ปีการศึกษา 2532

เครื่องชุมสายโทรศัพท์สาขาอัตโนมัติ

โดย

สมศักดิ์ อมรวิกรัยสรวง

สวัสดิ์ เอิบโชคชัย

อาจารย์ที่ปรึกษา

รศ.ดร. สิทธิชัย โภไคยอุดม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2532
ภาควิชา อิเลคทรอนิคส์
คณะวิศวกรรมศาสตร์
เรื่อง เครื่องชุมสายโทรศัพท์สาขาอัตโนมัติ
ผู้จัดทำ

1 สมศักดิ์ อมรวิกรัยสรวง 29.1242
2 สวัสดิ์ เอิบโชคชัย 29.1249

อาจารย์ที่ปรึกษา

(รศ.ดร.สิทธิชัย โภไคยอุดม)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

เครื่องชุมสายโทรศัพท์สาขาอัตโนมัติ

นาย สมศักดิ์ อมรวิกรัยสรวง
นาย สวัสดิ์ เอิบโชคชัย
รศ.ดร. สิทธิชัย โภไคยอุดม
อาจารย์ที่ปรึกษา
ปีการศึกษา 2532

บทคัดย่อ

เครื่องชุมสายโทรศัพท์สาขาอัตโนมัติขนาดเล็ก ในวิทยานิพนธ์ฉบับนี้เสนอการพัฒนาระบบชุมสายขนาดเล็ก ที่มีคุณสมบัติพิเศษ คือ มีการแยกทางผ่าน สัญญาณเสียง ของสัญญาณส่งและ สัญญาณรับ ออกจากกัน เพื่อให้สามารถพัฒนาเป็น เครื่องชุมสายโทรศัพท์สาขาอัตโนมัติระบบดิจิตอลได้ ระบบเครื่องชุมสายโทรศัพท์นี้ นอกจากจะใช้ไมโครโปรเซสเซอร์ ทำการควบคุมการทำงานทั้งหมดแล้ว ยังได้นำ ไอซีในระบบโทรศัพท์มาใช้ เพื่อให้ระบบที่พัฒนามีประสิทธิภาพสูง มีความแน่นอน แม่นยำ ขนาดเล็ก โดยในส่วนไมโครโปรเซสเซอร์ได้ใช้ ชิงเกิลชิพ ตระกูล MCS-51 เป็นส่วนควบคุมระบบ เพื่อให้ระบบมีขนาดเล็ก ประหยัด และ ได้พัฒนาภาษาระดับสูงเพื่อใช้กับ ชิงเกิลชิพ ตระกูล MCS-51 ทำให้การออกแบบระบบ และ เพิ่มเติมความสามารถต่อไปในอนาคต สามารถทำได้โดยง่าย.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้ง

Private Automatic Branch Exchange (PABX)

Somsak Amonwigraisroung
Sawatt Erbchokchai
Sittichai Phokaiyaudom Ph.D.
Advisor
1989

Abstract

The private automatic branch exchange(PABX) in this thesis presents how to develop PABX system to have a special capability in saparate speech paths to transmittion and reception signal for development to digital PABX. The system , by using microprocessor to control all operations system, uses IC in telephone system. It has good efficiency and reliability ,small size and more economical. And we develop high-level language for using with single-chip in the family of MCS-51. Besides adding the ability any further in the future can be done easily.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เนื้อเรื่อง	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	
2.1 ปัญหาและวิเคราะห์เครื่องชุมสายโทรศัพท์สาขาอัตโนมัติ	2
2.2 หลักการทำงานของเครื่องชุมสายโทรศัพท์สาขาอัตโนมัติ	3
2.3 ระบบสัญญาณตามของโทรศัพท์	6
2.4 ระบบไมโครโปรเซสเซอร์ MCS-51	8
บทที่ 3 การคำนวณและการสร้าง	
3.1 ส่วนเชื่อมต่อสายโทรศัพท์ภายใน	10
3.2 ส่วนเชื่อมต่อสายโทรศัพท์ภายนอก	12
3.3 ส่วนตรวจสอบสัญญาณเรียก	14
3.4 ส่วนสวิทช์ชิ่ง	16
3.5 ส่วนสร้างสัญญาณตามของโทรศัพท์	18
3.6 ส่วนตรวจสอบการกดหมายเลข	20
3.7 ส่วนควบคุมการทำงานของระบบ	22
บทที่ 4 การทดลองและผลการทดลอง	24
บทที่ 5 สรุปและวิจารณ์	25
ภาคผนวก ก. วิธีใช้งานเครื่องชุมสายโทรศัพท์อัตโนมัติ	
ข. รายละเอียดของ Compiler	
ค. Listing Program	
ง. รายละเอียดของไอซีต่างๆ	
กิตติกรรมประกาศ	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ในปัจจุบันความเจริญก้าวหน้าของวิทยาการทางด้านเทคโนโลยี่เกือบทุกสาขาได้เจริญก้าวหน้าไปอย่างรวดเร็ว เทคโนโลยีทางด้านระบบการสื่อสารก็ถือเป็นสิ่งจำเป็นอย่างยิ่งในการดำรงชีวิต และประกอบธุรกิจ เพราะถ้ามีการสื่อสารที่ดีจะทำให้เกิดการพัฒนาทางเศรษฐกิจที่ดีด้วย ระบบสื่อสารที่นิยมใช้กันในปัจจุบันมีมากมายที่จะตอบสนองความต้องการของมนุษย์ ระบบโทรศัพท์เป็นระบบสื่อสารระบบหนึ่งที่มีความสำคัญอย่างยิ่ง เพราะสะดวก รวดเร็ว ประหยัด และง่ายต่อการใช้งาน ระบบโทรศัพท์ จึงเป็นระยบบที่น่าสนใจในการค้นคว้า และพัฒนาให้เจริญก้าวหน้าต่อไป การที่จะติดต่อจากโทรศัพท์เครื่องหนึ่งไปยังโทรศัพท์อีกเครื่องหนึ่งนั้น จำเป็นต้องผ่านชุมสายโทรศัพท์ ดังนั้นการพัฒนาเครื่องชุมสายโทรศัพท์ให้ดีขึ้น จะทำให้ระบบสื่อสารทางโทรศัพท์มีประสิทธิภาพและก้าวหน้ายิ่งขึ้น

ปริญญานิพนธ์ฉบับนี้จะเสนอเครื่องชุมสายโทรศัพท์สาขาอัตโนมัติ(Private Automatic Branch Exchenge หรือ PABX)ที่ควบคุมการทำงานด้วยระบบไมโครโปรเซสเซอร์ ซึ่งมีผลทำให้การทำงานของระบบถูกต้อง แม่นยำ มากยิ่งขึ้น ขนาดของเครื่องมีขนาดเล็ก และเป็นระบบอัตโนมัติคือเครื่องโทรศัพท์จากภายนอกสามารถติดต่อเข้ามายังเครื่องโทรศัพท์ภายในหรือเครื่องลูกได้โดยตรง ไม่ต้องมีโอเปอร์เรเตอร์(operator) โดยอาศัยการกดหมายเลขระบุเครื่องลูกที่ต้องการติดต่ออีก 1 ตัวหลังจากการกดหมายเลขโทรศัพท์ทั้ง 7 ตัว เพื่อแจ้งผ่านองค์การโทรศัพท์แล้ว และทางผ่านของสัญญาณเสียงในเครื่องชุมสายโทรศัพท์นี้ ได้ใช้ไอซีสวิทส์ทำหน้าที่ตัดต่อระหว่างสัญญาณรับ-ส่ง แทนระบบเกาซึ่งยังใช้รีเลย์(relay)อยู่ ทำให้การถ่ายทอดของสัญญาณเสียงเป็นไปได้อย่างรวดเร็ว และเกิดประสิทธิภาพมากยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎี และ หลักการ

เครื่องชุมสายโทรศัพท์สาขาอัตโนมัติ คือ เครื่องที่ใช้ขยายคู่สายโทรศัพท์จากภายนอกให้สามารถนำมาใช้ภายในได้มากขึ้น เพื่ออำนวยความสะดวกในการใช้งาน และระบบตัดต่อยังใช้ระบบอัตโนมัติ โดยไม่ต้องมีพนักงานต่อสาย คือให้ผู้โทรศัพท์จากภายนอก เป็นผู้ทำหน้าที่แจ้งให้เครื่องทราบว่าต้องการติดต่อกับเครื่องใดเท่านั้น หลังจากนั้น เครื่องจะทำหน้าที่ตัดต่อสายให้เอง

2.1 ขีดความสามารถของเครื่องชุมสายโทรศัพท์สาขาอัตโนมัติ

เครื่องชุมสายอัตโนมัติที่พัฒนาขึ้นมีความสามารถต่างๆดังนี้

1 สามารถใช้ได้กับคู่สายโทรศัพท์ระบบโทน (DTMF Tone) หรือระบบกดปุ่มเท่านั้น แต่สามารถพัฒนาให้ใช้ได้กับระบบพัลซ์(Pulse) หรือระบบหมุนได้โดยการพัฒนาซอฟท์แวร์(softwere)เพื่อให้ไมโครโปรเซสเซอร์นับจำนวนพัลซ์เพื่อตรวจสอบการหมุนหมายเลขก็ได้

2 สามารถขยายคู่สายโทรศัพท์ของหมายเลขจากองค์การโทรศัพท์จาก 1 คู่สาย ให้เป็น 3 คู่สายภายใน

3 เครื่องลูกสามารถติดต่อถึงกันได้ โดยกดหมายเลขระบุเครื่องที่ต้องการติดต่อ เช่น 1,2,3

4 ถ้าเครื่องภายนอกไม่ระบุเครื่องลูกที่ต้องการติดต่อคือไม่กดหมายเลขอีก 1 ตัวหลังจากกดหมายเลข 7 ตัว หรือกดผิดพลาดแล้วเครื่องชุมสายจะทำหน้าที่ต่อคู่สายนอกเข้ากับเครื่องลูกเครื่องที่หนึ่ง (1) ให้โดยอัตโนมัติ

5 ขณะที่เครื่องลูกต่อพูดกันหรือต่อพูดกับสายนอก เครื่องลูกอื่นที่ไม่ใช่คู่สนทนาจะแอบฟังไม่ได้

6 เครื่องลูกสามารถเรียกออกข้างนอกได้โดยกด "0"หลังจากนั้นจะได้ยินสัญญาณไดอัล(Dial Tone) จากชุมสายขององค์การโทรศัพท์

7 สามารถโอนสายนอกให้กับเครื่องลูกเครื่องใดเครื่องหนึ่งได้

8 กรณีไฟดับ เครื่องชุมสายโทรศัพท์นี้ จะทำการต่อคู่สายนอกเข้ากับเครื่องลูกเครื่องที่หนึ่ง (1)โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 หลักการทำงานของเครื่องชุมสายโทรศัพท์สาขาอัตโนมัติ

เครื่องชุมสายโทรศัพท์สาขาอัตโนมัติที่พัฒนาขึ้นมีส่วนประกอบดังรูปที่ 2.1 และแต่ละส่วนมีหน้าที่ดังนี้

ส่วนเชื่อมต่อสายโทรศัพท์ภายนอก ใช้ ไอซีเทเลโฟน สปีช เนทเวอร์ค (Telephone Speech Network) ทำหน้าที่แปลงสัญญาณจากคู่สายนอกที่ติดต่อกับองค์การโทรศัพท์ให้เป็นสัญญาณรับ-ส่งที่แยกจากกัน

ส่วนเชื่อมต่อสายโทรศัพท์ภายใน ใช้ไอซี ซับสไคร์เบอร์ลูป อินเทอร์เฟส เซอร์กิต (Subscriber Loop Interface Circuit หรือ SLIC) ทำหน้าที่จ่ายไฟเลี้ยงให้กับเครื่องโทรศัพท์ภายใน และแปลงสัญญาณจากเครื่องโทรศัพท์ให้เป็นสัญญาณรับ-ส่งที่แยกจากกัน

ส่วนสวิทส์ชิ่ง ใช้ไอซี ครอส พอย์ท สวิทส์ (Cross Point Switch) ทำหน้าที่เป็นทางผ่านของสัญญาณเสียง โดยทำการตัดต่อระหว่างสัญญาณรับ-ส่งของคู่โทรศัพท์ ที่ต้องการติดต่อกัน ซึ่งส่วนสวิทส์ชิ่งนี้ ถูกควบคุมการทำงานด้วย ส่วนของไมโครโปรเซสเซอร์

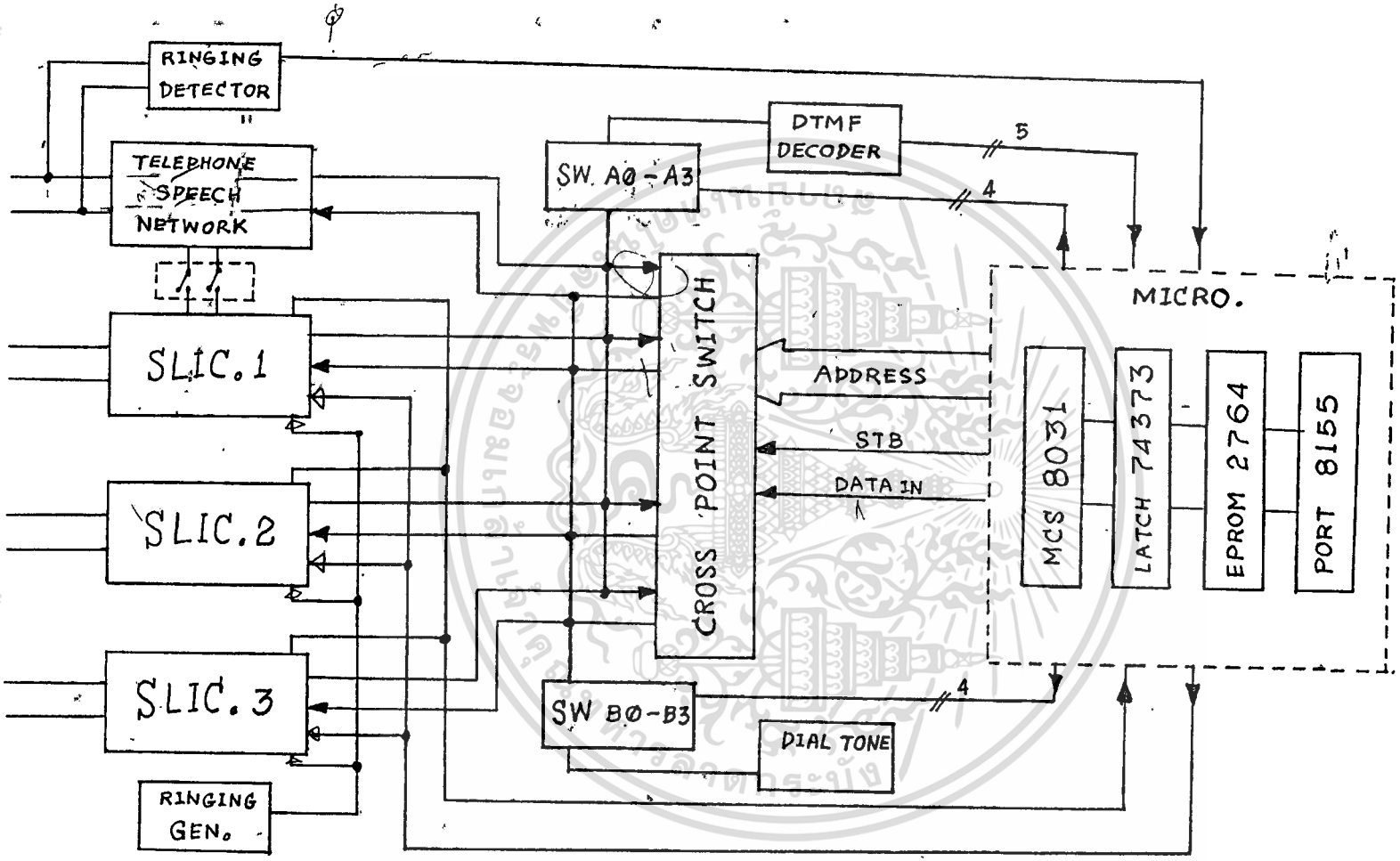
ส่วนตรวจสอบสัญญาณเรียก (Ringing Detector) จะทำหน้าที่ในการตรวจสอบว่ามีสัญญาณเรียกจากคู่สายนอกหรือไม่ ถ้ามีการเรียกจากคู่สายนอกก็จะส่งสัญญาณไปให้ไมโครโปรเซสเซอร์ทราบ

ส่วนตรวจสอบการกดหมายเลข (DTMF Decoder) จะทำหน้าที่แปลงสัญญาณความถี่ที่เกิดจากการกดปุ่มของเครื่องโทรศัพท์จากคู่สายภายนอกหรือคู่สายภายในที่ถูกกำหนดด้วยสวิทส์ A0-A3 ให้เป็นเลขแบบ ไบนารี่ โค๊ด เดซิมอล (Binary Code Decimal หรือ BCD) แล้วแจ้งให้ไมโครโปรเซอร์ทราบถึงหมายเลขที่กด และมีการตรวจสอบความถูกต้องของสัญญาณกดด้วย

ส่วนสร้างสัญญาณไดอัล (Dial Tone) ทำหน้าที่สร้างสัญญาณเสียงเพื่อส่งให้โทรศัพท์เครื่องลูก และคู่สายจากภายนอกทราบว่า ขณะนี้ไมโครโปรเซสเซอร์พร้อมที่จะรับสัญญาณจากการกดหมายเลข หรือเมื่อต้องการสร้างเป็นสัญญาณไม่ว่าง หรือสัญญาณเรียกกลับ ก็สามารถทำได้โดยการควบคุมการปิด-เปิดของสวิทส์ B0-B3

ส่วนสร้างสัญญาณเรียก (Ringing Generator) ทำหน้าที่สร้างสัญญาณเรียก แล้วควบคุมการปิด-เปิด จากไมโครโปรเซสเซอร์ เพื่อส่งเข้าสู่โทรศัพท์ เพื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.1 บล็อกไดอะแกรม ของเครื่องชุมสายโทรศัพท์สาขาอัตโนมัติ

การทำงานของเครื่องชุมสายโทรศัพท์สาขาอัตโนมัติมีหลักการดังนี้

1 ส่วนไมโครโปรเซสเซอร์จะตรวจสถานะการเรียกของคู่สายภายนอกจากส่วนตรวจสอบสัญญาณเรียก และตรวจสอบการเรียกใช้งานของคู่สายภายใน โดยตรวจสถานะการยกหู (on hook) หลังจากนั้นเครื่องก็จะส่งสัญญาณไดอัล เพื่อให้เครื่องที่ต้องการเรียกกดหมายเลขของเครื่องที่ต้องการติดต่อด้วย หมายเลขจากการกดจะถูกถอดรหัส โดยส่วนตรวจสอบหมายเลขการกด แล้วแจ้งให้ไมโครโปรเซสเซอร์ทราบว่าต้องการติดต่อกับเครื่องลูกภายในเครื่องใด หรือต้องการติดต่อกับคู่สายภายนอก

2 กรณีผู้เรียกต้องการติดต่อกับเครื่องลูกภายใน ไมโครโปรเซสเซอร์ก็จะทำการตรวจเช็คว่าเครื่องนั้นใช้งานอยู่หรือไม่ โดยตรวจสอบจากสถานะการยกหู ถ้าเครื่องที่ต้องการติดต่อถูกใช้งานอยู่ ไมโครโปรเซสเซอร์ก็จะส่งสัญญาณไม่ว่างกลับไปให้ผู้เรียกใช้งานทราบว่าเครื่องที่ต้องการติดต่อด้วยนั้นถูกใช้งานอยู่ หรือ ไม่ว่าง โดยการควบคุมการปิด-เปิดของสวิทส์ B1-B3 เพื่อให้เกิดสัญญาณไม่ว่าง แต่ถ้าเครื่องที่ต้องการติดต่อว่างอยู่คือไม่ถูกใช้งาน ไมโครโปรเซสเซอร์ก็จะส่งสัญญาณเรียกไปยังเครื่องที่ต้องการติดต่อด้วย โดยควบคุมการปิด-เปิด RN1-RN3 ของ SLIC ทั้ง 3 เพื่อให้เกิดสัญญาณเรียกที่โทรศัพท์เครื่องลูก และจะส่งสัญญาณเรียกกลับไปยังเครื่องที่เรียกใช้งานเพื่อบอกให้ทราบว่า เครื่องที่ต้องการติดต่อด้วยนั้นว่างอยู่และสามารถติดต่อได้โดยควบคุมการปิด-เปิดของสวิทส์ B1-B3 เพื่อให้เกิดสัญญาณเรียกกลับ และเมื่อมีผู้รับสายแล้ว ก็จะต่อทางผ่านสัญญาณเสียงของทั้ง 2 คู่สายเข้าด้วยกัน จึงสามารถพูดกันได้

3 กรณีผู้เรียกจากเครื่องลูกภายในต้องการติดต่อกับเครื่องภายนอก หรือ คู่สายภายนอกเครื่องชุมสายโทรศัพท์ก็จะต่อทางผ่านสัญญาณเสียงของคู่สายที่เรียกใช้งานเข้ากับคู่สายภายนอก หลังจากนั้นผู้เรียกก็จะได้ยินสัญญาณไดอัล จากองค์การโทรศัพท์แล้วก็สามารถกดหมายเลขของผู้ที่ต้องการติดต่อด้วยเหมือนกับการใช้โทรศัพท์โดยทั่วไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 ระบบสัญญาณต่างๆของโทรศัพท์

ลักษณะของสัญญาณต่างๆภายในระบบเครื่องชุมสายโทรศัพท์สาขาอัตโนมัติที่พัฒนาขึ้นเป็นดังนี้

1 สัญญาณให้หมุน (Dial Tone) คือ สัญญาณที่เครื่องชุมสายโทรศัพท์แจ้งให้ผู้ใช้โทรศัพท์ทราบว่า เครื่องชุมสายพร้อมแล้วที่จะให้ผู้ใช้โทรศัพท์กดหมายเลขโทรศัพท์ที่ต้องการจะติดต่อด้วย

ลักษณะของสัญญาณ เป็นสัญญาณที่มีความถี่ประมาณ 400 เฮิร์ท ดังต่อเนื่องกันตลอด

2 สัญญาณไม่ว่าง (Busy Tone) คือสัญญาณที่เครื่องชุมสายโทรศัพท์แจ้งผู้ใช้โทรศัพท์ หลังจาก กดหมายเลขที่ต้องการจะติดต่อด้วย ให้ทราบว่าไม่สามารถติดต่อคู่สายนั้นได้

ลักษณะของสัญญาณ เป็นสัญญาณที่มีความถี่ประมาณ 400 เฮิร์ท ดัง และ ดับทุกๆ 0.5 วินาที

3 สัญญาณเรียกกลับ (Ringback Tone) คือสัญญาณที่เครื่องชุมสายโทรศัพท์แจ้งผู้ใช้โทรศัพท์ หลังจากกดหมายเลขที่ต้องการ จะติดต่อแล้วให้ทราบว่า สามารถติดต่อคู่สายโทรศัพท์ที่ต้องการติดต่อได้

