



เครื่องวัดค่าเพาเวอร์แฟกเตอร์
POWER FACTOR METER

โดย

นายสิฏฐา จันทรแย้ม

นายสิทธา ศุภเลิศไพสิฐ

.....	วัน เดือน ปี..... 17 พ.ค. 2024
.....	เลขทะเบียน..... 034723
.....	เลขเรียกหนังสือ..... T 31020 66

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมอิเล็กทรอนิกส์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

ปีการศึกษา 2537

ปริญญาโทปีการศึกษา 2537

ภาควิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง
เรื่อง เครื่องวัดค่าเพาเวอร์แฟกเตอร์

ผู้จัดทำ

1. นายสิฏฐา จันทรแย้ม 34108421
2. นายสิทธิ ศุภเลิศไพสิฐ 34108422



(ศ.ดร.วัลลภ สุระกำพลธร)

อาจารย์ที่ปรึกษา

เครื่องวัดค่าเพาเวอร์แฟกเตอร์

สิฏฐา จันทรเยี่ยม

สิทธา ศุภเลิศไพสิฐ

ศ.ดร.วัลลภ สุระกำพลธร อาจารย์ที่ปรึกษา

ปีการศึกษา 2537

บทคัดย่อ

ในปฏิญานิพนธ์ฉบับนี้ เรียบเรียงขึ้นจากผลงานที่ได้พัฒนาขึ้นเป็นเครื่องวัดค่าเพาเวอร์แฟกเตอร์ (power factor meter) แบบดิจิทัล (digital) โดยการนำสัญญาณแรงดันไฟฟ้าและกระแสผ่านวงจรซีโรครอสซิงดีเทกเตอร์ (zero-crossing detector) เพื่อแปลงสัญญาณให้เป็นสัญญาณดิจิทัล (digital) และนำสัญญาณดังกล่าวไปผ่านวงจรปรับระดับแรงดันไฟฟ้า เพื่อปรับระดับแรงดันไฟฟ้าให้อยู่ในช่วง 0-5 โวลต์ จากนั้นจึงผ่านสัญญาณไปยังวงจรเฟสคอมพาราเตอร์ (phase comparator) เพื่อหาค่าความต่างเฟสของสัญญาณทั้งสอง นำสัญญาณไปผ่านวงจรผ่านความถี่ต่ำ (low-pass filter) ซึ่งจะทำการแปลงค่าความต่างเฟสเป็นแรงดันไฟฟ้าคงที่ซึ่งอยู่ในช่วง 0-3 โวลต์ ค่าที่ได้จะมากหรือน้อยขึ้นอยู่กับความต่างเฟสของสัญญาณทั้งสอง ซึ่งค่าความต่างเฟสนี้ก็คือค่าเพาเวอร์แฟกเตอร์ สัญญาณที่ได้จากวงจรผ่านความถี่ต่ำนำไปเข้าวงจรแปลงสัญญาณอนาล็อก (analog) ไปเป็นสัญญาณดิจิทัลขนาด 8 บิต (bit) เพื่อนำไปเข้าส่วนไมโครโปรเซสเซอร์ (microprocessor) เป็นค่าของเพาเวอร์แฟกเตอร์ที่ต้องการจริง แล้วแสดงผลด้วยจอ LCD

เครื่องวัดค่าเพาเวอร์แฟกเตอร์ที่ได้พัฒนาขึ้นมานี้เป็นเครื่องวัดที่มีความสะดวกในการใช้งาน เป็นเครื่องวัดค่าที่ไม่ขึ้นอยู่กับความถี่ วงจรประกอบด้วยอุปกรณ์เพียงเล็กน้อยทำให้ราคาไม่แพงเกินไป

POWER FACTOR METER

Sittha Chanyam

Sitha Suppalertpisit

Professor Doctor Wallop Surakampolthorn Advisor

1994

Abstract

This thesis is an application of Zero-Crossing Detector to develop Digital Power Factor Meter. The Zero-Crossing Detector converts voltage level in order to adjust voltage level interval 0-5 volts and passes signal through Phase Comparator. The Low-pass Filter receives the different signal form Phase Comparator and converts to voltage level interval 0-3 volts. The output from Low-pass Filter is converted by Analog to Digital Converter to 8-bit data. The microprocessor calculates 8-bit data to Power Factor value shown by LCD module. This project is developed for convenient use because it does not depend on frequency and low cost.

สารบัญ

บทที่ 1 บทนำ	1
1.1 เพาเวอร์แฟกเตอร์	1
1.2 เครื่องวัดค่าเพาเวอร์แฟกเตอร์	2
1.2.1 เครื่องวัดค่าเพาเวอร์แฟกเตอร์แบบไดนาโมมิเตอร์เฟสเดียว	3
1.2.2 เครื่องวัดค่าเพาเวอร์แฟกเตอร์แบบไดนาโมมิเตอร์ สำหรับโหลด 3 เฟสสมดุล	4
1.2.3 เครื่องวัดค่าเพาเวอร์แฟกเตอร์แบบอัลเทอร์เนทีฟ	4
1.3 เครื่องวัดค่าเพาเวอร์แฟกเตอร์แบบดิจิตอล	5
บทที่ 2 ทฤษฎี	6
2.1 วงจรซีโรครอสซิงดีเทกเตอร์	8
2.1.1 non-inverting zero-crossing detector	8
2.1.2 inverting zero-crossing detector	8
2.1.3 zero-crossing detector with hysteresis upper-threshold voltage	9
lower-threshold voltage	10
2.2 วงจรปรับระดับสัญญาณโดยใช้ไดโอด	11
2.3 วงจรผ่านความถี่ต่ำ	12
2.4 successive approximation ADC	13
2.5 โครงสร้างสถาปัตยกรรมของ MCS-48	17
2.5.1 โครงสร้างภายนอก	17
2.5.2 โครงสร้างภายใน	22
2.5.3 หน่วยความจำ	23
2.5.4 ส่วนติดต่ออุปกรณ์ภายนอก	26
2.5.5 สัญญาณการตรวจสอบและสัญญาณอินเทอร์รัพท์	29
2.5.6 ตัวนับโปรแกรมและสแตก	29
2.5.7 การอินเทอร์รัพท์	31
2.5.8 ตัวจับเวลา/ตัวนับ	33
2.5.9 สัญญาณนาฬิกาและวงจรฐานเวลา	34
2.5.10 การรีเซท	35

2.6 แนวทางการใช้งาน LCD module	36
2.6.1 การต่อเข้ากับระบบไมโครโปรเซสเซอร์	38
2.6.2 ชุดควบคุมคำสั่งและการแสดงข้อความ	39
บทที่ 3 การคำนวณและการสร้าง	45
บทที่ 4 การทดลองและผลการทดลอง	60
บทที่ 5 บทวิจารณ์และสรุป	61
ภาคผนวก	
วงจรเครื่องวัดค่าเพาเวอร์แฟกเตอร์	
กิตติกรรมประกาศ	
บรรณานุกรม	

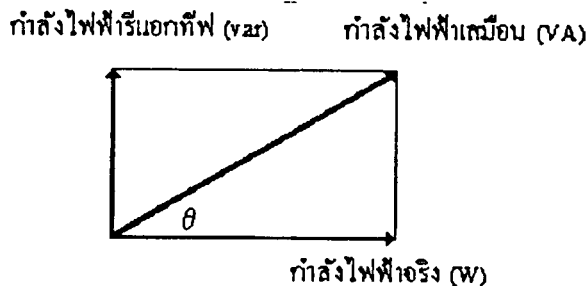
บทที่ 1

บทนำ

1.1 เพาเวอร์แฟกเตอร์

กำลังงานในระบบไฟฟ้ากระแสสลับสามารถแบ่งออกได้เป็น 2 ส่วนด้วยกัน คือ กำลังไฟฟ้าจริง (real power หรือ active power) มีหน่วยวัตต์เป็นวัตต์ (watt) เป็นกำลังงานที่สามารถเปลี่ยนแปลงโดยอุปกรณ์ไฟฟ้าไปเป็นพลังงานรูปอื่นได้ เช่น ความร้อน แสงสว่าง หรือกำลังงานกล และอีกส่วนหนึ่งคือ กำลังไฟฟ้ารีแอกทีฟ (reactive power) มีหน่วยวัตต์เป็น วาร์ (var) เป็นกำลังงานที่ไม่สามารถเปลี่ยนไปเป็นพลังงานรูปอื่นได้ แต่อุปกรณ์ไฟฟ้าที่ต้องทำงานโดยอาศัยสนามแม่เหล็ก เช่น หม้อแปลงไฟฟ้า มอเตอร์ (motor) ต้องใช้กำลังไฟฟ้ารีแอกทีฟ นี้สร้างสนามแม่เหล็ก ผลรวมของกำลังงานทั้ง 2 ส่วนนี้เรียกว่า กำลังไฟฟ้าเสมือน (apparent power) มีหน่วยเป็น โวลต์แอมแปร์ (VA) เป็นกำลังงานที่แหล่งจ่ายกำลังงานไฟฟ้าต้องจ่ายให้กับอุปกรณ์ไฟฟ้าต่างๆ และมีขนาดเท่ากับผลคูณของกระแสไฟฟ้าในวงจรกับแรงดันของแหล่งจ่ายกำลังไฟฟ้า อัตราส่วนของกำลังงานจริงต่อกำลังงานปรากฏ เราเรียกว่าเพาเวอร์แฟกเตอร์ ซึ่งเป็นตัวบอกให้ทราบว่าอุปกรณ์ไฟฟ้าต่างๆ ใช้กำลังงานจริงเป็นสัดส่วนเท่าไรเมื่อเทียบกับกำลังงานปรากฏ ดังนั้นในระบบไฟฟ้าที่มีค่าเพาเวอร์แฟกเตอร์สูง จะมีความสามารถหรือประสิทธิภาพในการทำงานดีกว่าระบบไฟฟ้าที่มีค่าเพาเวอร์แฟกเตอร์ต่ำกว่า

ผลรวมกำลังไฟฟ้าทั้งสองรวมเข้ากันทางด้านเฟเซอร์ (phasor) แสดงดังรูปที่ 1.1



รูปที่ 1.1 ค่ากำลังไฟฟ้าที่วัดได้ในระบบไฟฟ้ากระแสสลับ

เพาเวอร์แฟกเตอร์อาจเป็นแบบนำหน้าหรือตามหลัง (lead/lag) ก็ได้ ทั้งนี้ขึ้นอยู่กับทิศทางการไหลของกำลังงานที่ทำให้เกิดงานและกำลังงานรีแอกทีฟ ถ้ากำลังงานทั้งสองส่วนนี้ไหลไปในทิศทางเดียวกัน ค่าเพาเวอร์แฟกเตอร์ที่จุดนั้นจะเป็นแบบตามหลัง แต่ถ้าไหลไปคนละทิศทางแล้ว ค่าเพาเวอร์แฟกเตอร์ที่จุดนั้นจะเป็นแบบนำหน้า เนื่องจากตัวเก็บประจุ (capacitor) เป็นแหล่งกำเนิดกำลังงานรีแอกทีฟเพียงอย่างเดียวมันจึงมีค่าเพาเวอร์แฟกเตอร์แบบนำหน้าเสมอ สำหรับมอเตอร์เหนี่ยวนำจะมีค่าเพาเวอร์แฟกเตอร์เป็นแบบตามหลัง เพราะมันต้องการทั้งกำลังงานที่ทำให้เกิดงานและกำลังงานรีแอกทีฟ (ไหลเข้ามอเตอร์ทั้ง 2 ส่วน) สำหรับซิงโครนัสมอเตอร์ (synchronous motor) ที่ถูกกระตุ้นเกินขนาด (overexcited) นั้นสามารถจ่ายกำลังงานรีแอกทีฟเข้าสู่ระบบไฟฟ้าได้ แต่กำลังงานที่ทำให้เกิดงานต้องไหลเข้ามอเตอร์เสมอ ดังนั้นมันจึงมีค่าเพาเวอร์แฟกเตอร์เป็นแบบนำหน้าได้

ในระบบไฟฟ้านั้นค่าเพาเวอร์แฟกเตอร์เป็นเรื่องที่สำคัญ ดังนั้นจะอย่างไรที่จะทำให้เราทราบค่าเพาเวอร์แฟกเตอร์ได้ ปกติแล้วค่าเพาเวอร์แฟกเตอร์สามารถคำนวณได้ แต่เนื่องในการใช้งานจริงนั้น ค่าเพาเวอร์แฟกเตอร์มีการเปลี่ยนแปลงไม่แน่นอน ดังนั้นจึงเป็นการยากที่จะต้องมาคอยคำนวณค่าเพาเวอร์แฟกเตอร์ทุกครั้งที่มีการเปลี่ยนแปลง จึงมีการสร้างเครื่องมือวัดเพื่อช่วยให้เกิดความสะดวกในการวัดมากขึ้น ซึ่งก็คือบอกค่าเพาเวอร์แฟกเตอร์โดยไม่ต้องมาสังเกตการเปลี่ยนแปลงแล้วคำนวณมาอีกให้เสียเวลา

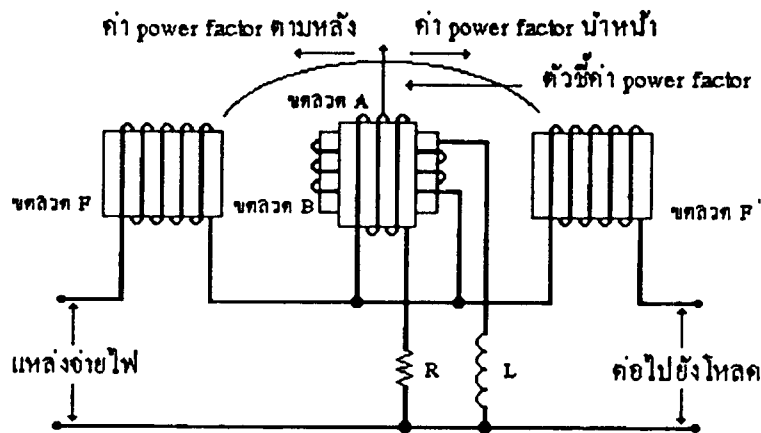
1.2 เครื่องวัดค่าเพาเวอร์แฟกเตอร์

ในการสร้างเครื่องวัดค่าเพาเวอร์แฟกเตอร์นั้น สามารถสร้างได้หลายแบบ ซึ่งจะได้อธิบายคร่าวๆ เพียง 3 แบบ ดังต่อไปนี้

1. เครื่องวัดค่าเพาเวอร์แฟกเตอร์แบบไดนาโมมิเตอร์เฟสเดียว (dynamometer type single-phase power factor meter)
2. เครื่องวัดค่าเพาเวอร์แฟกเตอร์แบบไดนาโมมิเตอร์สำหรับโหลด 3 เฟสสมดุล (dynamometer power factor meter for balanced three phase load)
3. เครื่องวัดค่าเพาเวอร์แฟกเตอร์แบบอัลเทอร์เนทีฟ (alternative power factor meter)

1.2.1 เครื่องวัดค่าเพาเวอร์แฟกเตอร์แบบไดนาโมมิเตอร์เฟสเดียว

เครื่องวัดค่าเพาเวอร์แฟกเตอร์ชนิดนี้ประกอบด้วยขดลวด F ซึ่งเป็นส่วนที่อยู่กับที่ และขดลวด A และ B ที่วางตั้งฉากกัน 90° เป็นส่วนเคลื่อนที่ แกนของขดลวด A และ B จะถูกต่อกับเข็มชี้บอกค่าเพาเวอร์แฟกเตอร์ดังในรูปที่ 1.2 จำนวนรอบ, ขนาด และวัสดุที่ใช้ทำขดลวด A และ B จะเหมือนกันหมด ขดลวด A และ B เราเรียกว่าขดลวดเคลื่อนที่ (moving coil) ในวงจรของเครื่องวัดค่าเพาเวอร์แฟกเตอร์ชนิดนี้ ขดลวด A จะต่ออนุกรมกับตัวต้านทานที่มีค่าความต้านทานสูงแล้วขนานกับโหลด (load) ที่จะวัด ส่วนขดลวด B จะอนุกรมกับค่าความเหนี่ยวนำของตัวเหนี่ยวนำ L แล้วขนานกับโหลดเหมือนกัน



รูปที่ 1.2 เครื่องวัดค่าเพาเวอร์แฟกเตอร์แบบไดนาโมมิเตอร์เฟสเดียว

ค่าความเหนี่ยวนำ L และค่าความต้านทาน R จะทำหน้าที่ปรับให้กระแสไหลในขดลวด A และ B เท่ากันที่ความถี่ของไฟฟ้ากระแสสลับที่จะวัด สนามแม่เหล็กที่เกิดขึ้นบนขดลวด F จะเป็นปฏิภาคโดยตรงกับกระแสที่ไหลไปยังโหลดที่จะวัด

หลักการทำงาน

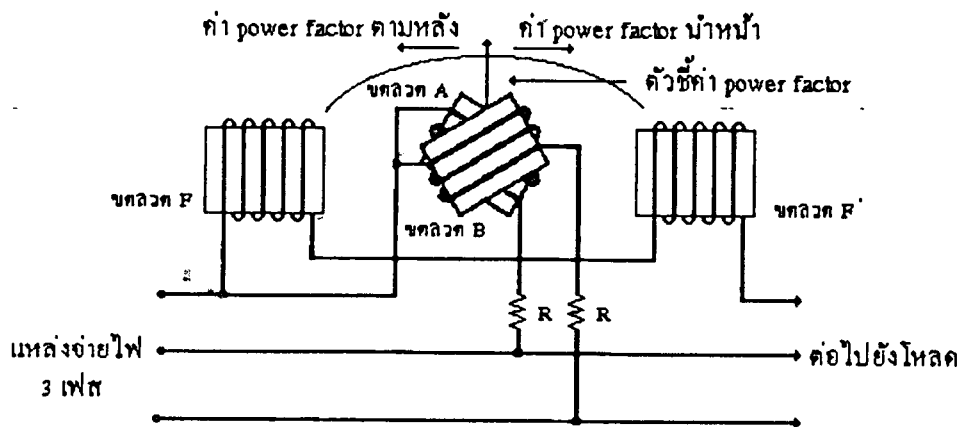
เมื่อกระแสและแรงดันของโหลดที่จะวัดค่าเพาเวอร์แฟกเตอร์มีเฟส (phase) ตรงกัน กระแสในขดลวด A ก็จะมีเฟสตรงกับในขดลวด F และ F' ขณะที่กระแสในขดลวด B จะมีเฟสตามแรงดัน 90° ซึ่งก็จะตามหลังกับกระแสในขดลวด F 90° ด้วยเช่นกัน แรงกิริยาในขดลวด A จะทำให้เข็มชี้อยู่ในภาวะสมดุล ในตำแหน่งที่ตั้งฉากกับขดลวด F ซึ่งขณะนั้นก็คือค่าเพาเวอร์แฟกเตอร์เท่ากับ 1 นั่นเอง

ถ้ากระแสในขดลวด B มีเฟสตรงกับกระแสในขดลวด F ขณะเดียวกัน กระแสในขดลวด A จะมีเฟสตรงข้ามกับกระแสในขดลวด F 90° ระบบขับเคลื่อนจะผลักดันให้ขดลวด B หมุนไปจนกระทั่งตั้งฉากกับแกนของขดลวด F เป็นผลให้เข็มของมิเตอร์ (meter) เคลื่อนที่ซึ่งไปยังตำแหน่งที่แสดงค่าเพาเวอร์แฟกเตอร์เป็นศูนย์ นั่นคือค่าเพาเวอร์แฟกเตอร์เป็น 1 ขดลวด A จะตั้งฉากกับขดลวด F ถ้าเป็นศูนย์ ขดลวด B จะตั้งฉากกับขดลวด F แทน

ในกรณีที่ค่าเพาเวอร์แฟกเตอร์อยู่ระหว่าง 0-1 ระบบขับเคลื่อนภายในมิเตอร์จะทำให้เข็มช้อยู่กึ่งกลาง ตัวชี้ค่าเพาเวอร์แฟกเตอร์จะเคลื่อนที่ท่ามุม $90^\circ - \theta$ กับแกนของขดลวด F โดย θ คือค่าเฟสของวงจรที่ทำการวัดเสมอ

1.2.2 เครื่องวัดค่าเพาเวอร์แฟกเตอร์แบบไดนาโมมิเตอร์สำหรับโหลด 3 เฟสสมดุล

มิเตอร์แบบนี้ใช้วัดค่าเพาเวอร์แฟกเตอร์ของระบบไฟฟ้า 3 เฟส มีโครงสร้างดังรูปที่ 1.3 ขดลวดเคลื่อนที่ A และ B จะวางท่ามุมกัน 120° และต่อคร่อมสายไฟฟ้าที่มีเฟสต่างกันของวงจรจ่ายแรงดัน ส่วนขดลวดไม่เคลื่อนที่ (ขดลวด F) จะต่ออนุกรมเข้ากับสายร่วมเพื่อตรวจจับกระแสไฟฟ้าในสาย ถ้าหากค่าเพาเวอร์แฟกเตอร์เท่ากับหนึ่ง ในสายแรงดันแต่ละเส้นจะมีแรงดันต่างเฟสกัน 120° ก็จะทำให้เข็มของมิเตอร์ชี้ที่ตำแหน่งกึ่งกลางตั้งฉากกับขดลวดเคลื่อนที่ทั้งสอง



รูปที่ 1.3 เครื่องวัดค่าเพาเวอร์แฟกเตอร์แบบไดนาโมมิเตอร์สำหรับโหลด 3 เฟสสมดุล

1.2.3 เครื่องวัดค่าเพาเวอร์แฟกเตอร์แบบอัลเตอร์เนทีฟ

เครื่องวัดค่าเพาเวอร์แฟกเตอร์แบบนี้มีโครงสร้างภายในเป็นมิเตอร์แบบแกนเหล็กเคลื่อนที่ กล่าวคือเป็นมิเตอร์ที่ใช้แผ่นเหล็กอ่อน (เหล็กที่ใช้ทำเป็นแม่เหล็กชั่วคราว) ยึดติดกับเข็มเป็นส่วนเคลื่อนที่สำหรับขดลวดนั้นจะอยู่กับที่

ภายในมิเตอร์จะประกอบด้วยขดลวด 3 ชุด วางทำมุมกัน 120° เฟลาของเข็มจะยึดติดกับแผ่นเหล็กอ่อนมีมุมเอียงขึ้นละ 120° 2 ชั้น สำหรับขดลวด 3 ชุดนี้จะต่อกับแรงดันทั้ง 3 เฟส เพื่อทำให้เกิดทอร์ก (torque) บนแกนเฟลาของเข็มซี่ ดังนั้นจึงไม่จำเป็นที่จะต้องมีส่วนสปริงที่ควบคุมตำแหน่งเข็ม เข็มของมันสามารถจะกวาดรอบหน้าปัดได้ 360° ส่วนใหญ่ เครื่องวัดค่าเพาเวอร์แฟกเตอร์ชนิดนี้จะใช้กับระบบไฟฟ้า 3 เฟสเท่านั้น

ข้อเสียของเครื่องวัดค่าเพาเวอร์แฟกเตอร์ชนิดนี้ก็คือ ความเที่ยงตรงจะน้อยกว่า 2 แบบแรกที่กำลังกล่าวมาแต่ข้อดีก็คือมีความทนทานมากกว่า

