สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

เครื่องเคาะจังหวะสำหรับนักดนตรี

นาย ธนภัทร ฉายากุล
นางสาว สุมณี วิวัฒนธีรกุล

วศ.
ว162ค
เลขหมู่ 2537
เลขทะเบียน
วัน,เดือน,ปี
6125548 68

โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บัณฑิต
ภาควิชาฟิสิกส์ประยุกต์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2537

Digital Metronome

Mr. Thanapat Chayakul
Miss Sumanee Viwattanateerakul

A Special Project Submitted in Partial Fulfillment of the
Requirement for the Degree of Bachelor of Science
Department of Applied Physics
Faculty of Science
King Mongkut's Institute of Technology Ladkrabang
1994

หัวข้อโครงงานพิเศษ	เครื่องเคาะจังหวะสำหรับนักดนตรี
โดย	นาย ธนภัทร ฉายากุล
	นางสาว สุมณี วิวัฒนธีรกุล
ภาควิชา	ฟิสิกส์ประยุกต์
อาจารย์ที่ปรึกษา	ผศ.ดร. วราวุฒิ เถาลัดดา
	ผศ.ดร. ปรีชา เทียนสมประสงค์

ภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
อนุมัติให้นับโครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตร์บันฑิต
ลายมือชื่อ

_____ หัวหน้าภาค
(ผศ.ดร. ปรีชา เทียนสมประสงค์)

คณะกรรมการโครงงานพิเศษ

_____ ประธานกรรมการ
(อ. สุน จ่างประยูร)

_____ กรรมการ
(ผศ.ดร. วราวุฒิ เถาลัดดา)

_____ กรรมการ
(ผศ.ดร. ปรีชา ยุพาพิน)

_____ กรรมการ
(อ. สุทิน เมฆไกรศรี)

ลิขสิทธิ์ของภาควิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

หัวข้อโครงงานพิเศษ	เครื่องเคาะจังหวะสำหรับนักดนตรี
นักศึกษา	นาย ธนภัทร ฉายากุล
	นางสาว สุมณี วิวัฒนธีรกุล
อาจารย์ที่ปรึกษา	ผศ.ดร. วราวุฒิ เถาลัดดา
	ผศ.ดร. ปรีชา เทียนสมประสงค์
ภาควิชา	ฟิสิกส์ประยุกต์
ปีการศึกษา	2537

บทคัดย่อ

โครงงานพิเศษนี้เป็นการศึกษาขั้นตอนการออกแบบเครื่องเคาะจังหวะสำหรับนักดนตรีมีใช้อยู่อย่างแพร่หลายในหมู่นักดนตรีทั้งมืออาชีพ และมือสมัครเล่น

เครื่อเคาะจังหวะสามารถควบคุมความดังของเสียงเคาะ ความเร็วของการเคาะ หรือแม้แต่สั่งปิดเครื่อง ด้วยการสั่งงานจากปุ่มที่หน้าปัทม์ โดยที่ทำเช่นนี้ได้ต้องออกแบบฮาร์ดแวร์ และซอฟท์แวร์ให้สัมพันธ์กันเป็นหนึ่งเดียว โดยฮาร์ดแวร์ที่เลือกใช้เป็นหน่วยควบคุมขนาดเล็กของบริษัทอินเทลในตระกูล MCS®-51 เป็นหัวใจหลัก มีสัญญาณความถี่ฐานเวลา 60 เฮิร์ทซไว้สำหรับใช้ในการนับเวลา และสัญญาณความถี่ฐานเวลาที่ควบคุมได้ด้วยซอฟท์แวร์ ไว้สำหรับให้จังหวะการเคาะ และติดต่อกับผู้ใช้ผ่านทางลำโพง ปุ่มกด ตัวเลข 3 หลัก และจอภาพแบบผลึกเหลวขนาด 20 ตัวอักษร 2 บรรทัด การทำงานทั้งหมดสั่งงานด้วยภาษาแอสเซมบลี โดยสามารถเปลี่ยนความดังของเสียงเคาะ ความเร็วของการเคาะ รูปแบบของการเคาะ และอัตราส่วนของตัวโน๊ตในหนึ่งห้องได้ด้วยการกดปุ่ม อีกทั้งยังสามารถกำหนดจำนวนห้องที่จะให้เคาะหรือจำนวนเวลาที่จะให้เคาะได้ด้วย โดยข้อมูลเหล่านี้จะไม่สูญหายไปหลังจากปิดเครื่อง

Special Project Title	Digital Metronome
Name	Mr. Thanapat Chayakul
	Miss. Sumanee Viwattanateerakul
Special Project Advisor	Mr. Varawuth Thaoludda
	Mr. Preecha Teansomprasong
Department	Applied Physics
Academic Year	1994

Abstract

This senior project is the product design step because the Metronome is widely used in professional and amateur musician group.

The system is controlled by front button such as to change speaker volume, tempo or to shutdown. All functions can work properly by unite designed hardware and software. The Intel MCS®-51, the main microcontroller unit, will receive 60 Hz base frequency for time count and programmable base frequency for tempo. The user interface units are speaker, keypad, 3-digit numerical display and 20 character 2 lines liquid crystal display module. All functions controlled by MCS®-51 assembly language are to change speaker volume, tempo, meter and time signature by front key and programmable measure count or time count. All programmable data will reserve forever after shutdown.

กิตติกรรมประกาศ

ขอขอบคุณ

อ. วรารุฒิ เถาลัดดา และ อ. ปรีชา เทียนสมประสงค์ ที่ให้โอกาสทำโครงงาน

อ. ศิริศักดิ์ เตชะทวีกุล ให้คำแนะนำเรื่องระเบียบการทำโครงงาน

อ. วิชาญ กกกนทา เอื้อเฟื้อเอกสารข้อมูล

พี่เล็ก สุรเมษ ให้คำแนะนำ สนับสนุนอุปกรณ์และเอกสารข้อมูลมากมาย

พี่คริต ให้คำแนะนำเรื่อง DAC และ VCO

พี่อ้อ ศิลารีเสริช สละเวลาตรวจสอบวงจร

คุณ พฤฒิพันธ์ ฉายากุล ดูแลเรื่องรูปประกอบ

คุณ ปกรณ์ พฤกษาพงษ์ ช่วยพิมพ์เนื้อหา

เพื่อนๆ ทุกคนที่ช่วยให้กำลังกายและกำลังใจจนงานนี้สำเร็จด้วยดี

สารบัญเนื้อเรื่อง

หน้า

บทคัดย่อ

Abstract

กิตติกรรมประกาศ

สารบัญเนื้อเรื่อง

สารบัญรูป

1 บทนำ 1

1.1 วัตถุประสงค์ 1

1.2 ลักษณะโครงสร้างของระบบ 1

1.3 ขอบข่ายการทำงานของเครื่องเคาะจังหวะ 2

2 ความรู้พื้นฐานเรื่องเวลาทางดนตรี 3

2.1 จังหวะ (Beat) 3

2.2 ความเร็วของจังหวะ (Tempo) 3

2.3 ตัวแสดงความเร็ว 4

2.3.1 คำที่ใช้อธิบาย 4

2.3.2 ตัวแสดงเมโทรโนม (Metronome Marking) 5

2.4 การเร่งหรือการลดความเร็วของเพลง 5

2.5 รูปแบบของจังหวะ 6

2.5.1 จังหวะคู่ 6

2.5.2 จังหวะสาม 6

2.5.3 จังหวะสี่ 6

2.6 ลีลาของจังหวะ 6

2.7 การแบ่ง และการรวมเวลา 6

2.8 ตัวโน๊ตทางดนตรี 7

2.9 โน๊ตสามพยางค์ 7

2.10 ตัวประจุด 8

2.11 ความสัมพันธ์ระหว่างลีลาของจังหวะ และความเร็วของจังหวะ 8

2.12 ตัวหยุด 9

2.13 อัตราส่วนของตัวโน๊ต และรูปแบบของจังหวะ 9

2.14 อัตราส่วนของตัวโน๊ต 9

2.15 ประเภทของรูปแบบของจังหวะ 10

2.15.1 รูปแบบอย่างง่าย 10
2.15.2 รูปแบบรวม 10
2.15.3 รูปแบบไม่สมมาตร 10
2.15.4 รูปแบบผสม 11
3 ฮาร์ดแวร์ของระบบ 12
3.1 ฮาร์ดแวร์ของ 8051 12
3.2 วงจรกำเนิดความถี่นาฬิกาบน 8051 16
3.3 หน่วยความจำภายนอก 16
3.3.1 การต่อหน่วยความจำภายนอก 17
3.4 วงจรตั้งเวลา/วงจรนับ 18
3.4.1 รีจิสเตอร์ควบคุมการทำงานของวงจรตั้งเวลา/วงจรนับ 19
3.4.1.1 Timer Control (TCON) 19
3.4.1.2 Timer Mode Control (TMOD) 20
3.4.2 การอินเตอร์รัพท์ของวงจรตั้งเวลา/วงจรนับ 20
3.4.3 โหมดการทำงานของวงจรตั้งเวลา/วงจรนับ 21
3.4.3.1 โหมด 0 21
3.4.3.2 โหมด 1 21
3.4.3.3 โหมด 2 22
3.4.3.4 โหมด 3 22
3.4.4 การนับ 22
3.5 การควบคุมอินเตอร์รัพท์ 22
3.5.1 รีจิสเตอร์ควบคุมการอินเตอร์รัพธ์ 23
3.5.1.1 Interrupt Enable (IE) 23
3.6 ระบบควบคุม 24
3.7 ส่วนติดต่อกับผู้ใช้ 26
3.7.1 การต่อเข้าระบบ 26
3.7.2 ส่วนเปล่งเสียง 26
3.7.3 การสร้างสัญญาณเสียง 26
3.7.4 ตัวเลขแสดงความเร็วของจังหวะ 27
3.7.5 ปุ่มกด 27
3.8 ความถี่ฐานเวลา 29
3.8.1 Voltage Control Oscillator (VCO) 29
3.8.2 การควบคุมความถี่ 30

3.8.3 การนำความถี่ไปใช้ 32
3.9 การทำงานนับเวลา 32
3.9.1 การนำความถี่ไปใช้ 32
3.10 ส่วนจ่ายพลังงานควบคุมได้ 33
3.10.1 การควบคุม 33
4 การใช้งานเครื่องและซอฟแวร์ 35
4.1 การใช้งานเครื่อง 35
4.2 ซอฟแวร์ 41
4.2.1 โปรแกรมเปิดเครื่อง 41
4.2.2 โปรแกรมตั้งค่าเมนู 41
4.2.2.1 โปรแกรมตั้งค่า Meter 41
4.2.2.2 โปรแกรมตั้งค่า Time Signature 42
4.2.2.3 โปรแกรมตั้งค่า Measure Count 42
4.2.2.4 โปรแกรมตั้งค่า Time Count 42
4.2.2.5 โปรแกรมตั้งค่า Volume 42
4.2.2.6 โปรแกรมตั้งค่า tempo 43
4.2.3 โปรแกรมแสดงผลทาง LCD และ 7-SEG 43
4.2.3.1 โปรแกรมแสดงผลที่ LCD 43
4.2.3.2 โปรแกรมแสดงผลที่ 7-SEG 43
4.2.4 โปรแกรมเคาะ 43
4.2.5 โปรแกรมสั่งหยุดขณะที่กำลังเคาะอยู่ 44
4.2.6 โปรแกรมปิดเครื่อง 44
5 สรุปการทำงาน 72
5.1 การปรับปรุง 72
หนังสืออ้างอิง 73
ภาคผนวก ก ฮาร์ดแวร์ของระบบ 74
ภาคผนวก ข ซอฟท์แวร์ของระบบ 79
ภาคผนวก ค การใช้งาน LCD Module 104
ภาคผนวก ง Data Sheet 111

สารบัญรูป

หน้า

รูป 1.1 ระบบรวมทั้งหมดของเครื่องเคาะจังหวะ 2
รูป 2.1 ตัวโน๊ตที่ใช้กันในปัจจุบัน 7
รูป 2.2 ความยาวของตัวโน๊ต 7
รูป 2.3 โน๊ตสามพยางค์ 8
รูป 2.4 ตัวหยุดที่ใช้กันในปัจจุบัน 9
รูป 2.5 การแสดงจังหวะในหนึ่งห้อง 9
รูป 3.1 สถาปัตยกรรมของ 8051 13
รูป 3.2 โครงสร้างการเขียนโปรแกรม 14
รูป 3.3 การจัดขาของ 8051 15
รูป 3.4 แมชชีนไซเคิล 16
รูป 3.5 แผนผังเวลาของระบบ 17
รูป 3.6 ลอจิกควบคุมการทำงานของวงจรตั้งเวลา/วงจรนับ 18
รูป 3.7 โหมดการทำงานของวงจรตั้งเวลา/วงจรนับ 21
รูป 3.8 วงจรส่วนควบคุมขนาดเล็ก 25
รูป 3.9 วงจรส่วนติดต่อกับผู้ใช้ 28
รูป 3.10 ส่วนประกอบของ LM331 29
รูป 3.11 วงจรแปลงข้อมูลดิจิตอลเป็นสัญญาณอนาลอก 31
รูป 3.12 วงจรแปลงกระแสเป็นแรงดัน และ buffer 31
รูป 3.13 วงจรกำเนิดความถี่ฐานเวลา 60 Hz 32
รูป 3.14 วงจรควบคุมแหล่งจ่ายไฟให้ระบบทั้งหมด 34
รูป ก.1 หน่วยควบคุมขนาดเล็ก 75
รูป ก.2 หน่วยติดต่อกับผู้ใช้ 76
รูป ก.3 หน่วยความถี่ฐาน 77
รูป ก.4 หน่วยจ่ายพลังงานและส่วนรีเซ็ตเมื่อเปิดครื่อง 78

บทที่ 1
บทนำ

ดนตรีเป็นภาษาสากลที่ทั่วโลกยอมรับ ดนตรีที่ดีมีอิทธิพลทำให้ธรรมชาติกลับเป็นดังธรรมชาติเช่นเดิม ดนตรีที่ดีมีอิทธิพลเหนือผู้คน คนที่ซาบซึ้งในดนตรีเป็นผู้ที่มีจิตใจละเอียดอ่อน ดังพระราชนิพนธ์ในพระบาทสมเด็จพระมงกุฎเกล้าเจ้าอยู่หัวที่ว่า "ชนใดไม่มีดนตรีกาล ในสันดานเป็นคนชอบกลนัก"

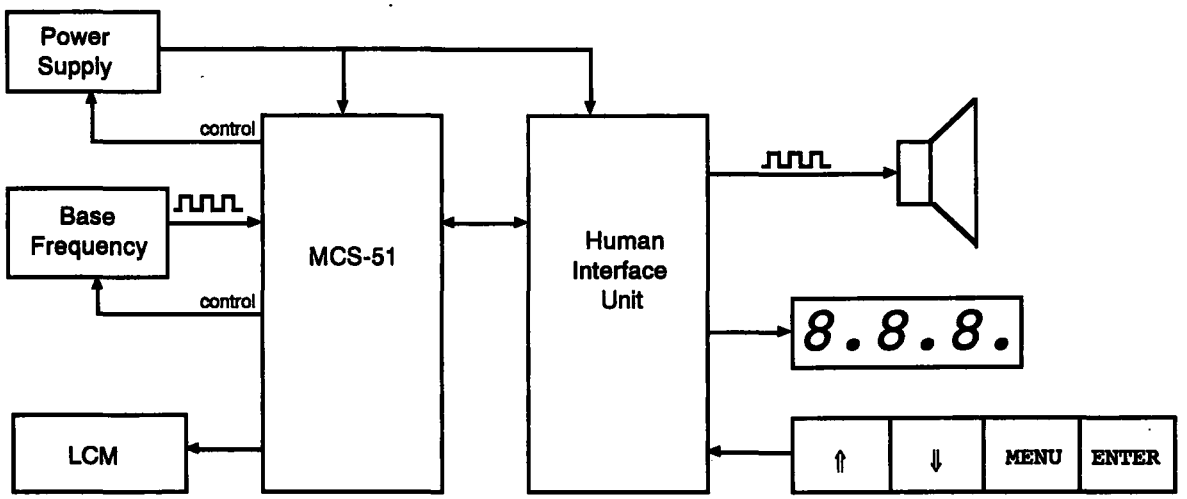
ตัวอย่างของผู้ที่ซาบซึ้งในดนตรีคือนักดนตรี นักดนตรีเป็นผู้ที่ขวนขวายใฝ่ที่จะเข้าถึงแก่นแท้ของดนตรี โดยมีความมุ่งหมายว่าดนตรีคือศิลปะ และดนตรีไม่เคยทำร้ายใคร นักดนตรีไม่ว่าจะเล่นเครื่องดนตรีอะไรย่อมต้องเข้าใจ และซาบซึ้งในเวลาของดนตรีหรือจังหวะนั่นเอง นักดนตรีที่ทำงานร่วมกันเป็นกลุ่มย่อมเลือกใช้เวลาทางดนตรีเดียวกัน สิ่งหนึ่งที่ช่วยในการแสดงเวลาทางดนตรีนั่นคือ เครื่องเคาะจังหวะสำหรับนักดนตรี (Metronome) ดังนั้น นักดนตรีทุกคนต้องรู้จักเครื่องเคาะจังหวะ และใช้ในการหัดเล่น ฝึกฝน จนกระทั่ง ใช้ในการเล่นร่วมกันหรือในการบันทึกเสียงในห้องอัด

1.1 วัตถุประสงค์

เครื่องเคาะจังหวะที่มีขายอยู่ตามท้องตลาดในปัจจุบัน แต่ละแบบจะมีข้อดีแตกต่างออกไป โครงงานพิเศษนี้จึงมีแนวคิดที่จะพัฒนาเครื่องต้นแบบที่รวมการทำงานหลายๆ อย่างที่นักดนตรีต้องการเข้าไว้ในเครื่องเคาะจังหวะสำหรับนักดนตรีนี้ โดยการได้ศึกษาเรื่องราวของไมโครโปรเซสเซอร์และการนำมาใช้งาน และจากความเข้าใจในธรรมชาติของเวลา ธรรมชาติของเสียง ยิ่งทำให้มีความคิดที่จะใช้ความรู้ทั้งหมดสร้างสรรงานที่มีประโยชน์แก่มวลชนดังกล่าว

1.2 ลักษณะโครงสร้างของระบบ

ระบบทั้งหมดถูกออกแบบให้ส่วนของซอฟท์แวร์ และฮาร์ดแวร์ประสานรวมกันเป็นหนึ่งเดียวดังรูป 1.1



รูป 1.1 ระบบรวมทั้งหมดของเครื่องเคาะจังหวะ

แต่ละส่วนถูกสั่งงานด้วยหน่วยควบคุมขนาดเล็กตระกูล MCS®-51 แม้กระทั่งแหล่งจ่ายพลังงานให้ระบบทั้งหมด

ระบบทั้งหมดประกอบด้วย

1. หน่วยควบคุมขนาดเล็ก
2. หน่วยติดต่อกับผู้ใช้
3. หน่วยความถี่ฐาน
4. หน่วยจ่ายพลังงาน

เมื่อมีซอฟแวร์สั่งงานที่ดีแล้วระบบทั้งหมดจะทำงานได้ตามวัตถุประสงค์

1.3 ขอบข่ายการทำงานของเครื่องเคาะจังหวะ

การทำงานของเครื่องทั้งหมดถูกควบคุมด้วยซอฟต์แวร์

1. สามารถเลือกความเร็วของโน๊ตตัวดำ (Quarter Note) ได้ตั้งแต่ 0-255
2. สามารถเลือกรูปแบบการเคาะได้ 4 แบบคือ โน๊ตตัวดำ (Quarter Note), โน๊ตตัวเขบ็ต 1 ชั้น (Eighth Note), โน๊ตสามพยางค์ (Triplets) และโน๊ตตัวเขบิต 2 ชั้น (Sixteenth Note)
3. สามารถกำหนดจำนวนห้อง (Measure) ของการเคาะหรือบอกจำนวนห้องที่เคาะไปแล้วได้
4. สามารถตั้งเวลาในการเคาะหรือบอกเวลาที่เคาะไปแล้วได้
5. สามารถเปลี่ยนความดังของเสียงเคาะได้ด้วยซอฟท์แวร์
6. สามารถเก็บค่าที่ตั้งไว้จากข้างบนไม่ให้สูญหายหลังปิดเครื่อง

บทที่ 2
ความรู้พื้นฐานเรื่องเวลาทางดนตรี

2.1 จังหวะ (Beat)

เวลาในปัจจุบันมีหน่วยที่เล็กที่สุดคือวินาที วินาทีจึงกลายเป็นหน่วยพื้นฐาน นั่นคือแต่ละวินาทีจะมีความยาวเท่ากับวินาทีก่อนหน้านี้ และวินาทีต่อไป วินาทีเป็นช่วงเวลาที่สัมบูรณ์ และไม่เปลี่ยนแปลง วินาทีจะเดินหน้าไปด้วยความเร็วคงที่ และจะรวมกันเป็นจำนวนที่แน่นอน กลายเป็นนาที

ในทำนองเดียวกัน จังหวะทางดนตรีก็เหมือนกับวินาทีของนาฬิกา จังหวะเป็นหน่วยที่เล็กที่สุดในการวัดเวลาทางดนตรี จังหวะจึงกลายเป็นหน่วยพื้นฐาน แต่ละจังหวะจะมีความยาวเท่ากับจังหวะก่อนหน้านี้ และจังหวะต่อไป แต่อย่างไรก็ดี จังหวะไม่ใช่จำนวนที่สัมบูรณ์ และไม่เปลี่ยนแปลง จังหวะจะไม่เดินหน้าไปด้วยความเร็วคงที่ตลอดเสมอไป จึงทำให้เวลาทางดนตรีแตกต่างจากเวลาของนาฬิกา วินาทีเป็นค่าคงที่เปลี่ยนแปลงไม่ได้ แต่ความเร็วของจังหวะสามารถเปลี่ยนแปลงได้โดยนักดนตรี ในชีวิตประจำวัน เราไม่สามารถเปลี่ยนแปลงช่วงเวลาวินาทีที่เดินหน้าไปอย่างคงที่ แต่ในทางดนตรี นักประพันธ์สามารถเร่งหรือยืดความเร็วของเพลงโดยเปลี่ยนความเร็วของจังหวะ ดังนั้นดนตรีจึงสามารถสร้างความประทับใจได้โดยการหลุดออกจากความเป็นจริงของเวลาที่เดินทางคงที่ต่อไป

จังหวะคือสิ่งที่ง่ายที่สุดของดนตรีที่ผู้ฟังสามารถเข้าถึง และซาบซึ้งได้ เรารับรู้จังหวะที่คงที่ทางกายภาพจากการเคลื่อนไหวอย่างสม่ำเสมอ เช่นการตบเท้า การปรบมือ และการเต้นรำ จังหวะการเคลื่อนไหวของนักเต้น จะสัมพันธ์กับจังหวะดนตรี เรามองเห็นการให้จังหวะได้ชัดเจนจากการสังเกตการตีกลองในเพลงร็อค เรารับรู้จังหวะเหมือนเป็นเหตุการณ์ที่พาเราเคลื่อนที่ไปผ่านเวลาด้วยความเร็วคงที่ บางครั้งจังหวะก็เป็นสิ่งลึกลับ เมื่อสามารถสร้างความรู้สึกให้ล่องลอยผ่านช่วงเวลา การที่ความรู้สึกเคลื่อนที่ไปในเวลานั้นถูกควบคุมโดยจังหวะนั่นเอง

2.2 ความเร็วของจังหวะ (Tempo)

เรามักจะให้จังหวะที่เร็วหมายถึงความตื่นเต้น หัวใจของเราจะเต้นเร็วขึ้นเมื่อเข้าถึงตัวดนตรี และจังหวะที่ช้าจะหมายถึงอารมณ์ที่สงบเยือกเย็น เราจะรู้สึกว่าร่างกาย และจิตใจของเราผ่อนคลายเมื่อเข้าถึงตัวดนตรี ตัวดนตรีเองไม่จำเป็นจะต้องมีจังหวะเพียงจังหวะเดียวหรือเพียงอารมณ์เดียว ผู้ประพันธ์เพลงสามารถเพิ่มหรือลดความรู้สึกเร่งเร้า ได้โดยการเปลี่ยนจังหวะภายในเพลงเดียวกัน

2.3 ตัวแสดงความเร็ว

ตัวแสดงความเร็วจะปรากฏอยู่ที่โน๊ตดนตรีเพื่อบอกให้นักดนตรีทราบความเร็วของเพลงที่จะเล่น ปกติแล้วตัวแสดงความเร็วจะเป็นภาษาอิตาเลียน แต่ความเร็วสามารถกำหนดให้แน่นอนลงไปอีกโดยใช้ตัวแสดงเมโทรโนม (Metronome Marking) แต่อย่างไรก็ตามไม่มีการกำหนดตายตัวว่าให้ใช้แบบไหน แต่ที่สำคัญจะต้องแสดงถึงความเร็วในระดับเดียวกัน เพื่อที่นักดนตรีจะสามารถเล่นเพลงที่ความเร็วที่ผู้ประพันธ์กำหนดไว้

2.3.1 คำที่ใช้อธิบาย

ครั้งหนึ่งภาษาลาตินเป็นภาษานานาชาติที่ใช้ในการสื่อสารระหว่างผู้คนต่างภาษา และวัฒนธรรม เช่นกันภาษาอิตาเลียนก็เป็นภาษากลางที่ใช้อธิบายดนตรี นักดนตรีทั่วโลกไม่จำเป็นที่จะสามารถทักทายตอนเช้าเป็นภาษาอิตาเลียน แต่ทั่วโลกใช้ภาษาอิตาเลียนในวงการดนตรี โดยสามารถสื่อสารกับผู้คนที่มีภาษาของตนเอง รายชื่อข้างล่างนี้เป็นช่วงของความเร็วอย่างกว้างๆ แต่ความเร็วนั้นเป็นความเร็วโดยประมาณ ตัวชื่อของความเร็วเองมีความหมายในตัวเองที่นักดนตรีแต่ละคนเข้าใจอยู่แล้ว

ภาษาอิตาเลียน	แปลเป็นไทย	ตัวแสดงเมโทรโนม
largo	ช้ามาก	40-60
larghetto	ช้ามาก แต่เร็วกว่า largo	
grave	ช้ามาก ท่าทางมั่นคง	
lento	ช้า	60-66
adagio	ช้า สบายๆ	66-76
andante	ช้าพอดีพอดี (ก้าวเท้าเดิน)	76-108
moderato	ปานกลาง	108-120
allegretto	เร็ว	
allegro	เร็วสนุกสนาน	120-168
vivace	มีชีวิตชีวา	
presto	เร็วมาก	168-200
prestissimo	เร็วที่สุดที่จะทำได้	200-208

คำภาษาอิตาเลียนอื่นที่ใช้เชื่อมคำข้างบนนี้เพื่อบอกรายละเอียดมากขึ้น ดังนี้

ภาษาอิตาเลียน	แปลเป็นภาษาอังกฤษ	แปลเป็นภาษาไทย
assai	very	มาก (วิเศษณ์)
meno	less	น้อย (คุณศัพท์)
molto	much	มาก (คุณศัพท์)
non troppo	not too much	ไม่มากนัก

ภาษาอิตาเลียน	แปลเป็นภาษาอังกฤษ	แปลเป็นภาษาไทย
piu	more	มากกว่า
poco a poco	little by little	ทีละน้อยทีละน้อย

ดังนั้น คำว่า *allegro assai* แปลว่า "เร็วมาก" และ *allegro non troppo* แปลว่า "ไม่เร็วมากนัก"

คำเหล่านี้แสดงความเร็วของจังหวะ และสามารถบอกถึงลักษณะของอารมณ์ ที่ไม่เพียงบอกเราว่าเพลงนี้เร็วแค่ไหน แต่ยังสามารถบอกความรู้สึกของเพลงดังกล่าวได้ด้วย

2.3.2 ตัวแสดงเมโทรโนม (Metronome Marking)

ความเร็วสามารถบอกได้ด้วย ตัวแสดงเมโทรโนม เพื่อแสดงถึงจำนวนจังหวะที่แน่นอนในหนึ่งนาที เช่น ตัวแสดงเมโทรโนม ♩ = 60 หมายความว่าจะมีจังหวะโน๊ตตัวดำ 60 จังหวะ ภายในหนึ่งนาที นั่นคือแต่ละจังหวะมีความยาวหนึ่งวินาที ความเร็วขนาดนี้ใกล้เคียงคำในภาษา อิตาเลียน คำว่า larghetto จะพบว่าระบบนี้ให้ความแม่นยำกว่าการบอกด้วยคำภาษาอิตาเลียน ดังนั้นในการแสดง นักดนตรีจะใช้ระบบนี้ แต่การเลือกใช้ตัวแสดงความเร็วก็ขึ้นอยู่กับนักดนตรีแต่ละคนว่าเขาหรือเธอพอใจจะใช้แบบไหนในการแสดง

ระบบบอกความเร็วแบบนี้ถูกคิดค้นโดยผู้ประดิษฐ์เมโทรโนม คือ Dietrich Winkel (ca 1812) เมโทรโนมของเขาประกอบด้วยลูกตุ้มน้ำหนักสองลูกที่ใช้สร้างเสียงที่มีช่วงเวลาคงที่ ถ้าปรับลูกตุ้มน้ำหนักเข้าใกล้ปลายด้านบน ความเร็วจะเพิ่มขึ้นหรือลดลงตามต้องการ ถึงแม้ว่าจะไม่มีเมโทรโนม วิธีที่ง่ายที่สุดในการบอกความเร็ว คือใช้การสังเกตเข็มวินาทีบนนาฬิกาข้อมือ แล้วนับจำนวนจังหวะภายใน 15 วินาที แล้วคูณด้วย 4 ก็จะได้จำนวนจังหวะในหนึ่งนาที

2.4 การเร่งหรือการลดความเร็วของเพลง

เมื่อเวลาทางดนตรีถูกกำหนดด้วยจังหวะ บางทีเราก็สามารถยืดหยุ่นจังหวะของเพลงได้ ผู้ประพันธ์สามารถเร่งความเร็วของเพลงของเขาได้โดยเปลี่ยนความเร็วทันทีทันใดหรือลดค่าของตัวโน๊ตลง (เปลี่ยนจากโน๊ตตัวดำไปเป็นโน๊ตตัวเขบ็ตหนึ่งชั้น) การเปลี่ยนความเร็วนี้บางทีก็แบ่งได้ยาก วิธีแรกคือกระโดดไปยังความเร็วค่าใหม่ทันทีเลยหรืออีกวิธีคือเพิ่มความเร็วเป็นสองหรือสี่เท่า อย่างไรก็ดีการแบ่งจังหวะออกเป็นส่วนๆ และค่อยๆ เพิ่มความเร็วขึ้น จะทำให้ผู้ประพันธ์เข้าถึงการเปลี่ยนแปลงความเร็วได้ดีกว่า การค่อยๆ เพิ่มความเร็วแสดงด้วยคำภาษาอิตาเลียนว่า *accelerando* ในทำนองเดียวกัน การค่อยๆ ลดความเร็วใช้คำว่า *ritardando* ดังนั้นการเร่ง และลดความเร็วของเพลงก็สามารถยืดหยุ่นได้เหมือนยางรัดของ

2.5 รูปแบบของจังหวะ

ทุกจังหวะมีความยาวเท่ากันแต่ไม่ได้หมายความว่าทุกจังหวะจะมีการเน้นเท่ากันหมด ในความเป็นจริงจังหวะจะถูกจับเรียงให้เป็นกลุ่มของจังหวะที่มีทั้งการเน้น และไม่เน้นเสียง เหมือนคำพูดที่เราสามารถแต่งให้มีทั้งคำเน้น และไม่เน้นพยางค์ โดยกลุ่มนี้จะประกอบด้วยจังหวะเน้นหนึ่งจังหวะ และตามด้วยจังหวะที่ไม่เน้นหนึ่งจังหวะหรือมากกว่า เรียกกลุ่มแบบนี้ว่า *ห้อง* หลายๆ ห้องเมื่อทำซ้ำเป็นรูปแบบเดียวกัน เรียก*รูปแบบของจังหวะ* เรารู้ว่าดนตรีเป็นเครื่องวัดเวลา นั่นคือเราใช้รูปแบบของจังหวะที่ประกอบด้วย ห้องที่มีทั้งจังหวะที่เน้น และไม่เน้น ในการวัดเวลาที่เดินทางผ่านไป

2.5.1 จังหวะคู่

รูปแบบพื้นฐานที่หนึ่งห้องประกอบด้วยสองจังหวะ คือจังหวะเน้นตามด้วยจังหวะไม่เน้น เรียกว่าจังหวะคู่เช่น เรานับ *1 2 1 2* ก็เป็นการเน้นจังหวะที่หนึ่งในทุกห้อง เรียกว่าจังหวะตก

2.5.2 จังหวะสาม

เป็นรูปแบบที่หนึ่งห้องประกอบด้วยจังหวะตกเน้น และตามด้วยจังหวะไม่เน้นสองจังหวะ เช่น เรานับ *1 2 3 1 2 3 1 2 3* จะมีความรู้สึกเหมือนการเต้นรำในจังหวะวอลท์ซ

2.5.3 จังหวะสี่

รูปแบบที่หนึ่งห้องประกอบด้วยสี่จังหวะ คือจังหวะตกเน้น และตามด้วยจังหวะไม่เน้นสามจังหวะ ถือเอารูปแบบจังหวะแบบนี้เป็น*แบบทั่วไป* หรือ *จังหวะเพลงเดิน*

2.6 ลีลาของจังหวะ

เราได้แบ่งเวลาออกเป็นหน่วยย่อยเรียกว่าจังหวะ หลายจังหวะรวมกันเป็นห้อง หลายห้องรวมกันเป็น รูปแบบของจังหวะ เหมือนกับที่เราแบ่งเวลาในนาฬิกาเป็นวินาที นาที และชั่วโมง อย่างไรก็ดี ดนตรีก็ไม่จำเป็นต้องเป็นไปตามส่วนของเวลาที่สร้างขึ้น แต่แปรแปลี่ยนไปมาตามช่วงเวลา การที่ดนตรีเลื่อนไหลไปเราเรียกว่า *ลีลาของจังหวะ* เพื่อแสดงลีลาของจังหวะจึงจำเป็นต้องมีระบบที่ช่วยในการแสดงออกมาเป็นรูปธรรม

2.7 การแบ่ง และการรวมเวลา

ลักษณะของเครื่องหมายแสดงลีลาของจังหวะ ในแถบยูโร-อเมริกา จะใช้*ระบบการแบ่งเวลา* คือจะแบ่งเวลาทั้งหมดออกเป็นช่วงเล็กๆ ส่วนนักดนตรีในแถบอื่นของโลก รวมทั้งอัฟริกา และเอเซียจะใช้ระบบที่แตกต่างไป คือใช้*ระบบการรวมเวลา* ที่สร้างลีลาของจังหวะโดยการรวมหน่วยของเวลาเข้าด้วยกัน และการใช้ระบบทั้งสองจะทำให้เกิดความแตกต่างกันอย่างมากของวัฒนธรรมทางดนตรีของพวกเรา จึงไม่แปลกใจเลยที่ชาวอเมริกันจะไม่เข้าใจในคณะเครื่องเคาะของอัฟริกันตะวันตก หรือ

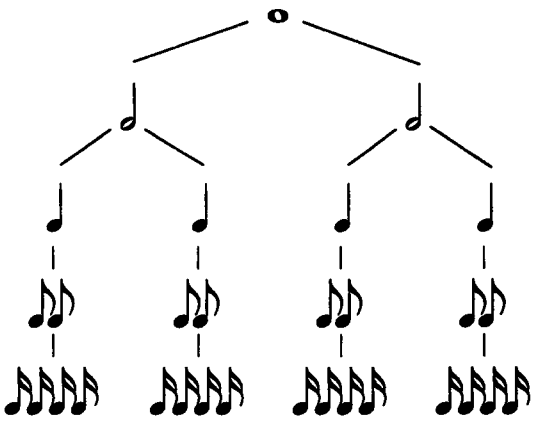
นักดนตรีอินเดียก็ยากที่จะเล่นตามวงเครื่องสายของบีโธเฟน เราจึงมีระบบการแสดงเวลาทางดนตรีสองแบบที่แตกต่างกันโดยสิ้นเชิง

2.8 ตัวโน๊ตทางดนตรี

ระบบการแสดงลีลาทางดนตรีของเรามีความสัมพันธ์กับช่วงเวลา ตัวโน๊ตเช่น โน๊ตตัวกลม, โน๊ตตัวขาว และโน๊ตตัวดำ ไม่ได้ใช้เพื่อแสดงช่วงเวลาที่ตายตัว เราใช้สัญลักษณ์แสดงช่วงเวลาที่สัมพันธ์กับค่าของโน๊ตตัวอื่น เช่นโน๊ตตัวขาวสองตัวรวมกันเท่ากับโน๊ตตัวกลม นั่นคือโน๊ตตัวกลมจะมีความยาวทางดนตรีเป็นสองเท่าของโน๊ตตัวขาว และโน๊ตห้าตัวที่ใช้กันทั่วไปคือ

รูป 2.1 ตัวโน๊ตที่ใช้กันในปัจจุบัน

และค่าของตัวโน๊ตก็จะสามารถแบ่งได้อย่างง่าย โน๊ตแต่ละตัวจะมีค่าเป็นครึ่งหนึ่งของโน๊ตตัวซ้ายมือ และมีค่าเป็นสองเท่าของโน๊ตตัวขวามือ รูปข้างล่างนี้แสดงค่าของตัวโน๊ตตัวต่าง ๆ

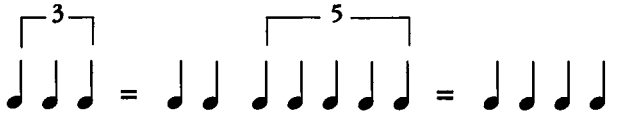


รูป 2.2 ความยาวของตัวโน๊ต

2.9 โน๊ตสามพยางค์

ความสัมพันธ์จากรูปข้างบน แสดงว่าค่าที่สัมพันธ์กับแบบสองเท่าคือ ครึ่งหนึ่ง, หนึ่งในสี่, หนึ่งในแปด และหนึ่งในสิบหก จึงเป็นการดีเมื่อเราเขียนจังหวะที่ความสัมพันธ์กันดังกล่าว แต่จะไม่ค่อยยืดหยุ่นมากพอที่จะแสดงให้เห็นถึงความเป็นอิสระของลีลาของจังหวะ นั่นคือไม่มีการแบ่งโน๊ตตัวกลมออกเป็นจำนวนคี่ แต่ถ้าเราต้องการเช่นนั้นก็ทำได้คือ โน๊ตสามพยางค์ โน๊ตสามพยางค์

จะแบ่งค่าของตัวโน๊ตให้ละเอียดขึ้นคือ อนุญาตให้แบ่งโน๊ตตัวกลมออกเป็นสามส่วนเท่าๆ กัน เขียนรวมโน๊ตทั้งสามด้วยเส้นโค้งหรือที่เรียกว่าสเลอ (slur) และเติมเลข 3 ไว้ข้างบนหรือข้างล่าง ซึ่งทำให้เราสามารถแสดงโน๊ตสามพยางค์ที่จะเล่นโดยใช้เวลาเท่ากับโน๊ตสองตัว เช่นเดียวกัน เราสามารถแบ่งให้เป็นจำนวนคี่แบบอื่นเช่น โน๊ตห้าพยางค์ แทนที่โน๊ตสี่ตัว หรือโน๊ตเจ็ดพยางค์ แทนที่โน๊ตหกตัว โดยเติมตัวเลขข้างบนหรือข้างล่าง



รูป 2.3 โน๊ตสามพยางค์

2.10 ตัวประจุด

อีกวิธีที่ทำให้ลีลาของจังหวะยืดหยุ่นขึ้นคือ การประจุด โดยการเติมจุดทางด้านขวาของตัวโน๊ต จะสามารถเพิ่มค่าของตัวโน๊ตอีกครึ่งหนึ่งของค่าของตัวเอง เช่นเรารู้ว่าโน๊ตตัวดำมีค่าเท่ากับโน๊ตตัวเขบ็ต หนึ่งชั้นสองตัว ถ้าเราเพิ่มจุดที่ตัวโน๊ตตัวดำจะมีค่าเป็น โน๊ตตัวเขบ็ตหนึ่งชั้นสองตัวบวกกับค่าอีกครึ่งหนึ่งของตัวมันเอง (โน๊ตตัวเขบ็ตหนึ่งชั้น) หรือเป็นโน๊ตตัวเขบ็ตหนึ่งชั้นสามตัวนั่นเอง ตัวประจุดจะทำให้เพลงมีความรู้สึกเหมือนเต้นรำ และมักจะใช้ในจังหวะสวิงของเพลงแจ๊ซ

2.11 ความสัมพันธ์ระหว่างลีลาของจังหวะ และความเร็วของจังหวะ

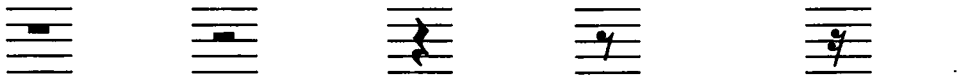
เมื่อเรารู้จักถึงความสัมพันธ์ของช่วงความยาวของโน๊ตแต่ละตัว แต่เรายังไม่สามารถกำหนดความยาวของโน๊ตนั้นได้อย่างเฉพาะเจาะจง คือเรารู้ว่าโน๊ตตัวขาวเล่นเร็วกว่าโน๊ตตัวกลมสองเท่า แต่โน๊ตตัวขาวตัวนั้นยาวนานเท่าไร

ในช่วงต้นของบท เรากำหนดความเร็วของจังหวะเป็นอัตราเร็วของจังหวะที่เคลื่อนที่ไป เพื่อที่จะกำหนดค่าของโน๊ตต่างๆ ลงไปเลยนั้น เราจำเป็นต้องกำหนดค่าของโน๊ตตัวหนึ่งเพื่อที่จะสามารถกำหนดค่าของโน๊ตตัวอื่นได้ด้วย เช่นถ้าเรากำหนดให้ในหนึ่งนาทีมี 60 จังหวะ เราจะรู้ว่าโน๊ตตัวดำมีความยาวหนึ่งวินาที หรือโน๊ตตัวขาวมีความยาวเป็นสองเท่าคือสองวินาที

ค่าความยาวของตัวโน๊ตจะแปรผันไปตามความเร็วของจังหวะ ตัวกำหนดเมโทรโนมเป็นตัวบอกความเร็วของจังหวะว่ามีกี่จังหวะในหนึ่งนาที ดังนั้น ♩ = 120 หมายถึง จะมีโน๊ตตัวดำ 120 ตัวในหนึ่งนาที และโน๊ตตัวดำจะมีความยาวครึ่งวินาที นั่นคือ โน๊ตตัวกลมจะยาวสองวินาที โน๊ตตัวขาวจะยาวหนึ่งวินาที และโน๊ตตัวเขบ็ตหนึ่งชั้นจะยาวหนึ่งในสี่ของวินาที

2.12 ตัวหยุด

ดนตรีแสดงเวลาผ่านทางเสียง แต่จะทำไม่ได้ถ้าไม่มีเสียง เหมือนกับที่เราไม่สามารถแบ่งวันแต่ละวันออกจากกันถ้าไม่มีกลางคืนมาเป็นตัวแบ่งกลางวันออกจากกัน เช่นเดียวกันในทางดนตรีแต่ละเสียงถูกแบ่งโดยความเงียบ ทุกๆ เสียงมาจาก และกลับไปสู่ความเงียบ เราสามารถแสดงความเงียบเหมือนกับการแสดงช่วงเวลาที่มีเสียง ตัวหยุดจะมีค่าเป็นสัดส่วนกันเหมือนตัวโน๊ต เช่นความยาวของโน๊ตตัวดำจะมีความยาวเท่ากับตัวหยุดของโน๊ตตัวดำ ตัวหยุดมีดังนี้



ตัวหยุดตัวกลม ตัวหยุดตัวขาว ตัวหยุดตัวดำ ตัวหยุดตัวเขบ็ตหนึ่งชั้น ตัวหยุดตัวเขบ็ตสองชั้น

รูป 2.4 ตัวหยุดที่ใช้กันในปัจจุบัน

2.13 อัตราส่วนของตัวโน๊ต และรูปแบบของจังหวะ

ในช่วงที่แล้วเรารู้ว่ารูปแบบของจังหวะคือกลุ่มของจังหวะที่เน้น และไม่เน้นหรือห้องที่รวมเข้าด้วยกัน เช่นจังหวะคู่ และจังหวะสาม ทำให้เราเข้าใจในความสัมพันธ์ระหว่างจังหวะ และความยาว ของตัวโน๊ต และเราจะเข้าใจในความสัมพันธ์ระหว่างความเร็วของจังหวะ และการแปรผันของความยาวของตัวโน๊ต ในที่สุดเราก็เข้าใจรูปแบบของจังหวะ

2.14 อัตราส่วนของตัวโน๊ต

ความเร็วของจังหวะบอกถึงความเร็วที่จังหวะเคลื่อนที่ไป รูปแบบของจังหวะบอกเราถึงกลุ่มของจังหวะในห้อง และความแปรผันของค่าของตัวโน๊ตทำให้เรารู้ถึงความสัมพันธ์จังหวะ และช่วงเวลา เมื่อรู้ทั้งหมดก็ทำให้เรารู้ค่าของตัวโน๊ตที่เคลื่อนที่ไปด้วยความเร็วที่แน่นอนภายในห้อง โดยไว้ที่จุดเริ่มต้นของเพลงด้วยอัตราส่วนของโน๊ต อัตราส่วนของโน๊ตประกอบด้วยตัวเลขสองตัว เลขตัวบนบอกถึงจำนวนจังหวะภายในห้อง และเลขตัวล่างบอกค่าของตัวโน๊ตในแต่ละจังหวะ เช่น $\frac{4}{4}$ เลขตัวบนบอกว่ามีสี่จังหวะภายในหนึ่งห้อง เลขตัวล่างบอกว่าแต่ละจังหวะคือโน๊ตตัวดำ จังหวะแรกหรือจังหวะตกเป็นจังหวะที่เน้น รูปต่อไปนี้แสดงวิธีการเขียนโน๊ตในห้องหนึ่งที่ถูกกำหนดเป็น $\frac{4}{4}$

รูป 2.5 การแสดงจังหวะในหนึ่งห้อง

2.15 ประเภทของรูปแบบของจังหวะ

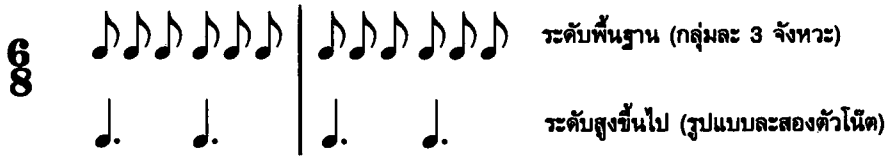
จังหวะหลายๆ รูปแบบถูกจัดเป็น แบบอย่างง่าย, แบบรวม, แบบไม่สมมาตร และแบบผสม

2.15.1 รูปแบบอย่างง่าย

รูปแบบอย่างง่ายนี้จะมีเลข 2,3 หรือ 4 เป็นเลขตัวบน และจังหวะจะถูกแบ่งเป็นสองส่วนเช่น

จังหวะคู่	จังหวะสาม	จังหวะสี่
2 2 2	3 3 3	4 4 4
2 4 8	2 4 8	2 4 8

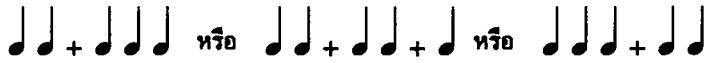
2.15.2 รูปแบบรวม

รูปแบบอย่างนี้พิจารณาจังหวะพื้นฐานบนสองระดับ ที่ระดับพื้นฐาน จังหวะพื้นฐานถูกแบ่งออกเป็นกลุ่มละสามจังหวะ และที่ระดับสูงขึ้นไป แต่ละกลุ่มจะประกอบด้วยสามจังหวะจะประกอบด้วยกลุ่มละ 2,3 หรือ 4 จังหวะในแต่ละห้อง ผู้นำเพลงมักจะใช้ระดับสูงขึ้นไป แต่นักดนตรีก็จะแบ่งจังหวะของผู้นำเพลงออกเป็นสามจังหวะ

การสร้างรูปแบบจังหวะแบบรวม ทำได้ง่ายๆ โดยคูณเลขตัวบนของรูปแบบอย่างง่ายด้วยสามที่เรียกรูปแบบรวมเพราะที่จังหวะเน้นมักจะเป็นแบบคู่ สาม หรือ สี่ แต่ จังหวะมักจะถูกแบ่งออกเป็นสามส่วน เช่น

จังหวะคู่	จังหวะสาม	จังหวะสี่
6 6 6	9 9 9	12 12 12
2 4 8	2 4 8	2 4 8

2.15.3 รูปแบบไม่สมมาตร

รูปแบบอย่างนี้จะประกอบไปด้วยจังหวะมีจำนวนแปลกออกไป เราสามารถแบ่งจังหวะเป็นกลุ่มละสอง และสามจังหวะ เพื่อสร้างรูปแบบที่น่าตื่นเต้นด้วยจังหวะเน้นที่น่าประหลาดใจ เช่น $\frac{5}{4}$ เราสามารถแบ่งเป็น



2.15.4 รูปแบบผสม

นักดนตรีสามารถเปลี่ยนรูปแบบของจังหวะได้ตามต้องการ เพลงที่มีการเปลี่ยนแปลงรูปแบบของจังหวะก็เพื่อที่จะเปลี่ยนจังหวะเน้น และเปลี่ยนความเร็วของจังหวะ จึงกลายมาเป็นรูปแบบผสม เช่น

นักดนตรีสามารถเปลี่ยนแปลงการรับรู้เวลาของนาฬิกาด้วยเสียง ผู้ประพันธ์สามารถควบคุมช่วงความยาวของเสียง ผ่านทางลีลาของจังหวะซึ่งสัมพันธ์กับจังหวะ ความเร็วของจังหวะ รูปแบบของจังหวะ และการแปรผันของค่าของตัวโน๊ต ค่าเหล่านี้สามารถอธิบาย และแสดงออกมาในรูปแบบต่างๆ ความเร็วของจังหวะสามารถแสดงด้วยคำในภาษาอิตาเลียน หรือตัวแสดงเมโทรโนม รูปแบบของจังหวะแสดงถึงสัดส่วนของจังหวะในห้อง และบอกถึงค่าของตัวโน๊ตแต่ละจังหวะ และสุดท้าย การแปรผันของค่าของตัวโน๊ต แสดงโดยสัญลักษณ์ที่บอกถึงช่วงเวลาของเสียงหรือความเงียบ

บทที่ 3
ฮาร์ดแวร์ของระบบ

3.1 ฮาร์ดแวร์ของ 8051

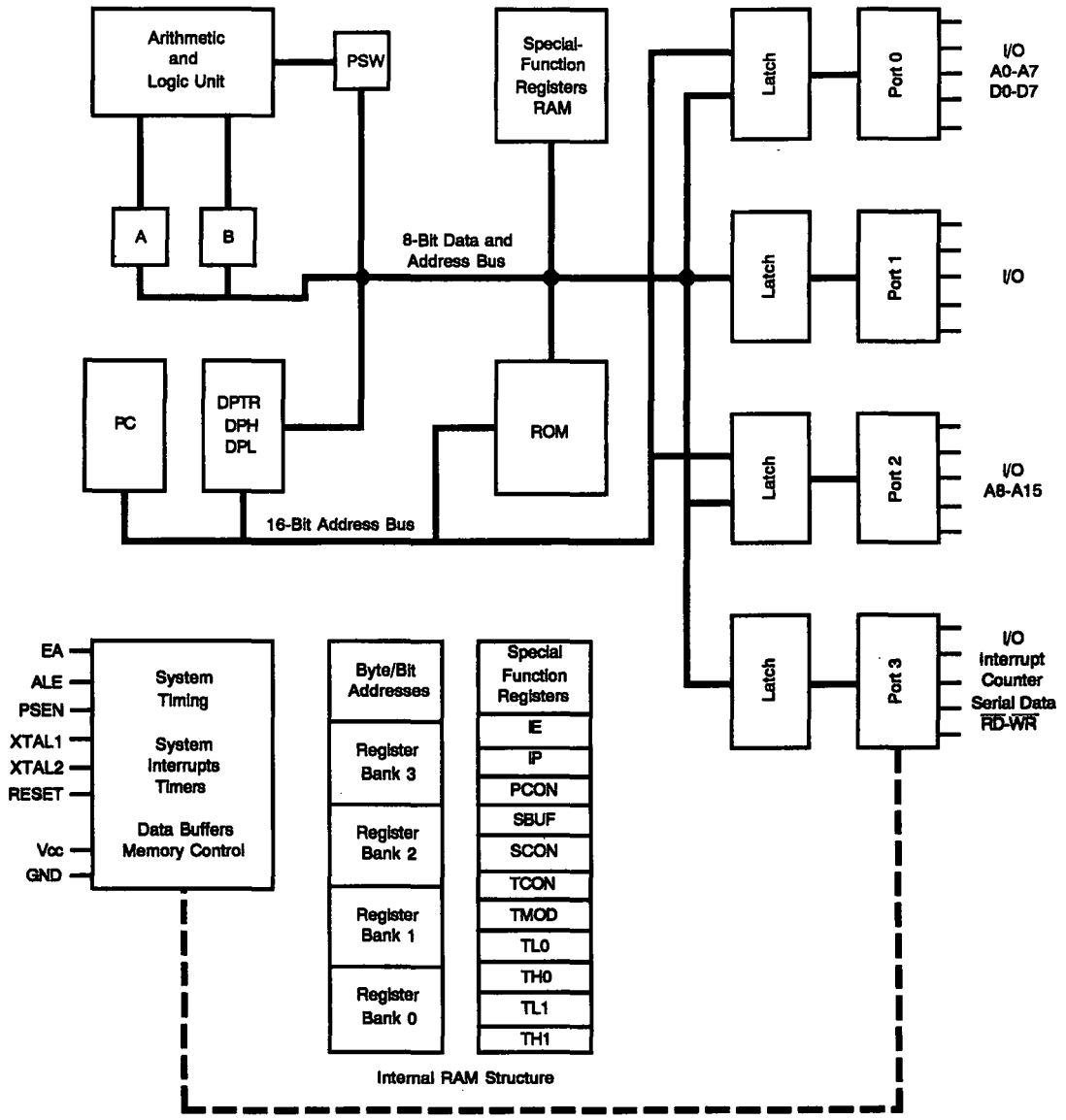
ไมโครคอนโทรลเลอร์ในตระกูล 8051 ประกอบด้วย ตั้งแต่ 8031 ถึง 8751 ทั้งโครงสร้างทั้งแบบ N-Channal Metal Oxide Semiconductor (NMOS) และ Complementary Metal Oxide Semiconductor (CMOS) บนตัวถังหลายแบบ ที่พิเศษคือ เบอร์ 8052 ที่สามารถถูกโปรแกรมด้วยภาษาเบสิกได้ รูป 3.1 แสดงถึงลักษณะเฉพาะของไมโครคอนโทรลเลอร์ตระกูลนี้ประกอบด้วย

- หน่วยความจำทั้งถาวร และชั่วคราวภายใน
- พอร์ทอินพุท และเอาท์พุท ที่สามารถโปรแกรมแต่ละบิทได้
- วงจรตั้งเวลา/วงจรนับในตัว
- การสื่อสารข้อมูลแบบอนุกรม
- หน่วยประมวลผลกลาง (CPU) คือ โปรแกรมเคาน์เตอร์ หน่วยประมวลผลคณิตศาสตร์และตรรก (ALU) รีจิสเตอร์ใช้งาน และวงจรสร้างความถี่นาฬิกา

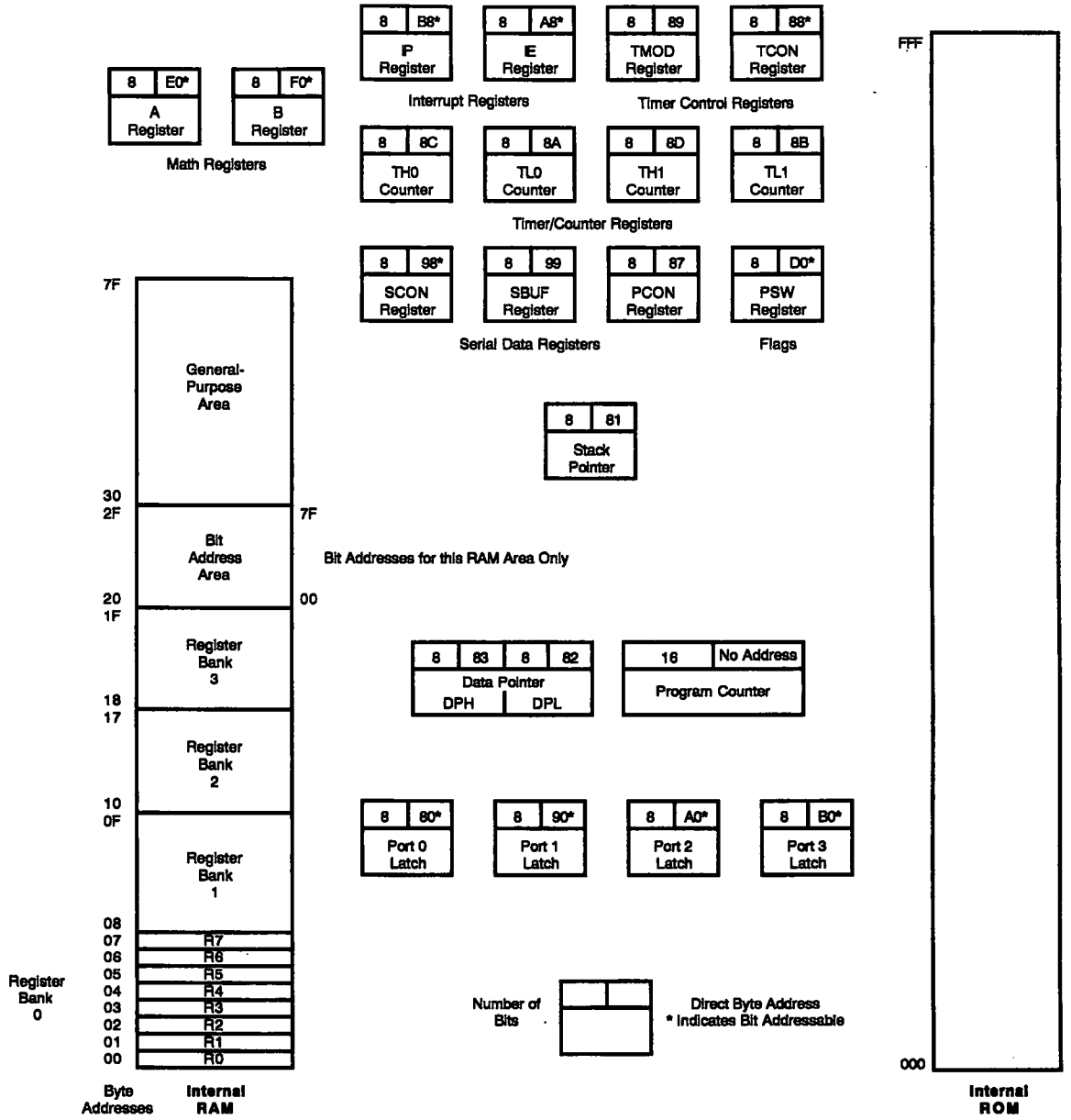
สถาปัตยกรรมของ 8051 ประกอบด้วย

- หน่วยประมวลผลกลางขนาด 8 บิท รวมทั้งรีจิสเตอร์ A (ตัวสะสม : acc) และ B
- โปรแกรมเคาน์เตอร์ (PC) และตัวชี้ตำแหน่งข้อมูล (DPTR) ขนาด 16 บิท
- โปรแกรมสเตตัสเวิร์ด (PSW) ขนาด 8 บิท
- ตัวชี้แสตก (SP) ขนาด 8 บิท
- หน่วยความจำภายในแบบ ROM หรือ EPROM (เบอร์ 8781) ขนาดตั้งแต่ 0K (8051) ถึง 4K (8051)
- หน่วยความจำชั่วคราวภายในขนาด 128 ไบท์ ประกอบด้วย
  - กลุ่มของรีจิสเตอร์ 4 กลุ่ม แต่ละกลุ่มมีรีจิสเตอร์ 8 ตัว
  - หน่วยความจำที่สามารถใช้งานแบบบิทได้จำนวน 16 ไบท์
  - หน่วยความจำข้อมูลใช้งานทั่วไป จำนวน 80 ไบท์
- ขาอินพุท และเอาท์พุทขนาด 32 บิท หรือจัดเป็นพอร์ทละ 8 บิทได้ 4 พอร์ท คือ P0-P3
- วงจรตั้งเวลา/วงจรนับขนาด 16 บิท 2 ตัว คือ T0 และ T1
- ตัวรับส่งข้อมูลแบบอนุกรม แบบฟูลดูเพล็ก : SBUF
- รีจิสเตอร์ควบคุม เช่น TCON, TMOD, SCON, PCON, IP และ IE
- รับอินเตอร์รัพท์จาก 2 แหล่งภายนอก และ 3 แหล่งภายใน

- วงจรกำเนิดความถี่นาฬิกา



รูป 3.1 สถาปัตยกรรมของ 8051



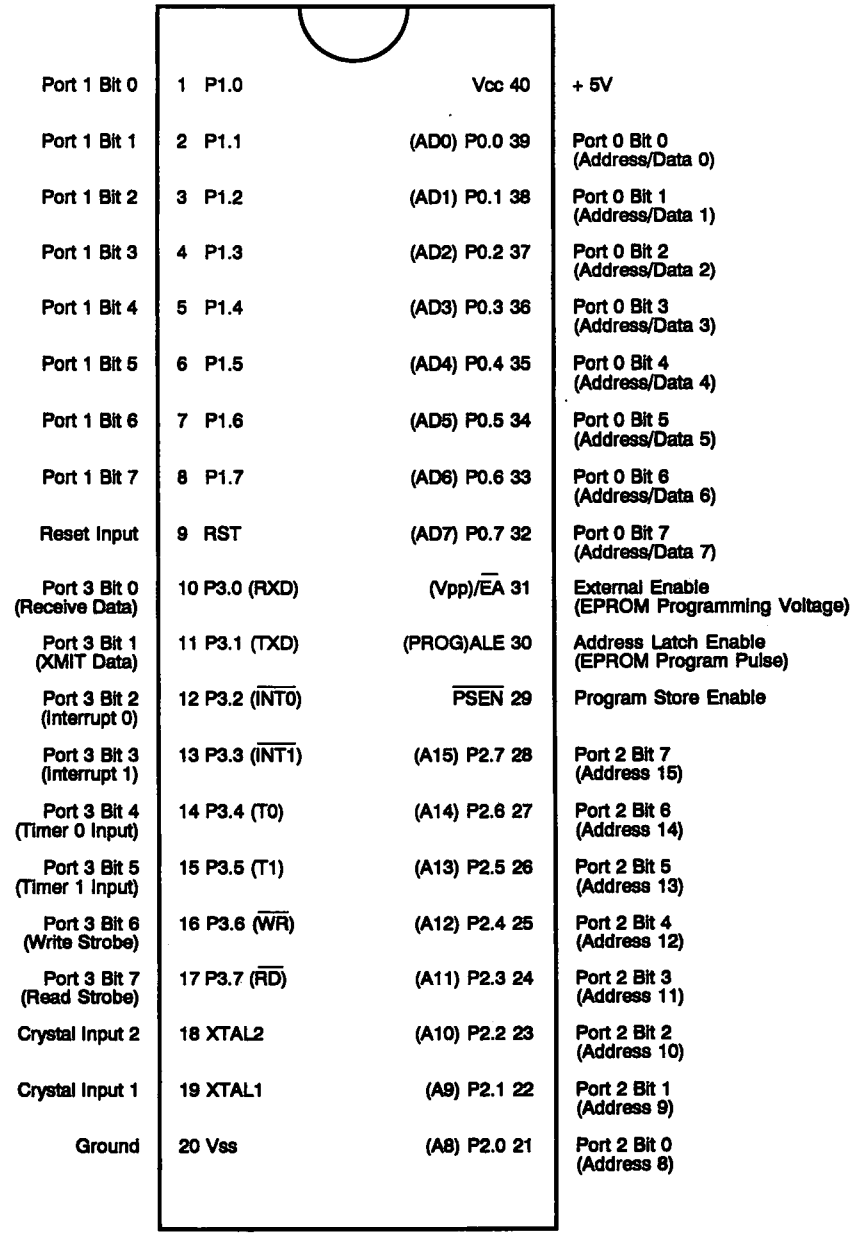
รูป 3.2 โครงสร้างการเขียนโปรแกรม

แบบจำลองการเขียนโปรแกรม บน 8051 แสดงไว้ในรูป 3.2 ประกอบด้วย รีจิสเตอร์ขนาด 8 บิท 16 บิท และตำแหน่งหน่วยความจำขนาด 8 บิท ซึ่งสามารถเรียกใช้ได้โดยตรงจากคำสั่งภายในของตัว 8051 แบบจำลองดังกล่าว ซับซ้อนขึ้น เพราะเหล่ารีจิสเตอร์พิเศษทั้งหลายที่จะทำให้ไมโครคอมพิวเตอร์ เป็นไมโครคอนโทรลเลอร์

รีจิสเตอร์ส่วนใหญ่มีหน้าที่เฉพาะของตัวเอง จึงมีตำแหน่งประจำของตัวเอง และมีชื่อเรียกเฉพาะ เช่น A, TH0 หรือ PC ตำแหน่งดังกล่าว ที่จริงแล้วก็คือ หน่วยความจำกลุ่มเดียวกับหน่วยความจำภายใน

รีจิสเตอร์แต่ละตัว ยกเว้น โปรแกรมเคาน์เตอร์ จะมีที่อยู่คือหน่วยความจำภายในขนาด 1 ไบท์นั่นเอง และจากรูป 3.2 ถ้ารีจิสเตอร์ตัวใดมีเครื่องหมาย* รีจิสเตอร์ตัวนั้นสามารถเข้าถึงได้ทั้งแบบบิท และแบบไบท์ นั่นคือ ที่ตำแหน่งของรีจิสเตอร์ตัวนั้นสามารถถูกอ่านหรือเปลี่ยนแปลงได้ หรือที่แต่ละบิทของตำแหน่งของรีจิสเตอร์ตัวนั้นสามารถถูกอ่านหรือเปลี่ยนแปลงได้

ขาต่างๆ ของ 8051 บนตัวถังดีนตะขาบขนาด 40 ขา แสดงไว้ในรูป 3.3 ประกอบด้วยชื่อเต็ม และชื่อย่อของสัญญาณแต่ละขา จะเห็นว่ามีขาหลายขาที่สามารถทำงานได้หลายอย่าง (หน้าที่อื่นบอกไว้ในวงเล็บที่ขาในรูป) แต่ไม่สามารถทำงานหลายอย่างได้พร้อมกัน



รูป 3.3 การจัดขาของ 8051

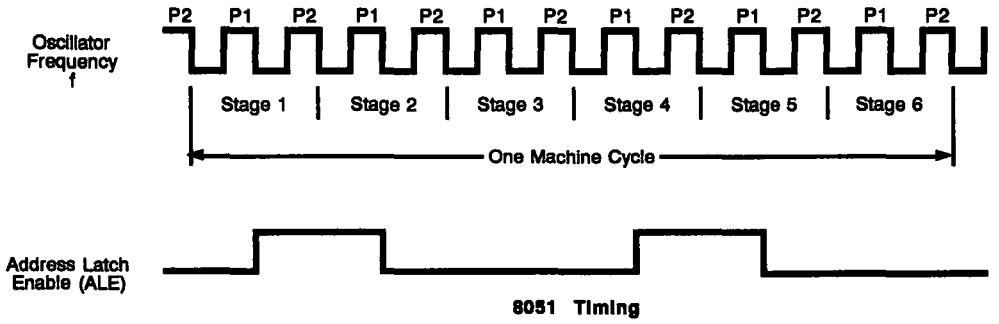
ดังนั้น ในการออกแบบจึงจำเป็นต้องเลือกใช้เพียงหน้าที่เดียวของขานั้นๆ มาใช้งาน เช่น พอร์ท 3 บิท 0 (P3.0) สามารถใช้เป็น I/O หรือเป็นขาอินพุท (RxD) ของ SBUF ตัวพักข้อมูลที่เข้ามาแบบอนุกรม ก็ต้องเลือกใช้เป็นเพียงขา I/O หรือ RxD

3.2 วงจรกำเนิดความถี่นาฬิกาบน 8051

หัวใจของ 8051 คือวงจรกำเนิดความถี่สัญญาณนาฬิกาบนตัวเองที่เพียงแค่ต่อคริสตอล และวงจรเนทเวอร์คเรโซแนนซ์ภายนอกเท่านั้น โดยความถี่ที่ได้จะเท่ากับความถี่ของคริสตอล ตัว 8051 ถูกออกแบบมาให้ทำงานได้ตั้งแต่ 1 MHz ถึง 16 MHz

ช่วงเวลาที่สั้นที่สุดของสัญญาณนาฬิกาความถี่ f เรียกว่า พัลส์ P ช่วงเวลาที่สั้นที่สุดของการทำงานคำสั่งที่ง่ายที่สุดเรียกว่าแมชชีนไซเคิล แมชชีนไซเคิลประกอบขึ้นด้วยสเตท 6 สเตท แต่ละสเตทไมโครคอนโทรลเลอร์จะใช้อ้างอิงในการทำงานต่างๆ เช่น อ่านออปโค้ดเข้ามา ตีความออปโค้ด ทำคำสั่งตามออปโค้ด หรือเขียนข้อมูล โดยหนึ่งสเตทประกอบด้วยพัลส์ 2 ลูก ดังรูป 3.4

คำสั่งแต่ละคำสั่งอาจต้องการหนึ่ง สอง หรือสี่แมชชีนไซเคิลเพื่อทำคำสั่งนั้น ก็ขึ้นอยู่กับชนิดของคำสั่ง คำสั่งจะถูกอ่านเข้ามา และกระทำโดยตัวไมโครคอนโทรลเลอร์เองอัตโนมัติ เมื่อเริ่มรีเซ็ตระบบก็จะทำการอ่านคำสั่งในหน่วยความจำโปรแกรมที่ตำแหน่ง 0000h



รูป 3.4 แมชชีนไซเคิล

3.3 หน่วยความจำภายนอก

การออกแบบระบบของ 8051 นี้ ไม่ได้ถูกจำกัดไว้เพียงหน่วยความจำชั่วคราว และถาวรภายในที่มีอยู่เท่านั้น โดยที่สามารถต่อหน่วยความจำทั้งสองอย่างเพิ่มภายนอก โดยสามารถเรียกใช้ได้โดยโปรแกรมเคาน์เตอร์ขนาด 16 บิท และ DPTR โดยมีขาสัญญาณควบคุมที่แตกต่างกัน เพื่อเรียกใช้หน่วยความจำถาวร และชั่วคราว ทั้งความถูกต้องของการเรียกใช้ก็ขึ้นอยู่กับ ช่วงแมชชีนไซเคิล และคำสั่งที่กำลังทำอยู่

