เครื่องแกะฮาร์ดล็อก

HARDLOCK SIMULATOR

นายกติพงศ์ ศิริสวัสดิ์ 38012001

ปริญญานิพนธ์สำหรับปริญญาอุตสาหกรรมศาสตร์บัณฑิต

สาขาเทคโนโลยีอิเล็กทรอนิกส์

ภาควิชาเทคนิคอุตสาหกรรม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2539

ภาควิชา เทคนิคอุตสาหกรรม

สาขา เทคโนโลยีอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องแกะฮาร์ดล็อก

NEVER LOCK

ผู้จัดทำ

1. นายกติพงศ์ ศิริสวัสดิ์ 38012001

..... อาจารย์ที่ปรึกษา

(อาจารย์ไพศาล สิทธิโยภาสกุล)

..... อาจารย์ที่ปรึกษา

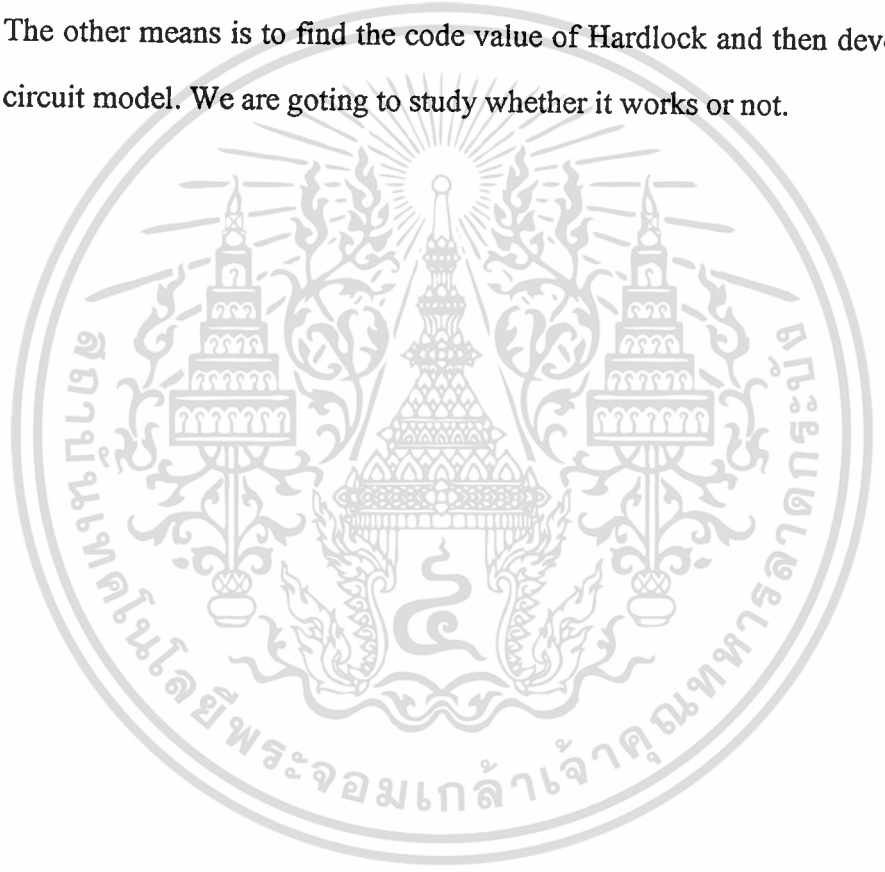
(อาจารย์บุญย์ชนะ ภู่ระหงษ์)

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เสนอการทำงานของฮาร์ดล็อก (Hardlock) และวิธีการแกะฮาร์ดล็อก โดยการแก้ที่ Software ซึ่งลักษณะการทำงานของ software ตรวจสอบฮาร์ดล็อกมีอยู่ด้วยกันหลายแบบ เวลาที่จะใช้ในการแกะโค็ดของฮาร์ดล็อกแต่ละตัวนั้นไม่เท่ากันขึ้นอยู่กับความยากง่ายของ software และยังคงมีอีกวิธีที่ไม่มีคนทำกันคือหาค่าโค็ดออกมาว่าโค็ดของ Hardlock คืออะไรแล้วทำวงจรฮาร์ดล็อกจำลองขึ้นมาใช้งานแทน โดยเราจะมาศึกษากันว่าสามารถทำได้หรือไม่

ABSTRACT

This thesis presents about the function of Hardlock and means to remove it by rectifying the software. There are vaious functions of software examining Hardlock. The time used to uncode each Hardlock differs which depends on the feasibility of the software. The other means is to find the code value of Hardlock and then develop the Hardlock circuit model. We are goting to study whether it works or not.



สารบัญ

หน้า

บทที่ 1
บทนำ

1.1 ชื่อโครงการ HardLock Simulator (เครื่องสร้างฮาร์ดล็อกจำลอง)

1.2 วัตถุประสงค์

1) เพื่อศึกษาการทำงานของอุปกรณ์ป้องกันการก๊อปปี้ (Hardlock) แบบต่างๆ จนสามารถสร้างอุปกรณ์จำลองขึ้นมาแทนของจริงได้
2) เพื่อช่วยให้สามารถใช้ซอฟแวร์ (Software) ราคาแพงได้โดยที่ไม่จำเป็นต้องใช้เงินมากขนาดเท่ากับการซื้อของใหม่มาใช้

1.3 แนวความคิดและที่มา

ถึงแม้ว่าในปัจจุบันราคาเครื่องคอมพิวเตอร์จะราคาถูกลง แต่ราคาของซอฟแวร์เฉพาะด้านที่มีการใช้งานเฉพาะกลุ่ม เช่น ซอฟแวร์ทางวิศวกรรม เป็นต้น หาได้มีราคาถูกลงตามไปด้วย ซึ่งโดยปกติงานทางด้านวิศวกรรมในปัจจุบันจำเป็นอย่างยิ่งที่จะต้องมีการใช้คอมพิวเตอร์เข้ามาเกี่ยวข้องด้วยกันทั้งสิ้น ดังนั้นซอฟแวร์ทางวิศวกรรมจึงมีความจำเป็นอย่างมาก แต่การจะจัดหาซอฟแวร์มาใช้นั้นจะต้องใช้ค่าใช้จ่ายสูงมาก บางโปรแกรมอาจถึงหนึ่งแสนบาท ซึ่งเป็นงบประมาณที่สูงเกินไปกับนักศึกษา หากจะขอก๊อปปี้จากผู้ที่มีอยู่แล้วหรือกับทางสถานศึกษาก็ไม่สามารถทำได้เนื่องจากว่าซอฟแวร์มีการตรวจสอบฮาร์ดล็อก ถ้าหากว่าไม่มีฮาร์ดล็อกก็ไม่สามารถใช้โปรแกรมได้

ด้วยเหตุผลข้างต้นนี้เอง จึงทำให้ได้คิดว่าน่าจะมีการสร้างอุปกรณ์จำลองขึ้นมาแทน Hardlock ได้ ซึ่งปกติเราก็เคยได้ยินว่ามีการแกะ Hardlock กันแล้ว แต่เป็นการแก้ที่ตัวโปรแกรมเอง โยไม่ให้มีการตรจสอบ Hardlock และด้วยวิธีการนี้ทำให้ในบางครั้งโปรแกรมที่ทำการแก้ไขเกิดข้อผิดพลาดได้ ดังนั้นจึงเล็งเห็นว่าควรจะมีการแก้ไขที่ตัว

Hardware มากกว่าแทนที่จะไปแก้ไขที่ Software จึงทำให้เกิดแนวความคิดที่จะสร้างเครื่องอ่านค่า Hardlock ขึ้นมา

1.4 โครงงานประกอบด้วยส่วนประกอบสำคัญ 2 ส่วนใหญ่ๆ คือ

1) ส่วนที่เป็น HARDWARE คือตัว BOARD MICROCONTROLLER ที่ใช้ในการดักค่าข้อมูลที่ส่งมาจากเครื่องคอมพิวเตอร์ และส่งกลับจาก HARDLOCK

2) ส่วนที่เป็น SOFTWARE แบ่งออกเป็น 2 ส่วนย่อยๆ คือ

- ◆ โปรแกรมควบคุมการทำงานของ BOARD MICROCONTROLLER
- ◆ โปรแกรมที่ใช้ในการรับและแปรข้อมูลที่ถูกส่งมาจาก BOARD MICROCONTROLLER

1.5 ผู้รับผิดชอบโครงการ

นายกติพงศ์ ศิริสวัสดิ์ รหัส 38012001
ภาควิชา เทคนิคอุตสาหกรรม
สาขาวิชา เทคโนโลยีอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์

1.6 ระยะเวลาดำเนินการ

เริ่ม มิถุนายน 2539 - มีนาคม 2540

1.7 แผนการดำเนินงานและระยะเวลาดำเนินโครงการ

รายละเอียดการดำเนินงาน	ช่วงระยะเวลาดำเนินงาน										
	มิ.ย	ก.ค	ส.ค	ก.ย	ต.ค	พ.ย	ธ.ค	ม.ค	ก.พ	มี.ค	เม.ย
ศึกษาข้อมูล	←	→									
ออกแบบวงจร		←	→								
ทดสอบวงจร			←	→							
ออกแบบ Software				←	→						
ทดสอบ Software						←	→				
ทดสอบการทำงานและปรับปรุง							←	→		→	
เขียนปริญญานิพนธ์						←	→			→	

1.8 ผลที่คาดว่าจะได้รับ

1) ได้ความรู้และทักษะในการออกแบบและการเขียนโปรแกรม
2) ได้ความรู้และทักษะการใช้ไมโครคอนโทรลเลอร์ชิพเดี่ยวเบอร์ #8751
3) รู้จักการใช้ชิพซัพพอร์ท #8255
4) รู้จักการอินเตอร์เฟส IBM PC
5) โปรแกรมทางด้านวิศวกรรมถูกใช้งานอย่างกว้างขวางมากขึ้น

1.9 งบประมาณ

2,000 บาท

1.10 <u>เครื่องมือที่จำเป็น</u>

1) BOARD MICROCONTROLLER 8031

2) LOGIC ANALIZER

3) LOGIC PROBE 40 MHz

4) HARDLOCK และ SOFTWARE ต้นฉบับ

5) SOFTWARE SOFTICE ของ NUMEGA

บทที่ 2
ทฤษฎีและการใช้งาน

ในการที่เราจะทำโครงงานขึ้นมานั้น ในบางครั้งผู้ทำก็มักประสบปัญหาเกี่ยวกับการนำอุปกรณ์มาประยุกต์ใช้ ดังนั้นเราจะต้องทำการศึกษาและเรียนรู้การประยุกต์ใช้งานอุปกรณ์ต่างๆ ที่คาดว่าจะต้องนำมาใช้ในการทำโครงงานนั้นๆ ซึ่งในบทนี้จะได้กล่าวถึงสถาปัตยกรรมของ 8051 และการสื่อสารด้วย RS-232

2.1 สถาปัตยกรรมของไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล 51

(Single Chip Microcontroller system-51 family Architectural)

ไมโครคอนโทรลเลอร์แบบชิพเดี่ยว (Single Chip Microcontroller) คือไมโครคอมพิวเตอร์แบบที่มีขนาดเล็กโดยบรรจุไว้ในแผงวงจรรวม (Integrated Circuit) เพียงชิพเดียวเหมาะสำหรับงานควบคุมอุปกรณ์อื่นๆ แบบอัตโนมัติ เพราะผู้ใช้สามารถเขียนโปรแกรมควบคุมการทำงานได้ตามต้องการ ไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล 51 หรือ MCS51 อันได้แก่ เบอร์ 8051 และ 8052 ซึ่งมีโครงสร้างและชุดคำสั่งแตกต่างกันเล็กน้อยดังตารางที่ 2.1

ตารางที่ 2.1 ตารางของไมโครคอนโทรลเลอร์แบบชิพเดี่ยวตระกูล 51

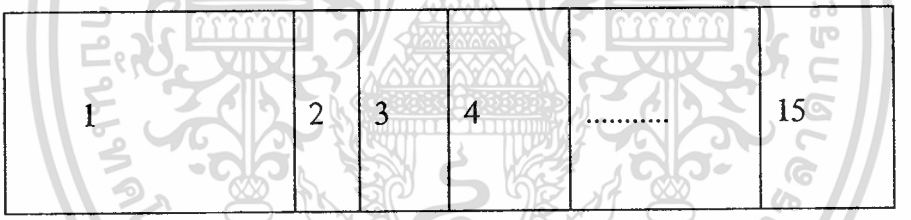
EMBEDDED CONTROLLERS

Feature	8051AH	8031AH	8751H	80C51BH	80C31BH	87C51	8052AH	8032AH	8752	8044H
Program Memory (Bytes)	4k	–	4k	4k	–	4k	8k	–	8k	8k
RAM (Bytes)	128	128	128	128	128	128	256	256	256	192
Program Memory Expansion (Off Chip) (Bytes)	64k	64k	64k	64k	64k	64k	64k	64k	64k	64k
Max Clock Frequency (Mhz)	12	12	12	16	16	16	16	12	12	12
Typical instruction Time (uS)	1	1	1	0.75	0.75	0.75	1	1	1	1
16-Bit Timer/Counters	2	2	2	2	2	2	3	3	3	2
Serial Communications	Synchronous Mode, Asychronous 9 or 10 – Bit Programable									HDLC
No. of I/O Lines	32	16	32	32	16	32	32	16	32	32
Interrupt Sources (Two Priority Levels)	5	5	5	5	5	5	6	6	6	5
Power Requirements (ICC Max mA.)	125	250	24	24	29	175	175	175	200	
Programmable Power Modes Idle Power Down	–	–	–	–	4.0 mA 50 uA	4.0 mA 51 uA	4.0 mA 52 uA	–	–	– 30mA