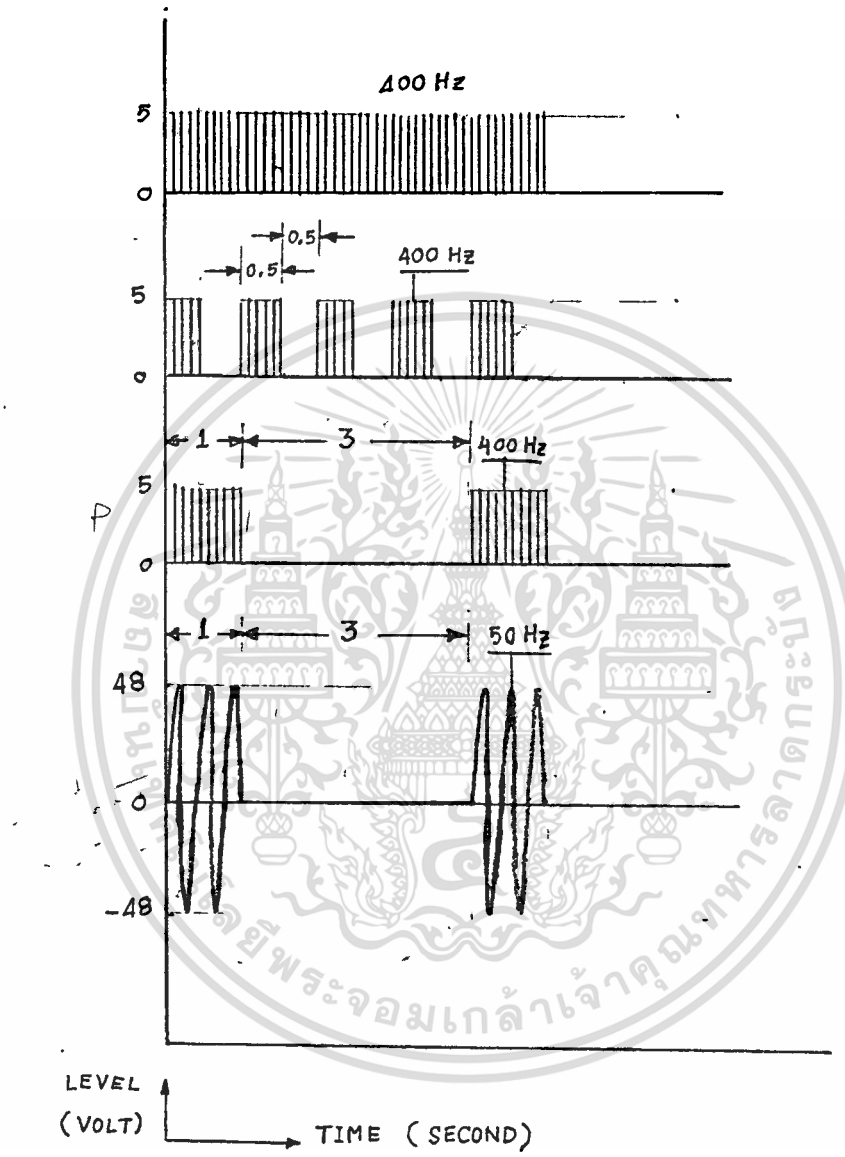
ลักษณะของสัญญาณ เป็นสัญญาณที่มีความถี่ประมาณ 400 เฮิร์ท ดังประมาณ 1 วินาที และดับประมาณ 3 วินาที

4 สัญญาณเรียก (Ringing Tone) คือสัญญาณที่เครื่องชุมสายโทรศัพท์ ส่งไปยังเครื่องโทรศัพท์ภายใน ทำให้กระดิ่งโทรศัพท์ดัง เพื่อแจ้งให้ทราบว่ามีผู้ต้องการติดต่อด้วย

ลักษณะของสัญญาณ เป็นสัญญาณ เอ ซี 50 เฮิร์ท (AC 50 Hz) ที่มีขนาดประมาณ 48 โวลต์ ดัง และ ดับ เป็นช่วงๆ เหมือนและพร้อมกับสัญญาณเรียกกลับ

ลักษณะและช่วงเวลาของสัญญาณต่างๆ ของโทรศัพท์แสดงดังรูป2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.2 ลักษณะและช่วงเวลา ของสัญญาณต่างๆ ของโทรศัพท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ระบบไมโครโปรเซสเซอร์ MCS-51

8031 เป็น ไมโครโพรเซสเซอร์ชิงเกิลชิพของบริษัทอินเทลตระกูล MCS-51 ซึ่งมีข้อได้เปรียบที่เหนือกว่า Z80 มากมาย ดังจะกล่าวถึงข้อดีต่าง ๆ ได้ดังนี้

1. มีหน่วยความจำ (RAM) ภายใน 128 ไบท์ โดยได้แบ่งออกเป็น รีจีสเตอร์ R0-R7 (4 bank), สแต็ก (STACK) และที่เหลือจากนั้นสามารถใช้ได้อย่างอิสระ โดยสามารถอ้างถึงได้ทั้งเป็น byte และ bit ทำให้ผู้ใช้สามารถนำ RAM ส่วนนี้เอาไว้ใช้ในโปรแกรมได้โดยสะดวก

2. มี พอร์ท ภายนอกถึง 4 พอร์ท ดังนี้

ก) พอร์ท 0 เมื่อทำงานกับหน่วยความจำ (ROM) หรือ หน่วยความจำ (RAM) ภายนอกจะมีสัญญาณของ address ไบท์ต่ำและ data มัลติเพล็กซ์กันโดยจะแยกสัญญาณทั้งสองออกจากกันได้โดยใช้สัญญาณ ALE มาทำการ latch

ข) พอร์ท 1 สามารถใช้เป็นอินพุทและเอาต์พุท (Input/Output) โดยอิสระ

ค) พอร์ท 2 เมื่อมีการติดต่อกับหน่วยความจำ ROM หรือ RAM ภายนอก จะมี address ไบท์สูงส่งออกมาที่พอร์ทนี้เพื่อประกอบกับ พอร์ท 0 รวมเป็น 16 BIT

ง) พอร์ท 3 ใช้เป็นฟังก์ชันพิเศษมีหน้าที่ต่าง ๆ ดังนี้

- WR เป็นขาสำหรับ สัญญาณ WRITE หน่วยความจำภายนอก
- RD เป็นขาสำหรับ สัญญาณ READ หน่วยความจำภายนอก
- T0 เป็นขาอินพุทสำหรับ counter 0
- T1 เป็นขาอินพุทสำหรับ counter 1
- INT0 ใช้อินเตอร์รัพท์ 0
- INT1 ใช้อินเตอร์รัพท์ 1
- RXD ใช้รับข้อมูลแบบ serial
- TXD ใช้ส่งข้อมูลแบบ serial

ถ้าบางฟังก์ชันพิเศษเหล่านี้ไม่ได้ใช้งาน ก็สามารถใช้เป็นอินพุทหรือ เอาต์พุทตามปกติได้

3. มีวงจรนับ/วงจรตั้งเวลา (Counter/Timer) ภายใน 2 ตัว โดยสามารถโปรแกรมให้ทำงานได้ ทั้งวงจรนับ และ วงจรตั้งเวลา

4. มีอินเตอร์เฟสข้อมูลแบบอนุกรม (Serial interface) อยู่ภายในเป็นชนิด Full duplex สามารถโปรแกรมให้มีการ INTERRUPT เมื่อรับหรือส่งครบไบท์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก) โหมด 0 ข้อมูลจะรับและส่งแบบ 8 บิท BAUD RATE = 1/12 ของความถี่สัญญาณนาฬิกา

ข) โหมด 1 ข้อมูลจะถูกส่งและรับแบบ 10 บิท โดยเพิ่ม START,STOP BIT และสามารถโปรแกรม BUAD RATE ได้

ค) โหมด 2 ข้อมูลจะถูกส่งและรับแบบ 11 บิท โดยเพิ่มข้อมูลขึ้นมาอีก 1 บิท (PROGRAM BIT)ซึ่งใช้เป็นพาริตี้บิทได้สามารถโปรแกรม BUAD RATE ได้ 2 ค่า 1/32 หรือ 1/64 ของความถี่สัญญาณนาฬิกา

ง) โหมด 3 เหมือนกับ โหมด 2 แต่ สามารถโปรแกรม BUAD RATE ได้

4. มี INTERRUPT จาก 5 แหล่งที่สามารถกำหนด ความสำคัญได้

ก) INTERRUPT 0 จะกระโดดไปที่ ADDRESS 0003H

ข) INTERRUPT 1 จะกระโดดไปที่ ADDRESS 0013H

ค) INTERRUPT timer0 จะกระโดดไปที่ ADDRESS 000BH

ง) INTERRUPT timer1 จะกระโดดไปที่ ADDRESS 001BH

จ) INTERRUPT จาก SERIAL PORT จะกระโดดไปที่ ADDRESS 002BH

5. มีวงจรสร้างสัญญาณนาฬิกาภายในเพียงแต่ต่อคริสตัลภายนอก และ ตัวเก็บประจุก็สามารถสร้างสัญญาณนาฬิกาให้กับซีพียูได้ทันที

6. ใช้ REGISTER ในการควบคุมอุปกรณ์ภายใน อินเตอร์รัพท์ และพอร์ทต่างๆ เรียกว่า SFR (Special Function Register) ในลักษณะ direct addressing mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

026944

บทที่ 3

การคำนวน และ การสร้าง

จากบทที่แล้วทำให้ทราบถึง การทำงานหลักของเครื่องชุมสายโทรศัพท์ ในบทนี้จะอธิบายถึงหน้าที่ และการทำงานของแต่ละส่วนโดยละเอียด ดังต่อไปนี้

3.1 ส่วนเชื่อมต่อสายโทรศัพท์ภายใน

หน้าที่ของส่วนเชื่อมต่อสายโทรศัพท์ภายในมีดังนี้

3.1.1 ส่วนนี้จะทำการจ่ายไฟ ดี ซี ให้กับโทรศัพท์เครื่องลูก เพื่อให้เครื่องโทรศัพท์สามารถทำงานได้

3.1.2 ทำหน้าที่แยกสัญญาณจากเครื่องโทรศัพท์ ให้เป็นสัญญาณรับ-ส่ง เพื่อนำไปสวิทส์ติดต่อกับคู่สายที่ต้องการ

3.1.3 สามารถปรับอัตราขยาย(Gain) ของสัญญาณรับ-ส่งของเครื่องโทรศัพท์ได้

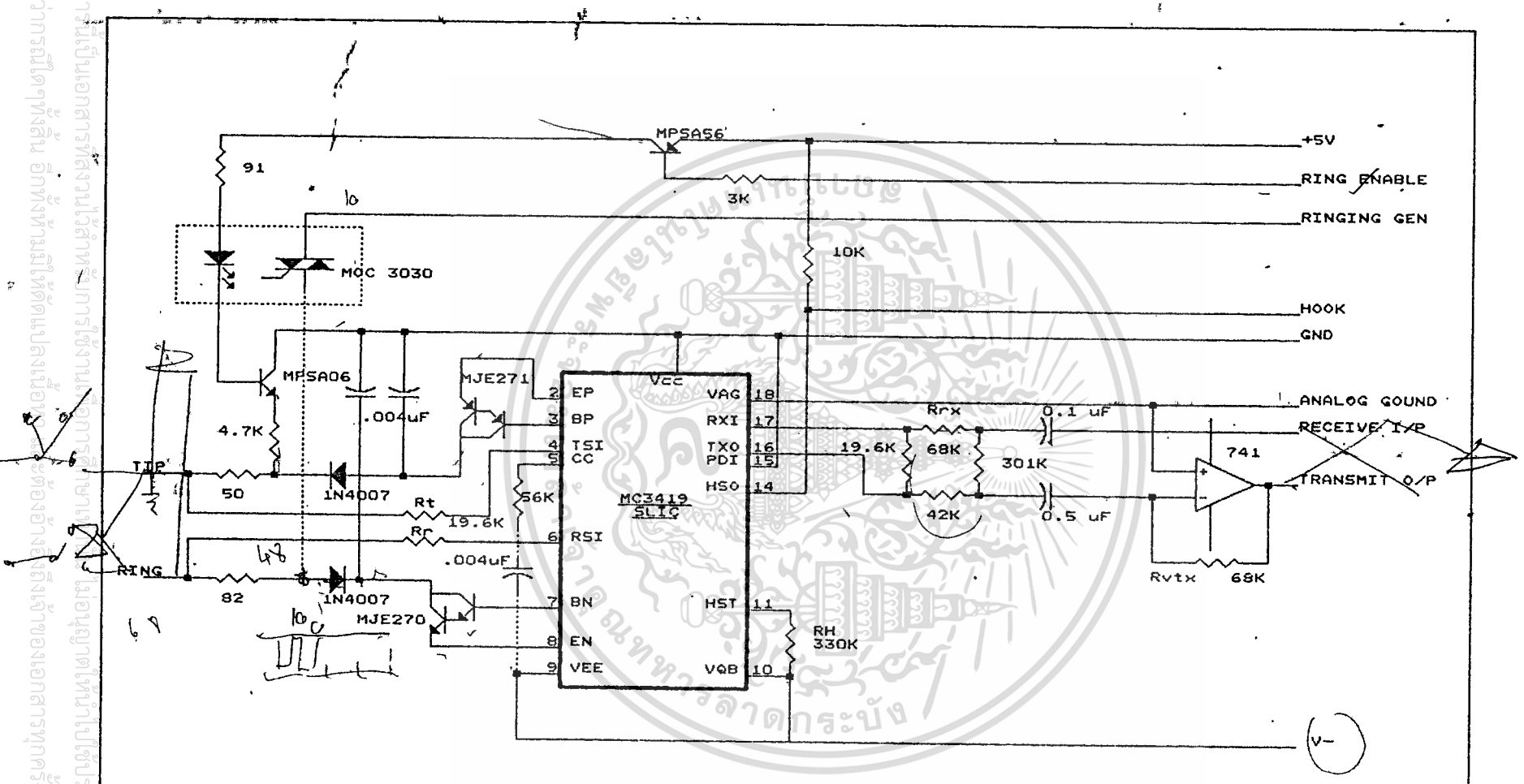
3.1.4 แสดงสถานะการยกหู และควบคุมการส่งสัญญาณเรียก

3.1.5 ป้องกันโอเวอร์โวลต์เตจ

การทำงานของวงจรส่วนเชื่อมต่อสายโทรศัพท์ภายใน แสดงดังรูปที่ 3.1

จากวงจรรูปที่ 3.1 ส่วนนี้ใช้ไอซี ซับสไครเบอร์ลูป อินเทอร์เฟส เซอร์กิต MC 3419 เป็นตัวหลักในการทำงาน โดยจะทำหน้าที่แยกสัญญาณจากคู่สายโทรศัพท์ให้เป็นสัญญาณรับ-ส่ง ที่แยกกัน หลังจากนั้นสัญญาณส่ง จะถูกขยายโดยออปแอม ซึ่งการปรับอัตราขยายของสัญญาณส่งก็จะปรับที่อัตราขยายของออปแอม และการปรับอัตราขยายของสัญญาณรับทำได้โดย การเปลี่ยนค่า Rrx คือ ถ้าค่า Rrx ลดลง อัตราขยายของสัญญาณรับก็จะเพิ่มขึ้น ขา 14 ของไอซี MC 3419 จะแสดงสถานะการยกหู คือ ถ้าไม่มีการยกหู จะให้สถานะลอจิกเป็น "1" (5V) และเมื่อมีการยกหูก็จะให้สถานะลอจิกเป็น "0" (0V) สถานะการยกหูนี้จะถูกอ่านโดยไมโครโปรเซสเซอร์ เพื่อให้ไมโครโปรเซสเซอร์รู้ถึงความต้องการใช้งานของเครื่องโทรศัพท์ภายใน หรือ ใช้ตรวจสถานะการใช้งานเมื่อมีผู้ต้องการติดต่อด้วย การควบคุมสัญญาณเรียกก็สามารถทำได้โดยการให้สถานะลอจิก "0" ที่ขาเบสของทรานซิสเตอร์ MPSA56 เพื่อให้ทรานซิสเตอร์ MPSA56,MPSA06 และ MOC3030 ทำงาน สัญญาณเรียกก็จะถูกส่งผ่านไปยังเครื่องโทรศัพท์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 ส่วนเชื่อมต่อสายโทรศัพท์ภายใน

Size	Document Number	REV
A	SUBSCRIBER LOOP INTERFACE CIRCUIT	
Date: March 13, 1990	Sheet 1 of 3	

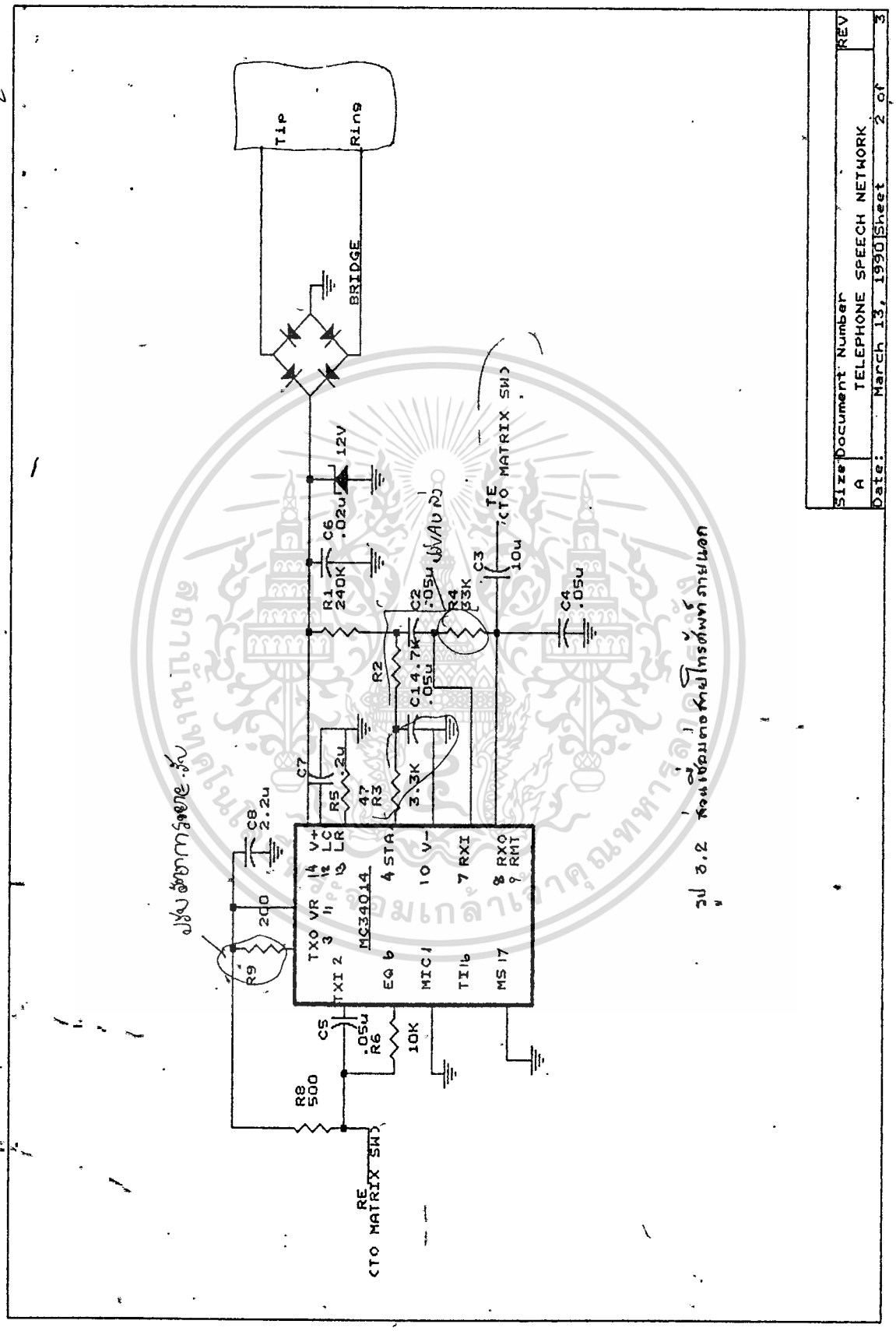
3.2 ส่วนเชื่อมต่อกับสายโทรศัพท์ภายนอก

หน้าที่ของส่วนเชื่อมต่อกับสายโทรศัพท์ภายนอก คือ จะทำการแยกสัญญาณที่ส่งมาจากคู่สายภายนอก ให้เป็นสัญญาณรับ-ส่ง เพื่อใช้ในการสวิทส์ติดต่อกับคู่สายที่ต้องการ และควบคุมการปรับอัตราขยายของสัญญาณรับ-ส่ง และสัญญาณข้างเคียง (Side Tone) ให้เหมาะสม

วงจรของส่วนเชื่อมต่อกับสายโทรศัพท์ภายนอก แสดงดังรูปที่ 3.2

จากวงจรรูปที่ 3.2 สัญญาณจากคู่สายภายนอกจะถูกเรียงทิศทางกระแสโดยบริดจ์เรคติไฟเออร์ (Bridge Rectifier) และมิรีเลย์ (Relay) เป็นตัวตัด-ต่อการยกหู ไอซี MC34014 (Telephone Speech Network) จะทำการแปลงสัญญาณจากคู่สายให้เป็นสัญญาณรับส่งที่แยกจากกัน และสามารถปรับอัตราขยายของสัญญาณรับ โดยการเปลี่ยนค่า R9 คือถ้าลดค่า R9 ลง อัตราขยายของสัญญาณรับก็จะมากขึ้น ส่วนอัตราขยายของสัญญาณส่งก็ปรับได้โดย การเปลี่ยนค่า R4 คือถ้าเพิ่มค่า R4 อัตราขยายของสัญญาณส่งก็จะมากขึ้นด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.2 วงจรอินเตอร์เฟสโทรศัพท์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