มีเหตุผลมากมายที่ทำให้ต้องต่อหน่วยความจำภายนอกเพิ่ม คือเมื่อเราเริ่มพัฒนาระบบ ถ้าเราใช้ 8051 ก็จะทำให้เสียค่าใช้จ่ายมาก จึงมีการออกแบบ 8751 ออกมาให้ใช้ โดยสามารถบันทึก

โปรแกรม และถ้ามีปัญหาก็ลบออกแล้วโปรแกรมเข้าไปใหม่ได้ หน่วยความจำบน 8751 เป็นแบบ EPROM ขนาดความจุ เพียง 4K จึงเป็นข้อจำกัดถ้าโปรแกรมของเราใหญ่กว่านี้

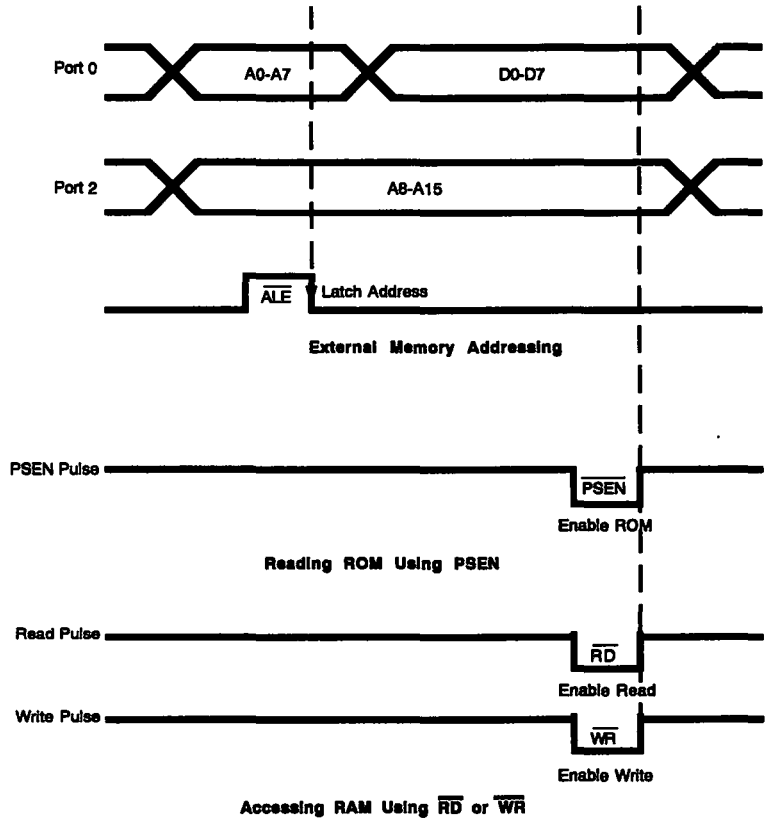
ทางออกคือ เลือกใช้ 8031 ที่ไม่มี ROM ในตัว สามารถใช้หน่วยความภายนอกได้เต็มที่ เพียงแต่ต่อขา $\overline{EA}$ ลงกราวนด์ได้เท่านั้น

3.3.1 การต่อหน่วยความจำภายนอก

เมื่อตัดสินใจใช้ 8031 แล้ว ก็จำเป็นต้องต่อหน่วยความจำโปรแกรมจากภายนอก และต่อขา $\overline{EA}$ ลงกราวนด์

8031 จะใช้ P0 เป็นทั้ง data bus และ low byte address bus ส่วน P2 จะใช้เป็น high byte address bus โดย low byte address จาก P0 จะต้องถูกแลทช์ให้ค้าง เมื่อเป็นสัญญาณ address ให้หน่วยความจำภายนอก ตัว 8031 จะส่งสัญญาณ ALE ออกมาเมื่อทำงานนี้ หลังจากนั้น P0 ก็จะสลับไปทำงานเป็น data bus ทันที

ถ้าต้องการเข้าถึง หน่วยความจำโปรแกรมภายนอก 8031 มีขา $\overline{PSEN}$ เป็นสัญญาณเพื่อทำงานนี้ และถ้าต้องการเข้าถึง หน่วยความจำข้อมูลภายนอก ก็ใช้สัญญาณจากขา $\overline{RD}$ และ $\overline{WR}$ แทน



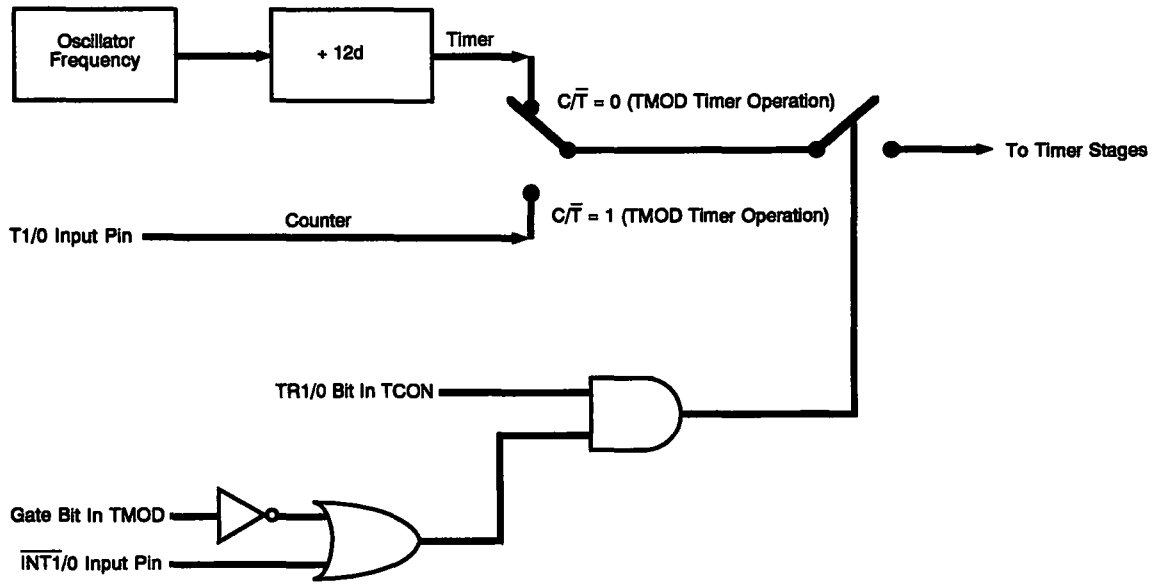
รูป 3.5 แผนผังเวลาของระบบ

3.4 วงจรตั้งเวลา/วงจรนับ

8031 มีวงจรนับขึ้นขนาด 16 บิทที่สามารถโปรแกรมได้อยู่ 2 ตัว คือ T0 และ T1 โดยสามารถนับสัญญาณนาฬิกาภายในตัว 8031 เอง หรือจากภายนอกก็ได้ ดังรูป 3.6

วงจรนับแต่ละตัวแบ่งออกเป็น รีจิสเตอร์ขนาด 8 บิท 2 ตัว เรียก Timer low (TL0, TL1) และ Timer high (TH0, TH1) เราสามารถโปรแกรมการทำงานของมันได้โดยสั่งในรีจิสเตอร์ควบคุมการทำงานของไทม์เมอร์ (TMOD) รีจิสเตอร์ควบคุมวงจรนับ และวงจรตั้งเวลา (TCON) และสั่งจากตัวโปรแกรมเอง

รีจิสเตอร์ TMOD ถูกแบ่งออกเป็นสองส่วนที่เหมือนกันทุกประการ แต่ละส่วนควบคุมวงจรตั้งเวลาแต่ละวงจรแยกจากกัน ส่วนรีจิสเตอร์ TCON มีส่วนนิบเบิ้ลบนเป็นส่วนที่ใช้ควบคุมการทำงานของวงจรตั้งเวลา และส่วนนิบเบิ้ลล่างใช้ควบคุมสัญญาณอินเตอร์รัพท์จากภายนอก



รูป 3.6 ลอจิกควบคุมการทำงานของวงจรตั้งเวลา/วงจรนับ

3.4.1 รีจิสเตอร์ควบคุมการทำงานของวงจรตั้งเวลา/วงจรนับ

3.4.1.1 Timer Control (TCON)

7	6	5	4	3	2	1	0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

บิท	สัญลักษณ์	การทำงาน
7	TF1	Timer 1 Overflow flag. เซ็ตเมื่อวงจรตั้งเวลาเพิ่มจาก 0ffh เป็น 00h เคลียร์เมื่อทำการบริการการอินเตอร์รัพท์ที่ตำแหน่ง 001Bh
6	TR1	Timer 1 Run Control bit เซ็ตเป็น 1 โดยโปรแกรมเพื่อเริ่มต้นการทำงานของวงจรตั้งเวลา เคลียร์เป็น 0 โดยโปรแกรมเพื่อให้หยุดทำงาน แต่ไม่รีเซ็ตค่าในวงจรตั้งเวลา
5	TF0	Timer 0 Overflow flag. เซ็ตเมื่อวงจรตั้งเวลาเพิ่มจาก 0ffh เป็น 00h เคลียร์เมื่อทำการบริการการอินเตอร์รัพท์ที่ตำแหน่ง 000Bh
4	TR0	Timer 0 Run Control bit เซ็ตเป็น 1 โดยโปรแกรมเพื่อเริ่มต้นการทำงานของวงจรตั้งเวลา เคลียร์เป็น 0 โดยโปรแกรมเพื่อให้หยุดทำงาน แต่ไม่รีเซ็ตค่าในวงจรตั้งเวลา
3	IE1	External interrupt 1 edge flag เซ็ตเป็น 1 เมื่อสัญญาณอินเตอร์รัพธ์ที่รับเข้ามาทางขา P3.3 ($\overline{INT1}$) เปลี่ยนจาก 1 เป็น 0 เคลียร์เมื่อทำการบริการการอินเตอร์รัพท์ที่ตำแหน่ง 0013h ไม่มีผลต่อวงจรตั้งเวลา
2	IT1	Exterenal interrupt 1 signal type control bit เซ็ตเป็น 1 โดยโปรแกรมเพื่อกำหนดให้รับการอินเตอร์รัพท์ที่ขอบขาลง เซ็ตเป็น 0 โดยโปรแกรมเพื่อกำหนดให้รับการอินเตอร์รัพท์ที่ระดับสัญญาณต่ำ ที่เข้ามาที่ $\overline{INT1}$
1	IE0	External interrupt 0 edge flag เซ็ตเป็น 1 เมื่อสัญญาณอินเตอร์รัพธ์ที่รับเข้ามาทางขา P3.2 ($\overline{INT0}$) เปลี่ยนจาก 1 เป็น 0 เคลียร์เมื่อทำการบริการการอินเตอร์รัพท์ที่ตำแหน่ง 0013h ไม่มีผลต่อวงจรตั้งเวลา
0	IT0	Exterenal interrupt 0 signal type control bit เซ็ตเป็น 1 โดยโปรแกรมเพื่อกำหนดให้รับการอินเตอร์รัพท์ที่ขอบขาลง เซ็ตเป็น 0 โดยโปรแกรมเพื่อกำหนดให้รับการอินเตอร์รัพท์ที่ระดับสัญญาณต่ำ ที่เข้ามาที่ $\overline{INT0}$

TCON สามารถเข้าถึงแบบบิทได้คือ TCON.0 ถึง TCON.7

3.4.1.2 Timer Mode Control (TMOD)

7	6	5	4	3	2	1	0
Gate	C/$\overline{T}$	M1	M0	Gate	C/$\overline{T}$	M1	M0
[Timer1]				[Timer0]			

บิท	สัญลักษณ์	การทำงาน
7/3	Gate	ออร์เกทในรูป 3.6 จะทำการควบคุมการทำงานของวงจรตั้งเวลา 1/0 โดยเซ็ตเป็น 0 โดยโปรแกรมเพื่อทำให้วงจรตั้งเวลาทำงานถ้าบิท TR1/0 ใน TCON ถูกเซ็ต และสัญญาณที่ขา $\overline{INT}$1/0 เป็น 1 เคลียร์โดยโปรแกรมเพื่อทำให้วงจรตั้งเวลาทำงานเพียงถ้าบิท TR1/0 ใน TCON ถูกเซ็ต
6/2	C/$\overline{T}$	เซ็ตเป็น 1 โดยโปรแกรมเพื่อให้วงจรทำงานเป็นวงจรนับเพื่อนับพัลส์จากภายนอกที่เข้ามาที่ขา P3.5 (T1) หรือ P3.4 (T0) เคลียร์เป็น 0 โดยโปรแกรมเพื่อให้วงจรทำงานเป็นวงจรตั้งเวลาเพื่อนับพัลส์ภายใน
5/1	M1	บิทเลือกโหมดการทำงานของวงจรตั้งเวลา/วงจรนับ บิทที่ 1
4/0	M0	บิทเลือกโหมดการทำงานของวงจรตั้งเวลา/วงจรนับ บิทที่ 0

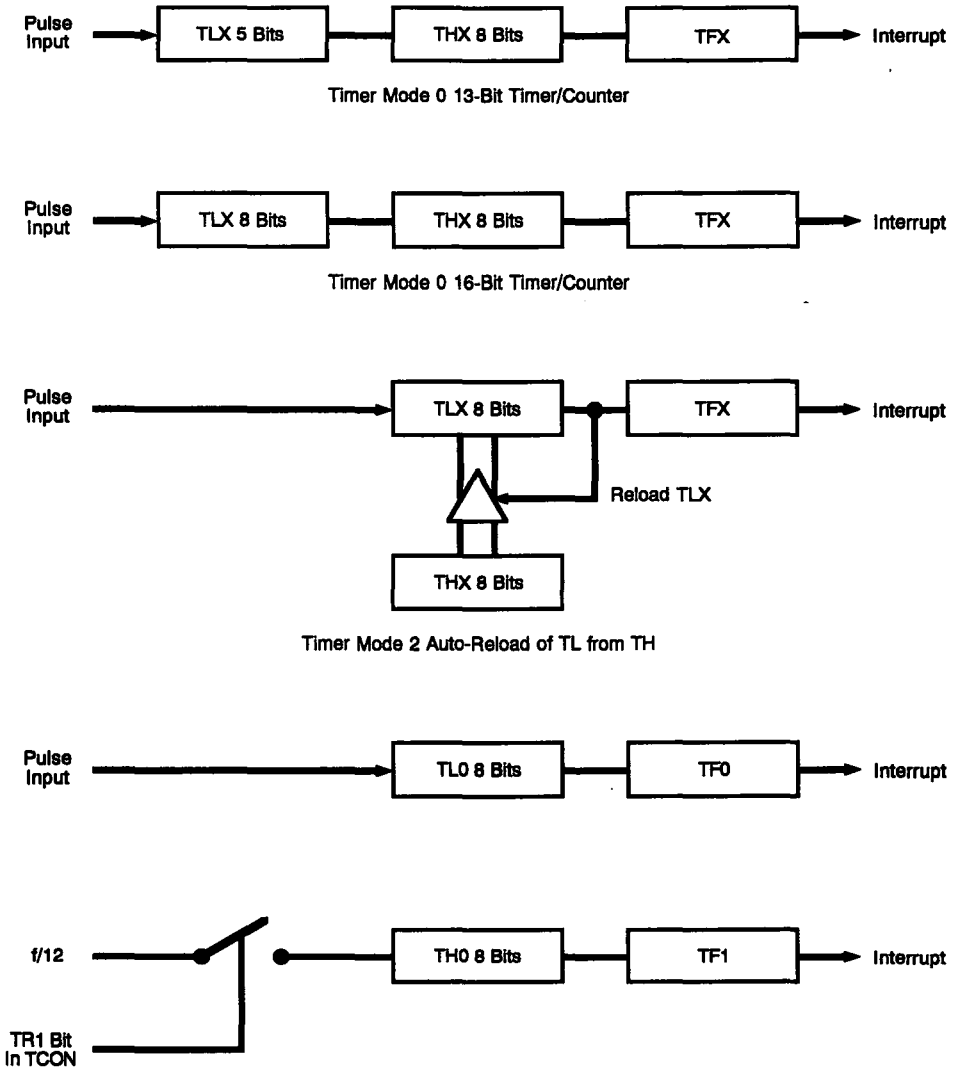
M1	M0	โหมด
0	0	0
0	1	1
1	0	2
1	1	3

TMOD ไม่สามารถเข้าถึงแบบบิทได้

3.4.2 การอินเตอร์รัพท์ของวงจรตั้งเวลา/วงจรนับ

วงจรนับที่อยู่บนชิพจะช่วยงานหน่วยประมวลผล คือ เมื่อโปรแกรมต้องการนับจำนวนพัลส์ภายในหรือภายนอกชิพ ค่าที่จะนับก็จะอยู่ที่วงจรนับ นั่นคือค่าเริ่มต้นต้องเท่ากับค่าสูงสุดลบด้วยค่าที่ต้องการนับแล้วบวกหนึ่ง วงจรนับจะเริ่มนับจากค่าเริ่มต้นไปเรื่อยๆ จนถึงค่าสูงสุดแล้วกลับมาเป็นศูนย์ จากนั้นก็จะทำการเซ็ตไทม์เมอร์แฟล็ก เงื่อนไขของแฟล็กดังกล่าว อาจจะตรวจสอบด้วยโปรแกรมเพื่อให้รู้ว่าการนับได้เสร็จแล้ว หรืออาจอาศัยการอินเตอร์รัพท์การตั้งเวลา

ถ้าวงจรนับถูกกำหนดให้เป็นวงจรตั้งเวลา นั่นคือการนับสัญญาณนาฬิกาภายใน 8051 ที่หารด้วย 12 แล้ว เช่น ถ้าความถี่ของ 8051 คือ 6 MHz สัญญาณนาฬิกาที่นับจะมีความถี่ 500 kHz



รูป 3.7 โหมดการทำงานของวงจรตั้งเวลา/วงจรนับ

3.4.3 โหมดการทำงานของวงจรตั้งเวลา/วงจรนับ

3.4.3.1 โหมด 0

คือวงจรตั้งเวลาขนาด 13 บิท ที่เหมือนกับ MCS®-48 วงจรตั้งเวลา 13 บิท ประกอบด้วยรีจิสเตอร์ THx ขนาด 8 บิท และ TLx ขนาด 5 บิท

3.4.3.2 โหมด 1

เหมือนกับในโหมด 0 เพียงแต่ รีจิสเตอร์ TLx มีขนาดถึง 8 บิทหรือเป็นวงจรตั้งเวลาขนาด 16 บิทนั่นเอง

3.4.3.3 โหมด 2

ประกอบด้วยรีจิสเตอร์ TLx ขนาด 8 บิทที่สามารถโหลดค่าเริ่มต้นที่เก็บไว้ใน THx ไปไว้ใน TLx ได้อัตโนมัติ นั่นคือ เมื่อวงจรนับนับถึงค่าสูงสุด ในจังหวะที่นับกลับมายังศูนย์ จะนำค่าที่อยู่ใน THx ไปเก็บไว้ใน TLx แล้วทำการนับต่อได้เลย

3.4.3.4 โหมด 3

วงจรตั้งเวลา T0 และ T1 สามารถถูกกำหนดให้ทำงานโหมดใดได้อิสระต่อกัน แต่ในโหมดนี้ทำไม่ได้ คือ ถ้ากำหนด T0 ในโหมด 3 ก็จะกลายเป็น วงจรนับขนาด 8 บิทที่อิสระต่อกัน 2 ชุด TL0 จะถูกควบคุมด้วยขาเกทในรูป 3.6 และจะเซ็ตแฟล็ก TF0 เมื่อโอเวอร์โฟลจาก 0FFh ไป 00h ส่วน TH0 จะรับสัญญาณนาฬิกาภายในชิพ ภายใต้การควบคุมของบิท TR1 เท่านั้น และจะเซ็ต TF1 เมื่อโอเวอร์โฟล T1 จะสามารถกำหนดให้ทำงานในโหมดใดก็ได้ แต่จะไม่มีการอินเตอร์รัพท์ เพราะ T0 ใช้แฟล็ก TF1 อยู่ เมื่อเปลี่ยนการทำงานของ T1 มาเป็นโหมด 3 วงจรนับก็จะหยุดทำงาน และคงค่าที่นับได้ไว้

3.4.4 การนับ

ข้อแตกต่างระหว่างการนับ และการตั้งเวลา คือ แหล่งกำเนิดสัญญาณนาฬิกา เมื่อเราใช้งานวงจรตั้งเวลาสัญญาณนาฬิกาที่ป้อนให้กับวงจร คือ สัญญาณนาฬิกาจากวงจรกำเนิดความถี่บนชิพที่ถูกหารด้วย 12 แต่ถ้าเราใช้งานเป็นวงจรนับแล้ว วงจรก็จะรับสัญญาณนาฬิกาที่ป้อนมาจากภายนอก เข้ามาทางขา T0 (P3.4) ให้กับวงจรนับที่ศูนย์ และที่เข้ามาจาก T1 (P3.5) ให้กับวงจรนับที่หนึ่ง และบิท $C/\overline{T}$ ใน TMOD จะต้องเซ็ตเป็น 1 เมื่อได้รับพัลส์เข้ามาทางขา Tx ดังกล่าว

สัญญาณที่ป้อนเข้าที่ขา Tx จะถูกตรวจจับทุกๆ ช่วง P2 เสตทที่ 5 ของทุกๆ แมชชีนไซเคิล ถ้ามีการเปลี่ยนแปลงสถานะจาก high เป็น low ค่าในวงจรนับก็จะเพิ่มขึ้นหนึ่งค่า และสถานะดังกล่าวต้องคงไว้ไปตลอดแมชชีนไซเคิลเพื่อให้การนับแน่นอนมากขึ้น นั่นคือสัญญาณนาฬิกาหลักจะผ่านไป 24 ลูก จึงทำให้สัญญาณความถี่สูงที่สุดที่สามารถตรวจจับได้มีค่าเท่ากับความถี่ของสัญญาณนาฬิกาหารด้วย 24 เช่น ถ้าระบบของเราใช้คริสตอล 6 MHz จะทำให้สามารถวัดสัญญาณที่ความถี่สูงสุด 250 kHz

3.5 การควบคุมอินเตอร์รัพท์

รีจิสเตอร์ IE เป็นรีจิสเตอร์ที่ใช้ควบคุมการอินเตอร์รัพท์จากแหล่งต่างๆ โดยสามารถควบคุมการอินเตอร์รัพท์ทั้งหมด หรือควบคุมเฉพาะบางอินเตอร์รัพท์ก็ได้

แต่ละบิทของรีจิสเตอร์ IE ถ้าถูกเซ็ตเป็นหนึ่งหมายถึง ยอมให้มีการอินเตอร์รัพท์จากแหล่งดังกล่าว และยังมีบิทที่เหลือชื่อ EA ที่เป็นตัวกำหนดการอินเตอร์รัพท์ทั้งหมด

3.5.1 รีจิสเตอร์ควบคุมการอินเตอร์รัพธ์

3.5.1.1 Interrupt Enable (IE)

7	6	5	4	3	2	1	0
EA	-	ET2	ES	ET	EX	ET0	EX0

บิท	สัญลักษณ์	การทำงาน
7	EA	Enable interrupt bit เคลียร์เป็น 0 โดยโปรแกรมเพื่อทำให้อินเตอร์รัพท์ทั้งหมดไม่ได้รับการบริการ เซ็ตเป็น 1 เพื่อสามารถควบคุมการอินเตอร์รัพท์จากแหล่งทั้งหมดได้โดยอิสระ
6	-	ไม่มีการใช้งาน
5	ET2	จองไว้ใช้ในอนาคต
4	ES	Enable serial port interrupt เซ็ตเป็น 1 โดยโปรแแกรมเพื่อตอบสนองการอินเตอร์รัพท์จากพอร์ทอนุกรม เคลียร์เป็น 0 เพื่อไม่ตอบสนองการอินเตอร์รัพท์จากพอร์ทอนุกรม
3	ET1	Enable timer 1 overflow interrupt เซ็ตเป็น 1 โดยโปรแแกรมเพื่อตอบสนองการอินเตอร์รัพท์จากวงจรตั้งเวลาที่ 1 เคลียร์เป็น 0 เพื่อไม่ตอบสนองการอินเตอร์รัพท์จากวงจรตั้งเวลาที่ 1
2	EX1	Enable external interrupt 1 เซ็ตเป็น 1 โดยโปรแกรมเพื่อตอบสนองการอินเตอร์รัพท์จากขา $\overline{\text{INT1}}$ เคลียร์เป็น 0 เพื่อไม่ตอบสนองการอินเตอร์รัพท์จากขา $\overline{\text{INT1}}$
1	ET0	Enable timer 0 overflow interrupt เซ็ตเป็น 1 โดยโปรแแกรมเพื่อตอบสนองการอินเตอร์รัพท์จากวงจรตั้งเวลาที่ 0 เคลียร์เป็น 0 เพื่อไม่ตอบสนองการอินเตอร์รัพท์จากวงจรตั้งเวลาที่ 0
0	EX0	Enable external interrupt 0 เซ็ตเป็น 1 โดยโปรแแกรมเพื่อตอบสนองการอินเตอร์รัพท์จากขา $\overline{\text{INT0}}$ เคลียร์เป็น 0 เพื่อไม่ตอบสนองการ อินเตอร์รัพท์จากขา $\overline{\text{INT0}}$

IE สามารถเข้าถึงแบบบิทได้คือ IE.0 ถึง IE.7

3.6 ระบบควบคุม

ประกอบด้วย 80C31 ทำงานที่ความถี่ 11.0592 MHz ต่อเพิ่มหน่วยความจำภายนอกทั้ง หน่วยความจำโปรแกรม (27C64) และหน่วยความจำข้อมูล (6116)

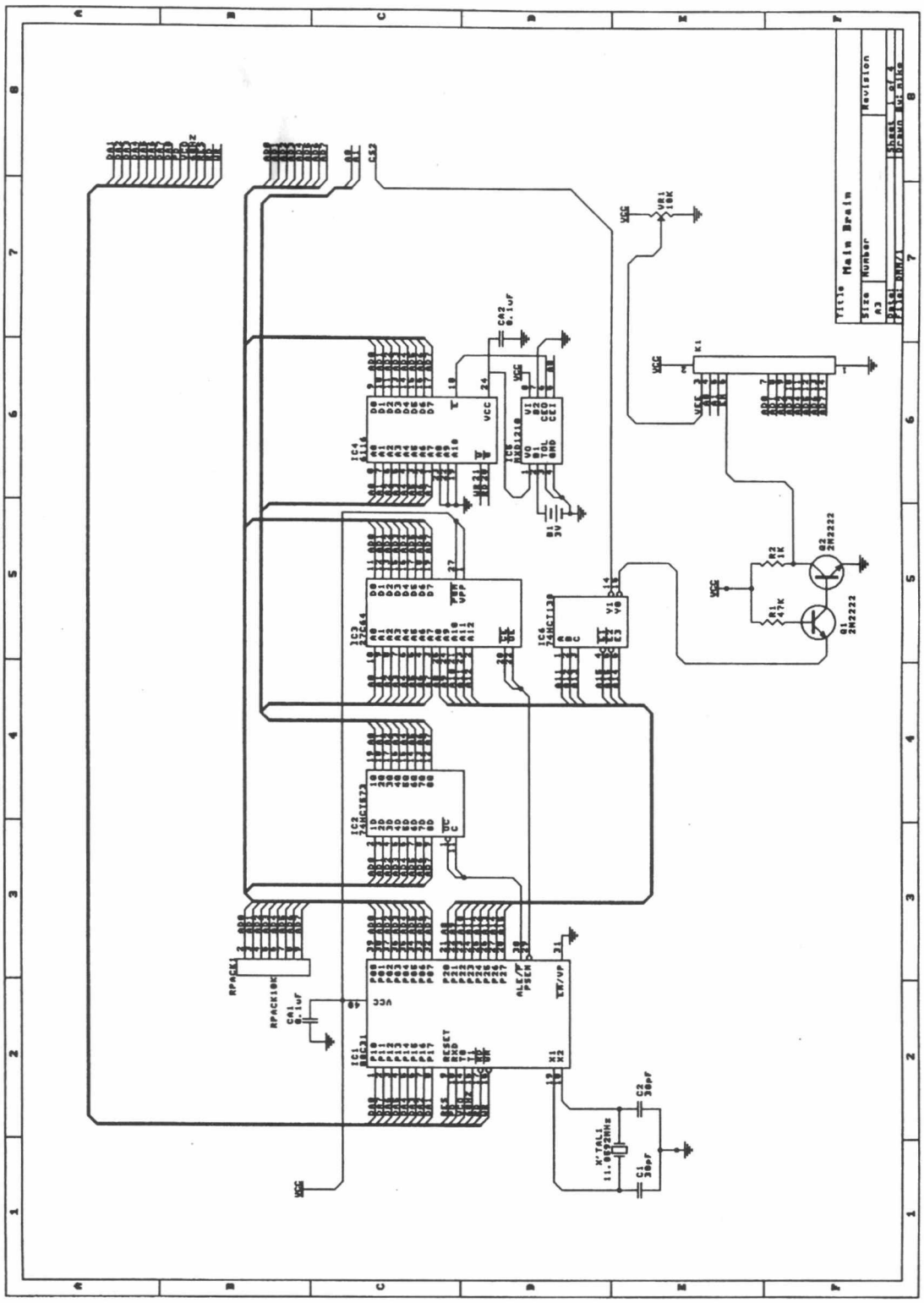
6116 จะใช้เก็บข้อมูลต่างๆ ที่ถูกตั้งโดยผู้ใช้เอง และข้อมูลเหล่านี้จะไม่สูญหายไปหลังจาก ปิดเครื่อง โดยมีชิพ MXD1210 เป็นอุปกรณ์ช่วยในการรักษาข้อมูลบน 6116 ไว้ตลอดไป

หน่วยความจำข้อมูลของระบบ ถูกแบ่งออกเป็น 3 ส่วนคือ

1. หน่วยความจำสำหรับเก็บข้อมูล ตำแหน่ง 0000h—00FFh
2. ชุดแสดงผลแบบผลึกเหลว (LCM) ตำแหน่ง 4000h—4003h
3. ส่วนติดต่อกับผู้ใช้ (82C55) ตำแหน่ง 4800h—4803h

สัญญาณที่ใช้เข้าถึงหน่วยความจำแบบต่างๆ

$\overline{PSEN}$	หน่วยความจำโปรแกรม
$\overline{RD}$, $\overline{WR}$	หน่วยความจำข้อมูล
CS1 จากขา Y0 ของ 74HCT138	ชุดแสดงผลแบบผลึกเหลว
CS2 จากขา Y1 ของ 74HCT138, $\overline{RD}$, $\overline{WR}$	ส่วนติดต่อกับผู้ใช้



รูป 3.8 วงจรส่วนควบคุมขนาดเล็ก

3.7 ส่วนติดต่อกับผู้ใช้

หัวใจหลักของส่วนนี้คือ ชิพสำเร็จรูปเบอร์ 8255 ที่ใช้ต่อเข้ากับระบบเพื่อเพิ่มพอร์ทให้กับไมโครโปรเซสเซอร์ ถึง 3 พอร์ท คือ พอร์ท A พอร์ท B และพอร์ท C โดยพอร์ท C สามารถแบ่งการทำงานได้ 2 ชุด อิสระต่อกัน คือ พอร์ท C บน ตั้งแต่บิท PC4—PC7 และพอร์ท C ล่าง ตั้งแต่บิท PC0—PC3

3.7.1 การต่อเข้าระบบ

8255 ถูกกำหนดให้อยู่ในส่วนของหน่วยความจำข้อมูลของ 8031 ช่วงตำแหน่ง 4800h—4803h นั่นคือพอร์ทต่างๆ จะมีตำแหน่งดังนี้

พอร์ท A	4800h
พอร์ท B	4801h
พอร์ท C	4802h
พอร์ทควบคุม	4803h

และทำการกำหนดลักษณะใช้งานแต่ละพอร์ท ดังนี้

พอร์ท A	เอาท์พุท
พอร์ท B	เอาท์พุท
พอร์ท C บน	เอาท์พุท
พอร์ท C ล่าง	อินพุท

ส่วนต่างๆ ที่ติดต่อกับผู้ใช้

3.7.2 ส่วนเปล่งเสียง

ระบบสามารถเปล่งเสียงออกมาได้ โดยการแปลงข้อมูลดิจิตอล 8 บิทให้เป็นสัญญาณเสียง โดยใช้วงจร R-2R ladder สัญญาณที่ออกมาจะอยู่ในรูปคลื่นสี่เหลี่ยม การที่เลือกใช้ DAC แบบ R-2R ladder นี้เพราะราคาถูก และการสร้างเสียงแบบนี้จะทำให้สามารถเปลี่ยนแปลงความดังของสัญญาณเสียงได้ด้วย

สัญญาณเสียงที่ได้จะถูกนำมาขยายอีกทีเพื่อให้มีความดังพอเหมาะ ด้วยวงจรขยายเสียงที่มีคุณภาพดี คือ ชิพเบอร์ LM386 นั่นเอง

3.7.3 การสร้างสัญญาณเสียง

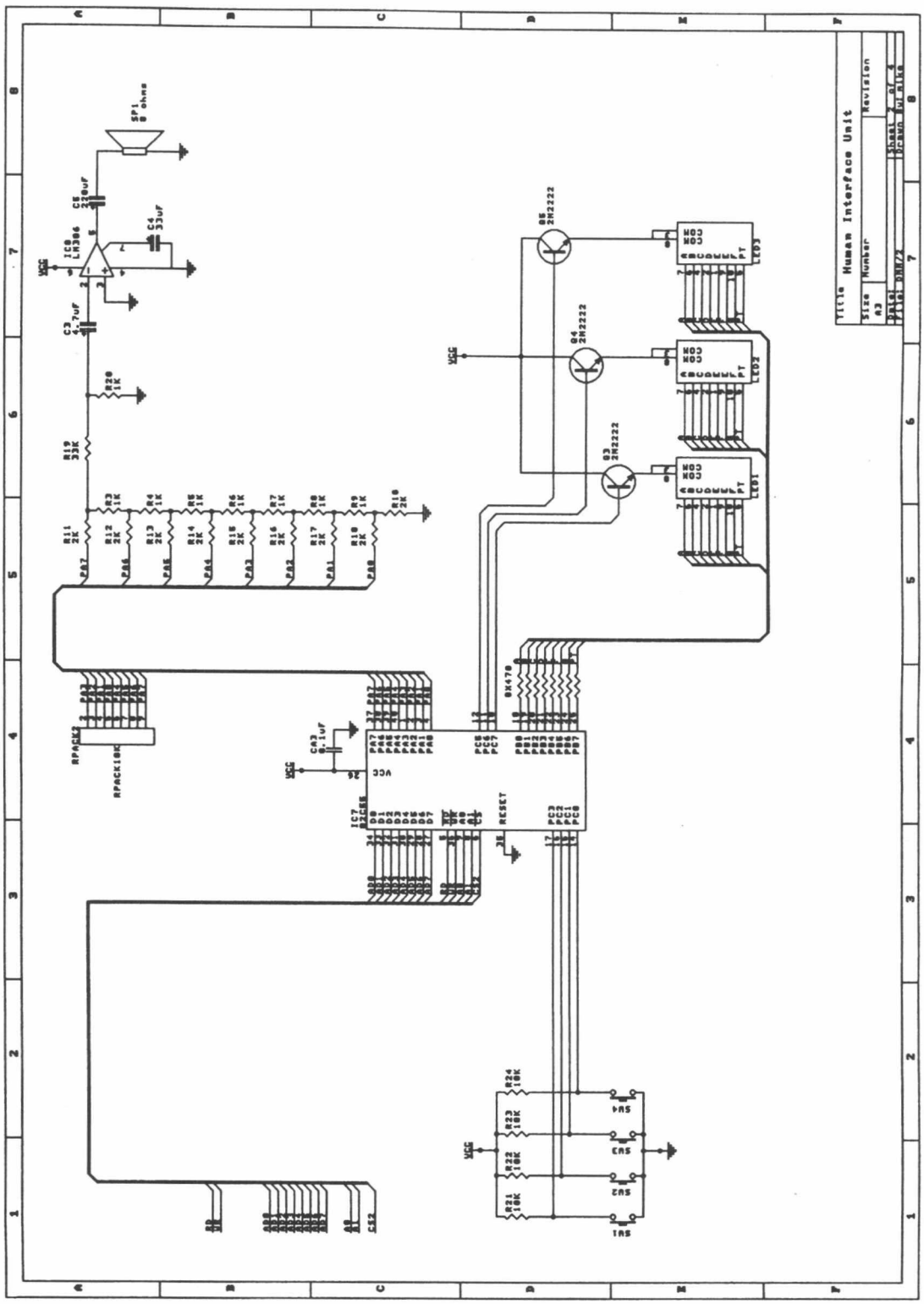
การสร้างสัญญาณเสียง วิธีที่ง่ายที่สุดคือ สร้างคลื่นสี่เหลี่ยมที่มีความถี่อยู่ในย่านเสียง เพียงแค่ส่งข้อมูลดิจิตอลออกมาสลับกับไม่ส่งข้อมูล เป็นช่วงเวลาที่เหมาะสมเท่านั้น ข้อมูลดิจิตอลที่ส่งออกมาถ้ามีค่ามาก ก็จะให้สัญญาณเสียงที่มีความดังของเสียงมาก ถ้าข้อมูลดิจิตอลที่ส่งออกมามีค่าน้อย ก็จะให้สัญญาณเสียงที่มีความดังของเสียงน้อยนั่นเอง จึงนำจุดนี้มาประยุกต์ใช้ให้สามารถปรับความดังของเสียงได้ด้วยโปรแกรม

3.7.4 ตัวเลขแสดงความเร็วของจังหวะ

ตัวเลขที่แสดงมีจำนวน 3 หลัก วิธีที่ประหยัดที่สุดคือการมัลติเพล็กซ์สายข้อมูล เพื่อให้ใช้สายข้อมูลเพียงชุดเดียว ตัวเลขที่ใช้เป็นแบบอาโนดร่วม สายข้อมูลที่ต่อเข้าคือพอร์ท B และใช้พอร์ท C บนเป็นสัญญาณมัลติเพล็กซ์ นั่นคือการที่จะทำให้ตัวเลขทั้งสามหลัก มองเห็นเหมือนติดตลอด ทำได้โดยส่งข้อมูลออกไปตลอดแล้วเลือกว่าตัวเลขหลักไหนจะติดด้วยความเร็วสูง

3.7.5 ปุ่มกด

เนื่องจากระบบมีความสะดวกสบายในการใช้ จึงมีปุ่มกดเพียง 4 ปุ่ม ปุ่มกดทั้งสี่ ต่อตรงเข้าที่พอร์ท C ล่าง โดยมีตัวต้านทาน 10kΩ พูลอัพได้ นั่นคือ ถ้าปุ่มไหนถูกกด ลอจิกของบิทนั้นจะเป็นศูนย์ เราสามารถตรวจสอบปุ่มกดได้โดยการอ่านข้อมูลของพอร์ท C ล่างเข้าไป แล้วตรวจสอบว่าบิทไหนเป็นศูนย์ เพื่อเทียบต่อไปว่าปุ่มใดถูกกด



รูป 3.9 วงจรส่วนติดต่อกับผู้ใช้

3.8 ความถี่ฐานเวลา

การที่จังหวะดนตรีดำเนินไปเรื่อยๆ เปรียบเหมือน สัญญาณนาฬิกาที่มีความถี่ต่างๆ กัน ถ้าความเร็วโน้ตตัวดำมาก เหมือนกับสัญญาณความถี่สูง ถ้าความเร็วของโน้ตตัวดำน้อย เหมือนกับสัญญาณมีความถี่ต่ำ เพียงแต่หน่วยของความเร็วของโน้ตตัวดำเป็นต่อนาที แตกต่างกับสัญญาณนาฬิกาที่มีหน่วยเป็นต่อวินาที

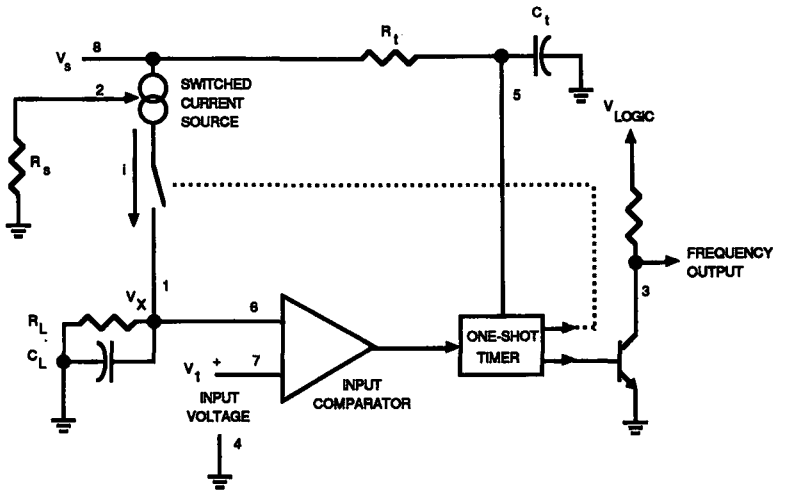
จากความคิดตรงนี้ จึงสามารถนำมาสร้าง ความเร็วโน้ตตัวดำ ได้จากวงจรกำเนิดความถี่นาฬิกาธรรมดา แล้วนำสัญญาณดังกล่าวมาหาร 60 เพื่อให้ได้สัญญาณความถี่ที่มีหน่วยเป็นต่อนาที เป็นความเร็วของโน้ตตัวดำ

3.8.1 Voltage Control Oscillator (VCO)

VCO เป็นการแปลงสัญญาณแบบหนึ่ง คือ แปลงจากสัญญาณรูปแรงดันให้เป็นสัญญาณความถี่ VCO ถูกนำมาใช้มากในวงจรเฟสล็อคลูป

หัวใจของส่วนนี้คือ LM331 ที่มีคุณลักษณะพิเศษจากชิพลักษณะเดียวกันคือ สามารถทำงานได้ด้วยไฟเลี้ยงเดี่ยวคือ 5V ให้แรงดันเอาท์พุทที่ใช้งานได้กับระดับลอจิกทุกชนิด, กันไฟต่ำ และราคาถูก หลักการของการใช้งาน LM331 เป็นวงจรแปลงแรงดันเป็นความถี่อย่างง่าย

LM331 เป็นวงจรเดี่ยวที่ถูกออกแบบมาเพื่อความแม่นยำ และความคล่องตัว ในการใช้งานเป็นวงจรแปลงแรงดันเป็นความถี่ หรือตรงกันข้าม บล็อกไดอะแกรมอย่างง่ายของ LM331 ได้แสดงดังรูป 3.10 ซึ่งประกอบไปด้วยแหล่งจ่ายกระแสที่ควบคุมได้ วงจรเปรียบเทียบแรงดันขาเข้า และวงจรตั้งเวลาแบบวันช็อต



รูป 3.10 ส่วนประกอบของ LM331

การเข้าใจการทำงานได้ดีที่สุดทำได้โดยพิจารณาวงรอบการทำงานของวงจร แปลงแรงดันเป็นความถี่ที่มีการต่อตัวต้านทาน และตัวเก็บประจุภายนอกไว้แล้ว

วงจรเปรียบเทียบแรงดัน ทำการเปรียบเทียบแรงดันขาเข้าที่เป็นบวก V_1 ที่เข้ามาที่ขา 7 กับแรงดัน V_x ที่ขา 6 ถ้า V_1 มากกว่า วงจรเปรียบเทียบแรงดันจะทำการทริกวงจรตั้งเวลาแบบวันช็อต เอาท์พุทของวงจรตั้งเวลาจะทำให้ทรานซิสเตอร์ผลิตความถี่ขาออก และแหล่งจ่ายกระแสทำงานเป็นเวลานาน $t = 1.1R_tC_t$ ในช่วงเวลานี้กระแส i จะไหลจ่ายแหล่งจ่ายกระแส โดยป้อนประจุที่มีค่าคงที่ $Q = i \times t$ ให้กับ C_L นั่นคือเป็นการประจุให้ V_x สูงกว่า V_1 เมื่อสิ้นสุดช่วงเวลาจะไม่มีกระแส i ไหลอีก และวงจรตั้งเวลาก็จะรีเซ็ตตัวเอง

ขณะที่ไม่มีกระแสไหลเข้ามาที่ขา 1 ดังนั้น C_L จะค่อย ๆ คายประจุผ่านทาง R_L จนกระทั่ง V_x ลดลงต่ำกว่า V_1 ก็จะทำให้ วงจรเปรียบเทียบแรงดันทำการทริกวงจรตั้งเวลา และเริ่มต้นคาบเวลาต่อไป

กระแสที่ไหลเข้า C_L จะมีค่าประมาณ $I_{AVE} = i \times (1.1 \times R_tC_t) \times f$ ของกระแสที่ไหลออกจาก C_L จะมีค่า $V_x/R_L = V_{in}/R_L$ ถ้า V_{in} เพิ่มเป็นสองเท่า ความถี่ก็จะเพิ่มขึ้นเป็นสองเท่าเช่นกันเพื่อให้สมดุล จึงทำให้วงจรแปลงแรงดันเป็นความถี่อย่างง่ายนี้ สามารถให้ความถี่ที่เที่ยงตรงแปรผันตามค่าแรงดันขาเข้าตลอดทั่วทั้งย่านความถี่

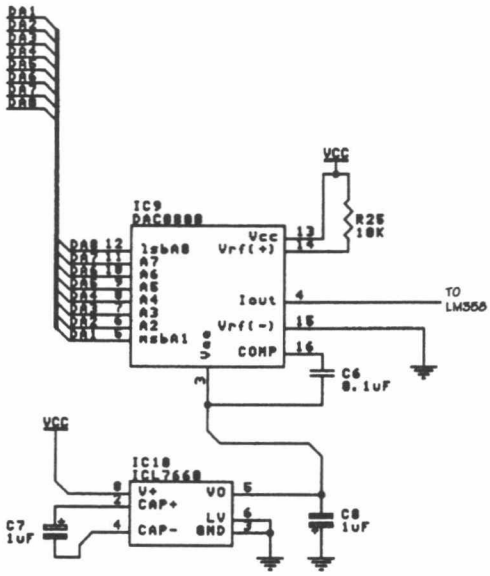
รายละเอียดเพิ่มเติม และการนำไปใช้งานจริง สามารถอ่านได้อย่างละเอียดใน data sheet ที่แนบมาให้ด้วยแล้ว

3.8.2 การควบคุมความถี่

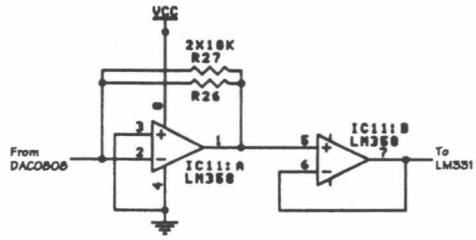
จากการออกแบบให้ระบบสามารถเปลี่ยนความเร็วของโน้ตตัวดำ ได้โดยเพียงการกดปุ่มเท่านั้น จึงต้องมีการควบคุมแรงดันที่ป้อนให้ VCO เพื่อให้ให้ได้ความถี่ตามต้องการ

การควบคุมการเปลี่ยนแปลงแรงดันที่ป้อนให้กับ VCO เพื่อให้สามารถเลือกความถี่ที่ต้องการนั้นทำได้ง่าย เพียงการใช้วงจรแปลงข้อมูลดิจิตอล 8 บิทให้เป็นระดับแรงดัน โดยเลือกใช้ชิพ DAC0808 ทำงานนี้ โดยให้แปลงข้อมูลจาก 00h—0FFh เป็นแรงดัน 0—2.5V เพื่อป้อนให้กับ LM331 ตัว DAC0808 เป็นชิพสำเร็จรูปที่ใช้การแปลงข้อมูลดิจิตอลเป็นอนาลอก ด้วยวงจร R-2R ladder โดยมีวงจรชดเชยภายในช่วยให้แรงดันที่ได้มีความเป็นเชิงเส้นมากกว่าการแปลงโดยใช้วงจร R-2R ladder ที่สร้างจากตัวต้านทานธรรมดา ที่สำคัญชิพตัวนี้ต้องการไฟเลี้ยงด้านลบ จึงสร้างไฟลบจากไฟบวก 5V ด้วยชิพสำเร็จรูป ICL7660 ที่ใช้งานง่ายโดยการต่อตัวเก็บประจุเพิ่มเข้าไปอีกเพียง 2 ตัวเท่านั้น อีกทั้งชิพตัวนี้ยังมีราคาถูก จึงเหมาะสมในงานนี้ ดังรูป 3.11

เอาท์พุทที่ออกมาจาก DAC0808 จะออกมาในรูปของกระแส ต้องทำการแปลงให้เป็นแรงดันเพื่อที่จะนำไปใช้ต่อไป จึงเลือกใช้ OPAMP เบอร์ LM358 เพราะสามารถทำงานได้ด้วยไฟเลี้ยงเดี่ยวขนาด 5V และการแปลงกระแสให้เป็นแรงดันโดยการใช้ OPAMP นี่จะทำให้สัมพันธ์กับแรงดันอ้างอิงของ DAC0808 มากขึ้นมากกว่าการใช้เพียงตัวต้านทานเท่านั้น ภายในตัว LM358 มี OPAMP 2 ตัว ดังนั้นจึงหา OPAMP ที่เหลือมาต่อเป็น buffer เพื่อให้สามารถจ่ายแรงดันได้แม่นยำมากขึ้น ดังรูป 3.12



รูป 3.11 วงจรแปลงข้อมูลดิจิตอลเป็นสัญญาณอนาลอก



รูป 3.12 วงจรแปลงกระแสเป็นแรงดัน และ buffer

จากนั้นก็ป้อนแรงดันให้กับวงจรผลิตความถี่ที่ควบคุมได้ด้วยแรงดัน ซึ่งใช้ LM331 ทำงานนี้ ได้ความถี่ออกมา

สรุปการควบคุมความถี่ฐานสำหรับความเร็วของโน้ตตัวดำเป็นดังนี้

	สัญญาณเข้า	สัญญาณออก
80C31		ข้อมูล 00h—0FFh
DAC0808	ข้อมูล 00h—0FFh	กระแสที่สัมพันธ์กับข้อมูลขาเข้า
LM358	กระแสจาก DAC0808	แรงดัน 0—2.5V
LM331	แรงดัน 0—2.5V	ความถี่ 0—3060 Hz

3.8.3 การนำความถี่ไปใช้

เมื่อเราสามารถควบคุมความถี่ให้เปลี่ยนจาก 0—3060 Hz ได้แล้ว ก็จะนำความถี่ดังกล่าวไปเป็นฐานเวลาของโน้ตตัวดำ โดยการป้อนเข้าที่ขา T0 ของ 80C31 แล้วให้วงจรนับที่ 0 ภายในตัว 80C31 ทำงานในโหมด 1

จากความถี่ของโน้ตตัวดำที่มีหน่วยเป็นต่อนาที และความถี่ฐานที่มีหน่วยเป็นต่อวินาที เรานำความถี่ฐานนี้มาหารด้วย 60 ก่อน เพื่อแปลงให้เป็นหน่วยต่อนาที จากนั้นความถี่ที่ได้สามารถนำมาสร้างให้เป็นรูปแบบของจังหวะแบบต่างๆ ได้ โดยนำมาหารต่อไปดังนี้

รูปแบบของจังหวะ	ตัวหาร
โน้ตตัวดำ	12
โน้ตตัวเขบ็ตหนึ่งชั้น	6
โน้ตสามพยางค์	4
โน้ตตัวเขบ็ตสองชั้น	3

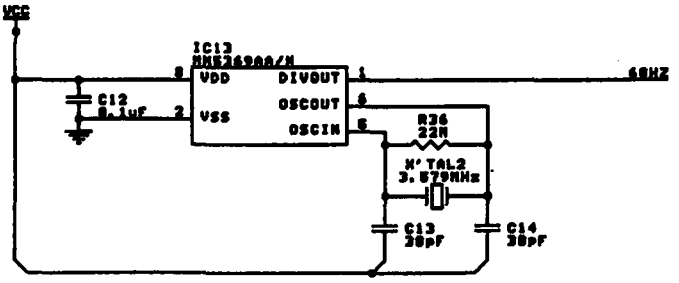
หลังจากนี้เรายังสามารถเปลี่ยนรูปแบบของจังหวะเพียงแต่ใช้ปุ่มกดอีกด้วย

3.9 การทำงานนับเวลา

ด้วยแนวความคิดเดียวกับการสร้างความถี่ฐานเวลาสำหรับสร้างความเร็วของโน้ตตัวดำ จึงเลือกใช้ชิพสำเร็จรูป เบอร์ MM5369AA/N ที่ออกแบบมาให้ใช้ในงานสร้างความถี่ฐาน 60 Hz สำหรับนาฬิกาดิจิตอล เป็นชิพที่ใช้งานง่าย โดยเพียงป้อนความถี่ 3.579 MHz ให้ก็สามารถนำความถี่ 60 Hz จากขา DIVOUT ไปใช้ได้เลย ดังรูป 3.13

3.9.1 การนำความถี่ไปใช้

จาก MM5369AA/N เราได้ความถี่ฐาน 60 Hz นำมาป้อนให้ 80C31 ที่ขา T1 แล้วให้วงจรนับที่ 1 ภายใน 80C31 ทำงานที่โหมด 2 โดยโปรแกรมให้หาร 60 ตลอด เพื่อให้ได้ความถี่ 1 Hz ใช้ในการนับเวลาต่อไป



รูป 3.13 วงจรกำเนิดความถี่ฐานเวลา 60 Hz

3.10 ส่วนจ่ายพลังงานควบคุมได้

การจ่ายพลังงานให้ระบบทำงานได้นั้นเลือกใช้วิธี แปลงไฟกระแสสลับขนาด 220V ทั่วไป มาเป็นไฟกระแสตรงขนาด +5V เพื่อเลี้ยงระบบทั้งหมด โดยระบบทั้งหมดปกติแล้วจะกินกระแส ประมาณ 200mA จึงสามารถเลือกใช้หม้อแปลงขนาดเล็กได้ และไม่กินเนื้อที่มากมายนัก

การแปลงไฟให้ได้ +5V นั้นเลือกใช้การเรียงกระแสแบบเต็มคลื่น (Full Wave Rectifier) แล้วนำมาเรียงแรงดันด้วยชิพ 7805 ได้ ไฟตรงคงที่ขนาด +5V เพื่อใช้เลี้ยงระบบทั้งหมด แต่ระบบ สามารถสั่งให้วงจรจ่ายพลังงานหยุดจ่ายพลังงานได้ โดยการเซ็ตบิท PD ซึ่งต่ออยู่กับ 80C31 ที่ขา P3.0 ให้เป็น "1" เท่านั้น

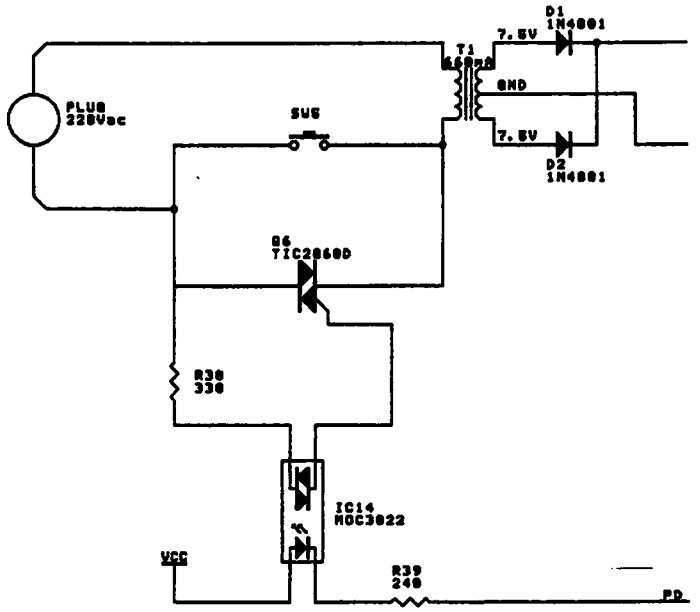
3.10.1 การควบคุม

ส่วนควบคุมทำการควบคุมการจ่ายพลังงานโดยควบคุมที่ไฟสลับ 220V ก่อนเข้าหม้อแปลง โดยใช้ไตรแอคเป็นเหมือนสวิทช์ที่จะให้ไฟ 220V ผ่านไปที่หม้อแปลงหรือไม่ ปกติแล้วไตรแอคถ้ามี การทริกที่ขาเกทแล้วจะยอมให้กระแสผ่านตัวมันเองทั้งสองทาง นั่นคือ ยอมให้กระแสสลับไหลผ่าน ตัวมันได้นั่นเอง

การควบคุมการทริกของไตรแอค เลือกใช้ออปโต้ไอโซเลเตอร์ เบอร์ MOC3022 ทำงานนี้ โดยโครงสร้างภายในของ MOC3022 เป็นการเชื่อมโยงทางแสงด้วยไดโอดเปล่งแสง และออปโต้ ไดแอค โดยใช้ออปโต้ไดแอคดังกล่าวนี้เป็นตัวทริกให้ไตรแอคนำกระแสโดยตราบใดที่ออปโต้ไดแอค ยังนำกระแสอยู่ ไตรแอคก็จะยังนำกระแสอยู่ ออปโต้ไดแอคจะนำกระแสก็ต่อเมื่อ ไดโอดเปล่งแสง ภายในนั้นเปล่งแสงอยู่ ดังนั้นการจะควบคุมให้ระบบทั้งหมดมีกำลังทำงานต่อไปหรือไม่นั้นทำได้ง่าย เพียงควบคุมที่บิท PD เท่านั้น

เริ่มต้น เมื่อต้องการเปิดเครื่อง เพียงแค่กดสวิทช์กดติดปล่อยดับที่ต่อขนานอยู่กับไตรแอค เมื่อเริ่มต้นจ่ายกำลังให้ระบบทั้งหมดชั่วคราว ระบบจะทำการรีเซ็ตตัวเองก่อน เมื่อ 80C31 ถูกรีเซ็ต นั้น บิททุกบิทของพอร์ททุกพอร์ทจะมีสถานะเป็น "1" จากนั้นเมื่อ 80C31 เริ่มทำคำสั่งแรก ถ้า เพียงสั่งให้เคลียร์บิท P3.0 ก็จะทำให้ไดโอดเปล่งแสงภายใน MOC3022 เปล่งแสง ต่อไปจะทำให้ ออปโต้ไดแอคนำกระแสตามไปด้วย นั่นคือเป็นการทริกให้ไตรแอคนำกระแสไปด้วยตลอดไปตราบที่ บิท P3.0 ยังคงเป็น "0" อยู่ กระบวนทั้งหมดนี้เกิดขึ้นภายในเวลาเสี้ยวของวินาที และสามารถ ปล่อยมือจากสวิทช์ได้โดยระบบก็ทำงานต่อไปเพราะมีกำลังจ่ายอยู่ตลอดเวลา

จากนั้นเราสามารถสั่งปิดเครื่องได้ด้วยโปรแกรม คือ เมื่อเราทำงานทุกอย่างเรียบร้อยแล้ว ต้องการปิดเครื่องก็ทำได้ง่ายเพียงเซ็ตบิท PD ให้เป็น "1" ไดโอดเปล่งแสงภายใน MOC3022 ก็ จะไม่เปล่งแสง ทำให้ออปโต้ส่วนจ่ายพลังงานก็ไม่มีไฟ +5V จ่ายให้ระบบทั้งหมด เท่ากับหยุดการ ทำงาน



รูป 3.14 วงจรควบคุมแหล่งจ่ายไฟให้ระบบทั้งหมด

บทที่ 4
การใช้งานเครื่องและซอฟท์แวร์

4.1 การใช้งานเครื่อง

1. กดสวิตซ์ ON เปิดเครื่องโดยอาศัย software
2. บนจอ LCD และ 7-seg จะแสดงค่าที่เคยตั้งไว้มีดังนี้

```
M:Quarter      TS:4/4
TC:00:00:00  MC:0000
```

M แสดงค่า Meter คือรูปแบบของการเคาะ ที่ต้องการใช้จะมี 4 รูปแบบ คือ Quarter (โน๊ตตัวดำ), Eighth (โน๊ตตัวเขบ็ตหนึ่งชั้น), Triplets (โน๊ตสามพยางค์) และ Sixteenth (โน๊ตตัวเขบ็ตสองชั้น)

TS แสดงค่า Time Signature คืออัตราส่วนของตัวโน๊ตเพื่อต้องการกำหนดจำนวนจังหวะในแต่ละห้อง มี 5 รูปแบบ คือ 2/2, 2/4, 3/4, 4/4 และ 6/8

TC แสดงค่า Time Count เพื่อตั้งเวลาหยุดการทำงานสามารถตั้งจำนวนชั่วโมง นาทีและวินาทีได้ โดยที่จำนวนชั่วโมงสามารถตั้งได้ถึง 99 ชั่วโมง ส่วนนาทีและวินาทีตั้งได้ถึง 59 ค่า

MC แสดงค่า Measure Count เพื่อตั้งจำนวนห้องเพื่อหยุดการทำงานเมื่อเคาะครบจำนวนห้อง สามารถตั้งได้ตั้งแต่ 0-9999 ห้อง

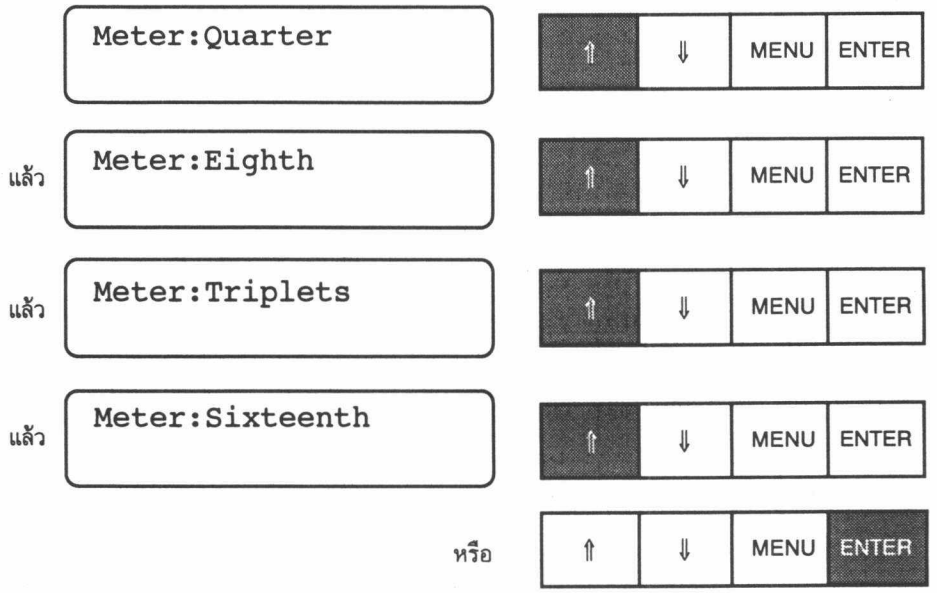
ส่วนตัวเลขบน 7-seg แสดงค่า Tempo คือความเร็วในการเคาะซึ่งตั้งได้ตั้งแต่ 0-255 ค่า

ผู้ใช้สามารถตัดสินใจเลือกการทำงานได้ 2 ทางคือ

1. กด MENU เพื่อตั้งค่าทั้งหมดใหม่

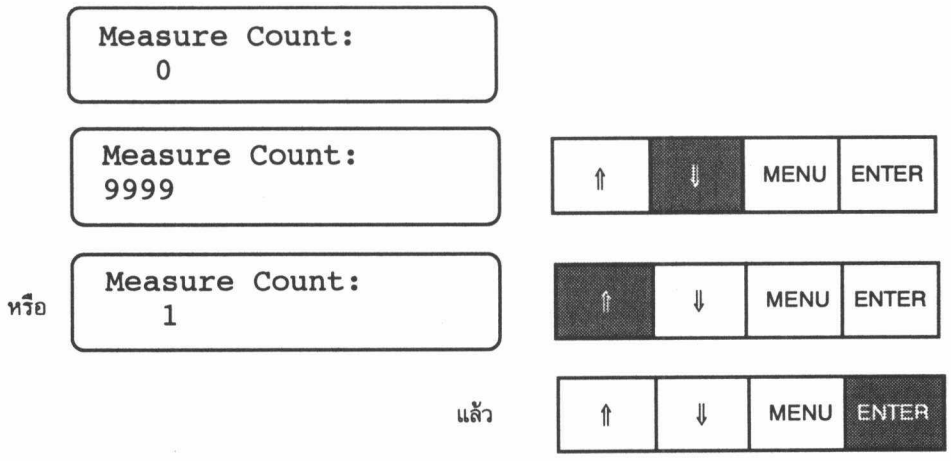
2. กด ENTER เพื่อสั่งเคาะตามค่าเดิมที่เคยตั้งไว้แล้ว

3. หากผู้ใช้เลือกกด MENU แล้วหน้าจอจะแสดง Meter ให้ผู้ใช้เลือกตั้งค่าใหม่โดยกดปุ่มขึ้นหรือลง จนได้ค่าที่ต้องการ แล้วกด ENTER เพื่อไปตั้งค่าอื่นต่อไป หรือถ้าไม่ตั้งค่าใหม่ก็สามารถกด ENTER ได้เลย

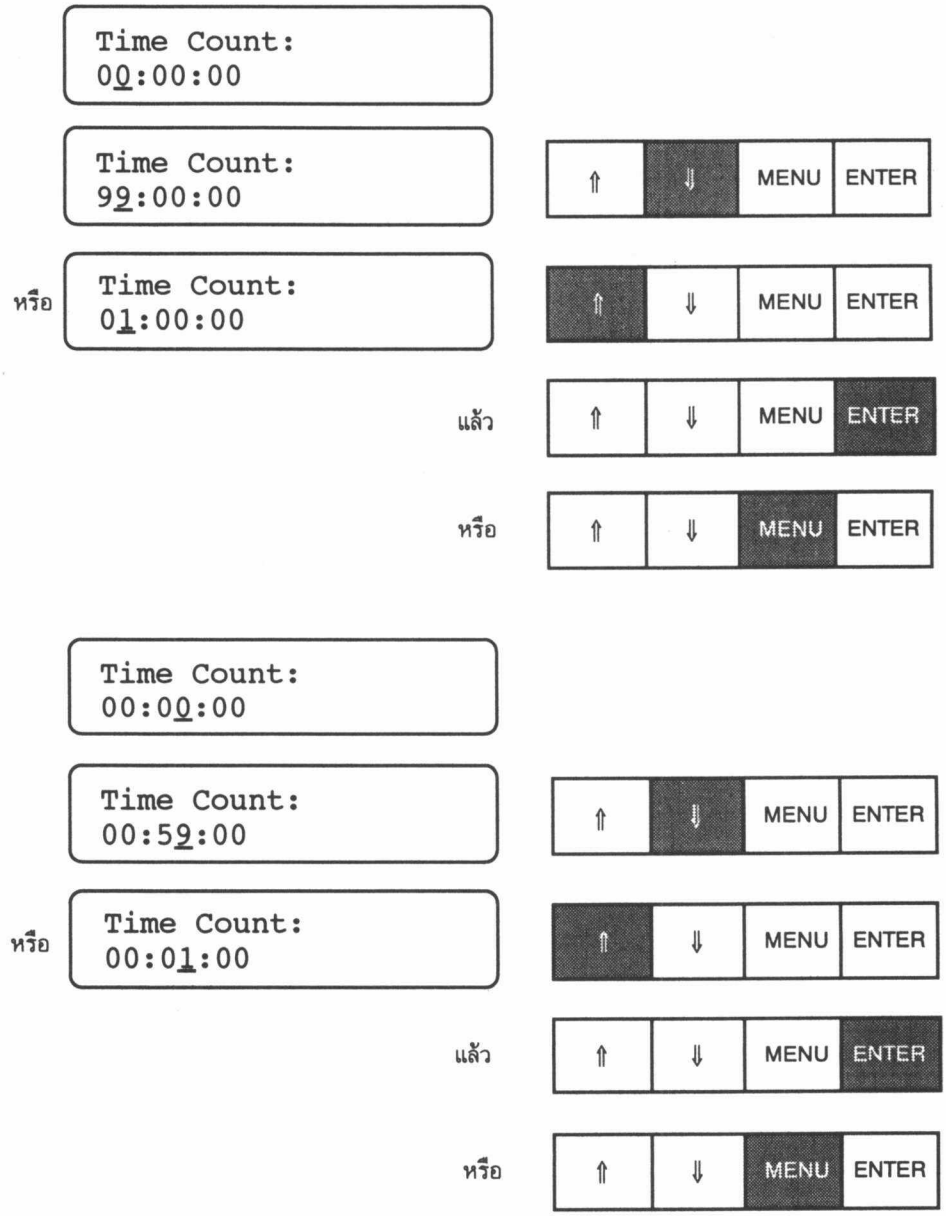


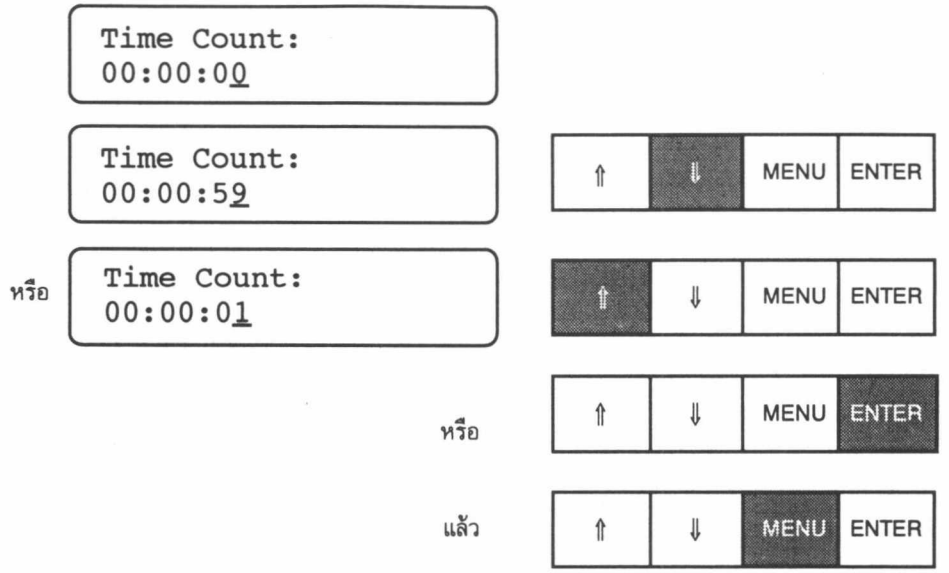
4. หน้าจอจะแสดง Time Signature และผู้ใช้ก็เลือกตั้งค่าได้ในลักษณะเดียวกับข้อ 3

5. หน้าจอจะแสดง Measure Count และผู้ใช้ก็เลือกตั้งค่าได้ในลักษณะเดียวกับข้อ 3 เช่นกัน

