สำนักหอสมุดกลาง_พระจอมเกล้าลาดกระบัง

เครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้

ELECTRIC APPLIANCE CONTROLLER

T045716

โดย

นางสาวเสริมศรี จารุวัฒนดิลก

หม่อมหลวงอรวรรณ สุขสวัสดิ์

เลขหมู่.....
เลขทะเบียน 45716
วัน, เดือน, ปี 13 ก.พ. 2546

.b.....
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมบัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2544

ภาควิชาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้

ผู้จัดทำ

1. นางสาวเสริมศรี จารุวัฒนดิลก
2. หม่อมหลวงอรวรรณ สุขสวัสดิ์

.....อาจารย์ที่ปรึกษา

(ผู้ช่วยศาสตราจารย์ ดร.เกียรติศักดิ์ คมวัชระ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมอุปกรณ์ไฟฟ้าภายในบ้านแบบโปรแกรมได้

นางสาว เสริมศรี จารุวัฒนดิลก
หม่อมหลวง อรวรรณ สุขสวัสดิ์
ผศ.ดร. เกียรติศักดิ์ คมวัชระ อาจารย์ที่ปรึกษา
ปีการศึกษา 2544

บทคัดย่อ

โครงงานนี้จัดทำขึ้นเพื่อเป็นประโยชน์ทางด้านการควบคุมการ เปิด – ปิด เครื่องใช้ไฟฟ้าภายในบ้านตามที่ได้ตั้งโปรแกรมเอาไว้ สามารถควบคุมอุปกรณ์ไฟฟ้าได้ 8 อุปกรณ์ และแต่ละอุปกรณ์ยังสามารถถูกควบคุมได้อีก 8 โปรแกรมย่อย มีแป้นสวิตช์เมตริก 4x3 จุด (Keypad) เป็นส่วนรับข้อมูล (Input) และมีการติดต่อ รับ – ส่ง ข้อมูลกับไอซีสร้างฐานเวลาจริง (Real Time Clock: RTC) ส่วนข้อมูลที่ได้จากการประมวลผลด้วยไมโครคอนโทรลเลอร์ที่นำมาประยุกต์ใช้เป็นส่วนประมวลผลกลาง จะถูกส่งไปยังส่วนแสดงผลแบบผลึกเหลว (LCD) และส่งสัญญาณไปขับรีเลย์ให้เปิด – ปิด อุปกรณ์เครื่องใช้ไฟฟ้าตามเวลาที่ได้ตั้งโปรแกรมเอาไว้

Abstract

The target of this project is for controlling the ON – OFF function of any appliance, which have been programmed, with capability of 8 channels to be controlled and each channel is controlled by 8 subprograms. The microcontroller is used in this project as the central processor. It is connected to the matrix switch (Keypad), which function as input, and also communicates with Real Time Clock (RTC). After processing, the data from the microcontroller will be transmitted to the output part, Liquid Crystal Display (LCD), and sent the signal to control the ON – OFF function of the appliance.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

สารบัญ ค

สารบัญตาราง ง

สารบัญภาพ จ

บทที่ 1 บทนำ 1

1.1 ความเป็นมา 1

1.2 วัตถุประสงค์ 1

1.3 ขอบเขตการทำโครงงาน 2

1.4 ขั้นตอนการดำเนินงาน 2

1.5 ประโยชน์ที่ได้รับจากโครงงาน 2

บทที่ 2 ทฤษฎีและหลักการ 3

2.1 โครงสร้างและสถาปัตยกรรมของไมโครคอรโทรลเลอร์ MCS-51 แบบแฟลช 3

2.2 การนำพอร์ตมาใช้งาน 9

2.3 จังหวะการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 11

2.4 กระบวนการรีเซตในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 14

2.4.1 เพาเวอร์ออนรีเซต 15

2.5 วงจรนาฬิกาของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 16

2.6 การจัดหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช 17

2.6.1 หน่วยความจำโปรแกรม 17

2.6.2 หน่วยความจำข้อมูล 19

2.7 การใช้งาน 8255 กับ AT89C52 25

2.7.1 ลักษณะพื้นฐานของ 8255 25

2.7.2 การจำแนกกลุ่มพอร์ต 8255 26

2.7.3 รูปแบบคำสั่งเพื่อกำหนดการทำงานของ 8255 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.7.4 การทำงานโหมด 0 ของ 8255 28

2.8 ไอซีสร้างฐานเวลาจริง 29

2.8.1 รายละเอียดขาต่อใช้งานของ DS1302 29

2.8.2 รูปแบบคำสั่งการทำงานของ DS1302 30

2.8.3 การทำงานของ DS1302 30

2.8.4 การจัดสรรหน่วยความจำใน DS1302 31

2.9 ส่วนแสดงผลแบบผลึกเหลว 32

2.9.1 โครงสร้างภายในของตัวควบคุมส่วนแสดงผลแบบผลึกเหลว 33

2.9.2 คำสั่งควบคุมส่วนแสดงผลแบบผลึกเหลว 34

2.9.3 การเขียนคำสั่งและข้อมูลให้แก่ส่วนแสดงผลแบบผลึกเหลว 38

2.9.4 จังหวะการทำงานของส่วนแสดงผลแบบผลึกเหลว 38

บทที่ 3 การออกแบบวงจร 40

3.1 การออกแบบฮาร์ดแวร์ 40

3.1.1 ภาครับข้อมูล 40

3.1.2 ภาคประมวลผล 41

3.1.3 ภาคแสดงผล 44

3.1.4 ภาคขับโหลด 45

3.2 การออกแบบซอฟต์แวร์ 45

บทที่ 4 การทดลองและผลการทดลอง 46

4.1 การทดลอง 46

4.1.1 เมื่อสวิตช์อยู่ที่โหมดตั้งเวลา 46

4.1.2 เมื่อสวิตช์อยู่ที่โหมดโปรแกรม 47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
บทที่ 5 บทวิจารณ์และสรุป	50
ภาคผนวก	
ก โฟลวชาร์ตแสดงการทำงานทั้งหมด	51
ข รูปการเชื่อมต่อวงจรทั้งหมดเข้าด้วยกัน	59
กิตติกรรมประกาศ	ฉ
หนังสืออ้างอิง	ช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตาราง

2.1 ตารางรายละเอียดบางส่วนของไมโครคอนโทรลเลอร์ตระกูล MCS-51 4

2.2 ตารางแสดงผลของการรีเซตที่มีต่อรีจิสเตอร์ภายในไมโครคอนโทรลเลอร์ 15

2.3 ตารางแสดงกลุ่มแอดเดรสที่ถูกใช้งานในฐานะรีจิสเตอร์ใช้งานทั่วไป 20

2.4 ตารางแสดงการสวิตช์เลือกแต่ละกลุ่มของรีจิสเตอร์ 20

2.5 ตารางแสดงการจัดกลุ่มของพอร์ต 26

2.6 ตารางแสดงหน้าที่การทำงานของสัญญาณไอซี 8255 26

2.7 ตารางแสดงตัวอย่างและความหมายในการกำหนดไบต์ควบคุมเพื่อใช้งาน 8255 28

2.8 ตารางแสดงการกำหนดบิตควบคุมการเลื่อนเคอร์เซอร์และตัวอักษรบนจอแสดงผล 36

3.1 แสดงรหัสที่ใช้ในการเลือกพอร์ต 42

3.2 แสดงบิตที่ใช้ในการรับ-ส่งของพอร์ตต่างๆ 43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

หน้า

รูปที่

2.1 แสดงโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม89CXX 4

2.2 แสดงสถาปัตยกรรมโครงสร้างภายในไมโครคอนโทรลเลอร์ MCS-51 5

2.3 แสดงการจัดวางขาใช้งานพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 5

2.4 แสดงโครงสร้างพอร์ต 0 (ต่อบิต) 6

2.5 แสดงโครงสร้างพอร์ต 1 (ต่อบิต) 6

2.6 แสดงการต่อความต้านทานภายในพูลอัพ 7

2.7 แสดงโครงสร้างพอร์ต 2 (ต่อบิต) 7

2.8 แสดงโครงสร้างพอร์ต 3 (ต่อบิต) 8

2.9 แสดงจังหวะการทำงานของไมโครคอนโทรลเลอร์ 12

2.9 (ก) แสดงการทำงานของคำสั่ง INC A 12

2.9 (ข) แสดงการทำงานของคำสั่ง ADD A, # DATA 12

2.9 (ข) แสดงการทำงานของคำสั่ง INC DTPR 12

2.9 (ข) แสดงการทำงานของคำสั่ง MOV X 12

2.9 (A) แสดงสัญญาณและไดอะแกรมเวลาในขณะที่ยังไม่มี
การกระทำคำสั่ง MOV X 13

2.9 (B) แสดงสัญญาณและไดอะแกรมเวลาในขณะกระทำคำสั่ง MOV X 13

2.10 แสดงไดอะแกรมเวลาของกระบวนการรีเซตภายในไมโครคอนโทรลเลอร์ 14

2.11 แสดงวงจรเพาเวอร์ออนรีเซต 16

2.12 แสดงการต่อวงจรนาฬิกาภายในชิพ 16

2.13 แสดงการต่อวงจรสร้างสัญญาณนาฬิกาจากภายนอก 17

2.14 แสดงการจัดพื้นที่หน่วยความจำโปรแกรมสำหรับไมโครคอนโทรลเลอร์
เบอร์ AT89C52 18

2.15 แสดงการจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในไมโครคอนโทรลเลอร์ 19

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

หน้า

รูปที่

2.16	แสดงหน่วยความจำข้อมูลภายในบริเวณที่อ้างถึงได้แบบบิต	21
2.17	แสดงรีจิสเตอร์ใช้งานพิเศษ	22
2.18	แสดงแผนภาพแบบบล็อกภายในและขาสัญญาณของไอซีเบอร์ 8255	25
2.19	แสดงความหมายของบิตภายในไบต์ข้อมูลควบคุมสำหรับ 8255	27
2.20	แสดงการจัดขาของ DS 1302	29
2.21	แสดงรูปแบบไบต์คำสั่งกำหนดการทำงานของ DS 1302	30
2.22	แสดงไดอะแกรมสัญญาณการเขียนข้อมูลลง DS1302	31
2.23	แสดงไดอะแกรมสัญญาณการอ่านข้อมูลจาก DS1302	31
2.24	แสดงการจัดสรรหน่วยความจำใน DS 1302	31
	2.24 (ก) แสดงการจัดสรรหน่วยความจำส่วนของแรม	31
	2.24 (ข) แสดงการจัดสรรหน่วยความจำส่วนของนาฬิกา	32
2.25	แสดงบล็อกไดอะแกรมภายในของชิพควบคุมส่วนแสดงผลแบบผลึกเหลว	33
3.1	แสดงบล็อกไดอะแกรมการทำงานโดยรวมของวงจร	40
3.2	แสดงวงจรเชื่อมต่อแป้นสวิตช์เมตริกเข้ากับ 8255	41
3.3	แผนภาพแสดงการสร้างสัญญาณเลือกอุปกรณ์ (CS\) ให้กับ 8255 โดยการถอดรหัสจากบัสแอดเดรส A2 – A7	43
3.4	แสดงการเชื่อมต่อระหว่าง 8255 และ AT89C52	44
3.5	แสดงการเชื่อมต่อส่วนแสดงผลแบบผลึกเหลวกับไมโครคอนโทรลเลอร์	44
3.6	แสดงการเชื่อมต่อของ 8255 กับรีเลย์	45
ก-1	แสดงโฟลวชาร์ตโปรแกรมหลัก	51
ก-2	แสดงโฟลวชาร์ตโปรแกรมการเปลี่ยนรหัสผ่าน	52
ก-3	แสดงโฟลวชาร์ตการตั้งเวลาแต่ละโหลดและแต่ละโปรแกรม (1)	53
ก-4	แสดงโฟลวชาร์ตการตั้งเวลาแต่ละโหลดและแต่ละโปรแกรม (2)	54
ก-5	แสดงโฟลวชาร์ตโปรแกรมย่อยการอ่านข้อมูลจากไอซีสร้างฐานเวลาจริง	55

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (ต่อ)

หน้า

รูปที่

ก-6 แสดงโฟลวชาร์ตโปรแกรมย่อยการเขียนข้อมูลไปยังไอซีสร้างฐานเวลาจริง 56

ก-7 แสดงโฟลวชาร์ตโปรแกรมย่อยตรวจสอบแต่ละโหลด 57

ก-8 แสดงโฟลวชาร์ตโปรแกรมย่อยตรวจสอบแต่ละโปรแกรม 58

ข แสดงการเชื่อมต่อวงจรทั้งหมดเข้าด้วยกัน 59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมา

ปัจจุบันประเทศไทยมีการเจริญเติบโตทางเศรษฐกิจอย่างรวดเร็ว ประชากรก็มีจำนวนเพิ่มมากขึ้น มีรายได้เพิ่มขึ้น ก่อให้เกิดความเจริญทางด้านวัตถุ มีการนำเอาเทคโนโลยีใหม่ๆ มาใช้ ทำให้ประชากรส่วนใหญ่มีความสะดวกสบายเพิ่มมากขึ้น

เนื่องจากในปัจจุบันนี้ภายในบ้าน หรือสถานที่ต่าง ไม่ว่าจะเป็น หอพัก อพาร์ทเม้น คอนโดมิเนียม โรงเรียน ที่ทำงาน ฯลฯ เราจะพบว่ามีอุปกรณ์เครื่องใช้ไฟฟ้าต่างๆ มากมายอยู่ในนั้นและถ้าหากเราสังเกตให้ดีจะพบว่า เราจะต้องทำการเปิด – ปิด อุปกรณ์เครื่องใช้ไฟฟ้าบางอย่าง ในเวลาเดิมทุกวันเป็นประจำ เช่น การเปิดและปิดไฟที่บ้านหรือสถานที่ทำงาน การเปิดและปิดเครื่องปรับอากาศ ฯลฯ

โครงงานชิ้นนี้จึงได้ออกแบบขึ้นมาเพื่อช่วยอำนวยความสะดวกในการที่เราต้องทำอะไรซ้ำๆ อย่างนี้เกือบจะทุกวัน ซึ่งบางครั้งเราอาจจะลืมปิดไฟ หรือเครื่องปรับอากาศเวลาเราออกจากสถานที่นั้นๆ ไปแล้ว ทำให้ต้องสูญเสียพลังงาน และค่าใช้จ่ายเพิ่มในการที่เราต้องเปิดเครื่องใช้ไฟฟ้านั้นๆ ทิ้งไว้ หรือจะเป็นการช่วยอำนวยความสะดวกขณะที่เราไม่อยู่บ้าน อุปกรณ์นี้จะช่วยเปิดไฟหน้าบ้านในตอนกลางคืนและปิดไฟในตอนเช้า เสมือนว่ามีคนอยู่ที่บ้านของเรา เป็นการรักษาความปลอดภัยระดับหนึ่งเมื่อเราไม่อยู่บ้าน

ทางผู้จัดทำจึงหวังว่า โครงงานชิ้นนี้จะเป็นประโยชน์ไม่มากก็น้อยสำหรับผู้ที่สนใจ และโครงงานชิ้นนี้ยังได้นำเสนอการทำงานของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 ให้กับผู้ที่สนใจอีกด้วย

1.2 วัตถุประสงค์

วัตถุประสงค์ของโครงงานนี้คือ สามารถสร้างวงจรที่สามารถ เปิด – ปิด เครื่องใช้ไฟฟ้าตามเวลาที่ได้ตั้งเอาไว้ โดยสามารถควบคุมได้ 8 อุปกรณ์ และแต่ละอุปกรณ์สามารถตั้งโปรแกรมได้อีก 8 โปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 ขอบเขตการทำงานของโครงงาน

1.3.1 สามารถตั้งเวลาปัจจุบันและเวลาที่ควบคุมเครื่องใช้ไฟฟ้าได้จากแป้นสวิตช์เมตริก

1.3.2 ทำงานแบบอัตโนมัติ

1.3.3 ควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์

1.3.4 แสดงผลทางส่วนแสดงผลแบบผลึกเหลว และแสดงสถานะทาง หลอดไฟ (LED)

1.3.5 มีแหล่งจ่ายไฟสำรองในกรณีที่ไฟฟ้าดับ

1.4 ขั้นตอนการดำเนินงาน

1.4.1 ศึกษารายละเอียดของโครงงาน

1.4.2 เสนอโครงงาน ออกแบบและทดสอบเขียนโปรแกรมควบคุมการทำงาน

1.4.3 จัดทำปริญญานิพนธ์

1.5 ประโยชน์ที่ได้รับจากโครงงาน

ประโยชน์ที่ได้รับจากโครงงานนี้ คือความรู้และประสบการณ์ จากการศึกษาและวิจัยการทำงานของวงจรต่างๆ การเขียนภาษาแอสเซมบลี บนไมโครคอนโทรลเลอร์ การใช้โปรแกรมโพรเทลในการออกแบบลายวงจร การนำความรู้ที่ได้จากการศึกษาภาคทฤษฎี ในชั้นเรียนมาใช้ในการปฏิบัติงานจริง และยังสามารถนำโครงงานนี้ไปใช้ประโยชน์ในชีวิตประจำวันได้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 โครงสร้างและสถาปัตยกรรมของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล MCS-51 ที่นำมาใช้ มีหน่วยความจำภายในเป็นแบบแฟลช (Flash Memory) ของยี่ห้อ Atmel Corporation มีเบอร์ขึ้นต้นด้วย AT89 ซึ่งมีคุณลักษณะพิเศษดังนี้

- หน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์ สามารถลบและเขียนใหม่ได้นับพันครั้ง จึงสามารถใช้งานในรูปแบบของไมโครคอนโทรลเลอร์ชิพเดี่ยวไม่ต้องใช้หน่วยความจำภายนอก ส่งผลให้สามารถใช้งานพอร์ตอินพุต และพอร์ตเอาท์พุตของไมโครคอนโทรลเลอร์ได้อย่างเต็มประสิทธิภาพ
- ด้วยการใช้หน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ทำให้สามารถป้องกันการคัดลอกข้อมูลของหน่วยความจำโปรแกรมได้อย่างดี
- ชุดคำสั่ง และสถาปัตยกรรมพื้นฐานเหมือนกับไมโครคอนโทรลเลอร์ MCS-51 ของผู้ผลิตอื่น ไม่ว่าจะเป็นอินเทล, ซีเมนส์ หรือดัลลัส

คุณสมบัติทางเทคนิคของไมโครคอนโทรลเลอร์ตระกูล MCS-51 อนุกรม AT89XX

1. เป็นไมโครคอนโทรลเลอร์ที่ใช้หน่วยประมวลผลกลาง (CPU) ขนาด 8 บิต
2. หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม ในบางเบอร์จะมีหน่วยความจำแบบอีอีพรอมเพิ่มเติม
3. ขาพอร์ต (PORT) เป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุต และเอาท์พุต
4. มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
5. ไทเมอร์/เคาน์เตอร์ (Timer/Counter) ขนาด 16 บิต อย่างน้อย 2 ตัว
6. สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
7. สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
8. มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิพ
9. มีวงจรสื่อสารอนุกรมแบบ SPI สำหรับอนุกรม AT89SXX
10. มีวอตช์ด็อกไทเมอร์ในตัว สำหรับอนุกรม AT89SXX

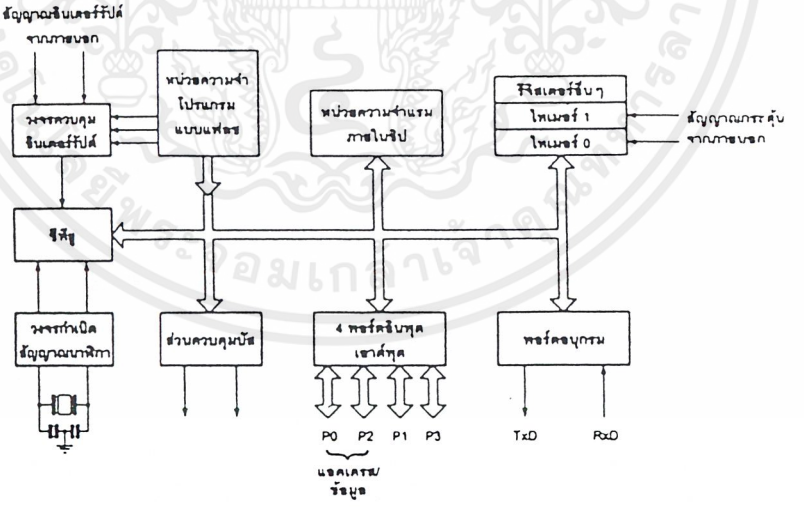
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในตารางที่ 2.1 แสดงรายละเอียดบางส่วนของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 แต่ละเบอร์ที่ Atmel Corporation ผลิตขึ้น และมีใช้งานอยู่ในปัจจุบัน

เบอร์ของไมโครคอนโทรลเลอร์	หน่วยความจำโปรแกรม	หน่วยความจำข้อมูล	จำนวนไทเมอร์/เคาน์เตอร์ 16 บิต
AT89C1051	แบบแฟลช ขนาด 1 กิโลไบต์	แรม 64 ไบต์	1
AT89C2051	แบบแฟลช ขนาด 2 กิโลไบต์	แรม 128 ไบต์	2
AT89C51	แบบแฟลช ขนาด 4 กิโลไบต์	แรม 128 ไบต์	2
AT89C52	แบบแฟลช ขนาด 8 กิโลไบต์	แรม 256 ไบต์	3
AT89C55	แบบแฟลช ขนาด 20 กิโลไบต์	แรม 256 ไบต์	3
AT89S8252	แบบแฟลช ขนาด 8 กิโลไบต์	แรม 256 ไบต์ อีอีพรอม 2 กิโลไบต์	3
AT89S53	แบบแฟลช ขนาด 12 กิโลไบต์	แรม 256 ไบต์	3

ตารางที่ 2.1

ในรูปที่ 2.1 แสดงโครงสร้างพื้นฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89CXX จะเห็นได้ว่า โครงสร้างของ AT89CXX จะเหมือนกับไมโครคอนโทรลเลอร์ตระกูล MCS-51 พื้นฐาน หากแต่แตกต่างกันเฉพาะหน่วยความจำโปรแกรมแบบแฟลชที่เพิ่มเติมเข้ามา หากเป็นไมโครคอนโทรลเลอร์ในอนุกรม 87XX หน่วยความจำโปรแกรมภายในจะเป็นแบบอีพรอม และบางเบอร์สามารถโปรแกรมได้ครั้งเดียว

