สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

วงจรจัดการหน่วยความจำสำหรับแสดงผลสัญญาณบนจอซีอาร์ที

MEMORY MANAGEMENT CIRCUIT FOR DISPLAY ON CRT MONITOR

โดย

นายพีรพล พัฒนวิจิตร

นายวินัย แก้วโกวิท

เลขหมู่.....
เลขทะเบียน 61724
วัน,เดือน,ปี 21 ก.ค. 2549

b 15520933
i

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2547

ภาควิชา วิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง วงจรจัดการหน่วยความจำสำหรับแสดงผลสัญญาณบนจอซีอาร์ที

MEMORY MANAGEMENT CIRCUIT DISPLAY ON CRT MONITOR

ผู้จัดทำ 1. นายพีรพล พัฒนวิจิตร รหัสประจำตัว 44010341

2. นายวินัย แก้วโกวิท รหัสประจำตัว 44010453

..... อาจารย์ที่ปรึกษา

(รศ.ดร. วันชัย ริ้วรุจา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรจัดการหน่วยความจำสำหรับแสดงผลสัญญาณบนจอซีอาร์ที

นาย พีรพล พัฒนวิจิตร
นาย วินัย แก้วโกวิท
รศ.ดร. วันชัย ริ้วรุจา , อาจารย์ที่ปรึกษา
ปีการศึกษา 2547

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้นำเสนอการสร้างวงจรจัดการหน่วยความจำสำหรับแสดงผลสัญญาณบนจอซีอาร์ที โดยใช้ในการจัดระบบหน่วยความจำหลายชุด เป็นที่เก็บข้อมูลสำรองไว้ก่อนที่จะนำไปใช้งานต่อๆไป

เนื่องจากในระบบการส่งข้อมูลเพื่อใช้ในการแสดงผลบนหน้าจอนั้นมีการส่งข้อมูลจำนวนมาก อีกทั้งยังต้องการความเร็วและความต่อเนื่อง จึงจำเป็นต้องทำการออกแบบพัฒนาวงจรที่ช่วยจัดการระบบส่งข้อมูลให้มีความเร็วและต่อเนื่องขึ้นมา รวมทั้งยังต้องสามารถเลือกที่จะวิเคราะห์สัญญาณในตำแหน่งที่ต้องการได้ด้วย ซึ่งเป็นข้อดีของการใช้ชุดหน่วยความจำมาเก็บข้อมูลสัญญาณก่อนนำไปแสดงผลบนหน้าจอ

โดยจะทำการออกแบบวงจรให้มีหน่วยความจำเป็น 3 ชุดเพื่อสำรองที่เก็บข้อมูลไว้ให้ระบบสามารถอ่านและเขียนข้อมูลได้ในเวลาเดียวกัน รวมทั้งยังสามารถปรับแต่งรูปสัญญาณให้ขยายใหญ่ขึ้นหรือเลื่อนช่วงที่จะพิจารณารูปสัญญาณอีกด้วย โดยจะใช้ไมโครคอนโทรลเลอร์เบอร์ PIC16F877 เพื่อควบคุมและจัดสรรลำดับในการอ่านและเขียนข้อมูลลงบนหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MEMORY MANAGEMENT CIRCUIT FOR DISPLAY ON CRT MONITOR

Mr. Peerapon Pattanavijit

Mr. Winai Kaewkowit

Associate Professor Vanchai Riewruja , Advisor

2004

Abstract

This Thesis presents a circuit that can manage memory system for display on CRT monitor. The circuit uses a random access memory and manages it to storage data before displaying.

Causes of there are many data transferred for monitor display system and must be fast and continuous. So it is necessary to design and develop circuit that can manage memory system for high speed and continuous. In addition of ability to capture signal data in which time that is the advantage of using memory to storage data before displaying on monitor.

By designing the circuit 3 pieces of static memory so the system can be written and read in the same time and it can capture data to zoom in or shift the signal to left, right, up or down. This project uses Microcontroller PIC16F877 to be main processor.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	**I**
บทคัดย่อภาษาอังกฤษ	**II**
สารบัญ	**III**
สารบัญตาราง	**IV**
สารบัญรูป	**V**
บทที่ 1 บทนำ	1
1.1 ความเป็นมาและความสำคัญ	1
1.2 ขอบเขตการศึกษา	1
1.3 วัตถุประสงค์ของโครงงาน	1
บทที่ 2 ทฤษฎี	3
2.1 ไมโครคอนโทรลเลอร์ (Microcontroller) ตระกูล PIC	3
2.1.1 PIC ชนิดต่างๆ	4
2.1.1.1 PIC แบบ OTP (One Time Programmable)	4
2.1.1.2 PIC แบบ EPROM (Erasable Programmable ROM)	4
2.1.1.3 PIC แบบ EEPROM/Flash (Electronically Erasable Programmable ROM)	5
2.1.2 ลักษณะของ PIC16F8XXX (FLASH MCUs)	5
2.1.3 โครงสร้างทางฮาร์ดแวร์ของไมโครคอนโทรลเลอร์ PIC16F87x	6
2.1.4 โครงสร้างการทำงานของไมโครคอนโทรลเลอร์ PIC16F87x	6
2.1.5 คุณสมบัติทางเทคนิคของ PIC16F87x/87x(A)	7
2.1.6 การจัดสรรหน่วยความจำข้อมูลของ PIC16F876/877	10
2.1.7 การจัดสรรหน่วยความจำโปรแกรม	12
2.1.8 โครงสร้าง MEMORY ของ 16F87x	13
2.1.8.1 PROGRAM MEMORY	13
2.1.8.2 DATA MEMORY	14
2.1.9 ชุดคำสั่งของ PIC 16F877x	16
2.1.10 การเข้าถึงรีจิสเตอร์และข้อมูลของไมโครคอนโทรลเลอร์ PIC	16
2.1.10.1 การเข้าถึงแบบทันทีทันใด	16
2.1.10.2 การเข้าถึงข้อมูลแบบโดยตรง	16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.10.3 การเข้าถึงข้อมูลโดยอ้อม 17
2.1.10.4 การเข้าถึงค่าข้อมูลแบบสัมพัทธ์ 17
2.2 พื้นฐานการรับส่งข้อมูล 17
2.2.1 การรับส่งข้อมูลแบบอนุกรม (Serial) 17
2.2.2 การรับส่งข้อมูลแบบขนาน (Parallel) 17
2.2.3 ข้อแตกต่างระหว่างการสื่อสารแบบขนานและการสื่อสารแบบอนุกรม 18
บทที่ 3 การออกแบบและการสร้าง 20
3.1 พอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ PIC 16F87x 20
3.1.1 พอร์ต A 20
3.1.2 พอร์ต B 22
3.1.3 พอร์ต C 24
3.1.4 พอร์ต D 26
3.1.5 พอร์ต E 27
3.2 การเปลี่ยนแอนะลอกเป็นดิจิตอล(ANALOG-TO-DIGITAL CONVERTER(A/D)) 28
3.2.1 รีจิสเตอร์ที่เกี่ยวข้อง 28
3.2.1.1 รีจิสเตอร์ ADCON0 28
3.2.1.2 รีจิสเตอร์ ADCON1 29
3.2.2 การทำงานของโมดูลแปลงสัญญาณแอนะลอกเป็นดิจิตอล 31
3.2.3 การเลือกความถี่สัญญาณนาฬิกาของการแปลงสัญญาณ 32
3.2.4 กระบวนการแปลงสัญญาณแอนะลอกเป็นดิจิตอล 33
3.2.5 ขั้นตอนการใช้งานโมดูลแปลงสัญญาณแอนะลอกเป็นดิจิตอล 33
บทที่ 4 การทดลองและผลการทดลอง 35
4.1 รายละเอียดวงจร 35
4.1.1 โหมดส่งข้อมูลต่อเนื่องปกติ 36
4.1.2 โหมดจับภาพนิ่ง 36
4.1.3 ปฏิบัติการเบื้องต้นในโหมดจับภาพนิ่ง 36
4.1.4 ฮาร์ดแวร์ที่ใช้ในระบบ 36
4.1.5 การสร้างวงจร 37
4.1.6 หลักการขยายสัญญาณ (Zoom) 41
4.1.7 หลักการเลื่อนสัญญาณ (Sliding) 42
4.1.8 วงจร 44
4.2 ผลการทดลอง 46

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ภาพถ่ายของอุปกรณ์และวงจร 55

บทที่ 5 บทสรุปและวิจารณ์ 57

5.1 ปัญหาและแนวทางแก้ไข 57

5.2 แนวทางในการพัฒนา 57

5.3 บทสรุปและวิจารณ์ 57

ภาคผนวก

- โปรแกรมสำหรับไมโครคอนโทรลเลอร์ 1
- โปรแกรมสำหรับไมโครคอนโทรลเลอร์ 2
- โปรแกรมสำหรับไมโครคอนโทรลเลอร์ 3
- ข้อมูลรายละเอียดของอุปกรณ์
  - PIC16F877
  - SRAM K6T1008C2E
  - SN74HC245N
  - 74HC573
  - SN74HC138

กิตติกรรมประกาศ

บรรณานุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 3.1 หน่วยความจำควบคุมเกี่ยวกับ A/D	28
ตารางที่ 3.2 แสดงบิต 7: ADFM	30
ตารางที่ 3.3 แสดงการแปลงสัญญาณให้ด้านซ้ายของ ADRESH	30
ตารางที่ 3.4 แสดงการแปลงสัญญาณให้ด้านขวาของ ADRESH	30
ตารางที่ 3.5 แสดงส่วนการใช้ช่องสัญญาณกับพอร์ต	31
ตารางที่ 4.1 แสดงรายละเอียดของสัญลักษณ์ต่างๆในวงจร	45

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

	หน้า
รูปที่ 2.1 PIC แบบ OTP (One Time Programmable)	4
รูปที่ 2.2 PIC แบบ EPROM (Erasable Programmable ROM)	5
รูปที่ 2.3 PIC แบบ EEPROM/Flash (Electronically Erasable Programmable ROM)	5
รูปที่ 2.4 ไดอะแกรมแสดงรูปแบบสถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบฮาร์วาร์ด	6
รูปที่ 2.5 โครงสร้างการทำงานของไมโครคอนโทรลเลอร์ PIC16F87x รุ่น 40 ขา เบอร์ PIC16F874(A)/877(A)	7
รูปที่ 2.6 PDIP-40 pin แบบ PIC16F874/877	9
รูปที่ 2.7 การจัดสรรหน่วยความจำข้อมูล RAM ของไมโครคอนโทรลเลอร์ PIC16F876(A)/877(A)	11
รูปที่ 2.8 การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ PIC16F876/877	13
รูปที่ 2.9 Memory map ของ PIC16F87X	14
รูปที่ 2.10 การรับส่งข้อมูลแบบขนาน	18
รูปที่ 2.11 รายละเอียดของ Printer Port หรือ Parallel Port	19
รูปที่ 3.1 บล็อกไดอะแกรมของขา RA3 : RA0 และ RA5	21
รูปที่ 3.2 บล็อกไดอะแกรมของขา RA4/TOCKI	22
รูปที่ 3.3 บล็อกไดอะแกรมของขา RB3 : RB0	23
รูปที่ 3.4 บล็อกไดอะแกรมของขา RB7 : RB4	24
รูปที่ 3.5 บล็อกไดอะแกรมของขา RC<2:0> และ RC<7:5>	25
รูปที่ 3.6 บล็อกไดอะแกรมของขา RC<4:3>	26
รูปที่ 3.7 บล็อกไดอะแกรมของพอร์ต D	27
รูปที่ 3.8 บล็อกไดอะแกรมของพอร์ต E	28
รูปที่ 3.9 ไดอะแกรมการทำงานของโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอลในPIC16F87x	32
รูปที่ 3.10 แสดงกลไกการแปลงสัญญาณอนาลอกเป็นดิจิตอลของไมโครคอนโทรลเลอร์ PIC16F87x	33
รูปที่ 4.1 ระบบ	37
รูปที่ 4.2 บล็อกไดอะแกรมของระบบ	38
รูปที่ 4.3 โฟลชาร์ตแสดงการทำงานของวงจร	39
รูปที่ 4.4 โฟลชาร์ตแสดง Write Cycle ของหน่วยความจำ	40
รูปที่ 4.5 โฟลชาร์ตแสดง Read Cycle ของหน่วยความจำ	40
รูปที่ 4.6 แสดงการขยายสัญญาณโดยการคัดลอกข้อมูลที่ 1 แอดเดรส ไป 4 แอดเดรส	41

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 4.7 แสดงการขยายสัญญาณบริเวณกลางจอ 41
รูปที่ 4.8 แสดงผลจากการขยายสัญญาณ 42
รูปที่ 4.9 แสดงตำแหน่งพิกัดต่างๆบนหน้าจอทั้ง 9 ส่วน 42
รูปที่ 4.10 แสดงขอบเขตการแสดงผลของการเลื่อนสัญญาณไปทางซ้ายและขวา 43
รูปที่ 4.11 โฟลชาร์ตแสดงการเลื่อนสัญญาณ 43
รูปที่ 4.12 แสดงรายอะเอียดวงจรจัดการหน่วยความจำสำหรับแสดงผลสัญญาณบนจอซีอาร์ที 44
รูปที่ 4.13 การใช้โปรแกรม Hyper Terminal จับค่า (Capture Text) ของสัญญาณ 46
รูปที่ 4.14 จับค่าสัญญาณออกมาเป็น TEXT ไฟล์ 46
รูปที่ 4.15 การแบ่งช่องหน้าจอออกเป็น 9 ส่วน 47
รูปที่ 4.16 ภาพผลการทดลอง ขณะยังไม่ทำการขยาย (1x) 47
รูปที่ 4.17 ภาพผลการทดลอง หลังทำการขยาย (2x) ตำแหน่งที่ 5 (กลางจอ) 47
รูปที่ 4.18 ภาพผลการทดลอง หลังทำการเลื่อนภาพมาทางซ้าย (2x) ตำแหน่งที่ 4 48
รูปที่ 4.19 ภาพผลการทดลอง หลังทำการเลื่อนภาพมาทางขวา (2x) ตำแหน่งที่ 6 48
รูปที่ 4.20 ภาพผลการทดลอง หลังทำการเลื่อนภาพขึ้นมาทางด้านบน (2x) ตำแหน่งที่ 2 48
รูปที่ 4.21 ภาพผลการทดลอง หลังทำการเลื่อนภาพลงมาทางด้านล่าง (2x) ตำแหน่งที่ 8 49
รูปที่ 4.22 ภาพผลการทดลอง หลังทำการเลื่อนภาพขึ้นมาทางด้านบนซ้าย (2x) ตำแหน่งที่ 1 49
รูปที่ 4.23 ภาพผลการทดลอง หลังทำการเลื่อนภาพขึ้นมาทางด้านบนซ้าย (2x) ตำแหน่งที่ 3 49
รูปที่ 4.24 ภาพผลการทดลอง หลังทำการเลื่อนภาพลงมาทางด้านล่างซ้าย (2x) ตำแหน่งที่ 7 50
รูปที่ 4.25 ภาพผลการทดลอง หลังทำการเลื่อนภาพลงมาทางด้านล่างขวา (2x) ตำแหน่งที่ 9 50
รูปที่ 4.26 ภาพผลการทดลองที่2 ขณะยังไม่ทำการขยาย (1x) 51
รูปที่ 4.27 ภาพผลการทดลองที่2 หลังทำการขยาย (2x) ตำแหน่งที่ 5 (กลางจอ) 51
รูปที่ 4.28 ภาพผลการทดลองที่2 หลังทำการเลื่อนภาพมาทางซ้าย (2x) ตำแหน่งที่ 4 51
รูปที่ 4.29 ภาพผลการทดลองที่2 หลังทำการเลื่อนภาพมาทางขวา (2x) ตำแหน่งที่ 6 52
รูปที่ 4.30 ภาพผลการทดลองที่2 หลังทำการเลื่อนภาพขึ้นมาทางด้านบน (2x) ตำแหน่งที่ 2 52
รูปที่ 4.31 ภาพผลการทดลองที่2 หลังทำการเลื่อนภาพลงมาทางด้านล่าง (2x) ตำแหน่งที่ 8 52
รูปที่ 4.32 ภาพผลการทดลองที่2 หลังทำการเลื่อนภาพขึ้นมาทางด้านบนซ้าย (2x) ตำแหน่งที่ 1 53
รูปที่ 4.33 ภาพผลการทดลองที่2 หลังทำการเลื่อนภาพขึ้นมาทางด้านบนซ้าย (2x) ตำแหน่งที่ 3 53
รูปที่ 4.34 ภาพผลการทดลองที่2 หลังทำการเลื่อนภาพลงมาทางด้านล่างซ้าย (2x) ตำแหน่งที่ 7 53
รูปที่ 4.35 ภาพผลการทดลองที่2 หลังทำการเลื่อนภาพลงมาทางด้านล่างขวา (2x) ตำแหน่งที่ 9 54
รูปที่ 4.36 ภาพไมโครคอนโทรลเลอร์และไอซีเบอร์ต่างๆ 55
รูปที่ 4.37 ภาพสายสัญญาณที่ใช้ และ บอร์ดโปรแกรม 55
รูปที่ 4.38 วงจรจัดการหน่วยความจำสำหรับแสดงผลสัญญาณบนจอซีอาร์ที 56

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาและความสำคัญ

ในปัจจุบันไมโครคอนโทรลเลอร์เป็นส่วนหนึ่งของงานทางด้านอิเล็กทรอนิกส์ ที่มีในการใช้ควบคุมทางอุตสาหกรรมเป็นอย่างมาก ซึ่งในปัจจุบันได้มีการพัฒนาไมโครคอนโทรลเลอร์ให้ทำงานในลักษณะ RISC (Reduce Instruction Set Computer) มีการใช้งาน 33-35 คำสั่ง หนึ่งคำสั่งใช้สัญญาณนาฬิกาเพียงลูกเดียวยกเว้นคำสั่งที่ใช้ในการกระโดด

ไมโครคอนโทรลเลอร์ตระกูล PIC ได้รับความนิยมให้นำไปควบคุมในงานต่างๆมากมาย โดยที่ในโครงงานนี้ จะเลือกใช้ไมโครคอนโทรลเลอร์ตระกูล PIC เบอร์ PIC16F877 ในการควบคุมและจัดสรรหน่วยความจำให้จัดเก็บและส่งข้อมูลออกไปแสดงผลยังหน้าจอซีอาร์ที กล่าวคือ ใช้เป็นส่วนที่พักข้อมูลก่อนจะส่งไปแสดงผลที่หน้าจอซีอาร์ที โดยที่ในการส่งข้อมูลของหน่วยความจำประเภท RAM แต่ละครั้งจะใช้เวลามาก ดังนั้นจึงต้องพัฒนาวงจรและโปรแกรมให้มีเสถียรภาพและทำให้เวลาในการส่งข้อมูลเร็วขึ้น

1.2 ขอบเขตการศึกษา

1. สร้างวงจรจัดการหน่วยความจำสำหรับแสดงผลสัญญาณบนจอซีอาร์ที
2. ศึกษาโครงสร้างและสถาปัตยกรรมของไมโครคอนโทรลเลอร์ตระกูล PIC เบอร์ PIC16F877
3. ศึกษาโครงสร้าง การประมวลผล และการติดต่อระหว่างไมโครคอนโทรลเลอร์กับหน่วยความจำประเภท RAM
4. ศึกษาพื้นฐานในการรับส่งข้อมูลในระบบคอมพิวเตอร์
5. สามารถใช้งานและเขียนโปรแกรมสั่งงานไมโครคอนโทรลเลอร์เบอร์ PIC 16F877
6. สามารถออกแบบและสร้างวงจรจริงที่ใช้ในการจัดการหน่วยความจำให้ทำงานได้อย่างมีเสถียรภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3 วัตถุประสงค์ของโครงงาน

1. สร้างวงจรจัดการหน่วยความจำสำหรับแสดงผลสัญญาณบนจอซีอาร์ทีซึ่งเป็นวงจรที่ใช้ในการจัดระบบและลำดับข้อมูลของหน่วยความจำประเภท RAM หลายชุด ที่ใช้จัดเก็บข้อมูลสำรองไว้ก่อนที่จะนำไปใช้งานต่อๆไป
2. ใช้เป็นส่วนที่พักของข้อมูลและสัญญาณ ซึ่งถูกประมวลผลเรียบร้อยแล้วก่อนที่จะนำไปแสดงผลที่ส่วนหน้าจอ โดยที่สัญญาณที่เข้ามาจะถูกแปลงจากสัญญาณอนาลอกเป็นสัญญาณดิจิตอลเสียก่อน จากนั้นจึงจะถูกส่งผ่านเข้าไปในหน่วยความจำ โดยการจัดสรรและติดต่อกับหน่วยความจำซึ่งจะถูกควบคุมโดยไมโครคอนโทรลเลอร์
3. ทำให้มีความต่อเนื่องของข้อมูลในการส่งข้อมูลไปแสดงผลยังหน้าจอ เพราะมีการสำรองข้อมูลไว้ที่หน่วยความจำประเภท RAM เป็นจำนวน 3 ตัว
4. สามารถเชื่อมต่อเข้า Interface แบบต่างๆได้อย่างมีเสถียรภาพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎี

วงจรจัดการหน่วยความจำสำหรับแสดงผลสัญญาณบนจอซีอาร์ที (Memory Management Circuit for Display on CRT Monitor) เป็นการนำเอาหลักการการใช้งานของไมโครคอนโทรลเลอร์ตระกูล PIC16F877 และการใช้งานหน่วยความจำประเภท RAM ขนาด 128 k x 8 เบอร์ K6T1008C2E มาสร้างเป็นวงจรเพื่อให้ใช้งานได้อย่างมีประสิทธิภาพ

2.1 ไมโครคอนโทรลเลอร์ (Microcontroller) ตระกูล PIC

PIC คือ ไมโครคอนโทรลเลอร์อีกตระกูลหนึ่ง ย่อมาจากคำว่า Peripheral Interface Controller ซึ่ง หลักการไมโครคอนโทรลเลอร์ตระกูลนี้ก็คือ พยายามรวมเอาทุกอย่างเอาไว้ในตัวของมันไม่ว่าจะเป็น PROGRAM MEMORY, RAM, EEPROM, SERIAL, I^2C, PWM, A/D ฯลฯ โดยไม่จำเป็นต้องต่ออุปกรณ์เสริมจากภายนอก ในตัวของ PIC จะมีฟังก์ชันที่ใช้ในการประมวลผล รวมทั้งหน่วยความจำ ซึ่งทำให้มันเหมือนกัน CPU ตัวหนึ่งเลย

ความเร็วของ PIC ภาคของความถี่สัญญาณนาฬิกา ปัจจุบันสามารถทำสัญญาณนาฬิกาได้ที่ 20 MHz ซึ่งทำให้หนึ่งคำสั่งของ PIC ใช้เวลาเพียง 0.2 µSec แต่อย่างไรก็ตามได้มีบริษัทอื่นได้ซื้อลิขสิทธิ์ PIC จาก Microchip และได้สร้าง Chip ที่มีความเร็วได้มากกว่าเดิมขึ้นไปอีก

หน่วยความจำของ PIC ในอดีตหน่วยความจำของ PIC จะค่อนข้างน้อย คืออยู่ระหว่าง 512 words ถึง 4 กิโลเวิร์ด แต่ในปัจจุบัน บริษัท Microchip ซึ่งเป็นเจ้าของ PIC ได้พัฒนาจนทำให้หน่วยความจำ (Memory) ของ PIC มีขนาดเป็นหลายสิบกิโลไบต์ และมีทีท่าว่าจะขยายได้ใหญ่ขึ้นเรื่อยๆ ในเรื่องของการนับขนาดของหน่วยความจำของ PIC จะนับไม่เหมือนปกติ โดยที่ หนึ่งคำสั่งของ PIC จะมีขนาด 14 bits ดังนั้นเราจะเรียกว่า 1 เวิร์ด ของ PIC จะมีขนาด 14 บิต เช่น PIC16F84A ระบุว่ามีหน่วยความจำ 1 k (ซึ่งหมายถึง 1 กิโลเวิร์ด) ถ้าคำนวณให้เป็นแบบ 1 ไบต์ = 8 บิต จะได้ว่า 1 x 1,024 x 14 = 14,336 บิต ดังนั้นก็คือ 14,336/(8 x 1,024) = 1.75 กิโลไบต์ นั่นเอง

สถาปัตยกรรมของ PIC ในตอนนี้มี 3 สายหลักๆ สมัยก่อนจะมีแค่สอง คือขึ้นต้นด้วย 16xxx,17xxx และใหม่ล่าสุดคือ 18xxx ถ้าพูดถึง คุณสมบัติที่เหนือกว่าเรียงจากน้อยสุดไปมากสุดก็คือ 16 -> 17 -> 18 คำสั่ง Assembly ของ 17 และมี 18 จะมีมากกว่า 16 ทำให้เขียนโปรแกรมได้ง่ายกว่า ราคาก็จะสูงกว่าด้วย แต่ที่เป็นที่นิยมก็คือตระกูล 16xxx

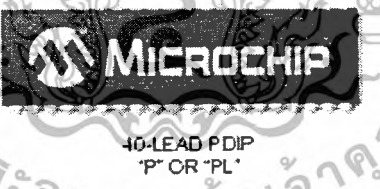
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุปหลักการสถาปัตยกรรมของ PIC จะยึดถือการออกแบบที่ว่ารวมทุกอย่างไว้ใน Chip ตัวเดียวโดยไม่ต้องต่ออุปกรณ์ใดๆ เพิ่มเติม ผลที่ตามมาก็คือแผ่นวงจรจะมีขนาดเล็ก และอุปกรณ์ที่ใช้จะไม่มาก บางงานอาจจะใช้แค่ PIC เพียงตัวเดียวโดยไม่ต้องใช้ Chip อื่นมาเพิ่มเติมเลย นี่คือคุณสมบัติพิเศษของ PIC ซึ่งปัจจุบันหลายบริษัทที่ผลิตไมโครคอนโทรลเลอร์ก็เริ่มจะหันมาเลียนแบบแนวทางนี้กันแล้ว แต่ทุกอย่างย่อมมีข้อเสีย เนื่องจากความคิดที่จะรวมทุกอย่างไว้ใน Chip ตัวเดียว ทำให้ Program Memoryและ Data Memory ไม่สามารถขยายโดยใช้กับ Memory ภายนอกได้ PIC จึงเหมาะสำหรับงานเล็กๆ ไม่ใช่งานใหญ่ๆ ที่ต้องใช้การคำนวณ และ Memory เยอะๆ เปรียบเทียบแล้วสำหรับ PIC ก็คือมดซึ่งเป็นแมลงที่สามารถยกของที่มีน้ำหนักมากกว่ามันได้หลายสิบเท่านั่นเอง

2.1.1 PIC ชนิดต่างๆ

MCU ในตระกูล PIC ถ้าแบ่งออกตามชนิดของ Program Memoryแบ่งได้เป็น 3 แบบคือ

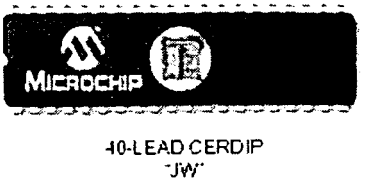
2.1.1.1 OTP (One Time Programmable) เป็นไอซีที่มีราคาถูกที่สุด ในสามประเภท สาเหตุก็มาจากว่าไอซีแบบ OTP จะสามารถทำการโปรแกรมได้แค่ครั้งเดียวเท่านั้น หลังจาก Chip ได้ถูกโปรแกรมไปแล้วจะไม่สามารถโปรแกรมเข้าไปใหม่ได้อีก ดังนั้นไอซีประเภทนี้จะนิยมใช้หลังจากได้พัฒนาโปรแกรมจนกระทั่งจุดบกพร่องต่างๆ ในโปรแกรม ไม่มีอีกแล้ว เพราะจะมีต้นทุนต่ำเมื่อเทียบตัว Memory ประเภทอื่น จะมีตัวอักษร C แสดงบนตัวไอซี เช่น 16C84, 16C74



รูปที่ 2.1PIC แบบ OTP (One Time Programmable)

2.1.1.2 EPROM (Erasable Programmable ROM) เป็นไอซีที่มี Program Memory ที่เมื่อเขียนโปรแกรมเข้าไปแล้วสามารถโปรแกรมใหม่ด้วยการลบโปรแกรมเดิมโดยให้แสง UV (Ultra Violet) ส่องผ่านเข้าไปยังไอซีประมาณ 5-10 นาที ดังนั้นที่ด้านบนของไอซีจะมีกรอบกระจกเพื่อให้แสดง UV สามารถส่องผ่านเข้าไปในตัวไอซีได้ แต่ก็มีจำนวนครั้งในการลบโปรแกรมเช่นกัน เมื่อลบโปรแกรมด้วยแสง UV มากๆ เข้าก็จะเกิดอาการด้าน คือโปรแกรมไม่เข้านั่นเอง จะมีตัวอักษร JW หรือว่าดูเอาว่ามีกรอบกระจกอยู่บนไอซีหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 PIC แบบ EPROM (Erasable Programmable ROM)

2.1.1.3 EEPROM / Flash (Electronically Erasable Programmable ROM) เป็นไอซีที่ออกมาไม่กี่ปีนี้เอง ส่วนของ Program Memoryสามารถอ่านหรือเขียนด้วยสัญญาณทางไฟฟ้า ใช้เวลาในการ ลบข้อมูลไม่กี่วินาที และสามารถลบ และเขียนใหม่ได้หลายพันครั้ง ทำให้เป็นที่นิยมที่สุดใน 3 ประเภท มีตัวอักษร F เป็นตัวบอก เช่น 16F84, 16F877

รูปที่ 2.3 PIC แบบ EEPROM / Flash (Electronically Erasable Programmable ROM)

2.1.2 ลักษณะของ PIC16F8XXX (FLASH MCUs)

- มีคำสั่งในภาษาแอสแซมบลี (Assembly) 35 คำสั่ง
- มี I/O, มีไทเมอร์ (Timer) มากกว่า 1 ตัว, Watch dog, I2C, USART, SPI, PWM
- มี A/D ขนาด 10 บิต
- มี Program Memory เป็นแบบ Flash ทำให้สามารถโปรแกรมใหม่ได้หลายครั้ง
- มี EEPROM ภายใน
- ในตระกูล 16F87X สนับสนุน In Circuit Debugger (ICD) เป็นผลทำให้ไม่จำเป็นต้องซื้อ Emulator ราคาแพง

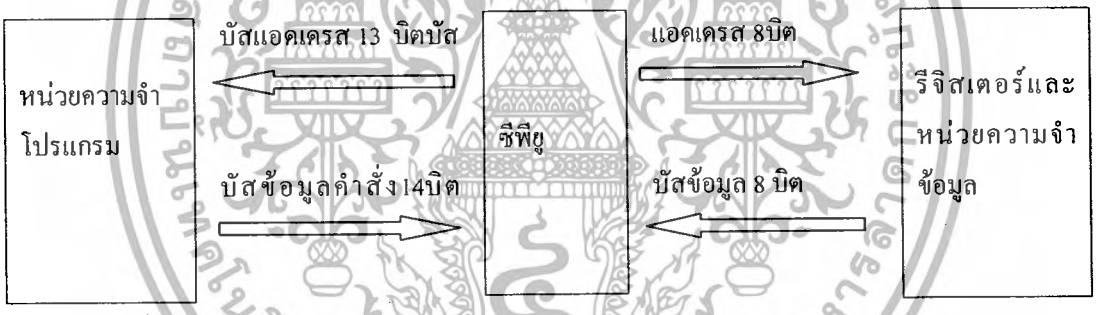
จุดด้อยของ PIC ก็คือ Program Memory มีลักษณะเป็น Page ทางบริษัท Microchip เองก็คงเห็นจุดด้อยตรงนี้จึงได้พัฒนาออกมาเป็น PIC 18CXXX ซึ่ง Program Memory ไม่ถูกแบ่งเป็น Page อีกต่อไป และเพิ่มคำสั่งแอสแซมบลี (เป็น 77 คำสั่ง รวมทั้งออกแบบให้ Program Memory มีขนาดใหญ่ขึ้นเพื่อรองรับ การเขียนโปรแกรมภาษา C หลังจากนั้นจึงได้ออก PIC 18FXXX ตามมา พร้อมกับ ICD2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.3 โครงสร้างทางฮาร์ดแวร์ของไมโครคอนโทรลเลอร์ PIC16F87x

ไมโครคอนโทรลเลอร์ตระกูล PIC มีสถาปัตยกรรมแบบฮาร์วาร์ด (Harvard architecture) กล่าวคือ มีการแยกหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลออกจากกัน โดยมีบัสสำหรับติดต่อแยกกันด้วย ดังแสดงในรูปที่ 2.4 จะเห็นว่าซีพียูภายในไมโครคอนโทรลเลอร์จะติดต่อกับหน่วยความจำโปรแกรมด้วยบัสแอดเดรส 13 บิต และบัสข้อมูลหน่วยความจำโปรแกรม 14 บิต ในขณะที่บัสสำหรับติดต่อกับหน่วยความจำข้อมูลและรีจิสเตอร์ภายในเป็นแบบ 8 บิตทั้งบัสแอดเดรสและบัสข้อมูล

นอกจากการจัดสถาปัตยกรรมแบบนี้แล้ว การกระทำคำสั่งของไมโครคอนโทรลเลอร์ PIC ยังใช้กระบวนการที่เรียกว่า "ไปป์ไลน์" (Pipeline) ทำให้สามารถ Fetch คำสั่งถัดไป ในขณะที่กำลัง Execute คำสั่งในปัจจุบัน ส่งผลให้ความเร็วในการทำงานของไมโครคอนโทรลเลอร์เพิ่มมากขึ้น นั่นจึงเป็นที่มาของความสามารถในการกระทำคำสั่ง 1 คำสั่งภายในสัญญาณนาฬิกา 1 ลูก (Fetch เป็นกระบวนการเรียกคำสั่งออกจากหน่วยความจำโปรแกรมแล้วแปลเป็นเลขฐานสิบหกเพื่อให้ซีพียูเข้าใจ ส่วนกระบวนการ Execute เป็นการกระทำคำสั่งให้เกิดผลลัพธ์ตามที่คำสั่งนั้นๆกำหนด)

