็น 0 วงจรนี้จะทำหน้าที่ให้สวิทช์ตัดวงจรและเมื่อสัญญาณเอาต์พุตจากวงจรเปรียบเทียบมีค่าเท่ากับ Vcc วงจรนี้จะทำหน้าที่สวิทช์ต่อวงจร

7. ภาคแสดงผล(DISPLAY)

ภาคแสดงผลจะแสดงให้ทราบ อย่างคือทราบถึงจำนวนเฟส และทราบถึงระดับเฟส -แสดงจำนวนเฟส การแสดงจำนวนเฟสจะใช้หลอดไฟ 3 หลอด เป็นตัวแสดงผลของแต่ละเฟส ถ้าแรงดันไฟฟ้ามาครบทั้ง 3 เฟส หลอดไฟฟ้าทั้ง 3 หลอดก็จะสว่างหมด แต่ถ้าแรงดันไฟฟ้าเฟสหนึ่งเฟสใดขาดหายไปหลอดไฟของเฟสนั้นก็จะไม่สว่าง



รูปที่ 5.1 แสดงส่วนประกอบของการทำงานการตรวจการเรียงลำดับเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปการทำงานของเครื่องตรวจการเรียงลำดับเฟส

1. กรณีแรงดันไฟฟ้ามาตรฐานทุกเฟส

- ลำดับเฟสเป็น ABC หรือ BCA หรือ CAB ต่อสัญญาณแสดงจำนวนเฟสสว่างหมดทั้ง

3 หลอด สวิตช์จะต่อวงจร

- ลำดับเฟสเป็น ACB หรือ CBA หรือ BAC หลอดสัญญาณแสดงจำนวนเฟสสว่างหมดทั้ง 3 หลอด สวิตช์จะตัดวงจร

2. กรณีแรงดันไฟฟ้าเฟสใดเฟสหนึ่งขาดไป

หลอดสัญญาณแสดงจำนวนเฟสของเฟสที่ขาดหายไปจะดับส่วนหลอดสัญญาณของเฟสที่มีอยู่จะสว่าง สวิตช์ตัดวงจร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

สรุป

เมื่อทำการจ่ายแรงดันมายังไมโครคอนโทรลเลอร์ CPU จะทำการประมวลผลคือ

1. เมื่อแรงดันที่จ่ายมาแต่ละสาย มีค่าระดับแรงดันเท่ากันหมดทั้ง 3 สาย CPU จะทำการเทียบค่าแรงดันทั้ง 3 เมื่อเท่ากันจริง ก็จะแสดงผลที่ Seven Segment
2. เมื่อแรงดันที่จ่ายมาแต่ละสาย โดยอาจมีบางสายมีระดับแรงดันที่ไม่เท่ากันเมื่อ CPU ประมวลผล จะส่งสัญญาณไป TRIP ให้ รีเลย์ turn off magnetic contractor และแสดงผลทาง LED ให้ทราบ
3. เมื่อแรงดันที่จ่ายมาแต่ละสาย โดยอาจมีบางสายไม่มีแรงดันเลย เมื่อ CPU ทำการประมวลผล จะส่งสัญญาณไป TRIP ให้ รีเลย์ turn off magnetic contractor และแสดงผลทาง LED ให้ทราบ

ปัญหาที่เกิดขึ้นและข้อเสนอแนะ

สามารถแบ่งปัญหาออกเป็น 2 ส่วนคือ Hardware and Software

ปัญหาส่วน Hardware

กล่าวถึงคุณสมบัติของเครื่องควบคุมระบบไฟฟ้า เดิมทีเครื่องต้นแบบก็สามารถทำงานได้เช่นกัน แต่เนื่องจากเครื่องต้นแบบ ใช้อุปกรณ์อิเล็กทรอนิกส์ ซึ่งเป็น Linear และมีผลกระทบเมื่ออุณหภูมิสูงขึ้น จึงใช้ไม่ได้ไม่ดีเท่าที่ควร แต่เมื่อใช้ไมโครคอนโทรลเลอร์มาเป็นตัวควบคุม จึงไม่จำเป็นต้องใช้อุปกรณ์อิเล็กทรอนิกส์มากนัก ซึ่งก็ได้ผลตามที่ต้องการ แต่ก็มีปัญหาบางส่วนเกิดขึ้น เนื่องจากสัญญาณนาฬิกา มีความถี่สูงพอควร จึงยากต่อการทำให้ได้สัญญาณตามต้องการ และที่ตัว CPU เอง ก็มีปัญหาเช่นกัน แต่ส่วนใหญ่เกี่ยวกับ Software

ปัญหาส่วน Software

Software ที่ใช้ใน MCS-48 มีจำกัดดังนั้น ในการเขียนโปรแกรมจึงเป็นการยากต่อผู้ที่เริ่มใช้ ไมโครคอนโทรลเลอร์บอร์ดนี้ แต่เนื่องจากได้มีซอฟต์แวร์เทอร์โบ 48 ช่วยในการเขียนโปรแกรมจึงทำให้ง่ายขั้นตอนการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] Intel Corporation, MCS 48 microcomputer user's manual
Literature Development, 3065 Bowers Avenue, Santa Clara,
(A 9505)
- [2] LINEAR DATA BOOK, Santa Clara California 95051: National
Semiconductor Corporation
- [3] นรินทร์ วัฒนกุล, ดิจิทัลเบื้องต้นและไมโครคอมพิวเตอร์พีซี, กรุงเทพฯ:
บริษัท ซีเอ็ค ยูเคชั่น, 2533
- [4] วิบูลย์ ชื่นแขก, ไมโครโปรเซสเซอร์, สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ,
กรุงเทพฯ

กิตติกรรมประกาศ

ขอขอบพระคุณ อ.ภากร หุตะสังกาศ เป็นอย่างสูง ที่ได้ให้การประสิทธิ์
ประสาทวิชา ให้คำแนะนำปรึกษาในเรื่องต่าง ๆ แก่คณะผู้จัดทำ และขอขอบพระคุณ
อ.ทรงชัย จิระทวีมาศ ที่ได้ให้คำแนะนำ และให้ความช่วยเหลือในการใช้ SOFTWARE
ขอขอบพระคุณ ผศ. พิพัฒน์ เล่าห์สงคราม ที่ได้ให้คำแนะนำ และคู่มือในการใช้ และ
ขอขอบคุณเพื่อน ๆ ที่ได้ช่วยเหลือ และช่วยเป็นกำลังใจ ในการทำปริญยานิพนธ์นี้จนสำเร็จ
ลุล่วงได้ด้วยดี




```

;*****
;This is main program of 3 phase monator.
;It will cheak condition throgouh many sub routines.
;*****

```

```
;
```

```
0000 CPU "8048.TBL"
```

```
0000 HOF "BIN8"
```

```
;
```

```
007F = DE0: EQU 01111111B
```

```
000F = DE1: EQU 00001111B
```

```
001F = DE2: EQU 00011111B
```

```
002F = DE3: EQU 00101111B
```

```
004F = DE4: EQU 01001111B
```

```
001F = POM: EQU 1FH
```

```
;
```

```
0000 ORG 000H
```

```
0000 27 MAIN: CLR A
```

```
0001 BD1F MOV R5,#POM ;initial delay time
```

```
0003 8A0F BBB: ORL P2,#00001111B ;initial status
```

```
0005 9A0F ANL P2,#00001111B ;select voltage phase 1
```

```
0007 9A0F ANL P2,#00001111B ;input form A/D
```

```
0009 B831 MOV R0,#31H ;store at $31
```

```
000B 08 INS A,BUS
```

```
000C A0 MOV @R0,A
```

```
000D ED03 DJNZ R5,BBB
```

```
;
```

```
000F B832 MOV R0,#32H
```

```
0011 BD1F MOV R5,#POM
```

```
0013 9A4F CCC: ANL P2,#01001111B ;select voltage phase 2
```

```
0015 8A4F ORL P2,#01001111B ;input form A/D
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0017 9A4F          ANL P2,#01001111B    ;store at $32
0019 08           INS A,BUS
001A A0           MOV @R0,A
001B ED13        DJNZ R5,CCC
;
001D B833        MOV R0,#33H
001F BD1F        MOV R5,#POM
0021 9A8F        DDD: ANL P2,#10001111B    ;select voltage phase 3
0023 8A8F        ORL P2,#10001111B    ;input form A/D
0025 9A8F        ANL P2,#10001111B    ;store at $33
0027 08           INS A,BUS
0028 A0           MOV @R0,A
0029 ED21        DJNZ R5,DDD
002B 00           NOP
002C B831        MOV R0,#31H    ;select phase 1 to change
002E F0           MOV A,@R0    ;form binary to BCD
002F AA           MOV R2,A    ;and load pattern to $21,$22,$23
0030 00           NOP    ;with display to 7-segment
0031 3461        CALL NODOUBLE
0033 5400        CALL LOAD
0035 7440        CALL DISPLAY1
0037 00           NOP
0038 B832        MOV R0,#32H    ;select phase 2 to change
003A F0           MOV A,@R0    ;form binary to BCD
003B AA           MOV R2,A    ;and load pattern to $21,$22,$23
003C 00           NOP    ;with display to 7-segment
003D 3461        CALL NODOUBLE
003F 5400        CALL LOAD
0041 7470        CALL DISPLAY2
0043 B833        MOV R0,#33H    ;select phase 3 to change

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่มีกรรมสิทธิ์ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และเผยแพร่ข้อมูลเชิง อื่นๆ เอกสารที่สงวนไว้เพื่อการศึกษาให้นำไปใช้

```

0045 F0          MOV A,@R0      ;form binary to BCD
0046 AA          MOV R2,A      ;and load pattern to $21,$22,$23
0047 00          NOP          ;with display to 7-segment
0048 3461        CALL NODOUBLE
004A 5400        CALL LOAD
004C 9400        CALL DISPLAY3
004E 3435        CALL TESTL
0050 B65A        JF0 LOSS1
0052 3448        CALL TESTU
0054 769E        JF1 LOSS2
0056 00          NOP
0057 0400        JMP MAIN
0059 00          NOP
005A BB1F        LOSS1: MOV R3,#01FH    ;display when voltage
005C B9FF        DLOSS: MOV R1,#OFFH   ;lower limit by display with
005E 99FF        DDLOSS: ANL P1,#OFFH  ;7-segment
0060 8938        ORL P1,#00111000B
0062 9938        ANL P1,#00111000B
0064 230F        MOV A,#DE1
0066 3A          OUTL P2,A
0067 237F        MOV A,#DE0
0069 3A          OUTL P2,A
006A 893F        ORL P1,#00111111B
006C 993F        ANL P1,#00111111B
006E 231F        MOV A,#DE2
0070 3A          OUTL P2,A
0071 237F        MOV A,#DE0
0073 3A          OUTL P2,A
0074 8971        ORL P1,#01110001B
0076 9971        ANL P1,#01110001B

