เครื่องชุมสายโทรศัพท์ตู้สาขาอัตโนมัติ

(PRIVATE AUTOMATIC BRANCH EXCHANGE)

โดย

นาย กิติพงศ์ แก้วบุตรดี 37013371

นาย ธรรมศักดิ์ ยังน้อย 37013378

นาย มงคล ยศสุนทร 37013389

อาจารย์ที่ปรึกษา

รศ. นิกร สุขุตมตันติ

วัน เดือน ปี 1 ต.ค. 2541
เลขทะเบียน 038340
เลขเรียกหนังสือ T 373.60 ส 474ค

ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

ภาควิชาเทคนิคอุตสาหกรรม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2539

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

038340

หัวข้อโครงงาน	เครื่องชุมสายโทรศัพท์ตู้สาขาอัตโนมัติ (PRIVATE AUTOMATIC BRANCH EXCHANGE)
โดย	นาย กิติพงศ์ แก้วบุตรดี 37013371 นาย ธรรมศักดิ์ ยังน้อย 37013378 นาย มงคล ยศสุนทร 37013389
อาจารย์ที่ปรึกษา	รศ. นิกร สุขุตมตันติ
ภาควิชา	เทคนิคอุตสาหกรรม
สาขาวิชา	เทคโนโลยีโทรคมนาคม
ปีการศึกษา	2539

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของหลักสูตรปริญญาอุตสาหกรรมศาสตรบัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

.....ประธานกรรมการ

(.....)

.....กรรมการ

(.....)

.....กรรมการ

(.....)

.....กรรมการ

(.....)

.....กรรมการ

(.....)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องชุมสายโทรศัพท์ตู้สาขาอัตโนมัติ

(PRIVATE AUTOMATIC BRANCH EXCHANGE)

โดย

นาย กิติพงศ์ แก้วบุตรดี 37013371

นาย ธรรมศักดิ์ ยังน้อย 37013378

นาย มงคล ยศสุนทร 37013389

อาจารย์ที่ปรึกษา

รศ. นิกร สุขุตมตันติ

ปีการศึกษา 2539

บทคัดย่อ

ปริญญานิพนธ์นี้นำเสนอโครงงานเครื่องชุมสายโทรศัพท์ตู้สาขาอัตโนมัติ มีคุณสมบัติคือ 3 สายภายนอก 16 สายภายใน ซึ่งเป็นระบบอัตโนมัติทั้งหมด โดยผู้ที่จะติดต่อนั้นสามารถต่อเลขหมายภายในได้ด้วยตนเอง และเครื่องเลขหมายภายในนั้นสามารถใช้พร้อม ๆ กันได้ และเครื่องชุมสายโทรศัพท์ตู้สาขาอัตโนมัตินี้ จะช่วยให้ผู้ต้องการศึกษาเกี่ยวกับระบบด้านนี้เข้าใจระบบได้ดียิ่งขึ้น โดยมีข้อมูลอยู่พร้อมแล้วสำหรับการศึกษาโครงงานนี้ต่อได้ และอุปกรณ์ที่ใช้ในโครงงานนี้ทั้งหมดสามารถหาซื้อได้ภายในประเทศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRIVATE AUTOMATIC BRANCH EXCHANGE

MR. KITTIPONG KAEWBUDDEE 37013371

MR. THAMMASAK YOUNGNOI 37013378

MR. MONGKOL YOSSOONTORN 37013389

ADVISOR

ASS PROF.. NIKORN SUKUTAMATANTI

FISCAL YEAR 1996

ABSTRACT

Private Automatic Branch Exchange consist of 3 external lines and 16 internal lines with an automatic system. It will benefit a telephoner to be able to connect to an internal number by himself. It will also make it possible to use several internal numbers at the same time.

Private Automatic Branch Exchange help benefit an interested person by providing the relevant information for any further study in the future. Moreover, all the equipment used in the Private Automation Branch Exchange are available domestically.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้าที่
บทคัดย่อ		1
บทที่ 1 บทนำ		1
1.1	วัตถุประสงค์ของโครงงาน	1
1.2	SPECIFICATION	1
1.3	BLOCK DIAGRAM	2
บทที่ 2 ทฤษฎีแบบชุมสาย PABX		4
2.1	ทฤษฎีระบบชุมสาย PABX	4
2.2	BLOCK DIAGRAM	12
บทที่ 3 การทำงานของวงจร และ CIRCUIT DIAGRAM		19
3.1	วงจร INTER FACE	19
3.2	วงจร TOT LINE INTERFACE CIRCUIT	19
3.3	วงจร DETECT BUSY	20
3.4	วงจร SWITCHING PART	20
3.5	SIGNAL	20
3.5.1	DIAL TONE	20
3.5.2	BUSY TONE	21
3.5.3	RINGING TONE	21
3.5.4	RINGING SIGNAL	21
3.6	VOICE / PLAYBACK DEVICE	21
บทที่ 4 สรุปผลและวิจารณ์		54
บรรณานุกรม		55
ภาคผนวก		56
ก. การใช้เครื่อง		
ข. โปรแกรมการทำงานของระบบ		
ค. คู่มือการทดสอบและโปรแกรมการทดสอบ		
ง. DATA ของอุปกรณ์ที่สำคัญในโครงงานปริญญานิพนธ์		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 วัตถุประสงค์ของโครงงาน

ปัจจุบันนี้เทคโนโลยีทางด้านโทรคมนาคมได้มีการพัฒนาและการแข่งขันสูงไม่ว่าจะเป็นระบบการสื่อสารผ่านดาวเทียม ระบบการสื่อสารผ่านเคเบิ้ลใยแก้ว ระบบโทรศัพท์เคลื่อนที่ซึ่งสังคมทุกวันนี้เป็นสังคมแห่ง ข่าวสาร และโครงข่ายขององค์การโทรศัพท์นั้น ยังไม่สามารถสนองตอบต่อความต้องการของผู้ใช้บริการได้เพียงพอ จึงทำให้เกิดมีการพัฒนาระบบชุมสายโทรศัพท์ ตู้สาขาอัตโนมัติ (PRIVATE AUTOMATIC BRANCH EXCHANGE) ขึ้นมาใช้งาน เพื่อเชื่อมต่อระหว่างโครงข่ายขององค์การโทรศัพท์ที่มีเลขหมายจำนวนน้อย และทำให้ขยายระบบเลขหมายภายใน ซึ่งมีจำนวนมากกว่าให้สามารถใช้งานได้

โดยโครงงานนี้ผู้จัดทำให้มีความต้องการให้เป็นแบบอัตโนมัติทั้งหมดโดยผู้ที่จะติดต่อนั้นสามารถต่อเลขหมายภายในได้ด้วยตนเอง และเครื่องเลขหมายภายในนั้นสามารถใช้พร้อม ๆ กัน โดยจะไม่มีสถานะไม่พร้อมสำหรับการใช้งานเลยสำหรับเลขหมายภายใน

1.2 SPECIFICATION

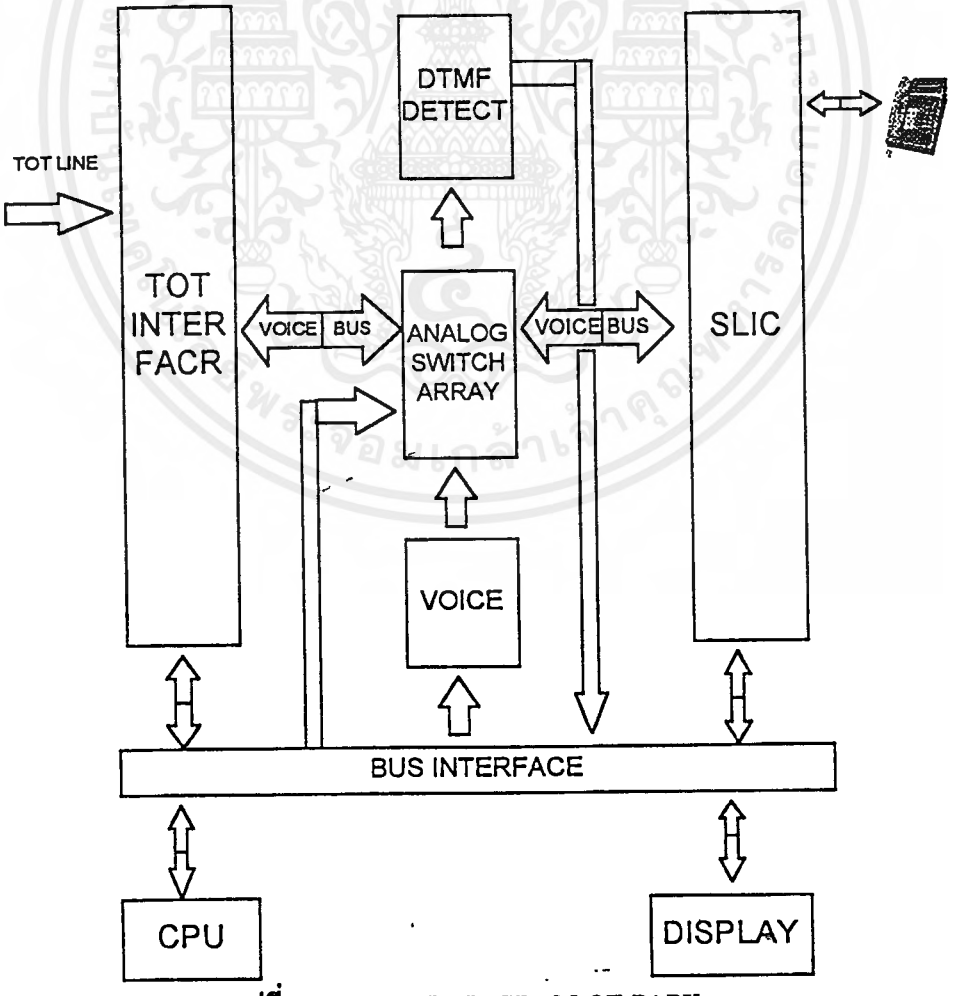
- คุณสมบัติ PABX โครงงานนี้
  - 3 IN PUT SUBSCRIBER LINE
  - 16 OUT PUT SUBSCRIBER LINE
  - AUTOMATIC RECEIVING & TRANSFERING
  - LCD DISPLAY
  - HIGH RELIABILITY
  - LOW NOISE
  - ANALOG SPEECH PATH CONTROLED BY CPU
  - FAST OPERATION
  - MINIMUM CROSSTALK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- <u>โครงงานนี้แบ่งเป็น 2 ส่วน คือ</u>

1. HARD WARE ประกอบด้วย
   - CPU
   - ELECTRONICS SWITCHING SYSTEM
   - SUBSCRIBER LINE INTERFACE CIRCUIT
   - TOT LINE INTERFACE CIRCUIT
   - SIGNALLING ต่าง ๆ ในระบบ
   - INTERFACE DEVICE ที่ทำหน้าที่เป็น INPUT/OUTPUT PORT
   - LCD DISPLAY UNIT
2. SOFTWARE
   - PART OF SWITCHING SYSTEM
   - PART OF DISPLAY

1.3 BLOCK DIAGRAM



รูปที่ 1.1 BLOCK DIAGRAM OF PABX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

- จาก BLOCK DIAGRAM เราสามารถแบ่งออกได้เป็น BLOCK ย่อยได้ 7 BLOCK คือ

1. TOT INTERFACE
2. ANALOG SWITCH ARRAY
3. VOICE
4. SUBSCRIBER LINE INTERFACE CIRCUIT (SLIC)
5. CPU
6. DIGITAL TONE MULTI FREQUENCY DETECT (DTMF)
7. DISPLAY

จาก BLOCK DIAGRAM ข้างต้นแต่ละ BLOCK จะทำหน้าที่ดังนี้

- TOT LINE INTERFACE

จะทำหน้าที่ติดต่อกับสายภายนอกโดยจะทำการ CHECK สัญญาณการโทรเข้าและทำการ COUPLING สัญญาณ และทำการ DETECT สัญญาณของสถานะสายต่าง ๆ เพื่อบอกให้ CPU ทราบสถานะของสายนั้น ๆ

- ANALOG SWITCH ARRAY

ANALOG SWITCH ARRAY จะทำหน้าที่เป็นตัวตัด,ต่อคู่สายต่าง ๆ และสัญญาณต่าง ๆ ให้กับคู่สายโดยจะทำตามที่ CPU สั่ง โดยจะเป็นตัวเชื่อมการติดต่อของคู่สายที่สำคัญยิ่ง

- SLIC

SLIC นั้นเป็นตัวหนึ่งที่มีความสำคัญยิ่งถึงจะทำการ INTERFACE กับสายภายใน ในการ FEED BATTERY ให้กับสายภายใน และทำการเชื่อมต่อสัญญาณ และทำการส่งสัญญาณกระดิ่งตลอดจนการตรวจสอบ การยกหูของสายภายใน แจ้งให้ CPU ทราบถึงสภาพสายภายในดังกล่าว

- CPU

CPU เป็นตัวหนึ่งที่มีความสำคัญ โดยจะทำหน้าที่ควบคุมระบบทั้งหมดไม่ว่าจะเป็นการสั่งงาน การเชื่อมต่อสายต่าง ๆ การจัดส่งสัญญาณต่าง ๆ สัญญาณควบคุม ตลอดจนถึงการตรวจเช็คสัญญาณต่าง ๆ ของระบบทั้งหมด

- DISPLAY

DISPLAY เป็นตัวทำการ INTERFACE กับผู้ใช้ โดยจะทำให้ผู้ใช้ทราบว่าสถานะของเครื่องอยู่ในสภาวะใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีระบบชุมสาย PABX

2.1 ทฤษฎีระบบชุมสาย PABX

SUBSCRIBER LINE INTERFACE CIRCUIT (SLIC) เป็นอุปกรณ์ที่ใช้ติดต่อกับเครื่องโทรศัพท์ ซึ่งทำหน้าที่รับสัญญาณที่ติดต่อระหว่างเครื่องโทรศัพท์กับอุปกรณ์ SWITCHING ซึ่งสัญญาณนี้เรียกว่า SUBSCRIBER SIGNALLING จะประกอบด้วยสัญญาณต่อไปนี้

- **สัญญาณที่ส่งจากเครื่องโทรศัพท์ไปยังอุปกรณ์ SWITCHING**

- **ON HOOK** หมายถึง สภาพวางหูหรือสภาพว่าง (IDLE) ลักษณะของวงจร OPEN LOOP HIGH IMPEDANCE

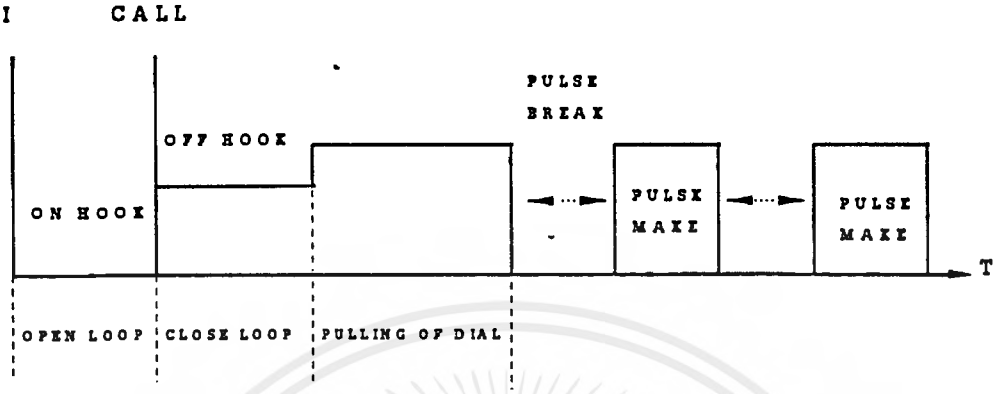
- **OFF HOOK** สภาพผู้เช่ายกหู (BUSY) สายจะมีสภาพ CLOSED LOOP LOW IMPEDANCE

- **DIALLING** ถ้าเครื่องโทรศัพท์เป็นแบบ ROTARY DIAL สัญญาณจะเป็นแบบ PULSING ค่า IMPEDANCE จะสูงต่ำสลับกันตามโค้ดที่หมุนตามรูป แต่ในเครื่องระบบกดปุ่มจะมีสัญญาณ DTMF ส่งออกไป

สัญญาณ DTMF (DIAL TONE MULTI FREQUENCY) เป็นสัญญาณโค้ดเลขหมายของผู้ที่เรียกส่งในรูปสัญญาณ 2 ความถี่ ซึ่งให้ข้อดีในแง่ลดเวลาในการส่งเลขหมายลงเหลือ 0.7 วินาที/เลขหมาย เมื่อเทียบกับระบบ PULSE ความผิดพลาดมีโอกาสน้อยมาก เพราะมีการตรวจสอบความถี่ 2 ชุด พร้อมกัน นอกจากนั้นยังใช้โค้ดได้มากกว่า เช่น เพิ่มโค้ด * (START) และ # (SQUARE) ซึ่งใช้ในงานบริการพิเศษ

การกำเนิดความถี่ที่ใช้ใน DTMF แบ่งเป็น 2 กลุ่ม คือ กลุ่มความถี่สูงตั้งแต่ความถี่ 1209 HZ และกลุ่มความถี่ต่ำตั้งแต่ 492 HZ ลงมา ดังแสดงในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 ROTARY DIAL

FREQUENCY Hz		
BUTION	GROUP 1	GROUP 2
1	697	1209
2	697	1336
3	697	1447
4	770	1209
5	770	1336
6	770	1209
7	852	1336
9	852	1447
*	941	1209
0	941	1336
#	941	1447

ตารางที่ 2.1 ตารางแสดงความถี่ DTMF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 สัญญาณที่ส่งมาจากอุปกรณ์ SWITCHING

- DIALLING TONE เป็นสัญญาณที่บอกให้ทราบว่าขณะนี้อุปกรณ์ SWITCHING พร้อมที่จะรับโค้ดการหมุนเลขหมายจากผู้เรียกให้ผู้เรียกทำการส่งเลขหมายได้สัญญาณ DIAL TONE นี้เป็นสัญญาณต่อเนื่องความถี่ 400 Hz MODULATE ด้วย 50 Hz

- BUSY TONE เป็นสัญญาณที่ส่งมาบอกให้ทราบว่าอุปกรณ์ไม่ว่าง เช่น ถ้ายกหูเครื่องโทรศัพท์แล้วได้ยินเสียงนี้ (แทนที่จะได้ยิน DIAL TONE) แสดงว่าอุปกรณ์ SWITCHING ไม่ว่างพร้อมที่จะรับเลขหมายที่หมุน แต่ถ้าได้ยิน BUSY TONE หลังจากหมุนเลขหมายครบแล้วแสดงว่าผู้ถูกเรียก (B SUBSCRIBER) ไม่ว่างหรืออุปกรณ์ SWITCHING ที่ต่อออกไปไม่ว่าง สัญญาณ BUSY TONE เป็นสัญญาณ SINE WAVE ความถี่ 400 Hz ส่งเป็นช่วง ๆ ส่ง 0.5 วินาที หยุด 0.5 วินาที

- RINGING TONE เป็นสัญญาณที่ผู้เรียก (A SUBSCRIBER) ได้ยินหลังจากหมุนเลขหมายครบแล้ว เพื่อบอกให้ทราบว่าการต่อสำเร็จ โดยขณะนี้อุปกรณ์ได้ส่งสัญญาณเรียก (RINGING SIGNAL) ไปยังผู้ถูกเรียก (B SUBSCRIBER) โดยใช้สัญญาณ SINE WAVE ความถี่ 400 Hz โดยส่ง 1 วินาที หยุด 4 วินาที

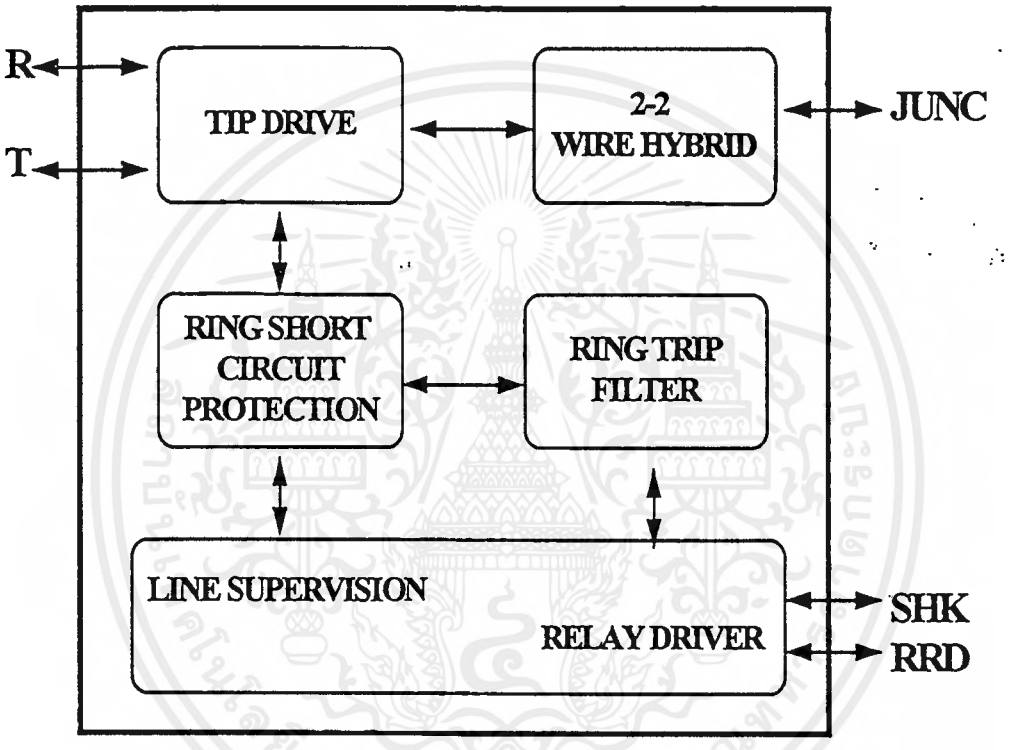
- RINGING SINGAL เป็นสัญญาณที่ส่งไปยังผู้ถูกเรียกซึ่งจะทำให้กระดิ่งดัง ซึ่งใช้สัญญาณ SINE WAVE ความถี่ 25 Hz ค่าแรงดัน 70-90 V_{rms} ช่วงการส่ง ส่ง 1 วินาที หยุด 4 วินาที

- NU TONE (NUMBER UNOBTAINABLE TONE) เป็นสัญญาณที่ใช้บอกให้ทราบว่าเลขหมายที่เรียกไม่มีใช้อยู่หรือเลขหมายนี้ได้เปลี่ยนแปลง ไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

❑ โครงสร้างของ SUBSCRIBER LINE INTERFACE CIRCUIT (SLIC)

โครงสร้างของ SLIC จะประกอบด้วยวงจรที่ใช้ติดต่อระหว่างเครื่องโทรศัพท์กับอุปกรณ์ทางด้าน SPEECH PATH และ CONTROL UNIT โดยสามารถรับ-ส่งสัญญาณ LINE SIGNALLING ติดต่อกันได้ซึ่งต้องประกอบวงจร LINE SUPERVISION 1 วงจร PROTECTION ใน SLIC จะต้องมีค่า LINE IMPEDANCE (Zi) ที่เหมาะสมกับเครื่องโทรศัพท์ และ OUTPUT IMPEDANCE (Zo) ซึ่งเหมาะสมวงจร SPEECH PATH โครงสร้างของ SLIC ประกอบไปด้วย BLOCK DIAGRAM ดังรูป 1



รูปที่ 2.2 LINE INTERFACE UNIT

TIP DRIVE ทำหน้าที่เป็น I/P รับสัญญาณ ANALOG ซึ่งสัญญาณ ANALOG นี้จะประกอบไปด้วยสัญญาณเสียงกับไฟ DC แล้วทำการเปลี่ยนให้เหลือเพียงสัญญาณเสียงส่งไปวงจร HYBRID ในวงจร TIP DRIVE ยังเป็นวงจรที่ทำหน้าที่ BATTERY FEED จ่ายกระแสไฟให้กับ LOOP ของ TIPRING เพื่อนำไปใช้จ่ายกระแสให้เครื่องโทรศัพท์

RING SHORT CIRCUIT PROTECTION เป็นวงจรป้องกันการ SHORT CIRCUIT ระหว่างสาย TIP-RING กรณีไม่มีเครื่องโทรศัพท์อยู่ เพื่อป้องกันไม่ให้วงจร TIP DRIVE เสียหายจาก SHORT CIRCUIT ระหว่าง TIP-RING และยังมีหน้าที่พิเศษอีกกรณีที่อุปกรณ์ SWITCHING ส่ง RINGING CURRENT ไปแล้วผู้ถูกเรียกยกหูทำให้มีกระแสไฟผ่านสูงเกินอัตราที่ทนได้ของวงจร TIP DRIVE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2-2 WIRE HYBRID เป็นวงจรที่ทำหน้าที่ขยายสัญญาณแบบสองทิศทาง โดยแยกเป็น TX และ RX แล้ว OUTPUT ก็นำมาเข้าวงจร HYBRID ทำให้เป็น 2 WIRE เพื่อที่จะส่งออกไปยัง JUNCTOR เพื่อส่งต่อไปยัง SPEECH PATH ต่อไป

RING TIP FILTER ทำหน้าที่ตรวจสอบว่า TIP-RING เป็น OPEN LOOP หรือ CLOSED LOOP แล้วส่งสัญญาณที่ได้ไปยัง LINE SUPERVISION

LINE SUPERVISION ทำหน้าที่สร้างสัญญาณ OFF/ON HOOK และ LINE STATUS เพื่อให้อุปกรณ์ CONTROL UNIT ทราบว่ามีการยกหู, วางหู

RELAY DRIVER เป็นวงจรที่ออกแบบไว้ทำการขับตัว RELAY เพื่อใช้ในการตัดต่อ RINGING CURRENT หรือไว้ทำการต่ออุปกรณ์การ TEST LINE เพื่อไว้ตรวจสอบต่ออุปกรณ์ปลายทาง

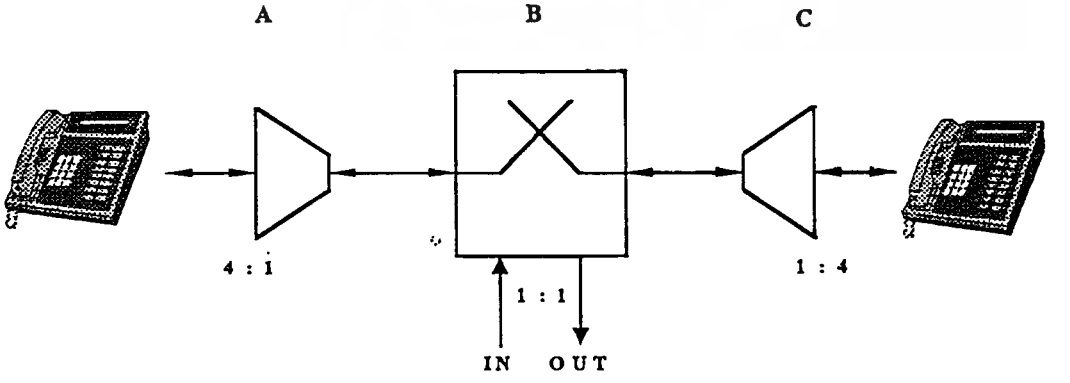
❑ SWITCHING UNIT

SWITCHING UNIT แบ่งเป็นส่วนประกอบได้เป็น 2 ส่วน คือ

1. SPEECH PATH
2. CONTROL PATH

1. SPEECH PATH

ระบบ SPEECH PATH จะประกอบด้วยภาคตัดต่อหลาย ๆ ภาค แต่ในโครงงานนี้จะใช้ภาคตัดต่อเป็นแบบ ELECTRONIC CROSS POINT ซึ่งควบคุมโดยสัญญาณ DIGITAL โดยทั่วไป SPEECH PATH จะเป็นตัวกำหนดอัตราส่วนของเครื่องต่ออัตราส่วนของอุปกรณ์ SWITCHING ซึ่งเรียกว่า CONCENTRATION RATIO ในระบบ SWITCHING จะกำหนดค่า CONCENTRATION RATIO ว่ามีค่ามากน้อยเท่าไรขึ้นอยู่กับว่า TRAFFIC ในการใช้งานมีมากเท่าไร CONCENTRATION ตามมาตรฐาน CCITT มีค่า 2:1, 4:1, 8:1



รูปที่ 2.3 SPEECH PATH DIAGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- จาก BLOCK DIAGRAM A เป็น CONCENTRATOR มีขนาด 4:1 หมายถึง เครื่องโทรศัพท์ 4 เครื่อง ใช้อุปกรณ์ SWITCHING ได้เพียง 1 ช่อง ใน BLOCK DIAGRAM A เรียกว่า CONCENTRATOR

- BLOCK DIAGRAM B เป็น DISTRIBUTION มีอัตรา CONCENTRATION RATIO เท่า 1:1 โดยจะไม่มีการจำกัดการ CALL

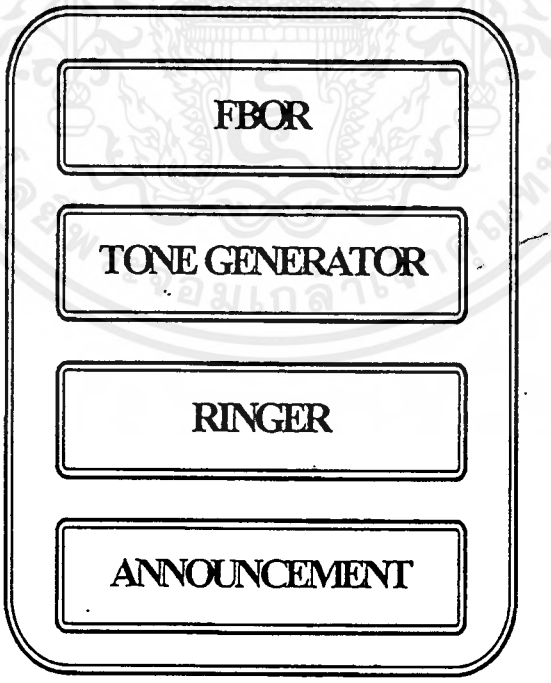
SUPERVISORY เป็นอุปกรณ์ที่รับส่งสัญญาณของระบบ PABX ซึ่งประกอบด้วยอุปกรณ์ดังนี้

1. PBOR (PUSHBUTON REGISTER) เป็นอุปกรณ์ที่ใช้รับสัญญาณ DIMF จากเครื่องโทรศัพท์แปลงสัญญาณ DIGITAL เพื่อนำไปใช้ต่อไป

2. TONE GENERATOR เป็นอุปกรณ์ผลิตสัญญาณสำหรับบอกให้ทราบว่าอุปกรณ์ที่ติดต่อมีสภาวะอย่างไร

3. RINGER ใช้ผลิตสัญญาณ RINGING CURRENT เพื่อส่งไปบอกอุปกรณ์ปลายทางให้รู้ว่ากำลังถูกติดต่อ

4. ANNOUNCEMENT เป็นอุปกรณ์ AUTO RECEIVER เพื่อบอกให้ผู้ติดต่อทราบว่าการติดต่อควรจะกระทำอย่างไร หรือบอกให้ทราบสภาวะของเครื่องให้ทราบ



รูปที่ 2.4 SUPERVISORY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. CONTROL PATH

DIGITAL SWITCHING SYSTEM ถูกควบคุมด้วย CONTROL PATH ที่ทำหน้าที่ควบคุมการทำงานของระบบทั้งหมด ไม่ว่าจะเป็นการเรียกออก การเรียกเข้า การต่อเลขหมายภายใน ฯลฯ ซึ่งประกอบด้วย

- CPU
- INTERFACE UNIT

CPU ในระบบชุมสายโทรศัพท์ PABX หรือระบบอื่นที่ทำงานแบบ AUTOMATIC มีความจำเป็นต้องใช้ CPU ในการสั่งงานและควบคุมงานต่าง ๆ ของระบบชุมสายทั้งหมด ดังนั้น CPU จึงเป็นสิ่งที่สำคัญมากที่สุดของระบบ โดยในระบบชุมสายนี้ได้เลือกใช้ CPU แบบ SINGLE CHIP MICROPROCESSOR เนื่องจากชุมสายที่ออกแบบเป็นขนาดเล็ก ใช้ CPU เพียงตัวเดียวก็เพียงพอต่อปริมาณงานต่าง ๆ แล้ว รูปแบบการใช้งานของ CPU แสดงใน BLOCK DIAGRAM ของระบบชุมสาย

CPU ใช้ควบคุมการทำงาน ELETRONICS SWITCHING UNIT, SLIC, TOT LIC ฯลฯ โดยจะทำงานตาม SOFTWARE ที่เขียนขึ้น และบรรจุอยู่ในระบบ MEMORY รวมทั้ง CPU ยังทำหน้าที่รับคำสั่งการต่อเลขหมายจาก SLIC และ LIC

INTERFACE UNIT เป็นอุปกรณ์ที่ทำหน้าที่เป็น INPUT/OUTPUT PORT สำหรับใช้ในการเขียนและอ่าน DATA จาก CPU ไปยังอุปกรณ์ HARDWARE รอบข้าง เช่น ELECTRONICS SWITCHING UNIT, SLIC, TOT LIC ฯลฯ เนื่องจาก CPU ไม่สามารถติดต่อกับอุปกรณ์ HARDWARE ต่าง ๆ ได้โดยตรง จึงต้องเชื่อมต่อผ่านอุปกรณ์ INTERFACE ที่เป็น INPUT/OUTPUT PORT ซึ่งการเขียนและอ่านข้อมูลจาก CPU ไปยัง INPUT/OUTPUT PORT ต้องมีการติดต่อไปยัง ADDRESS ที่ต้องการ

SOFTWARE

ชุมสาย DIGITAL SWITCHING SYSTEM ที่สร้างขึ้นนี้ใช้ SOFTWARE แบบ STORED PROGRAM CONTROL ในการสั่งงานจากการประมวลผลจากหน่วยความจำ และทำงานเป็นลำดับตามที่ส่วน CONTROL PATH ได้กำหนดไว้แล้ว การเพิ่มและปรับเปลี่ยนหน้าที่ต่าง ๆ สามารถที่จะกระทำได้ง่าย เพื่อให้มีความคล่องตัวสูง

CONTROL PATH จะทำงาน PROGRAM ที่อ่านข้อมูลเข้ามาอย่างต่อเนื่องจากหน่วยความจำ และควบคุมการทำงานของวงจรลอจิกต่าง ๆ ให้เป็นไปตาม PROGRAM ที่กำหนดไว้ โดยจะทำงานที่ความเร็วสูงมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบ *SOFTWARE* พื้นฐานของชุมสายโทรศัพท์

HARDWARE เป็นส่วนของอุปกรณ์หลาย ๆ อย่างที่สร้างเป็นระบบชุมสายขึ้นมา ลักษณะของ HARDWARE จะเป็นอุปกรณ์ที่ประกอบด้วย SPEECH PATH, CONTROL PATH ฯลฯ ส่วนของ SOFTWARE จะเป็นการกำหนดวิธีการทำงาน และประยุกต์วิธีการทำงานของอุปกรณ์ที่ใช้ในระบบชุมสาย เพื่อที่จะทำให้ชุมสายให้การบริการที่ดี

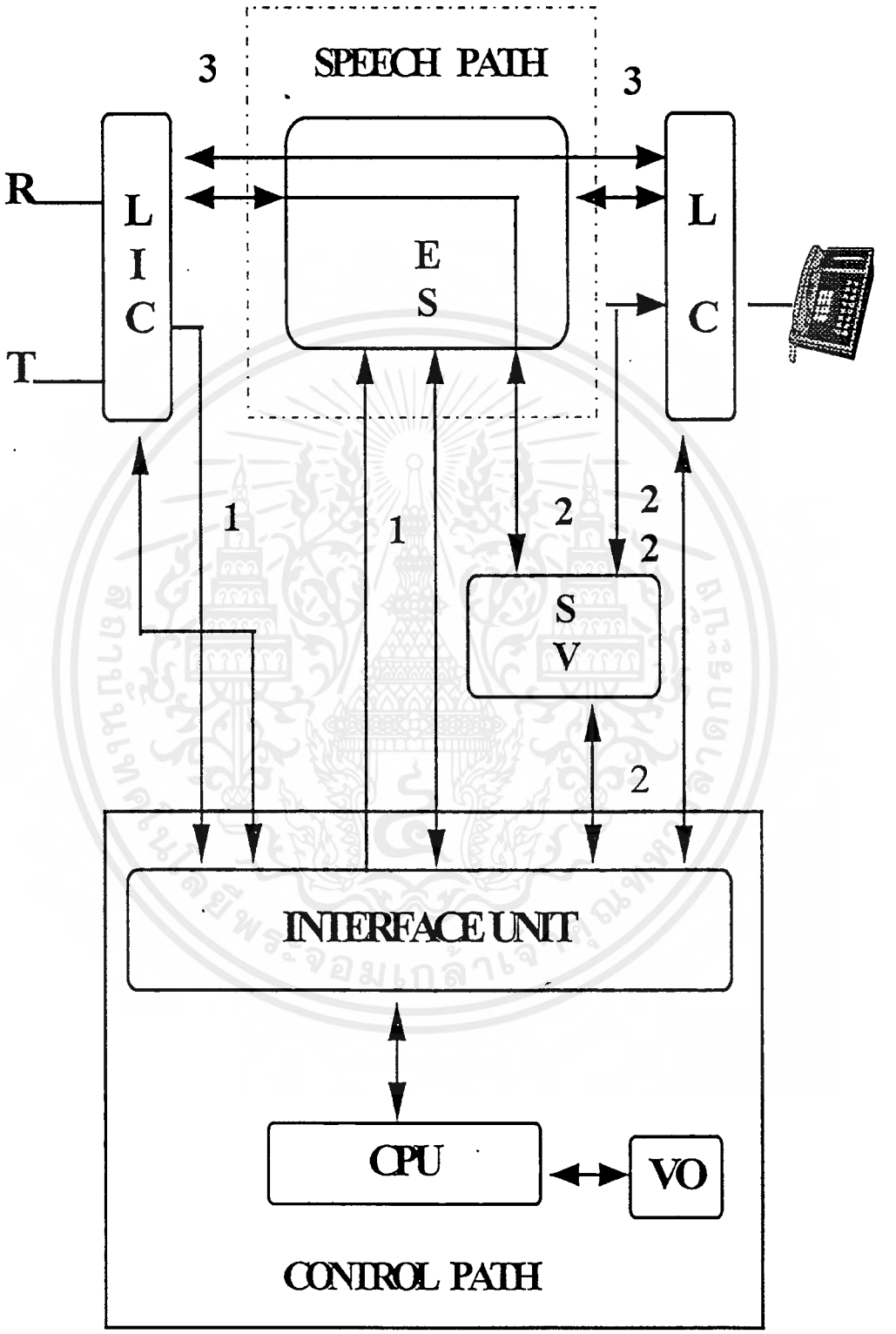
SOFTWARE เป็นกรรมวิธีในการทำงานของระบบชุมสาย โดยสามารถควบคุมการทำงานได้โดยทางตรงและโดยทางอ้อม ซึ่งจะมีส่วนของโปรแกรมและข้อมูลที่ถูกเก็บไว้ในหน่วยความจำในรูปแบบของสัญญาณไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใด ๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 BLOCK DIAGRAM

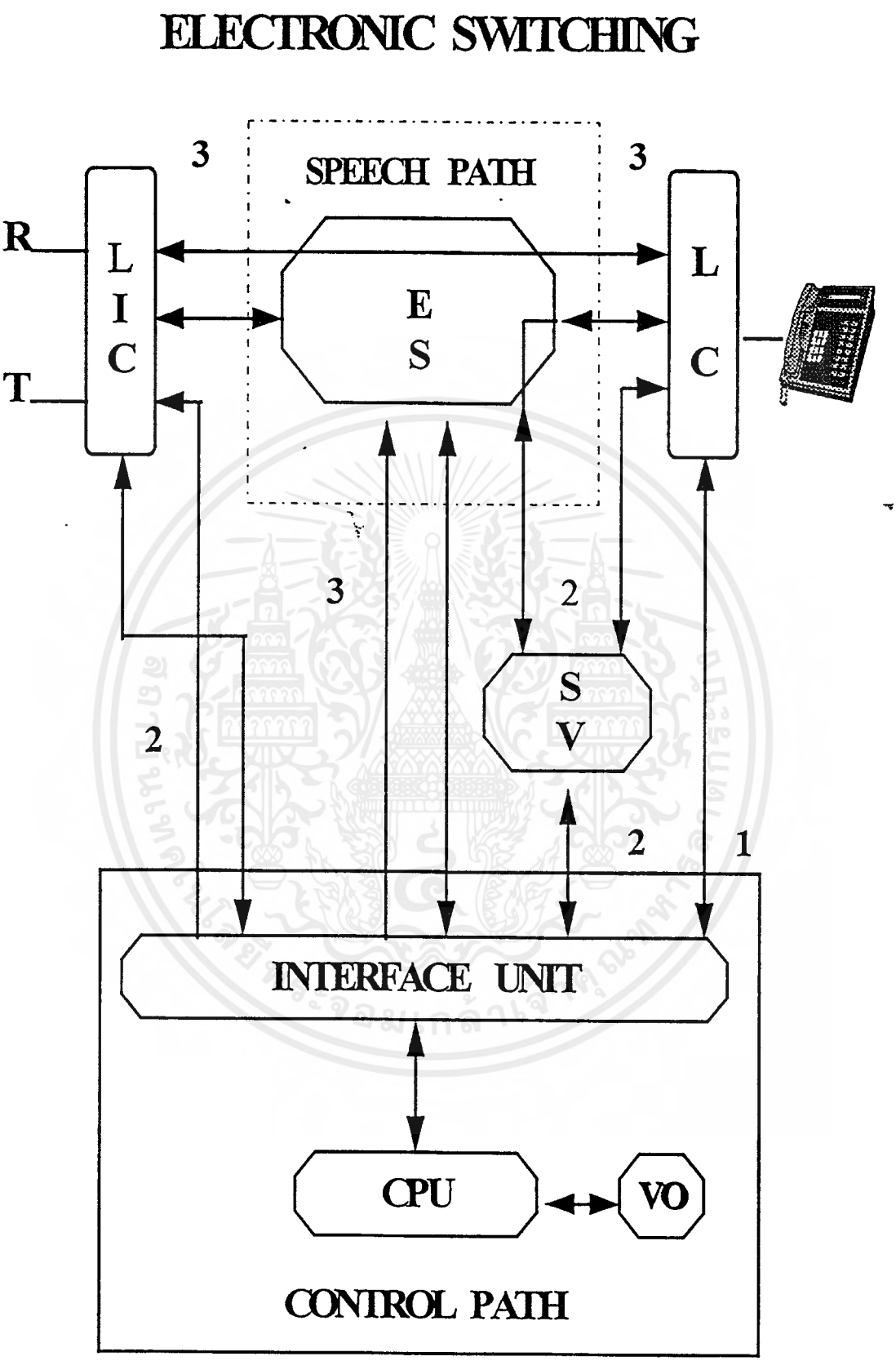
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRONIC SWITCHING



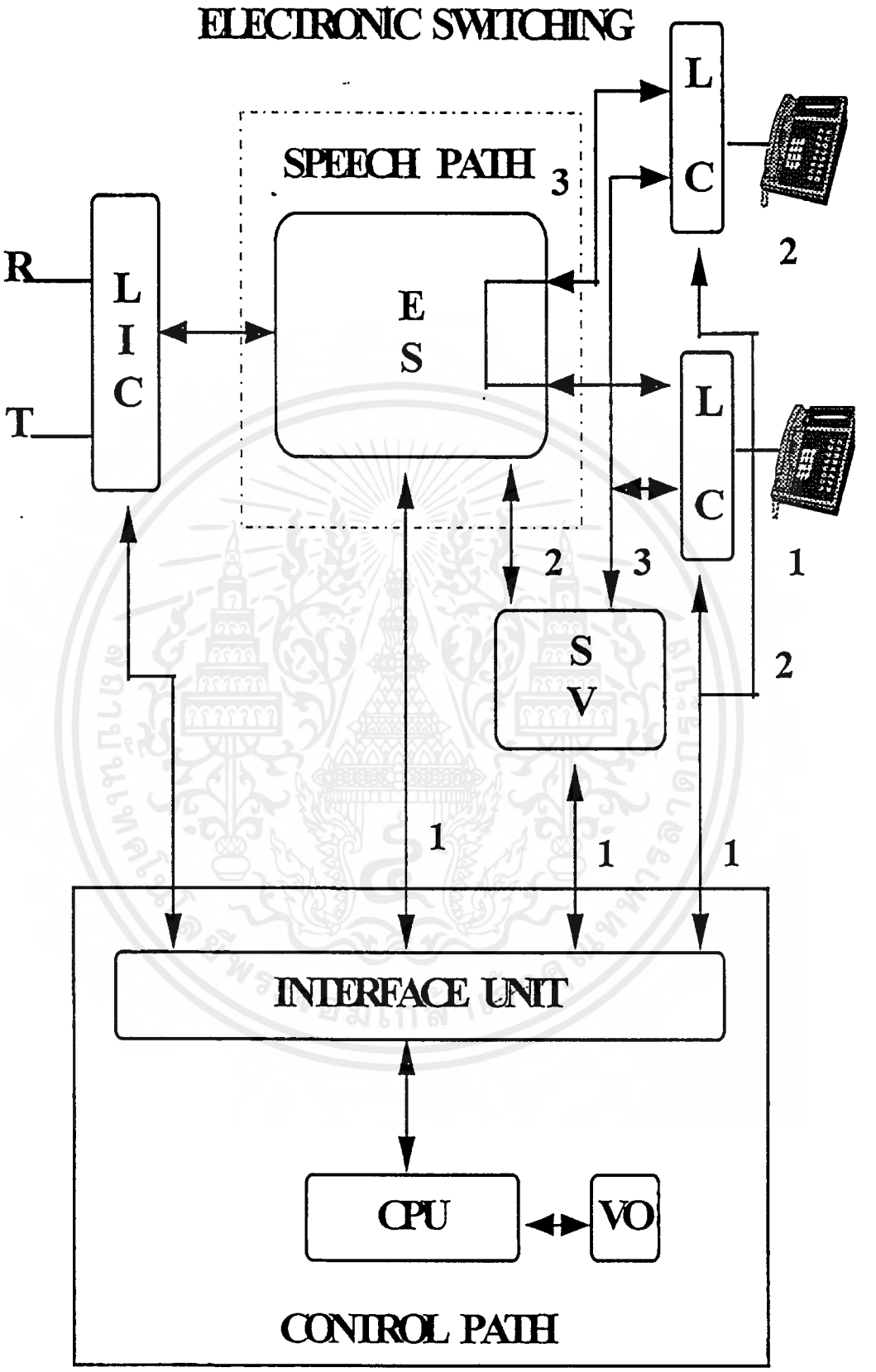
รูปที่ 2.5 TERMINATE CALL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



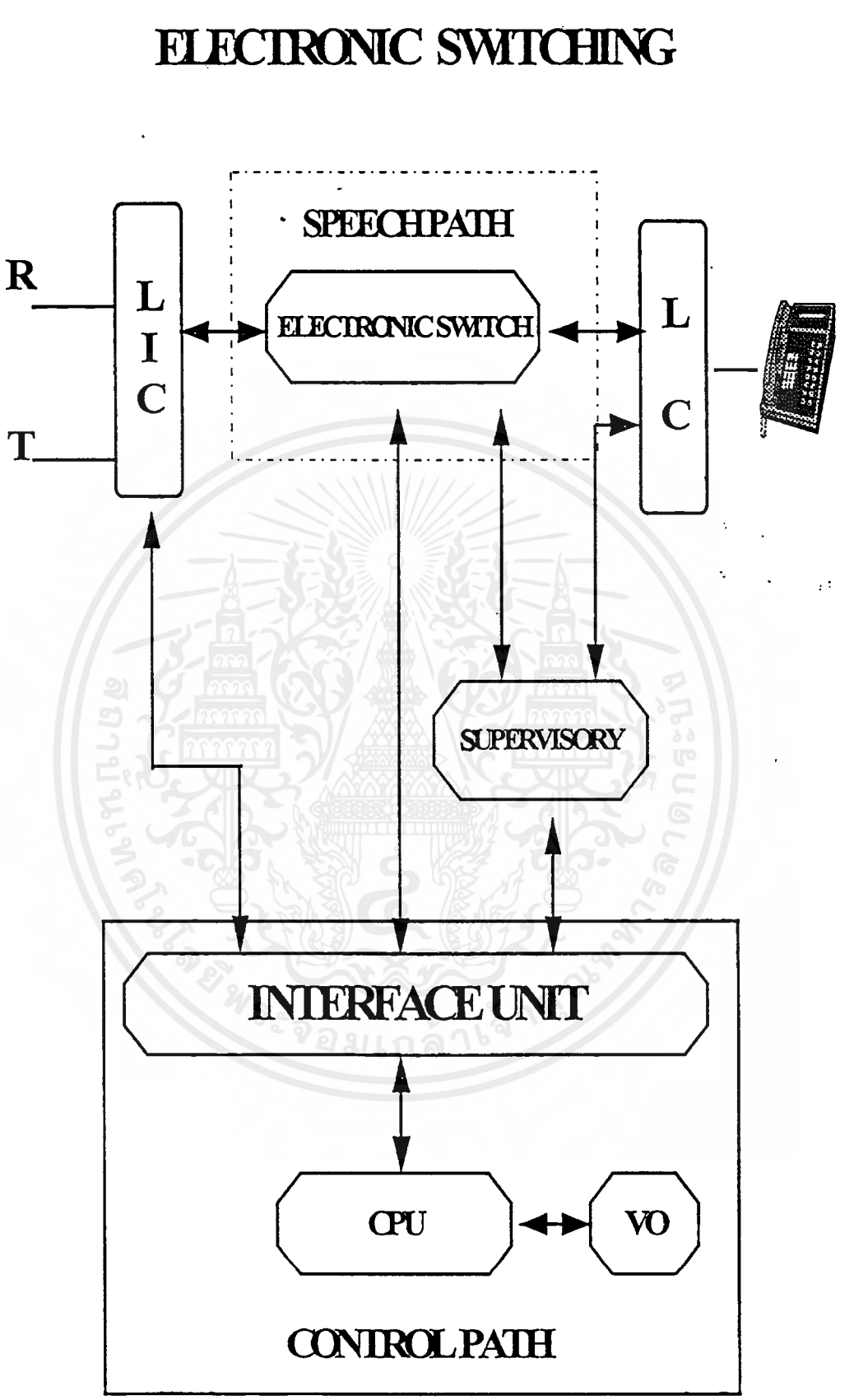
รูปที่ 2.6 ORIGINATE CALL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 27 INTRA CALL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 BLOCK DIAGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- *TERMINATED CALL เป็นการเรียกจาก SUB A ซึ่งเป็น LINE ภายนอกเข้ามาหา PABX*

- โดยทางชุมสายโทรศัพท์ส่งสัญญาณ RINGING ทางสาย RING & LIC ตรวจสอบว่ามีการส่ง RINGING เข้าก็จะ GENERATE สัญญาณส่งไปให้ CPU รับทราบ สัญญาณหมายเลข 1 CPU ก็จะสั่ง LIC CONNECT EXTERNAL LINE สั่งให้ ES ทำการต่อ CROSS POING ที่ SPEECH PATH

- CPU สั่งให้ SV ทำการต่อ ANNOUNCEMENT MACHINE เพื่อบอกให้ทราบว่า ขณะเครื่องได้ทำการรับสายเรียบร้อยแล้วรอรับสัญญาณ DTMF เพื่อทำการ ANALYSIS DIGIT แล้วทำการตรวจสอบดูว่าเครื่องที่ถูกเรียก (SUB B) ว่างหรือไม่ว่า ถ้าไม่ว่าก็จะสั่งให้ SV ส่ง BUSY TONE ให้ SUB A รอระยะเวลาหนึ่งแล้วทำการ DISCONNECTION EXTERNAL LINE แต่ถ้า SUB B ว่าง ก็จะส่ง RINGING ให้ SUB B แล้วจอง SPEECH PATH ไว้ ถ้า SUB B ยกหู LC ก็จะส่งสัญญาณให้ CPU ทราบแล้วทำการ DISCONNECTION RINGING

- ทำการต่อ SPEECH PATH แล้วตรวจสอบดูว่า SUB B วางหูหรือไม่ ถ้า SUB B ว่างจะทำการ DISCONNECTON SPEECH PATH เพื่อให้ SPEECH PATH ว่างต่อไปตามรูปที่ 2.5

- *ORIGINATE CALL การติดต่อจากสายภายในไปสายภายนอก*

- เมื่อ SUB A ยกหู (OFF HOOK) LC ก็จะทำการส่งสัญญาณไปยัง CPU CPU จะทำการตรวจสอบว่าเป็นเลขหมายอะไร ชนิดอะไร เมื่อตรวจสอบดูแล้วว่าถูกต้อง CPU ก็ตรวจสอบว่า SV ว่างหรือไม่ เพื่อให้บริการต่อเลขหมายต่อไป

- เมื่อทำการตรวจสอบเสร็จแล้วก็จะสั่งให้ ES คือ SPEECH PATH เข้ากับ SV ต่อเข้ากับ LC แล้วให้สัญญาณ BUSY TONE กรณี SV ไม่ว่างหรือให้ DIAL TONE เมื่อกรณี SV ว่าง เมื่อ SUB A ได้ DIAL TONE แล้วอุปกรณ์ SV จะรอรับเลขหมายที่จะทำการติดต่อจาก SUB A ว่าเป็นการต่อออกแบบ ORIGINATE CALL หรือ INTRA CALL ส่งให้ CPU ทำการวิเคราะห์เมื่อวิเคราะห์แล้วก็จะทำการตรวจสอบว่า LIC ว่างหรือไม่เพื่อทำการจอง EXTERNAL LINE ไว้

- เมื่อตรวจสอบอุปกรณ์ LIC และ ES ว่าว่างแล้ว CPU ก็จะทำการสั่งให้ ES ต่อกับ LIC ต่อเข้ากับ SPEECH PATH และต่อเข้ากับ LC แล้วทำการ MAKE BUSY ต่ออุปกรณ์ LIC ES LC ตำแหน่งที่ถูกใช้ต่อไปการทำงานอธิบายตามรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- *INTRA CALL การติดต่อภายใน PABX*

- เมื่อ SUB A ยกหู (OFF HOOK) LC ก็จะทำการส่งสัญญาณไปยัง CPU CPU จะทำการตรวจสอบว่าเป็นเลขหมายอะไร ชนิดอะไร เมื่อตรวจสอบดูแล้วว่าถูกต้อง CPU ก็จะตรวจสอบว่า SV ว่างหรือไม่เพื่อให้บริการเลขหมายต่อไป เมื่อตรวจสอบเสร็จเรียบร้อยแล้ว CPU ก็จะสั่งต่อ ES เข้ากับ SV

-เมื่อทำการต่อ ES เข้ากับ SV แล้ว CPU จะสั่งต่อ SPEECH PATH เข้ากับ LC แล้วให้สัญญาณ BUSY TONE กรณี SV จะรอรับเลขหมายที่จะทำการติดต่อจาก SUB A ว่าเป็นการต่อออกแบบ ORIGINATE CALL หรือแบบ INTRA CALL สั่งให้ CPU ทำการวิเคราะห์เมื่อวิเคราะห์แล้วว่าเป็นการเรียกแบบ INTRA CALL CPU จะทำการตรวจสอบว่าเลขหมายที่ถูกเรียก (SUB B) ว่างหรือไม่

-เมื่อ CPU ตรวจสอบดูแล้วว่า SUB B ว่าง CPU จะสั่งให้ SV ส่ง RINGING TONE ให้กับ SUB B แล้วสั่งให้ SV ต่อสัญญาณ RING BACK TONE ให้กับผู้เรียก (SUB A) เพื่อให้ SUB A รู้ว่าขณะนี้กำลังต่อกับเลขหมายที่ติดต่ออยู่ เมื่อ SUB B ยกหู (OFF HOOK) CPU จะสั่งยกเลิกการส่งสัญญาณ RINGING ให้กับ SUB B และยกเลิกการส่งสัญญาณ RING BACK TONE ให้กับ SUB A แล้วทำการต่อ SPEECH PATH ให้กับ SUB A และ SUB B แล้วส่งยกเลิกการติดต่อ SV ก็เสร็จสิ้นการติดต่อการทำงานของระบบเป็นไปตามรูป 2.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

การทำงานของวงจร

3.1 วงจร INTERFACE (รูปที่ 3.1)

ทำหน้าที่ติดต่อระหว่าง CPU กับภายนอก โดยจะทำหน้าที่ควบคุมการทำงานของ Port ต่าง ๆ การ Decode รหัส การ Interrupt โดยในส่วนของการ Decode Port จะใช้ IC 74LS138 ในการถอดรหัส โดยจะนำสัญญาณ A2, A3 และ A4 มาเข้าสัญญาณ A B และ C ของ 74LS138 โดยขา MREQ จะต่อเข้าสัญญาณ G2A และสัญญาณ IORQ จะต่อเข้าสัญญาณ G2B ของ 74LS138 และสัญญาณ G1 จะนำมาจากการเอา Address A5-A7 มา NOR กันโดย output ของ 74LS138 จะมี Y0-Y7 ในการ Select port โดยมี Port 1 เป็น Interface port 2 เป็น Sub 1 port 3 เป็น Sub 2 , Port 4 เป็น TOT 1, Port 5 เป็น TOT 2, Port 6 เป็น Switch port 7 เป็น Switch 2 และ Port 8 เป็น Swich 3 และในส่วนของการ Interrupt เราใช้ D-Flip flop 8 ตัวในการ latch สัญญาณ Interrupt โดยขา Q ของแต่ละตัวจะต่อเข้า port A ของ 8255 และขา Clear จะต่อเข้า port B ของ 8255 และ Q จะนำมา Nor กันด้วย 4078 และเข้า Interrupt 1 โดยเมื่อมีสัญญาณ Interrupt เข้ามาจะทำให้ out put ของ 4078 logic "0" ในการ Interrupt CPU และ CPU จะทำการ Check ที่ Port A ว่ามีสัญญาณใดเข้ามา และทำการ Clear ที่ Port B

3.2 วงจร TOT LINE INTERFACE CIRCUIT (รูปที่ 3.2)

เมื่อมีสัญญาณ Ringing เข้ามาที่จะผ่านการ Detect โดยอุปกรณ์ OPTO-COUPLE ซึ่งจะเปลี่ยนสัญญาณ Analog ให้เป็นสัญญาณ Digital โดยใช้ค่า R 5 เท่ากับ 5 Kω และ C3 เท่ากับ 100 UF และนำสัญญาณไปขับ Transistor BC 546 จากนั้น out put ที่ขา C จะเป็น Clock ไปที่ขา 1 IC 74LS393 และ output ของ 74LS393 จะต่อเข้า IC 4078 เพื่อนำไปเป็นสัญญาณ Interrupt TOT และ output 74LS393 ต่อเข้ากับ Port/output ของ IC 8255 เพื่อ check จำนวน Clock ที่เข้ามา

เมื่อ CPU รับรู้ว่ามี Ringing เข้ามา จึงส่งสัญญาณ Control มา เพื่อให้ Relay ทำการต่อสายนอกเข้ากับ PABX โดยผ่านหม้อแปลง Couple และ output ของหม้อแปลง Couple จะถูกต่อไปยัง Switching part และ Detect Busy

สัญญาณ Control ของ CPU ก็จะทำการ Clear memory ของ 74LS393 เพื่อรอรับสัญญาณ Ringing จากผู้เรียกอีก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ 038340 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 วงจร DETECT BUSY (รูปที่ 3.3)

สัญญาณจาก TOT Line จะถูก Band pass filter กรองความถี่ในช่วง 350-450 Hz จากนั้นมาผ่านวงจร Comparator ได้ out put เป็น Clock มาเข้า IC 74123 ซึ่ง IC 74123 จะให้ output เป็น clock pulse ขนาดเล็กมาก เกิดขึ้นในขอบขาลง จากนั้น output ของ 74123 จะถูก OR กับสัญญาณ Input ที่เข้า IC จากนั้นสัญญาณส่วนที่หนึ่งจะเข้า Schmitliger เพื่อสร้างขอบขาขึ้นและขาลงแล้วนำสัญญาณทั้งสองมาก OR กัน นำไปเข้าขา clock ของ IC 74LS377 เพื่อเป็นสัญญาณ Latch สัญญาณส่วนที่สองเข้า R-S latch โดยที่ output Q ต่อเข้ากับวงจร Monostable สร้าง clock 256 Hz จ่ายให้กับ IC 74LS161 และ output ของ 74LS161 จะถูกนำมาต่อเข้ากับ IC 74LS377 (ขา D0-D7) output ของ 74LS377 จะต่อเข้า R-2R ladder เพื่อเปลี่ยน digital เป็น analog และนำแรงดัน output ผ่าน windows op-amp จึงได้ out put busy

3.4 วงจร Switching part (รูปที่ 3.4.1, 3.4.2, 3.4.3,)

การทำงานของวงจร Switching จะประกอบด้วยส่วนสำคัญคือ Array Switch โดยเลือกใช้เบอร์ MT 8816 2 ตัวรวมกัน ซึ่งในแต่ละตัวจะมีขนาดเท่ากับ 8*16 เมื่อนำ 2 ตัวรวมเข้าด้วยกันจะมีขนาด 16*16 โดยจะทำการ ON และ OFF โดยผ่าน port ของ 8255 โดย port A จะเป็น port ทำการ OFF Switch โดยทำการส่ง Data ของ Switch ที่ต้องการมายัง port A ในการ ON-Switch จะทำการ ON ผ่าน port B โดยทำเช่นเดียวกับ Port A โดยอาจจะนำเอา A0, A1 และ Ship select ของ 8255 ที่ใช้งานมาทำการ Detect ว่าเป็นการ Read หรือ Write ซึ่งจะมี IC เบอร์ 74LS244 2 ตัว ทำการ latch สัญญาณ โดยจะนำเอาสัญญาณ junction จาก subscriber ทั้ง 16 เบอร์ต่อเข้าทางด้าน X0-X15 และทางด้าน Y จะประกอบไปด้วย chanel กลาง 8 chanel และ chanel ของ TOT 3 chanel โดย chanel กลางและ chanel ของ TOT จะมีตัว detect สัญญาณ DTMF ต่ออยู่ทุก ๆ chanel ทั้งหมด 11 chanel และจะมีสัญญาณ dial, ringback, voice ต่อเข้าทางด้าน Y ด้วย โดยใน DTMF ทั้ง 11 chanel จะต่อสัญญาณ DATA เข้าที่ 8255 อีก 2 ตัว

3.5 Signal

3.5.1 Dial tone (รูปที่ 3.5.1)

คือ สัญญาณ Sine wave 400 Hz mod กับสัญญาณ 50 Hz ซึ่งสร้างได้โดยใช้ IC 555 2 ตัว โดย IC 555 ตัวที่ 1 จะทำการสร้างสัญญาณ 50 Hz IC 555 ตัวที่ 2 สร้างสัญญาณ 400 Hz

ขา 3 IC 555 ตัวที่ 1 จะต่อเข้ากับ IC 555 ตัวที่ 2 และ output IC 555 ตัวที่ 2 จะถูกต่อกับ Band pass filter ก็จะได้สัญญาณ Dial tone

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2 Busy tone (รูปที่ 3.5.2)

คือ สัญญาณ sine wave 400 Hz ติด 0.5 วินาที หยุด 0.5 วินาที

สร้างโดยใช้ IC 555 2 ตัวโดย IC 555 ตัวที่ 1 สร้างความถี่ 1 Hz (ติด 0.5 วินาที ดับ 0.5 วินาที) นำ output ขา 3 มาต่อกับขา 4 IC 555 ตัวที่ 2 ซึ่ง IC 555 ตัวที่ 2 สร้างความถี่ 400 Hz นำ output ขา 3 ต่อเข้ากับ Band Pass filter ทำให้ได้ output คือสัญญาณ Busy tone

3.5.3 Ringback tone (รูปที่ 3.5.3)

คือ สัญญาณ sine wave 400 Hz ติด 1 วินาที หยุด 4 วินาที

สร้างโดยใช้ IC 555 ตัวที่ 1 สร้งความถี่ติด 1 หยุด 4 และนำ output ขา 3 ไปต่อขา 4 ของ IC 555 ตัวที่ 2 ซึ่ง IC 555 ตัวที่ 2 ผลิตความถี่ 400 Hz และ output (ขา 3) ผ่านวงจร band pass filter จึงได้สัญญาณ Ringback tone

3.5.4 Ringing signal (รูปที่ 3.5.4)

คือ สัญญาณ sine wave 25 Hz ติด 1 วินาที หยุด 4 วินาที โดยใช้ค่าแรงดัน 140 V โดยใช้ IC 555 ผลิตความถี่ 25 Hz โดยควบคุมการ ON-Off จาก IC 555 อีกตัวหนึ่งให้ติด 1 วินาที ดับ 4 วินาที สัญญาณที่ได้นำมาผ่าน NOT GATE เพื่อให้ได้สัญญาณเฟสที่หนึ่งและเฟสที่สอง โดยกลับกัน 180 องศา เพื่อนำสัญญาณที่ได้ไปขยาย โดยภาค dirver แบบ bridge amp ซึ่งใช้การ Coupling ด้วย Op-to couple ขับทรานซิสเตอร์ ทั้ง 4 ตัว โดยให้เฟสกลับกันซึ่งใช้ไฟเลี้ยง $\pm$ 35 volt จากการต่างของวงจรทั้งสองจะได้ output ประมาณ 140 V.