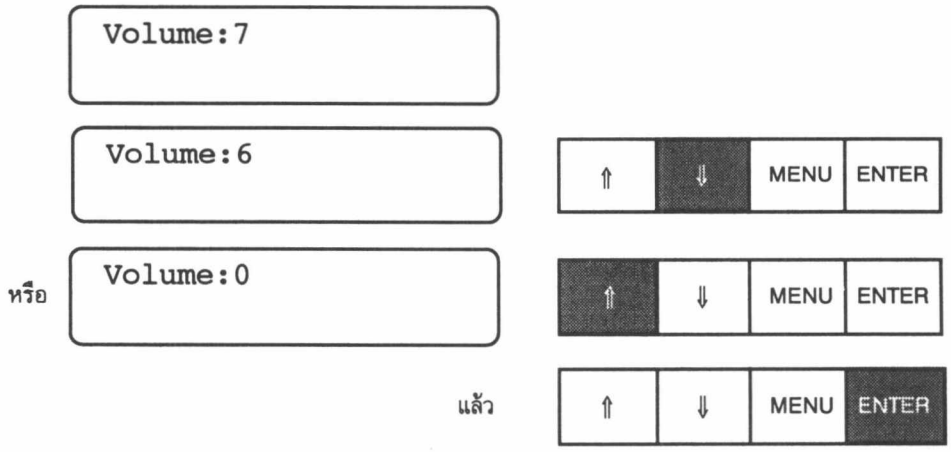


6. หน้าจอจะแสดง Time Count ผู้ใช้สามารถกดปุ่มขึ้นหรือลง เพื่อตั้งจำนวนชั่วโมง และกด ENTER เพื่อต้องการเปลี่ยนไปตั้งค่านาที เลือกกดปุ่มขึ้นหรือลงเพื่อค่าใหม่ และกด ENTER เพื่อต้องการเปลี่ยนไปตั้งค่าวินาที แล้วเลือกกดปุ่มขึ้นหรือลงเพื่อตั้งค่าใหม่อีกเช่นกัน จากนั้นหากผู้ใช้ยังต้องการเปลี่ยนตั้งค่า ชั่วโมง นาที และวินาทีใหม่ก็เลือกกด ENTER ได้เพราะปุ่ม ENTER ใช้สำหรับเปลี่ยนการตั้งค่าทั้งสาม ในลักษณะวนเฉพาะกรณีตั้งค่า Time Count นี้เท่านั้น และหากผู้ใช้ต้องการออกไปตั้งค่าอื่นต่อไปก็กด MENU ระหว่างเลือกค่าเหล่านี้ได้ทันที

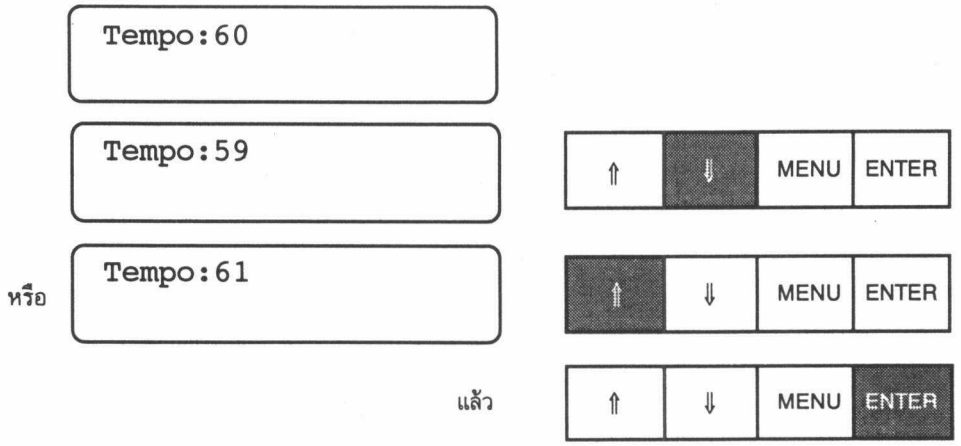




7. หน้าจอจะแสดง Volume และผู้ใช้ก็เลือกตั้งค่าได้ในลักษณะเดียวกับข้อ 3 โดยที่ความดังของเสียงจะเปล่งออกทางลำโพงตามค่าที่เปลี่ยนไป



8. หน้าจอจะแสดง Tempo และผู้ใช้ก็เลือกตั้งค่าได้ในลักษณะเดียวกับข้อ 3 โดยที่ค่าตัวเลขนั้นจะออกไปแสดงที่ 7-seg ด้วย



หลังจากนี้หากผู้ใช้ต้องการกลับไปตั้งค่าต่าง ๆ ใน MENU ใหม่ ก็กด MENU แต่ถ้าต้องการสั่งเคาะก็กด ENTER ได้เลย หรือผู้ใช้ต้องการปิดเครื่องหลังตั้งค่าทั้งหมดนี้แล้วก็สามารถกดปุ่มลง ได้ทันที โดยที่ค่าที่เพิ่งตั้งเหล่านั้นก็จะถูกเก็บไว้แสดงเมื่อมีการเปิดเครื่องครั้งต่อไป หรือไม่ผู้ใช้ก็สามารถปิดเครื่องหลังเคาะเสร็จแล้วก็ได้เช่นกัน

หลังจากที่การเคาะเสร็จสมบูรณ์แล้ว ผู้ใช้จะตั้งค่าใหม่หรือจะสั่งให้เคาะใหม่อีกครั้งก็เลือกกด MENU หรือ ENTER ได้เหมือนเดิม

4.2 ซอฟท์แวร์

ซอฟท์แวร์หรือโปรแกรมที่ใช้ในการควบคุมการทำงานทั้งหมดของเครื่อง ของเครื่องเคาะจังหวะสำหรับนักดนตรีนี้ พัฒนาขึ้นด้วยภาษาแอสเซมบลีของไมโครคอนโทรลเลอร์ตระกูล MCS®-51 เอง โดยสามารถควบคุมส่วนต่างๆ ของระบบได้อย่างมีประสิทธิภาพ

โปรแกรมควบคุมการทำงาน แบ่งเป็น 6 ส่วน คือ

4.2.1 โปรแกรมเปิดเครื่อง

4.2.2 โปรแกรมตั้งค่าเมนู

4.2.3 โปรแกรมแสดงผลทั้ง LCD และ 7-SEG

4.2.4 โปรแกรมเคาะ

4.2.5 โปรแกรมสั่งหยุดขณะที่กำลังเคาะอยู่

4.2.6 โปรแกรมปิดเครื่อง

4.2.1 โปรแกรมเปิดเครื่อง

หน้าที่การทำงาน

เพื่อเปิดเข้าสู่ระบบการทำงาน โดยการจ่ายกำลังจากหน่วยจ่ายพลังงาน

หลักการทำงาน

ใช้คำสั่ง clr p3.0 เพื่อไปทำให้ MOC3022 ทำงานและมีพลังงานจ่ายแก่ระบบ

4.2.2 โปรแกรมตั้งค่าเมนู

เป็นโปรแกรมหลักเพื่อตั้งค่าต่างๆ สำหรับการเคาะ ซึ่งแบ่งเป็นโปรแกรมย่อยๆ ได้ดังนี้

4.2.2.1 โปรแกรมตั้งค่า Meter

4.2.2.2 โปรแกรมตั้งค่า Time Signature

4.2.2.3 โปรแกรมตั้งค่า Measure Count

4.2.2.4 โปรแกรมตั้งค่า Time Count

4.2.2.5 โปรแกรมตั้งค่า Volume

4.2.2.6 โปรแกรมตั้งค่า Tempo

4.2.2.1 โปรแกรมตั้งค่า Meter

หน้าที่

เพื่อตั้งรูปแบบของตัวโน๊ตที่ใช้เคาะและเก็บไว้ยังหน่วยความจำ

หลักการทำงาน

รูปแบบของตัวโน๊ตจะมีทั้งหมด 4 แบบ โดยอาศัยการเลือกและเก็บไว้ที่ตำแหน่ง Address ที่ต้นโปรแกรม 48h โดยกำหนดให้ Meter มีค่าเป็น 80h

แทน Quarter โน๊ต และหากมีการเปลี่ยนค่าเพิ่มขึ้นหรือลดลง ก็อาศัยการเลื่อนบิทของค่า 80h ไปทางขวาหรือซ้ายตามลำดับ และเช็คค่าที่ได้เพื่อส่งรูปแบบออกไป

4.2.2.2 โปรแกรมตั้งค่า Time Signature

หน้าที่

เพื่อตั้งอัตราส่วนของตัวโน๊ตที่เคาะซึ่งจะสัมพันธ์กับจำนวนจังหวะในที่เคาะหนึ่งห้อง

หลักการทำงาน

มีทั้งหมด 5 ค่าด้วยกันคือ 2/2, 2/4, 3/4, 4/4 และ 6/8 อาศัยการเก็บค่าคล้ายข้อ 2.1 และเก็บที่ Time Sign ตำแหน่ง 49h และกำหนดให้ Time Sign มีค่าเริ่มต้นด้วย 02h และเมื่อมีการเปลี่ยนค่าขึ้นหรือลง ก็ให้มีการเลื่อนบิทไปทางซ้ายหรือขวาตามลำดับ

4.2.2.3 โปรแกรมตั้งค่า Measure Count

หน้าที่

สำหรับกำหนดจำนวนห้องที่ต้องการให้เคาะ

หลักการทำงาน

อาศัยการเก็บค่าที่เป็นเลขฐานสิบ โดยใช้ 2 Address คือ Measure ตำแหน่ง 54h และ (Measure+1) ตำแหน่ง 55h เมื่อ Measure เก็บเลขหลักสิบได้ถึง 99 และ (Measure+1) เก็บเลขหลักพันก็มีค่าได้ถึง 99 เช่นกัน

4.2.2.4 โปรแกรมตั้งค่า Time Count

หน้าที่

สำหรับกำหนดเวลาที่เคาะ

หลักการทำงาน

เวลาที่เก็บจะแบ่งเป็น วินาที นาที และชั่วโมง ซึ่งจะถูกเก็บเป็นเลขฐานสิบ มี 3 Address คือ timer, (timer+1) และ (timer+2) ที่เก็บค่าทั้ง 3 ตามลำดับ ตั้งแต่ Address 57h—59h

4.2.2.5 โปรแกรมตั้งค่า Volume

หน้าที่

กำหนดและเปล่งเสียงที่เคาะตามความดังที่ต้องการ

หลักการทำงาน

ค่าความดังจะมีทั้งหมด 7 ค่า คือ 0—7 โดยที่แต่ละค่าจะแทนด้วยเลขฐานสิบหก 8 บิท คือ 20h—0ffh เก็บไว้ที่ Volume ตำแหน่ง 4ah

4.2.2.6 โปรแกรมตั้งค่า tempo

หน้าที่

กำหนดความเร็วที่ใช้ในการเคาะ

หลักการทำงาน

ค่าความเร็วที่ใช้มีตั้งแต่ 0-255 ค่า โดยที่จะอาศัยการเก็บเป็นเลขฐานสิบไว้ 2 Address คือ temdec และ (temdec+1) ตำแหน่ง 41h และ 42h ที่ temdec จะเก็บเลขหลักสิบได้ถึง 99 ส่วน (temdec+1) เก็บหลักร้อยได้ถึง 2

4.2.3 โปรแกรมแสดงผลทาง LCD และ 7-SEG

หน้าที่

เพื่อติดต่อใช้งานได้ตามความต้องการของผู้ใช้ และแสดงผลให้รับทราบ และสามารถแบ่งหลักการทั้ง 2 อย่างได้ดังนี้

4.2.3.1 โปรแกรมแสดงผลที่ LCD

เราทำการ decode Address ช่วง 4000h—4003h สำหรับการใช้งาน LCD ซึ่งโปรแกรมกำหนดให้คำสั่ง write instruction อยู่ที่ Address 4000h เพราะสัญญาณ RS และ R/W เป็น 0 และคำสั่ง write data อยู่ที่ Address 4001h เพราะสัญญาณ RS เป็น 0 และ R/W เป็น 1

การใช้คำสั่งกำหนดค่าใน DDRAM ADDRESS ก็กำหนดให้ data pointer แทนด้วยค่าตำแหน่งของ write instruction และส่งค่าตัวเลขของคำสั่งนั้นๆ ออกไป หากทำการส่งข้อมูลก็ต้องกำหนดให้ data pointer แทนด้วย ค่าตำแหน่งของ write data และส่งรหัสแอสกีที่ตรงกับตัวเลขหรือตัวอักษรนั้นๆ ออกไป

4.2.3.2 โปรแกรมแสดงผลที่ 7-SEG

ใช้ 3 บิท บนของ Port C คือ บิท 5, 6 และ 7 สำหรับสวิทช์ ที่หลักหน่วย สิบ และร้อยของ 7-SEG ตามลำดับ เมื่อมีสัญญาณ high ไปที่ Transister นั้นๆ ส่วน port B ใช้ส่ง data 8 บิท แต่ตลอดการใช้งาน 7-SEG จะถูก scan ตลอดเวลา ดังนั้น โปรแกรมแสดงผลที่ 7-SEG จะถูกเรียกตลอดเวลา

4.2.4 โปรแกรมเคาะ

หน้าที่

เปล่งเสียงออกทางลำโพงตามค่าที่ต้องการที่ตั้งไว้ในเมนู และหยุดเคาะเมื่อเวลาผ่านไปเท่ากับเวลาที่ตั้งไว้ให้เคาะ หรือหยุดเคาะเมื่อจำนวนห้องที่เคาะไปแล้วเท่ากับจำนวนห้องที่ตั้งไว้ให้เคาะ

หลักการ

อาศัยการอินเทอร์รัพท์จาก Timer0 และ Timer1 โดยที่ Timer0 จะเป็นตัวกำหนดความเร็วของโน๊ตตัวดำสำหรับควบคุมจำนวนของเสียงที่เคาะ ส่วน Timer1 จะส่งเป็นความถี่ 60 เฮิร์ทซที่เป็นฐานเวลา 1 วินาทีสำหรับเวลาที่เพิ่มขึ้นในขณะเคาะ

4.2.5 โปรแกรมสั่งหยุดขณะที่กำลังเคาะอยู่

หน้าที่

เพื่อขัดจังหวะการเคาะเมื่อต้องการปิดเครื่องหรือตั้งค่าในเมนูใหม่ โดยที่ไม่จำเป็นต้องรอให้การเคาะสิ้นสุดตามเวลา หรือจำนวนห้องที่ตั้งไว้

หลักการทำงาน

ขณะที่โปรแกรมเคาะยังทำงานอยู่ ต้องมีการเช็คว่ามีการกดปุ่มขึ้นหรือไม่ ถ้ามีให้หยุดการเคาะทันที

4.2.6 โปรแกรมปิดเครื่อง

หน้าที่

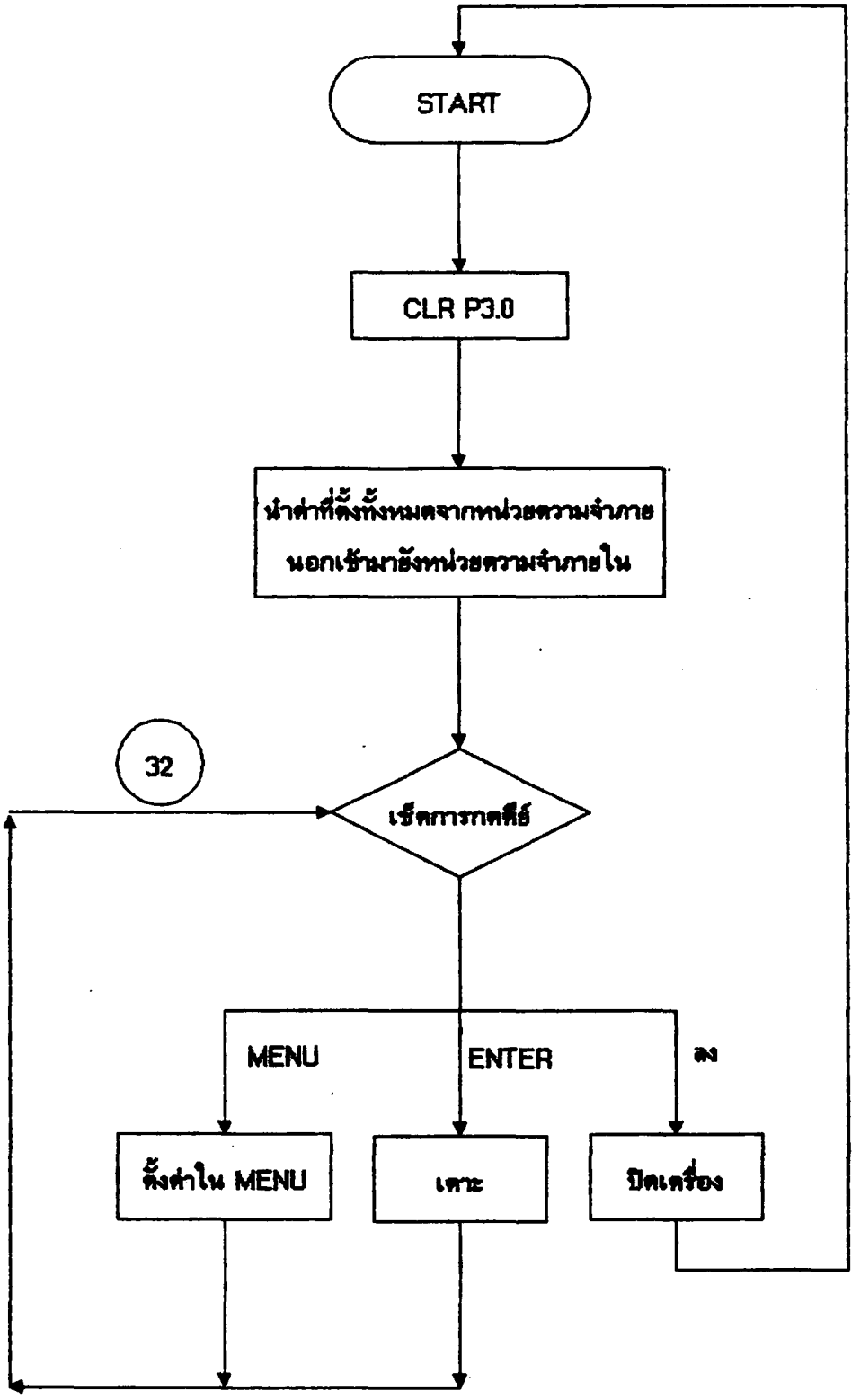
เพื่อปิดเครื่องหลังจากที่มีการสั่งหยุดขณะที่กำลังเคาะอยู่

หลักการทำงาน

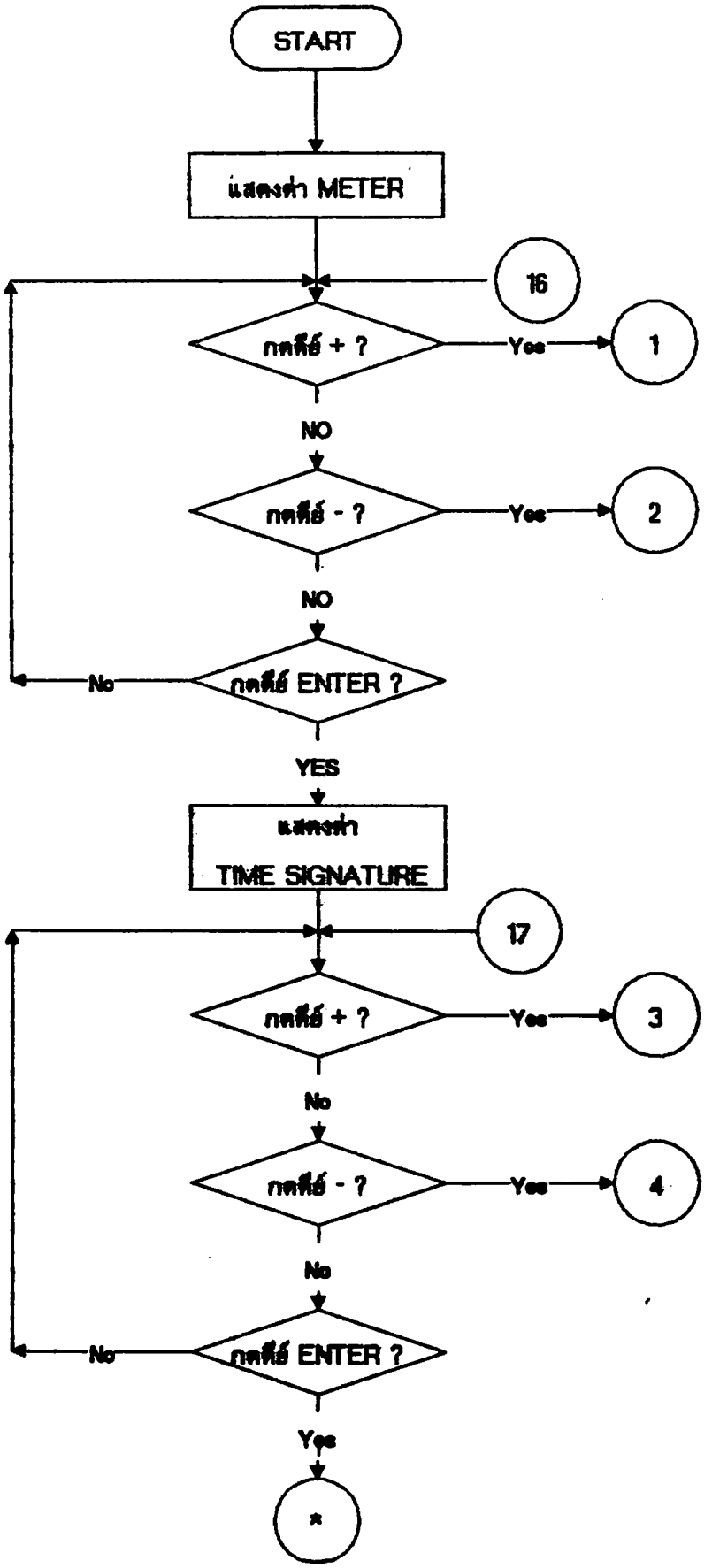
หลังจากที่หยุดเคาะในขณะทำงานแล้ว โปรแกรมต้องทำการเช็คปุ่มว่า มีการกดปุ่มลง MENU หรือ ENTER ถ้ามีการกดปุ่มลงก็ทำการเซ็ตบิท P3.0 เป็น 1 ทำให้ไปปิดหน่วยจ่ายพลังงาน จึงไม่มีการจ่ายกำลังเข้าวงจร

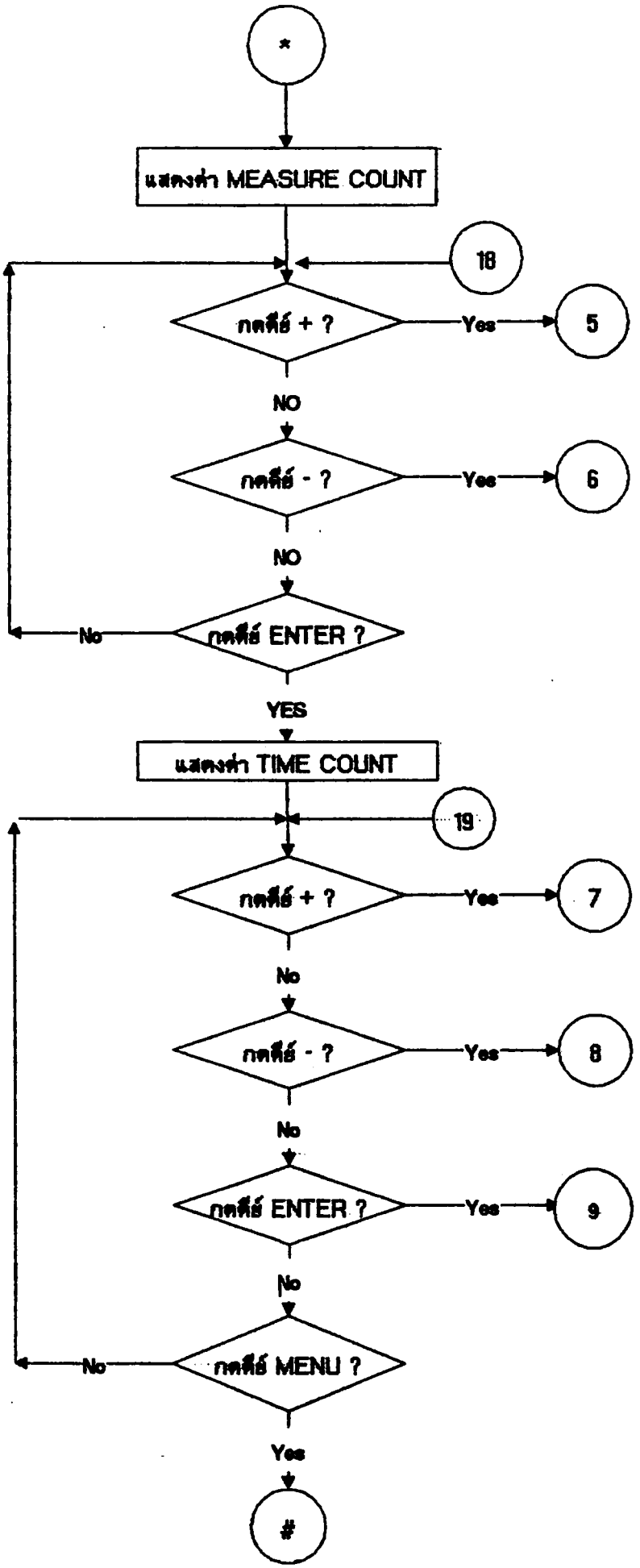
และต่อไปนี้คือแผนผังการทำงานของตัวโปรแกรม

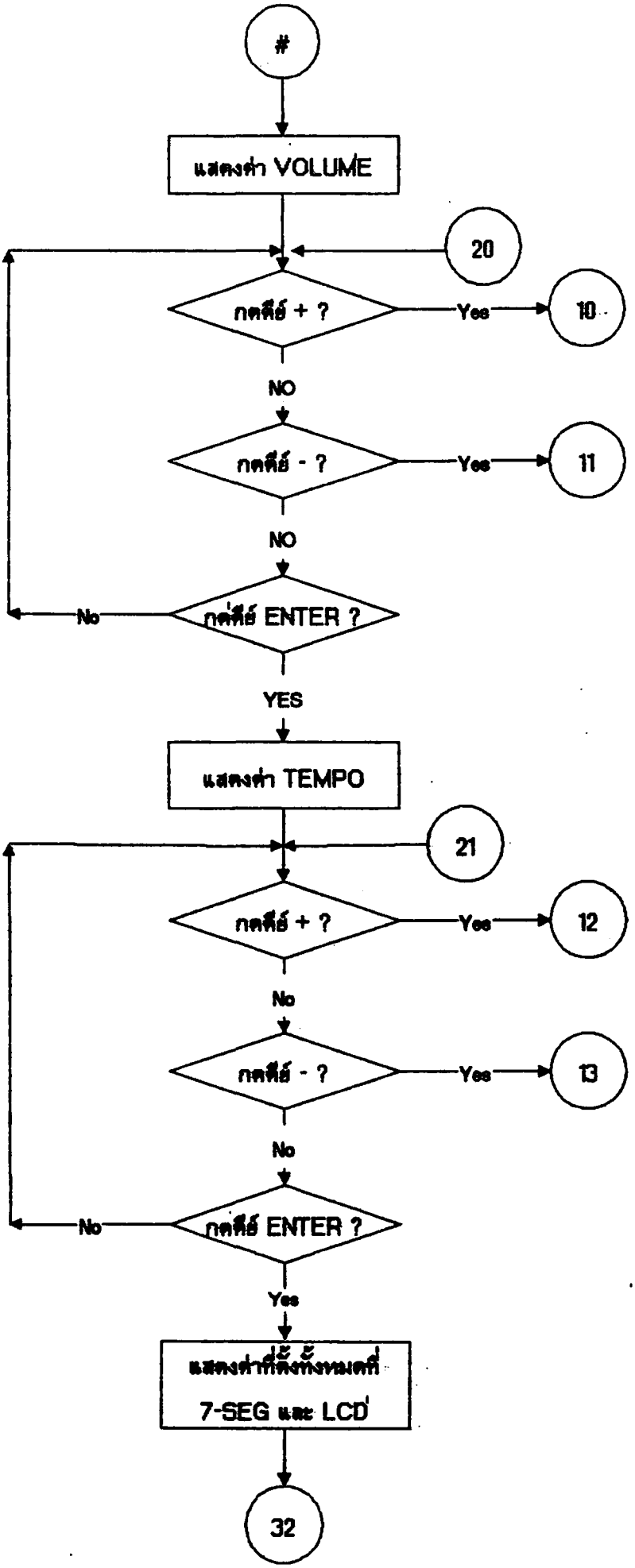
FLOWCHART แสดงขั้นตอนการใช้งานเครื่องเตาะจังหวะ

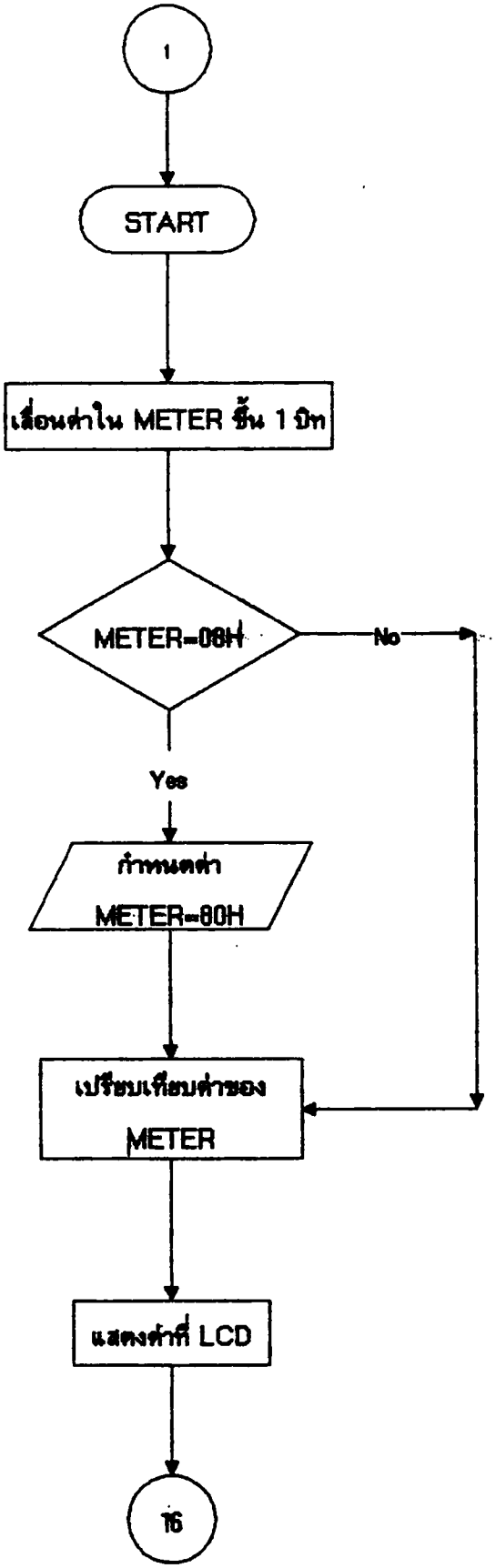


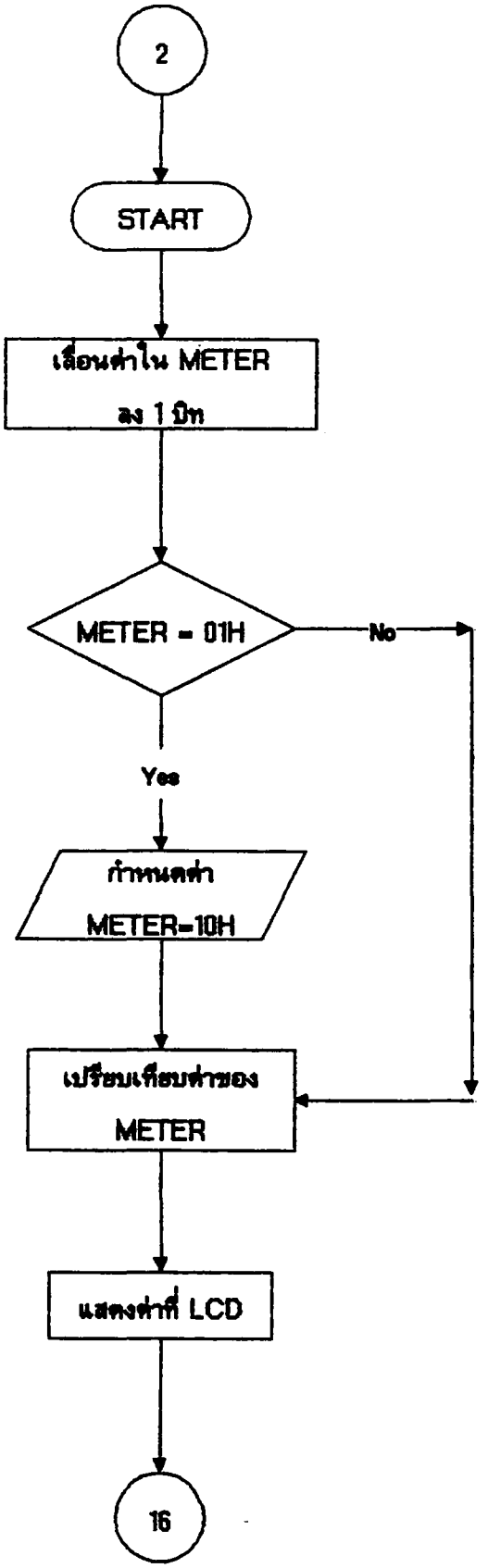
FLOWCHART ของโปรแกรมเมนู

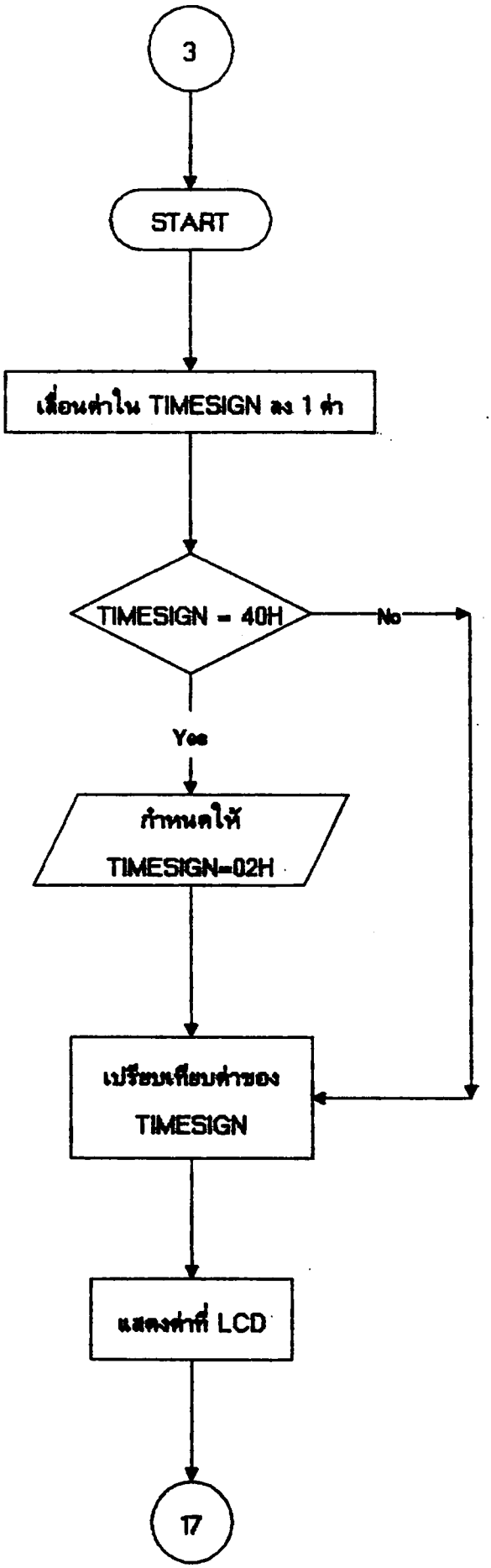


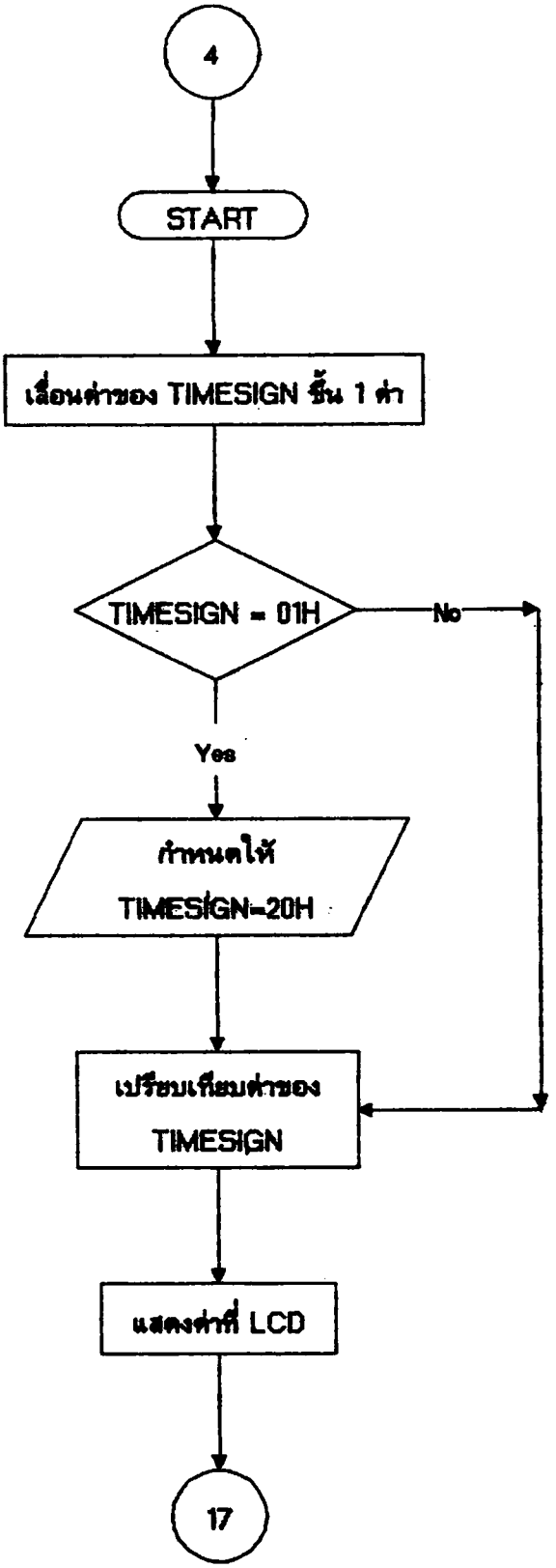

*
แสดงค่า MEASURE COUNT
18
กดคีย์ + ?
Yes
5
NO
กดคีย์ - ?
Yes
6
NO
กดคีย์ ENTER ?
No
YES
แสดงค่า TIME COUNT
19
กดคีย์ + ?
Yes
7
No
กดคีย์ - ?
Yes
8
No
กดคีย์ ENTER ?
Yes
9
No
กดคีย์ MENU ?
No
Yes
#

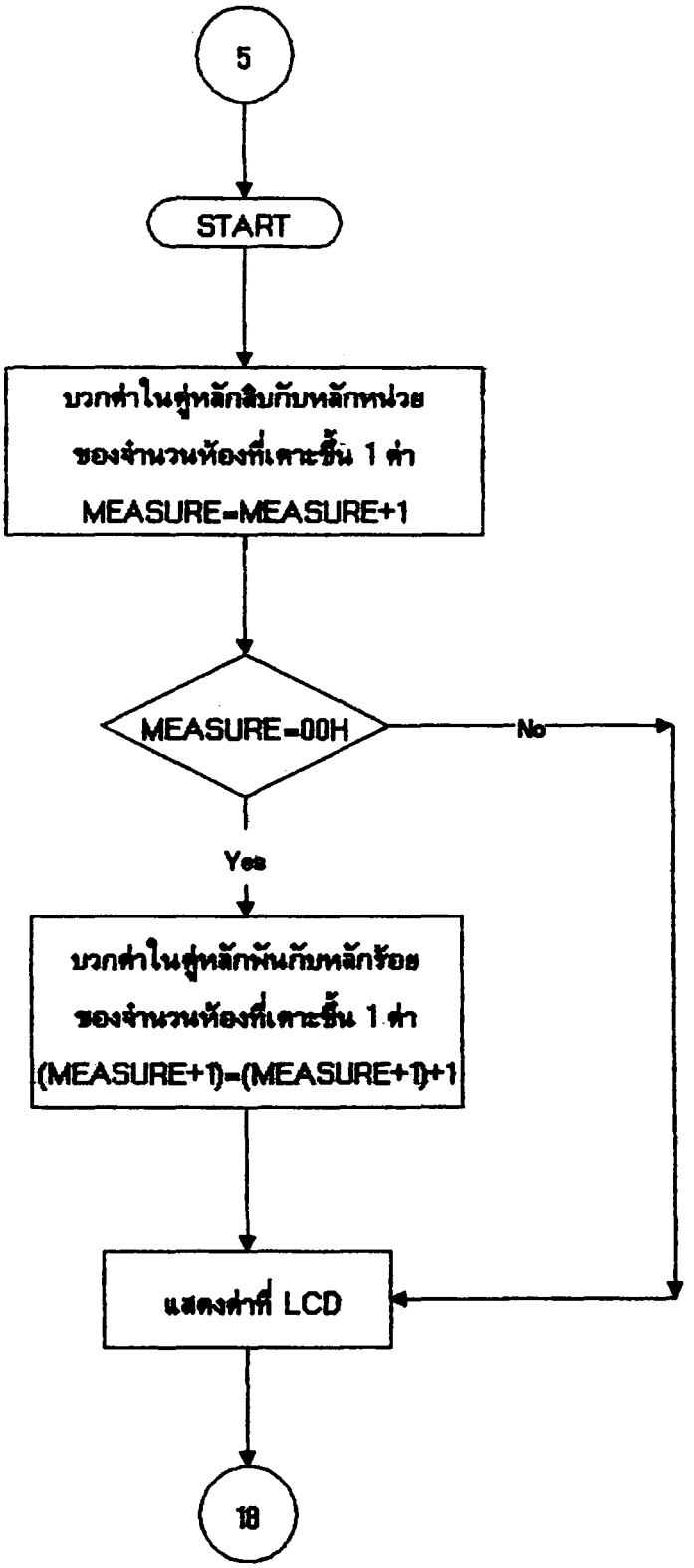

#
แสดงค่า VOLUME
20
กดคีย์ + ?
Yes
10
NO
กดคีย์ - ?
Yes
11
NO
No
กดคีย์ ENTER ?
YES
แสดงค่า TEMPO
21
กดคีย์ + ?
Yes
12
No
กดคีย์ - ?
Yes
13
No
No
กดคีย์ ENTER ?
Yes
แสดงค่าที่ตั้งทั้งหมดที่
7-SEG และ LCD
32

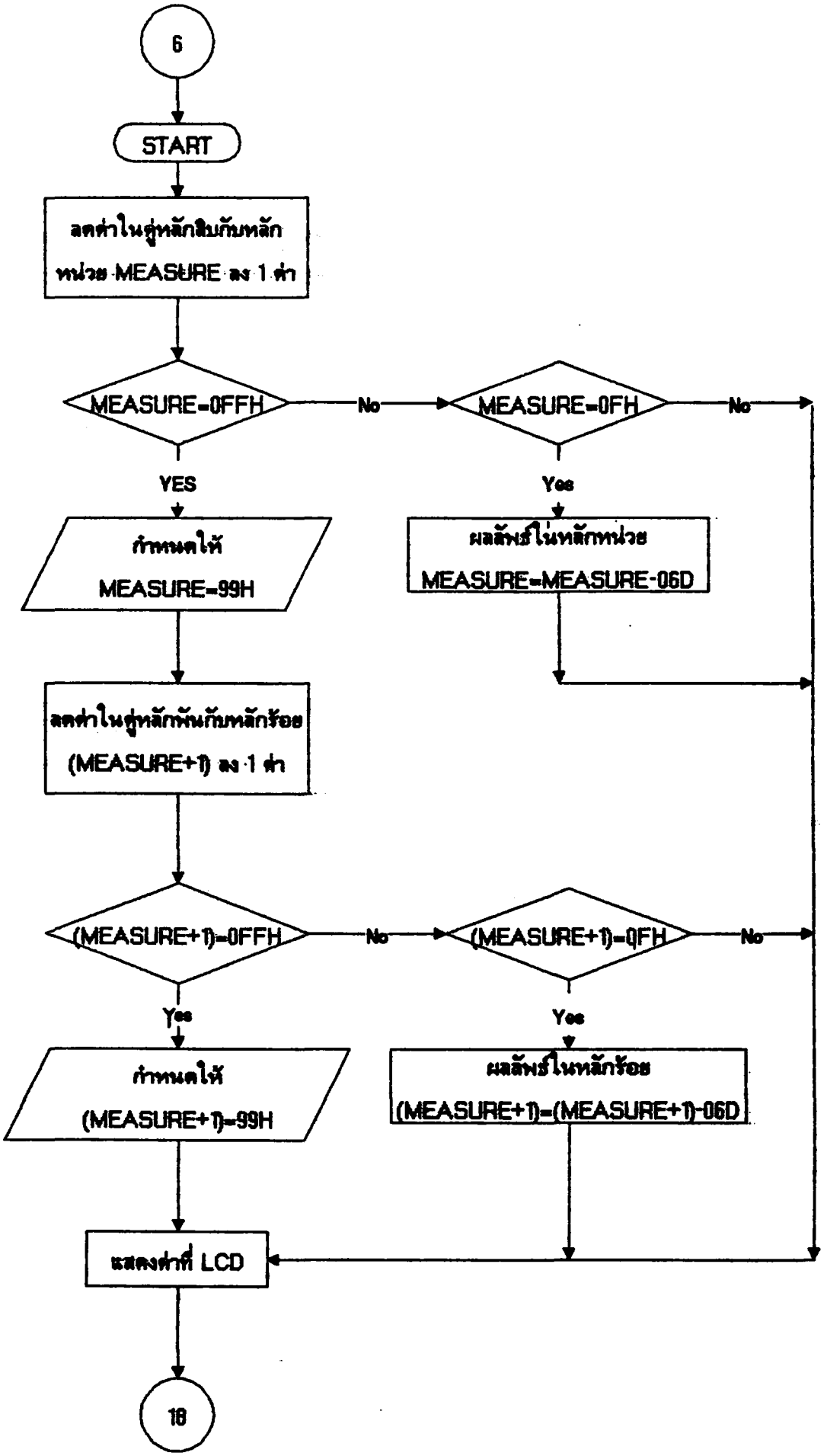

1
START
เลื่อนค่าใน METER ขึ้น 1 บิท
METER=08H
No
Yes
กำหนดค่า
METER=80H
เปรียบเทียบค่าของ
METER
แสดงค่าที่ LCD
16

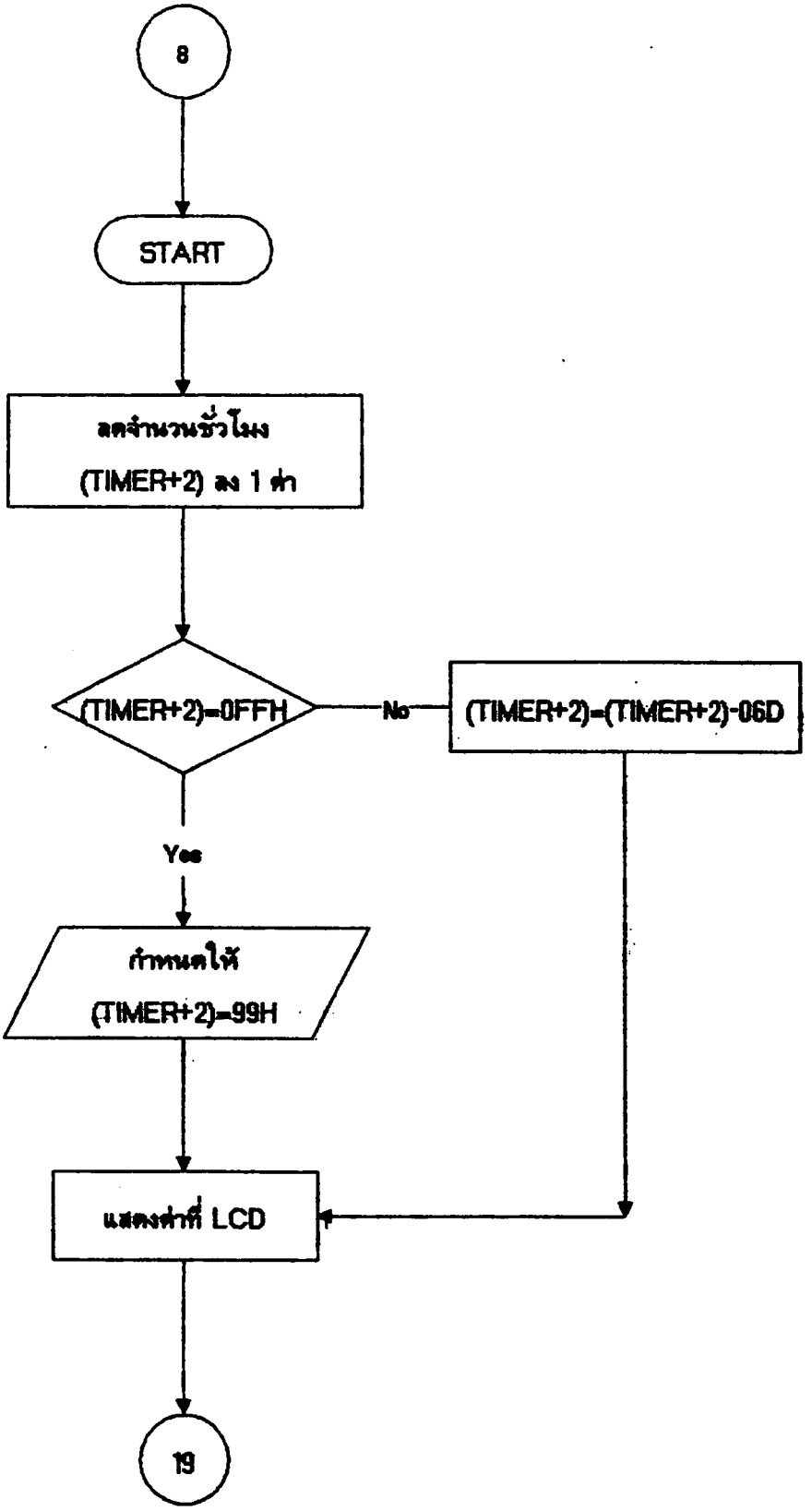

2
START
เลื่อนค่าใน METER
ลง 1 บิท
METER = 01H
No
Yes
กำหนดค่า
METER=10H
เปรียบเทียบค่าของ
METER
แสดงค่าที่ LCD
16

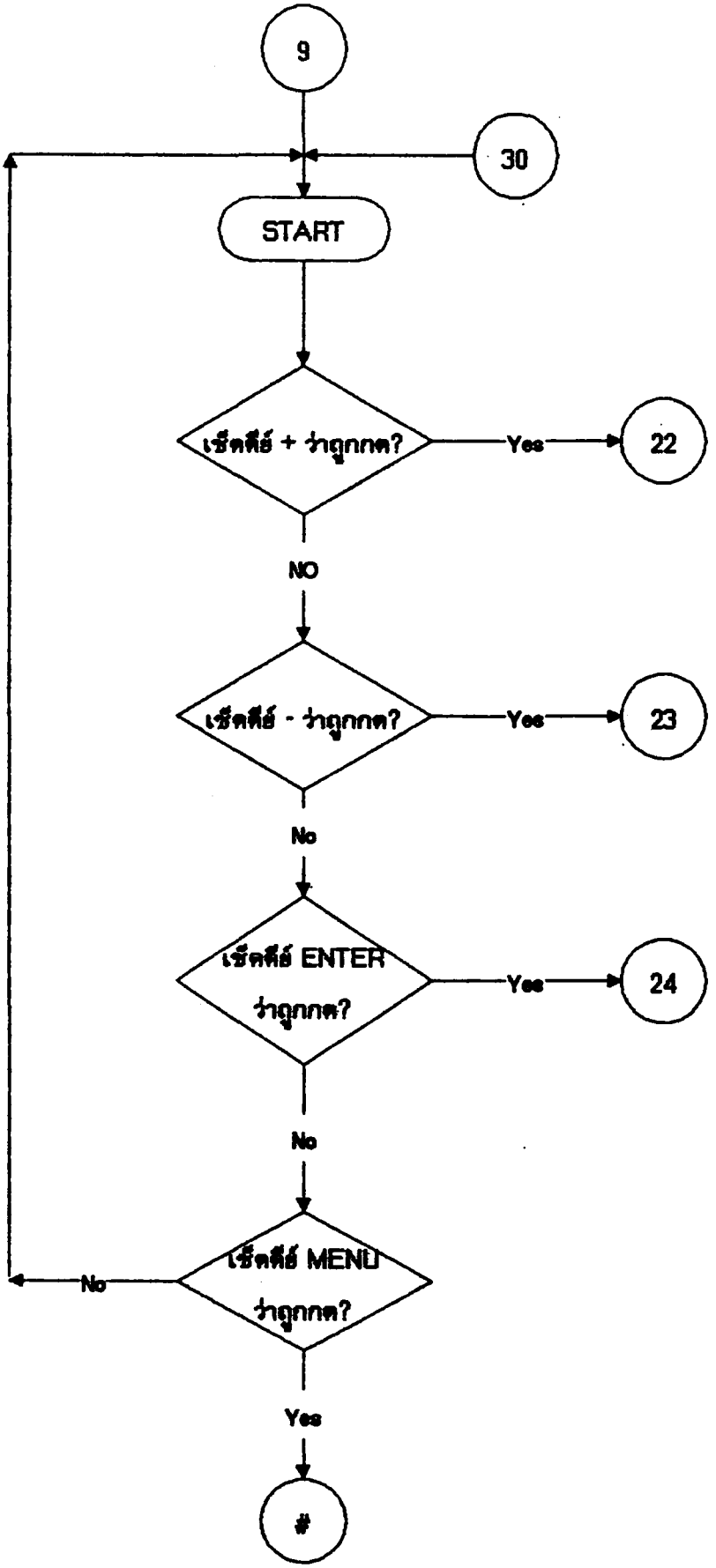

3
START
เลื่อนค่าใน TIMESIGN ลง 1 ค่า
TIMESIGN = 40H
No
Yes
กำหนดให้
TIMESIGN=02H
เปรียบเทียบค่าของ
TIMESIGN
แสดงค่าที่ LCD
17

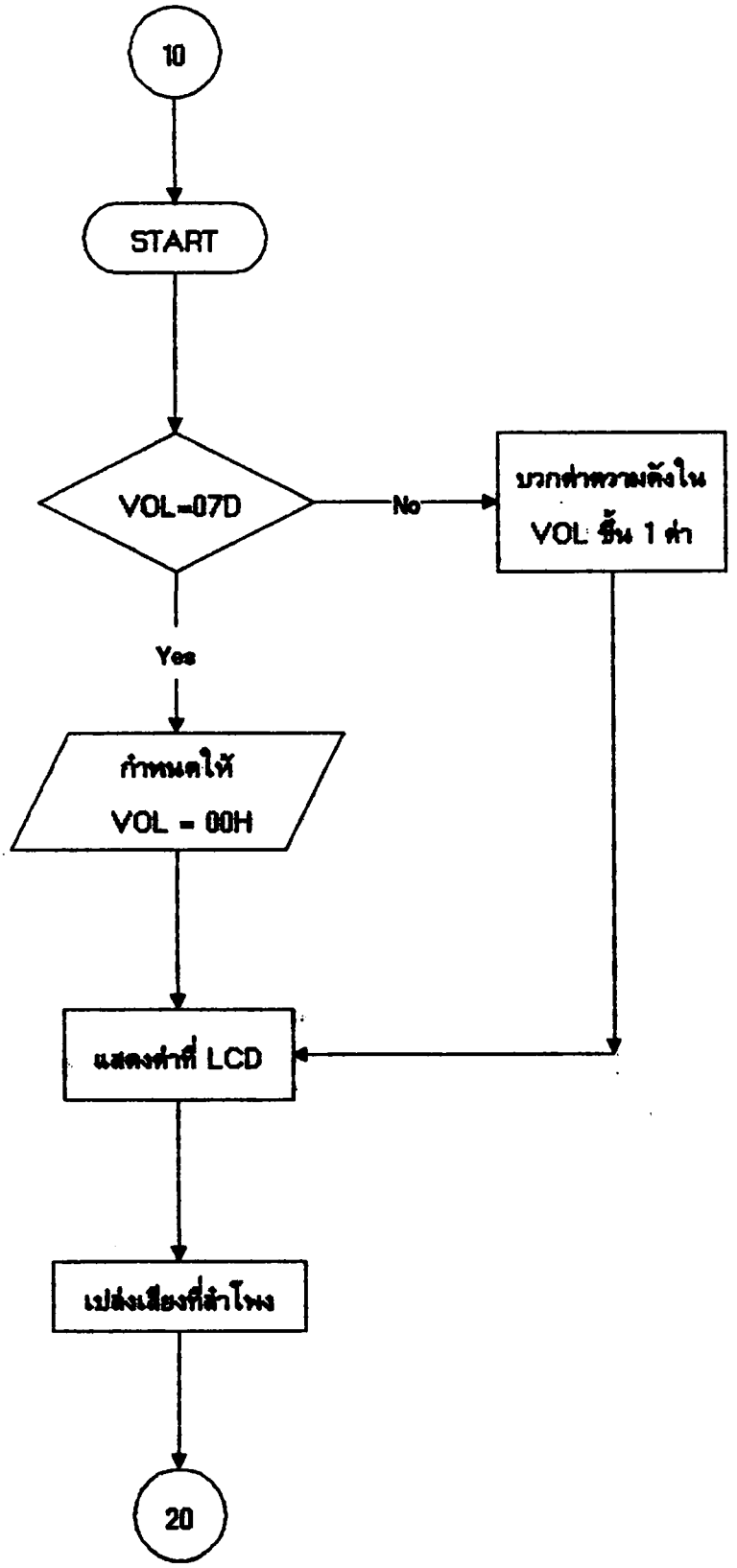

4
START
เลื่อนค่าของ TIMESIGN ขึ้น 1 ค่า
TIMESIGN = 01H
No
Yes
กำหนดให้
TIMESIGN=20H
เปรียบเทียบค่าของ
TIMESIGN
แสดงค่าที่ LCD
17

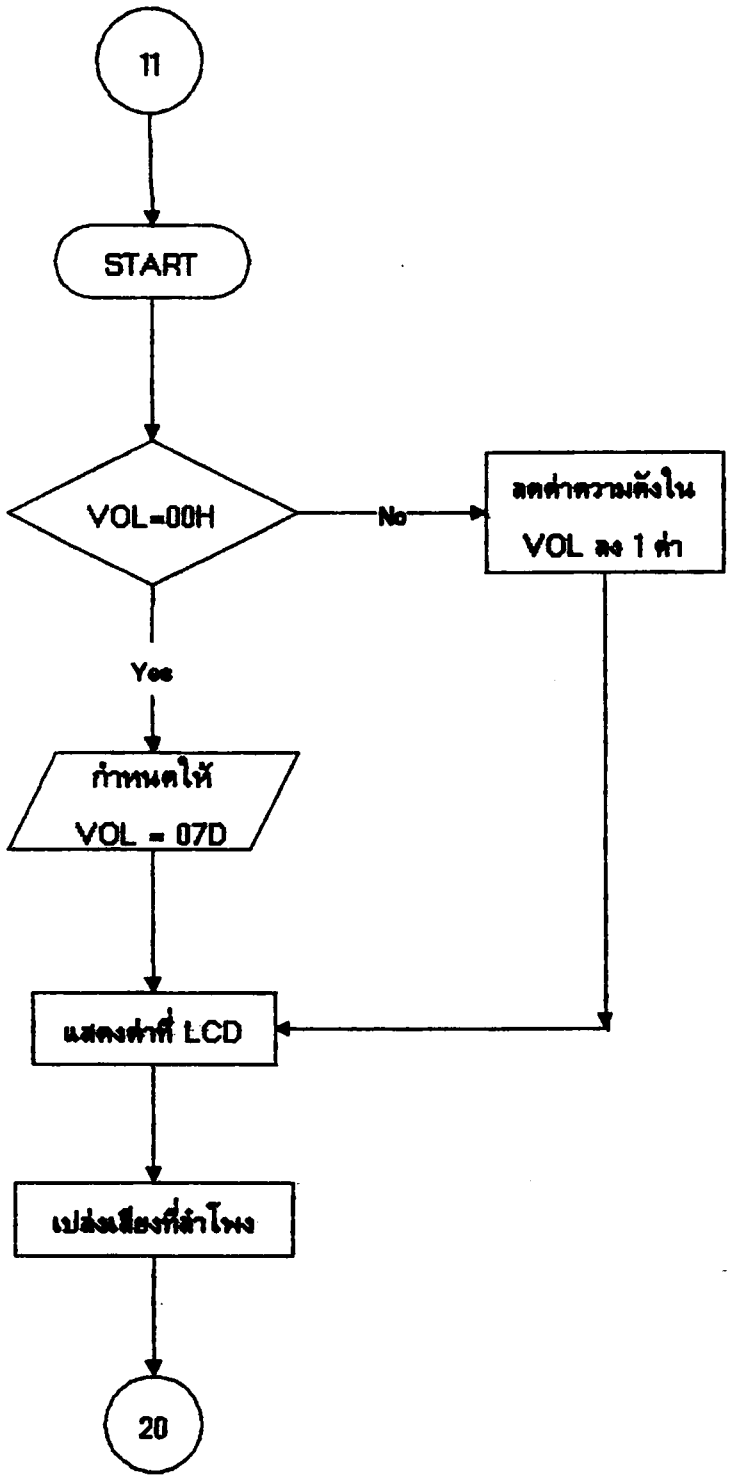

5
START
บวกค่าในตู้หลักสิบกับหลักหน่วย
ของจำนวนห้องที่เตาะขึ้น 1 ค่า
MEASURE=MEASURE+1
MEASURE=00H
No
Yes
บวกค่าในตู้หลักพันกับหลักร้อย
ของจำนวนห้องที่เตาะขึ้น 1 ค่า
(MEASURE+1)=(MEASURE+1)+1
แสดงค่าที่ LCD
18

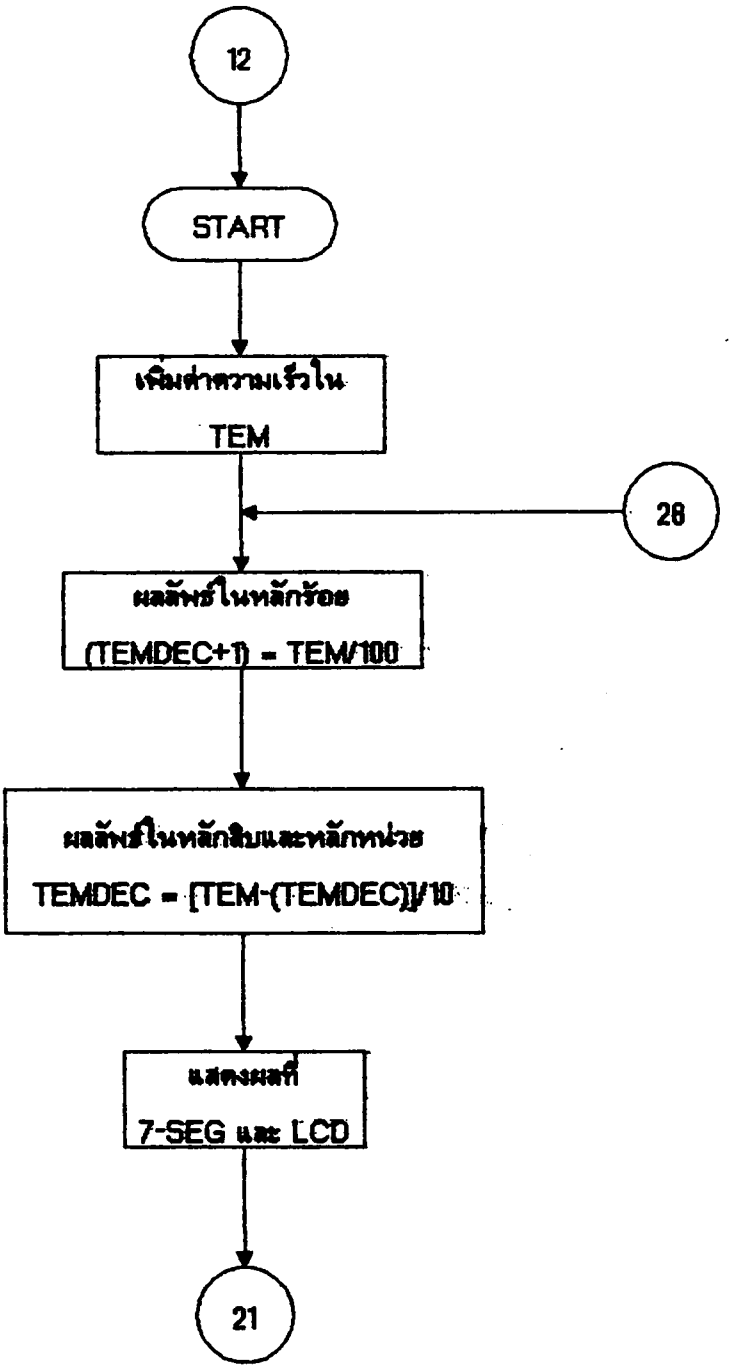

6
START
ลดค่าในตู้หลักสิบกับหลัก
หน่วย MEASURE ลง 1 ค่า
MEASURE=0FFH
No
MEASURE=0FH
No
YES
Yes
กำหนดให้
MEASURE=99H
ผลลัพธ์ในหลักหน่วย
MEASURE=MEASURE-06D
ลดค่าในตู้หลักพันกับหลักร้อย
(MEASURE+1) ลง 1 ค่า
(MEASURE+1)=0FFH
No
(MEASURE+1)=0FH
No
Yes
Yes
กำหนดให้
(MEASURE+1)=99H
ผลลัพธ์ในหลักร้อย
(MEASURE+1)=(MEASURE+1)-06D
แสดงค่าที่ LCD
18

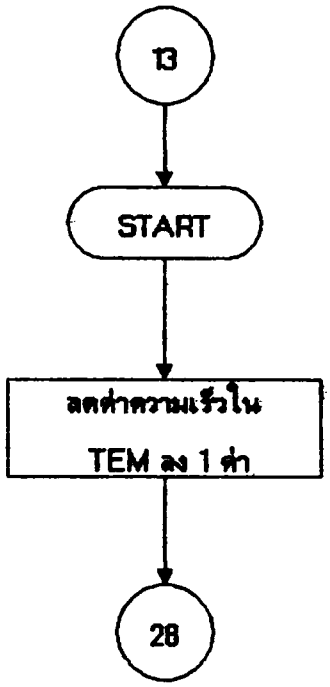

8
START
ลดจำนวนชั่วโมง
(TIMER+2) ลง 1 ค่า
(TIMER+2)=0FFH
No
(TIMER+2)=(TIMER+2)-06D
Yes
กำหนดให้
(TIMER+2)=99H
แสดงค่าที่ LCD
19

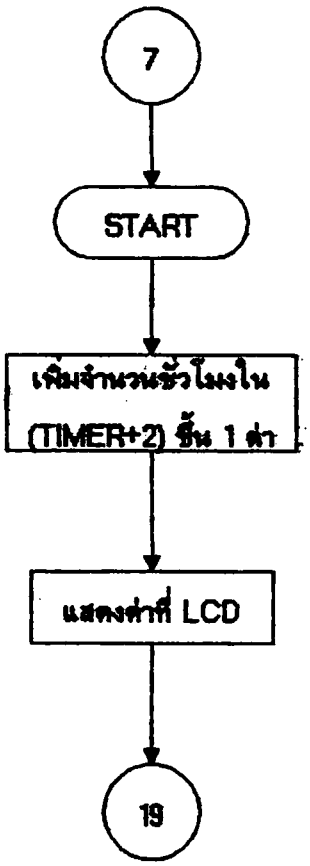

9
30
START
เช็คคีย์ + ว่าถูกกด?
Yes
22
NO
เช็คคีย์ - ว่าถูกกด?
Yes
23
No
เช็คคีย์ ENTER
ว่าถูกกด?
Yes
24
No
เช็คคีย์ MENU
ว่าถูกกด?
No
Yes
#