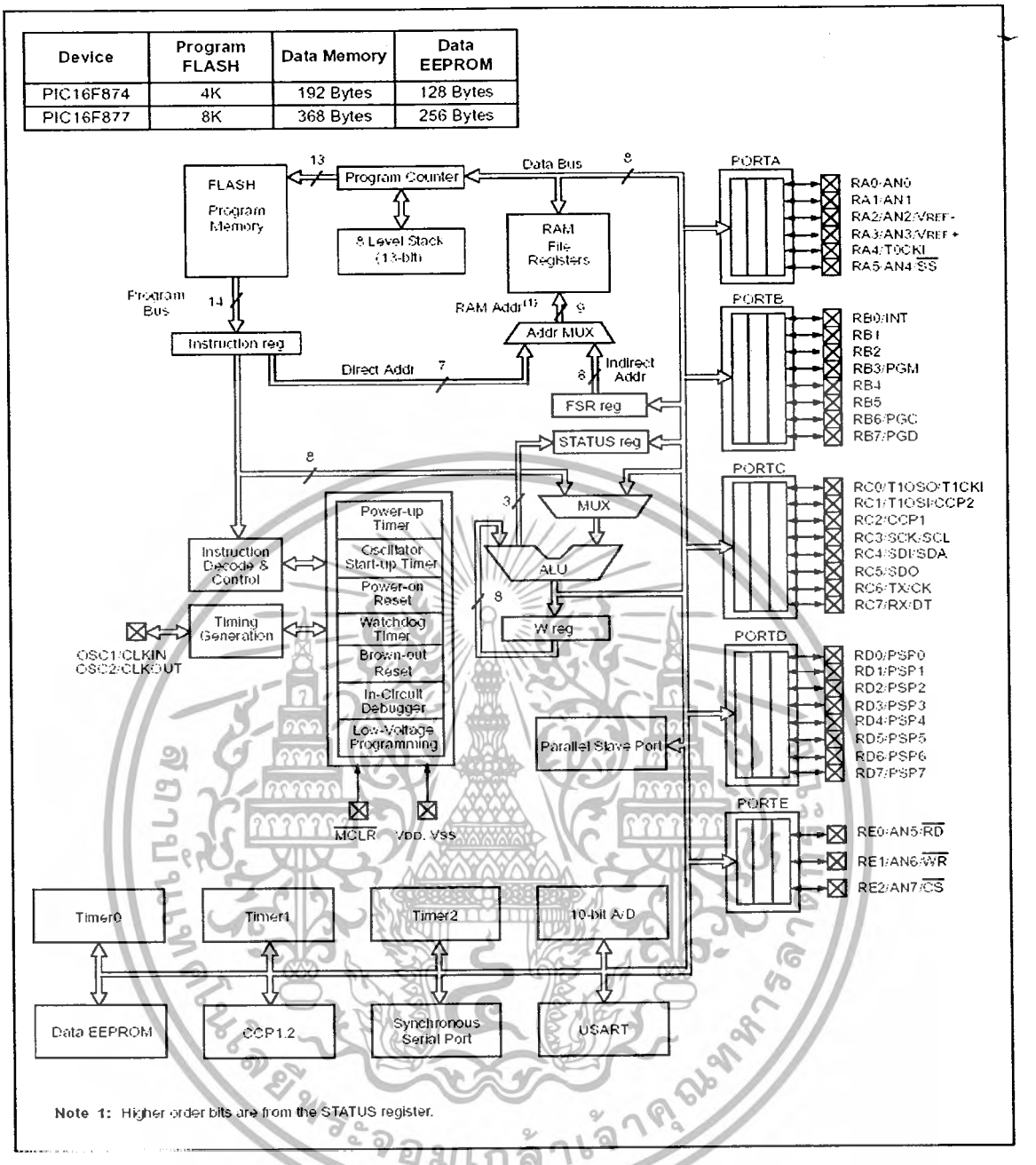


รูปที่ 2.4 ไดอะแกรมแสดงรูปแบบสถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบฮาร์วาร์ด

2.1.4 โครงสร้างการทำงานของไมโครคอนโทรลเลอร์ PIC16F87x

แสดงดังในรูปที่ 2.5 ส่วนประกอบหลักก็จะเหมือนกับไมโครคอนโทรลเลอร์ PIC16F84 แต่จะมีส่วนที่เพิ่มเติมเข้ามามากพอควร ได้แก่ วงจร Brown-out reset , ส่วนแก้ไขข้อมูลในวงจรหรือ In-circuit debugger , วงจรโปรแกรมข้อมูลด้วยแรงดันต่ำ (low-voltage Programming) , Timer ที่มีมากถึง 3 ตัว , วงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอลขนาด 10 บิต , วงจรเชื่อมต่ออุปกรณ์อนุกรม (SPI : Serial Peripheral Interfacing) , วงจรเชื่อมต่อระบบบัส I^2C , วงจรสื่อสารอนุกรม (USART : Universal Synchronous Receiver Transmitter) และโมดูลเปรียบเทียบสัญญาณ-ตรวจจับสัญญาณ-วงจรมอดูเลชั่นทางความกว้างของ Pulse หรือ PWM (CCP : Compare Capture Pulse-width modulation) นอกจากนั้นในอนุกรม PIC16F87xA จะมีวงจรเปรียบเทียบแรงดันอะนาลอกและโมดูลสร้างแรงดันอ้างอิงเพิ่มเติมเข้ามาอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 โครงสร้างการทำงานของไมโครคอนโทรลเลอร์ PIC16F87x รุ่น 40 ขา เบอร์ PIC16F874(A)/877(A)

2.1.5 คุณสมบัติทางเทคนิคของ PIC16F87x/87x(A)

คุณสมบัติหลัก

- ซีพียูเป็นแบบ RISC (Reduced Instruction-Set Computer) มีคำสั่งใช้งานเพียง 35 คำสั่ง
- สามารถกระทำคำสั่งโดยใช้สัญญาณเพียงหนึ่งลูก ยกเว้นคำสั่งการกระโดด
- ความถี่สัญญาณนาฬิกา ตั้งแต่ไฟตรงถึง 20 MHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

• หน่วยความจำโปรแกรม

2 กิโลเวิร์ด สำหรับ PIC16F870/871/872

4 กิโลเวิร์ด สำหรับ PIC16F873(A)/874(A)

8 กิโลเวิร์ด สำหรับ PIC16F876/876(A)/877/877(A)

• หน่วยความจำข้อมูลแรมหรือรีจิสเตอร์

128 ไบต์ สำหรับ PIC16F870/871/872

192 ไบต์ สำหรับ PIC16F873(A)/874(A)

368 ไบต์ สำหรับ PIC16F876(A)/877(A)

• ขนาดหน่วยความจำข้อมูล EEPROM

64 ไบต์ สำหรับ PIC16F870/871/872

128 ไบต์ สำหรับ PIC16F873(A)/874(A)

256 ไบต์ สำหรับ PIC16F876(A)/877(A)

• ตอบสนองแหล่งกำเนิด Interrupt สูงสุดถึง 15 แหล่งขึ้นกับเบอร์ของไมโครคอนโทรลเลอร์

• มีสแต็ก 8 ระดับ

• มีวงจรเพาเวอร์ออนรีเซต (POR)

• มีเพาเวอร์อัปไทเมอร์ (PWRT) และออสซิลเลเตอร์สตาร์ทอัปไทเมอร์ (OST)

• มีวงจรวอตช์ด็อกไทเมอร์ (WDT) ที่มีวงจรออสซิลเลเตอร์ในตัว ทำให้มีความน่าเชื่อถือในการทำงานสูง

• เลือกป้องกันข้อมูลทั้งในหน่วยความจำโปรแกรมและหน่วยความจำข้อมูลสามารถเลือกระดับการป้องกันได้

• มีโหมดประหยัดพลังงาน

• สามารถโปรแกรมโดยใช้แรงดัน +5 V ได้

• แก้ไขข้อมูลในหน่วยความจำโปรแกรมด้วยกระบวนการ ICD (In-circuit Debugger) ผ่านพอร์ตเพียง 2 ขา

• ซีพียูสามารถอ่านและเขียนหน่วยความจำโปรแกรมได้

• ไฟเลี้ยง +2 ถึง +5.5 V

• กระแสซิงก์และซอร์สของพอร์ต 25 mA

• การใช้พลังงานไฟฟ้าในกรณีไม่ขับโหลด

- น้อยกว่า 2 mA ที่ไฟเลี้ยง +5 V และสัญญาณนาฬิกา 4 MHz
- ค่า 20 μA ที่ไฟเลี้ยง +3 V และสัญญาณนาฬิกา 32 kHz
- น้อยกว่า 1 μA ในโหมดประหยัดพลังงานหรือสแตนด์บาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติพิเศษเพิ่มเติม

• ไทเมอร์ 3 ตัว คือ ไทเมอร์ 0 ขนาด 8 บิต มีปรีสเกลเลอร์ขนาด 8 บิตในตัว, ไทเมอร์ 1 ขนาด 16 บิต พร้อมปรีสเกลเลอร์ และ ไทเมอร์ 2 ขนาด 8 บิต มีปรีสเกลเลอร์, โพสต์สเกลเลอร์ และรีจิสเตอร์คาบเวลา (period register) ขนาด 8 บิตในตัว

• มีโมดูล CCP 2 ชุด โดย

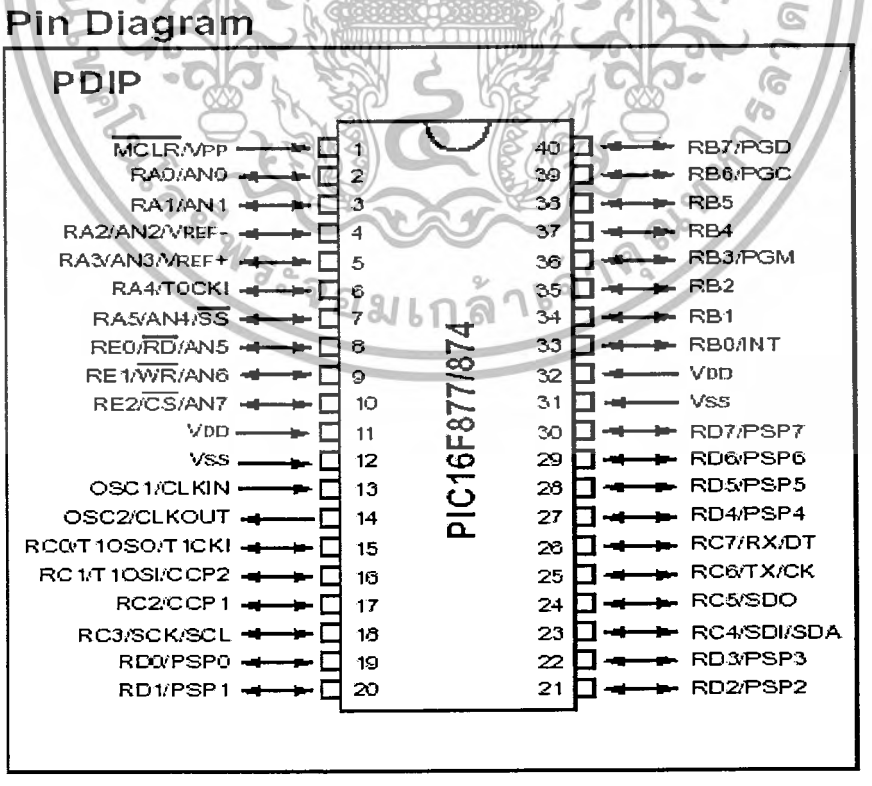
- ส่วนตรวจจับสัญญาณหรือ Captureมีขนาด 16 บิต ความละเอียดสูงสุด 12.5 นาโนวินาที
- ส่วนเปรียบเทียบสัญญาณ (Compare) มีขนาด 16 บิต ความละเอียดสูงสุด 200 นาโนวินาที
- วงจร PWM มีความละเอียดสูงสุด 10 บิต

• มีวงจรแปลงสัญญาณอะนาลอกเป็นสัญญาณดิจิตอลขนาด 10 บิต โดยที่ 5 ช่องสำหรับ PIC16F873(A)/876(A) และ 8 ช่องสำหรับ PIC16F874(A)/877(A)

• วงจรเชื่อมต่ออุปกรณ์ทั้ง SPI และบัส I^2C

• วงจรสื่อสารข้อมูลอนุกรม (USART) พร้อมการตรวจจับแอดเดรส 9 บิต

• มีวงจรตรวจจับระดับแรงดันไฟเลี้ยง (Brown-out detection) เพื่อการรีเซตซีพียู หรือเรียกว่า Brown-out Reset : BOR



รูปที่ 2.6 PDIP-40 pin แบบ PIC16F874/877

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 การจัดสรรหน่วยความจำข้อมูลของ PIC16F876/877

ใน PIC16F876/877 มีหน่วยความจำข้อมูลแรมสำหรับใช้งานทั่วไป 368 ไบต์ และมีรีจิสเตอร์ไฟล์ 8 บิต 57 ตัว หรือ 59 ตัวในเบอร์ PIC16F876A/877A ดังในรูปที่ 1-4 แต่ละแบงก์มีขนาดสูงสุด 128 ไบต์ แต่มีการใช้งานได้จริงในแต่ละแบงก์ต่างกัน โดนในแต่ละแบงก์มีการจัดสรรพื้นที่ ดังนี้

แบงก์ 0 มีช่วงแอดเดรส 0x00-0x7F

แอดเดรส 0x00-0x1F เป็นพื้นที่รีจิสเตอร์ไฟล์

แอดเดรส 0x20-0x7F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 96 ไบต์

แบงก์ 1 มีช่วงแอดเดรส 0x80-0xFF

แอดเดรส 0x80-0x9F เป็นพื้นที่ของรีจิสเตอร์ไฟล์ แต่มีบางแอดเดรสไม่ใช้งาน

แอดเดรส 0xA0-0xEF เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 80 ไบต์

แอดเดรส 0xF0-0xFF บรรจุข้อมูลเหมือนกับในแอดเดรส 0x70-0x7F ในแบงก์ 0 เพื่อช่วยให้สามารถใช้ข้อมูลจากแอดเดรส 0x70-0x7F ได้ง่ายขึ้น โดยไม่ต้องเปลี่ยนแบงก์

แบงก์ 2 มีช่วงแอดเดรส 0x100-0x17F

แอดเดรส 0x100-0x10F เป็นพื้นที่ของรีจิสเตอร์ไฟล์ แต่มีบางแอดเดรสไม่ใช้งาน

แอดเดรส 0x110-0x11F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 16 ไบต์

แอดเดรส 0x120-0x16F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 80 ไบต์

แอดเดรส 0x170-0x17F บรรจุข้อมูลเหมือนกับในแอดเดรส 0x70-0x7F ในแบงก์ 0 เพื่อช่วยให้สามารถใช้ข้อมูลจากแอดเดรส 0x70-0x7F ได้ง่ายขึ้น โดยไม่ต้องเปลี่ยนแบงก์

แบงก์ 3 มีช่วงแอดเดรส 0x180-0x1FF

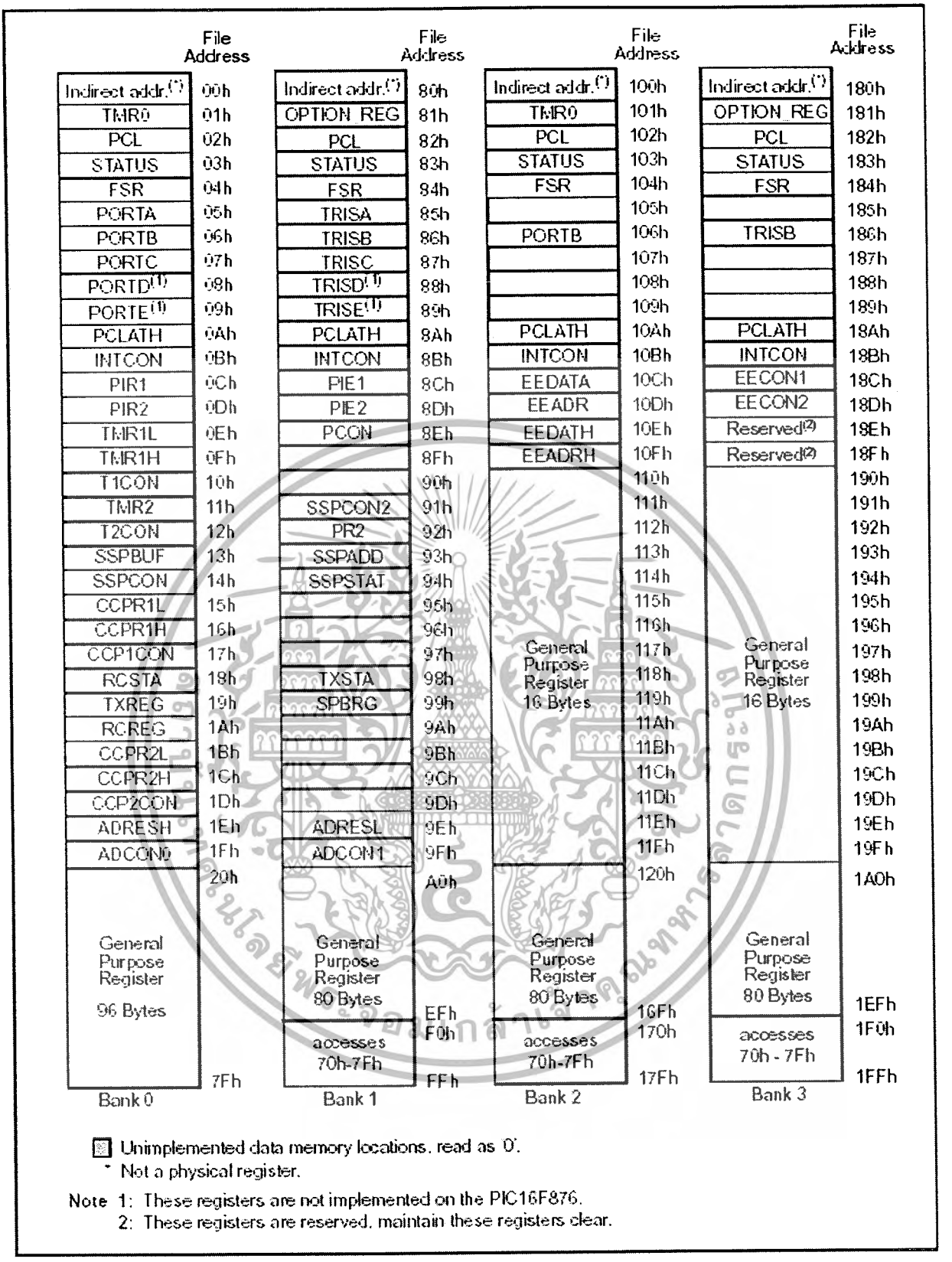
แอดเดรส 0x180-0x18F เป็นพื้นที่ของรีจิสเตอร์ไฟล์ แต่มีบางแอดเดรสไม่ใช้งาน

แอดเดรส 0x190-0x19F เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 16 ไบต์

แอดเดรส 0x1A0-0x1EF เป็นพื้นที่ของหน่วยความจำข้อมูลสำหรับใช้งานทั่วไป 80ไบต์

แอดเดรส 0x1F0-0x1FF บรรจุข้อมูลเหมือนกับในแอดเดรส 0x70-0x7F ในแบงก์ 0 เพื่อช่วยให้สามารถใช้ข้อมูลจากแอดเดรส 0x70-0x7F ได้ง่ายขึ้น โดยไม่ต้องเปลี่ยนแบงก์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 การจัดสรรหน่วยความจำข้อมูล RAM ของไมโครคอนโทรลเลอร์ PIC16F876(A)/877(A)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.7 การจัดสรรหน่วยความจำโปรแกรม

หน่วยความจำโปรแกรม (Program Memory) เป็นส่วนที่มีความสำคัญมากเพราะเป็นที่เก็บข้อมูลคำสั่งทั้งหมดซึ่งใช้ในการกำหนดให้ไมโครคอนโทรลเลอร์ทำงาน หน่วยความจำโปรแกรมของ PIC16F87x เป็นแบบแฟลช (Flash Memory) ทำให้สามารถลบและเขียนใหม่ได้นับแสนครั้ง แต่อย่างไรก็ตามโดยปกติ หน่วยความจำโปรแกรมหลังจากที่ทำการเขียนในขั้นตอนของการโปรแกรมแล้วก็จะมีไว้สำหรับอ่านออกมาได้เพียงทางเดียว

PIC16F87x มีโปรแกรมเคาน์เตอร์ (PC) ขนาด 13 บิตเพื่อกำหนดการเข้าถึงหน่วยความจำโปรแกรม โดยใน PIC16F876/877 จะมีขนาด 8k x 14 บิต (หรือ 8 กิโลเวิร์ด โดย 1 เวิร์ดมีขนาด 14 บิต) เนื่องจากไมโครคอนโทรลเลอร์ในอนุกรมนี้มีขนาดหน่วยความจำโปรแกรมที่ค่อนข้างใหญ่มากจึงต้องมีการจัดสรรเป็น Page หรือเป็นหน้า โดยในแต่ละ Page จะมีขนาด 2 กิโลเวิร์ด ทั้งนี้เนื่องจากชุดคำสั่งเกี่ยวกับการกระโดดของไมโครคอนโทรลเลอร์ตระกูล PIC สามารถอ้างอิงตำแหน่งของหน่วยความจำได้สูงสุด 2,048 ตำแหน่ง

ในรูปที่ 2.8 แสดงการจัดสรรพื้นที่ของหน่วยความจำโปรแกรมของ PIC16F876/877 เป็นการจัดสรรแบบปกติ มีขนาดหน่วยความจำโปรแกรม 8 กิโลเวิร์ด มีการสงวนแอดเดรส 0x0000 และ 0x0004 ไว้ ซึ่งเป็นรูปแบบมาตรฐานของการเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์ตระกูลนี้ สำหรับ PIC16F876/877 มีการแบ่งหน่วยความจำโปรแกรมออกเป็น 4 Page ดังนี้

- Page 0 มีแอดเดรสในช่วง 0x0000 - 0x07FF(โดยควรสงวนแอดเดรส 0x0000และ 0x0004ไว้)
- Page 1 มีแอดเดรสในช่วง 0x0800 - 0x0FFF
- Page 2 มีแอดเดรสในช่วง 0x1000 - 0x17FF
- Page 3 มีแอดเดรสในช่วง 0x1800 - 1x1FFF

นอกจากนั้นใน PIC16F87x ยังมีพื้นที่หน่วยความจำพิเศษสำหรับเก็บค่าของโปรแกรมเคาน์เตอร์ชั่วคราวขนาด 13 บิต เรียกว่า สแต็ก (Stack) ซึ่งจะมีบทบาทมากในการกระโดดไปทำงานยังโปรแกรมย่อยของ PIC16F87x โดยเมื่อกระทำคำสั่งให้กระโดดไปทำงานยังโปรแกรมย่อย ซีพียูจะทำการเก็บค่าโปรแกรมเคาน์เตอร์หรือ PC ในขณะนั้นไว้ในสแต็ก จากนั้นจึงกระโดดไปทำงานยังโปรแกรมย่อย เมื่อทำงานเรียบร้อยแล้ว ซีพียูจะไปอ่านค่า PC จากสแต็กกลับมา แล้วทำงานตามกระบวนการในโปรแกรมหลักต่อไป สำหรับสแต็กใน PIC16F87x มีขนาด 13 บิต สามารถเก็บค่าของ PC ได้ 8 ระดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

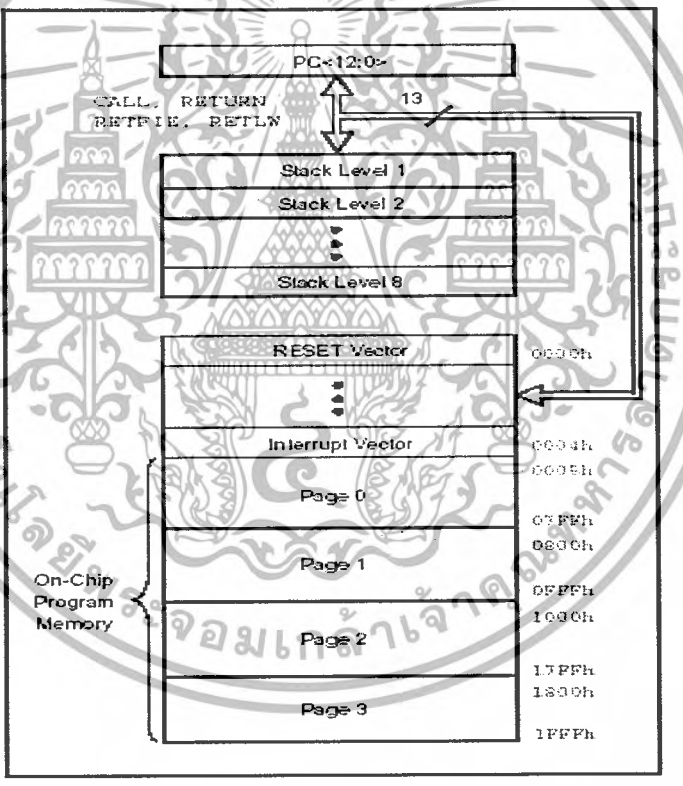
2.1.8 โครงสร้าง MEMORY ของ PIC16F87x

Memory ใน PIC16F87X มีอยู่ 4 ประเภทคือ

- Program Memory
- Data Memory
- Stack Memory
- EEPROM Memory

2.1.8.1 PROGRAM MEMORY

PIC16F87x จะมีขนาดของ Program Memoryซึ่งสามารถอ้างได้ถึง 8 กิโลไบต์โดย PIC16F877/876 จะมีขนาดหน่วยความจำเท่ากับ 8k x 14 และ PIC16F873/874 มีขนาด 4k x 14 ซึ่ง ตำแหน่ง reset vector จะอยู่ที่ 0000h และ Interrupt Vector จะอยู่ที่ 0004h



รูปที่ 2.8 การจัดสรรหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ PIC16F876/877

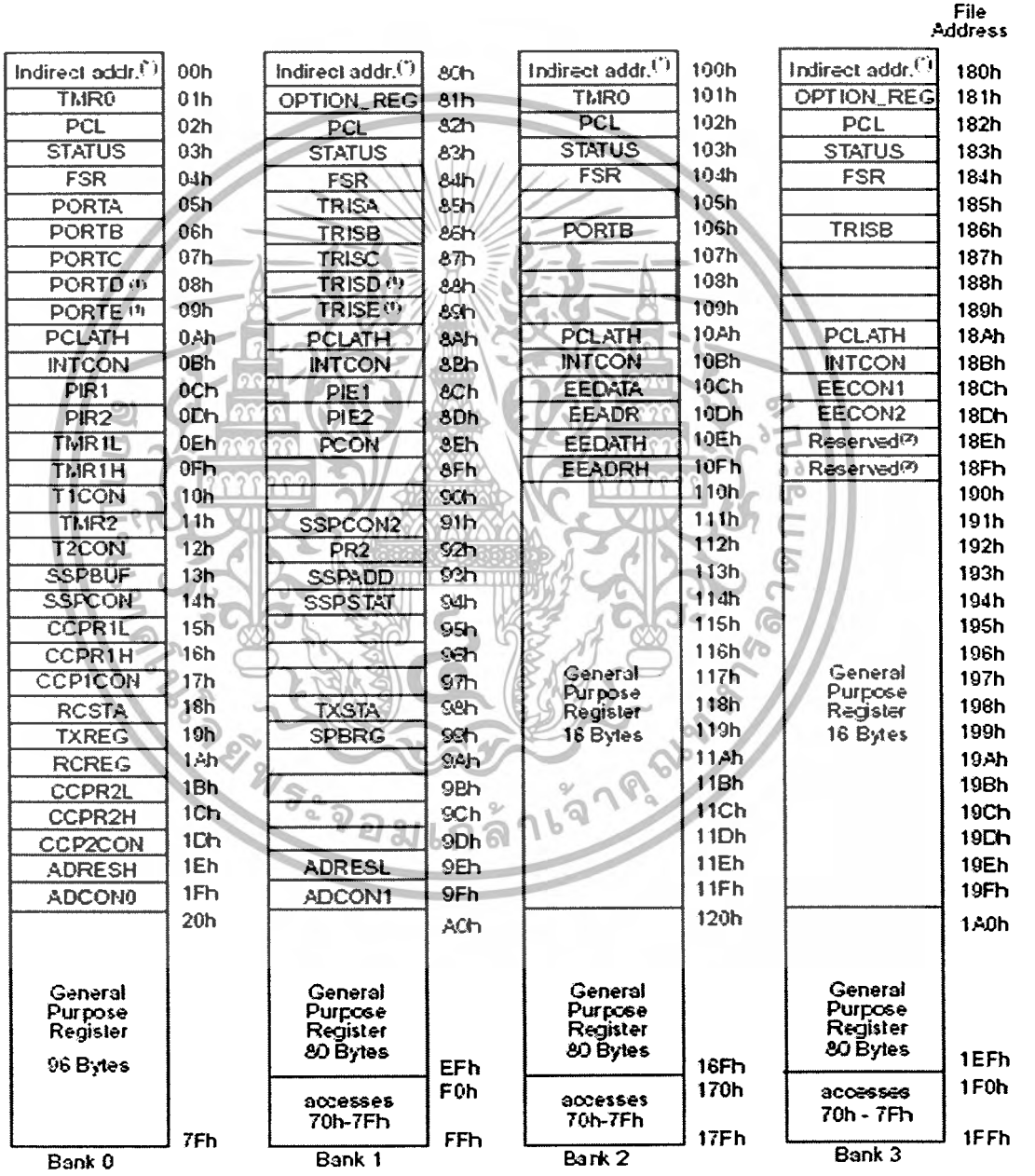
รีจิสเตอร์ที่เกี่ยวข้องกับ Program Memory ของ PIC คือ PCL และ PCLATH ซึ่งก็คือ Program counter LOW และ HIGH byte นั่นเอง โดยที่ LOW byte จะมีขนาด 8 บิต ส่วน HIGH byte จะมี ขนาด 5 บิต ซึ่งทำให้มีขนาดรวมกัน 13 บิต ก็คือสามารถอ้างหน่วยความจำได้ 8 กิโลไบต์นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.8.2 DATA MEMORY

Data Memory ของ pic16F87X จะมีอยู่ 2 ส่วนคือ Special Register Memory และ General Purpose Register

- **Special register** คือ Register ที่ถูกใช้ โดย MCU และ Peripheral Modules เพื่อใช้ในการควบคุมการทำงานของ MCU โดย Register พวกนี้จะมีชื่อเฉพาะอยู่แล้ว โดยสามารถเข้าถึงได้เหมือน RAM ปกติ
- **General purpose register** คือ Register ที่สามารถนำไปใช้งานทั่วๆไปได้ ขึ้นอยู่กับผู้ใช้งาน



รูปที่ 2.9 Memory map ของ PIC16F87X

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ PIC16F876 และ PIC16F877 จะมี General Purpose Register อยู่ที่ Address

• 20h – 7Fh

• A0h – EFh ถ้าอ้าง Address F0h – FFh PIC จะมองว่าเป็น 70h-7Fh

• 110h – 11Fh, 120h – 16Fh ถ้าอ้าง Address 170h – 17Fh PIC จะมองว่าเป็น 70h-7Fh

• 190h – 19Fh, 1A0h – 1EFh ถ้าอ้าง Address 1F0h – 1FFh PIC จะมองว่าเป็น 70h-7Fh

ในการอ้าง Memory ของ General Purpose Register ทำได้ 2 วิธีคือ แบบ Direct Access กับ Indirect Access

Direct Access ก็คือการอ้าง Address นั้นโดยตรงนั่นเอง เช่น

Read

movfw 0x21 ; อ่านข้อมูลที่ Address 0x20 ไปใส่ไว้ที่ w Register

Write

movwf 0x40 ; เขียนข้อมูลที่อยู่ใน w Register ไปไว้ใน Address 0x40

จะเห็นว่า Data Memory ของ PIC16F87X จะ แบ่งเป็น 4 แบงก์ ในการที่เราจะอ้าง Memory แบบDirect Access ที่แบงก์ไหนเราจะต้องสั่งให้ MCU ชี้ไปที่แบงก์นั้นเสียก่อน แล้วค่อยทำการ Direct Access โดย Register ที่เกี่ยวข้องก็คือ STATUS Register บิต 6 กับ 5 ซึ่งเรียกว่า RP1,RP0

• RP1,RP0 > 00 เลือก แบงก์ 0

• RP1,RP0 > 01 เลือก แบงก์ 1

• RP1,RP0 > 10 เลือก แบงก์ 2

• RP1,RP0 > 11 เลือก แบงก์ 3

เช่น

BSF STATUS,RP0 ; เลือก ไปยัง แบงก์ 1

BCF STATUS,RP1 ;

Indirect Access คือการอ้าง Memory ของ General Purpose Register ในตำแหน่งนั้นๆ โดยผ่าน Register 2 ตัว คือ FSR และ INDF ซึ่ง FSR จะทำหน้าที่เหมือน Pointer Address ส่วน Data ในการ Read/Write จะกระทำผ่านทาง INDF ตัวอย่างเป็นดังนี้

movlw 0x20 ; ใส่ค่า 20h ไปยัง w

movwf FSR ; ให้ Pointer ชี้ไปยัง Address 20h

clrf INDF ; ทำการ Clear ค่า ณ ตำแหน่ง Memory ที่ FSR ชี้อยู่

ในการใช้ Indirect Access สามารถอ้าง Memory ได้ทีละ 2 แบงก์ คือ 0 กับ 1 หรือ 2 กับ 3 โดย Register Bit ที่ใช้ในการควบคุมคือ STATUS Register บิต 7 (IRP) โดยที่ ถ้าบิตนี้เป็น

1 = Bank 2, 3 (100h – 1FFh)

0 = Bank 0, 1 (00h – FFh)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นต้องคอยระวังเวลาที่จะทำการอ้าง Memory ของ PIC ไม่ว่าจะเป็น Direct หรือ Indirect ก็ตาม เพราะโอกาสผิดพลาดเกิดขึ้นได้

2.1.9 ชุดคำสั่งของ PIC 16F877x

ไมโครชิป เทคโนโลยี ผู้ผลิตไมโครคอนโทรลเลอร์ตระกูล PIC ได้จัดแบ่งชุดคำสั่งของไมโครคอนโทรลเลอร์ PIC ออกเป็น 3 กลุ่มคือ

(1). กลุ่มคำสั่งจัดการข้อมูลระดับไบต์กับรีจิสเตอร์ไฟล์ (Byte-oriented File Register Operations) เป็นกลุมคำสั่งที่กระทำกับรีจิสเตอร์ไฟล์โดยตรงหรือโดยอ้อม โดยขนาดของข้อมูลอยู่ในระดับไบต์หรือ 8 บิต มีทั้งสิ้น 18 คำสั่ง

(2). กลุ่มคำสั่งจัดการข้อมูลระดับบิตกับรีจิสเตอร์ไฟล์ (Bit-oriented File Register Operations) เป็นกลุ่มคำสั่งที่ต้องเกี่ยวข้องหรือกระทำกับรีจิสเตอร์ไฟล์โดยตรงหรือโดยอ้อม โดยขนาดของข้อมูลที่ต้องประมวลผลอยู่ในระดับบิต มีทั้งสิ้น 4 คำสั่ง

(3). กลุ่มคำสั่งจัดการกับค่าคงที่และควบคุมการทำงาน (Literal and Control Operations) เป็นกลุ่มคำสั่งที่ผสมกันระหว่างการประมวลผลกับค่าคงที่, คำสั่งเกี่ยวกับการกระโดดไปยังโปรแกรมย่อยและออกจากโปรแกรมย่อย และคำสั่งกำหนดโหมดการทำงาน มีทั้งสิ้น 13 คำสั่ง

รูปแบบของข้อมูลคำสั่งทั้ง 14 บิตของไมโครคอนโทรลเลอร์จะแบ่งเป็น บิตของออปโค้ด (Opcode), บิตกำหนดตำแหน่งแอดเดรสของรีจิสเตอร์ และบิตเก็บค่าคงที่ ซึ่งจะแตกต่างกันไปในแต่ละกลุ่มคำสั่ง

2.1.10 การเข้าถึงรีจิสเตอร์และข้อมูลของไมโครคอนโทรลเลอร์ PIC

ไมโครคอนโทรลเลอร์ PIC16F87x มีกระบวนการเข้าถึงหน่วยความจำและรีจิสเตอร์ 4 แบบ คือ

2.1.10.1 การเข้าถึงแบบทันทีทันใด

จะเป็นการเข้าถึงค่าคงที่โดยตรง ด้วยการใช้ชุดคำสั่งที่เกี่ยวข้องกับค่าคงที่ (Literal) ร่วมกับรีจิสเตอร์ w หรือแอกคิวมูเลเตอร์ ตัวอย่างของคำสั่งที่แสดงให้เห็นถึงการเข้าถึงข้อมูลแบบทันทีทันใด ได้แก่ MOVLW k โดยที่ k คือค่าคงที่

2.1.10.2 การเข้าถึงข้อมูลแบบโดยตรง

เป็นการเข้าถึงข้อมูลหรือรีจิสเตอร์ด้วยการกำหนดแอดเดรสที่ต้องการเข้าถึงอย่างเจาะจง หรือระบุผ่านชื่อของรีจิสเตอร์ก็ได้ ดังตัวอย่าง

CLRF TEMP

เป็นคำสั่งที่ต้องการเคลียร์ค่าของรีจิสเตอร์ที่ชื่อว่า TEMP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

