



ปีการศึกษา 2533

เครื่องบริการสอบถามข้อมูลทางโทรศัพท์อัตโนมัติ

(Automatic Tele-Information Machine)

โดย

1. นายณัฐพล มงคลประดิษฐ์ 30.1073
2. นายพัลลภศิริ ศิริวิมลวารณ 30.1338
3. นายอนุรักษ์ คณิตพูนเกียรติ 30.1340

อาจารย์ที่ปรึกษา

ผศ. พลผดุง ผดุงกุล

ปริญญานิพนธ์ปีการศึกษา 2533

ภาควิชาอิเล็กทรอนิกส์

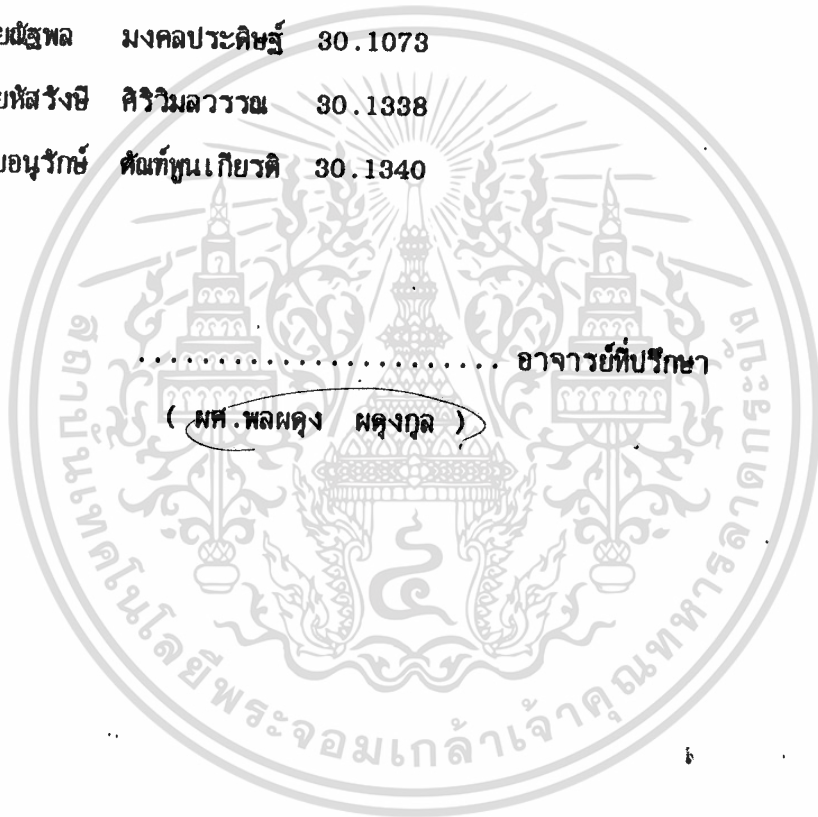
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องบริการสอบถามข้อมูลทางโทรศัพท์อัตโนมัติ

(Automatic Tele-Information Machine)

ผู้จัดทำ

1. นายณัฐพล มงคลประดิษฐ์ 30.1073
2. นายหัสรังษิ ศิริวิมลวราชน 30.1338
3. นายอนุรักษ์ ศันท์ทูนเกียรติ 30.1340



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

028747

## สารบัญ

	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎี	3
2.1 เทคโนโลยีคอมพิวเตอร์	3
2.2 วงจรกรองความถี่สัญญาณ	9
2.3 ทฤษฎีและการใช้งาน Z80 PIO	16
บทที่ 3 หลักการทำงาน	25
บทที่ 4 รายละเอียดการออกแบบและการทำงาน	28
4.1 ส่วนฮาร์ดแวร์	28
4.1.1 วงจรสำหรับการติดต่อกับโทรศัพท์	29
4.1.2 วงจรแปลงสัญญาณระหว่างสัญญาณเสียงกับสัญญาณดิจิทัล	41
4.1.3 วงจรส่วนที่ใช้สำหรับการติดต่อระหว่าง Z80 กับ IBM PC	54
4.1.4 วงจรส่วนควบคุมการทำงาน	60
4.2 ส่วนซอฟต์แวร์	62
4.2.1 ซอฟต์แวร์บนเครื่อง IBM PC	62
4.2.2 ซอฟต์แวร์บน Z80	74
บทที่ 5 การใช้งานโปรแกรมเทเลอินฟอร์เมชัน เวอร์ชัน 1.0	83
บทที่ 6 บทสรุปและวิจารณ์	98
กิตติกรรมประกาศ	100
หนังสืออ้างอิง	101
ภาคผนวก	102

เครื่องบริการสอบถามข้อมูลทางโทรศัพท์อัตโนมัติ  
(Automatic Tele-Information Machine)

ผศ.พล มงคลประดิษฐ์ รหัส 30.1073

หัสรังษิ ศิริวิมลวราณ รหัส 30.1338

อนุรักษ์ ตันทัพนเกียรติ รหัส 30.1340

ผศ.พลผดุง ผดุงกุล อาจารย์ที่ปรึกษา  
ปีการศึกษา 2533

บทคัดย่อ

โครงการเครื่องบริการสอบถามข้อมูลทางโทรศัพท์อัตโนมัติ มาจากแนวความคิดที่จะพัฒนาอุปกรณ์สำหรับนำไปใช้ในการให้บริการสอบถามข้อมูลต่างๆแก่ผู้ใช้ โดยผ่านการให้บริการทางโทรศัพท์ ซึ่งผู้ใช้สามารถสอบถามข้อมูลภายในชุดข้อมูลใดก็ได้ที่ต้องการทราบ โดยการกดรหัสประจำชุดข้อมูลตามตัวรหัสของข้อมูลที่ต้องการทราบ, จากนั้นเครื่องบริการก็จะทำการค้นหาข้อมูล แล้วตอบเป็นเสียงพูดออกทางคู่สายโทรศัพท์ อย่างอัตโนมัติ

โครงการนี้ จะใช้เครื่องไมโครคอมพิวเตอร์ IBM PC เป็นส่วนควบคุมหลัก โดยทำงานร่วมกับไมโครโปรเซสเซอร์ Z80 ซึ่งจะใช้เป็นตัวควบคุมวงจรมานอก โดยใช้การอินเทอร์เฟสแบบขนาน สำหรับโปรแกรมที่ใช้งานบนเครื่อง IBM PC จะเป็นส่วนที่ใช้งานการควบคุมการทำงานของวงจร ใช้บันทึกเสียงพูด รวมทั้งเป็นส่วนบันทึกและค้นหาข้อมูลที่ต้องการทราบ โดยแสดงผลในระบบเมนู ซึ่งง่ายต่อการใช้งานของพนักงานประจำเครื่อง

สำหรับโครงการที่เสร็จเรียบร้อยแล้วนี้ สามารถทำงานได้ตามจุดประสงค์ที่วางไว้ และโครงการนี้ สามารถนำมาพัฒนาต่อไปได้อีกหลายอย่าง เช่น พัฒนาให้สามารถใช้งานกับจำนวนคู่สายที่มากขึ้น หรือ พัฒนาให้ผู้ใช้สามารถทำการฝากข้อความไว้ได้ เป็นต้น

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการแข่งขันเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ขออนุญาตจากคณะกรรมการ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## AUTOMATIC TELE-INFORMATION MACHINE

Nattapol Mongkolpradit No. 30.1073

Hussarungsri Siriwimonwan No. 30.1338

Anurak Tanpoonkiat No. 30.1340

Asst.Prof.Polphadung Phadungkul advisor

Academic Year 1990

### Abstract

Automatic Tele-Information Machine project has come from the idea to develop equipments and respond to inquire variety information service through via telephone service. The user can inquire the data in the computer by press password for each particular data set that the user would like to know. After that, the machine will search the data correspond with code. When it find out, the machine will reply data in voice signal to the user via telephone line by automatically.

In this project, we use IBM personal computer in order to be main controller and work correlate with microprocessor Z80 (outside IBM PC circuitry main controller). By use interfacing in parallel form. The program on IBM PC is used to control record voice circuitry work and search the data that inquired. The user can use by facility since display on the screen in menu system form. Everybody can easily use it.

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Present, this project is complete. The machine can work and pass objective that we desire. If the machine will be developed or expanded next in the future. It can be wildspread such as development to use multiple telephone line with only one computer simultaneously reply or development for the user so that deposit the massages, for the persons that the user wants them to know in the future, in the computer.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในชีวิตประจำวันเราจะพบว่าระบบการสื่อสารมีความสำคัญอย่างยิ่ง ทำให้การติดต่อถึงกันไม่ว่าในแง่ของธุรกิจ การถามข่าวสาร การโทรศัพท์ใต้อตามความสุขสบายของกันและกัน และอื่น ๆ อีกมากมาย ล้วนก่อให้เกิดประโยชน์ทั้งสิ้น ที่จุดที่มีความเจริญทางการสื่อสารมาก ที่นั่นก็จะมีชื่อเสียงก้าวหน้าของบ้านเมืองมากตามไปด้วย

ในสมัยก่อนยุคเริ่มแรกการจะไปติดต่อถามหาข้อมูลจะเป็นในลักษณะของการไปติดต่อกับบุคคลที่ทราบเรื่องนั้นๆ หรือจากหนังสือที่ให้ความรู้ในเรื่องนั้น ต่อมาเมื่อมีความเจริญทางการสื่อสารโทรคมนาคมมากขึ้น การติดต่อกับเครื่องมือต่างๆ เช่น โทรศัพท์ โทรเลข โทรสาร ฯลฯ เริ่มมีบทบาทสำคัญ และยิ่งในปัจจุบันนี้ประชากรของโลกได้เพิ่มมากขึ้น การติดต่อสื่อสารถึงกันอาจมีปัญหาก็ได้ เช่น การโทรศัพท์ไปหาคนที่เราต้องการติดต่อกับแต่ปรากฏว่าเขาไม่อยู่ หรือคนที่รู้บอกเล่าให้อีกคนช่วยตอบให้โดยตนเองจะไปหาธุระ ซึ่งในเงื่อนไขนี้คนรับฝากข่าวสารอาจรับข่าวสารผิดๆ ถูกๆ ก็ได้

เนื่องจากความต้องการของคนในปัจจุบันมีความต้องการหลักในการสื่อสารคือ

1. ความรวดเร็ว และความสะดวกสบายในการติดต่อ
2. สามารถรับทราบข่าวสารได้ตลอดเวลาที่ต้องการ ไม่ว่าจะกลางวันหรือกลางคืน
3. ข่าวสารที่เที่ยงตรงถูกต้องแม่นยำ
4. สิ้นเปลืองค่าใช้จ่ายในการถามข่าวสารน้อยที่สุด

จากสิ่งทีวิเคราะห์มาทั้งหมดนี้เอง ทำให้คณะผู้จัดทำมีความคิดริเริ่มที่จะช่วยในทุกข้อตามที่กล่าวมาบรรลุดูประสงค์ให้มากที่สุด โดยสังเกตเห็นถึงการสื่อสารในโลกปัจจุบันเป็นหลัก ซึ่งมีแนวความคิดดังนี้

1. ใช้คอมพิวเตอร์ทำการเก็บข้อมูลเสียงพูดต่างๆ ไว้ในหน่วยความจำ ฮาร์ดดิสค์

ฟลอปปีดิสค์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. บุคคลที่ต้องการข้อมูลเหล่านี้สามารถมาดูได้หรือมาใช้บริการคอมพิวเตอร์ หรือโทรศัพท์ เข้ามาหาคอมพิวเตอร์
3. เนื่องจากคอมพิวเตอร์มีข้อมูลหลายชนิดที่จะให้ถาม ผู้ที่ต้องการติดต่อจะต้องกรรหส์ที่แนบโทรศัพท์เป็นการบอกว่าต้องการข้อมูลอะไร คอมพิวเตอร์จะทำการประมวลผล และส่งข้อมูลเสียงที่ต้องการไปให้ผู้ถาม
4. เพื่อให้ก่อประโยชน์สูงสุดในการใช้ข้อมูลในคอมพิวเตอร์ ควรจะถูกใช้ร่วมกับคู่สายโทรศัพท์ได้ค่อนข้างมากต่อคอมพิวเตอร์หนึ่งเครื่อง และสามารถตอบคำถามได้ทุกคู่สายในเวลาเดียวกันโดยใช้การแบ่งเวลา (Time Sharing)

ที่กล่าวมาทั้งหมดนี้เป็นแนวความคิดหลักในการทำโครงการนี้ขึ้นมา ภายใต้โครงการนี้ที่ชื่อว่า "เครื่องบริการสอบถามข้อมูลทางโทรศัพท์อัตโนมัติ" และมีชื่อเป็นภาษาอังกฤษว่า "AUTOMATIC TELE-INFORMATION MACHINE" ซึ่งคณะจัดทำได้พยายามที่จะพัฒนาระบบนี้ให้ใช้งานได้ง่ายที่สุด เพื่อที่จะสามารถนำไปใช้งานในอนาคต ซึ่งอาจจะถูกนำไปประยุกต์ใช้งานได้หลายแบบ เนื่องจากการเข้าถึงข้อมูลต่างๆจะต้องมีรหัสเข้าไปหา ดังนั้นข้อมูลแต่ละชนิดจะต้องมีรหัสประจำตัวเอง เช่น

- รหัส 1 สอบถามข้อมูลทางด้านธุรกิจประจำวัน
  - รหัส 2 สอบถามข้อมูลการประกาศผลการออกสลากกินแบ่งรัฐบาล
  - รหัส 3 สอบถามข้อมูลการประกาศผลการสอบคัดเลือกเข้ามหาวิทยาลัย
- และอีกมากมายต่อการนำไปใช้งาน

คณะผู้จัดทำหวังว่า โครงการนี้จะก่อให้เกิดประโยชน์ในการใช้งานต่อไปในอนาคตเป็นอย่างยิ่ง

## บทที่ 2

### ทฤษฎี

ทฤษฎีพื้นฐานที่จะถูกนำมาใช้ในการอธิบายการทำงานของโครงงานนี้ สามารถแยกการอธิบายออกเป็นเรื่องต่างๆ ซึ่งจะอธิบายเฉพาะเรื่องที่น่าสนใจ ได้ดังต่อไปนี้

2.1 เกล็ดำมอดดูเลชัน (Delta Modulation)

2.2 วงจรรองความถี่สัญญาณ (Filter)

2.3 ทฤษฎีและการใช้งาน Z80 PIO (Z80 Parallel I/O)

#### 2.1 เกล็ดำมอดดูเลชัน

การมอดดูเลทโดยทั่วไป สามารถแบ่งออกได้เป็น 2 ประเภทใหญ่ๆ ได้แก่

1. การมอดดูเลทเชิงอนาลอก (Analog Modulation) คือการมอดดูเลทสัญญาณอินพุท (Input Signal) แล้วได้สัญญาณเอาต์พุท (Output Signal) ออกมาในลักษณะที่ขนาดของสัญญาณ (Amplitude) มีความต่อเนื่องกับเวลา

2. การมอดดูเลทเชิงดิจิตอล (Digital Modulation) คือการมอดดูเลทสัญญาณอินพุท แล้วได้สัญญาณเอาต์พุทออกมาในลักษณะที่ขนาดของสัญญาณไม่มีความต่อเนื่องกับเวลา ซึ่งสัญญาณเอาต์พุทในลักษณะนี้ คือ สัญญาณดิจิตอล (Digital Signal)

การมอดดูเลทเชิงดิจิตอล มีหลายแบบ แต่ประเภทที่เป็นพื้นฐาน และนิยมใช้งานมากที่สุด มี 2 ประเภท ได้แก่

- เกล็ดำมอดดูเลชัน

- การมอดดูเลทแบบรหัสพัลส์ (Pulse Code Modulation)

สำหรับในที่จะอธิบายเฉพาะส่วนของ เอลด้ามอดดูเลชัน เนื่องจากในโครงการนี้ ได้มีการนำเฉพาะเอลด้ามอดดูเลชันมาใช้งานเท่านั้น

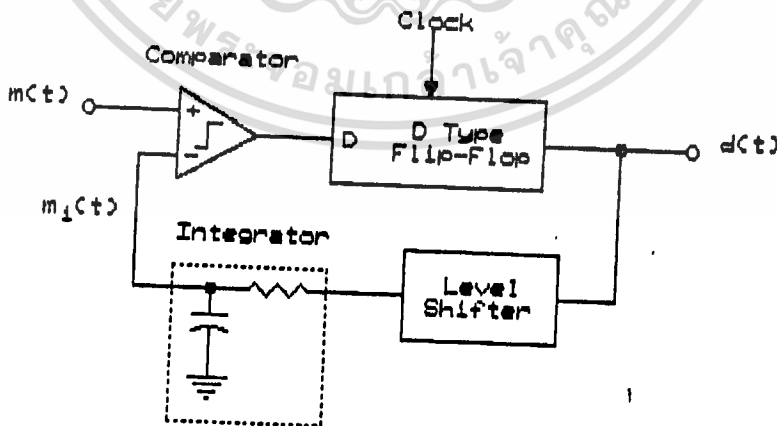
เพื่อให้สามารถเข้าใจได้ง่าย จะขออธิบายเป็นขั้นตอน โดยแยกอธิบายออกเป็น 2 ส่วน ดังนี้

2.1.1 เอลด้ามอดดูเลชันอย่างง่าย (Simple Delta Modulation)

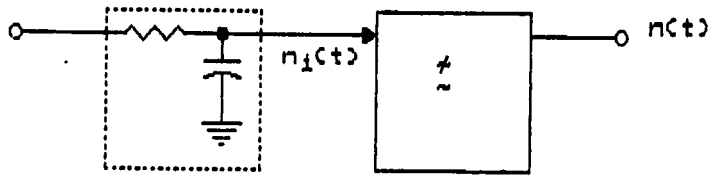
2.1.2 คอมแพนดด์เอลด้ามอดดูเลชัน (Companded Delta Modulation)

2.1.1 เอลด้ามอดดูเลชันอย่างง่าย

เอลด้ามอดดูเลชัน จะทำการมอดดูเลทสัญญาณอินพุทออกมาเป็นสัญญาณพัลส์ที่เป็นเลขฐานสอง ซึ่งได้มาจากการเปรียบเทียบขนาดของสัญญาณอินพุทปัจจุบันกับสัญญาณอินพุทในอดีตที่ถูกบ้อนกลับมา โดยสัญญาณอินพุทจะต้องถูกแซมปลิง (Sampling) ด้วยความถี่ของสัญญาณนาฬิกา (Clock) และการทำงานทั้งหมดก็จะต้องอ้างอิงถึงสัญญาณนาฬิกาไปด้วย ซึ่งแผนผังการทำงาน (Block Diagram) ของเอลด้ามอดดูเลชัน สามารถแสดงได้ดังรูปที่ 2.1



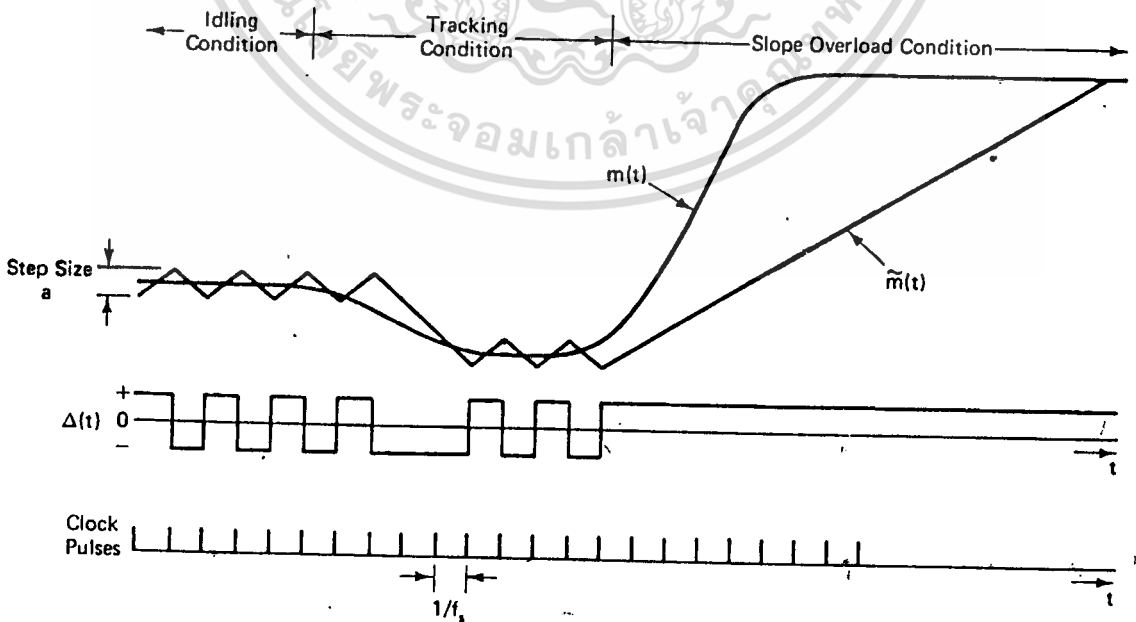
ก) วงจรเข้ารหัส (Encoder)



ข) วงจรถอดรหัส (Decoder)

รูปที่ 2.1 แผนผังการทำงานของเคลตามอคดูเลขขึ้น

จากรูปที่ 2.1 สัญญาณอินพุต  $m(t)$  จะถูกนำมาเปรียบเทียบกับสัญญาณอินทิเกรตที่ถูกรับย้อนกลับ  $m_j(t)$  ซึ่งถ้า  $m(t)$  มีขนาดของสัญญาณสูงกว่า  $m_j(t)$  สัญญาณเอาต์พุต  $d(t)$  จะเป็นสัญญาณพัลส์ทางค่านบวก (Positive Pulse) แต่ถ้า  $m(t)$  มีขนาดของสัญญาณต่ำกว่า  $m_j(t)$  สัญญาณเอาต์พุตที่ได้ ก็จะเป็นสัญญาณพัลส์ทางค่านลบ (Negative Pulse) ซึ่งสัญญาณเอาต์พุตที่ได้ไม่ว่าจะเป็นในกรณีใดก็ตาม จะถูกนำมาผ่านเข้าวงจรสร้างระดับสัญญาณ (Level Shifter) เพื่อนำเข้าวงจรอินทิเกรเตอร์ (Integrator) แล้วถูกย้อนกลับมาเป็นสัญญาณ  $m_j(t)$  เพื่อใช้เป็นสัญญาณเปรียบเทียบกับสัญญาณอินพุตที่สัญญาณนาฬิกาถูกต่อไป



รูปที่ 2.2 แสดงรูปคลื่นสัญญาณสำหรับเคลตามอคดูเลขขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

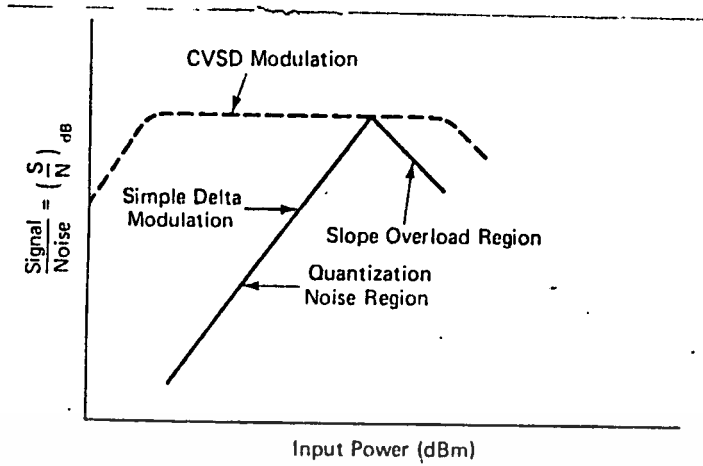
1. สภาวะไอเดิล (Idling Condition) คือ สภาวะที่ไม่มีสัญญาณอินพุตผ่านเข้า วงจรเคลต้ามอดคูลเลเตอร์ (Delta Modulator) ซึ่งจะทำให้วงจรอินทิเกรเตอร์สร้าง สัญญาณแรมพ์ (Ramp) ซึ่งในสภาวะนี้ สัญญาณเอาต์พุต  $d(t)$  ที่ได้ จะประกอบด้วยสัญญาณ พัลส์ทางด้านบวกและทางด้านลบสลับกันไป

2. สภาวะแทรค (Tracking Condition) คือ สภาวะปกติที่มีสัญญาณอินพุตเข้ามา อย่างต่อเนื่อง ซึ่งก็จะได้สัญญาณเอาต์พุตตามลักษณะของสัญญาณอินพุตที่ถูกมอดคูลเลท ตามที่ ได้อธิบายมาแล้ว

ทั้งสองสภาวะแรกนั้น ความแตกต่างของสัญญาณอินพุตและสัญญาณบ้อนกลับ จะก่อให้เกิด ความผิดพลาด (Error) ของสัญญาณขึ้น เราจะเรียกปรากฏการณ์เช่นนี้ว่า การเกิด สัญญาณรบกวนแบบแกรนูลาร์หรือแบบควอนไทซ์ (Granular or Quantization Noise) ซึ่งสัญญาณรบกวนนี้ สามารถลดลงได้ด้วยทั้งการลดค่าของระดับ "a" และ การเพิ่มอัตรา การสุ่ม (Sampling Rate)

3. สภาวะความชันโอเวอร์โหลด (Slope Overload Condition) คือ สภาวะ ที่สัญญาณอินพุตมีขนาดที่เปลี่ยนแปลงอย่างรวดเร็ว หรือ มีความชันของสัญญาณมากจนสัญญาณ บ้อนกลับมีความชันไม่ทัน ลักษณะเช่นนี้จะทำให้สัญญาณเอาต์พุตเกิดความผิดพลาด เนื่องจาก ระดับของสัญญาณที่เกิดขึ้นในแต่ละครั้งของ  $m_j(t)$  มีค่าคงที่ สภาวะนี้จะทำให้สัญญาณเกิด ความเพี้ยน (Distortion) ขึ้น ซึ่งนี่ก็คือข้อจำกัดของเคลต้ามอดคูลเลชัน ที่ไม่สามารถทำ การมอดคูลเลทสัญญาณอินพุตที่มีการเปลี่ยนแปลงขนาดอย่างรวดเร็ว รวมทั้งสัญญาณที่มีความถี่ สูง (High Frequency Signal)

เมื่อมีสภาวะความชันโอเวอร์โหลดเกิดขึ้น เราเรียกปรากฏการณ์เช่นนี้ว่า การเกิด สัญญาณรบกวนแบบความชันโอเวอร์โหลด (Slope Overload Noise) ซึ่งจะมีผลทำให้ สัญญาณรบกวนแบบควอนไทซ์เพิ่มขึ้น



รูปที่ 2.3 แสดงอัตราส่วน S/N ของเคลด้ามอดคูเลชัน

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

ถ้าให้ สัญญาณอินพุต

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

อัตราการเพิ่มสูงสุดของสัญญาณอินพุต

$$\left. \frac{dm(t)}{dt} \right|_{max} = 2\pi f_m A \quad \dots(2.2)$$

และ อัตราการเพิ่มของสัญญาณป้อนกลับ  $m_f(t)$

$$r(t) = a/(1/f_s) = af_s \quad \dots(2.3)$$

เมื่อ a คือ ขนาดของระดับการเปลี่ยนแปลงของสัญญาณป้อนกลับ

$f_s$  คือ ความถี่อัตราการสุ่ม

ซึ่งเราต้องการให้สมการ (2.2) และ (2.3) มีค่าเท่ากัน ดังนั้นจะได้ว่า

$$2\pi f_m A = af_s$$

$$f_m A = af_s/2\pi \quad \dots(2.4)$$

จะเห็นได้ว่า การลดสัญญาณรบกวนแบบความชันโอเวอร์โหลด สามารถทำได้โดย

การเพิ่มขนาดของระดับการเปลี่ยนแปลงของสัญญาณป้อนกลับ และ การเพิ่มอัตราการสุ่ม

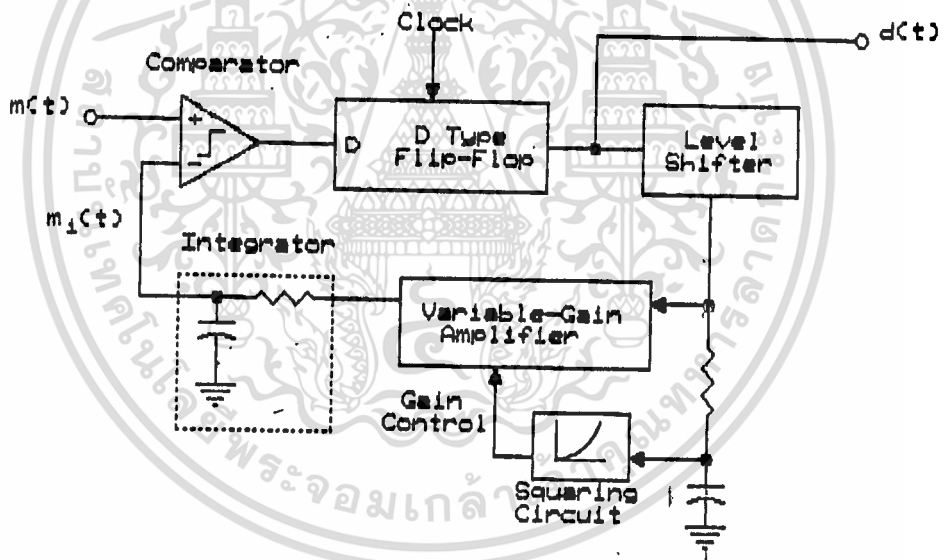
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ดูแลเห็นประโยชน์เชิงวิชาการด้านการศึกษา

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

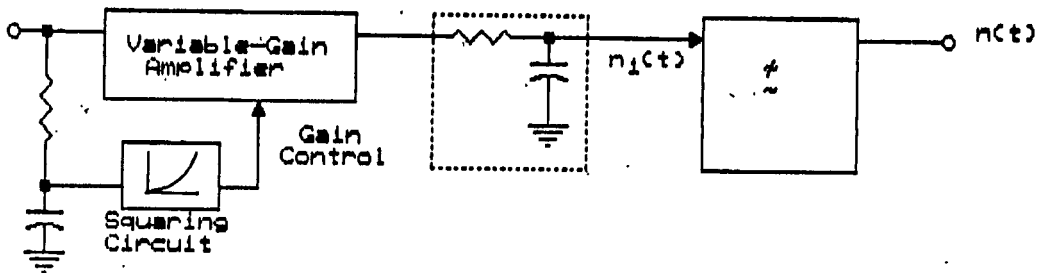
### 2.1.2 คอมแพนด์เคลต้ามอดดูเลชัน

จากรูปที่ 2.3 ซึ่งแสดงอัตราส่วน S/N ของเคลต้ามอดดูเลชันอย่างง่ายนั้น มีค่าไม่คงที่ ดังนั้นการที่จะทำให้อัตราส่วน S/N มีค่าคงที่นั้น จำเป็นต้องทำให้ค่าของระดับ "a" สามารถปรับได้ ซึ่งเมื่อสัญญาณอินพุตมีขนาดเล็ก ขนาดของ a ก็จะมีขนาดเล็ก เมื่อสัญญาณอินพุตมีขนาดเพิ่มขึ้น ขนาดของ a ก็จะต้องเพิ่มขึ้น

สำหรับการใช้งานทางด้านโทรศัพท์ เคลต้ามอดดูเลชันที่ใช้ จะต้องสามารถปรับได้ ซึ่งเราใช้แบบที่เรียกว่า เคลต้ามอดดูเลชันแบบความชันแปรเปลี่ยนต่อเนื่อง (Continuous Variable Slope Delta Modulation (CVSD)) ซึ่งสามารถเขียนแผนผังการทำงานได้ดังรูปที่ 2.4



ก) วงจรเข้ารหัส



ข) วงจรถอดรหัส

รูปที่ 2.4 แผนผังการทำงานของเคลต้ามอดดูเลชันแบบความชันแปรเปลี่ยนต่อเนื่อง

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์ของกระทรวงมหาดไทย กรุงเทพมหานคร เมื่อผู้รู้เห็นหนังสือฉบับนี้ขอสงวนสิทธิ์ในการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



จากรูป วงจรสแควร์ (Square Circuit) จะเป็นตัวกำหนดอัตราขยายแรงดันของวงจรขยายป้อนกลับ (Feedback Amplifier) ซึ่งจะมีค่าเพิ่มขึ้นตามขนาดของสัญญาณอินพุต ซึ่งด้วยวิธีนี้จะทำให้สัญญาณเอาต์พุตมีความเป็นเชิงเส้น (Linear) กับสัญญาณอินพุต

สำหรับในโครงงานนี้ จะใช้วิธีของเคลต์ตามอดดูเลชันในการแปลงสัญญาณเสียง เป็นสัญญาณดิจิทัล ซึ่งสัญญาณเอาต์พุตที่ได้จะออกมาให้ลักษณะของสัญญาณแบบอนุกรม (Serial) ดังนั้นจึงต้องนำสัญญาณนี้มาทำการแปลงเป็นสัญญาณแบบขนาน (Parallel) อีกทีหนึ่ง

## 2.2 วงจรกรองความถี่สัญญาณ

เนื่องจากในโครงงานนี้ จำเป็นต้องอาศัยวงจรกรองความถี่ เพื่อใช้กรองสัญญาณเสียงให้อยู่ในช่วงความถี่ที่ต้องการใช้งาน ดังนั้นในหัวข้อนี้ จะอธิบายเกี่ยวกับทฤษฎีพื้นฐานของวงจรกรองความถี่สัญญาณโดยทั่วไป

วงจรกรองความถี่สัญญาณ ที่นิยมใช้งานทางด้านสัญญาณเสียง และ การสื่อสารข้อมูล สามารถจำแนกออกได้เป็น 2 ประเภทใหญ่ๆ ดังนี้

2.2.1 วงจรกรองความถี่สัญญาณแบบพาสซีฟ (Passive Filter)

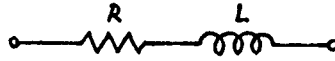
2.2.2 วงจรกรองความถี่สัญญาณแบบแอกทีฟ (Active Filter)

### 2.2.1 วงจรกรองความถี่สัญญาณแบบพาสซีฟ

เป็นวงจรกรองความถี่ที่ใช้อุปกรณ์ประเภทพาสซีฟในการประกอบวงจร ซึ่งอุปกรณ์เหล่านี้ ได้แก่ ตัวต้านทาน (Resistor), ตัวเก็บประจุไฟฟ้า (Capacitor) และ ตัวเหนี่ยวนำ (Inductor) เป็นต้น สำหรับการใช้งานในช่วงความถี่ของสัญญาณเสียง ในปัจจุบันที่ใช้ตัวเหนี่ยวนำ มักจะเกิดปัญหาขึ้น เนื่องจากค่าของความต้านทานภายในตัวมันเอง ซึ่งแบบจำลองของตัวเหนี่ยวนำ สามารถแสดงได้ดังรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสาร (0)28747



รูปที่ 2.5 แบบจำลองของตัวเหนี่ยวนำที่ใช้งานจริง

ค่าของ แฟคเตอร์คุณภาพ (Quality Factor) ของตัวเหนี่ยวนำ เป็นดังสมการ

$$Q_L = \omega L / R \quad \dots (2.5)$$

จะเห็นได้ว่า ถ้าค่าความต้านทานภายในของตัวเหนี่ยวนำมีค่ามาก ก็จะทำให้ค่าแฟคเตอร์คุณภาพมีค่าน้อย ดังนั้นการที่จะทำให้วงจรองความถี่มีความเที่ยงน้อย จำเป็นต้องใช้ตัวเหนี่ยวนำที่มีค่าของแฟคเตอร์คุณภาพมาก อย่างไรก็ตาม การใช้งานที่ความถี่ต่ำกว่า 1 KHz ตัวเหนี่ยวนำที่ใช้จะมีขนาดใหญ่ และราคาสูงความไปด้วย ดังนั้น การใช้งานด้วยตัวเหนี่ยวนำ จึงไม่ค่อยจะประสบความสำเร็จเท่าที่ควร

### 2.2.2 วงจรกรองความถี่สัญญาณแบบแอกทิฟ

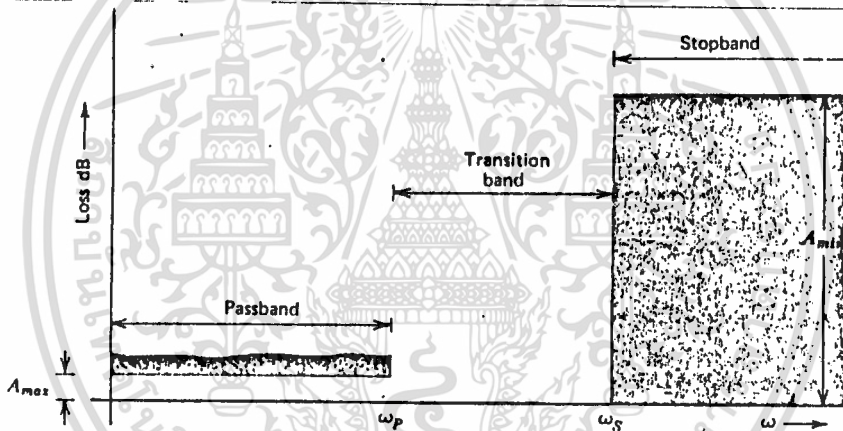
เป็นวงจรกรองความถี่ที่ใช้อุปกรณ์ประเภทพาสซีฟ ร่วมกับอุปกรณ์ประเภทแอกทิฟ ซึ่งส่วนใหญ่จะใช้ข้อออปแอมป์ (Operational Amplifier (Op-Amp)) ซึ่งวงจรกรองความถี่แบบนี้มีข้อดีหลายประการ สำหรับวงจรกรองความถี่แบบแอกทิฟที่ใช้ RC จะมีข้อดีดังนี้

1. วงจรที่ได้มีขนาดเล็ก
2. วงจรที่ได้มีความเชื่อถือได้ (Reliability) สูง
3. วงจรมีราคาถูกกว่าวงจรแบบพาสซีฟที่มีคุณสมบัติเหมือนกัน
4. การออกแบบง่ายกว่าการออกแบบวงจรกรองความถี่แบบพาสซีฟ
5. สามารถสร้างอัตราการขยายแรงดันของสัญญาณได้ ในขณะที่วงจรแบบพาสซีฟเฉพาะการสูญเสียแรงดัน (Voltage Loss)

ถ้าเราจะแบ่งประเภทของวงจรกรองความถี่สัญญาณตามคุณสมบัติการทำงาน จะสามารถจำแนกออกได้ดังนี้

