เครื่องเชื่อมโยงสัญญาณโทรศัพท์อัตโนมัติ

(AUTOMATIC TELEPHONE PATCHER)

โดย

1 นายดำรงค์ฤทธ์ สันติปาตี 34132209

2 นายวรพจน์ เถาลัดดา 34132221

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต

ภาควิชาเทคนิคอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

2536

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

033325

หัวข้อปริญญานิพนธ์ เครื่องเชื่อมโยงสัญญาณโทรศัพท์อัตโนมัติ

Automatic Telephone Patcher.

โดย

นายดำรงค์ฤทธิ์ สันติปาตี 34132209

นายวรพจน์ เถาล้ดดา 34132221

อาจารย์ที่ปรึกษา

อาจารย์ ดร.ไพศาล นาคพิพัฒน์

ภาควิชาเทคนิคอุตสาหกรรม

ปีการศึกษา 2536

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

..... ประธานกรรมการ

( )

..... กรรมการ

( )

..... กรรมการ

( )

..... กรรมการ

( )

..... กรรมการ

( )

..... กรรมการ

( )

..... กรรมการ

( )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2536

ภาควิชาเทคนิคอุตสาหกรรม

คณะวิศวกรรมศาสตร์

เรื่อง เครื่องเชื่อมโยงสัญญาณโทรศัพท์อัตโนมัติ

Automatic Telephone Patcher.

ผู้จัดทำ

นายดำรงค์ฤทธิ์ สันติปาตี 34132209

นายวรพจน์ เถาลัดดา 34132221

.....อาจารย์ที่ปรึกษา

(ดร.ไพศาล นาคพิพัฒน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ

โครงงานนี้เป็นการสร้างอุปกรณ์เชื่อมโยงสัญญาณเสียงจากคู่สายโทรศัพท์ 2 เลขหมายเข้าด้วยกัน เพื่อประโยชน์ในการใช้โทรศัพท์สาธารณะติดต่อไปยังเลขหมายอื่น โดยเรียกผ่านเลขหมาย "NON CHARGE" การควบคุมการเชื่อมโยงสามารถกระทำได้จากระยะไกลผ่านคู่สายโทรศัพท์โดยอาศัยสัญญาณพื้นฐานของระบบโทรศัพท์อัตโนมัติแบบกดปุ่มที่ใช้อยู่ในปัจจุบัน ได้แก่สัญญาณกระดิ่ง และสัญญาณรหัสปุ่มหมายเลข อุปกรณ์สำคัญที่ใช้ในโครงงานนี้คือ ไมโครคอมพิวเตอร์ชิพเดี่ยว เบอร์ 8748 และไอซีถอดรหัสสัญญาณปุ่มหมายเลขเบอร์ MT8870 เครื่องเชื่อมโยงสัญญาณโทรศัพท์นี้ ถูกออกแบบให้มีความสามารถในการป้องกันผู้อื่นเขามาใช้โดยการกำหนดรหัสผ่าน, สามารถเปลี่ยนแปลง หรือยกเลิกรหัสผ่านได้จากระยะไกล (ผ่านคู่สายโทรศัพท์), จำกัดเวลาใช้งานไว้ 3 นาที เมื่อครบเวลาอุปกรณ์เชื่อมโยงจะส่งสัญญาณเตือนให้ผู้ใช้กดปุ่มเพื่อต่อเวลาหรือยกเลิกการใช้งานได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSTRACT

This is the project to build up an automatic telephone patching circuit. It is used for connecting voice signals to/from 2 telephone lines for certain purpose. The controls are accomplished by using fundamental of automatic telephone signalling such as ringing signal, and DTMF signal, remotely. The components using in this project are 8048 series Single chip Microcomputer and MT8870 DTMF Receiver. It is designed to prevent unauthorized by password establishment. The password can be changed or cancelled from remote via calling. The connection time are limitted for 3 minutes. The warning tone are issued by itself before disconnection. The connection can be extended or immediately stoped by push button dialer at the user telephone.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การที่โครงงาน Automatic Telephone Patcher นี้ ประสพผลสำเร็จได้ด้วยดีนั้น ทางคณะผู้จัดทำ ขอขอบพระคุณอย่างสูงต่อท่านทั้งหลาย ที่กรุณาให้ความอนุเคราะห์ ความร่วมมือ ให้คำแนะนำและคำปรึกษา และอำนวยความสะดวก คณะผู้จัดทำโครงงานขอขอบพระคุณ

องค์การโทรศัพท์แห่งประเทศไทย, ส่วนพัฒนาบุคคลากร, สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง ที่ได้เสนอแผนและให้ความร่วมมือในโครงการพัฒนาบุคคลากรของ ทศท. อันเป็นผลให้ คณะผู้จัดทำโครงงานมีโอกาสในการนำเสนอโครงงานซึ่งเป็นส่วนหนึ่งของการศึกษา

อาจารย์ ดร.ไพศาล นาคพิพัฒน์ ที่ได้กรุณาให้คำปรึกษาในการจัดทำโครงงานและในการเขียนปริญญานิพนธ์

อาจารย์ทุกท่านในภาควิชาเทคนิคอุตสาหกรรม นักศึกษารุ่นพี่และนักศึกษาร่วมรุ่น ที่มีโอกาสได้พูดคุยแลกเปลี่ยนความคิดเห็น ซึ่งเป็นประโยชน์ต่อโครงงาน

สุดท้ายขอขอบคุณผู้บังคับบัญชา ผู้ร่วมงานทุกท่าน และครอบครัวที่ยินยอมอุทิศเวลาส่วนหนึ่งซึ่งควรเป็นเวลาส่วนรวมสำหรับเขาเหล่านั้น เพื่อมีโอกาสทุ่มเทให้กับโครงงาน จนกระทั่งสำเร็จด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทที่ 1	บทนำ	
	1.1 ความเป็นมาของโครงงาน	1-1
	1.2 วัตถุประสงค์ของโครงงาน	1-3
	1.3 ขั้นตอนของการดำเนินงาน	1-4
บทที่ 2	ทฤษฎีและหลักการ	
	2.1 หลักการที่ใช้ในเชื่อมโยงสัญญาณโทรศัพท์	2-1
	2.2 ระบบสัญญาณเลขหมายผู้เช่าแบบ DTMF	2-3
	2.3 MT8870 DTMF Receiver	2-6
	2.4 ไมโครคอมพิวเตอร์ชิพเดี่ยวตะกูล 8048	2-16
บทที่ 3	การทำงาน,การออกแบบและการสร้าง	
	3.1 การทำงาน	3-1
	3.2 การออกแบบและการสร้าง Hardware	3-3
	3.2.1 ไมโครคอมพิวเตอร์	3-5
	3.2.2 วงจรถอดรหัสสัญญาณ DTMF	3-6
	3.2.3 วงจรตรวจจับสัญญาณกระดิ่ง	3-7
	3.2.4 วงจรควบคุมการเชื่อมโยงและแสดงผล	3-8
	3.3 การออกแบบ Software	3-9
	3.3.1 ส่วนฐานเวลา	3-10
	3.3.2 ส่วนรับข้อมูลการกดปุ่มหมายเลขโทรศัพท์	3-10
	3.3.3 ส่วนนับสัญญาณกระดิ่ง	3-10
	3.3.4 ส่วนกำเนิดสัญญาณเตือน	3-13
	3.3.5 ส่วนกำหนดรหัสผ่าน	3-13
	3.3.6 ส่วนตรวจสอบรหัสผ่าน	3-13
	3.3.7 ส่วนควบคุมการเชื่อมโยง	3-17
	3.3.8 โปรแกรมหลัก	3-22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4 การใช้งาน,การทดลองและสรุปผล 4-1

ภาคผนวก ก เครื่องเชื่อมโยงสัญญาณโทรศัพท์อัตโนมัติ

ภาคผนวก ข 8748H/8035H/8749H/8039H

HMOS-E SINGLE-COMPONENT 8-BIT MICROCOMPUTER

ภาคผนวก ค MCS-48 INSTRUCTION SET

ภาคผนวก ง MT8870 Integrated DTMF Reciever

ภาคผนวก จ การใช้ Microcomputer แทนวงจรลอจิค

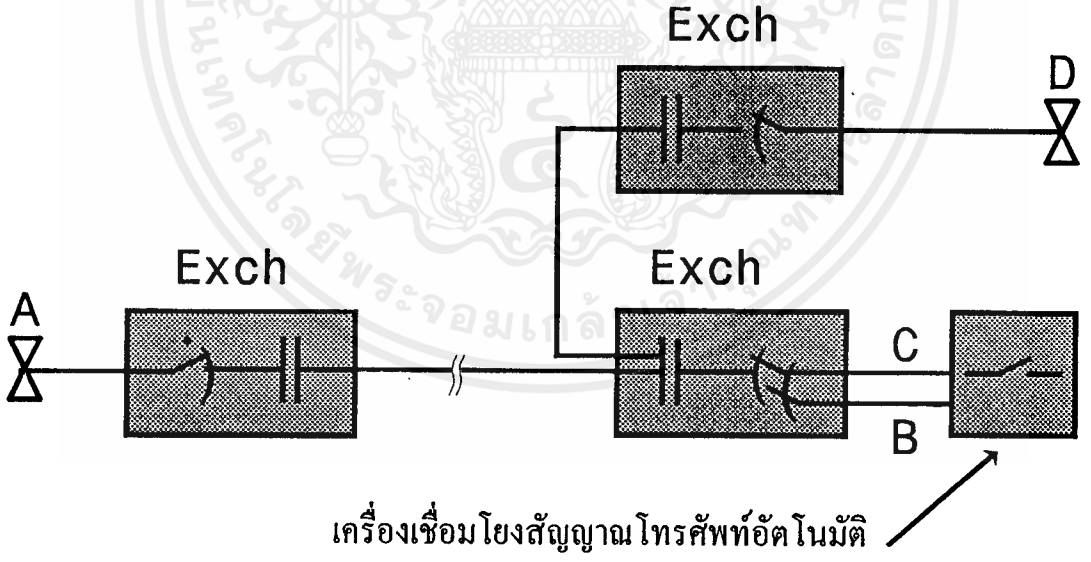
เอกสารอ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของโครงงาน

แนวความคิดในการทำโครงงานนี้ เกิดจากผู้จัดทำโครงงานเป็นพนักงานขององค์การโทรศัพท์ฯ ที่ต้องออกปฏิบัติงานในต่างจังหวัด มีความจำเป็นต้องใช้โทรศัพท์เพื่อการติดต่อประสานงาน ทั้งในและนอกเวลาทำการ เพื่อความสะดวกในการติดต่อ ที่ทำงานจึงมีเลขหมายโทรศัพท์ "NON CHARGE" คือเลขหมายที่ถูกกำหนดไว้ว่าไม่เสียค่าบริการเมื่อเรียกเข้า หมายความว่าถ้าใช้โทรศัพท์สาธารณะเรียกเข้า เมื่อสนทนาจบ และวางหู เครื่องจะคืนเหรียญ



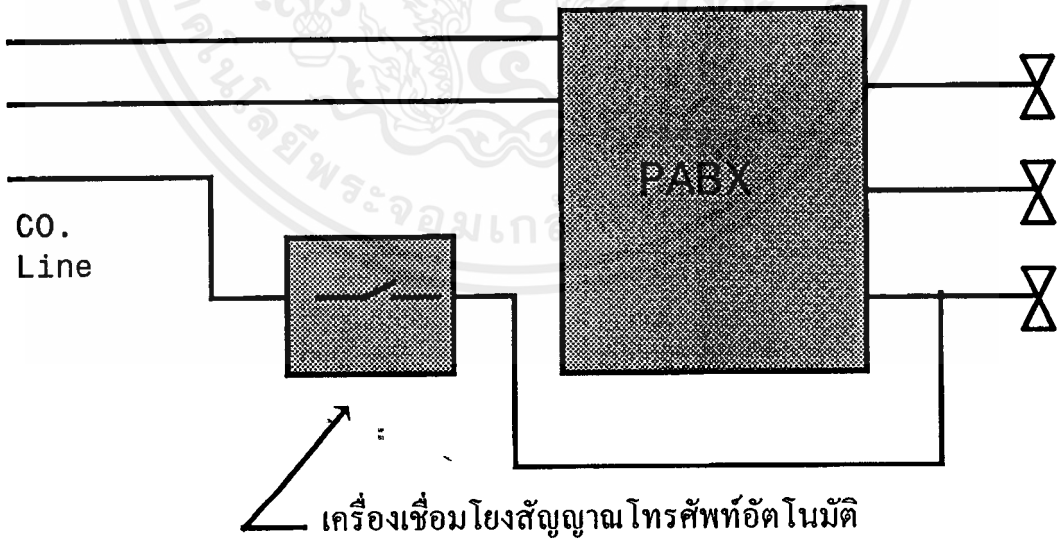
รูปที่ 1.1 แสดงการต่อเครื่องเชื่อมโยงสัญญาณโทรศัพท์
กับหมายเลข NON CHARGE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บางครั้งนอกเวลาทำการ มีความจำเป็นต้องติดต่อกับผู้บังคับบัญชาหรือครอบครัว จึงคิดสร้างอุปกรณ์ที่ช่วยเชื่อมโยงให้โดยอัตโนมัติ โดยต่อเครื่องเชื่อมโยงนี้เข้ากับคู่สายของเลขหมายโทรศัพท์ของที่ทำงานในกรุงเทพฯ ซึ่งเป็นเลขหมาย "NON CHARGE" และกับคู่สายของเลขหมายอื่นอีก 1 คู่สาย ผู้ใช้เพียงเรียกเข้าเลขหมาย "NON CHARGE" เหมือนการใช้โทรศัพท์ธรรมดา อุปกรณ์ช่วยเชื่อมโยงจะทำหน้าที่รับสาย, ยกหูโทรศัพท์ของอีกเลขหมายหนึ่งขึ้น และเชื่อมโยงสายทั้งสองเข้าด้วยกัน ผู้ใช้จะได้รับ Dial tone และสามารถกดปุ่มหมายเลขปลายทางได้

นอกจากใช้ประโยชน์ดังกล่าวข้างต้นแล้วยังสามารถนำมาใช้งานในกรณีอื่นได้อีกคือ

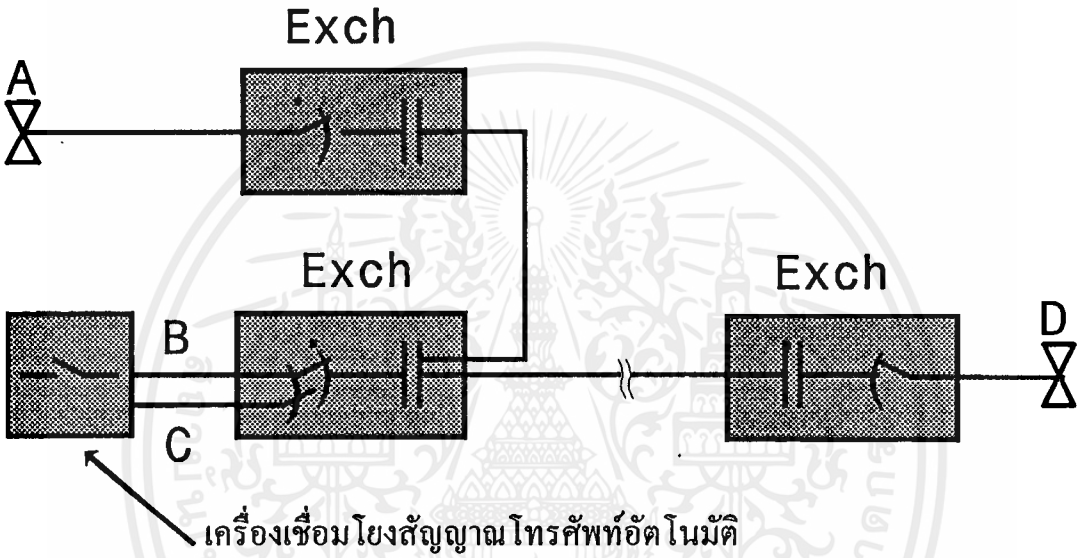
1 ใช้ช่วยให้ PABX รุ่นเก่ามีความสามารถ ในการต่อเข้าภายในโดยอัตโนมัติ (Direct Inward Dialer หรือ DID) โดยการเชื่อมโยงหมายเลขภายนอกหมายเลขหนึ่งเข้ากับคู่สายภายใน(Extension Line) คู่สายหนึ่ง



รูปที่ 2 แสดงการต่อใช้งานร่วมกับ PABX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2 ในบางกิจการ ที่ยอมรับภาระค่าโทรศัพท์ทางไกลของพนักงาน ในการติดต่อกับลูกค้าสามารถติดตั้งเครื่องเชื่อมโยงนี้ในสำนักงาน พนักงานสามารถใช้โทรศัพท์สาธารณะหรือที่บ้านโทรเข้ามายังสำนักงาน และต่อโทรศัพท์ทางไกลออกต่างจังหวัดผ่านเครื่องเชื่อมโยงนี้ได้ เหมือนกับพนักงานผู้นั้นใช้โทรศัพท์ในสำนักงาน



รูปที่ 3 แสดงการต่อใช้งานในกิจการทั่วไปที่ยอมรับภาระค่าโทรศัพท์ทางไกล

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 เพื่อเรียนรู้การทำงานของ Single Chip Microcomputer และการนำมาใช้ในงานควบคุมแทนวงจร Digital ที่ซับซ้อน

1.2.2 เรียนรู้ระบบสัญญาณเรียกของผู้เช่า (Subscriber's address signal) แบบสัญญาณหลายความถี่ (Multi-Frequency Signal system) และการใช้ IC เบอร์ MT8870 DTMF Receiver ที่ถูกผลิตขึ้นเพื่อใช้สำหรับถอดระหัสสัญญาณหมายเลขของเครื่องโทรศัพท์แบบกดปุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.3 ศึกษาเทคนิคการ Interface ระหว่าง Microcompoter กับอุปกรณ์ตรวจจับสัญญาณจากภายนอก และการส่งสัญญาณไปยังอุปกรณ์ภายนอก

1.2.4 เรียนรู้วิธีการทำงาน เพื่อให้ได้ผลิตภัณฑ์หรือชิ้นงานที่สามารถนำไปใช้ประโยชน์ตามความต้องการได้จริง

1.3 ขั้นตอนของการดำเนินงาน

ศึกษาคุณสมบัติของ Microprocessor ตระกูลต่างๆ และเลือกใช้เบอร์ที่เหมาะสมกับความต้องการในโครงงานนี้

ออกแบบสร้างและทดลองวงจรต่างๆ ทีละส่วน

นำวงจรต่างๆ ที่ผ่านการทดลองขั้นต้นมารวมกัน สร้างขึ้นเป็นต้นแบบ

พัฒนาโปรแกรมควบคุมระบบ

ทดสอบ แก้ไขปรับปรุงโปรแกรมและวงจรให้ทำงานได้ดีขึ้น

สร้างชิ้นงานจริง

ทดลองใช้งานจริง ทดสอบ และสรุปผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

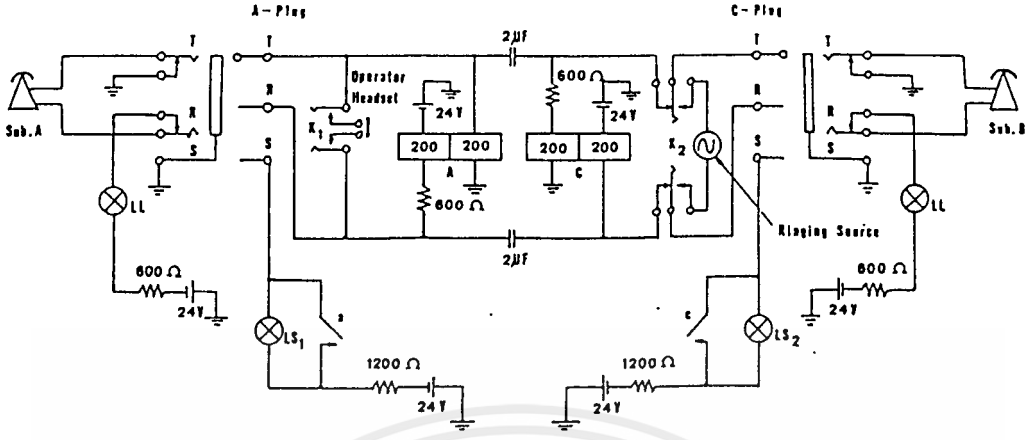
ทฤษฎีและหลักการ

2.1 หลักการเชื่อมโยงสัญญาณโทรศัพท์

สัญญาณเสียงที่ส่งผ่านคู่สายโทรศัพท์เป็นสัญญาณที่ซ้อนทับกับกระแสไฟตรง การเชื่อมโยงสัญญาณโทรศัพท์ 2 คู่สายเข้าด้วยกันอาจทำอย่างง่ายๆ โดยการเชื่อมโยงโดยตรงตามรูปที่ 2.1.1 ในการเชื่อมโยงเช่นนี้จะต้องคำนึงถึง DC Voltage ที่ปรากฏบนสายทั้งสองกล่าวคือตรงกันและมีเครื่องโทรศัพท์เครื่องๆ หนึ่งยกหูอยู่เพื่อให้เกิด DC Loop ขึ้นที่ Line Circuit ภายในอุปกรณ์ชุมสายของเลขหมายทั้งสองวงจรเชื่อมโยงที่นำมาใช้งาน ได้มาจากวงจรในตู้สลับสายโทรศัพท์ แบบพนักงานต่อแบบแบตเตอรี่ร่วมซึ่งยังมีใช้งานเป็นตู้สาขาภายใน (PABX) เนื่องจากราคาถูกมากและค่าใช้จ่ายรวมทั้งค่าบำรุงรักษาต่ำ รูปที่ 2.1.2 คือวงจรเชื่อมโยงที่กล่าวถึง

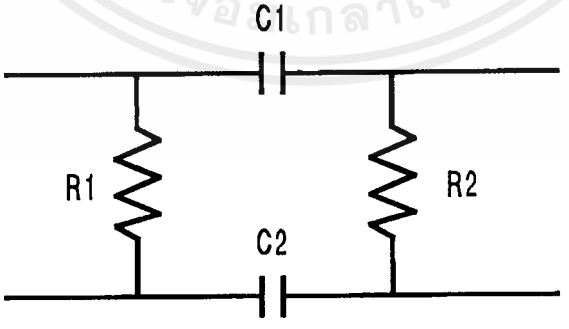
รูปที่ 2.1.1 การเชื่อมโยงสัญญาณโทรศัพท์อย่างง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1.2 วงจรตู้สลับสายโทรศัพท์ระบบแบตเตอรี่ร่วม

ตามรูป จะเห็น คาปาซิเตอร์สองตัวต่ออยู่เพื่อป้องกันการรบกวนกันของ DC Voltage จากวงจรปากพูดหูฟังของเครื่องโทรศัพท์ทั้งสอง การรบกวนที่ว่าหมายถึง การรบกวนของสถานะ การวางหูยกหู ซึ่งถ้าไม่ใช้คาปาซิเตอร์เป็นตัวกันแรงไฟ DC จะไม่สามารถจำแนกได้ว่าผู้เช่าฝ่ายใดวางหูหรือยกหู สัญญาณเสียงพูดซึ่งเป็น AC จะสามารถผ่านคาปาซิเตอร์ได้โดยเกิดความสูญเสียเพียงเล็กน้อย



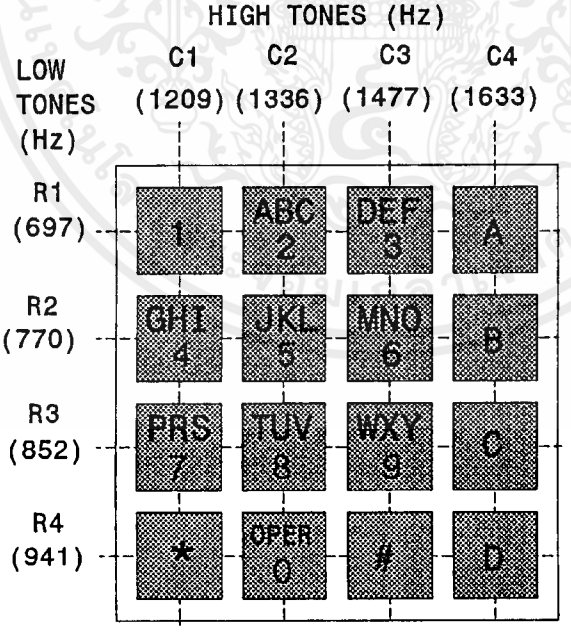
รูปที่ 2.1.3 แสดงวงจรเชื่อมโยงที่ใช้ในโครงงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักดังกล่าวนำมาใช้กับเครื่องเชื่อมโยงในโครงงานนี้โดยเขียนวงจรขึ้นใหม่ ดังรูปที่ 2.1.3 รีซีสเตอร์ R1 และ R2 ใช้แทนค่าความต้านทานภายในของเครื่องโทรศัพท์ในรูปที่ 2.1.1 ส่วนคาปาซิเตอร์ทั้งสองทำหน้าที่เช่นเดียวกับคาปาซิเตอร์ในรูปที่ 2.1.2

2.2 ระบบสัญญาณเลขหมายผู้เช่าแบบ DTMF

DTMF (Dual Tone Multi Frequency) ถูกพัฒนาขึ้นโดยห้องทดลองของ BELL เป็นระบบ Dialing ที่เร็วและเชื่อถือได้มากกว่าระบบการใช้หน้าปัทม์หมุนแบบเก่า ระบบสัญญาณ DTMF เป็นระบบ Tone Dialing หรือ TOUCH-TONE (TOUCH-TONE เป็นเครื่องหมายการค้าของ AT&T) และถูกนำมาใช้กับโทรศัพท์แบบกดปุ่มและอุปกรณ์อื่นๆ



รูปที่ 2.2.1 แสดงการวางตำแหน่ง แป้นกด DTMF มาตราฐาน พร้อมทั้งความถี่ของแต่ละปุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ DTMF มาตราฐานประกอบด้วยสัญญาณเสียงคู่หนึ่งซึ่งถูกเลือกมาจากกลุ่มความถี่มาตราฐาน 8 ความถี่ ความถี่มาตราฐานทั้งแปดถูกแบ่งเป็นสองกลุ่ม คือ กลุ่มความถี่ต่ำ 4 ความถี่ และความถี่สูง 4 ความถี่ สัญญาณ DTMF ที่ถูกต้องประกอบด้วย ผลรวมของสัญญาณความถี่จากกลุ่มความถี่ต่ำและกลุ่มความถี่สูงอย่างละหนึ่งความถี่ เพราะฉะนั้นจึงมีสัญญาณ DTMF ที่เป็นไปได้ทั้งหมด 16 รหัส ความถี่มาตราฐานในกลุ่มความถี่ต่ำได้แก่ 697, 770, 852,และ 941 Hz และเราเรียกความถี่ในกลุ่มนี้ว่าความถี่แถว R1,R2,R3,และ R4 ตามลำดับ ความถี่มาตราฐานในกลุ่มความถี่สูงได้แก่ 1209, 1336, 1477, และ 1633 Hz และเรียกว่าความถี่คอลัมน์ C1, C2, C3, และ C4 ตามลำดับ สัญญาณ DTMF ใดๆ กำเนิดขึ้นได้โดยอาศัยแป้นกดแมตทริกขนาด 4 X 4 ดังแสดงในรูปที่ 2.2.1

การวางรูปของแป้นกดดังรูปที่ 2.2.1 ถูกกำหนดให้เป็นมาตราฐานนานาชาติ ความถี่ที่เกิดจากตัวกำเนิดสัญญาณ DTMF ยอมให้เบี่ยงเบนไปจากมาตราฐาน 1.5% จะสังเกตุได้ว่าความถี่ทั้งหมดอยู่ในย่านความถี่เสียงของระบบโทรศัพท์

การเลือกความถี่มาตราฐาน ไม่ได้เลือกตามใจชอบโดยปราศจากความหมาย ผู้ออกแบบระบบ DTMF ต้องใช้ความระมัดระวังอย่างมากในการในการเลือก ความถี่อื่นที่อาจปรากฏในสายโทรศัพท์เช่น Dial tone สัญญาณรบกวนจากกระแสไฟฟ้าจะต้องไม่มาตรงกับย่านของสัญญาณ DTMF ยิ่งไปกว่านั้น Harmonic ของมันจะต้องไม่รบกวนกัน ดังนั้น ความถี่สูงสุด (1633 Hz) ต้องต่ำกว่า Harmonic ที่ 3 ของความถี่ต่ำสุด (697 Hz)

โทรศัพท์ที่ใช้ DTMF โดยทั่วไปจะมีแป้นกดขนาด 3 X 4 ปุ่ม แทนหมายเลข 0 - 9 และที่เหลืออีก 2 ปุ่มคือ * และ # ซึ่งสามารถนำไปใช้กับวัตถุประสงค์อื่นได้อีกหลายอย่าง แป้นกดขนาด 3 X 4 แทนความถี่แถว R1-R4 และความถี่คอลัมน์ C1-C3 โทรศัพท์ที่ใช้เพื่อวัถุประสงค์พิเศษบางเครื่องใช้ C4 ด้วย และเพิ่มสัญลักษณ์ขึ้นอีก 4 ตัวคือ A, B, C,และ D (ดังแสดงในรูปที่ 2.2.1) เพื่อให้สามารถเข้ารหัสสัญญาณ DTMF ได้ทั้ง 16 รหัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถได้ยินสัญญาณ DIMF ได้โดยเพียงยกหูโทรศัพท์ขึ้นฟังแล้วกดปุ่มใดปุ่มหนึ่ง เช่นกดปุ่มหมายเลข 8 จะเกิดความถี่ 852 Hz(R3) และ 1336 Hz(C2) ขึ้นพร้อมกัน สัญญาณนี้จะถูกถอดรหัส ด้วยตัวรับสัญญาณ DIMF ที่ชุมสายโทรศัพท์

ภายในอุปกรณ์ชุมสายโทรศัพท์ เลขหมาย 3 ตัวแรกจะเป็นตัวบอกถึงพื้นที่การให้บริการ หลังจากทำการเชื่อมโยงระหว่างผู้เรียกและผู้ถูกเรียกแล้ว ตัวรับสัญญาณ DIMF ภายในชุมสายก็จะหมดหน้าที่และไม่รับรู้สัญญาณ DIMF อีกต่อไป ผู้เช่าทั้งสองสามารถใช้แป้นกดเพื่อกำเนิดสัญญาณแบบสถานี-ต่อ-สถานี ได้โดยอิสระ

ในอดีตการเข้ารหัส DIMF ในกิจการโทรศัพท์ใช้วงจร LC ที่มีขนาดใหญ่และทรานซิสเตอร์ แต่ปัจจุบันมีตัวกำเนิด DIMF ที่อยู่ในรูปของ IC

ในระยะต่อมานักทดลองต้องรับมือกับ การถอดรหัสที่ไม่ค่อยแน่นอนโดยใช้ IC Phase lock loop เช่น 567 โดยใช้วงจรหลายชุดเพื่อ Detect แต่ละความถี่ ต้องปรับแต่ง Timing ด้วยอุปกรณ์ภายนอกอย่างแม่นยำ, ต้องใช้วงจรลอจิคเพื่อเปลี่ยน Output ให้อยู่ในรูปแบบที่ต้องการ และยังต้องใช้วงจรฟิลเตอร์ที่ Input ของ Tone detect แต่ละตัว ซึ่งทำให้วงจรซับซ้อนยิ่งขึ้น

ปัจจุบันมี IC ถอดรหัส DIMF ซึ่งมีความเชื่อถือได้สูงมาก ผลิตขึ้นมาโดยผู้ผลิตหลายบริษัท

ถึงแม้ว่าระบบ DTMF ถูกออกแบบขึ้นมาเพื่อการส่งเลขหมายโทรศัพท์ แต่มันก็ยังมีประโยชน์มากในระบบงานควบคุมและสอบถามข้อมูลระยะไกล ตัวอย่างการใช้งานในปัจจุบัน เช่น TELEBANKING และ PAGING

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 MT8870 DTMF Receiver

ไอซีใช้ถอดรหัสความถี่โทรศัพท์ชนิดกดปุ่ม (DTMF Receiver)ให้เป็นตัวเลข BCD ขนาด 4 bits โดยใช้งานร่วมกับคริสตอล 3.579 MHz

เมื่อเอ่ยถึงไอซีทางโทรศัพท์ ช่างหรือวิศวกรส่วนใหญ่มักจะส่ายหน้าสาเหตุก็เนื่องมาจากบ้านเรา ไม่ค่อยมีผู้นำเข้ามาจำหน่าย หายาก คู่มือไม่มี ขาดการเผยแพร่ ทั้งๆที่ในปัจจุบันโทรศัพท์มีส่วนสำคัญกับชีวิตประจำวันของเราอย่างมากมาย และนับวันก็จะยิ่งทวีความสำคัญเพิ่มมากขึ้นทุกที อุตสาหกรรมทางด้านโทรศัพท์ และการสื่อสารก็ขยายตัวขึ้นอย่างรวดเร็ว ดังนั้นช่างหรือวิศวกรอย่างพวกเราก็น่าจะมาศึกษาเอาไว้ เพื่อจะได้นำมาใช้งานได้ในโอกาสต่อไป

ก่อนอื่นขอให้ความหมายของคำว่า ถอดรหัสความถี่โทรศัพท์ อันหมายถึง การแปลงสัญญาณความถี่ซึ่งเกิดจากการกดปุ่มตัวเลขของโทรศัพท์ชนิดกดปุ่ม (ชนิด Tone หรือ DTMF) ให้เป็นระบบตัว เลขทางดิจิตอล ซึ่งไอซี MT8870 ใช้แปลงความถี่โทรศัพท์ให้เป็นเลขฐานสองขนาด 4 bit

ในยุคก่อน การออกแบบวงจรถอดรหัสความถี่ของโทรศัพท์ มักใช้ไอซีจำพวกเฟสล็อกลูปซึ่งสร้างปัญหาสารพัด ไม่ว่าเรื่องของความถี่ที่เปลี่ยนแปลงไป การปรับแต่งวงจร ขนาดของวงจรที่ใหญ่ เพราะต้องใช้ไอซีจำนวนมาก

คุณสมบัติของ MT8870

-เป็นตัวรับและถอดรหัสความถี่(DTMF8870 Receiver)

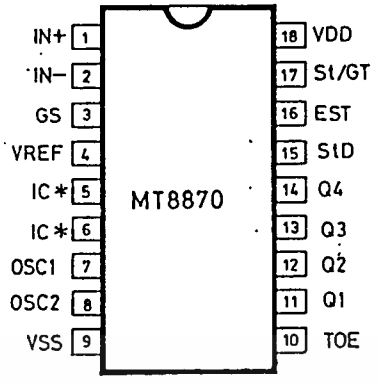
-กินไฟน้อย ใช้ไฟเลี้ยงระดับเดียวกับ TTL

-สามารถตั้งอัตราขยายภายในตัวไอซีได้

-สามารถปรับการ์ดไทม์(Guard time)

-เป็นไอซีคุณภาพสูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



* ต่อกับ VSS

MT8870BE 18 PIN PLASTIC
MT8870BC 18 PIN CERDIP

รูปที่ 2.3.1 แสดงรายละเอียดขาของ MT8870

การนำ MT8870 ไปใช้งาน

-นำไปใช้งานด้านรีโมตคอนโทรล

-เครื่องป้องกันโทรศัพท์ทางไกล

-ใช้ในงานเกี่ยวกับเครดิตการ์ด

-ใช้งานร่วมกับคอมพิวเตอร์

-ใช้ในเครื่องชุมสายขนาดย่อย หรือ PABX

-ใช้กับงานทางด้านโทรศัพท์ทั่วไป

-เครื่องกันขโมย

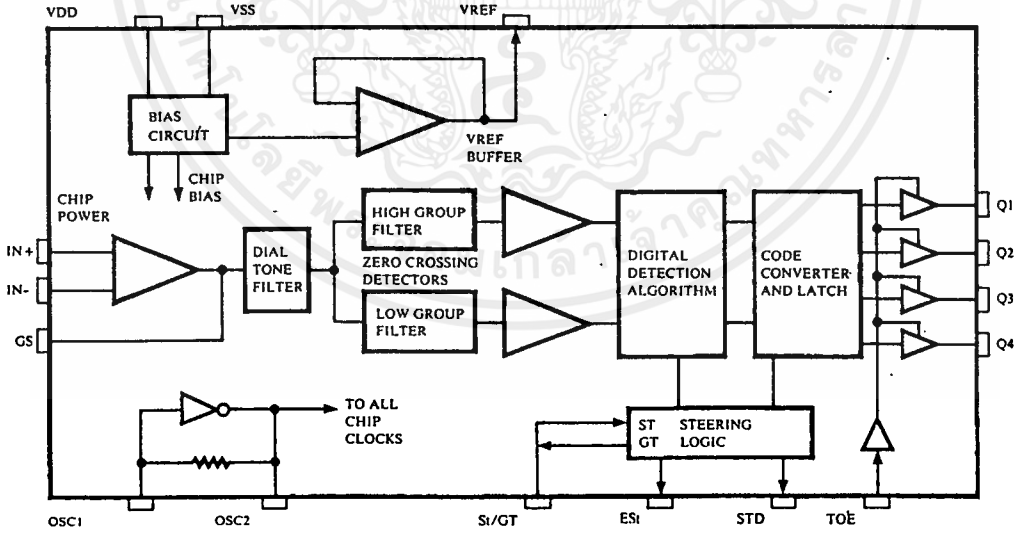
-การควบคุมอุปกรณ์ทางโทรศัพท์

-ใช้ทำเครื่องสอบถามทางโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงสร้างของ MT8870

โครงสร้างภายในของ MT8870 ประกอบไปด้วยวงจรกรองความถี่และวงจรถอดรหัสฟังก์ชันทางดิจิตอล เป็นไอซีที่สร้างโดยใช้เทคโนโลยี ISO^2_CMOS ในส่วนของวงจรกรองความถี่ใช้เทคนิคของสวิตช์คาปาซิเตอร์ฟิลเตอร์ สำหรับกรองความถี่สูงและต่ำ ส่วนวงจรถอดรหัสใช้เทคนิคการนับทางดิจิตอลเพื่อตรวจจับและถอดรหัสทั้ง 16 ความถี่ออกเป็นเลขฐานสองขนาด 4 bit และเช็คช่วงเวลาที่สัญญาณเข้ามา ส่วนภาค Input เป็นออปแอมป์ซึ่งสามารถปรับอัตราขยายได้โดยต่ออุปกรณ์ภายนอก Output เป็นวงจรแลตช์ 3 สถานะ รูปที่ 2.3.1 แสดงขาของ MT8870 และรูปที่ 2.3.2 แสดงโครงสร้างภายในของ MT8870



รูปที่ 2.3.2 แสดงโครงสร้างภายในของ MT8870

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถาบันวิทยบริการ จุฬาลงกรณ์มหาวิทยาลัย

ฟังก์ชันการทำงานภายใน MT8870

ภายใน MT8870 ประกอบด้วยส่วนสำคัญ 5 ส่วนคือ

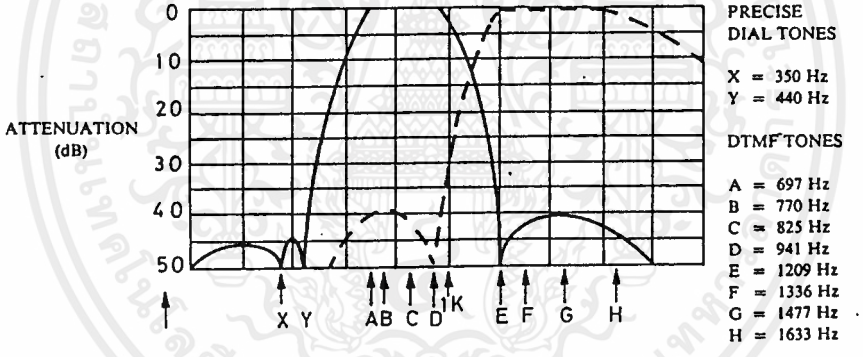
ภาคกรองความถี่(Filter section)

ภาคถอดรหัส(Decoder section)

ภาคตรวจสอบสัญญาณ(Steering circuit)

ภาคขยายสัญญาณความแตกต่าง(Differential input)

ภาคกำเนิดความถี่(Oscillator)



รูปที่ 2.3.3 แสดงความถี่ที่ได้จากภาคกรองความถี่

ภาคกรองสัญญาณความถี่

ในส่วนนี้จะแยกสัญญาณ DTMF ที่เข้ามาออกเป็น 2 กลุ่มความถี่ คือช่วงความถี่สูงและช่วงความถี่ต่ำ โดยใช้วงจรกรองแถบความถี่อันดับ 6 ชนิด สวิตช์คาปาซิเตอร์ (Six-order Switched capacitor band pass filter) ซึ่งความถี่ที่แยกได้มี 2 ช่วง คือช่วงความถี่สูงและความถี่ต่ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
033325

ภาคถอดรหัส

ความถี่ DTMF ที่ถูกกรองเรียบร้อยแล้วจะผ่านเข้าวงจรถอดรหัสความถี่ออกเป็นตัวเลข โดยใช้เทคนิคการการนับแบบดิจิตอล และมีการตรวจสอบความถี่ที่เข้า

F(low)	F(high)	NO	TOE	Q4	Q3	Q2	Q1
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1336	0	H	1	0	1	0
941	1209	*	H	1	0	1	1
941	1477	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0
-	-	ANY	L	Z	Z	Z	Z

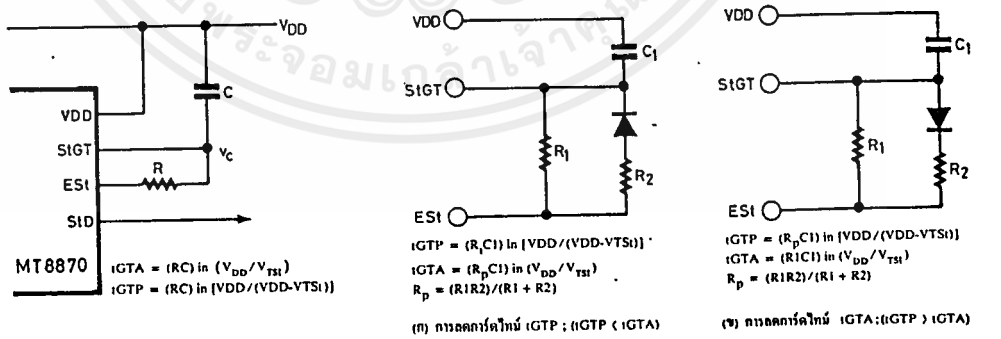
รูปที่ 2.3.4 แสดงค่าที่ถอดรหัสได้จากความถี่ต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาว่าเป็นความถี่มาตรฐาน DTMF หรือไม่ เพื่อป้องกันความถี่อื่นเข้ามาผสม เมื่อตรวจสอบว่าความถี่นั้นถูกต้อง สัญญาณที่ขา ESt (early steering) ก็จะแอคทีฟ สำหรับค่าที่ถอดรหัสได้จากความถี่ต่างๆ นั้น แสดงในรูปที่ 2.3.4

ภาคตรวจสอบสัญญาณ

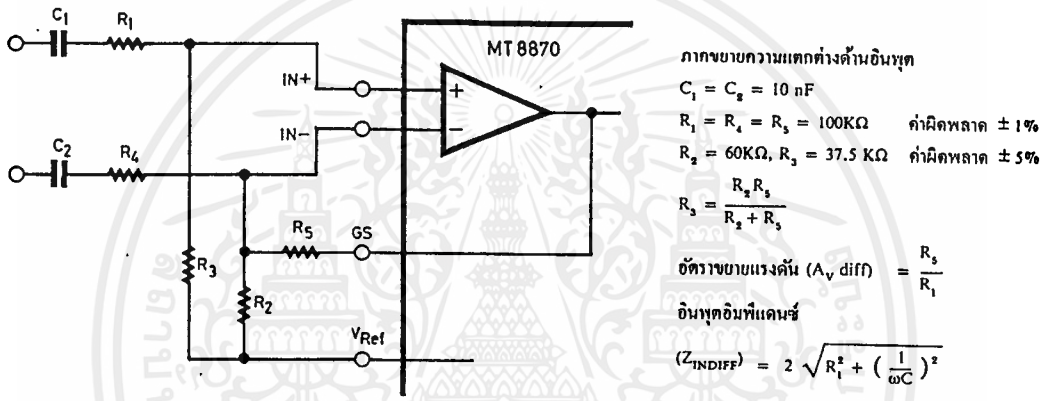
ก่อนที่จะมีการถอดรหัสความถี่ ออกไปที่ Output จะมีการตรวจสอบช่วงความถี่ที่เข้ามาว่ามีระยะเวลาตามที่กำหนดหรือไม่ โดยสังเกตจากระยะเวลาการกดปุ่มโทรศัพท์ ซึ่งต้องกดปุ่มให้มีความถี่ออกมาเป็นช่วงเวลาพอสมควรมิฉะนั้นวงจรส่วนนี้จะไม่รับ โดยถือว่าสัญญาณนั้นไม่ถูกต้อง ส่วนช่วงเวลายาวเท่าใดสามารถตั้งได้โดยใช้ RC ต่อภายนอก สัญญาณที่ขา ESt จะเป็น "High" นานใกล้เคียงกับระยะเวลาที่มีความถี่ DTMF เข้ามา จากรูปที่ 5 เมื่อขา ESt เป็น "High" ทำให้ V_c สูงขึ้น ตัวเก็บประจุ C จะคายประจุทำให้แรงดัน V_c สูงขึ้นจนถึงค่าเทรชโฮลด์ วงจรถอดรหัสจึงจะถอดรหัสออกเป็นตัวเลข BCD ขนาด 4 bit รายละเอียดการทำงาน ดูจากแผนภูมิเวลา หรือไทมิ่งไดอะแกรม (timing diagram) ในรูปที่ 9 จะเข้าใจได้ง่ายกว่า