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

แม้กระทั่งสิ่งอื่นใดที่ห้ามมิให้คัดลอกหรือเผยแพร่โดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0078 232F      MOV A,#DE3
007A 3A        OUTL P2,A
007B 237F      MOV A,#DE0
007D 3A        OUTL P2,A
007E 8971      ORL P1,#01110001B
0080 9971      ANL P1,#01110001B
0082 234F      MOV A,#DE4
0084 3A        OUTL P2,A
0085 237F      MOV A,#DE0
0087 3A        OUTL P2,A
0088 E95E      DJNZ R1,DDL0SS
008A EB5C      DJNZ R3,DLOSS
008C 99FF      ANL P1,#0FFH
008E 89FF      ORL P1,#0FFH
0090 BDFE      MOV R5,#0FFH
0092 BCFF      NOP0: MOV R4,#0FFH
0094 99FF      NOP1: ANL P1,#0FFH      ;off relay
0096 89FF      ORL P1,#0FFH
0098 EC94      DJNZ R4,NOP1
009A ED92      DJNZ R5,NOP0
009C 0400      JMP MAIN
;
009E BB1F      LOSS2: MOV R3,#1FH      ;display when voltage
00A0 B9FF      PLOSS: MOV R1,#0FFH      ;upper limit by display with
00A2 99FF      UPLOSS: ANL P1,#0FFH      ;7-segment
00A4 893E      ORL P1,#00111110B
00A6 993E      ANL P1,#00111110B
00A8 230F      MOV A,#DE1
00AA 3A        OUTL P2,A
00AB 237F      MOV A,#DE0

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00AD 3A	OUTL P2,A	
00AE 893F	ORL P1,#0011111B	
00B0 993F	ANL P1,#0011111B	
00B2 231F	MOV A,#DE2	
00B4 3A	OUTL P2,A	
00B5 237F	MOV A,#DE0	
00B7 3A	OUTL P2,A	
00B8 8971	ORL P1,#01110001B	
00BA 9971	ANL P1,#01110001B	
00BC 232F	MOV A,#DE3	
00BE 3A	OUTL P2,A	
00BF 237F	MOV A,#DE0	
00C1 3A	OUTL P2,A	
00C2 8971	ORL P1,#01110001B	
00C4 9971	ANL P1,#01110001B	
00C6 234F	MOV A,#DE4	
00C8 3A	OUTL P2,A	
00C9 237F	MOV A,#DE0	
00CB 3A	OUTL P2,A	
00CC E9A2	DJNZ R1,UPLOSS	
00CE EBA0	DJNZ R3,PLOSS	
00D0 89FF	ORL P1,#0FFH	
00D2 99FF	ANL P1,#0FFH	
00D4 BDFE	MOV R5,#0FFH	
00D6 BCFF	NOP2: MOV R4,#0FFH	
00D8 99FF	NOP3: ANL P1,#0FFH	;off relay
00DA 89FF	ORL P1,#0FFH	
00DC ECD8	DJNZ R4,NOP3	
00DE EDD6	DJNZ R5,NOP2	
00E0 0400	JMP MAIN	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่มีการแก้ไขทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

;*****
;This is subroutine of Test Program . The value are in
;condition of upper limit or lower limit.
;*****

```

```

0135          ORG 135H
;
0135 85      TESTL: CLR F0          ;innitial flag.
0136 BB03    MOV R3,#03H        ;select phase of voltage to test
0138 B831    MOV R0,#31H        ;and the other phase by test
013A F0      LOOP: MOV A,@R0     ;voltage is lower limit
013B 0382    ADD A,#082H        ;put data add to accumulator
013D E644    JNC OUTL1          ;old data at 200v =37,82=179
013F 18      INC R0
0140 EB3A    DJNZ R3,LOOP
0142 2445    JMP RT
0144 95      OUTL1:CPL F0
0145 83      RT:RET
0148          ORG 148H
;
0148 A5      TESTU: CLR F1        ;innitial flag
0149 BC03    MOV R4,#03H        ;select phase of phase to test
014B B831    MOV R0,#31H        ;and the other phase by test
014D F0      LOOP0: MOV A,@R0    ;voltage is upper limit
014E 0314    ADD A,#14H        ;put data add to accumulator
0150 F657    JC OUTL2
0152 18      INC R0
0153 EC4D    DJNZ R4,LOOP0
0155 2458    JMP RETUN
0157 B5      OUTL2: CPL F1
0158 83      RETUN: RET

```

```

;*****
;This subroutine will change binary to bcd
;and store in registor R4 and R3
;*****

```

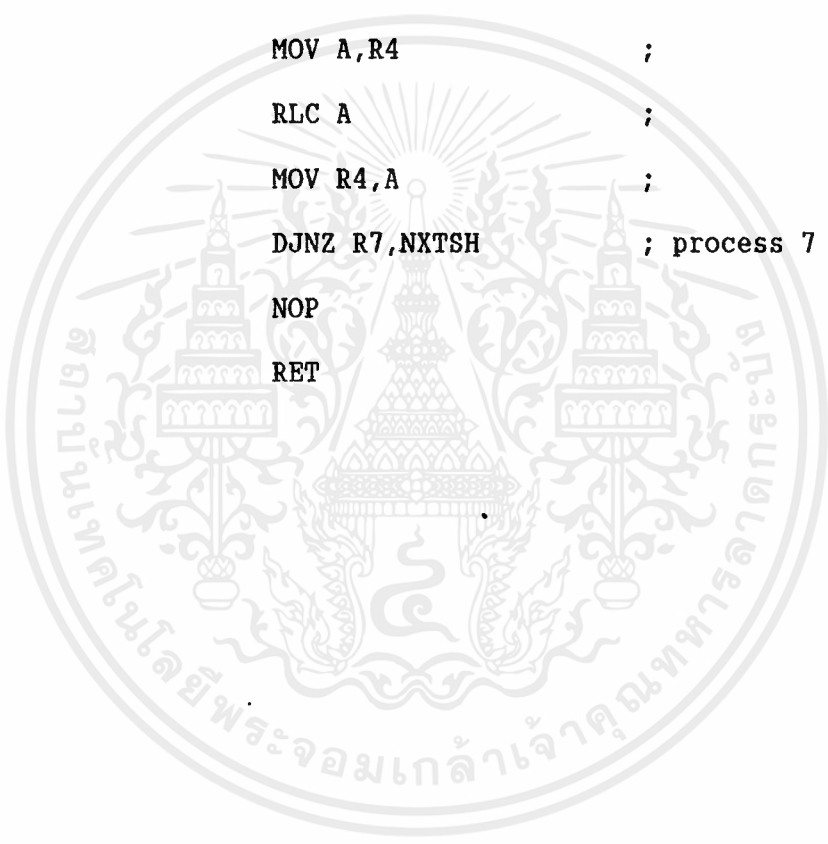
```

;
0161          ORG 161H
;
0161 97          NODOUBLE: CLR C
0162 246A         JMP CALC
0164 6A          DOUBLE: ADD A,R2          ; mutiply by 2
0165 AA          MOV R2,A                ;
0166 42          MOV A,T                ; sampling bias
0167 126A         JBO CALC                ; add with samplig
0169 1A          INC R2                  ;
016A 27          CALC: CLR A             ; initial registor
016B AC          MOV R4,A                ;
016C F7          RLC A                   ; check 9 th bit
016D AB          MOV R3,A                ;
016E BF08        MOV R7,#08H            ; set loop
0170 FB          NXTSH: MOV A,R3         ; check unit
0171 47          SWAP A                  ; if it greater 5
0172 03B0        ADD A,#0B0H            ;
0174 BE03        MOV R6,#03H           ;
0176 F67A        JC GREAT5             ;
0178 BE00        MOV R6,#00H           ;
017A 47          GREAT5: SWAP A         ; check ten
017B 03B0        ADD A,#0B0H            ; if it greater 5
017D FB          MOV A,R3                ;
017E E682        JNC LESS5              ;
0180 0330        ADD A,#30H             ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ควรเผยแพร่ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0182 6E	LESS5: ADD A,R6	; add with 00 or 03
0183 AB	MOV R3,A	; store vaule
0184 FA	MOV A,R2	;
0185 F7	RLC A	; shift process
0186 AA	MOV R2,A	;
0187 FB	MOV A,R3	;
0188 F7	RLC A	;
0189 AB	MOV R3,A	;
018A FC	MOV A,R4	;
018B F7	RLC A	;
018C AC	MOV R4,A	;
018D EF70	DJNZ R7,NXTSH	; process 7 times ?
018F 00	NOP	
0190 83	RET	



```

;*****
;This subroutine will load pattern witch correspond
;to valve in R4,R3.
;The pattern is store at $21 to $23.
;*****

```

```

;
0200          ORG 200H
;
0200 B821      LOAD: MOV R0,#21H          ;start address to store
0202 FC        MOV A,R4                ;find pattern correspond
0203 43F0      ORL A,#0F0H             ;look at table
0205 533F      ANL A,#3FH              ;
0207 E3        MOVP3 A,@A              ;find pattern for R4
0208 A0        MOV @R0,A               ;store first pattern
0209 FB        MOV A,R3                ;find pattern for R3
020A 47        SWAP A
020B 43F0      ORL A,#0F0H
020D 533F      ANL A,#3FH
020F E3        MOVP3 A,@A
0210 18        INC R0
0211 A0        MOV @R0,A               ;store second pattern
0212 FB        MOV A,R3
0213 43F0      ORL A,#0F0H
0215 533F      ANL A,#3FH
0217 E3        MOVP3 A,@A
0218 18        INC R0
0219 A0        MOV @R0,A               ;store third pattern
021A 00        NOP
021B 83        RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

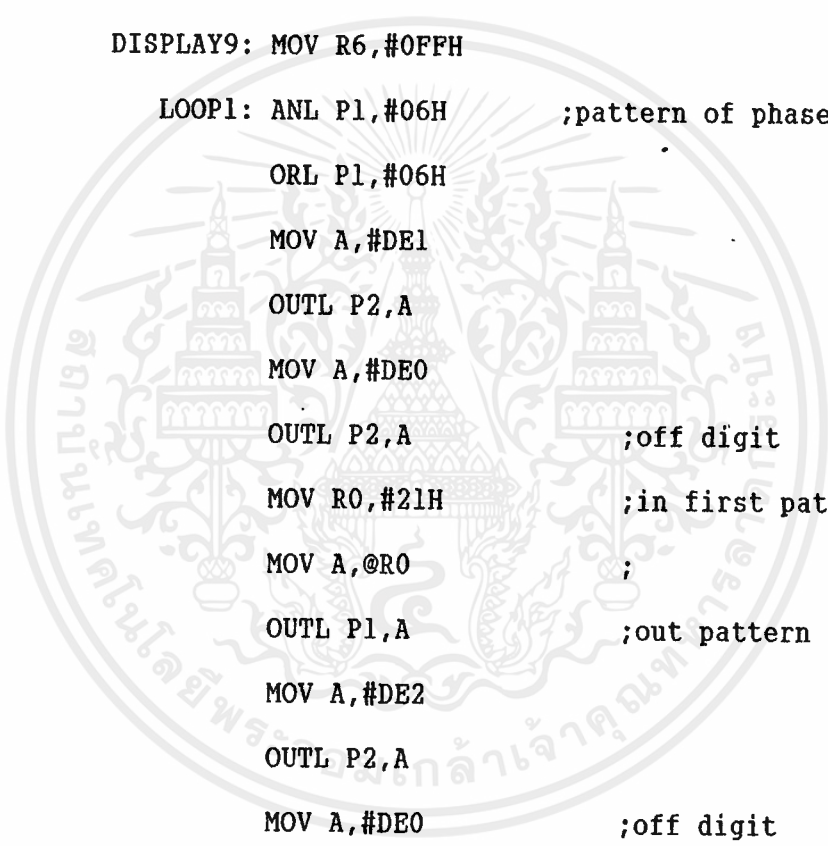
;*****
;This subroutine will send pattern of phase 1
;form $21 to $23 to 7-segment.
;*****