### วงจรกรองความถี่แบบโลว์พาส (Low Pass Filter (LP))

เป็นวงจรกรองความถี่ที่ยอมให้เฉพาะสัญญาณที่มีความถี่ต่ำกว่าที่กำหนดผ่านไปได้ ถ้าจะอธิบายอีกทางหนึ่งก็คือว่า ถ้าสัญญาณหนึ่งประกอบด้วยสัญญาณที่มีความถี่มากมายผ่านเข้ามาในวงจรนี้ สัญญาณช่วงที่มีความถี่ต่ำกว่าที่กำหนดจะมีการสูญเสียแรงดันน้อยมาก ในขณะที่สัญญาณช่วงที่มีความถี่สูงกว่าที่กำหนดจะมีการสูญเสียแรงดันมาก ซึ่งลักษณะของการสูญเสียแรงดันของสัญญาณ สามารถแสดงได้ดังรูปที่ 2.6



รูปที่ 2.6 คุณสมบัติของวงจรกรองความถี่แบบโลว์พาส

จากรูป จะเห็นว่า วงจรกรองความถี่แบบโลว์พาส จะยอมให้สัญญาณความถี่ต่ำ ตั้งแต่สัญญาณไฟกระแสตรง (dc) ไปจนถึงสัญญาณที่มีความถี่  $\omega_p$  ซึ่งเราเรียกว่า ความถี่คัทออฟ (Cutoff Frequency) โดยให้มีอัตราการสูญเสียของแรงดันมากที่สุดเท่ากับ  $A_{max}$  dB ช่วงความถี่ตั้งแต่ dc ไปจนถึง  $\omega_p$  เรียกว่า ช่วงความถี่ผ่าน (Passband) ส่วนความถี่ที่มีค่าตั้งแต่  $\omega_s$  ขึ้นไป จะมีอัตราการสูญเสียของแรงดันต่ำที่สุดเท่ากับ  $A_{min}$  dB ช่วงความถี่ตั้งแต่  $\omega_s$  ไปจนถึงอนันต์ (Infinity) เราเรียกว่า ช่วงความถี่หยุด (Stopband) ส่วนความถี่  $\omega_s$  เราเรียกว่า ความถี่ขอบของช่วงความถี่หยุด (Stopband Edge Frequency) และช่วงความถี่ตั้งแต่  $\omega_p$  ไปจนถึง  $\omega_s$  เรียกว่า ช่วงความถี่เปลี่ยนแปลง (Transition Band)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

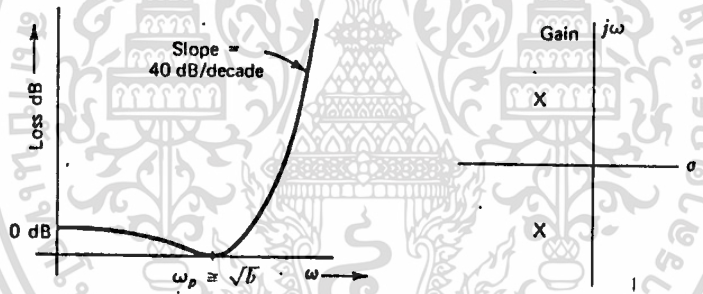
สำหรับฟังก์ชันของอัตราขยายแรงดันอันดับสอง (Second Order Gain Function) ของวงจรกรองความถี่แบบโลว์พาสเป็นดังสมการ

$$\text{Gain} = \frac{V_o}{V_{in}} = \frac{b}{s^2+as+b} = \frac{w_p^2}{s^2+(w_p/Q_p) s+w_p^2} \dots(2.6)$$

ถ้าเขียนในลักษณะของการสูญเสีย จะได้ดังสมการ

$$\text{Loss} = \frac{V_{in}}{V_o} = \frac{s^2+as+b}{b} \dots(2.7)$$

ซึ่งสามารถแสดงได้ดังรูปที่ 2.7



ก) แสดงอัตราการสูญเสีย      ข) แสดงกราฟโพล-ซีโร

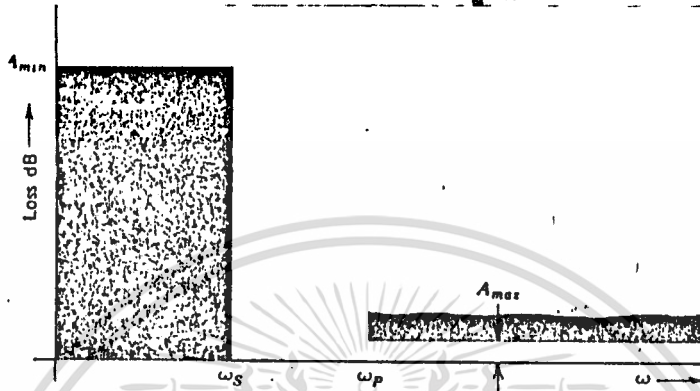
รูปที่ 2.7 ฟังก์ชันของวงจรกรองความถี่โลว์พาสอันดับสอง

จากรูป 2.7ก ที่ความถี่ต่ำจะมีอัตราการสูญเสียเท่ากับ 0 dB และ ที่ความถี่สูงจะมีอัตราการสูญเสียเพิ่มขึ้นตามฟังก์ชัน  $s^2$  ซึ่งมีค่าเท่ากับ 40 dB ในรูปที่ 2.7ข จะเห็นว่าฟังก์ชันของอัตราขยายแรงดันอันดับสองมีโพลเชิงซ้อน (Complex Pole) อยู่ 1 คู่ ซึ่งตำแหน่งของโพลคู่นี้จะเป็นตัวกำหนดรูปร่างของการตอบสนองความถี่ (Response) ในช่วงความถี่ผ่านของวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรกรองความถี่แบบไฮพาส (High Pass Filter (HP))

เป็นวงจรกรองความถี่ที่ยอมให้สัญญาณที่มีความถี่สูงกว่าที่กำหนดผ่านไป สำหรับ  
คุณสมบัติของวงจรกรองความถี่ประเภทนี้ แสดงได้ดังรูปที่ 2.8



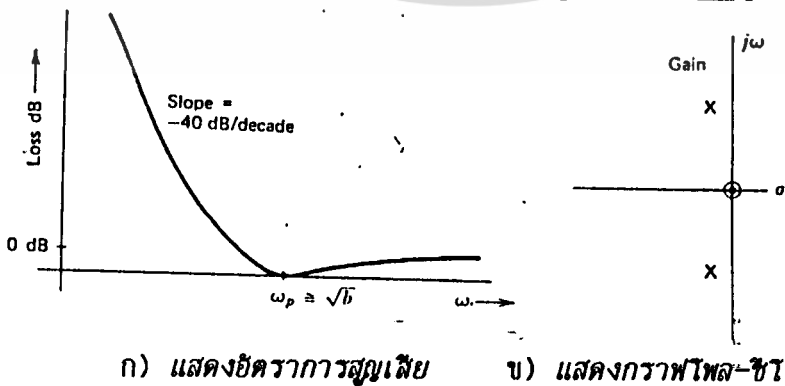
รูปที่ 2.8 คุณสมบัติของวงจรกรองความถี่แบบไฮพาส

จากรูปจะเห็นว่า ช่วงความถี่ผ่านจะเริ่มตั้งแต่ความถี่  $\omega_p$  จนถึงอนันต์ และช่วงความถี่หยุด จะเริ่มตั้งแต่ dc ไปจนถึงความถี่  $\omega_s$

สำหรับฟังก์ชันของอัตราขยายแรงดันอันดับสอง ของวงจรกรองความถี่แบบไฮพาส เป็นดังสมการ

$$\text{Gain} = \frac{V_o}{V_{in}} = \frac{s^2}{s^2 + as + b} = \frac{s^2}{s^2 + (\omega_p/Q_p)s + \omega_p^2} \dots (2.8)$$

ซึ่งสามารถแสดงได้ดังรูปที่ 2.9



ก) แสดงอัตราการสูญเสีย

ข) แสดงกราฟโพล-ซีโร

รูปที่ 2.9 ฟังก์ชันของวงจรกรองความถี่ไฮพาสอันดับสอง

จากรูป 2.8ก จะเห็นว่าที่ความถี่ต่ำจะมีอัตราการสูญเสียโดยมีความชันเท่ากับ -40 dB และที่ความถี่สูง จะมีอัตราการสูญเสียเท่ากับ 0 dB และในรูป 2.8ข พังก์ชันของอัตราขยายแรงดันจะมีพลเชิงซ้อน 1 คู่ และ ซีโรว์ 1 คู่ที่จุดกำเนิด (Origin)

### วงจรรองความถี่แบบแบนด์พาส (Band Pass Filter (BP))

เป็นวงจรรองความถี่ที่ยอมให้สัญญาณที่มีความถี่อยู่ในช่วงที่กำหนดผ่านไปได้ ซึ่งช่วงความถี่ผ่านจะต้องมีอัตราการสูญเสียต่ำมาก สำหรับคุณสมบัติของวงจรรองความถี่ชนิดนี้สามารถแสดงได้ดังรูปที่ 2.10

ซึ่งจากรูปสามารถอธิบายได้ว่า ช่วงความถี่ผ่านตั้งแต่  $\omega_1$  ไปจนถึง  $\omega_2$  จะมีอัตราการสูญเสียสูงสุดเท่ากับ  $A_{max}$  dB และ ช่วงความถี่หยุดได้แก่ ช่วงตั้งแต่ dc ไปถึง  $\omega_3$  และ ช่วงตั้งแต่  $\omega_4$  ไปถึงอนันต์ จะมีอัตราการสูญเสียต่ำสุดเท่ากับ  $A_{min}$  dB



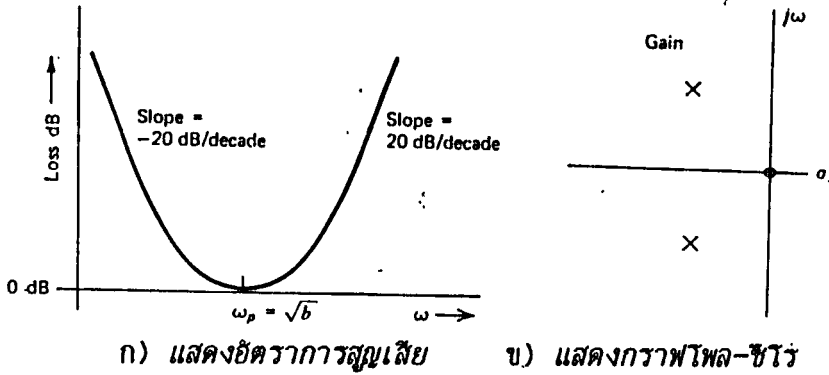
รูปที่ 2.10 คุณสมบัติของวงจรรองความถี่แบบแบนด์พาส

ฟังก์ชันของอัตราการขยายแรงดันอันดับสอง ของวงจรรองความถี่แบบแบนด์พาส เป็นดังสมการ

$$\text{Gain} = \frac{V_o}{V_{in}} = \frac{as}{s^2+as+b} = \frac{(w_p/Q_p)s}{s^2+(w_p/Q_p)s+w_p^2} \quad \dots(2.8)$$

ซึ่งสามารถแสดงได้ดังรูปที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับศิษย์งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

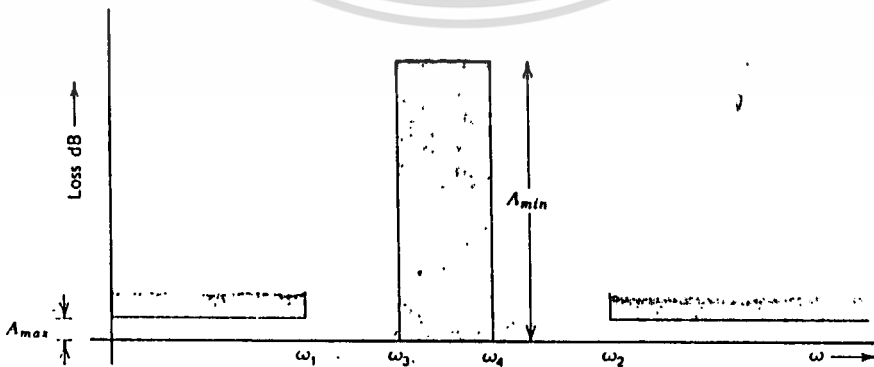


รูปที่ 2.11 พังก์ชันของวงจรรองความถี่แบนด์พาสอันดับสอง

จากรูป 2.11ก จะเห็นว่าที่ความถี่ต่ำและความถี่สูงจะมีอัตราการสูญเสียโดยมีความชันเท่ากับ  $-20$  dB และ  $20$  dB ตามลำดับ และที่ความถี่พาส  $\omega_p$  จะมีอัตราการสูญเสียเท่ากับ  $0$  dB และในรูป 2.11ข พังก์ชันของอัตราขยายแรงดันจะมีเฟสเชิงซ้อนอยู่  $1$  คู่ และซีโรที่จุดกำเนิด (Origin)

**วงจรรองความถี่แบบแบนด์รีเจ็คท์ (Band Reject Filter (BR))**

เป็นวงจรรองความถี่ที่ไม่ยอมให้สัญญาณที่มีความถี่อยู่ในช่วงที่กำหนดผ่านไปได้ โดยช่วงความถี่ที่กำหนดก็คือช่วงความถี่หยุดนิ่งเอง ซึ่งจะต้องมีอัตราการสูญเสียสูงมาก สำหรับคุณสมบัติของวงจรรองความถี่ชนิดนี้ สามารถแสดงได้ดังรูปที่ 2.12



รูปที่ 2.12 คุณสมบัติของวงจรรองความถี่แบบแบนด์รีเจ็คท์

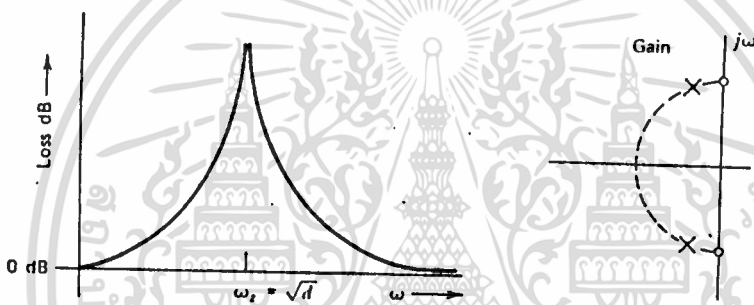
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปอธิบายได้ว่า ช่วงความถี่หยุดตั้งแต่  $\omega_3$  ไปจนถึง  $\omega_4$  จะมีอัตราการสูญเสียต่ำสุดเท่ากับ  $A_{\min}$  dB และช่วงความถี่ผ่านได้แก่ ช่วงตั้งแต่ dc ไปถึง  $\omega_1$  และ ช่วงตั้งแต่  $\omega_2$  ไปถึงอนันต์ จะมีอัตราการสูญเสียสูงสุดเท่ากับ  $A_{\max}$  dB

ฟังก์ชันของอัตราขยายแรงดันอันดับสอง ของวงจรกรองความถี่แบบแบนด์พาสเชิงซ้อนเป็นดังสมการ

$$\text{Gain} = \frac{V_o}{V_{in}} = \frac{s^2+d}{s^2+ast+b} = \frac{s^2+\omega_z^2}{s^2+(\omega_p/Q_p)s+\omega_p^2} \quad \dots(2.8)$$

ซึ่งสามารถแสดงได้ดังรูปที่ 2.13



ก) แสดงอัตราการสูญเสีย ข) แสดงกราฟโพล-ซีโร่

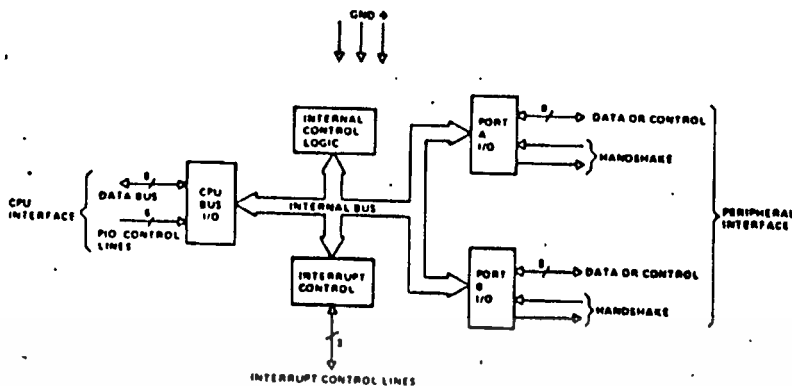
รูปที่ 2.13 ฟังก์ชันของวงจรกรองความถี่แบบแบนด์พาสเชิงซ้อน

จากรูป 2.13ก จะเห็นว่าที่ความถี่ต่ำและความถี่สูงจะมีอัตราการสูญเสียเท่ากับ 0 dB และที่ความถี่ซีโร่  $\omega_z$  จะมีอัตราการสูญเสียเป็นอนันต์ และในรูป 2.11ข ฟังก์ชันของอัตราขยายแรงดันจะมีโพลเชิงซ้อนอยู่ 1 คู่ ทางด้านซ้ายของเพลน  $s$  ( $s$  plane) และซีโร่เชิงซ้อน 1 คู่ที่แกน  $j\omega$

### 2.3 ทฤษฎีและการใช้งาน Z80 PIO

Z80 PIO เป็นอุปกรณ์อินพุทเอาต์พุทแบบขนานที่โปรแกรมได้ ซึ่งถูกออกแบบมาให้งานร่วมกับไมโครโปรเซสเซอร์ Z80 (Z80 Microprocessor) สำหรับแผนผังการทำงานของ Z80 PIO สามารถแสดงได้ดังรูปที่ 2.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 แสดงแผนผังการทำงานของ Z80 PIO

จะเห็นว่า Z80 PIO ประกอบด้วยพอร์ตอินพุทเอาต์พุท (Port Input/Output) จำนวน 2 พอร์ต ได้แก่ พอร์ต A และ B ซึ่งแต่ละพอร์ตจะมีเส้นข้อมูล (Data Line) จำนวน 8 เส้น และเส้นควบคุม (Control Lines) จำนวน 2 เส้น เส้นสัญญาณอินพุท และเอาต์พุททั้งหมด สามารถนำมาใช้งานร่วมกับอุปกรณ์ทีทีแอล (TTL) ได้

จากรูปข้างต้น ภายในของ Z80 PIO ยังประกอบด้วยส่วนการทำงานอีก 2 ส่วน คือ

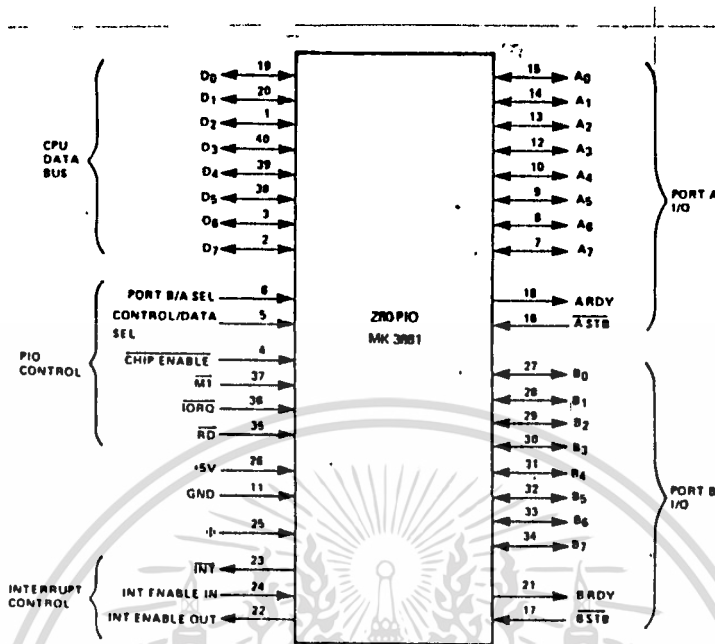
1. ส่วนควบคุมทางตรรกภายใน (Internal Control Logic) เป็นส่วนที่ใช้งาน การควบคุมข้อมูลภายในให้ถูกต้อง
2. ส่วนควบคุมอินเทอร์รัพท์ (Interrupt Control) เป็นส่วนที่ทำให้ Z80 PIO สามารถใช้งานในโหมดอินเทอร์รัพท์ของ Z80 ได้ โดยที่ Z80 PIO จะใช้เป็น ตัวสร้างสัญญาณอินเทอร์รัพท์ และรับสัญญาณ Interrupt Acknowledge จาก Z80 ซึ่งการทำงานของ Z80 PIO ในโหมดนี้ สามารถโปรแกรมได้

### 2.3.1 ขาสัญญาณต่างๆ ของ Z80 PIO

สำหรับแผนผังขาสัญญาณต่างๆ ของ Z80 PIO สามารถแสดงได้ดังรูปที่ 2.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 แสดงขาสัญญาณต่างๆ ของ Z80 PIO

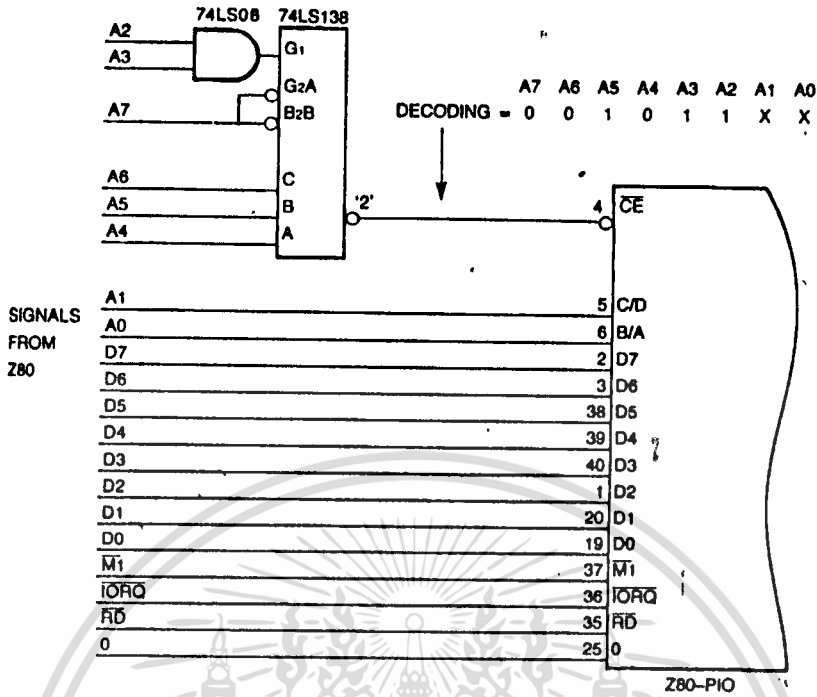
จากรูปข้างต้น ขาสัญญาณแต่ละขามีหน้าที่ ดังนี้

- D0-D7** เป็นขาข้อมูลที่ใช้ในการรับส่งข้อมูลกับ Z80
- B/A Select** เป็นขาอินพุตสำหรับทำการเลือกว่าต้องการใช้งานที่พอร์ทใด ซึ่งถ้ามีลอจิก 0 หมายถึง เลือกพอร์ท A ถ้ามีลอจิก 1 หมายถึง เลือกพอร์ท B
- C/D Select** เป็นขาอินพุตสำหรับทำการเลือกว่าต้องการติดต่อกับพอร์ทที่ใช้ควบคุม หรือพอร์ทสำหรับรับ-ส่งข้อมูล
- CE\** (*Chip Enable Input*) เป็นขาอินพุตเพื่อให้ Z80 อยู่ในสภาวะทำงาน
- CLOCK** เป็นขาอินพุตสำหรับรับสัญญาณนาฬิกา
- M1\** เป็นขาที่ใช้สำหรับรับสัญญาณ M1\ จาก Z80 ซึ่งขานี้จะใช้สำหรับการควบคุมการทำงานภายใน ถ้า M1\ และ RD\ เกิดการแอกทีฟพร้อมกันหมายความว่า Z80 จะทำการเพช (Fetch) คำสั่ง ซึ่งตัว Z80 PIO จะรับรู้สถานะการทำงานของ Z80 ตลอดเวลา และคำสั่งที่จำเป็นสำหรับการทำงานของ Z80 PIO ก็คือ RETI

- IORQ\** เป็นขาอินพุทที่รับสัญญาณ IORQ\ จาก Z80 ซึ่งสัญญาณนี้จะถูกใช้งานร่วมกับสัญญาณ B/A Select, C/D Select และ CE\ เพื่อใช้ในการควบคุมการรับส่งคำสั่งและข้อมูลระหว่าง Z80 กับ Z80 PIO
- RD\** เป็นขาอินพุทที่รับสัญญาณ RD\ จาก Z80 ซึ่งถ้าสัญญาณนี้แอกทีฟพร้อมๆ กับสัญญาณ IORQ\ และ CE\ หมายความว่า Z80 กำลังต้องการรับข้อมูลจาก Z80 PIO
- IEI และ IEO** เป็นขาอินพุทและเอาต์พุท ตามลำดับ ใช้ในการอีนาเบิล (Enable) หรือ ดิสเอเบิล (Disable) การตอบสนองต่อสัญญาณสไตรบ
- INT\** เป็นขาเอาต์พุทที่ใช้ในการขออินเทอร์รัพท์ Z80 ซึ่งจะแอกทีฟก็ต่อเมื่อ ถูกโปรแกรมให้ทำงานในโหมดของอินเทอร์รัพท์
- AO-A7** เป็นเส้นสัญญาณอินพุทเอาต์พุทสำหรับพอร์ท A
- ASTB\** เป็นขาอินพุทที่รับสัญญาณจากอุปกรณ์ภายนอก สำหรับการแฮนด์เชคข้อมูล (Data Handshaking) ไปยังรีจิสเตอร์ (Register) ภายในพอร์ท A
- ARDY\** (*Register A Ready*) เป็นขาเอาต์พุทที่ Z80 PIO บอกให้รู้ว่า ได้ทำการรับข้อมูลเข้ารีจิสเตอร์ A ในการณิที่เป็นพอร์ทอินพุท หรือส่งข้อมูลออกจากรีจิสเตอร์ A ในการณิที่เป็นพอร์ทเอาต์พุท เรียบร้อยแล้ว
- BO-B7** เป็นเส้นสัญญาณอินพุทเอาต์พุทสำหรับพอร์ท B
- BSTB\** เป็นขาอินพุทที่รับสัญญาณจากอุปกรณ์ภายนอก สำหรับการแฮนด์เชคข้อมูล (Data Handshaking) ไปยังรีจิสเตอร์ (Register) ภายในพอร์ท B
- BRDY\** (*Register B Ready*) เป็นขาเอาต์พุทที่ Z80 PIO บอกให้รู้ว่า ได้ทำการรับข้อมูลเข้ารีจิสเตอร์ B ในการณิที่เป็นพอร์ทอินพุท หรือส่งข้อมูลออกจากรีจิสเตอร์ B ในการณิที่เป็นพอร์ทเอาต์พุท เรียบร้อยแล้ว

### 2.3.2 การอินเทอร์เฟส Z80 PIO กับ Z80

การติดต่อระหว่าง Z80 กับ Z80 PIO เป็นดังรูปที่ 2.16



รูปที่ 2.16 แสดงการอินเทอร์เฟส Z80 PIO กับ Z80

จากรูป ขาสัญญาณ B/A ถูกต่อกับ A0 ของ Z80 และ C/D ถูกต่อกับ A1 ของ Z80 ซึ่งการต่อในลักษณะเช่นนี้ เราจะได้พอร์ตต่างๆ ดังต่อไปนี้

A1	A0	PORT SELECTED
0	0	A ข้อมูล
0	1	B ข้อมูล
1	0	A ควบคุม
1	1	B ควบคุม

ตารางที่ 2.1 แสดงพอร์ตต่างๆ ที่ได้คือทำได้

และจากรูปข้างต้น จะทำให้เราได้ว่า พอร์ตที่ทำการสืลัดทั้งหมด มี 4 แอดเดรส

ได้แก่ 2C, 2D, 2E และ 2F ตามลำดับ

### 2.3.3 การโปรแกรม Z80 PIO

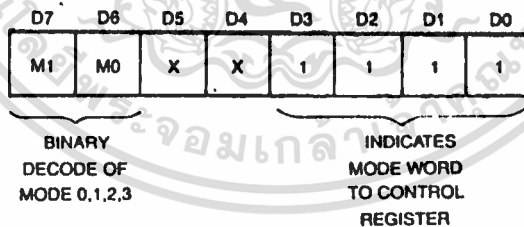
Z80 สามารถที่จะถูกโปรแกรมให้ทำงานโหมดต่างๆ ได้ทั้งหมด 4 โหมด ดังนี้

1. โหมด 0 ใช้เป็นพอร์ตเอาต์พุต (Output Port)
2. โหมด 1 ใช้เป็นพอร์ตอินพุต (Input Port)
3. โหมด 2 ใช้เป็นพอร์ต 2 ทิศทาง (The Bi-Directional Mode)
4. โหมด 3 เป็นโหมดที่ไม่มีการแฮนด์เชค

ซึ่งสำหรับในโครงการนี้ จะใช้งาน 2 โหมดแรก ดังนั้นการอธิบายจะทำเฉพาะโหมด 0 และ โหมด 1 เท่านั้น

#### การโปรแกรม Z80 PIO ในโหมด 0 และ 1

พอร์ตของ Z80 PIO ที่ถูกโปรแกรมให้อยู่ในโหมดนี้ จะทำงานในลักษณะของพอร์ตเอาต์พุต ซึ่งการโปรแกรม สามารถทำได้โดยให้ Z80 ส่งข้อมูลพอร์ตควบคุม ซึ่งเราเรียกข้อมูลชุดนี้ว่า **ข้อมูลควบคุม (Control Word)** สำหรับรายละเอียดของข้อมูลควบคุม เป็นดังรูปที่ 2.17



WE WILL WRITE OF AS THE CONTROL WORD

**รูปที่ 2.17 แสดงรายละเอียดของข้อมูลควบคุม**

จากรูป 4 บิตล่าง จะต้องมิต่ำเป็น 1 ทั้งหมด ซึ่งหมายความว่า การส่งข้อมูล XFH ไปที่พอร์ตควบคุมนั้น เป็นการโปรแกรมเพื่อให้พอร์ตฯ นั้น ทำงานในโหมดต่างๆ ซึ่งบิตที่ใช้ในการกำหนดโหมด ได้แก่ บิตที่ 6 และ 7 ดังนั้นถ้าต้องการโปรแกรมให้ Z80 PIO ทำงานในโหมดใด จะต้องส่งข้อมูลดังต่อไปนี้ไปที่พอร์ตควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

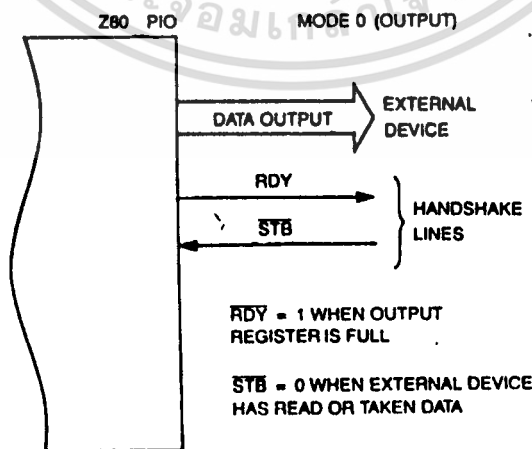
ข้อมูล	โหมด
0F	0
4F	1
8F	2
CF	3

การทำงานของ Z80 PIO ในโหมดแฮนด์เชค

การใช้งานในโหมดแฮนด์เชค คือ การใช้สัญญาณสโตรบ (Strobe) และ สัญญาณเรดดี (Ready) กับอุปกรณ์ภายนอก ถ้าไม่ต้องการใช้งานในโหมดนี้ จะต้องต่อขาสโตรบลงกราวนด์

สำหรับการแฮนด์เชค จะสามารถแบ่งการอธิบายแยกเป็น 2 โหมด ดังนี้

โหมด 0 เมื่อ Z80 ส่งข้อมูลผ่านพอร์ทัลที่ต้องการออกมาเรียบร้อยแล้ว Z80 PIO จะส่งสัญญาณเรดดีออกมา ซึ่งอุปกรณ์ภายนอกจะรับรู้ แล้วทำการรับข้อมูลจากพอร์ทัลไป จากนั้นอุปกรณ์ภายนอกจะส่งสัญญาณสโตรบออกมาเพื่อบอก Z80 PIO ให้ทราบ แผนผังการทำงานเป็นดังรูปที่ 2.18

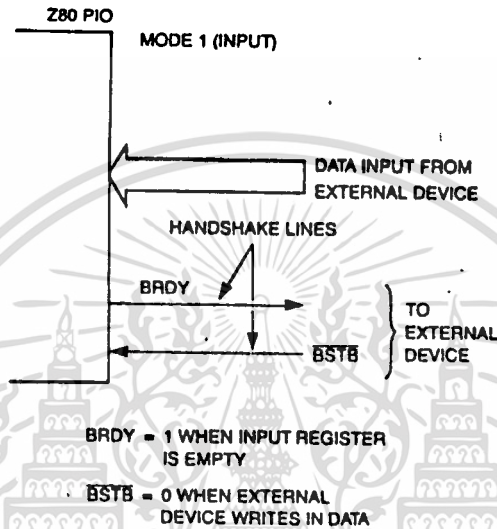


รูปที่ 2.18 แผนผังการทำงานแบบแฮนด์เชคของโหมด 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด 1 เมื่อมีสัญญาณสไตรบจากอุปกรณ์ภายนอก หมายความว่าอุปกรณ์ภายนอก ได้ทำการส่งข้อมูลมารอที่พอร์ทแล้ว ซึ่ง Z80 จะทำการรับค่านี้ผ่านพอร์ทดังกล่าวเข้าไป ซึ่งจะทำให้ข้อมูลในรีจิสเตอร์ภายในว่าง Z80 PIO จะทำการส่งสัญญาณเวคต์ออกมา เพื่อให้ อุปกรณ์ภายนอกทราบ แผนผังการทำงานเป็นดังรูปที่ 2.19



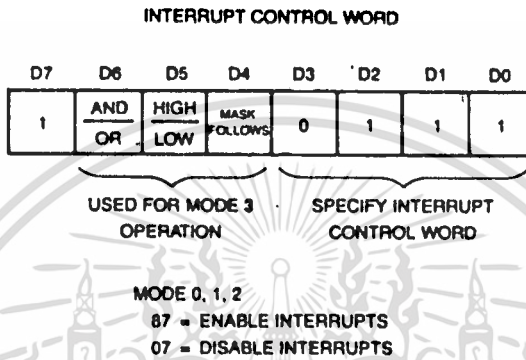
รูปที่ 2.19 แผนผังการทำงานแบบแฮนด์เชคของโหมด 1

การทำงานแบบแฮนด์เชคของทั้ง 2 โหมด นั้น สามารถใช้งานในโหมดอินเทอร์รัพท์ของ Z80 ได้ ถ้า Z80 PIO ถูกโปรแกรมให้ทำงานในโหมดนี้ โดย Z80 PIO จะเป็นตัวสร้างสัญญาณ INT\ เพื่อขออินเทอร์รัพท์ Z80 หลังจากขอบขาขึ้นของสัญญาณสไตรบ

### การโปรแกรมให้ Z80 PIO ทำงานในโหมดอินเทอร์รัพท์

Z80 PIO สามารถถูกโปรแกรมให้ทำงานร่วมกับ Z80 ในอินเทอร์รัพท์โหมด 2 ได้ ซึ่งเมื่อ Z80 PIO ถูกโปรแกรมเรียบร้อยแล้ว Z80 PIO จะทำการตอบสนองต่อสัญญาณสไตรบที่ขอบขาขึ้น โดยสร้างสัญญาณอินเทอร์รัพท์ขึ้น แล้วทำการส่งข้อมูลซึ่งเป็น 8 บิตล่างของแอดเดรสที่ใช้เป็นตารางอินเทอร์รัพท์เวคเตอร์ (Interrupt Vector Table) ซึ่งค่านี้จะต้องถูกโปรแกรมไว้ก่อนแล้ว ออกทางบัสข้อมูลเพื่อส่งให้กับ Z80 สำหรับแอดเดรส 8 บิตบนนั้น Z80 จะได้จากค่าที่อยู่ในรีจิสเตอร์ I ที่ถูกโปรแกรมอยู่ก่อนแล้วเช่นกัน

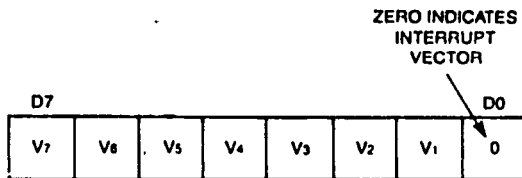
สำหรับการโปรแกรม Z80 PIO สามารถทำได้โดยการส่งข้อมูลที่ต้องการให้เป็น 8 บิตล่างของตารางอินเทอร์รัพท์เวคเตอร์ไปที่พอร์ทควบคุมก่อน แล้วส่งข้อมูลควบคุมอินเทอร์รัพท์ (Interrupt Control Word) ไปที่พอร์ทควบคุม ซึ่งข้อมูลควบคุมอินเทอร์รัพท์นี้จะใช้เป็นตัวอีนาเบิล หรือ ดิสเอเบิล การสร้างสัญญาณอินเทอร์รัพท์ โดยรายละเอียดของข้อมูลควบคุมอินเทอร์รัพท์นี้ สามารถแสดงได้ดังรูปที่ 2.20



รูปที่ 2.20 แสดงรายละเอียดของข้อมูลควบคุมอินเทอร์รัพท์

จากรูป 4 บิตล่างจะต้องมีค่า 7 หรือ 3 ซึ่งจะหมายความว่า เป็นการส่งข้อมูลสำหรับควบคุมอินเทอร์รัพท์ ส่วนบิตที่ 7 จะใช้เป็นตัวอีนาเบิลถ้ามีค่าเป็น 1 ถ้าเป็น 0 จะเป็นการดิสเอเบิล ดังนั้นถ้าต้องการอีนาเบิลจะต้องส่งข้อมูล 87h หรือ 83h ไปที่พอร์ทควบคุม ถ้าต้องการดิสเอเบิล จะต้องส่งข้อมูล 07h หรือ 03h ไปที่พอร์ทควบคุม

อนึ่ง การส่งข้อมูลสำหรับแอดเดรส 8 บิตล่าง ของตารางอินเทอร์รัพท์เวคเตอร์นั้น มีข้อกำหนดคือ ข้อมูลชุดนี้จะต้องมีบิตที่ 0 เป็น 0 เสมอ ซึ่งจะทำให้ Z80 PIO ทราบว่า เป็นการส่งข้อมูลแอดเดรส 8 บิตล่างของตารางอินเทอร์รัพท์เวคเตอร์ ดังรูปที่ 2.21