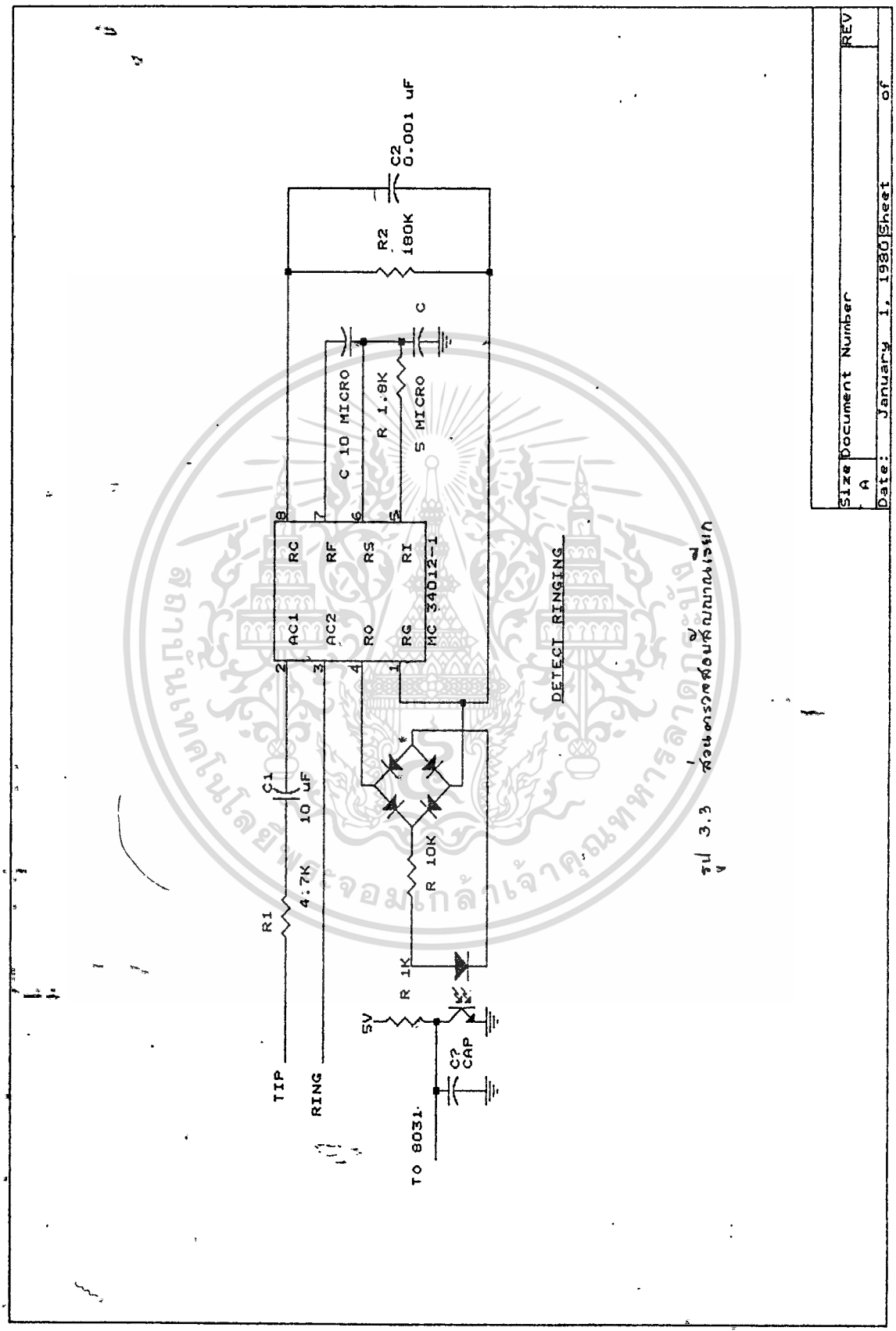
3.3 ส่วนตรวจสอบสัญญาณเรียก

หน้าที่ของส่วนนี้ คือ จะตรวจสอบสัญญาณเรียกจากคู่สายโทรศัพท์ภายนอก แล้วส่งเป็นสถานะลอจิกให้ไมโครโปรเซสเซอร์รู้ว่า มีการเรียกเข้า จากคู่สายภายนอกแล้ว

วงจรของส่วนตรวจสอบสัญญาณเรียก แสดงดังรูปที่ 3.3

จากวงจรรูปที่ 3.3 R1,C1 จะทำหน้าที่ควบคุม อินพุทอิมพีแดนซ์ (Input Impedance) ของคู่สายโทรศัพท์ในกรณีที่ความถี่ของสัญญาณเรียกต่างกัน และจะทำหน้าที่ป้องกันทรานส์เซี้ยน(Transients) R2 และ C2 จะทำหน้าที่เป็นตัวกำหนดความถี่ออสซิลเลต (Oscillator) ภายในไอซี และสัญญาณเอาท์พุท (Output) ที่ขา 4 ขณะมีสัญญาณเรียกเข้ามา จะทำให้ ออปโต้คัปเปอร์ 4N26 ทำงาน สถานะลอจิกที่ไมโครโปรเซสเซอร์อ่านได้ ขณะมีสัญญาณเรียก จะเป็นลอจิก "0" และ ขณะไม่มีสัญญาณเรียกจะเป็นลอจิก "1"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.3 ส่วนของวงจรตรวจสอบสัญญาณเรียกเข้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 ส่วนสวิทส์ชิ่ง

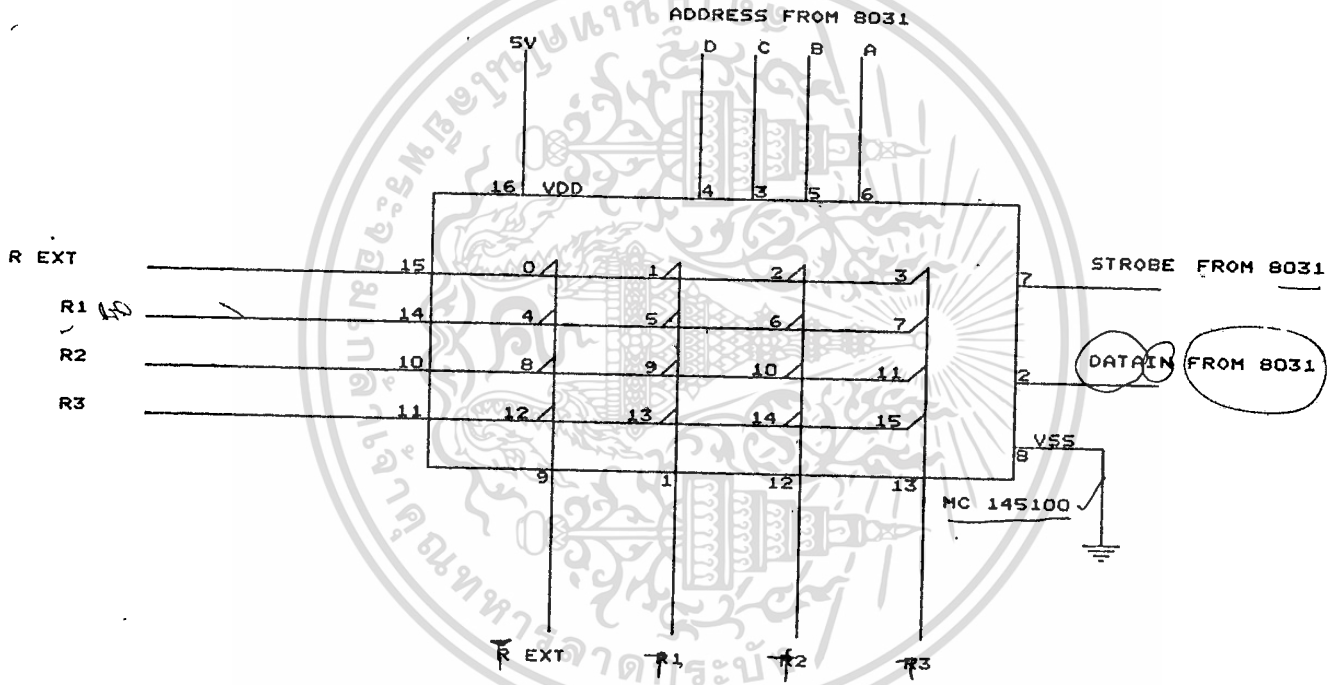
ส่วนนี้ทำหน้าที่เป็นทางผ่านของสัญญาณเสียง โดยทำการตัดต่อคู่สัญญาณรับ-ส่ง แทนการใช้รีเลย์ตัดต่อในระบบเก่า ซึ่งผลดี คือ สามารถควบคุมได้ง่ายจากไมโครโปรเซสเซอร์

วงจรส่วนสวิทส์ชิ่ง แสดงดังรูปที่ 3.4

จากรูปที่ 3.4 MC 145100 เป็นไอซี 4*4 ครอส พอยท์ สวิทส์ คือมีลักษณะเป็นเมตริกซ์ 4*4 ภายในประกอบด้วยสวิทส์ 16 ตัว สวิทส์แต่ละตัวมี ไดโอด โปรเทคชั่น (Diode Protection) และสวิทส์แต่ละตัวสามารถกำหนดให้ เปิด หรือ ปิด ได้โดยการใส่ตำแหน่ง 4 บิต (Address 4 bit) ของสวิทส์ (บิต D เป็นบิตที่มีนัยสำคัญสูงสุด (MSB) และบิต A เป็นบิตที่มีนัยสำคัญต่ำสุด (LSB))ที่ต้องการเปิด หรือ ปิด แล้วให้สัญญาณข้อมูล (Data In) เป็นลอจิก "1" เมื่อต้องการเปิดสวิทส์ และ เป็นลอจิก "0" เมื่อต้องการปิดสวิทส์ หลังจากนั้นก็ให้สัญญาณเก็บข้อมูล (Strobe) เป็นลอจิก "1" เพื่อเก็บข้อมูลที่โปรแกรมไว้ สวิทส์ก็จะทำงานตามการโปรแกรม

แผนภูมิเวลา (Timing Diagram) ของการโปรแกรมสวิทส์ ของ MC145100 เป็นดังรูปที่ 3.4.2 ในภาคผนวก ง.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



MAIRIX SWITCH

รูป 3.4 ส่วนสวิทช์ที่

Size A	Document Number	REV
Date: March 13, 1990	Sheet of	

3.5 ส่วนสร้างสัญญาณต่างๆของโทรศัพท์

รายละเอียด และคุณลักษณะของสัญญาณไดอัล สัญญาณไม่ว่าง สัญญาณเรียกกลับ และ สัญญาณเรียก ได้กล่าวไว้แล้ว ในส่วนนี้จะกล่าวถึง วิธีสร้างสัญญาณต่างๆเหล่านี้คือ

3.5.1 สัญญาณไดอัล

สัญญาณไดอัลสร้างโดยการใช้ไอซี ไทมเมอร์ (Timer 555) สร้างเป็นออสซิลเลเตอร์ กำเนิดสัญญาณที่มีความถี่ประมาณ 400 เฮิร์ท โดยความถี่ของสัญญาณคำนวนจาก

$$f = 1.44/(R1+2R2)C$$

$$= 1.44/((1K+(2*3.3K))*0.47 \text{ micro}$$

$$= 403 \text{ Hz}$$

สัญญาณไดอัลนี้ จะถูกส่งเข้าสู่ ขาสัญญาณรับ Rext-R3 ของโทรศัพท์แต่ละเครื่องโดยการเปิดสวิทส์ B0-B3 ซึ่งถูกควบคุมการเปิด-ปิด โดยไมโครโปรเซสเซอร์

3.5.2 สัญญาณไม่ว่าง

สัญญาณไม่ว่าง สร้างโดย การนำสัญญาณไดอัล มาควบคุมการเปิด-ปิดสลับกันทุกๆ 0.5 วินาที การควบคุมนี้เกิดจาก การตั้งเวลาไทมเมอร์ของไมโครโปรเซสเซอร์ให้ส่งสัญญาณเพื่อใช้ในการเปิด-ปิดสวิทส์ทุกๆ 0.5 วินาที แล้วไมโครโปรเซสเซอร์ก็จะควบคุมสวิทส์ B0-B3 ให้เปิด-ปิดสลับกัน เพื่อให้เกิดเป็นสัญญาณไม่ว่าง

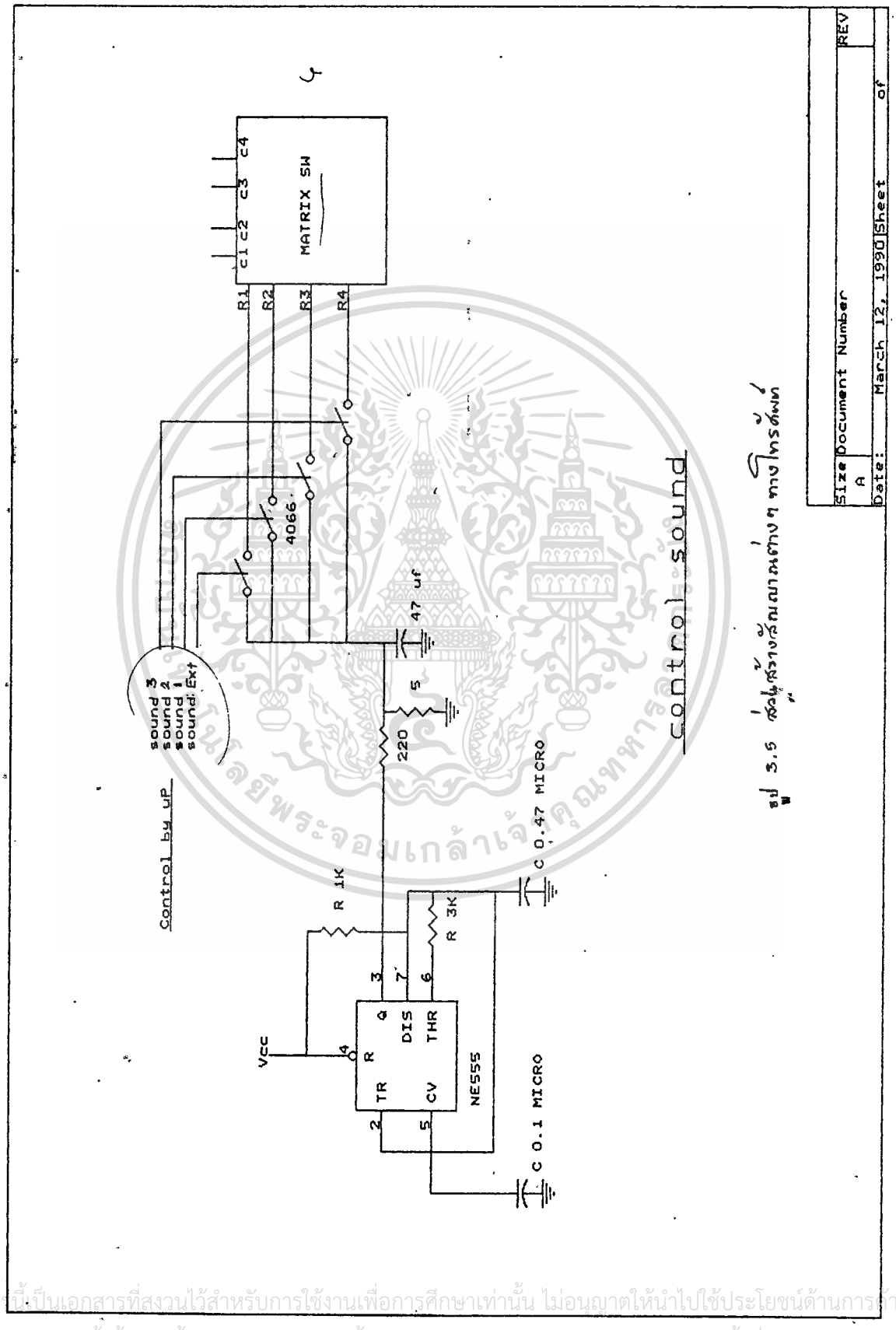
3.5.3 สัญญาณเรียกกลับ

สัญญาณเรียกกลับ มีหลักและวิธีการสร้าง คล้ายกับสัญญาณไม่ว่าง แต่ช่วงเวลาการเปิด-ปิดสวิทส์ จะต่างกันคือ จะเปิดสวิทส์ให้สัญญาณผ่าน 1 วินาที และจะปิดสวิทส์ไม่ให้สัญญาณผ่าน 3 วินาที สลับกันไปเรื่อยๆ ก็จะได้สัญญาณเรียกกลับที่ต้องการ

3.5.4 สัญญาณเรียก

วิธีสร้าง สัญญาณเรียก จะอาศัยการตั้งเวลาของไทมเมอร์ในการสร้างสัญญาณเรียกกลับ แล้วนำไปควบคุมเปิดสัญญาณเรียก (Ring Enable) ให้สัญญาณ Ringing Generator (เป็นสัญญาณ เอ ซี 48 โวลต์ 50 เฮิร์ท) ดัง 1 วินาที ดับ 3 วินาที ทำให้ได้สัญญาณเรียกตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.5 ส่วนสร้างสัญญาณเสียง ๆ ทางไกรงั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 ส่วนตรวจสอบการกดหมายเลข

ส่วนนี้ทำหน้าที่ในการตรวจสอบ หมายเลขจากการกด แล้วแสดงค่าของตัวเลขเลขปั่นเลขแบบ บี ซี ดี (BCD) และตรวจสอบคาบเวลาที่ถูกต้องของสัญญาณที่เกิดจากการกด ถ้าการกดหมายเลขทำให้เกิดสัญญาณที่มีคาบเวลาไม่ถูกต้องวงจรถอดรหัสภายในก็จะไม่ทำงาน ทำให้ไม่สามารถตรวจสอบการกดหมายเลขได้ ส่วนนี้ทำหน้าที่โดยใช้ ไอซี ดี ที เอ็ม เอฟ ดีโค้ดเดอร์ (DTMF Decoder) เบอร์ MT 8870 ซึ่งภายในวนนี้ทำ

หน้าที่โดยใช้ ไอซี ดี ที เอ็ม เอฟ ดีโค้ดเดอร์ (DTMF Decoder) เบอร์ MT 8870 ซึ่งภายในประกอบด้วยส่วนสำคัญ 5 ส่วนคือ

1 ภาคกรองความถี่ (Filter section)

2 ภาคถอดรหัส (Decoder Section)

3 ภาคตรวจสอบสัญญาณ (Steering Circuit)

4 ภาคขยายสัญญาณความแตกต่าง (Differential Input)

5 ภาคกำเนิดความถี่ (Oscillator)

โดยแต่ละส่วนจะทำหน้าที่สัมพันธ์กัน ในการตรวจสอบสัญญาณ วงจรของส่วนตรวจสอบการกดหมายเลข แสดงดังรูปที่ 3.6

จากรูปที่ 3.6 อนาลอกสวิทส์ (4066) จะถูกควบคุมจากไมโครโปรเซสเซอร์ให้ทำการเปิดสวิทส์ตัวใดตัวหนึ่ง เพื่อนำสัญญาณจากการกดในขาส่งสัญญาณ Texต, T1,T2 หรือ T3 มาทำการถอดรหัส และตรวจสอบหมายเลข โดย MT 8870 และ MT 8870 มีอัตราขยายสัญญาณความแตกต่าง เท่ากับ 0.1

อัตราขยายสัญญาณความแตกต่าง = R2 / R1

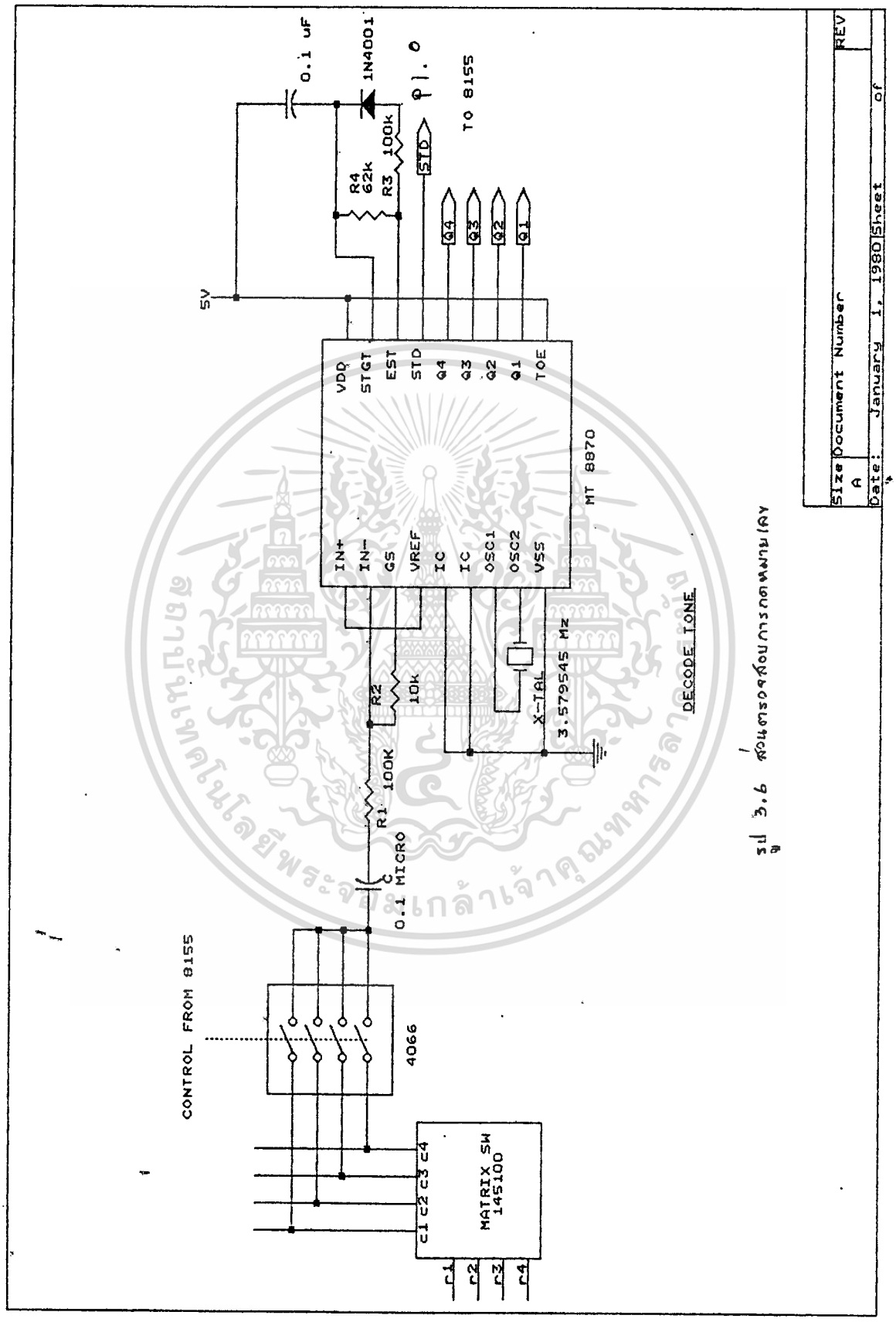
= 10K / 100K

= 0.1

สาเหตุที่ต้องลดทอนสัญญาณ เพื่อลดความไวในการรับสัญญาณจากการกด เนื่องจากการกดหมายเลขของโทรศัพท์เครื่องที่ถูกปิดสวิทส์ (ไม่ให้สัญญาณผ่าน) นั้น สามารถผ่านเข้ามาได้ แต่มีขนาด(Amplitude) เล็กลง จึงต้องทำการลดทอนสัญญาณ เพื่อให้สัญญาณที่ได้มีขนาดไม่พอเพียงในการนำไปตรวจสอบและถอดรหัส ส่วนสัญญาณที่ผ่านสวิทส์ที่ถูกเปิด(ให้สัญญาณผ่าน)จะมีขนาดพอที่จะนำไปถอดรหัสได้

R3,R4 และ D1 เป็นตัวกำหนดช่วงเวลาที่น้อยที่สุดของการปรากฏของสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.6 วงจรอุปกรณ์ตรวจสอบการกดแป้นพิมพ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่สามารถนำไปตรวจสอบ และถอดรหัสได้

