สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

สวิทช์ควบคุมระยะไกลด้วยรหัสอินฟราเรด

SWITCH REMOTE CONTROL BY INFARRED CODE

นางสาวสิรินรัตน์ เรืองเศรษฐวิศาล

นายสุระสิทธิ์ รุจิสฤษฏ์วงศ์

เลขหมู่.....
เลขทะเบียน 72109
วัน,เดือน,ปี - 8 มิ.ย. 2550

b 11763501
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมการวัดคุม

ภาควิชาวิศวกรรมการวัดคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SWITCH REMOTE CONTROL BY INFARRED CODE

SIRINRAT RUANGSETWISAL

SURASIT RUGISARITWONG

A THESIS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIRMENT FOR THE DEGREE OF

BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING

DEPARTMENT OF INSTRUMENTATION ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

2006

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาควิชาวิศวกรรมการวัดคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

.....

หัวข้อปริญญานิพนธ์ สวิทช์ควบคุมระยะไกลด้วยรหัสอินฟราเรด

SWITCH REMOTE CONTROL BY INFARRED CODE

นักศึกษาผู้จัดทำ นางสาวสิรินรัตน์ เรืองเศรษฐวิศาล **รหัสนักศึกษา** 46012133

นายสุระสิทธิ์ รุจิสฤษฎ์วงศ์ **รหัสนักศึกษา** 46012139

ปริญญา วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมการวัดคุม

ปีการศึกษา 2549

อาจารย์ผู้ควบคุมปริญญานิพนธ์	ลายมือชื่อ
รศ. วิริยะ กองรัตน์	

ภาควิชารับรองแล้ว

(รศ. ประภาษ อุคคกิมาพันธุ์)

หัวหน้าภาควิชาวิศวกรรมการวัดคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์	**สวิทช์ควบคุมระยะไกลด้วยรหัสอินฟราเรด**
	SWITCH REMOTE CONTROL BY INFARRED CODE
นักศึกษาผู้จัดทำ	**นางสาวสิรินรัตน์ เรืองเศรษฐวิศาล**
	นายสุระสิทธิ์ รุจิสฤษฎ์วงศ์
อาจารย์ที่ปรึกษา	**รศ. วิริยะ กองรัตน์**
ปีการศึกษา	2549

บทคัดย่อ

โครงงานนี้เป็นการทดลองออกแบบวงจรสวิตช์ควบคุมแสงสว่างของหลอดไฟ โดยใช้ไมโครคอนโทรลเลอร์ ทำงานร่วมกับอุปกรณ์ส่งสัญญาณรหัสอินฟราเรด เพื่อให้ทำการควบคุมอุปกรณ์แสงสว่างทั้งแบบเปิด-ปิด และแบบควบคุมมุมทริกของหน่วยควบคุมกำลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Thesis Title	Switch Remote Control By Infrared Code
Authors	Miss.Sirinrat Ruangsetwisal
	Mr.Surasit Rugisaritwong
Thesis Advisor	Assoc. Prof. Viriya Kongrat
Year	2549

ABSTRACT

This project is designed and construct switch controlled lighting system by micro controller. The system is used device control to controlled the lighting system in on-off or dimming mode manner .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

เนื่องจากสวิทช์ควบคุมระยะไกลด้วยรหัสอินฟราเรดนี้ประกอบด้วยส่วนสำคัญ 2 ส่วน คือฮาร์ดแวร์ และ ซอฟท์แวร์ จึงจำเป็นต้องใช้ความรู้และการค้นคว้าอย่างมาก ซึ่งความซับซ้อนทางกลวิธีทางส่วนของโปรแกรมนั้นมากเป็นพิเศษ ซึ่งคณะผู้จัดทำมีประสบการณ์ด้านนี้ไม่มากทั้งนี้ คณะผู้จัดทำได้ความอนุเคราะห์ด้านคำปรึกษาชี้แนะในแนวทางที่ถูกต้องของอาจารย์ผู้มีพระคุณ ซึ่งนอกจากจะได้รับการสั่งสอน ถ่ายทอดความรู้แล้วยังได้รับความห่วงใยต่อคณะผู้จัดทำฉันท์ศิษย์และอาจารย์อีกด้วย

ฉะนั้น ในโอกาสอันเหมาะสมนี้ ทางคณะผู้จัดทำขอได้กล่าวคำขอบพระคุณในความรู้ความห่วงใย และความปรารถนาดีที่คณะผู้จัดทำได้รับจากอาจารย์วิริยะ กองรัตน์ ณ ที่นี้ด้วยความสำนึกในพระคุณอย่างสูง ส่วนดีของปริญญานิพนธ์นี้ ที่อาจได้มีโอกาสเผยแพร่ไปยังผู้แสวงหาความรู้ในด้านนี้ คณะผู้จัดทำขอมอบให้เป็นเกียรติต่ออาจารย์ที่มีส่วนช่วยเหลือต่อผลงานชิ้นนี้ ส่วนข้อผิดพลาดหรือแนวทางที่ผิดนั้นคณะผู้จัดทำขอน้อมรับไว้เพื่อการแก้ไขต่อไป

คณะผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

หน้า

บทคัดย่อภาษาไทย.....I

บทคัดย่อภาษาอังกฤษ.....II

กิตติกรรมประกาศ.....III

สารบัญ.....IV

สารบัญภาพ.....VII

สารบัญตาราง.....VIII

บทที่ 1 บทนำ.....1

1.1 ความสำคัญของปริญญานิพนธ์.....1

1.2 วัตถุประสงค์ของปริญญานิพนธ์.....1

1.3 ขอบเขตของปริญญานิพนธ์.....1

บทที่ 2 ทฤษฎีเบื้องต้นของไมโครคอนโทรลเลอร์.....2

2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ตระกูล MCS-51.....2

2.2 การจัดการขาของไมโครคอนโทรลเลอร์ MCS-51.....3

2.3 โครงสร้างและการทำงานของพอร์ต.....3

2.4 จังหวะการทำงานของไมโครคอนโทรลเลอร์.....4

2.5 การจัดการหน่วยความจำ.....4

2.5.1 หน่วยความจำโปรแกรม (Porgram Memory).....4

2.5.2 หน่วยความจำข้อมูล (Data Memory).....5

2.5.3 รีจิสเตอร์ฟังก์ชั่นพิเศษ (Special Function Register:SFR).....6

2.5.4 รีจิสเตอร์แสดงสถานะของโปรแกรม (Program Status Word:PSW).....6

2.5.5 แอกคิวมูเลเตอร์ (Accumulator:ACC).....6

2.5.6 รีจิสเตอร์ B.....7

2.5.7 โปรแกรมเคาน์เตอร์ (Program Counter : PC).....7

2.5.8 สแต็กพอยน์เตอร์ (Stack Pointer : SP).....7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

2.5.9 รีจิสเตอร์ชี้ข้อมูลหรือดาต้าพอยน์เตอร์ (DataPointer:DPTR).....8
2.5.10 รีจิสเตอร์พอร์ต (Port Register)..... 8
2.5.11 รีจิสเตอร์บัฟเฟอร์ข้อมูลอนุกรม (Serial Data Buffer : SBUF).....8
2.5.12 รีจิสเตอร์ไทเมอร์ (Timer Register)..... 8
2.5.13 รีจิสเตอร์แคปเจอร์ (Capture Register).....9
2.5.14 รีจิสเตอร์ควบคุม (Control Register).....9
2.6 การเขียนโปรแกรมแอสเซมบลี.....11

บทที่ 3 ทฤษฎีของไมโครคอนโทรลเลอร์ 89C51.....13
3.1 สถาปัตยกรรมของ 89C51.....13
3.2 การทำงานของ 89C51.....21
3.3 ไดอะแกรมเวลาของการติดต่อกับหน่วยความจำ.....24
3.4 การรีเซท..... 28
3.5 รีจิสเตอร์ของ 89C51.....29
3.6 พื้นที่หน่วยความจำข้อมูลแฝงของ AT89C52..... 33
3.7 รีจิสเตอร์ฟังก์ชั่นพิเศษ (Special Function Register ,SFR)..... 34
3.8 การรับ-ส่งข้อมูลทางพอร์ทอนุกรม..... 55
3.9 การขัดจังหวะ..... 58
3.10 เมโมรีแม็พไอโอเทคนิค.....61

บทที่ 4 ส่วนประกอบของโครงงาน..... 61
4.1 ส่วนประกอบของโครงงาน..... 61
4.1.1 ส่วนของ HARDWARE.....61
4.1.2 ส่วนของ SOFTWAR..... 61
4.2 คุณลักษณะของอุปกรณ์..... 63
4.3 หลักการทำงานของอุปกรณ์..... .64
4.4 การทำงานของรีโมตคอนโทรล.....64

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

4.5 วงจรและการทำงาน.....65

บทที่ 5 สรุปผลการทดลอง.....66

5.1 สรุปผลการทดลอง.....66

5.2 ปัญหาที่พบ.....66

บรรณานุกรม.....68

ภาคผนวก.....69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่ หน้า

2.1 การเลือกแบงก์ของหน่วยความจำส่วนล่างเพื่อติดต่อกับรีจิสเตอร์แบงก์R0-R7..... 7

3.1 แสดงการอ้างอิงหน่วยความจำสำหรับข้อมูลภายใน..... ..31

3.2 แสดงค่ารีจิสเตอร์ R0 ถึง R7 ใน BANK..... ..33

3.3 แสดงค่าการกำหนดโหมดใน 2 บิต..... 41

3.4 การเลือกโหมดใน M0 และ M1..... 43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่ หน้า

2.1 โครงสร้างหน่วยความจำ..... 6

2.2 รูป IC8051..... ...10

3.1 สถาปัตยกรรมภายในของ 89C51.....13

3.2 ไดอะแกรมขาของ 89C51 แบบ DIP..... 14

3.3 โครงสร้างของพอร์ท 0.....15

3.4 โครงสร้างของพอร์ท 1.....16

3.5 โครงสร้างของพอร์ท2..... ..17

3.6 โครงสร้างของพอร์ท 3..... 18

3.7 ค่าในริสเตอร์เมื่อเกิดการรีเซท 89C51..... 19

3.8 วงจรออสซิลเลเตอร์ภายใน 89C51..... 21

3.9 89C51 ที่ทำงานโดยสัญญาณที่มาจากภายนอก..... 21

3.10 ลำดับสถานะการทำงานใน MCS 51..... 23

3.11 Timing Diagram ของการอ่านโปรแกรมจากหน่วยความจำภายนอก..... 24

3.12 วงจรที่มี Program Memory ภายนอก 89C51..... 25

3.13 Timing Diagram ของการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก 89C51..... .. 26

3.14 Timing Diagram ของการเขียนข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก 89C51..... 27

3.15 วงจรที่มีหน่วยความจำสำหรับข้อมูลที่อยู่ภายนอก 89C51..... 27

3.16 ไดอะแกรมตามเวลาของการรีเซท 28

3.17 ไดอะแกรมภาพของหน่วยความจำ 89C51..... .29

3.18 ค่าตำแหน่งของแต่ละบิต..... 31

3.19 Program Status Word (PSW)..... ...32

3.20 Special Function Register (SFR)..... 34

3.21 แผนภาพค่าตำแหน่งหน่วยความจำแต่ละบิต.....35

3.22 ชุดข้อมูลอนุกรมในโหมด 1..... ...38

3.23 ชุดข้อมูลอนุกรมในโหมด 2.....38

3.24 Serial Port Control Register (SCON)..... 39

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.25TMODTimer/Counter mode Register	42
3.26 Timer Mode 0 : 13 bit count	43
3.27 Timer Mode 2	44
3.28 Timer 0 Mode 3	45
3.29 TCON Timer Control Register	46
3.30 แหล่งกำเนิดสัญญาณขัดจังหวะ	48
3.31 Interrupt Enable Register	49
3.32 IP : Interrupt Priority Register	50
3.33 ระบบการขัดจังหวะของ AT89C51	51
3.34 PCON : Power Control Register	52
3.35 Power down และ Idle mode	53
3.36 Serial Port mode 1	56
3.37 ไดอะแกรมเวลาของการตอบสนองการขัดจังหวะ	59
4.1 โฟลวชาร์ตแสดงการทำงานโปรแกรมการกดคีย์เพื่อหรี่ไฟ	62
4.2 โฟลวชาร์ตแสดงการทำงานโปรแกรมการเปิด-ปิดไฟ	63
4.3 แสดงรูปคลื่นที่จุดต่างๆของการหรี่ไฟด้วยไมโครคอนโทรลเลอร์	64
5.1 รูปแสดงอุปกรณ์ ELECTRONIC ของอุปกรณ์	67
5.2 รูปแสดงอุปกรณ์ทั้งหมดของสวิตช์ควบคุมระยะไกลด้วยรหัสอินฟราเรด	67

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความสำคัญของปริญญานิพนธ์

ในปัจจุบันวัฒนาการของโลกก้าวไปไกลไม่หยุดยั้ง อุปกรณ์ไฟฟ้าต่างๆได้เข้ามามีบทบาทสำคัญต่อการดำรงชีวิตประจำวันของมนุษย์มากมาย อาทิเช่น แอร์ ทีวี ระบบไฟฟ้า วิทยุ และอื่นๆ ซึ่งในอดีตการควบคุมอุปกรณ์เหล่านี้ยังต้องใช้ตัวคนในการควบคุม ซึ่งอุปกรณ์บางชนิดที่ทำงานประจำเราจึงต้องคอยควบคุมอยู่บ่อยครั้งหรือแม้แต่ในบางครั้งเราต้องการควบคุมสิ่งเหล่านั้นในเวลาที่กำหนด เราก็ต้องคอยควบคุมด้วยคนแทน

แต่ในปัจจุบันนี้เราสามารถควบคุมอุปกรณ์เหล่านั้นได้ โดยไม่ต้องใช้คนเป็นผู้ควบคุม ซึ่งคือเราจะใช้รีโมตเป็นตัวควบคุมแทน มันจะคอยอำนวยความสะดวกและเป็นประโยชน์อย่างยิ่งในการควบคุมอุปกรณ์เหล่านี้

สวิทช์ควบคุมระยะไกลด้วยรหัสอินฟราเรดนี้ เป็นอุปกรณ์ที่ใช้งานง่ายการควบคุมซับซ้อนผู้ใช้งานไม่ต้องมีความรู้มากในการใช้งานดังนั้นจึงเหมาะกับทุกสภาพการใช้งาน

สวิทช์ควบคุมระยะไกลด้วยรหัสอินฟราเรดนี้เป็นการนำเอาวงจรที่มี ไมโครคอนโทรลเลอร์ (Micro controller) เป็นหัวใจสำคัญในการประมวลผลหรือเป็นตัวควบคุมอุปกรณ์ไฟฟ้าต่างๆ

1.2 วัตถุประสงค์ของปริญญานิพนธ์

1. เพื่อศึกษาการทำงานของไมโรคอนโทรลเลอร์ AT89C51 และสามารถนำไปประยุกต์ใช้งานร่วมกับรีโมต
2. เพื่อนำความทางทฤษฎีมาประยุกต์ใช้งานได้จริง
3. เพื่อให้สามารถนำอุปกรณ์ทางอิเล็กทรอนิกส์มาทำการพัฒนาให้มีประสิทธิภาพและเกิดคุณค่าสูงสุด
4. เพื่อสร้างเครื่องควบคุมทางอิเล็กทรอนิกส์ ที่สามารถนำไปใช้ควบคุมอุปกรณ์ไฟฟ้าได้

1.3 ขอบเขตของปริญญานิพนธ์

1. ออกแบบวงจรเพื่อให้สามารถควบคุมการเปิด-ปิดไฟจากรีโมตคอนโทรลได้
2. สร้างโปรแกรมภาษาแอสเซมบลีของไมโครคอนโทรลเลอร์ AT89C51
3. ทำการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีเบื้องต้นของไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ (Microcontroller) เป็นชื่อของอุปกรณ์อีเล็กทรอนิกส์แบบหนึ่งรวมเอาหน่วยประมวลผล หน่วยคำนวณทางคณิตศาสตร์และลอจิก วงจรรับสัญญาณอินพุต วงจรขับสัญญาณเอาต์พุต หน่วยคำนวณ วงจรกำเนิดสัญญาณนาฬิกาไว้ด้วยกัน ทำให้สามารถนำไปใช้งานแทนวงจรอีเล็กทรอนิกส์ที่ซับซ้อน ได้เป็นอย่างดีช่วยลดจำนวนอุปกรณ์และขนาดของระบบในขณะที่มีขีดความสามารถสูงขึ้น ภายใต้งบประมาณที่เหมาะสม

ไมโครคอนโทรลเลอร์มาจากคำ 2 คำรวมกันก็คือ "ไมโคร" (Micro) ซึ่งหมายถึง ไมโครโปรเซสเซอร์ (Microprocessor) ซึ่งเป็นอุปกรณ์ประมวลผลข้อมูลขนาดเล็ก ภายในประกอบด้วย หน่วยประมวลผลกลางหรือซีพียู (CPU : Central Processing Unit) หน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU: Arithmatic Logic Unit) วงจรเชื่อมต่อหน่วยความจำ และวงจรสัญญาณนาฬิกา อีกคำหนึ่งคือคำว่า "คอนโทรลเลอร์" (Controller) หมายถึง อุปกรณ์ควบคุม ดังนั้นไมโครคอนโทรลเลอร์จึงเป็นอุปกรณ์ที่ใช้ในการควบคุม โดยที่สามารถเขียนโปรแกรมเพื่อกำหนดรูปแบบการควบคุมได้อย่างอิสระ

2.1 คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51

- เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียู ขนาด 8 บิต
- ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลชลบ และเขียนใหม่ได้พันครั้ง
- หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรมในบางเบอร์ จะมีหน่วยความจำแบบอีอีพรอมเพิ่มเติม
- ขาพอร์ตเป็นแบบสองทิศทางสามารถใช้งานได้ทั้งอินพุตและเอาต์พุต
- มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
- ไทเมอร์/เคาน์เตอร์ ขนาด 16 บิตอย่างน้อย 2 ตัว
- สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
- สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
- มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ในภายในชิป
- มีวงจรสื่อสารอนุกรมแบบ SPI สำหรับในอนุกรม AT89Sxx
- มีวอตช์ด็อกไทเมอร์ในตัว สำหรับในอนุกรม AT89Sxx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 การจัดการขาของไมโครคอนโทรลเลอร์ MCS-51

ไมโครคอนโทรเลอร์ MCS-51 ทุกเบอร์จะมีสถาปัตยกรรมและขาใช้งานพื้นฐานเหมือนกัน โดยมีรายละเอียดขั้นต้น ดังนี้

- ขา VCC ใช้สำหรับต่อไฟเลี้ยง + 5V
- ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

ขาพอร์ต 0 (P0.0 – P0.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุตและเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาตพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุตสามารถทำได้ โดยการเขียนข้อมูล " 1 " ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย(Float) จึงมี อินพุต อิมพีแนดซ์ สูงสามารถใช้งานเป็นขาพอร์ตอินพุตได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งาน ในการติดต่อกับขาของแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7X) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วย เพื่อสลับการทำงานเป็นได้ทั้งขาติดต่อแอดเดรสและขาข้อมูล

2.3 โครงสร้างและการทำงานของพอร์ต

ไมโครคอนโทรเลอร์ MCS – 51 แบบแฟลชมีพอร์ตให้ใช้งานทั้งสิ้น 4 พอร์ต คือ พอร์ต 0 ถึงพอร์ต 3 และพอร์ตมีขนาด 8 บิต เป็นพอร์ตแบบ 2 ทิศทาง กล่าวคือ สามารถเป็นได้ทั้งอินพุตสำหรับรับสัญญาณข้อมูลและเอาต์พุตสำหรับส่งสัญญาณข้อมูลออก ทุกพอร์ตของตัว MCS-51 แบบแฟลชมีวงแลตช์และวงจรขับตลอดจนบัฟเฟอร์อินพุตที่พอร์ต 0 และพอร์ต 2 จะใช้งานเป็นพอร์ตอินพุตและเอาต์พุตสำหรับงานทั่วไปและใช้ในการติดต่อหน่วยความจำภายนอก สำหรับพอร์ต 3 ทั้งพอร์ตและพอร์ต 1 บางขา นอกจากจะใช้เป็นขาพอร์ตอินพุตเอาต์พุตตามปกติแล้ว ยังสามารถใช้งานในหน้าที่พิเศษได้อีก ขึ้นอยู่กับว่าเป็นไมโครคอนโทรลเลอร์MCS-51แบบ

แฟลชแบบใดแสดงวงจรภายในแต่ละพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชโดยรูปภายที่ 7 เป็นวงจรของพอร์ต 0 วงจรแลตช์ของแต่ละบิตในแต่ละพอร์ตก็คือวงจรดีฟลิฟลอปนั่นเองการอ่านค่าสถานะของพอร์ตและสถานะของวงจรแลตช์สามารถกระทำได้อย่างอิสระด้วยสัญญาณที่แยกจากกัน นั่นคือสัญญาณอ่านข้อมูลจากขาพอร์ต และสัญญาณอ่านข้อมูลของจากวงจรแลตช์ ส่วนการเขียนข้อมูลมายังพอร์ตต้องส่งสัญญาณมายังขา CLK ของดีฟลิปฟลอป ในขณะที่ข้อมูลจะผ่านมาทางขาบัสข้อมูลภายในเข้าสู่ขา D ของดีฟลิปฟลอปที่พอร์ตนี้มีวงจรมัลติเพล็กซ์สำหรับกำหนดลักษณะการทำงานของพอร์ตว่าต้องการใช้งานเป็นขาพอร์ตอินพุต เอาต์พุตปกติหรือใช้ในการติดต่อกับ หน่วยความจำภายนอกของตัวไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 จังหวะการทำงานของไมโครคอนโทรลเลอร์

ในการใช้งานไมโครคอนโทรลอร์ MCS-51 จะต้องทำความเข้าใจถึงจังหวะการทำงานของซีพียูและลำดับขั้นตอนการประมวลผลคำสั่ง ในการประมวลผลคำสั่งของซีพียูจะมีขั้นตอนหลักๆ 2 ขั้นตอนคือกระบวนการเฟตช์ (fetch) เป็นการเรียกคำสั่งออกจากหน่วยความจำโปรแกรมแล้วทำการแปลงรหัสคำสั่งนั้นเป็นภาษาเครื่องเพื่อเตรียมการประมวลผลขั้นตอนต่อมา ซึ่งนั่นก็คือ กระบวนการเอ็กซิคิวต์ (execute) เป็นการกระทำตามคำสั่งที่กำหนดหรือตามที่เฟตช์ขึ้นมาโดยกระบวนการก่อนหน้านี้ เมื่อทำการเอ็กซิคิวต์คำสั่งเรียบร้อยแล้วก็จะไปเริ่มกระบวนการเฟตช์คำสั่งใหม่ต่อไป

เมื่อเริ่มจ่ายไฟให้แก่ไมโครคอนโทรลเลอร์จะเกิดการรีเซตเกิดขึ้น ในลักษณะที่เรียกว่าเพาเวอร์ออนรีเซต (Power On Reset) ซีพียูเริ่มต้นการทำงานที่แอดเดรส 0000H ของหน่วยความจำโปรแกรม จังหวะการทำงานของซีพียูจะเป็นไปตามรูปแบบ โดยได้รับการกำหนดมาจากรอบการทำงานหรือแมชชีนไซเกิล (Machine Cycle) ในรูปภาพที่ 9 เป็น ไดอะแกรมเวลาแสดงจังหวะการทำงานของไมโครคอนโทรลเลอร์ NCS-51 โดยใน 1 รอบการทำงานหรือในหนึ่งแมชชีนไซเกิลจะแบ่งย่อยออกเป็น 6 สเตต (State) กำหนดชื่อเป็น S1-S6 ในแต่ละสเตตมีค่าเวลาเท่ากับ 2 คาบเวลาของสัญญาณนาฬิกา ถ้าสัญญาณนาฬิกามีความถี่ 12 MHz จะมีคาบเวลาเท่ากับ 1 ms คาบเวลาทั้งสองภายในหนึ่งสเตตจะเรียกว่า เฟส 1

2.5 การจัดการหน่วยความจำ

ในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชมีหน่วยความจำภายในหลัก ๆ อยู่ 2 ส่วนคือหน่วยความจำโปรแกรมและหน่วยความจำข้อมูล ซึ่งก็มีขนาดและการจัดสรรแตกต่างกันไปในแต่ละเบอร์ ในบทนี้จะกล่าวถึงรายละเอียดของการจัดสรรหน่วยความจำภายในตัวของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช การเชื่อมต่อกับหน่วยความจำ ภายนอกและข้อมูลเบื้องต้นของรีจิสเตอร์ฟังก์ชั่นพิเศษที่ใช้ควบคุมการทำงานของ ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช

2.5.1 หน่วยความจำโปรแกรม (Porgram Memory)

หน่วยความจำของโปรแกรมที่ใช้ในการเก็บข้อมูลของโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์หรือที่เรียกว่า โปรแกรมมอนิเตอร์ (Monitor Porgram) ถ้าหากใช้หน่วยความจำภายนอกมักจะบรรจุอยู่ในหน่วยความจำชนิดอีพรอม (EPROM : Erasable Programmable Read-only Memory) ซึ่งสามารถทำการอ่านได้เพียงอย่างเดียว

กรณีที่ใช้ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชที่มีหน่วยความจำโปรแกรมภายในแต่ต้องการติดต่อกับหน่วยความจำโปรแกรมภายนอกด้วย สามารถทำได้โดยต้องกำหนดแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของหน่วยความจำโปรแกรมให้ต่อจากแอดเดรสสุดท้ายของหน่วยความจำ โปรแกรมภายในของไมโครคอนโทรลเลอร์ ยกตัวอย่าง ไมโครคอนโทรลเลอร์ AT89C51 มีหน่วยความจำโปรแกรมขนาด 4 กิโลไบต์ มีแอดเดรสอยู่ระหว่าง 0000H – 0FFFH เมื่อต่อหน่วยความจำโปรแกรมภายนอกต้องกำหนดให้แอดเดรสอยู่ในช่วง 1000H-FFFFH

การต่อหน่วยความจำภายนอกแสดงดังในรูปภาพที่ 12 ขาพอร์ต P0.0-P0.7 ใช้เป็นขาข้อมูล D0-D7 และขาแอดเดรสไบต์ต่ำ โดยผ่านวงจรแลตช์ ซึ่งปกติใช้ไอซีเบอร์ 74hc573 และใช้สัญญาณ ALE และ PSEN ในการเลือกใช้งานขา P0.0-P0.7 เพื่อเป็นขาข้อมูลหรือขาแอดเดรสไบต์สูง A8-A15 ดังนั้นเมื่อมีการติดต่อกับหน่วยความจำโปรแกรมจากภายนอกไมโครคอนโทรลเลอร์ จะเหลือขาพอร์ตเพียง 16 บิต คือ ที่ขาพอร์ต P1.-P1.7 และ P3.0

2.5.2 หน่วยความจำข้อมูล (Data Memory)

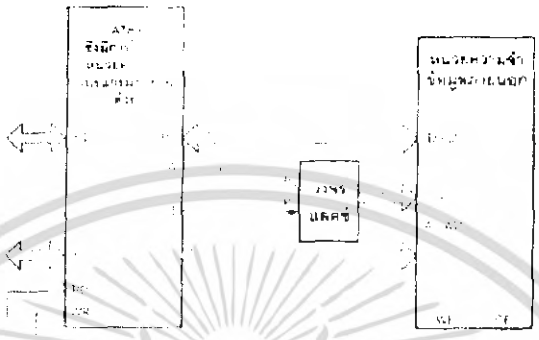
มีด้วยกัน 2 แบบคือ หน่วยความจำข้อมูลภายนอกและหน่วยความจำข้อมูลภายใน โดยไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89 สามารถติดต่อกับหน่วยความจำข้อมูลภายนอกได้สูงสุด 64 กิโลไบต์ โดยการใช้คำสั่ง MOVX ในการติดต่อกับหน่วยความจำข้อมูลภายนอก การติดต่อกับหน่วยความจำข้อมูลภายนอกของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชแสดงดังในรูปภาพที่ 10 จะเห็นได้ว่า มีลักษณะคล้ายกับการติดต่อกับหน่วยความจำโปรแกรมภายนอก แตกต่างกันที่มีสัญญาณที่ใช้สำหรับการอ่านและเขียนหน่วยความจำข้อมูลภายนอก นั่นคือ ขา RD และ WR

สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชในอนุกรม AT89 ทุกเบอร์จะมีหน่วยความจำข้อมูลภายในเป็นแบบแรม (*RAM: Random Access Memory) โดยแต่ละเบอร์จะมีขนาดแตกต่างกันไป ในเบอร์ AT89C51 มีหน่วยความจำข้อมูลภายในขนาด 128 ไบต์ ในขณะที่เบอร์ AT89C52 มีขนาด 256 ไบต์ สำหรับการจัดสรรหน่วยความจำข้อมูลภายในแบ่งเป็น 3 ส่วนคือ หน่วยความจำข้อมูลส่วนล่าง (Lower)และส่วนบน (Upper) และรีจิสเตอร์ฟังก์ชั่นพิเศษ (SFR:Special Function Register) แต่ละส่วนมีขนาด 128 ไบต์

ขนาดของหน่วยความจำข้อมูลของไมโครโทรลเลอร์ MCS-51 แบบแฟลชโดยแท้จริงและมีเพียง 256 ไบต์ แต่ด้วยการจัดการเข้าถึงที่แตกต่างกัน จึงดูเสมือนว่า ไมโครโทรลเลอร์ MCS-51 แบบแฟลชมีหน่วยความจำข้อมูลภายในสูงถึง 384 ไบต์ โดยในหน่วยความจำข้อมูลส่วนล่างขนาด 128 ไบต์ มีแอดเดรสอยู่ที่ 00H-7FH สามารถเข้าถึงได้โดยตรงและโดยอ้อมสำหรับหน่วยความจำข้อมูลส่วนบนมีขนาด 128 ไบต์เช่นกัน มีแอดเดรสอยู่ที่ 80H-FFH สามารถเข้าถึงแบบโดยอ้อมเท่านั้น ในขณะที่รีจิสเตอร์ SFR มีแอดเดรสอยู่ที่ 80H-FFH เช่นเดียวกับหน่วยความจำข้อมูลส่วนบน แต่สำหรับรีจิสเตอร์ SFR ใช้การเข้าถึงแบบโดยตรง ดังนั้นเพื่อความสะดวกและง่ายตลอดจนป้องกันความสับสนในการเขียนโปรแกรมสำหรับผู้เริ่มต้น จึงควรใช้หน่วยความจำข้อมูล

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้ในการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายในเพียง 128 ไบต์ จากหน่วยความจำข้อมูลส่วนล่างร่วมกับรีจิสเตอร์ SFRหน่วยความจำข้อมูล 16 ไบต์ถัดมาที่แอดเดรส 20H-2FH เป็นพื้นที่สำหรับใช้งานทั่วไป สามารถเข้าถึงได้ในระดับบิต (Stack:ที่พักข้อมูลชั่วคราวในกรณีที่ซีพียูมีการกระทำโดยไปทำงานในโปรแกรมย่อย) การเข้าถึงหน่วยความจำในส่วนนี้ต้องใช้การเข้าถึงระดับไบต์



ภาพที่ 2.1 โครงสร้างหน่วยความจำ

2.5.3 รีจิสเตอร์ฟังก์ชั่นพิเศษ (Special Function Register:SFR)

เป็นรีจิสเตอร์ที่ใช้ควบคุมการทำงานของไมโครคอนโทรลเลอร์ MCS-51 มีด้วยกัน 22 ตัว สำหรับเบอร์ AT89C51 และ 28 ตัวในเบอร์ AT89C52 และ อนุกรม TA89Sxx ทั้งนี้เนื่องจากใน AT89C52และAT89Sxxมีจำนวนไทเมอร์เคาน์เตอร์มากกว่าAT89C51

รีจิสเตอร์ SFR มีแอดเดรสอยู่ระหว่าง 80H-FFH ในพื้นที่ของหน่วยความจำข้อมูลส่วนบน สามารถเข้าถึงได้โดยตรง (Dircet Addressing) ในรูปภาพที่ 17 ดังแสดงการจัดสรรพื้นที่ของรีจิสเตอร์ SFR แต่ละตัวในหน่วยความจำข้อมูลส่วนบน

2.5.4 รีจิสเตอร์แสดงสถานะของโปรแกรม(Program Status Word:PSW)

เป็นรีจิสเตอร์ขนาด 8 บิต สามารถเข้าถึงได้ในระดับบิต จึงสามารถกำหนดค่าในแต่ละบิตของ รีจิสเตอร์ตัวนี้ได้อย่างอิสระ มีแอดเดรสอยู่ที่ D0H ทำหน้าที่เก็บสถานะของการทำงานของโปรแกรมในขณะนั้นจะเรียกสถานะต่าง ๆ ของโปรแกรม แฟลก (Flag) เมื่อซีพียูกระทำคำสั่งทางคณิตศาสตร์และลอจิกแล้วเกิดการเปลี่ยนแปลงสถานะขึ้น ผลของการเปลี่ยนแปลงนั้นจะมาปรากฏที่บิตต่าง ๆ ของรีจิสเตอร์ PSW รายละเอียดของแต่ละบิตในรีจิสเตอร์ PSW

2.5.5 แอกคิวมูเลเตอร์ (Accumulator:ACC)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีขนาด 8 บิต มีแอดเดรสอยู่ที่ตำแหน่ง E0H เป็นรีจิสเตอร์ที่ใช้สำหรับเก็บข้อมูลหรือผลลัพธ์ที่ได้จากการทำงานของไมโครคอนโทรลเลอร์และ โดยเฉพาะอย่างยิ่งในการคำนวณทางคณิตศาสตร์และลอจิก ก่อนที่จะส่งข้อมูลหรือผลลัพธ์ที่ได้ให้แก่ซีพียูเพื่อทำการประมวลผลต่อไป อาจเรียกสั้น ๆ ว่า รีจิสเตอร์ A หรือ ACC รีจิสเตอร์นี้สามารถเข้าถึงระดับบิตได้

2.5.6 รีจิสเตอร์ B

มีขนาด 8 บิต มีแอดเดรสอยู่ที่ F0H มีหน้าที่พิเศษ คือ หากต้องการคูณหรือหารทางคณิตศาสตร์ ต้องนำข้อมูลที่ต้องการหารหรือคูณมาเก็บไว้ในรีจิสเตอร์ B แล้วจึงกระทำคำสั่งการคูณหรือหารกับค่าในรีจิสเตอร์ A ต่อไป

2.5.7 โปรแกรมเคาน์เตอร์ (Program Counter : PC)

มีขนาด 16 บิต มีหน้าที่แจ้งแอดเดรสของหน่วยความจำโปรแกรมในตำแหน่งถัดไปที่ซีพียูจะต้องไปทำงาน รีจิสเตอร์ PC เป็นรีจิสเตอร์ตัวเดียวที่ไม่ได้จัดสรรไว้รวมกับรีจิสเตอร์ SFR ตัวอื่น ๆ การเปลี่ยนแปลงค่าของรีจิสเตอร์ PC จะขึ้นอยู่กับผลของการกระทำคำสั่งภายในหน่วยความจำโปรแกรมที่ผู้เขียนโปรแกรมกำหนด

ตารางที่ 2.1 การเลือกแบงก์ของหน่วยความจำส่วนล่างเพื่อติดต่อกับรีจิสเตอร์แบงก์R0-R7

RS1	RS0	แบงก์ของรีจิสเตอร์	ช่วงแอดเดรส
0	0	แบงก์ 0	00H-07H
0	1	แบงก์ 1	08H-0FH
1	0	แบงก์ 2	10H-17H
1	1	แบงก์ 3	18H-1FH

2.5.8 สแต็กพอยน์เตอร์ (Stack Pointer : SP)