VECTOR TO BE PLACED ON DATA BUS WHEN AN INTERRUPT OCCURS

รูปที่ 2.21 แสดงข้อกำหนดของข้อมูลแอดเดรส 8 บิตล่างของตารางอินเทอร์รัพท์เวคเตอร์

### บทที่ 3

#### หลักการทํางาน

ข้อกำหนดเบื้องต้นสำหรับคุณสมบัติทั้งหมดของโครงการนี้ ได้แก่

1. สามารถทำการรับสัญญาณกริ่งโทรศัพท์ที่ได้โดยอัตโนมัติ
2. สามารถรับข้อมูลจากโทรศัพท์โดยการกดจากแป้นตัวเลขบนเครื่องโทรศัพท์ แล้วนำข้อมูลที่รับมาได้ มาทำการค้นหาข้อมูลที่ต้องการที่ถูกบันทึกอยู่แล้ว ได้อย่างอัตโนมัติ
3. สามารถนำข้อมูลที่ค้นหาได้แล้ว มาเปลี่ยนเป็นสัญญาณเสียงเพื่อส่งออกผ่านทางคู่สายโทรศัพท์ ไปยังผู้ที่ต้องการทราบข้อมูลได้
4. สามารถทำการบันทึกสัญญาณเสียง และทำการไหลคสัญญาณเสียงออกมาได้
5. มีระบบจัดการเกี่ยวกับการบันทึกข้อมูล, การเปลี่ยนแปลงแก้ไขข้อมูล และยูทิลิตี้ต่างๆ ซึ่งผู้ใช้ (User) สามารถใช้งานได้ง่าย

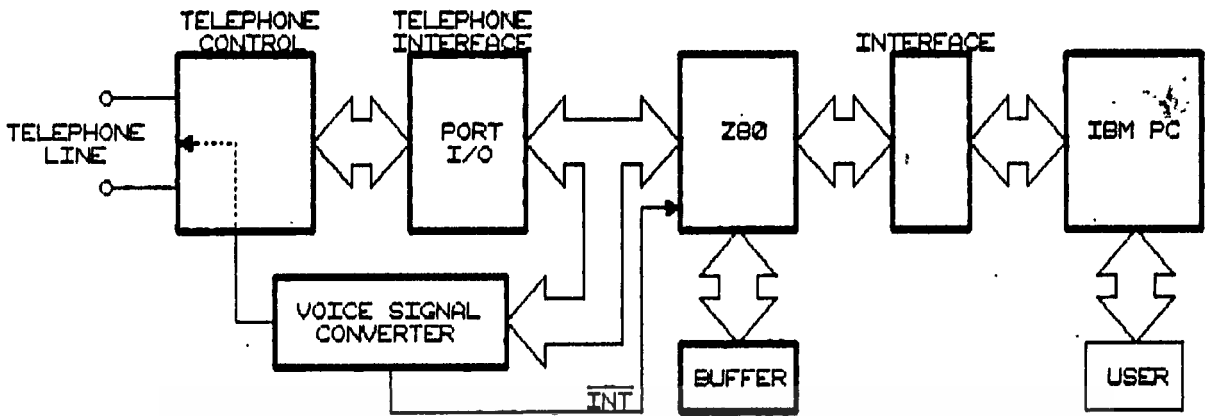
จากข้อกำหนดที่วางไว้ข้างต้น จะต้องทำการออกแบบและสร้างเพื่อให้มีคุณสมบัติดังกล่าว โดยสามารถแบ่งการออกแบบได้เป็นสองส่วน คือ

#### 3.1 ส่วนฮาร์ดแวร์ (Hardware Part)

#### 3.2 ส่วนซอฟต์แวร์ (Software Part)

#### 3.1 ส่วนฮาร์ดแวร์

หมายถึงส่วนของวงจรทั้งหมด ที่จะต้องถูกออกแบบให้สามารถทํางานได้ตามคำสั่งของซอฟต์แวร์ (Software) ซึ่งจากข้อกำหนดข้างต้น ทำให้สามารถนำมาเขียนเป็นแผนผังการทํางาน ได้ดังรูปที่ 3.1



รูปที่ 3.1 แผนผังการทำงานเบื้องต้นของส่วนฮาร์ดแวร์

จากแผนผังการทำงานข้างต้น ทำให้สามารถอธิบายการทำงานของส่วนฮาร์ดแวร์ (Hardware) ที่จะต้องกำหนดก่อนอย่างคร่าวๆ ได้ดังนี้

เริ่มแรก Z80 จะได้รับคำสั่งจาก IBM PC (ซึ่งถูกสั่งให้ทำงานอย่างอัตโนมัติ) ให้ทำการรอรับสัญญาณกริ่งจากโทรศัพท์ โดยการทำงานของวงจรในส่วนของวงจรควบคุมโทรศัพท์ (Telephone Control) เมื่อมีสัญญาณกริ่งโทรศัพท์จากคู่สายใดคู่สายหนึ่ง Z80 จะรับรู้ และทำการส่งงานให้วงจรในส่วนนี้ พร้อมทั้งจะส่งสัญญาณเสียงออกทางคู่สายโทรศัพท์ จากนั้น Z80 จะทำการไหลคข้อมูลสัญญาณเสียงที่ถูกบันทึกไว้แล้วจาก IBM PC ส่งผ่านเข้ามาทางวงจรส่วน Z80-IBM PC Interface แล้วทำการส่งผ่านออกวงจรแปลงสัญญาณเสียง (Voice Signal Converter) ออกทางคู่สายโทรศัพท์ที่ติดต่อกันอยู่ เพื่อบอกให้ผู้ใช้งานโทรศัพท์กดรหัสตัวเลขที่ต้องการทราบข้อมูล ซึ่ง Z80 จะทำการรับค่าเข้ามาโดยผ่านวงจรส่วน telephone Interface เมื่อ Z80 ได้รับรหัสข้อมูลครบถ้วนแล้ว ก็จะทำการส่งรหัสให้ IBM PC รับทราบ จากนั้น IBM PC จะทำการค้นหาข้อมูลตามรหัสที่ได้ เมื่อพบข้อมูลแล้ว ก็จะส่งข้อมูลสัญญาณเสียงที่ตรงกับข้อมูลที่พบนั้น ให้กับ Z80 เพื่อส่งออกผ่านวงจรแปลงสัญญาณเสียง เพื่อส่งเข้าคู่สายตามจังหวะของสัญญาณอินเทอร์วิทท์ที่วงจรแปลงสัญญาณเสียงขอมายัง Z80 เมื่อส่งสัญญาณเสียงเรียบร้อยแล้ว ก็จะสิ้นสุดการทำงาน โดย Z80 จะทำการรับคำสั่ง จาก IBM PC อีกครั้ง เพื่อรอสำหรับการทำงานใหม่

สำหรับรายละเอียดการออกแบบ และการทำงานของวงจรทั้งหมด จะแยกอธิบาย

ออกเป็นส่วนๆ วนบทต่อไป

### 3.2 ส่วนซอฟต์แวร์ (Software Part)

หมายถึงส่วนของโปรแกรมทั้งหมดที่ใช้ สำหรับควบคุมการทำงานของวงจร รวมทั้งโปรแกรมยูนิตีต่างๆ โดยจะแบ่งออกเป็นโปรแกรมที่เขียนสนับสนุนการทำงานของ IBM PC และ Z80 ซึ่ง

โปรแกรมบน IBM PC เขียนด้วยภาษาปาสคาล

โปรแกรมบน Z80 เขียนด้วยภาษาแอสเซมบลี

ข้อกำหนดเบื้องต้นสำหรับคุณสมบัติของโปรแกรมที่ใช้งานบน IBM PC เป็นดังนี้

- เป็นโปรแกรมที่ง่ายต่อการใช้งาน โดยจะต้องทำงานเป็นส่วนควบคุมหลัก
- ใช้ระบบ Pop-Up และ Pull-Down Menu
- มีโปรแกรมสำหรับการบันทึกข้อมูล และเปลี่ยนแปลงแก้ไขข้อมูล
- มีโปรแกรมสำหรับการบันทึกสัญญาณเสียง และไหลคสัญญาณเสียง
- ต้องสามารถติดต่อกับ Z80 ได้

สำหรับโปรแกรมในส่วนของ Z80 จะประกอบด้วย

- โปรแกรมติดต่อกับ IBM PC และโปรแกรมสนับสนุนการทำงานของ IBM PC
- โปรแกรมสำหรับการควบคุมการทำงานของวงจรทั้งหมด
- โปรแกรมบริการอินเทอร์เฟซสำหรับการรับและส่งข้อมูลสัญญาณเสียง

สำหรับรายละเอียด และการใช้งานโปรแกรม จะอธิบายในบทต่อไป

บทที่ 4

รายละเอียดการออกแบบและการทำงาน

สำหรับรายละเอียดการออกแบบและการทำงาน ของโครงงานนี้ สามารถแบ่งการอธิบายออกได้เป็น 2 ส่วน คือ

4.1 ส่วนฮาร์ดแวร์ จะกล่าวถึงขั้นตอนการออกแบบ และการทำงานทั้งหมด โดยจะแยกอธิบายออกเป็นส่วนๆ

4.2 ส่วนซอฟต์แวร์ จะกล่าวถึงลักษณะของอัลกอริทึม (Algorithm) และ ขั้นตอนการทำงานสำหรับโปรแกรมในส่วนของ Z80 และ IBM PC รวมทั้งการใช้งานโปรแกรม

4.1 ส่วนฮาร์ดแวร์

วงจรทั้งหมดจะใช้เครื่อง IBM PC เป็นส่วนควบคุมหลัก โดยจะทำงานร่วมกับ Z80 ซึ่งข้อกำหนดเบื้องต้นสำหรับวงจรทั้งหมด เพื่อให้ทำงานโดยได้รับการสนับสนุนจากซอฟต์แวร์สามารถอธิบายอย่างคร่าวๆ ได้ดังนี้

ก) IBM PC ใช้เป็นอุปกรณ์ติดต่อกับผู้ใช้ (User Interface) ซึ่งซอฟต์แวร์ในส่วนนี้จะต้องเขียนให้ผู้ใช้ สามารถใช้งานได้ง่าย โดยทำหน้าที่หลักเป็นตัวเก็บข้อมูล รับส่งข้อมูลกับ Z80 และที่สำคัญคือ ใช้เป็นตัวควบคุมการทำงานของวงจรทั้งหมด โดยผ่านทาง Z80

ข) Z80 ใช้เป็นตัวควบคุมวงจรรภายนอก โดยได้รับคำสั่งจาก IBM PC อีกทีหนึ่งซึ่งวงจรรภายนอกนี้ ประกอบด้วยวงจรดังต่อไปนี้

4.1.1 วงจรสำหรับการติดต่อกับโทรศัพท์ ทำหน้าที่รับสัญญาณกริ่งโทรศัพท์ รับรหัสตัวเลขจากโทรศัพท์ และติดต่อกับ IBM PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 วงจรสำหรับแปลงสัญญาณระหว่างสัญญาณเสียงกับสัญญาณดิจิทัล (Digital) ซึ่งประกอบด้วยวงจร 2 ส่วนคือ

- วงจรส่วนเคลตต้า มอดคูเลเตอร์/ดีมอดคูเลเตอร์ (Delta Modulator/Demodulator)
- วงจรส่วนที่ทำการแปลงสัญญาณ ระหว่างสัญญาณแบบขนาน และ สัญญาณแบบอนุกรม รวมทั้งวงจรสร้างสัญญาณซิงโครไนซ์ Z80

4.1.3 วงจรส่วนที่ใช้ในการติดต่อกันระหว่าง Z80 กับ IBM PC ใช้ในการรับส่งข้อมูลซึ่งกันและกัน ในลักษณะของการแฮนด์เชค

4.1.4 วงจรส่วนควบคุมการทำงาน (Circuit Control Part)

หมายเหตุ สำหรับวงจรที่ออกแบบและจัดสร้างนั้น ได้สนับสนุนการใช้งานสำหรับการทำงานใน 2 คู่สาย แต่จะต้องมีการพัฒนาในส่วนของซอฟต์แวร์อีกเพื่อให้การทำงานสมบูรณ์

4.1.1 วงจรสำหรับการติดต่อกับโทรศัพท์

วงจรในส่วนนี้ สามารถแบ่งการทำงานออกได้เป็น 2 ส่วนใหญ่ๆ คือ

1. วงจรส่วนควบคุมโทรศัพท์ (Telephone Control Part)
2. วงจรส่วนติดต่อกับวงจรควบคุมโทรศัพท์ (Telephone Control Interface)

วงจรส่วนควบคุมโทรศัพท์

หลักการการทำงานของวงจรส่วนควบคุมโทรศัพท์คือ เมื่อมีสัญญาณเสียงกริ่งเข้ามาวงจร ออปโตคัปเปิลเลอร์ (Optocoupler) จะทำหน้าที่เป็นตัวเชื่อมโยงระหว่างสัญญาณกริ่งโทรศัพท์ซึ่งมีค่าแรงดันสูงประมาณ 200 V<sub>AC</sub> กับวงจรนับซึ่งทำหน้าที่ในการรับเสียงกริ่ง 3 ครั้ง เมื่อนับได้ครบแล้ว Z80 จะรับรู้ และส่งสัญญาณไปบอก IBM PC ว่ามีการโทรศัพท์เข้ามา แล้ว พร้อมทั้งส่งสัญญาณเพื่อไปทำการสับสวิตช์ นำสายโทรศัพท์ต่อเข้ากับวงจร เพื่อทำการติดต่อกันโดยตรง Z80 จะรอการกดรหัสคำสั่งเข้ามา และก็จะทำตามคำสั่งนั้น จนจบงานการทำงานเสร็จสิ้น (การทำงานนี้ Z80 จะติดต่อกับ IBM PC ตลอดเวลา)

ซึ่งรายละเอียดในส่วนย่อยต่างๆ สามารถอธิบายได้ดังนี้

ออปโตคัปเปิลอร์ (Optocoupler)

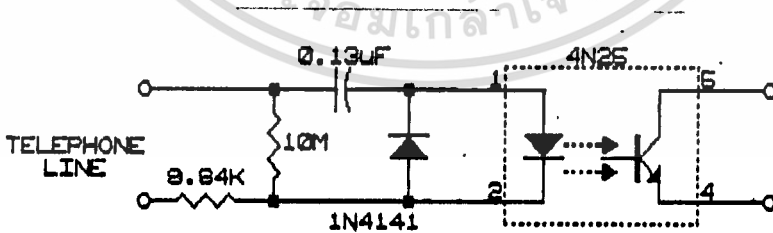
การนำเอาออปโตคัปเปิลอร์เข้ามาใช้ในวงจร ก็เพื่อจุดประสงค์ในการแยกส่วนของแรงดันไฟสูงที่เกิดจากสายโทรศัพท์ซึ่งมีค่าประมาณ 200 V<sub>AC</sub> ในขณะที่มีสัญญาณกริ่งดังเข้ามาทำให้เชื่อมต่อกับวงจรที่มีระดับไฟเลี้ยงที่มีระดับ 5 V<sub>DC</sub> ซึ่งตัวออปโตคัปเปิลอร์นี้จะมีความสำคัญอย่างยิ่งที่จะต้องนำมาใช้งานเพื่อในวงจรที่มีระดับแรงดันต่างกันมาก สามารถทำงานร่วมกันได้ ในส่วนของวงจรมันใช้ ออปโตคัปเปิลอร์ เบอร์ 4N25

สำหรับรายละเอียดของ 4N25 เป็นดังนี้

- วงจรมีโครงสร้างคล้ายกับ TTL มาตรฐาน
- ใช้ GaAs ไดโอดซึ่งปล่อยแสงอินฟราเรดไปยัง ซิลิกอน NPN ไฟด์ทรานซิสเตอร์
- การแปลง (Transfer) ของไฟกระแสตรงมีค่าสูง
- สามารถแยกวงจรที่มีระดับแรงดันที่ต่างกันได้ถึง + 2.5 KV
- ความเร็วในการสวิตซ์ซิงค์สูง โดยมีค่า

Rise time( $t_r$ ) = 2 us.

Fall time( $t_f$ ) = 2 us.



รูปที่ 4.1 วงจรใช้งานจริงในส่วนออปโตคัปเปิลอร์

จากรูป เป็นวงจรที่ใช้งานการทำงานจริง ซึ่งอุปกรณ์แต่ละตัว มีหน้าที่ดังนี้

- ค่าความต้านทาน 10M และ 9.64K ทำหน้าที่เป็นตัวแบ่งแรงดัน (Voltage Divider) ให้ได้ขนาดที่เหมาะสมเพื่อทำการขับ (Drive) อินฟราเรดไดโอด

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือที่สงวนไว้เพื่อการศึกษาเท่านั้น เมื่อผู้ใดเห็นประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

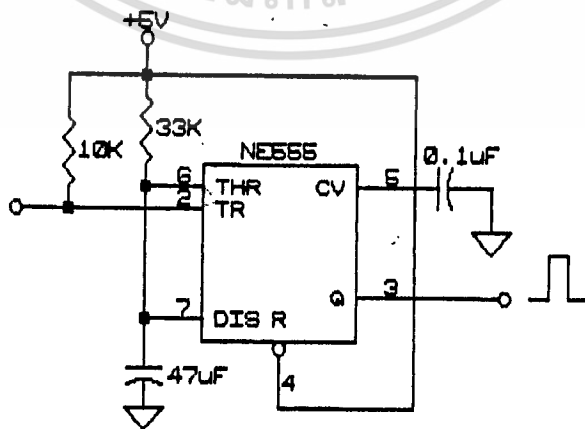
- ค่าตัวเก็บประจุ 0.13  $\mu\text{F}$  ทำหน้าที่กั้นไฟกระแสดตรง ไม่ให้ไดโอดอินฟาเรดทำงานเนื่องจากไม่ต้องการให้ไฟไต้ทรานซิสเตอร์ (Phototransistor) ทำงาน (Active) ในสภาวะปกติ แต่จะให้ทำงานในกรณีที่มีสัญญาณกริ่งคือสัญญาณ AC เท่านั้น

วงจรโมโนสเตเบิล (Monostable)

เป็นวงจรที่เชื่อมต่อกับไฟไต้ทรานซิสเตอร์ เพื่อทดสอบสัญญาณกริ่งของโทรศัพท์ที่โทรเข้ามา สัญญาณกริ่ง 1 ครั้งจะทำให้วงจรโมโนสเตเบิลสร้างสัญญาณทริก (Trig) 1 ครั้งตามจังหวะของเสียงกริ่ง สำหรับวงจรโมโนสเตเบิลของโครงการนี้ ใช้ไอซีเบอร์ NE555 (Timer)

รายละเอียดของ NE555 เป็นดังนี้

- เป็นไอซีที่มีการผลิต time delay และออสซิลเลเตอร์ได้ค่อนข้างจะถูกตองโดยในส่วนของ time delay จะใช้ร่วมกับอุปกรณ์ภายนอก 2 ตัวคือ ตัวความต้านทาน และตัวเก็บประจุ เพื่อทำการกำหนดค่า time delay เท่านั้น
- ปรับค่า duty cycle ได้
- เอาท์พุทขา 3 เป็นได้ทั้งจ่ายกระแสและดึงกระแสประมาณ 200mA
- เอาท์พุทและไฟเลี้ยงมีระดับแรงดันเท่ากับ TTL มาตรฐาน



รูปที่ 4.2 วงจรโมโนสเตเบิลที่ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป เป็นวงจรโมโนสเตเบิลที่ใช้งานจริง ซึ่งสามารถอธิบายได้ดังนี้

- ค่าของความต้านทาน 33K และตัวเก็บประจุ 47 uF เป็นตัวกำหนดความกว้างของพัลส์ (Pulse) โดยมีค่า

$$t = 1.1RC \quad \dots(4.1)$$

$$t = 1.1 \times (33 \times 10^3) \times (47 \times 10^{-6})$$

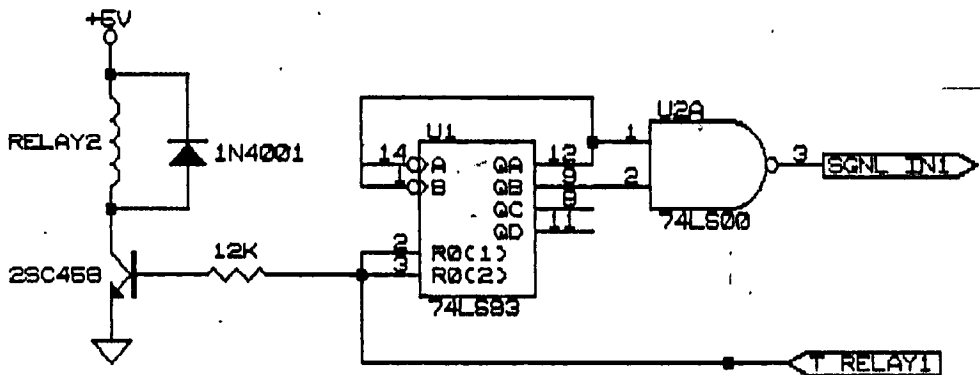
$$t = 1.7061 \text{ sec.}$$

- ขารีสตาร์ท(Reset) ทำหน้าที่รีเซ็ตไอซี NE555 ให้อยู่ในสภาวะที่จะทำงานในที่ที่ถูกต่อกับระดับแรงดันไฟ 5 โวลต์

- ขาทรigger (Trigger) ทำหน้าที่รอรับสัญญาณทรiggerโดยจะทรiggerที่ขอบสัญญาณขาลง (Tailing edge) เท่านั้น

วงจรมับสาม (Counter) และ วงจรรับสายโทรศัพท์

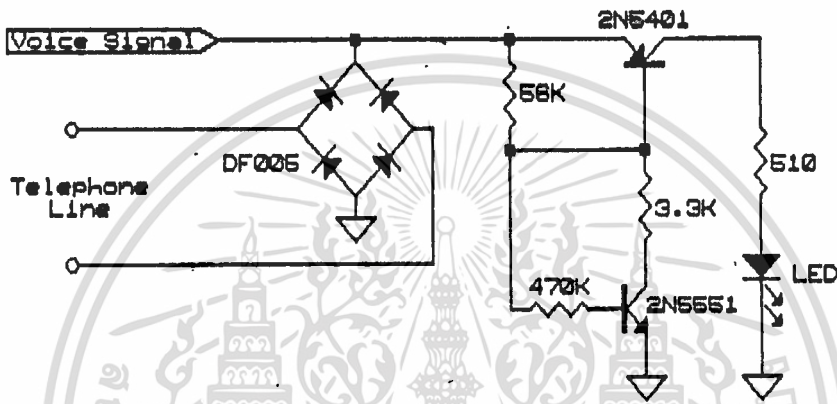
ทำหน้าที่ในการนับพัลส์ที่ส่งมาจากเอาต์พุตของวงจรมอนอสเตเบิล เมื่อมีสัญญาณกริ่งเข้ามา 3 ครั้งวงจรมอนอสเตเบิลจะทรigger โดยการส่งพัลส์ออกมา 3 ลูกเข้าไปในวงจรมับ เมื่อวงจรมับนับถึง 3 ครั้งเมื่อเดคา 9 และ 12 ของวงจรมับ (74LS93) อยู่ในระดับลอจิก 1 จะทำให้อาต์พุตที่ออกมาจากแชนแนล (74LS00) มีลอจิกเป็น 0 ซึ่งจะเป็สัญญาณที่เข้าไปบอก Z80 ว่ามีโทรศัพท์เข้ามาติดต่อแล้ว เมื่อ Z80 ตรวจสอบสัญญาณนี้ (SGNL IN1) Z80 ก็จะทำการส่งสัญญาณมาทรiggerทรานซิสเตอร์ให้แอคทิฟ เพื่อให้รีเลย์ (Relay) ทำการติดต่อคู่สายเข้ากับวงจร รวมทั้งทำการเคลียร์ (Clear) วงจรมับ โดยวงจรทั้งหมดเป็นดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้รูปที่ 4.3 วงจรมับสามและวงจรมับสายโทรศัพท์ ห้าไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ T RELAY1 เป็นสัญญาณที่ถูกส่งออกมาจาก Z80 ผ่านทางพอร์ตเอาต์พุตเพื่อทำการ ON ทรานซิสเตอร์ กระแสจะไหลผ่านขดลวดของรีเลย์ ทำให้รีเลย์อยู่ในสภาวะ ON สับสวิตช์สายโทรศัพท์เข้าวงจรเรกติไฟเออร์เต็มลูก (Full Wave Bridge Rectifier)

วงจรวัดจ็และแอกทีฟโหลด (Bridge and Active Load)



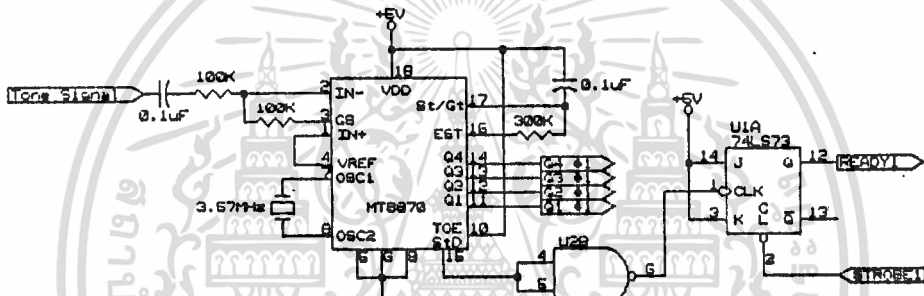
รูปที่ 4.4 วงจรวัดจ็และแอกทีฟโหลด

เนื่องจากสายโทรศัพท์ไม่มีระดับแรงดันกราวน์ (Ground) เปรียบเทียบ ดังนั้นเมื่อสายโทรศัพท์ถูกรีเลย์สับสวิตซ์ไปต่อที่วงจรวัดจ็ วงจรวัดจ็จะทำหน้าที่ในการสร้างระดับกราวน์เปรียบเทียบ โดยสัญญาณที่ออกจากวงจรวัดจ็จะมีลักษณะเช่นเดิม เนื่องจากว่า สัญญาณเสียงที่เข้ามาถูกยกระดับด้วยระดับแรงดันไฟตรงง้ให้อยู่เหนือระดับกราวน์ในขณะการพูดโทรศัพท์โดยทางชุมสายโทรศัพท์อยู่แล้ว วงจรวัดจ็ในโครงการง้ใช้ไอซีบริดจ์เบอร์ DF005

วงจรวัดจ็แอกทีฟโหลด เป็นโหลดที่จะต้องมีค่าความต้านทานประมาณเท่ากับค่าความต้านทานที่ทางชุมสายตรวจสอบได้จากการยกสายโทรศัพท์ ในกรณีที่ได้อ่าโหลดที่เหมาะสมแล้ว แอกทีฟโหลดจะเป็นตัวเสมือนหลอกง้ให้ทางชุมสายโทรศัพท์ว่ามีการรับสายโทรศัพท์แล้ว ซึ่งจะกอ้ให้เกิดการคิดค่าบริการทง้กับคู่สายค้ันทาง

วงจรแปลงความถี่ของระบบโทน (Tone) ของโทรศัพท์เป็นตัวเลขฐานสอง 4 บิต

เนื่องจากโครงงานชิ้นนี้ต้องการติดต่อกันระหว่างคอมพิวเตอร์และโทรศัพท์ โดยมีแนวความคิดว่าเมื่อโทรศัพท์โทรเข้ามาติดต่อกับคอมพิวเตอร์ ระบบจะทำการตรวจสอบว่ามีริงเข้ามาครบ 3 ครั้งหรือยัง ถ้าครบ คอมพิวเตอร์จะสั่งงาน ให้ระบบทำการรับสายโทรศัพท์ และจะรอการกรอกรหัสบนแป้นโทรศัพท์ของผู้ที่โทรศัพท์เข้ามาว่ามีรหัสเท่าไร ซึ่งแต่ละรหัสจะมีความหมายแตกต่างกันออกไปเป็นค่าที่ถูกกำหนดไว้ล่วงหน้า ซึ่งผู้จะใช้จะต้องทราบก่อน



รูปที่ 4.5 วงจรเข้ารหัสเลขฐานสอง 4 บิต

สำหรับการทำงานของวงจรสามารถอธิบายได้ดังนี้

จากรูปใช้ไอซีเฉพาะงานเบอร์ MT8870 (Integrated DTMF Receiver) ทำหน้าที่ในการแปลงความถี่ของสัญญาณที่ถูกผลิตออกมาจากการกดปุ่มบนแป้นโทรศัพท์ระบบโทน ถูกส่งมาทางสายโทรศัพท์เข้าวงจรบริดจ์และเข้าไอซีเบอร์ MT8870 ไอซี MT8870 จะทำการแปลงความถี่ที่ได้ออกมาในรูปสัญญาณดิจิทัล 4 บิตที่ขา 11-14(Q1-Q4) และสร้างสัญญาณ Std ออกมาให้วงจรที่ต่อร่วมทราบว่าข้อมูลถูกแปลงเรียบร้อยแล้วมานำรับไปได้

รายละเอียดของ MT8870 เป็นดังนี้



- OSC1 อินพุทของสัญญาณนาฬิกา
- OSC2 เอาท์พุทของสัญญาณนาฬิกา โดยใช้กับผลึก (Crystal) ที่มีความถี่ 3.5795 MHz ต่อเข้าระหว่าง OSC1 กับ OSC2 เพื่อจ่ายให้กับออสซิลเลเตอร์ (Oscillator) ที่อยู่ภายใน
- VSS ไฟเลี้ยงขั้วลบ
- TOE ใช้ควบคุม 3-state output enable กรณีที่มีค่าลอจิก 1 ข้อมูลของ Q1-Q4 จะออกมาแสดงผลข้างนอกได้ ถ้าเป็น 0 จะแสดงสภาวะไฮอิมพีแดนซ์ (High impedance)
- Q1-Q4 เป็นขาเอาท์พุทที่ถูกควบคุมแบบ 3-state ด้วยสัญญาณ TOE จะให้ค่าออกมาเมื่อได้รับค่าความถี่ที่ต้องการ และเอาท์พุทถูกฮิยาเบิล
- Std Delay steering output ให้ลอจิก 1 เมื่อข้อมูลที่ Q1-Q4 ถูกต้องแล้ว และจะมีค่ากลับลงเป็น 0 เมื่อระดับแรงดันของขา St/GT ต่ำกว่าค่า  $V_{Tst}$
- EST Early steering output จะให้ค่าลอจิก 1 เมื่อตรวจสอบพบความถี่ที่ต้องการที่ถูกกดเข้ามา (ดูตารางความถี่ข้างล่างได้) จะลดลงเป็นลอจิก 0 เมื่อไม่มีสัญญาณโทนเข้ามา
- St/GT Steering input/guard time output (bi-directional) ท่าหน้าที่ส่งสัญญาณควบคุมวงจร RC ภายนอกเพื่อควบคุมไทม์มิ่ง (Timing)
- VDD ไฟเลี้ยงค่าบวก

จากรูปวงจรที่ใช้งาน สามารถแบ่งหัวข้อในการอธิบายได้ดังนี้

- เราจะได้ค่าเกนเท่ากับ 1 เนื่องจาก เราต่อความต้านทาน 100K 2 ตัวเข้าที่ขา 2 และ 3 โดยความต้านทานที่ต่อที่ขา 3 เป็นความต้านทานที่ถูกต้องแบบการป้อนกลับแบบลบ (Negative feedback)
- $V_{REF}$  ถูกต่อเข้าขา IN+ เพื่อให้ระดับสัญญาณที่เข้ามา ถูกไบอัสให้อยู่ที่กึ่งกลางของระดับไฟเลี้ยงเท่ากับ  $V_{DD}/2$
- คริสตัล 3.579575 MHz ถูกต่อคร่อมระหว่าง OSC1 และ OSC2 เพื่อให้เกิดการผลิตสัญญาณนาฬิกาภายในไอซี

- ค่าความต้านทาน 300K และค่าความจุของตัวเก็บประจุของตัวเก็บประจุ 0.1uF ทำหน้าที่ในการกำหนด guard time ซึ่งเราสามารถที่จะเปลี่ยนแปลงเพิ่มหรือลดค่าของ guard time ได้ดังสมการ

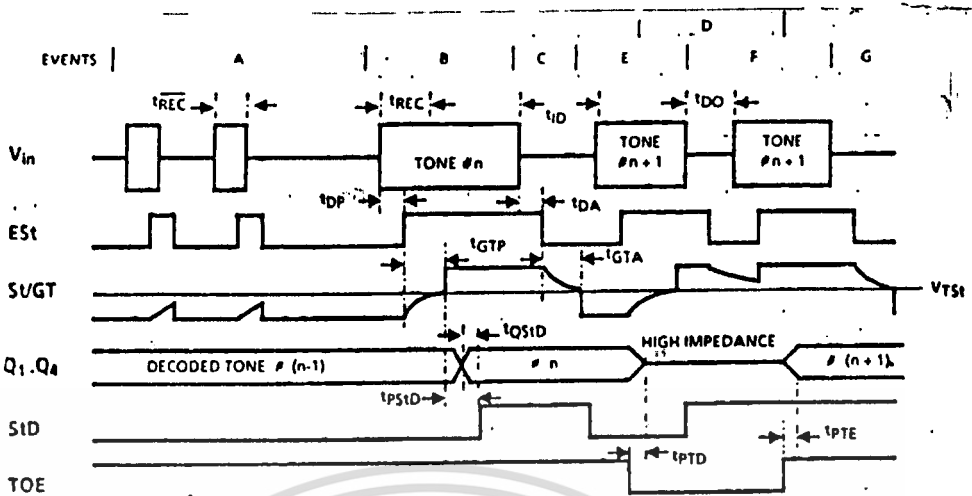
$$T_{GTA} = (RC) \ln [ V_{DD}/V_{Tst} ] \quad \dots(4.2)$$

$$T_{GTP} = (RC) \ln [ V_{DD}/(V_{DD}-V_{Tst}) ] \quad \dots(4.3)$$

- ขา TOE ถูกทำให้ enable บัฟเฟอร์แบบ 3-state ของ Q1-Q4 ตลอดเวลา

F <sub>LOW</sub>	F <sub>HIGH</sub>	NO	TOE	Q <sub>4</sub>	Q <sub>3</sub>	Q <sub>2</sub>	Q <sub>1</sub>
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1336	0	H	1	0	1	0
941	1209	.	H	1	0	1	1
941	1477	#	H	1	1	0	0
697	1633	A.	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0
-	-	ANY	L	Z	Z	Z	Z

เอกสารนี้เป็นเอกสารที่ ตารางที่ 4.1 ตารางแสดงค่าที่ค่าต่างๆที่ถูก decode ออกมาได้ โยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 แผนผังเวลาของ MT8870

จากแผนผังเวลาดังกล่าว สามารถอธิบายช่วงของเวลาได้ดังนี้

- A) ช่วงเวลาของสัญญาณโทน (Tone) มีค่าผิดพลาดเนื่องจากเวลาน้อยเกินไปเอาท์พุทจะไม่มี การเปลี่ยนแปลง
- B) โทน #n ถูกตรวจสอบพบและช่วงเวลาถูกต้อง สัญญาณโทนจะถูก decode และถูกแลตช์ (Latch) ไว้ที่เอาท์พุท
- C) ไอซีตรวจสอบพบสัญญาณโทนสิ้นสุด เอาท์พุทยังคงรักษาข้อมูลเดิมไว้ จนกว่าจะมี โทนที่ถูกต้องเข้ามาใหม่
- D) เอาท์พุทถูกทำให้อยู่ในสภาวะไฮอิมพีแดนซ์
- E) โทน #n+1 ถูกตรวจสอบและช่วงเวลาถูกต้องสัญญาณโทนจะถูก decode และถูกแลตช์ไว้ที่เอาท์พุท (ตอนนี้อยู่ในสภาวะ high impedance)
- F) ช่วงของโทน #n+1 ที่ขาดหายไปที่ยอมรับได้ เช่น เนื่องจากสัญญาณรบกวน (Noise) ทำให้ความถี่เปลี่ยนไป เอาท์พุทจะยังคงค่าเดิม
- G) โทน #n+1 ถูกตรวจสอบพบว่าสิ้นสุด เอาท์พุทยังคงรักษาข้อมูลเดิมไว้จนกว่าจะมี โทนใหม่ที่ถูกต้องเข้ามา

สำหรับสัญลักษณ์ต่างๆที่ใช้ มีความหมายดังนี้

$t_{REC}$  = ช่วงเวลาอย่างน้อยที่สุดที่โทนจะถูกตรวจพบ (Detect) ได้อย่างถูกต้อง

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของ บริษัท ทรานส์เทคโนโลยี จำกัด ขอสงวนสิทธิ์ในเนื้อหา ไม่สามารถนำออกเผยแพร่โดยไม่ได้รับอนุญาตจากบริษัทฯ

$t_{DO}$  = ช่วงเวลามากที่สุดที่ถูกลอยมือรับได้ในกรณีสัญญาณโชนถูกทำให้หายไปชั่วขณะ

$t_{DP}$  = เวลาที่ตรวจสอบพบการมาของสัญญาณที่ต้องการ

$t_{DA}$  = เวลาที่ตรวจสอบพบการหายไปของสัญญาณที่ต้องการ

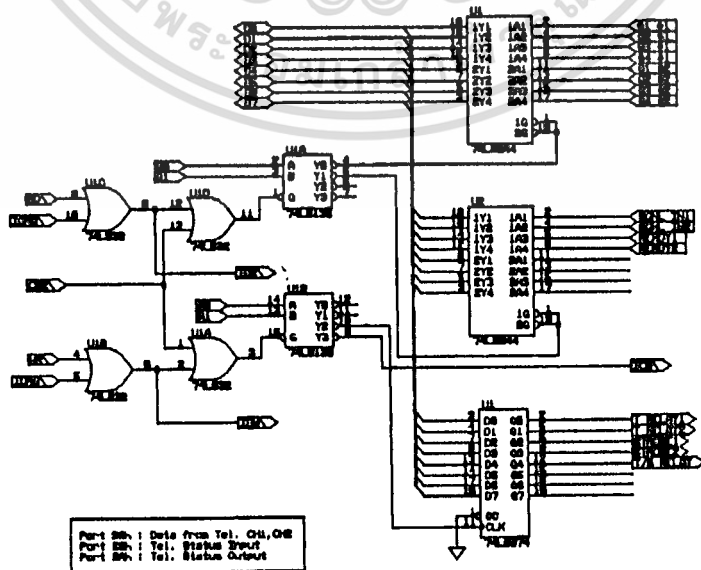
$t_{GTP}$  = สัญญาณ guard time ปรากฏ

$t_{GTA}$  = สัญญาณ guard time ไม่ปรากฏ

อนึ่ง สัญญาณ StD จะต้องถูกต่อกับฟลิปฟลอปแบบเจเค (JK Flip-Flop) เบอร์ 74LS73 ซึ่งถูกต่อวงจรให้ใช้งานในลักษณะของฟลิปฟลอปแบบที (T Flip-Flop) เพื่อการทำงานงานแบบแฮนด์เชคระหว่าง Z80 กับ MT8870 โดยสัญญาณจากขา Q จะใช้เป็นสัญญาณข้อมูลพร้อม (Data Ready) และสัญญาณเข้าขา CLR จะใช้เป็นสัญญาณสไตรบจาก Z80 เพื่อบอกว่าได้ทำการรับข้อมูลเข้าไปแล้ว