```

```

;
0340          ORG 340H
;
0340 BD5F      DISPLAY1: MOV R5,#5FH          ;inntial display delay time
0342 BEFF      DISPLAY9: MOV R6,#0FFH
0344 9906      LOOP1: ANL P1,#06H           ;pattern of phase 1
0346 8906      ORL P1,#06H
0348 230F      MOV A,#DE1
034A 3A        OUTL P2,A
034B 237F      MOV A,#DE0
034D 3A        OUTL P2,A                   ;off digit
034E B821      MOV R0,#21H                 ;in first pattern
0350 F0        MOV A,@R0
0351 39        OUTL P1,A                   ;out pattern
0352 231F      MOV A,#DE2
0354 3A        OUTL P2,A
0355 237F      MOV A,#DE0                   ;off digit
0357 3A        OUTL P2,A
0358 18        INC R0                       ;in second pattern
0359 F0        MOV A,@R0
035A 39        OUTL P1,A                   ;out pattern
035B 232F      MOV A,#DE3
035D 3A        OUTL P2,A
035E 237F      MOV A,#DE0                   ;off digit
0360 3A        OUTL P2,A
0361 18        INC R0                       ;in third pattern

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 0361 18 INC R0 ;in third pattern
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกแห่งที่มิให้นำไปใช้

```

0362 F0          MOV A,@R0
0363 39          OUTL P1,A          ;out pattern
0364 234F        MOV A,#DE4
0366 3A          OUTL P2,A
0367 237F        MOV A,#DE0          ;off digit
0369 3A          OUTL P2,A
036A EE44        DJNZ R6,LOOP1
036C ED42        DJNZ R5,DISPLAY9
036E 00          NOP
036F 83          RET

```



```

;*****
;This subroutine will send pattern of phase 2
;form $21 to $23 to 7-segment.
;*****

```

```

;
0370          ORG 370H
;
0370 BD5F          DISPLAY2: MOV R5,#5FH
0372 BE7F          DISPLAY8: MOV R6,#07FH          ;initial display delay time
0374 995B          LOOP2: ANL P1,#05BH          ;pattern of phase 2
0376 895B          ORL P1,#05BH
0378 230F          MOV A,#DE1
037A 3A          OUTL P2,A
037B 237F          MOV A,#DE0
037D 3A          OUTL P2,A          ;off digit
037E B821          MOV RO,#21H          ;in first pattern
0380 F0          MOV A,@RO
0381 39          OUTL P1,A          ;out pattern
0382 231F          MOV A,#DE2
0384 3A          OUTL P2,A
0385 237F          MOV A,#DE0          ;off digit
0387 3A          OUTL P2,A
0388 18          INC RO          ;in second pattern
0389 F0          MOV A,@RO
038A 39          OUTL P1,A          ;out pattern
038B 232F          MOV A,#DE3
038D 3A          OUTL P2,A
038E 237F          MOV A,#DE0          ;off digit
0390 3A          OUTL P2,A
0391 18          INC RO          ;in second pattern

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 0391 18 INC RO ;in second pattern
 ไม่หวังผลตอบแทนใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีรูปแบบไปใช้

0392 F0	MOV A,@R0	
0393 39	OUTL P1,A	;out pattern
0394 234F	MOV A,#DE4	
0396 3A	OUTL P2,A	
0397 237F	MOV A,#DE0	;off digit
0399 3A	OUTL P2,A	
039A EE74	DJNZ R6,LOOP2	
039C ED72	DJNZ R5,DISPLAY8	
039E 00	NOP	
039F 83	RET	



```

;*****
;This subroutine will send pattern of phase 3
;form $21 to $23 to 7-segment.
;*****

```

```
;
```

```

0400          ORG 400H
0400 BD5F     DISPLAY3: MOV R5,#5FH
0402 BEFF     DISPLAY7: MOV R6,#0FFH      ;initial display delay time
0404 994F     LOOP3: ANL P1,#04FH        ;pattern of phase 3
0406 894F     ORL P1,#04FH
0408 230F     MOV A,#DE1
040A 3A       OUTL P2,A
040B 237F     MOV A,#DE0
040D 3A       OUTL P2,A                  ;off digit
040E B821     MOV R0,#21H                ;in first pattern
0410 F0       MOV A,@R0
0411 39       OUTL P1,A                  ;out pattern
0412 231F     MOV A,#DE2
0414 3A       OUTL P2,A
0415 237F     MOV A,#DE0                  ;off digit
0417 3A       OUTL P2,A
0418 18       INC R0                      ;in second pattern
0419 F0       MOV A,@R0
041A 39       OUTL P1,A                  ;out pattern
041B 232F     MOV A,#DE3
041D 3A       OUTL P2,A
041E 237F     MOV A,#DE0                  ;off digit
0420 3A       OUTL P2,A
0421 18       INC R0                      ;in third pattern

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 0422 F0 MOV A,@R0
 ไม่วการณใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0423 39          OUTL P1,A          ;out pattern
0424 234F        MOV A,#DE4
0426 3A          OUTL P2,A
0427 237F        MOV A,#DE0          ;off digit
0429 3A          OUTL P2,A
042A EE04        DJNZ R6,LOOP3
042C ED02        DJNZ R5,DISPLAY7
042E 00          NOP
042F 83          RET

;
; DISPLAY PATTERN
;

0330             ORG 330H
0330 3F          DFB 3FH          ;0
0331 06          DFB 06H          ;1
0332 5B          DFB 5BH          ;2
0333 4F          DFB 4FH          ;3
0334 66          DFB 66H          ;4
0335 6D          DFB 6DH          ;5
0336 7D          DFB 7DH          ;6
0337 07          DFB 07H          ;7
0338 7F          DFB 7FH          ;8
0339 6F          DFB 6FH          ;9
0000             END

```

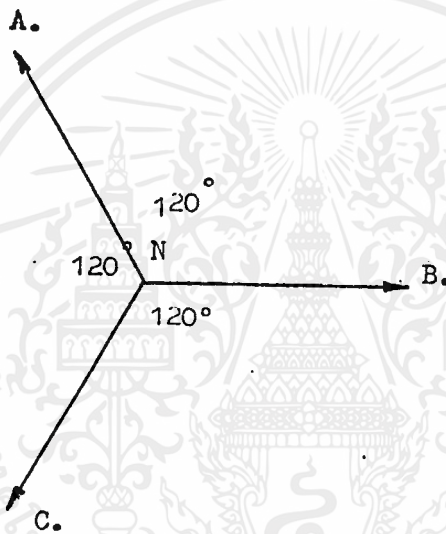
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0003	BBB	016A	CALC	0013	CCC
0021	DDD	005E	DDLOSS	007F	DE0
000F	DE1	001F	DE2	002F	DE3
004F	DE4	0340	DISPLAY1	0370	DISPLAY2
0400	DISPLAY3	0402	DISPLAY7	0372	DISPLAY8
0342	DISPLAY9	005C	DLOSS	0164	DOUBLE
017A	GREAT5	0182	LESS5	0200	LOAD
013A	LOOP	014D	LOOP0	0344	LOOP1
0374	LOOP2	0404	LOOP3	005A	LOSS1
009E	LOSS2	0000	MAIN	0161	NODOUBLE
0092	NOPO	0094	NOPI	00D6	NOP2
00D8	NOP3	0170	NXTSH	0144	OUTL1
0157	OUTL2	00A0	PLOSS	001F	POM
0158	RETUN	0145	RT	0135	TESTL
0148	TESTU	00A2	UPLOSS		

บทที่ 2.

ความรู้พื้นฐาน

ระบบไฟฟ้าที่มีใช้อยู่ในโรงงานอุตสาหกรรมของประเทศไทยในปัจจุบันนี้เป็นระบบ 3 เฟส 4 สาย คือประกอบด้วยสายเฟส 3 เส้น และสายนิวทรัลอีก 1 เส้น แรงดันไฟฟ้าระหว่างสายเฟสกับสายเฟส 380 โวลต์ และแรงดันไฟฟ้าระหว่างสายเฟสกับสายนิวทรัล 220 โวลต์ ซึ่งแต่ละเฟสจะห่างเฟสกันอยู่ 120° ทางไฟฟ้า ดังเวกเตอร์รูปที่ 2.1



รูปที่ 2.1. เวกเตอร์ของแรงดันไฟฟ้า 3 เฟส

$$V_A = V_m \sin(\omega t + 120^\circ) = V_m \angle 120^\circ$$

$$V_B = V_m \sin \omega t = V_m \angle 0^\circ$$

$$V_C = V_m \sin(\omega t - 120^\circ) = V_m \angle -120^\circ$$

การจักระบบเฟสจะมีผลต่อทิศทางการหมุนของมอเตอร์ ซึ่งจากรูปที่ 2.1 จะเห็นว่าแรงดันไฟฟ้าระบบ 3 เฟสเป็นการจักระบบแบบวงกลม ดังนั้นจำนวนวิธีของการจักระบบเฟสที่จะเกิดขึ้นได้ เป็นไปตามสมการ 2.1

$$\text{จำนวนวิธี} = (n-1) ! \quad 2.1$$

เมื่อ n คือจำนวนเฟส ($n=3$)

$$\therefore \text{จำนวนวิธี} = (3-1) ! = 2$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั่นคือการจับลำดับเฟสมีอยู่เพียง 2 กรณีเท่านั้น