จากคำสั่ง (Instruction) ตามที่มีการกำหนดไว้ และสัญญาณที่สร้างขึ้นมาจะอ้างอิงกับสัญญาณนาฬิกาที่สร้างจากวงจรออสซิลเลเตอร์เพื่อให้ทุก ๆ ส่วนในวงจรทำงานประสานกัน (Synchronize) อย่างถูกต้อง

ใน CPU นี้ยังประกอบด้วยส่วนย่อยอีกส่วนที่เรียกว่าส่วนประมวลผล (Arithmetic Logic Unit) ส่วนนี้จะทำหน้าที่ประมวลผลข้อมูลเช่น การบวก, ลบ, คูณ หรือหารข้อมูลแล้วนำผลลัพธ์ไปเก็บไว้ในรีจิสเตอร์หรือหน่วยความจำที่ต้องการ

ส่วนที่ 2 คือ หน่วยความจำ (Memory) มีไว้สำหรับจดจำข้อมูล ถ้าจะให้เห็นภาพพจน์ของหน่วยความจำได้ดีก็คือ หน่วยความจำเปรียบเหมือนกล่องเก็บเอกสารจำนวนมากที่นำมาต่อเรียงกันไว้ แต่ละกล่องก็มีเอกสาร 1 แผ่น ดังในรูปที่ 2.2 มีกล่องเอกสารทั้งหมด 15 กล่อง



รูปที่ 2.2 ภาพเสมือนของหน่วยความจำ

ถ้าต้องการเอาเอกสารจากกล่องใด หรือเอาเอกสารไปเก็บที่กล่องใด จะต้องรู้หมายเลขของกล่องข้อมูลเสียก่อน ซึ่งถ้าเป็นหน่วยความจำแล้วหมายเลขของกล่องก็คือตำแหน่งของหน่วยความจำหรือแอสเดรส (Address) นั่นเอง การนำเอาข้อมูลไปเก็บในหน่วยความจำเรียกว่าการเขียน (Write) ข้อมูล และการเอาข้อมูลออกจากหน่วยความจำจะเรียกว่าการอ่าน (Read)ข้อมูล ซึ่งแต่ละตำแหน่งของหน่วยความจำจะเก็บข้อมูลได้เพียงค่าเดียวเท่านั้น ในไมโครโปรเซสเซอร์ทั่วไปรวมทั้ง 8051 นั้นข้อมูลในแต่ลำตำแหน่งของหน่วยความจำจะมีค่าได้เพียง 8 หลักของเลขฐาน 2 (8 บิทเท่ากับ 1 ไบท์) ดังนั้นแต่ละตำแหน่งของหน่วยความจำจะเก็บข้อมูลมีค่าได้

ได้ระหว่าง0 ถึง 255 (00000000 ถึง 11111111 ในเลขฐาน 2) แต่จำนวนตำแหน่งที่เก็บข้อมูลได้ขึ้นกับไมโครโปรเซสเซอร์แต่ละเบอร์ การติดต่อกับหน่วยความจำต้องมีสัญญาณ 3 กลุ่มคือ

1) แอสเดรสหรือค่าตำแหน่งที่ต้องการติดต่อกับหน่วยความจำ ใน 8051 จะติดต่อกับหน่วยความจำประเภท Program Memory หรือ Data Memory ได้สูงสุดชนิดละ 65536 ตำแหน่งดังนั้น การอ้างอิงแต่ละตำแหน่งของหน่วยความจำ จะต้องใช้เส้นแสดงตำแหน่งในเลขฐาน 2ทั้งหมด 16 เส้น(2^{16} เท่ากับ 64*1024=65536)

2) ข้อมูลที่จะอ่านหรือเขียนกับหน่วยความจำที่ตำแหน่งในข้อ 1

3) สัญญาณควบคุมที่จะส่งไปยังหน่วยความจำ เพื่อบอกกับหน่วยความจำที่ว่าต้องการอ่านหรือเขียนข้อมูล

สัญญาณเหล่านี้จะถูกวงจรควบคุมภายใน 8051 สร้างมาจากวงจรถอดรหัสของคำสั่งที่ 8051 อ่านจากหน่วยความจำProgram Memoryเข้าไปทำงานนั่นเอง ในรูปที่ 2.1 หน่วย ความจำ ได้แก่ 4K ROM และ 128 Byte RAM ซึ่งขนาดของหน่วยความจำนี้มีขนาดต่าง ๆ กันตามเบอร์ไมโครคอนโทรลเลอร์ และจะอธิบายโดยละเอียดในข้อ 2

ส่วนที่ 3 อุปกรณ์อินพุทและเอาท์พุท (Input/Output Device) เป็นส่วนที่จะใช้ส่งข้อมูลเข้าหรือออกจาก 8051 ทำให้ 8051 ติดต่อกับภายนอกได้ดังในไดอะแกรมรูปที่2.1 อุปกรณ์อินพุทและเอาท์พุทได้แก่ 4 I/OPort Timer 1,Serial Port การทำงานของแต่ละส่วนมีดังนี้

1) 4 I/O Port คำว่าพอร์ทหมายถึงจุดที่จะติดต่อกับส่วนที่อยู่ภายนอก 4 I/O Portของ 8051เป็นที่ใช้สำหรับรับ-ส่งข้อมูล ซึ่งเป็นสัญญาณดิจิตอลเข้าหรือออกจากตัว MCS-51 พอร์ทมีทั้งหมด 4 พอร์ท โดยแต่ละพอร์ทจะรับ-ส่งมูลได้ 8 บิท มีพอร์ท P0,P1,P2 และP3 บางพอร์ทจะใช้ทำงานมากกว่า 1 อย่างก็ได้ เช่นพอร์ท P0 และ P2 จะใช้สำหรับการส่งค่าตำแหน่ง(Address) ของหน่วยความจำที่ต้องการติดต่อและพอร์ท P0 จะใช้รับส่งข้อมูลเมื่อติดต่อกับหน่วย

ความจำได้ด้วย แต่สิ่งเหล่านี้ไม่ได้เกิดขึ้นที่เวลาเดียวกัน แต่จะใช้วิธีทำงานตามลำดับ โดยควบคุมจากสัญญาณควบคุม (Control) ที่ถอด

1) รหัสมาจากแต่ละคำสั่งที่ให้คอมพิวเตอร์ทำงานนั่นเอง และสัญญาณทั้งหมดจะอ้างอิงกับจากสัญญาณนาฬิกา

2) Timer 0 และ Timer 1 เป็นวงจรนับที่สามารถกำหนดให้ทำการนับจำนวนไซเคิลของสัญญาณที่ต่อจากภายนอก 8051 หรือจำนวนไซเคิลของสัญญาณนาฬิกาภายใน 8051 ก็ได้ค่าจากการนับจะถูกอ่านหรือตั้งค่าเริ่มต้นของการนับได้โดย CPU

3) Serial port หรือพอร์ทอนุกรม CPU จะอ่านและเขียนข้อมูลกับ Serial Port เป็นแบบ 8 บิทแต่ข้อมูลจะถูกส่งออกจาก8051 เรียงไปทีละบิทออกจากขา TXD และในการรับข้อมูลเข้าก็จะรับเข้ามาทีละบิททางขา RXD แล้วจัดเรียงใหม่เป็น 8 บิทเพื่อให้ CPU อ่านไปใช้งานต่อไป

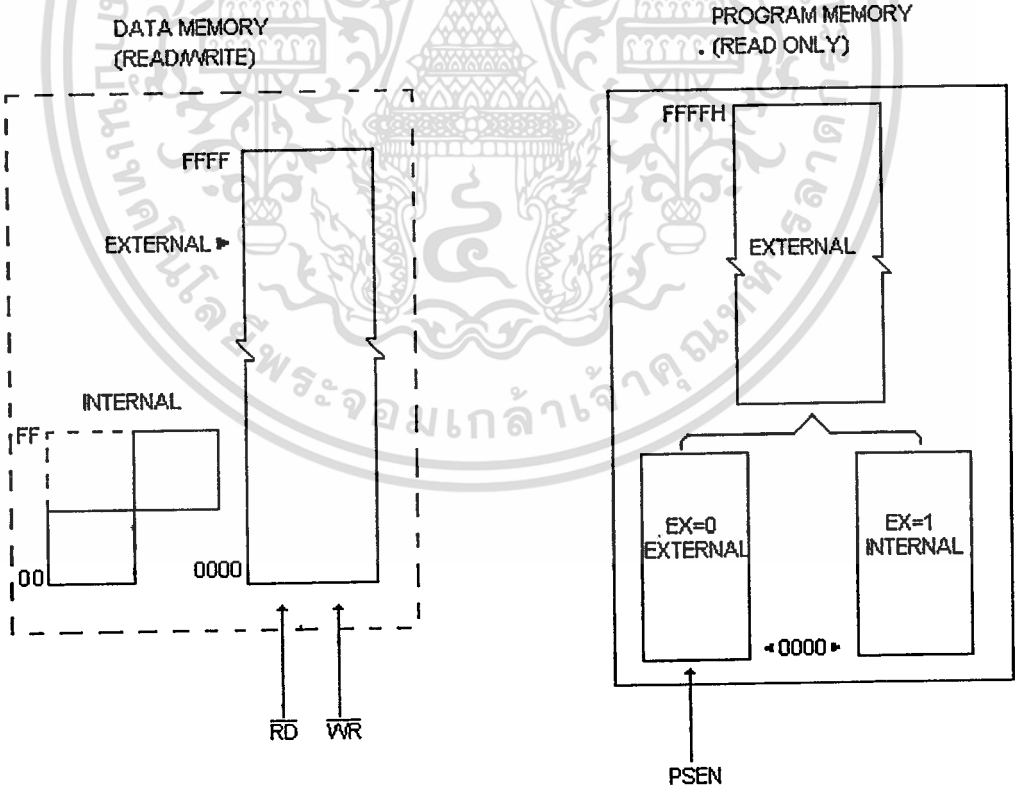
8051 มีพอร์ทให้ใช้งานได้หลายแบบทำให้สะดวกแก่การนำไปใช้งานต่าง ๆ มากมาย การจะนำพอร์ทเหล่านี้ ไปใช้งานได้จะต้องเขียนโปรแกมขึ้นมาควบคุม

2.1.2 การจัดการหน่วยความจำของ 8051

หน่วยความจำของ 8051 แบ่งออกไว้เป็น 2 แบบตามลักษณะของการใช้งาน คือ

1) Program Memory เป็นหน่วยความจำที่ใช้เก็บคำสั่งในรูปรหัสภาษาเครื่อง (Machine Language) ซึ่งต้องการให้ 8051 ทำงาน เมื่อ 8051 ทำงานก็จะอ่านข้อมูลที่เก็บในหน่วยความจำประเภทนี้เข้าไปถอดรหัสแล้วสร้างสัญญาณควบคุมส่วนอื่น ๆ ตามการทำงานของแต่ละคำสั่งนั้น หน่วยความจำแบบนี้จะต้องเป็นแบบ Read Only Memory (ROM) และผู้ใช้ต้องเขียนข้อมูลในแต่ละตำแหน่งของหน่วยความจำเป็นรหัสภาษาเครื่องของ 8051 ตามลำดับการทำงานที่ต้องการ (หน่วยความจำแบบ ROM เป็นแบบ Non volatite ซึ่งเมื่อปิดไฟแล้วข้อมูลก็ไม่มีการสูญหาย) การเขียนข้อมูลลงบน ROM จะต้องใช้เครื่องมือพิเศษในระหว่างการทำงานของ 8251ผู้ใชจะไม่สามารถใช้คำสั่งทำการเขียนข้อมูลในหน่วยความจำแบบนี้ได้ จำนวน

ตำแหน่งสูงสุดของหน่วยความตำแบบนี้ที่ 8051 จะใช้งานได้คือ 65536 ตำแหน่ง ค่าของตำแหน่ง (Addreess)จะเขียนเป็นเลขฐาน 16 ได้ตั้งแต่ 0000 H ถึง FFFFH หน่วยความจำตำแหน่ง 0000H ถึง FFFFHจำนวน 4 กิโลไบท์ นั้นผู้ใช้จะเลือกได้ว่าเป็นตำแหน่งของ ROM ที่อยู่ภายในหรือภายนอก 8051(ไมโครคอนโทรลเลอร์เบอร์อื่น ๆ เช่น 8052 จะมีขนาดของ ROM ส่วนนี้ได้ถึง 8 กิโลไบท์ ตำแหน่ง 0000H ถึง 1FFFH) ถ้าต้องการให้ 8051 ทำงานตามคำสั่งที่เก็บไว้ใน ROM ภายใน8051 ก็ให้ป้อนสัญญาณสภาวะลอจิก High(1) เข้าที่ขา EA ของ 8051 แต่ถ้าต้องการให้ทำงานในโปรแกรมที่เก็บไว้ใน ROM ภายนอก 8051 ก็ให้ต่อลอจิก Low (0) เข้าที่ขา EA ของ 8051ส่วนหน่วยความจำที่ตำแหน่ง 1FFFH ถึง FFFFH จะต้องต่ออยู่ภายนอก 8051 เสมอ ดังแสดงในแผนภูมิหน่วยความจำ (Memory Map) ในรูปที่ 2.3



รูปที่ 2.3 แผนภูมิหน่วยความจำของ 8051

Internal Memory หมายถึงหน่วยความจำนั้นอยู่ภายใน 8051 ส่วน External Memory หมายถึงหน่วยความจำนั้นอยู่ภายนอก 8051

ไมโครคอนโทรลเลอร์เบอร์ 8031, 8051 และ 8751 นั้น โดยโครงสร้างและรหัสคำสั่งจะเหมือนกันทุกประการแตกต่างกันที่

ก) 8031 จะไม่มี ROM ขนาด 4 กิโลไบท์อยู่ภายใน ผู้ใช้จะต้องเลือกการใช้งาน Program Memory อยู่ภายนอกวงจรรวมทั้งหมด 64 กิโลไบท์

