



ปีการศึกษา 2534

โทรศัพท์เต็อนัยอัตโนมัติโดยใช้ EPROM เก็บเสียง

(Automatic EPROM Dialer)

โดย

- | | | |
|----------------|-----------------|---------|
| 1. นาย ลีรี | แซ่เอ็ง | 31.1332 |
| 2. นาย สุนจน | ปกพัฒนกุล | 31.1354 |
| 3. นาย สุรัชย์ | นิทักษ์วงศ์กูชร | 31.1361 |

อาจารย์ที่ปรึกษา

ผศ. พลผดุง ผดุงกุล

008506

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2534

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าคุณทหารลาดกระบัง

เรื่อง โทรศัพท์เตือนภัยอัตโนมัติโดยใช้ EPROM เก็บเสียง

(Automatic EPROM Dialer)

ผู้จัดทำ

1. นาย สิริ แซ่เอ็ง 31.1332
2. นาย สุกจน์ ปกพัฒน์กุล 31.1354
3. นาย สุธัย พัทธพงษ์ศุภธร 31.1361



..... อาจารย์ที่ปรึกษา
(ผศ. พลพัฒน์ พัฒน์กุล)

โทรศัพท์เตือนภัยอัตโนมัติโดยใช้ EPROM เก็บเสียง

AUTOMATIC EPROM DIALER

นาย ลิริ แซ่เอ็ง 31.1332

นาย สุพจน์ ปกพัฒนกุล 31.1354

นาย สุรัชย์ นิตกษพงศ์ภูธร 31.1361

อาจารย์ที่ปรึกษา

ผศ. พลผดุง ผดุงกุล

ปีการศึกษา 2534

บทคัดย่อ

โครงการ "โทรศัพท์เตือนภัยอัตโนมัติโดยใช้ EPROM เก็บเสียง" (AUTOMATIC EPROM DIALER) มาจากแนวคิดที่จะพัฒนาขีดความสามารถของการเตือนภัยไปยังสถานที่หรือหน่วยงานที่ต้องการจะส่งสัญญาณเตือนภัยไปถึงเมื่อเกิดเหตุร้ายขึ้น โดยโครงการนี้จะใช้โทรศัพท์เป็นตัวส่งสัญญาณเสียงของเหตุร้ายต่างๆ ซึ่งข้อความของสัญญาณเสียงนั้นจะถูกอัดเก็บไว้ใน EPROM อยู่ก่อนแล้ว

หลักการทำงานของเครื่องคือ เมื่อเกิดเหตุร้ายขึ้น เครื่องจะทำการยกหูโทรศัพท์หมุนเบอร์ของสถานที่ที่ต้องการส่งข้อความที่อัดไว้ ถ้าสถานที่นั้นสายไม่ว่างหรือไม่มีคนรับสาย เครื่องจะทำการหมุนเบอร์ของสถานที่ใหม่ให้ โดยการทำงานเหล่านี้เครื่องจะจัดการให้โดยอัตโนมัติ ซึ่งจะมีไมโครโปรเซสเซอร์เบอร์ 8031 เป็นตัวควบคุมการทำงานทั้งหมด

AUTOMATIC EPROM DIALER

SIRI	SAEHENG	31.1332
SUPHOT	POKPATTANAKUL	31.1354
SURACHAI	PITAKPONGPUTORN	31.1361

ADVISOR

ASST.PROF POLPHADUNG PHADUNGKUL

SEMESTER 1991

ABSTRACT

Automatic EPROM Dialer project has come from the idea to develop the ability of the emergency call in order to transmit this call to the place or organization that has been expected. This project uses telephone to transmit voice signal that has been recorded in EPROM before.

When the emergency occurs, this project will off-hook the telephone and dial to the expected place then transmit the message. If that place cannot be contacted, it will contact the other place automatically. All functions are controlled by microprocessor 8031.

สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎี	2
2.1 เดลต้ามอดคูล์เลชั่น	2
2.2 ระบบสัญญาณต่างๆจากขมสายโทรศัพท์	8
2.3 EPROM เบอร์ 27010	9
2.4 ไมโครโปรเซสเซอร์ 8031	11
บทที่ 3 หลักการทำงาน	31
3.1 วงจรส่วนบันทึกลง EPROM	31
3.2 วงจรเตือนภัยทางโทรศัพท์อัตโนมัติ	33
บทที่ 4 รายละเอียดการออกแบบและการทำงาน	36
4.1 วงจรส่วนบันทึกลง EPROM	36
4.2 วงจรเตือนภัยทางโทรศัพท์อัตโนมัติ	46
บทที่ 5 การทดลองและผลการทดลอง	53
บทที่ 6 บทสรุปและวิจารณ์	54
เอกสารอ้างอิง	
กิตติกรรมประกาศ	
ภาคผนวก	

บทที่ 1

บทนำ

การใช้งานโทรศัพท์ที่เป็นที่แพร่หลายกันอย่างมากในปัจจุบัน ทั้งยังสามารถนำไปประยุกต์ใช้งานต่าง ๆ ได้อีก เช่น การแจ้งเหตุร้ายต่าง ๆ ซึ่งโดยปกติมนุษย์จะเป็นผู้ติดต่อแล้วส่งสัญญาณเสียงไปเองเมื่อเกิดเหตุร้ายขึ้น แต่เนื่องมาจากความก้าวหน้าทางเทคโนโลยี มนุษย์ไม่จำเป็นต้องทำเองทั้งหมด อาจมีอุปกรณ์ที่ช่วยในการทำงานบางส่วนหรือทั้งหมดได้

ในโครงงานนี้ จะเป็นการนำเอา EPROM มาช่วยในการเก็บสัญญาณเสียงและอื่น ๆ เช่น เบอร์โทรศัพท์ โปรแกรมควบคุมการทำงาน เป็นต้น อีกทั้งมีส่วนที่ทำหน้าที่ติดต่อส่งสัญญาณเสียงไปตามสายโทรศัพท์ ไปยังสถานที่ที่กำหนดไว้ก่อนแล้วเมื่อเกิดเหตุร้ายขึ้น โดยเครื่องสามารถหมุนเบอร์ และส่งสัญญาณเสียง (เสียงพูด) ให้โดยอัตโนมัติ อีกทั้งยังสามารถเปลี่ยนเบอร์ไปยังสถานที่ใหม่ได้ โดยอัตโนมัติเมื่อสถานที่แรกติดต่อไม่ได้ ซึ่งโครงงานนี้ สามารถนำไปประยุกต์ใช้งานอย่างอื่นได้อีก เช่น การส่งสัญญาณเสียงที่ได้บันทึกไว้ก่อนแล้วไปตามสายโทรศัพท์ ไปยังสถานที่ต่าง ๆ ตามเวลาที่กำหนดไว้ เป็นต้น

สุดท้ายนี้คณะผู้จัดทำหวังว่า โครงงานนี้ จะเป็นประโยชน์แก่ผู้ที่สนใจบ้างไม่มากก็น้อย

คณะผู้จัดทำ...



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎี

ทฤษฎีพื้นฐานที่จะถูกนำมาใช้ในการอธิบายการทำงานของโครงงานนี้สามารถแยกการ อธิบายออกเป็นเรื่องต่างๆ ซึ่งจะอธิบายเฉพาะเรื่องที่จำเป็น ได้ดังต่อไปนี้

2.1 เอลตามอดคูละชัน

การมอดคูละชันโดยทั่วไป สามารถแบ่งออกได้เป็น 2 ประเภทใหญ่ๆ ได้แก่

1. การมอดคูละชันเชิงอนาลอก (Analog Modulation) คือการมอดคูละชันสัญญาณอินพุต (Input Signal) แล้วได้สัญญาณเอาต์พุต (Output Signal) ออกมาในลักษณะที่ขนาดของสัญญาณ (Amplitude) มีความต่อเนื่องกับเวลา → Analog
2. การมอดคูละชันเชิงดิจิตอล (Digital Modulation) คือการมอดคูละชันสัญญาณอินพุตแล้วได้สัญญาณเอาต์พุตออกมาในลักษณะที่ขนาดของสัญญาณไม่มีความต่อเนื่องกับเวลา ซึ่งสัญญาณเอาต์พุตในลักษณะนี้ คือ สัญญาณดิจิตอล (Digital Signal)

การมอดคูละชันเชิงดิจิตอล มีหลายแบบ แต่ประเภทที่เป็นพื้นฐาน และนิยมใช้งานมากมี 2 ประเภท ได้แก่

- เอลตามอดคูละชัน
- การมอดคูละชันแบบรหัสพัลส์ (Pulse Code Modulation)

สำหรับในที่นี้จะอธิบายเฉพาะส่วนของ เอลตามอดคูละชัน เนื่องจากในโครงงานนี้ได้มีการนำเฉพาะเอลตามอดคูละชันมาใช้งานเท่านั้น

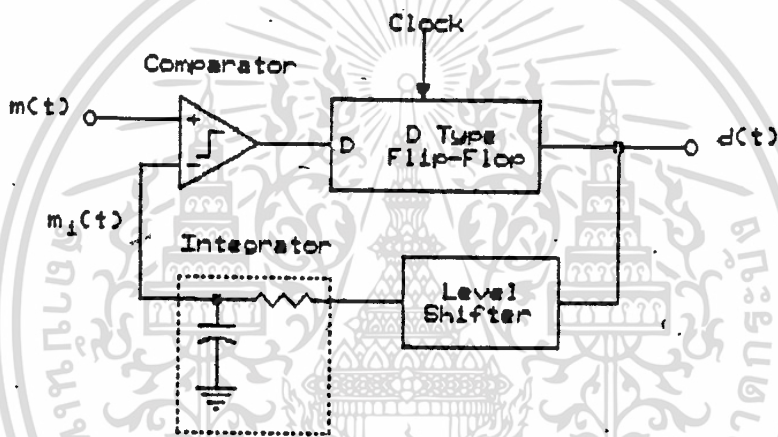
เพื่อให้สามารถเข้าใจได้ง่าย จะขออธิบายเป็นขั้นตอน โดยแยกอธิบายออกเป็น 2 ส่วนดังนี้

2.1.1 เอลตามอดคูละชันอย่างง่าย (Simple Delta Modulation)

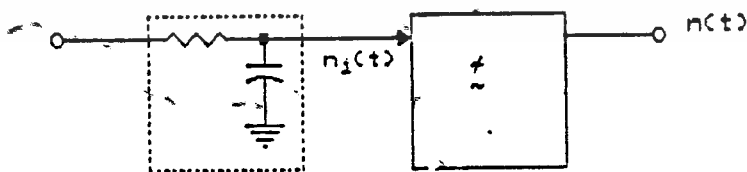
เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 เคลตามอดดูเลชั่นอย่างง่าย

เคลตามอดดูเลชั่น จะทำการมอดดูเลทสัญญาณอินพุตออกมาเป็นสัญญาณพัลส์ที่เป็นเลขฐานสองซึ่งได้มาจากการเปรียบเทียบขนาดของสัญญาณอินพุตปัจจุบันกับสัญญาณอินพุตในอดีตที่ถูกป้อนกลับมา โดยสัญญาณอินพุตจะต้องถูกแซมปลิง (Sampling) ด้วยความถี่ของสัญญาณนาฬิกา (Clock) และการทำงานทั้งหมดก็จะต้องอ้างอิงถึงสัญญาณนาฬิกาด้วย ซึ่งแผนผังการทำงาน (Block Diagram) ของเคลตามอดดูเลชั่น สามารถแสดงได้ดังรูปที่ 2.1



ก) วงจรเข้ารหัส (Encoder)

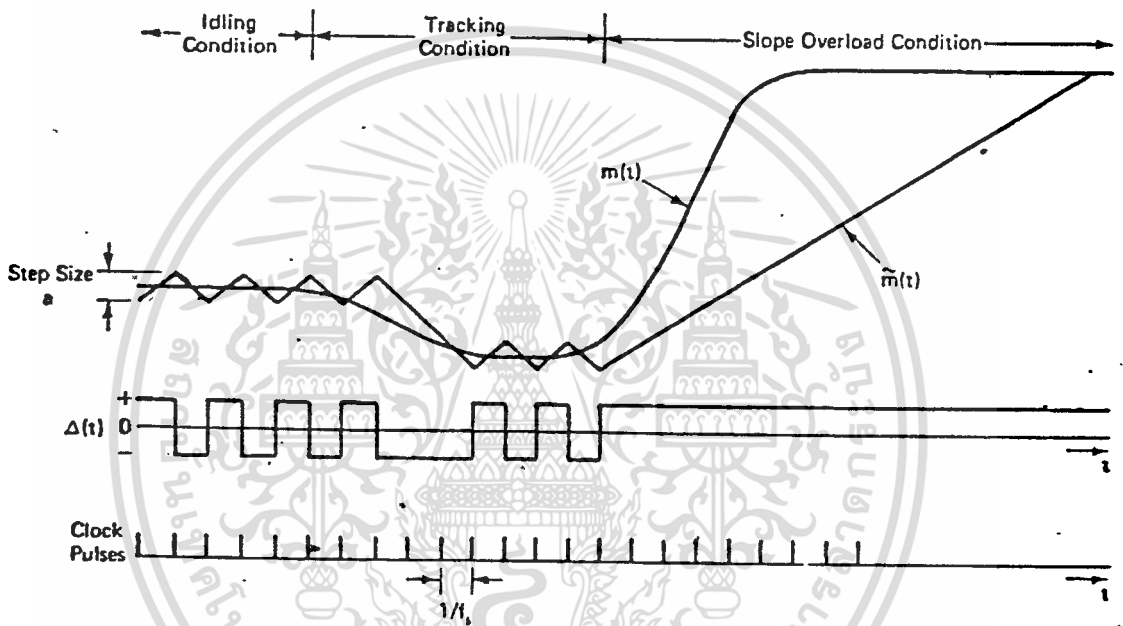


ข) วงจรถอดรหัส (Decoder)

รูปที่ 2.1 แผนผังการทำงานของเคลตามอดดูเลชั่น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.1 สัญญาณอินพุต $m(t)$ จะถูกนำมาเปรียบเทียบกับสัญญาณอินทิเกรตที่ถูกป้อนกลับ $m_1(t)$ ซึ่งถ้า $m(t)$ มีขนาดของสัญญาณสูงกว่า $m_1(t)$ สัญญาณเอาต์พุต $d(t)$ จะเป็นสัญญาณพัลส์ทางด้านบวก (Positive Pulse) แต่ถ้า $m(t)$ มีขนาดของสัญญาณต่ำกว่า $m_1(t)$ สัญญาณเอาต์พุตที่ได้ก็จะเป็นสัญญาณพัลส์ทางด้านลบ (Negative Pulse) ซึ่งสัญญาณเอาต์พุตที่ได้ไม่ว่าจะเป็นกรณีใดก็ตาม จะถูกนำผ่านเข้าวงจรสร้างระดับสัญญาณ (Level Shifter) เพื่อนำเข้าวงจรอินทิเกรเตอร์ (Integrator) แล้วถูกป้อนกลับมาเป็นสัญญาณ $m_1(t)$ เพื่อใช้เป็นสัญญาณเปรียบเทียบกับสัญญาณอินพุตที่สัญญาณนาฬิกาถัดไป



รูปที่ 2.2 แสดงรูปคลื่นสัญญาณสำหรับเดลตามอดูเลชัน

จากรูปที่ 2.2 สัญญาณ $m_1(t)$ จะมีลักษณะเป็นสัญญาณแบบสามเหลี่ยม (Triangle Signal) ซึ่งลักษณะของสัญญาณ $m_1(t)$ จะแบ่งออกเป็น 3 สภาวะ ได้แก่

1. สภาวะไอเดิล (Idling Condition) คือ สภาวะที่ไม่มีสัญญาณอินพุตผ่านเข้าวงจรเดลตามอดูเลเตอร์ (Delta Modulator) ซึ่งจะทำให้วงจรอินทิเกรเตอร์สร้างสัญญาณแรมพ์ (Ramp) ซึ่งในสภาวะนี้ สัญญาณเอาต์พุต $d(t)$ ที่ได้ จะประกอบด้วยสัญญาณพัลส์ทางด้านบวกและทางด้านลบสลับกันไป

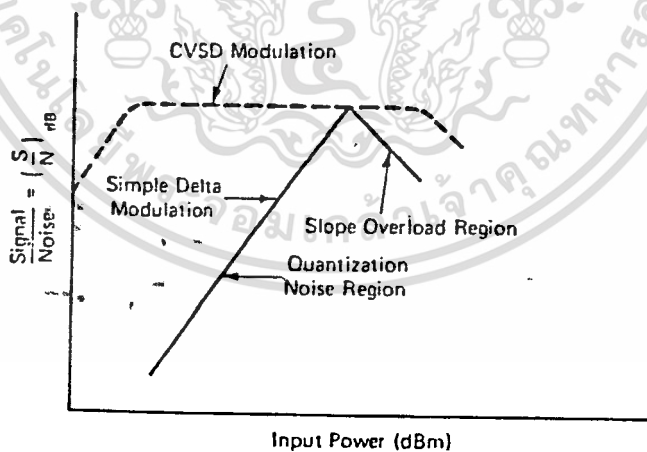
2. สภาวะแทรค (Tracking Condition) คือ สภาวะปกติที่มีสัญญาณอินพุตเข้ามาอย่างต่อเนื่อง ซึ่งก็จะได้สัญญาณเอาต์พุตตามลักษณะของสัญญาณอินพุตที่ถูกมอดูเลตตามที่ได้อธิบายมาแล้ว ซึ่งส่วนนี้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งสองสภาวะแรกนั้น ความแตกต่างของสัญญาณอินพุตและสัญญาณป้อนกลับ จะก่อให้เกิดความผิดพลาด (Error) ของสัญญาณขึ้น เราจะเรียกปรากฏการณ์เช่นนี้ว่า การเกิดสัญญาณรบกวนแบบแกรนูลาร์หรือแบบควอนไทซ์ (Granular or Quantization Noise) ซึ่งสัญญาณรบกวนนี้สามารถลดลงได้ด้วยทั้งการลดค่าของระดับ "a" และการเพิ่มอัตราการสุ่ม (Sampling Rate)

3. สภาวะความชันโอเวอร์โหลด (Slope Overload Condition) คือ สภาวะที่สัญญาณอินพุตมีขนาดที่เปลี่ยนแปลงอย่างรวดเร็ว หรือ มีความชันของสัญญาณมากกว่าสัญญาณป้อนกลับมีความชันไม่ทัน ลักษณะเช่นนี้จะทำให้สัญญาณเอาท์พุทเกิดความผิดพลาด เนื่องจากระดับของสัญญาณที่เกิดขึ้นในแต่ละครั้งของ $m_s(t)$ มีค่าคงที่ สภาวะนี้จะทำให้สัญญาณเกิดความเพี้ยน (Distortion) ขึ้น ซึ่งนี่ก็คือข้อจำกัดของเดลต้ามอดดูเลชัน ที่ไม่สามารถทำการมอดดูเลชันสัญญาณอินพุตที่มีการเปลี่ยนแปลงขนาดอย่างรวดเร็ว รวมทั้งสัญญาณที่มีความถี่สูง (High Frequency Signal)

เมื่อมีสภาวะความชันโอเวอร์โหลดเกิดขึ้น เราเรียกปรากฏการณ์เช่นนี้ว่า การเกิดสัญญาณรบกวนแบบความชันโอเวอร์โหลด (Slope Overload Noise) ซึ่งจะมีผลทำให้สัญญาณรบกวนแบบควอนไทซ์เพิ่มขึ้น



รูปที่ 2.3 แสดงอัตราส่วน S/N ของเดลต้ามอดดูเลชัน

จากรูป 2.3 จะเห็นว่าอัตราส่วน S/N (S/N Ratio) จะมีค่าเพิ่มขึ้นตามกำลัง (Power) ของสัญญาณอินพุต จนกระทั่งสภาวะความชันโอเวอร์โหลดเกิดขึ้น ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้าให้สัญญาณอินพุท

$$m(t) = A \sin 2\pi f_m t \quad \dots(2.1)$$

อัตราการเพิ่มสูงสุดของสัญญาณอินพุท

$$dm(t)/dt_{max} = 2\pi f_m A \quad \dots(2.2)$$

และ อัตราการเพิ่มของสัญญาณป้อนกลับ $m_r(t)$

$$r(t) = a/(1/f_m) = af_m \quad \dots(2.3)$$

เมื่อ a คือ ขนาดของระดับการเปลี่ยนแปลงของสัญญาณป้อนกลับ

f_m คือ ความถี่อัตราการลุ่ม

ซึ่งเราต้องการให้สมการ (2.2) และ (2.3) มีค่าเท่ากัน ดังนั้นจะได้ว่า

$$\begin{aligned} 2\pi f_m A &= af_m \\ f_m A &= af_m / 2\pi \end{aligned} \quad \dots(2.4)$$

จะเห็นได้ว่า การลดสัญญาณรบกวนแบบความชันโอเวอร์โหลด สามารถทำได้โดยการเพิ่มขนาดของระดับการเปลี่ยนแปลงของสัญญาณป้อนกลับ และ การเพิ่มอัตราการลุ่ม

2.1.2 คอมแพคต์เดลต้ามอดูเลชัน

จากรูปที่ 2.3 ซึ่งแสดงอัตราส่วน S/N ของเดลต้ามอดูเลชันอย่างง่ายนั้น มีค่าไม่คงที่ดังนั้นการที่จะทำให้อัตราส่วน S/N มีค่าคงที่นั้น จำเป็นต้องทำให้ค่าของระดับ "a" สามารถปรับได้ ซึ่งเมื่อสัญญาณอินพุทมีขนาดเล็ก ขนาดของ a ก็จะมีขนาดเล็ก เมื่อสัญญาณอินพุทมีขนาดเพิ่มขึ้น ขนาดของ a ก็จะต้องเพิ่มขึ้น

สำหรับการใช้งานทางด้านโทรศัทพ์ เดลต้ามอดูเลชันที่ใช้ จะต้องสามารถปรับได้ ซึ่งเราใช้แบบที่เรียกว่า เดลต้ามอดูเลชันแบบความชันแปรเปลี่ยนต่อเนื่อง (Continuous Variable Slope Delta Modulation (CVSD)) ซึ่งสามารถเขียนแผนผังการทำงานได้ดังนี้

2.2 ระบบสัญญาณต่างๆจากชุมสายโทรศัพท์

ลักษณะของสัญญาณต่างๆจากชุมสายโทรศัพท์ที่มีลักษณะดังต่อไปนี้

1. สัญญาณให้หมุน (dial tone) คือ สัญญาณที่ชุมสายโทรศัพท์แจ้งให้ผู้ใช้งานโทรศัพท์ทราบว่าเครื่องชุมสายโทรศัพท์พร้อมแล้วที่จะให้ผู้ใช้งานโทรศัพท์กดหรือหมุนหมายเลขที่ต้องการจะติดต่อด้วย .