### วงจรส่วนติดต่อกับวงจรควบคุมโทรศัพท์

วงจรมีส่วนนี้ จะใช้สำหรับการรับส่งข้อมูลระหว่าง Z80 กับ วงจรควบคุมโทรศัพท์ โดยวงจรมีส่วนนี้ จะเป็นวงจรพอร์ทอินพุทเอาต์พุทสำหรับ Z80 ซึ่งมีลักษณะการต่อวงจรเป็นดังรูปที่ 4.8



เอกสารนี้เป็นเอกสารที่สงวนไว้รูปที่ 4.8 วงจรส่วนติดต่อกับวงจรควบคุมโทรศัพท์ นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูป เบอร์พอร์ทและรายละเอียดหน้าที่ของแต่ละพอร์ทที่เลือกใช้ เป็นดังนี้

พอร์ท 28h เป็นพอร์ทอินพุท โดยใช้ไอซี 74LS244 สำหรับรับข้อมูลเลขฐานสองที่ได้จากการตีโค้ดจาก MT8870 โดย 4 บิตล่างจะเป็นของช่อง (Channel) ที่ 1 และ 4 บิตบนจะเป็นของช่องที่ 2

พอร์ท 29h เป็นพอร์ทอินพุท โดยใช้ไอซี 74LS244 สำหรับตรวจจับสัญญาณที่ได้จากวงจรของ MT8870 ซึ่งการตรวจจับสัญญาณ Z80 จะทำงานตามคำสั่งซอฟต์แวร์ ในลักษณะของการรับข้อมูลจากพอร์ทนี้ แบบวนลูบจนกว่าสัญญาณที่ต้องการเกิดขึ้น

หน้าที่ของแต่ละบิต เป็นดังนี้

บิตที่ 0,1 ใช้รับสัญญาณกริ่งโทรศัพท์ของช่องที่ 1 และ 2 ตามลำดับ

บิตที่ 2,3 ใช้รับสัญญาณข้อมูลพร้อมจากช่อง ที่ 1 และ 2 ตามลำดับ

พอร์ท 2Ah เป็นพอร์ทเอาต์พุท โดยใช้ไอซี 74LS374 สำหรับให้ Z80 ส่งสัญญาณต่างๆ เพื่อควบคุมวงจรวางควบคุมโทรศัพท์

หน้าที่ของแต่ละบิต เป็นดังนี้

บิตที่ 0,1 ใช้ส่งสัญญาณเพื่อทำการรับสายโทรศัพท์และเคลียร์วงจรมับ สำหรับช่องที่ 1 และ 2 ตามลำดับ

บิตที่ 2,3 ใช้ส่งสัญญาณสไตรบ์ให้กับ MT8870 สำหรับช่องที่ 1 และ 2 ตามลำดับ

บิตที่ 4 ใช้ส่งสัญญาณเพื่อทำการสับสวิทช์เลือกว่าต้องการให้เสียงออกทางลำโพงหรือทางคู่สายโทรศัพท์

#### 4.1.2 วงจรแปลงสัญญาณระหว่างสัญญาณเสียงกับสัญญาณดิจิทัล

สำหรับหลักการคร่าวๆของวงจรในส่วนนี้ก็คือ

##### 1. แปลงสัญญาณเสียงเป็นสัญญาณดิจิทัล

ในส่วนนี้ วงจรจะทำการแปลงสัญญาณเสียงโดยวิธีของ เกลต้ามอดคูลเลชัน ซึ่งจะแปลงเป็นสัญญาณดิจิทัลแบบอนุกรม จากนั้นสัญญาณที่ได้จะถูกนำมาแปลงเป็นสัญญาณแบบขนาน ด้วยวงจรอีกส่วนหนึ่ง

##### 2. แปลงสัญญาณดิจิทัลเป็นสัญญาณเสียง

ในส่วนนี้ วงจรจะทำการแปลงสัญญาณดิจิทัลซึ่งเป็นแบบขนาน ไปเป็นแบบอนุกรม จากนั้นสัญญาณที่ได้ จะถูกนำมาแปลงเป็นสัญญาณเสียงโดยวิธี เกลต้ามอดคูลเลชัน

ซึ่งวงจรทั้งหมดสามารถแบ่งการอธิบายออกได้เป็น 2 ส่วนดังนี้

##### วงจรส่วนเกลต้ามอดคูลเลเตอร์/ดีมอดคูลเลเตอร์

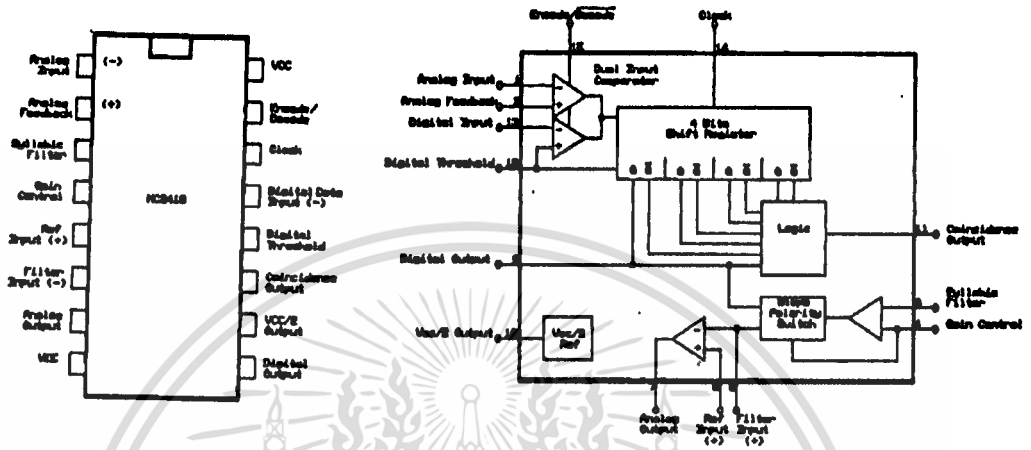
หัวใจสำคัญของวงจรในส่วนนี้คือ ไอซีเบอร์ MC3418 (Continuously Variable Slope Delta Modulator/Demodulator) ของโมโตโรล่า (Motorola) ซึ่งจะทำหน้าที่มอดคูลเลตสัญญาณเสียง ให้เป็นสัญญาณดิจิทัลแบบอนุกรมและดีมอดคูลเลตสัญญาณดิจิทัลแบบอนุกรมเป็นสัญญาณเสียง ซึ่งสามารถทำได้ทั้ง 2 ฟังก์ชันในตัวเดียวกัน โดยการควบคุมที่ขา 15 (Encode/Decode) ซึ่งลักษณะการต่อของวงจร จะเป็นดังรูปที่ 3.14

รายละเอียดของไอซี MC3418 เป็นดังนี้

- เป็นไอซีที่สร้างขึ้นเพื่อใช้มอดคูลเลตสัญญาณในช่วงความถี่เสียง
- มีระบบมอดคูลเลชันและดีมอดคูลเลชันในตัวเดียวกัน
- สามารถใช้งานร่วมกับไอซีประเภทซีมอส (CMOS) และทีทีแอล ได้

- การจ่ายไฟเลี้ยงสามารถกระทำได้ 2 แบบ คือ

1. ไฟเลี้ยงคู่ (Dual Supply) คือ ใช้ไฟบวกร่วมกับไฟลบ
2. ไฟเลี้ยงเดี่ยว (Single Supply) คือ ใช้ไฟบวกกับกราวนด์



ก) การวางขา

ข) แผนผังภายใน

รูปที่ 4.9 รายละเอียดของไอซี MC3418

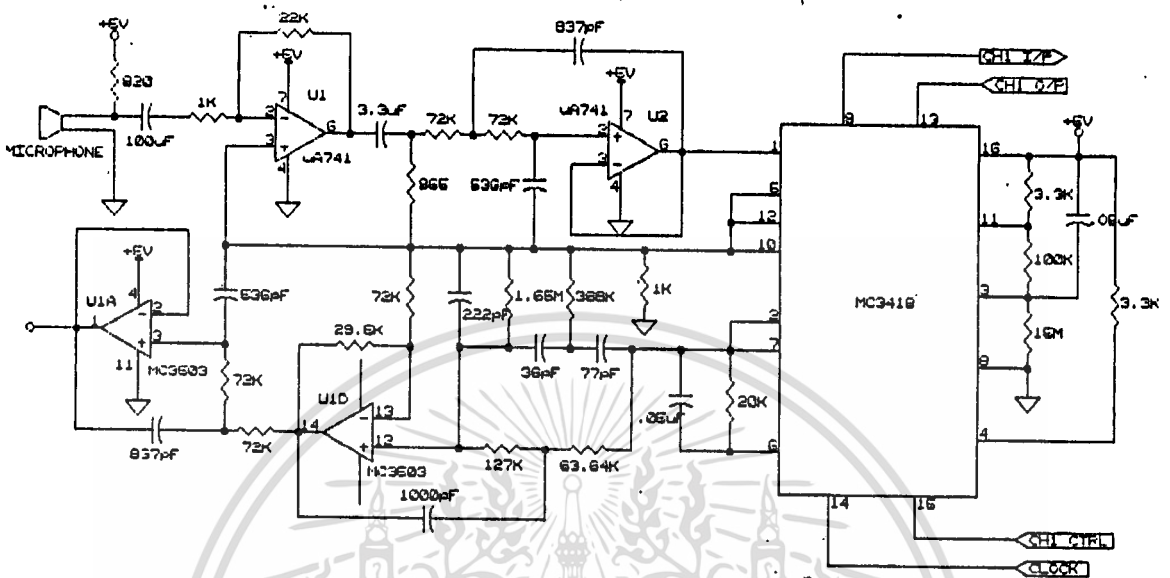
จากรูปข้างต้นสามารถอธิบายถึงรายละเอียดและหน้าที่ของสัญญาณในแต่ละขาได้ดังนี้

- ขา 1 : Analog Output เป็นขาที่รับสัญญาณเสียง ซึ่งจากแผนผังภายในจะเห็นว่า เป็นขาอินเวอร์ตติ่งของวงจรเปรียบเทียบแรงดัน (Voltage Comparator)
- ขา 2 : Analog Feedback เป็นขาอินเวอร์ตติ่งของวงจรเปรียบเทียบแรงดัน ซึ่งทำหน้าที่ในการนำสัญญาณกลับมาเปรียบเทียบ กับสัญญาณอินพุตที่จะเข้ามาในช่วงนาฬิกาถัดไป
- ขา 3 : Syllabic Filter เป็นขาอินพุต ที่รับสัญญาณจากวงจรกรองสัญญาณภายนอก เพื่อนำมาควบคุมขนาดของสัญญาณในแต่ละขั้นของส่วนที่เป็นอินทิเกรเตอร์
- ขา 4 : Gain Control Input เป็นขาอินพุตที่รับสัญญาณเพื่อควบคุมอัตราขยายของออปแอมป์ภายในตัวไอซี

- ขา 5 : **Reference Input** เป็นขาอนอินเวอร์ทติ้งของออปแอมป์ เพื่อใช้เป็นแรงดันอ้างอิงทางค่านีของสัญญาณเอาต์พุต
- ขา 6 : **Filter Input** เป็นขาอนอินเวอร์ทติ้งของออปแอมป์ เพื่อใช้ต่อกับวงจรภายนอก สำหรับสร้างเป็นวงจรอินทิเกรเตอร์
- ขา 7 : **Analog Output** เป็นขาเอาต์พุตของวงอินทิเกรเตอร์ที่สร้างขึ้น ซึ่งจะใช้เป็นสัญญาณเอาต์พุตในขณะที่ไอซีตัวนี้ทำงานเป็นวงจรถอดรหัส และเป็นสัญญาณป้อนกลับเข้าขา 2 ในกรณีทำงานเป็นวงจรเข้ารหัส
- ขา 8 : **V<sub>EE</sub>** เป็นขาที่ใช้สำหรับต่อกับไฟเลี้ยงที่มีค่าต่ำสุดที่จ่ายให้กับไอซีตัวนี้
- ขา 9 : **Digital Output** เป็นขาเอาต์พุตที่ได้มาจากการมอดคูเลทสัญญาณอนาลอกซึ่งสัญญาณนี้ จะมีการแกว่ง (Swing) อยู่ระหว่าง V<sub>CC</sub> กับ V<sub>EE</sub>
- ขา 10 : **V<sub>CC</sub>/2 Output** เป็นขาเอาต์พุตที่ถูกสร้างขึ้น เพื่อให้เกิดแรงดันที่มีค่าครึ่งหนึ่งของแรงดัน V<sub>CC</sub> ซึ่งจะถูกนำไปใช้งานในกรณีของการจ่ายไฟเลี้ยงเดี่ยว
- ขา 11 : **Coincidence Output** เป็นขาเอาต์พุตที่แสดงว่าชิฟรีจิสเตอร์ (Shift Reg) ภายใน ถูกเก็บด้วย 0 หรือ 1 ทั้งหมด ซึ่งสัญญาณนี้จะถูกนำไปใช้ในการ กำหนดอัตราการขยายสัญญาณของวงจรสร้างระดับสัญญาณ
- ขา 12 : **Digital Threshold** เป็นขาอินพุตเพื่อใช้เลือกว่าจะใช้งานกับไอซีประเภทใด
- ขา 13 : **Digital Data Input** เป็นขาอินพุต เพื่อรับสัญญาณดิจิทัลแบบอนุกรมจากภายนอกในการแปลงเป็นสัญญาณอนาลอก ใช้ความถี่ 16 KHz
- ขา 14 : **Clock Input** เป็นขาอินพุตที่รับสัญญาณนาฬิกา เพื่อใช้กำหนดจังหวะการทำงานภายในตัวไอซี ซึ่งความถี่ของสัญญาณนาฬิกาจะเป็นความถี่ที่ใช้สำหรับการแซมปลิงสัญญาณอนาลอกที่ต้องการมอดคูเลท
- ขา 15 : **Encode/Decode** เป็นขาที่ใช้ในการควบคุมการทำงานของไอซี ถ้าเป็น 1 ก็จะทำงานเป็นตัวเข้ารหัส ถ้าเป็น 0 ก็จะทำงานเป็นตัวถอดรหัส
- ขา 16 : **V<sub>CC</sub>** เป็นขาที่รับแรงดันไฟเลี้ยงค่าสูงสุดของไอซี ซึ่งจะมีค่าอยู่ระหว่าง 4.75 ถึง 16.5 โวลท์

สำหรับลักษณะการต่อวงจรเพื่อนำไปใช้งาน เป็นดังรูปที่ 4.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

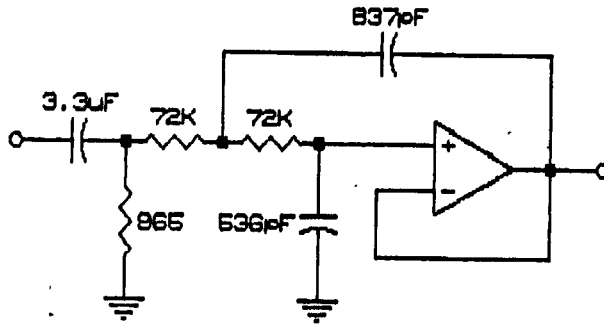


รูปที่ 4.10 วงจรส่วนเคลต้า มอดคูเลเตอร์/คิมมอดคูเลเตอร์

จากรูปข้างต้น จะสามารถแบ่งการทำงานของวงจรออกได้เป็น 2 กรณี คือ

ก) กรณีมอดคูเลท ใช้สำหรับคอนแปลงสัญญาณเสียงเป็นสัญญาณดิจิทัล ซึ่งขั้นตอนการทำงานของวงจรจะเป็นดังนี้

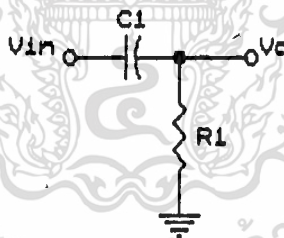
สัญญาณเสียงจะถูกส่งผ่านทางไมโครโฟน เข้าสู่วงจรปริ๊นซ์ ซึ่งใช้ขั้วบอแมมป์ เบอร์ uA741 ค่ออินสั๊กณะของวงจรขยายสัญญาณแบบกลับเฟส (Inverting Amplifier) ซึ่งมีอัตราขยายแรงดัน (Voltage Gain) ประมาณ 22 เท่า เพื่อทำการขยายสัญญาณ หลังจากนั้น สัญญาณที่ได้ จะถูกนำมากรอง เพื่อกำจัดสัญญาณที่ไม่ต้องการออก โดยใช้วงจรกรองความถี่แบนด์พาสแบบแอคทีฟ (Active Band Pass Filter) ซึ่งมีความถี่คัทออฟที่ 50 Hz (ซึ่งเป็นความถี่ของสัญญาณไฟฟ้าบ้าน) และ 3.3 KHz (ซึ่งเป็นความถี่สูงสุดที่ใช้ในการส่งสัญญาณเสียงทางคู่สายโทรศัพท์) ซึ่งวงจรกรองความถี่ดังกล่าว เป็นดังรูปที่ 4.11



รูปที่ 4.11 วงจรกรองความถี่สัญญาณทางภาคอินพุท

จากวงจรกรองความถี่สัญญาณดังรูป เราจะสามารถแยกการพิจารณาวงจรออกได้เป็น 2 ส่วน (วงจรสองส่วนนี้มีการต่อกันแบบคาสเคด (Cascade)) ดังนี้

1. วงจรส่วนไฮพาส เป็นวงจรกรองความถี่แบบพาสซีฟอันดับที่ 1 ประกอบด้วย ตัวเก็บประจุไฟฟ้า  $C_1$  และตัวต้านทาน  $R_1$  ซึ่งต่อขนานอยู่กับ ค่าความต้านทานทางค่านอินพุทของวงจรในส่วนที่ 2 ซึ่งมีค่ามากกว่า  $R_1$  มาก จึงไม่น่ามาคิด ดังนั้นจะได้ว่า



รูปที่ 4.12 วงจรกรองความถี่ไฮพาสที่ 1

สมการโนด (Node Equation)

$$V_o(sC_1 + 1/R_1) - V_{in}(sC_1) = 0 \quad \dots(4.4)$$

ทราบเฟอ์ฟังก์ชัน

$$\frac{V_o}{V_{in}} = \frac{sC_1}{sC_1 + (1/R_1)} = \frac{s}{s + (1/C_1R_1)} \quad \dots(4.5)$$

ซึ่งจะได้ความถี่คัตออฟที่

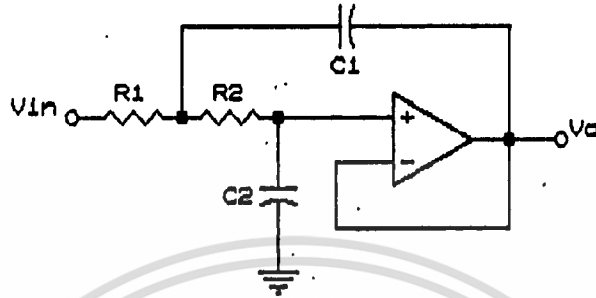
$$f_H = 1/2\pi(C_1R_1) \quad \dots(4.6)$$

แทนค่า

$$= 1/2\pi(3.3 \times 10^{-6})(165)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่ = 49.98 = 50 Hz อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. วงจรส่วนโวลท์พาส เป็นวงจรกรองความถี่สัญญาณแบบแอดทีฟอันดับที่ 2 ประกอบด้วยตัวออปแอมป์ และวงจร RC ดังรูปที่ 4.13



รูปที่ 4.13 วงจรกรองความถี่โวลท์พาสที่ใช้

พิจารณาวงจรโดยใช้การวิเคราะห์แบบโนด (Node Analysis) จะได้สมการดังนี้

$$\begin{bmatrix} 1 + \frac{1}{R_1} + \frac{1}{R_2} + sC_1} & -\frac{1}{R_2} \\ \frac{1}{R_2} & 1 + sC_2 \end{bmatrix} \begin{bmatrix} V_x \\ V_1 \end{bmatrix} = \begin{bmatrix} sC_1 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_3 \\ V_2 \end{bmatrix} \quad \dots(4.7)$$

จะได้ทรานเฟอร์ฟังก์ชันฟีดแบค (Feedback Transfer Function) ... (4.8)

$$T_{FB} = \left. \frac{V_1}{V_3} \right|_{V_2=0} = \frac{s/R_2C_2}{s^2 + s[(1/R_1C_1) + (1/R_2C_1) + (1/R_2C_2)] + (1/R_1R_2C_1C_2)}$$

ทรานสเฟอร์ฟังก์ชันฟีดฟอร์เวิร์ด (Feedforward Transfer Function) ... (4.9)

$$T_{FF} = \left. \frac{V_1}{V_2} \right|_{V_3=0} = \frac{1/R_1R_2C_1C_2}{s^2 + s[(1/R_1C_1) + (1/R_2C_1) + (1/R_2C_2)] + (1/R_1R_2C_1C_2)}$$

ซึ่งจะได้ทรานเฟอร์ฟังก์ชันของวงจรดังสมการ

$$T_V = \frac{kN_{FF}}{D - kN_{FB}} \quad \dots(4.10)$$

เมื่อ  $N_{FB}$ ,  $N_{FF}$  คือ เศษ (Numerator) ของ  $T_{FB}$ ,  $T_{FF}$  ตามลำดับ

$D$  คือ ส่วน (Denominator) ของทั้ง  $T_{FB}$  และ  $T_{FF}$

ดังนั้น จะได้ว่า

$$T_V = \frac{k/R_1R_2C_1C_2}{s^2+s\{(1/R_1C_1)+(1/R_2C_1)+((1-k)/R_2C_2)\}+(1/R_1R_2C_1C_2)} \dots(4.11)$$

จากฟังก์ชันของอัตราขยายแรงดันอันดับ 2 ของวงจรกรองความถี่แบบโลว์พาส

$$\frac{V_o}{V_{in}} = \frac{w_p^2}{s^2+(w_p/Q_p) s+w_p^2} \dots(2.6)$$

ซึ่งความถี่คัตออฟมีค่าเท่ากับความถี่โพล

$$w_L = w_p = \sqrt{1/R_1R_2C_1C_2} \dots(4.12)$$

$$f_L = 1/2\pi\sqrt{R_1R_2C_1C_2} \dots(4.13)$$

แทนค่า

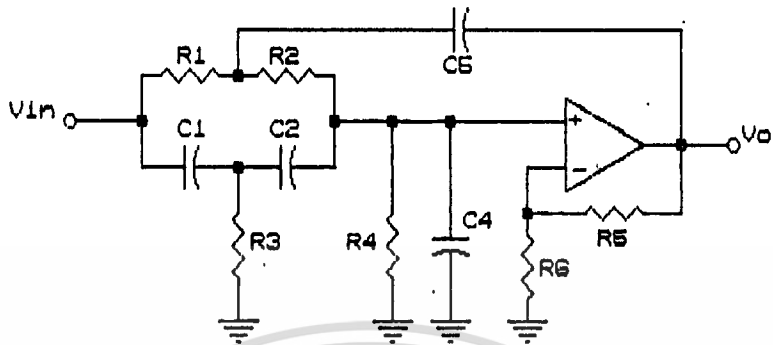
$$= 1/2\pi\sqrt{(72 \times 10^3)(72 \times 10^3)(837 \times 10^{-12})(536 \times 10^{-12})}$$

$$= 3.3 \text{ KHz}$$

หลังจากสัญญาณเสียงถูกกรองความถี่เรียบร้อยแล้ว ก็จะถูกมอดูเลตโดยไอซีเบอร์ MC3418 โดยผ่านเข้าทางขาที่ 1 (*Analog Input*) ซึ่งขาที่ 15 จะต้องถูกควบคุมให้มีสถานะเป็น '1' (*Encode*) จะทำให้ได้สัญญาณเอาต์พุต(Output) ในลักษณะของสัญญาณดิจิทัลแบบอนุกรม ออกทางขาที่ 9 (*Digital Output*)

ข) กรณีที่มีมอดูเลต ใช้สำหรับคอนแปลงสัญญาณดิจิทัลเป็นสัญญาณเสียง ซึ่งมีขั้นตอนการทำงานดังต่อไปนี้

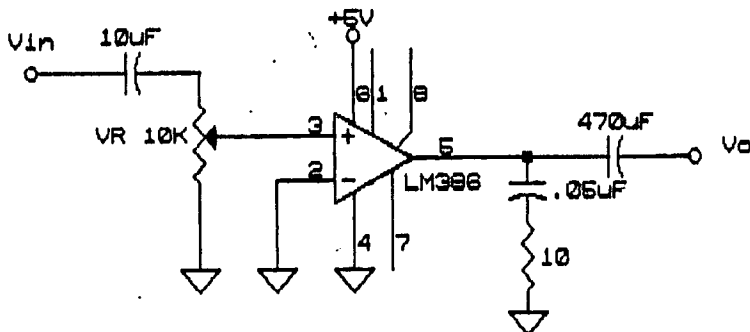
สัญญาณดิจิทัลที่ถูกส่งมาแบบอนุกรมจะถูกแปลงเป็นสัญญาณอนาลอก โดยเข้าทางขา 13 (*Digital Input*) ซึ่งเป็นอินพุตแบบกลับเฟส โดยขาที่ 15 จะต้องมิสถานะเป็น '0' (*Decode*) สัญญาณอนาลอกที่ได้จะถูกส่งออกทางขา 7 (*Analog Output*) ซึ่งสัญญาณที่ได้นี้ ตามคุณสมบัติของเคลคัมมอดูเลชัน จะทำให้มีสัญญาณความถี่สูงปะปนอยู่ ดังนั้นจึงจา



รูปที่ 4.14 วงจรกรองความถี่สำหรับสัญญาณจาก CVSD

วงจรข้างต้น จะทำการกรองสัญญาณที่ออกจากขา 7 ของ MC3418 ซึ่งมีสัญญาณเป็นแบบสามเหลี่ยม ให้ความเรียบขึ้น จากนั้นสัญญาณเสียงที่ได้จะถูกกรองสัญญาณอีกครั้ง ด้วยวงจรกรองความถี่สัญญาณแบนด์พาส เพื่อให้สัญญาณมีความถี่อยู่ในช่วง 50Hz ถึง 3.3 KHz ซึ่งวงจรและรายละเอียดจะเหมือนกับทางภาคอินพุต

ในขั้นสุดท้าย สัญญาณเสียงจะถูกนำมาขยายกำลัง โดยวงจรขยายกำลัง ซึ่งใช้ไอซี LM386 (Audio Power Amplifier) เพื่อให้สัญญาณเสียงที่ได้มีกำลังเพียงพอ สำหรับการส่งออกทางคู่สายโทรศัพท์ หรือซีบล่าโพง ซึ่งวงจรขยายกำลังที่ใช้มีอัตราขยายเท่ากับ 20 เท่า ดังแสดงในรูปที่ 4.15



รูปที่ 4.15 แสดงวงจรขยายกำลัง ด้วย LM386

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น การนำออกไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### วงจรส่วนดิจิทัล

หน้าที่ของวงจรในส่วนนี้ สามารถแบ่งออกได้ดังนี้

1. เป็นส่วนสร้างสัญญาณเพื่ออินเทอร์เฟซ Z80
2. เป็นพอร์ตอินพุท/เอาต์พุท (Port Input/Output) สำหรับการรับส่งข้อมูลสัญญาณเสียงกับ Z80
3. เป็นส่วนที่ทำหน้าที่แปลงสัญญาณระหว่างสัญญาณแบบขนานกับสัญญาณแบบอนุกรม

การทำงานของวงจร จะต้องอาศัยการทำงานของพอร์ตอินพุทเอาต์พุทที่ถูกตีโค้ดจาก Z80 จำนวน 4 พอร์ต ดังนี้

- พอร์ตเอาต์พุท สำหรับส่งข้อมูลเสียงไปยังวงจรถ่ายเสียงแบบอนาล็อกเลเตอร์ จำนวน 2 พอร์ต สำหรับการทํางาน 2 แชนแนล
- พอร์ตอินพุท สำหรับรับข้อมูลเสียงจากวงจรถ่ายเสียงแบบอนาล็อกเลเตอร์ จำนวน 1 พอร์ต
- พอร์ตควบคุม สำหรับควบคุมการทำงานของวงจรในส่วนแปลงสัญญาณทั้งหมด

ซึ่งวงจรในส่วนนี้ สามารถแบ่งออกได้เป็น 2 ส่วน ดังนี้

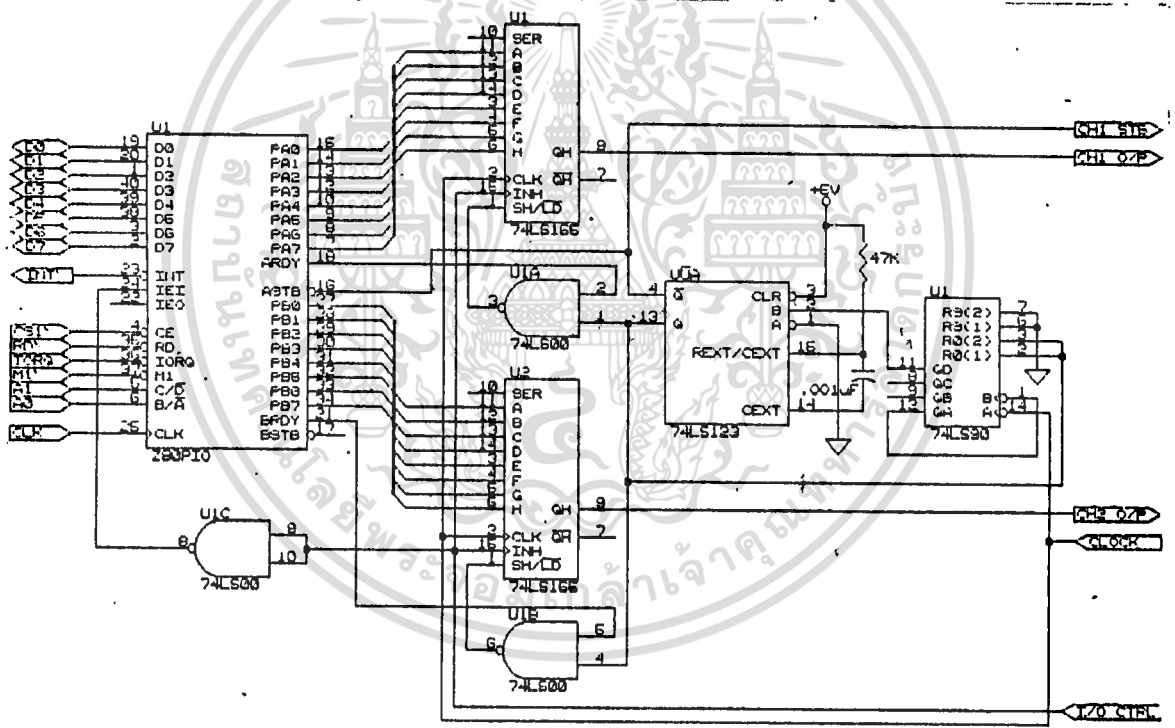
ก) วงจรถ่ายเสียงที่รับสัญญาณเสียงจาก Z80 แล้วแปลงสัญญาณจากขนานเป็นอนุกรม

วงจรถ่ายเสียงนี้จะใช้ Z80 PIO เป็นอุปกรณ์อินพุทเอาต์พุท เพื่อสนับสนุนการทำงานของอินเทอร์เฟซในโหมด 2 ของ Z80 เนื่องจาก Z80 PIO ถูกออกแบบมาเพื่อให้งานอินเทอร์เฟซโดยเฉพาะดังที่อธิบายไว้แล้ว

สำหรับวงจรถ่ายเสียงทั้งหมด สามารถแสดงได้ ดังรูปที่ 3.15 ซึ่งกำหนดให้ Z80 PIO ตัวที่ 1 ถูกตีโค้ด ได้พอร์ตแอดเดรสดังนี้

- พอร์ท 20h (Port A Data) - ใช้สำหรับการส่งข้อมูลในแชนแนลที่ 1
- พอร์ท 22h (Port A Control) - ใช้ควบคุมพอร์ท A
- พอร์ท 21h (Port B Data) - ใช้สำหรับการส่งข้อมูลในแชนแนลที่ 2
- พอร์ท 23h (Port B Control) - ใช้ควบคุมพอร์ท B

สำหรับการทำงานของ Z80 PIO จะให้ตอบสนองต่อสัญญาณสไตรบ เฉพาะพอร์ท A เท่านั้น ส่วนพอร์ท B จะถูกดิสเอเบิล (Disable) อินเทอร์รัพท์ โดยผ่านพอร์ทควบคุม B (23h) เนื่องจากวงจรในส่วนของพอร์ท B จะต้องอาศัยการตอบสนองต่อสัญญาณสไตรบของพอร์ท A เป็นหลัก ดังนั้นจึงขออธิบายเฉพาะวงจรในส่วนของพอร์ท A เท่านั้น



รูปที่ 4.16 วงจรรับสัญญาณจาก Z80 เพื่อนำมาแปลงเป็นสัญญาณแบบอนุกรม

วงจรข้างบนจะประกอบด้วยอุปกรณ์ต่างๆ ซึ่งแต่ละตัวมีหน้าที่ดังนี้

- Z80 PIO #1 เป็นอุปกรณ์สำหรับรับข้อมูลจาก Z80 และ สร้างสัญญาณอินเทอร์รัพท์ เพื่อตอบสนองต่อสัญญาณสไตรบ ซึ่งในที่นี้คือสัญญาณ ASTB\
- 74LS165 เป็นอุปกรณ์ในการแปลงสัญญาณแบบขนานเป็นแบบอนุกรม ซึ่งมีการทำ

เอกสารนี้เป็นเอกสารงานต้นฉบับนี้ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ก) เมื่อสัญญาณ SH/LD เป็น '0' 74LS165 จะทำการโหลดข้อมูลขนาด 8บิต เข้ามาเก็บไว้ในอินพุทรีจิสเตอร์ (Input Register) ภายใน
- ข) เมื่อสัญญาณ SH/LD เป็น '1' 74LS165 จะทำการชิฟท์ข้อมูลที่อยู่ในอินพุทรีจิสเตอร์ออกมาทีละบิต ตามจังหวะของสัญญาณนาฬิกา ซึ่งสัญญาณที่ขา INH จะต้องเป็น '0'

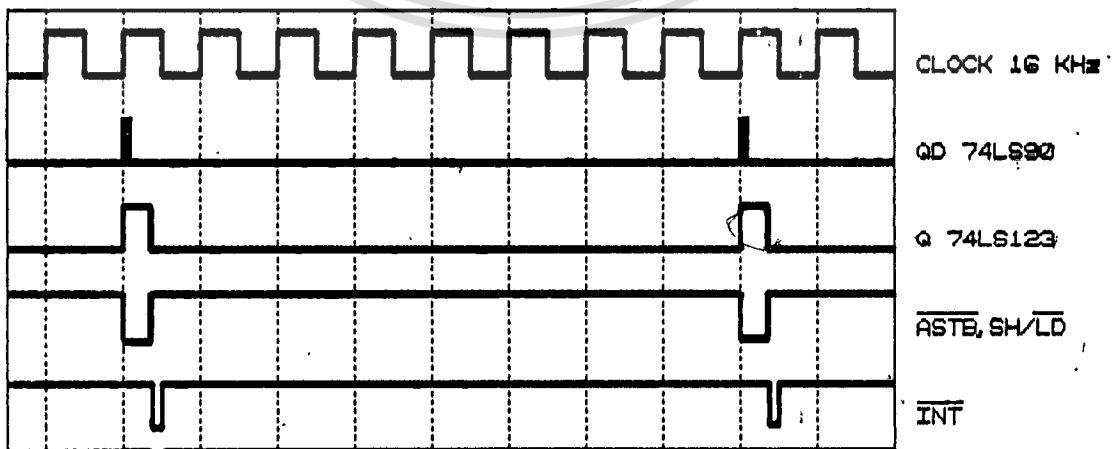
- 74LS123 เป็นไมโนสเทเบิล ซึ่งความกว้างของพัลส์ที่สร้างออกมา จะถูกควบคุมด้วย  $C_{EXT}$  และ  $R_{EXT}$

- 74LS90 เป็นวงจรมับสิบ (Decade Counter)

อุปกรณ์ 74LS123 และ 74LS90 เมื่อนำมาประกอบกันเป็นวงจรตั้งรูปแล้ว จะทำหน้าที่เป็นวงจรมับแปด ซึ่งสามารถอธิบายการทำงานของวงจรได้ดังนี้

74LS90 จะทำการนับตามจังหวะสัญญาณนาฬิกา เมื่อนับถึงแปด สัญญาณที่ขา  $Q_D$  จะเป็น '1' ซึ่งจะใช้ทริก 74LS123 ให้ทำการสร้างพัลส์ออกทางขา 13 ซึ่งจะถูกนำมาลบรีเซท (Reset) 74LS90 เพื่อให้ทำการนับใหม่ ซึ่งจะเห็นว่า สัญญาณพัลส์ที่เกิดขึ้น จะมีความถี่เป็น 1/8 เท่าของสัญญาณนาฬิกา