- กรณีที่ 1 ลำดับเฟสเป็น ABC หรือ BCA หรือ CAB
- กรณีที่ 2 ลำดับเฟสเป็น ACB หรือ CBA หรือ BAC

2.1. พิจารณาในกรณีที่แรงดันไฟฟ้ามาครบทั้ง 3 เฟส

ในหัวข้อนี้จะนำแรงดันไฟฟ้าระบบ 3 เฟส ในรูปที่ 2.1 มาพิจารณาซึ่งจะพิจารณาในกรณีที่ไม่มีแรงดันไฟฟ้าเฟสหนึ่งเฟสใดขาดหายไป นั่นคือทั้งเฟส A เฟส B และเฟส C มีแรงดันไฟฟ้าทุกเฟส โยที่จะยึกเฟสหนึ่งเฟสใดเป็นเฟสอ้างอิง 1 เฟส ที่เหลืออีก 2 เฟส จำทำการเลื่อนเฟสทั้งสองให้เลื่อนไป 120° ในทิศทางตรงข้ามกัน กล่าวคือเมื่อเลื่อนเฟสหนึ่งให้ลาหลังไป 120° ก็ต้องเลื่อนอีกเฟสหนึ่งให้หน้าหน้า 120° แล้วนำมารวมกัน

2.1.1 ถ้ายึดเอาเฟส B เป็นเฟสอ้างอิง แล้วเลื่อนเฟส A ให้ลาหลังไป 120° เลื่อนเฟส C ให้หน้าหน้าไป 120° จะทำให้เวกเตอร์ของแรงดันไฟฟ้าทั้ง 3 เฟส มีทิศทางเดียวกันดังรูปเวกเตอร์รูปที่ 2.2

N
A, B, C

รูปที่ 2.2. เลื่อนเฟสให้ตรงกัน

$$V_A = V_m \sin (\omega t + 120^\circ - 120^\circ)$$

$$= V_m \sin \omega t = V_m \angle 0^\circ$$

$$V_B = V_m \sin \omega t = V_m \angle 0^\circ$$

$$V_C = V_m \sin (\omega t - 120^\circ + 120^\circ)$$

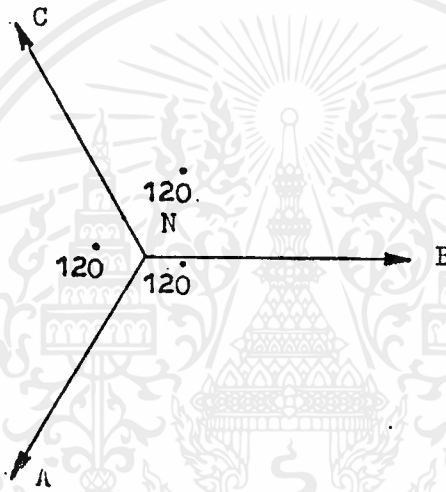
$$= V_m \sin \omega t = V_m \angle 0^\circ$$

จากนั้นรวมค่าแรงดันไฟฟ้าสูงสุดของเฟส A, B และ C

$$\therefore |V_A + V_B + V_C| = 3V_m$$

2.1.2 ถ้ายึดเอาเฟส B เป็นเฟสอ้างอิง

แล้วเลื่อนเฟส A ให้นำหน้าไป 120° เลื่อนเฟส C ให้ล่าหลังไป 120° จะได้เวกเตอร์ดังรูปที่ 2.3



รูปที่ 2.3 เลื่อนเฟสทางกัน 120°

$$\begin{aligned} V_A &= V_m \sin(\omega t + 120^\circ + 120^\circ) \\ &= V_m \sin(\omega t + 240^\circ) = V_m \angle 240^\circ \end{aligned}$$

$$V_B = V_m \sin \omega t = V_m \angle 0^\circ$$

$$\begin{aligned} V_C &= V_m \sin(\omega t - 120^\circ - 120^\circ) \\ &= V_m \sin(\omega t - 240^\circ) \\ &= V_m \sin(\omega t + 120^\circ) = V_m \angle 120^\circ \end{aligned}$$

$$\begin{aligned} |V_A + V_B + V_C| &= |V_m \angle 240^\circ + V_m \angle 0^\circ + V_m \angle 120^\circ| \\ &= |V_m \cos 240^\circ + j \sin 240^\circ + \cos 0^\circ + j \sin 0^\circ + \cos 120^\circ + j \sin 120^\circ| \\ &= |V_m (-0.5 - j0.866 + j0 - 0.5 + j0.866)| \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากหลักการข้างต้น ก็สามารถเปลี่ยนเฟสอ้างอิงจากเฟส B ไปเป็นเฟส A และเฟส C ซึ่งสรุปได้ตามตารางที่ 2.1

ตารางที่ 2.1 เมื่อมีแรงดันไฟฟ้าครบทั้ง 3 เฟส

	เลื่อนเฟสล่าช้า 120°	เลื่อนเฟสหน้า 120°	$ V_A+V_B+V_C $
A เป็นเฟสอ้างอิง	B	C	0
	C	B	$3V_m$
B เป็นเฟสอ้างอิง	C	A	0
	A	C	$3V_m$
C เป็นเฟสอ้างอิง	A	B	0
	B	A	$3V_m$

จากตารางที่ 2.1 จะเห็นได้ว่าไม่ว่าจะเอาเฟสไหนเป็นเฟสอ้างอิงและเลื่อนเฟสไหนหน้าหรือล่าช้าก็ตาม ผลที่ได้จากการรวมแรงดันไฟฟ้าทั้ง 3 เฟส มีเพียง 2 กรณีเท่านั้น คือ 0 และ $3V_m$ ผลรวมที่มีค่าเท่ากันมีลำดับเฟสเป็นดังนี้

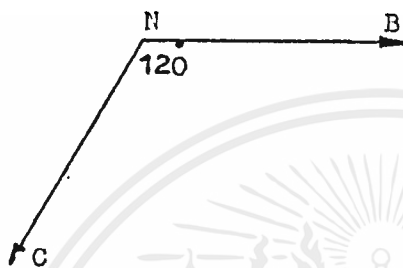
กรณีที่ 1 ลำดับเฟสเป็น ABC หรือ BCA หรือ CAB

กรณีที่ 2 ลำดับเฟสเป็น ACB หรือ BAC หรือ CBA

2.2 ทิวารณาในกรณีที่แรงดันไฟฟ้าเฟสหนึ่งเฟสโศกชกไป

ทิวารณาในกรณีแรงดันไฟฟ้าไม่ครบ 3 เฟส การทิวารณายังคงใช้หลักการเดิม คือยึกเฟสโศกเฟสหนึ่งเป็นเฟสอ้างอิง แล้วทำการเลื่อนเฟสที่เหลือ นำเอาแรงดันไฟฟ้าแต่ละเฟสมารวมกัน ซึ่งจะแยกทิวารณาได้ดังนี้ โดยยึกเอาเฟส B เป็นเฟสอ้างอิง

2.2.1 ถ้าเฟส A ชากไป จะได้เวกเตอร์ดังรูปที่ 2.4



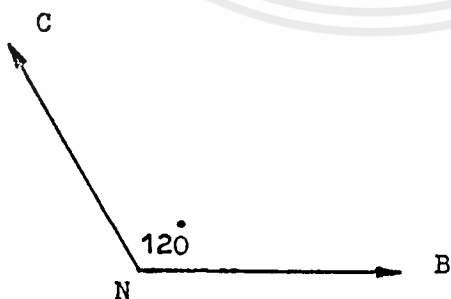
รูปที่ 2.4 เฟส A ชากไป

$$V_A = 0$$

$$V_B = V_m \sin \omega t = V_m \angle 0^\circ$$

$$V_C = V_m \sin (\omega t - 120^\circ) = V_m \angle -120^\circ$$

- ถ้าเฟส C ถูกเลื่อนให้ล่าหลัง 120° จะได้ภาพแนวเวกเตอร์ดังรูปที่ 2.5



รูปที่ 2.5 เลื่อนเฟส C ล้าหลัง 120°

$$V_A = 0$$

$$V_B = V_m \sin \omega t = V_m \angle 0^\circ$$

$$V_C = V_m \sin (\omega t - 120^\circ - 120^\circ)$$

$$= V_m \sin (\omega t - 240^\circ)$$

$$= V_m \sin (\omega t + 120^\circ)$$

$$= V_m \angle 120^\circ$$

$$|V_A + V_B + V_C| = |V_m (\cos 0^\circ + j \sin 0^\circ + \cos 120^\circ + j \sin 120^\circ)|$$

$$= |V_m (0.5 + j0.866)| = V_m$$

-ถ้าเฟส C ถูกเลื่อนให้หน้าหน้า 120° จะได้เวกเตอร์ดังรูปที่ 2.6



รูปที่ 2.6 เลื่อนเฟส C หน้าหน้า 120°

$$V_A = 0$$

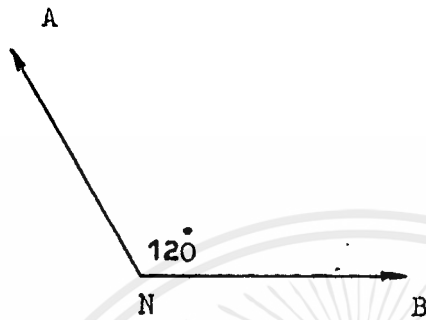
$$V_B = V_m \sin \omega t = V_m \angle 0^\circ$$

$$V_C = V_m \sin (\omega t - 120^\circ + 120^\circ)$$

$$= V_m \sin \omega t = V_m \angle 0^\circ$$

$$|V_A + V_B + V_C| = V_m + V_m = 2 V_m$$

2.2.2 ถ้าแรงดันไฟฟ้าเฟส C จากหายไป จะได้เวกเตอร์ดังรูปที่ 2.7



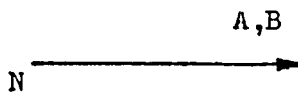
รูปที่ 2.7 เฟส C จากไป

$$V_A = V_m \sin(\omega t + 120^\circ) = V_m \angle 120^\circ$$

$$V_B = V_m \sin \omega t = V_m \angle 0^\circ$$

$$V_C = 0$$

- ถ้าเฟส A ถูกเลื่อนให้ล้าถึง 120° จะได้เวกเตอร์ดังรูปที่ 2.8

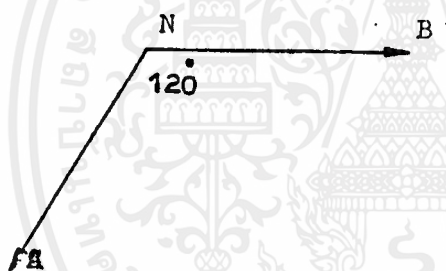


รูปที่ 2.8 เฟส A เลื่อนล้าถึง 120°

$$\begin{aligned}
 V_A &= V_m \sin(\omega t + 120^\circ - 120^\circ) \\
 &= V_m \sin \omega t = V_m \angle 0^\circ \\
 V_B &= V_m \sin \omega t = V_m \angle 0^\circ \\
 V_C &= 0
 \end{aligned}$$