รูปที่ 2.3.5 แสดงวงจรตรวจสอบสัญญาณอย่างง่าย และแสดงการกำหนดเวลาการ์ดไทม์(Gard time) พร้อมวิธีคำนวณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับคำว่าการ์ดไทม์ (guard time) นั้นหมายถึง ช่วงคาบเวลาของความถี่ที่เข้ามา ซึ่งจะต้องนานเท่ากับหรือมากกว่าช่วงเวลาที่เราตั้งไว้ จึงจะได้รับการยอมรับว่าสัญญาณความถี่นั้นถูกต้อง หรือพูดได้ว่าเวลาที่เราตั้งไว้โดย RC ก็คือการ์ดไทม์นั้นเอง เมื่อสัญญาณความถี่เข้ามานานเท่าหรือมากกว่าเวลาที่ตั้งไว้ จึงจะสามารถแปลงเป็นตัวเลขได้ ถ้าสัญญาณความถี่เข้ามาสั้นกว่า ก็จะไม่มีการถอดรหัสเป็นตัวเลขออกไป การตั้งเวลา และคำนวณเวลาดูได้จากรูปที่ 5



รูปที่ 2.3.6 แสดงการต่อวงจรภาค INPUT

ภาคขยายสัญญาณความแตกต่าง

วงจรส่วนอินพุตของ MT8870 เป็นภาคขยายออปแอมป์ที่สามารถปรับอัตราขยายโดยต่อวงจรภายนอกเพิ่มเข้าไปรูปที่ 6 แสดงการต่อวงจรภายนอกเข้ากับอินพุตซึ่งสามารถคำนวณอัตราขยายความแตกต่างของอินพุตและอิมพีแดนซ์ได้ดังนี้

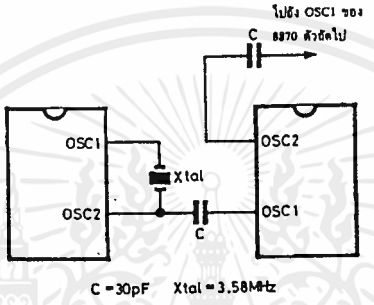
$$\text{อัตราขยาย } (A_v diff) = R_5/R_1$$

$$\text{อินพุตอิมพีแดนซ์ } (Z_{in} diff) = 2\sqrt{R_1^2 + (1/wC)^2}$$

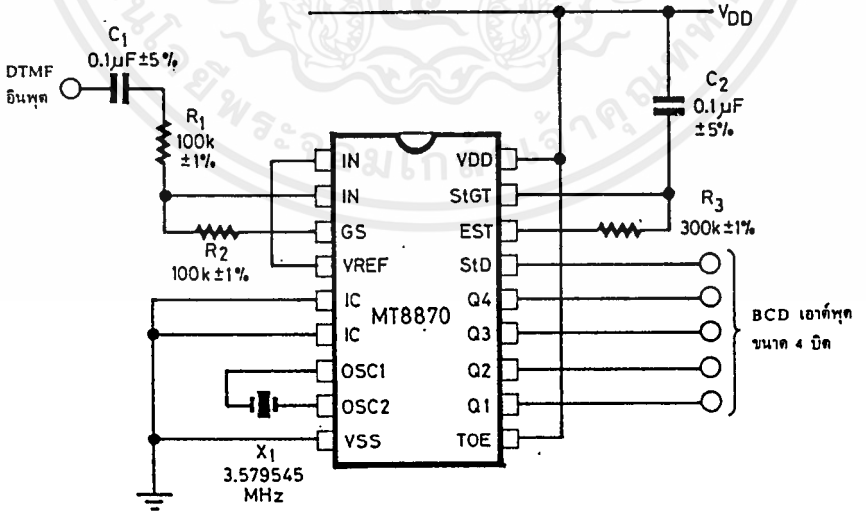
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคกำเนิดความถี่

จะมีวงจรเวลาอยู่ภายใน เพียงแต่ต่อแร่คริสตอลขนาด 3.58 MHz ก็สามารถใช้งานได้ทันที การต่อวงจรกำเนิดความถี่แสดงในรูปที่ 2.3.7



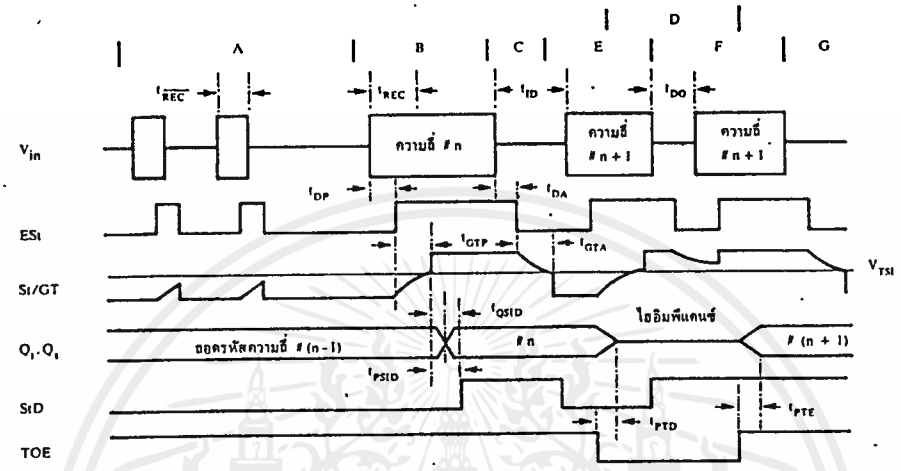
รูปที่ 2.3.7 แสดงการต่อวงจรผลิตความถี่



รูปที่ 2.3.8 แสดงวงจรใช้งานเบื้องต้นของ MT8870

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทำงาน



รูปที่ 2.3.9 แสดงแผนภูมิเวลา (timing diagram) ของ MT8870

อธิบายขั้นตอนการทำงาน

A – ตรวจพบความถี่เข้ามา แต่คาบเวลาไม่ถูกต้อง เอาต์พุตไม่เปลี่ยน

B – ความถี่ # n ถูกตรวจพบและมีคาบเวลาที่ถูกต้อง ความถี่ถูกถอดรหัสและแลตช์ไว้ที่เอาต์พุต

C – จบความถี่ # n ช่วงห่างถูกต้อง เอาตพุตยังคงแลตช์อยู่จนกว่าจะได้รับความถี่ที่ถูกต้องใหม่

D – เอาต์พุตเปลี่ยนเป็นไฮอิมพีแดนซ์

E – ความถี่ # n + 1 ถูกตรวจพบ คาบเวลาถูกต้อง ความถี่ถูกถอดรหัสและแลตช์ไว้

F – ความถี่ # n + 1 หายไป ช่วงห่างไม่ถูกต้อง เอาต์พุตยังคงแลตช์อยู่

C – จบความถี่ # n + 1 ช่วงห่างถูกต้อง เอาต์พุตยังแลตช์อยู่ จนถึงความถี่ใหม่ที่ถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายคำศัพท์ ในรูป 2.3.9

V_{in} - สัญญาณความถี่ DIMF ที่เข้ามา

ESt - Early Sterring output ใช้แสดงความถี่ที่ถูกต้อง

St/GT - Steering input/Guard Time output สำหรับต่อกับ RC ภายนอก

Q_1-Q_4 - เอาต์พุต BCD ขนาด 4 บิต

StD - Delayed Steering output ใช้แสดงว่าความถี่ที่ได้รับหรือหายไป มีคาบเวลาตามที่กำหนด เพื่อแสดงความถูกต้องของสัญญาณ

TOE - Tone Output Enable (input)ใช้ควบคุม Q_1-Q_4 ให้เป็นไฮอิมพีแดนซ์

t_{REC} - คาบเวลานานสุดที่ตรวจพบความถี่ DIMF แล้วยังไม่ถูกต้อง

t_{REC} - คาบเวลาสั้นสุดที่ต้องการเพื่อแสดงว่าสัญญาณถูกต้อง

t_{ID} - เวลาสั้นสุดระหว่างสัญญาณ DIMF ที่ถูกต้อง 2 สัญญาณ

t_{DO} - เวลานานสุดที่ยอมให้สัญญาณหายไปได้ในคาบเวลาความถี่ที่ถูกต้อง

t_{DP} - เวลาที่ใช้ในการตรวจพบสัญญาณความถี่ DIMF ที่ถูกต้อง

t_{DA} - เวลาที่ใช้ในการตรวจการหายไปของสัญญาณความถี่ DIMF ที่ถูกต้อง

t_{GTP} - การ์ดไทม์ของการปรากฏความถี่ DIMF

t_{GTA} - การ์ดไทม์ของการหายไปของความถี่ DIMF

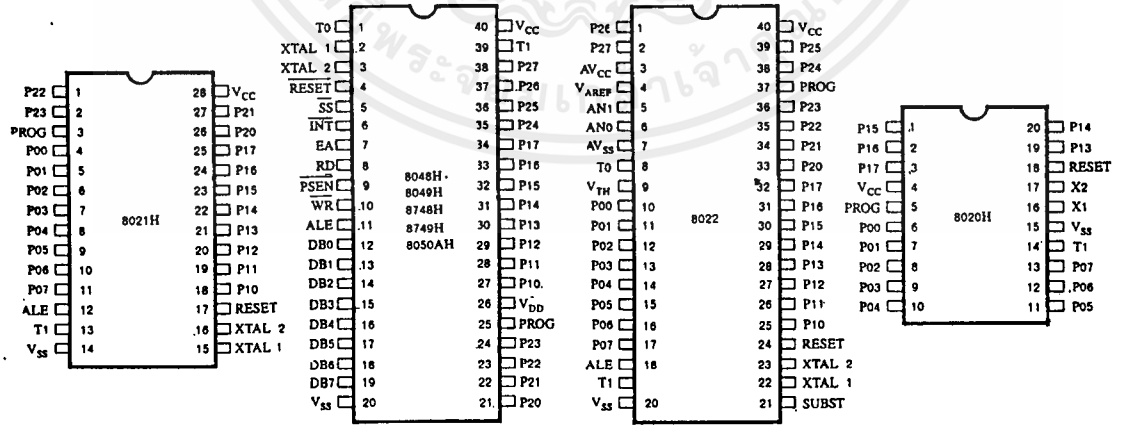
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ไมโครคอมพิวเตอร์ชิพเดี่ยว 8048 Single Chip Microcomputer

ความจริงแล้ว ซีพียูเบอร์ 8050/8749H/8049/8748/8048/8022/8021 หรือ 8020 มีลักษณะแตกต่างจากไมโครโปรเซสเซอร์ กล่าวคือมีลักษณะใกล้เคียงระบบไมโครคอมพิวเตอร์ ชิพประเภทนี้มี เซนทรัลโปรเซสเซอร์ วงจรสร้างสัญญาณคล็อก หน่วยความจำ และ อินพุท/เอาท์พุท อยู่ภายในครบถ้วน จึงเหมาะสำหรับงานควบคุมขนาดเล็ก ดังนั้นจึงเป็นที่นิยมใช้กันอย่างแพร่หลายแต่เราไม่ค่อยจะได้ยินข่าวมากนัก เพราะส่วนใหญ่ จะนำไปใช้ในงานควบคุม เช่น เครื่องซักผ้า เครื่องถ่ายเอกสาร และ โทรศัพท์ เป็นต้น

ลักษณะของซีพียู

ลักษณะของซีพียูจากโครงสร้างเป็นสาร NMOS ภายนอกเป็นไอซีขนาด 40 ขา ดังเบอร์ต่อไปนี้ 8048/8049/8748/8749/8050/8022 ส่วนเบอร์ 8021 เป็นขนาด 28 ขา และเบอร์ 8020 มีจำนวน 20 ขา ดังตัวอย่างแสดงในรูปที่ 2.4.1

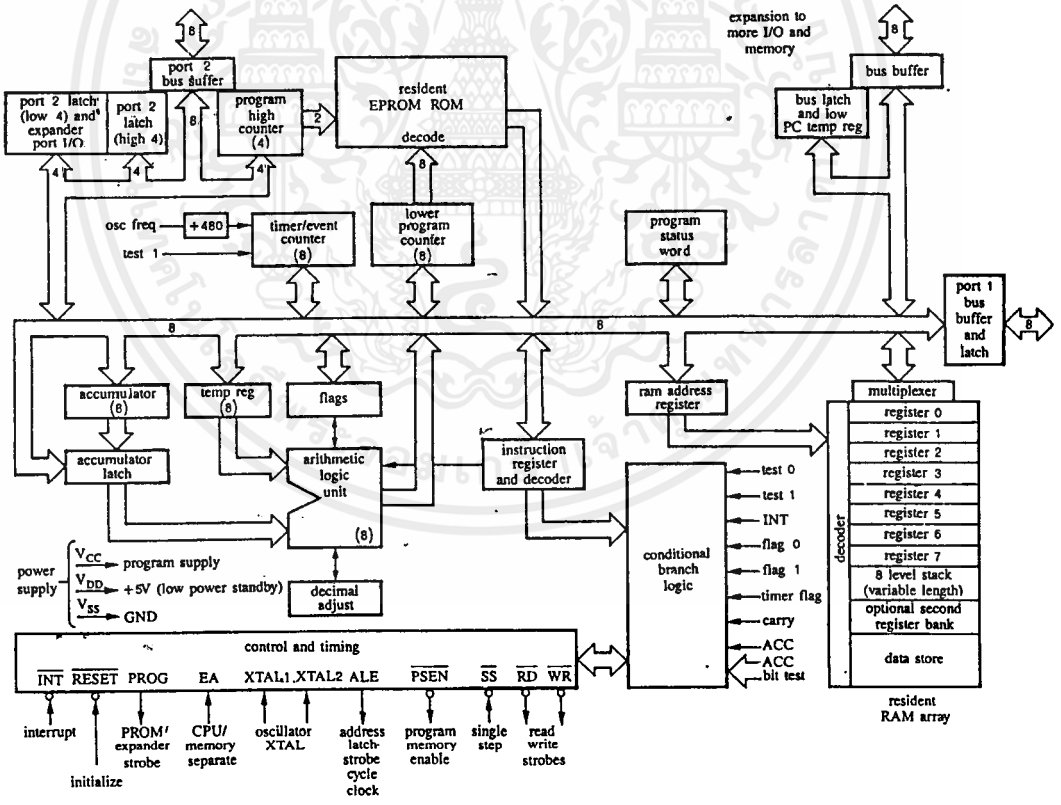


รูปที่ 2.4.1 ซีพียูเบอร์ต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	8050H	8749H	8049H	8748	8048H	8022	8021H	8020H
8-Bit CPU	✓	✓	✓	✓	✓	✓	✓	✓
Program Memory	4K×8 ROM	2K×8 EPROM	2K×8 EPROM	1K×8 EPROM	1K×8 ROM	2K×8 ROM	1K×8 ROM	1K×8 ROM
Data Memory	256×8 RAM	128×8 RAM	128×8 RAM	64×8 RAM	64×8 RAM	64×8 RAM	64×8 RAM	64×8 RAM
I/O Lines	27	27	27	27	27	28	21	13
8-Bit Timer/Counter	✓	✓	✓	✓	✓	✓	✓	✓
Reset	✓	✓	✓	✓	✓	✓	✓	✓
Interrupt	✓	✓	✓	✓	✓	✓	✓	✓
Oscillator and Clock	✓	✓	✓	✓	✓	✓	✓	✓

ตารางที่ 1 ลักษณะสมบัติของซีพียูเบอร์ต่าง ๆ

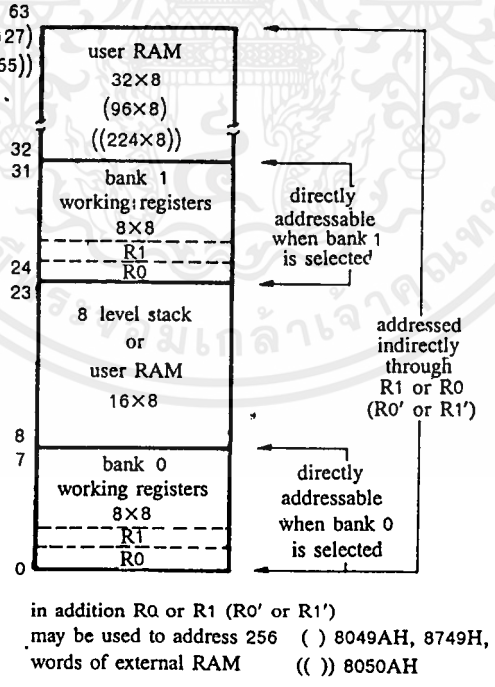


รูปที่ 2.4.2 สถาปัตยกรรมของซีพียู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์

รีจิสเตอร์แท้ ๆ ของซีพียูมีอยู่เพียง 3 ตัว คือ แอกคูมิวเลเตอร์ (accumulator) แฟล็ก (flag) ซึ่งเป็นรีจิสเตอร์ขนาด 8 บิท และโปรแกรมเคาน์เตอร์ (program counter) ขนาด 12 บิทแต่ในขณะเดียวกันบริษัท intel ซึ่งเป็นผู้ผลิตไอซีสามารถจัดรีจิสเตอร์ได้ถึง 16 ตัว โดยใช้หน่วยความจำชนิด RAM ที่มีอยู่แล้ว รีจิสเตอร์ทั้ง 16 ตัวนี้ แบ่งออกเป็น 2 แบงก์ (bank) คือ แบงก์ 0 และแบงก์ 1 แบงก์ 0 จะประกอบด้วยรีจิสเตอร์ R_0 ถึง R_7 ในการใช้งานเราจะใช้ได้ครั้งละ 1 แบงก์เท่านั้น ส่วนแบงก์ 1 จะใช้ในโปรแกรมย่อยที่ตอบสนองการอินเตอร์รัพท์ เพราะซีพียูไม่เก็บค่าของรีจิสเตอร์ เมื่อถูกอินเตอร์รัพท์ และไม่มีคำสั่งให้เราเก็บด้วย รีจิสเตอร์ R_0 ถึง R_7 จะทำหน้าที่เป็นรีจิสเตอร์ทั่วไป และเป็นตัวเคาน์เตอร์ได้ทุกตัว เทียบได้กับรีจิสเตอร์ B ในซีพียู Z-80 ยกเว้น R_0 และ R_1 ซึ่งทำหน้าที่พิเศษเพิ่มขึ้นคือ ทำหน้าที่ชี้หน่วยความจำเช่นเดียวกับ HL ในซีพียู Z-80



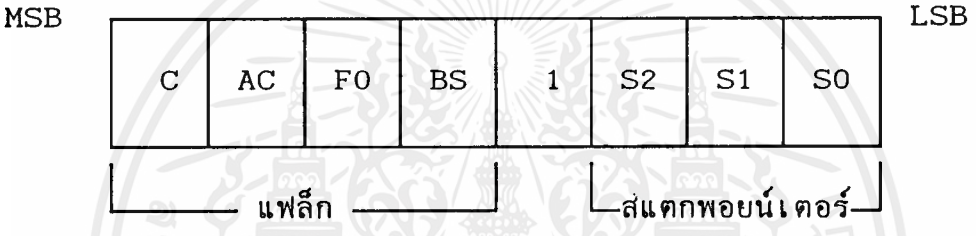
รูปที่ 2.4.3 ผังหน่วยความจำดาต้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แฟล็กและสแตกพอยน์เตอร์

สำหรับแฟล็กของซีพียู มีอยู่ 4 แฟล็ก ส่วนที่เหลือคือ สแตกพอยน์เตอร์ (stack pointer) จากตัวอย่างต่อไปนี้ จะเป็นการบ่งบอกตำแหน่งของแฟล็กและสแตกพอยน์เตอร์ ซึ่งแต่ละส่วนก็มีหลักและการทำงานดังนี้

แฟล็กและสแตกพอยน์เตอร์ (flag & stack pointer)



carry flag (C) เมื่อมีการทดจากการบวกหรือการลบจากการยืมที่ MSB (บิต 7) แฟล็กนี้จะมีลอจิกเป็น "1"

auxiliary carry (AC) เป็นแฟล็กตามแบบเหมือนกับ carry flag แต่แสดงผลของการดำเนินกรรมวิธีใน 4 บิตต่ำ คือทดครึ่ง (Half carry) จากบิต 3 ไปบิต 4

bank select (BS) จะแสดงว่าขณะนี้เลือกใช้วิธีรีจิสเตอร์ในแบงก์ใดอยู่ ถ้าใช้แบงก์ 0 ก็จะถูกรีเซต แต่ถ้าใช้แบงก์ 1 ก็จะเซต

F_0 คือแฟล็ก 0 เป็นแฟล็กของผู้ใช้ คือผู้เขียนโปรแกรมสามารถเซตและรีเซตแฟล็กนี้ เพื่อกำหนดเงื่อนไขในการกระโดดได้

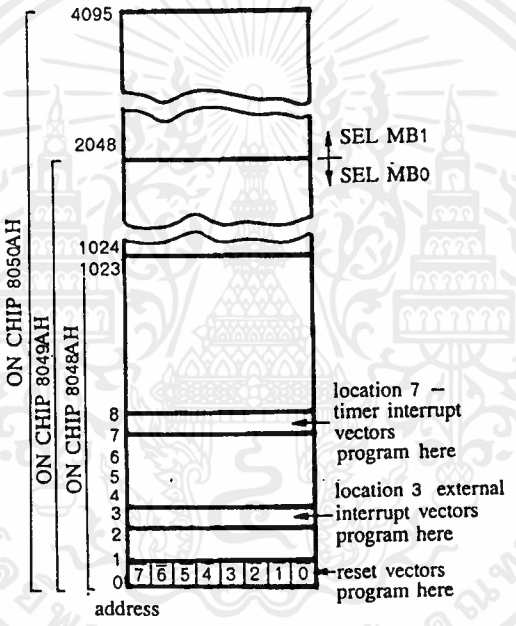
F_1 คือแฟล็ก 1 ทำหน้าที่เช่นเดียวกับ แฟล็ก 0

นอกจากนี้ยังมีรีจิสเตอร์ขนาด 4 บิต อีก 4 บิตเป็นไทม์เมอร์เคาน์เตอร์ (timer counter) อยู่ในวงจรนับ และฐานเวลากับรีจิสเตอร์ของพอร์ตทั้งสาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผังโปรแกรมเมโมรี (program memory map)

จากตารางคุณสมบัติและลักษณะแล้วจะพบว่า ซีพียูเบอร์ 8748/8749/8049 พัฒนาโปรแกรมให้เป็นแบบ EPROM ส่วนเบอร์ 8050/8048/8022/8021/8020 โปรแกรมเมโมรีจะเป็น ROM ดังแสดงในรูปที่ 2.4.4

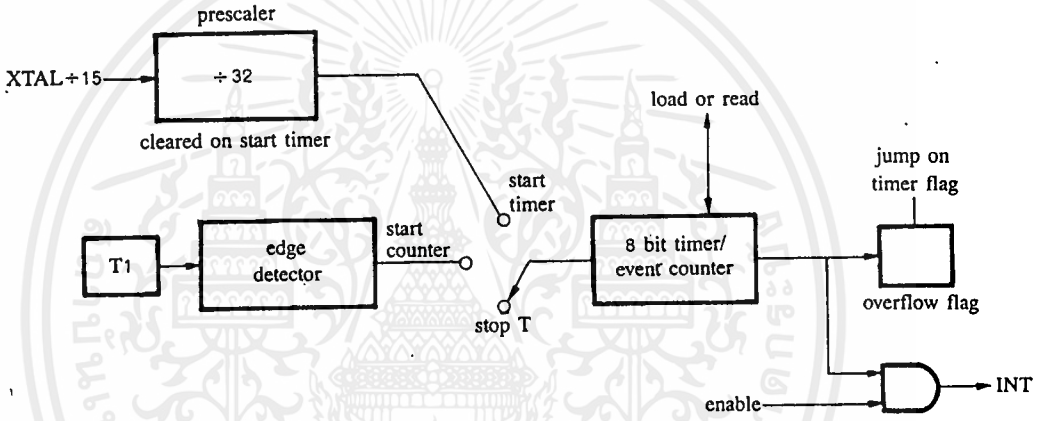


รูปที่ 2.4.4 ผังโปรแกรมเมโมรี

นอกจากนี้แล้วซีพียูยังมีวงจรนับ และฐานเวลาขนาด 8 บิตอยู่อีก 1 วงจร เป็นวงจรนับและฐานเวลาที่น่าใช้มากตัวหนึ่ง การใช้งานก็ง่าย เมื่อใช้เป็นวงจรฐานเวลาจะนำสัญญาณคล็อคซึ่งส่วนใหญ่จะเป็น 6 เมกะเฮิรตซ์ สัญญาณคล็อคที่ส่งมาจะถูกหารด้วย 15 และหารด้วย 32 เสียก่อนจากนั้นจึงส่งเข้าวงจรนับขนาด 8 บิต ซึ่งจะเพิ่มหนึ่งค่าในตัวนับ และจะนับไปมากที่สุด คือ 256 โดยเริ่มจาก 0 และจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเตอร์รัพท์ซีพียู 1 ครั้งและจะอินเตอร์รัพท์ทุก 80 ไมโครวินาทีถึง 20 มิลลิวินาที ตามค่าที่เรากำหนด ในวงจรนับหากเราใช้เป็นวงจรนับครึ่ง ให้ต่อสัญญาณคล็อคภายนอกเข้ากับขา T_1 ส่วนขา T_0 และ T_1 ทำหน้าที่พิเศษ ดังได้กล่าวมาแล้ว ขา T_0 จากการที่เราใช้คริสตอล 6 เมกะเฮิรตซ์ ป้อนเข้าไปในซีพียู และจะถูกหาร 3 ลงไปอีก (6 หาร 3) เหลือ 2 เมกะเฮิรตซ์ ที่นอกเหนือจากนี้สามารถใช้คำสั่งให้ส่งออกทางขา T_0 ได้



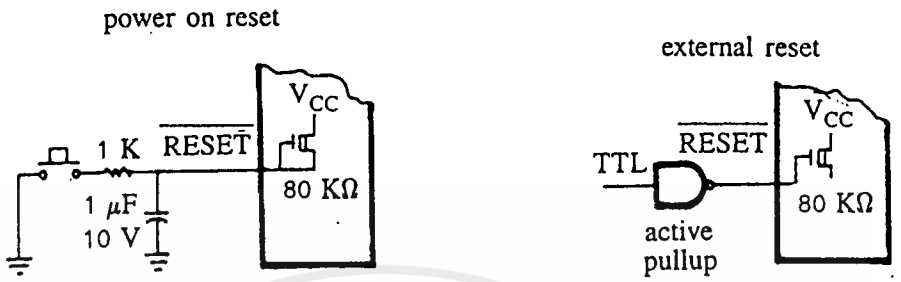
รูปที่ 2.4.5 timer/event counter

การรีเซตและอินเตอร์รัพท์

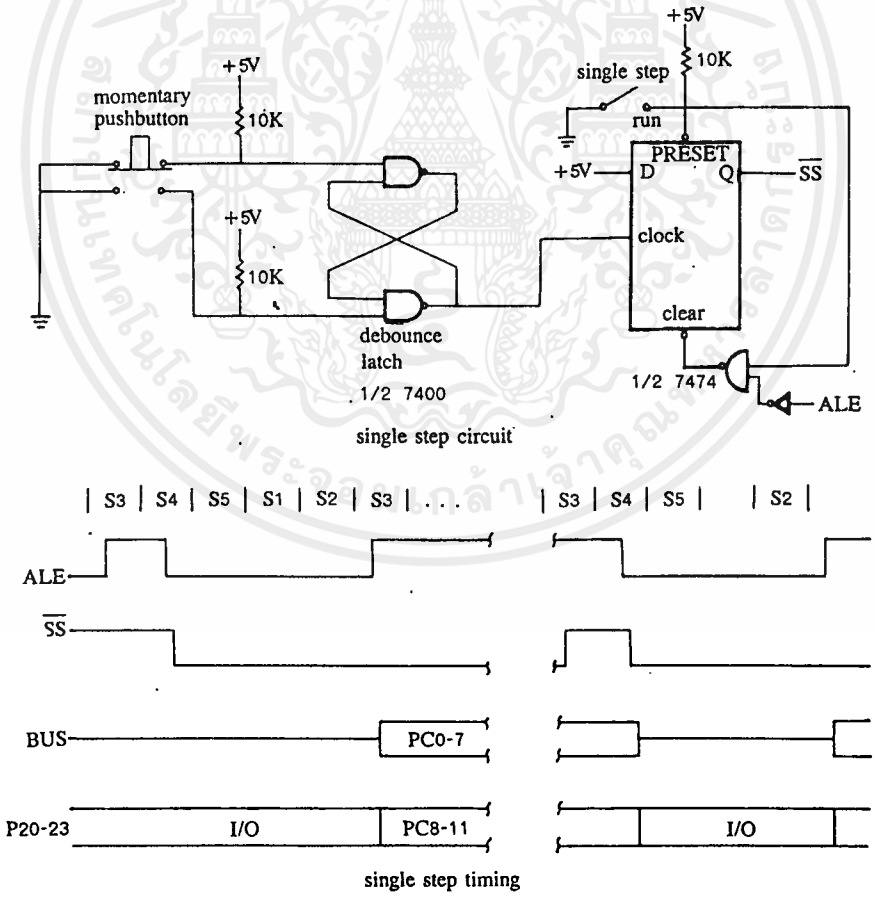
การรีเซตของซีพียูจะเริ่มที่ 000H แต่ถ้าหากต้องการอินเตอร์รัพท์ที่ขา INT ของซีพียู จะกระโดดมายัง 003H แต่ถ้าหากเราใช้ TCNT จะกระโดดไปยังต้นโปรแกรมที่ตอบสนองการอินเตอร์รัพท์ 007H

นอกจากนี้แล้วซีพียูยังมีขา SS (single step) ซึ่งถ้าเราให้ลอจิก "1" แก่ขานี้จะทำให้ซีพียูหยุดเพื่อปฏิบัติตามคำสั่งที่ค้างอยู่จนจบและรัน แต่เมื่อเราให้ลอจิก "0" นั่นคือ เราสามารถจัดวงจรให้ซีพียูทำงานทีละคำสั่งได้ ซึ่งเป็นประโยชน์อย่างมากในการตรวจสอบแก้โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4.6 การรีเซต



รูปที่ 2.4.7 single step operation

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดเกี่ยวกับขาต่าง ๆ

V_{SS} คือขาที่ 20 ทำหน้าที่เป็นกราวด์ให้แก่วงจร

V_{DD} คือขาที่ 26 ทำหน้าที่โปรแกรมเพาเวอร์ซัพพลาย +25 โวลต์

V_{CC} คือขาที่ 40 ทำหน้าที่ป้อนไฟ +5 โวลต์ให้แก่หน่วยความจำ (CPU)

PROG คือขาที่ 25 เป็นอินพุตโปรแกรมพัลส์ +23 โวลต์

P_{10}-P_{17} (พอร์ต 1) คือขาที่ 27-34 ทำหน้าที่เป็นพอร์ตได้ทั้งอินพุตและเอาต์พุต

P_{20}-P_{27} (พอร์ต 2) คือขาที่ 21-24 และ 35-38 ทำหน้าที่ใช้เป็นพอร์ตได้ทั้งอินพุตและเอาต์พุต

D_0-D_7 (บัส) คือขาที่ 12-19 ทำหน้าที่เป็นอินพุตและเอาต์พุตได้ โดยจะส่งข้อมูลเข้าและออก นอกจากนั้นยังสามารถทำเป็นแอดเดรสบัส (address bus) ในตัวได้ ในกรณีที่ใช้หน่วยความจำภายนอก

To คือขาที่ 1 เป็นขาอินพุต สามารถใช้ขานี้ให้เป็นเงื่อนไขการตัดสินใจได้ เช่น JTO หรือ JNTO และยังสามารถออกแบบทำให้เป็นขาเอาต์พุตได้ เมื่อใช้ ENTO CLK T1 คือขาที่ 29 เป็นขาอินพุต ทำหน้าที่รับสัญญาณคล็อคเข้าไปเพื่อการนับฐานเวลาได้ โดยใช้ STRT CNT และในขณะเดียวกัน ทำให้เป็นเงื่อนไขการตัดสินใจได้ เช่น JT1 หรือ JNT1

INT คือขาที่ 6 เป็นขาอินพุตแอกทีฟโลว์ คือต้องป้อนลอจิก "0" จะเกิดอินเตอร์รัพท์ทันที

RD คือขาที่ 8 เป็นขาเอาต์พุตแอกทีฟโลว์ ใช้ในกรณีที่ต้องการจะอ่านข้อมูลจากหน่วยความจำภายนอก ขานี้จึงจะส่งสัญญาณออกไป

RESET คือขาที่ 4 เป็นขาอินพุตแอกทีฟโลว์ ใช้กรณีที่ต้องการให้ซีพียูรีเซตตัวเอง

WR คือขาที่ 10 เป็นขาเอาต์พุตแอกทีฟโลว์ ใช้กรณีที่ต้องการจะเขียนข้อมูลจากหน่วยความจำภายนอก

ALE คือขาที่ 11 เป็นขาเอาต์พุตแอกทีฟโลว์ ใช้ทำการกระตุ้นเพื่ออนุญาตให้ส่งค่าแอดเดรสบัส (address bus)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

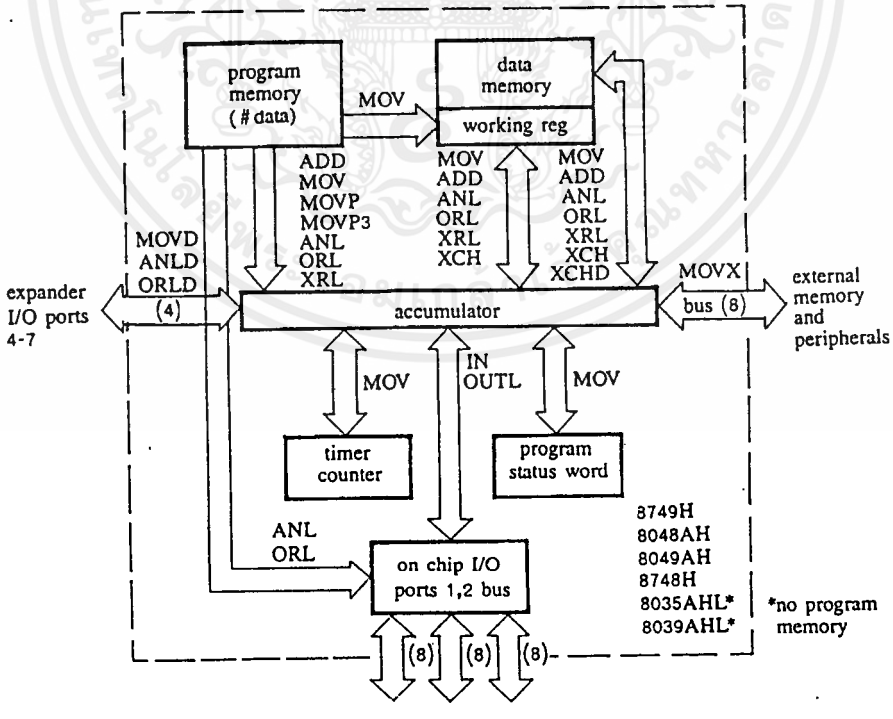
PSEN คือขาที่ 9 เป็นขาเอาต์พุตแอกทีฟโลว์ ใช้ในการ program store enable ซึ่งกรณีที่ซีพียูต้องการรับคำสั่งจากหน่วยความจำภายนอก

SS คือขาที่ 5 เป็นขาอินพุตแอกทีฟโลว์ ใช้ทำหน้าที่ตรวจสอบโปรแกรมในขั้นแรก

EA คือขาที่ 7 เป็นขาอินพุตแอกทีฟโลว์ กรณีต้องการโปรแกรมให้แก่ซีพียูเบอร์ 8748/8749 ต้องป้อนไฟ +18 โวลต์ แต่ถ้าเป็นซีพียูเบอร์ 8048/8049/8035/8039/8040 ต้องป้อน interal PULLUP ประมาณ 10

XTAL1 คือขาที่ 2 เป็นขาอินพุตข้างหนึ่งของผลึกคริสตอล

XTAL2 คือขาที่ 3 เป็นขาอินพุตอีกข้างหนึ่งของผลึกคริสตอล



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางคำสั่ง

MCS -48 INSTRUCTION SET

Symbols and Abbreviations Used

A	accumulator
AC	auxiliary carry
addr	12-bit program memory address
Bb	bit designator (b=0-7)
BS	bank switch
BUS	BUS port
C	carry
CLK	clock
CNT	event counter
CRR	conversion result register
D	mnemonic for 4-bit digit (nibble)
data	8-bit number or expression
DBF	memory bank flip-flop
FO,F1	flag 0, flag 1
I	interrupt
P	mnemonic for "in-page" operation
PC	program counter
P_p	port designator (p=1, 2 or 4-7)
PSW	program status word
Ri	data memory pointer (i=0, or 1)
Rr	register designator (r=0-7)
SP	stack pointer
T	timer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TF time flag

T0,T1 test 0, test 1

X mnemonic for external RAM

immediate data prefix

@ indirect address prefix

$ current value of program counter

(X) contents of X

((X)) contents of location addressed by X

is replaced by

Mnemonic	Description	Bytes	Cycle
Accumulator			
ADD A, R	Add register to A	1	1
ADD A, @R	Add data memory to A	1	1
ADD A, # data	Add immediate to A	2	2
ADDC A, R	Add register with carry	1	1
ADDC A, @R	Add data memory with carry	1	1

Mnemonic	Description	Bytes	Cycles
Registers			
INC R	Increment register	1	1
INC @R	Increment data memory	1	1
DEC R	Decrement register	1	1
Branch			
JMP addr	Jump unconditional	2	2
JMPP @A	Jump indirect	1	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mnemonic	Description	Bytes	Cycle
ADDC A, # data	Add immediate with carry	2	2
ANL A, R	And register to A	1	1
ANL A, @R	Add data memory to A	1	1
ANL A, # data	And immediate to A	2	2
ORL A, R	Or register to A	1	1
ORL A @R	Or data memory to A	1	1
ORL A, # data	Or immediate to A	2	2
XRL A, R	Exclusive Or register to A	1	1
XRL A, @R	Exclusive or data memory to A	1	1
XRL, A, # data	Exclusive or immediate to A	2	2
INC A	Increment A	1	1
DEC A	Decrement A	1	1
CLR A	Clear A	1	1
CPL A	Complement A	1	1
DA A	Decimal adjust A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1
Input/Output		1	2
IN A, P	Input port to A	1	2
OUTL P, A	Output A to port	2	2
ANL P, # data	And immediate to port	2	2
ORL P, # data	Or immediate to port	1	2
*INS A, BUS	Input BUS to A	1	2
*OUTL BUS, A	Output A to BUS	2	2
*ANL BUS, # data	And immediate to BUS	2	2
*ORL BUS, # data	Or immediate to BUS	1	2
MOVD A, P	Input Expander port to A	1	2
MOVD P, A	Output A to Expander port	1	2
ANLD P, A	And A to Expander port	1	2
ORLD P, A	Or A to Expander port		

Mnemonic	Description	Bytes	Cycle
DJNZ R, addr	Decrement register and jump	2	2
JC addr	Jump on carry = 1	2	2
JNC addr	Jump on carry = 0	2	2
JZ addr	Jump on A Zero	2	2
JNZ addr	Jump on A not Zero	2	2
JT0 addr	Jump on T0=1	2	2
JNT0 addr	Jump on T0=0	2	2
JT1 addr	Jump on T1=1	2	2
JNT1 addr	Jump on T1=0	2	2
JF0 addr	Jump on F0=1	2	2
JF1 addr	Jump on F1=1	2	2
JTF addr	Jump on timer flag = 1	2	2
JNI addr	Jump on INT = 0	2	2
JBb addr	Jump on Accumulator Bit	2	2
Subroutine			
CALL addr	Jump to subroutine	2	2
RET	Return	1	2
RETR	Return and restore status	1	2
Flags			
CLR C	Clear Carry	1	1
CPL C	Complement Carry	1	1
CLR F0	Clear Flag 0	1	1
CPL F0	Complement Flag 0	1	1
CLR F1	Clear Flag 1	1	1
CPL F1	Complement Flag 1	1	1
Data Moves			
MOV A, R	Move register to A	1	1
MOV A, @R	Move data memory to A	1	1
MOV A, # data	Move immediate to A	2	2
MOV R, A	Move A to register	1	1
MOV @R, A	Move A to data memory	1	1
MOV R, # data	Move immediate to register	2	2
MOV @R, # data	Move immediate to data memory	2	2
MOV A, PSW	Move PSW to A	1	1
MOV PSW, A	Move A to PSW	1	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mnemonic	Description	Bytes	Cycle
Data Moves (Cont' d)			
XCH A, R	Exchange A and register	1	1
XCH A, @R	Exchange A and data memory	1	1
XCHD A, @R	Exchange nibble of A and register	1	1
MOVX A, @R	Move external data memory to A	1	2
MOVX @R, A	Move A to external data memory	1	2
MOVP A, @A	Move to A from current page	1	2
MOVP3 A, @A	Move to A from Page 3	1	2
Timer/Counter			
MOV A, T	Read Timer/Counter	1	1
MOV T, A	Load Timer/Counter	1	1
STRT T	Start Timer	1	1
STRT CNT	Start Counter	1	1
STOP TCNT	Stop Timer/Counter	1	1
EN TCNTI	Enable Timer/Counter Interrupt	1	1
DIS TCNTI	Disable Timer/Counter Interrupt	1	1

Mnemonic	Description	Bytes	Cycle
Control			
EN I	Enable external Interrupt	1	1
DIS I	Disable external Interrupt	1	1
SEL RB0	Select register bank 0	1	1
SEL RB1	Select register bank 1	1	1
SEL MB0	Select memory bank 0	1	1
SEL MB1	Select memory bank 1	1	1
ENT0 CLK	Enable clock output on T0	1	1
NOP	No Operation	1	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 หลักการเชื่อมโยงสัญญาณโทรศัพท์

สัญญาณเสียงที่ส่งผ่านคู่สายโทรศัพท์เป็นสัญญาณที่ซ้อนทับกับกระแสไฟตรง การเชื่อมโยงสัญญาณโทรศัพท์ 2 คู่สายเข้าด้วยกันอาจทำอย่างง่ายๆ โดยการเชื่อมโยงโดยตรงตามรูปที่ 2.1.1 ในการเชื่อมโยงเช่นนี้จะต้องคำนึงถึง DC Voltage ที่ปรากฏบนสายทั้งสองกล่าวคือตรงกันและมีเครื่องโทรศัพท์เครื่องๆ หนึ่งยกหูอยู่เพื่อให้เกิด DC Loop ขึ้นที่ Line Circuit ภายในอุปกรณ์ชุมสายของเลขหมายทั้งสอง วงจรเชื่อมโยงที่นำมาใช้งาน ได้มาจากวงจรในตู้สลับสายโทรศัพท์ แบบพนักงานต่อแบบแบตเตอรี่ร่วมซึ่งยังมีใช้งานเป็นตู้สาขาภายใน (PABX) เนื่องจากราคาถูกมากและค่าใช้จ่ายรวมทั้งค่าบำรุงรักษาต่ำ รูปที่ 2.1.2 คือวงจรเชื่อมโยงที่กล่าวถึง

รูปที่ 2.1.1 การเชื่อมโยงสัญญาณโทรศัพท์อย่างง่าย

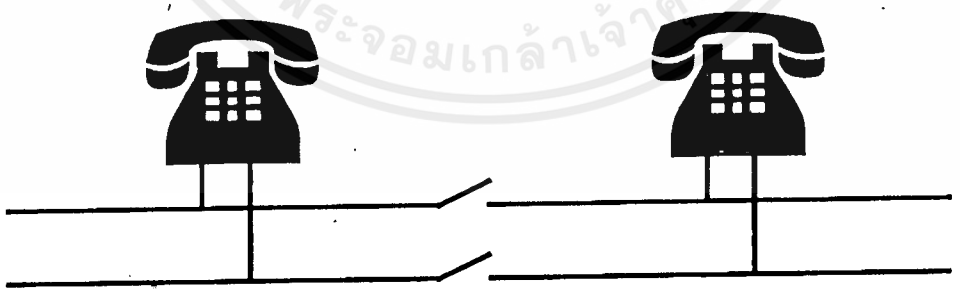
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการ

2.1 หลักการเชื่อมโยงสัญญาณโทรศัพท์

สัญญาณเสียงที่ส่งผ่านคู่สายโทรศัพท์เป็นสัญญาณที่ซ้อนทับกับกระแสไฟตรง การเชื่อมโยงสัญญาณโทรศัพท์ 2 คู่สายเข้าด้วยกันอาจทำอย่างง่ายๆ โดยการเชื่อมโยงโดยตรงตามรูปที่ 2.1.1 ในการเชื่อมโยงเช่นนี้จะต้องคำนึงถึง DC Voltage ที่ปรากฏบนสายทั้งสองกล่าวคือตรงกันและมีเครื่องโทรศัพท์เครื่องๆ หนึ่งยกหูอยู่เพื่อให้เกิด DC Loop ขึ้นที่ Line Circuit ภายในอุปกรณ์ชุมสายของเลขหมายทั้งสอง วงจรเชื่อมโยงที่นำมาใช้งาน ได้มาจากวงจรในตู้สลับสายโทรศัพท์ แบบพนักงานต่อแบบแบตเตอรี่ร่วมซึ่งยังมีใช้งานเป็นตู้สาขาภายใน (PABX) เนื่องจากราคาถูกมาก และค่าใช้จ่ายรวมทั้งค่าบำรุงรักษาต่ำ รูปที่ 2.1.2 คือวงจรเชื่อมโยงที่กล่าวถึง



รูปที่ 2.1.1 การเชื่อมโยงสัญญาณโทรศัพท์อย่างง่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การทำงาน, การออกแบบและการสร้าง

3.1 การทำงาน

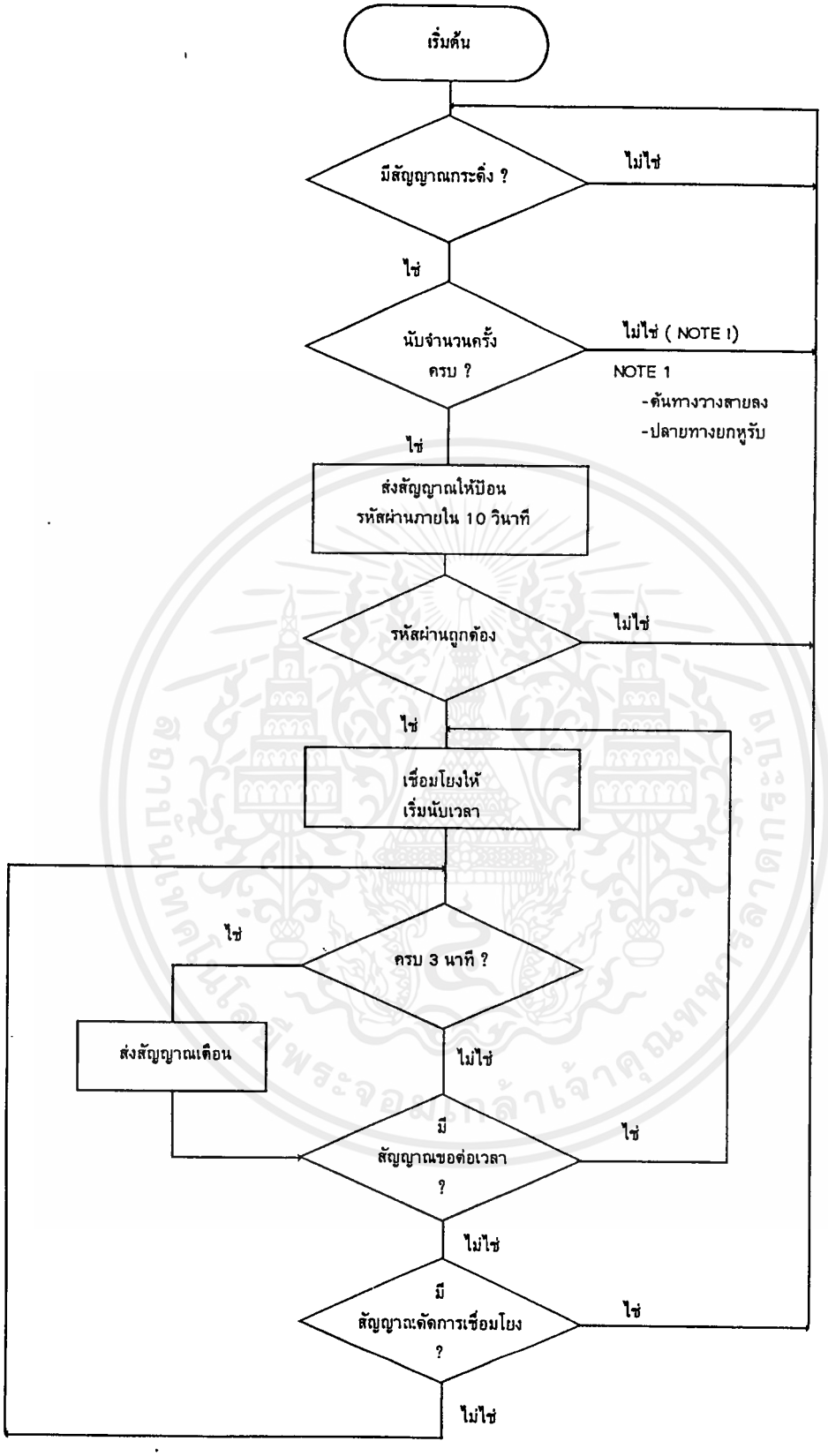
เมื่อเริ่มต้นการทำงาน จะเข้าสู่ขั้นตอนของการกำหนดรหัสผ่าน การกำหนดทำได้โดยกดปุ่มหมายเลขบนหน้าปัทม์ของโทรศัพท์ที่ต่อพ่วงอยู่ ถ้ากดหมายเลขไม่ครบภายในเวลาที่จำกัด จะกำหนดรหัสเริ่มต้น (Default) ให้ ผู้ใช้ที่มีหน้าที่ดูแลเครื่องเชื่อมโยงฯ สามารถใช้โทรศัพท์โทรเข้ามาเปลี่ยนแปลงรหัสผ่านได้ โดยใช้รหัสผ่านพิเศษ

มีวงจรตรวจจับสัญญาณกระดิ่งส่งให้ MPU เป็นตัวนับจำนวนครั้ง การตั้งจำนวนครั้งโดยการกำหนดใน Software เมื่อนับสัญญาณกระดิ่งได้ครบตามจำนวนครั้งที่ตั้งไว้ เครื่องเชื่อมโยงจะรับสายโดยอัตโนมัติ และเข้าสู่ขั้นตอนของการป้อนรหัสผ่าน

การป้อนรหัสผ่าน กำหนดให้กดปุ่มหมายเลขบนหน้าปัทม์โทรศัพท์ 3 หมายเลข ภายใน 10 วินาทีถ้ากดถูกต้องผู้ใช้จะได้รับ Dial Tone ของหมายเลขโทรศัพท์อีกคู่สายหนึ่งที่ต่ออยู่ ผู้ใช้สามารถกดหมายเลขของโทรศัพท์ที่ต้องการจะติดต่อด้วยได้เลย แต่ถ้ากดผิดเครื่องโยงจะยกเลิกการทำงานทั้งหมดและกลับไปอยู่ในสภาวะรอการเรียกครั้งต่อไป

การควบคุมการเชื่อมโยงอาศัยการกดปุ่มหมายเลขบนเครื่องโทรศัพท์ เครื่องเชื่อมโยงจำเป็นต้องมีการจำกัดเวลาการใช้ เพื่อป้องกันข้อผิดพลาดจากการที่ผู้ใช้วางหูโดยที่ยังไม่ได้ตัดการเชื่อมโยง ซึ่งจะทำให้ได้รับสัญญาณสายไม่ว่างในการเรียกเข้าในครั้งต่อไป เครื่องเชื่อมโยงจะส่งสัญญาณเตือนเมื่อครบเวลาที่กำหนด ถ้าต้องการสนทนาต่อ ผู้ใช้ต้องกดปุ่ม # ถ้าไม่กดเครื่องจะตัดการเชื่อมโยงภายใน 10 วินาที สำหรับการตัดการเชื่อมโยงจะใช้ปุ่ม *

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 Flowchart แสดงการทำงานของ

เครื่องเชื่อมโยงสัญญาณโทรศัพท์อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนแสดงผล ใช้ LED 4 ดวงเพื่อแสดงสภาวะของเครื่องเชื่อมโยงว่ากำลังทำงานอยู่ในขั้นตอนใดส่วนแสดงผลไม่มีความจำเป็นในการใช้งาน แต่จะมีประโยชน์สำหรับการวิเคราะห์ข้อขัดข้องที่อาจเกิดขึ้น การดูว่ากำลังทำงานอยู่ในขั้นตอนใดทำได้ดังนี้คือ

ตารางแสดงสภาวะการทำงานของเครื่องเชื่อมโยง

LED 1	LED 2	LED 3	LED 4	สภาวะ
ดับ	ดับ	ดับ	ดับ	ว่าง, รอการเรียกเข้า
ดับ	ติด	ติด	ดับ	สัญญาณกระดิ่งกำลังดัง
ดับ	ติด	ดับ	ดับ	สัญญาณกระดิ่งหยุด
ติด	ดับ	กระพริบ	ดับ	รอการป้อนรหัสผ่าน
ติด	ดับ	ดับ	ติด	เชื่อมโยงได้สำเร็จ, กำลังสนทนา
ติด	ดับ	กระพริบ	ติด	ส่งสัญญาณเตือนก่อนตัด

3.2 การออกแบบและการสร้าง HARDWARE

ไมโครคอมพิวเตอร์จะประกอบด้วยส่วนสำคัญ 3 ส่วนคือ

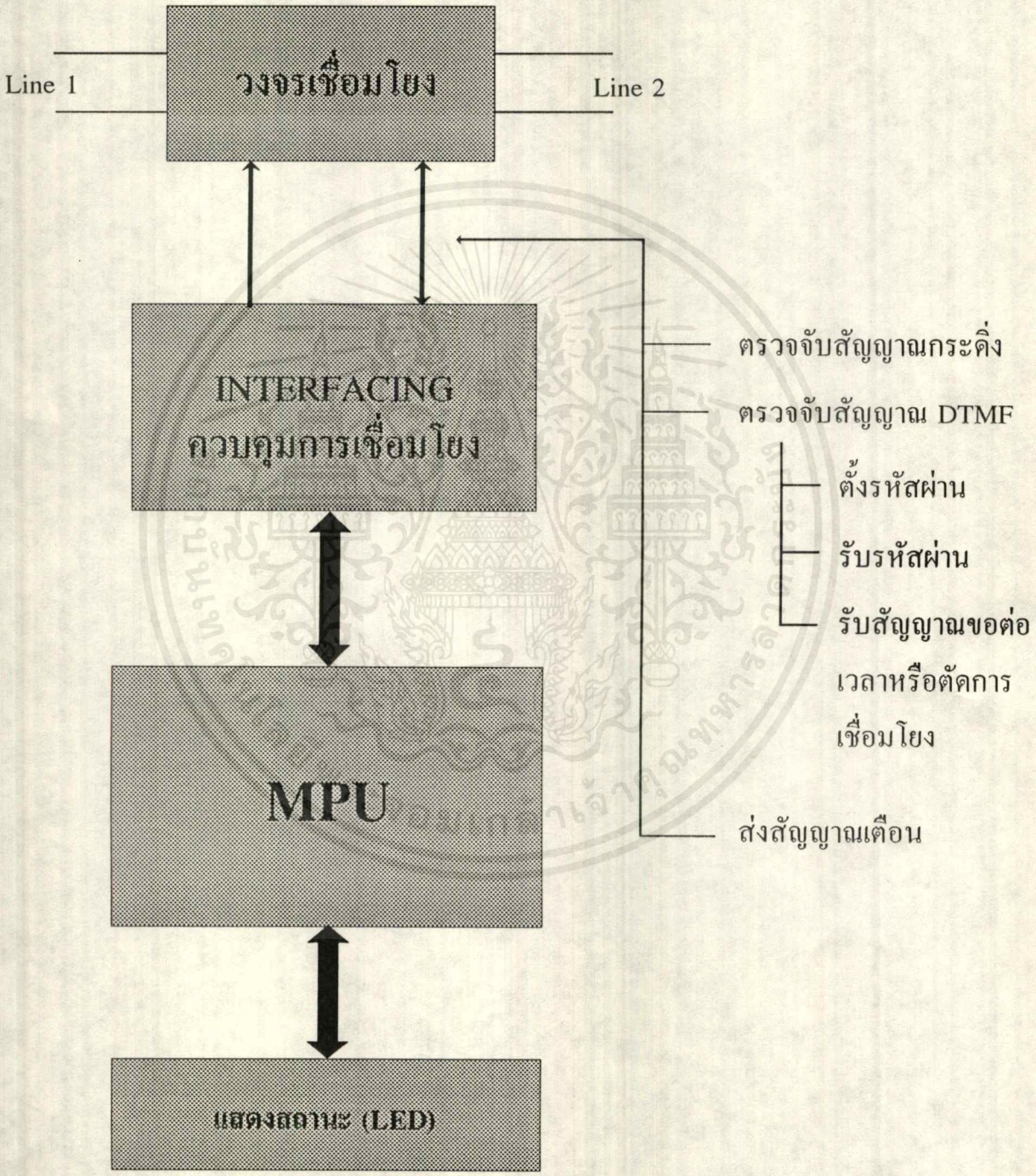
1 หน่วยประมวลผลกลาง

2 หน่วยความจำ

3 หน่วยรับข้อมูลและแสดงผลข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram
Automatic Telephone Patcher



รูปที่ 3.2.1 Block Diagram ของ Automatic Telephone Patcher

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

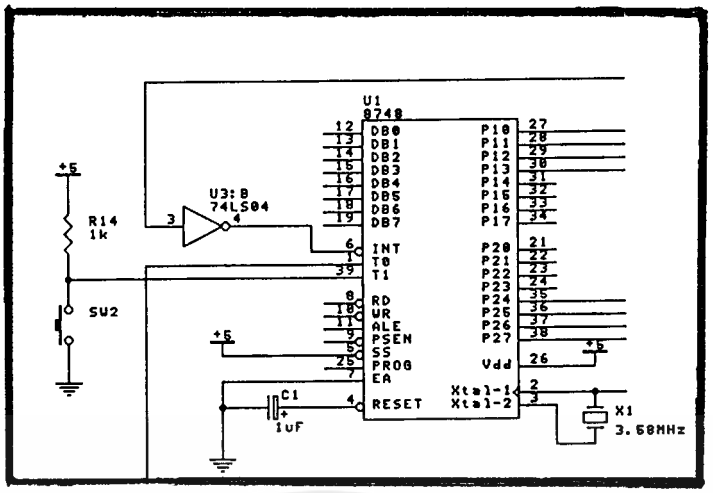
เมื่อไอซีเบอร์ 8748 ได้รวมเอาส่วนสำคัญทั้ง 3 ส่วนเข้าไว้ในตัวเดียวกัน การออกแบบวงจรจึงมีแต่เพียงส่วนที่เป็น Interface เท่านั้น ซึ่งได้แก่

- ส่วนถอดระหัสสัญญาณ DTMF
- ส่วนตรวจจับสัญญาณกระดิ่ง
- ส่วนควบคุมการเชื่อมโยงและแสดงผล
- ส่วนส่งสัญญาณเตือน

การที่เลือกใช้ ไมโครคอมพิวเตอร์ชิพเดียว 8748 ทำให้ลดจำนวนอุปกรณ์ลง เหลือเพียง Interface ที่จำเป็น ในแง่ของการออกแบบทำให้ Print Circuit Board มีขนาดเล็ก ลายวงจรลดความซับซ้อน ความเชื่อถือได้ในการใช้งานสูงขึ้น การออกแบบวงจรและการทำงานอธิบายเป็นส่วนๆได้ดังนี้

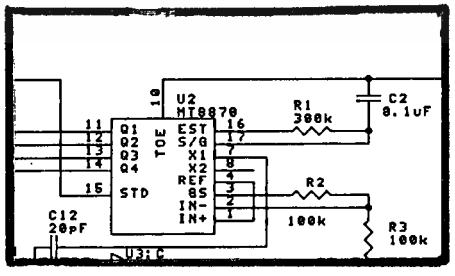
3.2.1 **ไมโครคอมพิวเตอร์** ประกอบด้วย ไมโครคอมพิวเตอร์ชิพเดียว เบอร์ 8748 (U1), คริสตอล 3.58 MHz (X1) Capacitor C1 ต่อกับขา RESET เป็นวงจร POWER ON RESET มีการทำงานคือ เมื่อป้อนไฟเข้าเครื่อง คุณสมบัติของ Capacitor จะเริ่ม Charge ประจุ ทำให้ขา RESET เป็นลอจิค 0 ชั่วขณะหนึ่ง คริสตอลใช้ 3.58 MHz เพราะต้องการที่จะใช้คริสตอลเพียงตัวเดียว กับ MPU 8748 และ MT8870 คริสตอลความถี่ 3.58 MHz เป็นคริสตอลที่ใช้กับ วงจรนาฬิกา และโทรศัพท์กดปุ่มทั่วไป หาได้ง่ายและมีราคาถูก จุดสำคัญของการ ใช้งาน 8748 อีกจุดหนึ่งคือขา EA (External Access) ถ้าต้องการให้ MPU 8748 เข้าถึงรหัสคำสั่งที่เก็บอยู่ที่ EPROM ที่อยู่ภายในตัวของมัน จะต้องให้ลอจิค 0 กับขา EA ออกแบบให้ Port 1 เป็น INPUT Port รับสัญญาณจาก ภาคถอดรหัส DTMF และใช้งานเพียง 4 Bits, Port 2 เป็น OUTPUT Port ส่งสัญญาณไปขับ LED และ RELAY ขา INT ต่อกับสัญญาณแสดงความพร้อมของสัญญาณ DTMF, ขา T0 ต่อกับภาคตรวจจับสัญญาณกระดิ่ง ขา T1 ต่อกับ SW2 ซึ่งทำหน้าที่เป็น MANUAL SWITCH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2.2 วงจรไมโครคอมพิวเตอร์ที่ใช้ในโครงงาน

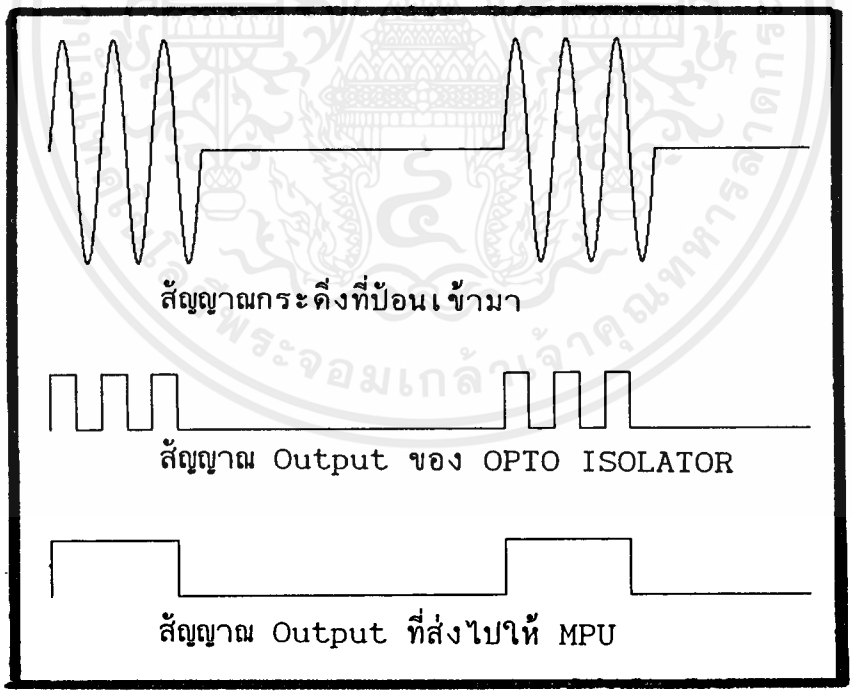
3.2.2 ส่วนถอดรหัสสัญญาณ DTMF ใช้ IC เบอร์ MT8870 เปลี่ยน Dual Tone Multi Frequency ให้เป็นรหัส Binary รวมทั้งกำเนิดสัญญาณ Strobe แสดงความพร้อมของข้อมูลหมายเลขปุ่มที่กดซึ่งจะส่งไป Interupt MPU ให้อ่านรหัส Binary เข้าไปประมวลผล สำหรับการทำงานได้กล่าวโดยละเอียดแล้วในบทที่ 2 การ Interface กับไมโครคอมพิวเตอร์ ออกแบบให้ Interupt MPU เมื่อได้รับสัญญาณ DTMF โดยการต่อขา STD ผ่าน Inverter ไปต่อกับขา INT ของ MPU 8748 ส่วนสัญญาณ CLOCK ได้มาจาก CLOCK GENERATOR ภายใน MPU สัญญาณ DTMF จะถูก Coupling มาจาก วงจรเชื่อมโยงทางเสียง



รูปที่ 3.2.3 วงจรถอดรหัสสัญญาณ DTMF

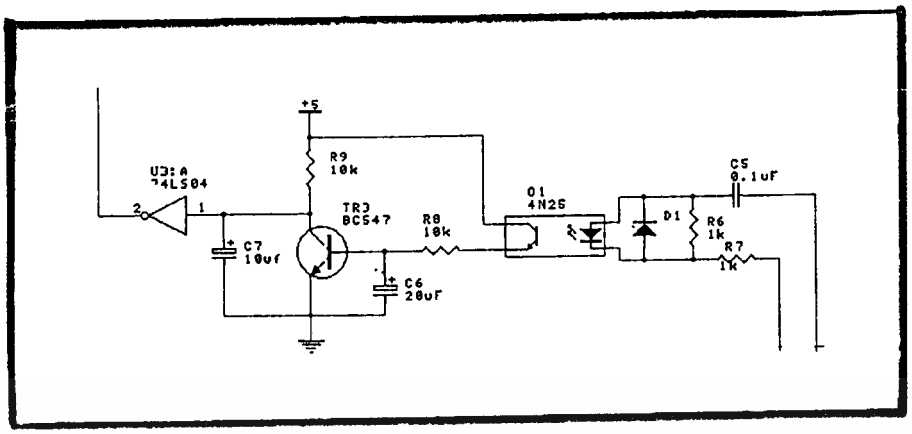
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 **ส่วนตรวจจับสัญญาณกระดิ่ง** รับสัญญาณกระดิ่งมาจาก LINE 1 ผ่าน CONTACT ของ RELAY RL1 ด้าน NORMALY CLOSED Capacitor C_{10} ทำหน้าที่ป้องกัน แรงไฟ DC R_6, R_7 ร่วมกับ Diode D_1 ทำหน้าที่ Rectifier ลดแรงดันของสัญญาณกระดิ่ง จาก 75-100 โวลท์ ลงมาให้พอเหมาะที่จะทำให้ OPTO ISOLATOR 4N25 ทำงานได้ (มีกระแสไหลประมาณ 10-15 mA) OPTO ISOLATOR เป็นอุปกรณ์ที่ภายในประกอบด้วย LED และ Photo Transistor อาศัยการ Coupling โดยทางแสง แต่จะแยกจากกันในทางไฟฟ้า เมื่อมีสัญญาณกระดิ่งเข้ามาทำให้ Photo Transistor นำกระแส R_8 และ C_8 ทำหน้าที่ Filter เพื่อให้ได้พัลซ์ ที่มีลักษณะ ON และ OFF ตามสัญญาณกระดิ่งที่ส่งเข้ามา ทรานซิสเตอร์ TR_3 และ Inverter U3:A ที่ทำหน้าที่เป็น Buffer ส่งสัญญาณให้ MPU ที่ขา TO



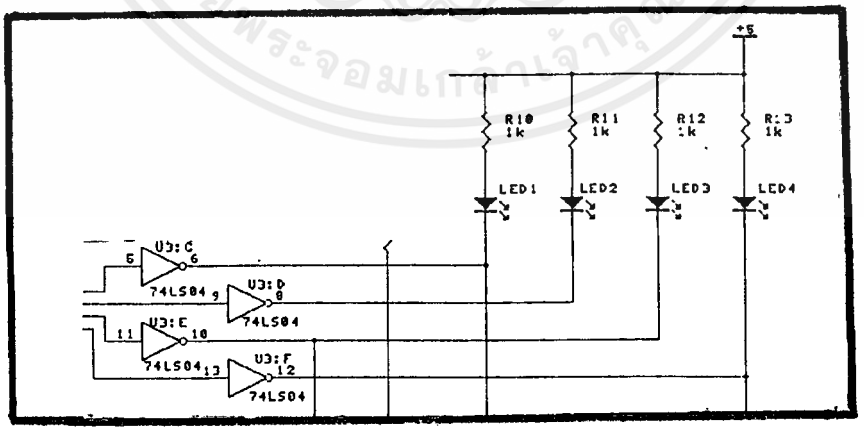
รูปที่ 3.2.4 แสดงการทำงานของภาคตรวจจับสัญญาณกระดิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2.5 วงจรภาคตรวจจับสัญญาณกระดิ่ง

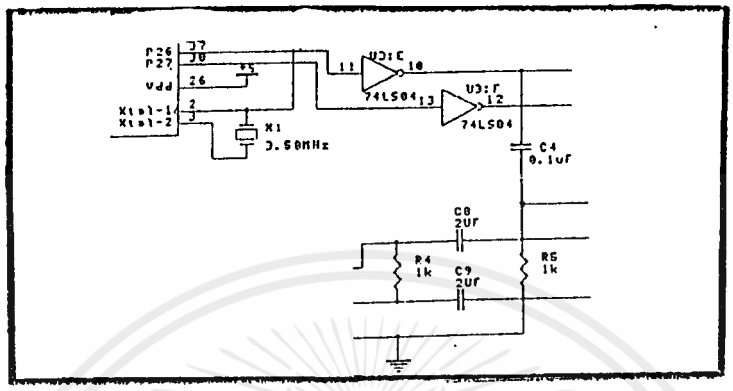
3.2.4 **ส่วนควบคุมการเชื่อมโยงและแสดงผล** วงจรเชื่อมโยงสัญญาณเสียงได้อธิบายแล้วในบทที่ 2 เรื่องหลักการเชื่อมโยงที่ใช้ในโครงงาน การควบคุมการเชื่อมโยงใช้ RELAY RL_1 และ RL_2 ซึ่งถูกขับด้วย TR_1 และ TR_2 ตามลำดับ การทำงานคือ เมื่อ MPU ควบคุมให้ RELAY RL_1 ทำงานจะส่ง LOGIC 1 ออกมาที่ Port 2 ขา P24 เนื่องจาก MPU มี Sink Current ต่ำมาก จึงต้องใช้ U3 เป็น Buffer การแสดงสถานะทำได้โดยต่อ LED ที่ OUTPUT ของ INVERTER แต่ละตัว



รูปที่ 3.2.6 แสดงวงจรควบคุมและแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5 ส่วนส่งสัญญาณเตือน ใช้ Capacitor C_4 Coupling สัญญาณเตือนที่กำเนิดขึ้นโดย Software เข้ามาในวงจรเชื่อมโยงสัญญาณเสียงที่จุดต่อของ C_8 และ R_5 จุดนี้จะเป็นจุดที่ Coupling สัญญาณ DTMF ไปยังวงจร ถอดรหัส



รูปที่ 3.2.7 แสดงภาคส่งสัญญาณเตือน

3.3 การออกแบบ Software

MPU เบอร์ 8748 ถึงแม้จะมี EPROM ที่ใช้เก็บรหัสคำสั่งอยู่ภายในแต่ก็สามารถเข้าถึงรหัสคำสั่งที่เก็บอยู่ใน EPROM ที่ต่ออยู่ภายนอกได้ ในขั้นพัฒนาโปรแกรมจะจัดวงจรให้ MPU อ่านรหัสคำสั่งใน EPROM ที่ต่ออยู่ภายนอก ใช้ EPROM Emulator ทำงานร่วมกับ PC (Personal Computer) โดยต่อผ่าน Serial Port (RS232C) แทน EPROM ที่ต่ออยู่ภายนอก การเขียนโปรแกรมใช้ Text Editor (สำหรับการทำโครงงานนี้ใช้ Sidekick เป็น Editor) เขียนโปรแกรมภาษา Assembly ซึ่งเป็นไฟล์ตัวอักษร (Text file)ขึ้น ใช้ Cross16 Meta Assembler ซึ่งเป็น Cross Assembler ที่ Run ใน PC เป็นตัวแปลไปเป็นไฟล์รหัสคำสั่ง ที่อยู่ในรูปรหัสตัวเลขฐานสิบหก แบบของอินเทล (INTEL HEX Format) ส่งไฟล์ที่ได้นี้ออกทาง Serial Port ไปยัง EPROM Emulator และสามารถทดสอบโปรแกรมได้ทันที โดยการ Reset CPU บนบอร์ดที่ทำการทดลอง เราสามารถแก้ไขโปรแกรม, แปล, และส่งไปยัง EPROM Emulator ซ้ำแล้วซ้ำอีกจนกว่าจะได้การทำงานของโปรแกรมที่สมบูรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทดสอบโปรแกรม เราจะแบ่งโปรแกรมออกเป็นส่วน และทดสอบไปทีละส่วนดังนี้

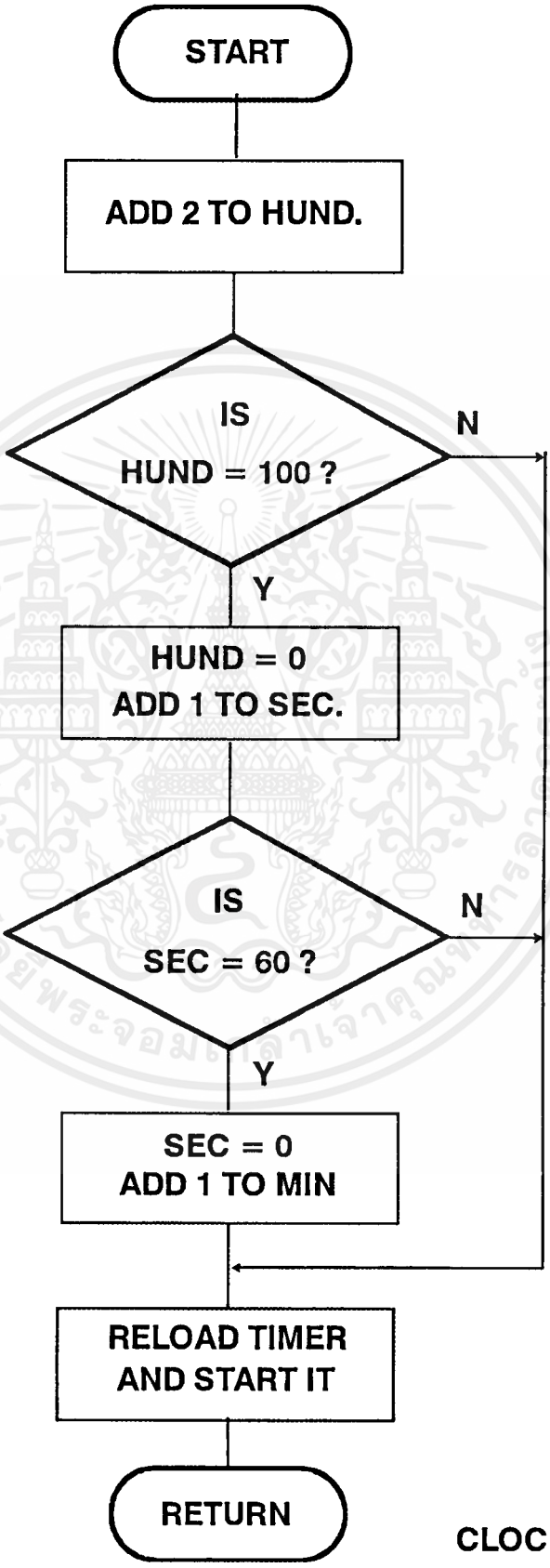
3.3.1 ส่วนของฐานเวลา (Real Time Clock) ใช้ความสามารถของ 8748 ในส่วนของ Timer Interupt โดยจะเกิดการ Interupt ทุกๆ 20 มิลลิวินาที จัด Memory ไว้ 3 Bytes ให้ชื่อว่า TMEBUF เพื่อเก็บค่าของเวลาหลักมิลลิวินาที, วินาที, และนาที การทำงานตาม Flow chart ในรูปที่ 3.3.1

3.3.2 ส่วนรับข้อมูลจากการกดปุ่มหมายเลขโทรศัพท์ IC ที่ทำหน้าที่ถอดรหัสหมายเลข จะส่งรหัสหมายเลข พร้อมทั้งส่งสัญญาณแสดงความพร้อมของข้อมูล เราใช้สัญญาณแสดงความพร้อมเป็นสัญญาณ Interupt ต่อผ่าน Inverter แล้วต่อเข้ากับขา INT ดังนั้นทุกครั้งรับสัญญาณ DTMF ได้ CPU จะอ่านรหัสของปุ่มหมายเลขเข้าทาง Port 1 ขา P10-P13 มาเก็บไว้ใน Memory ที่ให้ชื่อว่า KBDBUF และ Clear Flag F1 เมื่อไรที่โปรแกรมต้องการใช้รหัสหมายเลข ก็ตรวจดูที่ Flag นี้ และเมื่ออ่านรหัสหมายเลขไปใช้แล้วก็จะ Complement Flag เพื่อรอการรับรหัสหมายเลขครั้งต่อไป การทำงานแสดงใน Flow chart รูปที่ 3.3.2

3.3.3 ส่วนนับสัญญาณกระดิ่ง เป็น Subroutine ที่คอยตรวจดูการเปลี่ยนแปลงของขา T0ซึ่งต่ออยู่กับวงจรตรวจจับสัญญาณกระดิ่ง ในสภาวะปกติ ขา T0 จะเป็น Logic 0 เมื่อได้รับสัญญาณกระดิ่ง(Ring on) จะเปลี่ยนเป็น Logic 1 และเมื่อสัญญาณกระดิ่งหยุด (Ring off) จะกลับเป็น Logic 0 Subroutine นี้จะนับจำนวนครั้งของการเปลี่ยนแปลงจาก Logic 1 ไปเป็น Logic 0 การสิ้นสุดการทำงานและออกจาก Subroutine จะเกิดขึ้นได้ใน 2 กรณีคือ

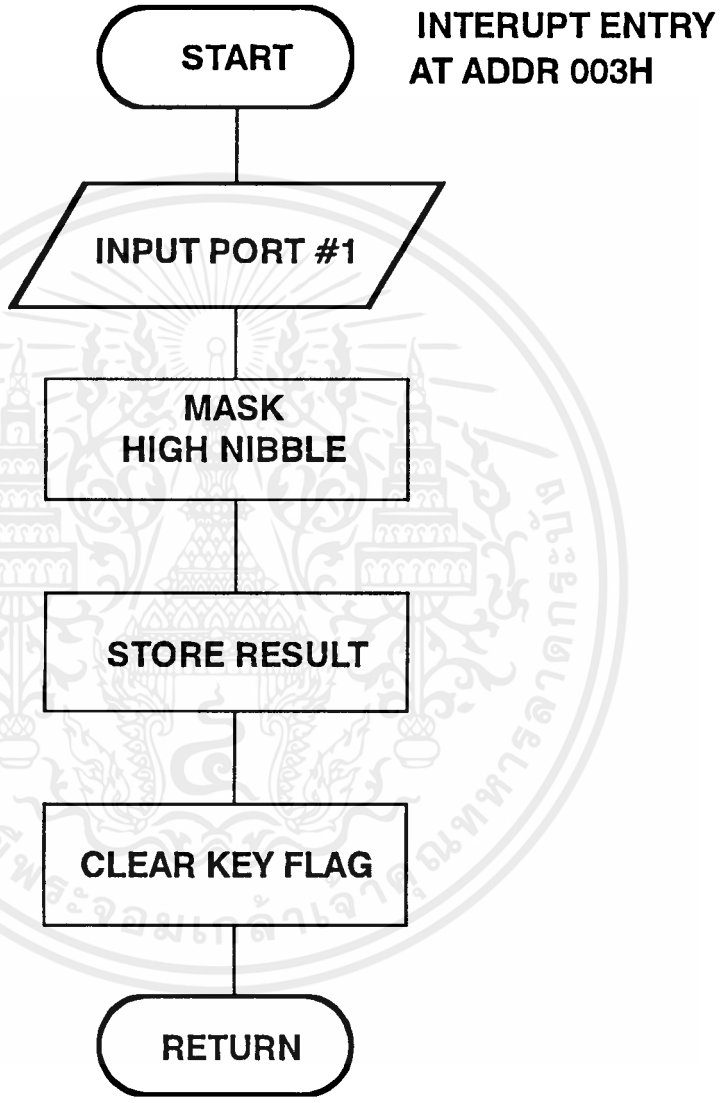
1 นับสัญญาณกระดิ่งครบ 10 ครั้ง การออกจาก Subroutine ในกรณีนี้เป็นการจบการทำงานตามปกติ Flag F0 จะถูก Clear

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3.1
CLOCK SUBROUTINE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3.2
DTMF REC. SUBROUTINE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

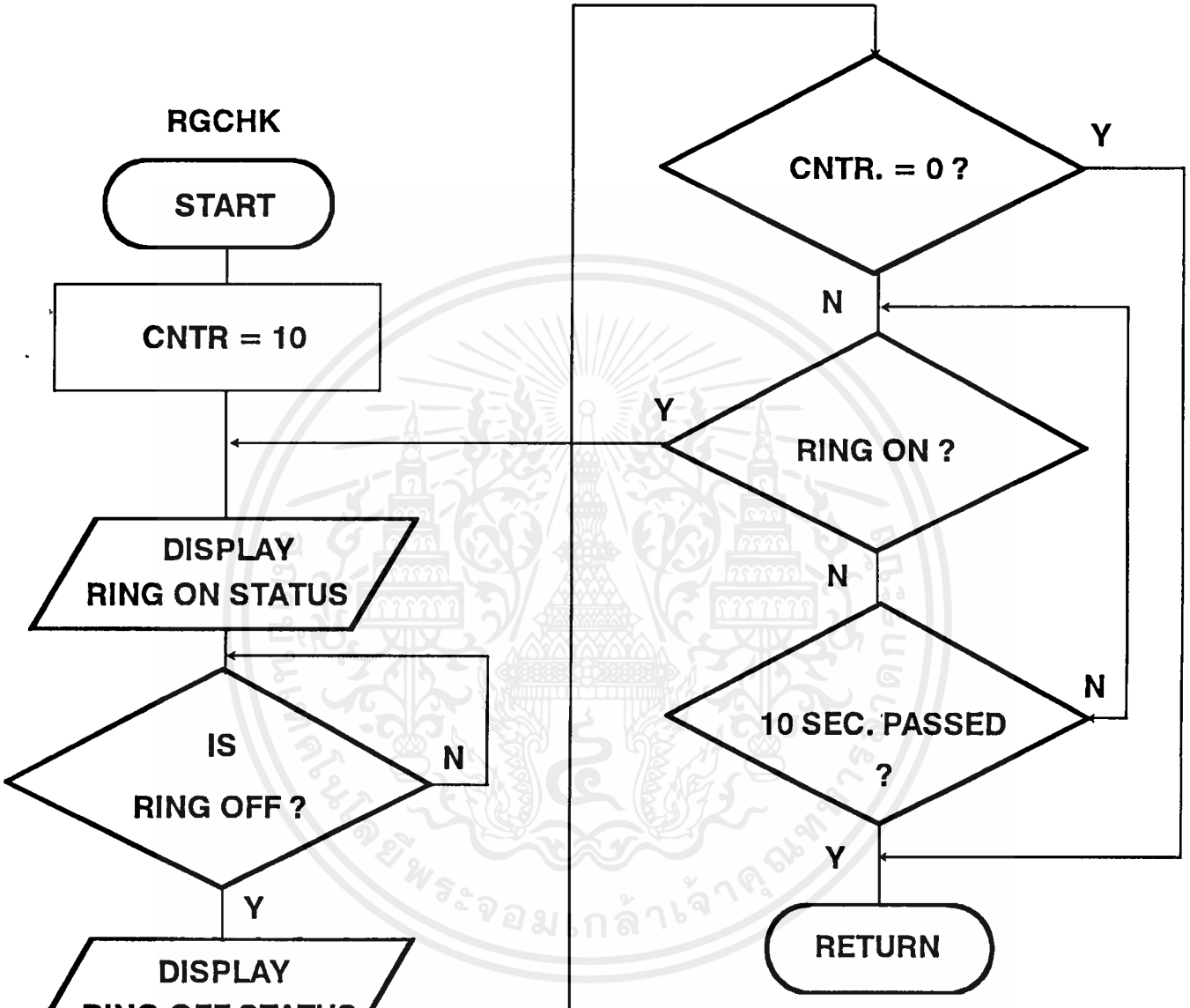
2 ในกรณีที่ผู้เรียกวางหูไปก่อน หรือเครื่องที่พ่วงอยู่กับเครื่องที่ถูกเรียกถูกยกหูขึ้นก่อน การเขียนโปรแกรมในส่วนนี้จะต้องเข้าไปอ่าน TMEBUF หลักวินาที โดยตรวจสอบเงื่อนไขว่าถ้าสัญญาณกระดิ่งขาดหายไปนานกว่า 10 วินาที ก่อนที่จะครบ 10 ครั้ง ก็ให้จบการทำงานของ Subroutine และ Flag F0 จะอยู่ในสภาวะ Set การทำงานด้ง Flow chart ในรูปที่ 3.3.3

3.3.4 ส่วนกำเนิดสัญญาณเตือน เป็น Subroutine ที่กำเนิดเสียงความถี่ประมาณ 1 KHz ด้งประมาณ ครึ่งวินาที และมีการวนรอบเพื่อหน่วงเวลาอีก ครึ่งวินาที สัญญาณที่ได้นี้ส่งออกทาง ขา P25 ผ่าน Coupling Capacitor เข้าไปในวงจรเชื่อมโยง การทำงานด้ง Flow chart ในรูปที่ 3.3.4

3.3.5 ส่วนสำหรับการกำหนดรหัสผ่าน ใช้ Memory 3 Bytes ให้ชื่อว่า WORD1 เพื่อเก็บรหัสผ่าน 3 ตัว และเรียกใช้ Subroutine กำเนิดสัญญาณเตือน 10 รอบเพื่อให้เกิดสัญญาณเตือน 10 ครั้ง ทุกรอบจะตรวจสอบ Flag F1 ถ้าอยู่ในสภาวะ Clear ก็อ่านรหัสใน KBDBUF มาไว้ใน WORD1 เพิ่มค่าตัวชี้ WORD1 ไปชี้ Byte ถัดไปและทำซ้ำจนได้รหัสผ่านครบ 3 ตัว แต่ถ้าไม่มีการป้อนรหัสผ่านหรือป้อนไม่ครบ 3 ตัวใน 10 รอบของการเรียกใช้ Subroutine กำเนิดสัญญาณเตือน ก็จะตั้งรหัสผ่านตามค่า Default ที่เขียนไว้ในโปรแกรมและออกจาก Subroutine การทำงานของ Subroutine กำหนดรหัสผ่านแสดงใน Flow chart รูปที่ 3.3.5