ลักษณะของสัญญาณ เป็นสัญญาณที่มีความถี่ประมาณ 400 Hz ดังต่อเนื่องกันตลอด

2. สัญญาณไม่ว่าง (busy tone) คือ สัญญาณที่ชุมสายโทรศัพท์แจ้งให้ผู้ใช้งานโทรศัพท์หลังจากกดหรือหมุนหมายเลขที่ต้องการจะติดต่อด้วย ให้ทราบว่าไม่สามารถติดต่อกับคู่สายโทรศัพท์นั้นได้

ลักษณะของสัญญาณ เป็นสัญญาณที่มีความถี่ประมาณ 400 Hz ดัง และ ดับ ทุกๆ 0.5 วินาที

3. สัญญาณเรียกกลับ (ringback tone) คือ สัญญาณที่ชุมสายโทรศัพท์แจ้งให้ผู้ใช้งานโทรศัพท์หลังจากกดหรือหมุนหมายเลขที่ต้องการจะติดต่อด้วย ให้ทราบว่าสามารถติดต่อกับคู่สายโทรศัพท์ตามที่ต้องการติดต่อนั้นได้

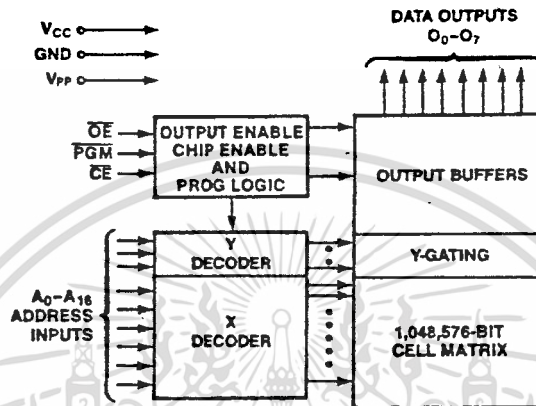
ลักษณะของสัญญาณ เป็นสัญญาณที่มีความถี่ประมาณ 400 Hz ดัง 1 วินาที และดับ 3 วินาทีสลับกัน

4. สัญญาณเรียก (ringing tone) คือ สัญญาณที่ชุมสายโทรศัพท์ส่งไปยังภายในเครื่องโทรศัพท์ ทำให้กระดิ่งดัง เพื่อแจ้งให้ทราบว่า มีผู้ต้องการจะติดต่อด้วย

ลักษณะของสัญญาณ เป็นสัญญาณกระแสสลับ (AC) 50 Hz ที่มีขนาดประมาณ 48 โวลต์ ดังและดับเป็นช่วง ๆ และพร้อมกับสัญญาณเรียกกลับ

2.3 EPROM เบอร์ 27010 (128K x 8 1E-WIDE EPROM)

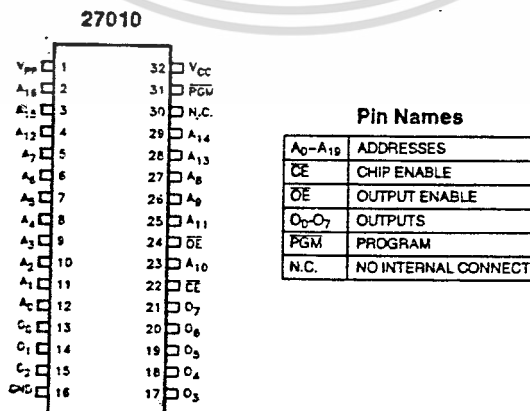
EPROM (เบอร์ 27010) เป็น EPROM ที่มีความจุของหน่วยความจำทั้งหมด 1M บิต โดยมีลักษณะการทำงานเป็นบล็อกไดอะแกรม (block diagram) ดังจะสามารถแสดงได้ดังรูปที่ 2.5



รูปที่ 2.5 บล็อกไดอะแกรมของ 27010

2.3.1 ขาสัญญาณต่างๆของ EPROM เบอร์ 27010

สำหรับแผนผังขาสัญญาณต่างๆ ของ EPROM (27010) สามารถแสดงได้ดังรูป 2.6



จากรูปข้างต้น ขาสัญญาณแต่ละขามีหน้าที่ดังนี้

$A_0 - A_{16}$	- เป็นสัญญาณสำหรับอ้างแอดเดรสของข้อมูล
$\overline{CE}$	- (CHIP ENABLE INPUT) เป็นขาอินพุตสำหรับใช้ในการเลือกให้อิซี 27010 อยู่ในสภาวะทำงาน
$\overline{OE}$	- (OUTPUT ENABLE) เป็นขาอินพุตเพื่อใช้ควบคุมการอ่านข้อมูลซึ่งส่งมาจาก CPU
$O_0 - O_7$	- ใช้สำหรับเป็นขาข้อมูลในการรับส่งกับ CPU
V_{PP}	- ใช้สำหรับในการโปรแกรม ($V_{PP} = 12.75 \text{ v}$)
V_{CC}	- เป็นขาไฟเลี้ยงของวงจรและใช้ในการโปรแกรม ($V_{CC} = 6.25 \text{ v}$)
$\overline{PGM}$	- เป็นขาอินพุตใช้สำหรับควบคุมการเขียนข้อมูล

2.3.2 การทำงานในโหมดต่างๆของ EPROM เบอร์ 27010

โหมดการทำงานของ EPROM (27010) จะแสดงดังในตารางที่ 2.1

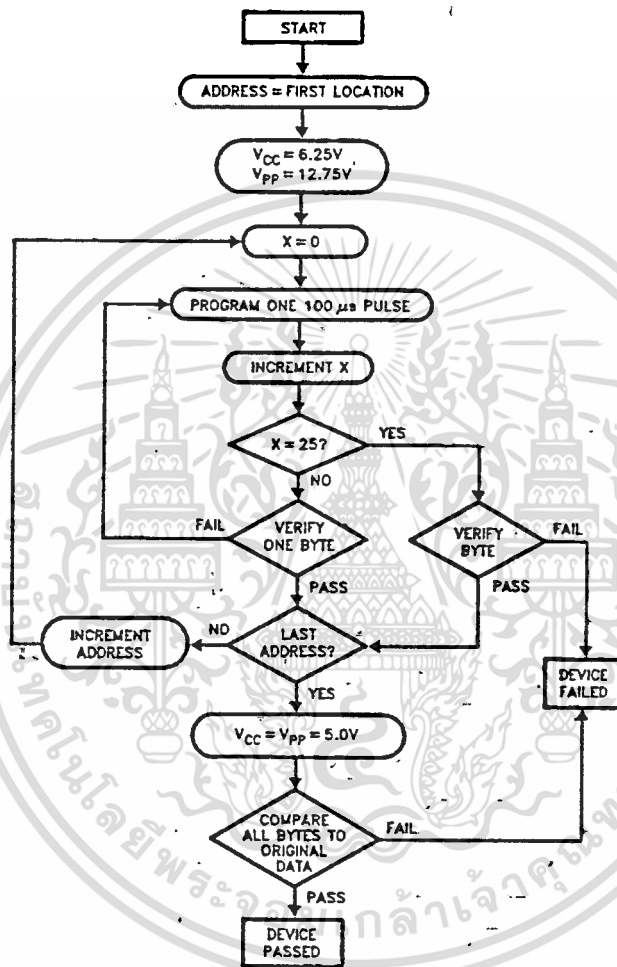
MODE	PIN	$\overline{CE}$	$\overline{OE}$	$\overline{PGM/WE}$	V_{PP}	V_{CC}	O/P
READ		V_{IL}	V_{IL}	X ⁽¹⁾	X	5.0V	D_{OUT}
PROGRAMING		V_{IL}	V_{IH}	V_{IL}	$V_{PP}^{(2)}$	$V_{CC}^{(3)}$	HIGH Z
* หมายเหตุ 1) X สามารถเป็นได้ทั้ง V_{IL} หรือ V_{IH} 2) $V_{PP} = 12.75 \text{ V}$ 3) $V_{CC} = 6.25 \text{ V}$							

ตารางที่ 2.1 โหมดการทำงานของ EPROM เบอร์ 27010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.3 การโปรแกรม EPROM เบอร์ 27010

การโปรแกรม EPROM เบอร์ 27010 ของอินเทล จะใช้อัลกอริทึมแบบควิกพัลส์ โปรแกรมมิ่งอัลกอริทึม (Quick-pulse Programming Algorithm) ซึ่งแสดงเป็นผังการทำงานได้ ดังในรูปที่ 2.7



รูปที่ 2.7 ผังการทำงาน การโปรแกรม EPROM เบอร์ 27010

2.4 ไมโครโปรเซสเซอร์ 8031

8031 ไมโครคอนโทรลเลอร์ เป็นไมโครโปรเซสเซอร์ในตระกูล MCS-51 ซึ่งไม่มีหน่วยความจำชนิดรอม (ROM) อยู่ในตัว ลักษณะของ 8031 เป็นดังนี้

- ใช้ HMOS เทคโนโลยีในการสร้าง การทำงานจะใช้แหล่งจ่ายไฟขนาด 5 โวลต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีวงจรออสซิลเลเตอร์ (Oscillator) และวงจรนาฬิกา (Clock) ในตัว
- ชุดแบงก์ (Bank) รีจิสเตอร์มี 4 ชุด โดยมีชุดละ 8 รีจิสเตอร์
- มีตัวตั้งเวลาและตัวนับ (Timer/Counter) 2 ตัว ขนาด 16 บิต
- มีพอร์ทอินพุตเอาต์พุตขนานแบบ 2 ทิศทาง พอร์ทละ 8 บิต จำนวน 16 เส้น และอีก 16 เส้นใช้ในการเข้าถึงทางแอดเดรส (Address) และข้อมูล (Data)
- พอร์ททางอนุกรมสามารถที่จะโปรแกรมการรับส่งแบบฟูลดูเพลกซ์ ที่ความเร็วสูง
- หนึ่งวัฏจักรคำสั่งจะใช้เวลา 1 ไมโครวินาที ด้วยการใช้คริสตัลความถี่ 12 MHz
- แอดเดรสข้อมูลภายนอกได้ 64 กิโลไบต์
- แอดเดรสโปรแกรมภายนอกได้ 64 กิโลไบต์
- สามารถกำหนดที่อยู่ข้อมูลขนาดไบต์ หรือบิตได้โดยตรง
- มีซอฟต์แวร์แฟล็ก (Flag Software) สำหรับผู้ใช้ที่จะกำหนดเองได้ถึง 128

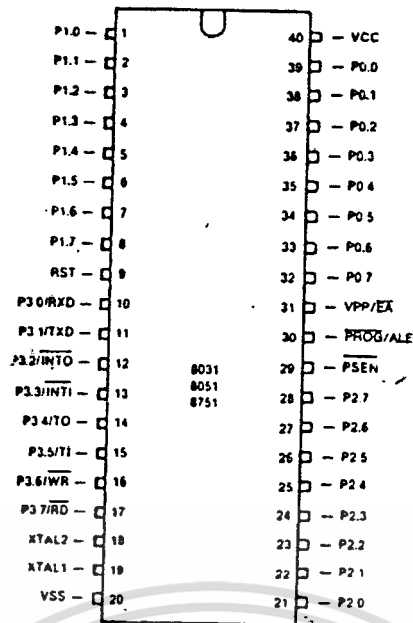
ตำแหน่งบิต

- โครงสร้างอินเทอร์รัพท์ (Interrupt) ทำได้ 5 แหล่ง พร้อมด้วยการจัดไพริอิตี (Priority) ได้ 2 ระดับ
- ตัวโปรเซสเซอร์สามารถใช้งานแบบบูลีน (Boolean) ได้ สำหรับการใช้อย่างงาน

ควบคุม

- มีคำสั่งคูณ และหารทางฮาร์ดแวร์ทำได้ภายใน 4 ไมโครวินาที (สำหรับคริสตัลความถี่ 12 MHz)
- ตัวเลขทางคณิตศาสตร์ ใช้ได้ทั้งแบบไบนารี และเดซิมีออล
- การใช้พื้นที่สแต็ก (Stack) สำหรับโปรแกรมย่อยต่างๆ ทำได้กว้างขึ้น
- ชุดคำสั่งของ MCS-51 มีถึง 256 คำสั่ง

ลักษณะการจัดขาของ 8031



รูปที่ 2.8 ลักษณะภายนอกของ 8031

จากรูป เป็นการจัดขาตามลักษณะภายนอกของ 8031 ซึ่งมีรายละเอียดดังนี้

- V_{SS} เป็นขาสําหรับต่อลงกราวด์
- V_{CC} ใช้แรงดันไฟตรงขนาด 5 โวลต์ และใช้ในการโปรแกรม
- PORT 0 (P0.0-P0.7 / AD0-AD7) เป็นอินพุตเอาต์พุตพอร์ท 8 บิต แบบโอเพนเดรนไบไดเรกชันแนล(Open Drain Bidirectional) สามารถที่จะรับโหลด TTL ได้ 8 ตัว การเขียนค่า "1" ไปที่พอร์ทนี้จะทำให้พอร์ทนี้อยู่ในสภาวะลอย(Float) ทำให้มันทำงานเป็นอินพุต มีสถานะเป็นอิมพีแดนซ์สูง ในการให้พอร์ทนี้บริการแบบอินพุตเอาต์พุตพอร์ท 0 จะทำงานเป็นตัวมัลติเพลกซ์ ด้วยสัญญาณแอดเดรสไบต์ค่ากับบัสข้อมูล ในการใช้งานด้านหน่วยความจำภายนอก จะใช้ลักษณะภายในเป็นตัวพูลอัพ พอร์ท 0 ยังใช้งานเป็นตัวส่งข้อมูลออกทางพอร์ทนี้

- PORT 1 (P1.0-P1.7) เป็นพอร์ทอินพุตเอาต์พุต 8 บิต แบบโอเพนเดรนไบไดเรกชันแนล(Open Drain Bidirectional) พร้อมด้วยการพูลอัพภายในถ้าเป็นพอร์ทเอาต์พุต บัฟเฟอร์สามารถรับโหลด TTL ตระกูล LS ได้ 4 ตัว พอร์ท 1 เมื่อถูกเขียนค่า "1" ด้วยโปรแกรม มันจะมีสถานะสูงด้วยการพูลอัพภายใน การให้สถานะเช่นนี้จะเป็นการทำให้พอร์ทนี้เป็นอินพุต ขณะที่พอร์ทนี้เป็นอินพุต การให้สัญญาณ "0" จะเป็นการจ่ายกระแสออกเนื่องจากการพูลอัพภายใน

- PORT 2 (P2.0-P2.7) เป็นพอร์ทอินพุตเอาต์พุต 8 บิต แบบโอเพนเดรนไบไดเรกชันแนล(Open Drain Bidirectional) พร้อมด้วยการพูลอัพภายใน พอร์ท 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ทำหน้าที่เป็นบัฟเฟอร์เอาต์พุต สามารถจ่ายโหลด TTL ตระกูล LS ได้ 4 ตัว พอร์ตนี้ จะถูกใช้งานเป็นตัวส่งแอดเดรสไบต์สูงด้วยเมื่อใช้งานร่วมกับหน่วยความจำภายนอก เพื่อให้แอดเดรสได้ถึง 16 บิต ด้วยการใช้งานแบบนี้ มันจะมีผลอ้อมภายในที่ช่วยให้การส่งค่า "1" ได้ระดับที่แน่นอน

- PORT 3 (P3.0-P3.7) เป็นพอร์ตอินพุตเอาต์พุต 8 บิต แบบพุลอัปภายใน นอกจากทำเป็นพอร์ตอินพุตเอาต์พุตที่สามารถรับโหลด TTL ตระกูล LS ได้ 4 ตัวแล้ว ยังใช้งานเป็นพิเศษสำหรับตระกูล MCS-51 คือ

ขาพอร์ต	การทำงานตามฟังก์ชันพิเศษ
P3.0	RxD พอร์ตอนุกรมอินพุต
P3.1	TxD พอร์ตอนุกรมเอาต์พุต
P3.2	$\overline{\text{INT0}}$ อินเทอร์รัพท์ภายนอกตัวที่ 1
P3.3	$\overline{\text{INT1}}$ อินเทอร์รัพท์ภายนอกตัวที่ 2
P3.4	TO สัญญาณกระตุ้นเข้าที่ตัวตั้งเวลาและตัวนับ 0 (Timer/Counter 0)
P3.5	T1 สัญญาณกระตุ้นเข้าที่ตัวตั้งเวลาและตัวนับ 1 (Timer/Counter 1)
P3.6	$\overline{\text{WR}}$ สัญญาณควบคุมการเขียน
P3.7	$\overline{\text{RD}}$ สัญญาณควบคุมการอ่าน

การทำงานตามฟังก์ชันดังกล่าว จะต้องเริ่มด้วยการส่งค่า "1" ไปแลตช์ (Latch) ที่ บิทรีจิสเตอร์ต่างๆก่อน

- RST ต้องคงสถานะ "1" เป็นเวลาอย่างน้อย 2 วงจรระหว่างที่ออสซิลเลเตอร์ ทำงานขณะที่ต้องรีเซ็ตทั้งระบบ โดยจะต่อความต้านทานพูลดาวน์ (8.2 กิโลโอห์ม) จากขา RST ลงกราวด์ และเพื่อให้ตัวชิพรีเซ็ตได้โดยอัตโนมัติขณะเปิดไฟ จะใช้ตัวเก็บประจุ (10 ไมโครฟารัด) ต่อคร่อมระหว่างขา RST กับ ขา V_{cc}

- $\text{ALE}/\overline{\text{PROG}}$ เป็นขาแอดเดรสแลตช์เอินเบิล (Address Latch Enable) ด้วยการส่งพัลส์ออกไปใช้สำหรับแลตช์ค่าแอดเดรสไบต์ต่ำจากพอร์ต 0 ในระหว่างการเข้าถึงข้อมูลจากหน่วยความจำภายนอก ALE จะถูกส่งสัญญาณนาฬิกาออกมา ในอัตราเร็วคงที่ ที่ 1/8 ของความถี่ออสซิลเลเตอร์ตลอดเวลาแม้ว่าจะไม่มีการเข้าถึงข้อมูลจากภายนอก ดังนั้นจึงสามารถที่จะใช้สัญญาณจากขานี้เป็นตัวตั้งเวลาภายนอกหรือเป็นความถี่สัญญาณนาฬิกาได้ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่อย่างไรก็ตามความถี่สัญญาณนี้จะลดความถี่ช้าลงไปเท่าหนึ่งระหว่างการทำงานแบบการเข้าถึงของหน่วยความจำข้อมูล

- $\overline{PSEN}$ เป็นลิตรอ่านข้อมูลจากโปรแกรมหน่วยความจำภายนอก เมื่อชิพทำงานด้วยโปรแกรมภายนอก ขา $\overline{PSEN}$ จะสร้างลิตรต่ำ 2 ครั้งในแต่ละวัฏจักรแมทชีน (Machine Cycle) สัญญาณจะมีสถานะสูงหรือพัลส์ต่ำทั้งสองรูปจะหายไปเมื่อทำงานในช่วงการอ่านหรือเขียนข้อมูลจากหน่วยความจำข้อมูลภายนอก และ $\overline{PSEN}$ จะไม่มีพัลส์ส่งออกมาถ้าชิพทำงานด้วยโปรแกรมหน่วยความจำภายใน

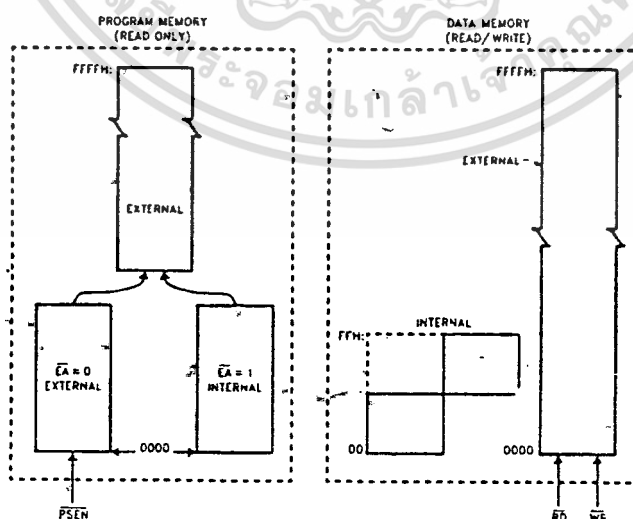
- $\overline{EA}/V_{PP}$ ถ้ามีสถานะสูง ตัว CPU ในชิพจะทำงานตามโปรแกรมที่อยู่ในหน่วยความจำภายใน (โดยที่โปรแกรมจะต้องไม่ยาวกว่า 4 กิโลไบต์) การทำให้ $\overline{EA}$ มีสถานะต่ำจะเป็นการควบคุมให้ CPU ทำงานตามโปรแกรมหน่วยความจำภายนอก ซึ่งขยายโปรแกรมได้ยาวถึง 64 กิโลไบต์ ขา $\overline{EA}$ จะต้องต่อลงกราวด์เช่นกันแม้ว่าจะไม่มี ROM อยู่ภายในก็ตาม

- XTAL 1 ใช้เป็นอินพุตเข้าสู่ตัวออสซิลเลเตอร์ขยายแบบอินเวอร์ท
- XTAL 2 ใช้เป็นเอาต์พุตเข้าสู่ตัวออสซิลเลเตอร์ขยายแบบอินเวอร์ท

2.4.1 การจัดหน่วยความจำของ MCS-51

มีการแบ่งหน่วยความจำออกเป็น 2 ส่วนแยกจากกันคือ หน่วยความจำโปรแกรม (Program memory) และ หน่วยความจำข้อมูล (Data memory)

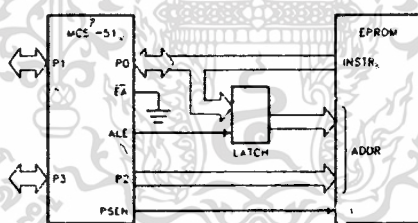
ดังรูปที่ 2.9



รูปที่ 2.9 การจัดหน่วยความจำของ MCS-51

หน่วยความจำโปรแกรม (Program memory) ใช้สำหรับเก็บโปรแกรม สามารถอ่านได้เพียงอย่างเดียว เขียนข้อมูลลงไปไม่ได้ มีขนาดได้ถึง 64 KByte หน่วยความจำโปรแกรม (Program memory) สามารถเป็นได้ทั้งหน่วยความจำ ROM ภายในและหน่วยความจำ ROM ภายนอกที่จะนำมาเพิ่มเติม หรือจะใช้แต่หน่วยความจำ ROM ภายนอกอย่างเดียวก็ได้ (8031 จะใช้แต่หน่วยความจำภายนอก) การเลือกจะใช้หน่วยความจำแบบไหนทำได้โดยการให้สัญญาณแก่ขา $\overline{EA}$ ของ MCS51 โดยถ้าใช้หน่วยความจำภายในให้ดึงขา $\overline{EA}$ ขึ้น Vcc แอดเดรสของหน่วยความจำภายในจะเริ่มจาก 0000 และแอดเดรสของหน่วยความจำที่นำมาเพิ่มจะมีแอดเดรสต่อจากหน่วยความจำภายใน แต่ถ้าใช้แต่หน่วยความจำภายนอกเพียงอย่างเดียวให้ดึงขาสัญญาณ $\overline{EA}$ ลงกราวด์ (ซึ่งใน 8031 จะต่อขา $\overline{EA}$ ลงกราวด์) แอดเดรสของหน่วยความจำภายนอกก็จะเริ่มจาก 0000H เป็นต้นไป การอ่านข้อมูลจากหน่วยความจำภายนอก MCS51 จะส่งสัญญาณ $\overline{PSEN}$ (Program Store Enable) ออกไป

