



ระบบควบคุมการเปิดปิดเครื่องใช้ไฟฟ้าด้วยเวลาจริง
ELECTRICAL EQUIPMENT CONTROLLER
WITH REAL TIME CLOCK



โดย
 นายเทอดศักดิ์ กังวาลวงศ์สกุล
 นายโอมฤทธิ์ ม้าลัย

วัน เดือน ปี.....-5.ค.ศ.2541
 เลขทะเบียน.....038521
 เลขเรียกหนังสือ.....T.4006๕๗๒๘๘

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
 สาขาวิชาวิศวกรรมการวัดคุมทางอุตสาหกรรม
 คณะวิศวกรรมศาสตร์
 สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
 ปีการศึกษา 2540

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีก **038521**

ปริญญานิพนธ์ปีการศึกษา 2540

ภาควิชาวิศวกรรมการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมการเปิด-ปิดเครื่องใช้ไฟฟ้าด้วยเวลาจริง

ELECTRICAL EQUIPMENT CONTROLLER WITH REAL TIME CLOCK

ผู้จัดทำ

- | | |
|---------------------------------|----------|
| 1. นายเทอดศักดิ์ กังวาลวงศ์สกุล | 37014151 |
| 2. นายโอมฤทธิ์ มาลัย | 37014610 |

อาจารย์ที่ปรึกษา

(ผศ. ทรงชัย วีระทวิมาศ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบควบคุมการเปิด - ปิดเครื่องใช้ไฟฟ้าด้วยเวลาจริง

นาย เทอดศักดิ์ กังวาลวงศ์สกุล

นาย โอมฤทธิ์ มาลัย

ผศ. ทรงชัย วีระทวีมาศ

อาจารย์ที่ปรึกษา

ปีการศึกษา 2540

บทคัดย่อ

ปริญญานิพนธ์นี้เป็นการนำเอาไมโครคอนโทรลเลอร์ในตระกูล MCS-51 เบอร์ AT89C52 ซึ่งสามารถตั้งเวลาเปิด-ปิด อุปกรณ์ไฟฟ้าที่ใช้เป็นประจำวันหรือประจำสัปดาห์ได้ด้วย โดยใช้สัญญาณของระบบซึ่งได้มาจากวงจร timer ภายใน CPU จึงทำให้ลดค่าใช้จ่ายของวงจรฮาร์ดแวร์ ซึ่งการควบคุมโปรแกรมมีโหมดการทำงานต่างๆ 8 โหมด โดยสามารถบันทึกได้ 40 รายการ

ELECTRICAL EQUIPMENT CONTROLLER WITH REAL TIME CLOCK

MR.TURDSAK KUNGWANWONGSAKUL

MR.OHMARIT MALAI

MR. SONGCHAI WEERATHAWEEMAS

ADVISOR

1997

ABSTRACT

This project uses Microcontroller #AT89C52 in the family of MCS-51 to control the electrical equipment with real time clock. This controller can set the time to on-off electrical equipment that use daily or weekly. We use the timer in the CPU as the real time clock. Thus we can save the hardware expense and this controller can record 40 records with 8 working types.

สารบัญ

เรื่อง	หน้า
1. บทนำ	1
2. โครงสร้างภายในของ 8051	2
2.1 โครงสร้างภายในของ 8051	2
2.2 พอร์ตของ 8051	5
2.3 หน่วยศูนย์กลางประมวลผล ซีพียู	8
2.4 การจัดการหน่วยความจำ	9
3.การอินเตอร์รัพต์	15
4. ไทม์เมอร์/เคาน์เตอร์	19
4.1 ไทม์เมอร์ โหมด	20
4.2 เคาน์เตอร์ โหมด	21
5.หน่วยแสดงผลข้อมูล	24
5.1 ขารับสัญญาณของ LCD	24
5.2 การต่อเข้าระบบไมโคร	24
5.3 ชุดคำสั่งอ่านเขียนข้อมูลและการแสดงข้อความ	25
5.4 ความเข้าใจพื้นฐาน	27
5.5 รายละเอียดแต่ละคำสั่ง	27
5.6 การอ่านและเขียนข้อมูลกับ DDRam/CGRam	29
5.7 แนวทางการเขียนโปรแกรมควบคุม	30
6. โครงสร้างและหลักการทำงาน	31
6.1 โฟล์วชาร์ทแสดงการทำงานของโปรแกรมหลัก	33
6.2 โฟล์วชาร์ทแสดงการเช็ค record	34
6.3 โฟล์วชาร์ทแสดงการป้อนค่าในการกดคีย์และแสดงผล	35
6.4 โฟล์วชาร์ทแสดงการกำเนิดสัญญาณเวลา ของระบบจาก timer ใน CPU	44
7. การทดลองและผลการทดลอง	52
8. บทวิจารณ์และสรุป	53
ภาคผนวก	
- คู่มือการใช้งาน	
หนังสืออ้างอิง	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

โครงการระบบควบคุมการปิด-เปิดเครื่องใช้ไฟฟ้าด้วยเวลาจริง (ELECTRICAL EQUIPMENT CONTROLLER WITH REAL TIME CLOCK) ซึ่งเป็นการนำเอาไมโครเซสเซอร์ เบอร์ 89C52 มาใช้งาน โดยการเขียนโปรแกรมควบคุมการทำงานอัดลงในตัว EPROM สำหรับใช้งานนั้นๆจะทำได้โดยการป้อนคำสั่งทางคีย์บอร์ด ส่วนการแสดงผลจะแสดงผลการทำงานที่จอ LCD และจะนำเอาทัชไปควบคุมอุปกรณ์ไฟฟ้าต่างๆ ให้ปิด-เปิดตามเวลาที่ต้องการได้ ซึ่งไมโครโปรเซสเซอร์ เบอร์ 89C52 ซึ่งจะมีวงจรมานาฬิกาภายในเป็นตัวนับและจับเวลา โดยมีโหมดการทำงาน 8 โหมด สามารถบันทึกรายการได้ 40 รายการ ซึ่งการควบคุมทำได้โดยการนำเอาสัญญาณไปประมวลผลแล้วไปจับยังรีเลย์อีกต่อไป

เนื้อหาของโครงการแบ่งได้ 5 ส่วนด้วยกันคือ

ส่วนแรก กล่าวถึงทฤษฎีของอุปกรณ์ต่างๆที่ใช้ในโครงการนี้ เช่น จอแสดงผล , ไอซีเบอร์ 89C52

ส่วนที่สอง กล่าวถึงการออกแบบและโครงสร้างการทำงาน

ส่วนที่สาม กล่าวถึงการทดลองและผลการทดลอง

ส่วนที่สี่ กล่าวถึงบทวิจารณ์และสรุป

ส่วนที่ห้า เป็นภาคผนวก ประกอบด้วย คู่มือการใช้งาน และหนังสืออ้างอิงต่างๆ

สำหรับในส่วนขอโปรแกรมทางคณะผู้จัดทำ ได้เก็บส่วนของโปรแกรมไว้ในแผ่น Floppy Disk ขนาด 3.5 ” ซึ่งได้แนบมาที่ท้ายเล่มของปฏิญานินพนธ์นี้ด้วย

คณะผู้จัดทำหวังเป็นอย่างยิ่งว่าโครงการนี้ จะเป็นประโยชน์ในการใช้ควบคุมอุปกรณ์ไฟฟ้าต่างๆในโรงงานอุตสาหกรรม อาคารพาณิชย์ รวมถึงบ้านพักอาศัย

บทที่ 2

โครงสร้างของ MCS-51

2.0 บทนำ

ไมโครคอนโทรลเลอร์แบบชิปเดี่ยวตระกูล MCS-51 นี้ผลิตโดยบริษัทอินเทลมีอยู่ด้วยกันหลายรูปแบบดังตารางที่ 2.1

ตารางที่ 2.1

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 8751UH	4K	128	4	2		✓					6/5	
8052AH	8032AH	8752BH	8K	256	4	3		✓					8/6	
80C510H	80C310H	87C51	4K	128	4	2		✓					5/5	✓
80C52	80C32	—	8K	256	4	3		✓					8/6	✓
81C51FA	80C51FA	87C51FA	8K	256	4	3		✓					14/7	✓
82C51FB	80C51FA	87C51FB	19K	256	4	3		✓					14/7	✓
83C152JA	80C152JA	—	JK	256	5	2		✓		✓	2		19/11	✓
—	80C152JD	—	—	256	7	2		✓		✓	2		19/11	✓
83C152JC	80C152JC	—	8K	256	5	2		✓		✓	2		19/11	✓
—	80C152JD	—	—	256	7	2		✓		✓	2		19/11	✓
83C452	80C452	87C452P	8K	256	5	2		✓					9/8	✓

คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51

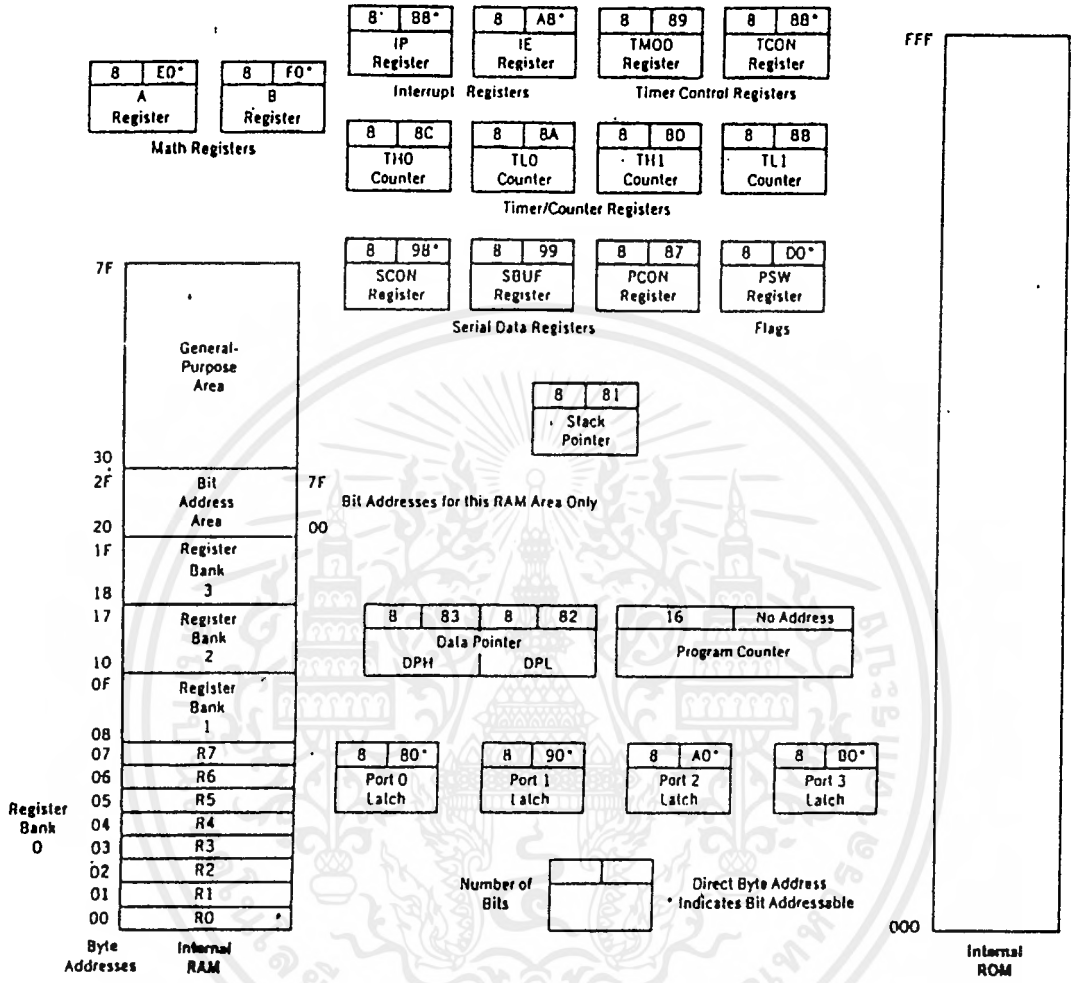
- ต้องการแหล่งจ่ายไฟ +5 V เพียงชุดเดียว
- มีหน่วยความจำโปรแกรม (PROGRAM MEMORY) ขนาด 4 กิโลไบต์สำหรับเบอร์ 8051 และ 8031, 8032 ไม่มีหน่วยความจำชุดนี้ ส่วน 8052 หน่วยความจำถึง 8 กิโลไบต์
- มีหน่วยความจำสำหรับเก็บข้อมูล (DATA MEMORY) ขนาด 128 ไบต์ สำหรับ 8052 มีถึง 256 ไบต์
- หน่วยความจำสำหรับโปรแกรมและค่า (PROGRAM MEMORY และ DATA MEMORY แยกจากกันอย่างละ 64 กิโลไบต์)
- คำสั่งที่ใช้เวลาน้อยที่สุดประมาณ 1 μ s เมื่อทำงานที่ความถี่ 12 MHz
- มี Timer/Counter ขนาด 16 บิต 2 ชุด (สำหรับ 8052 มี 3 ชุด) ทำงานได้ 4 โหมด
- รับอินเทอร์รัปต์ได้ 6 แหล่ง 5 เวกเตอร์
- มีพอร์ตรับส่งข้อมูลอนุกรม (UART) 2 พอร์ต ทั้งรับและส่งในเวลาเดียวกันได้ (Full Duplex) เลือกรูปแบบการส่งข้อมูลได้ 4 โหมด
- มีคำสั่งในการทำ AND, OR หรือ COMPLEMENT ได้ทั้งแบบ 8 บิต และ 1 บิต

2.1 โครงสร้างภายในของ 8051

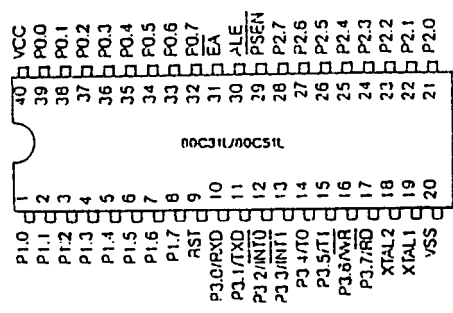
MCS-51 ใช้เทคโนโลยีในการผลิตเป็นแบบ NMOS และ CMOS เบอร์ 8032 และ 8052

จะมี ROM BASIC อยู่ภายในจึงสะดวก สำหรับโปรแกรมเมอร์ด้วยภาษาแบบสติกโครงสร้างภายในสำหรับเบอร์ 8051 ดังแสดงในรูป 2.1(a) และ 2.1(b)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 (บ) ตำแหน่งต่างๆ ของเรจิสเตอร์ และหน่วยความจำ



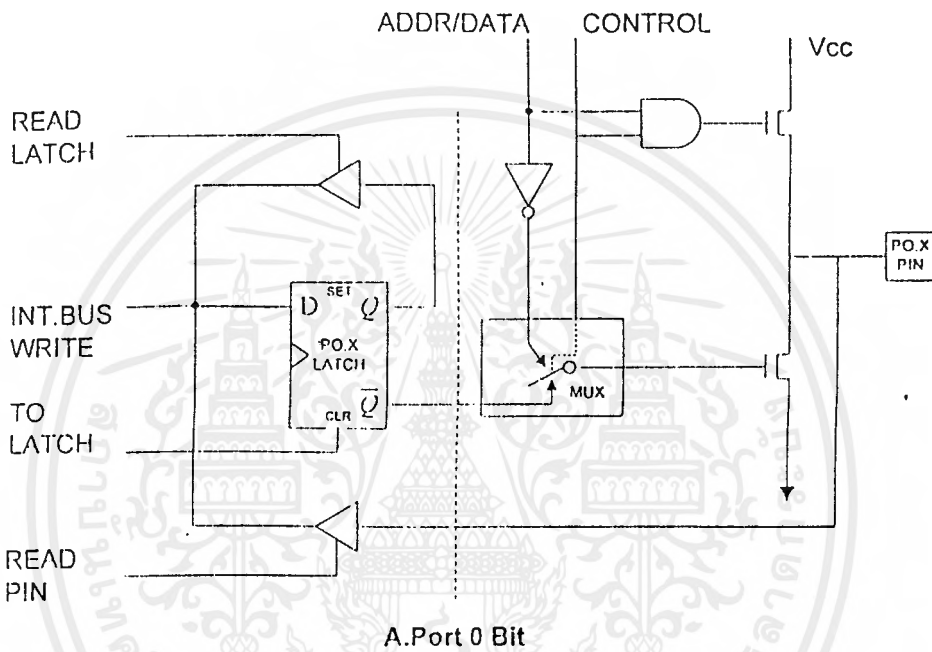
รูปที่ 2.2 การจัดวางขาของ 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ **รูปที่ 2.2** การจัดวางขาของ 8051 มีอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 พอร์ตของ 8051

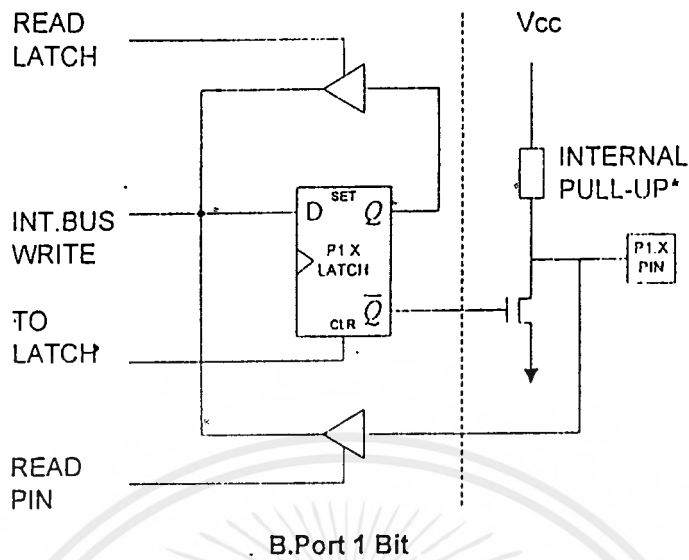
8051 เป็น ไมโครคอนโทรลเลอร์ 40 ขา มีขนาดต่างๆดังนี้

- Vcc(ขา 40) ต่อกับ +5 V
- Vss (ขา 20) เป็นขา GND
- PORT 0 (ขา 32-39) มีทั้งหมด 8 บิตคือ (P0.0 -P0.7) มีโครงสร้างแบบ Open Drain Bi-directional ดังแสดงดังรูป 2.3



รูปที่ 2.3 แสดงการสร้างพอร์ต 0 (บิต)

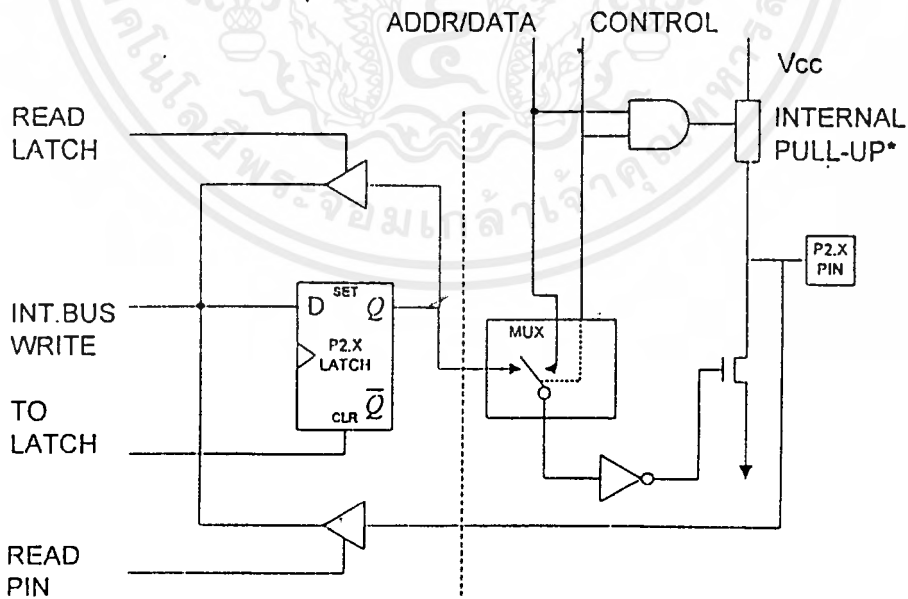
- พอร์ต 0 (ขา 32-39) มีทั้งหมด 8บิต คือ (P0.0- P0.7) ใช้งานได้ 2 หน้าที คือ ส่งแอดเดรส และค่าได้ออกไปให้หน่วยความจำภายนอกเมื่อทำการเขียนข้อมูลลงหน่วยความจำภายนอกควบคุมด้วยขา Control จากรูป 2.3 ประกอบและอีกหน้าหนึ่งเป็นพอร์ต I/O ถ้าต้องการ ให้ทำงานเป็นอินพุทพอร์ตต้องส่งลอจิก " 1" ไปยังพอร์ตนี้ จะมีผลให้ Q ของ D-FF เป็น " 0" ทำให้ FET ตัวล่างมีสถานะ OFF สัญญาณที่ใช้อินพุทพอร์ต PIN (พอร์ต P0.XPIN) จะใช้สัญญาณ READ LATH เมื่อ ถูกกระตุ้นที่ Tri- State Buffer ตัวบน
- พอร์ต 1 (ขา 1-8) มีทั้งหมด 8 บิต คือ (P1.0- P1.7) มีโครงสร้างคล้าย พอร์ต 0 แต่จะใช้ความต้านทานภายในทุกอัทแทน(Internal Pull Up Register) มีโครงสร้างดังรูป 2.4



B.Port 1 Bit

รูปที่ 2.4 โครงสร้างของพอร์ท 1 (บิต)

- พอร์ท 2 (ขา 21-28) มีทั้งหมด 8 บิต คือขา (P2.0-2.7) มีโครงสร้างคล้าย PORT 0 โดยมี FET ตัวล่างตัวเดียวส่วนด้านบนใช้ความต้านทานลูปอัพแทน (Internal PullUp) พอร์ทนี้ทำงาน 2 หน้าที่ คือสามารถใช้เป็นพอร์ทสำหรับส่งแอสเคต 8 บิตบน (A8- A15) และเป็น I/O พอร์ท ใช้งานทั่วไป เมื่อจะใช้งานเป็นอินพุตพอร์ท ต้องส่งลอจิก "1" มาที่พอร์ทนี้ก่อน เพื่อบังคับให้ FET อยู่ในสภาวะ OFF ดังแสดงในรูป 2.5



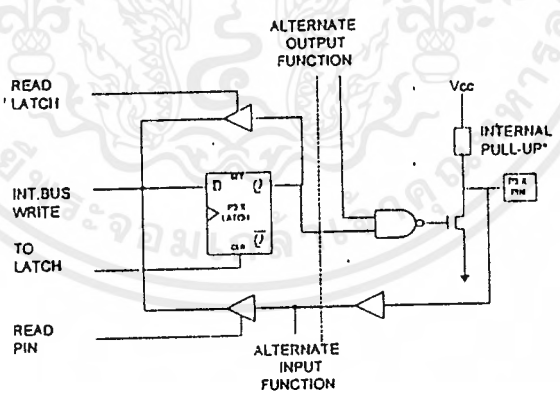
C.Port 2 Bit

รูปที่ 2.5 โครงสร้างของพอร์ท 2 (บิต)

- พอร์ต 3 (ขา 10-17) มีทั้งหมด 8 บิต คือขา (P3.0-P3.7) มีโครงสร้างคล้ายพอร์ต 1 พอร์ตนี้ทำหน้าที่เป็น I/O พอร์ต ถ้าจะให้พอร์ตนี้เป็น I/P PORT ก็ให้ส่งลอจิก "1" มาที่พอร์ติก่อน และอีกหน้าที่หนึ่งคือ ส่งสัญญาณออกมาและรับสัญญาณเข้าไป สัญญาณต่างๆมีดังนี้

- P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับสัญญาณแบบอนุกรม
- P3.1/RXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม
- P3.2/INT0 (Extcrnal Interupt) ใ้รับสัญญาณขั้จ้งหวะจากภายนอก
- P3.3/INT1 (Extcrnal Interupt) ใ้รับสัญญาณขั้จ้งหวะจากภายนอก
- P3.4/T0 (Timer/Counter 0 External Input) ขารับสัญญาณ ไปยังวงจร Timer/Counter 1 ซึ่งจะมีการทำงานเหมือนกับ T0
- P3.5/T1 (Timmer /Counter 1 External Input) ขารับสัญญาณเข้าไปยัง Timer/Counter 1 ซึ่งมีการทำงานเหมือนกับ T0
- P3.6/WR (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051
- P3.7/RD (External Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก

โครงสร้างของ (พอร์ต 3) ดังแสดงในรูป 2.6



รูปที่ 2.6 โครงสร้างของพอร์ต 3 (บิต)

- ALE (ขา 30) เป็นขาในการส่งส ไดรปสำหรับใช้ในการแลคซ์แอดเคุสไบต์ต่ำ(A0-A7) ที่ส่งออกมาจาก(พอร์ต 0) สัญญาณนี้จะออกทีพุกๆ 2 ครั้งใน 1 แมซซึน ไซเคิล(1/16 ของขาสัญญาณนาฬิกา)
- PSEN(ขา 29) เป็นขาที่ใช้ส่งส ไดรปสำหรับอ่านข้อมูลจาก Program Memory ภายนอก หน่วยความจำประเภท ROM EPROM) สัญญาณนี้จะส่งออกมา 2 ครั้งในแต่ละแมซซึน ไซเคิลแต่ถ้าเป็นการอ่าน

Internal Program Memory จะไม่มีการออกที่ขานี้

- EA (ขา 30) ถ้าป้อนลอจิก "0" เข้าที่ขาที่ซีพียูจะอ่านค่าจาก Program Memory ภายนอกชิพเท่านั้น แต่ถ้าถูกป้อนด้วยลอจิก "1" ก็จะอ่านโปรแกรมภายในชิพ
- RST (ขา 9) เป็นขารีเซ็ตซีพียูจะรีเซ็ตได้ก็ต่อเมื่อ ป้อนลอจิก "1" เข้าที่ขานี้นานอย่างน้อย 2 แมกซ์นาไซเคล เมื่อซีพียูรีเซ็ตค่าต่างๆในรีจิสเตอร์ต่างๆจะมีค่าดังตั้ง Table 2.1
- XTAL1(ขา 19) ใช้ต่อคริสตอลภายนอกโดยเป็นอินพุตเข้าสู่วงจรรอสซซิเลเตอร์
- XTAL2(ขา 18) ใช้ต่อคริสตอลภายนอกโดยเป็นเอาต์พุตของวงจรรอสซซิเลเตอร์