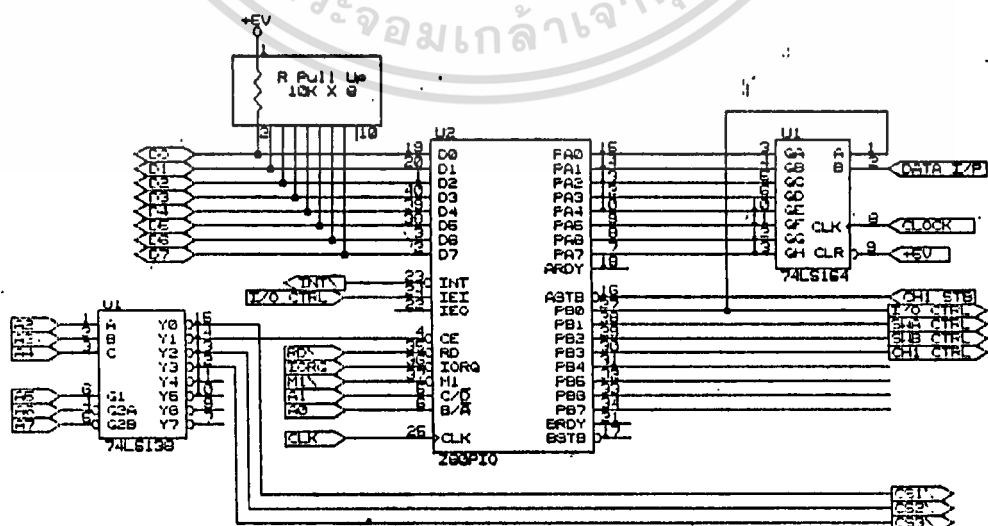
สำหรับการทำงานของวงจรทั้งหมดในส่วนนี้ สามารถอธิบายได้จากแผนผังเวลา ดังรูปที่ 4.11



เริ่มแรก วงจรนี้จะทำการนับแปด จากนั้นก็จะทำการสร้างสัญญาณสไตรบออกทางขา 4 ของ 74LS123 เพื่อให้พอร์ท A ทำการสร้างสัญญาณอินเทอร์รัพท์ (กำหนดให้มีการชีนาเบิล (Enable) อินเทอร์รัพท์ของพอร์ทควบคุมแล้ว) เมื่อ Z80 ได้รับสัญญาณอินเทอร์รัพท์แล้ว ก็จะกระโดดไปทำงานที่โปรแกรมบริการอินเทอร์รัพท์ ซึ่งโปรแกรมในส่วนนี้จะส่งให้ Z80 ทำการส่งข้อมูลออกทางพอร์ท A เมื่อข้อมูลถูกส่งออกมาแล้ว หลังจากสัญญาณบิตสุดท้ายของข้อมูลชุดที่แล้วถูกชิพท์ที่ขอบขาขึ้น (Leading Edge) ที่ 8 ของสัญญาณ Clock นั้น สัญญาณ SH/LD ของ 74LS165 จะเป็น '0' เพื่อให้ทำการโหลดข้อมูลจากพอร์ท A เข้าไปเก็บไว้ในอินพุทริจิสเตอร์ ซึ่งหลังจากสัญญาณขา 13 ของ 74LS123 ลง '0' แล้ว 74LS165 ก็จะอยู่ในสภาวะของการชิพท์ข้อมูล ซึ่งเมื่อสัญญาณ Clock ลูกต่อไปมา ก็จะทำให้เกิดการชิพท์ของข้อมูลที่ได้จากพอร์ท A ในลักษณะอนุกรม เมื่อครบจำนวน 8 บิต วงจรก็จะทำการสร้างสัญญาณสไตรบขึ้นอีก ซึ่งจะทำการห้วงจรทำงานเช่นที่กล่าวมาแล้วเรื่อยๆ จนกว่าโปรแกรมบริการอินเทอร์รัพท์จะทำการดิส เอ บิลอินเทอร์รัพท์ของพอร์ทควบคุม

ข) วงจรส่วนที่ใช้แปลงข้อมูลจากอนุกรมเป็นขนาน แล้วส่งให้ Z80 รับข้อมูล

วงจรในส่วนนี้จะใช้ Z80 PIO อีกตัวหนึ่ง เป็นอุปกรณ์อินพุทเอาท์พุท เพื่อสนับสนุนการทำงานของอินเทอร์รัพท์ในโหมด 2 เหมือนวงจรที่แล้ว ซึ่งวงจรในส่วนนี้เป็นดังรูปที่ 3.17



รูปที่ 4.18 วงจรแปลงข้อมูลเป็นแบบขนานแล้วส่งให้ Z80 รับข้อมูล

จากรูป Z80 PIO #2 จะถูกตีค่าโดย 74LS138 ซึ่งจะได้พอร์ทแอดเดรสดังนี้

พอร์ท 24h (Port A Data) - ใช้สำหรับการรับส่งข้อมูล

พอร์ท 26h (Port A Control) - ใช้ควบคุมพอร์ท A

พอร์ท 25h (Port B Data) - ใช้สำหรับควบคุมการทำงานของวงจร

พอร์ท 27h (Port B Control) - ใช้ควบคุมพอร์ท B

การทำงานของส่วนของพอร์ท A สามารถอธิบายได้จากแผนผังเวลารูปที่แล้ว ดังนี้

เริ่มแรก ในขณะที่วงจรมีขั้วจากวงจรวางตามสัญญาณนาฬิกา ข้อมูลแบบอนุกรมจากวงจรถอดค่ามอดดูเลเตอร์ จะถูกชิพที่ละบิตจนเป็นข้อมูลแบบขนานขนาด 8 บิต โดยไอซี 74LS164 เมื่อขา 1 มีสถานะเป็น '1' ซึ่งตอนนี้องจรนับจากวงจรวางจะทำการสร้างสัญญาณสไตรบสำหรับพอร์ท A (ASTB) ขึ้น ซึ่งจะทำให้พอร์ท A สร้างสัญญาณอินเทอร์รัพท์

เมื่อ Z80 ได้รับสัญญาณอินเทอร์รัพท์แล้ว ก็จะกระโดดไปทำงานที่โปรแกรมบริการอินเทอร์รัพท์ ซึ่งโปรแกรมในส่วนนี้จะสั่งให้ Z80 ทำการรับข้อมูลจากพอร์ท A จากนั้นวงจรถูกทำงานเช่นนี้เรื่อยไป จนกระทั่งโปรแกรมบริการอินเทอร์รัพท์จะทำการดีสเอเบิลอินเทอร์รัพท์ของพอร์ทควบคุม

สำหรับหน้าที่การทำงานในแต่ละบิตของพอร์ท B ซึ่งให้ใช้เป็นพอร์ทสำหรับควบคุมการทำงานของวงจร เป็นดังนี้

บิต 0 - ควบคุมการทำงานของ 74LS164, 74LS165 และ IEI ของ Z80 PIO ทั้งสองตัว

บิต 1,2 - ควบคุมอนาล็อกสวิตซ์

บิต 3 - ควบคุมการทำงานของ MC3418 ตัวที่ 1

อนึ่ง การตอบสนองต่อสัญญาณสไตรบ ของ Z80 PIO ทั้งสองตัว จะทำงานไม่พร้อมกัน ซึ่งจะสามารถทำได้โดยการควบคุมขา IEI ของ Z80 PIO ทั้งสองตัว ซึ่งมีสถานะแตกต่างกันเสมอ ด้วยสัญญาณบิต 0 ของพอร์ท B ของ Z80 PIO #2 (IEI (Input Enable

เอกสารที่ Interrupt)) ถ้าเป็น '1' จะตอบสนองต่อสัญญาณสไตรบ) อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.1.3 วงจรส่วนที่ใช้ในการติดต่อระหว่าง Z80 กับ IBM PC

การติดต่อระหว่าง Z80 กับ IBM PC ที่ใช้ในโครงการนี้ จะออกแบบให้ทำการติดต่อกันแบบขนาน เนื่องจากวงจรในลักษณะนี้มีความเร็วในการทำงานสูง การออกแบบค่อนข้างง่าย และระยะทางที่ใช้ในการติดต่อกันไม่ไกลกันนัก ซึ่งการติดต่อแบบขนานนี้ จำเป็นต้องมีพอร์ทอินพุทเอาต์พุทของแต่ละตัว คือ IBM PC และ Z80 โดยพอร์ทของแต่ละตัวจะถูกลำดับให้ชนกัน จำนวนพอร์ทจะเท่ากัน เนื่องจากสัญญาณจากบัสข้อมูลของ IBM PC และ Z80 ไม่สามารถติดต่อถึงกันได้โดยตรง จึงทำให้ต้องมีพอร์ทสำหรับการรับส่งข้อมูลของแต่ละตัว ดังที่กล่าวมาแล้ว โดยการทางานจะถูกควบคุมด้วยซอฟต์แวร์อีกทีหนึ่ง

สำหรับการออกแบบเพื่อให้การรับส่งข้อมูลระหว่าง Z80 กับ IBM PC ให้เป็นไปอย่างมีประสิทธิภาพ จะกำหนดให้มีบัสทั้งหมด 3 บัส ได้แก่

1. บัสข้อมูล (Data Bus) จะเป็นบัสขนาด 8 บิต ใช้เป็นเส้นทางในการรับส่งข้อมูลระหว่าง Z80 กับ IBM PC
2. บัสแสดงสถานะ (Status Bus) เป็นบัสขนาด 8 บิต จำนวน 2 บัส ซึ่งแต่ละบัสจะมีการส่งสัญญาณคนละทิศทางการใช้ เป็นสัญญาณควบคุมการทำงานของวงจร และควบคุมการรับส่งข้อมูลกันในลักษณะแชนด์เซค
  - 2.1 บัสบอกสถานะของ IBM PC (IBM PC Status Bus) เป็นบัสที่ใช้อบอกสถานะการทำงานของ IBM PC ให้แก่ Z80
  - 2.2 บัสบอกสถานะของ Z80 (Z80 Status Bus) เป็นบัสที่ใช้อบอกสถานะการทำงานของ Z80 ให้แก่ IBM PC

จากข้อกำหนดเบื้องต้น ทำให้สามารถแบ่งการออกแบบวงจรในส่วนนี้ได้เป็น 2 ส่วนคือ

1. วงจรอินเทอร์เฟซพอร์ทสำหรับ IBM PC (IBM PC Interface Port)

2. วงจรอินเทอร์เฟซพอร์ทสำหรับ Z80 (Z80 Interface Port) ใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## วงจรมินิเทอร์เฟสพอร์ทสำหรับ IBM PC

การที่จะทำให้ IBM PC สามารถมีบัสได้ทั้งหมด 3 ชุด ดังที่กล่าวมาแล้วนั้น จำเป็นจะต้องมีการออกแบบวงจรสำหรับการอินพุทและเอาต์พุทพอร์ท ทั้งหมด 4 พอร์ทดังนี้

1. พอร์ทเอาต์พุทข้อมูล (Data Output Port) ใช้เป็นพอร์ทสำหรับส่งข้อมูลให้ Z80
2. พอร์ทอินพุทข้อมูล (Data Input Port) ใช้เป็นพอร์ทสำหรับรับข้อมูลจาก Z80
3. พอร์ทแสดงสถานะ IBM PC (IBM PC Status Output Port) ใช้เป็นพอร์ทแสดงสถานะการทำงานของ IBM PC
4. พอร์ทรับสถานะ Z80 (Z80 Status Input Port) ใช้เป็นพอร์ทรับรู้สถานะการทำงานของ Z80

หลังจากการกำหนดข้างต้น จะต้องทำการเลือกช่วงของพอร์ทแอดเดรส ที่ว่างจากการใช้งานของ IBM PC ตามมาตรฐาน จากหนังสือ IBM PC Technical Reference เมื่อพิจารณาพอร์ทแอดเดรสแมพ (Port Address Map) สำหรับ IBM PC แล้ว จึงทำการเลือกพอร์ทแอดเดรสในช่วง 280h-284h ซึ่งเป็นช่วงที่ว่างจากการใช้ของ IBM PC

ดังนั้นจึงสามารถกำหนดพอร์ทแอดเดรสสำหรับพอร์ททั้ง 4 ได้ดังนี้

พอร์ท 280h - พอร์ทเอาต์พุทข้อมูล

พอร์ท 281h - พอร์ทอินพุทข้อมูล

พอร์ท 282h - พอร์ทแสดงสถานะ IBM PC

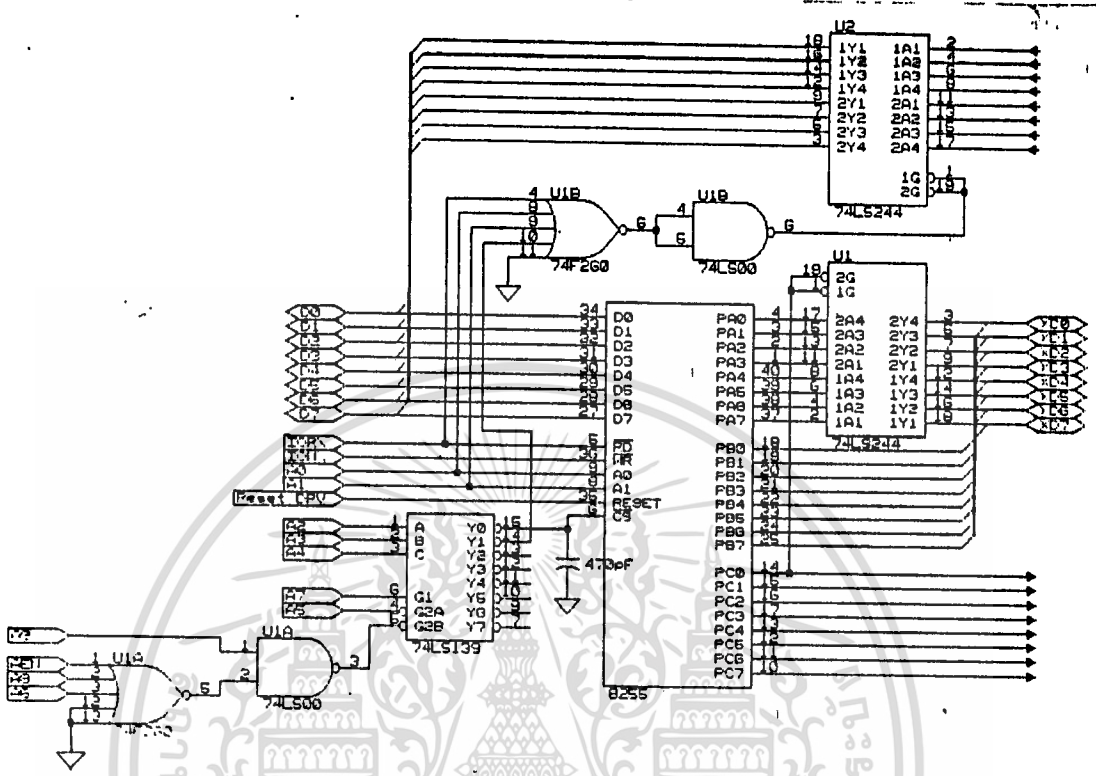
พอร์ท 284h - พอร์ทรับสถานะ Z80

พอร์ท 283h - พอร์ทควบคุม 8255 (8255 Control Port)

พอร์ท 280h-282h จะใช้ไอซี 8255 เป็นอุปกรณ์ I/O เนื่องจากการใช้งานง่าย สามารถโปรแกรมให้ใช้งานตามต้องการได้ และง่ายต่อการประกอบวงจร ส่วนพอร์ท 283h

เอกสารนี้จะใช้ 74LS244 ในการใช้งานเป็นพอร์ทอินพุทรวมค่านี้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งวงจรทั้งหมดในส่วนี้ เป็นดังรูปที่ 4.19



รูปที่ 4.19 วงจรอินเทอร์เฟซพอร์ทสำหรับ IBM PC

จากรูปบัสแอดเดรสของ CPU บนเครื่อง IBM PC จำนวน 10 เส้น จะถูกนำมาตีได้ต์ ร่วมกับสัญญาณ AEN เพื่อใช้ควบคุมการทำงานโดยกำหนดคีย์พอร์ท A (280h) และ C (282h) เป็นพอร์ทเอาต์พุต ส่วนพอร์ท B (281h) เป็นพอร์ทอินพุต ซึ่งในส่วนของโปรแกรม จะต้องทำการส่งค่า 82h ไปที่พอร์ทเบอร์ 283h ซึ่งเป็นพอร์ทควบคุมของ 8255

เพื่อให้บัสข้อมูล สามารถเป็นบัสที่ถูกต้องเชื่อมระหว่างพอร์ทอินพุตและพอร์ทเอาต์พุตหลายพอร์ทได้ จึงต้องมีการใช้ 74LS244 เพื่อเป็นบัฟเฟอร์ (Buffer) สำหรับข้อมูลเอาต์พุตจากพอร์ท A ซึ่งบัฟเฟอร์ตัวนี้ จะถูกควบคุมด้วย บิต 0 จากพอร์ท C

สำหรับหน้าที่และความหมายในแต่ละบิตของพอร์ท C ซึ่งใช้เป็นพอร์ทแสดงสถานะของ IBM PC เป็นดังนี้

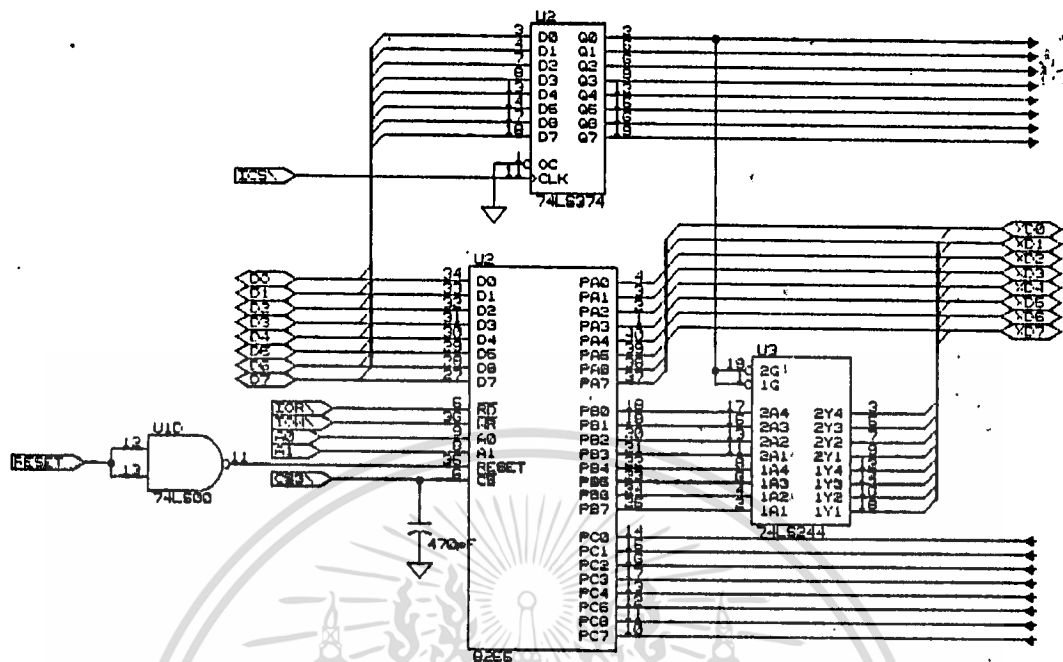
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บิต 0 - *IBM PC Output Ready* - ใช้เป็นสัญญาณที่บอก Z80 ว่า IBM PC ในขณะนี้ได้ทำการส่งข้อมูลออกทางบัสข้อมูลเรียบร้อยแล้ว (แอกทีฟ 0)
- บิต 1 - *IBM PC Input Ready* - ใช้เป็นสัญญาณที่บอก Z80 ว่า IBM PC ในขณะนี้ได้ทำการรับข้อมูลจากบัสข้อมูลเรียบร้อยแล้ว
- บิต 2,3 - *Part of Program* - ใช้เป็นข้อมูลที่บอกให้ Z80 กระโดดไปทำงานที่ส่วนใดของโปรแกรม
- บิต 4,5 - *Loop Ending Command* - ใช้เป็นสัญญาณเพื่อให้ Z80 เลิกการทำงานในลักษณะใดๆ ที่มีการวนลูป
- บิต 6 - *Quit* - ใช้เป็นสัญญาณที่บอกให้ Z80 เลิกการทำงาน
- บิต 7 - *Work Again* - ใช้บอกให้ Z80 ทำงานเหมือนเดิมอีกครั้งหนึ่ง

#### วงจรรีจิสเตอร์เฟสพอร์ทสำหรับ Z80

สำหรับการออกแบบวงจรรีจิสเตอร์ในส่วนนี้ จะเหมือนกับวงจรรีจิสเตอร์เฟสพอร์ทของ IBM PC ซึ่งพอร์ทแอดเดรสที่วางสำหรับ Z80 เมื่อใช้งานบนซิงเกิลบอร์ด (Single Board) ทั่วไป ได้แก่ พอร์ทแอดเดรสในช่วง 2Bh-2Fh โดยพอร์ทแอดเดรสแต่ละเบอร์ จะทำหน้าที่ดังนี้

- พอร์ท 2Bh - พอร์ทแสดงสถานะ Z80 (Z80 Status Output Port) - ใช้เป็นพอร์ทแสดงสถานะการทำงานของ Z80
- พอร์ท 2Ch - พอร์ทอินพุทข้อมูล (Data Input Port) - ใช้เป็นพอร์ทสำหรับรับข้อมูลจาก IBM PC
- พอร์ท 2Dh - พอร์ทเอาต์พุทข้อมูล (Data Output Port) - ใช้เป็นพอร์ทสำหรับส่งข้อมูลให้ IBM PC
- พอร์ท 2Eh - พอร์ทรับสถานะ IBM PC (IBM PC Status Input Port) - ใช้เป็นพอร์ทรับรู้สถานะการทำงานของ IBM PC
- พอร์ท 2Fh - พอร์ทควบคุม 8255 (8255 Control Port)



รูปที่ 4.20 วงจรอินเทอร์เฟซพอร์ทสำหรับ Z80

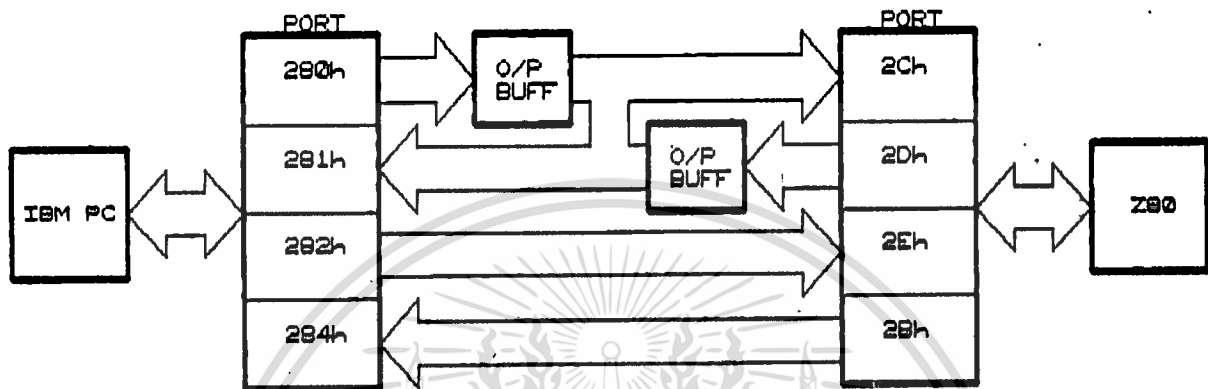
จะเห็นว่า ข้อมูลเอาต์พุตที่พอร์ท 2Dh จะต้องผ่านบัฟเฟอร์ เพื่อให้ข้อมูลภายในบัสข้อมูลไม่สับสน โดยการควบคุมโดยบิต 0 ของพอร์ท 2Bh ซึ่งความหมายของบิต 0 นี้ก็คือ Z80 Output Ready (แอกทีฟ '0') และบิต 1 คือ Z80 Input Ready

ส่วน 8255 จะต้องถูกควบคุมให้มิสสถานะการทำงานโดย พอร์ท A (2Ch) และ C (2Eh) เป็นพอร์ทอินพุต ส่วนพอร์ท B (2Dh) เป็นพอร์ทเอาต์พุต โดยต้องส่งค่า 99h ไปที่พอร์ทควบคุม 8255 (พอร์ท 2Fh)

รายละเอียดหน้าที่ในแต่ละบิตของพอร์ทแอสตสถานะ เป็นดังนี้

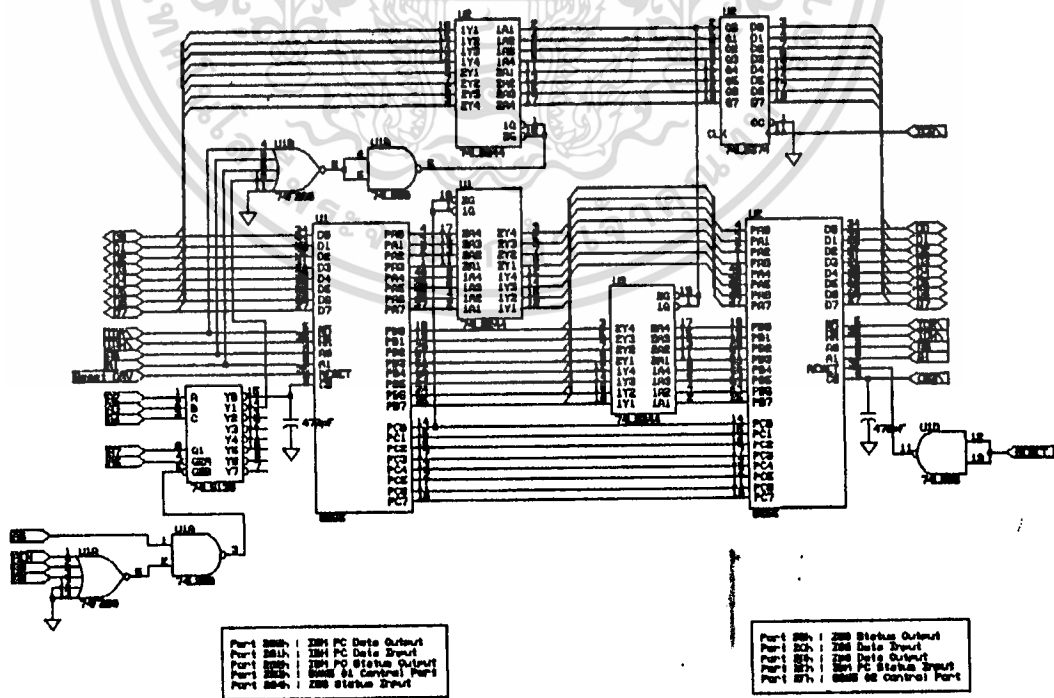
- บิต 0 - Z80 Output Ready - ใช้เป็นสัญญาณที่บอก IBM PC ว่า Z80 ในขณะนี้ได้ทำการส่งข้อมูลออกทางบัสข้อมูลเรียบร้อยแล้ว (แอกทีฟ 0)
- บิต 1 - Z80 Input Ready - ใช้เป็นสัญญาณที่บอก IBM PC ว่า Z80 ในขณะนี้ได้ทำการรับข้อมูลจากบัสข้อมูลเรียบร้อยแล้ว
- บิต 3-7 - ใช้เป็นสัญญาณที่ Z80 ใช้ตอบรับสัญญาณบิตเดียวกัน ที่ IBM PC ส่งมา

สำหรับการต่อกันระหว่างวงจรทั้งสองส่วน คือวงจรอินเทอร์เฟซพอร์ทของ IBM PC และวงจรอินเทอร์เฟซพอร์ทของ Z80 จะต้องติดตั้งแผนผังรูปที่ 4.21



รูปที่ 4.21 ลักษณะการต่อวงจรระหว่างวงจรอินเทอร์เฟซพอร์ทของ IBM PC และ Z80

ซึ่งจะได้วงจรสำเร็จงานลักษณะดังรูปที่ 4.22



รูปที่ 4.22 แสดงวงจรติดตั้งระหว่าง Z80 กับ IBM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

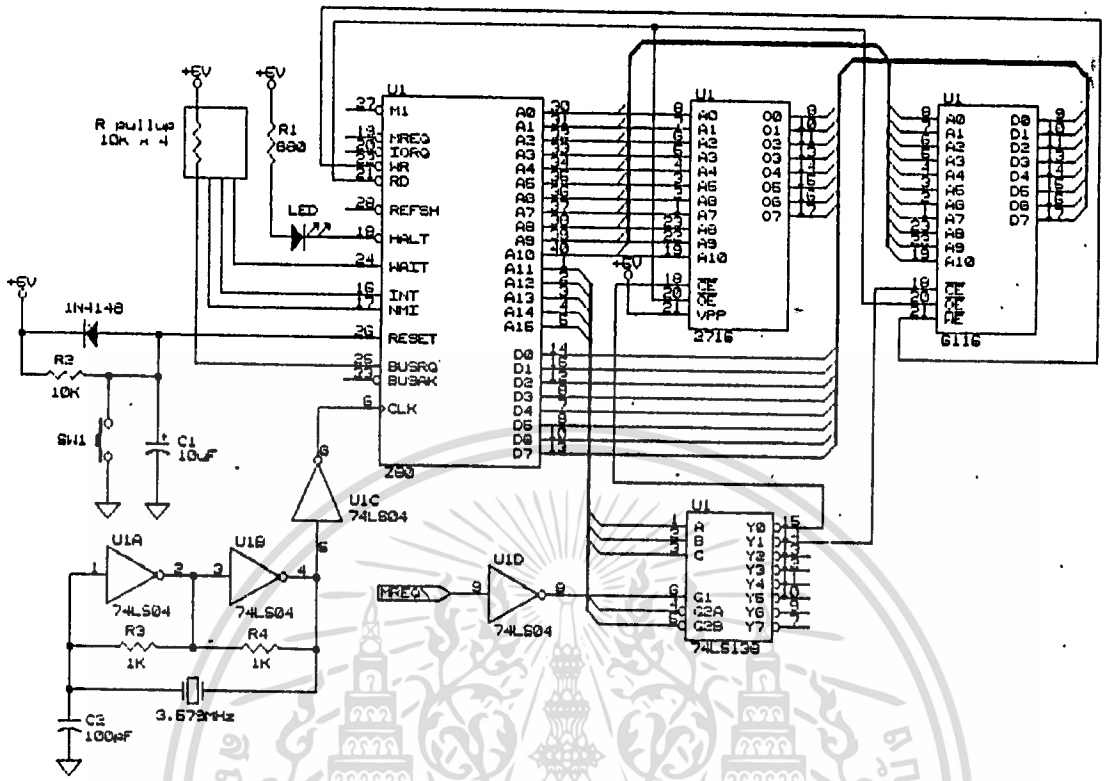
#### 4.1.4 วงจรส่วนควบคุมการทำงาน

เป็นส่วนที่ทำหน้าที่ควบคุมการทำงานของวงจรทั้งหมด โดยใช้ไมโครโปรเซสเซอร์ Z80 ซึ่งได้รับคำสั่งการทำงานจาก IBM PC สำหรับวงจรในส่วนนี้ จะประกอบด้วยส่วนสำคัญทั้งหมด 4 ส่วน คือ

1. ไมโครโปรเซสเซอร์ Z80 เป็นส่วนควบคุมหลัก
2. วงจรสร้างสัญญาณนาฬิกาที่มีความถี่ 3.57 MHz โดยการใช้คริสตัล 3.57 MHz เป็นฐานเวลา
3. วงจรดีโค้ทหน่วยความจำ (Memory Decoder) โดยใช้อิซี 74LS138
4. หน่วยความจำ (Memory) ซึ่งประกอบด้วย
  - อีพรอม (Erasable Programmable Read Only Memory (EPROM)) ซึ่งถูกดีโค้ทที่ตำแหน่งแอดเดรสที่ 0000h ถึง 07FFh ใช้เป็นหน่วยความจำสำหรับโปรแกรมหลักที่ทำหน้าที่ในการควบคุมการทำงานของวงจรทั้งหมด ในที่นี้ใช้อิซี อีพรอมเบอร์ 2716 ซึ่งมีหน่วยความจำ 2 กิโลไบต์
  - แรม (Random Access Memory (RAM)) ซึ่งถูกดีโค้ทที่ตำแหน่งแอดเดรสที่ 0800h ถึง 0FFFh ใช้เป็นบัฟเฟอร์สำหรับเก็บสถานะการทำงาน และ ใช้เป็นสแต็ก (Stack) ของระบบ ในที่นี้ใช้อิซี แรมเบอร์ 6116 ซึ่งมีหน่วยความจำ 2 กิโลไบต์

ซึ่งการเริ่มการดำเนินงานของวงจรทั้งหมดสามารถทำได้โดยการกดสวิทช์รีเซต โดย Z80 จะเริ่มทำงานตามโปรแกรมที่แอดเดรสตำแหน่งที่ 0000h

วงจรที่ใช้งานเป็นดังรูปที่ 4.23



รูปที่ 4.23 แสดงวงจรส่วนควบคุมการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

#### 4.2 ส่วนซอฟต์แวร์

สำหรับส่วนซอฟต์แวร์ในโครงการนี้ จะสามารถแยกย่อยได้เป็น 2 ส่วนเพื่อใช้สำหรับการทำงานของส่วนควบคุม 2 ส่วน ได้แก่

4.2.1 ซอฟต์แวร์บนเครื่อง IBM PC เป็นส่วนควบคุมหลัก และ ใช้ในการเก็บบันทึกค้นหาข้อมูล รวมทั้งใช้เป็นส่วนบันทึกสัญญาณเสียงลงบนฮาร์ดดิสค์

4.2.2 ซอฟต์แวร์บน Z80 เป็นส่วนควบคุมวงจรรายนอก ซึ่งจะได้รับคำสั่งจากซอฟต์แวร์บนเครื่อง IBM PC

ซึ่งซอฟต์แวร์ทั้งสองส่วนนี้ จะร่วมกันทำงานในลักษณะของการแฮนด์เชค ผ่านทางวงจรรีโมทเฟสทางด้านฮาร์ดแวร์ตั้งที่ได้กล่าวมาแล้ว

##### 4.2.1 ซอฟต์แวร์บนเครื่อง IBM PC

โปรแกรมที่เขียนขึ้นสำหรับการทำงานบนเครื่อง IBM PC นี้ ใช้ภาษาปาสคาล โดยใช้เทอร์โบปาสคาล เวอร์ชัน 5.0 เป็นคอมไพเลอร์ (Compiler) ซึ่งโปรแกรมนี้อาจทำงานในลักษณะของระบบเมนู

สำหรับการทำงานของโปรแกรมทั้งหมด สามารถแยกอธิบายได้เป็น 3 ส่วนใหญ่ ๆ ดังนี้

1. โปรแกรมส่วนฐานข้อมูล
2. โปรแกรมส่วนบันทึกและโหลดสัญญาณเสียง
3. โปรแกรมสำหรับการติดต่อกับโทรศัพท์

โปรแกรมในส่วนของการสร้างเมนู และยูทิลิตี้ต่างๆ จะไม่นำมากล่าวถึงในที่นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

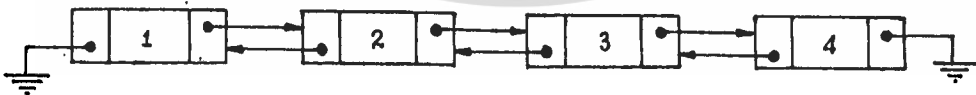
### โปรแกรมส่วนฐานข้อมูล

โปรแกรมส่วนนี้จะใช้ในการเก็บบันทึกและค้นหาข้อมูลต่างๆที่ต้องการ สำหรับลักษณะของการเก็บข้อมูลบนหน่วยความจำ และการค้นหาข้อมูล โปรแกรมจะทำงานในแบบของดับเบิลลิงก์ลิสต์ (Doubly Linklist) ซึ่งสามารถอธิบายหลักการได้ดังนี้

ลักษณะการเก็บข้อมูลสำหรับดับเบิลลิงก์ลิสต์นั้น ข้อมูลหนึ่งๆจะมีการชี้ (Point) โดยตัวชี้ (Pointer) 2 ตัว ภายในข้อมูลนั้น ตัวหนึ่งจะเป็นตัวอ้างอิงแอดเดรสสำหรับข้อมูลที่อยู่ก่อนหน้านี้นี้ และอีกตัวหนึ่งจะเป็นตัวอ้างอิงแอดเดรสสำหรับข้อมูลที่อยู่ถัดไป ดังนั้นเมื่อต้องการเก็บข้อมูล หนึ่ง จะต้องทำการจองหน่วยความจำทั้งหมด 3 ชุด คือ

1. หน่วยความจำสำหรับเก็บแอดเดรสของข้อมูลก่อนหน้า จำนวน 2 ไบต์
2. หน่วยความจำสำหรับเก็บข้อมูลตามจำนวนที่ต้องการ
3. หน่วยความจำสำหรับเก็บแอดเดรสของข้อมูลถัดไป จำนวน 2 ไบต์

สำหรับตัวชี้ข้อมูลก่อนหน้าของข้อมูลตัวแรกของลิสต์ และตัวชี้ข้อมูลถัดไปของข้อมูลตัวสุดท้าย จะชี้ไปที่ nil (สำหรับในภาษาปาสคาล) คือไม่มีชี้ไปที่ใดเลย ซึ่งแผนภาพสำหรับการเก็บข้อมูลแบบนี้ เป็นดังรูปที่ 4.24



รูปที่ 4.24 แผนภาพสำหรับการเก็บข้อมูลแบบดับเบิลลิงก์ลิสต์

ต่อไปนี้จะอธิบายการอธิบาย จะใช้ตัวชี้ตัวที่ 1 แทนตัวชี้สำหรับข้อมูลก่อนหน้า และตัวชี้ตัวที่ 2 สำหรับข้อมูลถัดไป

ซึ่งวิธีการต่างๆ ที่ใช้ในการจัดการข้อมูลแบบดับเบิลลิงก์ลิสต์ สามารถแยกการอธิบาย

ออกเป็นข้อๆ ได้ดังนี้

## 1. การเก็บข้อมูล

สำหรับข้อมูลตัวแรกของลิสต์นั้น การเก็บข้อมูลสามารถทำได้โดยการจองเนื้อที่หน่วยความจำสำหรับข้อมูลตามที่ต้องการ จากนั้นจะต้องทำการกำหนดให้ตัวชี้ทั้งสองตัวชี้ไปที่ nil เนื่องจากข้อมูลตัวแรก ไม่มีข้อมูลตัวก่อนหน้าและข้อมูลตัวถัดไปให้ชี้

ข้อมูลตัวถัดๆไป หลังจากทำการจองเนื้อที่หน่วยความจำแล้ว จะต้องทราบว่า จะทำการเก็บข้อมูลที่ตำแหน่งใดของลิสต์ โดยการตรวจสอบกับข้อมูลทั้งหมดภายในลิสต์ เริ่มจากข้อมูล ต้นลิสต์ (head of List) เรื่อยไปตามการชี้ของตัวชี้ตัวที่ 2 จนกว่าจะพบตำแหน่งที่ต้องการ ซึ่งการเก็บข้อมูลสามารถทำได้ 3 ตำแหน่ง คือ