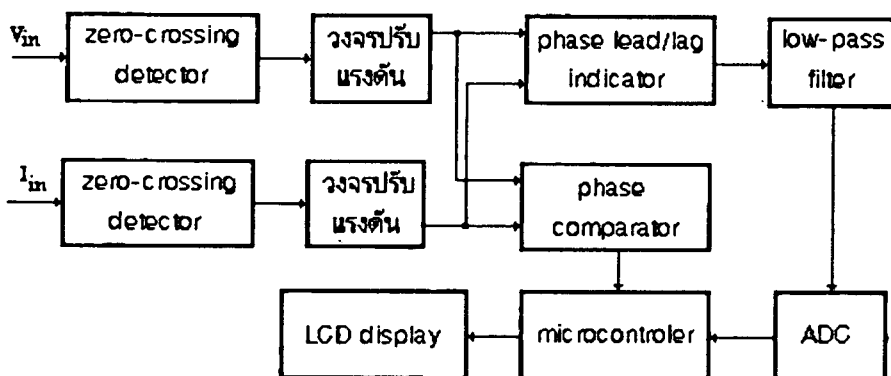
เครื่องวัดค่าเพาเวอร์แฟกเตอร์ที่กำลังกล่าวมานี้ เป็นตัวอย่างของเครื่องวัดค่าเพาเวอร์แฟกเตอร์ อย่างง่ายๆ ในปัจจุบันมีการพัฒนาเป็นระบบดิจิตอลที่สามารถแสดงผลเป็นตัวเลข ทำให้มีความแม่นยำเที่ยงตรงมากขึ้น

1.3 เครื่องวัดค่าเพาเวอร์แฟกเตอร์แบบดิจิตอล

ตามปกติเครื่องวัดค่าเพาเวอร์แฟกเตอร์ทางอิเล็กทรอนิกส์ (electronics) จะมีความไวขึ้นอยู่กับไฟ line และความถี่ที่เปลี่ยนแปลงไป เครื่องวัดค่าเพาเวอร์แฟกเตอร์ที่เป็นดิจิตอลจะต้องการวงจรทางอิเล็กทรอนิกส์ที่เป็นเชิงซ้อน แต่เครื่องวัดค่าเพาเวอร์แฟกเตอร์ในที่นี้จะเป็นอย่างง่าย และไม่ขึ้นอยู่กับความถี่ วงจรจะประกอบไปด้วยอุปกรณ์เพียงเล็กน้อย เช่น ออปแอมป์ (op-amp) และ เกท (gate) ต่างๆ ค่าเพาเวอร์แฟกเตอร์ที่ได้จะเป็นค่าไฟฟ้ากระแสตรงอยู่ในช่วง 0-3 โวลต์

บทที่ 2 ทฤษฎี

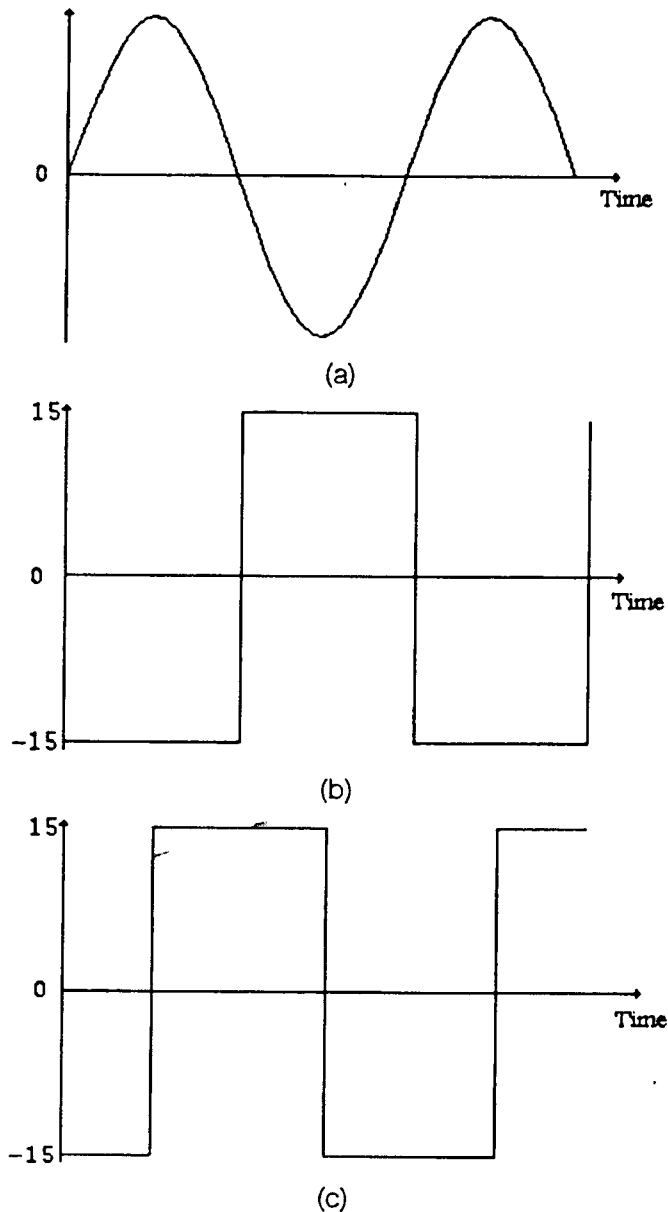
หลักการทํางาน



รูปที่ 2.1 แผนภาพแสดงการทํางานของเครื่องวัดค่าเฟาเวอร์เพกเตอร์

วงจรรีโครอสซิงดีเทกเตอร์เป็นวงจรถ่ายสัญญาณรูปซายน์ที่เป็นสัญญาณอินพุตดังรูปที่ 2.2(a) ให้เป็นสัญญาณบวกและลบแบบดิจิตอลดังแสดงในรูปที่ 2.2 (b) โดยที่เฟสของสัญญาณไม่เปลี่ยนแปลง โดยที่ไอซี (IC) LM 311 เป็นไอซีเปรียบเทียบแรงดัน เมื่อขาที่ 3 มีสัญญาณเข้ามา ถ้าแรงดันมีค่ามากกว่าขาที่ 2 จะทำให้เอาต์พุตออกมาเป็นบวก (ประมาณ +5 โวลต์) ซึ่งแรงดันที่เข้ามาน้อยกว่าที่ขา 2 จะให้เอาต์พุตออกมาเป็นลบ (ประมาณ -5 โวลต์) จากนั้นจะผ่านสัญญาณที่ได้ไปยังวงจรรีปรับระดับแรงดัน ซึ่งจะทำการปรับระดับแรงดันให้อยู่ในช่วง 0-5 โวลต์ดังแสดงในรูปที่ 2.2 (c) โดยใช้ไดโอดเป็นตัวผ่านสัญญาณทางด้านซีกบวกออกมาเพื่อเป็นสัญญาณอินพุตให้กับวงจรรีเปรียบเทียบการเลื่อนของสัญญาณในวงจรรีเปรียบเทียบนี้จะทำการเปรียบเทียบการเลื่อนออกจากสัญญาณทั้งสอง โดยใช้ XOR เกทเป็นตัวเปรียบเทียบ ค่าผิดพลาดที่ออกมา (ERROR) จะเป็นค่าของการเลื่อนห่างออกจากกันของสัญญาณทั้งสอง ซึ่งก็คือค่าของมุมที่ห่างกันนั่นเอง เมื่อได้ค่าผิดพลาดออกมาแล้วจะผ่านสัญญาณที่ได้ต่อไปยังวงจรรองความถี่สูงผ่านความถี่ต่ำ ในการออกแบบนี้ต้องการให้มีความถี่คัทออฟที่ความถี่ใกล้ๆศูนย์ให้มากที่สุด เพราะแรงดันที่ต้องการที่เอาต์พุตจะต้องเป็นแรงดันคงที่ที่มีความถี่เป็นศูนย์ ฉะนั้นจึงใช้วงจรรองความถี่ต่ำต่อกันถึง 3 ชุดเพื่อให้ได้สัญญาณที่เรียบซึ่งจะมีแรงดันที่เอาต์พุตอยู่ในช่วง 0-3 โวลต์ จากนั้นนำค่าแรงดันที่ได้ไปแปลงเป็นสัญญาณดิจิตอลขนาด 8 บิต โดยใช้ไอซีแปลงสัญญาณ ส่งค่าดิจิตอลที่ได้ไปยังส่วนของไมโครคอนโทรลเลอร์ สัญญาณดิจิตอลที่ได้จากการแปลงจะถูกส่ง

ดิจิตอลที่ได้ไปยังส่วนของไมโครคอนโทรลเลอร์ สัญญาณดิจิตอลที่ได้จากการแปลงจะถูกส่งเข้ามาทางเส้นของข้อมูล (DATA BUS) โดยไมโครคอนโทรลเลอร์ นี้จะทำการคำนวณค่าเพาเวอร์เฟกเตอร์จริงๆ ที่ได้ และเป็นตัวจัดการแสดงผลออกทางจอ LCD ผ่านทางพอร์ต 1 (PORT1) ของไอซี

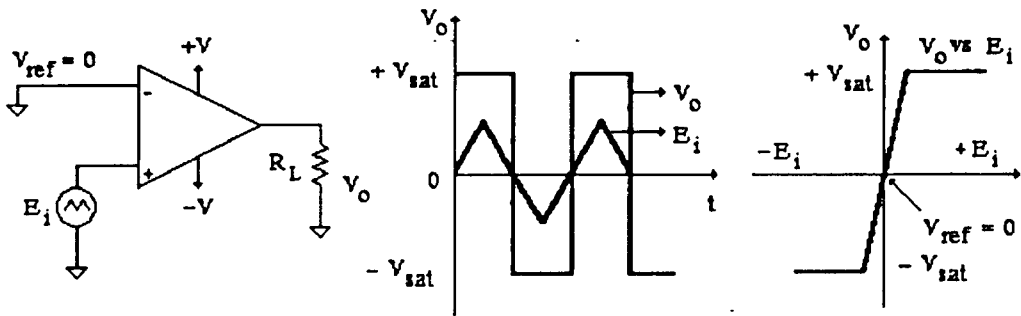


รูปที่ 2.2 กราฟแสดงสัญญาณของวงจรต่างๆ

2.1 วงจรซีโรครอสซิงดีเทกเตอร์

2.1.1 Noninverting Zero-Crossing Detector

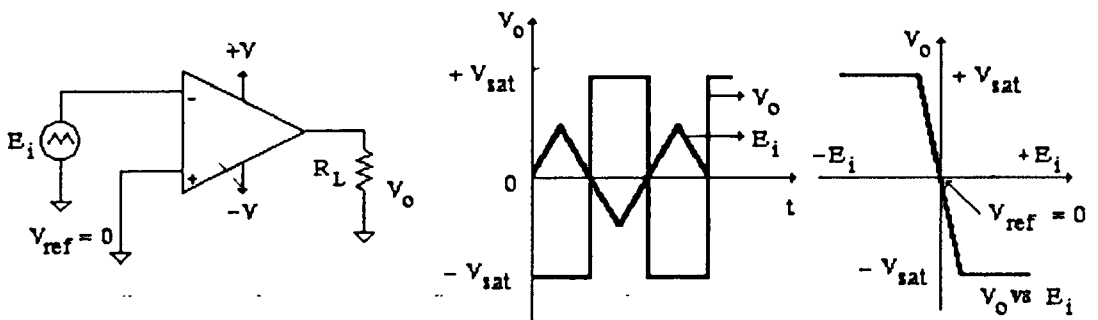
จากรูปที่ 2.3 ออปแอมป์เป็นตัวคอมพาราเตอร์ (comparator) โดยที่ขาบวกของออปแอมป์ต่อกับค่าแรงดันไฟฟ้า E_i และต่อกับแรงดันไฟฟ้าอ้างอิง (reference voltage) $V_{ref} = 0$ เมื่อ E_i มีค่ามากกว่า V_{ref} ทำให้ $V_o = +V_{sat}$ เพราะว่าแรงดันไฟฟ้าที่ขาบวกมีค่าเป็นบวกมากกว่าแรงดันไฟฟ้าที่ขาลบ ดังนั้น V_o จึงมีค่าเป็นบวก



รูปที่ 2.3 noninverting zero-crossing detector เมื่อ E_i มากกว่า V_{ref} . $V_o = +V_{sat}$

2.1.2 Inverting Zero-Crossing Detector

ขาลบของออปแอมป์ในรูปที่ 2.4 มีค่าแรงดันไฟฟ้าเท่ากับ E_i และมีแรงดันไฟฟ้าอ้างอิงเป็นศูนย์ ($V_{ref} = 0$) เป็นวงจร inverting zero-crossing detector เมื่อ E_i มากกว่า V_{ref} จะได้ $V_o = -V_{sat}$



รูปที่ 2.4 inverting zero-crossing detector เมื่อ E_i มากกว่า V_{ref} . $V_o = -V_{sat}$

2.1.3 Zero-Crossing Detector with Hysteresis

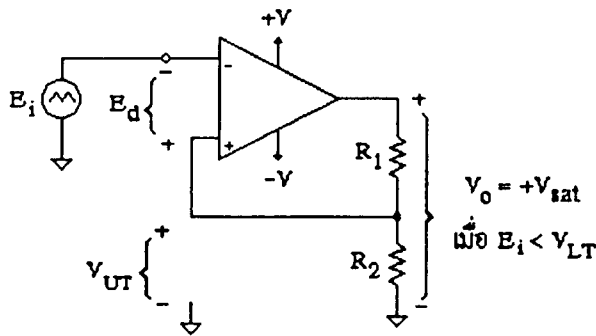
Upper-Threshold Voltage

ในรูปที่ 2.5(a) แรงดันไฟฟ้าเอาต์พุต V_O จะคร่อมระหว่าง R_1 และ R_2 ส่วนหนึ่งของ V_O จะถูกป้อนกลับเข้าที่ขาบวกของออปแอมป์ เมื่อ $V_O = +V_{sat}$ แรงดันไฟฟ้าที่ถูกป้อนกลับจะเรียกว่า upper-threshold voltage V_{UT} . V_{UT} สามารถแสดงโดยใช้ voltage divider ดังนี้

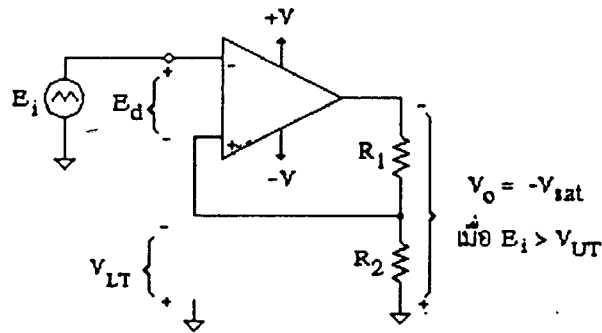
$$V_{UT} = \frac{R_2}{R_1 + R_2} (+V_{sat}) \dots \dots \dots (2.1)$$

เมื่อ E_i มีค่าน้อยกว่า V_{UT} ซึ่งก็คือ ค่าแรงดันไฟฟ้าที่ขาบวกจะมีค่ามากกว่าค่าแรงดันไฟฟ้าที่ขาลบ จะได้ $V_O = +V_{sat}$

ถ้าค่า E_i มีค่าเป็นบวกมากกว่า V_{UT} เพียงเล็กน้อย ค่า E_D จะกลับขั้ว และค่า V_O เริ่มลดลง ในขณะนี้ค่าบางส่วนของ V_O ที่ถูกป้อนกลับไปทางขาบวกจะมีค่าน้อยลง และ E_D มีค่ามากขึ้น V_O จะลดลงอย่างรวดเร็วและจะมีค่าเป็น $-V_{sat}$ วงจรจะคงที่ที่สภาวะดังในรูปที่ 2.5(b)



(a) Upper-threshold voltage, V_{UT}



(b) Lower-threshold voltage, V_{LT}

รูปที่ 2.5 วงจรแสดงถึงค่า V_{UT} , V_{LT}

Lower-Threshold Voltage

เมื่อ V_O มีค่าเป็น $-V_{sat}$ ค่าแรงดันไฟฟ้าที่ป้อนกลับไปที่ขาบวกของออปแอมป์เรียกว่า lower-threshold voltage V_{LT} . V_{LT} คิดได้จากสมการที่ 2.2

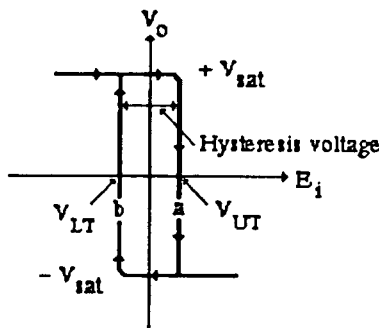
$$V_{LT} = \frac{R_2}{R_1 + R_2} (-V_{sat}) \dots \dots \dots (2.2)$$

ค่าของ V_{LT} จะเป็นลบเมื่อเทียบกับกราวด์ (ground) V_O จะมีค่าเท่ากับ $-V_{sat}$ ในขณะที่ E_i มีค่าเป็นบวกมากกว่า V_{LT} V_O จะเปลี่ยนกลับไปเป็น $+V_{sat}$ เมื่อ E_i มีค่าเป็นลบมากกว่า หรือ มีค่าต่ำกว่า V_{LT}

ค่านิยามของฮิสทีรีซิส (hysteresis)

วิธีมาตรฐานในการแสดงผลการทำงาน

ของตัวคอมพาราเตอร์บนกราฟเพียงกราฟเดียวแทนการใช้กราฟ 2 กราฟเหมือนในรูปที่ 2.3 และ รูปที่ 2.4 โดยการพลอต (plot) ค่า E_i บนแกนนอน และ ค่า V_O บนแกนตั้ง เราก็จะได้ output-input voltage characteristic ดังรูปที่ 2.6

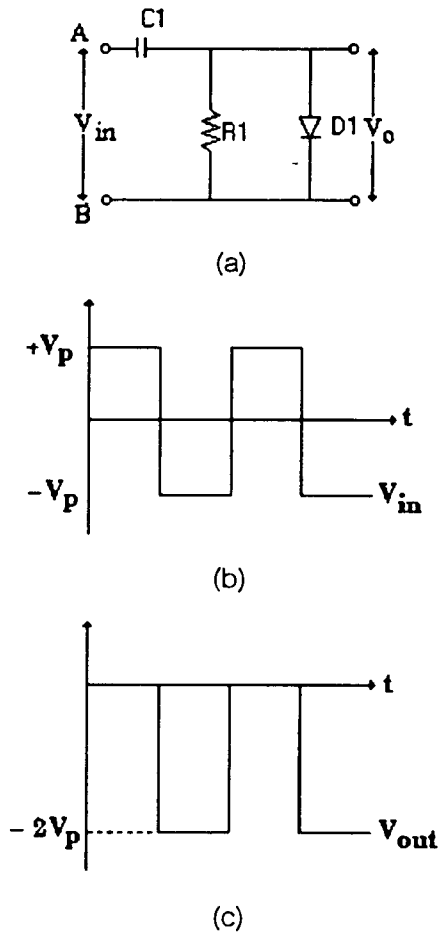


รูปที่ 2.6 กราฟของ V_O กับ E_i แสดงถึงค่าของแรงดันไฟฟ้าฮิสทีรีซิสในวงจรคอมพาราเตอร์

ถ้า E_i มีค่าน้อยกว่า V_{LT} (lower threshold voltage) จะได้ $V_O = +V_{sat}$ และเส้นตรง a แสดงถึง V_O เปลี่ยนแปลงจาก $+V_{sat}$ ไปสู่ $-V_{sat}$ ในขณะที่ E_i มีค่ามากกว่า V_{UT} (upper threshold voltage) เส้นตรง b แสดงถึง V_O เปลี่ยนแปลงจาก $-V_{sat}$ ไปสู่ $+V_{sat}$ เมื่อ E_i มีค่าน้อยกว่า V_{LT} ค่าความแตกต่างศักดาระหว่าง V_{UT} กับ V_{LT} จะเรียกว่าแรงดันไฟฟ้าฮิสทีรีซิส V_H

ถ้ากำหนดค่าแรงดันไฟฟ้าฮิสทีรีซิสมากกว่า ค่า peak-to-peak noise voltage ก็จะไม่เกิดค่าผิดพลาดของเอาต์พุตครอสซิง (output crossing) ดังนั้นค่า V_H จึงบอกเราได้ว่า วงจรจะสามารถทนค่า peak-to-peak noise voltage ได้เท่าไร

2.2 วงจรปรับระดับสัญญาณแบบใช้ไดโอด (diode)

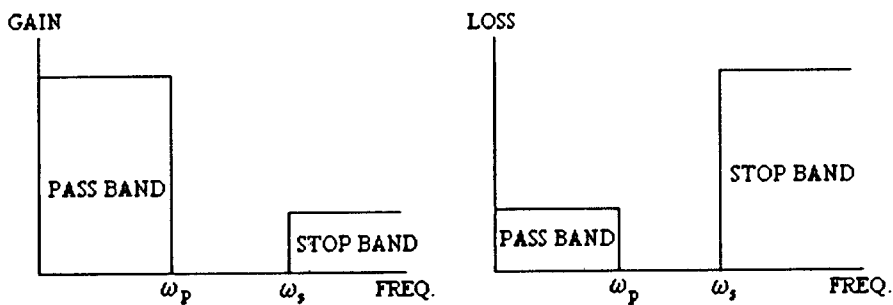


รูปที่ 2.7 วงจรปรับระดับสัญญาณและกราฟสัญญาณ

วงจรปรับระดับสัญญาณนั้นได้แสดงไว้ดังรูปที่ 2.7 วงจรนี้เป็นวงจรระดับสัญญาณให้ต่ำลง พิจารณาเมื่อสัญญาณเข้า (V_{in}) ผ่านเข้ามาในวงจรนี้ ในช่วงของสัญญาณเข้า ตัวเก็บประจุจะเริ่มอัดประจุจนกระทั่งมีค่าเท่ากับค่าแรงดันไฟฟ้าสูงสุดของสัญญาณเข้าภายในช่วงเวลานี้ แรงดันไฟฟ้าที่จุด A จะมีค่ามากกว่าแรงดันไฟฟ้าที่จุด B นั่นทำให้ไดโอด D_1 ได้รับไบอัสตรง (forward bias) ดังนั้นสัญญาณออกในช่วงนี้จึงมีค่าเป็น 0 โวลต์ เมื่อแรงดันไฟฟ้าของสัญญาณเข้าเปลี่ยนไปเป็นช่วงลบ คือแรงดันไฟฟ้าที่จุด B มีค่ามากกว่าที่จุด A คราวนี้ไดโอด D_1 ก็จะได้รับไบอัสย้อนกลับ (reverse bias) จึงทำให้ไดโอด D_1 มีค่าความต้านทานสูงมากเป็นอนันต์ เมื่อเป็นดังนั้นตัวเก็บประจุจึงคายประจุผ่านตัวความต้านทาน R การคายประจุของตัวเก็บประจุนั้น แรงดันไฟฟ้าของมันจะรวมกับศักดาช่วงลบของสัญญาณเข้าเพราะในช่วงนี้แรงดันไฟฟ้าที่ตกคร่อมตัวเก็บประจุจะมีทิศทางเสริมกับสัญญาณเข้าทำให้เราได้รับสัญญาณออกที่มีขนาดเท่ากับสัญญาณเข้า แต่มีระดับเฉลี่ยต่ำกว่าดังรูป

2.3 วงจรผ่านความถี่ต่ำ

คุณสมบัติพื้นฐานของวงจรผ่านความถี่ต่ำ คือผ่านความถี่ต่ำและบั่นทอนความถี่สูงดังรูปที่ 2.8 วงจรผ่านความถี่ต่ำจะต้องผ่านสัญญาณจาก DC จนถึงความถี่คัทออฟ (cut-off frequency) ω_p ซึ่งในช่วงของพาสแบนด์ (pass band) จะต้องสูญเสีย (loss) ไม่เกิน A_{max} dB ความถี่ที่สูงกว่า ω_s จนถึงอนันต์ (infinity) จะต้องสูญเสียอย่างต่ำ A_{min} dB ความถี่จาก $\omega_s - \infty$ เรียกว่าสตอปแบนด์ (stop band)

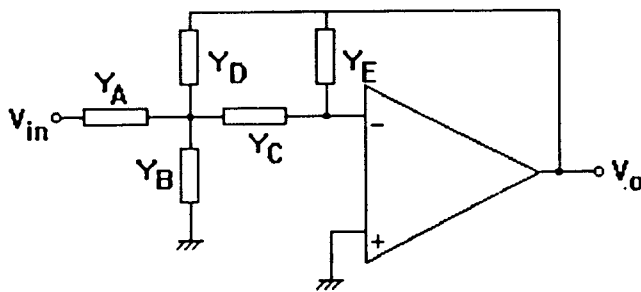


รูปที่ 2.8 คุณสมบัติของ LPF

Second order LPF จะมีรูปแบบสมการดังนี้

$$LPF (GAIN) = \frac{K \omega_p^2}{S^2 + \frac{\omega_p}{Q_p} S + \omega_p^2} ; GAIN = K \dots (2.3)$$