2.3 หน่วยศูนย์กลางประมวลผลซีพียู

ซีพียูเป็นมันสมองของระบบไมโครคอมพิวเตอร์ การอ่าน โปรแกรม และทำงานตามคำสั่ง โปรแกรมจะกระทำที่ส่วนนี้ โดยใช้ในส่วนคณิตศาสตร์ และตรรกศาสตร์ทำงานร่วมกับรีจิสเตอร์ A ,B,PSW (Program Status Word) ,SP (Stack Pointer) ตัวนับโปรแกรม (PC :Program Counter) ขนาด 16 บิต และตัวชี้ตำแหน่งข้อมูล (DPTR:Data Pointer) ส่วนคณิตศาสตร์และตรรกศาสตร์(ALU:Aritmetic Logic Unit) ALU นี้ทำงานในฟังก์ชันคณิตศาสตร์และตรรกศาสตร์ด้วยตัวแปรต่างๆขนาด 8 บิต ที่มีลักษณะการทำงานทางคณิตศาสตร์ เป็น บวก ลบ คูณ หาร รวมทั้งทางตรรกศาสตร์ เช่น AND OR XOR รวมทั้งการเลื่อนและวงรอบบิต การเคลียร์ค่าและการกลับค่า (Complement) เป็นต้น ALU ยังสามารถที่จะตัดสินใจในการ ให้ไปกระโดด ไปทำคำสั่งของ โปรแกรมในส่วนอื่นๆตามเงื่อนไขที่คั่งและยังแบ่งรีจิสเตอร์ชั่วคราวไว้

สำหรับเป็นทางผ่านชั่วคราวของข้อมูลในการถ่ายเทระบบภายในคำสั่งที่มีการใช้ ALU ALU ยังมีความสามารถที่จะเพิ่มค่าในรีจิสเตอร์ในลักษณะการบวกด้วย 1 (Increment) หรือ คำนวณที่อยู่ข้อมูลที่将会นำไปเก็บ หรือการลดค่าครั้งละหนึ่ง ในลักษณะการลบด้วยค่าหนึ่ง (Decrement) โดยอัตโนมัติ หรือใช้ในการเปรียบเทียบค่าของตัวแปรทั้งสอง สิ่งสำคัญในการทำงานทางสถาปัตยกรรมของ MCS - 51 คือความสามารถในการทำงานสำหรับข้อมูลขนาด 8 บิต และ 1 บิต การใช้งานในระดับบิตในการเซต เคลียร์ หรือกลับค่า การเคลื่อนย้าย การทดสอบ และใช้ในการคำนวณทางตรรกขนาด 1 บิต ความสามารถเช่นนี้เหมาะสำหรับใช้งานควบคุมของสัญญาณเข้าออกที่มีการคิดและออกแบบครรถด้วยพีชคณิตบูลีน ซึ่งโดยปกติทำได้ลำบากสำหรับไมโคร โพรเซสเซอร์ทั่วไป งานในลักษณะนี้จึงได้ชื่อออกอย่างหนึ่งว่า ตัวประมวลผลบูลีน (Boolean Processor)

- แอควิวเลเตอร์ (Accumulator : ACC)

MCS - 51 ก็เช่นเดียวกับ MCS - 48 ที่ใช้แอควิวเลเตอร์ที่ขนาด 8 บิตเป็นแอควิวเลเตอร์คำสั่งส่วนใหญ่จะอ้างถึงตัวเรจิสเตอร์นี้ โดยถือค่าภายในเป็นค่าตัวคั่ง และรับค่าผลลัพธ์ที่ได้จากคำสั่งทางคณิตศาสตร์ เช่น บวก ลบ คูณ หาร เข้ามาเก็บไว้ ตัวแอควิวเลเตอร์ยังสามารถใช้เป็นตัวແລ่งกระทำหรือถูกกระทำในการทำงานทางตรรก และใช้เป็นตัวกลางในการถ่ายเทข้อมูลในการติดต่อกับอุปกรณ์ภายนอกไอโอ และหน่วยความจำภายนอก รวมถึงการตรวจสอบตารางข้อมูล

- เรจิสเตอร์ B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นเรจิสเตอร์พิเศษที่ใช้งานสำหรับคำสั่งของการคูณและหาร โดยให้เป็นที่เก็บตัวคูณหรือตัวหารและเป็นที่เก็บผลลัพธ์ตัวที่สองหลังการคูณและเศษหลังการหาร

• เรจิสเตอร์คำแสดงสถานะโปรแกรม (Program Status Word : PSW)

เรจิสเตอร์ PSW เป็นเรจิสเตอร์ที่แสดงผลที่ได้หลังจากการใช้คำสั่งต่างๆ และใช้เป็นตัวเลือกกลุ่มการทำงานของเรจิสเตอร์กลุ่มต่างๆ

• ตัวชี้ข้อมูล (Data Pointer : DPTR)

DPTR เป็นเรจิสเตอร์ขนาด 16 บิตที่ประกอบด้วยไบต์สูง (DPH) และไบต์ต่ำ (DPL) ที่สามารถเลือกแบ่งออกเป็นเรจิสเตอร์ 8 บิตสองตัวที่ใช้ได้อย่างอิสระ หรือจะใช้รวมกันทั้ง 16 บิตก็ได้ ในการเพิ่มหรือลด เพื่อประโยชน์ในการใช้เป็นฐานของเลขที่อยู่ในเรจิสเตอร์ในการกระโดดโดยทางอ้อมในการใช้คำสั่งเกี่ยวกับตารางข้อมูล และชี้ตำแหน่งของหน่วยความจำภายนอก

• พอร์ต 0 ถึง 3

เรจิสเตอร์ P0 , P1 , P2 , P3 ของกลุ่มเรจิสเตอร์ฟังก์ชันพิเศษจะเป็นตัวเลขค่าของพอร์ต 0 , 1 , 2 , 3 ตามลำดับในขณะที่ใช้งาน

• บัฟเฟอร์ข้อมูลอนุกรม (Serial Data Buffer : SBUF)

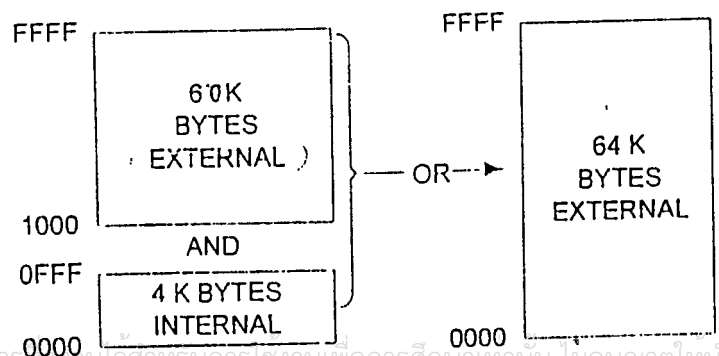
บัฟเฟอร์ข้อมูลอนุกรมแบ่งออกเป็นเรจิสเตอร์สองตัว ตัวหนึ่งเป็นบัฟเฟอร์การส่งและอีกตัวเป็นบัฟเฟอร์การรับ เมื่อข้อมูลถ่ายเทเข้า SBUF มันจะถ่ายเข้าบัฟเฟอร์ส่งซึ่งเป็นตัวจัดการส่งข้อมูลอนุกรมวิธีการเคลื่อนย้ายเข้า SBUF ขึ้นอยู่กับการติดตั้งโปรแกรม (Initial) การส่งเมื่อข้อมูลย้ายออกจาก SBUF จะเป็นการรับข้อมูลจากบัฟเฟอร์ตัวรับ

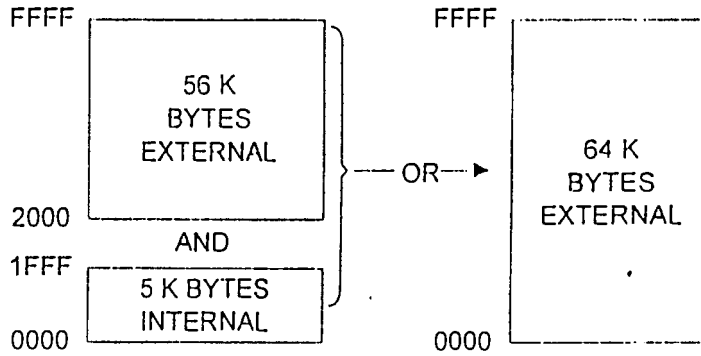
• เรจิสเตอร์ควบคุม (Control Register)

กลุ่ม SFR ที่เป็น IP , IE , TMOD , TCON , SCON และ PCON จะประกอบด้วยบิตที่ใช้ในการควบคุม และแสดงสถานะของการทำงานในระบบอินเทอร์รัพ ตัวจับเวลา/ตัวนับ และพอร์ตอนุกรม

2.4 การจัดการหน่วยความจำ

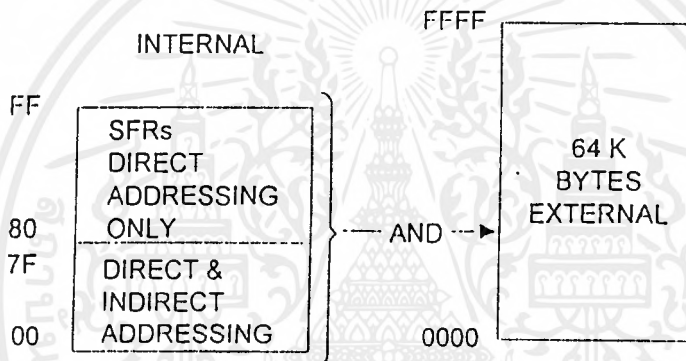
หน่วยความจำของ MCS - 51 มีด้วยกัน 2 ชนิดคือ หน่วยความจำโปรแกรม (Program Memory) และหน่วยความจำข้อมูล (Data Memory) หน่วยความจำโปรแกรมเป็นหน่วยความจำที่ใช้เก็บโปรแกรมสั่งงานบรรจุอยู่ในชิพ 8051 ส่วนที่เป็นหน่วยความจำโปรแกรมก็คือ ROM ขนาด 4 กิโลไบต์นั่นเอง แต่ถ้าเป็นเบอร์ 8052 ก็คือ ROM ขนาด 8 กิโลไบต์ ดังแสดงในรูป 2.7 และ 2.8



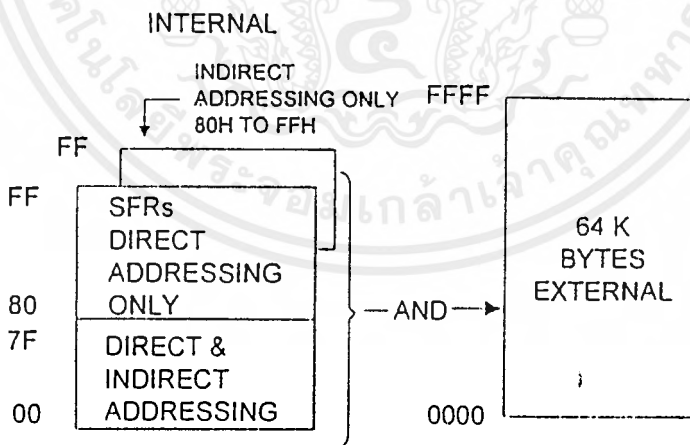


รูปที่ 2.8 ผังหน่วยความจำสำหรับเก็บโปรแกรมของ 8052

หน่วยความจำข้อมูลเป็นหน่วยความจำที่ใช้เก็บข้อมูล หน่วยความจำนี้สามารถเขียนข้อมูลลงไป และอ่านข้อมูลออกมาได้ ซึ่งเป็นหน่วยความจำภายในชิพมีเพียง 128 ไบต์ สำหรับเบอร์ 8051 และ 256 สำหรับเบอร์ 8052 ส่วนความจำภายนอกชิพมี 64 กิโลไบต์ดังแสดงในรูป 2.9 และ 2.10



รูปที่ 2.9 ผังหน่วยความจำข้อมูลของ 8051



รูปที่ 2.10 ผังหน่วยความจำข้อมูลของ 8052

บางครั้งอาจจะสงสัยว่าตำแหน่งของหน่วยความจำสำหรับโปรแกรมและข้อมูลมีตำแหน่งที่ซ้อนกัน ซึ่ทีจะรู้ได้อย่างไรว่าคิดค้กับหน่วยความจำที่เป็น โปรแกรมหรือข้อมูล บริษัทอินเทล ได้ออกแบบ แยกคำสั่งออกเป็น 3 ส่วน คือ

- MOV ใช้คิดค้กับ RAM ภายใน

- MOVC ใช้คิดค้กับหน่วยความจำโปรแกรม

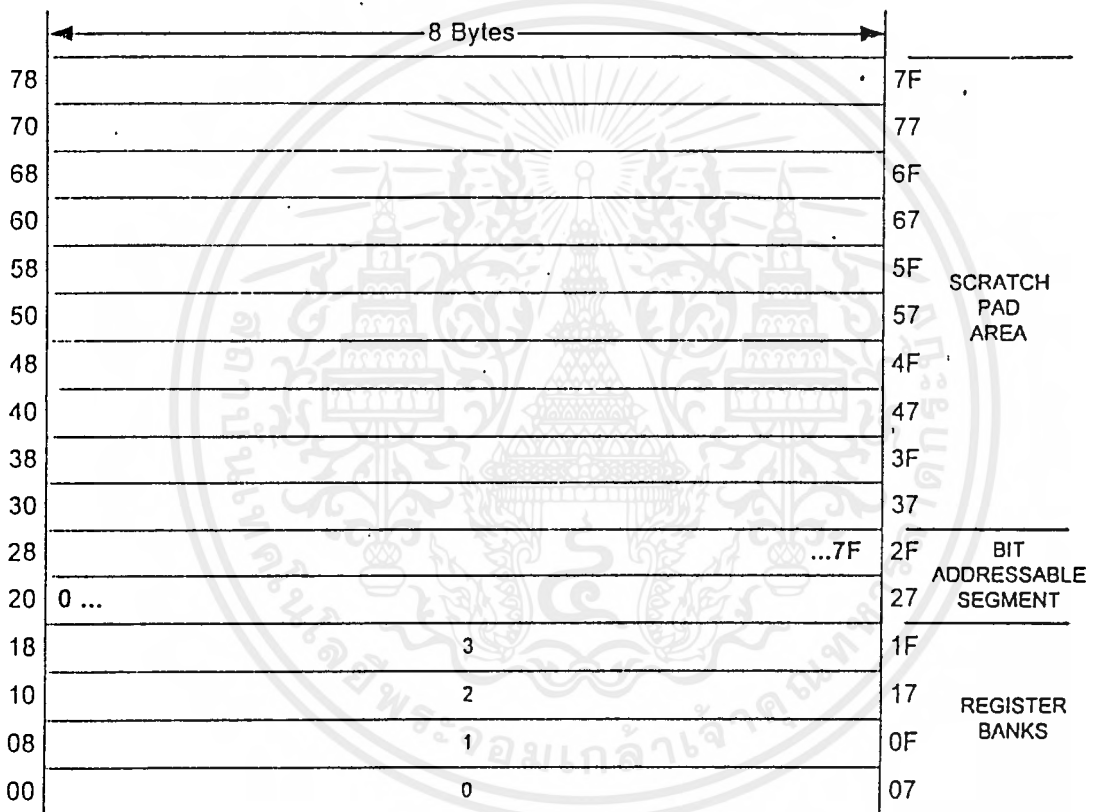
- MOVX ใช้ติดต่อกับหน่วยความจำข้อมูลภายนอกซีพ โดยระบุตำแหน่งผ่านเรจิสเตอร์ DPTR

2.4.1 พื้นที่หน่วยความจำที่เข้าถึงข้อมูลโดยทางอ้อมเท่านั้น (Indirect Address Area)

พื้นที่หน่วยความจำบริเวณ (80H - FFH) ตามรูป 2.9 เป็นพื้นที่ที่ซ้อนกันอยู่อย่างละ 128 ไบต์ โดยส่วนแรกจะเป็น SFR แอคเครส และ Indirect Area Address ดังนั้นผู้เขียน โปรแกรมถ้าจะติดต่อกับ SFR จะต้องใช้คำสั่งแบบเข้าถึงข้อมูลโดยตรงเท่านั้น ส่วนพื้นที่อีกส่วนหนึ่งจะเข้าถึงข้อมูลแบบทางอ้อมเท่านั้น

2.4.2 พื้นที่หน่วยความจำที่เข้าถึงข้อมูลโดยตรงและทางอ้อม (Direct and Indirect Address Area)

พื้นที่ 128 ไบต์ล่างสุดจะแบ่งเป็น 3 ส่วนดังรูป 2. 11



รูปที่ 2.11 128 ไบต์ของ RAM ที่เข้าถึงข้อมูลแบบทางตรงและทางอ้อม

1. เรจิสเตอร์แบงก์ (Register Bank 0 - 3)

ตั้งแต่ตำแหน่ง (00 - 1FH) จะเป็นส่วนของเรจิสเตอร์แบงก์ (0 - 3) โดยแบ่งเป็นแบงก์ละ 8 ไบต์รวมแล้วได้ 32 ไบต์ ถ้าซีพียูทำงานอยู่ที่แบงก์ 3 เมื่อถูกรีเซ็ตก็จะกลับมาทำงานที่แบงก์ 0 เสมอ และ SP จะมาเริ่มต้นที่ตำแหน่ง 07H ทันที

2. บริเวณหน่วยความจำที่ใช้คำสั่งเขียน/อ่านเกี่ยวกับบิตได้ (Bit Addressable Area)

ตั้งแต่ตำแหน่ง (20H - 1FH) จำนวน 16 ไบต์หรือถ้านับเป็นบิตจะได้เท่ากับ 128บิต ซึ่งตำแหน่งบิต 00 , 01 ,02 ,03 ,04 ,05 ,06 ,07 ก็คือตำแหน่งหน่วยความจำตำแหน่ง 20H ที่บิต 0 , 1 , 2 ,3 , 4 ,5 ,6 ,7 ตามลำดับ ดูรูป 2. 12

3. บริเวณหน่วยความจำที่ใช้งานทั่วไป (Scratch Pad Area)

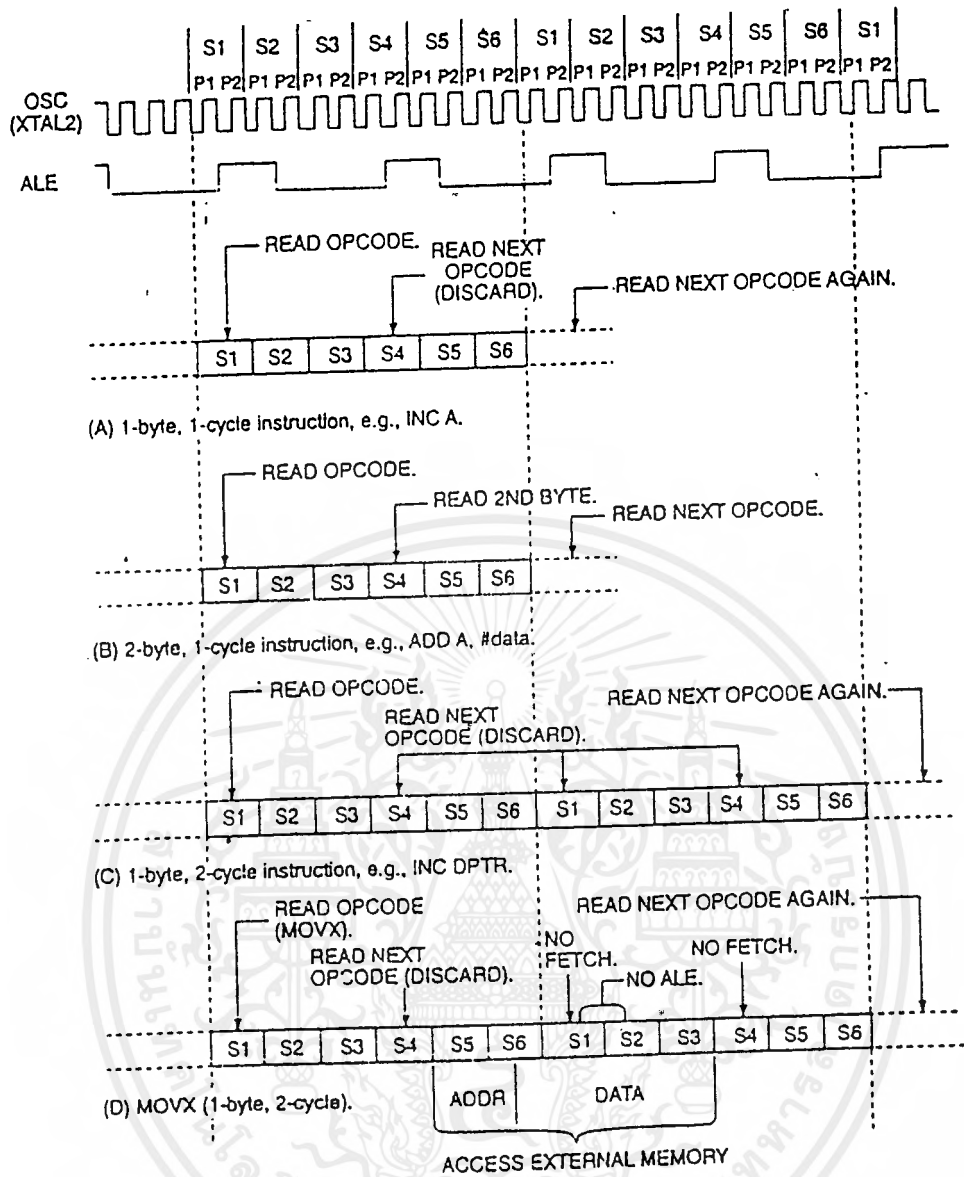
ตั้งแต่ตำแหน่ง (30H - 7FH) จะเขียนข้อมูลได้ที่ละไบต์เท่านั้น ไม่สามารถใส่คำสั่งเกี่ยวกับบิตได้ ถ้าซ้ายเนื้อที่สแตคมาบริเวณนี้ไปรบกวนการเขียนข้อมูลทับสแตค

RAM

Byte	(MSB)							(LSB)
7FH								
2FH	7F	7E	7D	7C	7B	7A	79	78
2EH	77	76	75	74	73	72	71	70
2DH	6F	6E	6D	6C	6B	6A	69	68
2CH	67	66	65	64	63	62	61	60
2BH	5F	5E	5D	5C	5B	5A	59	58
2AH	57	56	55	54	53	52	51	50
29H	4F	4E	4D	4C	4B	4A	49	48
28H	47	46	45	44	43	42	41	40
27H	3F	3E	3D	3C	3B	3A	39	38
26H	37	36	35	34	33	32	31	30
25H	2F	2E	2D	2C	2B	2A	29	28
24H	27	26	25	24	23	22	21	20
23H	1F	1E	1D	1C	1B	1A	19	18
22H	17	16	15	14	13	12	11	10
21H	0F	0E	0D	0C	0B	0A	09	08
20H	07	06	05	04	03	02	01	00

รูปที่ 2.12 แสดงตำแหน่งบิตต่าง ๆ ในแอดเดรส (20H - 2FH)

2.5 มั่งเวลาของซีพียู (CPU Timing)



รูปที่ 2.13 มั่งการทำงานของแต่ละคำสั่ง

การทำงานใน 1 คำสั่งคำสั่งจะกินเวลาเพียง 1 s เช่นคำสั่ง INC A ซึ่งเป็นคำสั่ง 1 ไบต์ 1 ไชเคลิล ซึ่งจะเป็นคล็อกไปเท่ากับ 12 ลูก โดยคล็อกลูกที่ 1 และ 2 จะอยู่ในช่วง S1 เฟส 1 และ S1 เฟส 2 และคล็อกลูกที่ 12 ก็อยู่ในช่วง S6P2 นั่นเอง (ปรกติแล้วซีพียูจะ RUN ด้วยความเร็วเท่ากับ 12 mHz ดังนั้นคล็อก 12 ลูกจะกินเวลาเท่ากับ 1 s

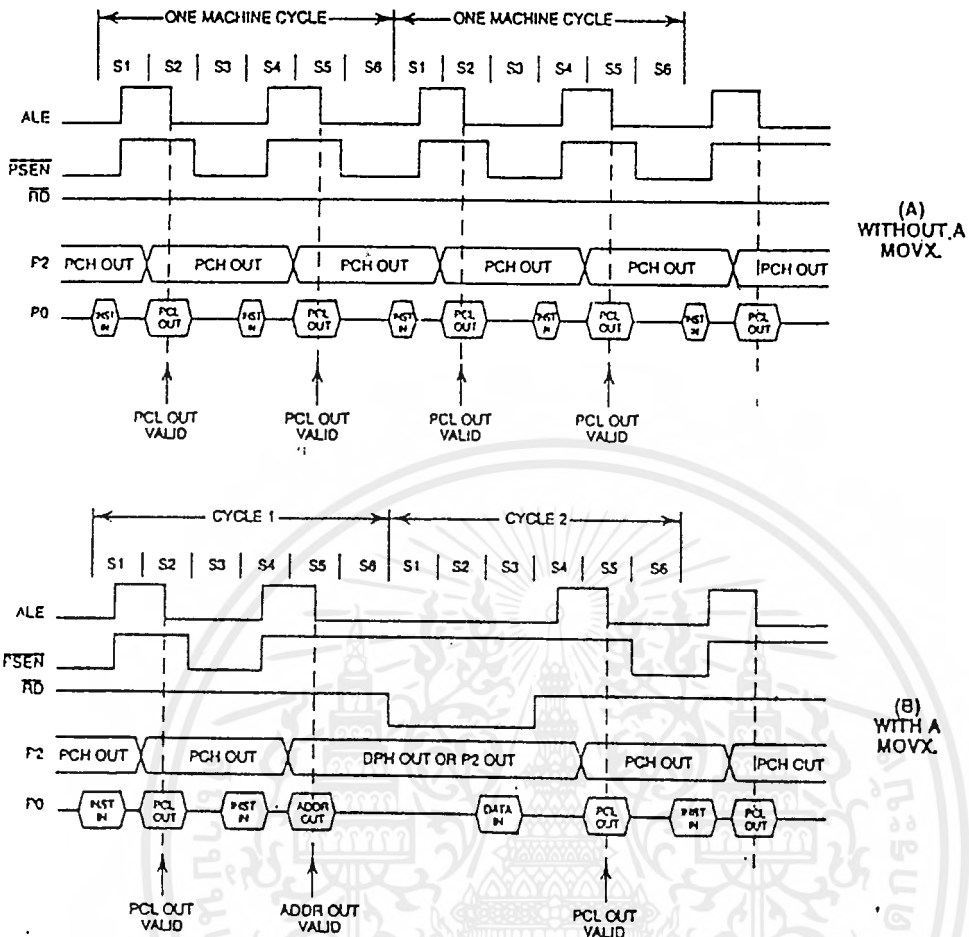
< คำว่า 1 เมชชีน ไชเคลิล คือช่วงการทำงานตั้งแต่ S1 จนถึง S6 >