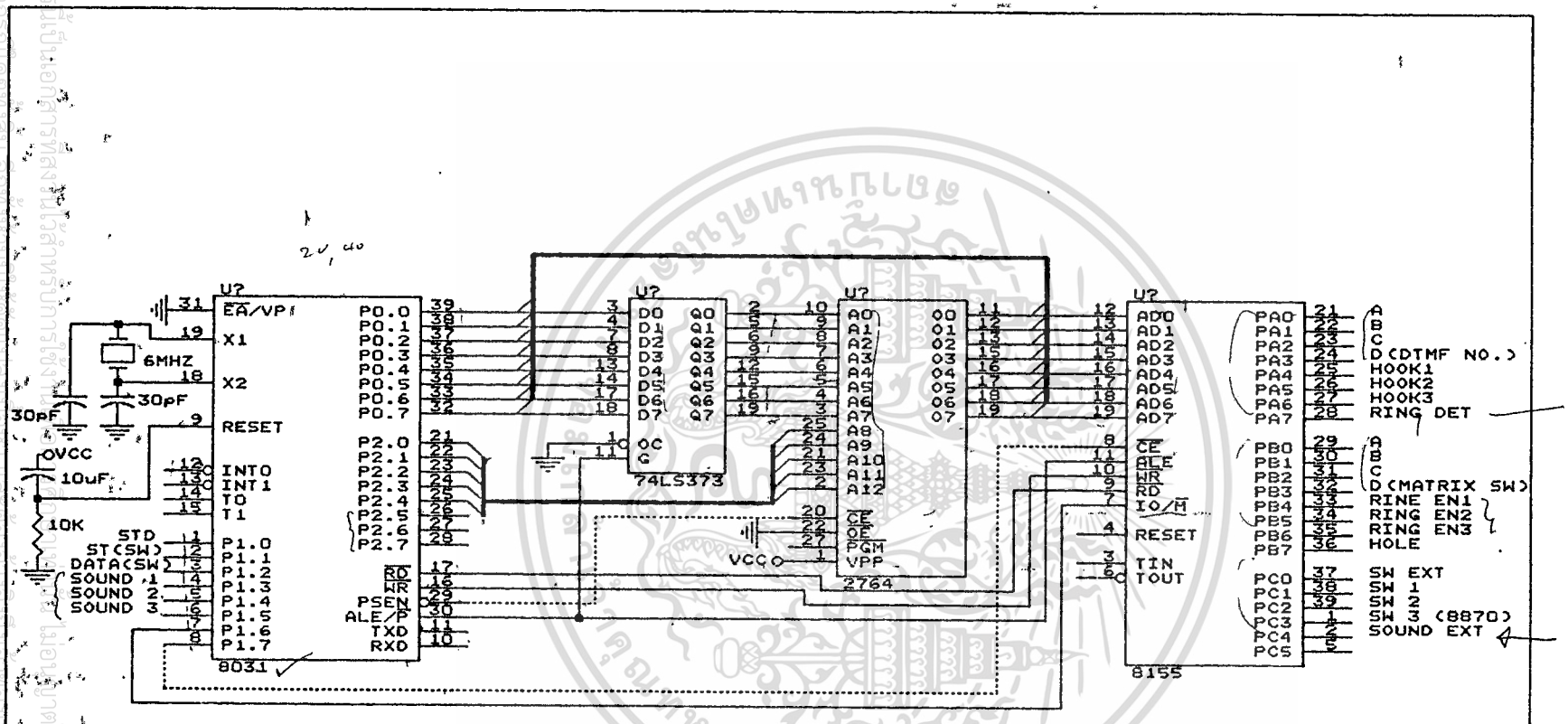
3.7 ส่วนควบคุมการทำงานของระบบ

วงจรส่วนนี้เป็นหัวใจในการทำงานของ PABX ซึ่งจะกำหนดฟังก์ชันการทำงานทั้งหมด เช่น ลักษณะการต่อสายนอก สายใน และ ตัดต่อสายต่างๆ และ เช็คการทำงานของส่วนต่างๆ ในที่นี้จะใช้ไมโครโปรเซสเซอร์ เบอร์ 8031 ซึ่งเป็น Single Chip ที่มี Timer/Counter , Port แบบขนาน และ Serial Port และใช้โปรแกรม Monitor กำหนดการทำงานภายใน เบอร์ 2764 ซึ่งเป็น EPROM ขนาด 8 KB ในการต่อกับหน่วยความจำภายนอกนี้จะทำให้ 8031 เหลือ Port 1 เพียง Port เดียว จึงได้ทำการเพิ่ม 8155 ซึ่งจะทำให้เพิ่ม Port แบบขนานได้อีก 3 Port และภายใน 8155 ยังมี RAM ให้อีก 256 Byte การนำ 8155 มาใช้งานกับ 8031 นี้สะดวกมาก เพราะ Intel ออกแบบมาให้ใช้กับ 8085 ซึ่งมีระบบ Bus เป็นแบบ Multiplex เช่นเดียวกับ 8031 โดยจากวงจรมีการใช้ขา Port 1 บิท 7 ของ 8031 ต่อไปที่ขา IO/M ของ 8155 เพื่อทำการเลือกว่าจะทำการติดต่อกับ Memory หรือ Port ขนาน และทำการต่อขา WR และ RD ไปที่ 8155 โดยตรงโดย CS จะทำการเลือกเมื่อมีการอ่าน หรือ เขียนกับ 8155

โครงสร้างของ Software

การเขียนโปรแกรมควบคุมระบบนั้นจะใช้ Timer ตัวที่ 1 มาทำเป็นนาฬิกาของระบบ เพื่อใช้ควบคุมการทำงานต่างๆ โดยจะทำงานประสานกับส่วนต่างๆ เช่น เขียนและอ่านข้อมูลจากส่วน DTMF Decoder , Analog Switch เช็คสัญญาณการยกหู และ กำหนดการ Enable สัญญาณ Ringing , อ่านค่า Ring Detector , และกำหนดให้มีการดังของ Busy , Ring, Ring Backในเครื่องโทรศัพท์ที่กำหนดโดย การใช้โปรแกรมกำหนดลักษณะรูปแบบสัญญาณ ส่วนทางด้านการเขียนโปรแกรม ทางกลุ่มได้พัฒนา Utilityเพื่อใช้ในการพัฒนาโปรแกรม คือ 8031 Emulator in Circuit และ Compiler ภาษา BASICเพื่อแปลโปรแกรมที่เขียนขึ้นเป็นภาษา Assembly ของ 8031ซึ่งจะสามารถพัฒนาโปรแกรมต่อไปได้รวดเร็วขึ้น โดยรูปแบบของภาษาอยู่ในภาคผนวก ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 3.7 ส่วนควบคุมการทำงานของระบบ

บทที่ 4

การทดลอง และ ผลการทดลอง

การทดสอบเครื่องชุมสายโทรศัพท์สาขาอัตโนมัติ

ได้ทำการทดสอบการทำงาน และ ผลของการทำงานของเครื่องดังต่อไปนี้

1. ทดสอบการทำงานของเครื่องชุมสายโทรศัพท์สาขาอัตโนมัติที่สร้างขึ้น โดยการทดสอบ การโทรศัพท์ออก และ การโทรศัพท์เข้า กับเครื่องชุมสายโทรศัพท์สาขาอัตโนมัติของ HOMEX รุ่น 205 ปรากฏว่าสามารถทำงานได้ผลเป็นที่น่าพอใจ

2. ทดสอบความเพี้ยนของสัญญาณเสียง โดยการฟังเสียงของผู้พูดจากการคุยทางโทรศัพท์ที่ผ่านเครื่องชุมสายโทรศัพท์ขององค์การโทรศัพท์ แต่ไม่ผ่านเครื่องชุมสายโทรศัพท์สาขาอัตโนมัติที่สร้างขึ้น แล้วเปรียบเทียบกับการคุยทางโทรศัพท์ที่ผ่านทั้งเครื่องชุมสายโทรศัพท์ขององค์การโทรศัพท์ และ เครื่องชุมสายโทรศัพท์สาขาอัตโนมัติที่สร้างขึ้น ปรากฏผลว่า เสียงนั้นใกล้เคียงกันมาก

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุป และวิจารณ์

เครื่องชุมสายโทรศัพท์สาขาอัตโนมัติที่สร้างขึ้นนี้ ประสบความสำเร็จ คือ สามารถทำงานได้ ในขั้นพื้นฐานตามที่ออกแบบไว้ แต่ก็ยังสามารถพัฒนาประสิทธิภาพ และฟังก์ชั่นการทำงานต่อไปได้อีก โดยส่วนใหญ่การพัฒนาฟังก์ชั่นความสามารถ จะขึ้นอยู่กับ ซอฟท์แวร์ที่ควบคุมการทำงาน

แนวทางการพัฒนาเครื่องชุมสายโทรศัพท์สาขาอัตโนมัติ

1 เพิ่มซอฟท์แวร์ที่ใช้ในการนับจำนวนพัลซ์ เพื่อทำให้เครื่องชุมสายโทรศัพท์สาขาอัตโนมัตินี้ สามารถใช้กับคู่สายระบบพัลซ์ หรือระบบหมุนได้

2 เพิ่มซอฟท์แวร์ในการสร้างเสียงดนตรี เพื่อให้เกิดเสียงดนตรีในขณะพักสายโทรศัพท์

3 เพิ่มฟังก์ชั่นการทำงานในการโอนสายโทรศัพท์ ให้มีหลายโหมด และสามารถเลือกใช้ได้ง่ายจากภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาค ผนวก ก.
วิธีใช้งานเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีใช้งาน

1)การติดต่อภายใน

-ยกหูโทรศัพท์ขึ้น จะได้ยินสัญญาณ dial tone กดหมายเลขที่ต้องการติดต่อ (เลข 1-3)

-ถ้าปลายทางไม่ว่างจะได้ยินสัญญาณไม่ว่าง (busy tone) ดัง 0.5 วินาที และ ดับ 0.5 วินาที ให้วางและโทรใหม่

-ถ้าปลายทางว่างจะได้ยินสัญญาณเรียกกลับ (ringback tone) ดัง 1 วินาที และ ดับ 3 วินาที และเครื่องปลายทางจะมีกระดิ่งดัง

-เมื่อปลายทางรับสายแล้ว จะสามารถพูดคุยกันได้

2)การติดต่อสายนอก

-กดหมายเลข 0 จะได้ยินสัญญาณแมวกรน (dial tone)จากองค์การ ฯ กดหมายเลขที่ต้องการติดต่อโทรออก

-ถ้าสายนอกไม่ว่างจะได้ยินสัญญาณไม่ว่าง (busy tone) ดัง 0.5 วินาที และ ดับ 0.5 วินาที ให้วางและโทรใหม่

3)การโอนสาย

-ขณะที่คุยกับสายใดๆ กดแคร่ 1 ครั้ง (กดไม่เกิน 1 วินาที)และ กดหมายเลขที่ ต้องการต้องการจะโอนไป เมื่อผู้รับยกหูและคุยกันแล้ว ให้วางหูจะเป็นการโอนสายทันที

4)การโทรเข้า

เมื่อสายนอกโทรเข้ามาจะทำงานตามโหมดที่ตั้งไว้คือ

โหมด 1

จะมีกระดิ่งดังทุกเครื่อง เมื่อมีผู้ใดรับสาย จะต่อสายนอกกับสายนั้นทันที โดยผู้อื่นจะแอบฟังไม่ได้

โหมด 2

จะรอให้สายนอกกดหมายเลขที่ต้องการติดต่อ (1-3) (ภายใน 5 วินาที) และจะมีกระดิ่งดังที่เครื่องนั้น ถ้าสายนอกไม่กดภายใน 5 วินาที จะไปดังที่เครื่องหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) <u>การตั้งโหมดการโทรเข้า</u>

-ยกหูโทรศัพท์ขึ้นจากเครื่องใดๆ กดปุ่ม '*' เมื่อต้องการตั้งโหมด 1

-ยกหูโทรศัพท์ขึ้นจากเครื่องใดๆ กดปุ่ม '#' เมื่อต้องการตั้งโหมด 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาค ผนวก ข.

รายละเอียดของ Compiler (WATT51.EXE)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในภาคผนวกนี้ จะอธิบายการใช้งาน COMPILER ซึ่งจะใช้แปลโปรแกรมภาษาระดับสูง ซึ่งมีรูปแบบคล้ายๆ ภาษา QUICK BASIC ของ MICROSOFT co,ltd. ซึ่งเป็นภาษาที่มีลักษณะเป็นโครงสร้าง เช่นเดียวกับ PASCAL, C. ผลที่ได้จากการแปลจะเป็น ภาษา ASSEMBLY ของ CPU single chip ตระกูล MCS-51. ถ้าพัฒนาโปรแกรมด้วยภาษา ASSEMBLY จะพบความยุ่งยากมาก จึงได้ใช้โปรแกรม compiler ที่เขียนขึ้นแปลภาษาสูงไปเป็นภาษา ASSEMBLY แล้วจึงใช้ ASSEMBLER ของ Intel แปลเป็น execute code ซึ่งผลที่ได้สามารถทำงานได้ผลดีมาก และ ในโปรแกรมยังได้เพิ่มส่วน Debug Information เพื่อใช้กับ การ์ด MCS-51 Emulator in circuit ที่เคยสร้างขึ้น ทำให้สามารถ Debug โปรแกรมได้ในระดับ Source Code

รูปแบบของภาษา

-เป็นภาษาโครงสร้างคล้าย PASCAL , C เช่น

```
if  test1 = test2
  if test2<test3
    test2=test3+1
  else
    test3=test2+1
  endif
endif
```

-สามารถแบ่งเป็น module subroutine & function ทั้ง pass by value และ pass by referense เช่น

```
sub  test( a,b,c)
  d=a+b
  d=c/2
endsub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-สามารถทำ inline assembly คือ การแทรกคำสั่งภาษา assembly ลงใน source program
-มีคำสั่งคำนวณ +,-,*,/ (เฉพาะ integer)
-มีตัวแปร pointer สามารถชี้ไปหน่วยความจำใด ๆได้

ASSIGNMENT statement

<statement> -> <term> = <expresssion>

โดยที่

<expression> -> <term> <operator> <term>

<operator> -> + | - | * | /

<term> -> <variable> | ^<variable> | <constant>

ตัวอย่างคำสั่งเช่น

test1 = test2

test1 = test2+test3

test1 = ^test2 +test3

โดยคำสั่งสุดท้าย หมายถึง นำค่า test2 ไปเป็น address เพื่อนำค่าใน memory ไปบวกกับ test3

โดย assignment statement นี้จะคำนวณได้ทีละคู่

SUBROUTINE statement

<statement> -> <variable> = <func> (<paras>)

<statement> -> CALL <sub> (<paras>)

<paras> ::= <para>,<paras> | <para>

<para> ::= | <number> | <varisble>

เช่นคำสั่ง

```
max =  findmax (no1,no2)
call outport(max)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

JUMP statement

<statement> -> LJMP <label>

<label> ::= <letter>|<letter><label>|<letter><digits>

เช่นคำสั่ง

```
LJMP loop1    ; long address jump to  label loop1:
```

IF statement

```
<statement> -> IF <term><relation operator><term>
                  <statement>
               ENDIF
<statement> -> IF <term><relation operator><term>
                  <statement>
               ELSE
                  <statement>
               ENDIF
```

ตัวอย่างเช่น

```
if  test1 = test2
  if test2<test3
    test2=test3+1
  else
    test3=test2+1
  endif
endif
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FOR statement

```
<statement> -> FOR <varible>=<term> TO <term>
                  <statement>
               NEXT
```

ตัวอย่างเช่น

```
; find sum from 1..10
j=0
for i=1 to 10
 j=j+i
next
```

REMARK statement

```
<statement> -> <remark>
<remark> ::= <statement> ;<charactor> | ;<charactor>
```

ตัวอย่างเช่น

```
;increment
i=i+1   ;increment i
```

RETURN statement

```
<statement> ->ENDSUB | ENDFUNC | ENDINT
```

BIT statement

```
<statement> -> SETB <bit variable>
<statement> -> CLR  <bit variable>
<statement> -> IF_b <bit variable>
                  <statement>
               ENDIF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
<statement> -> IF_b <bit variable>
                  <statement>
               ELSE
                  <statement>
               ENDIF
<statement> -> IF_nb <bit variable>
                  <statement>
               ENDIF
<statement> -> IF_nb <bit variable>
                  <statement>
               ELSE
                  <statement>
               ENDIF
```

INLINE ASSEMBLY statement

```
<statement> -> '<assembly code>
<statement> -> #asm
                  <assembly code>
               #easm
```

การใช้งานแต่ละคำสั่ง และ การสร้าง code

ASSIGNMENT เช่น

1. x = y
2. x = y + z
3. x = y / z
4. x = ^y + z
5. x = y + ^z
6. ^x= y + z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างการสร้าง code คำสั่งที่ 4 คือ

```
    x = ^y + z
->  mov r0,y    ; load pointer y
    mov a,@r0   ; load intend in memory
    add a,z     ; add to z
    mov x,a     ; move to x
```

IF statement

คำสั่ง

```
if x=y
 <statement 1>
else
 <statement 2>
endif
```

จะสร้าง code เป็น

```
        mov a,x       ; load accumulator with x
        clr c         ; clr carry flag
        subb a,y      ; compare a,y
        jz true       ; a-y =Ø  goto true
        ljmp else     ; else jump else:
true:   ..
        <statement 1>
         ..
        ljmp end

else:   ..
        <statement 2>
        ..
end:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BIT statement

```
if_b  watt
  <statement 1>
else
  <statement 2>
ensif
```

คำสั่งนี้จะทำการเช็ค ตัวแปรบิท watt ว่าถูก set หรือไม่ ถ้าใช่จะทำ statement1 ถ้าไม่ใช่จะทำ statement2

```
if_nb  watt
  <statement 1>
else
  <statement 2>
ensif
```

คำสั่งนี้จะทำการเช็ค ตัวแปรบิท watt ว่าถูก clear หรือไม่ ถ้าใช่จะทำ statement1 ถ้าไม่ใช่จะทำ statement2

FOR statement

```
for i= start to stop
 < statement1>
next
```

จะเป็นการวน loop โดยเริ่มที่ i=start ทำ statement1 และเพิ่ม i ขึ้นหนึ่งและทำ statement1 และเพิ่ม i ขึ้นหนึ่ง วนไปจนกระทั่ง i>stop

SUBROUTINE statement

```
CALL sub1(x,y,z)      หรือ    CALL  sub1()
```

โดยค่าในวงเล็บคือ parameter ที่ส่งไป subroutine
ซึ่งจะสร้าง code ได้เป็น

```
call sub1(x,y,z)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
->      push x
        push y
        push z
        lcall sub1
```

หรือ ถ้าเป็น function

```
 x = testfunc (1,watt)
->      mov a,1
        push ACC            ;push 1 to stack
        push watt
        lcall testfunc
        mov x,testfunc_     ;get return value in function_
```

DECLARE SUBROUTINE statement

ถ้าเป็น sub จะ declare & สร้าง code ได้เป็น

```
SUB  watt (i,j,k)        watt:  pop  sub_pch
  ..              ---->         pop  sub_pcl
  ..                            pop  k
  ..                            pop  j
  ..                            pop  i
  ..                            push sub_pcl
  ..                            push sub_pch
  ..                            ..
  ..                            ..
ENDSUB                          RET
```

แต่ถ้าเป็น function จะ declare ได้เป็น

```
func  watt(i,j,k)
  ..
endfunc
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INLINE ASSEMBLY statement

ใช้ ' นำหน้าหรือ ใช้ #asm และ #easm คล่อม เช่น

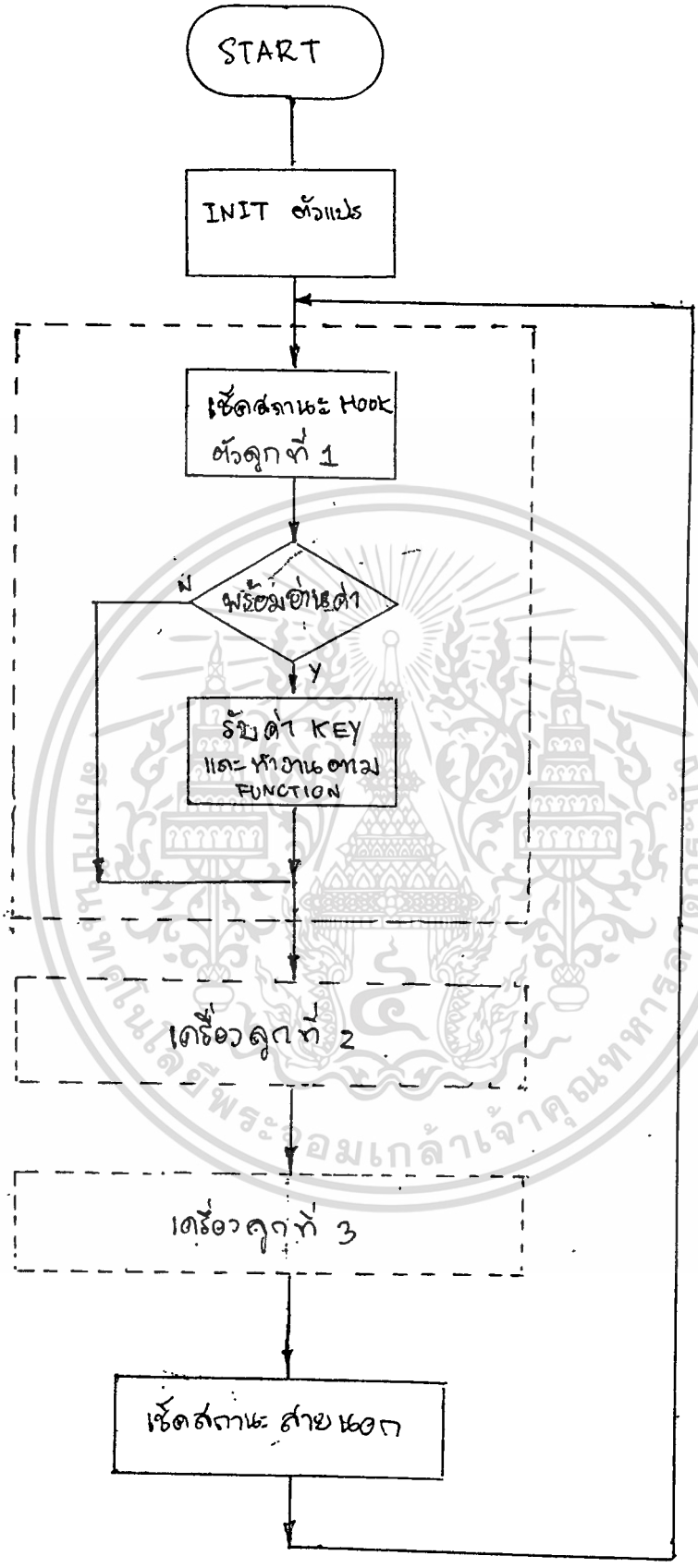
```
'mov a,1
#asm
    mov a,1
    add a,2
#easm
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค.