Negative feedback topology



รูปที่ 2.9 วงจร generalized NFB filter

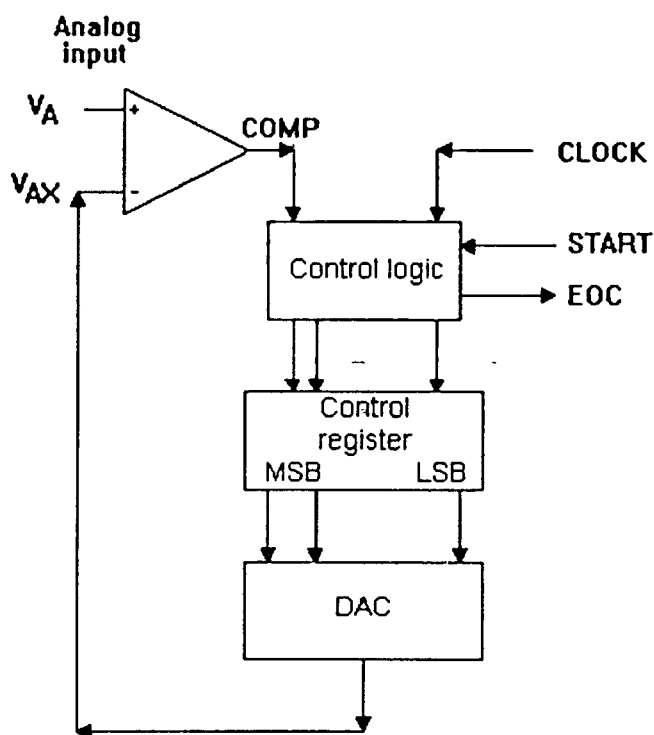
จากวงจร generalized NFB ในรูปที่ 2.9 โดยการใช้ node analysis ได้

$$T_v = \frac{V_o}{V_{in}} = \frac{-Y_A Y_C}{Y_E (Y_A + Y_B + Y_C + Y_D) + Y_C Y_D} \dots (2.4)$$

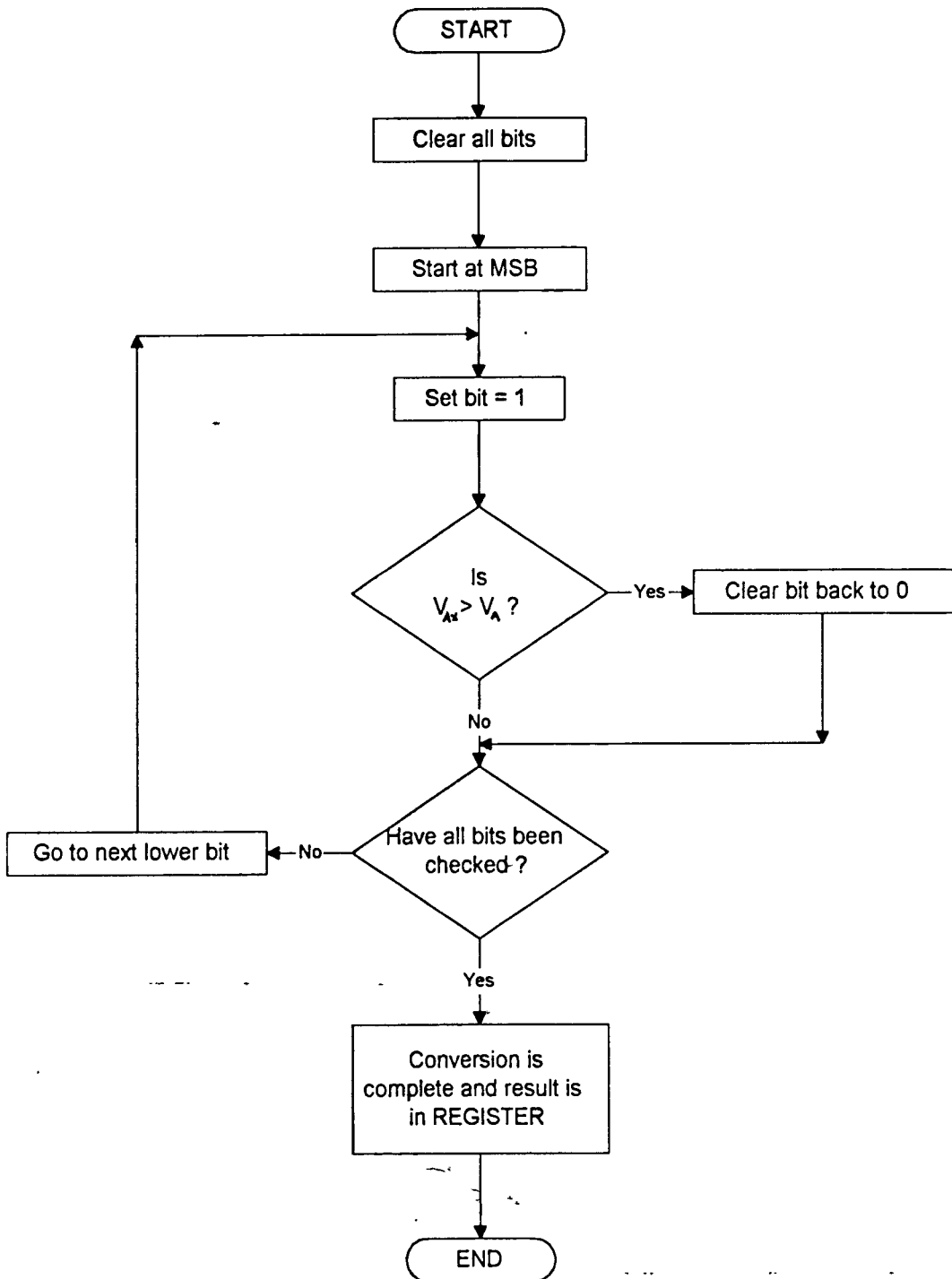
2.4 SUCCESSIVE APPROXIMATION ADC

successive approximation ADC เป็นวิธีการหนึ่งที่ใช้กันอย่างกว้างขวางในการแปลงสัญญาณอนาลอกไปเป็นสัญญาณดิจิทัล ซึ่งเป็นวงจรที่มีความซับซ้อนมากกว่าวิธีดิจิทัลแรมป์ (digital ramp) ADC แต่เวลาในการแปลง (conversion) สั้นกว่า ในที่นี้ SAC จะทำการตั้งค่าเวลาในการแปลง ซึ่งค่าเวลาในการแปลงจะไม่ขึ้นอยู่กับค่าอินพุตอนาลอก รูปที่แสดงใน 2.10(a) เป็นวิธีการซึ่งจะคล้ายๆกับการแปลงดิจิทัลแรมป์ ADC อย่างไรก็ตาม SAC ไม่จำเป็นต้องมีตัวนับที่ต้องใส่ให้ตรงส่วนของ DAC แต่ใช้รีจิสเตอร์ (register) แทน สัญญาณควบคุมจะถูกแต่งไว้ควบคุมบิตรีจิสเตอร์ (register bit) ต่อบิต (bit) ตั้งแต่รีจิสเตอร์ข้อมูลเป็นดิจิทัลที่เท่ากับสัญญาณอนาลอก V_A ซึ่งจะเป็นคำตอบของการแปลงการทำงานขั้นพื้นฐานจะแสดงโดยโฟลชาร์ต (flowchart) ในรูป 2.10(b) ตามโฟลชาร์ตนี้จะได้ตามตัวอย่างในรูป 2.11

ตามตัวอย่างเราจะเลือกตัวอย่างง่ายๆเพียงการแปลงแบบ 4 บิต แต่ละขั้นจะห่างกัน 1 โวลต์ ถึงแม้ว่าในความเป็นจริง SACs จะมีบิตมากกว่านี้และการแก้ปัญหาจะเล็กกว่านี้ดังตัวอย่าง แต่การทำงานจะเหมือนกันทุกประการ ในจุดที่ควรจะอธิบายว่า 4 บิต ที่แปลงเป็น DAC นั้นมีค่าเป็น 8.4.2 และ 1 โวลต์ ตามลำดับ



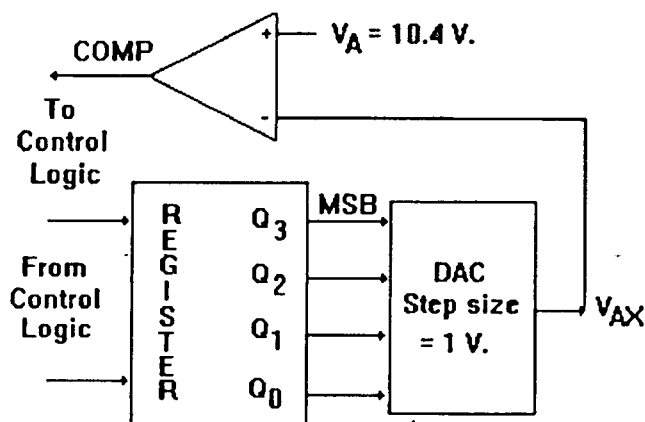
(a)



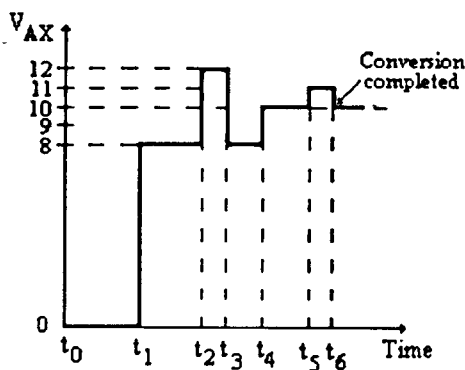
รูปที่ 2.10 (a) successive approximation (b) แผนภาพแสดงการทำงาน

ในที่นี้ให้สัญญาณอินพุตเป็น $V_A = 10.4$ โวลต์ การทำงานจะเริ่มตั้งแต่เคลียร์คอนโทรลลอจิก (control logic) ทุกบิต ให้เป็น 0 ดังนั้น $Q_3=Q_2=Q_1=0$ หรือ $[Q] = 0000$ ซึ่งค่านี้จะทำให้ DAC เอาท์พุท $V_{AX} = 0$ โวลต์ ที่เวลา t_0 ดังรูปที่ 2.11(b) ซึ่ง $V_{AX} < V_A$ และเอาท์พุทของคอมพาราเตอร์ (comparator) จะแสดงสถานะสูง (high)

ในขั้นตอนต่อมา (t_1) คอนโทรลลอจิกจะทำการเซต (set) ค่าบิตนัยสำคัญสูงสุด (most significant bit:MSB) ให้เป็น 1 ดังนั้น $[Q] = 1000$ จะทำให้ $V_{AX} < V_A$ เอาท์พุทที่คอมพาราเตอร์ จะยังคงสถานะสูง สถานะสูงนี้จะบอกคอนโทรลลอจิกว่าค่าที่ตั้งของ Q_3 ยังน้อยกว่า ดังนั้น ค่าบิตนัยสำคัญสูงสุด ต้องเป็นค่า 1



(a)



(b)

รูปที่ 2.11 แผนภาพแสดงการทำงานของ SAC 4 บิต โดยการใช้ DAC

ขั้นตอนต่อไปคอนโทรลลอจิกจะเลื่อนมายังบิตที่ต่ำกว่า คือ Q_2 ทำการตั้งค่า Q_2 ให้เป็น 1 ทำให้ $[Q] = 1100$ และ $V_{AX} = 12$ โวลต์ ที่เวลา t_2 พอ $V_{AX} > V_A$ เอาท์พุทของคอมพิวเตอร์จะเป็น สถานะต่ำ สถานะต่ำนี้จะบอกคอนโทรลลอจิกว่า V_{AX} มากเกินไป และคอนโทรลลอจิกจะทำการเคลียร์ (clear) Q_2 ให้เป็น 0 ที่เวลา t_3 ดังนั้นที่เวลาในรีจิสเตอร์จะกลับมาเป็น 1000 และค่าของ $V_{AX} = 8$ โวลต์

ขั้นตอนต่อไปเกิดขึ้นที่เวลา t_4 ซึ่งคอนโทรลลอจิกจะทำการตั้งค่าบิตที่ต่ำต่อไปคือ Q_2 ดังนั้น $[Q] = 1010$ และได้ $V_{AX} = 10$ โวลต์ ซึ่ง $V_{AX} < V_A$ เอาท์พุทที่คอนโทรลลอจิกจะแสดงสถานะสูงซึ่งบอก คอนโทรลลอจิกให้รักษา Q_1 ไว้ที่ค่า 1

ขั้นตอนสุดท้ายเกิดขึ้นที่เวลา t_5 ซึ่งคอนโทรลลอจิกจะทำการตั้งค่าบิตที่ต่ำต่อไปคือ Q_0 ดังนั้น $[Q] = 1011$ และได้ค่า $V_{AX} = 11$ โวลต์ ซึ่ง $V_{AX} > V_A$ เอาท์พุทที่คอนโทรลลอจิกจะแสดงสถานะต่ำ เป็นสัญญาณบอกว่ามากเกินไปและคอนโทรลลอจิกจะทำการเคลียร์ Q_0 กลับเป็น 0 ที่เวลา t_6

ถึงจุดนี้ทุกบิตในรีจิสเตอร์ได้ทำหมดแล้ว แสดงว่าการแปลงได้เสร็จสิ้นสมบูรณ์แล้ว และคอนโทรลลอจิกจะส่งสัญญาณ EOC ออกไป เพื่อบอกว่าค่าดิจิตอลที่เท่ากับค่า V_A ได้ถูกอยู่ในรีจิสเตอร์แล้ว ดังตัวอย่างค่าดิจิตอลที่ออกมาสำหรับ $V_A = 10.4$ โวลต์ คือ $[Q] = 1010$ สังเกตได้ว่าค่า 1010 จะเท่ากับค่าจริงๆคือ 10 โวลต์ ซึ่งน้อยกว่าค่าอนาลอกที่เข้ามา และค่านี้คือลักษณะจำเพาะอย่างหนึ่งของวิธี SACs

เวลาในการแปลงสัญญาณ (conversion Time) ในการทำงานจะอธิบายได้ดังนี้คอนโทรลลอจิกจะไปแต่ละบิตเพื่อตั้งค่าให้เป็น 1 แล้วตัดสินใจว่าจะให้รักษาค่า 1 นี้หรือไม่ และจะไปยังบิตต่อไป ขั้นตอนในแต่ละบิตจะให้เป็นหนึ่งรอบการทำงาน ดังนั้นเวลาในการแปลงสัญญาณทั้งหมดของ N-bit SACs จะเป็น N รอบการทำงาน จะได้

$$t_c \text{ ของ SACs} = N\text{-บิต} \times 1 \text{ วัฏจักรนาฬิกา (clock cycle)}$$

เวลาในการแปลงสัญญาณนี้ควรจะมีค่าเดียวกันกับค่าของ V_A ที่เป็นเช่นนั้นก็เพราะว่าคอนโทรลลอจิกมีขั้นตอนในการดูแต่ละบิตว่าแผนภาพแสดงการทำงานของโปรแกรม

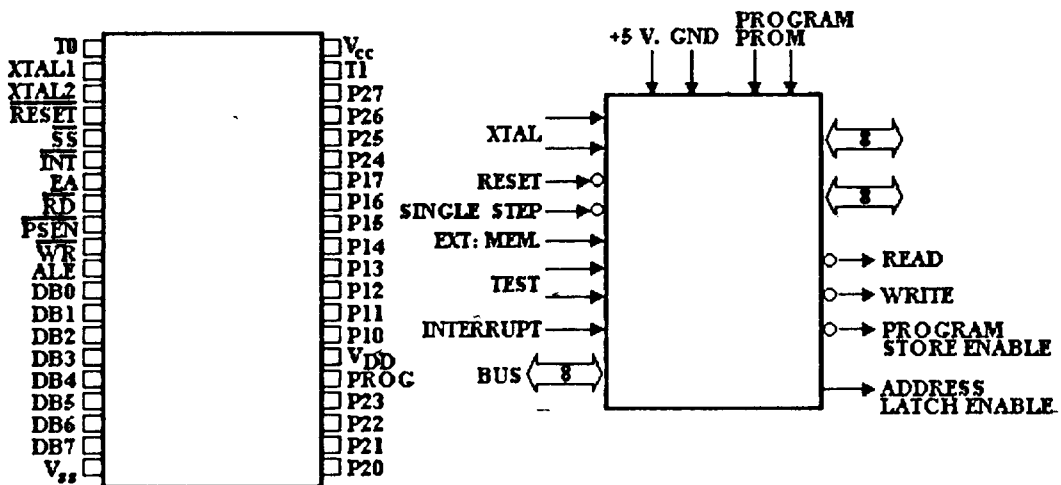
2.5 โครงสร้างสถาปัตยกรรมของ MCS-48

ลักษณะโครงสร้างของ MCS-48 แบ่งเป็น 2 ลักษณะ เช่นเดียวกับไมโครโปรเซสเซอร์ โดย (microprocessor) ทั่วไปคือ

1. โครงสร้างภายนอก
2. โครงสร้างภายใน

2.5.1 โครงสร้างภายนอก

ดูจากรายละเอียดของขาต่างๆใน MCS-48 ขาทั้งหมดของ MCS-48 ยกเว้นขาไฟเลี้ยงและขาสัญญาณนาฬิกาแล้ว ทุกขาทำหน้าที่เป็นสัญญาณอินพุตและเอาต์พุต โดยมีรายละเอียดของขาต่างๆ ดังนี้

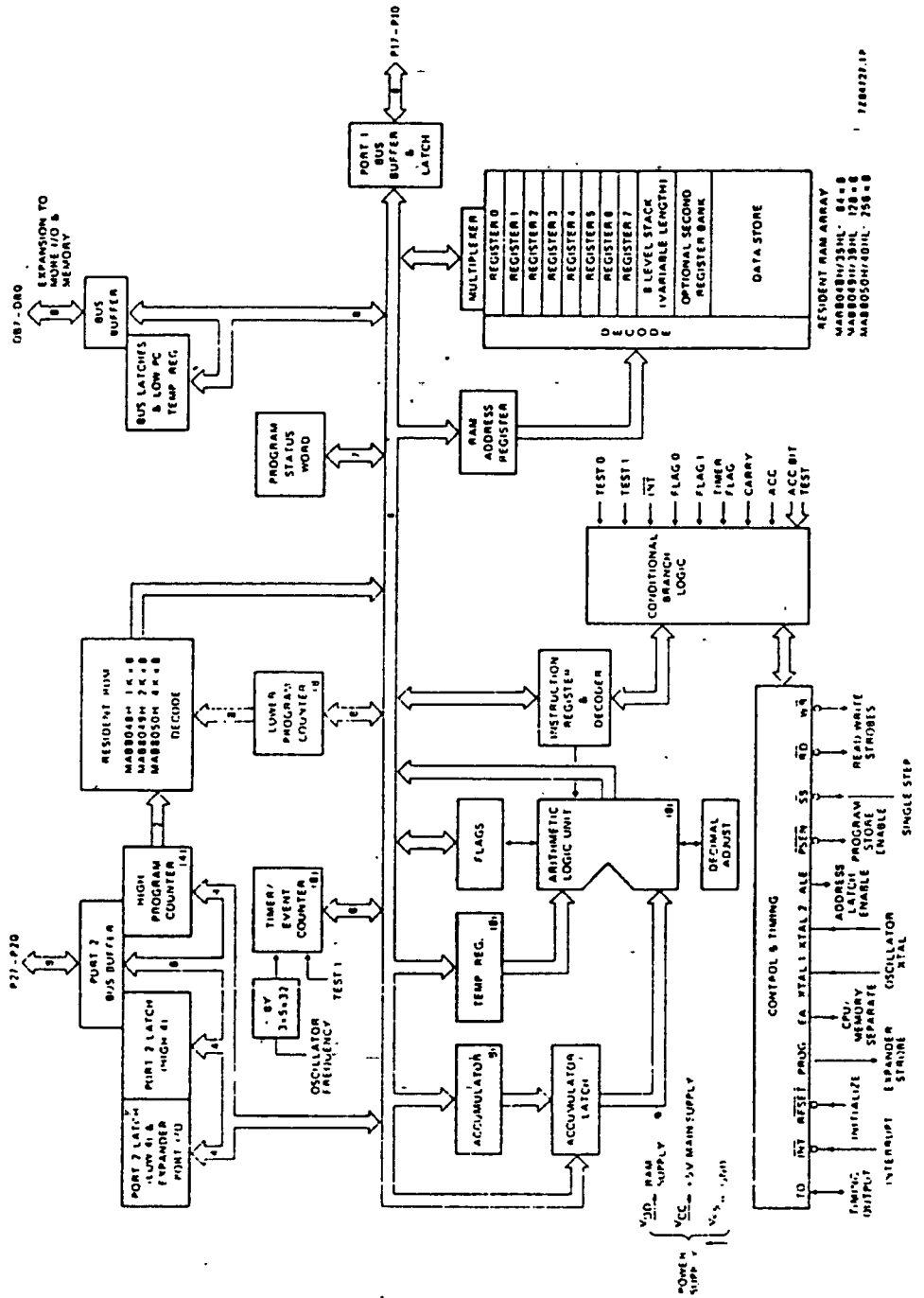


รูปที่ 2.12 ลักษณะการจัดขาภายนอกของ MCS-48

ขา 1	To	สามารถทดสอบสัญญาณเข้า เพื่อใช้ในการถ่ายเทข้อมูลต่างๆ ตามข้อกำหนดที่ตั้งไว้ โดยการใช้คำสั่ง JTo และ JNTo และสามารถที่จะใช้ To เป็นสัญญาณนาฬิกาส่งออกได้ด้วยคำสั่ง ENTto CLK To ยังถูกใช้เป็นขาควบคุมโหมด (mode) การโปรแกรมและซิงค์ (sync) ได้ด้วย
ขา 2	XTAL 1	สำหรับต่อขาข้างหนึ่งของคริสตอล (crystal) ภายนอก เพื่อใช้เป็นออสซิลเลเตอร์ (oscillator) ภายใน หรือแหล่งกำเนิดสัญญาณจากภายนอกต่อเข้านี้
ขา 3	XTAL 2	สำหรับต่อกับขาอีกข้างหนึ่งของคริสตอล
ขา 4	RESET	เป็นอินพุตสำหรับควบคุมให้ไมโครโปรเซสเซอร์เริ่มทำงานและยังเป็นสัญญาณสำหรับแลทช์ (latch) ตำแหน่งของหน่วยความจำโปรแกรมภายในเพื่อประโยชน์ในการอ่านรหัสข้อมูลโปรแกรม
ขา 5	SS	ขาซิงเกิลสเตป (single step) จะใช้ร่วมกับ ALE เพื่อทดสอบการทำงานครั้งละหนึ่งคำสั่ง
ขา 6	INT	เป็นขาอินพุตสำหรับสัญญาณอินเทอร์รัพท์ (interrupt) ภายนอก ถ้าต้องจะให้อินเทอร์รัพท์ได้ จะต้องสั่งซอร์ฟแวร์ (software) ด้วยการติดตั้งคำสั่ง EN I สัญญาณอิน-เทอร์รัพท์ ที่เข้ามาจะต้องอยู่ในสถานะต่ำอย่างน้อย 3 วัฏจักรแมชีน (machine cycle) เพื่อให้มั่นใจว่าตัวไมโครโปรเซสเซอร์จะรับหรือแซมปลิงอินเทอร์รัพท์ (sampling interrupt) ได้ ปกติหลักจากรีเซท (reset) หรือ เปิดเครื่องทุกครั้งจะยังไม่สามารถใช้งานด้านอิน-เทอร์รัพท์ได้
ขา 7	EA	เมื่อเป็นสถานะตรรกสูง จะเป็นการติดต่อรับรหัสข้อมูลจากภายนอกของซิงเกิลสเตปตัวนี้ และควบคุมให้ตัวนับโปรแกรม (program counter) เฟทช์ (fetch) รหัสจากโปรแกรมภายนอกทั้งหมด เพื่อใช้ประโยชน์ในการแก้ไขและทำอิมูเลท (emulate) หรือทดสอบกับเครื่องต้นแบบ