รูป 2. (a) แสดงการทำงานของคำสั่ง INC A ซึ่งเป็นคำสั่ง 1 ไบต์ทำงานเสร็จภายใน 1 เมชชีน ไชเคลิล

รูป 2. (b) แสดงการทำงานของคำสั่ง ADD A, #Data ซึ่งเป็นคำสั่ง 2 ไบต์แต่ทำงานเสร็จใน 1 เมชชีน ไชเคลิล

รูป 2. (c) แสดงการทำงานของคำสั่ง INC DPTR ซึ่งเป็นคำสั่ง 1 ไบต์แต่ทำงานเสร็จใน 2 เมชชีน ไชเคลิล

รูป 2. (d) แสดงการทำงานของคำสั่ง MOVX ซึ่งเป็นคำสั่ง 1 ไบต์ แต่ทำงานเสร็จใน 2 เมชชีน ไชเคลิล



รูปที่ 2.14 แสดงเวลาแสดงการคิดต่อกับหน่วยความจำภายนอก

รูป 2. (a) เป็นผังเวลาของสัญญาณซึ่งเกี่ยวข้องกับเฟิร์ทซ์เมื่อส่วนของหน่วยความจำโปรแกรมอยู่ภายนอก ดังนั้นสัญญาณที่จะนำไปใช้อ่านออฟโคดจากหน่วยความจำโปรแกรมก็คือ PSEN ซึ่งจะแอกทีฟ 2 ครั้งใน 1 เมชชีนไซเคิล ดังนั้นสัญญาณที่ใช้อ่านข้อมูลจากหน่วยความจำโปรแกรมจะใช้สัญญาณ

รูป 2. (b) เป็นผังเวลาของสัญญาณที่เกี่ยวข้องกับการอ่านข้อมูลจากหน่วยความจำข้อมูล สัญญาณ PSEN จะมีเพียง 1 ลูก เพราะเวลาถัดมาจะเป็นช่วงเวลาในการอ่านข้อมูลจากหน่วยความจำข้อมูล โดยใช้สัญญาณ RD

< อาจสรุปได้ง่ายๆว่าการอ่านข้อมูลจากหน่วยความจำโปรแกรมจะใช้สัญญาณ PSEN และการอ่านข้อมูลจากหน่วยความจำข้อมูลจะใช้สัญญาณ RD ส่วนสัญญาณ ALE คือสัญญาณที่ใช้ในการแลทซ์แอดเดรส A0-A7 นั่นเอง

บทที่ 3 การอินเทอร์รัพต์

3.0 บทนำ

การอินเทอร์รัพต์ คือการขัดจังหวะ โปรแกรมชั่วคราวแล้วมาทำโปรแกรมบริการอินเทอร์รัพต์ (Interrupt Service Routine ; ISR) อินเทอร์รัพต์ของ MCS -51 ได้มาจาก 6 แหล่ง และมีเพียง 5 เวกเตอร์ โดยอินเทอร์รัพต์จากพอร์ตอนุกรมไม่ว่าจะเป็นอินเทอร์รัพต์จากการส่งหรือการรับจะกระโดดไปยังแอดเดรส 0023H ดังรายละเอียดดังตารางที่ 1

ตารางที่ 1

ลำดับ	ชื่อสัญญาณอินเทอร์รัพต์	Vector Address	Priority
1	INT0	0003H	highest ↑ ↓ lowest
2	TF0	000BH	
3	INT1	0013H	
4	TF1	001BH	
5	TL,RI	0023H	

< ถ้ามีอินเทอร์รัพต์เข้ามาพร้อมกับ INT0 จะถือว่ามี Priority สูงสุด >

เราสามารถสั่งห้ามไม่ให้เกิดการขัดจังหวะทั้งหมดได้โดยเพียงแต่ไปรพเช็ทบิต EA Interrupt Enable Register (IE) ถ้าต้องการ Enable อินเทอร์รัพต์จากอุปกรณ์ตัวไหนก็เพียงแต่ไปเช็ทบิตของอุปกรณ์ตัวนั้นไว้แต่อย่าลืมเช็ทบิต EA รายละเอียดของ IE ดังแสดงในรูป 3.1

(MSB)								(LSB)
EA	X	ET2	ES	ET1	EX1	ET0	EX0	
Symbol	Position	Function						
EA	IE.7	disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.						
-	IE.6	reserved						
ET2	IE.5	enables or disables the Timer 2 overflow or capture interrupt. If ET2 = 0, the Timer 2 interrupt is disabled.						
ES	IE.4	enables or disables the Serial Port interrupt. If ES = 0, the Serial Port interrupt is disabled.						
ET1	IE.3	enables or disables the Timer 1 Overflow interrupt. If ET1 = 0, the Timer 1 interrupt is disabled.						
EX1	IE.2	enables or disables External Interrupt 1. If EX1 = 0, External Interrupt 1 is disabled.						
ET0	IE.1	enables or disables the Timer 0 Overflow interrupt. If ET0 = 0, the Timer 0 interrupt is disabled.						
EX0	IE.0	enables or disables External Interrupt 0. If EX0 = 0, External Interrupt 0 is disabled.						

คำอธิบายความหมายในแต่ละบิตใน IE

- EA : 0 จะเป็นการห้ามอินเทอร์รัพต์จากทุกแหล่ง
- 1 ขอมให้มีการอินเทอร์รัพต์ทั้งหมดได้
- ET2: 0 ห้ามอินเทอร์รัพต์จากไทม์เมอร์ 2
- 1 ขอมให้อินเทอร์รัพต์จากไทม์เมอร์ 2

- ES : 0 ห้ามอินเทอร์รัทจากพอร์ตบ
- 1 ขอมให้อินเทอร์รัทจากพอร์ตคอนนุกรม
- ET1: 0 ห้ามอินเทอร์รัทจากไทม์เมอร์ 1
- 1 ขอมให้อินเทอร์รัทจากไทม์เมอร์ 1
- EX1: 0 ห้ามอินเทอร์รัทจากสัญญาณภายนอกชนิดที่ 1
- 1 ขอมให้อินเทอร์รัทจากสัญญาณภายนอกชนิดที่ 1
- ET0: 0 ห้ามอินเทอร์รัทจากไทม์เมอร์ 0
- 1 ขอมให้อินเทอร์รัทจากไทม์เมอร์ 0
- EX0: 0 ห้ามอินเทอร์รัทจากสัญญาณภายนอกชนิดที่ 0
- 1 ขอมให้อินเทอร์รัทจากสัญญาณภายนอกชนิดที่ 0

■ อินเทอร์รัทจากภายนอกของ MCS - 51 ก็คือ EX0 และ EX1 ซึ่งจะตรวจสอบที่ขา *INT0* และ *INT1* ตามลำดับ

■ อินเทอร์รัทภายใน MCS - 51 ได้จากไทม์เมอร์ 0 และไทม์เมอร์ 1 โดยตรวจสอบที่ขา *TF0* และ *TF1* และอินเทอร์รัทจากพอร์ตคอนนุกรม โดยตรวจสอบที่ *TI* และ *RI*

■ ขา *INT0* และ *INT1* ใช้งาน 2 หน้าที่โดยที่เมื่อทำอินเทอร์รัทจะเป็นขา External Interrupt เมื่อไม่ทำอินเทอร์รัทจะใช้ Start ไทม์เมอร์/เคาน์เตอร์ หรือที่เรียกว่า Hardware Start

การขัดจังหวะของการอินเทอร์รัทเราสามารถกำหนดลำดับความสำคัญได้จาก Interrupt Priority Register (IP) ดังรายละเอียดดังรูป 3.2

		(MSB)						(LSB)	
		PCT	X	PT2	PS	PT1	PX1	PT0	PX0
Symbol	Position	Function							
PCT	IP.7	83C154/C154D only. Priority Interrupt circuit control bit. The priority register contents are valid and priority assigned interrupts can be processed when this bit is "0". When the bit is "1", the priority interrupt circuit is stopped, and interrupts can only be controlled by the interrupt enable register (IE).							
-	IP.6	reserved							
PT2	IP.5	defines the Timer 2 interrupt priority level. PT2 = 1 programs it to the higher priority level.							
PS	IP.4	defines the Serial Port interrupt priority level. PS = 1 programs it to the higher priority level.							
PT1	IP.3	defines the Timer 1 interrupt priority level. PT1 = 1 programs it to the higher priority level.							
PX1	IP.2	defines the External Interrupt 1 priority level. PX1 = 1 programs it to the higher priority level.							
PT0	IP.1	defines the Timer 0 interrupt priority level. PT0 = 1 programs it to the higher priority level.							
PX0	IP.0	defines the External Interrupt 0 priority level. PX0 = 1 programs it to the higher priority level.							

รูปที่ 3.2 รายละเอียดของ IP

คำอธิบายความหมายในแต่ละบิตใน (IP)

PT2 : 0 ไทม์เมอร์ 2 มีลำดับความสำคัญต่ำสุด

1 ไทม์เมอร์ 2 มีลำดับความสำคัญสูงสุด

- PS : 0 พอร์คสื่อสารอนุกรมมีลำดับความสำคัญต่ำสุด
- 1 พอร์คสื่อสารอนุกรมมีลำดับความสำคัญสูงสุด
- PT1 : 0 ไทม์เมอร์ 1 มีลำดับความสำคัญต่ำสุด
- 1 ไทม์เมอร์ 1 มีลำดับความสำคัญสูงสุด
- PT0 : 0 ไทม์เมอร์ 0 มีลำดับความสำคัญต่ำสุด
- 1 ไทม์เมอร์ 0 มีลำดับความสำคัญสูงสุด
- PX0 : 0 อินเตอร์รัพทภายนอกชนิด 0 มีลำดับความสำคัญต่ำสุด
- 1 อินเตอร์รัพทภายนอกชนิด 0 มีลำดับความสำคัญสูงสุด

< ถ้าทุกบิตเป็น 1 หมดจะไม่ได้ถือว่าทุกตัวมีลำดับความสำคัญสูงสุดเท่ากัน เมื่อสัญญาณมาพร้อมกันจะต้องตัดสินตามตารางที่ 1 >

Timer / Counter Register (TCON)

อยู่ใน SFR ตำแหน่งที่ (088H) ใช้ทำงานหลายหน้าที่คือ

- TFx ใช้เป็นที่เก็บแฟลคของไทม์เมอร์ 0, 1
- TRx ใช้สตาร์ทไทม์เมอร์ 0, 1
- IEx ใช้เป็นที่เก็บแฟลคของสัญญาณอินเตอร์รัพทภายนอกชนิด 0, 1
- ITx ใช้เลือกลักษณะของการทริกที่ขา INTx

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

รูปที่ 3.3 รายละเอียด TCON

บิต TF1 - บิตแสดงการเกิดโอเวอร์โฟลว์ของไทม์เมอร์ 1 จะเซ็ทเมื่อไทม์เมอร์ 1 โอเวอร์โฟลว์ และจะถูกเคลียเองเมื่อซีพียูย้ายการทำงานไปที่โปรแกรมบริการอินเตอร์รัพ

บิต TR1 - บิตควบคุมการนับของไทม์เมอร์ 1 ควบคุมจากโปรแกรม

ถ้าเป็น 1 ไทม์เมอร์ 1 เริ่มทำงานต่อ

ถ้าเป็น 0 ไทม์เมอร์ 1 หยุดทำงาน

กรณีนี้ต้องเซ็ทบิต GATE 1 ใน TMOD ก่อน

บิต TF0 - บิตแสดงการเกิดโอเวอร์โฟลว์ของไทม์เมอร์ 0 ถูกเซ็ทเมื่อไทม์เมอร์ 0 เกิดโอเวอร์โฟลว์เช่นเดียวกับ TF1

บิต TR0 - เช่นเดียวกับ TR1 แต่ใช้ควบคุมไทม์เมอร์ 0

บิต IE1 - บิตแสดงสัญญาณอินเทอร์รัพทภายนอกหมายเลข 1 เมื่อสัญญาณอินเทอร์รัพเข้ามาที่ขา *INT1* และถูกเคลียเองโดยคำสั่ง RETI ที่อยู่ในโปรแกรมส่วนบริการอินเทอร์รัพ

บิต IT1 - บิตเลือกประเภทการตรวจสอบสัญญาณอินเทอร์รัพที่เข้ามาทางขา *INT1*

ถ้าเป็น 1 จะตรวจสอบการเปลี่ยนระดับสัญญาณจาก 1 เป็น 0 ที่ขา *INT1*

ถ้าเป็น 0 จะตรวจสอบระดับศูนย์ของสัญญาณที่ขา *INT1*

Interrupt Vector ของ MCS - 51

มีอยู่ 5 ตำแหน่งดังแสดงในตารางที่ 2 ดังนี้

ตารางที่ 2

Source	Vector Addrsss
IE0	0003h
TF0	000Bh
IE1	0013h
TF1	001Bh
RI + TI	002Bh



บทที่ 4

ไทม์เมอร์ / เคาน์เตอร์

4.0 ไทม์เมอร์ / เคาน์เตอร์

ไมโครคอนโทรลเลอร์ตระกูล MCS - 51 สามารถเลือกให้มีการทำงานเป็นไทม์เมอร์ หรือ เคาน์เตอร์อย่างใดอย่างหนึ่ง ไทเมอร์ค่าในเรจิสเตอร์จะถูกเพิ่มค่าทุก ๆ เมกซ์ซินไซเคิล 1/12 คาบ เวลาของ OSC เรจิสเตอร์จะถูกเพิ่มค่าทีละหนึ่งเมื่อมีการเปลี่ยนสถานะ ซึ่งตรวจจับได้จากขา T0, T1 อยู่ที่ขา 14 และ 15 ตามลำดับ แล้วแต่จะเลือก สามารถเลือกได้โดยกำหนดค่าบิต CT ในเรจิสเตอร์ใช้งานเฉพาะ TMOD ดังแสดงในรูปที่ 4.1

GATE	When TRx (in TCON) is set and GATE = 1, TIMER/COUNTERx will run only while INTx pin is high (hardware control). When GATE = 0, TIMER/COUNTERx will run only while TRx = 1 (software control).
CT	Timer or Counter selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).
M1	Mode selector bit (NOTE 1)
M0	Mode selector bit (NOTE 1)

NOTE 1 :

M1	M0	Operating Mode
0	0	0 13-bit Timer
0	1	1 16-bit Timer/Counter
1	0	2 8-bit Auto-Reload Timer/Counter
1	1	3 (Timer 0) TLO is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.
1	1	3 (Timer 1) Timer/Counter 1 stopped.

รูปที่ 4.1 แสดงรายละเอียด TMOD

GATE - เมื่อบิต TRx ใน TCON และบิต GATEx ใน TMOD ถูกเซ็ทไทม์เมอร์หรือเคาน์เตอร์ จะเริ่ม RUN ก็ต่อเมื่อ

1. เมื่อขา INTO หรือ INT1 ถูกป้อนด้วย High
2. เมื่อใช้คำสั่งเซ็ทบิต GATE ใน TCON แล้วถ้าเซ็ทบิต TRx จะทำให้ไทม์เมอร์หรือเคาน์เตอร์เริ่ม RUN

CT - บิตเลือกการทำงานของไทม์เมอร์/เคาน์เตอร์โดยเลือกดังนี้
 ถ้า CT=0 เป็นการเลือกโหมดไทม์เมอร์
 ถ้า CT=1 เป็นการเลือกโหมดเคาน์เตอร์

Timer / Counter Register (TCON) อยู่ใน SFR ตำแหน่งที่ (008H)

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0
-----	-----	-----	-----	-----	-----	-----	-----

TF1	TCON.7	Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as processor vectors to the interrupt service routine.
TR1	TCON.6	Timer 1 run control bit. Set/cleared by software to turn Timer/Counter 1 ON/OFF.
TF0	TCON.5	Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as processor vectors to the service routine.
TR0	TCON.4	Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.
IE1	TCON.3	External Interrupt 1 edge flag. Set by hardware when External interrupt edge is detected. Cleared by hardware when interrupt is processed.

รูปที่ 4.2 แสดงรายละเอียด TCON

รายละเอียดได้กล่าวไว้แล้วในบทที่ 3

IE (Interrupt Enable Register)

ใช้ควบคุมอินเทอร์รัพได้ 6 แห่ง คือจาก (TI,RI) , TF0 , TF1 , INT0 , INT1 เราสามารถสั่งห้ามหรือไม่ห้ามการอินเทอร์รัพได้จากเรจิสเตอร์ชุดนี้

(MSB)								(LSB)
EA		X	ET2	ES	ET1	EX1	ET0	EX0
Symbol	Position	Function						
EA	IE.7	disables all Interrupts. If EA=0, no interrupt will be acknowledged. If EA=1 each interrupt source is individually enabled or disabled by setting or clearing its enable bit.						
-	IE.6	reserved						
ET2	IE.5	enables or disables the Timer 2 Overflow or capture Interrupt. If ET2=0, the Timer 2 interrupt is disabled.						
ES	IE.4	enables or disables the Serial Port interrupt. If ES=0, the Serial Port Interrupt is disabled.						
ET1	IE.3	enables or disables the Timer 1 Overflow interrupt. If ET1=0, the Timer 1 interrupt is disabled.						
EX1	IE.2	enables or disables External Interrupt 1. If EX1=0, External Interrupt 1 is disabled.						
ET0	IE.1	enables or disables the Timer 0 Overflow interrupt. If ET0=0, the Timer 0 interrupt is disabled.						
EX0	IE.0	enables or disables External Interrupt 0. If EX0=0, External Interrupt 0 is disabled.						

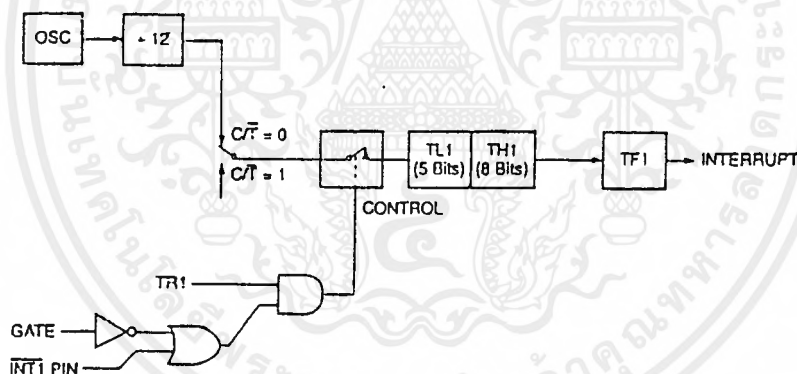
รูปที่ 4.3 แสดงรายละเอียดเรจิสเตอร์ IE

รายละเอียดได้กล่าวไปแล้วในบทที่ 3

4.1 โหมดเมอร์โหมด

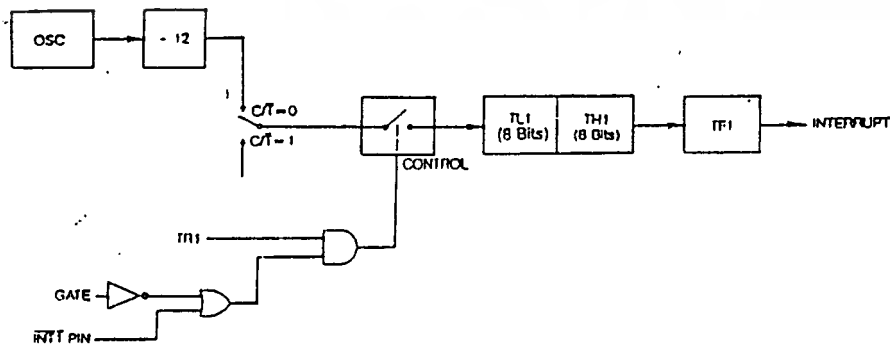
โหมดการทำงานแบ่งได้เป็น 4 โหมด ดังนี้

4.1.1 โหมด 0 โหมดนี้จะนับได้สูงสุดเท่ากับ 2



รูปที่ 4.4 Timer Mode 0

4.1.2 โหมด 1 โหมดนี้เหมือนโหมด 0 ต่างกันที่เรจิสเตอร์เป็นแบบ 16 บิต

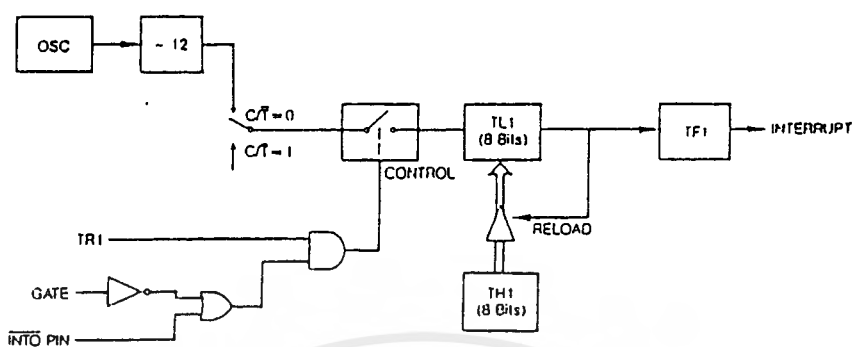


รูปที่ 4.5 Timer Mode 1

4.1.3 โหมด 2 8-bit Auto Reload โหมดนี้ใช้เรจิสเตอร์ TL0 หรือ TL1

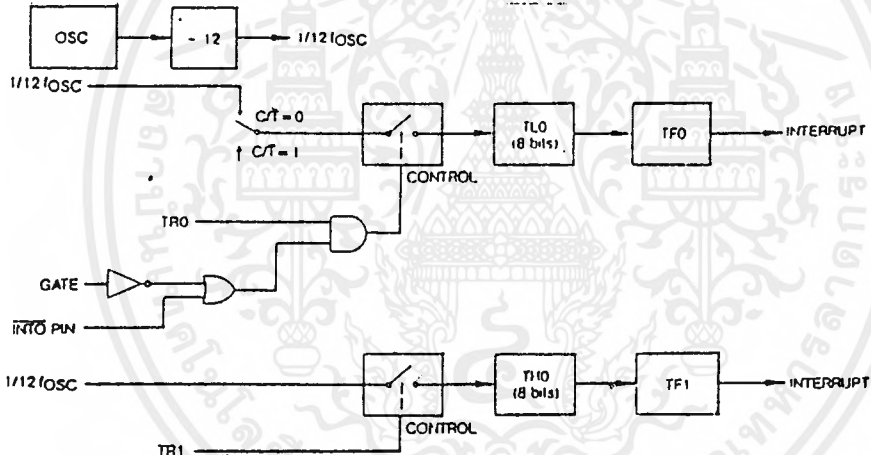
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนค่าที่จะไหลค้เข้ามาจะต้องเก็บไว้ก่อนใน TH0 หรือ TH1 ค่านี้จะถูกไหลค้เข้า TLO, TL1 โดยอัตโนมัติเมื่อเกิดโอเวอร์โฟลว์



รูปที่ 4.6 Timer Mode 2

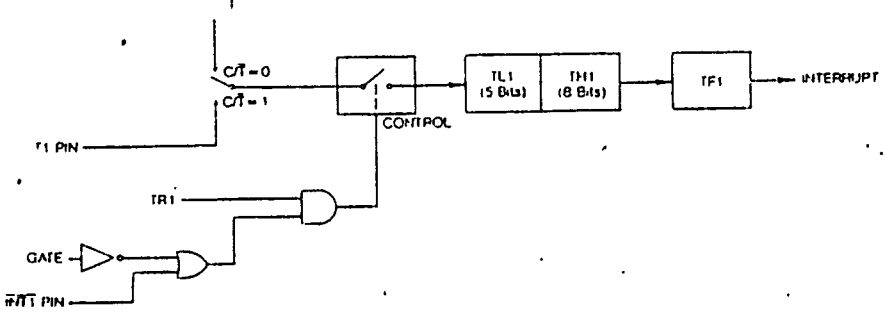
4.1.4 โหมด 3 โหมดนี้จะใช้ TLO ทำงานได้ทั้งโหมดเคาน์เตอร์ และ ไทม์มอร์เมื่อโอเวอร์โฟลว์จะเซ็ทที่ TFO ส่วน TH0 ทำงานโหมดเคาน์เตอร์เมื่อโอเวอร์โฟลว์ จะเซ็ทที่ TF1



รูปที่ 4.7 Timer Mode 3

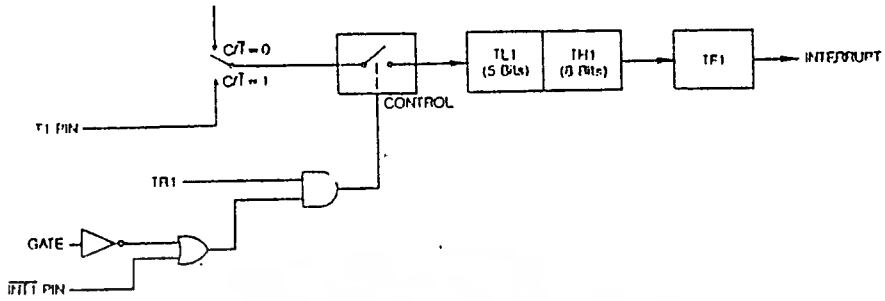
4.2 เคาน์เตอร์โหมด

อินพุตของเคาน์เตอร์โหมดจะเข้ามาทางขา T0, T1 ซึ่งคือพอร์ต 3 บิต 4 และ 5 ตามลำดับ ซึ่งตรงกับขา 14 และ 15 การเลือกโหมดนี้จะต้องโปรแกรมบิต CT = 1 เพื่อเข้าสู่โหมดเคาน์เตอร์ ฟังก์การทำงานดังแสดงในรูป