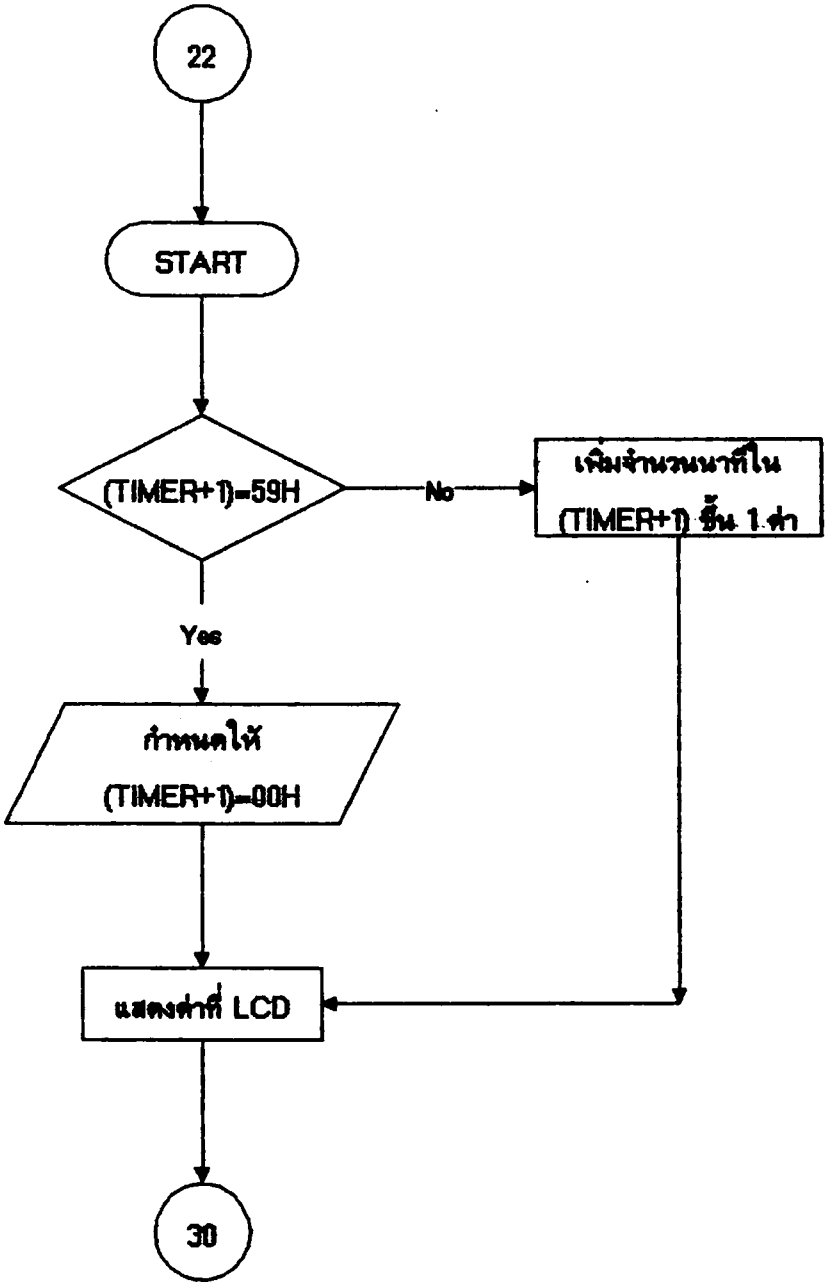

10
START
VOL=07D
No
บวกค่าความดังใน
VOL ขึ้น 1 ค่า
Yes
กำหนดให้
VOL = 00H
แสดงค่าที่ LCD
เปล่งเสียงที่ลำโพง
20

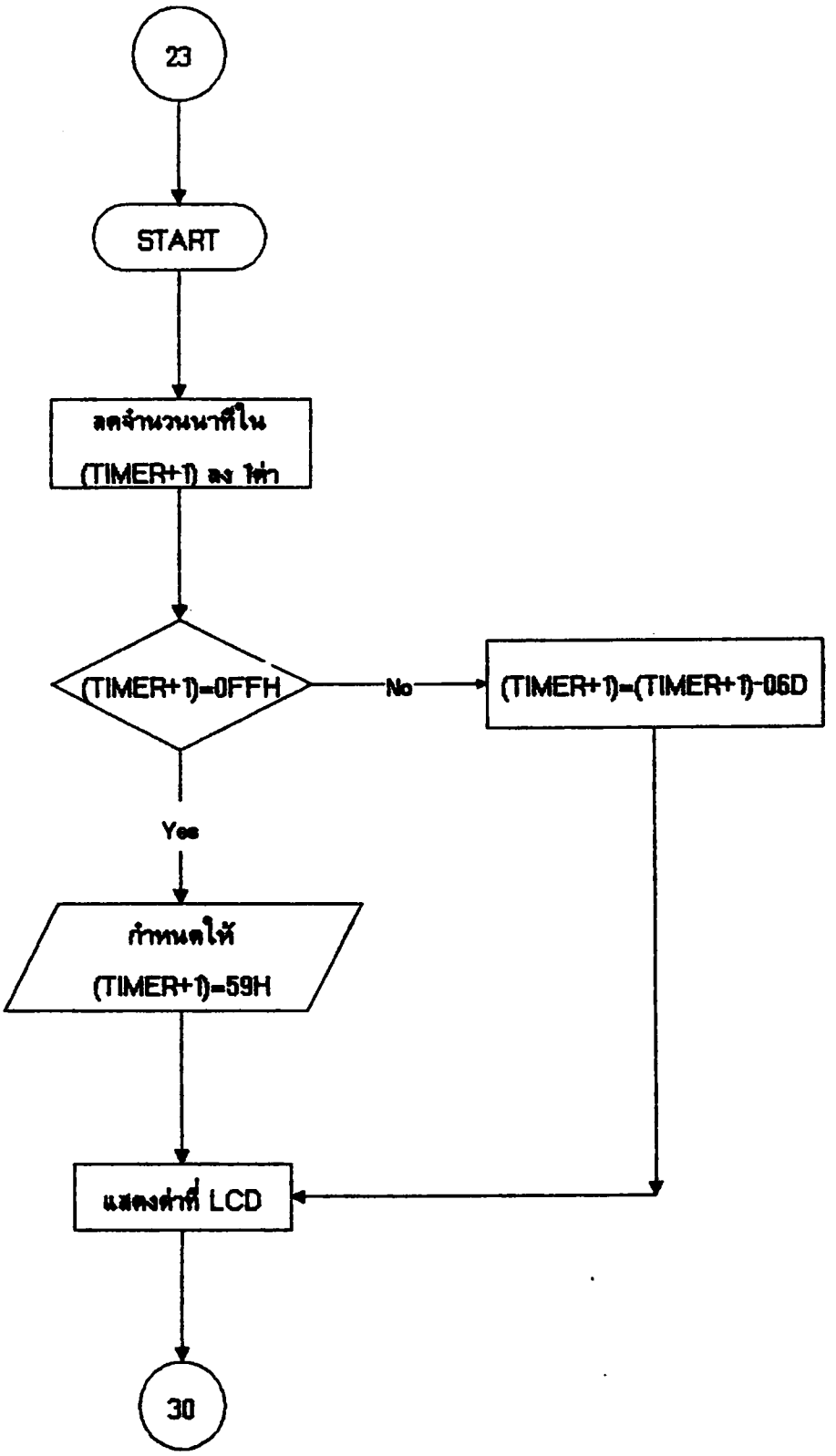

11
START
VOL=00H
No
ลดค่าความดังใน
VOL ลง 1 ค่า
Yes
กำหนดให้
VOL = 07D
แสดงค่าที่ LCD
เปล่งเสียงที่ลำโพง
20

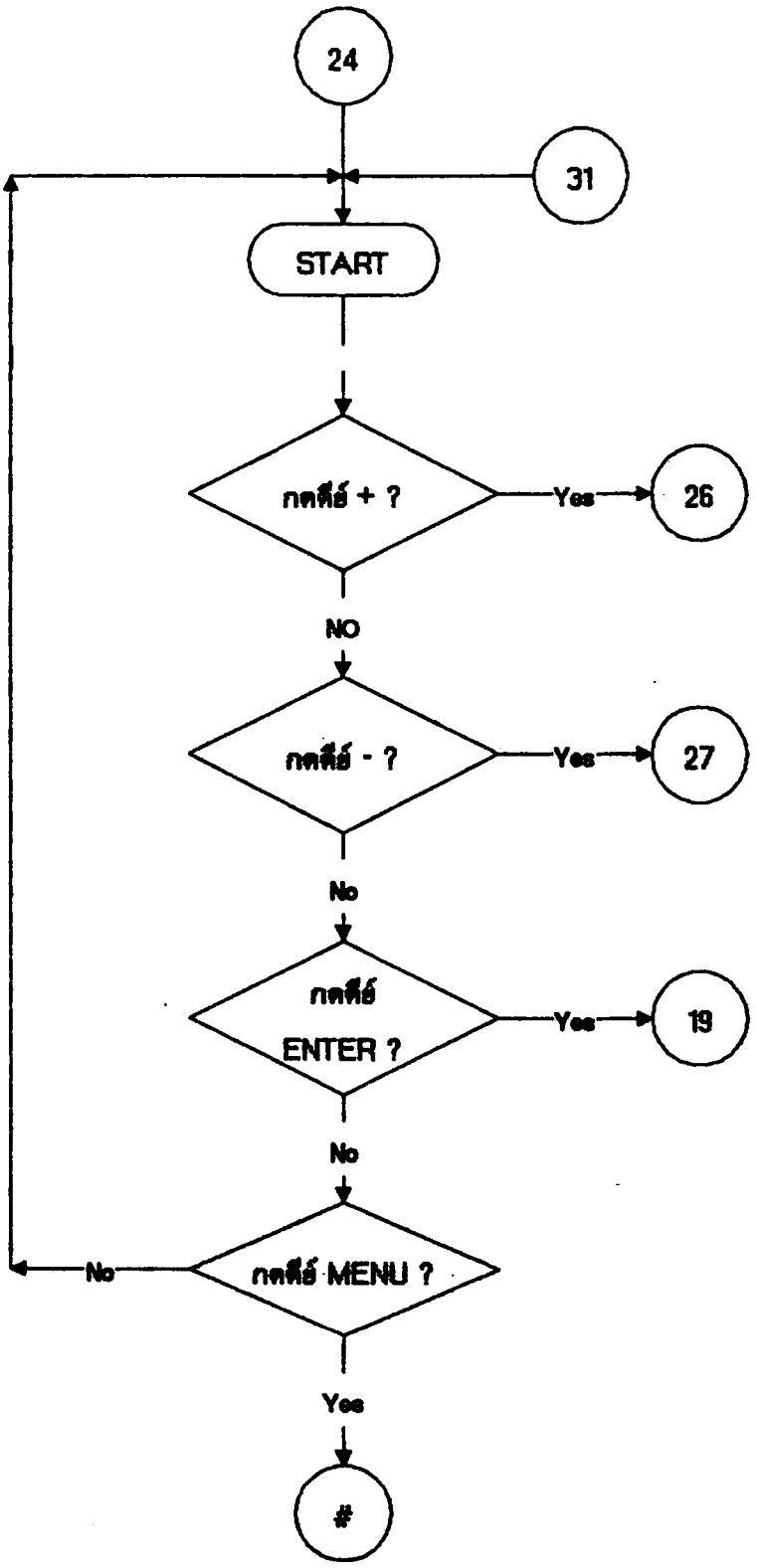

12
START
เพิ่มค่าความเร็วใน
TEM
28
ผลลัพธ์ในหลักร้อย
(TEMDEC+1) = TEM/100
ผลลัพธ์ในหลักสิบและหลักหน่วย
TEMDEC = [TEM-(TEMDEC)]/10
แสดงผลที่
7-SEG และ LCD
21

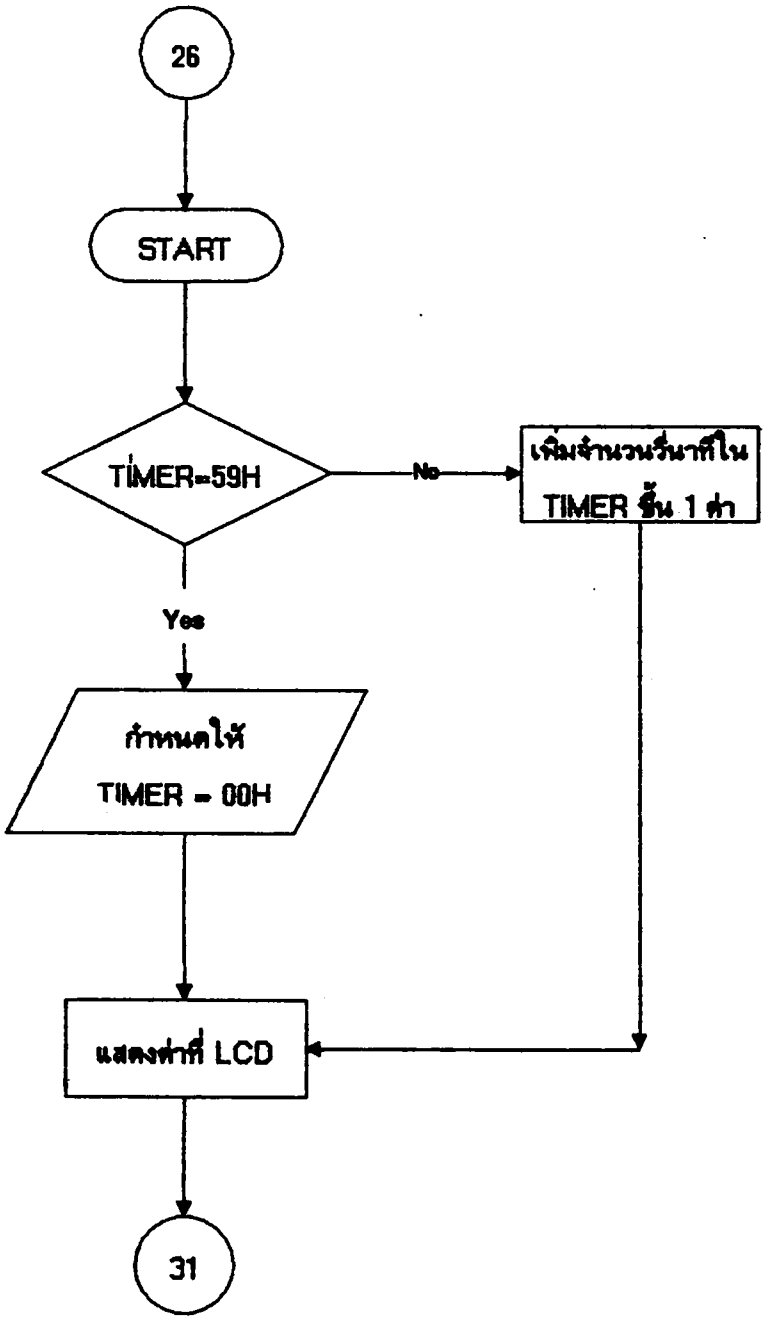

13
START
ลดค่าความเร็วใน
TEM ลง 1 ค่า
28

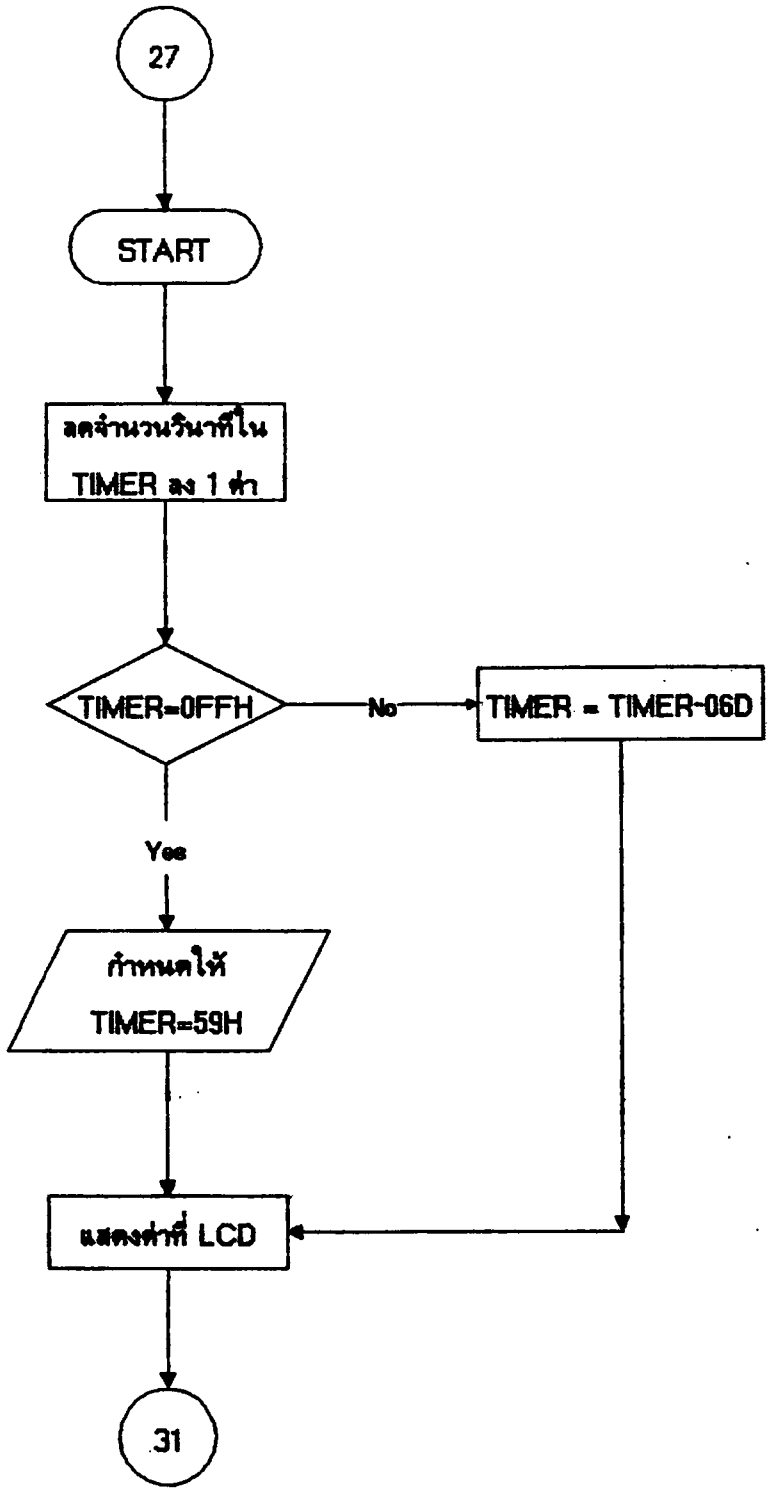

7
START
เพิ่มจำนวนรัวโมงใน
(TIMER+2) ขึ้น 1 ค่า
แสดงค่าที่ LCD
19


22
START
(TIMER+1)=59H
No
เพิ่มจำนวนนาทีใน
(TIMER+1) ขึ้น 1 ค่า
Yes
กำหนดให้
(TIMER+1)=00H
แสดงค่าที่ LCD
30

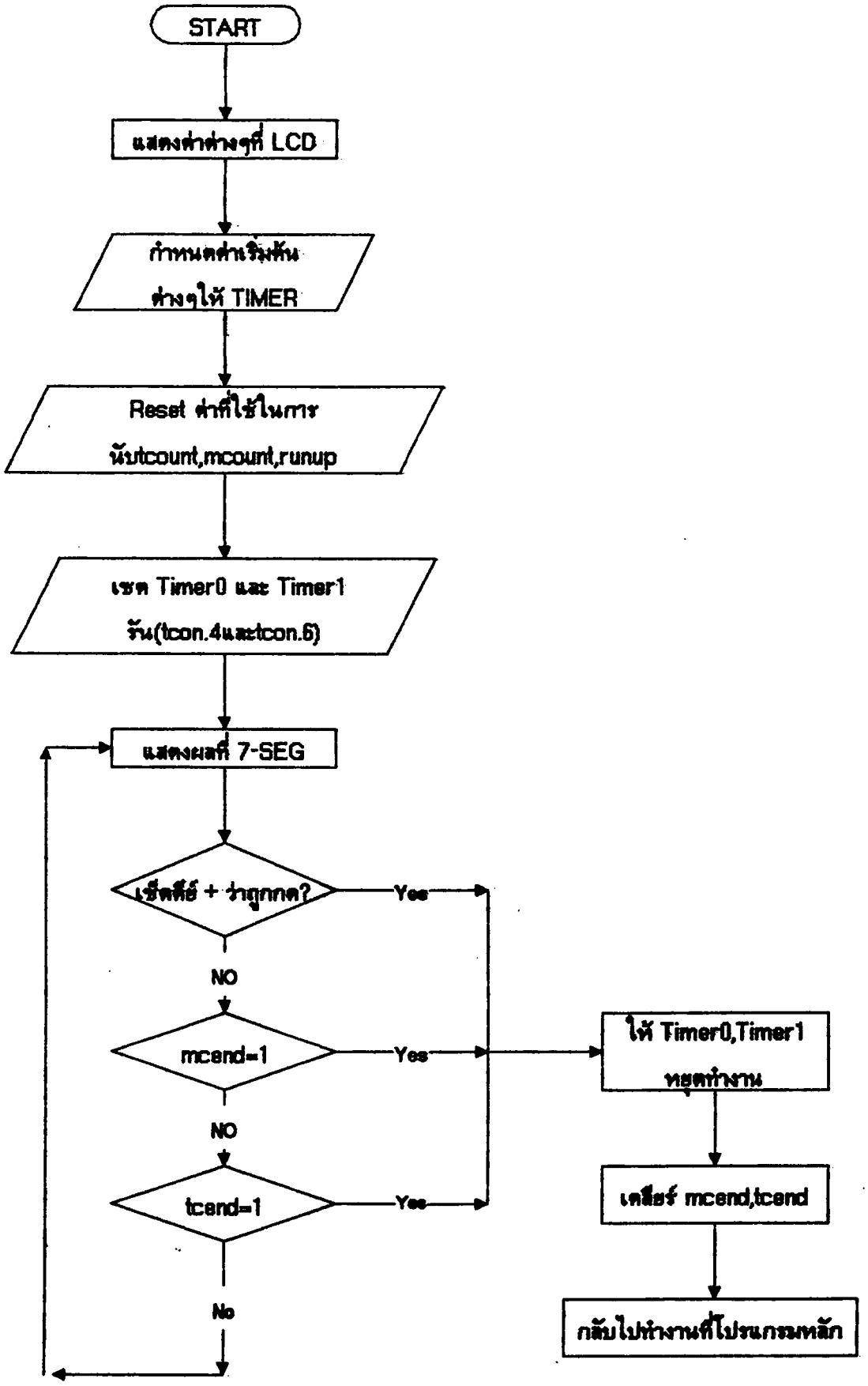

23
START
ลดจำนวนนาทีใน
(TIMER+1) ลง 1ค่า
(TIMER+1)=0FFH
No
(TIMER+1)=(TIMER+1)-06D
Yes
กำหนดให้
(TIMER+1)=59H
แสดงค่าที่ LCD
30


24
31
START
กดที่ + ?
Yes
26
NO
กดที่ - ?
Yes
27
No
กดที่
ENTER ?
Yes
19
No
กดที่ MENU ?
No
Yes
#

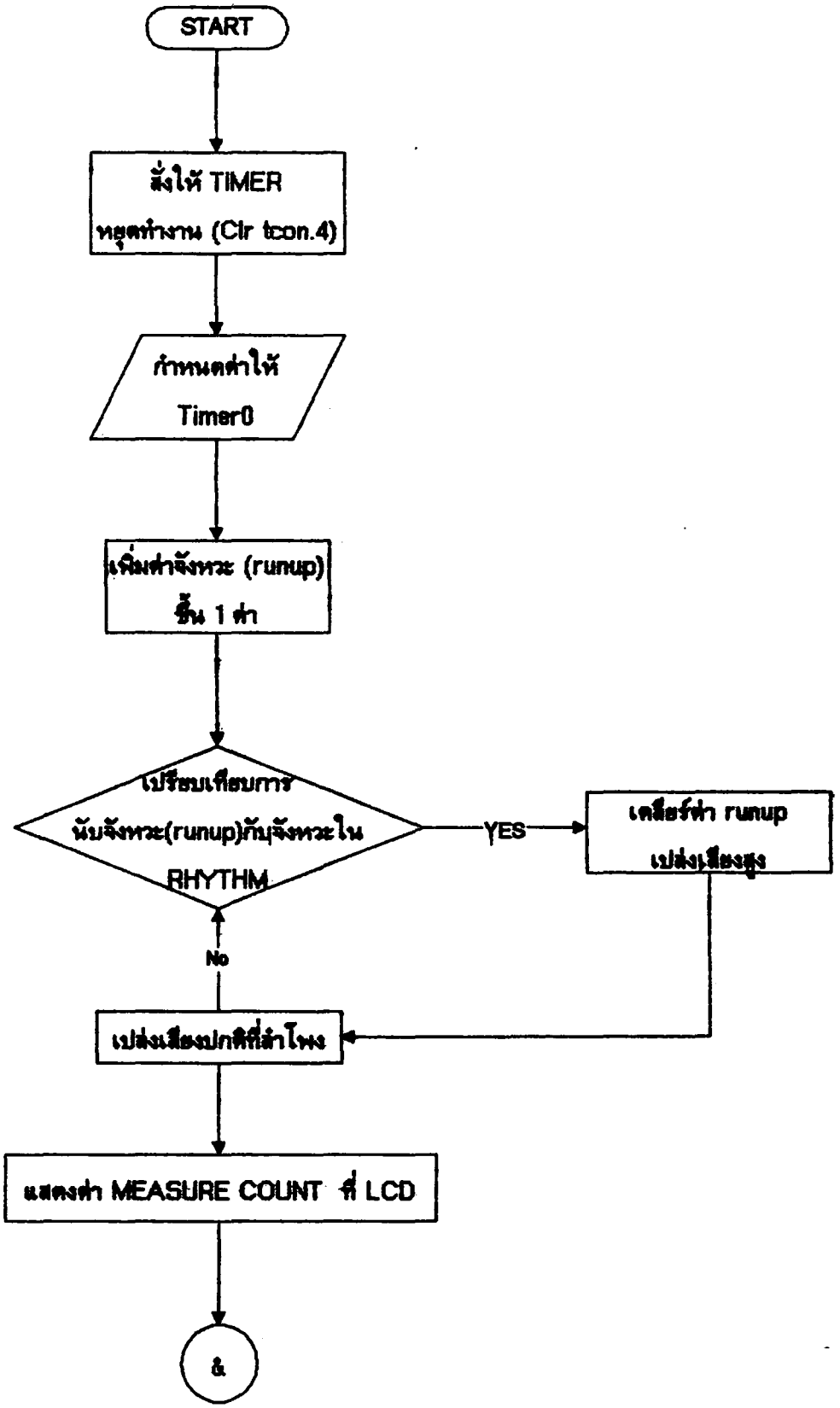

26
START
TIMER=59H
No
เพิ่มจำนวนวินาทีใน
TIMER ขึ้น 1 ค่า
Yes
กำหนดให้
TIMER = 00H
แสดงค่าที่ LCD
31

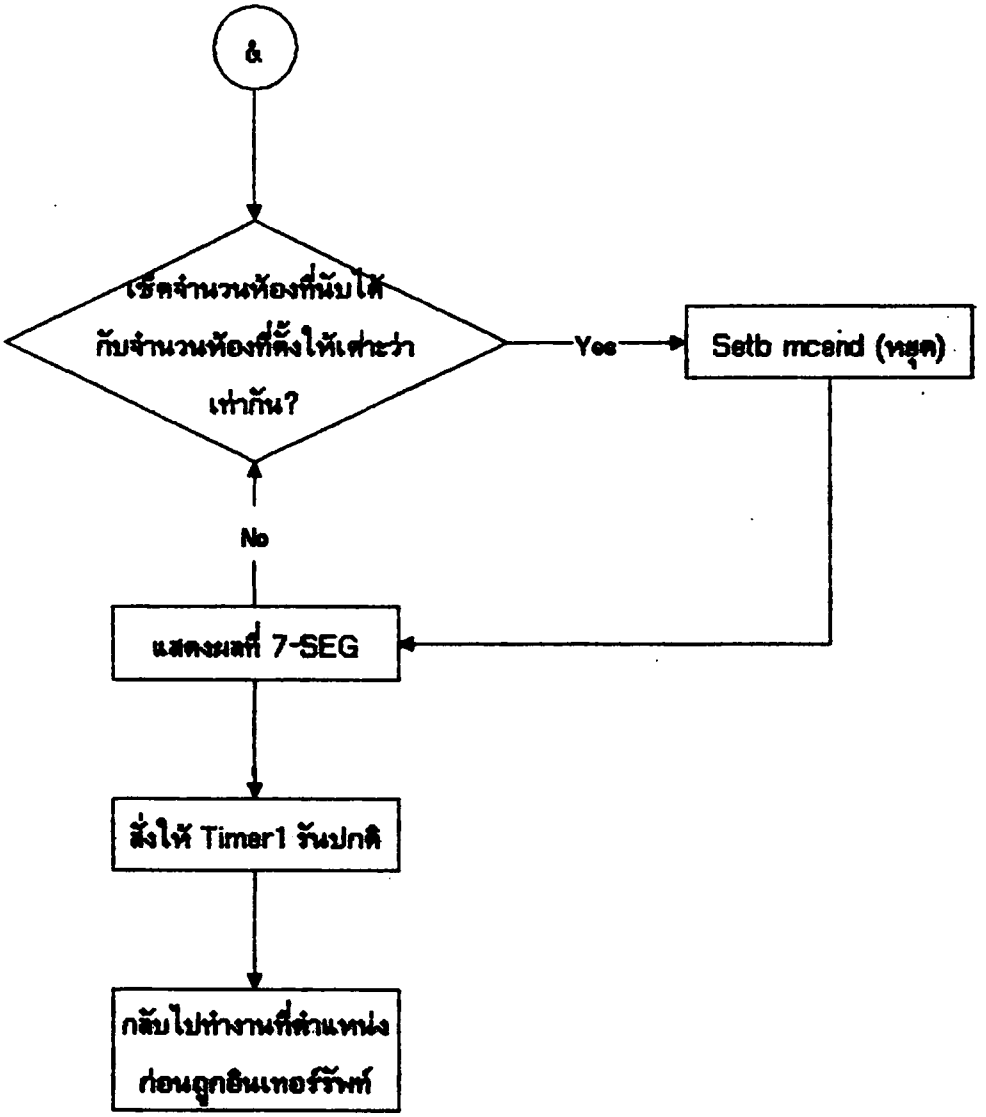

27
START
ลดจำนวนวินาทีใน
TIMER ลง 1 ค่า
TIMER=0FFH
No
TIMER = TIMER-06D
Yes
กำหนดให้
TIMER=59H
แสดงค่าที่ LCD
31

FLOWCHART ขั้นตอนการเจาะ

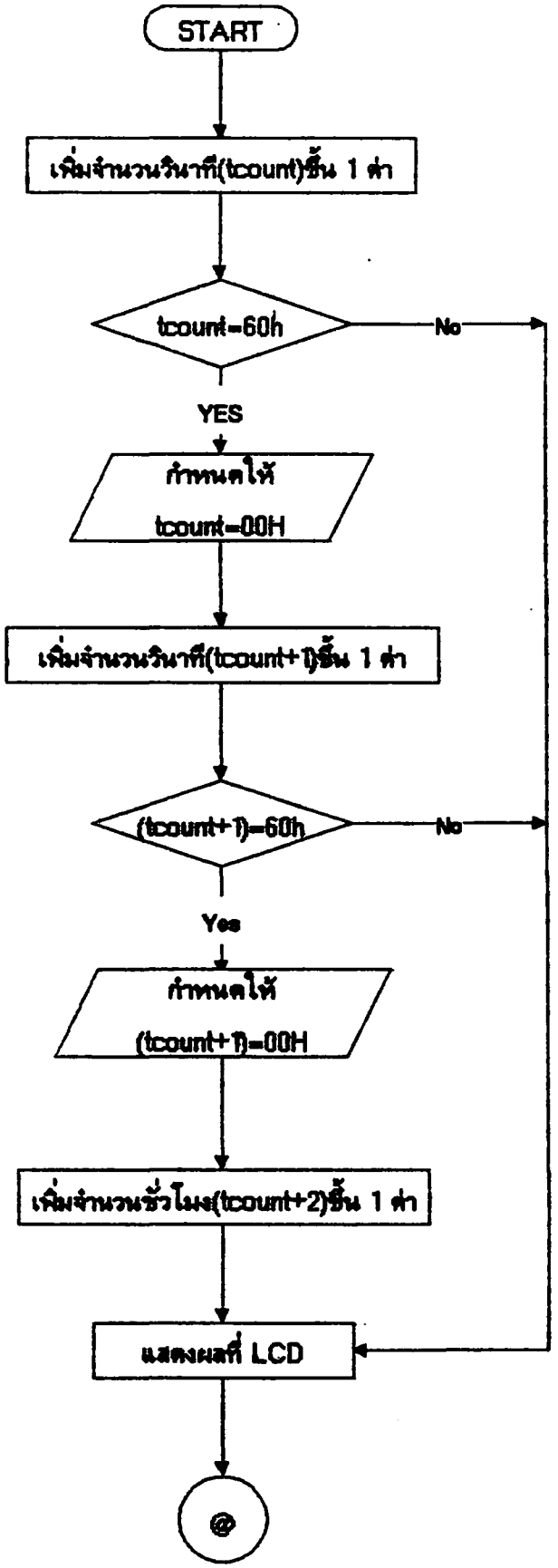


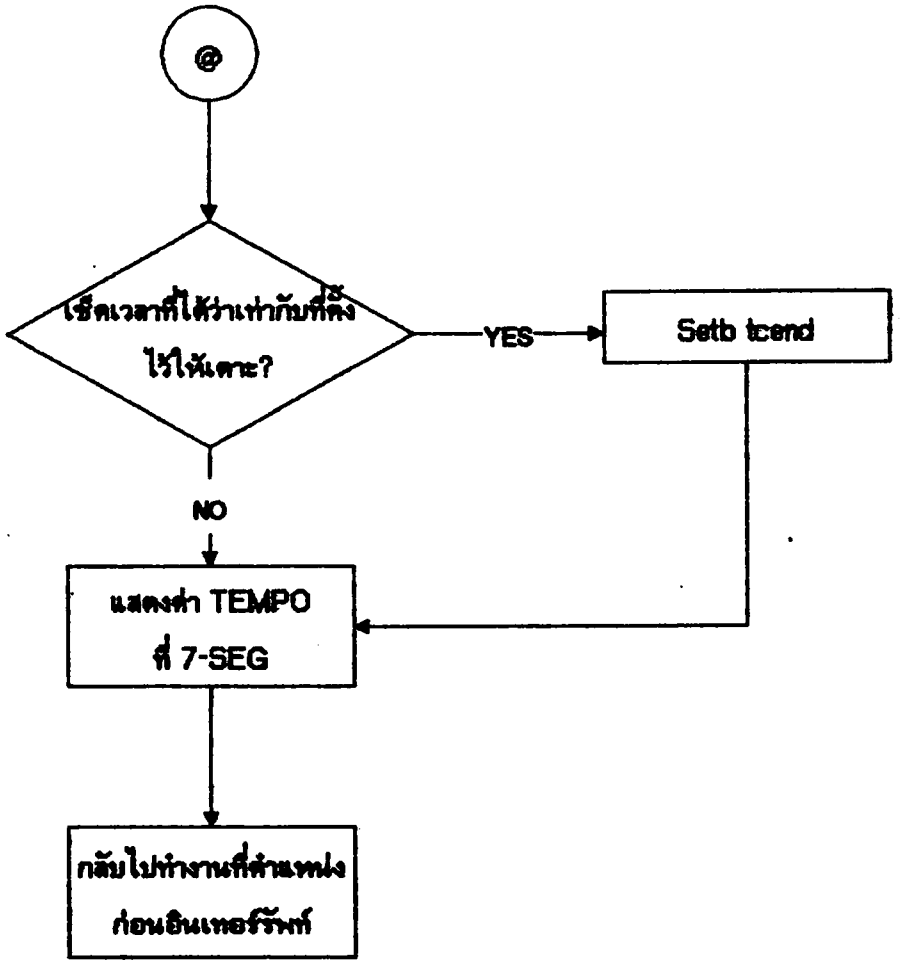
โปรแกรมอินเทอร์รัพท์ TIMER0



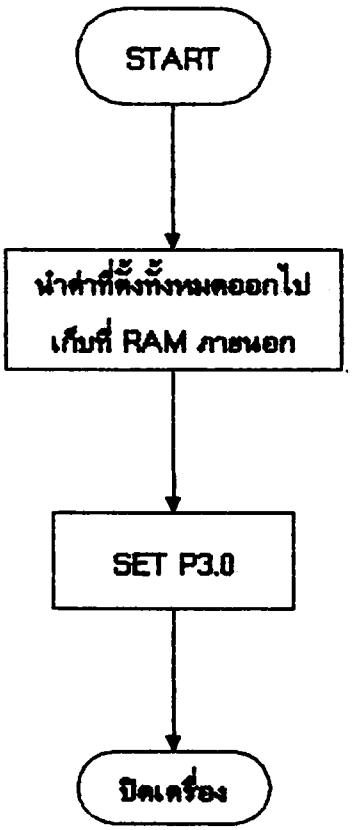

&
เช็คจำนวนห้องที่นับได้
กับจำนวนห้องที่ตั้งให้เซ็นเซ่าว่า
เท่ากัน?
Yes
Setb mcand (หยุด)
No
แสดงผลที่ 7-SEG
สั่งให้ Timer1 รันปกติ
กลับไปทำงานที่ตำแหน่ง
ก่อนถูกอินเทอร์รัพท์

โปรแกรมอินเทอร์รัพท์ TIMER1




@
เช็คเวลาที่ได้ว่าเท่ากับที่ตั้ง
ไว้ให้นเตาะ?
YES
Setb tcend
NO
แสดงค่า TEMPO
ที่ 7-SEG
กลับไปทำงานที่ตำแหน่ง
ก่อนอินเทอร์รัพท์

FLOWCHART สั่งปิดเครื่อง



บทที่ 5
สรุปการทำงาน

เครื่องคาะจังหวะสำหรับนักดนตรีนี้สามารถทำงานได้ตามที่นักดนตรีต้องการจริง ซึ่งเป็นประโยชน์ต่อวงการดนตรีเป็นอย่างมาก โดยผู้ที่สนใจในดนตรี สนใจที่จะเล่นดนตรีสามารถใช้เครื่องนี้ฝึกฝนทักษะต่าง ๆ ให้เกิดความชำนาญ เพื่อเป็นนักดนตรีที่ดีต่อไปได้ ส่วนนักดนตรีมืออาชีพนั้นสามารถใช้งานเครื่องเคาะจังหวะนี้ให้เหมาะสมกับงานที่ตนเองกำลังสร้างสรรอยู่และเกิดประโยชน์อย่างสูงสุดได้

โครงงานพิเศษชิ้นนี้สร้างขึ้นด้วยการรวบรวมความรู้ทางด้านดนตรี และอิเล็กทรอนิกส์เข้าด้วยกัน จากการศึกษาเรื่องราวต่าง ๆ ตลอดเวลาที่ได้รับการศึกษาเล่าเรียนที่ภาควิชาฟิสิกส์ประยุกต์นี้

การทำโครงงานชิ้นนี้แสดงให้เห็นว่า สิ่งต่าง ๆ ที่วนเวียนอยู่ในชีวิตประจำวันของเรานั้น เราสามารถสร้างหรือพัฒนาให้เหมาะสมกับความต้องการของผู้ใช้ได้อย่างมีประสิทธิภาพ

5.1 การปรับปรุง

1. ปรับปรุงส่วนของหน่วยความถี่ฐานให้มีความแม่นยำตลอดย่านความถี่ที่ต้องการ เพื่อให้ได้จังหวะที่แม่นยำขึ้น
2. ปรับปรุงให้สามารถเคาะจังหวะได้มากรูปแบบขึ้น
3. ปรับปรุงให้สามารถควบคุมการเคาะจังหวะได้ด้วยเครื่องคอมพิวเตอร์หรืออุปกรณ์ที่สามารถติดต่อแบบมิดี้ได้
4. ปรับปรุงส่วนเปล่งเสียงให้มีคุณภาพดีที่สุด

หนังสืออ้างอิง

Pen R., Introduction to MUSIC, McGraw-Hill, Singapore, 1992

Intel Corporation, MCS®-51 Architectural Overview

Ayala K., The 8051 Microcontroller Architecture, Programming, and Applications, Technical Texts, Inc, USA, 1991

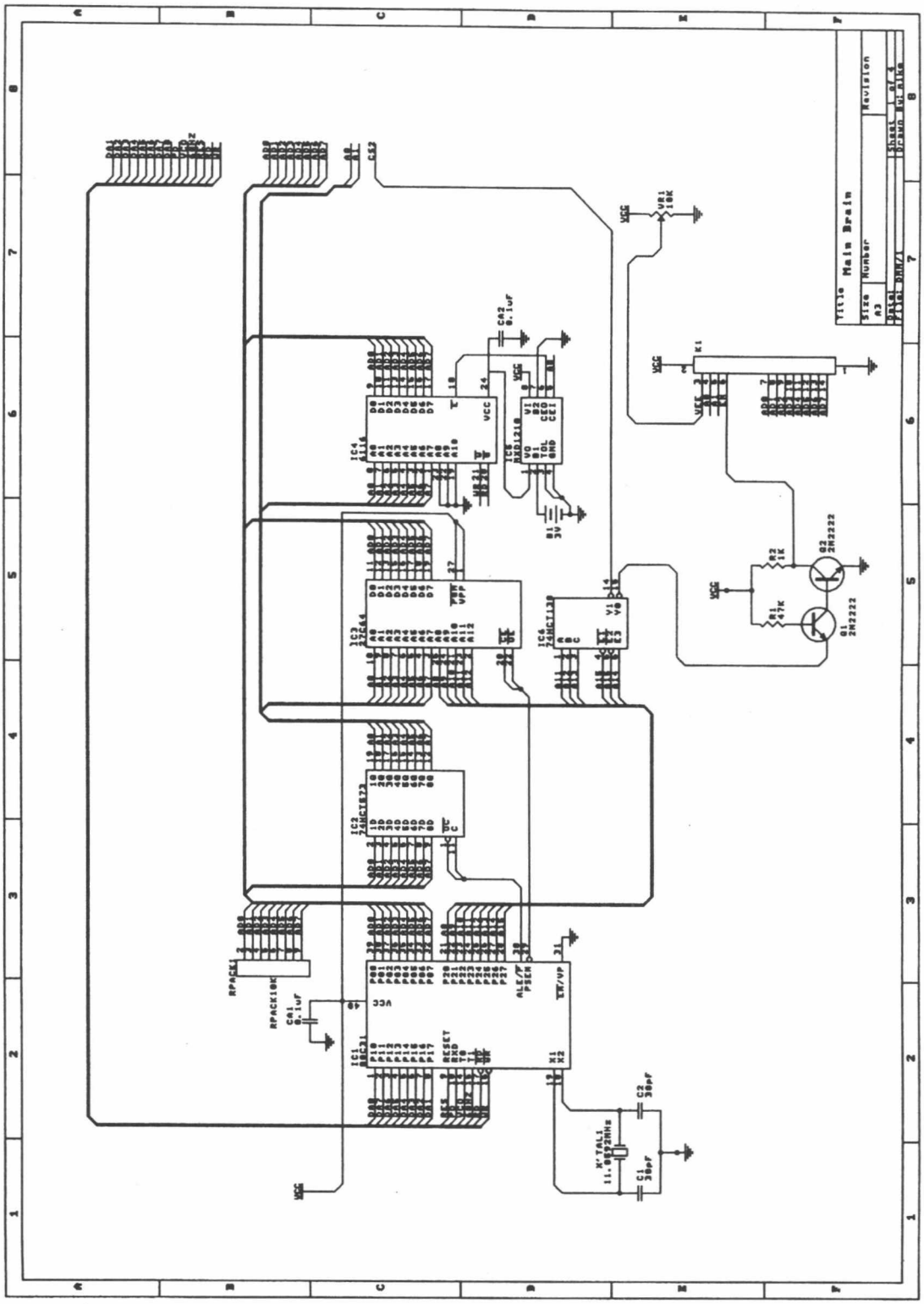
Horowitz P. and Hill W., The art of electronics Second Edition, Cambridge university press, 1989

สุเจตน์ จันทรังษี, ไมโครโปรเซสเซอร์ชิพเดี่ยว 8051, วิทยาลัยมหานคร, พศ. 2535

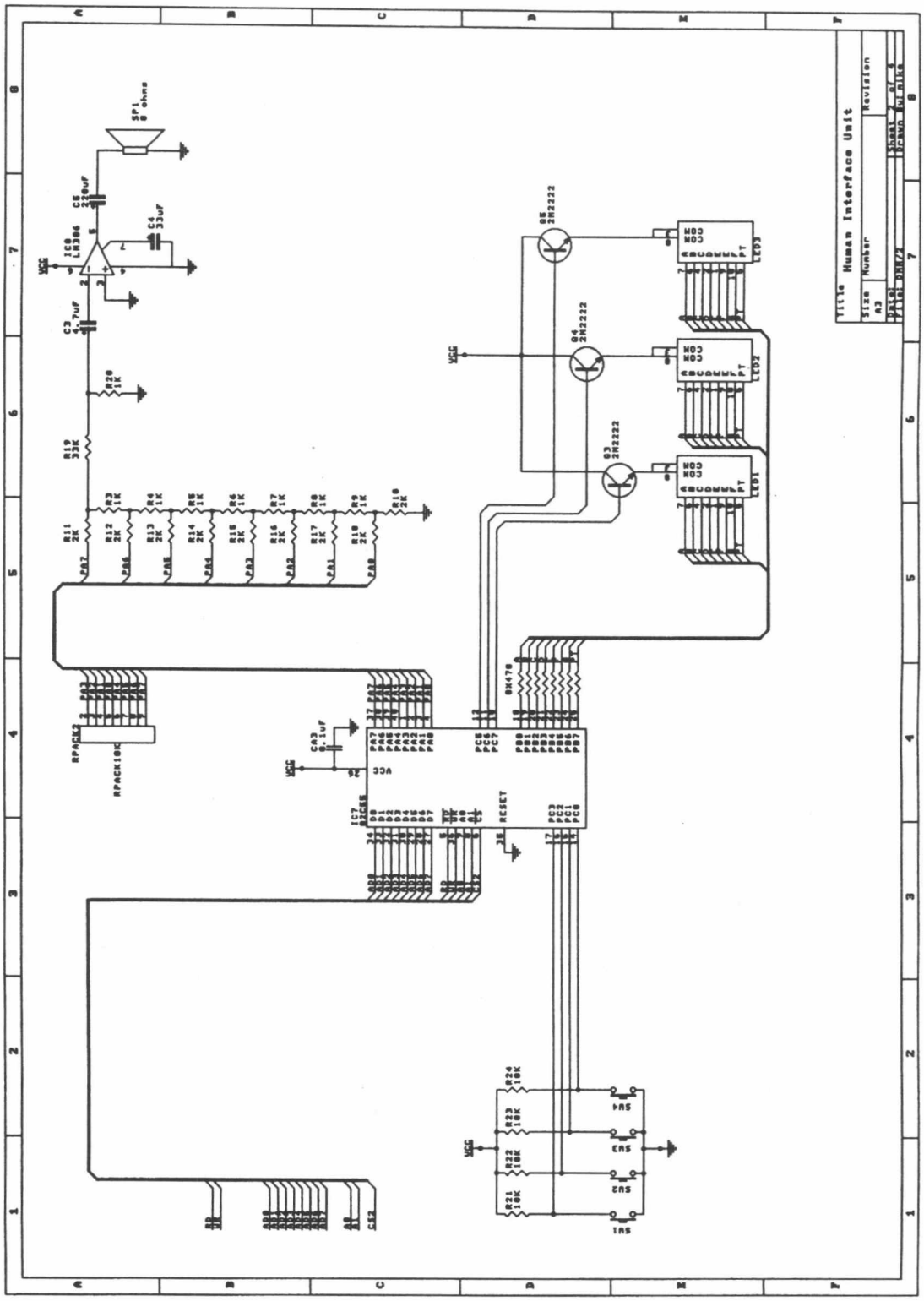
กอบกิจ เติมผาติ เกรียงศักดิ์ บุญเสริมวงศ์ และ ประสิทธิ โรจนนาวิน, ET BOARD ไมโครคอมพิวเตอร์แผ่นพิมพ์เดี่ยว, วารสาร เซมิคอนดักเตอร์ อิเลคทรอนิคส์ ฉบับที่ 66, พศ. 2528

สมยศ ดิษริยะกุล, สร้างวงจรกำเนิดสัญญาณความถี่สำหรับนาฬิกาดิจิตอล, วารสาร คอมพิวเตอร์ อิเลคทรอนิคส์ เวิลด์ ฉบับที่ 89, พศ. 2527