ขา 8	RD	จะเป็นสัญญาณควบคุมการรับ หรือ อ่านข้อมูลจากภายนอก ผ่านเข้าขาข้อมูลระหว่างการอ่าน และสัญญาณนี้จะสไตรบ (strobe) พัลส์ (pulse) ต่ำทุกครั้งที่มีการอ่านข้อมูลไม่ว่าจะอ่านรหัสจากภายนอกหรือภายใน
ขา 9	PSEN	เป็นสัญญาณ program store enable สัญญาณควบคุมการอ่านรหัสคำสั่งทุกคำสั่งจากหน่วยความจำภายนอก และสัญญาณนี้จะเกิดแอกทีฟต่ำให้มีการเฟลทซ์ข้อมูลโปรแกรมจากภายนอกได้เท่านั้น
ขา 10	WR	เป็นสัญญาณควบคุมการส่ง หรือเขียนข้อมูลออกภายนอกผ่านสายส่งข้อมูล (data bus) และ สัญญาณนี้จะสไตรบพัลส์ต่ำออกทุกครั้งที่มีการเขียนข้อมูล
ขา 11	ALE	เป็นสัญญาณ address latch enable สัญญาณนี้จะเกิดทุกวัฏจักรเมชั่นของการเฟลทซ์ในแต่ละครั้ง และใช้ประโยชน์ในการส่งสัญญาณนาฬิกาเอทท์พุท ด้วยการใช้สัญญาณขอบขาลงของสัญญาณนี้ เป็นการสไตรบแอดเดรสเพื่อถอดรหัส ข้อมูลจากหน่วยความจำข้อมูล และโปรแกรม
ขา 12-19	Do-D7	เป็นไบไดเรกชันนัลพอร์ท (bidirectional port) ทำหน้าที่ต่างๆ ดังนี้ <ol style="list-style-type: none"> 1. เป็นที่ส่งผ่านข้อมูลเข้าและออก ในขณะที่ซิงค์ด้วยสัญญาณ RD หรือ WR สไตรบ และสามารถทำหน้าที่เป็นพอร์ทสำหรับแลทซ์ข้อมูล 2. ใช้เป็นตัวส่งข้อมูลตัวนับโปรแกรม (program counter) บิตอันดับต่ำ ระหว่างการเฟลทซ์โปรแกรมจากภายนอก และรับรหัสคำสั่งภายใต้การส่งสัญญาณควบคุมของ PSEN 3. ใช้เป็นตัวส่งข้อมูลทั้งแอดเดรสและข้อมูลในการทำงานกับแรม (RAM) ภายนอกด้วยคำสั่ง MOVX (STORE) โดยใช้ขาควบคุม ALE, RD และ WR
ขา 20	V _{SS}	เป็นกราวด์ของวงจร

ขา 21-24	P20-P23	4 บิตแรกของควอซีไบไดเรกชันัลพอร์ต 2 (quasi-bidirectional port 2) ใช้เป็นตัวส่งค่า
ขา 25	PROG	ใช้เป็นสัญญาณเอาต์พุตสไตรบ สำหรับควบคุมการใช้ร่วมกับการขยายอินพุท/เอาต์พุทพอร์ทของ 8423 และใช้เป็นทีป้อนโปรแกรมพัลส์ขนาด +18 โวลต์ เข้าขานี้สำหรับการเขียนโปรแกรมเข้าตัวไมโครคอนโทรลเลอร์เบอร์ 8748/8749
ขา 26	V _{DD}	ปกติตัว MCS-48 นี้จะทำงานได้ด้วยการจ่ายแรงดัน 5 โวลต์จ่ายเข้าที่ขานี้และถ้าแหล่งจ่ายไฟตกจะทำให้ MCS-48 เข้าโหมดการใช้แหล่งจ่ายไฟสำรองเข้าที่ขานี้และการโปรแกรมรหัสเข้าตัวไมโครคอนโทรลเลอร์ เบอร์ 8748/8749 จะใช้การจ่ายแรงดันไฟ 21 โวลต์เข้าที่ขานี้เช่นกัน
ขา 27-34	P10-P17	เป็นพอร์ตขนาด 8 บิตแบบควอซีไบไดเรกชันัลพอร์ต 2 โดยมีตัวต้านทาน 50 โอห์มพูลล์อัฟ (pull-up) อยู่ในและขาเอาต์พุทจะเป็นแบบ 3 สถานะ
ขา 35-38	P24-P27	อันดับสูงของแอดเดรสตัวนับโปรแกรม และใช้ในการติดต่อกับหน่วยจ่ายนอก และใช้เป็น 4 บิต I/O bus สำหรับการขยาย port I/O ของ 8243 และอีก 4 บิตสูงใช้เป็น port I/O ธรรมดา
ขา 39	T1	เป็นสัญญาณที่ถูกทดสอบเข้าในตัวไมโครคอนโทรลเลอร์ ด้วยคำสั่ง JT1, JNT1 สามารถใช้ขา T1 ให้เป็นตัวนับเหตุการณ์ที่เกิดขึ้นจากภายนอกส่งเข้ามาด้วยการใช้คำสั่ง STRT CNT (START COUNTER)
ขา 40	V _{CC}	จ่ายแรงดันไฟ 5 โวลต์เข้าที่ขานี้ ในขณะที่ทำงานปกติและขณะที่อัดโปรแกรมรหัสเข้าตัว 8748/8749



รูปที่ 2.13 โครงสร้างสถาปัตยกรรมภายในของ MCS-48

2.5.2 โครงสร้างภายใน

โครงสร้างภายในของ MCS-48 เหมือนกับโครงสร้างของไมโครคอมพิวเตอร์ (microcomputer) ทั่วไป จะประกอบด้วยบัสที่ทำหน้าที่เชื่อมต่อกับอุปกรณ์อื่นๆ โดยมีส่วนประกอบภายในต่างๆ ดังรูปที่ 2.13 และรายละเอียดของแต่ละส่วนจะเป็นดังนี้ หน่วยประมวลผลทางคณิตศาสตร์ (ALU) หน่วยประมวลผลทางคณิตศาสตร์จะรับหรือทำงานร่วมกับข้อมูลจากแหล่งข้อมูล 1 กับ 2 แหล่ง เพื่อไปทำการประมวลผลทางคณิตศาสตร์ภายใต้การควบคุมของตัวถอดรหัสคำสั่งตัว หน่วยประมวลผลทางคณิตศาสตร์มีความสามารถที่จะทำงานตามหน้าที่ต่างๆ ดังต่อไปนี้

1. บวกพร้อมตัวทดหรือปราศจากตัวทด (add with or without carry)
2. ทำงานทางตรรก (AND, OR EXCLUSIVE OR)
3. การเพิ่มหรือลดค่าหนึ่งค่า (increment, decrement)
4. การแปลงกลับค่าบิต (bit complement)
5. วนบิตทางซ้ายหรือขวา (left or right rotate)
6. การสลับค่านิบเบิล (swap nibble)
7. การปรับค่าเป็น BCD decimal (BCD decimal adjust)

ถ้าการทำงานของหน่วยประมวลผลทางคณิตศาสตร์ให้ผลลัพธ์มากกว่า 8 บิตจะเกิดตัวทดจากบิตหลักสูงสุด (most significant bit, MSB) เข้าตัวแฟลกทด (carry flag) ภายในรีจิสเตอร์ PSW (program status word) จะถูกเซต (set)

แอกคิวมูเลเตอร์ (ACCUMULATOR)

จะเป็นตัวรีจิสเตอร์ที่สำคัญที่สุดของ MCS-48 ในตัวไมโครโปรเซสเซอร์ จะทำหน้าที่เป็นตัวรับข้อมูลจากแหล่งข้อมูลที่ส่งเข้ามา และเป็นตัวส่งผลลัพธ์ให้กับรีจิสเตอร์ตัวรับ (destination register) ในการทำงานของ ALU การติดต่อกับอุปกรณ์ภายนอกและติดต่อกับหน่วยความจำจะต้องผ่านแอกคิวมูเลเตอร์เสมอ

แฟลกตัวทด

เป็นตัวบอกสถานะการทำงานของ MCS-48 ว่าการทำงานเกินจำนวนบิตที่มีอยู่หรือไม่ ถ้าเกินแฟลกตัวทุดนี้จะถูกเซต แฟลกตัวนี้จะเป็นบิตหนึ่งของรีจิสเตอร์ PSW

ตัวถอดรหัส (INSTRUCTION REGISTER AND DECODER)

รหัสการทำงาน (operation code = op code) ของแต่ละคำสั่ง จะถูกแพทช์เข้ามา เก็บอยู่ในรีจิสเตอร์ถอดรหัสตัวนี้ จากนั้นจะถูกถอดรหัสเป็นสัญญาณควบคุมต่างๆ ตามรหัส ของคำสั่งนั้นๆ ต่อไป

2.5.3 หน่วยความจำ

หน่วยความจำของ MCS-48 นั้นแบ่งการทำงานออกได้เป็น 2 ชนิด คือ หน่วยความจำ โปรแกรม และ หน่วยความจำข้อมูล

หน่วยความจำโปรแกรม หน่วยความจำนี้จะใช้สำหรับเก็บชุดคำสั่งการทำงานของ MCS-48 ที่เขียนขึ้น ขนาดหน่วยความจำสูงสุดของส่วนนี้คือ 4 กิโลไบต์ (Kbyte) โดยส่วน หนึ่งจะอยู่ในชิพหรือเป็นการต่อเพิ่มจากภายนอกก็ได้ ในกรณีที่หน่วยความจำโปรแกรม ภายในชิพจะมีขนาดความจุและชนิดต่างๆ กัน แล้วแต่เบอร์ในตระกูล MCS-48 ดังตารางที่ 2.1 ตำแหน่งที่สำคัญของหน่วยความจำส่วนนี้คือ

(แอดเดรส 000) เมื่อถูกรีเซ็ต MCS-48 จะทำการเพทช์คำสั่งตัวแรกที่ตำแหน่งนี้

(แอดเดรส 003) MCS-48 จะเพทช์คำสั่งแรกที่ตำแหน่งนี้เมื่อมีสัญญาณอินเทอร์รัพท์ เข้ามา ถ้าอินเทอร์รัพท์ถูกติดตั้งให้เอ็นนาเบิล (enable) จะเป็นเหตุให้โปรแกรมกระโดดไปทำ งานโปรแกรมบริการอินเทอร์รัพท์ที่ตำแหน่งนี้

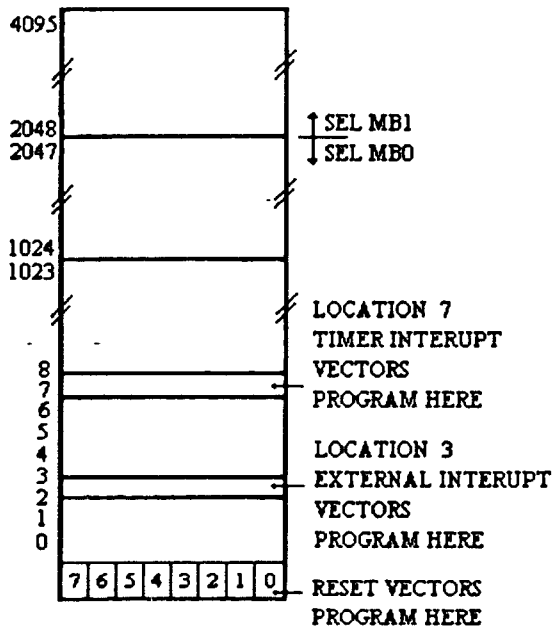
(address 007) MCS-48 จะ fetch คำสั่งแรกที่ตำแหน่งนี้ เมื่อมีการอินเทอร์รัพท์ ทำงานแบบจับเวลา/ตัวนับ มีผลจากการที่ตัวจับเวลา/ตัวนับเวลาเกิดโอเวอร์โฟลล์ (overflow)

ตำแหน่งแรกของทั้ง 3 ตำแหน่งดังกล่าว โดยปกติแล้วจะเขียนด้วยคำสั่ง JUMP ไปยังโปรแกรมบริการต่างๆ ที่เขียนไว้ ดังนั้นคำสั่งกระโดดตัวแรกจะถูกใช้งานหลังจากที่ โปรแกรมเริ่มทำงาน (initialize) ถูกเก็บเข้าที่แอดเดรส 000 ในการทำงานเริ่มต้น คำสั่งของการ กระโดดไปใช้บริการอินเทอร์รัพท์จากภายนอก เริ่มแรกจะถูกเก็บเข้าที่แอดเดรส 003 และคำ สั่งของการกระโดดไปใช้บริการ โปรแกรมการใช้อินเทอร์รัพท์ตัวจับเวลา/ตัวนับ เริ่มแรกจะถูก เพทช์จากแอดเดรส 007

เบอร์	หน่วยความจำภายใน		แรมสำรอง
	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

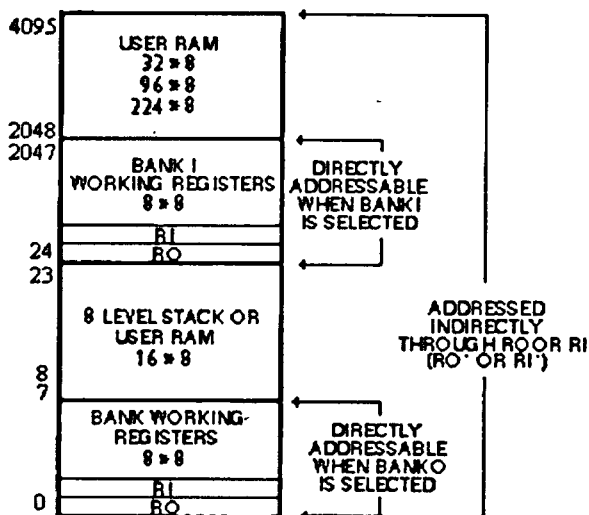
ตารางที่ 2.1 รายละเอียดต่างๆ ของตระกูล MCS-48

หน่วยความจำโปรแกรมสามารถใช้เป็นตัวเก็บข้อมูลคงที่ และโปรแกรมคำสั่ง ด้วยการใส่คำสั่ง MOVP และ MOVP3 ช่วยให้การอ่านตารางข้อมูลคงที่ทำได้ง่ายขึ้น



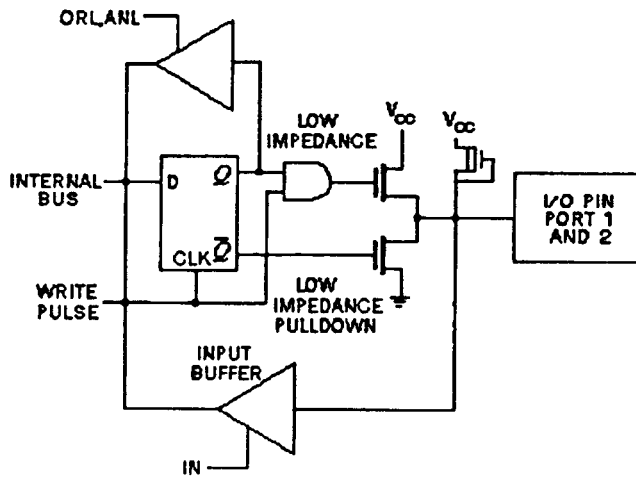
รูปที่ 2.14 PROGRAM MEMORY MAP

หน่วยความจำข้อมูล หน่วยความจำส่วนนี้จะป็นชนิดแรมที่อยู่ภายในชิพของ MCS-48 โดยมีขนาด 64.128 และ 256 ไบต์ ขึ้นอยู่กับเวอร์ชันของชิพตระกูล MCS-48 ดังตารางที่ 2.1 ภายในส่วนของหน่วยความจำส่วนนี้จะทำหน้าที่เป็นรีจิสเตอร์ใช้งาน 16 ตัว โดยแบ่งเป็น 2 ชุดๆ ละ 8 ตัว ซึ่งชุดแรกอยู่ที่แอดเดรส 0-7 เป็นแบงก์ 0 (banko) และชุดที่ 2 อยู่ที่แอดเดรส 24-31 เป็นแบงก์ 1 ในขณะที่ MCS-48 ทำงานอยู่นั้น รีจิสเตอร์จะถูกใช้งานเพียงชุดเดียว โดยสามารถจะเลือกชุดของรีจิสเตอร์จากคำสั่ง SEL RB (register bank switch) เป็นลักษณะที่สามารถจะเลือกเลขที่อยู่โดยตรง (direct address) ได้จากเลขที่อยู่ 0-7 ในแบงก์ 0 และ แบงก์ 1 อาจใช้งานป็นรีจิสเตอร์ขยายจากแบงก์แรกหรือใช้เป็นรีจิสเตอร์สำรองสำหรับการใช้งานระหว่างการใช้งานโปรแกรมย่อย และในโปรแกรมหลักยังคงใช้แบงก์แรกในการเก็บข้อมูลโดยทันทีด้วยแบงก์สวิตช์ (bank switch) ขณะเดียวกันถ้าแบงก์ 1 ไม่ถูกนำมาใช้แอดเดรส 24-31 ยังคงใช้เป็นแอดเดรสด้วยการเก็บข้อมูลแบบแรมได้



รูปที่ 2.15 DATA MEMORY MAP

นอกจากทำหน้าที่เป็นรีจิสเตอร์แล้ว ที่หน่วยความจำข้อมูลแอดเดรส 8-23 ก็จะเป็นเนื้อที่สแตก (stack) ด้วยการเก็บตัวนับโปรแกรมเป็นคู่ โดยใช้จำนวนครั้งละ 2 ตำแหน่ง หน่วยความจำตำแหน่งเหล่านี้จะถูกแอดเดรสด้วยตัวชี้แรมของ Ro กับ R1 การใช้งานพื้นที่นี้เหมาะสำหรับการใช้โปรแกรมย่อยหลายโปรแกรมเชื่อมโยงกันแต่ต้องมากกว่า 8 โปรแกรมย่อย เพราะเก็บได้เพียง 8 คู่ ทำนองเดียวกัน พื้นที่ของสแตก นี้ ถ้าไม่ถูกใช้เป็นสแตกรีจิสเตอร์ (stack register) ก็สามารถที่จะใช้เป็นที่เก็บข้อมูลแรมทั่วไปได้ แต่ถ้าใช้เป็นสแตกสำหรับโปรแกรมย่อยแล้ว ก็ไม่สามารถที่จะใช้เป็นที่เก็บข้อมูลแบบแรมในเวลาเดียวกันได้



รูปที่ 2.16 โครงสร้างของพอร์ทควอซีไบโตรีเกชันนัล

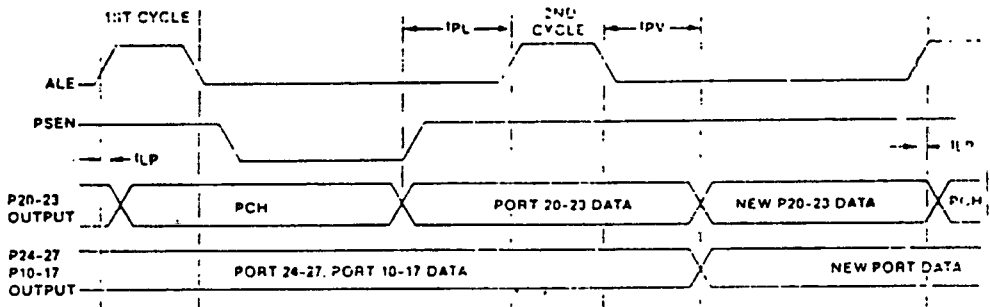
ในกรณีที่ผู้ใช้ต้องการใช้เนื้อที่แรมมากกว่าที่มีอยู่ในตัว MCS-48 ก็สามารถเพิ่มแรมภายนอกได้ โดยใช้คำสั่ง MOVX @R,A เมื่อใช้ R เป็นตัวรีจิสเตอร์ 0 หรือรีจิสเตอร์ 1 ทำหน้าที่เป็นตัวชี้ในการติดต่อกับแรมที่เพิ่มขึ้นภายใน 256 ไบต์ หน่วยความจำส่วนนี้โดยปกติจะใช้สำหรับเก็บข้อมูลชั่วคราว เมื่อมีการอ่านข้อมูลเข้าไปใน MCS-48 จะมองข้อมูลส่วนนี้ผิดกับข้อมูลที่อยู่ในส่วนของหน่วยความจำโปรแกรมที่กล่าวมาแล้ว ขณะเดียวกันการใช้รีจิสเตอร์ ตัวชี้ R0 และ R1 ของแรมสามารถที่จะเพิ่มรีจิสเตอร์ตัวชี้ได้อย่างง่ายดาย ด้วยคำสั่ง BANK SWITCH เมื่อต้องการใช้ตำแหน่งข้อมูลกับแรมถึง 4 ตำแหน่งในการเข้าถึงข้อมูลแต่ละครั้ง

2.5.4 ส่วนติดต่ออุปกรณ์ภายนอก

ในส่วนการติดต่ออุปกรณ์ภายนอกของ MCS-48 สามารถแบ่งออกได้ 2 ลักษณะคือ ทำหน้าที่เป็นพอร์ทกับบัส โดยจะแยกกล่าวดังรายละเอียดต่อไปนี้

พอร์ท ภายในตัว MCS-48 จะมีพอร์ทคล้ายกัน 2 พอร์ท คือ พอร์ท 1 (port1) และ พอร์ท 2 (port2) เป็นพอร์ทที่มีขนาด 8 บิต พอร์ททั้ง 2 ชนิดนี้เป็นพอร์ทชนิดควอซีไบโตรีเกชันนัลพอร์ท เนื่องจากโครงสร้างวงจรของพอร์ทแต่ละเส้นจะทำงานเป็นอินพุทและเอาต์พุท ด้วยลักษณะข้อมูลเอาต์พุทจะเป็นสแตติกแลทช์ (static latch) และข้อมูลจะยังคงอยู่จนกว่าจะมีการเขียนข้อมูลพอร์ทใหม่ จากรูปที่ 2.16 แสดงวงจรภายในของขาแต่ละขาของพอร์ททั้งสองจะเห็นได้ว่าตัวพูลอัพไฮอิมพีแดนซ์ (high impedance) ที่ต่อกับ V_{CC} จะหน้าที่ยกกระตบแรงดันไฟฟ้าและกระแสให้เพียงพอกับระดับลอจิก (logic) ของ TTL ที่จะมาต่อพ่วงจากวงจร