รูปที่ 4.8 ฟังก์การทำงานของเคาน์เตอร์โหมด

4.2.1 โหมด 0 โหมดนี้ตัวเคาน์เตอร์จะมีเพียง 13 บิต คือ TH = 8 บิต และ TL = 5 บิต โดย TLO นับได้สูงสุดเพียง 32 ดังนั้นโหมดนี้นับสูงสุดเท่ากับ $(256)(32) = 8192$ เท่านั้น

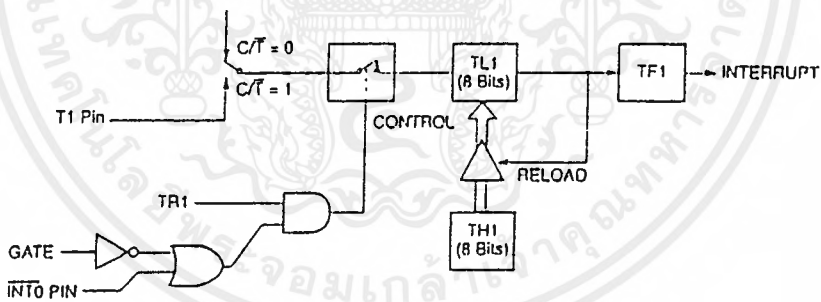


รูปที่ 4.9 Counter Mode 0

4.2.2 โหมด 1 16bit Counter มีการทำงานเหมือนโหมด 0 ต่างกันที่เคาน์เตอร์เป็น 16 บิต โดยแบ่งเป็น TL1 = 8 บิต TH1 = 8 บิต ค่าสูงสุดที่นับได้คือ $(256)(256) = 65536$

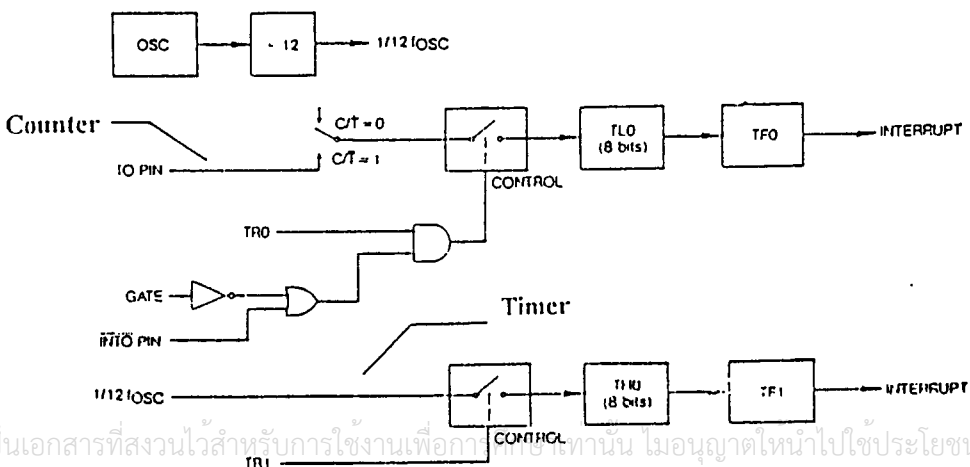
4.2.3 โหมด 2 8bit Auto Reload โหมดนี้จะใช้เคาน์เตอร์ TL1 เป็นตัวนับ นับได้เพียง 8 บิต

เวลาโหลดค่านับจะต้องโหลดเข้า TH1 เท่านั้น โหลดเข้าเพียงครั้งเดียว เมื่อนับครบก็จะเกิดโอเวอร์โฟลว์ทำให้ TF1 ถูกเซ็ท หลังจากนั้นค่าใน TH1 ก็จะถูกโหลดเข้า TL1 อย่างอัตโนมัติ ดังรูป



รูปที่ 4.10 Counter Mode 2

4.2.4 โหมด 3 8 bit Count โหมดนี้จะใช้ TF0 เป็นแฟลทของ TLO ซึ่งเป็นเคาน์เตอร์ 8 บิต



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.11 Counter Mode 3

บทที่ 5

หน่วยแสดงผลข้อมูล (LCD MODULE)

ปัจจุบัน LCD เป็นที่นิยมกันอย่างมาก สำหรับการแสดงผลในเครื่องมือเครื่องใช้ต่างๆ ทั้งนี้เนื่องจากมีความเหมาะสมด้วยประการทั้งปวง ทั้งในด้านการกินกระแสต่ำ สามารถแสดงผลเป็น ตัวอักษร และตัวเลข หรือแสดงเป็นกราฟฟิคได้(เฉพาะรุ่น) จะคิดปัญหาที่คือในด้านวงจร ซึ่งมีระบบการทำงานที่ซับซ้อน และหาอุปกรณ์ได้ค่อนข้างยาก แต่ขณะนี้ผู้ผลิต LCD จะทำรุ่นที่เป็น LCD MODULE ออกมา คือเป็น MODULE ที่มีตัว LCD และวงจรควบคุมให้พร้อม (เรียกว่า LCM) ซึ่งทำให้ ผู้ใช้สามารถต่อเข้ากับระบบไมโคร ใด้ง่ายและสะดวกสำหรับการเขียน โปรแกรม รวมทั้งมีจำหน่ายกัน อย่างกว้างขวาง และมีราคาที่เหมาะสม ทำให้ผู้ใช้งานทางด้าน ไมโครหันมาใช้แผงแสดงด้วย LCD MODULE กันมากขึ้น

LCD MODULE มีอยู่มากมายหลายรุ่น และมีคุณสมบัติแตกต่างกัน ไป ซึ่งแบ่งได้เป็น 2 กลุ่มหลัก คือ แบบ DOT MATRIX และแบบ GRAPHIC โดยแบบ DOT MATRIX จะแสดงผลเป็นตัวอักษร ขนาด 5*8 DOT และมีจำนวนอักษรและบรรทัดแตกต่างกัน ไปในแต่ละรุ่น ส่วนแบบ GRAPHIC จะสามารถแสดงผลในแบบ BIT-MAP คือจะสร้างเป็นภาพใดๆก็ได้ตามต้องการ แนวทางในการใช้งาน ของทั้งสองแบบ จะมีลักษณะใกล้เคียงกัน การใช้งาน โดยทั่วไปมักจะใช้แบบ DOT MATRIX มากกว่า เนื่องจากว่ามีราคาถูก และเพียงพอต่องานส่วนใหญ่ คุณสมบัติของ DOT MATRIX LCD MODULE สามารถสรุปเป็นข้อๆ ได้ดังนี้

1. มีเลือกหลายรุ่นตามการใช้งาน โดยมีจำนวนตัวอักษรและบรรทัดที่แตกต่างกันไป
2. ตัวอักษรแสดงด้วย DOT MATRIX ขนาด 5*8 DOT
3. สามารถต่อเข้ากับระบบไมโครได้ 2 ลักษณะ คือแบบ MEMORY MAP(20 PIN LCD BUS) และแบบผ่าน 8255 PORT (26-PIN 8255 BUS) โดยกรณี 26- PIN 8255 BUS จะใช้แผ่น PCB(DMCAD) เป็นตัว ADAPTER ทำให้เป็น 8255 BUS อีกที
4. การใช้งานง่ายและสะดวก ระบบไมโครเพียงแค่ส่งข้อมูลให้กับ LCD MODULE เท่านั้น ข้อความก็จะปรากฏบนแผงแสดง และค้างไว้ตลอด ทำให้ไม่ต้องเสียเวลาหลักของระบบไมโคร
5. มีคำสั่งพิเศษสำหรับอำนวยความสะดวกมากมาย เช่น CLEAR DISPLAY, HOME CURSOR, ON OFF CURSOR, BLINK CHARACTER และอื่นๆอีก
6. สามารถแสดงผลเป็นตัวอักษรภาษาอังกฤษและตัวเลขได้ 160 ตัว และสัญลักษณ์พิเศษอีก 32 ตัว รวมทั้งสามารถกำหนดอักษรที่ออกแบบเองได้อีก 8 ตัว
7. กินกระแสน้อย และมีน้ำหนักเบา รวมทั้งทำงานด้วยไฟเลี้ยงระดับ 5 V เท่านั้น

5.1 ขาสัญญาณของ LCD MODULE

PIN	SYMBOL	LEVEL	FUNCTION
1	V _{ss}	---	0 V GND
2	V _{cc}	---	+5 V POWER SUPPLY
3	V _{cc}	---	+V FOR LIQUID CRYSTAL DRIVE
4	RS	H/L	REGISTER SELECT H: DATA INPUT L: INSTRUCTION INPUT
5	R/W	H/L	H: DATA READ L: DATA WRITE
6	E	H	ENABLE SIGNAL (L→H)
7	DB 0	H/L	DATA BUS BIT 0
8	DB 1	H/L	DATA BUS BIT 1
9	DB 2	H/L	DATA BUS BIT 2
10	DB 3	H/L	DATA BUS BIT 3
11	DB 4	H/L	DATA BUS BIT 4
12	DB 5	H/L	DATA BUS BIT 5
13	DB 6	H/L	DATA BUS BIT 6
14	DB 7	H/L	DATA BUS BIT 7

5.2 การต่อเข้าระบบไมโคร

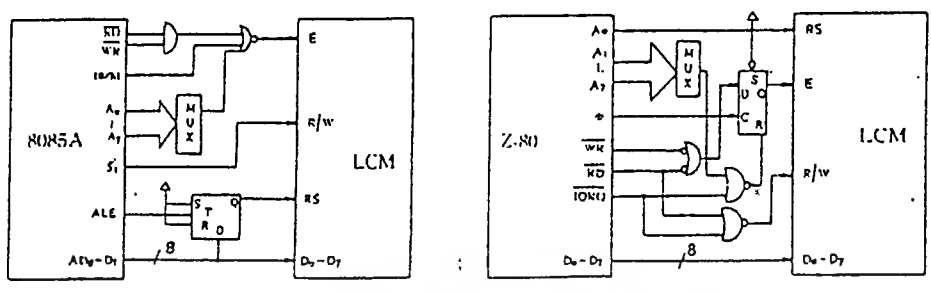
LC MODULE จะต่อเข้ากับระบบไมโครได้ 2 ลักษณะ คือแบบ MEMORY MAP โดยผ่าน LCD BUS ขนาด 20 PIN และแบบ I/O PORT โดยผ่าน 8255 BUS ขนาด 26 PIN ซึ่งทั้งสองนี้จะมีข้อดีข้อเสียแตกต่างกันไป โดยแต่ละแบบมีหลักการดังนี้

การต่อแบบ MEMORY MAP

1. สามารถต่อเข้ากับ CHIP เบอร์ต่างๆ ได้ เช่น 8051 หรือ Z80 โดยจะทำให้ระบบไมโครมองเห็น LCD MODULE ในลักษณะของ MEMORY ได้ทันที
2. ผู้ใช้สามารถเขียนและอ่านข้อมูลจาก LCD MODULE ได้ ทำให้มองเสมือนว่าเป็น MEMORY BUFFER ไปในตัว
3. เนื่องจากสามารถอ่านข้อมูลกลับได้ จึงทำให้สามารถตรวจสอบ FLAG ความพร้อมในขณะที่ LCD MODULE กำลังทำงานได้
4. ใช้ได้กับบอร์ดที่มี LCD BUS มาให้พร้อมเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

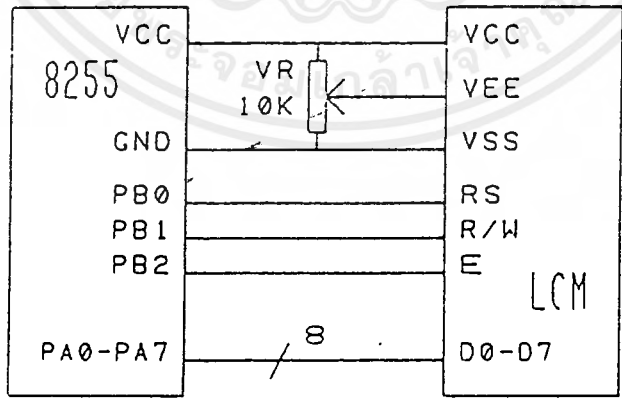
- 5. ทำให้กินพื้นที่หน่วยความจำไปส่วนหนึ่ง และต้องมีการ DECODE ให้ละเอียดพอคว
- 6. การจัดขาสัญญาจะต้องเป็นไปตามแบบของ CHIP แต่ละเบอร์ด้วย!



รูปที่ 5.1 การต่อแบบ MEMORY MAP

การต่อแบบ I/O PORT

- 1. สามารถต่อเข้ากับ I/O PORT ใดๆก็ได้ โดยใช้สายสัญญาณจำนวน 11 เส้น และใช้โปรแกรมเป็นตัวสร้างสัญญาณนี้ขึ้นมาให้ตรงกับข้อกำหนดของ LCD MODULE
- 2. ผู้ใช้จะเขียนข้อมูลให้ LCD MODULE ได้อย่างเดียว ซึ่งผู้ใช้ควรจะกำหนด MEMORY ส่วนหนึ่งให้เป็นเสมือน BUFFER ให้กับ LCD MODULE อีกที
- 3. เนื่องจากไม่สามารถอ่านข้อมูลกลับได้ จึงต้องใช้การหน่วงเวลาของระบบไมโครเองเพื่อรอให้ LCD MODULE กระทำขบวนการต่างๆ
- 4. ใช้ได้กับเบอร์ต่างๆไปที่มีพอร์ต
- 5. ไม่เปลืองส่วนของ MEMORY ในการใช้งาน
- 6. การจัดขาสัญญากระทำได้อิสระ



รูปที่ 5.2 การต่อแบบ I/O PORT

5.3 ชุดตั้งหรืออ่านเขียนข้อมูลและการแสดงข้อความ

การเขียนหรืออ่านข้อมูลกับ LCD MODULE ก็คือการกำหนดคุณสมบัติต่างๆในการใช้งานของ LCD ตามชุดคำสั่งควบคุม และรวมถึงการเขียนข้อมูลที่เป็นข้อความ เพื่อให้ปรากฏ บนแผงแสดงด้วย โดยมีรายละเอียดดังต่อไปนี้

ใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INSTRUCTION	R S	R W	DATA BIT								EXE. TIME ()	
			7	6	5	4	3	2	1	0		
CLEAR DISPLAY	0	0	0	0	0	0	0	0	0	0	1	1640
CURSOR AT HOME	0	0	0	0	0	0	0	0	0	1	#	1640
ENTRY MODE SET	0	0	0	0	0	0	0	1	I/O	S		40
DISPLAY ON/OFF	0	0	0	0	0	0	1	D	C	B		40
DISPLAY SHIFT	0	0	0	0	0	1	S/C	R/L	#	#		40
FUNCTION SET	0	0	0	0	1	DL	N	F	#	#		40
SET CGRAM ADD.	0	0	0	1	CGRAM ADDRESS						40	
SET DDRAM ADD.	0	0	1	DDRAM ADDRESS						40		
BUSY, ADD. READ	0	1	BF	ADDRESS						0		
CGRAM, DDRAM WR	1	0	WRITE DATA						40			
CGRAM, DDRAM RD	1	1	READ DATA						40			

5.4 ความเข้าใจพื้นฐาน

1. การเขียนข้อมูลให้กับ LCD MODULE จะแบ่งเป็น 2 ลักษณะ คือ INSTRUCTION และ DATA โดยจะกำหนดด้วยขาสัญญาณ RS คือถ้า RS = 0 จะหมายถึงส่งสัญญาณควบคุม (INSTRUCTION) หรืออ่านค่า FLAG สภาพการทำงานของ LCD MODULE และถ้า RS = 1 จะหมายถึงการเขียนหรืออ่าน DATA กับ LCD MODULE
2. หลักการในการเขียนข้อมูลให้ LCD MODULE นี้ คือเมื่อมีการเขียนข้อมูลเข้าไปแล้วตัว LCD MODULE จะต้องใช้เวลาในการทำงานชั่วขณะหนึ่ง (ตามค่า EXECUTE TIME ในตาราง) ซึ่งระบบไมโครสามารถตรวจสอบได้จาก BUSY FLAG (BF) และถ้าเรียบร้อยแล้ว จึงสามารถเขียนข้อมูลอันต่อไปได้ ในกรณีที่การต่อวงจรเป็นแบบ I/O PORT คือไม่สามารถอ่านข้อมูลย้อนกลับได้ ระบบไมโครก็จะต้องใช้วิธีหน่วงเวลาแทน
3. การเขียนข้อมูลให้กับ LCD MODULE สามารถทำได้ทั้งแบบ 8 BIT และ 4 BIT โดยกรณี 4 BIT จะใช้ขาสัญญาณ DATA เพียง 4เส้น คือ DB4- DB7 (ใช้สำหรับระบบไมโครแบบ 4 BIT หรือเพื่อการประหยัดสาย) การเขียนข้อมูลจะกระทำเหมือนกับ 8 BIT เพียงแต่ให้เขียน 2 ครั้ง คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DB4- DB7 ก่อน แล้วตามด้วย DB0- DB3 และจะต้องกำหนดคุณสมบัติตามค่า DL ในคำสั่ง FUNCTION SET ด้วย

4. DDRAM (DISPLAY DATA RAM) คือหน่วยความจำภายในตัว LCD MODULE ที่เป็น BUFFER ของข้อมูล โดยถ้าเขียนรหัส ASCII ใดๆลงไปหน่วยความจำนี้ ก็จะปรากฏเป็นตัวอักษรที่เป็นแสงแสดงทันที
5. CGRAM (CHARACTER GENERATOR RAM) คือหน่วยความจำภายในตัว LCD MODULE สำหรับเก็บภาพตัวอักษรที่ผู้ใช้สามารถสร้างได้เอง (8 ตัว) โยจะอ้าง ADDRESS ได้ทั้งหมด 64 BYTE คือ 8 ตัวอักษร คูณกับ 8 ROW

5.5 รายละเอียดของแต่ละคำสั่ง

1. CLEAR DISPLAY

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

สำหรับการ CLEAR DISPLAY โดยจะทำการเขียนตัวอักษร SPACE ลงใน DDRAM ทั้งหมด และกำหนดค่า DDRAM ADDRESS ให้เป็น 0 พร้อมทั้ง CURSOR จะกลับไปตำแหน่งซ้ายสุดของจอภาพ

2. CURSOR AT HOME

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

สำหรับกำหนดค่า DDRAM ADDRESS ให้เป็น 0 พร้อมทั้ง CURSOR จะไปอยู่ที่ตำแหน่งซ้ายบนสุดของจอภาพ โดยที่ข้อมูลใน DDRAM ไม่มีเปลี่ยนแปลง

3. ENTRY MODE SET

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	I/D

I/D = 0 กำหนดทิศทางของ CURSOR และ DDRAM ให้เป็นแบบ DECREMENT

I/D = 1 กำหนดทิศทางของ CURSOR และ DDRAM ให้เป็นแบบ INCREMENT

S = 0 เมื่อเขียนข้อมูลแล้ว ตัว CURSOR จะถูกไปทิศทางตามค่า I/D

S = 1 เมื่อเขียนข้อมูลแล้ว ตัว CURSOR จะอยู่กับที่ และตัวอักษรจะถูกดันไปทิศทางตามค่า I/D

การกำหนด I/D และ S นี้ ให้กำหนดก่อนการเขียนข้อมูลใน DDRAM และเมื่อกำหนดแล้ว จะต้อง

ไม่ใช่คำสั่ง CLEAR DISPLAY อีก

4. DISPLAY ON/OFF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
0	0	0	0	0	0	0	0	1	C	B

D = 0 กำหนดให้ OFF DISPLAY

D = 1 กำหนดให้ ON DISPLAY

C = 0 กำหนดให้ OFF CURSOR

C = 1 กำหนดให้ ON CURSOR โดย CURSOR จะเป็นเส้นขีดได้ตัวอักษร

B = 0 กำหนดให้ไม่มีการกระพริบที่ตำแหน่ง CURSOR

B = 1 กำหนดให้มีการกระพริบที่ตำแหน่ง CURSOR

5. DISPLAY SHIFT

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0	
0	0	0	0	0	0	1	S/C	R/L	*	*

S/C = 0 กำหนดให้เลื่อน CURSOR ตามทิศทาง R/L ไป 1 ตำแหน่ง

S/C = 1 กำหนดให้เลื่อนข้อความแฉงแสดงตามทิศทาง R/L ไป 1 COLUMN

R/L = 0 กำหนดให้ทิศทางไปทางซ้าย

R/L = 1 กำหนดให้ทิศทางไปทางขวา

6. FUNCTION SET

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	*	*

DL = 0 กำหนดให้ติดต่อกับ LCD MODULE เป็นแบบ 4 BIT

DL = 1 กำหนดให้ติดต่อกับ LCD MODULE เป็นแบบ 8 BIT

จะสังเกตว่า การกำหนดค่า DL นี้ สามารถกระทำได้ที่ DB4- DB7 ซึ่งถ้ามีการกำหนดให้เป็นแบบ 4 BIT ตั้งแต่แรก หลังจากจ่ายไฟเลี้ยง ก็จะทำให้ LCD MODULE มีการรับข้อมูลแบบ 4 BIT ทันที

N = 0 กำหนดจำนวนบรรทัดแบบ 1/8 DUTY และ 1/11 DUTY

N = 1 กำหนดจำนวนบรรทัดแบบ 1/16 DUTY

F = 0 กำหนดให้เป็นตัวอักษรแบบ 5*7 DOTS

F = 1 กำหนดให้เป็นตัวอักษรแบบ 5*10 DOTS (กรณีที LCD MODULE เป็นแบบ 5*7 อยู่แล้ว ก็จะไม่มีการผลอะไร)

7. SET CGRAM ADDRESS

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	1	CGRAM ADDRESS					

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการกำหนด ADDRESS ของ CGRAM เมื่อได้ทำการกำหนดไว้แล้ว การอ่านและเขียน DATA ที่
 ค่อยจากนี้ จะเป็นไปตาม ADDRESS ที่กำหนดทันที

8. SET DDRAM ADDRESS

RS R/W DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

0	0	1	DDRAM ADDRESS												
---	---	---	---------------	--	--	--	--	--	--	--	--	--	--	--	--

สำหรับการกำหนด ADDRESS ของ DDRAM เมื่อได้ทำการกำหนดไว้แล้ว การอ่านและเขียน DATA ที่
 ค่อยจากนี้ จะเป็นไปตาม ADDRESS ในแต่ละรุ่นจะมีความแตกต่างกันบ้าง เพราะจำนวนอักษรต่อบรรทัด
 ไม่เท่ากัน ซึ่งแสดงดังภาพต่อไปนี้

รุ่น DMC 202

00	01	02	03	04	05	06	07	08	09	0	0	0	0	0	0F	-10	11	12	13
										A	B	C	D	E					
40	41	42	43	44	45	46	47	48	49	4	4	4	4	4	4F	50	51	52	53
										A	B	C	D	E					

รุ่น DMC164

00	01	02	03	04	05	06	07	08	09	0	0	0	0	0	0F				
										A	B	C	D	E					
40	41	42	43	44	45	46	47	48	49	4	4	4	4	4	4F				
										A	B	C	D	E					
10	11	12	13	14	15	16	17	18	19	1	1	1	1	1	1F				
										A	B	C	D	E					
50	51	52	53	54	55	56	57	58	59	5	5	5	5	5	5F				
										A	B	C	D	E					

9. BUS FLAG AND ADDRESS READ

RS R/W DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

0	1	BF	ADDRESS												
---	---	----	---------	--	--	--	--	--	--	--	--	--	--	--	--

สำหรับการอ่านค่า BF (BUSY FLAGE) ซึ่งบอกถึงความพร้อมของ LCD MODULE ในการรับข้อมูล ถ้า
 BF = 0 หมายถึงว่าพร้อม ที่จะรับข้อมูลต่อไปได้ แต่ถ้า BF = 1 หมายถึงว่ายังไม่พร้อม นอกจากนี้ยังเป็น
 การอ่าน ADDRESS ของ CGRAM หรือ DDRAM ค้วย

5.6 การอ่านและเขียนข้อมูลกับ DDRAM / CGRAM

1. WRITE DATA TO DDRAM OR CGRAM

RS R/W DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

I	0	DATA
---	---	------

สำหรับการเขียนข้อมูลลงหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้ว ADDRESS จะถูกเพิ่มหรือลดลงโดยอัตโนมัติ ตามที่กำหนดจากค่า I/D ในคำสั่ง ENTRY MODE SET และการเขียนจะเป็น DDRAM หรือ CGRAM ก็ขึ้นอยู่กับว่า ก่อนหน้าคำสั่งนี้ มีการกำหนด ADDRESS ที่ใด

2. READ DATA FROM DDRAM OR CGRAM

RS R/W DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0

1	1	DATA
---	---	------

สำหรับการอ่านข้อมูลจากหน่วยความจำ DDRAM หรือ CGRAM โดยเมื่อทำการเขียนแล้ว ADDRESS จะถูกเพิ่มหรือลดโดยอัตโนมัติ ตามที่กำหนดจากค่า I/D ในคำสั่ง ENTRY MODE SET และการอ่านจะเป็น DDRAM หรือ CGRAM ก็ขึ้นอยู่กับว่า ก่อนหน้าคำสั่งนี้ มีการกำหนด ADDRESS ที่ใด

5.7 แนวทางการเขียนโปรแกรมควบคุม

- เมื่อจ่ายไฟเลี้ยงให้กับ LCD MODULE ครั้งแรก ภายในจะมีการ RESET ระบบโดยอัตโนมัติ ซึ่งจะใช้เวลา 10 ms หลังจากที่ระดับแรงไฟขึ้นถึง 4.5 V แล้ว ทั้งนี้ ระบบ RESET ดังกล่าวจะกระทำสิ่งต่างๆดังต่อไปนี้