รีจิสเตอร์ตัวชีสแต็ก มีขนาด 8 บิต มีแอดเดรสอยู่ที่ 81 ใช้ในการเก็บค่าตำแหน่งของตัวชีสแต็ก ซึ่งสามารถเปลี่ยนแปลงได้เมื่อซีพียูมีการกระโดยไปทำงานที่โปรแกรมย่อย หรือกระโดดจากโปรแกรมย่อมแล้วกลับมายังโปรแกรมหลัก เมื่อมีการีเซตเกิดขึ้น (รีเซต : การกระทำที่ส่งผลให้ซีพียูต้องเริ่มต้นการทำงานใหม่ตั้งแต่ต้น) ค่าของรีจิสเตอร์ SP จะเท่ากับ 07H ดังนั้นแอดเดรสแรกของพื้นที่ที่สำรองไว้ทำหน้าที่เป็นสแต็กจะเท่ากับ 08H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.9 รีจิสเตอร์ชี้ข้อมูลหรือดาต้าพอยน์เตอร์ (DataPointer:DPTR)

มีขนาด 16 บิตโดยแบ่งเป็นรีจิสเตอร์ชี้ข้อมูลไบต์สูง (DPH) และรีจิสเตอร์ชี้ข้อมูลไบต์ต่ำ (DPL) แต่ละตัวมีขนาด 8 บิต มีแอดเดรสอยู่ที่ 82H , สำหรับ DPL และ 83H สำหรับ DPH รีจิสเตอร์ DPTR นี้ใช้ในการเก็บค่าแอดเดรสของหน่วยความจำหรืออุปกรณ์ภายนอกที่ไมโครคอนโทรลเลอร์ต้องการติดต่อด้วย

2.5.10 รีจิสเตอร์พอร์ต (Port Register)

เป็นรีจิสเตอร์ขนาด 8 บิต ที่ใช้เก็บข้อมูลของแต่ละพอร์ตของไมโครคอนโทรลเลอร์ MCS-51 มี 4 ตัวคือ รีจิสเตอร์พอร์ต 0 หรือ P0 มีแอดเดรสอยู่ที่ 80H , รีจิสเตอร์พอร์ต 1 หรือ P1 มีแอดเดรสอยู่ที่ 90H, รีจิสเตอร์พอร์ต 2 หรือ P2 ซึ่งมีแอดเดรสอยู่ที่ A0H และรีจิสเตอร์พอร์ต3 หรือ P3 มีแอดเดรสอยู่ที่ B0H รีจิสเตอร์ทุกตัวสามารถเข้าถึงได้ในระดับบิต เมื่อต้องการอ่านหรือเขียนข้อมูลออกไปยังพอร์ตของไมโครคอนโทรลเลอร์ ซึ่งจะต้องมีการกระทำผ่านรีจิสเตอร์นี้ทุกครั้ง

2.5.11 รีจิสเตอร์บัฟเฟอร์ข้อมูลอนุกรม (Serial Data Buffer : SBUF)

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 99H ใช้ในการเก็บข้อมูลที่ส่งออกหรือรับเข้าของวงจรสื่อสารอนุกรมที่มีอยู่ไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช โดยภายในรีจิสเตอร์ SBUF นี้จะแบ่งออกเป็น 2 ส่วนคือรีจิสเตอร์ส่งข้อมูล (Transmit Buffer Register)และรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูล (Receive Buffer Register) เมื่อมีการเขียนข้อมูล มายังรีจิสเตอร์ SBUF ข้อมูลนั้นจะถูกส่งต่อไปยังบัฟเฟอร์สำหรับส่งข้อมูลเพื่อส่งออกจากเอาท์พุตไมโครคอนโทรลเลอร์ผ่านทางขา TxD หรือขา P3.1 ในกรณีที่การอ่านข้อมูลจาก รีจิสเตอร์SBUFข้อมูลจะถูกส่งผ่านไปยังรีจิสเตอร์บัฟเฟอร์สำหรับรับข้อมูลเพื่อส่งไปยัง ไมโครคอนโทรลเลอร์ต่อไปสำหรับการรับข้อมูลอนุกรมจากภายนอกนั้นจะผ่านมาทางขา RxD หรือ P3.0 ทางอินพุตของตัวไมโครคอนโทรลเลอร์ MCS-51 สำหรับรายละเอียดของรีจิสเตอร์ SBUF และที่เป็นวงจรสื่อสารอนุกรมภายในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชจะว่าด้วยเรื่องการสื่อสารผ่านพอร์ตอนุกรม

2.5.12 รีจิสเตอร์ไทเมอร์ (Timer Register)

เป็นรีจิสเตอร์ขนาด 16 บิต แบ่งเป็นไบต์สูงและไบต์ต่ำสุดเช่นเดียวกับรีจิสเตอร์ DPTR รีจิสเตอร์ไทเมอร์ใช้ในการเก็บค่าของตัวนับหรือเคาน์เตอร์ (Counter) ภายในตัวของไมโครคอนโทรลเลอร์ เพื่อใช้ในการสร้างฐานเวลา,จับเวลา หรือนับจำนวนพัลส์สัญญาณนาฬิกาภายใน บางทีเรียกว่ารีจิสเตอร์ตัวนี้ว่ารีจิสเตอร์ไทเมอร์/เคาน์เตอร์ ในไมโครคอนโทรลเลอร์ เบอร์ AT89C51 มีรีจิสเตอร์ไทเมอร์/เคาน์เตอร์ 2 ตัวแบ่งเป็น T0 หรือ Timer 0 และ T1 หรือ Tmier 1 ในรีจิสเตอร์ยังแบ่งเป็นรีจิสเตอร์ไทเมอร์ไบต์ต่ำ (TL) และรีจิสเตอร์ไทเมอร์ไบต์สูง (TH) เหมือนกัน โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ TLO มีแอดเดรสอยู่ที่ 8AH รีจิสเตอร์ TH0 มีแอดเดรสอยู่ที่ 8BH ในขณะที่ TL1 และ TH1 มีแอดเดรสอยู่ที่ 8CH และ 8DH สำหรับในเบอร์ AT89C52 และในอนุกรม AT89Sxx จะมีรีจิสเตอร์ไทเมอร์ / เคาน์เตอร์ถึง 3 ตัว โดยมีรีจิสเตอร์ TL2 และ TH2 ซึ่งมีแอดเดรสอยู่ที่ 0CCH และ 0CDH เพิ่มเติม

2.5.13 รีจิสเตอร์แคปเจอร์ (Capture Register)

เป็นรีจิสเตอร์ขนาด 16 บิต มีเฉพาะไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชเบอร์ 89C52 และในอนุกรม AT89Sxx เท่านั้น เนื่องจากต้องใช้ร่วมกับไทเมอร์ /เตอร์ 2(Timer 2) โดยรีจีสเตอร์แคปเจอร์นี้มีชื่อเรียกอย่างย่อว่า รีจิสเตอร์ RCAP2 ซึ่งสามารถแบ่งออกเป็นไบต์ต่ำ คือ RCAP2L มีแอดเดรสอยู่ที่0CAHและไบต์สูงคือRCAP2H มีแอดเดรสอยู่ที่0CBH

รีจิสเตอร์แคปเจอร์จะถูกใช้งานเมื่องกำหนดให้ไทเมอร์ 2 ซึ่งจะให้ทำงานในโหมดแคปเจอร์ ซึ่งเป็นโหมดที่กำหนดให้ไมโครคอนโทรลเลอร์ทำการตรวจจับการเปลี่ยนแปลงสถานะทางลอจิกที่ขา T2EX ทั้งนี้เพื่อใช้ประโยชน์ในการวัดคาบเวลา ความถี่ และการเปลี่ยนแปลงของสัญญาณพัลส์ที่ขา T2EX

2.5.14 รีจิสเตอร์ควบคุม (Control Register)

รีจิสเตอร์ PCON เป็นรีจิสเตอร์ที่เกี่ยวข้องกับการกำหนดอัตราการรับส่งข้อมูลของวงจรสื่อสารอนุกรมและกำหนดการทำงานในโหมดประหยัดพลังงานของไมคอนโทรลเลอร์ MCS- 51 แบบแฟลชรีจิสเตอร์ SCON เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานของวงจร สื่อสารอนุกรมภายในไมโครคอนโทรลเลอร์MCS-51แบบแฟลชรีจิสเตอร์ TCON และ T2CON เป็นรีจิสเตอร์ที่ใช้ในการควบคุมการทำงานของตัวไทเมอร์ / เคาน์เตอร์ภายในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช โดย T2CON ใช้สำหรับไทเมอร์ / เคาน์เตอร์ 2 ของไมโครคอนโทรลเลอร์ MCS- 51 แบบแฟลชเบอร์AT89C52และในอนุกรมAT89Sxx

รีจิสเตอร์ TNOD และ T2MOD เป็นรีจิสเตอร์ที่ใช้กำหนดโหมดหรือลักษณะในการทำงานของไทเมอร์/เคาน์เตอร์ภายในไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชโดย T2MOD ใช้สำหรับไทเมอร์/เคาน์เตอร์ 2 ของไมโครคอนโทรลเลอร์ MCS- 51 แบบแฟลชเบอร์ AT89C52 และในอนุกรมAT89Sxx

รีจิสเตอร์ IE และ IP เป็นรีจิสเตอร์ที่เกี่ยวข้องกับการตอบสนองการอินเตอร์รัปต์ (Interrupt : การขัดจังหวะการทำงานปกติของซีพียู)โดย IE เป็นรีจิสเตอร์สำหรับเอ็นเบิลหรือใช้ในการกำหนดลักษณะของการตอบสนองการอินเตอร์รัปต์ ในขณะที่ IP เป็น รีจิสเตอร์สำหรับ กำหนดลำดับความสำคัญของการตอบสนองการอินเตอร์รัปต์ว่า จะให้ซีพียูตอบสนองการเกิดอินเตอร์รัปต์ในลักษณะใดก่อนหรือหลัง

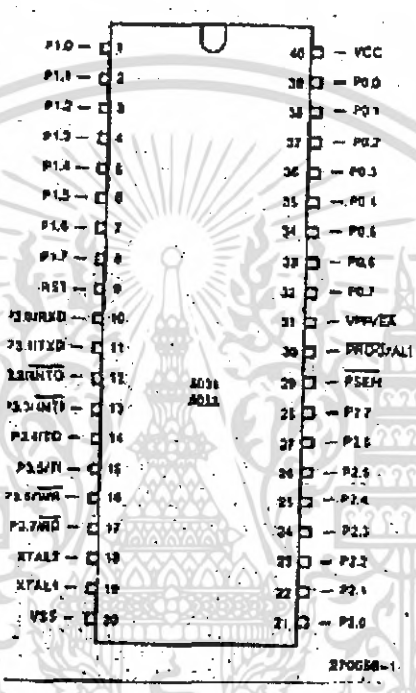
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่แต่ละขาของไอซีไมโครคอนโทรลเลอร์

1. Vcc เป็นขาที่ใช้สำหรับต่อไฟเพื่อเลี้ยงไอซี +5VDC

2. GND เป็นขากราวด์สำหรับต่อกับกราวด์ของระบบ

3. พอร์ต 1 (P1.0-P1.7), พอร์ต 2 (P2.0 – P2.7 0) มีจำนวน 8 ขาแต่ละขา เรียกได้เป็น 1 บิต สามารถที่จะกำหนดให้เป็นได้ทั้งพอร์ตอินพุตและ พอร์ตเอาต์พุตสำหรับใช้งานทั่วไป ถ้าต้องการให้ขาพอร์ตใดเป็นอินพุต ก็สามารถทำได้โดยการเขียนข้อมูล ลอจิก "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อ



ภาพที่ 2.2 IC8051

4. พอร์ต 3 (P3.0-P3.7) มีจำนวน 7 ขา แต่ละขาเรียกได้เป็น 1 บิต แต่ในส่วน ของวงจรภายในไอซีจะมีขาของ พอร์ต 3 อยู่ทั้งหมด 8 ขา เพียงแต่ขา P3.6 จะไม่ได้ต่อออกมาใช้งานภายนอกของตัวไอซี แต่ใช้เป็นขารับสถานะ ของผลการเปรียบเทียบสัญญาณ Analog Comparator Input ระหว่างพอร์ต PI.0 และ P1.1 จากภายนอก ดังนั้นขาทั้ง 7 ขาที่ต่อใช้งานภายนอกของไอซี สามารถที่จะกำหนดให้เป็นได้ทั้งพอร์ตอินพุต และพอร์ตเอาต์พุต สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุต สามารถที่จะทำได้โดยการเขียนข้อมูลให้เป็น ลอจิก "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการจะติดต่อด้วย นอกจากนั้นขาของ พอร์ต 3 จะยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ซึ่งจะมีรายละเอียดดังต่อไปนี้

4.1 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม

4.2 ใช้เป็นขาอินพุตสำหรับส่งข้อมูลจากการสื่อสารแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัพท์จากภายนอกช่อง 0

4.4 ใช้เป็นขาอินพุตรับสัญญาณอินเตอร์รัพท์จากภายนอกช่อง1

4.5 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 0

2.6 การเขียนโปรแกรมแอสเซมบลี

โครงสร้างของโปรแกรมแอสเซมบลี

โปรแกรมจะถูกเก็บไว้ในหน่วยความจำ ซึ่งอยู่ในรูปแบบของเลขไบนารี่ที่เรียกว่าภาษา Machine ซึ่งเป็นภาษาที่สามารถ ติดต่อให้คอมพิวเตอร์ เข้าใจได้ ภาษา Machine นี้จะจัดให้อยู่ในรูปแบบที่เป็นเลขฐานสิบหก (HEX) เช่น คำสั่ง 8 บิต 11101011B (B-ไบนารี่) เขียนได้เป็น 0EBH (H-ฐานสิบหก) แต่ก็เป็นการที่จะเข้าใจความหมายได้ยากในการใช้งาน การที่จะทำความเข้าใจภาษา Machine จะมีการใช้สัญลักษณ์ (Symbols) ที่เรียกว่า Mnemonicsเพื่อแทนความหมายของคำสั่ง เช่น MOV A ,#67H (หมายความว่านำข้อมูลค่าคงที่ 67H ไปเก็บไว้ใน reg. A) โปรแกรมที่เขียนด้วยรหัส Mnemonics เรียกว่า ภาษา Assembly และก่อนที่จะให้ CPU ทำงาน ตามโปรแกรมที่เขียนด้วยภาษา Assembly ได้ แต่ต้องเปลี่ยนให้เป็นภาษา Machine ก่อน โดยใช้ โปรแกรมแอสเซมเบลอร์

Statement

รูปแบบโปรแกรมภาษา Assembly ประกอบด้วย ส่วนต่างๆ (Field) 4 ส่วนคือ Label, Op-code, Operand และ Comment ในการเขียนในแต่ละคำสั่ง เรียกว่า Statement เช่น

Label

เลเบลจะต้องขึ้นต้นด้วยตัวอักษร "A-Z" หรือเครื่องหมาย " _ ", " . " จะเป็น ตัวอักษรจาก A ถึง Z หรือตัวเลขจาก 0 ถึง 9 หรือเครื่องหมาย " ? " เลเบลจะยาวกี่ตัวอักษรก็ได้ แต่เมื่อรวมทั้งบรรทัดแล้วต้องไม่เกิน 255 ตัวอักษร ในการเขียนเลเบลจะต้องมีเครื่องหมาย " : " (Colon)ตามหลังเลเบลด้วยทุกครั้ง เช่น LOOP

แนะนำ :: การใช้เลเบลควรจะให้สื่อต่อตัวโปรแกรมที่จะเรียกใช้ เช่น ชื่อและหน้าที่ยกตัวอย่าง โปรแกรมย่อยหน่วงเวลา ก็ควรตั้งชื่อเป็น DELAY:

Op-code เป็นส่วนของนีโมนิคที่จะแสดงจุดมุ่งหมายของการกระทำแต่ละคำสั่ง เช่น MOV ADD SUB ในช่องห่างระหว่าง Op-Code และโอเปอร์แรนด์จะต้องมีช่องว่างคั่นอยู่อย่างน้อย 1 ช่อง เช่น MOV A,R1

Operand โอเปอร์แรนด์เป็นส่วนที่แสดงถึงส่วนที่ถูกกระทำ อาจประกอบด้วย2 ส่วนที่มีการคั่นด้วย " , "หรือการแสดงถึงตำแหน่งที่จะกระทำ

Comment เป็นการเขียนคำอธิบายแต่ละบรรทัดหรือหมายเหตุนั้นและที่สำคัญจะต้องใช้เครื่องหมาย " ; " (Semi-Colon) นำหน้าด้วยทุกครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flowchart โฟว์ชาร์ต โฟว์ชาร์ตใช้แสดงลักษณะและอัลกอริทึมของโปรแกรมในรูปแบบผังกราฟที่เหมาะสม เป็นการลำดับงานแบบเป็นขั้นตอน

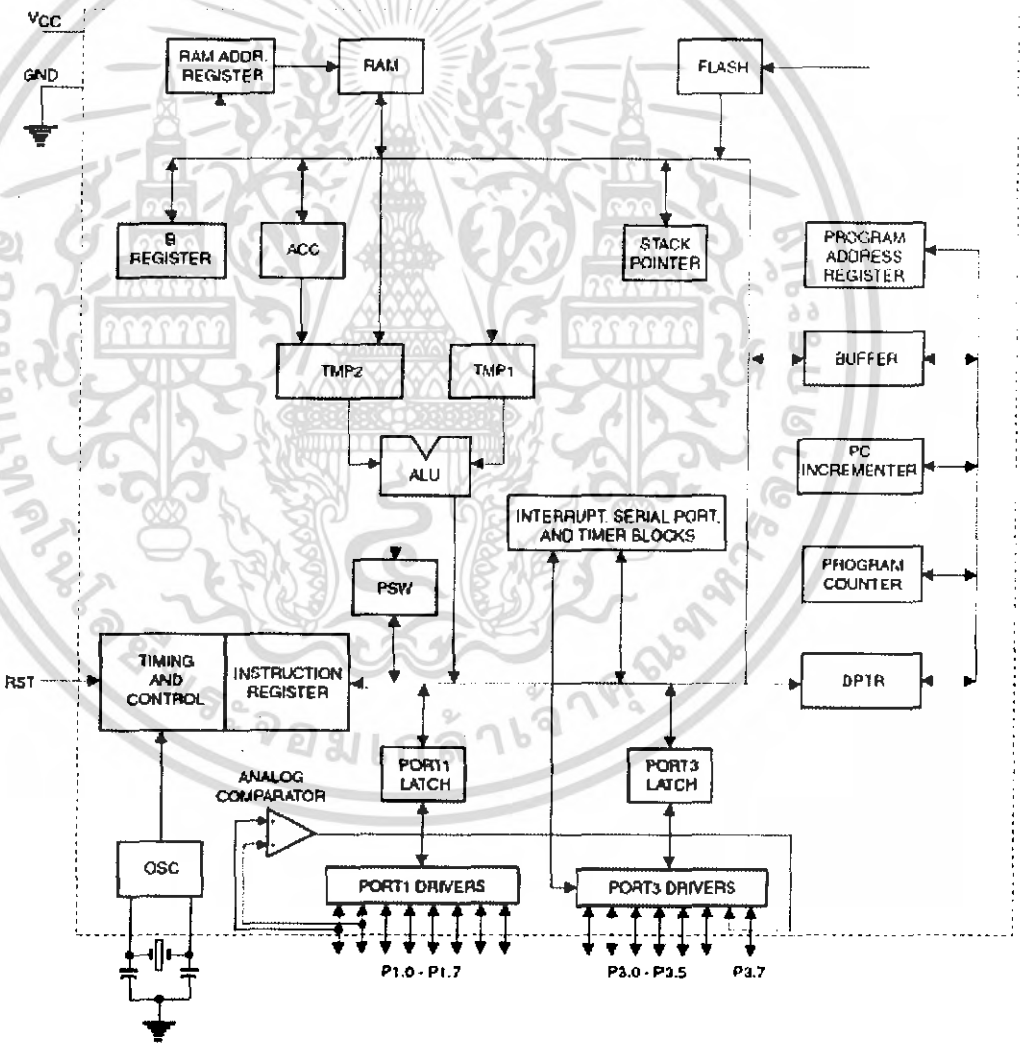
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

ทฤษฎีของไมโครคอนโทรลเลอร์ 89C51

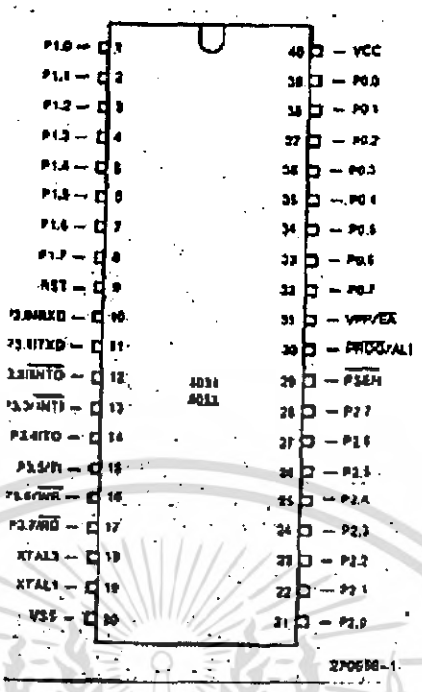
3.1 สถาปัตยกรรมของ 89C51

89C51 ใช้สถาปัตยกรรมเดียวกันกับ 8051 แต่ภายในของ 89C51 จะเปลี่ยนแปลงจากบล็อกที่เป็น ROM ของ 8051 เป็นบล็อก PE ROM ซึ่งมีขนาด 8 Kbyte ฉะนั้นเราจึงขอกล่าวว่า 89C51 เป็นตระกูลเดียวกันกับ 8951 จากรูปที่ 3.1 จะเป็นสถาปัตยกรรมภายในของ 89C51 ซึ่งจะอธิบายถึงส่วนย่อยๆ ของภายใน89C51 เพียงชิพเดียว และสัญญาณจากภายในจะต่อออกสู่ภายนอกทางขา (PIN) ของ89C51 ที่มีอยู่ 40 ขา



รูปที่ 3.1 สถาปัตยกรรมภายในของ 89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 ไดอะแกรมขาของ 89C51 แบบ DIP

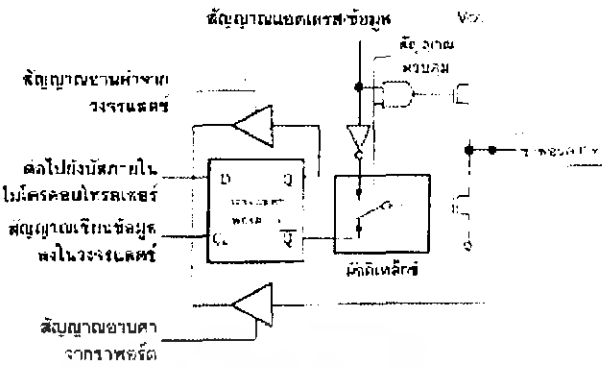
89C51 เป็นไมโครคอนโทรลเลอร์ที่บรรจุอยู่ในวงจรรวมแบบ Dual Inline Package (DIP) ซึ่งแต่ละข้างของ 89C51 มีขาอยู่ข้างละ 20 ขา รวมกันทั้งหมด 2 ข้างได้ 40 ขา มีการใช้งานต่างๆกันดังต่อไปนี้

VCC (ที่ขา 40)เป็นขาที่ต้องใช้ไฟเลี้ยง +5 V เข้าไปเพื่อให้วงจรรวมทำงานได้ ระดับโวลต์เตจของลอจิก 0 และ 1 ของ 89C51 จึงต่อเข้ากับอุปกรณ์ลอจิกแบบ TTL ได้โดยตรง

VSS (ที่ขา 20)เป็นขาที่ต้องต่อกับกราวด์ (Ground) ของแหล่งจ่ายไฟ การต่ออุปกรณ์ทั้งหมดจะต้องมีกราวด์ของอุปกรณ์ต่อเข้าด้วยกัน

PORT 0 (ที่ขา 32 – 39)เป็นพอร์ทขนานขนาด 8 บิตอยู่ที่ขา 32 ถึง 39 เริ่มจากบิต 0 ถึงบิต 7 ตามลำดับ ดังในรูปที่ 3.2 แต่ขาจะเขียนว่า P0.0 , P0.1 , ... , P0.7 P0.7 นั้นหมายถึงบิตที่ 7 ของพอร์ท 0 ซึ่งเป็นบิตที่มีนัยสำคัญสูงสุด (Most Significant) และ P0.0 คือบิต 0 ของพอร์ท 0 เป็นบิตที่มีนัยสำคัญต่ำสุด (Least Significant) พอร์ท 0 นี้ใช้ได้ทั้งการรับ-ส่ง ตำแหน่งของข้อมูลกับหน่วยความจำหรือใช้เป็นพอร์ทรับ-ส่งข้อมูลก็ได้ ข้อมูลที่ส่งออกทางพอร์ท 0 จะถูกแลตช์ (Latch) ไว้ที่ขาของพอร์ท โครงสร้างแต่ละบิตของพอร์ท 0 เป็นแบบ Open Drain Bidirec-tional ดังรูปที่ 3.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 โครงสร้างของพอร์ท 0

รูปที่ 3.3 เมื่อเปรียบเทียบกับรูปที่ 3.1 ส่วนที่ 1 ของรูปที่ 3.3 ก็คือ Port 0 Latch ในรูปที่ 3.1 และส่วนที่ 2 ของรูปที่ 3.3. ก็คือ Port 0 Driver ของรูปที่ 3.1 นั่นเอง

จากโครงสร้างในรูปที่3.3 เมื่อมีคำสั่งของการเขียนข้อมูลมายังพอร์ท 0 ข้อมูลจาก Internal Data Bus จะถูก Latch ไว้ที่ D-FF โดยสัญญาณ " Write To Latch " ที่ถูกสร้างมาจากส่วน Timing and Control และในการอ่านข้อมูลจากพอร์ท 0 จะอ่านได้ 2 แบบ คือ การอ่านข้อมูลที่ส่งไปเก็บไว้ที่พอร์ทก็จะมีสัญญาณ Read Latch มาเพื่ออ่านข้อมูล D-FF กลับเข้ามายัง Internal Data Bus การอ่านข้อมูลอีกแบบก็คือ การอ่านสภาวะของสัญญาณที่เข้ามาทางพอร์ท 0 ก็จะมีสัญญาณ Read Pin มาควบคุมการอ่าน พอร์ท 0 จะใช้งานหลายอย่างดังนี้

1. การใช้สำหรับส่งค่าตำแหน่ง ของหน่วยความจำภายนอกที่ต้องการติดต่อด้วยตำแหน่งหน่วยความจำสูงสุดที่จะติดต่อได้ก็คือ 64 Kbyte จึงมีค่าตำแหน่งหน่วยความจำ 16 บิต ของเลขฐานสองค่าตำแหน่งหน่วยความจำ 8 บิตล่างจะถูกส่งออกไปทางพอร์ท 0 และ 8 บิตบนจะถูกส่งออกไปทางพอร์ท 2

2. ใช้รับ-ส่งข้อมูลกับ Data Memory หรือใช้รับข้องมูลจาก Program Memory

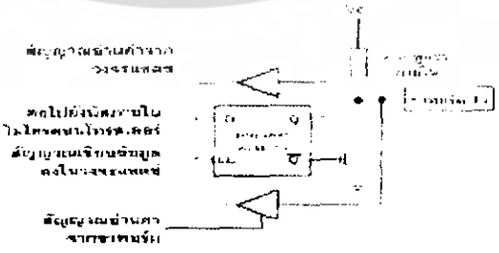
3. ใช้รับ-ส่งข้องมูลออกทางพอร์ทโดยตรงในกรณีที่ไม่มีการใช้หน่วยความจำของ Progran Memory หรือ Data Memory ภายนอก

วงจรภายในส่วน Timing and Control จะเป็นตัวสร้างสัญญาณมาควบคุมวงจรในรูปที่ 3.3 เพื่อให้การทำงานแต่ละอย่างข้างต้น เมื่อแต่ละบิตของพอร์ท 0 ทำงานตามข้อ 1 และ 2 ข้างต้น วงจร Timing and Control จะทำให้สภาวะลอจิกของขา Control เป็น 1 ซึ่งทำให้สวิทช์ MUX อยู่ในตำแหน่งข้างบน เมื่อพอร์ท 0 จะส่งข้อมูลซึ่งเป็นค่าตำแหน่งหน่วยความจำ หรือข้อมูลที่จะเขียนออกไปยังหน่วยความจำภายนอก ก็จะส่งค่าดังกล่าวออกมายัง ADDR/DATA ถ้าข้อมูลที่ส่งมาเป็น 1 จะทำให้สัญญาณออกจาก AND GATE เป็น 1 และสัญญาณที่ออกจาก Inverter เป็น 0 ดังนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FET ตัวบน ON (สภาวะ ON ของ FET คือความต้านทานของขา D และ S มีค่าต่ำมากเสมือนวงจรปิด) ส่วน FET ตัวล่าง OFF (สภาวะ OFF ของ FET คือความต้านทานระหว่างขา D และ S มีค่าสูงมากเสมือนกับเป็นวงจรเปิด) สภาวะลอจิกที่ขา P0.X PIN จะเป็น 1 แต่ถ้าข้อมูลที่ส่งออกมายัง ADDR/DATA เป็น 0 จะทำสัญญาณจาก AND GATE เป็น 0 และสัญญาณที่ออกจาก Inverter เป็น 1 ดังนั้น FET ตัวบนจะ OFF และสวิทช์ MUX จะอยู่ในตำแหนงด้านล่าง ดังนั้น FET ตัวล่างจะ ON หรือ OFF ก็แล้วแต่ข้อมูลที่ขา Q ของ D-FF เมื่อมีการเขียนข้อมูลจาก Internal Data Bus มายัง D-FF ก็จะมีสัญญาณ Write To Latch มายัง D-FF ด้วย ถ้าข้อมูลที่เขียนมาเป็น 1 ก็จะทำให้ขา Q มีสภาวะลอจิกเป็น 0 ทำให้ FET ตัวล่าง OFF ดังนั้นขา P0.X จะอยู่ในสภาวะอิมพีแดนซ์สูง (High Impedance) เพราะ FET ทั้ง 2 ตัว OFF แต่ถ้าข้อมูลที่เขียนมายัง D-FF เป็น 0 จะทำให้ FET ตัวล่าง ON แต่ตัวบน OFF ทำให้สภาวะลอจิกที่ขา P0.X เป็น 1 ดังนั้น Port 0 เมื่อให้ทำงานเป็นพอร์ทส่งข้อมูล (ไม่ใช่ส่งตำแหน่งหน่วยความจำ) จะไม่สามารถแสดงสภาวะลอจิก 1 ได้จึงต้องต่อตัวต้านทาน Pull Up ไว้ภายนอก ระหว่างขา P0.X กับไฟเลี้ยงวงจร ถ้าจะให้ Port 0 สำหรับรับข้อมูลเข้าจะต้องเขียน 1 มาเก็บไว้ยัง D-FF เสียก่อนเพื่อให้ขา P0.X อยู่ในสภาวะ High Impedance แล้วจึงใช้คำสั่งอ่านสภาวะลอจิกเข้าไปยัง Internal Data Bus ต่อไป โดยคำสั่งอานสภาวะลอจิกทางพอร์ท 0 ก็จะทำให้วงจร Timing and Control สร้างสัญญาณ Read Pin สำหรับการอ่านสภาวะลอจิกข้างต้น ถ้าไม่เขียน 1 มาเก็บไว้ยัง D-FF ก่อนที่จะอ่านข้อมูลแล้วอาจมข้อมูลค้างอยู่ที่ D-FF ทำให้ Q เป็น 0 และ Q เป็น 1 ซึ่งทำให้ FET ตัวล่าง ON สัญญาณที่ต่อเข้ามาที่ขา P0.X ไม่ว่าจะมีสภาวะลอจิกใดจะถูกดึงลงกราวด์ ดังนั้นเมื่ออ่านข้อมูลไปก็พบว่าเป็น 0 เสมอ ในการอ่านข้อมูลจากหน่วยความจำภายนอกนั้นวงจร Timing and Control ก็จะเขียนข้อมูลมายัง D-FF ให้เป็น 1 และสร้างสัญญาณ Control ให้มีลอจิกเป็น 0 ก่อนจะอ่านข้อมูลเข้าไปด้วย

Port 1 (ที่ขา 1-8)เป็นพอร์ทขนานขนาด 8 บิต ในรูปที่ 3.2 คือขา P1.0 ถึง P1.7 (ขา 1-8) P1.0 หมายถึงบิต 0 ของพอร์ท 1 ซึ่งเป็นบิต Least Significant Bit และบิต P1.7 หมายถึงบิตที่ 7 ของพอร์ท 1 ซึ่งเป็น Most Significant Bit โครงสร้างของพอร์ท 1 แต่ละบิตมีดังรูปที่ 3.4

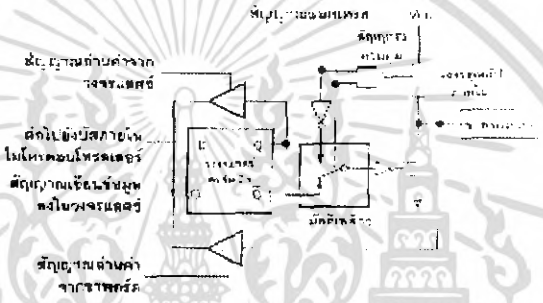


รูปที่ 3.4 โครงสร้างของพอร์ท 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่ 1 คือ Port 1 Latch ในรูปที่ 3.1 ซึ่งมีการทำงานเหมือนส่วนที่ 1 ของพอร์ท 0 ในรูปที่ 3.3 ส่วนที่ 2 คือ Port 1 Driver นี้จะมีตัวต้านทานต่ออยู่เป็น Internal Pull Up พอร์ท 1 นี้จะใช้ทำหน้าที่ในการรับ-ส่งข้อมูล เท่านั้น ข้อมูลที่ส่งมาทางพอร์ท 1 จะถูก Latch ไว้แล้วส่งออกไปทางแต่ละขา ก่อนที่จะอ่านข้อมูลเข้าไปทางพอร์ท 1 จะต้องเขียน 1 เข้าไปทุกบิตของพอร์ท 1 เสียก่อน เพื่อให้ FET อยู่ในสภาวะ OFF เสียก่อน มิฉะนั้นแล้วถ้ามีข้อมูล 0 ส่งออกมาค้างอยู่ที่ D-FF จะทำให้ FET อยู่ในสภาวะ ON ดังนั้นถ้าสัญญาณภายนอกส่งเข้ามาที่ขานี้ก็จะถูกลัดวงจรลงกราวด์ โดยไม่สนใจว่าสภาวะลอจิกของสัญญาณที่เข้ามาจะเป็นอะไร ข้อมูลที่อ่านเข้าไปจึงจะเป็น 0 เสมอ

Port 2 (ที่ขา 21-28)พอร์ทขนานขนาด 8 บิต คือขา P2.0 ถึง P2.7 (บิต 0 ถึงบิต 7 ของพอร์ท 2) ในรูปที่ 3.3 โครงสร้างของพอร์ท 2 แต่ละบิตจะมีดังรูปที่ 3.5



รูปที่ 3.5 โครงสร้างของพอร์ท 2

ลักษณะโครงสร้างจะเหมือนกับ Port 0 แตกต่างกันที่ใน Port 2 นั้นภาค Driver จะใช้งานเพียง 2 ลักษณะคือ

1. ใช้ส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อ ค่าตำแหน่งนี้เป็น 8 บิตบนของตำแหน่ง

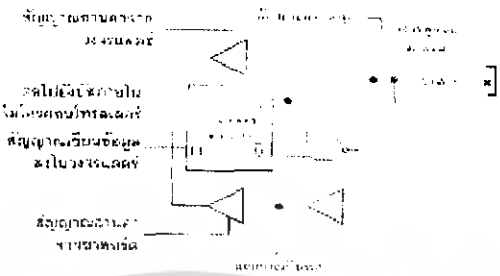
2. ใช้เป็นพอร์ทรับและส่งข้อมูลกับภายนอก

ดังนั้นภาค Driver ของพอร์ท 2 จึงแตกต่างจาก Driver ของพอร์ท 0 โดยที่ในพอร์ท 2 นั้นจะมีเฉพาะ ADDR (ตำแหน่งหน่วยความจำ) เข้ามาที่ MUX (Multiplexer) เท่านั้น นอกนั้นแล้วการทำงานจะเหมือนกันและที่เอาท์พุทของพอร์ท 2 จะมี Internal Pull Up ซึ่งเป็นตัวต้านทานและจะทำให้เอาท์พุทของพอร์ท 2 แสดงสถานะลอจิกเป็น 1 ได้ ถ้า FET อยู่ในสภาวะ OFF บางครั้งเรียกว่า " Quasi-bidirectional " เมื่อใช้เป็นพอร์ทอินพุทก็สามารถทำได้โดยการต่อสัญญาณภายนอกเข้ามาโดยตรง ถ้าสัญญาณภายนอกเข้ามาเป็น 0 ก็จะมีกระแสไหลออกจากพอร์ท (Source Current) ในการที่จะใช้พอร์ทนี้เป็นพอร์ทรับข้อมูลเข้าจะต้องเขียน 1 ไปยังแต่ละบิตของพอร์ทเสียก่อน ดังได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