ข) 8051 จะมี ROM ขนาด 4 กิโลไบท์อยู่ภายใน ถ้าต้องการเก็บคำสั่งควบคุมการทำงานไว้ในหน่วยความจำส่วนนี้ จะต้องส่งโปรแกรมคำสั่งไปให้โรงงานผู้ผลิตทำการเขียนใส่ใน ROM ให้ตั้งแต่ในขั้นตอนของการผลิตวงจรรวม ผู้ใช้ไม่สามารถแก้ไขโปรแกรมได้เอง ถ้าจะนำมาใช้งานโดยเก็บโปรแกรมไว้ในหน่วยความจำช่วง 4 กิโลไบท์แรกอยู่ภายนอกก็สามารถทำได้ โดยการต่อ ROM ไว้ภายนอก แล้วต่อขา EA ของ 8051 ไว้กับสัญญาณที่มีสภาวะลอจิกเป็น 0

ค) 8751 จะมีหน่วยความจำขนาด 4 กิโลไบท์เป็นแบบ EPROM (Erasable Program Read Only Memory) อยู่ภายในวงจรรวมเอาไว้ ใช้เก็บโปรแกรมคำสั่งที่จะให้ 8751 ทำงานผู้ใช้สามารถเขียนคำสั่งลงไปใน EPROM ได้เองโดยใช้เครื่องมือที่เรียกว่าเครื่องโปรแกรม EPROM (EPROM Programer) และผู้ใช้สามารถแก้ไขโปรแกรมที่อยู่ใน EPROM ได้โดยการล้างข้อมูลในทุกตำแหน่งของ EPROM ออกด้วยการฉายแสงอุลตร้าไวโอเลต (Ultraviolet) ผ่านกระจกใสบนวงจรรวมเข้าไปยังวงจรภายใน ตามเวลาที่กำหนดในคู่มือเฉพาะ(Data sheet) ของ 8751 จากนั้นก็ใช้เครื่องโปรแกรม EPROM เขียนโปรแกรมลงไปใหม่ 8751 นี้จะสะดวกมากสำหรับการพัฒนาโปรแกรม

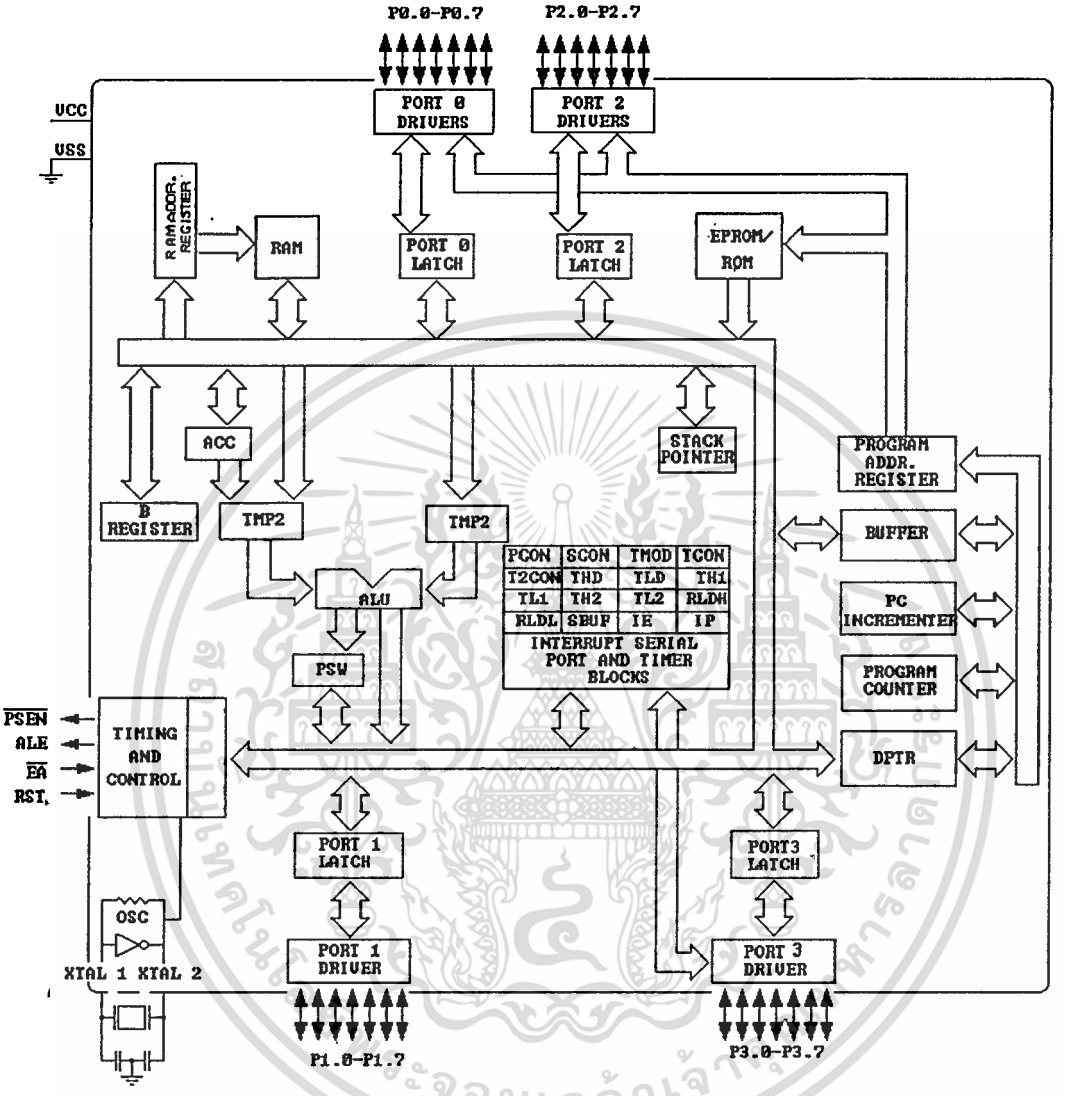
2) Data Memory เป็นหน่วยความจำที่ 8051 จะใช้สำหรับพักเก็บข้อมูล แล้วเรียกมาใช้ใหม่ในระหว่างการทำงานของ 8051 การอ่านหรือเขียนข้อมูลจากหน่วยความจำจะกระทำโดยคำสั่งที่เก็บไว้ใน Program Memory หน่วยความจำแบบนี้เป็นประเภท Random Access Memory (RAM) ถ้ามีไฟเลี้ยง

อยู่ข้อมูลที่เก็บไว้จะไม่สูญหาย แต่ถ้าปิดเครื่องหรือไม่จ่ายไฟให้แก่ RAMแล้ว ข้อมูลใน RAM ก็จะสูญหายไป การสูญหายของข้อมูลไม่ได้ความหมายว่าไม่ได้อะไรอยู่เลยแต่เป็นการที่มีข้อมูลใหม่ซึ่งไม่ใช่ข้อมูลที่เก็บไว้เดิมเข้ามาอยู่แทนที่เช่นเดิมเก็บข้อมูล 18Hไว้ที่ตำแหน่ง 1900H เมื่อปิดไฟแล้วเปิดใหม่ ข้อมูลที่ตำแหน่ง 1900H จะ ไม่ใช่ 18H อาจเป็นค่าอะไรก็ได้ ซึ่งเรียกการเกิดลักษณะแบบนี้ว่าข้อมูลสูญหายไป หน่วยความจำแบบ

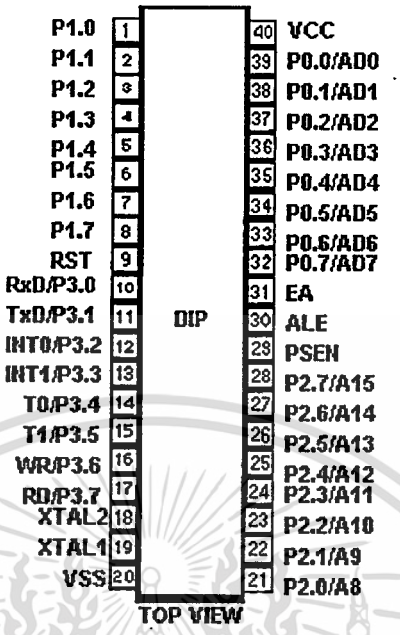
1) Data Memoryของ 8051 จะมีอยู่ 2 ชุด ชุดหนึ่งอยู่ภายใน 8051 จำนวน 128 ไบท์ที่ตำแหน่ง 00H ถึง 7FH (เบอร์ 8052 จะมี 256 ไบท์อยู่ที่ตำแหน่ง 00H ถึง FFH) และอีกชุดหนึ่งจะต้องต่ออยู่ภายนอกของวงจรรวม 8051 มีได้สูงสุด 65536 ไบท์ (64 กิโลไบท์) อยู่ที่ตำแหน่ง 0000H ถึง FFFFH ดังแสดงในรูปที่ 2.3 หน่วยความจำแบบ Data memory ภายใน 8051 ที่ตำแหน่ง 80H ถึง FFH นั้นไม่ได้มีอยู่ทุกตำแหน่ง จะมีเฉพาะในบางตำแหน่งซึ่งเรียกหน่วยความจำบางตำแหน่งนี้ว่าSpecial unction (SFR) เพราะจะใช้หน่วยความจำเหล่านี้ สำหรับงานพิเศษเท่านั้น แต่ละตำแหน่งของหน่วยความจำแบบ SFR นี้ อาจเป็น RAM หรือวงจรนับ (Counter) วงจรตั้งเวลา(Timer) ก็ได้เช่นเป็น Timer 0, Timer 1 ดังนั้นใน 8051 จึงไม่ถือว่า SFR เป็น Data Memory ถ้าเป็น 8052 ซึ่งมี Data Memory ขนาด 256 ไบท์ จะใช้บางตำแหน่งของหน่วยความจำช่วงตำแหน่ง 80H ถึง FFH เป็น SFR ส่วนตำแหน่งอื่นที่เหลือก็เป็น RAM เหมือนกับหน่วยความจำช่วง 00H ถึง 7FH นั่นเอง

2.1.3 สถาปัตยกรรมของ 8051

ในหัวข้อที่ 2.1.2 ได้กล่าวถึงไดอะแกรมภายในของ 8051 อย่างกว้าง ๆ ซึ่งพอจะบอกได้โดยสังเขปว่าประกอบด้วยส่วนใหญ่ ๆ อะไรบ้าง ในรูปที่ 2.4 เป็นสถาปัตยกรรมภายในของ 8051ซึ่งจะอธิบายถึงส่วนย่อยๆของภายใน 8051 เพียง ชิพเดียว และสัญญาณจากภายในจะต่อออกสู่ภายนอกทางขา (Pin) ของ 8051 ที่มีอยู่ 40 ขา ดังรูปที่ 2.5



รูปที่ 2.4 สถาปัตยกรรมภายในของ 8051



รูปที่ 2.5 ไดอะแกรมขาของ 8051 แบบ DIP

8051 ไมโครคอนโทรลเลอร์ที่บรรจุอยู่ในวงจรรวมแบบ Dual Inline Package (DIP) ซึ่งของ 8051 มีขาอยู่ข้างละ 20 ขารวมทั้งหมด 40 ขานั้นจะใช้งานต่าง ๆ กัน ดังนี้คือ

1) Vcc

ขา 40 เป็นขาที่ต้องป้อนไฟเลี้ยง+5 V เข้าไปเพื่อให้วงจรรวมทำงานได้ ระดับโวลเตจของลอจิก 0 และ 1 ของ 8051 จึงต่อเข้ากับอุปกรณ์ลอจิกแบบ TTL ได้โดยตรง

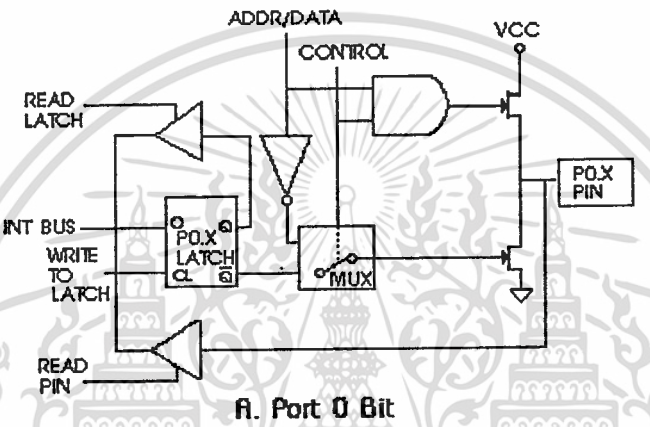
2) Vss

ขา 20 เป็นขาที่ต้องต่อกับกราวด์ (Ground) ของแหล่งจ่ายไฟ การต่ออุปกรณ์ทั้งหมดจะต้องมีกราวด์ขดงอุปกรณ์ต่อเข้าด้วยกัน

3) Port 0

เป็นพอร์ทขนานขนาด 8 บิท อยู่ที่ขา 39 ถึง 32 เริ่มจากบิท 0 ถึงบิท 7 ตามลำดับดังในรูปที่ 6 แต่ละขาจะเขียนว่า P0,0,P0.0....,P0.7 นั้น PO.7 หมายถึงบิท 7 ของ

พอร์ท 0ซึ่งเป็นบิทที่มีนัยสำคัญสูงสุด (Most Signifcant) และ P0.0 คือบิท 0 ของพอร์ท 0 เป็นบิทที่มีสำคัญต่ำสุด (Least Significant) พอร์ท 0 นี้ใช้ได้ทั้งการรับ-ส่งตำแหน่ง และข้อมูลกับหน่วยความจำหรือใช้เป็นพอร์ทรับ-ส่งข้อมูลก็ได้ ข้อมูลที่ส่งออกทางพอร์ท 0 จะถูก Latch ไว้ที่ขาของพอร์ท โครงสร้างแต่ละบิทของพอร์ท 0 เป็น แบบ Oper Drain Bidirectional ดังรูป 2.6