- ทำการ CLEAR จอภาพ ทั้งหมด
- กำหนดคุณสมบัติด้วยคำสั่ง FUNCTION SET คือ DL = 1 (ติดต่อกับระบบไมโครในแบบ 8 BIT), N = 0 (แสดงผล 1 บรรทัด), F = 0 (กำหนดตัวอักษรแบบ 5*7 DOT)
- กำหนดคุณสมบัติด้วยคำสั่ง DISPLAY ON/OFF คือ D = 0 (ไม่แสดงข้อมูล), C = 0 (CURSOR OFF), B = 0 (BLANK OFF)
- กำหนดคุณสมบัติด้วยคำสั่ง ENTRY MODE SET คือ I/D = 1 (INCREMENT), S = 0 (NO SHIFT)

การใช้งาน LCD MODULE ต้องรอให้ขบวนการ RESET ภายในทำงานเรียบร้อยก่อนซึ่งจะตรวจสอบได้ด้วย BF (BUSY FLAGE) หรืออาจจะใช้การหน่วงเวลาก็ได้

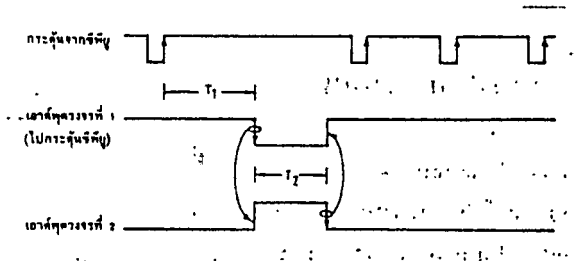
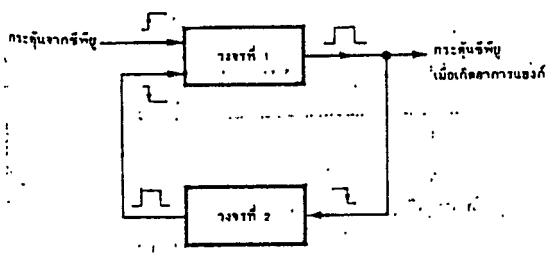
2. การใช้งาน LCD MODULE จะเกี่ยวข้องกับทางด้านโปรแกรมเป็นส่วนใหญ่ ชุดคำสั่งต่างๆรวมทั้งการอ่านหรือเขียนข้อมูลนั้น จะถูกกำหนดด้วยขาสัญญาทั้งหมดที่มีอยู่ปกติโปรแกรมจะต้องกำหนดคุณสมบัติต่างๆที่ต้องการไว้ที่ส่วนต้น และจากนั้นก็จะเป็นการอ่านหรือเขียนข้อมูลลงใน DDRAM ซึ่งก็คือข้อความที่จะให้แสดงนั่นเอง

บทที่ 6

โครงสร้างและหลักการทำงาน

1. แหล่งจ่ายไฟ (POWER SUPPLY) วงจรอิเล็กทรอนิกส์จำเป็นต้องมีแรงเคลื่อนไฟฟ้าสำหรับเลี้ยงวงจร ซึ่งแรงเคลื่อนไฟฟ้าที่ว่านี้ก็คือแรงเคลื่อนไฟฟ้ากระแสตรง ซึ่งสามารถหาได้จากแบตเตอรี่หรือถ่านไฟฉาย แต่ถ่านไฟฉายมีขีดจำกัดในเรื่องของพลังงาน เมื่อเราใช้พลังงานภายในหมดแล้ว เราจำเป็นต้องชาร์จหรือไม่กี่ต้องจัดหาใหม่ ซึ่งจะไม่สะดวกในการนำไปใช้งาน เนื่องจากว่าไดโอดยอมให้กระแสไหลผ่านได้ทางเดียว ดังนั้นไดโอดจึงมีบทบาทเป็นอย่างมาก สำหรับการเรียงกระแสไฟฟ้าจากไฟบ้าน เอ.ซี 220 โวลต์ ให้เป็นกระแสตรงตามที่เรากำหนดได้ โดยวงจรที่ง่ายที่สุดสำหรับการนี้คือ วงจรเร็คติฟาย โดยเมื่อใส่ค่า C ฟิวเตอร์ เข้าไปในวงจร C ฟิวเตอร์ก็จะกรองกระแส ดี.ซีที่ไม่เรียบให้เรียบต่อไป ซึ่งแหล่งจ่ายไฟที่นำมาใช้ในวงจรนี้เป็นวงจรเร็คติฟายแบบบริดจ์ที่ทำหน้าที่จ่ายไฟเลี้ยงให้กับระบบต่างๆ ในวงจร
2. วงจรตรวจสอบ BLANK ใช้ในการตรวจสอบ BLANK ที่ว่างของเทปหรือสัญญาณของเสียงที่ขาดหายไป โดยสัญญาณเสียงที่เข้ามาจะไม่คงที่เปลี่ยนแปลงอยู่เรื่อยๆ เพราะฉะนั้นจึงได้มีการนำเอาค่าความต้านทานปรับค่าได้ มาใช้ หลังจากที่เราใส่สัญญาณเสียงเข้ามาก็จะนำมาขยายผ่าน ออปแอมป์เบอร์ LM386 แลทางเอาต์พุตของออปแอมป์ ก็จะมี C และ R ต่ออยู่ เพื่อจะทำการกรองสัญญาณให้เรียบขึ้น และหน่วงเวลาในการทำงานของ BLANK และต่อเข้ากับทรานซิสเตอร์ เบอร์ BC549 โดยทรานซิสเตอร์จะทำหน้าที่เหมือนกับสวิทช์ ตัวหนึ่ง โดยเป็นทรานซิสเตอร์ ชนิด NPN โดยถ้ามีสัญญาณเสียงส่งเข้ามา ทรานซิสเตอร์ก็จะทำหน้าที่เหมือนเป็นสวิทช์ตัวหนึ่ง โดยสัญญาณจะเข้ามาที่ขา 4 ของ ฟลิปฟลอป ซึ่งเป็น IC ของเบอร์ 7474 ถ้ามีสัญญาณเสียงป้อนเข้ามาที่ขา 4 มันก็จะส่งค่าออกทาง Q เป็น 0 เพื่อบอกให้ไมโคร โปรเซสเซอร์ ว่ามีสัญญาณเสียงเข้ามาโดยจะใช้ การตรวจสอบโดยต่อ LED เข้าที่ขาที่ต่ออยู่ระหว่าง ขา Q ของฟลิปฟลอป กับ ตัว ไมโคร โปรเซสเซอร์ 89C51 โดยถ้ามีสัญญาณเสียงเข้ามา หลอดไฟ LED ก็จะติดสว่างขึ้นมา แต่ถ้ามีตำแหน่ง BLANK คือ ไม่มีสัญญาณเข้ามา ความจุไฟฟ้าและค่าความต้านทานที่ต่ออยู่ที่เอาต์พุต ของออปแอมป์ และ ทรานซิสเตอร์ ก็จะทำหน้าที่เหมือนสวิทช์ กัน ไม่ให้ส่งสัญญาณ ไปที่ขา ที่ 4 ของฟลิปฟลอป ทำให้หลอด LED ไม่ทำงาน
3. วงจรตรวจสอบสัญญาณป้อนเข้า จะทำหน้าที่คอยตรวจสอบสัญญาณที่ป้อนเข้ามาทาง คีย์บอร์ด หรือสวิทช์บนจอ LCD โดยจะใช้ฟลิปฟลอป ซึ่งเป็น IC เบอร์ 7474 ในการ ตรวจสอบสัญญาณ โดยเมื่อเราทำการกดคีย์บอร์ด หรือ ป้อนข้อมูลบนสวิทช์ที่อยู่ในวงจร LCD มันก็จะทำการส่งสัญญาณออกไปทาง ขา Q ไปให้ทรานซิสเตอร์ ทรานซิสเตอร์ ก็จะทำหน้าที่เหมือน เป็นสวิทช์ขยายสัญญาณเพื่อที่จะส่งไปให้ลำโพงส่งเสียงออกมาอีกต่อไป
4. LCD จะเป็นหน่วยแสดงผล ของค่าข้อมูลหรือค่าของโปรแกรมหลังจากที่เราทำการป้อนข้อมูล หรือ เปลี่ยนค่าข้อมูล ตามที่เราต้องการ ซึ่งก็ขึ้นอยู่กับการออกแบบของโปรแกรม การเขียน โปรแกรมที่เราต้องการนำมาใช้งาน เช่น นำมาใช้ในการควบคุม เทป ระบบไฟฟ้า เป็นต้น

5. วงจรป้องกันกันคอมพิวเตอร์แสงก์



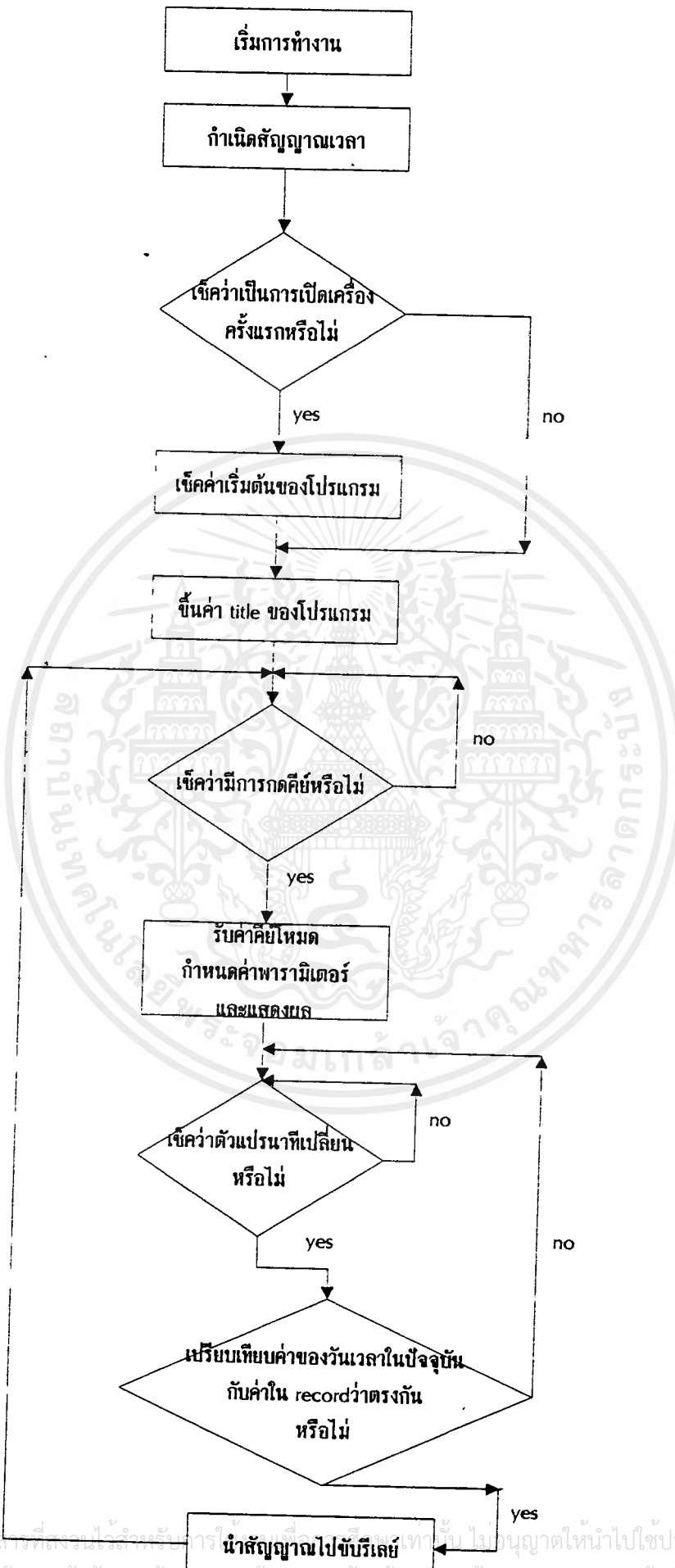
รูปที่ 1 บล็อกโคอะแกรมส่วนอินพุตและเอาต์พุตของวงจรวอลซ์-คอก

วงจรป้องกันคอมพิวเตอร์แสงก์ ที่นำมาใช้ คือ วงจรวอลซ์ - ค็อก

ก่อนอื่น ต้องเข้าใจหลักการทำงานของวงจรก่อนคือ วงจรวอลซ์-คอก นี้เป็นวงจรที่จะต้องได้รับสัญญาณกระตุ้นจาก CPU ตลอดเวลา ภายในช่วงเวลาที่มีการกำหนดไว้ โปรแกรมนี้ส่วนใหญ่จะเขียนแทรกอยู่ในโปรแกรมหลัก หรือส่วนของโปรแกรมที่มีการเรียกใช้เป็นประจำ ซึ่งช่วงเวลาที่มีการวนกลับมาทำโปรแกรมนี้ ก็หมายความว่า CPU ยังไม่หลุดออกจากโปรแกรม แต่ถ้า CPU ไม่สามารถกลับมาทำโปรแกรมในเวลาที่กำหนด แสดงว่า CPU อยู่ในสภาวะแสงก์ ในตอนนี้วงจร วอลซ์-คอก จะต้องไปกระตุ้นให้ CPU กลับมาทำงานเหมือนเดิม (สัญญาณที่ต่อ ไปกระตุ้นให้ CPU กลับมาทำงานตามเดิมนี้นี้ส่วนใหญ่จะนำไปต่อกับบารีเซค CPU)

วงจรวอลซ์-คอก ที่ออกแบบในตอนนี้ จะกำหนดไว้ว่า หาก CPU เกิดอาการแสงก์แล้ว วงจรวอลซ์-คอก จะต้องไปกระตุ้น CPU เป็นช่วงๆ ตลอดเวลาจนกว่า CPU จะส่งสัญญาณกระตุ้นกลับมาที่วงจรวอลซ์-คอก และอีกอย่างหนึ่ง เมื่อเปิดเครื่องครั้งแรกวงจรวอลซ์คอกจะต้องไปกระตุ้นให้ CPU ทำงานทันที โดยมีรูปโครงสร้างของระบบการทำงานดังรูป

ไฟล์วีซีอาร์ทีแสดงการทำงานของโปรแกรมหลัก



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานภายในหน่วยงานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟล์วาร์ทแสดงการเช็ค record

โหลดค่าจำนวน record ทั้งหมด
จากตัวแปร ondata แล้วเช็ค
cnt_chk = 0

โหลดข้อมูลใน record โดยใช้ cnt_chk เป็นตัวชี้ตำแหน่งข้อมูล

ทำการเช็คข้อมูลใน record กับเวลาจริง

ข้อมูลใน record
ตรงกับเวลาจริงหรือไม่

yes
ทำงานตามที่กำหนดไว้

no

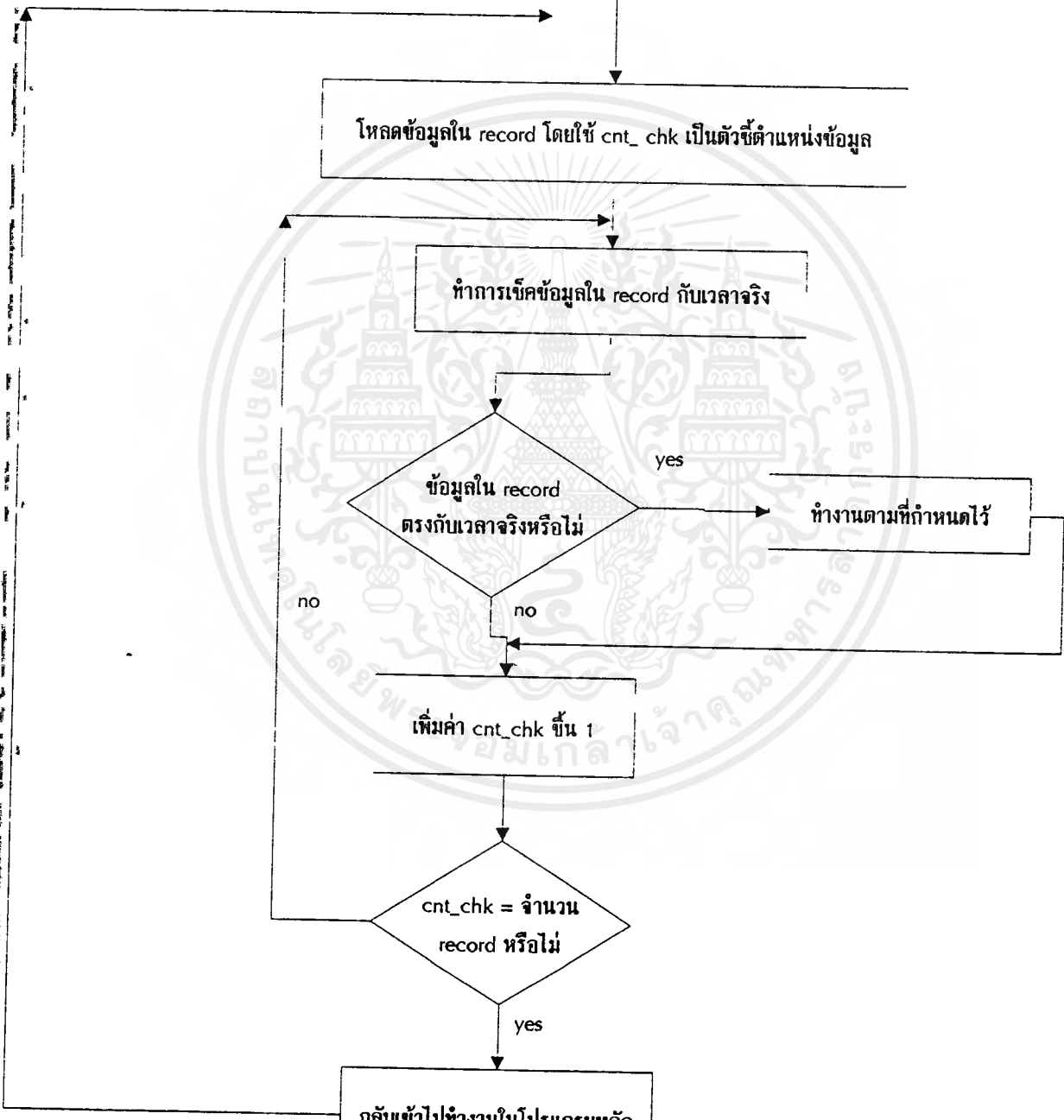
no

เพิ่มค่า cnt_chk ขึ้น 1

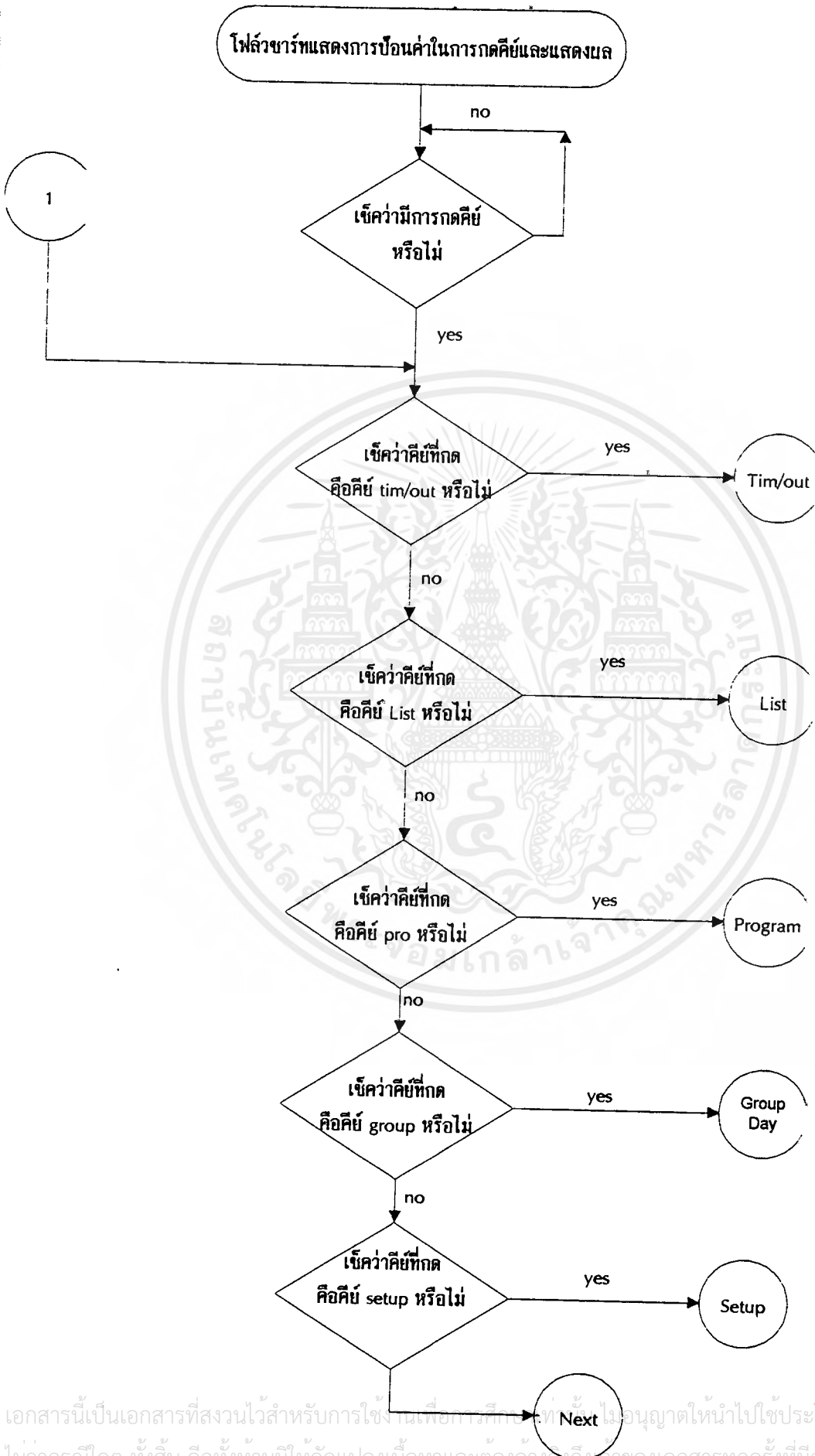
cnt_chk = จำนวน
record หรือไม่

yes

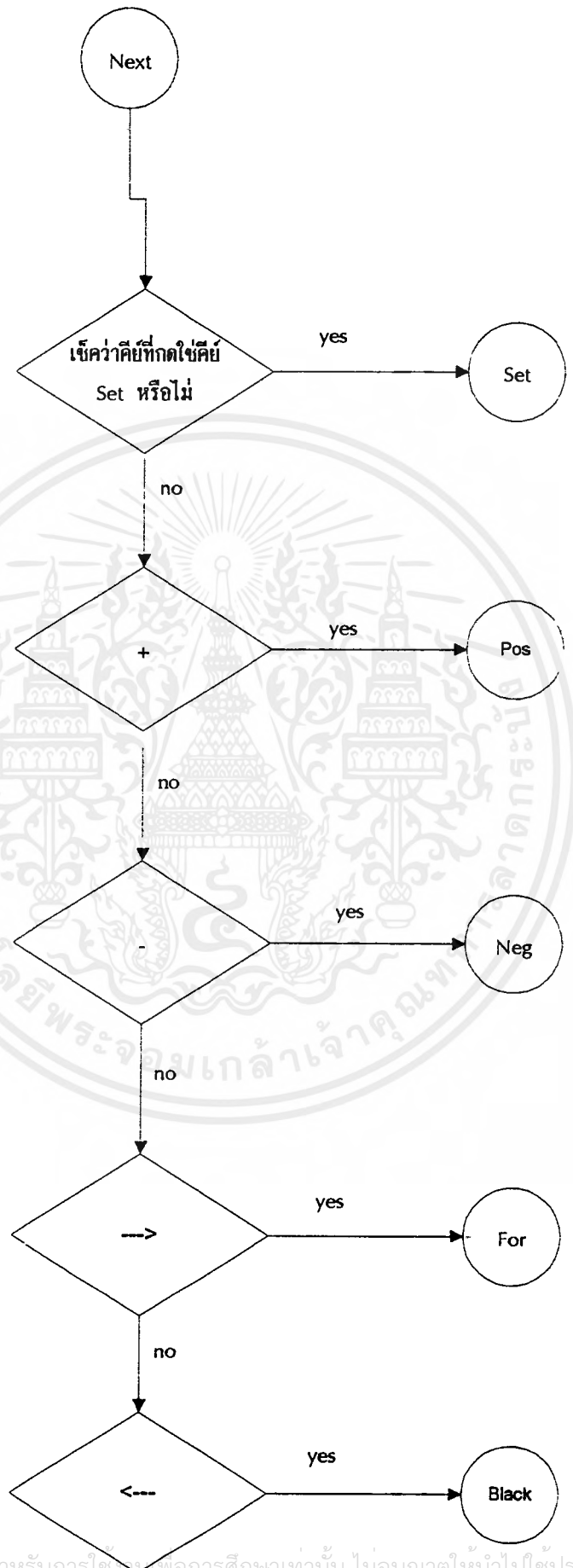
กลับเข้าไปทำงานในโปรแกรมหลัก



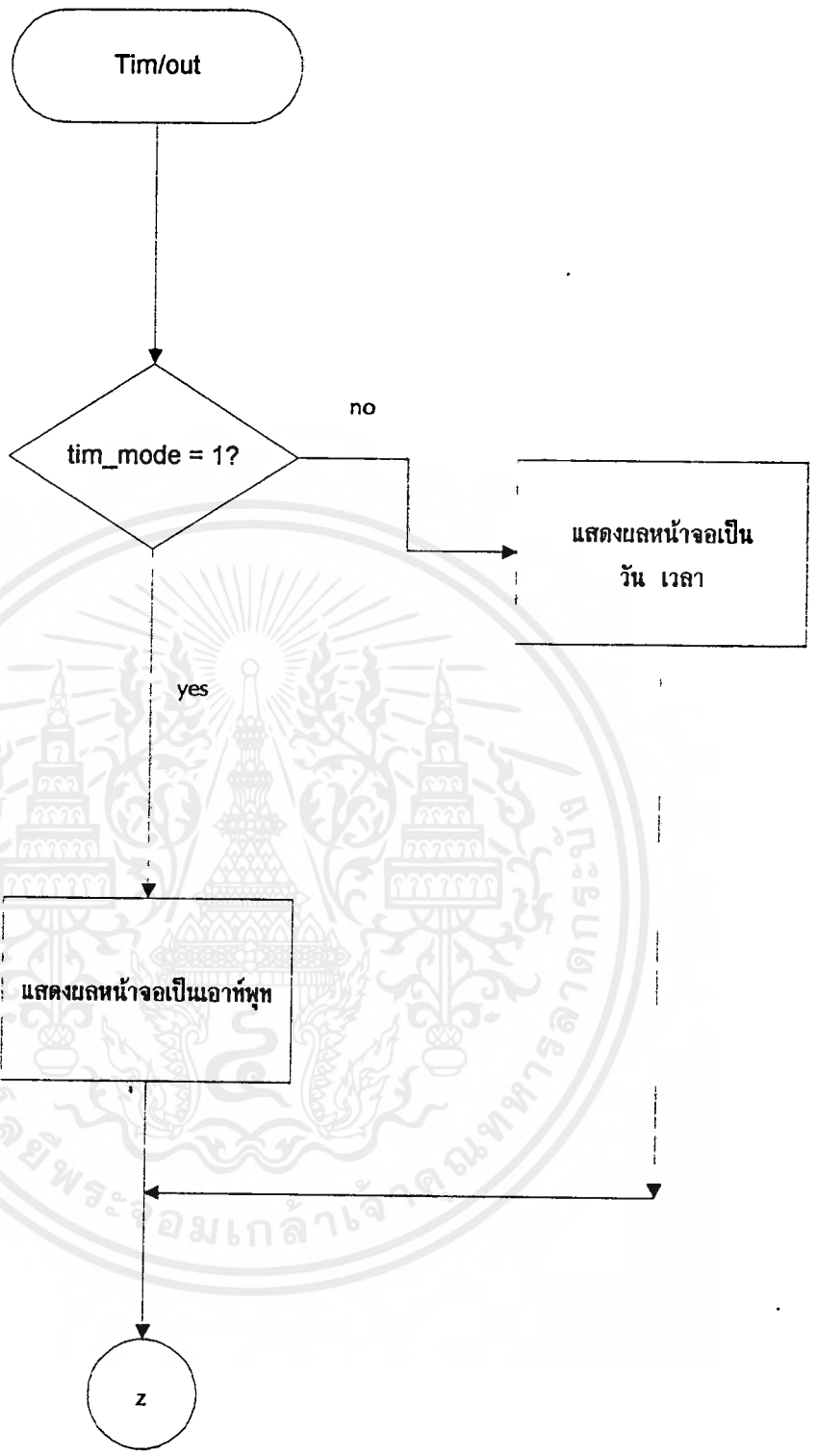
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