3.6 Voice/Playback device (รูปที่ 3.6)

เป็นเครื่องบันทึกเสียงโอเปอร์เรเตอร์ทำหน้าที่บันทึกข้อความซึ่งเป็นเสียงพูดว่า"ระบบโทรศัพท์อัตโนมัติ กรุณากดเลขหมายที่ต้องการติดต่อ"

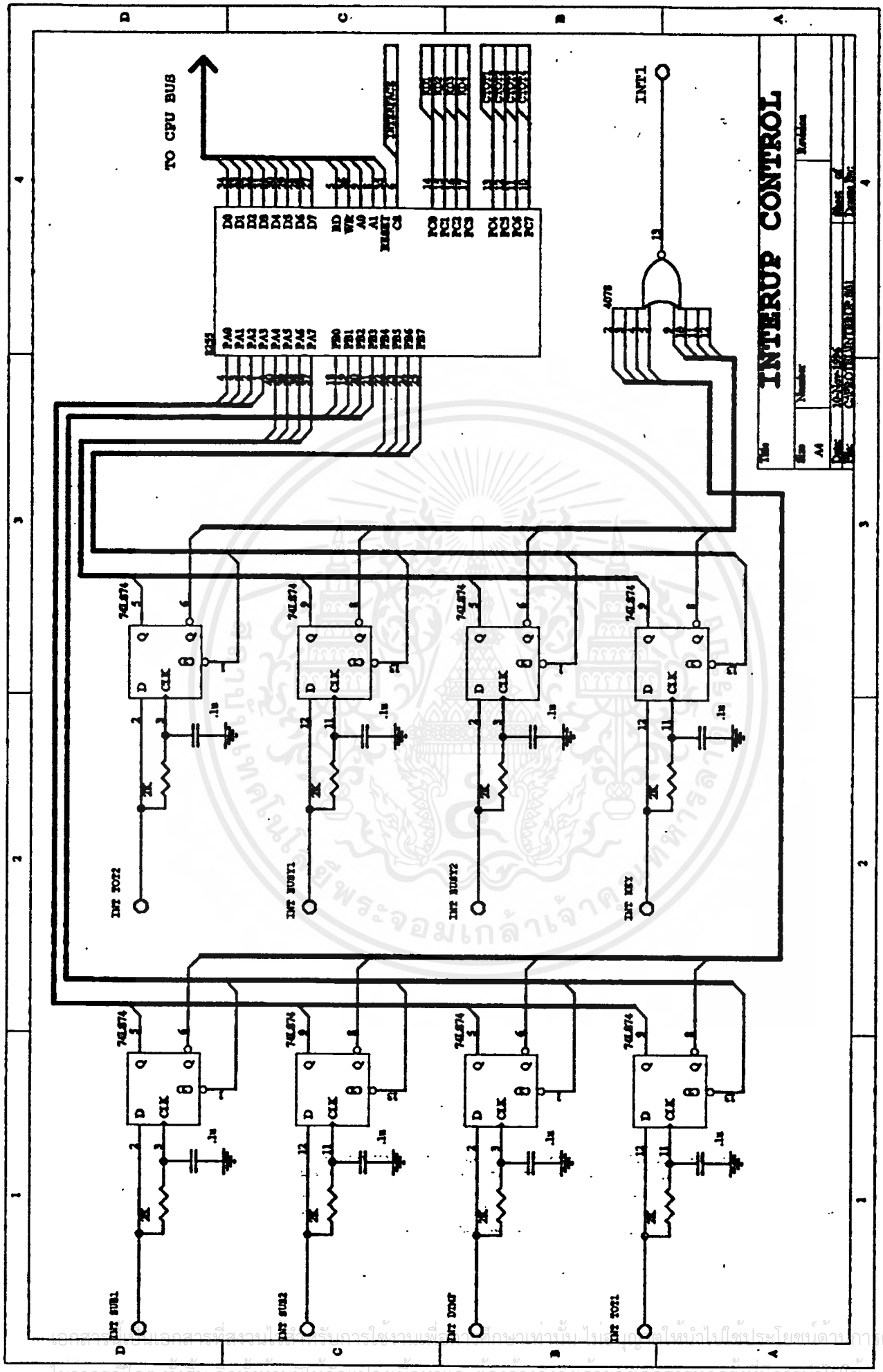
ISD 1402 คือ IC ที่ใช้ในส่วนของ Voice Recorder ซึ่งสามารถบันทึกเสียงได้ 20 วินาที โดยที่ IC เบอร์นี้ไม่จำเป็นต้องใช้อุปกรณ์ร่วมมากนัก โดยภายในตัว IC สามารถบันทึกเสียง, เรียกเสียงออกมาได้ และสามารถเก็บข้อมูลได้นานถึง 100 ปี และทำงานด้วยกำลังที่ต่ำมาก

การส่งงานให้ ISD 1402 ให้ play back สัญญาณเสียงได้ ที่ขา play โดยจะมีการสั่ง play back เมื่อมีการรับสายทาง TOT LINE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

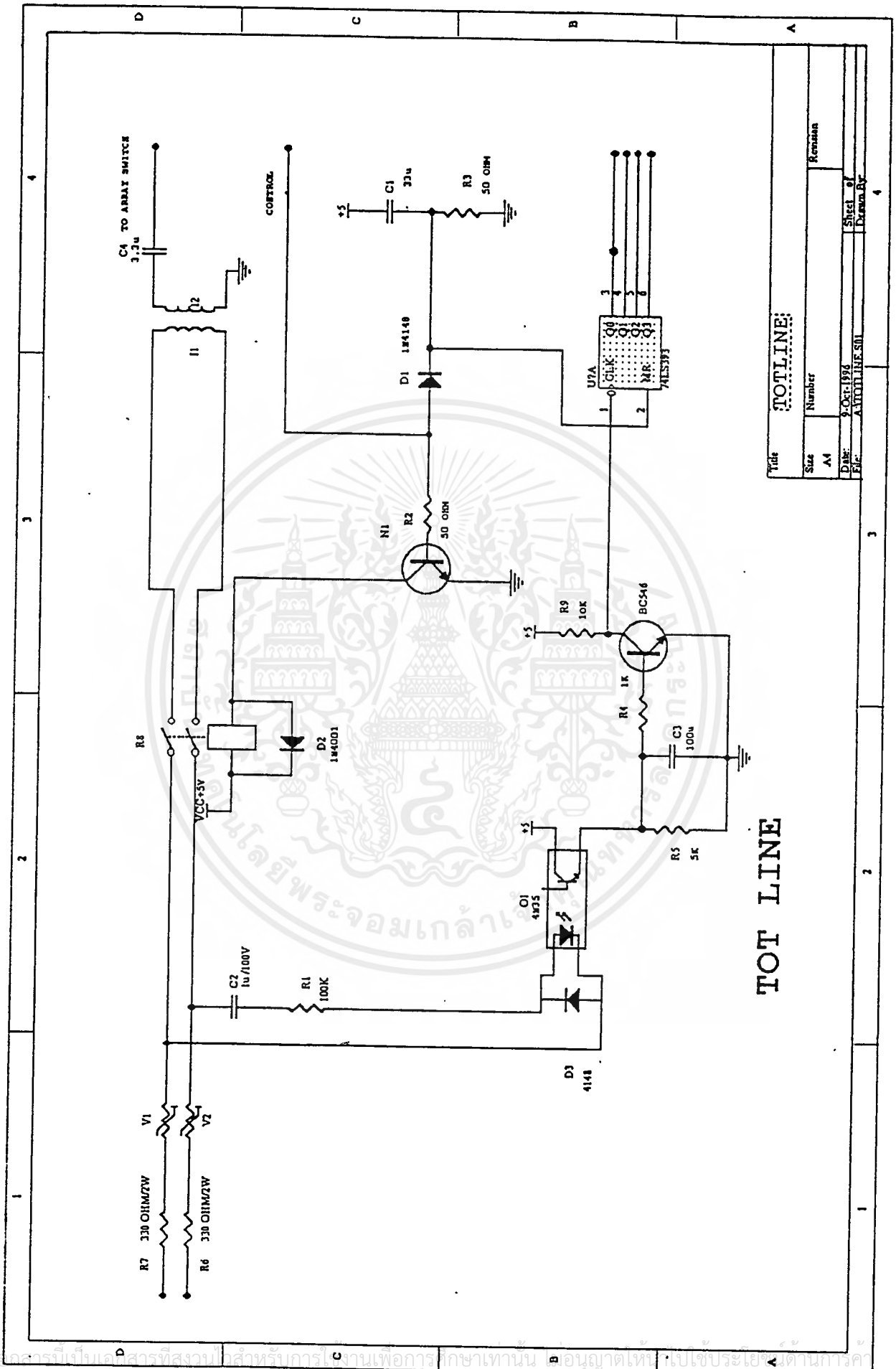
3.7 CIRCUIT DIAGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



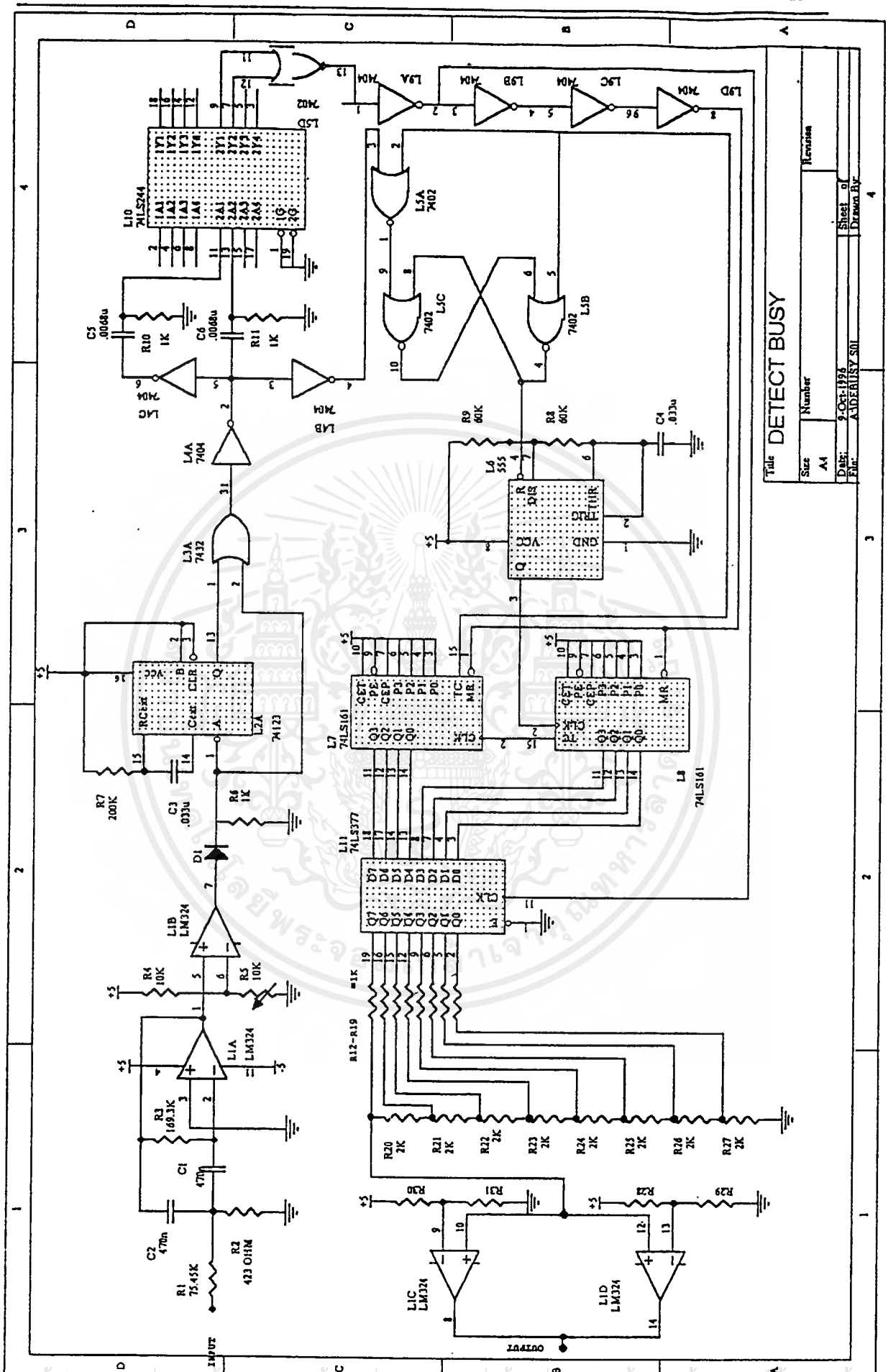
รูปที่ 3.1 วงจร Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



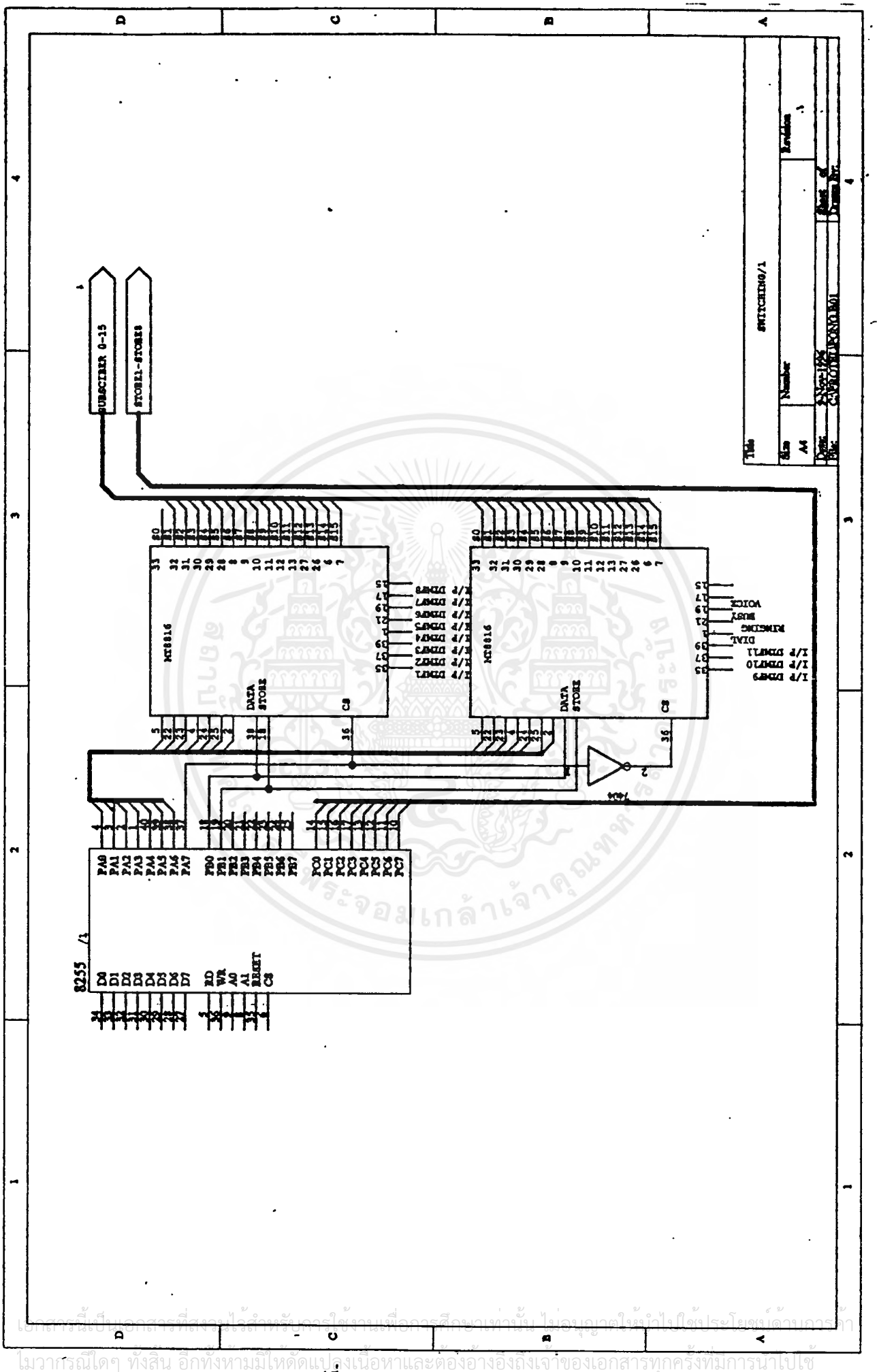
รูปที่ 3.2 TOT Interface Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้



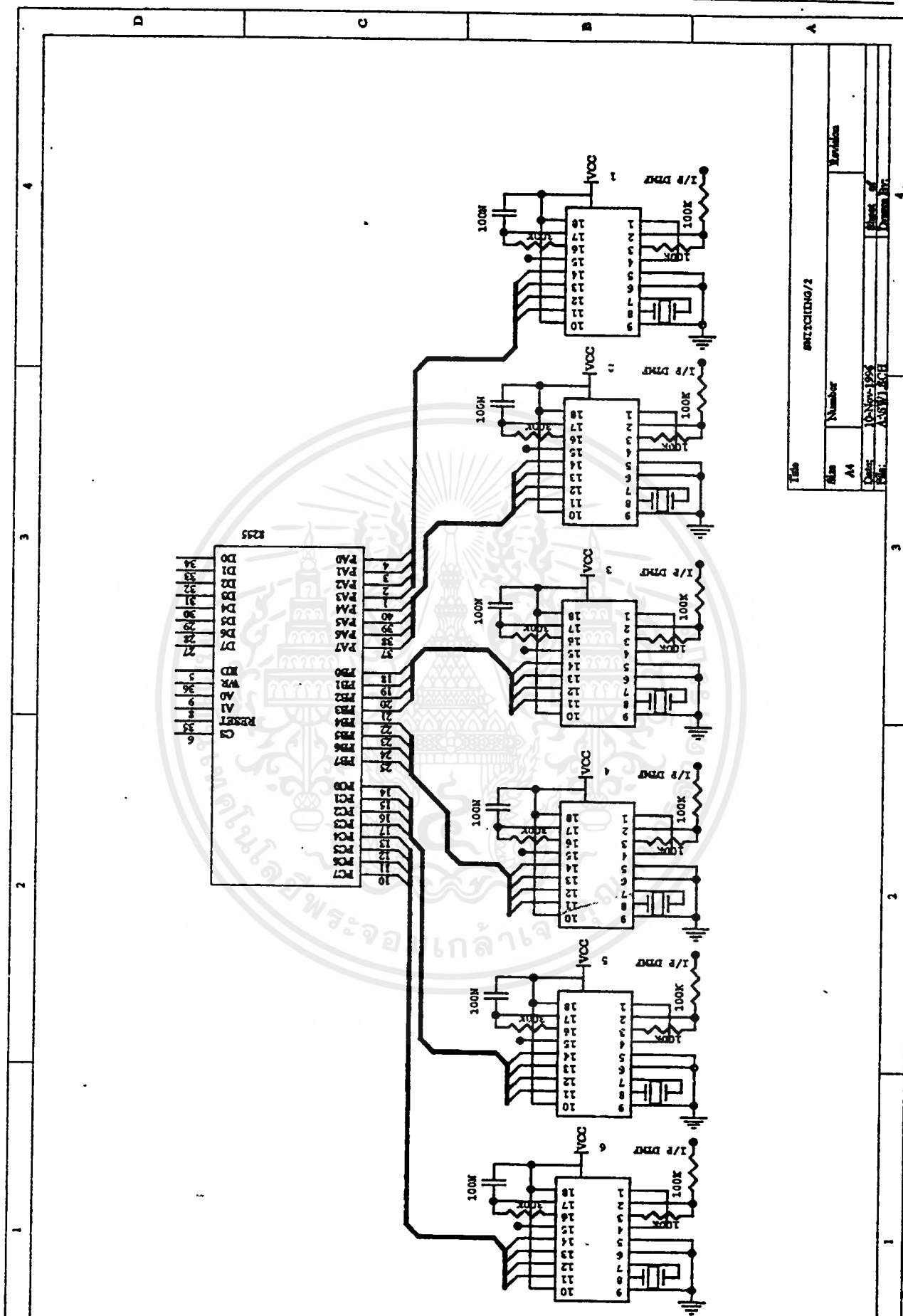
รูปที่ 3.3 Detect Busy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



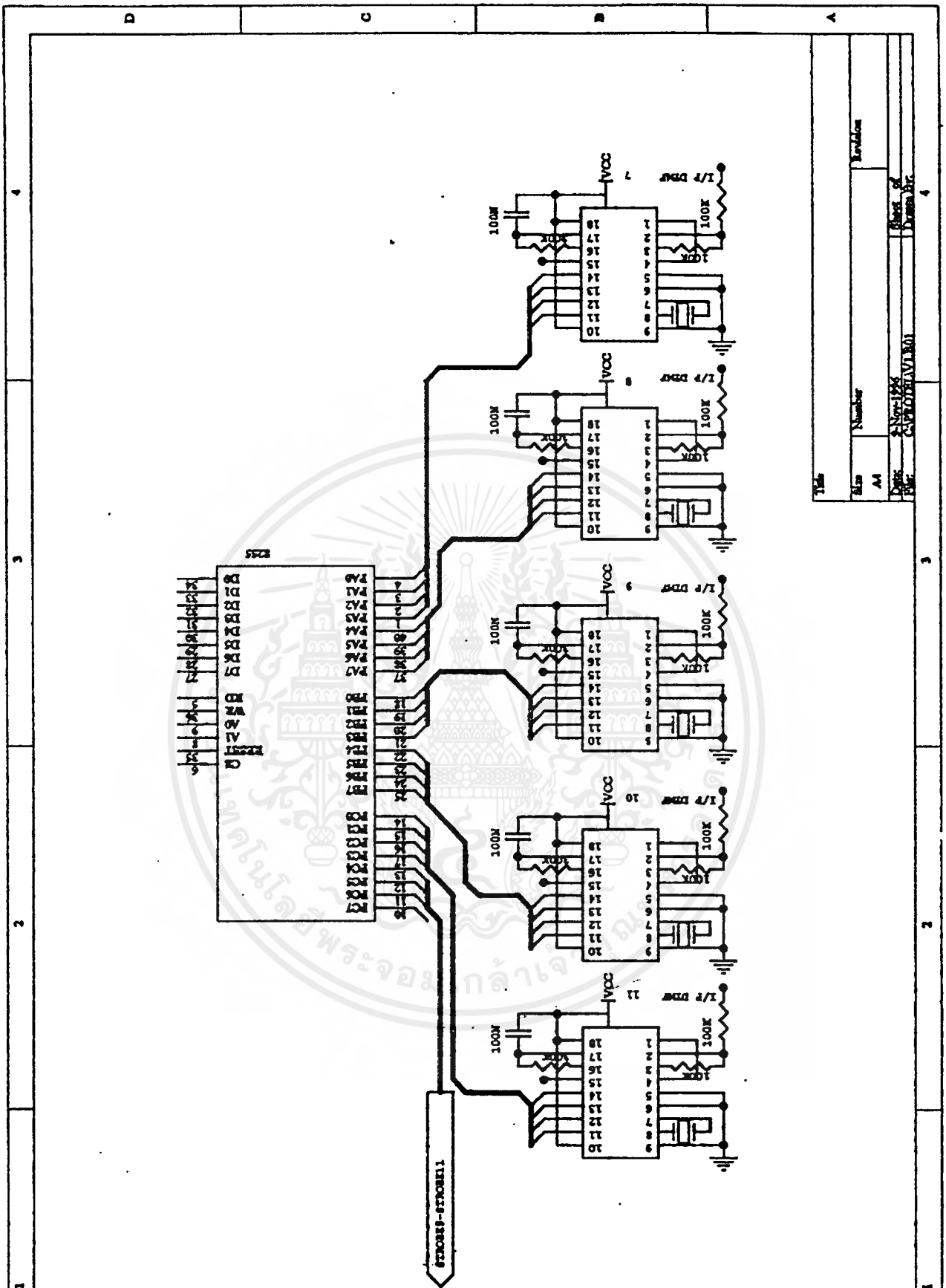
รูปที่ 3.4.1 Switching Part

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



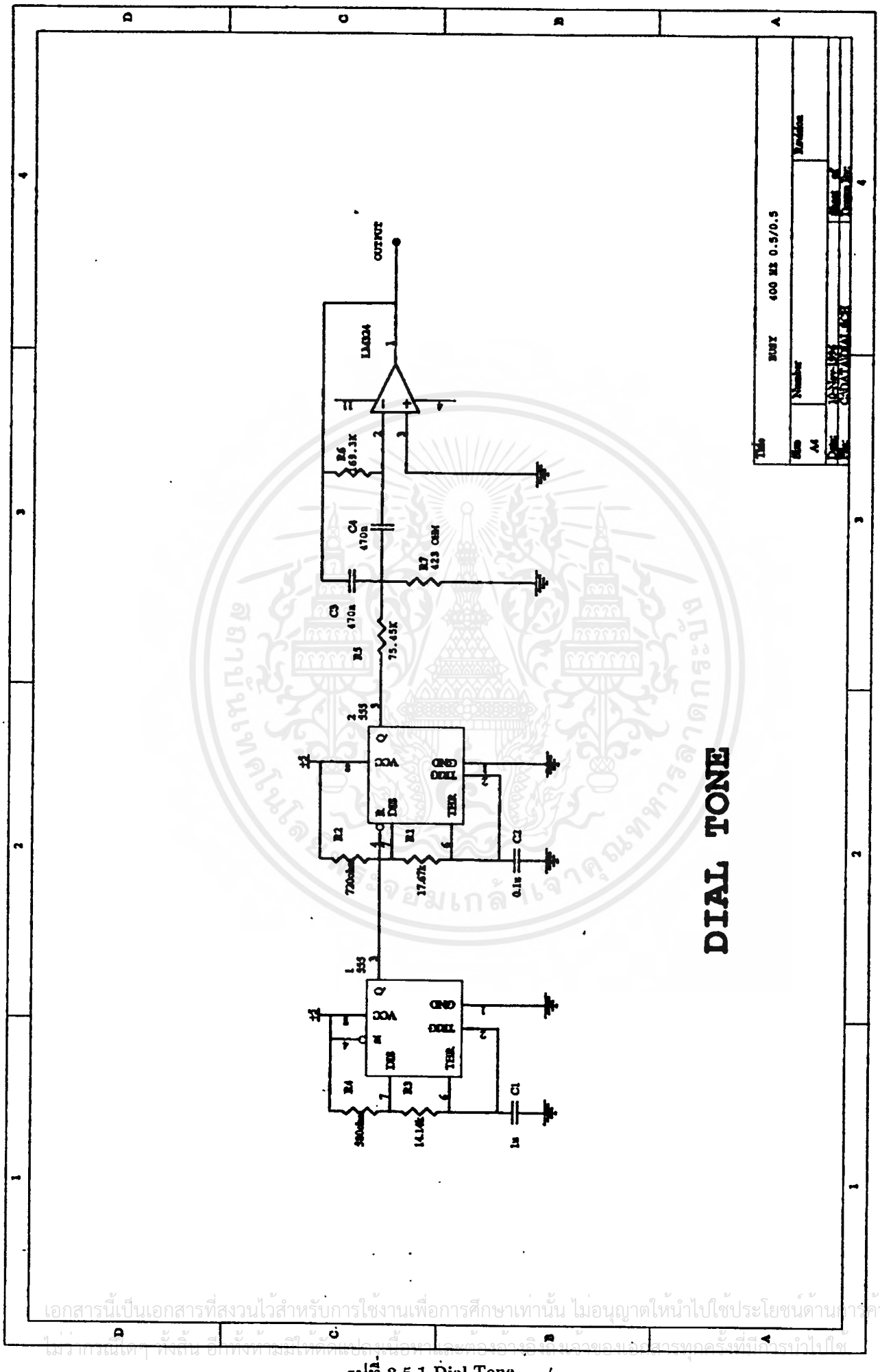
รูปที่ 3.4.2 Switching Part

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



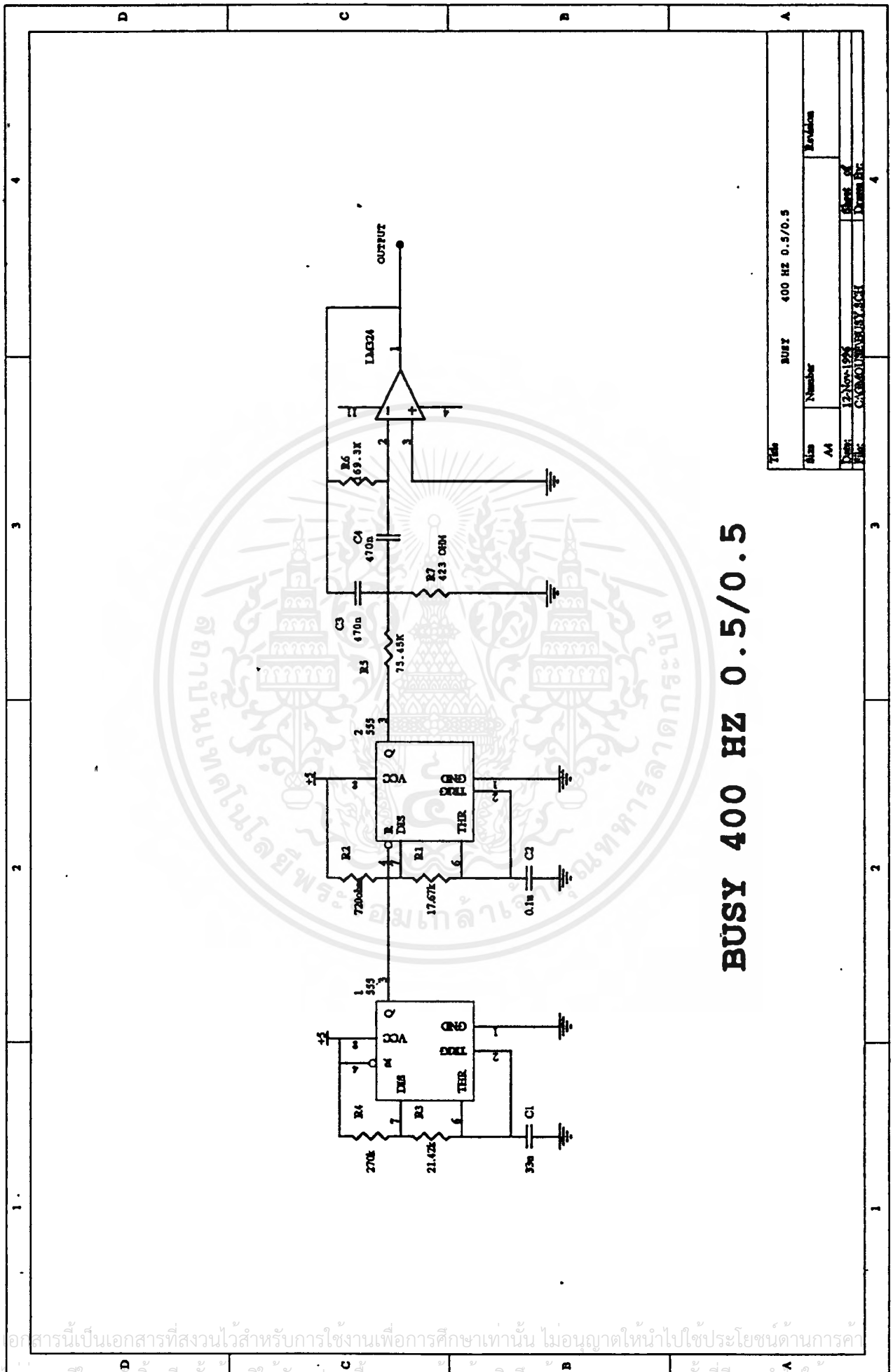
รูปที่ 3.4.3 Switching Part

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



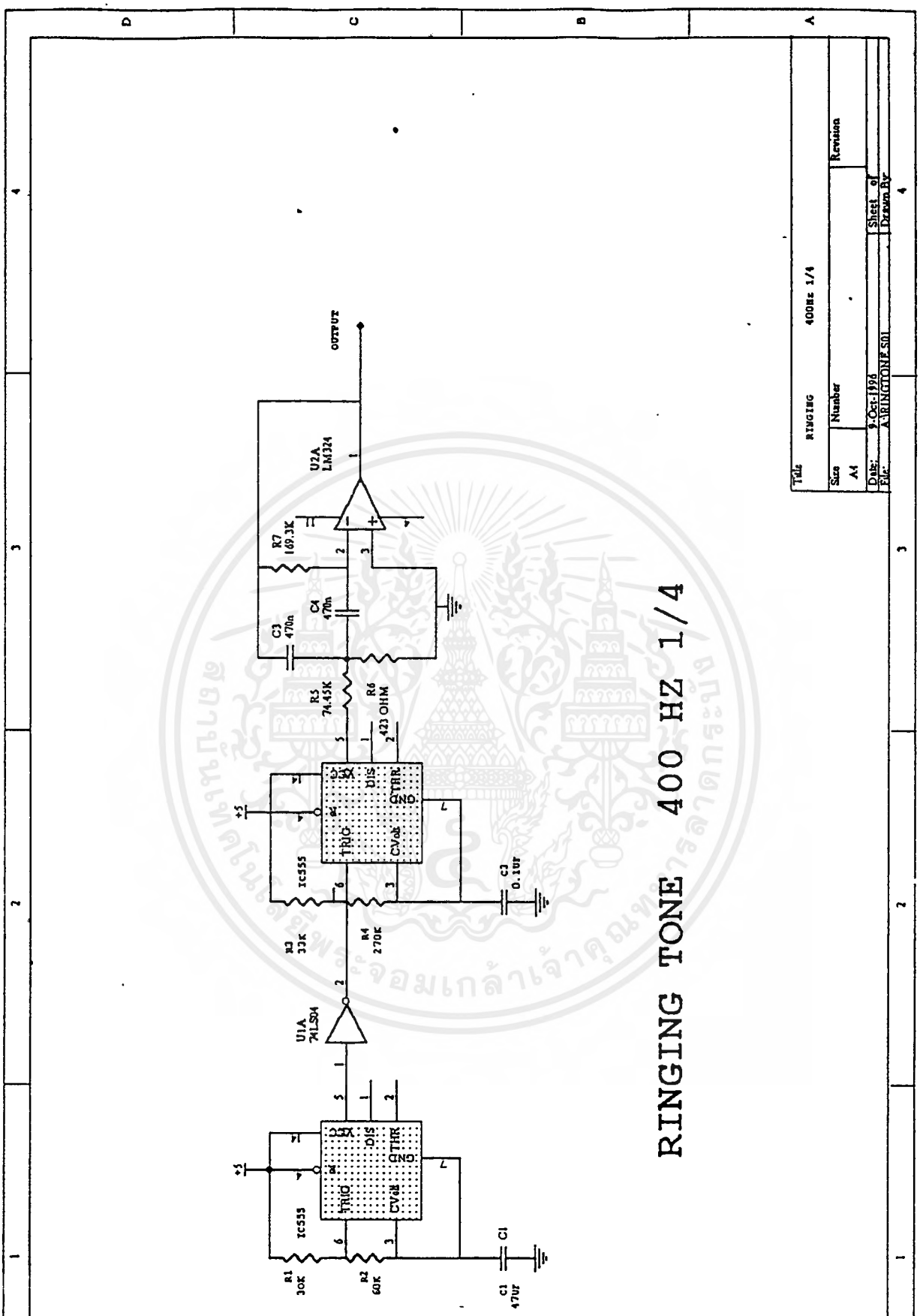
รูปที่ 3.5.1 Dial Tone

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



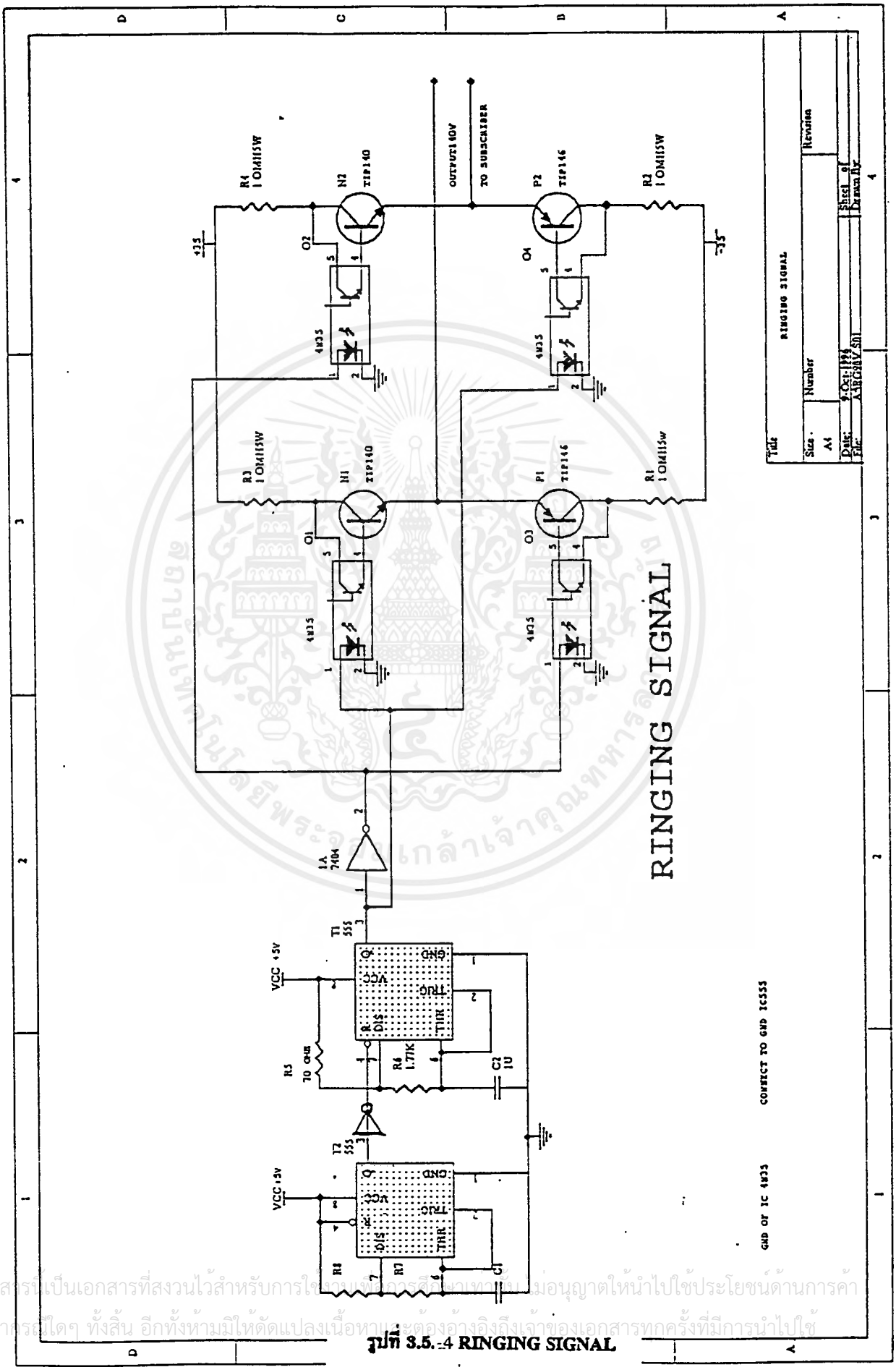
รูปที่ 3.5.2 Busy

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า



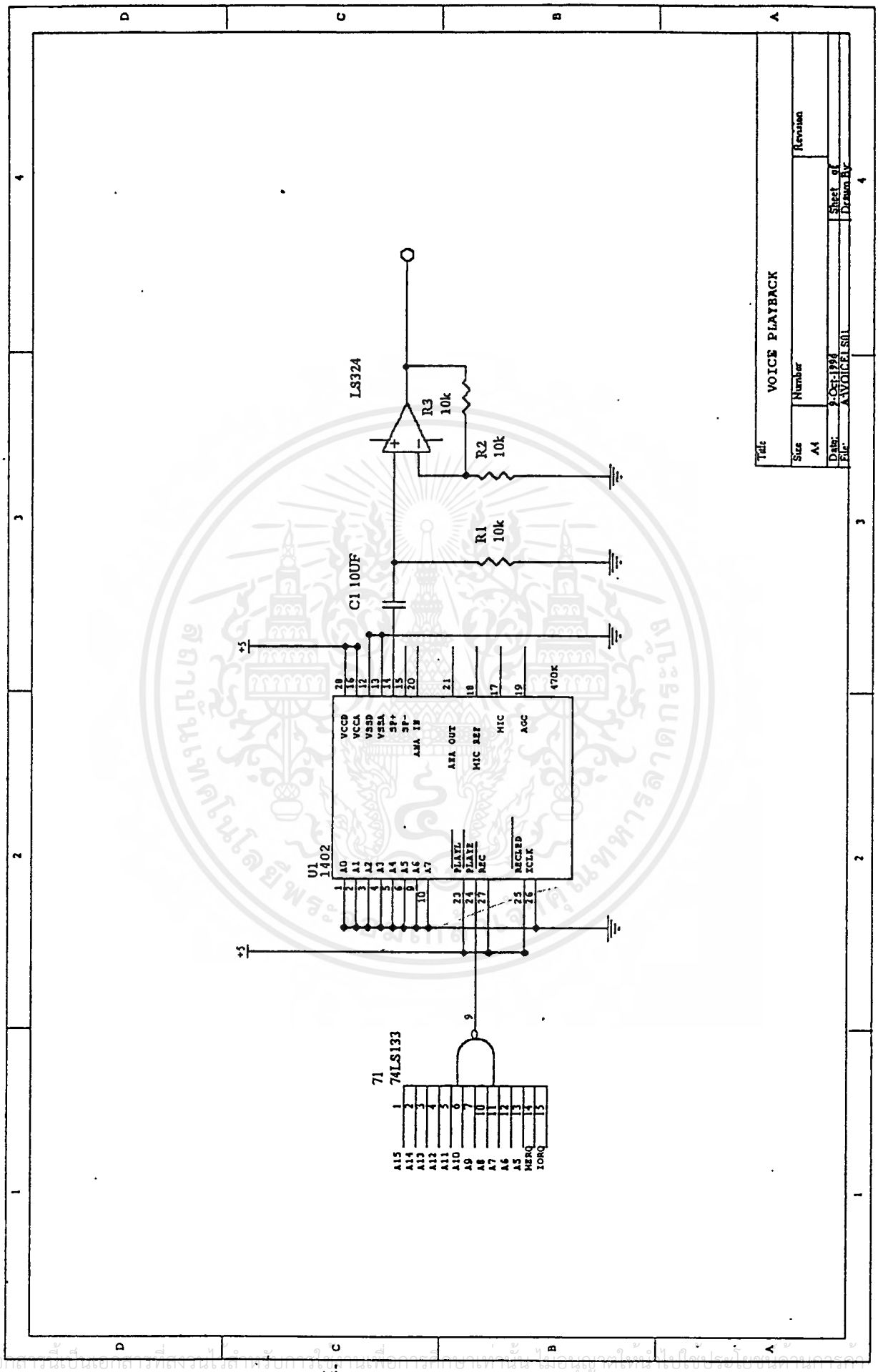
รูปที่ 3.5.3 Ringing Tone

ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของผลงานทุกครั้งที่มีการนำไปใช้



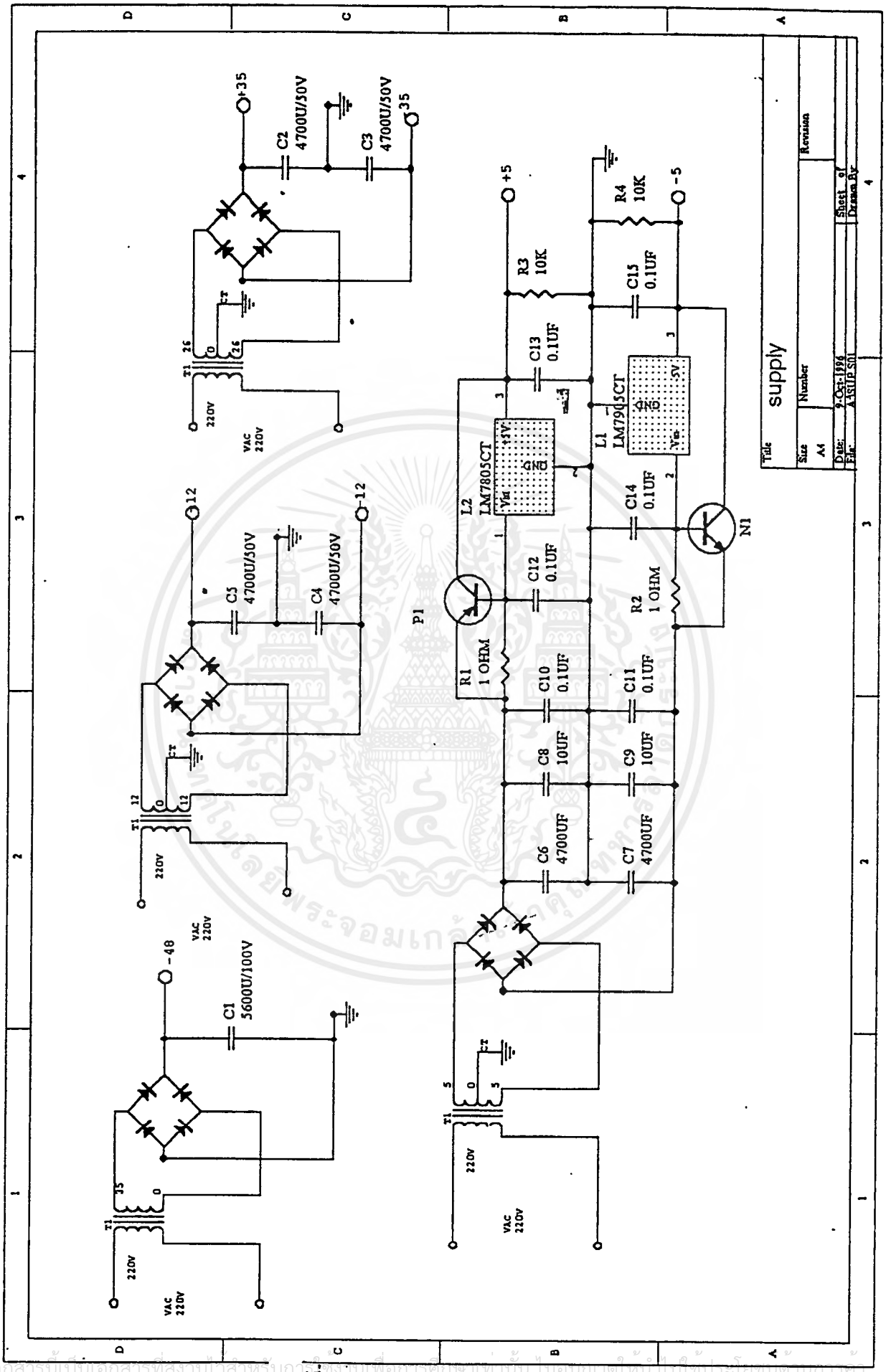
รูปที่ 3.5.-4 RINGING SIGNAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 Voice Playback

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า การจำหน่าย จ่ายแจก หรือการนำไปใช้ในเชิงพาณิชย์ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

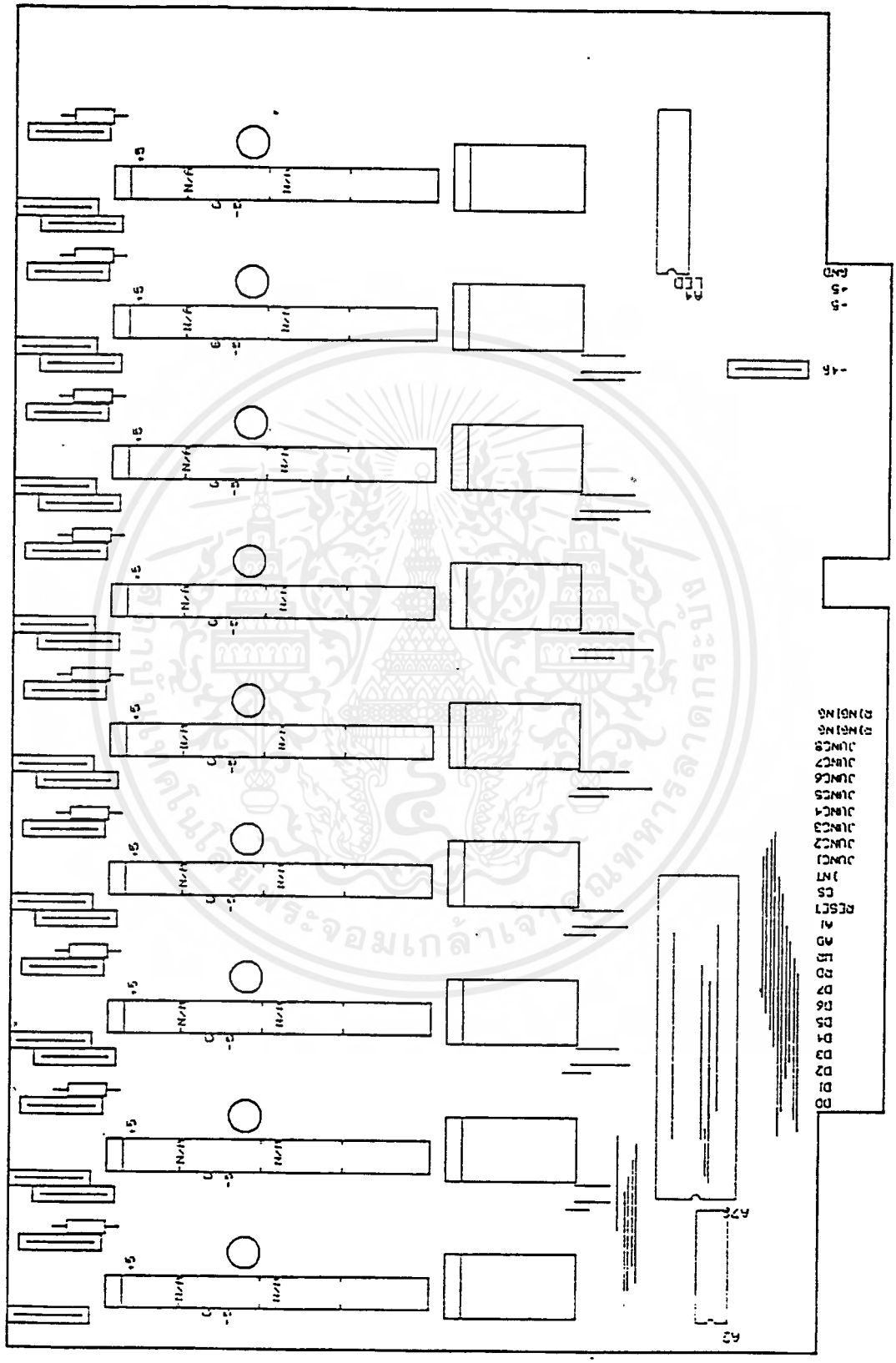


รูปที่ 3.7 Supply

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

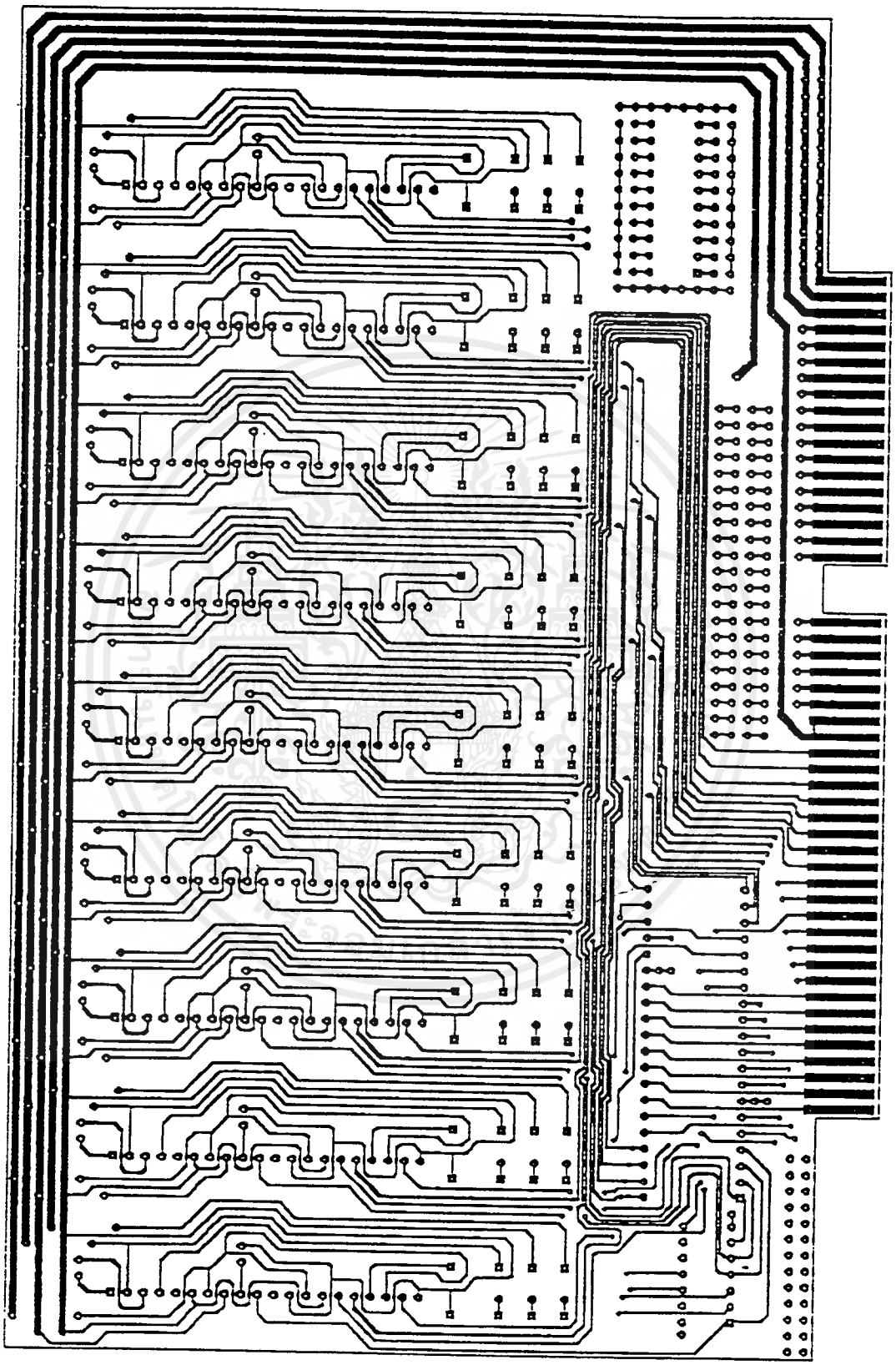
3.8 PRINT BOARD CIRCUIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



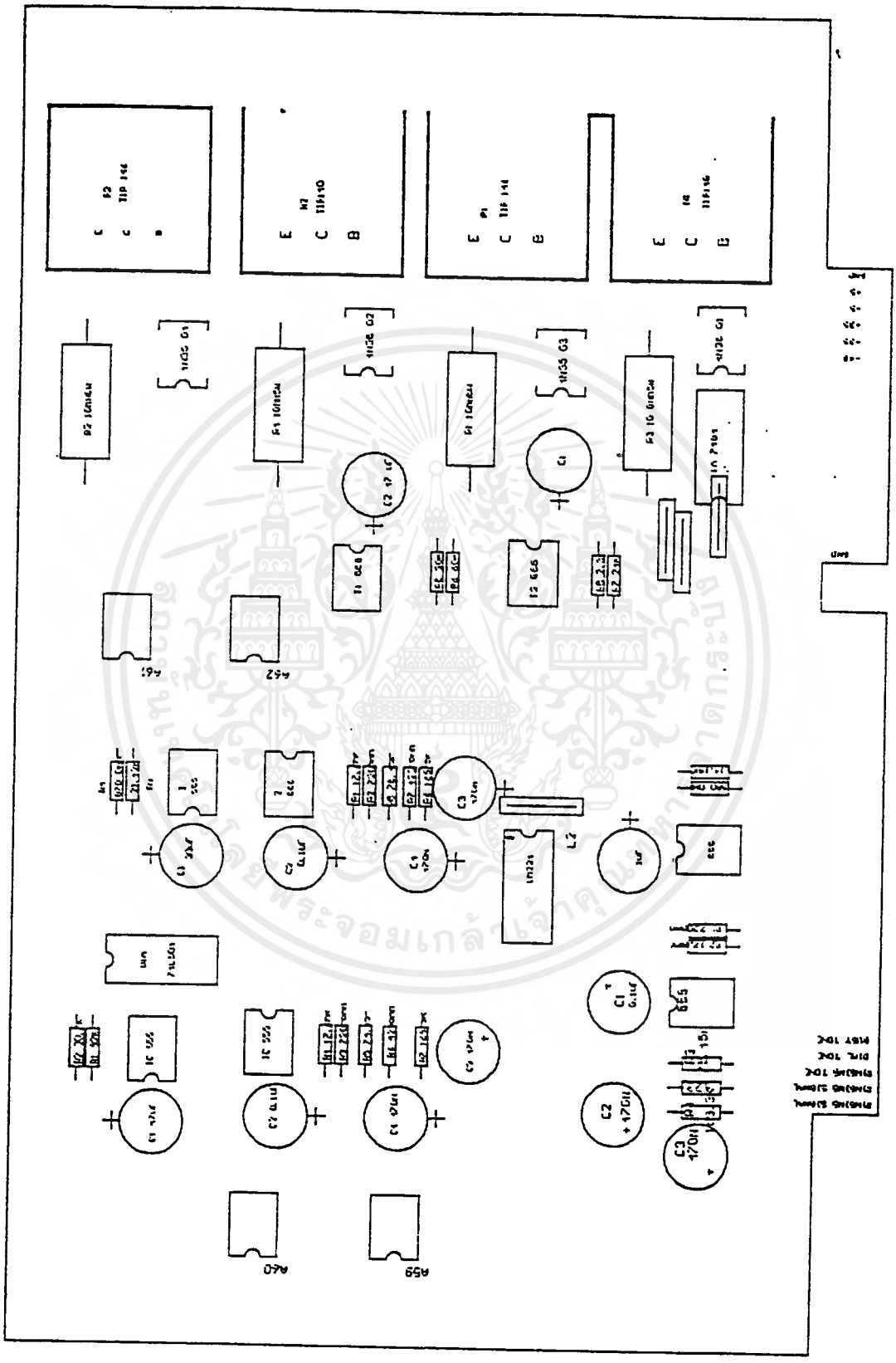
รูปที่ 3.8.1 TOP **SIDE** SUBSCRIBER BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ห้ามมิให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือนำไปใช้ในการแสวงหาผลประโยชน์ทางการค้า ในเอกสารทุกครั้งที่มีการนำไปใช้



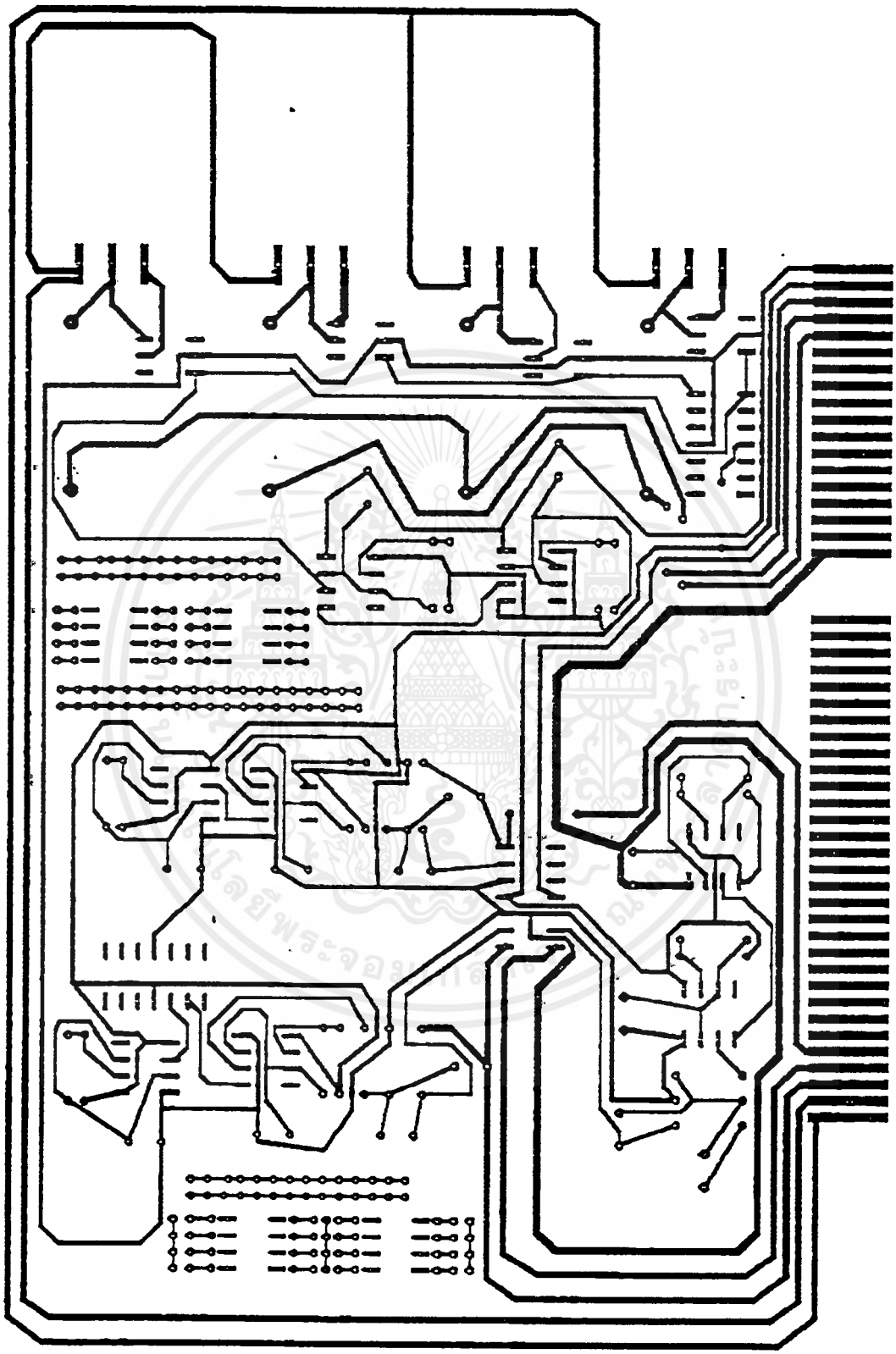
รูปที่ 3.8.2 BOTTOM **SIDE** : SUBSCRIBER BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