2.1.10.3 การเข้าถึงข้อมูลโดยอ้อม

ในกรณีที่ต้องติดต่อกับรีจิสเตอร์หลายๆตัวในคราวเดียวกัน ไมโครคอนโทรลเลอร์ PIC จะมีรีจิสเตอร์ FSR (File Select Register) และรีจิสเตอร์ INDF ช่วยในการติดต่อแบบโดยอ้อม โดยรีจิสเตอร์ FSR ทำหน้าที่เป็นตัวชี้แอดเดรสของหน่วยความจำที่ต้องการเข้าถึงแล้วอ่านข้อมูลนั้นๆ จากรีจิสเตอร์ INDF หรืออาจกล่าวได้ว่าใช้รีจิสเตอร์ FSR เก็บค่าแอดเดรสและรีจิสเตอร์ INDF เก็บค่าของข้อมูล

2.1.10.4 การเข้าถึงค่าข้อมูลแบบสัมพัทธ์

การเข้าถึงข้อมูลแบบนี้ประกอบด้วยการคำนวณค่าสัมพันธ์ของระยะห่างระหว่างแอดเดรสที่เริ่มต้นทำงานกับแอดเดรสที่ต้องการเข้าถึง โดยมีการใช้ค่าของโปรแกรมเคาน์เตอร์เข้ามาช่วย คำสั่งหนึ่งที่นำมาใช้ในการเข้าถึงข้อมูลสัมพั?ธ์นี้คือ RETLW

กระบวนการจะเริ่มต้นด้วยการกำหนดค่าออฟเซตลงในรีจิสเตอร์ w แล้วนำไปรวมกับค่าของโปรแกรมเคาน์เตอร์ก็จะได้ค่าของแอดเดรสของหน่วยความจำที่ต้องการเข้าถึง การเข้าถึงแบบนี้มีข้อจำกัดคือ จะต้องเข้าถึงข้อมูลเป็นช่วงๆ ช่วงละ 256 ตำแหน่ง

การเข้าถึงหน่วยความจำและรีจิสเตอร์ของไมโครคอนโทรลเลอร์ PIC เป็นเรื่องที่ผู้ใช้งานต้องให้ความสำคัญ เนื่องจากแต่ละรีจิสเตอร์มีวิธีการเข้าถึงที่แตกต่างกัน บางตัวสามารถเข้าถึงได้โดยตรง บางตัวต้องใช้วิธีการเข้าถึงทางอ้อม และในกรณีที่ต้องการเปิดตารางข้อมูล ควรใช้วิธีการเข้าถึงแบบสัมพัทธ์

2.2 พื้นฐานการรับส่งข้อมูล

การรับส่งข้อมูลในระบบคอมพิวเตอร์โดยทั่วไป จะหมายถึง การรับส่งข้อมูลเป็นจำนวนไบต์ๆให้กับอุปกรณ์ที่เกี่ยวข้องกับคอมพิวเตอร์ ซึ่งอาจแบ่งประเภทของการรับส่งข้อมูลได้ 2 แบบ

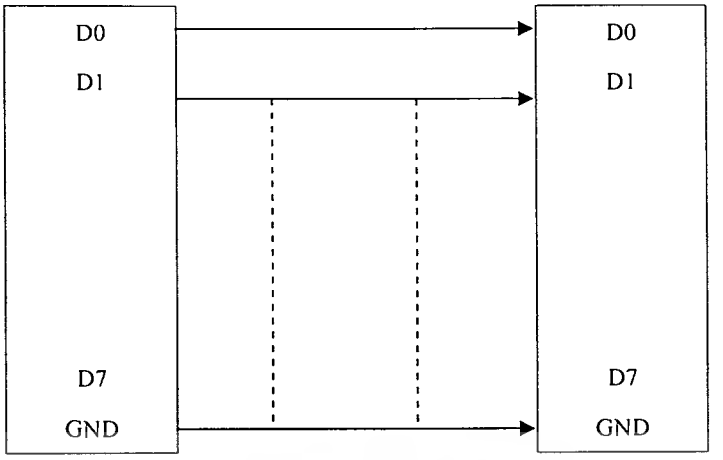
2.2.1 การรับส่งข้อมูลแบบอนุกรม (Serial)

การรับส่งข้อมูลแบบอนุกรม คือการรับส่งข้อมูลทีละบิตจนครบ 1 ไบต์ ถ้าต้องการส่งข้อมูล 1 ไบต์คือ $D_0 - D_7$ อาจจะส่งบิต D_0 ออกไปก่อนแล้วตามด้วย D_1 ไปเรื่อยๆจนถึง D_2

2.2.2 การรับส่งข้อมูลแบบขนาน (Parallel)

การรับส่งข้อมูลแบบขนานจะเป็นการรับส่งข้อมูลจำนวน 1 ไบต์ ออกไปทางพอร์ทในเวลาเดียวกัน ในระบบคอมพิวเตอร์ 1 ไบต์จะมีจำนวน 8 บิตคือ $D_0 - D_7$ ถ้ามีการส่งข้อมูลแบบขนานจะใช้สายสัญญาณอย่างน้อย 9 เส้น คือสาย Data 8 เส้น และสาย Ground 1 เส้น ดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
61724

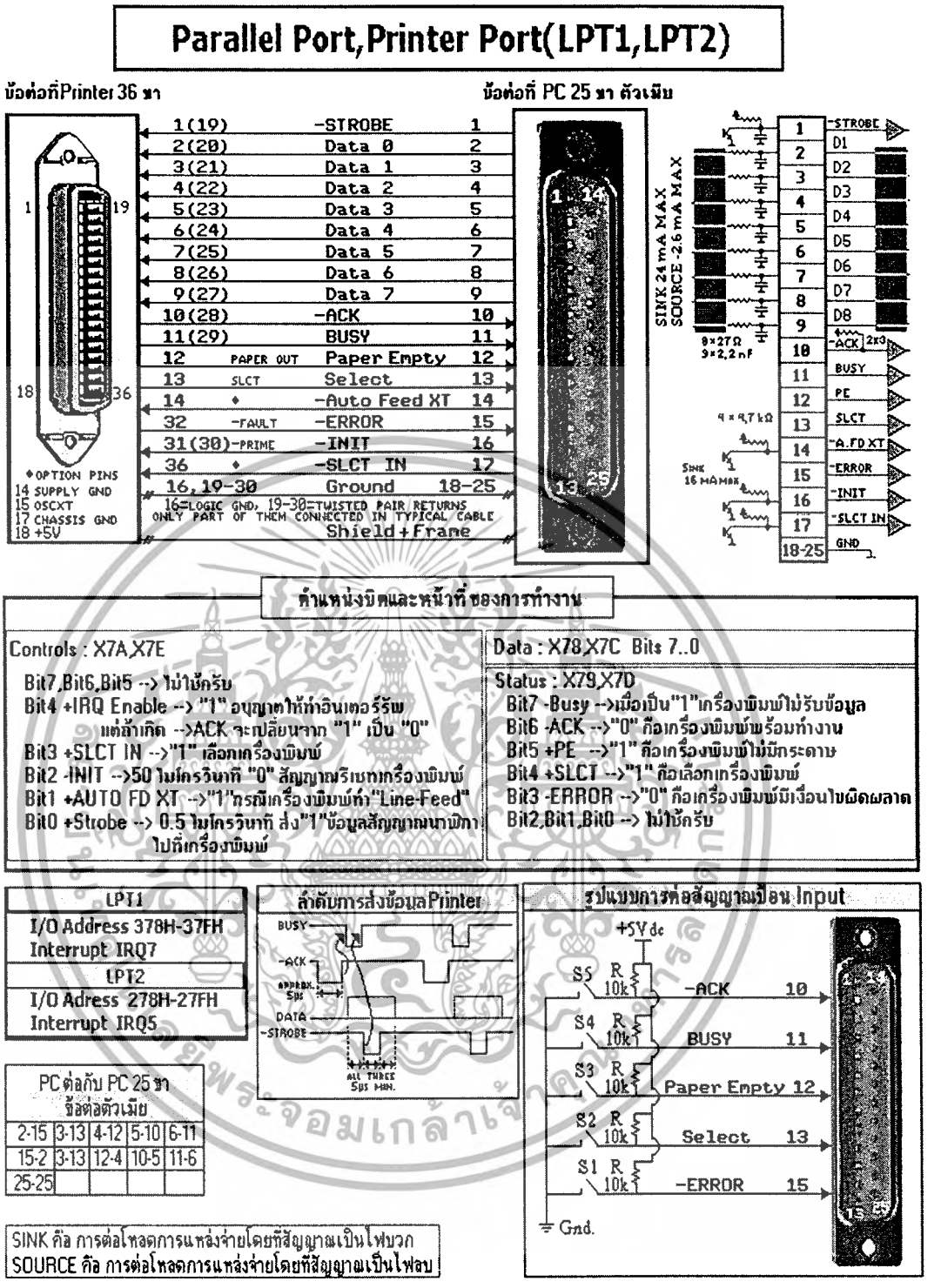


รูปที่ 2.10 การรับส่งข้อมูลแบบขนาน

2.2.3 ข้อแตกต่างระหว่างการสื่อสารแบบขนานและการสื่อสารแบบอนุกรม

1. การสื่อสารแบบขนาน เป็นการรับหรือส่งข้อมูลคราวละมากกว่า 1 บิตในเวลาเดียวกัน ทำให้การรับและส่งข้อมูลมีความเร็วสูง
   ส่วนการสื่อสารแบบอนุกรมจะเป็นการรับหรือส่งข้อมูลครั้งละ 1 บิต ทำให้ความเร็วในการสื่อสารลดลง
2. เนื่องจากการสื่อสารแบบขนาน เป็นการรับหรือส่งข้อมูลคราวละมากกว่า 1 บิต ในเวลาเดียวกัน ทำให้จำนวนของสายสัญญาณที่ใช้ในการส่งผ่านข้อมูลแบบขนานต้องมีมากตามจำนวนบิตของข้อมูลที่ทำการส่ง นอกจากนั้นยังมีสายที่ใช้สำหรับควบคุมและตรวจสอบการรับส่งข้อมูลด้วย ซึ่งอาจต้องใช้สายมากเป็น 2 เท่าของจำนวนบิทข้อมูลก็ได้
   ดังนั้นการสื่อสารแบบขนานจึงไม่เหมาะสมในการสื่อสารกับอุปกรณ์ภายนอกเป็นระยะทางไกลๆ เพราะจะทำให้สิ้นเปลืองค่าใช้จ่ายมาก
   การรับส่งข้อมูลแบบอนุกรม จึงมีข้อดีในเรื่องของจำนวนสายสัญญาณที่น้อยมากและไม่แปรตามจำนวนบิตของข้อมูล ซึ่งจะใช้สายอย่างน้อยเพียง 2 เส้น คือสายสัญญาณและสายกราวด์
3. โปรแกรมควบคุมของการรับส่งข้อมูลแบบอนุกรม จะมีความซับซ้อนมากกว่าการรับส่งข้อมูลแบบขนาน แต่สำหรับชิ้นงานในส่วนที่เป็น Memory ส่วนนี้ต้องการให้การรับและส่งข้อมูลมีความเร็วสูงและยังใช้ในระยะทางใกล้ๆ จึงเลือกใช้การสื่อสารแบบขนาน ถึงแม้ว่าจะต้องใช้สายสัญญาณที่ใช้ในการส่งผ่านข้อมูลจำนวนมากก็ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 รายละเอียดของ Printer Port หรือ Parallel Port

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการสร้าง

3.1 พอร์ตอินพุตเอาต์พุตของไมโครคอนโทรลเลอร์ PIC 16F87x

PIC16F87x มีพอร์ตให้ใช้งานตั้งแต่ 3-5 พอร์ต จำนวน 20-33 บิต ขึ้นอยู่กับเบอร์ของไมโครคอนโทรลเลอร์ และความสามารถในการจ่ายกระแสเอาต์พุตของขสพอร์ตที่ไฟเลี้ยง +5V คือ 25mA ต่อขาทั้งกระแสซิงก์และการะแสซอร์ซ ในขณะที่กระแสเอาต์พุตรวมของพอร์ต A, B และ E มีค่าสูงสุด 200mA ส่วนกระแสเอาต์พุตรวมของพอร์ต C และ D มีค่าสูงสุด 200mA ดังนั้นในการออกแบบเพื่อขับโหลดทางเอาต์พุตของขาพอร์ตต้องระวังเรื่องกระแสเอาต์พุตรวมที่ไมโครคอนโทรลเลอร์สามารถขับได้ด้วย

3.1.1 พอร์ต A

มีทั้งสิ้น 6 ช่องหรือ 6 บิต กำหนดชื่อขาเป็น RA0 – RA5 รีจิสเตอร์ที่ใช้ในการเก็บข้อมูลคือ PORTA มีแอดเดรสอยู่ที่ 0x05 (แบงก์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต แต่ใช้งานจริงเพียง 6 บิต ที่เหลือ 2 บิตต้องกำหนดให้เป็น “0” ส่วนการกำหนดทิสทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ TRISA ซึ่งมีแอดเดรสอยู่ที่ 0x85 (แบงก์ 1) มีขนาด 8 บิตและใช้เพียง 6 บิตเช่นกัน 2 บิตบนคือ บิต 6 และบิต 7 ต้องกำหนดให้เป็น “0” บิต 0 ของ TRISA ใช้กำหนดทิศทางของขาพอร์ต RA0 ไล่เรียงลำดับจนถึงบิต 5 ของ TRISA ใช้กำหนดทิศทางของขาพอร์ต RA5 หากต้องการกำหนดให้ขาพอร์ตในบิตใดเป็นอินพุตต้องเขียนข้อมูล “1” ไปยังบิตนั้น และในทางตรงข้ามหากต้องการกำหนดให้เป็นขาเอาต์พุตให้เขียนข้อมูล “0” ไปยังบิตนั้น

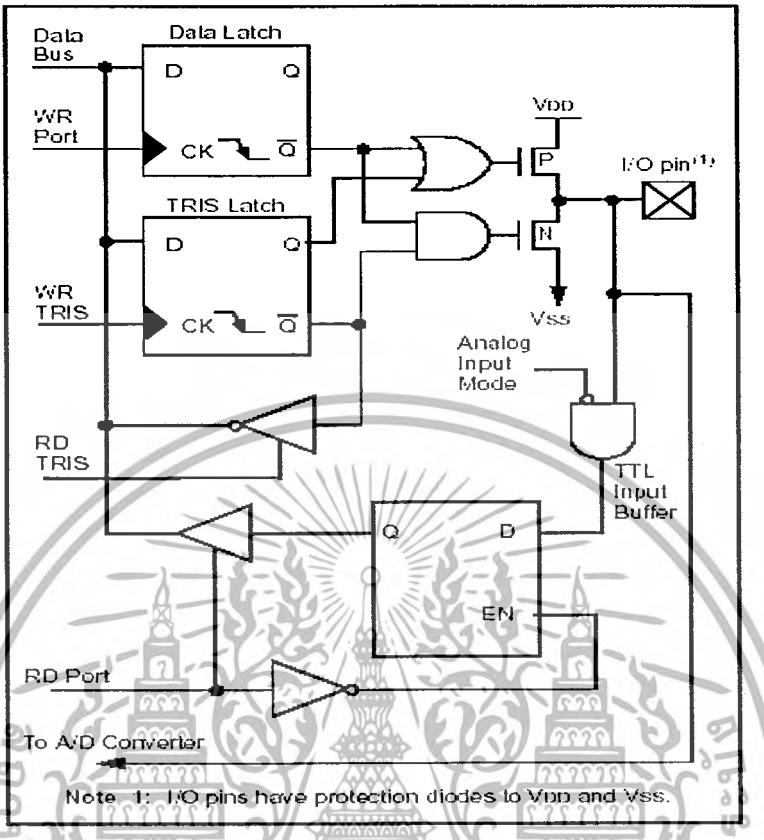
โครงสร้างทางฮาร์ดแวร์

พอร์ต A สามารถทำงานเป็นขาพอร์ตอินพุตเอาต์พุตปกติและเป็นขาอินพุตสัญญาณอนาลอกสำหรับวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลขนาด 10 บิตภายในไมโครคอนโทรลเลอร์ โดยขา RA0-RA3 และ RA5 จะมีการทำงานที่เหมือนกัน ส่วน RA4 จะแตกต่างตรงที่ขานี้นอกจากเป็นขาพอร์ตอินพุตเอาต์พุตปกติแล้ว ยังใช้เป็นขาอินพุตสำหรับไทเมอร์ 0 ภายในไมโครคอนโทรลเลอร์ด้วย และขา RA4 นี้ไม่สารถใช้งานเป็นขาอินพุตรับสัญญาณอนาลอกได้

ในรูปที่ 3.1แสดงไดอะแกรมของพอร์ต A ในบิต RA0-RA3 จะเห็นได้ว่าที่ขาพอร์ตจะมีแอนด์เกตทำหน้าที่เลือกลักษณะการทำงานของขาพอร์ตเมื่อเป็นขาอินพุต หากเลือกให้ขาพอร์ตนี้เป็นขาพอร์ตอินพุตดิจิตอล สัญญาณเลือกโหมดอินพุตอนาลอกจะต้องเป็นลอจิก “0” แต่ถ้าหากต้องการกำหนดให้เป็นขาอินพุตอนาลอก สัญญาณเลือกโหมดต้องเป็นลอจิก “1” สัญญาณอนาลอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ป้อนเข้ามายังขานี้ จะผ่านเข้าสู่วงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลหรือวงจรเปรียบเทียบแรงดันอนาลอก (เฉพาะในอนุกรม PIC 16F87xA) โดยตรง



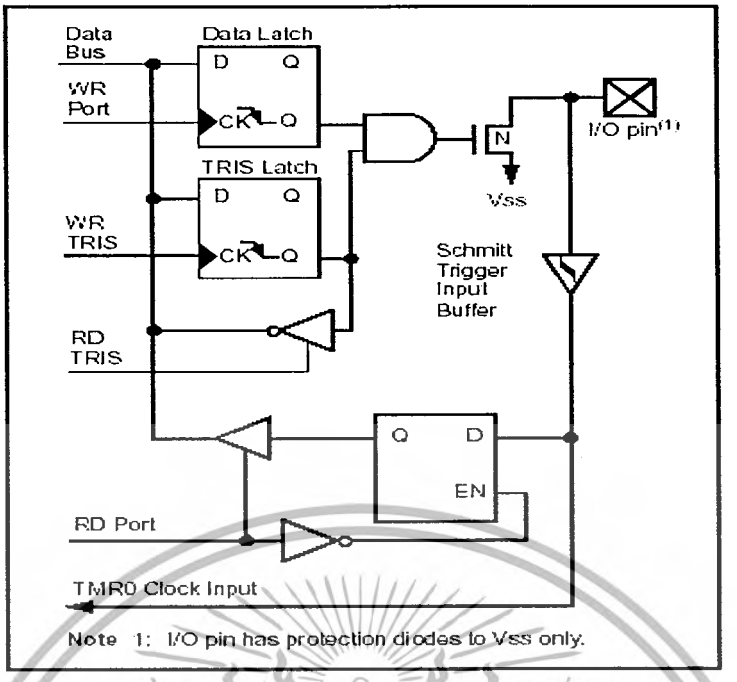
รูปที่ 3.1 บล็อกไดอะแกรมของขา RA3 : RA0 และ RA5

เมื่อขาพอร์ต RA0-RA3 ทำงาเป็นขาพอร์ตอินพุตดิจิตอล จะสามารถรับสัญญาณดิจิตอลระดับทีทีแอล (0-5V) ได้โดยตรง หากทำงานเป็นเอาต์พุตจะสามารถขับโหลดที่ต้องการกระแส 20mA ได้ หากนำมาขับ LED ต้องต่อตัวต้านทานจำกัดกระแส หรือถ้าใช้ไฟเลี้ยง 3V ก็จะขับ LED ได้โดยตรง

ในรูปที่ 3.2 เป็นไดอะแกรมของขาพอร์ต RA4/TOCKI โดยขานี้จะเป็นขาพอร์ตอินพุตเอาต์พุตปกติ และขาอินพุตรับสัญญาณนาฬิกาจากภายนอกของโมดูลไทเมอร์ 0วงจรอินพุตบัฟเฟอร์ที่ขาพอร์ตนี้เป็นแบบชมิตต์ทริกเกอร์ ทั้งนี้เพื่อจัดการให้สัญญาณอินพุตที่เข้ามามีความสมบูรณ์มากที่สุด และจะต้องต่อตัวต้านทานพูลอัปค่าประมาณ 4.7k-10k ที่ขานี้เมื่อใช้งานเป็นอินพุต

ส่วนพอร์ต RA5 ขานี้ก็มีความพิเศษไม่น้อย โดยนอกจากเป็นขาพอร์ตอินพุตเอาต์พุตดิจิตอลแล้วยังสามารถเลือกให้ทำงานเป็นขาพอร์ตอินพุตสัญญาณอนาลอกสำหรับวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลในช่อง 4 และเป็นขาเอาต์พุตของวงจรเปรียบเทียบแรงดันอนาลอกชุดที่ 2 ในไมโครคอนโทรลเลอร์อนุกรม PIC 16F87xA นอกจากนี้ยังใช้เป็นขา Slave Select (SS) ในกรณที่เลือกใช้การสื่อสารข้อมูลอนุกรมซิงโครนัสในแบบ SPI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 บล็อกไดอะแกรมของขา RA4/TOCKI

3.1.2 พอร์ต B

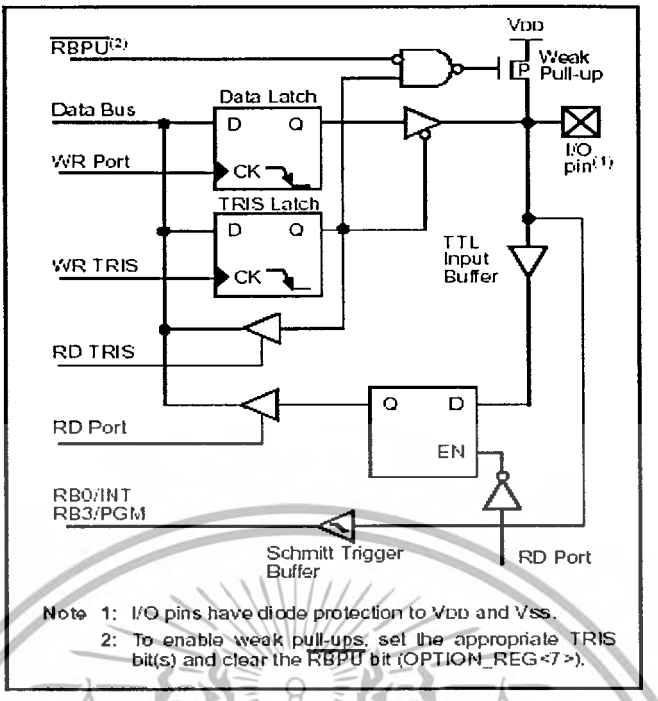
มี 8 บิต กำหนดชื่อขาเป็น RB0-RB7 รีจิสเตอร์ที่ใช้ในการเก็บข้อมูลคือ PORTB มีแอดเดรสอยู่ที่ 0x06 (แบงก์ 0) และ 0x106 (แบงก์ 2) เป็นรีจิสเตอร์ขนาด 8 บิต ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ TRISB ซึ่งมีแอดเดรสอยู่ที่ 0x86 (แบงก์ 1) และ 0x186 (แบงก์ 3) มีขนาด 8บิต เช่นเดียวกับพอร์ต A บิต 0 ของ TRISB ใช้กำหนดทิศทางของขาพอร์ต RB0 ไล่เรียงลำดับจนถึงบิต 7 ของ TRISB ใช้กำหนดทิศทางของขาพอร์ต RB7 หากต้องการกำหนดให้ขาพอร์ตในบิตใดเป็นอินพุตต้องเขียนข้อมูล "1" ไปยังบิตนั้น ในทางตรงกันข้ามหากต้องการกำหนดให้เป็นขาเอาต์พุตให้เขียนข้อมูล "0" ไปยังบิตนั้น

โครงสร้างทางฮาร์ดแวร์

พอร์ต B สามารถใช้งานในลักษณะต่างๆได้ 5 แบบคือ

1. เป็นขาพอร์ตอินพุตเอาต์พุตปกติ
2. เป็นขาอินพุตสัญญาณอินเตอร์รัปต์จากภายนอก โดยใช้ขา RB/INT
3. เป็นขาพอร์ตอินพุตสำหรับรับแรงดันโปรแกรมระดับต่ำ (Low Voltage Programming) โดยใช้ขา RB3/PGM
4. เป็นขาข้อมูลอนุกรมและสัญญาณนาฬิกาอนุกรมสำหรับการใช้งานโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ ซึ่งใช้ 2 ขา คือ RB7/PGD และ RB6/PGC
5. ใช้เป็นแหล่งกำเนิดสัญญาณอินเตอร์รัปต์แบบตรวจสอบการเปลี่ยนแปลงข้อมูลหรือระดับสัญญาณที่ขา RB4-RB7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 บล็อกไดอะแกรมของขา RB3 : RB0

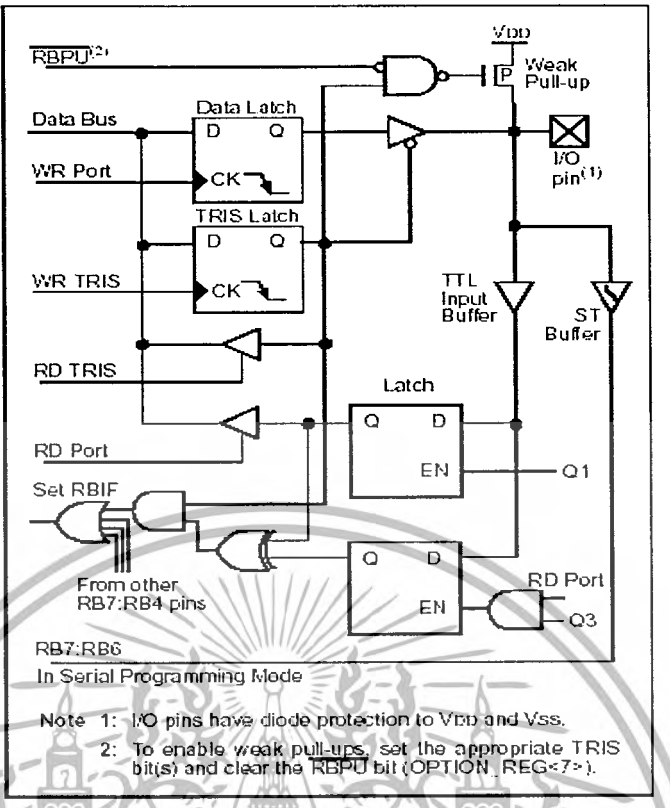
ในรูปที่ 3.3 แสดงไดอะแกรมของพอร์ต B ในบิต RB0-RB3 จะเห็นได้ว่าที่ขาพอร์ตจะมีวงจรพูลอัปแบบโปรแกรมได้ต่ออยู่ นั่นคือหากต้องการกำหนดให้เป็นขาอินพุต ต้องทำการเขียนข้อมูล "0" ไปยังบิต RBPU ในรีจิสเตอร์ OPTION_REG เพื่อเอ็นเอเบิลวงจรพูลอัพภายในขาพอร์ต B ในขณะที่หากกำหนดเป็นเอาต์พุต การพูลอัปที่ขาพอร์ต B นี้จะถูกยกเลิกอัตโนมัติ นอกจากนั้นการพูลอัปนี้จะได้รับการยกเลิกเมื่อเกิดพาวเวอร์ออนรีเซตขึ้น ในกรณีใช้ขา RB0/INT เพื่อรับสัญญาณอินเตอร์รัปต์จากภายนอก สัญญาณจะผ่านเข้าไปยังวงจรบัฟเฟอร์แบบชมิตต์ทริกเกอร์เพื่อให้สัญญาณที่ได้มีความแม่นยำและมีเสถียรภาพ

ในรูปที่ 3.4 เป็นไดอะแกรมของขาพอร์ต RB4-RB7 โดยขาพอร์ตในกลุ่มนี้มีความสามารถพิเศษพอสมควร โดยสามารถเลือกให้ทำงานเป็นขาพอร์ตอินพุตเอาต์พุตตามปกติ, ขาอินพุตรับแรงดันสำหรับการโปรแกรม (RB3), ขาสัญญาณสำหรับการโปรแกรม (RB6-RB7) และทำงานเป็นแหล่งกำเนิดสัญญาณอินเตอร์รัปต์ในแบบตรวจสอบการเปลี่ยนแปลงที่ขาพอร์ต RB4-RB7

วงจรบัฟเฟอร์ที่ขาพอร์ตนี้มีทั้งแบบทีทีแอลและชมิตต์ทริกเกอร์ ทั้งนี้เพื่อจัดการให้สัญญาณอินพุตที่เข้ามามีความเหมาะสมและสมบูรณ์มากที่สุด และยังคงสามารถรองรับการพูลอัปภายในแบบอัตโนมัติได้

ในกรณีที่มีการเอ็นเอเบิลการตอบสนองอินเตอร์รัปต์แบบตรวจสอบการเปลี่ยนแปลงลอจิกที่พอร์ต RB4-RB7 หากเกิดการอินเตอร์รัปต์ขึ้น บิต RBIF (บิต 0 ในรีจิสเตอร์ INTCON) จะเซตและหลังจากการตอบสนองการอินเตอร์รัปต์แล้ว ต้องเคลียร์บิต RBIF ด้วยกระบวนการทางซอฟต์แวร์เสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 บล็อกไดอะแกรมของขา RB7 : RB4

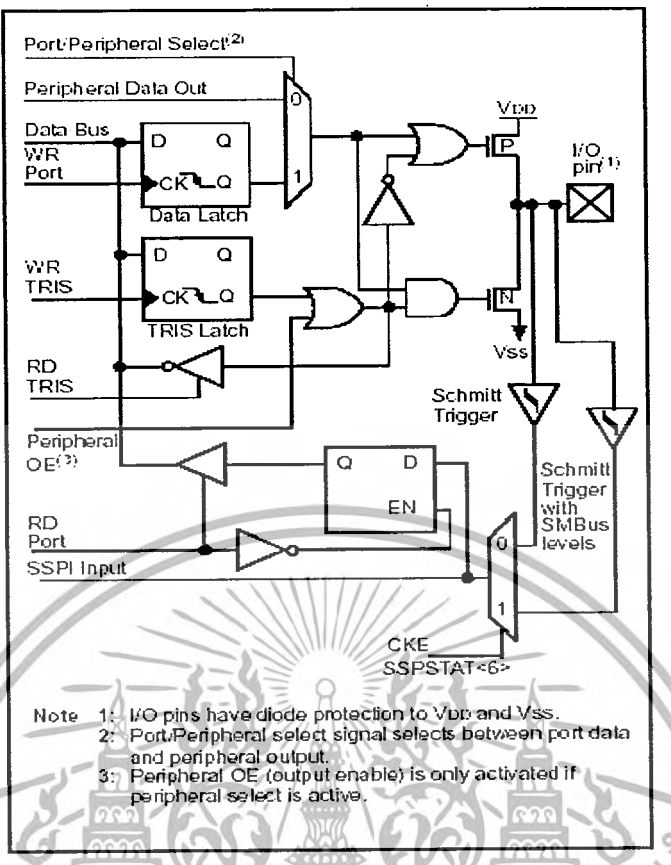
3.1.3 พอร์ต C

มีทั้งสิน 8 บิต กำหนดชื่อขาเป็น RC0-RC7 รีจิสเตอร์ที่ใช้เก็บข้อมูลคือ PORTC มีแอดเดรสอยู่ที่ 0x07 (แบงก์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ TRISC มีแอดเดรสอยู่ที่ 0x87 (แบงก์ 1) มีขนาด 8 บิต เช่นเดียวกับพอร์ต A และ B บิต 0 ของ TRISC ใช้กำหนดทิศทางของขาพอร์ต RC0 ไล่เรียงลำดับจนถึงบิต 7 ของ TRISC ใช้กำหนดทิศทางของขาพอร์ต RC7 หากต้องการกำหนดให้ขาพอร์ตในบิตใดเป็นอินพุตต้องเขียนข้อมูล "1" ไปยังบิตนั้น และในทางตรงข้ามหากต้องการกำหนดให้เป็นขาเอาต์พุต ให้เขียนข้อมูล "0" ไปยังบิตนั้น

โครงสร้างทางฮาร์ดแวร์

พอร์ต C สามารถใช้งานในลักษณะต่างๆได้หลายรูปแบบ และเป็นขาพอร์ตที่มีความสามารถสูงมาก ไม่ว่าจะเป็นขาพอร์ตอินพุตเอาต์พุตปกติ, ขาอินพุตเอาต์พุตออสซิลเลเตอร์ของโมดูลไทเมอร์1, ขาอินพุตสำหรับสัญญาณนาฬิกาของโมดูลไทเมอร์1, ขาเชื่อมต่อระบบบัส I^2C, ขาเชื่อมต่อแบบ SPI (Serial Peripheral Interface), ขาเชื่อมต่อพอร์ตอนุกรมแบบ USART, ขาอินพุตวงจรแคปเจอร์ (Capture) หรือวงจรตรวจจับสัญญาณขาเอาต์พุตของวงจรเปรียบเทียบสัญญาณ (Compare) และขาเอาต์พุตวงจร PWM(Pulse Width Modulation) หรืออาจกล่าวได้ว่าพอร์ต C เป็นพอร์ตสำหรับเชื่อมต่ออุปกรณ์ภายนอก(Peripheral Function Port) ที่มีความสมบูรณ์แบบมากที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



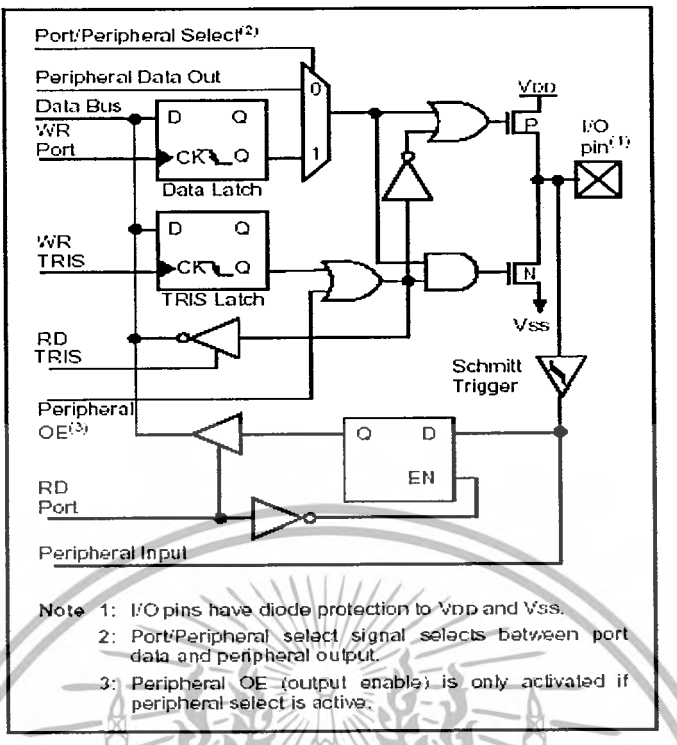
รูปที่ 3.5 บล็อกไดอะแกรมของขา RC<2:0> และ RC<7:5>

ในรูปที่ 3.5 แสดงไดอะแกรมของพอร์ต C ในบิต RC0-RC2 และ RC5-RC7 จะเห็นได้ว่ามีสัญญาณควบคุมการทำงานของขาพอร์ตมากมาย ทั้งนี้เนื่องจากขาพอร์ต C สามารถทำงานได้หลากหลายนั่นเอง สัญญาณควบคุมที่สำคัญคือ สัญญาณเลือกการทำงานระหว่างเป็นพอร์ตปกติหรือเป็นขาเชื่อมต่ออุปกรณ์พิเศษ (PORT/PERIPHERAL Select) และสัญญาณควบคุมการส่งข้อมูลของวงจรเชื่อมต่ออุปกรณ์(Peripheral Output Enable) สำหรับข้อมูลของวงจรเชื่อมต่ออุปกรณ์ที่ส่งออกและรับเข้ามาจะผ่านทางขาพอร์ตปกติ แต่เมื่อผ่านวงจรสำหรับเลือกสัญญาณข้อมูลแล้ว สายสัญญาณข้อมูลของพอร์ต (Data Bus) กับข้อมูลของวงจรเชื่อมต่ออุปกรณ์(Peripheral Output/Peripheral Input) จะแยกกัน