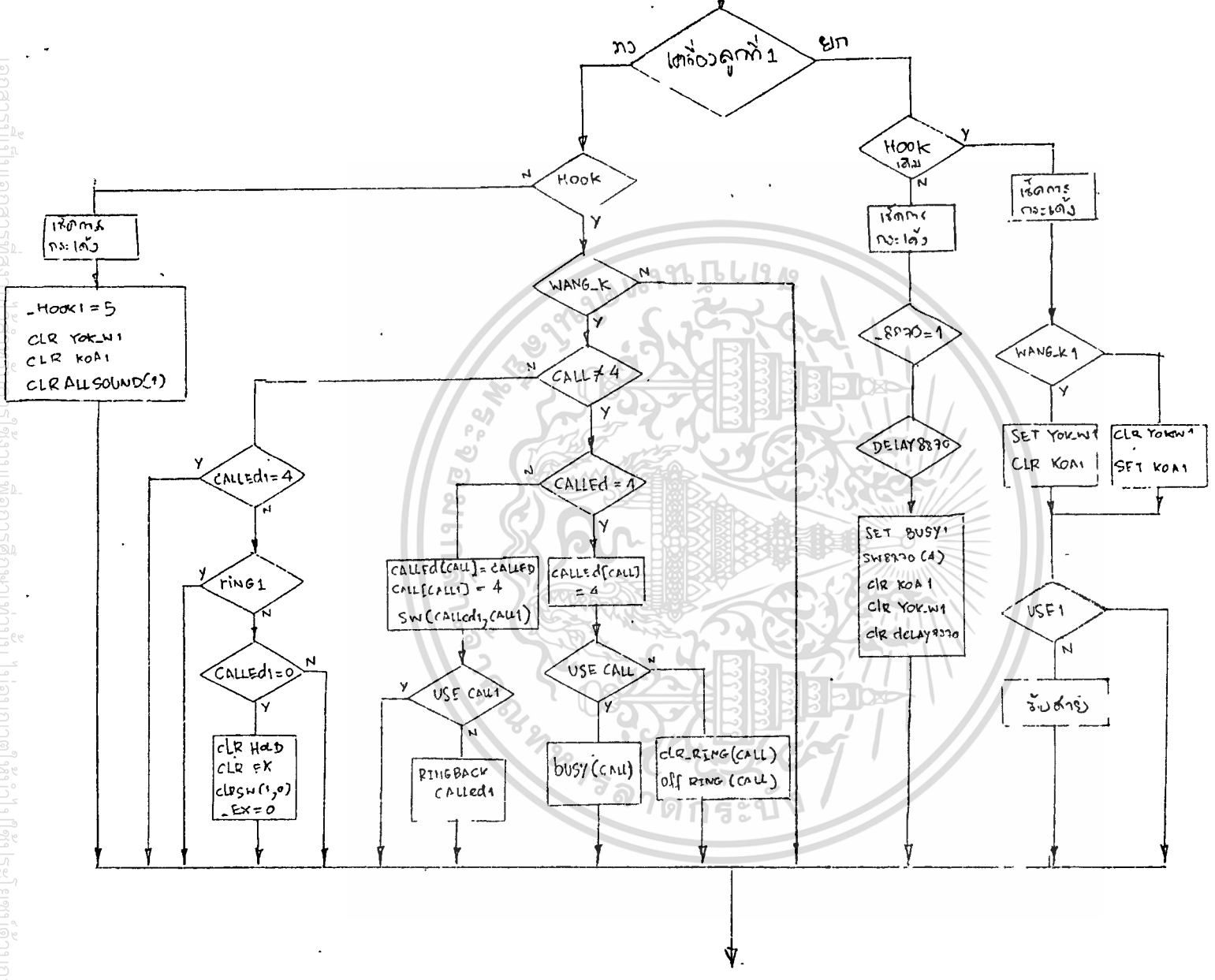
LISTING PROGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

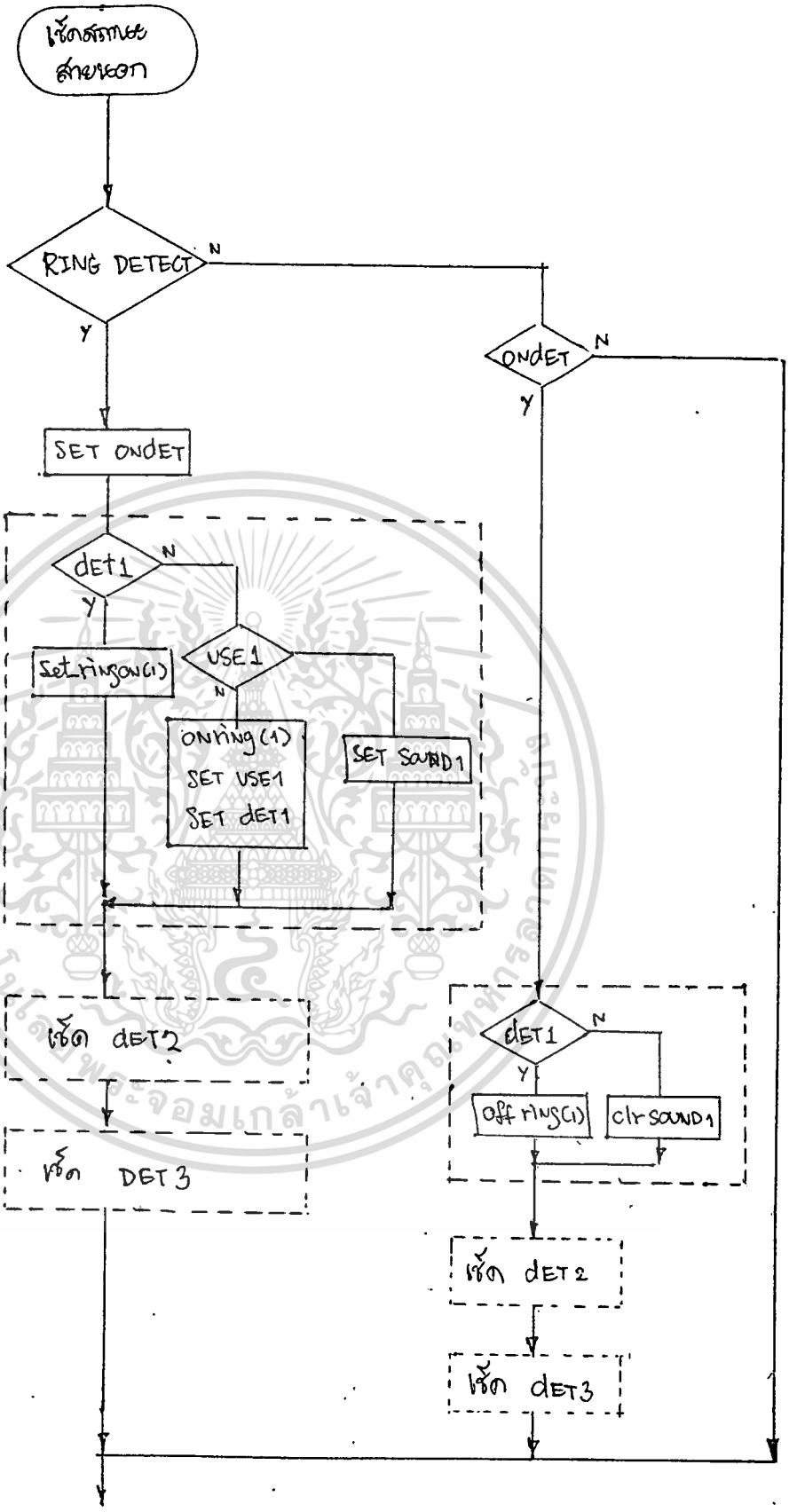


รูปที่ 1.4 FLOW CHART โปรแกรมการทำงานหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

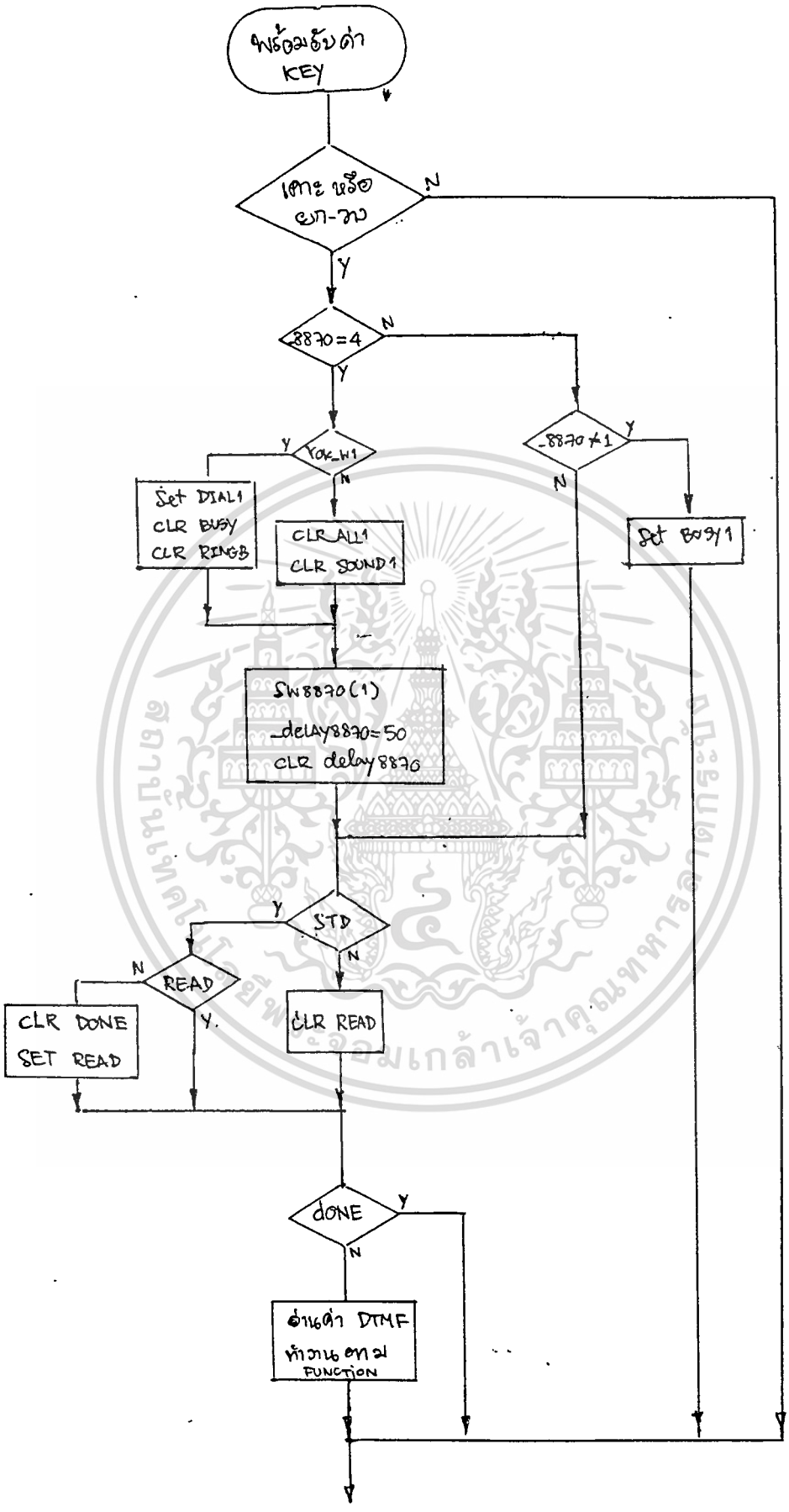


รูปที่ 2.ค FLOW CHART โปรแกรมเช็คสถานะเครื่องลูกที่ 1



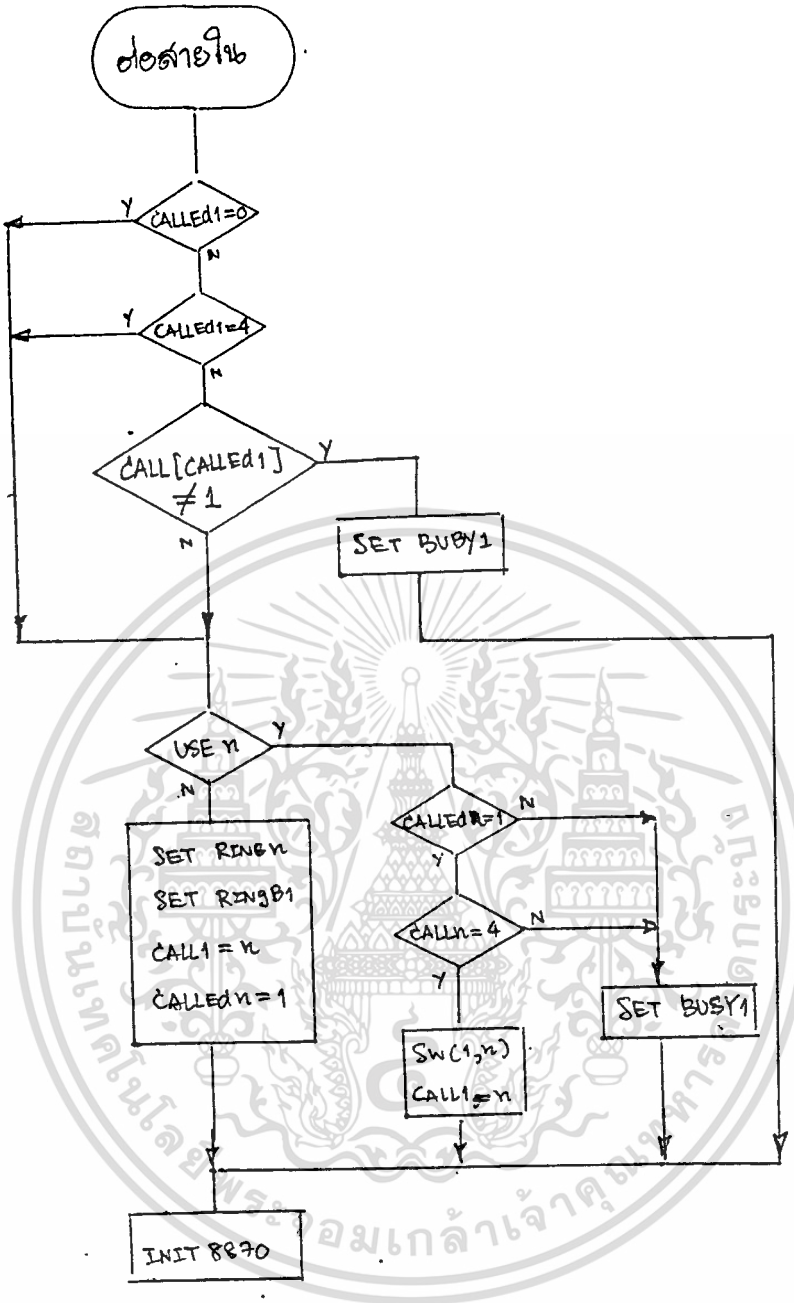
รูปที่ 3ค FLOW CHART โปรแกรม เช็คสถานะสายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



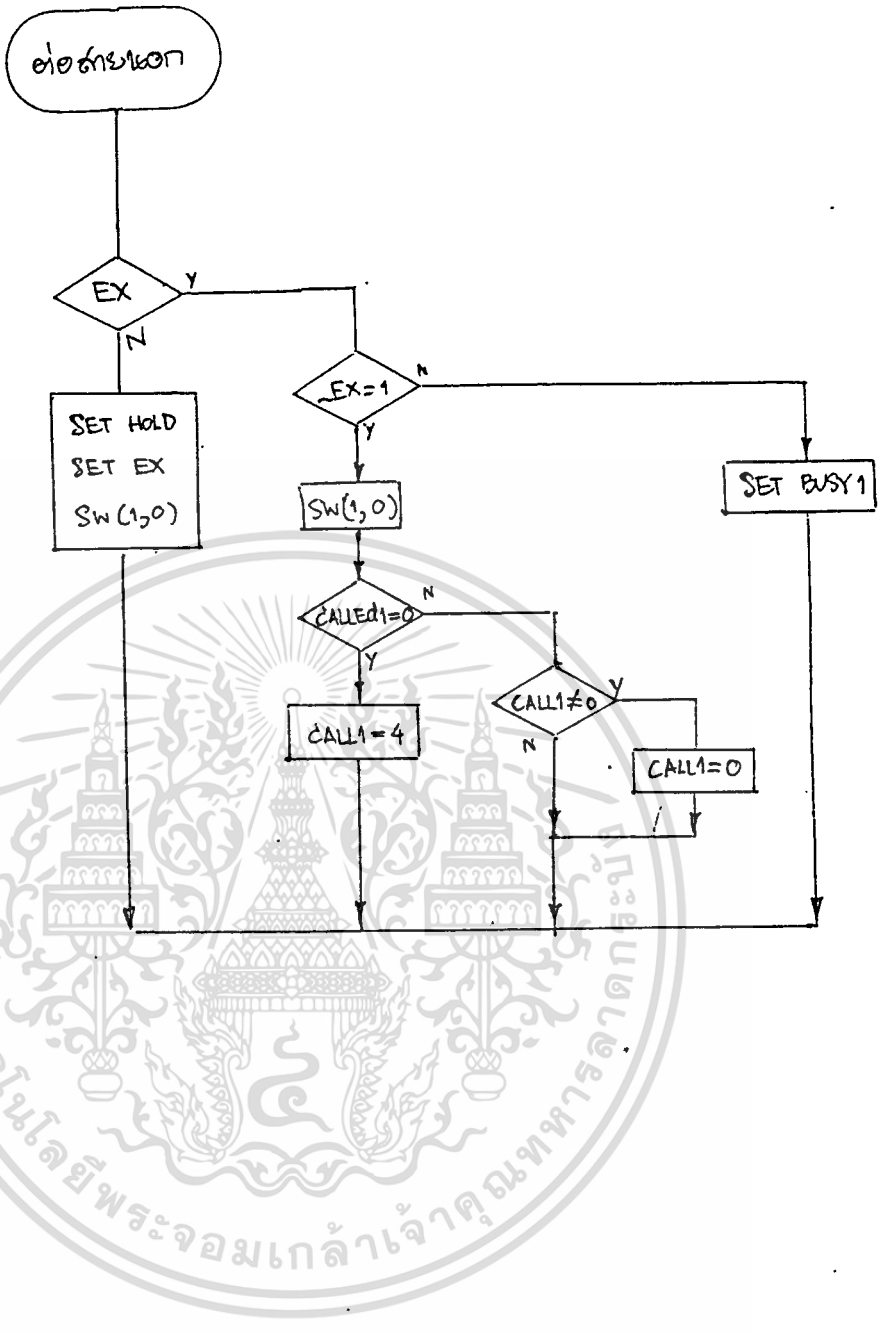
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.9 FLOW CHART โปรแกรมรับค่าคีย์



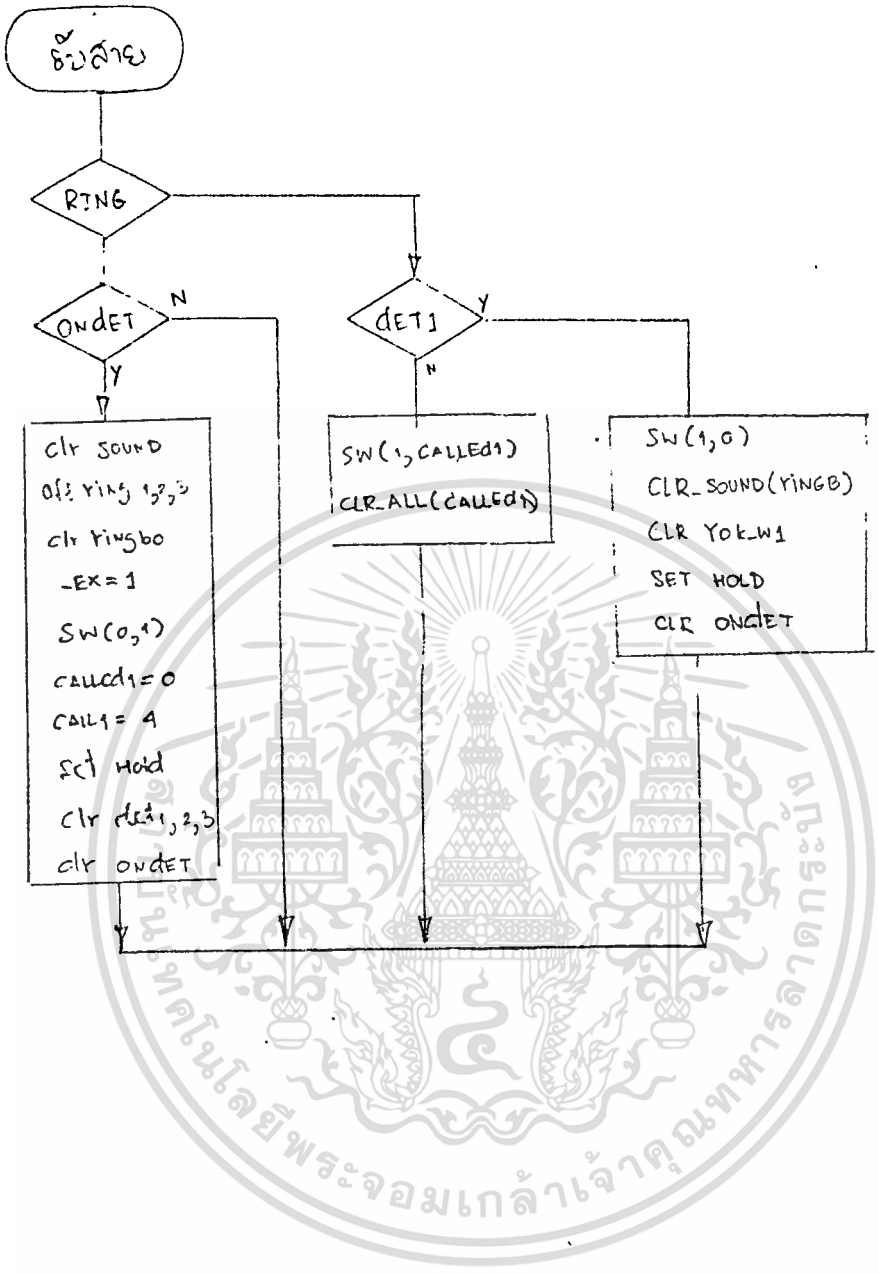
รูปที่ 5ค FLOW CHART โปรแกรม การต่อสายใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



