สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

แป้นแสดงผลการแข่งขันบาสเกตบอลแบบกึ่งอัตโนมัติ

SEMI-AUTOMATIC SCORE BOARD BASKETBALL CONTROLLER

T034106

โดย

นาย เมษา ใช่พาณิชย์ตระกูล

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมคอมพิวเตอร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เลขหมู่.....
เลขทะเบียน.....34106
วัน, เดือน, ปี.-5 ต.ค. 2542

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ ปีการศึกษา 2541

ภาควิชา วิศวกรรมคอมพิวเตอร์

คณะ วิศวกรรมศาสตร์สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง แป้นแสดงผลการแข่งขันบาสเกตบอลแบบกึ่งอัตโนมัติ

SEMI-AUTOMATIC SCORE BOARD BASKETBALL CONTROLLER

ผู้จัดทำ นาย เมษา ใช่พาณิชย์ตระกูล 35104331

(อาจารย์ที่ปรึกษา)

(อาจารย์ ประสาร ตังติสานนท์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แป้นแสดงผลการแข่งขันบาสเกตบอลแบบกึ่งอัตโนมัติ

ผู้จัดทำ นาย เมษา ใช่พาณิชย์ตระกูล 35104331

อาจารย์ที่ปรึกษา อาจารย์ ประสาร ตังติสานนท์

ปีการศึกษา 2541

บทคัดย่อ

โครงงานนี้เป็นการนำความรู้เกี่ยวกับระบบไมโครคอนโทรลเลอร์มาใช้ในการควบคุมระบบแสดงผลการแข่งขันบาสเกตบอล โดยหลักการจะแสดงข้อมูลที่สำคัญในการแข่งขันได้แก่ เวลา,คะแนน,ฟาว์ลรวมของทีม และ การขอเวลานอก ในการรับข้อมูลดังกล่าวนี้จะรับผ่านทางบอร์ดควบคุม เพื่อส่งสัญญาณให้ไมโครคอนโทลเลอร์ประมวลผลและควบคุมการสร้างสัญญาณเพื่อแสดงผลข้อมูลทั้งหมดบน 7-segment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SEMI-AUTOMATIC SCORE BOARD BASKETBALL CONTROLLER

NAME	MR. MESA	CHAIPARNITTARGOON	35104331
ADVISOR	MR. PRASARN	TANGTISANON	
	1998		

ABSTRACT

This thesis study about microcontroller use to be control monitor system of basketball competition. The principal of this thesis is input data via control board and generate control signal to display all important data in competition , time , score , fouls team and time out on 7-segment.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทที่ 1	บทนำ	
บทที่ 2	หลักการทำงานและทฤษฎีเบื้องต้น	2
2.1	โครงสร้างทางสถาปัตยกรรมของไมโครคอนโทรลเลอร์ MCS-51	2
2.1.1	การจัดขาลักษณะภายนอกของ MCS-51	4
2.1.2	การจัดการทางสถาปัตยกรรม	7
2.1.3	การจัดการหน่วยความจำ	9
2.1.4	เนื้อที่หน่วยความจำโปรแกรม	9
2.1.5	เนื้อที่หน่วยความจำข้อมูล	10
2.1.6	โครงสร้างทางอินเตอร์รัพต์ MCS-51	13
2.1.7	โครงสร้างระดับความสำคัญในการบริการอินเตอร์รัพต์ (Priority Level Structure)	15
2.1.8	ตัวจับเวลา/ตัวนับ (Timer/Counter)	17
2.1.9	การเชื่อมต่อแบบอนุกรม	22
2.2	หน่วยแสดงผล 7-segment	26
2.2.1	ขาสัญญาณของ 7-segment	26
2.2.2	โครงสร้างการทำงาน 7-segment	26
บทที่ 3	การออกแบบการทำงาน	30
3.1	การออกแบบด้านฮาร์ดแวร์	30
3.1.1	วงจรส่วนควบคุม	31
3.1.2	วงจรกำเนิดแรงดัน	34
3.2	การออกแบบด้านซอร์ฟแวร์	35
3.2.1	Main Program	35
3.2.2	Maintenanace Mode	36
3.2.3	Interrupt Service Routime Timer 0	40
3.2.4	Interrupt Service Routime Timer 1	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4	การทำงานของแป้นแสดงผลการแข่งขันบาสเกตบอล	42
4.1	การตรวจสอบระบบก่อนการใช้งาน	42
4.2	การควบคุมการทำงานเมื่อระบบพร้อม	42
4.2.1	การควบคุมการทำงานด้านคะแนนการแข่งขัน	42
4.2.2	การควบคุมการทำงานด้านฟาล์วรวมของแต่ละทีม	42
4.2.3	การควบคุมการทำงานด้านการขอเวลานอกของแต่ละทีม	42
4.2.4	การควบคุมการทำงานด้านเวลาในการแข่งขัน	44
บทที่ 5	ผลการทดลอง	45
5.1	การทดลองใช้งานแป้นแสดงผลการแข่งขันบาสเกตบอล	45
5.1.1	การทดลองควบคุมด้านเวลาในการแข่งขัน	45
5.1.2	การทดลองควบคุมด้านคะแนนการแข่งขัน	45
5.1.3	การทดลองควบคุมด้านการทำฟาล์วรวม	45
5.1.4	การทดลองควบคุมด้านการขอเวลานอก	45
5.2	ปัญหาที่พบ	46
5.3	แนวทางการพัฒนา	46
5.4	บทสรุป	47
ภาคผนวก		
กิตติกรรมประกาศ		
บรรณานุกรม		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2-1 ลักษณะการจัดวางขาและสัญลักษณ์ทางตรรกของ MCS-51	4
รูปที่ 2-2 โครงสร้างทางสถาปัตยกรรมภายในของ MCS-51	7
รูปที่ 2-3 แสดงแผนที่ของหน่วยความจำโปรแกรม	10
รูปที่ 2-4 แสดงแผนที่ของหน่วยความจำข้อมูลและการกำหนดตำแหน่งบิต	11
รูปที่ 2-5 แสดงตำแหน่งของรีจิสเตอร์ SFR และบิตแอดเดรสของ SFR	12
รูปที่ 2-6 แหล่งกำเนิดสัญญาณอินเตอร์รัพต์ทั้ง 5 ชนิดที่สามารถรับได้	13
รูปที่ 2-7 รีจิสเตอร์ใช้งานเฉพาะ IE	15
รูปที่ 2-8 รีจิสเตอร์ใช้งานเฉพาะ IP	16
รูปที่ 2-9 แสดงการทำงานในโหมด 0 ของตัวจับเวลา/ตัวนับ 1 ขนาด 13 บิต	18
รูปที่ 2-10 ตัวจับเวลา/ตัวนับ 1 ทำงานในโหมด 2 แบบโหลดใหม่ 8 บิต	19
รูปที่ 2-11 ใช้ตัวจับเวลา/ตัวนับ 1 ในโหมด 3 เป็นกลุ่มตัวนับขนาด 8 บิต	20
รูปที่ 2-12 7-segment	26
รูปที่ 2-13 คาโทดร่วม	26
รูปที่ 2-14 อาโนดร่วม	27
รูปที่ 2-15 การสแกน 7-segment	29
รูปที่ 3-1 แสดงบล็อกไดอะแกรมการทำงานของระบบ	30
รูปที่ 3-2 แสดงวงจรบอร์ดควบคุม	33
รูปที่ 3-3 วงจรกำเนิดแรงดัน	34
รูปที่ 3-4 แสดง Flow Chart ส่วน Main Program	35
รูปที่ 3-5 แสดง Flow Chart ส่วน Maintenance Mode	38
รูปที่ 3-6 แสดง Flow Chart Interrupt Service Routine T0	40
รูปที่ 3-7 แสดง Flow Chart Interrupt Service Routine T1	41
รูปที่ 4-1 แสดงตำแหน่งและชื่อของแต่ละสวิทซ์	43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

		หน้า
ตาราง 2-1	แสดงรายละเอียดของไมโครคอนโทรลเลอร์ ตระกูล MCS-51	3
ตาราง 2-2	รายละเอียดการทำงานของขา P 3.0-P 3.7	5
ตาราง 2-3	แสดงตำแหน่งเริ่มต้นของโปรแกรมบริการอินเตอร์รัพต์	10
ตาราง 2-4	ระดับความสำคัญในการบริการอินเตอร์รัพต์	17
ตาราง 2-5	TMOD : Timer/Counter Mode Control Register	20
ตาราง 2-6	TCON : Timer/Counter Control Register	21
ตาราง 2-7	SCON : รีจิสเตอร์ควบคุมพอร์ทอนุกรม	23
ตาราง 2-8	รายการอัตราบอดที่ใช้ตัวจับเวลา 1	25
ตาราง 2-9	การกำหนดค่าของเซกเมนต์ในการแสดงผลตัวเลข 0-9	27
ตาราง 2-10	การกำหนดค่าของเซกเมนต์ในการแสดงผลตัวเลข 0-9	28
ตาราง 3-1	แสดงการทำงานของอินพุตพอร์ท 8051	31
ตาราง 3-2	แสดงการทำงานของเอาท์พุตพอร์ท 8051	32
ตาราง 3-3	แสดงลำดับการทำงานของ Maintenance Mode	39
ตาราง 4-1	แสดงชื่อและหน้าที่การทำงานของสวิทซ์ในการจับเวลา	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่1
บทนำ

โครงงานชุดนี้จัดทำขึ้นเพื่อมีจุดประสงค์ในการนำไปใช้ในการควบคุมและแสดงผลการแข่งขันบาสเกตบอล เนื่องจากในปัจจุบันการควบคุมและแสดงผลการแข่งขันบาสเกตบอลที่มีอยู่โดยทั่วไป รวมถึงที่มีใช้อยู่ในชมรมบาสเกตบอล สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังขณะนี้ยังไม่ได้มาตรฐาน ทำให้พบปัญหาบ่อยครั้งในระหว่างทำการแข่งขัน เช่น ปัญหาทางด้านเวลาการแข่งขัน,คะแนน,จำนวนครั้งในการฟาล์ว เป็นต้น อีกทั้งเป็นควบคุมและแสดงผลการแข่งขันบาสเกตบอลที่มีในท้องตลาดขณะนี้มีราคาค่อนข้างแพงมาก จึงไม่สามารถจัดหามาใช้งานได้ง่าย ดังนั้นโครงงานชุดนี้จึงสามารถที่จะแก้ไขปัญหาดังกล่าวได้

โดยหลักการนั้นคือจะใช้ไมโครคอนโทรลเลอร์ในการควบคุมและแสดงผลการแข่งขันบาสเกตบอล สามารถจำแนกข้อมูลที่จะแสดงได้ดังนี้ เวลาในการแข่งขัน,คะแนนของแต่ละทีม, จำนวนครั้งในการทำฟาล์วของแต่ละทีม และจำนวนครั้งในการขอเวลานอก

ส่วนควบคุมจะรับสัญญาณจากบอร์ดควบคุม เพื่อส่งสัญญาณให้ไมโครคอนโทรลเลอร์ประมวลผล และควบคุมการสร้างสัญญาณไว้ให้กับส่วนแสดงผลจะรับสัญญาณจากไมโครคอนโทรลเลอร์เพื่อนำไปแสดงผลข้อมูลในการแข่งขันดังที่กล่าวมาแล้วนั้น ผ่านทาง 7-segment

โครงงานชุดนี้อาจจะยังไม่สมบูรณ์ที่สุดซึ่งทางผู้จัดทำหวังว่าคงมีผู้ที่สนใจได้นำหลักการนี้ไปพัฒนาให้เกิดประโยชน์ต่อส่วนรวมและเป็นการพัฒนาเทคโนโลยีให้ทันสมัยยิ่งขึ้นต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
หลักการทำงานและทฤษฎีเบื้องต้น

หลักการทำงานของแป้นควบคุมและแสดงผลการแข่งขันบาสเกตบอล อยู่ที่การรับข้อมูลจากบอร์ดควบคุมเพื่อส่งสัญญาณให้ไมโครคอนโทรลเลอร์ประมวลผลและควบคุมการสร้างสัญญาณให้กับส่วนแสดงผล เพื่อแสดงผลการแข่งขันบาสเกตบอลออกทาง 7-segment โดยข้อมูลที่จะรับและแสดงผลนั้นสามารถแบ่งได้เป็น เวลาในการแข่งขัน,คะแนนของแต่ละทีม,จำนวนครั้งในการฟาว์ลของแต่ละทีม และจำนวนครั้งในการขอเวลานอกของแต่ละทีม

โดยการทำงานส่วนใหญ่จะถูกควบคุมโดยไมโครคอนโทรลเลอร์ ซึ่งสามารถติดต่อกับอุปกรณ์รอบข้างโดยใช้ 8255 จำนวน 2 ตัวเป็นอุปกรณ์ควบคุมการทำงานอีกที ทั้งในการรับและแสดงข้อมูลการแข่งขัน

2.1 โครงสร้างทางสถาปัตยกรรมของไมโครคอนโทรลเลอร์ MCS-51

ลักษณะทั่วๆ ไปของ MCS-51 จะประกอบด้วย

1. สร้างโดยใช้ HMOS และ CHMOS เทคโนโลยีและการทำงานด้วยแหล่งจ่ายไฟขนาด 5 โวลต์เพียงแหล่งเดียว
2. ซีพียูมีขนาด 8 บิต
3. มีวงจรออสซิลเลเตอร์ และวงจรนาฬิกาบนชิพ
4. ชุดแบงก์ (BANK) รีจิสเตอร์มี 4 ชุด แต่ละชุดมีรีจิสเตอร์ 8 ตัวทำงาน
5. มีตัวจับเวลา/ตัวนับขนาด 16 บิต 2 ชุด และสำหรับเบอร์ 8032/8052 มี 3 ชุด
6. มีพอร์ตอินพุทเอาท์พุทแบบขนาน 2 ทิศทางจำนวน 4 พอร์ต พอร์ตละ 8 บิต รวมทั้งหมดเป็น 32 เส้นแต่จะเหลือเพียง 16 เส้นสำหรับเบอร์ 8031 อีก 16 เส้นใช้ในการเข้าถึงทางแอดเดรสและข้อมูล
7. พอร์ตแบบอนุกรมสามารถที่จะใช้โปรแกรมการรับส่งข้อมูลแบบฟูลดูเพล็กซ์ ที่ความเร็วสูง
8. หนึ่งวัฏจักรคำสั่งจะใช้เวลา 1 ไมโครวินาที ด้วยการใช้คริสตอล 12 เมกกะเฮิรตซ์
9. แอดเดรสข้อมูลภายนอกได้ 64 กิโลไบต์
10. แอดเดรสโปรแกรมภายนอกได้ 64 กิโลไบต์
11. สามารถกำหนดเลขที่อยู่ข้อมูลขนาดไบต์หรือบิตได้โดยตรง
12. มีซอฟต์แวร์แฟล็กสำหรับผู้ใช้ที่จะกำหนดเองได้ถึง 128 ตำแหน่งบิต
13. โครงสร้างอินเทอร์รัปต์ทำได้ 5 แหล่ง และ 6 แหล่งสำหรับ 8032/8052 พร้อมด้วยการจัดไพรโอริตี้ (Priority) ได้ 2 ระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

14. ตัวโปรเซสเซอร์สามารถใช้งานแบบบูลีน (Boolean) ได้สำหรับการใช้งานควบคุม
15. มีคำสั่งคูณและหารทางฮาร์ดแวร์ทำได้ภายใน 4 ไมโครวินาที
16. ตัวเลขทางคณิตศาสตร์ ใช้ได้ทั้งแบบไบนารีและเดซิมัล
17. การใช้พื้นที่สแต็กสำหรับโปรแกรมย่อยต่างๆ ทำได้กว้างขึ้น

ไมโครคอนโทรลเลอร์ ตระกูล MCS-51 จะมีทั้งแบบ มีหน่วยความจำโปรแกรมภายในตัว หรือไม่มีบนชิพเดียวกันและจะมีตำแหน่งที่ขาเหมือนกัน ตารางที่ แสดงถึงตารางรายละเอียดของเบอร์ต่างๆ ในตระกูล MCS-51 ที่มีจำหน่ายในท้องตลาด

8751H อยู่ในกลุ่มรุ่นเดียวกับ 8051AH ที่เราสามารถโปรแกรมได้ด้วยระบบไฟและสามารถลบโปรแกรมออกได้ด้วยแสงอุลตร้าไวโอเลต นอกเหนือจากไอซีที่แสดงในตารางที่ ที่ใช้เทคโนโลยี HMOS แล้วยังมีตระกูลอื่นที่ใช้เทคโนโลยี CHMOS ที่ประหยัดพลังงานได้มากกว่า 4 เท่าของ HMOS ที่มีจำหน่ายขณะนี้ คือเบอร์ 80C51, 80C31 และ 87C51

เบอร์	หน่วยความจำภายใน		ตั้งเวลา /	อินเตอร์รัพต์
	โปรแกรม	ข้อมูล	ตัวนับจำนวน	
8052 AH	8K * 8 ROM	256 * 8RAM	3 * 16 BIT	6
8051 AH	4K * 8 ROM	128 * 8RAM	2 * 16 BIT	5
8051	4K * 8 ROM	128 * 8 RAM	2 * 16 BIT	5
8032 AH	ไม่มี ROM	256 * 8 RAM	2 * 16 BIT	5
8031 AH	ไม่มี ROM	128 * 8 RAM	2 * 16 BIT	5
8031	ไม่มี ROM	128 * 8 RAM	2 * 16 BIT	5
8751 H	4K * EPROM	128 * 8 RAM	2 * 16 BIT	5
8751 H-12	4K * EPROM	128 * 8 RAM	2 * 16 BIT	5

ตารางที่ 2.1 แสดงรายละเอียดของไมโครคอนโทรลเลอร์ ตระกูล MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 การจัดขาลักษณะภายนอกของ MCS-51

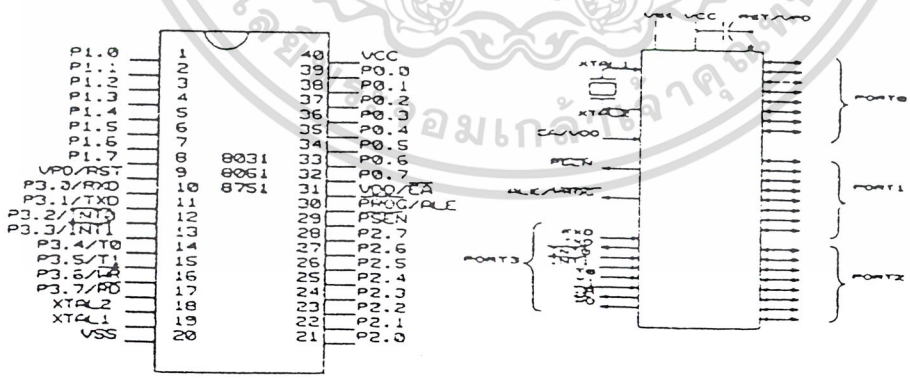
รูปที่ 2.1 แสดงการจัดขาตามลักษณะภายนอกของชิพ MCS-51 ซึ่งจะมีการแบ่งกลุ่มการจัดการขาตามสถาปัตยกรรมของ MCS-51 อยู่ 4 กลุ่ม คือ

1. กลุ่มขารับแหล่งจ่ายไฟฟ้า และสัญญาณนาฬิกา
2. กลุ่มขาแอดเดรสและข้อมูล
3. กลุ่มขาควบคุม
4. กลุ่มขาพอร์ตแบบขนานและอนุกรม

ขาบางขาจะมีหน้าที่ได้สองหน้าที่ขึ้นอยู่กับการติดตั้งด้วยซอร์ฟแวร์หรือฮาร์ดแวร์ ซึ่งมีรายละเอียดดังนี้คือ

- ขา V_{ss} (ขา 20) เป็นขาสำหรับต่อลงดิน

- ขา V_{cc} (ขา 40) เป็นขาที่ต่อแรงดันไฟกระแสตรงขนาด 5 โวลต์ และใช้สำหรับการโปรแกรมแบบ Open Drain Bidirectional สามารถที่จะรับโหลดทีทีแอลได้ 8 ตัว การเขียนค่า "1" ไปที่พอร์ตนี้ จะเป็นการปล่อยลอย (Float) ขาของพอร์ตนี้ ทำให้มันทำงานเป็นอินพุตมีสถานะอิมพีแดนซ์สูง ในการให้พอร์ตนี้บริการแบบไอโอ พอร์ต 0 จะทำงานเป็นมัลติเพล็กซ์ด้วยสัญญาณแอดเดรสไบต์ต่ำกับบัสข้อมูล สำหรับการใช้งานด้านหน่วยความจำภายนอก ในการใช้งานแบบนี้จะใช้ลักษณะภายในเป็นตัวพูลอัพ พอร์ต 0 ยังใช้งาน เป็นตัวส่งข้อมูลออกทางพอร์ตนี้ เมื่อใช้บริการทางด้านการตรวจสอบโปรแกรม ROM ภายใน และการโปรแกรมตัวอีพรอมภายในถ้าใช้งานในลักษณะนี้การพูลอัพภายนอกจะต้องต่อด้วยค่า 10 กิโลโอห์ม



รูปที่ 2.1 ก.)ลักษณะการจัดวางขาของ MCS-51 ข.)สัญลักษณ์ทางตรรกของ MCS-51

- ขา Port 1 (P1.0-P1.7) (ขา 1-8) เป็นพอร์ตไอโอ 8 บิตแบบ Open Drian Bidirectional พร้อมด้วยพูลอัพภายใน ถ้าเป็นพอร์ตเอาท์พุต บัฟเฟอร์สามารถขับโหลดทีทีแอล ตระกูลแอลเอสได้ 4 ตัว พอร์ต 1 เมื่อถูกเขียนค่า "1" ด้วยโปรแกรม มันจะมีสถานะสูงด้วยการพูลอัพภายใน การให้

สถานะเช่นนี้จะเป็นการกำหนดใช้งานพอร์ตนี้ให้เป็นอินพุต ขณะที่พอร์ต 1 เป็นอินพุตการให้สัญญาณต่ำลงจะเป็นการจ่ายกระแสออกเนื่องจากการพูลอัพภายใน ในเบอร์ 8052 ขา P1.0 และ P1.7 จะใช้งานเป็น T2 และ T2EX โดยขา T2 จะทำหน้าที่รับสัญญาณจากภายนอกให้ตัวตั้งเวลา 2 ทำงาน และขา T2EX จะเป็นอินพุตผ่านเข้าตัวตั้งเวลา 2 ถูกกระตุ้นให้ทำงานแบบปกติตามโปรแกรมที่ตั้งไว้ หรือแบบแคบเจอร์ (Capture)

- ขา Port 2 (P2.0-P2.7) (ขา 21-28) เป็นพอร์ตไอโอ 8 บิตแบบ Open Drian Bidirectional ด้วยการพูลอัพภายในพอร์ตที่ทำหน้าที่เป็นบัฟเฟอร์เอาท์พุตสามารถจ่ายโหลดทีทีแอลตระกูลแอลเอสไอได้ 4 ตัวพอร์ตจะถูกใช้งานเป็นตัวส่งแอดเดรสไบต์สูงด้วย เมื่อใช้งานร่วมกับหน่วยความจำภายนอก เพื่อให้สามารถอ้างแอดเดรสได้ถึง 16 บิต ด้วยการใช้งานแบบนี้มันจะมีพูลอัพภายใน ที่ช่วยในการส่งค่า "1" ได้ระดับที่แน่นอน นอกจากการใช้งานสำหรับแอดเดรสอันดับสูงใช้เป็นขาควบคุมในการใช้งานตรวจสอบ และเขียนโปรแกรมเบอร์ 8751 และตรวจสอบโปรแกรมภายใน 8051

- ขา Port (P3.0-P3.7) (ขา 10-17) เป็นพอร์ตไอโอ 3 บิตแบบพูลอัพภายในนอกจากทำเป็นพอร์ตไอโอ ที่สามารถรับโหลดทีทีแอลพวกตระกูลแอลเอสไอได้ 4 ตัวแล้ว ยังใช้งานเป็นพิเศษสำหรับตระกูล MCS-51 ดังตารางที่

ขาพอร์ต	ขา	การทำงานตามฟังก์ชั่นพิเศษ
P3.0	10	RXD พอร์ตอนุกรมอินพุท
P3.1	11	TXD พอร์ตอนุกรมเอาท์พุต
P3.2	12	INT0 อินเทอร์รัพท์ภายนอกตัวที่ 1
P3.3	13	INT1 อินเทอร์รัพท์ภายนอกตัวที่ 2
P3.4	14	T0 สัญญาณกระตุ้นเข้าที่ตัวตั้งเวลาและตัวนับ 0
P3.5	15	T1 สัญญาณกระตุ้นเข้าที่ตัวตั้งเวลาและตัวนับ 1
P3.6	16	WR สัญญาณควบคุมการเขียน
P3.7	17	RD สัญญาณควบคุมการอ่าน