72109

อธิบายในเรื่อง Port 0 และ Port 1 คือขา P3.0 ถึง P3.7 หรือขา 10-17 ตามลำดับ ในรูปที่ 3.2 พอร์ทนี้มีโครงสร้างดังรูปที่ 3.6



รูปที่ 3.6 โครงสร้างของพอร์ท 3

ส่วนที่ 1 ในรูปที่ 3.6 เป็นส่วน Latch ข้อมูลที่เขียนมายังพอร์ท 3 ทาง Internal Bus เหมือนกับพอร์ทอื่นๆ และพอร์ท 3 จะมี Internal Pull Up อยู่ทุกบิด แต่พอร์ท 3 นี้แต่ละบิตจะใช้การทำงานอื่นได้โดยใช้คำสั่งควบคุมการทำงาน ในส่วนที่ 2 จะมีสัญญาณ Alternative Output Function เป็นสัญญาณส่งออกในกรณีที่ใช้พอร์ท 3 ทำงานในฟังก์ชั่นอื่น และจุด Alternative Output Function เป็นจุดที่จะเอาสัญญาณไปเข้ากับส่วนอื่นตามการทำงานของบิตนั้นแต่ละบิตของพอร์ท 3 จะมีฟังก์ชั่นอื่นดังนี้

P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม

P3.1/TXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม

P3.2/$\overline{\text{INT0}}$ (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

P3.1/$\overline{\text{INT1}}$ (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

P3.4/TO (Time / Counter 0 External Input) ขารับสัญญาณเข้าไปยังวงจร Time / Counter 0 ที่ทำหน้าที่นับจำนวนไซเคิลของสัญญาณ TO นี้ หรือสัญญาณนาฬิกาก็ได้

P3.5 /T1 (Timer / Counter 1 External Input) ขารับสัญญาณเข้าไปยังวงจร Timer / Counter 1 ซึ่งมีการทำงานเหมือนกับ TO

P3.6/$\overline{\text{WR}}$ (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 89C51

P3.7/$\overline{\text{RD}}$ (External Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก RST (ที่ขา 9)

ขารีเซทขานี้จะใช้ทำการรีเซทการทำงานของ 89C51 ที่ขา RST ภายใน 89C51 จะมีตัวต้านทานต่อระหว่างขานี้กับกราวด์ ถ้าป้อนสัญญาณที่มีสภาวะลอจิกเป็น 1 เข้าไปที่ขานี้จะเป็นการรีเซทการทำงานของ 89C51 ดังนั้นจึงสามารถต่อตัวเก็บประจุภายนอกระหว่างขา RST กับไฟเลี้ยง

เอกสารนี้มีลิขสิทธิ์ ห้ามนำไปใช้ในการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

+5 V เพื่อให้เกิดการรีเซท เมื่อเริ่มป้อนไฟเลี้ยงให้กับ 89C51 ซึ่งเรียกว่า Power On Reset การรีเซจะทำให้ค่าในรีจิสเตอร์ต่างๆเปลี่ยนไปเป็นค่าหนึ่งดังตารางรูปที่ 3.7

Register	CONTENT
PC	0000H
ACC	00H
B	00H
PSW	00H
SP	00H
DPTRB	0000H
P0-P3	0FFH
IP	00H
IE	0X000000B
TMOD	00H
TCON	00H
T2CON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
TH2	00H
TL2	00H
RCAP2H	00H
RCAP2L	00H
SCON	00H
SBUF	NCETERMINAE
CCON	00H

รูปที่ 3.7 ค่าในริสเตอร์เมื่อเกิดการรีเซท 89C51

ในตารางรูปที่ 3.7 ช่องทางขวาเป็นค่าของทางรีจิสเตอร์ที่อยู่ทางซ้ายเมื่อสิ้นสุดการรีเซทในรีจิสเตอร์ SBUF เมื่อสิ้นสุดการรีเซทจะมีค่าที่ไม่แน่นอน และพอร์ทจะอยู่ในสภาวะลอจิก1 ทุกบิตตลอดเวลาที่สัญญาณของขา RST เป็น HIGH อยู่เมื่อสัญญาณที่ขา RST กลับเป็น 0 ก็จะออกจากการรีเซท 89C51 จะเริ่มทำงานจากคำสั่งที่อยู่ใน Program Memory ตำแหน่ง 0000H เพราะค่าของรีจิสเตอร์ PC (Program Counter) ซึ่งใช้ชี้ตำแหน่งโปรแกรมที่จะทำงาน ถูกเปลี่ยนให้เป็น 0000H ดังนั้นผู้ใช้จะต้องเขียนโปรแกรมมาเก็บไว้ที่ตำแหน่ง 0000H ในเครื่องไมโครคอมพิวเตอร์แบบบอร์ดเดี่ยว (Single Board Micro Computer) จะมีโปรแกรมที่เขียนเก็บไว้เริ่มจากตำแหน่ง 0000H นี้เรียกว่า มอนิเตอร์โปรแกรม (Monitor Program) ที่จะคอยรับการกดแป้นพิมพ์และแสดงผลทางตัวแสดงผลแบบ 7 Segment

ALE (ที่ขา 30) Address Latch Enable ขานี้จะส่งสัญญาณที่มีความถี่ 1/6 เท่าของสัญญาณนาฬิกาจากออสซิลเลเตอร์สัญญาณนี้จะส่งออกมาตลอดเวลา ยกเว้นบางครั้งของการติดต่อกับหน่วยความจำสำหรับข้อมูลภายนอก 89C51 สัญญาณนี้จะใช้บอกกับอุปกรณ์ภายนอก 89C51 ว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขณะนี้สัญญาณนี้ Active (เป็นลอจิก 1) จะมีการส่งข้อมูลที่เป็น 8 บิตล่างของตำแหน่ง หน่วยความจำภายนอก 89C51 ที่ต้องการติดต่อออกไปทางพอร์ท 0 อุปกรณ์ภายนอกจะใช้สัญญาณนี้ในการ Latch ข้อมูลไว้เพราะพอร์ท 0 อุปกรณ์ภายนกจะใช้สัญญาณนี้ในการ Latch ข้อมูลไว้เพราะพอร์ท 0 จะส่งค่าตำแหน่งหน่วยความจำออกมาเพียงชั่วขณะเท่านั้น ซึ่งในเวลาต่อมาพอร์ท 0 จะใช้รับ-ส่งขอมูลกับหน่วยความจำภายนอกสัญญาณ ALE จะสามารถต่อเข้ากับอุปกรณ์ TTL ชนิด LS ได้ถึง 8 อินพุท

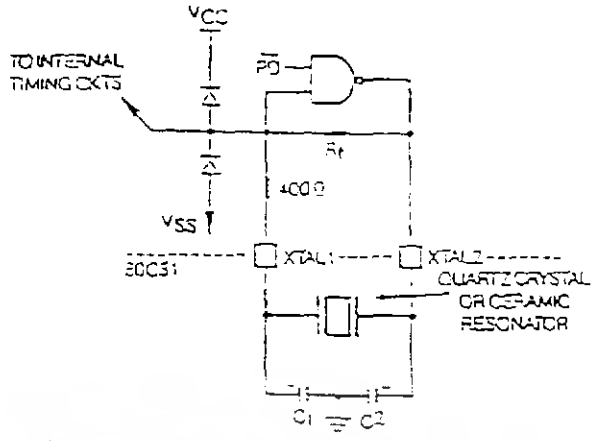
$\overline{\text{PSEN}}$ (ที่ขา 20) Program Store Enable เป็นขาที่ 29 ในรูปที่ 3.2 ขานี้ปกติจะให้ลอจิก 1 แต่จะส่งลอจิก 0 เมื่อต้องการอ่านคำสั่ง (Fetch Instruction) ที่จะนำไปทำงานมาจากหน่วยความจำสำหรับโปรแกรมภายนอก 89C51 ในกรณีที่อ่านคำสั่งซึ่งเก็บอยู่ในหน่วยความจำสำหรับโปรแกรมภายใน 89C51 แล้วสัญญาณนี้จะไม่เปลี่ยนลอจิกเป็น 0 ขา PSEN นี้สามารถต่อไปยังขาอินพุทของ TTL ชนิด LS ได้ถึง 8 อินพุท

$\overline{\text{EA}}$ (ที่ขา 31) External Access ขาที่ 31 ของรูปที่ 3.2 ขานี้เป็นขาอินพุทที่ต่อเข้าไปยังวงจร Timing 0 And Control ในรูปที่ 3.1 เพื่อควบคุมการสร้างสัญญาณ $\overline{\text{PSEN}}$ ถ้าป้อนสัญญาณลอจิก 0 เข้าไปที่ขา $\overline{\text{EA}}$ นี้แสดงว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ที่ต้องการให้ทำงานถูกเก็บไว้ภายนอก 89C51 จะต้องสร้างสัญญาณ $\overline{\text{PSEN}}$ ออกไปยังภายนอกเพื่อทำการ FETCH คำสั่งเข้ามาทำงาน แต่ถ้าสัญญาณที่ป้อนให้ขา $\overline{\text{EA}}$ เป็น 1 หมายความว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ถูกเก็บไว้ภายใน 89C51 การทำงานในตำแหน่งหน่วยความจำช่วงนี้จะอ่านคำสั่งต่างๆ จาก ROM ภายใน 89C51

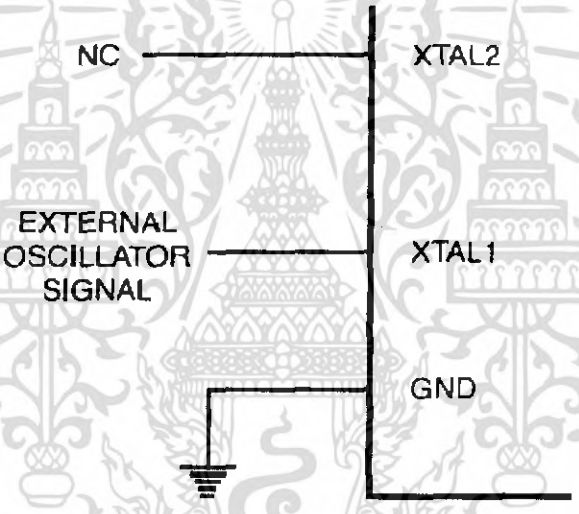
XTAL 1 (ที่ขา 19) ขาที่ 19 ของรูปที่ 3.2 นี้จะต่อเข้ากับขาของ Inverting Amplifier (วงจรขยายแบบป้อนกับเฟสสัญญาณ) ที่ประกอบเป็นวงจรออสซิลเลเตอร์ ในรูปที่ 3.8 จะเห็นวงจรภายในของออสซิลเลเตอร์ NAND GATE จะทำหน้าที่เป็นวงจรขยายแบบกลับเฟสของสัญญาณที่จะควบคุมให้มีการออสซิลเลตหรือไม่ก็ขึ้นกับสัญญาณ PD ซึ่งต่อมาจากบิต PD ของรีจิสเตอร์ PCON ถ้าต้องใช้สัญญาณนาฬิกามาควบคุมการทำงานของ 89C51 ก็ให้ป้อนสัญญาณเข้ามาที่จุดนี้ แต่ถ้าต้องการใช้วงจรออสซิลเลเตอร์ภายในก็ให้ต่อ Crytal หรือเซรามิคเรโซเนเตอร์ ดังรูปที่ 3.8 คาปาซิเตอร์ในวงจรควรมีค่าประมาณ 20 PF

XTAL 2 (ที่ขา 18) ขาที่ 18 ของรูปที่ 3.2 ขานี้เป็นจุดเอาท์พุทของวงจรขยายแบบกลับสัญญาณที่ประกอบเป็นวงจรออสซิลเลเตอร์ (อินพุทคือขา XTAL 1) ถ้าจะใช้สัญญาณนาฬิกาที่สร้างมาจากภายนอกของ 89C51 แล้ว ให้ปล่อยขานี้ลอยไว้แล้วป้อนสัญญาณนาฬิกาจากภายนกเข้ามาที่ขา XTAL 1 ดังรูปที่ 3.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 วงจรออสซิลเลเตอร์ภายใน 89C51



รูปที่ 3.9 89C51 ที่ทำงานโดยสัญญาณที่มาจากภายนอก

3.2 การทำงานของ 89C51

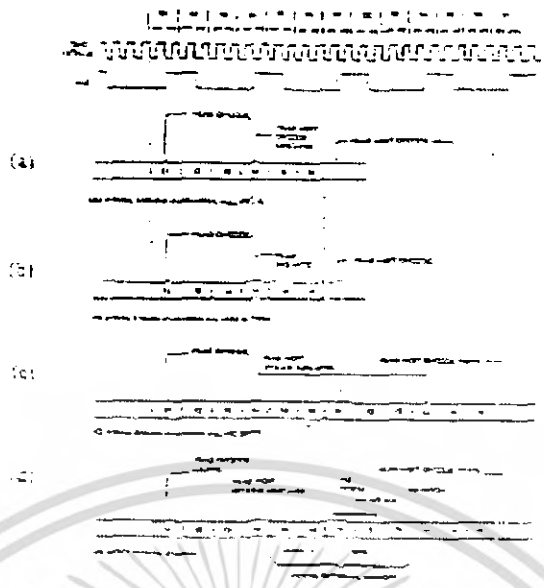
คอมพิวเตอร์จะทำงานด้วยวงจรที่เรียกว่า ฮาร์ดแวร์ (Hardware) ประกอบขึ้นมาเพียงอย่างเดียวไม่ได้ จะต้องมีโปรแกรมหรือคำสั่งที่จัดเรียงกันไว้ให้คอมพิวเตอร์ทำงานตามลำดับใน 89C51 ก็เช่นกันผู้ใช้ต้องเขียนโปรแกรมเป็นภาษาเครื่อง . ซึ่งอยู่ในรูปของเลขฐาน 2 เก็บไว้ในหน่วยความจำประเภท Program Memory แต่ละคำสั่งของ 89C51 อาจประกอบไปด้วย 1, 2 หรือ 3 ไบท์ แล้วแต่ว่าจะเป็นคำสั่งให้ทำงานอะไรคอมพิวเตอร์ก็เหมือนกับคนที่ต้องทำงานตามคำสั่ง เมื่อรับคำสั่งแล้วก็จะไปทำตามคำสั่งนั้นเสร็จสิ้นแล้วกลับมารับคำสั่งใหม่ต่อไป

จากรูปที่ 3.1 เมื่อเริ่มป้อนไฟเลี้ยงให้กับ 89C51 ซึ่งมีวงจรนับ Power On Reset ต่ออยู่จะมีการรีเซทเกิดขึ้น การทำงานภายใน89C51 จะเริ่มจากบล็อก Program counter ซึ่งเป็นวงจรนับ

เอกสารนี้ ... ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Counter Circuit) ชนิดหนึ่ง ส่งค่าตำแหน่งหน่วยความจำสำหรับโปรแกรมลงไปบนบัสหมายเลข 1 บัสนี้มีขนาด 16 บิท ค่าตำแหน่งหน่วยความจำนี้จะถูกส่งไปเก็บไว้ที่ 0 Program ADDR Register ที่เป็นวงจร Latch ข้อมูลซึ่งเป็นค่าตำแหน่งหน่วยความจำจะปรากฏที่บนบัส 16 บิต ค่าตำแหน่งหน่วยความจำนี้จะถูกส่งไปเก็บไว้ที่ 0 Program ADDR Register ที่เป็นวงจร Latch ข้อมูลซึ่งเป็นค่าตำแหน่งหน่วยความจำจะปรากฏที่บนบัส 16 บิต หมายเลข 2 ถ้าเป็นค่าตำแหน่งหน่วยความจำแรกหลังจากรีเซท ค่าตำแหน่งหน่วยความจำจะเป็น 0000H หน่วยความจำสำหรับโปรแกรมจะเลือกได้ว่าเป็น ROM ภายในหรือภายนอก 89C51 โดยการป้อนสถาวะลอจิกเข้าไปที่ 89C51 ทางขา EA ซึ่งต่ออยู่กับส่วนTiming And Control ทำการถอดรหัสแล้วสร้างสัญญาณควบคุมต่อไป แล้วแต่ว่าเป็นคำสั่งให้ทำงานอะไร ในกรณีที่เลือก ROM ภายนอก 89C51 โดยป้อนสัญญาณลอจิก 1 เข้าไปที่ขา EA จะทำให้วงจร Timing and Control ส่งสัญญาณไปยังพอร์ท 0 และพอร์ท 2 เพื่อส่งค่าตำแหน่งหน่วยความจำบนบัสหมายเลข 2 ออกไปชี้หน่วยความจำภายนอก จากนั้นจะอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาทางพอร์ท 0 ไปยัง Internal Data Bus แล้วนำไปเก็บไว้ที่ Instruction Register เรียกว่าเป็นช่วงองการ Fetch (Fetch Cycle) ช่วงต่อไปจะเป็นช่วงของการทำงานตามคำสั่งเรียกว่า Execute Cycle เช่นถ้าเป็นคำสั่งให้บวกข้อมูลในรีจิสเตอร์ Accumulator กับข้อมูลจากหน่วยความจำ Data Memory ภายใน RAM ตำแหน่ง 23H วงจร Timing and Control ก็จะส่งสัญญาณให้ Instruction Register ส่งค่าตำแหน่งหน่วยความจำ 23H ลงไปยัง Internal Data Bus แล้วนำข้อมูลไปเก็บไว้ที่ RAM ADDR Register เพื่อใช้ชี้ตำแหน่งหน่วยความจำ RAM จากนั้น Timing and Control จะสั่งให้ RAM ส่งข้อมูลที่เก็บอยู่ในหน่วยความจำตำแหน่ง 23H ลงมายังInternal Data Bus แล้วนำข้อมูลไปเก็บไว้ที่ TMP1 (วงจร Latch) ขณะเดียวกันวงจร Timing and Control ก็จะส่งสัญญาณไปยัง ACC ให้ส่งข้อมูลมายัง TMP2 (วงจร Latch) วงจร ALU ซึ่งโครงสร้างเป็นวงจรทำการคำนวณทางคณิตศาสตร์ (บวก ลบ คูณ หาร) และยังสามารถทำงานทางลอจิก (AND OR NOT XOR) จะทำการบวกเลขจาก TMP1 และ TMP2 เข้าด้วยกัน ผลลัพธ์ที่ได้จะส่งผ่าน Internal Data Bus กลับไปเก็บยัง ACC PSW (Program Status Word) ซึ่งจะทำหน้าที่เก็บสถานะผลลัพธ์ของการทำงานใน ALUเช่นผลลัพธ์มีค่าเกิน 8 บิตก็จะทำให้บิตหนึ่งใน PSW ถูก SET เป็น 1 การทำงานที่กล่าวมาข้างต้นจะขึ้นกับสัญญาณควบคุมที่สร้างมาจากวงจร Timing and Control และสัญญาณที่สร้างขึ้นนี้จะอ้างอิงกับสัญญาณนาฬิกาที่สร้างจากวงจร Oscillator ทำให้การทำงานต่างๆ เป็นไปตามลำดับที่ผู้ผลิตได้ออกแบบไว้ ดังในรูปที่ 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.10 ลำดับสถานะการทำงานใน MCS 51

คำสั่งแต่ละคำสั่งของ 89C51 จะใช้เลาทำงาน 1 , 2 หรือ 3 ไซเคิลของเครื่อง (Machine Cycle) แล้วแต่ว่าเป็นคำสั่งประเภทใด 1 ไซเคิลของเครื่องจะใช้เวลา 12 ไซเคิลของสัญญาณนาฬิกา ดังนั้นแต่ละคำสั่งของ 89C51 จะเวลาการทำงาน 12 , 24 หรือ 36 ไซเคิลของสัญญาณนาฬิกานั่นเอง แต่ละไซเคิลของเครื่องจะถูกแบ่งออกเป็น 6 State คือ S1 , S2 , S3 , S4 , S5 และ S6 แต่ละ State จะประกอบด้วย 2 ไซเคิลของสัญญาณนาฬิกา ในไซเคิลแรกจะเรียกว่าเฟส 1 (P1) และไซเคิลที่ 2 เรียกว่า เฟส 2 (P2) ในแต่ละเฟสจะนับตั้งแต่ขอบขาลงของสัญญาณนาฬิกาถึงขอบขาลงของสัญญาณนาฬิกาที่อยู่ถัดไปดังในรูปที่ 3.10 เมื่อ89C51 ทำงานเสร็จ 1 ไซเคิลของเครื่องก็จะเริ่มทำงาน State 1 Phase 1 (S1 P1) ของไซเคิลต่อไป 1 ไซเคิลของเครื่องวงจร Timing and Control จะสร้างสัญญาณ ALE ออกมา 2 ไซเคิล เพื่อ FETCH คำสั่งเข้าไป 2 ครั้งเสมอ ที่บริเวณขอบขาขึ้นของสัญญาณ ALE คำสั่งใดจะมีกี่ไบท์หรือใช้เวลาทำงานกี่ไซเคิลจะดูได้จากตารางชุดคำสั่ง 89C51

คำสั่งประเภท 1 ไบท์ 1 ไซเคิลของเครื่องได้แก่คำสั่ง INC A จะมีการอ่านคำสั่งจากหน่วยความจำสำหรับโปรแกรม 2 ครั้ง ที่เวลาประมาณขอบขาขึ้นของสัญญาณ ALE เมื่อคำสั่งแรกถูกอ่านเข้าไปที่เวลา ที่เวลาขอบขาขึ้นของสัญญาณ ALE แรกแล้วนำไปเก็บที่ Instruction Register เพื่อทำให้วงจร Timing and Control ถอดรหัสแล้วเข้าอยู่การ Execute ขณะเดียวกันก็จะเริ่มต้นการ Fetch คำสั่งที่อยู่ในหน่วยความจำคำสั่งถัดไปเข้ามาและคำสั่งที่ 2 จะถูกอ่านเข้ามาที่เวลาขอบขาขึ้นของสัญญาณ ALE ถัดไป วงจร Timing and Control เมื่อถอดรหัสคำสั่งแรกก็จะทราบว่าการทำงานคำสั่งนี้ให้สิ้นสุดจะใช้คำสั่งเพียง 1 ไบท์ ดังนั้นคำสั่งที่ถูกอ่านมาไบท์ที่ 2 จะไม่ถูกนำมาทำงาน เพียงแต่อ่านเข้ามาแล้วทิ้งไป (Discard) ดังในรูปที่ 3.10 a

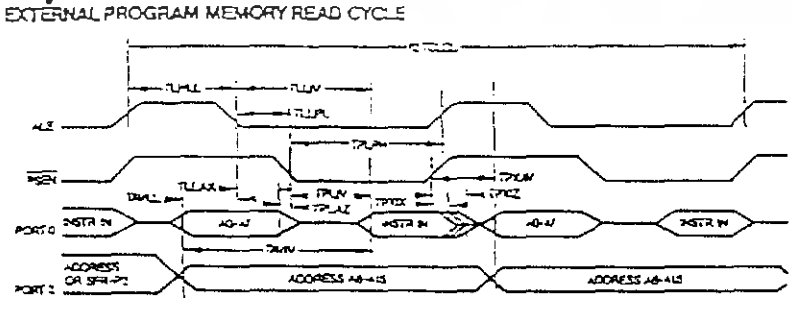
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งประเภท 2 ไบท์ และใช้เวลาเพียง 1 ไซเคิลของเครื่องได้แก่คำสั่ง ADD A # Data ในหนึ่งไซเคิลของเครื่องนี้ จะมีการอ่านคำสั่งเข้ามา 2 ไบท์ เหมือนกับคำสั่งประเภท 1 ไบท์ 1 ไซเคิลของเครื่อง แตกต่างกันที่ไบท์ที่ 2 จะถูกนำมาใช้งานด้วยไม่ได้ทิ้งไปดังในรูปที่ 3.10 b ตัวอย่างของคำสั่ง ADD A , # 33H จะเขียนเป็นภาษาเครื่องได้ 2 ไบท์ คือ 24 และ 33 เมื่ออ่านคำสั่งไบท์แรกคือ 24 เข้าไปไว้ที่ Instruction Register และ Timing and Control จะถอดรหัสเป็นคำสั่งบวกเลข ก็จะส่งสัญญาณไปยัง Accumulator ให้เอาข้อมูลไปไว้ที่ TMP1 เมื่อคำสั่งที่ 2 ถูอ่านเข้ามาที่ Instruction Register และ Timing and Control จะส่งให้เอาข้อมูลไบท์ที่ 2 ส่งลงไปยัง Internal Data Bus ไปเก็บยัง TMP1 จากนั้นวงจร ALU จะนำเอาข้อมูล TMP1 และ TMP2 มาบวกกัน ผลลัพธ์ที่ได้จะส่งออกมาจาก ALU ไปยัง Internal Data Bus แล้วไปเก็บไว้ที่ Accumulator

คำสั่งประเภท 1 , 2 หรือ 3 ไบท์ ที่ใช้เวลาทำงาน 2 ไซเคิลของเครื่อง เช่น คำสั่ง INC DPTR จะมีการอ่านคำสั่งเข้าไป4 ครั้ง ทุกๆ ขอบขาขึ้นของสัญญาณ ALE ที่มี 2 ครั้งต่อ 1 ไซเคิลของเครื่องถ้าเป็นคำสั่งประเภท 1 , 2 หรือ 3 ไบท์ วงจร Timing and Control จะเอาคำสั่ง 1 , 2 หรือ 3 ไบท์แรกเท่านั้นไปทำงาน ส่วนคำสั่งที่เหลือทิ้งไปดังในรูปที่ 3.10 c คำสั่ง 1 ไบท์ที่ใช้เวลาทำงาน 2 ไซเคิลของเครื่องที่กล่าวมาแล้วจะไม่รวมถึงคำสั่ง MOVX ซึ่งใช้ในการอ่านหรือเขียนข้อมูลกับหน่วยความจำ Data memory ภายนอก การทำงานของคำสั่งนี้จะมีการ Fetch คำสั่งเข้าไป 2 ไบท์ในไซเคิลของเครื่องแรก ในไซเคิลของเครื่องที่ 2 จะไม่มีการ Fetch คำสั่งเข้าไป แต่จะเป็นช่วงเวลาของการอ่านหรือเขียนข้อมูลกับ Data memory ภายนอก สัญญาณ ALE ซึ่งปกติเปลี่ยนเป็น 1 ที่ S1 P2 ก็จะไม่เปลี่ยนเป็น 1 ไซเคิลของเครื่องที่ 2 โดยจะเป็นศูนย์อยู่จนกว่าจะถึงเวลา S4 P2 ของไซเคิลของเครื่องที่ 2 สัญญาณ ALE จะเปลี่ยนเป็น 1 เพื่อการอ่านหรือเขียนข้อมูลกับ Data Memory ภายนอก

3.3 ไดอะแกรมเวลาของการติดต่อกับหน่วยความจำ

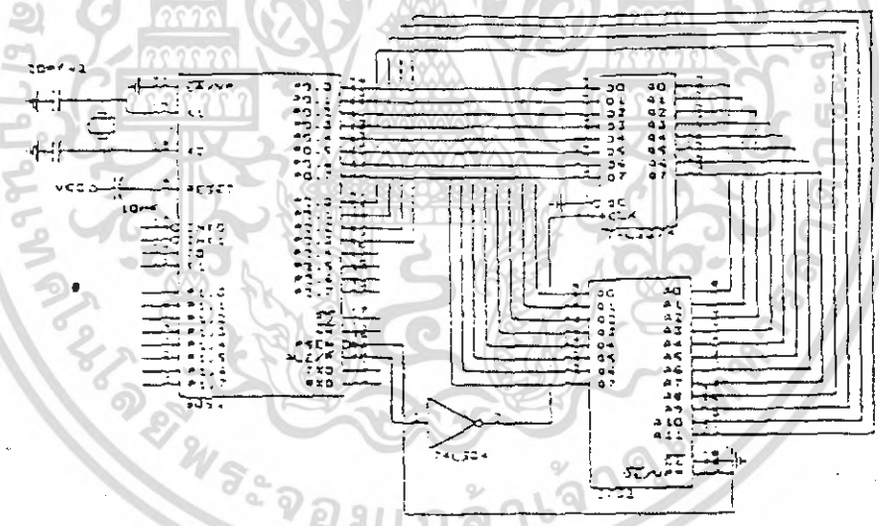
การอ่านข้อมูลจากหน่วยความจำสำหรับโปรแกรมภายนอก 89C51 นั้น ลำดับสัญญาณตามเวลา (Timing Diagram) ของสัญญาณที่ทำการอ่านคำสั่ง ดังรูปที่ 3.11



รูปที่ 3.11 Timing Diagram ของการอ่านโปรแกรมจากหน่วยความจำภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ่านคำสั่งจาก Program area ภายนอก จะเริ่มจาก 89C51 ส่งสัญญาณลอจิก 1 ออกมาทางขา ALE ขณะนี้สัญญาณที่ขา PSEN จะเป็น 1 จากนั้น Port 0 จะส่งค่าตำแหน่งหน่วยความจำ 8 บิตล่าง และพอร์ท 2 จะส่งตำแหน่งหน่วยความจำ 8 บิตบนออกมา แล้วสัญญาณ ALE จะกลับเป็น 0 อุปกรณ์ภายนอกจะสามารถใช้ขอบขาลงของสัญญาณ ALE เพื่อ Latch ตำแหน่งหน่วยความจำที่พอร์ท 0 เอาไว้ จากนั้นพอรท 0 ก็จะยกเลิกการส่งค่าตำแหน่งหน่วยความจำ เข้าสู่สภาวะ High Impedance และสัญญาณ PSEN จะเป็น 0 เพื่อเตรียมรับคำสั่ง ที่ส่งออกจากหน่วยความจำภายนอกเข้าไปยัง 89C51 เพื่อทำงานต่อไป เมื่อคำสั่งถูกอ่านเข้าไปเก็บใน Instruction Register (ดูรูปที่ 3.1) แล้วสัญญาณ PSEN จะกลับเป็น 1 พร้อมกับสัญญาณ ALE ก็จะกลับเป็น High เพื่อจะอ่านคำสั่งต่อไปทำงาน ข้อมูลในพอร์ท 2 จะคงที่ตลอดเวลา ตั้งแต่สัญญาณ ALE เป็น 1 จนกระทั่งสัญญาณALE เปลี่ยนเป็น 0 และกลับเป็น 1 อีกครั้ง จากนั้นจะเริ่มลำดับการ Fetch ข้อมูลไบท์ที่ 2 จากหน่วยความจำสำหรับโปรแกรม ซึ่งมีการเปลี่ยนแปลงของสัญญาณตามเวลา เหมือนกบการ Fetch ไบท์แรกนั่นเอง จาก Timing Diagram ดังกล่าว จะออกแบบวงจรที่มี Program Memory อยู่ภายนอก 89C51 ดังตัวอย่างในรูปที่ 3.12



รูปที่ 3.12 วงจรที่มี Program Memory ภายนอก 89C51

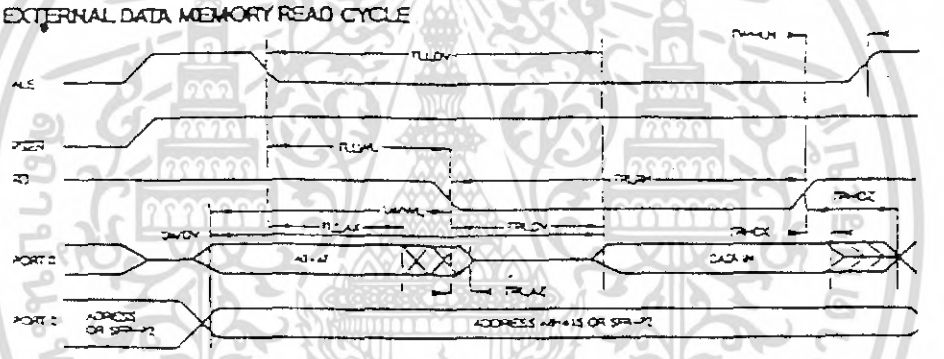
74LS374 ในรูปที่ 3.12 จะทำหน้าที่ Latch ตำแหน่งในหน่วยความจำ 8 บิตล่างที่เวลาขอบขาลงของสัญญาณ ALE ซึ่งสัญญาณ ALE จะถูกกลับให้เป็นตรงกันข้ามโดย Inverter 74LS04 ก่อนที่จะป้อนให้กับขา CK ของ 74LS374 และที่ขอบขาขึ้นของสัญญาณที่ออกจาก 74LS04 และ Latch ตำแหน่งหน่วยความจำข้อมูลที่ออกจาก 74LS374 จะเป็นค่า 8 บิตล่างของหน่วยความจำที่ต้องการติดต่อ ในวงจรได้ต่อค่าตำแหน่งหน่วยความจำ 8 บิตเข้ากับ A0 ถึง A7 ของ EPROM และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลจากพอร์ท 2 บิต P2.0 ถึง P2.3 จะต่อเข้ากับ A8-A11 ของ EPROM โดยตรงเพราะค่าตำแหน่งหน่วยความจำ 8 บิตบนที่ออกมาจากพอร์ท 2 จะคงที่ตลอดเวลา ขา PSEN ของ 89C51 จะถูกต่อเข้ากับขา OE ของ EPROM 2716 ดังนั้นเมื่อสัญญาณ PSEN มีสภาวะลอจิกเป็น 0 ก็จะส่งคำส่งที่เก็บไว้ใน EPROM ณ ตำแหน่งที่ชี้โดยข้อมูลที่ขา A0 ถึง A11 ออกมายังพอร์ท 0 และถูก 89C51 เก็บไปทำงานต่อไป

การอ่าน-เขียนข้อมูลกับหน่วยความจำสำหรับข้อมูลภายนอก 89C51 การอ่านเขียนข้อมูลกับ Data Memory ภายใน 89C51 นั้นจะมีสัญญาสร้างมาจากส่วน Timing and Control โดยที่ผู้ใช้ไม่จำเป็นที่จะต้องทำความเข้าใจแต่กรอ่านเขียนข้อมูลกับ Data Memory ภายนอก อันเนื่องมาจากคำสั่ง MOVX นั้น เมื่อคำสั่งดังกล่าวถูกอ่านเข้ามายัง Instruction Register แล้ว Timing and Control จะทำการถอดรหัสและสร้างสัญญาณควบคุมดังนี้

การอ่านข้อมูลจาก External Data Memory จะมีไดอะแกรมสัญญาณตามเวลาดังรูปที่ 3.13



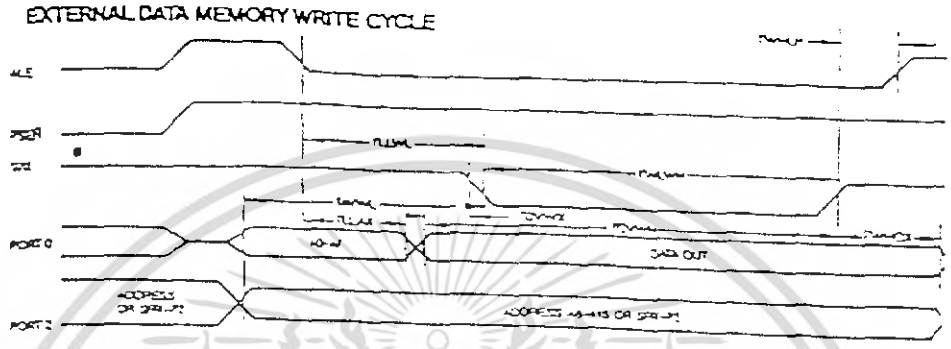
รูปที่ 3.13 Timing Diagram ของการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก 89C51