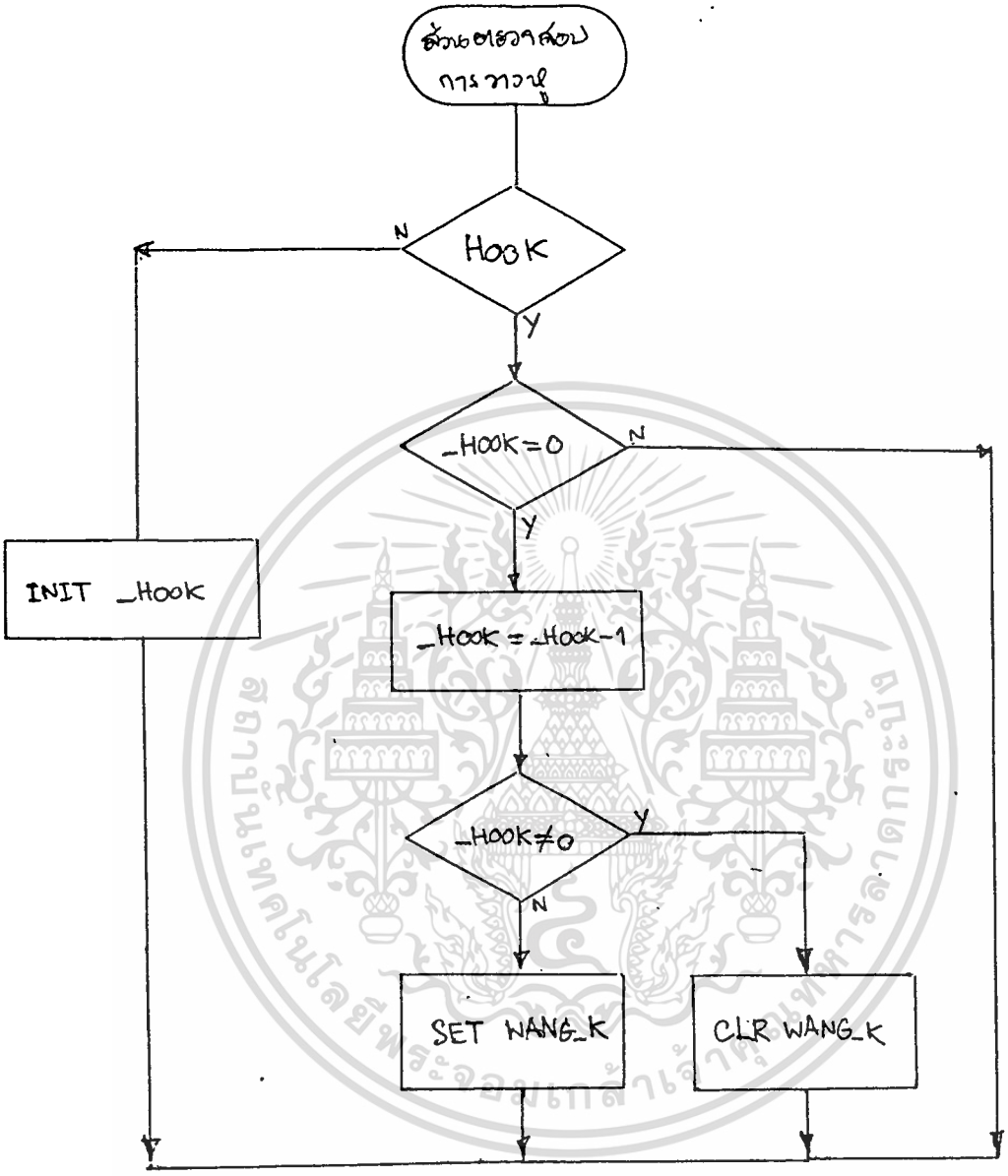
รูปที่ 6 ค. FLOW CHART โปรแกรมการต่อสายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



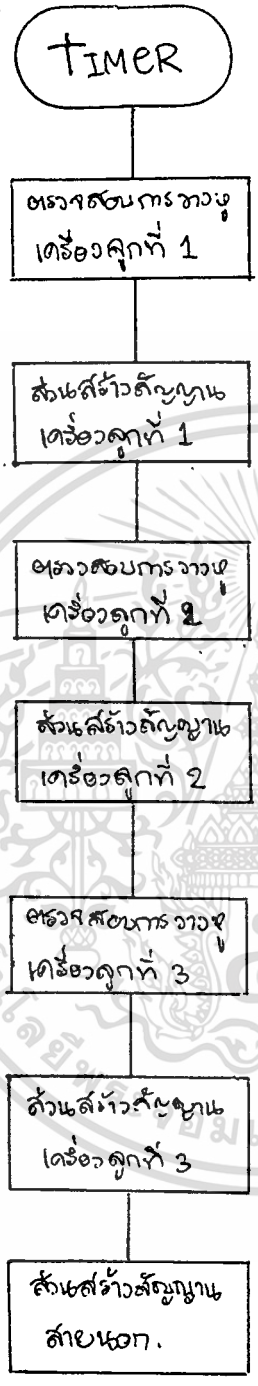
รูปที่ 7ค FLOW CHART โปรแกรม การรับสาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



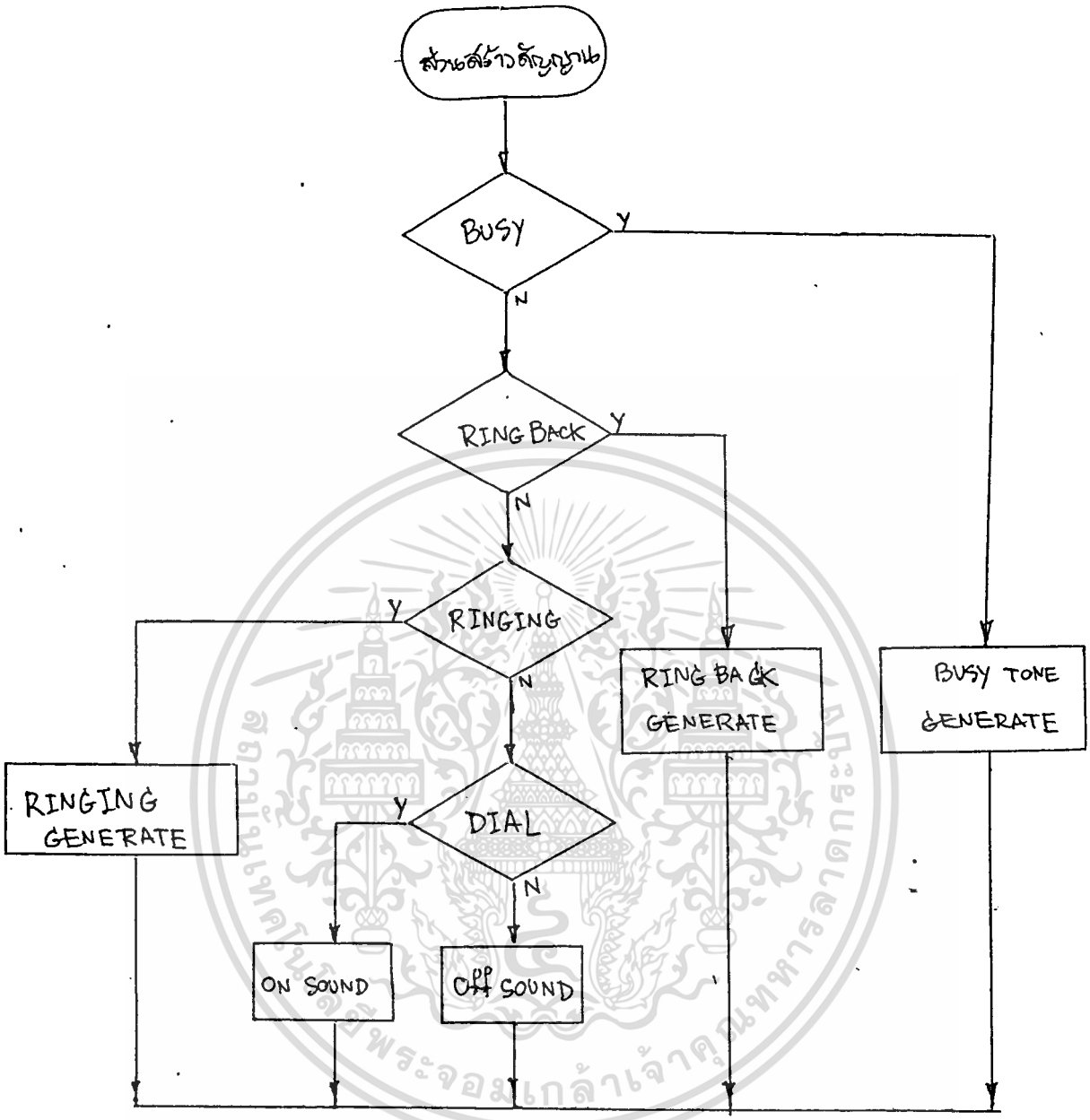
รูปที่ 8 ค. FLOW CHART โปรแกรมตรวจสอบการวางหู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



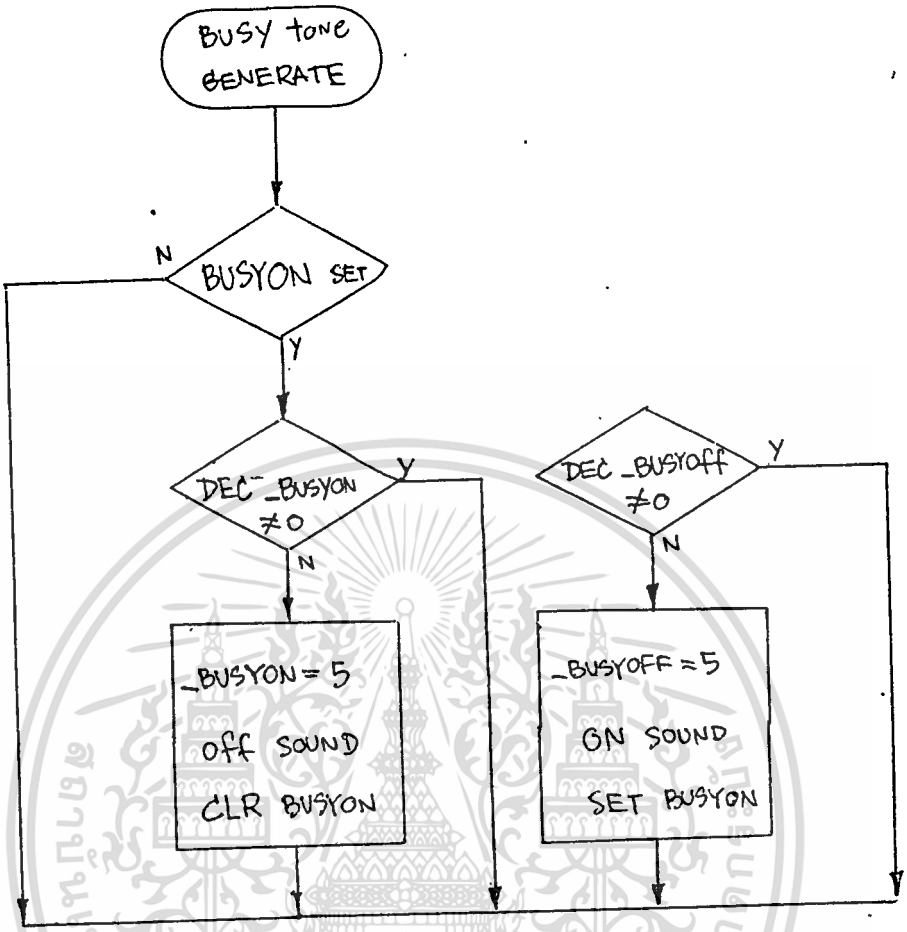
รูปที่ 3.3 FLOW CHART โปรแกรมควบคุมการส่งสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



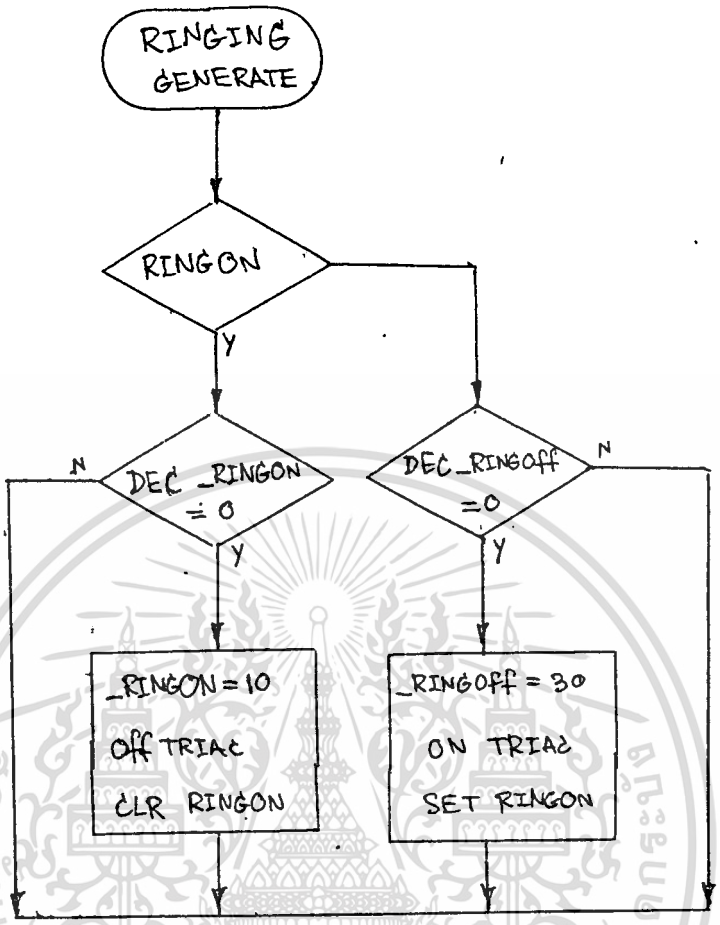
รูปที่ 10 ค FLOW CHART โปรแกรมส่วนสร้างสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 11 ก FLOW CHART โปรแกรม สร้างสัญญาณ BUSY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12 ก FLOWCHART โปรแกรมสร้างสัญญาณ RINGING

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#autostart 10h
#data   _ringbon1 _ringbon2 _ringbon3
#data   hook1  hook2  hook3
#data  sub_pch sub_pcl
#autostart 1Ah
#data  _busyon1 _busyon2   busyon3 index value  hook_
;-------start bit address------
#data  status_0 status_1 status_2 status_3  genbit tempbit
#data  A_8155  B_8155  C_8155
#data  status11 status22 status33

#data  call1 call2 call3  called1 called2 called3  no1 no2
#data   busyoff1  busyoff12  busyoff3
#data  _ringboff1 _ringboff2 _ringboff3
#data  _ringoff1  _ringoff2  _ringoff3
#data  _ringon1  _ringon2   _ringon3
#data  _delay8870  _8870  DTMF  read_call_ temp   KX
#data  y_hook1  y_hook2  y_hook3  w_hook1  w_hook2  w_hook3


std     bit P1.0
st_sw   bit P1.1
data_sw bit P1.2
sound1  bit P1.3
sound2  bit P1.4
sound3  bit P1.5
hook_1   bit A_8155.4
hook_2   bit A_8155.5
hook_3   bit A_8155.6
ring_det bit A_8155.7
ring_of1 bit B_8155.4
ring_of2 bit B_8155.5
ring_of3 bit B_8155.6
hold     bit B_8155.7
analog1  bit C_8155.0
analog2  bit C_8155.1
analog3  bit C_8155.2
analog4  bit C_8155.3
sound0   bit C_8155.4

'time1   equ -35h
'timeh   equ -82h
readed    bit genbit.0
done      bit genbit.1
delay8870 bit genbit.2
ondet     bit genbit.3

b0     bit tempbit.0
b1     bit tempbit.1
b2     bit tempbit.2
b3     bit tempbit.3
b4     bit tempbit.4
b5     bit tempbit.5

ringb0 bit status_0.0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
dial0   bit status 0.1
busy0   bit status 0.2
ex      bit status 0.3
busyon0  bit status 0.4
ringbon0  bit status 0.5

ringb1  bit status 1.0
dial1   bit status 1.1
busy1   bit status 1.2
hook1   bit status 1.3
busyon1  bit status 1.4
ringbon1  bit status 1.5
wang_k1   bit status 1.6
yok_w1    bit status 1.7

ringb2  bit status 2.0
dial2   bit status 2.1
busy2   bit status 2.2
hook2   bit status 2.3
busyon2  bit status 2.4
ringbon2  bit status 2.5
wang_k2   bit status 2.6
yok_w2    bit status 2.7

ringb3  bit status 3.0
dial3   bit status 3.1
busy3   bit status 3.2
hook3   bit status 3.3
busyon3  bit status 3.4
ringbon3  bit status 3.5
wang_k3   bit status 3.6
yok_w3    bit status 3.7

koa1      bit status11.0
ring1     bit status11.1
ringon1   bit status11.2
yhook1   bit status11.3
whook1   bit status11.4
use1     bit status11.5
det1     bit status11.6

koa2      bit status22.0
ring2     bit status22.1
ringon2   bit status22.2
yhook2   bit status22.3
whook2   bit status22.4
use2     bit status22.5
det2     bit status22.6

koa3      bit status33.0
ring3     bit status33.1
ringon3   bit status33.2
yhook3   bit status33.3
whook3   bit status33.4
use3     bit status33.5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
det3     bit status33.6


      org 2000h
    ' ljmp main
sub timer()
#asm
  push ACC
  push PSW
  mov psw,#00001000b   ;select bank 1
  mov TL0,#timel
  mov TH0,#timeh
#easm
 if  delay8870 !=0
  'dec  delay8870
  if  delay8870=0
   setb delay8870
  ; call sw8870(4)    ; test=======
  else
   clr delay8870
  endif
 else
  setb delay8870
  ;call sw8870(4)    ;testttttttttttttttt
 endif
  call timer_1()
  call timer_2()
  call timer_3()
#asm
  pop PSW
  pop ACC
#easm
endint


sub timer_1()

 if b yhook1
   'dec y_hook1
 endif
 if b whook1
   'dec w_hook1
 endif

 if b  hook1
   if  hook1 != 0
     'dec  hook1
     if  hook1 !=0
       clr wang_k1
     else
       setb wang_k1
     endif
   endif
 else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  else   ; not ringb1
   if b  dial1
    setb sound1
   else
    clr sound1
   endif
  endif  ; not ringb1
 endif   ; not busy1
}endif   ; not ring1
endsub


sub timer_2()
 if b yhook2
   'dec y_hook2
 endif
 if b whook2
   'dec w_hook2
 endif

 if b  hook2
    if _hook2 != 0
     'dec  hook2
     if _hook2 !=0
       clr wang_k2
     else
       setb wang_k2
     endif
   endif
 else
    hook2=5
 endif

 if b  ring2
 { if b  ringon2
 /    'dec _ringon2
      if  ringon2 = 0
         ringon2=15
        call set_ringof(2)
        clr ringon2
      endif
   else
     'dec _ringoff2
     if  ringoff2 = 0
        ringoff2=20
       call set_ringon(2)
       setb ringon2
     endif
   endif
 }else    ; not ring2
 {
 if b  busy2
   if b  busyon2
    'dec _busyon2
    if _busyon2 = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        busyon2=5
        clr sound2
        clr busyon2
      endif
    else
      'dec _busyoff2
      if _busyoff2 = 0
        _busyoff2=5
        setb sound2
        setb busyon2
      endif
    endif
  else    ; not busy2
   if b  ringb2
    if b  ringbon2
      'dec _ringbon2
      if _ringbon2 = 0
         ringbon2=15
        clr sound2
        clr ringbon2
      endif
    else
      'dec  ringboff2
      if _ringboff2 = 0
         ringboff2=20
        setb sound2
        setb ringbon2
      endif
    endif
   else   ; not ringb2
    if b  dial2
     setb sound2
    else
     clr sound2
    endif
   endif  ; not ringb2
  endif   ; not busy2
}endif   ; not ring2
endsub

sub timer_3()

 if b yhook3
   'dec y_hook3
 endif
 if b whook3
   'dec w_hook3
 endif

 if b  hook3
   if _hook3 != 0
     'dec _hook3
     if _hook3 !=0
       clr wang_k3
     else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    _hook1=5
  endif


  if_b  ring1
  { if_b  ringon1
      'dec _ringon1
      if _ringon1 = 0
        _ringon1=15
        call set_ringof(1)
        clr ringon1
      endif
    else
      'dec _ringoff1
      if _ringoff1 = 0
         ringoff1=20
        call set_ringon(1)
        setb ringon1
      endif
    endif
  }else    ; not ring1
  {
  if_b  busy1
    if_b  busyon1
      'dec  busyon1
      if  busyon1 = 0
         busyon1=5
        clr sound1
        clr busyon1
      endif
    else
      'dec  busyoff1
      if  busyoff1 = 0
         busyoff1=5
        setb sound1
        setb busyon1
      endif
    endif
  else    ; not busy1
   if_b  ringb1
    if_b  ringbon1
      'dec  ringbon1
      if _ringbon1 = 0
         _ringbon1=15
        clr sound1
        clr ringbon1
      endif
    else
      'dec _ringboff1
      if _ringboff1 = 0
        _ringboff1=20
        setb sound1
        setb ringbon1
      endif
    endif
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      setb wang_k3
    endif
  endif
else
  _hook3=5
endif


if_b  ring3
{ if_b  ringon3
    'dec _ringon3
    if _ringon3 = 0
      _ringon3=15
      call set_ringof(3)
      clr ringon3
    endif
  else
    'dec _ringoff3
    if _ringoff3 = 0
       _ringoff3=20
      call set_ringon(3)
      setb ringon3
    endif
  endif
}else    ; not ring3
{
if b  busy3
  if_b  busyon3
    'dec  busyon3
    if _busyon3 = 0
      _busyon3=5
      clr sound3
      clr busyon3
    endif
  else
    'dec _busyoff3
    if _busyoff3 = 0
       _busyoff3=5
      setb sound3
      setb busyon3
    endif
  endif
else    ; not busy3
 if_b  ringb3
  if_b  ringbon3
    'dec _ringbon3
    if _ringbon3 = 0
      _ringbon3=15
      clr sound3
      clr ringbon3
    endif
  else
    'dec _ringboff3
    if _ringboff3 = 0
      _ringboff3=20
      setb sound3
      setb ringbon3
    endif
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    endif
   else    ; not ringb3
    if b  dial3
     setb sound3
    else
     clr sound3
    endif
   endif  ; not ringb3
 endif    ; not busy3
}endif    ; not ring3
endsub ;timer3


sub sw8870(index)
#asm
   mov r2,index
   mov _8870,r2
   inc r2       ; must rotate before
   mov a,#80h   ; rotate mask  1000 0000
loopsw88:
   rl a
   djnz r2,loopsw88
   anl a,#0Fh
   anl C_8155,#0F0h
   orl C_8155,a
#easm
   call outport_c()
endsub

sub sw(no1, no2)
 temp=4*no1
 temp=temp+no2
 a=C_8155
 'anl a,#0fh
 'orl a,c_8155
 call outport_c()
 setb st_sw
 clr st_sw
 temp=4*no2
 temp=temp+no1
 a=C_8155
 'anl a,#0fh
 'orl a,c_8155
 call outport_c()
 setb st_sw
 clr st_sw
endsub

sub inport_A()
 'mov r0,#1
 'movx a,@r0
 'mov A_8155,a
 'anl a,#0fh
 'mov DTMF,a
endsub

sub outport_B()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  'mov r0,#2
  'mov a,B_8155
  'movx @r0,a
endsub

sub outport_C()
  'mov r0,#3
  'mov a,C_8155
  'movx @r0,a
endsub

sub clr_all1()
   clr busy1
   clr ringb1
   clr dial1
   clr ring1
endsub

sub clr_all2()
   clr busy2
   clr ringb2
   clr dial2
   clr ring2
endsub

sub clr_all3()
   clr busy3
   clr ringb3
   clr dial3
   clr ring3
endsub

sub clr_all()
   clr sound1
   clr sound2
   clr sound3
   clr sound0
endsub

sub clr_use()
  if_b det1
    if called1=4
      if call1=4
        clr use1
      endif
    endif
  endif