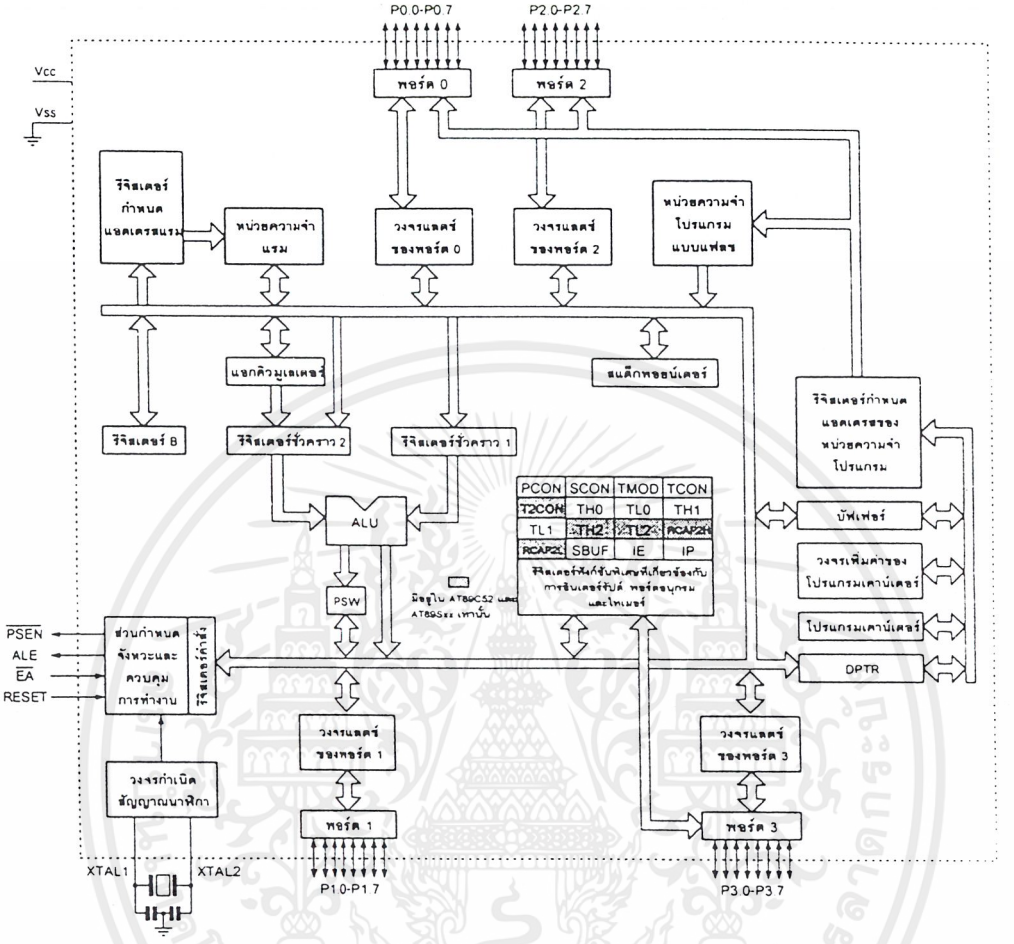


รูปที่ 2.1

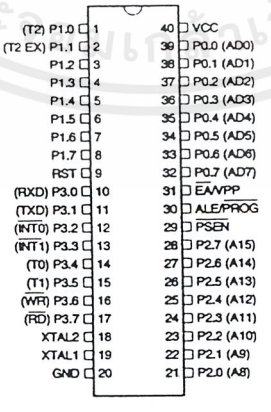
การจัดขาของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ทุกเบอร์จะมีสถาปัตยกรรมโครงสร้างภายในและการจัดวางขาใช้งานพื้นฐานเหมือนกันดังแสดงในรูปที่ 2.2 และ รูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2

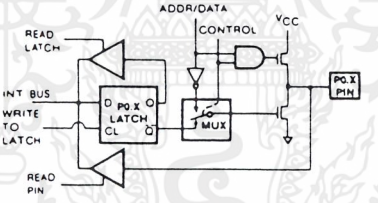


รูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยรายละเอียดขาการใช้งานขั้นต้นมีดังนี้

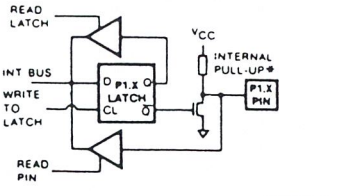
- ขา V_{cc} (40) ต่อกับไฟเลี้ยง +5 V
- ขา GND (20) เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ
- ขา **พอร์ต 0** (32-39) มีทั้งหมด 8 บิต (P0.0-P0.7) ใช้งานได้ 2 หน้าที่ คือ ส่งแอดเดรสไบต์ต่ำ (A0-A7) และข้อมูล (D0-D7) ออกไปให้หน่วยความจำภายนอก เมื่อทำการเขียนข้อมูลลงในหน่วยความจำภายนอกควบคุมด้วยขาควบคุม (Control) แสดงที่รูป 2.4 โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วย เพื่อสลับการทำงานให้เป็นได้ทั้งติดต่อแอดเดรสและส่งข้อมูล และอีกหน้าที่หนึ่งคือเป็นอินพุตพอร์ตหรือเอาท์พุตพอร์ตสำหรับใช้งานทั่วไป ถ้าต้องการกำหนดให้พอร์ต 0 ทำงานเป็นอินพุตพอร์ต ต้องส่งลอจิก "1" ไปยังพอร์ตนี้ จะมีผลให้ $\overline{Q}$ ของ ดี-ฟลิปฟลอป (D-FF) เป็น "0" ทำให้เฟ็ต (FET) ตัวล่างมีสภาวะปิด (OFF) สัญญาณที่ใช้อ่านขาอินพุตพอร์ต (ขาพอร์ต P0.X) จะใช้สัญญาณ Read Latch เมื่อถูกกระตุ้นที่ไตรสเตทบัฟเฟอร์ (Tri-State Buffer) ตัวบน



A. Port 0 Bit

รูปที่ 2.4

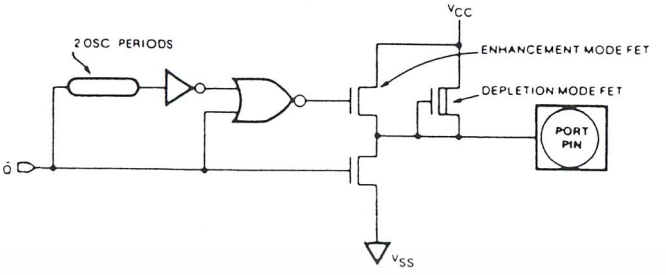
- ขา **พอร์ต** 1 (1-8) มีทั้งหมด 8 บิต (P1.0-P1.7) มีโครงสร้างคล้ายกับพอร์ต 0 ดังรูปที่ 2.5 พอร์ตนี้ทำหน้าที่เป็นอินพุตพอร์ตหรือเอาท์พุตพอร์ตสำหรับใช้งานทั่วไป แต่จะใช้ความต้านทานภายในพูลอัพแทน (Internal pull-up) ซึ่งแสดงดังรูปที่2.6



B. Port 1 Bit

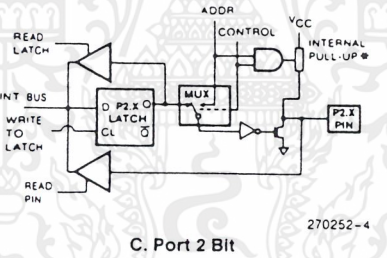
รูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6

- ขา **พอร์ต** 2 (21-28) **มีทั้งหมด** 8 บิต (P2.0-P2.7) มีโครงสร้างคล้ายกับพอร์ต 0 โดยมีเฟ็ตตัวล่างตัวเดียว ส่วนด้านบนใช้ความต้านทานพูลอัพแทน พอร์ตนี้ทำงาน 2 หน้าที่ คือ สามารถใช้เป็นพอร์ตสำหรับส่งแอดเดรสไบต์สูง (A8-A15) และเป็นอินพุตพอร์ตหรือเอาท์พุตพอร์ตสำหรับใช้งานทั่วไป เมื่อจะใช้งานเป็นอินพุตพอร์ต ต้องส่งลอจิก "1" มาที่พอร์ตนี้ก่อนเพื่อบังคับให้เฟ็ตอยู่ในสภาวะปิด ดังแสดงในรูปที่ 2.7



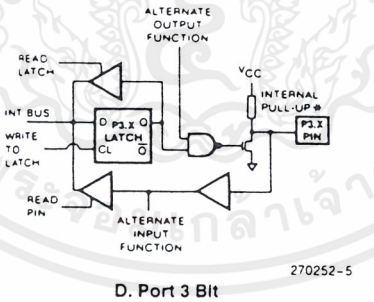
C. Port 2 Bit

รูปที่ 2.7

- ขา **พอร์ต** 3 (10-17) **มีทั้งหมด** 8 บิต (P3.0-P3.7) มีโครงสร้างคล้ายกับพอร์ต 1 แสดงดังรูปที่ 2.8 พอร์ตนี้ทำหน้าที่เป็นอินพุตพอร์ตหรือเอาท์พุตพอร์ตสำหรับใช้งานทั่วไป ถ้าจะให้พอร์ตนี้ทำงานเป็นอินพุตพอร์ต ก็ให้ส่งลอจิก "1" มาที่พอร์ตนี้ก่อน และนอกจากนั้นยังเป็นพอร์ตที่ทำหน้าที่การใช้งานพิเศษคือ ส่งสัญญาณควบคุมออกมา และรับสัญญาณเข้าไป ดังมีรายละเอียดขั้นต้นดังนี้
    - P3.0 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RXD (Serial Input Port)
    - P3.1 ใช้เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TXD (Serial Output Port)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- P3.2 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภาบนอกช่องที่ 0 หรือขา $\overline{\text{INT0}}$ (External Interrupt)
- P3.3 ใช้เป็นขาอินพุตสำหรับรับสัญญาณอินเตอร์รัปต์จากภาบนอกช่องที่ 1 หรือขา $\overline{\text{INT1}}$ (External Interrupt)
- P3.4 ใช้เป็นขาสำหรับรับสัญญาณไทเมอร์ที่ทำหน้าที่นับจำนวนไซเคิลของสัญญาณT1 หรือสัญญาณนาฬิกาก็ได้ จากภายนอกช่องที่ 0 หรือขา T0 (Timer/Counter 0 External Input)
- P3.5 ใช้เป็นขาสำหรับรับสัญญาณไทเมอร์ จากภายนอกช่องที่ 1 หรือขา T1 (Timer/Counter 1 External Input) ซึ่งมีหน้าที่การทำงานเหมือนกับ T0
- P3.6 ใช้เป็นขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำภายนอกในกรณีที่มีการเชื่อมต่อกับหน่วยความจำภายนอก หรือขา $\overline{\text{WR}}$ (External Data Memory Write Strobe)
- P3.7 ใช้เป็นขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำภายนอกในกรณีที่มีการเชื่อมต่อกับหน่วยความจำภายนอก หรือขา $\overline{\text{RD}}$ (External Data Memory Read Strobe)



D. Port 3 Bit

รูปที่ 2.8

- ขารีเซต (RST, 9) ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยการป้อนสัญญาณเพื่อรีเซต (ป้อนลอจิก "1") เข้าที่ขานี้โดยคงสถานะของสัญญาณการรีเซตไว้อย่างน้อย 2 แมชชีนไซเคิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างปกติ ภายหลังจากเกิดการรีเซตขึ้นแล้วค่าต่างๆในรีจิสเตอร์ใดๆ จะเป็นไปตามตารางที่ 2.2 ในหัวข้อที่ 2.4
- ขา $\overline{\text{ALE/PROG}}$ (Address Latch Enable/Program Pulse Input, 30) เป็นขาส่งสโตรบสำหรับใช้ในควบคุมการแลตช์แอดเดรสไบต์ต่ำ (A0-A7) ที่ส่งออกมาจากพอร์ต 0 เมื่อมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานหน่วยความจำภายนอก สัญญาณนี้จะแอกทีฟทุกๆ 2 ครั้ง ใน 1 แมชชีนไซเคิล (1/16 ของสัญญาณนาฬิกา) นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ ในรุ่นที่มีหน่วยความจำเป็นแบบอีอีพรอม

- ขา $\overline{\text{PSEN}}$ (Program Strobe Enable,29) เป็นขาที่ใช้ในการส่งสัญญาณเพื่อร้องขอการติดต่อกับหน่วยความจำภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ตัวไมโครคอนโทรลเลอร์จะส่งสัญญาณนี้ออกมา 2 ครั้ง ในแต่ละแมชชีนไซเคิล แต่ถ้าเป็นการอ่านค่าจากหน่วยความจำโปรแกรมภายใน จะไม่มีการส่งสัญญาณใดๆออกที่ขานี้
- ขา $\overline{\text{EA}}$/Vpp (External Access Enable/Programming Voltage Input, 31) ใช้สำหรับการเลือกการติดต่อหน่วยความจำโปรแกรมภายนอก หรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าป้อนลอจิก "0" เข้าที่ขานี้ จะเป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่หากป้อนลอจิก "1" เข้าที่ขานี้ จะเป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายในตัวไมโครคอนโทรลเลอร์ นอกจากนี้ ที่ขานี้ยังใช้เป็นขาอินพุตสำหรับรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับในไมโครคอนโทรลเลอร์ แบบแฟลชต้องการแรงดันสำหรับการโปรแกรม +12 โวลต์
- ขา XTAL1 ใช้ต่อกับคริสตอลภายนอกโดยเป็นอินพุตเข้าสู่วงจรสร้างสัญญาณนาฬิกา (Oscillator)
- ขา XTAL2 ใช้ต่อกับคริสตอลภายนอกโดยเป็นเอาท์พุตของวงจรสร้างสัญญาณนาฬิกา (Oscillator)

2.2 การนำพอร์ตมาใช้งาน

เนื่องจากพอร์ตทั้งหมดของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชสามารถเป็นได้ทั้งอินพุตและเอาท์พุต ดังนั้นจะขอกล่าวถึงการกำหนดลักษณะการทำงานให้แก่พอร์ตของไมโครคอนโทรลเลอร์ แบบแฟลช

การใช้งานเป็นพอร์ตอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการกำหนดให้เป็นพอร์ตอินพุต ต้องเริ่มต้นด้วยการเขียนข้อมูล "1" มาที่แต่ละบิตของพอร์ตที่ต้องการใช้งานเป็นอินพุต เพื่อหยุดการทำงานของเฟ็ตที่ใช้ในการขับสัญญาณเอาท์พุตของบิตนั้นๆ ทำให้ขาสัญญาณของพอร์ตเชื่อมต่อเข้ากับวงจรพูลอัพโดยตรง ส่งผลให้ขาพอร์ตนั้นมีลอจิกเป็น "1" สามารถรับสัญญาณลอจิก "0" จากอุปกรณ์ภายนอกได้ง่าย สัญญาณข้อมูลจากอุปกรณ์ภายนอกจะถูกส่งเข้ามาแล้วเก็บไว้ใน วงจรบัฟเฟอร์ภายในพอร์ต แล้วรอให้หน่วยประมวลผลกลางมาอ่านค่าไป เมื่อเป็นเช่นนี้อุปกรณ์ภายนอกที่เชื่อมต่อกับพอร์ตอินพุตของไมโครคอนโทรลเลอร์ แบบแฟลชควรกำหนดให้ทำงานในสภาวะลอจิก "0" จะดีและสะดวกสุด (ซึ่งในปัจจุบันอุปกรณ์อินพุตที่เชื่อมต่อกับไมโครคอนโทรลเลอร์แทบทั้งหมดทำงานที่ลอจิก "0" แล้ว)

และเนื่องจากที่ขาพอร์ต 0 ไม่มีวงจรพูลอัพภายใน หากมีการนำพอร์ต 0 ไปใช้งานเป็นพอร์ตอินพุตจะต้องต่อความต้านทานพูลอัพภายนอกเข้าที่ขาพอร์ต 0 ทุกขาด้วย

การใช้งานเป็นพอร์ตเอาท์พุต

โดยปกติแล้ว ขาพอร์ตจะกำหนดให้มีลักษณะเป็นเอาท์พุตอยู่แล้ว กล่าวคือ เมื่อต้องการส่งข้อมูล "0" ออกไปทางเอาท์พุตก็ให้เขียนข้อมูล "0" ไปยังวงจรแลตช์ซึ่งก็จะส่งต่อไปขับเฟ็ต ทำให้เฟ็ตทำงาน ที่ขาพอร์ตที่กำหนดให้ทำงานก็จะเกิดลอจิก "0" ขึ้น ในทางตรงกันข้าม หากต้องการส่งข้อมูล "1" ออกไป ก็ให้เขียนข้อมูล "1" ออกไปยังวงจรแลตช์ วงจรขับก็จะหยุดทำงาน ทำให้ที่ขาพอร์ตเชื่อมต่อกับวงจรพูลอัพภายในเกิดเป็นลอจิก "1" ที่ขาพอร์ตนั้น ซึ่งคล้ายกับการกำหนดให้เป็นขาอินพุตมาก เพียงแต่แตกต่างกันที่กระบวนการในการเคลื่อนย้ายข้อมูล โดยถ้าเป็นอินพุตจะมีสัญญาณมาอ่านข้อมูลที่บัฟเฟอร์ แต่ถ้าเป็นเอาท์พุตจะไม่มีการอ่านข้อมูลที่บัฟเฟอร์แต่อย่างใด เว้นแต่กรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกมาทางเอาท์พุต

เมื่อใช้งานพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเป็นพอร์ตเอาท์พุต แต่ละขา (หรือแต่ละบิต) ของแต่ละพอร์ตมีความสามารถในการจ่ายกระแสหรือที่เรียกว่า กระแสซอร์ส (Source Current) ได้สูงสุด 10 มิลลิแอมแปร์ และทุกขารวมกันในแต่ละพอร์ต (ทั้ง 8 บิต) สูงสุด 26 มิลลิแอมแปร์สำหรับพอร์ต 0 และ 15 มิลลิแอมแปร์ สำหรับพอร์ต1-3 ในกรณีที่ใช้งานทุกพอร์ตเอาท์พุตจะสามารถจ่ายกระแสได้รวมกันสูงสุด 71 มิลลิแอมแปร์ ดังนั้นในการใช้งานเป็นพอร์ตเอาท์พุตเพื่อไม่ให้เกิดปัญหาเกี่ยวกับความสามารถในการจ่ายกระแสจึงควรต่อวงจรบัฟเฟอร์ทางเอาท์พุตเพื่อช่วยในการขับกระแสอีกทางหนึ่ง

การอ่านค่าลอจิกจากพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครคอนโทรลเลอร์ MCS-51แบบแฟลชสามารถอ่านค่าลอจิกจากพอร์ตได้ 2 ลักษณะคืออ่านค่าจากพอร์ตโดยตรง และอ่านค่าจากวงจรแลตช์ของแต่ละพอร์ต

ในกรณีที่พอร์ตต่อกับขาเบสทรานซิสเตอร์ชนิด NPN และขาอีมิตเตอร์ของทรานซิสเตอร์ตัวนั้นต่อลงกราวด์ หากมีการส่งข้อมูล "1" ไปยังทรานซิสเตอร์ จะทำให้ทรานซิสเตอร์ทำงานสถานะลอจิกที่ขาพอร์ตจะเป็น "0" เนื่องจากเมื่อทรานซิสเตอร์ทำงาน จะเสมือนว่าขาพอร์ตนั้นถูกต่อลงกราวด์ ทำให้หากอ่านค่าลอจิกที่ขาพอร์ตจะได้ผลตรงกันข้ามกับที่ส่งออกมา แต่หากทำงานอ่านค่าลอจิกที่วงจรแลตช์จะได้ค่าที่ตรงกับค่าที่ต้องการส่งจริง ดังนั้น ในการอ่านค่าลอจิกพอร์ตจึงต้องเลือกวิธีการให้เหมาะสมกับอุปกรณ์ที่นำมาต่อด้วย

2.3 จังหวะการทำงานของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

ในการประมวลผลคำสั่งของหน่วยประมวลผลกลางจะมีขั้นตอนหลักๆ 2 ขั้นตอน คือ กระบวนการเฟตช์ (Fetch) เป็นการเรียกคำสั่งออกจากหน่วยความจำโปรแกรมแล้วทำการแปลงรหัสคำสั่งนั้นเป็นภาษาเครื่องเพื่อเตรียมการประมวลผล ขั้นตอนต่อมาคือ กระบวนการเอ็กซิคิวต์ (Execute) เป็นการกระทำตามคำสั่งที่กำหนดหรือตามที่เฟตช์ขึ้นมาโดยกระบวนการก่อนหน้านี้ เมื่อทำการเอ็กซิคิวต์คำสั่งเรียบร้อยแล้ว ก็จะเริ่มกระบวนการเฟตช์คำสั่งใหม่ต่อไป

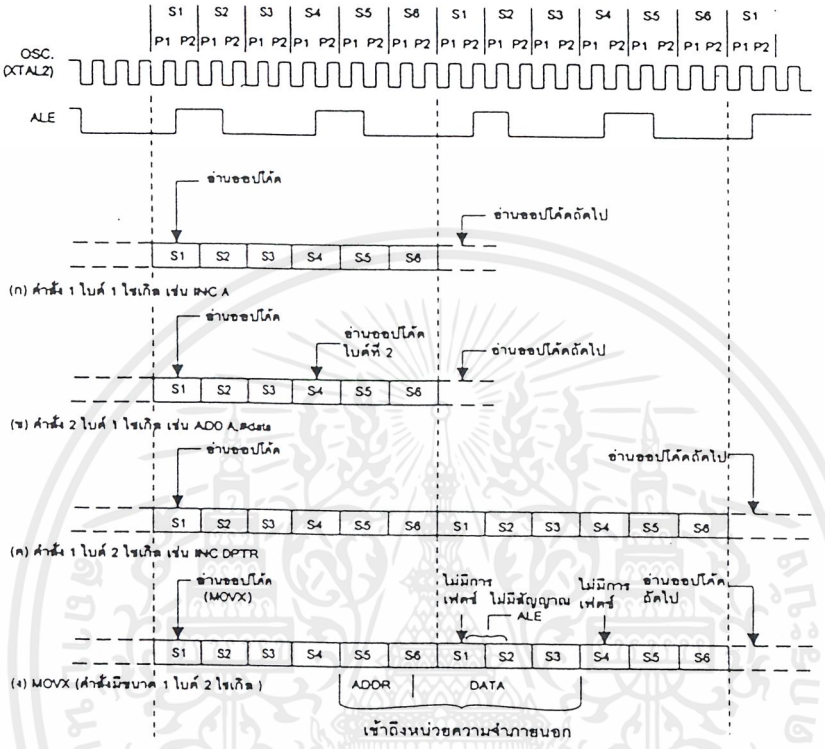
เมื่อเริ่มจ่ายไฟให้แก่ไมโครคอนโทรลเลอร์ จะเกิดการรีเซตในลักษณะที่เรียกว่า เพาเวอร์ออนรีเซต (Power-on Reset) ทำให้หน่วยประมวลผลกลางไปเริ่มต้นการทำงานที่แอดเดรส 0000H ของหน่วยความจำโปรแกรม จังหวะการทำงานของหน่วยประมวลผลกลางจะเป็นไปตามรูปแบบ โดยได้รับการกำหนดมาจากรอบการทำงาน (Machine Cycle) โดยไดอะแกรมเวลาในรูปที่ 2.9 แสดงจังหวะการทำงานของไมโครคอนโทรลเลอร์ โดยใน 1 รอบการทำงานจะแบ่งย่อยออกเป็น 6 สเตต (State) กำหนดชื่อเป็น S1-S6 ในแต่ละสเตตมีค่าเวลาเท่ากับ 2 คาบเวลาของสัญญาณนาฬิกา ถ้าสัญญาณนาฬิกามีความถี่ 12 เมกะเฮิร์ตซ์ จะมีคาบเวลาเท่ากับ 1 มิลลิวินาที คาบเวลาทั้งสองภายในหนึ่งสเตตเรียกว่า เฟส 1 (Phase1) และเฟส 2 (Phase 2)

ความเร็วในการทำงานภายในของไมโครคอนโทรลเลอร์เท่ากับ

ความถี่ของสัญญาณนาฬิกา (ค่าของคริสตอลที่ต่ออยู่ที่ขา XTAL1 และ XTAL2) / 12

เวลา 1 รอบการทำงาน = 1 / ความเร็วในการทำงานภายในของไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9