ในรูปที่ 3.6 เป็นไดอะแกรมของขาพอร์ต RC3-RC4 ทั้ง 2 ขานี้มีความพิเศาตรงที่สามารถใช้งานเป็นขาเชื่อมต่ออุปกรณ์อนุกรมแบบซิงโครนัส ซึ่งแบ่งเป็นระบบ SPI และระบบบัส I^2C จึงทำให้ต้องเพิ่มสายสัญญาณควบคุมอินพุตเข้ามาอีก 1 เส้น เพื่อเลือกสัญญาณอินพุตระหว่าง SPI และบัส

วงจรอินพุตบัฟเฟอร์ของขาพอร์ต C นี้เป็นแบบชมิตต์ทริกเกอร์ทั้งหมด ทั้งนี้เพื่อจัดการให้สัญญาณ อินพุตที่เข้ามามีความเหมาะสมและสมบูรณ์มากที่สุด และยังคงสามารถรับรองการพูลอัปภายในแบบอัตโนมัติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 บล็อกไดอะแกรมของขา RC<4:3>

3.1.4 พอร์ต D

มี 8 บิต กำหนดชื่อขาเป็น RD0-RD7 รีจิสเตอร์ที่ใช้ในการเก็บข้อมูลคือ PORTD มีแอดเดรสอยู่ที่ 0x08 (แบงก์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ TRISD มีแอดเดรสอยู่ที่ 0x88 (แบงก์ 1) มีขนาด 8 บิต หากต้องการกำหนดให้ขาพอร์ตในบิตใดเป็นอินพุตต้องเขียนข้อมูล "1" ไปยังบิตนั้น และในทางตรงข้ามหากต้องการกำหนดให้เป็นขาเอาต์พุต ให้เขียนข้อมูล "0" ไปยังบิตนั้น สำหรับพอร์ต D นี้จะมีเฉพาะไมโครคอนโทรลเลอร์ PIC รุ่น 40 ขาเท่านั้น

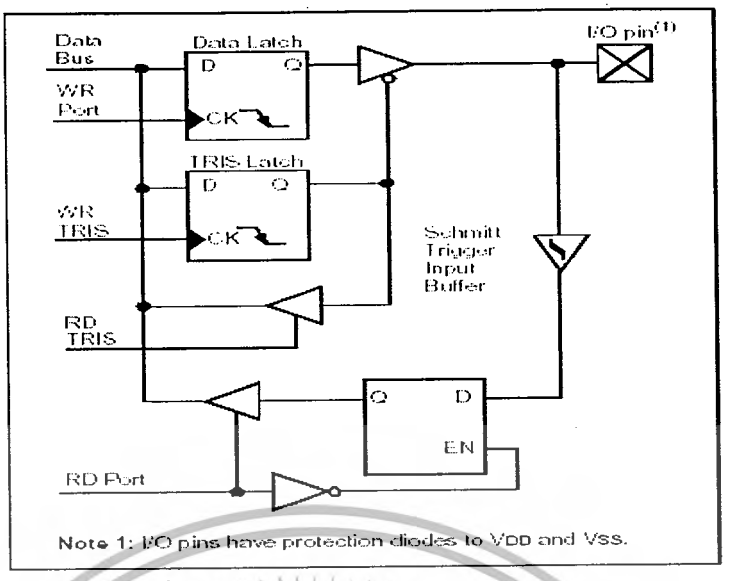
โครงสร้างทางฮาร์ดแวร์

พอร์ต D สามารถใช้งานได้ 2 โหมดคือ เป็นขาพอร์ตอินพุตเอาต์พุตปกติและเป็นส่วนขยายพอร์ตแบบขนาน (Parallel Slave Port : PSP) สำหรับเชื่อมต่อกับอุปกรณ์ภายนอกที่มีการจัดระบบบัสแบบไมโครโปรเซาเซอร์คือ มีสายข้อมูล 8 เส้น สายสัญญาณควบคุม 3 เส้นคือ สายสัญญาณควบคุมการอ่าน(RD : Read), เขียน (WR : Write) และเลือกอุปกรณ์(CS : Chip Select)

ในรูปที่ 3.7 แสดงไดอะแกรมของพอร์ต D ซึ่งมีโครงสร้างเหมือนกันทุกบิต เมื่อทำงานในโหมดพอร์ตอินพุตเอาต์พุตปกติ วงจรอินพุตจะเป็นแบบชมิตต์ทริกเกอร์ แต่เมื่อทำงานในโหมดขยายพอร์ตแบบขนานหรือ PSP วงจรอินพุตจะเปลี่ยนแปลงเป็นแบบทีทีแอล

การเลือกโหมดทำงานของพอร์ต D นี้ขึ้นกับบิต PSPMODE (บิต 4 ในรีจิสเตอร์ TRISE) ถ้าเป็น"0" เป็นการกำหนดให้พอร์ต D เป็นพอร์ตปกติ และถ้าเป็น "1" พอร์ต D จะทำงานในโหมดPSP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 บล็อกไดอะแกรมของพอร์ต D

3.1.5 พอร์ต E

มี 3 บิต กำหนดชื่อขาเป็น RE0-RE7 รีจิสเตอร์ที่ใช้ในการเก็บข้อมูลคือ PORTE มีแอดเดรสอยู่ที่ 0x09 (แบงก์ 0) เป็นรีจิสเตอร์ขนาด 8 บิต แต่ใช้งานเพียง 3 บิตล่าง คือ บิต0-บิต 2 เท่านั้นที่เหลือกำหนดให้เป็น "0" ส่วนการกำหนดทิศทางของพอร์ตนี้กระทำผ่านรีจิสเตอร์ TRISE ซึ่งมีแอดเดรสอยู่ที่ 0x89 (แบงก์ 1) มีขนาด 8 บิต โดยใช้ 3 บิตล่างในการกำหนดทิศทางของพอร์ต E ส่วนที่เหลือใช้ควบคุมการทำงานในโหมด PSP ของพอร์ต D

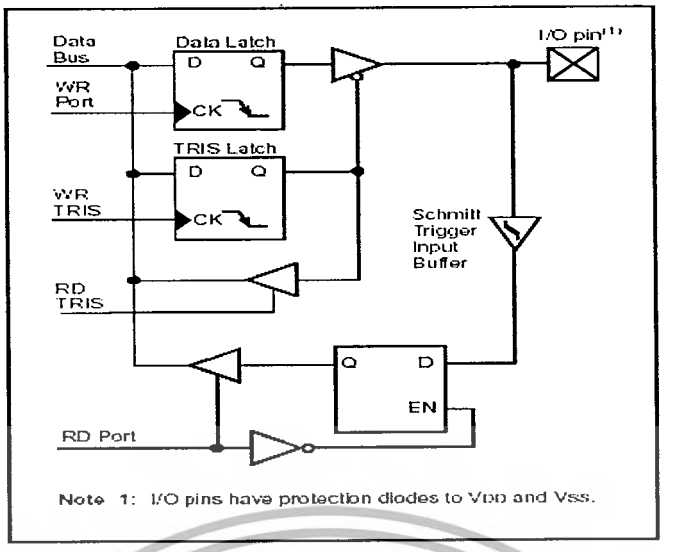
พอร์ต E สามารถใช้งานเป็นพอร์ตอินพุตเอาต์พุตปกติ, ขาอินพุตอนาลอกของโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล และขสควบคุมการติดต่อกับอุปกรณ์ภายนอกแบบ PSP ทั้งนี้ขึ้นอยู่กับการกำหนดข้อมูลของรีจิสเตอร์ที่ใช้ควบคุมการทำงานของพอร์ตนี่ เช่นเดียวกับพอร์ต D พอร์ต E จะมีเฉพาะไมโครคอนโทรลเลอร์ PIC รุ่น 40 ขาเท่านั้น

โครงสร้างทางฮาร์ดแวร์

ในรูปที่ 3.8 แสดงไดอะแกรมของพอร์ต E ในบิต RE0-RE2 เมื่อทำงานในโหมดพอร์ตอินพุตเอาต์พุตปกติ จะเห็นได้ว่ามีความคล้ายคลึงกับขาพอร์ตอื่นๆของ PIC16F87x ถ้าหากทำงานในโหมดพอร์ตอินพุตเอาต์พุตปกติ วงจรอินพุตของพอร์ตนี้จะเป็นแบบชมิตต์ทริกเกอร์ ในขณะที่หากทำงานในโหมดการแปลงสัญญาณอนาลอกเป็นดิจิตอล วงจรอินพุตจะเปลี่ยนเป็นแบบทีทีแอล

ถึงแม้ว่าพอร์ต E ใน PIC 16F87x มีจำนวนน้อยเพียง 3 บิต แต่สามารถเลือกรูปแบบการทำงานได้มากถึง 3 แบบ คือ เป็นพอร์ตอินพุตเอาต์พุตปกติ, อินพุตสำหรับวงจรแปลงสัญญาณอนาลอกเป็นดิจิตอลขนาด 10 บิตในไมโครคอนโทรลเลอร์ และพอร์ตสัญญาณควบคุมสำหรับติดต่อกับอุปกรณ์ในโหมด PSP ดังนั้นในการเลือกรูปแบบการทำงานต้องระมัดระวังเช่นเดียวกับพอร์ต C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 บล็อกไดอะแกรมของพอร์ต E

3.2 การเปลี่ยนอนาลอกเป็นดิจิตอล (ANALOG-TO-DIGITAL CONVERTER (A/D))

โมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอลสำหรับไมโครคอนโทรลเลอร์ PIC16F87x มีความละเอียด 10 บิต 5 ช่องอินพุตสำหรับรุ่น 28 ขา และ 8 ช่องอินพุตสำหรับรุ่น 40 ขา โดยขาพอร์ตที่ใช้งานร่วมด้วยคือขาพอร์ต RA0-RA3, RA5 และ RE0-RE2 (เฉพาะในรุ่น 40 ขา) การทำงานเพื่อรองรับอินพุตจำนวนมากจะใช้วิธีการมัลติเพล็กซ์ ซึ่งควบคุมด้วยกระบวนการทางซอฟต์แวร์ สำหรับการทำงานของโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอลใน PIC16F87x เป็นแบบซักเซสซีฟแอปพร็อกซิเมชั่น

3.2.1 รีจิสเตอร์ที่เกี่ยวข้อง

มีทั้งสิ้น 4 ตัว โดยแบ่งเป็นรีจิสเตอร์ควบคุมการทำงาน 2 ตัวคือ ADCON00 ซึ่งมีแอดเดรสอยู่ที่ 0x1F และ ADCON1 มีแอดเดรสอยู่ที่ 0x9F ส่วนอีก 2 ตัวคือรีจิสเตอร์ ADRESH และ ADRESL ซึ่งใช้ในการเก็บผลลัพธ์ของการแปลงสัญญาณโดย ADRESH มีแอดเดรสอยู่ที่ 0x1E ในแบงก์ 0 และ ADRESL มีแอดเดรสอยู่ที่ 0X9E ในแบงก์ 1 รีจิสเตอร์ทั้งสองมีขนาดตัวละ 8 บิต ต้องทำงานร่วมกันเพื่อรองรับข้อมูลดิจิตอล 10 บิตที่ได้จากการแปลงสัญญาณอนาลอก

- 3.2.1.1 รีจิสเตอร์ ADCON0

มีขนาด 8 บิต มีแอดเดรสอยู่ที่ 0x1F ในแบงก์ 0 เป็นรีจิสเตอร์หลักที่ใช้ในการควบคุมการทำงานของโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล รายละเอียดการใช้งานมีดังนี้

ADCON0 REGISTER (ADDRESS: 1Fh)

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/$\overline{\text{DONE}}$	—	ADON
bit 7							bit 0

ตารางที่ 3.1 หน่วยความจำควบคุมเกี่ยวกับ A/D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADCS1, ADCS0 (A/D Conversion Clock Select bits – บิต 7 และ 6): บิตเลือกความถี่ของสัญญาณนาฬิกาที่ใช้ในการแปลงสัญญาณอนาลอกเป็นดิจิตอล

"00" - ความถี่สัญญาณนาฬิกาหาร 2 (FOSC/2)

"01" - ความถี่สัญญาณนาฬิกาหาร 8 (FOSC/8)

"10" - ความถี่สัญญาณนาฬิกาหาร 32 (FOSC/32)

"11" - ความถี่สัญญาณนาฬิกาหารจากวงจร RC (FRC)

CHS, CHS1, CHS0(Analog Channel Select bits – บิต 5, 4 และ3): บิตเลือกช่องสัญญาณอนาลอก

"000" ช่อง 0 (AN0/RA0)

"001" ช่อง 1 (AN1/RA1)

"010" ช่อง 2 (AN2/RA2)

"011" ช่อง 3 (AN3/RA3)

"100" ช่อง 4 (AN4/RA4)

"101" ช่อง 5 (AN5/RA5) ไม่มีในรุ่น 28 ขา

"110" ช่อง 6 (AN6/RA6) ไม่มีในรุ่น 28 ขา

"111" ช่อง 7 (AN7/RA7) ไม่มีในรุ่น 28 ขา

GO/DONE (A/D Conversion Status bit – บิต 2): บิตแสดงสถานะการแปลงสัญญาณทำงานร่วมกับบิต ADON

กรณี บิต ADON เป็น "1"

"0" - การแปลงสัญญาณเสร็จสมบูรณ์ หรือยังไม่เริ่มการแปลงสัญญาณ

"1" – ยังอยู่ในระหว่างการแปลงสัญญาณ

บิตนี้สามารถเคลียร์ได้ด้วยกระบวนการทางฮาร์ดแวร์ 2 ลักษณะ คือ เมื่อการแปลงเสร็จสมบูรณ์จะเคลียร์เองอัตโนมัติ และเคลียร์เนื่องจากเกิดพาวเวอร์ออนรีเซต

บิต 1 ไม่ใช้งาน กำหนดเป็น "0"

ADON (A/D On bit – บิต 0): บิตเปิดการทำงานของโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล

"0" - ปิดการทำงาน

"1" - เปิดการทำงาน

- 3.2.1.2 รีจิสเตอร์ ADCON1

เป็นรีจิสเตอร์ควบคุมการทำงานของโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอลที่ต้องทำงานร่วมกับ ADCON0 มีขนาด 8 บิต มีแอดเดรสอยู่ที่ 0x9F ในแบงก์ 1 โดยรีจิสเตอร์ตัวนี้ใช้กำหนดการทำงานของขาพอร์ตที่เกี่ยวข้องกับโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล และใช้เลือกรูปแบบของข้อมูลผลลัพธ์ที่ได้จากการแปลงสัญญาณ มีรายละเอียดการทำงานดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADCON1 REGISTER (ADDRESS 9Fh)

U-0	U-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0
ADFM	—	—	—	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

ตารางที่ 3.2 แสดงบิต 7: ADFM

ADFM (A/D Result Format Select bit – บิต 7): บิตเลือกรูปแบบผลลัพธ์ของการแปลงสัญญาณ

"0" - เลือกผลลัพธ์แบบชิดซ้าย (Left Justified) มีรูปแบบดังนี้

หลังจากแปลงสัญญาณให้ด้านซ้ายของ ADRESH เป็น 0

ADRESH	ADRESL
000000xx	xxxxxxxx

ตารางที่ 3.3 แสดงการแปลงสัญญาณให้ด้านซ้ายของ ADRESH

"1" - เลือกผลลัพธ์แบบชิดซ้าย (Right Justified) มีรูปแบบดังนี้

หลังจากแปลงสัญญาณให้ด้านขวาของ ADRESL เป็น 0

ADRESH	ADRESL
xxxxxxxx	xx000000

ตารางที่ 3.4 แสดงการแปลงสัญญาณให้ด้านขวาของ ADRESH

บิต 5 และ 4 ไม่ใช้งาน กำหนดเป็น "0'

PCFG3, PCFG2, PCFG1, PCFG0 (A/D Port Configuration Control bits – บิต 3, 2, 1 และ 0): บิตกำหนดการทำงานของขาพอร์ตที่ใช้ในโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล

เมื่อการแปลง A/D เสร็จสิ้น ผลลัพธ์ของการแปลง A/D จะมีขนาด 10 บิต ซึ่งจะเก็บอยู่ในรีจิสเตอร์ 2 ตัวต่อกันคือ ADRESH:ADRESL ส่วนรีจิสเตอร์บิต GO/DONE/(ADCON0<2>)จะถูกเคลียร์และ ADIF จะถูกเซท (A/D Interrupt Flag)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PCFG3: PCFG0	AN7[1] RE2	AN6[1] RE1	AN5[1] RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF-	CHAN/ Refs[2]
0000	A	A	A	A	A	A	A	A	VDD	Vss	8/0
0001	A	A	A	A	VREF+	A	A	A	RA3	Vss	7/1
0010	D	D	D	A	A	A	A	A	VDD	Vss	5/0
0011	D	D	D	A	VREF+	A	A	A	RA3	Vss	4/1
0100	D	D	D	D	A	D	A	A	VDD	Vss	3/0
0101	D	D	D	D	VREF+	D	A	A	RA3	Vss	2/1
011x	D	D	D	D	D	D	D	D	VDD	Vss	0/0
1000	A	A	A	A	VREF+	VREF-	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	Vss	6/0
1010	D	D	A	A	VREF+	A	A	A	RA3	Vss	5/1
1011	D	D	A	A	VREF+	VREF-	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF+	VREF-	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF+	VREF-	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	Vss	1/0
1111	D	D	D	D	VREF+	VREF-	D	A	RA3	RA2	1/2

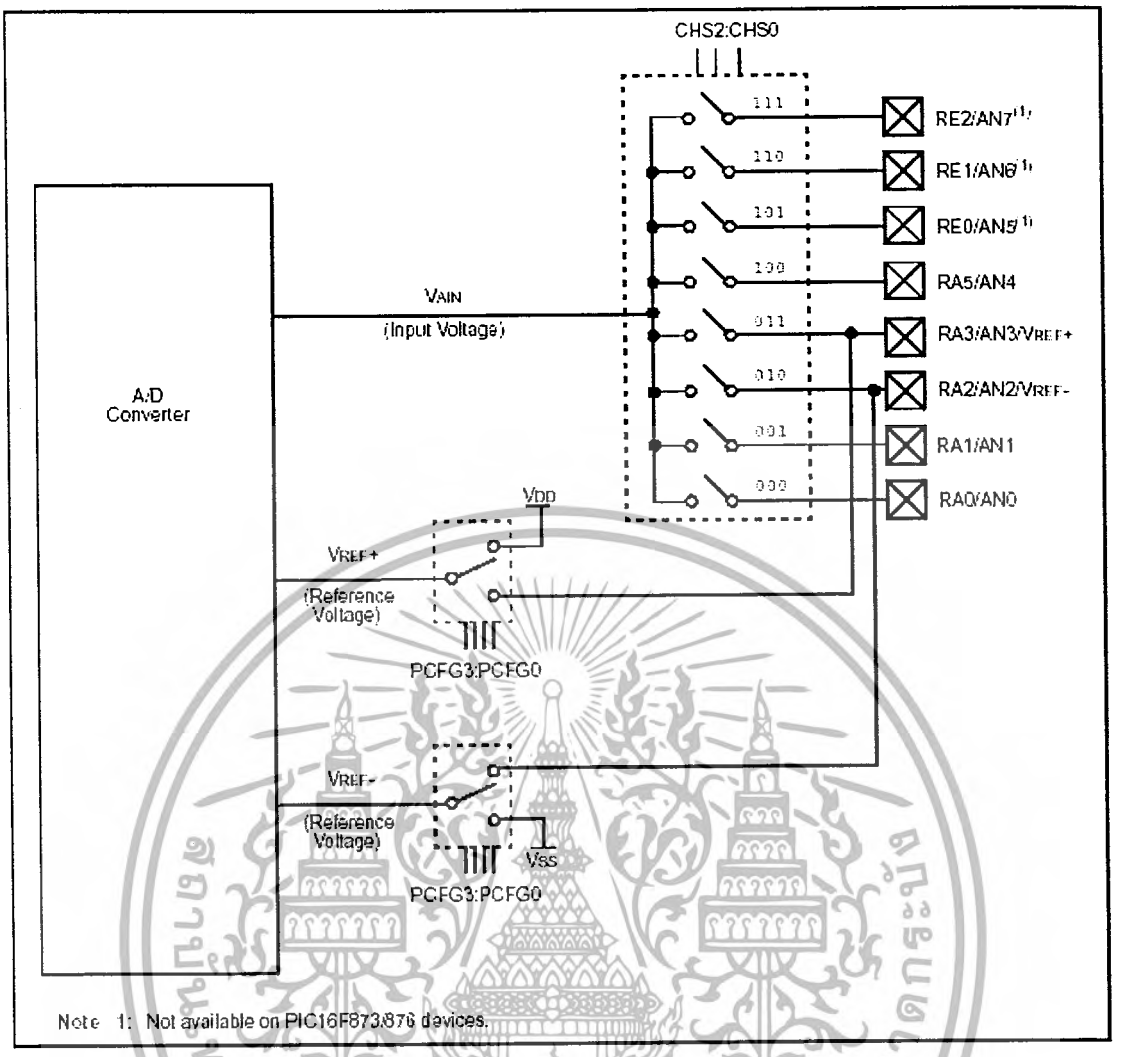
A = Analog input D = Digital I/O

ตารางที่ 3.5 แสดงส่วนการใช้ช่องสัญญาณกับพอร์ต

3.2.2 การทำงานของโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล

มีไดอะแกรมการทำงานแสดงดังรูปที่ 3.9 ขาพอร์ตที่ใช้เป็นอินพุตอนาลอกจะต้องได้รับการกำหนดมาจากบิต PCFG3:PCFG0 ในรีจิสเตอร์ ADCON1 ว่าจะให้ขาพอร์ตใดเป็นพอร์ตรับสัญญาณอนาลอก พอร์ตใดรับแรงดันอ้างอิง หรือพอร์ตใดเป็นพอร์ตดิจิตอล จากนั้นสัญญาณอนาลอกที่ต้องการแปลงจะถูกส่งผ่านวงจรเลือกช่องสัญญาณซึ่งควบคุมด้วยบิต CHS2:CHS0 ในรีจิสเตอร์ ADCON0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 ไดอะแกรมการทำงานของโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอลใน PIC16F87x

3.2.3 การเลือกความถี่สัญญาณนาฬิกาของการแปลงสัญญาณ

การเลือกความถี่สัญญาณนาฬิกาของการแปลงสัญญาณ นี้มีความสำคัญมาก เนื่องจากมีผลต่อค่าเวลารวมในการแปลงสัญญาณ โดยในโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอลของ PIC16F87x ต้องใช้เวลาแปลงสัญญาณ อนาลอกเป็นดิจิตอล 10 บิตเท่ากับ 12TAD โดยค่าเวลา T_{AD} คือค่าเลาที่ใช้ในการแปลงสัญญาณต่อบิต และค่าTAD ที่น้อยที่สุดที่ยังคงให้ผลการแปลงสัญญาณที่ถูกต้องคือ 1.6 ไมโครวินาที ดังนั้นการเลือกความถี่สัญญาณนาฬิกาต้องคำนึงถึงปัจจัยนี้ด้วย ในการเลือกสัญญาณความถี่ของ A/D สามารถเลือกได้ 4 แบบคือ

- 2Tosc
- 8Tosc
- 32Tosc
- Internal RC oscillator

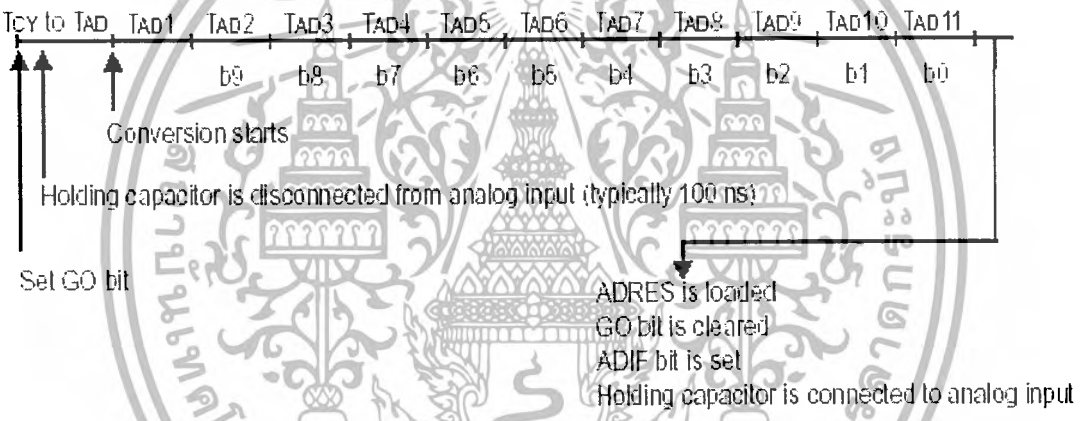
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งค่าเหล่านี้เราเลือกได้จากการเซตค่ารีจิสเตอร์ (ADCS1:ADCS0)

การเลือกความถี่ทำได้โดยกำหนดข้อมูลในบิต ADCS1 และ ADCS0 ในอนุกรม PIC16F87x มาตรฐานและสามารถใช้บิต ADCS2 เพิ่มในกรณีไมโครคอนโทรลเลอร์อนุกรม PIC16F87xA

3.2.4 กระบวนการแปลงสัญญาณอนาลอกเป็นดิจิตอล

เมื่อสั่งให้โมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอลทำงาน บิต GO/DONE ที่ใช้แสดงสถานะการทำงานจะเซท ถ้าหากมีการเคลียร์บิตนี้เกิดขึ้นในขณะที่การแปลงสัญญาณยังดำเนินอยู่ การแปลงสัญญาณจะถูกหยุดลงทันที แต่ข้อมูลของการแปลงสัญญาณที่ไม่สมบูรณ์จะไม่ได้รับการนำไปใช้งานนั่นคือถ้าแปลงไม่สมบูรณ์ค่าในรีจิสเตอรี ADRESH และ ADRESL จะยังเป็นค่าเดิมก่อนหน้านี้



รูปที่ 3.10 แสดงกลไกการแปลงสัญญาณอนาลอกเป็นดิจิตอลของไมโครคอนโทรลเลอร์ PIC16F87x

3.2.5 ขั้นตอนการใช้งานโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล

1. กำหนดค่าพารามิเตอร์เบื้องต้นต่างๆ
   1.1 กำหนดให้ขาพอร์ตที่ใช้งานเป็นอินพุต โดยเซตในรีจิสเตอร์ TRISA หรือ TRISE
   1.2 เลือกจำนวนช่องและแหล่งจ่ายแรงดันอ้างอิงที่บิต PCFG3:PCFG0 ในรีจิสเตอร์ ADCON1
   1.3 เลือกรูปแบบผลลัพธ์จากบิต ADFM ในรีจิสเตอร์ ADCON1
   1.4 เลือกช่องสัญญาณผ่านทางรีจิสเตอร์ ADCON1 และ ADCON2
   1.5 เลือกสัญญาณนาฬิกาจากข้อมูลที่บิต ADCS1:ADCS0 ในรีจิสเตอร์ ADCON0
   1.6 เอ็นเอเบิลโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล โดยเซตบิต ADON ในรีจิสเตอร์ ADCON0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ถ้ามีการใช้งานอินเตอร์รัปต์ให้กำหนดค่าดังต่อไปนี้ (ถ้าไม่ใช้การอินเตอร์รัปต์ข้ามไปข้อ 3)

   2.1 เคลียร์บิต ADIF ในรีจิสเตอร์ PIR1

   2.2 เซตบิต ADIE ในรีจิสเตอร์ PIE1

   2.3 เอ็นเอเบิลการอินเตอร์รัปต์โดยเซตบิต GIE ในรีจิสเตอร์ INTCON

3. หน่วงเวลาเพื่อให้พ้นช่วงเวลาแอกควิซิชั่น (Acquisition Time)

4. เริ่มต้นการแปลงสัญญาณ ด้วยการเซตบิต GO/DONE ในรีจิสเตอร์ ADCON0 (ห้ามเซตบิตนี้ในคำสั่งเดียวกับที่เซตบิต ADON เพราะจะทำให้เกิดความผิดพลาดในการแปลงสัญญาณได้)

5. หน่วงเวลาเพื่อรอให้จบกระบวนการแปลงสัญญาณ ซึ่งทำได้โดย

   5.1 วนรอค่าที่บิต GO/DONE ถ้าเป็น "0" หมายถึง เสร็จสิ้นการแปลงสัญญาณ

   5.2 รอให้เกิดอินเตอร์รัปต์ขึ้น

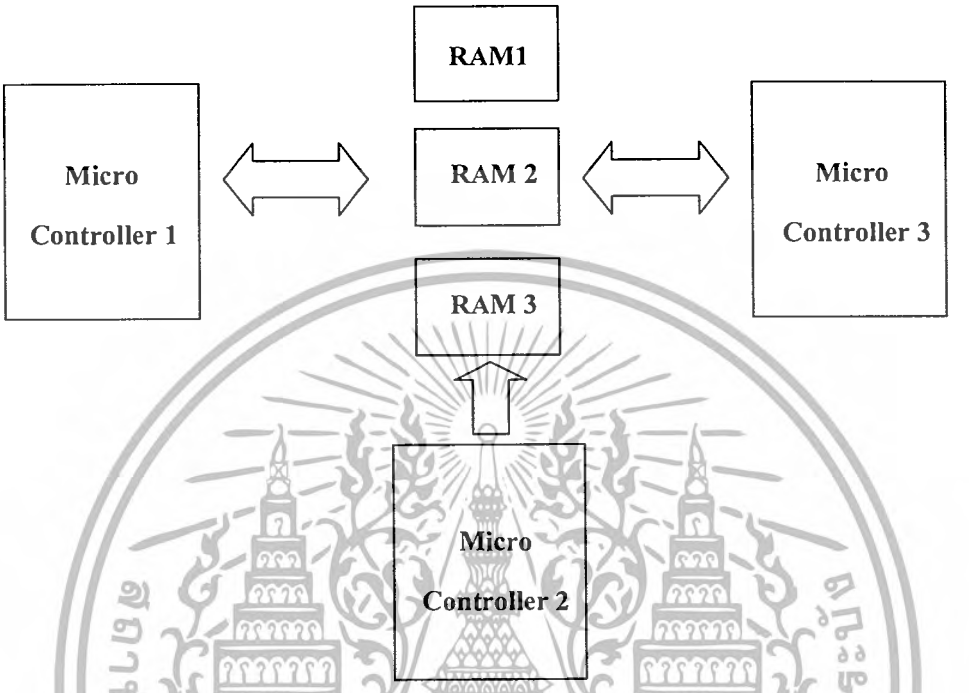
6. อ่านผลลัพธ์จากรีจิสเตอร์ ADRESH:ADRESL

7. ถ้าต้องการให้เกิดการแปลงสัญญาณอย่างต่อเนื่อง ให้หน่วงเวลาอย่างน้อย $2T_{AD}$ แล้วกลับไปทำในข้อ 3 (ถ้ามีการอินเตอร์รัปต์ ให้กลับไปทำข้อ 2.1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง



รูปที่4.1 รูประบบ

4.1 รายละเอียดวงจร

วงจรนี้เป็นการทดลองจำลองระบบการเก็บข้อมูลของออสซิโลส โคปแบบเก็บค่าได้ โดยจะทำการจำลองสัญญาณสี่เหลี่ยมเข้ามาในระบบ เพื่อนำไปจัดเก็บ เพื่อที่จะนำไปทำการปรับเปลี่ยนรูปสัญญาณก่อนนำไปแสดงผลบนหน้าจอซีอาร์ที ซึ่งการทำงานจะแบ่งออกเป็น 2 โหมดหลักคือ

1. โหมดส่งข้อมูลต่อเนื่องปกติ
2. โหมดจับภาพนิ่ง

โดยการทำงานในทั้งสองโหมดนี้ จะถูกควบคุมด้วยไมโครคอนโทรลเลอร์ 2 ที่เป็นทั้งตัวควบคุมและจัดลำดับการส่งผ่านข้อมูล และการเก็บข้อมูลทั้งหมด รวมทั้งยังเป็นหน่วยประมวลผลสัญญาณในตัวเองอีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1 โหมดส่งข้อมูลต่อเนื่องปกติ

ข้อมูลที่วัดได้จะถูกแปลงค่าแล้วส่งจากไมโครคอนโทรลเลอร์ 1 ไปเก็บไว้ในหน่วยความจำ RAM 1 ก่อนจะถูกส่งผ่านไปยังไมโครคอนโทรลเลอร์ 3 โดยที่ข้อมูลจะไม่ต้องผ่านการประมวลผลด้วยไมโครคอนโทรลเลอร์ 2 เลย เพียงแต่แค่เป็นการส่งผ่านเท่านั้น เพื่อที่จะนำข้อมูลไปประมวลผลต่อไปให้เร็วที่สุดและมีความต่อเนื่องมากที่สุด

4.1.2 โหมดจับภาพนิ่ง

ข้อมูลที่วัดได้จะถูกแปลงค่าแล้วส่งไมโครคอนโทรลเลอร์1 ไปเก็บไว้ยังหน่วยความจำ RAM 1 ก่อนที่จะ ถูกนำไปประมวลผลตามคำสั่งต่างๆตามการสั่งงานของผู้ใช้งานด้วยไมโครคอนโทรลเลอร์ 2 ไม่ว่าจะเป็น การขยายภาพของสัญญาณ ณ. เวลานั้น หรือการเลื่อนตำแหน่งการจับภาพนิ่ง ก่อนที่ข้อมูลที่ได้ผ่านการประมวลผลแล้ว จะถูกนำไปเก็บไว้ในหน่วยความจำ RAM 2 และ RAM 3 แล้วจึงค่อยทำการจัดลำดับการส่งข้อมูลผ่านไปยังไมโครคอนโทรลเลอร์ 3 เพื่อที่จะนำข้อมูลไปประมวลผลหรือนำไปแสดงผลต่อไป

4.1.3 ปฏิบัติการเบื้องต้นในโหมดจับภาพนิ่ง

1. เลื่อนภาพ ขึ้น ลง ซ้าย ขวา
2. ขยายภาพเข้า 2x
3. จับภาพนิ่งล่าสุดมาประมวลผล

4.1.4 ฮาร์ดแวร์ที่ใช้ในระบบ

1. ไมโครคอนโทรลเลอร์ เบอร์ PIC16F877
2. หน่วยความจำประเภท SRAM ขนาด 128 kb x 8 เบอร์ K6T1008C2E
3. Data Transparent SN74HC245N
4. Data Latch เบอร์ 74HC573
5. Multiplexer 3-8 เบอร์ SN74HC138

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.5 การสร้างวงจร

การต่ออุปกรณ์เข้ากับไมโครคอนโทรลเลอร์ 1

ADDRESS RAM (17 บิต จึงใช้ Data Latch เพื่อลดจำนวนขาที่ใช้)

ใช้ PORT B และ RC4 (บิตที่ 17)

DATA RAM (8 บิต)

ใช้ PORT D

RAM 1 Control (OE, WE, CS)

ใช้ PORT E (RE0, RE1, RE2)

CCP (2 บิต ติดต่อกับ ไมโครคอนโทรลเลอร์ 2)

ใช้ RC1 – RC2 (ขา CCP1 และ CCP2)

การต่ออุปกรณ์เข้ากับไมโครคอนโทรลเลอร์ 2

RAM 2, RAM 3 Control (OE , WE(Ram2) , WE(Ram3) , CS)

ใช้ RA5 และ PORT E (RE0, RE1, RE2)

ADDRESS RAM (17 บิต 2 ชุด จึงใช้ Data Latch เพื่อลดจำนวนขาที่ใช้)

ใช้ PORT B , PORT D และ RA4 ,RC4

CCP (4 บิต ติดต่อระหว่าง ไมโครคอนโทรลเลอร์ด้วยกัน)