การทำงานจะเริ่มจากการส่งค่าตำแหน่งหน่วยความจำภายนอก 8 บิตล่างออกพอร์ท 0 และ 8 บิตบนออกทางพอร์ท 2 เมื่อส่งค่าตำแหน่งแล้ว สัญญาณ ALE ซึ่งเดิมมีลอจิกเป็น 1 จะกลับมาเป็น 0 เพื่อให้อุปกรณ์ภายนอกสามารถ Latch ตำแหน่งหน่วยความจำไว้เหมือนกับในการอ่านข้อมูลจากหน่วยความจำสำหรับโปรแกรมภายนอก 89C51 เพื่อส่งไปยังหน่วยความจำ แม้ว่าข้อมูลทางพอร์ท 0 จะเปลี่ยนแปลงไปก็ยังมีค่าตำแหน่งความจำส่งไปยังหน่วยความจำในระหว่างการติดต่อกับ Data Memory นี้สัญญาณ PSEN จะเป็น 1 ตลอดเพราะสัญญาณ PSEN เป็น Active ก็ต่อเมื่อเป็นการติดต่อหน่วยความจำ สำหรับโปรแกรมภายนอก 89C51 เท่านั้น 89C51 ก็จะส่งสัญญาณ ลอจิก 0 ออกมาทางขา RD (P3.7) เพื่อบอกกับหน่วยความจำภายนอกว่าต้องการอ่านข้อมูลเข้าไปเมื่อ 89C51 ส่งสัญญาณ RD เป็นลอจิก 0 ทำให้พอร์ท 0 เข้าสู่สถานะ High Impedance พร้อมที่จะให้หน่วยความจำภายนอกส่งข้อมูลมาบนพอร์ท 0 ซึ่งส่งมาจากหน่วยความจำภายนอกถูกอ่านเข้าไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

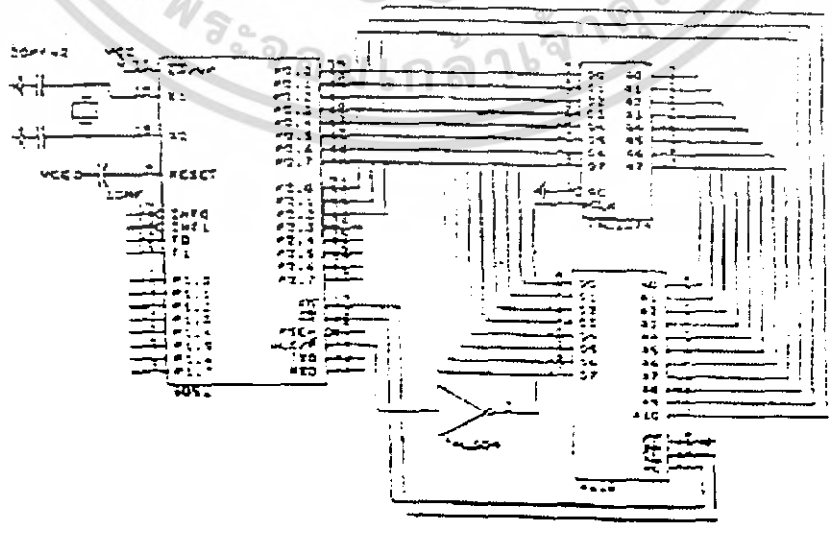
เก็บที่เวลาขอบขาขึ้นของสัญญาณ RD จากนั้นสัญญาณ ALE จะกลับเป็น 1 เพื่อเริ่มการทำงานในคำสั่งต่อไปในระหว่างการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอกนี้ พอร์ท 2 จะส่งค่าตำแหน่งหน่วยความจำ 8 บิตบนออกมาตลอดเวลา

การเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 89C51 จะมีไดอะแกรมสัญญาณตามเวลาดังรูป 3.14



รูปที่ 3.14 Timing Diagram ของการเขียนข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก 89C51

เมื่อ 89C51 ส่งค่าตำแหน่งหน่วยความจำ 8 บิตล่างไปทางพอร์ท 0 และหน่วยความจำ 8 บิตบนลงไปทางพอร์ท 2 แล้วสัญญาณ ALE ก็จะกลับเป็น 0 อุปกรณ์ภายนอกจะสามารถใช้สัญญาณนี้ในการ Latch ค่าตำแหน่งหน่วยความจำในพอร์ท 0 เหมือนกับการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก เมื่อสัญญาณALE เป็น 0 แล้ว89C51 จะส่งข้อมูลที่ต้องการเขียนไปยังพอร์ท 0 แล้วจะให้สัญญาณ WR เปลี่ยนสภาวะลอจิกเป็น 0 ขณะนี้หน่วยความจำภายนอกจะต้องเขียนข้อมูลไปเก็บยังตำแหน่งที่กำหนด จากนั้นสัญญาณ WR จะกลับเป็น 1 เพื่อเป็นการบอกสิ้นสุดการเขียนข้อมูลภายนอกที่สามารถอ่านและเขียนข้อมูลดังรูปที่ 3.15



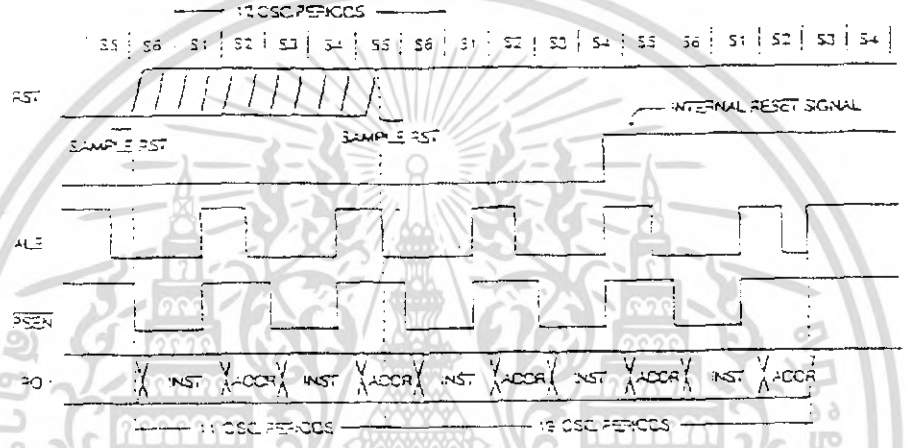
รูปที่ 3.15 วงจรที่มีหน่วยความจำสำหรับข้อมูลที่อยู่ภายนอก 89C51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

74LS374 ในรูปจะใช้สำหรับ Latch คาตำแหน่งความจำ 8 บิตล่างไว้ แม้ว่าข้อมูลบนพอร์ท 2 จะเปลี่ยนไป สัญญาณ RD และ WR จะอ่านหรือเขียนข้อมูลจากหน่วยความจำภายนอก 6116 เป็นหน่วยความจำแบบ RAMที่สามารถจะอ่านและเขียนข้อมูลได้

3.4 การรีเซท

เมื่อป้อนสัญญาณที่มีสาวะลอจิกเป็น 1 เข้าไปทางขวา RST ซึ่งจะไม่ได้เกิดการรีเซทขึ้นทันทีทันใดแต่สำหรับการเกิดรีเซทจะแสดงได้ดังไดอะแกรมเวลาในรูปที่ 3.16



รูปที่ 3.16 ไดอะแกรมตามเวลาของการรีเซท

ในรูปที่ 3.16 เป็น Timing Diagram ของการรีเซทสภาวะลอจิกของสัญญาณที่ขา RST ที่ถูกอ่านเขามาที่เวลาที่ S5 P2 ของทุกๆไซเคิลของเครื่อง ในกรณีที่เป็นคำสั่งซึ่งมีการทำงานเสร็จสิ้นใน 2 ไซเคิล ของเครื่องก็จะตรวจสอบเฉพาะสัญญาณที่อ่านเข้ามาในไซเคิลที่ 2 องการทำงาน ดังนั้นในการรีเซทจะต้องป้อนสัญญาณที่มีสภาวะลอจิก 1 เข้าไปที่ขาอย่างน้อย 2 ไซเคิลของเครื่องหรือ 24 ไซเคิล ของสัญญาณนาฬิกาที่สร้างจากวงจรออสซิลเลเตอร์ภายใน 89C51 เพื่อให้แน่ใจว่าสัญญาณรีเซท จะถูกอ่านเข้าไปตรวจสอบและทำงาน ขณะที่ทำงานรีเซท 89C51 ออสซิลเลเตอร์จึงต้องทำงานอยู่ด้วย เมื่อ 8C51 สุ่มข้อมูลที่ขา RST แล้วตรวจพบว่าเป็นสภาวะลอจิก 1 ก็จะสร้างสัญญาณรีเซทขึ้นภายในที่เวลา S2P4 ของไซเคิลเครื่องถัดไป ข้อมูลแต่ละพอร์ทที่ส่งออกมา จะยังคงปรากฏทีพอร์ทจนกว่าจะเกิดการรีเซทขึ้นซึ่งต้องใช้เวลา 19 ไซเคิลซึ่งได้จากสัญญาณจากออสซิลเลเตอร์นับตั้งแต่เวลา S5P2 ในไซเคิลขอเครื่องที่พบสัญญาณรีเซท ในระหว่างเวลา 19 ไซเคิลนี้ยังคงมีการ Fetch คำสั่งเข้าไปทำงานได้อยู่

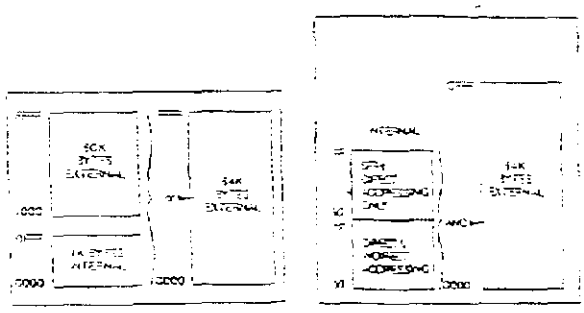
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สภาวะของสัญญาณลอจิกที่ขา RST จะถูกอ่านเข้าไปตรวจสอบที่เวลา S5P2 ของทุกๆ ไซเคิลของเครื่อง ดังนั้นถึงแม้ว่าสัญญาณที่ขา RSTจะมีลอจิกเป็น 1 มาก่อนก็จะไม่เกิดการตรวจสอบสัญญาณรีเซท ดังในรูปที่ 3.16 สัญญาณที่ขา RST อาจเป็น 1 มาตั้งแต่ State ที่ 6 ก็จะไม่เกิดอะไรขึ้นจนกระทั่ง 1 ไซเคิลของออสซิลเลเตอร์ต่อมาซึ่งเป็นเวลา S5P2 จึงจะเกิดการตรวจสอบสัญญาณที่ขา RST ถ้าคำสั่งนั้นมีการทำงานมากกว่า 1 ไซเคิลของเครื่อง 89C51 ก็จะต้องทำงานในคำสั่งนั้นให้เสร็จสิ้นเสียก่อนจึงจะเริ่มการรีเซทได้ โดย 89C51 จะดูสภาวะของสัญญาณที่ขา RST ของ S5P2 ในไซเคิลของเครื่องสุดท้ายเท่านั้น ดังนั้นใน S5P2 ไซเคิลเครื่องแรก ๆ ในคำสั่งอาจมีสภาวะลอจิกที่ขา RST เป็น 1 แต่ที่ S5P2 ของไซเคิลเครื่องสุดท้าย มีสภาวะลอจิกที่ขา RST เป็น 0ก็จะไม่เกิดการรีเซทขึ้นที่เวลา S5P2 เมื่อตรวจสอบสภาวะสัญญาณที่ขา RST แล้วพบว่าเป็น 1 จะต้องรอไปจนถึงเวลา S4P2 ที่ตรวจพบสัญญาณ RST มีลอจิกเป็น 1 จนถึง S4P2 ของไซเคิลเครื่องถัดไปจะยังคงมีการ Fetch คำสั่งเข้าไปทำงานอีก 2 คำสั่ง เมื่อสัญญาณรีเซทภายในเปลี่ยนเป็น 1 ก็จะเริ่มการรีเซท โดยการเขียนข้อมูล 0 ไปยัง Special Function Register ทุกตัวยกเว้นพอร์ท 0 ถึงพอร์ท 3 Stack Pointer และรีจิสเตอร์ SBUF ดังตารางในรูปที่ 3.7 ระหว่างนี้ข้อมูลใน RAM ภายใน 89C51 จะไม่เปลี่ยนแปลงในระหว่างการเขียนข้อมูลลงไปยัง SFR จะยังมีการ Fetch คำสั่งเข้ามาทำงานอีก 1 คำสั่งจนกว่าจะถึง S3P1 ของไซเคิลที่ 2 (นับแต่ไซเคิลของเครื่องที่ตรวจพบลอจิก 1 ที่ขา RST) ก็จะทำให้สภาวะลอจิกที่ขา ALE และ PSEN ค้างอยู่ที่สภาวะลอจิก 1 และจะเป็นอย่างนี้ไปจนกว่าสภาวะลอจิกที่ขา RST เป็น 0 เวลาตั้งแต่พบสัญญาณลอจิก 1 ที่ขา RST ที่เวลา S5P2 จนถึงเวลาที่ ALE และ PSEN ค้างอยู่ที่ 1 จะเท่ากับ 19 ไซเคิลของออสซิลเลเตอร์เมื่อสัญญาณที่ขา RST ถูกเปลี่ยนกลับเป็นลอจิก 0 89C51 จะรออีก 1 ถึง 2 ไซเคิลของเครื่องสัญญาณ ALE และ PSEN จะเริ่มเปลี่ยนแปลงเพื่อเริ่มกระบวนการ Fetch คำสั่งเข้าไปทำงานเริ่มจากคำสั่งในหน่วยความจำสำหรับโปรแกรมตำแหน่ง 0000H

3.5 รีจิสเตอร์ของ 89C51

หน่วยความจำของ 89C51 แบ่งออกเป็น 2 แบบ คือ หน่วยความจำสำหรับโปรแกรม (Program Area) และหน่วยความจำสำหรับเก็บข้อมูล (Data Area) ดังแสดงในไดอะแกรมรูปที่ 3.17

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 ไดอะแกรมภาพของหน่วยความจำ 89C51

หน่วยความจำสำหรับโปรแกรมเป็นหน่วยความจำที่ 89C51 โดยจะใช้สำหรับโปรแกรมภาษาเครื่องที่ 89C51 จะทำงานเมื่อเริ่มป้อนไฟเลี้ยงให้ 89C51 หรือมีการรีเซท (Reset) 89C51 จะทำให้เริ่มการทำงานจากคำสั่งในโปรแกรมตำแหน่งที่ 0000H เมื่อทำงาน 1 คำสั่งก็จะทำให้รีจิสเตอร์ PC ที่ชี้ตำแหน่งโปรแกรมมีค่าเพิ่มขึ้นเพื่อชี้ตำแหน่งของคำสั่งต่อไป ตำแหน่งสุดท้ายของหน่วยความจำซึ่งก็คือ FFFFH โดยหน่วยความจำสำหรับโปรแกรมนี้สามารถที่จะเลือกได้ว่าเป็นหน่วยความจำที่อยู่ภายใน 89C51 หรือภายนอก 89C51 ก็ได้ หน่วยความจำสูงสุดสำหรับโปรแกรมภายนอก 89C51 มีได้ถึง 64 KByte ทำให้สามารถใช้งานได้อย่างกว้างขวาง หน่วยความจำในช่วงนี้ 89C51 สามารถอ่านข้อมูลได้อย่างเดียว ไม่สามารถเขียนข้อมูลได้ระหว่างการทำงาน

หน่วยความจำสำหรับข้อมูลเป็นหน่วยความจำที่ 89C51 ใช้สำหรับเก็บข้อมูลหรือพักข้อมูลระหว่างที่ทำงานหน่วยความจำสำหรับข้อมูลมี 2 แบบ แบบที่หนึ่ง คือมีขนาด 128 ไบท์อยู่ภายใน 89C51 หน่วยความจำอีกแบบหนึ่งจะมีขนาด 64 กิโลไบท์ (Kbyte) ต้องต่อเพิ่มเติมเข้าไปภายนอก 89C51 หน่วยความจำภายในตำแหน่ง 0 ถึง 7FH นี้สามารถอ้างถึงได้โดยตรงคือมีการสั่งให้อ่านหรือเขียนข้อมูลไปยังตำแหน่งนั้นได้โดยตรง แต่หน่วยความจำตำแหน่ง 80H ถึง FFH นั้นเป็นรีจิสเตอร์ฟังก์ชั่นพิเศษ (Special Function Register , SFR) หน่วยความจำภายในช่วงนี้ใช้เป็นรีจิสเตอร์สำหรับงานเฉพาะอย่าง

หน่วยความจำสำหรับข้อมูลภายใน 89C51 ช่วง 00H ถึง 07FH สามารถแบ่งออกได้เป็น 3 กลุ่มคือ

1. Register bank 0.3 อยู่ในหน่วยความจำช่วงตำแหน่ง 00H ถึง 1FH หน่วยความจำนี้แบ่งออกเป็น 4 ชุด ชุดละ 8 ไบท์ แต่ละชุดเราเรียกว่า BANK แต่ละไบท์ใน 1 BANK จะมีชื่อของรีจิสเตอร์ว่า R0,R1,R2,R3,R4,R5,R6 และ R7 รีจิสเตอร์เหล่านี้จะรียกใช้งานในระหว่างการทำงานของโปรแกรมได้สะดวก และรีจิสเตอร์เหล่านี้จะเป็นชื่อซ้ำกันในทุก BANK การใช้งานจึงต้องเรียกใช้งานทีละ BANK เท่านั้น โดยการกำหนดให้รีจิสเตอร์ PSW ที่จะกล่าวถึงต่อไปในบทนี้ เมื่อมีการ Reset การทำงานของ 89C51 จะเริ่มการใช้งานรีจิสเตอร์ R0 ถึง R7 ที่ BANK 0 ซึ่งรีจิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

R0 ถึง R7 ในแต่ละ BANK นั้นจะอ้างอิงในหน่วยความจำสำหรับข้อมูลภายใน 89C51 ดังตารางที่ 2.2

ตัวอย่าง เมื่อกำลังมีการใช้งานในหน่วยความจำ BANK 1 และมีการอ้างรีจิสเตอร์ R7 เช่น คำสั่ง MOA A,R7 (รหัสภาษาเครื่อง EFH)

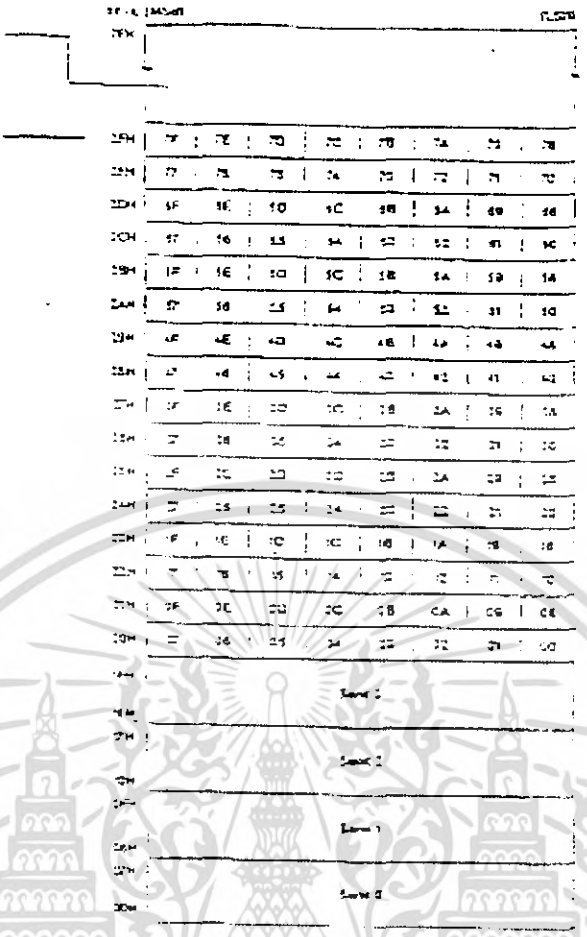
การทำงานของคำสั่งนี้คือการเอาข้อมูลจากตำแหน่ง FH ของหน่วยความจำภายใน 89C51 ไปไว้ยังรีจิสเตอร์ A นั่นเอง

2. Bit Address Area เป็นหน่วยความจำในช่วงตำแหน่ง 20H ถึง 2FH หน่วยความจำแต่ละบิต ในช่วงของหน่วยความจำดังกล่าวจะสามารถตรวจสอบหรือตั้งค่าเป็น 1 หรือ 0 ได้โดยการโปรแกรมภาษาเครื่อง แต่ละบิตของหน่วยความจำช่วงนี้ จะมีค่าของตำแหน่งดังใน Memory Map รูปที่ 3.18 เช่นบิตที่ 7 ของหน่วยความจำในตำแหน่ง 2FH จะมีค่าตำแหน่งเป็น 7FH นั่นเอง

ตารางที่ 3.1 แสดงการอ้างอิงหน่วยความจำสำหรับข้อมูลภายใน

รีจิสเตอร์	ตำแหน่งหน่วยความจำ			
	BANK 0	BANK 1	BANK 2	BANK 3
R0	0	8	10	18
R1	1	9	11	19
R2	2	A	12	1A
R3	3	B	13	1B
R4	4	C	14	1C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 ค่าตำแหน่งของแต่ละบิต

ในรูปที่ 3.18 ตัวเลขทางซ้ายเป็นค่าตำแหน่งของหน่วยความจำสำหรับข้อมูลภายใน 89C51 ซึ่งแต่ละบิตในตำแหน่งนั้นจะมีค่าเป็นเลขฐาน 16 ที่จะใช้เป็นค่าอ้างอิงในคำสั่งจัดการกับข้อมูลบิตนั้น

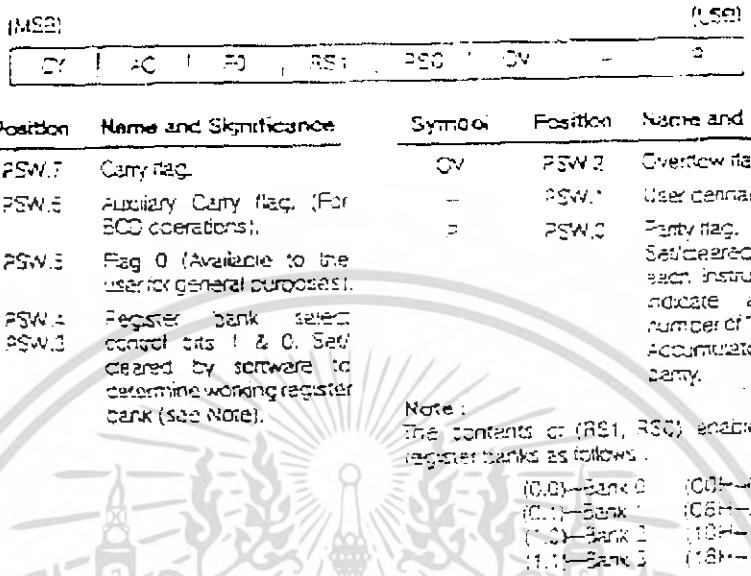
3. Scratched Pod Area เป็นช่วงของหน่วยความจำตำแหน่ง 30H ถึง 7FH หน่วย ความจำช่วงนี้จะใช้สำหรับเก็บข้อมูลทั่วไป ถ้ารีจิสเตอร์ Stack Pointer ชี้มายังหน่วยความจำช่วงนี้จะต้องระวังไม่ให้เกิดการเขียนทับของข้อมูลอันจะทำให้การทำงานของโปรแกรมผิดพลาดได้

ที่กล่าวมาแล้วคงจะพอเข้าใจถึงลักษณะการจัดการหน่วยความจำของ 89C51 ได้พอที่จะเริ่มเขียนโปรแกรมภาษาแอสเซมบลีของ 89C51 สรุปชุดคำสั่งของ 89C51 จะอยู่ในภาคผนวก ก

จากตารางคำสั่งในหน้า 12-13 ของภาคผนวก ก จะมีคำอธิบายการใช้งานหรือการทำงานของแต่ละคำสั่งใน 89C51 ไว้ด้วย คำสั่งของ 89C51 เป็นคำสั่งที่ประสิทธิภาพการทำงานสูงมาก ในขณะที่ 89C51 ทำงานจะมีรีจิสเตอร์ตัวหนึ่งที่เก็บสภาวะ (Flag) ที่เกิดขึ้นระหว่างการคำนวณ เช่น ตัวทด (Carry) หรือจะใช้เลือก BANK ของรีจิสเตอร์ภายใน 8051 ก็ได้ รีจิสเตอร์นี้คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Program Status Word (PSW) มีขนาน 8 บิต แต่ละบิตจะใช้เก็บสถานการณ์ทำงานต่าง ๆ ไว้ดังรูปที่ 3.19



รูปที่ 3.19 Program Status Word (PSW)

PSW.0 บิต 0 เรียกว่าบิตพาริตี้บิตนี้จะบอกว่าในรีจิสเตอร์ Accumulator หรือรีจิสเตอร์ A มี 1เป็นจำนวนคี่หรือคู่ เช่นในรีจิสเตอร์ A ขนาด 8 บิตมี 1 อยู่ 3 ตัวและมี 0 อยู่ 5 ตัวก็จะทำให้บิต PSW.0 นี้มีค่าเป็น 1 ถ้าใน Accumulator มี 1 อยู่เป็นจำนวนคู่ก็จะทำให้บิตนี้มีค่าเป็น 0

PSW.1 บิต 1 บิตนี้ไม่มีการใช้งาน

PSW.2 บิต 2 เรียกว่า Overflow Flag เป็นบิตที่บอกการคำนวณนั้นทำให้เกิดตัวทดขึ้นในระหว่างการคำนวณ ตัวทดนี้เป็นตัวทดที่เกิดจากบิต 6 ไปยังบิต 7 มีประโยชน์เมื่อทำการคำนวณแบบ Signed Integer

PSW.3 ,PSW.4 บิต 3และ4 2 บิตนี้จะใช้งานร่วมกันเพื่อเป็นตัวบอกว่าขณะนี้ใช้รีจิสเตอร์ R0 ถึง R7 ใน BANK ได้ดังตารางที่ 3.2

ตารางที่ 3.2 แสดงค่ารีจิสเตอร์ R0 ถึง R7 ใน BANK

บิต 4 (RB1)	บิต 3 (RB0)	Register bank	Address
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่างเช่น บิต 4 และบิต 3 มีค่าเป็น 10 เป็นการเลือกรีจิสเตอร์ BANK 2 หมายความว่าในรหัสคำสั่งช่วยจำที่อ้างอิงถึง R0 ก็จะอ้างถึงหน่วยความจำภายในที่ตำแหน่ง 10H

PSW.5 บิต 5 เรียกว่าบิตอเนกประสงค์เป็นบิตที่ผู้ใช้คำสั่งกำหนดค่าให้เป็น 0 หรือ 1 ก็ได้ โดยที่การทำงานของคำสั่งอื่นจะไม่ทำให้บิตนี้มีค่าเปลี่ยนแปลง บิตนี้มีประโยชน์สำหรับในการส่งสถานะของโปรแกรมระหว่างเรียกการทำงานของโปรแกรมย่อย (Subroutine)

PSW.6 บิต 6 เรียกว่า Auxiliary Carry Flag เป็นบิตที่ใช้สำหรับเก็บตัวทดที่เกิดขึ้นระหว่างการคำนวณ โดยตัวทดนี้เป็นตัวทดที่เกิดการคำนวณของบิต 3 ข้ามไปยังบิต4

PSW.7 บิต 7 เรียกว่า Carry Flag เป็นบิตที่บอกสถานการณ์คำนวณทางคณิตศาสตร์ว่าผลลัพธ์นั้นทำให้เกิดตัวทดขึ้นหรือไม่ เช่นการบวกเลขของ 2 จำนวนเข้าด้วยกันแล้วผลลัพธ์มีค่ามากกว่า 255 ก็จะทำให้เกิดตัวทดขึ้น เนื่องมาจากว่า Accumulator ที่ทำการบวกนี้สามารถเก็บข้อมูลได้เพียง 8 บิตเท่านั้น และทำให้บิตนี้มีค่าเป็น 1

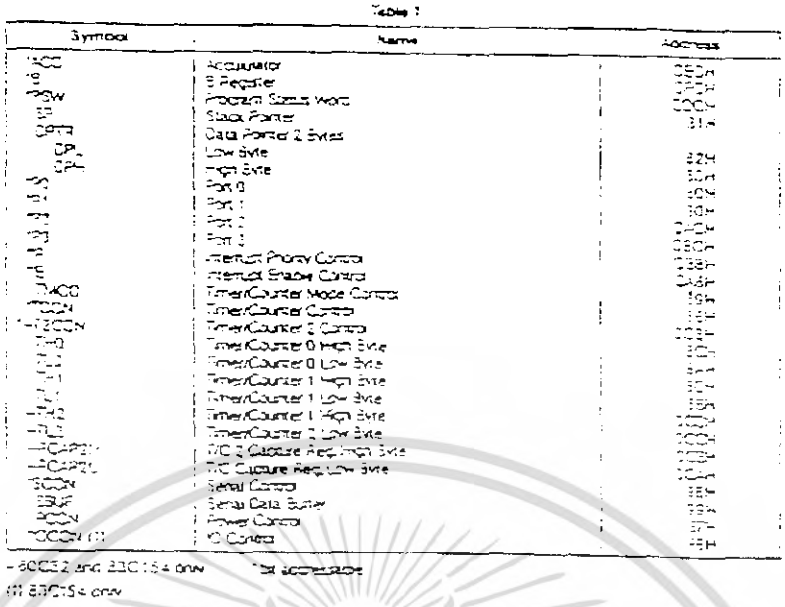
3.6 พื้นที่หน่วยความจำข้อมูลแฝงของ AT89C52

เป็นพื้นที่ที่อยู่ซ้อนทับกับค่าตำแหน่งเดียวกับ หมายเลขรีจิสเตอร์ฟังชั่นพิเศษเช่นเดียวกับไมโครคอนโทรลเลอร์ 8051 การเข้าถึงจะสามารถเข้าถึงโดยอ้อม (indirect) มีขนาด 128 ไบต์อยู่ที่ตำแหน่ง 80H-FFH

3.7 รีจิสเตอร์ฟังก์ชั่นพิเศษ (Special Function Register ,SFR)

ใน 89C51 จะใช้วิธีการกำหนดชื่อให้กับตำแหน่งของหน่วยความจำ สำหรับข้อมูลภายใน (Internal Data Memory) ที่เรียกว่า Symbolize เช่น การให้ชื่อหน่วยจำแต่ละตำแหน่งในแต่ละ BANK ซึ่งอยู่ในช่วงหน่วยความจำตำแหน่ง 00H ถึง 1FH แล้วในคำสั่งจะอ้างอิงหน่วยความจำแต่ละตำแหน่งโดยการใช้ชื่อ R0,R1,R2,R3,R4,R5,R6 และ R7 หน่วยความจำตำแหน่งเหล่านี้ จะเรียกอีกอย่างหนึ่งว่ารีจิสเตอร์ ซึ่งมีหน้าที่ในการเก็บหรือพักข้อมูล หรือใช้สำหรับการกระทำบางอย่าง รีจิสเตอร์บางอย่างใน 89C51 ที่เรียกว่า Special Function Register (SFR) เป็นรีจิสเตอร์ที่ใช้สำหรับงานเฉพาะ คือข้อมูลที่นำไปเก็บไว้รีจิสเตอร์เหล่านี้จะมีความหมายเฉพาะตัวของรีจิสเตอร์ที่แต่ละตำแหน่งของ SFR อาจจะไม่ใช่เป็นหน่วยความจำ (RAM) แต่อาจเป็นตัวนับ (Register Counter) , Shift Register หรือ Latch ซึ่งการอ้างอิงข้อมูลในแต่ละตำแหน่งนั้น 89C51 จะถือเสมือนว่าเป็นหน่วยความจำตำแหน่งหนึ่ง จึงเรียกการมองข้อมูลแต่ละตำแหน่งนี้ว่า Memory Map I/O รีจิสเตอร์กลุ่มนี้มีดังในรูป ที่ 3.20

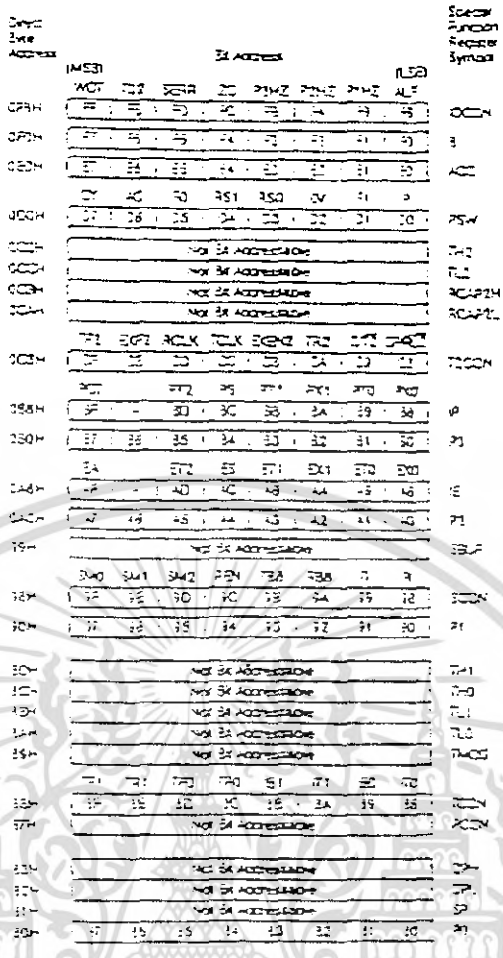
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาพที่ 3.20 Special Function Register (SFR)

ในรูปที่ 3.20 ช่อง Symbol ทางซ้ายจะเป็นสัญลักษณ์ของรีจิสเตอร์ ในช่องถัดมาคือชื่อของรีจิสเตอร์ตามสัญลักษณ์ที่อยู่ทางซ้าย ส่วนในช่องขวาสุดจะเป็นตำแหน่งของหน่วยความจำสำหรับข้อมูลภายใน 89C51 ที่แทนด้วยชื่อหรือสัญลักษณ์ทางซ้ายนั่นเอง เช่น ในบรรทัดแรกคือรีจิสเตอร์ชื่อ Accumulator ที่มีสัญลักษณ์ ACC รีจิสเตอร์นี้คือ หน่วยความจำสำหรับข้อมูลภายใน 89C51 ที่ตำแหน่ง 0E0H การอ่านหรือเขียนข้อมูลกับรีจิสเตอร์เหล่านี้ สามารถทำได้โดยการใช้คำสั่งในกลุ่มการย้ายข้อมูล (เช่น MOV A ,#25H หรือ MOV 0E0H,#25H) และรีจิสเตอร์บางตัวในกลุ่มนี้ ยังสามารถใช้คำสั่งในกลุ่ม Boolean Instruction เพื่อการทำงานในแต่ละบิตในรีจิสเตอร์เหล่านั้นได้ จากตารางในรูปที่ 3.20 รีจิสเตอร์ที่มีเครื่องหมาย * อยู่ข้างหน้าสามารถจะใช้คำสั่งกลุ่ม Boolean Instruction จัดการกับแต่ละบิตได้ รีจิสเตอร์ที่มีเครื่องหมาย + นำหน้าหมายความว่า รีจิสเตอร์นั้นมีเฉพาะใน 89C51 และ 89C154

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 แผนภาพค่าตำแหน่งหน่วยความจำแต่ละบิต

รูปที่ 3.21 ในช่องสี่เหลี่ยมเล็ก ๆ จะเป็นตำแหน่งของบิตนั้นในแต่ละรีจิสเตอร์ เช่นในช่องซ้ายของรีจิสเตอร์ TCON ซึ่งเป็นตำแหน่งค่าบิต 7 ของหน่วยความจำ 88H ถ้าต้องการให้บิตนี้มีค่าเป็น 0 ก็สามารถทำได้โดยใช้คำสั่ง