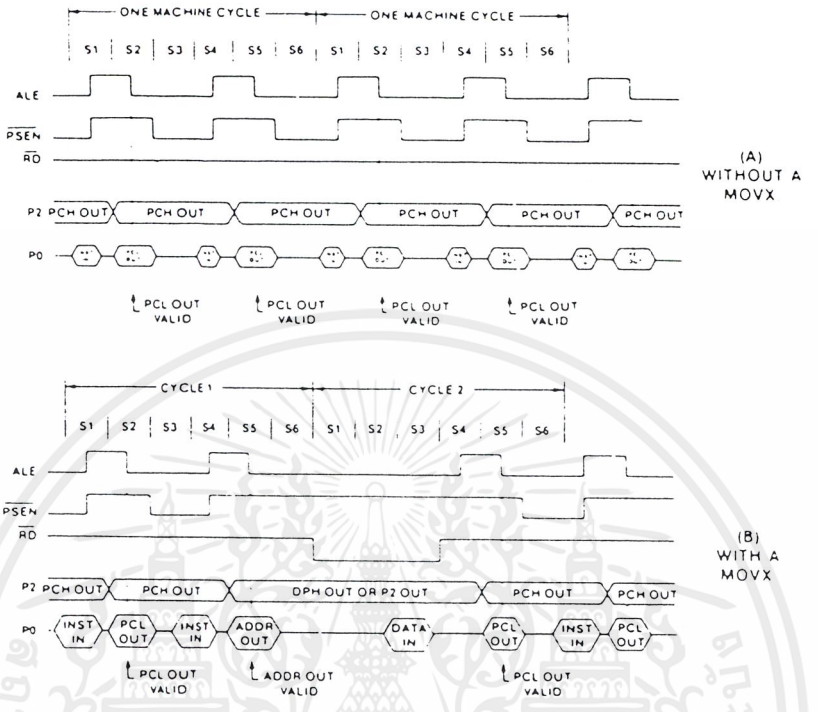
รูปที่ 2.9 (ก) แสดงการทำงานของคำสั่ง INC A ซึ่งเป็นคำสั่ง 1 ไบต์ ทำงานเสร็จใน 1 รอบการทำงาน

รูปที่ 2.9 (ข) แสดงการทำงานของคำสั่ง ADD A,#DATA ซึ่งเป็นคำสั่ง 2 ไบต์ แต่ทำงานเสร็จใน 1 รอบการทำงาน

รูปที่ 2.9 (ค) แสดงการทำงานของคำสั่ง INC DTPR ซึ่งเป็นคำสั่ง 1 ไบต์ แต่ทำงานเสร็จใน 2 รอบการทำงาน

รูปที่ 2.9 (ง) แสดงการทำงานของคำสั่ง MOVX ซึ่งเป็นคำสั่ง 1 ไบต์ แต่ทำงานเสร็จใน 2 รอบการทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 (A) และ (B)

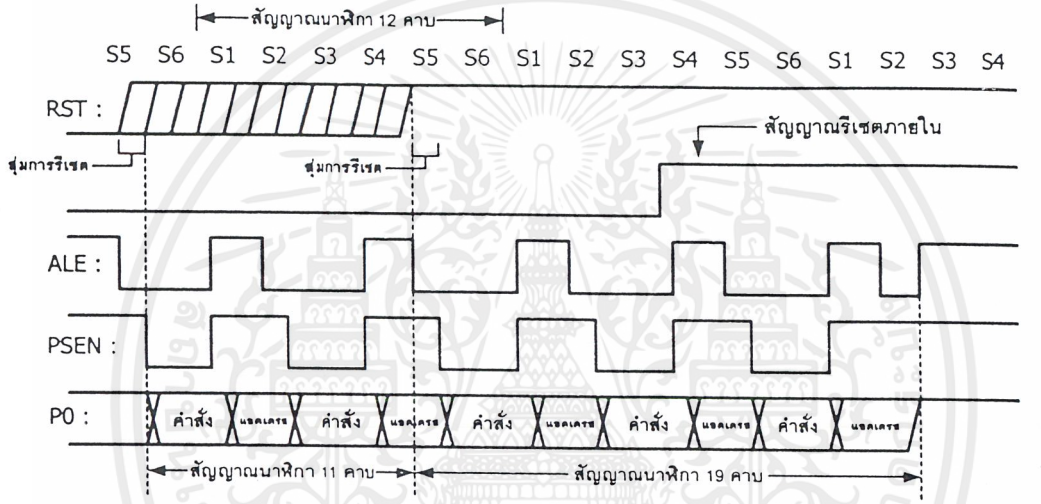
ในรูปที่ 2.9 แสดงสัญญาณและไดอะแกรมเวลาของการเข้าถึงหน่วยความจำโปรแกรมภายนอก โดยในรูปที่ 2.9 (A) เป็นไดอะแกรมเวลาในขณะที่ยังไม่มีการกระทำคำสั่ง MOVX สัญญาณที่ขา ALE และ $\overline{\text{PSEN}}$ จะเกิดการแอกทีฟ 2 ครั้งภายในหนึ่งรอบการทำงาน ในทุกครั้งที่ ALE เกิดการแอกทีฟที่พอร์ต 0 จะมีค่าของรีจิสเตอร์ PC ในไบต์ต่ำออกมา ในขณะที่พอร์ต 2 ก็จะมีค่าของ PC ในไบต์สูงเพื่อชี้ไปยังแอดเดรสต่อไปที่ต้องการไปดำเนินการ สำหรับขา $\overline{\text{PSEN}}$ ก็จะเกิดการแอกทีฟเมื่อมีการติดต่อกับหน่วยความจำโปรแกรมภายนอกในกรณีที่กระทำคำสั่ง MOVX เพื่อเข้าถึงหน่วยความจำข้อมูลภายนอก ที่ขา $\overline{\text{PSEN}}$ จะไม่เกิดการแอกทีฟ 2 ครั้งภายใน 1 รอบการทำงาน เนื่องจากบัสแอดเดรสและบัสข้อมูลจะถูกใช้ในการติดต่อหน่วยความจำข้อมูลภายนอกแทน คือจะเป็นช่วงเวลาในการอ่านข้อมูลจากหน่วยความจำข้อมูล โดยใช้สัญญาณ $\overline{\text{RD}}$ แต่สำหรับสัญญาณ ALE ยังคงแอกทีฟตามจังหวะการทำงานเหมือนเดิม แสดงในรูปที่ 2.9 (B)

อาจสรุปง่ายๆว่าการอ่านข้อมูลจากหน่วยความจำโปรแกรม จะใช้สัญญาณ $\overline{\text{PSEN}}$ และการอ่านข้อมูลจากหน่วยความจำข้อมูล จะใช้สัญญาณ $\overline{\text{RD}}$ ส่วนสัญญาณ ALE คือสัญญาณที่ใช้ในการแลตช์แอดเดรส A0-A7 นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 กระบวนการรีเซตในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

การรีเซตในไมโครคอนโทรลเลอร์ เกิดขึ้นได้เพียงกรณีเดียวคือที่ขารีเซต (ขา9) ได้รับลอจิก "1" อย่างน้อยเป็นเวลา 2 รอบการทำงาน หรือ 24 คาบเวลาของสัญญาณนาฬิกา โดยที่วงจรกำเนิดสัญญาณยังคงทำงานอยู่ เมื่อเกิดการรีเซตขึ้น ไมโครคอนโทรลเลอร์จะสร้างสัญญาณรีเซตภายในขึ้นเพื่อตอบสนองการรีเซตจากภายนอก ดังแสดงไดอะแกรมเวลาของกระบวนการรีเซตภายในไมโครคอนโทรลเลอร์ในรูปที่ 2.10



รูปที่ 2.10

สัญญาณรีเซตที่เกิดขึ้นจากภายนอกจะไม่มีความสอดคล้องกับสัญญาณนาฬิกาแต่อย่างใด เนื่องจากการรีเซตอาจเกิดขึ้นเมื่อใดก็ได้ โดยจะมีการสุ่มตรวจสอบสถานะลอจิกที่ขารีเซต ในสเตต 5 เฟส 2 ของทุกรอบการทำงาน เมื่อสามารถตรวจจับลอจิก "1" ที่ขารีเซตได้ และเมื่อเกิดสัญญาณรีเซตภายในเกิดขึ้น กระบวนการรีเซตของไมโครคอนโทรลเลอร์จะเริ่มต้นขึ้นอย่างเป็นทางการ ขาพอร์ตทั้งหมดยังคงรักษาข้อมูลปัจจุบันไว้ได้อีก 19 คาบเวลาของสัญญาณนาฬิกาหลังจากที่ขารีเซตได้รับลอจิก "1" จากนั้นจะเปลี่ยนแปลงค่ากลายเป็น FFH ทั้งหมด

สัญญาณ ALE และ $\overline{\text{PSEN}}$ จะกลายเป็นลอจิก "1" เมื่อการรีเซตเกิดขึ้นอย่างสมบูรณ์ และเมื่อขารีเซตถูกทำให้กลายเป็นลอจิก "0" อันเป็นการแจ้งให้ทราบว่า สามารถเริ่มต้นกระบวนการทำงานปกติได้แล้ว สัญญาณ ALE และ $\overline{\text{PSEN}}$ จะเกิดขึ้นใหม่อีกครั้งหลังจากที่ขารีเซตเป็นลอจิก "0" หนึ่งถึงสองรอบการทำงาน ทั้งนี้รอให้อุปกรณ์ที่ต่อร่วมอยู่กับไมโครคอนโทรลเลอร์อยู่ในสภาวะพร้อมทำงาน และสามารถทำงานในจังหวะที่สอดคล้องกับไมโครคอนโทรลเลอร์ได้อย่างเหมาะสม

ผลของการรีเซตที่มีต่อรีจิสเตอร์ภายในไมโครคอนโทรลเลอร์นั้นแสดงในตารางที่2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1. AT89C52 SFR Map and Reset Values

0F8H								0FFH
0F0H	B 00000000							0F7H
0E8H								0EFH
0E0H	ACC 00000000							0E7H
0D8H								0DFH
0D0H	PSW 00000000							0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000		0CFH
0C0H								0C7H
0B8H	IP XX000000							0BFH
0B0H	P3 11111111							0B7H
0A8H	IE 0X000000							0AFH
0A0H	P2 11111111							0A7H
98H	SCON 00000000	SBUF XXXXXXXX						9FH
90H	P1 11111111							97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000		8FH
80H	P0 11111111	SP 00000111	DPL 00000000	DPH 00000000				PCON 0XXX0000 87H

AT89C52

ตารางที่ 2.2

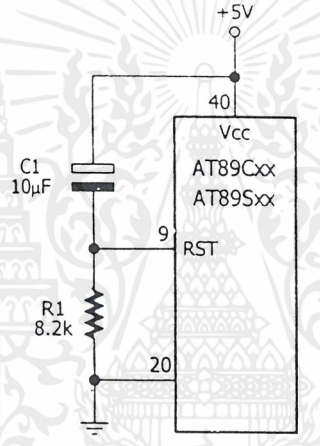
กลไกที่สำคัญอีกประการหนึ่งของกระบวนการรีเซตคือ ค่าของหน่วยความจำข้อมูลแรมภายในจะไม่ได้รับผลกระทบจากการรีเซตแต่อย่างใด ดังนั้นในการเขียนโปรแกรมควบคุม ที่ส่วนต้นของโปรแกรมจึงควรมีการเคลียร์ค่าของหน่วยความจำข้อมูลแรมภายในที่ต้องใช้งานเสมอ เพื่อป้องกันการใช้ข้อมูลที่เกิดขึ้นก่อนการรีเซตมาประมวลผลเนื่องจากโอกาสในการรับรู้ว่าข้อมูลของหน่วยความจำก่อนการรีเซตมีค่าเท่าใดมีน้อยมาก

2.4.1 เพาเวอร์ออนรีเซต (Power-On Reset)

เป็นการรีเซตที่เกิดขึ้นหลังจากมีการจ่ายไฟเลี้ยงให้แก่ไมโครคอนโทรลเลอร์ โดยมีวงจรพื้นฐานเพื่อทำให้เกิดการรีเซตแบบนี้ในรูปที่ 2.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

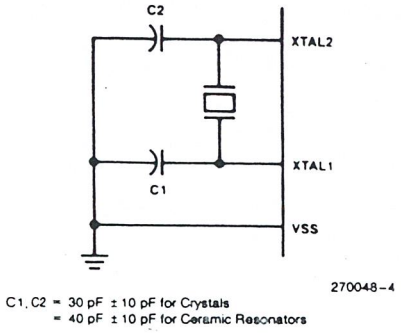
การทำงานของการรีเซตแบบนี้ กล่าวคือ เมื่อเริ่มต้นจ่ายไฟ ตัวเก็บประจุไฟฟ้าทำให้มีกระแสไหลผ่านตัวต้านทาน เกิดแรงดันตกคร่อมตัวต้านทาน ส่งผลให้ขา RST ได้รับแรงดันไฟตรงประมาณ +5 โวลต์ ซึ่งก็คือระดับแรงดันของลอจิก "1" โดยตรง จนกระทั่งเวลาผ่านไป ตัวเก็บประจุสามารถประจุแรงดันแรงดันไฟฟ้ามากขึ้นเรื่อยๆ ทำให้แรงดันตกคร่อมตัวต้านทานลดลงตามลำดับ และกลายเป็นศูนย์เมื่อตัวเก็บประจุสามารถประจุแรงดันได้อย่างเต็มที่ ดังนั้นการกำหนดค่าของตัวเก็บประจุและตัวต้านทานในวงจรนี้มีความสำคัญต่อการรีเซตของไมโครคอนโทรลเลอร์อย่างมาก เพราะต้องกำหนดให้ค่าเวลาของการประจุแรงดันของตัวเก็บประจุยาวนานมากพอที่จะทำให้ขารีเซต ได้รับลอจิก "1" อย่างน้อยเป็นเวลา 2 รอบการทำงาน



รูปที่ 2.11

2.5 วงจรนาฬิกาของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

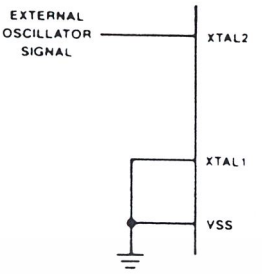
การต่อมีอยู่ด้วยกัน 2 รูปแบบ คือ แบบใช้วงจรนาฬิกาภายในชิพ (On-Chip Oscillator) และแบบต่อกับวงจรสร้างสัญญาณนาฬิกาจากภายนอก (External Clock Source) รูปที่ 2.12 และรูปที่ 2.13 ได้แสดงรูปแบบการต่อทั้งสองแบบตามลำดับ



รูปที่ 2.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



รูปที่ 2.13

2.6 การจัดหน่วยความจำของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

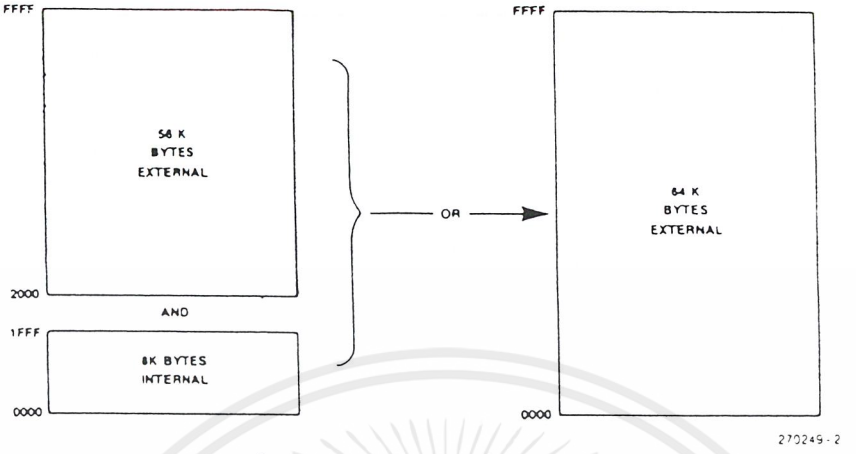
ในไมโครคอนโทรลเลอร์แบบแฟลชมีหน่วยความจำภายในหลักๆอยู่ 2 ส่วนคือ หน่วยความจำโปรแกรมและหน่วยความจำข้อมูล ซึ่งก็มีขนาดและการจัดสรรแตกต่างกันไปในแต่ละเบอร์ ซึ่งในหัวข้อนี้จะกล่าวถึงรายละเอียดของการจัดสรรหน่วยความจำภายในไมโครคอนโทรลเลอร์แบบแฟลช การเชื่อมต่อกับหน่วยความจำภายนอก และข้อมูลเบื้องต้นของรีจิสเตอร์ฟังก์ชันพิเศษที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์แบบแฟลชดังนี้

2.6.1 หน่วยความจำโปรแกรม (Program Memory)

ในระบบของไมโครคอนโทรลเลอร์ จำเป็นต้องมีหน่วยความจำสำหรับบรรจุคำสั่ง หรือโปรแกรมที่ผู้ใช้พัฒนาขึ้นจัดเก็บไว้ภายในหน่วยความจำ ที่เรียกว่าหน่วยความจำโปรแกรม โดยอาจจะประกอบอยู่ภายในตัวไอซีของไมโครคอนโทรลเลอร์เอง หรือเป็นไอซีของหน่วยความจำแยกออกต่างหากได้ ในกรณีหลังจำเป็นจะต้องมีการใช้พอร์ตอินพุต/เอาต์พุต ทำหน้าที่เป็นบัสแอดเดรสและข้อมูล เพื่อให้สามารถทำการเชื่อมต่อเข้ากับไอซีหน่วยความจำมาตรฐานทั่วไปได้

หน่วยความจำโปรแกรมใช้เก็บข้อมูลและคำสั่งใช้งานต่าง ๆ ซึ่งแม้ว่าจะไม่มีการกระแสไฟฟ้าให้กับระบบ ข้อมูลเหล่านี้ก็ยังคงอยู่ ไม่สูญหาย โครงสร้างของหน่วยความจำโปรแกรม มีลักษณะเช่นเดียวกับหน่วยความจำที่บรรจุอยู่ในไอซีหน่วยความจำประเภทต่างๆ เช่นหน่วยความจำแบบรอม (ROM) หรืออีพีรอม (EPROM) เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
45716



รูปที่ 2.14 แสดงการจัดพื้นที่หน่วยความจำโปรแกรม
สำหรับไมโครคอนโทรลเลอร์เบอร์ AT89C52

การจัดหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์แบบแฟลช สามารถติดต่อหน่วยความจำโปรแกรมได้สูงสุด 64 กิโลไบต์ ซึ่งในไมโครคอนโทรลเลอร์เบอร์ AT89C51 และ AT89C52 นั้นมีหน่วยความจำโปรแกรมภายในแบบรอม ขนาด 4 และ 8 กิโลไบต์ ตามลำดับ ฉะนั้นถ้าต้องการใช้หน่วยความจำโปรแกรมภายนอก ก็จะสามารถติดต่อได้อีก 60 และ56 กิโลไบต์ ตามลำดับ

ไมโครคอนโทรลเลอร์เบอร์ต่างๆ สามารถขยายให้ใช้งานหน่วยความจำภายนอกได้ทั้งสิ้น โดยในกรณีที่มีหน่วยความจำโปรแกรมภายในอยู่แล้ว การอ้างตำแหน่งแอดเดรสที่มีทั้งในหน่วยความจำโปรแกรมภายในและภายนอกนั้น จะต้องทำการพิจารณาระดับลอจิกของสัญญาณ $\overline{EA}$ ในขณะนั้นด้วย

การเชื่อมต่อหน่วยความจำโปรแกรมภายนอกกับไมโครคอนโทรลเลอร์

เนื่องจากระบบบัสแอดเดรส และบัสข้อมูลของไมโครคอนโทรลเลอร์ เป็นลักษณะแบบใช้การมัลติเพล็กซ์ จากพอร์ตเดียวกัน กล่าวคือ ในระยะเวลาเริ่มต้น เส้นสัญญาณเหล่านี้ของพอร์ตจะใช้ในการส่งค่าแอดเดรส ของตำแหน่งที่ต้องการติดต่อด้วย ในช่วงเวลาต่อมาจึงจะเปลี่ยนไปเป็นสภาวะอิมพีแดนซ์สูงเพื่อใช้งานในฐานะของบัสข้อมูล แต่เนื่องจากว่า EPROM ที่ใช้งานกันทั่วไปนั้นไม่ใช้การมัลติเพล็กซ์ และมีขาสัญญาณบัสแอดเดรสและบัสข้อมูลแยกออกจากกันโดยชัดเจน ดังนั้น การเชื่อมต่อ EPROM เพื่อทำหน้าที่เป็นหน่วยความจำโปรแกรม จึงจำเป็นต้องมีวงจรประเภทแลตช์ (Latch) ประกอบเพิ่มขึ้นมา เพื่อทำการค้างค่าของแอดเดรสที่ส่งออกมาจากไมโครคอนโทรลเลอร์ ในช่วงเวลาแรกให้กับขาสัญญาณแอดเดรสของอีพีรอมต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

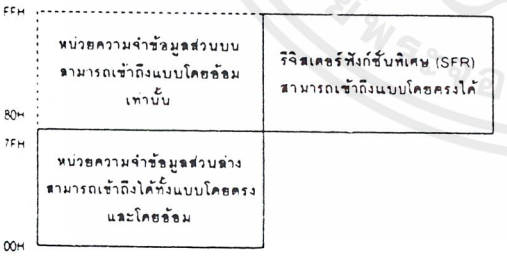
2.6.2 หน่วยความจำข้อมูล (Data Memory)

มีหน้าที่สำหรับเก็บข้อมูล หรือตัวแปรที่เกิดขึ้นในขณะที่กำลังประมวลผลโปรแกรมไว้เป็นการชั่วคราว โดยพื้นฐาน แล้ว หน่วยความจำข้อมูลจัดว่าเป็นหน่วยความจำแรม แบบ สแตติก (Static) ดังนั้นเมื่อไม่มีการจ่ายไฟฟ้าให้กับระบบ ก็จะมีผลให้ข้อมูลที่จัดเก็บไว้ภายในหน่วยความจำสูญหายไป พื้นที่ของหน่วยความจำข้อมูลของไมโครคอนโทรลเลอร์แบบแฟลช สามารถมีได้สูงสุดไม่เกิน 64 กิโลไบต์ และแยกเป็นสองลักษณะตามตำแหน่งที่ตั้งของหน่วยความจำนั้น คือ หน่วยความจำข้อมูลภายใน ซึ่งเป็นแรมที่อยู่ภายในตัวของไมโครคอนโทรลเลอร์เอง และหน่วยความจำข้อมูลภายนอก (External Data Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำแรมมาเพิ่มเติมเข้าไปในวงจร ลักษณะเดียวกับการนำไอซีอีพีรอมมาใช้งานเป็นหน่วยความจำโปรแกรมนั่นเอง

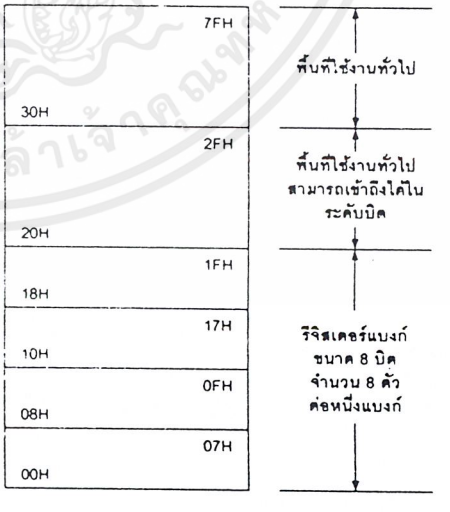
หน่วยความจำข้อมูลภายใน (Internal Data Memory)

หน่วยความจำข้อมูลภายในของไมโครคอนโทรลเลอร์แบบแฟลชเบอร์ AT89C52นั้น มีจำนวนทั้งหมด 256 ไบต์ โดยแยกออกได้เป็น 2 ลักษณะ คือพื้นที่เฉพาะสำหรับประมวลผลกลางใช้งานเท่านั้น ซึ่งก็คือ รีจิสเตอร์ และพื้นที่ใช้งานทั่วไปสำหรับโปรแกรมใช้งานที่ผู้ใช้สร้างขึ้นมา

สำหรับการจัดสรรหน่วยความจำข้อมูลภายในแบ่งเป็น 3 ส่วนคือหน่วยความจำข้อมูลส่วนล่าง (Lower), ส่วนบน (Upper) และรีจิสเตอร์ฟังก์ชันพิเศษ (SFR:Special Function Register) ซึ่งแต่ละส่วนมีขนาด 128 ไบต์