ตารางที่ 2.2 รายละเอียดการทำงานของขา P3.0-P3.7

การที่จะให้ทำงานตามฟังก์ชันข้างบน จะต้องเริ่มโปรแกรมด้วยการส่งค่า "1" ไปแลทช์ไว้ก่อนที่จะให้ทำงานตามฟังก์ชันข้างบน

- ขา RST (ขา 9) ต้องคงสถานะค่าสูงเป็นเวลาประมาณอย่างน้อย 2 วัฏจักรระหว่างที่ออสซิเลเตอร์ทำงานที่ต้องการเซ็ตทั้งระบบงาน โดยจะต่อรีจิสเตอร์พูลดาวน์ (8.2 กิโลโอห์ม) จากขา RST ไปลงดินและเพื่อให้ตัวชิพรีเซ็ตได้โดยอัตโนมัติ ขณะเปิดจะใช้คาปาซิเตอร์ (10 ไมโครฟารัด) ต่อคร่อมระหว่างขา RST กับขา V_{cc}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ขา ALE/PROG (ขา 30) เป็นขาแอดเดรสแลทช์อีนาเบิ้ลด้วยการส่งพัลส์ออกไป ใช้สำหรับแลทช์ค่าแอดเดรสไบท์ต่ำจากพอร์ต 0 ในระหว่างการเข้าถึงข้อมูลจากหน่วยความจำภายใน ALE จะถูกส่งสัญญาณออกมาในอัตราความเร็วคงที่ ที่ 1/8 ของความถี่ออสซิเลเตอร์ตลอดเวลาแม้ว่าจะไม่มีการเข้าถึงข้อมูลจากภายในดังนั้นจึงสามารถที่จะใช้สัญญาณจากขานี้เป็นตัวตั้งเวลาภายนอก หรือเป็นความถี่สัญญาณนาฬิกา แต่อย่างไรก็ตามความถี่สัญญาณนี้จะลดความถี่ช้าลงไปเท่าหนึ่ง ระหว่างการทำงานแบบการเข้าถึงของหน่วยความจำข้อมูลภายนอก ขานี้ยังใช้เป็นสัญญาณพัลส์เข้าสำหรับการควบคุมการโปรแกรม EPROM ภายในชิพ

- ขา PSEN (ขา29) Program Storage Enable เป็นสโตรบอ่านข้อมูลจากโปรแกรมหน่วยความจำภายนอก เมื่อชิพทำงานด้วยโปรแกรมภายนอก ขา PSEN จะสร้างสโตรบต่ำสองครั้งภายในแต่ละวัฏจักรแมชชีน สัญญาณจะมีสถานะสูง หรือพัลส์ต่ำทั้งสองลูกจะหายไป เมื่อทำงานในช่วงการอ่านหรือเขียนข้อมูลจากหน่วยความจำภายนอก และ PSEN จะไม่มีพัลส์ส่งออกถ้าชิพทำงานด้วยโปรแกรมหน่วยความจำภายใน

- ขา EA/V_{dd} (ขา 31) มีสถานะสูง ตัวซีพียูในชิพจะทำงานตามโปรแกรมที่อยู่ในหน่วยความจำภายใน (โดยที่โปรแกรมจะต้องไม่ยาวกว่า 4 กิโลไบต์สำหรับเบอร์ 8051 AH และ 8 กิโลไบต์สำหรับเบอร์ 8052 AH) การทำให้ EA มีสถานะต่ำ จะเป็นการควบคุมให้ซีพียูทำงานตามโปรแกรมหน่วยความจำภายนอก ซึ่งขยายโปรแกรมได้ยาวถึง 64 กิโลไบต์ ในตัว 8031 AH และ 8032 AH ขา EA จะต้องต่อลงดินเช่นกันแม้ว่าจะไม่มี ROM อยู่ภายในก็ตาม ในตัว 8751 AH จะใช้ขานี้จ่ายแรงดันขนาด 21 โวลต์ ขณะทำการเขียนโปรแกรมเข้า EPROM ของชิพ

- ขา XTAL1 (ขา 19)ใช้เป็นตัวอินพุทเข้าสู่ออสซิเลเตอร์ขยายแบบ Invert

- ขา XTAL2 (ขา 18) ใช้เป็นเอาท์พุทจากตัวออสซิเลเตอร์ขยายแบบ Invert

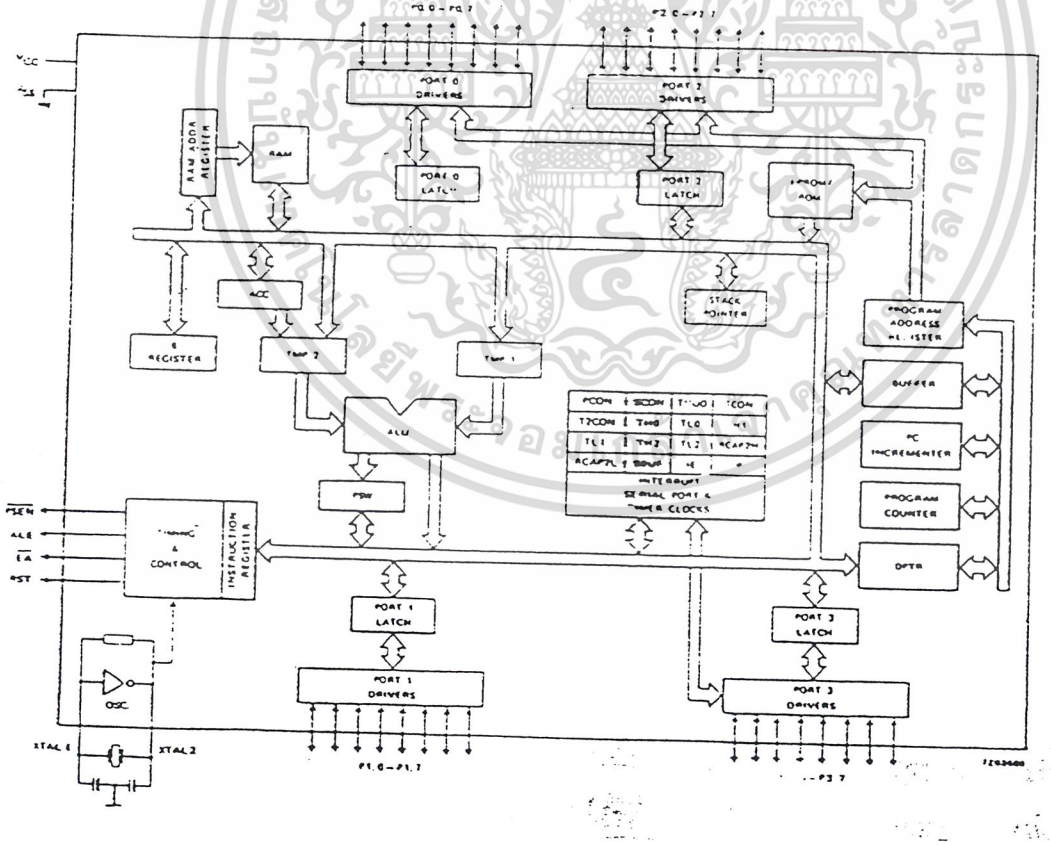
ตามตาราง 2.1 MCS-51 ทั้งสามกลุ่มคือ กลุ่มที่มี ROM และไม่มี ROM และพวก EPROM จะมีที่ขาใช้งานเหมือกันหมดยกเว้นขา 1 จะใช้งานเป็น T2 และขา 2 เป็น T2EX ในเบอร์ 8032/8052 ตลอดถึงจังหวะเวลา (Timing Diagram) และคุณสมบัติทางไฟฟ้าทั้งสามจะแตกต่างกันเฉพาะการโปรแกรมบนชิพ MCS-51 เท่านั้น ซึ่งแต่ละแบบจัดไปตามความต้องการของผู้ใช้ เช่น 8751 AH จะมีหน่วยความจำ EPROM ขนาด 4 กิโลไบต์เหมาะสำหรับการพัฒนาเครื่องต้นแบบและการผลิตอุปกรณ์ที่มีจำนวนจำกัด เมื่อต้องการจะเขียนโปรแกรมเข้า EPROM จะมีตัวเขียนโปรแกรมพิเศษสำหรับเขียนโปรแกรมที่ผู้ออกแบบเขียนขึ้นมาถ้าโปรแกรมมีส่วนผิดพลาดที่ต้องการแก้ไขได้ โดยการนำตัว 8751 AH นี้ไปล้างโปรแกรมเดิมออก ด้วยแสงอัลตราไวโอเลตและอัดโปรแกรมที่ได้แก้ไขแล้วเข้าไปใหม่ ทำเช่นนี้จนกระทั่งได้ โปรแกรมสมบูรณ์ และเมื่อต้องการผลิตจำนวนมากก็สามารถที่จะใช้ MCS-51 เบอร์ 8051 ที่มี 4 กิโลไบต์ของ ROM ซึ่งจะถูกอัดข้อมูลโปรแกรมตามความต้องการของผู้ออกแบบ โดยโรงงานผู้ผลิตชิพเบอร์นี้ การผลิตลักษณะนี้จะถูกกว่าการใช้เบอร์ 8751 แต่โปรแกรมภายในจะไม่สามารถลบ และโปรแกรมใหม่ได้หลังจากการผลิตไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนเบอร์ 8031 จะไม่มีหน่วยความจำของโปรแกรมบนชิพ แต่อาจต่อหน่วยความจำโปรแกรมจากภายนอกด้วย ROM EPROM หรือ PROM ได้ถึง 64 กิโลไบต์ดังนั้น 8031 จึงเหมาะสำหรับการใช้งานที่โปรแกรมมีขนาดใหญ่กว่า 4 กิโลไบต์ และสำหรับผู้ออกแบบที่ต้องการแยกส่วนของโปรแกรมออกจากชิพ

2.1.2 การจัดการทางสถาปัตยกรรม

รูปที่ 2.2 เป็นบล็อกไดอะแกรมที่บ่งลักษณะงานทางด้านสถาปัตยกรรมภายในของ MCS-51 โดยชิงเกิลชิพแต่ละตัวของตระกูลนี้ จะประกอบด้วยหน่วยสูนย์กลางการประมวลหน่วยความจำสองชนิดคือแบบ RAM และ ROM หรือ EPROM พอร์ตอินพุต เอาท์พุต โพรแกรีจิสเตอร์สถานะข้อมูล ส่วนวงจรตรรกในการสุ่ม ที่จำเป็นสำหรับตัวแปรของฟังก์ชันการต่อพ่วง ส่วนต่างๆ ที่กล่าวนี้จะติดต่อกันด้วยบัสขนาด 8 บิต และจะมีบัฟเฟอร์สำหรับการติดต่อกับภายนอกผ่านพอร์ตไอโอ เมื่อต้องการขยายหน่วยความจำหรือพอร์ตไอโอ



รูปที่ 2.2 โครงสร้างสถาปัตยกรรมภายในของ MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แอกคิวมูเลเตอร์ (Accumulator: Acc)

MCS-51ก็เช่นเดียวกับMCS-48 ที่ใช้แอกคิวมูเลเตอร์ที่มีขนาด 8 บิต เป็นแอกคิวมูเลเตอร์หลัก คำสั่งส่วนใหญ่จะอ้างถึงตัวรีจิสเตอร์นี้โดยถือค่าภายในเป็นค่าตัวตั้งและรับค่าผลลัพธ์ที่ได้จากคำสั่งทางคณิตศาสตร์ เช่น บวก ลบ คูณ หาร เข้ามาเก็บไว้ในตัวแอกคิวมูเลเตอร์ยังสามารถใช้ตัวแหล่งกระทำ หรือถูกกระทำในการทำงานทางตรรก และใช้เป็นตัวกลางในการถ่ายเทข้อมูลในการติดต่อกับอุปกรณ์ภายนอกไอโอและหน่วยความจำภายนอกรวมถึงการตรวจสอบข้อมูล

- รีจิสเตอร์ B

เป็นรีจิสเตอร์พิเศษที่ใช้งานสำหรับคำสั่งของการคูณและหาร โดยใช้เป็นที่เก็บตัวคูณหรือหาร และเป็นที่เก็บผลลัพธ์ตัวที่สองหลังการคูณและเศษของการหาร

- รีจิสเตอร์คำแสดงสถานะโปรแกรม (Program Status Word: PSW)

รีจิสเตอร์ PSW เป็นรีจิสเตอร์ที่แสดงผลที่ได้หลังจากการใช้คำสั่งต่างๆ และใช้เป็นตัวเลือกกลุ่มการทำงานของรีจิสเตอร์กลุ่มต่างๆ

- ตัวชี้สแตค (Stack Pointer: SP)

MCS-51 จะใช้หน่วยความจำข้อมูลภายใน เป็นบริเวณสแตคทางฮาร์ดแวร์สำหรับ การเชื่อมโปรแกรมหลักสแตคการผ่านพารามิเตอร์ระหว่างงาน ในแต่ละส่วนโปรแกรม และสแตคเก็บตัวแปรข้อมูลชั่วคราว หรือสแตคการเก็บสถานะระหว่างการบริการงานอินเทอร์รัพต์ไว้ภายในชิพ โดยที่ตัวชี้สแตคจะมีขนาด 8 บิต จะเพิ่มค่าอัตโนมัติก่อนที่ข้อมูลจะนำมาเก็บในหน่วยความจำระหว่างการใช้คำสั่งPUSHและCALL และจะลดค่าของตัวชี้สแตคลงจากที่ได้ถ่ายเทข้อมูลออกไปแล้วในคำสั่ง POP หรือ RETURN โดยทฤษฎีทางสถาปัตยกรรมMCS-51 สามารถใช้สแตคให้มีเนื้อที่ถึง 128 ไบต์ แต่ในทางปฏิบัติสำหรับโปรแกรมทั่วไปจะใช้น้อยกว่านี้ ตัวชี้สแตคจะเริ่มที่ตำแหน่ง 07H ดังนั้นสแตคจะเริ่มบรรจุข้อมูลที่ตำแหน่ง 08H MCS-51สามารถเปลี่ยนแปลงค่าในตัวชี้สแตคได้ ซึ่งจะเป็นการเปลี่ยนตำแหน่งสแตคไปยังที่ใดๆ ของหน่วยความจำข้ออมูลภายในชิพ

- ตัวชี้ข้อมูล (Data Pointer: DPTR)

ตัวชี้ข้อมูลเป็นรีจิสเตอร์ขนาด 16 บิตที่ประกอบด้วยไบต์สูง (DPH) และไบต์ต่ำ (DPL) ที่เราสามารถเลือกแบ่งออกเป็นรีจิสเตอร์ 8 บิตสองที่ใช้ได้อย่างอิสระหรือจะใช้ร่วมกันทั้ง 16 บิตก็ได้ ในการเพิ่มค่าหรือลดค่าเพื่อประโยชน์ในการใช้เป็นฐานของเลขที่อยู่ในรีจิสเตอร์ในการกระโดดโดยทางอ้อมในการใช้คำสั่งเกี่ยวกับตารางข้อมูลและชี้ตำแหน่งของหน่วยความจำภายนอก

- พอร์ต 0 ถึง 3

รีจิสเตอร์ P0, P1, P2 และ P3 ของกลุ่มรีจิสเตอร์ฟังก์ชันพิเศษ (Special Function Register: SFR) จะเป็นตัวรีจิสเตอร์ที่แลทซ์ค่าของพอร์ต 0, 1, 2 และ 3ตามลำดับในขณะใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รีจิสเตอร์ CAPTURE

ไอซีเบอร์ 8035/8052 จะมีคู่รีจิสเตอร์ (RCAP2H,RCAP2L) เพิ่มเติมเป็นรีจิสเตอร์แคปเจอร์สำหรับตัวตั้งเวลาหมายเลข 2 ในโหมดการใช้งานของรีจิสเตอร์ตัวนี้จะรับการเปลี่ยนแปลงที่เข้ามาที่ขา T2EX ตัว TH2 และ TL2 จะลอกข้อมูลเข้าไปในรีจิสเตอร์คู่ RCAP2H และ RCAP2L โดยการใช้ตัวตั้งเวลา จะมีโหมดการบรรจุอัตโนมัติขนาด 16 บิตสำหรับการใช้ตัวตั้งเวลา/ตัวนับ 2

- บัฟเฟอร์ข้อมูลอนุกรม (Serial Data Buffer: SBUF)

บัฟเฟอร์ข้อมูลอนุกรมแบ่งออกเป็นรีจิสเตอร์สองตัว ตัวหนึ่งเป็นบัฟเฟอร์การส่งและอีกหนึ่งเป็นบัฟเฟอร์การรับ เมื่อข้อมูลถ่ายเทเข้า SBUF มันจะถ่ายเข้าบัฟเฟอร์การส่งซึ่งเป็นตัวจัดการส่งข้อมูลอนุกรม วิธีการเคลื่อนย้ายเข้า SBUF ขึ้นอยู่กับการกำหนดติดตั้งโปรแกรม การส่งข้อมูลย้ายออกจาก SBUF จะเป็นการรับข้อมูลจากบัฟเฟอร์ตัวรับ

- รีจิสเตอร์ควบคุม (Control Register)

กลุ่ม SFR ที่เป็น IP, IE, TMOD, TCON, T2CON, SCON และ PCON จะประกอบด้วยบิตที่ใช้ในการควบคุมและแสดงสถานะของการใช้งานในระบบอินเทอร์รัพต์ตัวตั้งเวลา/ตัวนับและพอร์ตอนุกรม

2.1.3 การจัดการหน่วยความจำ

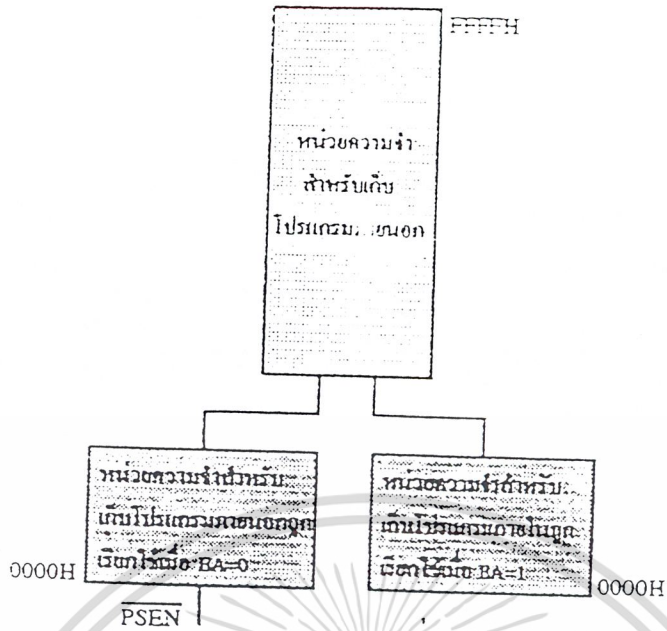
MCS-51แบ่งตามพื้นฐานหน่วยความจำของการกำหนดเลขที่อยู่แอดเดรสได้เป็น 3 ส่วน ประกอบด้วยเนื้อที่

1. 64 กิโลไบต์หน่วยความจำโปรแกรม
2. 64 กิโลไบต์หน่วยความจำข้อมูลภายนอก
3. 256 ไบต์เป็นหน่วยความจำข้อมูลภายใน ส่วนเบอร์ 8032/8052 มีขนาด 384 ไบต์

2.1.4 เนื้อที่หน่วยความจำโปรแกรม

หน่วยความจำโปรแกรมจะประกอบด้วย ส่วนภายในและภายนอกชิพ ถ้าขา EA มีสถานะสูง MCS-51 จะบริการโปรแกรมภายใน ถ้าโปรแกรมมีความยาวไม่เกิน 0FFFH (4K) หรือ 1FFFH (8K) สำหรับตัว 8052 ตำแหน่งตั้งแต่ 1000H ถึง 0FFFH(หรือ 2000H – 0FFFH สำหรับ 8052) จะเป็นการเฟทซ์ข้อมูลจากภายนอก ถ้าขา EA มีสถานะต่ำ MCS-51 จะเฟทซ์ข้อมูลภายนอกทั้งหมด ในทุกกรณีตัวนับโปรแกรมขนาด 16 บิต จะเป็นตัวกำหนดเลขที่อยู่โปรแกรมซึ่งจะแสดงโครงสร้างของหน่วยความจำได้ดังรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แสดงแผนที่ของหน่วยความจำโปรแกรม

ตำแหน่ง 00H ถึง 32H (หรือ 00H ถึง 2BH สำหรับเบอร์ 8032/8052) ในหน่วยความจำโปรแกรมจะสำรองสำหรับใช้บริการอินเทอร์รัพต์ตามตารางที่ 2.8

ชนิดของการอินเทอร์รัพต์ใน	ตำแหน่งเริ่มต้นของโปรแกรมบริการการอินเทอร์รัพต์ (Vector Interrupt)
อินเทอร์รัพต์ภายนอกชนิด 0 (IE0)	0003H
อินเทอร์รัพต์ของไทม์เมอร์ 0 (TF0)	000BH
อินเทอร์รัพต์ภายนอกชนิด 1 (IE1)	0013H
อินเทอร์รัพต์ของไทม์เมอร์ 1 (TF1)	001BH
อินเทอร์รัพต์ของพอร์ตสื่อสารอนุกรม (TI+RI)	0023H
อินเทอร์รัพต์ของไทม์เมอร์ 2 (TF2+EXF2)	002BH

ตารางที่ 2.3 แสดงตำแหน่งเริ่มต้นของโปรแกรมบริการอินเทอร์รัพต์

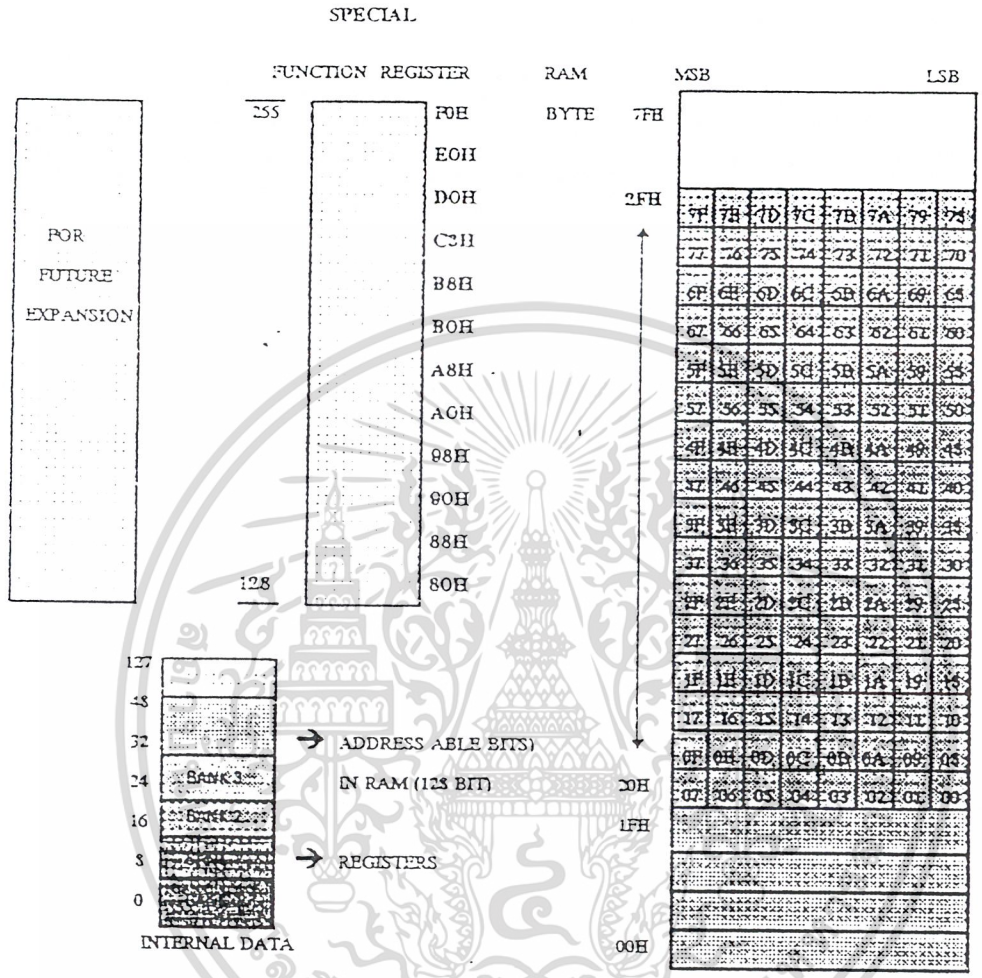
2.1.5 เนื้อที่หน่วยความจำข้อมูล

หน่วยความจำข้อมูลจะประกอบด้วยความจำข้อมูลภายในและภายนอก หน่วยความจำข้อมูลภายนอกจะเข้าถึงได้ด้วยการใช้คำสั่ง MOVX