$$|V_A + V_B + V_C| = V_m + V_m = 2 V_m$$

- ถ้าเฟส A ถูกเลื่อนให้หน้าหน้า 120° จะได้เวกเตอร์ดังรูปที่ 2.9

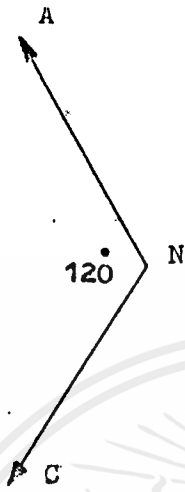


รูปที่ 2.9 เฟส A เลื่อนหน้าหน้า 120°

$$\begin{aligned}
 V_A &= V_m \sin(\omega t + 120^\circ + 120^\circ) \\
 &= V_m \sin(\omega t + 240^\circ) = V_m \angle 240^\circ \\
 V_B &= V_m \sin \omega t = V_m \angle 0^\circ \\
 V_C &= 0
 \end{aligned}$$

$$\begin{aligned}
 |V_A + V_B + V_C| &= |V_m(\cos 240^\circ + j\sin 240^\circ + \cos 0^\circ + j\sin 0^\circ)| \\
 &= |V_m(0.5 - j0.866)| = V_m
 \end{aligned}$$

2.2.3 ถ้าแรงดันไฟฟ้าเฟส B ซากหายไป จะได้เวกเตอร์ดังรูปที่ 2.10



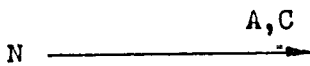
รูปที่ 2.10 เฟส B ซากไป

$$V_A = V_m \sin(\omega t + 120^\circ) = V_m \angle 120^\circ$$

$$V_B = 0$$

$$V_C = V_m \sin(\omega t - 120^\circ) = V_m \angle -120^\circ$$

- ถ้าเฟส A ถูกเลื่อนสลาหลัง 120° เฟส C ถูกเลื่อนนำหน้า 120° จะได้
เวกเตอร์ดังรูปที่ 2.11



รูปที่ 2.11 เลื่อนเฟสให้ตรงกัน

$$V_A = V_m \sin(\omega t + 120^\circ - 120^\circ)$$

$$= V_m \sin \omega t = V_m \angle 0^\circ$$

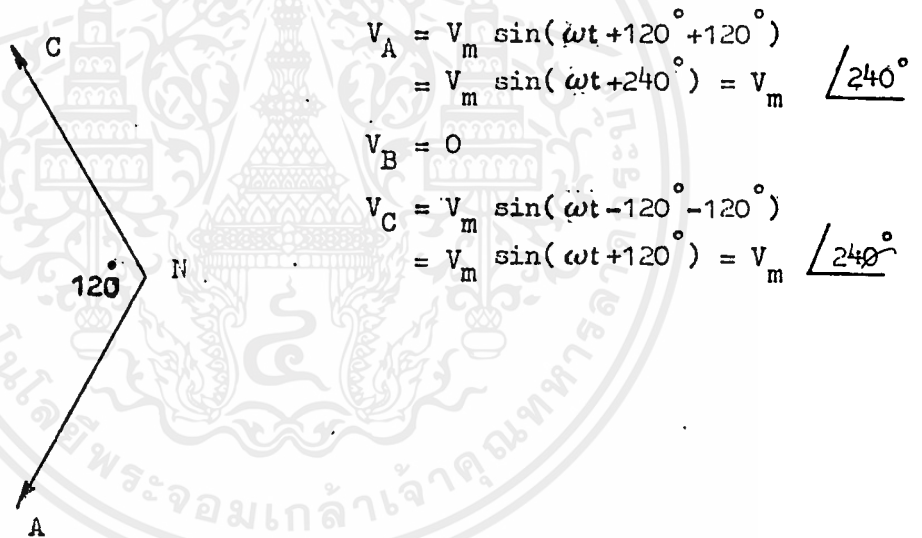
$$V_B = 0$$

$$V_C = V_m \sin(\omega t - 120^\circ + 120^\circ)$$

$$= V_m \sin \omega t = V_m \angle 0^\circ$$

$$|V_A + V_B + V_C| = V_m + V_m = 2 V_m$$

-ถ้าเฟส B ถูกเลื่อนหน้า 120° เฟส C ถูกเลื่อนล่าหลัง 120° จะได้
เวกเตอร์ทั้งรูปที่ 2.12



รูปที่ 2.12 เลื่อนเฟสต่างกัน 120°

$$|V_A + V_B + V_C| = |V_m (\cos 240^\circ + j \sin 240^\circ + \cos 120^\circ + j \sin 120^\circ)|$$

$$= |V_m (-1 + j0)| = V_m$$

ที่นำมาเป็นการพิจารณาที่ใช้เฟส B เป็นเฟสอ้างอิง ถ้าจะเอาเฟสอื่นเป็นเฟสอ้างอิงก็สามารถทำได้โดยใช้หลักการเดียวกัน ซึ่งค่าผลรวมของแรงดันไฟฟ้าสูงสุดทั้ง 3 เฟส สามารถสรุปได้ตาม ตารางที่ 2.2

ตารางที่ 2.2 เมื่อแรงดันไฟฟ้าบางเฟสขาดไป

	เฟสที่ขาดไป	เลื่อนเฟสลาหลัง 120°	เลื่อนเฟสหน้า 120°	$V_A + V_B + V_C$
เฟส A เป็นเฟสอ้างอิง	A	B	C	V_m
		C	B	$2V_m$
	B	-	C	V_m
		C	-	$2V_m$
	C	B	-	V_m
		-	B	$2V_m$
เฟส B เป็นเฟสอ้างอิง	A	C	-	V_m
		-	C	$2V_m$
	B	C	A	V_m
		A	C	$2V_m$
	C	-	A	V_m
		A	-	$2V_m$
เฟส C เป็นเฟสอ้างอิง	A	-	B	V_m
		B	-	$2V_m$
	B	A	-	V_m
		-	A	$2V_m$
	C	A	B	V_m
		B	A	$2V_m$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 2.2 จะสังเกตเห็นว่าไม่ว่าจะยึดเอาแรงดันไฟฟ้าเฟสใด เป็นเฟสอ้างอิง แรงดันไฟฟ้าเฟสใดซาก็หายไป และเลื่อนเฟสไหนอย่างไรก็ตามผลรวมของแรงดันไฟฟ้าสูงสุด จะมีเพียง 2 ค่าเท่านั้น คือ V_m และ $2V_m$ ไม่มีค่าเป็นศูนย์เลย

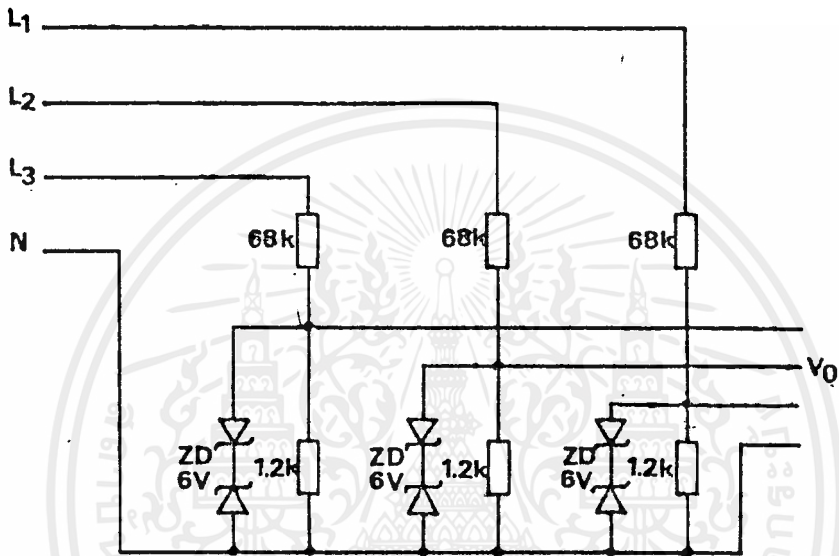
ในกรณีที่แรงดันไฟฟ้าซาก็หายไป 2 เฟส ยังคงเหลือแรงดันไฟฟ้าอยู่เฟสเดียว เพราะฉะนั้นในกรณีนี้ไม่ว่าจะยึดเอาแรงดันไฟฟ้าเฟสใดเป็นเฟสอ้างอิง แรงดันไฟฟ้าเฟสใดซาก็หายไป และเลื่อนเฟสไหนอย่างไรก็ตามผลรวมของแรงดันไฟฟ้าสูงสุด จะมีค่าเท่ากับ V_m เสมอ.



บทที่ 4.

การทำงานของวงจรและการทดลอง

4.1 วงจรลดแรงดันไฟฟ้า (Attenuator)



รูปที่ 4.1 วงจรลดแรงดันไฟฟ้า

วงจรลดแรงดันไฟฟ้า จะทำหน้าที่ลดแรงดันไฟฟ้า 220 โวลต์ให้ต่ำลง เพื่อ
 พอเหมาะกับวงจรของเครื่องกรวาล่ากับเฟส ในที่นี้จะใช้ตัวต้านทานลดแรงดันไฟฟ้าลง

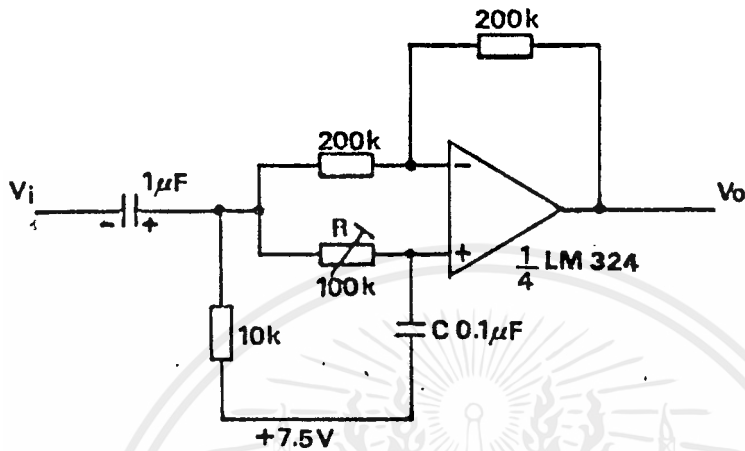
$$V_o = \frac{1.2 \times 220}{68 + 1.2} = 3.8 \text{ V.}$$

$$V_m = \sqrt{2} \times 3.8 = 5.4 \text{ V.}$$

ที่สำคัญเขาที่ทุหจะมีซึนเนอร์โคโอก 6 โวลท์ ท่อกล่อมตัวต้านทาน 1.2 กิโล-
 โห้มอยู่ทั้งนี้เพื่อป้องกันอันทรายที่จะ เกิดซึนกับวงจรของเก็องกรวาล่ากับเฟสในกรณีทีตัว
 ท้าทาน 68 กิโลโห้ม เกิดลัทวงจรซึนมา เพราะ ซึนเนอร์โคโอกจะเป็นทัวจวกักแรงแ
 กันไฟฟ้าเอาไว้ที่แรงแกน 6 โวลท์