รูปที่ 2.15 ก



รูปที่ 2.15 ข

แสดงการจัดสรรพื้นที่ของหน่วยความจำข้อมูลภายในไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บริเวณแอดเดรส 00H – 1FH จำนวน 32 ไบต์ จำแนกออกเป็นกลุ่ม (หรือแบงก์ Bank) ข้อมูลจำนวน 8 ไบต์ รวมทั้งหมดสี่กลุ่ม พื้นที่ข้อมูลในแต่ละกลุ่มจะถูกใช้งานในฐานะรีจิสเตอร์ใช้งานทั่วไป ซึ่งมีชื่อเรียกว่า รีจิสเตอร์ R0 – R7 ดังตารางที่ 2.3

แอดเดรส	รีจิสเตอร์แบงก์	ชื่อรีจิสเตอร์ใช้งาน
00H - 07H	0	R0 - R7
08H - 0FH	1	R0 - R7
10H - 17H	2	R0 - R7
18H - 1FH	3	R0 - R7

ตารางที่ 2.3

ดังนั้นการใช้งานผู้ใช้จะต้องให้ความระมัดระวังว่าต้องการรีจิสเตอร์นั้นๆ จากแบงก์ใด การสวิตช์เลือกแต่ละกลุ่มของรีจิสเตอร์นี้ก็ทำได้ง่าย เพียงการกำหนดค่าของบิตที่อยู่ภายในรีจิสเตอร์ PSW เท่านั้น ตามตารางที่ 2.4

รีจิสเตอร์	บิต RS0	บิต RS1	ตำแหน่งหน่วยความจำ
แบงก์ 0	0	0	0000H
แบงก์ 1	0	1	0008H
แบงก์ 2	1	0	0010H
แบงก์ 3	1	1	0018H

ตารางที่ 2.4

อย่างไรก็ตามโดยทั่วไปก็มักจะมีการใช้งานรีจิสเตอร์ R0 – R7 เฉพาะใน แบงก์ 0 เท่านั้น ดังนั้นพื้นที่ของแบงก์อื่นๆ ที่เหลือก็สามารถนำมาใช้ในลักษณะของหน่วยความจำข้อมูลภายในปกติด้วยการอ้างถึงหมายเลขของแอดเดรสนั้นๆ โดยตรง

บริเวณแอดเดรส 20H – 2FH จำนวน 16 ไบต์ บริเวณพื้นที่เป็นส่วนสำหรับผู้ใช้ซึ่งจะมีความพิเศษต่างไปจากหน่วยความจำส่วนอื่นๆ เนื่องจากผู้ใช้อาจสามารถอ้างถึงหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บริเวณนี้ได้ทั้งในลักษณะของไบต์ข้อมูลเช่นปกติ หรือจะเป็นบิตข้อมูลได้โดยตรง ดังนั้นหากมองในลักษณะบิตข้อมูลแล้ว ก็จะมีพื้นที่ตัวแปรแบบบิตให้ใช้งานได้มากถึง 128 บิต โดยตำแหน่งแรกของบิตจะเป็นบิตซึ่งเริ่มต้นนับจากบิตที่นัยสำคัญต่ำสุด (LSB) ของแอดเดรส 20H เรื่องไปจนกระทั่งถึงบิตที่ 127 ซึ่งเป็นบิตที่นัยสำคัญสูดสุด (MSB) ของแอดเดรส 2FH

รูปที่ 2.16 แสดงหน่วยความจำข้อมูลภายในบริเวณที่อ้างถึงได้แบบบิต

หน่วยความจำขนาด 128 ไบต์ถัดมา

พื้นที่ตั้งแต่บริเวณแอดเดรส 80H – FFH เป็นบริเวณของหน่วยความจำที่มีการใช้งานเฉพาะจาก AT89C52 เท่านั้น โดยจะนำมาใช้เป็นตำแหน่งของรีจิสเตอร์หน้าที่พิเศษจำนวน 20 ตำแหน่ง ดังแสดงในรูปที่ 2.17นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Symbol	Name	Address
*ACC	Accumulator	0E0H
*B	B Register	0F0H
*PSW	Program Status Word	0D0H
SP	Stack Pointer	81H
DPTR	Data Pointer 2 Bytes	
DPL	Low Byte	82H
DPH	High Byte	83H
*P0	Port 0	80H
*P1	Port 1	90H
*P2	Port 2	0A0H
*P3	Port 3	0B0H
*IP	Interrupt Priority Control	0B8H
*IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
*TCON	Timer/Counter Control	88H
*+T2CON	Timer/Counter 2 Control	0C8H
TH0	Timer/Counter 0 High Byte	8CH
TL0	Timer/Counter 0 Low Byte	8AH
TH1	Timer/Counter 1 High Byte	8DH
TL1	Timer/Counter 1 Low Byte	8BH
-TH2	Timer/Counter 2 High Byte	0CDH
+TL2	Timer/Counter 2 Low Byte	0CCH
+RCAP2H	T/C 2 Capture Reg. High Byte	0CBH
-RCAP2L	T/C 2 Capture Reg. Low Byte	0CAH
*SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H

* = Bit addressable
- = 8052 only

รูปที่ 2.17แสดงรีจิสเตอร์ใช้งานพิเศษ

สำหรับไมโครคอนโทรลเลอร์ AT89C52 จะมีหน่วยความจำข้อมูลภายในสำหรับการใช้งานเพิ่มมากขึ้นกว่าเบอร์อื่นๆ เช่น AT89C51 อีก 128 ไบต์ โดยจะอยู่บริเวณช่วงแอดเดรส 80H ถึง FFH เช่นกัน ซึ่งแม้ว่าจะเป็นพื้นที่ที่มีหมายเลขแอดเดรสเดียวกับส่วนของรีจิสเตอร์หน้าที่พิเศษ แต่ในความเป็นจริงแล้วจะเป็นพื้นที่หน่วยความจำอีกบริเวณหนึ่ง ซึ่งมีการซ้อนเกย (Overlap) กันให้อยู่ในบริเวณแอดเดรสช่วงนี้ ซึ่งหากว่าผู้ใช้งานต้องการจะเก็บข้อมูลในพื้นที่บริเวณนี้ ก็จะต้องใช้การอ้างถึงหน่วยความจำแบบโดยอ้อม (Indirect Addressing) เท่านั้น

รีจิสเตอร์ทำหน้าที่พิเศษ

รีจิสเตอร์ทำหน้าที่พิเศษ เป็นรีจิสเตอร์สำหรับการควบคุมหน้าที่และการทำงานของอุปกรณ์ หรือ พอร์ตของไมโครคอนโทรลเลอร์ทั้งหมด โดยมีตำแหน่งอยู่ในบริเวณแอดเดรส 80H – FFH การใช้งานรีจิสเตอร์หน้าที่พิเศษเหล่านี้สามารถทำได้ทั้งการระบุถึงชื่อของรีจิสเตอร์ หรือตำแหน่งแอดเดรสที่เป็นของรีจิสเตอร์นั้นก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับรีจิสเตอร์หน้าที่พิเศษนี้ มีข้อสังเกตว่ารีจิสเตอร์ที่อยู่ที่ตำแหน่งแอดเดรสที่เป็นจำนวนทวีคูณของค่า 8 จะสามารถอ้างถึงในระดับบิตได้ด้วย (นั่นคือแอดเดรส 80H, 88H, 90H, 98H, A0H, A8H, B0H, B8H, D0H, E0H และ F0H)

แอกคิวมูเลเตอร์ (Accumulator) หรือ ACC

เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่ในการเก็บข้อมูลที่จะส่งให้กับหน่วยทำงานภายในซีพียู และเก็บผลลัพธ์ที่ได้จากการทำงานนั้น การทำงานจริงของรีจิสเตอร์นี้มีลักษณะเช่นเดียวกับตัวแอกคิวมูเลเตอร์ของโปรเซสเซอร์ทั่วไป การใช้งานภายในโปรแกรมจะเรียกว่า "รีจิสเตอร์ A"

รีจิสเตอร์ B

เป็นรีจิสเตอร์ที่ใช้สำหรับการทำคำสั่งการคูณและหารตัวเลข ในกรณีที่ไม่ใช้ในการคำนวณทางด้านคณิตศาสตร์ ก็สามารถนำไปใช้งานเช่นเดียวกับรีจิสเตอร์ทั่วไปได้

โปรแกรมเคาน์เตอร์ (Program Counter)

เป็นรีจิสเตอร์ที่ใช้ในการชี้ตำแหน่งแอดเดรสของหน่วยความจำโปรแกรม ซึ่งจะต้องไปทำงานในลำดับถัดไป การใช้งานภายในโปรแกรมจะเรียกว่า "รีจิสเตอร์ PC"

สแต็กพอยน์เตอร์ (Stack Pointer)

เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บตำแหน่งของตัวชี้หรือพอยน์เตอร์ (Pointer) ของบริเวณสแต็ก (Stack) สำหรับเก็บข้อมูลแอกคิวมูเลเตอร์ รีจิสเตอร์ต่างๆ รวมทั้งข้อมูลจากโปรแกรมโดยปกติแล้วเมื่อทำการเริ่มต้นระบบใหม่ภายหลังจากการเริ่มจ่ายไฟฟ้า หรือมีการรีเซต (Reset) เกิดขึ้นภายในสแต็กพอยน์เตอร์ จะมีค่า 07H ซึ่งเป็นตำแหน่งแอดเดรสภายในบริเวณเนื้อที่ 128 ไบต์แรกของหน่วยความจำข้อมูลภายใน การใช้งานภายในโปรแกรมจะเรียกว่า "รีจิสเตอร์ SP"

ตัวชี้ข้อมูลหรือดาต้าพอยน์เตอร์ (Data Pointer)

เป็นรีจิสเตอร์ขนาด 16 บิต ซึ่งเรียกว่า รีจิสเตอร์ DPTR และสามารถใช้งานแยกออกเป็นรีจิสเตอร์ขนาด 8 บิต สองตัว คือ รีจิสเตอร์ DPH และ DPL เพื่อเก็บค่าของแอดเดรสของหน่วยความจำที่จะต้องใช้งานภายในโปรแกรม หรืออาจจะเป็นแอดเดรสของอุปกรณ์ภายนอก ซึ่งกำหนดให้ติดต่อกันโดยใช้ตำแหน่งของหน่วยความจำนั้นภายในโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมสเตตัสเวิร์ด (PSW)

รีจิสเตอร์นี้ทำหน้าที่บอกถึงแฟล็กสภาวะการทำงานต่างๆ รวมทั้งบิตสำหรับการกำหนดเลือกแบงก์ของรีจิสเตอร์ที่ใช้งานด้วย

รีจิสเตอร์ที่เกี่ยวข้องกับพอร์ต (Port Register)

รีจิสเตอร์เหล่านี้จะมีความเกี่ยวข้องกับการทำงานของพอร์ตอินพุต/เอาต์พุตโดยตรง ซึ่งแต่ละตัวจะเป็นรีจิสเตอร์ขนาด 8 บิต สามารถใช้งานได้ทั้งในลักษณะของการอินพุต หรือการเอาต์พุตข้อมูลได้ การดำเนินการใดๆ ที่เกี่ยวข้องกับพอร์ต ทั้งสี่ตัวนี้ จะมีผลทำให้ข้อมูลที่ตำแหน่งของพอร์ตเหล่านี้เปลี่ยนแปลงไปเช่นกัน นอกจากนี้ พอร์ต P0 และ P2 ยังสามารถนำมาใช้ในการติดต่อกับหน่วยความจำโปรแกรมหรือหน่วยความจำข้อมูลภายนอกได้ โดยพอร์ต P2 จะเป็นค่าของแอดเดรส 8 บิต บนหน่วยความจำ ส่วนพอร์ต P0 นั้นในช่วงเริ่มแรกจะเป็นค่าของแอดเดรส 8 บิตล่างของหน่วยความจำ ช่วงเวลาต่อมาจึงจะนำพอร์ต P0 ไปใช้เป็นบัสสำหรับการรับหรือส่งข้อมูลกับหน่วยอุปกรณ์ภายนอก สำหรับพอร์ต P3 นั้นนอกเหนือจากจะใช้ในฐานะของพอร์ต อินพุต/เอาต์พุต เช่นปกติแล้ว ยังนำมาใช้ในฐานะบัสควบคุมเกี่ยวกับสัญญาณอินเตอร์รัปต์ได้อีกด้วย

รีจิสเตอร์ SBUF

เป็นบัฟเฟอร์ขนาด 8 บิต สำหรับการสื่อสารข้อมูลแบบอนุกรมทั้งการรับและส่งข้อมูล ซึ่งตามความเป็นจริงแล้วบัฟเฟอร์มีอยู่ด้วยกันสองชุด และแยกออกจากกันอย่างชัดเจน สำหรับการส่งและการรับ โดยซีพียูจะทำการจัดการเลือกบัฟเฟอร์ที่เหมาะสมให้โดยอัตโนมัติ

รีจิสเตอร์ PCON

เป็นรีจิสเตอร์ที่ใช้ในการควบคุมหน้าที่การทำงานในสามลักษณะได้แก่ การควบคุมการทำงานของโปรเซสเซอร์ (บิต IDL และ PD) การกำหนดอัตราการทวีคูณของอัตราเร็วในการสื่อสารข้อมูลอนุกรม (บิต SMOD) และแฟล็กสภาวะสำหรับการใช้งานทั่วไป (บิต GR0 และ GR1)

รีจิสเตอร์ IP, IE, TMOD, SCON

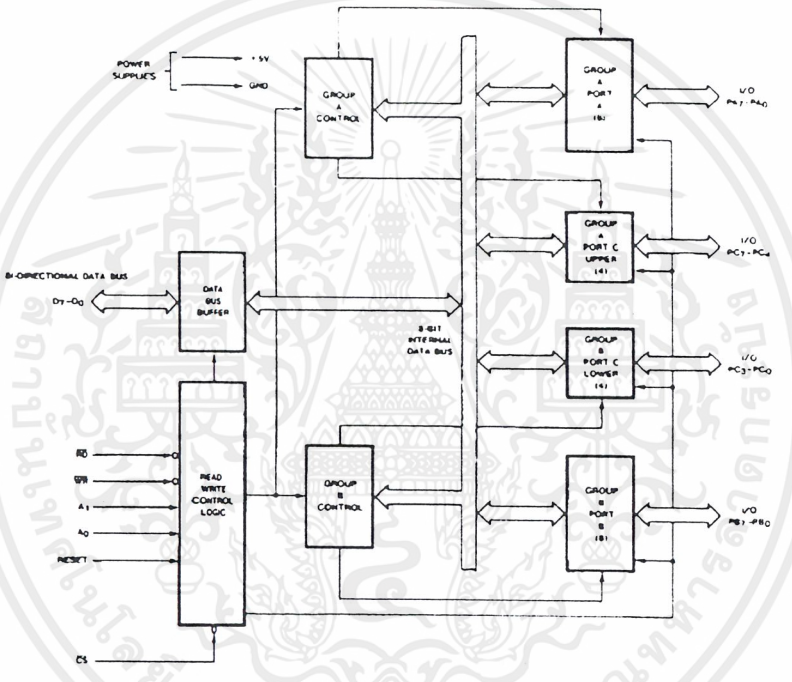
เป็นกลุ่มของรีจิสเตอร์ที่ทำหน้าที่กำหนดการควบคุม และการทำงานของการอินเตอร์รัปต์ต่าง ๆ ของ AT89C52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 การใช้งาน 8255 กับ AT89C52

2.7.1ลักษณะพื้นฐานของ 8255

ไอซีเบอร์ 8255 ออกแบบมาเพื่อทำหน้าที่เป็นพอร์ต สำหรับการรับ/ส่งข้อมูลแบบขนาน ระหว่างอุปกรณ์ภายนอก กับไมโครคอนโทรลเลอร์ การใช้งานของ 8255 นั้นสามารถเปลี่ยนแปลงลักษณะการทำงานของพอร์ต ให้เป็นการเอาต์พุตหรืออินพุต ได้สะดวก เพียงการส่งข้อมูลควบคุมจากไมโครคอนโทรลเลอร์ก่อนที่จะเริ่มต้นใช้งานเท่านั้น



รูปที่ 2.18แสดงแผนภาพแบบบล็อกภายในและขาสัญญาณของไอซีเบอร์ 8255

จากแผนภาพ จะเห็นว่า 8255 ประกอบด้วยบล็อกของหน่วยการทำงานหลายส่วน อยู่ภายในบล็อกทางด้านขวามือ จำนวน 4 บล็อก เป็นส่วนที่เชื่อมต่อกับอุปกรณ์ภายนอกโดยตรงผ่านทางเส้นสัญญาณที่ระบุชื่อว่า PA0 - PA7, PB0 - PB7 และPC0 - PC7 กลุ่มของสัญญาณเหล่านี้จำแนกออกเป็น 3 กลุ่มคือ พอร์ต A (PA) พอร์ต B (PB) และพอร์ต C (PC) สำหรับบล็อกถัดเข้ามาบริเวณส่วนกลางที่มีชื่อว่า GROUP A CONTROL และGROUP B CONTROL ทำหน้าที่กำหนดการทำงานของพอร์ตทั้งสาม บล็อกทั้งสองนี้เชื่อมต่อกับบล็อกอื่นๆ ผ่านทางบัสข้อมูลภายใน 8255 เอง สำหรับบล็อกการทำงานทางด้านซ้ายมือ ที่มีชื่อว่า Data Bus Buffer และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Read/Write Control Logic ทำหน้าที่เชื่อมต่อระหว่างระบบบัสของไมโครคอนโทรลเลอร์ กับ8255 เพื่อรับหรือส่งข้อมูลระหว่างกันตามระดับลอจิกของขาสัญญาณ $\overline{RD}$ และ $\overline{WR}$ ตามลำดับ

2.7.2 การจำแนกกลุ่มพอร์ต 8255

ในบรรดาพอร์ตทั้งสามของ 8255 คือ พอร์ต A พอร์ต B และพอร์ต C โดยพื้นฐานนั้น ล้วนเป็นพอร์ตแบบขนานที่ประกอบด้วยสัญญาณ 8 เส้น ซึ่งแต่ละเส้นจะแทนบิตของข้อมูลพอร์ต ซึ่งอาจจะกล่าวในอีกลักษณะว่าเป็นพอร์ตแบบ 8 บิต อย่างไรก็ตาม 8255 ได้จัดกลุ่มของพอร์ตเหล่านี้ออกเป็น 2 กลุ่ม คือ Group A และ Group B เพื่อประโยชน์ในการกำหนดรูปแบบการทำงานของพอร์ต ดังตารางที่ 2.5

ชื่อกลุ่ม	ลักษณะ
Group A	พอร์ต A จำนวน 8 บิต (ทุกบิตของพอร์ต) พอร์ต C จำนวน 4 บิต (เฉพาะ 4 บิตบนของพอร์ต)
Group B	พอร์ต B จำนวน 8 บิต (ทุกบิตของพอร์ต) พอร์ต C จำนวน 4 บิต (เฉพาะ 4 บิตล่างของพอร์ต)

ตารางที่ 2.5

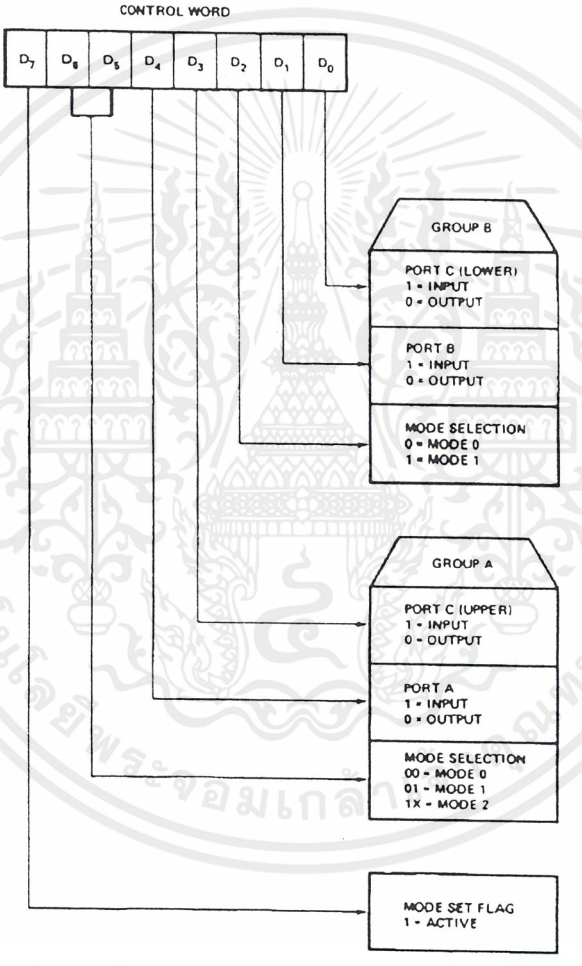
สัญญาณ	ความหมาย
D0 - D7	กลุ่มของเส้นสัญญาณข้อมูลของ 8255 เมื่อมีการเขียน หรืออ่าน
$\overline{CS}$	สัญญาณเลือกอุปกรณ์ เมื่อขาสัญญาณนี้เป็นระดับลอจิกต่ำ ซีพียูก็สามารถ เขียน หรืออ่านข้อมูลจาก 8255 ได้
$\overline{RD}$	สัญญาณบอกสถานะต้องการอ่านข้อมูลจากรีจิสเตอร์ของ 8255
$\overline{WR}$	สัญญาณบอกสถานะต้องการเขียนข้อมูลให้กับรีจิสเตอร์ของ 8255
A0-A1	Lสัญญาณระบุตำแหน่งรีจิสเตอร์ภายใน 8255 ที่ต้องการ
RESET	สัญญาณการรีเซตวงจรการทำงานภายใน 8255 เพื่อเริ่มต้นใหม่
PA0-PA7	กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ต A ของ 8255
PB0-PB7	กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ต B ของ 8255
PC0-PC7	กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับพอร์ต C ของ 8255

ตารางที่ 2.6 แสดงหน้าที่การทำงานของขาสัญญาณไอซี 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7.3 รูปแบบคำสั่งเพื่อกำหนดการทำงานของ 8255

การกำหนดให้พอร์ตทั้งสามของ 8255 ทำงานในลักษณะต่างๆ กัน หรือที่เรียกว่า โหมดการทำงาน (MODE) จะเริ่มด้วยการส่งค่าข้อมูลไบต์หนึ่งให้กับรีจิสเตอร์ควบคุมการทำงานภายใน 8255 ข้อมูลนี้จะเรียกว่า ไบต์ข้อมูลควบคุม (Control Word) โดยแต่ละบิตของข้อมูลนี้จะมีความหมายที่ระบุถึงความต้องการต่าง ๆ ไปดังแสดงในรูปที่ 2.19



รูปที่ 2.19 ความหมายของบิตภายในไบต์ข้อมูลควบคุมสำหรับ 8255

การส่งข้อมูลไบต์นี้จะต้องเริ่มต้นเป็นลำดับแรกก่อนที่จะได้มีการดำเนินการใดกับ 8255 ทั้งสิ้น ตามความหมายของบิตภายในตารางของรูปข้างต้น จะเห็นว่าการเลือกให้พอร์ตใดทำหน้าที่เป็นพอร์ตอินพุต ก็เพียงแต่กำหนดค่าข้อมูล 1 ให้กับบิตที่เกี่ยวข้องกับพอร์ตนั้น หรือกรณีตรงข้ามสำหรับการเอาต์พุตก็เพียงการกำหนดค่าข้อมูล 0 เท่านั้น อย่างไรก็ตาม การกำหนดให้ไบต์ข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมนี้มีผลอย่างถูกต้อง ก็จะต้องทำการกำหนดให้บิต D7 มีค่าเป็น 1 เสมอ สำหรับบิตที่บอกถึงโหมดการทำงาน (บิต D6-D5 และ D2) นั้นจะได้กล่าวรายละเอียดในหัวข้อต่อไป

2.7.4 การทำงานโหมด 0 ของ 8255

เมื่อ 8255 ได้รับการกำหนดให้ทำงานในโหมดนี้ จะทำให้พอร์ตต่างๆ มีหน้าที่เป็นพอร์ตอินพุต หรือเอาต์พุตได้เพียงลักษณะเดียวเท่านั้น การเริ่มต้นจะทำโดยการส่งไบต์ข้อมูลควบคุมให้กับรีจิสเตอร์ควบคุม ต่อไปจะได้แสดงให้เห็นถึงรูปแบบการกำหนดบิต เมื่อต้องการให้พอร์ต A,B และ C บนทำหน้าที่เป็นพอร์ตเอาต์พุต ส่วนพอร์ต C ล่างเป็นอินพุต แสดงที่ตารางที่ 2.7