CLR 8FH หรือจะทำให้เป็น 1 ก็ทำได้โดยคำสั่ง SETB 8FH Special Function Register

รีจิสเตอร์ในกลุ่ม Special Function Register **มีดังนี้**

1. Accumulator ตำแหน่งความจำภายในเท่ากับ 0E0H

รีจิสเตอร์นี้มีขนาด 8 บิต เป็นรีจิสเตอร์ที่ใช้มาก ซึ่งในรหัสคำสั่งช่วยจำจะอ้างถึงรีจิสเตอร์นี้โดยใช้สัญลักษณ์ A เช่น MOV A,#15H คำสั่งที่อ่านหรือเก็บข้อมูลกับหน่วยความจำภายนอกจะต้องกระทำผ่านรีจิสเตอร์นี้เท่านั้น เช่น MOVX @R0,A **หรือ** MOVX A,@R0 **เป็นต้น และข้อมูลที่อยู่ภายในรีจิสเตอร์นี้ ก็สามารถที่จะให้โปรแกรมตรวจสอบเพื่อกระโดดการทำงานไปยังตำแหน่งอื่นได้ เช่น** JZ rel

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. B Register ตำแหน่งหน่วยความจำภายในเท่ากับ 0F0H

เป็นรีจิสเตอร์ขนาด 8 บิต ที่ใช้ในคำสั่งการคูณ (MUL AB) และคำสั่งการหาร (DIV AB) เท่านั้น โดยรีจิสเตอร์ B นี้จะเก็บตัวคูณและผลลัพธ์บิต 8 ถึง 5 ในคำสั่งการคูณ ส่วนในคำสั่งการหารนั้นรีจิสเตอร์ B จะเก็บตัวหารและผลการหาร การเขียนข้อมูลไปยังรีจิสเตอร์นี้จะต้องใช้คำสั่งเคลื่อนย้ายข้อมูลไปยังตำแหน่ง 0F0H เช่น MOV 0F0H , 25H จะเป็นการกำหนดค่า 25H ให้กับรีจิสเตอร์ B

3. Program status word ตำแหน่งหน่วยความจำภายในเท่ากับ 0D0H

เป็นรีจิสเตอร์ขนาด 8 บิต ที่แต่ละบิตจะบอกสภาวะต่าง ๆ แต่ละบิตของ PSW จะสามารถกำหนดให้เป็น 1 หรือ 0 ได้ด้วยคำสั่ง SETB หรือ CLR ตามลำดับค่าตำแหน่งบิต 0 ถึงบิต 7 ของรีจิสเตอร์ PSW เท่ากับ D0H ถึง D7H ตามลำดับ

4. Stack Pointer ตำแหน่งหน่วยความจำในเท่ากับ 081H

เป็นรีจิสเตอร์ขนาด 8 บิต รีจิสเตอร์นี้จะใช้ตำแหน่งหน่วยความจำภายใน 89C51 ที่ใช้เก็บตำแหน่ง (Address) เดิมของโปรแกรมก่อนทำงานคำสั่ง CALL หรือตำแหน่งที่จะใช้เก็บข้อมูลด้วยคำสั่ง PUSH และ ตำแหน่งที่อ่านข้อมูลออกมาในคำสั่ง POP เมื่อทำการรีเซท 89C51 โดยการป้อนสัญญาณสภาวะลอจิก 1 เข้าไปที่ขา RST ของ 89C51 จะทำให้ข้อมูลในรีจิสเตอร์นี้มีค่าเป็น 07H หมายความว่ารีจิสเตอร์ SP ชี้หน่วยความจำภายใน 89C51 ที่ตำแหน่ง 07H ค่าของ SP จะเปลี่ยนแปลงไปโดยการใช้คำสั่งเคลื่อนย้ายข้อมูลหรือการทำงานของคำสั่ง PUSH , POP และ CALL

5. Data Pointer Register ตำแหน่งหน่วยความจำภายในเท่ากับ 82H และ 83H

รีจิสเตอร์ DTPR มีขนาด 16 บิต หน้าที่ของรีจิสเตอร์นี้ก็คือใช้สำหรับชี้ตำแหน่งในหน่วยความจำรีจิสเตอร์ DPTR นี้สามารถใช้อ้างถึงตำแหน่งหน่วยความจำได้สูงสุด 60x1024 ตำแหน่ง เช่นคำสั่ง MOVX A,@DPTR หรือใช้ชี้ตำแหน่งโปรแกรมที่ต้องการกระโดดข้ามไปทำงาน เช่น คำสั่ง JMP @A+DPTR รีจิสเตอร์ DPTR นี้ประกอบด้วยรีจิสเตอร์ขนาด 8 บิต 2 ตัวคือ DPH ซึ่งอยู่ที่ตำแหน่ง 83H และ DPL ซึ่งอยู่ตำแหน่ง 82H หน่วยความจำสำหรับข้อมูลภายใน 89C51 ดังนั้นการแก้ไขข้อมูลในรีจิสเตอร์ DPTR จึงทำได้ทั้งทีละ 16 บิต เช่นคำสั่ง MOV DPTR,#DATA16 หรือจักการทีละ 8 บิตโดยการแก้ไขข้อมลใน DPH หรือ DPL ด้วยคำสั่ง MOV 83H,#DATA8 หรือ MOV 83H,#DATA8

6. PORT 0 ถึง 3 ตำแหน่งหน่วยความจำภายในเท่ากับ 80H,90H,0A0H,0B0H

Special Function Register ชื่อ P0,P1,P2 และ P3 เป็นรีจิสเตอร์ขนาด 8 บิตของหน่วยความจำสำหรับข้อมูลภายใน 89C51 ที่ตำแหน่งเป็นการส่งข้อมูลไปยังพอร์ทนั้น ๆ ของ 89C51 ข้อมูลที่เขียนออกไปจะถูก LATCH ค้างไว้และปรากฏที่แต่ละบิตของพอร์ท เช่น MOV 80H,#18H จะปรากฏสภาวะลอจิก LLLHHLLL ที่ขาบิต 7 ถึง 0 ของพอร์ท 0 ตามลำดับ ในการอ่าน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลจากรีจิสเตอร์แต่ละตัวก็จะเป็นการอ่านสภาวะลอจิกของสัญญาณ ที่ปรากฏอยู่ที่แต่ละขาของพอร์ทนั้น เช่น MOV A,80H เป็นการอ่านสภาวะลอจิกจากพอร์ท 0 เข้ามายัง Accumulator การอ่านข้อมูลจากพอร์ทจะต้องเขียนข้อมูล 11111111B ไปไว้ที่พอร์ทนั้น ๆ เสียก่อน ทุกบิตของพอร์ท 0 ถึง 3 จะสามารถแก้ไขเปลี่ยนแปลงได้โดยใช้คำสั่ง SETB bit และ CLR bit

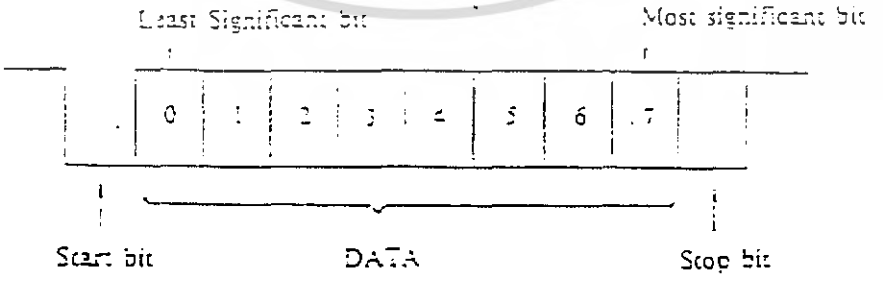
7. Serial Data Buffer ตำแหน่งหน่วยความจำภายในเท่ากับ 99H

รีจิสเตอร์นี้มีขนาด 8 บิตและมีตำแหน่งของหน่วยความจำสำหรับข้อมูลภายใน 89C51 เท่ากับ 99H โครงสร้างภายในแล้วรีจิสเตอร์นี้มี 2 ตัว ที่มีชื่อเดียวกัน ตัวหนึ่งสำหรับเก็บข้อมูลที่จะส่งแบบอนุกรมออกจาก 89C51 และอีกตัวหนึ่งสำหรับข้อมูลแบบอนุกรมที่เข้ามา ดังนั้น Serial Port ของ 89C51 จึงเรียกว่ามีการทำงานแบบ Full Duplex เพราะสามารถส่งและรับข้อมูลได้ในเวลาเดียวกันเนื่องจากรีจิสเตอร์สำหรับรับส่งและรับแยกออกจากกัน ข้อมูลที่ต้องการจะส่งออกก็ให้เขียนไปยังรีจิสเตอร์ SBUF และสั่งงานให้ส่งข้อมูลออกมา ข้อมูลในรีจิสเตอร์จะเริ่มส่งออกโดยเริ่มจากบิต 0 ถึง 7 ตามลำดับ ถ้าข้อมูลเข้ามาทางขา RXD ก็จะถูกเก็บไปไว้ในรีจิสเตอร์นี้โดยถือว่าข้อมูลบิตแรกที่เข้ามาคือบิต 0

Serial Port จะสามารถกำหนดให้การทำงานรับส่งข้อมูลแบบอนุกรมได้ 4 โหมด (MODE) โดยการกำหนดในรีจิสเตอร์ SCON (Serial Port Control Register) ซึ่งจะอธิบายต่อไปในข้อ 5.18 แต่ละโหมดการทำงานของ Serial Port มีดังนี้

MODE 0 : ในโหมดนี้ จะมีการรับหรือส่งข้อมูลแบบอนุกรมทางขา RXD และขา TXD จะส่งสัญญาณ Clock ที่ใช้สำหรับเลื่อน (Shift) ข้อมูล 1 ชุด ของข้อมูลจะประกอบด้วยข้อมูล 8 บิต เท่านั้น และจะเริ่มการรับ-ส่งข้อมูลจากบิต 0 จนถึงบิต 7 ตามลำดับ อัตราการส่งข้อมูลแบบอนุกรมจะเท่ากับ 1/12 เท่าของความถี่สัญญาณนาฬิกาที่ใช้กับ 89C51

MODE 1 : ข้อมูลที่รับ-ส่ง 1 ชุดในโหมดนี้จะมี 10 บิต ผ่านทางขา RXD และ TXD ตามลำดับเริ่มต้นการรับส่งข้อมูลด้วย Start bit 1 บิต (ลอจิกเป็น 0) ,ข้อมูล 8 บิต (เริ่มจากบิต 0) , Stop bit 1 บิต (ลอจิก 0) การส่งข้อมูลโหมดนี้มีดังรูปที่ 3.22

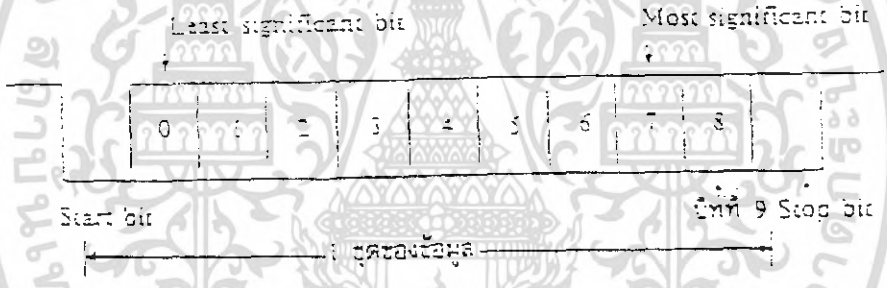


รูปที่ 3.22 ชุดข้อมูลอนุกรมในโหมด 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อรับข้อมูลอนุกรมเข้ามาข้อมูล 8 บิตจะถูกเก็บในรีจิสเตอร์ SBUF และ Stop bit จะถูกเก็บไปที่บิต RB8 ในรีจิสเตอร์ SCON ในการส่งข้อมูลออกก็จะเขียนข้อมูลที่ต้องการส่งไปยังรีจิสเตอร์ SBUF อัตราการส่งข้อมูลในโหมดนี้ สามารถกำหนดได้ตามต้องการโดยจะขึ้นกับการเกิด Overflow ใน Timer 1

MODE 2 : การรับ-ส่งข้อมูลของโหมด 2 1ชุดจะมี 11 บิต ข้อมูลจะส่งออกผ่านทางขา TXD และรับเข้ามาทางขา RXD ข้อมูลแต่ละชุดจะเริ่มต้นด้วย Start bit 1 บิต , ข้อมูล 8 บิต (เริ่มจากบิต 0) ,ข้อมูลบิตที่ 9 จำนวน 1 บิตและ Stop bit อีก 1 บิดข้อมูลบิตที่ 9 ที่จะส่งออกนี้สามารถกำหนดได้ว่าจะให้เป็น 1 หรือ 0 พาริตี้ของข้อมูลไปเป็นบิตที่ 9 เพื่อว่าปลายทางรับข้อมูลแล้วจะได้ใช้ตรวจสอบว่าข้อมูลที่รับเข้ามา 8 บิตมีพาริตี้บิตตรงกับบิตที่ 9 หรือไม่ ถ้าไม่ตรงก็แสดงว่ามีข้อผิดพลาดเกิดขึ้นระหว่างการส่งข้อมูลเข้ามานั้นข้อมูลบิตที่ 9 ก็จะถูกนำไปเก็บในบิต RB8 ของรีจิสเตอร์ SCON ชุดข้อมูลที่รับ-ส่งจะมีดังรูปที่ 3.23



รูปที่ 3.23 ชุดข้อมูลอนุกรมในโหมด 2

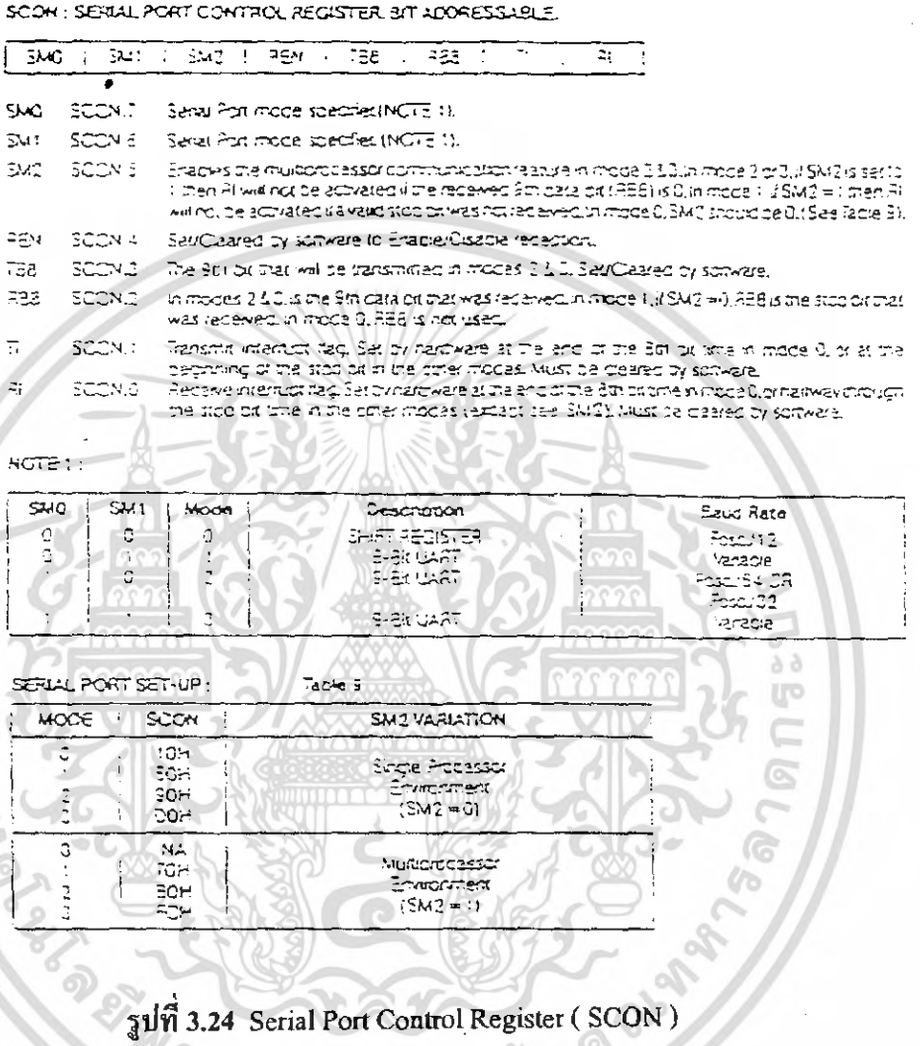
อัตราการส่งข้อมูลจะกำหนดให้เป็น 1/32 หรือ 1/64 เท่า ของความถี่สัญญาณนาฬิกาที่ใช้กับ 89C51 โดยการกำหนดบิต SMOD ในรีจิสเตอร์ PCON

MODE 3 : การส่งข้อมูลโหมดนี้ 1 ชุดมี 11 บิต เหมือนกับโหมด 2 ทุกประการแตกต่างกันตรงอัตราการส่งข้อมูลเท่านั้น คืออัตราการส่งข้อมูลในโหมด 3 นี้ สามารถกำหนดได้ตามต้องการโดยจะขึ้นกับการเกิด Overflow ใน Timer 1 เหมือนกับโหมด 1

8. SCON (Serial Port Control Register) ตำแหน่งหน่วยความจำภายในเท่ากับ 98H รีจิสเตอร์ SCON มีขนาด 8 บิต ใช้สำหรับควบคุมการส่งและรับข้อมูลผ่านทาง Serial Port แต่ละบิตของข้อมูลในรีจิสเตอร์นี้มีความหมายเฉพาะดังรูปที่ 3.24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8. SCON (Serial Port Control Register) ตำแหน่งหน่วยความจำภายในเท่ากับ 98H รีจิสเตอร์ SCON มีขนาด 8 บิต ใช้สำหรับควบคุมการส่งและรับข้อมูลผ่านทาง Serial Port แต่ละบิตของข้อมูลในรีจิสเตอร์นี้มีความหมายเฉพาะดังรูปที่ 3.24



SCON : SERIAL PORT CONTROL REGISTER, BIT ADDRESSABLE.

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0	SCON.7	Serial Port mode specifier (NOTE 1).
SM1	SCON.6	Serial Port mode specifier (NOTE 1).
SM2	SCON.5	Enables the multiprocessor communication feature in mode 2 & 3. In mode 2 or 3, if SM2 is set to 1 then RI will not be activated if the received 9th data bit (RB8) is 0. In mode 1, if SM2 = 1 then RI will not be activated if a valid stop bit was not received. In mode 0, SM2 should be 0. (See Table 3).
REN	SCON.4	Set/Cleared by software to Enable/Disable reception.
TB8	SCON.3	The 9th bit that will be transmitted in modes 2 & 3. Set/Cleared by software.
RB8	SCON.2	In modes 2 & 3, is the 9th data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
TI	SCON.1	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or at the beginning of the stop bit in the other modes. Must be cleared by software.
RI	SCON.0	Receive interrupt flag. Set by hardware at the end of the 8th bit time in mode 0, or halfway through the stop bit time in the other modes (except see SM2). Must be cleared by software.

NOTE 1 :

SM0	SM1	Mode	Description	Baud Rate
0	0	0	SHIFT REGISTER	Fosc./12
0	1	1	8-BIT UART	Variable
1	0	2	9-BIT UART	Fosc./64 OR Fosc./32
1	1	3	9-BIT UART	Variable

SERIAL PORT SET-UP : Table 3

MODE	SCON	SM2 VARIATION
0	10H	Single Processor Environment (SM2 = 0)
1	50H	
2	90H	
3	D0H	
0	NA	Multiprocessor Environment (SM2 = 1)
1	70H	
2	B0H	
3	F0H	

รูปที่ 3.24 Serial Port Control Register (SCON)

ในรูปที่ 3.24 บิต R1 จะเป็นชื่อของบิต 0 และ SNO จะเป็นบิต 7 ของรีจิสเตอร์ SCON ซึ่งความหมายหรือการทำงานของแต่ละบิตมีดังนี้

RI Receive Interrupt Flag

บิตนี้จะถูกกำหนดโดยฮาร์ดแวร์ให้มีค่าเป็น 0 หรือ 1 โดยที่ในการรับข้อมูลโหมด 0 นั้น บิต RB8 จะมีค่าเป็น 1 เมื่อมีข้อมูลเข้ามาครบทั้ง 8 บิต ส่วนในโหมดอื่นบิต RB8 จะเป็น 1 ก็ต่อเมื่อข้อมูลเข้ามาถึงเวลาครึ่งหนึ่งของ Stop bit (ยกเว้นบางกรณีให้ดูที่เรื่องบิต SM2 ของรีจิสเตอร์ SCON) บิตนี้จะสามารถ Clear ให้มีค่า เป็น 0 ได้ โดยใช้คำสั่ง CLR bit โดยค่าตำแหน่งของบิตมีค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เท่ากับ 98H บิตนี้มีประโยชน์ให้รู้ว่าข้อมูลได้เข้ามาอยู่ใน SBUF ครบทั้งชุดแล้วพร้อมที่ CPU จะอ่านไปเก็บในหน่วยความจำต่อไป หรืออาจกำหนดค่าในรีจิสเตอร์ IE และ IP เพื่อเมื่อมีข้อมูลเข้ามาทางพอร์ทอนุกรมแล้วจะทำให้เกิดการขัดจังหวะ (Interrupt) การทำงานของโปรแกรมหลัก (Main Program) แล้วกระโดดไปทำงานในโปรแกรมตอบสนองการขัดจังหวะ (Interrupt Service Routine)

TI Transmit Interrupt Flag

ค่าในบิต TI จะถูกกำหนดให้เป็น 1 หรือ 0 ด้วยฮาร์ดแวร์ โดยในการส่งข้อมูลแบบอนุกรมโหมด 0 บิตนี้จะเป็น 1 เพื่อจะบอกว่าการส่งข้อมูลในรีจิสเตอร์ SBUF ออกไปทางพอร์ทอนุกรมครบทั้ง 8 บิต แต่ถ้าเป็นการส่งข้อมูลแบบอนุกรมในโหมดอื่นจะทำให้ข้อมูลในบิต TI เป็น 1 เมื่อเริ่มการส่ง Stop bit ข้อมูลบิตนี้จะสามารถ Clear เป็น 0 ได้ด้วยคำสั่ง CLR bit โดยที่ค่าตำแหน่งของบิตนี้เท่ากับ 99H บิตนี้ยังมีประโยชน์เพื่อบอกว่าการส่งข้อมูลจาก SBUF ออกไปทางพอร์ทอนุกรมนั้นสิ้นสุดแล้วพร้อมที่จะให้โปรแกรมเขียนข้อมูลลงไปยัง SBUF สำหรับการส่งออกไปได้ นอกจากนี้การกำหนดค่ารีจิสเตอร์ IE และ IP ยังสามารถที่จะกำหนดให้เกิดการขัดจังหวะการทำงานของโปรแกรมได้เมื่อบิตนี้ถูกฮาร์ดแวร์ทำให้มีค่าเป็น 1

RB8

เมื่อมีการกำหนดให้รับข้อมูลในโหมด 2 และ 3 จะใช้บิตนี้สำหรับเก็บข้อมูลบิตที่ 9 ที่เข้ามาทางพอร์ทอนุกรม ส่วนในโหมด 1 นั้นบิตนี้จะเก็บ Stop bit ซึ่งมีค่าเป็น1 นั่นเอง ในโหมด 0 บิตนี้จะไม่ถูกใช้งาน ค่าตำแหน่งของบิตนี้คือ 9AH

TB8

ในการส่งข้อมูลแบบอนุกรมโหมด 2 และ 3 จะใช้บิตนี้เก็บข้อมูลบิตที่ 9 ส่วน โหมดอื่นจะไม่ใช้งานบิตนี้การกำหนดค่าในบิตนี้สามารถทำได้โดยใช้คำสั่ง SETB bit หรือ CLR bit ค่าตำแหน่งของบิตนี้คือ 9BH

REN Receive Enable

เป็นบิตที่จะใช้กำหนดให้ทำการรับข้อมูลเข้ามาจากทางพอร์ทอนุกรม (Serial Port) หรือไม่ถ้าบิตนี้เป็น 1 ก็จะรับข้อมูลเข้ามา แต่ถ้าเป็น 0 ก็จะไม่รับข้อมูลที่ขา RXD เข้ามา การให้บิตนี้เป็น1 หรือ 0 ทำได้โดยใช้ คำสั่ง SETB bit หรือ CLR bit ค่าตำแหน่งของบิตนี้คือ 9CH

SM2

เป็นบิตสำหรับควบคุมการทำงานของฮาร์ดแวร์ที่จะทำให้บิตR1 เป็น 1 หรือไม่ในกรณีที่บิต SM2 เป็น 0 ค่าในบิต R1 ก็จะเป็นไปตามที่ได้อธิบายมาแล้วในเรื่องบิต R1 ซึ่งแต่ถ้าบิต SM2=1 โหมด 2 และ 3 ซึ่งปกติแล้วบิต R1 จะเป็น 1 เมื่อข้อมูลบิตที่ 9 เข้ามา แต่เมื่อ SM2 เป็น 1 แล้ว R1 จะเป็น 1 ก็ต่อเมื่อข้อมูลบิตที่ 9 เข้ามามีค่าเป็น 1 ถ้าข้อมูลบิตที่ 9 เข้ามาเป็น 0 จะไม่ทำให้บิต R1 มีค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็น 1 (คือบิต R1 จะเป็น 0)ในโหมด 1 บิต R1 มีค่าเป็น 1 เมื่อข้อมูล Stop bit เข้ามายังพอร์ตอนุกรมถูกต้องแต่ถ้า Stop bit ไม่เข้ามายังพอร์ตอนุกรมอันอาจเกิดจากปัญหาในการส่งข้อมูลแล้ว บิต R1 จะมีค่าเป็น 0

SM0 , SM1

เป็น 2 บิต ที่ใช้งานร่วมกันเพื่อกำหนดโหมดของการรับ-ส่งของพอร์ตอนุกรม ค่าใน 2 บิตนี้จะกำหนดโหมดได้ ดังนี้

ตารางที่ 3.3 แสดงค่าการกำหนดโหมดใน 2 บิต

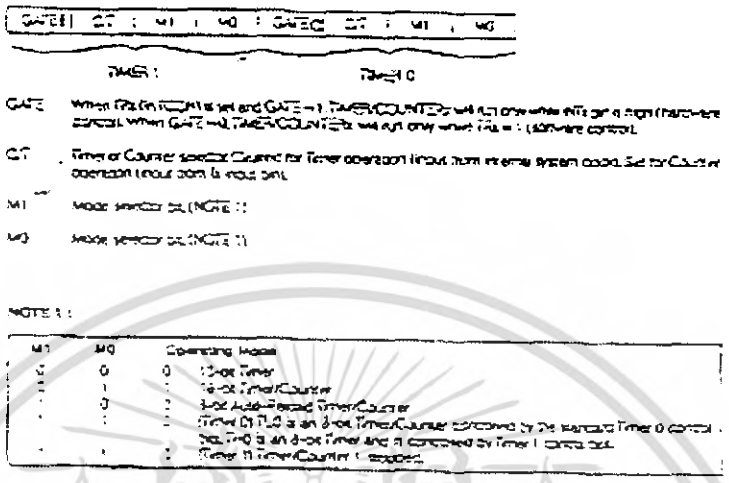
SM0	SM1	MODE	Description
0	0	0	Shift register
0	1	1	8-bit UART
1	0	2	9-bit UART
1	1	3	9-bit UART

การทำงานของแต่ละโหมดจะมีดังในข้อ 5.1.7

9. Timer Register TH0 , TL0 , TH1 , TL1 ตำแหน่งหน่วยความจำภายในเท่ากับ 8CH , 8AH , 8DH , 8BH ใน 89C51 จะมีวงจร Timer อยู่ 2 ชุดคือ Timer 0 และ Timer 1 ใน Timer แต่ละชุดจะมีรีจิสเตอร์ขนาด 8 บิตอยู่ 2 ตัว เพื่อเก็บค่าการนับของ Timer ได้สูงสุดถึง 16 บิตใ Timer 0 รีจิสเตอร์นี้คือ TH0 TL0 และ Timer 1 คือรีจิสเตอร์ TH1 TL1 TLx (x หมายถึง 0 หรือ 1) จะเก็บค่าการนับของ 8 บิตล่าง และ THxจะเก็บค่าการนับของ 8 บิตบน ผู้ใช้จะสามารถการทำงานของวงจร Timer ในโหมด Timer หรือโหมด Counter ได้โดยการกำหนดในรีจิสเตอร์ชื่อ TMOD (Timer/Counter Mode Control Register) การทำงานเป็น Timer นั้นจะให้รีจิสเตอร์ใน Timer 0 หรือ 1 ทำการนับจำนวนไซเคิลของสัญญาณนาฬิกาที่ผ่านวงจรหาร 12 ดังรูปที่ 3.26 เมื่อการนับครบถึงค่าสูงสุดที่รีจิสเตอร์ TLx และ THx จะเก็บได้คือค่าFFFFH แล้วยังนับต่อไปค่าที่ได้จากการนับจะเป็น 0000H ทำให้เกิดการ SET บิตบางบิตในรีจิสเตอร์ TCON เพื่อบอกสภาวะ Timer Overflow นี้ในการให้วงจร Timer ทำงานเป็น Counter ก็คือการใช้รีจิสเตอร์ TLx และ THx ทำการนับจำนวนไซเคิลของสัญญาณที่เข้ามาทางขา T0 หรือ T1 ของ 89C51 สัญญาณที่เขามาทางลอจิก 0 หรือ 1 เป็นแบบ TTL คือลอจิก 0 จะต้องมีโวลเตจไม่เกิน 0.6 V และลอจิก 1 จะต้องมีโวลเตจมากกว่า 2.4 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

10. TMOD Timer / Counter Mode Register ตำแหน่งหน่วยความจำภายในเท่ากับ 89H TMOD เป็นรีจิสเตอร์ขนาด 8 บิต ที่มีหน้าที่ควบคุมการทำงานของ Timer 0 และ Timer1 แต่ละบิตในรีจิสเตอร์นี้มีความหมายเฉพาะดังรูปที่ 3.25



รูปที่ 3.25 TMODTimer/Counter mode Register

ในรูปที่ 3.25 M0 เป็นชื่อของบิต 0 และ GATE ทางซ้ายสุดเป็นชื่อของบิต 7 รีจิสเตอร์นี้บ่งข้อมูลออกเป็น 2 ชุดชุดละ 4 บิต คือบิต0-3 ใช้สำหรับควบคุมการทำงานของ Timer0 แลบิต 4-7 ใช้ควบคุมการทำงานของ Timer 1 หน้าที่ในการควบคุม Timer ของแต่ละบิตที่มีชื่อเดียวกันจะเหมือนกัน

GATEx เป็นบิตที่ใช้ควบคุมให้ Timer x ทำงานหรือไม่ ถ้าบิตนี้ของ Timer x ถูกตั้งเป็น 1 จะทำให้ Timer งานก็ต่อเมื่อที่ขา INTx มีสภาวะลอจิกเป็น 1 และบิท TRx ในรีจิสเตอร์ TCON เป็น 1 ด้วย

$\overline{C/T}$ บิตนี้ใช้สำหรับเอกการทำงานของ Timer ว่าจะใช้เป็น Timer หรือ Counter ถ้าบิตนี้เป็น 1 ก็หมายความว่าเลือกการทำงานเป็น Counter ซึ่งจะนับจำนวนไซเคิลของสัญญาณที่ข้ามาทางขา Tx

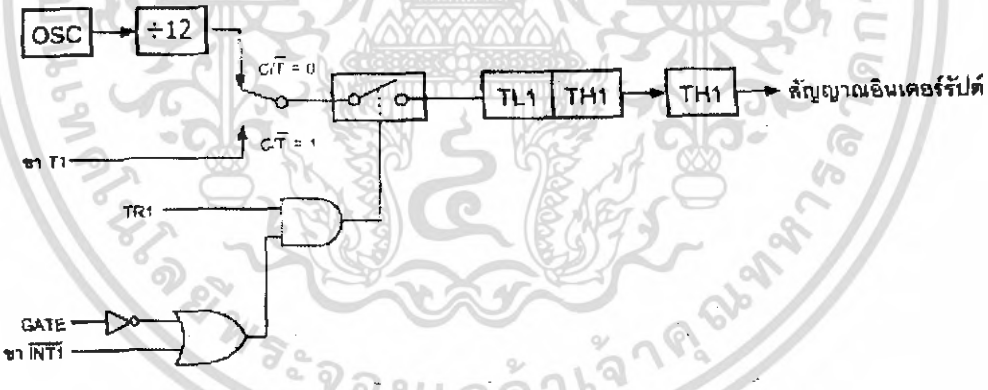
M1 , M0 เป็น 2 บิตที่ใช้ร่วมกันเพื่อเลือกโหมดการทำงานของ Timer การทำงานโหมด 0 , 1 และ 2 ของ Timer 0 จะเหมือนกับ Timer 1 แต่ในโหมด 3 การทำงานของทั้งสองจะต่างกัน ค่าใน M0 และ M1 จะเลือกโหมดการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3.4 การเลือกโหมดใน M0 และ M1

M1	M0	การทำงาน
0	0	โหมด 0 รีจิสเตอร์ THx และ TLx ทำตัวเป็นตัวนับ 13 บิต ค่าจากการนับ 8 บิตบนมาจาก 8 บิตของ THx และอีก 5 บิตล่างมาจากค่า 5 บิตล่างของรีจิสเตอร์ TLx โดยที่ 3 บิตบนของ TLx จะไม่ต้องสนใจเลย
0	1	โหมด 1 รีจิสเตอร์ TLx และ THx ทำตัวเป็นตัวนับ 16 บิต ค่าจากการนับ 8 บิตบนมาจาก 8 บิตของ THx และค่าจากการนับ 8 บิตล่างอยู่ในรีจิสเตอร์ TLx
1	0	โหมด 2 ในการนับของรีจิสเตอร์ TLx ขนาด 8 บิตเมื่อนับถึงค่าสูงสุดคือ FFH เมื่อทำการนับต่อไปจะเกิดการ Overflow แล้วก็จะ RELOAD เอาข้อมูลมาจาก THx เข้าไปยัง TLx เพื่อเป็นการเริ่มต้นในการนับครั้งต่อไป
1	1	โหมด 3 การทำงานของ Timer 0 และ Timer 1 จะต่างกันดังที่จะกล่าวต่อไป

การทำงานของแต่ละโหมดจะมีรายละเอียดดังนี้



รูปที่ 3.26 Timer Mode 0 : 13 bit count

รูปที่ 3.26 เป็นไดอะแกรมของวงจร Timer ภายใน 89C51 ที่ทำงานในโหมด 0 ซึ่ง Timer 0 และ Timer 1 ก็จะมีการทำงานเหมือนกันทุกประการ ในการอธิบายนี้จะใช้วงจรของ Timer 1 จากรูปจะเห็น Switch C/T ซึ่งถ้ากำหนดค่าในบิตC/T ของ TMOD เป็น 0 จะทำให้สวิตช์อยู่ในตำแหน่งบนเพื่อให้สัญญาณนาฬิกาที่ออกจากวงจรออสซิลเลเตอร์ผ่านวงจรหาร 12 ไปยังสวิตช์ Control ถ้าออสซิลเลเตอร์ผลิตสัญญาณนาฬิกา ความถี่ที่ 12 เมกกะเฮิรตซ์ก็จะมีสัญญาณความถี่ 1 เมกกะ