ใช้ RC0 – RC3 (ขา CCP1 และ CCP2)

Data Transparent Control (ใช้เพื่อควบคุมทิศทางและจังหวะการส่งสัญญาณของวงจร)

ใช้ RC5 –Rc7 ส่งสัญญาณไปยัง IC MUX3-8 เพื่อขยายเป็น 8 เส้น

การต่ออุปกรณ์เข้ากับไมโครคอนโทรลเลอร์ 3

ADDRESS RAM (17 บิต จึงใช้ Data Latch เพื่อลดจำนวนขาที่ใช้)

ใช้ PORT B และ RC4 (บิตที่ 17)

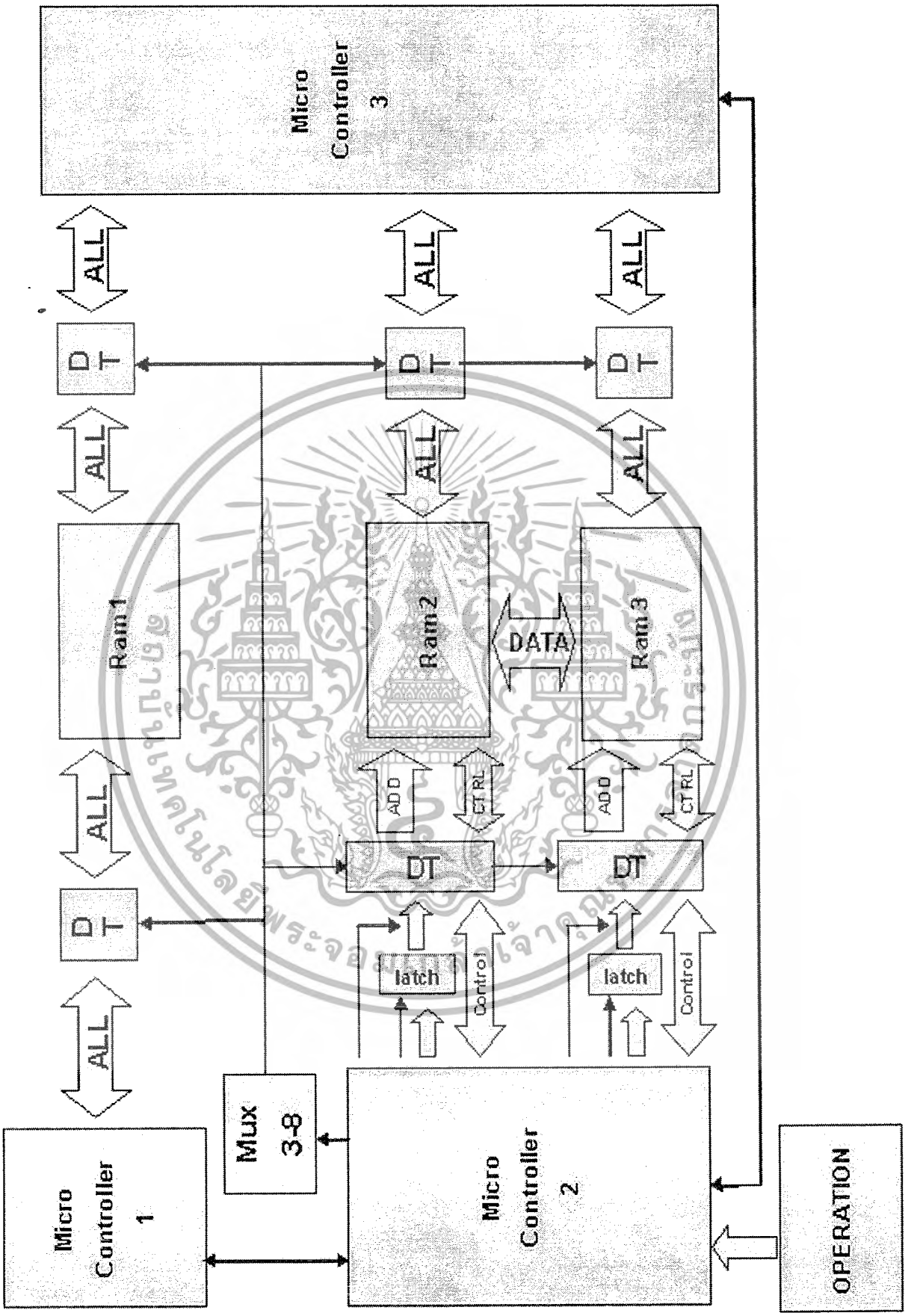
DATA RAM (8 Bit) ใช้ PORT D

RAM 1 Control (OE, WE, CS) ใช้ PORT E (RE0, RE1, RE2)

CCP (2 Bit ติดต่อกับ ไมโครคอนโทรลเลอร์ 2)

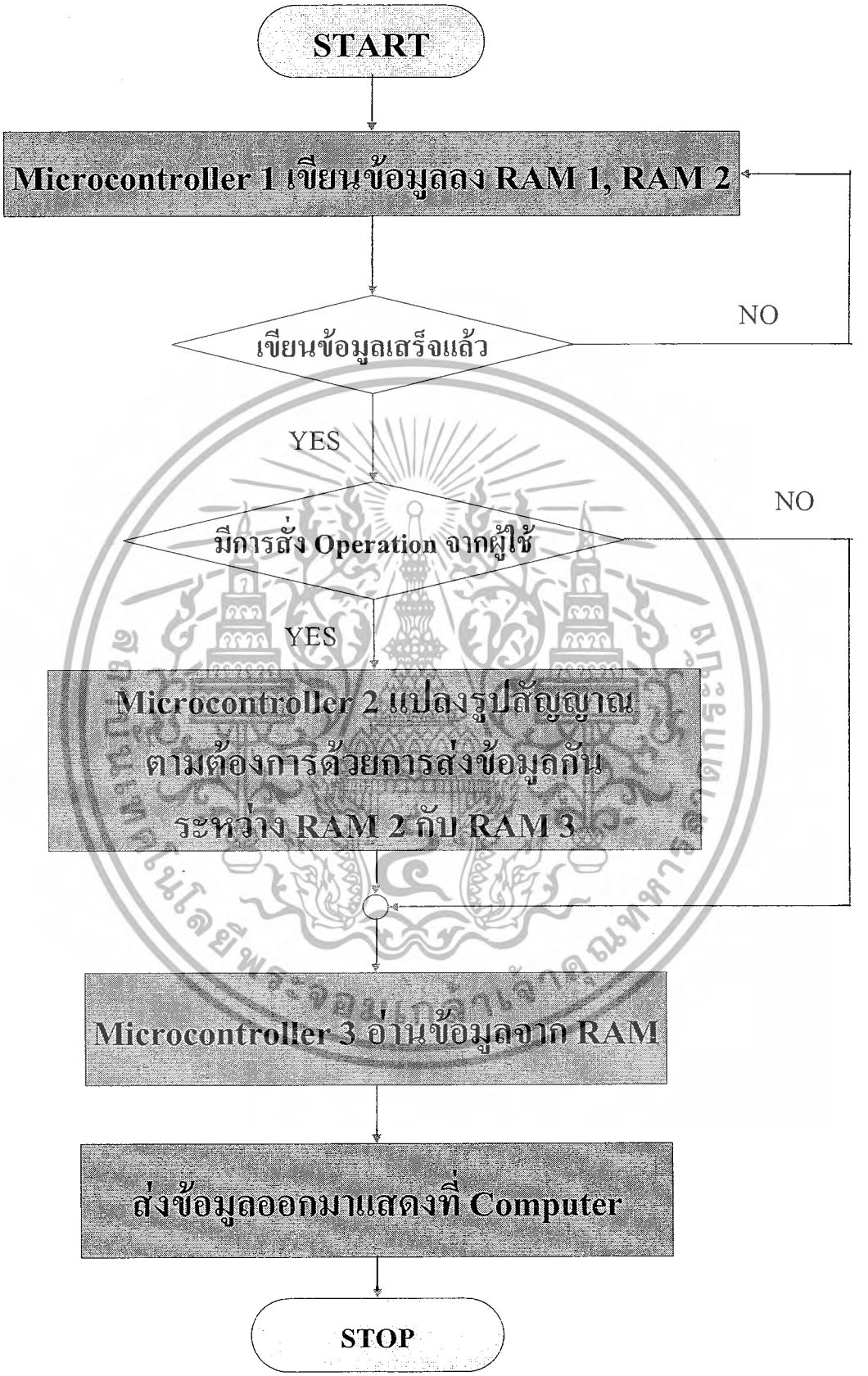
ใช้ RC1 – RC2 (ขา CCP1 และ CCP2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



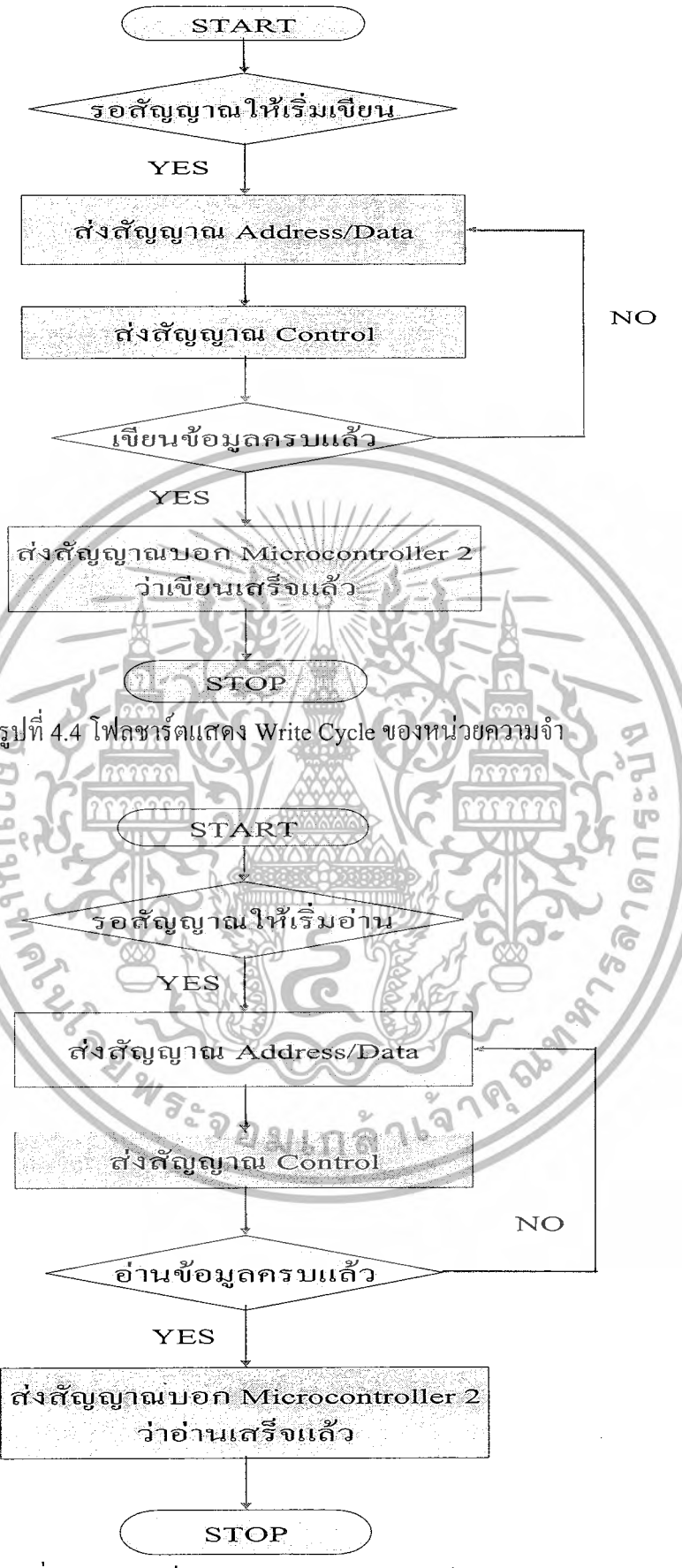
รูปที่ 4.2 บล็อกไดอะแกรมของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 โฟลชาร์ตแสดงการทำงานของวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 โฟลชาร์ตแสดง Write Cycle ของหน่วยความจำ

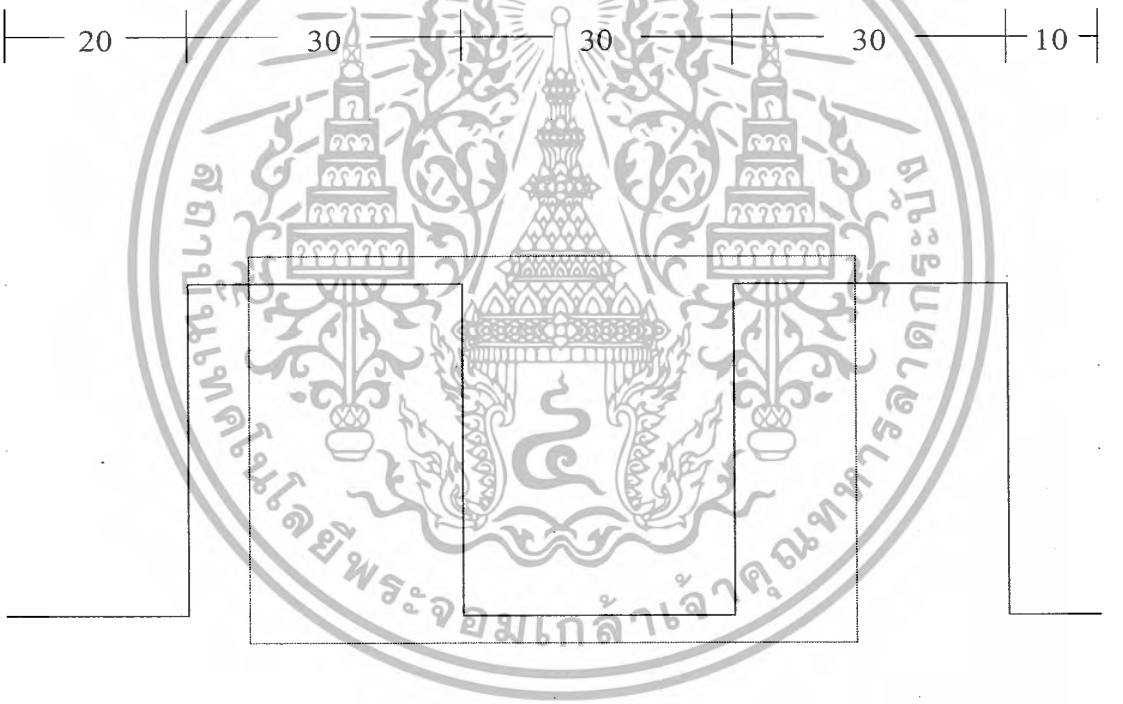
รูปที่ 4.5 โฟลชาร์ตแสดง Read Cycle ของหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.6 หลักการขยายสัญญาณ (Zoom)

จะใช้การคัดลอกค่าข้อมูลที่ RAM 2 เพียงแค่ที่ 1 แอดเดรส ไปยัง RAM 3 ทั้งหมด 4 แอดเดรส

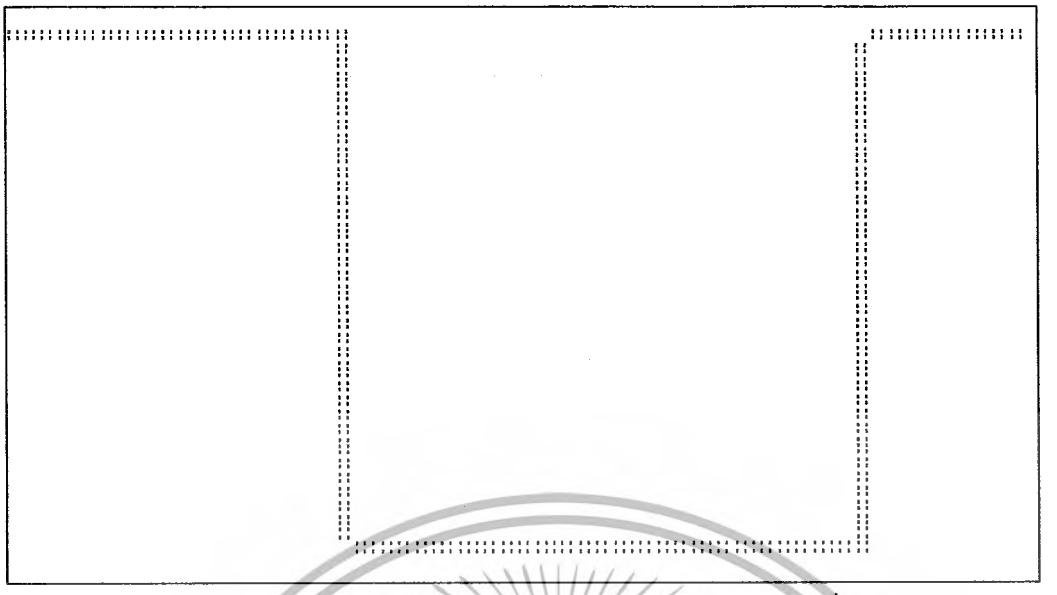
รูปที่ 4.6 แสดงการขยายสัญญาณโดยการคัดลอกข้อมูลที่ 1 แอดเดรส ไป 4 แอดเดรส



รูปที่ 4.7 แสดงการขยายสัญญาณบริเวณกลางจอ

จากรูปที่ 4.6 แสดงรูปการขยายสัญญาณ ซึ่งจะทำการขยายสัญญาณรูปสี่เหลี่ยมบริเวณตรงกลางจอโดยความละเอียดหน้าจอ กำหนดให้เป็น 120 x 80 จุด และจะได้ผลลัพธ์เป็นรูปสัญญาณที่ขยายแล้วดังรูปที่ 4.7

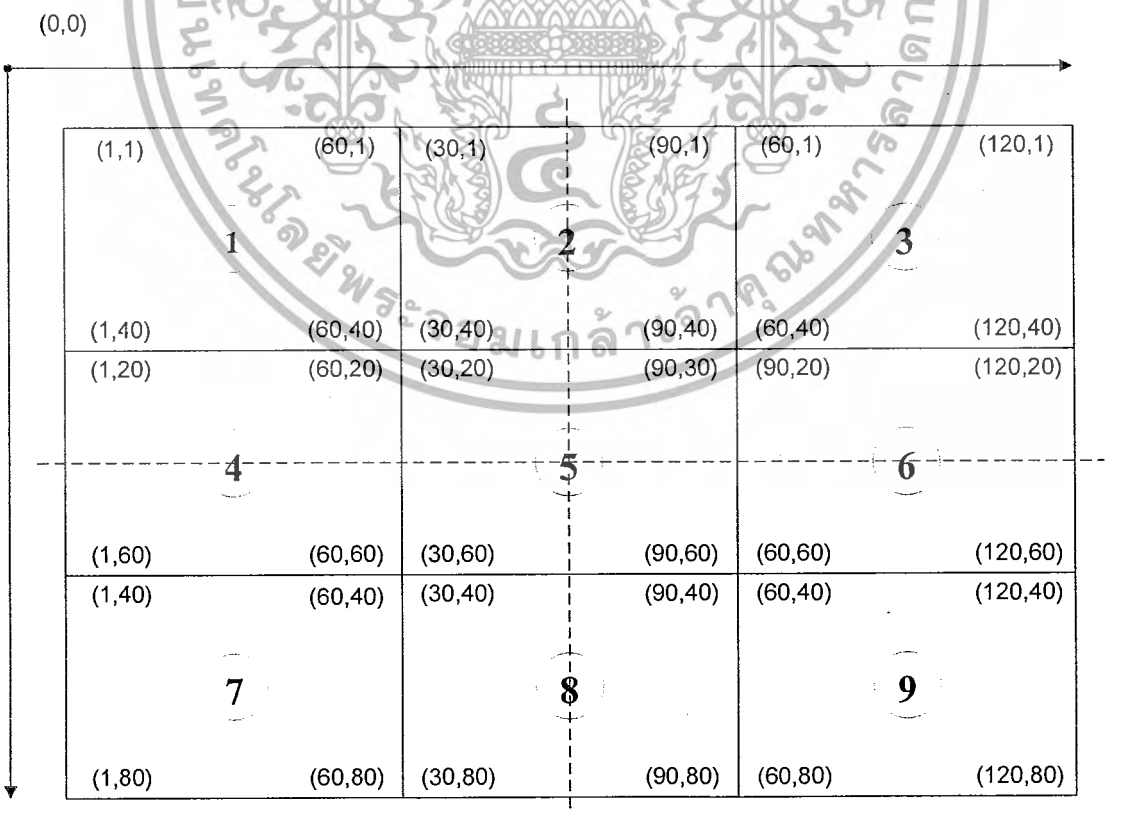
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 แสดงผลจากการขยายสัญญาณ

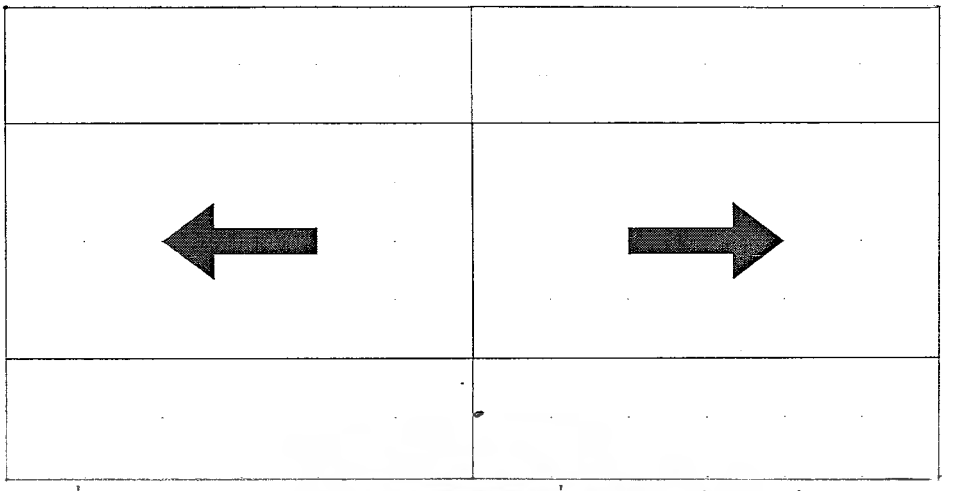
4.1.7 หลักการเลื่อนสัญญาณ (Sliding)

ใช้หลักการเดียวกับการขยายสัญญาณ แต่กำหนดจุดเริ่มต้นของการขยายไปยังตำแหน่งต่างๆตามที่แบ่งไว้ดังรูป 4.8

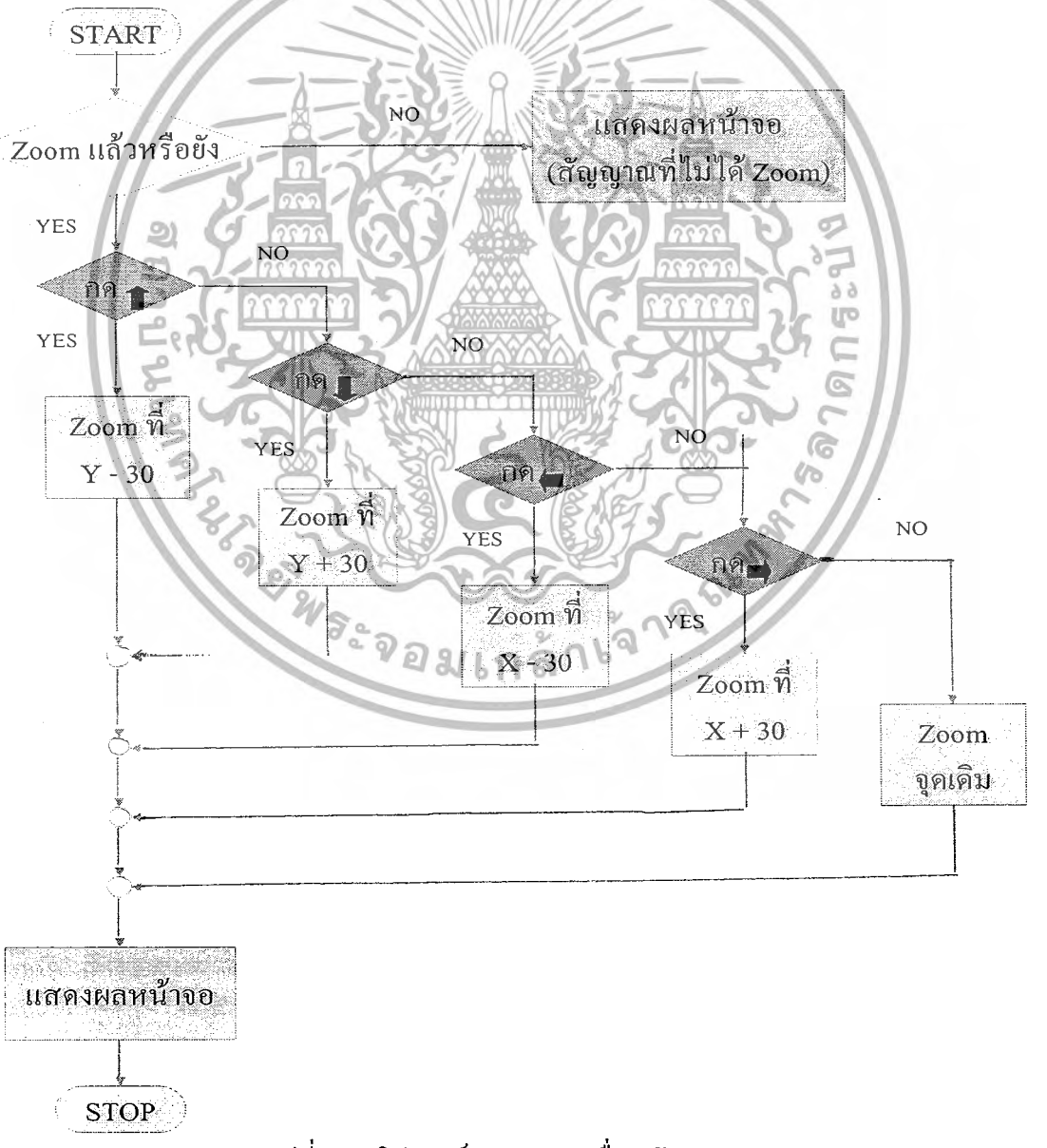


รูปที่ 4.9 แสดงตำแหน่งพิกัดต่างๆบนหน้าจอทั้ง 9 ส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



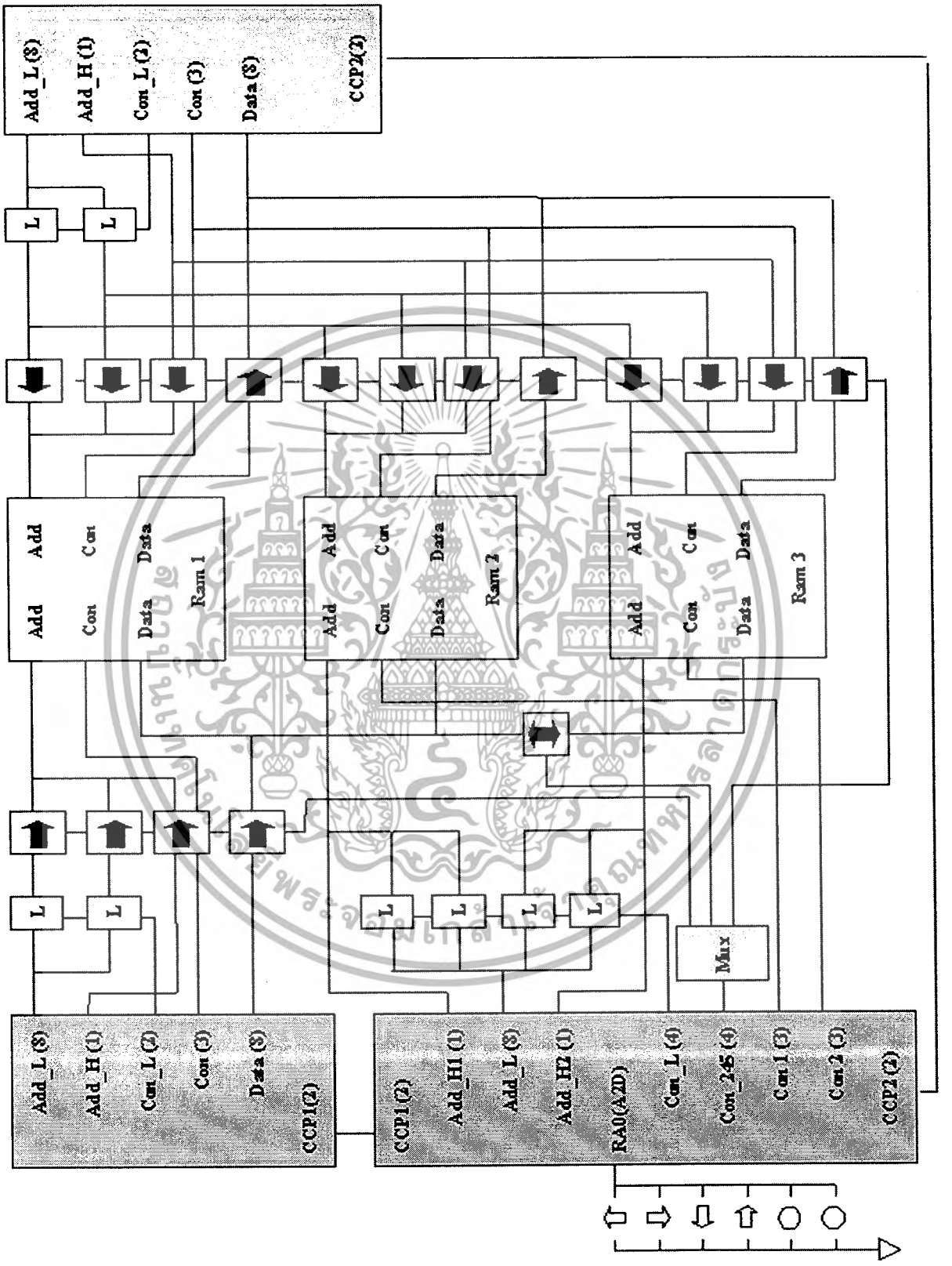
รูปที่ 4.10 แสดงขอบเขตการแสดงผลของการเลื่อนสัญญาณไปทางซ้ายและขวา



รูปที่ 4.11 โฟลชาร์ตแสดงการเลื่อนสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.8 วงจร



รูปที่ 4.12 แสดงรายละเอียดวงจรจัดการหน่วยความจำสำหรับแสดงผลสัญญาณบนจอ CRT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

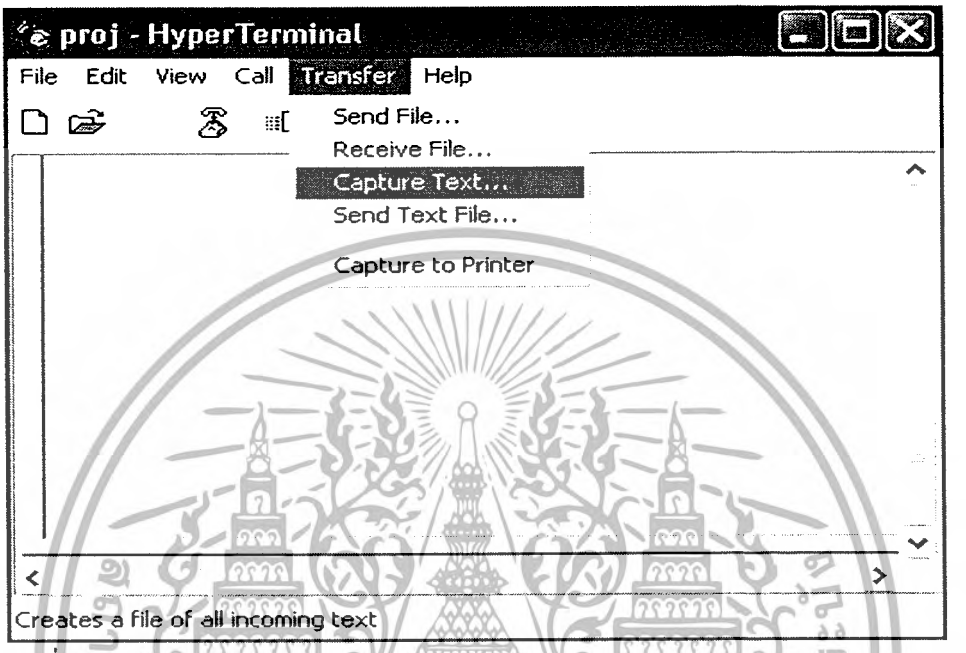
L	Data Latch ใช้สร้างสัญญาณ Address
⇧ ⇩ ⇦ ⇨ ○ ○	ปุ่มควบคุมการประมวลภาพ ให้เลื่อนขึ้นลงซ้ายขวา หรือขยายเข้า-ออกรูปสัญญาณ
Mux	Decoder ใช้สร้างสัญญาณควบคุมการทำงานของ 74LS245
⬅ ⬍	74LS245 ใช้เพื่อควบคุมทิศทางและจังหวะการส่งสัญญาณทั้งหมดของวงจร

ตารางที่ 4.1 แสดงรายละเอียดของสัญลักษณ์ต่างๆในวงจร

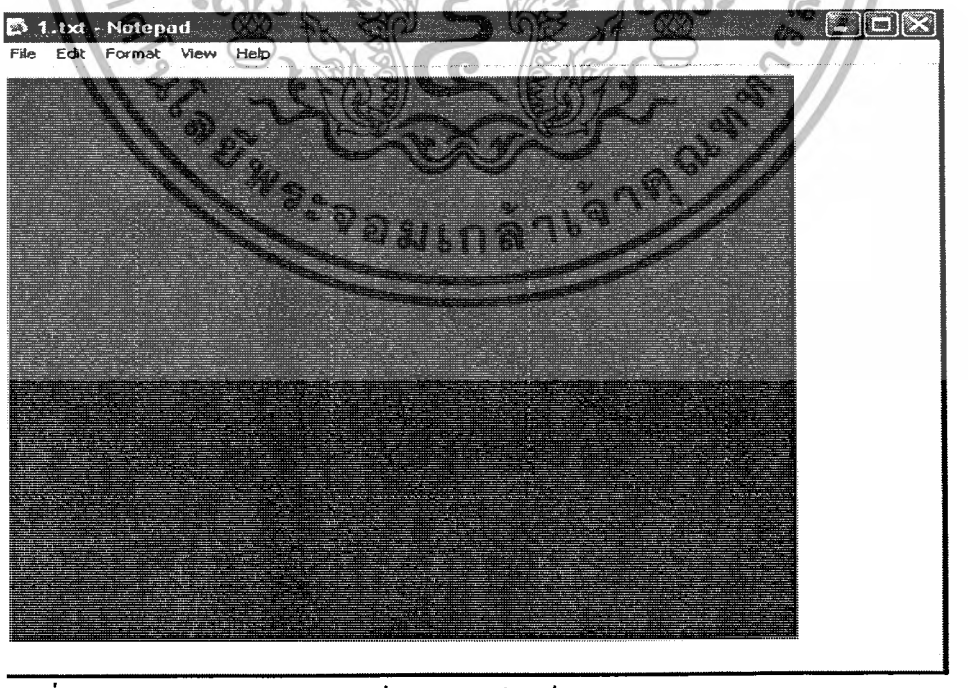
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดลอง

ผลการทดลองเป็นการนำข้อมูลที่ระบบส่งออกมาทางไมโครคอนโทรลเลอร์3 ผ่านทางสายอนุกรมมายังคอมพิวเตอร์ แล้วจึงใช้โปรแกรม Hyper Terminal จับค่า (Capture Text) ของสัญญาณออกมาซึ่งจะเป็นไฟล์ *.txt