หน่วยความจำข้อมูลภายในจะแบ่งเป็นลักษณะงานดังนี้คือ

1. จำนวน 128 ไบต์ของบริเวณตำแหน่งล่างในเนื้อที่ RAM ภายใน
2. อีก 128 ไบต์เป็นของบริเวณตำแหน่งบนของหน่วยความจำข้อมูลภายใน

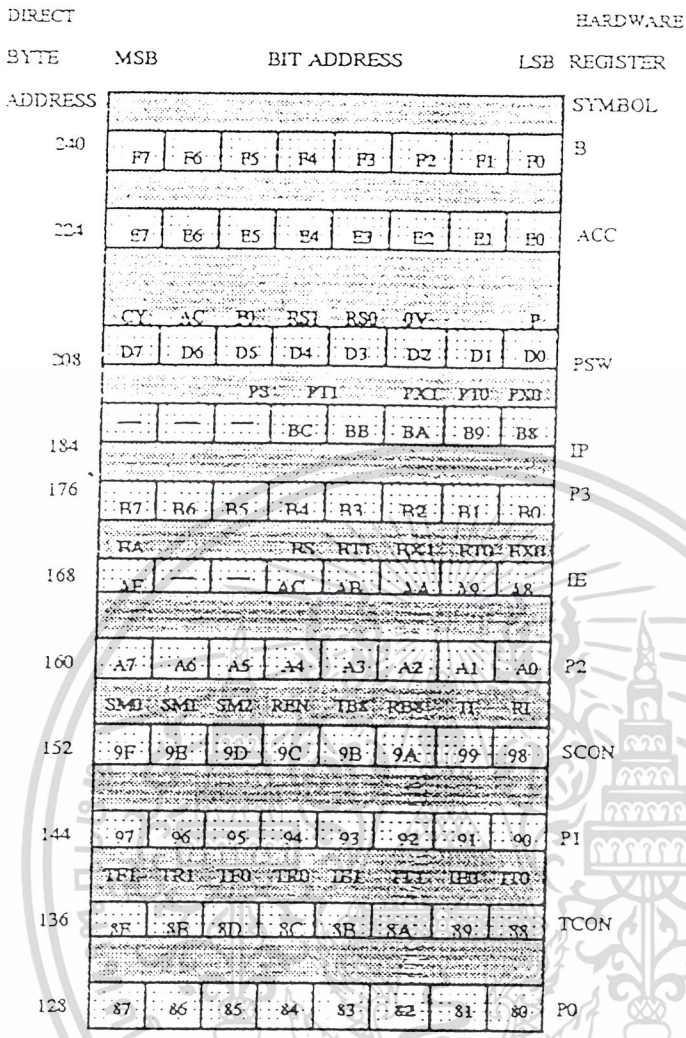
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 ก.) แสดงแผนที่ของหน่วยความจำข้อมูล ข.)แสดงแผนที่การกำหนดตำแหน่งบิต

ส่วนบนนี้จะมีเฉพาะในเบอร์ 8032/8052 เท่านั้น และส่วนของ 128 ไบต์อีกบริเวณหนึ่งใช้เป็นรีจิสเตอร์ฟังก์ชันพิเศษ ขณะที่ใช้ส่วนบนของหน่วยความจำข้อมูลภายในและบริเวณของ SFR ทั้ง 2 ส่วนนี้จะถูกปันส่วนให้ใช้ค่าแอดเดรสภายในแต่จะเข้าถึงในแต่ละบริเวณของทั้ง 2 บริเวณนี้ได้โดยการใช้โหมดการกำหนดเลขที่อยู่ต่างกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



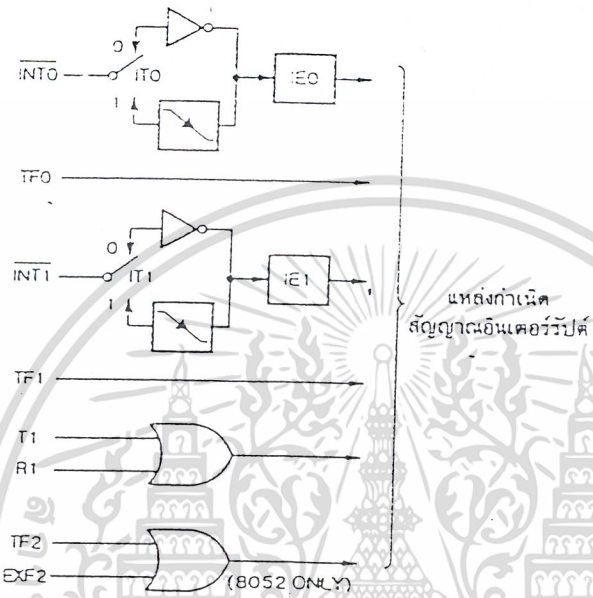
รูปที่ 2.5 แสดงตำแหน่งของรีจิสเตอร์ SFR และบิตแอดเดรสของ SFR

รูปที่ 2.4ก.) แสดงถึงแผนที่ของหน่วยความจำข้อมูลโดยแบ่งเป็น 4 แบงก์ ในแต่ละแบงก์มีรีจิสเตอร์ 8 ตัว มีตำแหน่งตั้งแต่ 0H-31H ในบริเวณของหน่วยความจำข้อมูลแบงก์เหล่านี้จะถูกเลือกให้ใช้อีนาเบิ้ลได้คราวละ 1 แบงก์ด้วยการกำหนดเริ่มแรกภายใน 2 บิตของรีจิสเตอร์ PSW ว่าจะเลือกใช้ในแบงก์ใดภายใน 4 แบงก์ และบริเวณตำแหน่งตั้งแต่ 20H-2FH จำนวน 16 ตำแหน่ง ตำแหน่งละ 1 ไบต์ สามารถที่จะกำหนดเลขที่อยู่ของแต่ละบิตได้ดังแสดงในรูปที่ 2.4ข.) เป็นบิตแรกแอดเดรสเนื้อที่รีจิสเตอร์ SFRสามารถที่จะกำหนดตำแหน่งได้เช่นกัน ดังรูปที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 โครงสร้างทางอินเตอร์รัพต์ MCS-51

MCS-51 สามารถรับสัญญาณอินเตอร์รัพต์ที่เกิดขึ้นอย่างน้อย 5 ชนิด (MCS-51 บางเบอร์ในตระกูลนี้สามารถรับสัญญาณอินเตอร์รัพต์ได้มากกว่า 5 ชนิด เช่น เบอร์ 8052 สามารถรับได้ 6 ชนิด) แหล่งกำเนิดสัญญาณอินเตอร์รัพต์ทั้ง 5 ชนิดที่ MCS-51 สามารถรับได้มีดังแสดงในรูปที่ 2.7



รูปที่ 2.6 แหล่งกำเนิดสัญญาณอินเตอร์รัพต์ทั้ง 5 ชนิดที่ สามารถรับได้

อินเตอร์รัพต์แต่ละชนิดที่ MCS-51 สามารถรับได้มีรายละเอียดดังนี้

1. อินเตอร์รัพต์ที่เกิดจากภายนอก (External Interrupt) เป็นอินเตอร์รัพต์ที่เกิดขึ้นจากภายนอก MCS-51 มีอยู่ 2 ชนิดด้วยกันคือ

- อินเตอร์รัพต์ภายนอกชนิด 0 รับได้จากขา INT0
- อินเตอร์รัพต์ภายนอกชนิด 1 รับได้จากขา INT1

ผู้ใช้สามารถกำหนดให้ MCS-51 ตรวจสอบสัญญาณอินเตอร์รัพต์ทั้งสองชนิดที่เกิดขึ้นที่ขา INT0, INT1 2 แบบด้วยกันคือ

- ตรวจสอบจากระดับสัญญาณ (level-activated)
- ตรวจสอบจากการเปลี่ยนสถานะสัญญาณ (transition-activated)

การตรวจสอบสถานะของสัญญาณอินเตอร์รัพต์ภายนอกที่ขาทั้งสอง สามารถเลือกได้เพียงอย่างใดอย่างหนึ่งขึ้นอยู่กับการกำหนดค่าบิต IT0, IT1 (Interrupt Type Control bit) ในรีจิสเตอร์ใช้งานเฉพาะ TCON ดังแสดงในรูปที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อ MCS-51 ตรวจพบสัญญาณอินเตอร์รัพต์จากภายนอก จะมีผลทำให้บิต IE0(จากขา INT0) หรือ IE1(จากขาINT1) ของรีจิสเตอร์ใช้งานเฉพาะ TCON ถูกเซ็ต บิตทั้งสองจะเป็นตัวบอกสถานะของสัญญาณอินเตอร์รัพต์ที่เกิดจากภายนอก โดยถูกเซ็ตเมื่อเกิดสัญญาณอินเตอร์รัพต์และจะถูกเคลียร์โดยฮาร์ดแวร์ภายใน MCS-51 เอง เมื่อซีพียูย้ายไปทำงานที่โปรแกรมบริการอินเตอร์รัพต์ต่อเมื่อ สัญญาณอินเตอร์รัพต์ภายนอกที่เกิดขึ้นเป็นชนิดที่ตรวจสอบได้จากการเปลี่ยนสถานะสัญญาณ แต่ถ้าสัญญาณอินเตอร์รัพต์ภายนอกที่เกิดขึ้นได้มาจากการตรวจสอบระดับสัญญาณเมื่อซีพียูย้ายไปทำงานที่โปรแกรมบริการอินเตอร์รัพต์จะไม่มีการเคลียร์บิต IE0 หรือ IE1 ให้ ในกรณีนี้จะเป็นหน้าที่ของวงจรกำเนิดสัญญาณอินเตอร์รัพต์ภายนอกที่ต้องทำหน้าที่ควบคุมสถานะของสัญญาณที่ขา INTX (INT0 หรือ INT1)ให้กลับสู่สภาพเดิมเอง มิฉะนั้นโปรแกรมหลักที่ทำงานอยู่จะถูกอินเตอร์รัพต์ไปเรื่อยๆ จนกระทั่งสัญญาณอินเตอร์รัพต์กลับมีค่าเป็น 1 อีกครั้ง

- อินเตอร์รัพต์ของไทม์เมอร์ 0 และไทม์เมอร์ 1 อินเตอร์รัพต์ของไทม์เมอร์ 0 หรือไทม์เมอร์ 1ถูกทำให้เกิดขึ้นโดยบิต TF0 หรือ TF1 ซึ่งถูกเซ็ตเมื่อไทม์เมอร์ 0 หรือไทม์เมอร์ 1 เกิดโอเวอร์โฟลว์ (overflow) (มีการเปลี่ยนค่าจาก 1 ทั้งหมดมาเป็น 0 ทั้งหมดในรีจิสเตอร์ที่ใช้เป็นไทม์เมอร์หรือเคาน์เตอร์ของไทม์เมอร์ 0 หรือไทม์เมอร์ 1)ยกเว้นไทม์เมอร์ 0 ในโหมด 3 ซึ่งหยุดการทำงานเมื่อมีอินเตอร์รัพต์จากไทม์เมอร์เกิดขึ้น บิต TF0 และ TF1 จะถูกเคลียร์โดยฮาร์ดแวร์ภายใน MCS-51 เองเมื่อซีพียูย้ายไปทำงานที่โปรแกรมบริการอินเตอร์รัพต์

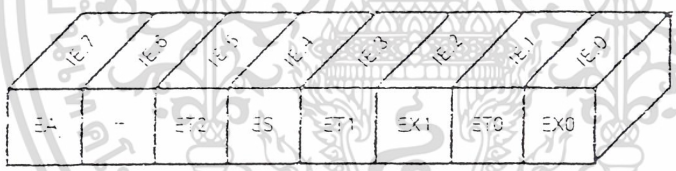
- อินเตอร์รัพต์ของพอร์ตสื่อสารอนุกรม (Serial Port Interrupt) พอร์ตสื่อสารอนุกรมของ MCS-51 สามารถทำให้เกิดสัญญาณอินเตอร์รัพต์ได้ สัญญาณอินเตอร์รัพต์ที่เกิดขึ้นได้มาจากบิต RI หรือ TI ที่นำมาผ่านเกตออร์(ดังแสดงในรูปที่ 2.6) และบิตที่ควบคุมการอินเตอร์รัพทั้งสองนี้จะไม่ถูกเคลียร์โดยฮาร์ดแวร์ใน MCS-51 เมื่อซีพียูไปทำงานในโปรแกรมบริการการอินเตอร์รัพต์ เพราะการเกิดอินเตอร์รัพต์ของพอร์ตสื่อสารอนุกรมอาจจะเกิดจากบิต RI หรือ TI ก็ได้ ดังนั้นโปรแกรมในส่วนบริการการอินเตอร์รัพต์จะต้องตรวจสอบเองว่าสัญญาณอินเตอร์รัพต์ที่เกิดขึ้นได้มาจากบิต RI หรือ TI และบิตทั้งสองจะถูกเคลียร์โดยซอร์ฟแวร์เท่านั้น

อินเตอร์รัพต์แต่ละชนิดที่กล่าวไปแล้ว สามารถควบคุมให้สามารถอินเตอร์รัพต์ MCS-51 ได้หรือไม่ โดยการควบคุมจากบิตต่างๆ ในรีจิสเตอร์ใช้งานเฉพาะ IE ดังแสดงในรูปที่ 2.7

- รีจิสเตอร์ใช้งานเฉพาะ IE (Interrupt Enable-Register) เข้าถึงข้อมูลได้ในระดับบิต การกำหนดให้บิตควบคุมการตอบสนองต่อสัญญาณอินเตอร์รัพต์แต่ละชนิดมีค่าเป็น 0 หมายถึงไม่ให้ MCS-51 ตอบสนองต่อสัญญาณอินเตอร์รัพต์ชนิดนั้น หากกำหนดให้บิตควบคุมการตอบสนองต่อสัญญาณอินเตอร์รัพต์แต่ละชนิดมีค่าเป็น 1 หมายถึงให้ MCS-51 ตอบสนองต่อสัญญาณอินเตอร์รัพต์ชนิดนั้น (บิต EA ต้องถูกเซ็ตไว้ก่อนด้วย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IE.7	EA	ใช้ควบคุมการตอบสนองต่อสัญญาณอินเตอร์รัพต์ทั้งหมด 0: MCS-51 จะไม่ตอบสนองต่อสัญญาณอินเตอร์รัพต์ใดๆ ทั้งสิ้น 1:อินเตอร์รัพต์แต่ละชนิดจะถูกควบคุมการตอบสนองอย่างอิสระจากบิตในรีจิสเตอร์นี้
IE.6	-	ไม่ถูกกำหนดการใช้งาน (สำรองไว้ใช้ใน MCS-51 เบอร์ใหม่ๆ ในอนาคต)
IE.5	ET2	ควบคุมการตอบสนองต่อสัญญาณอินเตอร์รัพต์ของไทม์เมอร์2 เมื่อเกิด overflow (มีใช้เฉพาะ MCS-51 บางเบอร์ที่มีไทม์เมอร์ 2 เช่น 8052)
IE.4	ES	ควบคุมการตอบสนองต่อสัญญาณอินเตอร์รัพต์ของพอร์ตสื่อสารอนุกรม
IE.3	ET1	ควบคุมการตอบสนองต่อสัญญาณอินเตอร์รัพต์ของไทม์เมอร์1 เมื่อเกิด overflow
IE.2	EX1	ควบคุมการตอบสนองต่อสัญญาณอินเตอร์รัพต์ภายนอกชนิด 1
IE.1	ET0	ควบคุมการตอบสนองต่อสัญญาณอินเตอร์รัพต์ของไทม์เมอร์0 เมื่อเกิด overflow
IE.0	EX0	ควบคุมการตอบสนองต่อสัญญาณอินเตอร์รัพต์ภายนอกชนิด 0



รูปที่ 2.7 รีจิสเตอร์ใช้งานเฉพาะ IE

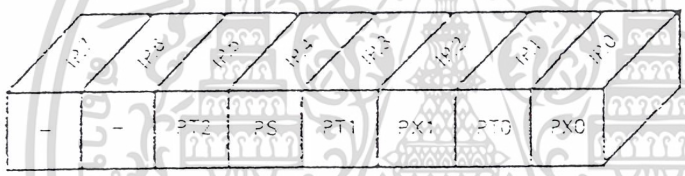
บิต EA ในรีจิสเตอร์ใช้งานเฉพาะ IE สามารถควบคุมการอินเตอร์รัพต์ใน MCS-51 ได้ทั้งหมด หากบิตนี้มีค่าเป็น 0 สัญญาณอินเตอร์รัพต์ทุกชนิดที่เกิดขึ้น จะไม่สามารถอินเตอร์รัพต์ MCS-51 ได้ แต่หากบิตนี้มีค่าเป็น 1 สัญญาณอินเตอร์รัพต์แต่ละชนิดจะถูกควบคุมให้อินเตอร์รัพต์ MCS-51 ได้อย่างอิสระ (ควบคุมจากบิต IE.0-IE.5)

บิต IE.5, IE.6 ไม่ถูกใช้ใน 8051 เพราะถูกสงวนไว้ใช้ใน MCS-51 เบอร์อื่นๆ ที่สามารถรับอินเตอร์รัพต์ได้เพิ่มขึ้น ดังนั้นซอร์ฟแวร์ของผู้ใช้ไม่ควรจะมีคำสั่งเขียนค่า 1 ลงไปในบิตเหล่านี้ เพื่อให้โปรแกรมยังคงสามารถใช้กับชิพเบอร์ใหม่ๆ ในตระกูลนี้ได้

2.2.7 โครงสร้างระดับความสำคัญในการบริการอินเตอร์รัพต์ (Priority Level Structure) อินเตอร์รัพต์แต่ละชนิดสามารถถูกเลือกระดับความสำคัญในการบริการได้ 2 ระดับ โดยการเซ็ต หรือเคลียร์บิตในรีจิสเตอร์ใช้งานเฉพาะ IP ดังแสดงในรูปที่ 2.8

ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ห้ามมิให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือใช้ประโยชน์ด้านการค้า ไม่ว่าทั้งหมดหรือบางส่วน โดยไม่ได้รับอนุญาตเป็นลายลักษณ์อักษร มหาวิทยาลัยขอสงวนสิทธิ์ในการดำเนินคดีกับผู้ที่ละเมิดลิขสิทธิ์ของมหาวิทยาลัยโดยไม่ต้องแจ้งให้ทราบล่วงหน้า ผู้ที่สนใจสามารถขออนุญาตได้ที่ สำนักวิทยบริการฯ โทร. 0-2326-4318 ต่อ 99-3111 หรือ 99-3112 ผู้ที่ละเมิดลิขสิทธิ์ต้องรับโทษตามที่กฎหมายกำหนด และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IP.7	-	ไม่ถูกกำหนดการใช้งาน (สำรองไว้ใช้ใน MCS-51 เบอร์ใหม่ๆ ในอนาคต)
IP.6	-	ไม่ถูกกำหนดการใช้งาน (สำรองไว้ใช้ใน MCS-51 เบอร์ใหม่ๆ ในอนาคต)
IP.5	PT2	กำหนดลำดับความสำคัญในการตอบสนองต่อสัญญาณอินเตอร์รัพต์ไทม์เมอร์ 2
IP.4	PS	กำหนดลำดับความสำคัญในการตอบสนองต่อสัญญาณอินเตอร์รัพต์พอร์ตสื่อสารอนุกรม
IP.3	PT1	กำหนดลำดับความสำคัญในการตอบสนองต่อสัญญาณอินเตอร์รัพต์ไทม์เมอร์ 1
IP.2	PX1	กำหนดลำดับความสำคัญในการตอบสนองต่อสัญญาณอินเตอร์รัพต์ภายนอกชนิด 1
IP.1	PT0	กำหนดลำดับความสำคัญในการตอบสนองต่อสัญญาณอินเตอร์รัพต์ไทม์เมอร์ 0
IP.0	PX0	กำหนดลำดับความสำคัญในการตอบสนองต่อสัญญาณอินเตอร์รัพต์ภายนอกชนิด 0



รูปที่ 2.8 รีจิสเตอร์ใช้งานเฉพาะ IP

- รีจิสเตอร์ใช้งานเฉพาะ IP (Interrupt Priority Register) เข้าถึงข้อมูลได้ในระดับบิต การให้บิตกำหนดความสำคัญของอินเตอร์รัพต์เป็น 0 หมายถึง ให้อินเตอร์รัพต์ชนิดนั้นมีลำดับความสำคัญต่ำ ส่วนการให้บิตกำหนดลำดับความสำคัญของอินเตอร์รัพต์เป็น 1 หมายถึงให้อินเตอร์รัพต์ชนิดนั้นมีลำดับความสำคัญสูง

ระดับความสำคัญในการบริการอินเตอร์รัพต์ที่เกิดขึ้นมีได้ 2 ระดับ ได้แก่

- อินเตอร์รัพต์ความสำคัญต่ำ (Low Priority Interrupt) อินเตอร์รัพต์ชนิดนี้สามาถถูกอินเตอร์รัพต์จากสัญญาณอินเตอร์รัพต์ระดับความสำคัญสูงได้ แต่จะไม่สามารถถูกอินเตอร์รัพต์โดยสัญญาณอินเตอร์รัพต์ระดับความสำคัญต่ำตัวอื่นๆ ได้

- อินเตอร์รัพต์ความสำคัญสูง (High Priority Interrupt) อินเตอร์รัพต์ประเภทนี้ไม่สามารถถูกอินเตอร์รัพต์โดยสัญญาณอินเตอร์รัพต์ชนิดอื่นได้เลย นั่นคือมีระดับความสำคัญสูงสุด

การจัดการกับสัญญาณอินเตอร์รัพต์ที่มีระดับความสำคัญเท่าเทียมกัน ที่เกิดขึ้นพร้อมกัน (Priority within level structure) ถูกใช้เพียงเพื่อแก้ปัญหาสัญญาณอินเตอร์รัพต์ที่มีระดับความสำคัญเท่ากันที่เกิดขึ้นพร้อมกันเท่านั้น (simultaneous request of the same priority level)

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย ห้ามมิให้นำไปใช้ประโยชน์ด้านการค้า ห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

แหล่งกำเนิดอินเตอร์รัพต์	ลำดับความสำคัญภายใน
อินเตอร์รัพต์ภายนอกชนิด 0 (IE0)	สูงสุด
อินเตอร์รัพต์ไทม์เมอร์ 0 (TF0)	
อินเตอร์รัพต์ภายนอกชนิด 1 (IE1)	
อินเตอร์รัพต์ไทม์เมอร์ 1 (TF1)	
อินเตอร์รัพต์ของพอร์ทสื่อสารอนุกรม (TI+RI)	
อินเตอร์รัพต์ไทม์เมอร์ 2 (TF2+EXF2)	ต่ำสุด

ตารางที่ 2.4 ระดับความสำคัญในการบริการอินเตอร์รัพต์

รีจิสเตอร์ใช้งานเฉพาะ มีบิตที่ไม่ถูกใช้งานอยู่บางบิต คือ IP.7 และ IP.6 โดยไม่ถูกใช้ใน 8051 และ 8052 ใน 8051 จะมีบิตที่ว่างเพิ่มมาอีก 1 บิต คือ IP.5 ดังนั้นซอร์ฟแวร์ของผู้ใช้ไม่ควรมีการเขียนค่า 1 ไปที่ตำแหน่งบิตเหล่านี้ เพราะมันอาจถูกนำไปใช้ใน MCS-51 เบอร์ใหม่ๆ ในอนาคตต่อไป

2.2.8 ตัวจับเวลา/ตัวนับ (Timer/Counter)

MCS-51 มี 16 บิต ตัวจับเวลา/ตัวนับ 2 ตัวคือ ตัวจับเวลา/ตัวนับ 0 และตัวจับเวลา/ตัวนับ 1 ส่วน 8032/8052 มีเพิ่มอีกหนึ่งชุด คือ ตัวจับเวลา/ตัวนับ 2 ขณะที่แต่ละตัวจับเวลา/ตัวนับ สามารถที่จะติดตั้งให้ทำงานเป็นตัวจับเวลาหรือตัวนับก็ได้

2.2.8.1 ตัวจับเวลา/ตัวนับ 0 และตัวจับเวลา/ตัวนับ 1

แต่ละตัวจะถูกติดตั้งให้ทำงานเป็นตัวจับเวลาหรือตัวนับได้ด้วยการเซ็ตหรือเคลียร์บิตที่ตัวควบคุมในรีจิสเตอร์ TMOD ในกลุ่ม SFR ในฟังก์ชันตัวจับเวลา ตัวรีจิสเตอร์จะเพิ่มค่าทุกๆ วัฏจักรแมชชีน ดังนั้นตัวเลขในรีจิสเตอร์จะเป็นจำนวนของวัฏจักรแมชชีน เนื่องจากแต่ละวัฏจักรแมชชีนประกอบด้วย 12 คาบของออสซิเลเตอร์ อัตราการนับแต่ละครั้งจะกินเวลาเป็น 1/12 ของความถี่ออสซิเลเตอร์