4.2 วงจรเลื่อนเฟสลาถหลัง 120° (Phase Shifter 120° Lagging)

4.2.1 การทำงานของวงจร



รูปที่ 4.2 วงจรเลื่อนเฟสลาถหลัง 120°

วงจรเลื่อนเฟสลาถหลัง 120° จากรูปที่ 4.2 ซึ่งจะใช้ออปแอมป์ LM 324 ซึ่งเป็นวงจรที่มีอัตราขยายเท่ากับ 1. ส่วนมุมที่เฟสเลื่อนไปสามารถคำนวณได้จากสมการ

$$\theta = -2 \tan^{-1} 2\pi fCR$$

จากสมการข้างบนจะเห็นว่ามุมที่จะเลื่อนไปนั้นขึ้นอยู่กับความถี่ตัวต้านทาน และตัวเก็บประจุ ในที่นี้ความถี่คือ 50 เฮิรท์ และกำหนดให้ค่าตัวเก็บประจุเท่ากับ 0.1 ไมโครฟาราด ดังนั้นมุมที่เลื่อนไปจึงขึ้นอยู่กับตัวต้านทาน โดยที่

$$R = \frac{1}{2\pi fC} \tan\left(-\frac{\theta}{2}\right)$$

$$\text{ต้องการ } \theta = -120^\circ$$

$$\therefore R = \frac{10^6}{2\pi \times 50 \times 0.1} \tan 60^\circ = 55 \text{ K.}$$

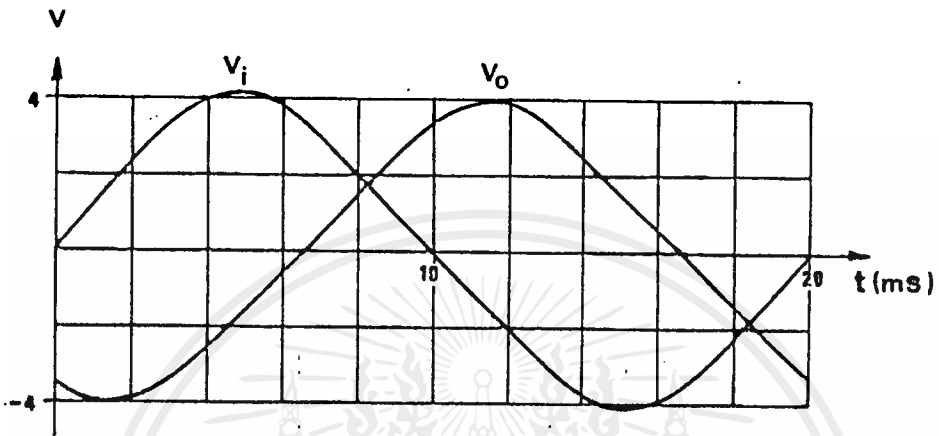
สำหรับค่าความต้านทาน ป้อนกลับ และความต้านทานที่ต่ออยู่กับขา ลบ(-) ของออปแอมป์ ต้องมีค่าเท่ากัน ทั้งนี้เพื่อต้องการให้มีค่าอัตราขยายเท่ากับ 1. ซึ่งค่าความต้านทานนี้สามารถเลือกใช้ได้ทั้ง 10 กิโลโอห์ม - 220 กิโลโอห์ม

4.2.2 การทดลอง

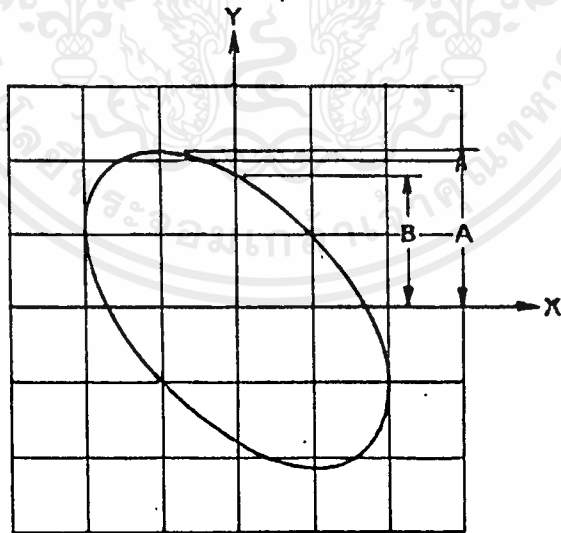
เมื่อสัญญาณ ไซน์ (sine wave) อินพุต ของวงจรเลื่อนเฟสลาติง 120° แล้วทำการวัดสัญญาณ เอาท์พุท ปรับค่าความต้านทานทำให้สัญญาณ อินพุท และ เอาท์พุท ท่างเฟสกัน ปรับค่าความต้านทานจนกระทั่งได้สัญญาณเอาท์พุทลาติงสัญญาณ อินพุทอยู่ 120° ซึ่งค่าความต้านทานนี้มีค่าประมาณ 60 กิโลโอห์ม ดังนั้นในการใช้งาน ตัวต้านทานจะเป็นแบบปรับค่าได้ขนาด 100 กิโลโอห์ม

สัญญาณที่ได้จากการทดลองก็แสดงในรูปที่ 4.3 ก. และ 4.3 ข. จากรูปที่ 4.3 ก. จะเห็นได้ว่าทั้งสัญญาณ อินพุท และ เอาท์พุท จะมีลักษณะเหมือนกันทุกประการ แต่ สัญญาณเอาท์พุทและสัญญาณอินพุท ซึ่งลาติงอยู่เท่าไรนั้น มุมที่ต่างเฟสกันหาได้จากรูปที่ 4.3 ข. ซึ่งจะได้อ่านค่ามุมต่างเฟสประมาณ 120°

ถ้าหากทดลองปรับขนาดของสัญญาณ อินพุทให้มีค่าสูงขึ้น สัญญาณ เอาท์พุทจะถูกขลิบที่ 6.5 โวลต์ หมายความว่าเมื่อสัญญาณ อินพุท มีค่าเกิน 6.5 โวลต์ ไปแล้วสัญญาณ เอาท์พุทจะมีรูปร่างผิดไปจากสัญญาณ อินพุท



รูปที่ 4.3 ก. สัญญาณอินพุตและเอาต์พุต

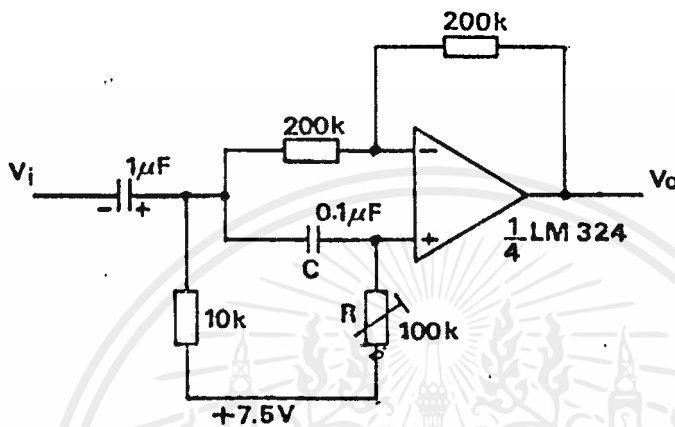


รูปที่ 4.3 ข. มุมท่างเฟสระหว่างสัญญาณอินพุตและเอาต์พุต

$$\theta = \sin^{-1} \frac{B}{A} = \sin^{-1} \frac{9}{10.5} = 121^\circ$$

4.3 วงจรเลื่อนเฟสหน้า 120° (Phase Shifter 120° Leading)

4.3.1 การทำงานของวงจร



รูปที่ 4.4 วงจรเลื่อนเฟสหน้า 120°

การทำงานของวงจรอาศัยหลักการเดียวกับวงจรเลื่อนเฟสลาถหลัง 120° ดังเกทวงจรจากรูปที่ 4.4 จะต่างกับวงจรจากรูปที่ 4.2 ตรงที่การสลับที่ระหว่าง ตัวต้านทาน กับ ตัวเก็บประจุ สำหรับ อีกรายขยายของวงจรยังเหมือนเดิม คือ เท่ากับหนึ่งสัญญาณเอาต์พุต นำหน้าสัญญาณอินพุต ซึ่งมุมที่ต่างเฟสกันคำนวณได้จากสมการ