รูปที่ 2.6 โครงสร้างของพอร์ท 0

ในรูปที่ 2.6 เมื่อเปรียบเทียบกับรูปที่ 2.4 ส่วนที่ 1 ของรูป 2.6 ก็คือ Port 0 Latch ในรูปที่ 2.4 และส่วนที่ 2 ของรูป 2.6 ก็คือ Port 0 Driver ของรูปที่ 2.4 นั่นเอง

จากโครงสร้างในรูปที่ 2.6 เมื่อมีคำสั่งการเขียนข้อมูลมายังพอร์ท 0 ข้อมูลจาก Internal Data Bus จะถูก Latch ไว้ที่ D-FF โดยสัญญาณ "Write to Latch " ที่ถูกสร้างมาจากส่วน Timing and Control และในการอ่านข้อมูลจากพอร์ท 0 จะอ่านได้ 2 แบบคือ การอ่านข้อมูลที่ส่งไปเก็บไว้ที่พอร์ทก็จะมีสัญญาณ Read Latch มาเพื่ออ่านข้อมูลจาก D-FF กลับเข้าไปยัง Internal Data Bus การอ่านข้อมูลอีกแบบก็คือการอ่านสภาวะของสัญญาณที่เข้ามาทางพอร์ท 0 ก็จะมีสัญญาณ Read Pin มาควบคุมการอ่านพอร์ท 0 จะใช้งานหลายอย่างดังนี้

ก) ใช้สำหรับส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อด้วย ตำแหน่งหน่วยความสูงสุดที่จะติดต่อได้ก็คือ 64Kbytes จึงมีค่าตำแหน่งหน่วยความจำ

16 บิทของของเลขฐาน 2 ค่าตำแหน่งหน่วยความจำ 8 บิทล่างจะถูกส่งออกไปทางพอร์ท 0 และ 8 บิทบนจะถูกส่งออกไปทางพอร์ท 2

ข) ใช้รับ-ส่งข้อมูลกับ Data Memory หรือใช้รับข้อมูลจาก Program Memory

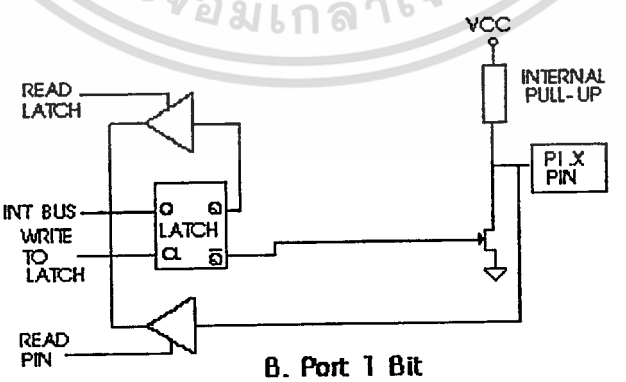
ค) ใช้รับ-ส่งข้อมูลผ่านทางพอร์ทโดยตรง ในกรณีที่ไม่มีการใช้หน่วยความจำของ Program Memory หรือ Data Memory ภายนอก

วงจรภายในส่วน Timing and Control จะเป็นตัวสร้างสัญญาณควบคุมวงจรในรูปที่ 2.6 เพื่อให้การทำงานแต่ละอย่างข้างต้น เมื่อแต่ละบิทของพอร์ท 0 ทำงานตามข้อ 1 และ 2 ข้างต้น วงจร Timing and Control จะทำให้สภาวะลอจิกของขา Control เป็น 1 ซึ่งทำให้สวิทซ์ MUX อยู่ในตำแหน่งข้างบน เมื่อพอร์ท 0 จะส่งข้อมูลซึ่งเป็นค่าตำแหน่งหน่วยความจำหรือข้อมูลที่จะเขียนไปยังหน่วยความจำภายนอกก็จะส่งค่าดังกล่าวมายัง ADDR/DATA ถ้าข้อมูลที่ส่งมาเป็น 1 จะทำให้สัญญาณออกจาก AND GATE เป็น 1 และสัญญาณที่ออกจาก Inverter เป็น 0 ดังนั้น FET ตัวบน ON (สภาวะ ON ของFET คือความต้านทานระหว่างขา D กับ S มีค่าต่ำมากเสมือนกับวงจรปิด) ส่วน FET ตัวล่าง OFF (สภาวะ OFF ของ FET คือความต้านทานระหว่างขา D กับ S มีค่าสูงมากเสมือนวงจรเปิด) สภาวะลอจิกที่ขา P0.X Pin จะเป็น 1 แต่ถ้าข้อมูลที่ส่งออกมายัง ADDR/DATA เป็น 0 ก็จะทำให้สัญญาณจาก AND GATE เป็น 0 และสัญญาณที่ส่งออกจาก Inverter เป็น 1 ดังนั้น FET ตัวบนจะ OFFส่วน FET ตัวล่างจะ ON ทำให้สภาวะลอจิกที่ขา P0.X Pin เป็น 0 เมื่อ 8051 ต้องการใช้พอร์ท 0สำหรับการอ่านข้อมูล จากหน่วยความจำภายนอก หรือใช้ทำงานในข้อ 3 ข้างบน ก็จะทำได้โดยวงจร Timing and Control ทำให้สภาวะลอจิกของสัญญาณ Control ในรูปเป็น 0 ทำให้เอาท์พุทจาก ANDGATE เป็น 0 FET ตัวบนจะ OFF และสวิทซ์ MUX จะอยู่ในตำแหน่งข้างล่าง ดังนั้น FET ตัวล่างจะ ON หรือ OFF ก็แล้วแต่ข้อมูลที่ขา Q ของ D-FF เมื่อมีการเขียนข้อมูลจาก Internal Data Bus มายัง D-FF ก็จะมีสัญญาณ Write to Latch มายัง D-FF ด้วย ถ้าข้อมูลที่เขียนมาเป็น 1 ก็จะทำให้ขา Q มีสภาวะลอจิกเป็น 0 ทำให้ FET ตัวล่าง OFF ดังนั้นขา P0.X อยู่ในสภาวะอิมพีแดนซ์สูง (High impedance) เพราะ FET ทั้ง 2 ตัว OFF แต่ถ้าข้อมูลที่เขียนมายัง D-FF เป็น 0 จะทำให้ FET ตัวล่าง ON แต่ตัวบน OFF ทำให้

สภาวะลอจิกที่ขา P0.X เป็น 1 ดังนั้น PORT 0 เมื่อให้ทำงานเป็นพอร์ทส่งข้อมูล (ไม่ใช่ส่งตำแหน่งหน่วยความจำ) จะไม่สามารถแสดงสภาวะลอจิก 1 ได้จึงต้องต่อตัวต้านทาน Pull Up ไว้ภายนอกระหว่างขา P0.X กับไฟเลี้ยงวงจรถ้าจะใช้พอร์ท 0 สำหรับรับข้อมูลเข้าจะต้องเขียน 1 มาเก็บไว้ยัง D-FF เสียก่อนเพื่อให้ขา P0.X อยู่ในสภาวะ High Impedance แล้วจึงใช้คำสั่งอ่านสภาวะลอจิกเข้าไปยัง Internal Data Bus ต่อไป โดยคำสั่งอ่านสภาวะลอจิกทางพอร์ท 0 ก็จะทำให้วงจร Timing and Control สร้างสัญญาณ Read Pin สำหรับการอ่านสภาวะลอจิกข้างต้น ถ้าไม่เขียน 1 มาเก็บไว้ยัง D-FF ก่อนที่จะอ่านข้อมูลแล้วอาจมีข้อมูลค้างอยู่ที่ D-FF ทำให้ Q เป็น 0 และ Q เป็น 1 ซึ่งทำให้ FET ตัวล่าง ON สัญญาณที่ต่อเข้าที่ขา P0.X ไม่ว่าจะมีสภาวะลอจิกใด จะถูกดึงลงกราวด์ ดังนั้นเมื่ออ่านข้อมูลเข้าไปก็จะพบว่าเป็น 0 เสมอ ในการอ่านข้อมูลจกหน่วยความจำภายนอกนั้นวงจร Timing and Control ก็จะเขียนข้อมูลมายัง D-FF ให้เป็น 1 และสร้างสัญญาณ Control ให้มีลอจิกเป็น 0 ก่อนจะอ่านข้อมูลเข้าไปด้วย

4) PORT 1

เป็นพอร์ทขนาน 8 บิทในรูปที่ 2.5 คือขา P1.0-P1.7 (ขา1-8) P1.0 หมายถึง บิท 0 ของพอร์ท 1 ซึ่งเป็นบิท Least Significant Bit และบิท P1.7 หมายถึง บิทที่ 7 ของพอร์ท 1 ซึ่งเป็นบิท Most Significant Bit โครงสร้างของพอร์ท 1 แต่ละบิทมีดังรูปที่ 2.7

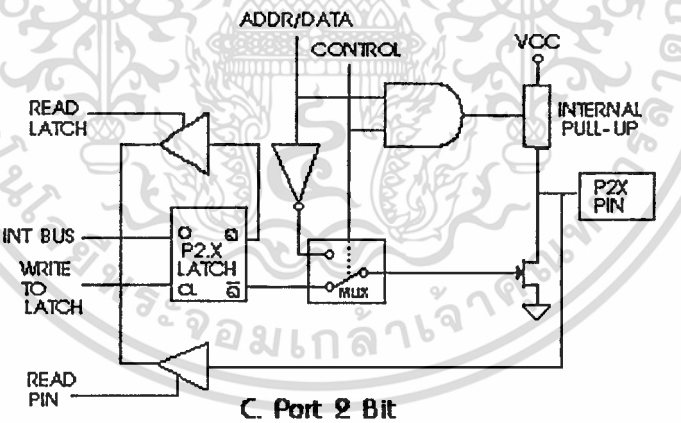


รูปที่ 2.7 โครงสร้างของ PORT 1

ส่วนที่ 1 คือ Port 1 Latch ในรูปที่ 2.4 ซึ่งก็จะมีการทำงานเหมือนส่วนที่ 1 ของพอร์ท 0 ในรูปที่ 2.6 ส่วนที่ 2 คือ Port 1 Driver ในรูปที่ 2.4 Port 1 Driver นี้จะมีตัวต้านทานต่ออยู่เป็น Internal Pull Up พอร์ท 1 นี้จะใช้ทำหน้าที่เป็นตัวรับ-ส่งข้อมูลเท่านั้น ข้อมูลที่ส่งออกมาทางพอร์ท 1 จะถูก Latch ไว้แล้วส่งออกไปทางแต่ละขาก่อนที่อ่านข้อมูลเข้าไปทางพอร์ทจะเขียน 1 ไปยังทุกบิทของพอร์ท 1 เสียก่อนเพื่อ FET อยู่ในสภาวะ OFF ก่อน มิฉะนั้นแล้วถ้ามีข้อมูล 0 ส่งออกมาค้างอยู่ที่ D-FF จะทำให้ FET อยู่ในสภาวะ ON ดังนั้นถ้าสัญญาณส่งเข้ามาที่ขานี้ก็จะถูกลักวงจรลงกราวด์โดยไม่สนใจว่าสภาวะลอจิกของสัญญาณที่เข้ามาจะเป็นอะไรข้อมูลที่เข้าไปเป็น 0 เสมอ

5) PORT 2

เป็นพอร์ทขนาน 8 บิทคือ ขา P2.0-P2.7 (บิท0-บิท7 ของพอร์ท 2) ในรูปที่ 2.5 โครงสร้างของพอร์ 2 แต่ละบิทจะมีดังรูปที่ 2.8



รูปที่ 2.8 โครงสร้างของ Port 2

ลักษณะโครงสร้างจะเหมือนกับ Port 0 แตกต่างกันที่ใน Port 2 นั้นภาค Driver จะใช้งานเพียง 2 ลักษณะ คือ

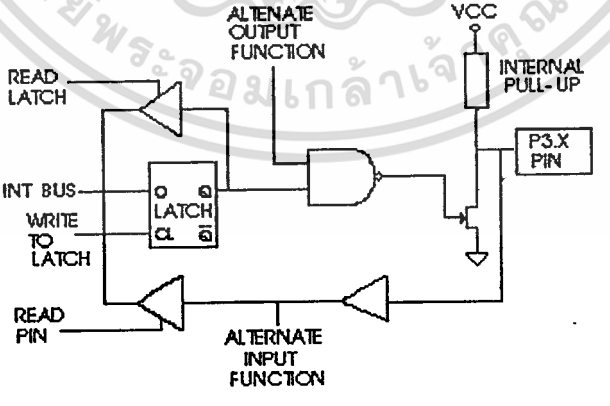
ก) ใช้ส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อ ค่าตำแหน่งนี้เป็น 8 บิทบนของค่าตำแหน่ง

ข) ใช้เป็นพอร์ทรับและส่งข้อมูลกับภายนอก

ดังนั้นภาค Driver ของพอร์ท 2 จึงแตกต่างจาก Driver ของพอร์ท 0 โดยที่ในพอร์ท 2 นั้นจะมีเฉพาะ ADDR (ตำแหน่งหน่วยความจำ) เข้ามาที่ MUX (Multiplexer) เท่านั้น นอกนั้นแล้วการทำงานจะเหมือนกันและที่เอาท์พุทของพอร์ท 2 จะมี Internal Pull Up ซึ่งเป็นตัวต้านทานและทำให้เอาท์พุทของพอร์ท2 แสดงสถานะลอจิกเป็น 1 ได้ถ้า FET อยู่ในสภาวะ OFF บางครั้งเรียกว่า "Quasi-Bidirectional" เมื่อใช้เป็นพอร์ทอินพุทก็สามารถทำได้โดยการต่อสัญญาณภายนอกเข้ามาโดยตรง ถ้าสัญญาณเป็น 0 ก็จะมรกระแสไหลออกจากพอร์ท (Source Current) ในการที่จะใช้พอร์ทรับข้อมูลเข้าจะต้องเขียน 1 ไปยังแต่ละบิทของพอร์ทเสียก่อน ดังได้อธิบายในเรื่อง Port 0 และ Port 1