รูปที่ 4.13 การใช้โปรแกรม Hyper Terminal จับค่า (Capture Text) ของสัญญาณ



รูปที่ 4.14 จับค่าสัญญาณออกมาเป็น TEXT ไฟล์

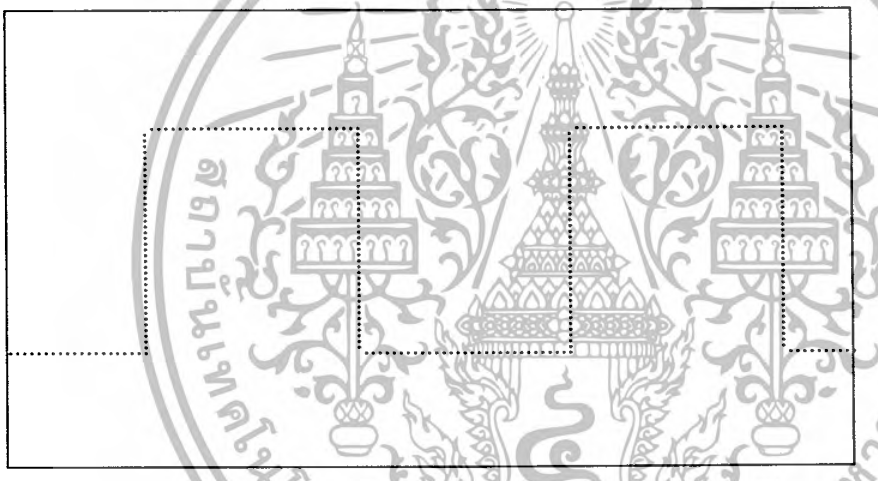
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทดลองนั้น จะทำการแบ่งหน้าจอออกเป็น 9 ส่วน เพื่อทำการเลื่อนตำแหน่งภาพหลังขยายขณะจับภาพนิ่งโดยความละเอียดหน้าจอ กำหนดให้เป็น 120 x 80 จุด

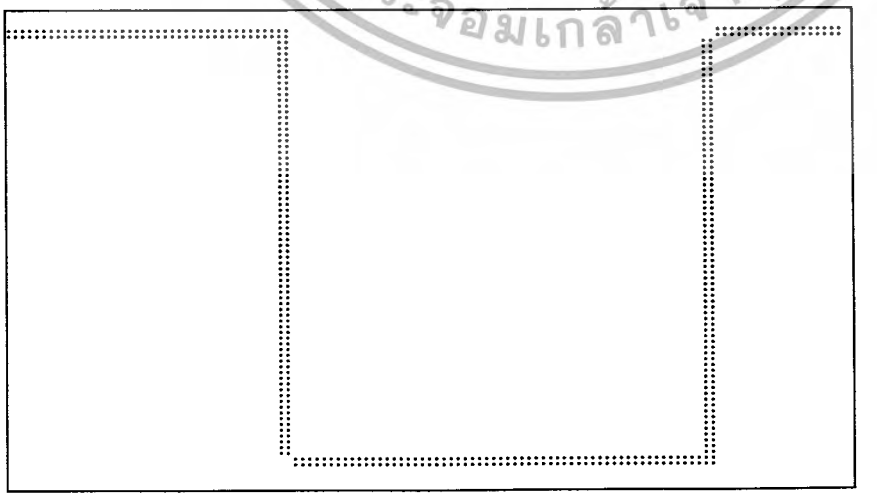
1	2	3
4	5	6
7	8	9

รูปที่ 4.15 การแบ่งช่องหน้าจอออกเป็น 9 ส่วน

การทดลองที่ 1 สัญญาณรูปสี่เหลี่ยม

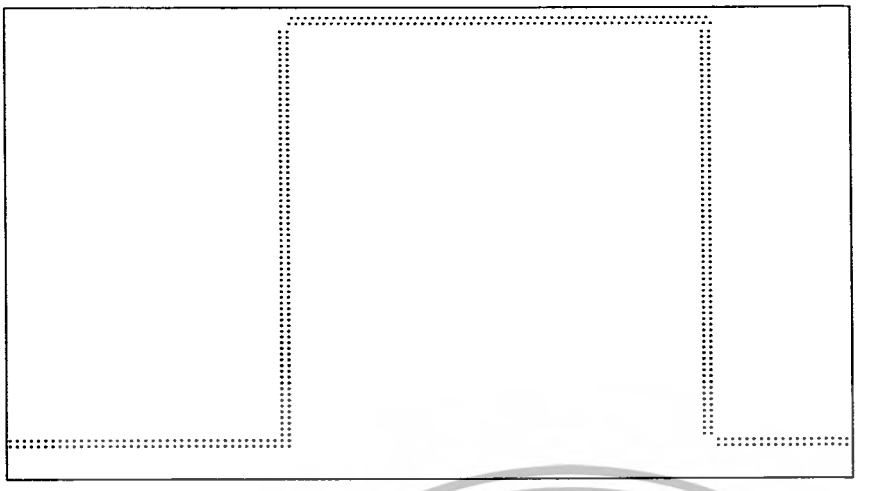


รูปที่ 4.16 ภาพผลการทดลอง ขณะยังไม่ทำการขยาย (1x)



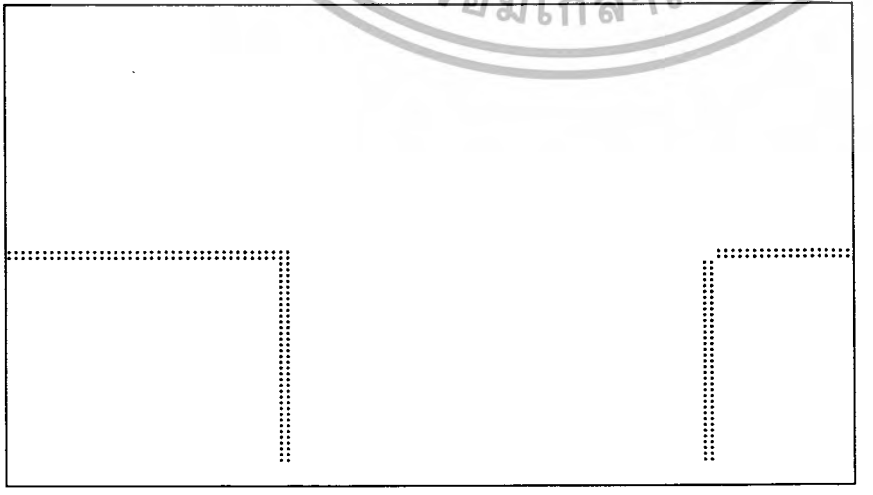
รูปที่ 4.17 ภาพผลการทดลอง หลังทำการขยาย (2x) ตำแหน่งที่ 5 (กลางจอ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



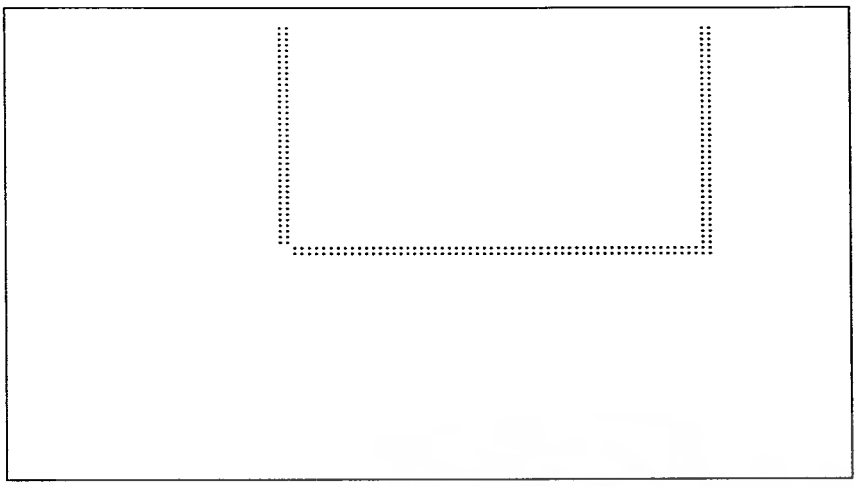
รูปที่ 4.18 ภาพผลการทดลอง หลังทำการเลื่อนภาพมาทางซ้าย (2x) ตำแหน่งที่ 4

รูปที่ 4.19 ภาพผลการทดลอง หลังทำการเลื่อนภาพมาทางขวา (2x) ตำแหน่งที่ 6



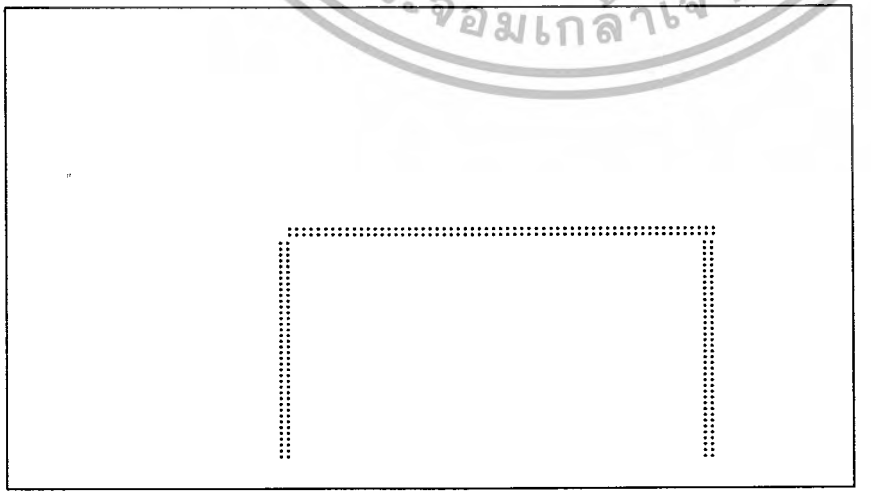
รูปที่ 4.20 ภาพผลการทดลอง หลังทำการเลื่อนภาพขึ้นมาทางด้านบน (2x) ตำแหน่งที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



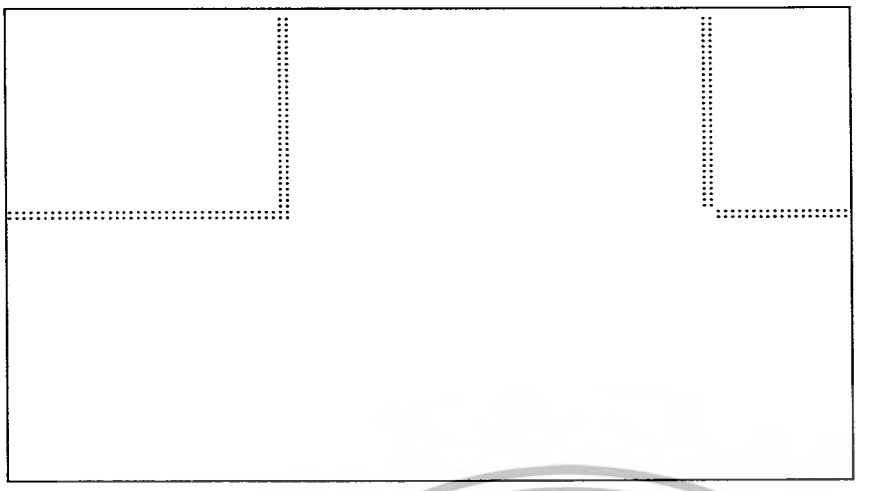
รูปที่ 4.21 ภาพผลการทดลอง หลังทำการเลื่อนภาพลงมาทางด้านล่าง (2x) ตำแหน่งที่ 8

รูปที่ 4.22 ภาพผลการทดลอง หลังทำการเลื่อนภาพขึ้นมาทางด้านบนซ้าย (2x) ตำแหน่งที่ 1



รูปที่ 4.23 ภาพผลการทดลอง หลังทำการเลื่อนภาพขึ้นมาทางด้านบนซ้าย (2x) ตำแหน่งที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

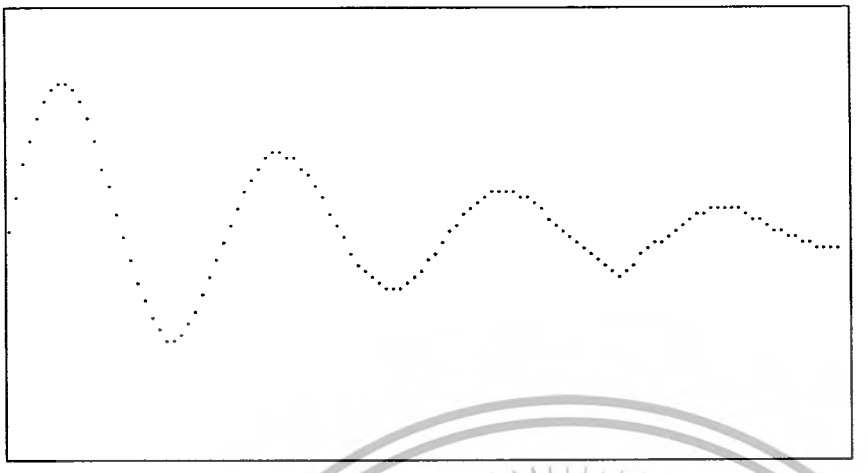


รูปที่ 4.24 ภาพผลการทดลอง หลังทำการเลื่อนภาพลงมาทางด้านล่างซ้าย (2x) ตำแหน่งที่ 7

รูปที่ 4.25 ภาพผลการทดลอง หลังทำการเลื่อนภาพลงมาทางด้านล่างขวา (2x) ตำแหน่งที่ 9

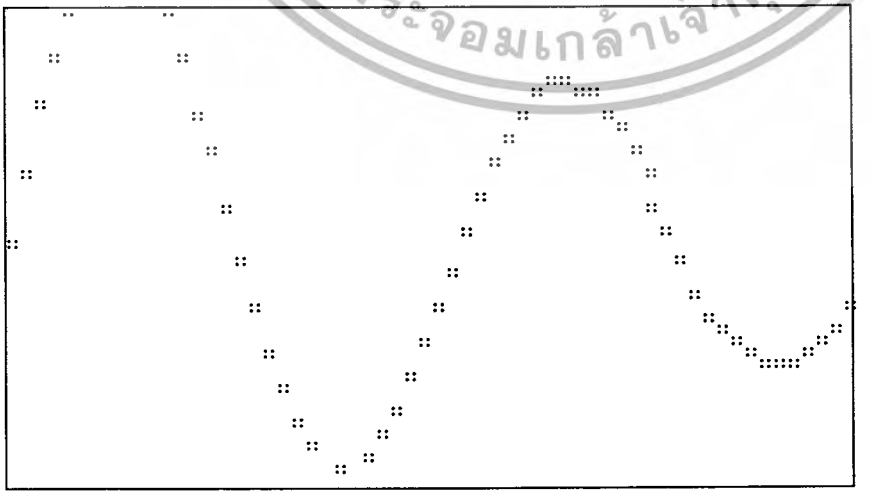
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทดลองที่ 2 รูปผลตอบสนองสัญญาณระดับ จากระบบที่มีฟังก์ชั่นถ่ายโอน $\frac{1}{S^2 + 0.2S + 1}$



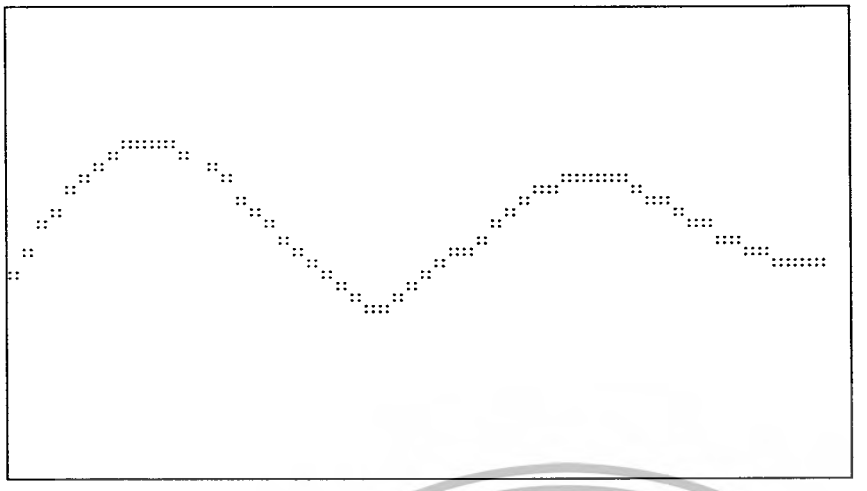
รูปที่ 4.26 ภาพผลการทดลองที่2 ขณะยังไม่ทำการขยาย (1x)

รูปที่ 4.27 ภาพผลการทดลองที่ 2 หลังทำการขยาย (2x) ตำแหน่งที่ 5 (กลางจอ)



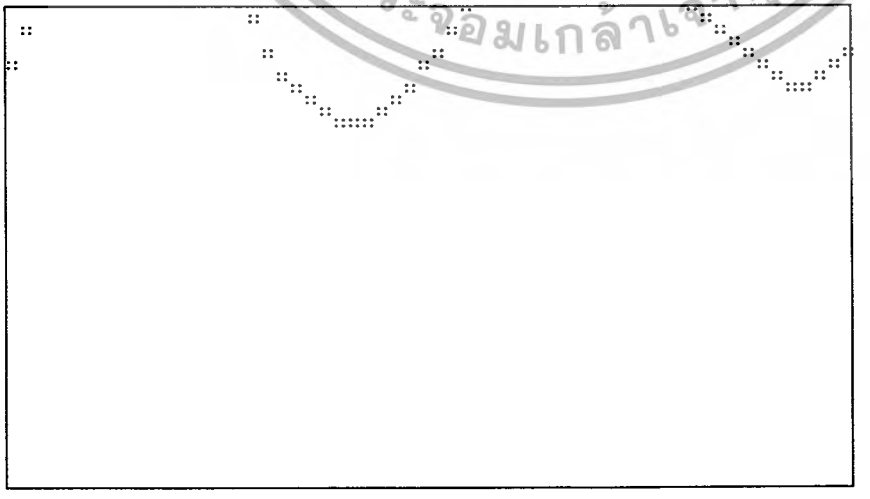
รูปที่ 4.28 ภาพผลการทดลองที่ 2 หลังทำการเลื่อนภาพมาทางซ้าย (2x) ตำแหน่งที่ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



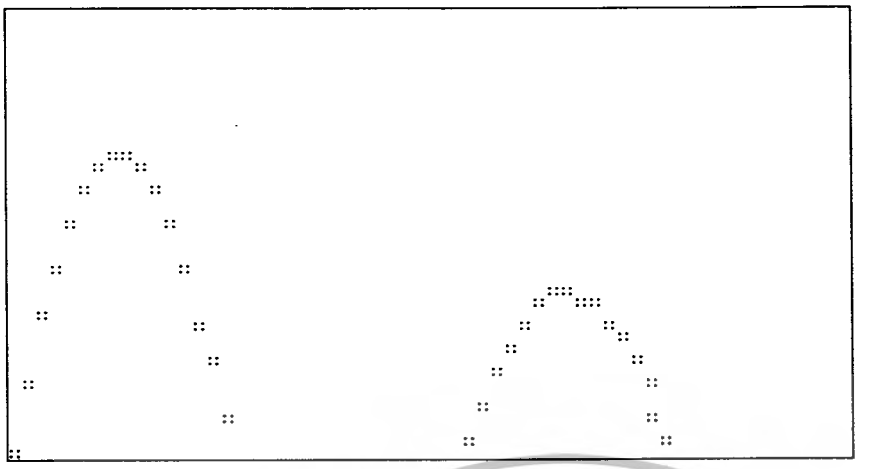
รูปที่ 4.29 ภาพผลการทดลองที่ 2 หลังทำการเลื่อนภาพมาทางขวา (2x) ตำแหน่งที่ 6

รูปที่ 4.30 ภาพผลการทดลองที่ 2 หลังทำการเลื่อนภาพขึ้นมาทางด้านบน (2x) ตำแหน่งที่ 2



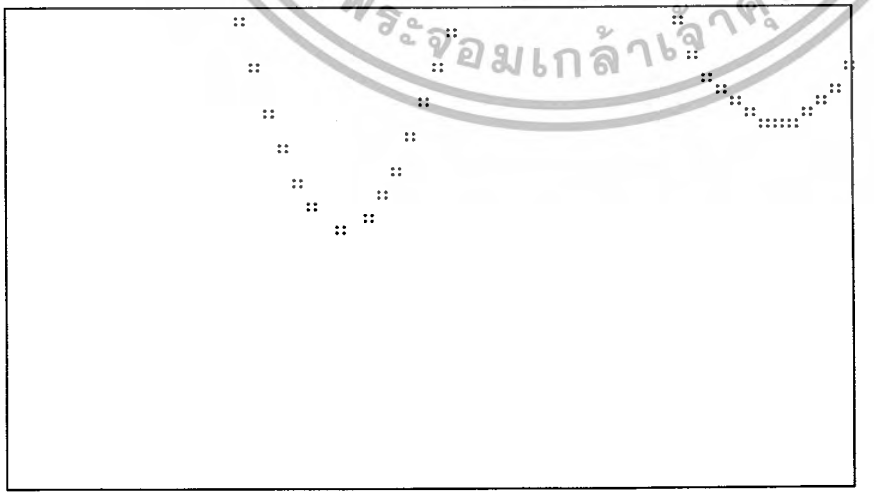
รูปที่ 4.31 ภาพผลการทดลองที่ 2 หลังทำการเลื่อนภาพลงมาทางด้านล่าง (2x) ตำแหน่งที่ 8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



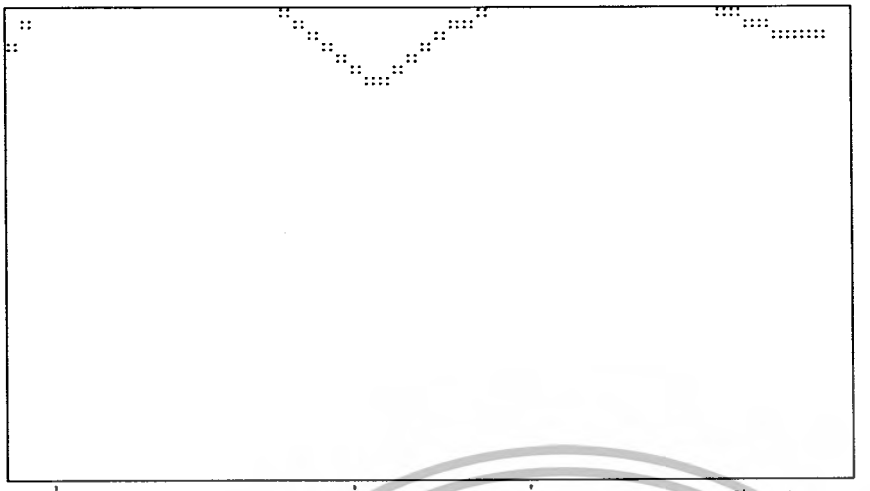
รูปที่ 4.32 ภาพผลการทดลองที่ 2 หลังทำการเลื่อนภาพขึ้นมาทางด้านบนซ้าย (2x) ตำแหน่งที่ 1

รูปที่ 4.33 ภาพผลการทดลองที่ 2 หลังทำการเลื่อนภาพขึ้นมาทางด้านบนซ้าย (2x) ตำแหน่งที่ 3



รูปที่ 4.34 ภาพผลการทดลองที่ 2 หลังทำการเลื่อนภาพลงมาทางด้านล่างซ้าย (2x) ตำแหน่งที่ 7

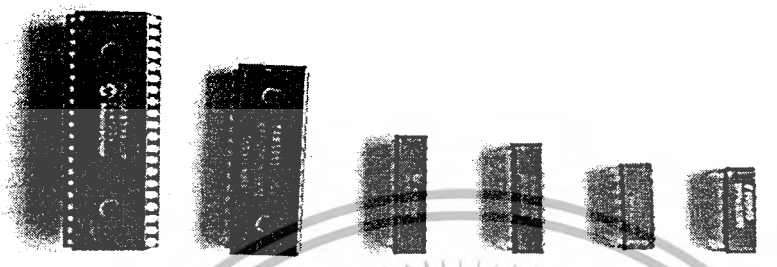
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



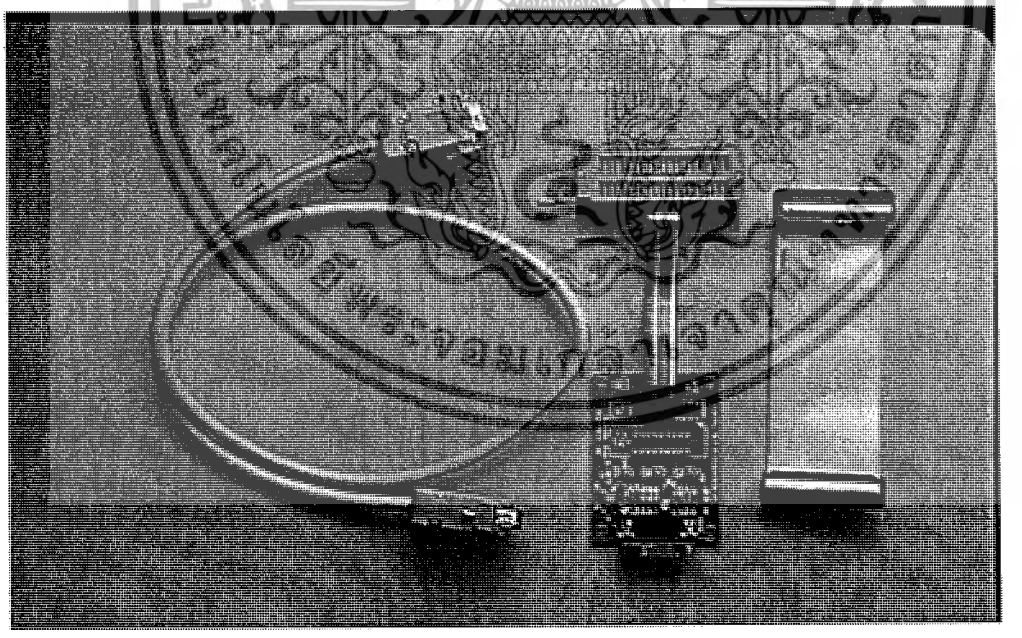
รูปที่ 4.35 ภาพผลการทดลองที่ 2 หลังทำการเลื่อนภาพลงมาทางด้านล่างขวา (2x) ตำแหน่งที่ 9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ภาพถ่ายของอุปกรณ์และวงจร



รูปที่ 4.36 ภาพไมโครคอนโทรลเลอร์และไอซีเบอร์ต่างๆ

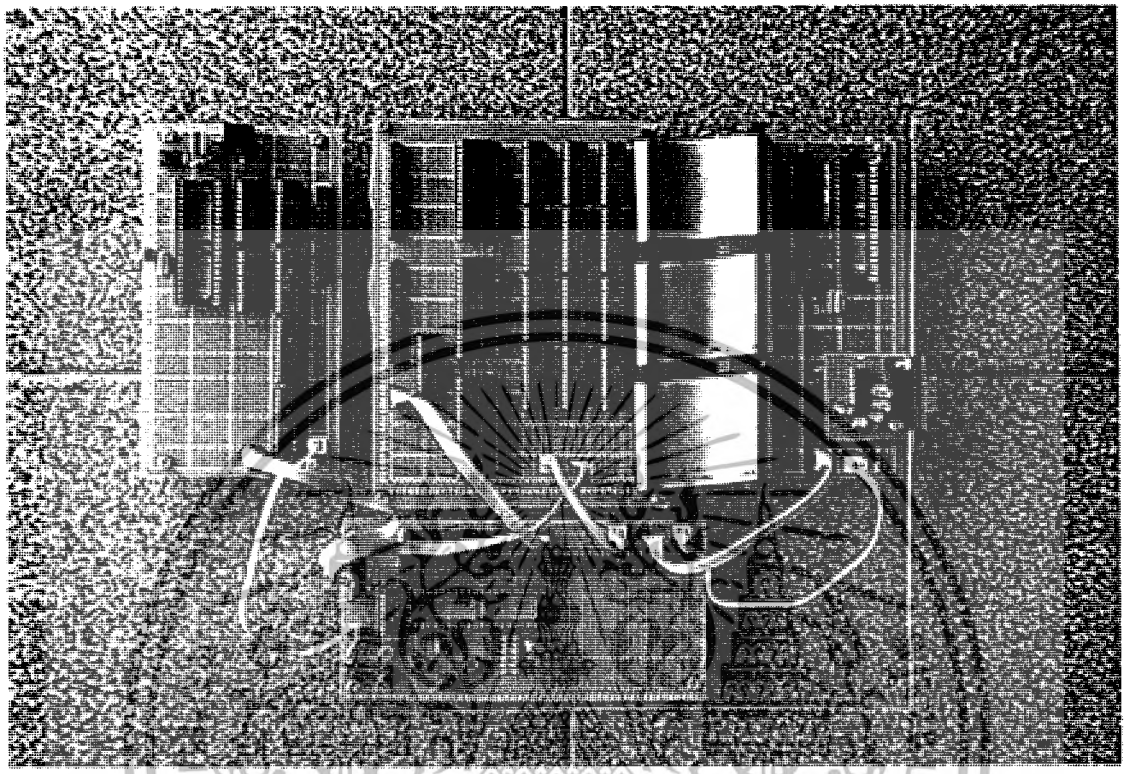


รูปที่ 4.37 ภาพสายสัญญาณที่ใช้ และ บอร์ดโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพวงจรจัดการหน่วยความจำสำหรับแสดงผลสัญญาณบนจอซีอาร์ที

(MEMORY MANAGEMENT CIRCUIT FOR DISPLAY ON CRT MONITOR)



รูปที่ 4.38 วงจรจัดการหน่วยความจำสำหรับแสดงผลสัญญาณบนจอซีอาร์ที

จากรูป คือ วงจรจัดการหน่วยความจำสำหรับแสดงผลสัญญาณบนจอซีอาร์ทีประกอบไปด้วยส่วนรับสัญญาณอินพุต, ส่วนของหน่วยความจำ, ส่วนควบคุมและลำดับข้อมูลของหน่วยความจำ และ ส่วนแสดงผล โดยเชื่อมต่อแต่ละส่วนด้วยสายสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
บทสรุปและวิจารณ์

5.1 บทสรุปและวิจารณ์

ในภาคการศึกษาที่ 1 ได้ทำการศึกษาการทำงานของอุปกรณ์ที่จะใช้แต่ละตัวเช่น ไมโครคอนโทรลเลอร์ตระกูล PIC หน่วยความจำสำรองประเภท RAM เป็นต้น แล้วจึงทำการออกแบบวงจร ที่ใช้หน่วยความจำสำรอง 4 ตัว เพื่อจุดประสงค์ให้สามารถทำการส่งข้อมูลได้อย่างรวดเร็วและต่อเนื่องที่สุด โดยไม่มีการปรับปรุงหรือเปลี่ยนแปลงรูปสัญญาณเลย

ในภาคการศึกษาที่ 2 ได้ทำการออกแบบวงจรเพื่อให้มีความสามารถที่จะทำการปรับเปลี่ยนรูปสัญญาณได้เพิ่มขึ้น คือ การจับภาพนิ่ง ขยายภาพ เลื่อนภาพไปทาง ซ้าย ขวา บน ล่าง โดยแบ่งหน้าจอออกเป็น 9 ส่วน แต่จะทำให้ความเร็วของการรับส่งข้อมูลอาจจะลดลงจากเดิม เพราะใช้หน่วยความจำเพียง 3 ตัว รวมทั้งยังต้องใช้เวลาในการประมวลผลมากขึ้นขณะที่ทำงานแบบจับภาพนิ่ง

5.2 ปัญหาและแนวทางแก้ไข

1. วงจรที่ใช้งานค่อนข้างจะมีความซับซ้อนเพราะใช้อุปกรณ์จำนวนมากในการออกแบบ ซึ่งในทางอุตสาหกรรมอาจจะเลือกให้ลดจำนวนอุปกรณ์ลง แต่อาจจะทำให้ประสิทธิภาพลดลง
2. การขยายและการเลื่อนภาพนิ่ง จะทำให้ความเร็วของการรับส่งข้อมูลอาจจะลดลง เพราะต้องใช้เวลาในการประมวลผลมากขึ้นซึ่งแนวทางแก้ไขอาจะต้องทำการออกแบบ เลือกใช้อุปกรณ์ที่มีความเร็วสูงขึ้น

5.3 แนวทางในการพัฒนา

1. อาจจะทำการเพิ่มความละเอียดของหน้าจอขึ้น แต่อาจจะต้องเปลี่ยน ตัวหน่วยความจำให้มีความจุสูงขึ้น
2. อาจจะเพิ่มฟังก์ชั่นการทำงานอีกเช่น การรับอินพุต 2 ทาง แล้วนำสัญญาณมาแสดงเปรียบเทียบกัน บนหน้าจอเดียวกัน
3. หากต้องการเพิ่มประสิทธิภาพทางความเร็วในการประมวลผล อาจจะต้องทำการเปลี่ยนอุปกรณ์ที่มีความความเร็วสูงขึ้น เช่นในตัวหน่วยความจำต้องดูที่ความเร็วในการจัดการข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.โปรแกรมสำหรับไมโครคอนโทรลเลอร์ 1

Initial Set Up

0000: MOVLW 00
0001: MOVWF 0A
0002: GOTO 0EC
0003: NOP
0004: MOVWF 7F
0005: SWAPF 03,W
0006: BCF 03.5
0007: BCF 03.6
0008: MOVWF 21
0009: MOVF 0A,W
000A: MOVWF 20
000B: CLRF 0A
000C: MOVF 04,W
000D: MOVWF 22
000E: MOVF 77,W
000F: MOVWF 23
0010: MOVF 78,W
0011: MOVWF 24
0012: MOVF 79,W
0013: MOVWF 25
0014: MOVF 7A,W
0015: MOVWF 26
0016: MOVF 7B,W
0017: MOVWF 27
0018: BCF 03.7
0019: BCF 03.5
001A: MOVLW 8C
001B: MOVWF 04
001C: BTFSS 00.2
001D: GOTO 020
001E: BTFSC 0C.2
001F: GOTO 037
0020: BTFSS 0B.5
0021: GOTO 024
0022: BTFSC 0B.2
0023: GOTO 03A
0024: MOVF 22,W
0025: MOVWF 04
0026: MOVF 23,W
0027: MOVWF 77
0028: MOVF 24,W
0029: MOVWF 78
002A: MOVF 25,W
002B: MOVWF 79
002C: MOVF 26,W
002D: MOVWF 7A
002E: MOVF 27,W
002F: MOVWF 7B
0030: MOVF 20,W
0031: MOVWF 0A
0032: SWAPF 21,W
0033: MOVWF 03
0034: SWAPF 7F,F
0035: SWAPF 7F,W
0036: RETFIE
0037: BCF 0A.3
0038: BCF 0A.4
0039: GOTO 042
003A: BCF 0A.3
003B: BCF 0A.4
003C: GOTO 03D
*

Declare Parameter

00D1: MOVLW 3A
00D2: MOVWF 04
00D3: BCF 03.7
00D4: MOVF 00,W
00D5: BTFSC 03.2
00D6: GOTO 0E9
00D7: MOVLW 06
00D8: MOVWF 78
00D9: MOVLW BF
00DA: MOVWF 77
00DB: CLRWDT
00DC: DECFSZ 77,F
00DD: GOTO 0DB
00DE: DECFSZ 78,F
00DF: GOTO 0D9
00E0: MOVLW 7A
00E1: MOVWF 77
00E2: DECFSZ 77,F
00E3: GOTO 0E2
00E4: NOP
00E5: NOP
00E6: CLRWDT
00E7: DECFSZ 00,F
00E8: GOTO 0D7
00E9: BCF 0A.3
00EA: BCF 0A.4
00EB: GOTO 28A
(RETURN)
*