ฮาร์ดแวร์สำหรับการเข้าถึงหน่วยความจำภายนอกแสดงดังรูปที่ 2.10

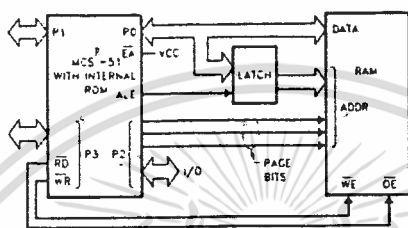


รูปที่ 2.10 ลักษณะการต่อกับหน่วยความจำภายนอก

จากรูป Port 0 และ 3 ของ MCS51 (8031) จะใช้เป็นแอดเดรสบััสสำหรับการติดต่อหน่วยความจำภายนอก นอกจากนั้น Port 0 ยังใช้เป็นบััสข้อมูลอีกด้วย ในการเข้าถึงหน่วยความจำภายนอก MCS51 จะส่งแอดเดรสบััสออกจาก Port 0 และ Port 3 หลังจากนั้นก็จะส่งสัญญาณ ALE (Address Latch Enable) ออกไปทำการแลทช์แอดเดรสไบต์ต่ำที่ Port 0 เอาไว้ แล้วก็เปลี่ยน Port 0 เป็นบััสข้อมูลแล้วส่งสัญญาณ $\overline{PSEN}$ ทำการนำข้อมูลเข้ามา

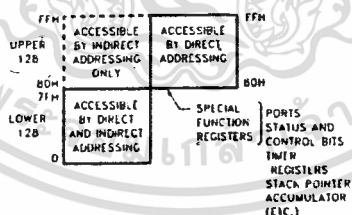
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำข้อมูล(Data memory) รูปที่ 2.9 ทางด้านขวาแสดงลักษณะการแบ่งหน่วยความจำภายในของ MCS51 ในรูปที่ 2.11 แสดงลักษณะการต่ออาร์ดแวร์สำหรับการเข้าถึงหน่วยความจำข้อมูลภายนอก เช่นเดียวกับการเข้าถึงหน่วยความจำโปรแกรม Port 0 จะมัลติเพลกซ์ระหว่างบัสแอดเดรสและบัสข้อมูล การอ่านและเขียนข้อมูล ที่พินจะใช้สัญญาณ $\overline{RD}$ และ $\overline{WR}$ สำหรับหน่วยความจำภายนอก ขนาดของหน่วยความจำข้อมูล นอกจากหน่วยความจำภายในขนาด 128 ไบต์ แล้ว หน่วยความจำภายนอกยังมีได้อีก 64 K



รูปที่ 2.11 แสดงการต่อสำหรับการเข้าถึงหน่วยความจำข้อมูลภายนอก

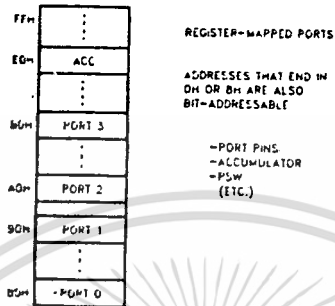
หน่วยความจำ ภายในของ MCS51 มีการแบ่งการใช้งานออกเป็นสองส่วน คือ 128 ไบต์ล่าง และ SFR ดังรูป



รูปที่ 2.12 หน่วยความจำข้อมูลภายใน

ในส่วนของ 128 ไบต์ล่าง ยังมีการแบ่งออกเป็นสามส่วน คือ ส่วนของรีจิสเตอร์ 32 ไบต์ ตั้งแต่แอดเดรส 00 ถึง 1FH โดยแบ่งรีจิสเตอร์ออกเป็น 4 ชุด ชุดละ 8 ตัว ในการใช้งานสามารถอ้างถึงได้เพียงครั้งละชุด ส่วนของบิตแอดเดรสเอเบิล (BIT ADDRESSABLE) ตั้งแต่แอดเดรส 20H ถึง 2FH และส่วนที่เหลือเป็นแรม ที่ใช้งานทั่วไป เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

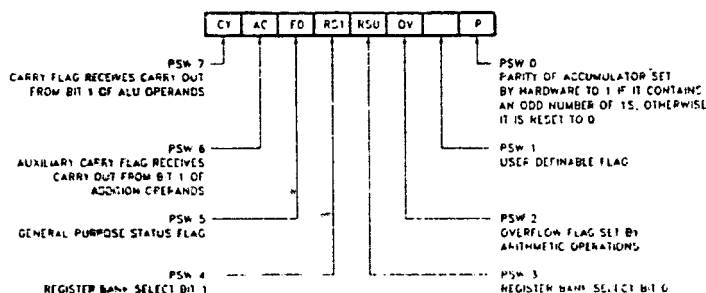
SFR เป็นส่วนของ แรม ภายในที่ทำหน้าที่พิเศษ สามารถเข้าถึงได้โดยการอ้างแอดเดรสตั้งแต่ 80H ถึง FFH เหมือนแรมทั่วไป SFR แต่ละตัวจะมีแอดเดรสต่างกันไป แอดเดรสตั้งแต่ 80H ถึง FFH ไม่ได้ใช้งานเป็น SFR ทั้งหมด อย่างไรก็ตามส่วนที่เหลือก็ไม่สามารถนำไปใช้งานได้ SFR ประกอบด้วยรีจิสเตอร์ต่างๆดังรูปที่ 2.13



รูปที่ 2.13 ส่วนประกอบของ SFR

โปรแกรมสเตตัสเวิร์ด(Program Status Word (PSW)) และ แฟล็ก(flag)

แฟล็กใน MCS51 ประกอบด้วยแมทแฟล็ก(math flag) 4 ตัว สำหรับแสดงผลการทำงานทางคณิตศาสตร์ และ ยูสเซอร์แฟล็ก(user flag) 3 ตัว สำหรับการกำหนดงานของผู้ใช้ แมทแฟล็ก ประกอบด้วยแครี่แฟล็ก (carry flag) , ออกซิลารีแครี่แฟล็ก (auxilliary carry flag) , โอเวอร์โฟลล์แฟล็ก (over flow flag) และ พาริตีแฟล็ก (Parity flag) ส่วนยูสเซอร์แฟล็ก ประกอบด้วย GFO , GF1 และ FO PSW ประกอบด้วยแฟล็ก และ บิต สำหรับการเลือกใช้ชุดของรีจิสเตอร์แมทแฟล็กทั้งหมด และยูสเซอร์แฟล็ก FO จะอยู่ใน PSW ดังรูปที่ 2.14 ส่วนยูสเซอร์แฟล็ก GFO และ GF1 จะอยู่ใน รีจิสเตอร์ PCON



เอกสารนี้เป็นเอกสารที่สงวนไว้รูปที่ 2.14 ใช้งาน Program Status Word าดให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.2 ตัวตั้งเวลา/ตัวนับ (Timer/Counter)

ใน 8051 มีตัวตั้งเวลา/ตัวนับ (Timer/Counter) เป็นรีจิสเตอร์ ขนาด 16 บิต จำนวน 2 ตัว คือ ตัวตั้งเวลา 0 (Timer 0) และ ตัวตั้งเวลา 1 (Timer 1) นอกจากนั้นตัวตั้งเวลาแต่ละตัวยังแบ่งออกเป็น รีจิสเตอร์ขนาด 8 บิต 2 ตัว คือ TH0 TL0 และ TH1 TL1 ตัวตั้งเวลาและตัวนับสามารถกำหนดการทำงานให้เป็นตัวตั้งเวลา หรือ ตัวนับ ได้อย่างใดอย่างหนึ่ง

- ในโหมดตัวตั้งเวลา (Timer Mode) ค่าของรีจิสเตอร์จะถูกเพิ่มค่าทุกๆ วัฏจักรแมชชีน (machine cycle) ดังนั้นจึงสามารถคิดว่าการทำงานในโหมดนี้จะเป็นการนับ วัฏจักรแมชชีนก็ได้ และเนื่องจากใน 1 วัฏจักรแมชชีนของ MCS-51 ประกอบไปด้วย 12 คาบของออสซิลเลเตอร์ (oscillator period) ดังนั้นอัตราเร็วในการนับ (Count Rate) จึงมีค่าเป็น $1/12$ ของความถี่ออสซิลเลเตอร์ที่ใช้ในระบบ

- ในโหมดตัวตั้งเวลา (Counter Mode) ค่าในรีจิสเตอร์จะถูกเพิ่มค่าทีละหนึ่งตาม การเปลี่ยนสถานะที่ขา T0 , T1 โดยการเปลี่ยนสถานะต้องเปลี่ยนค่าจาก 1 เป็น 0 (1 to 0 Transition)

การทำงานของตัวตั้งเวลา/ตัวนับ (Timer/Counter) ถูกควบคุมโดยรีจิสเตอร์ TMOD และ TCON

(MSB)				(LSB)			
GATE	C/T	M1	M0	GATE	C/T	M1	M0
Timer 1				Timer 0			
GATE	Gating control when set Timer/Counter "x" is enabled only while "INTx" pin is high and "TRx" control pin is set. When cleared Timer "x" is enabled whenever "TRx" control bit is set.	M1	M0	Operating Mode			
		0	0				
		0	1				
		1	0				
C/T	Timer or Counter Selector cleared for Timer operation (input from internal system clock) Set for Counter operation (input from "Tx" input pin)	1	1				
		1	1				

รูปที่ 2.15 TMOD : รีจิสเตอร์ควบคุมโหมดตัวตั้งเวลา/ตัวนับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในองค์กรเท่านั้น ไม่อนุญาตให้ส่งไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(MSB)				(LSB)			
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Symbol	Position	Name and Significance	Symbol	Position	Name and Significance
TF1	TCON.7	Timer 1 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine.	IE1	TCON.3	Interrupt 1 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
TR1	TCON.6	Timer 1 Run control bit. Set/cleared by software to turn Timer/Counter on/off.	IT1	TCON.2	Interrupt 1 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
TF0	TCON.5	Timer 0 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine.	IE0	TCON.1	Interrupt 0 Edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
TR0	TCON.4	Timer 0 Run control bit. Set/cleared by software to turn Timer/Counter on/off.	IT0	TCON.0	Interrupt 0 Type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.

รูปที่ 2.16 TCON : รีจิสเตอร์ควบคุมโหมดตัวตั้งเวลา/ตัวนับ (Timer/Counter Mode Control Register)

นอกจากผู้ใช้จะสามารถเลือกการทำงานของรีจิสเตอร์ให้เป็นตัวตั้งเวลา (Timer) หรือตัวนับ (Counter) ได้แล้ว ผู้ใช้ยังสามารถกำหนดให้ตัวตั้งเวลา หรือ ตัวนับ มีการทำงานที่พิเศษแยกย่อยลงไปอีกถึง 4 โหมด ตามความเหมาะสมของการใช้งาน (T0, T1 มีให้เลือก 4 โหมด แต่ T2 มีให้เลือก 3 โหมด) ดังจะได้กล่าวต่อไป

ตัวตั้งเวลา 0 และตัวตั้งเวลา 1 (TIMER 0 และ TIMER 1)

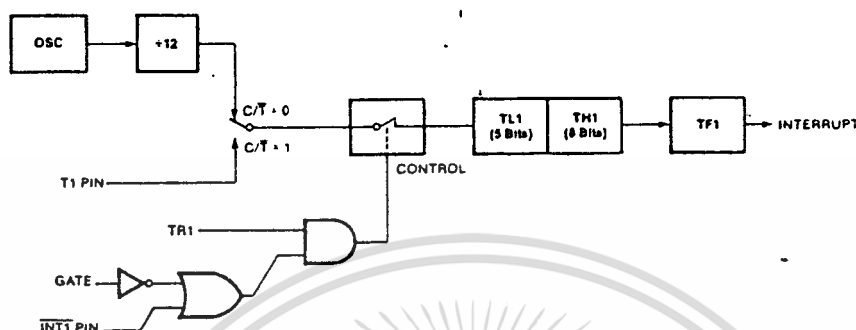
สามารถจะเลือกการทำงานให้เป็นตัวตั้งเวลา หรือ ตัวนับ ด้วยการเปลี่ยนค่าบิต C/T ซึ่งอยู่ในรีจิสเตอร์ TMOD ถ้าบิตนี้เป็น "0" หมายถึงให้เป็นตัวนับเวลา ถ้าเป็น "1" หมายถึงให้เป็นตัวนับ

ตัวตั้งเวลา/ตัวนับทั้ง 2 ตัวนี้มีโหมดการทำงานอยู่ 4 โหมด โดยการทำงานจะเป็นโหมดใดขึ้นอยู่กับค่าของบิต M0 , M1 ในรีจิสเตอร์ TMOD เช่นเดียวกับบิต C/T การทำงานในโหมด 0, 1, 2 จะคล้ายๆ กันสำหรับตัวตั้งเวลา/ตัวนับ ทั้ง 2 แต่ในโหมด 3 จะมีการทำงานที่ต่างออกไปจากทั้ง 3 โหมดดังนี้

โหมด 0 (MODE 0)

แต่ละตัวตั้งเวลา/ตัวนับ ในโหมดนี้จะถูกกำหนดการทำงานให้เป็นตัวนับ ขนาด 8 บิต ซึ่งจะถูกเพิ่มค่าต่อเนื่องนับสัญญาณจากรีจิสเตอร์แมกซีน หรือการเปลี่ยนสถานะลอจิกที่ขา T0, T1 ครบ 32 แล้ว (เป็นตัวนับ ขนาด 8 บิตที่ถูกรหัสด้วย 32) กล่าวอีกอย่างหนึ่งคือไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวตั้งเวลา แต่ละตัวจะทำงานเป็นตัวนับ เพื่อนับวัฏจักรแมชีน หรือการเปลี่ยนสถานะลอจิก ที่ขา T0, T1 ขนาด 13 บิตนั่นเอง โดยการนับสัญญาณ 32 ครั้งแรกจะใช้รีจิสเตอร์ TLx จำนวน 5 บิต ($2^5 = 32$) และนำมาประกอบกับรีจิสเตอร์ THx จำนวน 8 บิต เพื่อให้เป็นตัวนับ ขนาด 13 บิต



รูปที่ 2.17 ตัวตั้งเวลา/ตัวนับ 1 โหมด 0 : ตัวนับ 13 บิต
(Timer/Counter 1 Mode 0 : 13 bits counter)

จากรูปจะเห็นว่ารีจิสเตอร์ TL1 และ TH1 จะมาประกอบกันเข้าเป็นตัวนับ ขนาด 13 บิต เมื่อค่าในรีจิสเตอร์ทั้งสองเปลี่ยนจากเป็น 1 ทั้งหมดไปเป็น 0 ทั้งหมด จะมีผลไปเซตบิตแฟล็กอินเทอร์รัพต์ตัวตั้งเวลา (Timer Interrupt Flag) TF1 อินพุตที่ใช้ับถูกเอนเนเบิล(enable) ให้กับตัวตั้งเวลา เมื่อบิต TR1 และ Gate = 0 หรือสถานะที่ขา INT1 เป็น 1 อย่างใดอย่างหนึ่ง ดังแสดงในภาพ (การเซตให้ Gate เป็น 1 จะยอมให้ตัวตั้งเวลาถูกควบคุมโดยสถานะที่ขา INT1 ทำให้การวัดความกว้างของพัลส์ ทำได้ง่าย) TR1 เป็นบิตในรีจิสเตอร์ TCON ซึ่งแต่ละบิตมีความหมายดังแสดงในรูปที่ 2.16 ส่วน Gate เป็นบิตหนึ่งในรีจิสเตอร์ TMOD ดังได้กล่าวมาแล้ว

รีจิสเตอร์ขนาด 13 บิต ประกอบด้วย 8 บิตของรีจิสเตอร์ TH1 และ 5 บิตของรีจิสเตอร์ TL1 ส่วน 3 บิตบนของรีจิสเตอร์ TL1 ไม่ถูกกำหนดและไม่ถูกใช้ การเซตค่าของบิตรันแฟล็ก (Run Flag (TR1)) ก็ไม่ได้เคลียร์ค่าในรีจิสเตอร์ทั้งสอง

โหมด 1 (MODE 1)

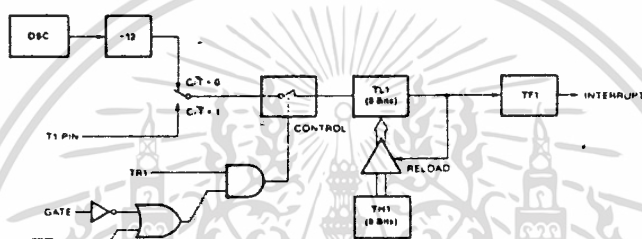
ในโหมด 1 นี้มีการทำงานเหมือนกับโหมด 0 ทุกประการ เว้นแต่ค่าในรีจิสเตอร์ตัวตั้งเวลา (Timer Register) จะถูกให้ทำงาน ทั้ง 16 บิตเลย นั่นคือในโหมดนี้ค่าของบิตรันแฟล็ก (Run Flag (TR1)) ก็ไม่ได้เคลียร์ค่าในรีจิสเตอร์ทั้งสอง

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวตั้งเวลา/ตัวนับ(Timer/Counter)จะเป็นตัวนับ หรือ ตัวตั้งเวลา ขนาด16 บิตแทน

โหมด 2 (MODE 2)

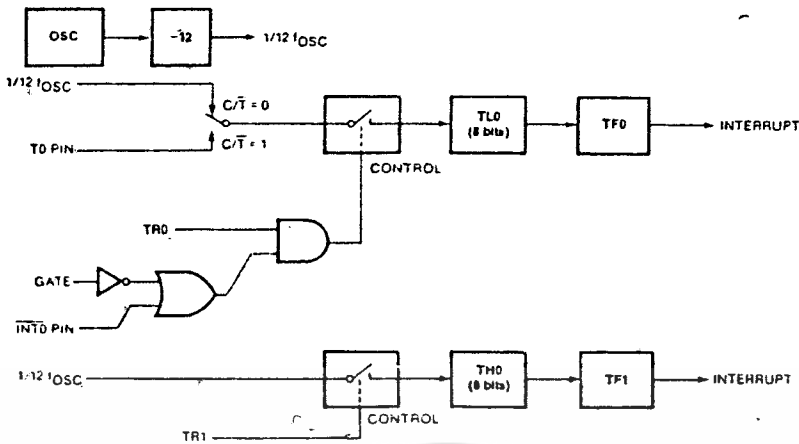
ในโหมด 2 จะกำหนดรีจิสเตอร์ตัวตั้งเวลา/ตัวนับ(Timer/Counter Register) ทำงานเป็นตัวนับ 8 บิต(8-bits counter (TLx)) ซึ่งมีการโหลดค่าเองโดยอัตโนมัติ ด้วยค่าในรีจิสเตอร์ THx เมื่อเกิดโอเวอร์โฟลด์ (Overflow) ในตัวตั้งเวลา/ตัวนับ (Timer/Counter) ซึ่งค่าในรีจิสเตอร์ THx นี้สามารถกำหนดได้ล่วงหน้าโดยซอฟต์แวร์ และจะไม่เปลี่ยนแปลงเมื่อถูกโหลดไปไว้ในรีจิสเตอร์ TL1



รูปที่ 2.18 ตัวตั้งเวลา/ตัวนับ 1 โหมด 2

โหมด 3 (MODE 3)

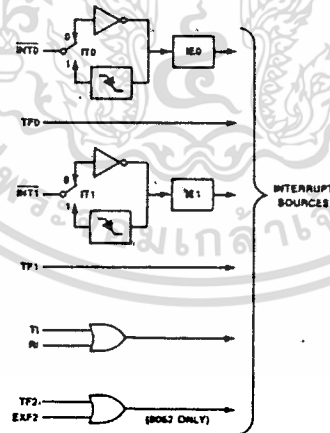
ในการทำงานของตัวตั้งเวลา/ตัวนับ (Timer/Counter) โหมด 3 นี้ ตัวตั้งเวลา 1 (Timer 1) จะไม่มีการนับ ซึ่งมีผลเหมือนกับให้ค่า $TR1 = 0$ ตัวตั้งเวลา 0 (Timer 0) ในโหมด 3 จะทำให้รีจิสเตอร์ TLO และ TH0 ทำงานเหมือนเป็นตัวนับขนาด 8 บิต แยกต่างหากจากกัน 2 ตัว สำหรับการทำงานในโหมด 3 ของตัวตั้งเวลา 0 (Timer 0) แสดงในรูปที่ 2.19 ซึ่งจะเห็นว่ารีจิสเตอร์ TLO ใช้ตัวตั้งเวลา 0 (Timer 0) ควบคุมบิต $C/\overline{T}$, Gate , $\overline{TR0}$, $\overline{INT0}$ และ $\overline{TFO}$ ส่วนรีจิสเตอร์ TH0 จะถูกบังคับให้เป็นตัวตั้งเวลา (นับจำนวนวัฏจักรแมซิน) และรับการให้ TR1 และ TF1 จากตัวตั้งเวลา 1 (Timer 1) ดังนั้นขณะรีจิสเตอร์ TH0 จะควบคุมการอินเทอร์รัพท์ตัวตั้งเวลา 1 (Timer 1 interrupt)



รูปที่ 2.19 ตัวตั้งเวลา/ตัวนับ 0 โหมด 3 : ตัวนับ 8 บิต 2 ตัว
(Timer/Counter 0 Mode 3 : Two 8-bits Counter)

2.4.3 อินเทอร์รัพท์ (Interrupt)

8051 มีแหล่งที่มาของการอินเทอร์รัพท์ (Interrupt Source) ให้ 5 อินเทอร์รัพท์ มีรายละเอียดดังนี้



รูปที่ 2.20 แหล่งที่มาของการอินเทอร์รัพท์

อินเทอร์รัพท์ภายนอก (External Interrupts) $\overline{\text{INT0}}$ และ $\overline{\text{INT1}}$

เป็นอินเทอร์รัพท์ที่เกิดขึ้นภายนอก แต่ละตัวสามารถเลือกให้เป็นเลเวลแอกทิเวท (Level-Activated) หรือ ทรานซิชันแอกทิเวท (Transition-Activated) อย่างใดอย่าง

หนึ่งขึ้นอยู่กับบิต IT0 และ IT1 ในรีจิสเตอร์ TCON

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แฟล็ก(Flag) ที่ทำให้เกิดการอินเทอร์รัพท์ในชิพจริงๆ แล้วคือค่าในบิต IE0 และ IE1 ของรีจิสเตอร์ TCON โดยเมื่อเกิดอินเทอร์รัพท์จากภายนอกขึ้น แฟล็กซึ่งทำให้เกิดอินเทอร์รัพท์จะถูกเคลียร์โดยฮาร์ดแวร์ เมื่อชิพถูกย้ายไปทำงานที่รoutines บริการการอินเทอร์รัพท์(interrupt service routine) ก็ต่อเมื่ออินเทอร์รัพท์ตัวนั้นเป็นชนิดทรานซิชัน-แอคทิเวท(transition-activated) แต่ถ้าอินเทอร์รัพท์ตัวนั้นเป็นชนิดเลเวลแอคทิเวท(level-activated) แล้ว เมื่อชิพถูกย้ายไปทำงานที่รoutines บริการการอินเทอร์รัพท์(interrupt service routine) จะไม่เคลียร์บิต IEx ให้ วงจรภายนอกที่ขออินเทอร์รัพท์ชิพจะต้องทำหน้าที่ควบคุมสถานะของสัญญาณที่ขา $\overline{INTx}$ ให้กลับสู่สภาพเดิมเอง

อินเทอร์รัพท์ตัวตั้งเวลา 0 และ 1(Timer 0 และ Timer 1 Interrupts) ถูกทำให้เกิดอินเทอร์รัพท์ได้โดยบิต TFO และ TF1 ซึ่งถูกเซ็ตโดยการเปลี่ยนค่าจาก 1 ทั้งหมดเป็น 0 ทั้งหมดในรีจิสเตอร์ตัวตั้งเวลาและตัวนับ(Timer/Counter Register) ของแต่ละตัว(ยกเว้นตัวตั้งเวลา 0(Timer 0) ในโหมด 3) เมื่ออินเทอร์รัพท์ตัวตั้งเวลา(Timer Interrupt) เกิดขึ้นแฟล็กที่ทำให้เกิดอินเทอร์รัพท์จะถูกเคลียร์โดยฮาร์ดแวร์บนชิพเมื่อชิพถูกย้ายไปทำงานในส่วนรoutines บริการ(Service Routine)

อินเทอร์รัพท์พอร์ทอนุกรม(Serial Port Interrupt) สามารถสร้างสัญญาณอินเทอร์รัพท์ได้ โดยสัญญาณที่จะทำให้เกิดอินเทอร์รัพท์ได้มาจากบิต TI หรือ RI โดยนำมาผ่านเกตแบบ OR และแฟล็ก(Flag) ทั้งสองตัวนี้จะไม่ถูกเคลียร์โดยฮาร์ดแวร์เมื่อชิพไปทำงานในส่วนรoutines บริการ(Service Routine) เพราะจริงๆแล้วในรoutines บริการจะต้องหาว่า RI หรือ TI เป็นตัวทำให้เกิดอินเทอร์รัพท์ขึ้น และบิตนั้นจะถูกเคลียร์ได้โดยซอฟต์แวร์เท่านั้น

บิตทั้งหมดที่ทำให้เกิดสัญญาณอินเทอร์รัพท์สามารถถูกเซ็ตหรือเคลียร์โดยซอฟต์แวร์ โดยมีผลเหมือนกับว่ามันถูกเซ็ตหรือถูกเคลียร์โดยฮาร์ดแวร์ นั่นคือ อินเทอร์รัพท์ทั้งหมดที่กล่าวมาสามารถเอเนเบิล(enable)ให้เกิดสัญญาณอินเทอร์รัพท์หรืออินเทอร์รัพท์ที่กำลังรอการบริการจากชิพอยู่สามารถถูกยกเลิกได้ด้วยซอฟต์แวร์

แหล่งที่มาของการอินเทอร์รัพท์(Interrupt Source)แต่ละชนิดเหล่านี้สามารถถูกเอเนเบิล(enable)เป็นตัว ๆ หรือถูกดิสเอเนเบิล(disable) เป็นตัวๆ โดยการเปลี่ยนแปลงบิตใน SFR IE ซึ่งแต่ละบิตมีความหมายดังในรูปที่ 2.21 และจะเห็นได้ว่ารีจิสเตอร์ IE จะมีบิตที่เป็นโกลบอลดิสเอเนเบิลบิต(Global Disable Bit (EA)) อยู่ด้วย ซึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่สามารถนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(MSB)				(LSB)			
EA	—	ET2	ES	ET1	EX1	ET0	EX0

Enable Bit = 1 enables the interrupt.
Enable Bit = 0 disables it.