ตำแหน่งบิต	ค่าข้อมูล	ความหมาย
D7	1	ระบุให้ทราบว่าเป็นไบต์ข้อมูลควบคุม
D6 และ D5	00	กำหนดโหมดการทำงานให้กับพอร์ต A เป็นโหมด 0
D4	0	ระบุว่าพอร์ต A เป็นการเอาต์พุตข้อมูล
D3	0	กำหนดให้เส้นสัญญาณสี่บิตบนของพอร์ต C เป็นการเอาต์พุตข้อมูล
D0	0	กำหนดโหมดการทำงานให้กับพอร์ต B เป็นโหมด 0
D1	0	ระบุว่าพอร์ต B เป็นการเอาต์พุตข้อมูล
D2	1	กำหนดให้เส้นสัญญาณสี่บิตล่างของพอร์ต C เป็นการอินพุตข้อมูล

ตารางที่ 2.7

ภายหลังจาก 8255 ได้รับการโปรแกรมจากค่าของไบต์ข้อมูลควบคุมนี้แล้ว ก็สามารถที่จะใช้งานพอร์ตทั้งหมดในฐานะพอร์ตเอาต์พุต/อินพุตเพื่อส่งข้อมูลออก หรือรับข้อมูลเข้าได้ตามต้องการ

ค่าไบต์ข้อมูลควบคุมนี้ จะต้องส่ง (หรือ เขียน) ให้กับรีจิสเตอร์ควบคุม ซึ่งหากกำหนดให้แอดเดรสของพอร์ตควบคุมเป็น 0403H สามารถแสดงคำสั่งได้ดังนี้

```
MOV   DPTR,#0403H        ;เซตค่าแอดเดรสรีจิสเตอร์ควบคุม
MOV   @DPTR,#80H         ;ส่งค่าไบต์ควบคุมออกไปยัง 8255
```

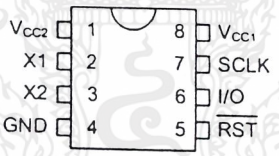
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 ไอซีสร้างฐานเวลาจริง (Real Time Clock : RTC)

ไอซีที่นำมาใช้เป็นของ ดัลลัสเซมิคอมดักเตอร์ (Dallas Semiconduct) เบอร์ DS1302 มีหน้าที่สร้างฐานเวลาจริงให้แก่ไมโครคอนโทรลเลอร์ โดยไอซีเบอร์นี้ จะให้ข้อมูลเกี่ยวกับเวลาทั้งหมด ไม่ว่าจะเป็นค่าเวลาที่ละเอียดถึงหลักวินาที, นาที, ชั่วโมง, วันที่ (Date), วันในสัปดาห์ (Day), เดือน และปี โดยสามารถปรับวันเดือนปีให้ตรงตามปฏิทินได้อย่างถูกต้อง รวมถึงการกำหนดวันในปีอธิกสุรธินด้วย คุณสมบัติทางเทคนิคที่สำคัญมีดังนี้

- เป็นไอซีสร้างฐานเวลาจริงให้ข้อมูลตั้งแต่วินาทีจนถึงปี รวมถึงการปรับวันในปีอธิกสุรทินด้วยสามารถให้ข้อมูลได้อย่างเที่ยงตรงถึงปีคริสตศักราช 2100
- มีหน่วยความจำแรม 31 ไบต์ สำหรับใช้เก็บข้อมูลทั่วไปได้
- สื่อสารอินพุต/เอาต์พุตแบบอนุกรม
- มีทั้งโหมด 24 ชั่วโมง และ 12 ชั่วโมงให้เลือก
- ใช้การสื่อสารแบบ 3 สาย

2.8.1 รายละเอียดขาต่อใช้งานของ DS1302



รูปที่ 2.20 แสดงการจัดขาของ DS1302 ซึ่งแต่ละขามีหน้าที่และการใช้งานดังนี้

X1,X2	ใช้ต่อกับคริสตอลความถี่มาตรฐาน 32.768 kHz เพื่อใช้เป็นฐานเวลาในการสร้างค่าเวลาจริง ในการใช้งานต้องต่อคริสตอลเข้ากับขาทั้งสองนี้
GND	ขากราวด์
$\overline{RST}$	ขารีเซต
I/O	ขาข้อมูลแบบ 2 ทิศทาง อินพุต/เอาท์พุต
SCLK	ขาสัญญาณนาฬิกาอนุกรม (Serial Clock) ใช้สำหรับเป็นสัญญาณอ้างอิงในการส่งข้อมูลแบบอนุกรม
V_{cc1}	ขาแหล่งจ่ายไฟสำรอง
V_{cc2}	ขาแหล่งจ่ายไฟหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยการต่อแหล่งจ่ายไฟจะต้องต่อทั้งที่ V_{cc1} และ V_{cc2} โดยถ้า V_{cc2} มีแรงดันสูงกว่า V_{cc1} อยู่ 0.2 โวลต์ V_{cc2} จะจ่ายพลังงานให้แก่ DS1302 แต่ถ้า V_{cc2} มีโวลต์มากกว่า V_{cc1} V_{cc1} ก็จะจ่ายพลังงานให้แก่DS1302

2.8.2 รูปแบบคำสั่งการทำงานของ DS1302

สามารถกำหนดการทำงานของ DS1302 ได้ด้วย ไบต์คำสั่งซึ่งแสดงในรูปที่ 2.21 ซึ่งแต่ละข้อมูลที่จะถูกส่งจะต้องส่งไบต์คำสั่งออกไปก่อน เพื่อกำหนดการทำงาน

บิตที่ 7	บิตที่ 6	บิตที่ 5	บิตที่ 4	บิตที่ 3	บิตที่ 2	บิตที่ 1	บิตที่ 0
1	RAM/$\overline{CK}$	A4	A3	A2	A1	A0	RD/$\overline{W}$

รูปที่ 2.21

ในบิตที่ 7 ของไบต์คำสั่งจะต้องเป็นลอจิก 1 เนื่องจากถ้าเป็นลอจิก 0 การเขียนข้อมูลไปยัง DS1302 จะไม่ทำงาน ในบิตที่ 6 ถ้าเป็นลอจิก 1 จะเป็นการเลือกการติดต่อกับหน่วยความจำข้อมูลภายใน ถ้าเป็นลอจิก 0 จะเป็นการเลือกการใช้งานติดต่อข้อมูลของเวลา บิตที่1 ถึง 5 ไว้ระบุตำแหน่งรีจิสเตอร์ที่จะเป็นอินพุตหรือเอาท์พุต ส่วนในบิตสุดท้าย เป็นการระบุว่า ถ้าเป็นลอจิก 1 จะเป็นการอ่านข้อมูลจาก DS1302 (เอาท์พุต) แต่ถ้าเป็นลอจิก 0 จะเป็นการเขียนข้อมูลไปยัง DS1302 (อินพุต) โดยจะเริ่มต้นไบต์คำสั่งนี้ด้วยการเลือกเป็นการเขียนข้อมูลไปยัง DS1302 เสมอ

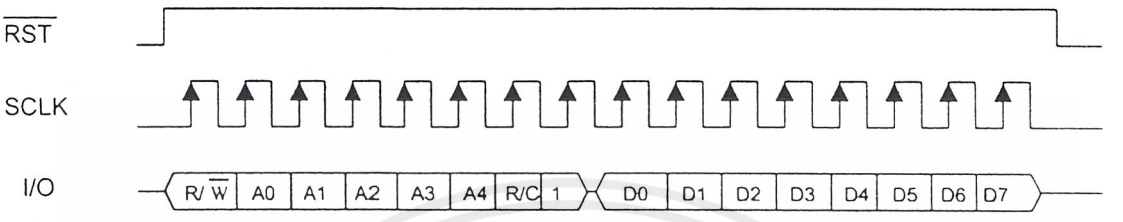
2.8.3 การทำงานของ DS1302

การทำงานมีด้วยกันอยู่ 2 โหมด คือโหมดเขียนข้อมูล และโหมดอ่านข้อมูล ในการใช้งาน DS1302 ตามปกติจะใช้งานเฉพาะโหมดอ่านข้อมูล เนื่องจากไมโครคอนโทรลเลอร์จะติดต่อกับ DS1302 เพื่ออ่านข้อมูลของเวลาไปใช้งาน โหมดการเขียนข้อมูลจะถูกใช้งานก็ต่อเมื่อต้องการตั้งค่าเวลาใหม่ หรือต้องการเขียนข้อมูลลงในหน่วยความจำใช้งานทั่วไป อย่างไรก็ตามเมื่อเริ่มต้นติดต่อกับ DS1302 จำเห็นต้องเข้าสู่โหมดการเขียนข้อมูลก่อนเพื่อกำหนดแอดเดรสที่ต้องการอ่านข้อมูล จากนั้นจึงเปลี่ยนโหมดการทำงานมาเป็นโหมดการอ่านข้อมูลต่อไป

โหมดการเขียนข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

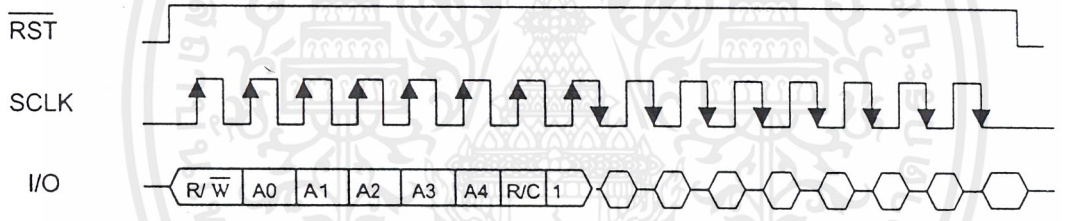
เริ่มต้นด้วยการรับไบต์คำสั่งการเขียนข้อมูลก่อน 8 บิต ตามสัญญาณ SCLK 8 ไซเคิล จากนั้นไบต์ข้อมูลจะถูกรับเข้ามาทีละบิตทุกขอบขาขึ้นของสัญญาณ SCLK 8 ไซเคิลถัดไป แสดงดังรูปที่ 2.22



รูปที่ 2.22

โหมดการอ่านข้อมูล

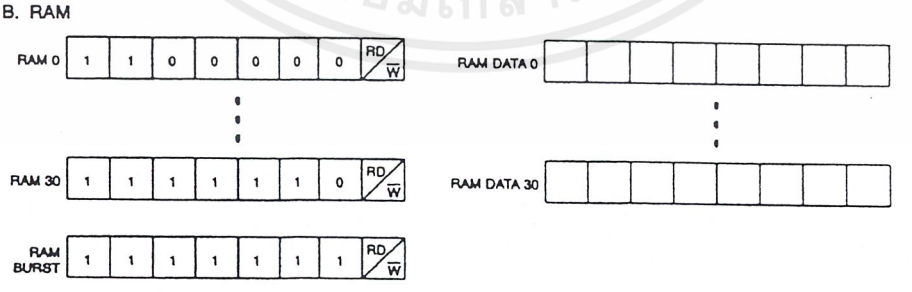
เริ่มต้นด้วยการรับไบต์คำสั่งการอ่านข้อมูลก่อน 8 บิต ตามสัญญาณ SCLK 8 ไซเคิล จากนั้นไบต์ข้อมูลจะถูกส่งออกไปละบิตทุกขอบขาขลงของสัญญาณ SCLK 8 ไซเคิลถัดไป แสดงดังรูปที่ 2.23



รูปที่ 2.23

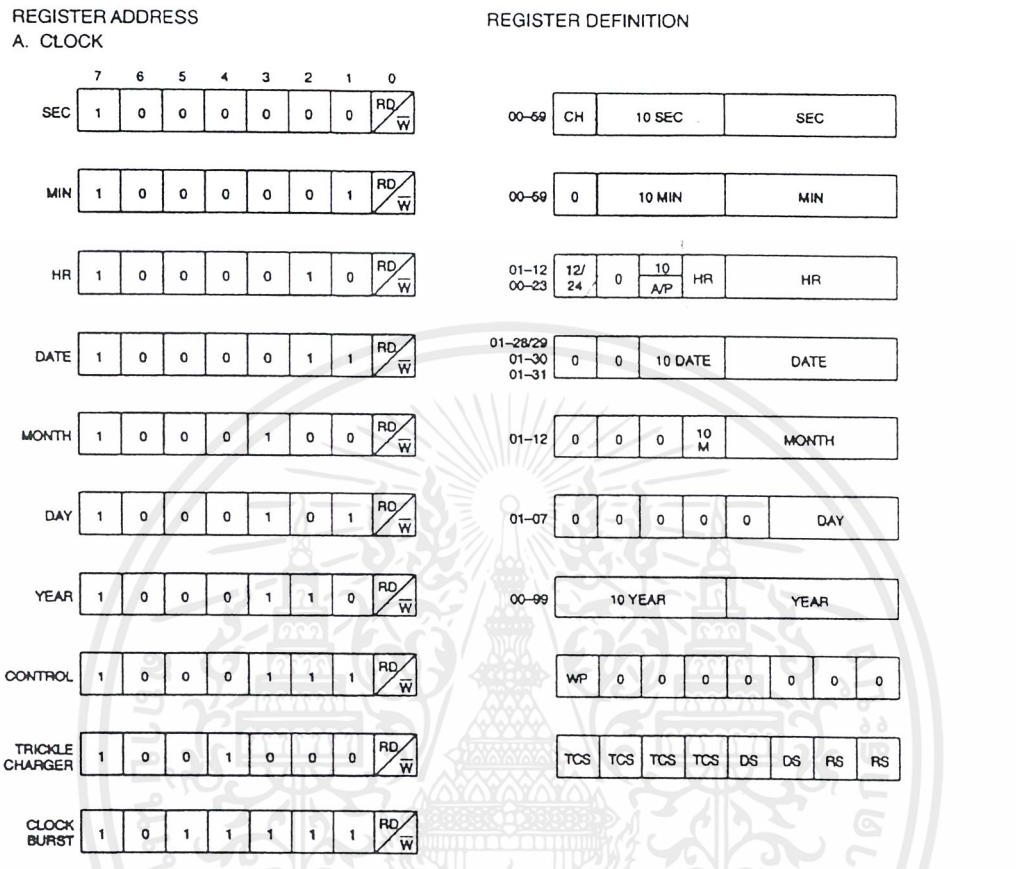
2.8.4 การจัดสรรหน่วยความจำใน DS1302

การจัดสรรหน่วยความจำใน DS1302แสดงไว้ดังรูปที่ 2.24 (ก) และ 2.24 (ข)



รูปที่ 2.24 (ก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 (ข)

2.9 ส่วนแสดงผลแบบผลึกเหลว (LCD Module)

ในส่วนแสดงผลแบบผลึกเหลว จะมีส่วนประกอบหลักๆ 3 ส่วนดังนี้

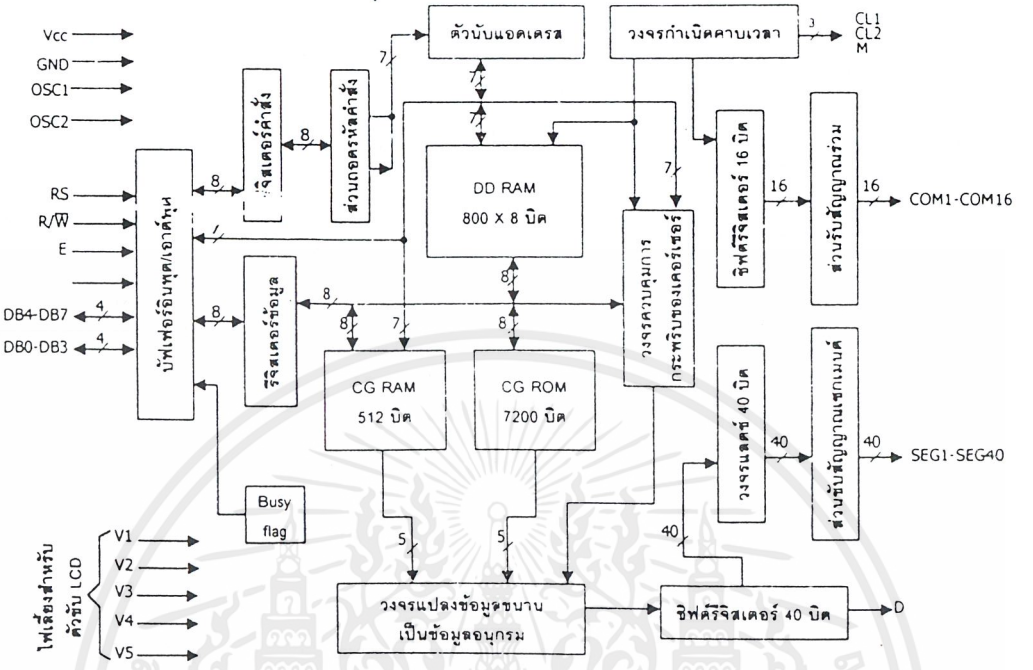
ตัวแสดงผล (Display) ภายในเป็นผลึกเหลวที่สามารถแสดงผลให้เห็นโดยอาศัยแสงจากภายนอก ดังนั้นจึงต้องมีมุมในการมองข้อมูลที่แสดงผลบนส่วนแสดงผลแบบผลึกเหลว

ตัวควบคุม (Controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของส่วนแสดงผลแบบผลึกเหลว เช่นลบจอภาพ แสดงตัวอักษร หรือเลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมนั้นใช้ชิพควบคุมโดยเฉพาะ

ตัวขับ (Driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลแสดงข้อมูลตามที่กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.9.1 โครงสร้างภายในของตัวควบคุมส่วนแสดงผลแบบผลึกเหลว



รูปที่ 2.25 เป็นบล็อกไดอะแกรมภายในของชิพควบคุมส่วนแสดงผลแบบผลึกเหลว

บัฟเฟอร์อินพุตเอาท์พุต เป็นส่วนที่ใช้ในการติดต่อรับส่งข้อมูลกับอุปกรณ์ภายนอก เพื่อที่จะถ่ายทอดข้อมูลเข้าออกภายในตัวควบคุม

รีจิสเตอร์คำสั่ง (Instruction Register : IR) เป็นรีจิสเตอร์ใช้รับข้อมูลคำสั่งจากอุปกรณ์ภายนอก เพื่อนำไปควบคุมการแสดงผล

รีจิสเตอร์ข้อมูล (Data Register : DR) เป็นรีจิสเตอร์ใช้รับข้อมูลจากอุปกรณ์ภายนอก เพื่อถ่ายทอดต่อไปยังหน่วยความจำที่ทำหน้าที่เก็บข้อมูลแสดงผล หรือนำข้อมูลไปสร้างตัวอักษรเพิ่มเติมในแรมเก็บตัวอักษร

แรมเก็บข้อมูลแสดงผล (Display Data RAM : DDRAM) เป็นหน่วยความจำแรมทำหน้าที่เก็บข้อมูลที่มาจากรีจิสเตอร์ DR ตัวควบคุมจะนำข้อมูลใน DDRAM นี้ไปเปิดตาราง (Look-Up-Lable) ของตัวอักษรที่เก็บไว้ในหน่วยความจำรอมและเก็บตัวอักษร เพื่อนำไปแสดงที่ตัวแสดงผล

รอมเก็บตัวอักษร (Character Generator ROM : CGROM) เป็นหน่วยความจำรอมที่ใช้เก็บข้อมูลตัวอักษรหรือสัญลักษณ์ที่สามารถอ่านออกไปแสดงที่ตัวแสดงผลได้ มีขนาด 7,200 บิต โดยจะถูกอ่านด้วยค่าของข้อมูลใน DDRAM

แรมเก็บตัวอักษร (Character Generator RAM : CGRAM) เป็นหน่วยความจำแรมที่ใช้เก็บอักษรที่มีการสร้างเพิ่มเติมขึ้นใหม่ ในกรณีที่ตัวอักษรใน CGROM ไม่เพียงพอ มีขนาด 512 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนและอ่านค่าไปใช้นั้นทำได้เช่นเดียวกับ CGROM คือ เขียนข้อมูลลงใน DDRAM แล้วตัวควบคุมจะมาอ่านค่าจาก CGRAM เอง

แฟลก BUSY เป็นส่วนที่ทำหน้าที่แจ้งสถานะการทำงานของตัวควบคุมให้อุปกรณ์ภายนอกทราบว่า ตัวควบคุมพร้อมที่จะรับข้อมูลหรือคำสั่งหรือไม่ ดังนั้นก่อนการส่งข้อมูลหรือคำสั่งมายังตัวควบคุมต้องตรวจสอบสถานะของแฟลก BUSY นี้เสียก่อน

ส่วนแสดงผลแบบผลึกเหลวที่นำมาใช้เป็นขนาด 16 ตัวอักษร 2 บรรทัด มีขาต่อใช้งาน 14 ขา ซึ่งมีรายละเอียดดังนี้

V_{ss} (ขา 1) ต่อกราวด์

V_{dd} (ขา 2) ต่อไฟเลี้ยง + 5 โวลต์

V_o (ขา 3) เป็นขาอินพุตรับแรงดันเพื่อหรับความเข้มของการแสดงผล

RS (ขา 4) เป็นขาอินพุตใช้ในการแยกชนิดข้อมูลที่ทำการประมวลผลในขณะนั้นว่าเป็นคำสั่งสำหรับรีจิสเตอร์ IR หรือเป็นข้อมูลสำหรับรีจิสเตอร์ DR โดยถ้าขานี้เป็น “0” ข้อมูลที่ส่งมาจะเป็นคำสั่ง แต่ถ้าขานี้เป็น “1” ข้อมูลที่ส่งมาจะเป็นข้อมูลสำหรับการแสดงผล

$R/\overline{W}$ (ขา 5) เป็นขาที่ใช้เลือกการอ่านหรือเขียนข้อมูลกับ LCD ถ้าเป็น “0” เป็นการกำหนดให้เขียนข้อมูล แต่ถ้าเป็น “1” จะเป็นการอ่านข้อมูล

E (ขา 6) เป็นขาอีนาเบิล LCD ทำงาน

D0-D7 (ขา 7-14) เป็นขาที่ใช้เป็นทางผ่านของข้อมูลระหว่าง ส่วนแสดงผลแบบผลึกเหลวกับอุปกรณ์ภายนอกขนาด 8 บิต

2.9.2 คำสั่งควบคุมส่วนแสดงผลแบบผลึกเหลว

ในการเขียนคำสั่งลงในตัวควบคุม ต้องกำหนดให้ขา RS และ ขา $R/\overline{W}$ เป็น “0” แล้วเขียนคำสั่งตามไป คำสั่งควบคุมส่วนแสดงผลแบบผลึกเหลว ที่สำคัญมี 9 คำสั่งดังนี้

1. **คำสั่งเคลียร์ตัวแสดงผล (Clear Display)**
   มีข้อมูลคำสั่งเป็น 01H เป็นคำสั่งที่ใช้เขียนข้อมูลช่องว่างเข้าไปใน DDRAM ทั้งหมด เมื่อตัวควบคุมเอ็กซิคิวต์คำสั่งนี้ จะทำการกำหนดแอดเดรสของ DDRAM เป็น 0 เคอร์เซอร์จะกลับไปอยู่ที่ตำแหน่งซ้ายมือสุดของจอแสดงผล แล้วเซตบิต I/D ให้เป็น “1”

2. **คำสั่ง Return Home**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องกำหนดให้บิต 1 ของข้อมูลเป็น “1” เป็นคำสั่งให้เคอร์เซอร์เคลื่อนที่กลับไปยังตำแหน่งซ้ายสุดของจอแสดงผล แต่ข้อมูลบนจอแสดงผลไม่เปลี่ยนแปลง นั่นคือ ข้อมูลคำสั่งของคำสั่งนี้จะเป็น 02H หรือ 03H ก็ได้