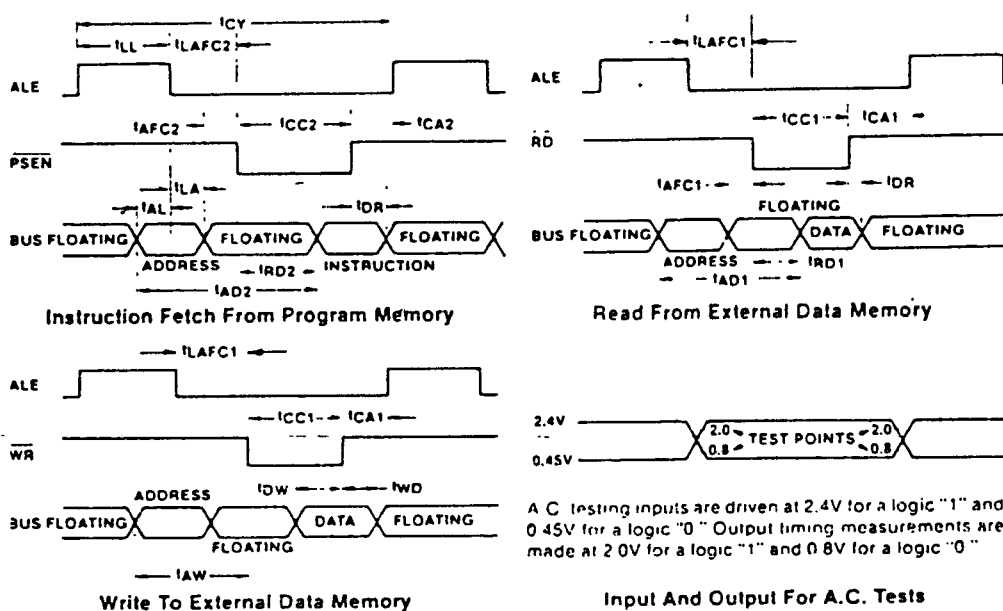
จะเห็นว่าในกรณีของการอินพุตนั้น สามารถจะรับการเปลี่ยนระดับลอจิกจากระดับสูงสู่ระดับต่ำเท่านั้น ดังนั้นในการใช้ขาใดขาหนึ่งของพอร์ตเป็นอินพุต สัญญาณนั้นต้องทำการเซทให้ขา นั้นอยู่ในระดับลอจิก 1 ก่อนเสมอ ในการรีเซท เริ่มแรกที่สำคัญในการอ่านหรือเขียนจาก พอร์ตจะใช้คำสั่ง ORL และ ANL เมื่อทำงานตัวไมโครโปรเซสเซอร์จะอ่านข้อมูลจาก พอร์ตด้วยคำสั่งอ่านข้อมูลและตามด้วยการเขียนข้อมูลกลับไปพอร์ต การเขียนข้อมูลมี ความจำเป็นต้องมีเพื่อให้ enable pull up low impedance ต่ำเกิดขึ้นชั่วขณะหนึ่ง แม้ข้อมูลจะเปลี่ยนแปลงจาก '1' ก็ตาม การกำหนดทำเช่นนี้ก็เพื่อที่จะกำหนด (config) ใช้พอร์ตเป็น ทั้งอินพุต และเอาท์พุทพอร์ตเดียวกันได้



รูปที่ 2.17 แผนภูมิจังหวะเวลาของการใช้พอร์ต 1 และ 2 เป็นพอร์ตเอาท์พุท

ในกรณีที่ใช้น้ำหนักความจำโปรแกรมภายนอก 4 บิตแรกของพอร์ต 2 จะทำหน้าที่ในการกำหนดเลขที่อยู่ 4 บิต อันดับสูงของหน่วยความจำโปรแกรมเพื่อทำการอ่านคำสั่งการทำงาน ในกรณีนี้ถ้าต้องใช้ 4 บิตแรกของพอร์ต 2 ทำหน้าที่สำหรับติดต่ออุปกรณ์ภายนอกอีก จะเกิดปัญหาที่ว่า MCS-48 ใช้พอร์ต 2 เป็นตัวซีแอดเดรส 4 บิตสุดท้ายอยู่แล้ว สามารถแก้ปัญหานี้ได้โดยการใช้การแลทซ์เข้าช่วย เพราะสัญญาณที่ออกมาที่ 4 บิตแรกของพอร์ต 2 จะเป็นลักษณะมัลติเพล็กซ์ (multiplex) ระหว่างข้อมูลที่ติดต่อกับอุปกรณ์ภายนอกกับเลขแอดเดรสที่ส่งออกมาโดยรูปแบบของการติดต่ออุปกรณ์ภายนอกจะออกมาในขณะช่วงขึ้นของสัญญาณ ALE แต่รูปแบบของแอดเดรสจะปรากฏเมื่อช่วงลงของสัญญาณ ALE ดังนั้นข้อมูลสำหรับติดต่ออุปกรณ์ภายนอกกับแอดเดรสจึงแยกออกมาได้ ดังรูปที่ 2.17 เป็นแผนภูมิจังหวะเวลาของการใช้พอร์ต 1 และพอร์ต 2 เป็นเอาท์พุทพอร์ตร่วมกับการส่งบิตแอดเดรสอันดับสูงของตัวนับโปรแกรม (PCH)

BUS ในส่วนนี้ พอร์ต 0 จะทำหน้าที่ 2 อย่างขึ้นอยู่กับกรณีที่ใช้คือในกรณีที่ใช้นหน่วยความจำโปรแกรมภายใน บัสจะทำหน้าที่เช่นเดียวกับพอร์ตทั่วไปที่เป็นพอร์ตชนิดควอซีไบไดเรกชันนัลพอร์ตซึ่งจะต่างกับพอร์ต 1 และพอร์ต 2 ที่กล่าวมาแล้ว โดยจะทำหน้าที่เป็นเอาต์พุทพอร์ตที่ latch ข้อมูลไว้ด้วยคำสั่ง OUTL และยังเป็นอินพุทพอร์ตที่ไปแลตช์ข้อมูลไว้ที่ พอร์ตด้วยคำสั่ง INS และทั้งสองคำสั่งนี้ จะสร้างพัลส์สไตรบที่ขา RD หรือ WR และที่ขอบจังหวะขอบขาลงของ RD หรือ WR จะเป็นช่วงจังหวะข้อมูลที่ปรากฏที่พอร์ตนี้ แต่พอร์ตนี้จะทำหน้าที่ผสมเป็นทั้งอินพุทพอร์ตและเอาต์พุทพอร์ตพร้อมกันไม่ได้ แต่ในกรณีที่ใช้นหน่วยความจำโปรแกรมภายนอกแล้วพอร์ตนี้จะทำหน้าที่เป็นบัสในการติดต่อกับหน่วยความจำโปรแกรมภายนอก ในขณะที่ทำการอ่านคำสั่ง จะให้แอดเดรสและอ่านคำสั่งเข้าที่บัสนี้ด้วยวิธี มัลติเพล็กซ์เช่นเดียวกัน



รูปที่ 2.18 แผนภูมิจังหวะเวลาของการเฟตช์โปรแกรมการอ่านและเขียนข้อมูลจากภายนอก

พอร์ท 2 ที่กล่าวมาแล้ว ในการมัลติเพล็กซ์นั้น MCS-48 จะส่งสัญญาณแอดเดรสออกมาที่พอร์ทนี้ก่อน ก็จะถูกแลตช์ไว้ที่อุปกรณ์การแลตช์แอดเดรสภายนอก แล้วจึงทำการอ่านคำสั่งที่ได้จากหน่วยความจำโปรแกรมภายนอก ในขณะที่ไม่มีการอ่านและเขียนบัสจะอยู่ในสถานะไฮอิมพีแดนซ์ดังรูปที่ 2.18 เป็นการแสดงช่วงจังหวะของการใช้บัสพอร์ท (bus port) ทำงานทั้งแบบการเพชชีโปรแกรม การอ่านและเขียนข้อมูลจากภายนอก ซึ่งจะต้องส่งค่าแอดเดรสออกมาก่อน แล้วจึงตามด้วยรหัสคำสั่งหรือข้อมูล โดยใช้ขา ALE, PSEN, WR และ RD เป็นตัวควบคุมบอกจังหวะของข้อมูลในช่วงจังหวะใดๆ ที่เกิดขึ้น

2.5.5 สัญญาณการตรวจสอบและสัญญาณอินเทอร์รัพท์

ในตระกูล MCS-48 ได้จัดขาไว้ 3 ขา สำหรับเป็นอินพุทและไว้ใช้ทดสอบเงื่อนไขเพื่อทำงานร่วมกับคำสั่ง JUMP ซึ่งได้แก่ To, T1 และ INT หลังจากทำการทดสอบเงื่อนไขของสัญญาณทั้งสามแล้ว การทำงานของ MCS-48 สามารถข้ามไปทำงานตามคำสั่งในโปรแกรมย่อยที่เขียนไว้เพื่อทำงานตามเงื่อนไขที่ถูกต้องโดยไม่จำเป็นต้องผ่านแอกคิวมูลเตอร์ นอกจากนี้แล้วสัญญาณทั้งสามนี้ยังสามารถทำหน้าที่อย่างอื่นได้อีกซึ่งจะอธิบายในหัวข้อต่อไป

2.5.6 ตัวนับโปรแกรมและสแตค

A ₁₁	A ₁₀	A ₉	A ₈	A ₇	A ₆	A ₅	A ₄	A ₃	A ₂	A ₁	A ₀
-----------------	-----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

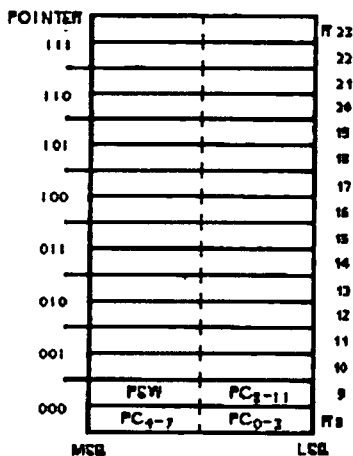
Conventional Program Counter

- Counts 000H to 7FFH
- Overflow 7FFH to 000H

รูปที่ 2.19 ตัวนับโปรแกรม

ตัวนับโปรแกรมเป็นตัวนับอิสระขนานแผนภาพแสดงการทำงานของโปรแกรมทำงานปกติ จะเพิ่มครั้งละหนึ่งเพื่อใช้ในการชี้แอดเดรสของหน่วยความจำโปรแกรมเพื่อเพชชีรหัสคำสั่งตามลำดับด้วย ขนาด 10, 11 และ 12 บิตของตัวนับโปรแกรมนี้จะใช้เป็นตัวชี้แอดเดรสขนาด 1024, 2048 และ 4096 ไบต์ของหน่วยความจำโปรแกรมบนชิพเบอร์ 8048AH, 8049AH และ 8050AH ตามลำดับ สำหรับบิต 11 ของตัวนับโปรแกรมจะไม่เปลี่ยนไปตามการเพิ่มของตัวนับโปรแกรมจะเปลี่ยนได้ด้วยคำสั่ง select program bank เท่านั้น เป็นการเลือกชุดของหน่วยความจำโปรแกรมส่วน 2 กิโลไบต์แรกหรือ 2 กิโลไบต์หลัง ในกรณีที่ใช้หน่วยความจำ

ภายนอก บิต 0-7 ของตัวนับโปรแกรมนี้จะส่งออกมาปรากฏที่บัส เมื่อสัญญาณ ALE เริ่มตกลง ส่วนบิต 8-11 ของตัวนับโปรแกรมจะถูกส่งออกจากทาง 4 บิตอันดับต่ำของพอร์ท 2 เพื่อใช้สำหรับเป็นตัวชี้แอดเดรสอันดับสูงของหน่วยความจำโปรแกรม



รูปที่ 2.20 บริเวณสแตคที่เก็บตัวนับโปรแกรม

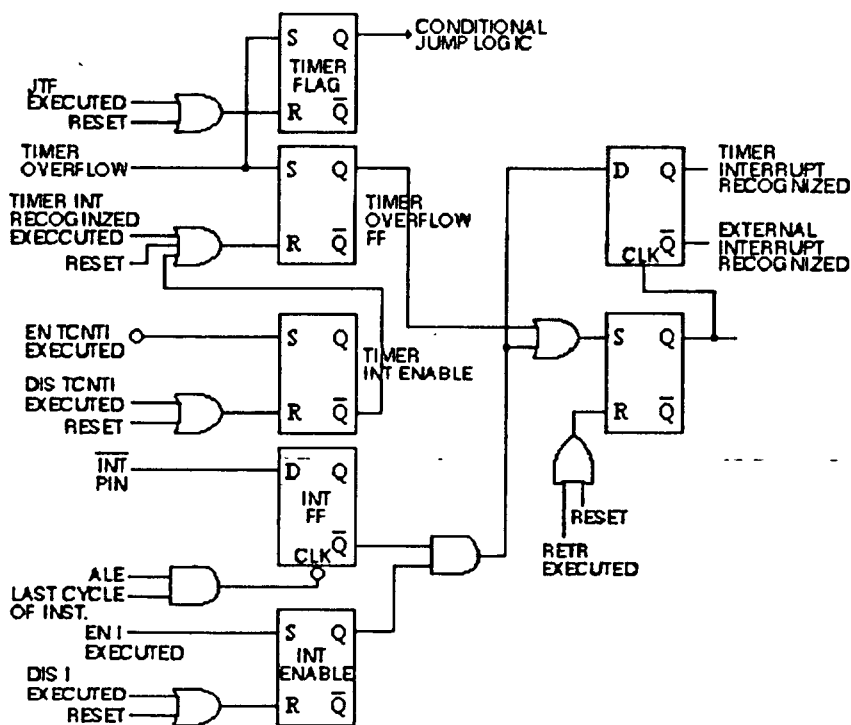
ในการอินเทอร์รัพท์ หรือใช้คำสั่ง CALL เข้าสู่โปรแกรมย่อย จะเป็นเหตุให้ค่าตัวนับโปรแกรมถูกย้ายไปเก็บไว้ที่บริเวณสแตครีจิสเตอร์ 8 คู่ เป็นโปรแกรมเคาท์เตอร์สแตค (program counter stack) ดังรูปที่ 2.20 ตัวชี้สแตคจะมีค่า 3 บิต แอดเดรสเป็นส่วนหนึ่งของ PSW จะเป็นตัวชี้สแตครีจิสเตอร์ 8 คู่ โดยที่ข้อมูลของแรมภายในตำแหน่งที่ 8-23 จะใช้เป็นสแตครีจิสเตอร์ 8 คู่สำหรับเก็บค่าตัวชี้โปรแกรม 12 บิต และค่านิบเบิล (nibble) สูงของรีจิสเตอร์ที่ทำหน้าที่เป็นค่าแสดงสถานะของโปรแกรม ตัวชี้สแตคเมื่อถูกติดตั้งเริ่มแรกจะชี้ที่ 000 ซึ่งครอบคลุมตำแหน่งแรมที่ R8 และ R9 เป็นคู่สแตครีจิสเตอร์ ดังนั้นเมื่อเกิดมีการอินเทอร์รัพท์หรือมีการเรียกโปรแกรมย่อย จะมีการเก็บข้อมูลของตัวชี้โปรแกรมไว้ที่ตำแหน่ง R8 และ R9 และตัวชี้สแตคจะเพิ่มขึ้นอีกหนึ่ง คือ 001 ชี้ที่คู่ตำแหน่ง R10 และ R11 ดังนั้นบริเวณสแตคสามารถใช้โครงข่ายการเรียกโปรแกรมย่อยซ้อนภายในโปรแกรมย่อยซึ่งจะเรียกได้ 8 ครั้ง โดยไม่เกิดการโอเวอร์โฟลของสแตคขึ้น ตัวชี้สแตคจะเกิดการวนกลับมาเริ่มที่ตำแหน่งตัวชี้สแตคตำแหน่งใหม่อีกครั้งหนึ่งและจะทับค่าเดิม เนื่องจากการเกิดโอเวอร์โฟลของ สแตคจากตำแหน่ง 111 มาหา 000 และจะเกิดอันเดอร์โฟล (underflow) จาก 000 ไป 111 จากการสิ้นสุดโปรแกรมย่อยด้วยการใช้คำสั่ง RET หรือ RETR จะเป็นเหตุให้ตัวชี้สแตกลดค่าลง และจะเป็นการย้ายข้อมูลในคู่สแตครีจิสเตอร์มาเข้าสู่ตัวชี้โปรแกรม

2.5.7 การอินเทอร์รัพท์

MCS-48 จะรับการอินเทอร์รัพท์ต่อเมื่อได้เซตอินาเบิล (set enable) แล้ว ลำดับการอินเทอร์รัพท์จะเกิดขึ้น เมื่อมีสัญญาณสถานะลอจิกเป็น 0 เข้าที่ขา INT โดยที่การอินเทอร์รัพท์จะกระตุ้นด้วยสัญญาณระดับต่ำและสามารถที่จะรับการอินเทอร์รัพท์เข้ามาได้หลายแหล่งด้วยกัน ดังแสดงการอินเทอร์รัพท์ทางลอจิกในรูปที่ 2.21 ขาอินเทอร์รัพท์จะถูกแซมปลิง (sampling) ทุกวัฏจักรคำสั่งในช่วงวัฏจักรการทำงานของ ALE และเมื่อตรวจพบ (ซึ่งสัญญาณอินเทอร์รัพท์จะต้องมีสถานะต่ำอย่างน้อย 3 วัฏจักรเมซินเพราะสัญญาณจะถูกตรวจที่วัฏจักรที่ 2 ของการทำงานของคำสั่ง) ก็จะถูกโปรแกรมย่อยการอินเทอร์รัพท์โดยก่อนเข้าสู่โปรแกรมย่อยบริการอินเทอร์รัพท์ตัวนับโปรแกรมและบิตที่ 4-7 ของค่าแสดงสถานะของโปรแกรมจะถูกเก็บไว้ที่สแตคและการควบคุมการทำงานต่างๆ เมื่อมีการอินเทอร์รัพท์นั้นจะถูกส่งผ่านไปทำงานที่ตำแหน่ง 003 ของหน่วยความจำโปรแกรม ซึ่งโดยปกติจะเขียนด้วยคำสั่งกระโดดแบบไม่มีเงื่อนไข (unconditional jump) ไปยังโปรแกรมบริการที่ใช้ตอบสนองการอินเทอร์รัพท์ การกลับคืนหลังจากบริการโปรแกรมย่อยของการอินเทอร์รัพท์จะกลับโดยใช้คำสั่ง RETR ซึ่งจะยังคงรักษาค่าแสดงสถานะของโปรแกรมไว้ การอินเทอร์รัพท์ของ MCS-48 เป็นการอินเทอร์รัพท์ระดับเดียว ดังนั้นระหว่างการบริการโปรแกรมอินเทอร์รัพท์อยู่ การอินเทอร์รัพท์จะไม่สามารถได้รับบริการ ดังนั้นจะต้องมีการอินาเบิลอินเทอร์รัพท์ทุกครั้งด้วยคำสั่ง RETR เสมอ ซึ่งจะอินาเบิลได้ หลังวัฏจักรเมซินที่ 2 ของการทำงานคำสั่ง RETR

การอินเทอร์รัพท์ที่เกิดจากตัวจับเวลา/ตัวนับที่เกิดจากการโอเวอร์โฟลภายในรีจิสเตอร์ของตัวจับเวลา/ตัวนับ ก็จะทำงานเช่นเดียวกับการอินเทอร์รัพท์ภายนอก จะต่างกันก็ตรงที่ว่า การอินเทอร์รัพท์ที่เกิดจากตัวจับเวลา/ตัวนับนั้น จะไปทำที่ตำแหน่ง 007 ซึ่งเป็นตำแหน่งที่โปรแกรมย่อยการอินเทอร์รัพท์ของตัวจับเวลา/ตัวนับอยู่ การอินเทอร์รัพท์จากภายนอกจะมีลำดับสูงกว่าการอินเทอร์รัพท์จากตัวจับเวลา/ตัวนับ หมายความว่าถ้ามีการอินเทอร์รัพท์จากภายนอกพร้อมกับการอินเทอร์รัพท์จากตัวจับเวลา/ตัวนับแล้ว MCS-48 จะตอบสนองต่อการอินเทอร์รัพท์จากภายนอกก่อนเสมอเมื่อต้องการดิสเอบิ้ล (disalbe) อินเทอร์รัพท์สามารถทำได้ 2 วิธีคือ การรีเซต MCS-48 กับการใช้คำสั่งดิสเอบิ้ล DIS I และ DIS TCNTI ทั้งการอินเทอร์รัพท์จากภายนอกและตัวจับเวลา/ตัวนับ

จังหวะเวลาอินเทอร์รัพท์ การอินเทอร์รัพท์จะอินนิตีลได้ด้วยโปรแกรมควบคุมด้วย การใช้คำสั่ง EN I และ DIS การอินเทอร์รัพท์จะดีสเอนิตีลได้ด้วยการรีเซทและจะยังคง ดีสเอนิตีลจนกว่าจะถูกอินนิตีลด้วยผู้ใช้ สัญญาณการขออินเทอร์รัพท์จะต้องสิ้นสุดก่อนที่จะทำงานกับคำสั่ง RETR เพื่อที่จะให้ตัวไมโครโปรเซสเซอร์สามารถที่จะบริการการอินเทอร์รัพท์ ตัวต่อไปได้ทันที มิฉะนั้นจะเข้าอินเทอร์รัพท์เดิม ฉะนั้นอุปกรณ์ต่อพ่วงการอินเทอร์รัพท์จะมีการป้องกันลักษณะเช่นนี้ ด้วยการรีเซทการร้องขอการอินเทอร์รัพท์ เมื่อไรที่ตัวไมโครโปรเซสเซอร์ได้อ่านหรือเขียนข้อมูลจากบัฟเฟอร์รีจิสเตอร์ของอุปกรณ์ต่อพ่วงแล้วถ้าอุปกรณ์การอินเทอร์รัพท์ไม่ต้องการเข้าถึงตัวไมโครโปรเซสเซอร์ก็อาจจะมีการใช้ MCS-48 เป็นตัวส่งสัญญาณตอบรับการรับรู้อินเทอร์รัพท์จากการร้องขอของอุปกรณ์ต่อพ่วงนั้น ขา INT ยังอาจใช้ทดสอบด้วยการใช้คำสั่งกระโดดอย่างมีเงื่อนไข JNl คำสั่งนี้จะใช้เป็นตัวตรวจสอบการอินเทอร์รัพท์ที่เกิดขึ้นก่อนการอินเทอร์รัพท์ MCS-48 จะเกิดขึ้น ถ้าการอินเทอร์รัพท์อยู่ในสภาวะดีสเอนิตีล ก็อาจใช้เป็นตัวตรวจสอบสัญญาณอินพุทธรรมดาทั่วไป เช่นเดียวกับขา To และ T1



รูปที่ 2.21 วงจรตรรกการอินเทอร์รัพท์

2.5.8 ตัวจับเวลา/ตัวนับ (timer/counter)

MCS-48 จะประกอบด้วยตัวจับเวลา/ตัวนับ การทำงานของทั้งสองจะคล้ายกันเพียงแต่ว่าตัวจะมีการส่งอินพุตเข้าตัวนับเท่านั้น

ตัวนับ เป็นตัวนับแบบเลขฐานสองขนาด 8 บิต ที่สามารถกำหนดหรืออ่านค่าได้ด้วยคำสั่ง MOV ที่ส่งผ่านเข้าหรือออกทางแอกคิวมูลเตอร์และการรีเซทจะไม่มีผลต่อค่าที่อยู่ในตัวนับ ตัวนับจะหยุดก็ต่อเมื่อมีการรีเซทหรือใช้คำสั่ง STOP TCNT และจะหยุดจนกระทั่งใช้คำสั่ง START T ในการทำงานเป็นตัวจับเวลาหรือ START CNT ในการทำงานเป็นตัวนับอีกครั้ง ทันทีที่ตัวนับเริ่มนับจะเพิ่มค่าตัวเองที่ละหนึ่ง จนกระทั่งถึงค่าสูงสุด FFH แล้วเริ่มนับใหม่ที่ 00 การนับจะเป็นเช่นนี้เรื่อยไปจนกว่าจะมีคำสั่ง STOP TCNT หรือการรีเซท การเพิ่มค่าจากค่าสูงสุดไปยังค่าศูนย์ (overflow) จะมีผลไปเซตค่าโอเวอร์โฟลของแฟลคฟลิปฟลอป (flag flip-flop) และสร้างสัญญาณการขออินเทอร์รัพท์ขึ้น สถานะของโอเวอร์โฟลแฟลค (overflow flag) สามารถตรวจสอบได้โดยคำสั่งการกระโดดแบบมีเงื่อนไข (JTF) โอเวอร์โฟลแฟลคจะถูกรีเซทก็ต่อเมื่อมีการทำงานของคำสั่ง JTF หรือการรีเซท MCS-48 สัญญาณอินเทอร์รัพท์จะถูกแลทช์และ OR กับการอินเทอร์รัพท์จากภายนอก การอินเทอร์รัพท์ ของตัวนับสามารถอินาเบิลหรือดิสเอนเบิลด้วยคำสั่ง EN TCNTI และ DIS TCNTI หากมีการอินาเบิลเมื่อตัวนับเกิดโอเวอร์โฟลจะไปทำงานโปรแกรมย่อยบริการการทำงานของตัวจับเวลา/ตัวนับที่ตำแหน่ง 007 ทันที ตัวนับสามารถทำหน้าที่เป็นตัวนับเหตุการณ์ (event counter) ได้โดยที่ขา T₁ จะทำหน้าที่เป็นอินพุตโดยสัญญาณจากขา T₁ จะถูกแซมปลิงหลายๆ ช่วงเริ่มต้นสถานะ 3 หรือใน MCS-48 รุ่นใหม่ๆ ก็จะเริ่มที่สถานะ 4 เมื่อมีการเปลี่ยนสถานะจาก 'สูง' ไป 'ต่ำ' ตัวนับจะเพิ่มค่าขึ้นหนึ่งเสมอและ T₁ จะต้องมีสถานะ '0' อย่างน้อย 1 วัฏจักรแมชชีนเพื่อให้แน่ใจว่าการนับจะไม่ขาดหายไป ดังนั้นความเร็วของการนับสูงสุดในแต่ละครั้งจะใช้เวลา 3 รอบคำสั่ง หรือทุกๆ 5.7 ไมโครวินาที เมื่อใช้คริสตอล 8 เมกกะเฮิร์ตซ์ (MHz.) และอินพุตเข้า T₁ จะต้องมียุติลจิกคองที่ '1' อย่างน้อย 1/5 วัฏจักรแมชชีน หลังจากการเปลี่ยนระดับในแต่ละครั้ง