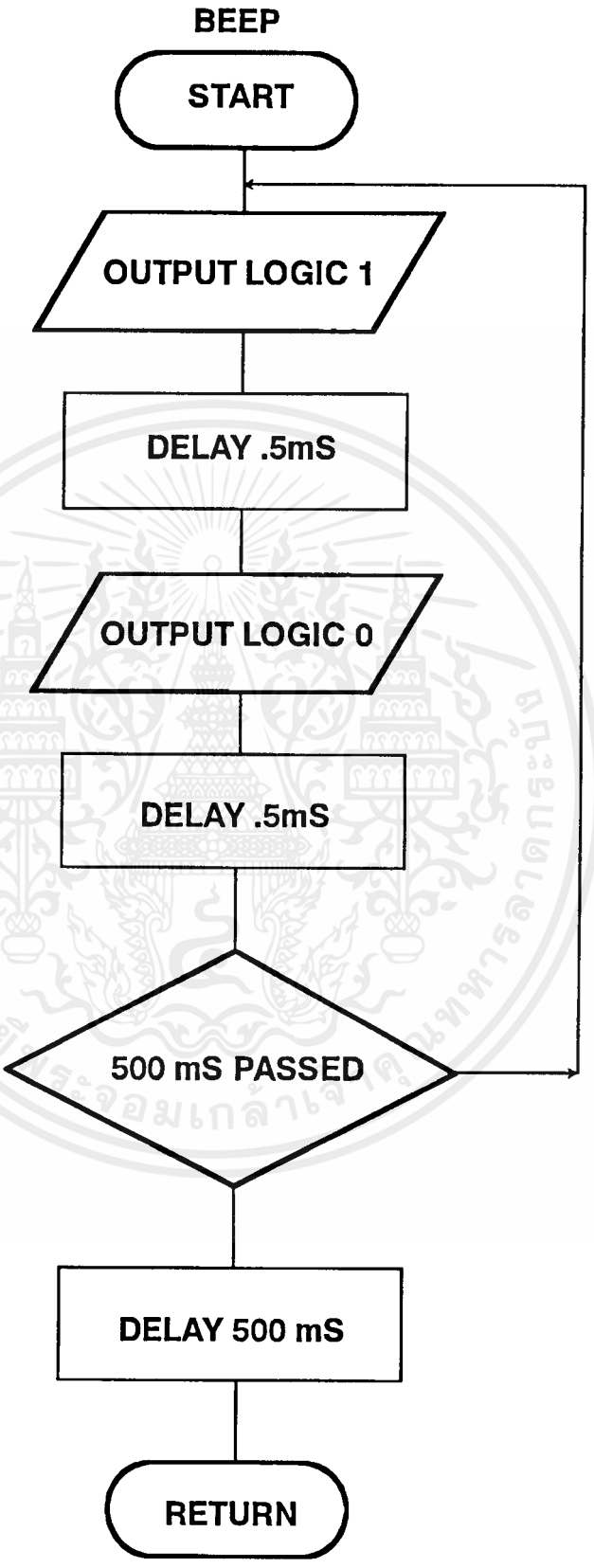
3.3.6 ส่วนของการตรวจสอบระหัสผ่าน เริ่มต้น Subroutine นี้จะตรวจสอบว่า WORD1 เป็น 0 หรือไม่ ถ้าใช่หมายถึงการอนุญาตให้ใช้งานโดยไม่ต้องมีรหัสผ่าน Flag F0 จะถูก Clear และจบการทำงานของ Subroutine นี้ ถ้าไม่ไช่ เข้าสู่ขั้นตอนของการตรวจสอบรหัสผ่าน ซึ่งมีการทำงานคล้ายกันกับส่วนสำหรับการกำหนดรหัสผ่าน แต่จะอ่าน KBDBUF เข้ามาเปรียบเทียบกับ WORD1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



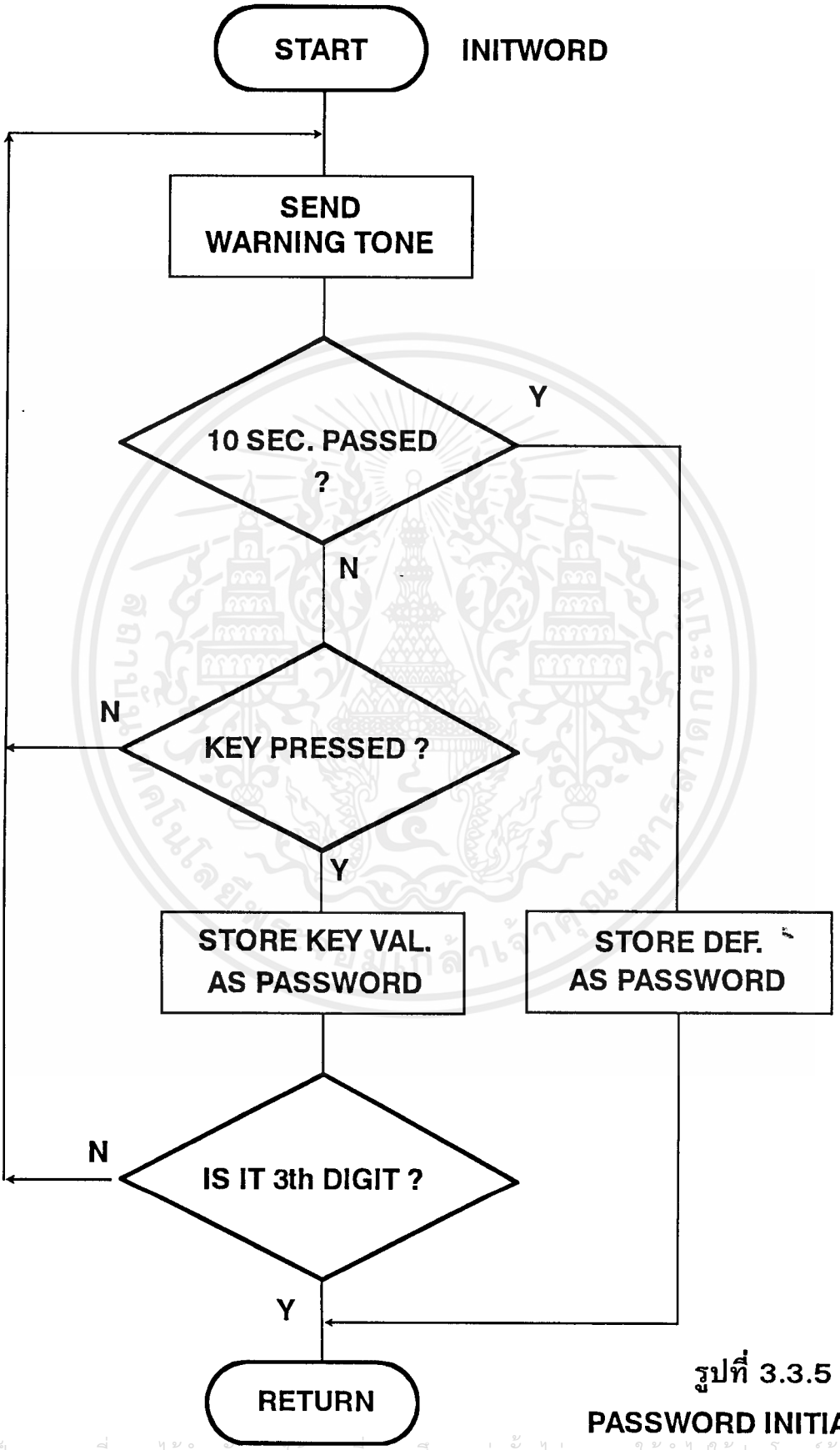
รูปที่ 3.3.3
RGCHK SUBROUTINE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3.4
BEEP SUBROUTINE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ [illegible] ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3.5
PASSWORD INITIALIZED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ 6 Byte ต่อไปจนครบ 3 Byte ผลของการเปรียบเทียบจะอยู่ใน Memory ที่ให้ชื่อว่า ERRBUF หลังจากเปรียบเทียบครบ 3 Byte แล้วถ้า ERRBUF ไม่เท่ากับ 0 แสดงว่ารหัสผ่านไม่ถูกต้อง F0 จะถูก Set และจบการทำงานของ Subroutine แต่ถ้า ERRBUF เท่ากับ 0 แสดงว่ารหัสผ่านถูกต้อง F0 จะถูก Clear ก่อนจบการทำงานของ Subroutine สำหรับลำดับการทำงานแสดงใน Flow chart รูปที่ 3.3.6

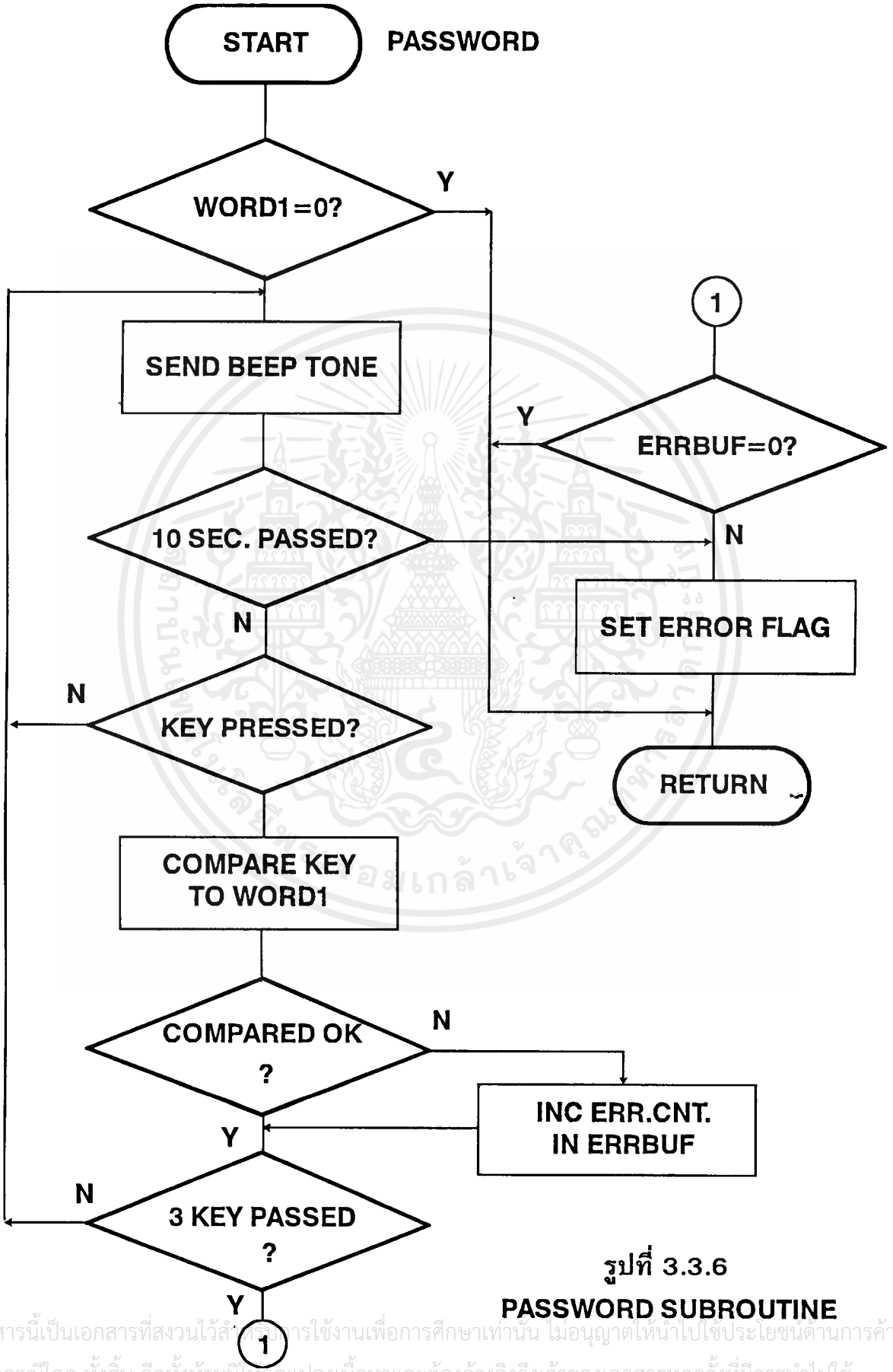
3.3.7 ส่วนควบคุมการเชื่อมโยง เริ่มต้น Subroutine ด้วยการ Clear IMEBUF หลักนาทีให้เป็น 0 แล้ววนรอบเพื่อตรวจสอบการหมดเวลา (ครบ3 นาที) และการตรวจสอบ Flag F1 ซึ่งเป็น Flag แสดงการกดปุ่มหมายเลข ถ้ามีการกดจะอ่าน KBDBUF มาตรวจสอบว่าเป็นรหัสใด

ถ้าเป็นหมายเลข 13 จะเรียก Subroutine สำหรับการกำหนดรหัสผ่าน นั่นคือสามารถเปลี่ยนแปลงรหัสผ่านได้ ถ้าเป็นหมายเลข 12 คือปุ่ม # โปรแกรมจะกลับไป Clear IMEBUF หลักนาทีให้เป็น 0 เป็นการเริ่มต้นการจับเวลาใหม่

ถ้าเป็นหมายเลข 11 คือปุ่ม * จะออกจาก Subroutine

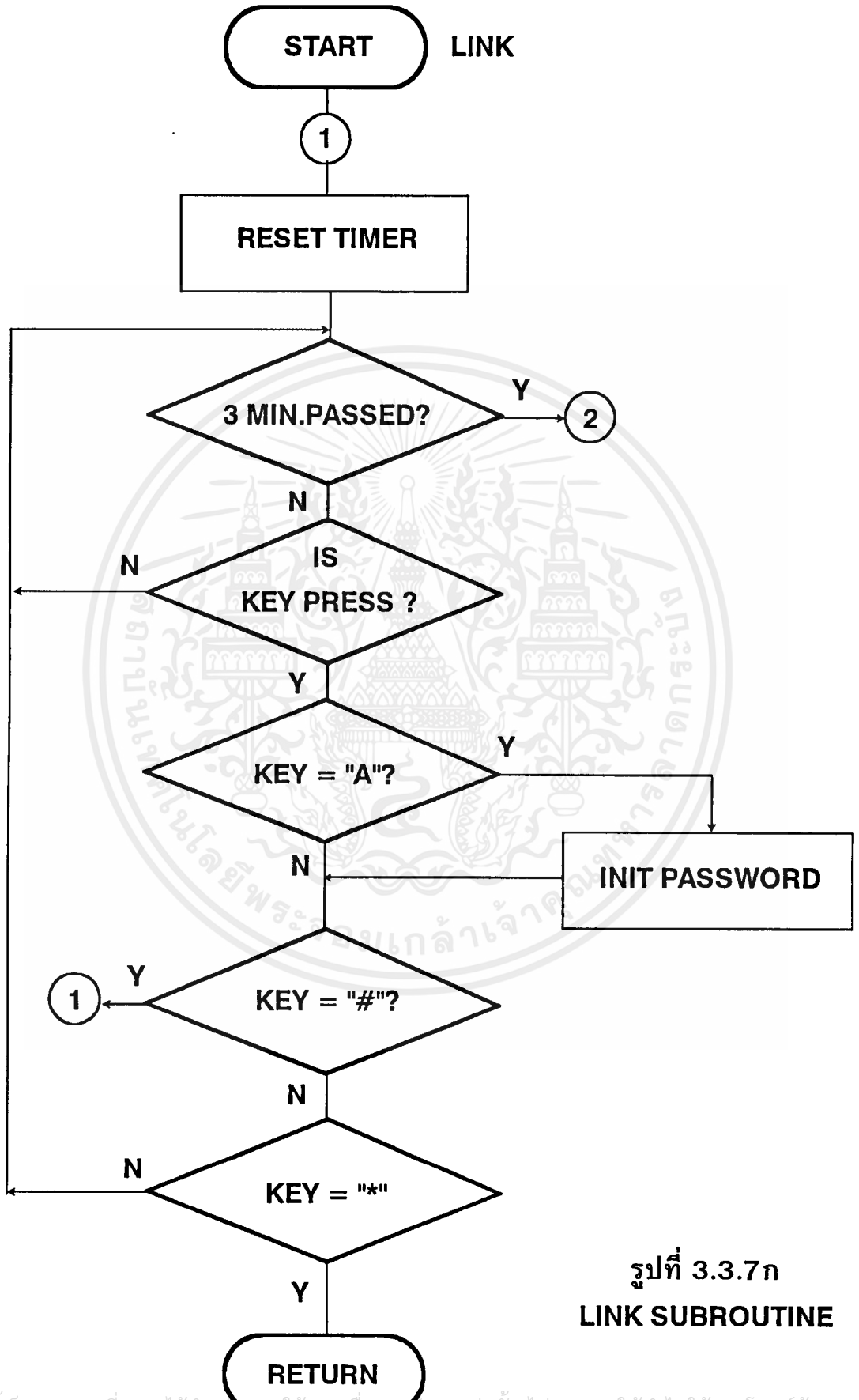
ในการตรวจสอบการหมดเวลา ถ้าพบว่าครบ 3 นาที ก็จะเข้าสู่วนรอบของการส่งสัญญาณเตือน และการตรวจการกดปุ่มหมายเลข ถ้าพบว่าเป็นปุ่ม # ก็จะเริ่มต้นการจับเวลาใหม่ วนเวียนไปเช่นนี้ สรุปได้ว่าการจบการทำงานของ Subroutine เกิดได้ 2 กรณี คือ มีการกดปุ่ม * หรือ ครบเวลา 3 นาทีโดยไม่มีการกดปุ่ม # ลำดับการทำงานแสดงด้วย Flow chart ในรูปที่ 3.3.7ก และ 3.3.7ข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



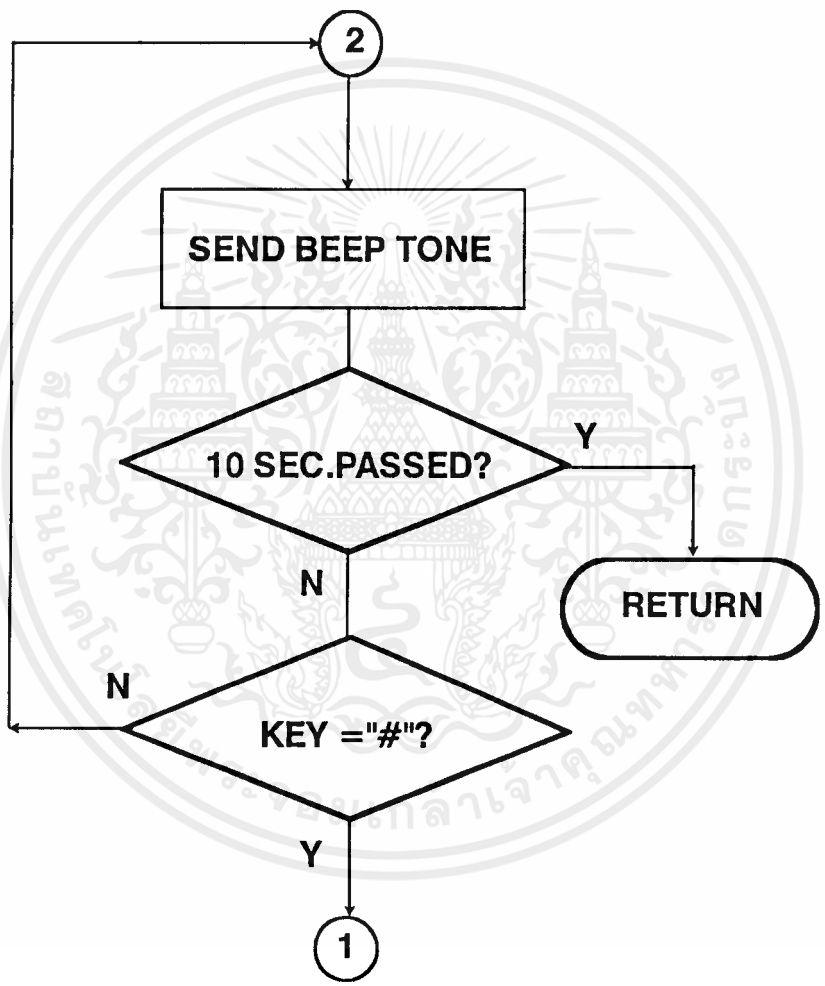
รูปที่ 3.3.6
PASSWORD SUBROUTINE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



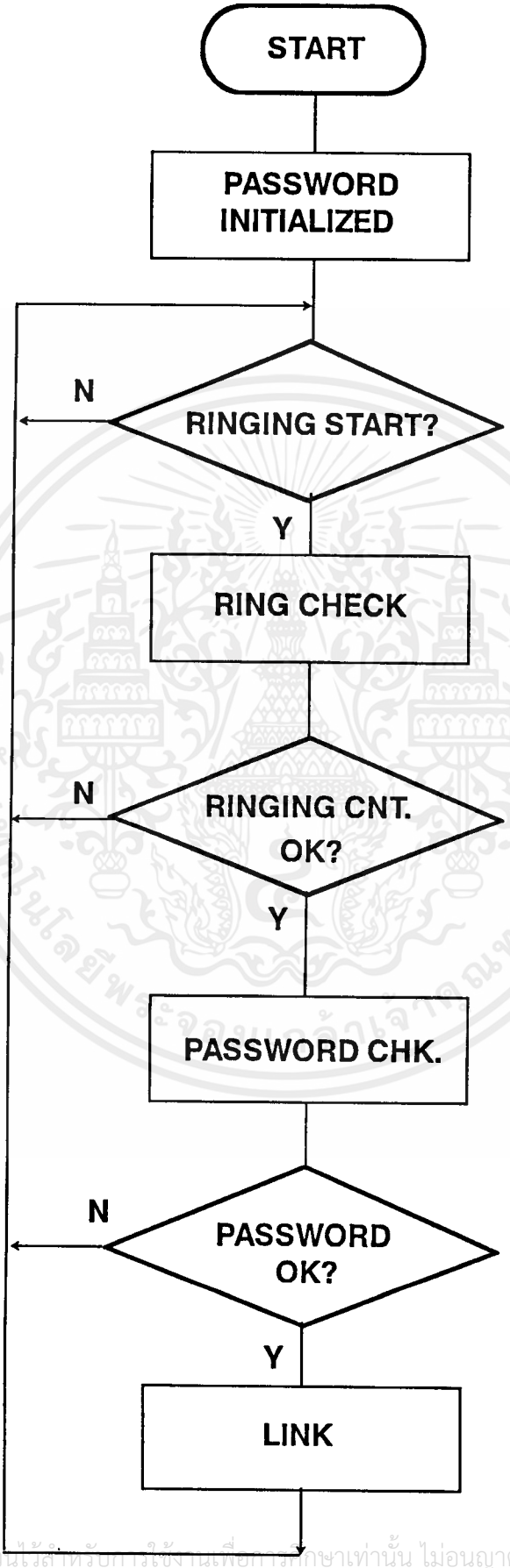
รูปที่ 3.3.7ก
LINK SUBROUTINE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3.7ข
LINK SUBROUTINE (CONTINUE)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3.8
MAIN PROGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.8 โปรแกรมหลัก เริ่มด้วยการวนรอบเมื่อเริ่มเปิดเครื่อง การ Clear buffer และRegister บางตัว และการเรียก Subroutine สำหรับการกำหนดระหัสผ่าน จากนั้นจะเข้าสู่วนรอบของโปรแกรมหลักซึ่งได้แก่ การตรวจสอบการเริ่มต้นของสัญญาณกระดิ่ง การเรียก Subroutine ของการนับสัญญาณกระดิ่ง การตรวจสอบระหัสผ่านที่รับเข้ามากับระหัสผ่านที่กำหนดไว้ และการเชื่อมโยงโดยแต่ละขั้นตอนจะส่งสัญญาณแสดงสภาวะ ออกทาง ขา P24 - P27 ของ 8748 เพื่อขับ LED 1 - LED 4 ในการควบคุมการเชื่อมโยงจะใช้ Relay 2 ตัว ในการเชื่อมโยงสัญญาณโทรศัพท์เข้าด้วยกัน โดยสัญญาณควบคุมการทำงานของ Relay ทั้งสองก็คือ ขา P24 และ P27 นั่นเอง สำหรับลำดับขั้นตอนของการทำงานแสดงใน Flow chart รูปที่ 3.3.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและสรุปผล

การทดลองทางด้าน Hardware

ภาคตรวจจับสัญญาณกระดิ่ง (RINGING DETECTOR)

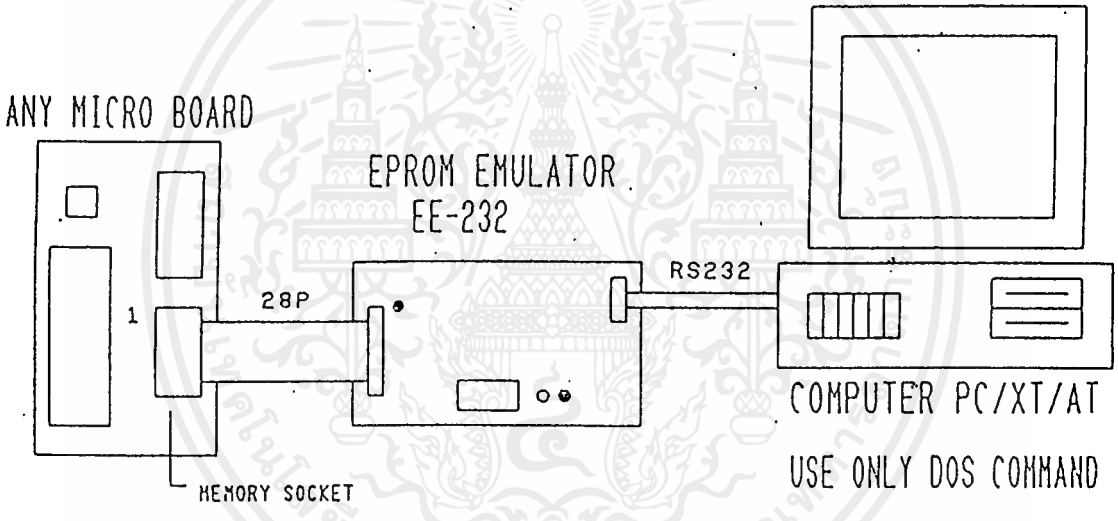
เนื่องจากเครื่องเชื่อมโยงสัญญาณโทรศัพท์นี้ ถูกออกแบบให้ต่ออยู่กับคู่สายโทรศัพท์ในลักษณะเดียวกับเครื่องพ่วง เพื่อรอรับสัญญาณกระดิ่งที่เรียกเข้ามา ดังนั้นวงจรตรวจจับสัญญาณกระดิ่ง ต้องมี Impedance สูงเพื่อไม่ให้ LOAD สัญญาณ Ringing จากคู่สายโทรศัพท์มากเกินไป แต่จากการทดลองถ้ามี Impedance สูงเกินไปจะทำให้ภาคตรวจจับสัญญาณกระดิ่งทำงานไวเกินไป การยกหูหรือวางหูเครื่องที่ต่อพ่วงอยู่ (DC Voltage บนคู่สายโทรศัพท์เปลี่ยนแปลงจาก 48 โวลท์เป็น 8 ~ 12 โวลท์) จะทำให้วงจรตรวจจับเริ่มทำงาน การแก้ไขทำได้โดยเลือกค่า C_5, R_6, R_7 ให้เหมาะสม อีกจุดหนึ่งที่เป็นปัญหาสำหรับการทำงานของวงจรตรวจจับสัญญาณกระดิ่งคือ ต้องเลือกค่า C_6 และ C_7 ให้เหมาะสม เพื่อให้สัญญาณที่ส่งให้ MPU เป็น Rectangular Wave โดยสมบูรณ์ มิฉะนั้นการนับจำนวนครั้งของสัญญาณกระดิ่งจะผิดพลาด สำหรับการทดลอง Interface ส่วนอื่นๆ ไม่ประสพปัญหาแต่ประการใด

การทดลองด้าน Software

Subroutine PASSWORD เมื่อเริ่มการทดลองโครงงาน Routine ที่เขียนขึ้นเพื่อการตรวจสอบรหัสผ่าน ทำในลักษณะที่เมื่อป้อนรหัสผ่านตัวใดผิด ก็จะยกเลิกการทำงานทันที วิธีนี้เดารหัสผ่านได้ง่าย จึงได้แก้ไขให้ต้องป้อนจนครบ 3 ตัว จึงจะเชื่อมโยงให้หรือยกเลิกการทำงานในกรณีป้อนรหัสผิด ด้วยวิธีนี้ผู้ที่เรียกเข้ามาจะไม่มีโอกาสทราบว่ารหัสผ่านตัวใดผิด แต่โปรแกรมจะซับซ้อนกว่าวิธีแรกมาก

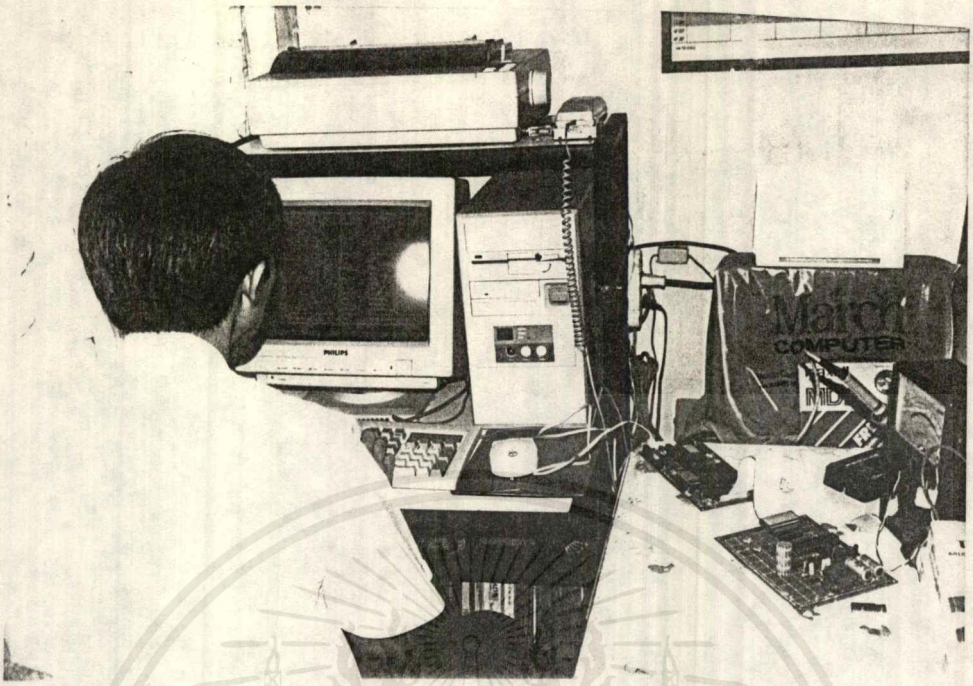
Subroutine CLOCK ซึ่งเป็น Real Time เป็น Internal Hardware Interupt ความเที่ยงตรงของเวลาขึ้นอยู่กับความถี่ของคริสตอลและค่าที่ Load ใน TIMER/EVENT COUNTER ในโครงงานนี้ใช้เพื่อจับเวลาช่วงสั้นๆ จึงไม่เน้นความเที่ยงตรงมากนัก

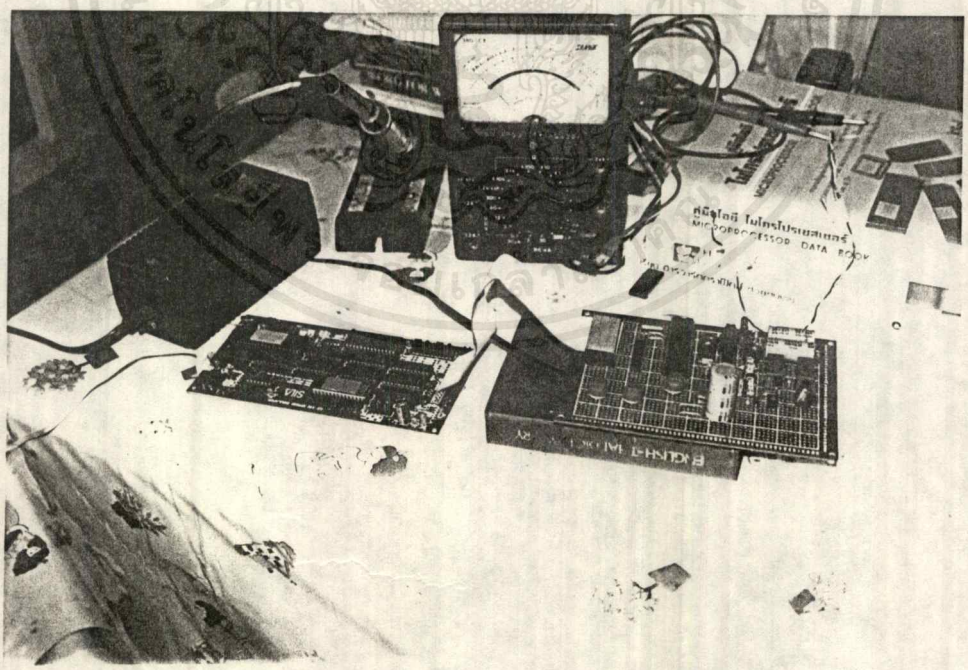
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



การทดลองโครงงานฯ โดยใช้ EPROM Emulator ร่วมกับ PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





เครื่องต้นแบบที่สร้างขึ้นและการ SETUP เพื่อการทดลอง

และการออกแบบ SOFTWARE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ATTENUATION AND AMPLITUDE VERSUS FREQUENCY RESPONSE

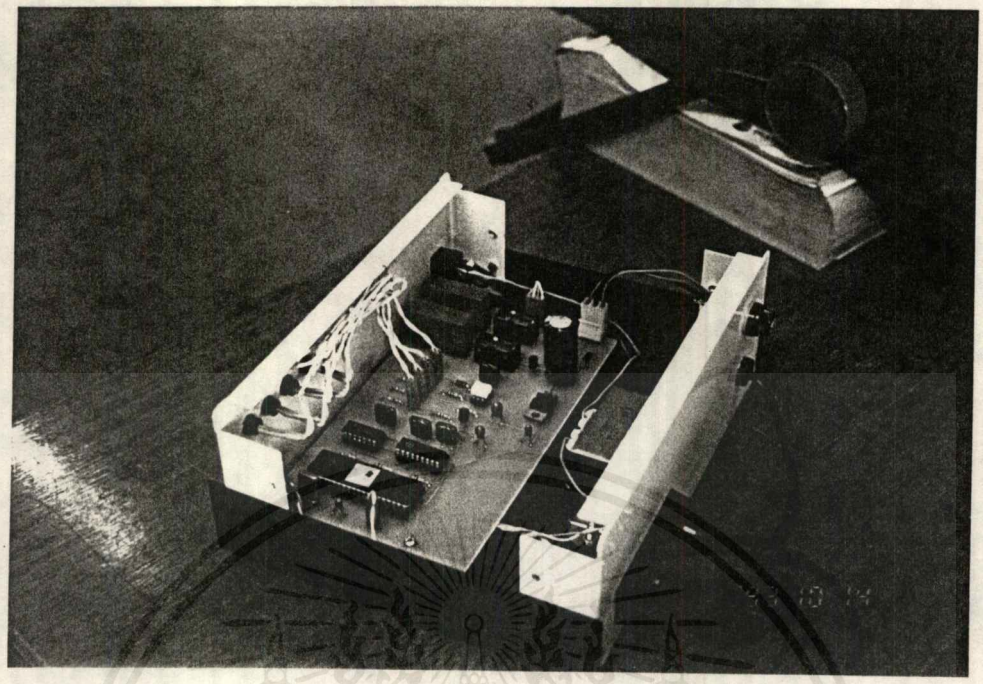
ความสูญเสียของสัญญาณที่ส่งผ่านวงจรเชื่อมโยง และการตอบสนองต่อความถี่เป็น Parameter ที่สำคัญโครงงานนี้

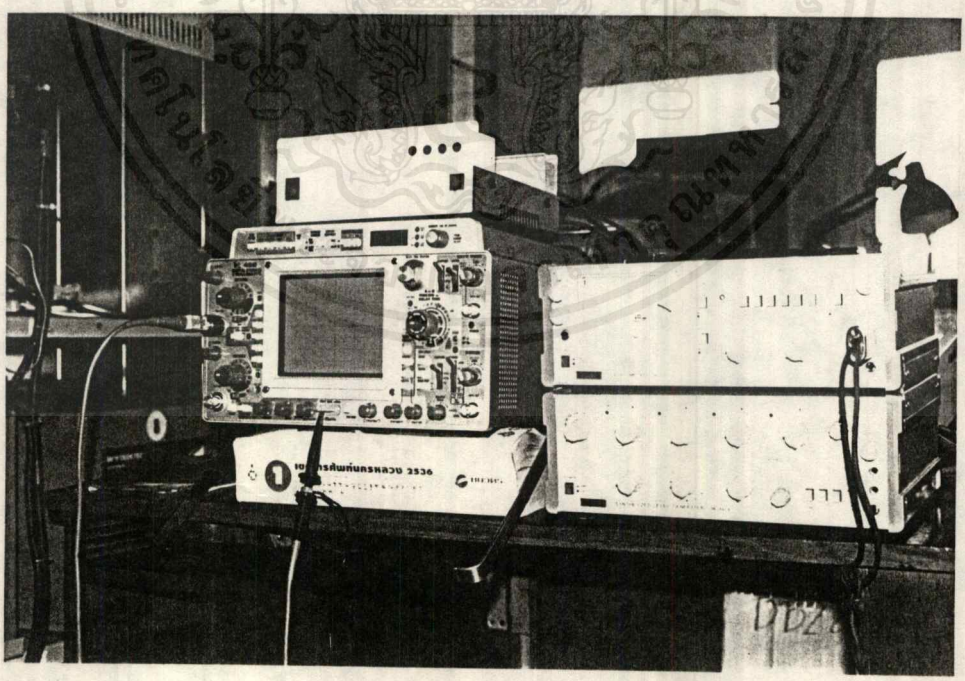
การทดสอบทำได้โดยการส่ง ความถี่เสียงย่านใช้งานของช่องสัญญาณโทรศัพท์ทั่วไปคือความถี่ 300 Hz ถึง 3400 Hz ผ่านวงจรเชื่อมโยง (ป้อนเข้าที่ขั้วต่อ Line 1) แล้ววัดสัญญาณที่ทางออก (Line 2) ด้วย LEVEL METER ที่มีความต้านทานภายใน 600 โอห์ม Terminated ในการทดลองจะป้อนสัญญาณที่มี LEVEL 0 dBm ความถี่ต่างๆแล้วบันทึกค่าที่อ่านจาก LEVEL METER ดังตารางข้างล่าง

ความถี่ (Hz)	สัญญาณที่ทางออก (dBm)
300	-4.9
500	-4.9
700	-4.7
1000	-4.5
1500	-4.4
2000	-4.3
2500	-4.3
3000	-4.3
3400	-4.3

สรุปผลการวัดค่าจากตาราง ที่ความถี่ 1 KHz (ซึ่งเป็นความถี่กึ่งกลางย่านของช่องสัญญาณโทรศัพท์มาตราฐาน และถือว่าความถี่นี้เป็นความถี่สำหรับการปรับเทียบ เราเรียกความถี่นี้ว่า TEST TONE) มีความสูญเสียในการส่งผ่านสัญญาณ (LOSS หรือ ATTENUATION) 4.5 dB ความเบี่ยงเบน (Diviation) หรือความราบเรียบ (Flatness) 0.6 dB (ดู Chart แสดง คุณลักษณะ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



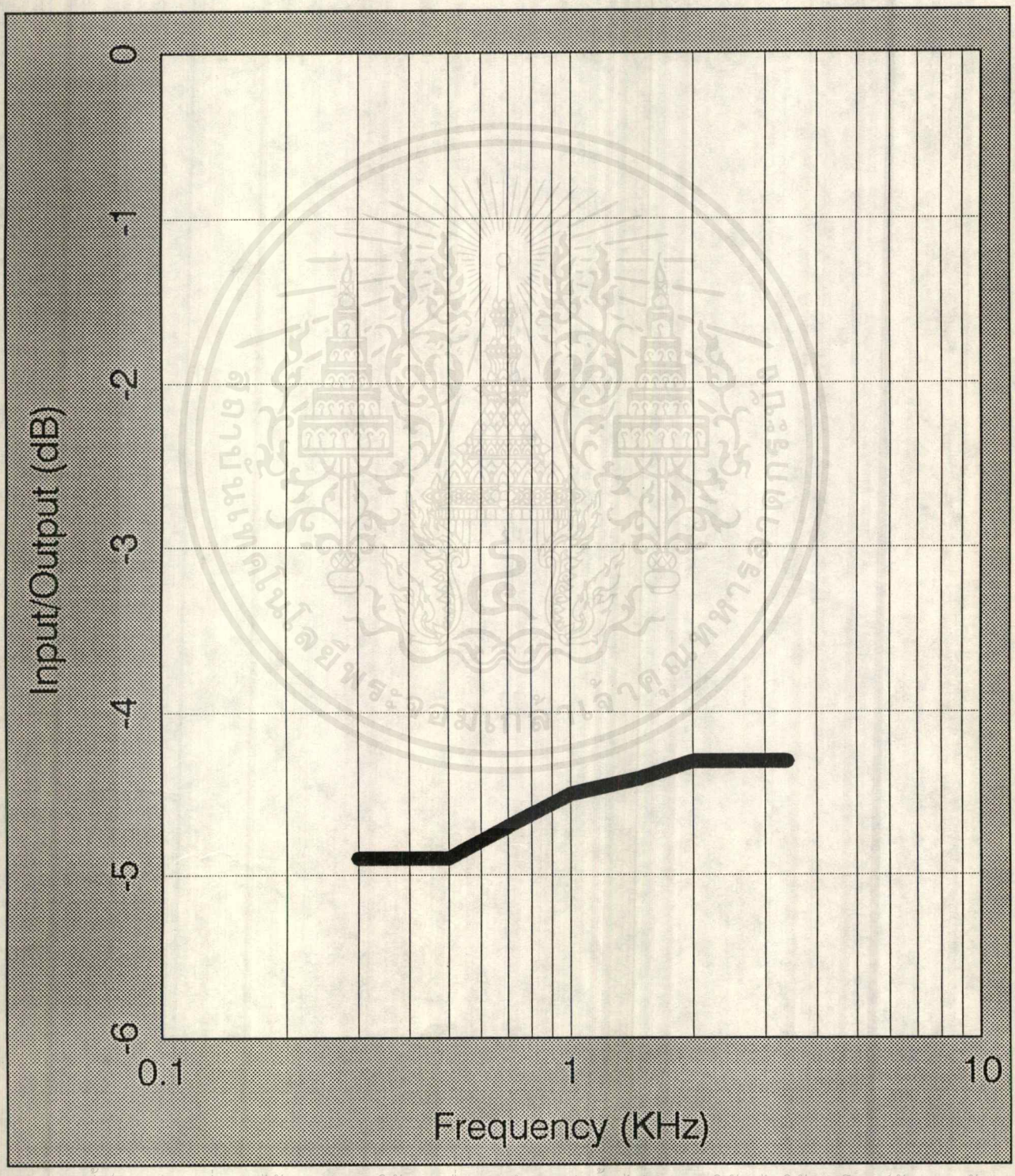


เครื่องเชื่อมโยงสัญญาณโทรศัพท์อัตโนมัติในการทดสอบทดสอบขั้นสุดท้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Amplitude Characteristics

AUTOMATIC TELEPHONE PATCHER



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปผล

1 ถ้าพิจารณาจากจำนวนไอซีและอุปกรณ์ที่ใช้ โครงงานนี้จะถูกพิจารณาว่าเป็นโครงงานขนาดเล็ก แต่ถ้าจะพิจารณาจากหน้าที่การทำงาน ถ้าโครงงานนี้ออกแบบสร้างขึ้น โดยอาศัยวงจรคอมไบเนชั่นลอจิกและซีเควนเชี่ยลลอจิก จะต้องใช้วงจรเกทและฟลิบฟลอบจำนวนมาก วงจรจะมีความซับซ้อน และยากต่อการแก้ไขหรือดัดแปลง ในกรณีที่ต้องการเปลี่ยนแปลงหรือเพิ่มเติมหน้าที่การทำงาน

ผู้จัดทำโครงงานได้ศึกษาเปรียบเทียบและเลือกเอาวิธีที่ใช้ไมโครคอมพิวเตอร์มาเป็นตัวควบคุม และออกแบบซอฟท์แวร์ให้มีความสามารถเพียงพอต่อความต้องการใช้งานอย่างไรก็ตาม ผู้ใช้แต่ละรายอาจมีความคิดเห็นและความต้องการ ในการใช้งานที่แตกต่างกันออกไป โครงงานที่ใช้ไมโครคอมพิวเตอร์เป็นตัวควบคุมจะมีโครงสร้างที่เป็นมาตราฐาน และสามารถรองรับความต้องการ ในการเปลี่ยนแปลงหน้าที่การทำงานได้เป็นอย่างดี