- **ต้นลิสต์** ทำได้โดยการ กำหนดให้ตัวชี้ตัวที่ 1 ของข้อมูลใหม่ ชี้ไปที่ nil และตัวชี้ตัวที่ 2 ชี้ไปที่แอดเดรสของต้นลิสต์ตัวเดิม แล้วกำหนดให้ ตัวชี้ตัวที่ 1 ของต้นลิสต์ตัวเดิม ชี้ไปที่แอดเดรสของข้อมูลตัวใหม่ ซึ่งจะทำการให้ ข้อมูลตัวใหม่กลายเป็นต้นลิสต์ใหม่แทน

- **ท้ายลิสต์ (End of List)** ทำได้โดยการ กำหนดให้ตัวชี้ตัวที่ 2 ของข้อมูลใหม่ ชี้ไปที่ nil และตัวชี้ตัวที่ 1 ชี้ไปที่ แอดเดรสของท้ายลิสต์เดิม แล้วกำหนดให้ ตัวชี้ตัวที่ 2 ของท้ายลิสต์เดิม ชี้ไปที่แอดเดรสของข้อมูลตัวใหม่ ซึ่งจะทำการให้ ข้อมูลตัวใหม่กลายเป็นท้ายลิสต์ใหม่แทน

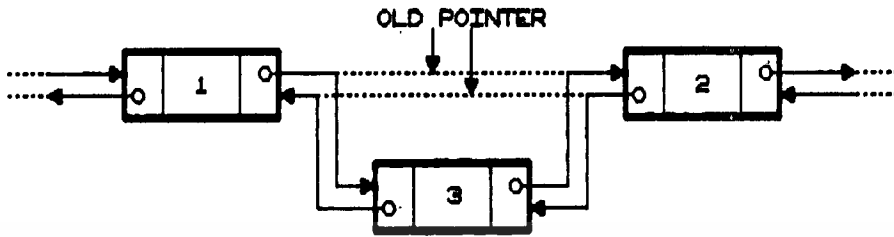
- **ภายในลิสต์** ซึ่งก็คือการแทรกข้อมูลนั่นเอง วิธีการก็คือ จะต้องทำการกำหนดตำแหน่งของตัวชี้ ภายในข้อมูล 3 ตัวดังนี้

**ข้อมูลก่อนหน้า** : กำหนดให้ตัวชี้ตัวที่ 2 ชี้ไปที่แอดเดรสของข้อมูลตัวใหม่

**ข้อมูลถัดไป** : กำหนดให้ตัวชี้ตัวที่ 1 ชี้ไปที่แอดเดรสของข้อมูลตัวใหม่

**ข้อมูลใหม่** : กำหนดให้ตัวชี้ตัวที่ 1 ชี้ไปที่แอดเดรสของข้อมูลก่อนหน้า และตัวชี้ตัวที่ 1 ชี้ไปที่ข้อมูลถัดไป

ซึ่งลักษณะของการแทรกข้อมูล สามารถแสดงได้ดังแผนภาพรูปที่ 4.25



รูปที่ 4.25 แผนภาพแสดงการแทรกข้อมูล

### 2. การค้นหาข้อมูล

การค้นหาข้อมูลสำหรับข้อมูลแบบลิงค์ลิสต์นี้ ทำได้เพียงวิธีเดียว คือ การค้นหาแบบซีควเอนเชียล (Sequential Searching) ซึ่งทำได้ 2 วิธี คือ

- เริ่มจากต้นลิสต์ คือ ทำการค้นหาจากต้นลิสต์ เรื่อยไปตามการชี้ของตัวชี้ตัวที่ 2 สำหรับข้อมูลถัดไป จนกระทั่งพบข้อมูลที่ต้องการ หรือค้นหาไปจนถึงท้ายลิสต์แล้ว
- เริ่มจากท้ายลิสต์ คือ ทำการค้นหาจากท้ายลิสต์ เรื่อยไปตามการชี้ของตัวชี้ตัวที่ 1 สำหรับข้อมูลก่อนหน้า จนกระทั่งพบข้อมูลที่ต้องการ หรือค้นหาไปจนถึงต้นลิสต์แล้ว

### 3. การลบข้อมูล

เป็นวิธีที่ตรงกันข้ามกับการเก็บข้อมูลที่อธิบายไว้แล้ว ซึ่งการลบข้อมูลออกจากลิสต์สามารถอธิบายได้ ตามตำแหน่งของข้อมูลที่ต้องการลบออก ดังนี้

- ต้นลิสต์ ทำได้โดยการ กำหนดค่าให้ตัวชี้ตัวที่ 1 ของข้อมูลถัดไป ชี้ไปที่ nil แล้วทำการยกเลิกการจองหน่วยความจำสำหรับข้อมูลที่ต้องการลบ ซึ่งในภาษาปาสคาล จะใช้คำสั่ง Dispose

- **ท้ายลิสต์** ทำได้โดยการ กำหนดให้ตัวชี้ตัวที่ 2 ของข้อมูลก่อนหน้า ชี้ไปที่ nil แล้วทำการยกเลิกการจองหน่วยความจำสำหรับข้อมูลที่ต้องการลบ

- **ภายในลิสต์** ทำได้โดยการ กำหนดให้ตัวชี้ตัวที่ 2 ของข้อมูลก่อนหน้า ชี้ไปที่แอดเดรสของข้อมูลถัดไป และ ตัวชี้ตัวที่ 1 ของข้อมูลถัดไป ชี้ไปที่แอดเดรสของข้อมูลก่อนหน้า แล้วทำการยกเลิกการจองหน่วยความจำสำหรับข้อมูลที่ต้องการลบ

สำหรับในโครงการนี้ โปรแกรมส่วนฐานข้อมูลจะใช้การเก็บข้อมูลแบบคีย์-ค่าจำนวน 2 ชุด ได้แก่

1. ชุดข้อมูลสำหรับเก็บข้อมูลหลัก เป็นส่วนสำหรับเก็บข้อมูลหลัก ซึ่งแบ่งข้อมูลออกเป็น 2 ส่วน คือ

- หมายเลข (Number) เป็นส่วนของข้อมูลที่เป็นหมายเลข จำนวนสูงสุด 10 หลัก
- ข้อมูล (Data) เป็นส่วนของข้อมูลหลักที่ต้องการเก็บ จำนวนสูงสุด 10 ตัวอักษร

2. ชุดข้อมูลสำหรับเก็บข้อมูลชื่อไฟล์ เป็นส่วนสำหรับเก็บชื่อไฟล์ตามข้อมูลหลัก ซึ่งแบ่งข้อมูลออกเป็น 2 ส่วน คือ

- ข้อมูล เป็นข้อมูลชุดเดียวกับข้อมูลในชุดข้อมูลแรก
- ชื่อไฟล์ เป็นส่วนของชื่อไฟล์ สำหรับข้อมูลนั้นๆ

ซึ่งข้อมูลทั้งหมด จะเก็บในลักษณะของข้อมูลแบบ สตริง (String)

ความสัมพันธ์ของข้อมูลทั้งสองชุดนี้ก็คือ ข้อมูลชุดแรกจะใช้สำหรับการค้นหาข้อมูลตามหมายเลขที่ต้องการ จากนั้น ก็จะใช้ข้อมูลชุดที่ 2 ในการค้นหาชื่อไฟล์สำหรับข้อมูลที่หาได้ ซึ่งชื่อไฟล์ในที่นี้ จะเป็นไฟล์ที่ใช้เก็บสัญญาณเสียงซึ่งสัมพันธ์กับข้อมูลที่หาได้ ซึ่งหมายความว่า จะต้องทำการโหลดสัญญาณเสียงไฟล์ใดสำหรับข้อมูลตามหมายเลขที่ต้องการ

## โปรแกรมส่วนบันทึกและโหลดสัญญาณเสียง

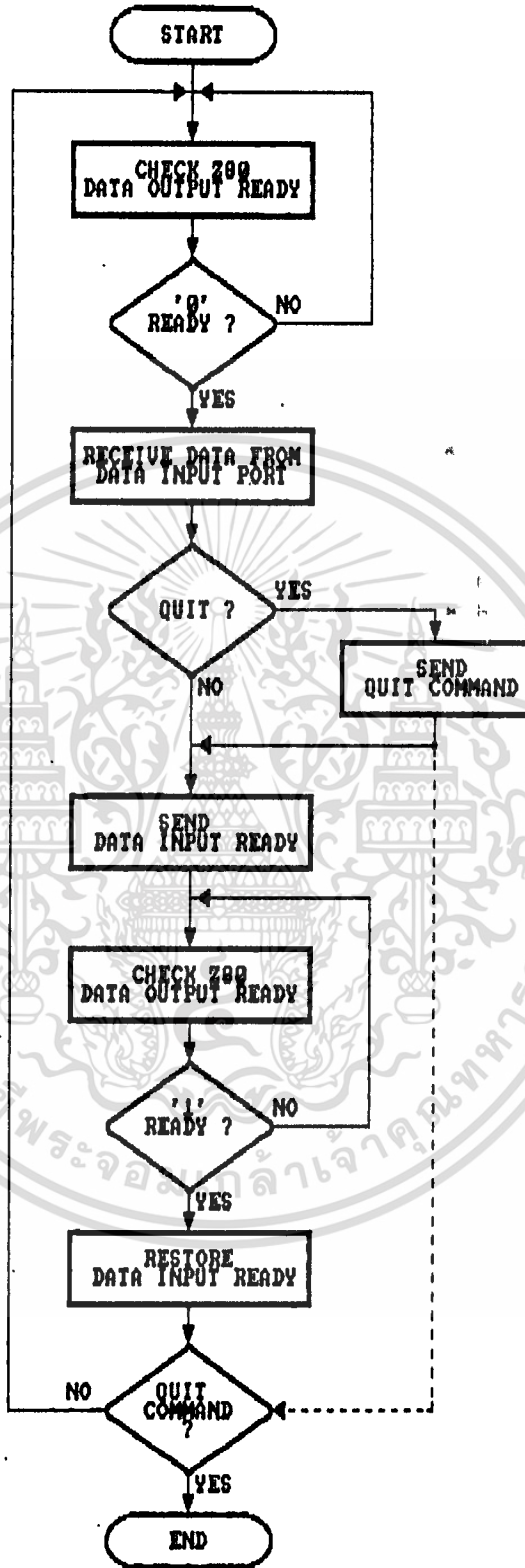
โปรแกรมส่วนนี้ จะใช้เป็นเสมือนอุปกรณ์สำหรับการบันทึกสัญญาณเสียงลงบนอุปกรณ์ความจำ เช่น แผ่นดิสก์ หรือ ฮาร์ดดิสก์ เป็นต้น รวมทั้งสามารถทำการโหลดสัญญาณเสียงที่บันทึกไว้แล้วออกมาฟังได้

สำหรับการจองหน่วยความจำบนเครื่อง IBM PC ในครั้งหนึ่งๆ นั้น จะสามารถจองได้อย่างมากที่สุด 1 เซกเมนต์ (Segment) เท่ากับ 64 กิโลไบต์ (Kilobytes) ซึ่งจากการทำงานในส่วนของฮาร์ดแวร์จะใช้สัญญาณนาฬิกาความถี่ 16 KHz ในการสุ่มสัญญาณสำหรับการแปลงสัญญาณแอนะล็อกด้วยโมดูลเลเตอร์ ซึ่งจะทำได้สัญญาณดิจิทัลจำนวน 16 กิโลบิต (Kilobit) หรือ 2 กิโลไบต์ ใน 1 วินาที ดังนั้นการบันทึกสัญญาณเสียงในแต่ละครั้งจึงสามารถบันทึกได้สูงสุด เท่ากับ 32 วินาที

การทำงานของโปรแกรมในส่วนนี้ จะใช้การรับส่งข้อมูลผ่านวงจรส่วนติดต่อระหว่าง IBM PC กับ Z80 ที่สร้างขึ้น โดยการทำงานจะต้องมีการแฮนด์เชคกันตลอดเวลา ซึ่งการอธิบายจะแยกออกเป็น 2 ส่วนคือ

### โปรแกรมบันทึกสัญญาณเสียง

อัลกอริธึม สำหรับการทำงานของโปรแกรมบันทึกเสียงนั้น คือ IBM PC จะทำการรับข้อมูลจาก Z80 (ซึ่ง Z80 รับมาจากวงจรแปลงสัญญาณ) ตามจังหวะการขออินเทอร์รัพท์จาก Z80 PIO ที่ถูกออกแบบไว้แล้ว เนื่องจากความถี่ของสัญญาณเสียงที่ถูกแปลงเป็นสัญญาณดิจิทัลนั้นมีค่าต่ำกว่าความถี่ของสัญญาณ Clock สำหรับการงานของ Z80 และ IBM PC หมายความว่า สัญญาณอินเทอร์รัพท์ที่เกิดขึ้นในแต่ละครั้ง หมายถึงข้อมูลเสียงพูดที่ถูกส่งมาในลักษณะอนุกรมได้ถูกจัดเรียงในลักษณะขนาน 8 บิต เรียบร้อย และ พร้อมแล้วสำหรับการทำงานของ Z80 ซึ่งการทำงานของโปรแกรมในส่วนนี้ สามารถอธิบายได้จากแผนผังการทำงานโปรแกรม (Flowchart) ดังรูปที่ 4.26



รูปที่ 4.26 แผนผังการทำงานโปรแกรมบันทึกสัญญาณเสียง สำหรับเครื่อง IBM PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

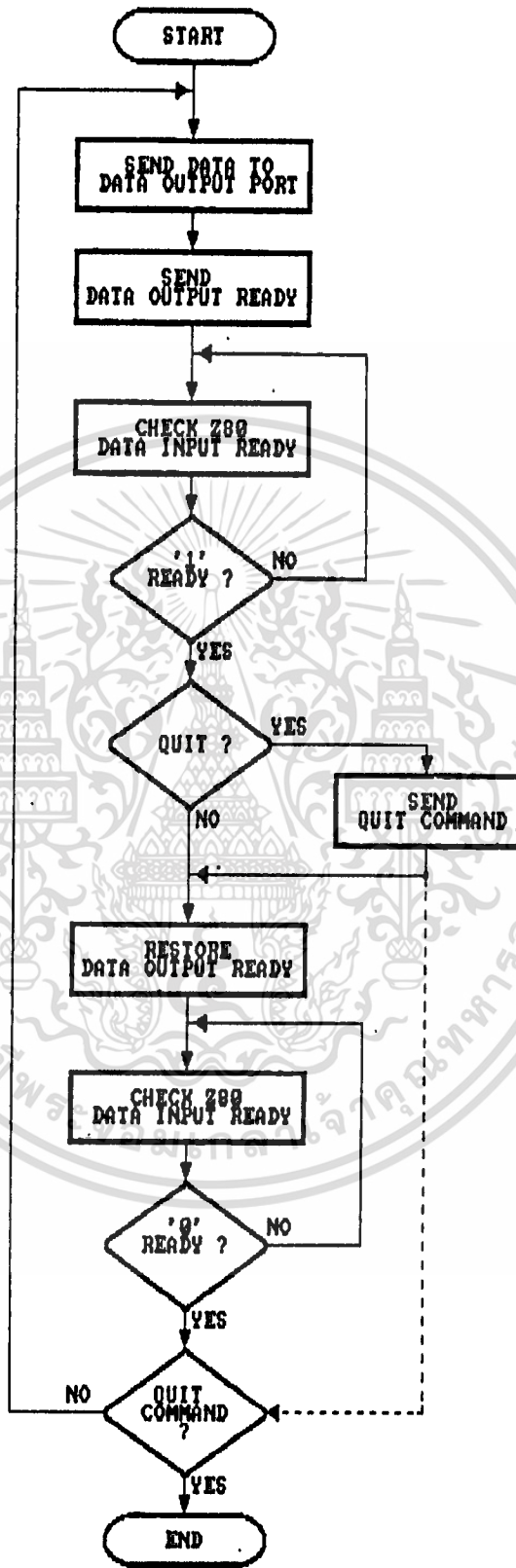
จากแผนผังการทำงานโปรแกรมข้างต้น สามารถอธิบายได้ดังนี้

1. โปรแกรมจะทำการตรวจสอบว่า ข้อมูลส่งมาที่บัสข้อมูลหรือยัง โดยตรวจที่บิต 0 ของพอร์ทรับสถานะ Z80 (*Data Output Ready*) ว่าแอกทีฟหรือไม่ ถ้าไม่ก็จะทำการวนรอสัญญาณนี้ ถ้าแอกทีฟ ก็จะทำงานในข้อต่อไป
2. รับข้อมูลจากพอร์ทอินพุทข้อมูลจำนวนหนึ่งไบต์
3. ตรวจสอบว่าขณะนี้หน่วยความจำที่จองไว้เต็มหรือยัง หรือ ผู้ใช้ต้องการหยุดการบันทึกสัญญาณเสียงหรือยัง ถ้าหน่วยความจำเต็ม หรือต้องการหยุด ก็จะทำการส่งสัญญาณหยุดการทำงานไปให้ Z80 เพื่อบอกให้ Z80 หยุดการส่งข้อมูล
4. ส่งสัญญาณบิตที่ 1 ของพอร์ทแสดงสถานะ IBM PC (*Data Input Ready*) เพื่อบอก Z80 ว่าได้ทำการรับข้อมูลเข้าไปเรียบร้อยแล้ว
5. รอการปรับค่าของสัญญาณเอาต์พุทเรียบร้อยจาก Z80
6. ทำการปรับค่าของสัญญาณอินพุทเรียบร้อยให้เป็น '0'
7. จากข้อ 3 ถ้าไม่มีการส่งสัญญาณหยุดการทำงาน โปรแกรมจะกระโดดกลับไปทำงานใหม่ตามข้อ 1 ถ้ามีก็จะหยุดการทำงานของโปรแกรมในส่วนนี้

หมายเหตุ การทำงานของโปรแกรมข้างต้น เป็นเพียงหลักการทำงานคร่าวๆ เท่านั้น ในโปรแกรมจริงจะต้องมีการตรวจสอบและรอการเซท หรือ รีเซท บิตข้อมูลอีกหลายบิตสำหรับการแฮนด์เชค ซึ่งข้อปลีกย่อยเหล่านี้ละเอียดเกินกว่าที่จะนำมากล่าวถึงในที่นี้

#### โปรแกรมไหลคสัญญาณเสียง

การทำงานของโปรแกรมในส่วนนี้ จะต้องเป็นไปตามจังหวะของสัญญาณอินเทอร์รัพท์ เช่นเดียวกับโปรแกรมในส่วนที่แล้ว โดย IBM PC จะเป็นฝ่ายรอเพื่อทำการส่งข้อมูลให้กับ Z80 ตามสัญญาณอินเทอร์รัพท์ ซึ่งหมายความว่า โปรแกรมในส่วนนี้ จะเป็นการสลับหน้าที่การทำงานกันของ Z80 กับ IBM PC ตามโปรแกรมในส่วนที่แล้ว ซึ่งขั้นตอนการทำงานของสามารถอธิบายได้ตามแผนผังการทำงานโปรแกรมดังรูปที่ 4.27



รูปที่ 4.27 แผนผังการทำงานโปรแกรมไหลตลอดตามเสียง สำหรับเครื่อง IBM PC เอกสารลิขสิทธิ์ที่แจ้งรุ่นและวิธีการใช้งานเพื่อการค้าเท่านั้น เมื่อผู้ซื้อได้เห็นใบรับประกันด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากแผนผังการทำงานโปรแกรมข้างต้น สามารถอธิบายได้ดังนี้

1. ส่งข้อมูลไปที่พอร์ทเอาต์พุตข้อมูลจำนวน 1 ไบท์ แล้วปรับบิต 0 ของพอร์ทแสดงสถานะ IBM PC เพื่อบอก Z80 ว่าส่งข้อมูลให้แล้ว
2. จะทำการวนลูปตรวจสอบบิต 1 ของพอร์ทแสดงสถานะ Z80 จนกระทั่งเป็น '1' เป็นการรับทราบว่า Z80 รับข้อมูลแล้ว
3. ตรวจสอบว่าส่งข้อมูลหมดหรือยัง หรือผู้ใช้ต้องการหยุดหรือไม่ ถ้าข้อมูลหมดแล้ว หรือต้องการหยุด ก็จะส่งสัญญาณหยุดการทำงานไปบอก Z80
4. ทำการปรับค่าของสัญญาณเอาต์พุตเรียบร้อย ให้เป็น '1'
5. รอการปรับค่าของสัญญาณอินพุตเรียบร้อยจาก Z80
6. จากข้อ 3 ถ้าไม่มีการส่งสัญญาณหยุดการทำงาน ก็จะกระโดดกลับไปทำงานตามข้อ 1 ใหม่ ถ้ามีก็จะหยุดการทำงานของโปรแกรมในส่วนนี้

การทำงานของโปรแกรมทั้งสองส่วนนี้ จะต้องทำงานพร้อมกับ Z80 เสมอ ซึ่งวิธีการส่งงาน Z80 ทำโดยการ ส่งสัญญาณบิตที่ 3,2 ของพอร์ทแสดงสถานะ IBM PC ซึ่ง Z80 จะทำการวนลูปรอรับสัญญาณ 2 บิตนี้อยู่ที่โปรแกรมหลัก หลังจากการกำหนดค่าเริ่มต้นของวงจรเรียบร้อยแล้ว ซึ่งความหมายของบิต 3,2 ที่ส่งให้ Z80 เป็นดังนี้

- 0,1 หมายความว่า ให้ Z80 กระโดดไปทำงานที่โปรแกรมย่อยสำหรับบันทึกสัญญาณเสียง
- 1,0 หมายความว่า ให้ Z80 กระโดดไปทำงานที่โปรแกรมย่อยสำหรับไหลลสัญญาณเสียง
- 1,1 หมายความว่า ให้ Z80 กระโดดไปทำงานที่โปรแกรมย่อยสำหรับการติดต่อกับโทรทัศน์ ซึ่งจะ เป็นส่วนองโปรแกรมที่จะอธิบายในหัวข้อต่อไป

อนึ่ง สัญญาณทั้งสองบิตนี้ เมื่อมีการเซทแล้ว หลังจากทีโปรแกรมทำงานเสร็จจะต้องมีการรีเซทสัญญาณทั้งสองนี้ ให้มีสถานะเหมือนตอนเริ่มต้น มิเช่นนั้นแล้ว การทำงานในครั้ง

### โปรแกรมสำหรับการติดต่อกับโทรศัพท์

โปรแกรมในส่วนนี้ เป็นจุดประสงค์หลักของโครงการนี้ การทำงานจะประกอบด้วย การทำงานของโปรแกรมในสองส่วนแรกที่อยู่มาแล้ว ลักษณะเด่นของโปรแกรมในส่วนนี้ คือ สามารถทำการเลือกชุดข้อมูลตามที่ต้องการ ตามรหัสประจำชุดข้อมูลที่เลือก โดยชุดข้อมูลประจำรหัสต่างๆ จำนวน 10 รหัส (จำนวนเท่ากับจำนวนปุ่มสำหรับเครื่องโทรศัพท์ โดยทั่วไป) จะถูกเก็บอยู่ในหน่วยความจำในลักษณะของข้อมูลแบบอาเรย์ (Array) ซึ่งสามารถเข้าถึง โดยพนักงานประจำเครื่อง

การทำงานของโปรแกรมในส่วนนี้ จะเริ่มต้นด้วยการส่งสัญญาณบีทที่ 3,2 ของพอร์ท แสดงสถานะ IBM PC เป็น 1,1 เพื่อให้ Z80 กระโดดไปทำงานที่โปรแกรมย่อยสำหรับการติดต่อกับโทรศัพท์ จากนั้น โปรแกรมก็จะทำงานตามแผนผังการทำงานของโปรแกรมดังรูปที่ 4.28

จากรูปที่ 4.28 จะสามารถอธิบายขั้นตอนการทำงานอย่างคร่าวๆ ของโปรแกรมในส่วนนี้ ได้ดังต่อไปนี้

1. วนลูปรอรับสัญญาณจาก Z80 ที่บอกว่ามีสัญญาณกริ่งโทรศัพท์มาแล้ว ซึ่งในขณะ นี้จะมีการตรวจสอบว่า ต้องการหยุดการทำงานโปรแกรมหรือไม่ ถ้าต้องการ โปรแกรมก็จะหยุดการทำงาน
2. เมื่อสัญญาณกริ่งมาแล้ว Z80 จะรับสายโทรศัพท์ แล้วส่งสัญญาณมาบอก IBM PC
3. ทำการส่งสัญญาณเพื่อบอกให้ Z80 ทำการส่งสัญญาณเสียงออกไป บอกให้ผู้ใช้ทำการกดรหัสประจำชุดข้อมูลที่ต้องการทราบ
4. รอรับหมายเลขจาก Z80 ซึ่งในขณะนี้จะมีการจับเวลาในการกดรหัสของผู้ใช้ จำนวน 60 วินาที ซึ่งถ้าเกิน 60 วินาทีแล้วผู้ใช้นี้ยังไม่กดรหัสเข้ามา จะถือว่า ผู้ใช้ได้เลิกการใช้งานแล้ว โปรแกรมจะทำการส่ง Z80 ให้ทำการวางสายโทรศัพท์ แล้วให้โปรแกรมกระโดดไปทำงานตามข้อ 1 ใหม่



5. ทำการรับข้อมูลหมายเลขมา แล้วทำการโหลดไฟล์ของชุดข้อมูล ตามหมายเลขรหัสที่ได้รับ
6. ส่งสัญญาณเพื่อบอกให้ Z80 ทำการส่งสัญญาณเสียงออกไป บอกให้ผู้ใช้ทราบข้อมูลที่ต้องการทราบเป็นจำนวนกี่บิต ตามที่ข้อมูลนั้นต้องการ
7. รอรับหมายเลขจาก Z80 ซึ่งในขณะนี้จะมีการจับเวลาในการกรหัสของผู้ใช้จำนวน 60 วินาที ซึ่งถ้าเกิน 60 วินาทีแล้วผู้ใช้ยังกรหัสเข้ามาไม่ครบ ก็จะถือว่า ผู้ใช้ได้เลิกการใช้งานแล้ว โปรแกรมจะทำการส่ง Z80 ให้ทำการวางสายโทรศัพท์ แล้วให้โปรแกรมกระโดดไปทำงานตามข้อ 1 ใหม่
8. ทำการรับข้อมูลหมายเลขมา แล้วทำการค้นหาข้อมูลตามหมายเลขที่ได้รับจากนั้นก็จะค้นหาว่าข้อมูลที่ได้ ตรงกับไฟล์สัญญาณเสียงไฟล์ใด แล้วทำการโหลดสัญญาณเสียงไฟล์นั้นไว้ที่หน่วยความจำ
9. ส่งสัญญาณเพื่อบอกให้ Z80 ทำการส่งสัญญาณเสียงตามที่โหลดไว้ออกไป จากนั้นก็จะส่งสัญญาณเสียงเพื่อบอกให้ผู้ใช้ทำการกดปุ่มใดๆ ถ้าต้องการใช้บริการต่อในเวลา 10 วินาที ถ้าเกิน 10 วินาทีแล้ว ยังไม่มีการกดของผู้ใช้ หมายความว่า ผู้ใช้ต้องการเลิกใช้บริการ โปรแกรมก็จะกระโดดไปทำงานตามข้อ 1 แต่ถ้าผู้ใช้ต้องการใช้บริการต่อ โปรแกรมก็จะกระโดดไปทำงานตามข้อ 3

เช่นกัน โปรแกรมในส่วนนี้ จะเป็นต้องทำงานในลักษณะแอสเซมบลีเชคกับ Z80 ซึ่งลักษณะขั้นตอนการทำงานของโปรแกรมสำหรับ Z80 จะกล่าวถึงในหัวข้อต่อไป

#### 4.2.2 ซอฟต์แวร์บน Z80

โปรแกรมที่เขียนขึ้นสำหรับการทำงานบน Z80 นี้ เขียนด้วยภาษาแอสเซมบลี โดยใช้โปรแกรม M80.COM เป็นแอสเซมเบลอร์ (Assembler) และใช้โปรแกรม L80.COM เป็นลิงค์เกอร์ (Linker) ซึ่งทั้ง 2 โปรแกรมนี้ทำงานบน Z80 & CP/M 2.2 Emulator v3.10 ซึ่งเป็นโปรแกรม ที่ใช้สำหรับจำลองการทำงานของ ระบบจัดการ ( Operating System (OS)) ที่เรียกว่า CP/M และการทำงานของไมโครโปรเซสเซอร์เบอร์ Z80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับออปโค้ด (Opcode) ที่ได้จากโปรแกรมในส่วนนี้ จะถูกเก็บอยู่ในหน่วยความจำส่วนที่เป็นอีพวอม บนวงจรส่วนควบคุมการทำงานดังที่กล่าวมาแล้ว

การแบ่งเนื้อที่หน่วยความจำของโปรแกรมในส่วนนี้จะ เป็นดังนี้

1. อีพวอม : มีการจัดเรียงแอดเดรสดังนี้

0000h - 03FFh เป็นส่วนของโปรแกรมหลัก

0400h - 04FFh เป็นส่วนของตารางอินเทอร์รัพท์เวคเตอร์

0500h - 057Fh เป็นส่วนของโปรแกรมบริการอินเทอร์รัพท์ สำหรับการบันทึก

สัญญาณเสียง

0580h - 05FFh เป็นส่วนของโปรแกรมบริการอินเทอร์รัพท์ สำหรับการไหลด

สัญญาณเสียง

2. แรม : มีการจัดเรียงแอดเดรสดังนี้

0800h เป็นส่วนที่ใช้เก็บค่าของข้อมูลแสดงสถานะ Z80

0801h เป็นส่วนที่ใช้เก็บค่าของข้อมูลในการควบคุมวงจรมายนอก

0802h เป็นส่วนที่ใช้เก็บค่าของรีจิสเตอร์ C ที่ใช้เป็นตัวนับ สำหรับการรับค่าหมายเลขจากปุ่มโทรศัพท์

0803h ,0804h เป็นส่วนที่ใช้เก็บค่าของคำสั่งจาก IBM PC ที่จะให้มีการทำงานต่อหรือไม่

0810h - 0819h เป็นส่วนที่ใช้เก็บข้อมูลหมายเลขที่รับได้จากโทรศัพท์ จำนวนมากที่สุด 10 ข้อมูล

0FFFh เป็นแอดเดรสแรกที่กำหนดให้ตัวชี้สแต็ค(Stack Pointer)

การทำงานของ Z80 นั้น จะเริ่มทำงานที่แอดเดรส 0000h หลังจากการให้รีเซท ซึ่งเริ่มแรกโปรแกรมจะทำการกำหนดค่าเริ่มต้นสำหรับอุปกรณ์ทางด้านฮาร์ดแวร์

1. กำหนดค่าไบท์สูงสำหรับแอดเดรสของตารางอินเทอร์รัพท์เวคเตอร์
2. กำหนดโหมดของการอินเทอร์รัพท์ เป็นโหมด 2
3. กำหนดให้ตัวชี้สแต็ค ชี้ไปที่แอดเดรสสุดท้ายของหน่วยความจำ ในที่นี้คือ OFFFH เนื่องจากค่าของตัวชี้สแต็คจะมีค่าลดลงเมื่อมีการใช้สแต็ค
4. กำหนดค่าเริ่มต้นของ 8255 ให้ทำงานดังนี้
  - พอร์ท A,C เป็นพอร์ทอินพุท
  - พอร์ท B เป็นพอร์ทเอาต์พุท
5. กำหนดค่าเริ่มต้นของพอร์ทแสดงสถานะ Z80 เป็น 01h
6. กำหนดค่าเริ่มต้นของ Z80 PIO ทั้งสองตัวให้ทำงานดังนี้
  - ตัวที่ 1 ทำงานในโหมด 0 ทั้งสองพอร์ท
  - ตัวที่ 2 พอร์ท A ทำงานในโหมด 1, พอร์ท B ทำงานในโหมด 0
7. กำหนดค่าไบท์ต่ำสำหรับแอดเดรสของตารางอินเทอร์รัพท์เวคเตอร์ โดยการส่งค่าไปที่พอร์ทควบคุมที่ต้องการให้ตอบสนองต่อสัญญาณสไตรบ
8. รีเซตเจเนอรัลพรีฟลอป และวงจรมีในวงจรมีติดต่อกับโทรศัพท์ที่อยู่ในสภาวะพร้อมที่จะทำงาน
9. กำหนดค่าเริ่มต้นให้รีจิสเตอร์ HL ซึ่งใช้เป็นตัวอ้างแอดเดรส สำหรับเก็บข้อมูลหมายเลขจากโทรศัพท์

หลังจากที่ Z80 ทำการกำหนดค่าเริ่มต้นเรียบร้อยแล้ว ก็จะทำการวนลูปรอรับคำสั่งจาก IBM PC โดยการตรวจสอบที่บิต 3,2 ดังที่ได้กล่าวมาแล้ว ,เมื่อได้รับคำสั่งแล้ว ก็จะกระโดดไปทำงานที่โปรแกรมย่อยต่างๆ ที่กำหนดไว้ ซึ่งโปรแกรมย่อยเหล่านี้มี 3 ส่วน คือ

1. โปรแกรมย่อยสำหรับการบันทึกสัญญาณเสียง
2. โปรแกรมย่อยสำหรับการไหลคสัญญาณเสียง
3. โปรแกรมย่อยสำหรับการติดต่อกับโทรศัพท์

สำหรับรายละเอียดการทำงานของแต่ละโปรแกรมย่อย สามารถอธิบายกันได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

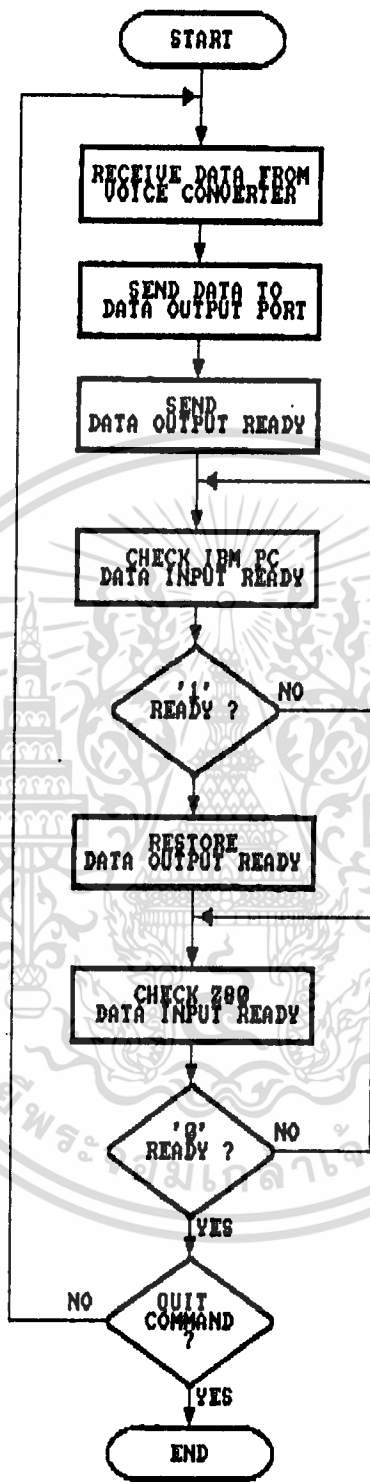
### โปรแกรมย่อยสำหรับการบันทึกสัญญาณเสียง

หลังจากที่ Z80 ได้รับคำสั่งจาก IBM PC ให้กระโดดมาทำงานที่โปรแกรมย่อยนี้ Z80 จะทำการกำหนดค่าเริ่มต้นสำหรับการทำงาน ได้แก่ ส่งสัญญาณเพื่อไปควบคุมวงจรร และ ส่งสัญญาณเพื่อทำการตัดลำโพงออกจากวงจร จากนั้นจะทำการรีนาเบิลอินเทอร์รัพท์ สำหรับพอร์ทควบคุม A ของ Z80 PIO ตัวที่ 2 และ ทำการรีนาเบิลอินเทอร์รัพท์ Z80 หลังจากนั้นโปรแกรมจะทำการวนลูป รอสัญญาณอินเทอร์รัพท์จาก Z80 PIO ซึ่งในขณะที่เดียวกันก็จะตรวจสอบคำสั่งจาก IBM PC สำหรับการเลิกการทำงาน ซึ่งการที่สัญญาณอินเทอร์รัพท์มาแต่ละครั้ง Z80 จะทำงานดังนี้

1. โปรแกรมจะทำการรับข้อมูลสัญญาณเสียง จากวงจรแปลงสัญญาณเสียง แล้วส่งข้อมูลที่ไต่ไปยังพอร์ทเอาต์พุตข้อมูล จากนั้นจะทำการส่งสัญญาณบิต 0 ของพอร์ท แสดงสถานะ IBM PC (*Data Output Ready*) เพื่อบอก IBM PC ให้ทราบว่าขณะนี้ Z80 ได้ทำการส่งข้อมูลออกมาที่บั๊สข้อมูลเรียบร้อยแล้ว
2. รอรับสัญญาณอินพุตเรียบร้อย '1' จาก Z80
3. ปรับค่าสัญญาณเอาต์พุตเรียบร้อยให้กลายเป็น '1'
4. รอการปรับค่าสัญญาณอินพุตเรียบร้อย '0' จาก Z80
5. ทำการตรวจสอบคำสั่งหยุดการทำงานซึ่งส่งมาพร้อมกับการปรับค่าอินพุตเรียบร้อย ถ้ามีคำสั่งหยุด โปรแกรมก็จะทำการดีสเอนเบิลอินเทอร์รัพท์พอร์ทควบคุมและอินเทอร์รัพท์ Z80 จากนั้นก็จะกลับเข้าสู่โปรแกรมหลักเพื่อรอคำสั่งจาก IBM PC
6. ถ้าไม่มีการหยุด Z80 ก็จะทำการรอสัญญาณอินเทอร์รัพท์ครั้งต่อไป ซึ่งจะทำให้ Z80 ทำงานตามข้อ 1 อีก

โปรแกรมข้างต้น จะถูกกำหนดคีย์อยู่ที่แอดเดรสตามที่ตารางอินเทอร์รัพท์เวคเตอร์ซึ่งแอดเดรสของตารางอินเทอร์รัพท์เวคเตอร์นั้นจะถูกอ้างโดยข้อมูล 2 ไบต์ดังนี้

1. ข้อมูลที่ถูกกำหนดภายในรีจิสเตอร์ I เป็นไบต์สูง
2. ข้อมูลที่ถูกกำหนดไปที่พอร์ทควบคุมของ Z80 PIO เป็นไบต์ต่ำ