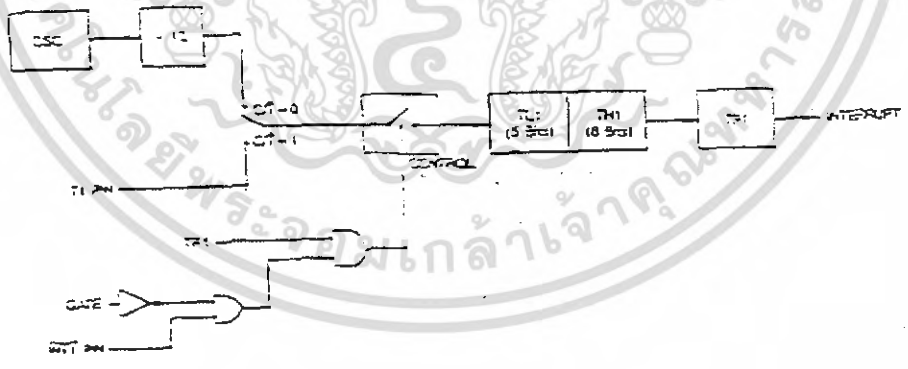
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เฮิรตซ์ ออกจากวงจรหาร 12 ถ้าบิตC/T เป็น 1 จะทำให้สวิตช์ C/T อยู่ในตำแหน่งข้างล่างเพื่อให้สัญญาณที่เข้ามาทาง T ผ่านไปยังสวิตช์ Control สัญญาณที่เข้ามายังสวิตช์ Control จะส่งผ่านไปยังวงจรนับหรือไม่ก็ขึ้นอยู่กับสัญญาณที่ออกมาจาก AND GATE ถ้าบิต TR1 ในรีจิสเตอร์ TCON เป็น 0 จะทำให้สภาวะของสัญญาณที่ออกจาก AND GATE เป็น 0 เสมอและจะไม่มีสัญญาณใดออกจากสวิตช์ Control ไปยังวงจรนับเลย รีจิสเตอร์ TL1 และ TH1 จะไม่ทำงาน แต่ถ้าบิต TR1 เป็น 1 จะทำให้สภาวะสัญญาณออกจาก AND GATE ไปควบคุมสวิตช์ Control ขึ้นอยู่กับสภาวะของสัญญาณ INT1 (หรือ INT0 ถ้าเป็น Timer0) และข้อมูลที่บิต GATE ของรีจิสเตอร์ TMOD ถ้าบิต GATE เป็น 0 หรือสัญญาณที่ขาINT1 มีสภาวะลอจิกเป็น 1 จะทำให้สัญญาณควบคุมสวิตช์ Control เป็น 1 ทำให้มีสัญญาณออกไปยังตัวนับรีจิสเตอร์ TL1 และ TH1 (หรือ TL0 และ TH0 ถ้าเป็น Timer0) รีจิสเตอร์ TL1 จะทำการนับโดยมีเพียง 5 บิต ถึง 7 บิต ของ TLx ที่ไม่ได้ใช้งานก็จะไม่สนใจการทำงานของ Timer0 และ Timer1 ในโหมดนี้จะมีการทำงานเหมือน Timer ในไมโครคอนโทรลเลอร์เบอร์ 8084 ทุกประการ

โหมด 1

ในโหมดนี้จะมีการทำงานของวงจรภายในของ Timer0 หรือ 1 เหมือนกับโหมด 0 ทุกประการแตกต่างกันที่ TLx จะถูกใช้งานทั้ง 8 บิต ทำให้ผลการนับใน TLx และ THx จะมีถึง 16 บิต

โหมด 2



รูปที่ 3.27 Timer Mode 2

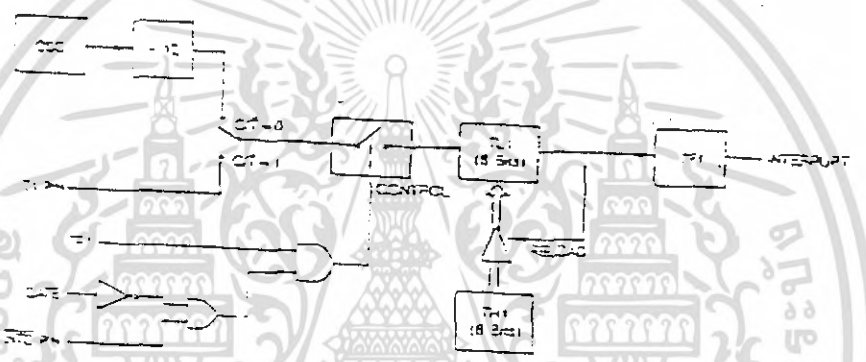
ในรูปที่ 3.27 เป็นไดอะแกรมของวงจร Timer 1 ใน 89C51 ที่ทำงานโหมด 2 Timer0 และ Timer1 มีกรทำงานในโหมด 2 เหมือนกันโดยจะสามารถกำหนดให้ทำหน้าที่เป็น Timer หรือ Counter ได้โดยบิต C/T และควบคุมการนับได้โดยข้อมูลในบิต TR1 และ GATE ในรีจิสเตอร์ TMOD กับสัญญาณที่เข้า INTx เมื่อเริ่มการทำงานข้อมูลในรีจิสเตอร์ TH1 จะถูกโหลดไปยัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ TL1 ทำให้รีจิสเตอร์ TL1 และ TH1 มีค่าเท่ากัน เมื่อเกิดการนับจำนวนไซเคิลของสัญญาณที่ออกจากสวิตช์ Control จะทำให้ค่าที่เกิดจากการนับในรีจิสเตอร์ TL1 เพิ่มขึ้นไปเรื่อยๆ ทีละ 1 จนถึง 0FFH ในการนับครั้งต่อไปจะทำให้บิต TF1 ในรีจิสเตอร์ TCON ไม่เป็น 1 และข้อมูลในรีจิสเตอร์ TH1 จะถูกโหลดไปยังรีจิสเตอร์ TL1 เพื่อเป็นค่าเริ่มต้นการนับต่อไป

โหมด3

การทำงานโหมด 3 ของ Timer 0 และ Timer1 ต่างกัน Timer1 ในโหมด 3 จะไม่ทำงาน Timer0 ในโหมด 3 จะทำงานเป็นตัวนับที่เสมือนมีตัวนับอยู่ 8 บิตอยู่ 2 ตัวคือ TL0 และ TH0 ทำงานแยกกันดังรูปที่ 3.28



รูปที่ 3.28 Timer 0 Mode 3

รีจิสเตอร์ TL0 จะเป็นตัวนับ 8 บิต ที่มีการนับสัญญาณจากออสซิลเลเตอร์หารด้วย 12 หรือนับสัญญาณที่เข้ามาทางขา T0 ขึ้นกับบิต TR0 และ GATE ในรีจิสเตอร์ TMOD กับสภาวะลอจิกของสัญญาณที่ขา INT0 เหมือนกับในการทำงานโหมด 0 , 1 , และ 2 แต่ค่าจากการนับนี้สูงสุดจะเพียง 255 เท่านั้น เมื่อค่าการนับเปลี่ยนจาก 0FFH เป็น 00H คือเกิดการ Overflow จะทำให้บิต TF0 ถูก SET เป็น 1 และอาจเกิดการขัดจังหวะ (Interrupt) การทำงานของโปรแกรมได้ถ้ามีการกำหนดค่าในรีจิสเตอร์ IE และ IP ตัวนับอีกตัวคือ รีจิสเตอร์ TH0 จะทำงานในโหมดของ Timer เท่านั้น คือจะนับจำนวนไซเคิลของสัญญาณที่ออกจากออสซิลเลเตอร์แล้วหารด้วย 12 การนับจะควบคุมได้ด้วยบิต TR1 ในรีจิสเตอร์ TMOD ถ้าบิตนี้เป็น 1 ก็จะมีสัญญาณเข้าไปยัง TH0 แต่ถ้าบิตนี้เป็น 0 ก็จะไมมีสัญญาณเข้าไปยัง TH0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

11. TCON Timer Control Register ตำแน่งหน่วยความจำภายในเท่ากับ 088H รีจิสเตอร์ขนาด 8 บิตนี้ใช้ควบคุมการทำงานและบอกสภาวะของ Timr0 และ Timer1 แต่บิตของรีจิสเตอร์นี้จะทำงานต่างกันดังในรูปที่ 3.29

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.
IT1	TCON.2	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered External interrupt.
IE0	TCON.1	External interrupt 0 edge flag. Set by hardware when External interrupt edge detected. Cleared by hardware when interrupt is processed.
IT0	TCON.0	Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered External interrupt.

รูปที่ 3.29 TCON Timer Control Register

ในรูปที่ 3.29 T0 เป็นชื่อของบิต 0 และ TF เป็นชื่อของบิต 7 ในรีจิสเตอร์ TCON แต่ละบิตมีหน้าที่การทำงานดังต่อไปนี้

IT0 Interrupt 0 เป็นบิตที่จะใช้กำหนดวิธีการขัดจังหวะโปรแกรม อันเนื่องจากสาวะของสัญญาณที่เข้ามาทางขา INT0

ถ้า IT0 เป็น 1 จะเกิดการขัดจังหวะโปรแกรม ถ้าสภาวะของสัญญาณที่ขา INT0 เปลี่ยนจาก 1 เป็น 0

ถ้า IT0 เป็น 0 จะเกิดการขัดจังหวะของโปรแกรม ถ้าเป็นสภาวะของสัญญาณที่ขา INT0 เปลี่ยนเป็น 0

IE0 บิตนี้จะเป็น 1 ถ้าสัญญาณที่เข้ามาทางขา INT0 มีสภาวะของลอจิกของสัญญาณตามที่กำหนดในบิต IT0 แล้วจะทำให้เกิดการขัดจังหวะของโปรแกรม เมื่อเกิดการกระโดดไปทำงานยังโปรแกรมตอบสนองการขัดจังหวะแล้วจะทำให้บิตนี้กลับเป็น 0

IT1 Interrupt 1 เป็นบิตที่จะใช้กำหนดวิธีการขัดจังหวะโปรแกรม อันเนื่องจากสภาวะของสัญญาณที่เข้ามาทางขา INT1

ถ้า IT1 เป็น 1 จะเกิดการขัดจังหวะโปรแกรม ถ้าสภาวะของสัญญาณที่ขา INT1 เปลี่ยนจาก 1 เป็น 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้า IT1 เป็น 0 จะเกิดการขัดจังหวะโปรแกรม ถ้าสภาวะของสัญญาณที่ขา INT1 เป็น 0

IE0 บิตนี้จะเป็น 1 ถ้าสัญญาณที่เข้ามาทางขา INT1 มีสภาวะลอจิกของสัญญาณ ตามที่กำหนดในบิต IT1 แล้วทำให้เกิดการขัดจังหวะโปรแกรมเหมือนกับ IT0 ทีทำงานกับสัญญาณ INT0

TR0 Timer 0 Run Control Bit บิตนี้ถ้าเป็น 0 Timer0 ไม่ทำการนับสัญญาณไม่ว่ากรณีใดๆ ทั้งสิ้น ถ้าบิตนี้เป็น 1 จะทำให้ Timer 0 ทำงานโดยขึ้นกับสัญญาณ GATE INT0 ข้อมูลในบิตนี้จะสามารถ SET เป็น 1 หรือเคลียร์เป็น 0

TF0 Timer0 Overflow Flag บิตนี้จะเป็น 1 เมื่อการนับของรีจิสเตอร์ใน Timer0 (TL0 หรือ TH0 ขึ้นอยู่กับโหมดของการทำงาน) เกิด Overflow ขึ้นคือเอาการนับเพิ่มไปจนถึงค่าสูงสุดแล้วนับต่อไปทำให้ค่าการนับกลับมาเริ่มต้นใหม่ที่ 0 หรือค่า Reload เมื่อ 89C51 กระโดดไปทำงานที่โปรแกรมตอบสนองการขัดจังหวะทำให้บิตนี้กลับเป็น 0

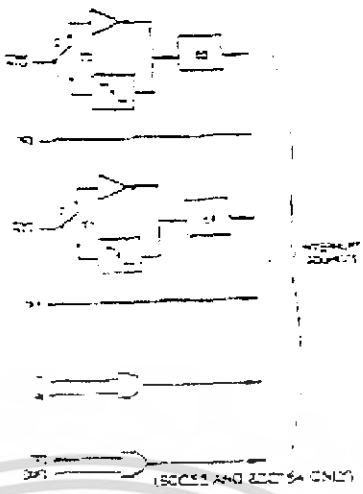
TR1 Timer1 Run Control Bit การทำงานจะเหมือนกับการทำงนของบิต TR0 แต่บิตนี้จะทำงานกับ Timer1

TF1 Timer 1 Overflow flag บิตนี้เหมือนกับบิต TF0 ต่างกันที่ขึ้นกับการำงานของ Timer1

12. บิตแรกที่กล่าวมานั้นจะเกี่ยวข้องกับการขัดจังหวะส่วน 4 บิตหลังนั้นได้กล่าวมาแล้วอย่างละเอียดในเรื่องโหมดการทำงานของ Timer

ในขณะที่ Timer ทำงานในโหมดของ Timer นั้น รีจิสเตอร์ที่ทำหน้าที่เป็นตัวนบจะมีค่าเพิ่มขึ้น 1 ทุกๆ 1 ไซเคิลของเครื่องซึ่งเท่ากับ 12 คาบของสัญญาณจากออสซิลเลเตอร์ ในกรณีที่ Timer ทำงานเป็น Counter เพื่อนับจำนวนไซเคิลของสัญญาณทีเข้ามาทางขา T0 และ T1 รีจิสเตอร์จะเพิ่มค่าไป 1 เมื่อมีการเปลี่ยนสถานะของสัญญาณที่ขาดังกล่าวจาก 1 ไป 0 โดยวงจรภายใน 89C51 จะตรวจสอบสถานะของสัญญาณที่ขาดังกล่าวในช่วงเวลาเฟส 2 ของ Stat5 (S5P2) ในทุกๆ 1 ไซเคิลของเครื่อง เช่นในเวลา S5P2 ของไซเคิลของเครื่องถัดมาพบว่าสัญญาณที่ขา T0 มีสภาวะลอจิกเป็น0 ก็จะทำให้ค่าในรีจิสเตอร์ตัวนับเพิ่มค่าไป 1 แต่ละไซเคิลของเครื่องจะเท่ากับ 12 ไซเคิลของสัญญาณออสซิลเลเตอร์. ดังนั้นสัญญาณที่จะนับได้จะต้องเป็น 1 อย่างน้อยให้ถูกจับได้ใน 1 ไซเคิลของเรื่องและเป็น 0 อย่างน้อยก็ต้องให้ถูกตรวจสอบได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



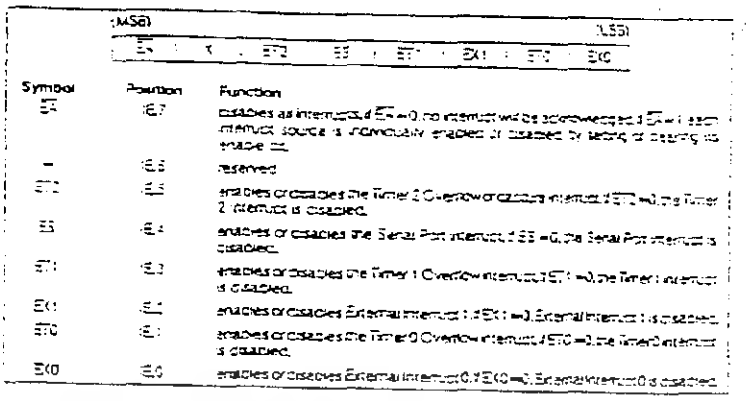
รูปที่ 3.30 แหล่งกำเนิดสัญญาณขัดจังหวะ

ในรูปที่ 3.30 จะสามารถทำให้เกิดการขัดจังหวะได้ 2 วิธี คือมีข้อมูลเข้ามาทางพอร์ทอนุกรมเก็บอยู่ที่รีจิสเตอร์ SBUF และกรณีที่ขอมูลใน SBUF ส่งออกไปทางอนุกรมหมดแล้ว ไม่ว่าเกิดกรณีใดๆก็ทำให้เกิดการขัดจังหวะขึ้น

สัญญาณภายนอกที่เข้ามายัง 89C51 ทางขา INT0 และ INT1 จะสามารถทำให้เกิดการขัดจังหวะการทำงาน 89C52 ได้ โดยสภาวะของสัญญาณนั้นเปลี่ยนจาก 1 เป็น 0 หรือเมื่อสัญญาณนั้นเป็น 0 แล้วแต่การกำหนดในบิต IT0 และ IT1 ของรีจิสเตอร์ TCON จะทำให้บิต TF0 , TF1 เป็น 1 ซึ่งสัญญาณจาก 2 บิตนี้จะสามารถทำให้ให้เกิดการขัดจังหวะได้เช่นกัน ดังเช่นสัญญาณขัดจังหวะที่ 2 และ 4 ในรูปที่3.30

แหล่งกำเนิดสัญญาณทั้ง 6 ที่สมารถทำให้เกิดการขัดจังหวะได้ 5 แบบนี้ ผู้ใช้สามารถกำหนดให้สัญญาณใดบ้างเกิดการขัดจังหวะเรียกว่า Enable หรือไม่ให้เกิดการขัดจังหวะเรียกว่า Disable โดยกากำหนดในรีจิสเตอร์ IE ซึ่งมี 8 บิตแต่ละบิตสามารถ Enable ให้ขัดจังหวะได้จากแต่ละสัญญาณดังรูปที่ 3.31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.31 Interrupt Enable Register

ถ้าต้องการ Enable บิตใดก็ให้โปรแกรมกำหนดค่าในแต่ละบิตนั้นเป็น 1 ถ้าในบิตนั้นเป็น 0 หมายถึง Disable การ Disable จะทำให้มีการขัดจังหวะการทำงานของโปรแกรมเนื่องจากสัญญาณขอขัดจังหวะนั้นๆ EX0 เป็นชื่อบิต 0 และ EA เป็นชื่อของบิต 7

EX0 บิตนี้ใช้สำหรับ Enable สัญญาณที่ข้ามาทางขา INT0 ให้เกิดการขัดจังหวะหรือไม่

ET0 Timer 0 Interrupt Enable Bit ข้อมูลบิตนี้จะใช้ Enable Disable สัญญาณขัดจังหวะที่มาจากวงจร Timer0

EX1 บิตนี้จะใช้ Enable หรือ Disable สัญญาณที่เข้ามาทางขา INT1 ให้เกิดการขัดจังหวะหรือไม่

ET1 Timer1 Interrupt Enable Bit บิตนี้จะใช้ Enable หรือ Disable สัญญาณขัดจังหวะจาก Timer1

ES ข้อมูลนี้บิตนี้จะ Enable หรือ Disable การขัดจังหวะจาก Serial Port อันเนื่องจากข้อมูลเข้ามายัง SBUF หรือข้อมูลจาก SBUF ได้ส่งออกหมดแล้ว

EA บิตนี้จะคอยควบคุมทั้ง 6 บิตที่กล่าวมาแล้ว ถ้าข้อมูลในบิตนี้เป็น 0 จะเป็นการ Disable ทุกบิตที่กล่าวมาแล้ว ทำให้ไม่เกิดการขัดจังหวะโปรแกรมได้เลย แต่ถ้าบิตนี้เป็น 1 การ Enable/Disable ใน 6 บิตที่กล่าวมาแล้วจะขึ้นกับข้อมูล รีจิสเตอร์นั้นคือ IP Interrupt Priority Register

การกำหนดให้บิตใด Enable หรือ Disable นั้นจะเป็นไปโดยอิสระไม่ขึ้นแก่กัน จึงสามารถกำหนดให้บิตใดหรือมากว่า 1 บิต Enable ก็ได้ ดังนั้น 89C51 จึงมี Register อีกตัวที่ใช้เลือกว่าถ้ามีสัญญาณขอการขัดจังหวะโปรแกรมเข้ามาพร้อมกันมากกว่า 1 แล้วจะทำให้โปรแกรมตอบสนองการขัดจังหวะอันใดก่อน รีจิสเตอร์นั้น คือ IP Interrupt Priority Register

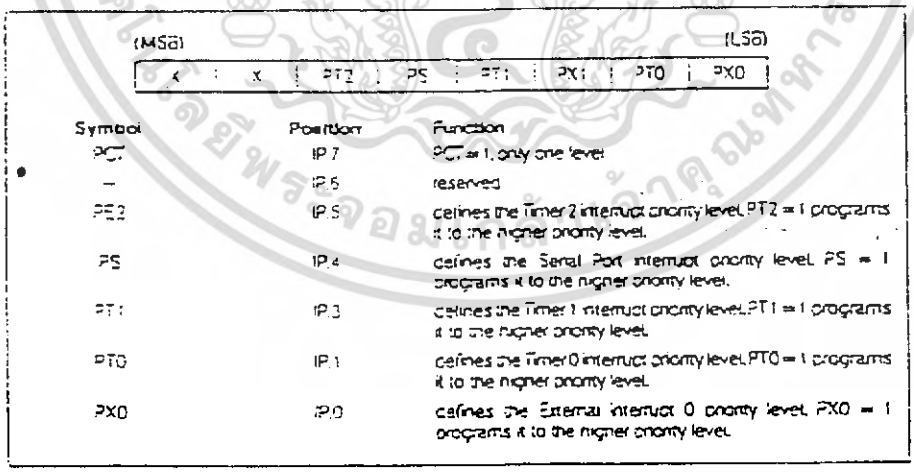
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

13. IP Interrupt Priority register ตำแหน่งหน่วยความจำภายในเท่ากับ 0B8H

ในการตอบสนองต่อสัญญาณขัดจังหวะของ 89C51 นั้นถ้าสัญญาณขัดจังหวะทั้งหมดเข้ามาพร้อมกัน 89C51 จะต้องเลือกทำงานโปรแกรมตอบสนองการขัดจังหวะโดยการตรวจสอบสัญญาณเรียงตามลำดับ ซึ่งเรียกว่าวิธีการ Polling สัญญาณจังหวะหนึ่งจะถูกตรวจสอบก่อนแล้วสัญญาณอื่น ๆ จะถูกตรวจสอบต่อมาถ้าสัญญาณนั้นขัดจังหวะ 89C51 จะสร้างคำสั่ง CVLL เป็นพิเศษขึ้นมาเพื่อไปทำงานโปรแกรมตอบสนองการขัดจังหวะของสัญญาณนั้น เมื่อเสร็จสิ้นแล้วจะกลับมาทำงานในโปรแกรมเดิมก่อนการขัดจังหวะ ทำให้เสมือนว่าสัญญาณแต่ละสัญญาณมีลำดับความสำคัญไม่เท่ากัน สัญญาณขัดจังหวะจะมีลำดับความสำคัญดังนี้ โดยจะเรียงลำดับความสำคัญสูงสุดถึงต่ำสุด

1. IE0
2. TF0
3. IE1
4. TF1
5. RI + TI

แต่ถ้าในการใช้งานบางครั้งจำเป็นที่จะต้องให้สัญญาณใดสัญญาณหนึ่งมีลำดับความสำคัญสูงสุด (Highest Priority) เพื่อจะทำงานโปรแกรมตอบสนองการขัดจังหวะได้ก่อนการขัดจังหวะของสัญญาณอื่น จะสามารถกำหนดลำดับความสำคัญของการขัดจังหวะได้ใหม่โดยการกำหนดข้อมูลในบิตของรีจิสเตอร์ IP (Interrupt Priority Register) ตามตำแหน่งของแต่ละบิตในรูปที่ 3.32



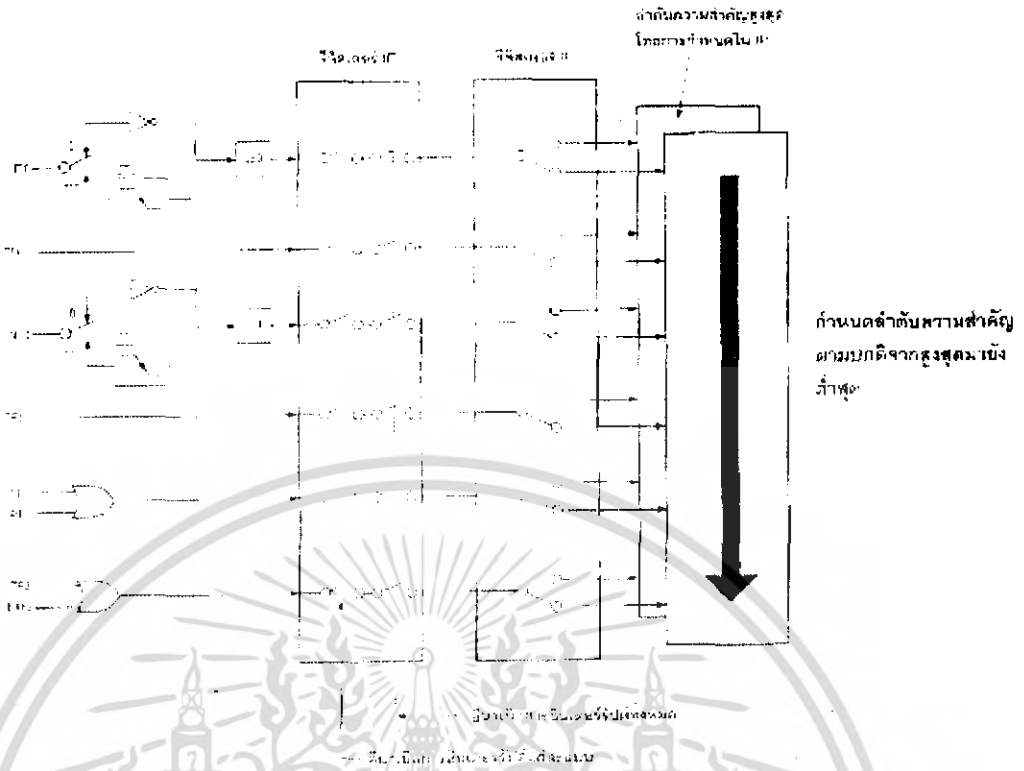
(MSB) (LSB)

-	-	PT2	PS	PT1	PX1	PT0	PX0

Symbol	Position	Function
PCY	IP.7	PCY = 1, only one level
–	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.
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 3.32 IP : Interrupt Priority Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.33 ระบบการขัดจังหวะของ AT89C51

รูปที่ 3.33 เป็นแผนภาพแสดงระบบขัดจังหวะของ 8951 ซึ่งแตกต่างจากของ 89C51 ตรงที่ 8051 จะมีสัญญาณขัดจังหวะมาจาก TF2, EFX2 คือชุดล่างในภาพ

ในรูปจะเห็นว่าแต่ละสัญญาณจะมีสวิตช์ควบคุมอยู่ 3 ตัว 2 ตัวแรกอยู่ในกรอบสี่เหลี่ยม IE Register และอีก 1 สวิตช์อยู่ในกรอบ IP Register สวิทช์ตัวแรกทางซ้ายสุดจะควบคุมข้อมูลด้วยข้อมูลแต่ละบิต บิต 0 ถึงบิต 5 ของรีจิสเตอร์ IE ถ้าข้อมูลเป็น 1 จะทำให้สวิตช์นั้นเปิดวงจร (Closed circuit) การควบคุมสวิตช์ทางซ้ายสุดของแต่ละสัญญาณจะไม่ขึ้นแก่กัน (Individual) สวิตช์ที่ 2 ถัดมาของทุกสัญญาณควบคุมร่วมกันด้วยบิต EA ในรีจิสเตอร์ IE ถ้าบิตนี้เป็น 0 สวิตช์ที่ 2 ของทุกสัญญาณจะเปิดวงจร (Opened Circuit) ทำให้ไม่มีสัญญาณขอขัดจังหวะผ่านไปได้ สวิตช์ 3 ทางขวาสุดจะใช้สำหรับเลือกว่าสัญญาณนั้นจะอยู่ในกลุ่มลำดับความสำคัญสูง (High Priority Interrupt) หรือลำดับความสำคัญต่ำ (Low Priority Interrupt)

ถ้าต้องการให้สัญญาณใดมีลำดับความสำคัญสูงก็ให้กำหนดบิตนั้นในรีจิสเตอร์ IP เป็น 1 สวิตช์ที่ 3 จะเลือกไปอยู่ในตำแหน่งบน ถ้าไม่ต้องการก็กำหนดให้บิตนั้นเป็น 0 บิตใดเป็น 1 เรียกว่าสัญญาณนั้นจะอยู่ในกลุ่มลำดับความสำคัญสูงและบิตใดเป็น 0 เรียกว่าสัญญาณนั้นอยู่ในกลุ่มลำดับความสำคัญต่ำ ถ้าในกลุ่มลำดับความสำคัญสูงมีเพียง 1 สัญญาณก็จะเรียกสัญญาณนั้นมีลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสำคัญสูงสุด ในกลุ่มลำดับความสำคัญเดียวกันก็จะมีการจัดลำดับความสำคัญเฉพาะกลุ่มโดยวิธี Polling เหมือนเดิม เช่น กรณีที่มีการกำหนดในบิตของรีจิสเตอร์ IP ให้มีลำดับความสำคัญสูงหรือต่ำเหมือนกันแล้ว เกิดมีความต้องการของการขัดจังหวะจากสัญญาณนั้น ๆ มาพร้อมกัน 89C51 ก็จะทำงานในโปรแกรมตอบสนองการขัดจังหวะของ Timer 1,External interrupt 1 และ Timer 0 พร้อมกัน 89C51 กำลังทำงานตอบสนองการขัดจังหวะของสัญญาณขัดจังหวะที่ลำดับความสำคัญต่ำอยู่ ถ้ามีสัญญาณสำคัญขัดจังหวะที่มีลำดับความสำคัญสูงกว่าเกิดขึ้น การทำงานของโปรแกรมก็จะกระโดดไปทำงาน ในตำแหน่งโปรแกรมตอบสนองของการขัดจังหวะของสัญญาณที่มีลำดับความสำคัญสูง เสร็จแล้วจึงกลับมาทำงานที่โปรแกรมตอบสนองการขัดจังหวะลำดับความสำคัญต่ำต่อไป แต่ละบิตของรีจิสเตอร์ IP นั้นจะบอกลำดับความสำคัญของแหล่งกำเนิดสัญญาณขัดจังหวะดังนี้

PX0 บิต 0 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะภายนอก 89C51 คือ INT0

PT0 บิต 1 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจาก Timer 0

PX1 บิต 2 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะภายนอกของ 89C51 คือ INT1

PT1 บิต 3 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจาก Timer 1

PT2 บิต 5 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจาก Timer 2 บิตนี้ใช้เฉพาะใน 8951 ที่มี Timer 2

PS บิต 3 เป็นลำดับความสำคัญของสัญญาณขอขัดจังหวะจาก Serial Port ในกรณีที่มีข้อมูลเข้ามาหรือส่งข้อมูลออกสิ้นสุดแล้วบิตที่เหลือก็จะไม่มีการใช้งาน

รายละเอียดของการขัดจังหวะจะกล่าวต่อไปในหัวข้อ 3.9

14. PCON (Power Control Register) ตำแหน่งหน่วยความจำภายในเท่ากับ 87H

Symbol	Position	Name and Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1, the baud rate is doubled when the serial port is being used in either modes 1,2 or 3.
HPD	PCON.6 (83C154 only)	Hard Power Down bit. Setting this bit allows CPU to enter a Power Down state on an external event (1 to 0 transition) on pin P1.5 (pin 3-5). The CPU puts the Hard Power Down mode when pin P1.5 (pin 3-5) go high or when reset is activated.
RPD	PCON.5 (83C154 only)	Resume from Idle or Power Down bit. When 0 RPD has no effect. When 1, RPD permits to exit from Idle or Power Down with any non enabled interrupt source (except timer 2). In this case the program start at the next address. When interrupt is enabled the appropriate interrupt routine is serviced.
-	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON.0	Idle mode bit. Setting this bit activates idle mode operation.

รูปที่ 3.34 PCON : Power Control Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

89C51 เป็นไมโครคอนโทรลเลอร์ที่สร้างขึ้นด้วยเทคโนโลยีทั้งแบบ CHMOS และ HMOS ซึ่งแบบ CHMOS มีข้อดีตรงที่ใช้กำลังไฟต่ำกว่าแบบ HMOS ดังนั้นต่อไปในอนาคตจึงจะมีแต่เฉพาะรุ่น CHMOS เท่านั้น นอกจากนี้แล้ว 89C51 ยังมีข้อดีอีกตรงที่สามารถลดการใช้กำลังไฟลงได้โดยการทำงานใน Idle Mode และ Power Down Mode ใน Idle Mode นั้นสัญญาณนาฬิกาจากออสซิลเลเตอร์จะป้อนให้เฉพาะส่วน Interrupt , Serial Port และ Timer ในส่วนอื่นจะไม่มีสัญญาณนาฬิกาไปเลี้ยงแต่มีไฟเลี้ยงให้กับทุกส่วนในวงจร การใช้กำลังไฟจึงลดลงมาก ส่วนใน Power Down Mode นั้น ออสซิลเลเตอร์จะหยุดทำงานทำให้ไม่มีสัญญาณนาฬิกาไปเลี้ยงส่วนใด ๆ ในวงจรเลยแต่ข้อมูลภายในรีจิสเตอร์จะยังคงอยู่ไม่สูนหายไป รายละเอียดของแต่ละโหมดจะได้กล่าวต่อไป

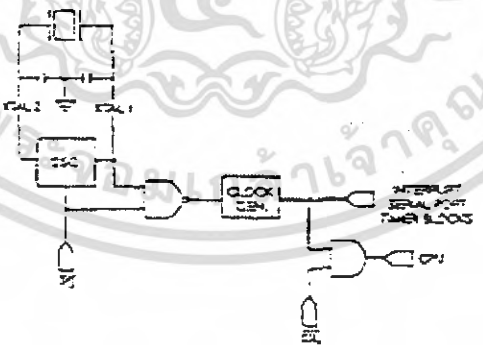
การสั่งงานให้ 89C51 ทำงานในโหมดของ Idle หรือ Power Down จะสามารถทำได้โดยใช้กำหนดค่าในรีจิสเตอร์ PCON (Power Control Register) แต่ละบิตในรีจิสเตอร์ PCON มีดังนี้

IDC บิต 0 ถ้าบิตนี้ถูก Set ให้เป็น 1 89C51 จะเข้าสู่การทำงานใน Idle Mode ทันที

PD บิต 1 ถ้าบิตนี้ถูก Set ให้เป็น 1 89C51 จะเข้าสู่การทำงานใน Power Down ทันที

GF0,GF1 บิต 2,3 เป็น General Purpose Flag bit บิตนี้สามารถกำหนดให้มี ค่าเป็น 0 หรือ 1 ได้ โดยโปรแกรม เพื่อส่งผ่านสถานการณ์ทำงานของ 89C51 ระหว่างแต่ละโปรแกรมย่อย

SMOD บิต 7 เป็นบิตมราใช้ร่วมในการกำหนดส่งข้อมูล (Baud Rate) ผ่านทางพอร์ทอนุกรม ซึ่งในการรับ-ส่งข้อมูลผ่านทางพอร์ทอนุกรมโหมด 1 และ 3 จะสามารถกำหนดอัตราการส่งข้อมูลได้ตามอัตรากาเกิด Overflow ใน Timer 1 ถ้าบิตนี้เป็น 1 จะทำให้อัตราการส่งข้อมูล เพิ่มขึ้น 2 เท่า รายละเอียดการส่งข้อมูลผ่านพอร์ทอนุกรมดูในข้อการรับส่งข้อมูลทางพอร์ทอนุกรม บิต 4,5,6 ไม่ได้ใช้งาน



รูปที่ 3.35 Power down และ Idle mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ldle Mode

ในรูปที่ 3.35 ขณะที่ 89C51 ทำงานตามปกติไปจนถึงคำสั่งที่ทำให้บิต 0 ของรีจิสเตอร์ PMOD มีค่าเป็น 1 ก็จะเข้าสู่การทำงานใน ldle Mode โดยสัญญาณ IDL จะเป็น LOW (สัญญาณจะตรงข้ามกับข้อมูลในบิต 0) ขณะนี้สัญญาณนาฬิกาออสซิลเลเตอร์ไม่ออกจาก AND GATE ไปยังส่วน CPU โดยจ่ายเฉพาะส่วย Interrupt , Timer และ Serial Port ในขณะนี้ 89C51 จะเสมือนหยุดการทำงานโดยข้อมูลใน Stack Pointer , Program Counter , Program Status Word , Accumulator และรีจิสเตอร์อื่น ๆ จะไม่เปลี่ยนแปลงข้อมูลพอร์ทต่าง ๆ จะยังคงค่าเดิมไว้เหมือนกับก่อนเข้าสู่ ldle mode และสัญญาณ ALE กับ PSEN จะเป็นลอจิก High ขณะนี้การใช้กระแสไฟของ 89C51 จะไม่มีการเปลี่ยนสถานะลอจิก การที่จะออกจาก ldle Mode ทำได้ 2 วิธี