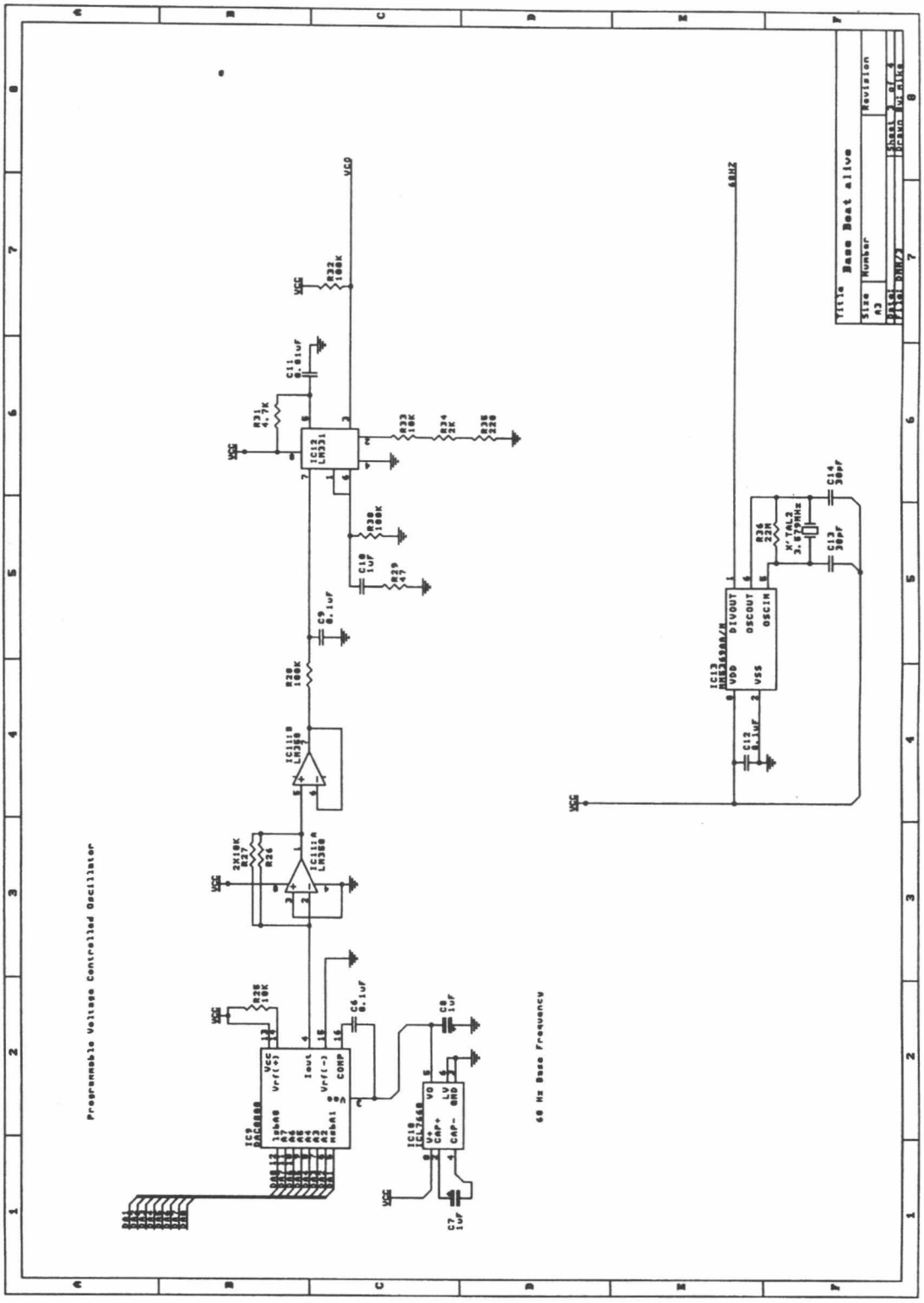
ภาคผนวก ก
ฮาร์ดแวร์ของระบบ



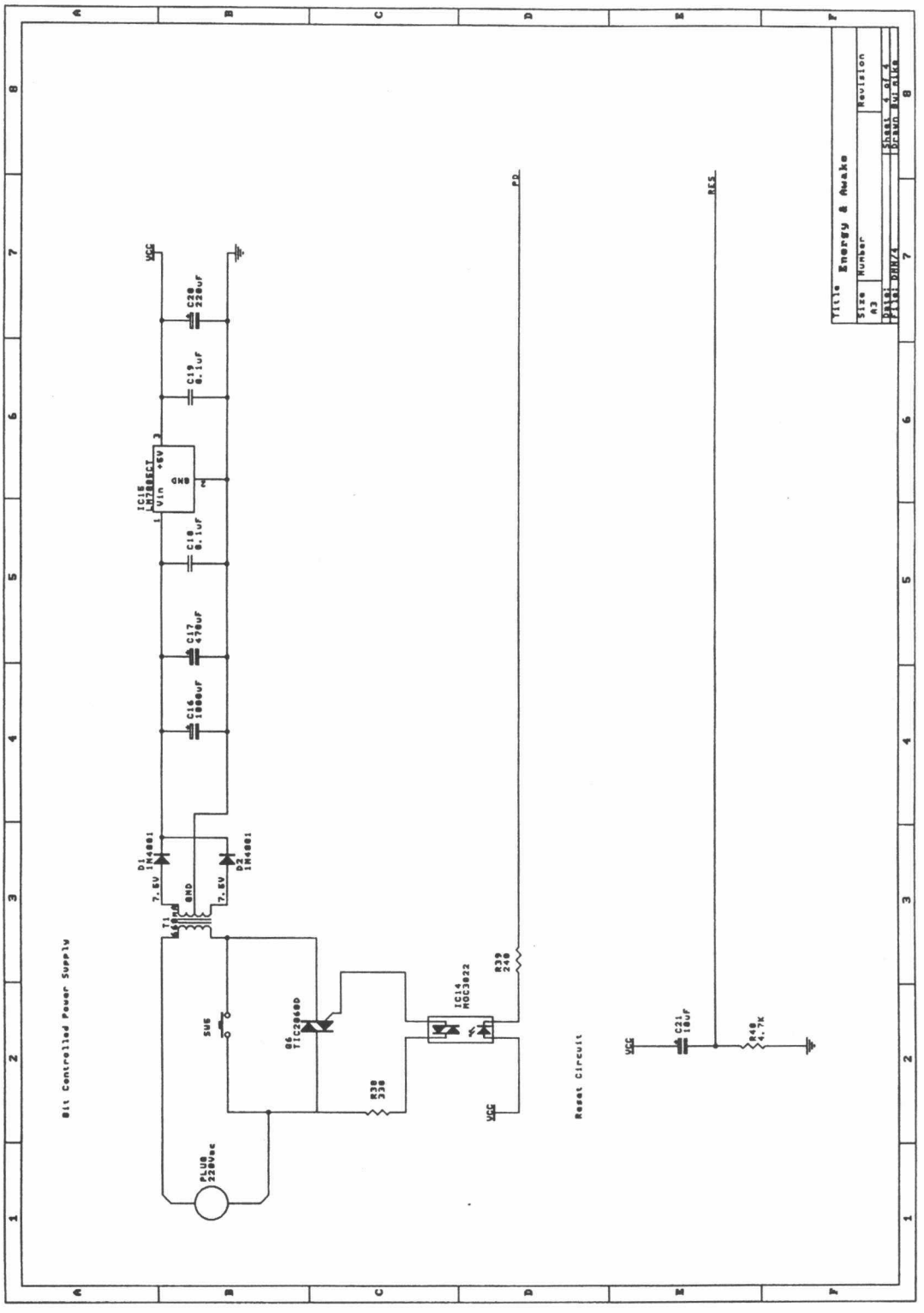
รูป ก.1 หน่วยควบคุมขนาดเล็ก



รูป ก.2 หน่วยติดต่อกับผู้ใช้



รูป ก.3 หน่วยความถี่ฐาน



รูป ก.4 หน่วยจ่ายพลังงานและส่วนรีเซ็ตเมื่อเปิดครื่อง

ภาคผนวก ข
ซอฟท์แวร์ของระบบ

```
;*****************************************************************************
;*                          DIGITAL METRONOME                                *
;*                                BY                                         *
;*              Thanapat    Chayakul            34504016                    *
;*              SUMANEE     VIWATTANATEERAKUL   34504042                    *
;*****************************************************************************

              org     0000h

divl          equ       32h
divh          equ       33h
runup         equ       34h
sndr0         equ       35h
sndr1         equ       36h
sndr2         equ       37h
vol           equ       39h             ;sound level
tem           equ       40h             ;tempo
temdec        equ       41h             ;keep tempo(BCD)value
tcount        equ       43h
dispbuf       equ       45h             ;keep value for send to 7-seg
mcount        equ       46h
meter         equ       48h             ;keep meter
timesign      equ       49h             ;keep code of timesignature
volume        equ       4ah
rul           equ       50h
no.c          equ       51h             ;number character for lcd
rhythm        equ       52h             ;keep rhythm for compare
measure       equ       54h             ;center  send Measure count
connect       equ       56h             ;center  send beat value
timer         equ       57h             ;center  send timer
oflag         equ       60h             ;Batt off flag
tcend         equ       7dh             ;time count end
mcend         equ       7eh             ;measure count end
instw         equ      4000h            ;write instruction
dataw         equ      4001h            ;write data
statr         equ      4002h            ;read  instruction
datar         equ      4003h            ;read  data
porta         equ      4800h            ;speaker
portb         equ      4801h            ;7-seg
portc         equ      4802h            ;high byte for 7-seg
ctrl          equ      4803h            ;control port

;Local   Address 10h,11h,12h,13h,14h,15h,16h,17h,18h,19h,1ah,1bh,1eh
;General register :r5,r6,r7  delay loop

              ;###############
              ;#     MENU    #
              ;###############

              jmp     begin

fill1:        db      00h,00h,00h,00h,00h,00h,00h,00h

              org     000bh
ti0:          jmp     t0serv          ;goto beat

fill2:        db      00h,00h,00h,00h,00h,00h,00h,00h
              db      00h,00h,00h,00h,00h

              org     001bh
ti1:          jmp     t1serv          ;goto time

              org     001eh

begin:        clr     p3.0            ;Awake
              lcall   getdata
```

```
                clr     tcend
                clr     mcend
                mov     dptr,#ctrl        ;8255 initialize
                mov     a,#81h
                movx    @dptr,a
                mov     a,oflag
                cjne    a,#16h,initdata   ;check default
                sjmp    maincont
initdata:       mov     oflag,#16h
                mov     meter,#10h        ;Quarter
                mov     timesign,#10h     ;4/4
                mov     measure,#00h      ;Measure
                mov     (measure+1),#00h
                mov     timer,#00h        ;Time
                mov     (timer+1),#00h
                mov     (timer+2),#00h
                mov     vol,#07h          ;Speak level
                mov     tem,#3ch          ;Tempo = 60
                lcall   putdata
maincont:       mov     sp,#08h
                mov     temdec,#00h
                mov     (temdec+1),#00h
                mov     mcount,#00h
                mov     (mcount+1),#00h
                mov     tcount,#00h
                mov     (tcount+1),#00h
                mov     (tcount+2),#00h
home:           acall   lcd               ;through the eye of god
                lcall   h1txt
                lcall   h2txt
                lcall   showmt
                lcall   showts
                lcall   showtime
                lcall   showmc
                lcall   tm11              ;convert tem to temdec
home2:          mov     p1,tem            ;Send Tempo to VCO
                lcall   seg
                mov     dptr,#portc
                movx    a,@dptr
                anl     a,#0fh
                cjne    a,#0bh,tap1       ;check MENU key pressed?
                acall   delay3
                acall   delay3

                    ;######
                    ;METER
                    ;######

lo:             acall   lcd
                acall   l1

;######################
;+ KEY SCAN SUBROUTINE
;######################

op:             lcall   seg
                mov     dptr,#portc
                movx    a,@dptr
                anl     a,#0fh
                cjne    a,#0eh,k2         ;check  + key pressed?
                acall   delay3
                mov     a,meter
                rr      a
                mov     meter,a
                cjne    a,#08h,k11
                mov     a,#80h
```

```
                mov     meter,a
k11:            acall   q               ;send METER to LCD
                jmp     op

;######################
;- KEY SCAN SUBROUTINE
;######################

k2:             cjne    a,#0dh,keyen    ;check  -key to pressed?
                acall   delay2
                mov     a,meter
                rl      a
                mov     meter,a
                cjne    a,#01h,k21
                mov     a,#10h
                mov     meter,a
k21:            acall   q               ;send METER to LCD
                jmp     op

tap1:           ljmp    chv1            ;display at LCD

;#########################
;ENTER KEY SCAN SUBROUTINE
;#########################
                    ;###############
                    ;TIME SIGNATURE
                    ;###############

keyen:          cjne    a,#07h,op       ;check  + key to pressed?
                acall   delay2
                acall   lcd
                mov     rul,#80h
                mov     no.c,#19
                mov     dptr,#char6
                lcall   lcdld
                mov     a,timesign
                acall   ts22

;######################
;+ KEY SCAN SUBROUTINE
;######################

ken1:           lcall   seg
                mov     dptr,#portc
                movx    a,@dptr
                anl     a,#0fh
                cjne    a,#0eh,ken2     ;check  ENTER key to pressed?
                acall   delay2
                mov     a,timesign
                rl      a
                mov     timesign,a
                cjne    a,#40h,ken      ;compare for check
                mov     a,#02h
                mov     timesign,a
ken:            acall   ts22

;######################
;- KEY SCAN SUBROUTINE
;######################

ken2:           cjne    a,#0dh,kn31     ;check  - key to pressed?
                acall   delay2
                mov     a,timesign
                rr      a
                mov     timesign,a
                cjne    a,#01h,k31      ;compare for check
```

```
              mov     a,#20h
              mov     timesign,a
k31:          acall   ts22

;#########################
;ENTER KEY SCAN SUBROUTINE
;#########################
                  ;#############
                  ;MEASURE COUNT
                  ;#############

kn31:         cjne    a,#07h,ken1       ;check enter key to pressed?
              acall   delay2
              acall   lcd
              mov     rul,#80h
              mov     no.c,#18
              mov     dptr,#char12
              lcall   lcdld
              acall   ld10              ;send value to LCD

;######################
;+ KEY SCAN SUBROUTINE
;######################

km1:          lcall   seg
              mov     dptr,#portc
              movx    a,@dptr
              anl     a,#0fh
              cjne    a,#0eh,km2        ;check + key to pressed?
              acall   delay3
              acall   delay3
              mov     a,measure
              add     a,#01d
              da      a                 ;chang hex in measure to BCD
              mov     measure,a
              cjne    a,#00h,ld17
              mov     a,(measure+1)
              add     a,#01d
              da      a                 ;chang hex in (measure+1) to BCD
              mov     (measure+1),a
              acall   ld10
              sjmp    km1

ld17:         acall   ld10
              sjmp    km1

;######################
;- KEY SCAN SUBROUTINE
;######################

km2:          cjne    a,#0dh,kmsen1     ;kmsen  ;check - key press?
              acall   delay3
              acall   delay3
              dec     measure
              mov     a,measure
              cjne    a,#0ffh,kmm2      ;compare measure for chack
              mov     measure,#99h
              dec     (measure+1)
              mov     a,(measure+1)
              cjne    a,#0ffh,kmm3      ;compare (measure+1) for chack
              mov     (measure+1),#99h
              acall   ld10              ;display at LCD
              sjmp    km1

kmsen1:       ajmp    kmsen
```

```
;#######################
;CALCULATION SUBROUTINE
;#######################

kmm2:       anl     a,#0fh
            cjne    a,#0fh,ld17
            mov     a,measure
            clr     c
            subb    a,#06d          ;change to BCD
            mov     measure,a
            acall   ld10
            sjmp    km1

kmm3:       anl     a,#0fh
            cjne    a,#0fh,ld17
            mov     a,(measure+1)
            clr     c
            subb    a,#06d
            mov     (measure+1),a   ;change to BCD
            acall   ld10
            sjmp    km1

;#################################
;MEASURE COUNT DISPLAY  SUBROUTINE
;#################################

ld10:       mov     a,(measure+1)
            anl     a,#0f0h
            cjne    a,#00h,ld11
            mov     a,#0c0h
            acall   oper
            mov     a,#20h
            acall   dtw
            mov     a,(measure+1)
            anl     a,#0fh
            cjne    a,#00h,ld12
            mov     a,#0c1h
            acall   oper
            mov     a,#20h
            acall   dtw
            mov     a,measure
            anl     a,#0f0h
            cjne    a,#00h,ld13
            mov     a,#0c2h
            acall   oper
            mov     a,#20h
            acall   dtw
            mov     a,measure
            anl     a,#0fh
            mov     r4,a
            mov     1eh,#0c3h
            acall   mc1
            ret

ld11:       mov     a,(measure+1)
            anl     a,#0f0h
            swap    a
            mov     r4,a
            mov     1eh,#0c0h
            acall   mc1
ld14:       mov     a,(measure+1)
            anl     a,#0fh
            mov     r4,a
            mov     1eh,#0c1h
            acall   mc1
ld15:       mov     a,measure
```

```
                anl     a,#0f0h
                swap    a
                mov     r4,a
                mov     1eh,#0c2h
                acall   mc1
ld16:           mov     a,measure
                anl     a,#0fh
                mov     r4,a
                mov     1eh,#0c3h
                acall   mc1
                ret

ld12:           mov     a,#0c0h
                acall   oper
                mov     a,#20h
                acall   dtw
                jmp     ld14

ld13:           mov     a,#0c0h
                acall   oper
                mov     a,#20h
                acall   dtw
                mov     a,#0c1h
                acall   oper
                mov     a,#20h
                acall   dtw
                jmp    ld15

kms1:           ljmp    km1

;##########################
;ENTER KEY SCAN SUBROUTINE
;##########################
                    ;####################
                    ;TIMER COUNT (HOUR)
                    ;####################

kmsen:          cjne    a,#07h,kms1
                acall   delay3
                acall   delay3
                mov     rul,#80h
                mov     no.c,#18
                acall   lcd
                mov     dptr,#char13
                lcall   lcdld
                lcall   ktd

;################################
;TIMER COUNT DISPLAY SUBROUTINE
;################################

po:             mov     a,#0c2h
                acall   oper
                mov     a,#3ah              ;send : to between time-counting
                acall   dtw
                mov     a,#0c5h
                acall   oper
                mov     a,#3ah
                acall   dtw
                mov     a,#0c1h
                acall   oper
                mov     a,#00001110b
                acall   oper
                acall   delay3
```

```
;#####################
;+ KEY SCAN SUBROUTINE
;#####################

kt02:       lcall   seg
            mov     dptr,#portc
            movx    a,@dptr
            anl     a,#0fh
            cjne    a,#0eh,kt03     ;check + key pressed?
            acall   delay3
            mov     a,(timer+2)
            add     a,#01d
            da      a
            mov     (timer+2),a
ktd2:       acall   ktd
            mov     a,#0c1h
            acall   oper
            mov     a,#00001110b
            acall   oper
            acall   delay3
            ajmp    kt02            ;check key

;#####################
;- KEY SCAN SUBROUTINE
;#####################

kt03:       cjne    a,#0dh,kt04     ;check - key pressed?
            acall   delay3
            dec     (timer+2)
            mov     a,(timer+2)
            cjne    a,#0ffh,ktd1
            mov     (timer+2),#99h
            jmp     ktd2            ;display at LCD

;#######################
;CALCULATION SUBROUTINE
;#######################
ktd1:       anl     a,#0fh
            cjne    a,#0fh,ktd2
            mov     a,(timer+2)
            clr     c
            subb    a,#06d          ;chang to BCD
            mov     (timer+2),a
            jmp     ktd2            ;display at LCD

;#########################
;ENTER KEY SCAN SUBROUTINE
;#########################
                ;#######
                ;MINUTE
                ;#######

kt04:       cjne    a,#07h,kv
            acall   delay3
            acall   delay3
            acall   ktd
            mov     a,#0c4h         ;bring curser back to last position
            acall   oper
            mov     a,#00001110b
            acall   oper
            acall   delay3

;#####################
;+ KEY SCAN SUBROUTINE
;#####################
```

```
kt55:       lcall   seg
            mov     dptr,#portc
            movx    a,@dptr
            anl     a,#0fh
kt06:       cjne    a,#0eh,kt07     ;check  + key pressed?
            acall   delay3
            mov     a,(timer+1)
            cjne    a,#59h,ktd31    ;compare (timer+1)
            mov     (timer+1),#00h
            jmp     ktd3

kt012:      ajmp    kt011

;#########################
;MENU KEY SCAN SUBROUTINE
;#########################

kv:         cjne    a,#0bh,kt02
            acall   delay3
            ljmp    vl

;###################################
;CALCULATION AND DISPLAY SUBROUTINE
;###################################

ktd31:      add     a,#01d
            da      a
            mov     (timer+1),a
ktd3:       acall   ktd
            mov     a,#0c4h
            acall   oper
            mov     a,#00001110b
            acall   oper
            acall   delay3
            ajmp    kt55

;######################
;- KEY SCAN SUBROUTINE
;######################

kt07:       cjne    a,#0dh,kt08
            acall   delay3
            dec     (timer+1)
            mov     a,(timer+1)
            cjne    a,#0ffh,ktd4
            mov     (timer+1),#59h
            jmp     ktd3

;#########################
;ENTER KEY SCAN SUBROUTINE
;#########################

kv1:        cjne    a,#0bh,kt55
            acall   delay3
            ljmp    vl

;#####################################
;CHANGE HEXADECIMAL TO BCD SUBROUTINE
;#####################################

ktd4:       anl     a,#0fh
            cjne    a,#0fh,ktd3
            mov     a,(timer+1)
            clr     c
            subb    a,#06d
            mov     (timer+1),a
```

```
                jmp     ktd3

;#########################
;ENTER KEY SCAN SUBROUTINE
;#########################
                    ;#######
                    ;SECOND
                    ;#######

kt08:           cjne    a,#07h,kv1
                acall   delay3
                acall   delay3
                acall   ktd
                mov     a,#0c7h             ;bring curser back to last position
                acall   oper
                mov     a,#00001110b
                acall   oper
                acall   delay3

;######################
;+ KEY SCAN SUBROUTINE
;######################

kt001:          lcall   seg
                mov     dptr,#portc
                movx    a,@dptr
                anl     a,#0fh
kt10:           cjne    a,#0eh,kt11         ;check  + key pressed?
                acall   delay3
                mov     a,timer
                cjne    a,#59h,ktd32        ;compare timer
                mov     timer,#00h
                jmp     ktd110

;###################################
;CALCULATION AND DISPLAY SUBROUTINE
;###################################

ktd32:          add     a,#01d
                da      a
                mov     timer,a
ktd110:         acall   ktd
                mov     a,#0c7h
                acall   oper
                mov     a,#00001110b
                acall   oper
                acall   delay3
                ajmp    kt001

;######################
;- KEY SCAN SUBROUTINE
;######################

kt11:           cjne    a,#0dh,kt112        ;check  - key pressed?
                acall   delay3
                dec     timer
                mov     a,timer
                cjne    a,#0ffh,ktd12       ;compare timer
                mov     timer,#59h
                jmp     ktd110

;####################################
;CHANGE HEXADECIMAL TO BCD SUBROUTINE
;####################################
```

```
ktd12:      anl     a,#0fh
            cjne    a,#0fh,ktd110
            mov     a,timer
            clr     c
            subb    a,#06d          ;change to BCD
            mov     timer,a
            jmp     ktd110

;#########################
;ENTER KEY SCAN SUBROUTINE
;#########################

kt112:      cjne    a,#07h,kv2      ;check ENTER key pressed?
            acall   delay3
            acall   delay3
            acall   ktd
            mov     a,#0c1h
            acall   oper
            mov     a,#00001110b
            acall   oper
            acall   delay3
kt011:      ljmp    kt02

;#########################
;MENU KEY SCAN SUBROUTINE
;#########################

kv2:        cjne    a,#0bh,kt001    ;check MENU key pressed?
            acall   delay3
            ljmp    v1

                ;#######
                ;VOLUME
                ;#######

v1:         acall   lcd
            mov     rul,#80h
            mov     no.c,#07
            mov     dptr,#char14
            lcall   lcdld
            mov     a,#87h
            acall   oper
            mov     a,vol
            add     a,#30h
            acall   dtw

;######################
;+ KEY SCAN SUBROUTINE
;######################

v13:        lcall   seg
            mov     dptr,#portc
            movx    a,@dptr
            anl     a,#0fh
            cjne    a,#0eh,v11      ;check  + key pressed?
            acall   delay3
            acall   delay3
            mov     a,vol
            cjne    a,#07d,v12      ;check ENTER key pressed?
            mov     vol,#00h
            jmp     v14

;#################################
;CALCULATION AND DISPLAY SUBROUTINE
;#################################
```

```
vl2:        mov     a,vol
            add     a,#01d
            da      a
            mov     vol,a
vl4:        mov     a,#87h
            acall   oper
            mov     a,vol
            add     a,#30h
            acall   dtw
            acall   cv
            acall   sndnorm
            jmp     vl3

;######################
;- KEY SCAN SUBROUTINE
;######################

vl1:        cjne    a,#0dh,tm       ;check - key pressed?
            acall   delay3
            acall   delay3
            mov     a,vol
            cjne    a,#00h,vl6
            mov     vol,#07d
            jmp     vl4

vl6:        dec     vol
            mov     a,vol
            jmp     vl4

;###############
;ENTER KEY SCAN
;#################
                ;######
                ;TEMPO
                ;######

tm:         cjne    a,#07h,vl3      ;check ENTER key pressed?
            acall   delay3
            acall   delay3
            acall   lcd
            mov     rul,#80h
            mov     no.c,#06
            mov     dptr,#char15
            lcall   lcdld
            acall   tm11
            acall   tm3
            acall   seg
            jmp     tm1

;######################
;SOUND LEVEL SUBROUTINE
;######################

sndnorm:    mov     dptr,#porta     ;speaker
            mov     sndr1,#20h
            mov     sndr0,#03h
            acall   loops
            ret

loops:      mov     r0,volume
            mov     a,r0
            clr     c
            subb    a,#20h      ;20h
            mov     sndr2,a
loops1:     mov     r6,sndr0
            mov     a,r0
```

```
echo:       mov     r7,sndr1
echo1:      movx    @dptr,a
            djnz    r7,echo1
            djnz    r6,echo
            mov     r6,sndr0
            mov     a,#00h
rest:       mov     r7,sndr1
rest1:      movx    @dptr,a
            djnz    r7,rest1
            djnz    r6,rest
            dec     r0
            mov     a,r0
            cjne    a,sndr2,loops1
            ret

;###############################
;COMPARE SOUND LEVEL SUBROUTINE
;###############################
cv:         mov     a,vol
            cjne    a,#00h,cv0      ;compare for send value
            mov     volume,#20h
            ret

cv0:        cjne    a,#01h,cv1      ;compare for send value
            mov     volume,#22h
            ret

cv1:        cjne    a,#02h,cv2      ;compare for send value
            mov     volume,#25h
            ret

cv2:        cjne    a,#03h,cv3      ;compare for send value
            mov     volume,#38h
            ret

cv3:        cjne    a,#04h,cv4      ;compare for send value
            mov     volume,#4ah
            ret

cv4:        cjne    a,#05h,cv5      ;compare for send value
            mov     volume,#60h
            ret

cv5:        cjne    a,#06h,cv6      ;compare for send value
            mov     volume,#0a0h
            ret

cv6:        cjne    a,#07h,cv       ;compare for send value
            mov     volume,#0ffh
            ret

;###########################
;ENTER KEY SCAN SUBROUTINE
;###########################
tm1:        lcall   seg
            mov     dptr,#portc
            movx    a,@dptr
            anl     a,#0fh
            cjne    a,#0eh,tm2      ;check + key pressed?
            acall   delay5
            acall   tm21
            mov     r3,#05h
long:       acall   seg             ;send to 7-seg
            djnz    r3,long
            acall   tm3
            acall   seg
```

```
                jmp     tm1

;########################
;CALCULATION SUBROUTINE
;########################
tm21:           inc     tem
tm11:           mov     p1,tem
                mov     a,tem
                mov     b,#100          ;divided by 100 to
                div     ab              ;determine num of hundreds
                mov     (temdec+1),a
                mov     a,#10           ;divided remain by ten to
                xch     a,b             ;determine number of tens
                div     ab              ;ten's digit in ACC
                                        ;remainder is one's digit
                swap    a
                add     a,b             ;pack BCD digits in ACC
                mov     temdec,a
                ret

;######################
;- KEY SCAN SUBROUTINE
;######################

tm2:            cjne    a,#0dh,tm8      ;check - key pressed?
                acall   delay5
                dec     tem
                acall   tm11
                mov     r3,#05h
long2:          acall   seg
                djnz    r3,long2
                acall   tm3
                acall   seg
                jmp     tm1

;############################
;ENTER KEY SCAN SUBROUTINE
;############################

tm8:            cjne    a,#07h,tm1
                acall   delay5
                lcall   putdata
                acall   seg
                ljmp    home

;#########################################################################
;+ KEY SCAN SUBROUTINE FOR INCREASE OR DECREASE VOLUME WHEN WORKING
;#########################################################################

chv1:           cjne    a,#07h,sleep
                acall   delay5
                lcall   dmn
                ljmp    home


sleep:          cjne    a,#0dh,home1
                acall   delay5
                lcall   putdata
                setb    p3.0
                sjmp    $

home1:          ljmp    home2

;#################################
;DISPLAY TEMPO AT LCD SUBROUTINE
;#################################
```

```
tm3:        mov     a,(temdec+1)
            anl     a,#0fh
            cjne    a,#00h,tm4
            mov     a,#86h
            acall   oper
            mov     a,#20h
            acall   dtw
            mov     a,temdec
            anl     a,#0f0h
            cjne    a,#00h,tm5
            mov     a,#87h
            acall   oper
            mov     a,#20h
            acall   dtw
            acall   tm61
            ret

tm4:        mov     r4,a
            mov     1eh,#86h
            acall   mc1
tm5:        mov     a,temdec
            anl     a,#0f0h
            swap    a
            mov     r4,a
            mov     1eh,#87h
            acall   mc1
tm61:       mov     a,temdec
            anl     a,#0fh
            mov     r4,a
            mov     1eh,#88h
            acall   mc1
            ret

;###################################
;DISPLAY TEMPO AT 7-SEG SUBROUTINE
;###################################

seg:        mov     a,(temdec+1)
            anl     a,#0fh
            cjne    a,#00h,cdg
            mov     a,#0ffh
            mov     dptr,#portb
            movx    @dptr,a
            mov     r1,#00h
            mov     a,#80h
            mov     dptr,#portc
cff:        movx    @dptr,a
            djnz    r1,cff
            mov     a,temdec
            anl     a,#0f0h
            cjne    a,#00h,cdg1
            mov     a,#0ffh
            mov     dptr,#portb
            movx    @dptr,a
            mov     r1,#00h
            mov     a,#40h
            mov     dptr,#portc
css:        movx    @dptr,a
            djnz    r1,css
            acall   cdg2
            ret

cdg:        mov     dptr,#table
            movc    a,@a+dptr
            cpl     a
            mov     dptr,#portb
```

```
                movx    @dptr,a
                mov     r1,#00h
                mov     a,#80h
                mov     dptr,#portc
cf:             movx    @dptr,a
                djnz    r1,cf
cdg1:           mov     a,temdec
                anl     a,#0f0h
                swap    a
                mov     dptr,#table
                movc    a,@a+dptr
                cpl     a
                mov     dptr,#portb
                movx    @dptr,a
                mov     r1,#00h
                mov     a,#40h
                mov     dptr,#portc
cs:             movx    @dptr,a
                djnz    r1,cs
cdg2:           mov     a,temdec
                anl     a,#0fh
                mov     dptr,#table
                movc    a,@a+dptr
                cpl     a
                mov     dptr,#portb
                movx    @dptr,a
                mov     r1,#00h
                mov     a,#20h
                mov     dptr,#portc
ch:             movx    @dptr,a
                djnz    r1,ch
                ret

;################################
;DISPLAY TIMER COUNT SUBROUTINE
;################################

ktd:            mov     a,(timer+2)
                anl     a,#0f0h
                swap    a
                mov     r4,a
                mov     1eh,#0c0h
                acall   mc1
                mov     a,(timer+2)
                anl     a,#0fh
                mov     r4,a
                mov     1eh,#0c1h
                acall   mc1
                mov     a,(timer+1)
                anl     a,#0f0h
                swap    a
                mov     r4,a
                mov     1eh,#0c3h
                acall   mc1
                mov     a,(timer+1)
                anl     a,#0fh
                mov     r4,a
                mov     1eh,#0c4h
                acall   mc1
                mov     a,timer
                anl     a,#0f0h
                swap    a
                mov     r4,a
                mov     1eh,#0c6h
                acall   mc1
                mov     a,timer
```

```
                anl     a,#0fh
                mov     r4,a
                mov     1eh,#0c7h
                acall   mc1
                ret

;#####################################
;COMPARE AND DISPLAY METER SUBROUTINE
;#####################################

q:              cjne    a,#80h,e
eight:          mov     rul,#89h
                mov     no.c,#09
                mov     dptr,#char3
                lcall   lcdld
                ret

e:              cjne    a,#40h,t
triplets:       mov     rul,#89h
                mov     no.c,#09
                mov     dptr,#char4
                lcall   lcdld
                ret

t:              cjne    a,#20h,s
sixteenth:      mov     rul,#89h
                mov     no.c,#09
                mov     dptr,#char5
                lcall   lcdld
                ret

s:              cjne    a,#10h,k1
quarter:        mov     rul,#89h
                mov     no.c,#09
                mov     dptr,#char2
                lcall   lcdld
                ret

k1:             jmp     op

l1:             mov     rul,#80h
                mov     no.c,#09
                mov     dptr,#char1
                lcall   lcdld
l2:             mov     a,meter
                lcall   q
                ret

;#################
;DELAY SUBROUTINE
;#################

delay:          mov     r5,#0ffh
                mov     r6,#80h
                mov     r7,#04h
delay1:         djnz    r5,delay1
                djnz    r6,delay1
                djnz    r7,delay1
                ret

delay2:         mov     r5,#00h
                mov     r6,#00h
                mov     r7,#02h
dl2:            djnz    r5,dl2
                djnz    r6,dl2
                djnz    r7,dl2
```

```
                ret

delay3:         mov     r6,#0ffh
                mov     r7,#092h
dlay:           djnz    r6,dlay
                djnz    r7,dlay
                ret

delay4:         mov     r7,#0ffh
dlay1:          djnz    r7,dlay1
                ret

delay5:         mov     r6,#00h
                mov     r7,#0e6h
dl1:            djnz    r6,dl1
                djnz    r7,dl1
                ret

;#######################
;LCD CONTROL SUBROUTINE
;#######################

lcd:            mov     a,#00111000b    ;function set
                acall   oper
                mov     a,#00000110b    ;Entry INC
                acall   oper
                mov     a,#00001100b    ;Display ON
                acall   oper
                acall   delay3
                mov     a,#01h          ;Clear Screen
                acall   oper
                acall   delay3          ;clear speed
                ret

;########################
;LCD DISPLAY SUBROUTINE
;########################

lcdld:          mov     a,rul           ;send data to 1st lcd's line
                lcall   lcdlds
                lcall   delay3          ;value was in buffer(lcd)
                ret

;#############################
;WRITE INSTRUCTION SUBROUTINE
;#############################

oper:           push    dph
                push    dpl
                mov     dptr,#instw     ;send order
                movx    @dptr,a
                acall   delay4
                pop     dpl
                pop     dph
                ret

;######################
;WRITE DATA SUBROUTINE
;######################

dtw:            push    dph
                push    dpl
                mov     dptr,#dataw     ;send  data
                movx    @dptr,a
                acall   delay4
                pop     dpl
```

```
              pop     dph
              ret

;#######################
;LCD DISPLAY SUBROUTINE
;#######################

lcdlds:       push    dph
              push    dpl
              lcall   oper
              pop     dpl
              pop     dph
              mov     r2,no.c          ;send number charactor to lcd
lcdlds1:      clr     a
              movc    a,@a+dptr
              push    dph
              push    dpl
              lcall   dtw
              pop     dpl
              pop     dph
              inc     dptr
              djnz    r2,lcdlds1
              ret

;#################################
;COMPARE TIMESIGNATURE SUBROUTINE
;#################################

ts22:         cjne    a,#02h,ts24   ;compare key and bring charactor to lcd
t22:          mov     dptr,#char7
              acall   t221
              mov     rhythm,#04h
              ret

ts24:         cjne    a,#04h,ts34   ;compare key and bring charactor to lcd
t24:          mov     dptr,#char8
              acall   t221
              mov     rhythm,#02h
              ret

ts34:         cjne    a,#08h,ts44   ;compare key and bring charactor to lcd
t34:          mov     dptr,#char9
              acall   t221
              mov     rhythm,#03h
              ret

ts44:         cjne    a,#10h,ts68   ;compare key and bring charactor to lcd
t44:          mov     dptr,#char10
              acall   t221
              mov     rhythm,#04h
              ret

ts68:         cjne    a,#20h,ts22   ;compare key and bring charactor to lcd
t68:          mov     dptr,#char11
              acall   t221
              mov     rhythm,#03h
              ret

t221:         mov     rul,#0c0h       ;send 2/2
              mov     no.c,#03
              acall   lcdld
              ret

;#######################################
;MEASURE COUNT DISPLAY AT LCD SUBROUTINE
;#######################################
```

```
mc6:        mov     (measure+1),#00h
mc0:        mov     measure,#00h
            mov     a,r4
            mov     1eh,#0c3h
            acall   mc1
            ret

mc1:        mov     a,1eh           ;send figure to lcd
            acall   oper
            mov     a,r4
            add     a,#30h
            acall   dtw
            ret

;*********************************************************************
;              Main Function to approach the Beat
;*********************************************************************

dmn:        lcall   lcd             ;Through to eye of god
            lcall   inittimer
            lcall   h1txt
            lcall   h2txt
            mov     mcount,#00h
            mov     (mcount+1),#00h
            mov     tcount,#00h
            mov     (tcount+1),#00h
            mov     (tcount+2),#00h
            mov     runup,#00h      ;Beat
            lcall   showmt
            lcall   showtime
            lcall   showmc
            lcall   tempodiv
            lcall   showts
            lcall   tsset
            lcall   tempomul
            mov     a,tem
            cjne    a,#00h,sett0    ;no beat
            sjmp    sett1
sett0:      setb    tcon.4          ;Timer 0 run
sett1:      setb    tcon.6          ;Timer 1 run
loop:       lcall   seg
            mov     dptr,#portc
            movx    a,@dptr
            anl     a,#0fh
            clr     c
            subb    a,#0eh
            jz      stop
            jb      mcend,stop
            jb      tcend,stop
            sjmp    loop

stop:       clr     tcon.4          ;Beat End
            clr     mcend
            clr     tcon.6          ;Timer End
            clr     tcend
            ret

inittimer:  mov     ie,#10001010b   ;Enable T0 & T1
            mov     tmod,#01100101b ;T0 mode 0 & T1 mode 2
            mov     tl0,divl        ;T0 divide tempo
            mov     th0,divh
            mov     tl1,#0c3h       ;T1 Start value
            mov     th1,#0c3h       ;   Reload value
            ret
```

```
;*************************************
; Get & Put data from pRAM
;*************************************

getdata:        mov     dptr,#0030h
                mov     r0,#30h
                mov     r7,#49d
getlp:          movx    a,@dptr
                mov     @r0,a
                inc     r0
                inc     dptr
                djnz    r7,getlp
                ret

putdata:        mov     dptr,#0000h
                mov     r0,#00h
                mov     r7,#80h
putlp:          mov     a,@r0
                movx    @dptr,a
                inc     r0
                inc     dptr
                djnz    r7,putlp
                ret

;*************************************
; Print Head text
;*************************************

h1txt:          mov     a,#80h
                lcall   oper
                mov     dptr,#tm1txt
                mov     r1,#20d
h1p1:           clr     a
                movc    a,@a+dptr
                lcall   dtw
                inc     dptr
                djnz    r1,h1p1
                ret

h2txt:          mov     a,#0c0h
                lcall   oper
                mov     dptr,#tm2txt
                mov     r1,#20d
h1p2:           clr     a
                movc    a,@a+dptr
                lcall   dtw
                inc     dptr
                djnz    r1,h1p2
                ret

;*************************************
; the Show must go on
;*************************************
;time
showtime:       mov     a,#0c3h             ;Line 2 Col 4
                lcall   oper
                mov     r1,#03h
                mov     r0,#(tcount+2)
shtlp:          mov     a,@r0
                anl     a,#0f0h
                swap    a
                add     a,#30h
                lcall   dtw
                mov     a,@r0
                anl     a,#0fh
                add     a,#30h
```

```
                lcall   dtw
                mov     a,r1
                clr     c
                subb    a,#01h
                jz      shtbk
                mov     a,#3ah          ;":"
                lcall   dtw
                dec     r0
                djnz    r1,shtlp
shtbk:          ret

;measure count
showmc:         mov     a,#0d0h         ;Line 2 Col 4
                lcall   oper
                mov     r0,#(mcount+1)
                mov     r1,#02h
shmlp:          mov     a,@r0
                anl     a,#0f0h
                swap    a
                add     a,#30h
                lcall   dtw
                mov     a,@r0
                anl     a,#0fh
                add     a,#30h
                lcall   dtw
                dec     r0
                djnz    r1,shmlp
                ret

;meter
showmt:         mov     a,#82h          ;Line 1 Col 3
                lcall   oper
                mov     a,meter
squart:         cjne    a,#10h,seight   ;Quarter Note
                mov     dptr,#char2
                sjmp    shmtcont
seight:         cjne    a,#80h,stript   ;Eighth Note
                mov     dptr,#char3
                sjmp    shmtcont
stript:         cjne    a,#40h,ssixtn   ;Triplets
                mov     dptr,#char4
                sjmp    shmtcont
ssixtn:         mov     dptr,#char5     ;Sixteenth Note
shmtcont:       mov     r1,#09h
shmtlp:         clr     a
                movc    a,@a+dptr
                lcall   dtw
                inc     dptr
                djnz    r1,shmtlp
                ret

;time signature
showts:         mov     a,#91h
                lcall   oper
                mov     a,timesign
sts22:          cjne    a,#02h,sts24    ;2/2
                mov     dptr,#char7
                sjmp    stscont
sts24:          cjne    a,#04h,sts34    ;2/4
                mov     dptr,#char8
                sjmp    stscont
sts34:          cjne    a,#08h,sts44    ;3/4
                mov     dptr,#char9
                sjmp    stscont
sts44:          cjne    a,#10h,sts68    ;4/4
                mov     dptr,#char10
```

```
                sjmp    stscont
sts68:          mov     dptr,#char11    ;6/8
stscont:        mov     r1,#03h
stslp:          clr     a
                movc    a,@a+dptr
                lcall   dtw
                inc     dptr
                djnz    r1,stslp
                ret

;*************************************
; Arrange value to suit
;*************************************
;timer 0 divider
tempodiv:       mov     a,meter
quart:          cjne    a,#10h,eighth
                mov     divl,#02fh      ;Quarter Note
                mov     divh,#0fdh
                sjmp    tdcont
eighth:         cjne    a,#80h,tript
                mov     divl,#097h      ;Eighth Note
                mov     divh,#0feh
                sjmp    tdcont
tript:          cjne    a,#40h,sixtn
                mov     divl,#00fh      ;Triplets
                mov     divh,#0ffh
                sjmp    tdcont
sixtn:          mov     divl,#04bh      ;Sixteenth Note
                mov     divh,#0ffh
tdcont:         ret

;time signature relate
tsset:          mov     a,timesign
tts22:          cjne    a,#02h,tts24    ;2/2
                mov     rhythm,#04h
                sjmp    tscont
tts24:          cjne    a,#04h,tts34    ;2/4
                mov     rhythm,#02h
                sjmp    tscont
tts34:          cjne    a,#08h,tts44    ;3/4
                mov     rhythm,#03h
                sjmp    tscont
tts44:          cjne    a,#10h,tts68    ;4/4
                mov     rhythm,#04h
                sjmp    tscont
tts68:          mov     rhythm,#03h     ;6/8
tscont:         ret

;upper value to beat
tempomul:       mov     a,meter
mquart:         cjne    a,#10h,meight   ;Quarter Note
                mov     b,#01h
                sjmp    tpmcont
meight:         cjne    a,#80h,mtript   ;Eighth Note
                mov     b,#02h
                sjmp    tpmcont
mtript:         cjne    a,#40h,msixtn   ;Triplets
                mov     b,#03h
                sjmp    tpmcont
msixtn:         mov     b,#04h          ;Sixteenth
tpmcont:        mov     a,rhythm
                mul     ab
                mov     rhythm,a
                ret
```

```
;count coming pulse
beat:           mov     a,runup
                cjne    a,rhythm,bcont
                lcall   incmc
                mov     runup,#00h
                lcall   sndhi
                sjmp    bcont2
bcont:          lcall   sndnorm
bcont2:         ret

;*************************************
; Service is come first
;*************************************
;beat service
t0serv:         clr     tcon.4
                mov     tl0,divl        ;T0 divide tempo
                mov     th0,divh
                inc     runup
                lcall   beat
                lcall   showmc
                lcall   chkm
                lcall   seg
                setb    tcon.4
                reti

;time service
t1serv:         lcall   inctime
                lcall   showtime
                lcall   chkt
                lcall   seg
                reti

;*************************************
; Progressive routine
;*************************************
;time progress
inctime:        mov     a,tcount        ;inc sec
                add     a,#01d
                da      a
                mov     tcount,a
                cjne    a,#60h,tcont    ;sec = 60 ?
                mov     tcount,#00h
                mov     a,(tcount+1)    ;inc min
                add     a,#01d
                da      a
                mov     (tcount+1),a
                cjne    a,#60h,tcont    ;min = 60 ?
                mov     (tcount+1),#00h
                mov     a,(tcount+2)    ;inc hr
                add     a,#01d
                da      a
                mov     (tcount+2),a
tcont:          ret

;measure progress
incmc:          clr     c
                mov     a,mcount
                add     a,#01d
                da      a
                mov     mcount,a
                mov     a,(mcount+1)
                addc    a,#00h
                da      a
                mov     (mcount+1),a
                ret
```

```
;************************************
; To hear the rhythm
;************************************

;play hi for i
sndhi:          mov     dptr,#porta
                mov     sndr1,#40h
                mov     sndr0,#01h
                lcall   loops
                ret

;************************************
; Sure ?
;************************************
;time check
chkt:           mov     a,tcount
                cjne    a,timer,ctcont
                mov     a,(tcount+1)
                cjne    a,(timer+1),ctcont
                mov     a,(tcount+2)
                cjne    a,(timer+2),ctcont
                setb    tcend
ctcont:         ret

;measure check
chkm:           mov     a,mcount
                cjne    a,#00h,chk
                mov     a,(mcount+1)
                cjne    a,#00h,chk
                sjmp    ckcont
chk:            mov     a,mcount
                cjne    a,measure,ckcont
                mov     a,(mcount+1)
                cjne    a,(measure+1),ckcont
                setb    mcend
ckcont:         ret

;************************************
; Character to play role
;************************************

tm1txt:         db      "M:             TS:    "
tm2txt:         db      "TC:           MC:     "
char1:          db      "Meter(M):"
char2:          db      "Quarter  "
char3:          db      "Eighth   "
char4:          db      "Triplets "
char5:          db      "Sixteenth"
char6:          db      "Time Signature(TS):"
char7:          db      "2/2"
char8:          db      "2/4"
char9:          db      "3/4"
char10:         db      "4/4"
char11:         db      "6/8"
char12:         db      "Measure Count(MC):"
char13:         db      "Time Count(TC):   "
char14:         db      "Volume:"
char15:         db      "Tempo:"
table:          db      3fh,06h,5bh,4fh,66h,6dh,7dh,07h
                db      7fh,6fh,77h,7ch,39h,5eh,79h,71h

                end
```

ภาคผนวก ค
การใช้งาน LCD Module

การต่อ LCD MODULE เข้ากับระบบไมโคร

การต่อแบบ MEMORY MAP

1. สามารถต่อเข้ากับ CHIP เบอร์ทั่วๆ ไปได้ เช่น 8051 หรือ Z80 โดยทำให้ระบบไมโครมองเห็น LCD MODULE ในลักษณะของ MEMORY ได้ทันที

2. ผู้ใช้สามมารถเขียนและอ่านข้อมูลจาก LCD MODULE ได้ ทำให้มองเสมือนเป็น MEMORY BUFFER ไปในตัว

3. เนื่องจากสามารถอ่านข้อมูลกลับได้ จึงทำให้สามารถตรวจสอบ FLAG ความพร้อมในขณะที่ LCD MODULE กำลังทำงานได้

4. ใช้ได้กับบอร์ดที่มี LCD BUS มาให้พร้อมเท่านั้น

5. ทำให้กินพื้นที่ของหน่วยความจำไปส่วนหนึ่ง และต้องมีการ DECODE ละเอียดพอควร

6. การจัดขาสัญญาณจะต้องเป็นไปแบบ CHIP แต่ละเบอร์

ขาสัญญาณต่างๆในการต่อใช้งาน LCD MODULE

PIN	SYMBOL	LEVEL	FUNCTION
1	Vss	---	0V GND
2	Vcc	---	+5V POWER SUPPLY
3	Vee	---	+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 (L → H)
7	DB0	H/L	DATA BUS BIT 0
8	DB1	H/L	DATA BUS BIT 1
9	DB2	H/L	DATA BUS BIT 2
10	DB3	H/L	DATA BUS BIT 3
11	DB4	H/L	DATA BUS BIT 4
12	DB5	H/L	DATA BUS BIT 5
13	DB6	H/L	DATA BUS BIT 6
14	DB7	H/L	DATA BUS BIT 7

รายละเอียดการใช้งาน LCD MODULE

ตารางคำสั่งของ DMC202

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

รายละเอียดของแต่ละคำสั่ง

1. CLEAR DISPLAY

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

สำหรับการ CLEAR DISPLAY โดยจะทำการเขียนตัวอักษร SPACE ลงใน DDRAM ทั้งหมด และ

กำหนดค่า DDRAM ADDRESS ให้เป็น 0 พร้อมทั้ง CURSOR จะกลับไปที่ตำแหน่งซ้ายบนสุดของจอภาพ

2.CURSOR AT HOME

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

สำหรับการกำหนดค่า DDRAM ADDRESS ให้เป็น 0 พร้อมทั้ง CURSOR จะไปอยู่ที่ตำแหน่งซ้ายบนสุดของจอภาพ 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 กำหนดทิศทางของ CURSOR และ DDRAM ให้เป็น DECREMENT

I/D = 1 กำหนดทิศทางของ CURSOR และ DDRAM ให้เป็น INCREMENT

S = 0 เมื่อเขียนข้อมูลแล้ว ตัว CURSOR จะถูกเลื่อนไปทิศทางตามค่า I/D

S = 1 เมื่อเขียนข้อมูลแล้ว ตัว CURSOR จะอยู่กับที่ และตัวอักษรจะถูกดันไปทิศทางตามค่า I/D

การกำหนดค่า I/D และ S นี้ ให้กำนดก่อนการเขียนข้อมูลใน DDRAM และเมื่อกำหนดแล้ว จะต้องไม่ใช้คำสั่ง 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 CURSOR

C = 1 กำหนดให้ ON CURSOR โดย CURSOR จะเป็นเส้นขีดใต้ตัวอักษร

B = 0 กำหนดให้ไม่มีการกระพริบที่ตำแหน่ง CURSOR

B = 1 กำหนดให้มีการกระพริบที่ตำแหน่ง CURSOR (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 กำหนดให้เลื่อน CURSOR R/L

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

R/L = 0 กำหนดให้มีทิศทางไปทางซ้าย

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 BIT

DL = 1 กำหนดให้การติดต่อกับ LCD MODULE เป็นแบบ 8 BIT

จะสังเกตว่า การกำนดค่า DL นี้ สามารถกระทำได้ที่ DB4-DB7 ซึ่งถ้ามีการกำหนดเป็นแบบ 4 BIT ตั้งแต่ครั้งแรก หลังจากจ่ายไฟเลี้ยง ก็จะทำให้ LCD MODULE มีการรับข้อมูลแบบ 4 BIT ทันที

N = 0 กำหนดจำนวนบรรทัดแบบ 1/8 DUTY และ 1/11 DUTY

N = 1 กำหนดจำนวนบรรทัดแบบ 1/16 DUTY

F = 0 กำหนดให้ตัวอักษรเป็นแบบ 5×7 DOTS

F = 1 กำหนดให้ตัวอักษรเป็นแบบ 5×10 DOTS (กรณีที่ LCD MODULE เป็นแบบ 5×7 อยู่แล้ว ก็จะไม่มีผลอะไร)

7.SET CGRAM ADDRESS

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

เมื่อได้ทำการกำหนดADDRES ของ CGRAM เมื่อได้ทำการกำหนดไว้แล้ว การอ่านและเขียน DATA ที่ต่อจากนี้ จะเป็นไปตาม ADDRESS ที่กำหนดทันที

8.SET DDRAM ADDRESS

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

สำหรับการกำหนด ADDRESS ของ DDRAM เมื่อได้ทำการกำหนดไว้แล้ว การอ่านและเขียน DATA ที่ต่อจากนี้ จะเป็นไปตาม ADDRESS ที่กำหนดทันที ซึ่งแสดงดังภาพต่อไปนี้

รุ่น DMC202

00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0E	0F	10	11	12	13
40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4E	4F	50	51	52	53

9.BUSY FLAG AND ADDRESS READ

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

สำหรับการอ่านค่าBF (BUSY FLAG) ซึ่งบอกถึงความพร้อมของ LCD MODULE ในการรับข้อมูล ถ้า BF = 0 หมายถึงว่าพร้อมที่จะรับข้อมูลต่อไปได้ แต่ถ้า BF = 1 หมายถึงว่ายังไม่พร้อม นอกจากนี้ยังเป็นการกำหนดค่า ADDRESS ของ CGRAM หรือ DDRAM ด้วย

การอ่านและเขียนข้อมูลกับ DDRAM/CGRAM

1. WRITE DATA TO DDRAM OR CGRAM

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

สำหรับการเขียนข้อมูลลงหน่วยความจำDDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้ว ADDRESS จะถูกเพิ่มหรือลงโดยอัตโนมัติตามที่กำหนดจากค่า I/D ในคำสั่ง ENTRY MODE SET และการเขียนจะเป็น DDRAM หรือ CGRAM ก็ขึ้นอยู่กับว่า ก่อนหน้าคำสั่งนี้ มีการกำหนด ADDRESS ที่ใด

2.READ DATA FROM DDRAM OR CGRAM

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

สำหรับการอ่านข้อมูลลงหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้ว ADDRESS จะถูกเพิ่มหรือลงโดยอัตโนมัติ ตามที่กำหนดจากค่า I/D ในคำสั่ง ENTRY MODE SET และการอ่านจะเป็น DDRAM หรือ CGRAM ก็ขึ้นอยู่กับว่า ก่อนหน้าคำสั่งนี้ มีการกำหนด ADDRESS ที่ใด

ภาคผนวก ง

Data Sheet

PRELIMINARY

AUGUST 1988

DATA SHEET 80C51-L / 80C31-L

CMOS SINGLE-CHIP 8 BIT 3V-MICROCONTROLLER

- **80C51-L – CMOS SINGLE-SHIP 8-BIT MICROCONTROLLER with factory mask-programmable ROM**
- **80C31-L - CMOS SINGLE-CHIP 8-BIT CONTROL-ORIENTED CPU with RAM and I/0**
- **80C51-L/C31-L: 0 TO 6 MHz, VCC = 2.7V TO 6V**

FEATURES

- **POWER CONTROL MODES**
- **128 x 8 BIT RAM**
- **32 PROGRAMMABLE I/O LINES**
- **TWO 16-BIT TIMER/COUNTERS**
- **64K PROGRAM MEMORY SPACE**
- **FULLY STATIC DESIGN**
- **HIGH PERFORMANCE SAJI VI CMOS PROCESS**
- **BOOLEAN PROCESSOR**
- **5 INTERRUPT SOURCES**
- **PROGRAMMABLE SERIAL PORT**
- **64K DATA MEMORY SPACE**
- **TEMPERATURE RANGE: 0 TO 70°C**

DESCRIPTION

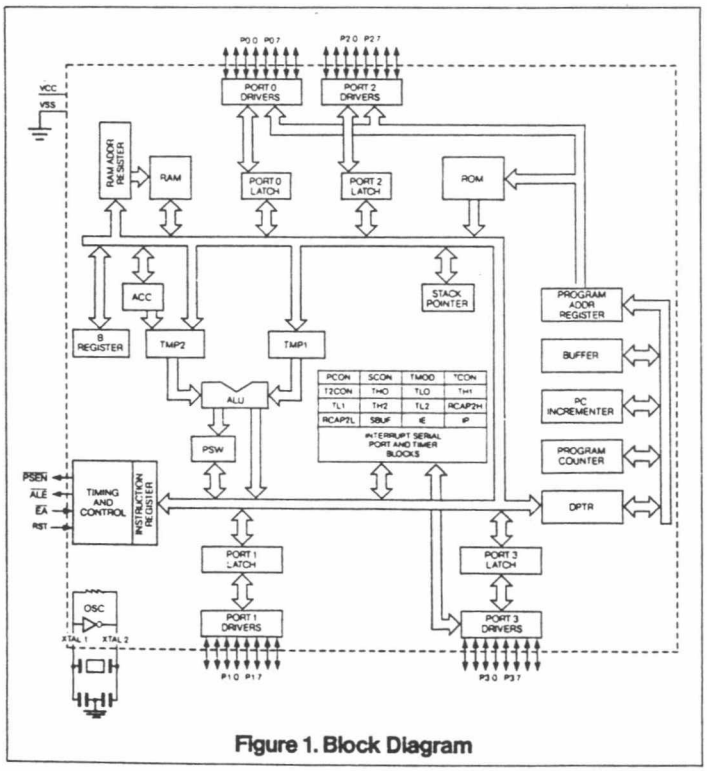


Figure 1. Block Diagram

MHS's 80C51 and 80C31 are high performance CMOS versions of the 8051/8031 NMOS single chip 8 bit μC and is manufactured using a self-aligned silicon gate CMOS process (SAJI VI).

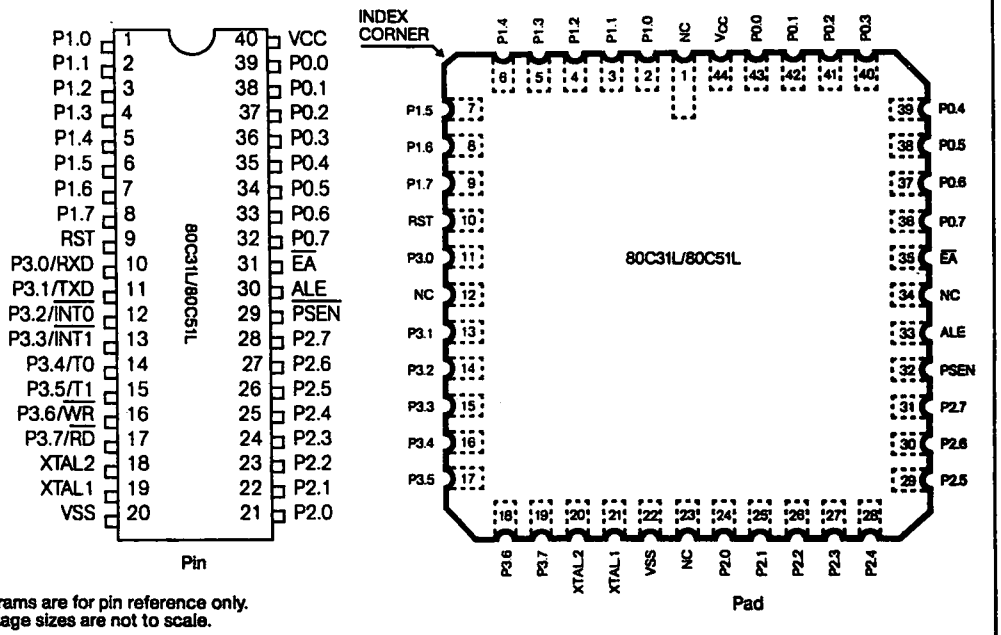
The fully static design of the MHS 80C51/80C31 allows to reduce system power consumption by bringing the clock frequency down to any value, even DC, without loss of data.

The 80C51 retains all the features of the 8051: 4K bytes of ROM; 128 bytes of RAM; 32 I/O lines; two 16 bit timers; a 5-source 2-level interrupt structure; a full duplex serial port; and on-chip oscillator and clock circuits.

In addition, the 80C51 has two software-selectable modes of reduced activity for further reduction in power consumption. In the Idle Mode the CPU is frozen while the RAM, the timers, the serial port, and the interrupt system continue to function. In the Power Down Mode the RAM is saved and all other functions are inoperative.

The 80C31 is identical to the 80C51 except that it has no on-chip ROM.

Figure 2. Configurations



Diagrams are for pin reference only.
Package sizes are not to scale.

IDLE AND POWER DOWN OPERATION

Figure 3 shows the internal Idle and Power Down clock configuration. As illustrated, Power Down operation stops the oscillator. Idle mode operation allows the interrupt, serial port, and timer blocks to continue to function while the clock to the CPU is gated off. These special modes are activated by software via the Special Function Register, Its hardware address is 87H. PCON is not bit addressable.

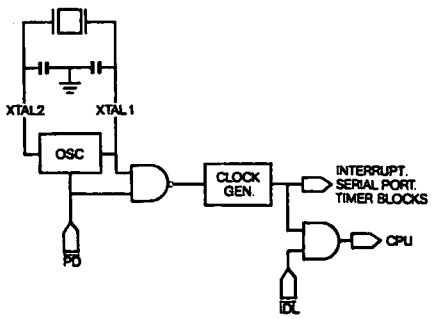


Figure 3. Idle and Power Down Hardware

PCON: Power Control Register

(MSB) (LSB)

SMOD	-	-	-	GF1	GF0	PD	IDL

Symbol	Position	Name and Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1, 2 or 3.
-	PCON.6	(Reserved)
-	PCON.5	(Reserved)
-	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON.0	Ide mode bit. Setting this bit activates idle mode operation.

If 1's are written to PD and IDL at the same time. PD takes precedence. The reset value of PCON is (OXXX0000).

Table 1. Status of the external pins during Idle and Power Down modes

Mode	Program Memory	ALE	PSEN	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Port Data	Port Data	Port Data	Port Data
Idle	External	1	1	Floating	Port Data	Address	Port Data
Power Down	Internal	0	0	Port Data	Port Data	Port Data	Port Data
Power Down	External	0	0	Floating	Port Data	Port Data	Port Data

IDLE MODE

The instruction that sets PCON.0 is the last instruction executed before the Idle mode is activated. Once in the Idle mode the CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, RAM, and all other registers maintain their data during Idle. *Table 1* describes the status of the external pins during Idle mode.

There are two ways to terminate the Idle mode. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating Idle mode. The interrupt is serviced, and following RETI, the next instruction to be executed will be the one following theinstruction that wrote a 1 to PCON.0.

The flag bits GF0 and GF1 may be used to determine whether the interrupt was received during normal execution or during the Idle mode. For example, the instruction that writes to PCON.0 can also set or clear one or both flag bits. When Idle mode is terminated by an enabled interrupt, the service routine can examine the status of the flag bits.

The second way of terminating the Idle mode is with a hardware reset. Since the oscillator is still running, the hardware reset needs to be active for only 2 machine cycles (24 oscillator periods) to complete the reset operation.

POWER DOWN MODE

The instruction that sets PCON.1 is the last executed prior to entering power down. Once in power down, the oscillator is stopped. The contents of the onchip RAM and the Special Function Register is saved during power down mode. A hardware reset is the only way of exiting the power down mode. The hardware reset initiates the Special Function Register *(see Table 1)*.

In the Power Down mode. V_{CC} may be lowered to minimize circuit power consumption. Care must be taken to ensure the voltage is not reduced until the power down mode is entered, and that the voltage is restored before the hardware reset is applied which frees the oscillator. Reset should not be released until the oscillator has restarted and stabilized.

Table 1 describes the status of the external pins while in the power down mode. It should be noted that if the power down mode is activated while in external program memory, the port data that is held in the Special Function Register P2 is restored to Port 2. If the data is a 1, the port pin is held high during the power down mode by the strong pullup, T1, shown in *Figure 4*.

STOP CLOCK MODE

Due to static design, the MHS 80C31/C51 clock speed can be reduced until 0 MHz without any data loss in memory or registers. This mode allows step by step utilization, and permits to reduce system power consumption by bringing the clock frequency down to any value. At 0 MHz, the power consumption is the same as in the Power Down Mode.

80C51 I/O PORTS

The I/O port drive of the 80C51 is similar to the 8051. The I/O buffers for Ports 1, 2, and 3 are implemented as shown in *figure 4*.

When the port latch contains a 0, all pFETS in *figure 4* are off while the nFET is turned on. When the port latch makes a 0-to-1 transition, the nFET turns off. The strong pullup pFET, T1, turns on for two oscillator periods, pulling the output high very rapidly. As the output line is drawn high, pFET T3 turns on through the inverter to supply the I_{OH} source current. This inverter and T3 form a latch which holds the 1 and is supported by T2. When Port 2 is used as an address port, for access to external program of data memory, any address bit that contains a 1 will have his strong pullup turned on for the entire duration of the external memory access.

When an I/O pin on Ports 1, 2, or 3 is used as an input, the user should be aware that the external circuit must sink current during the logical 1-to-0 transition. The maximum sink current is specified as ITL under the D.C. Specifications. When the input goes below approximately 2V, T3 turns off to save ICC current. Note, when returning to a logical 1, T2 is the only internal pullup that is on. This will result in a slow rise time if the user's circuit does not force the input line high.

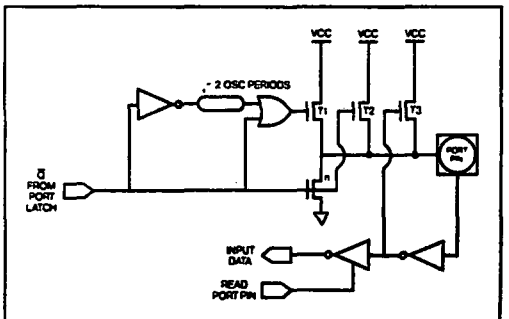


Fig. 4. I/O Buffers in the 80C51 (Ports 1, 2, 3)

80C51 PIN DESCRIPTIONS

V_{SS}
Circuit ground potential

V_{CC}
Supply voltage during normal, Idle, and Power Down operation.

Port 0
Port 0 is an 8-bit open drain bi-directional I/O port. Port 0 pins that have 1's written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1's. Port 0 also outputs the code bytes during program verification in the 80C51. External pullups are required during program verification. Port 0 can sink eight LS TTL inputs.

Port 1
Port 1 is an 8-bit bi-directional I/O port with internal pullups. Port 1 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups.

Port 1 also receives the low-order address bytes during program verification. In the 80C51, Port 1 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

Port 2
Port 2 is an 8-bit bi-directional I/O port with internal pullups. Port 2 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups. Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @ DPTR). In this application, it uses strong internal pullups when emitting 1's. During accesses to external Data Memory that uses 8-bit addresses (MOVX @ Ri), Port 2 emits the contents of the P2 Special Function Register.

It also receives the high-order address bits and control signals during progam verification in the 80C51. Port 2 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

Port 3
Port 3 is an 8-bit bi-directional I/O port with internal pullups. Port 3 pins that have 1's written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the pullups. It also serves the functions of various special features of the MCS-51 Family, as listed below.

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	INT0 (external interrupt 0)
P3.3	INT1 (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{WR}$ (external Data Memory write strobe)
P3.7	$\overline{RD}$ (external Data Memory read strobe)

Port 3 can sink/source three LS TTL inputs. It can drive CMOS inputs without external pullups.

RST
A high level on this for two machine cycles while the oscillator is running resets the device. An internal pull-down resistor permits Power-On reset using only a capacitor connected to V_{CC}.

ALE
Address Latch Enable output for latching the low byte of the address during accesses to external memory. ALE is activated as though for this purpose at a constant rate of 1/6 the oscillator frequency except during an external data memory access at which time one ALE pulse is skipped. ALE can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

$\overline{PSEN}$
Program Store Enable output is the read strobe to external Program Memory. PSEN is activated twice each machine cycle during fetches from external Program Memory. (However, when executing out of external Program Memory, two activations of PSEN are skipped during each access to external Data Memory). PSEN is not activated during fetches from internal Program Memory. PSEN can sink/source 8 LS TTL inputs. It can drive CMOS inputs without an external pullup.

$\overline{EA}$
When EA is held high, the CPU executes out of internal Program Memory (unless the Program Counter exceeds OFFFH). When EA is held low, the CPU executes only out of external Program Memory. EA must not be floated.

XTAL1
Input to the inverting amplifier that forms the oscillator. Receives the external oscillator signal when an external oscillator is used.

XTAL2
Output of the inverting amplifier that forms the oscillator, and input to the internal clock generator. This pin should be floated when an external oscillator is used.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output respectively, of an inverting amplifier which is configured for use as an on-chip oscillator, as shown in figure 5. Either a quartz crystal or ceramic resonator may be used.
To drive the device from an external clock source, XTAL1 should be driven while XTAL2 is left unconnected as shown in *figure 6*. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but munimum and maximum high and low times specified on the Data Sheet must be observed.

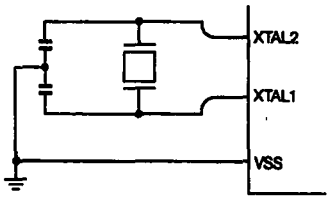


Figure 5. Crystal Oscillator

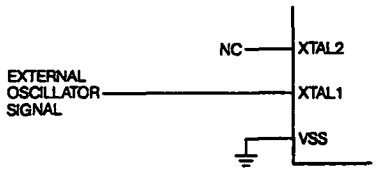


Figure 6. External Drive Configuration

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias:
Commercial ..... 0°C to 70°C
Industrial ..... −40°C to 85°C
Storage Temperature ..... −65°C to +150°C
Voltage on V_{CC} to V_{SS} ..... −0.5V to +7V
Voltage on Any Pin to V SS..... −0.5V to V_{CC} +0.5V
Power Dissipation 1W*

*This value is based on the maximum allowable die temperature and the thermal resistance of the package.

***NOTICE:**

Stresses at or above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions may affect device reliability.

DC CHARACTERISTICS

TA = −40°C to 85°C; VCc = 2.7V to 6V; VSS = 0V; F = 0 to 6 MHz

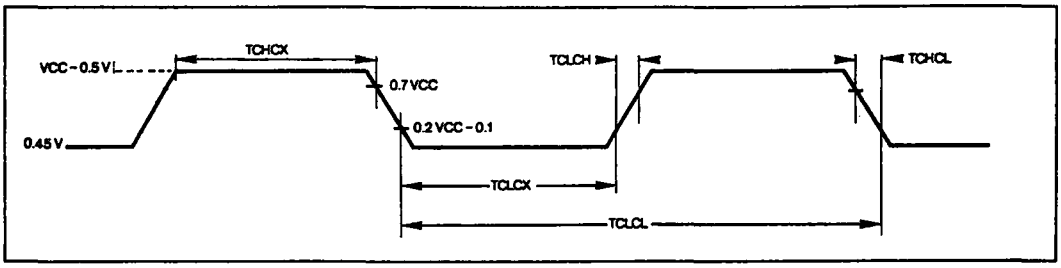
Symbol	Parameter	Min	Max	Unit	Test Conditions
VIL	Input Low Voltage	−0.5	0.2 V_{CC} −0.1	V	
VIH	Input High Voltage (Except XTALs and RST)	0.2 V_{CC} +0.9	V_{CC} +0.5	V	
VIH1	Input High Voltage to RST for Reset	0.7 V_{CC}	V_{CC} +0.5	V	
VIH2	Input High Voltage To XTAL 1	0.7 V_{CC}	V_{CC} +0.5	V	
VPD	Power Down Voltage To V_{CC} in PD Mode	2.0	6.0	V	
VOL	Output Low Voltage (Ports 1, 2, 3)		0.45	V	IOL = 1.6mA (note 1)
VOL1	Output Low Voltage Port 0, ALE, $\overline{PSEN}$		0.45	V	IOL = 3.2mA (note 1)
VOH	Output High Voltage Ports 1, 2, 3	0.9 V_{CC}		V	IOH = −10µa
		2.4		V	IOH = −60µA V_{CC} = 5V ± 10%
VOH1	Output High Voltage (Port 0 in External in External Bus Mode), ALE, $\overline{PSEN}$	0.9 V_{CC}		V	IOH = −40µA
		2.4		V	IOH = −400µA V_{CC} = 5V ± 10%
IIL	Logical 0 Input Current Ports 1,2,3		−50	µA	Vin = 0.45V
ILI	Input Leakage Current		± 10	µA	0.45 < Vin < V_{CC}
ITL	Logical 1 to 0 Transition Current (Ports 1, 2, 3)		−500	µA	Vin = 2.0V
ICCPD	Power Supply Current (Power Down Mode)	50	10	µA	V_{CC} = 2.0V to 5.5V (note 2)
RRST	RST Pulldown Resistor	50	150	kΩ	
CIO	Capacitance of I/O Buffer		10	pF	f_C = 1MHz, T_A = 25°C

Note 1:

Capacitive loading on Ports 0 and 2 may cause spurious noise pulses to be superimposed on the V_{OL}S of ALE and Ports 1 and 3. The noise is due to external bus capacitance discharging into the Port 0 and Port 2 pins when these pins make 1-to-0 transitions during bus operations. In the worst cases (capacitive loading 100 pF), the noise pulse on the ALE line may exceed 0.45V with maxi V_{OL} peak 0.6V. A Schmitt Trigger use is not necessary.

EXTERNAL CLOCK DRIVE CHARACTERISTICS (XTAL 1)

Symbol	Parameter	Variable Clock freq = 0 to 6 MHz		Unit
		Min	Max	
TCLCL	Oscillator Period	166		ns
TCHCX	High Time	20		ns
TCLCX	Low Time	20		ns
TCLCH	Rise Time		20	ns
TCHCL	Fall Time		20	ns



AC CHARACTERISTICS

(T_A = -40°C to 85°C, V_{CC} = 2.7V to 6V, V_{SS} = 0V)
(Load Capacitance for Port 0, ALE, and $\overline{PSEN}$ = 100pf; Load Capacitance for All Other Outputs = 80pf).

EXTERNAL PROGRAM MEMORY CHARACTERISTICS

Symbol	Parameter	Min	Max	Units
TLHLL	ALE Pulse Width	2TCLCL - 40		ns
TAVLL	Address Valid to ALE	TCLCL - 55		ns
TLLAX	Address Hold After ALE	TCLCL - 35		ns
TLLIV	ALE to Valid instr In		4TCLCL - 170	ns
TLLPL	ALE to $\overline{PSEN}$	TCLCL - 25		ns
TPLPH	$\overline{PSEN}$ Pulse Width	3TCLCL - 35		ns
TPLIV	$\overline{PSEN}$ to Valid Instr In		3TCLCL - 220	ns
TPXIX	Input Instr Hold After $\overline{PSEN}$	0		ns
TPXIZ	Input Instr Float After $\overline{PSEN}$		TCLCL - 20	ns
TPXAV	$\overline{PSEN}$ to Address Valid	TCLCL - 8		ns
TAVIV	Address to Valid Instr In		5TCLCL - 220	ns
TPLAZ	$\overline{PSEN}$ Low to Address Float		0	ns

See next page for External Data Memory Characteristics.

EXTERNAL DATA MEMORY CHARACTERISTICS

Symbol	Parameter	Min	Max	Units
TRLRH	$\overline{RD}$ Pulse Width	6TCLCL-100		ns
TWLWH	$\overline{WR}$ Pulse Width	6TCLCL-100		ns
TLLAX	Data Address Hold After ALE	TCLCL-35		ns
TRLDV	$\overline{RD}$ to Valid Data In		5TCLCL-165	ns
TRHDX	Data Hold After $\overline{RD}$	0		ns
TRHDZ	Data Float After $\overline{RD}$		2TCLCL-70	ns
TLLDV	ALE to Valid Data In		8TCLCL-150	ns
TAVDV	Address to Valid Data In		9TCLCL-165	ns
TLLWL	ALE to $\overline{WR}$ or $\overline{RD}$	3TCLCL-50	3TCLCL+50	ns
TAVWL	Address to $\overline{WR}$ or $\overline{RD}$	4TCLCL-130		ns
TQVWX	Data Valid to $\overline{WR}$ Transition	TCLCL-60		ns
TQVWH	Data Setup to $\overline{WR}$ High	7TCLCL-150		ns
TWHQX	Data Hold After $\overline{WR}$	TCLCL-50		ns
TRLAZ	$\overline{RD}$ Low to Address Float		0	ns
TWHLH	$\overline{RD}$ or $\overline{WR}$ High to ALE High	TCLCL-40	TCLCL+40	ns

MAXIMUM I_{CC} (mA)

	Operating (Note 3)			Idle (Note 4)		
Freq. V_{CC}	2.7V	5V	6V	2.7V	5V	6V
1 MHz	0.8 mA	1.5 mA	1.8 mA	400 µA	800 µA	1 mA
6 MHz	4 mA	8 mA	10 mA	1.2 mA	3.5 mA	3.8 mA

Note 2:

Power Down I_{CC} is measured will all output pins disconnected; EA=Port 0=V_{CC}; XTAL2 N.C.; RST=V_{SS}

Note 3:

I_{CC} is measured with all output pins disconnected; XTAL1 driven with TCLCH, TCHCL=5 ns, VIL=V_{SS} + 0.5V; V_{IH} V_{CC} - 0.5V; XTAL2 N.C.; EA=RST=Port 0=V_{CC}. I_{CC} would be slightly higher if a crystal oscillator used.

Note 4:

Idle I_{CC} is measured with all output pins disconnected; XTAL1 driven TCLCH, TCHCL=5 ns, V_{IL} =V_{SS} + 0.5V; V_{IH}=V_{CC} - 0.5V; XTAL2 N.C.; Port 0=V_{CC}; EA=RST=V_{SS}.

EXPLANATION OF THE AC SYMBOLS

Each timing symbol has 5 characters. The first character is always a 'T' (stands for time). The other characters, depending on their positions, stand for the name of a signal or the logical status of that signal. The following is a list all the characters and what they stand for.

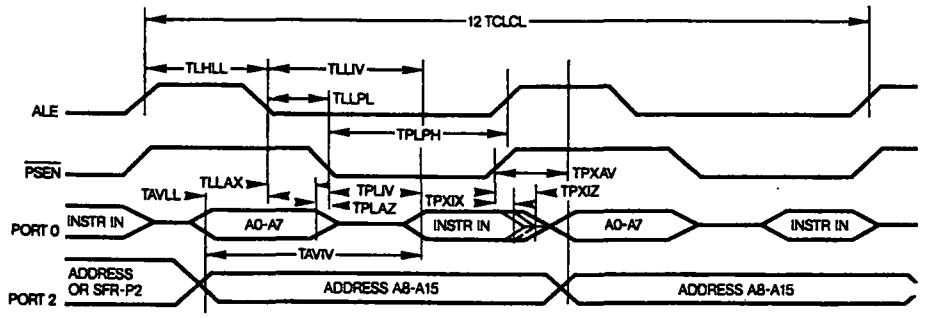
EXAMPLE:

TAVLL=Time for Address Valid to ALE low.
TLLPL=Time for ALE low to $\overline{PSEN}$ low.

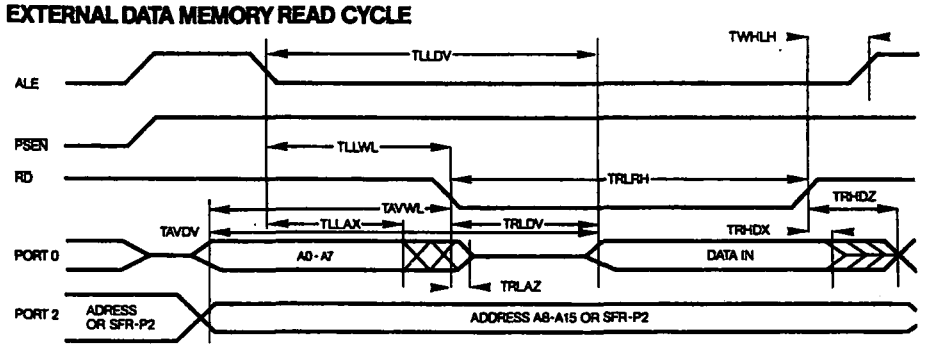
A: Address.
C: Clock.
D: Input data.
H: Logic level HIGH.
I: Instruction (program memory contents).
L: Logic level LOW, or ALE.
P: PSEN
Q: Output data.
R: READ signal.
T: Time.
V: Valid.
W: WRITE signal
X: No longer a valid logic level.
Z: Float.

AC TIMING DIAGRAMS

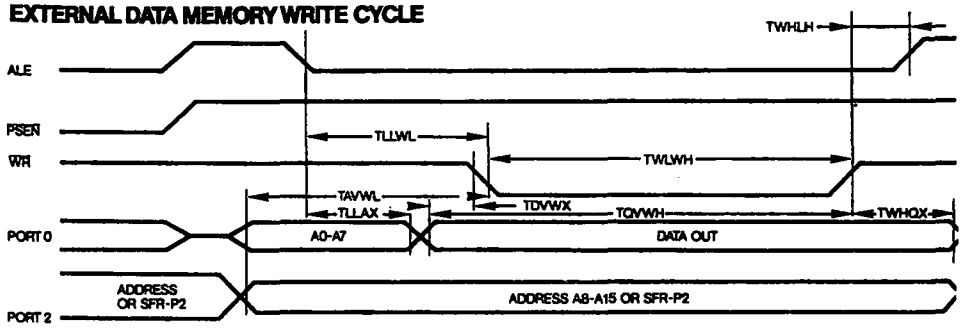
EXTERNAL PROGRAM MEMORY READ CYCLE



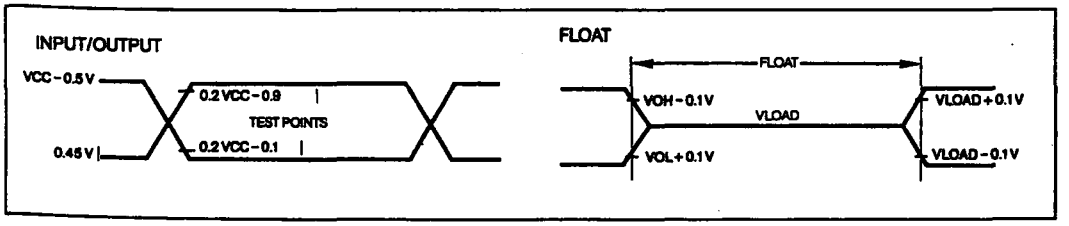
EXTERNAL DATA MEMORY READ CYCLE



EXTERNAL DATA MEMORY WRITE CYCLE



AC TESTING INPUT/OUTPUT, FLOAT WAVEFORMS



AC inputs during testing are driven at V_{CC} – 0.5 for a logic "1" and 0.45V for a logic "0". Timing measurements are made at VIH min for a logic "1" and VIL max for a logic "0". For timing purposes a port pin is no longer floating when a 100 mV change from load voltage occurs and begins to float when a 100 mV change from the loaded VOH/VOL level occurs. Iol/IoH ≥ ± 20 Ma.

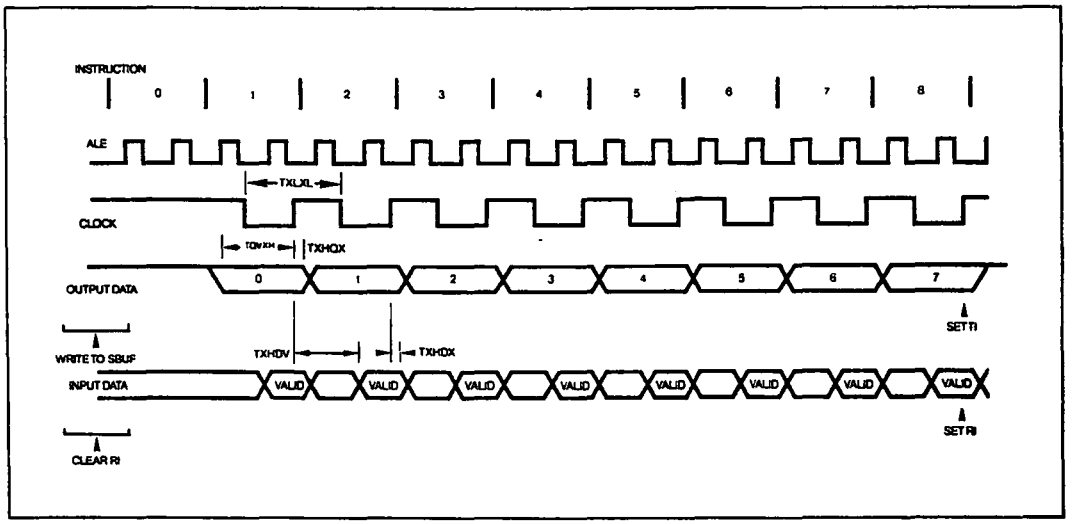
SERIAL PORT TIMING - SHIFT REGISTER MODE

A.C. CHARACTERISTICS:

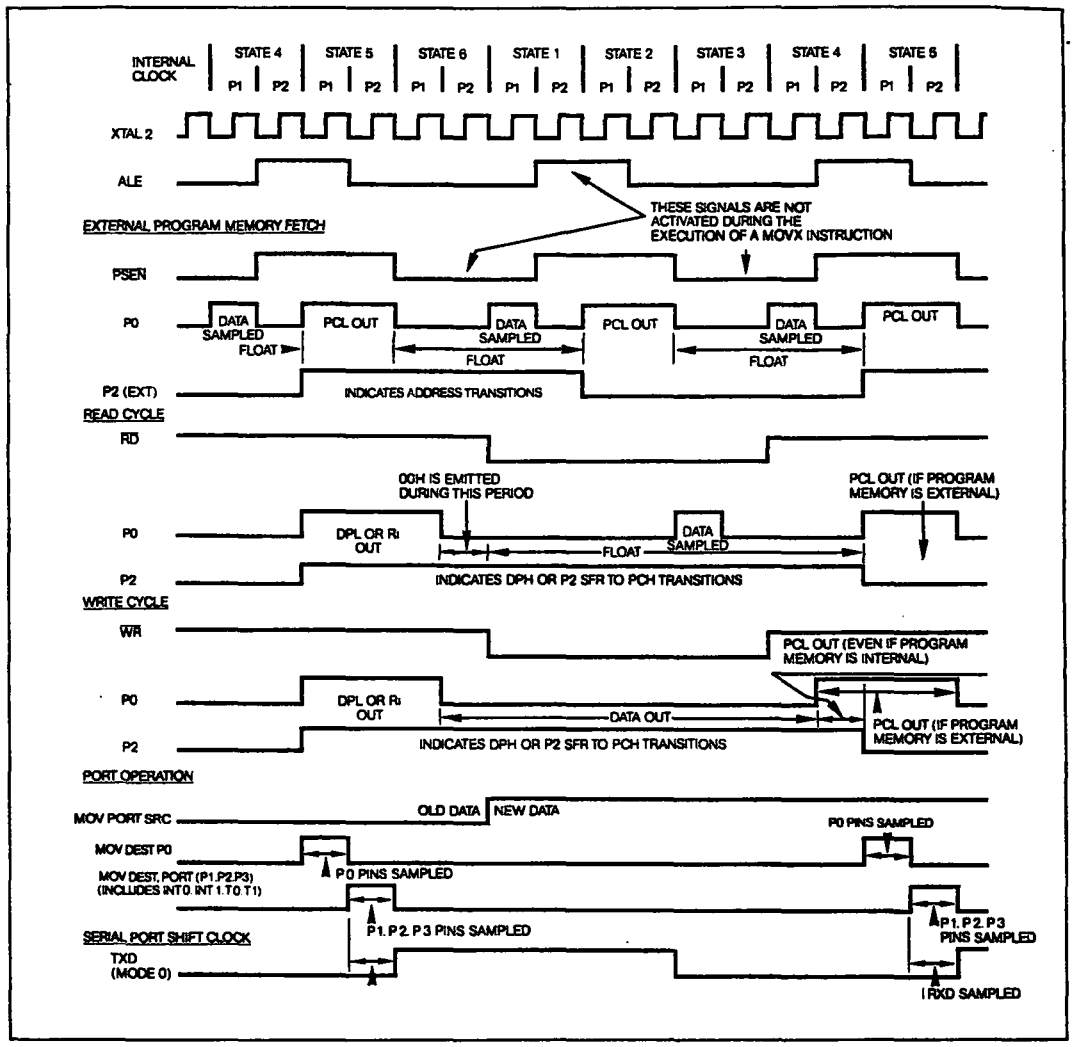
(T_A = 0°C to 70°C; V_{SS} = 0V; V_{CC} = 2.7V to 6V; Load Capacitance = 80 pF)

Symbol	Parameter	Min	Max	Units
TXLXL	Serial Port Clock Cycle Time	12TCLCL		µs
TQVXH	Output Data Setup to Clock Rising Edge	10TCLCL-133		ns
TXHQX	Output Data Hold After Clock Rising Edge	2TCLCL-117		ns
TXHDX	Input Data Hold After Clock Rising Edge	0		ns
TXHDV	Clock Rising Edge to Input Data Valid		10TLCL-133	ns

SHIFT REGISTER TIMING WAVEFORMS



CLOCK WAVEFORMS



This diagram indicates when signals are clocked internally. The time it takes the signals to propagate to the pins, however, ranges from 25 to 125 ns. This propagation delay is dependent on variables such as temperature and pin loading. Propagation also varies from output to output and component. Typically though ($T_A = 25°C$ fully loaded) RD and WR propagation delays are approximately 50 ns. The other signals are typically 85 ns. Propagation delays are incorporated in the AC specifications.

Table 1. MCS®-51 Instruction Set Description

ARITHMETIC OPERATIONS

Mnemonic		Description	Byte	Cyc
ADD	A,Rn	Add register to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1
ADD	A,#data	Add immediate data to Accumulator	2	1
ADDC	A,Rn	Add register to Accumulator with Carry	1	1
ADDC	A,direct	Add direct byte to A with Carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with Carry flag	1	1
ADDC	A,#data	Add immediate data to A with Carry flag	2	1
SUBB	A,Rn	Subtract register from A with Borrow	1	1
SUBB	A,direct	Subtract direct byte from A with Borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A with Borrow	1	1
SUBB	A,#data	Subtract immed. data from A with Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Incriment indirect RAM	1	1
INC	DPTR	Incriment Data Pointer	1	2
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
MUL	AB	Multiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1

LOGICAL OPERATIONS

Mnemonic		Destination	Byte	Cyc
ANL	A,Rn	AND register to Accumulator	1	1
ANL	A,direct	AND direct byte to Accumulator	2	1
ANL	A,@Ri	AND indirect RAM to Accumulator	1	1
ANL	A,#data	AND immediate data to Accumulator	2	1
ANL	direct,A	AND Accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to Accumulator	1	1
ORL	A,direct	OR direct byte to Accumulator	2	1
ORL	A,@Ri	OR indirect RAM to Accumulator	1	1
ORL	A,#data	OR immediate data to Accumulator	2	1
ORL	direct,A	OR Accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to Accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1
XRL	A,#data	Exclusive-OR immediate data to A	2	1
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator Left	1	1
RLC	A	Rotate A Left through the Carry flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate A Right through Carry flag	1	1
SWAP	A	Swap nibbles within the Accumulator	1	1

Table 1. (Cont.)

DATA TRANSFER

Mnemonic		Description	Byte	Cyc
MOV	A,Rn	Move register to Accumulator	1	1
MOV	A,direct	Move direct byte to Accumulator	2	1
MOV	A,@Ri	Move indirect RAM to Accumulator	1	1
MOV	A,#data	Move immediate data to Accumulator	2	1
MOV	Rn,A	Move Accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move Accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move Accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data 16	Load Data Pointer with a 16-bit constant	3	2
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to A	1	2
MOVC	A,@A+PC	Move Code byte relative to PC to A	1	2
MOVX	A,@Ri	Move External RAM (8-bit addr) to A	1	2
MOVX	A,@DPTR	Move External RAM (16-bit addr) to A	1	2
MOVX	@Ri,A	Move A to External RAM (8-bit addr)	1	2
MOVX	@DPTR,A	Move A to External RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte form stack	2	2
XCH	A,Rn	Exchange register with Accumulator	1	1
XCH	A,direct	Exchange direct byte with Accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order nibble ind RAM with A	1	1

BOOLEAN VARIABLE MANIPULATION

Mnemonic		Description	Byte	Cyc
CLR	C	Clear Carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set Carry flag	1	1
SETB	bit	Set direct Bit	2	1
CPL	C	Complement Carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to Carry flag	2	2
ANL	C,1 bit	AND complement of direct bit to Carry	2	2
ORL	C/bit	OR direct bit to Carry flag	2	2
ORL	C,1 bit	OR complement of direct bit to Carry	2	2
MOV	C/bit	Move direct bit to Carry flag	2	1
MOV	bit,C	Move Carry flag to direct bit	2	2

PROGRAM AND MACHINE CONTROL

Mnemonic		Description	Byte	Cyc
ACALL	addr 11	Absolute Subroutine Call	2	2
LCALL	addr 16	Long Subroutine Call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr 11	Absolute Jump	2	2
LJMP	addr 16	Long Jump	3	2
SJMP	rel	Short Jump (relative addr)	2	2
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if Accumulator is Zero	2	2
JNZ	rel	Jump if Accumulator is Not Zero	2	2
JC	rel	Jump if Carry flag is set	2	2
JNC	rel	Jump if No Carry flag	2	2

Table 1. (Cont.)

PROGRAM AND MACHINE CONTROL (cont.)

Mnemonic		Description	Byte	Cyc
JB	bit,rel	Jump if direct Bit set	3	2
JNB	bit,rel	Jump if direct Bit Not set	3	2
JBC	bit,rel	Jump if direct Bit is set & Clear bit	3	2
CJNE	A,direct,rel	Compare direct to A & Jump if Not Equal	3	2
CJNE	A,#data,rel	Comp. immed. to A & Jump if Not Equal	3	2
CJNE	Rn,#data,rel	Comp. immed. to reg & Jump if Not Equal	3	2
CJNE	@Ri,#data. rel	Comp. immed. to ind. & Jump if Not Equal	3	2
DJNZ	Rn,rel	Decrement register & Jump if Not Zero	2	2
DJNZ	direct. rel	Decrement direct & Jump if Not Zero	3	2
NOP		No operation	1	1

Notes on data addressing modes:

Rn	– Working register R0-R7
direct	– 128 internal RAM locations, any I/O port, control or status register
@Ri	– Indirect internal RAM location addressed by register R0 or R1
#data	– 8-bit constant included in instruction
#data 16	– 16-bit constant included as bytes 2 & 3 of instruction
bit	– 128 software flags, any I/O pin, control or status bit

Notes on program addressing modes:

addr 16	– Destination address for LCALL & LJMP may be anywhere within the 64-k program memory address space
Addr 11	– Destination address for ACALL & AJMP will be within the same 2-k page of program memory as the first byte of the following instruction
rel	– SJMP and all conditional jumps include an 8-bit offset byte. Range is +127–128 bytes relative to first byte of the following instruction.