ตัวจับเวลา เมื่อใช้คำสั่ง START T ตัวจับเวลา/ตัวนับจะทำหน้าที่เป็นตัวจับเวลาในการใช้คำสั่ง START T นั้นจะทำให้ฐานเวลากายในถูกหารด้วย 15 และ 32 ตามลำดับ จากผลที่ได้จะเพิ่มค่าของตัวจับเวลาที่ได้ตั้งได้ก่อนแล้ว เมื่อเกินค่าที่กำหนดจะทำการอินเทอร์รัพท์ ของตัวจับเวลา/ตัวนับ เพื่อไปยังโปรแกรมย่อยที่ต้องการ

ตัวอย่างเช่น เมื่อใช้คริสตอลขนาด 11 เมกกะเฮิร์ตซ์เป็นฐานเวลาให้ MCS-48 เมื่อผ่านตัวหาร 15 จะให้ความถี่ 733 กิโลเฮิร์ตซ์ (KHz.) และความถี่นี้จะถูกหารด้วย 32 อีกครั้ง จะให้ความถี่ 22.917 กิโลเฮิร์ตซ์ซึ่งแสดงว่าตัวตั้งเวลาจะเพิ่มขึ้นหนึ่งทุกๆ 44 ไมโครวินาที

2.5.9 สัญญาณนาฬิกาและวงจรฐานเวลา (clock and timer circuit)

แหล่งกำเนิดฐานเวลาสำหรับ MCS-48 สามารถสร้างได้จากอุปกรณ์ภายนอก เช่น คริสตอล, ขดลวดเหนี่ยวนำ (inductor) หรือแหล่งกำเนิดสัญญาณนาฬิกาภายนอก การทำงานของส่วนสัญญาณนาฬิกาและวงจรฐานเวลาแสดงในรูปที่ 2.22 โดยแยกกล่าวเป็นส่วนๆ ดังนี้

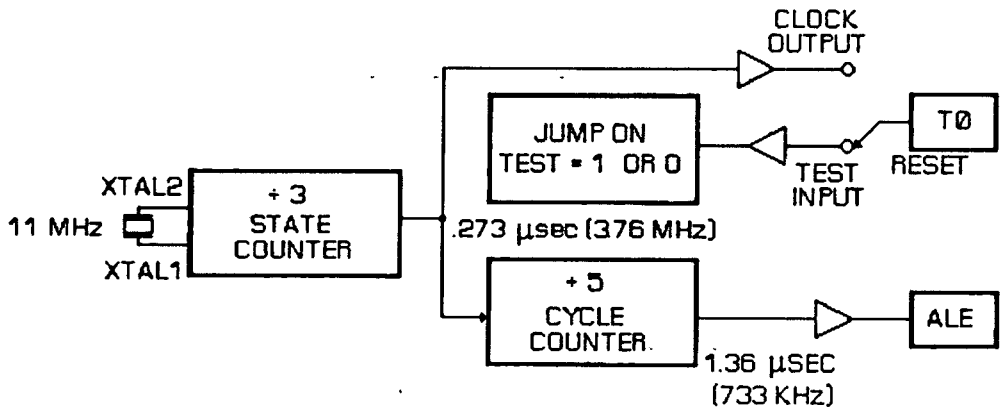
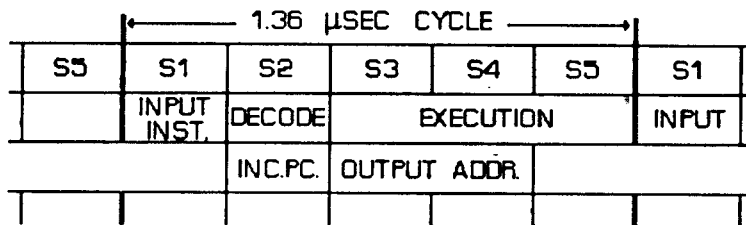


DIAGRAM OF 8048AH CLOCK UTILITIES



INSTRUCTION CYCLE

รูปที่ 2.22 แสดงโครงสร้างของส่วนสัญญาณนาฬิกาและจังหวะเวลาวัฏจักร

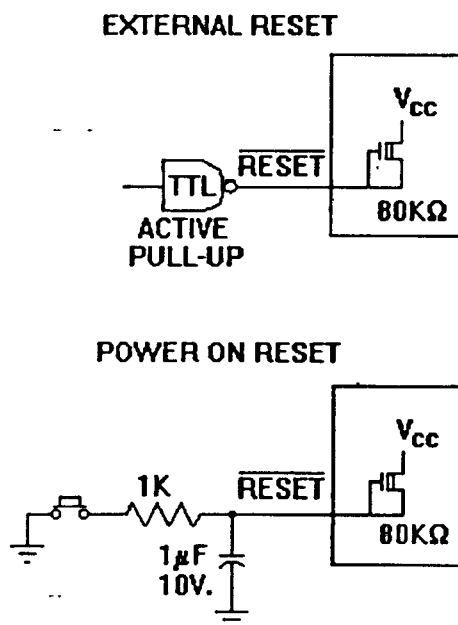
ตัวกำเนิดความถี่ (oscillator) ตัวกำเนิดความถี่เป็นวงจรแบบ series resonant ที่มีอัตราขยายสูงโดยมีช่วงมีความถี่ระหว่าง 1-6 เมกกะเฮิร์ตซ์ ซึ่งขา X1 เป็นขาอินพุตเข้าสู่ วงจรขยาย ขณะที่ขา X2 เป็นขาเอาต์พุตในการติดคริสตอล หรือ ขดลวดเหนี่ยวนำระหว่างขา X1 และ X2 จะทำให้เกิดการป้อนกลับและการเลื่อนเฟสสำหรับการกำเนิดความถี่ในกรณีที่ไม่ต้องการใช้ความถี่ที่แน่นอนสามารถใช้อุปกรณ์พวกขดลวดเหนี่ยวนำแทนคริสตอลได้ ซึ่งจะทำให้ความถี่อยู่ในช่วง 3-5 เมกกะเฮิร์ตซ์

ตัวนับสถานะ (state counter) เอาท์พุทของตัวกำเนิดความถี่จะถูกหารด้วย 3 ในตัวนับสถานะ เพื่อสร้างสัญญาณนาฬิกาสำหรับการทำงานของ MCS-48 สัญญาณนาฬิกาที่สามารถส่งออกไปยังขา T_0 ได้โดยใช้คำสั่ง ENT_0 CLK สัญญาณนาฬิกาที่ขา T_0 จะหยุดเมื่อถูกรีเซท

ตัวนับรอบ (cycle counter) สัญญาณนาฬิกาจะถูกหารด้วย 5 ในตัวนับรอบเพื่อสร้างสัญญาณนาฬิกาที่ประกอบด้วยรอบการทำงาน 5 สถานะ สัญญาณนาฬิกาเรียกว่า แอดเดรสแลทช์เอินาเบิล (address latch enable:ALE) เพื่อใช้เป็นสัญญาณติดต่อกับหน่วยความจำโปรแกรมภายนอกสัญญาณนี้จะออกมาอย่างต่อเนื่องตลอดเวลาขณะที่ MCS-48 ยังทำงานอยู่

2.5.10 การรีเซท

การรีเซทจะทำให้ MCS-48 เริ่มทำงานใหม่เมื่อขา รีเซท มีลอจิกเป็น 0 ซึ่งขานี้จะต้องทำงานแบบชmitt-ทริกเกอร์ (schmitt-trigger) โดยจะมีความต้านทานที่ใช้พูลล์อัปอยู่ภายใน MCS-48 และภายนอกจะต่อตัวเก็บประจุค่า 1 ไมโครฟารัดอยู่เพื่อทำให้เกิดช่วงเวลาเพียงพอในการรีเซท การรีเซท MCS-48 สามารถทำได้ดังแสดงดังรูปที่ 2.23



รูปที่ 2.23 แสดงการรีเซท MCS-48

หากมีสัญญาณรีเซทจากภายนอกที่ขารีเซทจะต้องมีค่าลอจิก 0 อย่างน้อย 50 มิลลิวินาที หลังจากแหล่งจ่ายไฟอยู่ในสภาพที่พร้อมจะใช้งานแล้วเมื่อมีการรีเซท ให้ MCS-48 จะทำให้เกิด ตัวนับโปรแกรมถูกเซทให้เป็น 0 ตัวชี้สแตคถูกเซทเป็น 0 รีจิสเตอร์แบงก์ 0 จะถูกเลือก หน่วยความจำโปรแกรมแบงก์ 0 ถูกเลือก บัสจะถูกเซทเป็นไฮอิมพีแดนซ์ ยกเว้นเมื่อขา EA มีสถานะลอจิก 1 พอร์ต 1 และพอร์ต 2 จะอยู่ในกรณีเป็นอินพุท อินเทอร์รัพท์ จะถูกดิสเอเบิล ตัวจับเวลาจะหยุด แฟลคของตัวจับเวลาถูกทำให้เป็น 0 แฟลค 0 และแฟลค 1 ถูกทำให้เป็น 0

2.6 แนวทางการใช้งาน LCD module

ปัจจุบัน LCD เป็นที่นิยมกันอย่างมาก สำหรับการแสดงผลในเครื่องมือ เครื่องใช้ต่างๆ ทั้งนี้เนื่องจากมีความเหมาะสมด้วยประการทั้งปวง ทั้งในด้านของการกินกระแสต่ำ สามารถแสดงผลเป็นตัวอักษรตัวเลข หรือแสดงเป็นกราฟฟิค (graphic) ได้ (เฉพาะรุ่น) จะติดปัญหาก็คือในด้านวงจร ซึ่งมีระบบการทำงานที่ซับซ้อนและหาอุปกรณ์ได้ค่อนข้างยาก แต่ขณะนี้ผู้ผลิต LCD จะทำรุ่นที่เป็น LCD module ออกมา คือเป็น module ที่มีตัว LCD และวงจรควบคุมมาให้พร้อม เรียกว่า LCM ซึ่งทำให้ผู้ใช้สามารถต่อเข้ากับระบบไมโครโปรเซสเซอร์ (microprocessor) ได้ง่ายและสะดวกสำหรับการเขียนโปรแกรม รวมทั้งมีจำหน่ายกันอย่างกว้างขวาง และมีราคาที่เหมาะสม ทำให้ผู้ใช้ทางด้านไมโครโปรเซสเซอร์หันมาใช้แผงแสดงด้วย LCD module กันมากขึ้น

LCD module มีอยู่มากมายหลายรุ่น และมีคุณสมบัติแตกต่างกันไป ซึ่งแบ่งได้เป็น 2 กลุ่มหลักคือ แบบดอทแมทริกซ์ (dot matrix) และ กราฟฟิค โดยแบบดอทแมทริกซ์จะแสดงผลเป็นอักษรขนาด 5x8 dot และมีจำนวนอักษรและบรรทัดแตกต่างกันไปในแต่ละรุ่น ส่วนแบบกราฟฟิคจะสามารถแสดงผลในแบบบิตแมพ (bitmap) คือสร้างเป็นภาพใดๆ ก็ได้ตามต้องการ แนวทางในการใช้งานของทั้ง 2 แบบ จะมีลักษณะใกล้เคียงกัน การใช้งานโดยทั่วไปมักจะใช้แบบดอทแมทริกซ์มากกว่า เนื่องจากมีราคาถูก และเพียงพอต่องานส่วนใหญ่ ดังนั้นจะกล่าวถึงการใช้งานกับดอทแมทริกซ์เท่านั้น คุณสมบัติของดอทแมทริกซ์ LCD module สามารถสรุปเป็นข้อๆ ได้ดังนี้

1. มีให้เลือกหลายรุ่นตามการใช้งาน โดยมีจำนวนตัวอักษรและบรรทัดแตกต่างกันไป
2. ตัวอักษรแสดงด้วยดอทแมทริกซ์ขนาด 5x8 dot
3. สามารถต่อเข้ากับระบบไมโครโปรเซสเซอร์ได้ 2 ลักษณะ คือแบบเมมโมรีแมพ (memory map) (20-pin LCD bus) และแบบต่อผ่าน 8255 พอร์ต (port) (26-pin 8255 bus)

โดยกรณี 26-pin 8255 bus จะใช้แผ่น PCB (DMCAD) เป็นตัวอะแดปเตอร์ (adapter) ทำให้เป็น 8255 bus อีกที

4. การใช้งานง่ายและสะดวก ระบบไมโครโปรเซสเซอร์เพียงแค่ส่งข้อมูลให้กับ LCD module เท่านั้น ข้อความก็จะปรากฏบนแผงแสดง และจะคงค้างไว้ตลอด ทำให้ไม่ต้องเสียเวลาหลักของระบบไมโครโปรเซสเซอร์

5. มีคำสั่งพิเศษสำหรับอำนวยความสะดวกมากมาย เช่น clear display, home cursor, on off cursor, blink character และอื่นๆ อีก

6. สามารถแสดงผลเป็นตัวอักษรภาษาอังกฤษและตัวเลขได้ 160 ตัว และสัญลักษณ์พิเศษอีก 32 ตัว รวมทั้งสามารถกำหนดอักษรที่ออกแบบเองได้อีก 8 ตัว

7. กินกระแสต่ำ และมีน้ำหนักเบา รวมทั้งทำงานได้ด้วยไฟเลี้ยงระดับ 5 โวลต์เท่านั้น

ขาสัญญาณของ LCD module

ขา	สัญลักษณ์	ระดับ	หน้าที่
1	V _{ss}		0 V. GND
2	V _{cc}		+ 5 V. power supply
3	V _{ee}		+ V for liquid crystal drive
4	RS	H/L	register select H: data input L: instruction input
5	R/W	H/L	H: data read L: data write
6	E	H	enable signal (LH)
7	DB 0	H/L	data bus bit 0
8	DB 1	H/L	data bus bit 1
9	DB 2	H/L	data bus bit 2
10	DB 3	H/L	data bus bit 3
11	DB 4	H/L	data bus bit 4
12	DB 5	H/L	data bus bit 5
13	DB 6	H/L	data bus bit 6
14	DB 7	H/L	data bus bit 7

ตารางที่ 2.2 แสดงขาสัญญาณของ LCD module

2.6.1 การต่อเข้ากับระบบไมโครโปรเซสเซอร์

LCD module จะต่อเข้ากับระบบไมโครโปรเซสเซอร์ได้ 2 ลักษณะ คือ แบบเมมโมรีแมพโดยผ่าน LCD bus ขนาด 20 pin และแบบ I/O port โดยผ่าน 8255 bus ขนาด 26 pin โดยแต่ละแบบจะมีหลักการดังนี้

การต่อแบบเมมโมรีแมพ

1. สามารถต่อเข้ากับชิพ (chip) เบอร์ต่างๆ ไปได้ เช่น 8051 หรือ z80 โดยจะทำให้ระบบไมโครโปรเซสเซอร์มองเห็น LCD module ในลักษณะของหน่วยความจำ (memory) ได้ทันที

2. ผู้ใช้สามารถเขียนและอ่านข้อมูลจาก LCD module ได้ ทำให้มองเสมือนว่าเป็นเมมโมรีบัฟเฟอร์ (memory buffer) ไปในตัว

3. เนื่องจากสามารถอ่านข้อมูลกลับได้ จึงทำให้สามารถตรวจสอบแฟล็ก (flag) ความพร้อมในขณะที่ LCD module กำลังทำงานได้

4. ใช้ได้กับบอร์ด (board) ที่มี LCD bus มาให้พร้อมเท่านั้น

5. ทำให้กินพื้นที่ของหน่วยความจำไปส่วนหนึ่ง และต้องมีการเข้ารหัส (decode) ละเอียดพอสมควร

6. การจัดขาสัญญาณจะต้องเป็นไปตามแบบของชิพแต่ละเบอร์ด้วย

การต่อแบบ I/O port

1. สามารถต่อเข้ากับ I/O port ใดๆ ก็ได้ โดยใช้สายสัญญาณจำนวน 11 เส้น และใช้โปรแกรมเป็นตัวสร้างสัญญาณขึ้นมากให้ตรงกับข้อกำหนดของ LCD module

2. ผู้ใช้จะเขียนข้อมูลให้ LCD module ได้อย่างเดียว ซึ่งผู้ใช้ควรจะกำหนดหน่วยความจำส่วนหนึ่งให้เป็นเสมือนบัฟเฟอร์ให้กับ LCD module อีกที

3. เนื่องจากไม่สามารถอ่านข้อมูลกลับได้ จึงต้องใช้การหน่วงเวลาของระบบไมโครโปรเซสเซอร์เอง เพื่อรอให้ LCD module กระทำขบวนการต่างๆ

4. ใช้ได้กับบอร์ดทุกๆ ไปที่มีพอร์ต

5. ไม่เปลืองส่วนของหน่วยความจำในการใช้งาน

6. การจัดขาสัญญาณกระทำได้อย่างอิสระ

2.6.2 ชุดควบคุมคำสั่งและการแสดงข้อความ

การเขียนหรืออ่านข้อมูลกับ LCD module ก็คือการกำหนดคุณสมบัติต่างๆ ในการใช้งานของ LCD ตามชุดคำสั่งควบคุม และรวมถึงการเขียนข้อมูลที่เป็นข้อความ เพื่อให้ปรากฏบนแผงแสดงด้วย โดยมีรายละเอียดตามตารางที่ 2.3

INSTRUCTION	RS	R/W	DATA BIT								EXE.TIME (μ S)	
			7	6	5	4	3	2	1	0		
CLEAR DISPLAY	0	0	0	0	0	0	0	0	0	0	1	1640
CURSOR AT HOME	0	0	0	0	0	0	0	0	0	1	*	1640
ENTRY MODE SET	0	0	0	0	0	0	0	0	1	I/D	S	40
DISPLAY ON/OFF	0	0	0	0	0	0	0	1	D	C	B	40
DISPLAY SHIFT	0	0	0	0	0	0	1	S/C	R/L	*	*	40
FUNCTION SET	0	0	0	0	1	DL	N	F	*	*		40
SET CGRAM ADD.	0	0	0	1	CGRAM ADDRESS						40	
SET DDRAM ADD.	0	0	1	DDRAM ADDRESS							40	
BUSY.ADD READ	0	1	BF	ADDRESS							0	
CGRAM.DDRAM WR	1	0	WRITE DATA								40	
CGRAM.DDRAM RD	1	1	READ DATA								40	

ตารางที่ 2.3 แสดงรายละเอียดของชุดคำสั่ง

ความเข้าใจพื้นฐาน

1. การเขียนข้อมูลให้กับ LCD module แบ่งเป็น 2 ลักษณะคือคำสั่งควบคุม (instruction) และข้อมูล (data) โดยจะกำหนดด้วยขาสัญญาณ RS คือถ้า RS จะหมายถึงส่งคำสั่งควบคุมหรืออ่านค่าแฟล็กสภาพการทำงานของ LCD module และถ้า RS=1 จะหมายถึงการเขียนหรือการอ่านข้อมูลกับ LCD module
2. หลักการในการเขียนข้อมูลให้ LCD module นี้ คือ เมื่อมีการเขียนข้อมูลไปแล้ว ตัว LCD module จะต้องใช้เวลาในการทำงานชั่วขณะหนึ่ง (ตามค่า execute time ในตาราง) ซึ่งระบบไมโครโปรเซสเซอร์สามารถตรวจสอบได้จากบิวชีแฟล็ก (busy flag : BF) และถ้าเรียบร้อยแล้ว จึงจะสามารถเขียนข้อมูลอันต่อไปได้ ในกรณีที่การต่อวงจรเป็นแบบอินพุท/เอาต์พุทพอร์ทคือไม่สามารถอ่านข้อมูลย้อนกลับได้ ระบบไมโครโปรเซสเซอร์ก็จะต้องหน่วงเวลาแทน
3. การเขียนข้อมูลให้กับ LCD module นี้ สามารถทำได้ทั้งแบบ 8 บิต และ 4 บิต โดยกรณี 4 บิต จะใช้สายสัญญาณส่งข้อมูลเพียง 4 เส้น คือ DB4 -DB7 (ใช้สำหรับระบบไมโครโปรเซสเซอร์แบบ 4 บิต หรือเพื่อการประหยัดสาย) การเขียนข้อมูลจะกระทำเหมือนกับ 8 บิต เพียงแต่ให้เขียน 2 ครั้ง คือ DB4 -DB7 ก่อน แล้วตามด้วย DB0 -DB3 และจะต้องกำหนดคุณสมบัติตามค่า DL ในชุดคำสั่ง (function set) ด้วย
4. DDRAM (display data RAM) คือหน่วยความจำภายในตัว LCD module ที่เป็นบัฟเฟอร์ของข้อมูล โดยถ้าเขียนรหัสแอสกี (ASCII) ใดๆ ลงไปในหน่วยความจำนี้ ก็จะปรากฏเป็นตัวอักษรที่แผงแสดงทันที
5. CGRAM (character generator RAM) คือหน่วยความจำภายในตัว LCD module สำหรับเก็บภาพตัวอักษรที่ผู้ใช้สามารถสร้างได้เอง (8 ตัว) โดยจะอ้างแอดเดรส (address) ได้ทั้งหมด 64 ไบต์ (byte) คือ 8 ตัวอักษร คูณกับ 8 row

รายละเอียดของแต่ละคำสั่ง

1. clear display

R	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
S									
0	0	0	0	0	0	0	0	0	1

สำหรับการ clear display โดยจะทำการเขียนตัวอักษรสเปซ (space) ลงใน DDRAM แอดเดรสทั้งหมดให้เป็น 0 พร้อมทั้งเคอร์เซอร์ (cursor) จะกลับไปตำแหน่งบนสุดของจอภาพ

2. cursor at home

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	*

สำหรับกำหนดค่า DDRAM แอดเดรสให้เป็น 0 พร้อมทั้งเคอร์เซอร์จะไปอยู่ที่ตำแหน่งซ้ายบนสุดของจอภาพ โดยที่ข้อมูลใน DDRAM ไม่มีการเปลี่ยนแปลง

3. entry mode set

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

I/D = 0 กำหนดทิศทางของเคอร์เซอร์ และ DDRAM ให้เป็นแบบลดทีละ 1 (decrement)

I/O = 1 กำหนดทิศทางของเคอร์เซอร์และ DDRAM ให้เป็นแบบเพิ่มทีละ 1 (increment)

S = 0 เมื่อเขียนข้อมูลแล้ว ตัวเคอร์เซอร์จะถูกเลื่อนไปทิศทางตามค่า I/D

S = 1 เมื่อเขียนข้อมูลแล้ว ตัวเคอร์เซอร์จะอยู่กับที่ และตัวอักษรจะถูกดันไปทิศทางตามค่า I/D