2 หลังจากสร้างชิ้นงานเสร็จได้ติดตั้งทดลองใช้งานจริงปรากฏผลเป็นที่น่าพอใจ

3 ระยะเวลาปฏิบัติงานตามแผน สามารถควบคุมการดำเนินการตามแผนงานได้เป็นอย่างดี

แนวทางพัฒนาต่อ

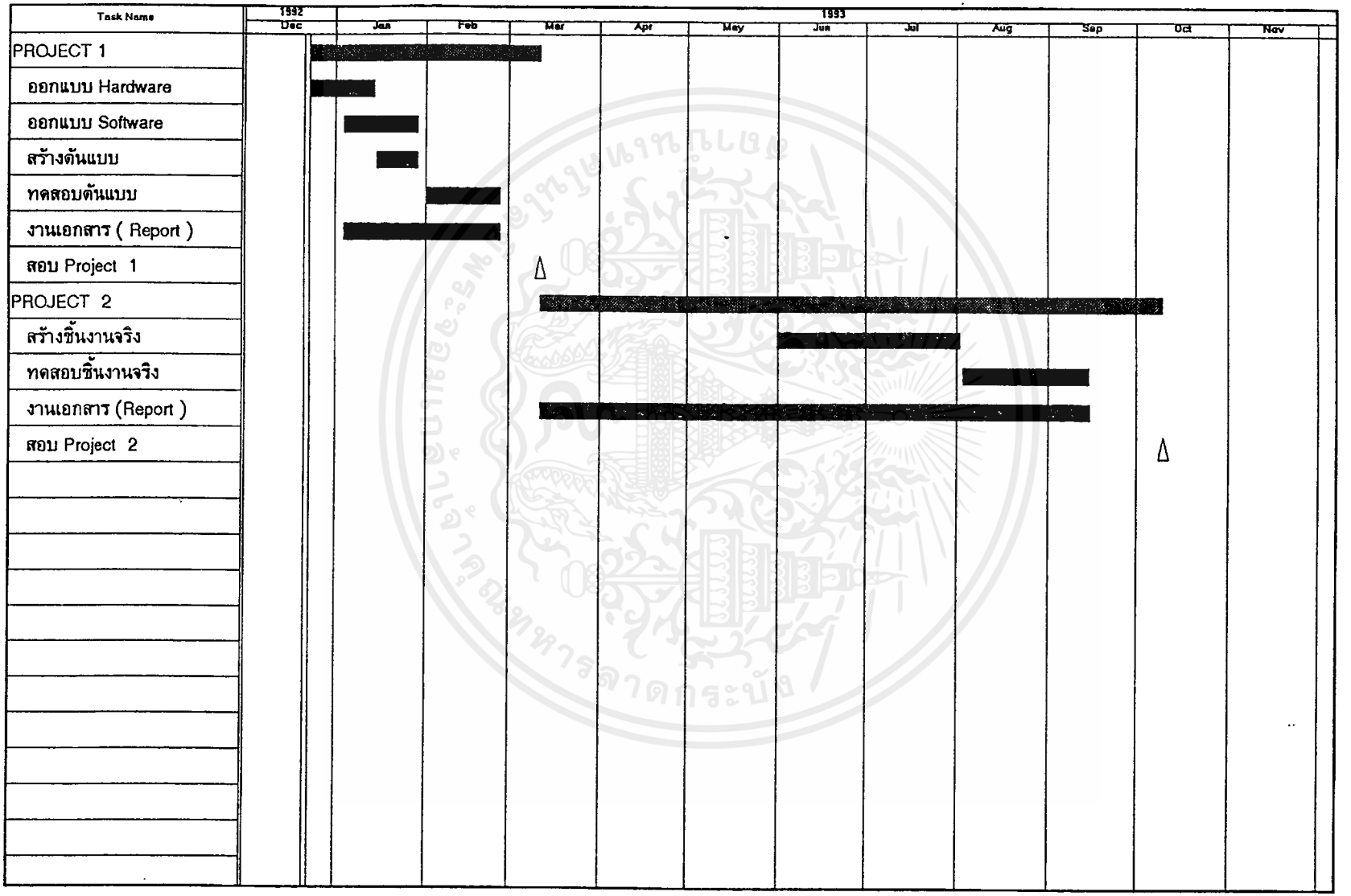
-แก้ไขโปรแกรมให้สามารถตั้งจำนวนครั้งของสัญญาณกระดิ่ง หรือ ตั้งระยะเวลาพูดได้

-สามารถใช้ I/O Port ที่เหลืออยู่ต่อกับวงจรขับ RELAY หรือ Sensor เพื่อใช้เป็น Remote Control หรือ Remote Sensing ได้

-เพิ่มเติมส่วนที่เป็นเครื่องช่วยหมุนโทรศัพท์อัตโนมัติ (Automatic Telephone Dialer) แล้วนำไปใช้ร่วมกับระบบวิทยุติดตามตัว (Pager) เพื่อใช้เป็นระบบแจ้งเหตุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTOMATIC TELEPHONE PATCHER

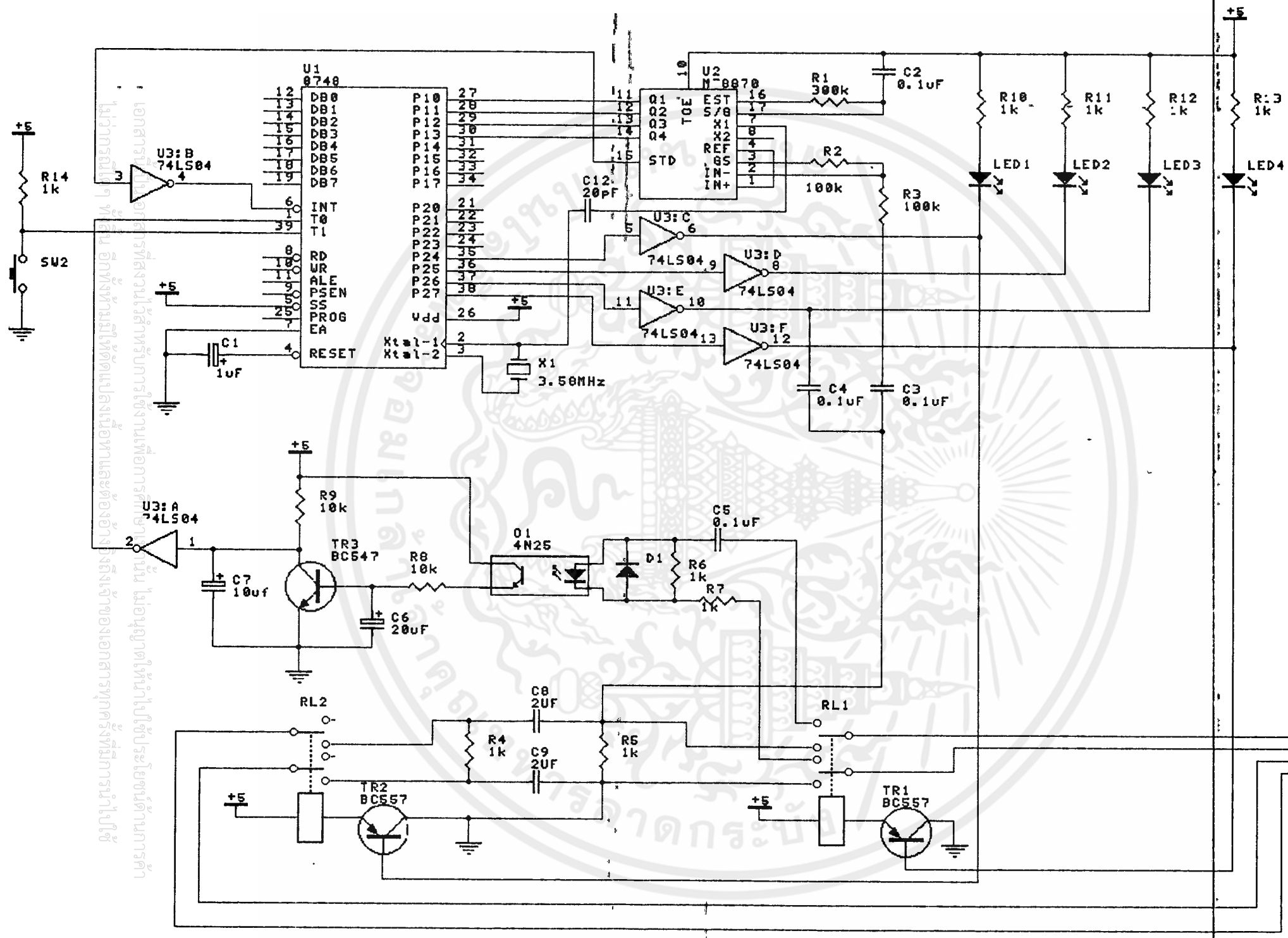


Printed: Dec/23/92
Page 1

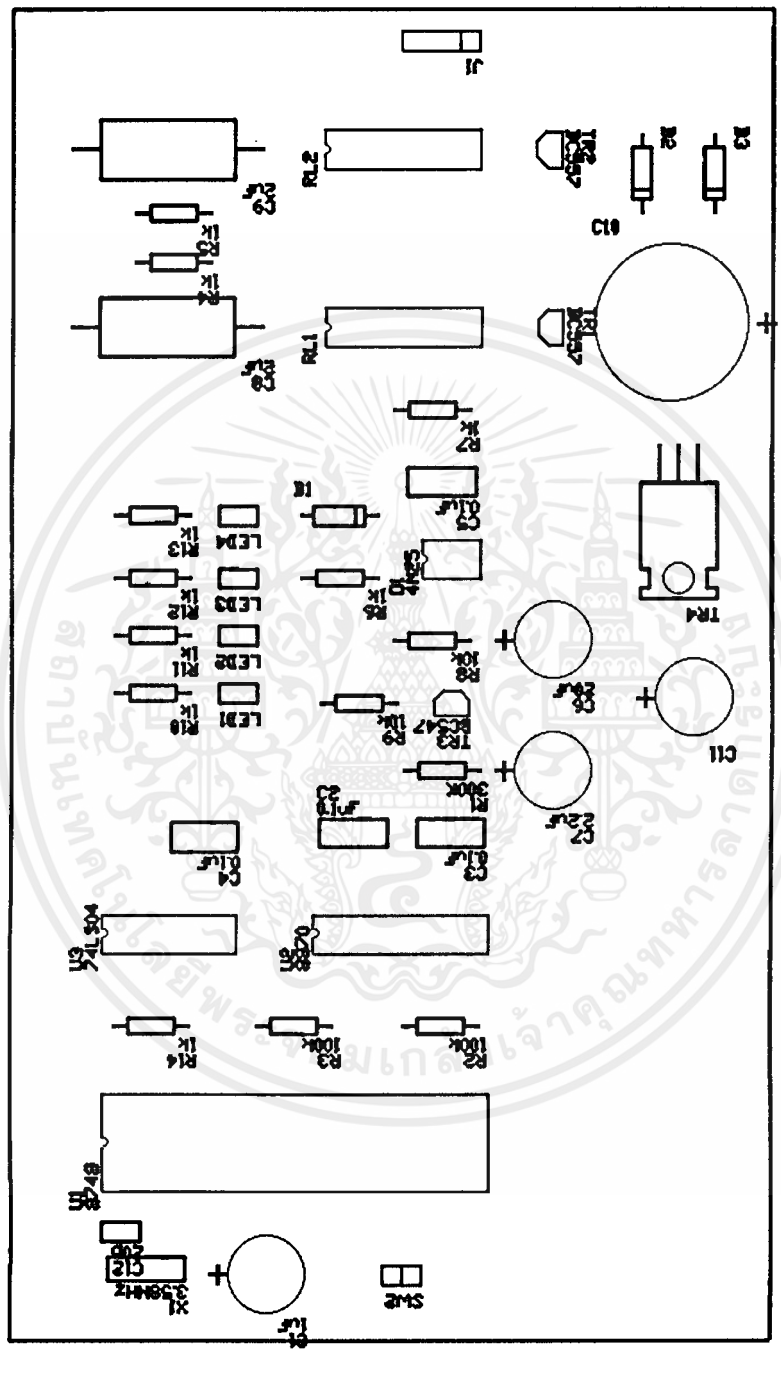
ผนวก ก

เครื่อง เชื่อมโยงสัญญาณโทรศัพท์อัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



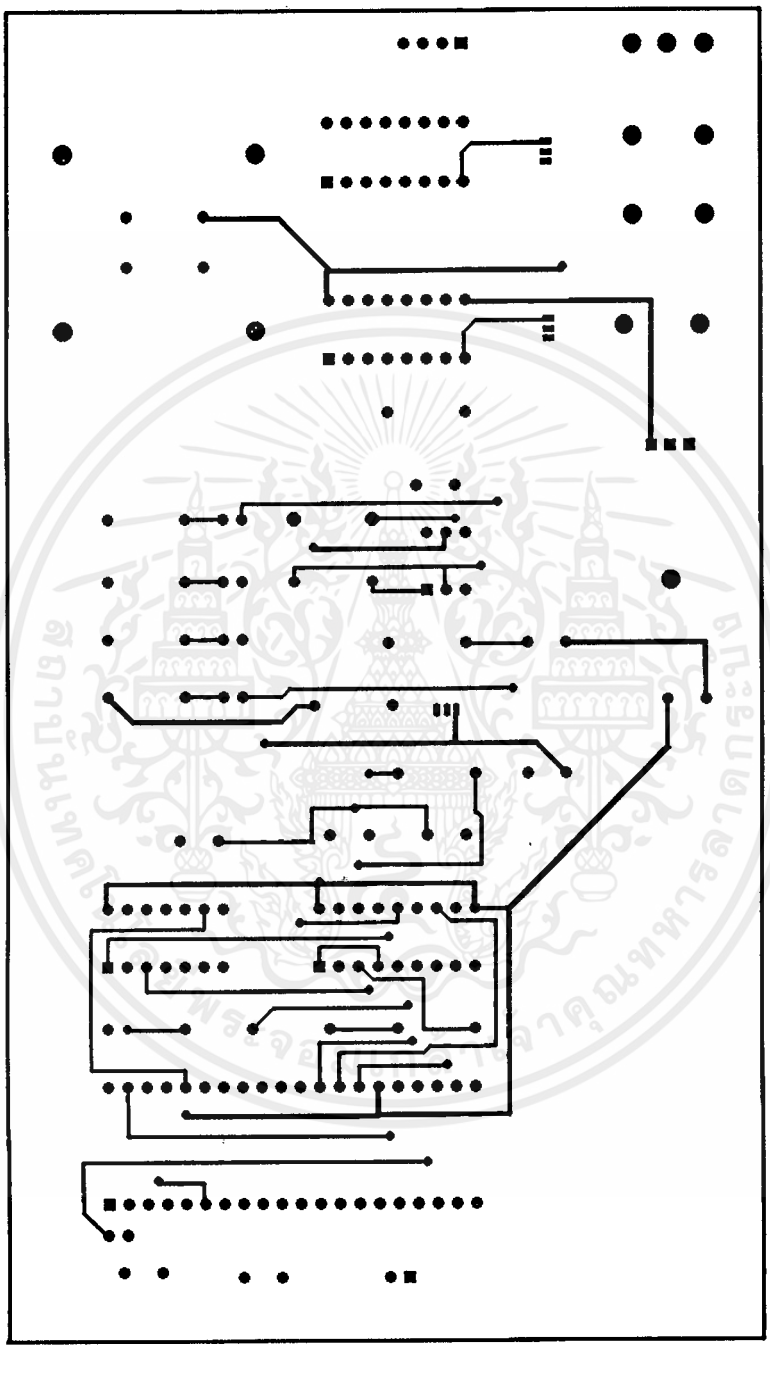
Title	AUTOMATIC TELEPHONE PATCHER		
Size A3	Number		Revision /
Date: 29-SEP 1993		Sheet 1 of 1	
File: PATCH3/1		Drawn By: TT.2U	



รูปแผ่นพิมพ์วงจรเครื่องเชื่อมโยงสัญญาณโทรศัพท์ด้านบน

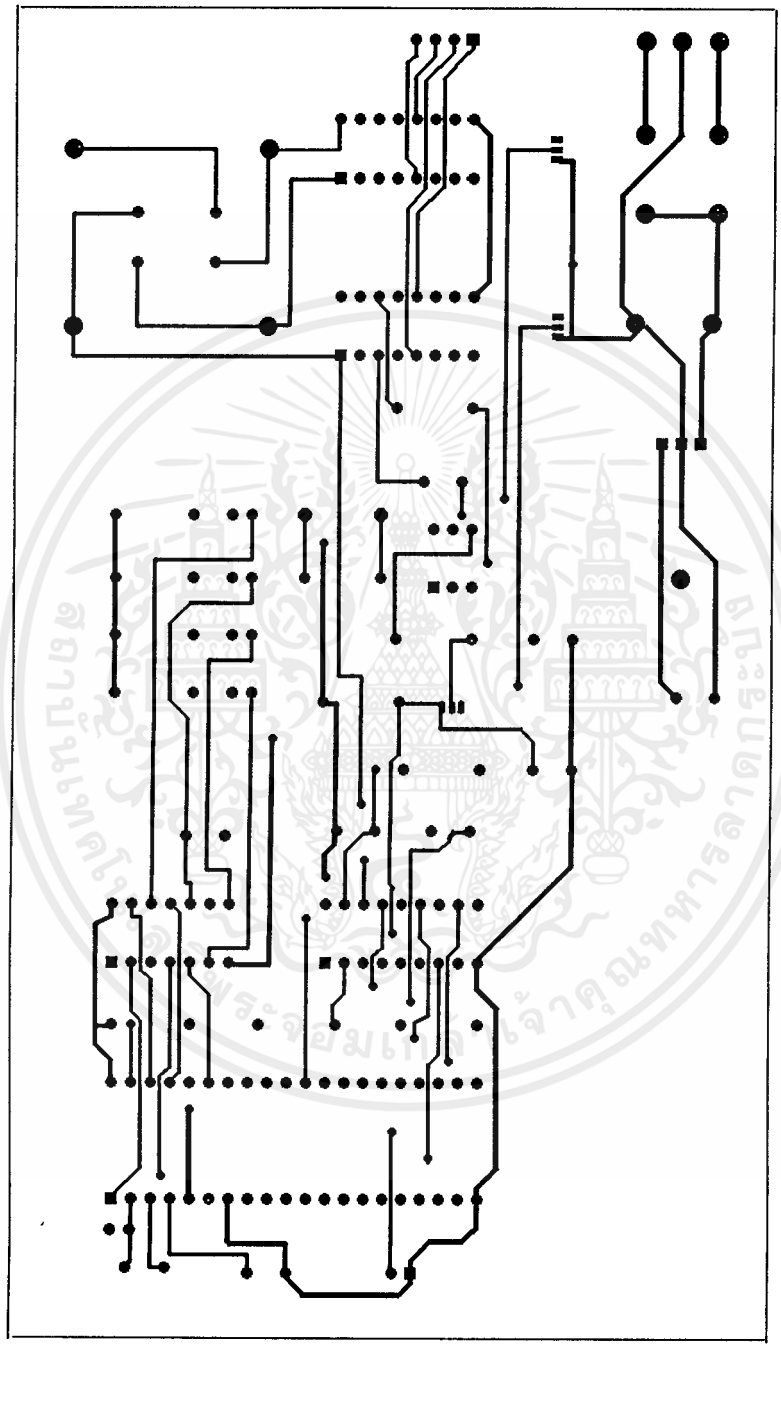
แสดงการวางอุปกรณ์ (Top Layer)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแผ่นพิมพ์วงจรเครื่องเชื่อมโยงสัญญาณโทรศัพท์ด้านบน (Component Side)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแผ่นพิมพ์วงจรเครื่องเชื่อมโยงสัญญาณโทรศัพท์ด้านล่าง (Solder Side)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bill of Material Detail

DESIGNATOR	PATTERN	COMMENT
C1	RB.2/.4	1UF
C2	RAD0.2	0.1UF
C3	RAD0.2	0.1UF
C4	RAD0.2	0.1UF
C5	RAD0.2	0.1UF
C6	RB.2/.4	20UF
C7	RB.2/.4	2.2UF
C8	AXIAL1.0	2UF
C9	AXIAL1.0	2UF
C10	RB.4/.8	1000UF
C11	RB.2/.4	2UF
C12	RAD0.1	20P
D1	DIODE0.4	1N4001
D2	DIODE0.4	1N4001
D3	DIODE0.4	1N4001
J1	SIP4	
LED1	RAD0.1	
LED2	RAD0.1	
LED3	RAD0.1	
LED4	RAD0.1	
O1	DIP6	4N25
R1	AXIAL0.4	300K
R2	AXIAL0.4	100K
R3	AXIAL0.4	100K
R4	AXIAL0.4	1K
R5	AXIAL0.4	1K
R6	AXIAL0.4	1K
R7	AXIAL0.4	1K
R8	AXIAL0.4	10K
R9	AXIAL0.4	10K
R10	AXIAL0.4	1K
R11	AXIAL0.4	1K
R12	AXIAL0.4	1K
R13	AXIAL0.4	1K
R14	AXIAL0.4	1K
RL1	DIP16	DPDT 5 VDC
RL2	DIP16	DPDT 5 VDC
SW2	SIP2	
TR1	TO-92B	BC557
TR2	TO-92B	BC557
TR3	TO-92B	BC547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TR4	TO-220	7805
U1	DIP40	8748
U2	DIP18	8870
U3	DIP14	74LS04
X1	XTAL1	3.58MHZ

Total Components = 46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        ;*********************************
                        ;* AUTOMATIC TELEPHONE PATCHER *
                        ;*********************************
0000                              CPU   "8048.TBL"
0000                              HOF   "INT8"
                        ;
                        ;KEY   FUNCTION
                        ; *    DISCONNECT
                        ; #    CONTINUE
                        ;
0020 =                  KBDBUF:   EQU   20H             ;DTMF RECEIVER BUFFER
0021 =                  ERRBUF:   EQU   21H             ;FOR PASSWORD CHECK
0022 =                  TMEBUF:   EQU   22H             ;3 BYTES TIME BUFFER
0025 =                  WORD1:    EQU   TMEBUF +3       ;3 BTYES PASSWORD
                        ;
0000 05                 START:    EN    I
0001 2400                         JMP   INIT            ;SET VALUE TO MAIN
                        ;
0003 AA                 INT:      MOV   R2,A            ;INTERUPT ENTRY
0004 09                           IN    A,P1            ;DTMF RECEIVER
0005 042E                         JMP   KBD             ;SKIP OVER TIMER INT.
                        ;
0007 D5                 CLOCK:    SEL   RB1             ;SAVE REGISTER
0008 AF                           MOV   R7,A            ;A TO R7
0009 B822                         MOV   R0,#TMEBUF      ;HUNDRED
000B F0                           MOV   A,@R0
000C 0302                         ADD   A,#2
000E A0                           MOV   @R0,A
000F D364                         XRL   A,#100
0011 9625                         JNZ   CLKE
0013 B000                         MOV   @R0,#0
0015 18                           INC   R0              ;SECOND
0016 F0                           MOV   A,@R0
0017 0301                         ADD   A,#1
0019 A0                           MOV   @R0,A
001A D33C                         XRL   A,#60
001C 9625                         JNZ   CLKE
001E B000                         MOV   @R0,#0
0020 18                           INC   R0
0021 F0                           MOV   A,@R0           ;MINUTE
0022 0301                         ADD   A,#1
0024 A0                           MOV   @R0,A
0025 235B               CLKE:     MOV   A,#5BH          ;COUNT
0027 62                           MOV   T,A
0028 FF                           MOV   A,R7            ;RESTORE A
0029 C5                           SEL   RB0
002A 25                           EN    TCNTI
002B 55                           STRT  T
002C 05                           EN    I
002D 93                           RETR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        ;*************************************
                        ;INTERUPT ROUTINE FOR DTMF DETECTOR
                        ;STORE KEY VALUE IN KBDBUF
                        ;F1 IS CLEARED
                        ;*************************************
                        ;
002E 530F       KBD:        ANL   A,#0FH
0030 B920                   MOV   R1,#KBDBUF
0032 A1                     MOV   @R1,A
0033 FA                     MOV   A,R2
0034 A5                     CLR   F1
0035 25                     EN    TCNTI
0036 55                     STRT  T
0037 05                     EN    I
0038 93                     RETR
                        ;
                        ;*************************************
                        ;TONE GENERATOR
                        ;*************************************
                        ;
0039 BD02       BEEP:       MOV    R5,#2H
003B BEAA       BEEP1:      MOV    R6,#0AAH
003D 8A40       BEEP2:      ORL    P2,#40H
003F BF40                   MOV    R7,#40H
0041 EF41       ON:         DJNZ   R7,ON
0043 9AB0                   ANL    P2,#0B0H
0045 BF40                   MOV    R7,#40H
0047 EF47       OFF:        DJNZ   R7,OFF
0049 EE3D                   DJNZ   R6,BEEP2
004B ED3B                   DJNZ   R5,BEEP1
                        ;
004D BD02       DELAY:      MOV   R5,#2
004F BFAA       DL1:        MOV   R7,#0AAH
0051 BE00       DL2:        MOV   R6,#00
0053 EE53       DL3:        DJNZ  R6,DL3
0055 EF51                   DJNZ  R7,DL2
0057 ED4F                   DJNZ  R5,DL1
0059 83                     RET
                        ;
                        ;*************************************
                        ;FLASH ROUTINE
                        ;HOOK ON 1 SEC. THEN HOOK OFF
                        ;*************************************
                        ;
005A 9A7F       FLASH:      ANL   P2,#7FH
005C 144D                   CALL  DELAY
005E 8A80                   ORL   P2,#80H
0060 83                     RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        ;*************************************
                        ;PASSWORD ENTER ROUTINE
                        ;RECEIVE 3 PASSWORDS
                        ;IF FAILED TO RECEIVE
                        ;STORE "* * *" AS DEFAULT
                        ;*************************************
                        ;
0061 A5         INITWORD: CLR   F1
0062 B5                   CPL   F1              ;PREPARE KBD FLAG
0063 BC0A                 MOV   R4,#10          ;10 BEEPS
0065 BB03                 MOV   R3,#3           ;3 PASSWORD
0067 B920                 MOV   R1,#KBDBUF
0069 B825                 MOV   R0,#WORD1
006B 1439       INITW1:   CALL  BEEP
006D EC7A                 DJNZ  R4,INITW2
006F B825                 MOV   R0,#WORD1
0071 230B                 MOV   A,#11           ;"*"
0073 A0                   MOV   @R0,A           ;DEFAULT 1
0074 18                   INC   R0
0075 A0                   MOV   @R0,A           ;DEFAULT 2
0076 18                   INC   R0
0077 A0                   MOV   @R0,A           ;DEFAULT 3
0078 0482                 JMP   INITWE
007A 766B       INITW2:   JF1   INITW1
007C B5                   CPL   F1
007D F1                   MOV   A,@R1
007E A0                   MOV   @R0,A
007F 18                   INC   R0
0080 EB6B                 DJNZ  R3,INITW1
0082 83         INITWE:   RET
                        ;
                        ;*************************************
                        ;PASSWORD CHECK ROUTINE
                        ;F0 SET IF FAILED
                        ;*************************************
                        ;
0083 85         PASSWORD: CLR   F0              ;PREPARE FLAG
0084 B825                 MOV   R0,#WORD1       ;CHECK PASSWORD
0086 2300                 MOV   A,#0            ;IF PASSWORD
0088 D0                   XRL   A,@R0           ;WERE NOT SET
0089 C6B5                 JZ    PASSEXIT        ;JUMP EXIT
008B BC0A                 MOV   R4,#10          ;10 BEEPS
008D BB03                 MOV   R3,#3           ;3 PASSWORD
008F B921                 MOV   R1,#ERRBUF      ;CLEAR ERROR
0091 B100                 MOV   @R1,#0
0093 B920       PASSW1:   MOV   R1,#KBDBUF
0095 1439                 CALL  BEEP
0097 EC9B                 DJNZ  R4,PASSW2
0099 04B4                 JMP   ERREXIT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
009B 7693       PASSW2:   JF1   PASSW1
009D B5                   CPL   F1
009E 230D                 MOV   A,#13
00A0 D1                   XRL   A,@R1
00A1 C6B5                 JZ    PASSEXIT
00A3 F1                   MOV   A,@R1
00A4 D0                   XRL   A,@R0
00A5 C6AA                 JZ    PASSW3
00A7 B921                 MOV   R1,#ERRBUF
00A9 11                   INC   @R1
00AA 18         PASSW3:   INC   R0
00AB EB93                 DJNZ  R3,PASSW1
00AD B921                 MOV   R1,#ERRBUF
00AF 2300                 MOV   A,#0
00B1 D1                   XRL   A,@R1
00B2 C6B5                 JZ    PASSEXIT
00B4 95         ERREXIT:  CPL   F0
00B5 83         PASSEXIT: RET
                ;
                ;*************************************
                ;LINK ROUTINE
                ;RETURN IF KEY " * " IS PRESSED
                ;OR TIMEOUT
                ;*************************************
                ;
00B6 B824       LINK:     MOV   R0,#TMEBUF+2
00B8 B000                 MOV   @R0,#0
00BA F0         HERE:     MOV   A,@R0
00BB D303                 XRL   A,#3
00BD C6E2                 JZ    TONE
00BF 76BA                 JF1   HERE
00C1 B5         CHK:      CPL   F1
00C2 B920                 MOV   R1,#KBDBUF
00C4 230E                 MOV   A,#14           ;FLASH
00C6 D1                   XRL   A,@R1
00C7 96CD                 JNZ   CHK1
00C9 145A                 CALL  FLASH
00CB 04B6                 JMP   LINK
00CD 230D       CHK1:     MOV   A,#13           ;PASSWORD SET
00CF D1                   XRL   A,@R1
00D0 96D6                 JNZ   CHK2
00D2 1461                 CALL  INITWORD
00D4 04B6                 JMP   LINK
00D6 230C       CHK2:     MOV   A,#12           ;CONTINUE
00D8 D1                   XRL   A,@R1
00D9 C6B6                 JZ    LINK
00DB 230B                 MOV   A,#11           :DISCONNECT
00DD D1                   XRL   A,@R1
00DE C6F9                 JZ    LINKEXIT
00E0 04BA                 JMP   HERE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
00E2 BC0A       TONE:     MOV  R4,#10          ;10 BEEPS
00E4 1439       TONE1:    CALL BEEP
00E6 ECEA                 DJNZ R4,CHKE
00E8 04F9                 JMP  LINKEXIT
00EA 76E4       CHKE:     JF1  TONE1
00EC B5                   CPL  F1
00ED B920                 MOV  R1,#KBDBUF
00EF 230C                 MOV  A,#12           ;CONTINUE
00F1 D1                   XRL  A,@R1
00F2 C6B6                 JZ   LINK
00F4 230B                 MOV  A,#11           ;DISCONNECT
00F6 D1                   XRL  A,@R1
00F7 96E4                 JNZ  TONE1
00F9 83         LINKEXIT: RET
                ;
                ;**************************************
                ;
0100                      ORG  100H
0100 BF64       INIT:     MOV  R7,#64H         ;POWER ON DELAY
0102 BE00       IT1:      MOV  R6,#00H         ;ABOUT 200 MSEC.
0104 EE04       ITT:      DJNZ R6,ITT
0106 EF02                 DJNZ R7,IT1
0108 B820                 MOV  R0,#20H         ;CLEAR BUFFER
010A BF0F                 MOV  R7,#0FH         ;15 BYTES
010C B000       IT2:      MOV  @R0,#00H
010E 18                   INC  R0
010F EF0C                 DJNZ R7,IT2
0111 2300                 MOV  A,#00
0113 D7                   MOV  PSW,A
0114 25                   EN   TCNTI
0115 55                   STRT T
                ;
0116 2310                 MOV  A,#10H
0118 3A                   OUTL P2,A
0119 1461                 CALL INITWORD
011B 9A00       MAIN:     ANL  P2,#00H         ;IDLE
011D 462C       CHKMNL:   JNT1 CONNECT         ;JUMP IF T0 = 0
011F 261D                 JNT0 CHKMNL          :LOOP IF NOT RINGING
0121 3433                 CALL RGCHK
0123 B61B                 JF0  MAIN
0125 2310                 MOV  A,#10H          ;ANSWER A CALL
0127 3A                   OUTL P2,A
0128 1483                 CALL PASSWORD
012A B61B                 JF0  MAIN
012C 2390       CONNECT:  MOV  A,#90H          ;CONNECT
012E 3A                   OUTL P2,A
012F 14B6                 CALL LINK
0131 241B                 JMP  MAIN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        ;*************************************
                        ;CHECK RINGING ROUTINE
                        ;F0 SET IF FAILED
                        ;*************************************
                        ;
0133 85         RGCHK:      CLR   F0
0134 BB0A                   MOV   R3,#10          ;SET COUNTER
0136 2360       RING:       MOV   A,#60H          ;
0138 3A                     OUTL  P2,A            ;SHOW RINGING ON
0139 3639       RING1:      JT0   RING1           ;WAIT FOR STOP
013B 2340                   MOV   A,#40H          ;
013D 3A                     OUTL  P2,A            ;SHOW RINGING OFF
013E B822                   MOV   R0,#TMEBUF      ;CLEAR TIME BUFFER
0140 B000                   MOV   @R0,#0          ;HUNDRED
0142 18                     INC   R0
0143 B000                   MOV   @R0,#0          ;SECOND
0145 EB49                   DJNZ  R3,TOUT
0147 2451                   JMP   RGEXIT
0149 3636       TOUT:       JT0   RING
014B F0                     MOV   A,@R0
014C D30A                   XRL   A,#10           ;TIME OUT 10 SEC.
014E 9649                   JNZ   TOUT
0150 95                     CPL   F0
0151 83         RGEXIT:     RET
                ;
0000                        END


0039  BEEP                003B  BEEP1               003D  BEEP2
00C1  CHK                 00CD  CHK1                00D6  CHK2
00EA  CHKE                011D  CHKMNL              0025  CLKE
0007  CLOCK               012C  CONNECT             004D  DELAY
004F  DL1                 0051  DL2                 0053  DL3
0021  ERRBUF              00B4  ERREXIT             005A  FLASH
00BA  HERE                0100  INIT                006B  INITW1
007A  INITW2              0082  INITWE              0061  INITWORD
0003  INT                 0102  IT1                 010C  IT2
0104  ITT                 002E  KBD                 0020  KBDBUF
00B6  LINK                00F9  LINKEXIT            011B  MAIN
0047  OFF                 0041  ON                  00B5  PASSEXIT
0093  PASSW1              009B  PASSW2              00AA  PASSW3
0083  PASSWORD            0133  RGCHK               0151  RGEXIT
0136  RING                0139  RING1               0000  START
0022  TMEBUF              00E2  TONE                00E4  TONE1
0149  TOUT                0025  WORD1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งาน เครื่องเชื่อมโยงสัญญาณโทรศัพท์อัตโนมัติ

INSTRUCTION MANUAL

ลักษณะทั่วไป (General)

เครื่องเชื่อมโยงสัญญาณโทรศัพท์อัตโนมัติ

ด้านหน้า มีสวิทช์ POWER ON-OFF อยู่ทางซ้ายมือ

มี MANUAL SWITCH สำหรับควบคุมการเชื่อมโยงด้วยมือ

มี LED 4 ดวง (L1 - L4) เพื่อแสดงสถานะการทำงาน

ด้านหลัง มีสายไฟ AC และกลักฟิวส์

มีขั้วต่อสายโทรศัพท์ 2 คู่ คือ LINE 1 และ LINE 2

การติดตั้ง (Installation)

ต่อ LINE 1 กับคู่สายของเลขหมายที่ผู้ใช้จะต้องเรียกเข้า หรือเบอร์ NON CHARGE

ต่อ LINE 2 กับอีกคู่สายหนึ่งที่เป็นเลขหมายชนิด DTMF

ต่อ เสียบปลั๊ก AC เข้ากับเต้าเสียบ

การเตรียมการ (Preparation)

เมื่อเปิดสวิทช์ POWER เครื่องเชื่อมโยงสัญญาณฯ จะส่งสัญญาณเตือนและรอรับการตั้งรหัสผ่าน (LED ที่ด้านหน้าของเครื่องจะแสดงสถานะการรอรับรหัสผ่านด้วย) ผู้ที่ติดตั้งจะต้องยกหูโทรศัพท์เครื่องที่ต่อพ่วงอยู่กับ LINE 1 ขึ้น แล้วกดปุ่มหมายเลข 3 หมายเลขภายใน 10 วินาที เป็นการตั้งรหัสผ่าน ถ้ากดไม่ครบ 3 หมายเลขหรือไม่กดเครื่องค่าปริยาย (Default) เป็น * * * โดยอัตโนมัติ ถ้ายังต้องการตั้งรหัสให้ปิดสวิทช์แล้วเปิดใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งาน (Operation)

เมื่อผ่านขั้นเตรียมการ เครื่องฯก็พร้อมที่จะทำงาน ผู้ใช้งานเครื่องเชื่อมโยงสัญญาณฯ ทำตามขั้นตอนต่อไปนี้

การใช้งานอัตโนมัติ

1 ใช้โทรศัพท์สาธารณะหรือโทรศัพท์แบบกดปุ่มทั่วไปเรียกเข้ายังเลขหมายที่ต่ออยู่กับ LINE 1
2 ฟังสัญญาณเรียกครบ 10 ครั้ง
3 เมื่อได้รับสัญญาณเตือนให้กดรหัสผ่านตามที่ตั้งไว้ 3 หมายเลข
4 ถ้ารหัสผ่านถูกต้องจะได้รับ Dial Tone จากอีกคู่สายหนึ่งที่ต่ออยู่กับ LINE 2
5 กดปุ่มเลขหมายปลายทางที่ต้องการติดต่อเหมือนการใช้โทรศัพท์ทั่วไป
6 ถ้าเลขหมายปลายทางที่ต้องการติดต่อด้วยไม่ว่างให้กดปุ่ม * วางหูลงแล้วกลับไปเริ่มข้อ 1 ถ้าผู้ใช้วางหูโดยไม่ได้กด * จะต้องรอเวลา 3 นาทีเพื่อให้เครื่องฯ กลับคืนไปอยู่ในสภาวะพร้อม
7 ถ้าว่างและมีผู้รับสายผู้ใช้จะสนทนาได้ 3 นาที
8 ถ้าต้องการเลิกสนทนาก่อนครบ 3 นาที ให้กดปุ่ม * แล้ววางหู
9 เมื่อครบ 3 นาทีเครื่องจะส่งสัญญาณเตือน 10 วินาทีก่อนตัดการเชื่อมโยงระหว่างนี้ผู้ใช้สามารถต่อเวลาพูดได้ครั้งละ 3 นาที โดยกดปุ่ม #

การใช้งานแบบต่อด้วย MANUAL SWITCH

เมื่อผู้ใช้เรียกเข้ามายังหมายเลขที่ต่ออยู่กับ LINE 1 และมีผู้ยกหูโทรศัพท์เครื่องที่ต่อพ่วงอยู่กับ LINE 1 ก่อนที่กระดิ่งจะดังครบ 10 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามข้อ 2 ของการใช้งานอัตโนมัติ ผู้ใช้สามารถสนทนากับผู้ที่ยกหูได้ตามปกติ ดังนั้นผู้ใช้สามารถบอกให้ผู้ที่ยกหูกด MANUAL SWITCH ให้ได้และจะเกิดการเชื่อมโยงขึ้นโดยข้ามขั้นตอนของการป้อนรหัสผ่านไป ผู้ใช้จะได้ Dial Tone และปฏิบัติต่อไปตามข้อ 5 ของการใช้งานอัตโนมัติ ผู้ที่ยกหูควรวางหูโทรศัพท์ลงเพื่อไม่ให้เกิดปัญหาในการเรียกเลขหมายปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

8748H/8035H/8749H/8039H
HMOS-E SINGLE-COMPONENT
8-BIT MICROCOMPUTER

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRELIMINARY

8748H/8035H/8749H/8039H
HMOS-E SINGLE-COMPONENT 8-BIT MICROCOMPUTER

- High Performance HMOS-E
- Interval Timer/Event Counter
- Two Single Level Interrupts
- Single 5-Volt Supply
- Over 96 Instructions; 90% Single Byte
- Compatible with 8080/8085 Peripherals
- Easily Expandable Memory and I/O
- Up to 1.35 μSec Instruction Cycle All Instructions 1 or 2 cycles

The Intel 8749H/8039H/8748H/8035H are totally self-sufficient, 8-bit parallel computers fabricated on single silicon chips using Intel's advanced N-channel silicon gate HMOS-E process.

The family contains 27 I/O lines, an 8-bit timer/counter, on-chip RAM and on-board oscillator/clock circuits. For systems that require extra capability, the family can be expanded using MCS®-80/MCS®-85 peripherals.

These microcomputers are designed to be efficient controllers as well as arithmetic processors. They have extensive bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory results from an instruction set consisting mostly of single byte instructions and no instructions over 2 bytes in length.