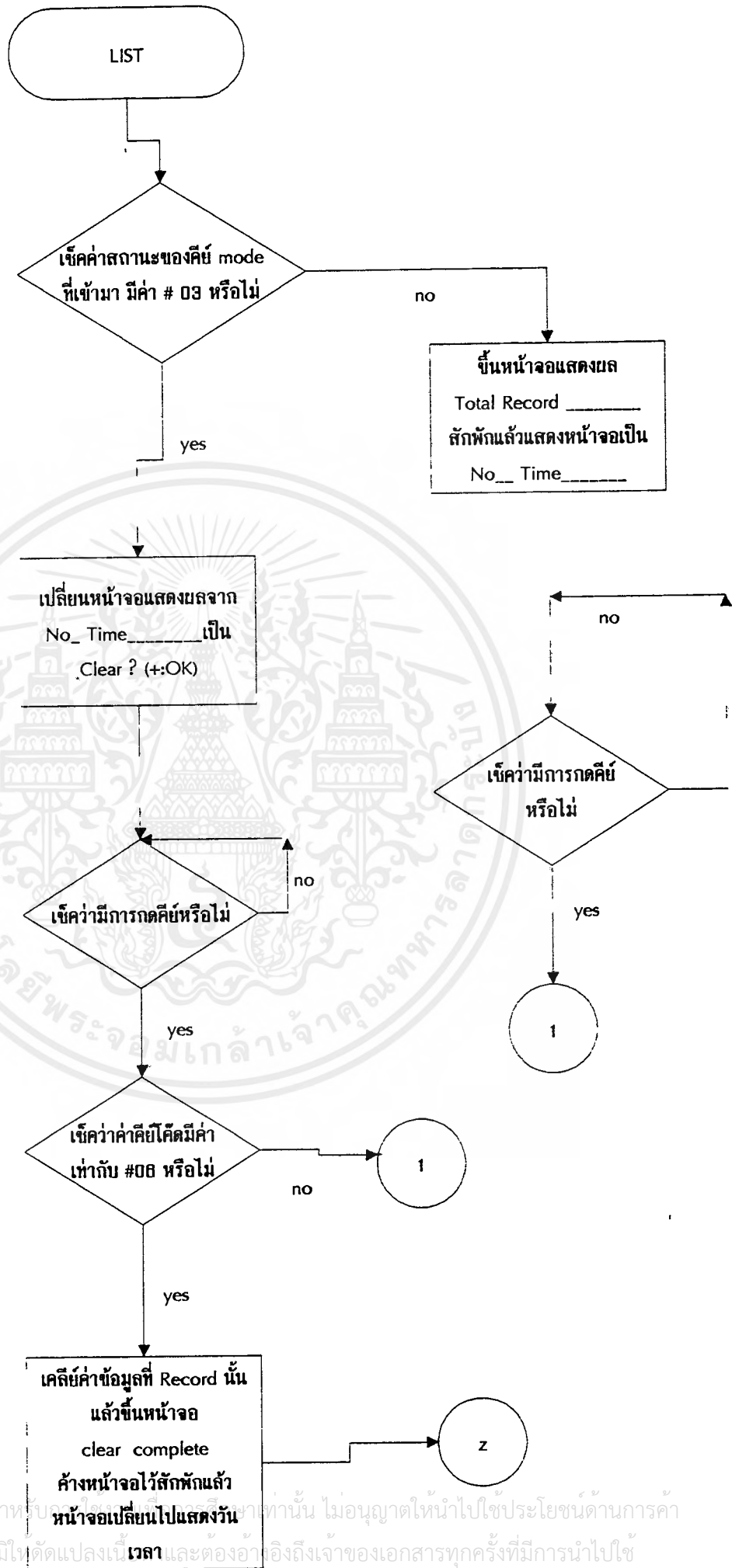
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานที่ถูกต้องเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



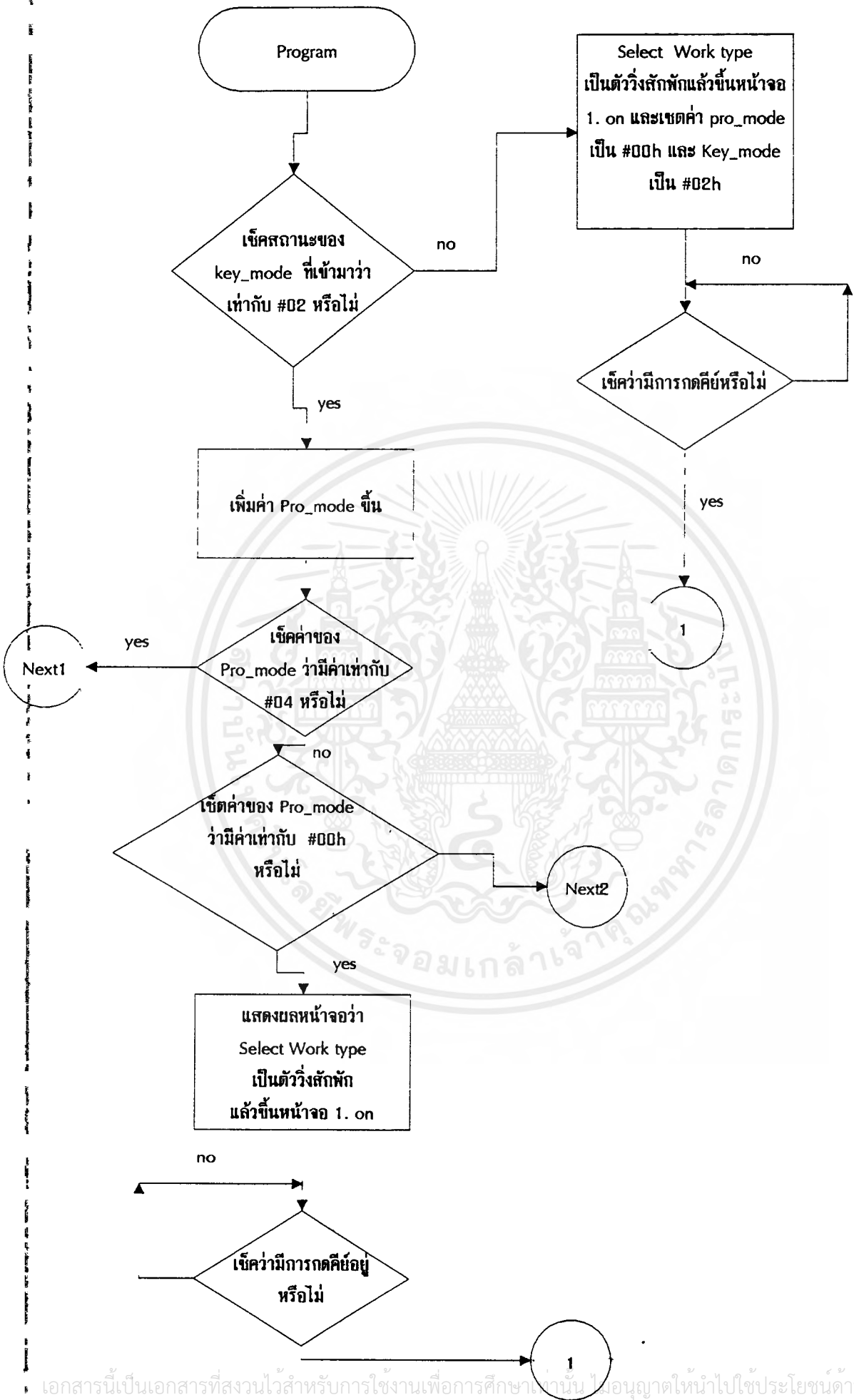
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ของเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับคุณเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

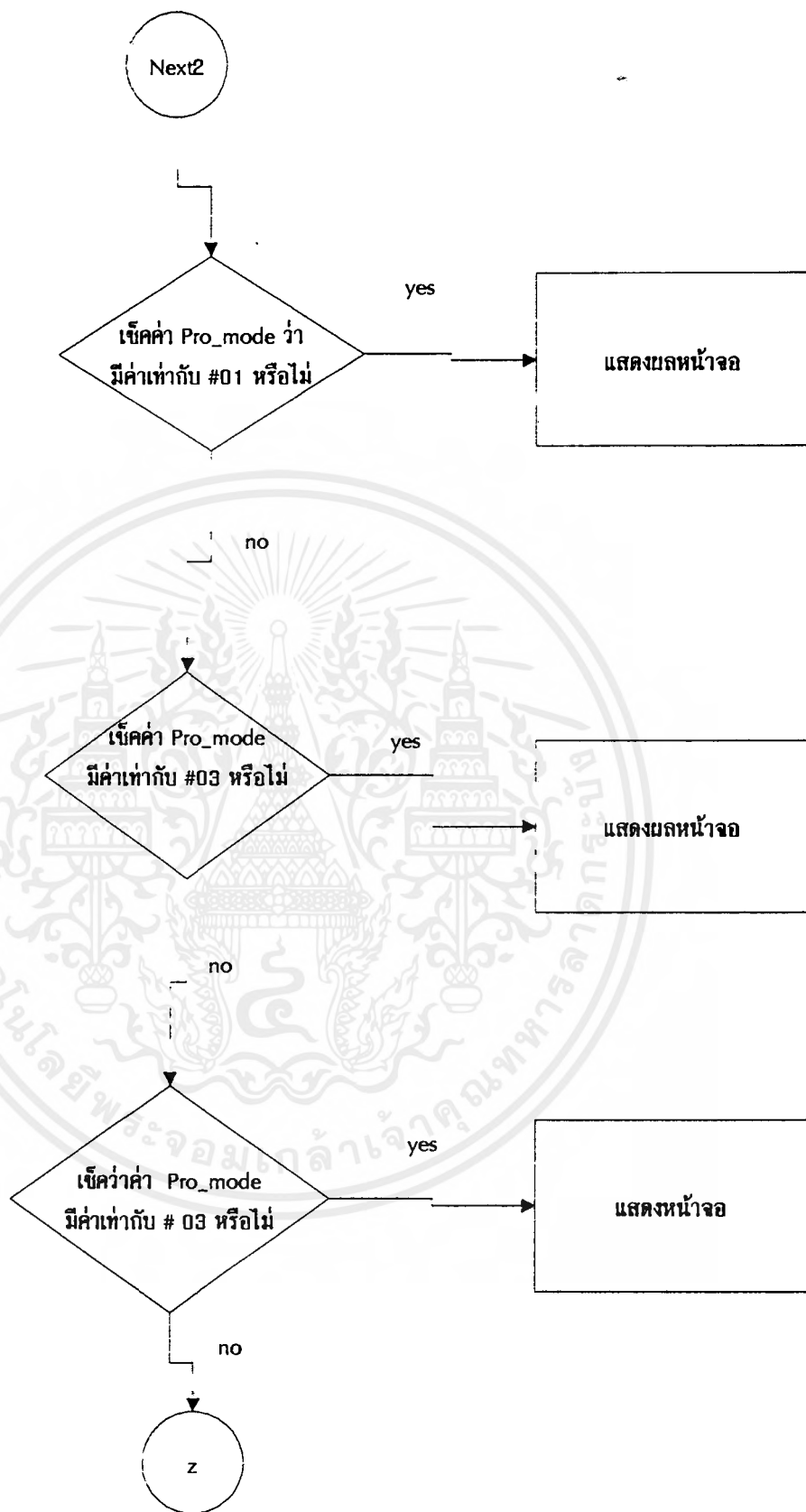
Next1

เพิ่มค่าข้อมูลใน #cnt 'data
ขึ้น 1 แล้ว Set ค่า Ket_mode
ให้เท่ากับ #00h และ
pro_mode =00h

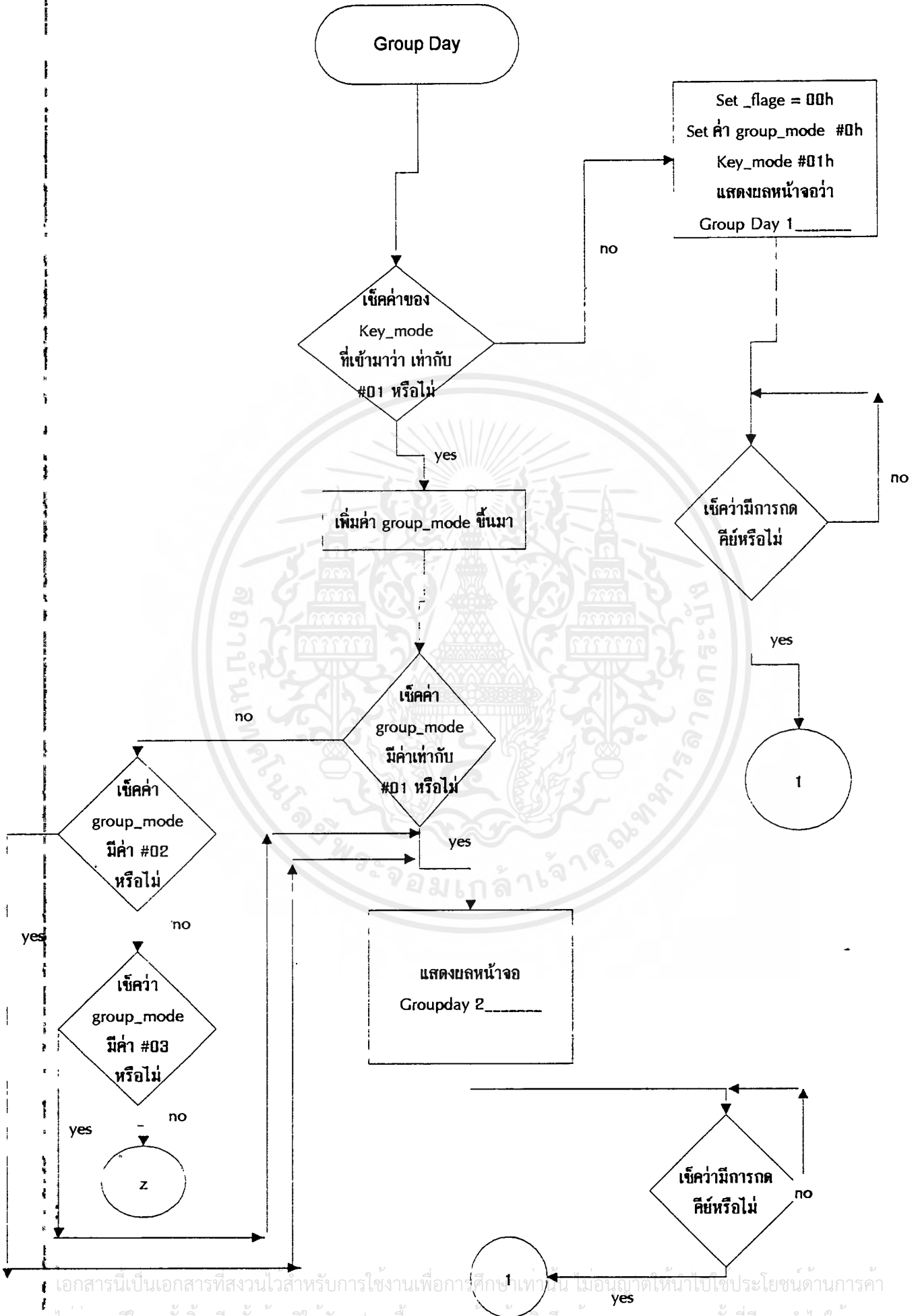
เช็คว่าใน cnt data ว่า
มีค่า = #40
หรือไม่

yes
แสดงผลหน้าจอว่า
" Error" สักครั้งแล้วขึ้นหน้าจอ
แสดงวันเวลา

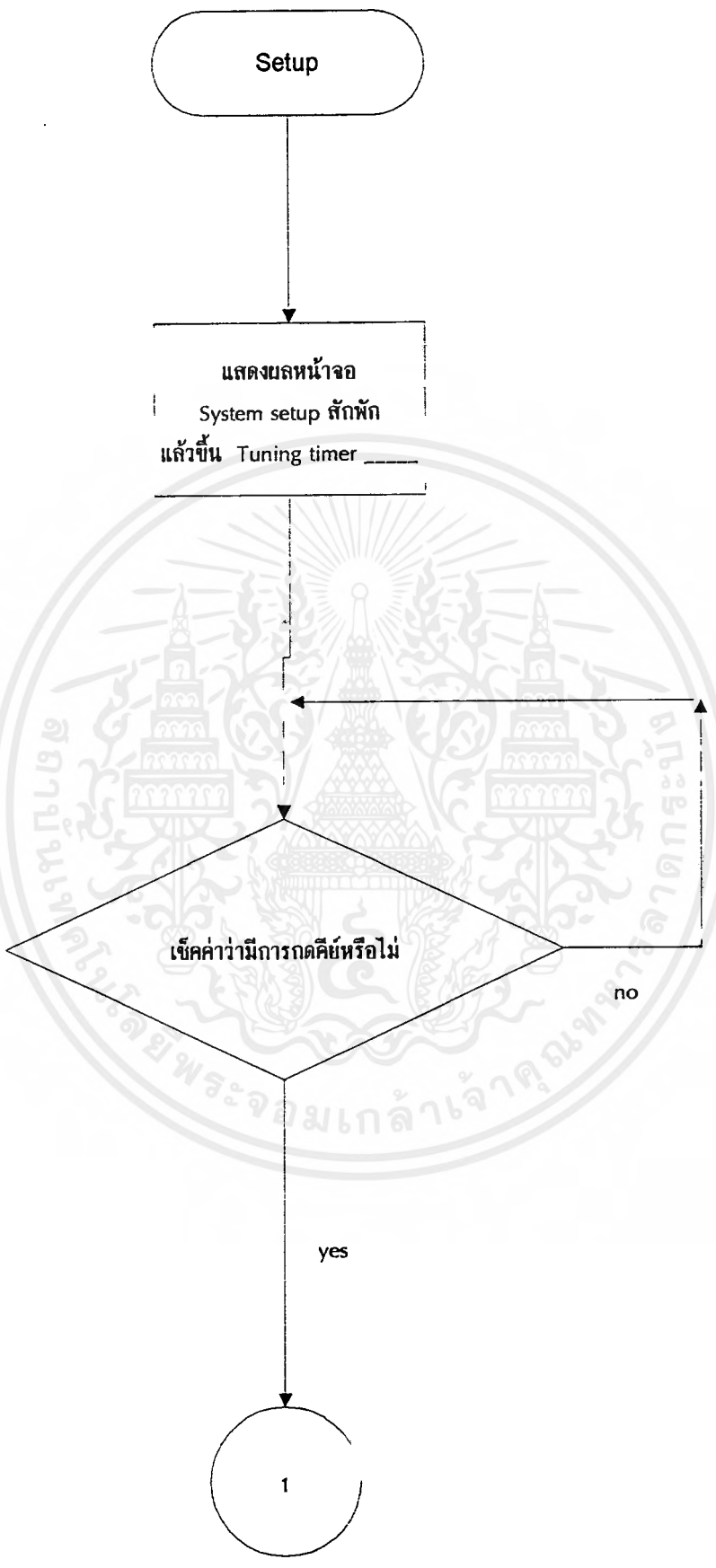
no
เก็บค่าข้อมูล วัน_เวลา_type
Group day o/p ไว้ใน
CG หรือใน DD Ram
แล้วแสดงผลหน้าจอว่า
<<<< O.K. >>>>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

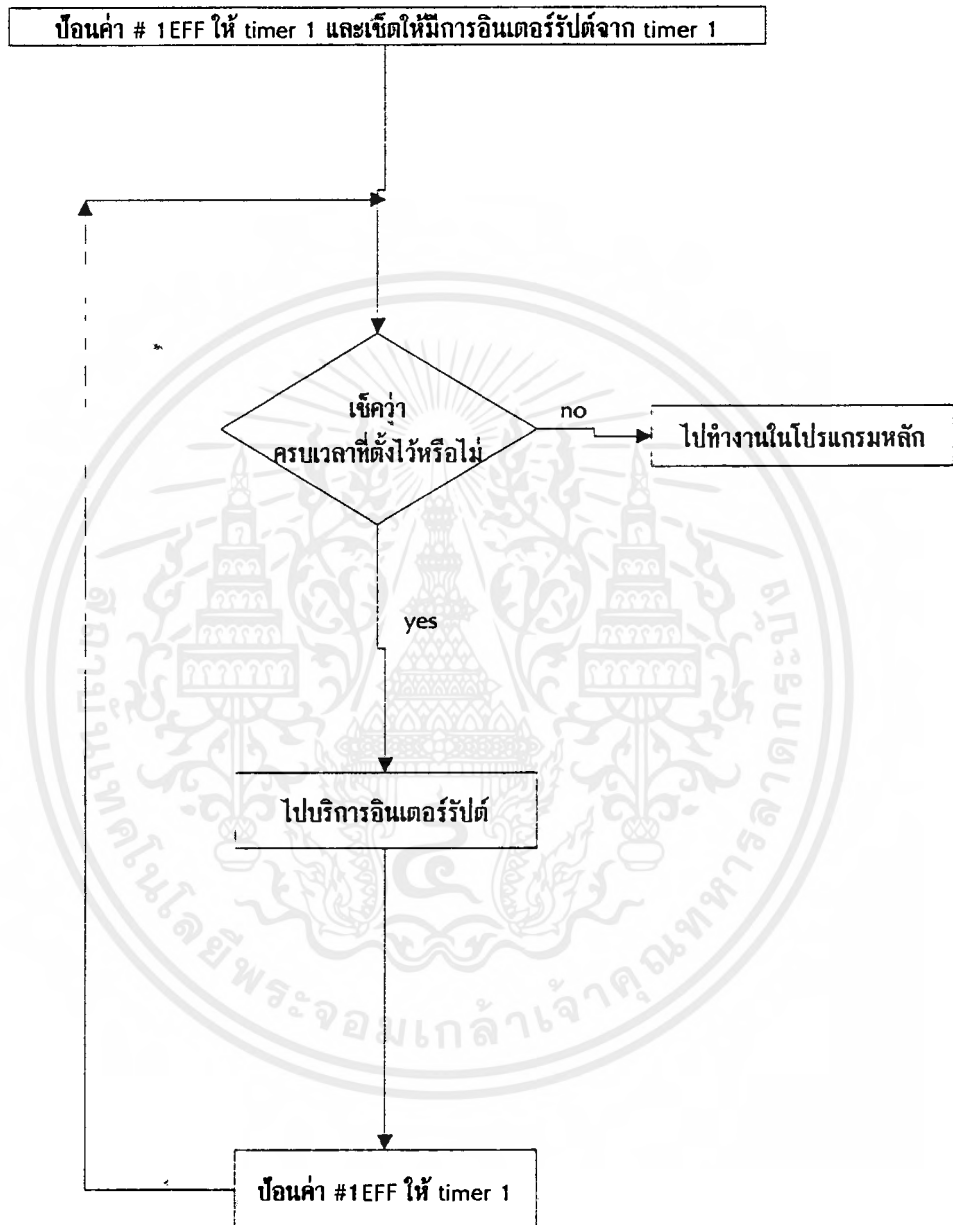


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อแปลให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