All mnemonics copyrighted © Intel Corporation 1979

Table 2. Instruction Opcodes in Hexadecimal Order

Hex Code	Number of Bytes	Mnemonic	Operands
00	1	NOP	
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	bit addr,code addr
11	2	ACALL	code addr
12	3	LCALL	code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	data addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	bit addr,code addr
21	2	AJMP	code addr
22	1	RET	
23	1	RL	A
24	2	ADD	A,data
25	2	ADD	A,data addr
26	1	ADD	A,@R0
27	1	ADD	A,@R1
28	1	ADD	A,R0
29	1	ADD	A,R1
2A	1	ADD	A,R2
2B	1	ADD	A,R3
2C	1	ADD	A,R4
2D	1	ADD	A,R5
2E	1	ADD	A,R6
2F	1	ADD	A,R7
30	3	JNB	bit addr,code addr
31	2	ACALL	code addr
32	1	RETI	
33	1	RLC	A
34	2	ADDC	A,#data
35	2	ADDC	A,data addr
36	1	ADDC	A,@R0
37	1	ADDC	A,@R1
38	1	ADDC	A,R0
39	1	ADDC	A,R1
3A	1	ADDC	A,R2
3B	1	ADDC	A,R3
3C	1	ADDC	A,R4
3D	1	ADDC	A,R5
3E	1	ADDC	A,R6
3F	1	ADDC	A,R7
40	2	JC	code addr
41	2	AJMP	code addr
42	2	ORL	data addr,A
43	3	ORL	data addr,#data
44	2	ORL	A,#data
45	2	ORL	A,data addr
46	1	ORL	A,@R0
47	1	ORL	A,@R1
48	1	ORL	A,R0
49	1	ORL	A,R1
4A	1	ORL	A,R2
4B	1	ORL	A,R3
4C	1	ORL	A,R4
4D	1	ORL	A,R5
4E	1	ORL	A,R6
4F	1	ORL	A,R7
50	2	JNC	code addr
51	2	ACALL	code addr
52	2	ANL	data addr,A
53	3	ANL	data addr,#data
54	2	ANL	A,#data
55	2	ANL	A,data addr
56	1	ANL	A,@R0
57	1	ANL	A,@R1
58	1	ANL	A,R0
59	1	ANL	A,R1
5A	1	ANL	A,R2
5B	1	ANL	A,R3
5C	1	ANL	A,R4
5D	1	ANL	A,R5
5E	1	ANL	A,R6
5F	1	ANL	A,R7
60	2	JZ	code addr
61	2	AJMP	code addr
62	2	XRL	data addr A
63	3	XRL	data addr,#data
64	2	XRL	A,#data
65	2	XRL	A,data addr

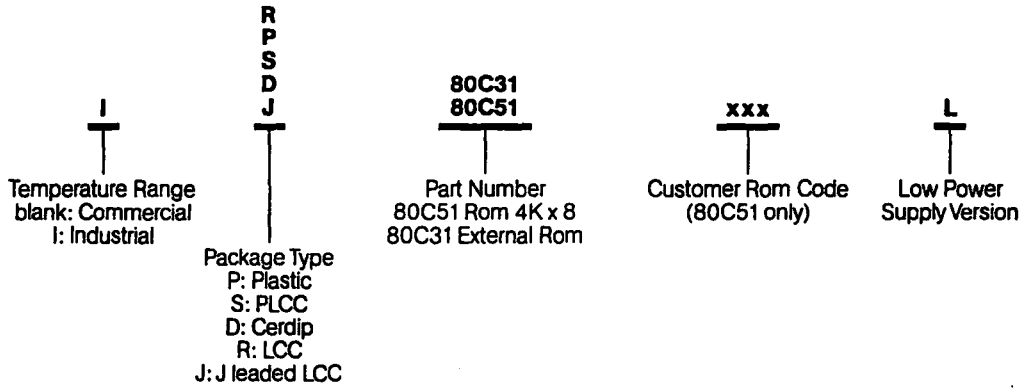
Table 2. (Cont.)

Hex Code	Number of Bytes	Mnemonic	Operands
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C, bit addr
73	1	JMP	@A+DPTR
74	2	MOV	A,#data
75	3	MOV	data addr,#data
76	2	MOV	@R0,#data
77	2	MOV	@R1,#data
78	2	MOV	R0,#data
79	2	MOV	R1,#data
7A	2	MOV	R2,#data
7B	2	MOV	R3,#data
7C	2	MOV	R4,#data
7D	2	MOV	R5,#data
7E	2	MOV	R6,#data
7F	2	MOV	R7,#data
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C,bit addr
83	1	MOVC	A,@A+PC
84	1	DIV	AB
85	3	MOV	data addr,data addr
86	2	MOV	data addr,@R0
87	2	MOV	data addr,@R1
88	2	MOV	data addr,R0
89	2	MOV	data addr,R1
8A	2	MOV	data addr,R2
8B	2	MOV	data addr,R3
8C	2	MOV	data addr,R4
8D	2	MOV	data addr,R5
8E	2	MOV	data addr,R6
8F	2	MOV	data addr,R7
90	3	MOV	DPTR,#data
91	2	ACALL	code addr
92	2	MOV	bit addr,C
93	1	MOVC	A,@A+DPTR
94	2	SUBB	A,#data
95	2	SUBB	A,data addr
96	1	SUBB	A,@R0
97	1	SUBB	A,@R1
98	1	SUBB	A,R0

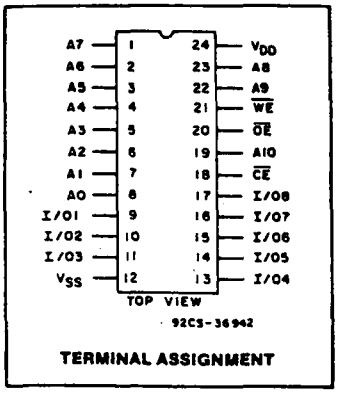
Hex Code	Number of Bytes	Mnemonic	Operands
99	1	SUBB	A,R1
9A	1	SUBB	A,R2
9B	1	SUBB	A,R3
9C	1	SUBB	A,R4
9D	1	SUBB	A,R5
9E	1	SUBB	A,R6
9F	1	SUBB	A,R7
A0	2	ORL	C,bit addr
A1	2	AJMP	code addr
A2	2	MOV	C,bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		reserved	
A6	2	MOV	@R0,data addr
A7	2	MOV	@R1,data addr
A8	2	MOV	R0,data addr
A9	2	MOV	R1,data addr
AA	2	MOV	R2,data addr
AB	2	MOV	R3,data addr
AC	2	MOV	R4,data addr
AD	2	MOV	R5,data addr
AE	2	MOV	R6,data addr
AF	2	MOV	R7,data addr
B0	2	ANL	C,bit addr
B1	2	ACALL	code addr
B2	2	CPL	bit addr
B3	1	CPL	C
B4	3	CJNE	A,#data,code addr
B5	3	CJNE	A,data addr,code addr
B6	3	CJNE	@R0,#data,code addr
B7	3	CJNE	@R1,#data,code addr
B8	3	CJNE	R0,#data,code addr
B9	3	CJNE	R1,#data,code addr
BA	3	CJNE	R2,#data,code addr
BB	3	CJNE	R3,#data,code addr
BC	3	CJNE	R4,#data,code addr
BD	3	CJNE	R5,#data,code addr
BE	3	CJNE	R6,#data,code addr
BF	3	CJNE	R7,#data,code addr
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A,data addr
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1
C8	1	XCH	A,R0
C9	1	XCH	A,R1
CA	1	XCH	A,R2
CB	1	XCH	A,R3

Table 2. (Cont.)

Hex Code	Number of Bytes	Mnemonic	Operands
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr,code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr
E0	1	MOVX	A,@DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR,A
F1	2	ACALL	code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A



CDM6116A



TERMINAL ASSIGNMENT

CMOS 2048-Word by 8-Bit Static RAM

Features:

- *Fully static operation*
- *Single power supply: 4.5 V to 5.5 V*
- *All inputs and outputs directly TTL compatible*
- *3-state outputs*
- *Industry standard 24-pin configuration*
- *Chip-enable gates address buffers for minimum standby current*
- *Data retention voltage: 2 V min.*

	CDM6116A-2	CDM6116A-3	CDM6116A-9
Access Time (max.)	200 ns	150 ns	250 ns
Output Enable Time (max.)	120 ns	60 ns	150 ns
Operating Temperature	0° to +70° C		-40° to +85° C
Operating Current (max.)	35 mA	35 mA	40 mA
Standby Current I_{DDS1} (max.)	30 μA	50 μA	100 μA

The RCA-CDM6116A is a CMOS 2048-word by 8-bit static random-access memory. It is designed for use in memory systems where high-speed, low power and simplicity in use are desirable. This device has common data inputs and data outputs and utilizes a single power supply of 4.5 V to 5.5 V. A chip-enable input and an output-enable input are provided for memory expansion and output buffer control.

The chip enable ($\overline{CE}$) gates the address and output buffers and powers down the chip to the low power standby mode.

The output enable ($\overline{OE}$) controls the output buffers to eliminate bus contention.

The CDM6116A-2 and CDM6116A-3 have an operating temperature range of 0° to +70° C. The CDM6116A-9 has an operating temperature range of -40° to +85° C.

The CDM6116A-2 and CDM6116A-3 are supplied in a 24-lead dual-in-line plastic package (E suffix). The CDM-6116A-9 is supplied in a 24-lead dual-in-line plastic package (E suffix) and a 24-lead dual-in-line side-brazed ceramic package (D suffix).

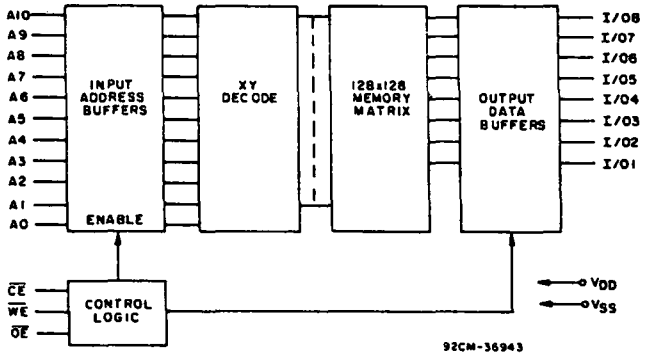


Fig. 1 - Functional block diagram.

TRUTH TABLE

$\overline{CE}$	$\overline{OE}$	$\overline{WE}$	A0 TO A10	MODE	I/01 TO I/08	DEVICE CURRENT
H	X	X	X	NOT SELECTED	HIGH Z	STANDBY
L	L	H	STABLE	READ	DATA OUT	ACTIVE
L	H	L	STABLE	WRITE	DATA IN	ACTIVE
L	L	L	STABLE	WRITE	DATA IN	ACTIVE

L = LOW H = HIGH X = H or L

MAXIMUM RATINGS, *Absolute-Maximum Ratings*

DC SUPPLY-VOLTAGE RANGE, (V_{DD}):
(Voltage referenced to V_{SS} terminal) ..... -0.3 to +7 V
INPUT VOLTAGE RANGE, ALL INPUTS ..... -0.3 to +7 V
DC INPUT CURRENT, ANY ONE INPUT ..... ±10 mA
POWER DISSIPATION PER PACKAGE (P_D):
For T_A = -40° to +60°C (PACKAGE TYPE E) ..... 500 mW
For T_A = +60° to +85°C (PACKAGE TYPE E) ..... Derate Linearly at 8 mW/°C to 300 mW
For T_A = -40° to +85°C (PACKAGE TYPE D) ..... 500 mW
DEVICE DISSIPATION PER OUTPUT TRANSISTOR
For T_A = FULL PACKAGE-TEMPERATURE RANGE (All Package Types) ..... 100 mW
OPERATING-TEMPERATURE RANGE (T_A)
CDM6116A-2, CDM6116A-3 (PACKAGE TYPE E) ..... 0 to +70°C
CDM6116A-9 (PACKAGE TYPES D, E) ..... -40 to +85°C
STORAGE TEMPERATURE RANGE (T_{stg}) ..... -55 to +125°C
LEAD TEMPERATURE (DURING SOLDERING):
At distance 1/16 ± 1/32 in. (1.59 ± 0.79 mm) from case for 10 s max. ..... +265°C

OPERATING CONDITIONS at T_A = 0 to +70°C, (CDM6116A-2, CDM6116A-3); T_A = -40° to +85°C (CDM6116A-9)

For maximum reliability, operating conditions should be selected so that operation is always within the following ranges:

CHARACTERISTIC		LIMITS ALL TYPES		UNITS
		MIN.	MAX.	
DC Operating Voltage Range		4.5	5.5	V
Input Voltage Range	V_{IH}	2.2	V_{DD} + 0.3	V
	V_{IL}	-0.3	0.8	V
Input Signal Rise or Fall Time Δ	t_r, t_f	—	5	μs

Δ Input signal rise and fall times longer than the maximum value can cause loss of stored data in the selected mode.

STATIC ELECTRICAL CHARACTERISTICS at T_A = 0 to +70°C (CDM6116A-2, CDM6116A-3); T_A = -40° to +85°C (CDM6116A-9), V_{DD} = 5 V ± 10%, Except as noted

CHARACTERISTIC		CONDITIONS	LIMITS									UNITS
			CDM6116A-2			CDM6116A-3			CDM6116A-9			
			MIN.	TYP.*	MAX.	MIN.	TYP.*	MAX.	MIN.	TYP.*	MAX.	
Standby Device Current	I_{DDS}	$\overline{CE}$ = V_{IH}	—	0.6	2	—	0.6	2	—	0.3	2	mA
	I_{DDS1}	$\overline{CE}$ = V_{DD}-0.2 V	—	1	30	—	1	50	—	1	100	μA
Output Voltage Low Level	V_{OL} Max.	I_{OL} = 2.1 mA	—	—	0.4	—	—	0.4	—	—	0.4	V
		I_{OL} = 1 μA	—	0.1	—	—	0.1	—	—	0.1	—	
Output Voltage High Level	V_{OH} Min.	I_{OH} = -1 mA	2.4	—	—	2.4	—	—	2.4	—	—	V
		I_{OH} = -1 μA	—	V_{DD}-0.1	—	—	V_{DD}-0.1	—	—	V_{DD}-0.1	—	
Input Leakage Current	I_{IN} Max.	V_{DD} = 5.5 V, V_{IN} = 0 V to V_{DD}	—	±0.1	±2	—	±0.1	±2	—	±0.1	±2	μA
3-State Output Leakage Current	I_{OUT}	$\overline{CE}$ or $\overline{OE}$ = V_{IH}, $V_{I/O}$ = 0 V to V_{DD}	—	±0.5	±2	—	±0.5	±2	—	±0.5	±2	
Operating Device Current	I_{OPER}#	V_{IN} = V_{IL}, V_{IH}	—	20	35	—	20	35	—	28	40	mA
Input Capacitance	C_{IN}	V_{IN} = 0 V, f = 1 MHz, T_A = 25°C	—	4	6	—	4	6	—	4	6	pF
Output Capacitance	$C_{I/O}$	$V_{I/O}$ = 0 V, f = 1 MHz, T_A = 25°C	—	6	8	—	6	8	—	6	8	

*Typical values are for T_A = 25°C and nominal V_{DD}.

#Outputs open circuited; cycle time = Min. t_{cycle}, duty = 100%.

CDM6116A

SIGNAL DESCRIPTIONS

A0-A10 (Address Inputs): These inputs must be stable prior to a write operation, but may change asynchronously during read operations.

I/01-I/08: 8-bit tristate data bus.

$\overline{CE}$ (Chip Enable): Powers down chip, disables Read and Write functions, and gates off address inputs.

$\overline{OE}$ (Output Enable): Enables tristate outputs if $\overline{CE}$ is low and $\overline{WE}$ is high.

$\overline{WE}$ (Write Enable): Enables Write function, if $\overline{CE}$ is low. $\overline{WE}$ will dominate if both $\overline{WE}$ and $\overline{OE}$ are low (i.e., the bus will be tristated and a Write will occur).

V_{DD}, V_{SS}: Power supply connections.

DYNAMIC ELECTRICAL CHARACTERISTICS at T_A = 0 to +70° C (CDM6116A-2, CDM6116A-3); T_A = -40° to +85° C (CDM6116A-9), V_{DD} = 5 V ± 10%, Input t_r, t_f = 10 ns; C_L = 100 pF and 1 TTL Load, Input Pulse Levels: 0.8 V to 2.4 V

CHARACTERISTIC Read Cycle Times See Fig. 2		LIMITS						UNITS
		CDM6116A-2		CDM6116A-3		CDM6116A-9		
		MIN.†	MAX.	MIN.†	MAX.	MIN.†	MAX.	
Read Cycle Time	t_{RC}	200	—	150	—	250	—	ns
Address Access Time	t_{AA}	—	200	—	150	—	250	
Chip Enable Access Time	t_{ACE}	—	200	—	150	—	250	
Chip Enable to Output Active	t_{CX}	15	—	15	—	15	—	
Output Enable to Output Valid	t_{OEV}	—	120	—	60	—	150	
Output Enable to Output Active	t_{OEX}	15	—	15	—	15	—	
Chip Disable to Output "High Z"	t_{CHZ}	0	60	0	50	0	80	
Output Disable to Output "High Z"	t_{OHZ}	0	60	0	50	0	80	
Output Hold from Address Change	t_{OH}	15	—	15	—	15	—	

†Time required by a limit device to allow for the indicated function.

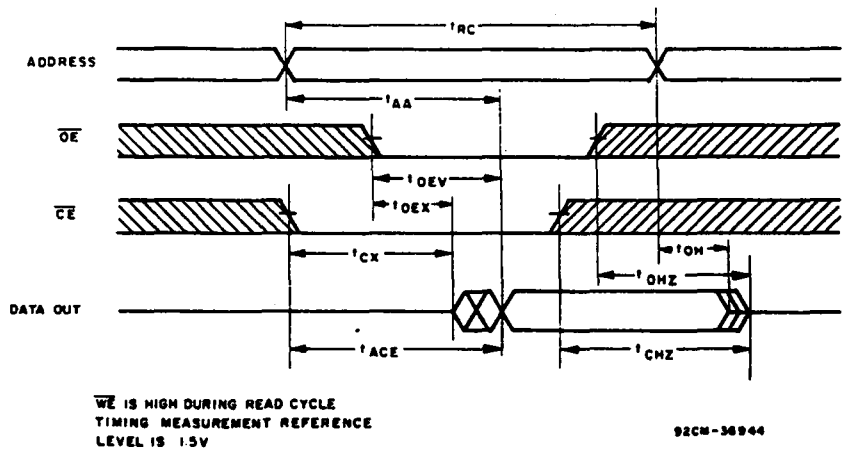


Fig. 2 - Read-cycle timing waveforms.

DYNAMIC ELECTRICAL CHARACTERISTICS at T_A = 0 to +70°C (CDM6116A-2, CDM6116A-3); T_A = -40° to +85°C (CDM6116A-9), V_{DD} = 5 V ± 10%, Input t_r, t_f = 10 ns; C_L = 100 pF and 1 TTL Load, Input Pulse Levels: 0.8 V to 2.4 V

CHARACTERISTIC		LIMITS						UNITS
		CDM6116A-2		CDM6116A-3		CDM6116A-9		
Write Cycle Times See Fig. 3		MIN.†	MAX.	MIN.†	MAX.	MIN.†	MAX.	
Write Cycle Time	t_{WC}	200	—	150	—	250	—	ns
Chip Enable to End of WRITE	t_{CW}	160	—	90	—	200	—	
Address Valid to End of WRITE	t_{AW}	160	—	90	—	200	—	
Address Setup Time	t_{AS}	0	—	0	—	0	—	
Write Pulse Width	t_{WP}	160	—	90	—	200	—	
Write Recovery Time	t_{WR}	10	—	0	—	10	—	
Output Disable to Output "High Z"	t_{OHZ}	0	60	0	50	0	80	
Write to Output "High Z"	t_{WHZ}	0	60	0	40	0	80	
Input Data Setup Time	t_{DW}	80	—	50	—	100	—	
Input Data Hold Time	t_{DH}	10	—	5	—	10	—	
Output Active from End of Write	t_{OW}	10	—	10	—	10	—	

†Time required by a limit device to allow for the indicated function.

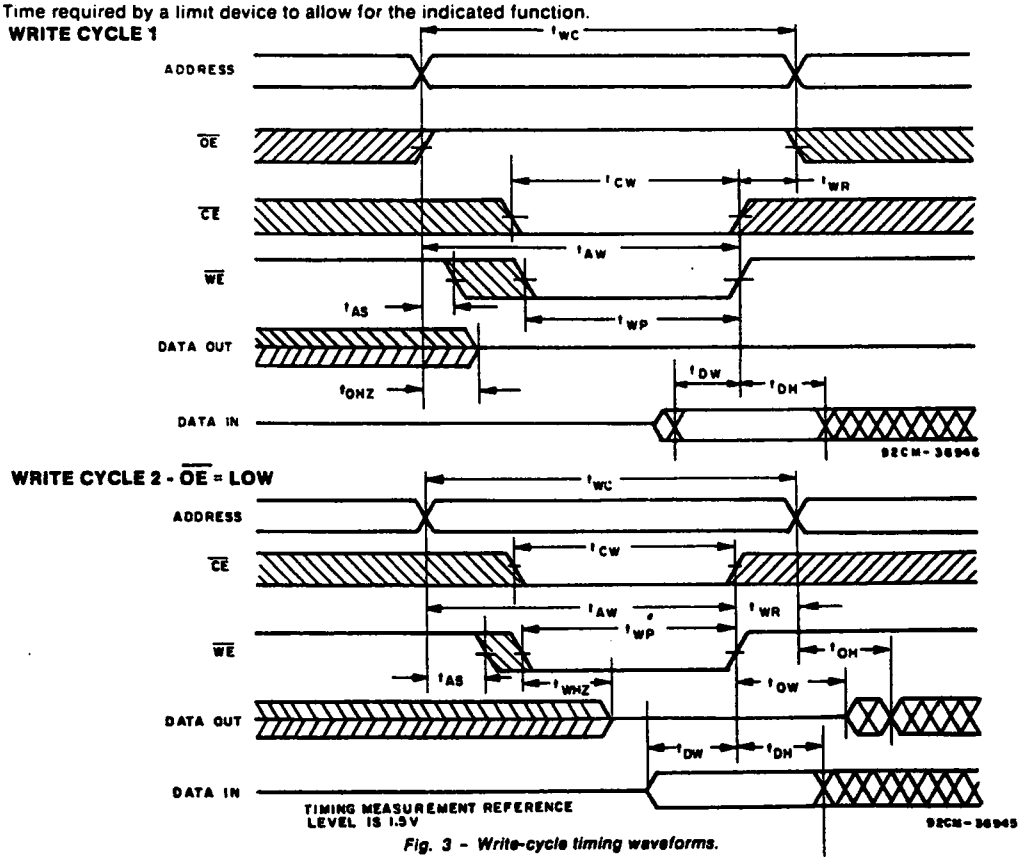


Fig. 3 - Write-cycle timing waveforms.

CDM6116A

DATA RETENTION CHARACTERISTICS at T_A = 0 to 70° C (CDM6116A-2, CDM6116A-3); T_A = -40 to +85° C (CDM6116A-9), Unless otherwise noted, See Fig. 4.

CHARACTERISTIC		TEST CONDITIONS	LIMITS		UNITS
			ALL TYPES		
			MIN.	MAX.	
Minimum Data Retention Voltage V_{DR} CDM6116A-2, CDM6116A-3, CDM6116A-9		T_A = 0 to 70° C, $\overline{CE} \geq V_{DD}$ - 0.2 V	2	—	V
	CDM6116A-9	T_A = -40 to 0° C, $\overline{CE} \geq V_{DD}$ - 0.2 V	4.5	—	
Data Retention Quiescent Current	$I_{DD}DR$*				
	CDM6116A-2	V_{DD} = 3 V, $\overline{CE} \geq$ 2.8 V	—	15	µA
	CDM6116A-3	V_{DD} = 3 V, $\overline{CE} \geq$ 2.8 V	—	25	
	CDM6116A-9	V_{DD} = 3 V, $\overline{CE} \geq$ 2.8 V	—	50	
Chip Disable to Data Retention Time	t_{CDR}	See Fig. 4	0	—	ns
Recovery to Normal Operation Time	t_R	See Fig. 4	*t_{RC}	—	

*$I_{DD}DR$=7.5 µA max. at T_A=0° to +40° C for CDM6116A-2 and CDM6116A-3.
*t_{RC} = Read Cycle Time.

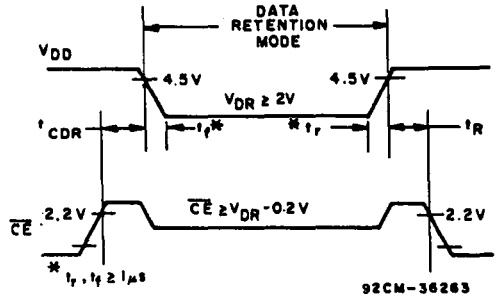


Fig. 4 - Low V_{DD} data retention timing waveforms.

DALLAS SEMICONDUCTOR

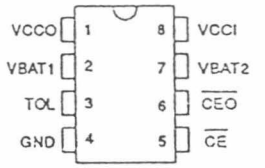
DS1210
Nonvolatile Controller Chip

FEATURES

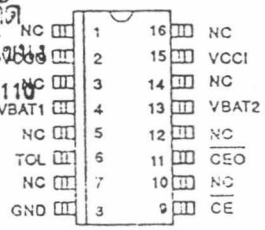
- Converts CMOS RAMs into nonvolatile memories
- Unconditionally write protects when V_{CC} is out of tolerance
- Automatically switches to battery when power fail occurs
- Space saving 8-pin DIP
- Consumes less than 100 nA of battery current
- Tests battery condition on power up
- Provides for redundant batteries
- Optional 5% or 10% power fail detection
- Low forward voltage drop on the V_{CC} switch
- Optional 16-pin SOIC surface mount package
- Optional industrial temperature range of -40°C to +85°C

SILA
บริษัท ศิลารีเลิร์ช จำกัด
1108/41 ศูนย์การค้าพระโขนง
ถนนสุขุมวิท กรุงเทพฯ 10110
โทร. 392-3886, 392-6183

PIN ASSIGNMENT



DS1210 8-Pin DIP (300 MIL)
See Mech. Drawing – Sect. 16, Pg. 1



DS1210S 16-Pin SOIC (300 MIL)
See Mech. Drawing – Sect. 16, Pg. 6

PIN DESCRIPTION

V_{CCO}	- RAM Supply
V_{BAT1}	- + Battery 1
TOL	- Power Supply Tolerance
GND	- Ground
$\overline{CE}$	- Chip Enable Input
$\overline{CEO}$	- Chip Enable Output
V_{BAT2}	- + Battery 2
V_{CCI}	- + Supply
NC	- No Connect

DESCRIPTION

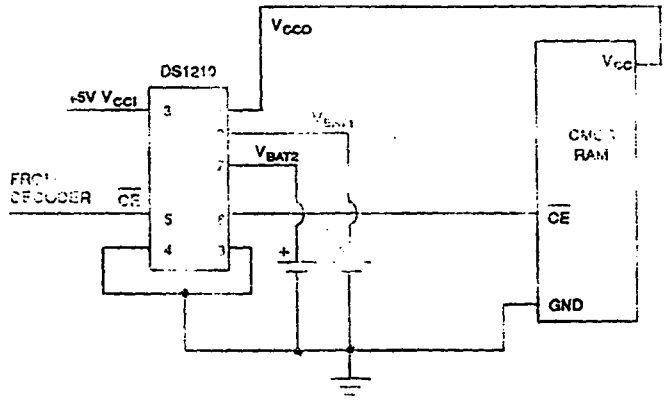
The DS1210 Nonvolatile Controller Chip is a CMOS circuit which solves the application problem of converting CMOS RAM into nonvolatile memory. Incoming power is monitored for an out-of-tolerance condition. When such a condition is detected, chip enable is inhibited to accomplish write protection and the battery is switched on to supply the RAM with uninterrupted power. Special circuitry uses a low-leakage CMOS process which affords precise voltage detection at extremely low battery consumption. The 8-pin DIP package keeps PC board real estate requirements to a minimum. By combining the DS1210 Nonvolatile Controller Chip with a CMOS memory and batteries, nonvolatile RAM operation can be achieved.

OPERATION

The DS1210 nonvolatile controller performs five circuit functions required to battery back up a RAM. First, a switch is provided to direct power from the battery or the incoming supply (V_{CCI}) depending on which is greater. This switch has a voltage drop of less than 0.3V. The second function which the nonvolatile controller provides is power fail detection. The DS1210 constantly monitors the incoming supply. When the supply goes out of tolerance a precision comparator detects power fail and inhibits chip enable ($\overline{CEO}$). The third function of write protection is accomplished by holding the $\overline{CEO}$ output signal to within 0.2 volts of the V_{CCI} or battery supply. If $\overline{CE}$ input is low at the time power fail detection occurs, the $\overline{CEO}$ output is kept in its present state until $\overline{CE}$ is returned high. The delay of write protection until the current memory cycle is completed prevents the corruption of data. Power fail detection occurs in the range of 4.75 volts to 4.5 volts with the tolerance Pin 3 grounded. If Pin 3 in connected to V_{CCO}, then power fail detection occurs in the range of 4.5 volts to 4.25 volts. During nominal supply conditions $\overline{CEO}$ will follow $\overline{CE}$ with a maximum propagation delay of 20ns. The fourth function the DS1210 performs is a battery status warning so that potential data loss is avoided. Each time that the circuit is powered up the battery voltage is checked with a precision comparator. If the battery voltage is less than 2.0 volts, the second memory cycle is inhibited. Battery status can, therefore, be determined by performing a read cycle after power-up to any location in memory, verifying that memory location content. A subsequent write cycle can then be executed to the same memory location altering the data. If the next read cycle fails to verify the written data, then the batteries are less than 2.0V and data is in danger of being corrupted. The fifth function of the nonvolatile controller provides for battery redundancy. In many applications, data integrity is paramount. In these applications it is often desirable to use two batteries to ensure reliability. The DS1210 controller provides an internal isolation switch which allows the connection of two batteries. During battery backup operation the battery with the highest voltage is selected for use. If one battery should fail, the other will take over the load. The switch to a redundant battery is transparent to circuit operation and to the user. A battery status warning will occur when the battery in use falls below 2.0 volts. A grounded V_{BAT2} pin will not activate a battery fail warning. In applications where battery redundancy is not required, a single battery should be connected to the BAT1 pin. The BAT2 battery pin must be grounded. The nonvolatile controller contains circuitry to turn off the battery back-up. This is to maintain the battery(s) at its highest capacity until the equipment is powered up and valid data is written to the SRAM. While in the freshness seal mode the $\overline{CEO}$ and V_{CCO} will be forced to V_{CL}. When the batteries are first attached to one or both of the V_{BAT} pins, V_{CCO} will not provide battery back-up until V_{CCI} exceeds V_{CCTP} as set by the T_{OL} pin, and then falls below V_{BAT}.

Figure 1 shows a typical application incorporating the DS1210 in a microprocessor-based system. Section A shows the connections necessary to write protect the RAM when V_{CC} is less than 4.75 volts and to back up the supply with batteries. Section B shows the use of the DS1210 to halt the processor when V_{CC} is less than 4.75 volts and to delay its restart on power-up to prevent spurious writes.

SECTION A – BATTERY BACKUP Figure 1



rrupted. The
provides for
data integrity
ten desirable
The DS1210
itch which al-
uring battery
ast voltage is
the other will
ant battery is
ser. A battery
y in use falls
not activate a
e battery re-
ry should be
tery pin must
tains circuit-
maintain the
equipment is
SRAM. While
V_{CCO} will be
t attached to
provide bat-
is set by the

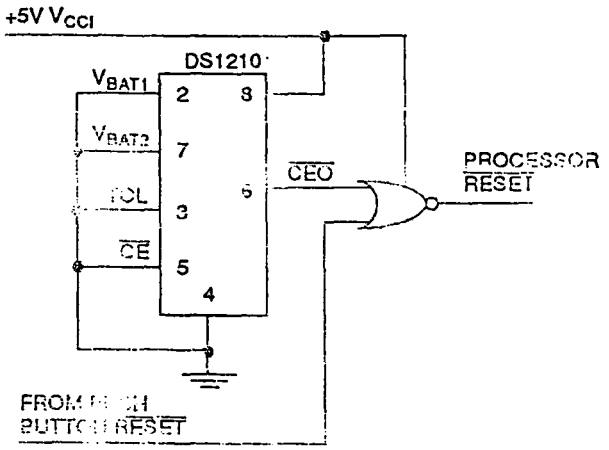
porting the
in Section A
protect the
o back up the
ne use of the
is less than
up to prevent

BATTERY BACKUP CURRENT DRAIN EXAMPLE

CONSUMPTION

DS1210 I_{BAT}	100 nA
RAM I_{CCO2}	10 µA
Total Drain	10.1 µA

SECTION B - PROCESSOR RESET



ABSOLUTE MAXIMUM RATINGS*

Voltage on any Pin Relative to Ground	-0.3V to +7.0V
Operating Temperature	0°C to 70°C
Storage Temperature	-55°C to +125°C
Soldering Temperature	260°C for 10 seconds

* This is a stress rating only and functional operation of the device at these or any other conditions outside those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods of time may affect reliability.

RECOMMENDED DC OPERATING CONDITIONS (0°C to 70°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Pin 8 Supply Voltage	V_{CC1}	4.75	5.0	5.5	V	1
Pin 3 Supply Voltage	V_{CC2}	4.75	5.0	5.5	V	1
Logic 1 Input	V_{IH}	2.2		V_{CC}+0.3	V	1
Logic 0 Input	V_{IL}	-0.3		+0.8	V	1
Battery Input	V_{BAT1}, V_{BAT2}	2.0		4.0	V	1,3

DC ELECTRICAL CHARACTERISTICS

(0°C to 70°C, V_{CCI} = 4.75V to 5.5V, Pin 3 = GND)
(V_{CCI} = 4.5 to 5.5V, Pin 3 = V_{CCO})

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Supply Current	I_{CCI}			5	mA	3
Supply Voltage	V_{CCO}	V_{CC}-0.2			V	1
Supply Current	I_{CCO1}			80	mA	4
Input Leakage	I_{IL}	-1.0		+1.0	µA	
Output Leakage	I_{LO}	-1.0		+1.0	µA	
$\overline{CEO}$ Output @2.4V	I_{OH}	-1.0			mA	5
$\overline{CEO}$ Output @0.4V	I_{OL}			4.0	mA	5
V_{CC} Trip Point (TOL=GND)	V_{CCTP}	4.50	4.62	4.74	V	1
V_{CC} Trip Point (TOL=V_{CCO})	V_{CCTP}	4.25	4.37	4.49	V	1

(0°C to 70°C, V_{CCI} = < V_{BAT})

$\overline{CEO}$ Output	V_{OHL}	V_{BAT}-0.2			V	7
V_{BAT1} or V_{BAT2} Battery Current	I_{BAT}			100	nA	2,3
Battery Backup Current @ V_{cco} = V_{BAT} - 0.3V	I_{CCO2}			50	µA	6,7

CAPACITANCE

(t_A = 25°C)

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
Input Capacitance	C_{IN}			5	pF	
Output Capacitance	C_{OUT}			7	pF	

AC ELECTRICAL CHARACTERISTICS

(0°C to 70°C, V_{CCI} = 4.75V to 5.5V, Pin 3 = GND)
(V_{CCI} = 4.5 to 5.5V, Pin3 = V_{CCO})

PARAMETER	SYMBOL	MIN	TYP	MAX	UNITS	NOTES
$\overline{CE}$ Propagation Delay	t_{PD}	5	10	20	ns	5
$\overline{CE}$ High to Power Fail	t_{PF}			0	ns	

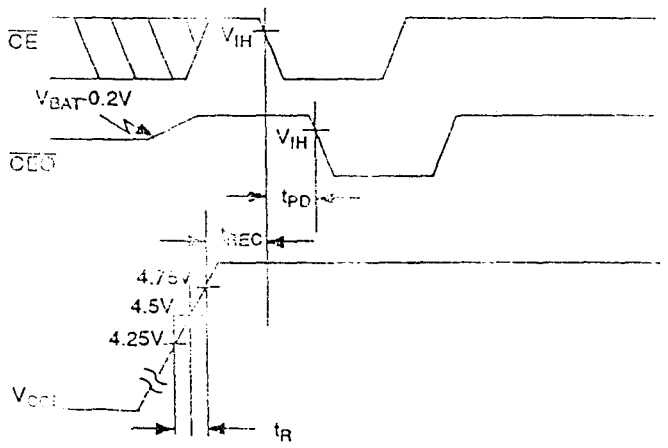
(0°C to 70°C, V_{CCI} < 4.75V, Pin 3 = GND; V_{CCI} < 4.5, Pin 3 = V_{CCO})

Recovery at Power Up	t_{REC}	2	50	125	ms	
V_{CC} Slew Rate Power Down	t_F	300			µs	
V_{CC} Slew Rate Power Down	t_{FB}	10			µs	
V_{CC} Slew Rate Power Up	t_R	0			µs	
$\overline{CE}$ Pulse Width	t_{CE}			1.5	µs	8

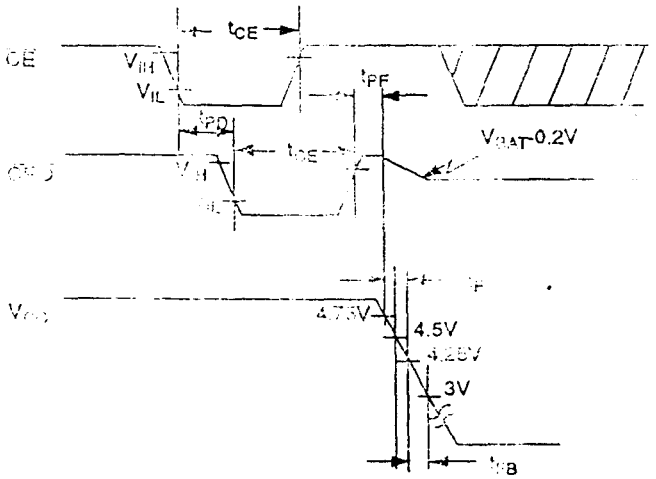
NOTES

1. All voltages are referenced to ground.
2. Only one battery input is required. Unused battery inputs must be grounded.
3. Measured with V_{CCO} and $\overline{CEO}$ open.
4. I_{CC01} is the maximum average load which the DS1210 can supply to the memories.
5. Measured with a load as shown in Figure 2.
6. I_{CC02} is the maximum average load current which the DS1210 can supply to the memories in the battery back-up mode.
7. t_{CE} max. must be met to ensure data integrity on power loss.
8. $\overline{CEO}$ can only sustain leakage current in the battery backup mode.

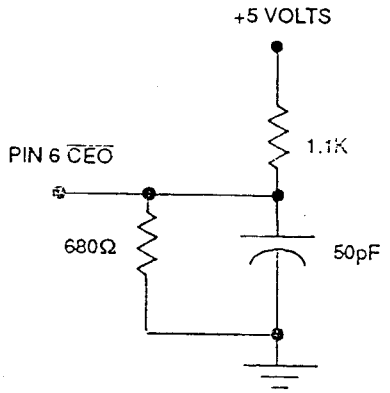
TIMING DIAGRAM - POWER UP



TIMING DIAGRAM - POWER DOWN



OUTPUT LOAD Figure 2



8255A/8255A-5
PROGRAMMABLE PERIPHERAL INTERFACE

- MCS-85™ Compatible 8255A-5
- 24 Programmable I/O Pins
- Completely TTL Compatible
- Fully Compatible with Intel® Microprocessor Families
- Improved Timing Characteristics
- Direct Bit Set/Reset Capability Easing Control Application Interface
- Reduces System Package Count
- Improved DC Driving Capability
- Available in EXPRESS
  - —Standard Temperature Range
  - —Extended Temperature Range

The Intel® 8255A is a general purpose programmable I/O device designed for use with Intel® microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.

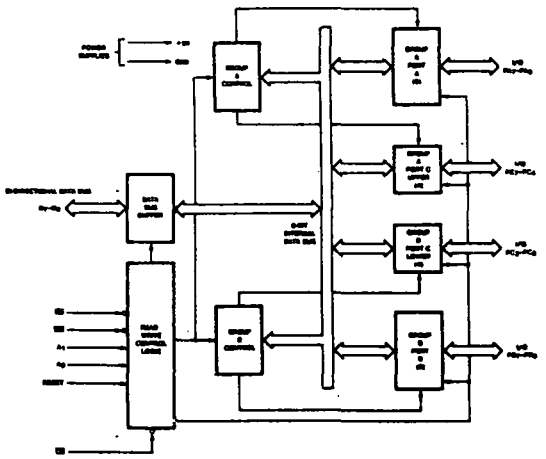


Figure 1. 8255A Block Diagram

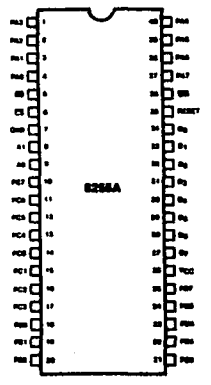


Figure 2. Pin Configuration

©INTEL CORPORATION, 1982

Order Number: 231306-001

8255A FUNCTIONAL DESCRIPTION

General

The 8255A is a programmable peripheral interface (PPI) device designed for use in Intel® microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 8255A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 8255A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

($\overline{CS}$)

Chip Select. A "low" on this input pin enables the communiction between the 8255A and the CPU.

($\overline{RD}$)

Read. A "low" on this input pin enables the 8255A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 8255A.

($\overline{WR}$)

Write. A "low" on this input pin enables the CPU to write data or control words into the 8255A.

(A_0 and A_1)

Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A_0 and A_1).

8255A BASIC OPERATION

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	INPUT OPERATION (READ)
0	0	0	1	0	PORT A – DATA BUS
0	1	0	1	0	PORT B – DATA BUS
1	0	0	1	0	PORT C – DATA BUS
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	DATA BUS – PORT A
0	1	1	0	0	DATA BUS – PORT B
1	0	1	0	0	DATA BUS – PORT C
1	1	1	0	0	DATA BUS – CONTROL
					DISABLE FUNCTION
X	X	X	X	1	DATA BUS – 3-STATE
1	1	0	1	0	ILLEGAL CONDITION
X	X	1	1	0	DATA BUS – 3-STATE

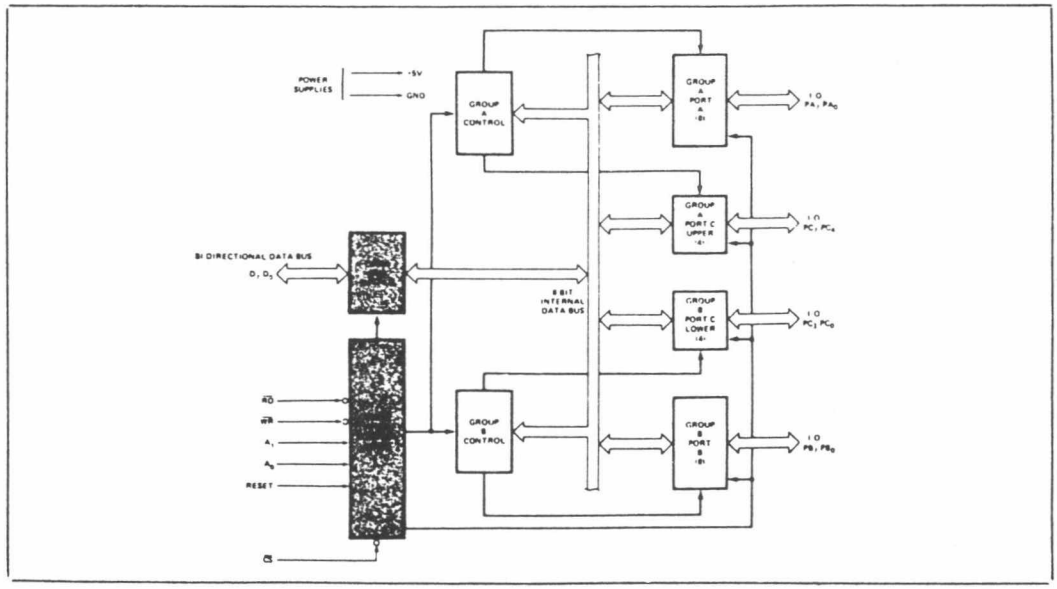


Figure 3. 8255A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Fupctions

231308-001

(RESET)

Reset. A "high" on this input clears the control register and all ports (A, B, C) are set to the input mode.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 8255A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A – Port A and Port C upper (C7-C4)
Control Group B – Port B and Port C lower (C3-C0)

The Control Word Register can Only be written into. No Read operation of the Control Word Register is allowed.

Ports A, B, and C

The 8255A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255A.

Port A. One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B. One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.

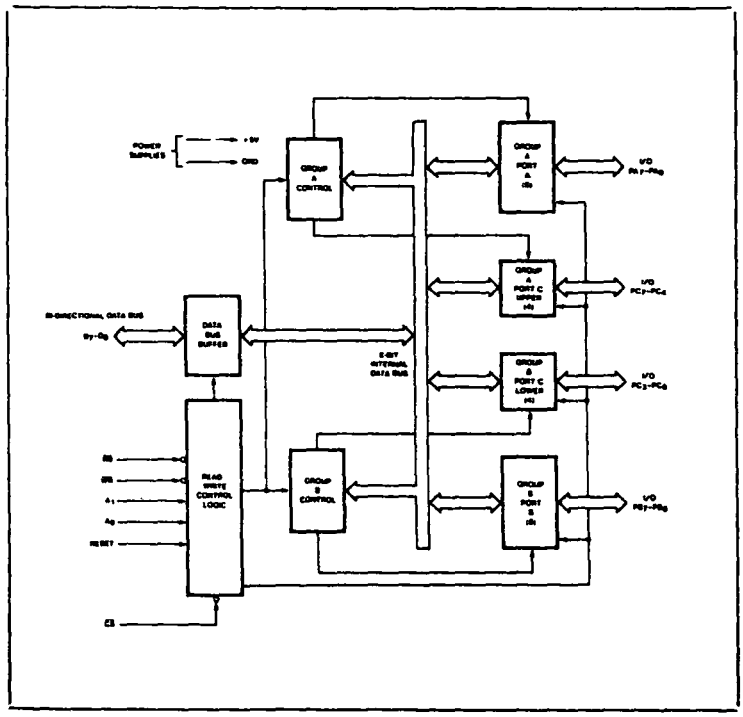
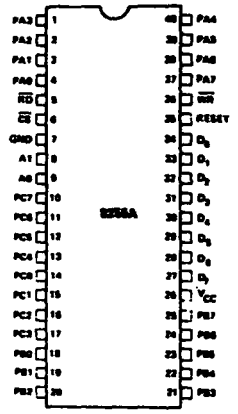


Figure 4. 8225A Block Diagram Showing Group A and Group B Control Functions

PIN CONFIGURATION



PIN NAMES

D_7-D_0	DATA BUS (BI-DIRECTIONAL)
RESET	RESET INPUT
$\overline{CS}$	CHIP SELECT
$\overline{RD}$	READ INPUT
$\overline{WR}$	WRITE INPUT
A0, A1	PORT ADDRESS
PA7-PA0	PORT A (BIT)
PB7-PB0	PORT B (BIT)
PC7-PC0	PORT C (BIT)
V_{CC}	+5 VOLTS
GND	0 VOLTS

231308-001

8255A OPERATIONAL DESCRIPTION

Mode Selection

There are three basic modes of operation that can be selected by the system software:

Mode 0 – Basic Input/Output
Mode 1 – Strobed Input/Output
Mode 2 – Bi-Directional Bus

When the reset input goes "high" all ports will be set to the input mode (i.e., all 24 lines will be in the high impedance state). After the reset is removed the 8255A can remain in the input mode with no additional initialization required. During the execution of the system program any of the other modes may be selected using a single output instruction. This allows a single 8255A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

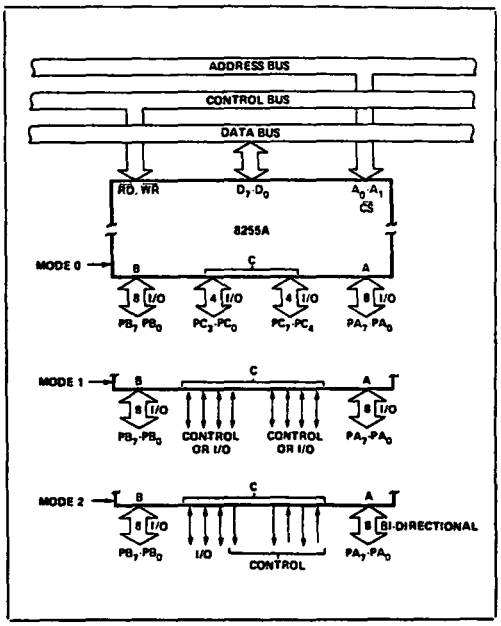


Figure 5. Basic Mode Definitions and Bus Interface

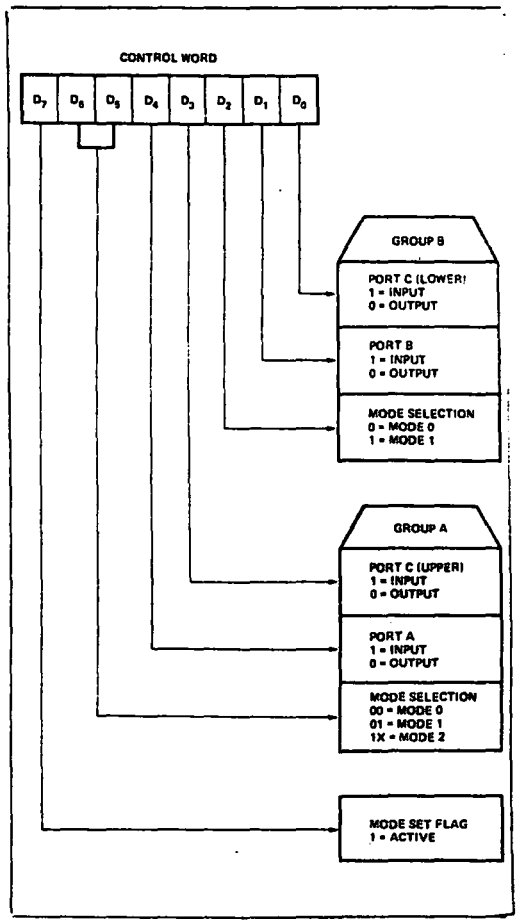


Figure 6. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 8255A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.

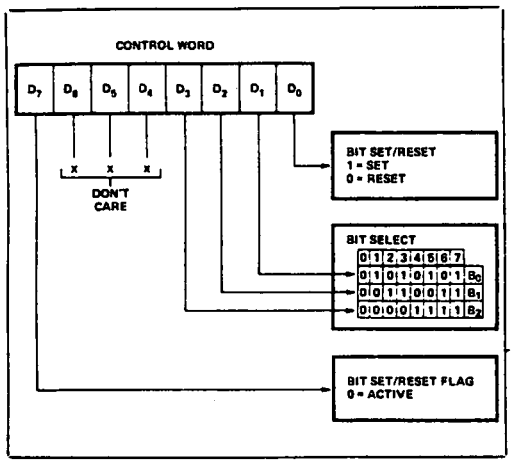


Figure 7. Bit Set/Reset Format

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the 8255A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET) — INTE is SET — Interrupt enable
(BIT-RESET) — INTE is RESET — Interrupt disable

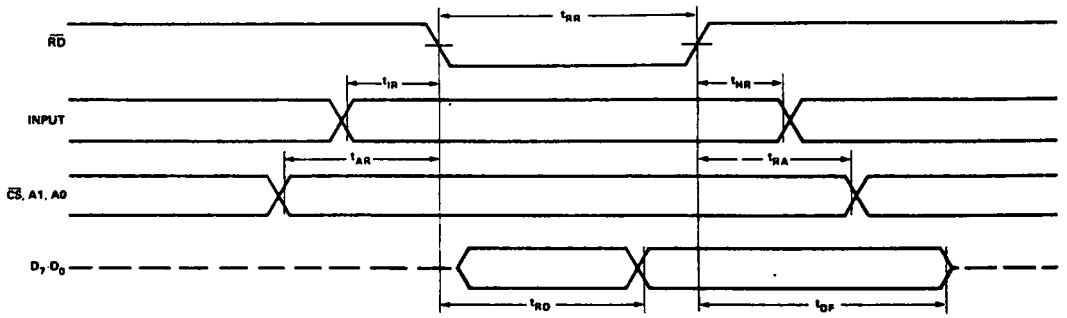
Note: All Mask flip-flops are automatically reset during mode selection and device Reset.

Operating Modes

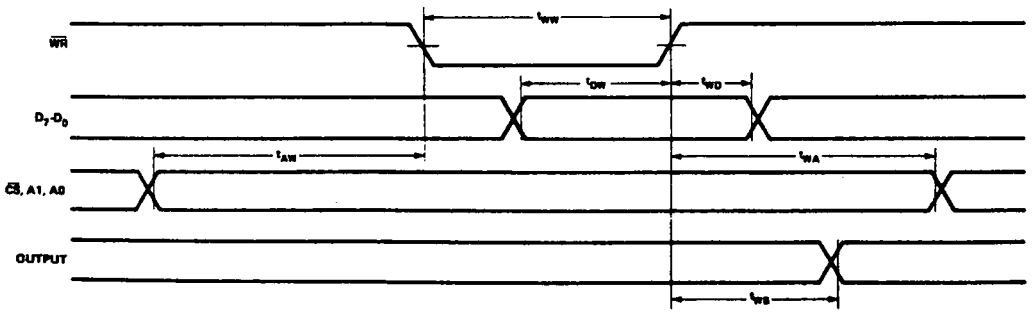
MODE 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.



MODE 0 (Basic Input)

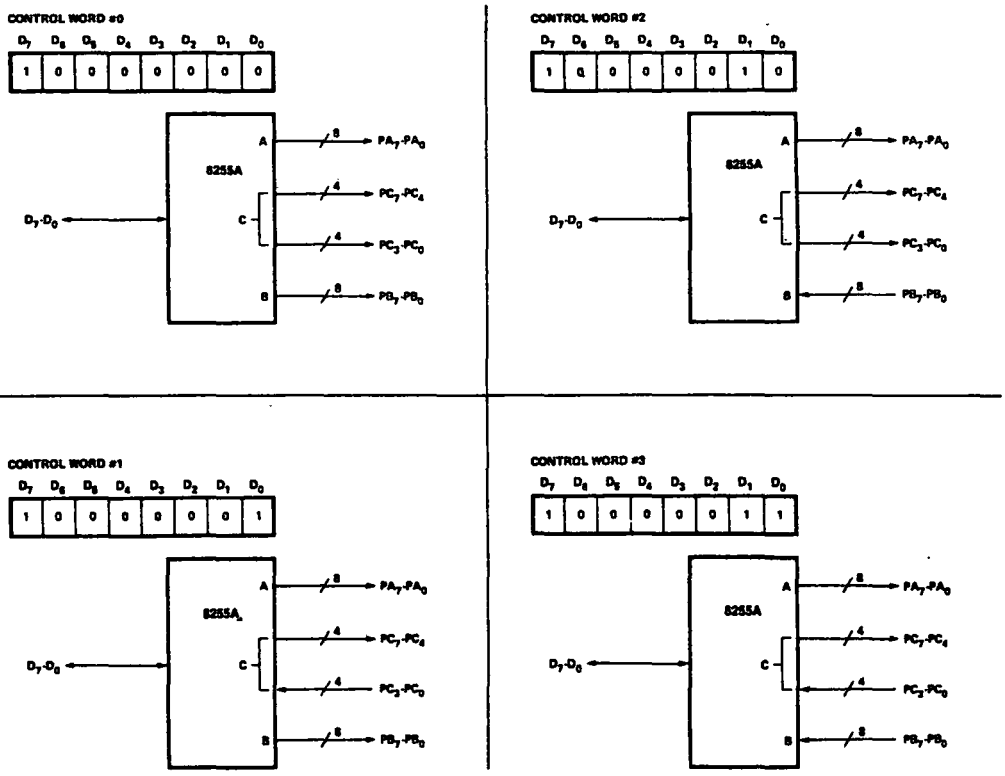


MODE 0 (Basic Output)

MODE 0 Port Definition

A		B		GROUP A			GROUP B	
D_4	D_3	D_1	D_0	PORT A	PORT C (UPPER)	#	PORT B	PORT C (LOWER)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

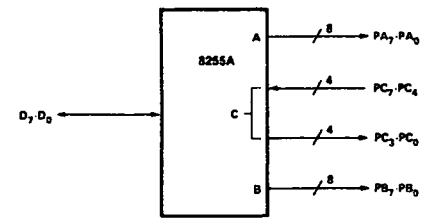
MODE 0 Configurations



231308-001

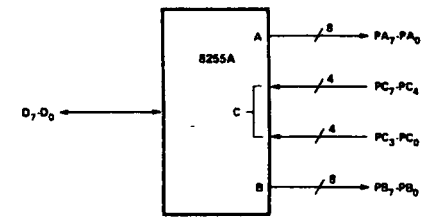
CONTROL WORD #4

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	0



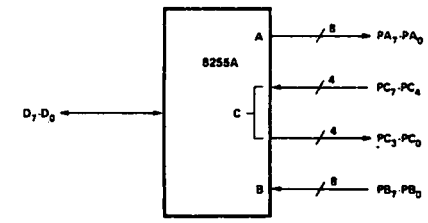
CONTROL WORD #5

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	1



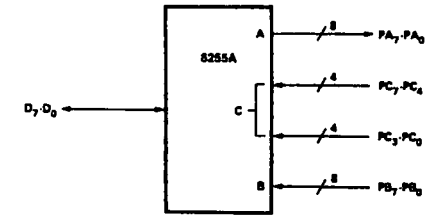
CONTROL WORD #6

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	0



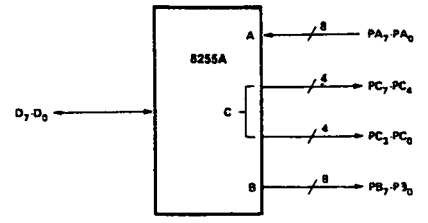
CONTROL WORD #7

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	1



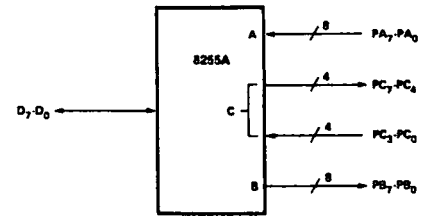
CONTROL WORD #8

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	0



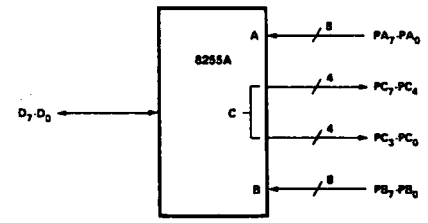
CONTROL WORD #9

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	1



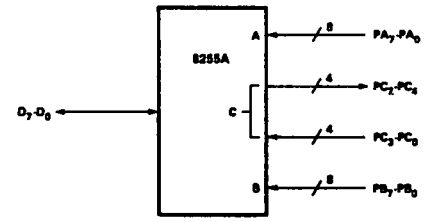
CONTROL WORD #10

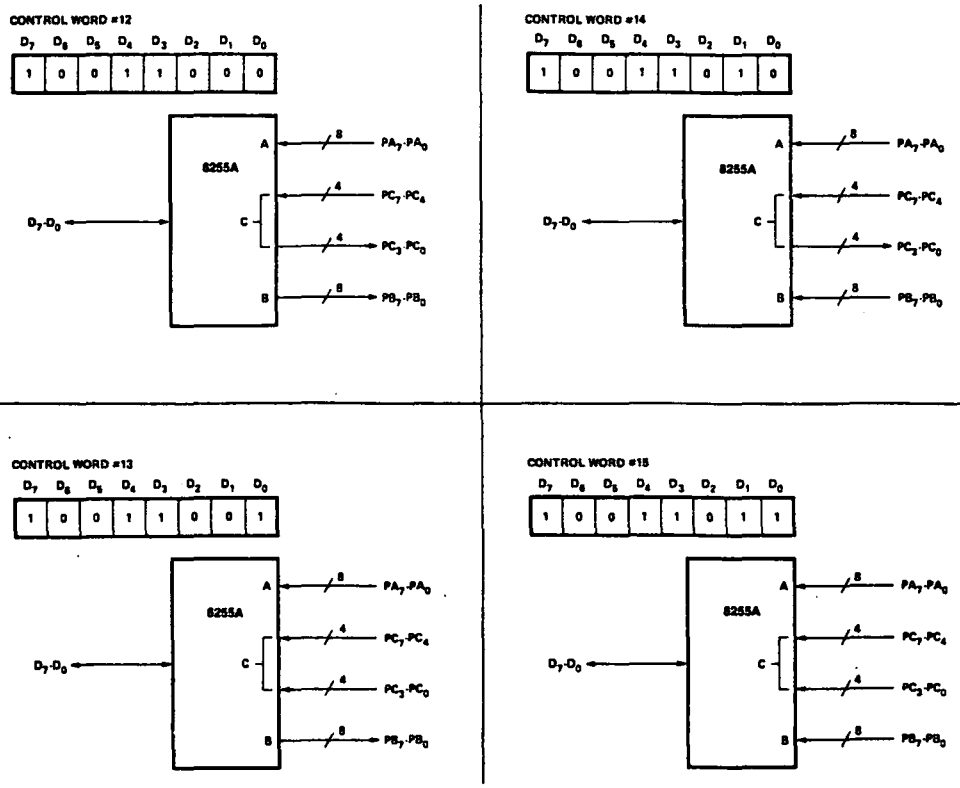
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	0



CONTROL WORD #11

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	1





Operating Modes

MODE 1 (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In mode 1, port A and Port B use the lines on port C to generate or accept these "handshaking" signals.

Mode 1 Basic Functional Definitions:

- Two Groups (Group A and Group B)
- Each group contains one 8-bit data port and one 4-bit control/data port.
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

Input Control Signal Definition

$\overline{STB}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by STB input being low and is reset by the rising edge of the RD input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the $\overline{STB}$ is a "one", IBF is a "one" and INTE is a "one" It is reset by the falling edge of $\overline{RD}$. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A

Controlled by bit set/reset of PC_4.

INTE B

Controlled by bit set/reset of PC_2.

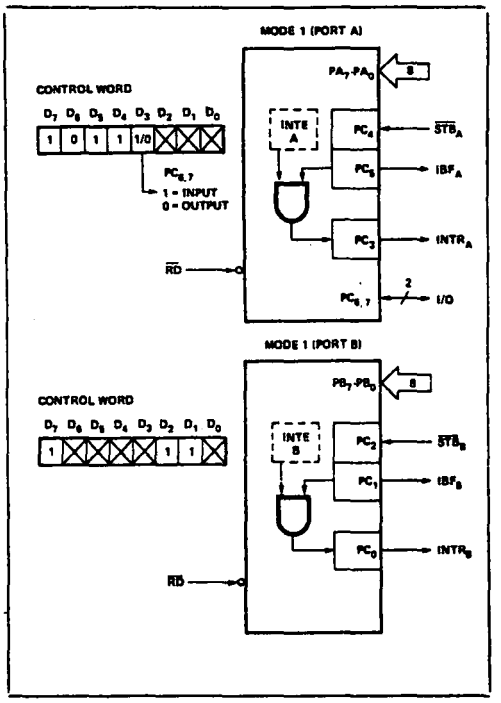


Figure 8. MODE 1 Input

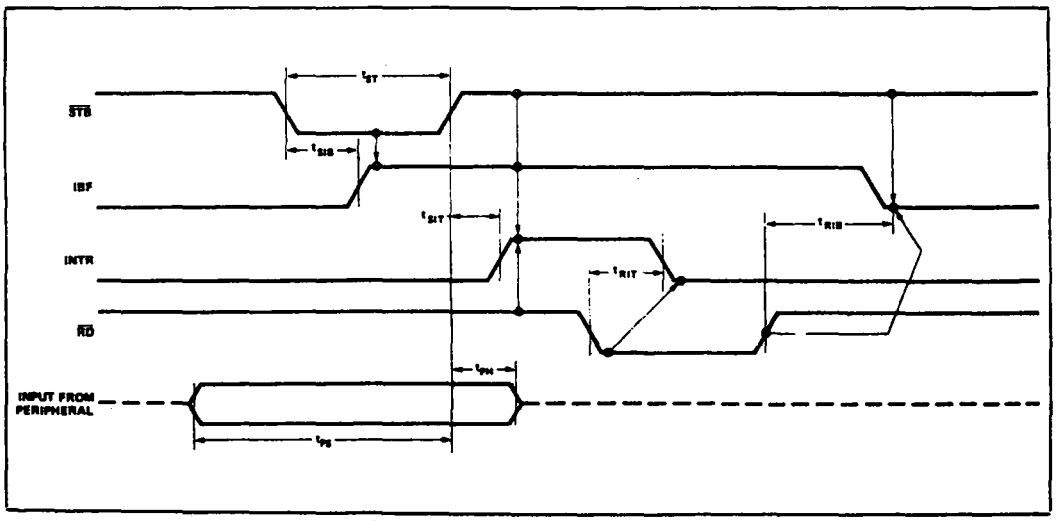


Figure 9. MODE 1 (Strobed Input)

Output Control Signal Definition

$\overline{OBF}$ (Output Buffer Full F/F). The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to the specified port. The $\overline{OBF}$ F/F will be set by the rising edge of the $\overline{WR}$ input and reset by $\overline{ACK}$ input being low.

$\overline{ACK}$ (Acknowledge Input). A "low" on this input informs the 8255A that the data from port A or port B has been accepted. In essence, a response from the peripheral device indicating that it has recieved the data output by the CPU.

INTR (Interrupt Request). A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when $\overline{ACK}$ is a "one", $\overline{OBF}$ is a "one", and INTE is a "one". It is reset by the falling edge of $\overline{WR}$.

INTR (Interrupt Request). A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when $\overline{ACK}$ is a "one", $\overline{OBF}$ is a "one", and INTE is a "one". It is reset by the falling edge of WR.

INTE A

Controlled by bit set/reset of PC_6

INTE B

Controlled by bit set/reset of PC_2

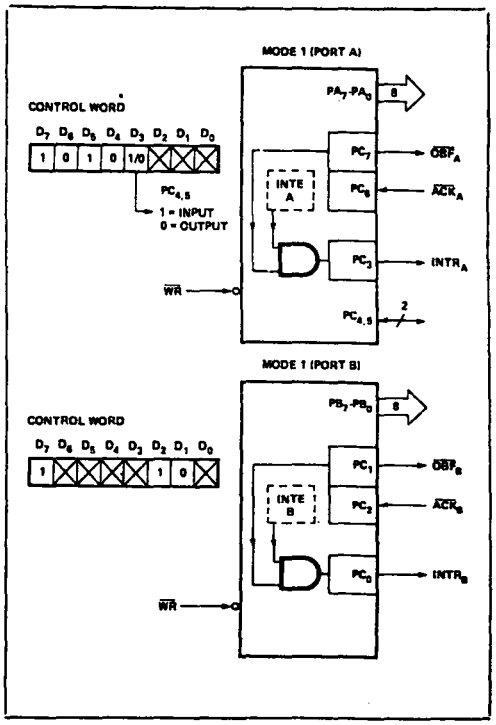


Figure 10. MODE 1 Output

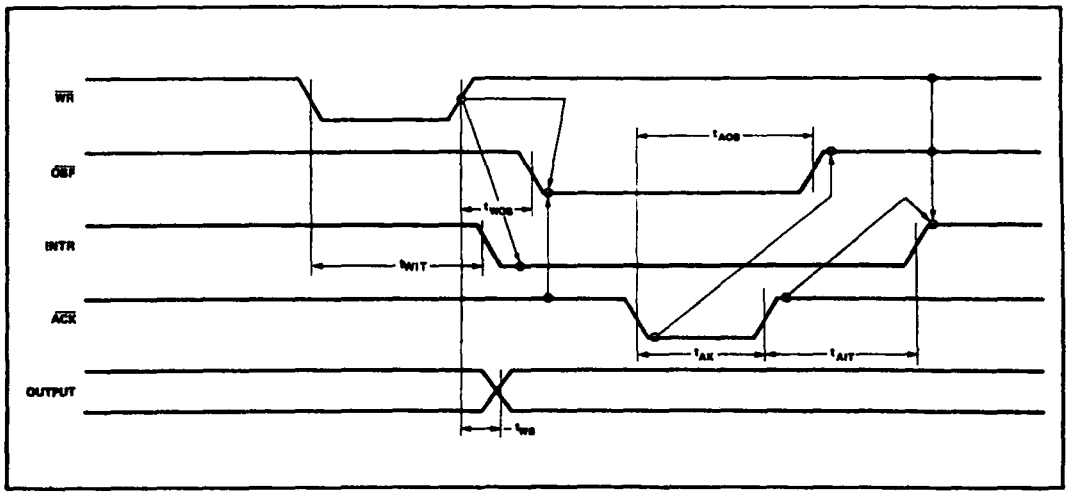


Figure 11. Mode 1 (Strobed Output)

Combinations of MODE 1

Port A and Port B can be individually defined as input or output in Mode 1 to support a wide variety of strobed I/O applications.

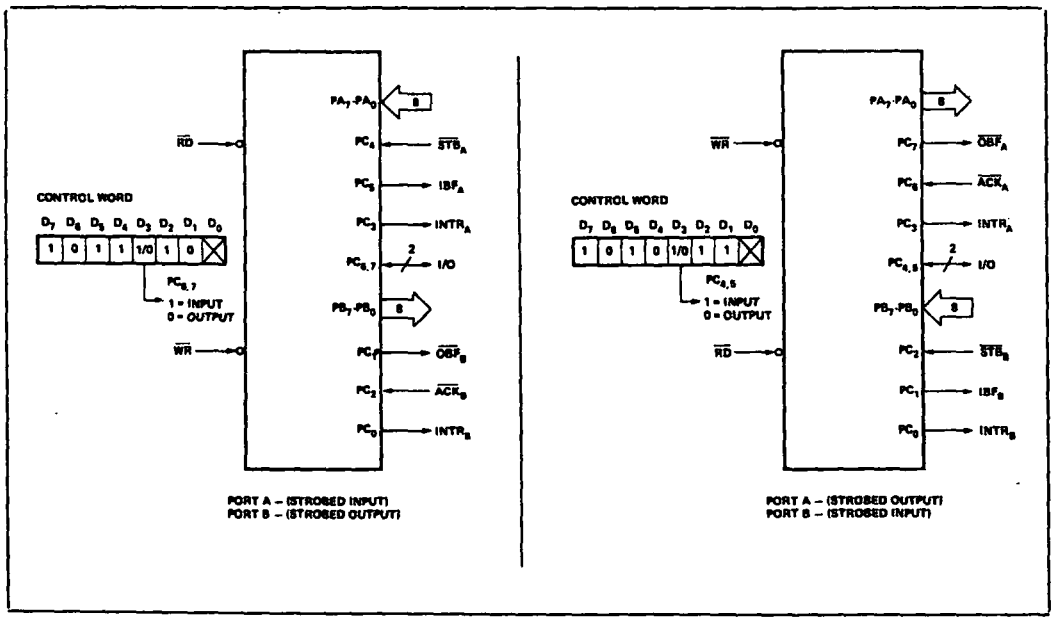


Figure 12. Combinations of MODE 1

Operating Modes

MODE 2 (Strobed Bidirectional Bus I/O). This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to MODE 1. Interrupt generation and enable/disable functions are also available.

MODE 2 Basic Functional Definitions:

- Used in Group A only.
- One 8-bit, bi-directional bus Port (Port A) and a 5-bit control Port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Bidirectional Bus I/O Control Signal Definition

INTR (Interrupt Request). A high on this output can be used to interrupt the CPU for both input or output operations.

Output Operations

$\overline{OBF}$ (Output Buffer Full). The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to port A.