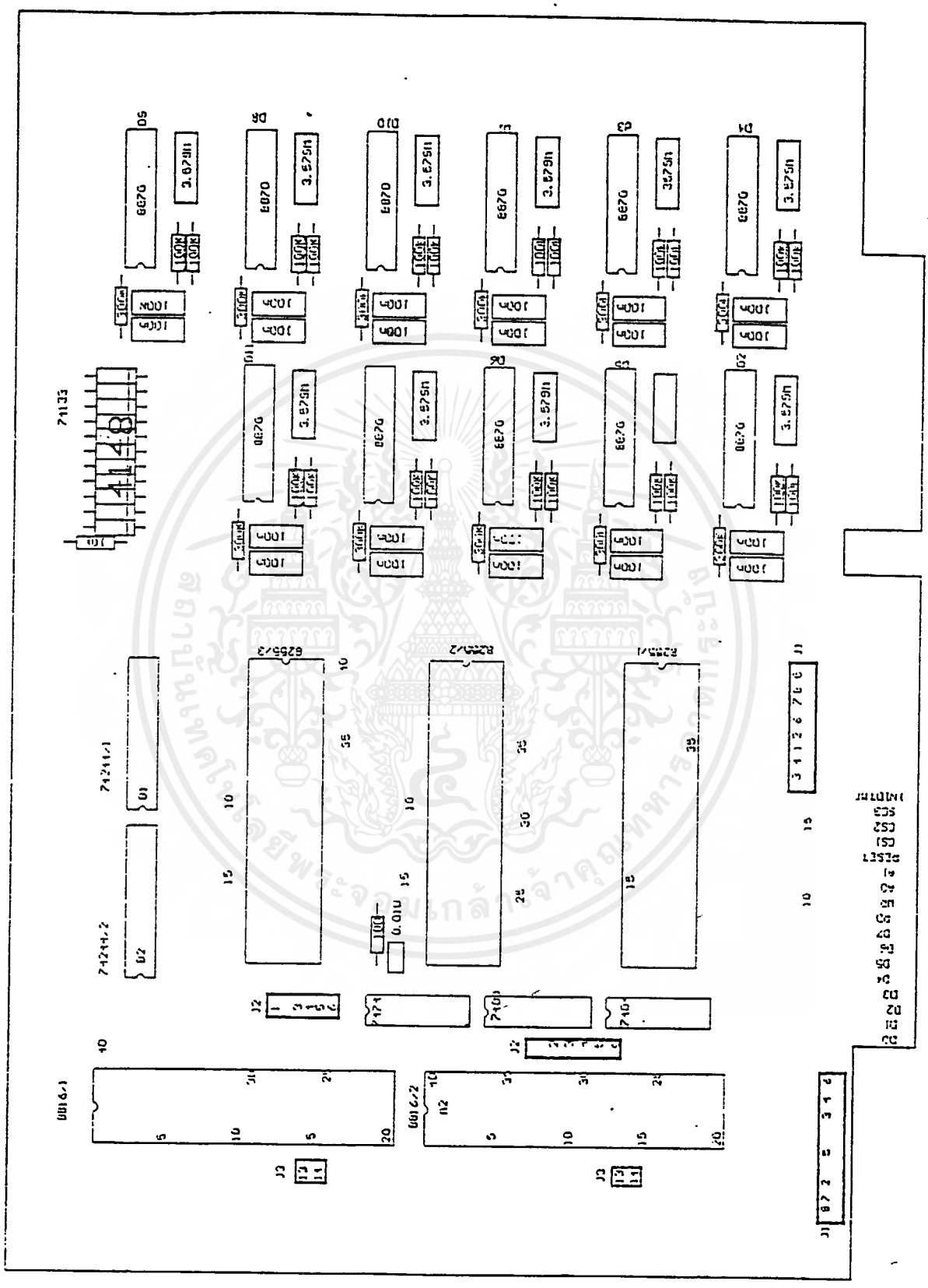
รูปที่ 3.8.3 TOP OVER LAY SIGNAL **BOARD**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



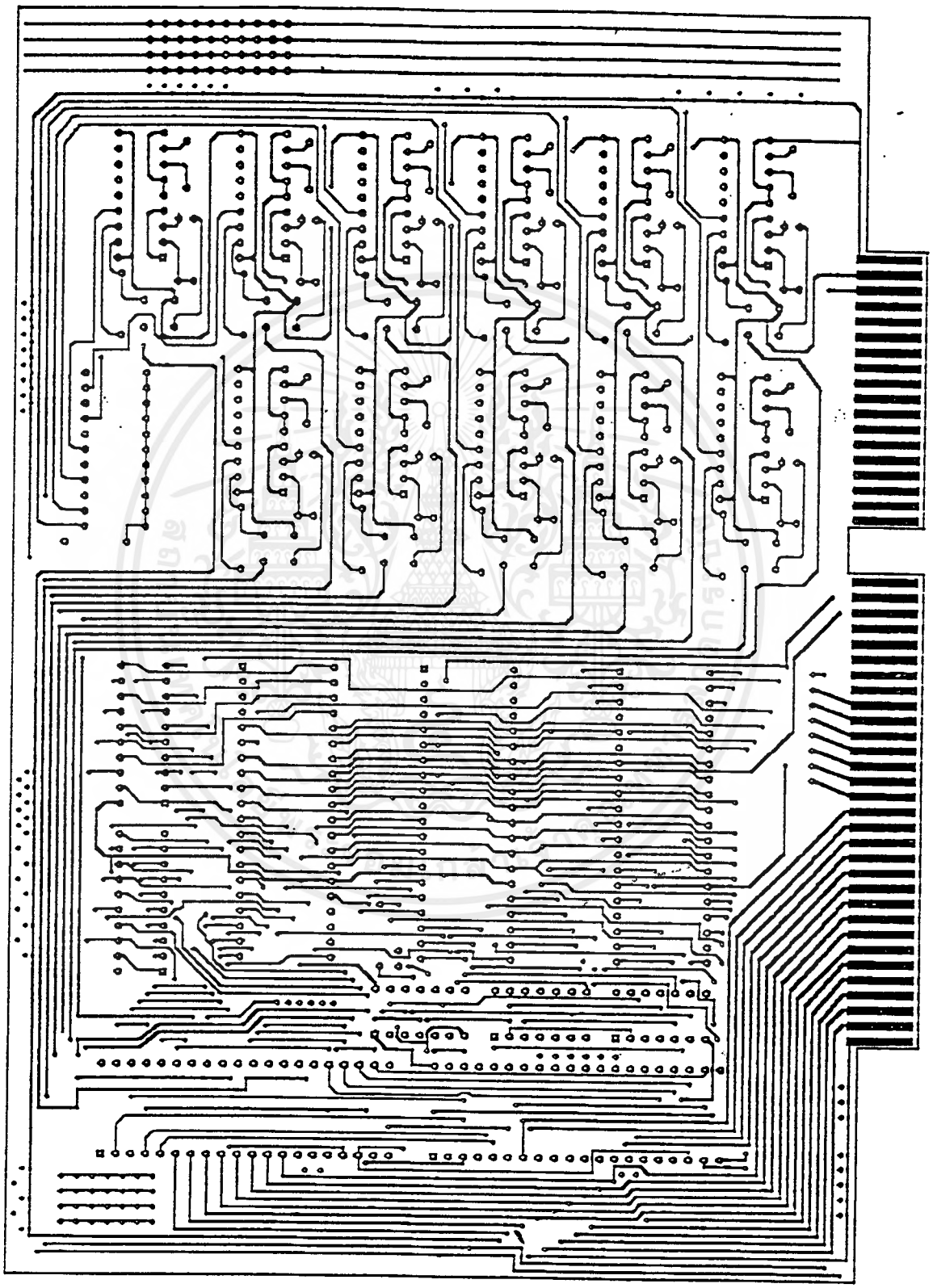
รูปที่ 3.8.4 BOTTOM **SIDE** SIGNAL BOARD

เอกสารนี้เป็น[illegible]ให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



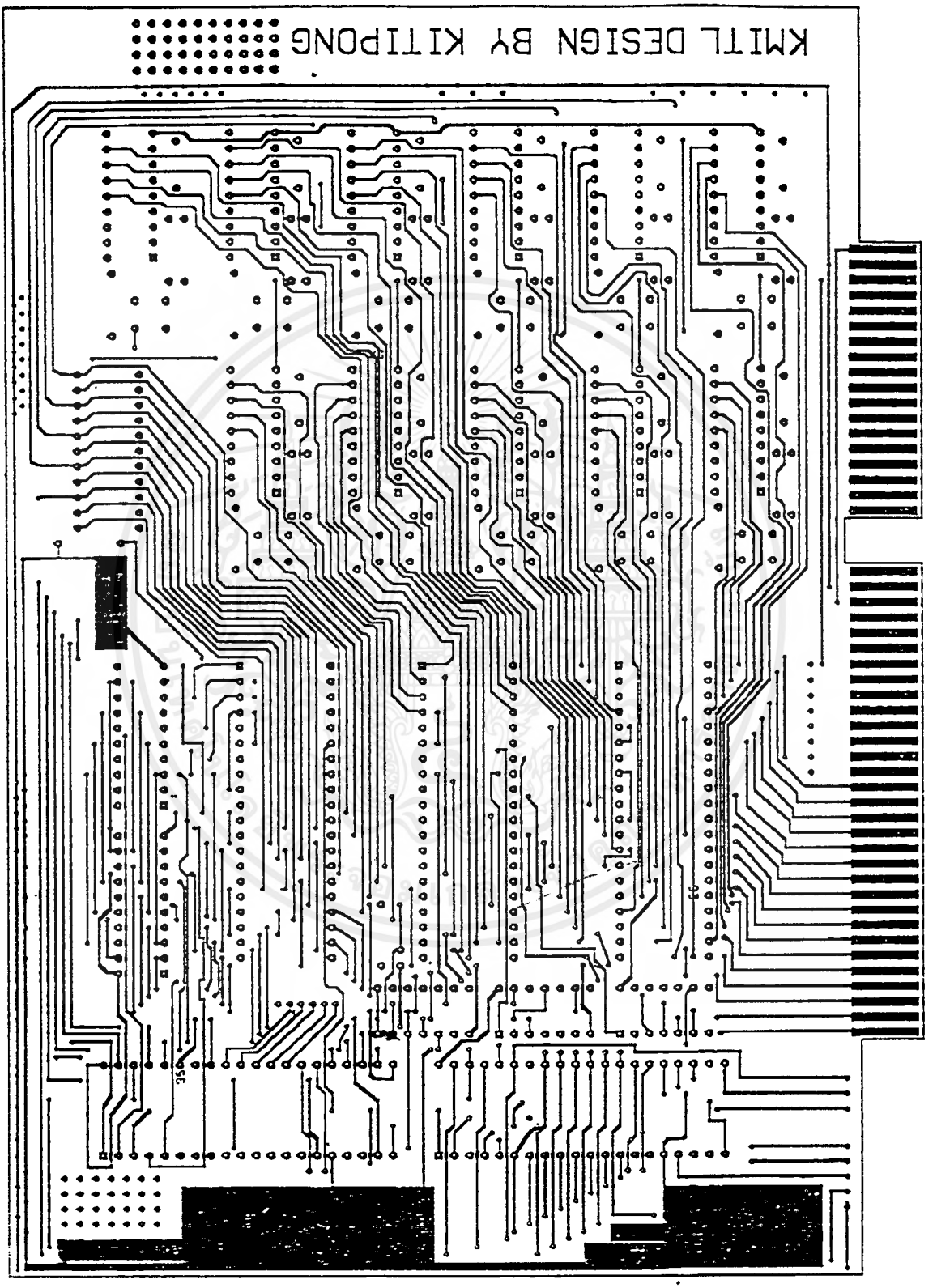
รูปที่ 3.8.5 TOP OVERLAY SWITCHING PART BOARD

เอกสารนี้มีไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



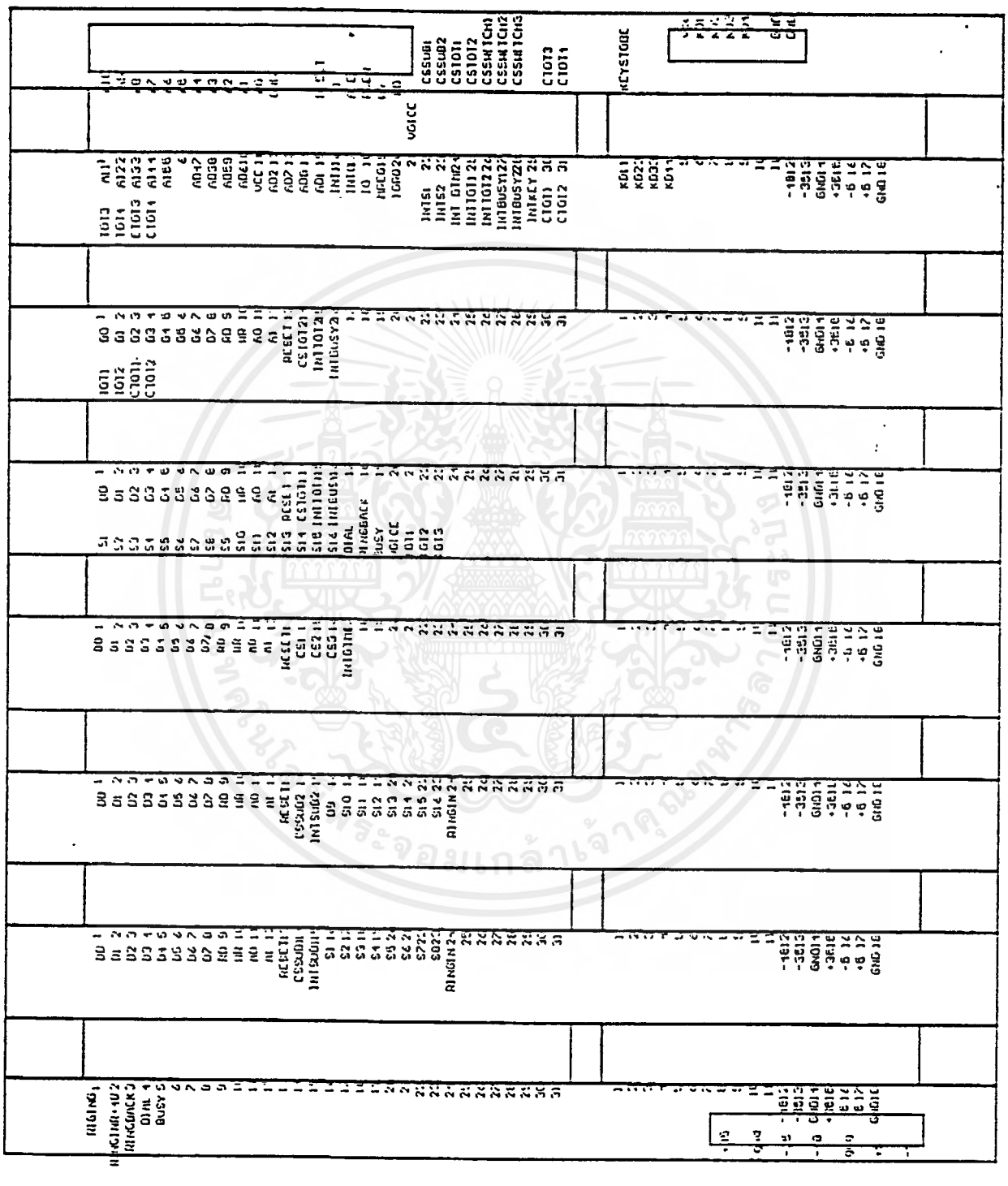
รูปที่ 3.8.6 BOTTOM **SIDE** SWITCHING PART **BOARD**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



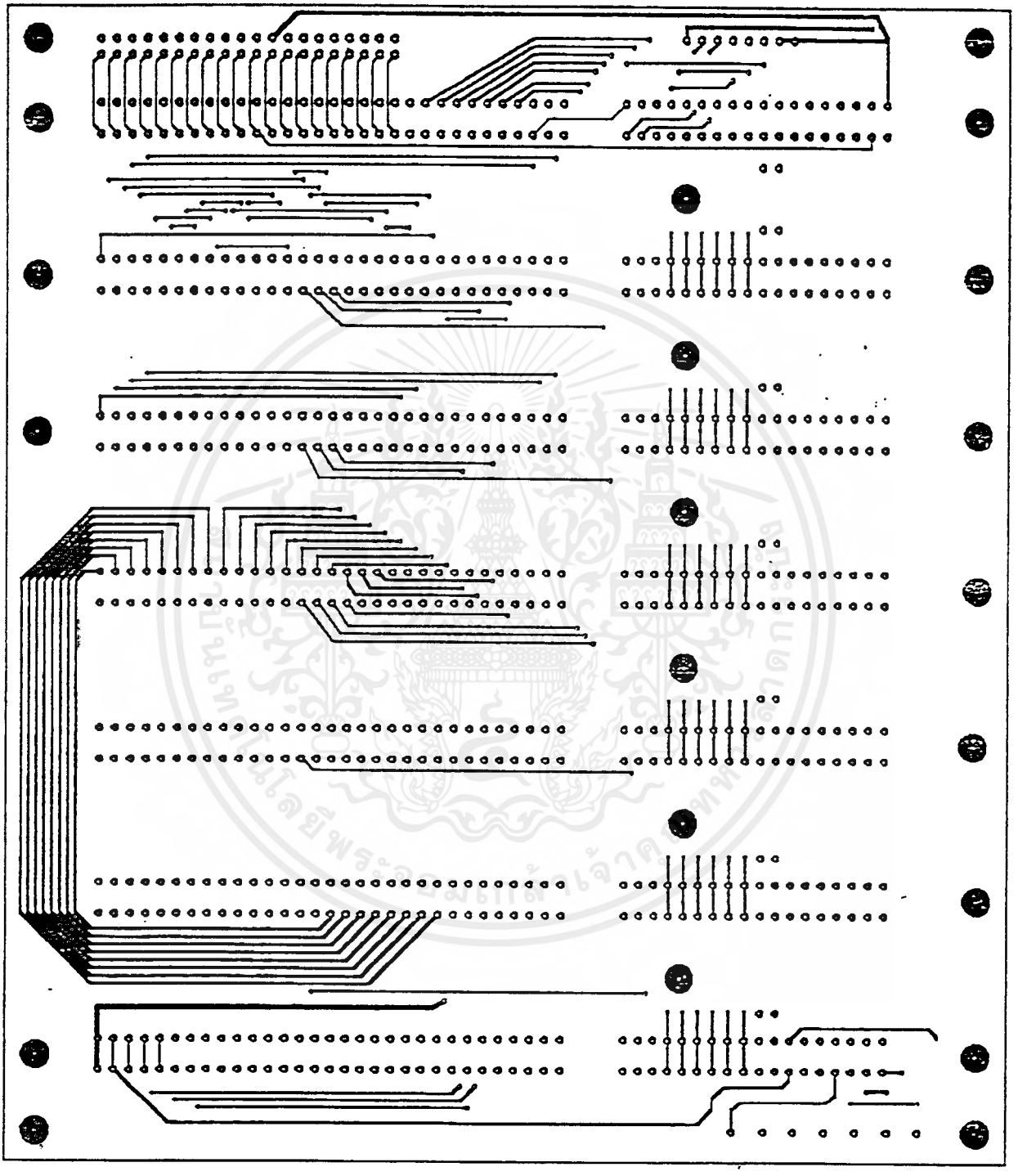
รูปที่ 3.8.7 TOP **SIDE** . SWITCHING PART **BOARD**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



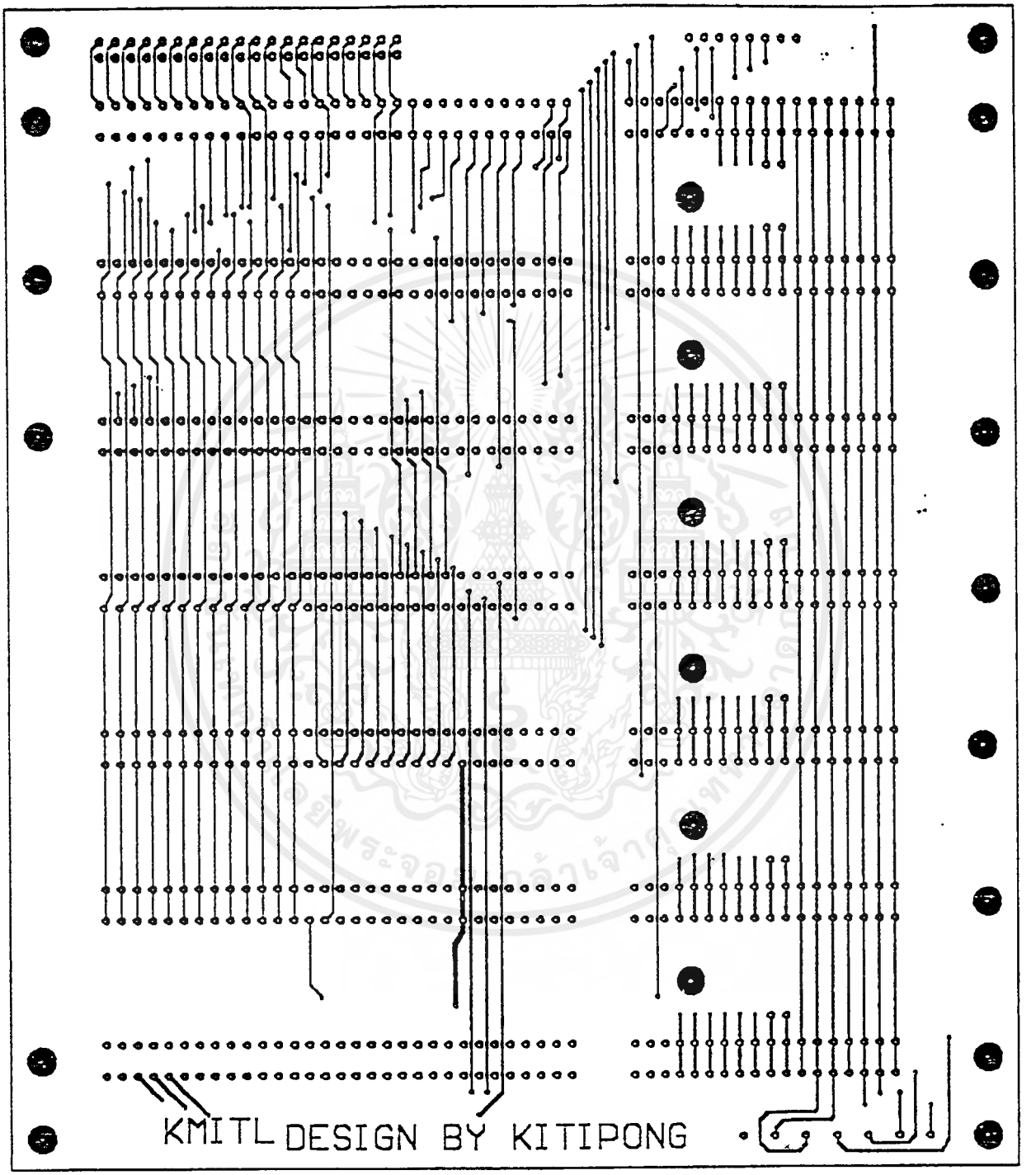
รูปที่ 3.8.8 TOP OVERLAY MASTER BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้



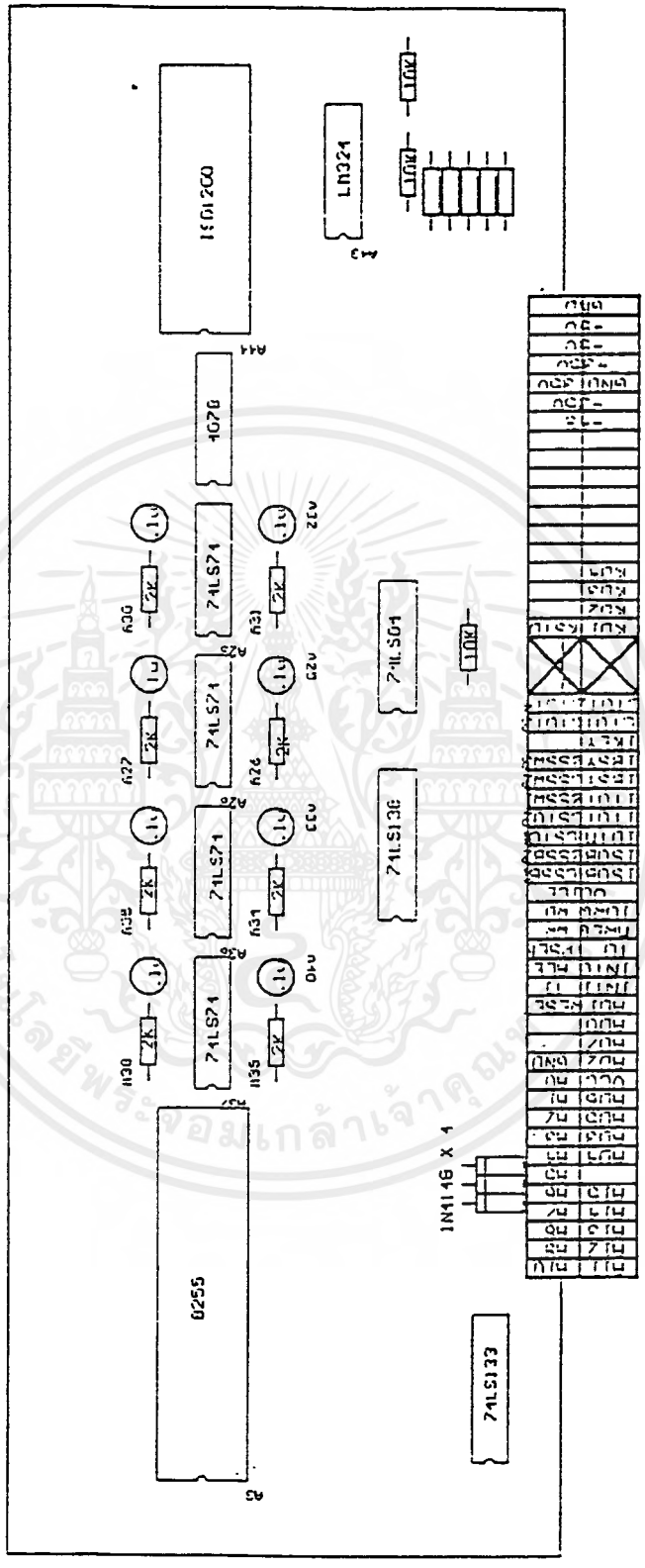
รูปที่ 3.8.9 BOTTOM **SIDE** . MASTER BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



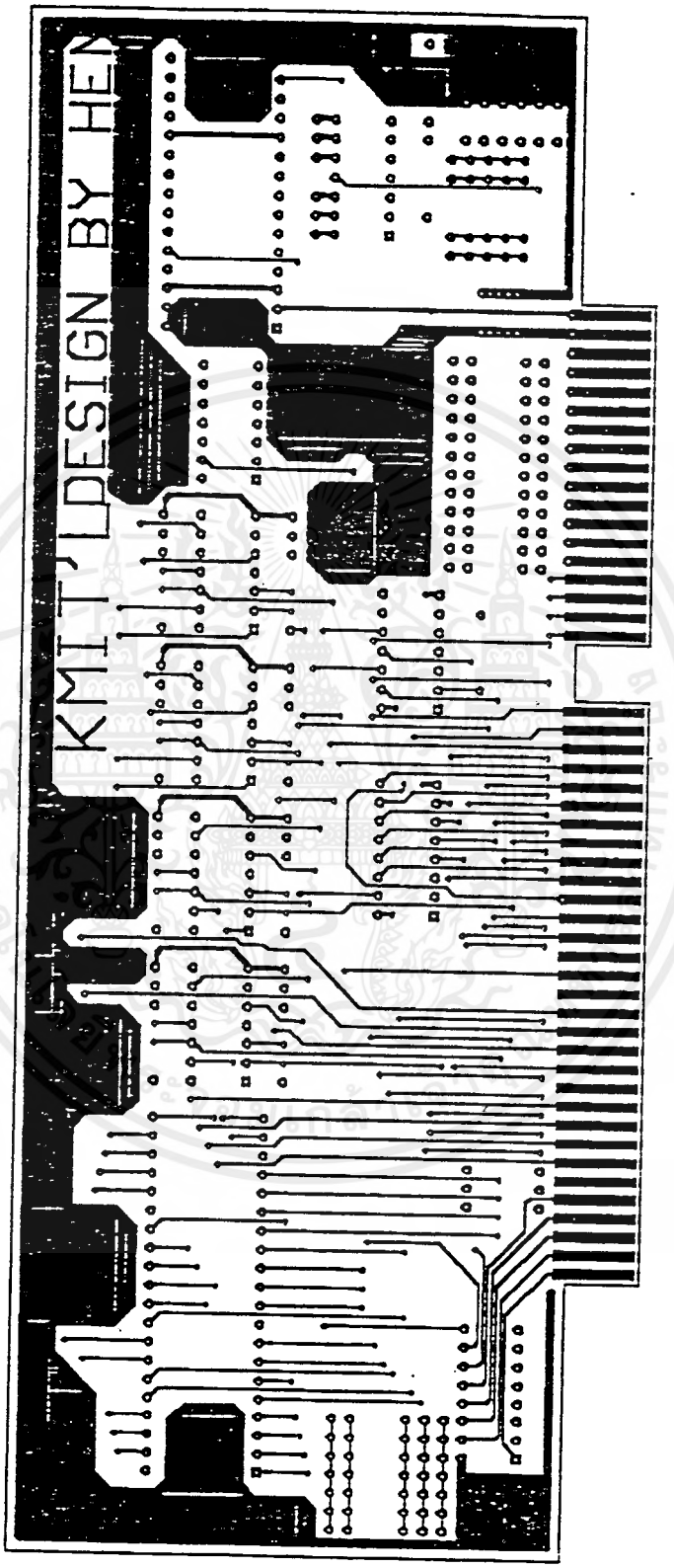
รูปที่ 3.8.10 TOP : **SIDE** : MASTER BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้



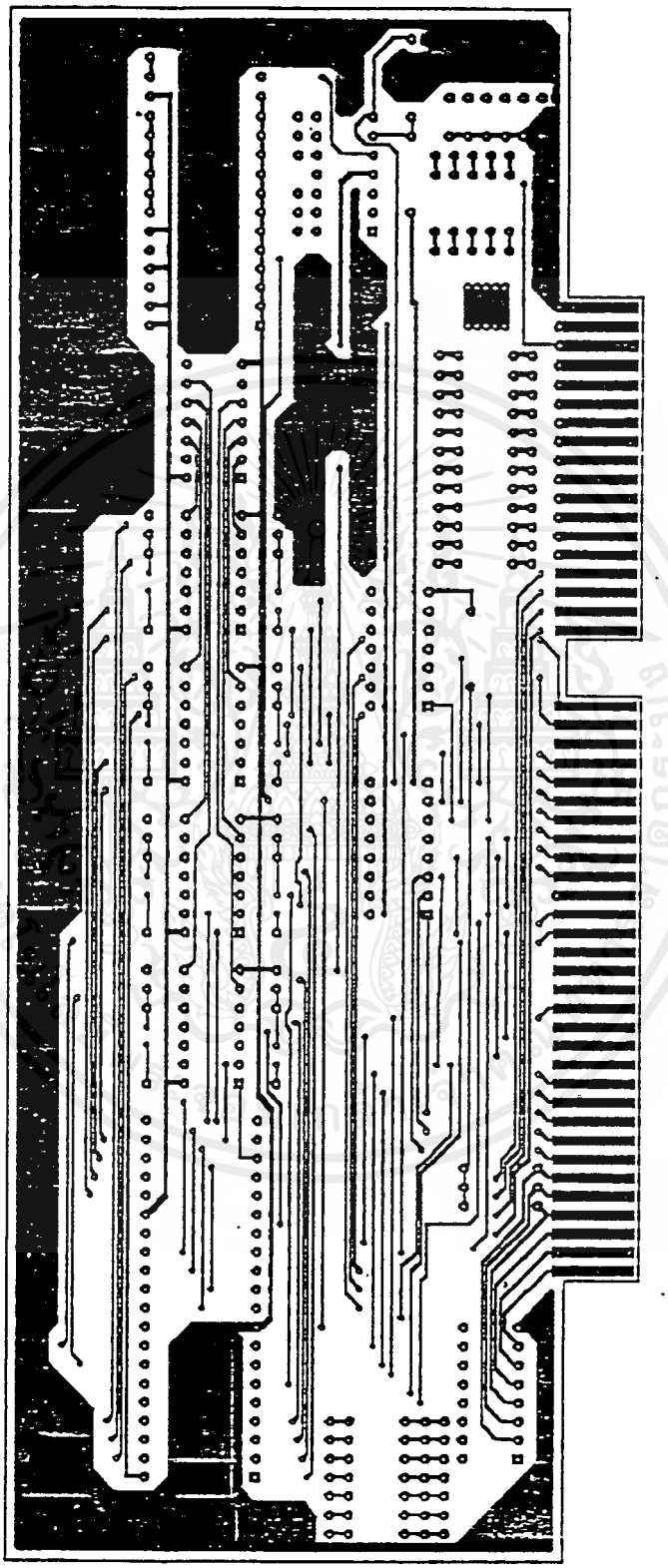
รูปที่ 3.8.11 TOP OVERLAY INTERFACE BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



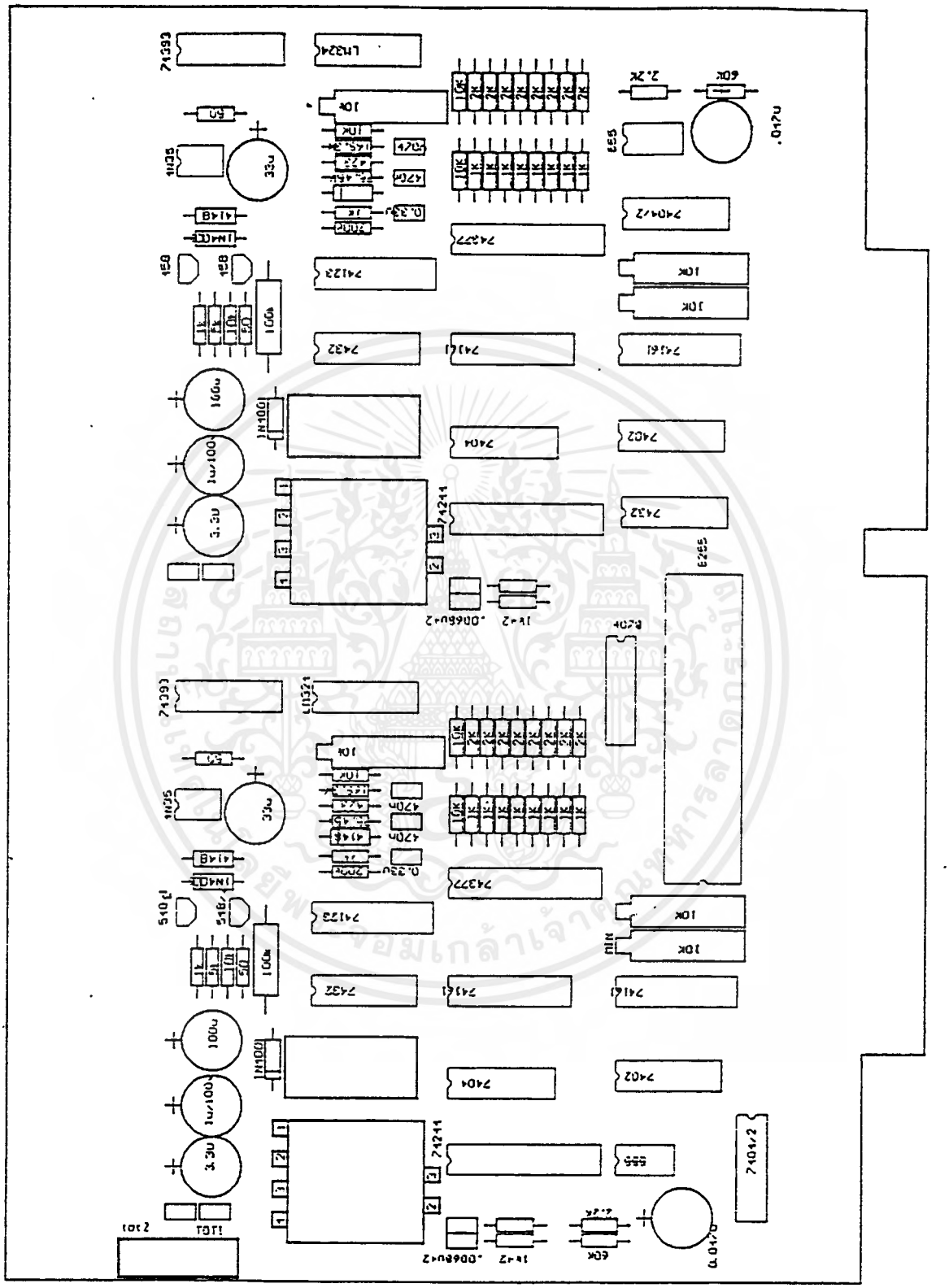
รูปที่ 3.8.12 TOP : **SIDE** : INTERFACE BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



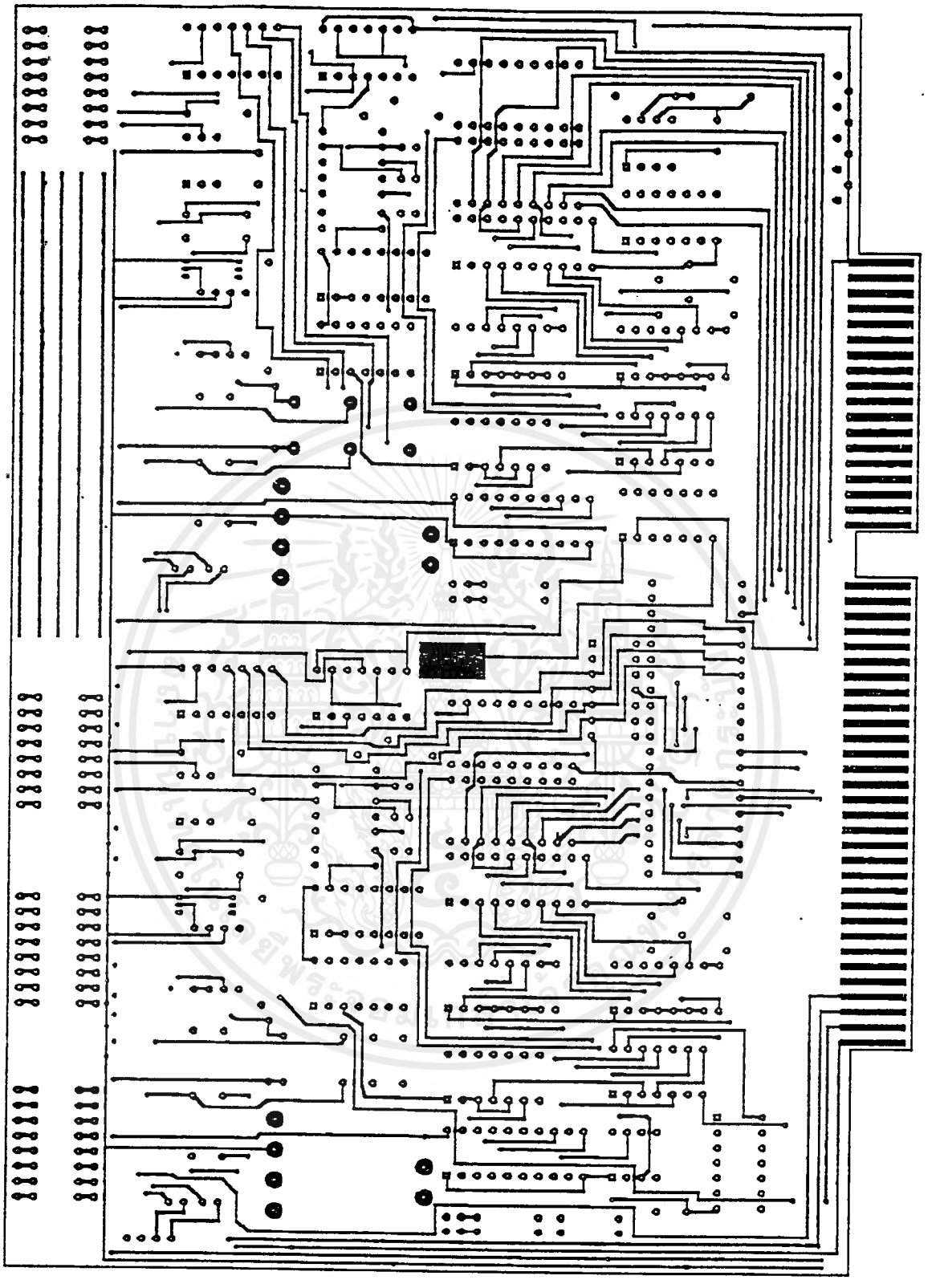
รูปที่ 3.8.13 BOTTOM SIED . INTERFACE BOARD .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



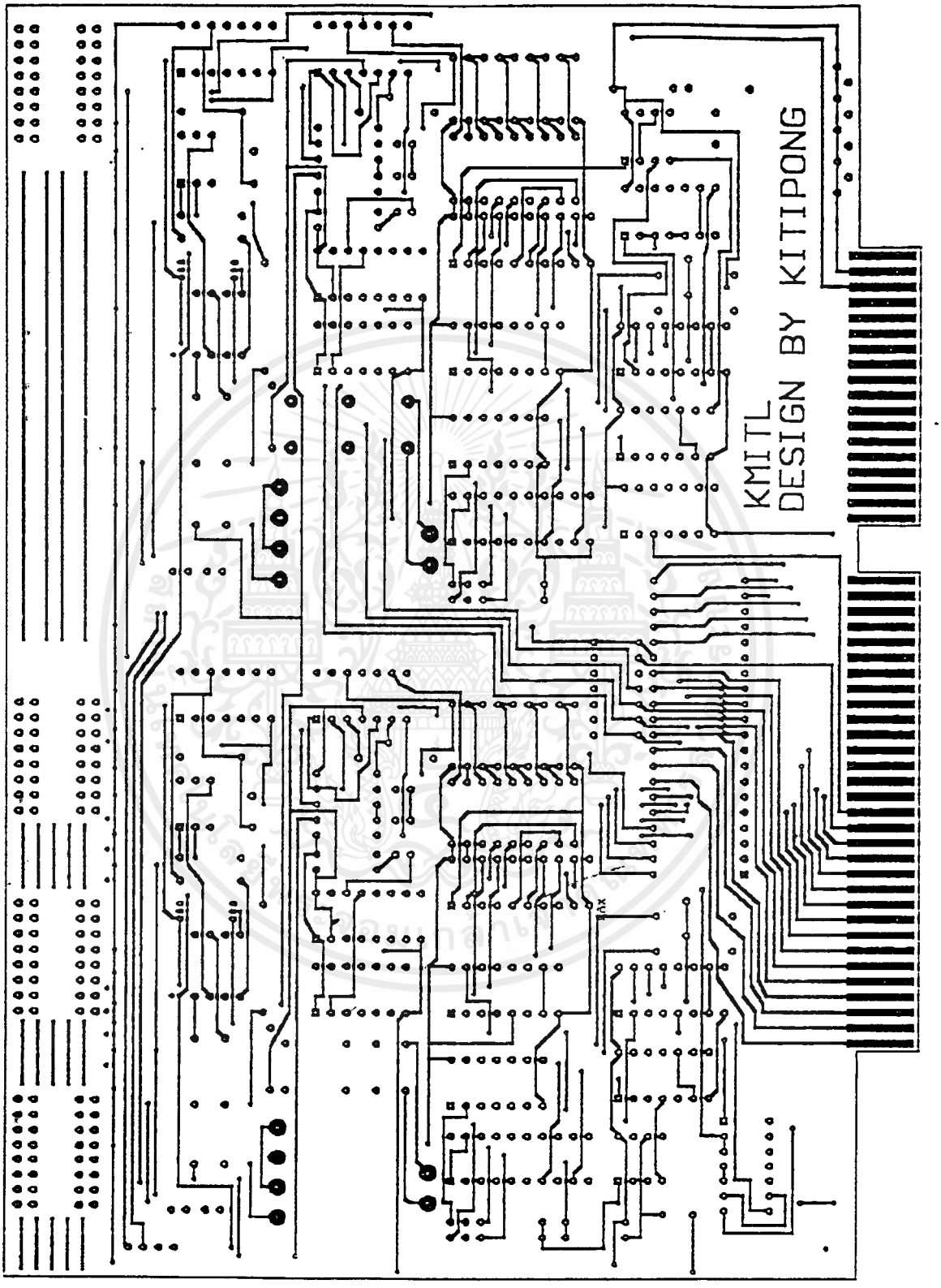
รูปที่ 3.8.14 TOP OVERLAY TOT LINE BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



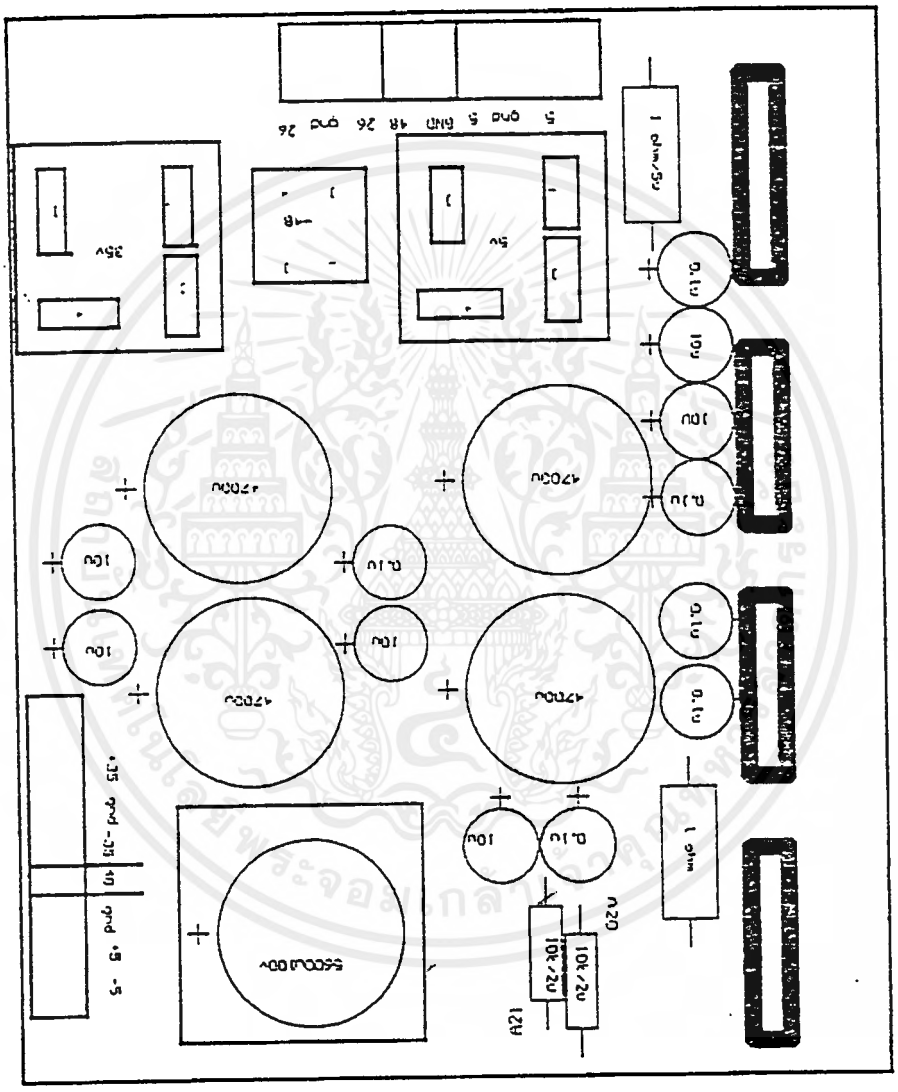
รูปที่ 3.8.15 BOTTOM : **SIDE** : TOT LINE **BOARD**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง



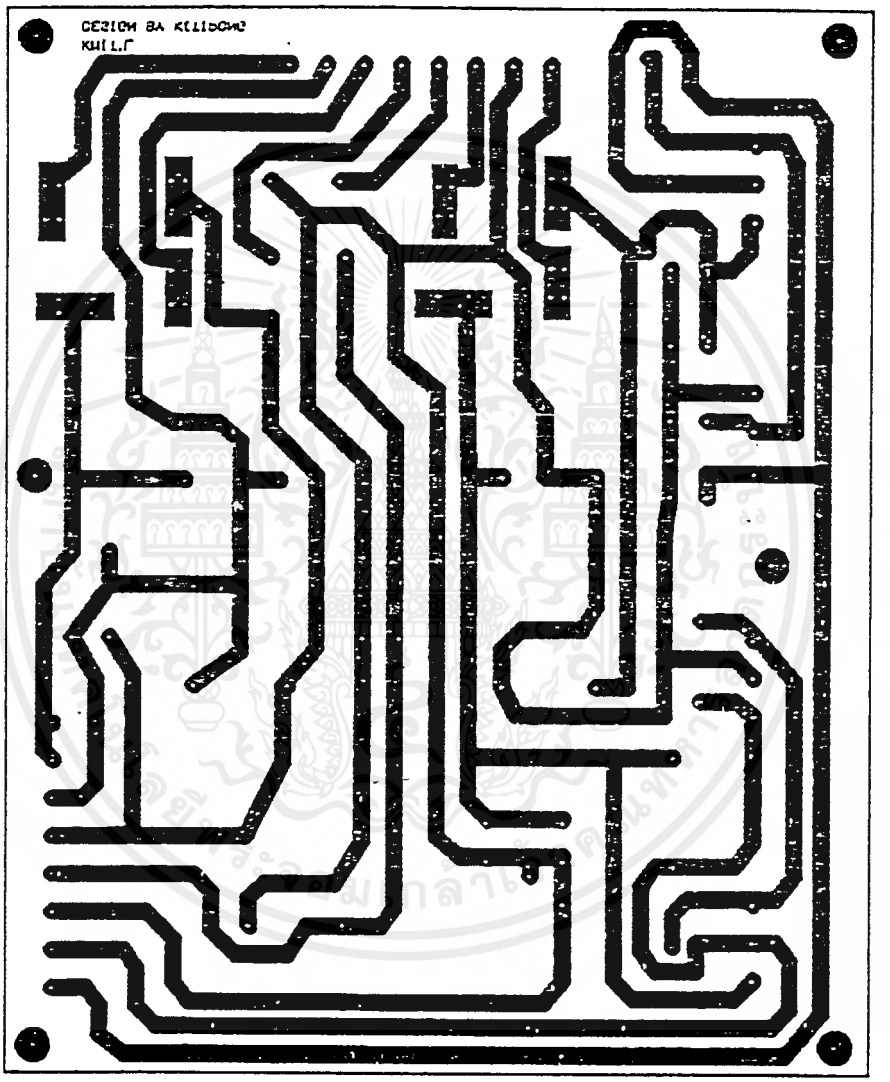
รูปที่ 3.8.16 TOP SIDE : TOT LINE BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8.17 TOP OVERLAY POWER SUPPLY **BOARD**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8.18 BOTTOM **SIDE** POWER SUPPLY **BOARD**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

บทที่ 4
สรุปผลและวิจารณ์

ในการจัดสร้างชุมสาย PABX ทางคณะของผู้จัดทำได้ทำการทดลอง และการวางแบบ การวางอุปกรณ์ของโครงงานนี้ ซึ่งใช้การแบ่งออกเป็นภาค ๆ โดยลงในแต่ละบอร์ด เพื่อจะได้ง่ายต่อการตรวจเช็คและตรวจซ่อม หลังจากนั้นจึงได้ทำการหาอุปกรณ์ที่ใช้ในโครงงานมาทำการทดลองแล้วจึงออกแบบจริง ซึ่งในขั้นตอนการทดลอง และออกแบบนั้นใช้ระยะเวลามาก และโครงงานนี้เป็นโครงงานที่มีทั้ง Hard ware และ Soft ware ที่สลับซับซ้อนหลายภาคที่ต้องนำมาต่อเข้าด้วยกัน จึงอาจมีบางส่วนที่ไม่สมบูรณ์มากนัก

<u>ปัญหาส่วนของ Hardware</u>

1. ความถี่ที่ผลิตขึ้นโดย IC 555 นั้นสามารถที่ออกมารบกวนระบบได้ก็คือ ความถี่ Busy, Dial, Ring back tone ซึ่งจะเป็นผลให้สัญญาณที่ได้มีเสียงแทรก การแก้ไขได้โดยใช้ Capaciter ค่าสูง ๆ มาต่อคร่อม Supply และต่อที่แหล่งจ่ายของ IC ทุก ๆ ตัวในวงจร

2. CPU จะต้องไม่มีสัญญาณรบกวน รวมทั้ง Supply ที่จ่ายให้ CPU จะต้องเรียบสม่ำเสมอ มิฉะนั้น CPU จะ Reset ตัวเองขณะที่ทำงาน

3. การทำ Print จะต้องให้มีขนาดของลายทองแดงใหญ่พอสมควร เพราะจะทำให้สัญญาณผ่านได้สะดวก

4. Slot ซึ่งใช้เสียบ Grad แต่ละ Grad จะต้องมีขนาดซึ่งพอดี ไม่เช่นนั้นจะทำให้เกิดการผิดพลาดของสัญญาณ อาจเกิดการช็อตได้

<u>ปัญหาส่วนของ Software</u>

เนื่องจากการพัฒนา Softwaare ในระบบ PABX จะต้องมีการทำงานแบบ Multitasking เพราะว่าจะต้องมีการตรวจสอบ Status ของสายภายในและภายนอกในเวลาเดียวกัน ซึ่งจากการค้นหาข้อมูลจาก Project รุ่นก่อน ๆ ทางด้านโปรแกรมของ PABX จะไม่สามารถให้คู่สายภายในและภายนอกทำงานในเวลาเดียวกันจึงได้มีการพยายามพัฒนาโปรแกรมให้มีความสามารถตรวจสอบสายภายในและภายนอกในเวลาเดียวกัน ซึ่งมีความสลับซับซ้อนมาก จำเป็นต้องเก็บสถานะตลอด ซึ่งจากการเขียนโปรแกรมจึงมีการ Error เกิดขึ้น จึงไม่สามารถทำให้โปรแกรมทำงานได้

คณะผู้จัดทำหวังเป็นอย่างยิ่งว่า โครงงานนี้จะเป็นประโยชน์ต่อผู้สนใจที่จะนำไปใช้งานจริง หรือเป็นแนวทางในการพัฒนาเทคโนโลยีทางด้านนี้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเอกสารฉบับนี้ทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. "LINE INTERFACE HANDBOOK" .MITEL, MITEL COPORATION. 1994

2."MICROELECTRONICS COMMUNICATION HANDBOOK" .MITEL,
MITEL COPORATION. 1994

3. "MCS-51 MICROCONTROLLER HANDBOOK" .INTEL,
INTEL COPORATION. 2537

4. "CMOS DATABOOLK" .MOTOROLA, พิมพ์ที่ หจก. เอชเอ็น การพิมพ์. 2532

5. "TTL DATABOOK" .MOTOROLA, พิมพ์ที่ หจก. เอชเอ็น การพิมพ์. 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก. การใช้เครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

การใช้เครื่อง

การติดตั้ง

ต่อเครื่องโทรศัพท์ภายในเข้ากับ Internal Line 11-26

ต่อคู่สายขององค์การโทรศัพท์เข้ากับ Extrenal Line 1-2-3

การเรียกภายใน

1. ยกหูจะได้ยินสัญญาณให้หมุน
2. กดเลขหมายภายใน 11-26 ที่ต้องการต่อ
3. ถ้าเลขหมายปลายทางว่าง จะได้รับสัญญาณเรียกกลับ รอจนมีผู้รับ
4. ถ้าเลขหมายปลายทางไม่ว่าง จะได้รับสัญญาณไม่ว่าง วางหูแล้วทำการเรียกใหม่

การเรียกภายนอก

1. ยกหูแล้วกดปุ่ม "9" จะได้รับสัญญาณ Dial tone จากชุมสายขององค์การโทรศัพท์
2. กดเลขหมายที่ต้องการติดต่อ
3. ถ้าเลขหมายปลายทางว่าง จะได้รับสัญญาณเรียกกลับ รอจนมีผู้รับ
4. ถ้าเลขหมายปลายทางไม่ว่าง จะได้รับสัญญาณไม่ว่าง วางหูแล้วทำการเรียกใหม่

การรับสายจากภายนอก

1. ชุมสาย PABX จะทำการรับสายโดยอัตโนมัติ เมื่อมีสัญญาณกระดิ่งเรียกเข้ามา

2. เมื่อมีการเรียกเข้าจากภายนอก จะมีระบบ Voice Playback Device แจ้งให้สายจากภายนอกทราบว่า "ระบบนี้เป็นระบบอัตโนมัติกรุณากดเลขหมายที่ท่านต้องการต่อเข้า"