ในฟังก์ชันตัวนับรีจิสเตอร์ จะเพิ่มค่าทุกครั้งที่ที่การเปลี่ยนแปลงสถานะจาก “1” เป็น “0” ที่เข้ามาที่ขา T0 หรือ T1 ในฟังก์ชันนี้สัญญาณภายนอกที่เข้ามาจะถูกรับแซมปลิ้ง (Sampling) ระหว่างช่วง S5P2 ของทุกวัฏจักรแมชชีน โดยถ้าแซมปลิ้งสัญญาณเข้าเป็นระดับสูงในวัฏจักรหนึ่ง ดังนั้นถ้าในวัฏจักรตัวต่อมาของสัญญาณเข้าเป็นระดับต่ำ รีจิสเตอร์จะนับเพิ่มหนึ่งค่า

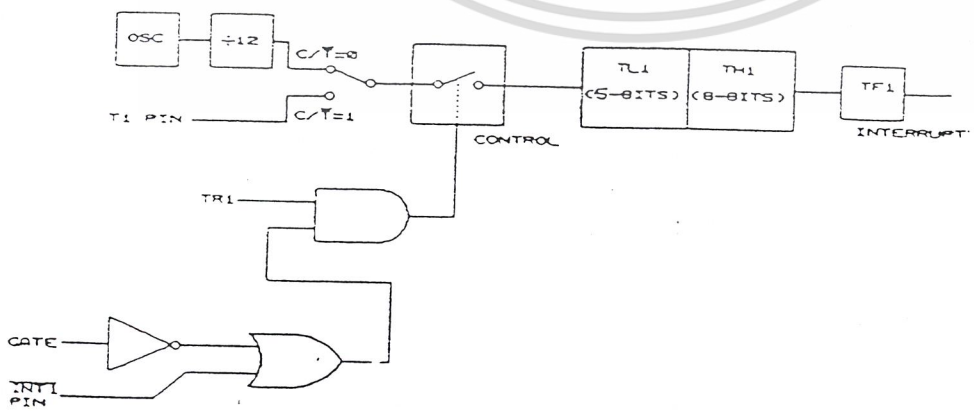
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

34106

ถ้ามีสัญญาณอินเตอร์รัพต์เกิดขึ้นพร้อมกัน 2 ชนิด โดยมีระดับความสำคัญในการบริการอินเตอร์รัพต์ไม่เท่ากัน สัญญาณอินเตอร์รัพต์ที่มีความสำคัญสูงกว่าจะได้รับการบริการก่อน แต่ถ้ามีการขออินเตอร์รัพต์พร้อมกัน 2 ชนิด ซึ่งมีระดับความสำคัญเท่าเทียมกัน ลำดับการบริการอินเตอร์รัพต์ภายในจะเป็นตัวกำหนดเองว่าอินเตอร์รัพต์ชนิดใดควรถูกบริการก่อน ดังนั้นภายในระดับความสำคัญของการบริการอินเตอร์รัพต์หนึ่งๆ จะมีระดับความสำคัญในการบริการอินเตอร์รัพต์ย่อยลงไปอีกระดับ (second priority structure) ดังแสดงในตารางที่ 2.9 โดยที่ค่าไพร่ของตัวนับ จะปรากฏที่รีจิสเตอร์ช่วง S3P1 ของวัฏจักร ซึ่งค่าหนึ่งที่รับเข้าไป จะใช้ช่วง 2 วัฏจักรแมชชีน (เท่ากับ 24 คาบ) ในการวัดค่าช่วงการเปลี่ยน 1 เป็น 0 ดังนั้นค่าสูงสุดในการนับจะมีอัตรา 1/24 เท่าของความถี่ออสซิลเลเตอร์และสัญญาณอินพุตที่จะนับนั้นจะไม่มีช่วงระยะห่างที่แน่นอนของดูตี้ไซเคิล (Duty cycle) และจะถูกนับเมื่อระดับแรงดันที่ถูกแซมปลิ้งในแต่ละครั้ง จะต้องมีช่วงคงที่อย่างน้อย 1 วัฏจักรแมชชีนก่อนที่จะเปลี่ยนค่าระดับแรงดันใหม่ ในการเลือกทำงานระหว่างตัวนับกับตัวจับเวลา จะเลือกได้ 4 โหมด คือ โหมด 0, 1, และ 2 เลือกได้ทั้งสองตัวของตัวจับเวลา ตัวจับเวลา/ตัวนับ ส่วนโหมด 3 ทำงานแตกต่างออกไป

- โหมด 0 การใช้ตัวจับเวลา/ตัวนับ 0 หรือ 1 อยู่ในโหมด 0 จะคล้ายกับการทำงานของ MCS-48 โดยตัวจับเวลาของ MCS-48 มีขนาด 8 บิต มีตัวพรีสเกเลอร์ (Proscaler) เป็นตัวหาร 12 รูปที่ 2.10 แสดงการทำงานในโหมด 0 ของตัวจับเวลา/ตัวนับ 1

ในโหมดนี้รีจิสเตอร์ตัวจับเวลาถูกกำหนดให้มี 13 บิต ด้วยการนับขึ้นเมื่อเป็น "1" หมดทุกบิต จะกลับมาที่ "0" ทุกบิตใหม่ เมื่อกลับเป็น "0"ทุกบิตจะเกิดการโอเวอร์โฟลว์ (Overflow) ไปทดให้แฟลกอินเตอร์รัพต์ TF1 ปรับเป็น "1" การควบคุมให้เริ่มนับตัวอินพุตจะควบคุมด้วยการอีนาเบิล TR1=1 ,GATE=0และขา INT=1 การปรับ GATE =1 เป็นการคิดตั้งตัวนับให้นับด้วยสัญญาณจากภายนอกที่เข้ามาที่ขา INT1 TR1 จะเป็นบิตควบคุมในรีจิสเตอร์ TMOD ของสัญญาณ SFR



รูปที่ 2.9 แสดงการทำงานในโหมด 0 ของตัวจับเวลา/ตัวนับ 1 ขนาด 13 บิต

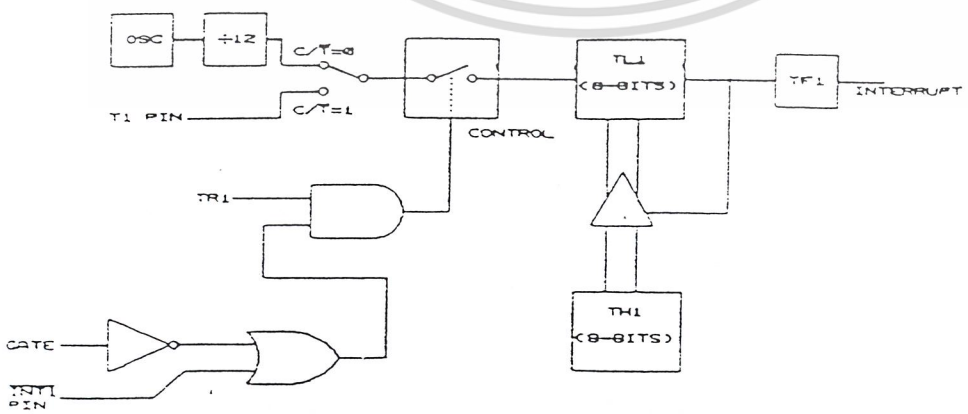
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รีจิสเตอร์ตัวนับจะมี 13 บิต ประกอบด้วย TH1 8 บิต และ TL1 อีก 5 บิตอันดับต่ำ ส่วนอีก 3 บิตที่เหลือในอันดับสูงของ TL1 จะไม่ใช้ การเซ็ตแฟลก TR1 ให้ทำงานจะไม่ได้เคลียร์ค่าในรีจิสเตอร์ของ TH1 และ TL1 การทำงานในโหมด 0 ในตัวจับเวลา/ตัวนับ 0 จะทำงานเหมือนกับตัวจับเวลาตัวนับ 1 โดยใช้ TR0 และ INT0 รวมกันควบคุมแทนสัญญาณต่างๆ ในรูปที่ 2.10 มีความแตกต่างในการควบคุมคือบิตของ GATE ทั้งสองตัวหนึ่งจะแทนตัวจับเวลา/ตัวนับ 1 (TMOD.7) และอีกตัวจะแทนตัวจับเวลา/ตัวนับ 0 (TMOD.3)

-โหมด1 โหมด 1 ทำงานเหมือนกับโหมด 0 ต่างกันเฉพาะการใช้รีจิสเตอร์ตัวจับเวลา/ตัวนับจะทำงานด้วยขนาด 16 บิต โดยไม่มีพรีสเคเลอร์ คือความถี่ 1/12 ของออสซิลเลเตอร์ เป็นความถี่ที่เข้ามาถูกหารด้วยค่า 16 บิตในรีจิสเตอร์ตัวนับ

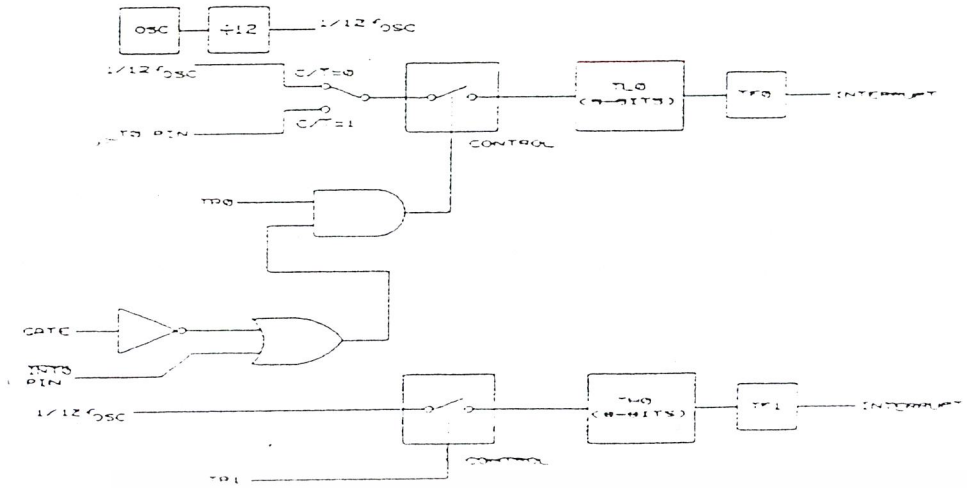
-โหมด2 โหมด 2 มีการทำงานโดยการกำหนดให้ตัวนับ 8 บิตของ TL1 และจะโหลดใหม่โดยอัตโนมัติทุกครั้ง เมื่อมีการโอเวอร์โฟลว์จาก TL1 ดังรูปที่ 2.11 ไม่เพียงแต่ TF1 จะปรับเป็น "1" แต่ TL1 จะถูกโหลดโดยอัตโนมัติจากค่าที่ตั้งไว้ใน TH1 ซึ่งค่าใน TH1 สามารถตั้งค่าได้ด้วยซอร์ฟแวร์ คือการใช้คำสั่ง MOV และบรรจุเข้าไปใหม่ที่ TL1 ทุกครั้งที่เกิดโอเวอร์โฟลว์ TH0 และ TF0 จะเป็นตัวร่วมทำงานในโหมดนี้

-โหมด 3 ถ้าใช้ตัวจับเวลา/ตัวนับ 1 ในโหมด 3 มีการทำงานเป็นตัวนับ มีผลเช่นเดียวกับการตั้ง TR1=0 และใช้ตัวจับเวลาตัวนับ 0 ในโหมด 3 จัดการให้ TL0 ใช้ตัวจับเวลา/ตัวนับ 0 ร่วมกับบิตควบคุมของ C/T,GATE,TR0,INT0 และ TF0 ตัว TH0 จะถูกล็อกให้ทำงานในฟังก์ชันตัวจับเวลาตามรูปที่ 2.12 และใช้บิตแฟลก TR1 และ TF1 เข้าร่วมทำงานในโหมด 3 ดังนั้นตัว TH0 ในโหมดนี้จะควบคุมการอินเตอร์รัพต์ของตัวจับเวลา 1 เป็นกลุ่มจับเวลาและขนาด 8 บิตสองตัว ตัวจับเวลา (เป็นวัฏจักรแมชชีนได้) และจะใช้บิตที่ TR1 และ TF1 ของตัวจับเวลา 1 เป็นตัวควบคุมดังนั้นจึงใช้ TH0 เป็นตัวจับเวลาเป็นการควบคุมการอินเตอร์รัพต์ด้วยตัวจับเวลา 1



รูปที่ 2.10 ตัวจับเวลา/ตัวนับ 1 ทำงานในโหมด 2 แบบโหลดใหม่ 8 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 ใช้ตัวจับเวลา/ตัวนับ 1 ในโหมด 3 เป็นกลุ่มตัวนับขนาด 8 บิต

โหมด 3 สามารถที่จะใช้ในงานที่ต้องการตัวจับเวลา/ตัวนับขนาด 8 บิตที่เพิ่มขึ้นด้วยการใช้ตัวจับเวลา 0 ในโหมด 3 ดังนั้น 8051 สามารถที่จะทำงานใช้ตัวจับเวลา/ตัวนับ ได้เป็น 3 ชุดขณะที่ 8052 ก็จะใช้งานได้ 4 ชุด เมื่อใช้ตัวจับเวลา 0 อยู่ในโหมด 3 ตัวจับเวลา 1 สามารถที่จะเปิด-ปิดให้เข้าสู่หรือออกจากการทำงานของโหมด 3 หรือสามารถที่จะยังคงใช้เป็นตัวสร้างอัตราบอดของการส่งข้อมูลอนุกรมหรือการใช้งานใดๆ ที่ไม่ต้องการการอินเตอร์รัพต์

GATE		ควบคุมเกต เมื่อเซตเป็น "1" จะเป็นอีนาเบิ้ล ตัวจับเวลา/ตัวนับเท่านั้น ขณะที่ขา INTx มีสถานะสูง และขาควบคุม TRx ใน TCON จะถูกเซตเป็น "1" เมื่อตัวนับภายในถูกเคลียร์ให้อีนาเบิ้ล เมื่อไรก็ตามที่บิตควบคุม TRx ถูกเซตเป็น "1"
C/T		เลือกการทำงานแบบตัวจับเวลาหรือตัวนับ ถ้าเป็น "0" จะเลือกทำงานเป็นตัวจับเวลา (โดยใช้สัญญาณนาฬิกาภายในเป็นสัญญาณเข้าอ้างอิง) ถ้าเป็น "1" จะเป็นการทำงานแบบตัวนับ และรับสัญญาณเข้าที่ขา Tx
M1	M0	การทำงาน
0	0	ทำงานแบบตัวจับเวลาของ MCS-48 ใช้ TLx เป็นตัวป้อนบิตอีก 5 บิต
0	1	การใช้ตัวจับเวลา/ตัวนับขนาด 16 บิตจะใช้ THx และ TLxเป็นตัวนับไม่มี Prescaler
1	0	การโหลดขนาด 8 บิตโดยอัตโนมัติที่ตัวนับและตัวจับเวลา โดยใช้ THx เก็บค่าที่ตั้งไว้และจะถ่ายเข้าไปที่ TLx ใหม่ทุกครั้งที่เกิด Overflow คือ TLx ถูกนับเป็น "0" หมด
1	1	ตัวจับเวลา 0 ทำงาน โดยให้ TL0 และ TH0 เป็นตัวนับแยกกัน

ตารางที่ 2.5 TMOD: Timer/Counter Mode Control Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1	TCON.7	ตัวจับเวลา 1 แฟลกเป็น “1” เมื่อเกิด Overflow ถูกเซตเป็นหนึ่งด้วยฮาร์ดแวร์ทางสัญญาณ เมื่อตัวจับเวลา/ตัวนับ Overflow และจะเคลียร์ตัวเองเมื่ออินเตอร์รัพต์ไปแล้ว
TR1	TCON.6	ตัวจับเวลา 1 เป็นตัวควบคุมบิตให้เริ่มทำงาน จะเซตหรือเคลียร์ด้วยซอฟต์แวร์ที่จะมาทำให้ ตัวจับเวลา/ตัวนับ 1 เริ่มหรือหยุดทำงาน
TF0	TCON.5	ตัวจับเวลา 0 แฟลกเป็น “1” เมื่อเกิด Overflow ถูกเซตเป็นหนึ่งด้วยฮาร์ดแวร์ทางสัญญาณ เมื่อตัวจับเวลา/ตัวนับ Overflow และจะเคลียร์ตัวเองเมื่ออินเตอร์รัพต์ไปแล้ว
TR0	TCON.4	ตัวจับเวลา 0 เป็นตัวควบคุมบิตให้เริ่มทำงาน จะเซตหรือเคลียร์ด้วยซอฟต์แวร์ที่จะมาทำให้ ตัวจับเวลา/ตัวนับ 1 เริ่มหรือหยุดทำงาน
IE1	TCON.3	เป็นแฟลกของสัญญาณอินเตอร์รัพต์1จะเซตด้วยฮาร์ดแวร์เมื่อสัญญาณของการอินเตอร์รัพต์ปรากฏเข้าที่ขาINT1 และเคลียร์เมื่อการทำงานอินเตอร์รัพต์สิ้นสุดลง
IT1	TCON.2	รูปแบบการควบคุมบิตของอินเตอร์รัพต์ 1 จะเซตหรือเคลียร์ได้ด้วยซอร์ฟแวร์ที่จะเป็นตัวกำหนดให้มีการกระตุ้นอินเตอร์รัพต์จากภายนอกที่เป็นขอบขาลงหรือเป็นระดับแรงดันต่ำ โดยถ้า IT=1 จะควบคุมอินเตอร์รัพต์แบบขอบขาลง และถ้า IT=0 จะควบคุมอินเตอร์รัพต์แบบระดับแรงดันต่ำ
IE0	TCON.1	เป็นแฟลกของสัญญาณอินเตอร์รัพต์1จะเซตด้วยฮาร์ดแวร์เมื่อสัญญาณของการอินเตอร์รัพต์ปรากฏเข้าที่ขาINT0 และเคลียร์เมื่อการทำงานอินเตอร์รัพต์สิ้นสุดลง
IT0	TCON.0	รูปแบบการควบคุมบิตของอินเตอร์รัพต์ 0 จะเซตหรือเคลียร์ได้ด้วยซอร์ฟแวร์ที่จะเป็นตัวกำหนดให้มีการกระตุ้นอินเตอร์รัพต์จากภายนอกที่เป็นขอบขาลงหรือเป็นระดับแรงดันต่ำ โดยถ้า IT=1 จะควบคุมอินเตอร์รัพต์แบบขอบขาลง และถ้า IT=0 จะควบคุมอินเตอร์รัพต์แบบระดับแรงดันต่ำ

ตารางที่ 2.6 TCON :Timer/Counter Control Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.9 การเชื่อมต่อแบบอนุกรม

พอร์ตอนุกรมเป็นแบบฟูลดูเพล็กซ์ สามารถที่จะใช้ส่งและรับข้อมูลพร้อมกันได้ โดยทำหน้าที่เป็นบัฟเฟอร์การรับ หมายถึงพอร์ตสามารถที่จะรับไบต์ที่สองก่อนที่ตัวแรกจะถูกรับไปจากรีจิสเตอร์ตัวรับ อย่างไรก็ตามไบต์ตัวแรกจะต้องถูกอ่านไปก่อนที่ช่วงเวลาการรับไบต์ที่สองจะสิ้นสุดมิฉะนั้นไบต์ตัวแรกจะถูกซ้อนและสูญหายไปได้ ในพอร์ตอนุกรมมีรีจิสเตอร์ตัวรับและส่ง จะเข้าถึงติดต่อกันด้วยรีจิสเตอร์ SBUF ใน SFR แม้ว่าโครงสร้างทางรีจิสเตอร์ทั้งสองจะแยกกันอยู่ก็ตาม

พอร์ตอนุกรมที่จะเลือกทำงานในโหมดต่างๆ ได้ใน 4 โหมด

- โหมด 0 ข้อมูลจะผ่านเข้าออกผ่าน RXD TXD ด้วยการเลื่อนสัญญาณนาฬิกาเอาท์พุต ข้อมูลจะเป็นลักษณะ 8 บิต ในการรับส่งแต่ละครั้ง โดยที่ส่งค่า LSB ก่อนอัตราบอดจะคงที่ที่ 1/12 ของความถี่ออสซิลเลเตอร์

- โหมด 1 จะเป็นการส่งข้อมูลขนาด 10 บิตผ่านออก TXD หรือรับเข้ามาผ่าน RXD โดยรูปแบบบิตจะประกอบด้วยหนึ่งบิตเริ่มต้นเป็น “0” แปดบิตข้อมูลโดย LSB เป็นตัวแรกที่รับและส่ง และอีกหนึ่งบิตสิ้นสุดมีค่า “1” การรับบิตสิ้นสุดจะนำไปเก็บที่บิต RB8 ของ SFR รีจิสเตอร์ SCON อัตราบอดแปรผันได้ตามการตั้งตัวจับเวลาซึ่งจะกล่าวในหัวข้อต่อไป

- โหมด 2 เป็นการส่งข้อมูลขนาด 11 บิตผ่านออกขา TXD หรือรับเข้ามาผ่านขา RXD ประกอบด้วยหนึ่งบิตเริ่มต้นมีค่า “0” แปดบิตข้อมูล โดย LSB เป็นตัวแรกที่รับและส่งข้อมูลบิตที่เก้าของข้อมูลสามารถที่จะโปรแกรมเลือกได้ และบิตสิ้นสุดค่า “1” อีกหนึ่งบิต ในการส่งบิตที่เก้าที่อยู่ในบิต TB8ของรีจิสเตอร์ SCON สามารถที่จะกำหนดเลือกเป็น “1” หรือ “0” ได้ ตัวอย่างเช่น การใช้งานแบบบิตพาริตี้ โดยการเลื่อนเอาบิต P ของ PSW มาไว้ใน TB8 เพื่อเป็นการส่งข้อมูลแบบมีการตรวจพาริตี้ของข้อมูลที่ส่ง ในการรับข้อมูลบิตที่เก้าจะเข้าไปเก็บที่ RB ใน SFR รีจิสเตอร์ SCON ขณะที่บิตสิ้นสุดจะไปรับเข้ามาเก็บ อัตราบอดสามารถเลือกเป็น 1/32 หรือ 1/64 ของความถี่ออสซิเลเตอร์ SCON เป็น SFR ที่ใช้ในการติดตั้งโหมดการทำงานพอร์ตอนุกรม เช่น การกำหนดค่า RB8 จะเป็นการใช้ตัวรับบิตที่เก้าด้วยหรือไม่ เป็นต้น ตารางที่ 5-6 เป็นตารางการใช้ SCON ในการควบคุมการทำงานของพอร์ตอนุกรม

- โหมด 3 เป็นการส่งข้อมูลขนาด 11 บิตผ่านออกขา TXD หรือรับเข้ามาผ่านขา RXD ประกอบด้วยบิตเริ่มต้นมีค่าเป็น “0” ข้อมูลแปดบิตโดย เป็นบิตแรกที่รับและส่งข้อมูลที่บิตเก้าของข้อมูลสามารถที่จะโปรแกรมได้ และบิตสิ้นสุดค่า “1” อีกหนึ่งบิต ในความเป็นจริงโหมด 3 จะคล้ายกับโหมด 2 ทุกประการ ยกเว้นอัตราบอด โดยอัตราบอดในโหมด 3 จะแปรผันได้ไปตามการโปรแกรมเลือกตัวจับเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทั้งสี่โหมดนี้ การส่งข้อมูลจะเริ่มติดตั้ง Initiated ด้วยคำสั่งใดๆ ที่ใช้ตัวรีจิสเตอร์ SBUF เป็นรีจิสเตอร์ตัวรับข้อมูลจากซีพียู และในโหมด 0 การรับข้อมูลเริ่มติดตั้งด้วยการใช้สถานะ RI=0 และ REN=1 ส่วนในโหมดอื่นการรับข้อมูลจะเริ่มติดตั้งด้วยการรับบิตเริ่มต้นเข้ามาตรวจสอบถ้า REN=1 การทำงานทั้งสี่โหมดของพอร์ตอนุกรมได้สรุปไว้ในตารางที่ 2.12 ต่อไปนี้เป็นการทำงานโดยละเอียดในโหมดต่างๆ ของพอร์ตอนุกรม

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

โดย SM0 , SM1 เป็นตัวกำหนดโหมดต่าง ๆ ของพอร์ตอนุกรม ดังนี้

SM0	SM1	โหมด	ลักษณะการทำงาน อัตราบอด
0	0	0	เลื่อนรจิสเตอร์ fosc/12
0	1	1	8-บิต UART แปรผันได้ตามการเลือกตัวจับเวลา
1	0	2	9-บิต UART fosc/64 หรือ fosc/32
1	1	3	9-บิต UART แปรผัน

* UART : Universal Asynchronous Receiver / Tranceiver

SM2 ควบคุมอินเทอร์เฟส การใช้โหมดเซนเซอร์หลายตัวในการสื่อสารซึ่งกันและกันในโหมด 2 และ 3 ถ้า SM2 เซตเป็น 1 ดังนั้น RI จะต้องไม่แอกทีฟ ถ้ามีการรับบิตที่เก้า ทำให้บิต RB8 นี้ เป็น 0ในโหมด 1 ถ้า SM2 เซตเป็น 1 ดังนั้น RI จะไม่แอกทีฟถ้า STOP บิตไม่ถูกรับเข้ามาในโหมด 0 SM2 ควรมีค่าเท่ากับ 0