$\overline{ACK}$ (Acknowledge). A "low" on this input enables the tri-state output buffer of port A to send out the data. Otherwise, the output buffer will be in the high impedance state.

INTE 1 (The INTE Flip-Flop Associated with OBF). Controlled by bit set/reset of PC_6.

Input Operations

$\overline{STB}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F). A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE Flip-Flop Associated with IBF). Controlled by bit set/reset of PC_4.

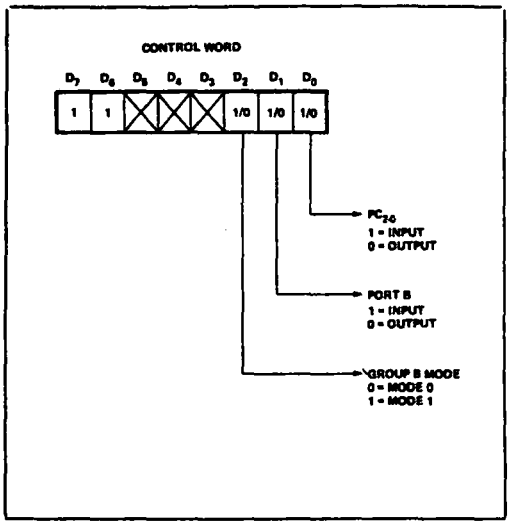


Figure 13. MODE Control Word

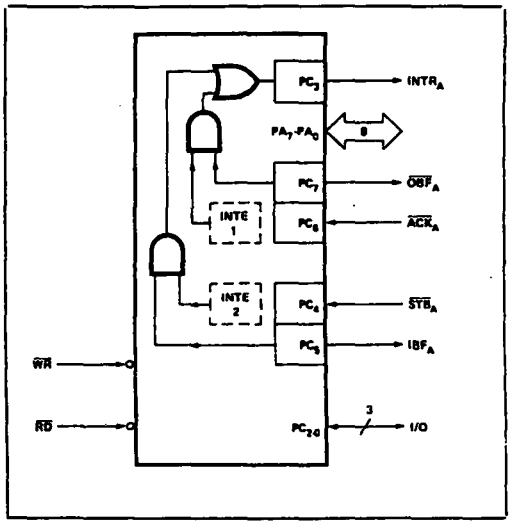


Figure 14. MODE 2

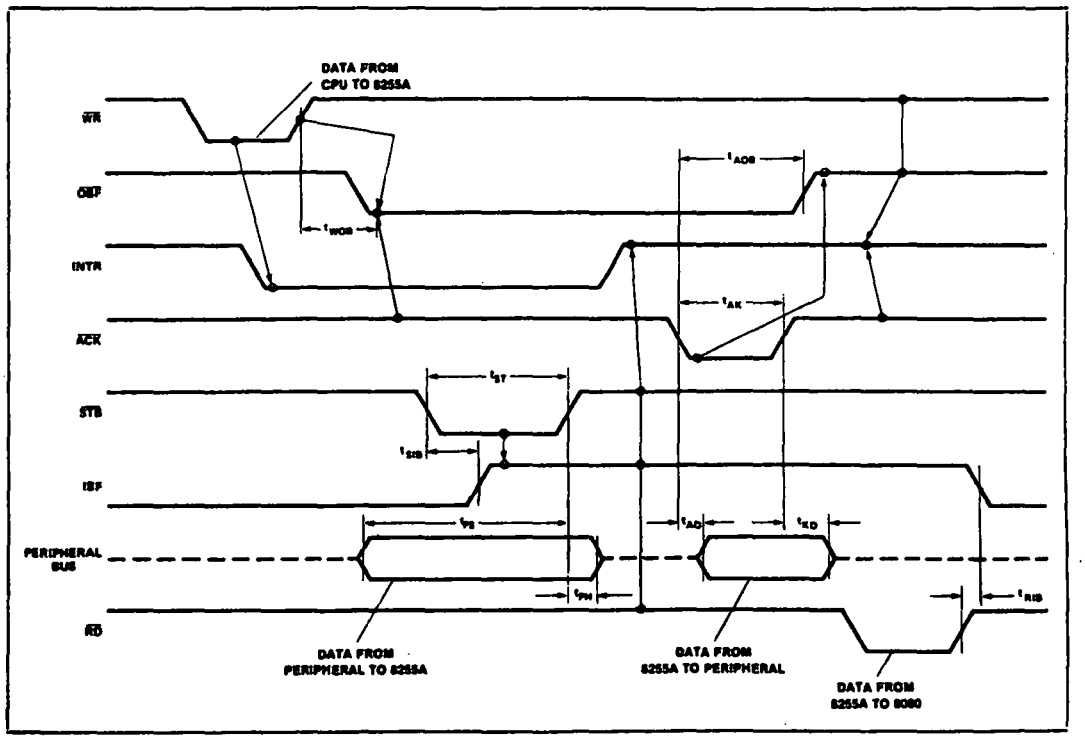


Figure 15. MODE 2 (Bidirectional)

NOTE: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
(INTR = IBF • $\overline{MASK}$ • $\overline{STB}$ • $\overline{RD}$ + $\overline{OBF}$ • $\overline{MASK}$ • $\overline{ACK}$ • $\overline{WR}$)

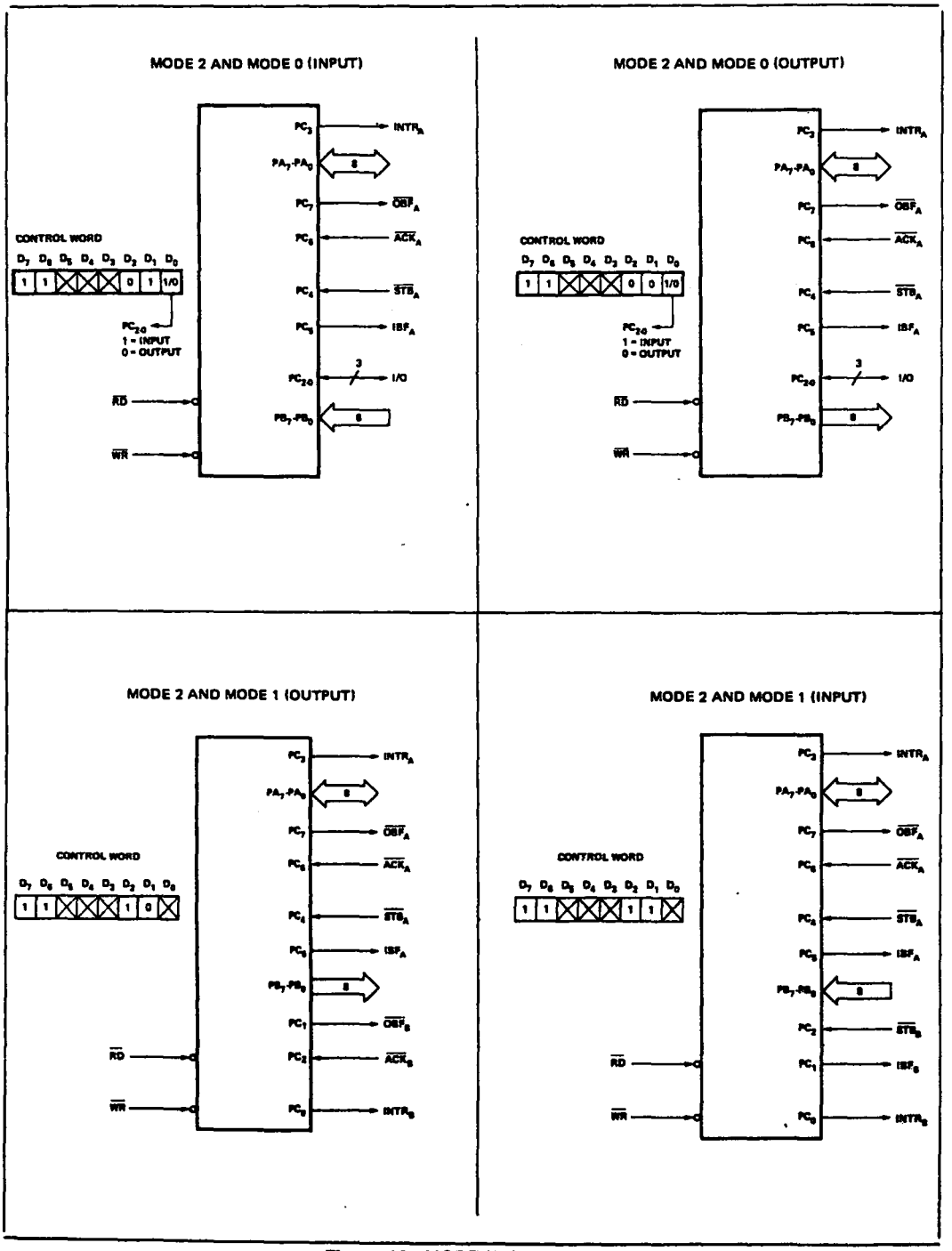


Figure 16. MODE ¼ Combinations

231308-001

Mode Definition Summary

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	
PA_0	IN	OUT	IN	OUT	⟷	
PA_1	IN	OUT	IN	OUT	⟷	
PA_2	IN	OUT	IN	OUT	⟷	
PA_3	IN	OUT	IN	OUT	⟷	
PA_4	IN	OUT	IN	OUT	⟷	
PA_5	IN	OUT	IN	OUT	⟷	
PA_6	IN	OUT	IN	OUT	⟷	
PA_7	IN	OUT	IN	OUT	⟷	
PB_0	IN	OUT	IN	OUT	—	MODE 0 OR MODE 1 ONLY
PB_1	IN	OUT	IN	OUT	—	
PB_2	IN	OUT	IN	OUT	—	
PB_3	IN	OUT	IN	OUT	—	
PB_4	IN	OUT	IN	OUT	—	
PB_5	IN	OUT	IN	OUT	—	
PB_6	IN	OUT	IN	OUT	—	
PB_7	IN	OUT	IN	OUT	—	
PC_0	IN	OUT	$INTR_B$	$INTR_B$	I/O	
PC_1	IN	OUT	IBF_B	$\overline{OBF}_B$	I/O	
PC_2	IN	OUT	$\overline{STB}_B$	$\overline{ACK}_B$	I/O	
PC_3	IN	OUT	$INTR_A$	$INTR_A$	$INTR_A$	
PC_4	IN	OUT	$\overline{STB}_A$	I/O	$\overline{STB}_A$	
PC_5	IN	OUT	IBF_A	I/O	IBF_A	
PC_6	IN	OUT	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$	
PC_7	IN	OUT	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$	

Special Mode Combination Considerations

There are several combinations of modes when not all of the bits in Port C are used for control or status. The remaining bits can be used as follows:

If Programmed as Inputs –
All input lines can be accessed during a normal Port C read.

If Programmed as Outputs –
Bits in C upper (PC_7-PC_4) must be individually accessed using the bit set/reset function.

Bits in C lower (PC_3-PC_0) can be accessed using the bit set/reset function or accessed as a threesome by writing into Port C.

Source Current Capability on Port B and Port C

Any set of eight output buffers, selected randomly from Ports B and C can source 1mA at 1.5 volts. This feature allows the 8255 to directly drive Darlington type drivers and high-voltage displays that require such source current.

Reading Port C Status

In Mode 0, Port C transfers data to or from the peripheral device. When the 8255 is programmed to function in Modes 1 or 2, Port C generates or accepts "hand-shaking" signals with the peripheral device. Reading the contents of Port C allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.

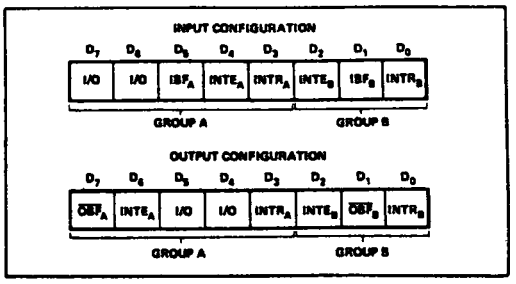


Figure 17. MODE 1 Status Word Format

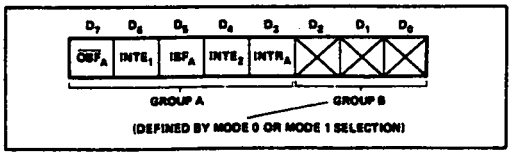


Figure 18. MODE 2 Status Word Format

APPLICATIONS OF THE 8255A

The 8255A is a very powerful tool for interfacing peripheral equipment to the microcomputer system. It represents the optimum use of available pins and is flexible enough to interface almost any I/O device without the need for additional external logic.

Each peripheral device in a microcomputer system usually has a "service routine" associated with it. The routine manages the software interface between the device and the CPU. The functional definition of the 8255A is programmed by the I/O service routine and becomes an extension of the system software. By examining the I/O devices interface characteristics for both data transfer and timing, and matching this information to the examples and tables in the detailed operational description, a control word can easily be developed to initialize the 8255A to exactly "fit" the application. Figures 19 through 25 present a few examples of typical applications of the 8255A.

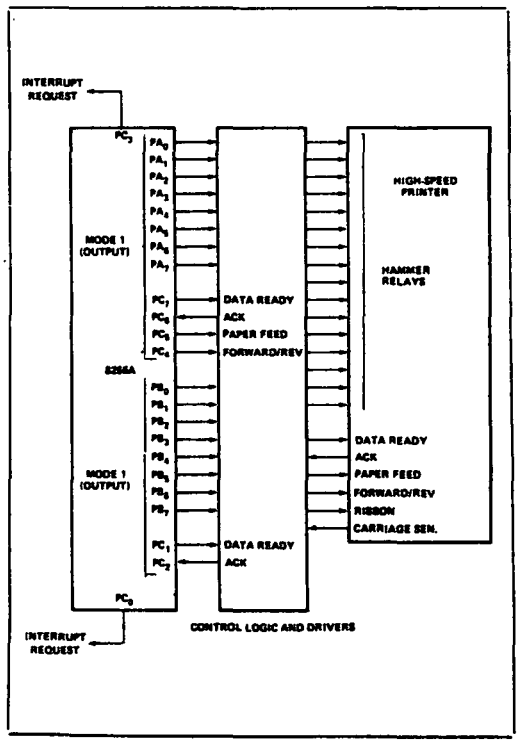


Figure 19. Printer Interface

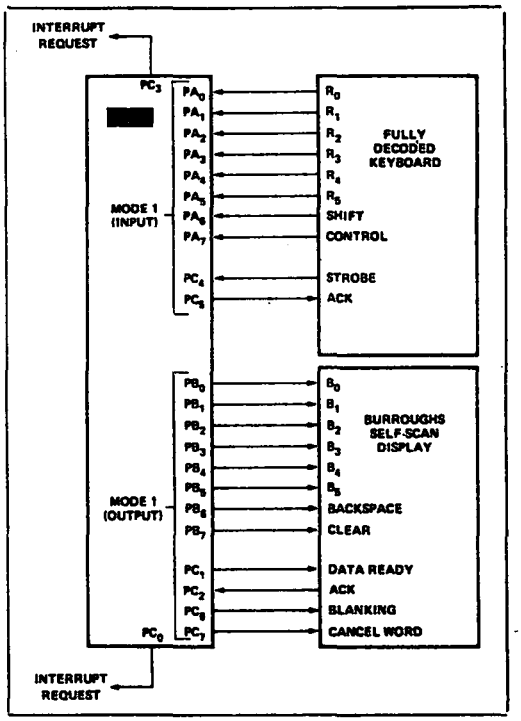


Figure 20. Keyboard and Display Interface

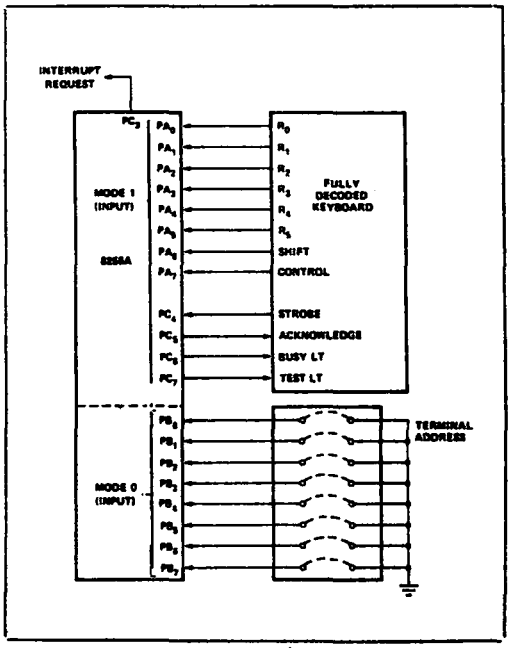


Figure 21. Keyboard and Terminal Address Interface

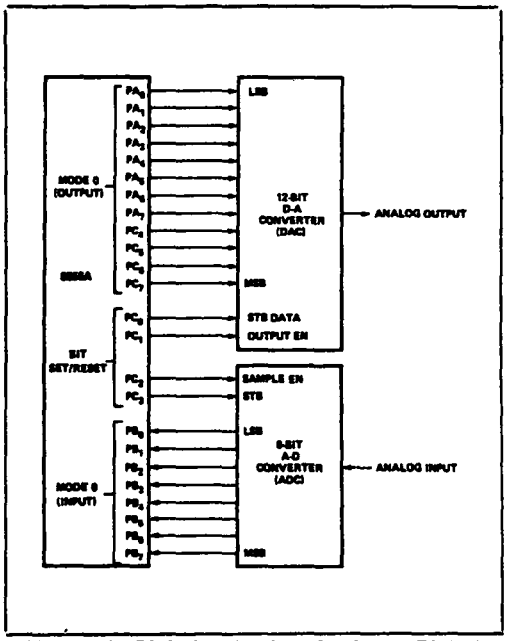


Figure 22. Digital to Analog, Analog to Digital

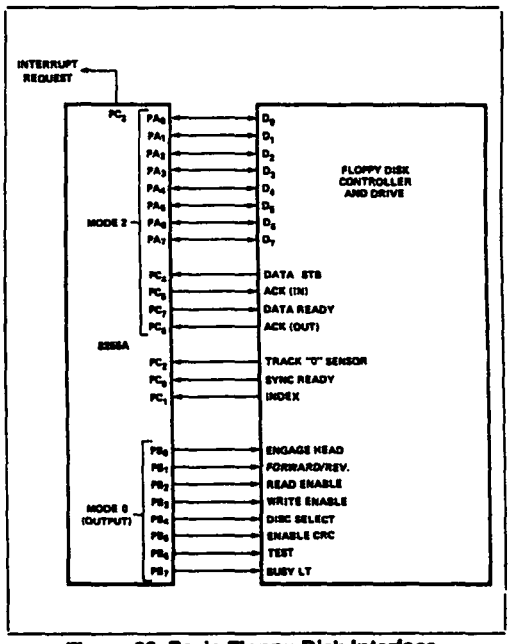


Figure 23. Basic Floppy Disk Interface

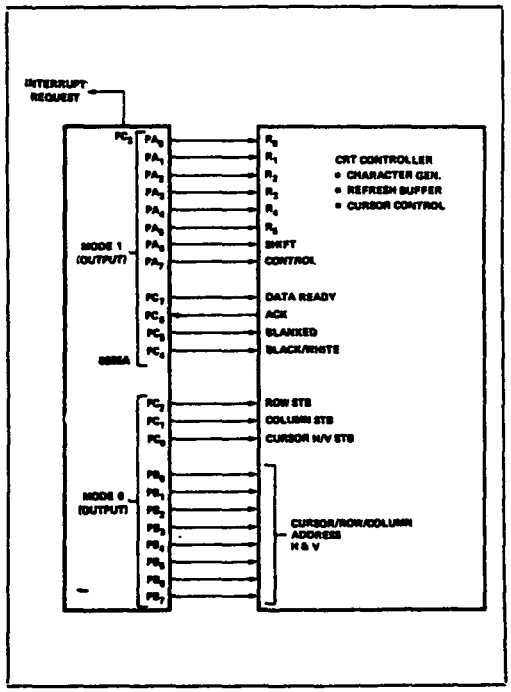


Figure 24. Basic CRT Controller Interface

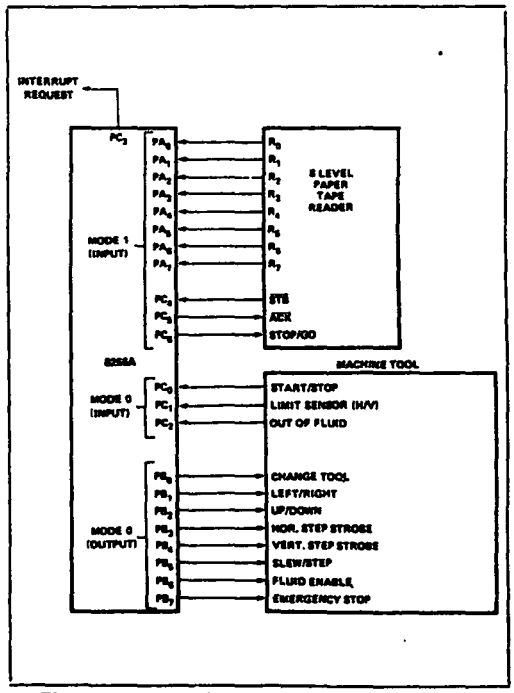


Figure 25. Machine Tool Controller Interface

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias. . . . . 0°C to 70°C
Storage Temperature . . . . . .–65°C to +150°C
Voltage on Any Pin
With Respect to Ground . . . . . –0.5V to +7V
Power Dissipation . . . . . 1 Watt

NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = +5V ±10%, GND = 0V) *

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	V	
V_{OL} (DB)	Output Low Voltage (Data Bus)		0.45*	V	I_{OL} = 2.5mA
V_{OL}(PER)	Output Low Voltage (Peripheral Port)		0.45*	V	I_{OL} = 1.7mA
V_{OH}(DB)	Output High Voltage (Data Bus)	2.4		V	I_{OH} = -400μA
V_{OH}(PER)	Output High Voltage (Peripheral Port)	2.4		V	I_{OH} = -200μA
I_{DAR}[1]	Darlington Drive Current	-1.0	-4.0	mA	R_{EXT} = 750Ω; V_{EXT} = 1.5V
I_{CC}	Power Supply Current		120	mA	
I_{IL}	Input Load Current		±10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} to .45V

NOTE:
1. Available on any 8 pins from Port B and C.

CAPACITANCE (T_A = 25°C, V_{CC} = GND = 0V)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance			10	pF	fc = 1MHz
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to GND

A.C. CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = +5V ±10%, GND = 0V) *

Bus Parameters

READ

Symbol	Parameter	8255A		8255A-5		Unit
		Min.	Max.	Min.	Max.	
t_{AR}	Address Stable Before READ	0		0		ns
t_{RA}	Address Stable After READ	0		0		ns
t_{RR}	READ Pulse Width	300		300		ns
t_{RD}	Data Valid From READ[1]		250		200	ns
t_{DF}	Data Float After READ	10	150	10	100	ns
t_{RV}	Time Between READs and/or WRITEs	850		850		ns

A.C. CHARACTERISTICS (Continued)

WRITE

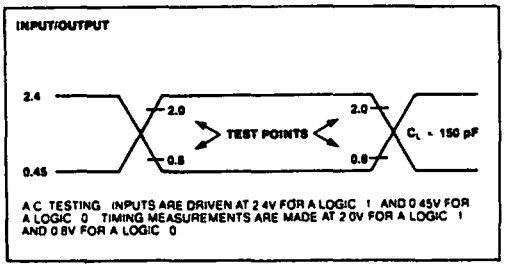
Symbol	Parameter	8255A		8255A-5		Unit
		Min.	Max.	Min.	Max.	
t_{AW}	Address Stable Before WRITE	0		0		ns
t_{WA}	Address Stable After WRITE	20		20		ns
t_{WW}	WRITE Pulse Width	400		300		ns
t_{DW}	Data Valid to WRITE (T.E.)	100		100		ns
t_{WD}	Data Valid After WRITE	30		30		ns

OTHER TIMINGS

Symbol	Parameter	8255A		8255A-5		Unit
		Min.	Max.	Min.	Max.	
t_{WB}	WR = 1 to Output[1]		350		350	ns
t_{IR}	Peripheral Data Before RD	0		0		ns
t_{HR}	Peripheral Data After RD	0		0		ns
t_{AK}	ACK Pulse Width	300		300		ns
t_{ST}	STB Pulse Width	500		500		ns
t_{PS}	Per. Data Before T.E. of STB	0		0		ns
t_{PH}	Per. Data After T.E. of STB	180		180		ns
t_{AD}	ACK = 0 to Output[1]		300		300	ns
t_{KD}	ACK = 1 to Output Float	20	250	20	250	ns
t_{WOB}	WR = 1 to OBF = 0[1]		650		650	ns
t_{AOB}	ACK = 0 to OBF = 1[1]		350		350	ns
t_{SIB}	STB = 0 to IBF = 1[1]		300		300	ns
t_{RIB}	RD = 1 to IBF = 0[1]		300		300	ns
t_{RIT}	RD = 0 to INTR = 0[1]		400		400	ns
t_{SIT}	STB = 1 to INTR = 1[1]		300		300	ns
t_{AIT}	ACK = 1 to INTR = 1[1]		350		350	ns
t_{WIT}	WR = 0 to INTR = 0[1,3]		450		450	ns

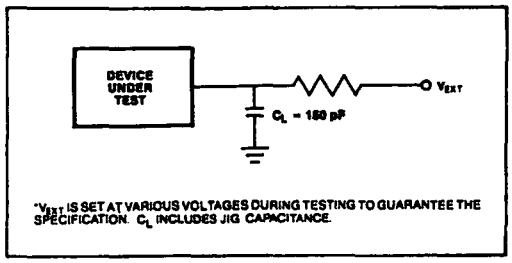
NOTES:
1. Test Conditions: C_L = 150 pF.
2. Period of Reset pulse must be at least 50μs during or after power on. Subsequent Reset pulse can be 500 ns min.
3. INTR↑ may occur as early as $\overline{WR}$↓.
* For Extended Temperature EXPRESS, use M8255A electrical parameters.

A.C. TESTING INPUT, OUTPUT WAVEFORM



A.C. TESTING INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC 1 AND 0.45V FOR A LOGIC 0. TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC 1 AND 0.8V FOR A LOGIC 0.

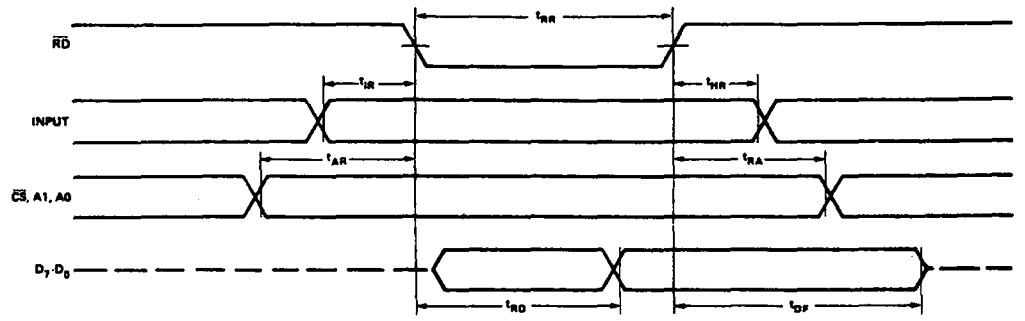
A.C. TESTING LOAD CIRCUIT



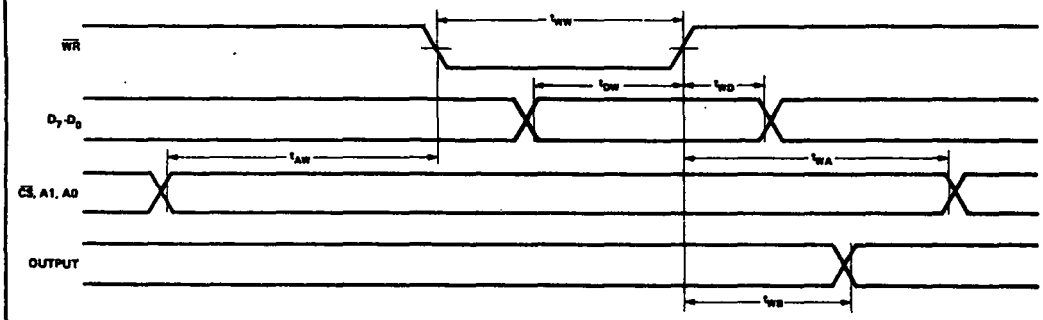
*V_{EXT} IS SET AT VARIOUS VOLTAGES DURING TESTING TO GUARANTEE THE SPECIFICATION. C_L INCLUDES JIG CAPACITANCE.

WAVEFORMS

MODE 0 (BASIC INPUT)

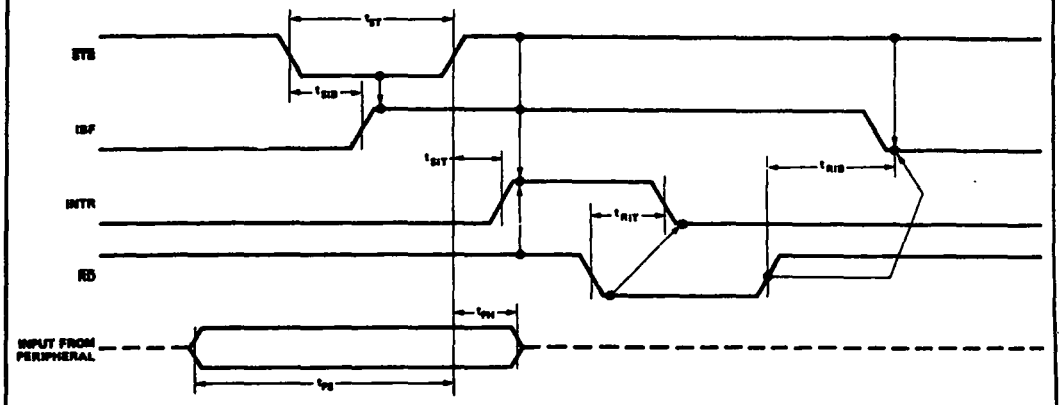


MODE 0 (BASIC OUTPUT)

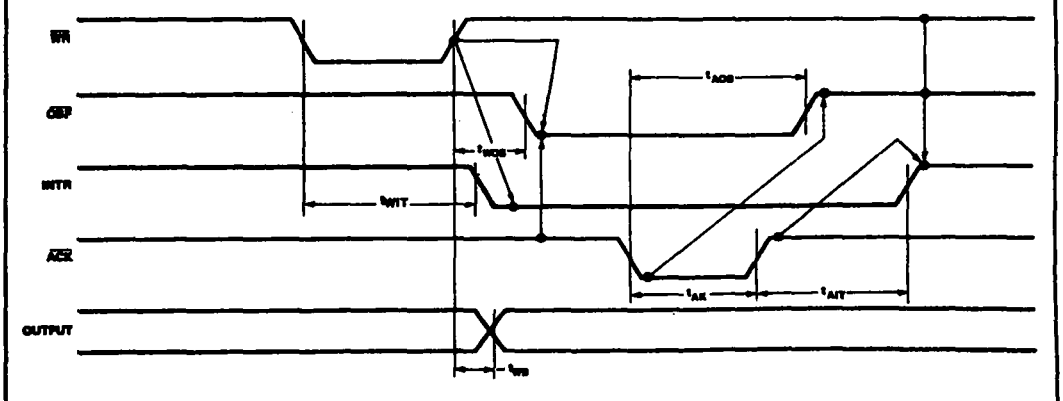


WAVEFORMS (Continued)

MODE 1 (STROBED INPUT)

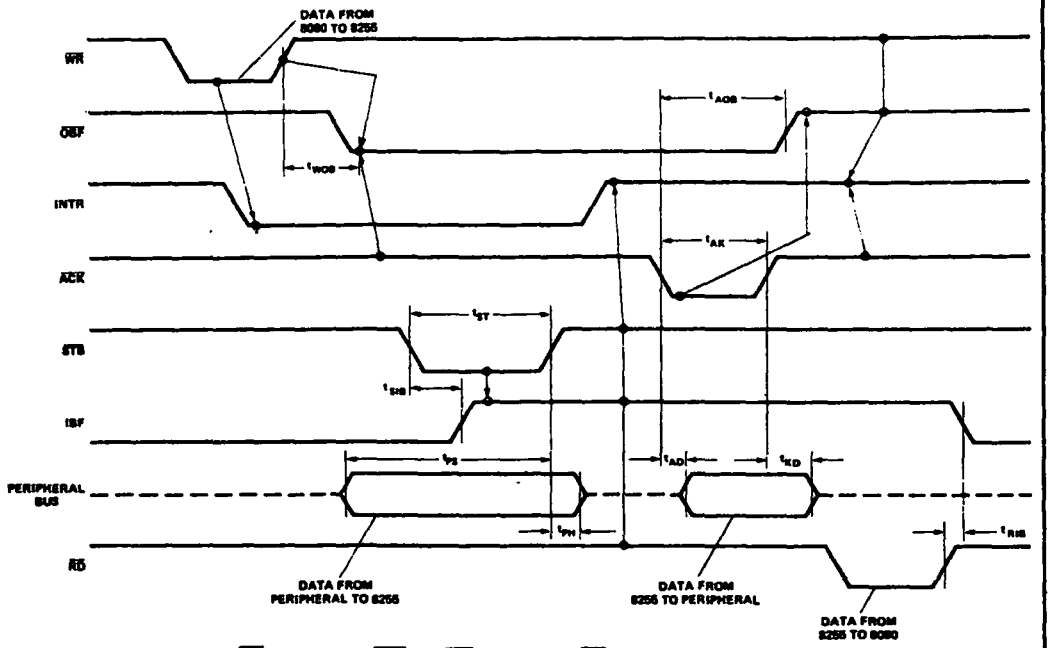


MODE 1 (STROBED OUTPUT)



WAVEFORMS (Continued)

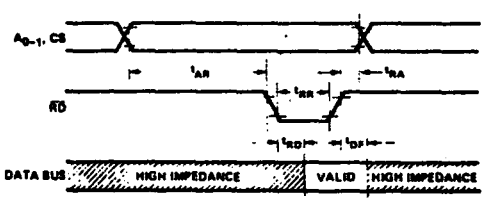
MODE 2 (BIDIRECTIONAL)



NOTE: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
(INTR = IBF • $\overline{MASK}$ • $\overline{STB}$ • $\overline{RD}$ + $\overline{OBF}$ • $\overline{MASK}$ • $\overline{ACK}$ • $\overline{WR}$)

WRITE TIMING

READ TIMING



National Semiconductor

DAC0808/DAC0807/DAC0806 8-Bit D/A Converters

General Description

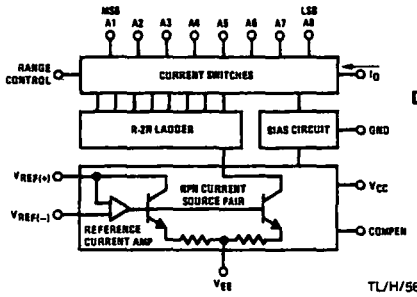
The DAC0808 series is an 8-bit monolithic digital-to-analog converter (DAC) featuring a full scale output current settling time of 150 ns while dissipating only 33 mW with ±5V supplies. No reference current (I_{REF}) trimming is required for most applications since the full scale output current is typically ±1 LSB of 255 I_{REF}/ 256. Relative accuracies of better than ±0.19% assure 8-bit monotonicity and linearity while zero level output current of less than 4 μA provides 8-bit zero accuracy for $I_{REF} \geq 2$ mA. The power supply currents of the DAC0808 series are independent of bit codes, and exhibits essentially constant device characteristics over the entire supply voltage range.

The DAC0808 will interface directly with popular TTL, DTL or CMOS logic levels, and is a direct replacement for the MC1508/MC1408. For higher speed applications, see DAC0800 data sheet.

Features

- Relative accuracy: ±0.19% error maximum (DAC0808)
- Full scale current match: ±1 LSB typ
- 7 and 6-bit accuracy available (DAC0807, DAC0806)
- Fast settling time: 150 ns typ
- Noninverting digital inputs are TTL and CMOS compatible
- High speed multiplying input slew rate: 8 mA/μs
- Power supply voltage range: ±4.5V to ±18V
- Low power consumption: 33 mW @ ±5V

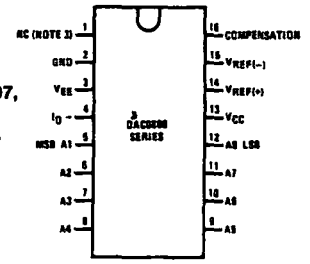
Block and Connection Diagrams



TL/H/5687-1

Order Number DAC0808, DAC0807, or DAC0806 See NS Package Number J16A, M16A or N16A

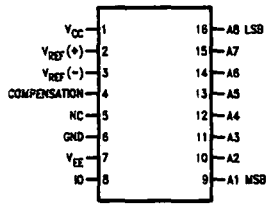
Dual-In-Line Package



TOP VIEW

TL/H/5687-2

Small-Outline Package



TL/H/5687-13

Top View

Ordering Information

ACCURACY	OPERATING TEMPERATURE RANGE	ORDER NUMBERS				
		J PACKAGE (J16A)*		N PACKAGE (N16A)*		SO PACKAGE (M16A)
8-bit	$-55°C \leq T_A \leq +125°C$	DAC0808LJ	MC1508L8			
8-bit	$0°C \leq T_A \leq +75°C$	DAC0808LCJ	MC1408L8	DAC0808LCN	MC1408P8	DAC0808LCM
7-bit	$0°C \leq T_A \leq +75°C$	DAC0807LCJ	MC1408L7	DAC0807LCN	MC1408P7	DAC0807LCM
6-bit	$0°C \leq T_A \leq +75°C$	DAC0806LCJ	MC1408L6	DAC0806LCN	MC1408P6	DAC0806LCM

*Note. Devices may be ordered by using either order number.

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

Power Supply Voltage	
V_{CC}	+18 V_{DC}
V_{EE}	−18 V_{DC}
Digital Input Voltage, V5–V12	−10 V_{DC} to +18 V_{DC}
Applied Output Voltage, V_O	−11 V_{DC} to +18 V_{DC}
Reference Current, I_{14}	5 mA
Reference Amplifier Inputs, V14, V15	V_{CC}, V_{EE}
Power Dissipation (Note 3)	1000 mW
ESD Susceptibility (Note 4)	TBD
Storage Temperature Range	−65°C to +150°C
Lead Temp. (Soldering, 10 seconds)	
Dual-In-Line Package (Plastic)	260°C
Dual-In-Line Package (Ceramic)	300°C
Surface Mount Package	
Vapor Phase (60 seconds)	215°C
Infrared (15 seconds)	220°C

Operating Ratings

Temperature Range	$T_{MIN} \le T_A \le T_{MAX}$
DAC0808L	$-55°C \le T_A \le +125°C$
DAC0808LC Series	$0 \le T_A \le +75°C$

Electrical Characteristics

(V_{CC} = 5V, V_{EE} = −15 V_{DC}, $V_{REF}/R14$ = 2 mA, DAC0808: T_A = −55°C to +125°C, DAC0808C, DAC0807C, DAC0806C, T_A = 0°C to +75°C, and all digital inputs at high logic level unless otherwise noted.)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
E_r	Relative Accuracy (Error Relative to Full Scale I_O)	*(Figure 4)*				%
	DAC0808L (LM1508-8), DAC0808LC (LM1408-8)				±0.19	%
	DAC0807LC (LM1408-7), (Note 5)				±0.39	%
	DAC0806LC (LM1408-6), (Note 5)				±0.78	%
	Settling Time to Within ½ LSB (Includes t_{PLH})	T_A = 25°C (Note 6), *(Figure 5)*		150		ns
t_{PLH}, t_{PHL}	Propagation Delay Time	T_A = 25°C, *(Figure 5)*		30	100	ns
TCI_O	Output Full Scale Current Drift			±20		ppm/°C
MSB	Digital Input Logic Levels	*(Figure 3)*				
V_{IH}	High Level, Logic "1"		2			V_{DC}
V_{IL}	Low Level, Logic "0"				0.8	V_{DC}
MSB	Digital Input Current	*(Figure 3)*				
	High Level	V_{IH} = 5V		0	0.040	mA
	Low Level	V_{IL} = 0.8V		−0.003	−0.8	mA
I_{15}	Reference Input Bias Current	*(Figure 3)*		−1	−3	μA
	Output Current Range	*(Figure 3)*				
		V_{EE} = −5V	0	2.0	2.1	mA
		V_{EE} = −15V, T_A = 25°C	0	2.0	4.2	mA
I_O	Output Current	V_{REF} = 2.000V, R14 = 1000Ω,				
		(Figure 3)	1.9	1.99	2.1	mA
	Output Current, All Bits Low	*(Figure 3)*		0	4	μA
	Output Voltage Compliance (Note 2)	$E_r \le 0.19\%$, T_A = 25°C				
	V_{EE} = −5V, I_{REF} = 1 mA				−0.55, +0.4	V_{DC}
	V_{EE} Below −10V				−5.0, +0.4	V_{DC}

Electrical Characteristics (Continued)

(V_{CC} = 5V, V_{EE} = −15 V_{DC}, V_{REF}/R14 = 2 mA, DAC0808: T_A = −55°C to +125°C, DAC0808C, DAC0807C, DAC0806C, T_A = 0°C to +75°C, and all digital inputs at high logic level unless otherwise noted.)

Symbol	Parameter	Conditions	Min	Typ	Max	Units
SRI_{REF}	Reference Current Slew Rate	*(Figure 6)*	4	8		mA/μs
	Output Current Power Supply Sensitivity	−5V ≤ V_{EE} ≤ −16.5V		0.05	2.7	μA/V
	Power Supply Current (All Bits Low)	*(Figure 3)*				
I_{CC}				2.3	22	mA
I_{EE}				−4.3	−13	mA
	Power Supply Voltage Range	T_A = 25°C, *(Figure 3)*				
V_{CC}			4.5	5.0	5.5	V_{DC}
V_{EE}			−4.5	−15	−16.5	V_{DC}
	Power Dissipation					
	All Bits Low	V_{CC} = 5V, V_{EE} = −5V		33	170	mW
		V_{CC} = 5V, V_{EE} = −15V		106	305	mW
	All Bits High	V_{CC} = 15V, V_{EE} = −5V		90		mW
		V_{CC} = 15V, V_{EE} = −15V		160		mW

Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: Range control is not required.

Note 3: The maximum power dissipation must be derated at elevated temperatures and is dictated by T_{JMAX}, θ_{JA}, and the ambient temperature, T_A. The maximum allowable power dissipation at any temperature is $P_D = (T_{JMAX} - T_A)/\theta_{JA}$ or the number given in the Absolute Maximum Ratings, whichever is lower. For this device, T_{JMAX} = 125°C, and the typical junction-to-ambient thermal resistance of the dual-in-line J package when the board mounted is 100°C/W. For the dual-in-line N package, this number increases to 175°C/W and for the small outline M package this number is 100°C/W.

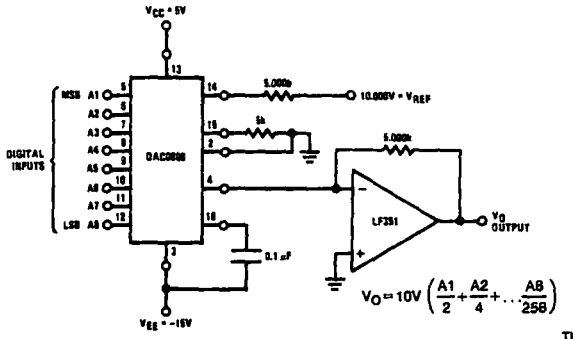
Note 4: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

Note 5: All current switches are tested to guarantee at least 50% of rated current.

Note 6: All bits switched.

Note 7: Pin-out numbers for the DAL080X represent the dual-in-line package. The small outline package pinout differs from the dual-in-line package.

Typical Application



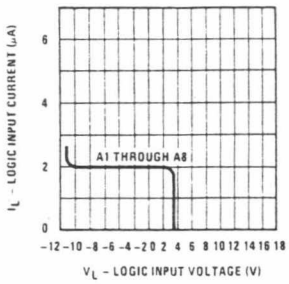
TL/H/5687–3

FIGURE 1. + 10V Output Digital to Analog Converter (Note 7)

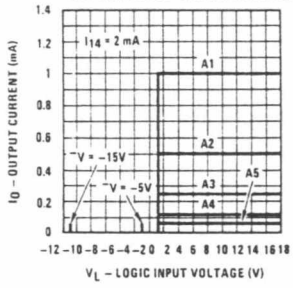
Typical Performance Characteristics

V_{CC} = 5V, V_{EE} = −15V, T_A = 25°C, unless otherwise noted

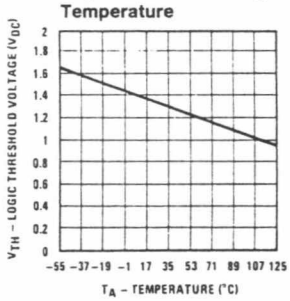
Logic Input Current vs Input Voltage



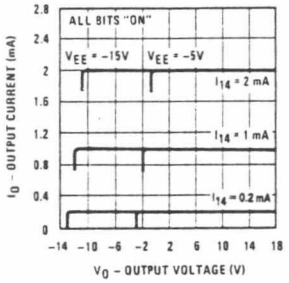
Bit Transfer Characteristics



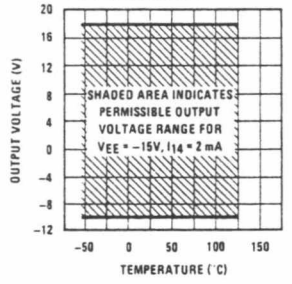
Logic Threshold Voltage vs Temperature



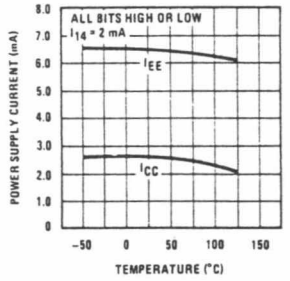
Output Current vs Output Voltage (Output Voltage Compliance)



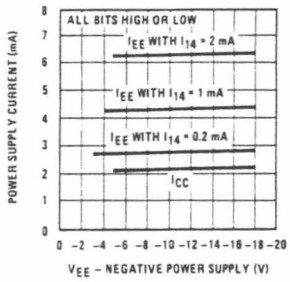
Output Voltage Compliance vs Temperature



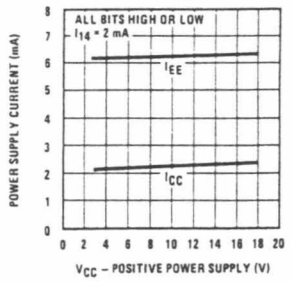
Typical Power Supply Current vs Temperature



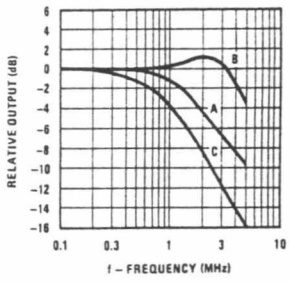
Typical Power Supply Current vs V_{EE}



Typical Power Supply Current vs V_{CC}



Reference Input Frequency Response



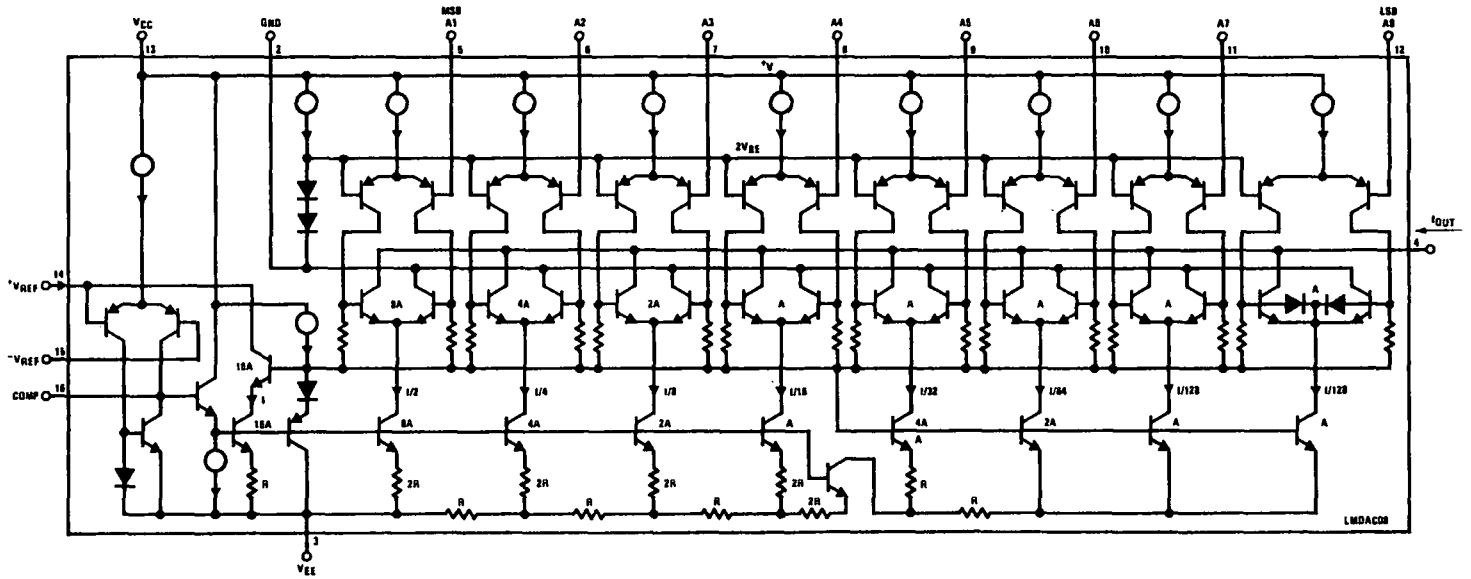
TL/H/5687–5

Unless otherwise specified: R14 = R15 = 1 kΩ, C = 15 pF, pin 16 to V_{EE}; R_L = 50Ω, pin 4 to ground.

Curve A: Large Signal Bandwidth Method of *Figure 7*, V_{REF} = 2 Vp-p offset 1 V above ground.

Curve B: Small Signal Bandwidth Method of *Figure 7*, R_L = 250Ω, V_{REF} = 50 mVp-p offset 200 mV above ground.

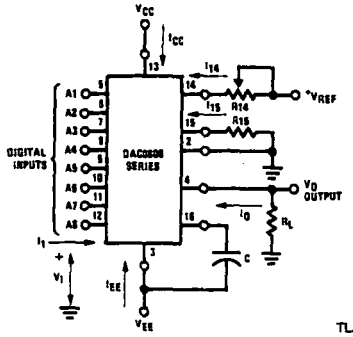
Curve C: Large and Small Signal Bandwidth Method of *Figure 9* (no op amp, R_L = 50Ω), R_S = 50Ω, V_{REF} = 2V, V_S = 100 mVp-p centered at 0V.



TL/H/5687–4

FIGURE 2. Equivalent Circuit of the DAC0808 Series (Note 7)

Test Circuits



V_I and I_1 apply to inputs A1–A8.

The resistor tied to pin 15 is to temperature compensate the bias current and may not be necessary for all applications.

$$I_O = K\left(\frac{A1}{2} + \frac{A2}{4} + \frac{A4}{16} + \frac{A5}{32} + \frac{A6}{64} + \frac{A7}{128} + \frac{A8}{256}\right)$$

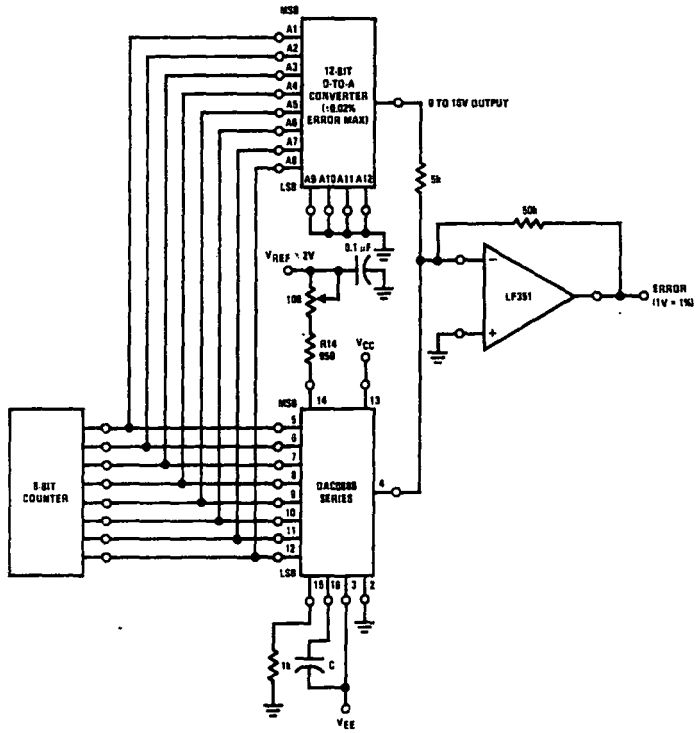
where $K \cong \frac{V_{REF}}{R14}$

and A_N = "1" if A_N is at high level

A_N = "0" if A_N is at low level

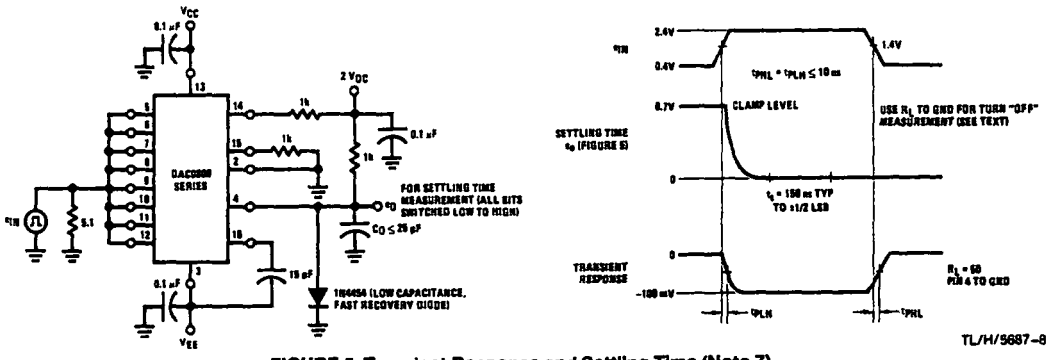
TL/H/5687–6

FIGURE 3. Notation Definitions Test Circuit (Note 7)



TL/H/5687–7

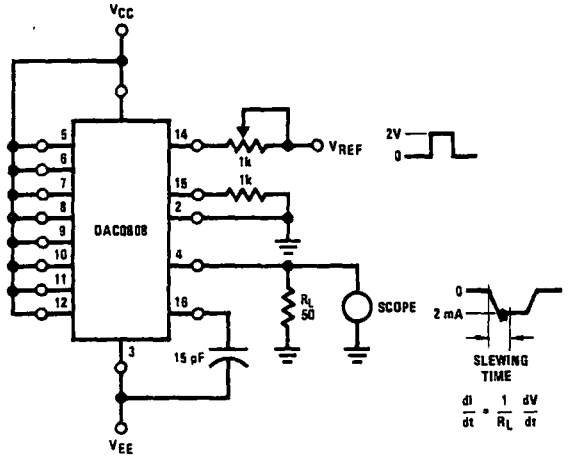
FIGURE 4. Relative Accuracy Test Circuit (Note 7)



TL/H/5687–8

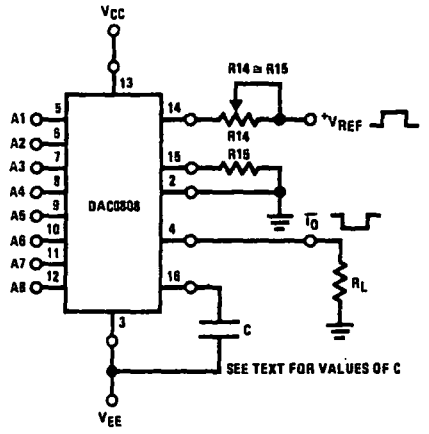
FIGURE 5. Transient Response and Settling Time (Note 7)

Test Circuits (Continued)



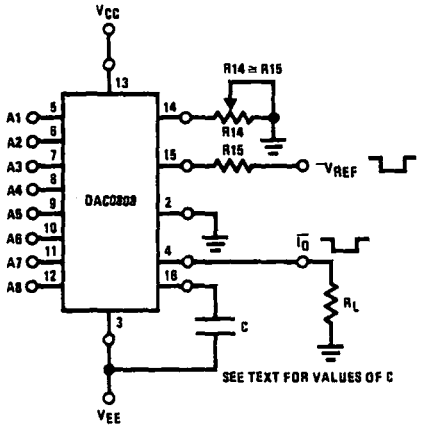
TL/H/5687-9

FIGURE 6. Reference Current Slew Rate Measurement (Note 7)



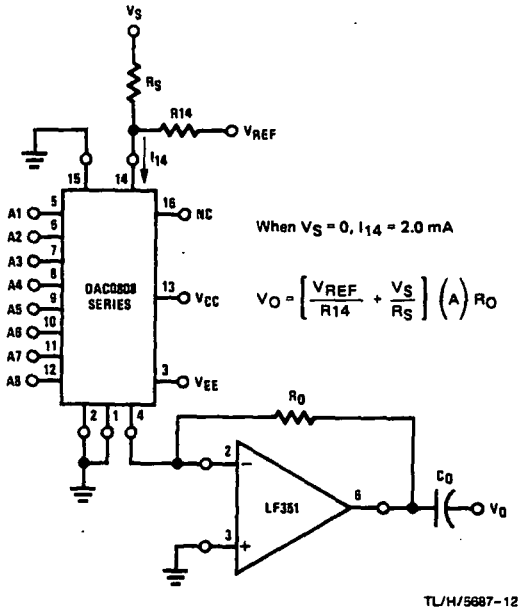
TL/H/5687-10

FIGURE 7. Positive V_{REF} (Note 7)



TL/H/5687-11

FIGURE 8. Negative V_{REF} (Note 7)



TL/H/5687-12

FIGURE 9. Programmable Gain Amplifier or Digital Attenuator Circuit (Note 7)

Application Hints

REFERENCE AMPLIFIER DRIVE AND COMPENSATION

The reference amplifier provides a voltage at pin 14 for converting the reference voltage to a current, and a turn-around circuit or current mirror for feeding the ladder. The reference amplifier input currrent, I_{14}, must always flow into pin 14, regardless of the set-up method or reference voltage polarity.

Connections for a positive voltage are shown in *Figure 7*. The reference voltage source supplies the full current I_{14}. For bipolar reference signals, as in the multiplying mode, R15 can be tied to a negative voltage corresponding to the minimum input level. It is possible to eliminate R15 with only a small sacrifice in accuracy and temperature drift.

The compensation capacitor value must be increased with increases in R14 to maintain proper phase margin; for R14 values of 1, 2.5 and 5 kΩ, minimum capacitor values are 15, 37 and 75 pF. The capacitor may be tied to either V_{EE} or ground, but using V_{EE} increases negative supply rejection.

Application Hints (Continued)

A negative reference voltage may be used if R14 is grounded and the reference voltage is applied to R15 as shown in *Figure 8*. A high input impedance is the main advantage of this method. Compensation involves a capacitor to V_{EE} on pin 16, using the values of the previous paragraph. The negative reference voltage must be at least 4V above the V_{EE} supply. Bipolar input signals may be handled by connecting R14 to a positive reference voltage equal to the peak positive input level at pin 15.

When a DC reference voltage is used, capacitive bypass to ground is recommended. The 5V logic supply is not recommended as a reference voltage. If a well regulated 5V supply which drives logic is to be used as the reference, R14 should be decoupled by connecting it to 5V through another resistor and bypassing the junction of the 2 resistors with 0.1 μF to ground. For reference voltages greater than 5V, a clamp diode is recommended between pin 14 and ground.

If pin 14 is driven by a high impedance such as a transistor current source, none of the above compensation methods apply and the amplifier must be heavily compensated, decreasing the overall bandwidth.

OUTPUT VOLTAGE RANGE

The voltage on pin 4 is restricted to a range of −0.55 to 0.4V when $V_{EE} = -5V$ due to the current switching methods employed in the DAC0808.

The negative output voltage compliance of the DAC0808 is extended to −5V where the negative supply voltage is more negative than −10V. Using a full-scale current of 1.992 mA and load resistor of 2.5 kΩ between pin 4 and ground will yield a voltage output of 256 levels between 0 and −4.980V. Floating pin 1 does not affect the converter speed or power dissipation. However, the value of the load resistor determines the switching time due to increased voltage swing. Values of R_L up to 500Ω do not significantly affect performance, but a 2.5 kΩ load increases worst-case settling time to 1.2 μs (when all bits are switched ON). Refer to the subsequent text section on Settling Time for more details on output loading.

OUTPUT CURRENT RANGE

The output current maximum rating of 4.2 mA may be used only for negative supply voltages more negative than −8V, due to the increased voltage drop across the resistors in the reference current amplifier.

ACCURACY

Absolute accuracy is the measure of each output current level with respect to its intended value, and is dependent upon relative accuracy and full-scale current drift. Relative accuracy is the measure of each output current level as a fraction of the full-scale current. The relative accuracy of the DAC0808 is essentially constant with temperature due to the excellent temperature tracking of the monolithic resistor ladder. The reference current may drift with temperature, causing a change in the absolute accuracy of output current. However, the DAC0808 has a very low full-scale current drift with temperature.

The DAC0808 series is guaranteed accurate to within ±½ LSB at a full-scale output current of 1.992 mA. This corresponds to a reference amplifier output current drive to the ladder network of 2 mA, with the loss of 1 LSB (8 μA) which is the ladder remainder shunted to ground. The input current to pin 14 has a guaranteed value of between 1.9 and 2.1 mA, allowing some mismatch in the NPN current source pair. The accuracy test circuit is shown in *Figure 4*. The 12-bit converter is calibrated for a full-scale output current of 1.992 mA. This is an optional step since the DAC0808 accuracy is essentially the same between 1.5 and 2.5 mA. Then the DAC0808 circuits' full-scale current is trimmed to the same value with R14 so that a zero value appears at the error amplifier output. The counter is activated and the error band may be displayed on an oscilloscope, detected by comparators, or stored in a peak detector.

Two 8-bit D-to-A converters may not be used to construct a 16-bit accuracy D-to-A converter. 16-bit accuracy implies a total error of ±½ of one part in 65,536 or ±0.00076%, which is much more accurate than the ±0.019% specification provided by the DAC0808.

MULTIPLYING ACCURACY

The DAC0808 may be used in the multiplying mode with 8-bit accuracy when the reference current is varied over a range of 256:1. If the reference current in the multiplying mode ranges from 16 μA to 4 mA, the additional error contributions are less than 1.6 μA. This is well within 8-bit accuracy when referred to full-scale.

A monotonic converter is one which supplies an increase in current for each increment in the binary word. Typically, the DAC0808 is monotonic for all values of reference current above 0.5 mA. The recommended range for operation with a DC reference current is 0.5 to 4 mA.

SETTLING TIME

The worst-case switching condition occurs when all bits are switched ON, which corresponds to a low-to-high transition for all bits. This time is typically 150 ns for settling to within ±½ LSB, for 8-bit accuracy, and 100 ns to ½ LSB for 7 and 6-bit accuracy. The turn OFF is typically under 100 ns. These times apply when $R_L \leq 500\Omega$ and $C_O \leq 25$ pF.

Extra care must be taken in board layout since this is usually the dominant factor in satisfactory test results when measuring settling time. Short leads, 100 μF supply bypassing for low frequencies, and minimum scope lead length are all mandatory.

2

LM158, LM258, LM358, LM2904

DUAL DIFFERENTIAL INPUT OPERATIONAL AMPLIFIERS

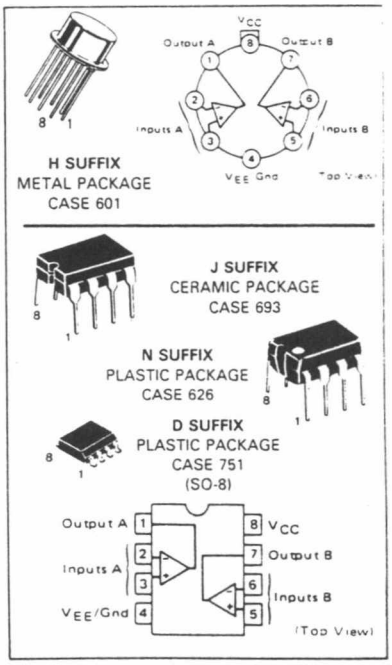
SILICON MONOLITHIC INTEGRATED CIRCUIT

DUAL LOW POWER OPERATIONAL AMPLIFIERS

Utilizing the circuit designs perfected for recently introduced Quad Operational Amplifiers, these dual operational amplifiers feature 1) low power drain, 2) a common mode input voltage range extending to ground/V_{EE}, 3) Single Supply or Split Supply operation and 4) pin outs compatible with the popular MC1558 dual operational amplifier. The LM158 Series is equivalent to one-half of an LM124.

These amplifiers have several distinct advantages over standard operational amplifier types in single supply applications. They can operate at supply voltages as low as 3.0 Volts or as high as 32 Volts with quiescent currents about one-fifth of those associated with the MC1741 (on a per amplifier basis). The common mode input range includes the negative supply, thereby eliminating the necessity for external biasing components in many applications. The output voltage range also includes the negative power supply voltage.

- Short Circuit Protected Outputs
- True Differential Input Stage
- Single Supply Operation: 3.0 to 32 Volts
- Low Input Bias Currents
- Internally Compensated
- Common Mode Range Extends to Negative Supply
- Single and Split Supply Operation
- Similar Performance to the Popular MC1558



MAXIMUM RATINGS (T_A = +25°C unless otherwise noted)

Rating	Symbol	LM158 LM258 LM358	LM2904	Unit
Power Supply Voltages Single Supply Split Supplies	 V_{CC} V_{CC}, V_{EE}	 32 ±16	 26 ±13	Vdc
Input Differential Voltage Range (1)	V_{IDR}	±32	±26	Vdc
Input Common Mode Voltage Range (2)	V_{ICR}	-0.3 to 32	-0.3 to 26	Vdc
Input Forward Current (3) (V_I < -0.3 V)	I_{IF}	50	–	mA
Output Short Circuit Duration	t_S	Continuous		
Junction Temperature Ceramic and Metal Packages Plastic Package	T_J	 175 150		°C
Storage Temperature Range Ceramic and Metal Packages Plastic Package	T_{stg}	 -65 to +150 -55 to +125		°C
Operating Ambient Temperature Range LM158 LM258 LM358 LM2904	T_A	 -55 to +125 -25 to +85 0 to +70 –	 – – – -40 to +105	°C

NOTES:
(1) Split Power Supplies.
(2) For Supply Voltages less than 32 V for the LM158/258/358 and 26 V for the LM2904, the absolute maximum input voltage is equal to the supply voltage.
(3) This input current will only exist when the voltage is negative at any of the input leads. Normal output states will reestablish when the input voltage returns to a voltage greater than -0.3 V.

ORDERING INFORMATION

Device	Temperature Range	Package
LM158H	-55 to +125°C	Metal Can
LM158J		Ceramic DIP
LM2904D	-40 to +105°C	SO-8
LM2904N		Plastic DIP
LM2904J	-40 to +85°C	Ceramic DIP
LM2904H		Metal Can
LM258D	-25 to +85°C	SO-8
LM258H		Metal Can
LM258J		Ceramic DIP
LM258N		Plastic DIP
LM358D	0 to +70°C	SO-8
LM358H		Metal Can
LM358J		Ceramic DIP
LM358N		Plastic DIP

ELECTRICAL CHARACTERISTICS (V_{CC} = 5.0 V, V_{EE} = Gnd, T_A = 25°C unless otherwise noted)

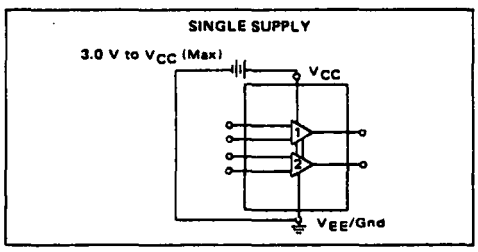
Characteristic	Symbol	LM158/LM258			LM358			LM2904			Unit
		Min	Typ	Max	Min	Typ	Max	Min	Typ	Max	
Input Offset Voltage	V_{IO}										mV
V_{CC} = 5.0 V to 30 V (26 V for LM2904), V_{IC} = 0 V to V_{CC} − 1.7 V, $V_O \simeq$ 1.4 V, R_S = 0 Ω											
T_A = 25°C		–	2.0	5.0	–	2.0	7.0	–	2.0	7.0	
T_A = T_{high} to T_{low} (Note 1)		–	–	7.0	–	–	9.0	–	–	10	
Average Temperature Coefficient of Input Offset Voltage T_A = T_{high} to T_{low} (Note 1)	$\Delta V_{IO}/\Delta T$	–	7.0	–	–	7.0	–	–	7.0	–	μV/°C
Input Offset Current	I_{IO}	–	3.0	30	–	5.0	50	–	5.0	50	nA
T_A = T_{high} to T_{low} (Note 1)		–	–	100	–	–	150	–	45	200	
Average Temperature Coefficient of Input Offset Current T_A = T_{high} to T_{low} (Note 1)	$\Delta I_{IO}/\Delta T$	–	10	–	–	10	–	–	10	–	pA/°C
Input Bias Current	I_{IB}	–	-45	-150	–	-45	-250	–	-45	-250	nA
T_A = T_{high} to T_{low} (Note 1)		–	-50	-300	–	-50	-500	–	-50	-500	
Input Common-Mode Voltage Range (Note 2)	V_{ICR}										V
V_{CC} = 30 V (26 V for LM2904)		0	–	28.3	0	–	28.3	0	–	24.3	
V_{CC} = 30 V (26 V for LM2904), T_A = T_{high} to T_{low}		0	–	28	0	–	28	0	–	24	
Differential Input Voltage Range	V_{IDR}	–	–	V_{CC}	–	–	V_{CC}	–	–	V_{CC}	V
Large Signal Open-Loop Voltage Gain	A_{VOL}										V/mV
R_L = 2.0 kΩ, V_{CC} = 15 V, For Large V_O Swing,		50	100	–	25	100	–	–	100	–	
T_A = T_{high} to T_{low} (Note 1)		25	–	–	15	–	–	–	–	–	
Channel Separation 1.0 kHz ≤ f ≤ 20 kHz, Input Referenced	–	–	-120	–	–	-120	–	–	-120	–	dB
Common-Mode Rejection Ratio R_S ≤ 10 kΩ	CMRR	70	85	–	65	70	–	50	70	–	dB
Power Supply Rejection Ratio	PSRR	65	100	–	65	100	–	50	100	–	dB
Output Voltage Range R_L = 2 kΩ (R_L ≥ 10 kΩ for LM2904)	V_{OR}	0	–	3.3	0	–	3.3	0	–	3.3	V
Output Voltage–High Limit (T_A = T_{high} to T_{low})(Note 1)	V_{OH}										V
V_{CC} = 30 V (26 V for LM2904), R_L = 2 kΩ		26	–	–	26	–	–	22	–	–	
V_{CC} = 30 V (26 V for LM2904), R_L = 10 kΩ		27	28	–	27	28	–	23	24	–	
Output Voltage–Low Limit V_{CC} = 5.0 V, R_L = 10 kΩ, T_A = T_{high} to T_{low} (Note 1)	V_{OL}	–	5.0	20	–	5.0	20	–	5.0	20	mV
Output Source Current V_{ID} = +1.0 V, V_{CC} = 15 V	I_{O+}	20	40	–	20	40	–	20	40	–	mA
Output Sink Current	I_{O-}										
V_{ID} = -1.0 V, V_{CC} = 15 V		10	20	–	10	20	–	10	20	–	mA
V_{ID} = -1.0 V, V_O = 200 mV		12	50	–	12	50	–	–	–	–	μA
Output Short Circuit to Ground (Note 3)	I_{os}	–	40	60	–	40	60	–	40	60	mA
Power Supply Current (T_A = T_{high} to T_{low})(Note 1)	I_{CC}										mA
V_{CC} = 30 V (26 V for LM2904), V_O = 0 V, R_L = ∞		–	1.5	3.0	–	1.5	3.0	–	1.5	3.0	
V_{CC} = 5 V, V_O = 0 V, R_L = ∞		–	0.7	1.2	–	0.7	1.2	–	0.7	1.2	

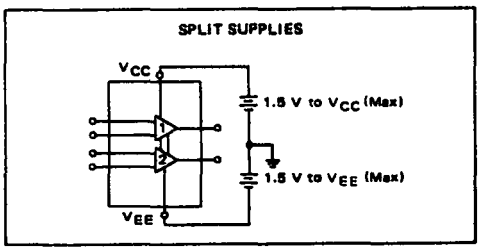
NOTES:

(1) T_{low} = −55°C for LM158 = −40°C for LM2904 = −25°C for LM258 = 0°C for LM358; T_{high} = +125°C for LM158 = +105°C for LM2904 = +85°C for LM258 = +70°C for LM358

(2) The input common-mode voltage or either input signal voltage should not be allowed to go negative by more than 0.3 V. The upper end of the common-mode voltage range is V_{CC} − 1.7 V.