Device	Internal Memory	
8039H	none	128 x 8 RAM
8035H	none	64 x 8 RAM
8749H	2K x 8 EPROM	128 x 8 RAM
8748H	1K x 8 EPROM	64 x 8 RAM

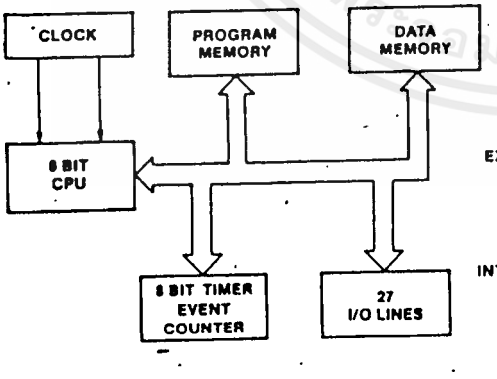


Figure 1.
Block Diagram

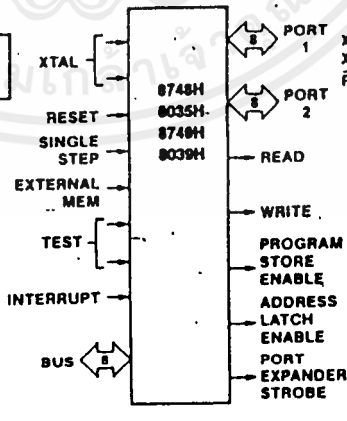


Figure 2.
Logic Symbol

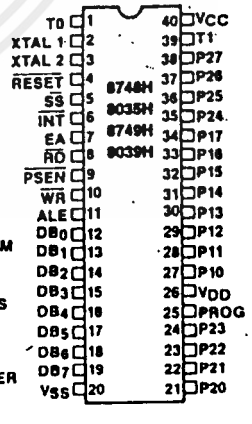


Figure 3.
Pin Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1. Pin Description

Symbol	Pin No.	Function	Device
V_{SS}	20	Circuit GND potential	All
V_{DD}	26	+5V during normal operation.	All
		Programming power supply (+21V).	8748H 8749H
V_{CC}	40	Main power supply; +5V during operation and programming.	All
PROG	25	Output strobe for 8243 I/O expander.	All
		Program pulse (+18V) input pin during programming.	8748H 8749H (See Note)
P10-P17 Port 1	27-34	8-bit quasi-bidirectional port.	All
P20-P23 P24-P27 Port 2	21-24 35-38	8-bit quasi-bidirectional port. P20-P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for 8243.	All
DB0-DB7 BUS	12-19	True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched.	All
(Con't)		Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.	
T0	1	Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENT0 CLK instruction	All
		Used during programming.	8748H 8749H
T1	39	Input pin testable using the JT1, and JNT1 instructions. Can be designated the timer/counter input using the STRT CNT instruction.	All
$\overline{INT}$	6	Interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. (Active low) interrupt must remain low for at least 3 machine cycles for proper operation.	All

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1. Pin Description (Continued)

Symbol	Pin No.	Function	Device
$\overline{RD}$	8	Output strobe activated during a BUS read. Can be used to enable data onto the bus from an external device. Used as a read strobe to external data memory. (Active low)	All
$\overline{RESET}$	4	Input which is used to initialize the processor. (Active low) (Non TTL V_{IH})	All
		Used during programming.	8748H 8749H
$\overline{WR}$	10	Output strobe during a bus write. (Active low) Used as write strobe to external data memory.	All
ALE	11	Address latch enable. This signal occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.	All
$\overline{PSEN}$	9	Program store enable. This output occurs only during a fetch to external program memory. (Active low)	ALL
$\overline{SS}$	5	Single step input can be used in conjunction with ALE to "single step" the processor through each instruction.	All
EA	7	External access input which forces all program memory fetches to reference external memory. Useful for emulation and debug. (Active high)	All
		Used during (18V) programming	8748H 8749H
XTAL1	2	One side of crystal input for internal oscillator. Also input for external source. (Non TTL V_{IH})	All
XTAL2	3	Other side of crystal input.	All

NOTE: On the 8749H/8039H, PROG must be clamped to V_{CC} when not programming. A diode should be used when using an 8243; otherwise, a direct connection is permissible.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 2. Instruction Set

Accumulator

Mnemonic	Description	Bytes	Cycles
ADD A, R	Add register to A	1	1
ADD A, @R	Add data memory to A	1	1
ADD A, # data	Add immediate to A	2	2
ADDC A, R	Add register with carry	1	1
ADDC A, @R	Add data memory with carry	1	1
ADDC A, # data	Add immediate with carry	2	2
ANL A, R	And register to A	1	1
ANL A, @R	And data memory to A	1	1
ANL A, # data	And immediate to A	2	2
ORL A, R	Or register to A	1	1
ORL A @R	Or data memory to A	1	1
ORL A, # data	Or immediate to A	2	2
XRL A, R	Exclusive or register to A	1	1
XRL A, @R	Exclusive or data memory to A	1	1
XRL, A, # data	Exclusive or immediate to A	2	2
INC A	Increment A	1	1
DEC A	Decrement A	1	1
CLR A	Clear A	1	1
CPL A	Complement A	1	1
DA A	Decimal adjust A	1	1
SWAP A	Swap nibbles of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1

Input/Output

Mnemonic	Description	Bytes	Cycles
IN A, P	Input port to A	1	2
OUTL P, A	Output A to port	1	2
ANL P, # data	And immediate to port	2	2
ORL P, # data	Or immediate to port	2	2
INS A, BUS	Input BUS to A	1	2
OUTL BUS, A	Output A to BUS	1	2
ANL BUS, # data	And immediate to BUS	2	2
ORL BUS, # data	Or immediate to BUS	2	2
MOVD A, P	Input expander port to A	1	2
MOVD P, A	Output A to expander port	1	2
ANLD P, A	And A to expander port	1	2
ORLD P, A	Or A to expander port	1	2

Registers

Mnemonic	Description	Bytes	Cycles
INC R	Increment register	1	1
INC @R	Increment data memory	1	1
DEC R	Decrement register	1	1

Branch

Mnemonic	Description	Bytes	Cycles
JMP addr	Jump unconditional	2	2
JMPP @A	Jump indirect	1	2
DJNZ R, addr	Decrement register and skip	2	2
JC addr	Jump on carry = 1	2	2
JNC addr	Jump on carry = 0	2	2
JZ addr	Jump on A zero	2	2
JNZ addr	Jump on A not zero	2	2
JT0 addr	Jump on T0 = 1	2	2
JNT0 addr	Jump on T0 = 0	2	2
JT1 addr	Jump on T1 = 1	2	2
JNT1 addr	Jump on T1 = 0	2	2
JF0 addr	Jump on F0 = 1	2	2
JF1 addr	Jump on F1 = 1	2	2
JTF addr	Jump on timer flag	2	2
JNI addr	Jump on INT = 0	2	2
JBb addr	Jump on accumulator bit	2	2

Subroutine

Mnemonic	Description	Bytes	Cycles
CALL addr	Jump to subroutine	2	2
RET	Return	1	2
RETR	Return and restore status	1	2

Flags

Mnemonic	Description	Bytes	Cycles
CLR C	Clear carry	1	1
CPL C	Complement carry	1	1
CLR F0	Clear flag 0	1	1
CPL F0	Complement flag 0	1	1
CLR F1	Clear flag 1	1	1
CPL F1	Complement flag 1	1	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 2. Instruction Set (Continued)

Data Moves

Mnemonic	Description	Bytes	Cycles
MOV A, R	Move register to A	1	1
MOV A, @R	Move data memory to A	1	1
MOV A, # data	Move immediate to A	2	2
MOV R, A	Move A to register	1	1
MOV @R, A	Move A to data memory	1	1
MOV R, # data	Move immediate to register	2	2
MOV @R, # data	Move immediate to data memory	2	2
MOV A, PSW	Move PSW to A	1	1
MOV PSW, A	Move A to PSW	1	1
XCH A, R	Exchange A and register	1	1
XCH A, @R	Exchange A and data memory	1	1
XCHD A, @R	Exchange nibble of A and register	1	1
MOVX A, @R	Move external data memory to A	1	2
MOVX @R, A	Move A to external data memory	1	2
MOVP A, @A	Move to A from current page	1	2
MOVP3 A, @A	Move to A from page 3	1	2

Timer/Counter

Mnemonic	Description	Bytes	Cycles
MOV A, T	Read timer/counter	1	1
MOV T, A	Load timer/counter	1	1
STRT T	Start timer	1	1
STRT CNT	Start counter	1	1
STOP TCNT	Stop timer/counter	1	1
EN TCNTI	Enable timer/counter interrupt	1	1
DIS TCNTI	Disable timer/counter interrupt	1	1

Control

Mnemonic	Description	Bytes	Cycles
EN I	Enable external interrupt	1	1
DIS I	Disable external interrupt	1	1
SEL RB0	Select register bank 0	1	1
SEL RB1	Select register bank 1	1	1
SEL MB0	Select memory bank 0	1	1
SEL MB1	Select memory bank 1	1	1
ENT0 CLK	Enable clock output on T0	1	1

Mnemonic	Description	Bytes	Cycles
NOP	No operation	1	1

D.C. CHARACTERISTICS: (T_A = 0° C to 70° C; V_{CC} = V_{DD} = 5V ± 10%; V_{SS} = 0V) (Continued)

Symbol	Parameter	Limits Min	Typ	Max	Unit	Test Conditions	Device
I_{LI}	Leakage Current (T1, $\overline{INT}$)			±10	μA	$V_{SS} \leq V_{IN} \leq V_{CC}$	All
I_{LI1}	Input Leakage Current (P10–P17, P20–P27, EA, $\overline{SS}$)			−500	μA	$V_{SS}+.45 \leq V_{IN} \leq V_{CC}$	All
I_{LI2}	Input Leakage Current $\overline{RESET}$	−10		−300	μA	$V_{SS} \leq V_{IN} \leq 3.8V$	All
I_{LO}	Leakage Current (BUS, T0) (High Impedance State)			±10	μA	$V_{SS} \leq V_{IN} \leq V_{CC}$	All
I_{DD} + I_{CC}	Total Supply Current*		80	100	mA		8035H
			95	110	mA		8039H
			80	100	mA		8748H
			95	110	mA		8749H

*I_{CC} + I_{DD} is measured with all outputs disconnected; $\overline{SS}$, $\overline{RESET}$, and $\overline{INT}$ equal to V_{CC}; EA equal to V_{SS}.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 . . . . . 1.0 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 device at these or any other conditions above those indicated in the operational sections of this specification is not implied.*

D.C. CHARACTERISTICS: (T_A = 0°C to 70°C; V_{CC} = V_{DD} = 5V ± 10%; V_{SS} = 0V)

Symbol	Parameter	Limits			Unit	Test Conditions	Device
		Min	Typ	Max			
V_{IL}	Input Low Voltage (All Except RESET, X1, X2)	-.5		.8	V		All
V_{IL1}	Input Low Voltage (RESET, X1, X2)	-.5		.6	V		All
V_{IH}	Input High Voltage (All Except XTAL1, XTAL2, RESET)	2.0		V_{CC}	V		All
V_{IH1}	Input High Voltage (X1, X2, RESET)	3.8		V_{CC}	V		All
V_{OL}	Output Low Voltage (BUS)			.45	V	I_{OL} = 2.0 mA	All
V_{OL1}	Output Low Voltage (RD, WR, PSEN, ALE)			45	V	I_{OL} = 1.8 mA	All
V_{OL2}	Output Low Voltage (PROG)			.45	V	I_{OL} = 1.0 mA	All
V_{OL3}	Output Low Voltage (All Other Outputs)			.45	V	I_{OL} = 1.6 mA	All
V_{OH}	Output High Voltage (BUS)	2.4			V	I_{OH} = -400 μA	All
V_{OH1}	Output High Voltage (RD, WR, PSEN, ALE)	2.4			V	I_{OH} = -100 μA	All
V_{OH2}	Output High Voltage (All Other Outputs)	2.4			V	I_{OH} = -40 μA	All

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A.C. CHARACTERISTICS: (T_A = 0°C to 70°C; V_{CC} = V_{DD} = 5V ± 10%; V_{SS} = 0V)

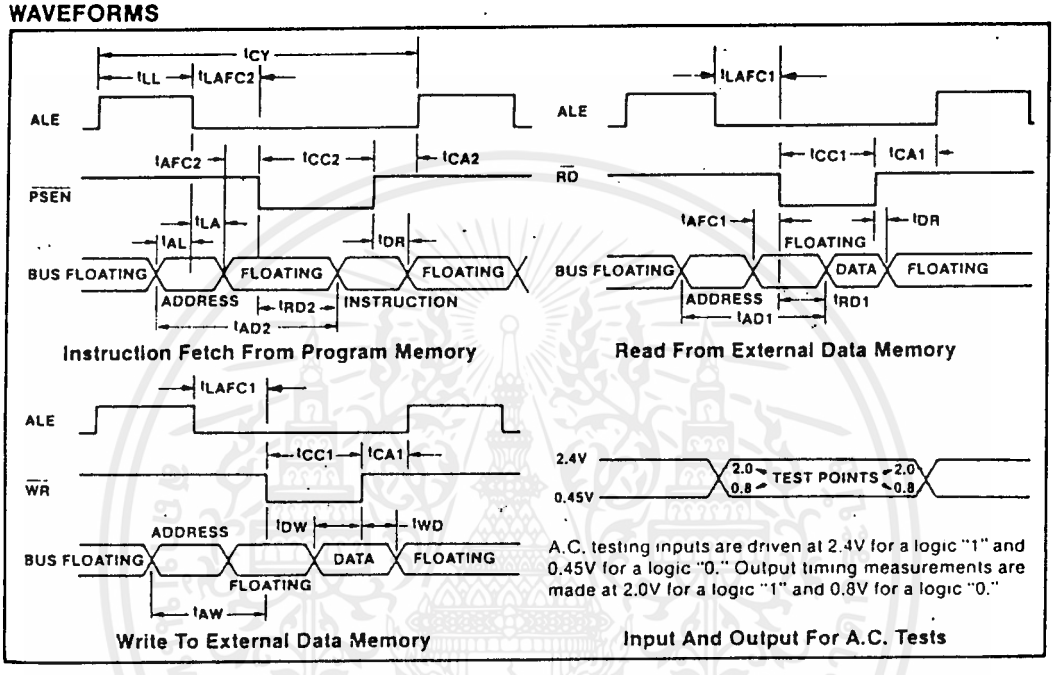
Symbol	Parameter	f (t) (Note 3)	11 MHz		Unit	Conditions (Note 1)
			Min	Max		
t	Clock Period	1/xtal freq	90.9	1000	ns	(Note 3)
t_{LL}	ALE Pulse Width	3.5t-170	150		ns	
t_{AL}	Addr Setup to ALE	2t-110	70		ns	(Note 2)
t_{LA}	Addr Hold from ALE	t-40	50		ns	
t_{CC1}	Control Pulse Width ($\overline{RD}$, $\overline{WR}$)	7.5t-200	480		ns	
t_{CC2}	Control Pulse Width ($\overline{PSEN}$)	6t-200	350		ns	
t_{DW}	Data Setup before $\overline{WR}$	6.5t-200	390		ns	
t_{WD}	Data Hold after $\overline{WR}$	t-50	40		ns	
t_{DR}	Data Hold ($\overline{RD}$, $\overline{PSEN}$)	1.5t-30	0	110	ns	
t_{RD1}	$\overline{RD}$ to Data in	6t-170		375	ns	
t_{RD2}	$\overline{PSEN}$ to Data in	4.5t-170		240	ns	
t_{AW}	Addr Setup to $\overline{WR}$	5t-150	300		ns	
t_{AD1}	Addr Setup to Data ($\overline{RD}$)	10.5t-220		730	ns	
t_{AD2}	Addr Setup to Data ($\overline{PSEN}$)	7.5t-200		460	ns	
t_{AFC1}	Addr Float to $\overline{RD}$, $\overline{WR}$	2t-40	140		ns	(Note 2)
t_{AFC2}	Addr Float to $\overline{PSEN}$	.5t-40	10		ns	(Note 2)
t_{LAFC1}	ALE to Control ($\overline{RD}$, $\overline{WR}$)	3t-75	200		ns	
t_{LAFC2}	ALE to Control ($\overline{PSEN}$)	1.5t-75	60		ns	
t_{CA1}	Control to ALE ($\overline{RD}$, $\overline{WR}$, $\overline{PROG}$)	t-65	25		ns	
t_{CA2}	Control to ALE ($\overline{PSEN}$)	4t-70	290		ns	
t_{CP}	Port Control Setup to $\overline{PROG}$	1.5t-80	50		ns	
t_{PC}	Port Control Hold to $\overline{PROG}$	4t-260	100		ns	
t_{PR}	$\overline{PROG}$ to P2 Input Valid	8.5t-120		650	ns	
t_{PF}	Input Data Hold from $\overline{PROG}$	1.5t	0	140	ns	
t_{DP}	Output Data Setup	6t-290	250		ns	
t_{PD}	Output Data Hold	1.5t-90	40		ns	
t_{PP}	$\overline{PROG}$ Pulse Width	10.5t-250	700		ns	
t_{PL}	Port 2 I/O Setup to ALE	4t-200	160		ns	
t_{LP}	Port 2 I/O Hold to ALE	.5t-30	15		ns	
t_{PV}	Port Output from ALE	4.5t+100		510	ns	
t_{0PRR}	T0 Rep Rate	3t	270		ns	
t_{CY}	Cycle Time	15t	1.36	15.0	μs	

Notes:

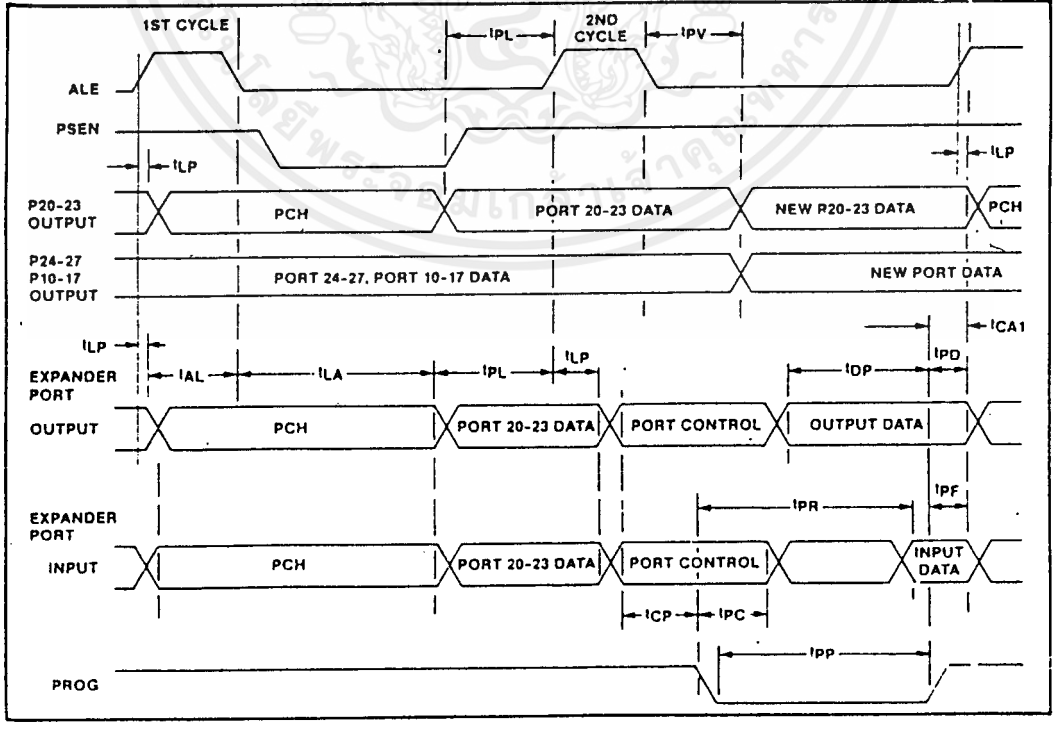
1. Control Outputs CL = 80pF
   BUS Outputs CL = 150pF
2. BUS High Impedance Load 20pF
3. f(t) assumes 50% duty cycle on X1, X2. Max clock period is for a 1 MHz crystal input.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAVEFORMS

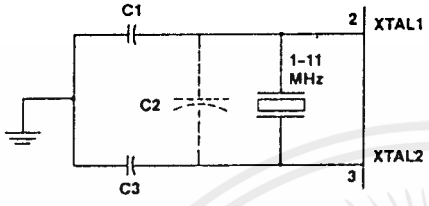


PORT 1/PORT 2 TIMING



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

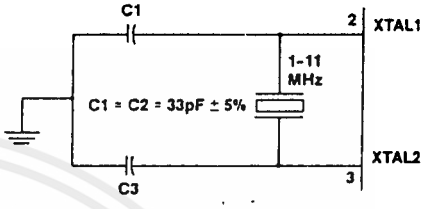
CRYSTAL OSCILLATOR MODE



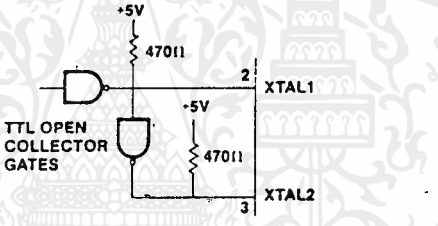
C1 = 5pF ± 1/2pF + (STRAY < 5pF)
C2 = (CRYSTAL + STRAY) < 8pF
C3 = 20pF ± 1pF + (STRAY < 5pF)

Crystal series resistance should be less than 30Ω at 11 MHz; less than 75Ω at 6 MHz; less than 180Ω at 3.6 MHz.

CERAMIC RESONATOR MODE



DRIVING FROM EXTERNAL SOURCE



For XTAL1 and XTAL2 define "high" as voltages above 4.6V and "low" as voltages below 1.6V. The duty cycle requirements for externally driving XTAL1 and XTAL2 using the circuit shown above are as follows: XTAL1 must be high 35-65% of the period and XTAL2 must be high 36-65% of the period. Rise and fall times must be faster than 20 nS.

PROGRAMMING, VERIFYING, AND ERASING THE 8749H (8748H) EPROM

Programming Verification

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL 1	Clock Input (3 to 4.0 MHz)
XTAL 2	
Reset	Initialization and Address Latching
Test 0	Selection of Program or Verify Mode
EA	Activation of Program/Verify Modes
BUS	Address and Data Input
	Data Output During Verify
P20-P22	Address Input
V_{DD}	Programming Power Supply
PROG	Program Pulse Input

WARNING:
An attempt to program a missocketed 8749H (8748H) will result in severe damage to the part. An indication of a properly socketed part is the appearance of the ALE clock output. The lack of this clock may be used to disable the programmer.

The Program/Verify sequence is:

1. V_{DD} = 5V, Clock applied or internal oscillator operating, RESET = 0V, TEST 0 = 5V, EA = 5V, BUS and PROG floating, P10 and P11 must be tied to ground.
2. Insert 8749H (8748H) in programming socket.
3. TEST 0 = 0V (select program mode)
4. EA = 18V (activate program mode)
5. Address applied to BUS and P20-22
6. RESET = 5V (latch address)
7. Data applied to BUS
8. V_{DD} = 21V (programming power)
9. PROG = V_{CC} or float followed by one 50ms pulse to 18V
10. V_{DD} = 5V
11. TEST 0 = 5V (verify mode)
12. Read and verify data on BUS
13. TEST 0 = 0V
14. RESET = 0V and repeat from step 5
15. Programmer should be at conditions of step 1 when 8749H (8748H) is removed from socket.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A.C. TIMING SPECIFICATION FOR PROGRAMMING 8748H/8749H ONLY:

(T_A = 25°C ± 5°C; V_{CC} = 5V ± 5%; V_{DD} = 21 ± .5V)

Symbol	Parameter	Min	Max	Unit	Test Conditions
t_{AW}	Address Setup Time to $\overline{RESET}$↓	$4t_{CY}$			
t_{WA}	Address Hold Time After $\overline{RESET}$↓	$4t_{CY}$			
t_{DW}	Data in Setup Time to PROG↓	$4t_{CY}$			
t_{WD}	Data in Hold Time After PROG↓	$4t_{CY}$			
t_{PH}	$\overline{RESET}$ Hold Time to Verify	$4t_{CY}$			
t_{VDDW}	V_{DD} Hold Time Before PROG↑	0	1.0	ms	
t_{VDDH}	V_{DD} Hold Time After PROG↓	0	1.0	ms	
t_{PW}	Program Pulse Width	50	60	ms	
t_{TW}	Test 0 Setup Time for Program Mode	$4t_{CY}$			
t_{WT}	Test 0 Hold Time After Program Mode	$4t_{CY}$			
t_{DO}	Test 0 to Data Out Delay		$4t_{CY}$		
t_{WW}	$\overline{RESET}$ Pulse Width to Latch Address	$4t_{CY}$			
t_r, t_f	V_{DD} and PROG Rise and Fall Times	0.5	100	μs	
t_{CY}	CPU Operation Cycle Time	3.75	5	μs	
t_{RE}	$\overline{RESET}$ Setup Time before EA↑	$4t_{CY}$			

NOTE: If Test 0 is high, t_{DO} can be triggered by $\overline{RESET}$↓.

D.C. TIMING SPECIFICATION FOR PROGRAMMING 8748H/8749H ONLY:

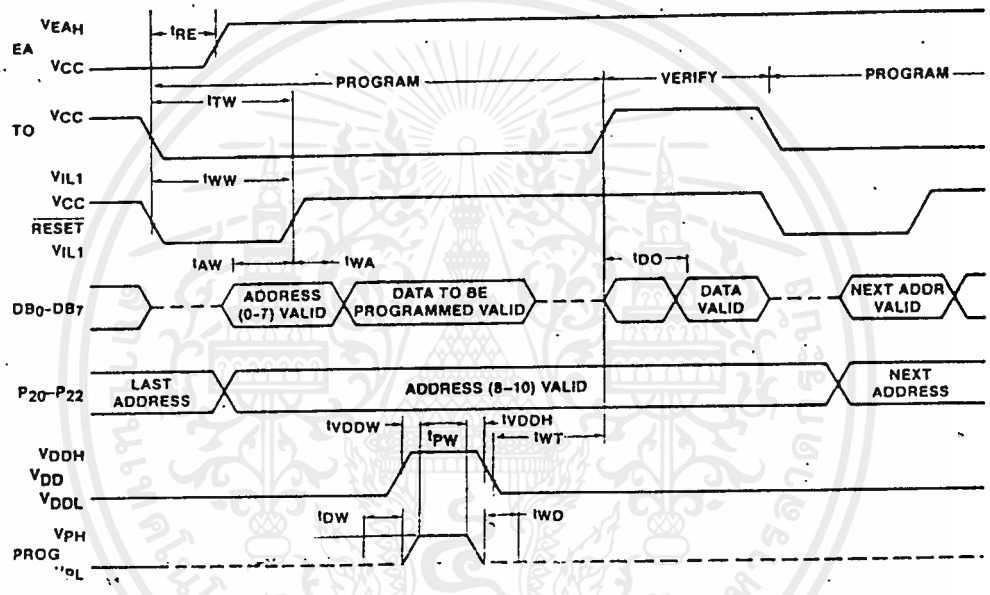
(T_A = 25°C ± 5°C; V_{CC} = 5V ± 5%; V_{DD} = 21 ± .5V)

Symbol	Parameter	Min	Max	Unit	Test Conditions
V_{DDH}	V_{DD} Program Voltage High Level	20.5	21.5	V	
V_{DDL}	V_{DD} Voltage Low Level	4.75	5.25	V	
V_{PH}	PROG Program Voltage High Level	17.5	18.5	V	
V_{PL}	PROG Voltage Low Level	4.0	V_{CC}	V	
V_{EAH}	EA Program or Verify Voltage High Level	17.5	18.5	V	
I_{DD}	V_{DD} High Voltage Supply Current		20.0	mA	
I_{PROG}	PROG High Voltage Supply Current		1.0	mA	
I_{EA}	EA High Voltage Supply Current		1.0	mA	

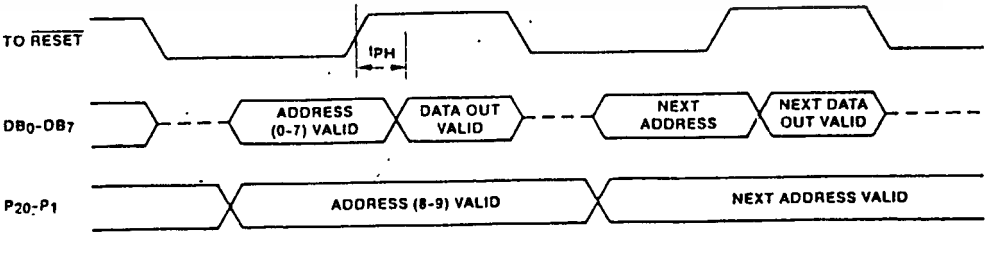
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAVEFORMS

COMBINATION PROGRAM/VERIFY MODE (EPROM'S ONLY)

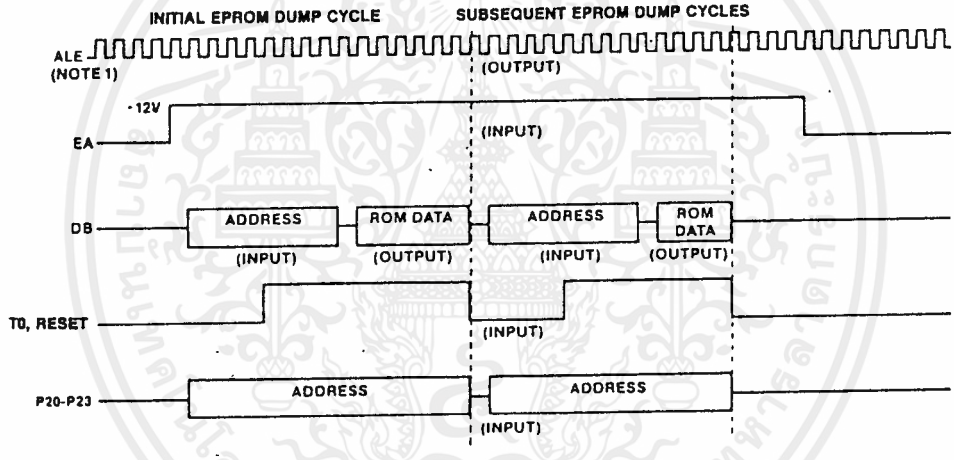


VERIFY MODE



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SUGGESTED EPROM VERIFICATION ALGORITHM FOR HMOS-E DEVICE ONLY



	48H	49H
A10	0	ADDR
A11	0	0

$V_{CC} = V_{DD} = 5V$

$V_{SS} = 0V$

NOTE: ALE is function of X1, X2 inputs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

MCS-48 INSTRUCTION SET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHAPTER 3
MCS®-48 INSTRUCTION SET

3.0 INTRODUCTION

The MCS®-48 instruction set is extensive for a machine of its size and has been tailored to be straightforward and very efficient in its use of program memory. All instructions are either one or two bytes in length and over 80% are only one byte long. Also, all instructions execute in either one or two cycles and over 50% of all instructions execute in a single cycle. Double cycle instructions include all immediate instructions, and all I/O instructions.

The MCS-48 microcomputers have been designed to handle arithmetic operations efficiently in both binary and BCD as well as handle the single-bit operations required in control applications. Special instructions have also been included to simplify loop counters, table look-up routines, and N-way branch routines.

3.0.1 Data Transfers

As can be seen in Figure 14.1, the 8-bit accumulator is the central point for all data transfers within the 8048. Data can be transferred between the 8 registers of each working register bank and the accumulator directly, i.e., the source or destination register is specified by the instruction. The remaining locations of the internal RAM array are referred to as Data Memory and are addressed indirectly via an address stored in either R0 or R1 of the active register bank. R0 and R1 are also used to indirecly address external data memory when it is present. Transfers to and from internal RAM require one cycle, while transfers to external RAM require two. Constants stored in Program Memory can be loaded directly to the accumulator and to the 8 working registers. Data can also be transferred directly between the accumulator and the on-

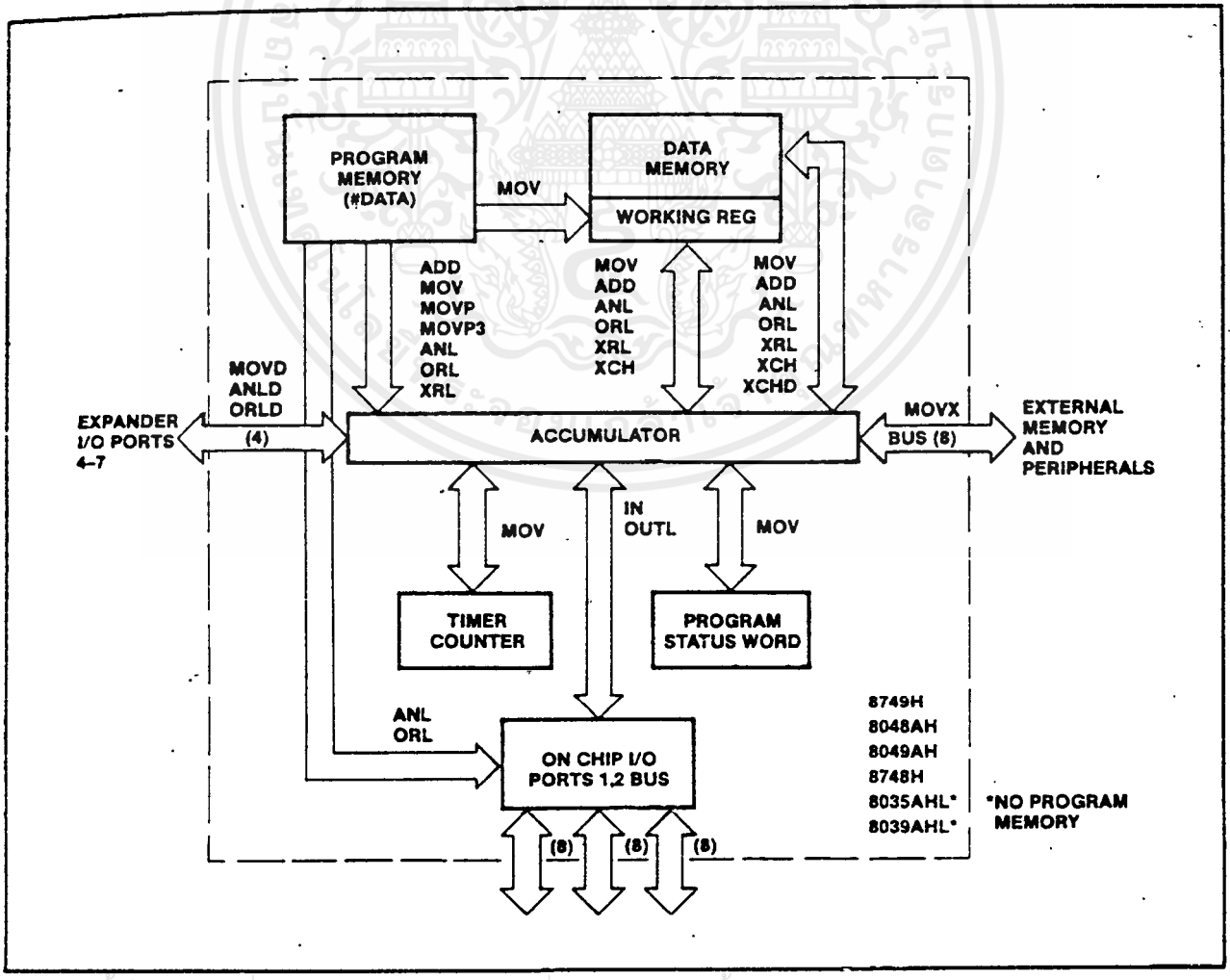


Figure 14-1. Data Transfer Instructions

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

board timer counter or the accumulator and the Program Status word (PSW). Writing to the PSW alters machine status accordingly and provides a means of restoring status after an interrupt or of altering the stack pointer if necessary.

3.0.2 Accumulator Operations

Immediate data, data memory, or the working registers can be added with or without carry to the accumulator. These sources can also be ANDed, ORed, or Exclusive ORed to the accumulator. Data may be moved to or from the accumulator and working registers or data memory. The two values can also be exchanged in a single operation.

In addition, the lower 4 bits of the accumulator can be exchanged with the lower 4-bits of any of the internal RAM locations. This instruction, along with an instruction which swaps the upper and lower 4-bit halves of the accumulator, provides for easy handling of 4-bit quantities, including BCD numbers. To facilitate BCD arithmetic, a Decimal Adjust instruction is included. This instruction is used to correct the result of the binary addition of two 2-digit BCD numbers. Performing a decimal adjust on the result in the accumulator produces the required BCD result.

Finally, the accumulator can be incremented, decremented, cleared, or complemented and can be rotated left or right 1 bit at a time with or without carry.

Although there is no subtract instruction in the 8048AH, this operation can be easily implemented with three single-byte single-cycle instructions.

A value may be subtracted from the accumulator with the result in the accumulator by:

- Complementing the accumulator
- Adding the value to the accumulator
- Complementing the accumulator

3.0.3 Register Operations

The working registers can be accessed via the accumulator as explained above, or can be loaded immediate with constants from program memory. In addition, they can be incremented or decremented or used as loop counters using the decrement and jump, if not zero instruction, as explained under branch instructions.

All Data Memory including working registers can be accessed with indirect instructions via R0 and R1 and can be incremented.

3.0.4 Flags

There are four user-accessible flags in the 8048AH: Carry, Auxiliary Carry, F0 and F1. Carry indicates overflow of the accumulator, and Auxiliary Carry is used to indiate overflow between BCD digits and is used during decimal-adjust operation. Both Carry and Auxiliary Carry are accessible as part of the program status word and are stored on the stack during subroutines. F0 and F1 are undedicated general-purpose flags to be used as the programmer desires. Both flags can be cleared or complemented and tested by conditional jump instructions. F0 is also accessible via the Program Status word and is stored on the stack with the carry flags.

3.0.5 Branch Instructions

The unconditional jump instruction is two bytes and allows jumps anywhere in the first 2K words of program memory. Jumps to the second 2K of memory (4K words are directly addressable) are made first by executing a select memory bank instruction, then executing the jump instruction. The 2K boundary can only be crossed via a jump or subroutine call instruction, i.e., the bank switch does not occur until a jump is executed. Once a memory bank has been selected all subsequent jumps will be to the selected bank until another select memory bank instruction is executed. A subroutine in the opposite bank can be accessed by a select memory bank instruction followed by a call instruction. Upon completion of the subroutine, execution will automatically return to the original bank; however, unless the original bank is reselected, the next jump instruction encountered will again transfer execution to the opposite bank.

Conditional jumps can test the following inputs and machine status:

- T0 Input Pin
- T1 Input Pin
- $\overline{\text{INT}}$ Input Pin
- Accumulator Zero
- Any bit of Accumulator
- Carry Flag
- F0 Flag
- F1 Flag

Conditional jumps allow a branch to any address within the current page (256 words) of execution. The conditions tested are the instantaneous values at the time the conditional jump is executed. For instance, the jump on accumulator zero instruction tests the accumulator itself, not an intermediate zero flag.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The decrement register and jump if not zero instruction combines a decrement and a branch instruction to create an instruction very useful in implementing a loop counter. This instruction can designate any one of the 8 working registers as a counter and can effect a branch to any address within the current page of execution.

A single-byte indirect jump instruction allows the program to be vectored to any one of several different locations based on the contents of the accumulator. The contents of the accumulator points to a location in program memory which contains the jump address. The 8-bit jump address refers to the current page of execution. This instruction could be used, for instance, to vector to any one of several routines based on an ASCII character which has been loaded in the accumulator. In this way ASCII key inputs can be used to initiate various routines.

3.0.6 Subroutines

Subroutines are entered by executing a call instruction. Calls can be made like unconditional jumps to any address in a 2K word bank, and jumps across the 2K boundary are executed in the same manner. Two separate return instructions determine whether or not status (upper 4-bits of PSW) is restored upon return from the subroutine.

The return and restore status instruction also signals the end of an interrupt service routine if one has been in progress.

3.0.7 Timer Instructions

The 8-bit on board timer/counter can be loaded or read via the accumulator while the counter is stopped or while counting. The counter can be started as a timer with an internal clock source or an event counter or timer with an external clock applied to the T1 input pin. The instruction executed determines which clock source is used. A single instruction stops the counter whether it is operating with an internal or an external clock source. In addition, two instructions allow the timer interrupt to be enabled or disabled.

3.0.8 Control Instructions

Two instructions allow the external interrupt source to be enabled or disabled. Interrupts are initially disabled and are automatically disabled while an interrupt service routines is in progress and re-enabled afterward.

There are four memory bank select instructions, two to designate the active working register bank and two to control program memory banks. The operation of the program memory bank switch is explained in section 13.1.2.

The working register bank switch instructions allow the programmer to immediately substitute a second 8-register working register bank for the one in use. This effectively provides 16 working registers or it can be used as a means of quickly saving the contents of the registers in response to an interrupt. The user has the option to switch or not to switch banks on interrupt. However, if the banks are switched, the original bank will be automatically restored upon execution of a return and restore status instruction at the end of the interrupt service routine.

A special instruction enables an internal clock, which is the XTAL frequency divided by three to be output on pin T0. This clock can be used as a general-purpose clock in the user's system. This instruction should be used only to initialize the system since the clock output can be disabled only by application of system reset.

3.0.9 Input/Output Instructions

Ports 1 and 2 are 8-bit static I/O ports which can be loaded to and from the accumulator. Outputs are statically latched but inputs are not latched and must be read while inputs are present. In addition, immediate data from program memory can be ANDed or ORed directly to Port 1 and Port 2 with the result remaining on the port. This allows "masks" stored in program memory to selectively set or reset individual bits of the I/O ports. Ports 1 and 2 are configured to allow input on a given pin by first writing a "1" out to the pin.

An 8-bit port called BUS can also be accessed via the accumulator and can have statically latched outputs as well. It too can have immediate data ANDed or ORed directly to its outputs, however, unlike ports 1 and 2, all eight lines of BUS must be treated as either input or output at any one time. In addition to being a static port, BUS can be used as a true synchronous bi-directional port using the Move External instructions used to access external data memory. When these instructions are executed, a corresponding $\overline{\text{READ}}$ or $\overline{\text{WRITE}}$ pulse is generated and data is valid only at that time. When data is not being transferred, BUS is in a high impedance state. Note that the OUTL, ANL, and the ORL instructions for the BUS are for use with internal program memory only.

The basic three on-board I/O ports can be expanded via a 4-bit expander bus using half of port 2. I/O expander devices on this bus consist of four 4-bit ports which are addressed as ports 4 through 7. These ports have their own AND and OR instructions like the on-board ports as well as move instructions to transfer data in or out. The expander AND and OR instructions, however, combine the contents of accumulator with the selected port rather than immediate data as is done with the on-board ports.

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

I/O devices can also be added externally using the BUS port as the expansion bus. In this case the I/O ports become "memory mapped", i.e., they are addressed in the same way as external data memory and exist in the external data memory address space addressed by pointer register R0 or R1.

3.1 INSTRUCTION SET DESCRIPTION

The following pages describe the MCS®-48 instruction set in detail. The instruction set is first summarized with instructions grouped functionally. This summary page is followed by a detailed description listed alphabetically by mnemonic opcode.

The alphabetical listing includes the following information.

- Mnemonic
- Machine Code
- Verbal Description
- Symbolic Description
- Assembly Language Example

The machine code is represented with the most significant bit (7) to the left and two byte instructions are represented with the first byte on the left. The assembly language examples are formulated as follows:

Arbitrary

Label: Mnemonic, Operand;

Descriptive Comment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MCS®-48 INSTRUCTION SET
Symbols and Abbreviations Used

A	Accumulator
AC	Auxiliary Carry
addr	12-Bit Program Memory Address
Bb	Bit Designator (b = 0–7)
BS	Bank Switch
BUS	BUS Port
C	Carry
CLK	Clock
CNT	Event Counter
CRR	Conversion Result Register
D	Mnemonic for 4-Bit Digit (Nibble)
data	8-Bit Number or Expression
DBF	Memory Bank Flip-Flop
F0, F1	Flag 0, Flag 1
I	Interrupt
P	Mnemonic for "in-page" Operation
PC	Program Counter
Pp	Port Designator (p = 1, 2 or 4–7)
PSW	Program Status Word
Ri	Data memory Pointer (i = 0, or 1)
Rr	Register Designator (r = 0–7)
SP	Stack Pointer
T	Timer
TF	Timer Flag
T0, T1	Test 0, Test 1
X	Mnemonic for External RAM
#	Immediate Data Prefix
@	Indirect Address Prefix
$	Current Value of Program Counter
(X)	Contents of X
((X))	Contents of Location Addressed by X
←	Is Replaced by

Mnemonics copyright Intel Corporation 1983.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADD A,R_r Add Register Contents to Accumulator

Encoding: | 0 1 1 0 | 1 r r r | 68H-6FH

Description: The contents of register 'r' are added to the accumulator. Carry is affected.

Operation: (A) ← (A) + (Rr) r = 0-7

Example:
```
ADDREG: ADD A,R6        ;ADD REG 6 CONTENTS
                        ;TO ACC
```

ADD A,@R_i Add Data Memory Contents to Accumulator

Encoding; | 0 1 1 0 | 0 0 0 i | 60H-61H

Description: The contents of the resident data memory location addressed by register 'i' bits 0–5** are added to the accumulator. Carry is affected.

Operation: (A) ← (A) + ((Ri)) i = 0-1

Example:
```
ADDM:  MOV R0, #01FH    ;MOVE '1F' HEX TO REG 0
       ADD A, @R0       ;ADD VALUE OF LOCATION
                        ;31 TO ACC
```

ADD A,#data Add Immediate Data to Accumulator

Encoding: | 0 0 0 0 | 0 0 1 1 | | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | 03H

Description: This is a 2-cycle instruction. The specified data is added to the accumulator. Carry is affected.

Operation: (A) ← (A) + data

Example:
```
ADDID:  ADD A,#ADDER:   ;ADD VALUE OF SYMBOL
                        ;ADDER' TO ACC
```

ADDC A,R_r Add Carry and Register Contents to Accumulator

Encoding: | 0 1 1 1 | 1 r r r | 78H-7FH

Description: The content of the carry bit is added to accumulator location 0 and the carry bit cleared. The contents of register 'r' are then added to the accumulator. Carry is affected.

Operation: (A) ← (A) + (Rr) + (C) r = 0-7

Example:
```
ADDRGC: ADDC A,R4       ;ADD CARRY AND REG 4
                        ;CONTENTS TO ACC
```

** 0–5 in 8048AH/8748H
0–6 in 8049AH/8749H
0–7 in 8050AH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADDC A,@R_i Add Carry and Data Memory Contents to Accumulator

Encoding: | 0 1 1 1 | 0 0 0 i | 70H-71H

Description: The content of the carry bit is added to accumulator location 0 and the carry bit cleared. Then the contents of the resident data memory location addressed by register 'i' bits 0–5** are added to the accumulator. Carry is affected.

Operation: (A) ← (A) + ((Ri)) + (C) i = 0-1

Example:
```
ADDMC: MOV R1,#40      ;MOVE '40' DEC TO REG 1
       ADDC A,@R1      ;ADD CARRY AND LOCATION 40
                       ;CONTENTS TO ACC
```

ADDC A,@data Add Carry and Immediate Data to Accumulator

Encoding: | 0 0 0 1 | 0 0 1 1 | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | 13H

Description: This is a 2-cycle instruction. The content of the carry bit is added to accumulator location 0 and the carry bit cleared. Then the specified data is added to the accumulator. Carry is affected.

Operation: (A) ← (A) + data + (C)

Example:
```
ADDC A,#225            ;ADD CARRY AND '225' DEC
                       ;TO ACC
```

ANL A,R_r Logical AND Accumulator with Register Mask

Encoding: | 0 1 0 1 | 1 r r r | 58H-5FH

Description: Data in the accumulator is logically ANDed with the mask contained in working register 'r'.

Operation: (A) ← (A) AND (Rr) r = 0-7

Example:
```
ANDREG: ANL A,R3       ;'AND' ACC CONTENTS WITH MASK
                       ;IN REG 3
```

ANL A,@R_i Logical AND Accumulator with memory Mask

Encoding: | 0 1 0 1 | 0 0 0 i | 50H-51H

Description: Data in the accumulator is logically ANDed with the mask contained in the data memory location referenced by register 'i' bits 0-5**.

Operation: (A) ← (A) AND ((Ri)) i = 0-1

Example:
```
ANDDM: MOV R0,#03FH    ;MOVE '3F' HEX TO REG 0
       ANL A, @R0      ;'AND' ACC CONTENTS WITH
                       ;MASK IN LOCATION 63
```

** 0–5 in 8048AH/8748H
0–6 in 8049AH/8749H
0–7 in 8050AH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ANL A,#data Logical AND Accumulator with Immediate Mask

Encoding: | 0 1 0 1 | 0 0 1 1 | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | 53H

Description: This is a 2-cycle instruction. Data in the accumulator is logically ANDed with an immediately-specified mask.

Operation: (A) ← (A) AND data

Examples:
```
ANDID:  ANL A,#0AFH      ;'AND' ACC CONTENTS
                         ;WITH MASK 10101111
        ANL A,#3 + X/Y   ;'AND' ACC CONTENTS
                         ;WITH VALUE OF EXP
                         ;'3 + XY/Y'
```

ANL BUS,#data* Logical AND BUS with Immediate Mask

Encoding: | 1 0 0 1 | 1 0 0 0 | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | 98H

Description: This is a 2-cycle instruction. Data on the BUS port is logically ANDed with an immediately-specified mask. This instruction assumes prior specification of an 'OUTL BUS, A' instruction.

Operation: (BUS) ← (BUS) AND data

Example:
```
ANDBUS: ANL BUS,#MASK    ;'AND' BUS CONTENTS
                         ;WITH MASK EQUAL VALUE
                         ;OF SYMBOL 'MASK'
```

ANL Pp,#data Logical AND Port 1-2 with Immediate Mask

Encoding: | 1 0 0 1 | 1 0 p p | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | 99H-9AH

Description: This is a 2-cycle instruction. Data on port 'p' is logically ANDed with an immediately-specified mask.

Operation: (Pp) ← (Pp) AND DATA p = 1-2

Example:
```
ANDP2:  ANL P2,#0F0H     ;'AND' PORT 2 CONTENTS
                         ;WITH MASK 'F0' HEX
                         ;(CLEAR P20-23)
```

* For use with internal program memory ONLY.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ANLD Pp,A Logical AND Port 4-7 with Accumulator Mask

Encoding:

1 0 0 1	1 1 p p

9CH-9FH

Description: This is a 2-cycle instruction. Data on port 'p' is logically ANDed with the digit mask contained in accumulator bits 0-3.