Symbol	Position	Function
EA	IE.7	disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	reserved
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	Serial Port interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.

User software should never write 1s to unimplemented bits, since they may be used in future MCS-51 products.

รูปที่ 2.21 IE รีจิสเตอร์การอินเทรปต์

(IE : Interrupt Enable Register)

โครงสร้างระดับความสำคัญ(Priority Level Structure)

แต่ละแหล่งที่มาของการอินเทรปต์(Interrupt Source)สามารถถูกเลือกระดับความสำคัญได้ 2 ระดับ โดยการเข้ารหัสหรือเคลียร์บิตในรีจิสเตอร์ IP ดังในรูปที่ 2.22

(MSB)							
—	—	PT2	PS	PT1	PX1	PT0	PX0

Priority bit = 1 assigns high priority.
Priority bit = 0 assigns low priority.

Symbol	Position	Function
—	IP.7	reserved
—	IP.6	reserved
PT2	IP.5	Timer 2 interrupt priority bit.
PS	IP.4	Serial Port interrupt priority bit.
PT1	IP.3	Timer 1 interrupt priority bit.
PX1	IP.2	External interrupt 1 priority bit.
PT0	IP.1	Timer 0 interrupt priority bit.
PX0	IP.0	External interrupt 0 priority bit.

User software should never write 1s to unimplemented bits, since they may be used in future MCS-51 products.

รูปที่ 2.22 IP รีจิสเตอร์ระดับความสำคัญของการอินเทรปต์

(IP : Interrupt Priority Register)

ระดับความสำคัญของการอินเทรปต์มีได้ 2 ระดับได้แก่

อินเทรปต์ระดับความสำคัญต่ำ(Low Priority Interrupt) : สามารถถูกอินเทรปต์โดยอินเทรปต์ระดับความสำคัญสูง (High Priority Interrupt) ได้ แต่ไม่สามารถถูกอินเทรปต์โดยอินเทรปต์ระดับความสำคัญต่ำตัวอื่นๆ ได้

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเทอร์รัพท์ระดับความสำคัญสูง(High Priority Interrupt) :ไม่สามารถถูกอินเทอร์รัพท์โดยแหล่งอินเทอร์รัพท์ตัวอื่นได้เลย

ถ้ามีการขออินเทอร์รัพท์พร้อมกัน 2 ชนิด ซึ่งมีระดับความสำคัญในการขออินเทอร์รัพท์ไม่เท่ากัน อินเทอร์รัพท์ที่มีระดับความสำคัญสูงกว่า(High Priority Interrupt) จะได้รับการบริการก่อน แต่ถ้ามีการขออินเทอร์รัพท์พร้อมกัน 2 ชนิด ซึ่งมีระดับความสำคัญเท่ากัน ลำดับการรับสัญญาณอินเทอร์รัพท์ภายในจะกำหนดเองว่าอินเทอร์รัพท์ชนิดใดจะถูกบริการก่อน ดังนั้นภายในระดับความสำคัญของการขออินเทอร์รัพท์หนึ่งๆ จะมีระดับความสำคัญในการขออินเทอร์รัพท์ย่อย(Second Priority Structure) ดังนี้

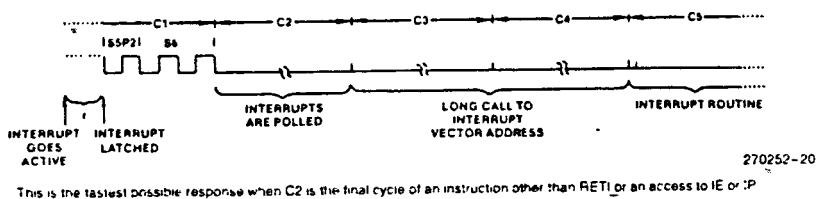
แหล่งที่มา(Source) ระดับความสำคัญ(Priority Within Level)

- | | |
|-------------|-----------------|
| 1. IEO | ความสำคัญสูงสุด |
| 2. TFO | |
| 3. IE1 | |
| 4. TF1 | |
| 5. RI+TI | |
| 6. TF2+EXF2 | ความสำคัญต่ำสุด |

สังเกตว่าระดับความสำคัญต่างๆ จะถูกใช้เพียงเพื่อแก้ปัญหาการเกิดสัญญาณอินเทอร์รัพท์พร้อมกันขึ้นมาเท่านั้น

MCS-51 จัดการกับสัญญาณอินเทอร์รัพท์อย่างไร

แฟล็กอินเทอร์รัพท์ (Interrupt Flag) ของอินเทอร์รัพท์แต่ละชนิด จะถูกตรวจสอบ(Sample) ที่ทุกๆ S5P2 ของทุกวัฏจักรแมชีน ค่าที่ตรวจสอบจะถูกรับเข้ามาระหว่างวัฏจักรที่ตามมา



รูปที่ 2.23 แผนผังเวลาการตอบสนองการอินเทอร์รัพท์

(Interrupt Response Timing Diagram)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การตอบสนองต่อสัญญาณอินเทอร์รัพท์จะมีแผนผังดังในรูปที่ 2.23 ซึ่งจะเห็นว่าประกอบไปด้วยวัฏจักรต่างๆ กัน คือ วัฏจักรตรวจจับอินเทอร์รัพท์ (จะตรวจสอบสถานะของบิตที่อินเทอร์รัพท์ซีพียูทุกๆ S5P2 ของทุกวัฏจักรแมชีน) วัฏจักรการหาชนิดของอินเทอร์รัพท์ (ตรวจสอบว่าเป็นอินเทอร์รัพท์ชนิดใด) และวัฏจักรการล้รางคำสั่ง LONG CALL ไปยังรูทีนบริการการอินเทอร์รัพท์ (interrupt service routine) ที่เหมาะสม (ใช้ 2 วัฏจักร) และ วัฏจักรของการทำคำสั่งที่อยู่ในรูทีนบริการการอินเทอร์รัพท์ (interrupt service routine) โดยมีรายละเอียดดังนี้

ถ้า แฟล็ก ของอินเทอร์รัพท์ตัวใดตัวหนึ่งอยู่ในสถานะถูกเซ็ทที่ S5P2 ของวัฏจักรตรวจจับอินเทอร์รัพท์ ซึ่งเป็นวัฏจักรก่อนหน้าวัฏจักร การหาชนิดของอินเทอร์รัพท์ซึ่งจะทำการค้นหาว่าอินเทอร์รัพท์ชนิดใดเป็นผู้ขอเข้ามา ในวัฏจักรถัดไปก็จะทำคำสั่ง LONG CALL ไปที่รูทีนบริการการอินเทอร์รัพท์ (interrupt service routine) ที่เหมาะสม วัฏจักรการทำคำสั่ง LONG CALL โดยอาร์ดแวรจะถูกระงับสำเร็จก็ต่อเมื่อไม่ถูกป้องกันโดยสภาวะดังต่อไปนี้

- 1 ซีพียูกำลังทำคำสั่งในรูทีนบริการการอินเทอร์รัพท์ (interrupt service routine) ของอินเทอร์รัพท์ซึ่งมีความสำคัญเทียบเท่าหรือสูงกว่าอยู่ในขณะนั้น
- 2 วัฏจักรที่ตรวจหาชนิดของอินเทอร์รัพท์ไม่ใช่วัฏจักรสุดท้ายของคำสั่งที่ซีพียูกำลังปฏิบัติงานอยู่ นั่นคือซีพียูกำลังทำงานของคำสั่งใดๆ ยังไม่เสร็จสิ้นขณะตรวจหาชนิดของอินเทอร์รัพท์พบ
- 3 คำสั่งที่กำลังปฏิบัติอยู่ในขณะนั้นเป็น RETI หรือคำสั่งใดๆ ที่มีการเขียนข้อมูลไปยังรีจิสเตอร์ IE หรือ IP

สภาวะทั้งสามนี้จะป้องกันมิให้ซีพียูทำคำสั่ง LONG CALL ไปยังรูทีนบริการการอินเทอร์รัพท์ (interrupt service routine) ขณะตรวจพบการขออินเทอร์รัพท์

สภาวะที่ 1 มีไว้เพื่อให้อินเทอร์รัพท์ซึ่งมีความสำคัญสูงกว่าได้รับการบริการก่อนโดยอินเทอร์รัพท์ที่มีความสำคัญต่ำกว่าไม่สามารถขัดจังหวะได้

สภาวะที่ 2 ทำให้แน่ใจว่าคำสั่งที่กำลังทำอยู่ในขณะนั้นจะถูกทำให้เสร็จก่อนการย้ายไปทำคำสั่งใดๆ ที่รูทีนบริการการอินเทอร์รัพท์ (interrupt service routine) ที่เกี่ยวข้อง

สภาวะที่ 3 ทำให้แน่ใจว่า ถ้าคำสั่งที่กำลังทำอยู่เป็น RETI หรือการเข้าถึงข้อมูลในรีจิสเตอร์ IE หรือ IP แล้ว อย่างน้อยจะต้องมีคำสั่งอีก 1 คำสั่งถูกปฏิบัติ ก่อนอินเทอร์รัพท์จะเกิดซ้ำขึ้นอีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รัฟท์ใดๆ จะถูกบริการ

วัฏจักรการตรวจหาชนิดของอินเทอร์รัฟท์จะถูกทำซ้ำแต่ละวัฏจักรแมซิน และค่าที่รับเข้ามา (ชนิดของอินเทอร์รัฟท์) จะเป็นค่าที่ปรากฏเมื่อ S5P2 ของวัฏจักรก่อนหน้านี้ สังเกตว่าถ้าบิตที่ทำหน้าที่อินเทอร์รัฟท์ถูกเซ็ท (Active) แต่ไม่ถูกตอบสนองเพราะ 1 ใน 3 ของสภาวะข้างต้น และปรากฏว่าไม่ทำงานเมื่อสภาวะที่ป้องกันการทำคำสั่ง LONG CALL หมดไปแล้ว อินเทอร์รัฟท์นี้จะไม่ได้รับการบริการ (ทำคำสั่งในรูทีนบริการการอินเทอร์รัฟท์) หรือกล่าวได้ว่าสถานะของอินเทอร์รัฟท์แต่ละตัวซึ่งครั้งหนึ่งเคยทำงาน แต่ไม่ได้รับการบริการเพราะถูกป้องกันด้วยสภาวะดังกล่าวมาข้างต้นจะไม่ถูกจำไว้ เพราะแต่ละวัฏจักรของการตรวจหาชนิดของอินเทอร์รัฟท์จะรับสถานะของอินเทอร์รัฟท์ใหม่เข้ามาเสมอ

สังเกตว่าถ้าอินเทอร์รัฟท์ที่มีความสำคัญสูงถูกทำงานก่อน S5P2 ของวัฏจักรแมซิน ที่มีชื่อย่อว่า C3 ในรูปที่ 2.23 และไม่มีสภาวะที่ป้องกันการทำคำสั่ง LONG CALL มันก็จะได้รับการบริการในระหว่าง C5 และ C6 โดยปราศจากการบริการของอินเทอร์รัฟท์ที่มีความสำคัญน้อยกว่า

ดังนั้นโปรเซสเซอร์จะรับบริการอินเทอร์รัฟท์โดยการทำคำสั่ง LONG CALL ไปที่รูทีนบริการการอินเทอร์รัฟท์ที่เหมาะสม ในบางกรณีมันจะเคลียร์บิตที่ก่อให้เกิดอินเทอร์รัฟท์ด้วยเลย แต่ในบางกรณีมันจะไม่เคลียร์ให้ ดังนั้นผู้ใช้ต้องบรรจุคำสั่งเคลียร์บิตเหล่านี้เองในซอฟต์แวร์

แฟล็กการอินเทอร์รัฟท์ภายนอก(external interrupt flag(IE0 และ IE1)) จะถูกเคลียร์ต่อเมื่อเป็นทรานซิชันแอคทิเวท(transition-activated) คำสั่ง LONG CALL ที่กระทำโดยฮาร์ดแวร์จะพুষ(PUSH) ข้อมูลของโปรแกรมเคาน์เตอร์ (Program Counter) ลงไปไว้ที่สแตก (stack) (ไม่มีการเก็บค่าของรีจิสเตอร์ PSW) และโหลดค่าของโปรแกรมเคาน์เตอร์ใหม่ด้วยแอดเดรสที่ตรงกับรูทีนบริการการอินเทอร์รัฟท์ที่เกี่ยวข้องกับสัญญาณอินเทอร์รัฟท์ขณะนั้น ดังแสดงข้างล่าง

แหล่งกำเนิด	แอดเดรสเวคเตอร์
IE0	0003H
TFO	000BH
IE1	0013H
TF1	001BH
RI+TI	0023H
TF2+EXF2	002BH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำคำสั่งในรoutinesบริการการอินเทอร์รัพต์นั้นจะทำได้เรื่อยๆ จนกระทั่งพบคำสั่ง RETI ซึ่งเป็นตัวบอกชี้พินัยว่า routine อินเทอร์รัพต์สิ้นสุดลงแล้ว จากนั้นซีพียูจะไปพอฟ (POP) เอาค่า 2 ไบท์สูงสุดจากสแตกและโหลดให้กับโปรแกรมเคาน์เตอร์ เพื่อกลับไปทำงานเดิมที่ทำอยู่ก่อนได้รับอินเทอร์รัพต์

สังเกตว่าคำสั่ง RET ธรรมดาจะสามารถพอฟ (POP) เอาค่า 2 ไบท์สูงสุดจากสแตกและโหลดให้กับโปรแกรมเคาน์เตอร์ได้เช่นเดียวกับคำสั่ง RETI แต่หากใช้ RET แล้วระบบควบคุมการอินเทอร์รัพต์ (interrupt control system) จะยังคิดว่าอินเทอร์รัพต์ยังคงถูกกระทำอยู่

อินเทอร์รัพต์ภายนอก (EXTERNAL INTERRUPT)

แหล่งกำเนิดอินเทอร์รัพต์จากภายนอกสามารถถูกโปรแกรมให้เป็นเลเวลแอกทิเวท (level-activated) หรือ ทรานซิชันแอกทิเวท (transition-activated) ได้ โดยการเซ็ทหรือเคลียร์บิต IT1, IT0 ในรีจิสเตอร์ TCON โดยถ้า ITx มีค่าเป็น 0 อินเทอร์รัพต์ภายนอกจะเป็นชนิดเลเวลแอกทิเวท (level-activated) โดยสถานะต่ำที่ขา $\overline{INTx}$ จะเป็นการอินเทอร์รัพต์ซีพียู ถ้าบิต ITx=1 , อินเทอร์รัพต์ภายนอก x จะเป็นชนิดทรานซิชันแอกทิเวท (transition-activated) (edge-triggered) ในโหมดนี้ถ้าการตรวจสอบที่ขา $\overline{INTx}$ พบว่ามีสถานะเป็นสูง (high) ใน 1 วัฏจักรและเป็นสถานะต่ำ (low) ในวัฏจักรถัดไป แฟล็กการขออินเทอร์รัพต์ (interrupt request flag) IEx ในรีจิสเตอร์ TCON จะถูกเซ็ท แล้วบิต IEx นี้จะไปอินเทอร์รัพต์ซีพียูต่อไป

เพราะว่าอินเทอร์รัพต์ภายนอกถูกตรวจสอบ 1 ครั้ง ในแต่ละวัฏจักรแมชีน และสัญญาณอินพุตสูงหรือต่ำควรจะต้องค้างไว้อย่างน้อย 12 คาบ (period) ของความถี่ออสซิลเลเตอร์ เพื่อให้แน่ใจในการตรวจสอบถ้าอินเทอร์รัพต์ภายนอกเป็นทรานซิชันแอกทิเวท (transition-activated) แหล่งกำเนิดอินเทอร์รัพต์ภายนอกจะต้องคงค่าเป็นสถานะต่ำอย่างน้อย 1 วัฏจักร เพื่อให้มั่นใจว่าการเปลี่ยนแปลงถูกตรวจพบอย่างแน่นอน ดังนั้น แฟล็กการขออินเทอร์รัพต์ (interrupt request flag) จะถูกเซ็ท IEx จะถูกเคลียร์โดยอัตโนมัติ เมื่อซีพียูทำคำสั่ง LONG CALL ไปที่ routines บริการการอินเทอร์รัพต์

ถ้าอินเทอร์รัพต์ภายนอก เป็นเลเวลแอกทิเวท (level-activated) วงจรที่กำเนิดอินเทอร์รัพต์ภายนอกต้องรักษาสถานะการขออินเทอร์รัพต์ (Active) ไว้จนกระทั่งไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินเทอร์เน็ตเกิดขึ้นจริงๆ และมันต้องเปลี่ยนสถานะของการใช้อินเทอร์เน็ตให้กลับมามีค่าเหมือนเดิม (Deactive) ก่อนที่ฐานบริการการอินเทอร์เน็ตจะถูกกระทำเสร็จ มิฉะนั้นแล้วฐานบริการการอินเทอร์เน็ตเดียวกันนี้จะถูกทำงานซ้ำอีก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 หลักการทำงาน

ข้อกำหนดเบื้องต้นสำหรับโครงงานนี้ได้แก่

1. สามารถเก็บเบอร์โทรศัพท์ของสถานที่ที่ต้องการจะติดต่อได้ 3 เบอร์ และบันทึกเสียงลงใน EPROM ได้ประมาณ 1 นาที เพื่อสำหรับใช้ในการเตือนภัย
2. เมื่อได้รับไฟเลี้ยงจากวงจรเตือนภัยแล้ว สามารถตรวจสอบสัญญาณกริ่งโทรศัพท์และโทรไปยังผู้รับ (สถานีดับเพลิง) หรือ เปลี่ยนเบอร์โทรศัพท์ได้โดยอัตโนมัติ เมื่อไม่มีผู้รับสายภายใน 15 วินาที
3. สามารถนำข้อมูลที่เก็บใน EPROM แปลงกลับเป็นสัญญาณเสียง เพื่อส่งไปตามคู่สายโทรศัพท์ได้

จากข้อกำหนดดังกล่าวที่วางไว้ข้างต้น จะต้องออกแบบสร้างเพื่อให้ได้คุณสมบัติดังกล่าว โดยสามารถแบ่งการออกแบบได้เป็น 2 เครื่อง คือ

- วงจรส่วนบันทึกลง EPROM
- วงจรเตือนภัยทางโทรศัพท์อัตโนมัติ

เราจะกล่าวถึงรายละเอียดของวงจรทั้งสองดังต่อไปนี้

3.1 วงจรส่วนบันทึกลง EPROM

จะขอแยกเป็น 2 ส่วน คือ

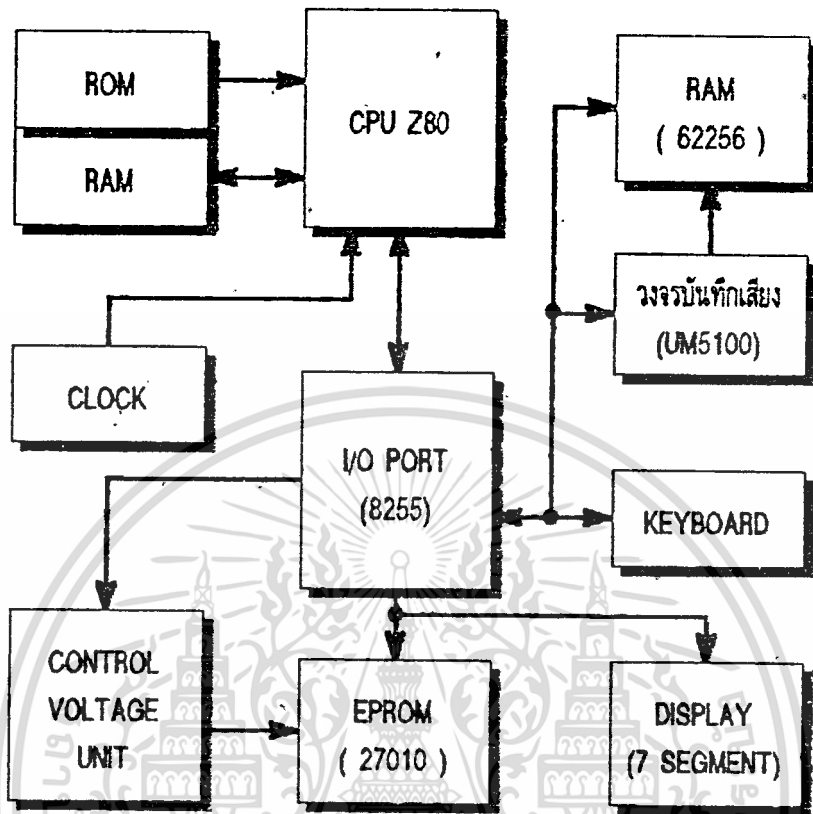
3.1.1 ส่วนฮาร์ดแวร์ (Hardware Part)

3.1.2 ส่วนซอฟต์แวร์ (Software Part)

3.1.1 ส่วนฮาร์ดแวร์

หมายถึง ส่วนของวงจรทั้งหมดที่จะถูกออกแบบให้สามารถทำงาน ได้ตามคำสั่งของซอฟต์แวร์ ซึ่งจากข้อกำหนดข้างต้นสามารถนำมาเขียนแผนผังการทำงาน ได้ดังรูปที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1 แผนผังการทำงานส่วนฮาร์ดแวร์ของวงจรมันติกเสียง EPROM