6) PORT 3

คือขา P3.0-P3.7 หรือขา 10-17 ตามลำดับในรูปที่ 2.5 พอร์ทนี้มีโครงสร้างดังรูปที่ 2.9



รูปที่ 2.9 โครงสร้างของพอร์ท 3

ส่วนที่ 1 ในรูปที่ 2.9 เป็นส่วน Latch ข้อูลที่เขียนมายังพอร์ท 3 ทาง Internal Bus เหมือนกับพอร์ทอื่นๆและพอร์ท 3 จะมี Internal Pull Up อยู่ทุกบิทแต่พอร์ท 3 นี้แต่ละบิทจะใช้ในการทำงานอื่นได้โดยใช้คำสั่งควบคุมการทำงาน ในส่วนที่ 2 จะมีสัญญาณ Altinative Output Function ที่สร้างมาจากส่วน Timing and Control สัญญาณ Altinative Output Function เป็นสัญญาณที่ส่งออกในกรณีที่ใช้พอร์ท 3 ทำงานใน Function อื่นและจุด Altinative Input Function เป็นจุดที่จะเอาสัญญาณไปเข้ากับส่วนอื่นตามการทำงานของบิทนั้น แต่ละบิทของพอร์ท 3 จะมีFunction อื่นดังนี้

ก) P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม

ข) P3.1/TXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม

ค) P3.2/INT0 (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

ง) P3.3/INT1 (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

จ) P3.4/T0 (Timer Counter 0 External Input) ขารับสัญญาณเข้าไปยังวงจร Timer Counter 0 ที่ทำหน้าที่นับจำนวน Cycle ของสัญญาณ T0 นี้หรือสัญญาณนาฬิกาก็ได้

ฉ) P3.5/T1 (Timer Counter 1 External Input) ขารับสัญญาณเข้าไปยังวงจร Timer Counter 1 ซึ่งมีการทำงานเหมือนกับ T0

ช) P3.6/WR (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051

ซ) P3.7/RD (External Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก

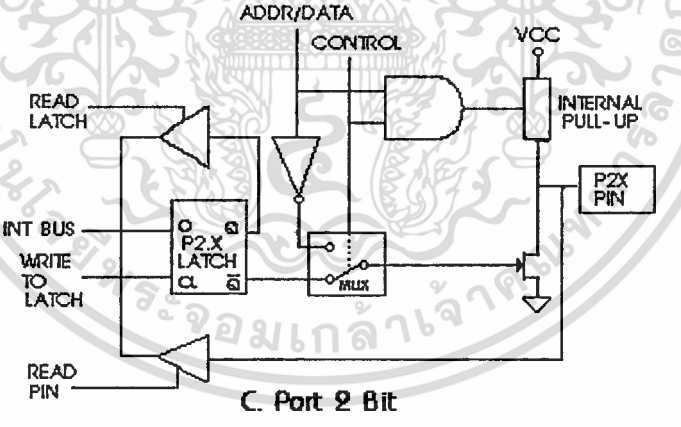
7) RST

ขา Reset ขานี้จะใช้ทำการ Reset การทำงานของ 8051 ที่ขา RST ภายใน 8051 จะมีตัวต้านทานต่อระหว่างขานี้กับกราวด์ (Ground) ถ้าป้อนสัญญาณที่มีสภาวะลอจิก 1 เข้าไปที่ขานี้จะเป็นการReset การทำงานของ 8051 ดังนั้นจึงสามารถต่อตัว

ส่วนที่ 1 คือ Port 1 Latch ในรูปที่ 2.4 ซึ่งก็จะมีการทำงานเหมือนส่วนที่ 1 ของพอร์ท 0 ในรูปที่ 2.6 ส่วนที่ 2 คือ Port 1 Driver ในรูปที่ 2.4 Port 1 Driver นี้จะมีตัวต้านทานต่ออยู่เป็น Internal Pull Up พอร์ท 1 นี้จะใช้ทำหน้าที่เป็นตัวรับ-ส่งข้อมูลเท่านั้น ข้อมูลที่ส่งออกมาทางพอร์ท 1 จะถูก Latch ไว้แล้วส่งออกไปทางแต่ละขาก่อนที่อ่านข้อมูลเข้าไปทางพอร์ทจะเขียน 1 ไปยังทุกบิทของพอร์ท 1 เสียก่อนเพื่อ FET อยู่ในสภาวะ OFF ก่อน มิฉะนั้นแล้วถ้ามีข้อมูล 0 ส่งออกมาค้างอยู่ที่ D-FF จะทำให้ FET อยู่ในสภาวะ ON ดังนั้นถ้าสัญญาณส่งเข้ามาที่ขานี้ก็จะถูกลักวงจรลงกราวด์โดยไม่สนใจว่าสภาวะลอจิกของสัญญาณที่เข้ามาจะเป็นอะไรข้อมูลที่เข้าไปเป็น 0 เสมอ

5) PORT 2

เป็นพอร์ทขนาน 8 บิทคือ ขา P2.0-P2.7 (บิท0-บิท7 ของพอร์ท 2) ในรูปที่ 2.5 โครงสร้างของพอร์ 2 แต่ละบิทจะมีดังรูปที่ 2.8



รูปที่ 2.8 โครงสร้างของ Port 2

ลักษณะโครงสร้างจะเหมือนกับ Port 0 แตกต่างกันที่ใน Port 2 นั้นภาค Driver จะใช้งานเพียง 2 ลักษณะ คือ

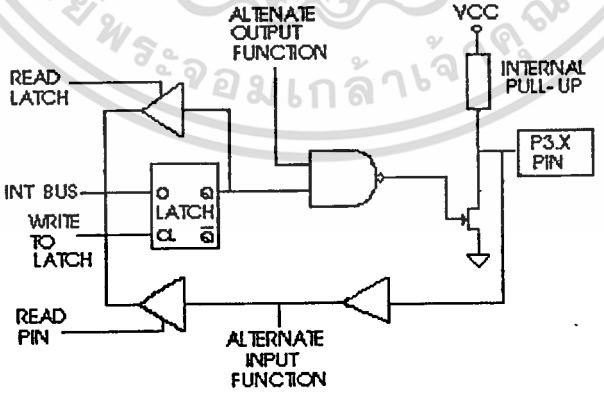
ก) ใช้ส่งค่าตำแหน่งหน่วยความจำภายนอกที่ต้องการติดต่อ ค่าตำแหน่งนี้เป็น 8 บิทบนของค่าตำแหน่ง

ข) ใช้เป็นพอร์ทรับและส่งข้อมูลกับภายนอก

ดังนั้นภาค Driver ของพอร์ท 2 จึงแตกต่างจาก Driver ของพอร์ท 0 โดยที่ในพอร์ท 2 นั้นจะมีเฉพาะ ADDR (ตำแหน่งหน่วยความจำ) เข้ามาที่ MUX (Multiplexer) เท่านั้น นอกนั้นแล้วการทำงานจะเหมือนกันและที่เอาท์พุทของพอร์ท 2 จะมี Internal Pull Up ซึ่งเป็นตัวต้านทานและทำให้เอาท์พุทของพอร์ท2 แสดงสถานะลอจิกเป็น 1 ได้ถ้า FET อยู่ในสภาวะ OFF บางครั้งเรียกว่า "Quasi-Bidirectional" เมื่อใช้เป็นพอร์ทอินพุทก็สามารถทำได้โดยการต่อสัญญาณภายนอกเข้ามาโดยตรง ถ้าสัญญาณเป็น 0 ก็จะมรกระแสไหลออกจากพอร์ท (Source Current) ในการที่จะใช้พอร์ทรับข้อมูลเข้าจะต้องเขียน 1 ไปยังแต่ละบิทของพอร์ทเสียก่อน ดังได้อธิบายในเรื่อง Port 0 และ Port 1

6) PORT 3

คือขา P3.0-P3.7 หรือขา 10-17 ตามลำดับในรูปที่ 2.5 พอร์ทนี้มีโครงสร้างดังรูปที่ 2.9



รูปที่ 2.9 โครงสร้างของพอร์ท 3

ส่วนที่ 1 ในรูปที่ 2.9 เป็นส่วน Latch ข้อูลที่เขียนมายังพอร์ท 3 ทาง Internal Bus เหมือนกับพอร์ทอื่นๆและพอร์ท 3 จะมี Internal Pull Up อยู่ทุกบิทแต่พอร์ท 3 นี้แต่ละบิทจะใช้ในการทำงานอื่นได้โดยใช้คำสั่งควบคุมการทำงาน ในส่วนที่ 2 จะมีสัญญาณ Altinative Output Function ที่สร้างมาจากส่วน Timing and Control สัญญาณ Altinative Output Function เป็นสัญญาณที่ส่งออกในกรณีที่ใช้พอร์ท 3 ทำงานใน Function อื่นและจุด Altinative Input Function เป็นจุดที่จะเอาสัญญาณไปเข้ากับส่วนอื่นตามการทำงานของบิทนั้น แต่ละบิทของพอร์ท 3 จะมีFunction อื่นดังนี้

ก) P3.0/RXD (Serial Input Port) เป็นขาที่ใช้รับข้อมูลแบบอนุกรม

ข) P3.1/TXD (Serial Output Port) เป็นขาที่ใช้ส่งข้อมูลแบบอนุกรม

ค) P3.2/INT0 (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

ง) P3.3/INT1 (External Interrupt) ใช้รับสัญญาณขัดจังหวะจากภายนอก

จ) P3.4/T0 (Timer Counter 0 External Input) ขารับสัญญาณเข้าไปยังวงจร Timer Counter 0 ที่ทำหน้าที่นับจำนวน Cycle ของสัญญาณ T0 นี้หรือสัญญาณนาฬิกาก็ได้

ฉ) P3.5/T1 (Timer Counter 1 External Input) ขารับสัญญาณเข้าไปยังวงจร Timer Counter 1 ซึ่งมีการทำงานเหมือนกับ T0

ช) P3.6/WR (External Data Memory Write Strobe) ขาสัญญาณควบคุมการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051

ซ) P3.7/RD (External Data Memory Read Strobe) ขาสัญญาณควบคุมการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก

7) RST

ขา Reset ขานี้จะใช้ทำการ Reset การทำงานของ 8051 ที่ขา RST ภายใน 8051 จะมีตัวต้านทานต่อระหว่างขานี้กับกราวด์ (Ground) ถ้าป้อนสัญญาณที่มีสภาวะลอจิก 1 เข้าไปที่ขานี้จะเป็นการReset การทำงานของ 8051 ดังนั้นจึงสามารถต่อตัว

เก็บประจุ (Capacitor) ภายนอกระหว่างขา RST กับไฟเลี้ยง +5 Volts เพื่อให้เกิดการ Reset เมื่อเริ่มป้อนไฟเลี้ยงให้กับ 8051 ซึ่งเรียกว่า Power On Reset การ Reset จะทำให้ค่ารีจิสเตอร์ต่างๆ เปลี่ยนไปเป็นค่า 1 ดังในตารางที่ 2.2

ตารางที่ 2.2 ค่าของรีจิสเตอร์เมื่อเกิดการ Reset 8051

REGISTER	CONTENT
PC	0000H
ACC	00H
B	00H
PSW	00H
SP	00H
DPTR	0000H
P0-P3	0FFH
IP	00H
IE	0X000000B
TMOD	00H
TCON	00H
T2CON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
TH2	00H
TL2	00H
RCAP2H	00H
RCAP2L	00H

REGISTER	CONTENT
SCON	00H
SBUF	Indeterminate
IOCON	0 0 H

ในตารางที่ 2.2 ช่องทางขวาเป็นค่าของรีจิสเตอร์ที่อยู่ทางซ้ายเมื่อสิ้นสุดการ Reset ในรีจิสเตอร์ SBUF เมื่อสิ้นสุดการ Reset จะมีค่าที่ไม่แน่นอนและพอร์ทจะอยู่ในสภาวะลอจิก 1 ทุกบิทตลอดเวลาที่สัญญาณของขา RST เป็น HIGH อยู่

เมื่อสัญญาณที่ขา RST กลับเป็น 0 ก็จะออกจาก Reset 8051 จะเริ่มทำงานจากคำสั่งที่อยู่ใน Program Menory ตำแหน่ง 0000H เพราะค่าของรีจิสเตอร์ PC (Program Counter) ซึ่งใช้ชี้ตำแหน่งโปรแกรมที่จะทำงานถูกเปลี่ยนให้เป็น 0000H ดังนั้นผู้ใช้จะต้องเขียนโปรแกรมมาเก็บไว้ที่ 0000Hในเครื่องไมโครคอมพิวเตอร์แบบบอร์ดเดี่ยว (Sigle Board Microcomputer) จะมีโปรแกรมที่เขียนเก็บไว้จะเริ่มจากตำแหน่ง 0000H นี้เรียกว่า Monitor Program ที่จะคอยรับการกดแป้นพิมพ์ (Keyboard) และแสดงผลทางตัวแสดงผล (Display แบบ 7 Segment)