เอกสารนี้เป็นรูปที่ 4.2.9 แผนผังการทำงานโปรแกรมบันทึกสัญญาณเสียง สำหรับ Z80 ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### โปรแกรมย่อยสำหรับการไหลคสัญญาณเสียง

หลังจากที่ Z80 ได้รับคำสั่งจาก IBM PC ให้กระโดดมาทำงานที่โปรแกรมย่อยนี้ Z80 จะทำการกำหนดค่าเริ่มต้นสำหรับการทำงาน ได้แก่ ส่งสัญญาณเพื่อไปควบคุมวงจร และส่งสัญญาณเพื่อทำการต่อลำโพงเข้ากับวงจร จากนั้นจะทำการอินทนาการเปิดอินเทอร์รัพท์สำหรับพอร์ตควบคุม A ของ Z80 PIO คู่ที่ 2 และ ทำการอินทนาการเปิดอินเทอร์รัพท์ Z80

หลังจากนั้นโปรแกรมจะทำการวนลูป รอสัญญาณอินเทอร์รัพท์จาก Z80 PIO ซึ่งในขณะเดียวกันก็จะตรวจสอบคำสั่งจาก IBM PC สำหรับการเลิกการทำงาน ซึ่งการที่สัญญาณอินเทอร์รัพท์มาแต่ละครั้ง Z80 จะทำงานดังนี้

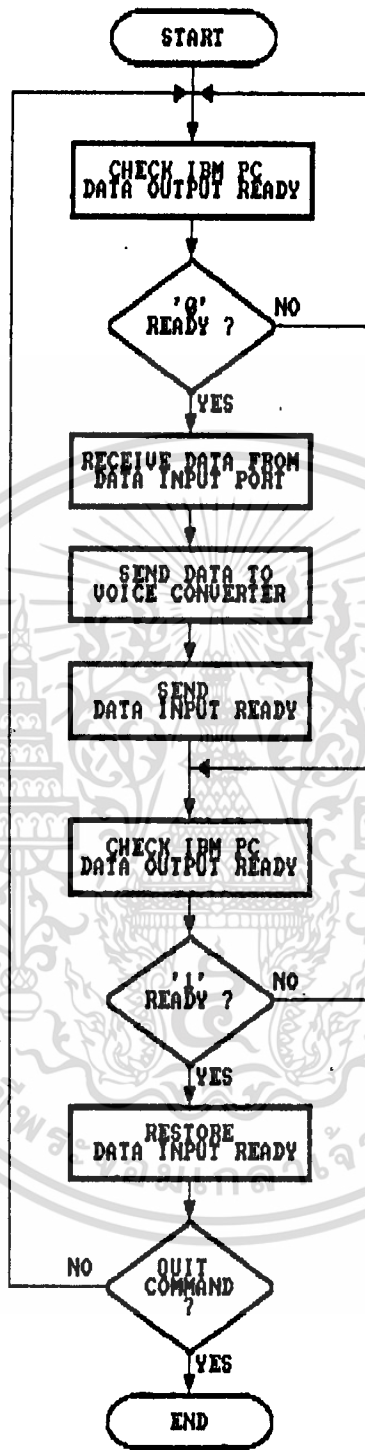
1. โปรแกรมจะทำการรอรับสัญญาณเอาต์พุทเรียบร้อย '0' จาก IBM PC
2. เมื่อได้รับสัญญาณเอาต์พุทเรียบร้อยแล้ว จะทำการรับข้อมูลทางพอร์ตอินพุทข้อมูล แล้วส่งข้อมูลที่ได้ออกไปสู่วงจรแปลงสัญญาณเสียง จากนั้นก็จะส่งสัญญาณอินพุทเรียบร้อยให้ IBM PC รับรู้
3. รอรับการปรับค่าสัญญาณเอาต์พุทเรียบร้อยเป็น '1' จาก IBM PC
4. ทำการปรับค่าสัญญาณอินพุทเรียบร้อย
5. ทำการตรวจสอบคำสั่งหยุดการทำงาน ซึ่งส่งมาพร้อมกับการปรับค่าสัญญาณอินพุทเรียบร้อย ถ้ามีคำสั่งหยุด โปรแกรมก็จะทำการดีสเอเบิลอินเทอร์รัพท์ของพอร์ตควบคุม และอินเทอร์รัพท์ Z80 จากนั้นก็จะกลับเข้าสู่โปรแกรมหลัก เพื่อรอรับคำสั่งจาก IBM PC ต่อไป
6. ถ้าไม่มีกัรหยุด Z80 ก็จะทำการรอสัญญาณอินเทอร์รัพท์ครั้งต่อไป ซึ่งจะทำให้ Z80 ทำงานตามข้อ 1 อีก

สำหรับแผนผังการทำงานโปรแกรมในส่วนนี้ เป็นดังรูปที่ 4.30

### โปรแกรมย่อยสำหรับการติดต่อกับโทรศัพท์

เริ่มแรก โปรแกรมจะทำการกำหนดค่าเริ่มต้นด้วยการส่งสัญญาณไปควบคุมวงจรภาย

เอกสารนี้อื่นจากนั้นโปรแกรมก็จะทำงานตามแผนผังการทำงานโปรแกรม ดังรูปที่ 4.31 ระเบียบนด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

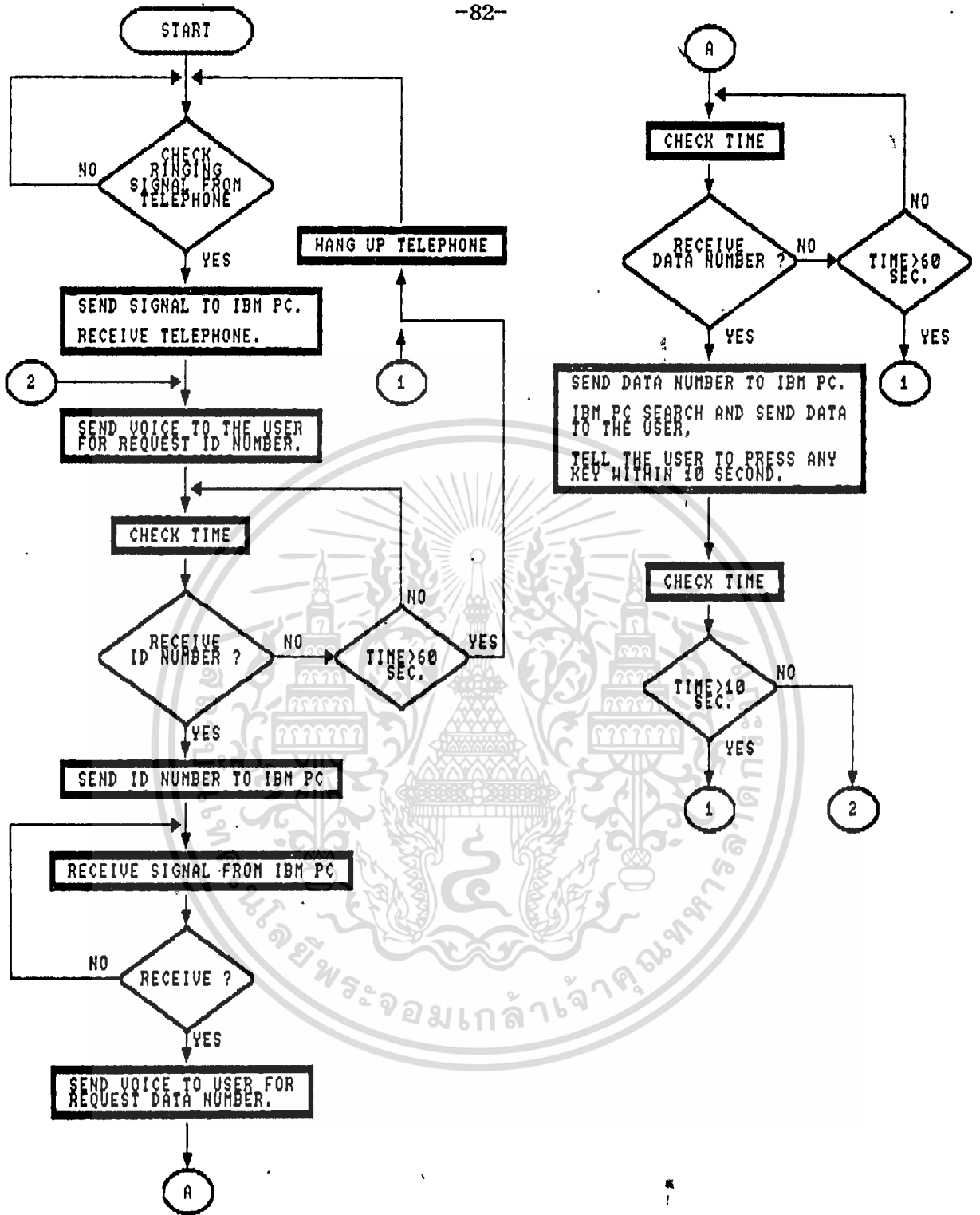


รูปที่ 4.30 แผนผังการทำงานของโปรแกรมไหลคัสสัญญาณเสียง สำหรับ Z80.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากแผนผังการทำงานโปรแกรมข้างต้น สามารถอธิบายได้ดังนี้

1. โปรแกรมจะทำการวนลูปรอรับสัญญาณกริ่งโทรศัพท์ ถ้าสัญญาณกริ่งมาก็จะส่ง  
มุลไปบอก IBM PC แล้วทำการรับสายโทรศัพท์
2. รอรับสัญญาณจาก IBM PC เพื่อทำการส่งสัญญาณเสียงไปบอกผู้ใช้ ให้ทำการ  
รหัสประจำชุดข้อมูลที่ต้องการทราบ
3. ทำการรอรับรหัสจากโทรศัพท์ ซึ่งในขณะที่เดียวกันนี้จะทำการตรวจสอบสัญญาณ  
จาก IBM PC ที่บอกว่าการจับเวลาเกิน 60 วินาทีแล้ว ถ้าตรวจพบก่อน  
แกมก็จะทำการวางสายโทรศัพท์ และกระโดดไปทำงานที่ข้อ 1
4. ถ้าได้รับรหัส ก็จะส่งรหัสนี้ไปให้ IBM PC จากนั้นก็จะรอรับสัญญาณเพื่อทำ  
ส่งสัญญาณเสียงไปบอกผู้ใช้ให้ทำการกรหัสข้อมูลที่ต้องการทราบจำนวนหลัก  
หมดไว้แล้ว ซึ่งในขณะที่เดียวกันนี้จะทำการตรวจสอบสัญญาณจาก IBM PC  
บอกว่าการจับเวลาเกิน 60 วินาทีแล้ว ถ้าตรวจพบก่อน โปรแกรมก็จะทำ  
วางสายโทรศัพท์ และกระโดดไปทำงานที่ข้อ 1
5. ถ้าได้รับรหัส ก็จะส่งรหัสนี้ไปให้ IBM PC เพื่อทำการค้นหาข้อมูล จากนั้นก็  
ทำการรอรับสัญญาณเพื่อทำการส่งสัญญาณเสียงสำหรับข้อมูลที่ต้องการทราบ รว  
บอกผู้ใช้ให้ทำการกดปุ่มใดๆ หลังจากฟังข้อมูลเสร็จแล้ว ภายใน 10 วินาที
6. ถ้าไม่มีการกดปุ่ม โปรแกรมก็จะทำการวางสายโทรศัพท์ และกระโดดไปทำ  
ที่ข้อ 1 ใหม่
7. ถ้ามีการกดปุ่ม โปรแกรมก็จะกระโดดไปทำงานที่ข้อ 2



รูปที่ 4.31 แผนผังการทำงานโปรแกรมติดต่อกับโทรศัพท์ สำหรับ Z80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 5

### การใช้งานโปรแกรม

#### เทเลอินฟอร์เมชัน เวอร์ชัน 1.0

ในบทนี้จะกล่าวถึง ขั้นตอนและวิธีการใช้งานโปรแกรม เทเลอินฟอร์เมชัน เวอร์ชัน 1.0 (Tele-Information Software Tool Version 1.0) ซึ่งเป็นซอฟต์แวร์ใช้งานบนเครื่อง IBM PC เพื่อสนับสนุนการทำงานของฮาร์ดแวร์ที่สร้างขึ้น โดยโปรแกรมในส่วนนี้จะทำงานร่วมกับโปรแกรมที่ทำงานบน Z80 ในลักษณะของการแชนด์เซค

สำหรับโปรแกรมนี้ ไฟล์ที่จะต้องนำไปใช้งาน ได้แก่

1. TINFO.EXE เป็นไฟล์โปรแกรมหลัก
2. TINFO.CON เป็นไฟล์สำหรับเก็บค่าคงที่ (Configuration) ของโปรแกรม TINFO.EXE
3. TINFO.COD เป็นไฟล์สำหรับเก็บรายการชุดข้อมูลที่ต้องการใช้ในการให้บริการ สอบถามข้อมูล

ซึ่ง 2 ไฟล์หลังนี้ จะต้องอยู่ในราก (Directory) ปัจจุบัน แต่ถ้าไม่มีก็ยังสามารถใช้งานได้ แต่ต้องทำการกำหนดค่าต่างๆ ที่ต้องการเองในภายหลัง

อนึ่ง โปรแกรมจะต้องถูกใช้งานบนเอ็มเอสดอส (MS DOS) ตั้งแต่เวอร์ชัน 2.0 ขึ้นไป

#### ขั้นตอนการใช้งาน

เริ่มแรกจะต้องทำการเปิดสวิทช์เครื่องภายนอกก่อน (ถ้าต้องการทำงานในส่วนที่ต้องติดต่อกับโทรศัพท์ จะต้องต่อสายเข้ากับเครื่องก่อน เพื่อการทำงานที่ไม่ผิดพลาด) แล้วทำการรัน (Run) โปรแกรม TINFO.EXE ซึ่งโปรแกรมนี้จะทำการกำหนดค่าเริ่มต้นสำหรับการคำนวณ (ถ้ามีการคำนวณ) ซึ่งค่าเริ่มต้นเหล่านี้จะขึ้นอยู่กับค่าเริ่มต้นของโปรแกรม ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงาน จากนั้นจะมีคำสั่งเพื่อให้ผู้ใช้ทำการกดสวิทช์วีเซทการทำงานบนเครื่องภายนอก ซึ่งจะทำการให้เครื่องภายนอกนี้ ทำการกำหนดค่าเริ่มต้นต่างๆ แล้วรอคำสั่งจาก IBM PC เพื่อการทำงานในลักษณะเช่นนี้ เชคต่อไป

หลังจากการกดวีเซทแล้ว ให้กดปุ่มใดๆ บนแป้นพิมพ์ (Keyboard) ซึ่งโปรแกรมจะทำงานต่อโดยการโหลดค่าคงที่ต่างๆ จากไฟล์ TINFO.CON และ TINFO.COD ตามลำดับ หลังจากนั้น โปรแกรมจะเข้าสู่การทำงานในระบบเมนู ซึ่งผู้ใช้สามารถใช้งานได้ง่าย โดยโปรแกรมทั้งหมด จะแบ่งการทำงานออกเป็น 5 เมนู ดังต่อไปนี้

- 5.1 File ใช้เป็นยูทิลิตี้ สำหรับจัดการเกี่ยวกับไฟล์
- 5.2 DataBase ใช้เป็นส่วนของการจัดการฐานข้อมูล
- 5.3 Voice ใช้สำหรับการบันทึก และโหลดสัญญาณเสียง
- 5.4 Execute ใช้สำหรับการให้บริการสอบถามข้อมูลผ่านทางโทรศัพท์
- 5.5 Options ใช้สำหรับการกำหนดค่าคงที่ของโปรแกรม

ซึ่งการใช้งานในแต่ละเมนูนั้น สามารถทำได้โดยการเลื่อนไฮไลท์ (Hilight) มาที่ชื่อเมนูที่ต้องการแล้วกดปุ่ม ENTER หรือ อาจจะได้โดยการกดปุ่ม Alt แล้วตามด้วยตัวอักษรตัวแรกของชื่อเมนูนั้น ในขณะที่อยู่ในเมนูหลัก สำหรับรายละเอียดการใช้งานในแต่ละเมนู เป็นดังนี้

### 5.1 File

เมนูในส่วนนี้เป็นการใช้งานเกี่ยวกับการจัดการไฟล์ ซึ่งอาศัยการทำงานของดอสเป็นหลัก โดยสามารถแบ่งการใช้งานได้ในลักษณะของเมนูย่อย 4 เมนู ได้แก่

5.1.1 Directory ใช้สำหรับเมื่อต้องการดูว่าในรากที่ต้องการ มีไฟล์ใดอยู่บ้าง ซึ่งลักษณะการใช้งาน จะเหมือนกับคำสั่ง DIR บนดอส

5.1.2 Change dir ใช้สำหรับการเปลี่ยนรากจากปัจจุบัน ไปสู่รากที่ต้องการ ด้านการคำ  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.3 OS shell ใช้สำหรับเมื่อ ผู้ใช้ต้องการใช้คำสั่งบนดอสเป็นการชั่วคราว โดยโปรแกรมจะทำการรันไฟล์ COMMAND.COM

ถ้าต้องการกลับเข้าสู่โปรแกรมหลัก ก็สามารถทำได้โดยการใช้คำสั่ง EXIT

5.1.4 Quit เป็นคำสั่งสำหรับการออกจากการทำงานของโปรแกรมนี ซึ่ง อาจทำได้โดยการกด Alt-X เมื่ออยู่บนเมนูหลัก

ลักษณะของเมนูในส่วนนี้ เป็นดังรูปที่ 5.1

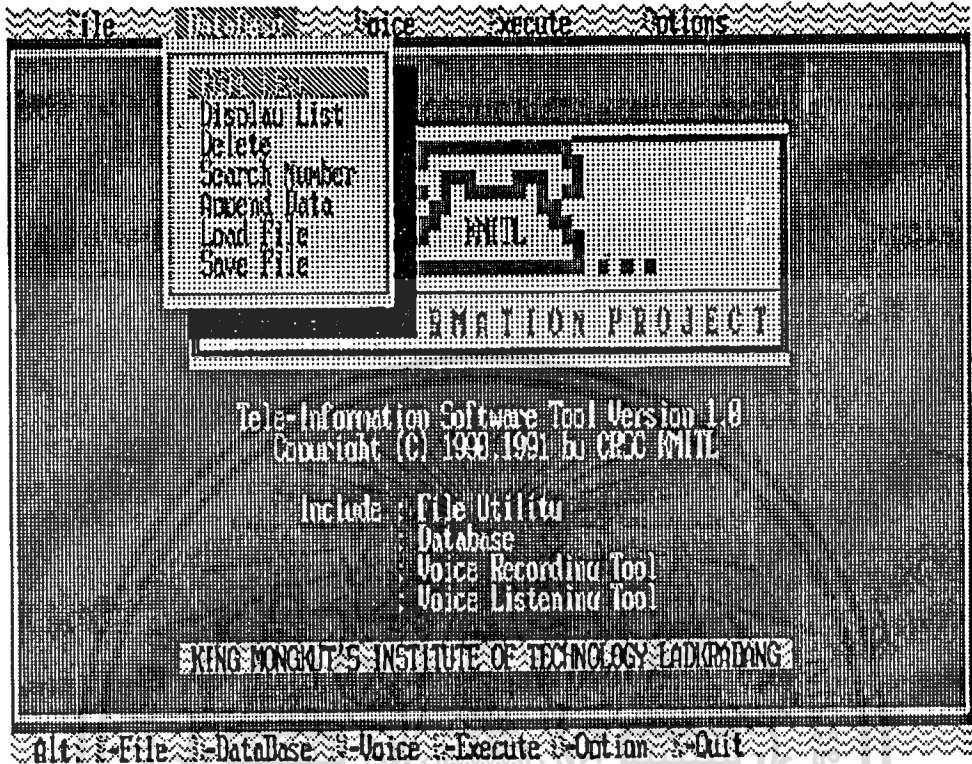


รูปที่ 5.1 แสดงเมนู File

## 5.2 DataBase

เป็นส่วนที่ำใช้สำหรับการจัดการฐานข้อมูล ซึ่งทำงานในลักษณะของดับบลีลิงค์ลิสต์ ดังที่ได้กล่าวมาแล้ว สำหรับเมนูนี้ สามารถแบ่งการทำงานออกเป็นเมนูย่อยได้ 7 เมนู ดังที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
แสดงงานรูปที่ 5.2  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.2 แสดงเมนู DataBase

### 5.2.1 Edit List

เป็นเมนูสำหรับการกำหนดค่าต่าง ๆ บนลิสต์ปัจจุบัน รวมทั้งการสร้างลิสต์ใหม่ ซึ่งเมนูนี้สามารถแยกย่อยได้อีกเป็นดังนี้

1. File Name ใช้สำหรับการกำหนดชื่อไฟล์ของข้อมูล.
2. Title ใช้สำหรับการกำหนดหัวเรื่องของข้อมูล
3. Number Head ใช้สำหรับการกำหนดชื่อของหมายเลข
4. Data Head ใช้สำหรับการกำหนดชื่อของข้อมูล
5. Number Length ใช้สำหรับการกำหนดความยาวของหมายเลขตามที่ต้องการ ซึ่งสามารถกำหนดได้สูงสุด 10 ตัวเลข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. Data Length ใช้สำหรับการกำหนดความยาวสูงสุดของข้อมูลตามต้องการ ซึ่งสามารถกำหนดค่าได้สูงสุด 10 ตัวอักษร

ซึ่งการกำหนดค่าในข้อ 5.6 ทำได้โดยการเลื่อนไฮไลท์มาที่เมนูที่ต้องการแล้วกด Enter จากนั้นก็ใช้ปุ่มลูกศรทางขวา สำหรับการเพิ่มค่า และ ปุ่มลูกศรทางซ้าย สำหรับการลดค่า

7. Edit ใช้สำหรับการเขียนข้อมูลเก็บลงในลิสต์ ซึ่งทำได้โดยการ

- การเขียนหมายเลข เขียนในช่องหมายเลข ซึ่งจะเขียนได้สูงสุดตามจำนวนที่กำหนดไว้ จากนั้นจะต้องกดปุ่ม TAB เพื่อทำการเขียนข้อมูล

- การเขียนข้อมูล เขียนในช่องข้อมูล ซึ่งจะเขียนได้สูงสุดตามจำนวนที่กำหนดไว้ จากนั้นจะต้องกดปุ่ม ENTER สำหรับการเขียนหมายเลขต่อไป หรือ TAB สำหรับการแก้ไขหมายเลขปัจจุบัน

สำหรับการออกจากเมนู Edit นี้ยังไม่สมบูรณ์ ซึ่งจะต้องทำโดยการ กำหนดให้หมายเลขที่เขียนก่อนที่ตรงการจะออก เป็น ' ' คือไม่มีอะไรเลย

การออกจากเมนู Edit List นี้ทำได้โดยการกดปุ่ม ESC

### 5.2.2 Display List

เมนูย่อยในส่วนนี้จะ เป็นหัวใจหลักของเมนู DataBase ซึ่งใช้งานในการแสดงข้อมูลที่เก็บอยู่ในลิสต์ การแก้ไขข้อมูล การกำหนดชื่อไฟล์ของสัญญาณเสียงสำหรับข้อมูล และ การทดสอบฟังสัญญาณเสียงของแต่ละข้อมูลที่ต้องการ ซึ่งลักษณะของโปรแกรมในส่วนนี้ จะเป็นดังรูปที่ 5.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

File	View	Trace	Execute	Solutions
Database Display				
Filename: A-LOTTERY				
Title: Lottery Number List				
8090002	tail2	TAIL2:NUM5:NUM2		
8090003	tail5	TAIL5:NUM7:NUM1:NUM2		
1848225	win5	WIN:NUM5:5		
2231586	win2	WIN:NUM2:189		
3730417	win2	WIN:NUM2:189		
3897236	win5	WIN:NUM5:58		
4365276	win5	WIN:NUM5:5		
5684269	win5	WIN:NUM5:5		
5684526				
6794930	win4	WIN:NUM4:18		
7435898	win1	WIN:NUM1:288		
8432268	win4	WIN:NUM4:18		
9437592	win4	WIN:NUM4:18		
9547627	win5	WIN:NUM5:58		

Edit Voice File    Delete    Append    Insert    Break    Exit

รูปที่ 5.3 แสดงลักษณะของโปรแกรมในส่วน Display List

จากรูป จะเห็นว่า หน้าจอจะแสดงข้อมูลต่างๆ ไว้ ได้แก่

1. ชื่อไฟล์ของข้อมูลชุดนี้
2. หัวเรื่องของข้อมูลชุดนี้
3. ลำดับที่และหน้าของข้อมูลที่ไฮไลต์อยู่
4. ข้อมูล ซึ่งประกอบด้วย หมายเลข ข้อมูล และ ชื่อฟิลด์ซึ่งสัญญาณเสียงสำหรับข้อมูลนั้นๆ

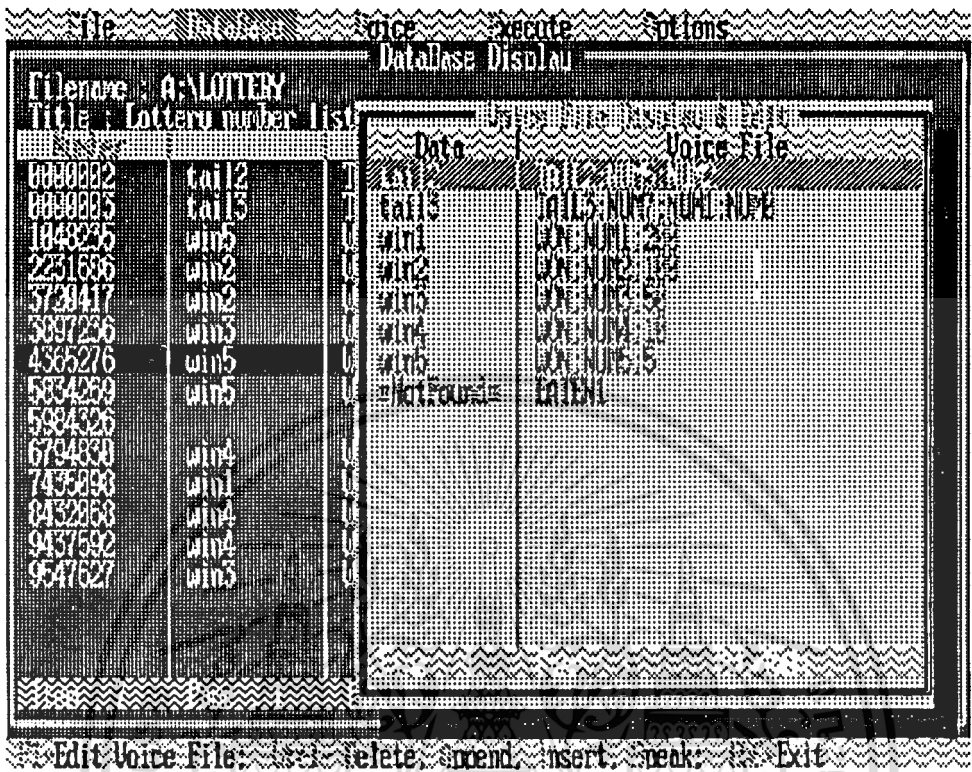
การทำงานของโปรแกรมในส่วนนี้ จะใช้ปุ่มลูกศรขึ้นลงสำหรับการเลื่อนไฮไลต์ไปที่ข้อมูลที่ต้องการ และปุ่ม PgUp, PgDn สำหรับเปลี่ยนหน้า นอกจากนี้ยังมีการใช้ปุ่มอื่นๆ อีก สำหรับการดำเนินงานในลักษณะต่างๆ กัน ได้แก่

1. Ctrl-D ใช้สำหรับการลบข้อมูลสำหรับบรรทัดที่ไฮไลต์ปรากฏอยู่ โดยบรรทัดที่อยู่ถัดไปจะขึ้นมาแทนที่
2. Ctrl-A ใช้สำหรับการแก้ไขข้อมูล โดยเคอร์เซอร์ (Cursor) จะปรากฏขึ้นที่ช่องของข้อมูลให้ทำการแก้ไขได้ เมื่อแก้ไขเสร็จให้กดปุ่ม ENTER
3. Ctrl-I ใช้สำหรับการแทรกหมายเลขที่บรรทัดที่ไฮไลต์ปรากฏอยู่ โดยหมายเลขที่จะแทรกจะต้องมีค่าอยู่ระหว่างหมายเลขของบรรทัด เดิมกับบรรทัดล่าง เมื่อแทรกหมายเลขแล้ว ข้อมูลเดิมที่บรรทัดล่างจะเลื่อนลง 1 บรรทัด
4. Ctrl-S ใช้สำหรับการทดสอบสัญญาณเสียง โดยโปรแกรมจะทำการโหลดสัญญาณเสียง ตามชื่อไฟล์ที่ปรากฏอยู่ในช่องของรายการชื่อไฟล์ที่ไฮไลต์ปรากฏอยู่ ซึ่งไฟล์นั้นจะต้องอยู่ในรากสำหรับไฟล์สัญญาณเสียงที่กำหนดไว้แล้ว ถ้าไม่พบไฟล์นั้น โปรแกรมจะแสดงให้ทราบ
5. ESC ให้เมื่อต้องการออกจากโปรแกรมในส่วนนี้

อนึ่ง ข้อมูลที่อยู่ในช่องข้อมูล สามารถเขียนในลักษณะหลายข้อมูลได้ โดยการเขียนเครื่องหมาย ';' (Semicolon) คั่นระหว่างข้อมูล ซึ่งโปรแกรมจะทราบว่ามีย่อยข้อมูลอยู่ในข้อมูลหมายเลขเดียวกัน

นอกจากนี้ ยังมีปุ่มสำคัญซึ่งใช้สำหรับการกำหนดชื่อไฟล์ของสัญญาณเสียงประจำข้อมูลต่างๆที่แตกต่างกันออกไป ซึ่งปุ่มนี้ได้แก่ ปุ่มฟังก์ชัน 1 หรือ F1 เมื่อกดปุ่มนี้แล้วจะทำให้โปรแกรมเข้าสู่หน้าต่าง (Window) ใหม่ ดังแสดงในรูปที่ 5.4

จากรูปที่ 5.4 หน้าต่างนี้จะแสดงส่วนของข้อมูล ซึ่งโปรแกรมได้ทำการแยกแยะจากข้อมูลในช่องข้อมูลบนหน้าต่างที่แล้ว โดยโปรแกรมจะนำมาสร้างเป็นลิสต์อีกหนึ่งลิสต์ ดังที่ได้อธิบายไว้แล้วในบทที่แล้ว การกำหนดชื่อไฟล์สัญญาณเสียงประจำข้อมูล สามารถทำได้โดยการใช้นู๊บลูกศรขึ้นลง และ PgUp, PgDn ในการบังคับไฮไลต์ให้ไปที่ข้อมูลตามต้องการ ซึ่งผู้ใช้สามารถทำการเขียนชื่อไฟล์ที่ต้องการลงไปได้ทันที และในลักษณะเดียวกับหน้าต่างที่แล้ว ผู้ใช้สามารถกำหนดชื่อไฟล์ได้หลายชื่อไฟล์สำหรับข้อมูลตัวเดียวด้วยการคั่นด้วยเครื่องหมาย ';' เช่นกัน



รูปที่ 5.4 แสดงหน้าต่างสำหรับการกำหนดชื่อไฟล์สัญญาณเสียง

จะเห็นได้ว่า ที่หน้าต่างนี้จะปรากฏข้อมูล Not Found ที่ตำแหน่งสุดท้ายของลิสต์ ซึ่งข้อมูลนี้จะใช้สำหรับการกำหนดชื่อไฟล์ที่ต้องการ เมื่อโปรแกรมทำการค้นหาข้อมูลใดๆ แล้วไม่พบข้อมูลนั้น

การออกจากหน้าต่างนี้ สามารถทำได้โดยการกดปุ่ม ESC ซึ่งหน้าต่างนี้จะปิดลงและโปรแกรมจะทำการแสดงชื่อไฟล์ที่กำหนดใหม่ สำหรับข้อมูลแต่ละข้อมูลที่ปรากฏอยู่

### 5.2.3 Delete

ใช้สำหรับการลบหมายเลข และข้อมูลตามต้องการออกจากลิสต์ ซึ่งสามารถทำได้ 2 แบบ ดังที่ปรากฏบนเมนู ดังนี้

1. Delete Number สำหรับการลบหมายเลขที่ต้องการออกจากลิสต์ ซึ่งโปรแกรมจะถามหมายเลขที่ต้องการลบ ถ้าไม่พบหมายเลขดังกล่าว โปรแกรมจะแสดงให้ทราบ
2. Delete All สำหรับลบข้อมูลทั้งหมดในลิสต์

#### 5.2.4 Search number

ใช้สำหรับการค้นหาข้อมูลตามหมายเลขที่ต้องการ โดยโปรแกรมจะแสดงข้อมูลที่ค้นพบ และจะทำการโหลดสัญญาณเสียงตามชื่อไฟล์ที่ถูกกำหนดไว้แล้ว

#### 5.2.5 Append Data

ใช้สำหรับการแก้ไขข้อมูล ตามหมายเลขที่ต้องการ ซึ่งลักษณะการใช้งาน จะเหมือนกับการทำงานบนหน้าต่าง Display List

#### 5.2.6 Load File

ใช้สำหรับการโหลดไฟล์ที่เป็นฐานข้อมูลซึ่งถูกสร้างและถูกเก็บไว้บนแผ่นดิสก์ ซึ่งโปรแกรมจะทำการโหลดที่รากสำหรับฐานข้อมูลที่ถูกกำหนดไว้แล้ว

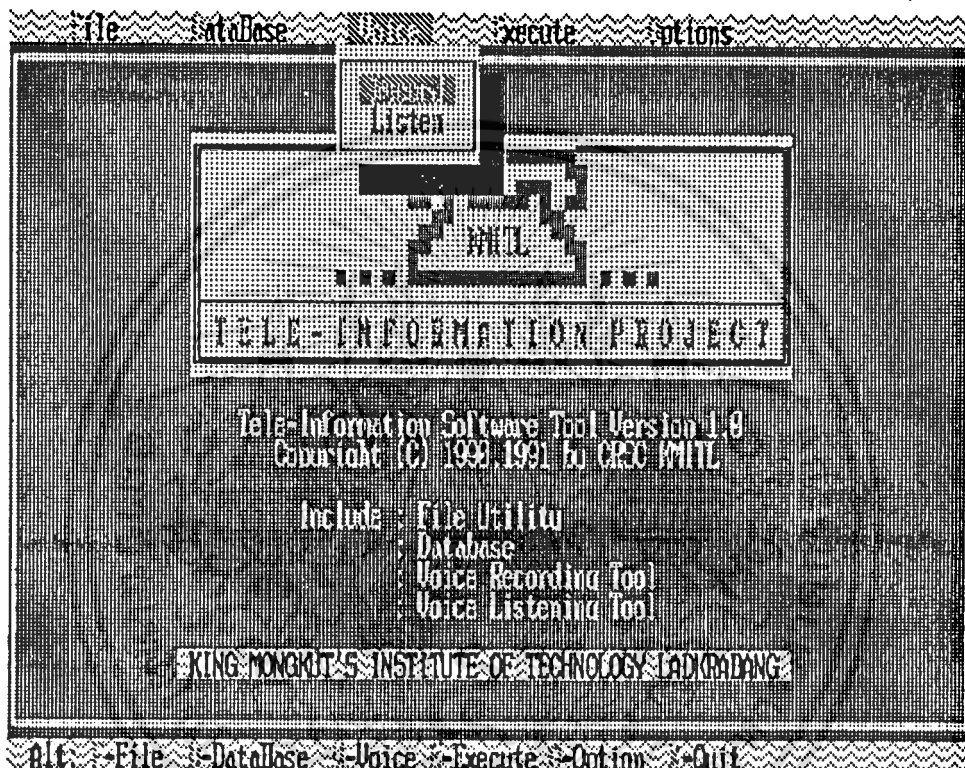
#### 5.2.7 Save File

ใช้สำหรับการเก็บข้อมูลในลิสต์ลงบนแผ่นดิสก์ โดยโปรแกรมจะทำการเก็บข้อมูลเป็น 2 ไฟล์ ดังนี้

1. ไฟล์ไม่มีนามสกุล เป็นไฟล์ที่ใช้เก็บข้อมูลหลัก ได้แก่ หมายเลขและข้อมูลประจำหมายเลขนั้น
2. ไฟล์สกุล .VLT เป็นไฟล์ที่ใช้เก็บข้อมูลรอง ได้แก่ ข้อมูลที่ถูกแยกแล้วและชื่อไฟล์สัญญาณเสียงประจำข้อมูลนั้น

### 5.3 Voice

เป็นส่วนที่ ๑ ใช้สำหรับการบันทึกสัญญาณเสียง และโพลสดสัญญาณเสียงซึ่งแบ่งการทำงาน ออกเป็น 2 เมนู ดังรูปที่ 5.5



รูปที่ 5.5 แสดงเมนู Voice

#### 5.3.1 Record

๑ ใช้สำหรับการบันทึกสัญญาณเสียง ซึ่งลักษณะของโปรแกรมที่แสดง เป็นดังรูปที่ 5.6

การใช้งานเริ่มแรก โปรแกรมจะให้กำหนดชื่อไฟล์ที่ต้องการบันทึก หลังจากนั้นโปรแกรมจะทำการตรวจสอบว่ามีไฟล์นี้อยู่ในรากสำหรับไฟล์สัญญาณเสียงหรือไม่ ซึ่งจะให้ผู้ใช้ตัดสินใจที่จะบันทึกทับไฟล์เดิมหรือไม่ ถ้าไม่ต้องการก็จะต้องกำหนดชื่อไฟล์ใหม่

หลังจากการกำหนดชื่อไฟล์เรียบร้อยแล้ว การเริ่มบันทึกทำได้โดยการกดปุ่ม ENTER

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่สามารถเผยแพร่โดยไม่ได้รับอนุญาต



รูปที่ 5.6 แสดงลักษณะของโปรแกรมในส่วนของการบันทึกสัญญาณเสียง

ขณะที่บันทึกเสียง โปรแกรมจะแสดงให้เห็นว่าได้มีการบันทึกเป็นจำนวนเท่าใดแล้ว โดยแสดงให้เห็นในลักษณะของอักขระแบบกราฟิก ซึ่งการบันทึกจะเป็นไปในอัตรา 2 กิโลไบต์ ต่อเวลา 1 วินาที ดังที่ได้อธิบายไว้แล้วในบทที่ 4 ถ้าต้องการหยุดการบันทึกเมื่อใด ให้กดปุ่มใดๆ โปรแกรมจะทำการเก็บไฟล์ลงแผ่นดิสก์ ด้วยนามสกุล .VCE ที่รากสำหรับไฟล์สัญญาณเสียงที่ถูกกำหนดไว้ การบันทึกนี้สามารถทำได้มากที่สุด 64 กิโลไบต์ ซึ่งถ้าไม่มีการหยุดการบันทึกโดยผู้ใช้งาน โปรแกรมจะหยุดเองเมื่อถึง 64 กิโลไบต์ แล้วทำการเก็บไฟล์