REN ตัวอินเทอร์เฟสอนุกรมการรับ เซตเป็น '1' ด้วยโปรแกรมในการเลือกอินเทอร์เฟส รรับและเป็น '0' ด้วยโปรแกรม เมื่อให้เป็นคลีอเปิดการรับ

TB8 เป็นข้อมูลบิตที่เก้า ซึ่งจะถูกส่งในโหมด 2 และ 3 ซึ่งจะให้เป็น '1' หรอ '0'ได้ด้วยการโปรแกรม

RB8 ในโหมด 2 และ 3 ข้อมูลบิตที่เก้าจะถูกรับไปในโหมด 1 ถ้า SM2 = 0 RB8 จะกลายเป็น STOP บิตที่ถูกรับไปในโหมด 0 RB8 ไม่ใช้

TI เป็นแฟลกอินเตอร์รัพต์การส่ง เซตด้วยฮาร์ดแวร์คือสัญญาณปลายช่วงเวลาของบิตที่แปดในโหมด 0 หรือที่จุดเริ่มต้นของบิต STOP ในโหมดอื่น ในการส่งแบบอนุกรมของทุกโหมดจะต้องเคลียร์บิตนี้ด้วยโปรแกรมหลังการสมเสร็จ

RI เป็นแฟลกอินเตอร์รัพต์การรับ เซตด้วย ฮาร์ดแวร์คือสัญญาณที่ปลายช่วงเวลาของบิตที่แปดในโหมด 0 หรือที่จุดครึ่งทางของช่วงบิต STOP ในโหมดอื่น ในการรับแบบอนุกรมยกเว้นกรณีใช้ SM2 จะต้องเคลียร์บิตด้วยโปรแกรมหลังการรับข้อมูลไปแล้ว

ตารางที่ 2.7 SCON:รีจิสเตอร์ควบคุมพอร์ตอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

2.1.9.1 อัตราบอด

อัตราบอดในโหมด 0 ของการใช้พอร์ตอนุกรมจะคงที่ที่ความถี่ออสซิลเลเตอร์คือ

อัตราบอดในโหมด 0 = Oscillator Frequency/12

อัตราบอดในโหมด 2 จะขึ้นอยู่กับการปรับค่าบิตใน SMOD ของ SFR ในรีจิสเตอร์ PCON ถ้า SMOD=0 ซึ่งจะเป็นค่าที่ถูกรีเซ็ตแต่แรก หลังการรีเซ็ตอัตราบอดจะเป็น 1/64 ของความถี่ออสซิลเลเตอร์ ถ้า SMOD=1 อัตราบอดจะเป็น 1/32 ของความถี่ออสซิลเลเตอร์มีสูตรเป็น

อัตราบอดในโหมด 2 =(2 SMOD/64)*Oscillator Frequency

ใน MCS-51 อัตราบอดในโหมด 1 และ 3 ถูกกำหนดได้ด้วยอัตราโอเวอร์โฟลว์ที่เกิดขึ้นจากการกำหนดค่าในรีจิสเตอร์ TH1 ของตัวจับเวลา 1 ส่วนใน 8052 อัตราบอดเหล่านี้สามารถคำนวณได้จากตัวจับเวลา 1 หรือตัวจับเวลา 2 หรือใช้ทั้งสองตัว โดยตัวใดตัวหนึ่งสำหรับส่งและอีกตัวหนึ่งสำหรับรับ

การใช้ตัวจับเวลา 1 เป็นตัวสร้างอัตราบอด เมื่อใช้ตัวจับเวลา 1 เป็นตัวสร้างอัตราบอด อัตราบอดในโหมด 1 และ 3 จะถูกคำนวณด้วยอัตราโอเวอร์โฟลว์ที่เกิดขึ้นในตัวจับเวลา 1 และค่าบิตใน SMOD ซึ่งสูตรการคำนวณเป็นดังนี้

อัตราบอดในโหมด 1,3 = (2 SMOD/32)*Timer 1 Overflow Rate

การอินเตอร์รัพต์ตัวจับเวลา 1 ควรที่จะดิสเอเบิ้ลในการใช้งานแบบนี้ ตัวจับเวลาในตัวมันเองสามารถที่จะถูกกำหนดให้ใช้เป็นตัวจับเวลาหรือตัวนับการทำงานในโหมด 3 ในการใช้งานลักษณะนี้มันจะถูกกำหนดให้เป็นตัวจับเวลาในโหมดแบบบรรจุอัตโนมัติ (โดยตั้งให้ HIGH NIBBLE ของ TMOD=0010B)ในกรณีนี้อัตราบอดคำนวณได้ดังสูตร

อัตราบอดในโหมด 1,3 = (2 SMOD/32)*(Oscillator Frequency/12*(256-TH1))

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

			Timer		
BAUD	fosc	SMOD	C/T	MODE	RELOAD VALUE
Mode 0 Max 1 MHz	12 MHz	X	X	X	X
Mode 2 Max 375 K	12 MHz	1	X	X	X
Modes 1,3 62.5K	12MHz	1	0	2	FFH
19.2K	11.059 MHz	1	0	2	FDH
9.6K	11.059 MHz	0	0	2	FDH
4.8K	11.059 MHz	0	0	2	FAH
2.4K	11.059 MHz	0	0	2	F4H
1.2K	11.059 MHz	0	0	2	E8H
137.5	11.059 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEBH

ตารางที่ 2.8 เป็นรายการอัตราบอดที่ใช้ตัวจับเวลา 1

ในอัตราการบอดมีค่าต่ำมาก ก็สามารถที่จะทำได้โดยการตั้งตัวจับเวลา 1 ให้สามารถรองรับการอินเตอร์รัพต์ได้ และกำหนดให้ตัวจับเวลาทำงานเป็น 16 บิต (โดยตั้งค่า HIGH NIBBLE ของ TMOD=0001B) และใช้ตัวจับเวลา 1 ให้ทำการอินเตอร์รัพต์เมื่อเกิดโอเวอร์โฟลว์ และบรรจุค่า 16 บิตไปใหม่ ในกรณีนี้ต้องการอินเตอร์รัพต์ที่ตัวจับเวลา 1 จึงให้ IE.3=1 ในกรณีถ้าตัวจับเวลา 1 กำลังทำงานที่บิต C/T=0 อัตราการนับเป็น 1/12 ของความถี่ออสซิเลเตอร์ ถ้าตัวจับเวลาทำงานที่บิต C/T=1 อัตราการนับจะใช้ความถี่ภายนอกที่ส่งเข้ามาซึ่งจะมีความถี่สูงสุดที่จะใช้ได้คือ 1/24 ของความถี่ออสซิเลเตอร์

ตารางที่ 2.8 เป็นรายการอัตราบอดที่ใช้กันทั่วไป และค่าต่างๆ ที่จะใส่ในตัวจับเวลา 1 แสดงแผนภูมิฟังก์ชันของพอร์ตอนุกรมในโหมด 3 จากการคำนวณ เพื่อให้ได้อัตราบอดตามกำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 หน่วยแสดงผล 7-segment

2.2.1 ขาสัญญาณของ 7-segment

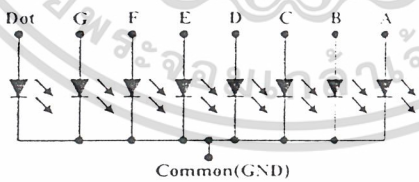
7-segment จะใช้การจัดวางโครงสร้างของ LED 7 ดวง ดังรูปที่ 2-12 เพื่อแสดงผล ซึ่ง LED แต่ละดวงจะแทนเซกเมนท์ต่างๆ โดยมีชื่อเรียกเป็นเซกเมนท์ A,B,C,D,E,F,G,H

รูปที่ 2-12 7-segment

2.2.2 โครงสร้างการทำงานของ 7-segment

มีโครงสร้างการทำงาน 2 แบบ

1) คาโทดร่วม (Common Cathode) คือการนำขาคาโทดมาต่อร่วมกัน โดยปกติแล้วจะมีการต่อใช้งานดังรูปที่ 2-9 กล่าวคือขาคาโทดจะถูกต่อลงกราวด์ เพราะฉะนั้นการทำงานของเซกเมนท์จะขึ้นอยู่กับแรงดันที่ขาคาโทด โดยที่ถ้าแรงดันเกินประมาณ 1.5 โวลท์จะทำให้ LED ของเซกเมนท์สามารถทำงาน(สว่าง) และถ้าแรงดันที่ขาคาโทดต่ำกว่านี้จะทำให้ LED ของเซกเมนท์ไม่ทำงาน(ดับ)



รูปที่ 2-13 คาโทดร่วม

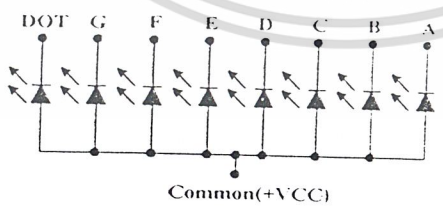
การกำหนดค่าของเซกเมนท์ในการแสดงผลตัวเลข 0-9 จากตารางที่ 2.14 ถ้าต้องการให้เซกเมนท์ทำงานจะต้องป้อนลอจิก “1” ให้กับขาคาโนด และถ้าไม่ต้องการให้เซกเมนท์ทำงาน จะต้องป้อนลอจิก “0” ให้กับขาอาโนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวเลข	เซกเมนท์								ข้อมูล
	H	G	F	E	D	C	B	A	
0	0	0	1	1	1	1	1	1	3FH
1	0	0	0	0	0	1	1	0	06H
2	0	1	0	1	1	0	1	1	5BH
3	0	1	0	0	1	1	1	1	4FH
4	0	1	1	0	0	1	1	0	66H
5	0	1	1	0	1	1	0	1	6DH
6	0	1	1	1	1	1	0	1	7DH
7	0	0	0	0	0	1	1	1	07H
8	0	1	1	1	1	1	1	1	7FH
9	0	1	1	0	1	1	1	1	6FH

ตารางที่ 2.9 แสดงการกำหนดค่าของเซกเมนท์ในการแสดงผลตัวเลข 0-9

2) อาโนดร่วม (Common Anode) คือการนำขาอาโนดมาต่อร่วมกัน โดยปกติแล้วการใช้งานจะนำขาอาโนดร่วมมาต่อกับแรงดันที่เป็นบวกดังรูปที่ 2-10 ดังนั้นการทำงานของ LED ในแต่ละเซกเมนท์ จะขึ้นกับแรงดันของขาคาโทด ถ้าระดับแรงดันที่ขาอาโนดทำให้เกิดการไบอัสตรง (Forward Bias) และมีแรงดันตกคร่อม LED เกิน1.5 โวลท์จะสามารถทำให้ LED ของเซกเมนท์นั้นทำงาน(สว่าง) ถ้าไม่เป็นดังที่กล่าวมา LED ของเซกเมนท์นั้น จะไม่ทำงาน(ดับ)



รูปที่ 2-14 อาโนดร่วม

การกำหนดค่าของเซกเมนท์ต่างๆ เพื่อแสดงผลตัวเลข 0-9 จากตารางที่ 2.15 ถ้าต้องการให้เซกเมนท์ทำงานจะต้องป้อนลอจิก "0"ให้กับขาคาโทด และถ้าไม่ต้องการให้เซกเมนท์ทำงานจะต้องป้อนลอจิก "1"ให้กับขาคาโทด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

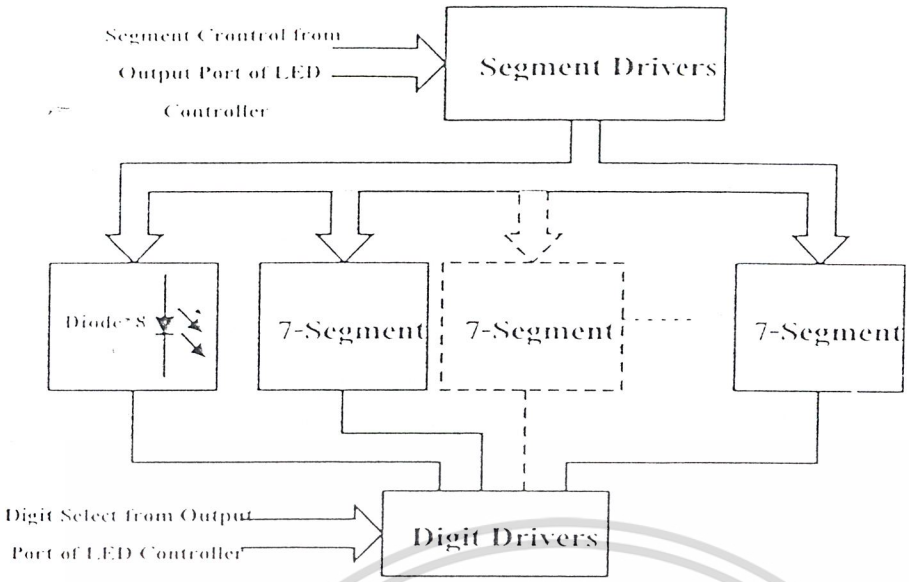
ตัวเลข	เซกเมนท์								ข้อมูล
	H	G	F	E	D	C	B	A	
0	1	1	0	0	0	0	0	0	C0H
1	1	1	1	1	1	0	0	1	F9H
2	1	0	1	0	0	1	0	0	A4H
3	1	0	1	1	0	0	0	0	B0H
4	1	0	0	1	1	0	0	1	99H
5	1	0	0	1	0	0	1	0	92H
6	1	0	0	0	0	0	1	0	82H
7	1	1	1	1	1	0	0	0	F8H
8	1	0	0	0	0	0	0	0	80H
9	1	0	0	1	0	0	0	0	90H

ตารางที่ 2.10 แสดงการกำหนดค่าของเซกเมนท์ในการแสดงผลตัวเลข 0-9

หลักการแสดงผลแบบ Multiplex

การแสดงผลใช้วงจรขับตามรูปที่ 2-11 จะทำงานได้ในลักษณะการแสดงผลที่น้อยๆ เพราะถ้ามีการแสดงผลมากขึ้น เช่น แสดงผล 7-segment 10 หลัก ดังนั้นการแสดงผลขณะที่เซกเมนท์ทุกตัว 'ON' ทั้งหมด สมมติว่าแต่ละ 7-segment 1 หลัก กินกระแสประมาณ 80 mA เพราะฉะนั้นส่วนแสดงผล 10 หลัก จะกินกระแสทั้งหมด 800 mA ซึ่งเป็นกระแสที่สูงมาก การแก้ปัญหาที่กล่าวมาจะใช้หลักในการสร้างภาพนิ่ง คือการให้ภาพกลับมาแสดงใหม่ 25 ภาพต่อวินาทีจากเหตุผลดังกล่าว ถ้ามีการแสดงผลเป็นดังรูปที่ 2-11 จะทำโดยการส่งสัญญาณ 'ON' และ 'OFF' เป็นพัลส์ไปที่หลักต่างๆ โดยที่จะต้องกลับมาป้อนพัลส์ไปที่หลักเดิม ภายในเวลา 20 mS (50 Hz) จะเสมือนว่าหลักนั้นมีกาทำงานโดยจ่ายกระแสอย่างต่อเนื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2-15 การสแกน 7-segment

ตัวอย่าง กระบวนการแสดงผลจะเป็นดังนี้

1. ส่งข้อมูลไปยังส่วนแสดงผลหลักที่ 1
2. 'ON' หลักที่ 1
3. รอขณะที่ หลักที่ 1 'ON'
4. 'OFF' หลักที่ 1
5. ส่งข้อมูลไปยังส่วนแสดงผลหลักที่ 2
6. 'ON' หลักที่ 2
7. รอขณะที่ หลักที่ 2 'ON'
8. 'OFF' หลักที่ 2
9. ทำขั้นตอนเหล่านี้ซ้ำไปทุกๆ หลักจนครบ แล้วกลับไปทำงานหลักที่ 1 ใหม่

โดยปกติแล้ว การกลับมาแสดงผลควรจะมามากกว่า 30 ครั้ง ต่อ 1 วินาที เพื่อไม่ให้เกิดการกระพริบของ LED (Segment) แต่การใช้งานที่ดีที่สุดจะประมาณ 50-60 ครั้งต่อวินาที (20 mS)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

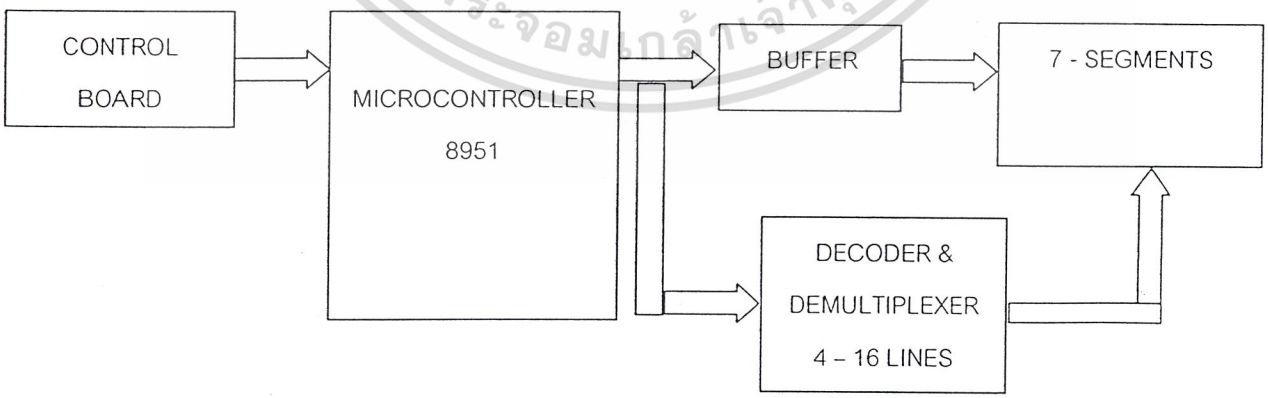
การออกแบบการทำงาน

การออกแบบการทำงานของแป้นควบคุมและแสดงผลการแข่งขันบาสเกตบอลจะแบ่งออกเป็น 2 ส่วนใหญ่ๆ คือ ทางด้านฮาร์ดแวร์และซอร์ฟแวร์ โดยจะเน้นการทำงานหลักๆ เพื่อง่ายต่อการเข้าใจ การทำงานทั้งหมดจะถูกควบคุมโดยไมโครคอนโทรลเลอร์ โดยมีการรับข้อมูลได้ทางเดียวคือผ่านทางบอร์ดควบคุม เพื่อส่งมาให้ไมโครคอนโทรลเลอร์ประมวลผลและควบคุมการสร้างสัญญาณในการแสดงผลข้อมูลการแข่งขันออกทาง 7-segment การทำงานของแป้นควบคุมและแสดงผลการแข่งขันบาสเกตบอลมีความสามารถคือ

1. สามารถตั้งเวลาที่ใช้ในการแข่งขันได้ทั้งหมด 4 ค่า คือ 5 , 10 , 15 และ 20 นาที
2. แสดงผลการทำคะแนนของแต่ละทีม
3. แสดงผลจำนวนครั้งในการทำฟาล์วของแต่ละทีม
4. แสดงผลจำนวนครั้งในการขอเวลานอกของแต่ละทีม

3.1 การออกแบบด้านฮาร์ดแวร์ (Hardware)

การออกแบบด้านฮาร์ดแวร์นั้นจะเป็นการออกแบบทีละส่วนของการทำงาน โดยจะมีการทำงานเป็นสองส่วนใหญ่ๆ คือ ส่วนของการรับข้อมูล และส่วนของการแสดงผลข้อมูล



รูปที่ 3-1 แสดงบล็อกไดอะแกรมการทำงานของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 วงจรส่วนควบคุม

การออกแบบวงจรส่วนควบคุมหลัก เพื่อควบคุมการทำงานของอุปกรณ์ ซึ่งเป็นตัวกลางการติดต่อระหว่างผู้ใช้กับอุปกรณ์ หรืออุปกรณ์กับผู้ใช้ โดยหลักการทำงานของส่วนควบคุมหลักจะเป็นส่วนที่ทำการสร้างสัญญาณเพื่อไปควบคุมการทำงานของหน่วยแสดงผลและทำการประมวลผลในโครงงาน ในโครงงานนี้ใช้ไมโครคอนโทรลเลอร์ 8051 ซึ่งเป็นการติดต่อกัน โดยใช้สายนำสัญญาณข้อมูล (Data bus) ไมโครคอนโทรลเลอร์จะเป็นตัวควบคุมหลัก แต่ระดับของสัญญาณที่ออกจะเป็นระดับสัญญาณ TTL คือลอจิก "1" จะมีแรงดันเท่ากับ 5 โวลต์ และลอจิก "0" จะมีแรงดันเท่ากับ 0 โวลต์ ในส่วนควบคุมนี้จะประกอบด้วยการทำงานหลายๆ ส่วนซึ่งได้แก่

- ส่วนฐานเวลาของระบบ (Real Time Clock)

ระบบจำเป็นต้องมีฐานเวลาที่ใช้กำหนดเวลาการทำงานของอุปกรณ์ควบคุม ซึ่งในโครงงานนี้กำหนดให้ 8051 ใช้ฐานเวลาจากคริสตัล 12 MH_z เพื่อความสะดวกในการเขียนซอฟแวร์ควบคุมการนับเวลาถอยหลังของระบบ

- ส่วนอินพุต

การออกแบบส่วนอินพุตให้กับไมโครคอนโทรลเลอร์นั้น มีลักษณะการเชื่อมต่อแบบไม่มัลติเพลกซ์ (Non-Multiplex) โดยการต่อสวิทช์แต่ละตัวเข้ากับพอร์ทอินพุตของระบบ โดยจะมีตัวต้านทาน pull up ไว้ที่ อินพุตพอร์ท ซึ่งเหมาะที่ใช้งานกับสวิทช์ประเภท DIP หรือสวิทช์คีย์บอร์ดที่มีจำนวนน้อย

สวิทช์ลำดับที่	หน้าที่	หมายเลขพอร์ท
SW 1 – SW 2	การขอเวลานอกของทีม B	# PORT 1
SW 3 – SW 4	การทำฟาล์วรวมของทีม B	
SW 5 – SW 6	Countdown Mode	
SW 7 – SW 8	การทำคะแนนของทีม B	
SW 9 – SW 10	การขอเวลานอกของทีม A	# PORT 3
SW 11 – SW 12	การทำฟาล์วรวมของทีม A	
SW 13 – SW 14	Countdown Mode	
SW 15 – SW 16	การทำคะแนนของทีม A	

ตาราง 3-1 แสดงการทำงานของอินพุตพอร์ท 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

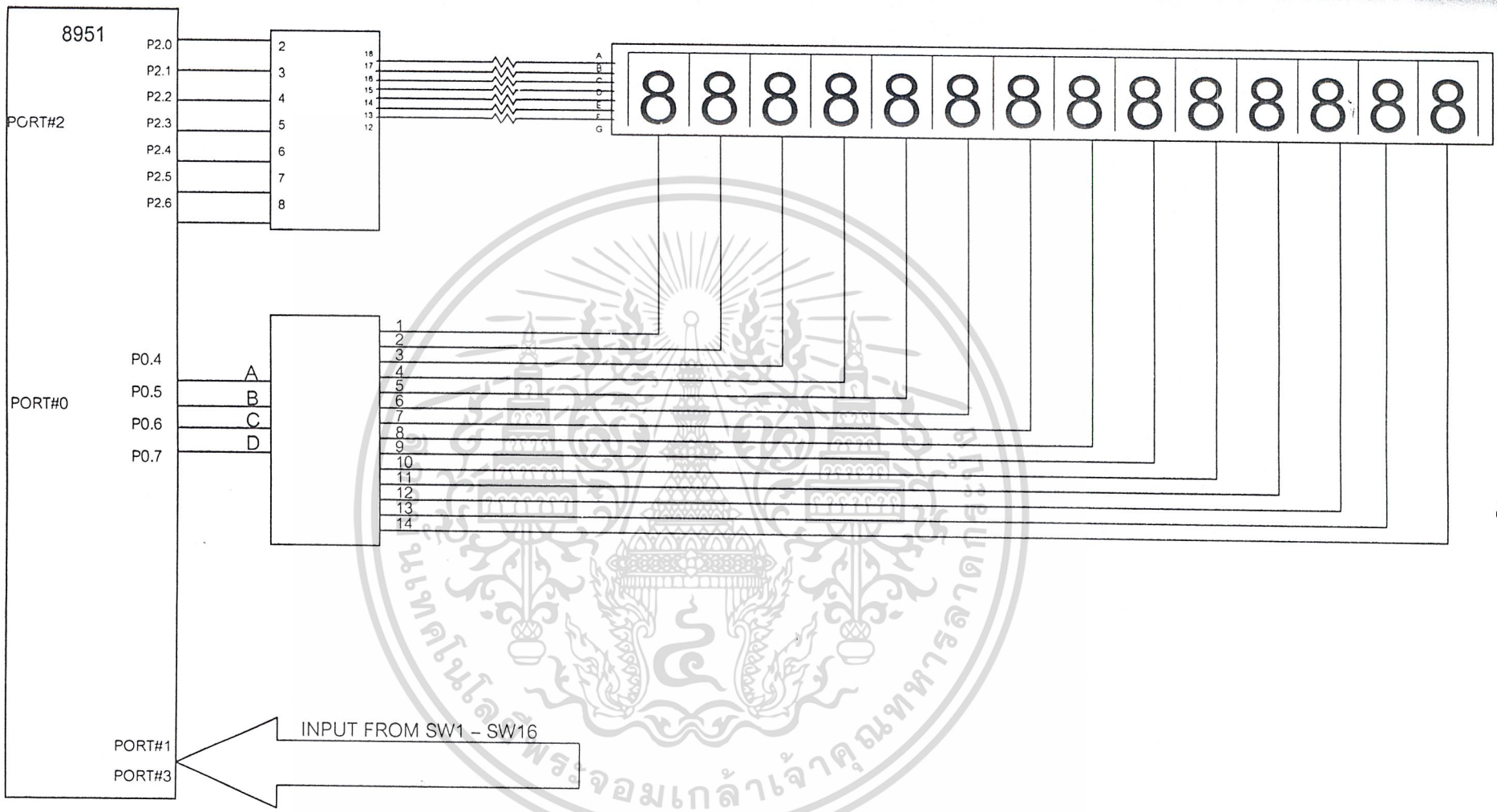
- ส่วนเอาท์พุต

การออกแบบส่วนของเอาท์พุต สำหรับติดต่อกับหน่วยแสดงผลได้ใช้ IC 7425245 เข้าช่วยในการรักษาระดับสัญญาณก่อนที่จะต่อเข้ากับ 7-segment และ IC 7425154 เข้ามาช่วยในการควบคุม 7-segment โดยได้ใช้ port # 0 และ port # 1 ซึ่งแต่ละพอร์ทมีกี่ทำงานดังนี้