  if_b det2
    if called2=4
      if call2=4
        clr use2
      endif
    endif
  endif
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  if b det3
   if called3=4
     if call3=4
       clr use3
     endif
   endif
  endif

endsub


sub off_ring()
  call set_ringof(1)
  call set_ringof(2)
  call set_ringof(3)
endsub

sub set_call (index,value)
 'dec index
  index = #call1 +index
  ^index= value
endsub

sub read_call (index)
 'dec index
  index = #call1 +index
  read_call_ =^index
endsub


func set_called (index,value)
 'dec index
  index = #called1 +index
  ^index=value
endfunc

func hook( index)
#asm
   mov r2,index
   mov a,#08    ; rotate mask  00001000
loophook:
   rl a
   djnz r2,loophook
   anl a,A_8155
   mov hook_,a
#casm
endfunc

sub set_ringof( index)
#asm
   mov r2,index
   mov a,#08    ; rotate mask  00001000
loopringof:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    rl a
    djnz r2,looprinqof
    orl B_8155,a
#easm
    call outport_B()
endsub

sub set_ringon( index)
#asm
    mov r2,index     ; ring on when port=0
    mov a,#0F7h      ; rotate mask  1111 0111
looprinqon:
    rl a
    djnz r2,looprinqon
    anl B_8155,a
#easm
    call outport_B()
endsub

sub set_busy(index)
#asm
    mov a,index
    rl a
    rl a
    mov dptr,#ins_setbu
    jmp @a+dptr
ins_setbu:
     setb busy0      ; case 0
     ret
     nop          ; insert to 4 byte
     setb busy1      ; case 1
     ret
     nop
     setb busy2      ; case 2
     ret
     nop
     setb busy3      ; case 2
     ret
     nop
     ret             ; case 4
#easm
endsub

sub clr_alln(index)
#asm
    mov a,index
    rl a
    rl a
    rl a
    mov dptr,#ins_clrall
    jmp @a+dptr
ins_clrall:
     clr busy0       ; case 0
     clr ringb0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    ret
    nop
    nop
    nop
    clr busy1     ; case 1
    clr ringb1
    clr dial1
    ret
    nop
    clr busy2     ; case 2
    clr ringb2
    clr dial2
    ret
    nop
    clr busy3     ; case 3
    clr ringb3
    clr dial3
    ret
    nop
                  ; case 4
#easm
endsub

sub set_ringb(index)
#asm
   mov a,index
   rl a
   rl a
   mov dptr,#ins_setrb
   jmp @a+dptr
ins_setrb:
    setb ringb0     ; case 0
    ret
    nop         ; insert to 4 byte
    setb ringb1     ; case 1
    ret
    nop
    setb ringb2     ; case 2
    ret
    nop
    setb ringb3     ; case 2
    ret
    nop
    ret           ; case 4
#easm
endsub

sub clr_ringb(index)
#asm
   mov a,index
   rl a
   rl a
   mov dptr,#ins_clrrb
   jmp @a+dptr
ins_clrrb:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    clr ringb0     ; case 0
    ret
    nop          ; insert to 4 byte
    clr ringb1     ; case 1
    ret
    nop
    clr ringb2     ; case 2
    ret
    nop
    clr ringb3     ; case 2
    ret
    nop
    ret            ; case 4
#easm
endsub

sub clr_ring(index)
#asm
   mov a,index
   dec a
   rl a
   rl a
   mov dptr,#ins_clrr
   jmp @a+dptr
ins_clrr:
    clr ring1     ; case 1
    ret
    nop
    clr ring2     ; case 2
    ret
    nop
    clr ring3     ; case 2
    ret
    nop
    ret           ; case 4
#easm
endsub

sub init8155()
#asm
  mov r0,#0
  mov a,#0Eh
  movx @r0,a
#easm
endsub

sub init8870()
  call sw8870(4)
  clr delay8870
  delay8870=0
  setb done
  clr readed
endsub

sub case_1()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if DTMF=1
 setb busy1
 call init8870()
 clr koa1
 clr yok_w1
 'ret
endif
if DTMF=2
    if called1 !=0
      if called1 != 4
       temp = read_call(called1)
       if temp!=1
         setb busy1
         'ljmp endcase12
       endif
      endif
    endif

    if_b use2
       if called2=1
         if call2=4
           call sw(1,2)
           call1=2
         else
           setb busy1
         endif
        else
         setb busy1
        endif
    else
        setb ring2
        setb ringb1
        call1 =2
        called2=1
     endif
  'endcase12:
   call init8870()
   clr koa1
   clr yok_w1
   'ret
 endif
if DTMF=3
    if called1 !=0
      if called1 != 4
       temp = read_call(called1)
       if temp!=1
         setb busy1
         'ljmp endcase13
       endif
      endif
    endif

    if_b use2
       if called2=1
         if call2=4
           call sw(1,2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            call1=2
          else
            setb busy1
          endif
        else
          setb busy1
        endif
      else
        setb ring2
        setb ringb1
        call1 =2
        called2=1
      endif
   'endcase13:
    call init8870()
    clr koa1
    clr yok w1
    'ret
 endif
 '
endsub   ; end case_t1

sub case_t2()
 if DTMF=2
  setb busy2
  call init8870()
  clr koa2
  clr yok w2
  'ret
 endif
 if DTMF=1
     if called2 !=0
       if called2 != 4
        temp = read call(called2)
        if temp!=2
          setb busy2
          'ljmp endcase21
        endif
       endif
     endif

     if b use1
        if called1=2
          if call2=4
            call sw(2,1)
            call2=1
          else
            setb busy2
          endif
        else
          setb busy2
        endif
      else
        setb ring1
        setb ringb2
        call2 =1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          called1=2
        endif
    'endcase21:
     call init8870()
     clr koa2
     clr yok w2
     'ret
  endif
   if DTMF=3
      if called1 !=0
        if called1 != 4
         temp = read call(called1)
         if temp!=1
           setb busy1
           'ljmp endcase23
         endif
       endif
      endif

      if b use2
         if called2=1
           if call2=4
             call sw(1,2)
             call1=2
           else
             setb busy1
           endif
         else
           setb busy1
         endif
       else
         setb ring2
         setb ringb1
         call1 =2
         called2=1
       endif
    'endcase23:
     call init8870()
     clr koa1
     clr yok w1
     'ret
 endif

endsub   ; end case t2



sub check_h1()
  if b hook 1
  {   if b hook1
      {  ;---------- left 2---------
         if b yhook1
          if y_hook1=0
            clr yhook1
          else
            'ljmp next_st2
          endif
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    endif

    if b  wanq k1
    {  clr use1     ; add 11
       if call1 !=4
       {  if called1 = 4
          {   call set_called (call1 ,4)
              temp= hook( call1)
            if temp !=0
              call clr ringq(call1)
              call set_ringof(call1)
            else
              call set busy(call1)
            endif
          }else
            call set_called(call1,called1)
            call set_call( call1,4)
            call sw(called1,call1)
            temp=hook(call1)
            if temp!=0
              call set_ringb(called1)
            endif
          endif
        }else
        { if called1=4
            'ljmp next_st2
          else
            if b ring1
            'ljmp next_st2
            endif
            if called1=0
             clr hold
             clr ex
            endif
          endif
        }endif
     called1 =4
     call1 =4
     call clr_all1()
  }endif   ;--if wanq k1
}else
  ;--------- left }---------
   if b  whook1
     if w_hook1=0
       setb hook1
       clr whook1
     else
       'ljmp next_st2
     endif
   else
     setb whook1
     w_hook1=3
     'ljmp next_st2
   endif
;-------- end bound ---------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      hook1=b
     clr yok w1
     clr koa1
     ;-call1=4
     call clr all()
    endif    ;--else hook1
}else
 {  if b  hook1
    { ;--------- right 2 --------
      if b  yhook1
        if y hook1=0
          clr hook1
          clr yhook1
        else
          'ljmp next st2
        endif
      else
        setb yhook1
        y hook1=3
        'ljmp next st2
      endif
      ;---------- end check bound-----
      ; clr ring1  ;will check after
      call set ringof(1)
      if b  wang k1
        setb yok w1
        clr  koa1
      else
        setb koa1
        clr  yok w1
      endif
      if nb  use1
      {  if b  ring1
         {  if b det1
              call sw(1,0)
              ;call clr all0()  ;******ringb0
              clr yok w1
              clr det1
              setb hold
              clr ondet
             else
              call sw(1,called1)
              call clr alln(called1)
              clr  yok w1
              clr  ring1
              call set ringof(1)
             endif
          }else
           {  if b  ondet
                call clr all()
                call off_ring()
                clr ringb0
                 EX =1
                call sw(0,1)
                called1=0
                call1 =4
                setb hold
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                call clr_use()
                clr det1
                clr det2
                clr det3
                clr ondet
                clr yok_w1
              endif
            }endif
          }endif
           setb use1

      }
      else
        ;----------right 1---------
        if b whook1
          if w hook1=0
            clr whook1
          else
            'ljmp next_st2
          endif
        endif

        if _8870=1
          if b delay8870
            setb busy1
            ; 8870=4
            call sw8870(4)
            clr koal
            clr yok_w1
            clr delay8870
          endif
        endif
      endif
    }endif  ;--else hook 1

'next_st2:
endsub  ;----- end check h1


sub check_h2()
  if b hook_2
  {   if b hook2
      {  ;---------- left 2---------
         if b yhook2
          if y_hook2=0
            clr yhook2
          else
            'ljmp next_st3
          endif
         endif

         if b  wang_k2
         {  clr use2     ; add 11
            if call2 !=4
            {  if called2 = 4
               {  call set_called (call2 ,4)
                  temp= hook( call2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                if temp !=0
                  call clr ringq(call2)
                  call set ringof(call2)
                else
                  call set busy(call2)
                endif
              }else
                call set called(call2,called2)
                call set call( call2,4)
                call sw(called2,call2)
                temp=hook(call2)
                if temp!=0
                  call set ringb(called2)
                endif
               endif
           }else
           { if called2=4
              'ljmp next st3
             else
               if b ringq2
                'ljmp next st3
               endif
               if called2=0
                clr hold
                clr ex
               endif
              endif
          }endif
       called2 =4
       call2 =4
       call clr all2()
     }endif    ;-if wang k2
   }else
     ;--------- left 1-----
      if b  whook2
        if w hook2=0
          setb hook2
          clr whook2
        else
          'ljmp next st3
        endif
      else
        setb whook2
        w hook2=3
        'ljmp next st3
      endif
     ;-------- end bound --------
      hook2=5
     clr yok w2
     clr koa2
     ;-call2=4
     call clr all2()
    endif     ;- else hook2
}else
 {  if b  hook2
    { ;--------- right 2 --------
      if b  yhook2
        if y hook2=0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      clr hook2
      clr yhook2
    else
      'ljmp next st3
    endif
  else
    setb yhook2
    y_hook2=3
    'ljmp next st3'
  endif
  ;---------- end check bound-----
  ;clr ring2    -will check alter
  call set ringof(2)
  if_b  wang_k2
    setb yok_w2
    clr  koa2
  else
    setb koa2
    clr  yok_w2
  endif
  if_nb  use2
  {  if_b  ring2
     {  if_b det2
          call sw(2,0)
          ;call clr all0()  ;******ringb0
          clr yok_w2
          clr det2
          setb hold
          clr ondet
        else
          call sw(2,called2)
          call clr all_n(called2)
          clr  yok_w2
          clr  ring2
          call set ringof(2)
        endif
      }else
       {  if_b  ondet
            call clr all()
            call off ring()
            clr ringb0
             EX =2
            call sw(0,2)
            called2=0
            call2 =4
            setb hold
            call clr use()
            clr det2
            clr det2
            clr det3
            clr ondet
            clr yok_w2
          endif
      }endif
  }endif
  setb use2
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      else
        ;---------- right ] ------ -
        if b whook2
          if w hook2=0
            clr whook2
          else
            'ljmp next st3
          endif
        endif

        if  8870=2
          if b delay8870
            setb busy2
            ; 8870=4
            call sw8870(4)
            clr koa2
            clr yok w2
            clr delay8870
          endif
        endif
      endif
    }endif  ;--else hook 2

'next st3:
endsub  ;----- end check h2

sub check h3()
  if b hook 3
  {   if b hook3
      {  ;---------- left 2 ---------
         if b yhook3
          if y hook3=0
            clr yhook3
          else
            'ljmp next st4
          endif
         endif

         if b  wang k3
         {  clr use3     ; add 33
            if call3 !=4
            {  if called3 = 4
               {   call set called (call3 ,4)
                   temp= hook( call3)
                  if temp !=0
                    call clr ring(call3)
                    call set ringof(call3)
                  else
                    call set busy(call3)
                  endif
                }else
                  call set called(call3,called3)
                  call set call( call3,4)
                  call sw(called3,call3)
                  temp=hook(call3)
                  if temp!=0
                    call set ringb(called3)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            endif
           endif
         }else
         { if called3=4
            'ljmp next_st4
           else
            if b ring3
             'ljmp next_st4
            endif
            if called3=0
             clr hold
             clr ex
            endif
           endif
         }endif
       called3 =4
       call3 =4
       call clr_all3()
     }endif    ;--if wang_k3
   }else
     ;--------- left 3---------
      if b  whook3
        if w_hook3=0
          setb hook3
          clr whook3
        else
          'ljmp next_st4
        endif
      else
        setb whook3
        w_hook3=3
        'ljmp next_st4
      endif
     ;-------- end bound --------
      hook3=5
     clr yok_w3
     clr koa3
     ;-call3=4
     call clr_all3()
    endif     ;--else hook3
}else
 {  if b  hook3
    { ;--------- right 2 --------
      if b  yhook3
        if y_hook3=0
          clr hook3
          clr yhook3
        else
          'ljmp next_st4
        endif
      else
        setb yhook3
        y_hook3=3
        'ljmp next_st4
      endif
     ;-------- end check bound-----
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; clr ring3  ;will check after
call set ringof(3)
if b  wang k3
  setb yok_w3
  clr  koa3
else
  setb koa3
  clr  yok_w3
endif
if nb  use3
{  if b  ring3
   {  if b det3
        call sw(3,0)
        ;call clr all0()  ;******ringb0
        clr yok_w3
        clr det3
        setb hold
        clr ondet
       else
        call sw(3,called3)
        call clr alln(called3)
        clr  yok_w3
        clr  ring3
        call set ringof(3)
       endif
   }else
    {  if b  ondet
         call clr all()
         call off ring()
         clr ringb0
          MX =3
         call sw(0,3)
         called3=0
         call3 =4
         setb hold
         call clr use()
         clr det3
         clr det2
         clr det3
         clr ondet
         clr yok_w3
       endif
   }endif
  }endif
   setb use3

}
else
  ;---------- right 3-----------
  if b whook3
    if w_hook3=0
      clr whook3
    else
      ljmp next st4
    endif
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        endif

        if _8870=3
          if b delay8870
            setb busy3
            ; 8870=4
            call sw8870(4)
            clr koa3
            clr yok_w3
            clr delay8870
          endif
        endif
      endif
    }endif  ;--else hook 3

'next_st4:
endsub  ;----- end check h3

sub  check_k1()
 ' mov c,koa1
 ' orl c,yok_w1
 ' mov b0,c
 if b  b0   ;--= if b  koa1|yok_w1
 {  if _8870=4
    { if b yok_w1
        setb dial1
        clr busy1
        clr ringb1
      else
        call clr_all1()
        clr sound1
      endif
      ; 8870=1
      call sw8870(1)
      delay8870=5
      clr delay8870
    }else
      if _8870 !=1
       setb busy1
       'ljmp next_k2
      endif
     endif
    if b  std
      if nb readed
       clr  done
       setb readed
      endif
    else
      clr readed
    endif
    if_nb done
      call case_t1()  ;----station 1
    endif
 }endif  ;---koa | yok w

'next_k2:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
endsub  ;---end check k1

sub  check_k2()
 ' mov c,koa2
 ' orl c,yok_w2
 ' mov b0,c
 if_b  b0   ;--- if_b  koa2|yok_w2
 {   if `8870=4
     { if_b yok_w2
         setb dial2
         clr busy2
         clr ringb2
       else
         call clr_all2()
         clr sound2
       endif
       ; 8870=2
       call sw8870(2)
        delay8870=5
       clr delay8870
     }else
       if _8870 !=2
        setb busy2
        'ljmp next_k3
       endif
      endif
     if_b  std
       if_nb readed
        clr  done
        setb readed
       endif
     else
       clr readed
     endif
     if nb done
       call case_t2()  ;---station 2
     endif
 }endif  ;--koa | yok w

'next_k3:

endsub  ;---end check k2

sub  check_k3()
 ' mov c,koa3
 ' orl c,yok_w3
 ' mov b0,c
 if_b  b0   ;--- if_b  koa3|yok_w3
 {   if _8870=4
     { if_b yok_w3
         setb dial3
         clr busy3
         clr ringb3
       else
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        call clr all3()
        clr sound3
      endif
      ; 8870=3
      call sw8870(3)
       delay8870=5
      clr delay8870
    }else
      if 8870 !=3
       setb busy3
       'ljmp next k4
      endif
     endif
    if b std
      if nb readed
       clr done
       setb readed
      endif
    else
      clr readed
    endif
    if nb done
     ; call case_t3()  ;----station 3
    endif
 }endif  ;---koa | yok w

'next k4:

endsub  ;---end check k3

'main:   ;========== main ==========
  busyon1=5
  busyoff1=5
  ringbon1=15
  ringboff1=20
  busyon2=5
  busyoff2=5
  ringbon2=15
  ringboff2=20
  ringon1=15
  ringoff1=20
  ringon2=15
  ringoff2=20

  clr p1.7   ;enable
  call init8155()
  call clr all1()
  call clr all2()
  call clr all3()
  call set ringof(1)
  call set ringof(2)
  call set ringof(3)

  clr sound1
  clr sound2
  clr sound3
  clr  sound0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA

MC3419-1L
MC3419A-1L
MC3419C-1L

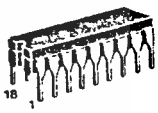
TELEPHONE LINE-FEED CIRCUIT

. . . designed as the heart of a circuit to provide BORSHT functions for telephone service in Central Office, PABX, and Subscriber Carrier equipment. This circuit provides dc power for the telephone (Battery), Overvoltage protection, Supervision features such as hook status and dial pulsing, two-wire differential to four-wire single-ended conversions and suppression of longitudinal signals at the two-wire input (Hybrid), and facilitates ringing insertion, Ring trip detection and Testing.

- Totally Upward Compatible with the MC3419
- All Key Parameters Externally Programmable
- Current Sensing Outputs Monitor Status of Both Tip and Ring Leads for Auxiliary Functions such as: Ground Key, Ring Trip, Message Waiting Lamp, etc.
- On-Hook Power Below 5.0 mW
- Digital Hook Status Output
- Powerdown Input
- Ground Fault Protection
- Operates from Single −20 V to −56 V Power Source
- Size and Weight Reduction Over Conventional Approaches
- The sale of this product is licensed under Patent No. 4,004,109. All royalties related to this patent are included in the unit price.

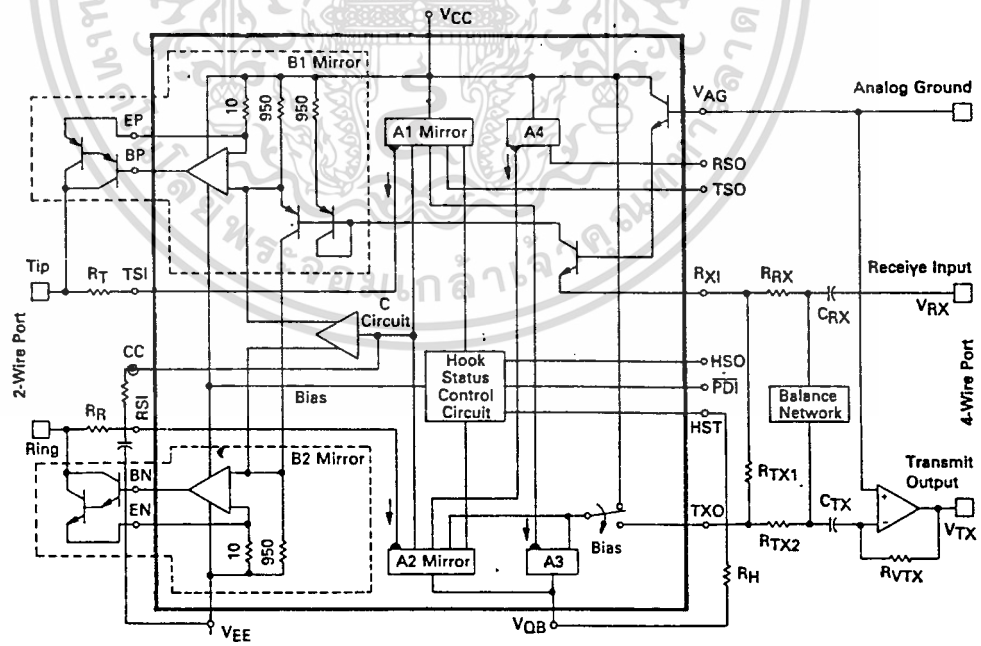
SUBSCRIBER LOOP INTERFACE CIRCUIT (SLIC)

BIPOLAR LASER-TRIMMED INTEGRATED CIRCUIT



L SUFFIX
CERAMIC PACKAGE
CASE 726-01

FUNCTIONAL BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA

MC34012-1
MC34012-2
MC34012-3

Advance Information

TELEPHONE TONE RINGER

- Complete Telephone Bell Replacement Circuit with Minimum External Components
- On-Chip Diode Bridge and Transient Protection
- Direct Drive for Piezoelectric Transducers
- Base Frequency Options—MC34012-1, 1.0 kHz
  MC34012-2, 2.0 kHz
  MC34012-3, 500 Hz
- Input Impedance Signature Meets Bell and EIA Standards
- Rejects Rotary Dial Transients

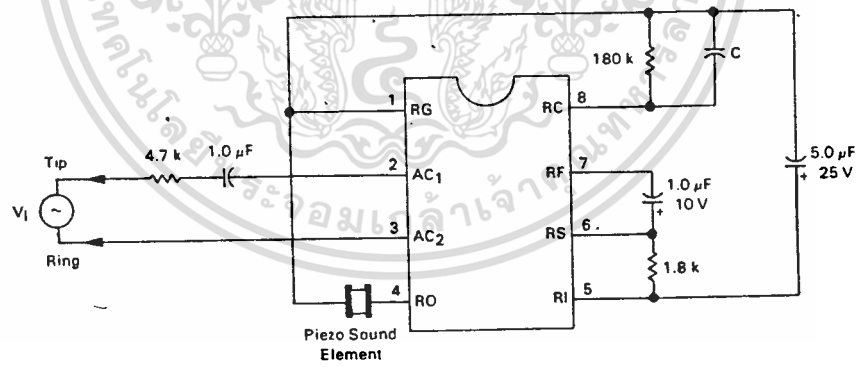
TELEPHONE TONE RINGER

BIPOLAR LINEAR/I^2L

N SUFFIX
PLASTIC PACKAGE
CASE 626-04

D SUFFIX
PLASTIC PACKAGE
CASE 751-01
SO-8

APPLICATION CIRCUIT



MC34012-1: C = 1000 pF
MC34012-2: C = 500 pF
MC34012-3: C = 2000 pF

This document contains information on a new product. Specifications and information herein are subject to change without notice

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

MOTOROLA

MC34014

Specifications and Applications Information

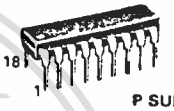
TELEPHONE SPEECH NETWORK WITH DIALER INTERFACE

The MC34014 is a Telephone Speech Network integrated circuit which incorporates adjustable transmit, receive, and sidetone functions, a dc loop interface circuit, tone dialer interface, and a regulated output voltage for a pulse/tone dialer. Also included is an equalization circuit which compensates gains for line length variations. The conversion from 2-to-4 wire is accomplished with a supply voltage as low as 1.5 volts. The MC34014 is packaged in a standard 18-pin (0.3" wide) plastic DIP, and a 20-pin surface mount PLCC package.