สำหรับรายละเอียดการออกแบบ และการทำงานของวงจรถัดไป จะแยกอธิบายออกเป็นส่วนๆ ในบทต่อไป

3.1.2 ส่วนซอฟต์แวร์

หมายถึง ส่วนของโปรแกรมทั้งหมด ที่ใช้สำหรับควบคุมการทำงานของวงจรถัดไป ซึ่งจะทำให้การโปรแกรมลงบน Z80

- โปรแกรมสำหรับบันทึกข้อมูล และ เปลี่ยนแปลงแก้ไขข้อมูล
- โปรแกรมสำหรับบันทึกเสียง และ โหลดสัญญาณเสียง
- โปรแกรมสำหรับตรวจสอบ EPROM และบันทึกลง EPROM

ส่วนวงจรบันทึกลง EPROM สามารถอธิบายการทำงานได้คร่าวๆ ดังนี้

Z80 จะรับหมายเลขโทรศัพท์จากวงจรส่วนคีย์บอร์ดและแสดงผลเป็นแบบเจ็ดเซกเมนต์ (7 segment) ให้ผู้ใช้ได้ทราบว่ากดเบอร์โทรศัพท์ถูกหรือไม่ ในส่วนนี้สามารถเก็บเบอร์โทรศัพท์ได้ 3 เบอร์ หลังจากนั้นก็จะเริ่มการบันทึกเสียงโดยผ่านวงจรแปลงสัญญาณอนาล็อกให้เป็นสัญญาณดิจิทัล เข้ามาเก็บในสแตติกแรม (Static RAM) ซึ่งสามารถบันทึกเสียงได้เป็นเวลาประมาณ 1 นาที หลังจากนั้นสามารถหวนคำพูดได้ ถ้าไม่พอใจก็ทำการบันทึกใหม่ได้ เมื่อพอใจแล้วก็จะทำการบันทึกเสียง เบอร์โทรศัพท์ และโปรแกรมควบคุมต่างๆ ลงใน EPROM เพื่อที่จะไปใช้ในวงจรเตือนภัยทางโทรศัพท์อัตโนมัติต่อไป

3.2 วงจรเตือนภัยทางโทรศัพท์อัตโนมัติ

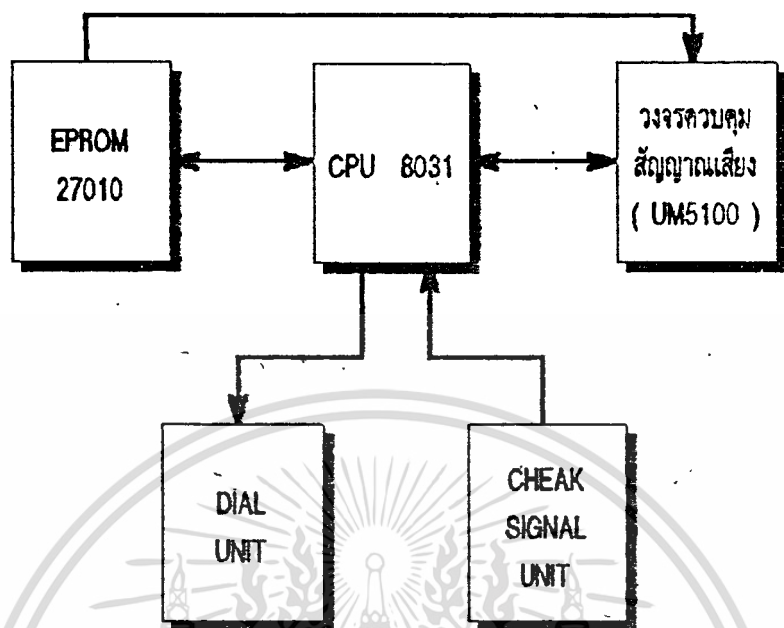
จะขอแยกเป็น 2 ส่วน คือ

3.2.1 ส่วนฮาร์ดแวร์ (Hardware Part)

3.2.2 ส่วนซอฟต์แวร์ (Software Part)

3.2.1 ส่วนฮาร์ดแวร์

หมายถึง ส่วนของวงจรทั้งหมดที่จะถูกออกแบบให้สามารถทำงาน ได้ตามคำสั่งของซอฟต์แวร์ ซึ่งจากข้อกำหนดข้างต้นสามารถนำมาเขียนแผนผังการทำงาน ได้ดังรูปที่ 3.2



รูปที่ 3.2 แผนผังการทำงานส่วนฮาร์ดแวร์ของวงจรเตือนภัยทางโทรศัพท์อัตโนมัติ

สำหรับรายละเอียดการออกแบบ และการทำงานของวงจรทั้งหมด จะแยกอธิบายออกเป็นส่วนๆ ในบทต่อไป

3.2.2 ส่วนซอฟต์แวร์

หมายถึง ส่วนของโปรแกรมทั้งหมด ที่ใช้สำหรับควบคุมการทำงานของวงจรส่วนนี้ ซึ่งจะทำการโปรแกรมลงบน 8031

โปรแกรมของวงจรเตือนภัยทางโทรศัพท์อัตโนมัติ มีลักษณะดังนี้

- โปรแกรมแยกจากส่วนวงจรบันทึกเลข EPROM โดยอิสระ
- มีโปรแกรมส่วนการติดต่อและควบคุมการส่งสัญญาณเสียง

เอกสารนี้เป็นเอกสารของบริษัทเอกชน ไม่ควรเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีโปรแกรมการหมุนเบอร์โทรศัพท์ การเปลี่ยนเบอร์โทรศัพท์ และตรวจสอบผลของการติดต่อ

ส่วนวงจรเตือนภัยทางโทรศัพท์อัตโนมัติ สามารถอธิบายการทำงานคร่าวๆ ได้ดังนี้

เมื่อวงจรได้รับไฟเลี้ยงแล้ว ระบบจะทำการตัดระบบโทรศัพท์สายอื่นทิ้งทันที หลังจากนั้นก็จะเริ่มส่งสัญญาณไปควบคุมให้วงจรหมุนเบอร์โทรศัพท์ทำงาน เพื่อโทรไปยังสถานที่ที่เราต้องการ แล้วทำการตรวจสอบว่าสายว่างหรือเปล่า มีคนรับหรือยัง ถ้าสายไม่ว่างก็จะหมุนเบอร์ใหม่ให้ ถ้าสายว่างก็จะเริ่มส่งข้อมูลผ่านวงจรแปลงสัญญาณเสียง ส่งไปรวมสายโทรศัพท์ทันที



บทที่ 4

รายละเอียดการออกแบบและการทำงาน

สำหรับรายละเอียดการออกแบบและการทำงานของโครงงานนี้จะขอแบ่งออกเป็น 2 หัวข้อใหญ่ คือ

4.1 วงจรส่วนบันทึกถาวร EPROM

4.2 วงจรเตือนภัยทางโทรศัพท์อัตโนมัติ

4.1 วงจรส่วนบันทึกถาวร EPROM

เราจะอธิบายได้เป็นสองส่วนคือ

4.1.1 ส่วนฮาร์ดแวร์

4.1.2 ส่วนซอฟต์แวร์

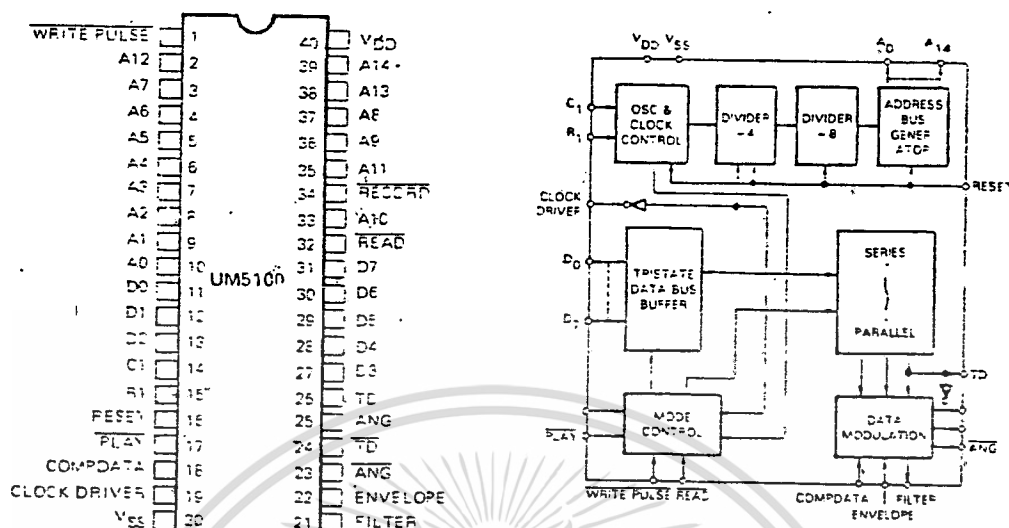
4.1.1 ส่วนฮาร์ดแวร์ สามารถแบ่งได้ดังนี้

วงจรบันทึกเสียงพูดและเรียกเสียงกลับ

โครงงานนี้จะใช้ไอซีเบอร์ UM 5100 ในการบันทึกเสียงพูดและเรียกเสียงกลับ โดยไอซีเบอร์นี้จะมีลักษณะเด่นดังนี้

- เป็นชิปเดี่ยวแบบ CMOS LSI สำหรับการบันทึกเสียงพูดและเรียกเสียงกลับ
- ประกอบด้วย RC ออสซิลเลเตอร์ (RC oscillator), บัสแอดเดรสและบัสข้อมูล (address/data bus), โหมดคอนโทรลเลอร์ (mode controller) และส่วนเดมอดูเลชัน (delta modulation)
- วงจรเดมอดูเลชันเป็นแบบความชันแปรเปลี่ยนต่อเนื่อง (Continuous Variable Slope Delta Modulator)

ซึ่งมีลักษณะการจัดขาและบล็อกไดอะแกรมดังรูป



รูปที่ 4.1 ลักษณะการจัดขาและบล็อกโคโตะแกรม

คำอธิบายขาต่างๆ

- V_{DD} - เป็นขาไฟเลี้ยงบวก
- V_{SS} - เป็นขากกราวด์วงจร
- A₀-A₁₄ - บัสแอดแตรส
- D₀-D₇ - บัสข้อมูล
- RECORD - เป็นขาอินพุต ทำงานที่โลจิก "0" ทริกขาขึ้นเมื่อต้องการบันทึกเสียง
- WRITE PULSE - เป็นขาเอาต์พุต ทำงานที่โลจิก "0"
- READ - เป็นขาเอาต์พุต ทำงานที่โลจิก "0" ขานี้จะให้สัญญาณควบคุมแก่หน่วยความจำภายนอก ขานี้จะทำงานเมื่ออยู่ในโหมดการเรียกเสียงกลับ
- PLAY - เป็นขาอินพุต ทำงานที่โลจิก "0" ทริกขาขึ้นเมื่อต้องการเรียกเสียงกลับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้ทำซ้ำโดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เคาท์เตอร์ภายใน (internal counter) และทำให้ชิพไม่ทำงาน

- ANG และ $\overline{\text{ANG}}$ - สัญญาณเสียงเอาต์พุตที่มีเฟสตรงกันข้าม
- FILTER - เป็นขาเอาต์พุต สร้างรูปคลื่นเอนVELOPE (envelope waveform) ด้วยวงจรอินทิเกรตภายใน
- ENVELOPE - เป็นขาอินพุต สัญญาณเอนVELOPEจะเข้าไปยังขานี้ เพื่อที่จะมอดคูเลตกับขนาดของเสียง
- TD และ $\overline{\text{TD}}$ - เป็นขาเอาต์พุตช่วยสำหรับความถี่ต่ำ
- COMPDATA - เป็นขาอินพุตสำหรับตรวจสอบอัตราการเปลี่ยนแปลงความชันซึ่งถูกรวบรวมจากการเปรียบเทียบระหว่างสัญญาณอินพุต และสัญญาณป้อนกลับ
- C, R - เป็นขา RC ออสซิลเลเตอร์ (RC oscillator)
- CLOCK DRIVER - เป็นขาเอาต์พุตสำหรับกำเนิดสัญญาณ

วงจรอัดเสียงใส่ RAM

เนื่องจากโครงงานชิ้นนี้ต้องการอัดเสียงให้ได้มากที่สุดประมาณ 1 นาที ดังนั้นจึงต้องทำการแปลงวงจรอัดเสียงของ UM5100 ให้ได้ตามที่ต้องการ ซึ่งผลที่ได้แสดงได้ดังรูปที่ 4.2

จากรูป 4.2 เมื่อสัญญาณเสียงผ่านทางไมโครโฟนเข้าสู่วงจรปริโมคซึ่งใช้อปแอมป์เบอร์ LM324 หลังจากนั้นสัญญาณที่ได้จะผ่านวงจรแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลซึ่งเป็นแบบเวิลด์ตามอดคูเลชันและสัญญาณจะเข้ามาเข้า UM5100 ตัว UM5100 จะทำการแปลงสัญญาณที่ได้เป็น 8 บิต และทำการเก็บใน RAM เบอร์ 62256 ซึ่งมีทั้งหมด 4 ตัว

สัญญาณควบคุมในการรีเซ็ต UM5100 ,อัดและแปลงเสียงกลับจะถูกควบคุมโดยพอร์ท C2H และสัญญาณในการเลือก RAM นั้นจะผ่านพอร์ท COH เข้าสู่ตัว 74LS138 ซึ่งเป็นไอซีที่ใช้เป็นตัวดีโคเดอร์ ส่วนขา A0 และ A14 เป็นขาเอาต์พุตของ UM5100 จะนำไปต่อกับพอร์ท C2H เพื่อผ่านกระบวนการทางซอฟต์แวร์ในการควบคุมการอัดว่า ต้องการสัญญาณเสียงนานเท่าไร

วงจรควบคุมการทำงานของเครื่องบันทึกลง EPROM

เป็นหน้าที่ทำหน้าที่ควบคุมการทำงานทั้งหมดของวงจรทั้งหมด โดยใช้ไมโครโปรเซสเซอร์ Z80 ควบคุม สำหรับวงจรในส่วนนี้จะประกอบด้วยส่วนสำคัญทั้งหมด 4 ส่วน คือ

1. ไมโครโปรเซสเซอร์ Z80 เป็นส่วนควบคุมหลัก
2. วงจรสร้างสัญญาณนาฬิกาที่ความถี่ 3.57 MHz โดยการใช้คริสตัล 3.57 MHz เป็นฐานเวลา
3. วงจรดีโค้ดหน่วยความจำ (Memory Decoder) และวงจรดีโค้ดพอร์ทโดยใช้ไอซี 74LS139
4. หน่วยความจำ (Memory) ซึ่งประกอบด้วย

- EPROM (Erasable Programmable Read Only Memory) ใช้สำหรับเก็บโปรแกรมหลักในการควบคุมการทำงานทั้งหมดของวงจร ในที่นี้ใช้ EPROM เบอร์ 27C64 ซึ่งมีหน่วยความจำ 8 กิโลไบต์
- + RAM (Random Access Memory) ใช้สำหรับเป็นบัฟเฟอร์สำหรับเก็บสถานะการทำงานและใช้เป็นสแต็ก (Stack) ของระบบ ในที่นี้ใช้ไอซีเบอร์ 6116 ซึ่งมีหน่วยความจำ 2 กิโลไบต์

วงจรที่ใช้งานเป็นดังรูปที่ 4.3

วงจรส่วนแสดงผลและสแกนคีย์

การทำงานในส่วนนี้จะมีเราใช้พอร์ท 2 พอร์ทในการควบคุมโดยจะมีพอร์ท C1H เป็นตัวควบคุมการแสดงผลและพอร์ท C2H จะเป็นส่วนควบคุมการสแกนคีย์และเลือก LED ที่จะแสดงผลโดยผ่านไอซีเบอร์ 74LS138 ซึ่งเป็นไอซีดีโค้ดเดอร์ (DECODER) โดย $D_0 - D_3$ ของ C2H จะต่อกับ 74LS138 และ $D_4 - D_7$ ของ C2H จะเป็นส่วนอินพุตของส่วนสแกนคีย์ ดังรูปที่ 4.4

วงจรส่วนควบคุมการอัด EPROM

ในส่วนนี้จะประกอบไปด้วย

1. ส่วนจ่ายไฟเลี้ยง ซึ่งใช้สำหรับในการอัด EPROM จะมีอยู่ 2 แบบคือ

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการให้ข้อมูลเท่านั้น ไม่สามารถนำข้อมูลไปใช้เพื่อการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของพอร์ท COH โดย ถ้าลอจิก (LOGIC) เป็น "1" จะเป็น 5 Volt และถ้าลอจิกเป็น "0" จะเป็น 12.75 Volt (ต่อกับขา V_{CC} ของ EPROM 27010)

- เปลี่ยนจาก 5 Volt เป็น 6.25 Volt ซึ่งควบคุมโดยใช้ขา D_0 ของพอร์ท COH เหมือนส่วนแรกโดยถ้าลอจิกเป็น "1" เป็น 5 Volt และถ้าลอจิกเป็น "0" จะเป็น 6.25 Volt (ต่อกับขา V_{CC} EPROM 27010)

2. ส่วนที่ใช้ในการอ้างขาแอดเดรส ขาข้อมูลและสัญญาณควบคุมต่างของ EPROM

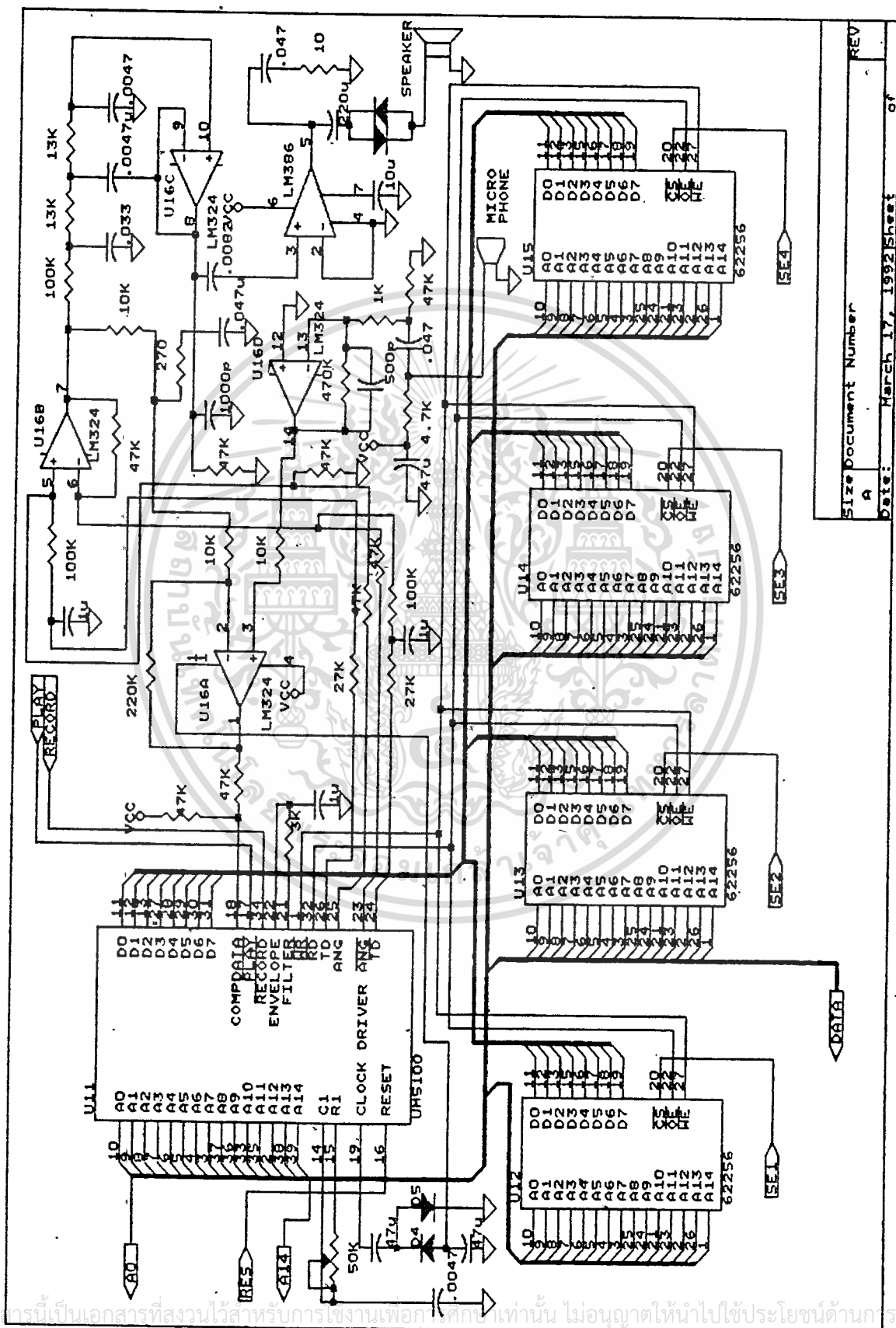
- ขาแอดเดรสของ EPROM 27010 มีทั้งหมด 17 ขา ในส่วนนี้ใช้พอร์ท 80H, 81H และขา D_0 ของพอร์ท COH เป็นตัวควบคุม
- ขาข้อมูลของ EPROM 27010 จะใช้ พอร์ท 82H เป็นตัวควบคุม
- ขาควบคุมต่างๆของ EPROM 27010 จะใช้ D_4-D_8 ของ พอร์ท 82H เป็นตัวควบคุมขา OE, CE และ PGM ตามลำดับ

ในส่วนนี้จะแสดงได้ดังในรูปที่ 4.5

4.1.2 ส่วนซอฟต์แวร์ สามารถเขียนเป็นผังการทำงานได้ดังรูปที่ 4.6

จากรูปสามารถอธิบายได้ดังนี้

เมื่อเปิดเครื่อง โปรแกรมจะทำการตรวจสอบ EPROM ว่าสามารถใช้ได้หรือไม่ ถ้าไม่แสดงคำว่า " ERROR " ออกมา ถ้าใช้ได้จะแสดงคำว่า " EPROM GOOD " จากนั้นจะให้ผู้ใช้ทำการป้อนเบอร์โทรศัพท์ ซึ่งสามารถป้อนได้ 3 หมายเลข แล้วบันทึกเสียงพูดลงไปในสแตติกแรม จากนั้นจะเรียกเสียงที่ถูกบันทึกไว้กลับมาให้ผู้บันทึกฟังเพื่อตรวจสอบว่าใช้ได้หรือไม่ ถ้าไม่ได้ก็จะให้บันทึกใหม่ ถ้าได้ก็จะทำการบันทึกโปรแกรมควบคุมต่างๆ เบอร์โทรศัพท์ สัญญาณเสียง ลงใน EPROM ต่อไป ระหว่างทำการบันทึก โปรแกรมจะทำการตรวจสอบ EPROM ไปด้วยพร้อมกัน