การกำหนด I/D และ S นี้ ให้กำหนดก่อนการเขียนข้อมูลใน DDRAM และเมื่อกำหนดแล้วจะต้องไม่ใช่คำสั่ง clear display อีก

4. display on/off

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

D = 0 กำหนดให้ off display

D = 1 กำหนดให้ on display

C = 0 กำหนดให้ off เคอร์เซอร์

C = 1 กำหนดให้ on เคอร์เซอร์ โดยเคอร์เซอร์จะเป็นเส้นขีดใต้ตัวอักษร

B = 0 กำหนดให้ไม่มีการกระพริบที่ตำแหน่งเคอร์เซอร์

B = 1 กำหนดให้มีการกระพริบที่ตำแหน่งเคอร์เซอร์ (กระพริบเป็นรูปสี่เหลี่ยม)

5. display shift

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	*	*

S/C = 0 กำหนดให้เลื่อนเคอร์เซอร์ตามทิศทาง R/L ไป 1 ตำแหน่ง

S/C = 1 กำหนดให้เลื่อนข้อความบนแผงแสดงตามทิศทาง R/L ไป 1 column (เลื่อนทุกบรรทัด)

R/L = 0 กำหนดไม่มีทิศทางไปทางซ้าย

R/L = 1 กำหนดให้มีทิศทางไปทางขวา

6. function set

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	*	*

DL = 0 กำหนดให้การติดต่อกับ LCD module เป็นแบบ 4 บิต

DL = 1 กำหนดให้การติดต่อกับ LCD module เป็นแบบ 8 บิต

จะสังเกตเห็นว่า การกำหนดค่า DL นี้ สามารถกระทำได้ที่ DB4 - DB7 ซึ่งถ้ามีการกำหนดให้เป็นแบบ 4 บิต ตั้งแต่ครั้งแรก หลังจากจ่ายไฟเลี้ยง ก็จะทำให้ LCD module มีการรับข้อมูลแบบ 4 บิตทันที

N = 0 กำหนดจำนวนบรรทัดแบบ 1/8 ดิวตี้ (duty) และ 1/11 ดิวตี้

D = 1 กำหนดจำนวนบรรทัดแบบ 1/16 ดิวตี้

F = 0 กำหนดให้ตัวอักษรเป็นแบบ 5x7 dots

F = 1 กำหนดให้ตัวอักษรเป็นแบบ 5x10 dots (กรณีที่ LCD module เป็นแบบ 5x7 dots อยู่แล้วก็จะมีผลอะไร)

7. set CGRAM address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	CGRAM ADDRESS					

สำหรับการกำหนดแอดเดรสของ CGRAM เมื่อได้ทำการกำหนดไว้แล้ว การอ่านและเขียนข้อมูลต่อจากนี้จะเป็นไปตามแอดเดรสที่กำหนดทันที

8. set DDRAM address

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	1	DDRAM ADDRESS						

สำหรับการกำหนดแอดเดรสของ DDRAM เมื่อได้ทำการกำหนดไว้แล้ว การอ่านและเขียนข้อมูลต่อจากนี้ จะเป็นไปตามแอดเดรสที่กำหนดทันที ตำแหน่งของแอดเดรสของแต่ละรุ่นจะมีความแตกต่างกันบ้าง เพราะจำนวนต่อบรรทัดไม่เท่ากัน

9. busy flag and address read

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	1	BF	ADDRESS						

สำหรับการอ่านค่าบิตซีเฟลก ซึ่งบอกถึงความพร้อมของ LCD module ในการรับข้อมูล ถ้า BF = 0 หมายถึงว่าพร้อมที่จะรับข้อมูลต่อไปได้ แต่ถ้า BF = 1 หมายถึงว่ายังไม่พร้อม นอกจากนี้ยังเป็นการอ่านค่าแอดเดรสของ CGRAM หรือ DDRAM อีกด้วย

การอ่านและการเขียนข้อมูลกับ DDRAM/CGARAM

1. write data to DDRAM or CGRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	0	DATA							

สำหรับการเขียนข้อมูลลงหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้วแอดเดรสจะถูกเพิ่มหรือลดโดยอัตโนมัติ ตามที่กำหนดจากค่า I/D ในคำสั่ง entry mode set และการเขียนจะเป็น DDRAM หรือ CGRAM ก็ขึ้นอยู่กับว่าก่อนหน้าคำสั่งนี้ มีการกำหนดแอดเดรสที่ได้

2. read data from DDRAM or CGRAM

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
1	1	DATA							

สำหรับการอ่านข้อมูลจากหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้วแอดเดรสจะถูกเพิ่มหรือลดโดยอัตโนมัติ ตามที่กำหนดจากค่า I/D ในคำสั่ง entry mode set และการอ่านจะเป็น DDRAM หรือ CGRAM ก็ขึ้นอยู่กับว่า ก่อนหน้าคำสั่งนี้มีการกำหนดแอดเดรสที่ได้

บทที่ 3
การคำนวณและการสร้าง

ส่วนของวงจร voltage zero-crossing detector ซึ่งเป็นตัวเปรียบเทียบแรงดัน (voltage comparator) ทำให้ได้ค่าแรงดันไฟฟ้าฮิสทีรีซิสประมาณ 100 มิลลิโวลต์ (mV)

ซึ่งหาได้จากการคำนวณจากสูตร

$$\begin{aligned} V_{UT} &= \frac{R_3}{R_2 + R_3} (+V_{sat}) \\ &= \frac{100}{12K + 100} (+5) \\ &= 41 \text{ mV} \end{aligned}$$

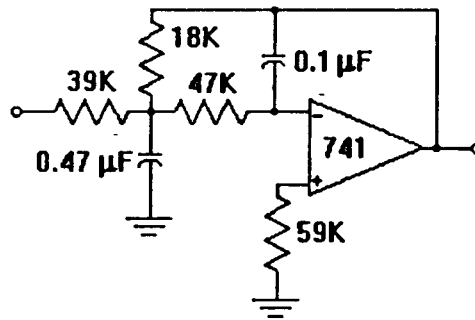
และ

$$\begin{aligned} V_{LT} &= \frac{R_7}{R_6 + R_7} (-V_{sat}) \\ &= \frac{100}{12K + 100} (-5) \\ &= -41 \text{ mV} \end{aligned}$$

ดังนั้นจะได้

$$V_H = V_{UT} - V_{LT} = 41 - (-41) = 82 \text{ mV}$$

หาค่าความถี่คัทออฟ, ค่าเกน และค่า Q ของวงจรผ่านความถี่ต่ำ



เปรียบเทียบกับวงจรในรูปที่ 2.9 และสมการที่ 2.4 จะได้ว่า

$$Y_A = \frac{1}{39K} = 25.64 \times 10^{-6}$$

$$Y_C = \frac{1}{47K} = 21.28 \times 10^{-6}$$

$$Y_D = \frac{1}{10K} = 100 \times 10^{-6}$$

$$Y_B = 0.1 \times 10^{-6} \text{ s}$$

$$Y_E = 0.47 \times 10^{-6} \text{ s}$$

แทนค่าลงในสมการที่ 2.4 จะได้

$$\begin{aligned} T_V &= \frac{-(25.64 \times 21.28)}{0.1s(25.64 + 21.28 + 100 + 0.47s) + 21.28 \times 100} \\ &= \frac{-545.6}{0.047s^2 + 14.69s + 2128} \\ &= \frac{-11609}{s^2 + 312.5s + 45276.6} \end{aligned}$$

เทียบสมการที่คิดได้กับสมการที่ 2.3 จะได้ว่า

$$\omega_p = 212.78$$

$$f_p = \frac{\omega_p}{2\pi} = 33.87 \text{ Hz}$$

$$K = \frac{11609}{45276.6} = 0.256$$

$$Q = \frac{312.5}{212.78} = 1.47$$

ในการคำนวณหาค่าเพาเวอร์แฟกเตอร์ โดยการเปรียบเทียบค่าแรงดันไฟฟ้าไปเป็นค่า
มอมของความต่างเฟสแล้วนำไปหาค่าเพาเวอร์แฟกเตอร์ และเปรียบเทียบค่าแรงดันไฟฟ้าไป
เป็นระดับเลขจำนวนเต็ม ซึ่งถ้ามีทศนิยมเกินจุดห้าให้ปัดขึ้น

ค่าแรงดันไฟฟ้า (โวลต์)	ค่ามอม (องศา)	ค่าเพาเวอร์ แฟกเตอร์	ระดับ
3.00	90.0	0	255.00
2.99	89.7	0.006030	254.15
2.98	89.4	0.011263	253.30
2.97	89.1	0.016496	252.45
2.96	88.8	0.021728	251.60
2.95	88.5	0.026960	250.75
2.94	88.2	0.032191	249.90
2.93	87.9	0.037421	249.05
2.92	87.6	0.042650	248.20
2.91	87.3	0.047878	247.35
2.90	87.0	0.053105	246.50
2.89	86.7	0.058330	245.65
2.88	86.4	0.063553	244.80
2.87	86.1	0.068775	243.95
2.86	85.8	0.073995	243.10
2.85	85.5	0.079213	242.25
2.84	85.2	0.084429	241.40
2.83	84.9	0.089642	240.55
2.82	84.6	0.094854	239.70
2.81	84.3	0.100062	238.85
2.80	84.0	0.105268	238.00
2.79	83.7	0.110470	237.15
2.78	83.4	0.115670	236.30
2.77	83.1	0.120867	235.45
2.76	82.8	0.126060	234.60
2.75	82.5	0.131250	233.75
2.74	82.2	0.136436	232.90
2.73	81.9	0.141619	232.05
2.72	81.6	0.146797	231.20
2.71	81.3	0.151972	230.35
2.70	81.0	0.157142	229.50
2.69	80.7	0.162308	228.65
2.68	80.4	0.167470	227.80
2.67	80.1	0.172627	226.95
2.66	79.8	0.177780	226.10
2.65	79.5	0.182927	225.25

2.64	79.2	0.188070	224.40
2.63	78.9	0.193207	223.55
2.62	78.6	0.198339	222.70
2.61	78.3	0.203466	221.85
2.60	78.0	0.208587	221.00
2.59	77.7	0.213702	220.15
2.58	77.4	0.218812	219.30
2.57	77.1	0.223915	218.45
2.56	76.8	0.229012	217.60
2.55	76.5	0.234103	216.75
2.54	76.2	0.239188	215.90
2.53	75.9	0.244266	215.05
2.52	75.6	0.249338	214.20
2.51	75.3	0.254402	213.35
2.50	75.0	0.259460	212.50
2.49	74.7	0.264511	211.65
2.48	74.4	0.269554	210.80
2.47	74.1	0.274590	209.95
2.46	73.8	0.279618	209.10
2.45	73.5	0.284639	208.25
2.44	73.2	0.289652	207.40
2.43	72.9	0.294657	206.55
2.42	72.6	0.299654	205.70
2.41	72.3	0.304642	204.85
2.40	72.0	0.309623	204.00
2.39	71.7	0.314595	203.15
2.38	71.4	0.319558	202.30
2.37	71.1	0.324513	201.45
2.36	70.8	0.329458	200.60
2.35	70.5	0.334395	199.75
2.34	70.2	0.339322	198.90
2.33	69.9	0.344240	198.05
2.32	69.6	0.349149	197.20
2.31	69.3	0.354048	196.35
2.30	69.0	0.358938	195.50
2.29	68.7	0.363818	194.65
2.28	68.4	0.368687	193.80
2.27	68.1	0.373547	192.95
2.26	67.8	0.378396	192.10
2.25	67.5	0.383235	191.25
2.24	67.2	0.388064	190.40
2.23	66.9	0.392882	189.55
2.22	66.6	0.397689	188.70
2.21	66.3	0.402485	187.85
2.20	66.0	0.407270	187.00
2.19	65.7	0.412044	186.15

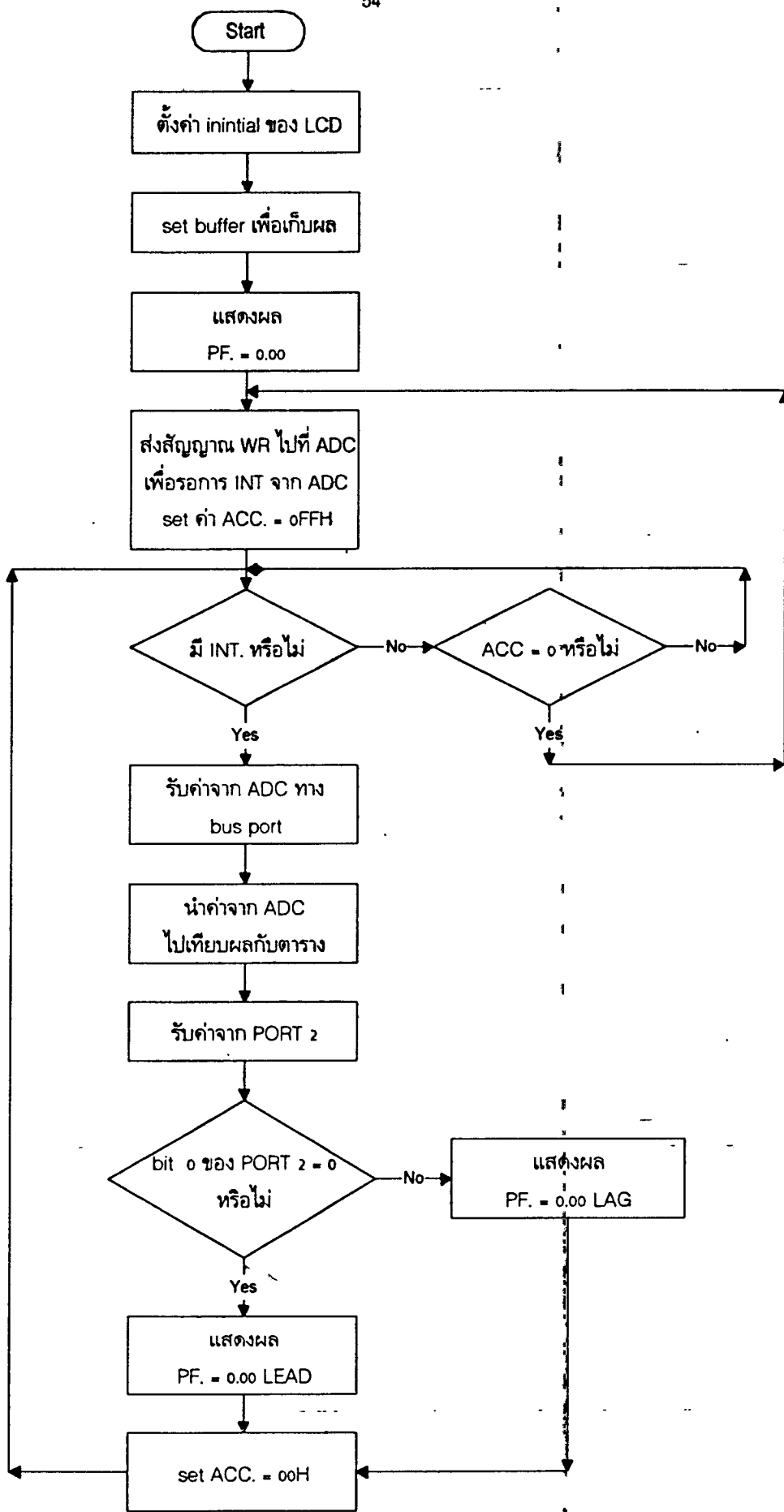
2.18	65.4	0.416807	185.30
2.17	65.1	0.421558	184.45
2.16	64.8	0.426298	183.60
2.15	64.5	0.431026	182.75
2.14	64.2	0.435742	181.90
2.13	63.9	0.440447	181.05
2.12	63.6	0.445139	180.20
2.11	63.3	0.449819	179.35
2.10	63.0	0.454487	178.50
2.09	62.7	0.459142	177.65
2.08	62.4	0.463785	176.80
2.07	62.1	0.468415	175.95
2.06	61.8	0.473033	175.10
2.05	61.5	0.477637	174.25
2.04	61.2	0.482228	173.40
2.03	60.9	0.486806	172.55
2.02	60.6	0.491371	171.70
2.01	60.3	0.495922	170.85
2.00	60.0	0.500460	170.00
1.99	59.7	0.504984	169.15
1.98	59.4	0.509494	168.30
1.97	59.1	0.513990	167.45
1.96	58.8	0.518472	166.60
1.95	58.5	0.522940	165.75
1.94	58.2	0.527393	164.90
1.93	57.9	0.531832	164.05
1.92	57.6	0.536257	163.20
1.91	57.3	0.540667	162.35
1.90	57.0	0.545062	161.50
1.89	56.7	0.549442	160.65
1.88	56.4	0.553807	159.80
1.87	56.1	0.558157	158.95
1.86	55.8	0.562492	158.10
1.85	55.5	0.566811	157.25
1.84	55.2	0.571115	156.40
1.83	54.9	0.575403	155.55
1.82	54.6	0.579675	154.70
1.81	54.3	0.583931	153.85
1.80	54.0	0.588172	153.00
1.79	53.7	0.592396	152.15
1.78	53.4	0.596604	151.30
1.77	53.1	0.600796	150.45
1.76	52.8	0.604971	149.60
1.75	52.5	0.609130	148.75
1.74	52.2	0.613272	147.90
1.73	51.9	0.617397	147.05

1.72	51.6	0.621506	146.20
1.71	51.3	0.625597	145.35
1.70	51.0	0.629671	144.50
1.69	50.7	0.633728	143.65
1.68	50.4	0.637768	142.80
1.67	50.1	0.641790	141.95
1.66	49.8	0.645794	141.10
1.65	49.5	0.649781	140.25
1.64	49.2	0.653750	139.40
1.63	48.9	0.657701	138.55
1.62	48.6	0.661634	137.70
1.61	48.3	0.665549	136.85
1.60	48.0	0.669446	136.00
1.59	47.7	0.673325	135.15
1.58	47.4	0.677185	134.30
1.57	47.1	0.681026	133.45
1.56	46.8	0.684849	132.60
1.55	46.5	0.688653	131.75
1.54	46.2	0.692438	130.90
1.53	45.9	0.696204	130.05
1.52	45.6	0.699952	129.20
1.51	45.3	0.703680	128.35
1.50	45.0	0.707388	127.50
1.49	44.7	0.711078	126.65
1.48	44.4	0.714747	125.80
1.47	44.1	0.718398	124.95
1.46	43.8	0.722028	124.10
1.45	43.5	0.725639	123.25
1.44	43.2	0.729230	122.40
1.43	42.9	0.732801	121.55
1.42	42.6	0.736352	120.70
1.41	42.3	0.739883	119.85
1.40	42.0	0.743393	119.00
1.39	41.7	0.746884	118.15
1.38	41.4	0.750353	117.30
1.37	41.1	0.753802	116.45
1.36	40.8	0.757231	115.60
1.35	40.5	0.760639	114.75
1.34	40.2	0.764026	113.90
1.33	39.9	0.767392	113.05
1.32	39.6	0.770737	112.20
1.31	39.3	0.774060	111.35
1.30	39.0	0.777363	110.50
1.29	38.7	0.780644	109.65
1.28	38.4	0.783904	108.80
1.27	38.1	0.787143	107.95

1.26	37.8	0.790360	107.10
1.25	37.5	0.793555	106.25
1.24	37.2	0.796729	105.40
1.23	36.9	0.799881	104.55
1.22	36.6	0.803011	103.70
1.21	36.3	0.806118	102.85
1.20	36.0	0.809204	102.00
1.19	35.7	0.812268	101.15
1.18	35.4	0.815309	100.30
1.17	35.1	0.818328	99.45
1.16	34.8	0.821325	98.60
1.15	34.5	0.824299	97.75
1.14	34.2	0.827251	96.90
1.13	33.9	0.830180	96.05
1.12	33.6	0.833086	95.20
1.11	33.3	0.835969	94.35
1.10	33.0	0.838830	93.50
1.09	32.7	0.841667	92.65
1.08	32.4	0.844482	91.80
1.07	32.1	0.847273	90.95
1.06	31.8	0.850041	90.10
1.05	31.5	0.852786	89.25
1.04	31.2	0.855507	88.40
1.03	30.9	0.858205	87.55
1.02	30.6	0.860880	86.70
1.01	30.3	0.863531	85.85
1.00	30.0	0.866158	85.00
0.99	29.7	0.868762	84.15
0.98	29.4	0.871341	83.30
0.97	29.1	0.873897	82.45
0.96	28.8	0.876429	81.60
0.95	28.5	0.878937	80.75
0.94	28.2	0.881421	79.90
0.93	27.9	0.883881	79.05
0.92	27.6	0.886317	78.20
0.91	27.3	0.888728	77.35
0.90	27.0	0.891115	76.50
0.89	26.7	0.893478	75.65
0.88	26.4	0.895816	74.80
0.87	26.1	0.898129	73.95
0.86	25.8	0.900418	73.10
0.85	25.5	0.902682	72.25
0.84	25.2	0.904922	71.40
0.83	24.9	0.907137	70.55
0.82	24.6	0.909327	69.70
0.81	24.3	0.911492	68.85

0.80	24.0	0.913632	68.00
0.79	23.7	0.915747	67.15
0.78	23.4	0.917837	66.30
0.77	23.1	0.919902	65.45
0.76	22.8	0.921941	64.60
0.75	22.5	0.923956	63.75
0.74	22.2	0.925945	62.90
0.73	21.9	0.927909	62.05
0.72	21.6	0.929847	61.20
0.71	21.3	0.931760	60.35
0.70	21.0	0.933647	59.50
0.69	20.7	0.935509	58.65
0.68	20.4	0.937345	57.80
0.67	20.1	0.939155	56.95
0.66	19.8	0.940940	56.10
0.65	19.5	0.942699	55.25
0.64	19.2	0.944432	54.40
0.63	18.9	0.946140	53.55
0.62	18.6	0.947821	52.70
0.61	18.3	0.949476	51.85
0.60	18.0	0.951106	51.00
0.59	17.7	0.952709	50.15
0.58	17.4	0.954286	49.30
0.57	17.1	0.955837	48.45
0.56	16.8	0.957362	47.60
0.55	16.5	0.958861	46.75
0.54	16.2	0.960334	45.90
0.53	15.9	0.961780	45.05
0.52	15.6	0.963200	44.20
0.51	15.3	0.964593	43.35
0.50	15.0	0.965960	42.50
0.49	14.7	0.967301	41.65
0.48	14.4	0.968615	40.80
0.47	14.1	0.969902	39.95
0.46	13.8	0.971163	39.10
0.45	13.5	0.972398	38.25
0.44	13.2	0.973606	37.40
0.43	12.9	0.974787	36.55
0.42	12.6	0.975941	35.70
0.41	12.3	0.977069	34.85
0.40	12.0	0.978170	34.00
0.39	11.7	0.979244	33.15
0.38	11.4	0.980291	32.30
0.37	11.1	0.981312	31.45
0.36	10.8	0.982305	30.60
0.35	10.5	0.983272	29.75

0.34	10.2	0.984212	28.90
0.33	9.9	0.985124	28.05
0.32	9.6	0.986010	27.20
0.31	9.3	0.986869	26.35
0.30	9.0	0.987701	25.50
0.29	8.7	0.988506	24.65
0.28	8.4	0.989283	23.80
0.27	8.1	0.990034	22.95
0.26	7.8	0.990757	22.10
0.25	7.5	0.991454	21.25
0.24	7.2	0.992123	20.40
0.23	6.9	0.992765	19.55
0.22	6.6	0.993379	18.70
0.21	6.3	0.993967	17.85
0.20	6.0	0.994527	17.00
0.19	5.7	0.995061	16.15
0.18	5.4	0.995566	15.30
0.17	5.1	0.996045	14.45
0.16	4.8	0.996496	13.60
0.15	4.5	0.996920	12.75
0.14	4.2	0.997317	11.90
0.13	3.9	0.997687	11.05
0.12	3.6	0.998029	10.20
0.11	3.3	0.998343	9.35
0.10	3.0	0.998631	8.50
0.09	2.7	0.998891	7.65
0.08	2.4	0.999124	6.80
0.07	2.1	0.999329	5.95
0.06	1.8	0.999507	5.10
0.05	1.5	0.999658	4.25
0.04	1.2	0.999781	3.40
0.03	0.9	0.999877	2.55
0.02	0.6	0.999945	1.70
0.01	0.3	0.999986	0.85
0	0	1	0