Interrupt to clear watchdog timer

003D: CLRWDT
003E: BCF 0B.2
003F: BCF 0A.3
0040: BCF 0A.4
0041: GOTO 024

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CCP interrupt

0042: BSF 38.1

0043: BCF 0C.2

0044: BCF 0A.3

0045: BCF 0A.4

0046: GOTO 024

*

XY to Address Function

00BD: CLRF 3E

00BE: MOVLW 78

00BF: MOVWF 3D

00C0: MOVF 3B,W

00C1: MOVWF 40

00C2: MOVF 3A,W

00C3: MOVWF 3F

00C4: GOTO 0A6

00C5: MOVF 3C,W

00C6: ADDWF 78,W

00C7: MOVWF 34

00C8: MOVF 79,W

00C9: MOVWF 35

00CA: MOVLW 00

00CB: BTFSC 03.0

00CC: MOVLW 01

00CD: ADDWF 35,F

00CE: CLRF 36

00CF: CLRF 37

00D0: RETLW 00

*

Separate Address Function

(17 bit to 8+8+1 bit)

0047: MOVF 3A,W

0048: MOVWF 28

0049: MOVF 3B,W

004A: MOVWF 29

004B: MOVF 3C,W

004C: MOVWF 2A

004D: RETLW 00

*

Write Ram Function

(specific address

and data)

004E: MOVLW 00

004F: BSF 03.5

0050: MOVWF 06

0051: BCF 03.5

0052: MOVF 3A,W

0053: MOVWF 06

0054: BSF 03.5

0055: BCF 05.0

0056: BCF 03.5

0057: BSF 05.0

0058: MOVLW 00

0059: BSF 03.5

005A: MOVWF 06

005B: BCF 03.5

005C: MOVF 3B,W

005D: MOVWF 06

005E: BSF 03.5

005F: BCF 05.1

0060: BCF 03.5

0061: BSF 05.1

0062: BSF 03.5

0063: BCF 09.1

0064: BCF 03.5

0065: BCF 09.1

0066: BSF 03.5

0067: BCF 05.5

0068: BCF 03.5

0069: BCF 05.5

006A: BSF 03.5

006B: BCF 09.2

006C: BCF 03.5

006D: BSF 09.2

006E: BSF 03.5

006F: BCF 09.0

0070: BCF 03.5

0071: BSF 09.0

0072: MOVLW 00

0073: BSF 03.5

0074: MOVWF 06

0075: BCF 03.5

0076: MOVF 3D,W

0077: MOVWF 06

0078: BSF 03.5

0079: BCF 05.5

007A: BCF 03.5

007B: BSF 05.5

007C: BSF 03.5

007D: BCF 09.2

007E: BCF 03.5

007F: BCF 09.2

0080: BSF 03.5

0081: BCF 09.0

0082: BCF 03.5

0083: BCF 09.0

0084: RETLW 00

*

Clear Ram Function (write

data with "0" in all address)

0085: CLRF 37

0086: CLRF 36

0087: CLRF 35

0088: CLRF 34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0089: MOVF 37,W
008A: MOVWF 3D
008B: MOVF 36,W
008C: MOVWF 3C
008D: MOVF 35,W
008E: MOVWF 3B
008F: MOVF 34,W
0090: MOVWF 3A
0091: CALL 047
0092: MOVF 28,W
0093: MOVWF 3A
0094: MOVF 29,W
0095: MOVWF 3B
0096: MOVF 2A,W
0097: MOVWF 3C
0098: CLRF 3D
0099: CALL 04E
009A: MOVLW 01
009B: ADDWF 34,F
009C: BTFSC 03.0
009D: INCF 35,F
009E: BTFSC 03.2
009F: INCF 36,F
00A0: BTFSC 03.2
00A1: INCF 37,F
00A2: GOTO 089
00A3: BCF 0A.3
00A4: BCF 0A.4
00A5: GOTO 15D
(RETURN)
*

Main program

Initial Hardware setup

00EC: CLRF 04
00ED: BCF 03.7
00EE: MOVLW 1F
00EF: ANDWF 03,F
00F0: MOVLW 07
00F1: BSF 03.5
00F2: MOVWF 1F
00F3: MOVLW 81
00F4: MOVWF 19
00F5: MOVLW 26
00F6: MOVWF 18
00F7: MOVLW 90
00F8: BCF 03.5
00F9: MOVWF 18
00FA: BSF 38.0
00FB: BCF 38.1
00FC: BCF 38.2
00FD: MOVLW FF
00FE: MOVWF 39
00FF: MOVLW 06
0100: BSF 03.5
0101: MOVWF 1F
0102: BCF 03.5
0103: MOVF 1F,W
0104: ANDLW 38
0105: IORLW 01
0106: MOVWF 1F
0107: BCF 14.5
0108: BCF 39.5
0109: MOVF 39,W
010A: BSF 03.5
010B: MOVWF 07
010C: BCF 03.5
010D: BSF 39.4
010E: MOVF 39,W
010F: BSF 03.5
0110: MOVWF 07
0111: BCF 03.5
0112: BCF 39.3
0113: MOVF 39,W
0114: BSF 03.5
0115: MOVWF 07
0116: MOVLW 00
0117: BCF 03.5
0118: MOVWF 14
0119: BSF 03.5
011A: MOVWF 14
011B: BCF 09.4
011C: MOVLW 02
011D: MOVWF 77
011E: BTFSC 77.3
011F: GOTO 122
0120: BCF 03.5
0121: GOTO 12C
0122: MOVLW 07
0123: BCF 03.5
0124: CLRF 01
0125: MOVLW 81
0126: MOVWF 04
0127: BCF 03.7
0128: MOVF 00,W
0129: ANDLW C0
012A: IORLW 0F
012B: MOVWF 00
012C: CLRWDT
012D: MOVLW 81
012E: MOVWF 04
012F: MOVF 00,W
0130: ANDLW C0
0131: IORWF 77,W
0132: MOVWF 00
0133: CLRF 10
0134: MOVLW 00
0135: MOVWF 78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0136: MOVWF 12
0137: MOVLW 00
0138: BSF 03.5
0139: MOVWF 12
013A: ANDLW F8
013B: BTFSS 03.2
013C: GOTO 140
013D: BCF 03.5
013E: BCF 07.1
013F: BSF 03.5
0140: BCF 03.5
0141: CLRF 1D
0142: CLRF 1D
0143: MOVLW 04
0144: ANDLW F8
0145: BTFSC 03.2
0146: BCF 07.2
0147: CLRF 17
0148: MOVLW 04
0149: MOVWF 17
*

Setup Interrupt Using

014A: BSF 03.5
014B: BSF 0D.0
014C: BSF 0C.5
014D: BCF 03.5
014E: BSF 0B.5
014F: MOVLW C0
0150: IORWF 0B,F
0151: BSF 03.5
0152: BCF 01.7
0153: BCF 03.5
*

Start Program

0154: MOVLW 00
0155: BSF 03.5
0156: MOVWF 06
0157: MOVWF 08
0158: BCF 03.5

Wait for Start Write Ram

Interrupt

0159: BTFSS 38.1
015A: GOTO 159
015B: GOTO 15C

Clear Ram

015C: GOTO 085

Write start

015D: MOVF 31,W
015E: SUBLW 14
015F: BTFSC 03.2
0160: GOTO 163
0161: MOVLW 00
0162: GOTO 163
0163: MOVF 32,W
0164: SUBLW 0A
0165: BTFSC 03.2
0166: GOTO 169
0167: MOVLW 00
0168: GOTO 169
0169: CLRF 3B
016A: MOVF 31,W
016B: MOVWF 3A
016C: MOVF 32,W
016D: MOVWF 3C
016E: CALL 0BD
016F: DECFSZ 2D,W
0170: GOTO 172
0171: GOTO 174
0172: MOVLW 00
0173: GOTO 174
0174: MOVF 37,W
0175: MOVWF 3D
0176: MOVF 36,W
0177: MOVWF 3C
0178: MOVF 35,W
0179: MOVWF 3B
017A: MOVF 34,W
017B: MOVWF 3A
017C: CALL 047
017D: MOVF 28,W
017E: MOVWF 3A
017F: MOVF 29,W
0180: MOVWF 3B
0181: MOVF 2A,W
0182: MOVWF 3C
0183: MOVLW 01
0184: MOVWF 3D
0185: CALL 04E
0186: MOVLW 01
0187: ADDWF 34,F
0188: BTFSC 03.0
0189: INCF 35,F
018A: BTFSC 03.2
018B: INCF 36,F
018C: BTFSC 03.2
018D: INCF 37,F
018E: INCF 2D,F
018F: GOTO 174
0190: DECFSZ 2D,W
0191: GOTO 193
0192: GOTO 195
0193: MOVLW 00
0194: GOTO 195

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0195: MOVF 37,W
0196: MOVWF 3D
0197: MOVF 36,W
0198: MOVWF 3C
0199: MOVF 35,W
019A: MOVWF 3B
019B: MOVF 34,W
019C: MOVWF 3A
019D: CALL 047
019E: MOVF 28,W
019F: MOVWF 3A
01A0: MOVF 29,W
01A1: MOVWF 3B
01A2: MOVF 2A,W
01A3: MOVWF 3C
01A4: MOVLW 01
01A5: MOVWF 3D
01A6: CALL 04E
01A7: MOVLW 01
01A8: ADDWF 34,F
01A9: BTFSC 03.0
01AA: INCF 35,F
01AB: BTFSC 03.2
01AC: INCF 36,F
01AD: BTFSC 03.2
01AE: INCF 37,F
01AF: INCF 2D,F
01B0: GOTO 195
01B1: DECFSZ 2D,W
01B2: GOTO 1B4
01B3: GOTO 1B6
01B4: MOVLW 00
01B5: GOTO 1B6
01B6: MOVF 31,W
01B7: SUBLW 14
01B8: BTFSC 03.2
01B9: GOTO 1BC
01BA: MOVLW 00
01BB: GOTO 1BC
01BC: INCF 32,F
01BD: CLRF 3B
01BE: MOVF 31,W
01BF: MOVWF 3A
01C0: MOVF 32,W
01C1: MOVWF 3C
01C2: CALL 0BD
01C3: MOVF 28,W
01C4: MOVWF 3A
01C5: MOVF 29,W
01C6: MOVWF 3B
01C7: MOVF 2A,W
01C8: MOVWF 3C
01C9: MOVLW 01
01CA: MOVWF 3D
01CB: CALL 04E
01CC: MOVF 31,W
01CD: SUBLW 32
01CE: BTFSC 03.2
01CF: GOTO 1D2
01D0: MOVLW 00
01D1: GOTO 1D2
01D2: CLRF 3B
01D3: MOVF 31,W
01D4: MOVWF 3A
01D5: MOVF 32,W
01D6: MOVWF 3C
01D7: CALL 0BD
01D8: MOVF 28,W
01D9: MOVWF 3A
01DA: MOVF 29,W
01DB: MOVWF 3B
01DC: MOVF 2A,W
01DD: MOVWF 3C
01DE: MOVLW 01
01DF: MOVWF 3D
01E0: CALL 04E
01E1: MOVF 31,W
01E2: SUBLW 50
01E3: BTFSC 03.2
01E4: GOTO 1E7
01E5: MOVLW 00
01E6: GOTO 1E7
01E7: CLRF 3B
01E8: MOVF 31,W
01E9: MOVWF 3A
01EA: MOVF 32,W
01EB: MOVWF 3C
01EC: CALL 0BD
01ED: MOVF 28,W
01EE: MOVWF 3A
01EF: MOVF 29,W
01F0: MOVWF 3B
01F1: MOVF 2A,W
01F2: MOVWF 3C
01F3: MOVLW 01
01F4: MOVWF 3D
01F5: CALL 04E
01F6: MOVF 31,W
01F7: SUBLW 6E
01F8: BTFSC 03.2
01F9: GOTO 1FC
01FA: MOVLW 00
01FB: GOTO 1FC
01FC: CLRF 3B
01FD: MOVF 31,W
01FE: MOVWF 3A
01FF: MOVF 32,W
0200: MOVWF 3C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0201: CALL 0BD
0202: MOVF 28,W
0203: MOVWF 3A
0204: MOVF 29,W
0205: MOVWF 3B
0206: MOVF 2A,W
0207: MOVWF 3C
0208: MOVLW 01
0209: MOVWF 3D
020A: CALL 04E
020B: INCF 2D,F
020C: GOTO 1B6
020D: DECFSZ 31,W
020E: GOTO 210
020F: GOTO 212
0210: MOVLW 00
0211: GOTO 212
0212: MOVF 32,W
0213: SUBLW 46
0214: BTFSC 03.2
0215: GOTO 218
0216: MOVLW 00
0217: GOTO 218
0218: CLRF 3B
0219: MOVF 31,W
021A: MOVWF 3A
021B: MOVF 32,W
021C: MOVWF 3C
021D: CALL 0BD
021E: DECFSZ 2D,W
021F: GOTO 221
0220: GOTO 223
0221: MOVLW 00
0222: GOTO 223
0223: MOVF 37,W
0224: MOVWF 3D
0225: MOVF 36,W
0226: MOVWF 3C
0227: MOVF 35,W
0228: MOVWF 3B
0229: MOVF 34,W
022A: MOVWF 3A
022B: CALL 047
022C: MOVF 28,W
022D: MOVWF 3A
022E: MOVF 29,W
022F: MOVWF 3B
0230: MOVF 2A,W
0231: MOVWF 3C
0232: MOVLW 01
0233: MOVWF 3D
0234: CALL 04E
0235: MOVLW 01
0236: ADDWF 34,F
0237: BTFSC 03.0
0238: INCF 35,F
0239: BTFSC 03.2
023A: INCF 36,F
023B: BTFSC 03.2
023C: INCF 37,F
023D: INCF 2D,F
023E: GOTO 223
023F: DECFSZ 2D,W
0240: GOTO 242
0241: GOTO 244
0242: MOVLW 00
0243: GOTO 244
0244: MOVF 37,W
0245: MOVWF 3D
0246: MOVF 36,W
0247: MOVWF 3C
0248: MOVF 35,W
0249: MOVWF 3B
024A: MOVF 34,W
024B: MOVWF 3A
024C: CALL 047
024D: MOVF 28,W
024E: MOVWF 3A
024F: MOVF 29,W
0250: MOVWF 3B
0251: MOVF 2A,W
0252: MOVWF 3C
0253: MOVLW 01
0254: MOVWF 3D
0255: CALL 04E
0256: MOVLW 01
0257: ADDWF 34,F
0258: BTFSC 03.0
0259: INCF 35,F
025A: BTFSC 03.2
025B: INCF 36,F
025C: BTFSC 03.2
025D: INCF 37,F
025E: INCF 2D,F
025F: GOTO 244
0260: DECFSZ 2D,W
0261: GOTO 263
0262: GOTO 265
0263: MOVLW 00
0264: GOTO 265
0265: MOVF 37,W
0266: MOVWF 3D
0267: MOVF 36,W
0268: MOVWF 3C
0269: MOVF 35,W
026A: MOVWF 3B
026B: MOVF 34,W
026C: MOVWF 3A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

026D: CALL 047
026E: MOVF 28,W
026F: MOVWF 3A
0270: MOVF 29,W
0271: MOVWF 3B
0272: MOVF 2A,W
0273: MOVWF 3C
0274: MOVLW 01
0275: MOVWF 3D
0276: CALL 04E
0277: MOVLW 01
0278: ADDWF 34,F
0279: BTFSC 03.0
027A: INCF 35,F
027B: BTFSC 03.2
027C: INCF 36,F
027D: BTFSC 03.2
027E: INCF 37,F
027F: INCF 2D,F
0280: GOTO 265
*

Write Finish and create

Interrupt

0281: BCF 39.1
0282: MOVF 39,W
0283: BSF 03.5
0284: MOVWF 07
0285: BCF 03.5
0286: BCF 07.1
0287: MOVLW 05
0288: MOVWF 3A
0289: GOTO 0D1
028A: BCF 39.1
028B: MOVF 39,W
028C: BSF 03.5
028D: MOVWF 07
028E: BCF 03.5
028F: BSF 07.1
0290: BCF 38.1
0291: GOTO 154
0292: SLEEP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.โปรแกรมสำหรับไมโครคอนโทรลเลอร์ 2

Initial Setup

0000: MOVLW 00
0001: MOVWF 0A
0002: GOTO 2FF
0003: NOP
0004: MOVWF 7F
0005: SWAPF 03,W
0006: BCF 03.5
0007: BCF 03.6
0008: MOVWF 21
0009: MOVF 0A,W
000A: MOVWF 20
000B: CLRF 0A
000C: MOVF 04,W
000D: MOVWF 22
000E: MOVF 77,W
000F: MOVWF 23
0010: MOVF 78,W
0011: MOVWF 24
0012: MOVF 79,W
0013: MOVWF 25
0014: MOVF 7A,W
0015: MOVWF 26
0016: MOVF 7B,W
0017: MOVWF 27
0018: BCF 03.7
0019: BCF 03.5
001A: MOVLW 8C
001B: MOVWF 04
001C: BTFSS 00.2
001D: GOTO 020
001E: BTFSC 0C.2
001F: GOTO 03D
0020: MOVLW 8D
0021: MOVWF 04
0022: BTFSS 00.0
0023: GOTO 026
0024: BTFSC 0D.0
0025: GOTO 040
0026: BTFSS 0B.5
0027: GOTO 02A
0028: BTFSC 0B.2
0029: GOTO 043
002A: MOVF 22,W
002B: MOVWF 04
002C: MOVF 23,W
002D: MOVWF 77
002E: MOVF 24,W
002F: MOVWF 78
0030: MOVF 25,W
0031: MOVWF 79
0032: MOVF 26,W
0033: MOVWF 7A
0034: MOVF 27,W
0035: MOVWF 7B
0036: MOVF 20,W
0037: MOVWF 0A
0038: SWAPF 21,W
0039: MOVWF 03
003A: SWAPF 7F,F
003B: SWAPF 7F,W
003C: RETFIE
003D: BCF 0A.3
003E: BCF 0A.4
003F: GOTO 04B
0040: BCF 0A.3
0041: BCF 0A.4
0042: GOTO 050
0043: BCF 0A.3
0044: BCF 0A.4
0045: GOTO 046
*

Declare Parameter

00EA: MOVLW 47
00EB: MOVWF 04
00EC: BCF 03.7
00ED: MOVF 00,W
00EE: BTFSC 03.2
00EF: GOTO 102
00F0: MOVLW 06
00F1: MOVWF 78
00F2: MOVLW BF
00F3: MOVWF 77
00F4: CLRWDT
00F5: DECFSZ 77,F
00F6: GOTO 0F4
00F7: DECFSZ 78,F
00F8: GOTO 0F2
00F9: MOVLW 7A
00FA: MOVWF 77
00FB: DECFSZ 77,F
00FC: GOTO 0FB
00FD: NOP
00FE: NOP
00FF: CLRWDT
0100: DECFSZ 00,F
0101: GOTO 0F0
0102: RETLW 00
*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Interrupt to clear watchdog timer

0046: CLRWDT
0047: BCF 0B.2
0048: BCF 0A.3
0049: BCF 0A.4
004A: GOTO 02A

CCP1 interrupt

004B: BSF 42.5
004C: BCF 0C.2
004D: BCF 0A.3
004E: BCF 0A.4
004F: GOTO 02A

CCP2 interrupt

0050: BSF 42.7
0051: BCF 0D.0
0052: BCF 0A.3
0053: BCF 0A.4
0054: GOTO 02A

*

XY to Address Function

0191: MOVLW 78
0192: MOVWF 4A
0193: MOVF 49,W
0194: MOVWF 4B
0195: GOTO 157
0196: MOVF 78,W
0197: ADDWF 47,W
0198: MOVWF 36
0199: MOVF 48,W
019A: MOVWF 37
019B: MOVLW 00
019C: BTFSC 03.0
019D: MOVLW 01
019E: ADDWF 37,F
019F: CLRF 38
01A0: CLRF 39

*

Separate Address 1 Function

(17 bit to 8+8+1 bit)

017E: MOVF 47,W
017F: MOVWF 30
0180: MOVF 48,W
0181: MOVWF 31
0182: MOVF 49,W
0183: MOVWF 32
0184: RETLW 00

Separate Address 2 Function

(17 bit to 8+8+1 bit)

0185: MOVF 47,W
0186: MOVWF 33
0187: MOVF 48,W
0188: MOVWF 34
0189: MOVF 49,W
018A: MOVWF 35
018B: RETLW 00

*

Float all pin

0055: BSF 03.5
0056: BCF 05.1
0057: BCF 03.5
0058: BSF 05.1
0059: MOVLW 10
005A: BSF 03.5
005B: MOVWF 05
005C: MOVLW 07
005D: MOVWF 09
005E: BCF 03.5
005F: RETLW 00

*

Open Port function

0060: MOVF 45,W
0061: MOVWF 77
0062: MOVLW 01
0063: SUBWF 45,W
0064: BTFSC 03.2
0065: GOTO 077
0066: MOVLW 02
0067: SUBWF 77,W
0068: BTFSC 03.2
0069: GOTO 08A
006A: MOVLW 03
006B: SUBWF 77,W
006C: BTFSC 03.2
006D: GOTO 09D
006E: MOVLW 04
006F: SUBWF 77,W
0070: BTFSC 03.2
0071: GOTO 0B0
0072: MOVLW 05
0073: SUBWF 77,W
0074: BTFSC 03.2
0075: GOTO 0C3
0076: GOTO 0D6
0077: BCF 43.5
0078: MOVF 43,W
0079: BSF 03.5
007A: MOVWF 07
007B: BCF 03.5
007C: BSF 07.5
007D: BCF 43.6
007E: MOVF 43,W

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

007F: BSF 03.5
0080: MOVWF 07
0081: BCF 03.5
0082: BCF 07.6
0083: BCF 43.7
0084: MOVF 43,W
0085: BSF 03.5
0086: MOVWF 07
0087: BCF 03.5
0088: BCF 07.7
0089: GOTO 0E9
008A: BCF 43.5
008B: MOVF 43,W
008C: BSF 03.5
008D: MOVWF 07
008E: BCF 03.5
008F: BCF 07.5
0090: BCF 43.6
0091: MOVF 43,W
0092: BSF 03.5
0093: MOVWF 07
0094: BCF 03.5
0095: BSF 07.6
0096: BCF 43.7
0097: MOVF 43,W
0098: BSF 03.5
0099: MOVWF 07
009A: BCF 03.5
009B: BCF 07.7
009C: GOTO 0E9
009D: BCF 43.5
009E: MOVF 43,W
009F: BSF 03.5
00A0: MOVWF 07
00A1: BCF 03.5
00A2: BSF 07.5
00A3: BCF 43.6
00A4: MOVF 43,W
00A5: BSF 03.5
00A6: MOVWF 07
00A7: BCF 03.5
00A8: BSF 07.6
00A9: BCF 43.7
00AA: MOVF 43,W
00AB: BSF 03.5
00AC: MOVWF 07
00AD: BCF 03.5
00AE: BCF 07.7
00AF: GOTO 0E9
00B0: BCF 43.5
00B1: MOVF 43,W
00B2: BSF 03.5
00B3: MOVWF 07
00B4: BCF 03.5
00B5: BCF 07.5
00B6: BCF 43.6
00B7: MOVF 43,W
00B8: BSF 03.5
00B9: MOVWF 07
00BA: BCF 03.5
00BB: BCF 07.6
00BC: BCF 43.7
00BD: MOVF 43,W
00BE: BSF 03.5
00BF: MOVWF 07
00C0: BCF 03.5
00C1: BSF 07.7
00C2: GOTO 0E9
00C3: BCF 43.5
00C4: MOVF 43,W
00C5: BSF 03.5
00C6: MOVWF 07
00C7: BCF 03.5
00C8: BSF 07.5
00C9: BCF 43.6
00CA: MOVF 43,W
00CB: BSF 03.5
00CC: MOVWF 07
00CD: BCF 03.5
00CE: BCF 07.6
00CF: BCF 43.7
00D0: MOVF 43,W
00D1: BSF 03.5
00D2: MOVWF 07
00D3: BCF 03.5
00D4: BSF 07.7
00D5: GOTO 0E9
00D6: BCF 43.5
00D7: MOVF 43,W
00D8: BSF 03.5
00D9: MOVWF 07
00DA: BCF 03.5
00DB: BCF 07.5
00DC: BCF 43.6
00DD: MOVF 43,W
00DE: BSF 03.5
00DF: MOVWF 07
00E0: BCF 03.5
00E1: BCF 07.6
00E2: BCF 43.7
00E3: MOVF 43,W
00E4: BSF 03.5
00E5: MOVWF 07
00E6: BCF 03.5
00E7: BCF 07.7
00E8: GOTO 0E9
00E9: RETLW 00
*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Read Ram

011D: CALL 055
011E: MOVLW 01
011F: SUBWF 44,W
0120: ADDLW FD
0121: BTFSC 03.0
0122: GOTO 131
0123: ADDLW 03
0124: GOTO 147
0125: MOVLW 02
0126: MOVWF 45
0127: CALL 060
0128: GOTO 131
0129: MOVLW 03
012A: MOVWF 45
012B: CALL 060
012C: GOTO 131
012D: MOVLW 04
012E: MOVWF 45
012F: CALL 060
0130: GOTO 131
0147: BSF 0A.0
0148: BCF 0A.1
0149: BCF 0A.2
014A: ADDWF 02,F
014B: GOTO 125
014C: GOTO 129
014D: GOTO 12D
*

Read Start Interrupt

0131: BCF 43.3
0132: MOVF 43,W
0133: BSF 03.5
0134: MOVWF 07
0135: BCF 03.5
0136: BCF 07.3
0137: MOVLW 05
0138: MOVWF 47
0139: CALL 0EA
013A: BCF 43.3
013B: MOVF 43,W
013C: BSF 03.5
013D: MOVWF 07
013E: BCF 03.5
013F: BSF 07.3
0140: BTFSS 42.7
0141: GOTO 140
0142: GOTO 143
0143: CLRF 45
0144: CALL 060
0145: BCF 42.7
0146: RETLW 00
*

Write Ram

0103: CALL 055
0104: MOVLW 01
0105: MOVWF 45
0106: CALL 060
0107: BCF 43.0
0108: MOVF 43,W
0109: BSF 03.5
010A: MOVWF 07
010B: BCF 03.5
010C: BCF 07.0
010D: MOVLW 05
010E: MOVWF 47
010F: CALL 0EA
0110: BCF 43.0
0111: MOVF 43,W
0112: BSF 03.5
0113: MOVWF 07
0114: BCF 03.5
0115: BSF 07.0

wait for write finish interrupt

0116: BTFSS 42.5
0117: GOTO 116
0118: GOTO 119
0119: CLRF 45
011A: CALL 060
011B: BCF 42.5
011C: RETLW 00
*

Set to Normal mode function

*
014E: CALL 103
014F: BCF 42.0
0150: MOVLW 01
0151: MOVWF 44
0152: CALL 11D
0153: BSF 42.3
0154: BCF 0A.3
0155: BCF 0A.4
0156: GOTO 3EC
(RETURN)
*

Cut ram function (for shift and zoom)

018C: CLRF 48
018D: MOVF 45,W
018E: MOVWF 47
018F: MOVF 46,W
0190: MOVWF 49

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

01A1: MOVF 39,W
01A2: MOVWF 4A
01A3: MOVF 38,W
01A4: MOVWF 49
01A5: MOVF 37,W
01A6: MOVWF 48
01A7: MOVF 36,W
01A8: MOVWF 47
01A9: CALL 17E
01AA: MOVF 3A,F
01AB: BTFSS 03.2
01AC: GOTO 1B6
01AD: MOVF 3B,F
01AE: BTFSS 03.2
01AF: GOTO 1B6
01B0: MOVF 3C,F
01B1: BTFSS 03.2
01B2: GOTO 1B6
01B3: MOVF 3D,F
01B4: BTFSC 03.2
01B5: GOTO 1B8
01B6: MOVLW 00
01B7: GOTO 1B8
01B8: MOVF 3D,W
01B9: MOVWF 4A
01BA: MOVF 3C,W
01BB: MOVWF 49
01BC: MOVF 3B,W
01BD: MOVWF 48
01BE: MOVF 3A,W
01BF: MOVWF 47
01C0: CALL 185
01C1: CLRF 2A
01C2: MOVLW 28
01C3: MOVWF 2A
01C4: MOVF 2A,F
01C5: BTFSC 03.2
01C6: GOTO 2DF
01C7: MOVF 3D,W
01C8: MOVWF 4A
01C9: MOVF 3C,W
01CA: MOVWF 49
01CB: MOVF 3B,W
01CC: MOVWF 48
01CD: MOVF 3A,W
01CE: MOVWF 47
01CF: CALL 185
01D0: CLRF 2D
01D1: MOVLW 3C
01D2: MOVWF 2D
01D3: MOVF 2D,F
01D4: BTFSC 03.2
01D5: GOTO 2DD
01D6: MOVLW 00
01D7: BSF 03.5
01D8: MOVWF 06
01D9: BCF 03.5
01DA: MOVF 30,W
01DB: MOVWF 06
01DC: BSF 03.5
01DD: BCF 05.2
01DE: BCF 03.5
01DF: BSF 05.2
01E0: MOVLW 00
01E1: BSF 03.5
01E2: MOVWF 06
01E3: BCF 03.5
01E4: MOVF 31,W
01E5: MOVWF 06
01E6: BSF 03.5
01E7: BCF 05.3
01E8: BCF 03.5
01E9: BSF 05.3
01EA: MOVLW 00
01EB: BSF 03.5
01EC: MOVWF 08
01ED: BCF 03.5
01EE: MOVF 33,W
01EF: MOVWF 08
01F0: BSF 03.5
01F1: BCF 05.2
01F2: BCF 03.5
01F3: BSF 05.2
01F4: MOVLW 00
01F5: BSF 03.5
01F6: MOVWF 08
01F7: BCF 03.5
01F8: MOVF 34,W
01F9: MOVWF 08
01FA: BSF 03.5
01FB: BCF 05.3
01FC: BCF 03.5
01FD: BSF 05.3
01FE: BSF 03.5
01FF: BCF 09.1
0200: BCF 03.5
0201: BCF 09.1
0202: BSF 03.5
0203: BCF 09.2
0204: BCF 03.5
0205: BSF 09.2
0206: BSF 03.5
0207: BCF 09.0
0208: BCF 03.5
0209: BSF 09.0
020A: BSF 03.5
020B: BCF 05.5
020C: BCF 03.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

020D: BCF 05.5
020E: MOVLW 01
020F: MOVWF 47
0210: CALL 0EA
0211: BSF 03.5
0212: BCF 09.0
0213: BCF 03.5
0214: BCF 09.0
0215: BSF 03.5
0216: BCF 05.5
0217: BCF 03.5
0218: BSF 05.5
0219: MOVF 3D,W
021A: MOVWF 4A
021B: MOVF 3C,W
021C: MOVWF 49
021D: MOVF 3B,W
021E: MOVWF 48
021F: MOVF 3A,W
0220: MOVWF 47
0221: CALL 185
0222: MOVLW 00
0223: BSF 03.5
0224: MOVWF 08
0225: BCF 03.5
0226: MOVF 33,W
0227: MOVWF 08
0228: BSF 03.5
0229: BCF 05.2
022A: BCF 03.5
022B: BSF 05.2
022C: MOVLW 00
022D: BSF 03.5
022E: MOVWF 08
022F: BCF 03.5
0230: MOVF 34,W
0231: MOVWF 08
0232: BSF 03.5
0233: BCF 05.3
0234: BCF 03.5
0235: BSF 05.3
0236: BSF 03.5
0237: BCF 09.1
0238: BCF 03.5
0239: BCF 09.1
023A: BSF 03.5
023B: BCF 09.2
023C: BCF 03.5
023D: BSF 09.2
023E: BSF 03.5
023F: BCF 09.0
0240: BCF 03.5
0241: BSF 09.0
0242: BSF 03.5
0243: BCF 05.5
0244: BCF 03.5
0245: BCF 05.5
0246: MOVLW 01
0247: MOVWF 47
0248: CALL 0EA
0249: BSF 03.5
024A: BCF 09.0
024B: BCF 03.5
024C: BCF 09.0
024D: BSF 03.5
024E: BCF 05.5
024F: BCF 03.5
0250: BSF 05.5
0251: MOVF 3D,W
0252: MOVWF 4A
0253: MOVF 3C,W
0254: MOVWF 49
0255: MOVF 3B,W
0256: MOVWF 48
0257: MOVF 3A,W
0258: MOVWF 47
0259: CALL 185
025A: MOVLW 00
025B: BSF 03.5
025C: MOVWF 08
025D: BCF 03.5
025E: MOVF 33,W
025F: MOVWF 08
0260: BSF 03.5
0261: BCF 05.2
0262: BCF 03.5
0263: BSF 05.2
0264: MOVLW 00
0265: BSF 03.5
0266: MOVWF 08
0267: BCF 03.5
0268: MOVF 34,W
0269: MOVWF 08
026A: BSF 03.5
026B: BCF 05.3
026C: BCF 03.5
026D: BSF 05.3
026E: BSF 03.5
026F: BCF 09.1
0270: BCF 03.5
0271: BCF 09.1
0272: BSF 03.5
0273: BCF 09.2
0274: BCF 03.5
0275: BSF 09.2
0276: BSF 03.5
0277: BCF 09.0
0278: BCF 03.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0279: BSF   09.0
027A: BSF   03.5
027B: BCF   05.5
027C: BCF   03.5
027D: BCF   05.5
027E: MOVLW 01
027F: MOVWF 47
0280: CALL  0EA
0281: BSF   03.5
0282: BCF   09.0
0283: BCF   03.5
0284: BCF   09.0
0285: BSF   03.5
0286: BCF   05.5
0287: BCF   03.5
0288: BSF   05.5
0289: MOVF  3D,W
028A: MOVWF 4A
028B: MOVF  3C,W
028C: MOVWF 49
028D: MOVF  3B,W
028E: MOVWF 48
028F: MOVF  3A,W
0290: MOVWF 47
0291: CALL  185
0292: MOVLW 00
0293: BSF   03.5
0294: MOVWF 08
0295: BCF   03.5
0296: MOVF  33,W
0297: MOVWF 08
0298: BSF   03.5
0299: BCF   05.2
029A: BCF   03.5
029B: BSF   05.2
029C: MOVLW 00
029D: BSF   03.5
029E: MOVWF 08
029F: BCF   03.5
02A0: MOVF  34,W
02A1: MOVWF 08
02A2: BSF   03.5
02A3: BCF   05.3
02A4: BCF   03.5
02A5: BSF   05.3
02A6: BSF   03.5
02A7: BCF   09.1
02A8: BCF   03.5
02A9: BCF   09.1
02AA: BSF   03.5
02AB: BCF   09.2
02AC: BCF   03.5
02AD: BSF   09.2
02AE: BSF   03.5
02AF: BCF   09.0
02B0: BCF   03.5
02B1: BSF   09.0
02B2: BSF   03.5
02B3: BCF   05.5
02B4: BCF   03.5
02B5: BCF   05.5
02B6: MOVLW 01
02B7: MOVWF 47
02B8: CALL  0EA
02B9: BSF   03.5
02BA: BCF   09.0
02BB: BCF   03.5
02BC: BCF   09.0
02BD: BSF   03.5
02BE: BCF   05.5
02BF: BCF   03.5
02C0: BSF   05.5
02C1: MOVF  3D,W
02C2: MOVWF 4A
02C3: MOVF  3C,W
02C4: MOVWF 49
02C5: MOVF  3B,W
02C6: MOVWF 48
02C7: MOVF  3A,W
02C8: MOVWF 47
02C9: CALL  185
02CA: MOVLW 01
02CB: ADDWF 36,F
02CC: BTFSC 03.0
02CD: INCF  37,F
02CE: BTFSC 03.2
02CF: INCF  38,F
02D0: BTFSC 03.2
02D1: INCF  39,F
02D2: MOVF  39,W
02D3: MOVWF 4A
02D4: MOVF  38,W
02D5: MOVWF 49
02D6: MOVF  37,W
02D7: MOVWF 48
02D8: MOVF  36,W
02D9: MOVWF 47
02DA: CALL  17E
02DB: INCF  2D,F
02DC: GOTO  1D1
02DD: INCF  2A,F
02DE: GOTO  1C2
02DF: RETLW 00
*
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Zoom to center function