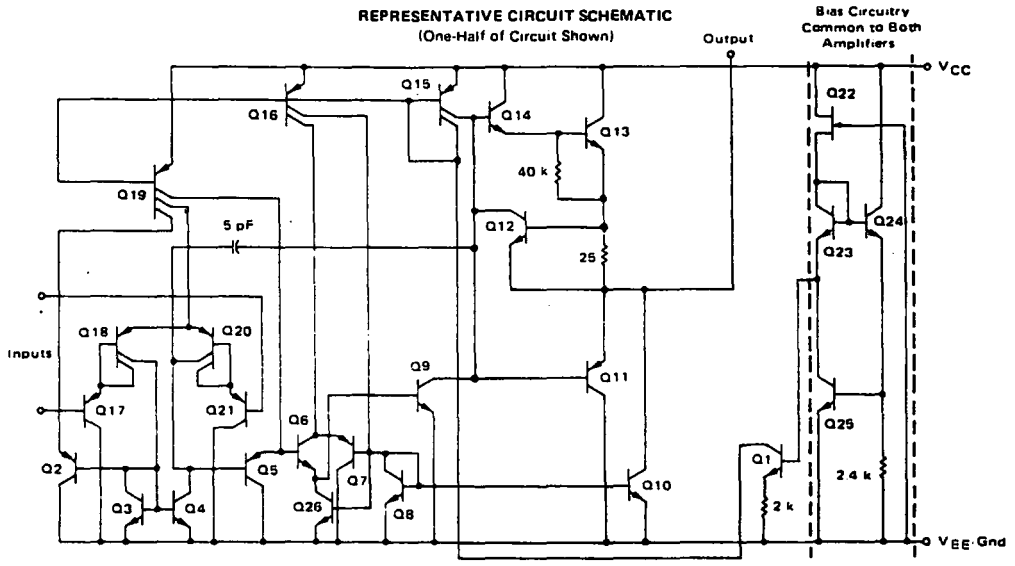
(3) Short circuits from the output to V_{CC} can cause excessive heating and eventual destruction. Destructive dissipation can result from simultaneous shorts on all amplifiers.



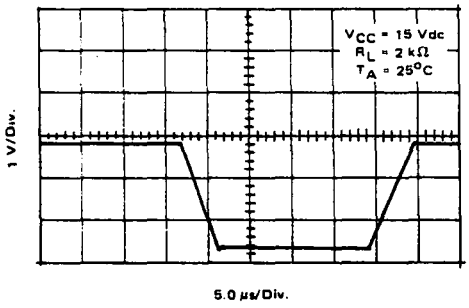


2

REPRESENTATIVE CIRCUIT SCHEMATIC
(One-Half of Circuit Shown)



LARGE SIGNAL VOLTAGE FOLLOWER RESPONSE



CIRCUIT DESCRIPTION

The LM158 Series is made using two internally compensated, two-stage operational amplifiers. The first stage of each consists of differential input devices Q20 and Q18 with input buffer transistors Q21 and Q17 and the differential to single ended converter Q3 and Q4. The first stage performs not only the first stage gain function but also performs the level shifting and transconductance reduction functions. By reducing the transconductance a smaller compensation capacitor (only 5 pF) can be employed, thus saving chip area. The transconductance reduction is accomplished by splitting the collectors of Q20 and Q18. Another feature of this input stage is that the input common-mode range can include the negative supply or ground, in single supply operation, without saturating either the input devices or the differential to single-ended converter. The second stage consists of a standard current source load amplifier stage.

Each amplifier is biased from an internal-voltage regulator which has a low temperature coefficient thus giving each amplifier good temperature characteristics as well as excellent power supply rejection.

TYPICAL PERFORMANCE CURVES

FIGURE 1 – INPUT VOLTAGE RANGE

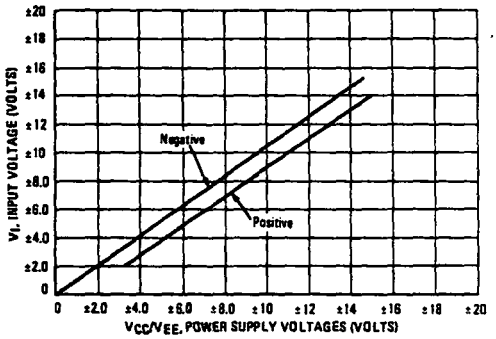


FIGURE 2 – OPEN LOOP FREQUENCY

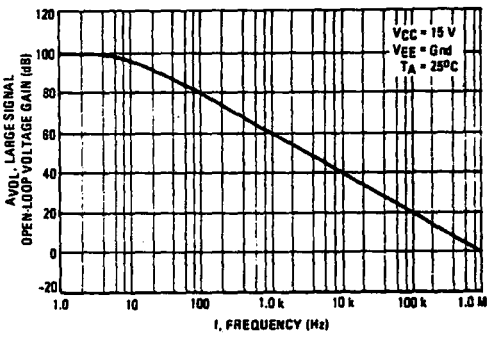


FIGURE 3 – LARGE-SIGNAL FREQUENCY RESPONSE

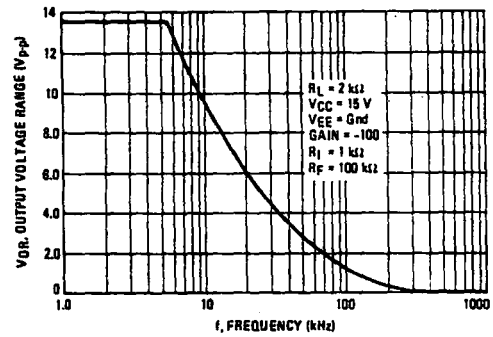


FIGURE 4 – SMALL-SIGNAL VOLTAGE FOLLOWER PULSE RESPONSE (Non-Inverting)

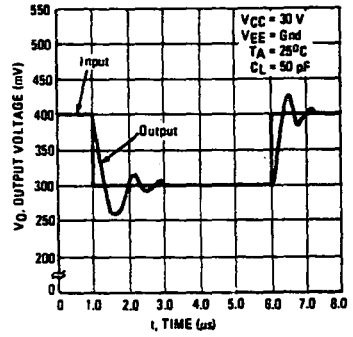


FIGURE 5 – POWER SUPPLY CURRENT versus POWER SUPPLY VOLTAGE

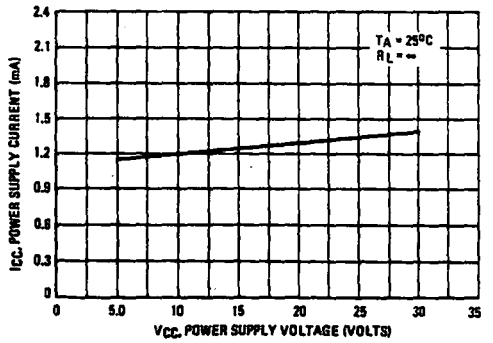
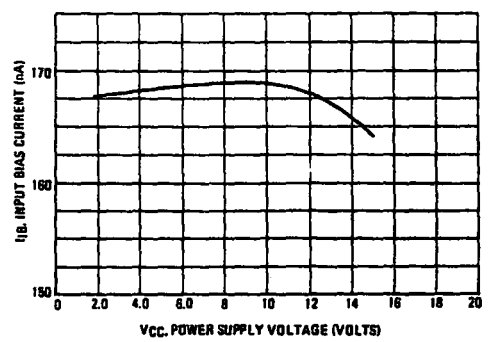
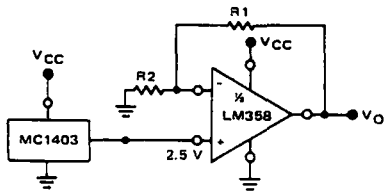


FIGURE 6 – INPUT BIAS CURRENT versus SUPPLY VOLTAGE



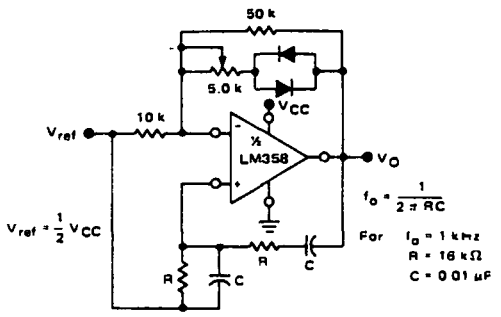
APPLICATIONS INFORMATION

FIGURE 7 – VOLTAGE REFERENCE



$$V_O = 2.5\,V\left(1 + \frac{R1}{R2}\right)$$

FIGURE 8 – WIEN BRIDGE OSCILLATOR

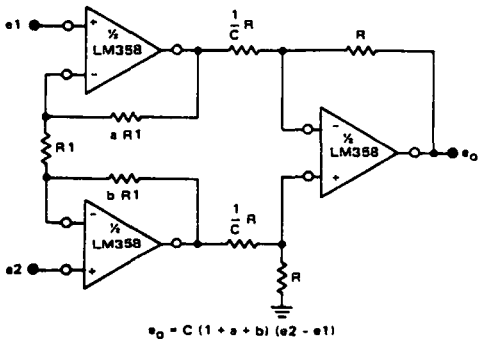


$$V_{ref} = \frac{1}{2} V_{CC}$$

$$f_o = \frac{1}{2\pi RC}$$

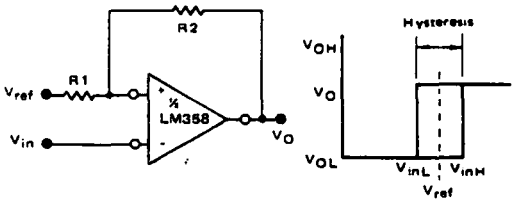
For f_o = 1 kHz
R = 16 kΩ
C = 0.01 μF

FIGURE 9 – HIGH IMPEDANCE DIFFERENTIAL AMPLIFIER



$$e_o = C\,(1 + a + b)\,(e2 - e1)$$

FIGURE 10 – COMPARATOR WITH HYSTERESIS

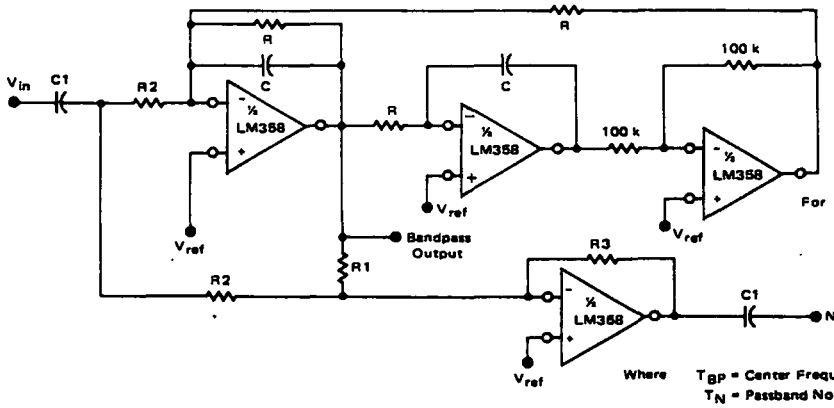


$$V_{inL} = \frac{R1}{R1 + R2}\,(V_{OL} - V_{ref}) + V_{ref}$$

$$V_{inH} = \frac{R1}{R1 + R2}\,(V_{OH} - V_{ref}) + V_{ref}$$

$$H = \frac{R1}{R1 + R2}\,(V_{OH} - V_{OL})$$

FIGURE 11 – BI-QUAD FILTER



$$f_o = \frac{1}{2\pi RC}$$

R1 = QR

$$R2 = \frac{R1}{T_{BP}}$$

$$V_{ref} = \frac{1}{2} V_{CC}$$

$R3 = T_N\,R2$

C1 = 10 C

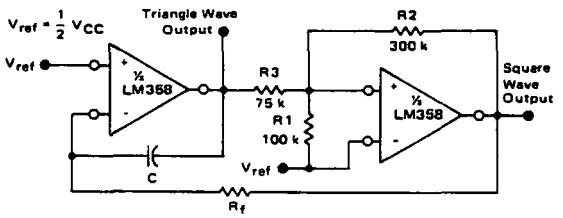
For f_o = 1 kHz
Q = 10
T_{BP} = 1
T_N = 1

R = 160 kΩ
C = 0.001 μF
R1 = 1.6 MΩ
R2 = 1.6 MΩ
R3 = 1.6 MΩ

Where T_{BP} = Center Frequency Gain
T_N = Passband Notch Gain

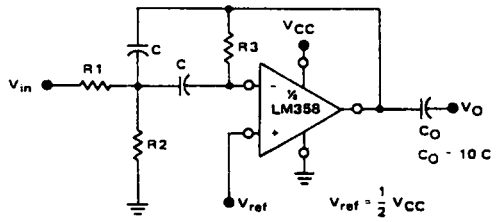
APPLICATIONS INFORMATION (continued)

FIGURE 12 – FUNCTION GENERATOR



$$f = \frac{R1 + R_C}{4 C R_f R1} \quad \text{if} \quad R3 = \frac{R2\ R1}{R2 + R1}$$

FIGURE 13 – MULTIPLE FEEDBACK BANDPASS FILTER



Given f_o = Center Frequency
$A(f_o)$ = Gain at Center Frequency

Choose Value f_o, C
Then

$$R3 = \frac{Q}{\pi f_o C}$$

$$R1 = \frac{R3}{2 A(f_o)}$$

$$R2 = \frac{R1\ R3}{4Q^2 R1 - R3}$$

For less than 10% error from operational amplifier

$$\frac{Q_o f_o}{BW} < 0.1$$ Where f_o and BW are expressed in Hz.

If source impedance varies, filter may be preceded with voltage follower buffer to stabilize filter parameters.

National Semiconductor

LM131A/LM131, LM231A/LM231, LM331A/LM331 Precision Voltage-to-Frequency Converters

General Description

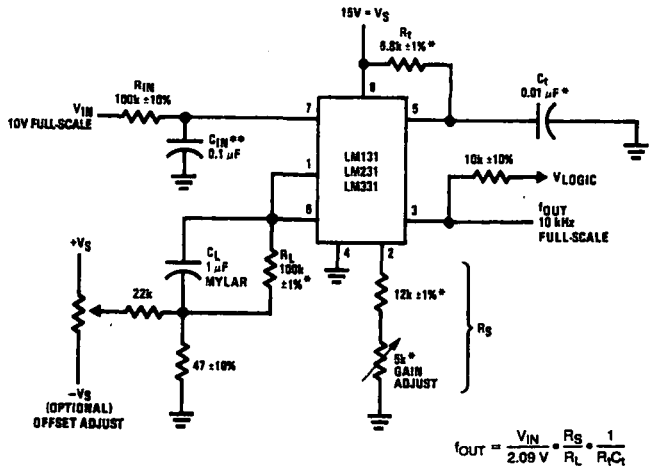
The LM131/LM231/LM331 family of voltage-to-frequency converters are ideally suited for use in simple low-cost circuits for analog-to-digital conversion, precision frequency-to-voltage conversion, long-term integration, linear frequency modulation or demodulation, and many other functions. The output when used as a voltage-to-frequency converter is a pulse train at a frequency precisely proportional to the applied input voltage. Thus, it provides all the inherent advantages of the voltage-to-frequency conversion techniques, and is easy to apply in all standard voltage-to-frequency converter applications. Further, the LM131A/LM231A/LM331A attains a new high level of accuracy versus temperature which could only be attained with expensive voltage-to-frequency modules. Additionally the LM131 is ideally suited for use in digital systems at low power supply voltages and can provide low-cost analog-to-digital conversion in microprocessor-controlled systems. And, the frequency from a battery powered voltage-to-frequency converter can be easily channeled through a simple photoisolator to provide isolation against high common mode levels.

The LM131/LM231/LM331 utilizes a new temperature-compensated band-gap reference circuit, to provide excellent accuracy over the full operating temperature range, at power supplies as low as 4.0V. The precision timer circuit has low bias currents without degrading the quick response necessary for 100 kHz voltage-to-frequency conversion. And the output is capable of driving 3 TTL loads, or a high voltage output up to 40V, yet is short-circuit-proof against V_{CC}.

Features

- Guaranteed linearity 0.01% max
- Improved performance in existing voltage-to-frequency conversion applications
- Split or single supply operation
- Operates on single 5V supply
- Pulse output compatible with all logic forms
- Excellent temperature stability, ±50 ppm/°C max
- Low power dissipation, 15 mW typical at 5V
- Wide dynamic range, 100 dB min at 10 kHz full scale frequency
- Wide range of full scale frequency, 1 Hz to 100 kHz
- Low cost

Typical Applications



*Use stable components with low temperature coefficients. See Typical Applications section.

**0.1μF or 1μF, See "Principles of Operation."

FIGURE 1. Simple Stand-Alone Voltage-to-Frequency Converter with ±0.03% Typical Linearity (f = 10 Hz to 11 kHz)

Absolute Maximum Ratings (Note 1)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

	LM131A/LM131	LM231A/LM231	LM331A/LM331
Supply Voltage	40V	40V	40V
Output Short Circuit to Ground	Continuous	Continuous	Continuous
Output Short Circuit to V_{CC}	Continuous	Continuous	Continuous
Input Voltage	−0.2V to +V_S	−0.2V to +V_S	−0.2V to +V_S
	T_{MIN} T_{MAX}	T_{MIN} T_{MAX}	T_{MIN} T_{MAX}
Operating Ambient Temperature Range	−55°C to +125°C	−25°C to +85°C	0°C to +70°C
Power Dissipation (P_D at 25°C) and Thermal Resistance (θ_{jA})			
(H Package) P_D	670 mW	570 mW	570 mW
θ_{jA}	150°C/W	150°C/W	150°C/W
(N Package) P_D		500 mW	500 mW
θ_{jA}		155°C/W	155°C/W
Lead Temperature (Soldering, 10 sec.)			
Dual-In-Line Package (Plastic)	260°C	260°C	260°C
Metal Can Package (TO-5)	260°C	260°C	260°C
ESD Susceptibility (Note 4)	TBD V	TBD V	TBD V

Electrical Characteristics T_A = 25°C unless otherwise specified (Note 2)

Parameter	Conditions	Min	Typ	Max	Units
VFC Non-Linearity (Note 3)	4.5V ≤ V_S ≤ 20V		±0.003	±0.01	% Full-Scale
	T_{MIN} ≤ T_A ≤ T_{MAX}		±0.006	±0.02	% Full-Scale
VFC Non-Linearity In Circuit of *Figure 1*	V_S = 15V, f = 10 Hz to 11 kHz		±0.024	±0.14	%Full-Scale
Conversion Accuracy Scale Factor (Gain)	V_{IN} = −10V, R_S = 14 kΩ				
LM131, LM131A, LM231, LM231A		0.95	1.00	1.05	kHz/V
LM331, LM331A		0.90	1.00	1.10	kHz/V
Temperature Stability of Gain	T_{MIN} ≤ T_A ≤ T_{MAX}, 4.5V ≤ V_S ≤ 20V				
LM131/LM231/LM331			±30	±150	ppm/°C
LM131A/LM231A/LM331A			±20	±50	ppm/°C
Change of Gain with V_S	4.5V≤V_S ≤ 10V		0.01	0.1	%/V
	10V ≤ V_S ≤ 40V		0.006	0.06	%/V
Rated Full-Scale Frequency	V_{IN} = −10V	10.0			kHz
Gain Stability vs Time (1000 Hrs)	T_{MIN} ≤ T_A ≤ T_{MAX}		±0.02		% Full-Scale
Overrange (Beyond Full-Scale) Frequency	V_{IN} = −11V	10			%
INPUT COMPARATOR					
Offset Voltage			±3	±10	mV
LM131/LM231/LM331	T_{MIN} ≤ T_A ≤ T_{MAX}		±4	±14	mV
LM131A/LM231A/LM331A	T_{MIN} ≤ T_A ≤ T_{MAX}		±3	±10	mV
Bias Current			−80	−300	nA
Offset Current			±8	±100	nA
Common-Mode Range	T_{MIN} ≤ T_A ≤ T_{MAX}	−0.2		V_{CC}−2.0	V
TIMER					
Timer Threshold Voltage, Pin 5		0.63	0.667	0.70	× V_S
Input Bias Current, Pin 5	V_S = 15V				
All Devices	0V≤$V_{PIN\,5}$ ≤ 9.9V		±10	±100	nA
LM131/LM231/LM331	$V_{PIN\,5}$ = 10V		200	1000	nA
LM131A/LM231A/LM331A	$V_{PIN\,5}$ = 10V		200	500	nA
$V_{SAT\ PIN\ 5}$ (Reset)	I = 5 mA		0.22	0.5	V

Electrical Characteristics $T_A = 25°C$ unless otherwise specified (Note 2) (Continued)

Parameter	Conditions	Min	Typ	Max	Units
CURRENT SOURCE (Pin 1)					
Output Current	$R_S = 14\ k\Omega$, $V_{PIN\ 1} = 0$				
LM131, LM131A, LM231, LM231A		126	135	144	μA
LM331, LM331A		116	136	156	μA
Change with Voltage	$0V \le V_{PIN\ 1} \le 10V$		0.2	1.0	μA
Current Source OFF Leakage					
LM131, LM131A			0.01	1.0	nA
LM231, LM231A, LM331, LM331A			0.02	10.0	nA
All Devices	$T_A = T_{MAX}$		2.0	50.0	nA
Operating Range of Current (Typical)			(10 to 500)		μA
REFERENCE VOLTAGE (Pin 2)					
LM131, LM131A, LM231, LM231A		1.76	1.89	2.02	V_{DC}
LM331, LM331A		1.70	1.89	2.08	V_{DC}
Stability vs Temperature			±60		ppm/°C
Stability vs Time, 1000 Hours			±0.1		%
LOGIC OUTPUT (Pin 3)					
V_{SAT}	$I = 5$ mA		0.15	0.50	V
	$I = 3.2$ mA (2 TTL Loads), $T_{MIN} \le T_A \le T_{MAX}$		0.10	0.40	V
OFF Leakage			±0.05	1.0	μA
SUPPLY CURRENT					
LM131, LM131A, LM231,	$V_S = 5V$	2.0	3.0	4.0	mA
LM231A	$V_S = 40V$	2.5	4.0	6.0	mA
LM331, LM331A	$V_S = 5V$	1.5	3.0	6.0	mA
	$V_S = 40V$	2.0	4.0	8.0	mA

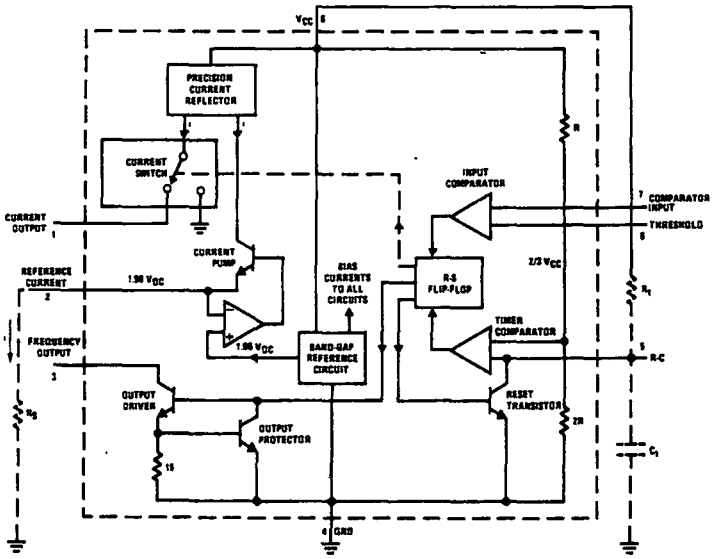
Note 1: Absolute Maximum Ratings indicate limits beyond which damage to the device may occur. DC and AC electrical specifications do not apply when operating the device beyond its specified operating conditions.

Note 2: All specifications apply in the circuit of *Figure 3*, with $4.0V \le V_S \le 40V$, unless otherwise noted.

Note 3: Nonlinearity is defined as the deviation of f_{OUT} from $V_{IN} \times (10\ kHz/-10\ V_{DC})$ when the circuit has been trimmed for zero error at 10 Hz and at 10 kHz, over the frequency range 1 Hz to 11 kHz. For the timing capacitor, C_T, use NPO ceramic, Teflon®, or polystyrene.

Note 4: Human body model, 100 pF discharged through a 1.5 kΩ resistor.

Functional Block Diagram



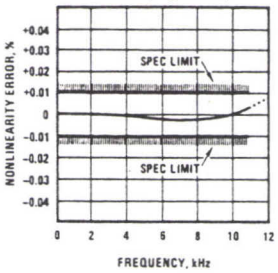
TL/H/5680–2

FIGURE 1a

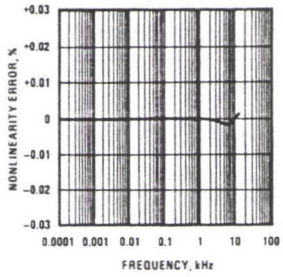
Typical Performance Characteristics

(All electrical characteristics apply for the circuit of *Figure 3*, unless otherwise noted.)

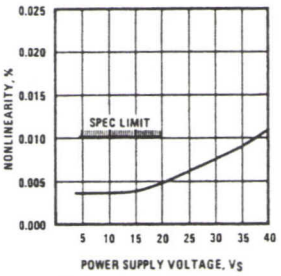
Nonlinearity Error, LM131 Family, as Precision V-to-F Converter (*Figure 3*)



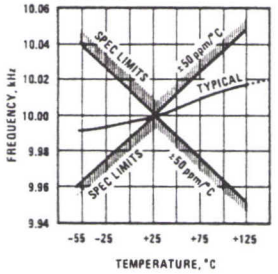
Nonlinearity Error, LM131 Family



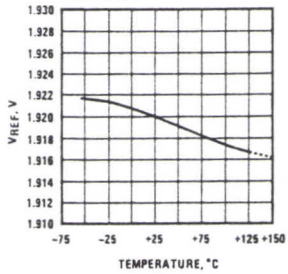
Nonlinearity vs Power Supply Voltage



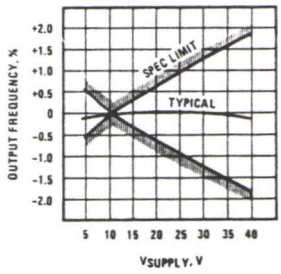
Frequency vs Temperature, LM131A



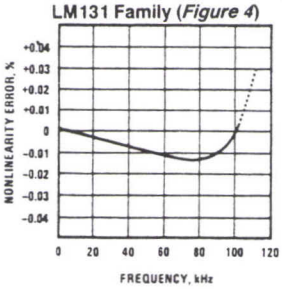
V_{REF} vs Temperature, LM131A



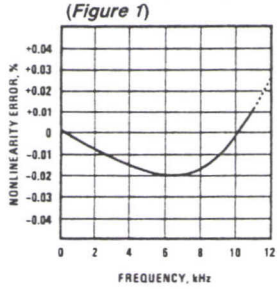
Output Frequency vs V_{SUPPLY}



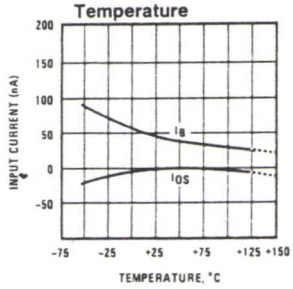
100 kHz Nonlinearity Error, LM131 Family (*Figure 4*)



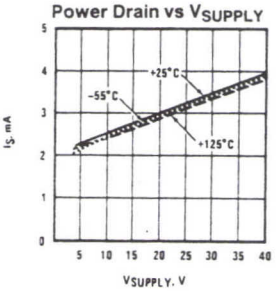
Nonlinearity Error, LM131 (*Figure 1*)



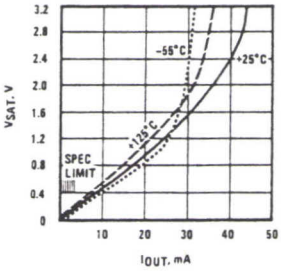
Input Current (Pins 6, 7) vs Temperature



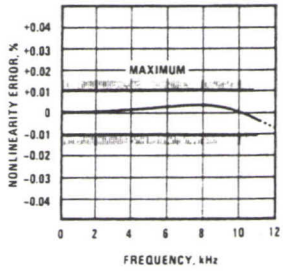
Power Drain vs V_{SUPPLY}



Output Saturation Voltage vs I_{OUT} (Pin 3)



Nonlinearity Error, Precision F-to-V Converter (*Figure 6*)



TL/H/5680–3

Typical Applications (Continued)

PRINCIPLES OF OPERATION OF A SIMPLIFIED VOLTAGE-TO-FREQUENCY CONVERTER

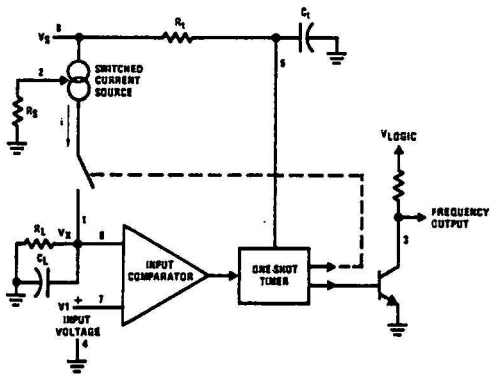
The LM131 is a monolithic circuit designed for accuracy and versatile operation when applied as a voltage-to-frequency (V-to-F) converter or as a frequency-to-voltage (F-to-V) converter. A simplified block diagram of the LM131 is shown in *Figure 2* and consists of a switched current source, input comparator, and 1-shot timer.

The operation of these blocks is best understood by going through the operating cycle of the basic V-to-F converter, *Figure 2*, which consists of the simplified block diagram of the LM131 and the various resistors and capacitors connected to it.

The voltage comparator compares a positive input voltage, V1, at pin 7 to the voltage, V_X, at pin 6. If V1 is greater, the comparator will trigger the 1-shot timer. The output of the timer will turn ON both the frequency output transistor and the switched current source for a period $t = 1.1\ R_tC_t$. During this period, the current i will flow out of the switched current source and provide a fixed amount of charge, $Q = i \times t$, into the capacitor, C_L. This will normally charge V_X up to a higher level than V1. At the end of the timing period, the current i will turn OFF, and the timer will reset itself.

Now there is no current flowing from pin 1, and the capacitor C_L will be gradually discharged by R_L until V_X falls to the level of V1. Then the comparator will trigger the timer and start another cycle.

The current flowing into C_L is exactly $I_{AVE} = i \times (1.1 \times R_tC_t) \times f$, and the current flowing out of C_L is exactly $V_X/R_L \cong V_{IN}/R_L$. If V_{IN} is doubled, the frequency will double to maintain this balance. Even a simple V-to-F converter can provide a frequency precisely proportional to its input voltage over a wide range of frequencies.



TL/H/5680–4

FIGURE 2. Simplified Block Diagram of Stand-Alone Voltage-to-Frequency Converter Showing LM131 and External Components

DETAIL OF OPERATION, FUNCTIONAL BLOCK DIAGRAM (*FIGURE 1a*)

The block diagram shows a band gap reference which provides a stable 1.9 V_{DC} output. This 1.9 V_{DC} is well regulated over a V_S range of 3.9V to 40V. It also has a flat, low temperature coefficient, and typically changes less than ½% over a 100°C temperature change.

The current pump circuit forces the voltage at pin 2 to be at 1.9V, and causes a current $i = 1.90V/R_S$ to flow. For $R_S = 14k$, $i = 135\ \mu A$. The precision current reflector provides a current equal to i to the current switch. The current switch switches the current to pin 1 or to ground depending on the state of the R_S flip-flop.

The timing function consists of an R_S flip-flop, and a timer comparator connected to the external R_tC_t network. When the input comparator detects a voltage at pin 7 higher than pin 6, it sets the R_S flip-flop which turns ON the current switch and the output driver transistor. When the voltage at pin 5 rises to ⅔ V_{CC}, the timer comparator causes the R_S flip-flop to reset. The reset transistor is then turned ON and the current switch is turned OFF.

However, if the input comparator still detects pin 7 higher than pin 6 when pin 5 crosses ⅔ V_{CC}, the flip-flop will not be reset, and the current at pin 1 will continue to flow, in its attempt to make the voltage at pin 6 higher than pin 7. This condition will usually apply under start-up conditions or in the case of an overload voltage at signal input. It should be noted that during this sort of overload, the output frequency will be 0; as soon as the signal is restored to the working range, the output frequency will be resumed.

The output driver transistor acts to saturate pin 3 with an ON resistance of about 50Ω. In case of overvoltage, the output current is actively limited to less than 50 mA.

The voltage at pin 2 is regulated at 1.90 V_{DC} for all values of i between 10 μA to 500 μA. It can be used as a voltage reference for other components, but care must be taken to ensure that current is not taken from it which could reduce the accuracy of the converter.

PRINCIPLES OF OPERATION OF BASIC VOLTAGE-TO-FREQUENCY CONVERTER (*FIGURE 1*)

The simple stand-alone V-to-F converter shown in *Figure 1* includes all the basic circuitry of *Figure 2* plus a few components for improved performance.

A resistor, $R_{IN} = 100\ k\Omega \pm 10\%$, has been added in the path to pin 7, so that the bias current at pin 7 (−80 nA typical) will cancel the effect of the bias current at pin 6 and help provide minimum frequency offset.

The resistance R_S at pin 2 is made up of a 12 kΩ fixed resistor plus a 5 kΩ (cermet, preferably) gain adjust rheostat. The function of this adjustment is to trim out the gain tolerance of the LM131, and the tolerance of R_t, R_L and C_t.

Typical Applications (Continued)

For best results, all the components should be stable low-temperature-coefficient components, such as metal-film resistors. The capacitor should have low dielectric absorption; depending on the temperature characteristics desired, NPO ceramic, polystyrene, Teflon or polypropylene are best suited.

A capacitor C_{IN} is added from pin 7 to ground to act as a filter for V_{IN}. A value of 0.01 μF to 0.1 μF will be adequate in most cases; however, in cases where better filtering is required, a 1 μF capacitor can be used. When the RC time constants are matched at pin 6 and pin 7, a voltage step at V_{IN} will cause a step change in f_{OUT}. If C_{IN} is much less than C_L, a step at V_{IN} may cause f_{OUT} to stop momentarily. A 47Ω resistor, in series with the 1 μF C_L, is added to give hysteresis effect which helps the input comparator provide the excellent linearity (0.03% typical).

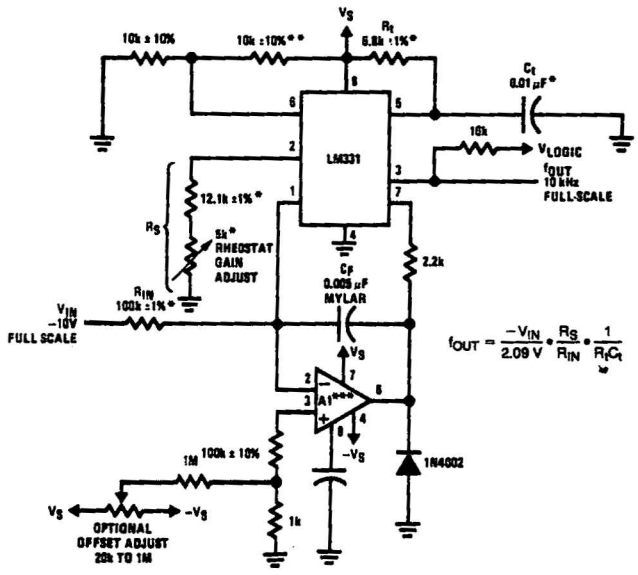
DETAIL OF OPERATION OF PRECISION V-TO-F CONVERTER (*FIGURE 3*)

In this circuit, integration is performed by using a conventional operational amplifier and feedback capacitor, C_F. When the integrator's output crosses the nominal threshold level at pin 6 of the LM131, the timing cycle is initiated.

The average current fed into the op amp's summing point (pin 2) is $i \times (1.1\ R_tC_t) \times f$ which is perfectly balanced with $-V_{IN}/R_{IN}$. In this circuit, the voltage offset of the LM131 input comparator does not affect the offset or accuracy of the V-to-F converter as it does in the stand-alone V-to-F converter; nor does the LM131 bias current or offset current. Instead, the offset voltage and offset current of the operational amplifier are the only limits on how small the signal can be accurately converted. Since op amps with voltage offset well below 1 mV and offset currents well below 2 nA are available at low cost, this circuit is recommended for best accuracy for small signals. This circuit also responds immediately to any change of input signal (which a stand-alone circuit does not) so that the output frequency will be an accurate representation of V_{IN}, as quickly as 2 output pulses' spacing can be measured.

In the precision mode, excellent linearity is obtained because the current source (pin 1) is always at ground potential and that voltage does not vary with V_{IN} or f_{OUT}. (In the stand-alone V-to-F converter, a major cause of non-linearity is the output impedance at pin 1 which causes i to change as a function of V_{IN}).

The circuit of *Figure 4* operates in the same way as *Figure 3*, but with the necessary changes for high speed operation.



TL/H/5680–5

*Use stable components with low temperature coefficients. See Typical Applications section.

**This resistor can be 5 kΩ or 10 kΩ for V_S = 8V to 22V, but must be 10 kΩ for V_S = 4.5V to 8V.

***Use low offset voltage and low offset current op amps for A1: recommended types LM108, LM308A, LF411A

FIGURE 3. Standard Test Circuit and Applications Circuit, Precision Voltage-to-Frequency Converter

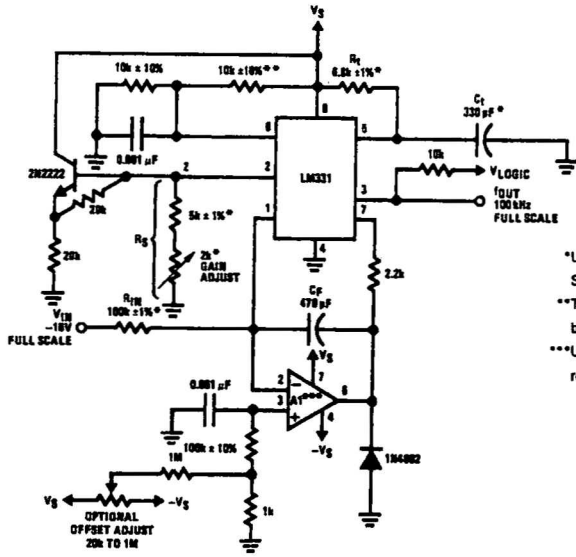
Typical Applications (Continued)

DETAILS OF OPERATION, FREQUENCY-TO-VOLTAGE CONVERTERS *(FIGURES 5 AND 6)*

In these applications, a pulse input at f_{IN} is differentiated by a C-R network and the negative-going edge at pin 6 causes the input comparator to trigger the timer circuit. Just as with a V-to-F converter, the average current flowing out of pin 1 is $I_{AVERAGE} = i \times (1.1\ R_tC_t) \times f$.

In the simple circuit of *FIGURE 5*, this current is filtered in the network $R_L = 100\ k\Omega$ and $1\ \mu F$. The ripple will be less than 10 mV peak, but the response will be slow, with a 0.1 second time constant, and settling of 0.7 second to 0.1% accuracy.

In the precision circuit, an operational amplifier provides a buffered output and also acts as a 2-pole filter. The ripple will be less than 5 mV peak for all frequencies above 1 kHz, and the response time will be much quicker than in *Figure 5*. However, for input frequencies below 200 Hz, this circuit will have worse ripple than *Figure 5*. The engineering of the filter time-constants to get adequate response and small enough ripple simply requires a study of the compromises to be made. Inherently, V-to-F converter response can be fast, but F-to-V response can not.



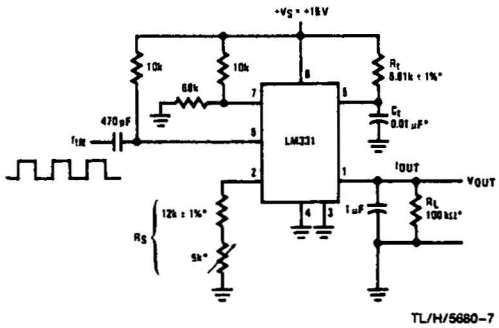
*Use stable components with low temperature coefficients. See Typical Applications section.

**This resistor can be 5 kΩ or 10 kΩ for V_S = 8V to 22V, but must be 10 kΩ for V_S = 4.5V to 8V.

***Use low offset voltage and low offset current op amps for A1: recommended types LF411A or LF356.

TL/H/5680–6

FIGURE 4. Precision Voltage-to-Frequency Converter, 100 kHz Full-Scale, ±0.03% Non-Linearity

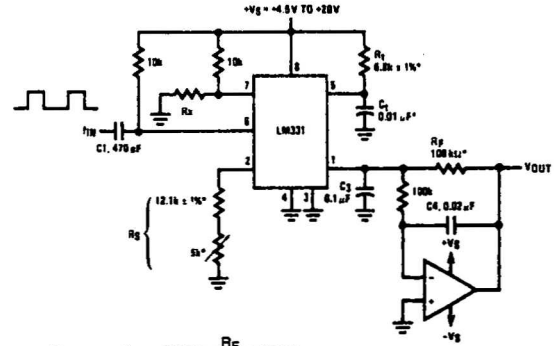


TL/H/5680–7

$$V_{OUT} = f_{IN} \times 2.09V \times \frac{R_L}{R_S} \times (R_tC_t)$$

*Use stable components with low temperature coefficients.

FIGURE 5. Simple Frequency-to-Voltage Converter, 10 kHz Full-Scale, ±0.06% Non-Linearity



$$V_{OUT} = -f_{IN} \times 2.09V \times \frac{R_F}{R_S} \times (R_tC_t)$$

TL/H/5680–8

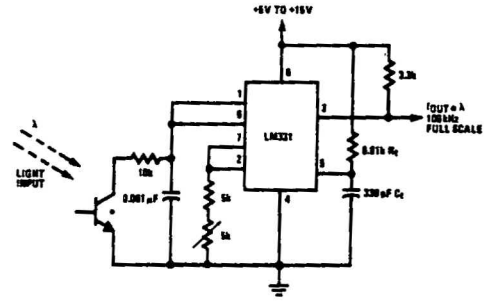
$$\text{SELECT } R_X = \frac{(V_S - 2V)}{0.2\ mA}$$

*Use stable components with low temperature coefficients.

FIGURE 6. Precision Frequency-to-Voltage Converter, 10 kHz Full-Scale with 2-Pole Filter, ±0.01% Non-Linearity Maximum

Typical Applications (Continued)

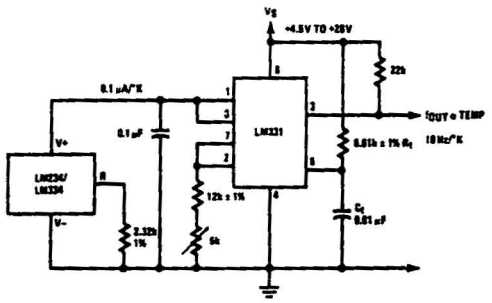
Light Intensity to Frequency Converter



TL/H/5680–9

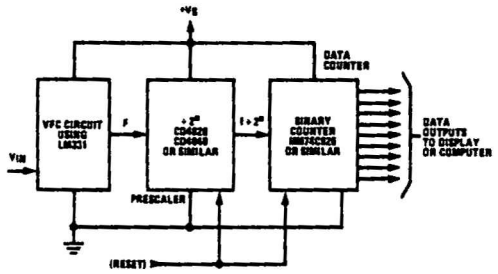
*L14F-1, L14G-1 or L14H-1, photo transistor (General Electric Co.) or similar

Temperature to Frequency Converter



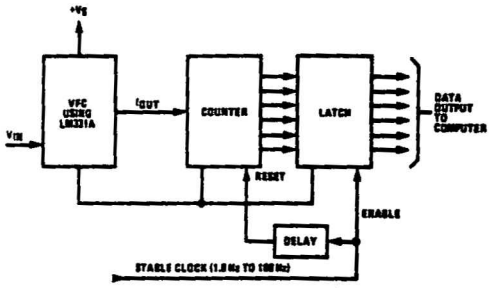
TL/H/5680–10

Long-Term Digital Integrator Using VFC



TL/H/5680–11

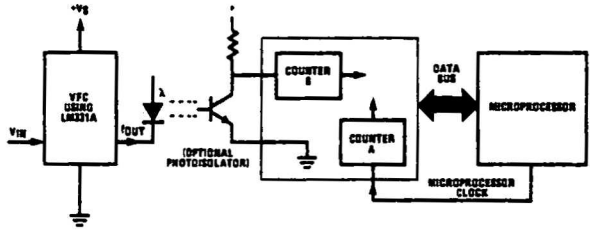
Basic Analog-to-Digital Converter Using Voltage-to-Frequency Converter



TL/H/5680–12

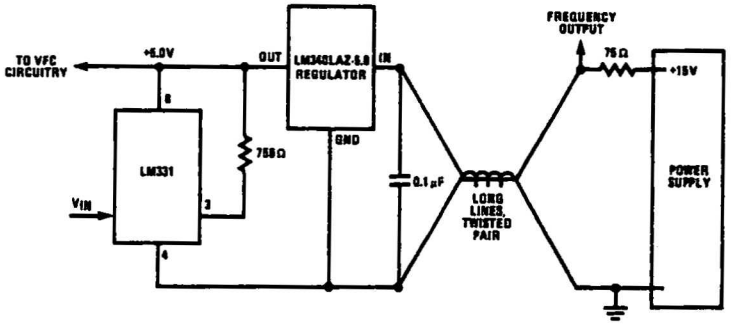
Typical Applications (Continued)

Analog-to-Digital Converter with Microprocessor



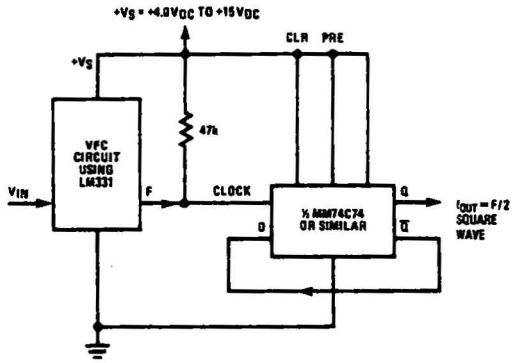
TL/H/5680–13

Remote Voltage-to-Frequency Converter with 2-Wire Transmitter and Receiver



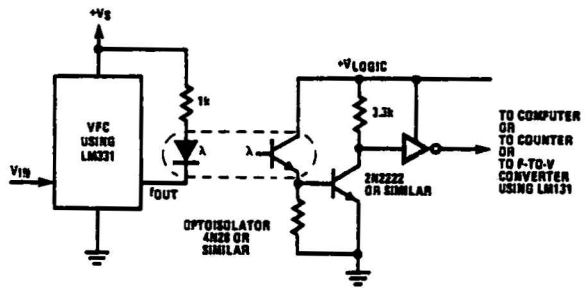
TL/H/5680–14

Voltage-to-Frequency Converter with Square-Wave Output Using ÷ 2 Flip-Flop



TL/H/5680–15

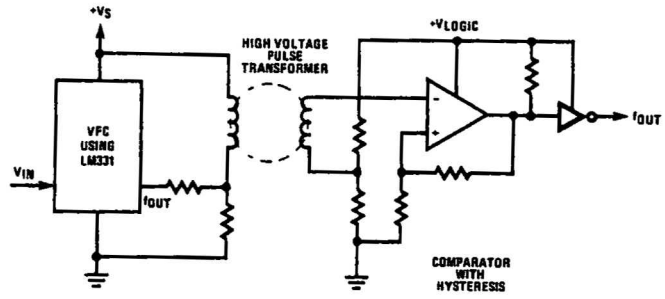
Voltage-to-Frequency Converter with Isolators



TL/H/5680–16

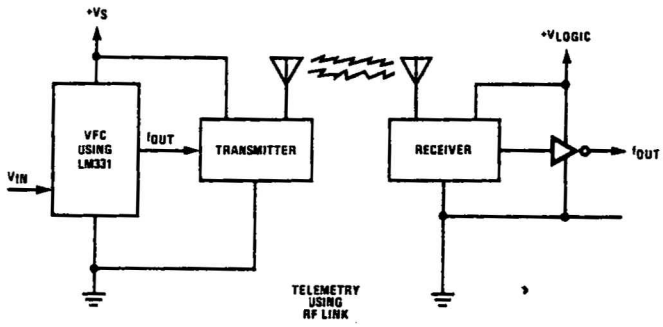
Typical Applications (Continued)

Voltage-to-Frequency Converter with Isolators



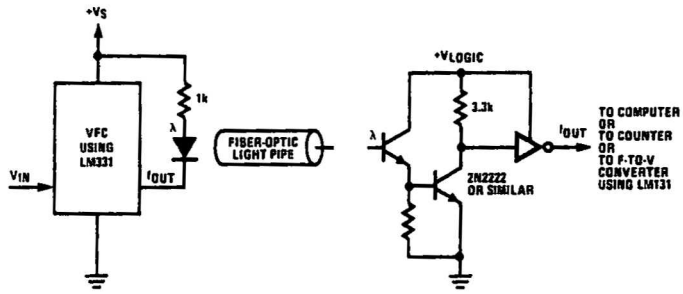
TL/H/5680–17

Voltage-to-Frequency Converter with Isolators



TL/H/5680–18

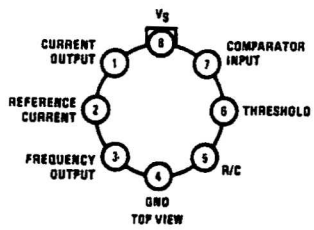
Voltage-to-Frequency Converter with Isolators



TL/H/5680–19

Connection Diagrams

Metal Can Package

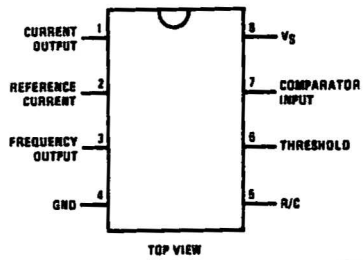


TL/H/5680–20

Note: Metal case is connected to pin 4 (GND.)

Order Number LM131AH, LM131H, LM231AH, LM231H, LM331AH or LM331H
See NS Package Number H08C

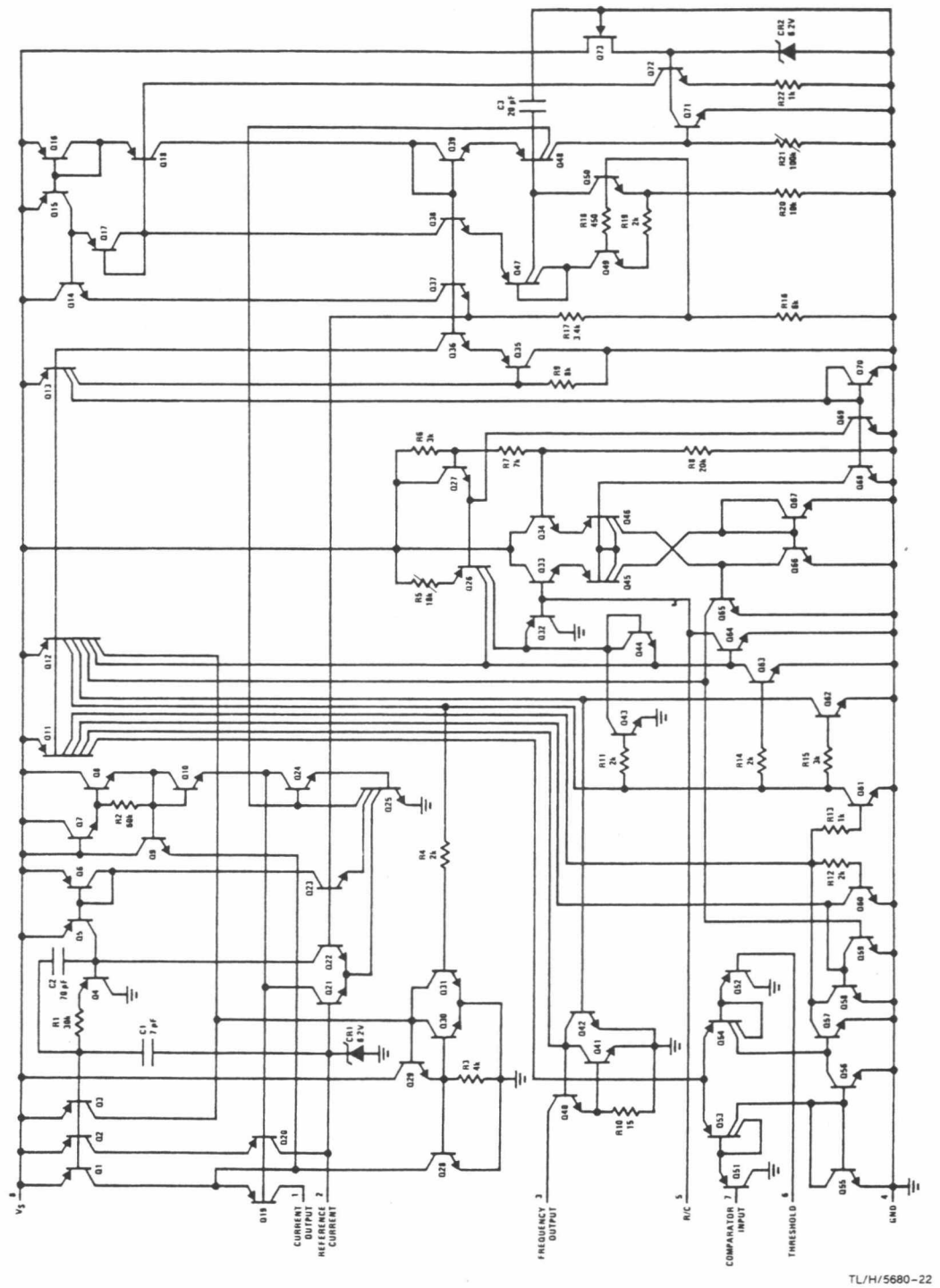
Dual-In-Line Package



TL/H/5680–21

Order Number LM231AN, LM231N, LM331AN, or LM331N
See NS Package Number N08E

Schematic Diagram



TL/H/5680–22

National Semiconductor Corporation

LM386 Low Voltage Audio Power Amplifier

General Description

The LM386 is a power amplifier designed for use in low voltage consumer applications. The gain is internally set to 20 to keep external part count low, but the addition of an external resistor and capacitor between pins 1 and 8 will increase the gain to any value up to 200.

The inputs are ground referenced while the output is automatically biased to one half the supply voltage. The quiescent power drain is only 24 milliwatts when operating from a 6 volt supply, making the LM386 ideal for battery operation.

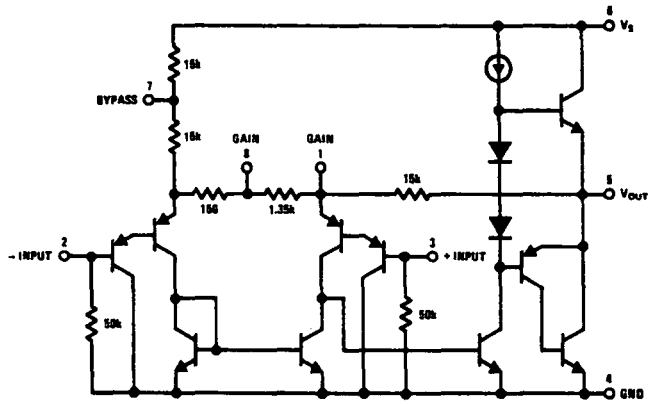
Features

- Battery operation
- Minimum external parts
- Wide supply voltage range — 4V–12V or 5V–18V
- Low quiescent current drain — 4 mA
- Voltage gains from 20 to 200
- Ground referenced input
- Self-centering output quiescent voltage
- Low distortion
- Eight pin dual-in-line package

Applications

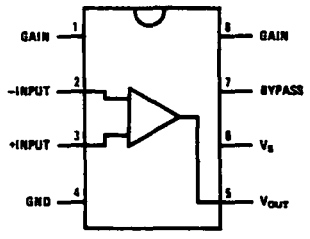
- AM-FM radio amplifiers
- Portable tape player amplifiers
- Intercoms
- TV sound systems
- Line drivers
- Ultrasonic drivers
- Small servo drivers
- Power converters

Equivalent Schematic and Connection Diagrams



TL/H/6976–1

Dual-In-Line and Small Outline Packages



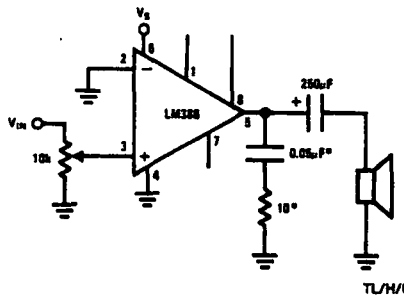
TL/H/6976–2

Top View

Order Number LM386M-1, LM386N-1, LM386N-3 or LM386N-4
See NS Package Number M08A or N08E

Typical Applications

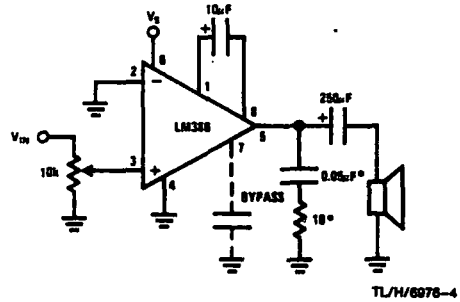
Amplifier with Gain = 20
Minimum Parts



TL/H/6976–3

*Required for LM386N-4 only.

Amplifier with Gain = 200



TL/H/6976–4

*Required for LM386N-4 only.

Absolute Maximum Ratings

If Military/Aerospace specified devices are required, contact the National Semiconductor Sales Office/ Distributors for availability and specifications.

Supply Voltage (LM386N-1, -3, LM386M-1)	15V
Supply Voltage (LM386N-4)	22V
Package Dissipation (Note 1) (LM386N-4)	1.25W
Input Voltage	±0.4V
Storage Temperature	−65°C to +150°C
Operating Temperature	0°C to +70°C
Junction Temperature	+150°C
Soldering Information	
Dual-In-Line Package	
Soldering (10 sec)	+260°C
Small Outline Package	
Vapor Phase (60 sec)	+215°C
Infrared (15 sec)	+220°C

See AN-450 "Surface Mounting Methods and Their Effect on Product Reliability" for other methods of soldering surface mount devices.

Electrical Characteristics $T_A = 25°C$

Parameter	Conditions	Min	Typ	Max	Units
Operating Supply Voltage (V_S)					
LM386N-1, -3, LM386M-1		4		12	V
LM386N-4		5		18	V
Quiescent Current (I_Q)	$V_S = 6V$, $V_{IN} = 0$		4	8	mA
Output Power (P_{OUT})					
LM386N-1, LM386M-1	$V_S = 6V$, $R_L = 8\Omega$, THD = 10%	250	325		mW
LM386N-3	$V_S = 9V$, $R_L = 8\Omega$, THD = 10%	500	700		mW
LM386N-4	$V_S = 16V$, $R_L = 32\Omega$, THD = 10%	700	1000		mW
Voltage Gain (A_V)	$V_S = 6V$, f = 1 kHz		26		dB
	10 μF from Pin 1 to 8		46		dB
Bandwidth (BW)	$V_S = 6V$, Pins 1 and 8 Open		300		kHz
Total Harmonic Distortion (THD)	$V_S = 6V$, $R_L = 8\Omega$, $P_{OUT} = 125$ mW f = 1 kHz, Pins 1 and 8 Open		0.2		%
Power Supply Rejection Ratio (PSRR)	$V_S = 6V$, f = 1 kHz, $C_{BYPASS} = 10$ μF Pins 1 and 8 Open, Referred to Output		50		dB
Input Resistance (R_{IN})			50		kΩ
Input Bias Current (I_{BIAS})	$V_S = 6V$, Pins 2 and 3 Open		250		nA

Note 1: For operation in ambient temperatures above 25°C, the device must be derated based on a 150°C maximum junction temperature and 1) a thermal resistance of 80°C/W junction to ambient for the dual-in-line package and 2) a thermal resistance of 170°C/W for the small outline package.

Application Hints

GAIN CONTROL

To make the LM386 a more versatile amplifier, two pins (1 and 8) are provided for gain control. With pins 1 and 8 open the 1.35 kΩ resistor sets the gain at 20 (26 dB). If a capacitor is put from pin 1 to 8, bypassing the 1.35 kΩ resistor, the gain will go up to 200 (46 dB). If a resistor is placed in series with the capacitor, the gain can be set to any value from 20 to 200. Gain control can also be done by capacitively coupling a resistor (or FET) from pin 1 to ground.

Additional external components can be placed in parallel with the internal feedback resistors to tailor the gain and frequency response for individual applications. For example, we can compensate poor speaker bass response by frequency shaping the feedback path. This is done with a series RC from pin 1 to 5 (paralleling the internal 15 kΩ resistor). For 6 dB effective bass boost: R ≅ 15 kΩ, the lowest value for good stable operation is R = 10 kΩ if pin 8 is open. If pins 1 and 8 are bypassed then R as low as 2 kΩ can be used. This restriction is because the amplifier is only compensated for closed-loop gains greater than 9.

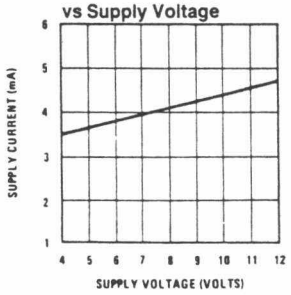
INPUT BIASING

The schematic shows that both inputs are biased to ground with a 50 kΩ resistor. The base current of the input transistors is about 250 nA, so the inputs are at about 12.5 mV when left open. If the dc source resistance driving the LM386 is higher than 250 kΩ it will contribute very little additional offset (about 2.5 mV at the input, 50 mV at the output). If the dc source resistance is less than 10 kΩ, then shorting the unused input to ground will keep the offset low (about 2.5 mV at the input, 50 mV at the output). For dc source resistances between these values we can eliminate excess offset by putting a resistor from the unused input to ground, equal in value to the dc source resistance. Of course all offset problems are eliminated if the input is capacitively coupled.

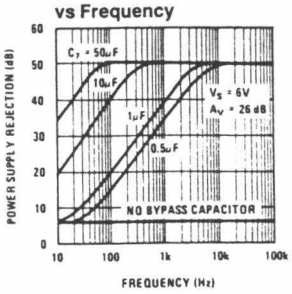
When using the LM386 with higher gains (bypassing the 1.35 kΩ resistor between pins 1 and 8) it is necessary to bypass the unused input, preventing degradation of gain and possible instabilities. This is done with a 0.1 μF capacitor or a short to ground depending on the dc source resistance on the driven input.

Typical Performance Characteristics

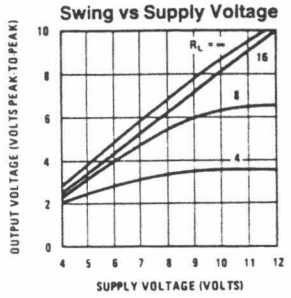
Quiescent Supply Current vs Supply Voltage



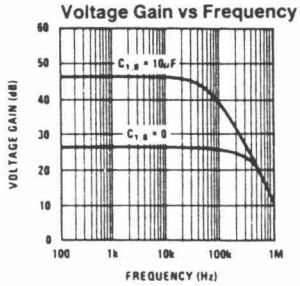
Power Supply Rejection Ratio (Referred to the Output) vs Frequency



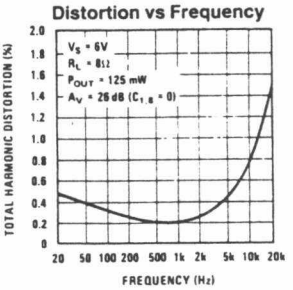
Peak-to-Peak Output Voltage Swing vs Supply Voltage



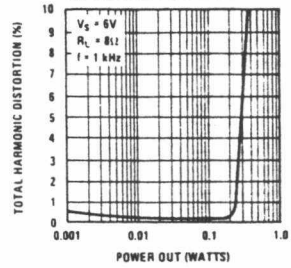
Voltage Gain vs Frequency



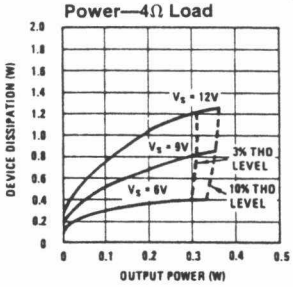
Distortion vs Frequency



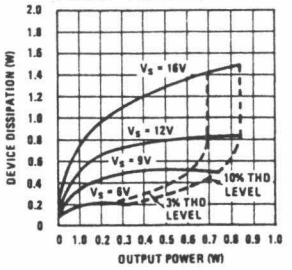
Distortion vs Output Power



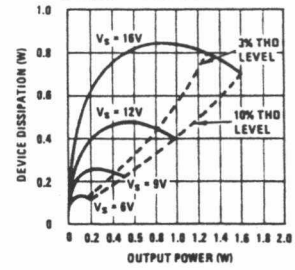
Device Dissipation vs Output Power—4Ω Load



Device Dissipation vs Output Power—8Ω Load



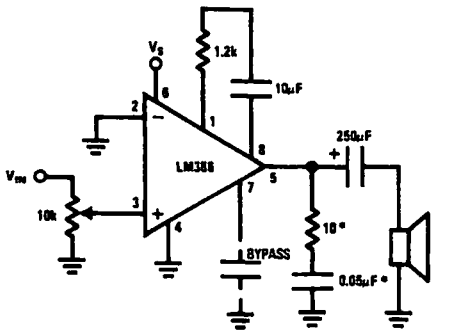
Device Dissipation vs Output Power—16Ω Load



TL/H/6976–5

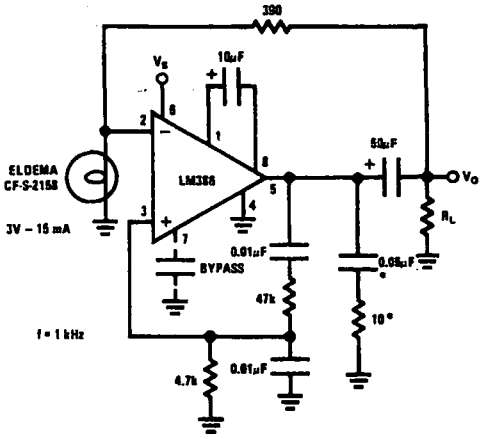
Typical Applications (Continued)

Amplifier with Gain = 50



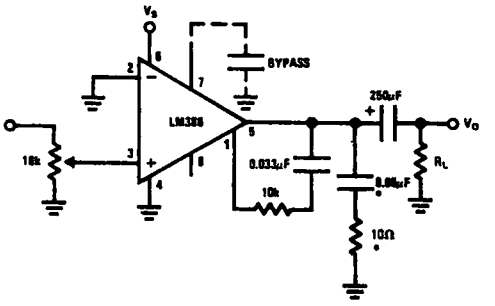
TL/H/6976–6

Low Distortion Power Wienbridge Oscillator



TL/H/6976–7

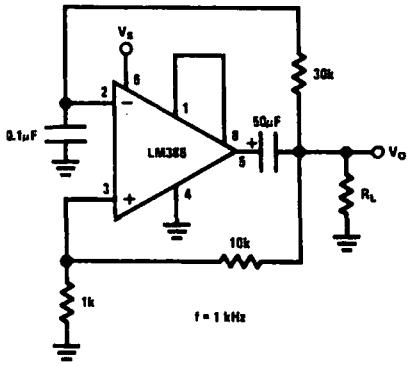
Amplifier with Bass Boost



TL/H/6976–8

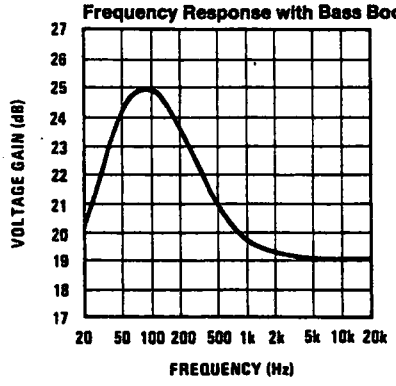
*Required for LM386N-4 only.

Square Wave Oscillator



TL/H/6976–9

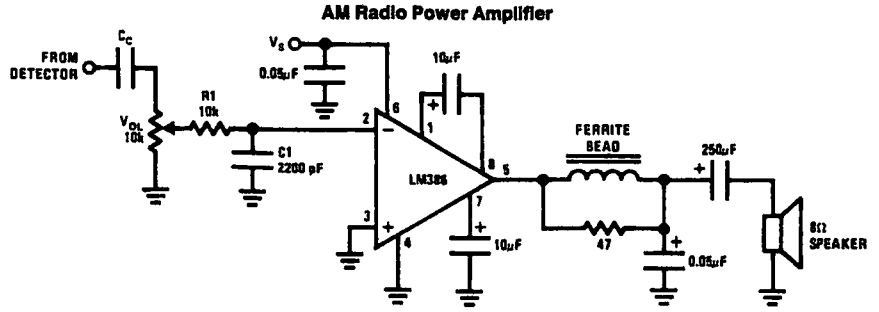
Frequency Response with Bass Boost



TL/H/6976–10

Typical Applications (Continued)

AM Radio Power Amplifier



TL/H/6976–11

Note 1: Twist supply lead and supply ground very tightly.

Note 2: Twist speaker lead and ground very tightly.

Note 3: Ferrite bead is Ferroxcube K5-001-001/3B with 3 turns of wire.

Note 4: R1C1 band limits input signals.

Note 5: All components must be spaced very close to IC.