3. **คำสั่งเลือกโหมดการป้อนข้อมูล** (Entry Mode Set)
มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิตที่7	บิตที่6	บิตที่5	บิตที่4	บิตที่3	บิตที่2	บิตที่1	บิตที่0
0	0	0	0	0	1	I/D	S

บิต S เป็นบิตที่ใช้ในการกำหนดลักษณะของการแสดงผล เมื่อมีการป้อนข้อมูล ถ้าหากบิต S เป็น”1” เมื่อเกิดข้อมูลใหม่บนจอแสดงผล ตัวเคอร์เซอร์จะอยู่กับที่ แต่ตัวอักษรข้อมูลเดิมจะถูกดันไปทางซ้าย แต่หากว่าเป็น ”0” เมื่อเกิดข้อมูลใหม่ตัวเคอร์เซอร์จะเลื่อนไปทางขวามือ

บิต I/D เป็นบิตที่ใช้ในการกำหนดว่า เมื่อเขียนหรืออ่านข้อมูลแล้ว ทำให้แอดเดรสของ DDRAM เพิ่มขึ้นหรือลดลงหนึ่งแอดเดรส โดยถ้าบิตนี้เป็น “1” แอดเดรสของ DDRAM จะเพิ่มขึ้น แต่ถ้าเป็น “0” แอดเดรสจะลดลง

4. **คำสั่งควบคุมการแสดงผล**
มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิตที่7	บิตที่6	บิตที่5	บิตที่4	บิตที่3	บิตที่2	บิตที่1	บิตที่0
0	0	0	0	1	D	C	B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต D ใช้ควบคุมการเปิดปิดจอแสดงผล ถ้าบิตนี้เป็น “1” จะเป็นการเปิดจอแสดงผล ถ้าเป็น “0” จะเป็นการปิดจอแสดงผล

บิต C ใช้ควบคุมการแสดงตัวเคอร์เซอร์บนจอแสดงผล ถ้าต้องการให้มีเคอร์เซอร์แสดงผลบนจอแสดงผล ต้องกำหนดให้บิตนี้เป็น “1” ถ้ากำหนดให้เป็น “0” จะเป็นการปิดเคอร์เซอร์

บิต B ใช้ควบคุมการกะพริบของเคอร์เซอร์ ถ้าบิตนี้เป็น “1” เคอร์เซอร์จะกะพริบ

5. **คำสั่งควบคุมการเลื่อนเคอร์เซอร์และข้อมูลตัวอักษร**

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิตที่7	บิตที่6	บิตที่5	บิตที่4	บิตที่3	บิตที่2	บิตที่1	บิตที่0
0	0	0	1	S/C	R/L	*	*

การควบคุมการเลื่อนเคอร์เซอร์และตัวอักษรบนจอแสดงผลขึ้นอยู่กับการกำหนดบิต S/C และ R/L ซึ่งสามารถสรุปได้ตามตารางที่ 2.8 ดังนี้

S/C	R/L	ลักษณะการเลื่อน	ข้อมูลคำสั่ง
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย	10H-13H
0	1	เลื่อนเคอร์เซอร์ไปทางขวา	14H-17H
1	0	เลื่อนตัวอักษรใหม่ไปทางซ้าย	18H-1BH
1	1	เลื่อนตัวอักษรใหม่ไปทางขวา	1CH-1FH

ตารางที่ 2.8

6. **คำสั่งกำหนดฟังก์ชันการทำงาน**

มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิตที่7	บิตที่6	บิตที่5	บิตที่4	บิตที่3	บิตที่2	บิตที่1	บิตที่0
0	0	1	D/L	N	F	*	*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต DL ใช้กำหนดจำนวนบิตที่ใช้ติดต่อส่งผ่านข้อมูล ถ้าบิตเป็น "0" จะเป็นการติดต่อแบบ 4 บิต แต่ถ้าเป็น "1" จะเป็นแบบ 8 บิต

บิต N ใช้กำหนดจำนวนบรรทัดของการแสดงผล ถ้าเป็น "0" จะแสดงผล 1 บรรทัด ถ้าเป็น "1" จะแสดง 2 บรรทัด ในกรณีที่จอแสดงผลสามารถแสดงได้มากกว่า 2 บรรทัด และต้องการให้แสดงผลมากกว่า 2 บรรทัด ก็กำหนดบิต N นี้ให้เป็น "1"

บิต F ใช้เลือกความละเอียดของตัวอักษรในการแสดงผล ถ้าบิตนี้เป็น "0" จะเป็นการแสดงผลแบบ 5X7 จุด และถ้าเป็น "1" จะแสดงผลเป็นแบบ 5x10 จุด

7. **คำสั่งเลือกแอดเดรสของ CCRAM**
   เมื่อต้องการกำหนดแอดเดรสCGRAM ต้องกำหนดให้บิต 7 เป็น "0" บิต 6 เป็น "1" ส่วนอีก 6 บิตที่เหลือแทนด้วยค่าแอดเดรส CGRAM จะต้องทำการกำหนดแอดเดรสด้วยคำสั่งนี้ ก่อนที่จะอ่านหรือเขียนข้อมูลให้CGRAM อยู่ระหว่าง 00H-3FH

8. **คำสั่งเลือกแอดเดรสของ DDRAM**
   ใช้ในการเลือกแอดเดรส DDRAM ก่อนที่จะทำการอ่านหรือเขียนข้อมูล โดยบิต 7 ต้องเป็น "1" และข้อมูลอีก 7 บิต ที่เหลือจะเป็นค่าแอดเดรสของ DDRAM ซึ่งแอดเดรสของ DDRAM จะอยู่ระหว่าง 8CH-0FFH ทั้งนี้จำนวนแอดเดรสยังขึ้นกับการกำหนดสถานะที่บิต N ด้วย หากบิต N เป็น "0" แอดเดรสของDDRAM จะอยู่ระหว่าง 80H-0CFH และถ้าบิต N เป็น "1" แอดเดรสของ DDRAM จะมี 2 ช่วงคือ 8CH-87H และ 0C0H-070H

9. **คำสั่งอ่านแฟลก BUSY และแอดเดรส**
   มีรายละเอียดของรูปแบบข้อมูลคำสั่งดังนี้

บิตที่7	บิตที่6	บิตที่5	บิตที่4	บิตที่3	บิตที่2	บิตที่1	บิตที่0
BF	A	A	A	A	A	A	A

เป็นคำสั่งที่ใช้อ่านแฟลก BUSY (BF) โดยแฟลกนี้จะเป็นตัวบอกสถานะของตัวควบคุมส่วนแสดงผลแบบผลึกเหลว ว่าพร้อมจะรับข้อมูลอยู่หรือไม่ ถ้าหากบิต BF เป็น "0" แสดงว่าตัวควบคุมส่วนแสดงผลแบบผลึกเหลวพร้อมรับข้อมูลหรือคำสั่ง แต่หาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น “1” แสดงว่าขณะนี้ตัวควบคุมส่วนแสดงผลแบบผลึกเหลวยังอยู่ในกระบวนการทำงานภายในหรือกำลังประมวลผลข้อมูลอยู่ ยังไม่พร้อมที่จะรับข้อมูลหรือคำสั่ง เมื่อต้องการอ่านแฟลกต้องกำหนดให้ขา R/$\overline{W}$ เป็น “1” ด้วย แต่สัญญาณที่ RS ยังต้องเป็น “0” อยู่เพราะข้อมูลนี้เป็นข้อมูลคำสั่ง

2.9.3 การเขียนคำสั่งและข้อมูลให้แก่ส่วนแสดงผลแบบผลึกเหลว

ในการเขียนข้อมูลเพื่อควบคุมให้ส่วนแสดงผลแบบผลึกเหลวแสดงผลตามที่ต้องการ ต้องส่งคำสั่ง แล้วกำหนดโหมดการทำงานให้แก่ส่วนแสดงผลแบบผลึกเหลวก่อน จากนั้นจึงค่อยส่งข้อมูลที่ต้องการแสดงผลเนื่องจากบัสข้อมูลของส่วนแสดงผลแบบผลึกเหลวมี 8 เส้นคือ D0-D7 และใช้เป็นทางผ่านของทั้งคำสั่งและข้อมูล ดังนั้นในการส่งคำสั่งและข้อมูลจึงต้องอาศัยการกำหนดสัญญาณลอจิกที่ขา RS ถ้าหากที่ขา RS ได้ลอจิก “0” หมายความว่า ข้อมูลที่ป้อนให้แก่ส่วนแสดงผลแบบผลึกเหลวขณะนั้นเป็นคำสั่ง ในทางตรงกันข้าม หากขา RS ได้รับลอจิก “1” ข้อมูลที่ป้อนให้ขณะนั้นเป็นข้อมูลที่ใช้ในการแสดงผล

เมื่อต้องการเขียนหรืออ่านข้อมูลใน CGRAM และ DDRAM เริ่มต้น ต้องกำหนดแอดเดรสที่ต้องการอ่านหรือเขียนก่อน โดยใช้คำสั่งเลือกแอดเดรส จากนั้นกำหนดให้ขา RS เป็น “1” เพื่อแจ้งให้ตัวควบคุมภายในส่วนแสดงผลแบบผลึกเหลวทราบว่าข้อมูลที่ปรากฏต่อไปนี้เป็นข้อมูลปกติไม่ใช่คำสั่ง

ในกรณีที่ต้องการอ่านข้อมูลต้องกำหนดให้ขา R/$\overline{W}$ เป็น “1” ข้อมูลขนาด 8 บิต (หรือ 4 บิต) ก็จะปรากฏบนบัสข้อมูล โดยข้อมูลที่อ่านออกมาได้จะเป็นข้อมูลจากแอดเดรสของ CGRAM หรือ DDRAM ตามที่ต้องการ

ในกรณีที่ต้องการเขียนข้อมูล เมื่อกำหนดแอดเดรสและป้อนลอจิก “1” ให้ขา RS แล้ว ต้องกำหนดให้ขา R/$\overline{W}$ เป็น “0” ข้อมูลที่อยู่บนบัสข้อมูลจะถูกเขียนลงในรีจิสเตอร์ DR จากนั้นจึงถ่ายทอดลงใน DDRAM ต่อไป

2.9.4 จังหวะการทำงานของส่วนแสดงผลแบบผลึกเหลว

ในการติดต่อกับส่วนแสดงผลแบบผลึกเหลว จะต้องมีการหน่วงเวลาหลังจากที่ทำการส่งรหัสคำสั่งหรือข้อมูลเนื่องจากต้องรอให้คอนโทรลเลอร์ภายในส่วนแสดงผลแบบผลึกเหลว แปลความหมายของรหัสคำสั่งและทำงานตามคำสั่งให้เรียบร้อยก่อน จากนั้นจึงจะรับข้อมูลหรือดำเนินการต่อไป

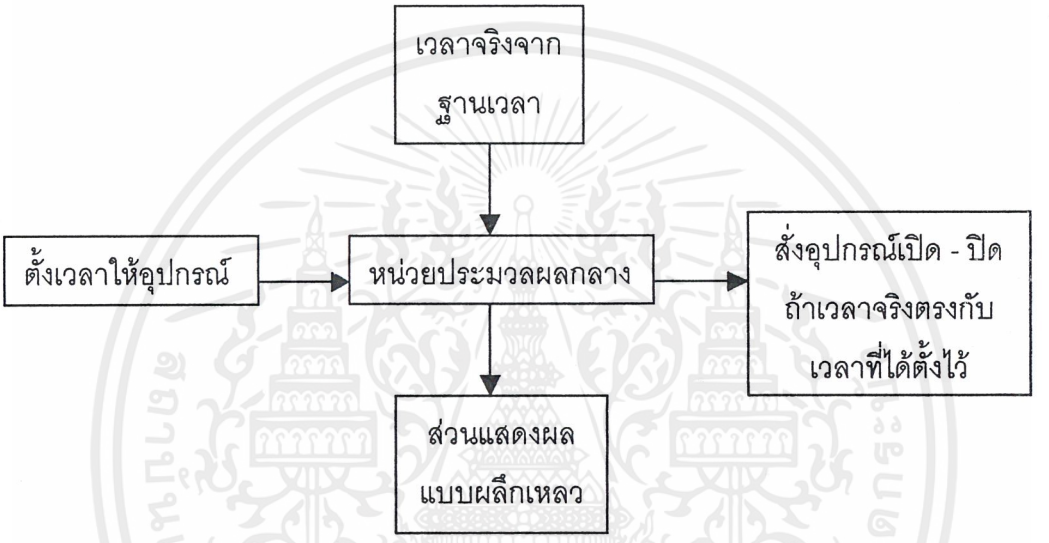
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้น ในการใช้งานส่วนแสดงผลแบบผลึกเหลว ต้องมีการหน่วงเวลารอให้ส่วนแสดงผลแบบผลึกเหลวพร้อมทำงาน จากนั้นก็จะกำหนดลอจิกให้แก่ขา RS ของส่วนแสดงผลแบบผลึกเหลว แล้วควรหน่วงเวลาไว้อีกชักพักเพื่อให้คอนโทรลเลอร์ภายในส่วนแสดงผลแบบผลึกเหลวแปลความหมายของลอจิกที่ขา RS ว่า ข้อมูลต่อไปที่จะได้รับนั้นเป็นรหัสคำสั่งหรือเป็นข้อมูลที่ต้องการแสดงผล จากนั้นจะเป็นการส่งข้อมูลมารอที่บัสข้อมูล D0-D7 ขั้นต่อไปจะเป็นการส่งสัญญาณพัลส์ไปที่ขา E เพื่ออีนาเบิลส่วนแสดงผลแบบผลึกเหลว ให้รับข้อมูลจากบัสข้อมูลเข้าไป โดยพัลส์ที่ป้อนเข้าที่ขา E ของส่วนแสดงผลแบบผลึกเหลวต้องเป็นพัลส์ขาขึ้น จากนั้นให้หน่วงเวลาไว้สักครู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบวงจร

3.1 การออกแบบฮาร์ดแวร์



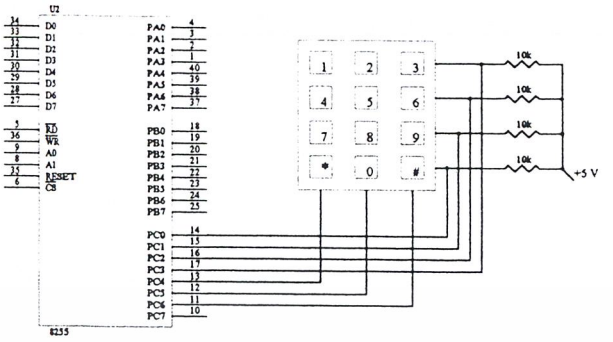
รูปที่ 3.1 แสดงบล๊อกไดอะแกรมการทำงานโดยรวมของวงจร

3.1.1 ภาครับข้อมูล

ในที่นี้เราใช้แป้นสวิตช์เมตริก เป็นส่วนรับข้อมูลจากผู้ใช้เพื่อติดต่อกับส่วนต่างๆ ของโครงงานนี้ เช่น การป้อนรหัสผ่าน การตั้งเวลาจริง การเลือกอุปกรณ์ การเลือกโปรแกรม และการโปรแกรมเวลาให้อุปกรณ์เครื่องใช้ไฟฟ้า เปิด – ปิด ภาครับข้อมูล

ส่วนรับข้อมูล จะทำการรับค่าจากการกดแป้นสวิตช์เมตริกซึ่งมีการต่อเป็นลักษณะ การต่อวงจรแบบเมตริก (Matrix Switch) สวิตช์จะถูกต่อกันในแนวแกนนอน เรียกว่าแถว (Row) และแกนตั้ง เรียกว่าหลัก (Column) และเพื่อให้ได้ค่าจากการกดสวิตช์ ต้องใช้หลักการสแกน (Scan) โดยมีหลักการเบื้องต้นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 วงจรเชื่อมต่อแป้นสวิตช์เมตริกเข้ากับ8255

จากการต่อวงจรดัง รูปที่3.2 จะใช้พอร์ตC ของ 8255 ซึ่งเชื่อมต่อกับไมโครคอนโทรลเลอร์ เชื่อมต่อเข้ากับแป้นสวิตช์เมตริกทั้ง 7 เส้น คือสายของหลัก 3 สาย C0-C2 และสายทางแถวอีก 4 สาย R0-R3 โดยที่สายแถวนี้จะต่อความต้านทานพูลอัพไว้เพื่อกำหนดสถานะเริ่มต้นที่ไม่มีการกดสวิตช์ โดยไมโครคอนโทรลเลอร์จะทำการส่งข้อมูล "0" ไปยังสายหลักทั้งสามตามลำดับ ในทุกครั้งที่มีการส่งข้อมูลไปยังสายหลักของแป้นสวิตช์เมตริก ไมโครคอนโทลเลอร์จะทำการอ่านค่าที่สายแถวทุกสายเข้ามาด้วย หากไม่มีการกดสวิตช์ ค่าของสายแถวก็จะยังเป็น "1" ทั้งหมด ถ้าหากมีการกดสวิตช์ ค่าของสายแถวจะไม่เป็น "1111" อีกต่อไป เป็นการบอกให้ทราบว่า มีการกดแป้นสวิตช์เมตริกขึ้นแล้ว จากนั้นไมโครคอนโทรลเลอร์ก็จะทำการค้นหาตำแหน่งต่อไปจากการค้นหาตำแหน่ง สิ่งที่ได้มาจะเป็นค่าของตำแหน่งของแป้นนั้น และนำค่าที่ได้ไปใช้ในกระบวนการต่อไป

3.1.2 ภาคประมวลผล

ส่วนประมวลผลที่ใช้เป็นไมโครคอนโทลเลอร์ ตระกูล MCS-51 เบอร์ AT89C52

3.1.2.1 ไมโครคอนโทรลเลอร์ ใช้ในการควบคุม ประมวลผลกลาง สามารถแบ่งแยกได้เป็น 2 ส่วนดังนี้

1) ส่วนของฮาร์ดแวร์ (Hardware) คือส่วนของภาคประมวลผลกลาง โดยจะแบ่งเป็นส่วนของไมโครคอนโทรลเลอร์เอง และส่วนที่ใช้ติดต่อกับอุปกรณ์ภายนอก

- ส่วนของไมโครคอนโทรลเลอร์
  - จะมีส่วนของวงจรรีเซต (Reset) และวงจรกำเนิดความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ติดต่อกับฐานเวลาจริง

- ส่วนที่ใช้ติดต่อกับอุปกรณ์ภายนอก
  - ติดต่อรับ – ส่ง ข้อมูลกับ 8255 เพื่อติดต่อกับอุปกรณ์ต่างๆได้เพิ่มขึ้น
  - ส่งข้อมูลออกไปทางพอร์ต 0 เพื่อนำไปแสดงผลยังส่วนแสดงผลแบบผลึกเหลว

2) ส่วนของ Software คือส่วนของโปรแกรมที่ใช้ควบคุมการทำงานของระบบทั้งหมด โครงงานชิ้นนี้มีการโปรแกรมลงไปในไมโครคอนโทรลเลอร์ ซึ่งใช้โปรแกรมที่เป็นภาษาเครื่องเก็บไว้ในหน่วยความจำสำหรับเก็บโปรแกรม โดยจะทำงานในลักษณะคล้ายมอนิเตอร์ (Monitor) คอยควบคุมระบบ และอุปกรณ์ภายนอกที่เชื่อมอยู่ สำหรับไมโครคอนโทรลเลอร์ AT89C52 นี้ สามารถเขียนขึ้นโดยภาษาแอสเซมบลี (Assembly) ที่ใช้คำสั่งเป็นคำที่สื่อความหมายอย่างย่อ (Mnemonic) แทนรหัสภาษาเครื่อง แล้วใช้ตัวแปรภาษา หรือ แอสเซมเบลอร์ (Assembler) แปลเป็นรหัสภาษาเครื่องได้

3.1.2.2 การเชื่อมต่อ 8255 กับ ไมโครคอนโทรลเลอร์

เมื่อพิจารณาแผนภาพของ 8255 จะเห็นว่ามีขาสัญญาณแอดเดรสจำนวน 2 เส้น คือ A0 และ A1 ทำให้แอดเดรสที่จะอ้างถึงได้มีค่าเป็น 2^2 หรือเท่ากับ 4ตำแหน่ง ซึ่งแต่ละตำแหน่งจะหมายถึงการระบุรีจิสเตอร์หรือพอร์ตภายใน 8255 ดังตารางต่อไปนี้

A1	A0	ชื่อของรีจิสเตอร์
0	0	พอร์ต A
0	1	พอร์ต B
1	0	พอร์ต C
1	1	รีจิสเตอร์ควบคุม

ตารางที่ 3.1 รหัสที่ใช้ในการเลือกพอร์ต

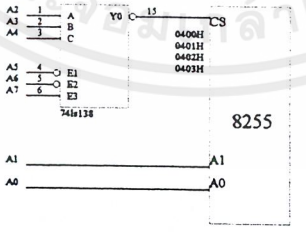
เมื่อพิจารณาค่าของแอดเดรสเหล่านี้รวมกับระดับลอจิกของขาสัญญาณ RD\และ WR\ จะเป็นการอ่านค่าหรือเขียนข้อมูลทางขาสัญญาณ D0-D7 ให้กับรีจิสเตอร์เหล่านั้นตามลำดับ ดังตารางต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RD\	WR\	A1	A0	ความหมาย
0	1	0	0	ส่งหรือเขียนข้อมูลให้กับพอร์ต A
1	0	0	0	รับหรืออ่านข้อมูลจากพอร์ต A
0	1	0	1	ส่งหรือเขียนข้อมูลให้กับพอร์ต B
1	0	0	1	รับหรืออ่านข้อมูลจากพอร์ต B
0	1	1	0	ส่งหรือเขียนข้อมูลให้กับพอร์ต C
1	0	1	0	รับหรืออ่านข้อมูลจากพอร์ต C
0	1	1	1	ส่งหรือเขียนข้อมูลให้กับรีจิสเตอร์ควบคุม
1	0	1	1	เป็นสภาวะที่ไม่ถูกต้อง

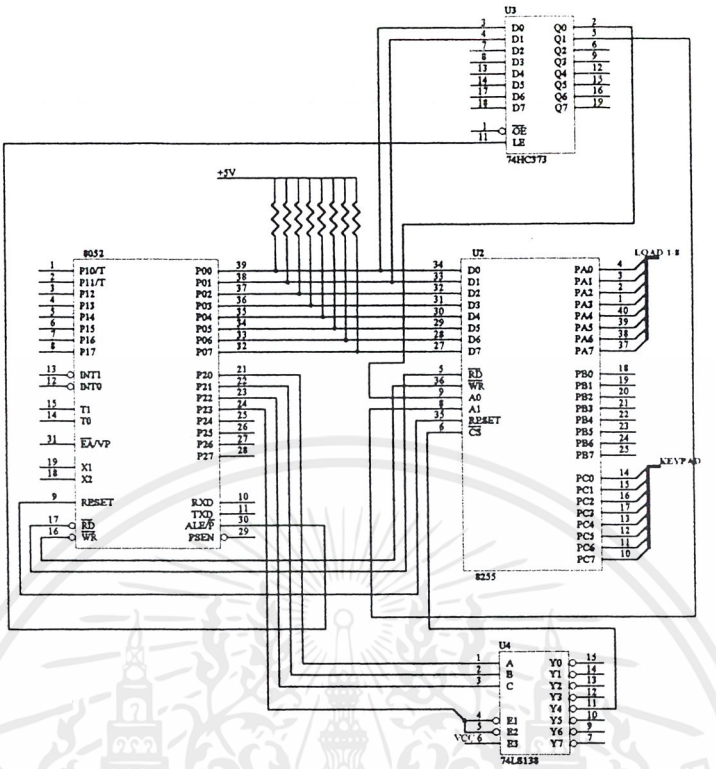
ตารางที่ 3.2 แสดงบิทที่ใช้ในการ รับ – ส่ง ของพอร์ตต่างๆ