$$\theta = -180 - 2 \tan^{-1} 2\pi fCR$$

เมื่อกำหนดให้ความถี่คงที่ 50 เฮิรตซ์ และ ตัวเก็บประจุ คงที่ 0.1 ไมโครฟาราด

$$\therefore R = -\frac{1}{2\pi fC} \tan \frac{\theta+180}{2}$$

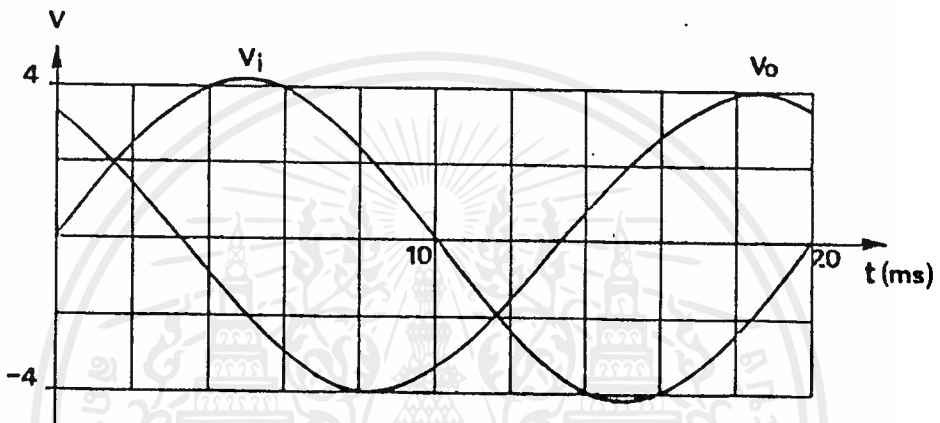
เมื่อต้องการ $\theta = 120^\circ$

$$R = -\frac{10^6}{2\pi \times 50 \times 0.1} \tan \frac{120+180}{2}$$

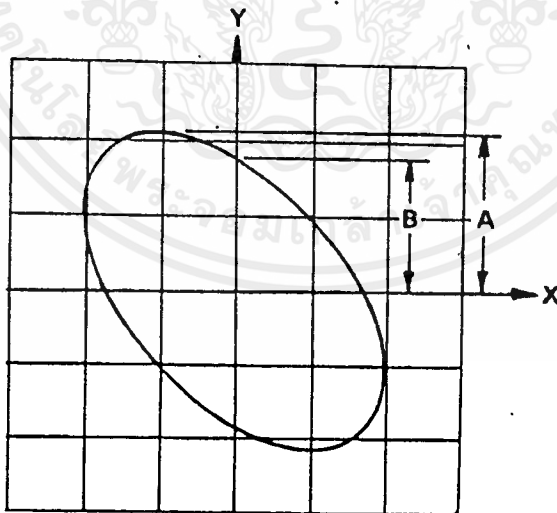
$$= 18.4 \text{ K.}$$

4.3.2 การทดลอง

ในการทดลองสัญญาณ เอาท์พุท จะนำหน้าสัญญาณ อินพุทโดยการปรับค่าความต้านทาน เมื่อสัญญาณ เอาท์พุทนำหน้าสัญญาณ อินพุท 120° ค่าความต้านทานเท่ากับ 25 กิโลโอห์ม สัญญาณที่ได้จากการทดลอง ดังแสดงในรูปที่ 4.5 ก. และ 4.5 ข.



รูปที่ 4.5 ก. สัญญาณอินพุท และสัญญาณเอาท์พุท



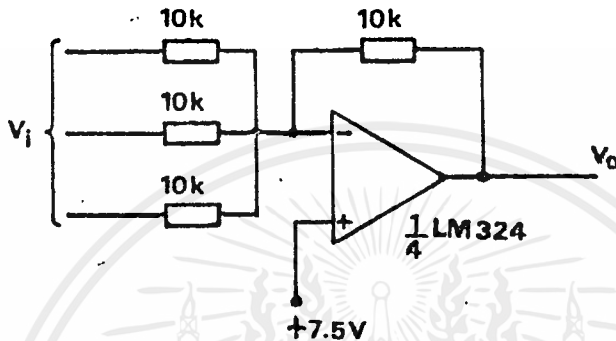
รูปที่ 4.5 ข. มุมทแยงระหว่างสัญญาณอินพุทและเอาท์พุท

$$\theta = \sin^{-1} \frac{B}{A} = \sin^{-1} \frac{9}{10.5} = 121^\circ$$

ถ้าเพิ่มสัญญาณ อินพุท จะทำให้สัญญาณเอาท์พุทถูกลบที่ 6.5 โวลต์

4.4 วงจรรวมสัญญาณไฟฟ้า (Summing Amplifier)

4.4.1 การทำงานของวงจร



รูปที่ 4.6 วงจรรวมสัญญาณไฟฟ้า

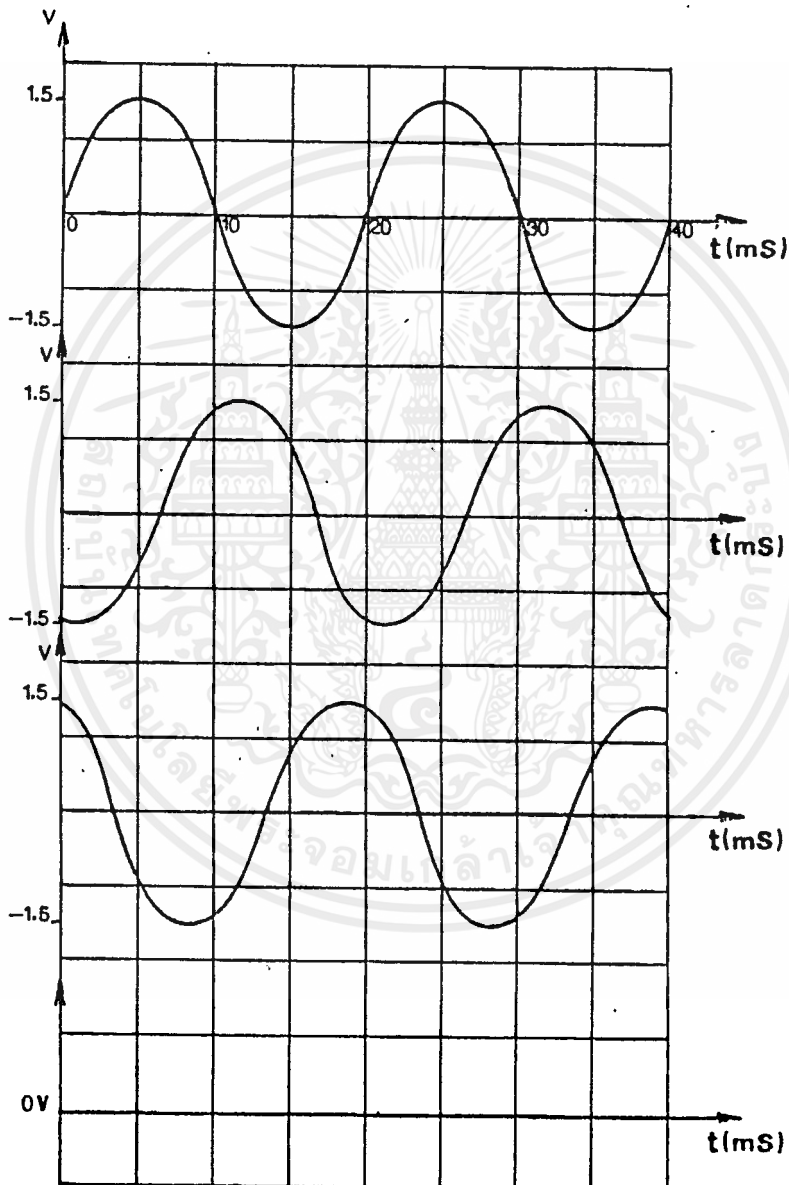
จากรูปที่ 4.6 เป็นวงจรรวมสัญญาณไฟฟ้า ที่ใช้ ออปแอมป์ LM 324 วงจรนี้จะมี อัตราขยายเท่ากับ 1. สัญญาณ เอาท์พุท จะต่างเฟสกับผลรวมของสัญญาณ อินพุทอยู่ 180° ทั้งนี้เพราะเป็นวงจรขยายแบบ กลับต้วถั่ว

อินพุทจะมีอยู่ 3 ขั้ว ซึ่งจะท่อบรับสัญญาณจากวงจร เลื่อนเฟสต่างหลัง 120° ขั้วหนึ่ง ท่อบรับสัญญาณจากวงจร เลื่อนเฟสหน้าหน้า 120° ขั้วหนึ่ง และท่อบรับสัญญาณอ้างอิงจากวงจรลดแรงกั้นไฟฟ้า อีกขั้วหนึ่ง ซึ่งจะเอาสัญญาณไหนท่อบรับเข้ากับขั้วไหนก็ไ้

4.4.2 การทดลอง

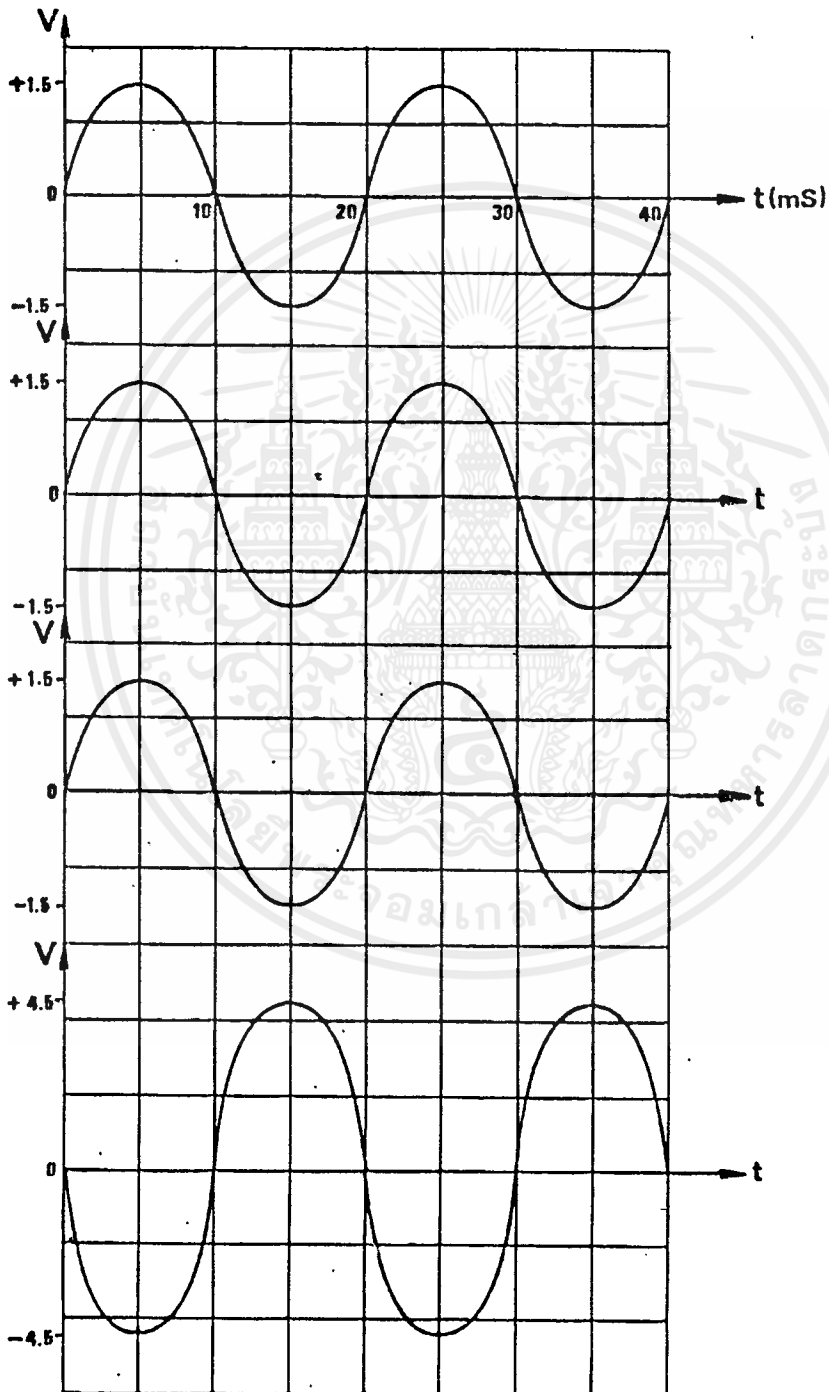
สัญญาณที่เข้าสู่วงจรรวมสัญญาณไฟฟ้านี้ จะมีอยู่ 2 กรณีคือ กรณีที่สัญญาณทั้งสามสัญญาณ มีเฟสตรงกัน และกรณีที่สัญญาณทั้งสามสัญญาณต่างเฟสกัน 120° ทั้งนี้ในการทดลองจึงทดลองใน 2 กรณี