หน้าที่	พอร์ท
ส่งสัญญาณควบคุม 7-segment ที่ตำแหน่งขาสัญญาณ a	P 0.0
ส่งสัญญาณควบคุม 7-segment ที่ตำแหน่งขาสัญญาณ b	P 0.1
ส่งสัญญาณควบคุม 7-segment ที่ตำแหน่งขาสัญญาณ c	P 0.2
ส่งสัญญาณควบคุม 7-segment ที่ตำแหน่งขาสัญญาณ d	P 0.3
ส่งสัญญาณควบคุม 7-segment ที่ตำแหน่งขาสัญญาณ e	P 0.4
ส่งสัญญาณควบคุม 7-segment ที่ตำแหน่งขาสัญญาณ f	P 0.5
ส่งสัญญาณควบคุม 7-segment ที่ตำแหน่งขาสัญญาณ g	P 0.6
ควบคุม IC 7425154 ที่ตำแหน่งขาสัญญาณ A	P 1.0
ควบคุม IC 7425154 ที่ตำแหน่งขาสัญญาณ B	P 1.1
ควบคุม IC 7425154 ที่ตำแหน่งขาสัญญาณ C	P 1.2
ควบคุม IC 7425154 ที่ตำแหน่งขาสัญญาณ D	P 1.3

ตารางที่ 3-2 แสดงการทำงานของเอาท์พุตพอร์ท 8051

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

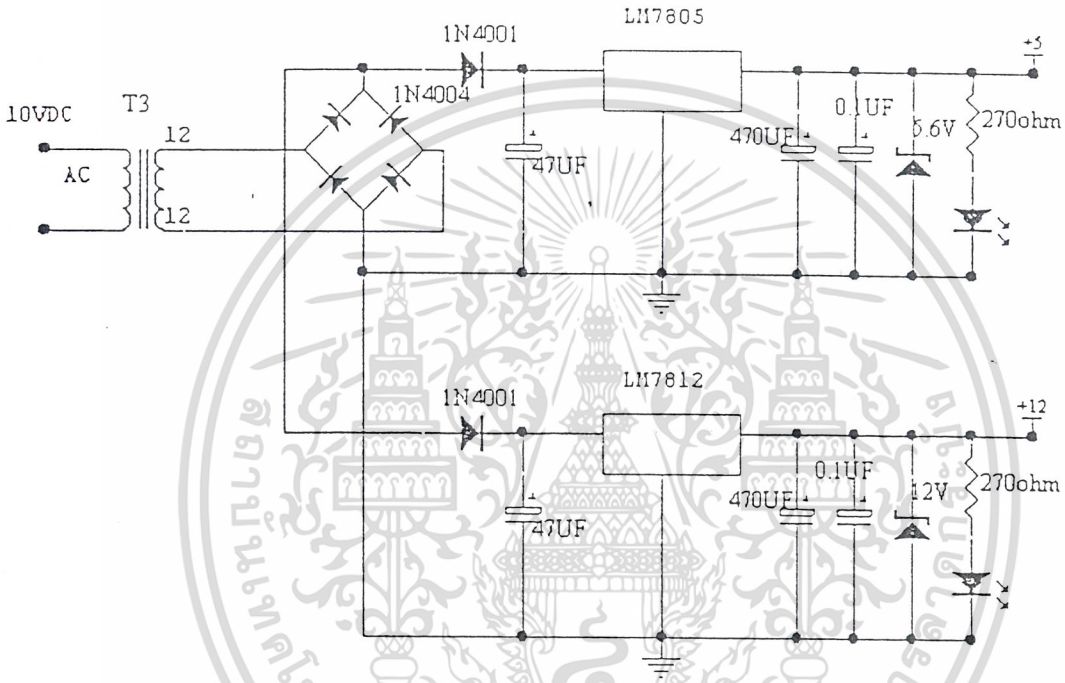


รูปที่ 3-2 แสดงวงจรบอร์ดควบคุม

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

3.1.2 วงจรกำเนิดแรงดัน (Power Supply)

วงจรกำเนิดแรงดันจะทำหน้าที่เป็นวงจรกำเนิดแรงดันเพื่อเลี้ยงส่วนต่างๆ ของวงจรให้วงจรทำงานได้ซึ่งวงจรต้องการไฟเลี้ยงทั้งขนาด +12 โวลท์ ในส่วนของวงจรตรวจจับสัญญาณจาก SENSOR และขนาด +5 โวลท์ ในส่วนของวงจรอื่นๆการทำงานของวงจรกำเนิดแรงดันแสดงดังรูป 3-2



รูป 3-3 วงจรกำเนิดแรงดัน

การทำงานของวงจรเริ่มจากมีการรับแรงดันไฟสลับ (AC) 220 โวลท์ ผ่านมาที่หม้อแปลงและบริดจ์เพื่อทำหน้าที่ในการแปลงแรงดันจากแรงดันไฟสลับเป็นแรงดันไฟตรง (DC) และใช้ IC 7805 เป็นตัวกรองแรงดัน (Regulator) ให้เรียบจากนั้นแรงดันที่ได้จะถูกกำจัดโดยซีเนอร์ไดโอด(Zener Diode) ขนาด 5.6 โวลท์ เพื่อไม่ให้แรงดันเกินกว่า 5.6 โวลท์ นั่นก็คือประมาณ 5 โวลท์ เพื่อป้อนให้กับวงจร

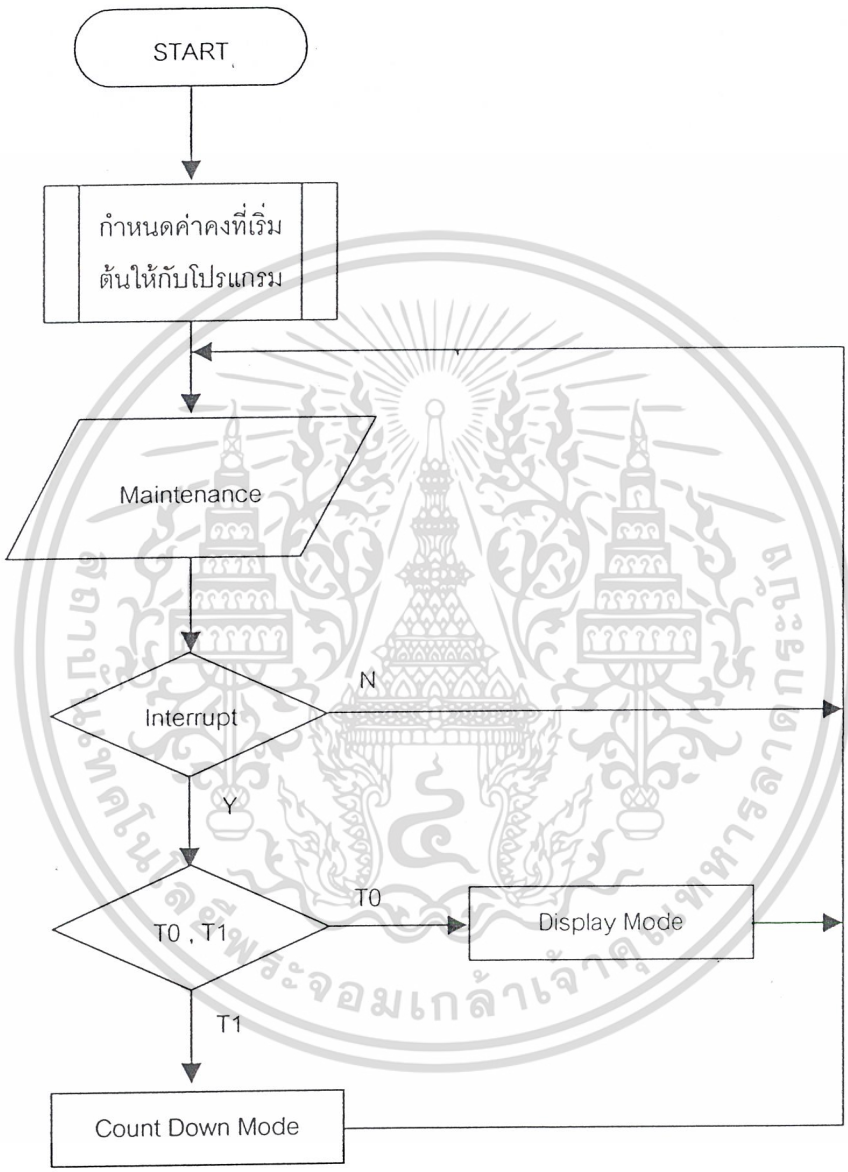
ส่วนแรงดันไฟ 12 โวลท์นั้นมีลักษณะการทำงานเหมือนกับแรงดันไฟ 5 โวลท์ ต่างกันที่ตัวกรองแรงดันซึ่งใช้ IC 7812 และซีเนอร์ไดโอดขนาด 12 โวลท์เป็นตัวจำกัดแรงดันเพื่อให้ได้แรงดันไฟ 12 โวลท์ สำหรับป้อนให้กับวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การออกแบบด้านซอร์ฟแวร์

การออกแบบด้านซอร์ฟแวร์จะออกแบบเป็นส่วนๆ และจะอธิบายการทำงานเป็นส่วนๆ คือ

3.2.1 Main Program

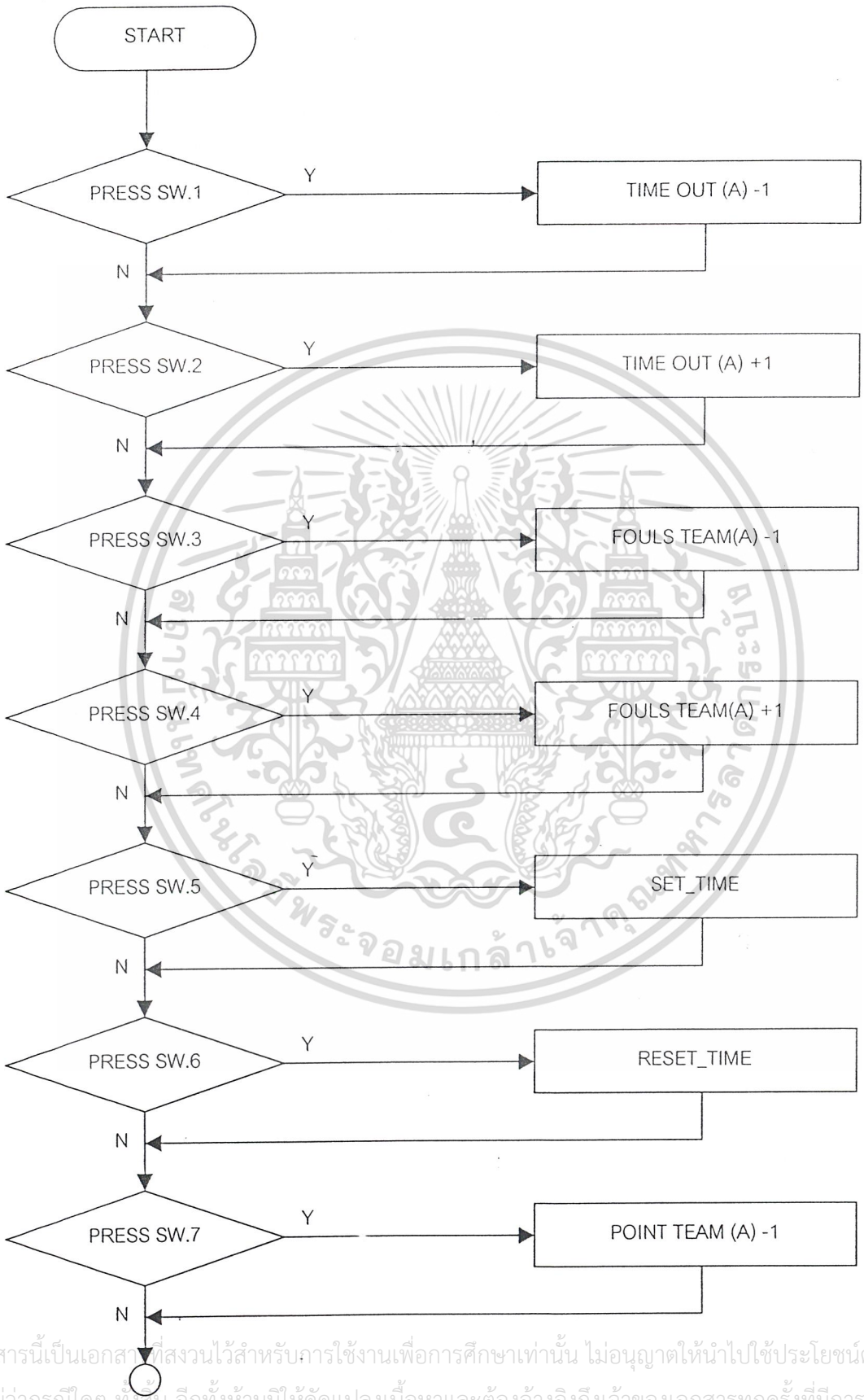


รูปที่ 3-4แสดง Flow Chart ส่วน Main Program

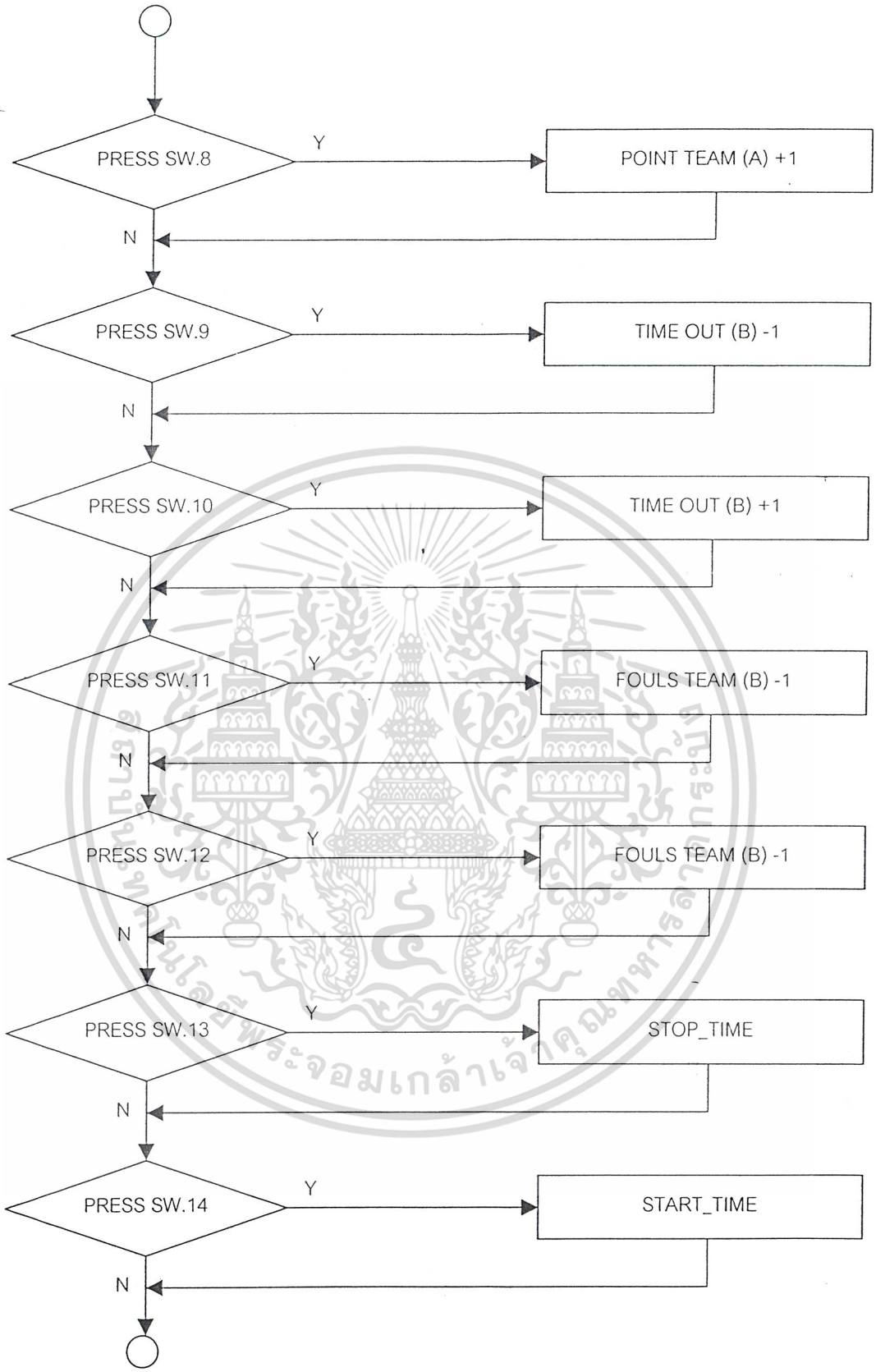
เริ่มต้นการทำงานเป็นส่วนของการกำหนดค่าเริ่มต้นต่างๆให้กับโปรแกรมแล้วต่อไป เป็นการเข้าสู่การควบคุมการทำงานเรียกว่า “Maintainance Mode”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

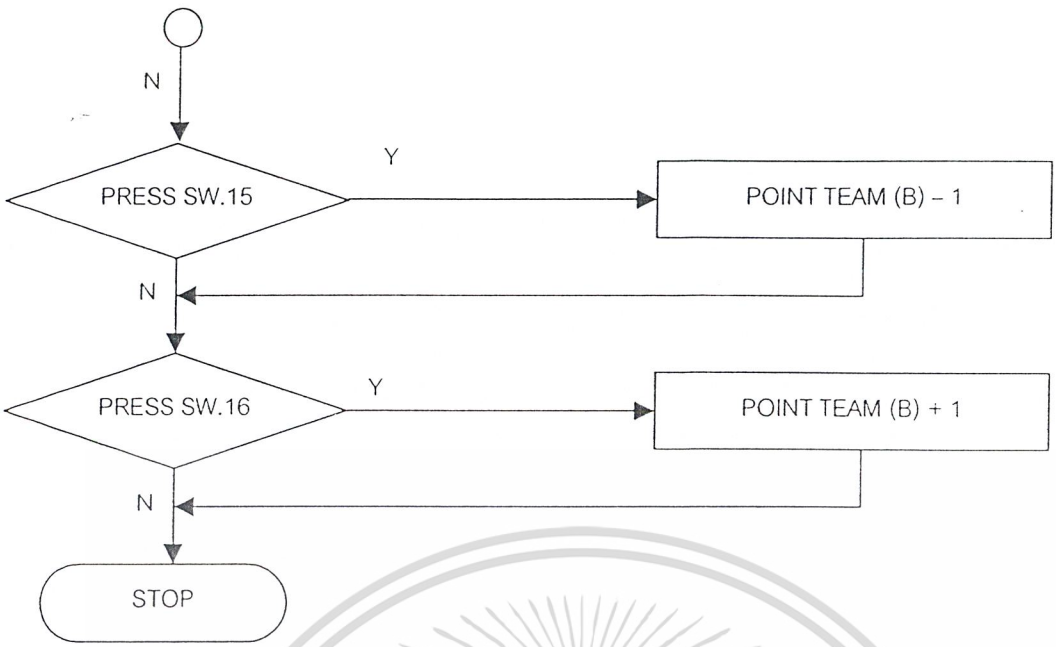
3.2 Maintenance Mode



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3-5 Flow Chart แสดงการทำงานใน Maintenance Mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3-4 สามารถลำดับการทำงาน Maintenance Mode ได้ดังนี้

ลำดับที่	หมายเลขสวิทซ์ที่ตรวจสอบ	ความหมายของสวิทซ์	ผลการทำงานเมื่อเกิดการกดสวิทซ์
1	SW 1	ลบจำนวนการขอเวลานอกของทีม A	TIME OUT (A) -1
2	SW 2	บวกจำนวนการขอเวลานอกของทีม A	TIME OUT (A) +1
3	SW 3	ลบจำนวนการฟาล์วทีมของทีม A	FOULS TEAM (A) -1
4	SW 4	บวกจำนวนการฟาล์วทีมของทีม A	FOULS TEAM (A) +1
5	SW 5	RESET TIME	เคลียร์ค่าเวลาให้เป็นศูนย์
6	SW 6	SET TIME	ตั้งค่าเริ่มต้นสำหรับการนับเวลาถอยหลัง
7	SW 7	ลบคะแนนของทีม A	POINT (A) –1
8	SW 8	บวกคะแนนของทีม A	POINT (A) +1
9	SW 9	ลบจำนวนการขอเวลานอกของทีมB	TIME OUT (B) -1
10	SW 10	บวกจำนวนการขอเวลานอกของทีม B	TIME OUT (B) +1
11	SW 11	ลบจำนวนการฟาล์วทีมของทีม B	FOULS TEAM (B) -1
12	SW 12	บวกจำนวนการฟาล์วทีมของทีม B	FOULS TEAM (B) +1
13	SW 13	STOP TIME	หยุดการนับเวลาถอยหลัง
14	SW 14	START TIME	เริ่มการนับเวลาถอยหลัง
15	SW 15	ลบคะแนนของทีม B	POINT (B) –1
16	SW 16	บวกคะแนนของทีม B	POINT (B) +1