02E0: BSF 42.0
02E1: MOVLW 05
02E2: MOVWF 45
02E3: CALL 060
02E4: MOVF 2B,W
02E5: SUBLW 1E
02E6: BTFSC 03.2
02E7: GOTO 2EA
02E8: MOVLW 00
02E9: GOTO 2EA
02EA: MOVF 2C,W
02EB: SUBLW 14
02EC: BTFSC 03.2
02ED: GOTO 2F0
02EE: MOVLW 00
02EF: GOTO 2F0
02F0: MOVF 2B,W
02F1: MOVWF 45
02F2: MOVF 2C,W
02F3: MOVWF 46
02F4: CALL 18C
02F5: CALL 055
02F6: MOVLW 04
02F7: MOVWF 45
02F8: CALL 060
02F9: MOVLW 03
02FA: MOVWF 44
02FB: CALL 11D
02FC: BCF 0A.3
02FD: BCF 0A.4
02FE: GOTO 3EE
(RETURN)
*

Main program

Initial Hardware setup

02FF: CLRF 04
0300: BCF 03.7
0301: MOVLW 1F
0302: ANDWF 03,F
0303: BSF 03.5
0304: CLRF 1F
0305: MOVLW 81
0306: MOVWF 19
0307: MOVLW 26
0308: MOVWF 18
0309: MOVLW 90
030A: BCF 03.5
030B: MOVWF 18
030C: BCF 42.0
030D: BSF 42.1
030E: BCF 42.2
030F: BCF 42.3
0310: BCF 42.4
0311: BCF 42.5
0312: BCF 42.6
0313: BCF 42.7
0314: MOVLW FF
0315: MOVWF 43
0316: MOVLW 80
0317: BSF 03.5
0318: MOVWF 1F
0319: BCF 03.5
031A: MOVF 1F,W
031B: ANDLW 38
031C: IORLW C1
031D: MOVWF 1F
031E: BSF 03.5
031F: BCF 09.4
0320: BCF 03.5
0321: BCF 14.5
0322: BCF 43.5
0323: MOVF 43,W
0324: BSF 03.5
0325: MOVWF 07
0326: BCF 03.5
0327: BSF 43.4
0328: MOVF 43,W
0329: BSF 03.5
032A: MOVWF 07
032B: BCF 03.5
032C: BCF 43.3
032D: MOVF 43,W
032E: BSF 03.5
032F: MOVWF 07
0330: MOVLW 00
0331: BCF 03.5
0332: MOVWF 14
0333: BSF 03.5
0334: MOVWF 14
0335: BCF 03.5
0336: CLRF 10
0337: MOVWF 78
0338: MOVWF 12
0339: MOVLW 00
033A: BSF 03.5
033B: MOVWF 12
*

Setup interrupt using

033C: MOVLW 04
033D: ANDLW F8
033E: BTFSS 03.2
033F: GOTO 343
0340: BCF 03.5
0341: BCF 07.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0342: BSF  03.5
0343: BCF  03.5
0344: CLRF 17
0345: MOVLW 04
0346: MOVWF 17
0347: ANDLW F8
0348: BTFSC 03.2
0349: BCF  07.1
034A: CLRF 1D
034B: MOVLW 04
034C: MOVWF 1D
034D: BSF  03.5
034E: BSF  0C.2
034F: BSF  0D.0
0350: MOVLW C0
0351: BCF  03.5
0352: IORWF 0B,F
      *

Start program
0353: BCF  43.0
0354: MOVF 43,W
0355: BSF  03.5
0356: MOVWF 07
0357: BCF  03.5
0358: BSF  07.0
0359: BCF  43.3
035A: MOVF 43,W
035B: BSF  03.5
035C: MOVWF 07
035D: BCF  03.5
035E: BSF  07.3
035F: CALL 055
0360: CALL 103
0361: BCF  42.3
0362: BTFSC 42.3
0363: GOTO 412
      *

Read operation switch
0364: BSF  1F.2
0365: BTFSC 1F.2
0366: GOTO 365
0367: BSF  03.5
0368: MOVF 1E,W
0369: BCF  03.5
036A: MOVWF 3E
036B: MOVF 1E,W
036C: MOVWF 3F
036D: CLRF 40
036E: CLRF 41
036F: MOVF 2F,F
0370: BTFSC 03.2
0371: GOTO 374
0372: MOVLW 00
0373: GOTO 374
0374: MOVF 41,F
0375: BTFSS 03.2
0376: GOTO 381
0377: MOVF 40,F
0378: BTFSS 03.2
0379: GOTO 381
037A: MOVF 3F,F
037B: BTFSS 03.2
037C: GOTO 381
037D: MOVF 3E,W
037E: SUBLW 78
037F: BTFSC 03.0
0380: GOTO 383
0381: MOVLW 01
0382: MOVWF 2F
0383: MOVF 41,F
0384: BTFSS 03.2
0385: GOTO 394
0386: MOVF 40,F
0387: BTFSS 03.2
0388: GOTO 394
0389: MOVF 3F,W
038A: SUBLW 00
038B: BTFSC 03.0
038C: GOTO 396
038D: XORLW FF
038E: BTFSS 03.2
038F: GOTO 394
0390: MOVF 3E,W
0391: SUBLW 18
0392: BTFSC 03.0
0393: GOTO 396
0394: MOVLW 02
0395: MOVWF 2F
0396: MOVF 41,F
0397: BTFSS 03.2
0398: GOTO 3A7
0399: MOVF 40,F
039A: BTFSS 03.2
039B: GOTO 3A7
039C: MOVF 3F,W
039D: SUBLW 00
039E: BTFSC 03.0
039F: GOTO 3A9
03A0: XORLW FF
03A1: BTFSS 03.2
03A2: GOTO 3A7
03A3: MOVF 3E,W
03A4: SUBLW AE
03A5: BTFSC 03.0
03A6: GOTO 3A9
03A7: MOVLW 03
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

03A8: MOVWF 2F
03A9: MOVF 41,F
03AA: BTFSS 03.2
03AB: GOTO 3BA
03AC: MOVF 40,F
03AD: BTFSS 03.2
03AE: GOTO 3BA
03AF: MOVF 3F,W
03B0: SUBLW 01
03B1: BTFSC 03.0
03B2: GOTO 3BC
03B3: XORLW FF
03B4: BTFSS 03.2
03B5: GOTO 3BA
03B6: MOVF 3E,W
03B7: SUBLW 58
03B8: BTFSC 03.0
03B9: GOTO 3BC
03BA: MOVLW 04
03BB: MOVWF 2F
03BC: MOVF 41,F
03BD: BTFSS 03.2
03BE: GOTO 3CD
03BF: MOVF 40,F
03C0: BTFSS 03.2
03C1: GOTO 3CD
03C2: MOVF 3F,W
03C3: SUBLW 02
03C4: BTFSC 03.0
03C5: GOTO 3CF
03C6: XORLW FF
03C7: BTFSS 03.2
03C8: GOTO 3CD
03C9: MOVF 3E,W
03CA: SUBLW 20
03CB: BTFSC 03.0
03CC: GOTO 3CF
03CD: MOVLW 05
03CE: MOVWF 2F
03CF: MOVF 41,F
03D0: BTFSS 03.2
03D1: GOTO 3E0
03D2: MOVF 40,F
03D3: BTFSS 03.2
03D4: GOTO 3E0
03D5: MOVF 3F,W
03D6: SUBLW 02
03D7: BTFSC 03.0
03D8: GOTO 3E2
03D9: XORLW FF
03DA: BTFSS 03.2
03DB: GOTO 3E0
03DC: MOVF 3E,W
03DD: SUBLW B6
03DE: BTFSC 03.0
03DF: GOTO 3E2
03E0: MOVLW 06
03E1: MOVWF 2F
*
Operate signal
03E2: MOVF 2F,W
03E3: ADDLW F9
03E4: BTFSC 03.0
03E5: GOTO 411
03E6: ADDLW 07
03E7: GOTO 414
03E8: MOVLW 01
03E9: ADDWF 29,F
03EA: GOTO 411
Normal mode
03EB: GOTO 14E
03EC: GOTO 411
Zoom center
03ED: GOTO 2E0
03EE: GOTO 411
Shift right
03EF: MOVLW 1E
03F0: ADDWF 2B,W
03F1: MOVWF 44
03F2: MOVWF 45
03F3: MOVF 2C,W
03F4: MOVWF 46
03F5: CALL 18C
03F6: GOTO 411
Shift left
03F7: MOVLW 1E
03F8: SUBWF 2B,W
03F9: MOVWF 44
03FA: MOVWF 45
03FB: MOVF 2C,W
03FC: MOVWF 46
03FD: CALL 18C
03FE: GOTO 411
Shift down
03FF: MOVLW 14
0400: ADDWF 2C,W
0401: MOVWF 44
0402: MOVF 2B,W
0403: MOVWF 45
0404: MOVF 44,W
0405: MOVWF 46
0406: CALL 18C
0407: GOTO 411

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Shift up

0408: MOVLW 14

0409: SUBWF 2C,W

040A: MOVWF 44

040B: MOVF 2B,W

040C: MOVWF 45

040D: MOVF 44,W

040E: MOVWF 46

040F: CALL 18C

0410: GOTO 411

Read ram

0414: BCF 0A.0

0415: BCF 0A.1

0416: BSF 0A.2

0417: ADDWF 02,F

0418: GOTO 3E8

0419: GOTO 3EB

041A: GOTO 3ED

041B: GOTO 3EF

041C: GOTO 3F7

041D: GOTO 3FF

041E: GOTO 408

*

0411: GOTO 362

0412: GOTO 353

0413: SLEEP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.โปรแกรมสำหรับไมโครคอนโทรลเลอร์ 3

Initial Set Up

0000: MOVLW 00
0001: MOVWF 0A
0002: GOTO 06F
0003: NOP
0004: MOVWF 7F
0005: SWAPF 03,W
0006: BCF 03.5
0007: BCF 03.6
0008: MOVWF 21
0009: MOVF 0A,W
000A: MOVWF 20
000B: CLRF 0A
000C: MOVF 04,W
000D: MOVWF 22
000E: MOVF 77,W
000F: MOVWF 23
0010: MOVF 78,W
0011: MOVWF 24
0012: MOVF 79,W
0013: MOVWF 25
0014: MOVF 7A,W
0015: MOVWF 26
0016: MOVF 7B,W
0017: MOVWF 27
0018: BCF 03.7
0019: BCF 03.5
001A: MOVLW 8C
001B: MOVWF 04
001C: BTFSS 00.5
001D: GOTO 020
001E: BTFSC 0C.5
001F: GOTO 03D
0020: MOVLW 8C
0021: MOVWF 04
0022: BTFSS 00.2
0023: GOTO 026
0024: BTFSC 0C.2
0025: GOTO 040
0026: BTFSS 0B.5
0027: GOTO 02A
0028: BTFSC 0B.2
0029: GOTO 043
002A: MOVF 22,W
002B: MOVWF 04
002C: MOVF 23,W
002D: MOVWF 77
002E: MOVF 24,W
002F: MOVWF 78
0030: MOVF 25,W
0031: MOVWF 79
0032: MOVF 26,W
0033: MOVWF 7A
0034: MOVF 27,W
0035: MOVWF 7B
0036: MOVF 20,W
0037: MOVWF 0A
0038: SWAPF 21,W
0039: MOVWF 03
003A: SWAPF 7F,F
003B: SWAPF 7F,W
003C: RETFIE
003D: BCF 0A.3
003E: BCF 0A.4
003F: GOTO 046
0040: BCF 0A.3
0041: BCF 0A.4
0042: GOTO 04F
0043: BCF 0A.3
0044: BCF 0A.4
0045: GOTO 04A
*

Declare Parameter

0054: MOVLW 30
0055: MOVWF 04
0056: BCF 03.7
0057: MOVF 00,W
0058: BTFSC 03.2
0059: GOTO 06C
005A: MOVLW 06
005B: MOVWF 78
005C: MOVLW BF
005D: MOVWF 77
005E: CLRWDT
005F: DECFSZ 77,F
0060: GOTO 05E
0061: DECFSZ 78,F
0062: GOTO 05C
0063: MOVLW 7A
0064: MOVWF 77
0065: DECFSZ 77,F
0066: GOTO 065
0067: NOP
0068: NOP
0069: CLRWDT
006A: DECFSZ 00,F
006B: GOTO 05A
006C: BCF 0A.3
006D: BCF 0A.4
006E: GOTO 17C
*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Interrupt to clear watchdog timer

0046: BCF 0C.5
0047: BCF 0A.3
0048: BCF 0A.4
0049: GOTO 02A
004A: CLRWDT
*

CCP interrupt

004B: BCF 0B.2
004C: BCF 0A.3
004D: BCF 0A.4
004E: GOTO 02A
004F: BCF 2E.0
0050: BCF 0C.2
0051: BCF 0A.3
0052: BCF 0A.4
0053: GOTO 02A
*

Main Program

Initial Hardware setup

006F: CLRF 04
0070: BCF 03.7
0071: MOVLW 1F
0072: ANDWF 03,F
0073: BSF 03.5
0074: CLRF 1F
0075: MOVLW 81
0076: MOVWF 19
0077: MOVLW 26
0078: MOVWF 18
0079: MOVLW 90
007A: BCF 03.5
007B: MOVWF 18
007C: BCF 2E.0
007D: BCF 2E.1
007E: MOVLW FF
007F: MOVWF 2F
0080: MOVLW 06
0081: BSF 03.5
0082: MOVWF 1F
0083: BCF 03.5
0084: MOVF 1F,W
0085: ANDLW 38
0086: IORLW 01
0087: MOVWF 1F
0088: BCF 14.5
0089: BCF 2F.5
008A: MOVF 2F,W
008B: BSF 03.5
008C: MOVWF 07
008D: BCF 03.5
008E: BSF 2F.4
008F: MOVF 2F,W
0090: BSF 03.5
0091: MOVWF 07
0092: BCF 03.5
0093: BCF 2F.3
0094: MOVF 2F,W
0095: BSF 03.5
0096: MOVWF 07
0097: MOVLW 00
0098: BCF 03.5
0099: MOVWF 14
009A: BSF 03.5
009B: MOVWF 14
009C: BCF 09.4
009D: MOVLW 02
009E: MOVWF 77
009F: BTFSC 77.3
00A0: GOTO 0A3
00A1: BCF 03.5
00A2: GOTO 0AD
00A3: MOVLW 07
00A4: BCF 03.5
00A5: CLRF 01
00A6: MOVLW 81
00A7: MOVWF 04
00A8: BCF 03.7
00A9: MOVF 00,W
00AA: ANDLW C0
00AB: IORLW 0F
00AC: MOVWF 00
00AD: CLRWDT
00AE: MOVLW 81
00AF: MOVWF 04
00B0: MOVF 00,W
00B1: ANDLW C0
00B2: IORWF 77,W
00B3: MOVWF 00
00B4: CLRF 10
00B5: MOVLW 00
00B6: MOVWF 78
00B7: MOVWF 12
00B8: MOVLW 00
00B9: BSF 03.5
00BA: MOVWF 12
*

Setup Interrupt Using

00BB: MOVLW 04
00BC: ANDLW F8
00BD: BTFSS 03.2
00BE: GOTO 0C2
00BF: BCF 03.5
00C0: BCF 07.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00C1: BSF 03.5
00C2: BCF 03.5
00C3: CLRF 17
00C4: MOVLW 04
00C5: MOVWF 17
00C6: MOVLW 00
00C7: ANDLW F8
00C8: BTFSC 03.2
00C9: BCF 07.1
00CA: CLRF 1D
00CB: CLRF 1D
00CC: BSF 03.5
00CD: BSF 0C.2
00CE: BSF 0C.5
00CF: BCF 03.5
00D0: BSF 0B.5
00D1: MOVLW C0
00D2: IORWF 0B,F

Start Program

00D3: BCF 2F.1
00D4: MOVF 2F,W
00D5: BSF 03.5
00D6: MOVWF 07
00D7: BCF 03.5
00D8: BSF 07.1
00D9: BSF 03.5
00DA: BCF 01.7
00DB: BCF 03.5
00DC: MOVLW FF
00DD: BSF 03.5
00DE: MOVWF 06
00DF: MOVWF 08
00E0: BCF 05.5
00E1: BCF 03.5
00E2: BSF 05.5
00E3: BSF 03.5
00E4: BCF 09.2
00E5: BCF 03.5
00E6: BCF 09.2
00E7: BSF 03.5
00E8: BCF 09.0
00E9: BCF 03.5
00EA: BSF 09.0
*

Wait for Start Read Ram

Interrupt

00EB: BTFSS 2E.0
00EC: GOTO 0EB
00ED: GOTO 0EE

Read Start

00EE: MOVLW 00
00EF: BSF 03.5
00F0: MOVWF 08
00F1: BCF 03.5
00F2: MOVF 2B,W
00F3: SUBLW 26
00F4: BTFSC 03.2
00F5: GOTO 0F8
00F6: MOVLW 00
00F7: GOTO 0F8
00F8: MOVF 2A,F
00F9: BTFSC 03.2
00FA: GOTO 0FD
00FB: MOVLW 00
00FC: GOTO 0FD
00FD: MOVF 2D,F
00FE: BTFSC 03.2
00FF: GOTO 102
0100: MOVLW 00
0101: GOTO 102
0102: CLRF 29
0103: MOVLW 01
0104: ADDWF 2B,W
0105: SUBWF 29,W
0106: BTFSC 03.0
0107: GOTO 16E
0108: MOVF 2A,F
0109: BTFSS 03.2
010A: GOTO 10D
010B: BCF 07.0
010C: GOTO 10E
010D: BSF 07.0
010E: BCF 2F.0
010F: MOVF 2F,W
0110: BSF 03.5
0111: MOVWF 07
0112: MOVLW 01
0113: BCF 03.5
0114: MOVWF 28
0115: MOVLW 00
0116: BSF 03.5
0117: MOVWF 06
0118: BCF 03.5
0119: MOVF 28,W
011A: MOVWF 06
011B: BSF 03.5
011C: BCF 05.0
011D: BCF 03.5
011E: BSF 05.0
011F: MOVLW 00
0120: BSF 03.5
0121: MOVWF 06
0122: BCF 03.5
0123: MOVF 29,W
0124: MOVWF 06

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0125: BSF 03.5
0126: BCF 05.1
0127: BCF 03.5
0128: BSF 05.1
0129: BSF 03.5
012A: BCF 09.1
012B: BCF 03.5
012C: BSF 09.1
012D: BSF 03.5
012E: BCF 05.5
012F: BCF 03.5
0130: BCF 05.5
0131: BSF 03.5
0132: BCF 09.2
0133: BCF 03.5
0134: BSF 09.2
0135: BSF 03.5
0136: BCF 09.0
0137: BCF 03.5
0138: BCF 09.0
0139: MOVLW FF
013A: BSF 03.5
013B: MOVWF 08
013C: BCF 03.5
013D: MOVF 08,W
013E: MOVWF 2C
013F: BSF 03.5
0140: BCF 05.5
0141: BCF 03.5
0142: BSF 05.5
0143: BSF 03.5
0144: BCF 09.2
0145: BCF 03.5
0146: BCF 09.2
0147: BSF 03.5
0148: BCF 09.0
0149: BCF 03.5
014A: BSF 09.0
014B: MOVF 2D,W
014C: SUBLW 78
014D: BTFSS 03.2
014E: GOTO 15C

*

Print data to screen

014F: MOVLW 4E
0150: BTFSS 0C.4
0151: GOTO 150
0152: MOVWF 19
0153: MOVLW 52
0154: BTFSS 0C.4
0155: GOTO 154
0156: MOVWF 19
0157: MOVF 2D,F
0158: BTFSC 03.2
0159: GOTO 15C
015A: MOVLW 00
015B: GOTO 15C
015C: MOVF 2C,F
015D: BTFSS 03.2
015E: GOTO 164
015F: MOVLW 20
0160: BTFSS 0C.4
0161: GOTO 160
0162: MOVWF 19
0163: GOTO 168
0164: MOVLW 2E
0165: BTFSS 0C.4
0166: GOTO 165
0167: MOVWF 19
0168: INCF 2D,F
0169: INCF 28,F
016A: GOTO 115
016B: INCF 2A,F
016C: INCF 29,F
016D: GOTO 103

*

Read Finish and create Interrupt

016E: MOVLW FF
016F: BSF 03.5
0170: MOVWF 06
0171: MOVWF 08
0172: BCF 03.5
0173: BCF 2F.1
0174: MOVF 2F,W
0175: BSF 03.5
0176: MOVWF 07
0177: BCF 03.5
0178: BCF 07.1
0179: MOVLW 05
017A: MOVWF 30
017B: GOTO 054
017C: BCF 2F.1
017D: MOVF 2F,W
017E: BSF 03.5
017F: MOVWF 07
0180: BCF 03.5
0181: BSF 07.1
0182: BCF 2E.0
0183: GOTO 0DC
0184: SLEEP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

Devices Included in this Data Sheet:

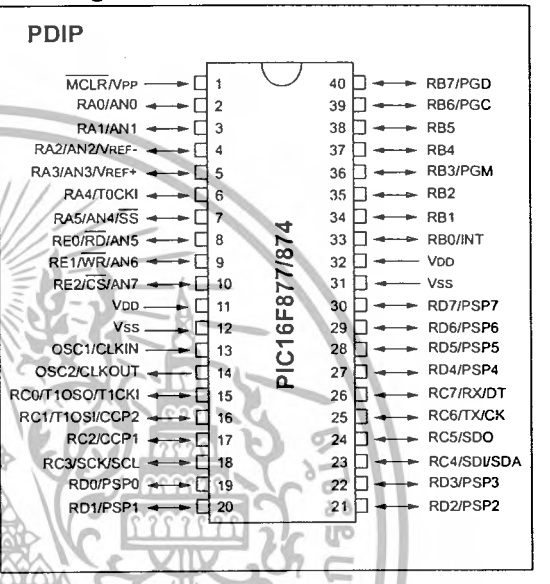
- PIC16F873
- PIC16F874
- PIC16F876
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
  Up to 368 x 8 bytes of Data Memory (RAM)
  Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and
  Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature ranges
- Low-power consumption:
  - < 0.6 mA typical @ 3V, 4 MHz
  - 20 µA typical @ 3V, 32 kHz
  - < 1 µA typical standby current

Pin Diagram

PDIP



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I²C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

© 2001 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

128Kx8 bit Low Power CMOS Static RAM

FEATURES

- Process Technology: TFT
- Organization: 128Kx8
- Power Supply Voltage: 4.5~5.5V
- Low Data Retention Voltage: 2V(Min)
- Three state output and TTL Compatible
- Package Type: 32-DIP-600, 32-SOP-525, 32-TSOP1-0820F/R

GENERAL DESCRIPTION

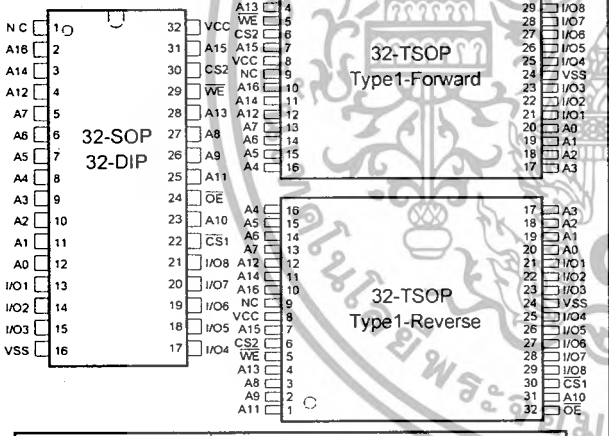
The K6T1008C2E families are fabricated by SAMSUNG's advanced CMOS process technology. The families support various operating temperature ranges and have various package types for user flexibility of system design. The families also support low data retention voltage for battery back-up operation with low data retention current.

PRODUCT FAMILY

Product Family	Operating Temperature	Vcc Range	Speed	Power Dissipation		PKG Type
				Standby (I_{SB1}, Max)	Operating (I_{CC2}, Max)	
K6T1008C2E-L	Commercial(0~70°C)	4.5~5.5V	55[1]/70ns	50µA	50mA	32-DIP-600, 32-SOP-525 32-TSOP1-0820F/R
K6T1008C2E-B				10µA		
K6T1008C2E-P	Industrial(-40~85°C)			50µA		32-SOP -525 32-TSOP1-0820F/R
K6T1008C2E-F				15µA		

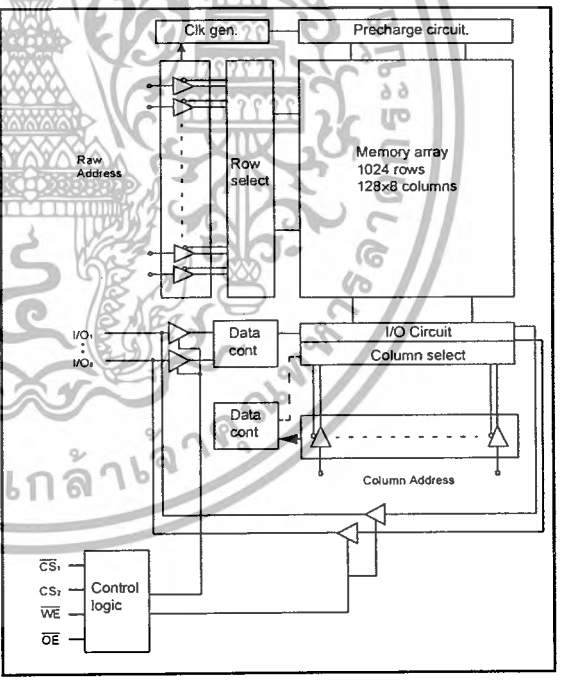
1. The parameters are tested with 50pF test load

PIN DESCRIPTION



Name	Function
$\overline{CS_1}$, CS_2	Chip Select Input
$\overline{OE}$	Output Enable Input
$\overline{WE}$	Write Enable Input
I/O_1~I/O_8	Data Inputs/Outputs
A_0~A_{16}	Address Inputs
Vcc	Power
Vss	Ground
N.C.	No Connection

FUNCTIONAL BLOCK DIAGRAM



SAMSUNG ELECTRONICS CO., LTD. reserves the right to change products and specifications without notice.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

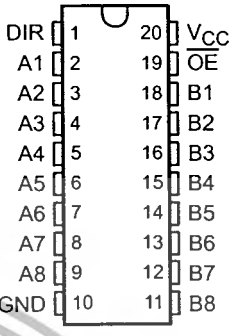
SN54LS245, SN74LS245
OCTAL BUS TRANSCEIVERS WITH 3-STATE OUTPUTS

SDLS146A – OCTOBER 1976 – REVISED FEBRUARY 2002

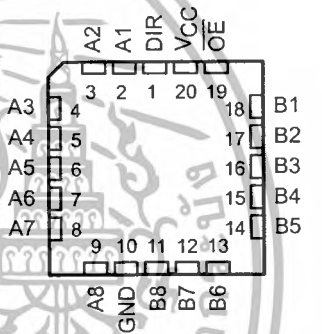
- **3-State Outputs Drive Bus Lines Directly**
- **PNP Inputs Reduce dc Loading on Bus Lines**
- **Hysteresis at Bus Inputs Improves Noise Margins**
- **Typical Propagation Delay Times Port to Port, 8 ns**

TYPE	I_{OL} (SINK CURRENT)	I_{OH} (SOURCE CURRENT)
SN54LS245	12 mA	–12 mA
SN74LS245	24 mA	–15 mA

SN54LS245 . . . J OR W PACKAGE
SN74LS245 . . . DB, DW, N, OR NS PACKAGE
(TOP VIEW)



SN54LS245 . . . FK PACKAGE
(TOP VIEW)



description

These octal bus transceivers are designed for asynchronous two-way communication between data buses. The control-function implementation minimizes external timing requirements.

The devices allow data transmission from the A bus to the B bus or from the B bus to the A bus, depending on the logic level at the direction-control (DIR) input. The output-enable ($\overline{OE}$) input can disable the device so that the buses are effectively isolated.

ORDERING INFORMATION

T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP – N	Tube	SN74LS245N	SN74LS245N
	SOIC – DW	Tube	SN74LS245DW	LS245
		Tape and reel	SN74LS245DWR	
	SOP – NS	Tape and reel	SN74LS245NSR	74LS245
	SSOP – DB	Tape and reel	SN74LS245DBR	LS245
–55°C to 125°C	CDIP – J	Tube	SN54LS245J	SN54LS245J
		Tube	SNJ54LS245J	SNJ54LS245J
	CFP – W	Tube	SNJ54LS245W	SNJ54LS245W
	LCCC – FK	Tube	SN54LS245FK	SN54LS245FK

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2002, Texas Instruments Incorporated

On products compliant to MIL-PRF-38535, all parameters are tested unless otherwise noted. On all other products, production processing does not necessarily include testing of all parameters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SN54HC573A, SN74HC573A
OCTAL TRANSPARENT D-TYPE LATCHES WITH 3-STATE OUTPUTS

SCLS147C – DECEMBER 1982 – REVISED MARCH 2001

- **High-Current 3-State Outputs Drive Bus Lines Directly or up to 15 LSTTL Loads**
- **Bus-Structured Pinout**

description

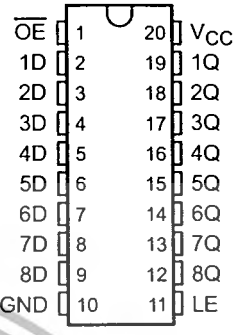
These octal transparent D-type latches feature 3-state outputs designed specifically for driving highly capacitive or relatively low-impedance loads. They are particularly suitable for implementing buffer registers, I/O ports, bidirectional bus drivers, and working registers.

While the latch-enable (LE) input is high, the Q outputs respond to the data (D) inputs. When LE is low, the outputs are latched to retain the data that was set up.

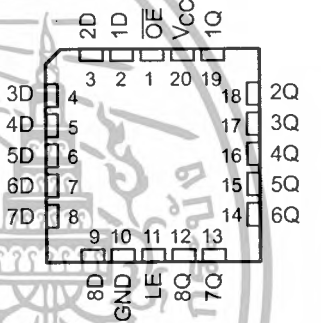
A buffered output-enable ($\overline{OE}$) input can be used to place the eight outputs in either a normal logic state (high or low logic levels) or the high-impedance state. In the high-impedance state, the outputs neither load nor drive the bus lines significantly. The high-impedance state and increased drive provide the capability to drive bus lines without interface or pullup components.

$\overline{OE}$ does not affect the internal operations of the latches. Old data can be retained or new data can be entered while the outputs are in the high-impedance state.

SN54HC573A . . . J OR W PACKAGE
SN74HC573A . . . DB, DW, N, OR PW PACKAGE
(TOP VIEW)



SN54HC573A . . . FK PACKAGE
(TOP VIEW)



ORDERING INFORMATION

T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
–40°C to 85°C	PDIP – N	Tube	SN74HC573AN	SN74HC573AN
	SOIC – DW	Tube	SN74HC573ADW	HC573A
		Tape and reel	SN74HC573ADWR	
	SSOP – DB	Tape and reel	SN74HC573ADBR	HC573A
	TSSOP – PW	Tape and reel	SN74HC573APWR	HC573A
–55°C to 125°C	CDIP – J	Tube	SNJ54HC573AJ	SNJ54HC573AJ
	CFP – W	Tube	SNJ54HC573AW	SNJ54HC573AW
	LCCC - FK	Tube	SNJ54HC573AFK	SNJ54HC573AFK

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2001, Texas Instruments Incorporated
On products compliant to MIL-PRF-38535, all parameters are tested unless otherwise noted. On all other products, production processing does not necessarily include testing of all parameters.

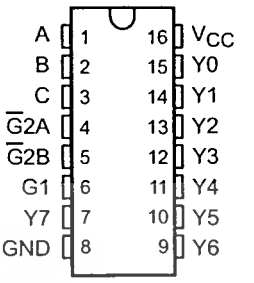
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SN54HC138, SN74HC138
3-LINE TO 8-LINE DECODERS/DEMULTIPLEXERS

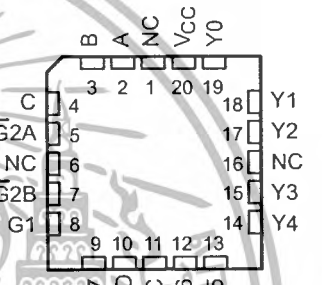
SCLS107D – DECEMBER 1982 – REVISED DECEMBER 2002

- Targeted Specifically for High-Speed Memory Decoders and Data-Transmission Systems
- Wide Operating Voltage Range of 2 V to 6 V
- Outputs Can Drive Up To 10 LSTTL Loads
- Low Power Consumption, 80-μA Max I_{CC}
- Typical t_{pd} = 15 ns
- ±4-mA Output Drive at 5 V
- Low Input Current of 1 μA Max
- Incorporate Three Enable Inputs to Simplify Cascading and/or Data Reception

SN54HC138 . . . J OR W PACKAGE
SN74HC138 . . . D, DB, N, NS, OR PW PACKAGE
(TOP VIEW)



SN54HC138 . . . FK PACKAGE
(TOP VIEW)



NC – No internal connection

description/ordering information

The 'HC138 devices are designed to be used in high-performance memory-decoding or data-routing applications requiring very short propagation delay times. In high-performance memory systems, these decoders can be used to minimize the effects of system decoding. When employed with high-speed memories utilizing a fast enable circuit, the delay times of these decoders and the enable time of the memory are usually less than the typical access time of the memory. This means that the effective system delay introduced by the decoders is negligible.

The conditions at the binary-select inputs at the three enable inputs select one of eight output lines. Two active-low and one active-high enable inputs reduce the need for external gates or inverters when expanding. A 24-line decoder can be implemented without external inverters, and a 32-line decoder requires only one inverter. An enable input can be used as a data input for demultiplexing applications.

ORDERING INFORMATION

T_A	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
–40°C to 85°C	PDIP – N	Tube	SN74HC138N	SN74HC138N
	SOIC – D	Tube	SN74HC138D	HC138
		Tape and reel	SN74HC138DR	
	SOP – NS	Tape and reel	SN74HC138NSR	HC138
	SSOP – DB	Tape and reel	SN74HC138DBR	HC138
	TSSOP – PW	Tube	SN74HC138PW	HC138
		Tape and reel	SN74HC138PWR	
–55°C to 125°C	CDIP – J	Tube	SNJ54HC138J	SNJ54HC138J
	CFP – W	Tube	SNJ54HC138W	SNJ54HC138W
	LCCC – FK	Tube	SNJ54HC138FK	SNJ54HC138FK

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2002, Texas Instruments Incorporated
On products compliant to MIL-PRF-38535, all parameters are tested unless otherwise noted. On all other products, production processing does not necessarily include testing of all parameters.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้สำเร็จลุล่วงไปได้ ด้วยคำแนะนำและคำปรึกษาจากท่าน รศ.ดร. วันชัย ริ้วรุจา ผู้ทำปริญญานิพนธ์รู้สึกทราบซึ้งในความอนุเคราะห์จากท่านและขอการบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณบิดา มารดา ครอบครัว และเพื่อนๆของข้าพเจ้าที่คอยให้กำลังใจมาโดยตลอด

ขอขอบคุณ นาย ธวัชชัย คำศรี ด้วยคำแนะนำและให้คำปรึกษาจากท่าน ทำให้ปริญญานิพนธ์เล่มนี้สำเร็จลุล่วง คุณค่าและประโยชน์อันพึงมีจากปริญญานิพนธ์นี้ ขอมอบแด่ผู้มีพระคุณทุกท่าน

นายพีรพล พัฒนวิจิตร

นายวินัย แก้วโกวิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] " PIC 16F87X datasheet " 2001 Microchip Technology Inc.

[2] ประจิน พลังสันติกุล , "เรียนรู้และใช้งาน CCS C คอมไพเลอร์ เขียนโปรแกรมภาษา C ควบคุมไมโครคอนโทรลเลอร์ PIC"

[3] Nigel Gardner, PIC micro MCU C An introduction to programming The Microchip PIC in CCS C

Internet Resource List

http://www.microchip.com

http://www.thaimcu.com

http://www.thaiio.com

http://www.es.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้