8) ALE

Address Latch Enable ขานี้จะส่งสัญญาณที่มีความถี่ 1/6 เท่าของสัญญาณนาฬิกาจาก. ออสซิเลเตอร์ สัญญาณนี้จะส่งออกมาตลอดเวลา ยกเว้นบางครั้งของการติดต่อกับหน่วยความจำภายนอก 8051ว่าขณะนี้สัญญาณนี้ Active (เป็นลอจิก 1) จะมีการส่งข้อมูลที่เป็น 8 บิทล่างของตำแหน่งหน่วยความจำภายนอก 8051 ที่ต้องการติดต่อกับทางพอร์ท 0 อุปกรณ์ภายนอกนี้จะใช้สัญญาณนี้ในการ Latch ข้อมูลไว้เพราะพอร์ท 0 จะส่งค่าตำแหน่งหน่วยความจำออกมาเพียงชั่วขณะเท่านั้น ซึ่งในเวลาต่อมาพอร์ท 0 จะใช้รับ-ส่งข้อมูลกับหน่วยความจำภายนอก สัญญาณ ALE จะสามารถต่อเข้าอุปกรณ์TTL ชนิด LS ได้ถึง 8 Input

9) PSEN

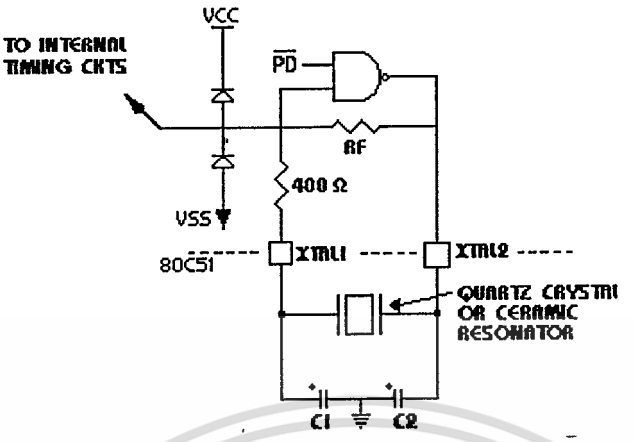
Program Store Enable เป็นขาที่ 29 ในรูปที่ 2.5 ขานี้ปกติจะให้ลอจิก 1 แต่จะส่งลอจิก 0 เมื่อต้องการอ่านคำสั่ง (Fetch Instruction) ที่จะนำไปทำงานมาจากหน่วยความจำสำหรับโปรแกรมภายนอก 8051 ในกรณีที่อ่านคำสั่งซึ่งเก็บอยู่ในหน่วยความจำสำหรับโปรแกรมภายใน 8051 แล้วสัญญาณนี้จะไม่เปลี่ยนลอจิกเป็น 0 ขา PSEN นี้สามารถต่อไปยังขาอินพุทของ TTL ชนิด LS ได้ถึง 8 อินพุท

10) EA

External Access ขา 31 ของรูปที่ 2.5 ขานี้เป็นขาอินพุทที่ต่อเข้าไปยังวงจร Timing and Control ในรูปที่ 2.4 เพื่อควบคุมการสร้างสัญญาณ PSEN ถ้าป้อนสัญญาณลอจิก 0 เข้าไปที่ขา EA นี้แสดงว่าโปรแกรมในตำแหน่ง 0000H-0FFFH ที่ต้องการถูกเก็บไว้ภายนอก 8051 จะต้องสร้างสัญญาณ PSEN ออกไปยังภายนอก เพื่อทำการ FETCH คำสั่งเข้ามาทำงาน แต่ถ้าสัญญาณที่ป้อนเข้าขา EA เป็น 1 หมายความว่าโปรแกรมในตำแหน่ง 0000H ถึง 0FFFH ถูกเก็บไว้ภายใน 8051 การทำงานตำแหน่งหน่วยความจำช่วงนี้จะอ่านคำสั่งต่าง ๆ จาก ROM ภายใน 8051

11) XTAL 1

ขาที่ 19 ของรูปที่ 2.5 ขานี้จะต่อเข้ากับขาของ Inverting Amplifier (วงจรขยายแบบป้อนกลับเฟสสัญญาณ) ที่ประกอบเป็นวงจรออสซิลเลเตอร์ ในรูปที่ 2.10 จะเห็นวงจรภายในออสซิลเลเตอร์ NAND Gate จะทำหน้าที่เป็นวงจรขยายแบบกลับเฟสของสัญญาณที่จะควบคุมให้มีการออสซิเลตหรือไม่ก็ขึ้นกับสัญญาณ PD ซึ่งจากบิท PD ของรีจิสเตอร์ PCON ถ้าต้องการใช้สัญญาณนาฬิกา (Clock Signal)จากภายนอกมาเป็นสัญญาณนาฬิกา ควบคุมการทำงานของ 8051 ก็ให้ป้อนสัญญาณเข้ามาที่จุดนี้แต่ถ้าต้องการใช้วงจรออสซิเลเตอร์ภายในก็ให้ต่อ Crytal หรือเซรามิคเรโซเนเตอร์ดังรูปที่ 2.10 คาปาซิเตอร์ในวงจรควรมีค่าประมาณ 20 pF



รูปที่ 2.10 วงจรออสซิเลเตอร์ภายใน 8051

12) XTAL 2

ขาที่ 18 ของรูปที่ 2.5 ขานี้เป็นจุดเอาท์พุทของวงจรขยายแบบกลับเฟสสัญญาณที่ประกอบเป็นวงจรออสซิเลเตอร์ (อินพุทคือขา XTAL 1) ถ้าจะใช้สัญญาณนาฬิกาที่สร้างมาจากภายนอกมาเป็นสัญญาณนาฬิกาของ 8051 แล้ว ให้ปล่อยขานี้ลอยไว้แล้วป้อนสัญญาณนาฬิกาจากภายนอกเข้ามาที่ขา XTAL 1 ดังรูปที่ 2.11

รูปที่ 2.11 8051 ที่ทำงานโดยสัญญาณที่มาจากภายนอก

2.1.4 การทำงานของ 8051

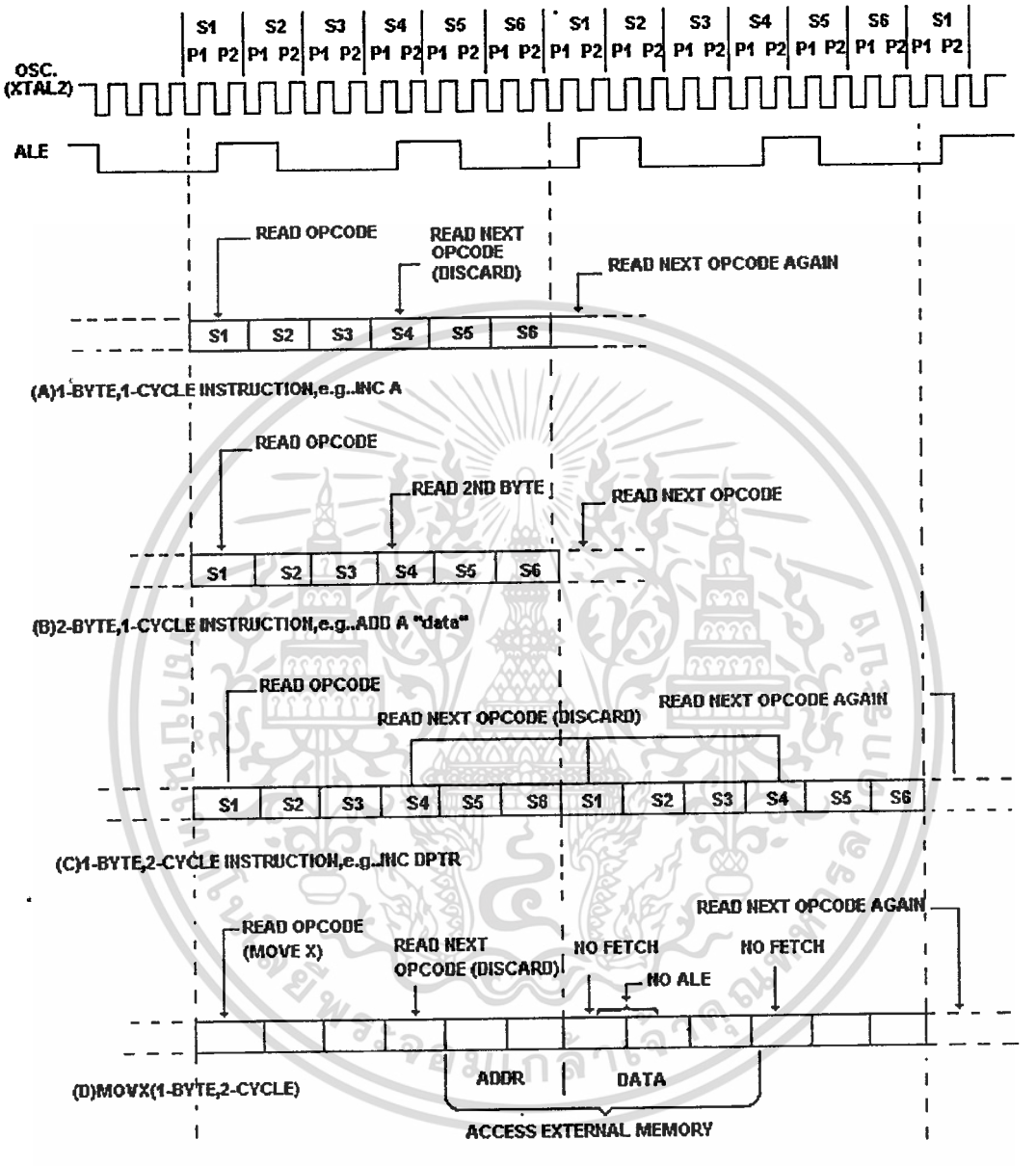
คอมพิวเตอร์จะทำงานด้วยวงจรที่เรียกว่าฮาร์ดแวร์ (Hardware) ประกอบขึ้นมาเพียงอย่างเดียวไม่ได้จะต้องมีโปรแกรมหรือคำสั่งที่จัดเรียงกันไว้ให้คอมพิวเตอร์ทำงานตามลำดับใน 8051 ก็เช่นกัน ผู้ใช้ต้องเขียนโปรแกรมเป็นภาษาเครื่อง ซึ่งอยู่ในรูปของเลข

ด้วย 1, 2 หรือ 3 ไบท์แล้วแต่ว่าจะเป็นคำสั่งให้ทำงานอะไร คอมพิวเตอร์ก็จะเหมือนกับคนที่จะต้องทำงานตามคำสั่ง เมื่อรับคำสั่งแล้วก็จะไปทำตามคำสั่งนั้นเสร็จสิ้นแล้วก็กลับมารับคำสั่งต่อไป

จากรูปที่ 2.4 เมื่อเริ่มป้อนไฟเลี้ยงให้กับ 8051 ซึ่งมีวงจร Power on reset ต่ออยู่จะมีการรีเซ็ทเกิดขึ้น การทำงานภายใน 8051 จะเริ่มจากบล็อก Program Counter ซึ่งเป็นวงจรนับ (Counter Circuit) ชนิดหนึ่งส่งค่าตำแหน่งหน่วยความจำสำหรับโปรแกรมลงไปบนบัส (Bus) หมายเลข 1 บัสนี้มีขนาด 16 บิท ค่าตำแหน่งหน่วยความจำนี้จะถูกส่งไปเก็บไว้ที่ Program ADDR Register ที่เป็นวงจร Latch ข้อมูลซึ่งเป็นค่าตำแหน่งหน่วยความจำจะปรากฏที่บนบัส 16 บิทหมายเลข 2 ถ้าเป็นค่าตำแหน่งหน่วยความจำแรกหลังจากรีเซ็ทค่าตำแหน่งหน่วยความจำจะเป็น 0000H หน่วยความจำสำหรับโปรแกรมจะเลือกได้ว่าเป็น ROM ภายในหรือภายนอก 8051 โดยการป้อนสภาวะลอจิกเข้าไปที่ 8051 ทางขา EA ซึ่งต่ออยู่กับส่วน Timing and Control ทำหน้าที่เป็นวงจรถอดรหัส (Decoder) แล้วสร้างสัญญาณควบคุมต่อไปถ้าป้อนสัญญาณลอจิก 0 เข้าไปที่ขา EA จะเป็นการเลือกใช้ ROM ภายใน 8051 โดยที่วงจร Timing and Control จะสร้างสัญญาณไปยัง ROM ภายในให้ส่งข้อมูลที่เป็นคำสั่งจากตำแหน่งที่ถูกชี้ด้วยค่าตำแหน่งที่ส่งมาบนบัสหมายเลข 2 ข้อมูลจาก ROMจะถูกส่งไปยังบัสหมายเลข 3 ที่เรียกว่า Internal Data Bus แล้วนำไปเก็บไว้ที่ Instruction Register (เป็นวงจร Latch) เพื่อส่งต่อไปให้กับวงจร Timing and Control ทำการถอดรหัสแล้วควบคุมการทำงานส่วนอื่นๆ ต่อไปแล้วแต่ว่าเป็นคำสั่งให้ทำงานอะไร ในกรณีที่เลือก ROM ภายนอก 8051 โดยป้อนสัญญาณลอจิก 1 เข้าไปที่ขา EA จะทำให้วงจร Timing and Control ส่งสัญญาณไปยังพอร์ท 0 และพอร์ท 2 เพื่อส่งค่าตำแหน่งหน่วยความจำบนบัสหมายเลข 2 ออกไปชี้หน่วยความจำภายนอก จากนั้นจะอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาทางพอร์ท 0 ไปยัง Internal Data Bus แล้วไปเก็บที่ Instruction Register เพื่อทำงานต่อไปเหมือนกับตอนอ่านคำสั่งจาก ROM ภายใน การทำงานในช่วงส่งค่าตำแหน่งหน่วยความจำไปยังหน่วยความจำแล้วอ่านข้อมูลที่เป็นคำสั่งกลับเข้ามาเก็บไว้ในInstruction Register เรียกว่าเป็นช่วงของ Fetch (Fetch Cycle) ช่วง