วิธีที่ 1 โดยทำการขัดจังหวะจากสัญญาณขัดจังหวะทั้ง 6 ที่กล่าวมาแล้ว เมื่อมีสัญญาณขอขัดจังหวะจากแหล่งใดก็ตาม จะทำให้บิต 0 ของรีจิสเตอร์ PCON มีค่าเป็น 0 และการทำงานของ 89C51 จะออกจาก ldle Mode โดยกระโดดไปทำงานยังตำแหน่งของโปรแกรมตอบสนองการขัดจังหวะนั้นๆ เมื่อเสร็จสิ้นการทำงานของโปรแกรมตอบสนองการขัดจังหวะโดยการทำงานคำสั่ง RETI ก็จะกลับมาทำงานยังคำสั่งที่มีอยู่ต่อจากคำสั่งทำให้บิต 0 ของรีจิสเตอร์ PMOD เป็น I ซึ่งทำให้การทำงนเข้าสู่ ldle Mode เช่นคำสั่งที่ตำแหน่ง 200H คือ MOV PCON,#IH ที่เป็นคำสั่งที่ทำให้บิต IDC มีค่า 1 ดังนั้นเมื่อทำงานที่คำสั่งนี้เสร็จสิ้นก็จะหยุดการทำงาน และเมื่อเกิดการขัดจังหวะเนื่องจากสัญญาณขัดจังหวะใด ๆ ก็ตาม 89C51 จะออกจาก ldle Mode ไปทำงานที่โปรแกรมตอบสนองการขัดจังหวะ เมื่อเสร็จสิ้นการทำงานของโปรแกรมการตอบสนองการขัดจังหวะแล้วจะกระโดดมาทำงานที่ตำแหน่งของคำสั่ง MOV PCON,#1H

วิธีที่ 2 ก็คือการป้อนสัญญาณที่มีสภาวะลอจิก I เข้าไปยังขา RST เพื่อทำการรีเซท 89C51 สัญญาณรีเซทนี้จะต้องมีลอจิกเป็น 1 ในระหว่างนี้ 89C51 จะทำงานในคำสั่งต่อจากคำสั่งที่ทำให้บิต 0 ของ PCON เป็น 1 เข้าสู่ Idle Mode ต่อไปอีก 2-3 คำสั่ง ก่อนที่ทุกอย่างจะเข้าสู่การรีเซท (ดูรายละเอียดการรีเซทในหัวข้อจาก 3.4) ดังนั้นจะต้องระวังคำสั่งที่อยู่ต่อจากคำสั่งที่ทำให้เข้าสู่ ldle Mode อาจทำให้ข้อมูลบนพอร์ทเปลี่ยนแปลงจนทำให้อุปกรณ์ที่มาต่อเสียหายเมื่อกลับออกจากตัว ldle Mode

ในวิธีที่ I นั้นแสดงว่าการเข้าสู่โปรแกรมตอบสนองการขัดจังหวะจะเป็นได้ 2 กรณี คือ ขณะที่ทำงานตามปกติแล้วมีสัญญาณขัดจังหวะก็กระโดดไปทำงาน ในส่วนโปรแกรมตอบสนองการขัดจังหวะ หรือในกรณีที่อยู่ใน ldle Mode แล้วมีสัญญาณขัดจังหวะก็จะกระโดดไปทำงานในโปรแกรมตอบสนองการขัดจังหวะ จึงอาจทำให้โปรแกรมตอบสนองการขัดจังหวะนั้นมาจากกรณีใด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

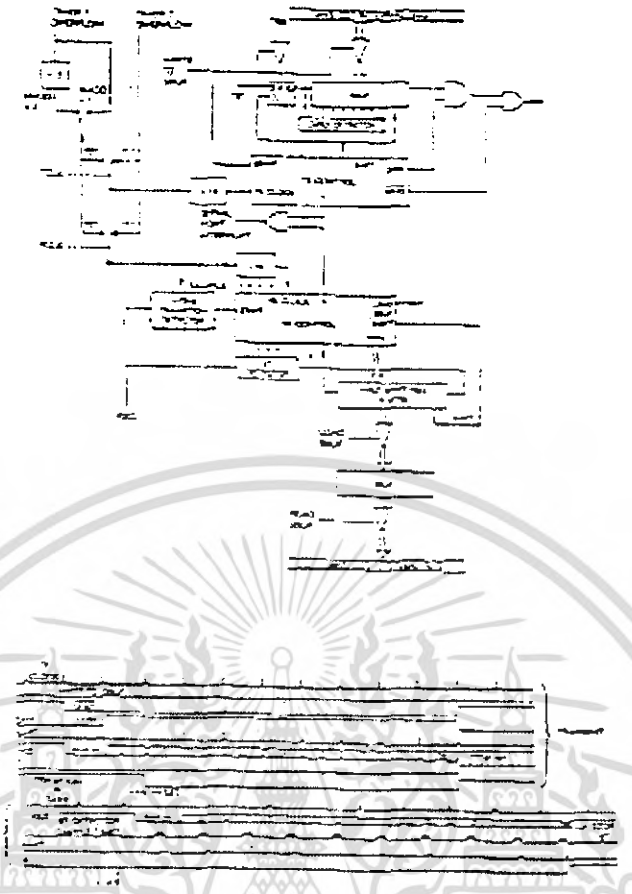
Power Down Mode

ในการเข้าสู่ Power down mode นั้นจะทำได้โดยการใช้โปรแกรมกำหนดให้บิต PD หรือบิต 1ของรีจิสเตอร์ PCON มีค่าเป็น 1 เช่น MOV PCON,#2 เมื่อ 89C51 ทำงานที่คำสั่งนี้เสร็จสิ้น สัญญาณ PD ในรูปที่ 3.35 จะเป็น 0 เพราะตรงข้ามกับข้อมูลในบิต PD ทำให้การทำงานจะเข้าสู่ Power Down Mode ทันที ในโหมดนี้ออสซิลเลเตอร์จะหยุดการทำงานทำให้มีสัญญาณนาฬิกาไปยังส่วนต่าง ๆ ภายใน 89C51 ดังนั้นจะไม่มีการทำงานใด ๆ รวมทั้งข้อมูลในรีจิสเตอร์ทุกตัวจะไม่เปลี่ยนแปลง และข้อมูลในแรมภายใน ก็จะไม่เปลี่ยนแปลงขณะนี้สัญญาณออกจากขา ALE และ PSEN จะเป็น 0 การใช้กำลังไฟของ AT89C51 จะต่ำมากอีกทั้งสามารถลดไฟเลี้ยงวงจรที่ขา VCC ลงได้จนถึง 2 โวล์ทโดยไม่ทำให้ข้อมูลใด ๆ ใน AT89C51 สูญหายไป การออกจาก Power Down Mode ทำได้วิธีเดียวคือ การป้อนสัญญาณลดจิก 1 เข้าไปยังขา RST ซึ่งทำให้เข้าสู่สถาวะการรีเซท AT89C51 แต่จะทำให้ข้อมูลใน SFR เปลี่ยนแปลงไป ถ้าในขณะที่อยู่ใน Power Down Mode มีการลดไฟเลี้ยงวงจรจะต้องให้ไฟเลี้ยงวงจรกับมาอยู่ที่ 5 โวล์ท ก่อนที่จะเข้าสู่การรีเซท

3.8 การรับ-ส่งข้อมูลทางพอร์ทอนุกรม

ในการรับ-ส่งข้อมูลแบบอนุกรมผ่านทางพอร์ทอนุกรมนั้น จะต้องมีการกำหนดโหมดการทำงานในรีจิสเตอร์ SCON และในบางโหมดของการทำงานจะสามารถกำหนดอัตราการส่งข้อมูลได้โดยการโปรแกรมใน Timer ข้อมูลที่จะส่งออกหรือรับเข้าทางพอร์ทอนุกรมจะอยู่ที่รีจิสเตอร์ SBUF การทำงานของวงจรภายในแต่ละโหมดมีดังนี้ โดยแสดงการทำงานในโหมด 1 เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.36 Serial Port mode 1

จากรูปที่ 3.36 บิต SMOD จะเป็นตัวเลือกว่า สัญญาณ Timer 1 Overflow ที่ส่งไปยังวงจรหาร 16 จะถูกหาร 2 ก่อนหรือไม่ ถ้า SMOD เป็น 1 สัญญาณ Timer 1 จะไม่ถูกหาร แต่ถ้า SMOD เป็นสัญญาณ 1 Timer Overflow จะถูกหาร 2 ก่อนที่จะเข้าวงจรหาร 16 การส่งข้อมูลจะเริ่มจากการที่มีคำสั่งเขียนข้อมูลไปยังรีจิสเตอร์ SBUF จะมีสัญญาณ Write to SBUF เกิดขึ้นเพื่อรับข้อมูลจาก Internal Bus ด้านบนไปเก็บรีจิสเตอร์ SBUF และทำให้เอาต์พุตของ D FLIP FLOP ทางซ้ายของ SBUF มีค่าเป็น I และเป็นบิตที่ 9 ของการส่งข้อมูล สัญญาณ Write to SBUF ยังส่งไปยัง TX control ด้วยขณะนี้ข้อมูลในวงจรหาร 16 มีค่าเป็นอะไรไม่ทราบจึงจะรอจนกว่าข้อมูลในวงจรหาร 16 นับเพิ่มขึ้นจนถึงค่าสูงสุดแล้ววนกลับมาเป็น 0 คือเกิดการวนกลับทำให้เริ่มการส่งข้อมูลที่เวลา S1P1 ของไซเคิลเครื่องถัดไป (การส่งข้อมูลออกจะสัมพันธ์กับการเกิด Overflow ในวงจรหาร 16) สัญญาณ SEND จาก TX Control เปลี่ยนสภาวะลอจิกเป็น 0 แล้วเริ่มส่งข้อมูลที่เป็น Start bit (0) ออกไป เมื่อส่ง Start Bit ออกไปแล้ววงจร Tx Control ก็จะทำให้สัญญาณ DATA เป็น 1 เพื่อเลื่อนข้อมูลใน SBUF ออกไป เริ่มจากบิต 0 จนถึงบิตที่ 7 การส่งข้อมูลนี้จะเกิดขึ้นเมื่อสัญญาณ Tx Clock เปลี่ยนสภาวะจาก 0 เป็น 1 ดังในรูปที่ 3.36 ขณะที่ข้อมูลถูกเลื่อนออกไปนั้นจะมี 0 ถูกเลื่อนเข้ามา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางซ้ายของรีจิสเตอร์ เมื่อข้อมูลเลื่อนออกไปทั้ง 8 บิตแล้วบิตที่ 9 ซึ่งเป็น 1 และตอนต้นอยู่ทางซ้ายจะถูกเลื่อนมาอยู่ในตำแหน่งสุดท้ายทางขวาของรีจิสเตอร์ SBUF และทางซ้ายของหลักนี้จะมี 0 อยู่ทั้ง 8 บิตใน SBUF ทำให้ Zero Detector ส่งสัญญาณ Shift ออกไปเป็นการส่งข้อมูลบิตสุดท้าย (บิต 7) ออกไป ก็จะรออีก 1 TX Clock (Bit Cloek) ก็จะทำให้ขา TXD ส่งข้อมูล Stop Bit (1) ออกมา สัญญาณ DATA ซึ่งมีสภาวะลอจิกเป็น 1 มาตั้งแต่เริ่มส่งข้อมูลบิต 0 ก็จะกลับไปเป็น 0 และบิต TI จะเป็น 1 เพื่อบอกการสิ้นสุดการส่งข้อมูลทั้งหมดจะสิ้นสุดเมื่อสัญญาณ TX Clock ไซเคิลที่ 10 นับตั้งแต่สัญญาณ SEND เปลี่ยนสภาวะลอจิกเป็น 0

การรับข้อมูลจะขึ้นกับอัตราการเกิด Overflow ใน Timer 1แล้วหาร 2 หรือไม่ขึ้นกับค่าของบิต SMOD สัญญาณนี้จะเข้าวงจรหาร 15 และเป็นตัวกำหนดอัตราการรับข้อมูล การรับข้อมูลจะเริ่มจากวงจร 1-TO-0 Transition Detector พบว่าสัญญาณที่ขา RXD เปลี่ยนจาก 1 เป็น 0 ซึ่งหมายถึงข้อมูล Start bit เข้ามา การตรวจสอบนี้จะกระทำด้วยอัตราเดียวกับสัญญาณที่เข้าวงจรหาร 16 เมื่อพบการเปลี่ยนสถานะลอจิกที่ขา RXD ก็จะเริ่มการรับข้อมูล ขณะนี้จะรีเซทวงจรหาร 16 ให้มีค่าเป็น 0 เพื่อสร้างสัญญาณ RX Clock ให้เข้าจังหวะ (Synchronous) กับข้อมูลที่เข้ามาโดยสัญญาณ RX Clock จะเป็น 1 เมื่อการนับของวงจรหาร 16 มีค่าเป็น 15 ขณะที่วงจรหาร 16 นับถึง 7 , 8 และ 9 จะมีการตรวจสอบข้อมูลที่เข้ามาทางขา RXD เพื่อเป็นการตรวจข้อมูลนั้นเป็นอะไรถ้าอย่างน้อยข้อมูล 2 ถึง 3 เป็นค่าใดก็จะถือว่าข้อมูลที่เข้ามาเป็นค่านั้น ถ้าในการตรวจสอบ Star Bit แล้วพบว่าผิดพลาด คือไม่เป็น 0 ก็จะรีเซทการทำงานเพื่อไปตรวจสอบการเปลี่ยนสถานะจาก 1 เป็น 0 ของข้อมูลที่ขา RXD ใหม่แต่ถ้าพบ Start Bit ก็จะเก็บข้อมูลทั้งหมดที่เข้ามาโดยเลื่อนข้อมูลเข้าไปยัง Input Shift Register ที่สัญญาณควบคุมการเลื่อนข้อมูล (Shift) ส่งมาจาก RX Control ในเริ่มต้นการรับข้อมูลจะมีการเขียนข้อมูล 1FFH ไปเก็บใน Input Shift Register ขณะที่ข้อมูลถูกเลื่อนเข้าไปทางขวาของ Input Shift Register ก็จะมี 1 ถูกเลื่อนออกไปทางซ้ายทุกครั้งที่มีข้อมูลเข้ามา เมื่อ Start Bit ที่รับเข้ามาถูกเลื่อนไปทางซ้ายสุดของ Input Shift Register ก็จะมีสัญญาณไปบอก RX Control Block หลังจากข้อมูลบิตสุดท้ายเข้ามาแล้วก็จะโหลด (Load) เอาข้อมูล 8 บิตไปเก็บในรีจิสเตอร์ SBOF พร้อมทั้ง Set ค่าในบิต RI และ RB8 ของรีจิสเตอร์ SCON แต่การโหลดข้อมูลไปเก็บนี้จะเกิดขึ้นได้ก็ต่อเมื่อ

1. RI = 0 และ
2. SM2 = 0 หรือถ้า SM2 = จะต้องได้รับ stop bit เป็น 1

ถ้าไม่มีสภาวะใดสภาวะใดสภาวะหนึ่งดังกล่าวแล้ว ข้อมูลที่รับเข้ามาจะถูกทิ้งไปคือ ไม่โหลดไปเก็บในรีจิสเตอร์ SBUF ถ้ามีสภาวะดังกล่าวถูกต้อง stop bit จะถูกนำไปเก็บในรีจิสเตอร์ SBUF และบิต RI จะเป็น 1

แต่ไม่ว่าทั้ง 2 กรณีจะเกิดหรือไม่ก็จะกลับไปสู่การตรวจสภาวะเปลี่ยนจาก 1 เป็น 0 ที่ขา RXD เพื่อรับข้อมูลต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการรับข้อมูลแบบอนุกรมโหมด 1 นี้อัตราการส่งข้อมูลแต่ละบิต (Band Rate) จะขึ้นกับอัตราการเกิด Overflow ใน Timer 1 ดังสมการ

$$Baudrate = \frac{2^{SMOD}}{32} \times (Timer_1 OverflowRate)$$

ในขณะที่ใช้ Timer 1 เป็นตัวกำหนด Baud Rate นี้จะต้อง Disable ไม่ให้เกิดการขัดจังหวะเนื่องมาจากการ Overflow Timer 1 อาจใช้ในโหมดของ Timer หรือ Counter ก็ได้ซึ่งเมื่อการนับในรีจิสเตอร์ตัวนับมีค่าสูงสุดแล้วกลับมาเป็น 0 ก็จะเกิด Overflow เช่นเดียวกัน แต่โดยปกติแล้วจะใช้ Timer 1 นี้ในโหมดของ Timer ที่มีการทำงานแบบ Auto Reload โหมด 2 เพื่อว่าเมื่อค่าในการนับโดย

รีจิสเตอร์ TL1 ถึงค่าสูงสุดก็จะโหมดค่าในรีจิสเตอร์ TH1 มาไว้ใน TL1 สำหรับเป็นค่าเริ่มต้นการนับต่อไป ซึ่ง Baud rate จะมีค่า

$$Baudrate = \frac{2^{SMOD}}{32} \times \frac{OscillatorFrequency}{12 \times [256 - (TH1)]}$$

โดยที่ SMOD เป็นบิตหนึ่งในรีจิสเตอร์ PCO

3.9 การขัดจังหวะ

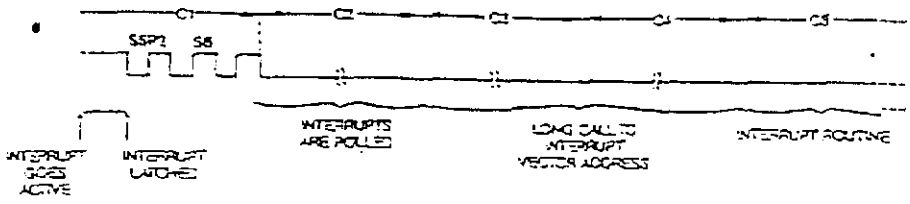
การขัดจังหวะคือสภาวะหนึ่งที่คอมพิวเตอร์กำลังทำงานอยู่แล้วถูกขัดจังหวะด้วยสัญญาณหรือคำสั่งพิเศษที่ทำให้คอมพิวเตอร์ต้องละจากงานที่กำลังทำอยู่ ไปทำงานในโปรแกรมตอบสนองการขัดจังหวะนั้น เมื่อเสร็จแล้วก็จะกลับมาทำงานเดิมต่อไปได้ ใน 89C51 จะสามารถขัดจังหวะการทำงานได้ 6 แหล่งคือ

1. INT0 ,INT1 เป็น 2 ขาของ 89C51 ที่จะรับสัญญาณจากภายนอก ของการขัดจังหวะจะเกิดขึ้น ถ้าสัญญาณที่ขาดังกล่าวมีสภาวะลอจิกเป็น 0 หรือเปลี่ยนจาก 1 เป็น 0 โดยเลือกด้วยการกำหนดในบิต IT0 หรือ IT1 ในรีจิสเตอร์ TCON

2. TF0 ,TF1 เป็นบิตหนึ่งที่จะบอกการทำงานของ Timer 0, Timer 1 เมื่อเกิด Overflow ขึ้นใน Timer จะทำให้บิตนี้เป็น 1 และเกิดการขัดจังหวะการทำงานของ 89C51 ได้

3. TI ,RI เป็น 2 บิต ในรีจิสเตอร์ SCON ถ้าบิตนี้ถูกเซตให้เป็น 1 โดยฮาร์ดแวร์ อันเนื่องจากเสร็จสิ้นการส่งหรือรับข้อมูลจะสามารถทำให้เป็นการขัดจังหวะได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.37 ไดอะแกรมเวลาของการตอบสนองการขัดจังหวะ

89C51 จะทำการอ่านสัญญาณจากทั้ง 6 แหล่งที่เวลา S5P2 ของทุก ๆ ไซเคิลของเครื่อง (Machine Cycle) เข้ามาเก็บและในช่วงของไซเคิลของเครื่องถัดไปก็จะตรวจสอบสถานะของสัญญาณทั้ง 6 ที่เก็บเข้ามา ถ้าสัญญาณนั้นมีการขัดจังหวะที่ถูกต้อง 89C51 ก็จะละทิ้งการทำงานเดิมไว้ชั่วคราวแล้วสร้างคำสั่ง LCALL ขึ้นมาภายใน 89C51 เพื่อไปทำงานในโปรแกรมตอบสนองการขัดจังหวะแต่ละสัญญาณนั้น เมื่อทำงานในโปรแกรมตอบสนองการขัดจังหวะเสร็จสิ้นก็จะสามารถกลับมาทำงานเดิมได้ โดยคำสั่ง RET1 เป็นคำสั่งสุดท้ายในโปรแกรมตอบสนองการขัดจังหวะ สัญญาณขัดจังหวะ จากแต่ละแหล่งจะมีตำแหน่งหน่วยของความจำที่เก็บตัวโปรแกรมตอบสนองการขัดจังหวะไว้ต่างกันดังนี้

สัญญาณที่ขอขัดจังหวะ ตำแหน่งเริ่มต้นโปรแกรมตอบสนองการขัดจังหวะ

1	INT0	0003H
2	TF0	000BH
3	INT1	0013H
4	TF1	001BH
5	TI , RI	0023H

ตำแหน่งเริ่มต้นโปรแกรมนี้เป็นตำแหน่งใน Program area เช่น ถ้ามีสัญญาณของ INT0 เข้ามาแล้ว 89C51 ตรวจสอบว่ามีการขัดจังหวะถูกต้อง ก็จะละทิ้งการทำงานเดิม แล้วไปทำงานที่โปรแกรมตอบสนองการขัดจังหวะที่มีตำแหน่งเริ่มต้นอยู่ที่ตำแหน่ง 0003H เมื่อเสร็จสิ้นการทำงานของโปรแกรมตอบสนองการขัดจังหวะจะต้องมีคำสั่งตัว RETI อยู่ก็เพื่อกลับมาสู่การทำงานเดิมได้ 89C51 โดยจะทำงานตรวจสอบสัญญาณดังกล่าวว่ามีสัญญาณใดขอการขัดจังหวะมาบ้างได้โดยวิธี Polling คือการตรวจสอบเรียงตามลำดับจาก 1 , 2 , 3 , 4 และ 5 ตามลำดับ ดังนั้นถ้ามีการขอขัดจังหวะเข้ามาพร้อมๆ กันAT89C51ซึ่งตรวจสอบการขอขัดจังหวะแบบ Polling จะพบว่าสัญญาณมีการขอขัดจังหวะจากสัญญาณต้น ๆ ก่อนจึงตอบสนองต่อการขอขัดจังหวะของสัญญาณต้น ๆ ก่อนหรืออีกนัยหนึ่งก็คือ สัญญาณขอการขัดจังหวะต้น ๆ จะมีลำดับความสำคัญสูงสุด (Highest

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Priority) และในสัญญาณที่ 5 จะมีลำดับความสำคัญต่ำที่สุด (Lowest Priority) อย่างไรก็ตาม สามารถที่จะจัดลำดับความสำคัญของสัญญาณขัดจังหวะนี้ใหม่ เพื่อให้มีการตอบสนองของการขัดจังหวะสัญญาณขอการขัดจังหวะลำดับหลังได้ โดยการโปรแกรมในรีจิสเตอร์ IP (Interrupt Priority Register) และสามารถกำหนดว่าจะให้ทำโปรแกรมตอบสนองการขัดจังหวะเมื่อมีสัญญาณขอขัดจังหวะเข้ามาหรือไม่ก็ได้ โดยการโปรแกรมในรีจิสเตอร์ IE (Interrupt Enable Register)

เมื่อ 89C51 ทำการตรวจสัญญาณขอการขัดจังหวะที่เก็บเข้ามาเมื่อเวลา S5P2 แล้วพบว่ามีการขอขัดจังหวะนั้น แม้ว่ามีการ Enable ในรีจิสเตอร์ IE ถูกต้อง แต่จะต้องมีเงื่อนไขดังนี้ด้วย

1. ไม่ได้กำลังทำงานในส่วนโปรแกรมตอบสนองของการขัดจังหวะของสัญญาณ ที่มีลำดับความสำคัญสูงกว่าหรือเท่ากัน เช่น กำลังทำงานในโปรแกรมตอบสนองการขัดจังหวะของสัญญาณ INT0 อยู่แล้วมีการขอขัดจังหวะจากสัญญาณ INT1 อีกจะไม่เกิดการทิ้งงานเดิม คือไม่มีการไปทำงานที่โปรแกรมตอบสนองการขัดจังหวะของสัญญาณ INT1

2. เนื่องจากการส่งสัญญาณเข้าไปเพื่อตรวจสอบนั้นจะทำให้เวลาของ S5P5 ของในไซเคิลสุดท้ายของคำสั่ง และคำสั่งที่อยู่ถัดมาจะต้องในเวลาทำงาน 2 ไซเคิลของเครื่อง ดังนั้นการตรวจสอบจะกระทำในไซเคิลแรก แม้ว่ามีการขอการขัดจังหวะเข้ามา ก็จะไม่ทำโปรแกรมตอบสนองการขัดจังหวะ จะต้องอ่านสัญญาณที่เวลา S5P2 อีกครั้งแล้วไปตรวจสอบที่ไซเคิลที่ 2 ของคำสั่งถ้ามีการขอขัดจังหวะถูกต้องจึงจะข้ามไปทำงานในโปรแกรมตอบสนองการขัดจังหวะ

3. คำสั่งที่กำลังทำงานอยู่ขณะที่ตรวจสอบสัญญาณการขอตัวขัดจังหวะ จะต้องไม่ใช่คำสั่ง RET หรือคำสั่งใด ๆ ก็ตามที่พยายามเขียนข้อมูลไปยังรีจิสเตอร์ IE หรือ IP

สัญญาณขอขัดจังหวะที่ถูกอ่านเข้าไปที่เวลา S5P2 นี้ไม่ว่าได้รับการตอบสนองหรือไม่จะถูกทิ้งไป แล้วอ่านเข้าไปใหม่ทุกเวลา S5P2

3.10 เมโมรีแม็พไอโอเทคนิค

เป็นวิธีการเพิ่มหมายเลขพอร์ตอินพุตเอาต์พุต โดยการกำหนดหมายเลขพอร์ตให้แทนที่ที่ตำแหน่งของหน่วยความจำ ตำแหน่งใดตำแหน่งหนึ่ง ซึ่งเป็นหน่วยความจำประเภทหน่วยความจำภายนอก ซึ่ง ณ ตำแหน่งนั้นไม่มีหน่วยความจำจริงต่ออยู่ เพียงแต่ใช้หมายเลขของหน่วยความจำเป็นตัวอ้างอิงในการติดต่อ ในกรณีของ ชุดจำลอง พีแอลซี จะต้องใช้หมายเลขพอร์ตซึ่งได้จากวิธีการดังกล่าวนี้เป็นหมายเลขที่แน่นอน (แอ็ดทีฟที่หมายเลขเพียงหมายเลขเดียว เพื่อป้องกันความผิดพลาดที่จะเกิดขึ้นกับอุปกรณ์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ส่วนประกอบของโครงงาน

4.1 ส่วนประกอบของโครงงานแบ่งออกเป็น 2 ส่วนใหญ่ ๆ คือ

4.1.1 HARDWARE

จะมีส่วนของคอนโทรลเลอร์ เป็นส่วนที่สำคัญมากในโครงงานนี้โดยใช้ Micro controller ตระกูล MCS-51 เบอร์ที่ใช้ได้แก่เบอร์ AT89C51 ทฤษฎีและการทำงานได้บรรยายแล้วในบทที่ 2 ซึ่งเราได้ทำการทดลองในรูปแบบต่างๆแบ่งได้เป็น 3 แบบ

1. การเปิด-ปิด โดยใช้รีโมตเป็นตัวควบคุม ซึ่งสามารถควบคุมการเปิด-ปิดไฟได้ทั้งหมด 4 ดวง

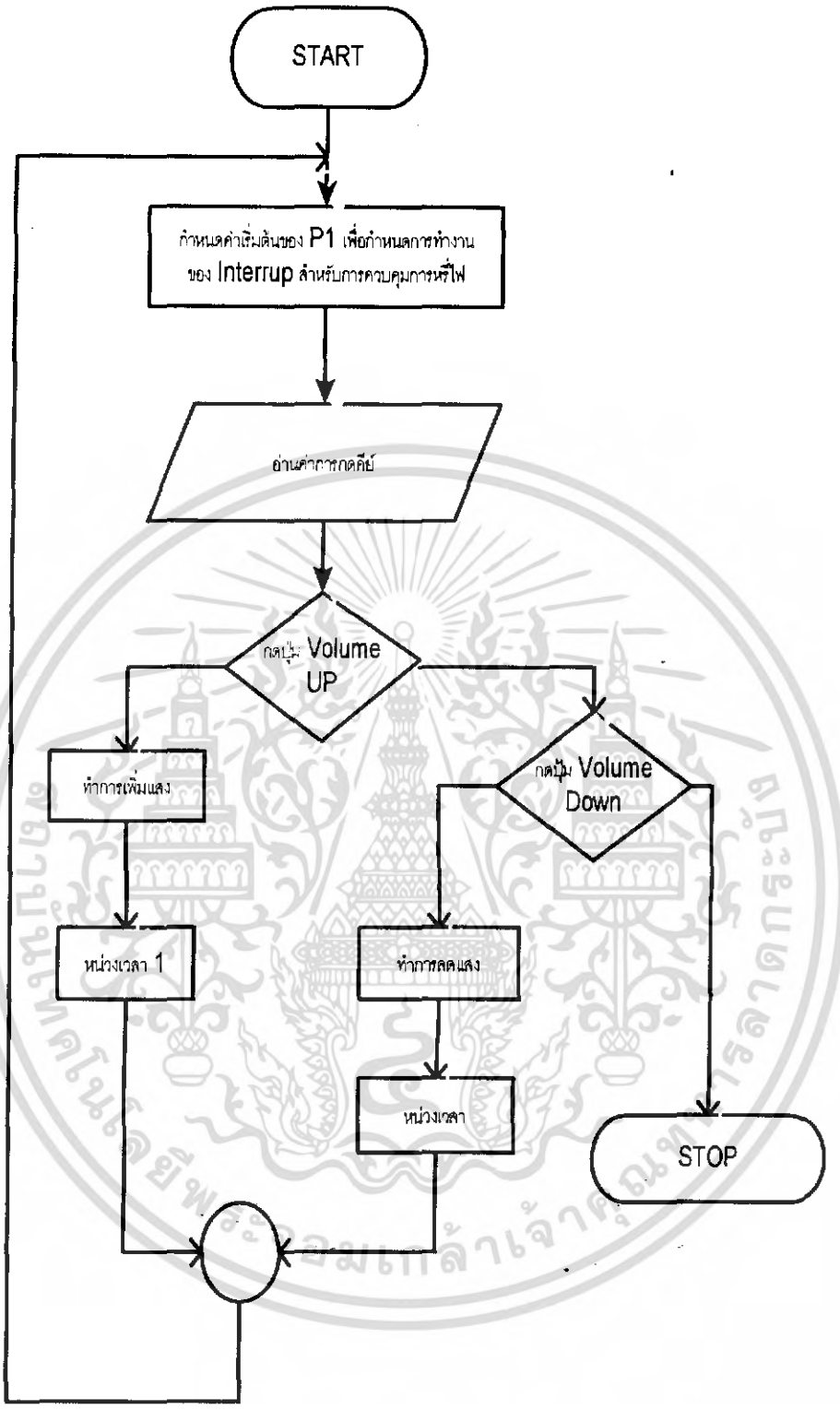
2. การกดคีย์เพื่อทำการหรี่ไฟ ซึ่งสามารถควบคุมไฟได้ทั้งหมด 1 ดวงการทดลองนี้เป็นการเซ็ค Phase Control

3. การเปิดไฟด้วยรีโมต แต่เมื่อทำการปิดไฟ ไฟจะค่อยๆหรี่ลง ซึ่งสามารถควบคุมไฟได้ 1 ดวง

4.1.2. SOFTWARE

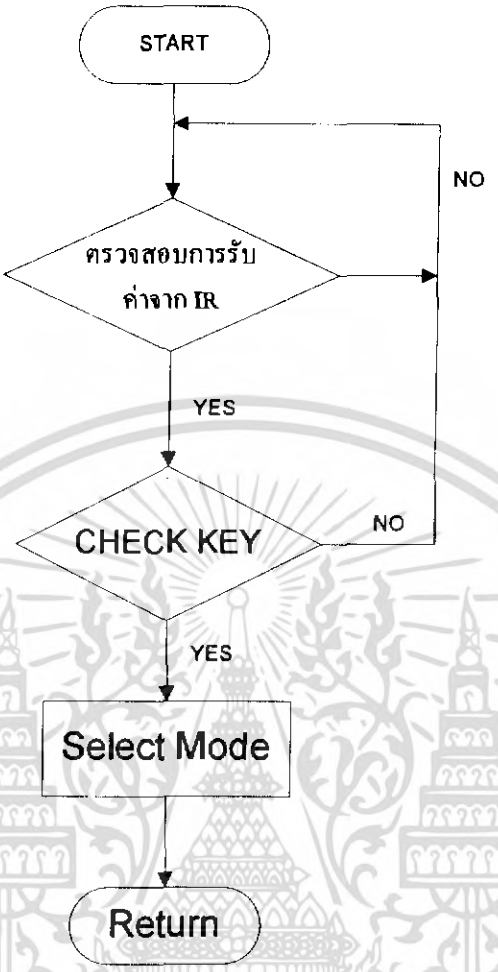
เขียนด้วยภาษาแอสเซมบลี เพื่อให้ทำงานตามที่เราสั่งให้ Software จะทำงานร่วมกับ Hardware กล่าวคือ ในการป้อนคำสั่งต่างๆในรีโมตเพื่อให้สามารถควบคุมหลอดไฟ ก็จะมี Software นี้เป็นคำสั่งผ่านอุปกรณ์ให้ทำงานตามแต่ละขั้นตอนให้เป็นไปอย่างมีระเบียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.1 โฟลวชาร์ตแสดงการทำงานโปรแกรมการกดคีย์เพื่อหรี่ไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2 โฟลวชาร์ตแสดงการทำงานโปรแกรมการเปิด-ปิดไฟ

4.2 คุณลักษณะของอุปกรณ์

1. ใช้กับไฟฟ้า 220 V
2. CPU ตระกูล MCS51 เบอร์ AT89C51
3. สามารถควบคุมหลอดไฟได้ 4 ดวง
4. หลอดไฟที่เลือกใช้ที่กำลังไฟไม่เกิน 100 W
5. รีโมตสามารถควบได้ไม่เกิน 15 เมตร

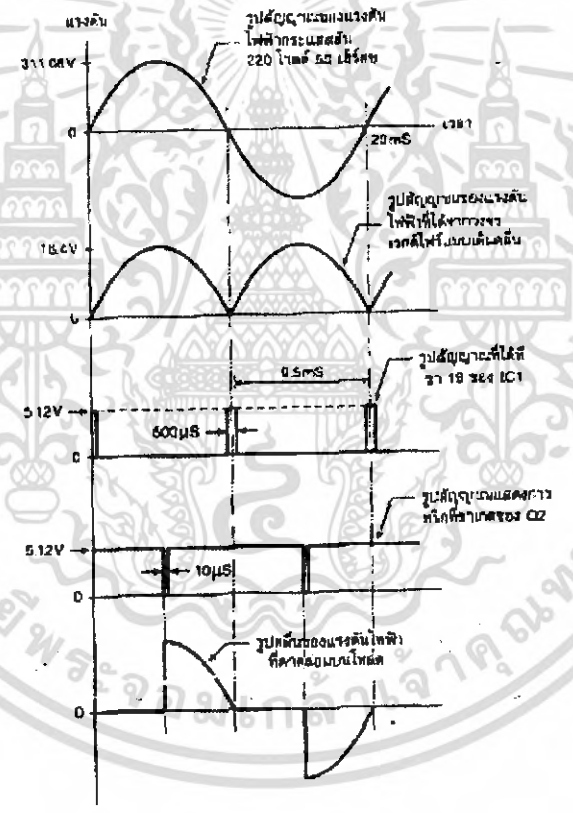
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 หลักการทำงานของอุปกรณ์

หลักการทำงานของวงจรอาศัยตัวไมโครคอนโทรลเลอร์เป็นตัวเลือกมุมเฟสที่ทำการทริกเกอร์ไตรแอก ให้ไตรแอกทำงานตามมุมที่ไมโครคอนโทรลเลอร์กำหนด โดยหัวใจในการทำงานของวงจรนี้ก็คือไมโครคอนโทรลเลอร์เบอร์ 89C51 เป็นตัวทริกเกอร์

หลักการทำงานของไตรแอกเพื่อทำการหรี่ไฟหรี่เร่งไฟได้คือ ต้องมีสัญญาณซิงค์ไตรแอกจึงจะสามารถทำงานได้