Operation: (Pp) ← (Pp) AND (A0-3) p = 4-7

Note: The mapping of port 'p' to opcode bits 0-1 is as follows:

1 0	Port
0 0	4
0 1	5
1 0	6
1 1	7

Example:

```
ANDP4:  ANLD P4,A       ;'AND' PORT 4 CONTENTS
                        ;WITH ACC BITS 0-3
```

CALL address Subroutine Call

Encoding:

a_{10} a_9 a_8 1	0 1 0 0

a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

Page	Hex Op Code
0	14
1	34
2	54
3	74
4	94
5	B4
6	D4
7	F4

Description: This is a 2-cycle instruction. The program counter and PSW bits 4-7 are saved in the stack. The stack pointer (PSW bits 0-2) is updated. Program control is then passed to the location specified by 'address'. PC bit 11 is determined by the most recent SEL MB instruction.

A CALL cannot begin in locations 2046-2047 or 4094-4095. Execution continues at the instruction following the CALL upon return from the subroutine.

Operation:

$((SP)) \leftarrow (PC), (PSW_{4-7})$
$(SP) \leftarrow (SP) + 1$
$(PC_{8-10}) \leftarrow (addr_{8-10})$
$(PC_{0-7}) \leftarrow (addr_{0-7})$
$(PC_{11}) \leftarrow DBF$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Example: Add three groups of two numbers. Put subtotals in locations 50, 51 and total in location 52.

```
        MOV R0,#50      ;MOVE '50' DEC TO ADDRESS
                        ;REG 0
BEGADD: MOV A,R1        ;MOVE CONTENTS OF REG 1
                        ;TO ACC
        ADD A,R2        ;ADD REG 2 TO ACC
        CALL SUBTOT     ;CALL SUBROUTINE 'SUBTOT'
        ADDC A R3       ;ADD REG 3 TO ACC
        ADDC A,R4       ;ADD REG 4 TO ACC
        CALL SUBTOT     ;CALL SUBROUTINE 'SUBTOT'
        ADDC A,R5       ;ADD REG 5 TO ACC
        ADDC A,R6       ;ADD REG 6 TO ACC
        CALL SUBTOT     ;CALL SUBROUTINE 'SUBTOT'
SUBTOT: MOV @R0,A       ;MOVE CONTENTS OF ACC TO
                        ;LOCATION ADDRESSED BY
                        ;REG 0
        INC R0          ;INCREMENT REG 0
        RET             ;RETURN TO MAIN PROGRAM
```

CLR A Clear Accumulator

Encoding: | 0 0 1 0 | 0 1 1 1 | 27H

Description: The contents of the accumulator are cleared to zero.

Operation: A ← 0

CLR C Clear Carry Bit

Encoding: | 1 0 0 1 | 0 1 1 1 | 97H

Description: During normal program execution, the carry bit can be set to one by the ADD, ADDC, RLC, CPL C, RRC, and DAA insructions. This instruction resets the carry bit to zero.

Operation: C ← 0

CLR F1 Clear Flag 1

Encoding: | 1 0 1 0 | 0 1 0 1 | A5H

Description: Flag 1 is cleared to zero.

Operation: (F1) ← 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CLR F0 Clear Flag 0

Encoding: | 1 0 0 0 | 0 1 0 1 | 85H

Description: Flag 0 is cleared to zero.

Operation: (F0) ← 0

CPL A Complement Accumulator

Encoding: | 0 0 1 1 | 0 1 1 1 | 37H

Description: The contents of the accumulator are complemented. This is strictly a one's complement. Each one is changed to zero and vice-versa.

Operation: (A) ← NOT (A)

Example: Assume accumulator contains 01101010.

```
CPLA: CPL A          ;ACC CONTENTS ARE COMPLE-
                     ;MENTED TO 10010101
```

CPL C Complement Carry Bit

Encoding: | 1 0 1 0 | 0 1 1 1 | A7H

Description: The setting of the carry bit is complemented; one is changed to zero, and zero is changed to one.

Operation: (C) ← NOT (C)

Example: Set C to one; current setting is unknown.

```
CTO1: CLR C          ;C IS CLEARED TO ZERO
      CPL C          ;C IS SET TO ONE
```

CPL F0 Complement Flag 0

Encoding: | 1 0 0 1 | 0 1 0 1 | 95H

Description: The setting of flag 0 is complemented; one is changed to zero, and zero is changed to one.

Operation: F0 ← NOT (F0)

CPL F1 Complement Flag 1

Encoding: | 1 0 1 1 | 0 1 0 1 | B5H

Description: The setting of flag 1 is complemented; one is changed to zero, and zero is changed to one.

Operation: (F1) ← NOT (F1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DA A Decimal Adjust Accumulator

Encoding: | 0 1 0 1 | 0 1 1 1 | 57H

Description: The 8-bit accumulator value is adjusted to form two 4-bit Binary Coded Decimal (BCD) digits following the binary addition of BCD numbers. The carry bit C is affected. If the contents of bits 0–3 are greater than nine, or if AC is one, the accumulator is incremented by six.

The four high-order bits are then checked. If bits 4–7 exceed nine, or if C is one, these bits are increased by six. If an overflow occurs, C is set to one.

Example: Assume accumulator contains 10011011.

```
DA A                        ;ACC Adjusted to 00000001
                            ;WITH C SET

C AC 7     4 3     0
0  0 1 0 0 1 1 0 1 1
     0 0 0 0 0 1 1 0        ADD SIX TO BITS 0-7
0  1 1 0 1 0 0 0 0 1
     0 1 1 0                ADD SIX TO BITS 4-7
1  0 0 0 0 0 0 0 0 1        OVERFLOW TO C
```

DEC A Decrement Accumulator

Encoding: | 0 0 0 0 | 0 1 1 1 | 07H

Description: The contents of the accumulator are decremented by one. The carry flag is not affected.

Operation: (A) ← (A) -1

Example: Decrement contents of external data memory location 63.

```
MOV R0,#3FH         ;MOVE '3F' HEX TO REG 0
MOVX A, @R0         ;MOVE CONTENTS OF
                    ;LOCATION 63 TO ACC
DEC A               ;DECREMENT ACC
MOVX @R0,A          ;MOVE CONTENTS OF ACC TO
                    ;LOCATION 63 IN EXPANDED
                    ;MEMORY
```

DEC Rr Decrement Register

Encoding: | 1 1 0 0 | 1 r r r | C8H–CFH

Description: The contents of working register 'r' are decremented by one.

Operation: (Rr) ← (Rr) -1 r = 0–7

Example:

```
DECR1: DEC R1       ;DECREMENT CONTENTS OF REG 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DIS I External Interrupt

Encoding: | 0 0 0 1 | 0 1 0 1 | 15H

Description: External interrupts are disabled. A low signal on the interrupt input pin has no effect.

DIS TCNTI Disable Timer/Counter Interrupt

Encoding: | 0 0 1 1 | 0 1 0 1 | 35H

Description: Timer/counter interrupts are disabled. Any pending timer interrupt request is cleared. The interrupt sequence is not initiated by an overflow, but the timer flag is set and time accumulation continues.

DJNZ R_r, address Decrement Register and Test

Encoding: | 1 1 1 0 | 1 r r r | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 | E8H-EFH

Description: This is a 2-cycle instruction. Register 'r' is decremented, then tested for zero. If the register contains all zeros, program control falls through to the next instruction. If the register contents are not zero, control jumps to the specified 'address'.

The address in this case must evaluate to 8-bits, that is, the jump must be to a location within the current 256-location page.

Example: $(Rr) \leftarrow (Rr) - 1$ r = 0-7
If Rr not 0
$(PC_{0-7}) \leftarrow$ addr
Note: A 12-bit address specification does not cause an error if the DJNZ instruction and the jump target are on the same page. If the DJNZ instruction begins in location 255 of a page, it must jump to a target address on the following page.

Example: Increment values in data memory locations 50-54.

```
        MOV R0,#50          ;MOVE '50' DEC TO ADDRESS
                            ;REG 0
        MOV R3,#5           ;MOVE '5' DEC TO COUNTER
                            ;REG 3
INCRT:  INC @R0             ;INCREMENT CONTENTS OF
                            ;LOCATION ADDRESSED BY
                            ;REG 0
        INC R0              ;INCREMENT ADDRESS IN REG 0
        DJNZ R3, INCRT      ;DECREMENT REG 3 — JUMP TO
                            ;'INCRT' IF REG 3 NONZERO
        NEXT —              ;'NEXT' ROUTINE EXECUTED
                            ;IF R3 IS ZERO
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EN I Enable External Interrupt

Encoding: | 0 0 0 0 | 0 1 0 1 | 05H

Description: External interrupts are enabled. A low signal on the interrupt input pin initiates the interrupt sequence.

EN TCNTI Enable Timer/Counter Interrupt

Encoding: | 0 0 1 0 | 0 1 0 1 | 25H

Description: Timer/counter interrupts are enabled. An overflow of the timer/counter initiates the interrupt sequence.

ENT0 CLK Enable Clock Output

Encoding: | 0 1 1 1 | 0 1 0 1 | 75H

Description: The test 0 pin is enabled to act as the clock output. This function is disabled by a system reset.

Example:
```
EMTST0:  ENT0 CLK          ;ENABLE T0 AS CLOCK OUTPUT
```

IN A,Pp Input Port or Data to Accumulator

Encoding: | 0 0 0 0 | 1 0 p p | 09H-0AH

Description: This is a 2-cycle instruction. Data present on port 'p' is transferred (read) to the accumulator.

Operation: (A) ← (Pp) p = 1-2

```
INP12:  IN A,P1        ;INPUT PORT 1 CONTENTS TO ACC
        MOV R6,A       ;MOVE ACC CONTENTS TO REG 6
        IN A,P2        ;INPUT PORT 2 CONTENTS TO ACC
        MOV R7,A       ;MOVE ACC CONTENTS TO REG 7
```

INC A Increment Accumulator

Encoding: | 0 0 0 1 | 0 1 1 1 | 17H

Description: The contents of the accumulator are incremented by one. Carry is not affected.

Operation: (A) ← (A) +1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Example: Increment contents of location 100 in external data memory.

```
INCA:  MOV R0,#100      ;MOVE '100' DEC TO ADDRESS REG 0
       MOVX A,@R0       ;MOVE CONTENTS OF LOCATION
                        ;100 TO ACC
       INC A            ;INCREMENT A
       MOVX @R0,A       ;MOVE ACC CONTENTS TO
                        ;LOCATION 101
```

INC R_r Increment Register

Encoding: | 0 0 0 1 | 1 r r r | 18H–1FH

Description: The contents of working register 'r' are incremented by one.

Operation: (Rr) ← (Rr) + 1 r = 0–7

Example:
```
INCR0:  INC R0          ;INCREMENT CONTENTS OF REG 0
```

INC @R_i Increment Data Memory Location

Encoding: | 0 0 0 1 | 0 0 0 i | 10H–11H

Description: The contents of the resident data memory location addressed by register 'i' bits 0–5** are incremented by one.

Operation: ((Ri)) ← ((Ri)) + 1 i = 0–1

Example:
```
INCDM:  MOV R1,#03FH    ;MOVE ONES TO REG 1
        INC @R1         ;INCREMENT LOCATION 63
```

INS A,BUS* Strobed Input of BUS Data to Accumulator

Encoding: | 0 0 0 0 | 1 0 0 0 | 08H

Description: This is a 2-cycle instruction. Data present on the BUS port is transferred (read) to the accumulator when the RD pulse is dropped. (Refer to section on programming memory expansion for details.)

Operation: (A) ← (BUS)

Example:
```
INPBUS:  INS A,BUS      ;INPUT BUS CONTENTS TO ACC
```

* For use with internal program memory ONLY.

** 0–5 in 8048AH/8748H
0–6 in 8049AH/8749H
0–7 in 8050AH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JBb address Jump If Accumulator Bit Is Set

Encoding:

b_2 b_1 b_0 1	0 0 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

Accumulator Bit	Hex Op Code
0	12
1	32
2	52
3	72
4	92
5	B2
6	D2
7	F2

Description: This is a 2-cycle instruction. Control passes to the specified address if accumulator bit 'b' is set to one.

Operation:

	b = 0–7
$(PC_{0-7}) \leftarrow$ addr	If Bb = 1
(PC) = (PC) + 2	If Bb = 0

Example:
```
JB4IS1: JB4 NEXT    ;JUMP TO 'NEXT' ROUTINE
                    ;IF ACC BIT 4 = 1
```

JC address Jump If Carry Is Set

Encoding:

1 1 1 1	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0	F6H

Description: This is a 2-cycle instruction. Control passes to the specified address if the carry bit is set to one.

Operation:

$(PC_{0-7}) \leftarrow$ addr	If C = 1
(PC) = (PC) + 2	If C = 0

Example:
```
JC1: JC OVFLOW      ;JUMP TO 'OVFLOW' ROUTINE
                    ;IF C = 1
```

JF0 address Jump If Flag 0 Is Set

Encoding:

1 0 1 1	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0	B6H

Description: This is a 2-cycle instruction. Control passes to the specified address if flag 0 is set to one.

Operation:

$(PC_{0-7}) \leftarrow$ addr	If F0 = 1
(PC) =(PC) + 2	If F0 = 0

Example:
```
JF0IS1: JF0 TOTAL   ;JUMP TO 'TOTAL' ROUTINE IF F0 = 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JF1 address Jump If Flag 1 Is Set

Encoding:

0	1	1	1	0	1	1	0	76H
a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0	

Description: This is a 2-cycle instruction. Control passes to the specified address if flag 1 is set to one.

Operation:
$(PC_{0-7}) \leftarrow$ addr — If F1 = 1
(PC) = (PC + 2) — If F1 = 0

Example:
```
JF1IS1: JF1 FILBUF      ;JUMP TO 'FILBUF'
                        ;ROUTINE IF F1 = 1
```

JMP address Direct Jump within 2K Block

Encoding:

a_{10}	a_9	a_8	0	0	1	0	0	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0

Page	Hex Op Code
0	04
1	24
2	44
3	64
4	84
5	A4
6	C4
7	E4

Description: This is a 2-cycle instruction. Bits 0–10 of the program counter are replaced with the directly-specified address. The setting of PC bit 11 is determined by the most recent SELECT MB instruction.

Operation:
$(PC_{8-10}) \leftarrow$ addr 8–10
$(PC_{0-7}) \leftarrow$ addr 0–7
$(PC_{11}) \leftarrow$ DBF

Example:
```
JMP SUBTOT              ;JUMP TO SUBROUTINE 'SUBTOT'
JMP $-6                 ;JUMP TO INSTRUCTION SIX
                        ;LOCATIONS BEFORE CURRENT
                        ;LOCATION
JMP 2FH                 ;JUMP TO ADDRESS '2F' HEX
```

JMPP @A Indirect Jump within Page

Encoding:

1	0	1	1	0	0	1	1	B3H

Description: This is a 2-cycle insruction. The contents of the program memory location pointed to by the accumulator are substituted for the 'page' portion of the program counter (PC bits 0–7).

เอกสารนี้เป็นเอกสาร... เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Operation: $(PC_{0-7}) \leftarrow ((A))$

Example: Assume accumulator contains 0FH.

```
JMPPAG: JMPP @A          ;JUMP TO ADDRESS STORED IN
                         ;LOCATION 15 IN CURRENT PAGE
```

JNC address Jump If Carry Is Not Set

Encoding:

1 1 1 0	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0	E6H

Description: This is a 2-cycle instruction. Control passes to the specified address if the carry bit is not set, that is, equals zero.

Operation:
$(PC_{0-7}) \leftarrow$ addr — If C = 0
(PC) = (PC) + 2 — If C = 1

Example:

```
JC0: JNC NOVFLO          ;JUMP TO 'NOVFLO' ROUTINE
                         ;IF C = 0
```

JNI address Jump If Interrupt Input Is Low

Encoding:

1 0 0 0	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0	86H

Description: This is a 2-cycle instruction. Control passes to the specified address if the interrupt input signal is low (= 0), that is, an external interrupt has been signaled. (This signal initiates an interrupt service sequence if the external interrupt is enabled.)

Operation:
$(PC_{0-7}) \leftarrow$ addr — If I = 0
(PC) = (PC) + 2 — If I = 1

Example:

```
LOC 3: JNI EXTINT        ;JUMP TO 'EXTINT' ROUTINE
                         ;IF I = 0
```

JNT0 address Jump If Test 0 Is Low

Encoding:

0 0 1 0	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0	26H

Description: This is a 2-cycle instruction. Control passes to the specified address, if the test 0 signal is low.

Operation:
$(PC_{0-7}) \leftarrow$ addr — If T0 = 0
(PC) = (PC) + 2 — If T0 = 1

Example:

```
JT0LOW: JNT0 60          ;JUMP TO LOCATION 60 DEC
                         ;IF T0 = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JNT1 address **Jump If Test 1 Is Low**

Encoding: | 0 1 0 0 | 0 1 1 0 | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 | 46H

Description: This is a 2-cycle instruction. Control passes to the specified address, if the test 1 signal is low.

Operation: $(PC_{0-7}) \leftarrow$ addr — If T1 = 0
(PC) = (PC) + 2 — If T1 = 1

JNZ Address **Jump If Accumulator Is Not Zero**

Encoding: | 1 0 0 1 | 0 1 1 0 | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 | 96H

Description: This is a 2-cycle instruction. Control passes to the specified address if the accumulator contents are nonzero at the time this instruction is executed.

Operation: $(PC_{0-7}) \leftarrow$ addr — If $A \neq 0$
(PC) = (PC) + 2 — If A = 0

Example:
```
JACCN0: JNZ 0ABH      ;JUMP TO LOCATION 'AB' HEX
                      ;IF ACC VALUE IS NONZERO
```

JTF address **Jump If Timer Flag Is Set**

Encoding: | 0 0 0 1 | 0 1 1 0 | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 | 16H

Description: This is a 2-cycle instruction. Control passes to the specified address if the timer flag is set to one, that is, the timer/counter register has overflowed. Testing the timer flag resets it to zero. (This overflow initiates an interrupt service sequence if the timer-overflow interrupt is enabled.)

Operation: $(PC_{0-7}) \leftarrow$ addr — If TF = 1
(PC) = (PC) + 2 — If TF = 0

Example:
```
JTF1: JTF TIMER       ;JUMP TO 'TIMER' ROUTINE
                      ;IF TF = 1
```

JT0 address **Jump If Test 0 Is High**

Encoding: | 0 0 1 1 | 0 1 1 0 | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 | 36H

Description: This is a 2-cycle instruction. Control passes to the specified address if the test 0 signal is high (= 1).

Operation: $(PC_{0-7}) \leftarrow$ addr — If T0 = 1
(PC) = (PC) + 2 — If T0 = 0

Example:
```
JT0HI: JT0 53         ;JUMP TO LOCATION 53 DEC
                      ;IF T0 = 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JT1 address Jump If Test 1 Is High

Encoding: | 0 1 0 1 | 0 1 1 0 | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 | 56H

Description: This is a 2-cycle instruction. Control passes to the specified address if the test 1 signal is high (= 1).

Operation: $(PC_{0-7}) \leftarrow$ addr — If T1 = 1
(PC) = (PC) + 2 — If T1 = 0

Example:
```
JT1HI: JT1 COUNT        ;JUMP TO 'COUNT' ROUTINE
                        ;IF T1 = 1
```

JZ address Jump If Accumulator Is Zero

Encoding: | 1 1 0 0 | 0 1 1 0 | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 | C6H

Description: This is a 2-cycle instruction. Control passes to the specified address if the accumulator contains all zeros at the time this instruction is executed.

Operation: $(PC_{0-7}) \leftarrow$ addr — If A = 0
(PC) = (PC) + 2 — If A ≠ 1

Example:
```
JACCO: JZ 0A3H          ;JUMP TO LOCATION 'A3' HEX
                        ;IF ACC VALUE IS ZERO
```

MOV A,#data Move Immediate Data to Accumulator

Encoding: | 0 0 1 0 | 0 0 1 1 | a_7 a_6 a_5 a_4 | a_3 a_2 a_1 a_0 | 23H

Description: This is a 2-cycle instruction. The 8-bit value specified by 'data' is loaded in the accumulator.

Operation: (A) ← data

Example:
```
MOV A,#0A3H             ;MOVE 'A3' HEX TO ACC
```

MOV A,PSW Move PSW Contents to Accumulator

Encoding: | 1 1 0 0 | 0 1 1 1 | C7H

Description: The contents of the program status word are moved to the accumulator.

Operation: (A) ← (PSW)

Example: Jump to 'RB1SET' routine if PSW bank switch, bit 4, is set.
```
BSCHK:  MOV A,PSW       ;MOVE PSW CONTENTS TO ACC
        JB4 RB1SET      ;JUMP TO 'RB1SET' IF ACC BIT 4 = 1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOV A,R_r Move Register Contents to Accumulator

Encoding: | 1 1 1 1 | 1 r r r | F8H–FFH

Description: 8-bits of data are removed from working register 'r' into the accumulator.

Operation: (A) ← (Rr) r = 0–7

Example:
```
MAR:  MOV A,R3          ;MOVE CONTENTS OF REG 3 TO ACC
```

MOV A,@R_i Move Data Memory Contents to Accumulator

Encoding | 1 1 1 1 | 0 0 0 i | F0H–F1H

Description: The contents of the resident data memory location addressed by bits 0–5** of register 'i' are moved to the accumulator. Register 'i' contents are unaffected.

Operation: (A) ← ((Ri)) i = 0–1

Example: Assume R1 contains 00110110.
```
MADM:  MOV A,@R1        ;MOVE CONTENTS OF DATA MEM
                        ;LOCATION 54 TO ACC
```

MOV A,T Move Timer/Counter Contents to Accumulator

Encoding: | 0 1 0 0 | 0 0 1 0 | 42H

Description: The contents of the timer/event-counter register are moved to the accumulator.

Operation: (A) ← (T)

Example: Jump to "EXIT" routine when timer reaches '64', that is, when bit 6 set—assuming initialization 64,
```
TIMCHK:  MOV A,T        ;MOVE TIMER CONTENTS TO ACC
         JB6 EXIT       ;JUMP TO 'EXIT' IF ACC BIT 6 = 1
```

MOV PSW,A Move Accumulator Contents to PSW

Encoding: | 1 1 0 1 | 0 1 1 1 | D7H

Description: The contents of the accumulator are moved into the progam status word. All condition bits and the stack pointer are affected by this move.

Operation: (PSW) ← (A)

Example: Move up stack pointer by two memory locations, that is, increment the pointer by one.
```
INCPTR:  MOV A,PSW      ;MOVE PSW CONTENTS TO ACC
         INC A          ;INCREMENT ACC BY ONE
         MOV PSW.A      ;MOVE ACC CONTENTS TO PSW
```

** 0–5 in 8048AH/8748H
0–6 in 8049AH/8749H
0–7 in 8050AH

เอกสารนี้จัดทำขึ้นสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOV R_r,A Move Accumulator Contents to Register

Encoding: | 1 0 1 0 | 1 r r r | A8H-AFH

Description: The contents of the accumulator are moved to register 'r'.

Operation: (Rr) ← (A) r = 0-7

Example:
```
MRA: MOV R0,A        ;MOVE CONTENTS OF ACC TO REG 0
```

MOV R_r,#data Move Immediate Data to Register

Encoding: | 1 0 1 1 | 1 r_2 r_1 r_0 | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | B8H-BFH

Description: This is a 2-cycle instruction. The 8-bit value specified by 'data' is moved to register 'r'.

Operation: (Rr) ← data r = 0-7

Examples:
```
MIR4: MOV R4,#HEXTEN     ;THE VALUE OF THE SYMBOL
                         ;'HEXTEN' IS MOVED INTO REG 4
MIR 5: MOV R5,#PI*(R*R)  ;THE VALUE OF THE EXPRESSION
                         ;'PI*(R*R)' IS MOVED INTO REG 5
MIR 6: MOV R6, #0ADH     ;'AD' HEX IS MOVED INTO REG 6
```

MOV @ R_i,A Move Accumulator Contents to Data Memory

Encoding: | 1 0 1 0 | 0 0 0 i | A0H-A1H

Description: The contents of the accumulator are moved to the resident data memory location whose address is specified by bits 0-5** of register 'i'. Register 'i' contents are unaffected.

Operation: ((Ri)) ← (A) i = 0-1

Example: Assume R0 contains 00000111.
```
MDMA: MOV @R0,A          ;MOVE CONTENTS OF ACC TO
                         ;LOCATION 7 (REG 7)
```

MOV @ R_i,#data Move Immediate Data to Data memory

Encoding: | 1 0 1 1 | 0 0 0 i | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | B0H-B1H

Description: This is a 2-cycle instruction. The 8-bit value specified by 'data' is moved to the resident data memory location addressed by register 'i', bits 0-5**

Operation: ((Ri)) ← data i = 0-1

Examples: Move the hexadecimal value AC3F to locations 62-63.
```
MIDM: MOV R0,#62         ;MOVE '62' DEC TO ADDR REG 0
      MOV @R0,#0ACH      ;MOVE 'AC' HEX TO LOCATION 62
      INC R0             ;INCREMENT REG 0 to '63'
      MOV @R0,#3FH       ;MOVE '3F' HEX TO LOCATION 63
```

** 0-5 in 8048AH/8748H
0-6 in 8049AH/8749H
0-7 in 8050AH

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOV T,A Move Accumulator Contents to Timer/Counter

Encoding: | 0 1 1 0 | 0 0 1 0 | 62H

Description: The contents of the accumulator are moved to the timer/event-counter register.

Operation: (T) ← (A)

Example: Initialize and start event counter.

```
INITEC: CLR A          ;CLEAR ACC TO ZEROS
        MOV T,A        ;MOVE ZEROS TO EVENT COUNTER
        START CNT      ;START COUNTER
```

MOVD A,Pp Move Port 4-7 Data to Accumulator

Encoding: | 0 0 0 0 | 1 1 p p | 0CH-0FH

Description: This is a 2-cycle instruction. Data on 8243 port 'p' is moved (read) to accumulator bits 0-3. Accumulator bits 4-7 are zeroed.

Operation: (0-3) ← (Pp) p = 4-7
(4-7) ← 0

Note: Bits 0-7 of the opcode are used to represent ports 4-7. If you are coding in binary rather than assembly language, the mapping is as follows:

Bits 1 0	Port
0 0	4
0 1	5
1 0	6
1 1	7

Example:

```
INPPT5: MOVD A,P5      ;MOVE PORT 5 DATA TO ACC
                       ;BITS 0-3, ZERO ACC BITS 4-7
```

MOVD Pp,A Move Accumulator Data to Port 4-7

Encoding: | 0 0 1 1 | 1 1 p p | 3CH-3FH

Description: This is a 2-cycle instruction. Data in accumulator bits 0-3 is moved (written) to 8243 port 'p'. Accumulator bits 4-7 are unaffected. (See NOTE above regarding port mapping.)

Operation: (Pp) ← (A_{0-3}) P = 4-7

Example: Move data in accumulator to ports 4 and 5.

```
OUTP45: MOVD P4,A      ;MOVE ACC BITS 0-3 TO PORT 4
        SWAP A         ;EXCHANGE ACC BITS 0-3 and 4-7
        MOVD P5,A      ;MOVE ACC BITS 0-3 TO PORT 5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOVP A,@A Move Current Page Data to Accumulator

Encoding: | 1 0 1 0 | 0 0 1 1 | A3H

Description: The contents of the program memory location addressed by the accumulator are moved to the accumulator. Only bits 0–7 of the program counter are affected, limiting the program memory reference to the current page. The program counter is restored *following* this operation.

Operation: $(PC_{0-7}) \leftarrow (A)$
$(A) \leftarrow ((PC))$
Note: This is a 1-byte, 2-cycle instruction. If it appears in location 255 of a program memory page, @A addresses a location in the *following* page.

Example:
```
MOV128: MOV A,#128      ;MOVE '128' DEC TO ACC
        MOVP A,@A       ;CONTENTS OF 129th LOCATION IN
                        ;CURRENT PAGE ARE MOVED TO ACC
```

MOVP3 A,@A Move Page 3 Data to Accumulator

Encoding: | 1 1 1 0 | 0 0 1 1 | E3H

Description: This is a 2-cycle instruction. The contents of the program memory location (within page 3) addressed by the accumulator are moved to the accumulator. The program counter is restored following this operation.

Operation: $(PC_{0-7}) \leftarrow (A)$
$(PC_{8-11}) \leftarrow 0011$
$(A) \leftarrow ((PC))$

Example: Look up ASCII equivalent of hexadecimal code in table contained at the beginning of page 3. Note that ASCII characters are designated by a 7-bit code; the eighth bit is always reset.

```
TABSCH: MOV A,#0B8H     ;MOVE 'B8' HEX TO ACC (10111000)
        ANL A,#7FH      ;LOGICAL AND ACC TO MASK BIT
                        ;7 (00111000)
        MOVP3 A,@A      ;MOVE CONTENTS OF LOCATION '38'
                        ;HEX IN PAGE 3 TO ACC (ASCII '8')
```

Access contents of location in page 3 labelled TAB1.
Assume current program location is not in page 3.

```
TABSCH: MOV A,#LOW TAB 1  ;ISOLATE BITS 0-7 OF LABEL
                          ;ADDRESS VALUE
        MOVP3 A,@A        ;MOVE CONTENTS OF PAGE 3
                          ;LOCATION LABELED 'TAB1' TO ACC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOVX A,@R_i Move External-Data-Memory Contents to Accumulator

Encoding: | 1 0 0 0 | 0 0 0 i | 80H-81H

Description: This is a 2-cycle instruction. The contents of the external data memory location addressed by register 'i' are moved to the accumulator. Register 'i' contents are unaffected. A read pulse is generated.

Operation: (A) ← ((Ri)) i = 0-1

Example: Assume R1 contains 01110110.

```
MAXDM:  MOVX A,@R1      ;MOVE CONTENTS OF LOCATION
                        ;118 TO ACC
```

MOVX @R_i,A Move Accumulator Contents to External Data Memory

Encoding: | 1 0 0 1 | 0 0 0 i | 90H-91H

Description: This is a 2-cycle instruction. The contents of the accumulator are moved to the external data memory location addressed by register 'i'. Register 'i' contents are unaffected. A write pulse is generated.

Operation: ((Ri)) ← A i = 0-1

Example: Assume R0 contains 11000111.

```
MXDMA:  MOVX @R0,A      ;MOVE CONTENTS OF ACC TO
                        ;LOCATION 199 IN EXPANDED
                        ;DATA MEMORY
```

NOP The NOP Instruction

Encoding: | 0 0 0 0 | 0 0 0 0 | 00H

Description: No operation is performed. Execution continues with the following instruction.

ORL A,R_r Logical OR Accumulator With Register Mask

Encoding: | 0 1 0 0 | 1 r r r | 48H-4FH

Description: Data in the accumulator is logically ORed with the mask contained in working register 'r'.

Operation: (A) ← (A) OR (Rr) r = 0-7

Example:

```
ORREG:  ORL A,R4        ;'OR' ACC CONTENTS WITH
                        ;MASK IN REG 4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ORL A,@R_i Logical OR Accumulator With Memory Mask

Encoding: | 0 1 0 0 | 0 0 0 i | 40H-41H

Description: Data in the accumulator is logically ORed with the mask contained in the resident data memory location referenced by register "i", bits 0–5**.

Operation: (A) ← (A) OR ((Ri)) i = 0-1

Example:
```
ORDM:  MOV R0,#3FH      ;MOVE '3F' HEX TO REG 0
       ORL A,@R0        ;'OR' AC CONTENTS WITH MASK
                        ;IN LOCATION 63
```

ORL A,#data Logical OR Accumulator With Immediate Mask

Encoding: | 0 1 0 0 | 0 0 1 1 | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | 43H

Description: This is a 2-cycle instruction. Data in the accumulator is logically ORed with an immediately-specified mask.

Operation: (A) ← (A) OR data

Example:
```
ORID:  ORL A,#'X'       ;'OR' ACC CONTENTS WITH MASK
                        ;01011000 (ASCII VALUE OF 'X')
```

ORL BUS,#data* Logical OR BUS With Immediate Mask

Encoding: | 1 0 0 0 | 1 0 0 0 | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | 88H

Description: This is a 2-cycle instruction. Data on the BUS port is logically ORed with an immediately-specified mask. This instruction assumes prior specification on an 'OUTL BUS,A' instruction.

Operation: (BUS) ← (BUS) OR data

Example:
```
ORBUS: ORL BUS,#HEXMSK  ;'OR' BUS CONTENTS WITH MASK
                        ;EQUAL VALUE OF SYMBOL 'HEXMSK'
```

ORL Pp, #data Logical OR Port 1 or 2 With Immediate Mask

Encoding: | 1 0 0 0 | 1 0 p p | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | 89H-8AH

Description: This is a 2-cycle instruction. Data on port 'p' is logically ORed with an immediately-specified mask.

Operation: (Pp) ← (Pp) OR data p = 1-2

Example:
```
ORP1:  ORL P1, #0FFH    ;'OR' PORT 1 CONTENTS WITH MASK
                        ;'FF' HEX (SET PORT 1 TO ALL ONES)
```

* For use with internal program memory ONLY.

** 0–5 in 8048AH/8748H
0–6 in 8049AH/8749H
0–7 in 8050AH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RET Return Without PSW Restore

Encoding: | 1 0 0 0 | 0 0 1 1 | 83H

Description: This is a 2-cycle instruction. The stack pointer (PSW bits 0–2) is decremented. The program counter is then restored from the stack. PSW bits 4–7 are not restored.

Operation: (SP) ← (SP)–1
(PC) ← ((SP))

RETR Return with PSW Restore

Encoding: | 1 0 0 1 | 0 0 1 1 | 93H

Description: This is a 2-cycle instruction. The stack pointer is decremented. The program counter and bits 4–7 of the PSW are then restored from the stack. Note that RETR should be used to return from an interrupt, but should not be used within the interrupt service routine as it signals the end of an interrupt routine by resetting the Interrupt in Progress flip-flop.

Operation: (SP) ← (SP)–1
(PC) ← ((SP))
(PSW 4–7) ← ((SP))

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RL A Rotate Left without Carry

Encoding: | 1 1 1 0 | 0 1 1 1 | E7H

Description: The contents of the accumulator are rotated left one bit. Bit 7 is rotated into the bit 0 position.

Operation: (An + 1) ← (An)
(A0) ← (A7) n = 0-6

Example: Assume accumulator contains 10110001.

```
RLNC: RL A          ;NEW ACC CONTENTS ARE 01100011
```

RLC A Rotate Left through Carry

Encoding: | 1 1 1 1 | 0 1 1 1 | F7H

Description: The contents of the accumulator are rotated left one bit. Bit 7 replaces the carry bit; the carry bit is rotatd into the bit 0 position.

Operation: (An + 1) ← (An)
n = 0-6
(A0) ← (C)
(C) ← (A7)

Example: Assume accumulator contains a 'signed' number; isolate sign without changing value.

```
RLTC: CLR C         ;CLEAR CARRY TO ZERO
      RLC A         ;ROTATE ACC LEFT, SIGN
                    ;BIT (7) IS PLACED IN CARRY
      RR A          ;ROTATE ACC RIGHT — VALUE
                    ;(BITS 0-6) IS RESTORED,
                    ;CARRY UNCHANGED, BIT 7
                    ;IS ZERO
```

RR A Rotate Right without Carry

Encoding: | 0 1 1 1 | 0 1 1 1 | 77H

Description: The contents of the accumulator are rotated right one bit. Bit 0 is rotated into the bit 7 position.

Operation: (An) ← (An + 1) n = 0-6
(A7) ← (A0)

Example: Assume accumulator contains 10110001.

```
RRNC: RR A          ;NEW ACC CONTENTS ARE 11011000
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RRC A Rotate Right through Carry

Encoding: | 0 1 1 0 | 0 1 1 1 | 67H

Description: The contents of the accumulator are rotated right one bit. Bit 0 replaces the carry bit; the carry bit is rotated into the bit 7 position.

Operation: (An) ← (An + 1) n = 0–6
(A7) ← (C)
(C) ← (A_0)

Example: Assume carry is not set and accumulator contains 10110001.

```
RRTC: RRC A          ;CARRY IS SET AND ACC
                     ;CONTAINS 01011000
```

SEL MB0 Select Memory Bank 0

Encoding: | 1 1 1 0 | 0 1 0 1 | E5H

Description: PC bit 11 is set to zero on next JMP or CALL instruction. All references to program memory addresses fall within the range 0–2047.

Operation: (DBF) ← 0

Example: Assume program counter contains 834 Hex.

```
SEL MB0              ;SELECT MEMORY BANK 0
JMP $+20             ;JUMP TO LOCATION 58 HEX
```

SEL MB1 Select Memory Bank 1

Encoding: | 1 1 1 1 | 0 1 0 1 | F5H

Description: PC bit 11 is set to one on next JMP or CALL instruction. All references to program memory addresses fall within the range 2048–4095.

Operation: (DBF) ← 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SEL RB0 Select Register Bank 0

Encoding: | 1 1 0 0 | 0 1 0 1 | C5H

Description: PSW bit 4 is set to zero. References to working registers 0–7 address data memory locations 0–7. This is the recommended setting for normal program execution.

Operation: (BS) ← 0

SEL RB1 Select Register Bank 1

Encoding: | 1 1 0 1 | 0 1 0 1 | D5H

Description: PSW bit 4 is set to one. References to working registers 0–7 address data memory locations 24–31. This is the recommended setting for interrupt service routines, since locations 0–7 are left intact. The setting of PSW bit 4 in effect at the time of an interrupt is restored by the RETR instruction when the interrupt service routine is completed.

Operation: (BS) ← 1

Example: Assume an external interrupt has occurred, control has passed to program memory location 3, and PSW bit 4 was zero before the interrupt.

Operation:

```
LOC3:  JNI INIT          ;JUMP TO ROUTINE 'INIT' IF
                         ;INTERRUPT INPUT IS ZERO
INIT:  MOV R7,A          ;MOVE ACC CONTENTS TO
                         ;LOCATION 7
       SEL RB1           ;SELECT REG BANK 1
       MOV R7,#0FAH      ;MOVE 'FA' HEX TO LOCATION 31
       .
       .
       .
       SEL RB0           ;SELECT REG BANK 0
       MOV A,R7          ;RESTORE ACC FROM LOCATION 7
       RETR              ;RETURN — RESTORE PC AND PSW
```

STOP TCNT Stop Timer/Event-Counter

Encoding: | 0 1 1 0 | 0 1 0 1 | 65H

Description: This instruction is used to stop both time accumulation and event counting.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Example: Disable interrupt, but jump to interrupt routine after eight overflows and stop timer. Count overflows in register 7..

```
START:  DIS TCNTI          ;DISABLE TIMER INTERRUPT
        CLR A              ;CLEAR ACC TO ZEROS
        MOV T,A            ;MOVE ZEROS TO TIMER
        MOV R7,A           ;MOVE ZEROS TO REG 7
        STRT T             ;START TIMER
MAIN:   JTF COUNT          ;JUMP TO ROUTINE 'COUNT'
                           ;IF TF = 1 AND CLEAR TIMER FLAG
        JMP MAIN           ;CLOSE LOOP
COUNT:  INC R7             ;INCREMENT REG 7
        MOV A,R7           ;MOVE REG 7 CONTENTS TO ACC
        JB3 INT            ;JUMP TO ROUTINE 'INT' IF ACC
                           ;BIT 3 IS SET (REG 7 = 8)
        JMP MAIN           ;OTHERWISE RETURN TO ROUTINE
                           ;MAIN


INT:    STOP TCNT          ;STOP TIMER
        JMP 7H             ;JUMP TO LOCATION 7 (TIMER)
                           ;INTERRUPT ROUTINE
```

STRT CNT Start Event Conter

Encoding:

0 1 0 0	0 1 0 1

45H

Description: The test 1 (T1) pin is enabled as the event-counter input and the counter is started. The event-counter register is incremented with each high-to-low transition on the T1 pin.

Example: Initialize and start event counter. Assume overflow is desired with first T1 input.

```
STARTC: EN TCNTI           ;ENABLE COUNTER INTERRUPT
        MOV A,#0FFH        ;MOVE 'FF'HEX (ONES) TO ACC
        MOV T,A            ;MOVES ONES TO COUNTER
        STRT CNT           ;ENABLE T1 AS COUNTER
                           ;INPUT AND START
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

STRT T Start Timer

Encoding: | 0 1 0 1 | 0 1 0 1 | 55H

Description: Timer accumulation is initiated in the timer register. The register is incremented every 32 instruction cycles. The prescaler which counts the 32 cycles is cleared but the timer register is not.

Example: Initialize and start timer.

```
STARTT: CLR A          ;CLEAR ACC TO ZEROS
        MOV T,A        ;MOVE ZEROS TO TIMER
        EN TCNTI       ;ENABLE TIMER INTERRUPT
        STRT T         ;START TIMER
```

SWAP A Swap Nibbles within Accumulator

Encoding: | 0 1 0 0 | 0 1 1 1 | 47H

Description: Bits 0-3 of the accumulator are swapped with bits 4-7 of the accumulator.

Operation: $(A_{4-7}) \rightleftarrows (A_{0-3})$

Example: Pack bits 0-3 of locations 50-51 into location 50.

```
PCKDIG: MOV R0, #50    ;MOVE '50' DEC TO REG 0
        MOV R1, #51    ;MOVE '51' DEC TO REG 1
        XCHD A,@R0     ;EXCHANGE BITS 0-3 OF ACC
                       ;AND LOCATION 50
        SWAP A         ;SWAP BITS 0-3 AND 4-7 OF ACC
        XCHD A,@R1     ;EXCHANGE BITS 0-3 OF ACC AND
                       ;LOCATION 51
        MOV @R0,A      ;MOVE CONTENTS OF ACC TO
                       ;LOCATION 50
```

XCH A,R_r Exchange Accumulator-Register Contents

Encoding: | 0 0 1 0 | 1 r r r | 28H-2FH

Description: The contents of the accumulator and the contents of working register 'r' are exchanged.

Operation: (A) $\rightleftarrows$ (Rr) r = 0-7

Example: Move PSW contents to Reg 7 without losing accumulator contents.

```
XCHAR7: XCH A,R7       ;EXCHANGE CONTENTS OF REG 7
                       ;AND ACC
        MOV A, PSW     ;MOVE PSW CONTENTS TO ACC
        XCH A,R7       ;EXCHANGE CONTENTS OF REG 7
                       'AND ACC AGAIN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XCH A,@R_i Exchange Accumulator and Data Memory Contents

Encoding: | 0 0 1 0 | 0 0 0 i | 20H–21H

Description: The contents of the accumulator and the contents of the resident data memory location addressed by bits 0–5** of register 'i' are exchanged. Register 'i' contents are unaffected.

Operation: (A) ⇆ ((Ri)) i = 0–1

Example: Decrement contents of location 52.

```
DEC52: MOV R0,#52      ;MOVE '52' DEC TO ADDRESS REG 0
       XCH A,@R0       ;EXCHANGE CONTENTS OF ACC
                       ;AND LOCATION 52
       DEC A           ;DECREMENT ACC CONTENTS
       XCH A,@R0       ;EXCHANGE CONTENTS OF ACC
                       ;AND LOCATION 52 AGAIN
```

XCHD A,@R_i Exchange Accumulator and Data Memory 4-Bit Data

Encoding: | 0 0 1 1 | 0 0 0 i | 30H–31H

Description: This instruction exchanges bits 0–3 of the accumulator with bits 0–3 of the data memory location addressed by bits 0–5** of register 'i'. Bits 4–7 of the accumulator, bits 4–7 of the data memory location, and the contents of register 'i' are unaffected.

Operation: (A_{0-3}) ⇆ ((Ri0–3)) i = 0–1

Example: Assume program counter contents have been stacked in locations 22–23.

```
XCHNIB: MOV R0,#23     ;MOVE '23' DEC TO REG 0
        CLR A          ;CLEAR ACC TO ZEROS
        XCHD A,@R0     ;EXCHANGE BITS 0-3 OF ACC AND
                       ;LOCATION 23 (BTS 8-11 OF PC ARE
                       ;ZEROED, ADDRESS REFERS
                       :TO PAGE 0)
```

XRL A,R_r Logical XOR Accumulator With Register Mask

Encoding: | 1 1 0 1 | 1 r r r | D8H–DFH

Description: Data in the accumulator is EXCLUSIVE ORed with the mask contained in working register 'r'.

Operation: (A) ← (A) XOR (Rr) r = 0–7

Example:

```
XORREG: XRL A,R5       ;'XOR' ACC CONTENTS WITH
                       ;MASK IN REG 5
```

** 0–5 in 8048AH/8748H
0–6 in 8049AH/8749H
0–7 in 8050AH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XRL A,@R_i Logical XOR Accumulator With Memory Mask

Encoding: | 1 1 0 1 | 0 0 0 i | D0H-D1H

Description: Data in the accumulator is EXCLUSIVE ORed with the mask contained in the data memory location addressed by register 'i', bits 0–5.**

Operation: (A) ← (A) XOR ((Ri)) i = 0–1

Example:
```
XORDM:  MOV R1,#20H     ;MOVE '20' HEX TO REG 1
        XRL A,@R1       ;'XOR' ACC CONTENTS WITH MASK
                        ;IN LOCATION 32
```

XRL A,#data Logical XOR Accumulator With Immediate Mask

Encoding: | 1 1 0 1 | 0 0 1 1 | | d_7 d_6 d_5 d_4 | d_3 d_2 d_1 d_0 | D3H

Description: This is a 2-cycle instruction. Data in the accumulator is EXCLUSIVE ORed with an immediately-specified mask.