ต่อไปจะเป็นช่วงของการทำงานตามคำสั่งเรียกว่า Execute Cycle เช่นถ้าเป็นคำสังให้บวกข้อมูลในรีจิสเตอร์ Accumulator กับข้อมูลจากหน่วยความจำ Data Memory ภายใน RAM ตำแหน่ง 23Hวงจร Timing and Control ก็จะส่งสัญญาณให้ Instruction Register ส่งค่าตำแหน่งหน่วยความจำ 23H ลงไปยัง Instruction Data Bus แล้วนำข้อมูลไปเก็บไว้ที่ RAM ADDR Register เพื่อใช้ชี้ตำแหน่งหน่วยความจำ RAM จากนั้น Timing and Control จะสั่งให้ RAM ส่งข้อมูลที่เก็บอยู่ในหน่วยความจำตำแหน่ง 23H ลงมายัง Internal Data Bus แล้วนำข้อมูลไปเก็บไว้ที่ TMP1 (วงจร Latch) ขณะเดียวกันวงจร Timing and Control ก็จะส่งสัญญาณไปยัง ACC ให้ส่งข้อมูลมายัง TMP2 (วงจรLatch) วงจร ALU ซึ่งโครงสร้างเป็นวงจรทำการคำนวณทางคณิตศาสตร์ (บวก, ลบ, คูณ, หาร) และยังสามารถทำงานทางลอจิก (AND, OR, NOT, XOR) จะทำการบวกเลขจาก TMP1 และ TMP2 เข้าด้วยกันผลลัพธ์ที่ได้จะส่งผ่าน Internal Data Bus กลับไปเก็บยัง ACC PSW (Program StatusWord) ซึ่งจะทำหน้าที่เก็บสถานะผลลัพธ์ของการทำงานใน ALU เช่นผลลัพธ์การบวกมีค่าเกิน 8 บิท ก็จะทำให้บิทหนึ่งใน PWS ถูก Set เป็น 1

การทำงานที่กล่าวมาข้างต้นจะขึ้นกับสัญญาณควบคุมที่สร้างมาจากวงจร Timing and Control และสัญญาณที่สร้างขึ้นนี้จะอ้างอิงกับสัญญาณนาฬิกาที่สร้างมาจากวงจร Oscillator ทำให้การทำงานต่างๆ เป็นไปตามลำดับที่ผู้ผลิตได้ออกแบบไว้ ดังรูปที่ 2.12



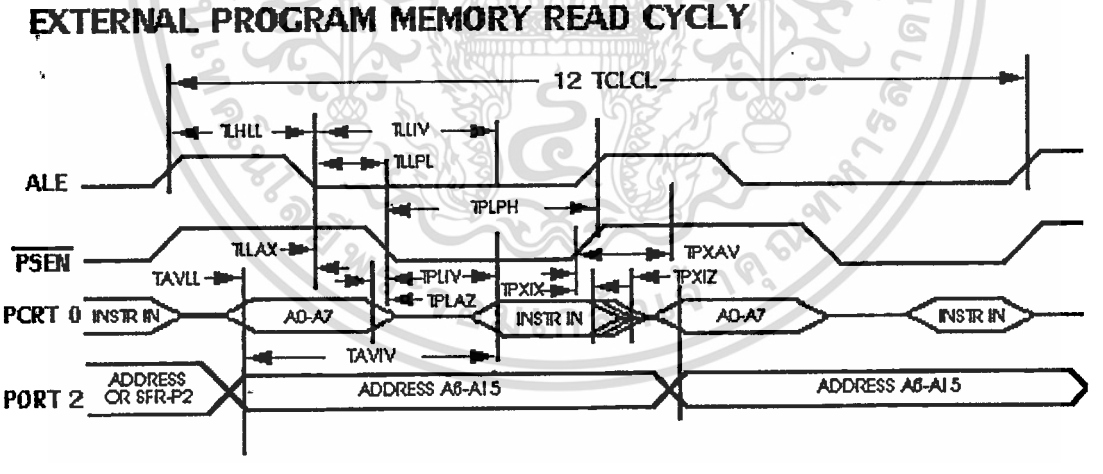
รูปที่ 2.12 ลำดับสถานะการทำงานใน MCS-51

คำสั่งแต่ละคำสั่งของ 8051 จะใช้เวลาทำงาน 1, 2 หรือ 3 ไซเคิลของเครื่อง (Machine Cycle) แล้วแต่ว่าเป็นคำสั่งประเภทใด 1 ไซเคิลของเครื่องใช้เวลา 12 ไซเคิลของสัญญาณนาฬิกา ดังนั้นแต่ละคำสั่งของ 8051 จะใช้เวลาการทำงาน 12, 24 หรือ 36 ไซเคิลของสัญญาณนาฬิกานั่นเอง แต่ละไซเคิลของเครื่องจะถูกแบ่งออกเป็น

6 State คือ S1, S2, S3, S4, S5 และ S6 แต่ละ State จะประกอบด้วย 2 ไซเคิลของสัญญาณนาฬิกา ในไซเคิลแรกจะเรียกว่าเฟส 1 (P1) และไซเคิลที่ 2 เรียกเฟส 2 (P2) ในแต่ละเฟสจะนับตั้งแต่ขอบขาลงของสัญญาณนาฬิกาถึงขอบขาลงของสัญญาณนาฬิกาที่อยู่ถัดไปดังในรูปที่ 2.12 เมื่อ 8051 ทำงานเสร็จ 1 ไซเคิลของเครื่องก็จะเริ่มทำงาน State 1 Phase 1 (S1P1) ของไซเคิลต่อไป ใน 1 ไซเคิลของเครื่องวงจร Timing and Control จะสร้างสัญญาณ ALE ออกมา 2 ไซเคิลเพื่อ Fetch คำสั่งเข้าไป 2 ครั้งเสมอ ที่บริเวณขอบขาขึ้นของสัญญาณ ALE คำสั่งใดจะมีกี่ไบท์หรือใช้เวลาทำงานกี่ไซเคิลจะดูได้จากตารางชุดคำสั่ง 8051

2.1.5 ไดอะแกรมเวลาของการติดต่อกับหน่วยความจำ

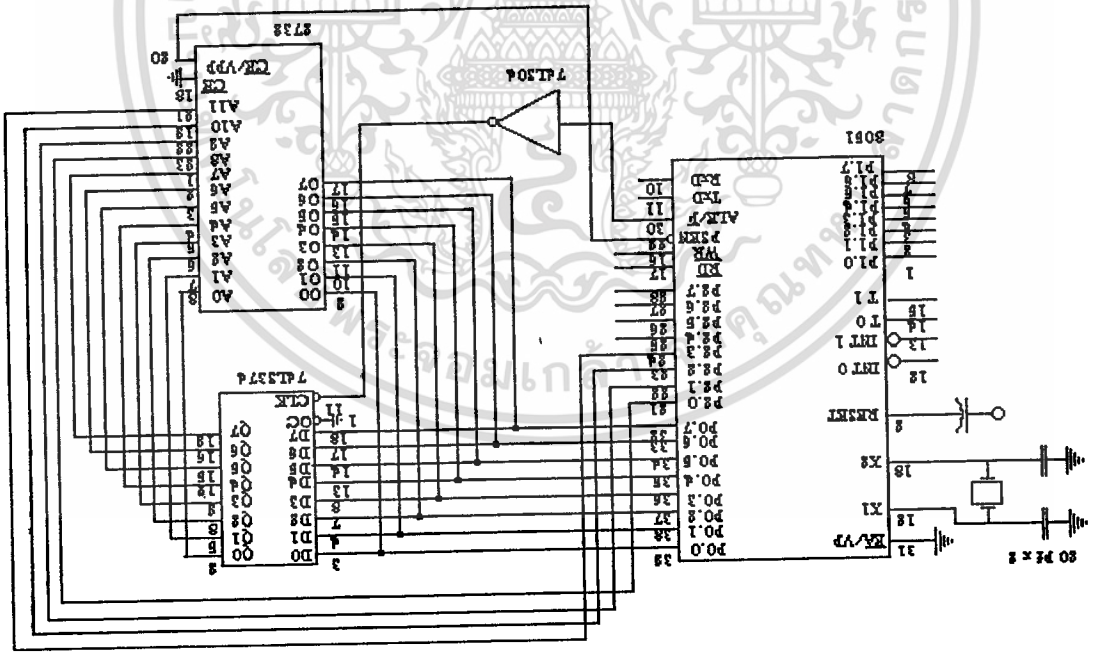
การอ่านข้อมูลจากหน่วยความจำสำหรับโปรแกรมภายนอก 8051 นั้น ลำดับสัญญาณตามเวลา(Timing Diagram) ของสัญญาณที่ทำการอ่านคำสั่งมีดังรูปที่ 2.13



รูปที่ 2.13 Timing Diagram ของการอ่านโปรแกรมจากหน่วยความจำภายนอก

การอ่านคำสั่ง (Fetch) จาก Program area ภายนอกจะเริ่มจาก 8051 ส่งสัญญาณลอจิก 1 ออกมาทางขา ALE ขณะนี้สัญญาณที่ขา PSEN จะเป็น 1 จากนั้น Port 0 จะส่งค่าตำแหน่งหน่วยความจำ 8 บิทล่างและพอร์ท 2 จะส่งตำแหน่งหน่วยความจำ 8 บิทบน

ออกมาแล้วสัญญาณ ALE จะกลับเป็น 0 อุปกรณ์ภายนอกจะสามารถใช้ขอบขาลงของสัญญาณ ALE เพื่อ Latch ตำแหน่งหน่วยความจำที่พอร์ท 0 ไว้ จากนั้นพอร์ท 0 ก็จะยกเลิกการส่งตำแหน่งหน่วยความจำเข้าสู่สภาวะ High Impedance และสัญญาณ PSEN จะเป็น 0 เพื่อเตรียมรับคำสั่งที่ส่งออกจากหน่วยความจำภายนอกเข้าไปยัง 8051 เพื่อทำงานต่อไป เมื่อคำสั่งถูกอ่านเข้าไปเก็บใน Instruction Register (ดูรูป 2.4) แล้วสัญญาณ PSEN จะกลับเป็น 1 พร้อมกับสัญญาณALE ก็จะกลับเป็น High เพื่อการอ่านคำสั่งต่อไปทำงาน ข้อมูลในพอร์ท 2 จะคงที่ตลอดเวลา ตั้งแต่สัญญาณ ALE เป็น 1 จนกระทั่งสัญญาณ ALE เปลี่ยนเป็น 0 และกลับเป็น 1 อีกครั้งหนึ่งจากนั้นจะเริ่มลำดับการ Fetch ข้อมูลไบท์ที่ 2 จากหน่วยความจำสำหรับโปรแกรม ซึ่งจะมีการเปลี่ยนแปลงของสัญญาณตามเวลาเหมือนกับการ Fetch ไบท์แรกนั่นเอง จาก Timing Diagram ดังกล่าวจะออกแบบวงจรที่มี Program Memory อยู่ภายนอก 8051 ได้ดังตัวอย่างในรูปที่ 2.14



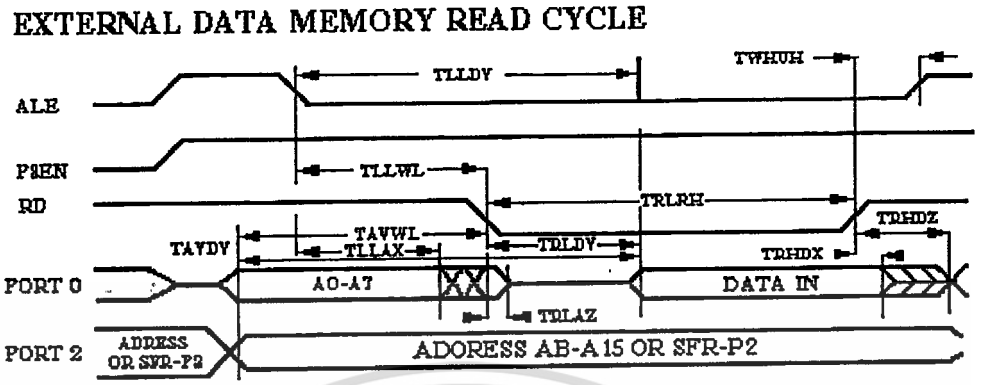
รูปที่ 2.14 วงจรที่มี Program Memory อยู่ภายนอก 8051

74LS374 ในรูปที่ 2.14 จะทำหน้าที่ Latch ตำแหน่งในหน่วยความจำ 8 บิทล่างที่เวลาขอบขาลงของสัญญาณ ALE ซึ่งสัญญาณ ALE จะถูกกลับให้เป็นตรงข้ามโดย Inverter 74LS04 ก่อนที่จะป้อนให้กับขา CK ของ 74LS374 และที่ขอบขาขึ้นของสัญญาณที่ออกจาก 74LS04 จะ Latch ตำแหน่งหน่วยความจำ ข้อมูลที่ออกจาก 74LS374 จะเป็นค่า 8 บิทล่างของตำแหน่งหน่วยความจำที่ต้องการติดต่อ ในวงจรได้ต่อค่าตำแหน่งหน่วยความจำ 8 บิทเข้ากับ A0 ถึง A7 ของ EPROM และข้อมูลจากพอร์ท 2 บิท P2.0 ถึง P2.3 จะต่อเข้ากับ A8-A11 ของ EPROM โดยตรงเพราะค่าตำแหน่งหน่วยความจำ 8 บิทบนที่ออกมาจากพอร์ท 2 จะคงที่ตลอดเวลา ขา PSEN ของ 8051 จะถูกต่อเข้ากับขา OE ของ EPROM 2716 ดังนั้นเมื่อสัญญาณ PSEN มีสภาวะลอจิกเป็น 0 ก็จะส่งคำสั่งที่เก็บใน EPROM ณ ตำแหน่งที่ชี้โดยข้อมูลที่ขา A0 ถึง A11 ออกมายังพอร์ท 0 และถูก 8051 เก็บไปทำงานต่อไป