การทำงานเริ่มจากสัญญาณไฟสลับจากหม้อแปลงจะผ่านไดโอดบริดจ์ เป็นตัวทำไบแอสทรานซิสเตอร์ทำงานเป็นช่วงๆ เสมือนเป็นพัลส์ช่วงแคบๆ เพื่อที่จะเป็นสัญญาณซิงค์ ส่วนอีกตัวที่สำคัญไม่แพ้กันก็คือ ตัวแยกแรงดันไฟสูงกับไฟต่ำหรือที่เราเรียกว่า ออปโต้ไอโซเลต (Opto Isolate) นอกจากนี้ยังทำหน้าที่เลือกมุมการทริกไตรแอกให้ทำการเปิด-ปิดไฟหรือหรี่ไฟ



รูปที่ 4.3 แสดงรูปคลื่นที่จุดต่างๆของการหรี่ไฟด้วยไมโครคอนโทรลเลอร์

4.4 การทำงานของรีโมตคอนโทรล

รีโมตคอนโทรลประเภทที่ใช้แสงอินฟราเรดทุกชนิดจะมารับส่งข้อมูลหรือสัญญาณพัลส์ผ่านตัวส่งอินฟราเรด โดยใช้ค่าความถี่ในช่วง 30-40 กิโลเฮิรตซ์ เหตุที่ใช้ความถี่สูงเพื่อเป็นการป้องกันแสงจากแหล่งกำเนิดแสงแหล่งอื่นๆมารบกวน โดยสัญญาณที่ส่งมาจากรีโมตอินฟราเรดจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีรูปแบบเหมือนเลขฐานสอง เนื่องจากผู้ผลิตรีโมตคอนโทรลมีอยู่หลายบริษัท ซึ่งจะทำการเข้ารหัสรีโมตแต่ละยี่ห้อไม่เหมือนกันแต่อยู่ในพื้นฐานเดียวกันคือจะใช้เลขฐานองและความกว้าง-ความคบของพัลส์ในช่วง " High " และในช่วง " Low " เป็นตัวกำหนดการทำงานซึ่งโดยสรุปแล้วจะมีความแตกต่างกันประมาณ 3 ประการดังนี้

1. Pulse – Width – Code Signal จะใช้หลักการกำหนดความกว้างของพัลส์แทนลอจิกหรือ 1 โดยทั่วไปแล้วจะให้ลอจิก 0 มีความกว้างของพัลส์ประมาณ 550 ไมโครวินาที และให้ลอจิก 1 มีความกว้างของพัลส์ประมาณ 2200 ไมโครวินาที

2. Space-Code Signal จะใช้หลักการตัวกำหนดช่องว่างของสัญญาณแทนลอจิก 0 หรือ 1 โดยทั่วแล้วจะให้ลอจิก0 มีความกว้างของช่องว่าง ประมาณ 550 ไมโครวินาทีและให้ลอจิก 1 มีความกว้างของช่องว่างประมาณ 1650 ไมโครวินาที

3. Shift – Code Signal จาก 2 แบบข้างต้นที่กล่าวมาแล้วจึงนำมาสู่แบบที่ 3 จะใช้การกำหนดทั้งความกว้างของพัลส์ และความกว้างของช่องว่างของพัลส์เป็นตัวกำหนดลอจิก 0 หรือ 1 โดยถ้าความกว้างของช่องว่างองพัลส์แคบ (ประมาณ 550 ไมโครวินาที) และความกว้างของพัลส์ยาว (ประมาณ 1100 ไมโครวินาที) จะสื่อความหมายว่าเป็นลอจิก 1 แต่ถ้าความกว้างของช่องว่างยาวและดวากว้างของพัลส์แคบจะแทนความหมายว่าเป็นลอจิก 0

4.5 วงจรและการทำงาน

ในรายละเอียดของวงจรที่สมบูรณ์แสดงดังรูปในภาคผนวก ส่วนในการทำงานของโครงงานทำงานได้ 3 แบบ

แบบที่ 1 การเปิด-ปิดไฟอัตโนมัติ ทำได้โดยการกดปุ่มรีโมตตัวที่เราเลือก โดยแต่ละปุ่มสามารถให้ไฟกดติด-กดดับแต่ละดวงไม่เหมือนกัน ถ้าต้องการให้เอาท์พุททั้ง 4 ช่องเป็นแบบกดติด-กดดับก็ให้กดปุ่มเดิมซ้ำ

แบบที่ 2 การหรี่ไฟทำได้โดยการกดคีย์ โดยสามารถให้หลอดไฟทำงานได้ 1 ดวง ถ้าต้องการเพิ่มความสว่างให้กดคีย์ UP ไปเรื่อยๆจนกระทั่งถึงค่าความสว่างมากสุดและถ้าต้องการลดความสว่างลงให้กดคีย์ DOWN ไปเรื่อยๆจนกระทั่งไฟดับ

แบบที่ 3 การเปิดไฟและการปิดไฟ โดยสามารถให้หลอดไฟทำงานได้ 1 ดวง ในขั้นตอนการเปิดไฟจะเหมือนแบบที่ 1 แต่ในขั้นตอนการปิดไฟนั้นไฟจะค่อยๆทำการหรี่ลงจนกระทั่งไฟดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลอง

5.1 สรุปผลการทดลอง

จากการทำ Project นี้ตัวโปรแกรมสามารถควบคุมการจ่ายกระแสไฟฟ้าให้กับโหลด 3 แบบคือ

1. การเปิด-ปิดไฟโดยใช้รีโมต ซึ่งผู้ใช้จะต้องทำการกดปุ่มรีโมตเพื่อให้ไดรแอกมีสถานะเป็น ON หรือ OFF โดยสามารถควบคุมการเปิด-ปิดไฟได้ 4 ดวง

2. การหรี่ไฟโดยใช้สวิตช์เป็นตัวควบคุมความสามารถควบคุมการทำงานของหลอดไฟได้ 1 ดวง

3. การเปิด-ปิดไฟโดยใช้รีโมต แต่ในสถานการณ์ปิดนั้นไฟจะทำการหรี่ลงจนกระทั่งดับ

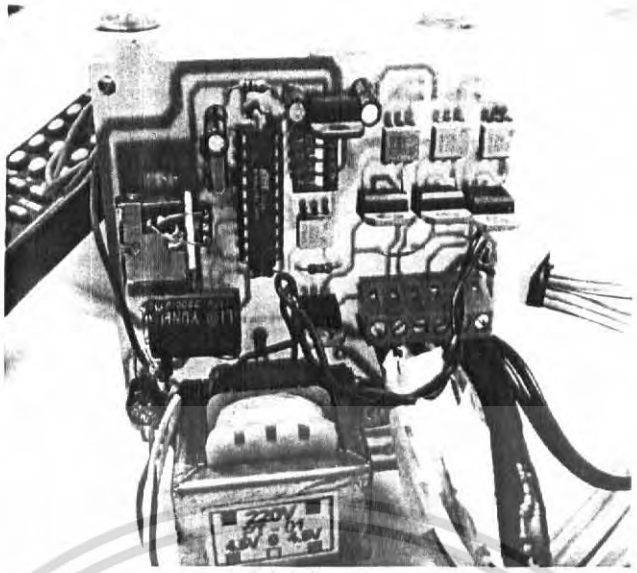
5.2 ปัญหาที่พบ

ปัญหาที่พบมีมากมายโดยปัญหาแรกที่พบได้แก่ขั้นตอนการทดลองการหรี่ไฟโดยใช้รีโมต ซึ่งขณะกด Remote เพื่อควบคุม หลอดจะดับไปเนื่องจากไมโครคอนโทรลเลอร์เสียเวลาไปกับการถอดรหัส Remote จึงทำให้การหรี่ไฟทำได้ไม่ดีนักเราจึงใช้การหรี่ไฟด้วยสวิตช์และการเปิด-ปิดไฟด้วยรีโมตแต่ในการปิดไฟนั้นไฟจะค่อยๆทำการหรี่จนกระทั่งดับเอง

ปัญหาอีกอย่างที่เราพบคือการสร้างมุมทริกยังไม่สมบูรณ์ โดยวงจรสามารถจับค่าได้เฉพาะซีกบวกของไฟฟ้า AC เท่านั้นเราจึงใช้โปรแกรมเป็นตัวการแก้ปัญหาโดยการใช้ Delay เข้ามาช่วยแทน

5.3 รูปของอุปกรณ์การทำงาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.1 รูปแสดงอุปกรณ์ ELECTRONIC ของอุปกรณ์

รูปที่ 5.2 รูปแสดงอุปกรณ์ทั้งหมดของสวิตช์ควบคุมระยะไกลด้วยรหัสอินฟราเรด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ผศ.สมยศ จุณณะปิยะ . การใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51 , 2537

เดชฤทธิ์ มณีธรรม , สำเริง เต็มราม . คัมภีร์ไมโครคอนโทรลลอร์ MCS-51. กรุงเทพ : เคทีพี คอมพ์ แอนด์ คอนซัลท์ , 2548

รศ.ธีรวัฒน์ ประกอบผล . ภาษาแอสเซมบลีสำหรับ MCS-51 . กรุงเทพ : สมาคมส่งเสริมเทคโนโลยี (ไทย- ญี่ปุ่น) , 2546

วรพจน์ กรแก้ววัฒนากุล , ชัยวฒน์ ลิ้มพรจิตรวิไล. เรียนรู้และปฏิบัติการไมโครคอนโทรลเลอร์ . กรุงเทพ : อินโนเวตีฟ เอ็กพอริเมนต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของโปรแกรมการเปิด-ปิดไฟ

```
;_____________________________________________
;
;   REMOTE CONTROL FOR  4 LOADS  MANAGEMENT
;   USE 89C2051 AND SOLID STATE RELAY
;   REMOTE   S O N Y
;   UNIT NUMBER IS SELECTED BY 16 COMBINATION OF DIP SWITCH
;   PUSH 2 NUMBER FOR SELECT THE UNIT NO.
;   AND THEN UP/DOWN [BUTTON] FOR SELECT   'ON'  OR  'OFF'  RESPECTIVELY
;   OR PUSH THE POWER  [BUTTON] MODE ALL THE LOADS  WILL BE SHUT DOWN
WITH IN 5 SECOND
;   WHEN POWER IS ON  THEN  ALL  FOUR LOADS IS ON LIKE AS MAIN SWITCH
;_____________________________________________
;VARIABLE DECLARATION

MIN          EQU 60    ;(120 0.5 SEC)
TIMEINTV          EQU 10    ; 5 SEC
SYNCTIM      EQU 1120  ;1130

POWER        EQU 9AH
MUTE         EQU 92H
DISP         EQU 0D5H
ONE          EQU 10H
TWO          EQU 18H
THREE        EQU 14H
FOUR         EQU 1CH
FIVE         EQU 12H
SIX          EQU 1AH
SEVEN        EQU 16H
EIGHT        EQU 1EH
NINE         EQU 11H
ZERO         EQU 19H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SLEEP       EQU 0D6H
SELUP       EQU 0F2H
SELDW       EQU 0FAH
SELECT      EQU 0F3H
PICMD       EQU 96H
VOLUP       EQU 94H
VOLDW       EQU 9CH
CHUP        EQU 90H
CHDW        EQU 98H
TV_VIDEO            EQU 5AH
;_____________________________________________

;_____________________________________________
;
;   I/O ASSIGNMENT BIT CONTROL
;_____________________________________________
REMOTE      EQU P3.2    ;EXT[0] BIT

GREEN       EQU P3.4
RED         EQU P3.5

LOAD1       EQU P1.0
LOAD2       EQU P1.1
LOAD3       EQU P1.2
LOAD4       EQU P1.3

SEL1        EQU P1.4
SEL2        EQU P1.5
SEL3        EQU P1.6
SEL4        EQU P1.7

;_____________________________________________
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;VARIABLE
;INTERNAL RAM
CNTL        EQU 10H ;BYTE
CNTH        EQU 11H ;BYTE
HEXBUF      EQU 12H ;2 BYTES

REMD0       EQU 14H ;BYTE
REMD1       EQU 15H ;BYTE
UNIT        EQU 16H ;BYTE

MODE        EQU 17H ;BYTE
NUM         EQU 18H ;BYTE
SECFLG      EQU 19H ;BYTE
OPERFLG     EQU 1AH ;BYTE
SEC_CNT     EQU 1BH ;BYTE
MIN_CNT     EQU 1CH ;BYTE
OPERTIME        EQU 1DH ;BYTE
STATION         EQU 1EH ;BYTE
SLEPFLG     EQU 1FH ;BYTE
STACK       EQU 30H
;_____________________________________________
            ORG 0000H
            LJMP START
;_____________________________________________
;
;   INTERRUPT
;   VECTOR
;   ASSIGNMENT
;_____________________________________________
            ORG 0003H
EX0_INT:    JMP EX0_SERV
;_____________________________________________
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ORG 000BH
TF0_INT:        JMP INT0_SERV
;_____________________________________________
                ORG 0013H
EX1_INT:        RETI
;_____________________________________________
                ORG 001BH
TF1_INT:        RETI
;_____________________________________________
                ORG 0023H
SR_INT:         RETI
;_____________________________________________

                ORG 0100H

EX0_SERV:       PUSH ACC
                PUSH DPH
                PUSH DPL
                SETB RS0        ;SEL BANK1

HERE:           JNB REMOTE,HERE
                MOV R7,#20
                DJNZ R7,$
                MOV R2,#12      ;SET 12 BIT DATA FOR SONY
HD_BIT:         NOP             ;JNB REMOTE,EXIT    ;IF = 0 THEN RETURN
                CALL WAIT1200
                JNB REMOTE,REMD_1

REMD_0:         CLR C
                MOV A,REMD0
                RLC A
                MOV REMD0,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV A,REMD1
                RLC A
                MOV REMD1,A
                DJNZ R2,HD_BIT
                CALL DECODE
EXIT:           CLR RS0
                POP DPL
                POP DPH
                POP ACC
                SETB EX0
                RETI
;_____________________________________________
REMD_1:         SETB C
                MOV A,REMD0
                RLC A
                MOV REMD0,A
                MOV A,REMD1
                RLC A
                MOV REMD1,A
                CALL WAIT600
                DJNZ R2,HD_BIT
                CLR RS0
                POP DPL
                POP DPH
                POP ACC
                RETI
;_____________________________________________
WAIT1200:       MOV R7,#3       ;WAIT FOR 1200 uS
W1200:          MOV R6,#183
                DJNZ R6,$
                DJNZ R7,W1200
                RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
WAIT600:    MOV R7,#3       ;WAIT FOR 600 uS
W600:       MOV R6,#90
            DJNZ R6,$
            DJNZ R7,W600
            RET
;_____________________________________________
;
;
;
; T I M E R  [0] INTERRUPT SERVICE ROUTINE
;
;_____________________________________________
INT0_SERV:          PUSH ACC
            PUSH DPH
            PUSH DPL
            SETB RS0      ;SEL BANK1

            MOV DPL,CNTL
            MOV DPH,CNTH
            INC DPTR
            MOV A,DPL
            ORL A,DPH
            JZ SECOND

            MOV CNTL,DPL
            MOV CNTH,DPH

            CLR RS0
            POP DPL
            POP DPH
            POP ACC
            RETI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SECOND:     CALL SECFLAG
            MOV DPTR,#0FFFFH-1800  ;3600 @1 SEC  54000 @15 SEC
            MOV CNTL,DPL
            MOV CNTH,DPH

            CLR RS0
            POP DPL
            POP DPH
            POP ACC
            RETI

SECFLAG:    MOV A,SECFLG
            XRL A,#0FFH
            JZ SEC0
            MOV SECFLG,#0FFH
            CALL CHKTIME
            CLR GREEN
            RET

SEC0:       MOV SECFLG,#0
            SETB GREEN
            RET
;_____________________________________________
INIT_SERIAL:        CPL A
            CLR TR1
            MOV SCON,#01010000B   ;set serial N81
            MOV TMOD,#00100010B
            ANL PCON,#01111111B
            MOV TH0,#0
            MOV TH1,#0FDH          ;0FD = 9600  0FAH  = 4800
            RET
;_____________________________________________
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CHKMODE:    RET
;_____________________________________________________
NUMERIC:            MOV A,OPERFLG
                    XRL A,#0FFH
                    JNZ CHKUNIT
                    MOV A,NUM

                    CJNE A,#1,CHKNUM2
                    JB LOAD1,LD1_OFF
                    SETB LOAD1
                    MOV OPERTIME,#5
                    RET
LD1_OFF:            CLR LOAD1
                    MOV OPERTIME,#5
                    RET

CHKNUM2:    CJNE A,#2,CHKNUM3
                    JB LOAD2,LD2_OFF
                    SETB LOAD2
                    MOV OPERTIME,#5
                    RET
LD2_OFF:            CLR LOAD2
                    MOV OPERTIME,#5
                    RET

CHKNUM3:    CJNE A,#3,CHKNUM4
                    JB LOAD3,LD3_OFF
                    SETB LOAD3
                    MOV OPERTIME,#5
                    RET
LD3_OFF:            CLR LOAD3
                    MOV OPERTIME,#5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                RET

CHKNUM4:        CJNE A,#4,CHKNUMRET
                JB LOAD4,LD4_OFF
                SETB LOAD4
                MOV OPERTIME,#5
                RET
LD4_OFF:          CLR LOAD4
                MOV OPERTIME,#5
                RET

CHKNUMRET: RET

CHKUNIT:          JMP UNITNUMSEL
;_____________________________________________
DELAYD:         MOV R7,#0FFH
DLYD:           MOV R6,#0FFH
                DJNZ R6,$
                DJNZ R7,DLYD
                RET
;_____________________________________________
DECODE:         CALL DELAYD
                MOV A,REMD0          ;DATA REMOTE
                XRL A,REMD1
                MOV REMD1,A

                MOV A,REMDI
DECOD1:         CJNE A,#CHUP,DECOD2
                MOV A,OPERFLG
                XRL A,#0FFH
                JNZ DEC1RET
                CLR LOAD1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CLR LOAD2
                CLR LOAD3
                CLR LOAD4
                MOV OPERTIME,#5
DEC1RET:        RET

DECOD2:         CJNE A,#CHDW,DECOD3
                MOV A,OPERFLG
                XRL A,#0FFH
                JNZ DEC2RET
                SETB LOAD1
                SETB LOAD2
                SETB LOAD3
                SETB LOAD4
                MOV OPERTIME,#5
DEC2RET:        RET

DECOD3:         CJNE A,#POWER,DECOD8
                MOV A,OPERFLG
                XRL A,#0FFH
                JNZ DEC3RET
                MOV SLEPFLG,#0FFH
DEC3RET:        RET
DECOD8:         CJNE A,#ONE,DECOD9
                MOV NUM,#1
                JMP NUMERIC

DECOD9:         CJNE A,#TWO,DECOD10
                MOV NUM,#2
                JMP NUMERIC

DECOD10:        CJNE A,#THREE,DECOD11
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV NUM,#3
                JMP NUMERIC

DECOD11:        CJNE A,#FOUR,DECOD12
                MOV NUM,#4
                JMP NUMERIC

DECOD12:        CJNE A,#FIVE,DECOD13
                MOV NUM,#5
                JMP NUMERIC

DECOD13:        CJNE A,#SIX,DECOD14
                MOV NUM,#6
                JMP NUMERIC

DECOD14:        CJNE A,#SEVEN,DECOD15
                MOV NUM,#7
                JMP NUMERIC

DECOD15:        CJNE A,#EIGHT,DECOD16
                MOV NUM,#8
                JMP NUMERIC

DECOD16:        CJNE A,#NINE,DECOD17
                MOV NUM,#9
                JMP NUMERIC

DECOD17:        CJNE A,#ZERO,DECOD18
                MOV NUM,#0
                JMP NUMERIC

DECOD18:        RET
```

เอกสารนี้มีลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;_____________________________________________
; ********** HTOD SUB **********
; HEX TO DECIMAL
; IN  = DPTR
; OUT = HEXBUF,HEXBUF+1
; REG = A,R0,R1,R2,R3,R4,R5,DPTR

HTOD:       CLR A       ;CLEAR OUTPUT
            MOV HEXBUF,A
            MOV HEXBUF+1,A

            MOV R4,#16  ;SHIFT 16 BIT
HTOD1:      MOV A,DPL
            RLC A
            MOV DPL,A
            MOV A,DPH
            RLC A
            MOV DPH,A

            MOV R5,#2       ;2 BYTE ADD DECIMAL
            MOV R0,#HEXBUF+1 ;INDEX O/P
HTOD2:      MOV A,@R0
            ADDC A,ACC
            DA A
            MOV @R0,A
            DEC R0
            DJNZ R5,HTOD2
            DJNZ R4,HTOD1
            RET
;_____________________________________________
; ********** DTOH SUB **********
; DECIMAL TO HEX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
; IN  = R1,R2,R3
; OUT = DPTR
; REG = A,R0,R1,R2,R3,R4,R5,DPTR

DTOH:       MOV R4,#16
DTOH1:      MOV R5,#3    ;SHIFT & SUB
            MOV R0,#9    ;INDEX TO R1 OF BLANK 1 REG
            CLR C
DTOH2:      MOV A,@R0
            RRC A
            PUSH PSW     ;-[
            JNB ACC.7,DTOH3
            CLR C
            SUBB A,#30H
DTOH3:      JNB ACC.3,DTOH4
            CLR C
            SUBB A,#03H
DTOH4:      MOV @R0,A
            INC R0
            POP PSW      ;-]
            DJNZ R5,DTOH2

            MOV A,DPH
            RRC A
            MOV DPH,A
            MOV A,DPL
            RRC A
            MOV DPL,A
            DJNZ R4,DTOH1
            RET
;_____________________________________________
UNITNUMSEL:MOV R2,#4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV A,NUM
                SWAP A
                MOV NUM,A

                MOV  A,STATION      ;TEMP REGISTER FOR COMPARE WITH DIP
SWITCH
                MOV R4,A

                MOV R3,#0
                MOV DPH,R3
                MOV DPL,R4
                CALL HTOD

                MOV R3,HEXBUF
                MOV R4,HEXBUF+1

UNITLP:         MOV A,NUM
                RLC A
                MOV NUM,A
                MOV A,R4
                RLC A
                MOV R4,A
                MOV A,R3
                RLC A
                MOV R3,A
                DJNZ R2,UNITLP

                MOV HEXBUF,#0
                MOV HEXBUF+I,R4

                MOV RI,#0
                MOV R2,#0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV R3,HEXBUF+1

                CALL DTOH
                MOV R4,DPL
                MOV R3,DPH
                MOV STATION,R4

CMP:            MOV A,P1
                CPL A
                SWAP A
                ANL A,#0FH
                XRL A,R4
                JNZ UNITSELRET
                MOV OPERFLG,#0FFH       ;WHEN SEL NUMBER IS EQUAL THEN
OPERATING
UNITSELRET: RET
;_____________________________________________
CHKTIME:    MOV A,SLEPFLG
                XRL A,#0FFH
                JZ CHKSLP

                MOV A,OPERFLG
                XRL A,#0FFH
                JNZ CHKTIMRET
                CLR RED
                INC OPERTIME
                MOV A,OPERTIME
                XRL A,#15
                JNZ CHKTIMRET
                SETB RED
                MOV OPERTIME,#0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV OPERFLG,#0
CHKTIMRET:  RET
;_____________________________________________
CHKSLP:         JB RED,CHKSLP1
                SETB RED
                JMP CHKSLPAPP
CHKSLP1:         CLR RED
                JMP CHKSLPAPP

CHKSLPAPP:  INC OPERTIME
                MOV A,OPERTIME
                XRL A,#20
                JNZ CHKTIMRET
                SETB RED
                SETB LOAD1
                SETB LOAD2
                SETB LOAD3
                SETB LOAD4
                MOV OPERTIME,#0
                MOV OPERFLG,#0
                MOV SLEPFLG,#0
                RET
;_____________________________________________
START:          MOV R7,#0FFH
STRDLY:         MOV R6,#0FFH
                DJNZ R6,$
                DJNZ R7,STRDLY

                MOV IE,#0
                MOV SP,#STACK

                CALL INIT_SERIAL    ;TIMER/SERIAL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV DPTR,#0FFFFH-1800   ;500 mSEC
                MOV CNTL,DPL
                MOV CNTH,DPH
                MOV OPERFLG,#0
                MOV STATION,#0
                MOV OPERTIME,#0
                MOV SLEPFLG,#0
                CLR LOAD1
                SETB LOAD2
                CLR LOAD3
                SETB LOAD4
                SETB TR0
                SETB TR1
                SETB EX0        ;EXTR INT0
                SETB IT0        ;EDGE DETECT
                SETB ET0        ;T0 INT ENA
                SETB EA

;_____________________________________________________
OPER:           CALL CHKMODE
                JMP OPER
;_____________________________________________________

                END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของโปรแกรมการกดคีย์เพื่อทำการหรี่ไฟ

```
;_____________________________________________
;
;   REMOTE CONTROL FOR  4 LOADS  MANAGEMENT
;   USE 89C2051 AND SOLID STATE RELAY
;   REMOTE   S O N Y
;   UNIT NUMBER IS SELECTED BY 16 COMBINATION OF DIP SWITCH
;   PUSH 2 NUMBER FOR SELECT THE UNIT NO.
;   AND THEN UP/DOWN [BUTTON] FOR SELECT   'ON'  OR  'OFF'  RESPECTIVELY
;   OR PUSH THE POWER  [BUTTON] MODE ALL THE LOADS  WILL BE SHUT DOWN
WITH IN 5 SECOND
;   WHEN POWER IS ON  THEN  ALL  FOUR LOADS IS ON LIKE AS MAIN SWITCH
;   VOLUME  +  IS  LIGHT UP
;   VOLUME  -   IS  LIGHT DOWN
;_____________________________________________
;VARIABLE DECLARATION

MIN          EQU 60    ;(120 0.5 SEC)
TIMEINTV          EQU 10    ; 5 SEC
SYNCTIM      EQU 1120  ;1130

POWER        EQU 9AH
MUTE         EQU 92H
DISP         EQU 0D5H
ONE          EQU 10H
TWO          EQU 18H
THREE        EQU 14H
FOUR         EQU 1CH
FIVE         EQU 12H
SIX          EQU 1AH
SEVEN        EQU 16H
EIGHT        EQU 1EH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
NINE        EQU 11H
ZERO        EQU 19H
SLEEP       EQU 0D6H
SELUP       EQU 0F2H
SELDW       EQU 0FAH
SELECT      EQU 0F3H
PICMD       EQU 96H
VOLUP       EQU 94H
VOLDW       EQU 9CH
CHUP        EQU 90H
CHDW        EQU 98H
TV_VIDEO            EQU 5AH
;_____________________________________________

;_____________________________________________
;
;   I/O ASSIGNMENT BIT CONTROL
;_____________________________________________
SW1         EQU P1.7
SW2         EQU P1.6

REMOTE      EQU P3.2    ;EXT[0] BIT

GREEN       EQU P3.4
RED         EQU P3.5

LOAD1       EQU P1.0
LOAD2       EQU P1.1
LOAD3       EQU P1.2
LOAD4       EQU P1.3

SEL1        EQU P1.4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SEL2            EQU P1.5
SEL3            EQU P1.6
SEL4            EQU P1.7

;_____________________________________________
;VARIABLE
;INTERNAL RAM
CNTL        EQU 10H ;BYTE
CNTH        EQU 11H ;BYTE
HEXBUF      EQU 12H ;2 BYTES

REMD0       EQU 14H ;BYTE
REMD1       EQU 15H ;BYTE
UNIT        EQU 16H ;BYTE

MODE        EQU 17H ;BYTE
NUM         EQU 18H ;BYTE
SECFLG      EQU 19H ;BYTE
OPERFLG     EQU 1AH ;BYTE
SEC_CNT     EQU 1BH ;BYTE
MIN_CNT     EQU 1CH ;BYTE
OPERTIME    EQU 1DH ;BYTE
STATION             EQU 1EH ;BYTE
SLEPFLG             EQU 1FH ;BYTE
PHASE       EQU 20H ;BYTE
DIMFLG              EQU 21H ;BYTE
HALFCYC             EQU 22H ;BYTE
STEPLIGHT   EQU 23H ;BYTE
STACK       EQU 30H
;_____________________________________________
            ORG 0000H
            LJMP START
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;_____________________________________________
;
;    I N T E R R U P T
;    V E C T O R
;    A S S I G N M E N T
;_____________________________________________
          ORG 0003H
EX0_INT:      RETI
;_____________________________________________
          ORG 000BH
TF0_INT:      RETI
;_____________________________________________
          ORG 0013H
EX1_INT:      JMP EX1_SERV
;_____________________________________________
          ORG 001BH
TF1_INT:      RETI
;_____________________________________________
          ORG 0023H
SR_INT:       RETI
;_____________________________________________

          ORG 0100H


WAIT1200:     MOV R7,#3       ;WAIT FOR 1200 uS
W1200:        MOV R6,#183
              DJNZ R6,$
              DJNZ R7,W1200
              RET

WAIT600:      MOV R7,#3       ;WAIT FOR 600 uS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
W600:       MOV R6,#90
            DJNZ R6,$
            DJNZ R7,W600
            RET
;_____________________________________________
WAIT300:    MOV R7,#3       ;WAIT FOR 300 uS
W300:       MOV R6,#45
            DJNZ R6,$
            DJNZ R7,W300
            RET
;_____________________________________________
EX1_SERV:   PUSH ACC
            PUSH DPH
            PUSH DPL
            SETB RS0        ;SEL BANK1

                SETB LOAD1
                MOV A,STEPLIGHT
                MOV R2,A
PHDLY:              CALL WAIT600
                DJNZ R2,PHDLY
          CLR LOAD1
                CALL WAIT300
                SETB LOAD1


              CLR RS0
            POP DPL
            POP DPH
            POP ACC
            RETI
;_____________________________________________
START:      MOV R7,#0FFH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
STRDLY:     MOV R6,#0FFH
            DJNZ R6,$
            DJNZ R7,STRDLY

            MOV IE,#0
            MOV SP,#STACK

            MOV DPTR,#0FFFFH-1800   ;500 mSEC
            MOV CNTL,DPL
            MOV CNTH,DPH
                MOV OPERFLG,#0
                MOV STATION,#0
                MOV OPERTIME,#0
                MOV SLEPFLG,#0
                MOV STEPLIGHT,#1
                SETB LOAD1
                SETB LOAD2
                SETB LOAD3
                SETB LOAD4
                SETB EX1
                SETB IT1
           SETB EA

;_____________________________________________________
OPER:       JNB SW1,INCSTEP
                JNB SW2,DECSTEP
                JMP OPER

INCSTEP:    SETB SW1
                INC STEPLIGHT
                MOV A,STEPLIGHT
                XRL A,#40
```

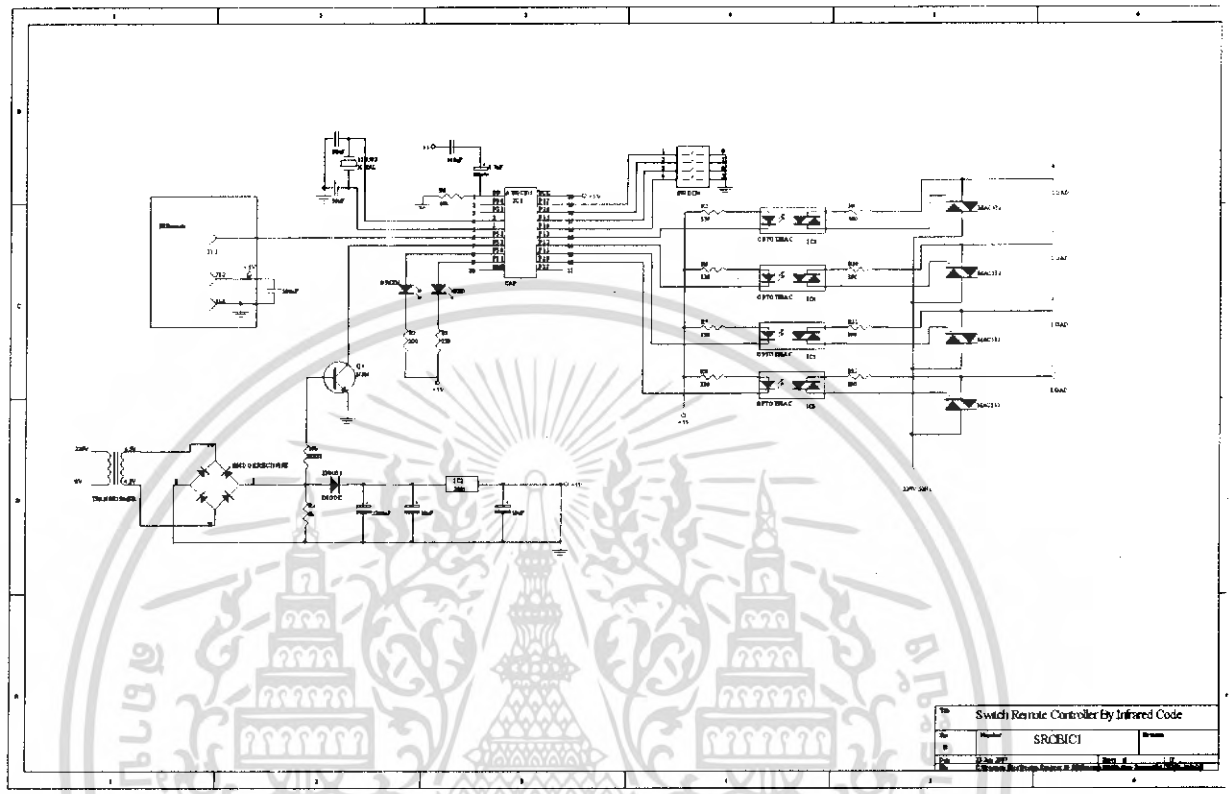
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JNZ INCSTP_RET
                MOV STEPLIGHT,#39
INCSTP_RET:     CALL DELAYSTEP
                JMP OPER


DECSTEP:        SETB SW2
                DEC STEPLIGHT
                MOV A,STEPLIGHT
                XRL A,#0H
                JNZ DECSTP_RET
                MOV STEPLIGHT,#1
DECSTP_RET:     CALL DELAYSTEP
                JMP OPER

DELAYSTEP:      MOV R7,#2FH
DLYSTP:         MOV R6,#0FFH
          DJNZ R6,$
          DJNZ R7,DLYSTP
          RET
;_____________________________________________

                END
```

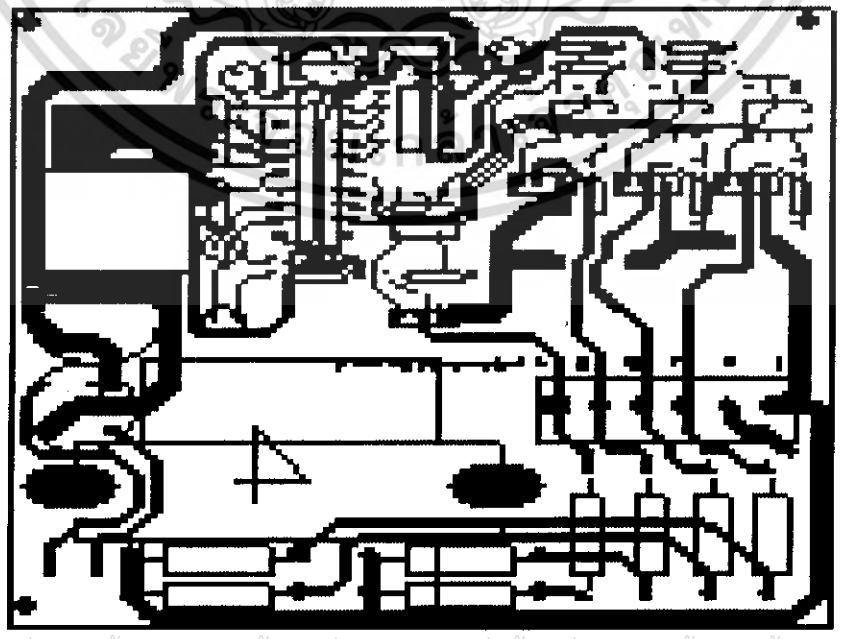
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรสมบูรณ์ของสวิทช์ควบคุมระยะไกลด้วยรหัสอินฟราเรด

(Switch Remote Controller By Infrared Code)



ลายทองแดงของวงจรสวิทช์ควบคุมระยะไกลด้วยรหัสอินฟราเรด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้