แผนภาพแสดงการทำงานของโปรแกรม

โปรแกรมที่ใช้ในการควบคุมการแสดงผลทางจอ LCD

```

;          R2          : LOCAL COUNTER
;          R3          : GLOBAL VAR
;          CPU         "8048.TBL"
;          HOF         "BIN8"
DISB:     EQU         32      ;DISPLAY BUFFER ADDRESS
DATAB    EQU         49      ;BUFFER OF DATA ADC
          ORG         000H
          JMP         MAIN
          ORG         003H   ;INTERRUPT ROUTINE
          JMP         INT
;MAIN PROGRAM
MAIN:     SEL         RB0
          DIS         TCNTI  ;FOR EFFECT TIMER INTERRUPT
          CALL        INITLCD
          MOV         R2,#0CH ;ADDRESS OF TABLE2
          CALL        SETBUF
          CALL        DISPLY
AGAIN:    MOV         R0,#00H
          MOV         A,#00H
          MOVX        @R0,A
          MOV         A,#0FFH
          EN         I
LOOP:    JNZ         LOOP
          JMP         AGAIN
INT:     SEL         RB1
          MOV         R0,#00H
          MOV         R1,#DATAB
          MOVX        A,@R0
          MOV         @R1,A
          ADD         A,#0F0H  ;CHECK IS OVER 16
          JC         PF
          MOV         R2,#1CH  ;DISPLAY POWER FACTER = 1.00
          CALL        SETBUF
          CALL        DISPLY
          MOV         R6,#0FFH ;SET R6 FOR CHECK PF
          JMP         INT1

```

```

PF:      MOV      A,@R1
        MOVVP3   A,@A      ;CHANGE DATA FROM HEX TO DEC
        MOV      @R1,A
        CALL     CHNDATA   ;CHANGE DATA
        MOV      A,#0FFH
        OUTL     P2,A
        IN       A,P2
        CALL     CHKL
        MOV      A,R6
        JZ       INT1
        MOV      R6,#00H   ;RESET REG. R6 FOR CHECK PF NEW
        MOV      R2,#0CH
        CALL     SETBUF
        JMP      INT2
INT1:    CALL     DISPLY   ;CALL DISPLAY LCD
INT2:    CLR      A
        SEL      RB0
        RETR
;PAGE LCD DISPLAY
        ORG      200H
; ***** INITIALL LCD DISPLAY *****
; P1.0-P1.3 : PIN D4-D7 (DATA READ/WRITE LCD)
; P1.4 : PIN RS (REGISTER SELECTION)
; P1.5 : PIN R/W (READ/WRITE)
; P1.6 : PIN E (ENABLE SIGNAL PULSE)
; *****EVENT*****
INITLCD: MOV      A,#02H   ;SETS TO 4 BIT OPERATION
        CALL     EPULSE   ;ENABLE SIGNAL PLUSE (WRITE)
        MOV      A,#02H   ;FUNCTION SET 28
        CALL     EPULSE   ;DL= 0 4 BIT, N=1 2 LINE ,F=0 5*7 DOT
        MOV      A,#08H
        CALL     EPULSE
        MOV      A,#00H   ;DISPLAY ON-OFF CONTROL
        CALL     EPULSE   ;C=0 CURSOR OFF,B=0 BLINK OFF
        MOV      A,#0CH
        CALL     EPULSE
        MOV      A,#00H   ;ENTRY MODE SET
        CALL     EPULSE   ;I/D=1 INCREMENT,S=0 RIGHT
        MOV      A,#06H
        CALL     EPULSE
        CALL     CLRLCD   ;CLEAR DISPLAY
        RET

```

```

; ***** CLEAR DISPLAY *****
CLR_LCD:  MOV     A,#00H
          CALL    EPULSE
          MOV     A,#01H
          CALL    EPULSE
          MOV     A,#03H
          CALL    DELAY
          RET

; ***** WRITE DATA BYTE TO LCD *****
WR_BYTE:  MOV     R7,A
          SWAP    A
          ANL    A,#0FH
          ORL    A,#50H
          CALL    EPULSE
          MOV     A,R7
          ANL    A,#0FH
          ORL    A,#50H
          CALL    EPULSE
          RET

; ***** GENERATE ENABLE PULSE & SEND DATA OR
INSTRUCTION*****
EPULSE:   ORL    A,#40H
          OUTL   P1,A
          MOV    R2,#0FFH
EP1:      DJNZ   R2,EP1
          ANL   P1,#0BFH
          RET

; ***** MOVE DISPLAY BUFFER TO LCD *****
DISPLY:   MOV    A,#08 H      ;FIRST LINE
          CALL   EPULSE
          MOV    A,#02H
          CALL   EPULSE
          MOV    R0,#DISB     ;START BUFFER DISPLAY
          CALL   LINE
          MOV    A,#0CH      ;SECOND LINE
          CALL   EPULSE
          MOV    A,#00H
          CALL   EPULSE
          CALL   LINE
          RET

```

```

LINE:      MOV      R3,#08H
LINE1:    MOV      A,@R0
          CALL     WRBYTE
          MOV      A,#50H
          CALL     DELAY
          INC      R0
          DJNZ    R3,LINE1
          RET

```

;PAGE TABLE

```

ORG      300H
DFB      00H,00H,00H,00H,00H,00H,00H,00H,00H,00H
DFB      00H,00H,00H,00H,00H,00H,00H,99H,99H,99H
DFB      99H,99H,99H,99H,99H,99H,99H,99H,99H,98H
DFB      98H,98H,98H,98H,98H,98H,98H,97H,97H,97H
DFB      97H,97H,97H,96H,96H,96H,96H,96H,96H,95H
DFB      95H,95H,95H,95H,95H,95H,94H,94H,94H,94H
DFB      93H,93H,93H,93H,92H,92H,92H,92H,91H,91H
DFB      91H,90H,90H,90H,90H,90H,89H,89H,89H,88H
DFB      88H,88H,87H,87H,87H,87H,86H,86H,86H,85H
DFB      85H,85H,84H,84H,84H,83H,83H,83H,82H,82H
DFB      82H,81H,81H,81H,80H,80H,79H,79H,79H,78H
DFB      78H,77H,77H,77H,76H,76H,75H,75H,75H,74H
DFB      74H,74H,73H,73H,72H,72H,71H,71H,70H,70H
DFB      70H,69H,69H,68H,68H,67H,67H,67H,66H,65H
DFB      65H,65H,64H,64H,63H,63H,62H,62H,61H,61H
DFB      60H,60H,59H,59H,58H,58H,57H,57H,56H,56H
DFB      55H,55H,54H,54H,53H,53H,52H,51H,51H,50H
DFB      50H,50H,49H,48H,48H,47H,47H,46H,46H,45H
DFB      45H,44H,44H,43H,42H,42H,41H,41H,40H,40H
DFB      39H,38H,38H,37H,37H,36H,35H,35H,34H,34H
DFB      33H,32H,32H,31H,31H,30H,30H,29H,28H,28H
DFB      27H,27H,26H,25H,25H,24H,24H,23H,22H,22H
DFB      21H,21H,20H,20H,19H,18H,18H,17H,17H,16H
DFB      15H,15H,14H,14H,13H,12H,12H,11H,11H,10H
DFB      09H,08H,08H,07H,07H,06H,06H,05H,05H,04H
DFB      03H,03H,02H,02H,01H,00H

```

; **** SET UP DISPLAY BUFFER ****

```

SETBUF:  MOV      R0,#DISB
          MOV      R3,#10H
BLOCK:   MOV      A,R2
          MOVP     A,@A
          MOV      @R0,A
          INC      R2
          INC      R0
          DJNZ    R3,BLOCK
          RET

```

```

TABLE2:    DFB    "PF. = 0.00  "
TABLE3:    DFB    "  IN PHASE  "
; ***** CHANGE LEAD OR LAG *****
CHKL:      MOV    R0,#43
           JBO    CHKL1
           MOV    R2,#44H    ;FOR LEAD
           JMP    CHKL2
CHKL1:     MOV    R2,#40H    ;FOR LAG
CHKL2:     MOV    R3,#04H
CHKL3:     MOV    A,R2
           MOVP   A,@A
           MOV    @R0,A
           INC    R2
           INC    R0
           DJNZ   R3,CHKL3
           RET

TABLE4:    DFB    "LAG "
TABLE5:    DFB    "LEAD"
; ***** FOR CHANGE DATA FROM ADC TO LCD *****
CHNDATA:   ANL    A,#0FH
           ORL    A,#30H
           MOV    R0,#41
           MOV    @R0,A
           MOV    R1,#DATAB
           MOV    A,@R1    ;@R1 IS STORE DATA FROM ADC
           SWAP   A
           ANL    A,#0FH
           ORL    A,#30H
           DEC    R0
           MOV    @R0,A
           RET

; ***** DELAY SUBROUTINE *****
DELAY:     MOV    R2,#00H    ;A=3 EQU 2MS
DE2:       DJNZ   R2,DE2
           DEC    A
           JNZ    DELAY
           RET
           END

```

บทที่ 4

การทดลองและผลการทดลอง

ในการทดลองจะมีปัญหาว่าจะทำการแยกไฟเลี้ยงสำหรับ ส่วนนอกและส่วน ดิจิตอล เพราะอาจมีปัญหาเนื่องจากสัญญาณรบกวน และการปรับค่าความต้านทานที่ปรับค่า ได้เพื่อให้ได้ค่าเพาเวอร์แฟกเตอร์ที่ใกล้เคียงกับกับค่าจริง และมีปัญหาในส่วนการแสดงผล การนำหน้า/ตามหลัง เนื่องจากผลของวงจรมีค่าคงที่ซึ่งอาจเกิดเพราะการที่วงจรมีการหน่วง เวลาไม่เท่ากันในส่วนของแต่ละส่วน

ผลการทดลอง

ตารางผลการทดลองวัดค่าเพาเวอร์แฟกเตอร์จากเครื่องใช้ไฟฟ้าต่างๆ โดยเครื่องวัด ค่าเพาเวอร์แฟกเตอร์แบบดิจิตอล

เครื่องใช้ไฟฟ้าที่วัด	ค่าเพาเวอร์แฟกเตอร์ที่วัดได้
พัดลมจากคอมพิวเตอร์	0.67
คอมพิวเตอร์	0.98
หลอดฟลูออเรสเซนต์	0.35
พัดลมเบอร์ 1	0.85
พัดลมเบอร์ 2	0.95
พัดลมเบอร์ 3	0.96

เนื่องจากไม่สามารถตรวจสอบค่าที่วัดได้จากเครื่องวัดค่าเพาเวอร์แฟกเตอร์กับค่าจาก เครื่องวัดอื่นๆ ที่เป็นมาตรฐาน ดังนั้นจึงไม่สามารถบอกถึงประสิทธิภาพของเครื่องมือนี้ได้ แต่ ค่าที่วัดได้จากหลอดฟลูออเรสเซนต์นั้นมีค่าที่ตรงกับค่าที่บอกไว้ที่ฉลาก และค่าที่วัดได้จาก เครื่องใช้ไฟฟ้าอื่นๆ ก็มีค่าเพาเวอร์แฟกเตอร์ใกล้เคียงกับค่าโดยทั่วไป

บทที่ ๕ บทวิจารณ์และสรุป

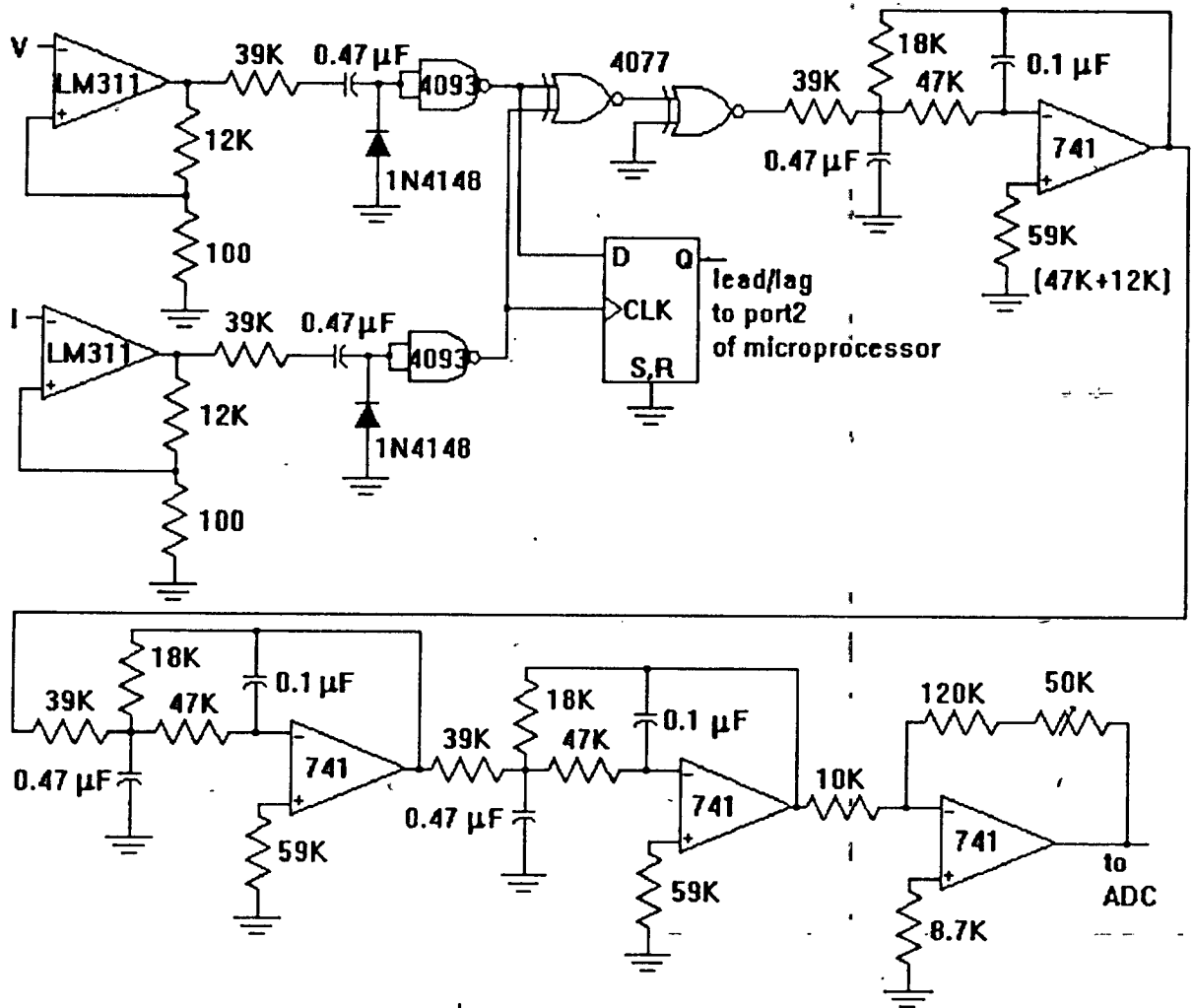
เครื่องวัดค่าเพาเวอร์แฟกเตอร์ที่ได้เป็นเครื่องวัดที่ออกแบบแบบง่ายๆ ราคาอุปกรณ์ที่นำมาต่อประกอบราคาถูก เหมาะสมกับการใช้งานได้ง่ายผลที่ได้จากเครื่องวัดมีความถูกต้องแม่นยำใกล้เคียงกับความเป็นจริงมากโดยจะมีค่าผิดพลาดไปบ้างก็อาจจะเนื่องจากการคำนวณหรือการแปลงค่าต่าง ๆ ภายในวงจร โดยที่วงจรนี้ออกแบบโดยไม่ขึ้นอยู่กับแอมพลิจูด (amplitude) ของสัญญาณที่เข้ามา แต่จะเทียบการเลื่อนห่างของเฟสที่แตกต่างกันเพียงอย่างเดียว

เนื่องจากวงจรซีโรครอสซิงดิเทกเตอร์นั้นมีความแรงดันไฟฟ้าฮิสทีรีซิสที่ทำให้มีความผิดพลาดของการวัดเนื่องจากวงจรทั้ง 2 วงจรนั้นอาจมีความแรงดันไฟฟ้าฮิสทีรีซิสไม่เท่ากันซึ่งอาจเป็นค่าของความต้านทานหรือการที่ไอซีที่ใช้มีคุณสมบัติที่ไม่เหมือนกัน

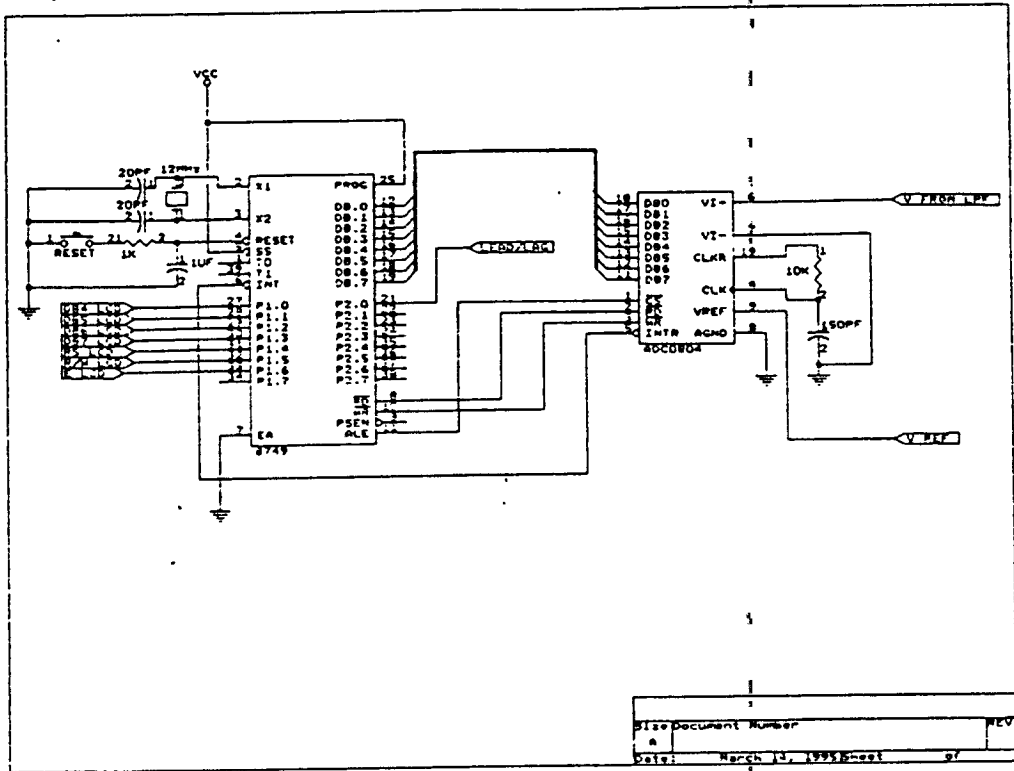
ในการทดลองนี้ ตอนแรกการเขียนโปรแกรมนั้นคณะผู้จัดทำตั้งใจจะใช้การคำนวณค่าเพาเวอร์แฟกเตอร์โดยรับค่าจากวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล โดยการคำนวณค่า $\cos(\phi)$ แทนการใช้เปิดตารางจากค่าเพาเวอร์ที่ได้คำนวณไว้แล้ว แต่ไม่สามารถที่จะเขียนโปรแกรมดังกล่าวได้ จึงใช้วิธีการเปิดตารางแทนซึ่งการที่คำนวณไว้แล้วนั้นจะมีค่าผิดพลาดเนื่องจากการปัดค่าต่างๆ ทำให้ค่าที่ได้นั้นผิดพลาดได้ สำหรับผู้ที่สนใจจะทำโครงการนี้ควรจะต้องปรับปรุงการเขียนโปรแกรมนี้ด้วย และแก้ไขข้อบกพร่องเนื่องจากค่าแรงดันไฟฟ้าฮิสทีรีซิสของวงจรซีโรครอสซิงดิเทกเตอร์ที่ทำให้เกิดค่าที่ผิดพลาดได้

ถ้ามีข้อผิดพลาดประการใดในการเขียนปริญาณิพนธ์ครั้งนี้ทางคณะผู้จัดทำต้องขออภัยไว้ ณ ที่นี้ด้วย

ภาคผนวก



วงจรเครื่องวัดค่าเพาเวอร์แฟกเตอร์



กิตติกรรมประกาศ

ขอขอบคุณผู้ที่มีรายนามดังต่อไปนี้

1. ศ.ดร.วัลลภ สุระกำพลธร อาจารย์ที่ปรึกษา
2. นายสุทธิ จรรยาสุทธิวงศ์ นักศึกษาภาควิชาอิเล็กทรอนิกส์
3. นายศักดิ์สิทธิ์ สุขภิบาล นักศึกษาภาควิชาอิเล็กทรอนิกส์
4. นายสุทธิชัย ทองผา นักศึกษาภาควิชาอิเล็กทรอนิกส์
5. นายสุเทพ เตชะบุญเกียรติ นักศึกษาภาควิชาอิเล็กทรอนิกส์
6. น.ส.สุธิญา บุษบรราช นักศึกษาภาควิชาโทรคมนาคม

บรรณานุกรม

- ชเนส คิริไตรวัฒนาพร . 'การวัดและเครื่องมือวัด ตอน เพาเวอร์แฟกเตอร์มิเตอร์' .
เซมิคอนดักเตอร์อิเล็กทรอนิกส์ . 2536 . ฉบับที่ 130 . หน้า 74-76
- เศกสิทธิ์ คำชมภู . 'สื่อยอดวัตต์มิเตอร์ ตอน 1 การทำงานของวงจร' .
เซมิคอนดักเตอร์อิเล็กทรอนิกส์ . 2537 . ฉบับที่ 135 . หน้า 52-57
- เศกสิทธิ์ คำชมภู . 'สื่อยอดวัตต์มิเตอร์ ตอนจบ การสร้างและการใช้งาน' .
เซมิคอนดักเตอร์อิเล็กทรอนิกส์ . 2537 . ฉบับที่ 136 . หน้า 36-45
- ชูชัย ธารสารตั้งเจริญ . ทฤษฎีวงจรอิเล็กทรอนิกส์ . สำนักพิมพ์ฟิลิกส์เซ็นเตอร์ .
กรุงเทพฯ
- พิพัฒน์ เลาทสงคราม . ไมโครคอนโทรลเลอร์ . สถาบันเทคโนโลยีพระจอมเกล้า
เจ้าคุณทหารลาดกระบัง
- SILA RESEARCH CO,LTD. , "*User's manual Dot Matrix LCD
Module*"
- Coughlin,Robert F. and Driscoll,Frederick F. , "*Operational
Amplifier and Linear Integrated Circuits*" , Prentice- Hall
Inc., Englewood Cliffs , N. J.
- Barna,Arpad and Porat,Dan I. , "*Operational Amplifiers*" ,
John Wiley & Sons Inc. , New York , 1989
- T.K.M Babu and Wong,Tak Wing , 1989 , "*Novel low-cost
Electronic Power Factor Meter*" , International Journal of
Electronics , 1989 , Vol. 67, No.1 , pp. 147-151
- M.S.M Ani,AI and M.A.H Karim,Abdul , "*Digital power factor
meter design based on voltage-to -frequency conversion*" ,
International Journal of Electronecs , 1982 , Vol.52 , No.5 ; -
pp.463-470
- Hafeth,Basim A. and M.A.H Rakim,Abdul , "*Digital power factor
meter based on non-linear analogue-todigital
conversion*" , International Journal of Electronics , 1985 ,
Vol.58 , No.3 , pp.513-519
- Irvine,Robert G. , "*Operational Amplifier Characteristics and
Applications*" , Prentice-Hall Inc. , Englewood Cliffs , N.J ,
1987