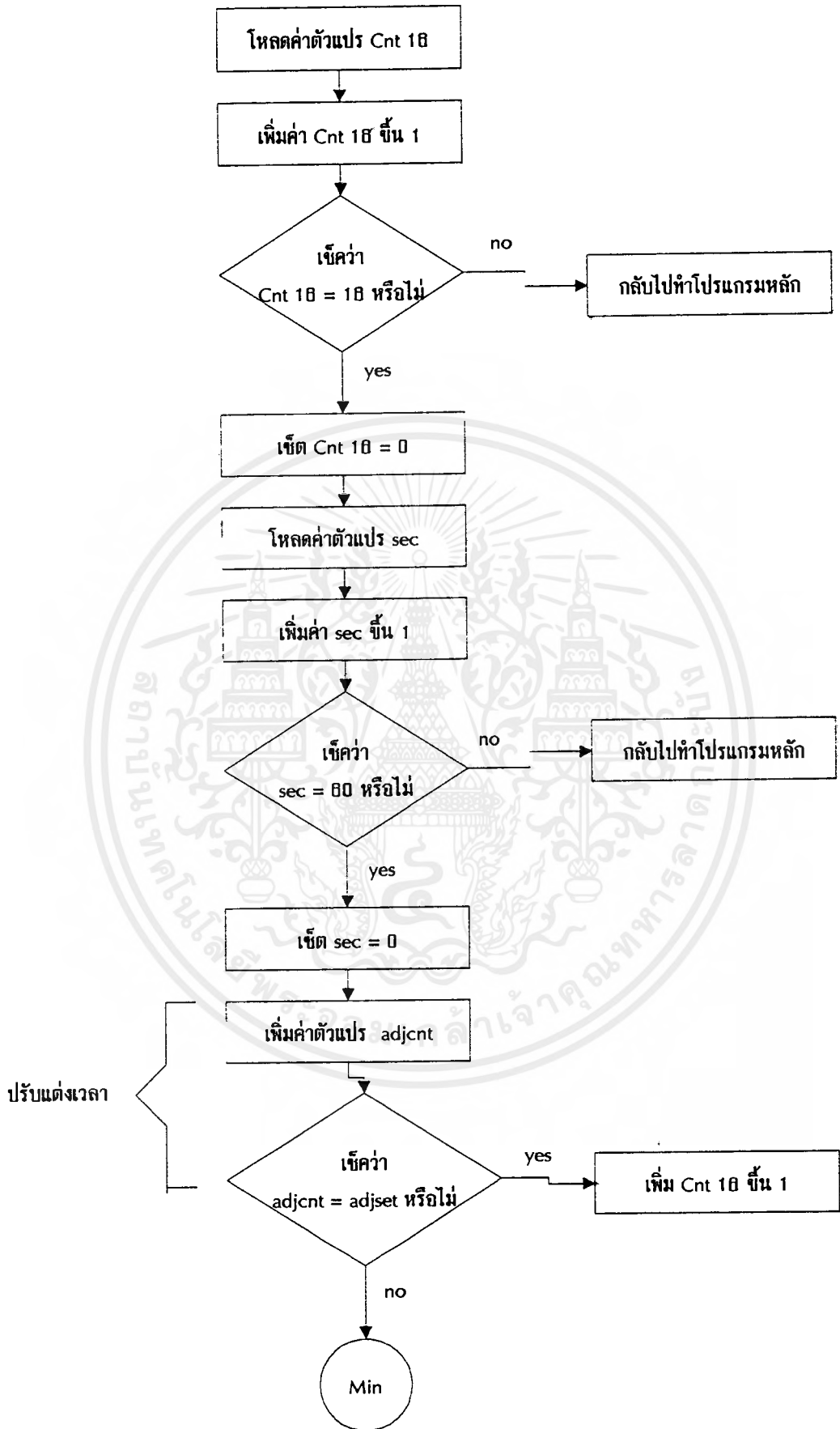


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

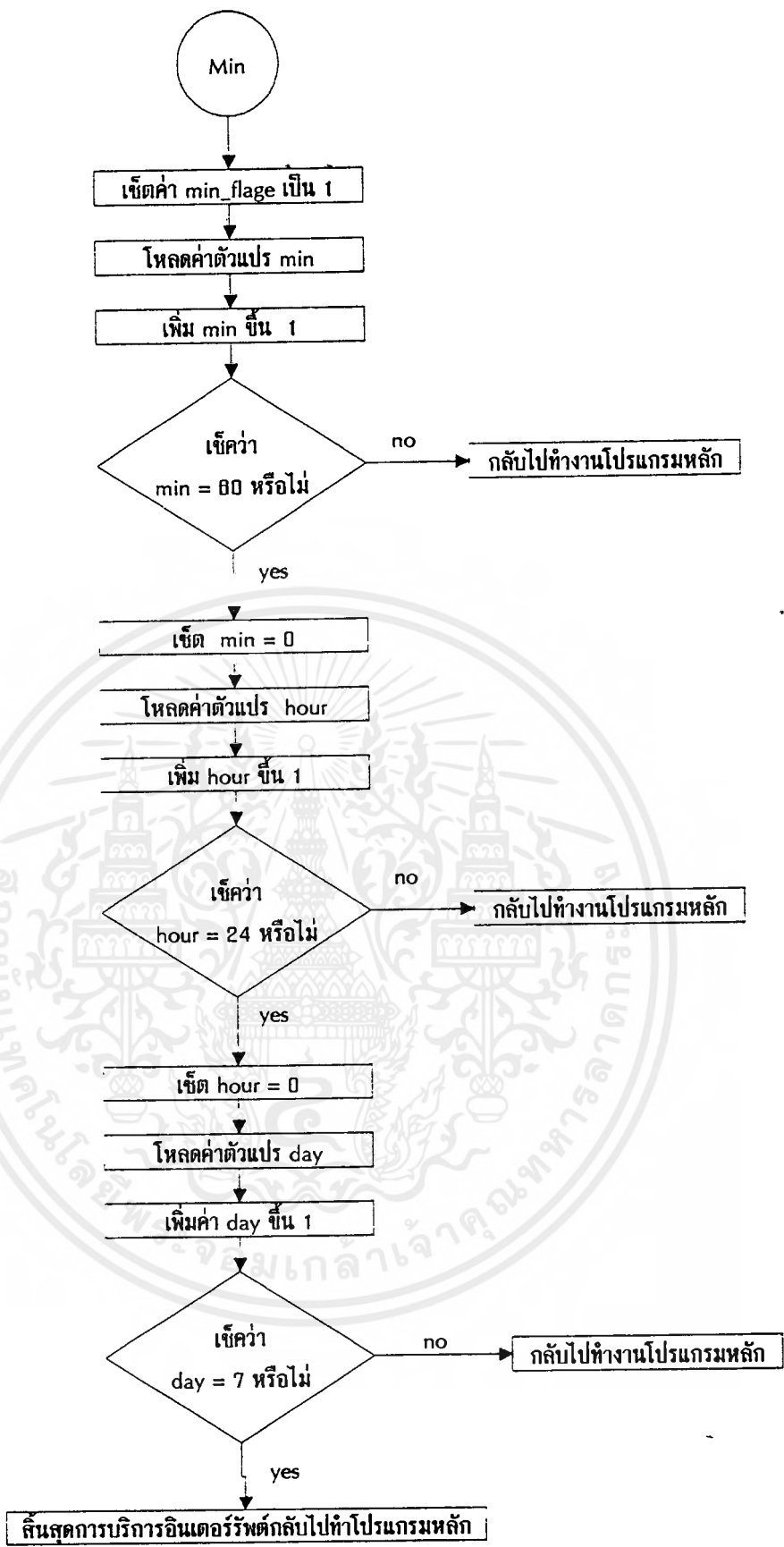
โพลีวาร์ทแสดงการกำเนิดสัญญาณเวลาของระบบจาก timer ใน CPU



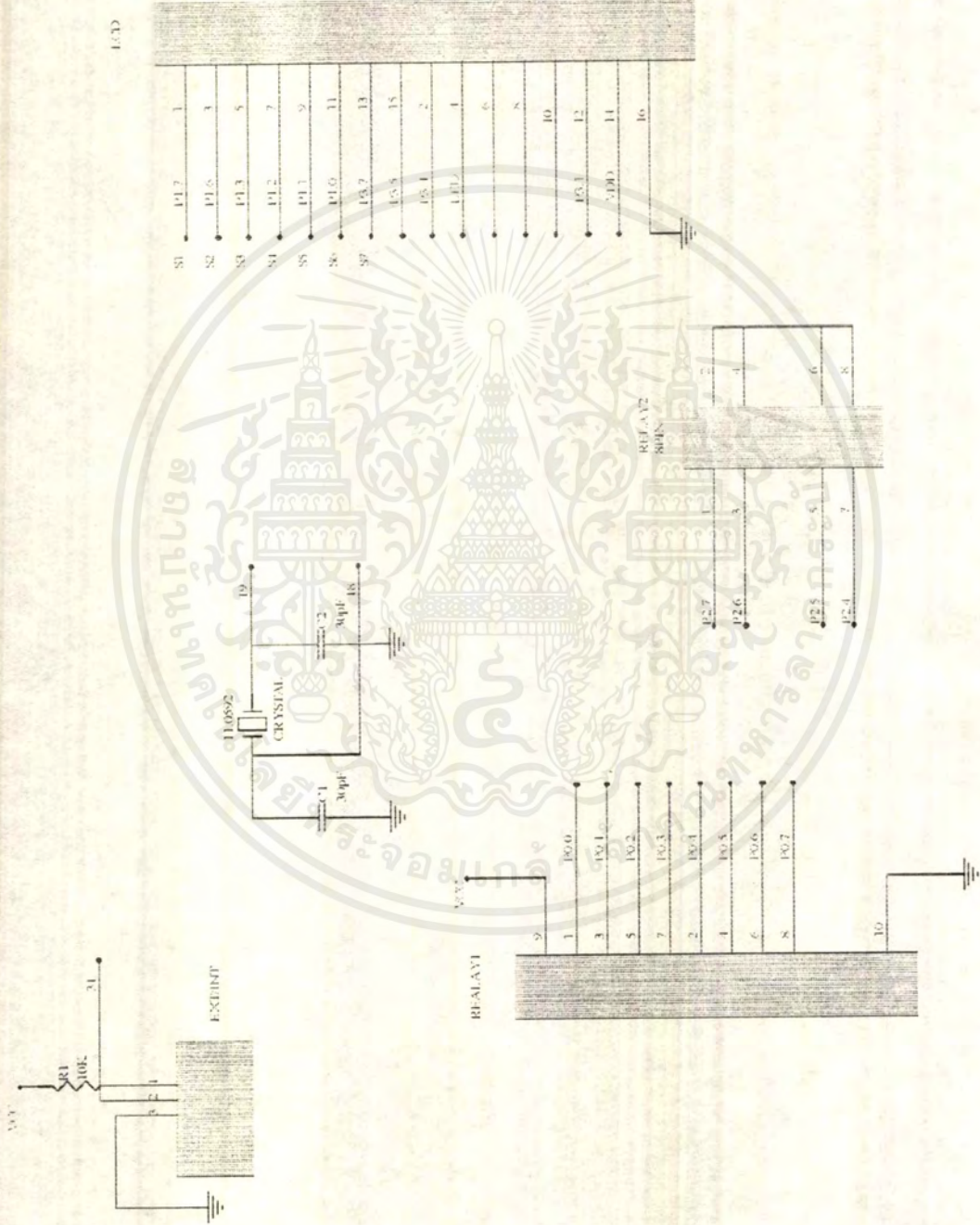
ไฟล์ชาร์ตแสดงการบริการอินเทอร์เน็ต



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

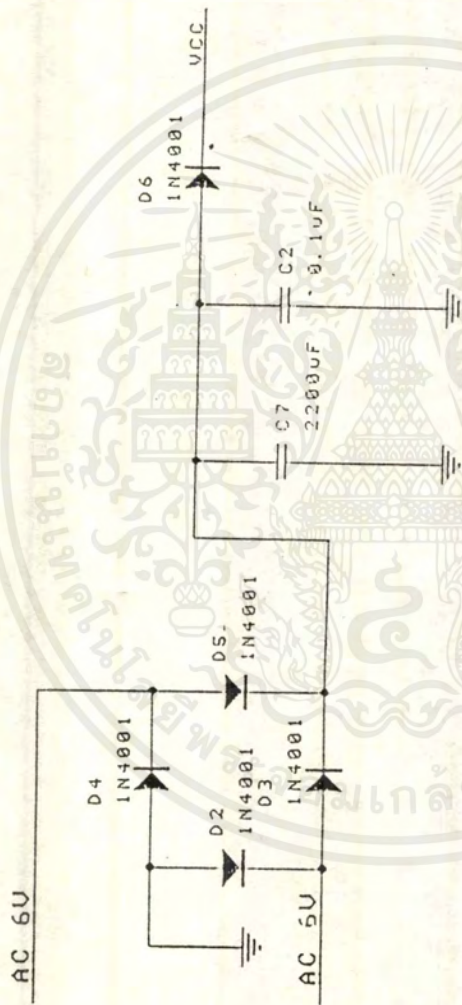


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



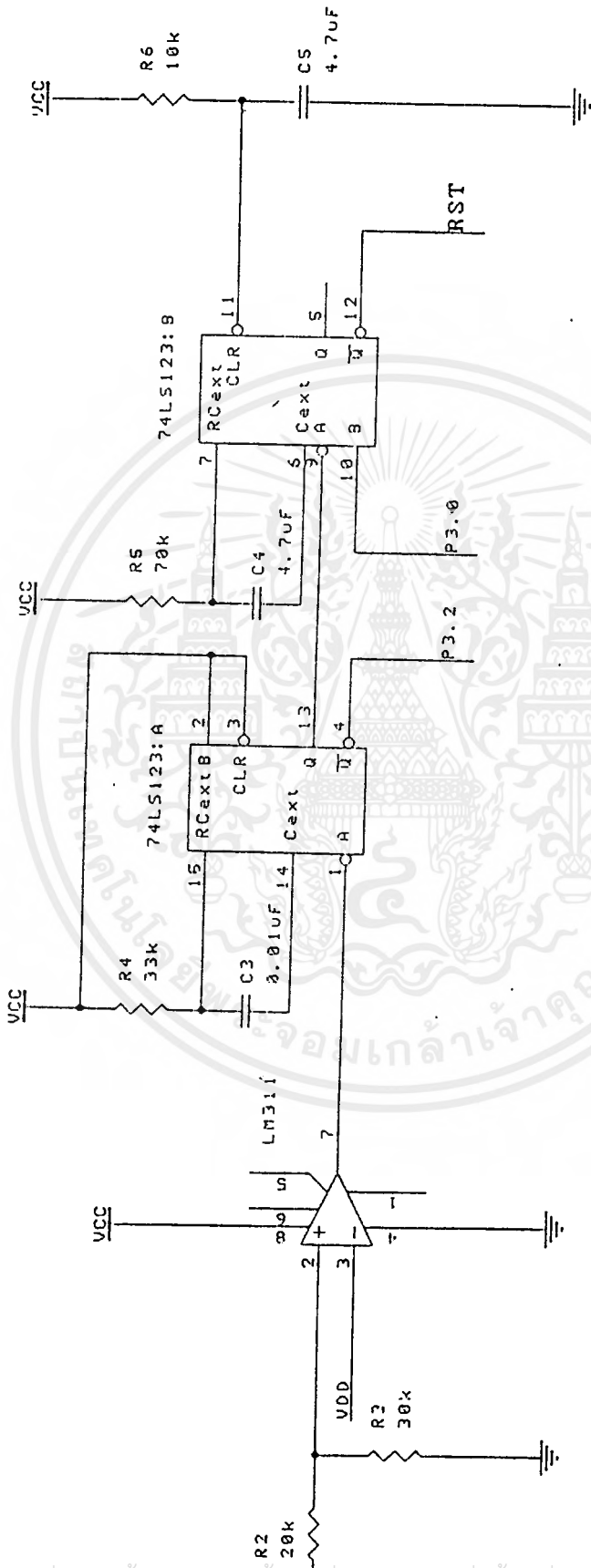
รูปที่ 6.2 รูปแสดงจุดต่อเชื่อมของ ไอซีเบอร์ 89C52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



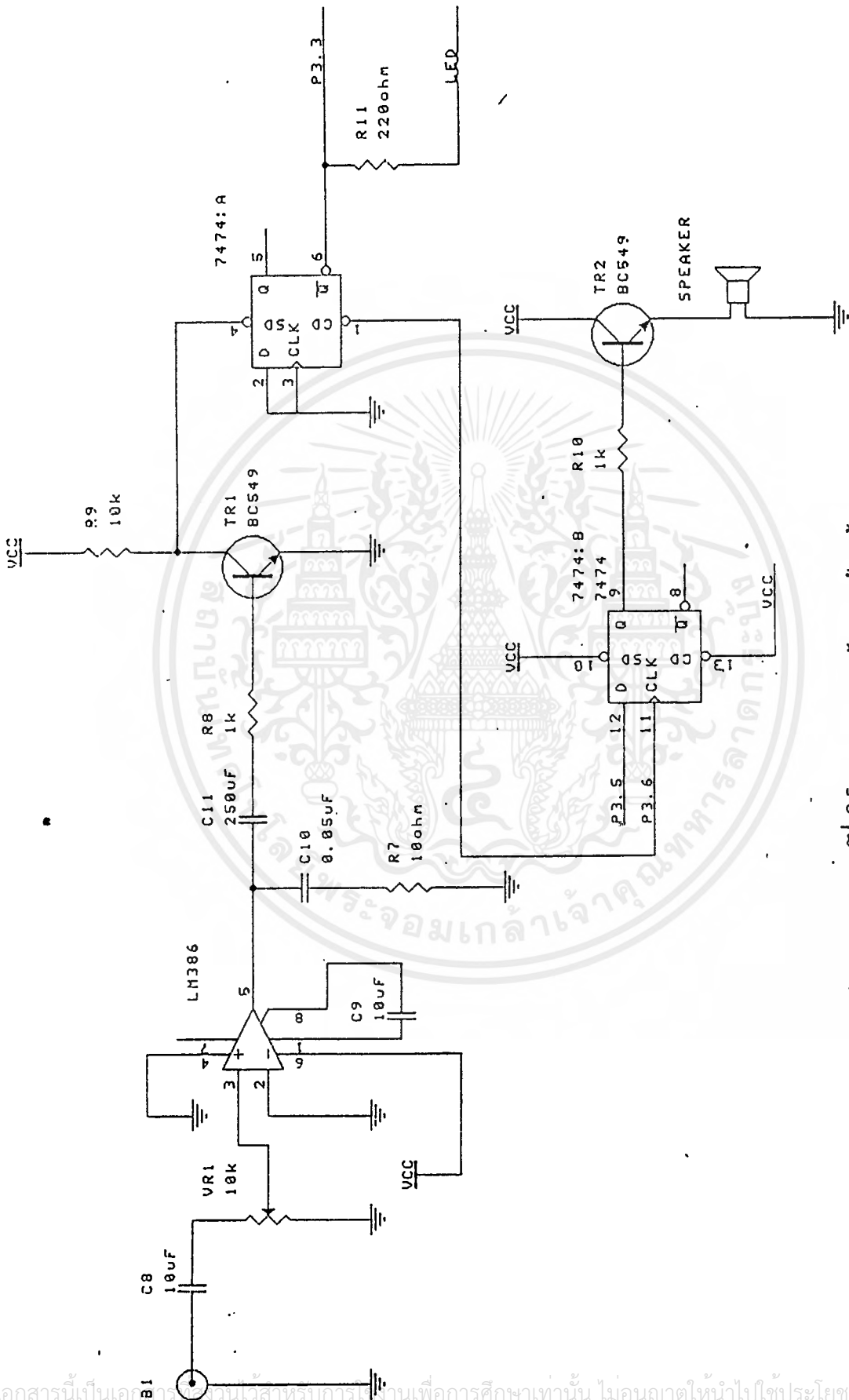
รูปที่ 6.3 วงจรแหล่งจ่ายไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 6.4 วงจรป้องกันคอมพิวเตอรืเอนก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 6.5 วงจรตรวจสอบสัญญาณป้อนเข้า

เอกสารนี้เป็นเอกสารสงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ทำการแก้ไขใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

การทดลองและผลการทดลอง

ระบบการควบคุมการเปิด-ปิดเครื่องใช้ไฟฟ้า โดยช่องสัญญาณเทปและเวลาจริงเครื่องนี้ ประกอบด้วยชนิดการทำงาน 8 ชนิด โดยในแต่ละชนิดมีการทำงานดังนี้

การทำงานของชนิดการทำงานทั้ง 8 ชนิด

1. On คือเมื่อเวลาจริงตรงกับค่าวันเวลาที่เรากำหนด ตำแหน่งเอาต์พุตที่เรากำหนดไว้ก็จะOn
2. Off คือเมื่อเวลาจริงตรงกับค่าวันเวลาที่เรากำหนด ตำแหน่งเอาต์พุตที่เรากำหนดไว้ก็จะOff
3. On and off (no sound) คือเมื่อวันเวลาจริงตรงกับค่าวันเวลาที่เรากำหนด ตำแหน่งเอาต์พุตที่เรากำหนดไว้ก็จะOn และจะOffเมื่อสัญญาณเสียงขาดหายไปนานเท่ากับเวลาที่เรากำหนดในตัวแปรdelaysnd
4. Off and on (no sound) คือเมื่อวันเวลาจริงตรงกับค่าวันเวลาที่กำหนดตำแหน่งเอาต์พุตที่เรากำหนดไว้ก็จะOff และจะOnเมื่อสัญญาณเสียงขาดหายไปนานเท่ากับเวลาที่เรากำหนดในตัวแปรdelaysnd
5. Time length (On) (when Off rec. destroyed) (day don't check) เมื่อเรากำหนดข้อมูลในrecordเรียบร้อยแล้ว เอาต์พุตที่เรากำหนดไว้ก็จะOnตามเวลาที่เรากำหนด เมื่อครบเวลาเอาต์พุตเหล่านั้นก็จะOff แล้วrecordนั้นก็จะถูกลบ
6. Time length (Off) (when On rec. destroyed) (day don't check) เมื่อเรากำหนดข้อมูลในrecordเรียบร้อยแล้ว เอาต์พุตที่เรากำหนดไว้ก็จะOffตามเวลาที่เรากำหนด เมื่อครบเวลาเอาต์พุตเหล่านั้นก็จะOn แล้วrecordนั้นก็จะถูกลบ
7. On 1-59 seconds (depend on setup {delaytim}) and Off เมื่อเวลาจริงตรงกับค่าเวลาที่เรากำหนด ตำแหน่งเอาต์พุตที่เรากำหนดไว้ก็จะOnเป็นเวลาตามที่เรากำหนด โดยอยู่ในช่วง 1-59 วินาที
8. On and Off alternate 1-30 min เมื่อเรากำหนดข้อมูลในrecordเรียบร้อยแล้ว เอาต์พุตที่เรากำหนดไว้ก็จะOn และOff สลับกันไป เป็นเวลาตามที่เรากำหนดในช่วง 1-30 นาที

จากการทดลองระบบควบคุมการเปิด-ปิดเครื่องใช้ไฟฟ้านี้สามารถทำงานได้เพียง 5 ชนิดการทำงานคือ ชนิดการทำงานที่1,2,5,6 และ8

บทที่ 8

บทวิจารณ์และสรุป

จากการทดลองโครงงานนี้กลุ่มของผู้จัดทำสามารถจำแนกข้อดีและข้อเสียของโครงงานนี้ได้ดังนี้

นี้

ข้อดี

1. ประหยัด เนื่องจากสัญญาณของระบบได้มาจากวงจรtimerภายในCPU ดังนั้นจึงประหยัดค่าใช้จ่ายทางด้านฮาร์ดแวร์ในเรื่องของRTC (Real Time Clock)
2. มีวิธีการใช้งานที่ง่าย และสะดวก
3. ขนาดกระทัดรัด
4. มีชนิดการทำงานให้เลือกใช้งานได้หลายชนิด
5. สามารถควบคุมการเปิด-ปิดอุปกรณ์ไฟฟ้าได้โดยตรงจากคีย์บอร์ด
6. สามารถตั้งเวลาเปิด-ปิดอุปกรณ์ไฟฟ้าที่ใช้เป็นประจำวันหรือประจำสัปดาห์

ข้อเสีย

1. ในชนิดการทำงานที่ 5 และ 6 ช่วงเวลาที่เราคือต้องการ On หรือ Off อุปกรณ์ไฟฟ้าจะมีค่าเวลาของการ On หรือ Off น้อยกว่าเวลาที่เรากำหนดไว้ 0-59 วินาที ทั้งนี้เนื่องจากเครื่องควบคุมนี้มีการเช็คข้อมูลในrecordทุกๆนาทีก และชนิดการทำงานที่ 5 และ 6 เมื่อเราทำการ โปรแกรมเรียบร้อยแล้วอุปกรณ์ไฟฟ้าจะ On หรือ Off ทันทีแล้วแต่ชนิดการทำงาน ดังนั้นหากเราทำการ โปรแกรมเสร็จในช่วงระหว่างนาทีก ค่าวินาทีเหล่านั้นจะเป็นค่าที่ทำให้การ On หรือ Off น้อยกว่าช่วงเวลาที่เรากำหนด ตัวอย่างเช่นหากเราทำการ โปรแกรมชนิดการทำงานที่ 5 เสร็จที่เวลา 02:05:12 โดยมีหน่วยเป็นชั่วโมง นาทีและวินาทีตามลำดับ และเราต้องการ ให้ On อุปกรณ์ 5 นาที เมื่อถึงเวลา 02:10:00 อุปกรณ์ไฟฟ้าจะ Off ทันทีซึ่งหมายถึงน้อยกว่าเวลาที่เรากำหนด On 12 วินาที
2. ในการเคลียร์ข้อมูลในrecord อาจทำให้เกิดการสับสน ทั้งนี้เพราะในการเคลียร์ค่าข้อมูล เครื่องควบคุมเครื่องนี้จะนำค่าข้อมูลของrecord ที่ขุดสุดมาบันทึกที่ตำแหน่งของrecord ที่เราต้องการเคลียร์

สรุป

โครงงานนี้สามารถทำงานได้ 5 ชนิดการทำงาน จากชนิดการทำงานทั้งหมด 8 ชนิดคือชนิดการทำงานที่ 1, 2, 5, 6 และ 8 ทั้งนี้เนื่องจาก 3 ชนิดการทำงานที่เหลือมีปัญหาในเรื่องของการ โปรแกรมที่ซับซ้อน แต่ในส่วนทางด้านฮาร์ดแวร์ทางกลุ่มผู้จัดทำได้ทำการออกแบบเตรียมพร้อมไว้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งาน