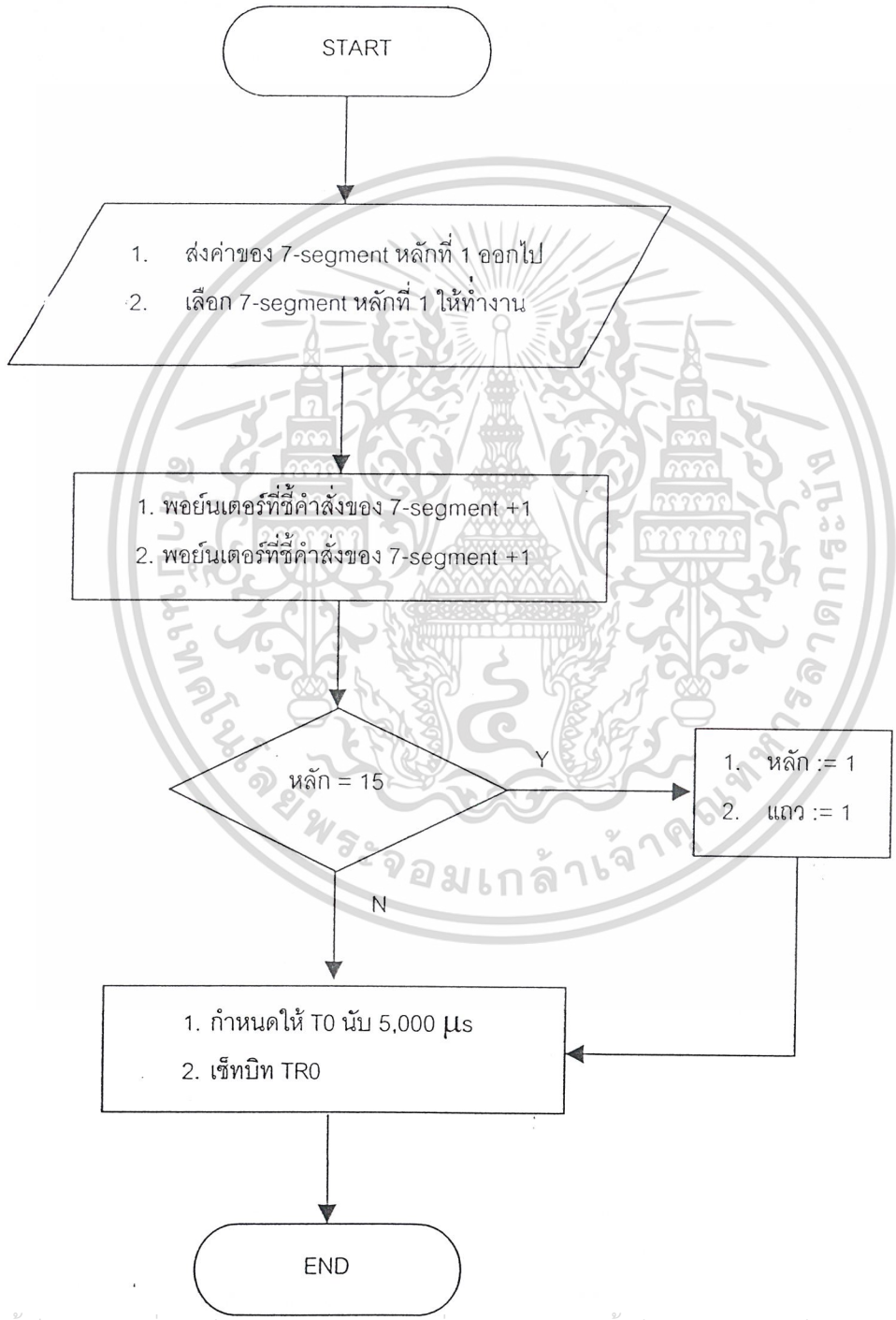
ตาราง 3-3 แสดงลำดับการทำงานของMaintenance Mode

การทำงานของ Maintainace Mode จะมีลักษณะเป็นการทำงานซ้ำ คือเมื่อทำงานตามลำดับจากลำดับที่ 1-16 แล้วจะกลับไปเมื่อเริ่มต้นการทำงานในลำดับที่ 1 ใหม่ไปเรื่อยๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 Interrupt Service Routime Timer 0 (To) : Display Mode

ในส่วนนี้จะเริ่มต้นจากการส่งค่าของ 7-segment หลักที่ 1 ออกไปแล้วเลือกให้ 7-segment หลักที่ 1 ทำงาน และจะเพิ่มค่าพอย์นเตอร์ที่ชี้ค่าของ 7-segment บวก 1 รวมทั้งพอย์นเตอร์ที่เลือกหลัก 7-segment บวก 1 ซึ่งเมื่อแสดงผลครบ 14 หลักแล้ววนไปทำการแสดงผลหลักที่ 1 ใหม่อีกไปเรื่อยๆ โดยจะเว้นระหว่างของการแสดงผลแต่ละหลัก 5000 μs ซึ่งจะทำให้ตาคนเรามองเห็นเหมือน 7-segment ทำงานแสดงผลข้อมูลอยู่ตลอดเวลา

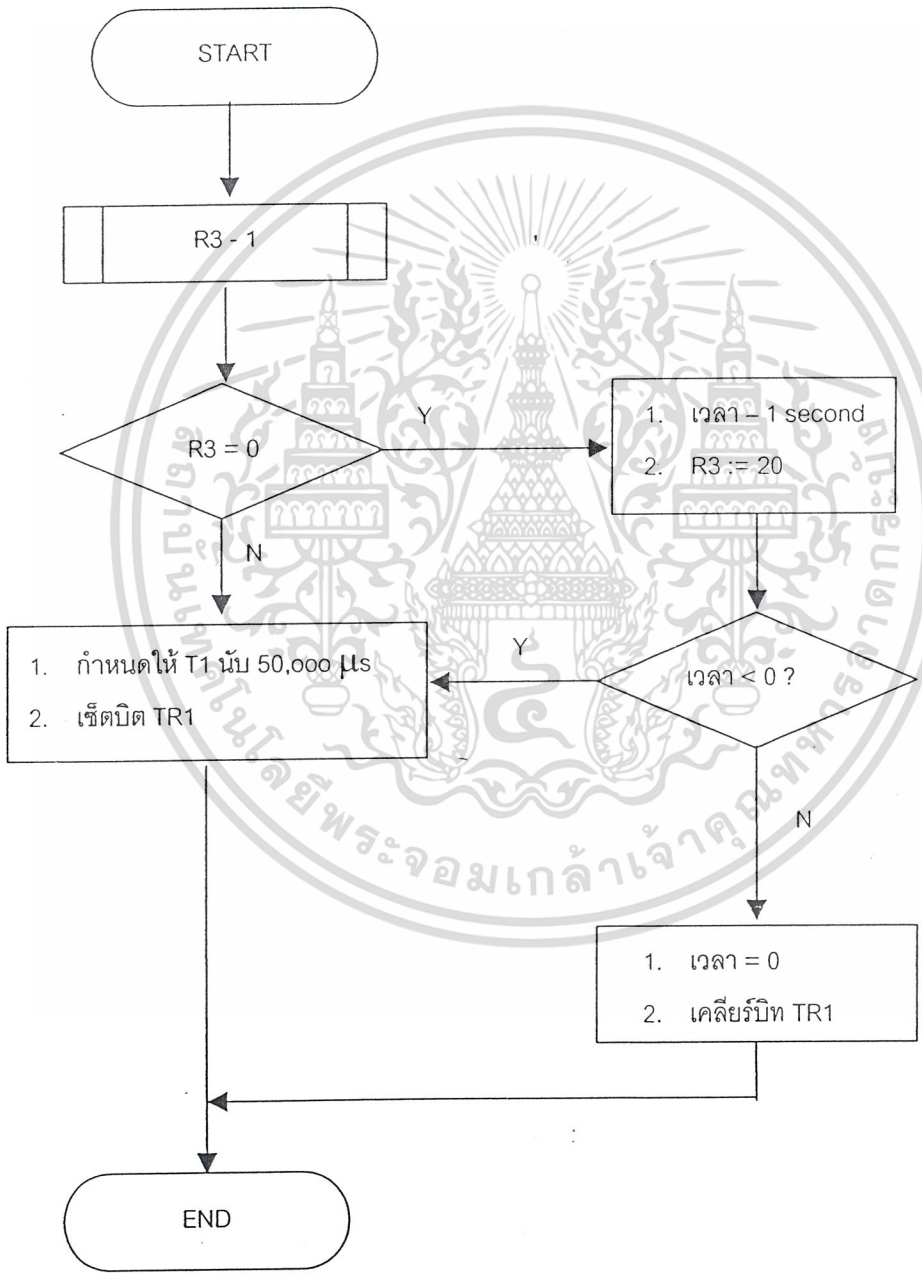


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3-6 Flow Chart แสดงการทำงาน Interrupt Service Routine T0

3.2.4 Interrupt Service Routime Timer 1 (T_1) : Couter 1 SECOND

ในส่วนนี้จะเริ่มต้นจากการลบค่ารีจิสเตอร์ R_3 ซึ่งในส่วนของโปรแกรมหลักนั้น เมื่อเริ่มต้นได้กำหนดค่า 20 ให้กับรีจิสเตอร์ R_3 และในไทม์เมอร์ 1 นี้ ได้กำหนดให้เกิดการอินเตอร์รัปต์เมื่อนับครบ 50000 μs ดังนั้นเมื่อ ไทม์เมอร์ 1 ทำการอินเตอร์รัปต์ครบ 20 ครั้ง ค่าในรีจิสเตอร์ R_3 จะเท่ากับศูนย์ ซึ่งหมายถึงได้นับเวลาครบ 1 วินาที ซึ่งทำการลบค่าเวลาออก 1 วินาที จากนั้นจะกำหนดค่าเริ่มต้น 20 ให้กับรีจิสเตอร์ R_3 ใหม่อีกครั้งหนึ่ง



รูปที่ 3- 7Flow Chart แสดงการทำงาน ของ Interrupt Service Routine T1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทำงานของแป้นแสดงผลการแข่งขันบาสเกตบอลแบบกึ่งอัตโนมัติ

การทำงานของแป้นแสดงผลการแข่งขันบาสเกตบอลแบบกึ่งอัตโนมัตินี้ ได้ออกแบบการทำงานให้ควบคุมผ่านสวิทช์ทั้งหมด 16 สวิทช์ โดยสามารถแสดงตำแหน่งและหน้าที่การทำงานของแต่ละสวิทช์ได้ดังรูปที่ 4-1

4.1 การตรวจสอบระบบก่อนการใช้งาน

โดยปกติเมื่อป้อนไฟเลี้ยงขนาด 220 V ให้กับบอร์ดควบคุมแล้วจะสามารถเริ่มการทำงานได้เลย โดยได้กำหนดค่าเริ่มต้นของข้อมูลการแข่งขันทุกอย่างไว้ที่ศูนย์ ดังนั้นหากพบว่า เมื่อป้อนไฟเลี้ยงให้กับบอร์ดควบคุมแล้ว 7-segment ไม่ได้แสดงข้อมูลเริ่มต้นที่ศูนย์ทุกๆ หลัก ให้ตรวจสอบฮาร์ดแวร์ของบอร์ดควบคุม โดยตรวจสอบประกอบกับวงจรโดยรวมของบอร์ดควบคุม ดังแสดงไว้ในรูปที่ 3-1

4.2 การควบคุมการทำงานเมื่อระบบพร้อม

4.2.1 การควบคุมการทำงานด้านคะแนนการแข่งขัน

เมื่อต้องการควบคุมการทำงานด้านคะแนนการแข่งขัน สามารถที่จะเพิ่มหรือลบคะแนนได้ครั้งละ 1 คะแนนเท่านั้น โดยได้แยกสวิทช์การเพิ่มและลบคะแนนการแข่งขันของแต่ละทีมไว้สองข้างของบอร์ดควบคุม ซึ่งการแสดงผลการทำคะแนนของแต่ละทีมนั้นสามารถแสดงได้ระหว่าง 0-999 คะแนน โดยแสดงผ่าน 7-segment 3 หลัก ดังรูปที่ 4-1

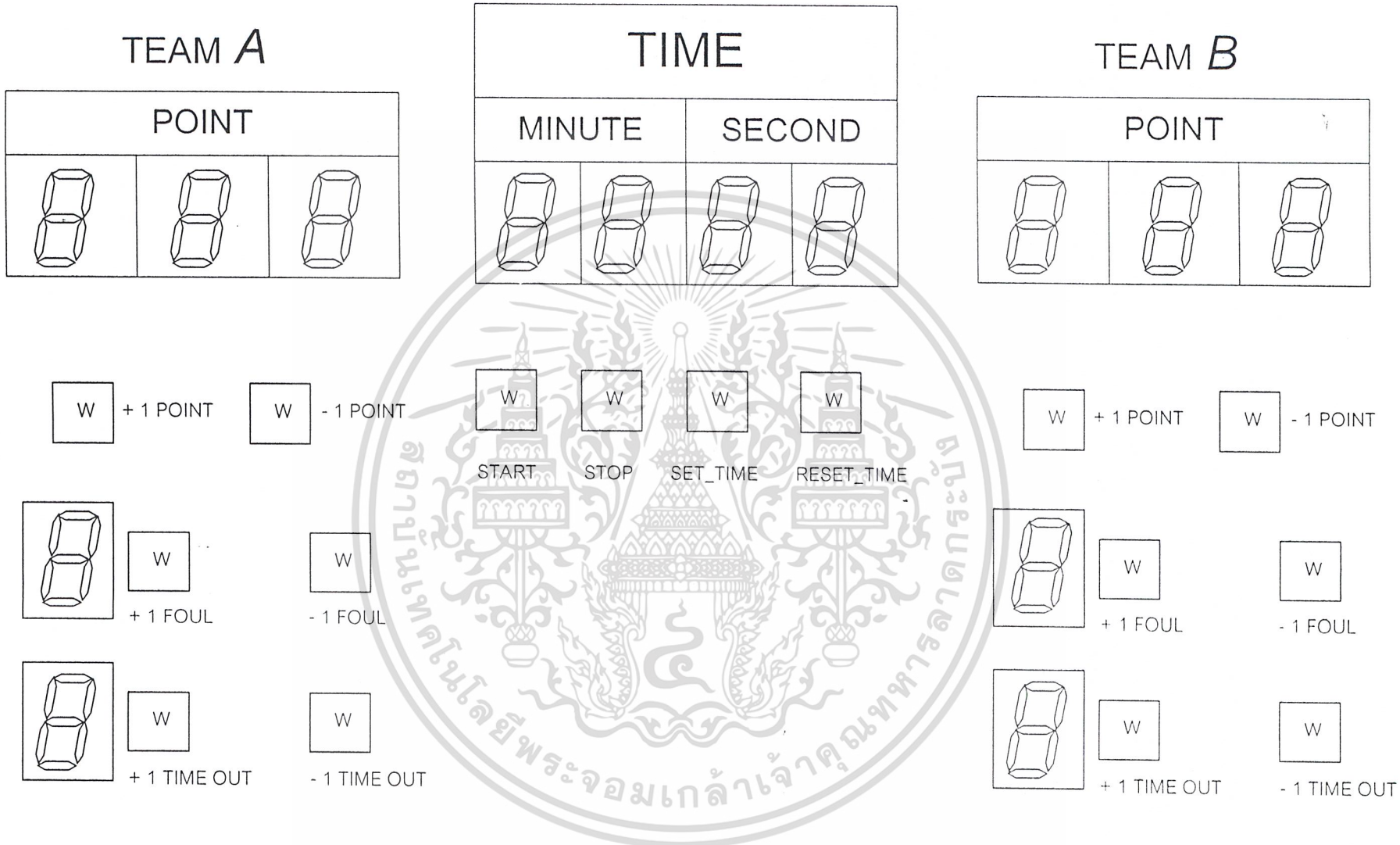
4.2.2 การควบคุมการทำงานด้านฟาล์วรวมของแต่ละทีม

เมื่อต้องการควบคุมการทำงานด้านฟาล์วรวมของแต่ละทีม สามารถที่จะเพิ่มหรือลบฟาล์วได้ครั้งละ 1 ครั้งเท่านั้น โดยได้แยกสวิทช์การเพิ่มและลบฟาล์วรวมของแต่ละทีมไว้สองข้างของบอร์ดควบคุม ซึ่งการแสดงผลฟาล์วรวมของแต่ละทีมนั้นสามารถแสดงได้ระหว่าง 0-7 ครั้ง โดยแสดงผ่าน 7-segment 1 หลัก ดังรูปที่ 4-1

4.2.3 การควบคุมการทำงานด้านการขอเวลานอกของแต่ละทีม

เมื่อต้องการควบคุมการทำงานด้านการขอเวลานอกของแต่ละทีม สามารถที่จะเพิ่มหรือลบการขอเวลานอกได้ครั้งละ 1 ครั้งเท่านั้น โดยได้แยกสวิทช์การเพิ่มและลบการขอเวลานอกของแต่ละทีมไว้สองข้างของบอร์ดควบคุม ซึ่งการแสดงผลการขอเวลานอกของแต่ละทีมนั้นสามารถแสดงได้ระหว่าง 0-2 ครั้ง โดยแสดงผ่าน 7-segment 1 หลัก ดังรูปที่ 4-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-1 แสดงตำแหน่งและชื่อของแต่ละสวิทช์

4.2.4 การควบคุมการทำงานด้านเวลาในการแข่งขัน

เมื่อต้องการควบคุมการทำงานด้านเวลาในการแข่งขัน สามารถที่จะทำได้ผ่านสวิทช์ที่อยู่กลางบอร์ดควบคุมทั้ง 4 สวิทช์ โดยแต่ละสวิทช์มีหน้าที่การทำงานดังนี้

ชื่อสวิทช์	หน้าที่การทำงาน
SET_TIME	ใช้ในการตั้งเวลาในการแข่งขันโดยได้กำหนดไว้ให้สามารถตั้งเวลาในการแข่งขันได้ทั้งหมด 4 ค่าคือ 20,15,10 และ 5 นาที โดยทุกครั้งที่มีการกดสวิทช์จะมีการเปลี่ยนแปลงค่าจาก 20-5 นาทีตามลำดับ (หมายเหตุ:- การกดสวิทช์จะมีผลต่อการทำงานเมื่อค่าเริ่มต้นของเวลาในขณะนั้นเป็น 0 นาทีเท่านั้น)
RESET_TIME	ใช้ในการเคลียร์ค่าเวลาในการแข่งขันให้เป็น 0 นาที การกดสวิทช์จะมีผลต่อการทำงานเมื่อเวลาในการแข่งขันหยุดเดินและได้กดสวิทช์ค้างไว้ชั่วขณะหนึ่งเท่านั้น
START_TIME	ใช้ในการเริ่มนับเวลาในการแข่งขัน ซึ่งสามารถนับถอยหลังได้ถึง 0 นาที
STOP_TIME	ใช้ในการหยุดนับเวลาในการแข่งขัน

ตารางที่ 4-1 แสดงชื่อและหน้าที่การทำงานของสวิทช์ในการจับเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

ผลการทดลอง

5.1 การทดลองใช้งานแป้นแสดงผลการแข่งขันบาสเกตบอลแบบกึ่งอัตโนมัติ

การทดลองใช้งานแป้นแสดงผลการแข่งขันบาสเกตบอลจะแบ่งออกตามชนิดของข้อมูลการแข่งขันที่นำเสนอ สามารถแบ่งได้ 4 ชนิดคือ

5.1.1 การทดลองควบคุมทางด้านเวลาในการแข่งขัน

- เมื่อระบบพร้อมที่จะทำงาน จะสามารถกำหนดเวลาในการแข่งขันได้ทั้งหมด 4 ค่า คือ 20,15,10 และ 5 นาทีตามลำดับ โดยควบคุมการตั้งค่าเวลาการแข่งขันผ่านทางสวิทช์ SET_TIME ซึ่งการกดสวิทช์ SET_TIME จะมีผลต่อการทำงานตั้งค่าเวลา ก็ต่อเมื่อเวลาในการแข่งขันขณะนั้นเท่ากับ 0 นาที
- ในระหว่างที่กำลังทำงานนับเวลาในการแข่งขันนั้นจะไม่สามารถเคลียร์ค่าเวลาในการแข่งขันได้ ซึ่งถ้าหากต้องการเคลียร์ค่าเวลาในการแข่งขันจะต้องกดหยุดการทำงานนับเวลาก่อน
- การกดเคลียร์ค่าเวลาในการแข่งขันจะต้องกดสวิทช์ RESET_TIME ค้างไว้ชั่วขณะหนึ่ง จึงจะมีผลทำให้เกิดการทำงานเคลียร์ค่าเวลาในการแข่งขัน

5.1.2 การทดลองควบคุมทางด้านคะแนนการแข่งขัน

- เมื่อระบบพร้อมที่จะทำงานจะกำหนดค่าเริ่มต้นในการนับคะแนนให้เป็น 0 คะแนนทั้ง 2 ทีม
- การนับคะแนนขึ้นลงสามารถทำได้ครั้งละ 1 คะแนนเท่านั้น
- สามารถนับคะแนนได้ 0-999 คะแนนต่อทีม

5.1.3 การทดลองควบคุมทางด้านการทำฟาล์วรวม

- เมื่อระบบพร้อมที่จะทำงานจะกำหนดค่าเริ่มต้นในการทำฟาล์วรวมของแต่ละทีมให้เป็น 0 ครั้ง
- การนับฟาล์วรวมสามารถนับได้ทีละ 1 ครั้งเท่านั้น
- สามารถนับฟาล์วรวมได้ระหว่าง 0-7 ครั้งต่อทีม

5.1.4 การทดลองควบคุมทางด้านการขอเวลานอก

- เมื่อระบบพร้อมที่จะทำงานจะกำหนดค่าเริ่มต้นในการขอเวลานอกของแต่ละทีมให้เป็น 0 ครั้ง
- การนับจำนวนการขอเวลานอกสามารถนับได้ทีละ 1 ครั้งเท่านั้น
- สามารถนับจำนวนการขอเวลานอกได้ระหว่าง 0-2 ครั้งต่อทีม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ปัญหาที่พบ

จากการทดลองใช้งานแป้นแสดงผลการแข่งขันบาสเกตบอลแบบกึ่งอัตโนมัตินี้ พบว่ายังไม่สามารถใช้งานได้อย่างสมบูรณ์แบบในการแข่งขันจริงซึ่งการที่จะสร้างแป้นแสดงผลให้สามารถใช้งานได้จริงในการแข่งขันนั้น บอร์ดแสดงข้อมูลจะต้องเป็น 7-segment ขนาด 4 x 7 นิ้ว และในการแสดงผลตัวเลข 1 ตัว ต้องใช้ LED ถึง 140 ตัว ซึ่งอุปกรณ์ที่จำเป็นขณะนี้มีราคาสูง จึงเกิดปัญหาทางด้านงบประมาณ หากจะทำให้สมบูรณ์จะทำให้ต้นทุนสูงตามไปด้วย

5.3 แนวทางการพัฒนา

เนื่องจากระบบของแป้นแสดงผลการแข่งขันบาสเกตบอลแบบกึ่งอัตโนมัตินี้ ยังไม่สมบูรณ์พอจึงต้องมีการเพิ่มเติมอุปกรณ์บางส่วนเพื่อการพัฒนาต่อไป ซึ่งพอจะมีแนวทางดังนี้

5.3.1 การพัฒนาเพิ่มการนับฟาว์ลบุคคล เนื่องจากระบบนี้ยังขาดการนับจำนวนครั้งในการฟาว์ลของนักกีฬาแต่ละคน ซึ่งตามกติกาการแข่งขันบาสเกตบอล ในการแข่งขันแต่ละครั้งนักกีฬาสามารถฟาว์ลได้ 5 ครั้ง หากเกินจากนี้จะหมดสิทธิ์ในการเล่นในการแข่งขันนั้นๆ ซึ่งการเพิ่มการนับฟาว์ลบุคคลนี้ต้องเพิ่มอุปกรณ์แสดงผล สำหรับนักกีฬาทุกคนทั้งตัวจริงและสำรองของแต่ละทีม

5.3.2 การพัฒนาเพิ่มการแสดงข้อมูลฟาว์ลเทคนิค เนื่องจากแป้นแสดงนี้ยังขาดการแสดงผลฟาว์ลเทคนิค ซึ่งต้องเพิ่มอุปกรณ์แสดงผลฟาว์ลเทคนิคของทั้งผู้เล่นและโค้ช โดยอาจจะแยกสีหลอดไฟแสดงผลให้ต่างกับการฟาว์ลบุคคล เพื่อให้สังเกตได้ง่าย

5.3.3 การพัฒนาเพิ่มอุปกรณ์จับเวลา 30 วินาที ตามกติกาจะต้องจับเวลาการบุกของผู้เล่นฝ่ายบุกไม่ให้เกิน 30 วินาที ซึ่งตามกติกาบาสเกตบอลสมัครเล่นของ FIBA ที่แก้ไขใหม่ (1998-2002) กำหนดให้มีอุปกรณ์จับเวลา 30 วินาที เป็นอุปกรณ์มาตรฐานจำนวน 2 อัน การเพิ่มอุปกรณ์นี้ต้องเพิ่มส่วนแสดงผล และสามารถทำงานร่วมกับบอร์ดควบคุมได้

5.3.4 การพัฒนาเพิ่มการแสดงชื่อทีมการแข่งขัน จะแสดงผลทางบอร์ดแสดงผล ให้แสดงชื่อทีมการแข่งขันได้นี้ต้องมีการเพิ่ม LED อีกหลายๆ ดวง และปรับปรุงส่วนควบคุมให้แสดงผลออกมาเป็นชื่อทีมของทั้ง 2 ทีมที่ทำการแข่งขันได้

5.3.5 การเพิ่มอุปกรณ์การพิมพ์ผลการแข่งขัน เนื่องจากหลังการแข่งขันเสร็จสิ้น ควรมีการเก็บรวบรวมข้อมูลการแข่งขันไว้เป็นหลักฐาน ดังนั้นจึงควรเพิ่มอุปกรณ์การพิมพ์ผลการแข่งขัน, คะแนนของแต่ละทีม,จำนวนฟาล์วของแต่ละบุคคล และข้อมูลอื่นๆ เพื่อเป็นหลักฐานของกรรมการจัดการแข่งขันและผู้เล่นของทั้ง 2 ทีม

5.3.6 การเพิ่มแป้นแสดงผลที่ประกอบด้วย 7-segment ขนาดใหญ่ เพื่อให้นักกีฬารวมถึงคนดูสามารถรับทราบผลการแข่งขันที่ชัดเจน ทำให้เกมการแข่งขันน่าตื่นเต้นและต่อเนื่องขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4 บทสรุป