3. คู่สายภายนอกต้องทำการกดปุ่มเลขหมาย 11-26 ชุมสาย PABX ก็จะทำการต่อเข้ากับเลขหมายภายในโดยอัตโนมัติ โดยจะมีสัญญาณกระดิ่งไปเรียกผู้รับยังเลขหมายภายในที่ถูกเรียก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข.โปรแกรมการทำงานของระบบ PABX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

```
;*************************************************************************
;**  PROGRAM :  PABX CONTROL FOR 3 TOT LINE AND 16 SUBCRIBER             *
;**  PROGRAMER: Mr.THAMMASAK YOUNGNOI                                    *
;**  HARDWARE:  PABX 3 TO 16 LINE BY 3U COMPLATE IN OCT 31,96            *
;*************************************************************************

;***************************[ DATA EQU DEFINE ]*****************************

COMMAND         EQU     0E060H ; Read-Write Register
READBUSY        EQU     0E061H ; Read BF(Busy Flag) and address
WRITEDATA       EQU     0E062H.; Write character
READDATA        EQU     0E063H ; Read Data from DD ram
ADDR_DATA       EQU     00000H ; DATA DISPLAY IN RAM

INTERF_A        EQU     0E000H
INTERF_B        EQU     0E001H
INTERF_C        EQU     0E002H
INTERF_COM      EQU     0E003H
SUB1_A          EQU     0E004H
SUB1_B          EQU     0E005H
SUB1_C          EQU     0E006H
SUB1_COM        EQU     0E007H
SUB2_A          EQU     0E008H
SUB2_B          EQU     0E009H
SUB2_C          EQU     0E00AH
SUB2_COM        EQU     0E00BH
TOT1_A          EQU     0E00CH
TOT1_B          EQU     0E00DH
TOT1_C          EQU     0E00EH
TOT1_COM        EQU     0E00FH
TOT2_A          EQU     0E010H
TOT2_B          EQU     0E011H
TOT2_C          EQU     0E012H
TOT2_COM        EQU     0E013H
SW1_A           EQU     0E014H
SW1_B           EQU     0E015H
SW1_C           EQU     0E016H
SW1_COM         EQU     0E017H
SW2_A           EQU     0E018H
SW2_B           EQU     0E019H
SW2_C           EQU     0E01AH
SW2_COM         EQU     0E01BH
SW3_A           EQU     0E01CH
SW3_B           EQU     0E01DH
SW3_C           EQU     0E01EH
SW3_COM         EQU     0E01FH

;******[ STATUS  BIT FLAG ]************

POWER           EQU     7FH     ; INITIAL POWER FLAG
RUN             EQU     7EH     ; RUN MODE FLAG
VOICE_FLAG      EQU     7DH     ; FLAG FOR TIME COUNTER
TOT1_FLAG       EQU     7CH     ; FLAG FOR TIME COUNTER
TOT2_FLAG       EQU     7BH     ; FLAG FOR TIME COUNTER
TOT3_FLAG       EQU     7AH     ; FLAG FOR TIME COUNTER
TOTF1           EQU     79H     ; FLAG CHANEL TOT
TOTF2           EQU     78H     ; FLAG CHANEL TOT
TOTF3           EQU     77H     ; FLAG CHANEL TOT

;***************[ INTERNAL MEMORY ]*****************
DPL0            EQU     30H     ; DPL BACKUP NUMBER 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DPH0            EQU     31H     ; DPH BACKUP NUMBER 0
DPL1            EQU     32H     ; DPL BACKUP NUMBAE 1
DPH1            EQU     33H     ; DPH BACKUP NUMBER 1
SCAN            EQU     34H     ; TIMER COUNTER
CHL             EQU     35H     ; LOW ASCII FOR HEX2ASCII
CHH             EQU     36H     ; HI ASCII FOR HEX2ASCII
CLOCK           EQU     37H     ; CLOCK COUNTER
KEY             EQU     38H     ; INPUT KEY BIT 4 SET IF HAS INPUT KEY
ADDR_SW         EQU     39H     ; SWITCH POINTER IN MAP SWITCH
MODE            EQU     3AH     ; MODE OPERATION SET BY KEYBOARD
TOT1_RGCT       EQU     3BH     ; RINGING COUNTER FOR TOT CHANAL 1
TOT2_RGCT       EQU     3CH     ; RINGING COUNTER FOR TOT CHANAL 2
TOT3_RGCT       EQU     3DH     ; RINGING COUNTER FOR TOT CHANAL 3
TOT1_COUNT      EQU     3EH     ; TIME COUNTER FOR TOT1
TOT2_COUNT      EQU     3FH     ; TIME COUNTER FOR TOT1
TOT3_COUNT      EQU     40H     ; TIME COUNTER FOR TOT1
VOICE_COUNT     EQU     41H     ; TIME COUNTER FOR VOICE
DTMF_FLAG       EQU     42H     ; 42H TO 44H = 3 BYTE FLAG FOR DTMF KEY
; TO            EQU     44H
DTMF_DATA       EQU     45H     ; 45H TO 4FH = 11 BYTE DTMF DATA
; TO            EQU     4FH
HOOK            EQU     50H     ; 50H TO 51H = 2 BYTE HOOK FOR 16 EXT
; TO            EQU     51H
DTMF_BAK        EQU     52H
; TO            EQU     53H
DTMF_USE        EQU     54H
; TO            EQU     55H

;/----------------------[ MAP SWITCH DATA ADDR 60H TO 7FH ]-----------------\
MAP_SW          EQU     60H     ; DATA FOR MAP SWITCH ADDR 60H TO 7FH = 32 BYTE
; TO            EQU     7FH

;*****************[ USER DATA ]**********************
MAP_EXT01       EQU     00H
MAP_EXT02       EQU     01H
MAP_EXT03       EQU     02H
MAP_EXT04       EQU     03H
MAP_EXT05       EQU     04H
MAP_EXT06       EQU     05H
MAP_EXT07       EQU     06H
MAP_EXT08       EQU     07H
MAP_EXT09       EQU     08H
MAP_EXT10       EQU     09H
MAP_EXT11       EQU     0AH
MAP_EXT12       EQU     0BH
MAP_EXT13       EQU     0CH
MAP_EXT14       EQU     0DH
MAP_EXT15       EQU     0EH
MAP_EXT16       EQU     0FH
MAP_TOT1        EQU     00H
MAP_TOT2        EQU     10H
MAP_TOT3        EQU     20H
MAP_DIAL        EQU     30H
MAP_RING_BACK   EQU     40H
MAP_BUSY        EQU     50H
MAP_VOICE       EQU     60H
MAP_CH1         EQU     80H
MAP_CH2         EQU     90H
MAP_CH3         EQU     A0H
MAP_CH4         EQU     B0H
MAP_CH5         EQU     C0H
MAP_CH6         EQU     D0H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MAP_CH7         EQU     E0H
MAP_CH8         EQU     F0H

;*****************************************************************************
;**                          [ INTERRUPT VECTOR ]                           **
;*****************************************************************************

                ORG     0000H
                JMP     ST

                ORG     0003H
                JMP     INT_IE0

                ORG     000BH
                RETI

                ORG     0013H
                RETI

                ORG     001BH
                RETI

                ORG     0023H
                RETI

                ORG     002BH
                JMP     INT_EXT

                ORG 0030H

;*****************************************************************************
;**                      [ INITERRUPT SERVICE ROUTINE ]                     **
;*****************************************************************************

;*****************************************************************************
;**                        [ INTERRUPT IE0 ]                                **
;**                      INTERUP FOR SYSTEM                                 **
;*****************************************************************************

INT_IE0:
                CLR     IE.7
                PUSH    DPL
                PUSH    DPH
                PUSH    A
                PUSH    B

                MOV     DPTR,#INTERF_A
                MOVX    A,@DPTR
                JB      A.3,TOT_CARD1
                JB      A.4,TOT_CARD2
                JB      A.7,KEYBOARD
                JMP     INT_IE0_END

TOT_CARD1:
                MOV     DPTR,#TOT1_C
                MOVX    A,@DPTR
                MOV     B,A
                ANL     A,#0FH
                CJNE    A,#0,TOT_CH1
                MOV     A,B
                SWAP    A
                ANL     A,#0FH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CJNE    A,#0,TOT_CH2
            JMP     INT_IE0_END
TOT_CARD2:
            MOV     DPTR,#TOT2_C
            MOVX    A,@DPTR
            MOV     B,A
            ANL     A,#0FH
            CJNE    A,#0,TOT_CH3
            JMP     INT_IE0_END
TOT_CH1:
            SETB    A.7
            MOV     TOT1_RGCT,A
            JMP     INT_IE0_END
TOT_CH2:
            SETB    A.7
            MOV     TOT2_RGCT,A
            JMP     INT_IE0_END
TOT_CH3:
            SETB    A.7
            MOV     TOT3_RGCT,A
            JMP     INT_IE0_END
KEYBOARD:

            MOV     DPTR,#INTERF_C
            MOVX    A,@DPTR
            SETB    A.7
            MOV     KEY,A
            JMP     INT_IE0_END

INT_IE0_END:

            MOV     DPTR,#INTERF_B          ;
            MOV     A,#00                   ;
            MOVX    @DPTR,A                 ;  RESET INTERRUPT
            MOV     A,#FFH                  ;
            MOVX    @DPTR,A                 ;


            POP     B
            POP     A
            POP     DPH
            POP     DPL
            SETB    IE.7

            RETI            ;

;*************************************************************************
;**                        [ INTERRUPT EXT ]                            *
;**              TIME COUNTER FOR SYSTEM  &  SCAN DISPLAY               *
;*************************************************************************

INT_EXT:
            CLR     IE.7
            PUSH    DPL
            PUSH    DPH
            PUSH    A

            DB      C2H,CFH    ; <= CLR T2CON.7
            INC     SCAN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     A,SCAN
        CJNE    A,#8H,EXT00                     ; 1 TURN = 1/16 SEC
        INC     CLOCK

        MOV     DPTR,#ADDR_DATA             ;
        MOV     A,CLOCK                     ;
        LCALL   HEX2ASCII                   ;
        MOV     A,CHH                       ; PRINT CLOCK
        MOVX    @DPTR,A                     ;
        INC     DPTR                        ;
        MOV     A,CHL                       ;
        MOVX    @DPTR,A                     ;

        LCALL   PRINTF                      ; PRINT DISPLAY
        MOV     SCAN,#00                    ; RESTART SCAN COUNTER

;       LCALL     SAND_BUSY                   ;
;
;       LCALL     SAND_DIAL                   ;   SAND SIGNAL
;
;       LCALL     SAND_RING_BACK              ;
;       LCALL     SAND_RINGING                ;


EXT00:  POP     A
        POP     DPH
        POP     DPL
        SETB    IE.7
        RETI

;-------------------------------------------------------------------------

;*****************************[ START PROGRAM ]*********************************
;**      *******     *********       ***     ********      *********  *
;**     ***   ***    *********     *******   *********     *********  *
;**     ****          ***       **     **   **     **        ***      *
;**      *******      ***       *********   ********        ***      *
;**          ****     ***       **     **   *********       ***      *
;**     ***   ***     ***       **     **   **     **       ***      *
;**      *******      ***       **     **   **     **       ***      *
;***************************************************************************

ST:

        MOV     R0,#FFH             ;
MAN0:   MOV     R1,#FFH             ;
MAN1:   DJNZ    R1,MAN1             ;
        DJNZ    R0,MAN0             ;


        MOV     DPTR,#MENU00
        LCALL   INIT_DATA_DIS

        LCALL   INIT_PORT

        LCALL   INIT_DATA

        LCALL   INIT                            ; INITAL DATA DISPLAY

        LCALL   INIT_INT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL       CLR_ALL_MAP_SW

            LCALL       CLRSCR
            LCALL       PRINTF
READ_KEY:
            LCALL       PRINT_DTMF                      ;  PRINT DTMF TO DISPLAY

            ;LCALL       PRINTF
            ;********************* PROGRAM RUN *****************
            LCALL       HOOK_UPDATE                     ; PREPARE HOOK

            ;LCALL       PRINT_DIAL
            LCALL       PRINT_HOOK
            ;LCALL       PRINT_BHOOK
            ;LCALL       PRINT_BUSY
            ;LCALL       PRINT_RINB


            LCALL       DTMF_STATUS                     ; PREPARE DTMF STATUS
            LCALL       DTMF_SERVICE                    ; PREPARE DTMF KEY

            ;LCALL       TRAFFIC                         ; TRAFFIC CONTROL


            JMP         READ_KEY            ; GOTO LOOP

;*************************************************************************
;**                          [ PRINT A TO DISPLAY HOT PRINT ]             *
;*************************************************************************
HPRINT:
            PUSH        DPH
            PUSH        DPL

            MOV         DPTR,#ADDR_DATA+19
            MOVX        @DPTR,A

            POP         DPL
            POP         DPH

;*************************************************************************
;**                          [ PRINT DTMF TO DISPLAY ]                    **
;*************************************************************************
PRINT_DTMF:
            PUSH        0
            PUSH        1
            PUSH        2
            PUSH        DPH
            PUSH        DPL

            MOV         R0,#8
            MOV         R2,DTMF_BAK
            MOV         DPTR,#ADDR_DATA+25
PD_LOOP:
            MOV         A,R0
            DEC         A
            CPL         A
            ANL         A,#7
            MOV         A,R1

            MOV         A,R2
            JB          A.0,PD_READ_KEY
            MOV         A,#'_'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOVX        @DPTR,A
            JMP         PD_NEXT_STD
PD_READ_KEY:
            MOV         A,R1
            LCALL       DTMF_PORT
            LCALL       HEX2ASCII
            MOV         A,CHL
            MOVX        @DPTR,A
PD_NEXT_STD:
            INC         DPTR
            MOV         A,R2
            RR          A
            MOV         R2,A
            DJNZ        R0,PD_LOOP


            MOV         R0,#3
            MOV         R2,DTMF_BAK+1
PD1_LOOP:
            MOV         A,R0
            DEC         A
            CPL         A
            ANL         A,#7
            ADD         A,#8
            MOV         A,R1

            MOV         A,R2
            JB          A.0,PD1_READ_KEY
            MOV         A,#'_'
            MOVX        @DPTR,A
            JMP         PD1_NEXT_STD
PD1_READ_KEY:
            MOV         A,R1
            LCALL       DTMF_PORT
            LCALL       HEX2ASCII
            MOV         A,CHL
            MOVX        @DPTR,A
PD1_NEXT_STD:
            INC         DPTR
            MOV         A,R2
            RR          A
            MOV         R2,A
            DJNZ        R0,PD1_LOOP



            POP         DPL
            POP         DPH
            POP         2
            POP         1
            POP         0
            RET

;*************************************************************************
;**                          [ PRINT HOOK TO DISPLAY ]                   **
;*************************************************************************
PRINT_HOOK:
            PUSH        0
            PUSH        1
            PUSH        2
            PUSH        DPH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            PUSH        DPL

            MOV         R0,#8
            MOV         R2,HOOK
            MOV         DPTR,#ADDR_DATA+3
PH_LOOP:
            MOV         A,R2
            JB          A.0,PH_READ_KEY
            MOV         A,#'_'
            MOVX        @DPTR,A
            JMP         PH_NEXT_STD
PH_READ_KEY:

            MOV         A,#'1'
            MOVX        @DPTR,A
PH_NEXT_STD:
            INC         DPTR
            MOV         A,R2
            RR          A
            MOV         R2,A
            DJNZ        R0,PH_LOOP


            MOV         R0,#8
            MOV         R2,HOOK+1
PH1_LOOP:
            MOV         A,R2
            JB          A.0,PH1_READ_KEY
            MOV         A,#'_'
            MOVX        @DPTR,A
            JMP         PH1_NEXT_STD
PH1_READ_KEY:
            MOV         A,#'1'
            MOVX         @DPTR,A
PH1_NEXT_STD:
            INC         DPTR
            MOV         A,R2
            RR          A
            MOV         R2,A
            DJNZ        R0,PH1_LOOP

            POP         DPL
            POP         DPH
            POP         2
            POP         1
            POP         0
            RET

;*****************************************************************************
;**                          [ PRINT DIAL TO DISPLAY ]                        **
;*****************************************************************************
PRINT_DIAL:
            PUSH        0
            PUSH        2
            PUSH        DPH
            PUSH        DPL

            MOV         R0,#16
            MOV         DPTR,#ADDR_DATA+23
PDA_LOOP:
            MOV         A,R0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          DEC       A
          MOV       R2,A

          LCALL     EXT_DIAL

          JB        A.0,PDA_READ_KEY
          MOV       A,#'_'
          MOVX      @DPTR,A
          JMP       PDA_NEXT_STD
PDA_READ_KEY:
          MOV       A,#'1'
          MOVX      @DPTR,A
PDA_NEXT_STD:

          INC       DPTR

          DJNZ      R0,PDA_LOOP

          POP       DPL
          POP       DPH
          POP       2
          POP       0
          RET

;*****************************************************************************
;**                          [ PRINT BUSY TO DISPLAY ]                        **
;*****************************************************************************
PRINT_BUSY:
          PUSH      0
          PUSH      2
          PUSH      DPH
          PUSH      DPL

          MOV       R0,#16
          MOV       DPTR,#ADDR_DATA+23
PBS_LOOP:
          MOV       A,R0
          DEC       A
          MOV       R2,A

          LCALL     EXT_BUSY

          JB        A.0,PBS_READ_KEY
          MOV       A,#'_'
          MOVX      @DPTR,A
          JMP       PBS_NEXT_STD
PBS_READ_KEY:
          MOV       A,#'1'
          MOVX      @DPTR,A
PBS_NEXT_STD:

          INC       DPTR

          DJNZ      R0,PBS_LOOP

          POP       DPL
          POP       DPH
          POP       2
          POP       0
          RET

;*****************************************************************************
```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;**                          [ PRINT:RING BACK TO DISPLAY ]
**
;*****************************************************************************
PRINT_RINB:
            PUSH        0
            PUSH        2
            PUSH        DPH
            PUSH        DPL

            MOV         R0,#16
            MOV         DPTR,#ADDR_DATA+23
PRB_LOOP:
            MOV         A,R0
            DEC         A
            MOV         R2,A

            LCALL       EXT_BUSY

            JB          A.0,PRB_READ_KEY
            MOV         A,#'_'
            MOVX        @DPTR,A
            JMP         PRB_NEXT_STD
PRB_READ_KEY:
            MOV         A,#'1'
            MOVX        @DPTR,A
PRB_NEXT_STD:

            INC         DPTR

            DJNZ        R0,PRB_LOOP

            POP         DPL
            POP         DPH
            POP         2
            POP         0
            RET

;*****************************************************************************
;**                          [ PRINT HOOKB TO DISPLAY ]                      *
;*****************************************************************************
PRINT_BHOOK:
            PUSH        0
            PUSH        2
            PUSH        DPH
            PUSH        DPL

            MOV         R0,#16
            MOV         DPTR,#ADDR_DATA+23
PHB_LOOP:
            MOV         A,R0
            DEC         A
            MOV         R2,A

            LCALL       EXT_HOOK

            JB          A.0,PHB_READ_KEY
            MOV         A,#'_'
            MOVX        @DPTR,A
            JMP         PHB_NEXT_STD
PHB_READ_KEY:
            MOV         A,#'1'
            MOVX        @DPTR,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PHB_NEXT_STD:

          INC       DPTR

          DJNZ      R0,PHB_LOOP

          POP       DPL
          POP       DPH
          POP       2
          POP       0
          RET


;*****************************************************************************
;**                          [ TRAFFIC CONTROL ]                             *
;*****************************************************************************
TRAFFIC:
          PUSH      A
          PUSH      B
          PUSH      0
          PUSH      1

          MOV       R0,#16
TF_NEXT_CH:
          MOV       A,R0
          DEC       A
          CPL       A
          ANL       A,#0FH
          MOV       R1,A                    ; R1 = CHANEL TO PROCESSS
          MOV       B,#0
          JMP       TF_ST

TF_LOOP_CHK:
          DJNZ      R0,TF_NEXT_CH
          JMP       TF_END

TF_ST:
         LCALL      EXT_ASSIGN
         CJNE       A,#1,TF_NO_ASSIGN

         MOV        A,R1
         LCALL      EXT_MEDIA
         LCALL      DTMF_READ_FLAG
         ANL        A,#00000011B
TF_NO_ASSIGN:
          ORL       B,A

          MOV       A,B                     ;
          RL        A                       ; SHIFT REG.B   <- 1BIT
          MOV       B,A                     ;

;          MOV       A,R1
;          LCALL     EXT_DTMF_ENABLE
;          ANL       A,#00000001B
;          ORL       B,A
;
;          MOV       A,B                     ;
;          RL        A                       ; SHIFT REG.B   <- 1BIT
;          MOV       B,A                     ;
;
;          MOV       A,R1
;          LCALL     EXT_DIAL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;           ANL         A,#00000001B
;           ORL         B,A
;
;           MOV         A,B                     ;
;           RL          A                       ; SHIFT REG.B  <- 1BIT
;           MOV         B,A                     ;
;
;           MOV         A,R1
;           LCALL       EXT_RING_BACK
;           ANL         A,#00000001B
;           ORL         B,A
;
;           MOV         A,B                     ;
;           RL          A                       ; SHIFT REG.B  <- 1BIT
;           MOV         B,A                     ;
;
;           MOV         A,R1
;           LCALL       EXT_RINGING
;           ANL         A,#00000001B
;           ORL         B,A
;
;           MOV         A,B                     ;
;           RL          A                       ; SHIFT REG.B  <- 1BIT
;           MOV         B,A                     ;
;
;           MOV         A,R1
;           LCALL       EXT_HOOK
;           ANL         A,#00000001B
;           ORL         B,A
;
;           MOV         A,B                     ;
;           RL          A                       ; SHIFT REG.B  <- 1BIT
;           MOV         B,A                     ;
;
;           MOV         A,R1
;           LCALL       EXT_LINK
;           ANL         A,#00000001B
;           ORL         B,A

          MOV         A,R1
          LCALL       HOOK_STATUS
          JB          A.0,TF_ON_HOOK
          JB          A.1,TF_OFF_HOOK

                                                ;---- HOOK N/A --------
;           MOV         A,B
;           ANL         A,#11100011B
;           CJNE        A,#11100010B,TF_ON_TOT
;           MOV         A,R1
;           LCALL       CHANEL_CALL
;           JMP         TF_MODE_END
;TF_ON_TOT:
;           MOV         A,B
;           ANL         A,#11100011B
;           CJNE        A,#01100010B,TF_MODE_END
;           MOV         A,R1
;           LCALL       CALL_ON_TOT
          JMP         TF_MODE_END

TF_ON_HOOK:                                     ;----- HOOK ON ----------
          MOV         A,#'N'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL       HPRINT
            LCALL       PRINTF


            MOV         A,B
            ANL         A,#00000000B
            CJNE        A,#00000000B,TF_MODE_CRP
            MOV         A,R1
            LCALL       ON_CHANEL
            JMP         TF_MODE_END

TF_MODE_CRP:

;           MOV         A,B
;           ANL         A,#00000111B
;           CJNE        A,#00000100B,TF_MODE_END
;           MOV         A,R1
;           LCALL       CHANEL_RESPONE
            JMP         TF_MODE_END

TF_OFF_HOOK:                                        ;-------- HOOK OFF -------

            MOV         A,#'F'
            LCALL       HPRINT
            LCALL       PRINTF

            MOV         A,B
            ANL         A,#00000000B
            CJNE        A,#00000000B,TF_MODE_DC
            MOV         A,R1
            LCALL       OFF_CHANEL
            JMP         TF_MODE_END

TF_MODE_DC:
;           MOV         A,B
;           ANL         A,#00000011B
;           CJNE        A,#00000011B,TF_MODE_END
;           MOV         A,R1
;           LCALL       DELETE_CALL

TF_MODE_END:
            JMP         TF_LOOP_CHK
TF_END:

            POP         1
            POP         0
            POP         B
            POP         A
            RET

;*************************************************************************
;**                    [ DELETE CALL CHANEL FROM PATH ]                 **
;*************************************************************************
DELETE_CALL:

            PUSH        0
            PUSH        1

            MOV         R0,A
            ;************** CLEAR CHANEL CALL *************
            SETB        A.6
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL       EXT_HOOK
            MOV         A,R0
            SETB        A.6
            LCALL       EXT_LINK
            MOV         A,R0
            SETB        A.6
            LCALL       EXT_ASSIGN

            MOV         A,R0
            LCALL       EXT_MEDIA

            SETB        A.3
            SWAP        A
            ORL         A,R0
            MOV         ADDR_SW,A
            LCALL       CLR_MAP_SW
            ;************** CLEAR CHANEL TO CALL **********
            MOV         A,R0
            LCALL       EXT_CALL
            MOV         R1,A
            LCALL       EXT_MEDIA

            SETB        A.3
            SWAP        A
            ORL         A,R1
            MOV         ADDR_SW,A
            LCALL       CLR_MAP_SW

            POP         1
            POP         0
            RET

;*****************************************************************************
;**                          [ CHANEL CALL ]                                 **
;*****************************************************************************
CHANEL_CALL:

            PUSH        0
            PUSH        1
            PUSH        2
            PUSH        3
            PUSH        4

            MOV         R0,A
            LCALL       EXT_MEDIA
            MOV         R4,A                    ;R4 = MEDIA TO USE
            MOV         R1,#DTMF_DATA
            ADD         A,R1
            MOV         R1,A
            MOV         A,@R1
            MOV         R1,A                    ; R1 = DTMF DATA FOR CHANEL

            MOV         R2,#16
CHC_LOOP:
            MOV         A,R2
            DEC         A
            MOV         R3,A
            LCALL       EXT_NUMBER
            CJNE        A,1,CHC_NEXT
            JMP         CHC_CALL
CHC_NEXT:
            DJNZ        R2,CHC_LOOP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CHC_FAIL:
        MOV     A,R0
        SETB    A.7
        LCALL   EXT_BUSY
        MOV     A,R0
        SETB    A.7
        LCALL   EXT_HOOK

        JMP     CHC_END
CHC_CALL:
        MOV     A,R3            ;
        ANL     A,#0FH          ;  CHECK CHANEL NON USE
        LCALL   EXT_HOOK        ;
        CJNE    A,#0,CHC_FAIL   ;

        ;******************** SET CHANEL TO CALL ************
        MOV     A,R3
        SETB    A.7
        LCALL   EXT_RINGING                         ;SET RINGING
        MOV     A,R4
        SWAP    A
        ANL     A,#70H
        MOV     R2,A

        MOV     A,R3
        ORL     A,R2
        SETB    A.7
        LCALL   EXT_MEDIA
        ;********************** SET CHANEL CALL ************
        MOV     A,R0
        SETB    A.6
        LCALL   EXT_DTMF_ENABLE
        MOV     A,R0
        SETB    A.7
        LCALL   EXT_RING_BACK
        MOV     A,R0
        SETB    A.7
        LCALL   EXT_LINK
        MOV     A,R4
        LCALL   DTMF_WRITE_FLAG

        ;***************** CLOSE SW ***************
        MOV     A,R0
        ANL     A,#0FH
        MOV     R0,A
        LCALL   EXT_MEDIA

        SETB    A.3
        SWAP    A
        ORL     A,R0
        MOV     ADDR_SW,A
        LCALL   SET_MAP_SW

        MOV     A,R3
        MOV     R1,A
        MOV     A,R0
        SETB    A.7
        LCALL   EXT_CALL

CHC_END:
        POP     4
        POP     3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            POP         2
            POP         1
            POP         0
            RET

;*****************************************************************************
;**                              [ CALL ON TOT ]                            **
;*****************************************************************************
CALL_ON_TOT:

            PUSH        0
            PUSH        1
            PUSH        2
            PUSH        3
            PUSH        4

            MOV         R0,A
            LCALL       EXT_MEDIA
            MOV         R4,A                    ;R4 = MEDIA TO USE
            MOV         R1,#DTMF_DATA
            ADD         A,R1
            MOV         R1,A
            MOV         A,@R1
            MOV         R1,A                    ; R1 = DTMF DATA FOR CHANEL

            CJNE        A,#09,COT_END
            JNB         TOTF1,USE_TOT1
            JNB         TOTF2,USE_TOT2
            JNB         TOTF3,USE_TOT3
                                                ; TOT FAIL---------------
            MOV         A,R0                    ;
            SETB        A.7                     ;
            LCALL       EXT_BUSY                ;
            JMP         COT_END                 ;
USE_TOT1:
            SETB        TOTF1
            ;**************** CLOSE SW ***************
            MOV         A,#MAP_TOT1
            ORL         A,R0
            MOV         ADDR_SW,A
            LCALL       SET_MAP_SW
            JMP         USE_TOT_CH
USE_TOT2:
            SETB        TOTF2
            ;**************** CLOSE SW ***************
            MOV         A,#MAP_TOT2
            ORL         A,R0
            MOV         ADDR_SW,A
            LCALL       SET_MAP_SW
            JMP         USE_TOT_CH
USE_TOT3:
            SETB        TOTF3
            ;**************** CLOSE SW ***************
            MOV         A,#MAP_TOT3
            ORL         A,R0
            MOV         ADDR_SW,A
            LCALL       SET_MAP_SW
            JMP         USE_TOT_CH
USE_TOT_CH:
            ;********************** SET CHANEL CALL ***********
            MOV         A,R0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            SETB        A.6
            LCALL       EXT_DTMF_ENABLE
            MOV         A,R0
            SETB        A.7
            LCALL       EXT_LINK
            MOV         A,R4
            LCALL       DTMF_WRITE_FLAG
COT_END:
            POP         4
            POP         3
            POP         2
            POP         1
            POP         0
            RET

;*************************************************************************
;**                          [ CHANEL OFF NON CALL ]                     **
;*************************************************************************
OFF_CHANEL:
            PUSH        0

            CLR         A.7
            SETB        A.6
            MOV         R0,A

            LCALL       EXT_HOOK
            MOV         A,R0
            LCALL       EXT_DIAL
            MOV         A,R0
            LCALL       EXT_DTMF_ENABLE
            MOV         A,R0
            LCALL       EXT_SHIFT
            MOV         A,R0
            LCALL       EXT_ASSIGN
            MOV         A,R0
            LCALL       EXT_BUSY

            POP         0
            RET

;*************************************************************************
;**                          [ CHANEL ON TO START ]                      **
;*************************************************************************
ON_CHANEL:

            PUSH        0

            ;********************* RESET CONSTANT **************

            ANL         A,#0FH
            SETB        A.6
            MOV         R0,A
            LCALL       EXT_BUSY
            MOV         A,R0
            LCALL       EXT_RINGING
            MOV         A,R0
            LCALL       EXT_RING_BACK
            MOV         A,R0
            LCALL       EXT_LINK

            ;******************** SET CONSTANT ************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     A,R0
        CLR     A.6
        SETB    A.7
        MOV     R0,A
        LCALL   EXT_HOOK
        MOV     A,R0
        LCALL   EXT_DIAL
        MOV     A,R0
        LCALL   EXT_DTMF_ENABLE
        MOV     A,R0
        LCALL   EXT_SHIFT
        MOV     A,R0
        LCALL   EXT_ASSIGN

        ;***************** RESET DTMF FLAG ***************

        MOV     A,R0
        ANL     A,#0FH
        LCALL   DTMF_WRITE_FLAG

        ;****************** ASSIGN MEDIA TO EXT **************

        MOV     A,R0
        ANL     A,#0FH
        MOV     R0,A

        LCALL   MEDIA_ASSIGN
        ANL     A,#7
        SWAP    A
        ORL     A,R0
        SETB    A.7
        LCALL   EXT_MEDIA

        POP     0
        RET

;*************************************************************************
;**                    [ CHANEL RESPONE TO CHANEL CALL ]                **
;*************************************************************************
CHANEL_RESPONE:

        PUSH    0

        ;********************** RESET CONSTANT **************

        ANL     A,#0FH
        SETB    A.6
        MOV     R0,A
        LCALL   EXT_BUSY
        MOV     A,R0
        LCALL   EXT_RINGING
        MOV     A,R0
        LCALL   EXT_RING_BACK
        MOV     A,R0
        LCALL   EXT_LINK
        MOV     A,R0
        LCALL   EXT_DIAL
        MOV     A,R0
        LCALL   EXT_DTMF_ENABLE
        MOV     A,R0
        LCALL   EXT_ASSIGN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ;******************** SET CONSTANT ************

        MOV     A,R0
        CLR     A.6
        SETB    A.7
        MOV     R0,A
        LCALL   EXT_HOOK
        MOV     A,R0
        LCALL   EXT_SHIFT

        ;****************.RESET DTMF FLAG ***************

        MOV     A,R0
        ANL     A,#0FH
        LCALL   DTMF_WRITE_FLAG
        ;**************** COLSE SW ***************
        MOV     A,R0
        ANL     A,#0FH
        MOV     R0,A
        LCALL   EXT_MEDIA
        SETB    A.4
        SWAP    A
        ORL     A,R0
        MOV     ADDR_SW,A
        LCALL   SET_MAP_SW

        POP     0
        RET

;*************************************************************************
;**                      [ DTMF SERVICE CONTROL ]
;*************************************************************************
DTMF_SERVICE:

        PUSH    A
        PUSH    B
        PUSH    0
        PUSH    1
        PUSH    2
        PUSH    3
        PUSH    4
        PUSH    5
        PUSH    6
        PUSH    DPH
        PUSH    DPL

        MOV     R3,DTMF_USE             ; R3 SENSE FROM STD 1 TO 8

        MOV     R4,DTMF_USE+1           ; R4 SENSE FROM STD 9 TO 11

        MOV     R0,#16          ; EXT 16 CHANEL
DSV_LOOP:
        MOV     A,R0
        DEC     A
        PUSH    A               ; KEEP EXT CHANEL

        LCALL   EXT_STATUS              ; CHECK STATUS TO ENABLE DTMF
        ANL     A,#10110111B            ; DTMF ENABLE,HOOK ON,BUSY OFF
        CJNE    A,#10100001B,NOT_DTMF1   ; RINGING,RING BACK OFF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            POP     A                       ; GET EXT CHANEL
            LCALL   EXT_MEDIA
            MOV     R6,A                    ; KEEP MEDIA TO USE BY EXT TO R6
            LCALL   DTMF_READ_FLAG
            MOV     R5,A

            MOV     A,R6
            CJNE    A,#0,DSV_1STD
            MOV     A,R3
            JMP     DSV_STD0
DSV_1STD:
            CJNE    A,#1,DSV_2STD
            MOV     A,R3
            JMP     DSV_STD1
DSV_2STD:
            CJNE    A,#2,DSV_3STD
            MOV     A,R3
            JMP     DSV_STD2
DSV_3STD:
            CJNE    A,#3,DSV_4STD
            MOV     A,R3
            JMP     DSV_STD3
DSV_4STD:
            CJNE    A,#4,DSV_5STD
            MOV     A,R3
            JMP     DSV_STD4
DSV_5STD:
            CJNE    A,#5,DSV_6STD
            MOV     A,R3
            JMP     DSV_STD5
DSV_6STD:
            CJNE    A,#6,DSV_7STD
            MOV     A,R3
            JMP     DSV_STD6
DSV_7STD:
            MOV     A,R3

            RR      A
DSV_STD6:   RR      A
DSV_STD5:   RR      A
DSV_STD4:   RR      A
DSV_STD3:   RR      A
DSV_STD2:   RR      A
DSV_STD1:   RR      A
DSV_STD0:
            ANL     A,#1                    ; DETECT STD SIGNAL
            CJNE    A,#1,NOT_DTMF           ;

            MOV     A,R5
            JB      A.0,DSV_DTMF2

            MOV     A,R6
            LCALL   DTMF_PORT
            MOV     R2,A

            MOV     A,R6
            MOV     R1,#DTMF_DATA
            ADD     A,R1
            MOV     R1,A
            MOV     A,R2

            MOV     @R1,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV         A,R5
            SETB        A.0
            SWAP        A
            ANL         A,#F0H
            MOV         R5,A
            MOV         A,R6
            ANL         A,#0FH
            ORL         A,R5
            LCALL       DTMF_WRITE_FLAG

            MOV         A,R0                ;
            DEC         A                   ;  RESET DIAL
            SETB        A.6                 ;
            LCALL       EXT_DIAL            ;
NOT_DTMF1:
            JMP         NOT_DTMF
DSVEX1:
            JMP         DSV_LOOP

DSV_DTMF2:
            JB          A.1,NOT_DTMF

            MOV         A,R6
            LCALL       DTMF_PORT
            MOV         R2,A

            MOV         A,R6
            MOV         R1,#DTMF_DATA
            ADD         A,R1
            MOV         R1,A

            MOV         A,R2
            SWAP        A
            ANL         A,#F0H
            MOV         R2,A
            MOV         A,@R1
            ANL         A,#0FH
            ORL         A,R2

            MOV         @R1,A

            MOV         A,R5
            SETB        A.1
            SWAP        A
            ANL         A,#F0H
            MOV         R5,A
            MOV         A,R6
            ANL         A,#0FH
            ORL         A,R5
            LCALL       DTMF_WRITE_FLAG
            JMP         NOT_DTMF
NOT_DTMF:
            DJNZ        R1,DSVEX1

            POP         DPL
            POP         DPH
            POP         6
            POP         5
            POP         4
            POP         3
            POP         2
            POP         1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            POP         0
            POP         B
            POP         A
            RET

;*****************************************************************************
;**                              [ DTMF STATUS ]                            *
;*****************************************************************************
DTMF_STATUS:

            PUSH        A
            PUSH        B
            PUSH        0
            PUSH        1
            PUSH        2
            PUSH        3
            PUSH        DPH
            PUSH        DPL

            MOV         R0,DTMF_BAK

            MOV         DPTR,#SW3_C                 ; R3 SENSE FROM STD 1 TO
            MOVX        A,@DPTR                    ;
            MOV         R1,A                       ; R4 SENSE FROM STD 9 TO 1

            MOV         R2,#8
            MOV         R3,#1
            MOV         B,#0
NEXT_STD1:
            MOV         A,R1
            ANL         A,R3
            CJNE        A,#0,CHK_BAK1
            JMP         NO_KEY1
CHK_BAK1:
            MOV         A,R0
            ANL         A,R3
            CJNE        A,#0,NO_KEY1

            MOV         A,B
            ORL         A,R3
            MOV         B,A
NO_KEY1:
            MOV         A,R3
            RL          A
            MOV         R3,A
            DJNZ        R2,NEXT_STD1

            MOV         DTMF_BAK,R1

            MOV         DTMF_USE,B

;********************************* STD 9 120 11 ********************

            MOV         R0,DTMF_BAK+1

            MOV         DPTR,#SW2_C
            MOVX        A,@DPTR
            SWAP        A
            ANL         A,#07H
            MOV         R1,A

            MOV         R2,#3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV         R3,#1
            MOV         B,#0
NEXT_STD2:
            MOV         A,R1
            ANL         A,R3
            CJNE        A,#0,CHK_BAK2
            JMP         NO_KEY2
CHK_BAK2:
            MOV         A,R0
            ANL         A,R3
            CJNE        A,#0,NO_KEY2

            MOV         A,B
            ORL         A,R3
            MOV         B,A

NO_KEY2:
            MOV         A,R3
            RL          A
            MOV         R3,A
            DJNZ        R2,NEXT_STD2

            MOV         DTMF_BAK+1,R1

            MOV         DTMF_USE+1,B

            POP         DPL
            POP         DPH
            POP         3
            POP         2
            POP         1
            POP         0
            POP         B
            POP         A
            RET

;*****************************************************************************
;**                          [ DTMF FLAG READ ]                               *
;*****************************************************************************
DTMF_READ_FLAG:                                     ; AL MEDIA CHANEL
                                                    ; AH FLAG WRITE A.4 A.5
            PUSH        0
            PUSH        1
            PUSH        2

            MOV         R0,A
            RR          A                   ;
            RR          A                   ; SELECT BYTE TO USE DTMF FLAG
            ANL         A,#1                ;
            MOV         R1,#DTMF_FLAG       ;
            ADD         A,R1                ;
            MOV         R1,A                ;
            MOV         A,@R1               ; READ BYTE TO USE
            MOV         R2,A                ; KEEP BYTE TO USE IN R2

            MOV         A,R0                ; GET MEDIA TO USE BY EXT
            ANL         A,#3                ;

            CJNE        A,#0,DFR_CHK1       ;
            MOV         A,R2                ;
            JMP         DFR_1               ;
DFR_CHK1:                                   ;
```

นี่เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CJNE    A,#1,DFR_CHK2       ;
            MOV     A,R2                ;   SELECT BIT DTMF FLAG D1 D2
            JMP     DFR_2               ;
DFR_CHK2:                               ;
            CJNE    A,#2,DFR_CHK2       ;
            MOV     A,R2                ;
            JMP     DFR_3               ;
DFR_CHK3:                               ;
            MOV     A,R2                ;
                                        ;
            RR      A                   ;
            RR      A                   ;
DFR_3:      RR      A                   ;
            RR      A                   ;
DFR_2:      RR      A                   ;
            RR      A                   ;
DFR_1:      ANL     A,#3                ;

            POP     2
            POP     1
            POP     0
            RET

;*****************************************************************************
;**                          [ DTMF FLAG WRITE ]                            **
;*****************************************************************************
DTMF_WRITE_FLAG:                        ; AL MEDIA CHANEL
                                        ; AH FLAG WRITE A.4 A.5
            PUSH    0
            PUSH    1
            PUSH    2
            PUSH    3

            MOV     R0,A
            ANL     A,#0FH
            RR      A                   ;
            RR      A                   ; SELECT BYTE TO USE DTMF FLAG
            ANL     A,#1                ;
            MOV     R1,#DTMF_FLAG       ;
            ADD     A,R1                ;
            MOV     R1,A                ;
            MOV     A,@R1
            MOV     R3,A
            MOV     A,R0                ; READ BYTE TO USE
            SWAP    A
            ANL     A,#3
            MOV     R2,A                ; KEEP BYTE TO USE IN R2

            MOV     A,R0                ; GET MEDIA TO USE BY EXT
            ANL     A,#3                ;

            CJNE    A,#0,DWF_1CHK       ;
            MOV     A,R3
            ANL     A,#11111100B
            MOV     R3,A
            MOV     A,R2                ;
            JMP     DWF_1A               ;
DWF_1CHK:                               ;
            CJNE    A,#1,DWF_2CHK       ;
            MOV     A,R3
            ANL     A,#11110011B
            MOV     R3,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV         A,R2                    ;   SELECT BIT DTMF FLAG D1 D2
            JMP         DWF_2A                  ;
DWF_2CHK:                                       ;
            CJNE        A,#2,DWF_3CHK           ;
            MOV         A,R3
            ANL         A,#11001111B
            MOV         R3,A
            MOV         A,R2                    ;
            JMP         DWF_3A                  ;
DWF_3CHK:                                       ;
            MOV         A,R3
            ANL         A,#00111111B
            MOV         R3,A
            MOV         A,R2                    ;
                                                ;
            RL          A                       ;
            RL          A                       ;
DWF_3A:     RL          A                       ;
            RL          A                       ;
DWF_2A:     RL          A                       ;
            RL          A                       ;
DWF_1A:     ORL         A,R3
            MOV         @R1,A

            POP         3
            POP         2
            POP         1
            POP         0
            RET


;*************************************************************************
;**                          [ DTMF PORT ]                                *
;*************************************************************************
DTMF_PORT:
            CJNE        A,#0,DP_CHK1
            JMP         DP0
DP_CHK1:
            CJNE        A,#1,DP_CHK2
            JMP         DP1
DP_CHK2:
            CJNE        A,#2,DP_CHK3
            JMP         DP2
DP_CHK3:
            CJNE        A,#3,DP_CHK4
            JMP         DP3
DP_CHK4:
            CJNE        A,#4,DP_CHK5
            JMP         DP4
DP_CHK5:
            CJNE        A,#5,DP_CHK6
            JMP         DP5
DP_CHK6:
            CJNE        A,#6,DP_CHK7
            JMP         DP6
DP_CHK7:
            CJNE        A,#7,DP_CHK8
            JMP         DP7
DP_CHK8:
            CJNE        A,#8,DP_CHK9
            JMP         DP8
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DP_CHK9:
            CJNE        A,#9,DP_CHK10
            JMP         DP9
DP_CHK10:
            CJNE        A,#10,DP_END
            JMP         DP10

DP0:        MOV         DPTR,#SW1_A
            MOVX        A,@DPTR
            ANL         A,#0FH
            JMP         DP_END

DP1:        MOV         DPTR,#SW1_A
            MOVX        A,@DPTR
            SWAP        A
            ANL         A,#0FH
            JMP         DP_END

DP2:        MOV         DPTR,#SW1_B
            MOVX        A,@DPTR
            ANL         A,#0FH
            JMP         DP_END

DP3:        MOV         DPTR,#SW1_B
            MOVX        A,@DPTR
            SWAP        A
            ANL         A,#0FH
            JMP         DP_END

DP4:        MOV         DPTR,#SW1_C
            MOVX        A,@DPTR
            ANL         A,#0FH
            JMP         DP_END

DP5:        MOV         DPTR,#SW1_C
            MOVX        A,@DPTR
            SWAP        A
            ANL         A,#0FH
            JMP         DP_END

DP6:        MOV         DPTR,#SW2_A
            MOVX        A,@DPTR
            ANL         A,#0FH
            JMP         DP_END

DP7:        MOV         DPTR,#SW2_A
            MOVX        A,@DPTR
            SWAP        A
            ANL         A,#0FH
            JMP         DP_END

DP8:        MOV         DPTR,#SW2_B
            MOVX        A,@DPTR
            ANL         A,#0FH
            JMP         DP_END

DP9:        MOV         DPTR,#SW2_B
            MOVX        A,@DPTR
            SWAP        A
            ANL         A,#0FH
            JMP         DP_END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DP10:       MOV         DPTR,#SW2_C
            MOVX        A,@DPTR
            ANL         A,#0FH
            JMP         DP_END

DP_END:
            RET

;*****************************************************************************
;**                          [ MEDIA ASSIGN ]                                *
;*****************************************************************************
MEDIA_ASSIGN:
            PUSH        0
            PUSH        1
            PUSH        2
            PUSH        3
            PUSH        B

            MOV         B,#0                ; B FLAG SET IF MEDIA = MEDIA USE
            MOV         R0,#8               ; R0 = LOOP MEDIA
MAS_MEDIA:
            MOV         A,R0                ;   R1 = /(R0-1)
            DEC         A                   ;   R0 = (8->1)
            CPL         A                   ;   R1 = (0->7)
            ANL         A,#7                ;
            MOV         R1,A                ;

            MOV         R2,#16              ; R2 = LOOP EXT
MAS_EXT:
            MOV         A,R2                ;
            DEC         A                   ;
            MOV         R3,A                ; R3 = R2-1
            LCALL       EXT_ASSIGN          ;
            CJNE        A,#1,MAS_NON        ;
            MOV         A,R3                ;
            LCALL       EXT_MEDIA           ;
            CJNE        A,1,MAS_NON        ;
            MOV         B,#1                ; B = 1 WHEN MEDIA = MEDIA USE
MAS_NON:
            DJNZ        R2,MAS_EXT          ;
            MOV         A,B                         ;
            CJNE        A,#1,MAS_END        ;
                                            ;
            DJNZ        R0,MAS_MEDIA
MAS_END:
            MOV         A,R1                ; A = MEDIA NO USE

            POP         B
            POP         3
            POP         2
            POP         1
            POP         0
            RET

;*****************************************************************************
;**                          [ SAND BUSY ]     SAND BUSY TO EXT CHANEL     **
;*****************************************************************************

SAND_BUSY:
            PUSH        A
            PUSH        0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            PUSH        DPH
            PUSH        DPL
            MOV         R0,#16
SB_LOOP:
            MOV         A,R0
            DEC         A
            LCALL       EXT_BUSY
            CJNE        A,#1,SB_ON
            JMP         SB_OFF
SB_NEXT:
            DJNZ        R0,SB_LOOP
            JMP         SB_END
SB_ON:
            MOV         A,R0
            DEC         A
            ORL         A,#MAP_BUSY
            MOV         ADDR_SW,A
            LCALL       SET_MAP_SW
            JMP         SB_NEXT
SB_OFF:
            MOV         A,R0
            DEC         A
            ORL         A,#MAP_BUSY
            MOV         ADDR_SW,A
            LCALL       CLR_MAP_SW
            JMP         SB_NEXT
SB_END:
            POP         DPL
            POP         DPH
            POP         0
            POP         A
            RET

;*************************************************************************
;**                          [ SAND DIAL ]    SAND DIAL TO EXT CHANEL     **
;*************************************************************************

SAND_DIAL:
            PUSH        A
            PUSH        0
            PUSH        DPH
            PUSH        DPL

            MOV         R0,#16
SD_LOOP:
            MOV         A,R0
            DEC         A
            LCALL       EXT_DIAL
            CJNE        A,#1,SB_OFF
            JMP         SD_ON
SD_NEXT:
            DJNZ        R0,SD_LOOP
            JMP         SD_END
SD_ON:
            MOV         A,R0
            DEC         A
            ORL         A,#MAP_DIAL
            MOV         ADDR_SW,A
            LCALL       SET_MAP_SW
            JMP         SD_NEXT
SD_OFF:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV         A,R0
            DEC         A
            ORL         A,#MAP_DIAL
            MOV         ADDR_SW,A
            LCALL       CLR_MAP_SW
            JMP         SD_NEXT
SD_END:
            POP         DPL
            POP         DPH
            POP         0
            POP         A
            RET

;*****************************************************************************
;**                        [ SAND RING BACK ]  SAND RING BACK TO EXT CHANEL   **
;*****************************************************************************
SAND_RING_BACK:
            PUSH        A
            PUSH        0
            PUSH        DPH
            PUSH        DPL
            MOV         R0,#16
SRB_LOOP:
            MOV         A,R0
            DEC         A
            LCALL       EXT_RING_BACK
            CJNE        A,#1,SRB_ON
            JMP         SD_OFF
SRB_NEXT:
            DJNZ        R0,SRB_LOOP
            JMP         SD_END
SRB_ON:
            MOV         A,R0
            DEC         A
            ORL         A,#MAP_RING_BACK
            MOV         ADDR_SW,A
            LCALL       SET_MAP_SW
            JMP         SRB_NEXT
SRB_OFF:
            MOV         A,R0
            DEC         A
            ORL         A,#MAP_RING_BACK
            MOV         ADDR_SW,A
            LCALL       CLR_MAP_SW
            JMP         SRB_NEXT
SRB_END:
            POP         DPL
            POP         DPH
            POP         0
            POP         A
            RET

;*****************************************************************************
;**                        [ SAND RINGING ]  SAND RINGING TO EXT CHANEL **
;*****************************************************************************
SAND_RINGING:
            PUSH        A
            PUSH        0
            PUSH        1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            PUSH        2
            PUSH        DPH
            PUSH        DPL
                                                    ; SAND RINGING SUB1
            MOV         R0,#8
            MOV         R1,#0
SRG_NEXT1:
            MOV         A,R0
            DEC         A
            CPL         A
            ANL         A,#7
            MOV         R2,A
            LCALL       EXT_RINGING
            LCALL       SRG_ADD_RINGING
            DJNZ        R0,SRG_NEXT1

            MOV         DPTR,#SUB1_B                ; WRITE RINGING TO PORT
            MOV         A,R1
            MOVX        @DPTR,A
                              ;-------------------------------------
            MOV         R0,#8                       ; SAND RINGING SUB2
            MOV         R1,#0
SRG_NEXT2:
            MOV         A,R0
            DEC         A
            CPL         A
            ANL         A,#7
            MOV         R2,A
            SETB        A.3
            LCALL       EXT_RINGING
            LCALL       SRG_ADD_RINGING
            DJNZ        R0,SRG_NEXT2

            MOV         DPTR,#SUB2_B                ; WRITE RINGING TO PORT
            MOV         A,R1
            MOVX        @DPTR,A

            POP         DPL
            POP         DPH
            POP         2
            POP         1
            POP         0
            POP         A
            RET

;---------------------------------END SAND RINGING--------------
SRG_ADD_RINGING:                                    ;
            CJNE        R2,#0,SRG_CHK_B1            ;
            JMP         SRG_B0                      ;       SUB RUTINE ADD DATA
SRG_CHK_B1:                                         ;       RINGING TO R1
            CJNE        R2,#1,SRG_CHK_B2            ;       ADDR RINGING FORM R2
            JMP         SRG_B1                      ;
SRG_CHK_B2:                                         ;
            CJNE        R2,#2,SRG_CHK_B3            ;
            JMP         SRG_B2                      ;
SRG_CHK_B3:                                         ;
            CJNE        R2,#3,SRG_CHK_B4            ;
            JMP         SRG_B3                      ;
SRG_CHK_B4:                                         ;
            CJNE        R2,#4,SRG_CHK_B5            ;
            JMP         SRG_B4                      ;
SRG_CHK_B5:                                         ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          CJNE      R2,#5,SRG_CHK_B6          ;
          JMP       SRG_B5                    ;
SRG_CHK_B6:                                   ;
          CJNE      R2,#6,SRG_B7              ;
          JMP       SRG_B6                    ;
                                              ;
SRG_B7:   RL        A                         ;
SRG_B6:   RL        A                         ;
SRG_B5:   RL        A                         ;
SRG_B4:   RL        A                         ;
SRG_B3:   RL        A                         ;
SRG_B2:   RL        A                         ;
SRG_B1:   RL        A                         ;
SRG_B0:   ORL       A,R1                      ;
          MOV       R1,A                      ;
          RET                                 ;

;*************************************************************************
;**                          [ SAND TOT CH    ]                          *
;*************************************************************************
SAND_TOT:
          PUSH      A
          PUSH      DPH
          PUSH      DPL

          MOV       A,#0
          JNB       TOTF1,ST_CHK2
          SETB      A.0

ST_CHK2:  JNB       TOTF2,ST_CHK3
          SETB      A.1

ST_CHK3:  JNB       TOTF3,ST_SAND
          SETB      A.2
ST_SAND:
          SWAP      A
          MOV       DPTR,#INTERF_C
          MOVX      @DPTR,A

          POP       DPL
          POP       DPH
          POP       A
          RET

;*************************************************************************
;**                          [ HOOK STATUS ]    0=NC 1=ON HOOK 2=OFF HOOK  *
;*************************************************************************
HOOK_STATUS:
          PUSH      B
          PUSH      0
          PUSH      1

          MOV       R0,#HOOK
          PUSH      A
          JB        A.3,HS_BYTE2
          ANL       A,#07
          JMP       HS_CHK_HOOK
HS_BYTE2:
          INC       R0
          ANL       A,#07
HS_CHK_HOOK:
          LCALL     CHK_BIT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CJNE    A,#1,HS_CHK_OFF
            JMP     HS_CHK_ON
HS_CHK_OFF:
            POP     A
            ANL     A,#0FH
            LCALL   EXT_HOOK
            CJNE    A,#1,HS_CLR_A
            MOV     A,#2
            JMP     HS_END
HS_CHK_ON:
            POP     A
            ANL     A,#0FH
            LCALL   EXT_HOOK
            CJNE    A,#0,HS_CLR_A
            MOV     A,#1
            JMP     HS_END
HS_CLR_A:
            MOV     A,#0
HS_END:

            POP     1
            POP     0
            POP     B
            RET                         ;

;*************************************************************************
;**                          [ HOOK UPDATE ]                            *
;*************************************************************************
HOOK_UPDATE:
            PUSH    DPH
            PUSH    DPL

            MOV     DPTR,#SUB1_A
            MOVX    A,@DPTR
            CPL     A
            MOV     HOOK,A

            MOV     DPTR,#SUB2_A
            MOVX    A,@DPTR
            CPL     A
            MOV     HOOK+1,A

            POP     DPL
            POP     DPH
            RET

;*************************************************************************
;**          [ CHECK BIT IN BYTE ] A=BIT NUMBER 0-7 R0=ADDR OF BYTE     *
;*************************************************************************

CHK_BIT:
            ANL     A,#07H

            CJNE    A,#0,CB10
            MOV     A,@R0
            JMP     CB00

CB10:       CJNE    A,#1,CB20
            MOV     A,@R0
            JMP     CB01
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CB20:       CJNE    A,#2,CB30
            MOV     A,@R0
            JMP     CB02

CB30:       CJNE    A,#3,CB40
            MOV     A,@R0
            JMP     CB03

CB40:       CJNE    A,#4,CB50
            MOV     A,@R0
            JMP     CB04

CB50:       CJNE    A,#5,CB60
            MOV     A,@R0
            JMP     CB05

CB60:       CJNE    A,#6,CB70
            MOV     A,@R0
            JMP     CB06

CB70:       MOV     A,@R0

CB07:       RR      A
CB06:       RR      A
CB05:       RR      A
CB04:       RR      A
CB03:       RR      A
CB02:       RR      A
CB01:       RR      A
CB00:
            ANL     A,#1

            RET

;*************************************************************************
;**                          [ ON SWITCH ]                              *
;*************************************************************************

ON_SW:
            CLR     IE.7
            PUSH    A
            PUSH    DPH
            PUSH    DPL

            MOV     DPTR,#SW3_A
            MOV     A,ADDR_SW
            MOVX    @DPTR,A

            MOV     DPTR,#SW3_B
            MOV     A,#1H
            MOVX    @DPTR,A

            MOV     DPTR,#SW3_B
            MOV     A,#3H
            MOVX    @DPTR,A

            MOV     DPTR,#SW3_B
            MOV     A,#1H
            MOVX    @DPTR,A

            MOV     DPTR,#SW3_B
            MOV     A,#0H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOVX    @DPTR,A

            POP     DPL
            POP     DPH
            POP     A
            SETB    IE.7
            RET

;*****************************************************************************
;**                          [ OFF SWITCH ]                                 *
;*****************************************************************************
OFF_SW:
            CLR     IE.7
            PUSH    A
            PUSH    DPH
            PUSH    DPL

            MOV     DPTR,#SW3_A
            MOV     A,ADDR_SW
            MOVX    @DPTR,A

            MOV     DPTR,#SW3_B
            MOV     A,#0H
            MOVX    @DPTR,A

            MOV     DPTR,#SW3_B
            MOV     A,#2H
            MOVX    @DPTR,A

            MOV     DPTR,#SW3_B
            MOV     A,#0H
            MOVX    @DPTR,A

            POP     DPL
            POP     DPH
            POP     A
            SETB    IE.7
            RET

;*****************************************************************************
;**                          [ MEAS MAP SW ]                                **
;*****************************************************************************
MEAS_MAP_SW:
            PUSH    1

            MOV     A,ADDR_SW

            RR      A
            RR      A
            RR      A                       ; ADDR = R1:A
            ANL     A,#00011111B
            ADD     A,#MAP_SW
            MOV     R1,A
            MOV     A,ADDR_SW
            ANL     A,#00000111B            ; R0 IT DATA IN ADDR
                                            ; R1 IT ADDR TO BYTE FOR MAP
            CJNE    A,#0,MMS10
            MOV     A,@R1
            JMP     MMS00
MMS10:      CJNE    A,#1,MMS20
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV     A,@R1
            JMP     MMS01
MMS20:      CJNE    A,#2,MMS30
            MOV     A,@R1
            JMP     MMS02
MMS30:      CJNE    A,#3,MMS40
            MOV     A,@R1
            JMP     MMS03
MMS40:      CJNE    A,#4,MMS50
            MOV     A,@R1
            JMP     MMS04
MMS50:      CJNE    A,#5,MMS60
            MOV     A,@R1
            JMP     MMS05
MMS60:      CJNE    A,#6,MMS70
            MOV     A,@R1
            JMP     MMS06
MMS70:      MOV     A,@R1
MMS07:      RR      A
MMS06:      RR      A
MMS05:      RR      A
MMS04:      RR      A
MMS03:      RR      A
MMS02:      RR      A
MMS01:      RR      A
MMS00:
            ANL     A,#1

            POP     1
            RET

;*****************************************************************************
;**                          [ CLEAR MAP SW ]                                 *
;*****************************************************************************

CLR_MAP_SW:
            PUSH    B
            PUSH    1
            PUSH    A                       ; A IT ADDR TO BIT FOR MAP
            PUSH    DPH
            PUSH    DPL

            MOV     A,ADDR_SW

            RR      A
            RR      A                       ; ADDR = R1:A
            RR      A
            ANL     A,#00011111B
            ADD     A,#MAP_SW
            MOV     R1,A
            MOV     A,ADDR_SW
            ANL     A,#00000111B            ; R0 IT DATA IN ADDR
                                            ; R1 IT ADDR TO BYTE FOR MA
            MOV     B,@R1

            CJNE    A,#0,CMS00
            CLR     B.0
            JMP     CMSEND
CMS00:
            CJNE    A,#1,CMS01
            CLR     B.1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            JMP     CMSEND
CMS01:
            CJNE    A,#2,CMS02
            CLR     B.2
            JMP     CMSEND
CMS02:
            CJNE    A,#3,CMS03
            CLR     B.3
            JMP     CMSEND
CMS03:
            CJNE    A,#4,CMS04
            CLR     B.4
            JMP     CMSEND
CMS04:
            CJNE    A,#5,CMS05
            CLR     B.5
            JMP     CMSEND
CMS05:
            CJNE    A,#6,CMS06
            CLR     B.6
            JMP     CMSEND
CMS06:
            CJNE    A,#7,CMSEND
            CLR     B.7
CMSEND:
            MOV     @R1,B

            LCALL   OFF_SW

            POP     DPL
            POP     DPH
            POP     A
            POP     1
            POP     B
            RET

;*****************************************************************************
;**                          [ SET MAP SW ]                                   *
;*****************************************************************************

SET_MAP_SW:
            PUSH    B
            PUSH    1
            PUSH    A                       ; A IT ADDR TO BIT FOR MAP
            PUSH    DPH
            PUSH    DPL

            MOV     A,ADDR_SW

            RR      A
            RR      A                       ; ADDR = R1:A
            RR      A
            ANL     A,#00011111B
            ADD     A,#MAP_SW
            MOV     R1,A
            MOV     A,ADDR_SW
            ANL     A,#00000111B            ; R0 IT DATA IN ADDR
                                            ; R1 IT ADDR TO BYTE FOR MA
            MOV     B,@R1

            CJNE    A,#0,SMS00
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            SETB    B.0
            JMP     SMSEND
SMS00:
            CJNE    A,#1,SMS01
            SETB    B.1
            JMP     SMSEND
SMS01:
            CJNE    A,#2,SMS02
            SETB    B.2
            JMP     SMSEND
SMS02:
            CJNE    A,#3,SMS03
            SETB    B.3
            JMP     SMSEND
SMS03:
            CJNE    A,#4,SMS04
            SETB    B.4
            JMP     SMSEND
SMS04:
            CJNE    A,#5,SMS05
            SETB    B.5
            JMP     SMSEND
SMS05:
            CJNE    A,#6,SMS06
            SETB    B.6
            JMP     SMSEND
SMS06:
            CJNE    A,#7,SMSEND
            SETB    B.7
SMSEND:
            MOV     @R1,B

            LCALL   ON_SW

            POP     DPL
            POP     DPH
            POP     A
            POP     1
            POP     B
            RET

;*************************************************************************
;**                          [ CLEAR ALL MAP SW ]                       **
;*************************************************************************

CLR_ALL_MAP_SW:
            PUSH    1
            MOV     R1,#00
CAMS00:     MOV     ADDR_SW,R1
            LCALL   CLR_MAP_SW
            DJNZ    R1,CAMS00
            POP     1
            RET

;*************************************************************************
;**                    [ CONVERT HEX IN BYTE TO ASCII ]                 **
;*************************************************************************

HEX2ASCII:
            PUSH    B
            MOV     B,A
            ANL     A,#0FH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            LCALL   BIT2CHAR
            MOV     CHL,A

            MOV     A,B
            SWAP    A
            ANL     A,#0FH
            LCALL   BIT2CHAR
            MOV     CHH,A
            POP     B
            RET                                 ; RETURN TO PROGRAM CALL

BIT2CHAR:                                       ;
            CJNE    A,#0AH,NEXT00               ;
            MOV     A,#41H                      ;
            JMP     H2A                         ;
NEXT00:                                         ;
            CJNE    A,#0BH,NEXT01               ;
            MOV     A,#42H                      ;
            JMP     H2A                         ;
NEXT01:                                         ;
            CJNE    A,#0CH,NEXT02               ;
            MOV     A,#43H                      ; INTERNAL SUBROUTINE
            JMP     H2A                         ;
NEXT02:                                         ;
            CJNE    A,#0DH,NEXT03               ;
            MOV     A,#44H                      ;
            JMP     H2A                         ;
NEXT03:                                         ;
            CJNE    A,#0EH,NEXT04               ;
            MOV     A,#45H                      ;
            JMP     H2A                         ;
NEXT04:                                         ;
            CJNE    A,#0FH,NEXT05               ;
            MOV     A,#46H                      ;
            JMP     H2A                         ;
NEXT05:                                         ;
            ADD     A,#48                       ;
H2A:        RET                                 ;

;*************************************************************************
;**                          [ INIT DATA TO RAM ]                        *
;*************************************************************************
INIT_DATA:
            PUSH    A
            PUSH    DPH
            PUSH    DPL

            MOV     R0,#70
INTD_LOOP:
            MOV     DPTR,#EXT01
            MOV     A,#0
            MOVC    A,@A+DPTR
            MOVX    @DPTR,A
            INC     DPTR
            DJNZ    R0,INTD_LOOP

            POP     DPL
            POP     DPH
            POP     A
            RET
;*************************************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;**                    [ INITIAL DATA DISPLAY TO RAM ]                          *
;*****************************************************************************

INIT_DATA_DIS:

          MOV     DPL0,DPL                    ; ADDR DATA TRANFER I/P
          MOV     DPH0,DPH                    ; IN DPTR

          MOV     DPTR,#ADDR_DATA             ; DATA SCAN
          MOV     DPL1,DPL
          MOV     DPH1,DPH

          MOV     R0,#40
IDD00:
          MOV     DPL,DPL0
          MOV     DPH,DPH0
          MOV     A,#0
          MOVC    A,@A+DPTR
          INC     DPTR
          MOV     DPL0,DPL
          MOV     DPH0,DPH

          MOV     DPL,DPL1
          MOV     DPH,DPH1
          MOVX    @DPTR,A
          INC     DPTR
          MOV     DPL1,DPL
          MOV     DPH1,DPH

          DJNZ    R0,IDD00
          RET

;*****************************************************************************
;**                          [ PRINT DISPLAY ]                                *
;*****************************************************************************

PRINTF:
          CLR     IE.7
          PUSH    A
          PUSH    0
          PUSH    DPH
          PUSH    DPL

          LCALL   CLRSCR
          MOV     R0,#20
          MOV     DPTR,#ADDR_DATA

          MOV     A,#00
          LCALL   LCD_COM
STR1:
          MOV     A,#0
          MOVX    A,@DPTR                     ; XCH MOVX OR MOVC
          LCALL   WRITE
          INC     DPTR
          DJNZ    R0,STR1

          MOV     A,#0C0H                     ;line 2
          LCALL   LCD_COM

          MOV     R0,#20
STR3:
          MOV     A,#0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOVX    A,@DPTR                         ; XCH MOVX OR MOVC
        LCALL   WRITE
        INC     DPTR
        DJNZ    R0,STR3

        POP     DPL
        POP     DPH
        POP     0
        POP     A

        SETB    IE.7

        RET

;*****************************************************************************
;**                          [ INITIAL INT ]                                 *
;*****************************************************************************

INIT_INT:

        MOV     SCAN,#0


        MOV     CLOCK,#0

        MOV     IE,#10100101B
        MOV     IP,#00100000B
;       MOV     TMOD,#00100000B
        MOV     TCON,#00000001B

        DB      75H,C8H,04H                     ; MOV   T2CON,#00000100B
        DB      75H,CAH,FFH                     ; MOV   RCAP2L,#023H
        DB      75H,CBH,3FH                     ; MOV   RCAP2H,#0CFH

        RET

;*****************************************************************************
;**                      [ CLEAR SCREEN DISPLAY ]                            *
;*****************************************************************************

CLRSCR:
        PUSH    DPL
        PUSH    DPH
        PUSH    A

        MOV     A,#01                           ;  CLS DISPLAY
        MOV     DPTR,#COMMAND                   ;
        MOVX    @DPTR,A                         ;
        LCALL   WAITBF                          ;

        POP     A
        POP     DPH
        POP     DPL

        RET

;*****************************************************************************
;**                      [ Write ASCII to LCD ]                              *
;*****************************************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
WRITE:      PUSH    DPL
            PUSH    DPH

            MOV     DPTR,#WRITEDATA
            MOVX    @DPTR,A
            LCALL   WAITBF                          ;Wait LCD module ready

            POP     DPH
            POP     DPL

            RET

;*****************************************************************************
;**                          [ WRITE COMMAND ]                                *
;*****************************************************************************

LCD_COM:
            PUSH    DPL
            PUSH    DPH


            MOV     DPTR,#COMMAND
            MOVX    @DPTR,A
            LCALL   WAITBF                          ;Wait busy flag

            POP     DPH
            POP     DPL

            RET

;*****************************************************************************
;**                          [ Wait for ready ]                               *
;*****************************************************************************

WAITBF:     PUSH    DPL
            PUSH    DPH
            MOV     DPTR,#READBUSY
RDY1:       MOVX    A,@DPTR
            JB      ACC.7,RDY1                      ;Busy Flag
            POP     DPH
            POP     DPL
            RET

;*****************************************************************************
;**                          [ INITIAL DISPLAY ]                              *
;*****************************************************************************

INIT:       PUSH    DPL
            PUSH    DPH
            MOV     DPTR,#COMMAND

            MOV     A,#38H                          ;8bit, 2 line, 5x7 dot
            MOVX    @DPTR,A
            LCALL   WAITBF

            MOV     A,#0FH
            MOVX    @DPTR,A
            LCALL   WAITBF

            MOV     A,#6                            ;increment cursor
            MOVX    @DPTR,A
            LCALL   WAITBF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV     A,#1                        ;clear and home
            MOVX    @DPTR,A
            LCALL   WAITBF

            POP     DPH
            POP     DPL
            RET

;*****************************************************************************
;**                          [ WRITE CG RAM ]                                 *
;*****************************************************************************
WRITE_CG:
            PUSH    A
            PUSH    3
            PUSH    DPH
            PUSH    DPL

            MOV     R3,#8                       ;R3 FOR 8 DATA TO CG RAM

            MOV     A,R0                        ;R0 ADDR CG RAM
            RL      A
            RL      A
            RL      A
            SETB    A.6
            CLR     A.7

            PUSH    DPH
            PUSH    DPL
            MOV     DPTR,#COMMAND
            MOVX    @DPTR,A
            LCALL   WAITBF
            POP     DPL
            POP     DPH
CG_NEXT:
            MOV     A,#0
            MOVC    A,@A+DPTR
            INC     DPTR

            PUSH    DPH
            PUSH    DPL
            MOV     DPTR,#WRITEDATA
            MOVX    @DPTR,A
            LCALL   WAITBF
            POP     DPL
            POP     DPH

            DJNZ    R3,CG_NEXT
            POP     DPL
            POP     DPH
            POP     3
            POP     A
            RET

;*****************************************************************************
;**                          [ DELAY ]                                        **
;*****************************************************************************
DELAY:
            PUSH    B                           ; save bystander
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV     B,#0                    ; 256 loops
            DJNZ    B,$                     ; 2.17 us/loop

            MOV     B,#0                    ; 256 loops
            DJNZ    B,$                     ; 2.17 us/loop

            POP     B                       ; restore bystander

            DJNZ    ACC,DELAY               ; repeat ACC times

            RET

;*************************************************************************
;**                          [ INITIAL PORT ]                           *
;*************************************************************************

INIT_PORT:
            PUSH    DPL
            PUSH    DPH
            PUSH    A

            MOV     DPTR,#INTERF_COM
            MOV     A,#91H
            MOVX    @DPTR,A

            MOV     DPTR,#SUB1_COM
            MOV     A,#90H
            MOVX    @DPTR,A

            MOV     DPTR,#SUB2_COM
            MOV     A,#90H
            MOVX    @DPTR,A

            MOV     DPTR,#TOT1_COM
            MOV     A,#9BH
            MOVX    @DPTR,A

            MOV     DPTR,#TOT2_COM
            MOV     A,#9BH
            MOVX    @DPTR,A

            MOV     DPTR,#SW1_COM
            MOV     A,#9BH
            MOVX    @DPTR,A

            MOV     DPTR,#SW2_COM
            MOV     A,#9BH
            MOVX    @DPTR,A

            MOV     DPTR,#SW3_COM
            MOV     A,#89H
            MOVX    @DPTR,A

            MOV     DPTR,#INTERF_B
            MOV     A,#FFH
            MOVX    @DPTR,A

            POP     A
            POP     DPH
            POP     DPL
            RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*****************************************************************************
;**                        [ EXT BUSY CONTROL ]  A=0;READ,A.7;SET,A.6;RESET **
;*****************************************************************************

EXT_BUSY:
          PUSH    B
          PUSH    0
          PUSH     DPH
          PUSH     DPL

          MOV     DPTR,#EXT01
          MOV     R0,A
          ANL     A,#0FH
          MOV     B,#4
          MUL     AB
          MOV     B,A

          MOV     A,DPL                          ;  ADD DPTR,(EXT X 4)
          ADD     A,B                            ;
          MOV     DPL,A                          ;
          MOV     A,DPH                          ;
          ADDC    A,#00                          ;
          MOV     DPH,A                          ;
          MOV     A,R0
          JB      A.7,EB_SET
          JB      A.6,EB_RESET

          MOVX    A,@DPTR                        ;
          JB      A.4,EB_BSET                    ;
          MOV     A,#0                           ;
          JMP     EB_END                         ;  READ BUSY
EB_BSET:                                         ;
          MOV     A,#1                           ;
          JMP     EB_END                         ;
EB_SET:
          MOVX    A,@DPTR                        ;
          SETB    A.4                            ;  SET BUSY
          MOVX    @DPTR,A                        ;
          JMP     EB_END                         ;
EB_RESET:
          MOVX    A,@DPTR                        ;
          CLR     A.4                            ; RESET BUSY
          MOVX    @DPTR,A                        ;
EB_END:
         POP       DPL
         POP       DPH
          POP      0
          POP      B
          RET

;*****************************************************************************
;**                        [ EXT HOOK CONTROL ]  A=0;READ,A.7;SET,A.6;RESET **
;*****************************************************************************

EXT_HOOK:
          PUSH     B
          PUSH     0
          PUSH     DPH
          PUSH     DPL

          MOV      DPTR,#EXT01
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV     R0,A
            ANL     A,#0FH
            MOV     B,#4
            MUL     AB
            MOV     B,A

            MOV     A,DPL                       ;  ADD DPTR,(EXT X 4)
            ADD     A,B                         ;
            MOV     DPL,A                       ;
            MOV     A,DPH                       ;
            ADDC    A,#00                       ;
            MOV     DPH,A                       ;
            MOV     A,R0
            JB      A.7,EH_SET
            JB      A.6,EH_RESET

            MOVX    A,@DPTR                     ;
            JB      A.0,EH_BSET                 ;
            MOV     A,#0                        ;
            JMP     EH_END                      ;  READ HOOK
EH_BSET:                                        ;
            MOV     A,#1                        ;
            JMP     EH_END                      ;
EH_SET:
            MOVX    A,@DPTR                     ;
            SETB    A.0                         ;  SET HOOK
            MOVX    @DPTR,A                     ;
            JMP     EH_END                      ;
EH_RESET:
            MOVX    A,@DPTR                     ;
            CLR     A.0                         ; RESET HOOK
            MOVX    @DPTR,A                     ;
EH_END:
            POP     DPL
            POP     DPH
            POP     0
            POP     B
            RET

;*************************************************************************:
;**                        [ EXT DIAL CONTROL ]  A=0;READ,A.7;SET,A.6;RESET *:
;*************************************************************************:

EXT_DIAL:
            PUSH    B
            PUSH    0
            PUSH    DPH
            PUSH    DPL


            MOV     DPTR,#EXT01
            MOV     R0,A
            ANL     A,#0FH
            MOV     B,#4
            MUL     AB
            MOV     B,A

            MOV     A,DPL                       ;  ADD DPTR,(EXT X 4)
            ADD     A,B                         ;
            MOV     DPL,A                       ;
            MOV     A,DPH                       ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            ADDC    A,#00                           ;
            MOV     DPH,A                           ;
            MOV     A,R0
            JB      A.7,ED_SET
            JB      A.6,ED_RESET

            MOVX    A,@DPTR                         ;
            JB      A.3,ED_BSET                     ;
            MOV     A,#0                            ;
            JMP     ED_END                          ;  READ DIAL
ED_BSET:                                            ;
            MOV     A,#1                            ;
            JMP     ED_END                          ;
ED_SET:
            MOVX    A,@DPTR                         ;
            SETB    A.3                             ;  SET DIAL
            MOVX    @DPTR,A                         ;
            JMP     ED_END                          ;
ED_RESET:
            MOVX    A,@DPTR                         ;
            CLR     A.3                             ; RESET DIAL
            MOVX    @DPTR,A                         ;
ED_END:
            POP       DPL
            POP       DPH
            POP     0
            POP     B
            RET

;*************************************************************************
;**                    [ EXT RINGING CONTROL ]  A=0;READ,A.7;SET,A.6;RESET :
;*************************************************************************

EXT_RINGING:
            PUSH    B
            PUSH    0
            PUSH     DPH
            PUSH     DPL

            MOV     DPTR,#EXT01
            MOV     R0,A
            ANL     A,#0FH
            MOV     B,#4
            MUL     AB
            MOV     B,A

            MOV     A,DPL                           ;  ADD DPTR,(EXT X 4)
            ADD     A,B                             ;
            MOV     DPL,A                           ;
            MOV     A,DPH                           ;
            ADDC    A,#00                           ;
            MOV     DPH,A                           ;
            MOV     A,R0
            JB      A.7,ERG_SET
            JB      A.6,ERG_RESET

            MOVX    A,@DPTR                         ;
            JB      A.1,ERG_BSET                    ;
            MOV     A,#0                            ;
            JMP     ERG_END                         ;  READ RINGING
ERG_BSET:
            MOV     A,#1                            ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            JMP     ERG_END                         ;
ERG_SET:
            MOVX    A,@DPTR                         ;
            SETB    A.1                             ;  SET RINGING
            MOVX    @DPTR,A                         ;
            JMP     ERG_END                         ;
ERG_RESET:
            MOVX    A,@DPTR                         ;
            CLR     A.1                             ; RESET RINGING
            MOVX    @DPTR,A                         ;
ERG_END:
          POP       DPL
          POP       DPH
           POP      0
           POP      B
           RET

;*************************************************************************
;**                    [ EXT RING BACK CONTROL ]  A=0;READ,A.7;SET,A.6;RESET *
;*************************************************************************

EXT_RING_BACK:
            PUSH    B
            PUSH    0
            PUSH     DPH
            PUSH     DPL

            MOV     DPTR,#EXT01
            MOV     R0,A
            ANL     A,#0FH
            MOV     B,#4
            MUL     AB
            MOV     B,A

            MOV     A,DPL                           ;  ADD DPTR,(EXT X 4)
            ADD     A,B                             ;
            MOV     DPL,A                           ;
            MOV     A,DPH                           ;
            ADDC    A,#00                           ;
            MOV     DPH,A                           ;
            MOV     A,R0
            JB      A.7,ERB_SET
            JB      A.6,ERB_RESET

            MOVX    A,@DPTR                         ;
            JB      A.2,ERB_BSET                    ;
            MOV     A,#0                            ;
            JMP     ERB_END                         ; READ RING BACK
ERB_BSET:                                           ;
            MOV     A,#1                            ;
            JMP     ERB_END                         ;
ERB_SET:
            MOVX    A,@DPTR                         ;
            SETB    A.2                             ; SET RING BACK
            MOVX    @DPTR,A                         ;
            JMP     ERB_END                         ;
ERB_RESET:
            MOVX    A,@DPTR                         ;
            CLR     A.2                             ; RESET RING BACK
            MOVX    @DPTR,A                         ;
ERB_END:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          POP       DPL
          POP       DPH
           POP      0
           POP      B
           RET

;*****************************************************************************
;**                        [ EXT ACTIVE CONTROL ]  A=0;READ,A.7;SET,A.6;RESET *
;*****************************************************************************

EXT_ACTIVE:
           PUSH     B
           PUSH     0
           PUSH      DPH
           PUSH      DPL


           MOV      DPTR,#EXT01
           MOV      R0,A
           ANL      A,#0FH
           MOV      B,#4
           MUL      AB
           MOV      B,A

           MOV      A,DPL                    ;  ADD DPTR,(EXT X 4)
           ADD      A,B                      ;
           MOV      DPL,A                    ;
           MOV      A,DPH                    ;
           ADDC     A,#00                    ;
           MOV      DPH,A                    ;
           MOV      A,R0
           JB       A.7,EA_SET
           JB       A.6,EA_RESET

           MOVX     A,@DPTR                  ;
           JB       A.7,EA_BSET              ;
           MOV      A,#0                     ;
           JMP      EA_END                   ; READ ACTIVE
EA_BSET:                                     ;
           MOV      A,#1                     ;
           JMP      EA_END                   ;
EA_SET:
           MOVX     A,@DPTR                  ;
           SETB     A.7                      ; SET ACTIVE
           MOVX     @DPTR,A                  ;
           JMP      EA_END                   ;
EA_RESET:
           MOVX     A,@DPTR                  ;
           CLR      A.7                      ; RESET ACTIVE
           MOVX     @DPTR,A                  ;
EA_END:
          POP       DPL
          POP       DPH
           POP      0
           POP      B
           RET


;*****************************************************************************
;**                        [ EXT LONG CALL CONTROL ]  A=0;READ,A.7;SET,A.6;RESET *
;*****************************************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
EXT_LONG_CALL:
            PUSH    B
            PUSH    0
            PUSH     DPH
            PUSH     DPL

            MOV     DPTR,#EXT01
            MOV     R0,A
            ANL     A,#0FH
            MOV     B,#4
            MUL     AB
            MOV     B,A

            MOV     A,DPL                       ;  ADD DPTR,(EXT X 4)
            ADD     A,B                         ;
            MOV     DPL,A                       ;
            MOV     A,DPH                       ;
            ADDC    A,#00                       ;
            MOV     DPH,A                       ;
            MOV     A,R0
            JB      A.7,ELC_SET
            JB      A.6,ELC_RESET

            MOVX    A,@DPTR                     ;
            JB      A.6,ELC_BSET                ;
            MOV     A,#0                        ;
            JMP     ELC_END                     ; READ ACTIVE
ELC_BSET:                                       ;
            MOV     A,#1                        ;
            JMP     ELC_END                     ;
ELC_SET:
            MOVX    A,@DPTR                     ;
            SETB    A.6                         ; SET ACTIVE
            MOVX    @DPTR,A                     ;
            JMP     ELC_END                     ;
ELC_RESET:
            MOVX    A,@DPTR                     ;
            CLR     A.6                         ; RESET ACTIVE
            MOVX    @DPTR,A                     ;
ELC_END:
          POP        DPL
          POP        DPH
           POP      0
           POP      B
           RET

;*************************************************************************
;**                    [ EXT LINK CONTROL ]   A=0;READ,A.7;SET,A.6;RESET      *
;*************************************************************************

EXT_LINK:
            PUSH    B
            PUSH    0
            PUSH     DPH
            PUSH     DPL

            MOV     DPTR,#EXT01
            INC     DPTR
            MOV     R0,A
            ANL     A,#0FH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV     B,#4
            MUL     AB
            MOV     B,A

            MOV     A,DPL                           ;  ADD DPTR,(EXT X 4)
            ADD     A,B                             ;
            MOV     DPL,A                           ;
            MOV     A,DPH                           ;
            ADDC    A,#00                           ;
            MOV     DPH,A                           ;
            MOV     A,R0
            JB      A.7,EL_SET
            JB      A.6,EL_RESET

            MOVX    A,@DPTR                         ;
            JB      A.7,EL_BSET                     ;
            MOV     A,#0                            ;
            JMP     EL_END                          ; READ ACTIVE
EL_BSET:                                            ;
            MOV     A,#1                            ;
            JMP     EL_END                          ;
EL_SET:
            MOVX    A,@DPTR                         ;
            SETB    A.7                             ; SET ACTIVE
            MOVX    @DPTR,A                         ;
            JMP     EL_END                          ;
EL_RESET:
            MOVX    A,@DPTR                         ;
            CLR     A.7                             ; RESET ACTIVE
            MOVX    @DPTR,A                         ;
EL_END:
           POP      DPL
           POP      DPH
            POP     0
            POP     B
            RET

;*************************************************************************>
;**              [ EXT DTMF ENABLE CONTROL ]  A=0;READ,A.7;SET,A.6;RESET  *>
;*************************************************************************>

EXT_DTMF_ENABLE:
            PUSH    B
            PUSH    0
            PUSH     DPH
            PUSH     DPL

            MOV     DPTR,#EXT01
            MOV     R0,A
            ANL     A,#0FH
            MOV     B,#4
            MUL     AB
            MOV     B,A

            MOV     A,DPL                           ;  ADD DPTR,(EXT X 4)
            ADD     A,B                             ;
            MOV     DPL,A                           ;
            MOV     A,DPH                           ;
            ADDC    A,#00                           ;
            MOV     DPH,A                           ;
            MOV     A,R0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            JB      A.7,EDE_SET
            JB      A.6,EDE_RESET

            MOVX    A,@DPTR                         ;
            JB      A.5,EDE_BSET                    ;
            MOV     A,#0                            ;
            JMP     EDE_END                         ; READ EXT DTMF ENABLE
EDE_BSET:                                           ;
            MOV     A,#1                            ;
            JMP     EDE_END                         ;
EDE_SET:
            MOVX    A,@DPTR                         ;
            SETB    A.5                             ; SET EXT DTMF ENABLE
            MOVX    @DPTR,A                         ;
            JMP     EDE_END                         ;
EDE_RESET:
            MOVX    A,@DPTR                         ;
            CLR     A.5                             ; RESET EXT DTMF ENABLE
            MOVX    @DPTR,A                         ;
EDE_END:
          POP       DPL
          POP       DPH
           POP      0
           POP      B
           RET

;*************************************************************************
;**                        [ EXT MEDIA CONTROL ]  A.7=0,READ A.7=1,WRITE  *
;*************************************************************************
EXT_MEDIA:
          PUSH      B
          PUSH      0
           PUSH     DPH
           PUSH     DPL

          MOV       DPTR,#EXT01
          MOV       R0,A
          ANL       A,#0FH
          MOV       B,#4
          MUL       AB
          MOV       B,A

          MOV       A,DPL                           ;  ADD DPTR,(EXT X 4)
          ADD       A,B                             ;
          MOV       DPL,A                           ;
          MOV       A,DPH                           ;
          ADDC      A,#00                           ;
          MOV       DPH,A                           ;
          INC       DPTR
          INC       DPTR
          MOV       A,R0
          JB        A.7,EM_WRITE

          MOVX      A,@DPTR                         ;
          ANL       A,#7                            ;
          JMP       EM_END                          ; READ ACTIVE
EM_WRITE:
          MOV       A,R0                            ; MOV MEDIA TO 0-2
          SWAP      A                               ;
          ANL       A,#7                            ;
          MOV       R0,A                            ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOVX        A,@DPTR
            ANL         A,#F8H
            ORL         A,R0
            MOVX        @DPTR,A
EM_END:
            POP         DPL
            POP         DPH
            POP         0
            POP         B
            RET

;*****************************************************************************
;**                              [ EXT SHIFT CONTROL ]  A.7=0,READ A.7=1,WRITE  *
;*****************************************************************************
EXT_SHIFT:
            PUSH        B
            PUSH        0
             PUSH       DPH
             PUSH       DPL

            MOV         DPTR,#EXT01
            MOV         R0,A
            ANL         A,#0FH
            MOV         B,#4
            MUL         AB
            MOV         B,A

            MOV         A,DPL                   ;  ADD DPTR,(EXT X 4)
            ADD         A,B                     ;
            MOV         DPL,A                   ;
            MOV         A,DPH                   ;
            ADDC        A,#00                   ;
            MOV         DPH,A                   ;
            INC         DPTR
            INC         DPTR
            MOV         A,R0
            JB          A.7,ES_WRITE

            MOVX        A,@DPTR                 ;
            RR          A
            RR          A
            RR          A
            ANL         A,#7                    ;
            JMP         ES_END                  ; READ ACTIVE
ES_WRITE:
            MOV         A,R0                    ; MOV MEDIA TO 0-2
            RL          A
            ANL         A,#38H                  ;
            MOV         R0,A                    ;

            MOVX        A,@DPTR
            ANL         A,#C7H
            ORL         A,R0
            MOVX        @DPTR,A
ES_END:
            POP         DPL
            POP         DPH
            POP         0
            POP         B
            RET

;*****************************************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;**                    [ EXT CALL CHANEL CONTROL ]  A.7=0,READ A.7=1,WRITE    **
;*************************************************************************
EXT_CALL:
          PUSH      B                                   ; DATA WRITE IN R1
          PUSH      0
           PUSH     DPH
           PUSH     DPL
                                                        ; DATA READ IN A
          MOV       DPTR,#EXT01                         ; DATA EXT CHANEL IN AL
          MOV       R0,A                                ;
          ANL       A,#0FH                              ;
          MOV       B,#4                                ;
          MUL       AB
          MOV       B,A

          MOV       A,DPL                               ;  ADD DPTR,(EXT X 4)
          ADD       A,B                                 ;
          MOV       DPL,A                               ;
          MOV       A,DPH                               ;
          ADDC      A,#00                               ;
          MOV       DPH,A                               ;
          INC       DPTR
          MOV       A,R0
          JB        A.7,ECC_WRITE

          MOVX      A,@DPTR
          JMP       ECC_END
ECC_WRITE:
          MOV       A,R1
          MOVX      @DPTR,A
ECC_END:
          POP       DPL
          POP       DPH
          POP       0
          POP       B
          RET

;*************************************************************************
;**                    [ EXT NUMBER TO CALL CONTROL ]  A.7=0,READ A.7=1,WRITE **
;*************************************************************************
EXT_NUMBER:
          PUSH      B                                   ; DATA WRITE IN R1
          PUSH      0
           PUSH     DPH
           PUSH     DPL
                                                        ; DATA READ IN A
          MOV       DPTR,#EXT01                         ; DATA EXT CHANEL IN AL
          MOV       R0,A                                ;
          ANL       A,#0FH                              ;
          MOV       B,#4                                ;
          MUL       AB
          MOV       B,A

          MOV       A,DPL                               ;  ADD DPTR,(EXT X 4)
          ADD       A,B                                 ;
          MOV       DPL,A                               ;
          MOV       A,DPH                               ;
          ADDC      A,#00                               ;
          MOV       DPH,A                               ;
          INC       DPTR
          INC       DPTR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          INC       DPTR
          MOV       A,R0
          JB        A.7,ENB_WRITE

          MOVX      A,@DPTR
          JMP       ENB_END
ENB_WRITE:
          MOV       A,R1
          MOVX      @DPTR,A
ENB_END:
          POP       DPL
          POP       DPH
          POP       0
          POP       B
          RET

;*************************************************************************
;**                [ EXT DTMF INPUT CONTROL ]  A.7=0,READ A.7=1,WRITE **
;*************************************************************************
EXT_DTMF:
          PUSH      B                          ; DATA WRITE IN R1
          PUSH      0                          ; DATA READ IN A
           PUSH     DPH
           PUSH     DPL

          MOV       DPTR,#EXT01                ; DATA EXT CHANEL IN AL
          MOV       R0,A                       ;
          ANL       A,#0FH                     ;
          MOV       B,#4                       ;
          MUL       AB
          MOV       B,A

          MOV       A,DPL                      ;  ADD DPTR,(EXT X 4)
          ADD       A,B                        ;
          MOV       DPL,A                      ;
          MOV       A,DPH                      ;
          ADDC      A,#00                      ;
          MOV       DPH,A                      ;
          INC       DPTR
          INC       DPTR
          INC       DPTR
          INC       DPTR
          MOV       A,R0
          JB        A.7,EDF_WRITE

          MOVX      A,@DPTR
          JMP       EDF_END
EDF_WRITE:
          MOV       A,R1
          MOVX      @DPTR,A
EDF_END:
          POP       DPL
          POP       DPH
          POP       0
          POP       B
          RET

;*************************************************************************
;**                          [ EXT STATUS ]                             **
;*************************************************************************
EXT_STATUS:
```

เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          PUSH      B                                   ;
           PUSH     DPH
           PUSH     DPL
                                                        ; DATA READ IN A
          MOV       DPTR,#EXT01                         ; DATA EXT CHANEL IN AL
          ANL       A,#0FH                              ;
          MOV       B,#4                                ;
          MUL       AB
          MOV       B,A

          MOV       A,DPL                               ;  ADD DPTR,(EXT X 4)
          ADD       A,B                                 ;
          MOV       DPL,A                               ;
          MOV       A,DPH                               ;
          ADDC      A,#00                               ;
          MOV       DPH,A                               ;

          MOVX      A,@DPTR

          POP       DPL
          POP       DPH
          POP       B
          RET

;*************************************************************************
;**                  [ EXT MEDIA ASSIGN ]  A=0;READ,A.7;SET,A.6;RESET    **
;*************************************************************************

EXT_ASSIGN:
          PUSH      B
          PUSH      0

          MOV       DPTR,#EXT01
          MOV       R0,A
          ANL       A,#0FH
          MOV       B,#4
          MUL       AB
          MOV       B,A

          MOV       A,DPL                               ;  ADD DPTR,(EXT X 4)
          ADD       A,B                                 ;
          MOV       DPL,A                               ;
          MOV       A,DPH                               ;
          ADDC      A,#00                               ;
          MOV       DPH,A                               ;
          INC       DPTR
          INC       DPTR

          MOV       A,R0
          JB        A.7,EAS_SET
          JB        A.6,EAS_RESET

          MOVX      A,@DPTR                             ;
          JB        A.7,EAS_BSET                        ;
          MOV       A,#0                                ;
          JMP       EAS_END                             ; READ ACTIVE
                                                        ;
EAS_BSET:                                               ;
          MOV       A,#1                                ;
          JMP       EAS_END                             ;

EAS_SET:
          MOVX      A,@DPTR                             ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            SETB    A.7                             ; SET ACTIVE
            MOVX    @DPTR,A                         ;
            JMP     EAS_END                         ;
EAS_RESET:
            MOVX    A,@DPTR                         ;
            CLR     A.7                             ; RESET ACTIVE
            MOVX    @DPTR,A                         ;
EAS_END:
            POP     0
            POP     B
            RET

;*************************************************************************
;**                          [ TOT STATUS ]                             *
;*************************************************************************
TOT_STATUS:
            PUSH    B                               ;
                                                    ; DATA READ IN A
            MOV     DPTR,#TOT1                      ; DATA EXT CHANEL IN AL
            ANL     A,#03H                          ;
            MOV     B,#2                            ;
            MUL     AB
            MOV     B,A

            MOV     A,DPL                           ;  ADD DPTR,(EXT X 4)
            ADD     A,B                             ;
            MOV     DPL,A                           ;
            MOV     A,DPH                           ;
            ADDC    A,#00                           ;
            MOV     DPH,A                           ;

            MOVX    A,@DPTR

            POP     B
            RET

;*************************************************************************
;**                [ TOT ACTIVE CONTROL ]  A=0;READ,A.7;SET,A.6;RESET *
;*************************************************************************

TOT_ACTIVE:                                         ;       A1 A0
            PUSH    B                               ;        0  0  = TOT1
            PUSH    0                               ;        0  1  = TOT2
                                                    ;        1  0  = TOT3
            MOV     DPTR,#TOT1                      ;
            MOV     R0,A                            ;
            ANL     A,#03H                          ;
            MOV     B,#2
            MUL     AB
            MOV     B,A

            MOV     A,DPL                           ;  ADD DPTR,TOT CHANEL
            ADD     A,B                             ;
            MOV     DPL,A                           ;
            MOV     A,DPH                           ;
            ADDC    A,#00                           ;
            MOV     DPH,A                           ;
            MOV     A,R0
            JB      A.7,TA_SET
            JB      A.6,TA_RESET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOVX    A,@DPTR                         ;
            JB      A.7,TA_BSET                     ;
            MOV     A,#0                            ;
            JMP     TA_END                          ; READ ACTIVE
TA_BSET:                                            ;
            MOV     A,#1                            ;
            JMP     TA_END                          ;
TA_SET:
            MOVX    A,@DPTR                         ;
            SETB    A.7                             ; SET ACTIVE
            MOVX    @DPTR,A                         ;
            JMP     TA_END                          ;
TA_RESET:
            MOVX    A,@DPTR                         ;
            CLR     A.7                             ; RESET ACTIVE
            MOVX    @DPTR,A                         ;
TA_END:
            POP     0
            POP     B
            RET

;*************************************************************************
;**                  [ TOT RING BACK CONTROL ]  A=0;READ,A.7;SET,A.6;RESET *
;*************************************************************************

TOT_RING_BACK:                                      ;       A1 A0
            PUSH    B                               ;        0  0  = TOT1
            PUSH    0                               ;        0  1  = TOT2
                                                    ;        1  0  = TOT3
            MOV     DPTR,#TOT1                      ;
            MOV     R0,A                            ;
            ANL     A,#03H                          ;
            MOV     B,#2
            MUL     AB
            MOV     B,A

            MOV     A,DPL                           ;  ADD DPTR,TOT CHANEL
            ADD     A,B                             ;
            MOV     DPL,A                           ;
            MOV     A,DPH                           ;
            ADDC    A,#00                           ;
            MOV     DPH,A                           ;
            MOV     A,R0
            JB      A.7,TRB_SET
            JB      A.6,TRB_RESET

            MOVX    A,@DPTR                         ;
            JB      A.6,TRB_BSET                    ;
            MOV     A,#0                            ;
            JMP     TRB_END                         ; READ ACTIVE
TRB_BSET:                                           ;
            MOV     A,#1                            ;
            JMP     TRB_END                         ;
TRB_SET:
            MOVX    A,@DPTR                         ;
            SETB    A.6                             ; SET ACTIVE
            MOVX    @DPTR,A                         ;
            JMP     TRB_END                         ;
TRB_RESET:
            MOVX    A,@DPTR                         ;
            CLR     A.6                             ; RESET ACTIVE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOVX    @DPTR,A                         ;
TRB_END:
            POP     0
            POP     B
            RET

;*************************************************************************
;**                    [ TOT BUSY CONTROL ]  A=0;READ,A.7;SET,A.6;RESET *
;*************************************************************************

TOT_BUSY:                                           ;       A1 A0
            PUSH    B                               ;        0  0  = TOT1
            PUSH    0                               ;        0  1  = TOT2
                                                    ;        1  0  = TOT3
            MOV     DPTR,#TOT1                      ;
            MOV     R0,A                            ;
            ANL     A,#03H                          ;
            MOV     B,#2
            MUL     AB
            MOV     B,A

            MOV     A,DPL                           ;  ADD DPTR,TOT CHANEL
            ADD.    A,B                             ;
            MOV     DPL,A                           ;
            MOV     A,DPH                           ;
            ADDC    A,#00                           ;
            MOV     DPH,A                           ;
            MOV     A,R0
            JB      A.7,TBS_SET
            JB      A.6,TBS_RESET

            MOVX    A,@DPTR                         ;
            JB      A.5,TBS_BSET                    ;
            MOV     A,#0                            ;
            JMP     TBS_END                         ; READ ACTIVE
TBS_BSET:                                           ;
            MOV     A,#1                            ;
            JMP     TBS_END                         ;
TBS_SET:
            MOVX    A,@DPTR                         ;
            SETB    A.5                             ; SET ACTIVE
            MOVX    @DPTR,A                         ;
            JMP     TBS_END                         ;
TBS_RESET:
            MOVX    A,@DPTR                         ;
            CLR     A.5                             ; RESET ACTIVE
            MOVX    @DPTR,A                         ;
TBS_END:
            POP     0
            POP     B
            RET


;*************************************************************************
;**                     [ TOT CHANEL CONTROL ]  A=0;READ,A.7;SET,A.6;RESET   **
;*************************************************************************

TOT_CHANEL:                                         ;       A1 A0
            PUSH    B                               ;        0  0  = TOT1
            PUSH    0                               ;        0  1  = TOT2
                                                    ;        1  0  = TOT3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     DPTR,#TOT1
        MOV     R0,A
        ANL     A,#03H
        MOV     B,#2
        MUL     AB
        MOV     B,A

        MOV     A,DPL                   ;  ADD DPTR,TOT CHANEL
        ADD     A,B
        MOV     DPL,A
        MOV     A,DPH
        ADDC    A,#00
        MOV     DPH,A
        MOV     A,R0
        JB      A.7,TCH_SET
        JB      A.6,TCH_RESET

        MOVX    A,@DPTR
        JB      A.4,TCH_BSET
        MOV     A,#0                    ; READ ACTIVE
        JMP     TCH_END
TCH_BSET:
        MOV     A,#1
        JMP     TCH_END
TCH_SET:
        MOVX    A,@DPTR                 ; SET ACTIVE
        SETB    A.4
        MOVX    @DPTR,A
        JMP     TCH_END
TCH_RESET:
        MOVX    A,@DPTR                 ; RESET ACTIVE
        CLR     A.4
        MOVX    @DPTR,A
TCH_END:
        POP     0
        POP     B
        RET

;*************************************************************************
;**              [ TOT DTMF ENABLE CONTROL ]  A=0;READ,A.7;SET,A.6;RESET
;*************************************************************************

TOT_DTMF_FLAG:                          ;        A1 A0
        PUSH    B                       ;         0  0  = TOT1
        PUSH    0                       ;         0  1  = TOT2
                                        ;         1  0  = TOT3
        MOV     DPTR,#TOT1
        MOV     R0,A
        ANL     A,#03H
        MOV     B,#2
        MUL     AB
        MOV     B,A

        MOV     A,DPL                   ;  ADD DPTR,TOT CHANEL
        ADD     A,B
        MOV     DPL,A
        MOV     A,DPH
        ADDC    A,#00
        MOV     DPH,A
        INC     DPTR
        MOV     A,R0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          JB        A.7,TDFF_SET
          JB        A.6,TDFF_RESET

          MOVX      A,@DPTR                         ;
          JB        A.7,TDFF_BSET                    ;
          MOV       A,#0                            ;
          JMP       TDFF_END                         ; READ ACTIVE
TDFF_BSET:                                          ;
          MOV       A,#1                            ;
          JMP       TDFF_END                         ;
TDFF_SET:
          MOVX      A,@DPTR                         ;
          SETB      A.7                             ; SET ACTIVE
          MOVX      @DPTR,A                         ;
          JMP       TDFF_END                         ;
TDFF_RESET:
          MOVX      A,@DPTR                         ;
          CLR       A.7                             ; RESET ACTIVE
          MOVX      @DPTR,A                         ;
TDFF_END:
          POP       0
          POP       B
          RET

;*************************************************************************
;**                  [ TOT DTMF INPUT CONTROL ]  A.7=0,READ A.7=1,WRITE     *
;*************************************************************************
TOT_DTMF:
          PUSH      B                               ; DATA WRITE IN R1
          PUSH      0                               ; DATA READ IN A
          MOV       DPTR,#TOT1                      ; DATA EXT CHANEL IN AL
          MOV       R0,A                            ;
          ANL       A,#03H                          ;
          MOV       B,#2                            ;
          MUL       AB
          MOV       B,A

          MOV       A,DPL                           ;  ADD DPTR,(EXT X 4)
          ADD       A,B                             ;
          MOV       DPL,A                           ;
          MOV       A,DPH                           ;
          ADDC      A,#00                           ;
          MOV       DPH,A                           ;
          INC       DPTR
          INC       DPTR
          MOV       A,R0
          JB        A.7,TDF_WRITE

          MOVX      A,@DPTR
          JMP       TDF_END
TDF_WRITE:
          MOV       A,R1
          MOVX      @DPTR,A
TDF_END:
          POP       0
          POP       B
          RET