- Transmit, Receive, and Sidetone Gains Set by External Resistors
- Loop Length Equalization for Transmit, Receive, and Sidetone Functions
- Operates Down to 1.5 volts (V+) in Speech Mode
- Provides Regulated Voltage for CMOS Dialer
- Speech Amplifiers Muted During Pulse and Tone Dialing
- DTMF Output Level Adjustable with a Single Resistor
- Compatible with 2-Terminal Electret Microphones
- Compatible with Receiver Impedances of 150 Ω and Higher

TELEPHONE SPEECH NETWORK WITH DIALER INTERFACE

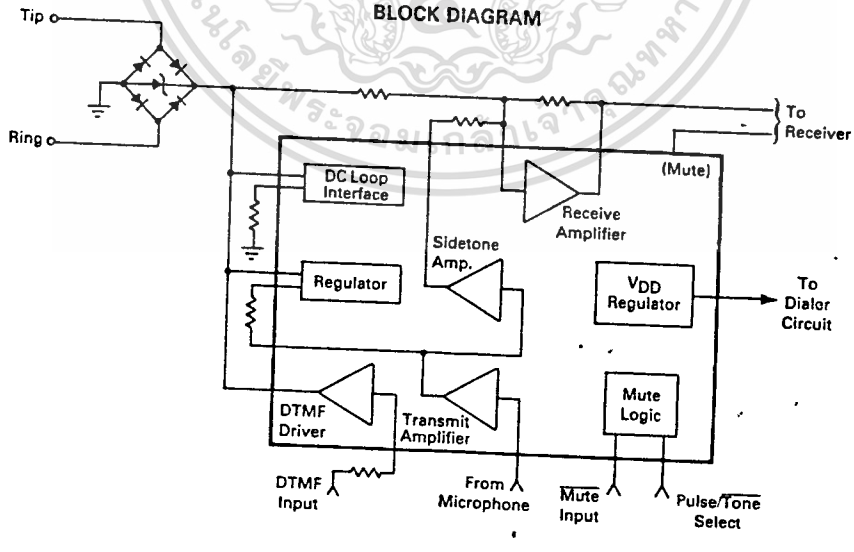
SILICON MONOLITHIC INTEGRATED CIRCUIT



P SUFFIX
PLASTIC PACKAGE
CASE 707-02

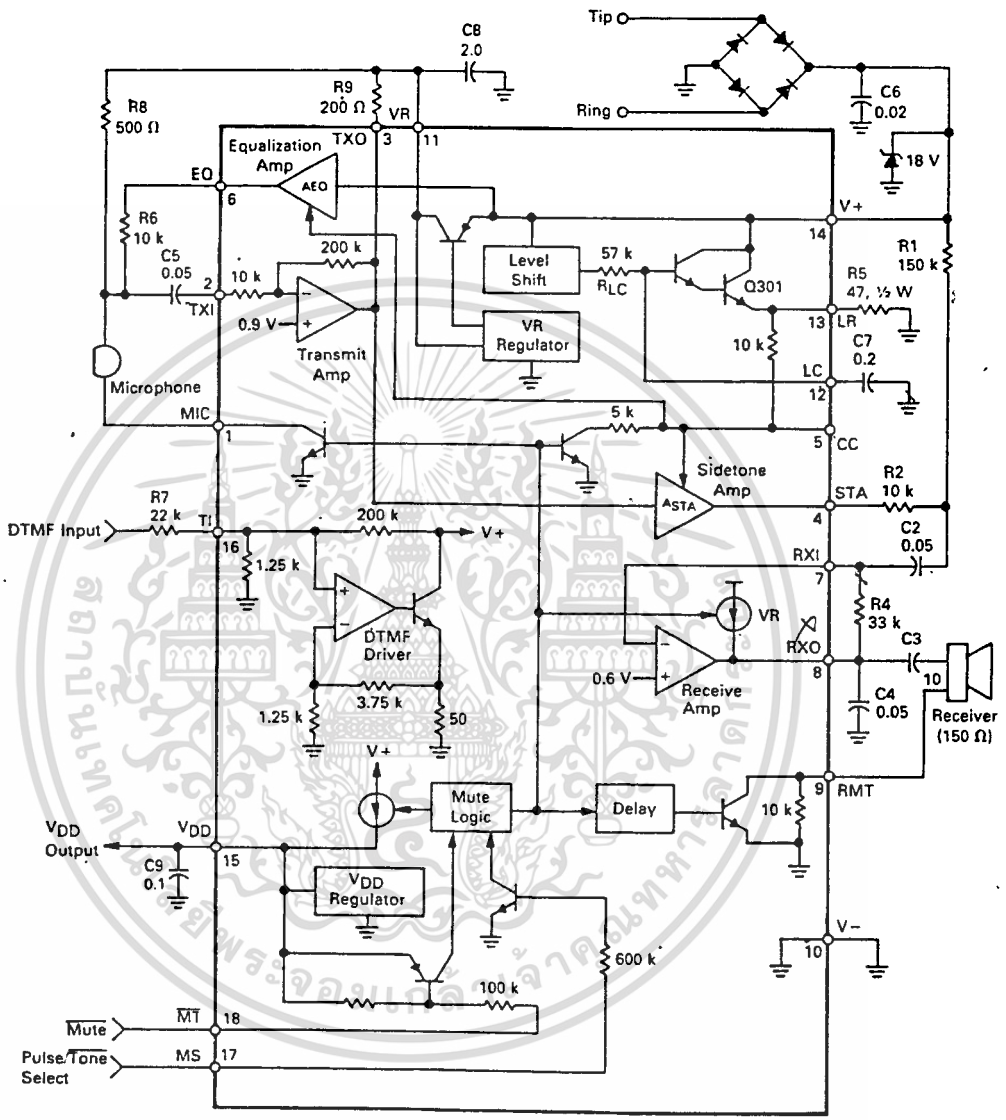
FN SUFFIX
PLASTIC PACKAGE
CASE 775-01
PLCC-20

BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 1 — TEST CIRCUIT



NOTE: Pin numbers are for 18 pin DIP.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA

MC142100
MC145100

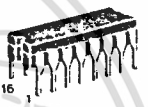
4 × 4 CROSSPOINT SWITCH WITH CONTROL MEMORY

The MC142100 and MC145100 consist of 16 crosspoint switches (analog transmission gates) organized in 4 rows and 4 columns. Both devices have 16 latches, each of which controls the state of a particular switch. Any of the 16 switches can be selected by applying its address to the device and a pulse to the strobe input. The selected crosspoint will turn on if during strobe, Data In was a one and will turn off if during strobe, Data In was a zero. In addition the MC145100 will reset all non-selected switches in the same row as the selected switch. Other switches are unaffected. In the MC145100, an internal power-on reset turns off all switches as power is applied.

- Internal Latches Control State of Switches
- Power-On Reset (MC145100 Only)
- Low On Resistance — Typically on 110 Ω @ 10 Vdc
- Large Analog Range ($V_{DD} - V_{SS}$)
- All Pins Are Diode Protected
- Matched Switch Characteristics
- High CMOS Noise Immunity
- MC142100 Pin-for-Pin Replacement for CD22100

CMOS MSI
(LOW-POWER COMPLEMENTARY MOS)

4 × 4 CROSSPOINT SWITCH WITH CONTROL MEMORY

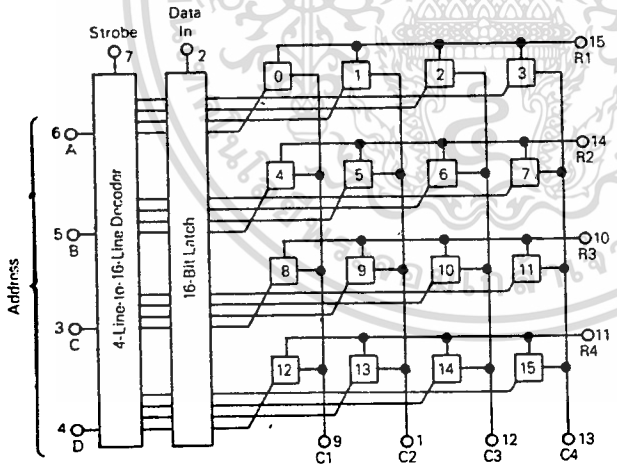


L SUFFIX
CERAMIC PACKAGE
CASE 620

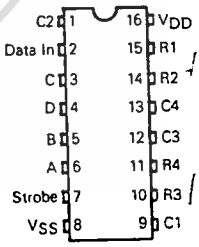
P SUFFIX
PLASTIC PACKAGE
CASE 648

ORDERING INFORMATION

MC14XXXX — Suffix Denotes
- L Ceramic Package
- P Plastic Package
- A Extended Operating Temperature Range
- C Limited Operating Temperature Range



PIN ASSIGNMENTS



MAXIMUM RATINGS (Voltages referenced to V_{SS}, Pin 8)

Rating	Symbol	Value	Unit
DC Supply Voltage	V_{DD}	−0.5 to +18	Vdc
Input Voltage, All Inputs	V_{in}	−0.5 to V_{DD} + 0.5	Vdc
Through Current	I	25	mAdc
Operating Temperature Range AL Device CL/CP Device	T_A	 −55 to +125 −40 to +85	°C
Storage Temperature Range	T_{stg}	−65 to +150	°C

This device contains circuitry to protect the inputs against damage due to high static voltages or electric fields; however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this high impedance circuit. For proper operation it is recommended that V_{in} and V_{out} be constrained to the range $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{DD}$.
Unused control inputs must always be tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}).

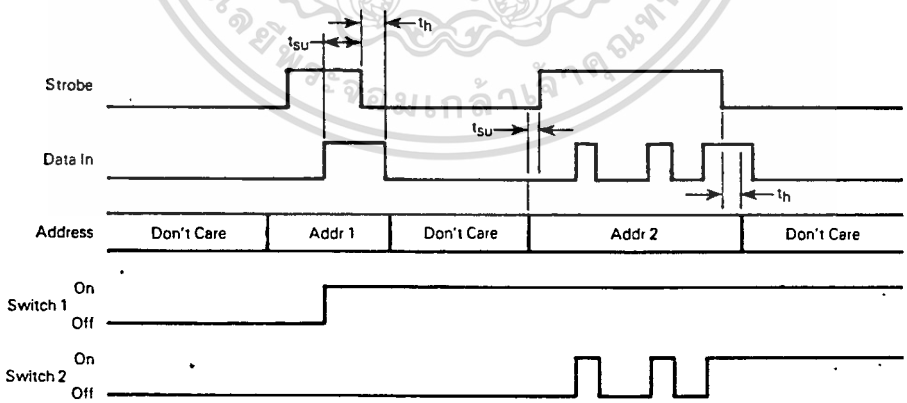
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

SWITCHING CHARACTERISTICS (continued) (V_{SS} = 0, T_A = 25°C, C_L = 50 pF)

Characteristics		Symbol	V_{DD} Vdc	Min	Typ	Max	Unit
Sine Wave Distortion (R_L = 1 kΩ, f = 1 kHz)	See Figure 3	–	10	–	0.5	–	%
Frequency Response (Switch On) (R_L = 1kΩ, 20 Log_{10} V_{out}/V_{in} = – 3.0 dB)	See Figure 3	–	10	–	15	–	MHz
Feedthrough Attenuation (Switch Off) (V_{in} = 10 V_{PP}, F = 1.6 kHz, R_L = 1 kΩ, C_L = 15 pF)	See Figure 3	–	10	–	– 80	–	dB
Frequency for Signal Crosstalk (V_{in} = 10 V_{PP}, Switch A On, Switch B Off, R_L = 1 kΩ, C_L = 15 pF)	– 40 dB	–	10	–	1500	–	kHz
	– 110 dB	–	–	–	0.1	–	kHz
	See Figure 4						
Crosstalk Controls to Output (R_L = 10 kΩ)	See Figure 5	–	10	–	70	–	mV

Address				Switch Selected		MC145100 Only Switches Cleared			Address				Switch Selected		MC145100 Only Switches Cleared		
A	B	C	D						A	B	C	D					
0	0	0	0	C1R1	0	1	2	3	0	0	0	1	C1R3	8	9	10	11
1	0	0	0	C2R1	1	0	2	3	1	0	0	1	C2R3	9	8	10	11
0	1	0	0	C3R1	2	0	1	3	0	1	0	1	C3R3	10	8	9	11
1	1	0	0	C4R1	3	0	1	2	1	1	0	1	C4R3	11	8	9	10
0	0	1	0	C1R2	4	5	6	7	0	0	1	1	C1R4	12	13	14	15
1	0	1	0	C2R2	5	4	6	7	1	0	1	1	C2R4	13	12	14	15
0	1	1	0	C3R2	6	4	5	7	0	1	1	1	C3R4	14	12	13	15
1	1	1	0	C4R2	7	4	5	6	1	1	1	1	C4R4	15	12	13	14

TIMING DIAGRAM
MC145100/MC142100



รูป 3.4.2

MITEL

ISO²-CMOS™ MT8870
Integrated DTMF Receiver
Advance Information

AUG. 1982

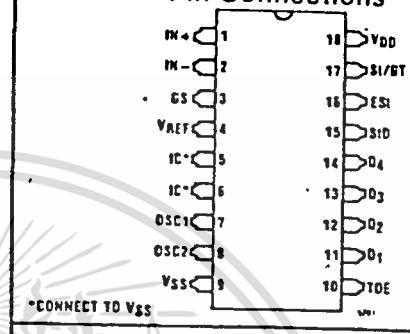
Features

- Full receiver in single 18-pin package
- Central Office quality
- Low power-consumption
- Adjustable acquisition and release times

Applications

- PABX
- Central Office
- Key Systems
- Mobile Radio
- Remote Control
- Remote Data Entry

Pin Connections



Ordering Information

MT8870AC Cerdip, −40°C to +85°C

Description

The MITEL MT8870 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 and dial-tone rejection; 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 3-state bus interface.

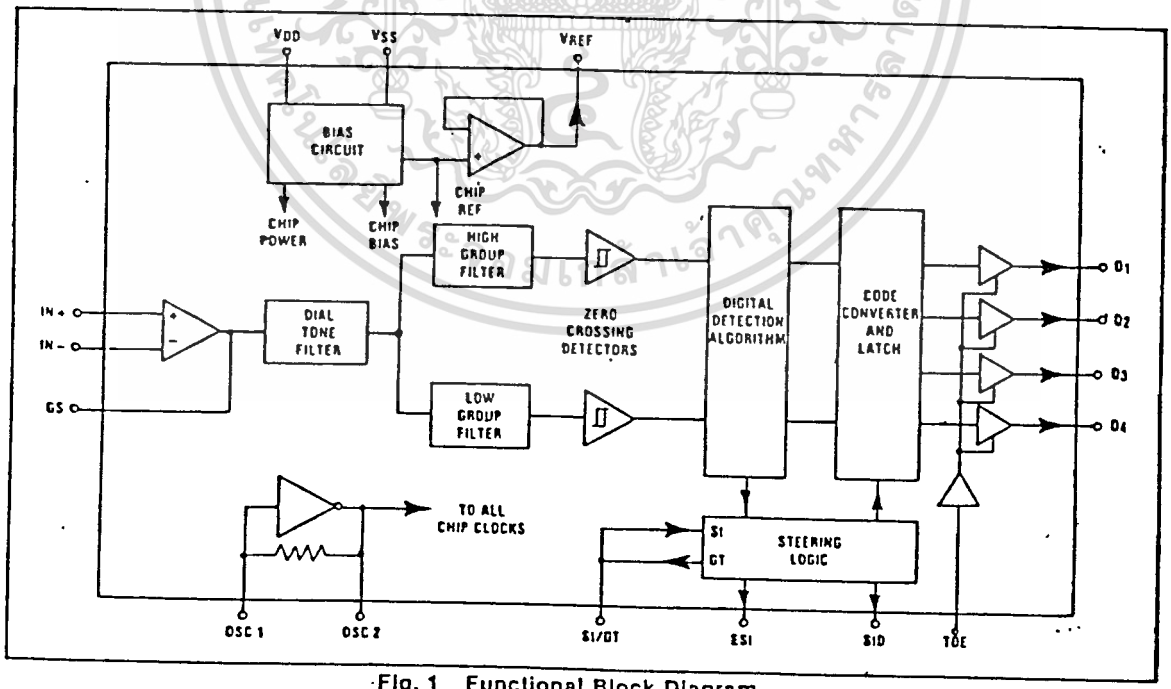


Fig. 1 Functional Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Receiver Performance Characteristics

CHARACTERISTIC	SYMBOL	MIN	TYP	MAX	UNIT	TEST CONDITIONS*
Power Supply Voltage	V_{DD}		5		V	
Operating Power Consumption	P_O		10		mW	
Output High (Source) Current	I_{OH}		1		mA	V_{OUT} = 4.6V
Output Low (Sink) Current	I_{OL}		2		mA	V_{OUT} = 0.4V
Acceptable Signal/Noise Ratio			14		dB	480-3400 Hz
Reference Output Voltage	V_{REF}		2.5		V	Pin 4. No Load.
Steering Threshold Voltage	V_{TSI}		2.5		V	Pin 17
Tone Present Detect Time	t_{DP}		8		ms	
Tone Absent Detect Time	t_{DA}		3		ms	
Twist Accept/Reject	V_H/V_L		± 10		dB	
Frequency Detect Bandwidth	$\Delta f_A, \Delta f_R$	± 2.5		± 3.5	%Nom	
Valid Signal Levels		−30		0	dBm	Measured at GS,Pin 3
Tone Present Guard-Time	t_{GTP}		25		ms	Fig. 2
Tone Absent Guard-Time	t_{GTA}		35		ms	Fig. 2
Valid Input Signal Range		−24		+6	dBm	Fig. 2
Min. Valid Signal Duration	t_{REC}			40	ms	Fig. 2
Max. Spurious Signal Rejected	t_{REC}	20			ms	Fig. 2
Min. Valid Interdigit Pause	t_{ID}			40	ms	Fig. 2
Max. Dropout Tolerated	t_{DO}	20			ms	Fig. 2

*Unless noted T_A = 25°C, V_{DD} = 5V, f_{CLK} = 3.579545 MHz. Voltages referred to V_{SS}.

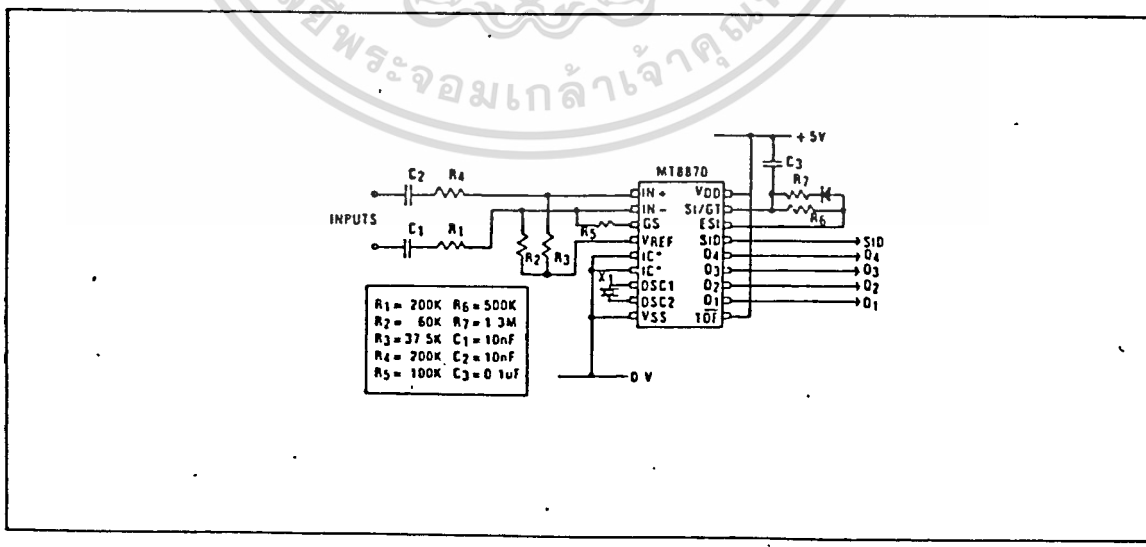


Fig. 2 Typical Receiver Test Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

วิทยานิพนธ์นี้สำเร็จลุล่วงไปด้วยดี เพราะได้รับคำแนะนำ และ ความช่วยเหลือจาก ท่านอาจารย์ที่ปรึกษา รองศาสตราจารย์ ดร. สิทธิชัย โภไคยอดม

ทางผู้จัดทำจึงขอขอบพระคุณต่อ รองศาสตราจารย์ ดร. สิทธิชัย โภไคยอุดม

ขอขอบคุณ รุ่นพี่ ที่บริษัท WAVEGATE ที่ให้คำแนะนำ และให้คำปรึกษา

ขอขอบคุณ บริษัท อินเตอร์ฟาร์อีสท์ วิศวการ ที่กรุณาให้ความช่วยเหลือ ให้ยืมเครื่องมือ และสถานที่ในการทดลอง

ผู้จัดทำ

เอกสารอ้างอิง

1. " MICRO CONTROLLER HANDBOOK ", INTEL COPORATION, 1989

2. " TELECOMMUNICATION DEVICE DATA ", MOTOROLA INC., 1987

3. วิศิษฐ์ สรรพอาษา, "ทางผ่านสัญญาณเสียงแบบดิจิตอลสำหรับชุมสายโทรศัพท์อิเลคทรอนิคส์ สาขาอัตโนมัติขนาดเล็ก", วิทยานิพนธ์สำหรับปริญญาวิศวกรรมศาสตร์มหาบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า ลาดกระบัง, พ.ศ. 2527

4. จิโรจน์ ศรีนามวงศ์, "เครื่องชุมสายโทรศัพท์สาขาอัตโนมัติ แบบเรียกเครื่องลูกได้โดยไม่ต้องผ่านต้องผ่านโอเปอร์เรเตอร์", วิทยานิพนธ์สำหรับปริญญาวิศวกรรมศาสตร์มหาบัณฑิต คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าลาดกระบัง, พ.ศ. 2527

5. ถนอมศักดิ์ บุณยะลีพรรณ, "เครื่องชุมสายโทรศัพท์สาขาอัตโนมัติ ควบคุมด้วยไมโครโปรเซสเซอร์", วิทยานิพนธ์สำหรับปริญญาวิศวกรรมศาสตร์มหาบัณฑิต คณะวิศ