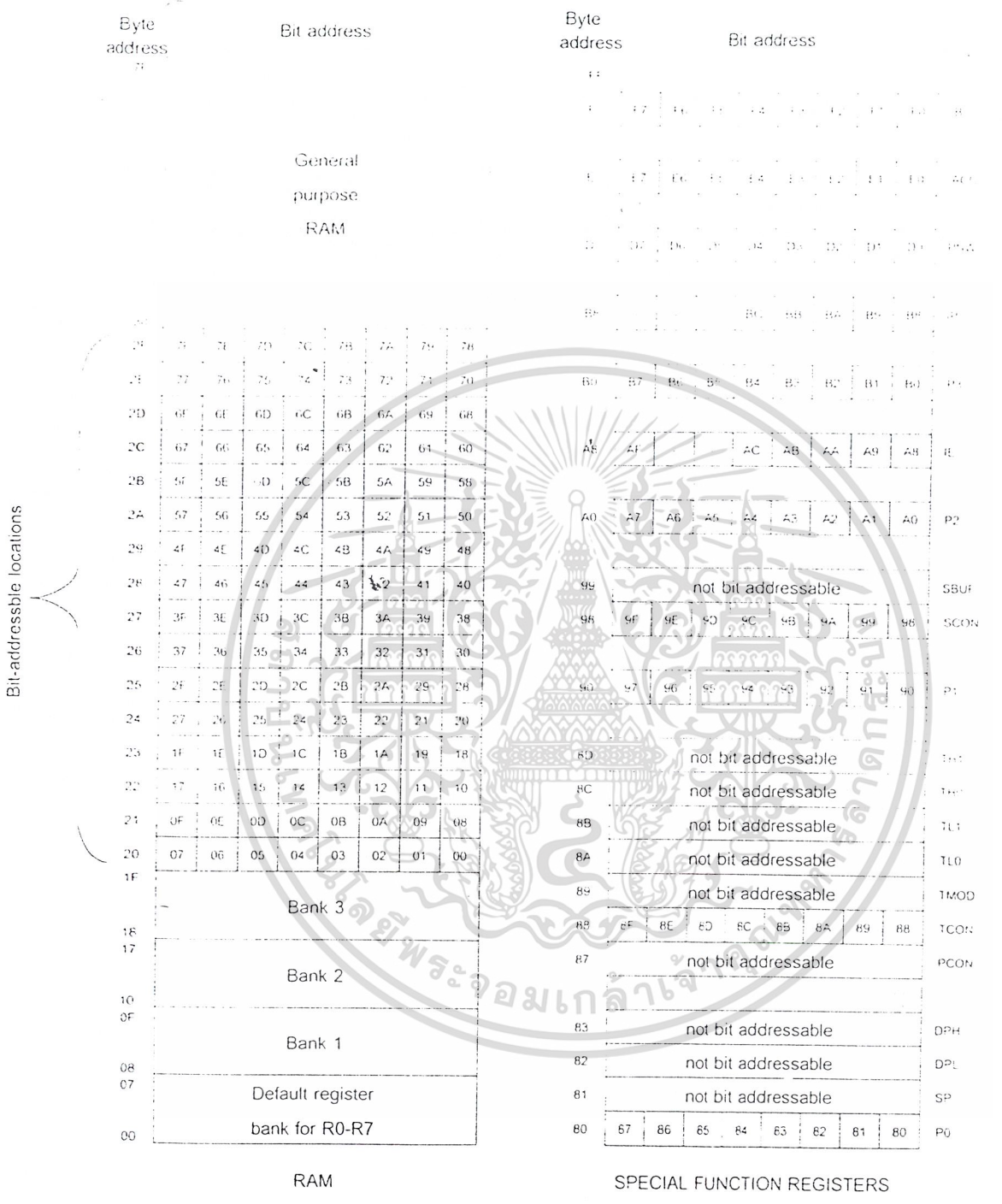
แป้นแสดงผลการแข่งขันบาสเกตบอลแบบกึ่งอัตโนมัตินี้ สามารถใช้งานได้ในการแสดงข้อมูลผลการแข่งขัน ซึ่งได้แก่ เวลาในการแข่งขัน,คะแนนของแต่ละทีม,จำนวนฟาว์ลรวมแต่ละทีม และจำนวนการขอเวลานอกของแต่ละทีม ซึ่งควบคุมการทำงานทั้งหมดผ่านทางบอร์ดควบคุม เพื่อส่งสัญญาณให้ไมโครคอนโทรลเลอร์ประมวลผลและควบคุมการสร้างสัญญาณให้กับส่วนแสดงผล โดยส่วนแสดงผลจะรับสัญญาณจากไมโครคอนโทรลเลอร์ เพื่อนำไปแสดงผลข้อมูลในการแข่งขันผ่านทาง 7-segment

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางแสดงตำแหน่งของรีจีสเตอร์ฟังก์ชั่นพิเศษ (SFR's)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1. MCS®-51 Instruction Set Description

ARITHMETIC OPERATIONS

Mnemonic		Description	Byte	Cyc
ADD	A,Rn	Add register to Accumulator	1	1
ADD	A,direct	Add direct byte to Accumulator	2	1
ADD	A,@Ri	Add indirect RAM to Accumulator	1	1
ADD	A,#data	Add immediate data to Accumulator	2	1
ADDC	A,Rn	Add register to Accumulator with Carry	1	1
ADDC	A,direct	Add direct byte to A with Carry flag	2	1
ADDC	A,@Ri	Add indirect RAM to A with Carry flag	1	1
ADDC	A,#data	Add immediate data to A with Carry flag	2	1
SUBB	A,Rn	Subtract register from A with Borrow	1	1
SUBB	A,direct	Subtract direct byte from A with Borrow	2	1
SUBB	A,@Ri	Subtract indirect RAM from A with Borrow	1	1
SUBB	A,#data	Subtract immed. data from A with Borrow	2	1
INC	A	Increment Accumulator	1	1
INC	Rn	Increment register	1	1
INC	direct	Increment direct byte	2	1
INC	@Ri	Incriment indirect RAM	1	1
INC	DPTR	Incriment Data Pointer	1	2
DEC	A	Decrement Accumulator	1	1
DEC	Rn	Decrement register	1	1
DEC	direct	Decrement direct byte	2	1
DEC	@Ri	Decrement indirect RAM	1	1
MUL	AB	Multiply A & B	1	4
DIV	AB	Divide A by B	1	4
DA	A	Decimal Adjust Accumulator	1	1

LOGICAL OPERATIONS

Mnemonic		Destination	Byte	Cyc
ANL	A,Rn	AND register to Accumulator	1	1
ANL	A,direct	AND direct byte to Accumulator	2	1
ANL	A,@Ri	AND indirect RAM to Accumulator	1	1
ANL	A,#data	AND immediate data to Accumulator	2	1
ANL	direct,A	AND Accumulator to direct byte	2	1
ANL	direct,#data	AND immediate data to direct byte	3	2
ORL	A,Rn	OR register to Accumulator	1	1
ORL	A,direct	OR direct byte to Accumulator	2	1
ORL	A,@Ri	OR indirect RAM to Accumulator	1	1
ORL	A,#data	OR immediate data to Accumulator	2	1
ORL	direct,A	OR Accumulator to direct byte	2	1
ORL	direct,#data	OR immediate data to direct byte	3	2
XRL	A,Rn	Exclusive-OR register to Accumulator	1	1
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	1
XRL	A,@Ri	Exclusive-OR indirect RAM to A	1	1
XRL	A,#data	Exclusive-OR immediate data to A	2	1
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	1
XRL	direct,#data	Exclusive-OR immediate data to direct	3	2
CLR	A	Clear Accumulator	1	1
CPL	A	Complement Accumulator	1	1
RL	A	Rotate Accumulator Left	1	1
RLC	A	Rotate A Left through the Carry flag	1	1
RR	A	Rotate Accumulator Right	1	1
RRC	A	Rotate A Right through Carry flag	1	1
SWAP	A	Swap nibbles within the Accumulator	1	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1. (Cont.)

DATA TRANSFER

Mnemonic		Description	Byte	Cyc
MOV	A,Rn	Move register to Accumulator	1	1
MOV	A,direct	Move direct byte to Accumulator	2	1
MOV	A,@Ri	Move indirect RAM to Accumulator	1	1
MOV	A,#data	Move immediate data to Accumulator	2	1
MOV	Rn,A	Move Accumulator to register	1	1
MOV	Rn,direct	Move direct byte to register	2	2
MOV	Rn,#data	Move immediate data to register	2	1
MOV	direct,A	Move Accumulator to direct byte	2	1
MOV	direct,Rn	Move register to direct byte	2	2
MOV	direct,direct	Move direct byte to direct	3	2
MOV	direct,@Ri	Move indirect RAM to direct byte	2	2
MOV	direct,#data	Move immediate data to direct byte	3	2
MOV	@Ri,A	Move Accumulator to indirect RAM	1	1
MOV	@Ri,direct	Move direct byte to indirect RAM	2	2
MOV	@Ri,#data	Move immediate data to indirect RAM	2	1
MOV	DPTR,#data 16	Load Data Pointer with a 16-bit constant	3	2
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to A	1	2
MOVC	A,@A+PC	Move Code byte relative to PC to A	1	2
MOVX	A,@Ri	Move External RAM (8-bit addr) to A	1	2
MOVX	A,@DPTR	Move External RAM (16-bit addr) to A	1	2
MOVX	@Ri,A	Move A to External RAM (8-bit addr)	1	2
MOVX	@DPTR,A	Move A to External RAM (16-bit addr)	1	2
PUSH	direct	Push direct byte onto stack	2	2
POP	direct	Pop direct byte form stack	2	2
XCH	A,Rn	Exchange register with Accumulator	1	1
XCH	A,direct	Exchange direct byte with Accumulator	2	1
XCH	A,@Ri	Exchange indirect RAM with A	1	1
XCHD	A,@Ri	Exchange low-order nibble ind RAM with A	1	1

BOOLEAN VARIABLE MANIPULATION

Mnemonic		Description	Byte	Cyc
CLR	C	Clear Carry flag	1	1
CLR	bit	Clear direct bit	2	1
SETB	C	Set Carry flag	1	1
SETB	bit	Set direct Bit	2	1
CPL	C	Complement Carry flag	1	1
CPL	bit	Complement direct bit	2	1
ANL	C,bit	AND direct bit to Carry flag	2	2
ANL	C,/ bit	AND complement of direct bit to Carry	2	2
ORL	C,bit	OR direct bit to Carry flag	2	2
ORL	C,/ bit	OR complement of direct bit to Carry	2	2
MOV	C,bit	Move direct bit to Carry flag	2	1
MOV	bit,C	Move Carry flag to direct bit	2	2

PROGRAM AND MACHINE CONTROL

Mnemonic		Description	Byte	Cyc
ACALL	addr 11	Absolute Subroutine Call	2	2
LCALL	addr 16	Long Subroutine Call	3	2
RET		Return from subroutine	1	2
RETI		Return from interrupt	1	2
AJMP	addr 11	Absolute Jump	2	2
LJMP	addr 16	Long Jump	3	2
SJMP	rel	Short Jump (relative addr)	2	2
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	2
JZ	rel	Jump if Accumulator is Zero	2	2
JNZ	rel	Jump if Accumulator is Not Zero	2	2
JC	rel	Jump if Carry flag is set	2	2
JNC	rel	Jump if No Carry flag	2	2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 1. (Cont.)

PROGRAM AND MACHINE CONTROL (cont.)

Mnemonic		Description	Byte	Cyc
JB	bit,rel	Jump if direct Bit set	3	2
JNB	bit,rel	Jump if direct Bit Not set	3	2
JBC	bit,rel	Jump if direct Bit is set & Clear bit	3	2
CJNE	A,direct,rel	Compare direct to A & Jump if Not Equal	3	2
CJNE	A,#data,rel	Comp. immed. to A & Jump if Not Equal	3	2
CJNE	Rn,#data,rel	Comp. immed. to reg & Jump if Not Equal	3	2
CJNE	@Ri,#data,rel	Comp. immed. to ind. & Jump if Not Equal	3	2
DJNZ	Rn,rel	Decrement register & Jump if Not Zero	2	2
DJNZ	direct,rel	Decrement direct & Jump if Not Zero	3	2
NOP		No operation	1	1

Notes on data addressing modes:

Rn	– Working register R0-R7
direct	– 128 internal RAM locations, any I/O port, control or status register
@Ri	– Indirect internal RAM location addressed by register R0 or R1
#data	– 8-bit constant included in instruction
#data 16	– 16-bit constant included as bytes 2 & 3 of instruction
bit	– 128 software flags, any I/O pin, control or status bit

Notes on program addressing modes:

addr 16	– Destination address for LCALL & LJMP may be anywhere within the 64-k program memory address space
Addr 11	– Destination address for ACALL & AJMP will be within the same 2-k page of program memory as the first byte of the following instruction
rel	– SJMP and all conditional jumps include an 8-bit offset byte. Range is +127 –128 bytes relative to first byte of the following instruction

All mnemonics copyrighted © Intel Corporation 1979

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 2. Instruction Opcodes in Hexadecimal Order

Hex Code	Number of Bytes	Mnemonic	Operands
00	1	NOP	
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	bit addr,code addr
11	2	ACALL	code addr
12	3	LCALL	code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	data addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	bit addr,code addr
21	2	AJMP	code addr
22	1	RET	
23	1	RL	A
24	2	ADD	A,data
25	2	ADD	A,data addr
26	1	ADD	A,@R0
27	1	ADD	A,@R1
28	1	ADD	A,R0
29	1	ADD	A,R1
2A	1	ADD	A,R2
2B	1	ADD	A,R3
2C	1	ADD	A,R4
2D	1	ADD	A,R5
2E	1	ADD	A,R6
2F	1	ADD	A,R7
30	3	JNB	bit addr,code addr
31	2	ACALL	code addr
32	1	RETI	

Hex Code	Number of Bytes	Mnemonic	Operands
33	1	RLC	A
34	2	ADDC	A,#data
35	2	ADDC	A,data addr
36	1	ADDC	A,@R0
37	1	ADDC	A,@R1
38	1	ADDC	A,R0
39	1	ADDC	A,R1
3A	1	ADDC	A,R2
3B	1	ADDC	A,R3
3C	1	ADDC	A,R4
3D	1	ADDC	A,R5
3E	1	ADDC	A,R6
3F	1	ADDC	A,R7
40	2	JC	code addr
41	2	AJMP	code addr
42	2	ORL	data addr,A
43	3	ORL	data addr,#data
44	2	ORL	A,#data
45	2	ORL	A,data addr
46	1	ORL	A,@R0
47	1	ORL	A,@R1
48	1	ORL	A,R0
49	1	ORL	A,R1
4A	1	ORL	A,R2
4B	1	ORL	A,R3
4C	1	ORL	A,R4
4D	1	ORL	A,R5
4E	1	ORL	A,R6
4F	1	ORL	A,R7
50	2	JNC	code addr
51	2	ACALL	code addr
52	2	ANL	data addr,A
53	3	ANL	data addr,#data
54	2	ANL	A,#data
55	2	ANL	A,data addr
56	1	ANL	A,@R0
57	1	ANL	A,@R1
58	1	ANL	A,R0
59	1	ANL	A,R1
5A	1	ANL	A,R2
5B	1	ANL	A,R3
5C	1	ANL	A,R4
5D	1	ANL	A,R5
5E	1	ANL	A,R6
5F	1	ANL	A,R7
60	2	JZ	code addr
61	2	AJMP	code addr
62	2	XRL	data addr A
63	3	XRL	data addr,#data
64	2	XRL	A,#data
65	2	XRL	A,data addr

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 2. (Cont.)

Hex Code	Number of Bytes	Mnemonic	Operands
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C,bit addr
73	1	JMP	@A+DPTR
74	2	MOV	A,#data
75	3	MOV	data addr,#data
76	2	MOV	@R0,#data
77	2	MOV	@R1,#data
78	2	MOV	R0,#data
79	2	MOV	R1,#data
7A	2	MOV	R2,#data
7B	2	MOV	R3,#data
7C	2	MOV	R4,#data
7D	2	MOV	R5,#data
7E	2	MOV	R6,#data
7F	2	MOV	R7,#data
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C,bit addr
83	1	MOVC	A,@A+PC
84	1	DIV	AB
85	3	MOV	data addr,data addr
86	2	MOV	data addr,@R0
87	2	MOV	data addr,@R1
88	2	MOV	data addr,R0
89	2	MOV	data addr,R1
8A	2	MOV	data addr,R2
8B	2	MOV	data addr,R3
8C	2	MOV	data addr,R4
8D	2	MOV	data addr,R5
8E	2	MOV	data addr,R6
8F	2	MOV	data addr,R7
90	3	MOV	DPTR,#data
91	2	ACALL	code addr
92	2	MOV	bit addr,C
93	1	MOVC	A,@A+DPTR
94	2	SUBB	A,#data
95	2	SUBB	A,data addr
96	1	SUBB	A,@R0
97	1	SUBB	A,@R1
98	1	SUBB	A,R0

Hex Code	Number of Bytes	Mnemonic	Operands
99	1	SUBB	A,R1
9A	1	SUBB	A,R2
9B	1	SUBB	A,R3
9C	1	SUBB	A,R4
9D	1	SUBB	A,R5
9E	1	SUBB	A,R6
9F	1	SUBB	A,R7
A0	2	ORL	C,bit addr
A1	2	AJMP	code addr
A2	2	MOV	C,bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5	1	reserved	
A6	2	MOV	@R0,data addr
A7	2	MOV	@R1,data addr
A8	2	MOV	R0,data addr
A9	2	MOV	R1,data addr
AA	2	MOV	R2,data addr
AB	2	MOV	R3,data addr
AC	2	MOV	R4,data addr
AD	2	MOV	R5,data addr
AE	2	MOV	R6,data addr
AF	2	MOV	R7,data addr
B0	2	ANL	C,bit addr
B1	2	ACALL	code addr
B2	2	CPL	bit addr
B3	1	CPL	C
B4	3	CJNE	A,#data,code addr
B5	3	CJNE	A,data addr,code addr
B6	3	CJNE	@R0,#data,code addr
B7	3	CJNE	@R1,#data,code addr
B8	3	CJNE	R0,#data,code addr
B9	3	CJNE	R1,#data,code addr
BA	3	CJNE	R2,#data,code addr
BB	3	CJNE	R3,#data,code addr
BC	3	CJNE	R4,#data,code addr
BD	3	CJNE	R5,#data,code addr
BE	3	CJNE	R6,#data,code addr
BF	3	CJNE	R7,#data,code addr
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A,data addr
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1
C8	1	XCH	A,R0
C9	1	XCH	A,R1
CA	1	XCH	A,R2
CB	1	XCH	A,R3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

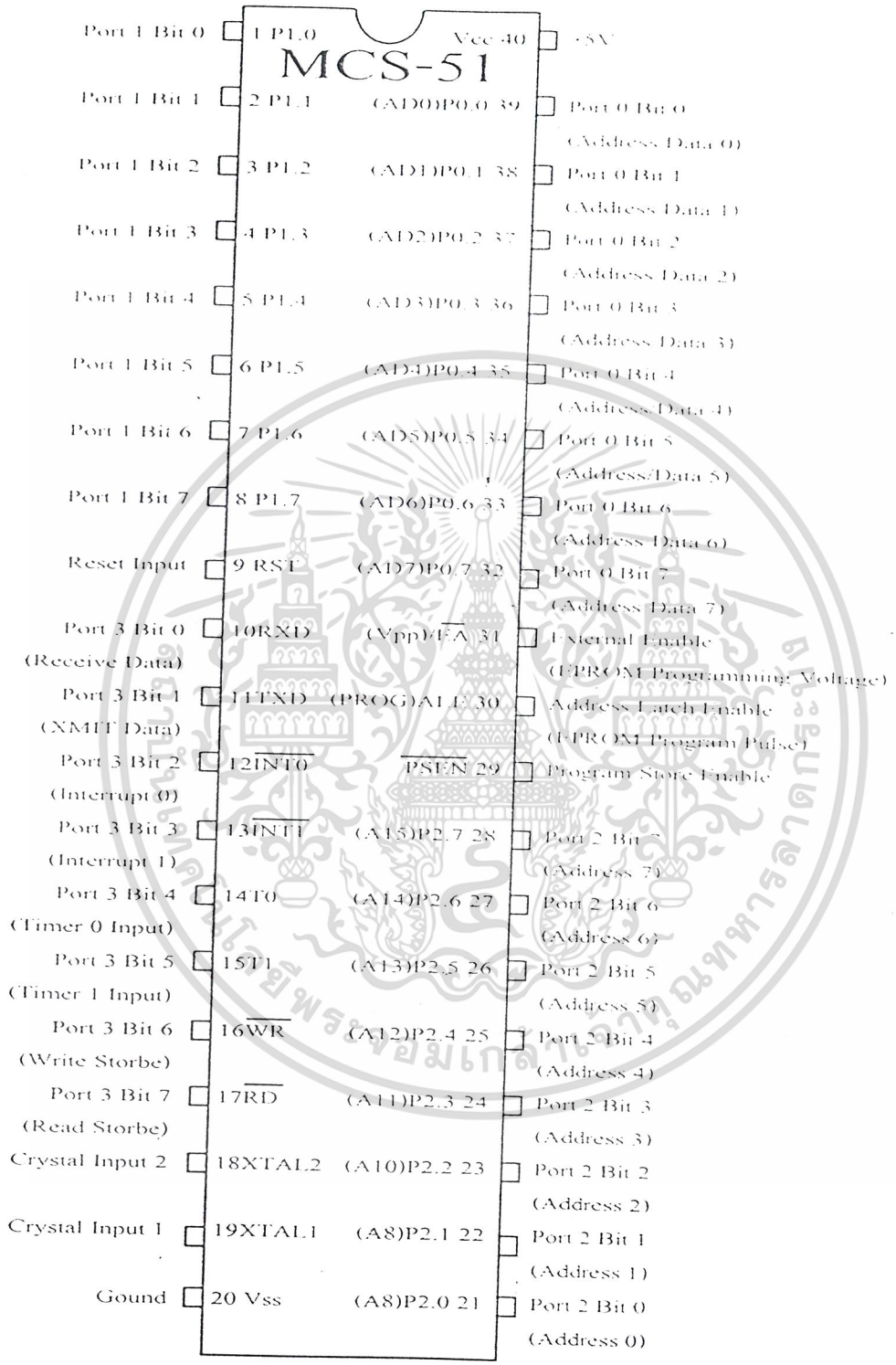
Table 2. (Cont.)

Hex Code	Number of Bytes	Mnemonic	Operands
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1
C8	1	XCH	A,R0
C9	1	XCH	A,R1
CA	1	XCH	A,R2
CB	1	XCH	A,R3
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr,code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr
E0	1	MOVX	A,@DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr

Hex Code	Number of Bytes	Mnemonic	Operands
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR,A
F1	2	ACALL	code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไดอะแกรมขา ของ 8051 แบบ DIP



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลงได้ต้องขอขอบคุณผู้ให้การช่วยเหลือทุกท่านโดยเฉพาะอาจารย์ประสาร ดังติสานนท์ ที่ให้การสนับสนุนทางด้านงบประมาณ ข้อเสนอแนะและข้อมูลในการทำโครงงาน

ขอขอบคุณ เพื่อนๆ พี่ๆ น้องๆ ที่เป็นกำลังใจและได้ให้คำแนะนำปรึกษาและที่ได้ให้ความช่วยเหลือ ตลอดจนเอกสารที่ใช้ประกอบการทำโครงงานนี้

ขอขอบคุณเจ้าหน้าที่และอาจารย์ประจำภาควิชาวิศวกรรมคอมพิวเตอร์ที่ให้คำแนะนำ และความสะดวกในการยืมอุปกรณ์และคู่มือที่ใช้ทำโครงงานนี้

สุดท้ายขอบคุณพ่อแม่ที่ได้ให้กำลังใจและกำลังทรัพย์ในการเรียนเมื่อมีปัญหาและความรักในการอบรมเลี้ยงดูจนมีวันนี้

ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. KENNETH J. AYALA "The 8051 Microcontroller Architecture Programming And Application" West Publishing Company.1991
2. Microcontroller Reference Manual "SILA RESEARCH"
3. สุนทร วิทูสุรพจน์ "การโปรแกรมภาษาแอสแซมบลี้" บริษัทซีเอ็ดยูเคชั่น จำกัด (มหาชน)
4. คู่มือ "LAB MICROPROCESSOR – BASED SYSTEM DESIGN (EECP 0215) MCS-8051" ห้องปฏิบัติการไมโครโปรเซสเซอร์ ภาควิชาวิศวกรรมไฟฟ้า มหาวิทยาลัยเทคโนโลยีมหานคร
5. กติกาบาสเกตบอลเปลี่ยนแปลง ฉบับ 1998-2002 โดยคณะกรรมการผู้ตัดสินบาสเกตบอล สมาคมบาสเกตบอลแห่งประเทศไทยในพระบรมราชูปถัมภ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้