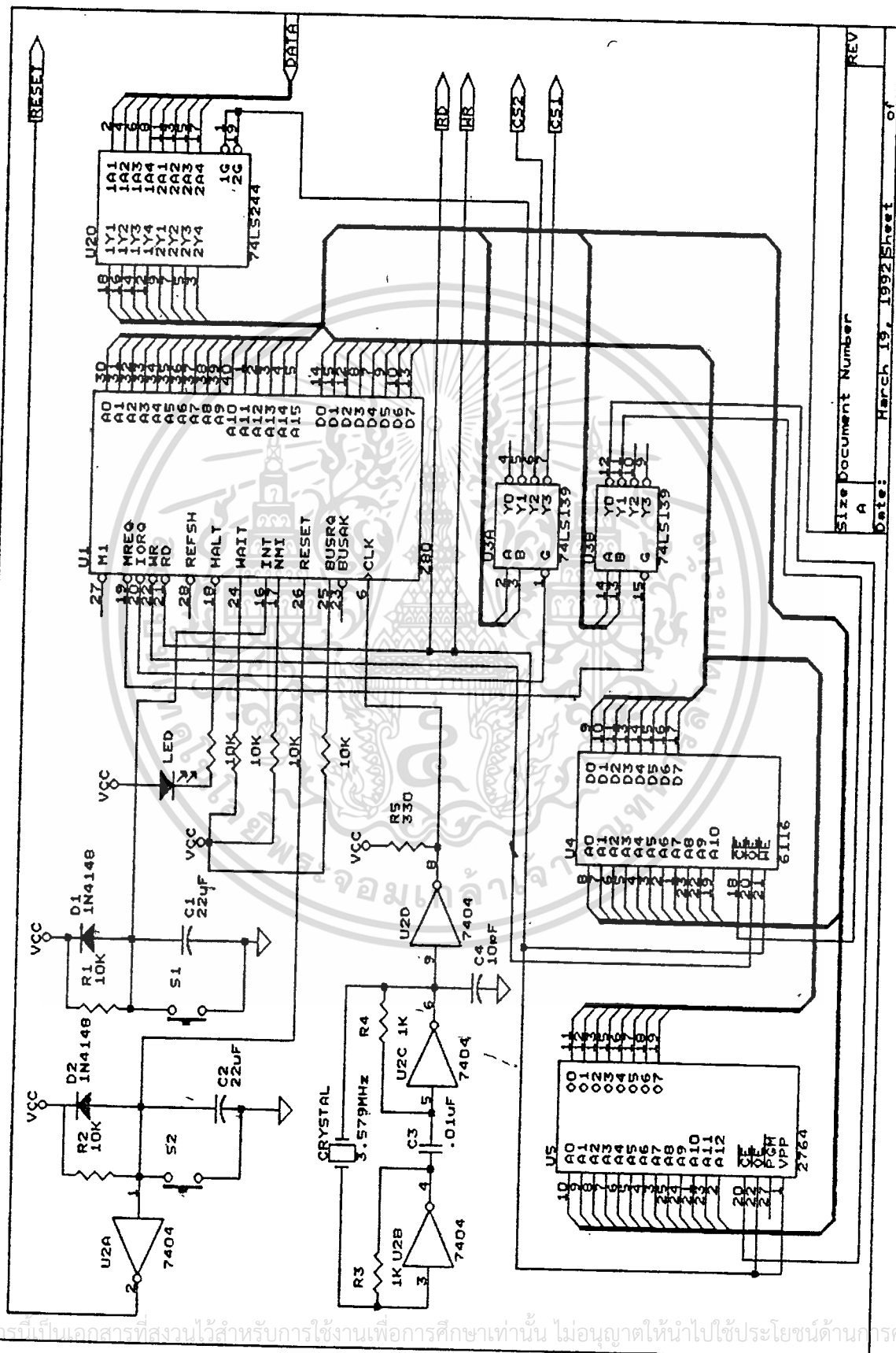


Size Document Number
A
Date: March 17, 1992 Sheet of REV

รูปที่ 4.2 วงจรต่อเสียงใส่ EPROM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

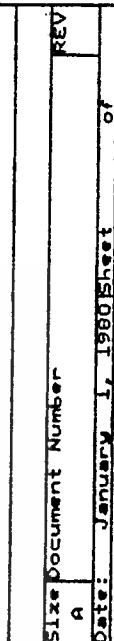


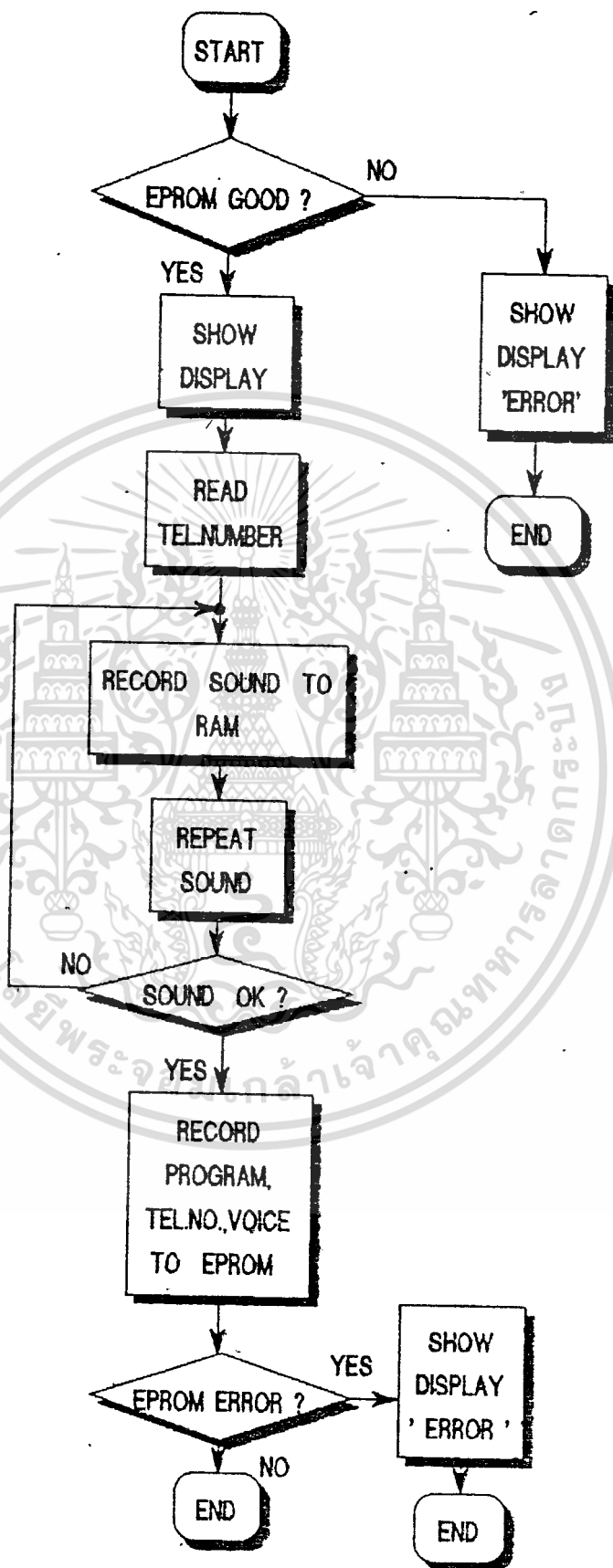
Size Document Number
A
Date: March 19, 1992 Sheet of REV

รูปที่ 4.3 วงจรควบคุมการทำงานของเครื่องบันทึก EPROM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 รูปที่ 4.6 ขั้นตอนการทำงานของส่วนซอฟต์แวร์ของวงจรส่วนบันทึกลง EPROM

4.2 วงจรเตือนภัยทางโทรศัพท์อัตโนมัติ

เราจะอธิบายได้เป็นสองส่วนคือ

4.2.1 ส่วนอาร์คแวร์

4.2.2 ส่วนซอฟต์แวร์

4.2.1 ส่วนอาร์คแวร์ สามารถแบ่งได้ดังนี้

วงจรหมุนเบอร์โทรศัพท์

วงจรที่ใช้จะใช้ไอซี เบอร์ TCM 5087 ซึ่งไอซีเบอร์นี้เป็นโทเนดีโคเดอร์ (tone decoder) ทำหน้าที่ในการหมุนเบอร์โทรศัพท์แบบใช้ความถี่คู่ การหมุนเบอร์ทำได้โดยการใส่ข้อมูลที่เป็นลอจิก "0" หรือ "1" เข้าที่ขา R_1-R_4 และ C_1-C_4 ของไอซี โดยขา R_1-R_4 จะทำงาน (active) ที่ลอจิก "0" ส่วน C_1-C_4 จะทำงาน ที่ "1" ในการหมุนเบอร์ต่างๆจะให้ขา R_1-R_4 หรือ C_1-C_4 ทำงาน ดังตารางที่ 4.1

	C_1	C_2	C_3
R_1	1	2	3
R_2	4	5	6
R_3	7	8	9
R_4	*	0	#

ตารางที่ 4.1 ลักษณะของเบอร์โทรศัพท์

ตัวอย่างเช่น ถ้าต้องการหมุนเบอร์ 1 ก็จะต้องส่งข้อมูลให้ R_1 เป็น "0" ส่วน R_2-R_4 เป็น "1" และ C_1 เป็น "1" ส่วน C_2-C_4 เป็น "0" เป็นต้น ซึ่งเบอร์แต่ละเบอร์จะต้องส่งข้อมูลให้ไอซี ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

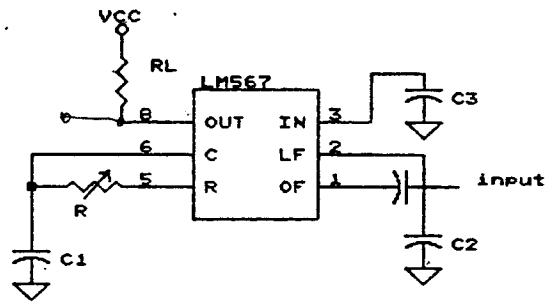
เบอร์	R_1	R_2	R_3	R_4	C_1	C_2	C_3	C_4	เลขฐานสิบ
1	0	1	1	1	1	0	0	0	120
2	0	1	1	1	0	1	0	0	116
3	0	1	1	1	0	0	1	0	114
4	1	0	1	1	1	0	0	0	184
5	1	0	1	1	0	1	0	0	180
6	1	0	1	1	0	0	1	0	178
7	1	1	0	1	1	0	0	0	216
8	1	1	0	1	0	1	0	0	212
9	1	1	0	1	0	0	1	0	210
0	1	1	1	0	0	1	0	0	228
สภาวะปกติ	1	1	1	1	0	0	0	0	240

ตารางที่ 4.2 ข้อมูลที่ต้องส่งให้อีซีในการหมุนเบอร์

วงจรตรวจสอบสัญญาณจากสายโทรศัพท์

ในการตรวจจับสัญญาณจากสายโทรศัพท์ว่าเป็น สัญญาณให้หมุน(dial tone) ,สัญญาณไม่ว่าง(busy tone) หรือ สัญญาณเรียกกลับ(ring back tone) นั้น เนื่องจากสัญญาณทั้ง 3 นี้ เป็นสัญญาณที่มีความถี่ 400 Hzเหมือนกันหมด ต่างกันเพียงช่วงเวลาที่มีความถี่ 400 Hz ปรากฏอยู่เท่านั้น เราจึงใช้อีซีเบอร์ NE 567 ซึ่งเป็นไอซีเฟสล็อก-ลูป(phase lock loop) โดยตั้งความถี่ศูนย์กลาง (f_c) ให้ได้ 400 Hz ซึ่งอยู่กับการตั้งค่าของ C และ R ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 การต่อวงจรของ NE 567

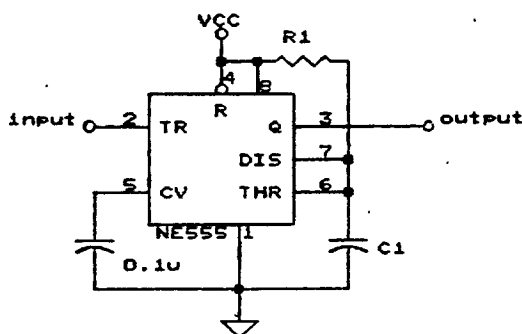
หลักการคำนวณของ NE 567

- ขั้น 1 กำหนดค่า $f_o = 400 \text{ Hz}$
- ขั้น 2 หา R_1, C_1 จาก $f_o = 1.1/R_1 C_1$
- ขั้น 3 หา C_2 จาก $C_2 = 130/f_o \text{ } \mu\text{F}$
- ขั้น 4 หา C_3 จาก $C_3 > 2C_2 > 260/f_o \text{ } \mu\text{F}$

การทำงานของ NE 567 นั้นที่ขาเอาต์พุต(output) ปกติจะเป็น "1" แต่ถ้ามีสัญญาณอินพุต(input) 0 ที่มีความถี่ตรงกับความถี่ศูนย์กลางเอาต์พุต(output) จะตกลงเป็น "0" เมื่อสัญญาณที่ตรงกับความถี่ศูนย์กลางนี้หมดไป เอาต์พุต(output) จะกลับเป็น "1" ดังเดิม

วงจรโมโนสเตเบิล

เนื่องจากสัญญาณที่ออกจากเอาต์พุตของ NE 567 อาจมีการกระเพื่อมเนื่องจากสัญญาณรบกวนภายในสายโทรศัพท์ ดังนั้นเราจึงใช้ไอซีเบอร์ 555 ซึ่งเป็นไอซีโมโนสเตเบิล (monostable) โดยปกติเอาต์พุต จะเป็น "0" แต่เมื่อมีสัญญาณที่เป็น "0" มาเข้าที่ขาอินพุต จะทำให้ค่าของเอาต์พุต เปลี่ยนเป็น "1" เป็นเวลานานตามที่หน่วงเวลาเอาไว้ ซึ่งเวลาที่หน่วงจะขึ้นอยู่กับค่า R และ C โดยที่ $T = 1.1 R_1 C_1$ วินาที ดังรูป



รูปที่ 4.8 การต่อวงจรของ 555

วงจรเตือนภัยทางโทรศัพท์อัตโนมัติ

วงจรนี้เป็นวงจรที่รวมเอาวงจรหมุนเบอร์โทรศัพท์ วงจรตรวจสอบสัญญาณจากสายโทรศัพท์ วงจรโมโนสเตเบิลและวงจรเรียกเสียงกลับ มาใช้งาน โดยจะมีไมโครโปรเซสเซอร์ 8031 เป็นตัวควบคุม

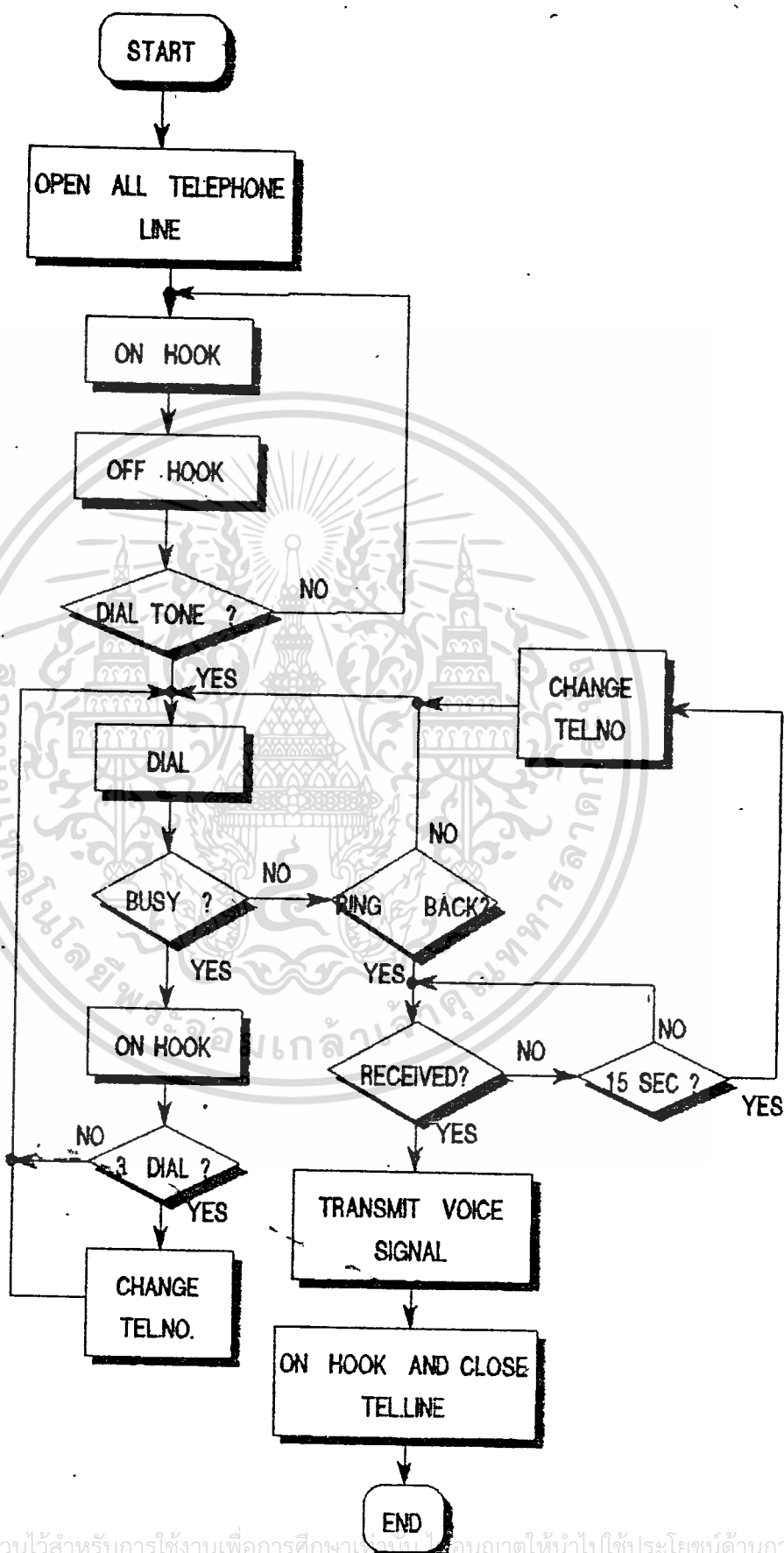
ตอนแรก 8031 จะส่งสัญญาณออกทาง P3.0 ไปยังรีเลย์เพื่อทำการตัดโทรศัพท์ออก และจะมี P3.1 เป็นตัวส่งสัญญาณไปยังรีเลย์อีกตัวหนึ่งซึ่งเป็นตัวควบคุมการยกหู วางหู เมื่อยกหูแล้ว จะใช้ P3.2 ตรวจสอบสัญญาณจากเอาต์พุตของ 555 ว่ามีสัญญาณให้หมุนหรือยัง ถ้ามีแล้วก็นำข้อมูลซึ่งเป็นเบอร์โทรศัพท์จาก EPROM 27010 ออกทาง P1 ไปเพื่อเป็นอินพุตของ TCM 5087 ให้ทำการหมุนเบอร์ แล้วจะตรวจสอบทาง P3.2 ว่าเป็นสัญญาณไม่ว่างหรือสัญญาณเรียกกลับ โดยจะมี UM 5100 เป็นตัวส่งสัญญาณเสียงออกไปทางสายโทรศัพท์เมื่อทราบว่ามีการรับโทรศัพท์แล้ว ในการส่งสัญญาณเสียงออกไปนั้นเนื่องจาก UM 5100 สามารถที่จะออสซิลเลตได้ด้วยตัวเองดังนั้นจึงไม่สามารถที่จะต่อแอตเตนเชอร์กับ EPROM 27010 ได้โดยตรงเพราะ EPROM 27010 นี้จะแอตเตนเชอร์ได้ต่อกับ 8031 แล้วซึ่งถ้าออสซิลเลตโดยตรงจะทำให้แอตเตนเชอร์ของ 8031 กับ UM 5100 เกิดการชนกัน ดังนั้นจึงใช้วิธีส่งข้อมูลของสัญญาณเสียงโดยการให้ 8031 ตรวจสอบขา A0 ของ UM 5100 โดยเมื่อมีการเปลี่ยนแปลงของขา A0 ทุกครั้ง 8031 จะทำการรับข้อมูลสัญญาณเสียงจาก EPROM 27010 แล้วส่งผ่านพอร์ต P1 ไปยังขา D0-D7 ของ UM 5100 เพื่อให้ส่งสัญญาณเสียงออกไป

4.2.2 ส่วนซอฟต์แวร์ สามารถเขียนเป็นผังการทำงานได้ดังรูปที่ 4.9

ตอนแรกโปรแกรมจะสั่งให้อาร์ตเวิร์กตัดสายโทรศัพท์ทั้งหมดออก เนื่องจากว่าอาจมีผู้ใช้โทรศัพท์อยู่ แล้วจึงทำการวางหูโทรศัพท์เพื่อให้ชุมสายโทรศัพท์ส่งสัญญาณให้หมุนมา หลังจากนั้นก็จะทำการยกหูแล้วตรวจสอบว่ามีสัญญาณให้หมุนมาหรือยัง ถ้าไม่มีก็จะวางหูแล้วยกหูใหม่ ถ้ามีจะทำการหมุนเบอร์โทรศัพท์แล้วตรวจสอบว่าเป็นสัญญาณไม่ว่าง หรือสัญญาณเรียกกลับ ถ้าเป็นสัญญาณไม่ว่างก็จะวางหู แล้วยกหูหมุนใหม่ จนกระทั่งครบ 3 ครั้ง ถ้ายังไม่มีคนรับจะเปลี่ยนเบอร์หมุน

ถ้าเป็นสัญญาณเรียกกลับ โปรแกรมจะตรวจสอบว่าภายใน 15 วินาที ว่ามีผู้รับสายหรือยัง ถ้ายังจะหมุนใหม่ ถ้ามีจึงทำการส่งสัญญาณเสียงออกไป

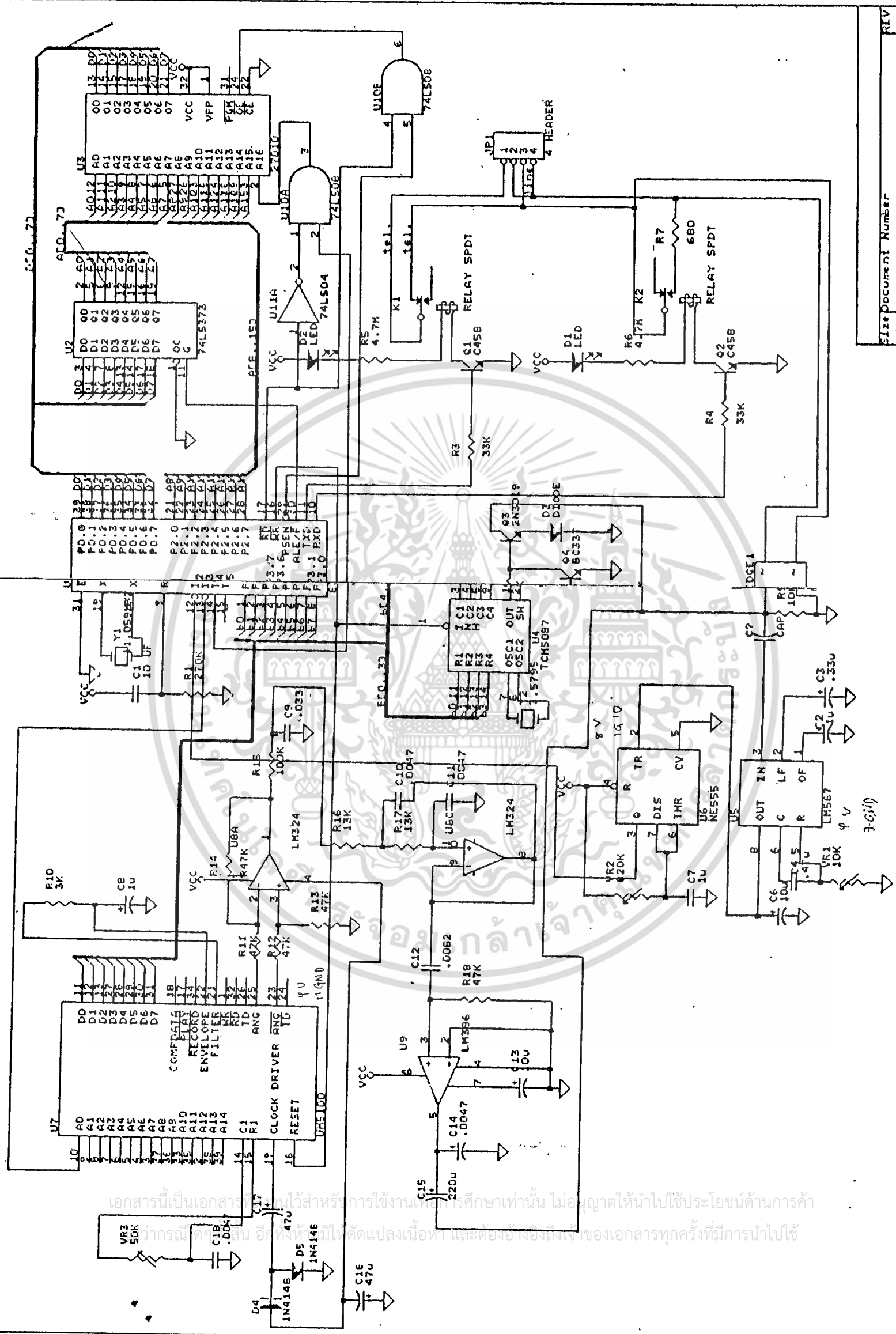




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.9 ผังการทำงานลวนของหน่วยควบคุมของวงจรเตือนภัยทางโทรศัพท์อัตโนมัติ



เอกสารนี้เป็นเอกสารที่... ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า...
เอกสารนี้... ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า...
เอกสารนี้... ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า...