### 5.3.2 Listen

ใช้สำหรับการฟังเสียงที่ได้ทำการบันทึกแล้ว ซึ่งลักษณะของโปรแกรม และการใช้งานจะเหมือนกับส่วนที่แล้ว ในขณะที่ทำการฟังเสียง เราสามารถหยุดได้โดยการกดปุ่มใดๆ หรือโปรแกรมจะหยุดเอง เมื่อทำการโหลดสัญญาณเสียงครบตามจำนวนที่เก็บไว้

## 5.4 Execute

โปรแกรมในส่วนนี้ เป็นส่วนสำคัญและเป็นหัวใจหลักสำหรับโครงการนี้ ซึ่งเป็นส่วนที่เข้าสำหรับการให้บริการสอบถามข้อมูล การทำงานของโปรแกรมนี้ แบ่งออกเป็น 2 เมนู ดังนี้

### 5.4.1 Data File Code

เป็นส่วนที่ผู้ใช้สามารถกำหนดชุดข้อมูลสำหรับรหัสแต่ละรหัส ซึ่งสามารถกำหนดได้ทั้งหมด 12 ชุดข้อมูล ได้แก่ 1-0 และ \*,# ซึ่งส่วนนี้แบ่งออกเป็นเมนูย่อย 3 เมนู ดังนี้

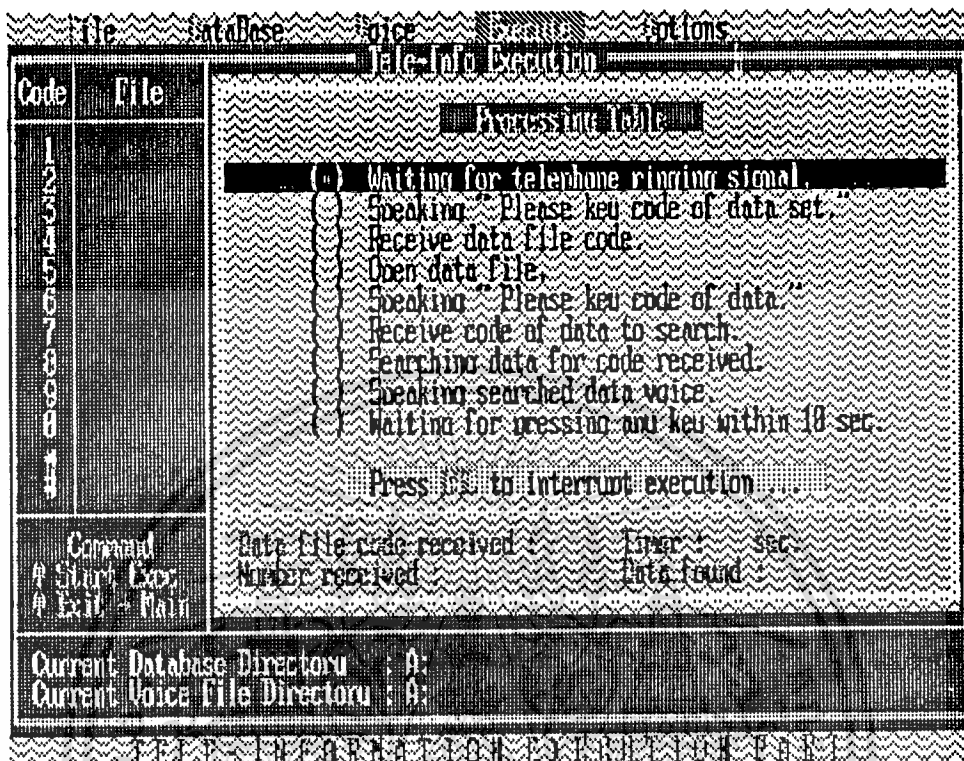
1. Edit Code File ใช้ในการเขียน โดยการเลื่อนไฮไลท์ไปที่รหัสที่ต้องการ แล้วสามารถเขียนได้ทันที
2. Load Code File ใช้ในการโหลดข้อมูลที่เก็บไว้แล้ว จากไฟล์ TINFO.COD
3. Save Code File ใช้ในการเก็บข้อมูลที่เขียนขึ้น ลงไฟล์ TINFO.COD

### 5.4.2 Execution

เป็นส่วนที่ทำงานสำหรับการติดต่อกับโทรศัพท์ โดยการเริ่มทำงาน โปรแกรมจะทำการตรวจสอบไฟล์ทั้งหมดที่จำเป็นต้องใช้ในการทำงานส่วนนี้ ว่ามีอยู่หรือไม่ ถ้าไม่มีโปรแกรมจะแสดงให้ทราบ และจะไม่สามารถให้โปรแกรมในส่วนนี้ทำงานได้ สำหรับไฟล์เหล่านี้ ได้แก่

1. ไฟล์ข้อมูลที่กำหนดไว้ในส่วนของ Data File Code
2. ไฟล์สัญญาณเสียงที่จำเป็น ได้แก่ หมายเลข, คำแนะนำการใช้งาน เป็นต้น

หลังจากการตรวจสอบไฟล์ทั้งหมดแล้ว โปรแกรมจะเปิดหน้าต่างใหม่ ซึ่งลักษณะของหน้าต่างในส่วนนี้ เป็นดังรูปที่ 5.7



รูปที่ 5.7 แสดงหน้าต่างสำหรับเมนู Execution

จากรูป จะเห็นว่าหน้าจอจะแสดงข้อมูลต่างๆ ดังนี้

1. ชุดข้อมูลประจำรหัสที่ถูกกำหนดไว้ก่อนในส่วนของ Data File Code
2. รากประวัติสำหรับไฟล์ฐานข้อมูลและไฟล์สัญญาณเสียง
3. เมนูคำสั่ง สำหรับสั่งโปรแกรมให้ เริ่มทำงานหรือออกจากโปรแกรม
4. ตารางแสดงขั้นตอนการทำงาน (Processing Table) และ ข้อมูลที่ค้นหา

หลังจากที่ผู้เข้าทำการสั่งให้โปรแกรมเริ่มทำงาน จะเกิดไฮไลท์ขึ้นที่บรรทัดแรกของ ตารางแสดงขั้นตอนการทำงาน หมายความว่า โปรแกรมกำลังรอสัญญาณโทรศัพท์ ซึ่งขั้นตอนนี้จะเป็น เพียงขั้นตอนเดียวที่ผู้เข้าสามารถที่จะหยุดการทำงานได้ โดยการกดปุ่ม ESC

เมื่อมีสัญญาณกริ่งโทรศัพท์มา โปรแกรมจะทำงานตามขั้นตอนที่แสดงตามตาราง ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. ส่งสัญญาณเสียงเพื่อบอกให้ทำการกรหัสสำหรับชุดข้อมูลที่ต้องการทราบ
2. ทำการรับรหัส ซึ่งในขณะที่โปรแกรมจะทำการจับเวลา ให้ทำการกรหัสภายใน 60 วินาที โดยโปรแกรมจะแสดงจำนวนเวลาที่ทำการจับ ที่ส่วนของ Timer : sec. ถ้าเกินเวลาที่กำหนด โปรแกรมจะหยุด แล้วกลับโปรอรับสัญญาณกรังโทรศัพท์ครั้งใหม่
3. ทำการเปิดไฟล์ตามรหัสนั้นๆ ถ้ารหัสที่กดเข้ามาไม่มีไฟล์สำหรับรหัสนั้น โปรแกรมจะส่งสัญญาณเสียงไปบอก แล้วกระโดดมาทำงานตามขั้นตอนที่ 1 ใหม่

โปรแกรมจะแสดงหมายเลขรหัสที่รับได้ ที่ส่วนของ Data File Code Receive รวมทั้งจะสร้างไฮไลท์ที่ตารางของรหัสและชุดข้อมูลที่อยู่ทางซ้ายมือ

4. หลังจากเปิดไฟล์แล้ว โปรแกรมจะส่งสัญญาณเสียงเพื่อบอกให้กรหัสข้อมูลที่ต้องการทราบ

5. ทำการรับรหัส ซึ่งในขณะที่โปรแกรมจะทำการจับเวลา ให้ทำการกรหัสภายใน 60 วินาที โดยโปรแกรมจะแสดงจำนวนเวลาที่ทำการจับ ที่ส่วนของ Timer : sec. ถ้าเกินเวลาที่กำหนด โปรแกรมจะหยุด แล้วกลับโปรอรับสัญญาณกรังโทรศัพท์ครั้งใหม่ และรหัสที่รับมาได้ จะถูกแสดงที่ส่วนของ Number receive :

6. ทำการค้นหาข้อมูลตามรหัสที่ได้รับมา ข้อมูลที่ค้นพบ จะถูกแสดงที่ส่วนของ Data found :

7. ส่งสัญญาณเสียงตามข้อมูลที่ค้นพบ รวมทั้งบอกให้ทำการกดปุ่มใดๆ ถ้าต้องการใช้บริการต่อ ภายใน 10 วินาที

8. ทำการรอรับการกดปุ่มใดๆ โดยการจับเวลาจะแสดงในลักษณะของอีกขระกราฟฟิก ถ้าเกินเวลาที่กำหนด โปรแกรมหยุด แล้วกลับโปรอรับสัญญาณกรังโทรศัพท์ครั้งใหม่ ถ้ามีการกดภายในเวลาที่กำหนด โปรแกรมจะกระโดดไปทำงานตามขั้นตอนที่ 1 ใหม่

## 5.5 Options

เมนูในส่วนนี้ จะใช้สำหรับการกำหนดค่าคงที่สำหรับโปรแกรมหลัก ซึ่งแบ่งออกเป็นเมนูย่อยได้ดังนี้

ใช้สำหรับการกำหนดรากสำหรับเก็บหรือโหลดไฟล์ ที่ใช้ในโปรแกรมหลัก โดยแบ่ง ออกได้เป็น

- DataBase Directory ใช้ในการกำหนดรากสำหรับไฟล์ฐานข้อมูล
- Voice Directory ใช้ในการกำหนดรากสำหรับไฟล์สัญญาณเสียง

### 5.5.2 Load Config

ใช้สำหรับโหลดข้อมูลค่าคงที่ที่ถูกเก็บไว้แล้ว ในไฟล์ชื่อ TINFO.CON

### 5.5.3 Save Option

ใช้สำหรับเก็บข้อมูลค่าคงที่ที่กำหนดไว้ ลงแผ่นดิสก์ ด้วยไฟล์ชื่อ TINFO.CON ซึ่งผู้  
ใช้สามารถกำหนดชื่อไฟล์ใหม่ในการเก็บได้ เพื่อให้สามารถเก็บค่าคงที่ได้หลายรูปแบบ

## บทที่ 6

### บทสรุปและวิจารณ์

จากการพัฒนาโครงงานนี้มาทั้งหมด ทำให้สามารถสรุปคุณสมบัติหลักในการทำงานของเครื่องบริการสอบถามข้อมูลทางโทรศัพท์อัตโนมัติ ได้ดังนี้

1. มีระบบการสร้าง จัดเก็บ และค้นหาข้อมูล ซึ่งเป็นโปรแกรมที่ทำงานบนเครื่อง IBM PC สำหรับโปรแกรม
2. มีระบบการบันทึก และไหลดสัญญาณเสียง โดยแสดงผลทางจอภาพ IBM PC
3. มีระบบการบริการผ่านทางโทรศัพท์ ซึ่งผู้ใช้สามารถสอบถามข้อมูลประจำชุดข้อมูลใดก็ได้ ผ่านทางการใช้โทรศัพท์ ในส่วนนี้ถือเป็นหัวใจหลักสำหรับคุณสมบัติการทำงาน ของเครื่องนี้

นอกจากนี้ โปรแกรมยังมีระบบการจัดการด้านยูทิลิตี้เกี่ยวกับไฟล์ และการกำหนด ตัวเลือก (Options)

สำหรับปัญหาและข้อจำกัดบางอย่างสำหรับโครงงานนี้ รวมทั้งช่องทางการแก้ไขนั้น สามารถอธิบายเฉพาะที่สำคัญ ได้ดังนี้

1. ปัญหาที่เกิดจากสัญญาณรบกวน ซึ่งมาจากการทำงานของวงจรส่วนควบคุมการทำงาน ที่สร้างขึ้นโดยการบัดกรีแบบไวร์แรป (Wire Wrapping) บนแผ่นวงจรพิมพ์แอนะล็อก และการใช้ไฟเลี้ยงร่วมกับวงจรส่วนอื่นๆ ซึ่งทำให้การบันทึกสัญญาณเสียงนั้น มีการบันทึกสัญญาณรบกวนลงไปด้วย เนื่องจากสัญญาณรบกวนที่อินพุตของแอมพลิฟายเออร์ การแก้ไขสามารถทำได้โดยการสร้างแผ่นวงจรพิมพ์ขึ้นแทนการไวร์แรป และแยกไฟเลี้ยงสำหรับวงจรส่วนควบคุมการทำงานนี้กับวงจรส่วนอื่นๆ

2. ข้อจำกัดการทำงานของเครื่องนี้ ที่สำคัญคือ การทำงานร่วมกับเครื่อง IBM PC นั้น จำกัดเฉพาะกับการทำงานบนเครื่องที่ทำงานที่ความถี่นาฬิกาตั้งแต่ 10 MHz ขึ้นไป ถ้าให้ทำงานที่ความถี่ต่ำกว่านี้ จะปรากฏว่าการบันทึกและไหลคัสต์สัญญาณเสียงนั้นผิดพลาด ซึ่งสัญญาณเสียงที่ได้นั้นจะเพี้ยนและมีความถี่ต่ำลง ซึ่งข้อจำกัดนี้มาจากการที่ IBM PC ทำงานไม่ทันกับจังหวะการทำงานอินเทอร์พรีทของ Z80 สำหรับข้อจำกัดนี้สามารถแก้ไขได้ 2 วิธี ได้แก่

2.1 แก้ไขโปรแกรมบน IBM PC บางส่วน โดยเขียนขึ้นใหม่ด้วยภาษาแอสเซมบลี แทนการใช้ภาษาปาสคาล สำหรับโปรแกรมในส่วนที่ใช้งานร่วมกับ Z80 ที่ทำงานในช่วงของการอินเทอร์พรีท

2.2 แก้ไขโครงสร้างของโปรแกรม รวมทั้งฮาร์ดแวร์ ให้มีการใช้งานของหน่วยความจำชั่วคราว (Buffer) สำหรับการเก็บข้อมูลสัญญาณเสียง ก่อนที่จะมีการบันทึกหรือไหลคัสต์สัญญาณเสียง ตามจังหวะของการอินเทอร์พรีท

การทำงานทั้งหมดนั้น ถือว่าผ่านจุดประสงค์ขั้นต้นแล้ว ตามที่ได้สรุปคุณสมบัติไว้ในตอนแรก ซึ่งโครงการนี้สามารถนำมาพัฒนาให้มีคุณสมบัติการทำงานมากขึ้นกว่าเดิม โดยแนวทางในการพัฒนาโครงการนี้ มีดังนี้

1. พัฒนาให้สามารถทำงานได้กับตู้สายโทรศัพท์จำนวนมากขึ้น
2. พัฒนาให้ผู้ใช้สามารถฝากข้อมูลผ่านทางโทรศัพท์ได้ โดยวิธีสลับผ่าน
3. พัฒนาให้สามารถใช้งานได้กว้างขวางขึ้น เช่น ให้สามารถนับจำนวนเลข โดยโปรแกรมได้อย่างอัตโนมัติ เป็นต้น

### กิติกรรมประกาศ

โครงการเครื่องบริการสอบถามข้อมูลทางโทรศัพท์อัตโนมัติ และปริ้นท์เนอร์เล่มนี้สำเร็จไปห้วงดีไฉนนั้น เนื่องจากการได้รับการสนับสนุนช่วยเหลือจากหลายๆ ฝ่ายด้วยกัน โดยการให้ข้อมูล อุปกรณ์ ซอฟแวร์สนับสนุน ซึ่งเป็นประโยชน์อย่างมากต่อการดำเนินงาน

ขอขอบคุณอาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์พลผดุง ผดุงกุล และ อาจารย์วิชา ศรีบัญชาพงศ์ ที่ได้กรุณาให้คำแนะนำ และอำนวยความสะดวกในค่านอุปกรณ์เครื่องมือและสถานที่ทดลองวิจัยเป็นอย่างดี

นอกจากนั้น ขอขอบคุณกลุ่ม

- โครงการ MIDI ที่สนับสนุนให้คำแนะนำในการใช้งานโปรแกรมสนับสนุนซอฟต์แวร์ของโครงการสามารถใช้กับ mouse ได้
- โครงการ Electronics Mail ซึ่งสนับสนุนอุปกรณ์ทางอิเล็กทรอนิกส์

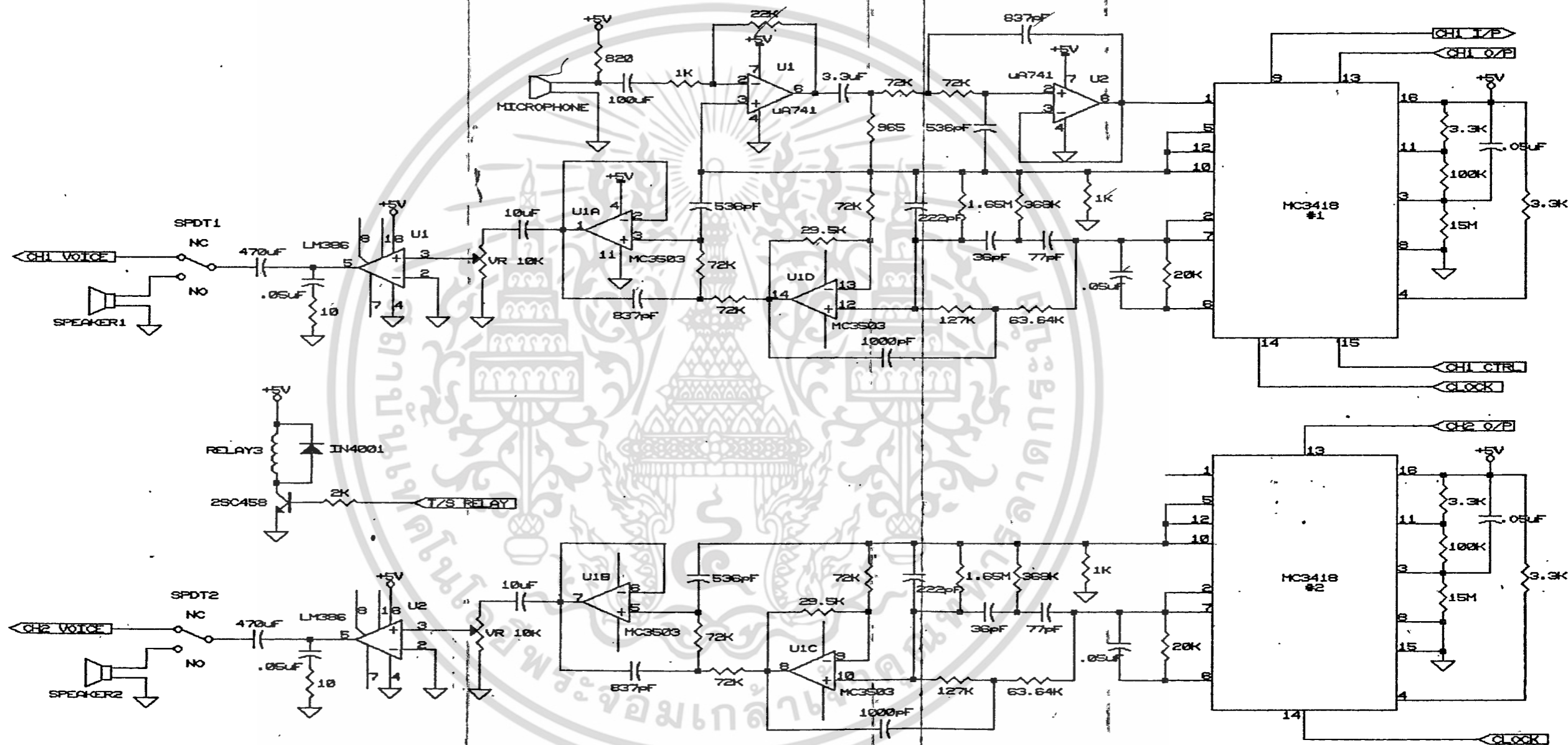
ท้ายสุดนี้กลุ่มผู้จัดทำโครงการขอขอบคุณเป็นอย่างสูงต่อ สำนักวิจัยและบริการคอมพิวเตอร์เป็นอย่างยิ่งที่ให้อุปกรณ์ต่างๆ

หนังสืออ้างอิง

1. James W. Coffron, *Z80 Application*, Sybex Inc., 1983.
2. William Simnema, Tom McGovern, *Digital, Analog, and Data Communication*, Second Edition. Prentice-Hall International Inc., 1986.
3. Gobind Daryanani, *Principles of Active Network Synthesis and Design*, Bell Laboratories Inc., John Wiley & Sons, 1976.
4. Rodney Zaks, *Programming the Z80*, Third Edition, Sybex Inc., 1980.
5. *IBM PC/XT Technical Reference Manual*, International Business Machines Corporation, 1983.
6. *Motorola Linear/Interface Device*, Motorola.
7. *Optoelectronics Data Book*, Texas Instruments Inc., , 1983-84.
8. *The TTL Data Book*, Texas Instruments Inc., Second Edition.
9. *Turbo Pascal Version 5.0.*, Borland International Inc.,
10. สุรศักดิ์ สงวนพงษ์, *แคว้นวิช เทอร์โบปาสคาล*, บริษัทซีเบ็คยูเคชั่น, 2532.

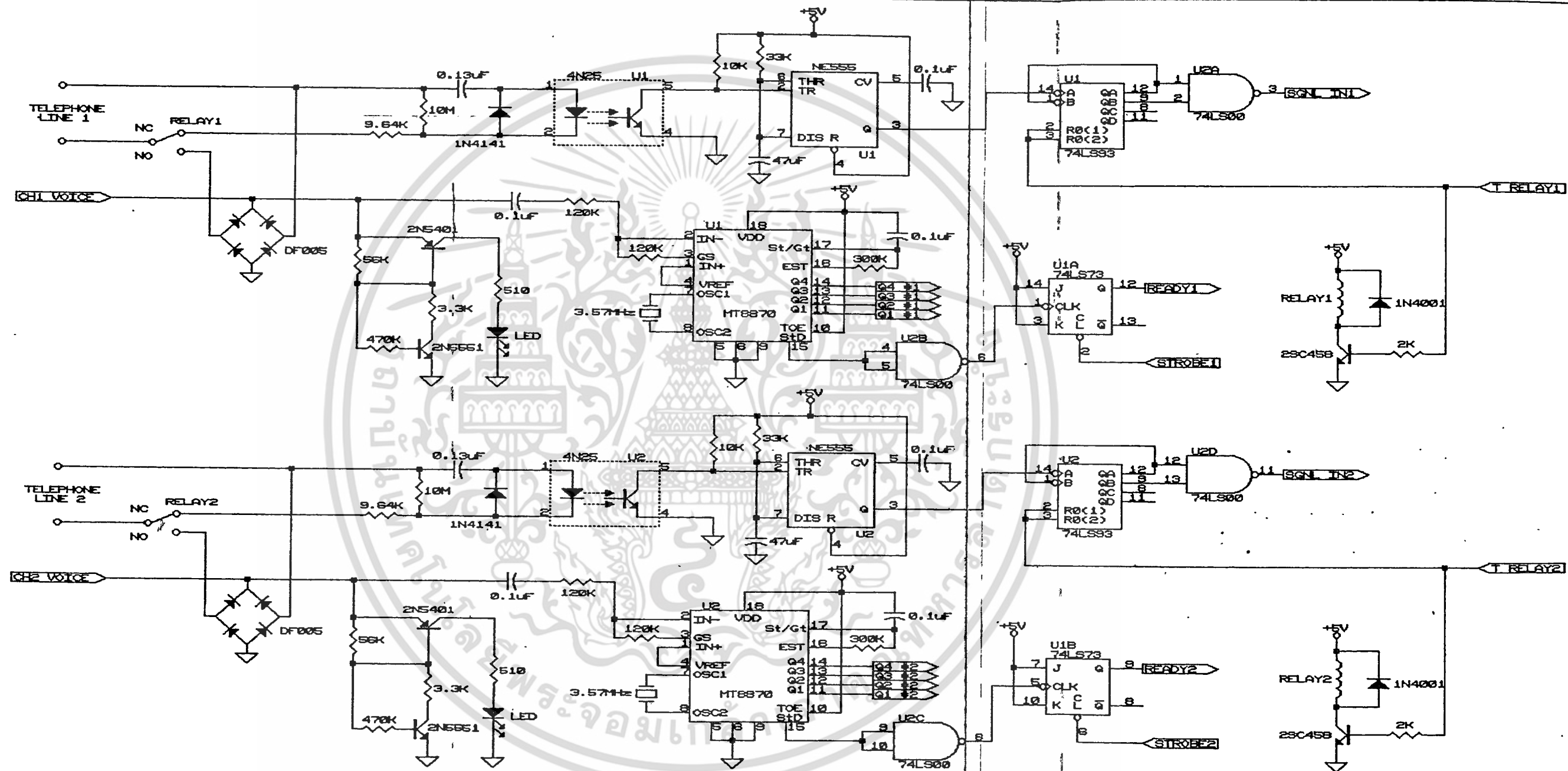


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



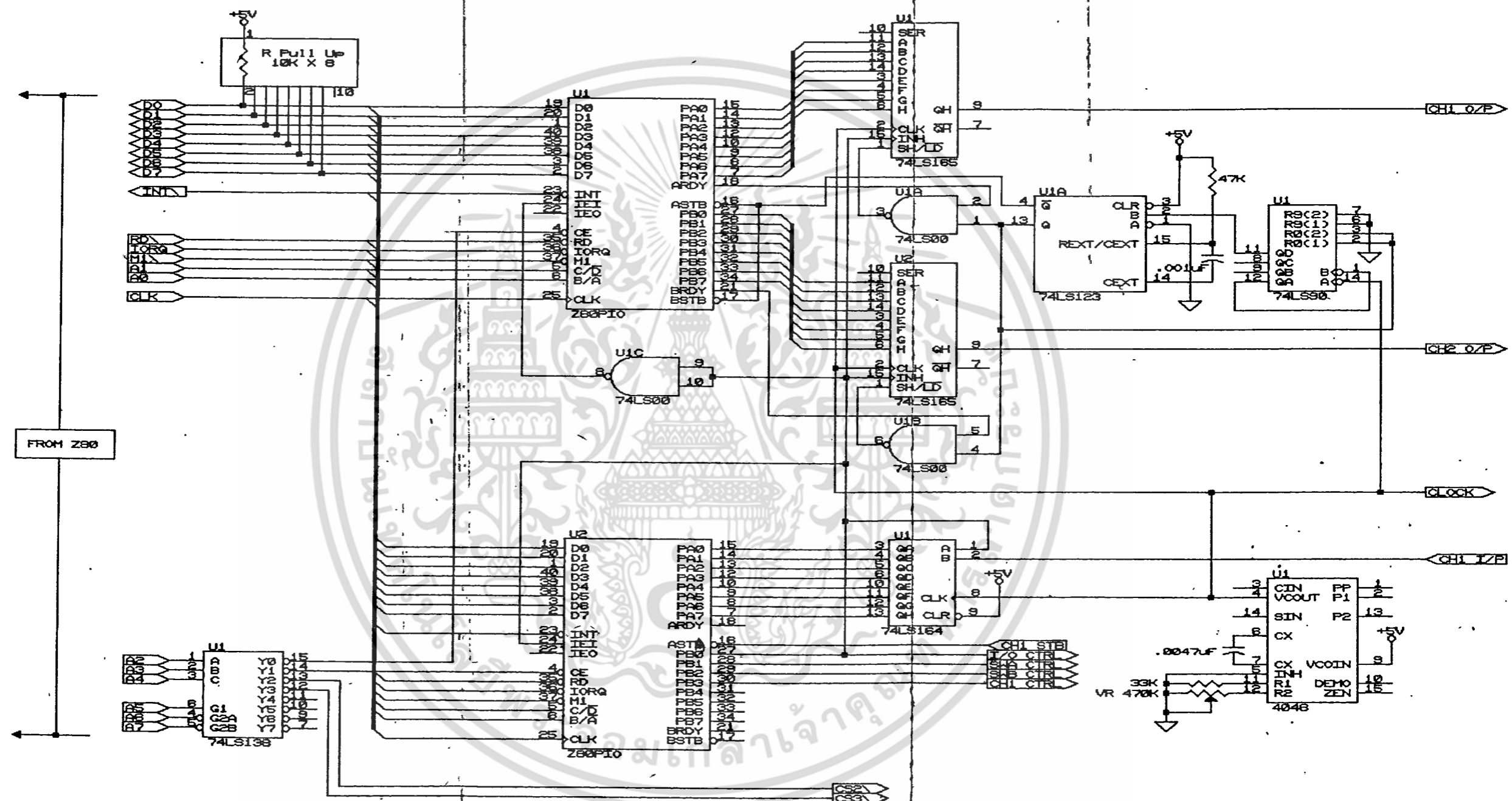
Electronics Project	KING MONKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG	
Student	Title	
Anurak T.	TELE-INFORMATION PROJECT SCHEMATIC	
Hassarangsi S.	Size	Document Number
Nattapol M.	B	VOICE CONVERTER PART
Electronics '4C	Date:	November 20, 1990 Sheet 2 of

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Electronics Project	KING MONKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG	
Student	Title	
Anurak T. Hassarangsi S. Nattapol M.	TELE-INFORMATION PROJECT SCHEMATIC	
Electronics '4C	Size Document Number	1.
	B	TELEPHONE INTERFACE PART
	Date: January 3, 1991	Sheet 1 of

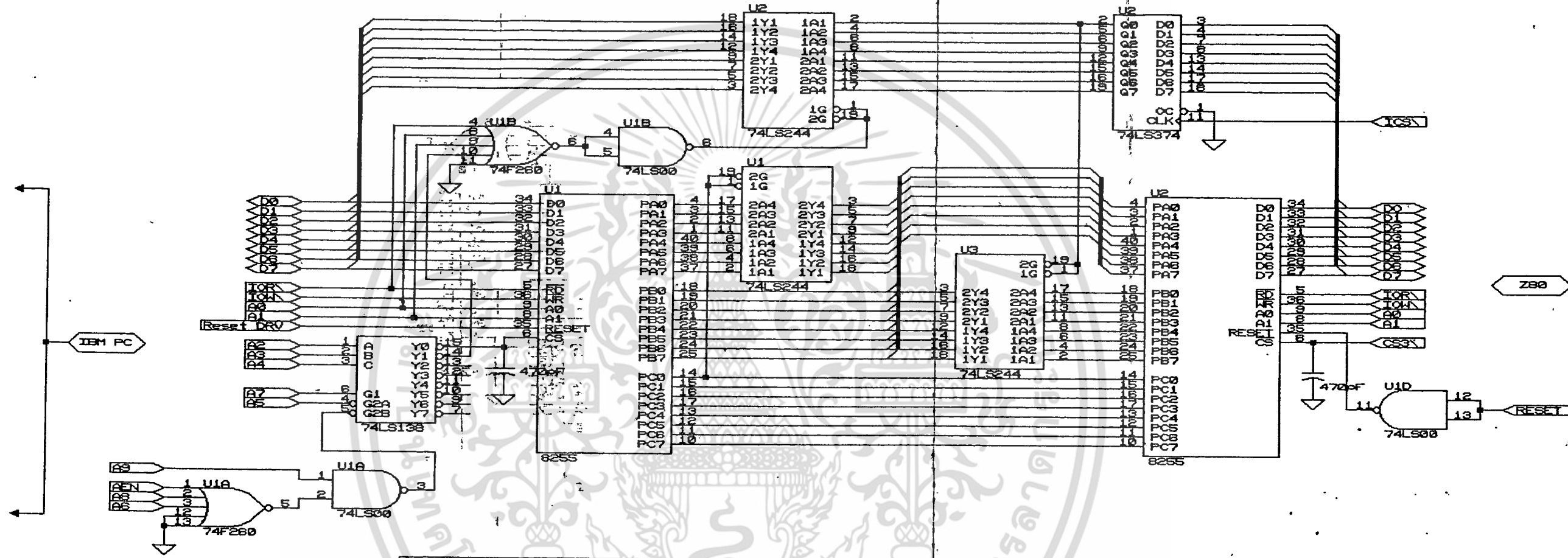
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



CLOCK : 16 KHz

Electronics Project	KING HONGKUT'S INSTITUTE OF TECHNOLOGY LADKABANG	
Student	Title	
Arunak T. Hassarangsri S. Nattapol M.	TELE-INFORMATION PROJECT SCHEMATIC	
Electronics '4C	Size	Document Number
	B	VOICE DATA INTERFACE PART
	Date: November 29, 1980	Sheet 3 of

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Pq 0004h :: IBM PC Data Output  
 Pq 0001h :: IBM PC Data Input  
 Pq 0003h :: IBM PC Status Output  
 Pq 0002h :: 8255 #1 Control Port  
 Pq 0004h :: Z80 Status Input

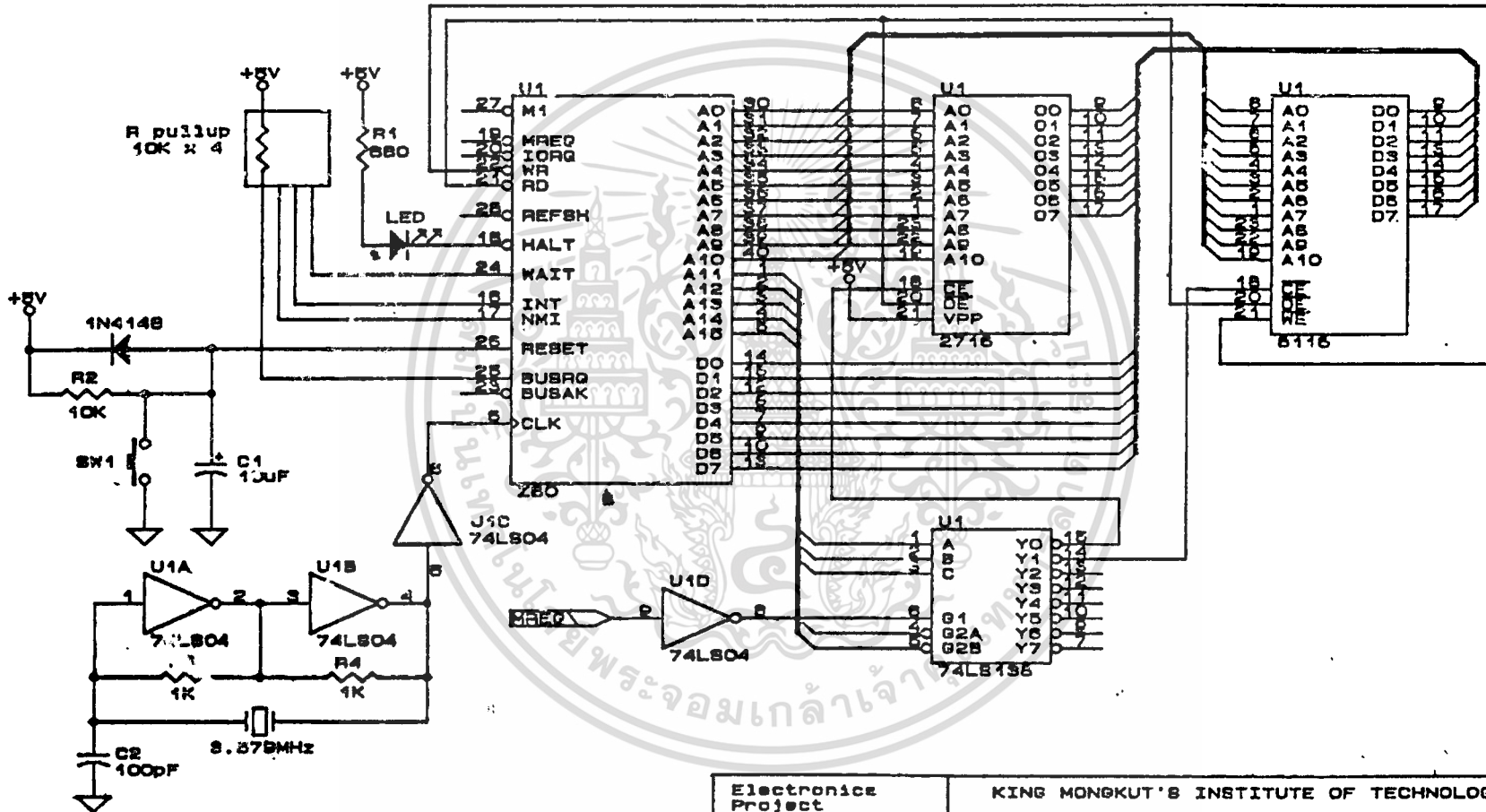
Pq 0000h :: Z80 Status Output  
 Pq 0001h :: Z80 Data Input  
 Pq 0002h :: Z80 Data Output  
 Pq 0003h :: IBM PC Status Input  
 Pq 0004h :: 8255 #2 Control Port

Electronics Project	KING HONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG	
Student	Title	
Anurak T. Hassarangri S. Nattapol M.	TELE-INFORMATION PROJECT SCHEMATIC	
Electronics '4C	Size	Document Number
	B	IBM PC & Z80 INTERFACE PART
	Date:	February 27, 1991 Sheet ... 4 of ...

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Electronics Project	KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKABANG		
Student	Title		
Anurak T. Hassanangeri S. Nattapol M.	TELE-INFORMATION PROJECT SCHEMATIC		
Electronics '4C	Size	Document Number	REV
	A	TELEPHONE INPUT/OUTPUT	1.0
	Date:	February 27, 1991	Sheet 5 of 6



Electronic Project	KING MONKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG		
Student	Title		
Anurak T. Haseerangeri S. Nattapol M.	TELE-INFORMATION PROJECT SCHEMATIC		
Size	Document Number	REV	
A	Z80 CPU CONTROL BOARD	1.0	
Electronics '4C	Date: February 27, 1991	Sheet	6 of 6