- เมื่อสัญญาณ อินพุตต่างเฟสกัน 120°



รูปที่ 4.7 สัญญาณเมื่อ อินพุตต่างเฟสกัน 120°

- เมื่อสัญญาณ อินพุตมีเฟสตรงกัน

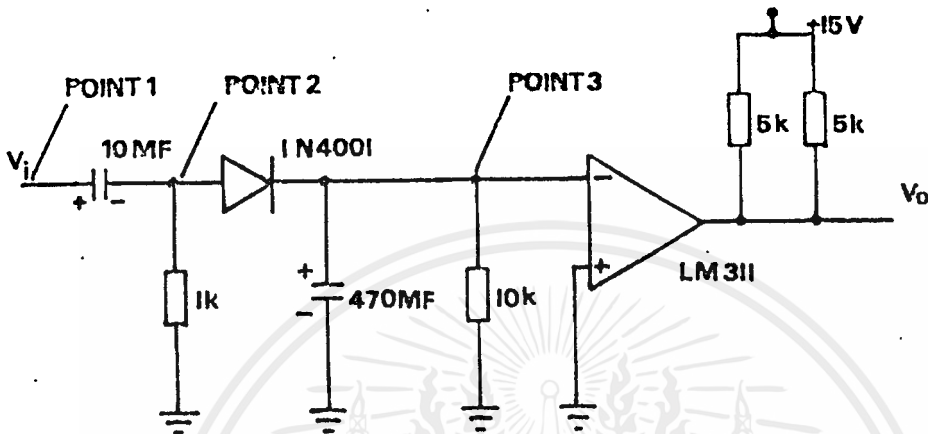


รูปที่ 4.8 สัญญาณเมื่อ อินพุตเฟสตรงกัน

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5 วงจรเปรียบเทียบ (Comparator)

4.5.1 การทำงานของวงจร



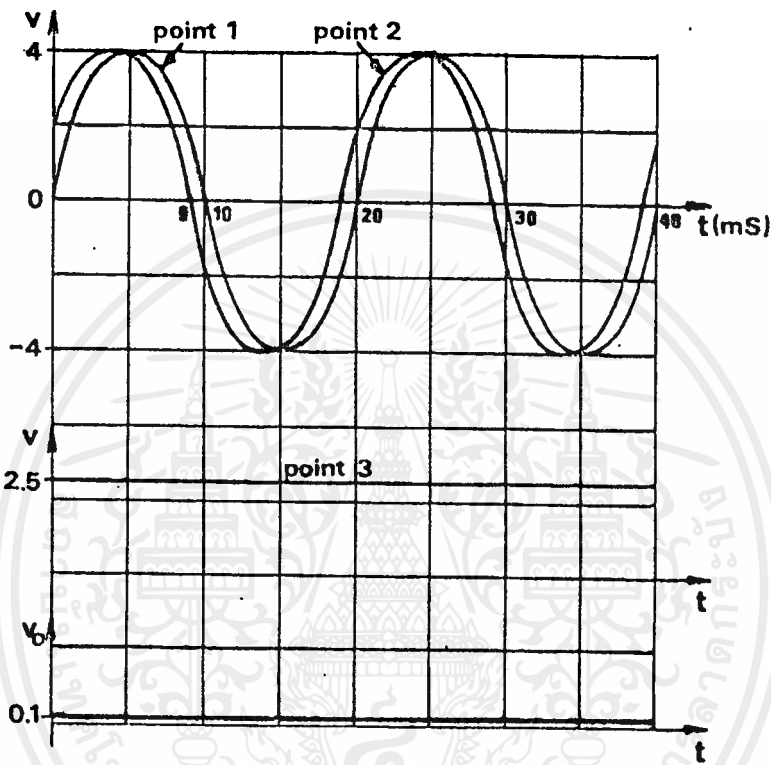
รูปที่ 4.9 วงจรเปรียบเทียบ

วงจรเปรียบเทียบดังรูปที่ 4.9 จะใช้ ออปแอมป์ LM 311 ซึ่งถือเป็นวงจรขยายแบบ เปิดวงจร (Open loop) โดยที่ขา ลบ(-) ของ ออปแอมป์ จะต่อกับสัญญาณ อินพุตส่วนลบ บวก (+) จะต่อกับศักย์ค่าไฟฟ้้าอ้างอิงซึ่งในที่นี้ใช้ 0 โวลต์ หรือต่อกับ กราวด์นั่นเอง

สัญญาณ อินพุตซึ่งเป็นไฟฟ้้ากระแสสลับ ก่อนที่จะเข้าสู่ขา ลบ(-) ของออปแอมป์ จะถูก เรียงกระแสให้เป็นไฟฟ้้ากระแสตรงก่อนสัญญาณ เอาท์พุทจะมีอยู่เพียง 2 ระดับคือ เป็น 0 โวลต์ ค่าหนึ่ง และมีค่าสูงกว่า 0 โวลต์อีกค่าหนึ่ง จากวงจรเมื่อสัญญาณ อินพุต เป็น 0 โวลต์ เท่ากับศักย์ค่าไฟฟ้้าอ้างอิง ออปแอมป์ จะไม่ทำงานคือไม่มีกระแสไฟฟ้้าไหล เข้าหรือออกจาก ออปแอมป์เสมือนกับว่า ออปแอมป์ เกิดการ เปิดวงจรดังนั้นสัญญาณ เอาท์พุท จึงมีค่าเท่ากับ ไฟเลี้ยงวงจรเมื่อสัญญาณ อินพุตมีค่ามากกว่า 0 โวลต์ จะมีกระแสไหลเข้า ทางขาเอาท์พุทของ ออปแอมป์เสมือนกับว่า ออปแอมป์ท่วงจร เอาท์พุทจึงทำให้สัญญาณเอาท์พุท มีค่าเป็น 0 โวลต์

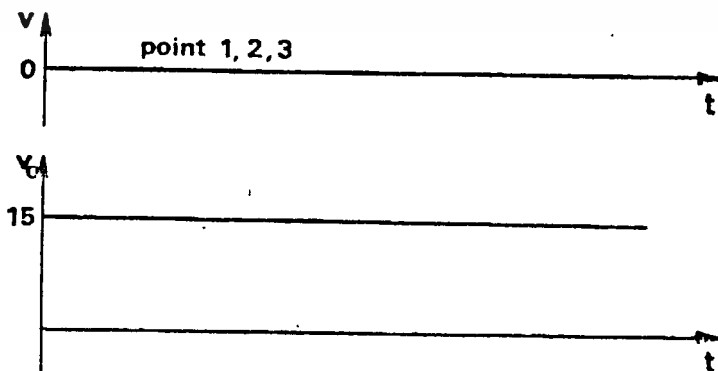
4.5.2 การทดลอง

- เมื่อสัญญาณ อินพุตมีค่าแรงดันสูงสุด สูงกว่า 2.5 โวลต์จะเกิดสัญญาณ ดังรูปที่ 4.10 ก.



รูปที่ 4.10 ก. สัญญาณที่รบกวน. ๓.๓ ในรูปที่ 4.๙ เมื่ออินพุตมากกว่า 2.5 โวลต์

- เมื่อสัญญาณ อินพุตมีค่า 0-2.5 โวลต์

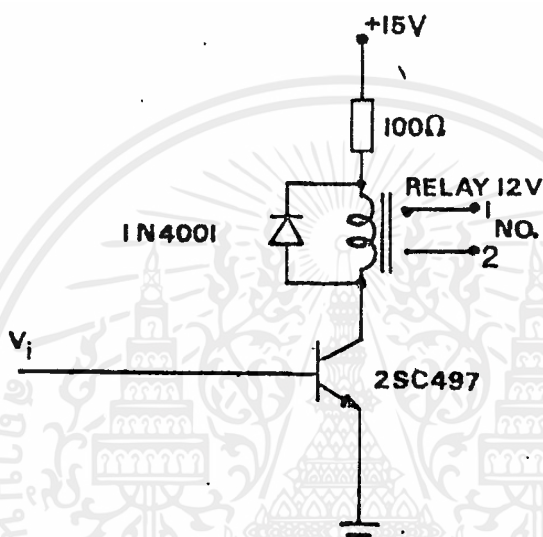


รูปที่ 4.10 ข. สัญญาณเมื่อ อินพุต 0-2.5 โวลต์

∴ เมื่ออินพุต 0–2.5 โวลต์ เอาท์พุทเท่ากับ ไฟเลี้ยงวงจร
เมื่ออินพุทมากกว่า 0.5 โวลต์ เอาท์พุทเท่ากับ 0 โวลต์

4.6 วงจรกักตอสวิตช์ (Switch Controller)

4.6.1 การทำงานของวงจร



รูปที่ 4.11 วงจรกักตอสวิตช์

วงจรนี้จะรับสัญญาณจากวงจร เปรียบเทียบแรงดัน ถ้าวงจร เปรียบเทียบมี
เอาท์พุท เป็น 0 โวลต์ ทราานซิสเตอร์จะเปิดวงจรทำให้รีเลย์ ไม่ทำงาน แต่ถ้าเอาท์พุท
ของวงจร เปรียบเทียบมีแรงดันไฟฟ้า รีเลย์ก็จะทำงานการนำไปใช้งานจะต่อจากขั้ว 1
และ 2 ของ รีเลย์ไปใช้ในการกักตอสวิตช์

รีเลย์ที่ใช้มีขนาด 12 โวลต์ โดยที่ไฟเลี้ยงวงจรเท่ากับ 15 โวลต์ดังนั้นจึงต้อง
ต่อ ความต้านทาน 100Ω โอห์ม เพื่อลดแรงดันที่รีเลย์ให้มาเหลือ 12 โวลต์

4.6.2 การทดลอง

เมื่อไม่มีแรงดันอินพุทวงจรก็จะไม่ทำงาน ขั้ว 1 และ 2 จะเปิดวงจรเมื่อมีแรง
ดันอินพุทวงจรจะทำงานโดยที่แรงดันคร่อม รีเลย์มีค่า 12 โวลต์ แรงดันคร่อม ตัวต้านทาน
3 โวลต์ ส่วนแรงดันคร่อมทรานซิสเตอร์ 0 v.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