รูปที่ 4.10 วงจรเตือนภัยทางโทรศัพท์โมด

Size	Document Number	REV
B		
Date:	March 19, 1992	Sheet 1 of 1

บทที่ 5

การทดลองและผลการทดลอง

จะแยกกล่าวเป็นสองส่วนคือ

5.1 วงจรส่วนบันทึกใส่ EPROM

5.2 วงจรเตือนภัยทางโทรศัพท์อัตโนมัติ

5.1 วงจรส่วนบันทึกใส่ EPROM

ในส่วนนี้ได้ทำการทดลองต่อวงจรและทดสอบส่วนต่างๆ โดยใช้ Z80 เป็นตัวควบคุม ได้ผลดังนี้

วงจรสำหรับบันทึกเสียงนั้นใช้อัตราการสุ่มสัญญาณ 16 kHz เสียงที่ได้สามารถฟังได้ชัดเจนพอสมควร ทั้งนี้ถ้าแหล่งจ่ายไฟเลี้ยงมีริปเปิ้ล (Ripple) มากเสียงที่ได้ออกมา ก็จะมีเสียงซ่าออกมาด้วยทำให้คุณภาพของเสียงลดลง

ในส่วนบันทึกใส่ EPROM จะมีการแบ่งออกเป็นโปรแกรมและเบอร์โทรศัพท์กับสัญญาณเสียงที่แปลงเสร็จแล้ว ในส่วนที่บันทึกโปรแกรมและเบอร์โทรศัพท์นั้นสามารถที่จะบันทึกลง EPROM เข้าๆ ได้ถ้าข้อมูลยังไม่ถูกต้อง แต่ในส่วนของสัญญาณเสียงนั้นเราไม่สามารถจะบันทึกเข้าได้เพราะติดปัญหาจากการที่เราไม่สามารถควบคุมการเล่นกลับของสัญญาณเสียงได้ ดังนั้นในการใช้งานเราจึงต้องใช้ EPROM ที่มีคุณภาพดีหรือผ่านการบันทึกมาไม่ก็ครั้ง

5.2 วงจรส่วนเตือนภัยทางโทรศัพท์อัตโนมัติ

ในส่วนนี้ได้ทำการทดลองต่อวงจรและทดสอบส่วนต่างๆ โดยใช้ 8031 เป็นตัวควบคุม ได้ผลดังนี้

สำหรับส่วนนี้ได้นำ EPROM จากส่วนที่แล้วมาใช้ โดยลองให้เครื่องหมุนเบอร์ที่ได้ถอดไว้ แล้วทำการตรวจสอบสัญญาณภายในคู่สายโทรศัพท์ดู ซึ่งถ้าเครื่องตรวจสอบได้ว่าเป็นสัญญาณไม่ว่างเครื่องจะหมุนเบอร์ใหม่ให้ แต่ถ้ามีคนรับเครื่องก็จะส่งสัญญาณเสียงที่ได้บันทึกไว้แล้วออกไป ในส่วนนี้ไอซีเบอร์ LM 567 ซึ่งเป็นตัวตรวจจับสัญญาณในคู่สายโทรศัพท์ เป็นไอซีที่ทำการควบคุมได้ยาก ดังนั้นในการที่จะให้ตรวจสอบสัญญาณได้ตรงตามที่ต้องการ จึงต้องใช้ค่า R และ C ที่เหมาะสม จากการทดลองส่วนนี้ได้ผลตามที่ต้องการ ไม่ว่าจะเป็นกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

บทสรุปและวิจารณ์

จากการพัฒนาโครงงานนี้มาทั้งหมด ทำให้สามารถสรุปคุณสมบัติหลักในการทำงานของ "เครื่องโทรศัพท์เตือนภัยอัตโนมัติโดยใช้ EPROM เก็บเสียง" ได้ดังนี้

1. มีส่วนที่ใช้ในการบันทึกโปรแกรม หมายเลขโทรศัพท์และสัญญาณเสียงลง EPROM โดยทั้งหมดนี้ควบคุมโดยไมโครโปรเซสเซอร์เบอร์ Z80 และแสดงผลโดย 7 เซกเมนต์ (7 SEGMENT) 8 หลัก รวมทั้งใช้ปูมในการควบคุมและรับหมายเลขโทรศัพท์
2. สามารถที่จะเปลี่ยนสัญญาณเสียงและเบอร์โทรศัพท์ให้เป็นที่พอใจก่อนทำการบันทึกลงใน EPROM ได้ด้วยตัวเอง
3. มีระบบที่สามารถทำการหมุนเบอร์โทรศัพท์และทำการเปลี่ยนเบอร์โทรศัพท์ให้โดยอัตโนมัติ ถ้าสถานที่ที่ต้องการส่งสัญญาณเสียงไปถึงนั้นติดต่อไม่ได้
4. สามารถที่จะตรวจสอบสัญญาณต่างๆภายในคู่สายโทรศัพท์ได้

แนวทางประยุกต์ใช้งานโครงงาน

- สามารถที่จะนำเอาส่วนที่เป็นส่วนบันทึกเสียงใส่ EPROM นั้นไปใช้ ในโรงงานอุตสาหกรรมที่ต้องการใช้เสียงในการควบคุมการสั่งงานเพื่อให้ง่ายต่อความเข้าใจของผู้ถูกใช้งาน หรือนำไปใช้ในระบบรักษาความปลอดภัยโดยการเตือนจะเป็นเสียงที่บันทึกไว้แล้ว

เอกสารอ้างอิง

1. Lance, A. Leventhal Winthrop Saville, " Z80 ASSEMBLY LANGUAGE SUBROUTINES "
2. " Memory Components Handbook ", int. 1
3. " Linear Circuits Data Book " , Texas Instruments
4. " Embedded Controller Handbook Volme 1 8 Bit " , int. 1
5. " Microprocessor and Peripheral Handbook Volume II - Peripheral " , int. 1
6. " Programming the Z80 THIRD EDITION " , RODNAAY ZAKS
7. " SEMICONDUCTOR DATA BOOK HD74LS SERIES " , HITACHI
8. " Analog DATA Manual 1983 " Signetics
9. " เครื่องบริกวรลอบถามข้อมูลทางโทรศัพท์อัตโนมัติ " , ปรินญาณินพณ์สำหรับวิศวกรรมศาสตรบัณฑิต ภาควิชาอิเล็กทรอนิกส์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง , 2533
10. " คู่มือ/เทียบเบอร์ไอซี TTL " , ซีเอ็ด

กิตติกรรมประกาศ

โครงการเครื่องโทรศัพท์เตือนภัยอัตโนมัติโดยใช้ EPROM เก็บเสียง และปริณิพจน์เล่มนี้สำเร็จไปได้ด้วยดีนั้น เนื่องจากการสนับสนุนช่วยเหลือจากหลายๆฝ่ายด้วยกันโดยการให้ข้อมูล อุปกรณ์ ซอฟแวร์สนับสนุน และคำปรึกษาต่างๆ ซึ่งเป็นประโยชน์อย่างมากต่อการดำเนินงาน

ขอขอบคุณอาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์พลดวง ผดุงกุล ที่ได้กรุณาให้คำแนะนำปรึกษา และอำนวยความสะดวกในด้านอุปกรณ์เครื่องมือและสถานที่ทดลองวิจัยเป็นอย่างดี

ขอขอบคุณเพื่อนๆทุกคนที่ได้ช่วยเหลือ อีกทั้งพี่ปริณิญาโทที่ให้คำปรึกษาและให้ยืมอุปกรณ์ ทำให้โครงการนี้สำเร็จได้ด้วยดี

ผู้จัดทำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ORG 0000H

PORT0: EQU 0C0H
PORT1: EQU 0C1H
PORT2: EQU 0C2H
PORT3: EQU 0C3H
PORT4: EQU 080H
PORT5: EQU 081H
PORT6: EQU 082H
PORT7: EQU 083H
PORT51: EQU 040H

STACK: EQU 4700H
NUMBER: EQU 4090H ;1 ADDRESS
NUMBKEY: EQU 40A0H
NUM1: EQU 40A1H
NUM2: EQU 40A2H
NUM3: EQU 40A3H

FIXDIS: EQU 4000H ;20H ADDRESS (DISPLAY)
BCSAVE: EQU 4500H ;2 ADDRESS (TRANFER)
DESAVE: EQU 4502H ;2 ADDRESS (TRANFER)
LOOPDIS: EQU 4504H ;2 ADDRESS (DISPLAY)
ASAVE: EQU 4506H ;1 ADDRESS (DISPLAY2)
CHKTEL: EQU 4520H

DISTEST: EQU 4300H ;1 ADDRESS
SAVEDA7: EQU 4370H ;2 ADDRESS (DISPLAY)
SAVEDA9: EQU 43A0H

TELENUM: EQU 4050H
RAMNUM: EQU 4360H
INTSTOP: EQU 4362H
NUMAO: EQU 4364H ; NUMBER OF AO

CHKEY1: EQU 4508H ;1 ADDRESS (CHECK KEY)
CHKEY2: EQU 4509H ;1 ADDRESS (CHECK KEY)

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EXX
CALL    NOINT
LD      A,A0H          ;RESET UM5100
OUT      (PORT0),A
CALL    DELAY1
LD      HL,INTSTOP
LD      (HL),01H
CALL    DELAY1
EX      AF,AF'
EXX
EI
RET

```

RUN:

```

XOR     A
INC     L
LD      (HL),A
INC     L
LD      (HL),A
INC     L
LD      (HL),A
LD      HL,DATARUN
LD      BC,10H
LD      (BCSAVE),BC
LD      DE,FIXDIS
LD      (DESAVE),DE
CALL    TRANSFER

```

RUN1:

```

START:  LD      A,20H
START1: LD      (ASAVE),A
        LD      BC,0800H
        LD      HL,FIXDIS

```

```

START2: LD      A,(HL)
        OUT     (PORT1),A
        LD      A,C
        OUT     (PORT2),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IN      A,(PORT2)
OR      OCOH
LD      E,A
CP      OFOH
JR      C,START5
LD      A,(CHKEY2)
CP      C
JR      NZ,START3
LD      A,E
LD      (CHKEY1),A
START3: CALL    DELAY1
XOR      A
OUT      (PORT1),A
START4:  INC      L
INC      C
DJNZ     START2
LD      A,(ASAVE)
DEC      A
JR      NZ,START1
JR      START
START5:  CALL     TEST
CALL     GOOD
LD      IY,DATA7
LD      (SAVEDA7),IY
LD      IY,DATA9
LD      (SAVEDA9),IY
LD      IY,DATA3
CALL     TRAN
CALL     DISPLAY
LD      HL,NUMBER      ;4090H  TELEPHONE NUMBER ?
LD      (HL),03H
XOR      A
LD      L,A
LD      A,(IY+03H)
LD      (HL),A
LD      HL,DESAVE
INC      (HL)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น. ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ; >>>>> SCAN KEYBOARD <<<<<<
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SCAN1: LD A,20H
SCAN1: LD (ASAVE),A
LD BC,0800H
LD H,40H

LD L,C ;HL = 4000H

SCAN2: LD A,(HL)
OUT (PORT1),A
LD A,C
OUT (PORT2),A

IN A,(PORT2)
OR OCOH
LD E,A
CP OFOH
JR C,KEY
LD A,(CHKEY2)
CP C
JR NZ,SCAN3
LD A,E
LD (CHKEY1),A
SCAN3: CALL DELAY1
XOR A
OUT (PORT1),A
SCAN4: INC L
INC C
DJNZ SCAN2
LD A,(ASAVE)
DEC A
JR NZ,SCAN1
JR SCAN

```

```

; %%%% FUNCTION KEY %%%%

```

```

KEY: PUSH BC
PUSH HL
CALL DELAY1
LD A,C

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      (CHKEY2),A
LD      A,(CHKEY1)
CP      E
JR      Z,RET
LD      A,E
LD      (CHKEY1),A
CP      0E2H
JP      NC,FUNCTION
CALL    YESFUNC
LD      L,90H
INC      (HL)
DEC      (HL)
JR      Z,RET

```

NUMB:

```

LD      L,A0H ;NUMBKEY 40A0H
LD      L,(HL)
LD      A,07H
CP      (HL)
JR      Z,RET
INC      (HL)
LD      A,E
SUB      0DOH
LD      HL,DATA1
ADD      A,L
LD      L,A
LD      E,(HL)
ADD      A,20H
LD      L,A
LD      D,(HL)
LD      (IX+0),D
INC      IX
LD      HL,4008H
LD      (HL),E
LD      L,02H
LD      BC,07H
CALL    TRANSFER

```

RET:

```

POP      HL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FUNCTION: JP Z,DEL
LD A,E
CP OE4H
JR Z,SPEAK
JP NC,RESOUND
CALL YESFUNC
LD L,90H
DEC (HL)
JP Z,SOUNDF

LD L,A0H
INC (HL)

CALL TRAN

INC IY
LD A,(IY+00H)
LD (400CH),A

CALL DISPLAY

XOR A
LD L,A
LD A,(IY+03H)
LD (HL),A
LD HL,DESAVE
INC (HL)
JP RET

NOINT: LD HL,NOFUN
INC (HL)
DEC (HL)
RET NZ
POP BC
POP BC

JP RET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NOFUNC:      LD      HL,NOFUN
              INC      (HL)
              DEC      (HL)
              RET      NZ
              POP      BC
              JP      RET

```

```

YESFUNC:     PUSH    HL
              LD      HL,NOFUN
              INC      (HL)
              DEC      (HL)
              POP      HL
              RET      Z
              POP      BC
              JP      RET

```

```

;          $$$$$$ RECORD SOUND TO RAM $$$$$$

```

```

SPEAK:       CALL    NOFUNC
              LD      A,AOH          ;RES 5100
              OUT     (PORT0),A
              CALL    DELAY1
              LD      DE,02FFH
              LD      HL,INTSTOP     ;INTERUPT TO STOP(4162H)
              XOR     A
              LD      (HL),A
              LD      H,43H
              LD      HL,NUMAO       ;NUMBER OF AO (4364H)
              LD      (HL),A
              LD      L,60H
              LD      (HL),A        ;NUMBER RAM 1 2 3 4(4360H)

              OUT     (PORT2),A     ;USE RAM 1
              OUT     (PORT0),A     ;RECORD
              LD      A,80H
              OUT     (PORT0),A     ;CLEAR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น;CHECK INTERRUPT TO STOP
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

IN      A,(PORT2)
RLA
CALL    C,ADDAO      ;TEST A0
RLA      ;TEST A14
JP      NC,SPEAK1
SPEAK2: CALL    STOPS
IN      A,(PORT2)
RLA
CALL    C,ADDAO
RLA
JP      NC,SPEAK3
JR      SPEAK2

```

```

SPEAK3: LD      L,60H
INC      (HL)
LD      A,(HL)
OUT      (PORT2),A
CP      04H
JR      NZ,SPEAK1
LD      A,A0H      ;RES 5100
OUT      (PORT0),A
CALL    DELAY1
JP      RET
; LD      SP,STACK
; JP      SCAN
;
;      &&&&&& REPEAT SOUND &&&&&&

```

```

RESOUND: CALL    NOFUNC
LD      A,E
CP      0E6H
JP      Z,RECPRO

JP      NC,TEST
LD      HL,NUMAO      ;NUMBER OF A0
LD      C,(HL)
LD      IY,4160H
LD      DE,02FFH

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      A,A0H                ;RES  5100
OUT     (PORT0),A
CALL    DELAY1
XOR     A                    ;USE RAM 1
OUT     (PORT2),A
LD      B,A
LD      A,COH                ;PLAY SOUND
OUT     (PORT0),A
CALL    DELAY1
LD      A,80H                ;CLEAR CONTROL
OUT     (PORT0),A
OUT     (PORT1),A            ;SHOW DISPLAY

RESOUND1: CALL STOP3          ;TEST INTERRUPT TO STOP
IN      A,(PORT2)
RLA
CALL    C,DECA0              ;TEST A0
RLA                      ;TEST A14
JP      NC,RESOUND1

RESOUND2: CALL STOP3          ;TEST INTERRUPT TO STOP
IN      A,(PORT2)
RLA
CALL    C,DECA0              ;TEST A0
RLA
JP      NC,RESOUND3

JR      RESOUND2

RESOUND3: LD      L,60H        ;NUMBER RAM(1 2 3 4)RAMNUM
INC     B
LD      A,B
OUT     (PORT2),A
DEC     A
CP      (HL)
JR      C,RESOUND1
LD      A,A0H                ;RESET UM5100
OUT     (PORT0),A
CALL    DELAY1
JP      RET
LD      SP,STACK

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

; @@@@@@ TEST EPROM @@@@@@

```

TEST:      LD      HL,DISTEST
            LD      (HL),10H
            LD      B,08H
            LD      A,01H
            OUT     (PORT1),A
            LD      DE,00H

            LD      A,89H          ;CONTROL PORT 8255 (IN DATA)
            OUT     (PORT7),A

TEST1:     LD      A,2CH          ;Z,VCC>LOW      (OC)
            OUT     (PORT0),A

            LD      A,E
            OUT     (PORT4),A
            LD      A,D
            OUT     (PORT5),A
            CALL    SHOW

            LD      A,24H          ;READ DATA ,VCC>LOW (04H)
            OUT     (PORT0),A
            IN      A,(PORT6)      ;READ DATA

            INC     A
            JR      NZ,ERROR
            INC     DE
            LD      A,D
            OR      E
            JR      NZ,TEST1

            LD      (HL),10H
            LD      B,08H
            LD      DE,00H

```

```

TEST2:      LD      A,2DH                ;Z,VCC>LOW (2DH)
            OUT      (PORT0),A

            LD      A,E
            OUT      (PORT4),A
            LD      A,D
            OUT      (PORT5),A
            CALL     SHOW
            LD      A,25H                ;READ DATA (05H)
            OUT      (PORT0),A
            IN       A,(PORT6)          ;READ DATA
            INC      A
            JR       NZ,ERROR
            INC      DE
            LD      A,D
            OR       E
            JR       NZ,TEST2
            RET

;          ^^^^^^  DISPLAY BETWEEN RECORD SOUND TO RAM  ^^^^^^

SHOW:       CP       (HL)
            RET      C
            LD      A,(HL)
            ADD     A,10H
            LD      (HL),A
            LD      A,B
            OUT      (PORT2),A
            DEC     B
            RET     NZ
            LD      B,08H
            RET

;          ++++++  DISPLAY IF EPROM IS GOOD  ++++++

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

GOOD:      LD      HL,DATA6
            LD      BC,14H
            LD      (BCSAVE),BC
            LD      DE,FIXDIS
            LD      (DESAVE),DE
            CALL    TRANFER
            LD      DE,OBOOH          ; 10H
            LD      (LOOPDIS),DE
GOOD1:     LD      DE,(LOOPDIS)      ; 1 PICTURE 13 LOOP
            LD      H,40H
            LD      A,07H
            CALL    DISPLAY2
            LD      A,05H
            CP      D
            CALL    Z,DELAY2
            DEC     D
            JR      NZ,GOOD2
            LD      A,00H
            OUT     (PORT1),A
            RET
GOOD2:     LD      HL,BCSAVE
            DEC     (HL)
            LD      C,(HL)
            INC     L
            LD      B,(HL)
            INC     E
            LD      (LOOPDIS),DE
            LD      H,40H
            CALL    TRANFER
            JR      GOOD1

```

; ++++++ DISPLAY IF EPRAM IS ERROR ++++++

```

ERROR:     LD      HL,DATA5
            LD      BC,10H
            LD      (BCSAVE),BC
            LD      DE,FIXDIS

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      (DESAVE),DE
CALL    TRANSFER
LD      DE,0700H
LD      (LOOPDIS),DE
ERROR1: LD      DE,(LOOPDIS)      ; 1 PICTURE 13 LOOP
LD      H,40H
LD      A,05H
CALL    DISPLAY2
DEC     D
JR      NZ,ERROR2
LD      SP,STACK
JP      SCAN

```

```

ERROR2: LD      HL,BCSAVE
DEC     (HL)
LD      C,(HL)
INC     L
LD      B,(HL)
INC     E
LD      (LOOPDIS),DE
LD      H,40H
CALL    TRANSFER
JR      ERROR1

```

```

;      ; ; ; ; TELEPHONE NUMBER IS WRONG ; ; ; ;

```

```

DEL:    CALL    YESFUNC
LD      L,90H
INC     (HL)
DEC     (HL)
JP      Z,RET

LD      L,A0H
LD      L,(HL)
DEC     (HL)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังไม่มีให้ตัดแบลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      L,07H
LD      BC,07H
LD      DE,4008H
LDDR
LD      L,01H
LD      (HL),00H
JP      RET

```

```

;          ##### RECORD PROGRAM TO RAM #####

```

```

SETIN:   EQU      89H
SETOUT:  EQU      80H
NOTDATA: EQU      A8H          ;OE>H VCC>H CE>L PGM>H
PRODATA: EQU      AAH
READATA: EQU      A0H

RECPRO:  LD      1Y,0900H
        LD      HL,00H
        LD      C,20H
        LD      DE,0300H      ;E > NUMBER OF DATA
RECPRO1: LD      A,SETOUT      ;DATA TO EPROM (80)
        OUT     (PORT7),A
        LD      A,(1Y+0)
        OUT     (PORT6),A
        LD      A,L
        OUT     (PORT4),A
        LD      A,H
        OUT     (PORT5),A
        LD      A,NOTDATA      ;OE>H VCC>H CE>L PGM>H
        OUT     (PORT0),A

        LD      A,PRODATA      ;PGM>L(1)
        OUT     (PORT0),A
        LD      B,14H
RECPRO2: NOP
        DJNZ    RECPRO2
        LD      A,NOTDATA
        OUT     (PORT0),A      ;PGM>H(1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

NOP
LD      A,SETIN          ;PORT C > IN
OUT     (PORT7),A
LD      A,L
OUT     (PORT4),A
LD      A,H
OUT     (PORT5),A
LD      A,READATA       ;OE>L(1) [ VCC>H CE>L 1(3)
OUT     (PORT0),A
IN      A,(PORT6)
CP      (IY+0)
JP      NZ,RECPR03
LD      A,NOTDATA
OUT     (PORT0),A       ;PGM>H(1)
LD      C,20H
INC     IY
INC     HL
DEC     DE
LD      A,D
OR      E
JP      NZ,RECPR01
LD      A,NOTDATA
OUT     (PORT0),A       ;PGM>H(1)
LD      A,SETIN
OUT     (PORT7),A