ดังนั้นโดยทั่วไปจึงมักกำหนดให้แอดเดรสของ 8255 ทั้งสี่ตำแหน่งนี้ อยู่ในแอด - เดรสช่วงใดช่วงหนึ่งของระบบ เช่น 0400H, 0401H, 0402H และ0403H โดยขาสัญญาณแอด-เดรสที่นอกเหนือไปจาก A0 และ A1 นำมาข้ายังตัวถอดรหัสแอดเดรส เพื่อสร้างสัญญาณเลือกอุปกรณ์ (CS\) ในช่วงแอดเดรสที่ต้องการ อย่างตัวอย่างในรูป 3.3



รูปที่ 3.3 แผนภาพแสดงการสร้างสัญญาณลือกอุปกรณ์ (CS\) ให้กับ 8255 โดยการถอดรหัสจากบัสแอดเดรส A2-A7

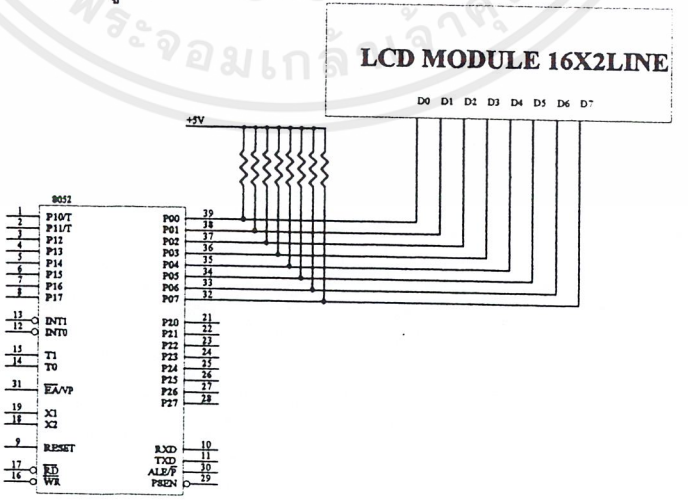
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 การเชื่อมต่อระหว่าง 8255 และ AT89C52

3.1.3 ภาคแสดงผล

การแสดงผลต่างๆ ในโครงงานนี้ ไม่ว่าจะเป็น วัน เดือน ปี เวลา ฯลฯ จะแสดงผลทางส่วนแสดงผลแบบผลึกเหลว โดยจะนำขาสัญญาณ D0 – D7 ของส่วนแสดงผลแบบผลึกเหลวต่อกับไมโครคอนโทรลเลอร์ทางพอร์ต 0 ดังรูปที่ 3.5

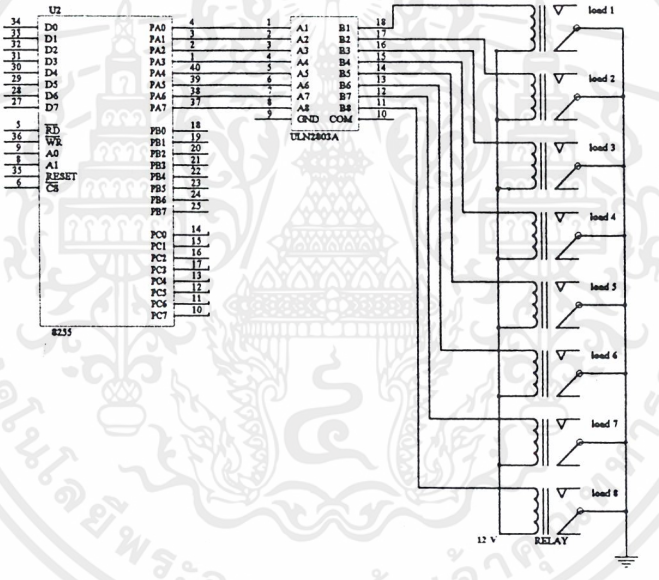


รูปที่ 3.5 แสดงการเชื่อมต่อ ส่วนแสดงผลแบบผลึกเหลว กับ ไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 ภาคขับโหลด

สัญญาณเอาท์พุตทั้ง 8 ช่องที่ส่งไปควบคุมการทำงาน เปิด – ปิด ของรีเลย์จะออกจากขาพอร์ต PA.0 – PA.7 จากนั้นสัญญาณทั้ง 8 เส้นนี้จะไปเข้าที่อินพุตของไอซี ULN2803 ซึ่งเป็นไอซีไดรเวอร์แบบอินเวอร์เตอร์ที่สามารถจ่ายกระแสได้สูงสุด 500 มิลลิแอมป์ และมีไดโอดป้องกัน (แรงดันย้อนกลับจากการยุบตัวของสนามแม่เหล็กในขดลวดรีเลย์) อยู่ภายใน และสามารถนำเอาสัญญาณเอาท์พุตที่ออกจากไอซี ULN2803 ทั้ง 8 เส้นไปใช้ขับรีเลย์ 12 โวลต์ ทั้ง 8 ตัวได้โดยตรง ส่วนอีกขั้วหนึ่งของรีเลย์ทั้ง 8 ตัวจะต่ออยู่กับไฟบวก 12 โวลต์ รอไว้อยู่แล้ว ดังรูปที่ 3.6



รูปที่ 3.6 การเชื่อมต่อของ 8255 กับ รีเลย์

ส่วนการเชื่อมต่อวงจรทั้งหมดรวมกัน แสดงในภาคผนวก รูปที่ ข

3.2 การออกแบบซอฟต์แวร์

ดูโฟลวชาร์ตในภาคผนวก ก รูปที่ ก-1 ถึง ก-8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

4.1 การทดลอง

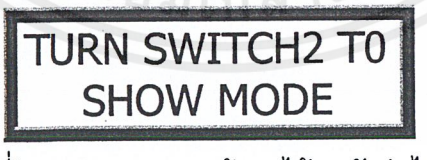
4.1.1 เมื่อสวิตช์อยู่ที่โหมดตั้งเวลา

ต้องใส่เวลาปัจจุบัน หรือเวลาที่ต้องการตั้งให้ฐานเวลาจริง

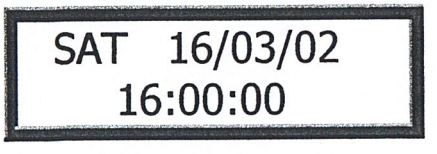
DAY	หมายถึง	วัน เช่น วันจันทร์ (MON) ให้กด “2”
DT	หมายถึง	วันที่ เช่น วันที่18 ให้กด “18”
MO	หมายถึง	เดือน เช่น เดือนมีนาคม ให้กด”3”
YR	หมายถึง	ปีคริสตศักราช เช่น ปี2002 ให้กดเฉพาะเลขท้าย 2 ตัว เท่านั้น เป็น “02”
HH	หมายถึง	ชั่วโมง ระบบ 24 ชั่วโมง เช่น เวลาบ่ายโมง ให้กด “13”
MM	หมายถึง	นาที เช่น 25นาที ให้กด “25”

ในที่นี้จะสมมติว่า จะตั้งเวลาให้เป็น วันเสาร์ ที่ 16 มีนาคม 2002 เวลา 16.00น.
ให้กด “7”,”16”,”03”,”02”,”16”และ”00”ตามลำดับ

แล้วหน้าจอของส่วนแสดงผลแบบผลึกเหลว จะแสดงผลดังนี้



จากนั้น ให้เลื่อนสวิตช์2 ไปอยู่ที่โหมดแสดงเวลา แล้วจะได้ผลดังต่อไปนี้ และนาฬิกาจะเริ่มทำงาน



ผลการทดลอง สามารถเขียนเวลาลงในไอซีสร้างฐานเวลาจริงได้ และนำข้อมูลที่เขียน ไปแสดงผลยังส่วนแสดงผลแบบผลึกเหลว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 เมื่อสวิตช์อยู่ที่โหมดโปรแกรม

เมื่อเข้ามาในโหมดนี้ ก็จะเริ่มจากการป้อนรหัสผ่านก่อน จากนั้นส่วนแสดงผลแบบผลึกเหลวจะแสดงผลดังนี้

หมายความว่า ถ้ากด "*" หมายถึง ต้องการเปลี่ยนรหัสผ่าน (Change Password)

ถ้ากด "0" หมายถึง ออกจากโหมดการทำงานนี้

ถ้ากด "#" หมายถึง เข้าสู่การโปรแกรม

ในที่นี้จะเป็นการโปรแกรม จึงเลือกกด "#" เพื่อเข้าสู่การโปรแกรม

จากนั้นส่วนแสดงผลแบบผลึกเหลวจะแสดงผลดังนี้

หมายความว่า ให้เลือกอุปกรณ์ที่ต้องการจะทำการโปรแกรม มีอุปกรณ์ทั้งหมด 8 อุปกรณ์

ในที่นี้ ต้องการเลือกอุปกรณ์ที่ 1 จึงเลือกกด "1"

จากนั้นส่วนแสดงผลแบบผลึกเหลวจะแสดงดังนี้

หมายความว่า ให้เลือกโปรแกรมที่ต้องการจะใช้ ตั้งแต่ 1-8

กด "0" หมายถึงออกไปเลือกอุปกรณ์ใหม่

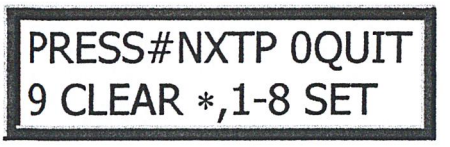
โดยที่ โปรแกรมเลขคี่ (1, 3, 5และ7) จะหมายถึงการสั่งให้อุปกรณ์เริ่มการทำงาน

โปรแกรมเลขคู่ (2, 4, 6และ8) จะหมายถึงการสั่งให้อุปกรณ์หยุดการทำงาน

ในที่นี้เลือกโปรแกรมที่ 1 (ให้อุปกรณ์ทำงาน) จึงกด"1"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นส่วนแสดงผลแบบผลึกเหลวจะแสดงดังนี้



หมายความว่า ถ้ากด "#" หมายถึง โปรแกรมถัดไป (Next Program)

ถ้ากด "0" หมายถึง ออกไปเลือกโปรแกรมใหม่

ถ้ากด "9" หมายถึง ลบข้อมูลในโปรแกรมนั้น (Clear)

ถ้ากด "8" หมายถึง ทุกวัน (Everyday: EDAY)

ถ้ากด "1-7" หมายถึง วันอาทิตย์ (SUN) – วันเสาร์ (SAT)

ถ้ากด "*" หมายถึง ทุกๆวันนั้น โดยต้องระบุวันที่ต้องการตามด้วย เช่น ต้องการเลือกให้อุปกรณ์เริ่มทำงาน ทุกๆ วันเสาร์ ต้องกด "*" ตามด้วย "7"

หมายความว่า ในอุปกรณ์ที่ 1 โปรแกรมที่ 1 ต้องการจะให้ทำงานวันอะไร เวลาเท่าไร

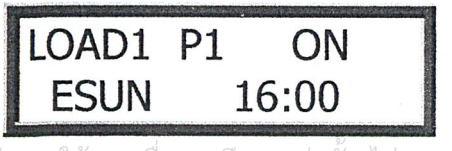
DAY หมายถึง วันที่ต้องการสั่งให้อุปกรณ์นี้ทำงาน ให้เลือกกด "*","1-8" ตามที่ได้กล่าวมาแล้วข้างต้น

HH:MM หมายถึง เวลา ชั่วโมง:นาที (ระบบ 24 ชั่วโมง)

ในที่นี้ต้องการ ให้อุปกรณ์ที่ 1 เริ่มทำงาน ทุกวันอาทิตย์เวลา 16.00น. ต้องทำดังนี้

ให้กด "*","1","16"และ"00" ตามลำดับ

จากนั้นส่วนแสดงผลแบบผลึกเหลวจะแสดงดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากนั้น ออกจากโปรแกรม โดยกด"0" และเลื่อนสวิตช์1 ให้อยู่ที่โหมดทำงานอัตโนมัติ (Auto) หากต้องการจะดูว่า ได้ใส่โปรแกรมอะไรไว้บ้างก็สามารถดูได้

ผลการทดลอง เมื่อตั้งเวลาในอุปกรณ์ที่ 1 โปรแกรมที่ 1 แล้ว และเวลาที่ได้ตั้งเอาไว้ตรงกับฐานเวลาจริง จะทำให้อุปกรณ์ที่ 1 เริ่มทำงานโดยจะแสดงผลออกมาทางหลอดไฟที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทวิจารณ์และสรุป

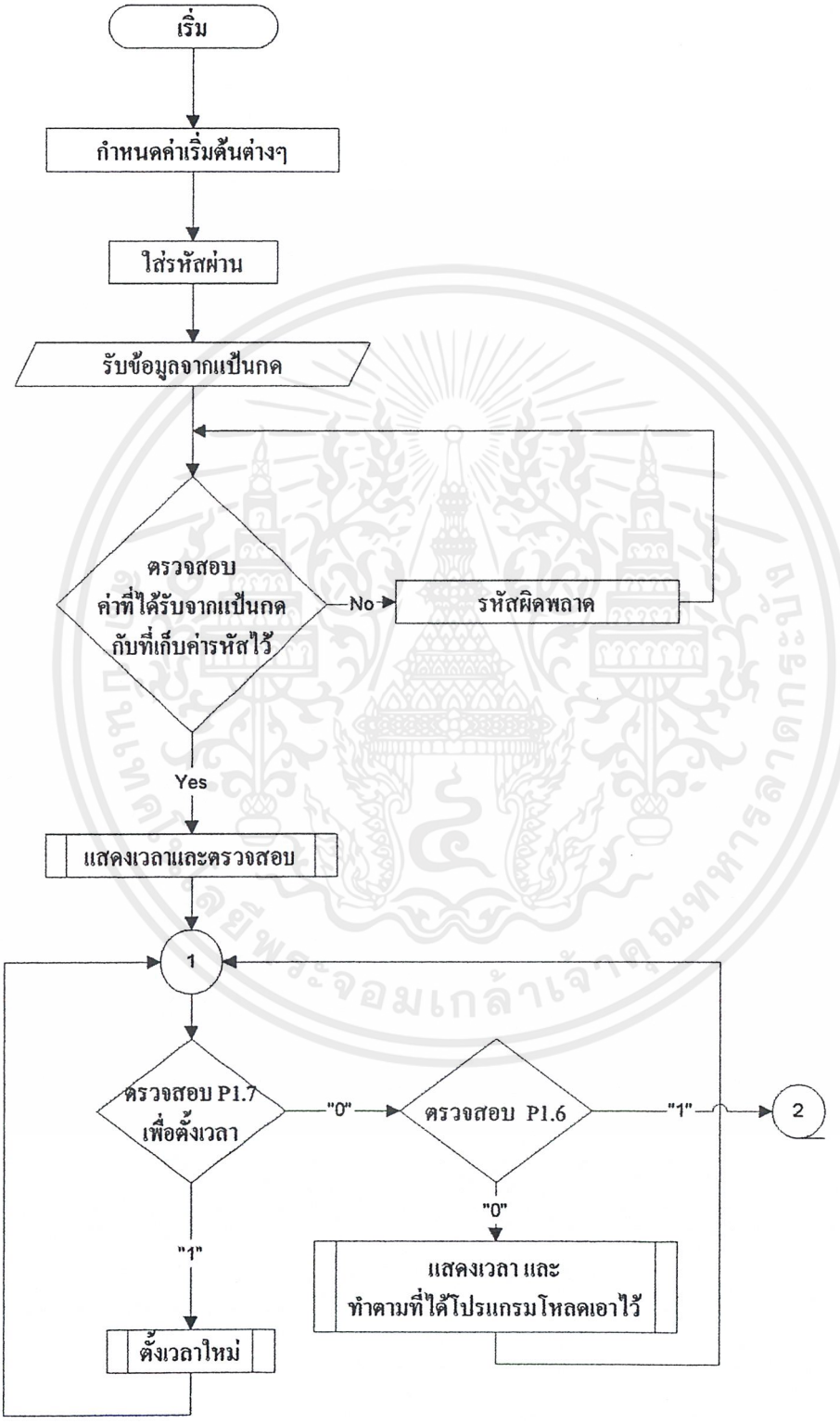
ปริญญานิพนธ์ฉบับนี้ได้นำเสนอวิธีการสร้างเครื่องควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าภายในบ้านแบบโปรแกมได้ ที่สามารถควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าได้ 8 อุปกรณ์ โดยแต่ละอุปกรณ์สามารถถูกสั่งงานได้อีก 8 โปรแกรมย่อย โดยแบ่งเป็น คำสั่งให้อุปกรณ์ทำงาน 4 คำสั่ง และคำสั่งให้อุปกรณ์หยุดการทำงานอีก 4 คำสั่ง ในการทดลองนั้นเราจะดูผลการทำงานได้จากหลอดไฟ ที่จะเชื่อมติดต่อกับรีเลย์ ซึ่งเมื่อรีเลย์ทำงานก็จะทำให้หลอดไฟทำงานด้วย ดังมีรายละเอียดอยู่ในบทที่ 4

ผลการทดลองจะเห็นว่า การติดต่อระหว่างผู้ใช้กับเครื่องควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าภายในบ้านแบบโปรแกรมได้จะทำได้โดยการสื่อสารผ่านแป้นสวิตช์เมตริกและส่วนแสดงผลแบบผลึกเหลว สามารถเขียนข้อมูลลงในไอซีสร้างฐานเวลาจริงได้ ซึ่งก็คือการตั้งเวลาให้กับเครื่องนั่นเอง และการป้อนคำสั่งให้อุปกรณ์ต่างๆ ทำงานตามที่เราต้องการ นอกจากนั้นเมื่อเวลาจากฐานเวลาจริง ตรงกับเวลาที่เราตั้งให้อุปกรณ์ต่างๆ แล้ว อุปกรณ์เหล่านั้นก็จะทำงานตามที่เราได้โปรแกรมเอาไว้ โดยแสดงผลผ่านทางหลอดไฟ และสามารถเรียกดูโปรแกรมต่างๆ ที่ได้โปรแกรมไว้แล้วได้ด้วย

เนื่องจากโครงงานนี้ได้จัดทำขึ้นเป็นครั้งแรก ขนาด ความสวยงาม และ การใช้งานอาจจะทำได้ไม่ดีนัก และ อาจจะยังมีข้อผิดพลาดบางอย่างที่ผู้จัดทำอาจคาดไม่ถึง ซึ่งถ้าหากจะมีการนำโครงงานนี้ไปใช้งานจริง ก็ควรจะต้องมีการปรับปรุงอยู่บ้าง เช่น การติดตั้งเต้ารับกับตัวเครื่อง หรือจะเป็นการเดินสายไฟไปยังอุปกรณ์ไฟฟ้า การปรับปรุงส่วนโปรแกรมเพื่อให้เกิดความเข้าใจระหว่างผู้ใช้กับ เครื่องควบคุมอุปกรณ์เครื่องใช้ไฟฟ้าภายในบ้านแบบโปรแกรมได้ให้ง่ายขึ้น เป็นต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

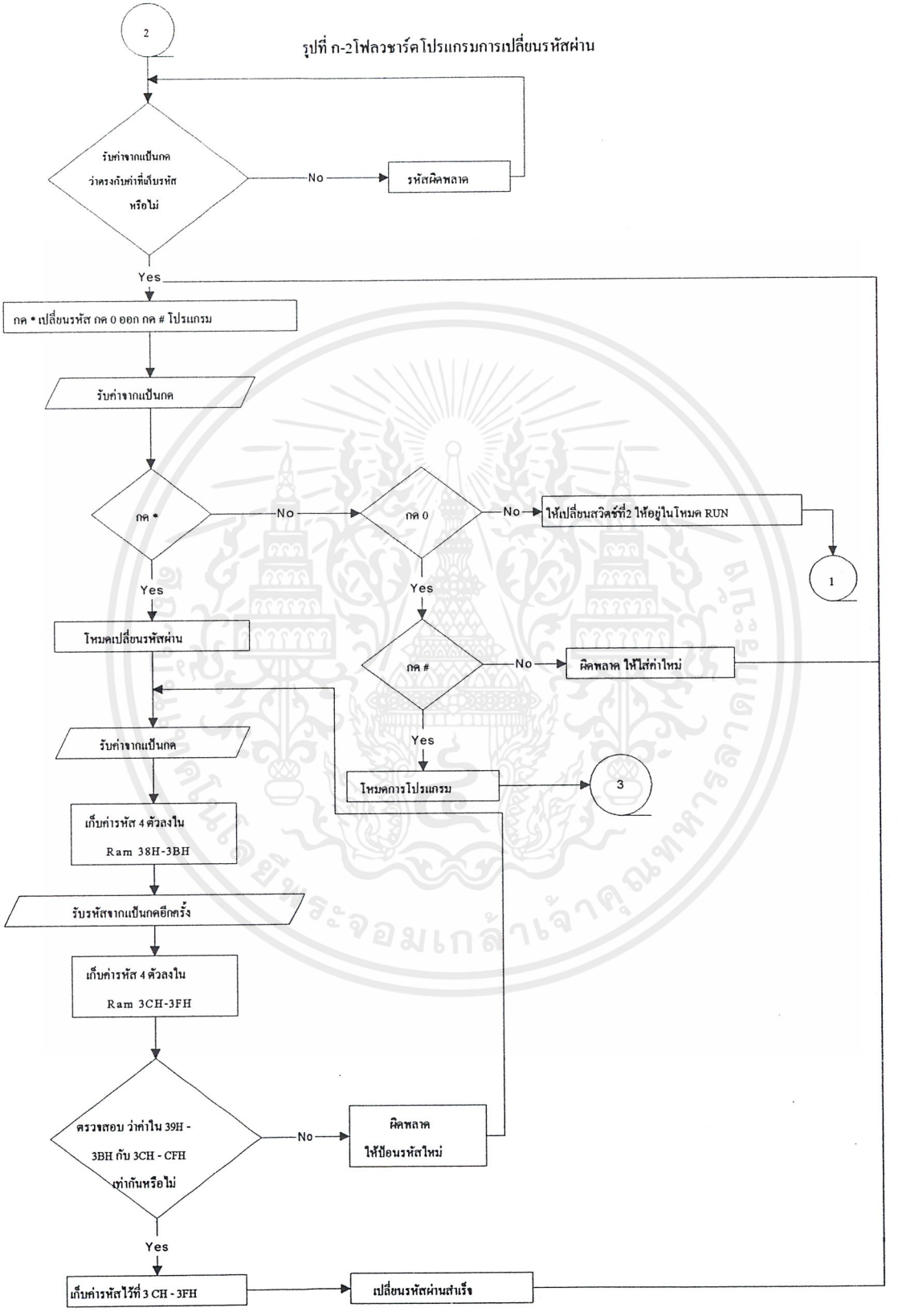
ภาคผนวก ก



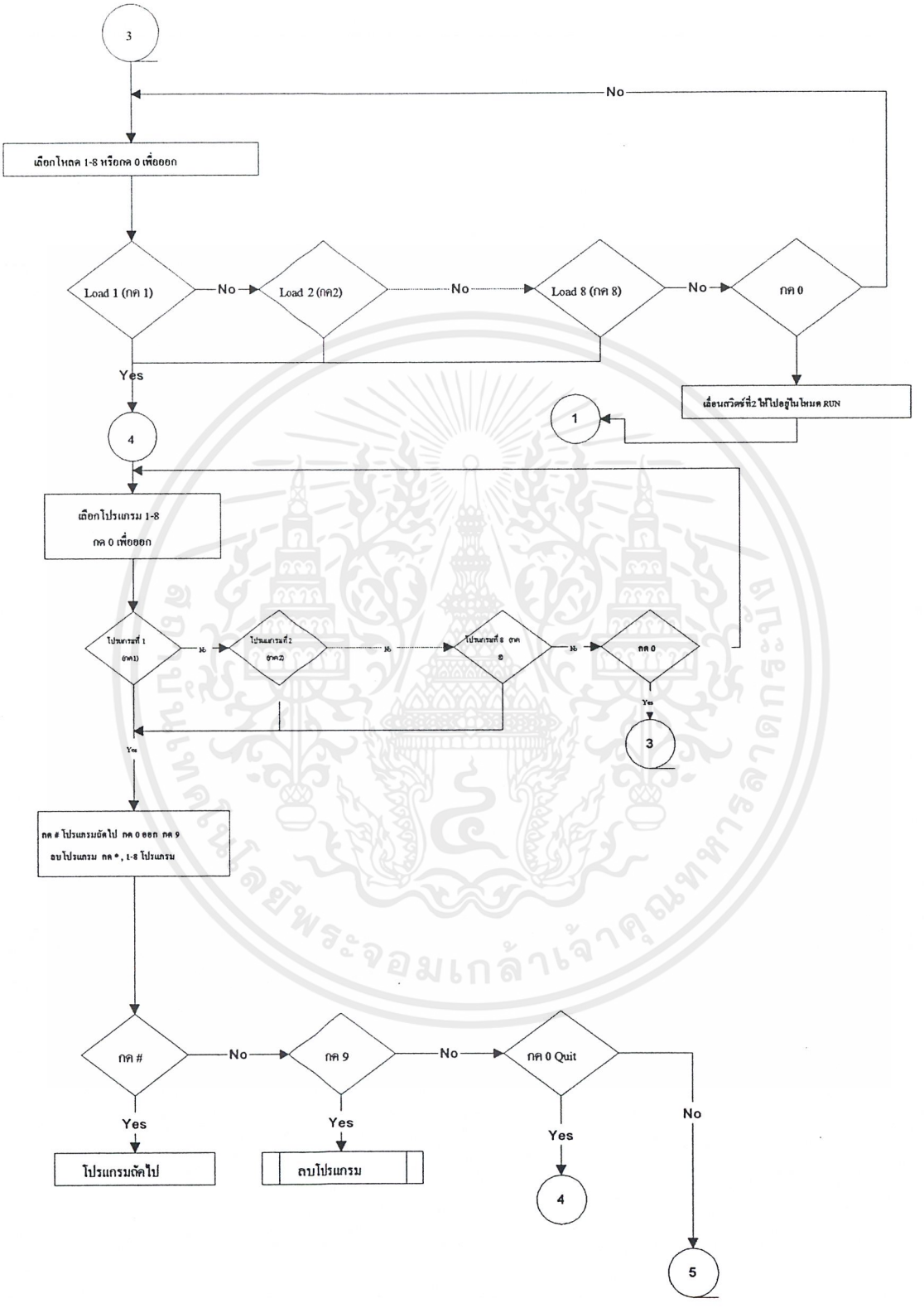
รูปที่ ก-1 โฟลวชาร์ตโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ก-2โฟลวชาร์ตโปรแกรมการเปลี่ยนรหัสผ่าน