ก่อนที่จะเราจะทำรายการ โปรแกรมต่างๆ เราจะต้องมีการเชื่อมต่อระบบต่างๆ โดยทำการกำหนดวันใน Group Day คือกลุ่มวันที่เราต้องการให้อุปกรณ์หรือระบบที่เราต้องการควบคุมให้ทำงาน หรือไม่ทำงานประจำสัปดาห์ โดยเราสามารถ กำหนดตั้งวันได้ 4 Group Day เราจะเริ่มจากการเชื่อมต่อระบบก่อน โดย เมื่อเราทำการเปิดเครื่องหน้าจอ LCD จะแสดง Title จากนั้นจะแสดงวันและเวลาโดยมีหน้าจอดังนี้

SUN:00:00:00

1. การเชื่อมต่อระบบ (System Set up)

ในการเชื่อมต่อระบบต้องกดที่ปุ่ม Set up จากนั้นหน้าจอจะแสดง System setup แล้วจะแสดงหน้าจอเป็น

Tuning Time → 20

ซึ่งหมายถึงค่าในการทำให้ ช่วงเวลาในแต่ละวินาทีบนเครื่องมีค่าใกล้เคียงกับเวลาจริงมากที่สุด ซึ่งเริ่มแรกเราจะกำหนดไว้ที่ 20 ในการปรับค่าเราทำได้โดยการกดปุ่ม + หรือ - จากนั้นเรากดปุ่ม → หน้าจอจะแสดง

Speaker → ON

หมายถึงว่าเราต้องการให้มีเสียงในการกดปุ่มแต่ละครั้งหรือไม่ โดยเริ่มแรกเราได้กำหนดให้เป็น ON ไว้ ในการปรับจะทำได้โดยการ กดปุ่ม + หรือ - จากนั้นกดปุ่ม → หน้าจอจะแสดง

DELAY TYPE → 05

หมายถึงเวลาที่ต้องการให้อุปกรณ์ไฟฟ้า ON และ OFF ใน การทำงาน type 7 โดยมันมีหน่วยเป็นวินาที เริ่มแรกเรากำหนดค่าไว้ที่ 05 ในการปรับค่าจะทำได้โดยการกดปุ่ม + หรือ - จากนั้นกดปุ่ม → หน้าจอจะแสดง

SOUND DELAY 05 →

หมายถึงช่วงเวลาในการเช็คแบล็ก ในการทำงาน type 3,4 การปรับค่า กดปุ่ม + หรือ - จากนั้นกดปุ่ม → หน้าจอจะแสดง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Out → Speaker → 01

หมายถึง การปรับความดังของเสียงจากการกดปุ่ม การปรับค่า กดปุ่ม + หรือ - จากนั้นกดปุ่ม
→ หน้าจอจะแสดง

Out put → Enable

หมายถึงว่าเราต้องการให้มันขับเอาท์พุทหรือไม่ เริ่มแรกเรากำหนด Enable คือต้องการให้ขับเอาท์พุท
ซึ่งจะเป็นการสิ้นสุดของการเช็คอ้พระบบ ในการย้อนกลับไปเช็คอ้พระบบก่อนหน้าให้กดปุ่ม ←

2. การกำหนดวันทำงาน ใน 1 สัปดาห์ (Group Day)

โดยเมื่อเรากดปุ่ม GroupDay หน้าจอจะแสดงดังนี้

GroupDay 1_-----

ซึ่งตำแหน่งหลังตัวเลขแสดงวัน โดยเริ่มจาก SUN ไปจนถึง SAT หากเราต้องการให้เครื่องใช้ไฟฟ้าทำ
งาน ในวันใดบ้าง เราก็ทำการกดปุ่ม Set เซอร์เซอร์จะปรากฏที่ตำแหน่งวัน อาทิตย์ โดยการกดปุ่ม +
เมื่อต้องการให้เครื่องใช้ไฟฟ้าหรืออุปกรณ์ที่เราต้องการควบคุมทำงานในวันอาทิตย์ และกดปุ่ม - เมื่อ
ไม่ต้องการให้เครื่องใช้ไฟฟ้าหรืออุปกรณ์ที่เราไม่ต้องการควบคุมทำงานในวันอาทิตย์ โดยหน้าจอจะ
แสดงผล * หมายถึง ให้ทำงาน

- หมายถึง ไม่ทำงาน

ซึ่งในวันอื่นๆก็ทำเช่นเดียวกัน โดยเลื่อนเซอร์เซอร์ไปในตำแหน่งวันที่ต้องการ กำหนดให้ทำงานหรือไม่
ทำงาน โดยกดปุ่ม → หรือ ← เมื่อเรียบร้อยแล้วเรากดปุ่ม Set อีกครั้ง และในการกำหนดวันทำงาน
ใน GroupsDay อื่นๆทำได้โดยการกดปุ่ม GroupDay เครื่องจะวนจาก GroupDay 1→2→3→4 ไป
เรื่อยๆ

3. การโปรแกรม (PROGRAM)

เมื่อกดปุ่ม Prog ครั้งแรกจะเป็นการให้เลือกชนิดการทำงาน โดยมีชนิดการทำงาน 8 ชนิดคือ

1.On เป็นการเลือกให้ ON อุปกรณ์ไฟฟ้า

2.Off เป็นการเลือกให้ Off อุปกรณ์ไฟฟ้า

3.On and Off (no sound) คือให้อุปกรณ์ไฟฟ้า On และOff เมื่อมีช่องว่างของเทปตามเวลาที่

กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. Off and On (no sound) คือให้อุปกรณ์ไฟฟ้า Off และ On เมื่อมีช่องว่างของเทปตามเวลาที่กำหนด

5. time lenght (On) (when off rec. destroyed)(day don't check) คือเมื่อทำการป้อนโปรแกรมแล้วอุปกรณ์ไฟฟ้าจะ On ตามช่วงเวลาที่ได้โปรแกรมไว้ แล้วจึง Off เมื่อ Off แล้ว record นั้นก็จะถูกทำลาย

6. time lenght (Off)(when on rec. destroyed)(day don't check) คือเมื่อทำการป้อนโปรแกรมแล้วอุปกรณ์ไฟฟ้าจะ Off ตามช่วงเวลาที่ได้โปรแกรมไว้แล้วจึง On เมื่อ On แล้ว record นั้นก็จะถูกทำลาย

7. On 1-59 seconds (depend on setup {delaytim}) and Off คือจะ On อุปกรณ์ไฟฟ้าเป็นช่วงเวลาภายใน 1-59 วินาที ซึ่งเราตั้งค่าเวลาได้ตอนที่เราระเซ็ตอ้ระบบ จากนั้นจะ Off

8. On and Off alternate 1-30 min เป็นการ On และ Off อุปกรณ์ไฟฟ้าสลับกัน โดยช่วงเวลาที่ทำการ On หรือ Off จะอยู่ภายในช่วงเวลา 1-30 นาที

หลังจากเราเลือกชนิดการทำงานแล้วเมื่อเรากดปุ่ม Prog อีกครั้ง จะเป็นการเลือกมาให้อุปกรณ์ไฟฟ้าทำงานวันใดในรอบสัปดาห์ โดยเลือก Groupday 1-4 โดยชนิดการทำงาน 5,6 และ 8 จะไม่สามารถกำหนดวันทำงานได้ โดยจะทำงานทันทีที่เราโปรแกรมเสร็จ เมื่อเรากดปุ่ม Prog อีกครั้งหน้าจอจะให้ใส่เวลาที่ต้องการให้อุปกรณ์เริ่มทำงาน โดยชนิดการทำงานที่ 5, 6 และ 8 เวลาที่ใส่จะเป็นช่วงเวลาที่อุปกรณ์ทำงานไม่ใช่เวลาที่เริ่มทำงาน จากนั้นเมื่อเรากดปุ่ม Prog อีกครั้งจะเป็นการเลือกว่าต้องการให้ขับเอาท์พุทควัโหวน ซึ่งมีอยู่ 8 ตัว เมื่อกดปุ่ม Prog อีกครั้งจะขึ้นหน้าจอ

<<<<<< O.K. >>>>>>

เป็นการสิ้นสุดการ โปรแกรม ข้อมูลต่างๆจะเก็บไว้ในหน่วยความจำ แต่หากเราโปรแกรมเกิน 40 รายการ หน้าจอจะแสดง

Error! full buff

ซึ่งเราต้องทำการลบรายการบางรายการออกก่อนแล้วจึง โปรแกรมรายการเข้าไปใหม่

4. LIST

เป็นการแสดงรายละเอียดของ record ต่างๆ

1. เมื่อเรากดปุ่ม List ครั้งแรกหน้าจอจะขึ้น

Total Records =

เพื่อเป็นการบอกจำนวน Record ที่มีอยู่ทั้งหมด จากนั้นจะแสดงรายละเอียดของ Record NO.1 ซึ่งหากเราต้องการทราบรายละเอียดของ Record NO.1 ทั้งหมด ให้เรากดปุ่ม + และเมื่อแสดงรายละเอียดของ Record NO.1 เสร็จแล้ว มันจะแสดงรายละเอียดของ Record NO.2 ค่อไป หากเราต้องการ

ทราบรายละเอียดของ Record NO. อื่นๆ เราสามารถเลือกได้โดยกดปุ่ม → หรือ ← เมื่อนำจอขึ้น Record NO. ที่ต้องการแล้ว เราจึงกดปุ่ม + เพื่อดูรายละเอียดภายใน NO. นั้นๆ

2. เมื่อกดปุ่ม List อีกครั้ง หน้าจอจะแสดง

Clear? [+OK]

ซึ่งเป็นการถามว่าต้องการเคลีย Record ที่กำลังแสดงอยู่หรือไม่ หากต้องการให้กดปุ่ม + จากนั้นจะขึ้นหน้าจอ

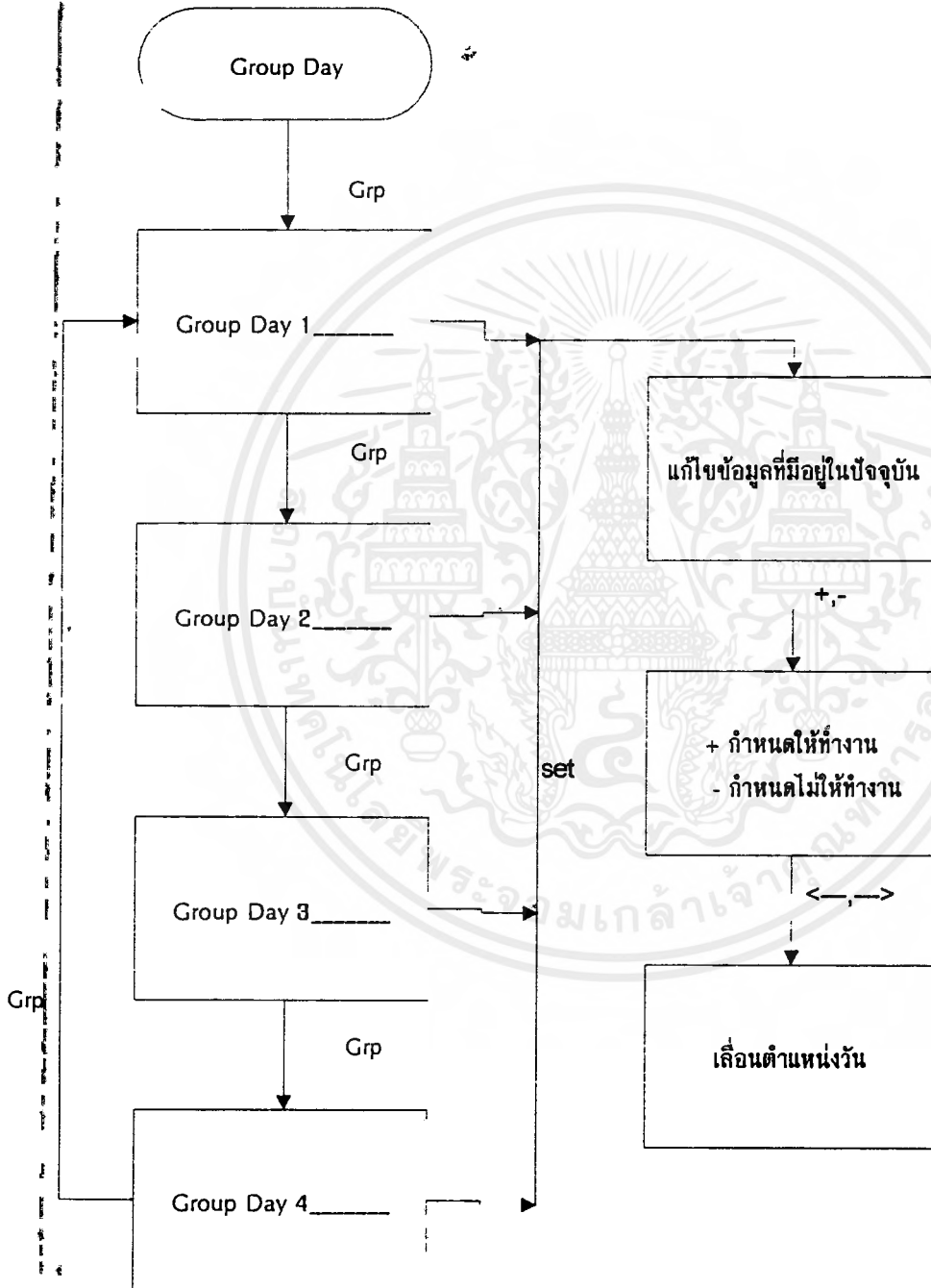
Clear Complete

ซึ่งเป็นการบอกให้เราทราบว่า Record นั้น ได้ถูกเคลียไปแล้ว

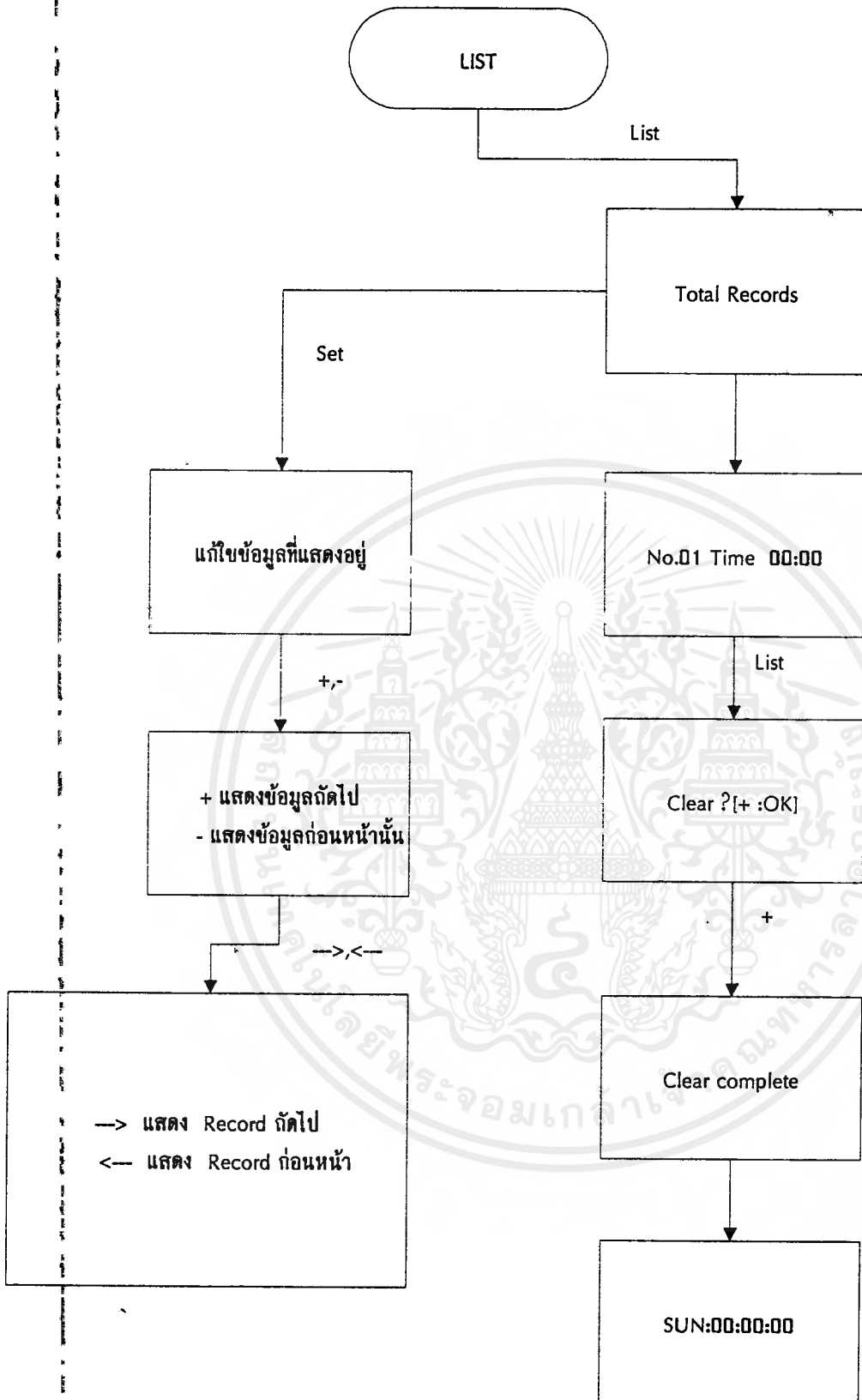
5. Tim/out

เป็นปุ่มเพื่อแสดง วัน-เวลา และเอาท์พุทที่ทำงานอยู่ ณ ปัจจุบันสลับกัน โดยหากเดิมแสดงวัน เวลาอยู่เมื่อกดปุ่ม Tim/out จะเปลี่ยนเป็นการแสดงเอาท์พุท โดยทั้งวัน-เวลา และเอาท์พุท เราสามารถปรับหรือ set ได้ โดยกดปุ่ม set หนึ่งครั้งจะปรากฏเลขเซอร์ จากนั้นเราก็ทำการเซตค่าต่างๆตามต้องการ แล้วกดปุ่ม set อีกครั้งเป็นอันเรียบร้อย

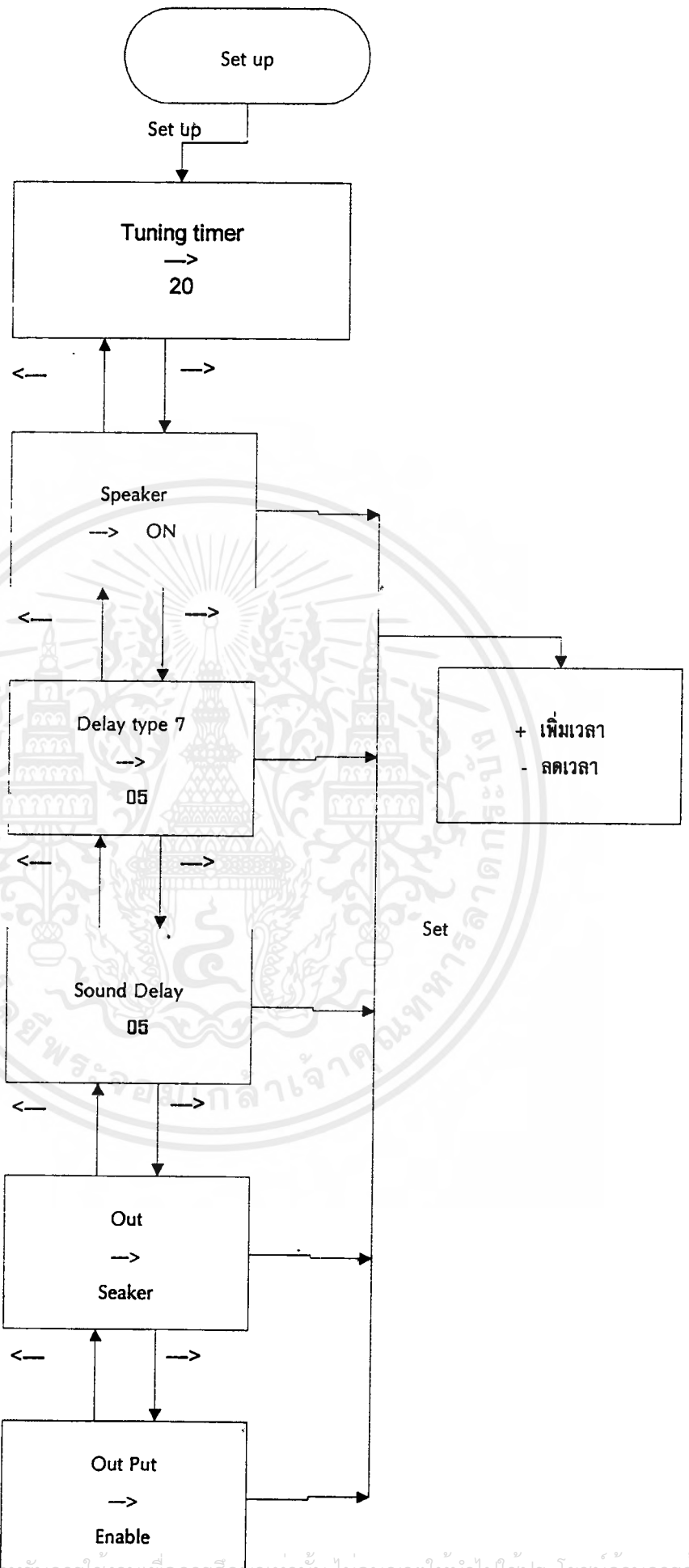
สรุปรูปแบบการกตัญญูและการแสดงผล



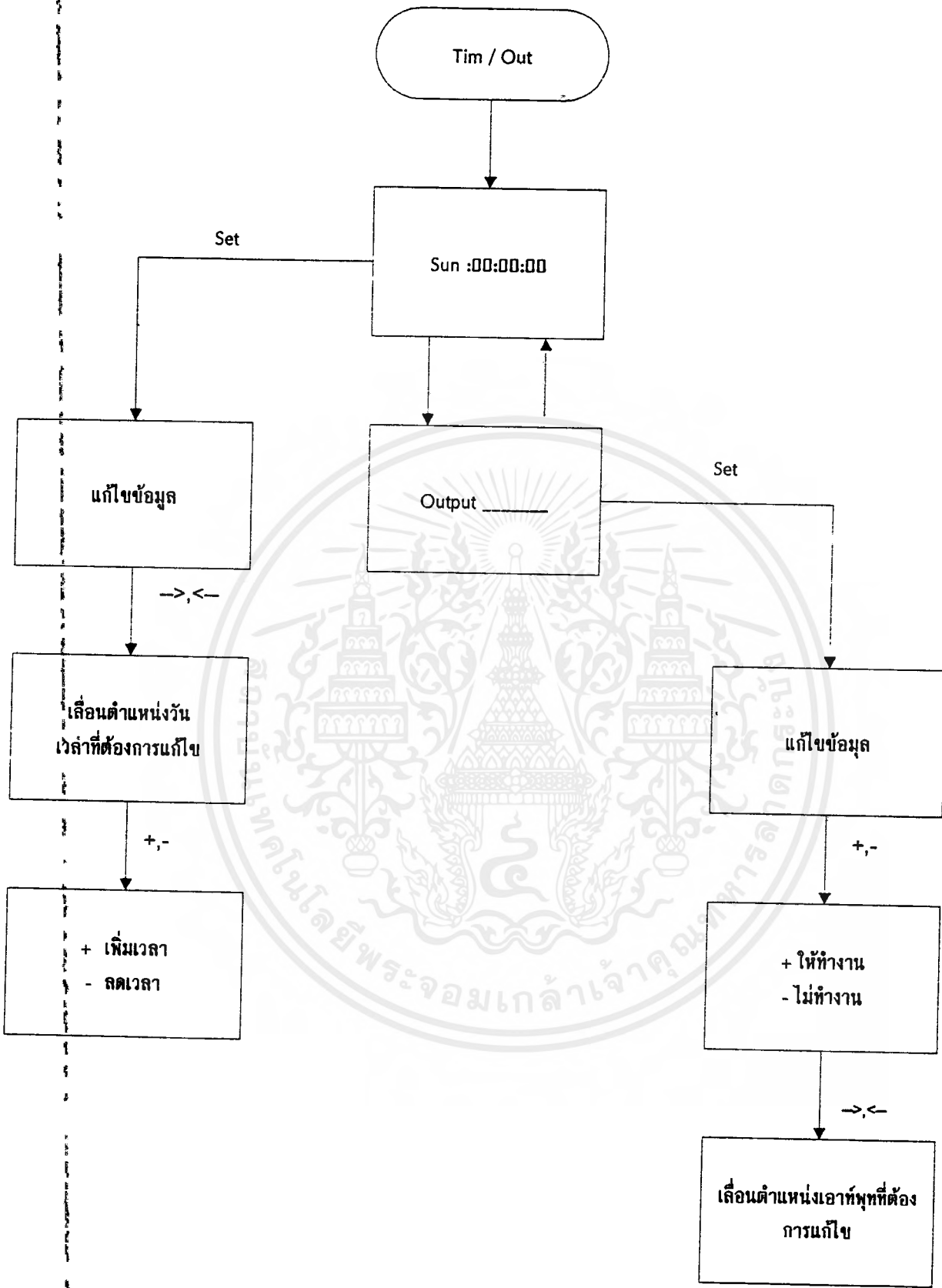
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่มีการตีพิมพ์ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรณีใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



1. ผ.ศ. ทิพนันต์ เลาสงคราม , “ภาษาแอสเซมบลีสำหรับ MCS-48 และ MCS-51”,
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า
เจ้าคุณทหารฯ ลาดกระบัง
2. ผ.ศ. สมยศ จุณณะปิยะ , “การใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51”,
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า
เจ้าคุณทหารฯ ลาดกระบัง
3. พันจันทร์ ธนวิวัฒนเสถียร , “ไมโครโปรเซสเซอร์ ทฤษฎีและการประยุกต์”,
ซีเอ็ดยูเคชั่น
4. สุนทร วิบูลย์พรหม , “การโปรแกรมภาษาแอสเซมบลี ของไมโครคอนโทรลเลอร์
ตระกูล MCS-51”, ซีเอ็ดยูเคชั่น