```

TEL1:

```

LD      IY,NUM1
LD      HL,0800H
LD      E,01H
CALL    RECTEL

```

```

DEC     IY
INC     (IY+00H)
DEC     (IY+00H)
JR      Z,TEL2

```

```

LD      HL,NUM1
LD      E,(HL)

```

```

LD      IY,TELENUM
LD      HL,0801H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CALL RECTEL
PUSH IY

TEL2: LD IY,NUM2
LD HL,0810H
LD E,01H
CALL RECTEL

DEC IY
INC (IY+00H)
DEC (IY+00H)
JR Z,TEL3

LD HL,NUM2
LD E,(HL)
POP IY
LD HL,0811H
CALL RECTEL
PUSH IY

TEL3: LD IY,NUM3
LD HL,0820H
LD E,01H
CALL RECTEL

LD HL,NUM3
LD E,(HL)
POP IY
LD HL,0821H

CALL RECTEL

TIME: LD IY,NUMAO
LD HL,0830H
LD E,01H
CALL RECTEL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL    TRANPGM
CALL    DISPGM
JR      DUMPSO
;       JP      PSO

RECPR03: LD      A,NOTDATA
OUT      (PORT0),A      ;PGM>H(1)
DEC      C
JP      NZ,RECPR01
JP      ERROR

RECTEL:  LD      A,SETOUT      ;DATA TO EPROM (80)
OUT      (PORT7),A
LD      A,(1Y+0)
OUT      (PORT6),A
LD      A,L
OUT      (PORT4),A
LD      A,H
OUT      (PORT5),A
LD      A,NOTDATA      ;OE>H VCC>H CE>L PGM>H
OUT      (PORT0),A
LD      A,PRODATA      ;PGM>L(1)
OUT      (PORT0),A
LD      B,14H

RECTEL2: NOP
DJNZ     RECTEL2
LD      A,NOTDATA
OUT      (PORT0),A      ;PGM>H(1)
NOP
LD      A,SETIN      ;PORT C > 1N
OUT      (PORT7),A
LD      A,L
OUT      (PORT4),A
LD      A,H
OUT      (PORT5),A
LD      A,READATA      ;OE>L(1) [ VCC>H CE>L ](3)
OUT      (PORT0),A
IN      A,(PORT6)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CP      (1Y+0)
JR      NZ,RECTEL3
LD      A,NOTDATA
OUT     (PORT0),A      ;PGM>H(1)
LD      C,20H
INC     IY
INC     HL
DEC     E
JR      NZ,RECTEL
LD      A,NOTDATA
OUT     (PORT0),A      ;PGM>H(1)
LD      A,SETIN
OUT     (PORT7),A
RET
RECTEL3: LD      A,NOTDATA
OUT     (PORT0),A      ;PGM>H(1)
DEC     C
JR      NZ,RECTEL
JP      ERROR
; ##### RECORD SOUND TO EPROM #####
CLEAR   EQU      4366H
PGML    EQU      4368H
READ    EQU      436AH
PAGE    EQU      436CH
DUMPSO: LD      A,88H
LD      (CLEAR),A      ;CLEAR CONTROL
LD      A,8AH
LD      (PGML),A      ;PGM > L
LD      A,80H
LD      (READ),A
LD      A,00H
LD      (PAGE),A
LD      A,08H
OUT     (PORT2),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      1Y,RANNUM      ;NUMBER OF RAM
SET     3,(1Y+0)      ; +08H
LD      A,(1Y+4)      ;NUMBER OF AO (4364H)
LD      (1Y+5),A
XOR     A
LD      (1Y+15H),A
LD      HL,0900H

```

```

LD      A,A8H          ;RES 5100 ,0E>H
OUT     (PORT0),A
CALL    DELAY1

```

```

LD      DE,02FFH
LD      A,08H
OUT     (PORT2),A
LD      (1Y+13H),A      ;NUMBER RAM RECORD
LD      A,C8H          ;PLAY SOUND
OUT     (PORT0),A
CALL    DELAY1
LD      A,(CLEAR)      ;CLEAR CONTROL
OUT     (PORT0),A
LD      A,FFH
OUT     (PORT1),A      ;SHOW DISPLAY

```

DUMPO:

```

;
CALL    STOP3          ;TEST INTERUPT TO STOP
IN      A,(PORT2)
RLA
CALL    C,DUMPAO      ;TEST AO

```

```

CALL    EPROM

```

DUMP1:

```

IN      A,(PORT2)
RLA
JR      NC,DUMP1

```

```

RLA          ;TEST A14
JP      NC,DUMPO

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
DUMP2: IN A,(PORT2)
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        RLA
        CALL    C,DUMPA0          ;TEST A0

        CALL    EPROM

DUMP3:   IN      A,(PORT2)
        RLA
        JR      NC,DUMP3

        RLA
        JP      NC,DUMP4
        CALL    STOP3            ;TEST INTERRUPT TO STOP
        JR      DUMP2

DUMPA:   INC      (1Y+13H)        ;NUMBER RAM(1 2 3 4)
        LD      A,(1Y+13H)
        OUT     (PORT2),A
        DEC     A
        CP      (1Y+0)
        JR      C,DUMPO
        LD      A,ACH            ;RESET UM5100
        OUT     (PORT0),A
        CALL    DELAY1
        LD      SP,STACK
        JP      SCAN

EPROM:   LD      C,08H            ;02H
EPROM1:  LD      A,SETOUT         ;DATA TO EPROM (80)
        OUT     (PORT7),A
        IN      A,(PORT51)       ;DATA TO CPU
        OUT     (PORT6),A
        LD      A,L
        OUT     (PORT4),A
        LD      A,H
        OUT     (PORT5),A
        LD      A,(CLEAR)        ;OE>H VCC>H CE>L PGM>H
        OUT     (PORT0),A
        LD      A,(PGML)         ;PGM>L(1)
        OUT     (PORT0),A
        LD      B,19H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EPROM2:

```
NOP
DJNZ     EPROM2
LD       A,(CLEAR)
OUT      (PORT0),A      ;PGM>H(1)
NOP
LD       A,SETIN        ;PORT C > IN
OUT      (PORT7),A
LD       A,L
OUT      (PORT4),A
LD       A,H
OUT      (PORT5),A
LD       A,(READ)       ;OE>L(1) [ VCC>H CE>L ](3)
OUT      (PORT0),A
IN       A,(PORT6)
LD       B,A
IN       A,(PORT51)
CP       B
JR       NZ,EPROM4
```

EPROM3:

```
LD       A,(CLEAR)
OUT      (PORT0),A      ;PGM>H(1)
LD       C,08H
INC      HL
LD       A,H
OR       L
RET      NZ
LD       HL,PAGE
LD       A,(HL)
AND      A
JR       Z,CNGPAGE
LD       A,FFH
OUT      (PORT1),A
LD       A,05H
OUT      (PORT2),A
HALT
```

CNGPAGE:

```
INC      (HL)
LD       L,66H
INC      (HL)
LD       L,68H
INC      (HL)
LD       L,6AH
```

```

INC      (HL)
LD       HL,00H
RET

EPROM4:  LD      A,(CLEAR)
          OUT     (PORT0),A      ;PGM>H(1)
          DEC     C
          JR      NZ,EPROM1
          INC     (1Y+15H)
          LD      A,FFH
          CP      (1Y+15H)
          JR      NZ,EPROM3
          JR      PSO
          JP      ERROR

;        \\\\\\\ CHEAK TIME RECORD SOUND TO EPROM  /////

DUMPAO:  CALL     EPROM
DUMPA:   IN       A,(PORT2)
          RLA
          JR      C,DUMPA
          DEC     E
          RET     NZ
          DEC     E
          DEC     D
          RET     NZ
          LD      D,02H
          DEC     (1Y+5H)
          RET     NZ
          LD      A,ACH
          OUT     (PORT0),A
          CALL    DELAY1

          JP      SOUND

```

```

;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

```

```

PSO: LD DE, FIXDIS

```

```

LD A,H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      (DE),A
INC     E
LD      A,L
LD      (DE),A
INC     E
LD      A,89H
OUT     (PORT7),A

```

```

TST1:   LD      A,0CH                ;Z,VCC>LOW
OUT     (PORT0),A
LD      A,L
OUT     (PORT4),A
LD      A,H
OUT     (PORT5),A
LD      A,04H                ;READ DATA ,VCC>LOW
OUT     (PORT0),A
IN      A,(PORT6)            ;READ DATA
LD      (DE),A
INC     HL
INC     DE
LD      A,E
CP      50H
JR      NZ,TST1

;~~~~~
POP     HL
LD      SP,STACK
JP      SCAN

```

```

;          !!!!! PROGRAM SCAN DISPLAY !!!!!

```

```

DISPGM: LD      DE,(LOOPDIS)        ; 1 PICTURE 13 LOOP
LD      H,40H
LD      A,05H
CALL    DISPLAY2
LD      A,13H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      A,0AH
CP      D
CALL    Z,DELAY2
DEC     D
JR      NZ,DISPGM1
RET

```

```

DISPGM1: LD      HL,BCSAVE
DEC      (HL)
LD      C,(HL)
INC     L
LD      B,(HL)
INC     E
LD      (LOOPDIS),DE
LD      H,40H
CALL    TRNFER
JR      DISPGM

```

```

DISPLAY: LD      DE,(LOOPDIS) ; 1 PICTURE 13 LOOP
LD      H,40H
LD      A,05H
CALL    DISPLAY2
LD      A,0EH
CP      D
CALL    Z,DELAY2
LD      A,07H
CP      D
CALL    Z,DELAY2
DEC     D
JR      NZ,DISPLAY1
RET

```

```

DISPLAY1: LD      HL,BCSAVE
DEC      (HL)
LD      C,(HL)
INC     L
LD      B,(HL)
INC     E

```

```

LD      (LOOPDIS),DE
LD      H,40H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CALL TRANSFER
JR DISPLAY

DISPLAY2: LD (ASAVE),A
 LD BC,0800H
 LD L,C ;HL = 4000H

DISPLAY3: LD A,(HL)
 OUT (PORT1),A
 INC L
 LD A,C
 OUT (PORT2),A
 INC C
 CALL DELAY1
 DJNZ DISPLAY3
 LD A,(ASAVE)
 DEC A
 JR NZ,DISPLAY2
 RET

TRANPGM: LD HL,DATA8
 LD BC,20H
 LD (BCSAVE),BC
 LD DE,FIXDIS
 LD (DESAVE),DE
 CALL TRANSFER
 LD DE,1500H ; 1 PICTURE 13 LOOP
 LD (LOOPDIS),DE
 RET

TRAN: LD HL,DATA2
TRAN1: LD BC,20H
 LD (BCSAVE),BC
 LD DE,FIXDIS
 LD (DESAVE),DE
 CALL TRANSFER

LD DE,1500H ; 1 PICTURE 13 LOOP

LD (LOOPDIS),DE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RET

```
TRANSFER:      LD      DE,(DESAVE)      ; GET BC,HL
                LDIR                      ; DATA TO DISPLAY
                RET
```

; &&&&&& PROGRAM DELAY &&&&&&

```
DELAY1:         PUSH    BC
                LD      BC,0AAH
DEL1:           DEC     BC
                LD      A,B
                OR      C
                JR      NZ,DEL1
                POP     BC
                RET
```

```
DELAY2:         LD      A,60H
                DEC     D
                CALL    DISPLAY2
                RET
```

```
DELAY3:         PUSH    BC
                LD      BC,2AH
                JR      DEL1
```

; " " " " " DISPLAY :SOUND: " " " " "

```
SOUNDF:         LD      (HL),03H
                LD      IY,DATA3
                LD      HL,NOFUN
                LD      (HL),01H
```

```
SOUND:          LD      HL,DATA4
                LD      BC,10H
                LD      (BCSAVE),BC
```

```
                LD      DE,FIXDIS
                LD      (DESAVE),DE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

CALL    TRANSFER
LD      DE,0700H
LD      (LOOPDIS),DE
SOUND1: LD      DE,(LOOPDIS)      ; 1 PICTURE 13 LOOP
LD      H,40H
LD      A,05H
CALL    DISPLAY2
DEC      D
JR      NZ,SOUND2
JP      RET

```

```

SOUND2: LD      HL,BCSAVE
DEC      (HL)
LD      C,(HL)
INC      L
LD      B,(HL)
INC      E
LD      (LOOPDIS),DE
LD      H,40H
CALL    TRANSFER
JR      SOUND1

```

<<<<<< CHEAK TIME RECORD SOUND >>>>>>

```

ADDAO:  IN      A,(PORT2)
RLA
JR      C,ADDAO
DEC      E
RET      NZ
DEC      E
DEC      D
RET      NZ
LD      HL,NUMAO
INC      (HL)
LD      B,A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

SEGMENT:      PUSH      IY
               LD        IY,(SAVEDA7)
               LD        A,(IY+00H)
               INC       IY
               OUT       (PORT1),A
               CP        3AH
               JR        NZ,SEGMENT1
               LD        IY,DATA7

```

```

SEGMENT1:     LD        (SAVEDA7),IY
               POP       IY
               LD        D,02H
               LD        A,B
               RET

```

```

;      ===== CHECK TIME REPEAT SOUND =====

```

```

DECAO:        IN        A,(PORT2)
               RLA
               JR        C,DECAO
               DEC       E
               RET       NZ
               DEC       E
               DEC       D
               RET       NZ

```

```

EX           AF,AF'

```

```

SEGMENT2      PUSH      IY
               LD        IY,(SAVEDA9)
               LD        A,(IY+00H)
               INC       IY
               OUT       (PORT1),A
               CP        90H
               JR        NZ,SEGMENT3
               LD        IY,DATA9

```

```

SEGMENT3:     LD        (SAVEDA9),IY
               LD        D,02H

```

```

LD      A,AOH
OUT     (PORT0),A
CALL    DELAY1
LD      SP,STACK
JP      SCAN
SEGMENT4:
EX      AF,AF'
POP     IY
RET

```

```

;      ?????? CHEAK INTERUPT TO STOP SOUND  ??????

```

```

STOPS:  PUSH    HL
        LD      HL,INTSTOP
        INC     (HL)
        DEC     (HL)
        JR      Z,STOP6
        LD      A,AOH
        OUT     (PORT0),A
        CALL    DELAY1
        LD      (HL),OOH
        LD      SP,STACK

```

```

;      POP     HL
        JP      SCAN

```

```

STOP6:  POP     HL
        RET

```

```

STOP3:  PUSH    HL
        LD      HL,INTSTOP
        INC     (HL)
        DEC     (HL)
        JR      Z,STOP4
        LD      (HL),OOH

```

```

STOP:   LD      A,AOH      ;RESET UM5100

```

```

        OUT     (PORT0),A
        CALL    DELAY1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

LD      SP,STACK
JP      SCAN
STOP4   POP    HL
RET

;      [##### DATA SHOW DISPLAY ]]]]]]

```

```

ORG      0800H

```

```

DATA1:   DEFB      FEH,EOH,BEH,B6H,66H,F2H,DAH,60H
          DEFB      00H,00H,00H,00H,00H,00H,00H,00H
          DEFB      F6H,FCH,00H,00H,00H,00H,00H,00H
          DEFB      00H,00H,00H,00H,00H,00H,00H,00H
          DEFB      2BH,1BH,4DH,2DH,1DH,4EH,2EH,1EH
          DEFB      00H,00H,00H,00H,00H,00H,00H,00H
          DEFB      4BH,27H

```

```

DATA2:   DEFB      00H,00H,00H,00H,00H,00H,00H,00H
          DEFB      1EH,9EH,1CH,00H,60H,00H,00H,7AH
          DEFB      EEH,1EH,EEH,00H,00H,00H,00H,00H
          DEFB      00H,00H

```

```

DATA3:   DEFB      00H,DAH,F2H,61H,0DBH,0F3H

```

```

DATA4:   DEFB      00H,00H,00H,00H,00H,00H,00H,00H
          DEFB      B6H,3AH,38H,2AH,7AH,00H,00H,00H

```

```

DATA5:   DEFB      00H,00H,00H,00H,00H,00H,00H,00H
          DEFB      9EH,0AH,0AH,3AH,0AH,00H,00H,00H

```

```

DATA6:   DEFB      00H,00H,00H,00H,00H,00H,00H,00H
          DEFB      F6H,3AH,3AH,7AH,00H,00H,00H,00H
          DEFB      00H,00H

```

```

DATA7:   DEFB      C6H,3AH

```

```

,DATA7:   DEFB      80H,40H,20H,10H,08H,04H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
DATA8: DEFB 00H,00H,CEH,0AH,3AH,F6H,0AH,EEH
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DEFB 2AH,22H,00H,00H,F6H,8AH,3AH,7AH
DEFB 00H,00H,00H,00H,00H,00H,00H,00H

DATA9: DEFB 80H,82H,12H,90H
DATARUN: DEFB 00H,00H,0AH,9EH,EEH,7AH,76H,00H
;DATA9: DEFB 80H,82H,92H
END
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ORG    0000H

LJMP   SPLIT

; interrupt service routine

ORG    000BH
PUSH   PSW
MOV     PSW,#00H
INC     R2
POP     PSW
RETI

ORG    0030H
SPLIT: MOV   R2,#00H           ;register for number of interrup
MOV     P3,#0COH
MOV     P1,#0FH
MOV     PSW,#00H             ;main use register bank0
MOV     R0,#04H

;MAIN PROGRAM

MAIN:  LCALL  CHK_N
       LCALL  HOOK
       LCALL  DIAL
       LCALL  RESPONSE
       CJNE   R3,#01H,MAIN
       LCALL  RING_BACK
       CJNE   R3,#01H,MAIN
       lcall  out_snd
       LCALL  RECEIVER
       CJNE   R3,#01H,MAIN
       LCALL  OUT_SND
       SJMP   $

```

;procedure hook

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

HOOK:  SETB  P3.5
        LCALL DELAY                                ;register for tel.number
        SETB  P3.0
        LCALL DELAY
NHOOK:  CLR   P3.1                                ;off hook
        LCALL DELAY
        SETB  P3.1                                ;on hook
        LCALL DELAY
        SETB  P3.2
        LCALL DELAY
        JB    P3.2,out_hook
        SJMP  NHOOK
OUT_HOOK:  RET

```

```

;procedure chk_n
CHK_N:  NOP
CHK_NO:  DEC  RO
        CJNE  RO,#03H,CHK_2
        MOV   DPTR,#0820H
        SJMP  NON_ZERO
CHK_2:  CJNE  RO,#02H,CHK_1
        MOV   DPTR,#0810H
        SJMP  NON_ZERO
CHK_1:  CJNE  RO,#01H,CHK_0
        MOV   DPTR,#0800H
CHK_0:  MOV   RO,#04H
        SJMP  CHK_NO
NON_ZERO:  CLR  P1.0
        RET

```

```

;procedure dial
DIAL:  MOV   PSW,#10H                                ;DIAL USE REGISTER BANK2
        MOVX  A,@DPTR
        MOV   RO,A
TEL_NO:  INC  DPTR
        LCALL DELAY

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีเหตุที่เปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LCALL DELAY
MOV    P1,#0FH
DJNZ   R0,TEL_NO
MOV    PSW,#00H
RET
```

```
;procedure response
```

```
RESPONSE: LCALL RST_TMR
```

```
NOL: SETB P3.2
```

```
lcall delay
```

```
JB     P3.2,HAVE
```

```
CJNE   R2,#45H,NOL
```

```
MOV    R3,#00H
```

```
SJMP   NONE
```

```
HAVE:  MOV    R3,#01H
```

```
NONE:  RET
```

```
;procedure ring_back
```

```
RING_BACK: LCALL RST_TMR
```

```
RING: LCALL CHK_SIGN
```

```
CJNE   R2,#13H,R_TEST
```

```
R_TEST: JC     RING
```

```
CJNE   R1,#01H,kk
```

```
kk: jnc  busy
```

```
MOV    R3,#01H
```

```
SJMP   OUT_RING
```

```
BUSY:  MOV    R3,#00H
```

```
OUT_RING: RET
```

```
;procedure receiver
```

```
RECEIVER: LCALL CHK_SIGN
```

```
CJNE   R2,#59H,S6_TEST
```

```
S6_TEST: JC     RECEIVER
```

```
CJNE   R1,#01H,READY
```

```
DETE_7S: LCALL CHK_SIGN
```

```
CJNE   R2,#9FH,S7_TEST
```

```
S7_TEST: JC     DETE_7S
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        CJNE    R1,#05H,READY
DETE_12S: LCALL  CHK_SIGN
        CJNE    R2,#0E4H,S12_TEST
S12_TEST: JC     DETE_12S
        CJNE    R1,#07H,READY
        NO: MOV   R3,#00H
        SJMP    OUT_RE
        READY: MOV R3,#01H
        OUT_RE: RET

```

```

;procedure restart timer

```

```

RST_TMR: CLR    IE.1                                ;ITTEROUTINE USE REGISTER BANK0
        CLR    IE.7
        MOV    R2,#00H
        CLR    TCON.4
        MOV    TMOD,#01H
        MOV    TH0,#00H
        MOV    TLO,#00H
        SETB   IE.7
        SETB   IE.1
        SETB   TCON.4
        RET

```

```

;procedure check signal

```

```

CHK_SIGN: setb   p3.2
        jb     p3.2,start1
        mov    r0,#00h
        sjmp   chk_re
start1:  mov    r0,#01h
CHK_RE:  MOV    R7,#00H
        MOV    R6,#00H
        MOV    R4,#4FH
CHK_RE0: MOV    R5,#OFFH
CHK_RE1: SETB   P3.2
        JNB    P3.2,NON1
        CJNE   R6,#OFFH,LOOP
        MOV    R6,#00H
        INC    R7

```

```

        SJMP    NON1
LOOP:   INC     R6
NON1:   DJNZ    R5,CHK_RE1
        DJNZ    R4,CHK_RE0
        CJNE    R7,#15H,OK1
OK1:    JNC     OVER
        CJNE    R7,#01H,OK2
OK2:    JC      UNDER
        SJMP    NON2
UNDER:  CPL     P3.4
        CLR     C
        SJMP    MUA
OVER:   SETB    C
        CPL     P3.3
MUA:    RLC     A
        XRL     A,R0
        RRC     A
        JNC     NON2
        INC     R1
        MOV     A,R0
        RRC     A
        CPL     C
        RLC     A
        MOV     R0,A
NON2:   NOP
        RET

```

```

;procedure    delay

```

```

DELAY:  MOV     PSW,#10H      ;use register bank2
        MOV     R0,#03H
DELAY0: MOV     R1,#OFFH
DELAY1: MOV     R2,#OFFH
DELAY2: NOP
        DJNZ    R2,DELAY2
        DJNZ    R1,DELAY1
        DJNZ    R0,DELAY0
        CLR     PSW.4
        RET

```

```
OUT_SND:  CLR    P3.6
           MOV    A,#55H
XX:        RR     A
           mov    p1,a
           SJMP   XX
           RET
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้