;*************************************************************************
;**  เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ[ TOT CALL_EXT CONTROL ]  A.7=0,READ A.7=1,WRITE  *
;*************************************************************************
```

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
TOT_CALL_EXT:
          PUSH      B
          PUSH      0

          MOV       DPTR,#EXT01
          MOV       R0,A
          ANL       A,#03H
          MOV       B,#2
          MUL       AB
          MOV       B,A

          MOV       A,DPL                          ;  ADD DPTR,(EXT X 4)
          ADD       A,B                            ;
          MOV       DPL,A                          ;
          MOV       A,DPH                          ;
          ADDC      A,#00                          ;
          MOV       DPH,A                          ;
          INC       DPTR
          MOV       A,R0
          JB        A.7,TCE_WRITE

          MOVX      A,@DPTR                        ;
          ANL       A,#0FH                         ;
          JMP       TCE_END                        ; READ ACTIVE
TCE_WRITE:
          MOV       A,R1                           ; MOV MEDIA TO 0-2
          ANL       A,#0FH                         ;
          MOV       R0,A                           ;

          MOVX      A,@DPTR
          ANL       A,#F0H
          ORL       A,R0
          MOVX      @DPTR,A
TCE_END:
          POP       0
          POP       B
          RET

;*****************************************************************************
;**                          [ DATA PATTERNS ]                                **
;*****************************************************************************

;--------------------------MODE 00-----------------
MENU00:   DB   '    PABX INITIAL   ]'
          DB   '  KMITL VERSION 01  '

;--------------------------MODE 01-----------------
MENU01:   DB   '         HALT        ]'
          DB   '                      '

;--------------------------MODE 02-----------------
MENU02:   DB   '        RUNNING       '
          DB   'CHN:__________________'

;--------------------------MODE 03-----------------
MENU03:   DB   '   SWITCH SET DIAL   '
          DB   'CHN:__________________'

;--------------------------MODE 04-----------------
MENU04:   DB   '   SWITCH SET BUSY   '
          DB   'CHN:__________________'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;--------------------------MODE 05-----------------
MENU05:     DB   'SWITCH SET RING BACK'
            DB   'CHN:__________________'

;--------------------------MODE 06-----------------
MENU06:     DB   '  RINGING CONTROL  ]'
            DB   'CHN:__________________'

;--------------------------MODE 07-----------------
MENU07:     DB   '  TOT LINE CONTROL   '
            DB   'CHN:___ A:0 B:0 C:0  '

;--------------------------MODE 08-----------------
MENU08:     DB   'CLK:00 INT[          ]'
            DB   'DTMF1 OSW.00 IS OFF  '

;--------------------------MODE 09-----------------
MENU09:     DB   '     PABX INITIAL    ]'
            DB   '  KMITL VERSION 01   '

;--------------------------MODE 0A-----------------
MENU0A:     DB   '          HALT       ]'
            DB   '                     '

;--------------------------MODE 0B-----------------
MENU0B:     DB   '        RUNNING      '
            DB   'CHN:__________________'

;--------------------------MODE 0C-----------------
MENU0C:     DB   '   SWITCH SET DIAL   '
            DB   'CHN:__________________'

;--------------------------MODE 0D-----------------
MENU0D:     DB   '   SWITCH SET BUSY   '
            DB   'CHN:__________________'

;--------------------------MODE 0E-----------------
MENU0E:     DB   'SWITCH SET RING BACK'
            DB   'CHN:__________________'

;--------------------------MODE 0F-----------------
MENU0F:     DB   '  RINGING CONTROL  ]'
            DB   'CHN:__________________'

;--------------------------MODE 10-----------------
MENU10:     DB   '  TOT LINE CONTROL   '
            DB   'CHN:___ A:0 B:0 C:0  '

EXT01:      DB   C0H
            DB   00H
            DB   00H
            DB   11H

EXT02:      DB   C0H
            DB   00H
            DB   00H
            DB   12H

EXT03:      DB   C0H
            DB   00H
            DB   00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DB   13H

EXT04:  DB   C0H
        DB   00H
        DB   00H
        DB   14H

EXT05:  DB   C0H
        DB   00H
        DB   00H
        DB   15H

EXT06:  DB   C0H
        DB   00H
        DB   00H
        DB   16H

EXT07:  DB   C0H
        DB   00H
        DB   00H
        DB   17H

EXT08:  DB   C0H
        DB   00H
        DB   00H
        DB   18H

EXT09:  DB   C0H
        DB   00H
        DB   00H
        DB   19H

EXT10:  DB   C0H
        DB   00H
        DB   00H
        DB   20H

EXT11:  DB   C0H
        DB   00H
        DB   00H
        DB   21H

EXT12:  DB   C0H
        DB   00H
        DB   00H
        DB   22H

EXT13:  DB   C0H
        DB   00H
        DB   00H
        DB   23H

EXT14:  DB   C0H
        DB   00H
        DB   00H
        DB   24H

EXT15:  DB   C0H
        DB   00H
        DB   00H
        DB   25H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
EXT16:      DB   C0H
            DB   00H
            DB   00H
            DB   26H

TOT1:       DB   80H

TOT2:       DB   80H

TOT3:       DB   80H

DATA1:      DB   01000B
            DB   00000B
            DB   00000B
            DB   11111B
            DB   11011B
            DB   00000B
            DB   01110B
            DB   11111B

DATA2:      DB   01110B
            DB   01110B
            DB   00010B
            DB   01110B
            DB   01110B
            DB   00000B
            DB   01110B
            DB   11111B

            END
;*******************************[ END OF PROGRAM ]*******************************
;**                                                                             *
;**                      ******   **  **   *****                               *
;**                      **       *** **   ** ***                              *
;**                      ****     ******   **  **                              *
;**                      **       ** ***   ** ***                              *
;**                      ******   **  **   *****                               *
;**                                                                             *
;*********************************************************************************


;*********************************************************************************
;**                              [          ]                                   *
;*********************************************************************************
            :
            PUSH     A
            PUSH     B
            PUSH     0
            PUSH     1
            PUSH     2
            PUSH     DPH
            PUSH     DPL



            POP      DPL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค. คู่มือการทดสอบและโปรแกรมการทดสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดสอบเครื่องด้วย Key Board

1. การทดสอบ Signal Busy, Dial, Ringback tone to Subscriber 0-15 พิจารณาที่ Display ส่วน Array Switch

SW :	O O	: OFF

- ตัวเลขหลักที่ 1 คือ รหัสสัญญาณ
  - 3 คือ Dial Signal
  - 4 คือ Ringback tone
  - 5 คือ Busy Signal
- ตัวเลขหลักที่ 2 คือ รหัส Subscriber
  - O - F Subscriber

1.1 Busy to Sub scriber

ขั้นตอน
1. กด Key เลือกรหัส
2. ที่ Display ส่วน Array Switch (5X)
3. โดย X คือ Sub ที่ต้องการ busy
4. กด On Array Switch

1.2 Ringback tone to Subscriber

ขั้นตอน
1. กด Key เลือกรหัส
2. ที่ Display ส่วน Array Switch (4X)
3. โดย X คือ Sub ที่ต้องการติดต่อ
4. กด On Array Switch

1.3 Dial to Subscriber

ขั้นตอน
1. กด Key เลือกรหัส
2. ที่ Display ส่วน Array Switch (3X)
3. โดย X คือ Sub ที่ต้องการติดต่อ
4. กด On Array Switch

หมายเหตุ : การยกเลิกสัญญาณไปยัง Subscriber มีขั้นตอนที่ 1-3 ส่วนขั้นตอนที่ 4 กด Off Array Switch

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. การต่อ Subscriber 2 เครื่องเข้าด้วยกัน

พิจารณาที่ Disp;ay ส่วน Array Switch

SW :	X X	: OFF

- X ตัวที่ 1 คือ รหัส Channal กลางมี 9-F
- X ตัวที่ 2 คือ รหัส Subscriber มี 0-F

ตัวอย่าง การต่อ Subscriber 1 กับ Subscriber 2

1. เลือกรหัส Channal กลาง 9
2. เลือกรหัส Subscriber 1

Display

SW :	9 1	: OFF

3. กด On Array Switch

SW :	9 1	: ON

4. เลือกรหัส Subscriber

Display

SW :	9 2	: OFF

5. กด On Array Switch
6. สามารถติดต่อ Subscriber 1 และ Subscriber 2 เข้าด้วยกัน

3. การต่อ Subscriber 0-15 เข้ากับ TOT

1. กด TOT 1, 2, 3 Toggle
2. เลือก Subscriber ที่ต้องการติดต่อ TOT

พิจารณา Display ส่วน Array Switch

SW :	X X	: OFF

- X หลักที่ 1 คือ รหัส Channal กลาง TOT มี 0 = TOT 1, 1 = TOT 2, 2 = TOT 3
- X หลักที่ 2 คือ รหัส Subscriber 0 - F

3. กด ON Array Switch
4. สามารถติดต่อ Subscriber เข้ากับ TOT ได้

4. การต่อ Ringing เข้ากับ Subscriber

พิจารณา Display ส่วน Ringing

Ring : X : OFF

X คือ รหัส Subscriber 0 - F

1. เลือก Subscriber ที่ต้องการ Ringing เช่น 1

Ring : 1 : OFF

2. กด ON Ringing

Ring : 1 : ON

3. กด OFF Ringing

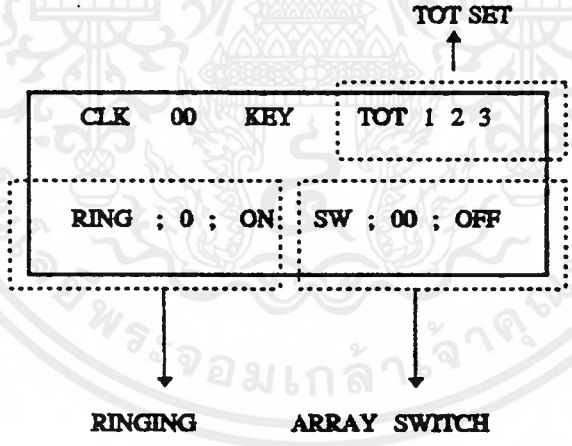
Ring : 1 : OFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Key Board

TOT 1 TOGGLE	+ 1 FOR RINGING	+10 FOR ARRAY SWITCH	+1 FOR ARRAY SWITCH
TOT 2 TOGGLE	-1 FOR RINGING	-10 FOR ARRAY SWITCH	-1 FOR ARRAY SWITCH
TOT 3 TOGGLE	ON RINGING	CLR ALL	ON ARRAY SWITCH
	OFF RINGING	SET 00	OFF ARRAY SWITCH

Display



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            RETI
;-------------------------------------------------------------------------------
;*****************************[ START PROGRAM ]*********************************
;          * * *       * * * * *         *         * * * *       * * * * *
;        *       *         *           *   *       *       *         *
;                *         *         *     *     *       *         *
;      \   * * *           *         * * * * *     * * * *           *
;        *                 *         *       *     *       *         *
;        *       *         *         *       *     *       *         *
;          * * *           *         *       *     *       *         *
;*******************************************************************************

ST:
            LCALL   INIT_DATA_DIS
            LCALL   INIT_PORT

            LCALL   INIT
            LCALL   INIT_INT
            LCALL   CLR_ALL_MAP_SW

            LCALL   CLRSCR
            LCALL   PRINTF
READ_KEY:
            MOV     A,KEY
            JNB     A.7,JM_END
            CLR     A.7
            MOV     KEY,#0

            ANL     A,#0FH

            CJNE    A,#3,KEY_1
            INC     ADDR_SW
JM_END:     JMP     END_KEY

KEY_1:      CJNE    A,#7,KEY_2
            DEC     ADDR_SW
            JMP     END_KEY

KEY_2:      CJNE    A,#14,KEY_3
            MOV     ADDR_SW,#0
            JMP     END_KEY

KEY_3:      CJNE    A,#15,KEY_4
            LCALL   CLR_MAP_SW
            JMP     END_KEY

KEY_4:      CJNE    A,#11,KEY_5
            LCALL   SET_MAP_SW
            JMP     END_KEY

KEY_5:      CJNE    A,#6,KEY_6
            MOV     A,ADDR_SW
            SUBB    A,#10H
            MOV     ADDR_SW,A
            JMP     END_KEY

KEY_6:      CJNE    A,#2,KEY_7
            MOV     A,ADDR_SW
            ADD     A,#10H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*************************************************************************
;
;                TEST PROGRAM      01
;
;*************************************************************************
;***************************[ DATA EQU DEFINE ]***************************

COMMAND     EQU     0E060H     ; Read-Write Register
READBUSY    EQU     0E061H     ; Read BF(Busy Flag) and address
WRITEDATA   EQU     0E062H     ; Write character
READDATA    EQU     0E063H     ; Read Data from DD ram
ADDR_DATA   EQU     00000H     ; DATA DISPLAY IN RAM

PORT_A      EQU     0E0A0H     ; PORT A IN 8255 NO.2
PORT_B      EQU     0E0A1H     ; PORT B IN 8255 NO.2
PORT_C      EQU     0E0A2H     ; PORT C IN 8255 NO.2
PORT_COM    EQU     0E0A3H     ; PORT CONTROL IN 8255 NO.2
INTERF_A    EQU     0E000H
INTERF_B    EQU     0E001H
INTERF_C    EQU     0E002H
INTERF_COM  EQU     0E003H
SUB1_A      EQU     0E004H
SUB1_B      EQU     0E005H
SUB1_C      EQU     0E006H
SUB1_COM    EQU     0E007H
SUB2_A      EQU     0E008H
SUB2_B      EQU     0E009H
SUB2_C      EQU     0E00AH
SUB2_COM    EQU     0E00BH
TOT1_A      EQU     0E00CH
TOT1_B      EQU     0E00DH
TOT1_C      EQU     0E00EH
TOT1_COM    EQU     0E00FH
TOT2_A      EQU     0E010H
TOT2_B      EQU     0E011H
TOT2_C      EQU     0E012H
TOT2_COM    EQU     0E013H
SW1_A       EQU     0E014H
SW1_B       EQU     0E015H
SW1_C       EQU     0E016H
SW1_COM     EQU     0E017H
SW2_A       EQU     0E018H
SW2_B       EQU     0E019H
SW2_C       EQU     0E01AH
SW2_COM     EQU     0E01BH
SW3_A       EQU     0E01CH
SW3_B       EQU     0E01DH
SW3_C       EQU     0E01EH
SW3_COM     EQU     0E01FH

DPL0        EQU     30H        ; DPL BACKUP NUMBER 0
DPH0        EQU     31H        ; DPH BACKUP NUMBER 0
DPL1        EQU     32H        ; DPL BACKUP NUMBAE 1
DPH1        EQU     33H        ; DPH BACKUP NUMBER 1
GEN0        EQU     34H        ;  \
GEN1        EQU     35H        ;   |
GEN2        EQU     36H        ;   |
GEN3        EQU     37H        ;    \   VARIABLE DATA
GEN4        EQU     38H        ;    /   IN MEMORY 34H TO 3BH
GEN5        EQU     39H        ;   |
GEN6        EQU     3AH        ;   |
GEN7        EQU     3BH        ;  /
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
TIME        EQU     3DH         ; TIMER COUNTER
INT0        EQU     3EH         ; INTERUP 0 COUNTER
INT1        EQU     3FH         ; INTERUP 1 COUNTER
CHL         EQU     40H         ; LOW ASCII FOR HEX2ASCII
CHH         EQU     41H         ; HI ASCII FOR HEX2ASCII
DISPLAY     EQU     43H         ; DISPLAY COUNTER
KEY         EQU     44H         ; INPUT KEY BIT 4 SET IF HAS INPUT KEY
ADDR_SW     EQU     45H         ; SWITCH POINTER IN MAP SWITCH
ADDR_RING   EQU     46H         ; RINGING ADDR CONTROL
RING_D      EQU     47H         ; RINGING DATA
; T0        EQU     48H         ; 2 BYTE
TOT1_F      EQU     49H         ; FLAG FOR TOT1
TOT2_F      EQU     4AH         ; FLAG FOR TOT2
TOT3_F      EQU     4BH         ; FLAG FOR TOT3
MAP_SW      EQU     60H         ; DATA FOR MAP SWITCH ADDR 60H TO 7FH =32 BYTE


;*****************************[ INTERRUPT VECTOR ]*****************************
            ORG     0000H
            JMP     ST

            ORG     0003H
            JMP     INT_IE0

            ORG     000BH
            JMP     INT_TF0

            ORG     0013H
            JMP     INT_IE1

            ORG     001BH
            JMP     INT_TF1

            ORG     0023H
            JMP     INT_COM

            ORG     002BH
            JMP     INT_EXT

            ORG 0030H

;************************[ INITORRUPT SERVICE RUTENE ]*********************
INT_IE0:
            CLR     IE.7
            PUSH    DPL
            PUSH    DPH
            PUSH    A

            MOV     DPTR,#INTERF_B
            MOV     A,#00
            MOVX    @DPTR,A
            MOV     A,#FFH
            MOVX    @DPTR,A
            MOV     DPTR,#INTERF_C
            MOVX    A,@DPTR
            SETB    A.7
            MOV     KEY,A

            POP     A
            POP     DPH
            POP     DPL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            SETB    IE.7

            RETI                ;
;-------------------------------------------------------------------------------
INT_TF0:
            RETI                ;
;-------------------------------------------------------------------------------
INT_IE1:                        ;
            RETI
;-------------------------------------------------------------------------------
INT_TF1:
            RETI
;-------------------------------------------------------------------------------
INT_COM:
            RETI
;-------------------------------------------------------------------------------
INT_EXT:
            CLR     IE.7
            PUSH    DPL
            PUSH    DPH
            PUSH    A

            DB      C2H,CFH     ; CLR T2CON.7
            INC     TIME
            MOV     A,TIME

            CJNE    A,#5H,EXT00  ; NUMBER OF TURN PER SCAN 1 TURN = 1/16 SEC

            MOV     DPTR,#ADDR_DATA+4

            INC     DISPLAY

            MOV     A,DISPLAY
            MOV     P1,A                        ; PRINT CLK
            LCALL   HEX2ASCII

            MOV     A,CHH
            MOVX    @DPTR,A
            INC     DPTR
            MOV     A,CHL
            MOVX    @DPTR,A

            MOV     DPTR,#INTERF_C              ;PRINT KEY
            MOVX    A,@DPTR
            LCALL   HEX2ASCII
            MOV     DPTR,#ADDR_DATA+11
            MOV     A,CHL
            MOVX    @DPTR,A

            LCALL   PRINTF                      ; PRINT DISPLAY
            MOV     TIME,#00

            MOV         DPTR,#INTERF_B
            MOV         A,#00H
            MOVX        @DPTR,A
            MOV         A,#FFH
            MOVX        @DPTR,A

EXT00:      POP     A
            POP     DPH
            POP     DPL
            SETB    IE.7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     ADDR_SW,A
                JMP     END_KEY

KEY_7:          CJNE    A,#10,KEY_8
                LCALL   CLR_ALL_MAP_SW
                JMP     END_KEY

KEY_8:          CJNE    A,#0CH,KEY_9
                LCALL   SAND_VOICE                      ; END
                JMP     END_KEY

KEY_9:          CJNE    A,#0,KEY_A
                LCALL   TOT1_TOG
                JMP     END_KEY

KEY_A:          CJNE    A,#4,KEY_B
                LCALL   TOT2_TOG
                JMP     END_KEY

KEY_B:          CJNE    A,#8,KEY_C
                LCALL   TOT3_TOG
                JMP     END_KEY

KEY_C:          CJNE    A,#1,KEY_D
                INC     ADDR_RING
                JMP     END_KEY

KEY_D:          CJNE    A,#5,KEY_E
                DEC     ADDR_RING
                JMP     END_KEY

KEY_E:          CJNE    A,#9,KEY_F
                LCALL   ON_RING
                JMP     END_KEY

KEY_F:          CJNE    A,#13,END_CHK
                LCALL   OFF_RING
                JMP     END_KEY
END_CHK:        JMP     READ_KEY
END_KEY:
                MOV     A,ADDR_SW
                LCALL   HEX2ASCII
                MOV     DPTR,#ADDR_DATA+34
                MOV     A,CHH
                MOVX    @DPTR,A
                INC     DPTR
                MOV     A,CHL
                MOVX    @DPTR,A

                MOV     A,ADDR_SW                       ; PRINT STATUS SWITCH
                LCALL   MEAS_MAP_SW
                CJNE    A,#0,MAIN01

                MOV     DPTR,#ADDR_DATA+37
                MOV     A,#'0'
                MOVX    @DPTR,A
                INC     DPTR
                MOV     A,#'F'
                MOVX    @DPTR,A
                INC     DPTR
                MOV     A,#'F'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOVX    @DPTR,A

            JMP     MAIN02
MAIN01:
            MOV     DPTR,#ADDR_DATA+37
            MOV     A,#'0'
            MOVX    @DPTR,A
            INC     DPTR
            MOV     A,#'N'
            MOVX    @DPTR,A
            INC     DPTR
            MOV     A,#' '
            MOVX    @DPTR,A
MAIN02:
            MOV     A,ADDR_RING
            LCALL   HEX2ASCII
            MOV     DPTR,#ADDR_DATA+25
            MOV     A,CHL
            MOVX    @DPTR,A

            MOV     A,ADDR_RING                    ; PRINT STATUS RING
            LCALL   MEAS_RING
            CJNE    A,#0,MAIN03

            MOV     DPTR,#ADDR_DATA+27
            MOV     A,#'0'
            MOVX    @DPTR,A

            INC     DPTR
            MOV     A,#'F'
            MOVX    @DPTR,A
            INC     DPTR
            MOV     A,#'F'
            MOVX    @DPTR,A

            JMP     EEE
MAIN03:
            MOV     DPTR,#ADDR_DATA+27
            MOV     A,#'0'
            MOVX    @DPTR,A
            INC     DPTR
            MOV     A,#'N'
            MOVX    @DPTR,A
            INC     DPTR
            MOV     A,#' '
            MOVX    @DPTR,A
EEE:
            MOV     DPTR,#ADDR_DATA+17  ;   TOT 1
            JNB     TOT1_F,PT1_0
            MOV     A,#'1'
            MOVX    @DPTR,A
            JMP     PT2_ST
PT1_0:
            MOV     A,#'_'
            MOVX    @DPTR,A
PT2_ST:                                 ;   TOT 2
            INC     DPTR
            JNB     TOT2_F,PT2_0
            MOV     A,#'2'
            MOVX    @DPTR,A
            JMP     PT3_ST
PT2_0:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            MOV     A,#'_'
            MOVX    @DPTR,A
PT3_ST:                                 ;   TOT 3
            INC     DPTR
            JNB     TOT3_F,PT3_0
            MOV     A,#'3'
            MOVX    @DPTR,A
            JMP     PT_END
PT3_0:
            MOV     A,#'_'
            MOVX    @DPTR,A
PT_END:
            LCALL   SAND_RING
            JMP     READ_KEY
;*****************************[ TOT1 TOGGLE ]*********************************

TOT1_TOG:
        JB      TOT1_F,TOT1_RESET
        SETB    TOT1_F
        JMP     TOT1_END
TOT1_RESET:
        CLR     TOT1_F
TOT1_END:
        LCALL   SAND_TOT
        RET

;*****************************[ TOT2 TOGGLE ]*********************************

TOT2_TOG:
        JB      TOT2_F,TOT2_RESET
        SETB    TOT2_F
        JMP     TOT2_END
TOT2_RESET:
        CLR     TOT2_F
TOT2_END:
        LCALL   SAND_TOT
        RET

;*****************************[ TOT3 TOGGLE ]*********************************

TOT3_TOG:
        JB      TOT3_F,TOT3_RESET
        SETB    TOT3_F
        JMP     TOT3_END
TOT3_RESET:
        CLR     TOT3_F
TOT3_END:
        LCALL   SAND_TOT
        RET

;*****************************[ SAND TOT ]************************************
SAND_TOT:
        MOV     A,#0
        JNB     TOT1_F,ST_2
        SETB    A.4
ST_2:
        JNB     TOT2_F,ST_3
        SETB    A.5
ST_3:
        JNB     TOT3_F,ST_END
        SETB    A.6
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ST_END:
        MOV     DPTR,#INTERF_C
        MOVX    @DPTR,A
        RET

;*****************************[ ON RINGING ]*********************************
ON_RING:
        PUSH    A
        PUSH    0

        MOV     A,ADDR_RING
        ANL     A,#0FH
        JB      A.3,OR_BYTE2
        MOV     R0,#RING_D
        JMP     OR_CHK
OR_BYTE2:
        MOV     R0,#RING_D+1
        ANL     A,#7
OR_CHK:
        CJNE    A,#0,OR_NEXT1
        MOV     A,@R0
        ORL     A,#00000001B
        MOV     @R0,A
        JMP     OR_END
OR_NEXT1:
        CJNE    A,#1,OR_NEXT2
        MOV     A,@R0
        ORL     A,#00000010B
        MOV     @R0,A
        JMP     OR_END
OR_NEXT2:
        CJNE    A,#2,OR_NEXT3
        MOV     A,@R0
        ORL     A,#00000100B
        MOV     @R0,A
        JMP     OR_END
OR_NEXT3:
        CJNE    A,#3,OR_NEXT4
        MOV     A,@R0
        ORL     A,#00001000B
        MOV     @R0,A
        JMP     OR_END
OR_NEXT4:
        CJNE    A,#4,OR_NEXT5
        MOV     A,@R0
        ORL     A,#00010000B
        MOV     @R0,A
        JMP     OR_END
OR_NEXT5:
        CJNE    A,#5,OR_NEXT6
        MOV     A,@R0
        ORL     A,#00100000B
        MOV     @R0,A
        JMP     OR_END
OR_NEXT6:
        CJNE    A,#6,OR_NEXT7
        MOV     A,@R0
        ORL     A,#01000000B
        MOV     @R0,A
        JMP     OR_END
OR_NEXT7:
        CJNE    A,#7,OR_END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     A,@R0
        ORL     A,#10000000B
        MOV     @R0,A
OR_END:
        POP     C
        POP     A
        RET

;*****************************[ OFF RINGING ]*********************************
OFF_RING:
        PUSH    A
        PUSH    0

        MOV     A,ADDR_RING
        ANL     A,#0FH
        JB      A.3,OFR_BYTE2
        MOV     R0,#RING_D
        JMP     OFR_CHK
OFR_BYTE2:
        MOV     R0,#RING_D+1
        ANL     A,#7
OFR_CHK:
        CJNE    A,#0,OFR_NEXT1
        MOV     A,@R0
        ANL     A,#11111110B
        MOV     @R0,A
        JMP     OFR_END
OFR_NEXT1:
        CJNE    A,#1,OFR_NEXT2
        MOV     A,@R0
        ANL     A,#11111101B
        MOV     @R0,A
        JMP     OFR_END
OFR_NEXT2:
        CJNE    A,#2,OFR_NEXT3
        MOV     A,@R0
        ANL     A,#11111011B
        MOV     @R0,A
        JMP     OFR_END
OFR_NEXT3:
        CJNE    A,#3,OFR_NEXT4
        MOV     A,@R0
        ANL     A,#11110111B
        MOV     @R0,A
        JMP     OFR_END
OFR_NEXT4:
        CJNE    A,#4,OFR_NEXT5
        MOV     A,@R0
        ANL     A,#11101111B
        MOV     @R0,A
        JMP     OFR_END
OFR_NEXT5:
        CJNE    A,#5,OFR_NEXT6
        MOV     A,@R0
        ANL     A,#11011111B
        MOV     @R0,A
        JMP     OFR_END
OFR_NEXT6:
        CJNE    A,#6,OFR_NEXT7
        MOV     A,@R0
        ANL     A,#10111111B
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     @R0,A
        JMP     OFR_END
OFR_NEXT7:
        CJNE    A,#7,OFR_END
        MOV     A,@R0
        ANL     A,#01111111B
        MOV     @R0,A
OFR_END:
        POP     0
        POP     A
        RET

;*****************************[ MEAS RINGING ]*******************************
MEAS_RING:
        PUSH    0

        MOV     A,ADDR_RING
        ANL     A,#0FH
        JB      A.3,MER_BYTE2
        MOV     R0,#RING_D
        JMP     MER_CHK
MER_BYTE2:
        MOV     R0,#RING_D+1
        ANL     A,#7
MER_CHK:
        CJNE    A,#0,MER_NEXT1
        MOV     A,@R0
        ANL     A,#00000001B
        JMP     MER_END
MER_NEXT1:
        CJNE    A,#1,MER_NEXT2
        MOV     A,@R0
        ANL     A,#00000010B
        JMP     MER_END
MER_NEXT2:
        CJNE    A,#2,MER_NEXT3
        MOV     A,@R0
        ANL     A,#00000100B
        JMP     MER_END
MER_NEXT3:
        CJNE    A,#3,MER_NEXT4
        MOV     A,@R0
        ANL     A,#00001000B
        JMP     MER_END
MER_NEXT4:
        CJNE    A,#4,MER_NEXT5
        MOV     A,@R0
        ANL     A,#00010000B
        JMP     MER_END
MER_NEXT5:
        CJNE    A,#5,MER_NEXT6
        MOV     A,@R0
        ANL     A,#00100000B
        JMP     MER_END
MER_NEXT6:
        CJNE    A,#6,MER_NEXT7
        MOV     A,@R0
        ANL     A,#01000000B
        JMP     MER_END
MER_NEXT7:
        CJNE    A,#7,MER_END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     A,@R0
        ANL     A,#10000000B
MER_END:
        CJNE    A,#0,MER_SET
        JMP     MER_ENDD
MER_SET:
        MOV     A,#1
MER_ENDD:

        POP     0
        RET

;*****************************[ SAND RINGING ]*********************************
SAND_RING:
        MOV     A,RING_D
        MOV     DPTR,#SUB1_B
        MOVX    @DPTR,A
        MOV     A,RING_D+1
        MOV     DPTR,#SUB2_B
        MOVX    @DPTR,A
        RET

;*****************************[ SAND VOICE ]***********************************
SAND_VOICE:
          PUSH      DPH
          PUSH      DPL
          MOV       DPTR,#FFE0H
          MOVX      @DPTR,A
          POP       DPL
          POP       DPH
          RET

;*****************************[ ON SWITCH ]************************************

ON_SW:
            CLR     IE.7
            PUSH    A
            PUSH    DPH
            PUSH    DPL

            MOV     DPTR,#SW3_A
            MOV     A,ADDR_SW
            MOVX    @DPTR,A

            MOV     DPTR,#SW3_B
            MOV     A,#1H
            MOVX    @DPTR,A

            MOV     DPTR,#SW3_B
            MOV     A,#3H
            MOVX    @DPTR,A

            MOV     DPTR,#SW3_B
            MOV     A,#1H
            MOVX    @DPTR,A

            MOV     DPTR,#SW3_B
            MOV     A,#0H
            MOVX    @DPTR,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        POP     DPL
        POP     DPH
        POP     A
        SETB    IE.7
        RET

;*****************************[ OFF SWITCH ]*********************************

OFF_SW:
        CLR     IE.7
        PUSH    A
        PUSH    DPH
        PUSH    DPL

        MOV     DPTR,#SW3_A
        MOV     A,ADDR_SW
        MOVX    @DPTR,A

        MOV     DPTR,#SW3_B
        MOV     A,#0H
        MOVX    @DPTR,A

        MOV     DPTR,#SW3_B
        MOV     A,#2H
        MOVX    @DPTR,A

        MOV     DPTR,#SW3_B
        MOV     A,#0H
        MOVX    @DPTR,A

        POP     DPL
        POP     DPH
        POP     A
        SETB    IE.7
        RET

;*****************************[ MEAS MAP SW ]********************************

MEAS_MAP_SW:
        PUSH    1

        MOV     A,ADDR_SW

        RR      A
        RR      A                       ; ADDR = R1:A
        RR      A
        ANL     A,#00011111B
        ADD     A,#MAP_SW
        MOV     R1,A
        MOV     A,ADDR_SW
        ANL     A,#00000111B    ; R0 IT DATA IN ADDR
                                ; R1 IT ADDR TO BYTE FOR MAP SW
        CJNE    A,#0,MMS10
        MOV     A,@R1
        JMP     MMS00
MMS10:  CJNE    A,#1,MMS20
        MOV     A,@R1
        JMP     MMS01
MMS20:  CJNE    A,#2,MMS30
        MOV     A,@R1
        JMP     MMS02
MMS30:  CJNE    A,#3,MMS40
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     A,@R1
                JMP     MMS03
MMS40:          CJNE    A,#4,MMS50
                MOV     A,@R1
                JMP     MMS04
MMS50:          CJNE    A,#5,MMS60
                MOV     A,@R1
                JMP     MMS05
MMS60:          CJNE    A,#6,MMS70
                MOV     A,@R1
                JMP     MMS06
MMS70:          MOV     A,@R1
MMS07:          RR      A
MMS06:          RR      A
MMS05:          RR      A
MMS04:          RR      A
MMS03:          RR      A
MMS02:          RR      A
MMS01:          RR      A
MMS00:
                ANL     A,#1

                POP     1
                RET

;*****************************[ CLEAR MAP SW ]*********************************

CLR_MAP_SW:
                PUSH    B
                PUSH    1
                PUSH    A               ; A IT ADDR TO BIT FOR MAP SW
                PUSH    DPH
                PUSH    DPL

                MOV     A,ADDR_SW

                RR      A
                RR      A               ; ADDR = R1:A
                RR      A
                ANL     A,#00011111B
                ADD     A,#MAP_SW
                MOV     R1,A
                MOV     A,ADDR_SW
                ANL     A,#00000111B    ; R0 IT DATA IN ADDR
                                        ; R1 IT ADDR TO BYTE FOR MAP SW
                MOV     B,@R1

                CJNE    A,#0,CMS00
                CLR     B.0
                JMP     CMSEND
CMS00:
                CJNE    A,#1,CMS01
                CLR     B.1
                JMP     CMSEND
CMS01:
                CJNE    A,#2,CMS02
                CLR     B.2
                JMP     CMSEND
CMS02:
                CJNE    A,#3,CMS03
                CLR     B.3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JMP     CMSEND
CMS03:
        CJNE    A,#4,CMS04
        CLR     B.4
        JMP     CMSEND
CMS04:
        CJNE    A,#5,CMS05
        CLR     B.5
        JMP     CMSEND
CMS05:
        CJNE    A,#6,CMS06
        CLR     B.6
        JMP     CMSEND
CMS06:
        CJNE    A,#7,CMSEND
        CLR     B.7
CMSEND:
        MOV     @R1,B

        LCALL   OFF_SW

        POP     DPL
        POP     DPH
        POP     A
        POP     1
        POP     B
        RET

;*****************************[ SET MAP SW ]*********************************

SET_MAP_SW:
        PUSH    B
        PUSH    1
        PUSH    A               ; A IT ADDR TO BIT FOR MAP SW
        PUSH    DPH
        PUSH    DPL

        MOV     A,ADDR_SW

        RR      A
        RR      A               ; ADDR = R1:A
        RR      A
        ANL     A,#00011111B
        ADD     A,#MAP_SW
        MOV     R1,A
        MOV     A,ADDR_SW
        ANL     A,#00000111B    ; R0 IT DATA IN ADDR
                                ; R1 IT ADDR TO BYTE FOR MAP SW

        MOV     B,@R1

        CJNE    A,#0,SMS00
        SETB    B.0
        JMP     SMSEND
SMS00:
        CJNE    A,#1,SMS01
        SETB    B.1
        JMP     SMSEND
SMS01:
        CJNE    A,#2,SMS02
        SETB    B.2
        JMP     SMSEND
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SMS02:
            CJNE    A,#3,SMS03
            SETB    B.3
            JMP     SMSEND
SMS03:
            CJNE    A,#4,SMS04
            SETB    B.4
            JMP     SMSEND
SMS04:
            CJNE    A,#5,SMS05
            SETB    B.5
            JMP     SMSEND
SMS05:
            CJNE    A,#6,SMS06
            SETB    B.6
            JMP     SMSEND
SMS06:
            CJNE    A,#7,SMSEND
            SETB    B.7
SMSEND:
            MOV     @R1,B

            LCALL   ON_SW

            POP     DPL
            POP     DPH
            POP     A
            POP     1
            POP     B
            RET

;*****************************[ CLEAR ALL MAP SW ]*****************************

CLR_ALL_MAP_SW:
            PUSH    1
            MOV     R1,#00
CAMS00:     MOV     ADDR_SW,R1
            LCALL   CLR_MAP_SW
            DJNZ    R1,CAMS00
            POP     1
            RET

;*********************[ CONVERT HEX IN BYTE TO ASCII ]**********************
HEX2ASCII:
            MOV     GEN0,A
            ANL     A,#0FH
            LCALL   BIT2CHAR
            MOV     CHL,A

            MOV     A,GEN0
            LCALL   XCHA
            ANL     A,#0FH
            LCALL   BIT2CHAR
            MOV     CHH,A

            RET
BIT2CHAR:
            CJNE    A,#0AH,NEXT00
            MOV     A,#41H
            JMP     H2A
NEXT00:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CJNE    A,#0BH,NEXT01
            MOV     A,#42H
            JMP     H2A
NEXT01:
            CJNE    A,#0CH,NEXT02
            MOV     A,#43H
            JMP     H2A
NEXT02:
            CJNE    A,#0DH,NEXT03
            MOV     A,#44H
            JMP     H2A
NEXT03:
            CJNE    A,#0EH,NEXT04
            MOV     A,#45H
            JMP     H2A
NEXT04:
            CJNE    A,#0FH,NEXT05
            MOV     A,#46H
            JMP     H2A
NEXT05:
            ADD     A,#48

H2A:        RET

;*******************[ INITIAL DATA DISPLAY TO RAM ]*************************
INIT_DATA_DIS:

            MOV     DPTR,#DATA_DIS
            MOV     DPL0,DPL
            MOV     DPH0,DPH

            MOV     DPTR,#ADDR_DATA
            MOV     DPL1,DPL
            MOV     DPH1,DPH

            MOV     R0,#40
IDD00:
            MOV     DPL,DPL0
            MOV     DPH,DPH0
            MOV     A,#0
            MOVC    A,@A+DPTR
            INC     DPTR
            MOV     DPL0,DPL
            MOV     DPH0,DPH

            MOV     DPL,DPL1
            MOV     DPH,DPH1
            MOVX    @DPTR,A
            INC     DPTR
            MOV     DPL1,DPL
            MOV     DPH1,DPH

            DJNZ    R0,IDD00
            RET
;************************[ PRINT DISPLAY ]*********************************

PRINTF:
            CLR     IE.7
            LCALL   CLRSCR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     R0,#20
        MOV     DPTR,#ADDR_DATA

        MOV     A,#00
        LCALL   LCD_COM
STR1:
        MOV     A,#0
        MOVX    A,@DPTR         ; XCH MOVX OR MOVC
        LCALL   WRITE
        INC     DPTR
        DJNZ    R0,STR1

        MOV     A,#0C0H         ;line 2
        LCALL   LCD_COM

        MOV     R0,#20
STR3:
        MOV     A,#0
        MOVX    A,@DPTR         ; XCH MOVX OR MOVC
        LCALL   WRITE
        INC     DPTR
        DJNZ    R0,STR3
        SETB    IE.7

        RET

;************************[ INITIAL INT ]************************************>

INIT_INT:

        MOV     IE,#10100101B
        MOV     IP,#00100000B
;        MOV     TMOD,#00100000B
        MOV     TCON,#00000001B

;        MOV     SCON,#01010100B
;        MOV     TH1,#FFH              ;BOARD RATE
        MOV     TIME,#0
        MOV     INT0,#0
        MOV     INT1,#0
        MOV     DISPLAY,#0
        DB      75H,C8H,04H     ; MOV     T2CON,#00000100B
        DB      75H,CAH,FFH     ; MOV     RCAP2L,#023H
        DB      75H,CBH,3FH     ; MOV     RCAP2H,#0CFH
        MOV     TOT1_F,#0
        MOV     TOT2_F,#0
        MOV     TOT3_F,#0
        MOV     RING_D,#0
        MOV     RING_D+1,#0
        MOV     ADDR_RING,#0


        RET

;************************[ CLEAR SCREEN DISPLAY ]****************************>

CLRSCR:
        PUSH    DPL
        PUSH    DPH
        PUSH    A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     A,#01           ;  CLS DISPLAY
        MOV     DPTR,#COMMAND   ;
        MOVX    @DPTR,A         ;
        LCALL   WAITBF          ;

        POP     A
        POP     DPH
        POP     DPL

        RET

;************************[ Write ASCII to LCD ]*******************************

WRITE:  PUSH    DPL
        PUSH    DPH

        MOV     DPTR,#WRITEDATA
        MOVX    @DPTR,A
        LCALL   WAITBF      ;Wait LCD module ready

        POP     DPH
        POP     DPL

        RET

;************************[ WRITE COMMAND ]************************************

LCD_COM:
        PUSH    DPL
        PUSH    DPH


        MOV     DPTR,#COMMAND
        MOVX    @DPTR,A
        LCALL   WAITBF          ;Wait busy flag

        POP     DPH
        POP     DPL

        RET

;*****************************[ Wait for ready ]*****************************

WAITBF: PUSH    DPL
        PUSH    DPH
        MOV     DPTR,#READBUSY
RDY1:   MOVX    A,@DPTR
        JB      ACC.7,RDY1      ;Busy Flag
        POP     DPH
        POP     DPL
        RET

;************************[ INITIAL DISPLAY ]*********************************

INIT:   PUSH    DPL
        PUSH    DPH
        MOV     DPTR,#COMMAND

        MOV     A,#38H          ;8bit, 2 line, 5x7 dot
        MOVX    @DPTR,A
        LCALL   WAITBF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     A,#0FH
        MOVX    @DPTR,A
        LCALL   WAITBF

        MOV     A,#6            ;increment cursor
        MOVX    @DPTR,A
        LCALL   WAITBF

        MOV     A,#1            ;clear and home
        MOVX    @DPTR,A
        LCALL   WAITBF

        POP     DPH
        POP     DPL
        RET

;************************************[ DELAY ]*****************************

DELAY:
        PUSH    B                       ; save bystander

        MOV     B,#0                    ; 256 loops
        DJNZ    B,$                     ; 2.17 us/loop

        MOV     B,#0                    ; 256 loops
        DJNZ    B,$                     ; 2.17 us/loop

        POP     B                       ; restore bystander

        DJNZ    ACC,DELAY               ; repeat ACC times

        RET

;***************[ REGISTOR A IS EXCHANGE BETWEEN AL,AH ]********************
XCHA:
        RR      A
        RR      A
        RR      A
        RR      A
        RET
;*************************[ INITIAL PORT ]*********************************

INIT_PORT:
        PUSH    DPL
        PUSH    DPH
        PUSH    A

        MOV     DPTR,#INTERF_COM
        MOV     A,#91H
        MOVX    @DPTR,A

        MOV     DPTR,#SUB1_COM
        MOV     A,#90H
        MOVX    @DPTR,A

        MOV     DPTR,#SUB2_COM
        MOV     A,#90H
        MOVX    @DPTR,A

        MOV     DPTR,#TOT1_COM
        MOV     A,#9BH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOVX    @DPTR,A

        MOV     DPTR,#TOT2_COM
        MOV     A,#98H
        MOVX    @DPTR,A

        MOV     DPTR,#SW1_COM
        MOV     A,#98H
        MOVX    @DPTR,A

        MOV     DPTR,#SW2_COM
        MOV     A,#98H
        MOVX    @DPTR,A

        MOV     DPTR,#SW3_COM
        MOV     A,#89H
        MOVX    @DPTR,A

        MOV     DPTR,#INTERF_B
        MOV     A,#FFH
        MOVX    @DPTR,A

        POP     A
        POP     DPH
        POP     DPL
        RET

;*************************[ DATA ]*********************************

DATA_DIS:  DB   'CLK:00 KEY:0 TOT:123'
           DB   'RING:0:OFF SW:00:OFF'

           END
;*******************************[ END OF PROGRAM ]*************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ง. DATA ของอุปกรณ์ที่สำคัญในโครงงานปริญญานิพนธ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

intel

ARCHITECTURAL OVERVIEW OF THE MCS®-51 FAMILY OF MICROCONTROLLERS

INTRODUCTION

The 8051 is the original member of the MCS®-51 family, and is the core for all MCS-51 devices. The features of the 8051 core are:

- 8-bit CPU optimized for control applications
- Extensive Boolean processing (single-bit logic) capabilities
- 64K Program Memory address space
- 64K Data Memory address space
- 4K bytes of on-chip Program Memory
- 128 bytes of on-chip Data RAM
- 32 bidirectional and individually addressable I/O lines
- Two 16-bit timer/counters
- Full duplex UART
- 6-source/5-vector interrupt structure with two priority levels
- On-chip clock oscillator

The basic architectural structure of this 8051 core is shown in Figure 1.

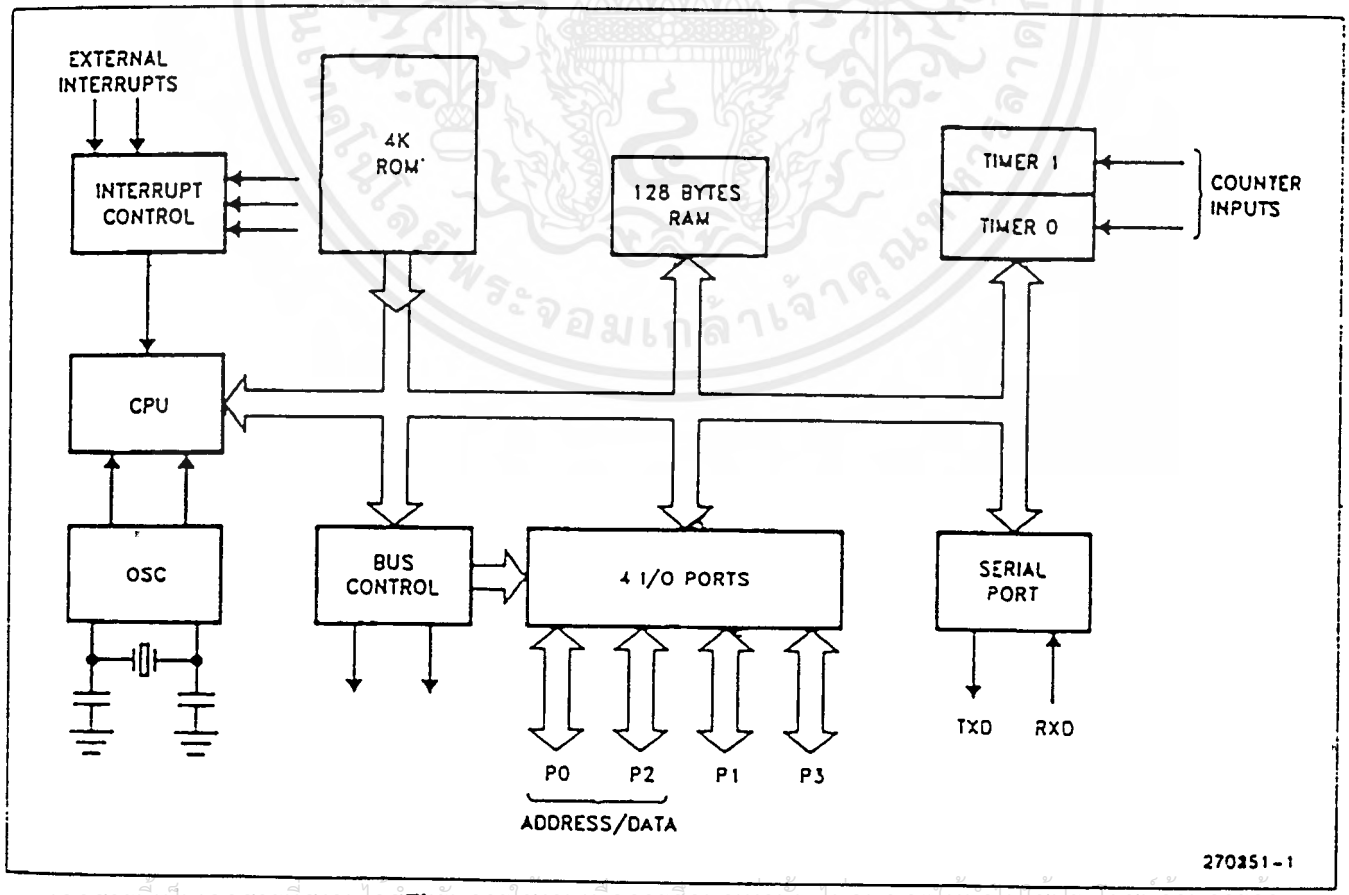


Figure 1. Block Diagram of the 8051 Core

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Each device on the MCS-51 family consists of all the core features plus some additional features. A feature comparison of all the MCS-51 devices is shown in Table 1.

Table 1. The MCS®-51 Family of Microcontrollers

Device	ROMless Version	EPROM Version	ROM Bytes	RAM Bytes	8-Bit I/O Ports	16-Bit Timer/ Counters	Programmable Counter Array (PCA)	UART	Serial Expansion Port (SEP)	Global Serial Channel (GSC)	DMA Channels	A/D Channels	Interrupt Sources/ Vectors	Power Down and Idle Modes
8051	8031	—	4K	128	4	2		✓					6/5	
8051AH	8031AH	8751H 8751BH	4K	128	4	2		✓					6/5	
8052AH	8032AH	8752BH	8K	256	4	3		✓					8/6	
80C51BH	80C31BH	87C51	4K	128	4	2		✓					6/5	✓
83C51FA	80C51FA	87C51FA	8K	256	4	3	✓	✓					14/7	✓
83C51FB	80C51FA	87C51FB	16K	256	4	3	✓	✓					14/7	✓
83C51GA	80C51GA	87C51GA	4K	128	4	2		✓	✓			8	8/7	✓
83C152JA	80C152JA	—	8K	256	5	2		✓		✓	2		10/11	✓
—	80C152JB	—	—	256	7	2		✓		✓	2		10/11	✓
83C152JC	80C152JC	—	8K	256	5	2		✓		✓	2		10/11	✓
—	80C152JD	—	—	256	7	2		✓		✓	2		10/11	✓
83C451	80C451	—	4K	128	7	2		✓					6/5	✓
83C452	80C452	87C452P	8K	256	5	2		✓					9/8	✓

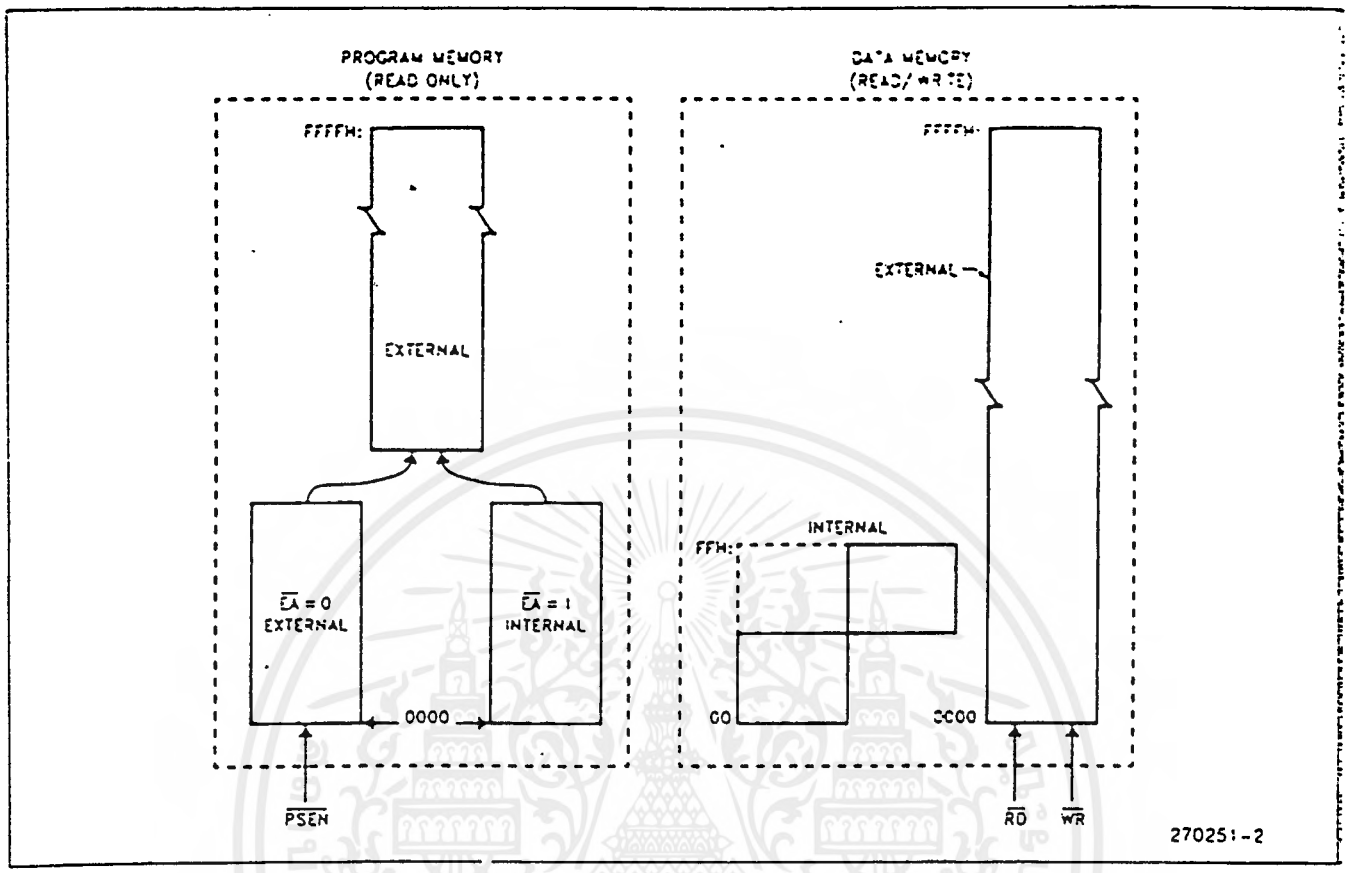


Figure 2. MCS®-51 Memory Structure

CHMOS Devices

Functionally, the CHMOS devices (designated with "C" in the middle of the device name) are all fully compatible with the 8051, but being CMOS, draw less current than an HMOS counterpart. To further exploit the power savings available in CMOS circuitry, two reduced power modes are added:

- Software-invoked Idle Mode, during which the CPU is turned off while the RAM and other on-chip peripherals continue operating. In this mode, current draw is reduced to about 15% of the current drawn when the device is fully active.
- Software-invoked Power Down Mode, during which all on-chip activities are suspended. The on-chip RAM continues to hold its data. In this mode the device typically draws less than 10 μA.

Although the 80C51BH is functionally compatible with its HMOS counterpart, specific differences between the two types of devices must be considered in the design of an application circuit if one wishes to ensure complete interchangeability between the HMOS and CHMOS devices. These considerations are discussed in the Application Note AP-252, "Designing with the 80C51BH".

For more information on the individual devices and features listed in Table 1, refer to the Hardware Descriptions and Data Sheets of the specific device.

MEMORY ORGANIZATION IN MCS®-51 DEVICES

Logical Separation of Program and Data Memory

All MCS-51 devices have separate address spaces for Program and Data Memory, as shown in Figure 2. The logical separation of Program and Data Memory allows the Data Memory to be accessed by 8-bit addresses, which can be more quickly stored and manipulated by an 8-bit CPU. Nevertheless, 16-bit Data Memory addresses can also be generated through the DPTR register.

Program Memory can only be read, not written to. There can be up to 64K bytes of Program Memory. In the ROM and EPROM versions of these devices the lowest 4K, 8K or 16K bytes of Program Memory are provided on-chip. Refer to Table 1 for the amount of on-chip ROM (or EPROM) on each device. In the ROMless versions all Program Memory is external. The read strobe for external Program Memory is the signal $\overline{PSEN}$ (Program Store Enable).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลง เผยแพร่ หรือนำไปใช้ในเชิงพาณิชย์ ทั้งนี้ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Data Memory occupies a separate address space from Program Memory. Up to 64K bytes of external RAM can be addressed in the external Data Memory space. The CPU generates read and write signals, $\overline{RD}$ and $\overline{WR}$, as needed during external Data Memory accesses.

External Program Memory and external Data Memory may be combined if desired by applying the $\overline{RD}$ and $\overline{PSEN}$ signals to the inputs of an AND gate and using the output of the gate as the read strobe to the external Program/Data memory.

Program Memory

Figure 3 shows a map of the lower part of the Program Memory. After reset, the CPU begins execution from location 0000H.

As shown in Figure 3, each interrupt is assigned a fixed location in Program Memory. The interrupt causes the CPU to jump to that location, where it commences execution of the service routine. External Interrupt 0, for example, is assigned to location 0003H. If External Interrupt 0 is going to be used, its service routine must begin at location 0003H. If the interrupt is not going to be used, its service location is available as general purpose Program Memory.

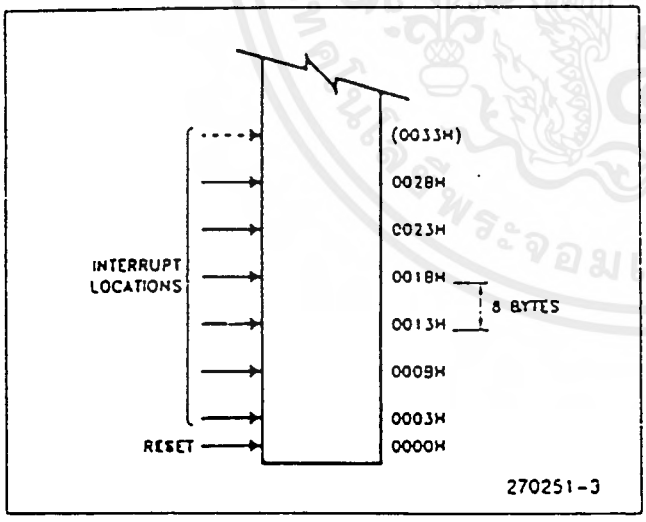


Figure 3. MCS®-51 Program Memory

The interrupt service locations are spaced at 8-byte intervals: 0003H for External Interrupt 0, 000BH for Timer 0, 0013H for External Interrupt 1, 001BH for Timer 1, etc. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within that 8-byte interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.

The lowest 4K (or 8K or 16K) bytes of Program Memory can be either in the on-chip ROM or in an external ROM. This selection is made by strapping the $\overline{EA}$ (External Access) pin to either V_{CC} or V_{SS}.

In the 4K byte ROM devices, if the $\overline{EA}$ pin is strapped to V_{CC}, then program fetches to addresses 0000H through 0FFFH are directed to the internal ROM. Program fetches to addresses 1000H through FFFFH are directed to external ROM.

In the 8K byte ROM devices, $\overline{EA}$ = V_{CC} selects addresses 0000H through 1FFFH to be internal, and addresses 2000H through FFFFH to be external.

In the 16K byte ROM devices, $\overline{EA}$ = V_{CC} selects addresses 0000H through 3FFFH to be internal, and addresses 4000H through FFFFH to be external.

If the $\overline{EA}$ pin is strapped to V_{SS}, then all program fetches are directed to external ROM. The ROMless parts must have this pin externally strapped to V_{SS} to enable them to execute properly.

The read strobe to external ROM, $\overline{PSEN}$, is used for all external program fetches. $\overline{PSEN}$ is not activated for internal program fetches.

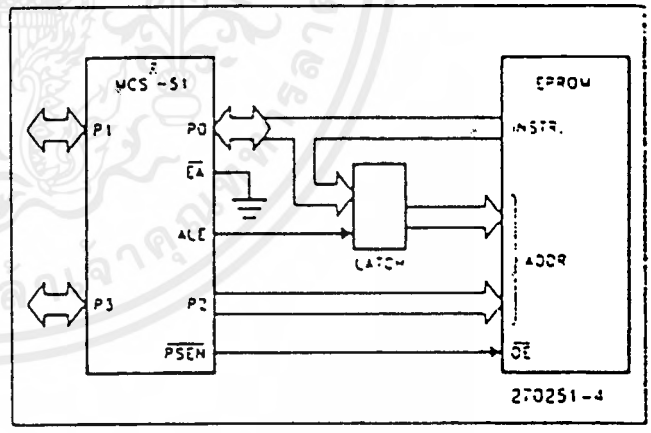


Figure 4. Executing from External Program Memory

The hardware configuration for external program execution is shown in Figure 4. Note that 16 I/O lines (Ports 0 and 2) are dedicated to bus functions during external Program Memory fetches. Port 0 (P0 in Figure 4) serves as a multiplexed address/data bus. It emits the low byte of the Program Counter (PCL) as an address, and then goes into a float state awaiting the arrival of the code byte from the Program Memory. During the time that the low byte of the Program Counter is valid on P0, the signal ALE (Address Latch Enable) clocks this byte into an address latch. Meanwhile, Port 2 (P2 in Figure 4) emits the high byte of the Program Counter (PCH). Then $\overline{PSEN}$ strobes the EPROM and the code byte is read into the microcontroller.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษา ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้

Program Memory addresses are always 16 bits wide, even though the actual amount of Program Memory used may be less than 64K bytes. External program execution sacrifices two of the 8-bit ports, P0 and P2, to the function of addressing the Program Memory.

Data Memory

The right half of Figure 2 shows the internal and external Data Memory spaces available to the MCS-51 user.

Figure 5 shows a hardware configuration for accessing up to 2K bytes of external RAM. The CPU in this case is executing from internal ROM. Port 0 serves as a multiplexed address/data bus to the RAM, and 3 lines of Port 2 are being used to page the RAM. The CPU generates $\overline{RD}$ and $\overline{WR}$ signals as needed during external RAM accesses.

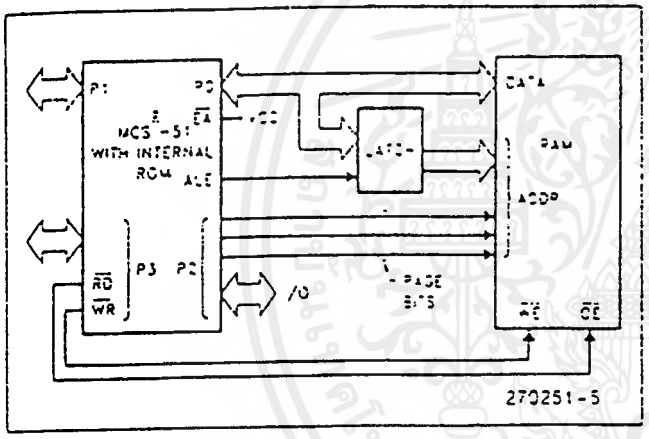


Figure 5. Accessing External Data Memory. If the Program Memory is Internal, the Other Bits of P2 are Available as I/O.

There can be up to 64K bytes of external Data Memory. External Data Memory addresses can be either 1 or 2 bytes wide. One-byte addresses are often used in conjunction with one or more other I/O lines to page the RAM, as shown in Figure 5. Two-byte addresses can also be used, in which case the high address byte is emitted at Port 2.

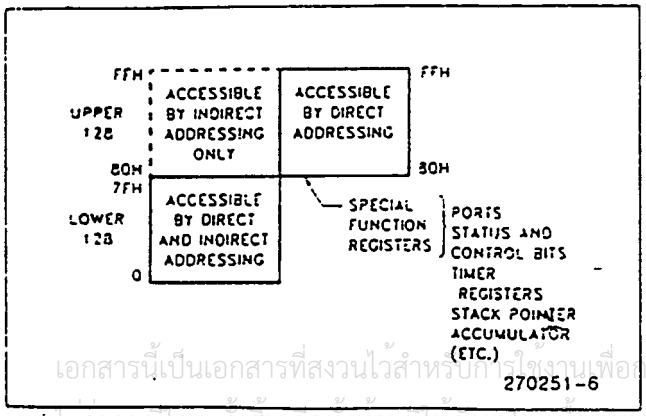


Figure 6. Internal Data Memory

Internal Data Memory is mapped in Figure 6. The memory space is shown divided into three blocks, which are generally referred to as the Lower 128, the Upper 128, and SFR space.

Internal Data Memory addresses are always one byte wide, which implies an address space of only 256 bytes. However, the addressing modes for internal RAM can in fact accommodate 384 bytes, using a simple trick. Direct addresses higher than 7FH access one memory space, and indirect addresses higher than 7FH access a different memory space. Thus Figure 6 shows the Upper 128 and SFR space occupying the same block of addresses, 80H through FFH, although they are physically separate entities.

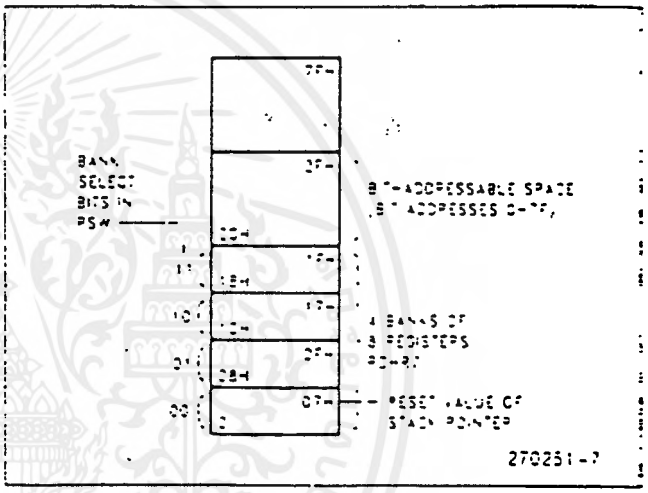


Figure 7. The Lower 128 Bytes of Internal RAM

The Lower 128 bytes of RAM are present in all MCS-51 devices as mapped in Figure 7. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word (PSW) select which register bank is in use. This allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing.

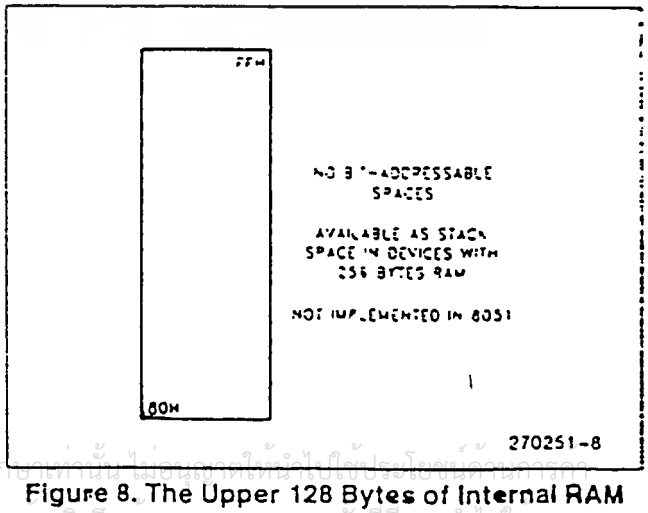


Figure 8. The Upper 128 Bytes of Internal RAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

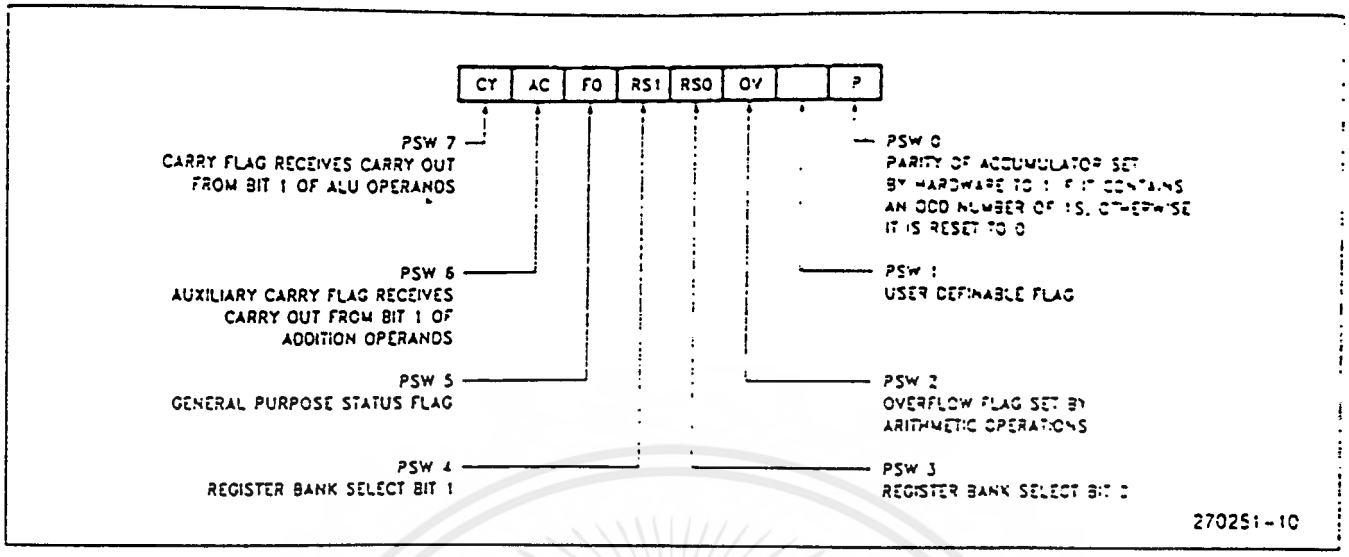


Figure 10. PSW (Program Status Word) Register in MCS®-51 Devices

The next 16 bytes above the register banks form a block of bit-addressable memory space. The MCS-51 instruction set includes a wide selection of single-bit instructions, and the 128 bits in this area can be directly addressed by these instructions. The bit addresses in this area are 00H through 7FH.

All of the bytes in the Lower 128 can be accessed by either direct or indirect addressing. The Upper 128 (Figure 8) can only be accessed by indirect addressing. The Upper 128 bytes of RAM are not implemented in the 8051, but are in the devices with 256 bytes of RAM. (See Table 1).

Figure 9 gives a brief look at the Special Function Register (SFR) space. SFRs include the Port latches, timers, peripheral controls, etc. These registers can only be accessed by direct addressing. In general, all MCS-51 microcontrollers have the same SFRs as the 8051, and at the same addresses in SFR space. However, enhancements to the 8051 have additional SFRs that are not present in the 8051, nor perhaps in other proliferations of the family.

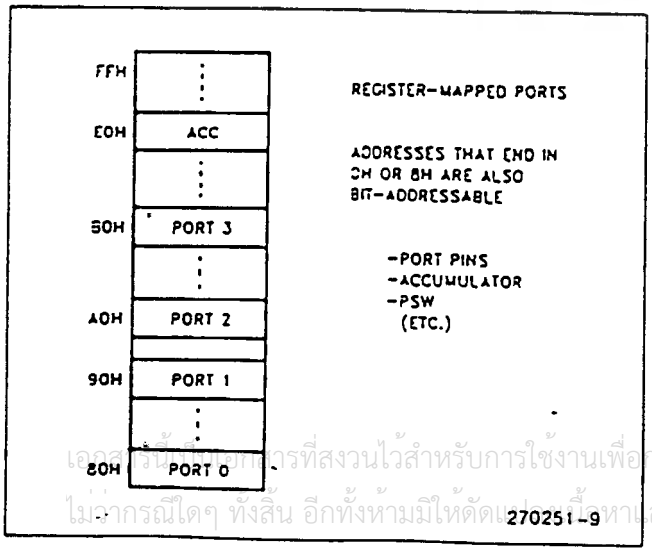


Figure 9. SFR Space

Sixteen addresses in SFR space are both byte- and bit-addressable. The bit-addressable SFRs are those whose address ends in 000B. The bit addresses in this area are 80H through FFH.

THE MCS®-51 INSTRUCTION SET

All members of the MCS-51 family execute the same instruction set. The MCS-51 instruction set is optimized for 8-bit control applications. It provides a variety of fast addressing modes for accessing the internal RAM to facilitate byte operations on small data structures. The instruction set provides extensive support for one-bit variables as a separate data type, allowing direct bit manipulation in control and logic systems that require Boolean processing.

An overview of the MCS-51 instruction set is presented below, with a brief description of how certain instructions might be used. References to "the assembler" in this discussion are to Intel's MCS-51 Macro Assembler, ASM51. More detailed information on the instruction set can be found in the MCS-51 Macro Assembler User's Guide (Order No. 9800937 for ISIS Systems, Order No. 122752 for DOS Systems).

Program Status Word

The Program Status Word (PSW) contains several status bits that reflect the current state of the CPU. The PSW, shown in Figure 10, resides in SFR space. It contains the Carry bit, the Auxiliary Carry (for BCD operations), the two register bank select bits, the Overflow flag, a Parity bit, and two user-definable status flags.

The Carry bit, other than serving the functions of a Carry bit in arithmetic operations, also serves as the "Accumulator" for a number of Boolean operations.

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The bits RS0 and RS1 are used to select one of the four register banks shown in Figure 7. A number of instructions refer to these RAM locations as R0 through R7. The selection of which of the four banks is being referred to is made on the basis of the bits RS0 and RS1 at execution time.

The Parity bit reflects the number of 1s in the Accumulator: P = 1 if the Accumulator contains an odd number of 1s, and P = 0 if the Accumulator contains an even number of 1s. Thus the number of 1s in the Accumulator plus P is always even.

Two bits in the PSW are uncommitted and may be used as general purpose status flags.

Addressing Modes

The addressing modes in the MCS-51 instruction set are as follows:

DIRECT ADDRESSING

In direct addressing the operand is specified by an 8-bit address field in the instruction. Only internal Data RAM and SFRs can be directly addressed.

INDIRECT ADDRESSING

In indirect addressing the instruction specifies a register which contains the address of the operand. Both internal and external RAM can be indirectly addressed.

The address register for 8-bit addresses can be R0 or R1 of the selected register bank, or the Stack Pointer. The address register for 16-bit addresses can only be the 16-bit "data pointer" register, DPTR.

REGISTER INSTRUCTIONS

The register banks, containing registers R0 through R7, can be accessed by certain instructions which carry a 3-bit register specification within the opcode of the instruction. Instructions that access the registers this way are code efficient, since this mode eliminates an address byte. When the instruction is executed, one of the eight registers in the selected bank is accessed. One of four banks is selected at execution time by the two bank select bits in the PSW.

REGISTER-SPECIFIC INSTRUCTIONS

Some instructions are specific to a certain register. For example, some instructions always operate on the Accumulator, or Data Pointer, etc., so no address byte is needed to point to it. The opcode itself does that. Instructions that refer to the Accumlator as A assemble as accumulator-specific opcodes.

IMMEDIATE CONSTANTS

The value of a constant can follow the opcode in Program Memory. For example,

```
MOV A, #100
```

loads the Accumulator with the decimal number 100. The same number could be specified in hex digits as 64H.

INDEXED ADDRESSING

Only Program Memory can be accessed with indexed addressing, and it can only be read. This addressing mode is intended for reading look-up tables in Program Memory. A 16-bit base register (either DPTR or the Program Counter) points to the base of the table, and the Accumulator is set up with the table entry number. The address of the table entry in Program Memory is formed by adding the Accumulator data to the base pointer.

Another type of indexed addressing is used in the "case jump" instruction. In this case the destination address of a jump instruction is computed as the sum of the base pointer and the Accumulator data.

Arithmetic Instructions

The menu of arithmetic instructions is listed in Table 2. The table indicates the addressing modes that can be used with each instruction to access the <byte> operand. For example, the ADD A, <byte> instruction can be written as:

```
ADD     A,7FH       (direct addressing)
ADD     A,@R0       (indirect addressing)
ADD     A,R7        (register addressing)
ADD     A,#127      (immediate constant)
```

The execution times listed in Table 2 assume a 12 MHz clock frequency. All of the arithmetic instructions execute in 1 μs except the INC DPTR instruction, which takes 2 μs, and the Multiply and Divide instructions, which take 4 μs.

Note that any byte in the internal Data Memory space can be incremented or decremented without going through the Accumulator.

One of the INC instructions operates on the 16-bit Data Pointer. The Data Pointer is used to generate 16-bit addresses for external memory, so being able to increment it in one 16-bit operation is a useful feature.

The MUL AB instruction multiplies the Accumulator by the data in the B register and puts the 16-bit product into the concatenated B and Accumulator registers.

Table 2. A List of the MCS®-51 Arithmetic Instructions

Mnemonic	Operation	Addressing Modes				Execution Time (μs)
		Dir	Ind	Reg	Imm	
ADD A,<byte>	A = A + <byte>	X	X	X	X	1
ADDC A,<byte>	A = A + <byte> + C	X	X	X	X	1
SUBB A,<byte>	A = A − <byte> − C	X	X	X	X	1
INC A	A = A + 1	Accumulator only				1
INC <byte>	<byte> = <byte> + 1	X	X	X		1
INC DPTR	DPTR = DPTR + 1	Data Pointer only				2
DEC A	A = A − 1	Accumulator only				1
DEC <byte>	<byte> = <byte> − 1	X	X	X		1
MUL AB	B.A = B x A	ACC and B only				4
DIV AB	A = Int [A/B] B = Mod [A/B]	ACC and B only				4
DA A	Decimal Adjust	Accumulator only				1

The DIV AB instruction divides the Accumulator by the data in the B register and leaves the 8-bit quotient in the Accumulator, and the 8-bit remainder in the B register.

Oddly enough, DIV AB finds less use in arithmetic "divide" routines than in radix conversions and programmable shift operations. An example of the use of DIV AB in a radix conversion will be given later. In shift operations, dividing a number by 2^n shifts its n bits to the right. Using DIV AB to perform the division completes the shift in 4 μs and leaves the B register holding the bits that were shifted out

The DA A instruction is for BCD arithmetic operations. In BCD arithmetic, ADD and ADDC instructions should always be followed by a DA A operation, to ensure that the result is also in BCD. Note that DA A will not convert a binary number to BCD. The DA A operation produces a meaningful result only as the second step in the addition of two BCD bytes.

Table 3. A List of the MCS®-51 Logical Instructions

Mnemonic	Operation	Addressing Modes				Execution Time (μs)
		Dir	Ind	Reg	Imm	
ANL A,<byte>	A = A .AND. <byte>	X	X	X	X	1
ANL <byte>,A	<byte> = <byte> .AND. A	X				1
ANL <byte>,#data	<byte> = <byte> .AND. #data	X				2
ORL A,<byte>	A = A .OR. <byte>	X	X	X	X	1
ORL <byte>,A	<byte> = <byte> .OR. A	X				1
ORL <byte>,#data	<byte> = <byte> .OR. #data	X				2
XRL A,<byte>	A = A .XOR. <byte>	X	X	X	X	1
XRL <byte>,A	<byte> = <byte> .XOR. A	X				1
XRL <byte>,#data	<byte> = <byte> .XOR. #data	X				2
CRL A	A = 00H	Accumulator only				1
CPL A	A = .NOT. A	Accumulator only				1
RL A	Rotate ACC Left 1 bit	Accumulator only				1
RLC A	Rotate Left through Carry	Accumulator only				1
RR A	Rotate ACC Right 1 bit	Accumulator only				1
RRC A	Rotate Right through Carry	Accumulator only				1
SWAP A	Swap Nibbles in A	Accumulator only				1

Logical Instructions

Table 3 shows the list of MCS-51 logical instructions. The instructions that perform Boolean operations (AND, OR, Exclusive OR, NOT) on bytes perform the operation on a bit-by-bit basis. That is, if the Accumulator contains 00110101B and <byte> contains 01010011B, then

```
ANL     A,<byte>
```

will leave the Accumulator holding 00010001B.

The addressing modes that can be used to access the <byte> operand are listed in Table 3. Thus, the ANL A,<byte> instruction may take any of the forms

```
ANL     A,7FH       (direct addressing)
ANL     A,@R1       (indirect addressing)
ANL     A,R6        (register addressing)
ANL     A,#53H      (immediate constant)
```

All of the logical instructions that are Accumulator-specific execute in 1μs (using a 12 MHz clock). The others take 2 μs.

Note that Boolean operations can be performed on any byte in the lower 128 internal Data Memory space or the SFR space using direct addressing, without having to use the Accumulator. The XRL <byte>, #data instruction, for example, offers a quick and easy way to invert port bits, as in

```
XRL     P1,#0FFH
```

If the operation is in response to an interrupt, not using the Accumulator saves the time and effort to stack it in the service routine.

The Rotate instructions (RL A, RLC A, etc.) shift the Accumulator 1 bit to the left or right. For a left rotation, the MSB rolls into the LSB position. For a right rotation, the LSB rolls into the MSB position.

The SWAP A instruction interchanges the high and low nibbles within the Accumulator. This is a useful operation in BCD manipulations. For example, if the Accumulator contains a binary number which is known to be less than 100, it can be quickly converted to BCD by the following code:

```
MOV     B,#10
DIV     AB
SWAP    A
ADD     A,B
```

Dividing the number by 10 leaves the tens digit in the low nibble of the Accumulator, and the ones digit in the B register. The SWAP and ADD instructions move the tens digit to the high nibble of the Accumulator, and the ones digit to the low nibble.

Data Transfers

INTERNAL RAM

Table 4 shows the menu of instructions that are available for moving data around within the internal memory spaces, and the addressing modes that can be used with each one. With a 12 MHz clock, all of these instructions execute in either 1 or 2 μs.

The MOV <dest>, <src> instruction allows data to be transferred between any two internal RAM or SFR locations without going through the Accumulator. Remember the Upper 128 byes of data RAM can be accessed only by indirect addressing, and SFR space only by direct addressing.

Note that in all MCS-51 devices, the stack resides in on-chip RAM, and grows upwards. The PUSH instruction first increments the Stack Pointer (SP), then copies the byte into the stack. PUSH and POP use only direct addressing to identify the byte being saved or restored.

Table 4. A List of the MCS®-51 Data Transfer Instructions that Access Internal Data Memory Space

Mnemonic	Operation	Addressing Modes				Execution Time (μs)
		Dir	Ind	Reg	Imm	
MOV A,<src>	A = <src>	X	X	X	X	1
MOV <dest>,A	<dest> = A	X	X	X		1
MOV <dest>, <src>	<dest> = <src>	X	X	X	X	2
MOV DPTR,#data16	DPTR = 16-bit immediate constant.				X	2
PUSH <src>	INC SP : MOV "@SP",<src>	X				2
POP <dest>	MOV <dest>,"@SP" : DEC SP	X				2
XCH A,<byte>	ACC and <byte> exchange data	X	X	X		1
XCHD A,@Ri	ACC and @Ri exchange low nibbles		X			1

but the stack itself is accessed by indirect addressing using the SP register. This means the stack can go into the Upper 128, if they are implemented, but not into SFR space.

In devices that do not implement the Upper 128, if the SP points to the Upper 128, PUSHed bytes are lost, and POPped bytes are indeterminate.

The Data Transfer instructions include a 16-bit MOV that can be used to initialize the Data Pointer (DPTR) for look-up tables in Program Memory, or for 16-bit external Data Memory accesses.

The XCH A, <byte> instruction causes the Accumulator and addressed byte to exchange data. The XCHD A,@Ri instruction is similar, but only the low nibbles are involved in the exchange.

To see how XCH and XCHD can be used to facilitate data manipulations, consider first the problem of shifting an 8-digit BCD number two digits to the right. Figure 11 shows how this can be done using direct MOVs, and for comparison how it can be done using XCH instructions. To aid in understanding how the code works, the contents of the registers that are holding the BCD number and the content of the Accumulator are shown alongside each instruction to indicate their status after the instruction has been executed.

		2A	2B	2C	2D	2E	ACC
MOV	A,2EH	00	12	34	56	78	78
MOV	2EH,2DH	00	12	34	56	56	78
MOV	2DH,2CH	00	12	34	34	56	78
MOV	2CH,2BH	00	12	12	34	56	78
MOV	2BH,#0	00	00	12	34	56	78

(a) Using direct MOVs: 14 bytes, 9 μs

		2A	2B	2C	2D	2E	ACC
CLR	A	00	12	34	56	78	00
XCH	A,2BH	00	00	34	56	78	12
XCH	A,2CH	00	00	12	56	78	34
XCH	A,2DH	00	00	12	34	78	56
XCH	A,2EH	00	00	12	34	56	78

(b) Using XCHs: 9 bytes, 5 μs

Figure 11. Shifting a BCD Number Two Digits to the Right

After the routine has been executed, the Accumulator contains the two digits that were shifted out on the right. Doing the routine with direct MOVs uses 14 code bytes and 9 μs of execution time (assuming a 12 MHz clock). The same operation with XCHs uses less code and executes almost twice as fast.

To right-shift by an odd number of digits, a one-digit shift must be executed. Figure 12 shows a sample of code that will right-shift a BCD number one digit, using the XCHD instruction. Again, the contents of the registers holding the number and of the Accumulator are shown alongside each instruction.

		2A	2B	2C	2D	2E	ACC
	MOV R1,#2EH	00	12	34	56	78	XX
	MOV R0,#2DH	00	12	34	56	78	XX
	loop for R1 = 2EH:						
LOOP:	MOV A,@R1	00	12	34	56	78	78
	XCHD A,@R0	00	12	34	58	78	76
	SWAP A	00	12	34	58	78	67
	MOV @R1,A	00	12	34	58	67	67
	DEC R1	00	12	34	58	67	67
	DEC R0	00	12	34	58	67	67
	CJNE R1,#2AH,LOOP						
	loop for R1 = 2DH:	00	12	38	45	67	45
	loop for R1 = 2CH:	00	18	23	45	67	23
	loop for R1 = 2BH:	08	01	23	45	67	01
	CLR A	08	01	23	45	67	00
	XCH A,2AH	00	01	23	45	67	08

Figure 12. Shifting a BCD Number One Digit to the Right

First, pointers R1 and R0 are set up to point to the two bytes containing the last four BCD digits. Then a loop is executed which leaves the last byte, location 2EH, holding the last two digits of the shifted number. The pointers are decremented, and the loop is repeated for location 2DH. The CJNE instruction (Compare and Jump if Not Equal) is a loop control that will be described later.

The loop is executed from LOOP to CJNE for R1 = 2EH, 2DH, 2CH and 2BH. At that point the digit that was originally shifted out on the right has propagated to location 2AH. Since that location should be left with 0s, the lost digit is moved to the Accumulator.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EXTERNAL RAM

Table 5 shows a list of the Data Transfer instructions that access external Data Memory. Only indirect addressing can be used. The choice is whether to use a one-byte address, @Ri, where Ri can be either R0 or R1 of the selected register bank, or a two-byte address, @DPTR. The disadvantage to using 16-bit addresses if only a few K bytes of external RAM are involved is that 16-bit addresses use all 8 bits of Port 2 as address bus. On the other hand, 8-bit addresses allow one to address a few K bytes of RAM, as shown in Figure 5, without having to sacrifice all of Port 2.

All of these instructions execute in 2 μs, with a 12 MHz clock.

Table 5. A List of the MCS®-51 Data Transfer Instructions that Access External Data Memory Space

Address Width	Mnemonic	Operation	Execution Time (μs)
8 bits	MOVX A,@Ri	Read external RAM @Ri	2
8 bits	MOVX @Ri,A	Write external RAM @Ri	2
16 bits	MOVX A,@DPTR	Read external RAM @DPTR	2
16 bits	MOVX @DPTR,A	Write external RAM @DPTR	2

Note that in all external Data RAM accesses, the Accumulator is always either the destination or source of the data.

The read and write strobes to external RAM are activated only during the execution of a MOVX instruction. Normally these signals are inactive, and in fact if they're not going to be used at all, their pins are available as extra I/O lines. More about that later.

LOOKUP TABLES

Table 6 shows the two instructions that are available for reading lookup tables in Program Memory. Since these instructions access only Program Memory, the lookup tables can only be read, not updated. The mnemonic is MOVC for "move constant"

If the table access is to external Program Memory, then the read strobe is $\overline{PSEN}$.

Table 6. The MCS®-51 Lookup Table Read Instructions

Mnemonic	Operation	Execution Time (μs)
MOVC A,@A + DPTR	Read Pgm Memory at (A + DPTR)	2
MOVC A,@A + PC	Read Pgm Memory at (A + PC)	2

The first MOVC instruction in Table 6 can accommodate a table of up to 256 entries, numbered 0 through 255. The number of the desired entry is loaded into the Accumulator, and the Data Pointer is set up to point to beginning of the table. Then

```
MOVC    A,@A + DPTR
```

copies the desired table entry into the Accumulator.

The other MOVC instruction works the same way, except the Program Counter (PC) is used as the table base, and the table is accessed through a subroutine. First the number of the desired entry is loaded into the Accumulator, and the subroutine is called:

```
        MOV     A,ENTRY_NUMBER
        CALL    TABLE
```

The subroutine "TABLE" would look like this:

```
TABLE:  MOVC    A,@A + PC
        RET
```

The table itself immediately follows the RET (return) instruction in Program Memory. This type of table can have up to 255 entries, numbered 1 through 255. Number 0 can not be used, because at the time the MOVC instruction is executed, the PC contains the address of the RET instruction. An entry numbered 0 would be the RET opcode itself.

Boolean Instructions

MCS-51 devices contain a complete Boolean (single-bit) processor. The internal RAM contains 128 addressable bits, and the SFR space can support up to 128 other addressable bits. All of the port lines are bit-addressable, and each one can be treated as a separate single-bit port. The instructions that access these bits are not just conditional branches, but a complete menu of move, set, clear, complement, OR, and AND instructions. These kinds of bit operations are not easily obtained in other architectures with any amount of byte-oriented software.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 7. A List of the MCS®-51 Boolean Instructions

Mnemonic		Operation	Execution Time (μs)
ANL	C,bit	C = C .AND. bit	2
ANL	C,/bit	C = C .AND. .NOT. bit	2
ORL	C,bit	C = C .OR. bit	2
ORL	C,/bit	C = C .OR. .NOT. bit	2
MOV	C,bit	C = bit	1
MOV	bit,C	bit = C	2
CLR	C	C = 0	1
CLR	bit	bit = 0	1
SETB	C	C = 1	1
SETB	bit	bit = 1	1
CPL	C	C = .NOT. C	1
CPL	bit	bit = .NOT. bit	1
JC	rel	Jump if C = 1	2
JNC	rel	Jump if C = 0	2
JB	bit,rel	Jump if bit = 1	2
JNB	bit,rel	Jump if bit = 0	2
JBC	bit,rel	Jump if bit = 1; CLR bit	2

The instruction set for the Boolean processor is shown in Table 7. All bit accesses are by direct addressing. Bit addresses 00H through 7FH are in the Lower 128, and bit addresses 80H through FFH are in SFR space.

Note how easily an internal flag can be moved to a port pin:

```
MOV     C,FLAG
MOV     P1.0,C
```

In this example, FLAG is the name of any addressable bit in the Lower 128 or SFR space. An I/O line (the LSB of Port 1, in this case) is set or cleared depending on whether the flag bit is 1 or 0.

The Carry bit in the PSW is used as the single-bit Accumulator of the Boolean processor. Bit instructions that refer to the Carry bit as C assemble as Carry-specific instructions (CLR C, etc). The Carry bit also has a direct address, since it resides in the PSW register, which is bit-addressable.

Note that the Boolean instruction set includes ANL and ORL operations, but not the XRL (Exclusive OR) operation. An XRL operation is simple to implement in software. Suppose, for example, it is required to form the Exclusive OR of two bits:

```
C = bit1 .XRL. bit2
```

The software to do that could be as follows:

```
        MOV     C,bit1
        JNB     bit2,OVER
        CPL     C
OVER:   (continue)
```

First, bit1 is moved to the Carry. If bit2 = 0, then C now contains the correct result. That is, bit1 .XRL. bit2 = bit1 if bit2 = 0. On the other hand, if bit2 = 1 C now contains the complement of the correct result. It need only be inverted (CPL C) to complete the operation.

This code uses the JNB instruction, one of a series of bit-test instructions which execute a jump if the addressed bit is set (JC, JB, JBC) or if the addressed bit is not set (JNC, JNB). In the above case, bit2 is being tested, and if bit2 = 0 the CPL C instruction is jumped over.

JBC executes the jump if the addressed bit is set, and also clears the bit. Thus a flag can be tested and cleared in one operation.

All the PSW bits are directly addressable, so the Parity bit, or the general purpose flags, for example, are also available to the bit-test instructions.

RELATIVE OFFSET

The destination address for these jumps is specified to the assembler by a label or by an actual address in Program Memory. However, the destination address assembles to a relative offset byte. This is a signed (two's complement) offset byte which is added to the PC in two's complement arithmetic if the jump is executed.

The range of the jump is therefore −128 to +127 Program Memory bytes relative to the first byte following the instruction.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Jump Instructions

Table 8 shows the list of unconditional jumps.

Table 8. Unconditional Jumps in MCS®-51 Devices

Mnemonic	Operation	Execution Time (μs)
JMP addr	Jump to addr	2
JMP @A + DPTR	Jump to A + DPTR	2
CALL addr	Call subroutine at addr	2
RET	Return from subroutine	2
RETI	Return from interrupt	2
NOP	No operation	1

The Table lists a single "JMP addr" instruction, but in fact there are three—SJMP, LJMP and AJMP—which differ in the format of the destination address. JMP is a generic mnemonic which can be used if the programmer does not care which way the jump is encoded.

The SJMP instruction encodes the destination address as a relative offset, as described above. The instruction is 2 bytes long, consisting of the opcode and the relative offset byte. The jump distance is limited to a range of −128 to +127 bytes relative to the instruction following the SJMP.

The LJMP instruction encodes the destination address as a 16-bit constant. The instruction is 3 bytes long, consisting of the opcode and two address bytes. The destination address can be anywhere in the 64K Program Memory space.

The AJMP instruction encodes the destination address as an 11-bit constant. The instruction is 2 bytes long, consisting of the opcode, which itself contains 3 of the 11 address bits, followed by another byte containing the low 8 bits of the destination address. When the instruction is executed, these 11 bits are simply substituted for the low 11 bits in the PC. The high 5 bits stay the same. Hence the destination has to be within the same 2K block as the instruction following the AJMP.

In all cases the programmer specifies the destination address to the assembler in the same way: as a label or as a 16-bit constant. The assembler will put the destination address into the correct format for the given instruction. If the format required by the instruction will not support the distance to the specified destination address, a "Destination out of range" message is written into the List file.

The JMP @A + DPTR instruction supports case jumps. The destination address is computed at execution time as the sum of the 16-bit DPTR register and the Accumulator. Typically, DPTR is set up with the address of a jump table, and the Accumulator is given an index to the table. In a 5-way branch, for example, an integer 0 through 4 is loaded into the Accumulator. The code to be executed might be as follows:

```
MOV     DPTR,#JUMP_TABLE
MOV     A,INDEX_NUMBER
RL      A
JMP     @A + DPTR
```

The RL A instruction converts the index number (0 through 4) to an even number on the range 0 through 8, because each entry in the jump table is 2 bytes long:

```
JUMP_TABLE:
        AJMP    CASE_0
        AJMP    CASE_1
        AJMP    CASE_2
        AJMP    CASE_3
        AJMP    CASE_4
```

Table 8 shows a single "CALL addr" instruction, but there are two of them—LCALL and ACALL—which differ in the format in which the subroutine address is given to the CPU. CALL is a generic mnemonic which can be used if the programmer does not care which way the address is encoded.

The LCALL instruction uses the 16-bit address format, and the subroutine can be anywhere in the 64K Program Memory space. The ACALL instruction uses the 11-bit format, and the subroutine must be in the same 2K block as the instruction following the ACALL.

In any case the programmer specifies the subroutine address to the assembler in the same way: as a label or as a 16-bit constant. The assembler will put the address into the correct format for the given instructions.

Subroutines should end with a RET instruction, which returns execution to the instruction following the CALL.

RETI is used to return from an interrupt service routine. The only difference between RET and RETI is that RETI tells the interrupt control system that the interrupt in progress is done. If there is no interrupt in progress at the time RETI is executed, then the RETI is functionally identical to RET.

Table 9 shows the list of conditional jumps available to the MCS-51 user. All of these jumps specify the destination address by the relative offset method, and so are limited to a jump distance of −128 to +127 bytes from the instruction following the conditional jump instruction. Important to note, however, the user specifies to the assembler the actual destination address the same way as the other jumps: as a label or a 16-bit constant.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น

Table 9. Conditional Jumps in MCS®-51 Devices

Mnemonic	Operation	Addressing Modes				Execution Time (μs)
		Dir	Ind	Reg	Imm	
JZ rel	Jump if A = 0	Accumulator only				2
JNZ rel	Jump if A = 0	Accumulator only				2
DJNZ <byte>,rel	Decrement and jump if not zero	X		X		2
CJNE A,<byte>,rel	Jump if A = <byte>	X			X	2
CJNE <byte>,#data,rel	Jump if <byte> ≠ #data		X	X		2

There is no Zero bit in the PSW. The JZ and JNZ instructions test the Accumulator data for that condition.

The DJNZ instruction (Decrement and Jump if Not Zero) is for loop control. To execute a loop N times, load a counter byte with N and terminate the loop with a DJNZ to the beginning of the loop, as shown below for N = 10:

```
        MOV     COUNTER,#10
LOOP:   (begin loop)
          .
          .
          .
        (end loop)
        DJNZ    COUNTER,LOOP
        (continue)
```

The CJNE instruction (Compare and Jump if Not Equal) can also be used for loop control as in Figure 12. Two bytes are specified in the operand field of the instruction. The jump is executed only if the two bytes are not equal. In the example of Figure 12, the two bytes were the data in R1 and the constant 2AH. The initial data in R1 was 2EH. Every time the loop was executed, R1 was decremented, and the looping was to continue until the R1 data reached 2AH.

Another application of this instruction is in "greater than, less than" comparisons. The two bytes in the operand field are taken as unsigned integers. If the first is less than the second, then the Carry bit is set (1). If the first is greater than or equal to the second, then the Carry bit is cleared.

CPU TIMING

All MCS-51 microcontrollers have an on-chip oscillator which can be used if desired as the clock source for the CPU. To use the on-chip oscillator, connect a crystal or ceramic resonator between the XTAL1 and XTAL2 pins of the microcontroller, and capacitors to ground as shown in Figure 13.

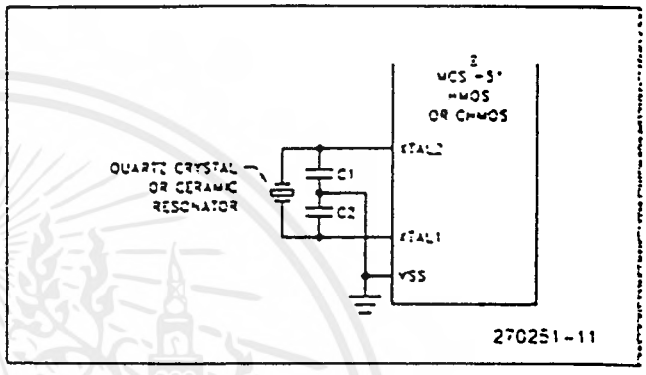
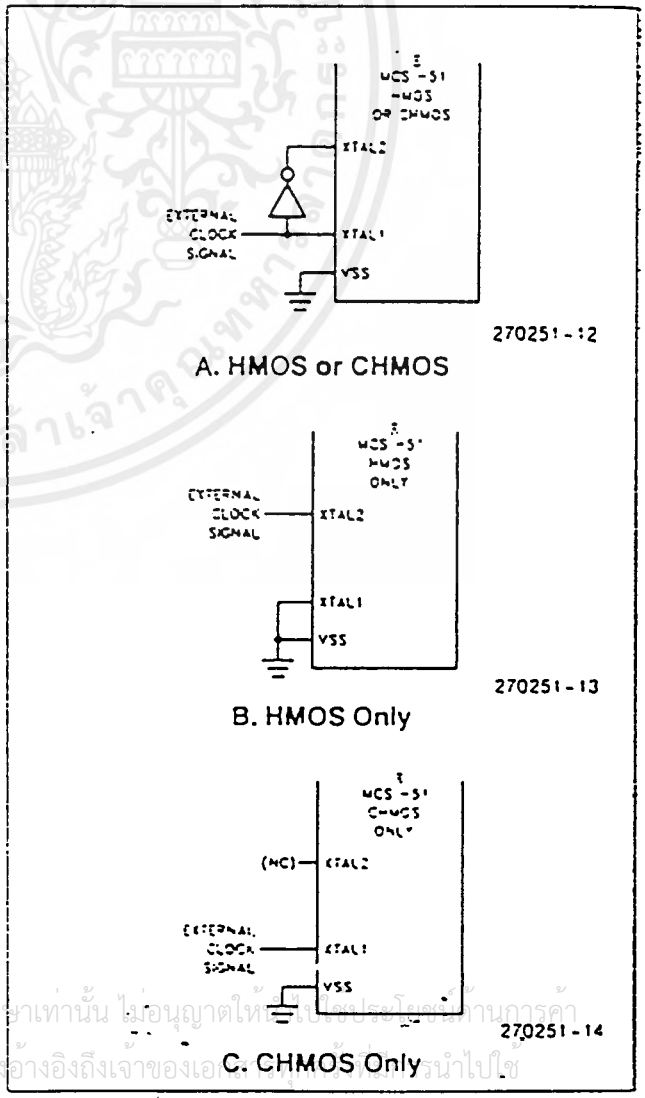


Figure 13. Using the On-Chip Oscillator



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Examples of how to drive the clock with an external oscillator are shown in Figure 14. Note that in the HMOS devices (8051, etc.) the signal at the XTAL2 pin actually drives the internal clock generator. In the CHMOS devices (80C51BH, etc.) the signal at the XTAL1 pin drives the internal clock generator. If only one pin is going to be driven with the external oscillator signal, make sure it is the right pin.

The internal clock generator defines the sequence of states that make up the MCS-51 machine cycle.

Machine Cycles

A machine cycle consists of a sequence of 6 states, numbered S1 through S6. Each state time lasts for two oscillator periods. Thus a machine cycle takes 12 oscillator periods or 1 μs if the oscillator frequency is 12 MHz.

Each state is divided into a Phase 1 half and a Phase 2 half. Figure 15 shows the fetch/execute sequences in

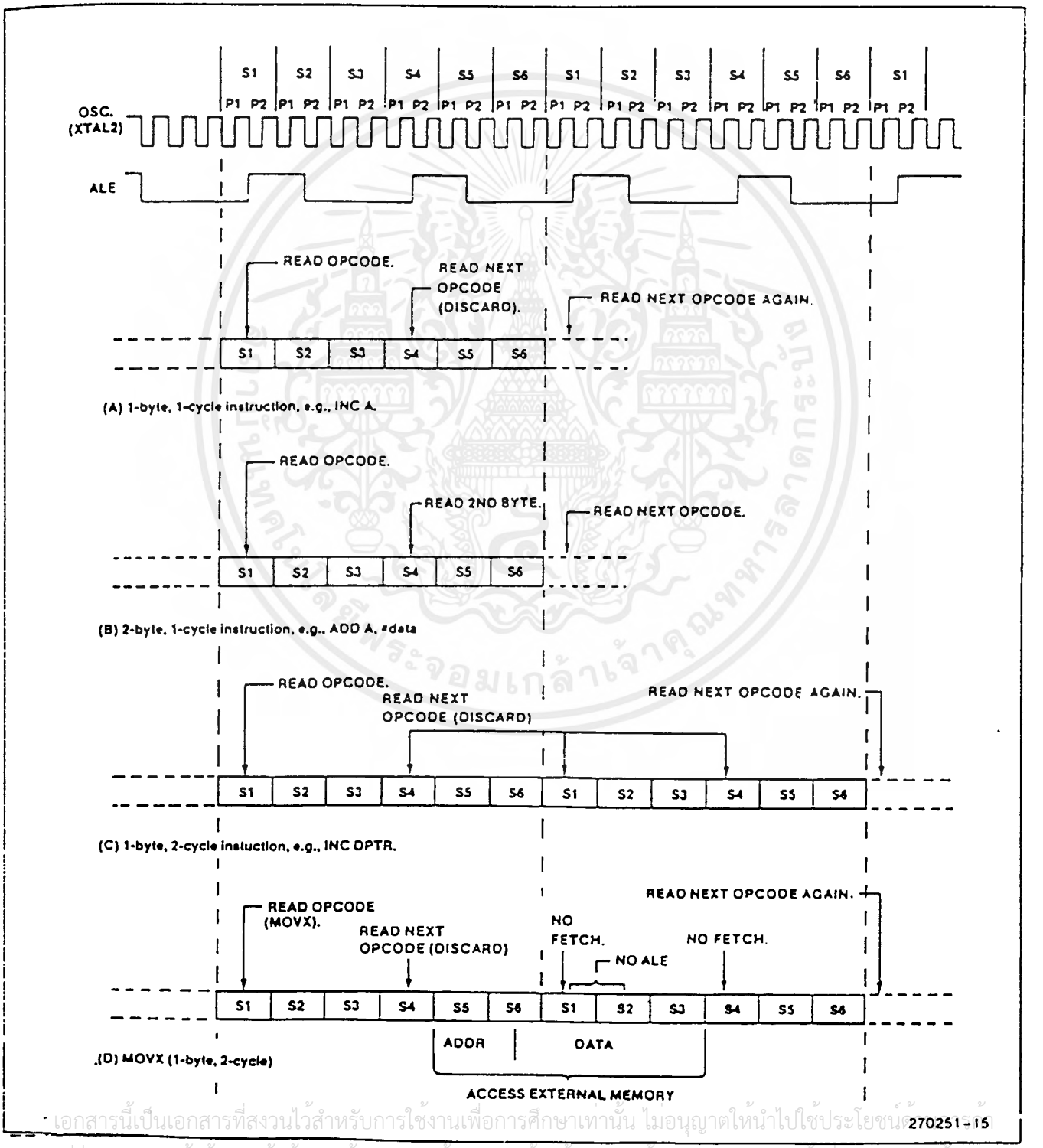


Figure 15. State Sequences in MCS®-51 Devices

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

states and phases for various kinds of instructions. Normally two program fetches are generated during each machine cycle, even if the instruction being executed doesn't require it. If the instruction being executed doesn't need more code bytes, the CPU simply ignores the extra fetch, and the Program Counter is not incremented.

Execution of a one-cycle instruction (Figure 15A and B) begins during State 1 of the machine cycle, when the opcode is latched into the Instruction Register. A second fetch occurs during S4 of the same machine cycle. Execution is complete at the end of State 6 of this machine cycle.

The MOVX instructions take two machine cycles to execute. No program fetch is generated during the second cycle of a MOVX instruction. This is the only time program fetches are skipped. The fetch/execute sequence for MOVX instructions is shown in Figure 15(D).

The fetch/execute sequences are the same whether the Program Memory is internal or external to the chip. Execution times do not depend on whether the Program Memory is internal or external.

Figure 16 shows the signals and timing involved in program fetches when the Program Memory is external. If Program Memory is external, then the Program Memory read strobe $\overline{PSEN}$ is normally activated twice per machine cycle, as shown in Figure 16(A).

If an access to external Data Memory occurs, as shown in Figure 16(B), two $\overline{PSEN}$s are skipped, because the address and data bus are being used for the Data Memory access.

Note that a Data Memory bus cycle takes twice as much time as a Program Memory bus cycle. Figure 16 shows the relative timing of the addresses being emitted at Ports 0 and 2, and of ALE and $\overline{PSEN}$. ALE is used to latch the low address byte from P0 into the address latch.

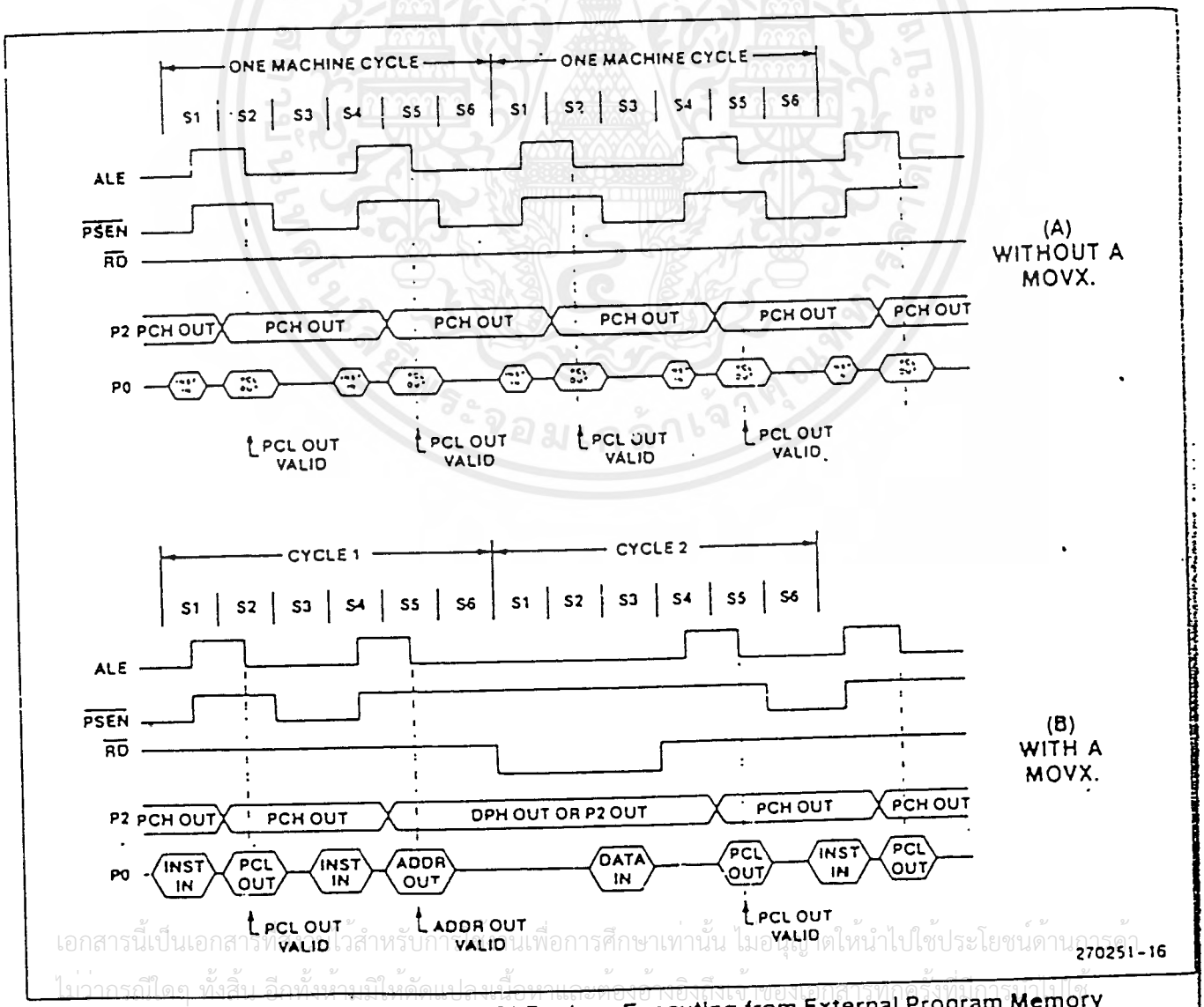


Figure 16. Bus Cycles in MCS®-51 Devices Executing from External Program Memory

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของลิขสิทธิ์ทุกครั้งที่มีการนำไปใช้

When the CPU is executing from internal Program Memory, $\overline{PSEN}$ is not activated, and program addresses are not emitted. However, ALE continues to be activated twice per machine cycle and so is available as a clock output signal. Note, however, that one ALE is skipped during the execution of the MOVX instruction.

Interrupt Structure

The 8051 core provides 5 interrupt sources: 2 external interrupts, 2 timer interrupts, and the serial port interrupt. What follows is an overview of the interrupt structure for the 8051. Other MCS-51 devices have additional interrupt sources and vectors as shown in Table 1. Refer to the appropriate chapters on other devices for further information on their interrupts.

INTERRUPT ENABLES

Each of the interrupt sources can be individually enabled or disabled by setting or clearing a bit in the SFR named IE (Interrupt Enable). This register also contains a global disable bit, which can be cleared to disable all interrupts at once. Figure 17 shows the IE register for the 8051.

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

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

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

*These reserved bits are used in other MCS-51 devices.

Figure 17. IE (Interrupt Enable) Register in the 8051

INTERRUPT PRIORITIES

Each interrupt source can also be individually programmed to one of two priority levels by setting or clearing a bit in the SFR named IP (Interrupt Priority). Figure 18 shows the IP register in the 8051.

A low-priority interrrupt can be interrupted by a high-priority interrupt, but not by another low-priority interrupt. A high-priority interrupt can't be interrupted by any other interrupt source.

If two interrupt requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If interrupt requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence.

Figure 19 shows, for the 8051, how the IE and IP registers and the polling sequence work to determine which if any interrupt will be serviced.

(MSB)							(LSB)
—	—	—	PS	PT1	PX1	PT0	PX0

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

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

*These reserved bits are used in other MCS-51 devices

Figure 18. IP (Interrupt Priority) Register in the 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

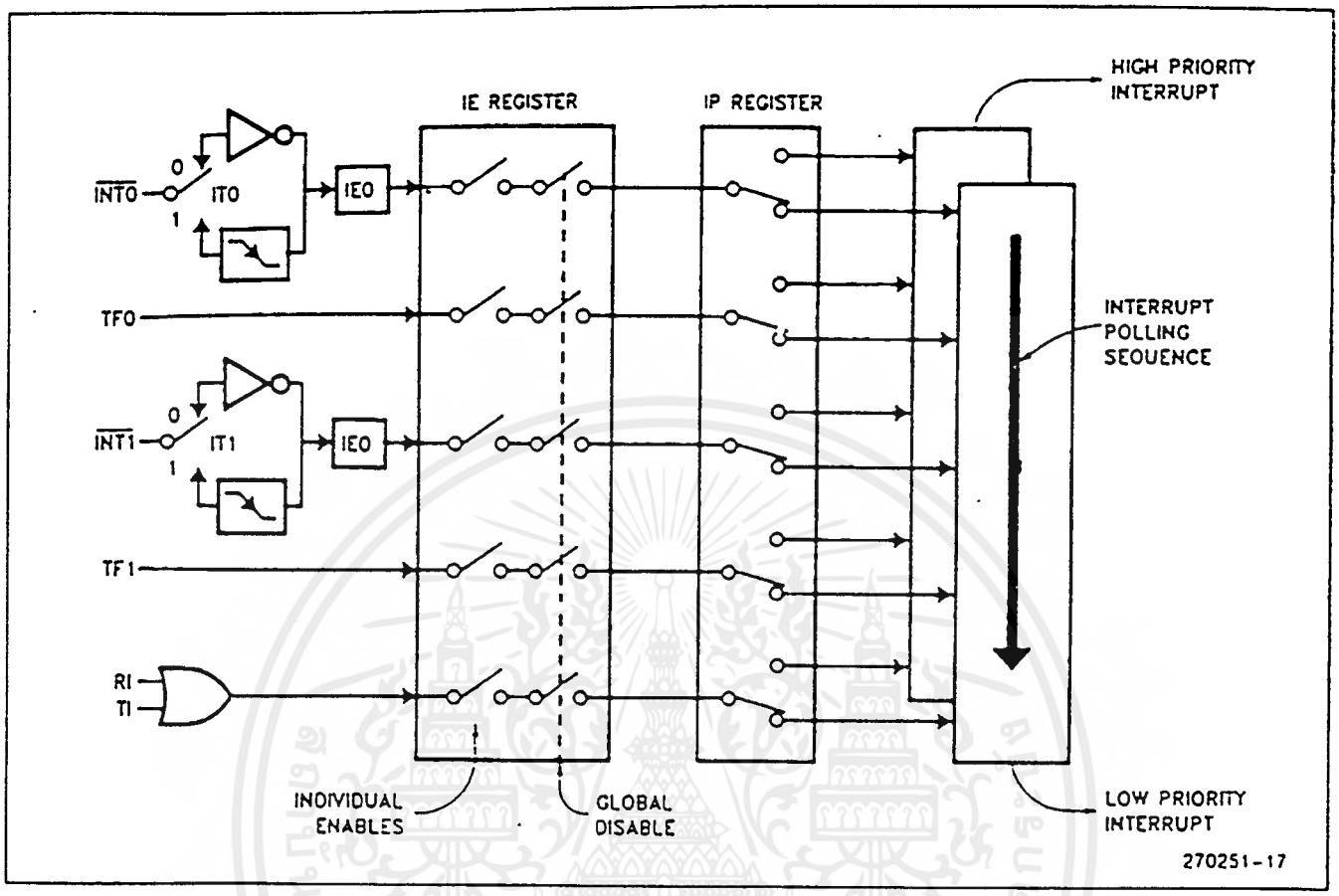


Figure 19. 8051 Interrupt Control System

In operation, all the interrupt flags are latched into the interrupt control system during State 5 of every machine cycle. The samples are polled during the following machine cycle. If the flag for an enabled interrupt is found to be set (1), the interrupt system generates an LCALL to the appropriate location in Program Memory, unless some other condition blocks the interrupt. Several conditions can block an interrupt, among them that an interrupt of equal or higher priority level is already in progress.

The hardware-generated LCALL causes the contents of the Program Counter to be pushed onto the stack, and reloads the PC with the beginning address of the service routine. As previously noted (Figure 3), the service routine for each interrupt begins at a fixed location.

Only the Program Counter is automatically pushed onto the stack, not the PSW or any other register. Having only the PC be automatically saved allows the programmer to decide how much time to spend saving which other registers. This enhances the interrupt response time, albeit at the expense of increasing the programmer's burden of responsibility. As a result, many interrupt functions that are typical in control applications—toggling a port pin, for example, or reloading a timer, or unloading a serial buffer—can often be completed in less time than it takes other architectures to commence them.

SIMULATING A THIRD PRIORITY LEVEL IN SOFTWARE

Some applications require more than the two priority levels that are provided by on-chip hardware in MCS-51 devices. In these cases, relatively simple software can be written to produce the same effect as a third priority level.

First, interrupts that are to have higher priority than 1 are assigned to priority 1 in the IP (Interrupt Priority) register. The service routines for priority 1 interrupts that are supposed to be interruptible by "priority 2" interrupts are written to include the following code:

```
        PUSH    IE
        MOV     IE,#MASK
        CALL    LABEL
          .....
(execute service routine)
          .....
        POP     IE
        RET
LABEL:  RETI
```

ไฟล์นี้จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

As soon as any priority 1 interrupt is acknowledged, the IE (Interrupt Enable) register is re-defined so as to disable all but "priority 2" interrupts. Then, a CALL to LABEL executes the RETI instruction, which clears the priority 1 interrupt-in-progress flip-flop. At this point any priority 1 interrupt that is enabled can be serviced, but only "priority 2" interrupts are enabled.

POPping IE restores the original enable byte. Then a normal RET (rather than another RETI) is used to terminate the service routine. The additional software adds 10 μs (at 12 MHz) to priority 1 interrupts.

ADDITIONAL REFERENCES

The following application notes are found in the *Embedded Control Applications* handbook. (Order Number: 270648)

1. AP-69 "An Introduction to the Intel MCS®-51 Single-Chip Microcomputer Family"
2. AP-70 "Using the Intel MCS®-51 Boolean Processing Capabilities"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MH88510/11
Subscriber Line Interface Circuit (SLIC)

ISSUE 3 January 1994

Features

- High Gain Version MH88511
- Compatible with popular MH88500
- Operates with a wide range of battery voltages
- Constant current battery feed
- Dry line compatible
- Overvoltage and short circuit protection
- Ringing feed
- Off-hook detection and LED indicator drive
- Dial pulse detection
- Ring trip filter with auto ring trip
- Relay driver
- Transformerless 2-2 wire conversion
- Low power consumption
- Mute of incoming audio
- Few external components

Applications

Line interface for:

- PABX
- Intercom
- Key Telephone System

Ordering Information

MH88510 20-Pin SIL Package
MH88511 20-Pin SIL Package
0°C to 70°C

Description

The Mitel MH88510/11 Subscriber Line Interface Circuit provides a complete interface between the telephone line and a speech switch requiring only a single bidirectional switch per crosspoint. The functions provided by the MH88510/11 include bidirectional differential to single-ended conversion in the speech path, line battery feed, ringing feed and loop and dial pulse detection. The device is fabricated as a thick film hybrid which incorporates various technologies for optimum circuit design and very high reliability.

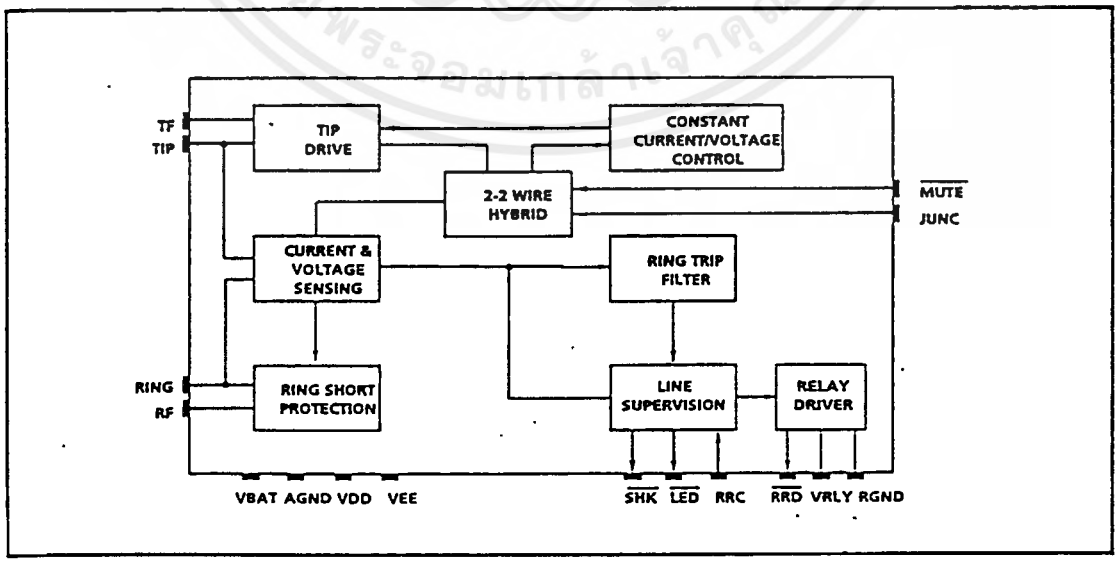


Figure 1 - Functional Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

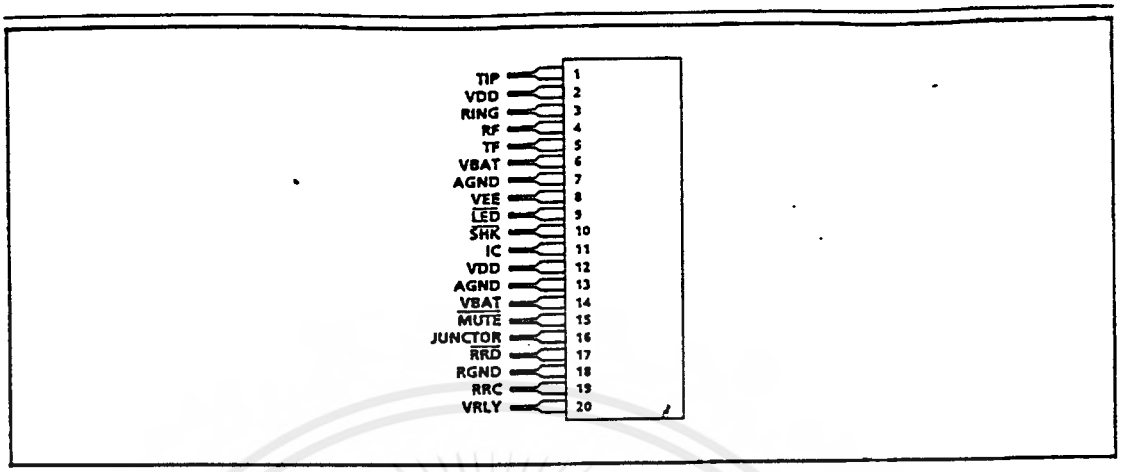


Figure 2 - Pin Connections

Pin Description

Pin	Name	Description
1	TIP	**Tip Lead.** Connects to the "Tip" lead (A-wire) of the telephone line.
2	VDD	**Positive Power Supply Voltage.** Normally +5V. This provides current for both internal circuitry as well as the loop. Not internally connected to Pin 12.
3	RING	**Ring Lead.** Connects to the "Ring" lead (B-wire) of the telephone line.
4	RF	Connects to Ring Relay contact see Figure 5.
5	TF	Leave open circuit.
6	VBAT	**Battery voltage supply.** Normally -24V or -48V. Not internally connected to Pin 14.
7	AGND	**Analog ground.** Supply and battery ground. Internally connected to pin 13. For optimum performance connect pin 7 to Pin 13.
8	VEE	**Negative Analog Supply Voltage.** Normally -5V
9	$\overline{\text{LED}}$	**LED Drive Output.** Drives an LED directly. (Off-Hook condition, logic low).
10	$\overline{\text{SHK}}$	**Switch Hook Detect Output.** A logic low indicates Off-Hook condition.
11	IC	**Internal Connection.** This pin is connected internally.
12	VDD	**Positive Power Supply Voltage.** Normally +5V. This provides current for both internal circuitry as well as the loop. Not internally connected to Pin 2.
13	AGND	**Analog ground.** Supply and battery ground. Internally connected to pin 13. For optimum performance connect pin 13 to Pin 7.
14	VBAT	**Battery voltage supply.** Normally -24V or -48V. Not internally connected to Pin 6.
15	$\overline{\text{MUTE}}$	**Mute Input.** A logic low will mute signals coming from TIP-RING to the JUNC.
16	JUNC	**Receive/transmit audio speech path.** (Referenced to 0V GND).
17	$\overline{\text{RRD}}$	**Ring Relay Drive Output.** Connects to the ring relay coil. A logic low activates the relay.
18	RGND	**Relay Ground.** Return path for relay supply voltage. Normally connected to AGND.
19	RRC	**Ring Relay Control Input.** A logic high activates the Ring Relay Drive ($\overline{\text{RRD}}$) output.
20	VRLY	**Relay Positive Supply Voltage.** Normally +5V. Connects to the relay coil and the relay supply voltage. An internal clamp diode from VRLY to RGND is provided.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description

The BORSH Functions

The MH88510/11 performs all of the BORSH functions of Battery Feed, Overvoltage Protection, Ringing, Supervision and Hybrid (2-2 wire).

Battery Feed

The MH88510/11 powers the telephone set with constant DC loop current for short lines and automatically reverts to constant voltage for long lines. Since the TIP drive is powered down from the V_{DD} supply, loop current flows through both the V_{Bat} supply and the VDD supply.

Overvoltage Protection

The MH88510/11 is protected from short term (20mS) transients (+250V) between TIP and RING, TIP and ground, and RING and Ground. However, additional protection circuitry may be needed depending on the requirements which must be met. Normally, simple external shunt protection as shown in Figures 5 and 6 is all that is required.

Ringing

The ringing insertion circuitry has the capability to provide ringing voltage to the telephone set by simply adding an external relay, ring generator and a 200Ω limiting resistor. The internal relay driver switches ringing voltage onto the line via the external ring relay. The SLIC provides two internal 300Ω battery feed resistors through which the ringing current will flow. A clamp diode is included which suppresses voltage transients during relay switching caused by the relay coil. In addition, the circuit prevents connection of the ringing source during off-hook conditions. See Figure 5 for typical application.

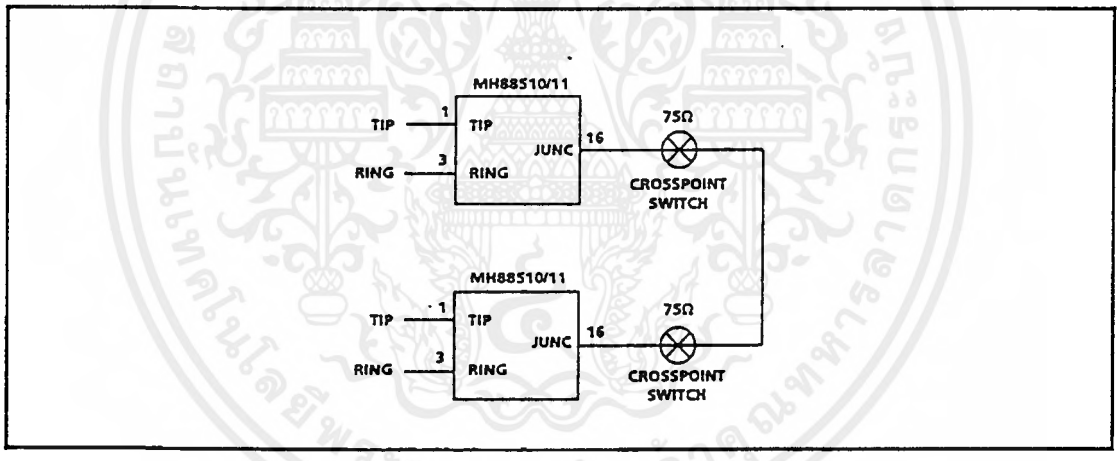


Figure 3 - SLIC and Crosspoint Switch Connection

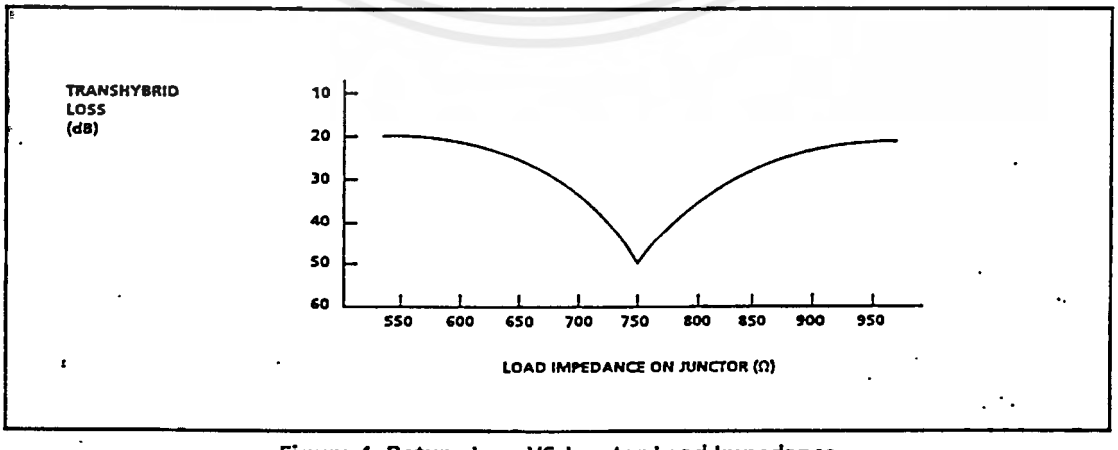


Figure 4- Return Loss VS Junctor Load Impedance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Supervision

The loop detection circuit determines whether a low enough impedance is across TIP and RING to be recognized as an off-hook condition. When an off-hook condition occurs, the $\overline{SHK}$ and $\overline{LED}$ outputs toggle to a logic low level. These outputs also toggle during incoming dial pulses. The $\overline{SHK}$ output has low drive capability while the $\overline{LED}$ output can drive an $\overline{LED}$ directly. The detection circuit engages a ringing filter during applied ringing. The ring trip detection circuit also prevents false off-hook detection due to the current associated with the AC ringing voltage as well as current transients when the ringing voltage is switched in and out.

Hybrid

The 2-2 wire hybrid circuit converts the incoming balanced signal at TIP and RING of the telephone line into a ground referenced output signal at JUNC of the SLIC, and converts the ground referenced input signal at JUNC of the SLIC into a non-balanced output signal at TIP and RING of the telephone line.

Return Loss at TIP-RING

To maximize return loss, the impedance at TIP-RING should match the SLIC's impedance (600Ω). However, the SLIC's input impedance is dependent on the JUNC termination resistance. For a 600Ω SLIC input impedance, the JUNC must be terminated with 754Ω.

Figure 3 illustrates a typical connection between two SLICs through two crosspoint switches. Optimum return loss occurs when JUNC is terminated with 754Ω. Since the JUNC input/output impedance is 604Ω and the crosspoint switches resistances are 75Ω + 75Ω, this configuration gives optimum return loss as shown in Figure 4.

MUTE

A logic low at the $\overline{MUTE}$ input results in muted signals coming from TIP and RING to the JUNC terminal while allowing signals from the JUNC terminal to TIP and RING to be transmitted.

TIP-RING Drive Circuit

The audio input ground referenced signal at JUNC is converted to a differential output signal at TIP and RING. The output signal consists of the audio signal superimposed on the DC battery feed current. The TIP- RING drive circuit is optimized for good 2-wire longitudinal balance.

In addition, the TIP-RING Drive Circuit has the capability to drive a dry line (a line with no DC current flowing); the AC Electrical Characteristics apply (except for longitudinal balance), even when the loop current drops to zero. Therefore, the MH88510/11 has the capability to drive a line much longer than 2000Ω providing the user is not concerned with loop current, $\overline{SHK}$ detection or ringing generator current.

Short Circuit Protection

The MH88510/11 is protected from long term (infinite) short circuit conditions occuring between TIP and RING, TIP and AGND, and RING and AGND.

Line Impedance

The MH88510/11's TIP-RING (Zin) impedance is fixed at 600Ω. For correct SLIC impedance, JUNC must be appropriately terminated. See AC Electrical Characteristics.

Transmit and Receive Gain

Transmit Gain (JUNC to TIP-RING) and Receive Gain (TIP-RING to JUNC) are fixed. For correct gain, the SLIC input impedance must match the line impedance and JUNC must be appropriately terminated. In the case of the MH88511, this gain has been optimised for Dry line applications.

Trunk Interface Applications

The MH88511 can be used as a trunk interface. See Figure 7. The 600Ω transformer provides the DC isolation between the Central Office and the system interface. To use the MH88511 as the trunk interface and the MH88510 as the line interface, the system will meet the loss level plan of the Chinese standard. See Figure 8a and 8b.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

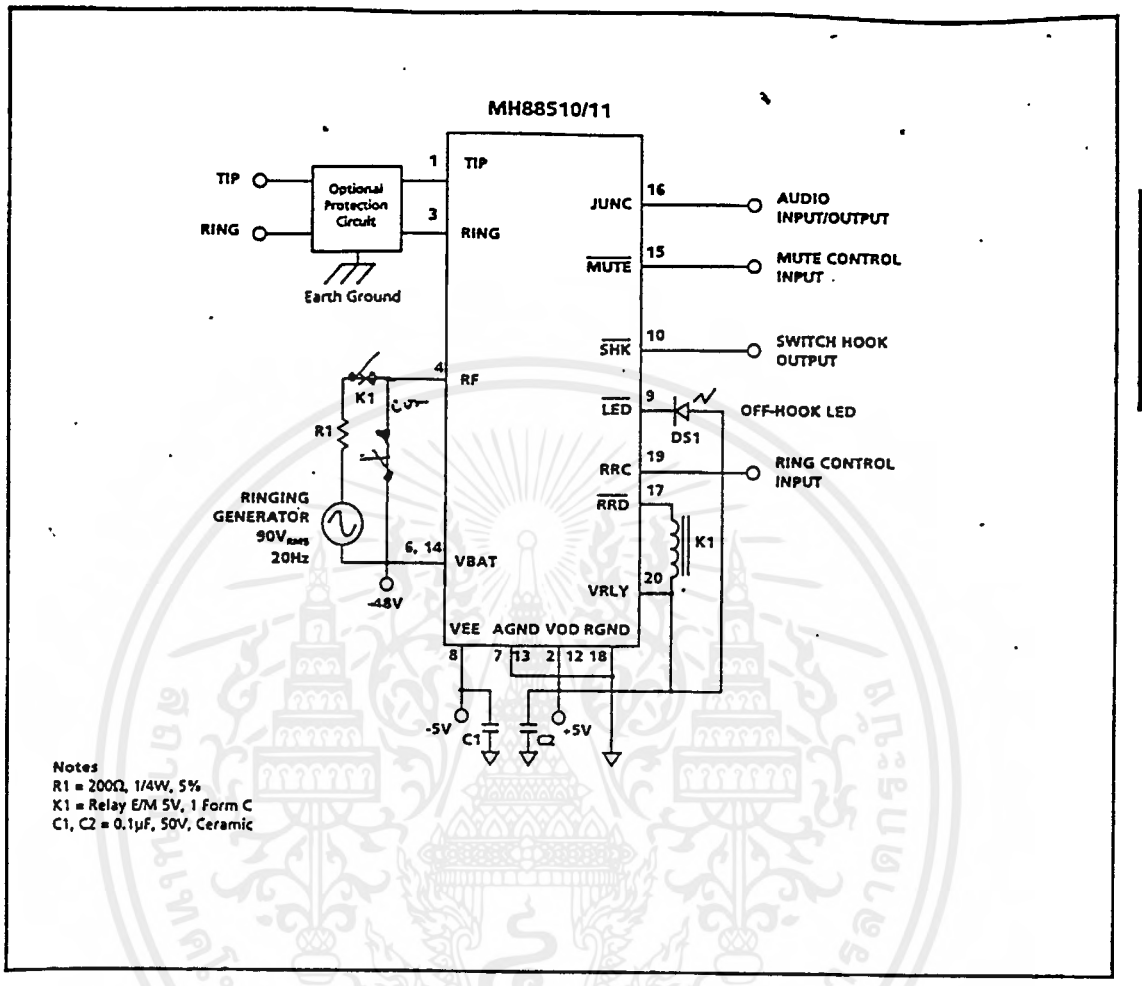


Figure 5 - Typical Application Circuit

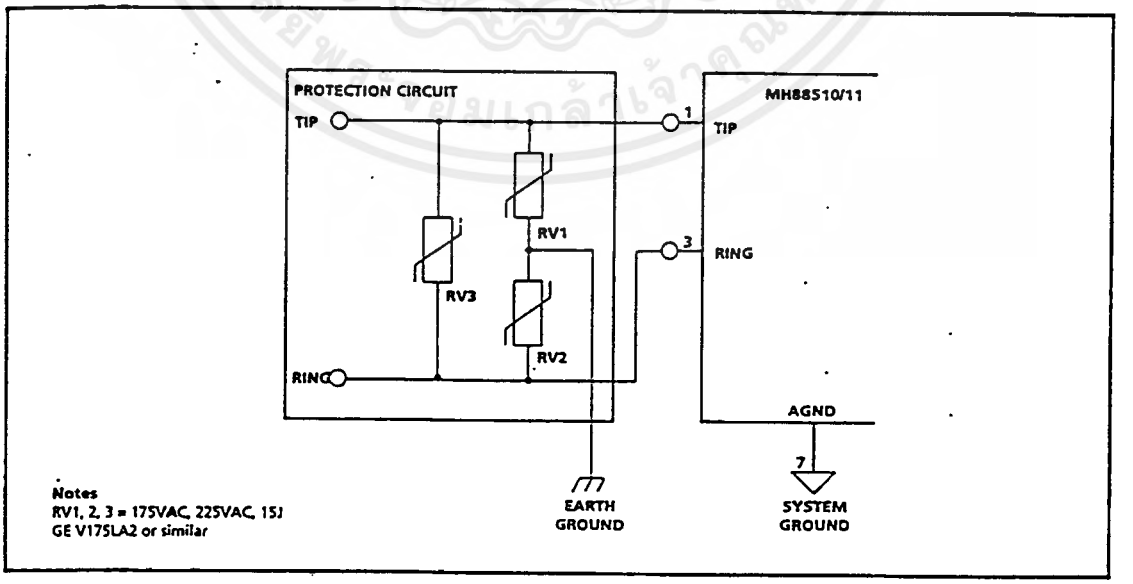


Figure 6 - Typical Application Circuit

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MH88510/11

Absolute Maximum Ratings* - Voltages are with respect to AGND.

	Parameter	Symbol	Min.	Max.	Units
1	DC Supply Voltages	V_{DD}	-.03	15	V
		V_{EE}	0.3	-15	V
2	DC Battery Voltage	V_{BAT}	0.3	-60	V
3	DC Relay Voltage	V_{RLY}	-0.3	20	V
4	AC Ring Generator Voltage			150	V_{RMS}
5	Storage Temperature	T_S	-55	125	C°

* Exceeding these values may cause permanent damage. Functional operation under these conditions is not implied.

DC Electrical Characteristics†

		Characteristics	Sym	Min	Typ‡	Max	Units	Test Conditions
1		Supply Current: Open Loop	I_{DD}			15	mA	R_{Loop}=Open
			I_{EE}			15	mA	
			I_{BAT}			15	mA	
		Normal Loop	I_{DD}			43	mA	R_{Loop}=1000Ω
			I_{EE}			15	mA	
			I_{BAT}			43	mA	
		Short Loop	I_{DD}			43	mA	R_{Loop}=0Ω
			I_{EE}			15	mA	
			I_{BAT}			43	mA	
2		Power Consumption①						
		Open Loop	PC			830	mW	R_{Loop} = Open
		Normal Loop	PC			2360	mW	R_{Loop} = 1000Ω
		Short Loop	PC			2360	mW	R_{Loop} = 0Ω
3	$\overline{SHK}$	Low Level Output Voltage②	V_{OL}	-3.0			V	I_{OL} = 2μA
		High Level Output Voltage	V_{OH}	3.0			V	I_{OH} = 2μA
4	$\overline{LED}$	Sink Current, LED to AGND②	I_{OL}	0.6			mA	V_{OL} = -1.5V
		Sink Current, LED to VDD	I_{OH}	2.5			mA	V_{OL} = 3.25V
5	$\overline{RRD}$	Sink Current, Relay to VDD	I_{OL}	100			mA	V_{OL} = 0.35V
		Clamp Diode Current	I_{CD}	150			mA	
6	RRC	High Level Input Voltage③	V_{IH}	3.5			V	I_{IL} = 1.0mA
7	$\overline{MUTE}$	Low Level Input Voltage④	V_{IL}			0.8	V	I_{IL} = 0.5mA

† DC Electrical Characteristics are over recommended operating conditions with V_{DD} at + 5.0V ±5% and V_{EE} at-5V±5% unless otherwise stated.

‡ Typical figures are at 25°C with nominal ±5V supplies and are for design aid only.

① Supply Current and Power Consumption characteristics are over recommended operating conditions with V_{DD} at 5.0V, V_{EE} at -5.0V and V_{BAT} at -48.0V. Note that loop current flows through both the V_{BAT} supply and the V_{DD} supply.

② $\overline{SHK}$ output consists of a 100kΩ resistor in series with an op-amp with a minimum output voltage swing of ±3.25V. $\overline{LED}$ output consists of a 2.5kΩ resistor in series with the $\overline{SHK}$ op-amp output.

③ RRC input consists of a 5kΩ resistor in series with the base lead of the relay driver transistor (grounded emitter).

④ The $\overline{MUTE}$ input is internally pulled up. With no input connection, the voltage level at the $\overline{MUTE}$ input is typically at 1.5V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Electrical Characteristics†

	Characteristics	Sym	Min	Typ‡	Max	Units	Test Conditions
1	Return Loss at 2-Wire		20	30		dB	Reference 600Ω @ 1kHz
2	Impedance at Junctor			604		Ω	
3	Longitudinal to Metallic Balance		50	60		dB	40Hz - 4kHz
4	Longitudinal to Junctor Balance		50	60		dB	40Hz - 4kHz
5	Signal Output Overload Level						%THD≤5%
	at 2-Wire		3.5			dBm	Reference: 600Ω
	at Junctor		3.5			dBm	Reference: 754Ω
6	Total Harmonic Distortion	THD					Input 0.5V 1kHz
	at 2-Wire				1.0	%	
	at Junctor				1.0	%	
7	Idle Channel Noise	Nc					
	at 2-Wire				12	dBmC	Reference: 600Ω
	at Junctor				12	dBmC	Reference: 754Ω
8	Power Supply Rejection Ratio at	PSRR					Ripple 0.1V 1kHz
	2-Wire and Junctor V_{DD}			25		dB	
	V_{EE}			25		dB	
	V_{BAT}			25		dB	
9	Mute Attenuation		30			dB	Input 0.5V MUTE = 0.0V @ 1kHz

AC Gains Table - MH88510

	Characteristics	Sym	Min	Typ‡	Max	Units	Test Conditions
1	Gain 2-Wire to Junctor		1.05	1.12	1.19	V/V	Input 0.5V 1kHz
			0.42	0.98	1.51	dBV	Input 0.5V 1kHz
2	Frequency Response Gain		-0.3		0.3	dB	200Hz - 3400Hz
	(relative to gain at 1kHz)						
3	Gain Junctor to 2-Wire		0.96	1.00	1.04	V/V	Input 0.5V 1kHz
			-0.35	0.0	0.35	dBV	Input 0.5V 1kHz
4	Frequency Response Gain		-0.3		0.3	dB	200Hz - 3400Hz
	(relative to gain at 1kHz)						

AC Gains Table - MH88511

	Characteristics	Sym	Min	Typ‡	Max	Units	Test Conditions
1	Gain 2-Wire to Junctor		1.15	1.19	1.22	V/V	Input 0.5V 1kHz
			1.21	1.50	1.72	dBV	Input 0.5V 1kHz
2	Frequency Response Gain		-0.3		0.3	dB	200Hz - 3400Hz
	(relative to gain at 1kHz)						
3	Gain Junctor to 2-Wire		1.04	1.08	1.11	V/V	Input 0.5V 1kHz
			0.34	0.67	0.91	dBV	Input 0.5V 1kHz
4	Frequency Response Gain		-0.3		0.3	dB	200Hz - 3400Hz
	(relative to gain at 1kHz)						

† AC Electrical Characteristics are over recommended operating conditions unless otherwise stated.
‡ Typical figures are at 25°C and are for design aid only: not guaranteed and not subject to production testing.
Note 1: All of the above test conditions use 754Ω connected between JUNC and GRD, and 600Ω connected between TIP and RING unless otherwise stated.
Note 2: All of the above test conditions use 200Hz to 6400Hz unless otherwise stated.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Recommended Operating Conditions

	Parameter	Sym	Min	Typ	Max	Units	Test Conditions
1	DC Supply Voltage	VDD	4.75	5.0	7.35	V	
		VEE	-4.75	-5.0	-8.40	V	
2	DC Battery Voltage	VBAT	-23	-48	-56	V	
3	DC Relay Voltage	VRLY		5.0	15	V	
4	AC Ring Generator Voltage			90	105	VRMS	
	Ringing Generator Frequency		17		33	Hz	
5	Operating Temperature	TOP	0		70	°C	

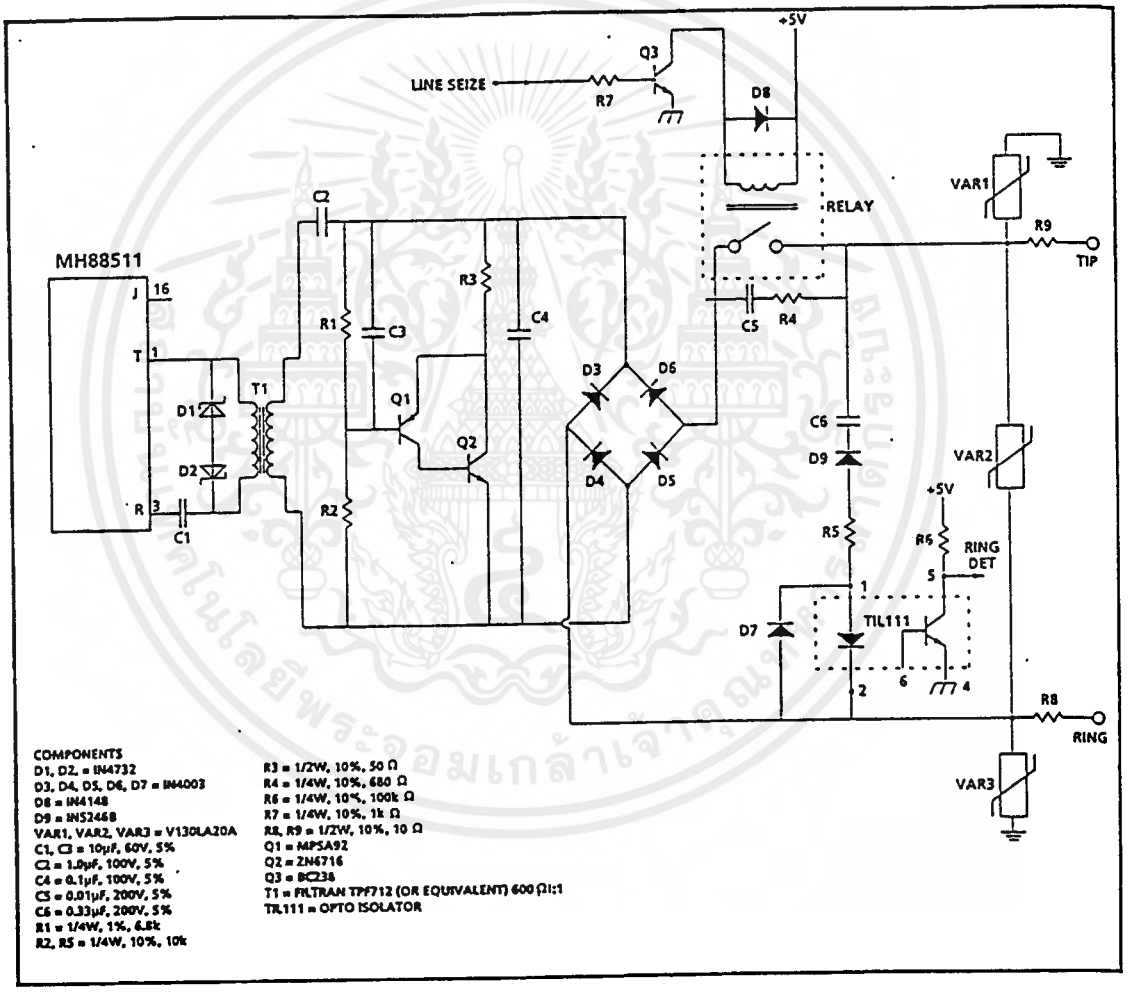


Figure 7 - MH88511 as a Trunk Interface Application

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

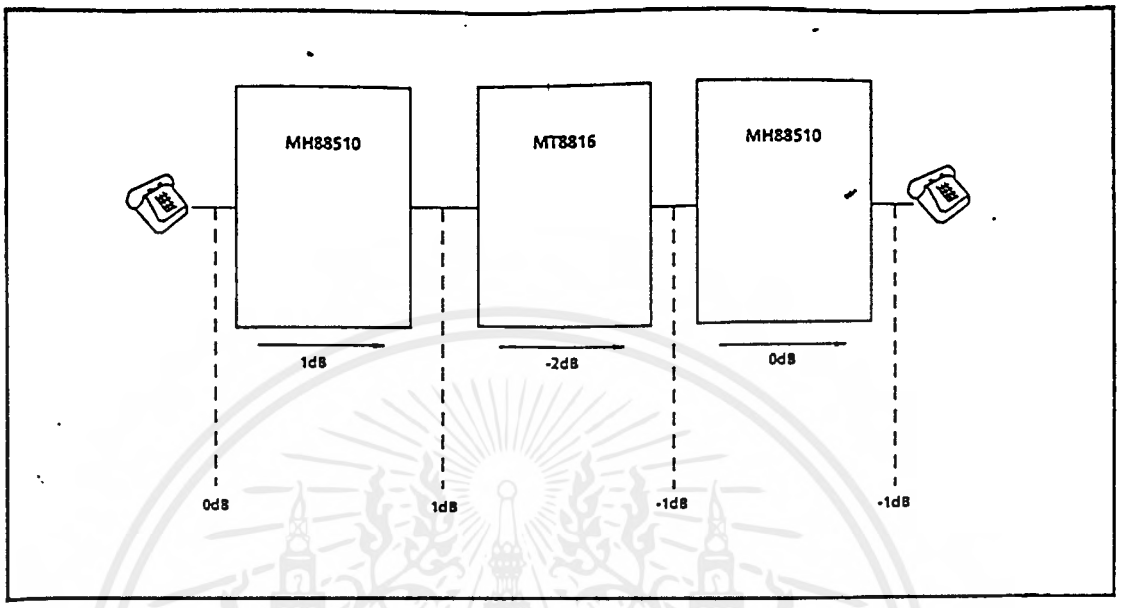


Figure 8a - Line to Line

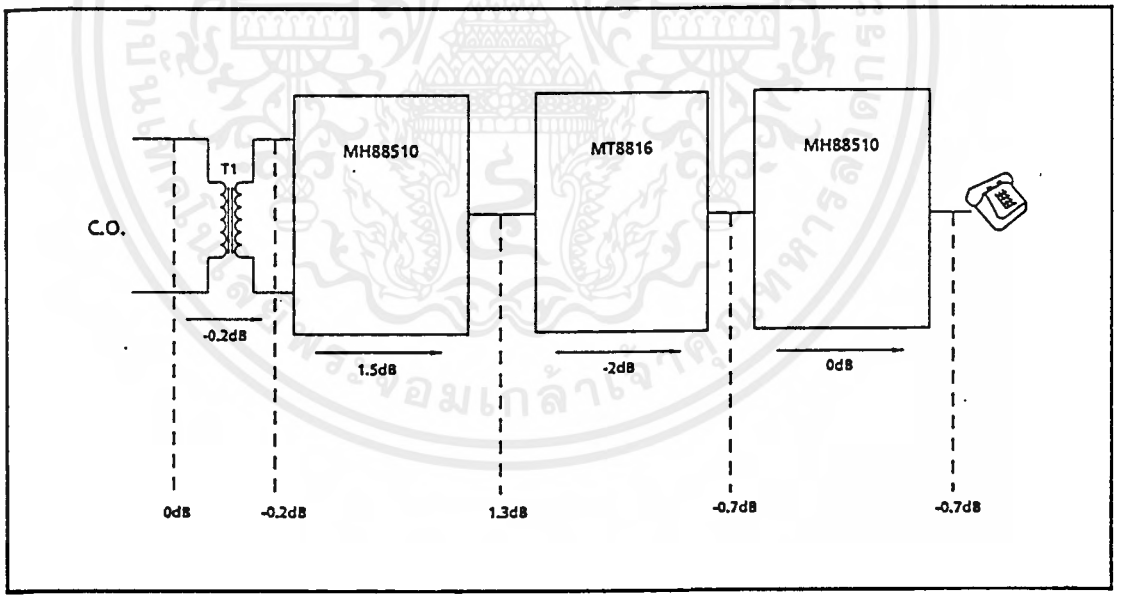


Figure 8b - Line to Line

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

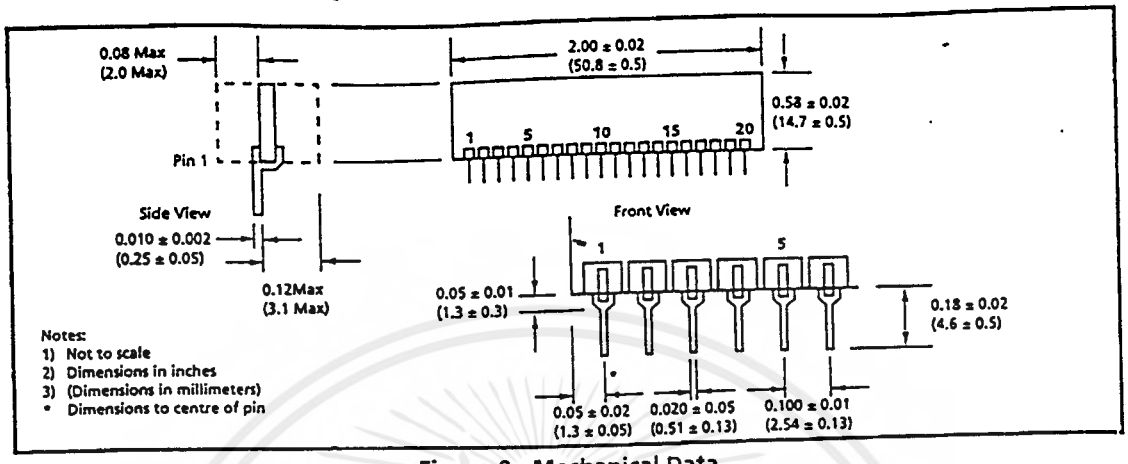


Figure 9 - Mechanical Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MITEL

ISO-CMOS MT8816
8 x 16 Analog Switch Array

9161-002-101-NA ISSUE 2 November 1988

Features

- Internal control latches and address decoder
- Short set-up and hold times
- Wide operating voltage: 4.5V to 13.2V
- 12Vpp analog signal capability
- R_{ON} 65Ω max. @ V_{DD} = 12V, 25°C
- $\Delta R_{ON} \leq 10\Omega$ @ V_{DD} = 12V, 25°C
- Full CMOS switch for low distortion
- Minimum feedthrough and crosstalk
- Separate analog and digital reference supplies
- Low power consumption ISO-CMOS technology

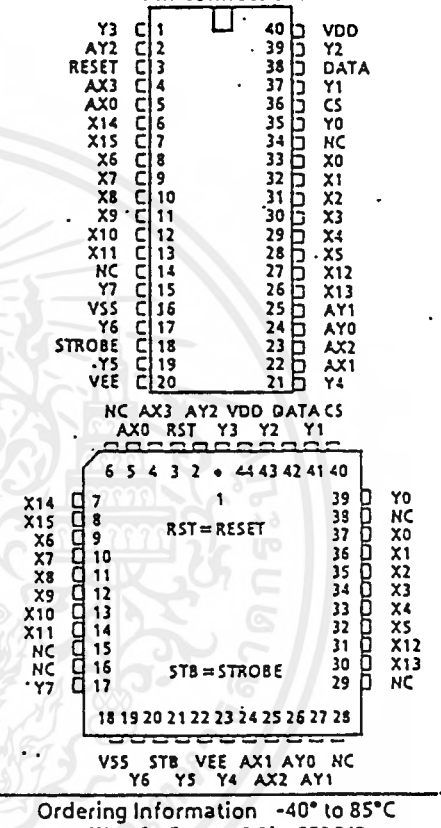
Applications

- Key systems
- PBX systems
- Mobile radio
- Test equipment /instrumentation
- Analog/digital multiplexers
- Audio/Video switching

Description

The Mitel MT8816 is fabricated in MITEL's ISO-CMOS technology providing low-power dissipation and high reliability. The device contains a 8X16 array of crosspoint switches along with a 7 to 128 line decoder and latch circuits. Any one of the 128 switches can be addressed by selecting the appropriate seven address bits. The selected switch can be turned on or off by applying a logical one or zero to the DATA input. V_{SS} is the ground reference of the digital inputs. The range of the analog signal is from V_{DD} to V_{EE}. Chip Select (CS) allows the crosspoint array to be cascaded for matrix expansion.

Pin Connections



Ordering Information -40° to 85°C

MT8816AC	40 Pin CERDIP
MT8816AE	40 Pin PLASTIC DIP
MT8816AP	44 Pin PLCC

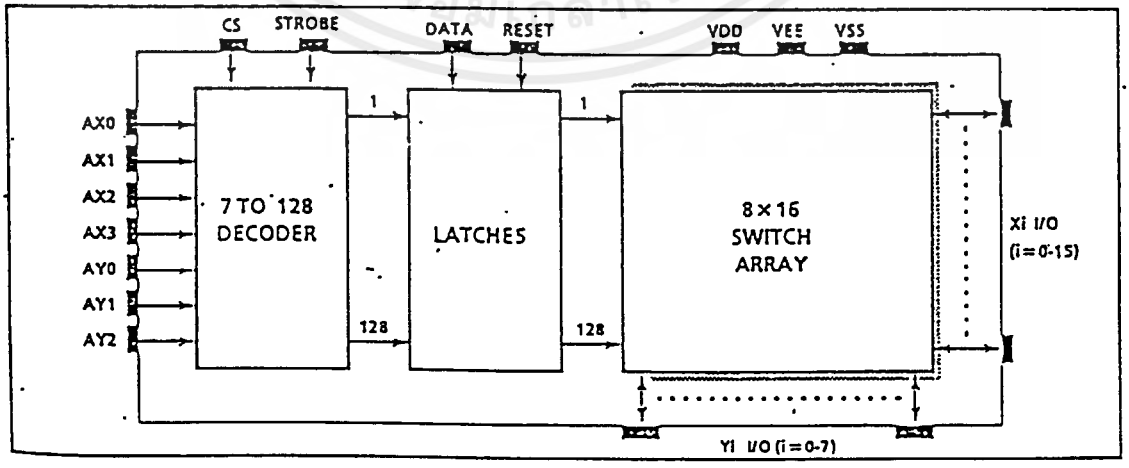


Figure 1- Functional Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MT8816 ISO-CMOS

Absolute Maximum Ratings* - Voltages are with respect to V_{EE} unless otherwise stated.

	Parameter		Symbol	Min	Max	Units
1	Supply Voltage		V_{DD} V_{SS}	−0.3 −0.3	15.0 V_{DD}+0.3	V V
2	Analog Input Voltage		V_{INA}	−0.3	V_{DD}+0.3	V
3	Digital Input Voltage		V_{IN}	V_{SS}−0.3	V_{DD}+0.3	V
4	Current on any I/O Pin		I		±15	mA
5	Storage Temperature		T_S	−65	+150	°C
6	Package Power Dissipation	PLASTIC DIP CERDIP	P_D P_D		0.6 1.0	W W

*Exceeding these values may cause permanent damage. Functional operation under these conditions is not implied.

Recommended Operating Conditions - Voltages are with respect to V_{EE} unless otherwise stated.

	Characteristics	Sym	Min	Typ	Max	Units	Test Conditions
1	Operating Temperature	T_O	−40	25	85	°C	
2	Supply Voltage	V_{DD} V_{SS}	4.5 V_{EE}		13.2 V_{DD}−4.5	V V	
3	Analog Input Voltage	V_{INA}	V_{EE}		V_{DD}	V	
4	Digital Input Voltage	V_{IN}	V_{SS}		V_{DD}	V	

DC Electrical Characteristics† - Voltages are with respect to $V_{EE}=V_{SS}=0V$, V_{DD} = 12V unless otherwise stated.

	Characteristics	Sym	Min	Typ‡	Max	Units	Test Conditions
1	Quiescent Supply Current	I_{DD}		1	100	µA	All digital inputs at $V_{IN}=V_{SS}$ or V_{DD}
				0.4	1.5	mA	All digital inputs at $V_{IN}=2.4+V_{SS}$; $V_{SS}=7.0V$
				5	15	mA	All digital inputs at $V_{IN}=3.4V$
2	Off-state Leakage Current (See G.9 in Appendix)	I_{OFF}		±1	±500	nA	$\|V_{Xi}-V_{Yj}\|=V_{DD}-V_{EE}$ See Appendix, Fig. A.1
3	Input Logic "0" level	V_{IL}			0.8+V_{SS}	V	$V_{SS}=7.5V$; $V_{EE}=0V$
4	Input Logic "1" level	V_{IH}	2.0+V_{SS}			V	$V_{SS}=6.5V$; $V_{EE}=0V$
5	Input Logic "1" level	V_{IH}	3.3			V	
6	Input Leakage (digital pins)	I_{LEAK}		0.1	10	µA	All digital inputs at $V_{IN}=V_{SS}$ or V_{DD}

† DC Electrical Characteristics are over recommended temperature range.
‡ Typical figures are at 25°C and are for design aid only; not guaranteed and not subject to production testing.

DC Electrical Characteristics- Switch Resistance - V_{DC} is the external DC offset applied at the analog I/O pins.

	Characteristics	Sym	25°C		70°C		85°C		Units	Test Conditions
			Typ	Max	Typ	Max	Typ	Max		
1	On-state Resistance $V_{DD}=12V$ $V_{DD}=10V$ $V_{DD}=5V$ (See G.1, G.2, G.3 in Appendix)	R_{ON}	45 55 120	65 75 185		75 85 215		80 90 225	Ω Ω Ω	$V_{SS}=V_{EE}=0V$, $V_{DC}=V_{DD}/2$, $\|V_{Xi}-V_{Yj}\|=0.4V$ See Appendix, Fig. A.2
2	Difference in on-state resistance between two switches (See G.4 in Appendix)	ΔR_{ON}	5	10		10		10	Ω	$V_{DD}=12V$, $V_{SS}=V_{EE}=0$, $V_{DC}=V_{DD}/2$, $\|V_{Xi}-V_{Yj}\|=0.4V$ See Appendix, Fig. A.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Electrical Characteristics[1] - Crosspoint Performance - Voltages are with respect to V_{DD} = 5V, V_{SS} = 0V, V_{EE} = −7V, unless otherwise stated.

	Characteristics	Sym	Min	Typ[2]	Max	Units	Test Conditions
1	Switch I/O Capacitance	C_S		20		pF	f = 1 MHz
2	Feedthrough Capacitance	C_F		0.2		pF	f = 1 MHz
3	Frequency Response Channel "ON" 20LOG(V_{OUT}/V_{Xi}) = −3dB	F_{3dB}		45		MHz	Switch is "ON"; V_{INA} = 2Vpp sine wave; R_L = 1kΩ See Appendix, Fig. A.3
4	Total Harmonic Distortion (See G.5, G.6 in Appendix)	THD		0.01		%	Switch is "ON"; V_{INA} = 2Vpp sine wave f = 1kHz; R_L = 1kΩ
5	Feedthrough Channel "OFF" Feed. = 20LOG (V_{OUT}/V_{Xi}) (See G.8 in Appendix)	FDT		−95		dB	All Switches "OFF"; V_{INA} = 2Vpp sine wave f = 1kHz; R_L = 1kΩ. See Appendix, Fig. A.4
6	Crosstalk between any two channels for switches Xi−Yi and Xj−Yj. Xtalk = 20LOG (V_{Yj}/V_{Xi}). (See G.7 in Appendix).	X_{talk}		−45		dB	V_{INA} = 2Vpp sine wave f = 10MHz; R_L = 75Ω.
				−90		dB	V_{INA} = 2Vpp sine wave f = 10kHz; R_L = 600Ω.
				−85		dB	V_{INA} = 2Vpp sine wave f = 10kHz; R_L = 1kΩ.
				−80		dB	V_{INA} = 2Vpp sine wave f = 1kHz; R_L = 10kΩ. Refer to Appendix, Fig. A.5 for test circuit.
7	Propagation delay through switch	t_{PS}			30	ns	R_L = 1kΩ; C_L = 50pF

[1] Timing is over recommended temperature range. See Fig. 2 for control and I/O timing details.
[2] Typical figures are at 25°C and are for design aid only; not guaranteed and not subject to production testing.
Crosstalk measurements are for Plastic DIPS only, crosstalk values for PLCC packages are approximately 5dB better.

AC Electrical Characteristics[1] - Control and I/O Timings - Voltages are with respect to V_{DD} = 5V, V_{SS} = 0V, V_{EE} = −7V, unless otherwise stated.

	Characteristics	Sym	Min	Typ[2]	Max	Units	Test Conditions
1	Control Input crosstalk to switch (for CS, DATA, STROBE, Address)	CX_{talk}		30		mVpp	V_{IN} = 3V squarewave; R_{IN} = 1kΩ, R_L = 10kΩ. See Appendix, Fig. A.6
2	Digital Input Capacitance	C_{DI}		10		pF	f = 1MHz
3	Switching Frequency	F_O			20	MHz	
4	Setup Time DATA to STROBE	t_{DS}	10			ns	R_L = 1kΩ, C_L = 50pF [3]
5	Hold Time DATA to STROBE	t_{DH}	10			ns	R_L = 1kΩ, C_L = 50pF [3]
6	Setup Time Address to STROBE	t_{AS}	10			ns	R_L = 1kΩ, C_L = 50pF [3]
7	Hold Time Address to STROBE	t_{AH}	10			ns	R_L = 1kΩ, C_L = 50pF [3]
8	Setup Time CS to STROBE	t_{CSS}	10			ns	R_L = 1kΩ, C_L = 50pF [3]
9	Hold Time CS to STROBE	t_{CSH}	10			ns	R_L = 1kΩ, C_L = 50pF [3]
10	STROBE Pulse Width	t_{SPW}	20			ns	R_L = 1kΩ, C_L = 50pF [3]
11	RESET Pulse Width	t_{RPW}	40			ns	R_L = 1kΩ, C_L = 50pF [3]
12	STROBE to Switch Status Delay	t_S		40	100	ns	R_L = 1kΩ, C_L = 50 pF [3]
13	DATA to Switch Status Delay	t_D		50	100	ns	R_L = 1kΩ, C_L = 50 pF [3]
14	RESET to Switch Status Delay	t_R		35	100	ns	R_L = 1kΩ, C_L = 50 pF [3]

[1] Timing is over recommended temperature range. See Fig. 2 for control and I/O timing details.
Digital input rise time (tr) and fall time (tf) = 5ns.
[2] Typical figures are at 25°C and are for design aid only; not guaranteed and not subject to production testing.
[3] Refer to Appendix, Fig. A.7 for test circuit.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

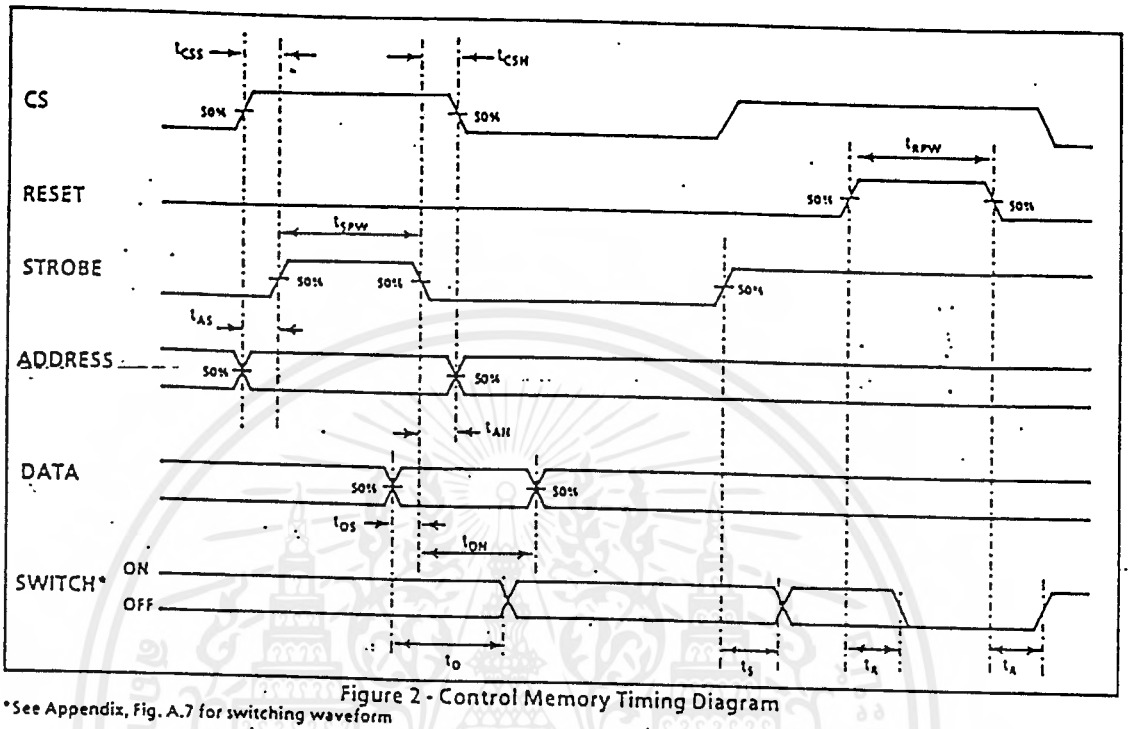


Figure 2 - Control Memory Timing Diagram

*See Appendix, Fig. A.7 for switching waveform

AX0	AX1	AX2	AX3	AY0	AY1	AY2	Connection*
0	0	0	0	0	0	0	X0 - Y0
1	0	0	0	0	0	0	X1 - Y0
0	1	0	0	0	0	0	X2 - Y0
1	1	0	0	0	0	0	X3 - Y0
0	0	1	0	0	0	0	X4 - Y0
1	0	1	0	0	0	0	X5 - Y0
0	1	1	0	0	0	0	X12 - Y0
1	1	1	0	0	0	0	X13 - Y0
0	0	0	1	0	0	0	X6 - Y0
1	0	0	1	0	0	0	X7 - Y0
0	1	0	1	0	0	0	X8 - Y0
1	1	0	1	0	0	0	X9 - Y0
0	0	1	1	0	0	0	X10 - Y0
1	0	1	1	0	0	0	X11 - Y0
0	1	1	1	0	0	0	X14 - Y0
1	1	1	1	0	0	0	X15 - Y0
0	0	0	0	1	0	0	X0 - Y1
1	0	0	0	1	0	0	X15 - Y1
0	0	0	0	0	1	0	X0 - Y2
1	0	0	0	0	1	0	X15 - Y2
0	0	0	0	1	1	0	X0 - Y3
1	0	0	0	1	1	0	X15 - Y3
0	0	0	0	0	0	1	X0 - Y4
1	0	0	0	0	0	1	X15 - Y4
0	0	0	0	1	0	1	X0 - Y5
1	0	0	0	1	0	1	X15 - Y5
0	0	0	0	0	1	1	X0 - Y6
1	0	0	0	0	1	1	X15 - Y6
0	0	0	0	1	1	1	X0 - Y7
1	0	0	0	1	1	1	X15 - Y7

Figure 3 - Address Decode Truth Table

*Switch connections are not in ascending order

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description

Pin # *	Name	Description
1	Y3	Y3 Analog (Input/Output): this is connected to the Y3 column of the switch array.
2	AY2	Y2 Address Line (Input).
3	RESET	Master RESET (Input): this is used to turn off all switches regardless of the condition of CS. Active High.
4,5	AX3,AX0	X3 and X0 Address Lines (Inputs).
6,7	X14, X15	X14 and X15 Analog (Inputs/Outputs): these are connected to the X14 and X15 rows of the switch array.
8-13	X6-X11	X6-X11 Analog (Inputs/Outputs): these are connected to the X6-X11 rows of the switch array.
14	NC	No Connection
15	Y7	Y7 Analog (Input/Output): this is connected to the Y7 column of the switch array.
16	VSS	Digital Ground Reference.
17	Y6	Y6 Analog (Input/Output): this is connected to the Y6 column of the switch array.
18	STROBE	STROBE (Input): enables function selected by address and data. Address must be stable before STROBE goes high and DATA must be stable on the falling edge of the STROBE. Active High.
19	Y5	Y5 Analog (Input/Output): this is connected to the Y5 column of the switch array.
20	VEE	Negative Power Supply.
21	Y4	Y4 Analog (Input/Output): this is connected to the Y4 column of the switch array.
22, 23	AX1,AX2	X1 and X2 Address Lines (Inputs).
24, 25.	AY0,AY1	Y0 and Y1 Address Lines (Inputs).
26, 27	X13, X12	X13 and X12 Analog (Inputs/Outputs): these are connected to the X13 and X12 rows of the switch array.
28 - 33	X5-X0	X5-X0 Analog (Inputs/Outputs): these are connected to the X5-X0 rows of the switch array.
34	NC	No Connection.
35	Y0	Y0 Analog (Input/Output): this is connected to the Y0 column of the switch array.
36	CS	Chip Select (Input): this is used to select the device. Active High.
37	Y1	Y1 Analog (Input/Output): this is connected to the Y1 column of the switch array.
38	DATA	DATA (Input): a logic high input will turn on the selected switch and a logic low will turn off the selected switch. Active High.
39	Y2	Y2 Analog (Input/Output): this is connected to the Y2 column of the switch array.
40	VDD	Positive Power Supply.

* Plastic DIP and CERDIP only

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description

The MT8816 is an analog switch matrix with an array size of 8×16. The switch array is arranged such that there are 8 columns by 16 rows. The columns are referred to as the Y inputs/outputs and the rows are the X inputs/outputs. The crosspoint analog switch array will interconnect any X I/O with any Y I/O when turned on and provide a high degree of isolation when turned off. The control memory consists of a 128 bit write only RAM in which the bits are selected by the address inputs (AY0-AY2, AX0-AX3). Data is presented to the memory on the DATA input. Data is asynchronously written into memory whenever both the CS (Chip Select) and STROBE inputs are high and are latched on the falling edge of STROBE. A logical "1" written into a memory cell turns the corresponding crosspoint switch on and a logical "0" turns the crosspoint off. Only the crosspoint switches corresponding to the addressed memory location are altered when data is written into memory. The remaining switches retain their previous states. Any combination of X and Y inputs/outputs can be interconnected by establishing appropriate patterns in the control memory. A logical "1" on the RESET input will asynchronously return all memory locations to logical "0" turning off all crosspoint switches regardless of whether CS is high or low. Two voltage reference pins (V_{SS} and V_{EE}) are provided for the MT8816 to enable switching of negative analog signals. The range for digital signals is from V_{DD} to V_{SS} while the range for analog signals is from V_{DD} to V_{EE}. V_{SS} and V_{EE} pins can be tied together if a single voltage reference is needed.

Address Decode

The seven address inputs along with the STROBE and CS (Chip Select) are logically ANDed to form an enable signal for the resettable transparent latches. The DATA input is buffered and is used as the input to all latches. To write to a location, RESET must be low and CS must go high while the address and data are set up. Then the STROBE input is set high and then low causing the data to be latched. The data can be changed while STROBE is high, however, the corresponding switch will turn on and off in accordance with the DATA input. DATA must be stable on the falling edge of STROBE in order for correct data to be written to the latch.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MITEL

ISO²-CMOS MT8870B/MT8870B-1 Integrated DTMF Receiver

9161-002-051-NA ISSUE 2 December 1987

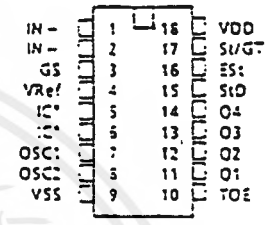
Features

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

Applications

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

Pin Connections



* Connect to VSS

Ordering information -40°C to +85°C

MT8870BE/MT8870BE-1 Plastic DIP
MT8870BC/MT8870BC-1 Ceramic

Description

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

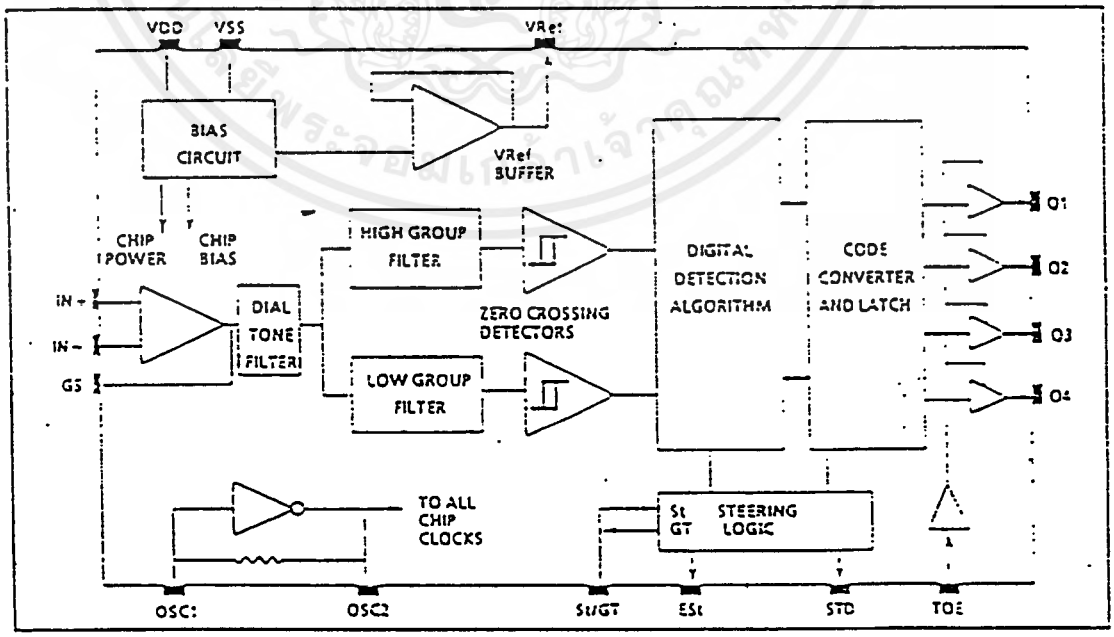


Figure 1 - Functional Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings[1]

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

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

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

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

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

DC Electrical Characteristics - V_{DD} = 5.0V ± 5%, V_{SS} = 0V. Voltages are with respect to ground (V_{SS}) unless otherwise stated.

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

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

Gain Setting Amplifier

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

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

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

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

		Characteristics	Sym	Min	Typ	Max	Units	Notes
1	SIGNAL CONDITION	Valid input signal levels (each tone of composite signal)		-29			dBm	1,2,3,5,6,9
				27.5			mV_{RMS}	1,2,3,5,6,9
						+1	dBm	1,2,3,5,6,9
						869	mV_{RMS}	1,2,3,5,6,9
2		Positive twist accept			10		dB	2,3,6,9
3		Negative twist accept			10		dB	2,3,6,9
4		Freq. deviation accept		± 1.5% ± 2Hz			Nom.	2,3,5,9
5		Freq. deviation reject		± 3.5%			Nom.	2,3,5,9
6		Third tone tolerance			+16		dB	2,3,4,5,9
7		Noise tolerance			-12		dB	2,3,4,5,7,9,10
8		Dial tone tolerance			-22		dB	2,3,4,5,8,9,11

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

NOTES

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

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

NOTES

1. dBm = decibels above or below a reference power of 1 mW into a 600 ohm load.
2. Digit sequence consists of all DTMF tones
3. Tone duration = 40 ms, tone pause = 40 ms.
4. Signal condition consists of nominal DTMF frequencies
5. Both tones in composite signal have an equal amplitude
6. Tone pair is deviated by ± 1.5% ± 2Hz.
7. Bandwidth limited (3 kHz) Gaussian noise.
8. The precise dial tone frequencies are (350 Hz and 440 Hz) ± 2 %.
9. For an error rate of better than 1 in 10,000.
10. Referenced to lowest level frequency component in DTMF signal.
11. Referenced to the minimum valid accept level.
12. For guard time calculation purposes.
13. Referenced to Fig. 10 inout DTMF Tone Level at -25 dBm (-23 dBm at GS Pin) interference frequency range between 480-3400 Hz.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

[1] V_{DD} = 5.0V, V_{SS} = 0V, T_A = 25°C and f_C = 3.579545 MHz, using test circuit shown in Figure 2

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

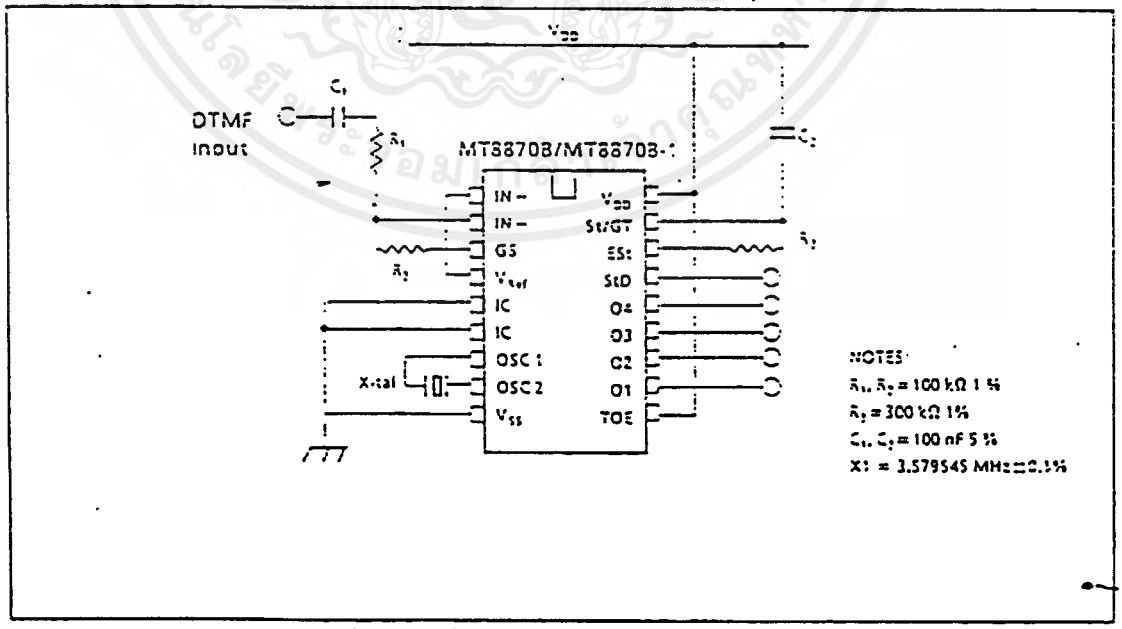


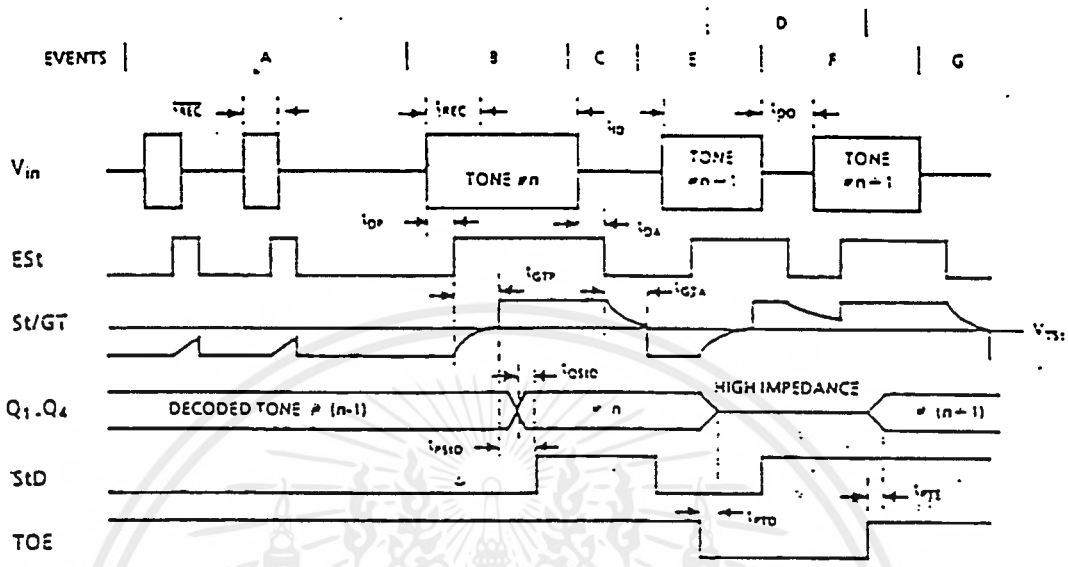
Figure 2 - Single-Ended Input Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Description

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



EXPLANATION OF EVENTS

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

EXPLANATION OF SYMBOLS

V_{in} DTMF COMPOSITE INPUT SIGNAL.
ESt EARLY STEERING OUTPUT. INDICATES DETECTION OF VALID TONE FREQUENCIES.
St/GT STEERING INPUT/GUARD TIME OUTPUT. DRIVES EXTERNAL RC TIMING CIRCUIT.
Q_1-Q_4 4-BIT DECODED TONE OUTPUT.
StD DELAYED STEERING OUTPUT. INDICATES THAT VALID FREQUENCIES HAVE BEEN PRESENT/ABSENT FOR THE REQUIRED GUARD TIME THUS CONSTITUTING A VALID SIGNAL.
TOE TONE OUTPUT ENABLE (INPUT). A LOW LEVEL SHIFTS Q_1-Q_4 TO ITS HIGH IMPEDANCE STATE.

t_{REZ} MAXIMUM DTMF SIGNAL DURATION NOT DETECTED AS VALID.
t_{REC} MINIMUM DTMF SIGNAL DURATION REQUIRED FOR VALID RECOGNITION
t_{ID} MINIMUM TIME BETWEEN VALID DTMF SIGNALS.
t_{DO} MAXIMUM ALLOWABLE DROPOUT DURING VALID DTMF SIGNAL.
t_{DP} TIME TO DETECT THE PRESENCE OF VALID DTMF SIGNALS.
t_{DA} TIME TO DETECT THE ABSENCE OF VALID DTMF SIGNALS.
t_{GTP} GUARD TIME, TONE PRESENT.
t_{GTA} GUARD TIME, TONE ABSENT.

Figure 3- Timing Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description

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

Filter Section

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

Decoder Section

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

Steering Circuit

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

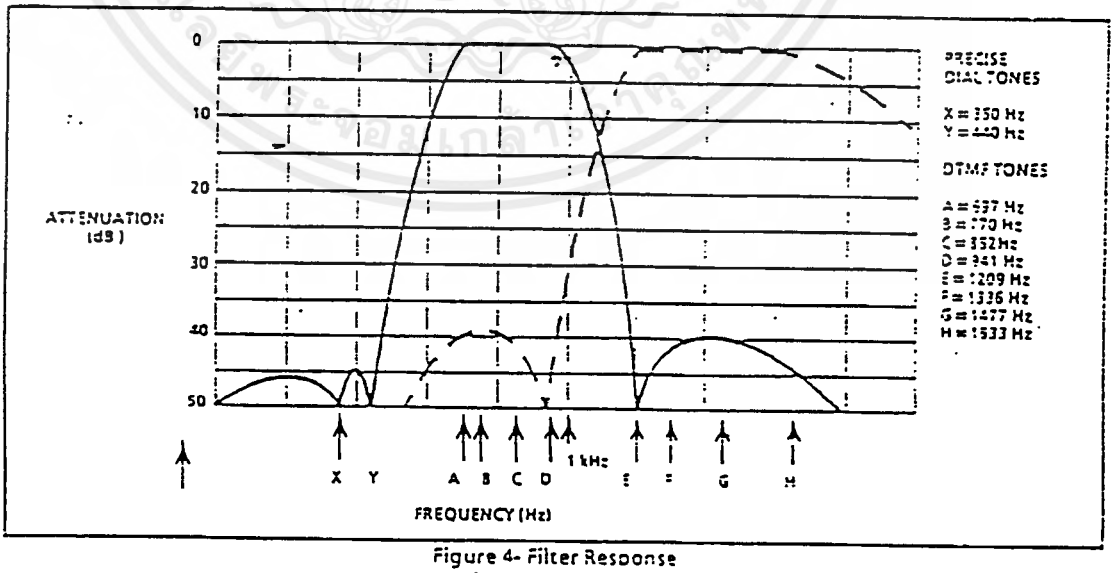


Figure 4- Filter Response

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

Table 1. Functional Decode Table

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

Guard Time Adjustment

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

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

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

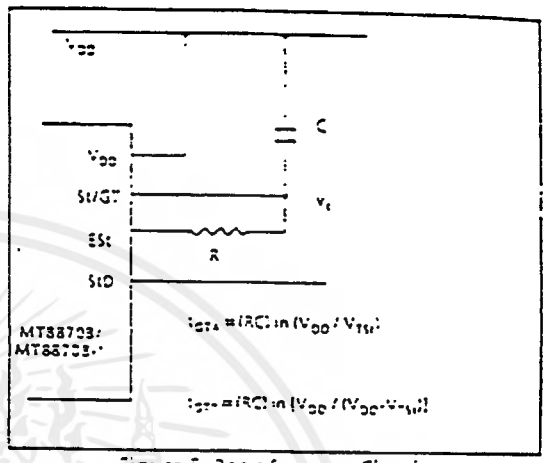


Figure 5- Basic Steering Circuit

Different steering arrangements may be used to select independently the guard times for tone present (t_{GTP}) and tone absent (t_{GTA}). This may be necessary to meet system specifications which place both accept and reject limits on both tone duration and interdigital pause. Guard time adjustment also allows the designer to tailor system parameters such as talk off and noise immunity. Increasing t_{REC} improves talk-off performance since it reduces the probability that tones simulated by speech will maintain signal condition long enough to be registered. Alternatively, a relatively short t_{REC} with a long t_{DO} would be appropriate for extremely noisy environments where fast acquisition time and immunity to tone dropouts are required. Design information for guard time adjustment is shown in Figure 8.

Differential Input Configuration

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

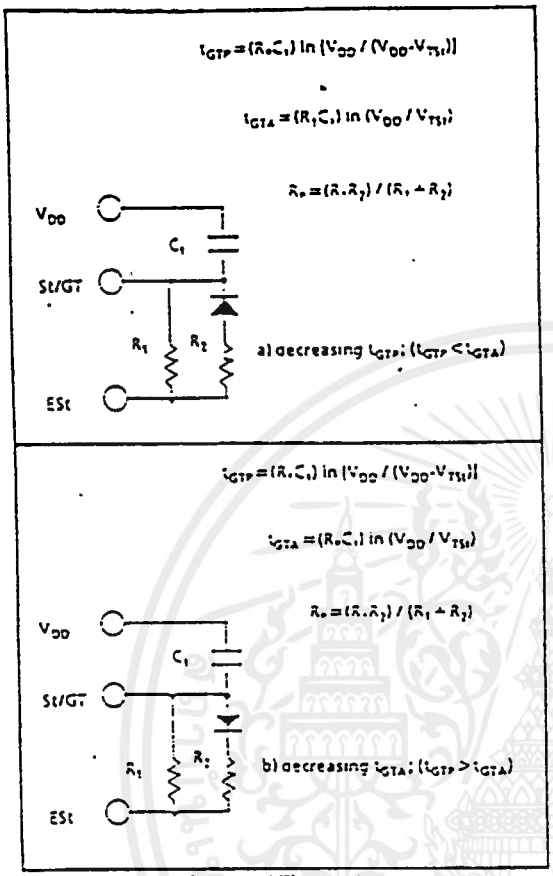


Figure 6- Guard Time Adjustment

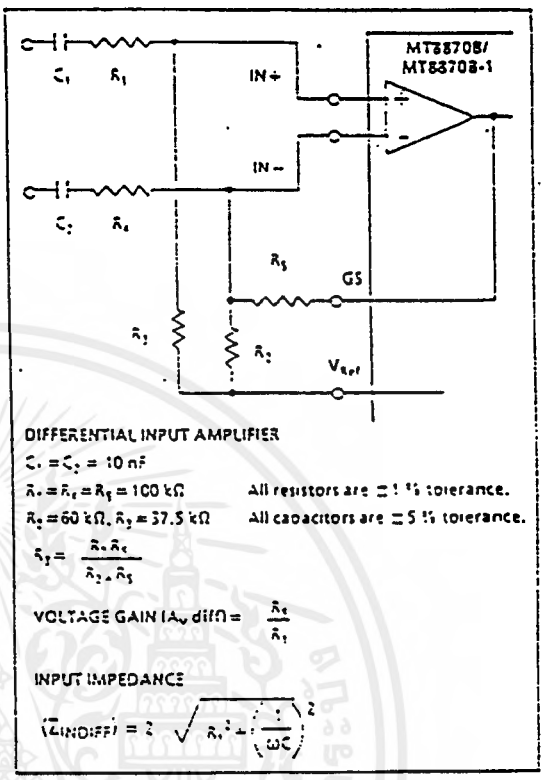


Figure 7- Differential Input Configuration

Crystal Oscillator

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

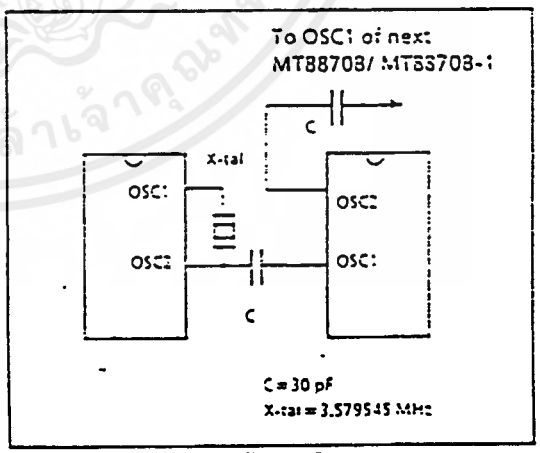


Figure 8- Oscillator Connection

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION

RECEIVER SYSTEM FOR BRITISH TELECOM SPEC POR 1151

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

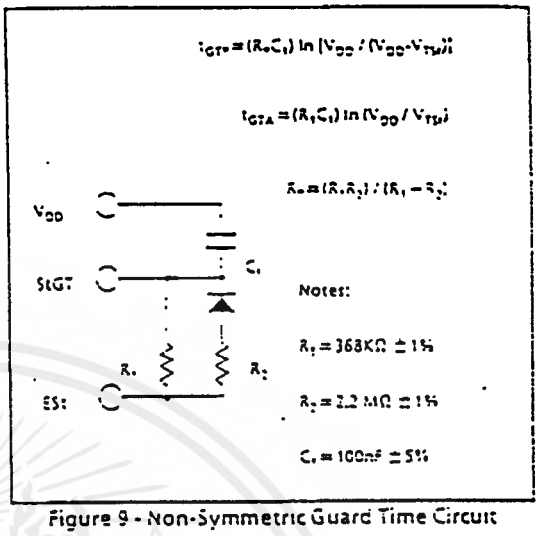


Figure 9 - Non-Symmetric Guard Time Circuit

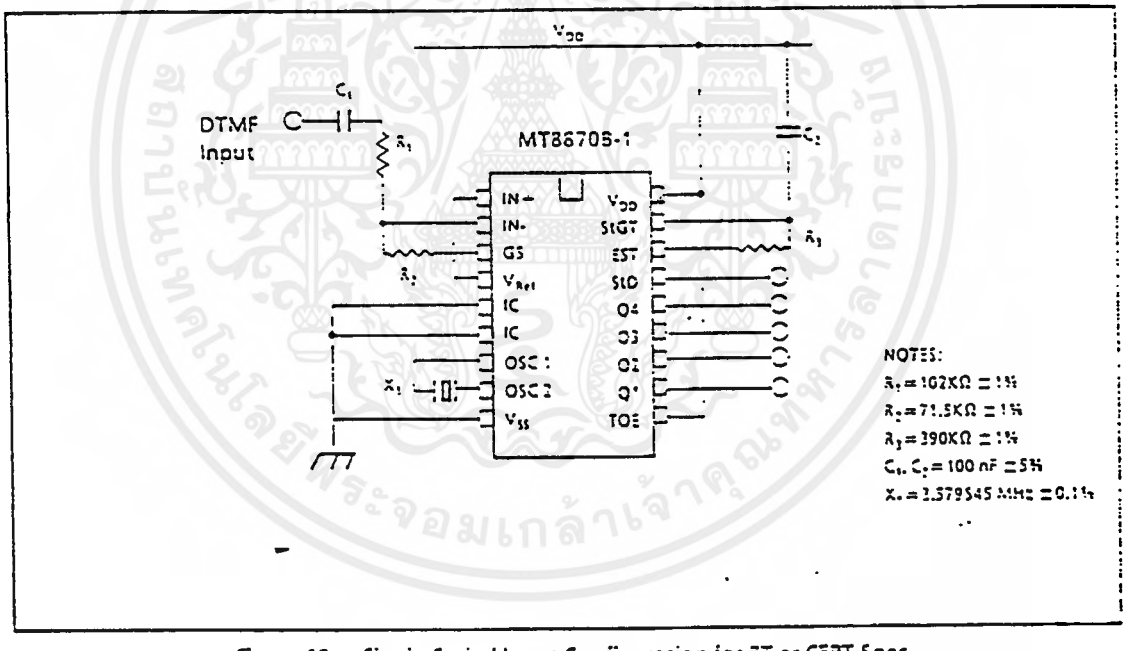


Figure 10 - Single-Ended Input Configuration for BT or CEPT Spec

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRELIMINARY DATA SHEET

INFORMATION STORAGE DEVICES

ISD1200/1400 Series

Single-Chip Voice Record/Playback Devices 10-, 12-, 16-, and 20-Second Durations

GENERAL DESCRIPTION

Information Storage Devices' ISD1200/1400 ChipCorder™ Series provides high-quality, single-chip record/playback solutions to short duration messaging applications. The CMOS devices include an on-chip oscillator, microphone pre-amplifier, automatic gain control, antialiasing filter, smoothing filter, and speaker amplifier. A minimum record/playback subsystem can be configured with a microphone, a speaker, several passives, two push-buttons, and a power source.

Recordings are stored in nonvolatile memory cells, providing zero-power message storage. This unique solution is made possible through ISD's patented Direct Analog Storage Technology (DAST®), whereby voice and audio signals are stored directly, in their natural analog form, into EEPROM memory. Direct analog storage allows natural voice reproduction in a single-chip solid-state solution.

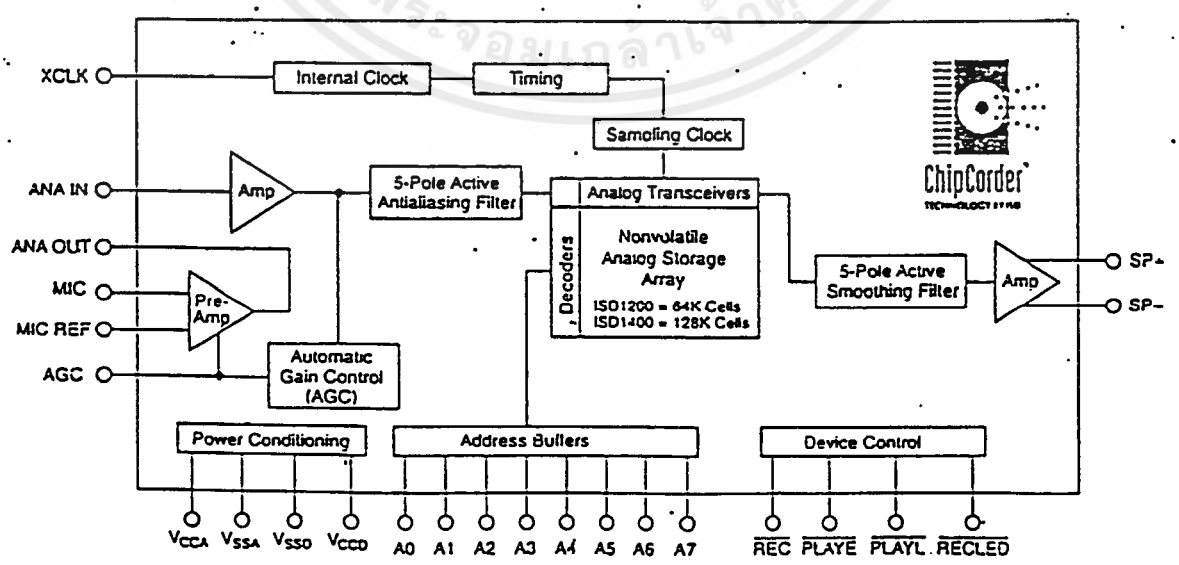
ISD1200/1400 SERIES SUMMARY

Part Number	Minimum Duration (Seconds)	Maximum Input Sample Rate (KHz)	Upper Pass Band (KHz)
ISD1210	10	6.4	2.7
ISD1212	12	5.3	2.3
ISD1416	16	8.0	3.4
ISD1420	20	6.4	2.7

FEATURES

- Easy-to-use single-chip voice record/playback solution
  - No external ICs required
  - Minimizes external components
- High-quality, natural voice/audio reproduction
- Push-button interface
  - Playback can be edge- or level-activated
- Zero-power message storage
  - Eliminates battery backup circuits
- 100-year message retention (typical)
- 100 K record cycles (typical)
- On-chip clock source
- No programmer or development system needed
- Fully addressable to handle multiple messages
- Automatic power-down mode
  - Enters standby mode immediately following a record or playback cycle
  - Standby current 0.5 uA (typical)
- Single power supply
- Available in DIP, SOIC, and die form for Chip-on-Board (COB) Module assembly

ISD1200/1400 SERIES BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DETAILED DESCRIPTION

Basic Operation

The ISD1200/1400 ChipCorder Series devices are controlled by a single signal, $\overline{REC}$, and either of two push-button control playback signals, $\overline{PLAYE}$ (edge-activated playback), and $\overline{PLAYL}$ (level-activated playback). The ISD1200/1400 parts are configured for simplicity of design in a single-message application. Using the address lines will allow multiple message applications. Device operation is explained on page 4.

Speech Quality

ISD's patented DAST technology provides natural record and playback. The input voice signals are stored directly in nonvolatile EEPROM cells, and reproduced without the synthetic effect often heard with digital solid-state speech solutions. A complete sample is stored in a single cell, minimizing the memory necessary to store a recording of a given duration.

Automatic Power-Down Mode

At the end of a playback or record cycle, the ISD1200/1400 Series devices automatically return to a low-power standby mode, consuming typically 0.5 µA. During a playback cycle, the device powers down automatically at the end of the message. During a record cycle, the device powers down immediately after $\overline{REC}$ is released HIGH.

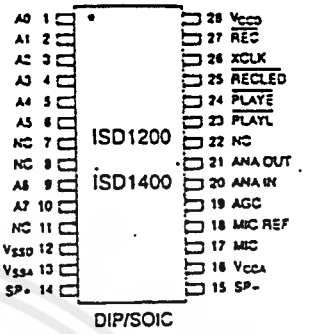
Addressing (optional)

In addition to providing simple message playback, the ISD1200/1400 Series provides a full addressing capability.

The ISD1200 Series storage array has 80 distinct addressable segments, while the ISD1400 Series storage array has 160 distinct addressable segments, providing the following resolutions.

Part Number	Resolution
ISD1210	125 ms
ISD1212	150 ms
ISD1416	100 ms
ISD1420	125 ms

ISD1200/1400 SERIES PINOUTS



DIP/SOIC

Note: NC means Must Not Connect

PIN DESCRIPTIONS

Note: The $\overline{REC}$, $\overline{PLAYL}$, and $\overline{PLAYE}$ signals are all debounced for 50 ms, on the rising edge to prevent a false retriggering from a push-button switch.

Record ($\overline{REC}$)

The $\overline{REC}$ input is an active-LOW record signal. The device records whenever $\overline{REC}$ is LOW. This signal must remain LOW for the duration of the recording. $\overline{REC}$ takes precedence over either playback ($\overline{PLAYE}$ or $\overline{PLAYL}$) signal. If $\overline{REC}$ is pulled LOW during a playback cycle, the playback immediately ceases and recording begins.

A record cycle is completed when $\overline{REC}$ is pulled HIGH. An end-of-message marker is internally recorded, enabling a subsequent playback cycle to terminate appropriately. The device automatically powers down to standby mode when $\overline{REC}$ goes HIGH.

Playback, Edge-Activated ($\overline{PLAYE}$)

When a LOW-going transition is detected on this input signal, a playback cycle begins. Playback continues until an end-of-message marker is encountered or the end of the memory space is reached. Upon completion of the playback cycle, the device automatically powers down into standby mode. Taking $\overline{PLAYE}$ HIGH during a playback cycle will not terminate the current cycle.

Playback, Level-Activated ($\overline{PLAYL}$)

When this input signal transitions from HIGH to LOW, a playback cycle is initiated. Playback continues until $\overline{PLAYL}$ is pulled HIGH, an end-of-message marker is detected, or the end of the device space is reached. The device automatically powers down to standby mode upon completion of the playback cycle.

Note: In playback, if either $\overline{PLAYE}$ or $\overline{PLAYL}$ is held LOW during $\overline{EOM}$ or OVERFLOW, the device will still enter

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

standby and the internal oscillator and timing generator will stop. However, the rising edge of $\overline{\text{PLAYE}}$ and $\overline{\text{PLAYL}}$ will no longer be debounced and any subsequent falling edge present on the input pins will initiate another playback.

Record LED Output ($\overline{\text{RECLED}}$)

The output $\overline{\text{RECLED}}$ is LOW during a record cycle. It can be used to drive an LED to provide feedback that a record cycle is in progress. In addition, $\overline{\text{RECLED}}$ pulses LOW momentarily when an end-of-message marker is encountered in a playback cycle.

Microphone Input (MIC)

The microphone is usually AC-coupled to this pin via a series capacitor. The user-selectable value of the input series capacitor (together with the 10K ohm resistance internal to the chip) determines the low-frequency cutoff for the ISD1200/1400 Series passband.

Microphone Reference (MIC REF)

When MIC REF is AC coupled to the microphone ground, the recorded noise level is significantly reduced. Ground noise is referenced to the preamplifier. If this pin is not used, it must NOT be connected to any signal or voltage. It must float.

Analog Output (ANA OUT)

The microphone signal is amplified and is output to the ANA OUT pin. The voltage gain of the preamp is determined by the voltage level at the Automatic Gain Control (AGC) pin. The preamplifier has a maximum gain of about 24 dB for small input signal levels.

Analog Input (ANA IN)

The external capacitor connects ANA IN to the ANA OUT pin. The value of the external capacitor, together with the 3 KΩ input impedance at ANA IN, can be chosen to give additional cutoff at the low-frequency end of the voice passband. The ANA IN pin may also be used to input alternative sources of analog signals (instead of the microphone signal) through a coupling capacitor.

Automatic Gain Control (AGC)

The purpose of the AGC is to dynamically adjust the preamplifier gain, and therefore extend the range of input signals which can be applied to the microphone input without causing distortion. The AGC can considerably extend the range of recordable sound from whispers to loud voices. To use the AGC feature, an external resistor and capacitor should be connected in parallel between the AGC pin and Ground. Recommended values are 470 KΩ and 4.7 µF. The "attack" time of the gain control is determined by the source resistance (5 KΩ) and the external capacitor. The "release" time is determined by the external resistor and capacitor. For AGC voltages of 1.5 Volts and below, the preamplifier is at its maximum gain of 24 dB. Reduction in preamplifier gain occurs for voltages of approximately 1.8 Volts. If the AGC function is not desired, the AGC pin can be tied to Ground and the preamplifier gain will be held at its highest level of approximately 24 dB.

If operating at voltages above 5.5 V, insert a 5.1 KΩ resistor in series with the capacitor from pin 20 to pin 21 to minimize distortion.

Speaker Outputs (SP+, SP–)

The SP+ and SP– pins provide direct drive for loudspeakers with impedances as low as 16 ohms. A single output may be used, but, for direct-drive loudspeakers, the two opposite-polarity outputs provide an improvement in output power of up to four times over a single-ended connection. Furthermore, when SP+ and SP– are used, a speaker-coupling capacitor is not required. A single-ended connection will require an AC-coupling capacitor between the SP pin and the speaker. The speaker outputs are in a high-impedance state during a record cycle, and held at V_{SSA} during Power Down.

Optional External Clock (XCLK)

This signal is normally tied to ground in applications circuits. If, however, greater timing precision is desired, (internal clock has ±2.5 % tolerance over temperature and voltage range), the chip can be externally clocked through this pin. If the XCLK is not used, this input should be connected to ground.

V_{CCA} and V_{CCD}

Analog and digital circuits internal to the ISD1200/1400 Series use separate power buses to minimize noise on the chip. These power buses are brought out to separate pins on the package and should be tied together as close to the supply as possible. It is important that the power supply be decoupled as close as possible to the package.

V_{SSA} and V_{SSD} (Ground)

Similar to V_{CCA} and V_{CCD}, the analog and digital circuits internal to the ISD1200/1400 Series use separate ground buses to minimize noise. These pins should be tied together as close as possible to the device.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 1. OPERATIONAL MODES

Address Ctrl. (HIGH)	Function	Typical Use	Jointly* Compatible
A0	Message cueing	Fast-forward through messages	A4
A1	Delete $\overline{\text{EOM}}$ markers	Position $\overline{\text{EOM}}$ marker at the end of the last message	A3, A4
A2	Unused		
A3	Looping	Continuous playback from Address 0	A1
A4	Consecutive addressing	Record/Play multiple consecutive messages	A0, A1
A5	Unused		

** Indicates additional operational modes which can be used simultaneously with the given mode.*

Address Inputs (A0-A7)

The Address Inputs have two functions, depending upon the level of the two Most Significant Bits (MSB) of the address.

If either of the two MSBs is LOW, the inputs are ALL interpreted as address bits and are used as the start address for the current Record or Playback cycle. The address pins are inputs only and do not output internal address information as the operation progresses. Address inputs are latched by the falling edge of $\overline{\text{PLAYE}}$, $\overline{\text{PLAYL}}$ or $\overline{\text{REC}}$.

OPERATIONAL MODES

The ISD1200/1400 Series is designed with several built-in operational modes provided to allow maximum functionality with a minimum of additional components, described in detail below. The operational modes use the address pins on the ISD1200/1400 devices, but are mapped outside the valid address range. When the two Most Significant Bits (MSBs) are HIGH, the remaining address signals are interpreted as mode bits and NOT as address bits. Therefore, operational modes and direct addressing are not compatible and cannot be used simultaneously.

There are two important considerations for using operational modes. First, all operations begin initially at address 0, which is the beginning of the ISD1200/1400 address space. Later operations can begin at other address locations, depending on the operational mode(s) chosen. In addition, the address pointer is reset to 0 when the device is changed from Record to Playback, Playback to Record, or when a Power-Down cycle is executed.

Second, an Operational Mode is executed when any of the control inputs, $\overline{\text{PLAYE}}$, $\overline{\text{PLAYL}}$, or $\overline{\text{REC}}$, go LOW and the two MSBs are HIGH. This Operational Mode remains in effect until the next LOW-going control input signal, at which point the current address/mode levels are sampled and executed.

(Note: The two MSBs are on pins 9 and 10 for each ISD1200/1400 Series member.)

OPERATIONAL MODE DESCRIPTIONS

The Operational Modes can be used in conjunction with a microcontroller, or they can be hard-wired to provide the desired system operation.

A0 — Message Cueing ($\overline{\text{PLAYE}}$ or $\overline{\text{PLAYL}}$ only)

Message Cueing allows the user to skip through messages, without knowing the actual physical addresses of each message. Each control input LOW pulse causes the internal address pointer to skip to the next message. This mode should be used for Playback only, and is typically used with the A4 Operational Mode.

A1 — Delete $\overline{\text{EOM}}$ Markers ($\overline{\text{REC}}$ only)

The A1 Operational Mode allows sequentially recorded messages to be concatenated into a single message with only one $\overline{\text{EOM}}$ marker set at the end of the combined message. When this operational mode is configured, messages recorded sequentially are played back as one continuous message.

A2 — Unused

A3 — Message Looping ($\overline{\text{PLAYE}}$ or $\overline{\text{PLAYL}}$ only)

The A3 Operational Mode allows for the automatic, continuously repeated playback of the message located at the beginning of the address space. A message CAN completely fill the ISD1200/1400 device and will loop from beginning to end.

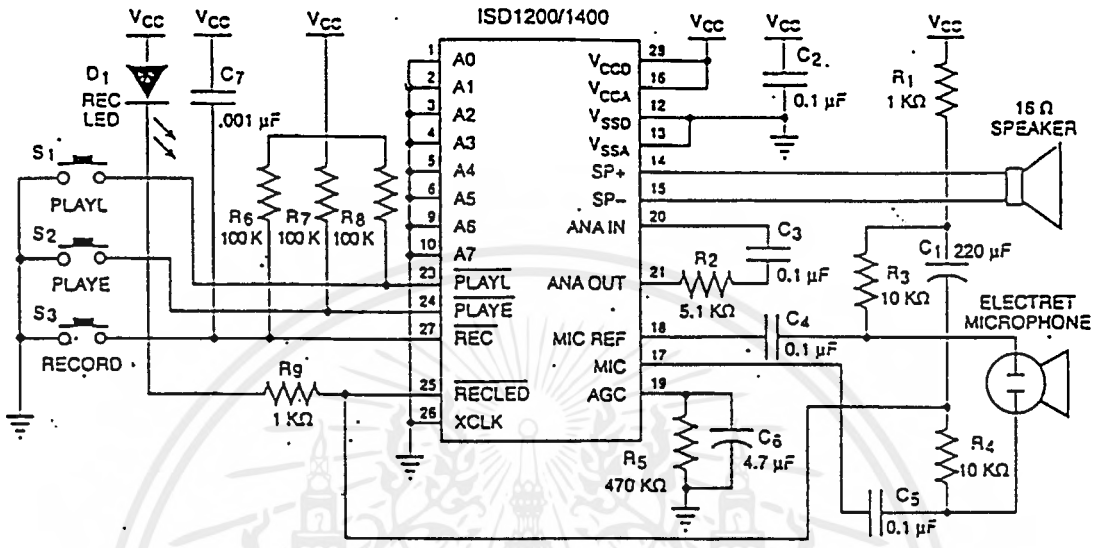
A4 — Consecutive Addressing

During normal operations, the address pointer will reset when a message is played through to an $\overline{\text{EOM}}$ marker. The A4 Operational Mode inhibits the address pointer reset, allowing messages to be played back consecutively.

A5 — Unused

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATION SCHEMATIC



Note: ISD Application Notes and Design Manual available

FUNCTIONAL DESCRIPTION EXAMPLE

The following example operating sequence demonstrates the functionality of the ISD1200/1400 Series devices.

1. Record a message filling the address space.

Pulling the $\overline{\text{REC}}$ signal LOW initiates a record cycle from the beginning of the message space. If $\overline{\text{REC}}$ is held LOW, the recording continues until the message space has been filled. Once the message space is filled, recording ceases. The device will automatically power down after $\overline{\text{REC}}$ is pulled HIGH.

2. Edge-activated playback.

Pulling the $\overline{\text{PLAYE}}$ signal LOW initiates a playback cycle from the beginning of the message space. The rising edge of $\overline{\text{PLAYE}}$ has no effect on operation. If a recording has filled the message space, the entire message is played. When the device reaches the end of the message space, it automatically powers down. A subsequent falling edge on $\overline{\text{PLAYE}}$ initiates a new play cycle from the start address.

3. Level-activated playback.

Pulling the $\overline{\text{PLAYL}}$ signal LOW initiates a playback cycle from the beginning of the message space. If $\overline{\text{PLAYL}}$ remains LOW, the device plays through to the end of the message and subsequently enters the power-down mode.

4. Level-activated playback (truncated).

If $\overline{\text{PLAYL}}$ is pulled HIGH any time during the play-back cycle, the device stops playing and enters the power-down mode. A subsequent falling edge on $\overline{\text{PLAYL}}$ initiates a new play cycle from the start address.

5. Record (interrupting playback).

The $\overline{\text{REC}}$ signal takes precedence over other operations. Any LOW-going transition on $\overline{\text{REC}}$ initiates a new record operation from the beginning of the start address, regardless of any current operation in progress.

6. Record a message, partially filling the address space.

A record operation need not fill the entire message space. Releasing the $\overline{\text{REC}}$ signal HIGH before filling the message space causes the recording to stop and an end-of-message marker to be placed. The device powers down automatically.

7. Play back a message, partially filling the address space.

Pulling the $\overline{\text{PLAYE}}$ or $\overline{\text{PLAYL}}$ signal LOW initiates a playback cycle which is then completed when the end-of-message marker is encountered. Playback ceases and the device powers down.

8. $\overline{\text{RECLED}}$ operation.

The $\overline{\text{RECLED}}$ output pin provides an active-LOW signal which can be used to drive an LED as a "record in-progress" indicator. It returns to a HIGH state when the $\overline{\text{REC}}$ pin is released HIGH or when the recording is completed due to the message space being filled.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATIONS NOTE

Some users may experience an unexpected recording taking place when their circuit is powered up, or the batteries are changed and V_{CC} rises faster than REC. This undesired recording prevents playback of the previously recorded message. A spurious End Of Message ($\overline{EOM}$) marker appears at the very beginning of the memory, preventing access to the original message, and nothing is played.

To prevent this occurrence, place a capacitor (approx. .001 µF) between the control pin ($\overline{REC}$) and V_{CC}. This pulls the control pin voltage up with V_{CC} as it rises. Once the voltage is HIGH, the pull-up device will keep the pin HIGH until intentionally pulled LOW, preventing the false $\overline{EOM}$ marker.

Since this anomaly is dependent upon factors such as the capacitance of the user's printed circuit board, not all circuit designs will exhibit the spurious marker. It is recommended, however, that the capacitor is included for design reliability. A more detailed explanation and resolution of this occurrence is described in the ISD Application Notes and Design Manual.

ABSOLUTE MAXIMUM RATINGS (ISD1200/1400 SERIES – PACKAGED)

Condition	Value
Temperature under bias	- 65° C to + 125° C
Storage temperature range	- 65° C to + 150° C
Voltage applied to any pin	(V_{SS} - 0.3 V) to (V_{CC} + 0.3 V)
Voltage applied to any pin (Input current limited to ± 20 mA)	(V_{SS} - 1.0 V) to (V_{CC} + 1.0 V)
Lead temperature (soldering - 10 seconds)	300° C
V_{CC} - V_{SS}	- 0.3 V to + 7.0 V

Stresses above those listed may cause permanent damage to the device. Exposure to the absolute maximum ratings may affect device reliability.

DC PARAMETERS (ISD1200/1400 SERIES – PACKAGED)

Operating Conditions: T_A = 0° to 70° C, V_{CC} = 4.5 V to 6.5 V [1], V_{SS} = 0 V [2]; unless otherwise noted

Symbol	Parameters	Min	Typ [3]	Max	Units	Conditions
V_{IL}	Input Low Voltage			0.8	V	
V_{IH}	Input High Voltage	2.4			V	
V_{OL}	Output Low Voltage			0.4	V	I_{OL} = 4.0 mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = - 1.6 mA
I_{CC}	V_{CC} Current @ 5.5 V (Operating)		15	30	mA	V_{CC} = 5.5 V, R_{EXT} = ∞ [4]
I_{CC}	V_{CC} Current @ 6.5 V (Operating)		15	35	mA	V_{CC} = 6.5 V, R_{EXT} = ∞ [4]
I_{SB}	V_{CC} Current (Standby)		0.5	10	µA	[4] [5]
I_{IL}	Input Leakage Current			±1	µA	[4] [5]
I_{ILPD}	Input Current HIGH w/Pull Down			130	µA	Force V_{CC} [6] [7]
R_{EXT}	Output Load Impedance	16			Ω	Speaker Load
R_{MIC}	Preamp In Input Resistance		10		KΩ	Pins 17, 18
$R_{ANA\ In}$	Ana In Input Resistance		3		KΩ	
A_{PRE1}	Preamp Gain 1		24		dB	AGC = 0.0 V
A_{PRE2}	Preamp Gain 2		- 45	- 15	dB	AGC = 2.5 V
A_{ARP}	Ana In to SP+/-		22		dB	
R_{AGC}	AGC Output Resistance		5		KΩ	
I_{PREH}	Preamp Out Source		- 2		mA	@ V_{OUT} = 1.0 V
I_{PREL}	Preamp In Sink		0.5		mA	@ V_{OUT} = 2.0 V

Notes: 1. $V_{CC} = V_{CCA} = V_{CCD}$.
2. $V_{SS} = V_{SSA} = V_{SSD}$.
3. Typical values @ T_A = 25° C and 5.0 V.
4. V_{CCA} and V_{CCD} connected together.
5. $\overline{REC}$, $\overline{PLAYL}$ and $\overline{PLAYE}$ must be V_{CCD}.
6. Pin 26.
7. Applies only to ISD120(X) and ISD140(X) future version (see page 13).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC PARAMETERS (ISD1200/1400 SERIES – PACKAGED)

Operating Conditions: T_A = 0° to 70° C, V_{CC} = 4.5 V to 6.5 V [1], V_{SS} = 0 V [2]; unless otherwise noted

Symbol	Characteristic	Min	Typ [3]	Max	Units	Conditions
THD	Total Harmonic Distortion		1		%	@ 1 KHz
T_{LED1}	$\overline{\text{RECLED}}$ ON Delay		5		µsec	
T_{LED2}	$\overline{\text{RECLED}}$ OFF Delay		48.6		msec	
T_S	A3 Loop Setup Time	300			nsec	
T_H	A3 Loop Hold Time	0			nsec	
T_{RPUD}	Record Power-Up Delay		32		msec	
T_{RPDD}	Record Power-Down Delay		32		msec	
T_{PPUD}	Play Power-Up Delay		32		msec	
T_{PPDD}	Play Power-Down Delay		8.1		msec	
P_{OUT}	Speaker Output Power		12.2		mW	R_{EXT} = 16 Ω
V_{OUT}	Voltage Across Speaker Pins		1.25	2.5	V p-p	R_{EXT} = 600 Ω
V_{IN1}	MIC Input Voltage			20	mV	Peak-to-Peak [4]
V_{IN2}	ANA IN Input Voltage			50	mV	Peak-to-Peak

AC PARAMETERS

Symbol	Characteristic	ISD-1210	ISD-1212	ISD-1416	ISD-1420	Units	Conditions
F_S	Sampling Frequency (max)	6.4	5.3	8	6.4	KHz	Internal Oscillator
BW	Bandwidth (max)	2.7	2.3	3.4	2.7	KHz	3 dB Roll-Off Point [5]
T_{RMW}	Record Pulse Width (max)	10	12	16	20	sec.	
T_{PLAY}	Playback Duration (min)	10	12	16	20	sec.	
T_{LED2}	$\overline{\text{RECLED}}$ OFF Delay	48.6	58.3	38.9	48.6	msec	[6]
T_{RPUD}	Rec. Power-Up Delay	32	39	26	32	msec	[6]
T_{RPDD}	Rec. Power-Down Delay	32	39	26	32	msec	[6]
T_{PPUD}	Play Power-Up Delay	32	39	26	32	msec	[6]
T_{PPDD}	Play Power-Down Delay.	8.1	9.7	6.5	8.1	msec	[6]

Notes: 1. $V_{CC} = V_{CCA} = V_{CCD}$.
2. $V_{SS} = V_{SSA} = V_{SSD}$.
3. Typical values @ T_A = 25° C, 5.0 V, and 6.2 KHz sample rate.
4. With 12 KΩ series resistor at ANA IN.
5. Low-frequency cutoff depends upon value of external capacitors (see Pin Descriptions).
6. Typical values @ T_A = 25° C and 5.0 V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้