Operation: (A) ← (A) XOR data

Example:
```
XORID:  XOR A,#HEXTEN   ;XOR CONTENTS OF ACC WITH MASK
                        ;EQUAL VALUE OF SYMBOL 'HEXTEN'
```

** 0–5 in 8048AH/8748H
0–6 in 8049AH/8749H
0–7 in 8050AH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผนวก ง

MT8870

Integrated DTMF Receiver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MITEL

ISO²-CMOS MT8870B/MT8870B-1
Integrated DTMF Receiver

9161-002-051-NA ISSUE 2 December 1987

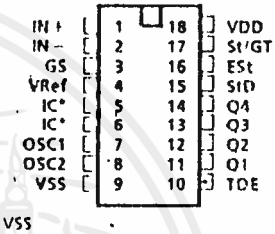
Features

- Complete DTMF Receiver
- Low Power Consumption
- Internal Gain Setting Amplifier
- Adjustable Guard Time
- Central Office Quality

Applications

- Receiver System for British Telecom (BT) or CEPT Spec (MT8870B-1)
- Paging Systems
- Repeater Systems/Mobile Radio
- Credit Card Systems
- Remote Control
- Personal Computers

Pin Connections



* Connect to VSS

Ordering Information -40°C to +85°C

MT8870BE/MT8870BE-1 Plastic DIP
MT8870BC/MT8870BC-1 Cerdip

Description

The MT8870B/MT8870B-1 is a complete DTMF receiver integrating both the bandsplit filter and digital decoder functions, fabricated in Mitel's double poly ISO²-CMOS technology. The filter section uses switched capacitor techniques for high and low group filters; the decoder uses digital counting techniques to detect and decode all 16 DTMF tone pairs into a 4-bit code. External component count is minimized by on chip provision of a differential input amplifier, clock oscillator and latched three-state bus interface.

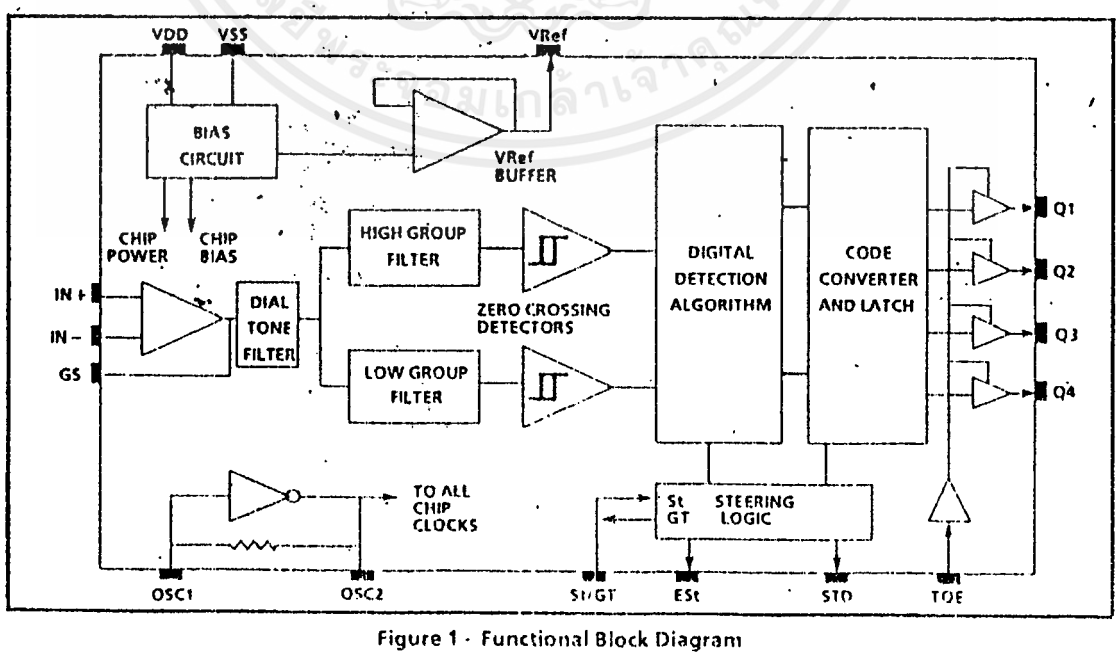


Figure 1 - Functional Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings†

	Parameter	Symbol	Min	Max	Units
1	Power supply voltage V_{DD}-V_{SS}			6	V
2	Voltage on any pin		$V_{SS}-0.3$	$V_{DD}+0.3$	V
3	Current at any pin (other than supply)			10	mA
4	Operating temperature	T_A	−40	+85	°C
5	Storage temperature		−65	+150	°C
6	Package power dissipation			1000	mW

†Exceeding these values may cause permanent damage. Functional operation under these conditions is not implied.
Derate above 75 °C at 16 mW / °C. All leads soldered to board.

Recommended Operating Conditions - Voltages are with respect to ground (V_{SS}) unless otherwise stated.

		Characteristics	Sym	Min	Typ†	Max	Units	Test Conditions
1		Positive Supply Voltages	V_{DD}		5		V	$V_{SS}=0V$
2		Oscillator Clock Frequency	fc		3.579545		MHz	
3		Oscillator Frequency Tolerance	Δfc		±0.1		%	

† Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.

DC Electrical Characteristics- $V_{DD}=5.0V\pm5\%$, $V_{SS}=0V$. Voltages are with respect to ground (V_{SS}) unless otherwise stated.

		Characteristics	Sym	Min	Typ†	Max	Units	Test Conditions
1	SUPPLY	Operating supply voltage	V_{DD}	4.75	5.0	5.25	V	
2		Operating supply current	I_{DD}		3.0	9.0	mA	
3		Power consumption	P_O		15	45	mW	f = 3.58 MHz; V_{DD} = 5V
4	INPUTS	High level input	V_{IH}	3.5			V	
5		Low level input voltage	V_{IL}			1.5	V	
6		Input leakage current	I_{IH}/I_{IL}		0.1		μA	$V_{IN}=V_{SS}$ or V_{DD}
7		Pull-up (source) current	I_{SO}		7.5	15	μA	TOE (pin 10) = 0V
8		Input impedance (IN+, IN−)	R_{IN}		10		MΩ	@ 1 kHz
9		Steering threshold voltage	V_{TSt}	2.2		2.5	V	
10	OUTPUTS	Low level output voltage	V_{OL}			$V_{SS}+0.03$	V	No load
11		High level output voltage	V_{OH}	V_{DD}-0.03			V	No load
12		Output low (sink) current	I_{OL}	1	2.5		mA	$V_{OUT}=0.4$ V
13		Output high (source) current	I_{OH}	0.4	0.8		mA	$V_{OUT}=4.6$ V
14		V_{Ref} output voltage	V_{Ref}	2.4		2.7	V	No load
15		V_{Ref} output resistance	R_{OR}		10		kΩ	

† Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Operating Characteristics[1] - Voltages are with respect to ground (V_{SS}) unless otherwise stated

Gain Setting Amplifier

	Characteristics	Sym	Min	Typ[2]	Max	Units	Test Conditions
1	Input leakage current	I_{IN}		100		nA	$V_{SS} \leq V_{IN} \leq V_{DD}$
2	Input resistance	R_{IN}		10		MΩ	
3	Input offset voltage	V_{OS}		25		mV	
4	Power supply rejection	PSRR		60		dB	1 kHz
5	Common mode rejection	CMRR		60		dB	-3.0V ≤ V_{IN} ≤ 3.0V
6	DC open loop voltage gain	A_{VOL}		65		dB	
7	Open loop unity gain bandwidth	f_c		1.5		MHz	
8	Output voltage swing	V_O		4.5		V_{pp}	$R_L \geq 100K\Omega$ to V_{SS}
9	Maximum capacitive load (GS)	C_L		100		pF	
10	Maximum resistive load (GS)	R_L		50		KΩ	
11	Common mode range	V_{CM}		3.0		V_{pp}	No Load

[1] V_{DD} = 5 V, V_{SS} = 0 V, T_A = 25°C

[2] Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.

MT8870B AC Electrical Characteristics[1] - Voltages are with respect to ground (V_{SS}) unless otherwise stated

		Characteristics	Sym	Min	Typ	Max	Units	Notes
1	SIGNAL CONDITIONS	Valid input signal levels (each tone of composite signal)		-29			dBm	1,2,3,5,6,9
				27.5			mV_{RMS}	1,2,3,5,6,9
						+1	dBm	1,2,3,5,6,9
						869	mV_{RMS}	1,2,3,5,6,9
2		Positive twist accept			10		dB	2,3,6,9
3		Negative twist accept			10		dB	2,3,6,9
4		Freq. deviation accept		±1.5% ± 2Hz			Nom.	2,3,5,9
5		Freq. deviation reject						
6		Third tone tolerance			16		dB	2,3,4,5,9
7		Noise tolerance			-12		dB	2,3,4,5,7,9,10
8		Dial tone tolerance			+22		dB	2,3,4,5,8,9,11

[1] V_{DD} = 5 V, V_{SS} = 0, T_A = 25°C and f_C = 3.579545 MHz using test circuit shown in Figure 2.

NOTES:

1. dBm = decibels above or below a reference power of 1 mW into a 600 ohm load.
2. Digit sequence consists of all DTMF tones
3. Tone duration = 40 ms, tone pause = 40 ms.
4. Signal condition consists of nominal DTMF frequencies
5. Both tones in composite signal have an equal amplitude.
6. Tone pair is deviated by ± 1.5% ± 2Hz.
7. Bandwidth limited (3KHz) Gaussian noise.
8. The precise dial tone frequencies are (350 Hz and 440 Hz) ± 2 %.
9. For an error rate of better than 1 in 10,000
10. Referenced to lowest level frequency component in DTMF signal
11. Referenced to the minimum valid accept level
12. For guard time calculation purposes.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Electrical Characteristics - Voltages are with respect to ground (V_{SS}) unless otherwise stated

		Characteristics	Sym	Min	Typ†	Max	Units	Conditions
1	TIMING	Tone present detect time	t_{DP}	5	11	14	ms	Note 12
2		Tone absent detect time	t_{DA}	0.5	4	8.5	ms	Note 12
3		Tone duration accept	t_{REC}			40	ms	User adjustable
4		Tone duration reject	$t_{\overline{REC}}$	20			ms	User adjustable
5		Interdigit pause accept	t_{ID}			40	ms	User adjustable
6		Interdigit pause reject	t_{DO}	20			ms	User adjustable
7	OUTPUTS	Propagation delay (St to Q)	t_{PQ}		8	11	µs	TOE = V_{DD}
8		Propagation delay (St to StD)	t_{PStD}		12		µs	TOE = V_{DD}
9		Output data setup (Q to StD)	t_{QStD}		3.4		µs	TOE = V_{DD}
10		Propagation delay (TOE to Q ENABLE)	t_{PTE}		50		ns	R_L = 10kΩ C_L = 50 pF
11		Propagation delay (TOE to Q DISABLE)	t_{PTD}		300		ns	R_L = 10kΩ C_L = 50 pF
12	CLOCK	Crystal /clock frequency	f_C	3.5759	3.5795	3.5831	MHz	
13		Clock input rise time	t_{LHCL}			110	ns	Ext. clock
14		Clock input fall time	t_{HLCL}			110	ns	Ext. clock
15		Clock input duty cycle	DC_{CL}	40	50	60	%	Ext. clock
16		Capacitive load (OSC2)	C_{LO}			30	pF	

† V_{DD} = 5.0V, V_{SS} = 0V, T_A = 25°C and f_c = 3.579545 MHz, using test circuit shown in Figure 2.
‡ Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.

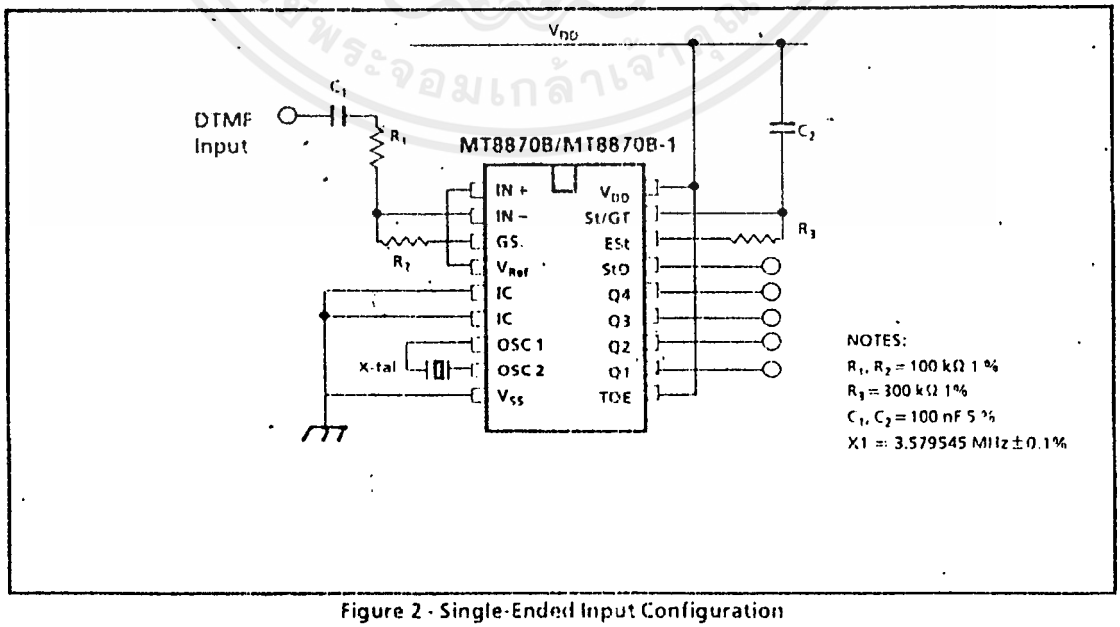


Figure 2 - Single-Ended Input Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MT8870B-1 AC Electrical Characteristics[1] - Voltages are with respect to ground (V_{SS}) unless otherwise stated

		Characteristics	Sym	Min	Typ	Max	Units	Notes
1	SIGNAL CONDITIONS	Valid input signal levels (each tone of composite signal)		-31			dBm	1,2,3,5,6,9
				21.8			mV_{RMS}	1,2,3,5,6,9
						+1	dBm	1,2,3,5,6,9
						869	mV_{RMS}	1,2,3,5,6,9
2		Input Signal Level Reject		-37			dBm	1,2,3,5,6,9
				10.9			mV_{RMS}	1,2,3,5,6,9
3		Positive twist accept				6	dB	2,3,6,9
4		Negative twist accept				6	dB	2,3,6,9
5		Freq. deviation accept		±1.5% ± 2Hz				2,3,5,9
6		Freq. deviation reject		±3.5%				2,3,5,9
7		Third tone tolerance		-18.5			dB	2,3,4,5,9,13
8		Noise tolerance			-12		dB	2,3,4,5,7,9,10
9		Dial tone tolerance			+22		dB	2,3,4,5,8,9,11

[1] V_{DD} = 5 V, V_{SS} = 0, T_A = 25° C and f_C = 3.579545 MHz using test circuit shown in Figure 2

NOTES

1. dBm = decibels above or below a reference power of 1 mW into a 600 ohm load
2. Digit sequence consists of all DTMF tones
3. Tone duration = 40 ms, tone pause = 40 ms.
4. Signal condition consists of nominal DTMF frequencies
5. Both tones in composite signal have an equal amplitude.
6. Tone pair is deviated by ± 1.5% ± 2Hz.
7. Bandwidth limited (3 kHz) Gaussian noise.
8. The precise dial tone frequencies are (350 Hz and 440 Hz) ± 2 %.
9. For an error rate of better than 1 in 10,000
10. Referenced to lowest level frequency component in DTMF signal.
11. Referenced to the minimum valid accept level.
12. For guard time calculation purposes.
13. Referenced to Fig. 10 Input DTMF Tone Level at -25 dBm (-28 dBm at 3rd Fin) Interference Frequency Range between 480-3400 Hz.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description

Pin #	Name	Description
1	IN+	Non-Inverting Op-Amp (Input).
2	IN−	Inverting Op-Amp (Input).
3	GS	Gain Select. Gives access to output of front end differential amplifier for connection of feedback resistor.
4	V_{Ref}	Reference Voltage (Output). Nominally $V_{DD}/2$ is used to bias inputs at mid-rail (see Fig.2).
5	IC	Internal Connection. Must be tied to V_{SS}.
6	IC	Internal Connection. Must be tied to V_{SS}.
7	OSC1	Clock (Input).
8	OSC2	Clock (Output). A 3.579545 MHz crystal connected between pins OSC1 and OSC2 completes the internal oscillator circuit.
9	V_{SS}	Negative Power Supply (Input).
10	TOE	Three State Output Enable (Input). Logic high enables the outputs Q1-Q4. This pin is pulled up internally.
11-14	Q1-Q4	Three State Data (Output). When enabled by TOE, provide the code corresponding to the last valid tone pair received (see Table 1). When TOE is logic low, the data outputs are high impedance.
15	StD	Delayed Steering (Output). Presents a logic high when a received tone-pair has been registered and the output latch updated; returns to logic low when the voltage on St/GT falls below V_{TSt}.
16	ESt	Early Steering (Output). Presents a logic high once the digital algorithm has detected a valid tone pair (signal condition). Any momentary loss of signal condition will cause ESt to return to a logic low.
17	St/GT	Steering Input/Guard time (Output) Bidirectional. A voltage greater than V_{TSt} detected at St causes the device to register the detected tone pair and update the output latch. A voltage less than V_{TSt} frees the device to accept a new tone pair. The GT output acts to reset the external steering time-constant; its state is a function of ESt and the voltage on St.
18	V_{DD}	Positive power supply (Input).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION

RECEIVER SYSTEM FOR BRITISH TELECOM SPEC POR 1151

The circuit shown in Fig. 10 illustrates the use of MT8870B-1 device in a typical receiver system. BT Spec defines the input signals less than -34 dBm as the non-operate level. This condition can be attained by choosing a suitable values of R_1 and R_2 to provide 3 dB attenuation, such that -34 dBm input signal will correspond to -37 dBm at the gain setting pin GS of MT8870B-1. As shown in the diagram, the component values of R_3 and C_2 are the guard time requirements when the total component tolerance is 6%. For better performance, it is recommended to use the non-symmetric guard time circuit in Fig. 9.

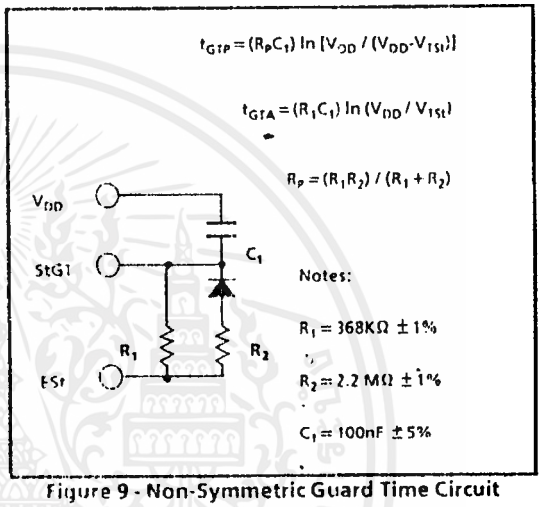


Figure 9 - Non-Symmetric Guard Time Circuit

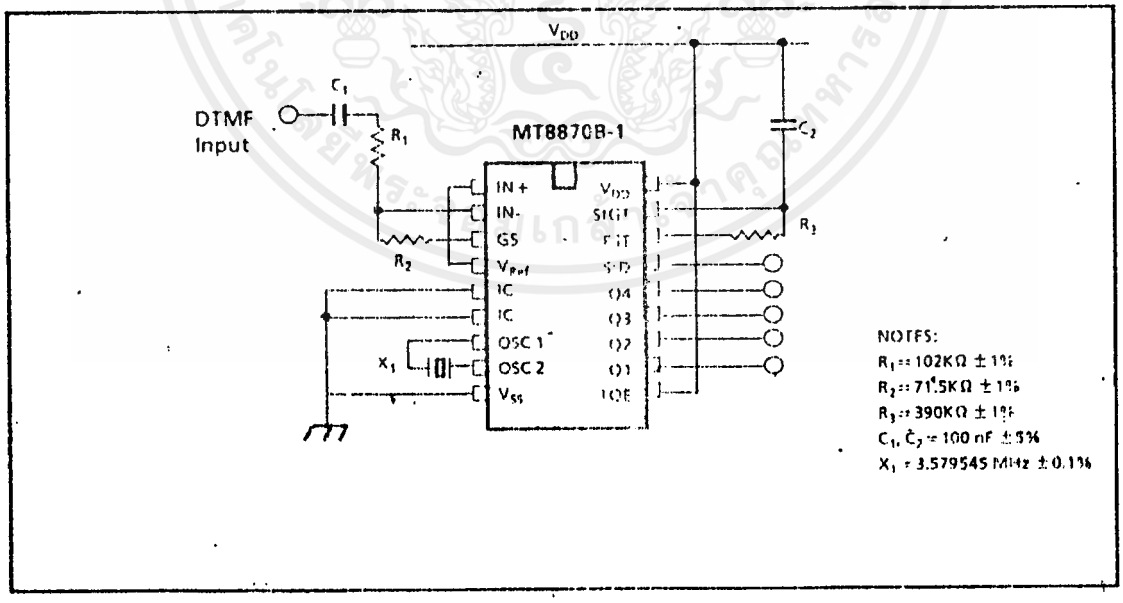


Figure 10 - Single-Ended Input Configuration for BT or CEPT Spec

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

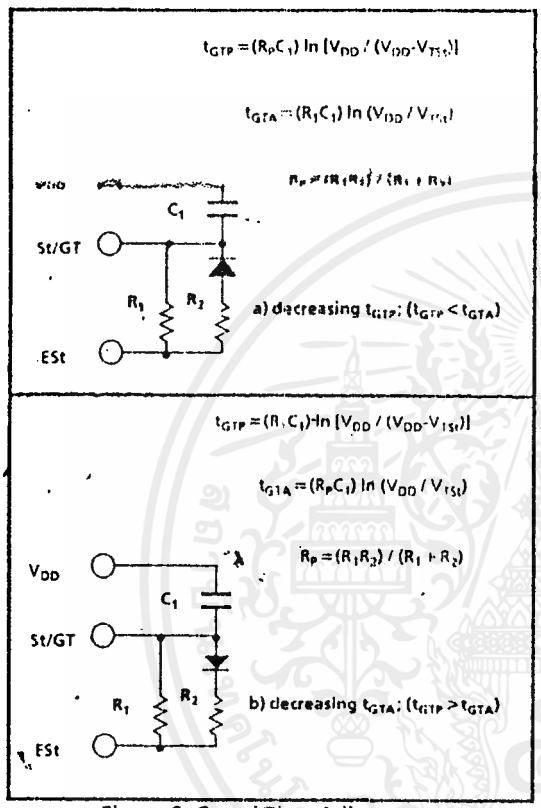


Figure 6- Guard Time Adjustment

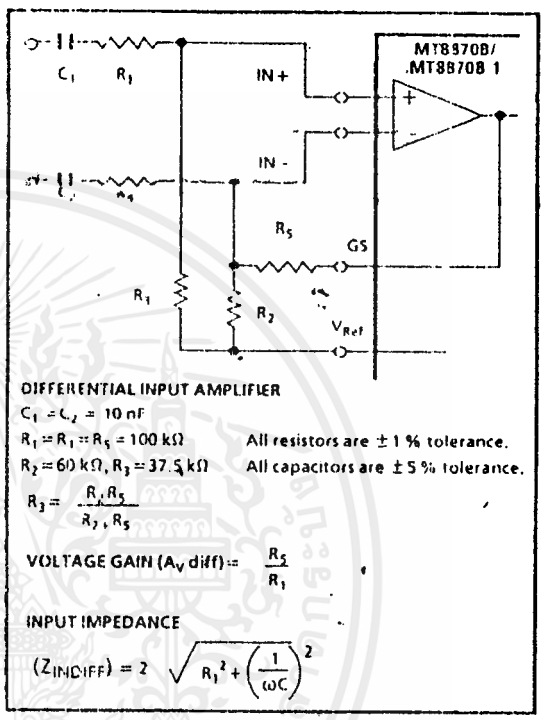


Figure 7- Differential Input Configuration

Crystal Oscillator

The internal clock circuit is completed with the addition of an external 3.579545 MHz crystal and is normally connected as shown in Figure 2 (Single Ended Input Configuration). However, it is possible to configure several MT8870B/MT8870B-1 devices employing only a single oscillator crystal. The oscillator output of the first device in the chain is coupled through a 30 pF capacitor to the oscillator input (OSC1) of the next device. Subsequent devices are connected in a similar fashion. Refer to Figure 8. for details. The problems associated with unbalanced loading are not a concern with the arrangement shown, ie; precision balancing capacitors are not required.

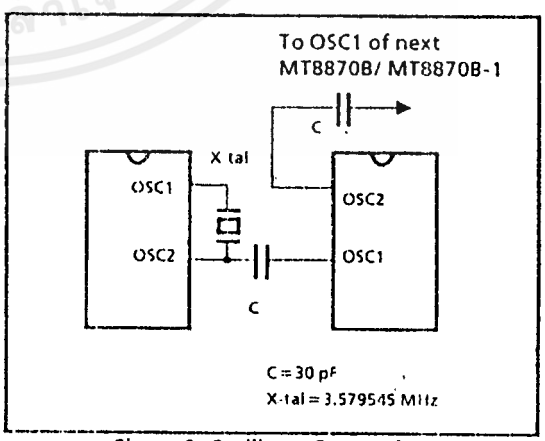


Figure 8- Oscillator Connection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

f_{LOW}	f_{HIGH}	NO.	TOE	Q_4	Q_3	Q_2	Q_1
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1336	0	H	1	0	1	0
941	1209	*	H	1	0	1	1
941	1477	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0
-	-	ANY	L	Z	Z	Z	Z

L = LOGIC LOW, H = LOGIC HIGH, Z = HIGH IMPEDANCE

Table 1. Functional Decode Table

pause between signals. Thus, as well as rejecting signals too short to be considered valid, the receiver will tolerate signal interruptions (dropout) too short to be considered a valid pause. This facility, together with the capability of selecting the steering time constants externally, allows the designer to tailor performance to meet a wide variety of system requirements.

Guard Time Adjustment

In many situations not requiring selection of tone duration and interdigital pause, the simple steering circuit shown in Figure 5 is applicable. Component values are chosen according to the formula:

$$t_{REC} = t_{DP} + t_{GTP}$$
$$t_{ID} = t_{DA} + t_{GTA}$$

The value of t_{DP} is a device parameter (see Figure 3) and t_{REC} is the minimum signal duration to be recognized by the receiver. A value for C of 0.1 μF is recommended for most applications, leaving R to be selected by the designer.

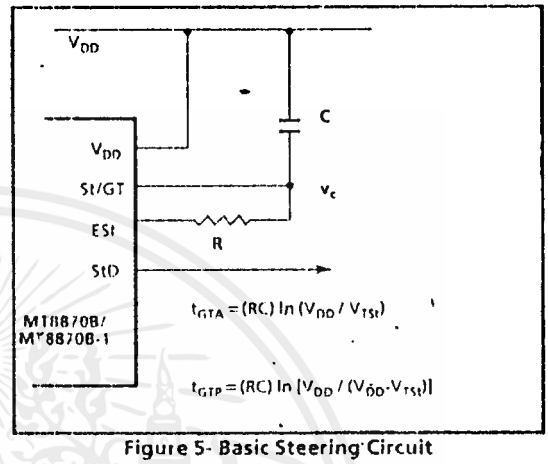


Figure 5- Basic Steering Circuit

Different steering arrangements may be used to select independently the guard times for tone present (t_{GTP}) and tone absent (t_{GTA}). This may be necessary to meet system specifications which place both accept and reject limits on both tone duration and interdigital pause. Guard time adjustment also allows the designer to tailor system parameters such as talk off and noise immunity. Increasing t_{REC} improves talk-off performance since it reduces the probability that tones simulated by speech will maintain signal condition long enough to be registered. Alternatively, a relatively short t_{REC} with a long t_{DO} would be appropriate for extremely noisy environments where fast acquisition time and immunity to tone dropouts are required. Design information for guard time adjustment is shown in Figure 6.

Differential Input Configuration

The input arrangement of the MT8870B/MT8870B-1 provides a differential-input operational amplifier as well as a bias source(V_{Ref}) which is used to bias the inputs at mid-rail. Provision is made for connection of a feedback resistor to the op-amp output (GS) for adjustment of gain. In a single-ended configuration, the input pins are connected as shown in Figure 2 with the op-amp connected for unity gain and V_{Ref} biasing the input at $\frac{1}{2}V_{DD}$. Figure 7 shows the differential configuration, which permits the adjustment of gain with the feedback resistor R_5.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description

The MT8870B/MT8870B-1 monolithic DTMF receiver offers small size, low power consumption and high performance. Its architecture consists of a bandsplit filter section, which separates the high and low group tones, followed by a digital counting section which verifies the frequency and duration of the received tones before passing the corresponding code to the output bus.

Filter Section

Separation of the low group and high group tones is achieved by applying the DTMF signal to the inputs of two sixth-order switched capacitor bandpass filters, the bandwidths of which correspond to the low and high group frequencies. The filter section also incorporates notches at 350 and 440 Hz for exceptional dial tone rejection (see Figure 4). Each filter output is followed by a single order switched capacitor filter section which smooths the signals prior to limiting. Limiting is performed by high-gain comparators which are provided with hysteresis to prevent detection of unwanted low-level signals. The outputs of the comparators provide full rail logic swings at the frequencies of the incoming DTMF signals.

Decoder Section

Following the filter section is a decoder employing digital counting techniques to determine the frequencies of the incoming tones and to verify that they correspond to standard DTMF frequencies. A complex averaging algorithm protects against tone simulation by extraneous signals such as voice while providing tolerance to small frequency deviations and variations. This averaging algorithm has been developed to ensure an optimum combination of immunity to talk-off and tolerance to the presence of interfering frequencies (third tones) and noise. When the detector recognizes the presence of two valid tones (this is referred to as the "signal condition" in some industry specifications) the "Early Steering" (ESt) output will go to an active state. Any subsequent loss of signal condition will cause ESt to assume an inactive state (see "Steering Circuit").

Steering Circuit

Before registration of a decoded tone pair, the receiver checks for a valid signal duration (referred to as character recognition condition) This check is performed by an external RC time constant driven by ESt. A logic high on ESt causes v_c (see Figure 5) to rise as the capacitor discharges. Provided signal condition is maintained (ESt remains high) for the validation period (t_{GTP}), v_c reaches the threshold (V_{TSt}) of the steering logic to register the tone pair, latching its corresponding 4-bit code (see Table 1) into the output latch. At this point the GT output is activated and drives v_c to V_{DD}. GT continues to drive high as long as ESt remains high. Finally, after a short delay to allow the output latch to settle, the delayed steering output flag (StD) goes high, signalling that a received tone pair has been registered. The contents of the output latch are made available on the 4-bit output bus by raising the three state control input (TOE) to a logic high. The steering circuit works in reverse to validate the interdigit

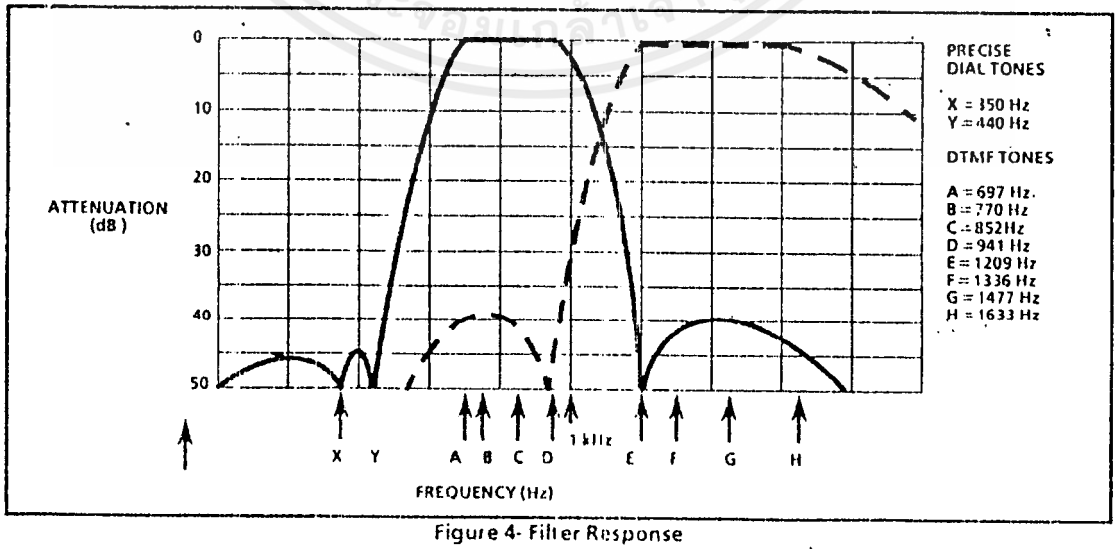
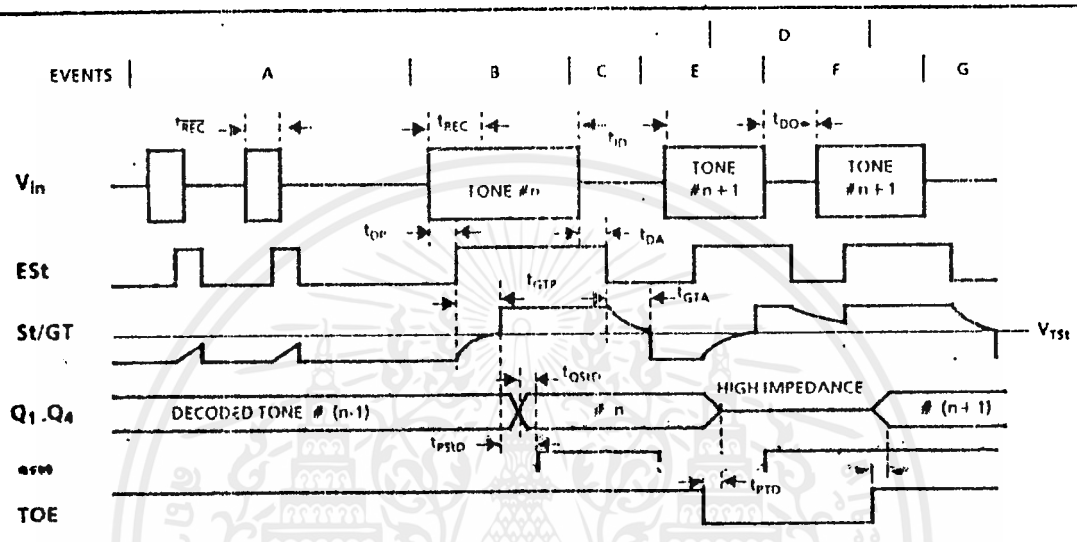


Figure 4- Filter Response

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



EXPLANATION OF EVENTS

A) TONE BURSTS DETECTED, TONE DURATION INVALID, OUTPUTS NOT UPDATED.
B) TONE #n DETECTED, TONE DURATION VALID, TONE DECODED AND LATCHED IN OUTPUTS.
C) END OF TONE #n DETECTED, TONE ABSENT DURATION VALID, OUTPUTS REMAIN LATCHED UNTIL NEXT VALID TONE.
D) OUTPUTS SWITCHED TO HIGH IMPEDANCE STATE.
E) TONE #n+1 DETECTED, TONE DURATION VALID, TONE DECODED AND LATCHED IN OUTPUTS (CURRENTLY HIGH IMPEDANCE).
F) ACCEPTABLE DROPOUT OF TONE #n+1, TONE ABSENT DURATION INVALID, OUTPUTS REMAIN LATCHED.
G) END OF TONE #n+1 DETECTED, TONE ABSENT DURATION VALID, OUTPUTS REMAIN LATCHED UNTIL NEXT VALID TONE.

EXPLANATION OF SYMBOLS

V_{in} DTMF COMPOSITE INPUT SIGNAL.
ESt EARLY STEERING OUTPUT. INDICATES DETECTION OF VALID TONE FREQUENCIES.
St/GT STEERING INPUT/GUARD TIME OUTPUT. DRIVES EXTERNAL RC TIMING CIRCUIT.
Q_1-Q_4 4-BIT DECODED TONE OUTPUT.
StD DELAYED STEERING OUTPUT. INDICATES THAT VALID FREQUENCIES HAVE BEEN PRESENT/ABSENT FOR THE REQUIRED GUARD TIME THUS CONSTITUTING A VALID SIGNAL.
TOE TONE OUTPUT ENABLE (INPUT). A LOW LEVEL SHIFTS Q_1-Q_4 TO ITS HIGH IMPEDANCE STATE.
t_{REC} MAXIMUM DTMF SIGNAL DURATION NOT DETECTED AS VALID
t_{REC} MINIMUM DTMF SIGNAL DURATION REQUIRED FOR VALID RECOGNITION
t_{ID} MINIMUM TIME BETWEEN VALID DTMF SIGNALS.
t_{DO} MAXIMUM ALLOWABLE DROPOUT DURING VALID DTMF SIGNAL
t_{DP} TIME TO DETECT THE PRESENCE OF VALID DTMF SIGNALS.
t_{DA} TIME TO DETECT THE ABSENCE OF VALID DTMF SIGNALS
t_{GTP} GUARD TIME, TONE PRESENT.
t_{GTA} GUARD TIME, TONE ABSENT.

Figure 3- Timing Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผนวก จ

การใช้ Microprocessor/Microcomputer แทนวงจรลอจิค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเลือกใช้ Microprocessor/Microcomputer

ในการออกแบบผลิตภัณฑ์มีเรื่องที่ต้องตัดสินใจอยู่ 2 เรื่องคือ

1 จะใช้ Microcomputer หรือไม่

2 และถ้าจะใช้ จะเลือกใช้ เบอร์ใด

แน่นอนว่าการตัดสินใจในทั้งสองเรื่องจะต้องทำให้ได้ผลิตภัณฑ์ที่มีราคาไม่แพง ราคาต่อหน่วยของผลิตภัณฑ์ ไม่ว่าจะใช้ Microcomputer หรือไม่ จะหาได้จาก

$$P = (F/N) + V$$

เมื่อ P คือ ราคาต่อหน่วย, F คือ ค่าใช้จ่ายคงตัว, V คือ ค่าใช้จ่ายแปรผันได้ และ N คือจำนวนหน่วยผลิตภัณฑ์ที่วางแผนผลิต

ค่าใช้จ่ายคงตัว คือค่าใช้จ่ายที่ไม่ขึ้นกับจำนวนสินค้าที่ผลิต ประกอบด้วยค่าใช้จ่ายดังต่อไปนี้

1 ค่าใช้จ่ายในการตั้งราคารวมทั้งการวิจัยตลาด

2 ค่าใช้จ่ายในการโฆษณาและส่งเสริมการขาย

3 ค่าใช้จ่ายในการเลือก Microcomputer

4 ค่าใช้จ่ายที่เกิดขึ้นในการทำให้ผลิตภัณฑ์มีคุณสมบัติตามข้อกำหนด

ค่าใช้จ่ายแปรผันได้ คือค่าใช้จ่ายที่เปลี่ยนแปลงไปตามปริมาณสินค้าที่ผลิต ประกอบด้วย

1 ราคาของอุปกรณ์ส่วนประกอบ

2 ค่าใช้จ่ายในสายการผลิต

3 ค่าใช้จ่ายในการทดสอบสินค้าก่อนนำออกสู่ตลาด

ขณะที่กำลังตัดสินใจว่าจะใช้ Microcomputer หรือ ไอซีลอจิก ต้องนำอีก 2 เรื่องต่อมาขึ้นมาพิจารณาคือ

1 การเปลี่ยนแปลงผลิตภัณฑ์ในภายหลัง

2 การบริการหลังการขาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลิตภัณฑ์ที่ถูกสร้างขึ้นโดยใช้ Microcomputer สามารถเปลี่ยนแปลงไปอย่างมากได้อย่างง่ายดายโดยการเขียนโปรแกรมขึ้นใหม่ หมายความว่าจะต้องเปลี่ยน ROM หรือ PROM ตัวหนึ่ง ในขณะที่ ผลิตภัณฑ์ที่ประกอบขึ้นจาก ไอซีลอจิกอาจต้องเปลี่ยน วงจรแผ่นพิมพ์ทั้งแผ่น หรือมากกว่านั้น

ราคาค่าบริการของผลิตภัณฑ์ที่ใช้ Microcomputer จะต่ำกว่าผลิตภัณฑ์ที่ทำขึ้นจาก ไอซีลอจิก ด้วยเหตุผล 2 ประการคือ

ประการแรก ผลิตภัณฑ์ที่สร้างขึ้นโดยใช้ Microcomputer มีอุปกรณ์น้อยชิ้นทำให้มีชิ้นส่วนที่อาจชำรุดได้น้อยกว่า และถ้ามีชิ้นส่วนที่ชำรุดจะถูกตรวจพบได้ง่ายกว่า

เหตุผลประการที่สอง เราสามารถเขียนโปรแกรมวินิจฉัย (Diagnositic Program) เพื่อตรวจสอบ อุปกรณ์ ทุกตัวบนแผ่นพิมพ์ (ใช้ LED ต่อกับ OUTPUT ต่างๆ เพื่อเป็นตัวบอกสถานะ) ถึงแม้ว่าโปรแกรมวินิจฉัยอาจทำงานไม่ได้ในวงจรที่เสียหายอย่างหนัก แต่หลักการนี้สามารถใช้ได้ดีในการใช้งานทั่วไป

เราจึงต้องเปรียบเทียบค่าใช้จ่ายในการผลิต และคำนึงถึงความประหยัดในการบริการหลังการขายด้วย (ในกรณีผลิตเพื่อจำหน่าย)

สมมุติว่าเลือกที่จะใช้ไมโครคอมพิวเตอร์ จะต้องพิจารณาว่าจะเลือกใช้เบอร์ใดหรือตระกูลใด ลองพิจารณาถึงการเลือกใช้ไมโครคอมพิวเตอร์ในแง่ค่าใช้จ่ายคงที่และค่าใช้จ่ายแปรผันได้ โดยพิจารณาค่าใช้จ่ายแปรผันได้ก่อน

อาจเห็นได้ไม่ชัดในทันที แต่ว่าชุดคำสั่งของไมโครคอมพิวเตอร์และความเร็วในการทำงาน (Execution Speed) มีผลกระทบกับค่าใช้จ่ายแปรผันได้ ในการทำงานอย่างเดียวกัน ชุดคำสั่งของ CPU ที่มีประสิทธิภาพด้อยกว่าจะต้องชุดใช้คำสั่งที่ยาวกว่าซึ่งจะมีผลกระทบกับขนาดของ MEMORY และความเร็วในการทำงาน

บางระบบใช้จำนวนชิพน้อยกว่า (ค่าใช้จ่ายแปรผันได้) ต่ำกว่าบางระบบมีค่าใช้จ่ายในการพัฒนา (ค่าใช้จ่ายคงที่) ต่ำกว่าจึงต้องพิจารณาว่าจุดใดเป็นจุดที่ค่าใช้จ่ายคงที่มีความสำคัญมากกว่าค่าใช้จ่ายแปรผันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] หนังสือคู่มือช่าง "Basic Telephony & Principle of Exchange"
ICP-537,ศูนย์ฝึกโทรคมนาคม
องค์การโทรศัพท์แห่งประเทศไทย

[2] นต.ธวัชชัย เสื่อนฉวี, หนังสือเทคโนโลยีโทรศัพท์ กรุงเทพมหานคร 2531

[3] ซีเอ็ดยูเคชั่นบริษัท จำกัด คู่มือไอซีไมโครโปรเซสเซอร์ พศ.2529

[4] นรินทร์ วัฒนกุล ดิจิตอลเบื้องต้นและไมโครคอมพิวเตอร์ชิพเดี่ยว
บริษัท ซีเอ็ดยูเคชั่น จำกัด

[5] นอ.บุญเลิศ เอี่ยมทัศนา,แฟ้ม CPU ตอน 8048 เซมิคอนดักเตอร์ อีเล็คโทรนิคส์
ฉบับที่ 69 หน้า 140-147 บริษัท ซีเอ็ดยูเคชั่น จำกัด

[6] วารสาร เซมิคอนดักเตอร์ อีเล็คโทรนิคส์ ฉบับที่ 88 หน้า 210-214 เรื่อง
IC น่าสน MT8870 บริษัท ซีเอ็ดยูเคชั่น จำกัด

[7] Osborne, Introduction to Microcomputer Volume 1

[8] Intel Corporation, MCS-48 MICROCOMPUTER USER'S MANUAL.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้