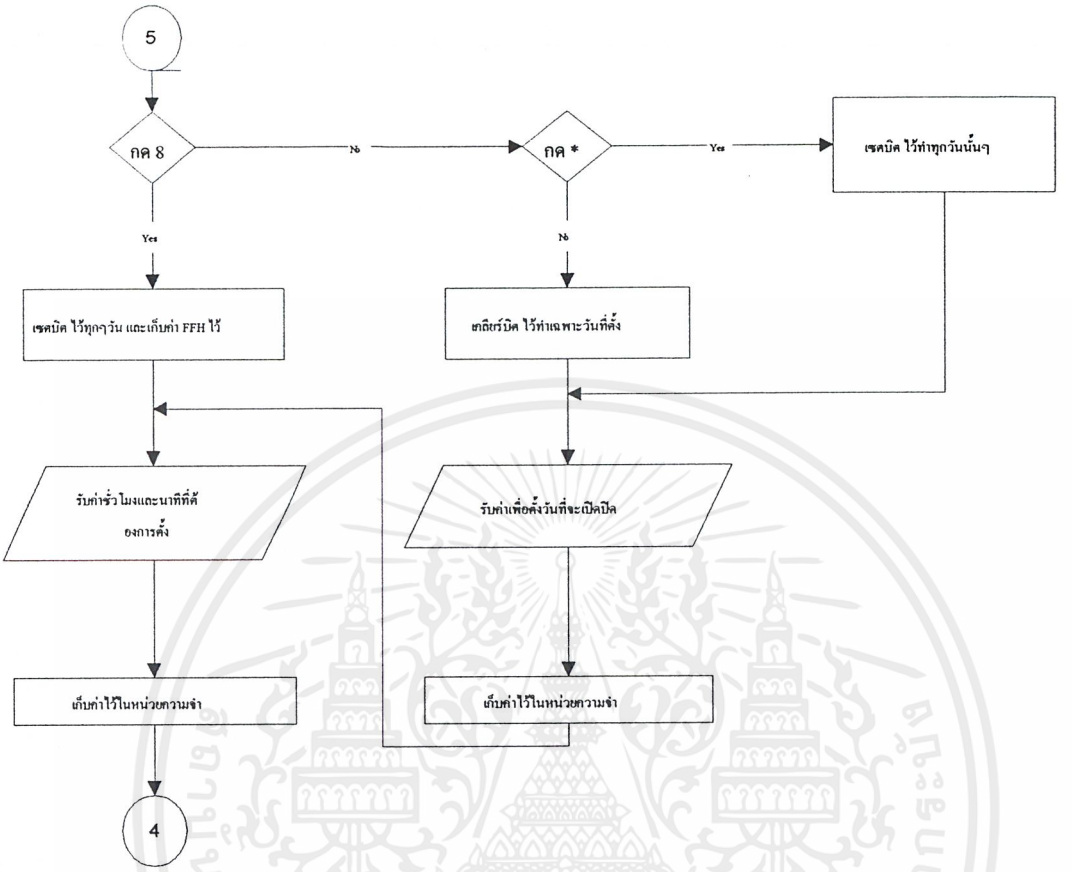


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



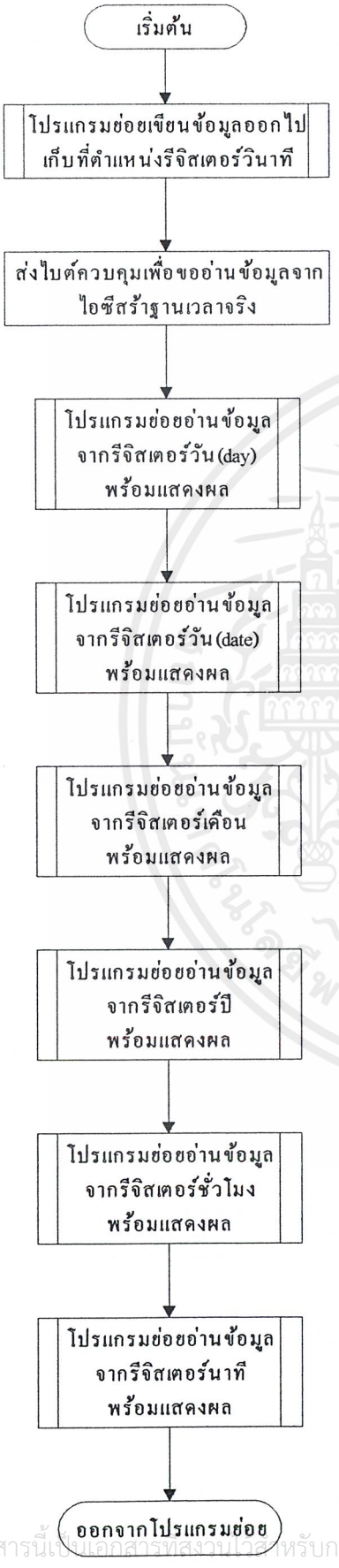
รูปที่ ก-3โฟลวชาร์ตการตั้งเวลาแต่ละโหลดและแต่ละโปรแกรม (1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



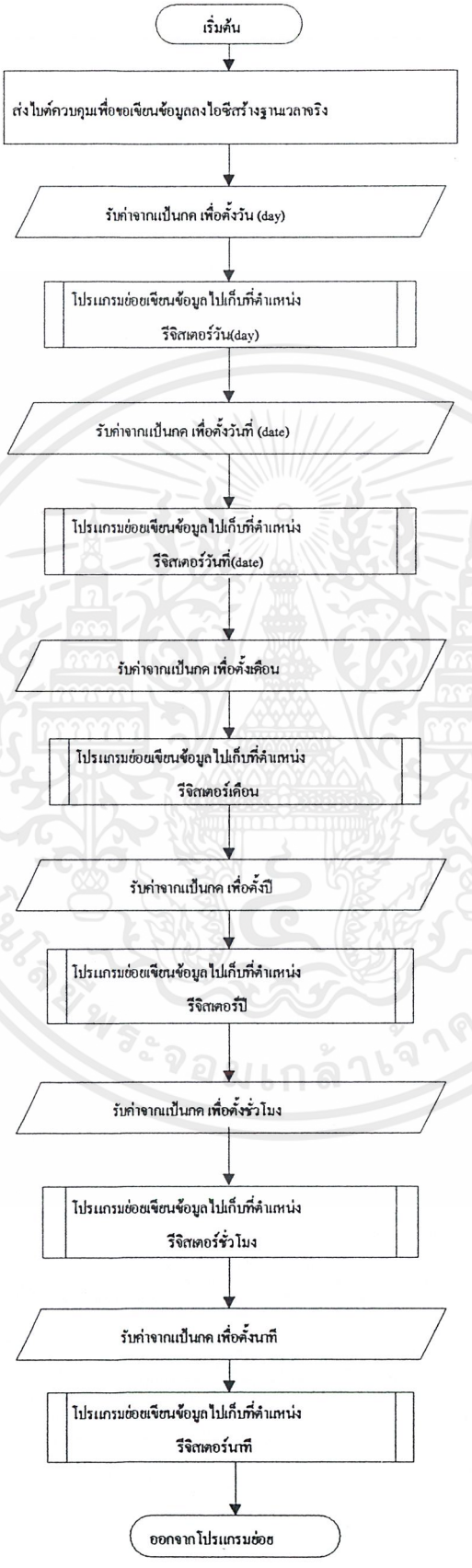
รูปที่ ก-4 โฟลวชาร์ตการตั้งเวลาแต่ละโหลดและแต่ละโปรแกรม (2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



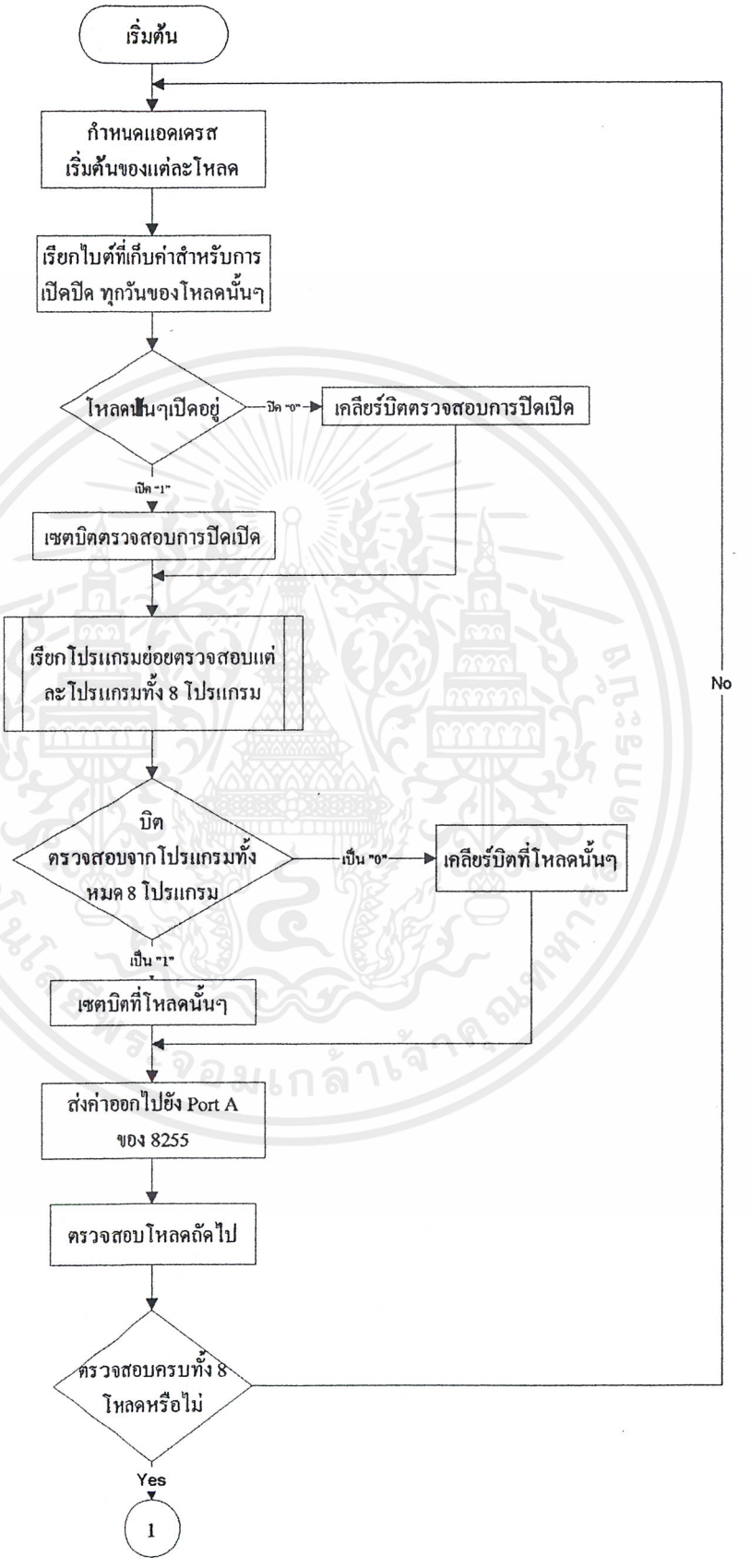
รูปที่ ก-5
โฟลวชาร์ตโปรแกรมย่อยการอ่าน
ค่าจากไอซีสร้างฐานเวลาจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ก-6 โฟลวชาร์ตโปรแกรมย่อยการเขียนข้อมูลไปยังไอซีสร้างฐานเวลาจริง

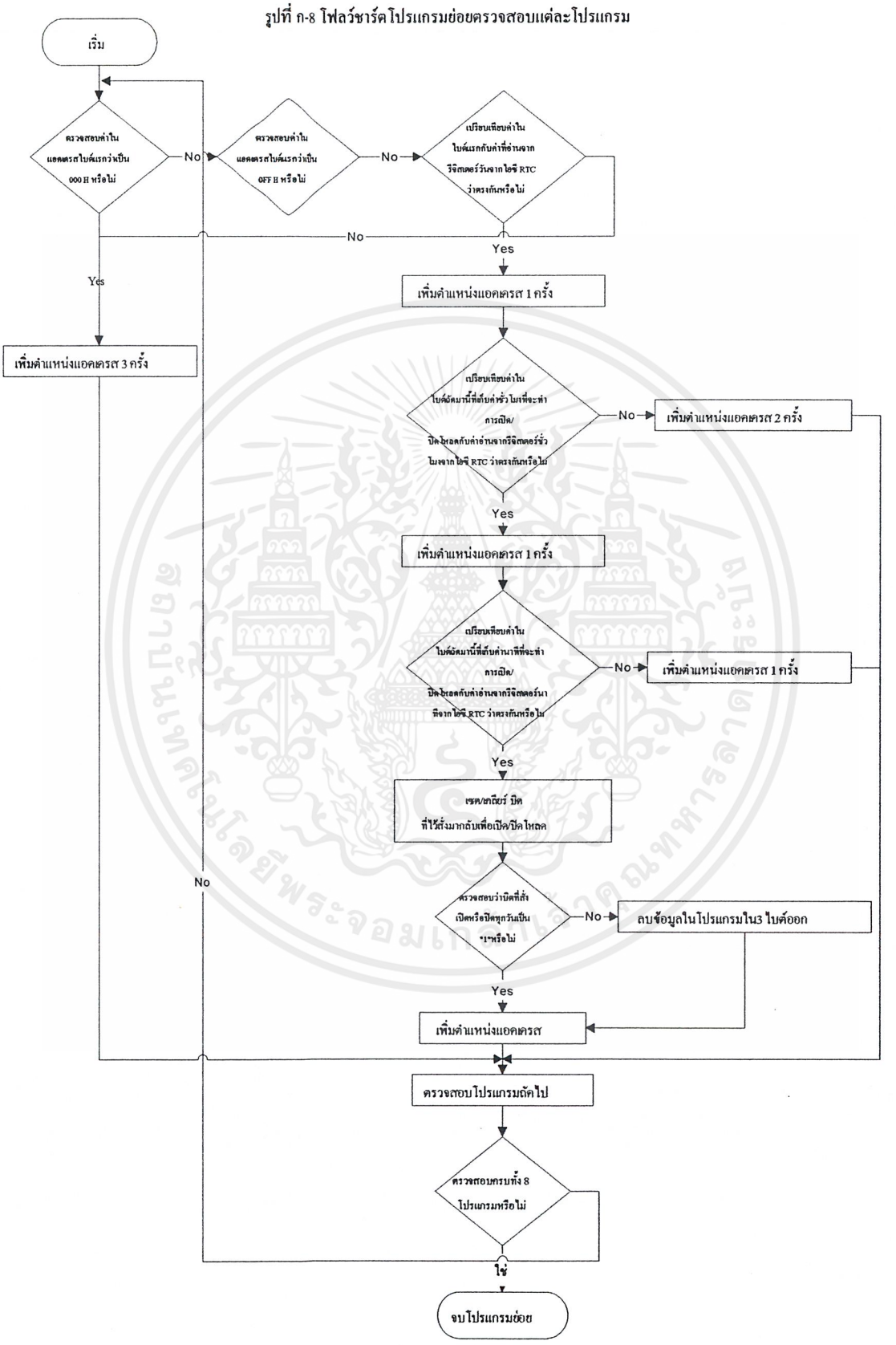
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



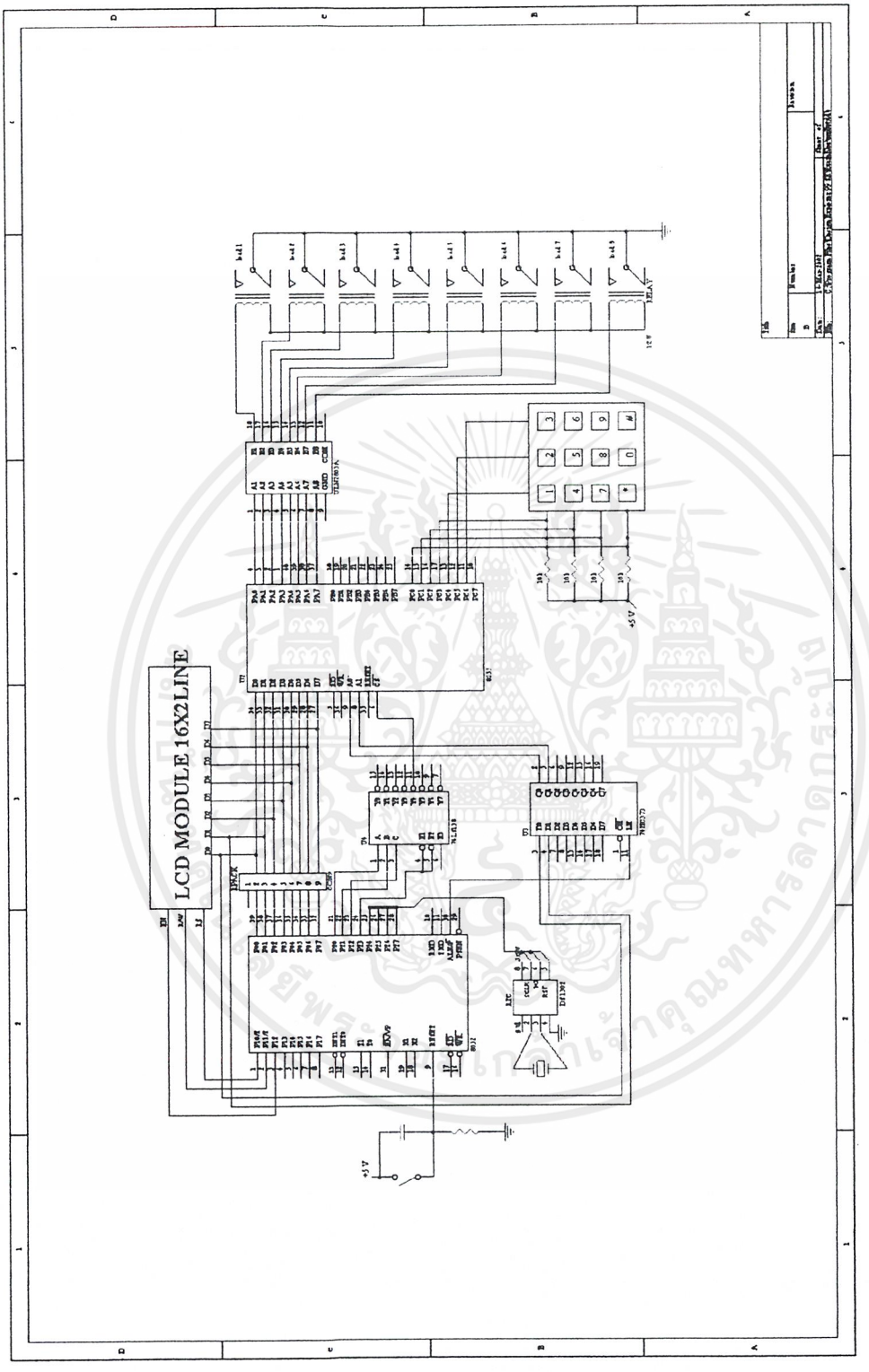
รูปที่ ก-7 โฟลวชาร์ตโปรแกรมย่อยตรวจสอบแต่ละโหลด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ก-8 โฟลว์ชาร์ตโปรแกรมย่อยตรวจสอบแต่ละโปรแกรม



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ข แสดงการเชื่อมต่อวงจรทั้งหมดเข้าด้วยกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานและปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงไปด้วยดี เนื่องจากได้รับความช่วยเหลืออย่างดียิ่งของผู้ช่วยศาสตราจารย์ ดร.เกียรติศักดิ์ คมวัชระ อาจารย์ที่ปรึกษาโครงงาน ซึ่งได้ให้คำแนะนำและข้อคิดเห็นต่างๆ ที่เป็นประโยชน์ต่อการทำโครงงานด้วยดีตลอดมา ผู้จัดทำโครงงานขอขอบคุณคณะอาจารย์ภาควิชาวิศวกรรมระบบควบคุมที่คอยผลักดันและกระตุ้นให้ทำโครงงานจนสำเร็จลุล่วงไปได้ด้วยดี

สุดท้ายนี้ ผู้จัดทำโครงงานใคร่กราบขอบพระคุณ บิดา มารดา และขอขอบคุณพี่ๆ น้องๆ และเพื่อนทุกคนที่ให้กำลังใจและความช่วยเหลือ แก่ผู้จัดทำโครงงานเสมอมาจนสำเร็จการศึกษา

เสริมศรี จารุวัฒนดิลก

อรวรรณ สุขสวัสดิ์

มีนาคม 2544

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

ชัยวัฒน์ ลิ้มพรจิตรวิไล และวรพจน์ กรแก้ววัฒนกุล, "เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ MCS-51 Flash Microcontroller", อินโนเวตีฟ เอ็กซ์เพอริเมนต์, 475 หน้า, 2542.

สมยศ จุณณะปิยะ "การใช้งานไมโครคอนโทรลเลอร์ MCS-51", ภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง, 358 หน้า, 2543.

สุนทร วิทูรสุรพจน์ "การโปรแกรมภาษาแอสเซมบลีของไมโครคอนโทรลเลอร์ตระกูล 8051", ซีเอ็ดยูเคชั่น, 198 หน้า, 2537.

สุเจตน์ จันทรังษ์ "ไมโครคอนโทรลเลอร์ซิพเดี่ยว 8051", วิทยาลัยมหานคร, 178 หน้า, 2535.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้