1) การอ่าน-เขียนข้อมูลกับหน่วยความจำสำหรับข้อมูลภายนอก 8051

การอ่าน-เขียนข้อมูลกับ Data memory ภายใน 8051 นั้นจะมีสัญญาณสร้างมาจากส่วน Timing and Control โดยที่ผู้ใช้ไม่จำเป็นที่จะต้องทำความเข้าใจ แต่การอ่าน-เขียนข้อมูลกับData Memory ภายนอก อันเนื่องมาจากคำสั่ง MOVX นั้น เมื่อคำสั่งดังกล่าวถูกอ่านเข้ามายัง Instruction Register แล้ว Timing and Control จะทำการถอดรหัสแล้วสร้างสัญญาณควบคุมดังนี้

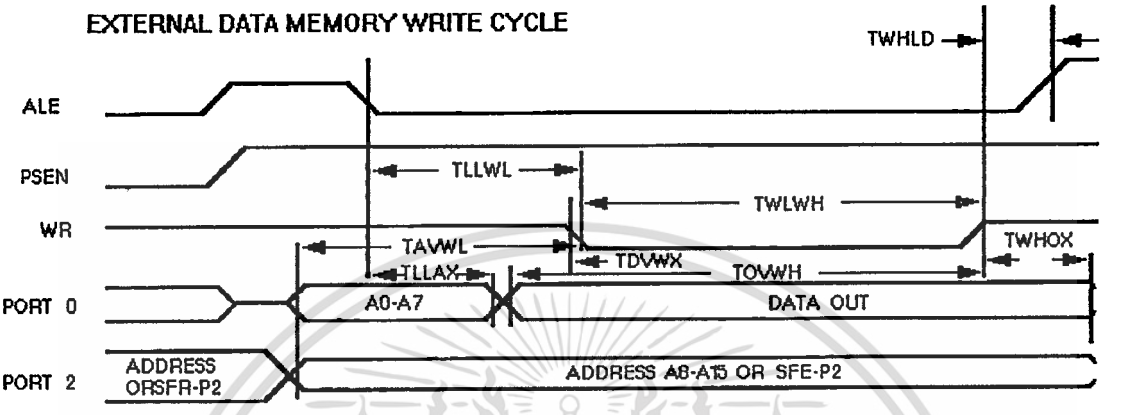
การอ่านข้อมูลจาก External Data Memory จะมีไดอะแกรมสัญญาณตามเวลาดังรูปที่ 2.15



รูปที่ 2.15 Timing diagram ของการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอก 8051

การทำงานจะเริ่มจากการส่งค่าตำแหน่งหน่วยความจำภายนอก 8 บิทล่างออกทางพอร์ท 0 และ 8 บิทบนทางพอร์ท 2 เมื่อส่งค่าตำแหน่งแล้ว สัญญาณ ALE ซึ่งเดิมมีลอจิกเป็น 1 จะกลับมาเป็น 0 เพื่อให้อุปกรณ์ภายนอกสามารถ Latch ตำแหน่งหน่วยความจำไว้เหมือนกับในการอ่านข้อมูลจากหน่วยความจำสำหรับโปรแกรมภายนอก 8051 เพื่อส่งไปยังหน่วยความจำ แม้ว่าข้อมูลบนพอร์ท 0 จะเปลี่ยนแปลงไปก็จะยังมีค่าตำแหน่งหน่วยความจำส่งไปยังหน่วยความจำในระหว่างการติดต่อกับ Data Memory นี้สัญญาณ PSEN จะเป็น 1 ตลอดเพราะสัญญาณ PSEN จะ Active (เป็น 0) ก็ต่อเมื่อเป็นการติดต่อกับหน่วยความจำสำหรับโปรแกรมภายนอก 8051 เท่านั้น 8051จะส่งสัญญาณลอจิก 0 ออกมาทางขา RD (P3.7) เพื่อบอกกับหน่วยความจำภายนอกว่าต้องการอ่านข้อมูลเข้าไปเมื่อ 8051 ส่งสัญญาณ RD เป็นลอจิก 0 จะทำให้พอร์ต 0 เข้าสู่สถาวะ high-impedance พร้อมที่จะทำให้หน่วยความจำภายนอกส่งข้อมูลมาบนพอร์ต 0 ข้อมูลบนพอร์ต 0 ซึ่งส่งมาจากหน่วยความจำภายนอกจะถูกอ่านเข้าไปเก็บที่เวลาขอบขาขึ้นของสัญญาณ RD จากนั้นสัญญาณ ALE ก็จะกลับเป็น 1 เพื่อเริ่มการทำงานในคำสั่งต่อไปในระหว่างการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอกนี้ พอร์ต 2 จะส่งค่าตำแหน่งหน่วยความจำ 8 บิทบนออกมาตลอดเวลา

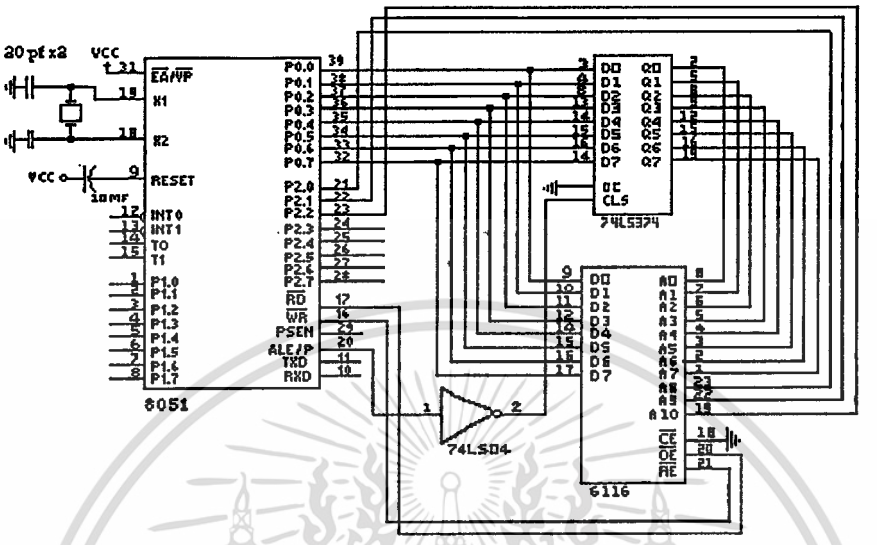
2) การเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051 จะมีไดอะแกรมสัญญาณตามเวลาดังรูปที่ 2.16



รูปที่ 2.16 Timing diagram ของการเขียนข้อมูลไปยังหน่วยความจำสำหรับข้อมูลภายนอก 8051

เมื่อ 8051 ส่งค่าตำแหน่งหน่วยความจำ 8 บิทล่างไปยังพอร์ต 0 และ 8 บิทบนลงไปทางพอร์ต 2 แล้ว สัญญาณ ALE จะกลับเป็น 0 อุปกรณ์ภายนอกจะสามารถใช้สัญญาณนี้ในการ Latch ค่าตำแหน่งหน่วยความจำบนพอร์ต 0 เหมือนกับในการอ่านข้อมูลจากหน่วยความจำสำหรับข้อมูลภายนอกเมื่อสัญญาณ ALE เป็น 0 แล้ว 8051 จะส่งข้อมูลที่ต้องการเขียนไปยังพอร์ต 0 แล้วจะให้สัญญาณ WR เปลี่ยนสภาวะเป็น 0 ขณะนี้หน่วยความจำภายนอกจะต้องเขียนข้อมูลไปเก็บยังตำแหน่งที่กำหนด จากนั้นสัญญาณ WR จะกลับเป็น 1 เพื่อเป็นการบอกสิ้นสุดการเขียนข้อมูลแล้วสัญญาณ ALE จะกลับเป็น 1 เพื่อ Fetch คำสั่งต่อไปมาทำงาน

หน่วยความจำสำหรับข้อมูลภายนอกที่สามารถเขียนและอ่านข้อมูลได้ จะสามารถเขียนเป็นวงจรได้ดังรูปที่ 2.17

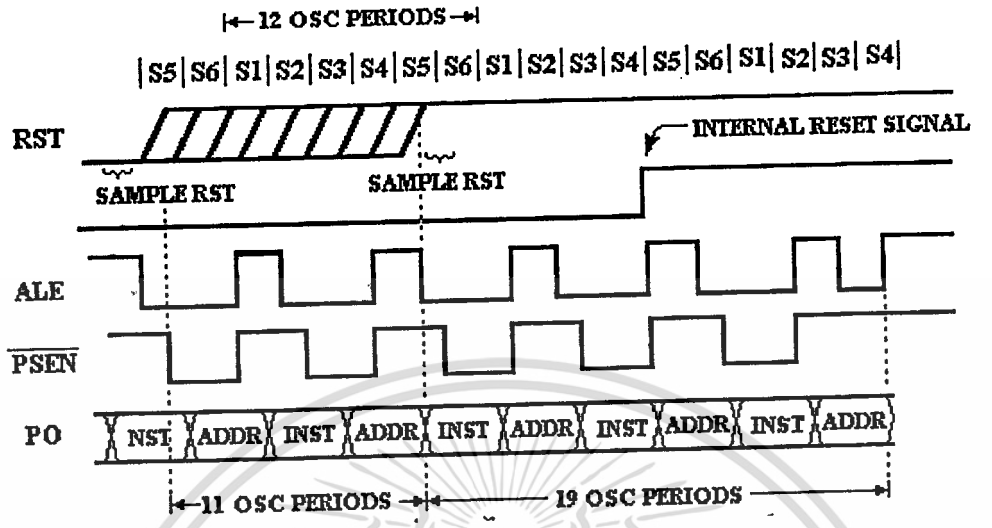


รูปที่ 2.17 วงจรที่มีหน่วยความจำสำหรับข้อมูลที่อยู่ภายนอก 8051

74LS374 ในรูปจะใช้สำหรับ Latch ค่าตำแหน่งหน่วยความจำ 8 บิทล่างไว้แม้ว่าข้อมูลบนพอร์ต 2 จะเปลี่ยนไปสัญญาณ RD และ WR จะอ่านหรือเขียนข้อมูลจากหน่วยความจำภายนอก 6116 เป็นหน่วยความจำแบบ RAM ที่สามารถจะอ่านและเขียนข้อมูลได้

2.1.6 การรีเซ็ท

เมื่อป้อนสัญญาณที่มีสภาวะลอจิก 1 เข้าไปทางขา RST จะไม่ได้เกิดการรีเซ็ทขึ้นทันทีทันใด แต่ลำดับการเกิดรีเซ็ทจะแสดงได้ดังไดอะแกรมตามรูปที่ 2.18



รูปที่ 2.18 ไดอะแกรมตามเวลาของการรีเซ็ท

ในรูปที่ 2.18 เป็น Timing Diagram ของการรีเซ็ท สภาวะลอจิกของสัญญาณที่ขา RST จะถูกอ่านเข้ามาที่เวลา S5P2 (เฟส 2 State 5) ของทุกๆ ไซเคิลของเครื่อง ในกรณีที่เป็นคำสั่งซึ่งมีการทำงานเสร็จสิ้นใน 2 ไซเคิลของเครื่องก็จะตรวจสอบเฉพาะสัญญาณที่อ่านเข้ามาในไซเคิลที่ 2 ของการทำงาน ดังนั้นในการรีเซ็ทจะต้องป้อนสัญญาณที่มีลอจิก 1 เข้าไปที่ขานี้เป็นเวลาอย่างน้อย 2 ไซเคิลของเครื่องหรือ 24 ไซเคิลของสัญญาณนาฬิกาที่สร้างจากวงจรออสซิเลเตอร์ภายใน 8051 ออสซิเลเตอร์จึงจะต้องทำงานอยู่ด้วย เมื่อ 8051 สุ่มข้อมูลที่ขา RST แล้วตรวจพบว่าเป็นสภาวะลอจิก 1 ก็จะสร้างสัญญาณรีเซ็ทขึ้นภายใน ที่เวลา S2P4 ของไซเคิลเครื่องถัดไป ข้อมูลที่แต่ละพอร์ทส่งออกมาจะยังคงปรากฏที่พอร์ทจนกว่าจะเกิดการรีเซ็ทขึ้นซึ่งต้องใช้เวลา 19 ไซเคิลนี้จะยังคงมีการ Fetch คำสั่งเข้าไปทำงานได้อยู่

สภาวะของสัญญาณลอจิกที่ขา Rst จะถูกอ่านเข้าไปที่เวลา S5P2 ของทุกๆ ไซเคิลของเครื่อง ดังนั้นถึงแม้ว่าสัญญาณที่ขา RST จะมีลอจิกเป็น 1 มาก่อนก็จะยังไม่เกิดการตรวจสอบสัญญาณรีเซ็ท ดังในรูปที่2.18 สัญญาณที่ขา RST อาจเป็น 1 มาตั้งแต่ State ที่ 6 ก็จะไม่เกิดอะไรขึ้นจนกระทั่ง 1 ไซเคิลของออสซิเลเตอร์ต่อมาซึ่งเป็นเวลา S5P2 จึงจะเกอดการตรวจสอบสัญญาณที่ขา RST ถ้าคำสั่งนั้นมีการทำงานมากกว่า 1 ไซเคิลของ

เครื่อง 8051 ก็จะต้องทำงานในคำสั่งนั้นให้เสรจสิ้นเสียก่อนจึงจะเริ่มการรีเซ็ทได้ โดย 8051 จะดูสภาวะของสัญญาณที่ขา RST ของ S5P2 ในไซเคิลของเครื่องสุดท้ายเท่านั้น ดังนั้นใน S5P2 ของไซเคิลเครื่องแรกๆ ในคำสั่งอาจมีสภาวะลอจิกที่ขา RST เป็น 1 แต่ที่ S5P2 ของไซเคิลของเครื่องสุดท้าย มีสภาวะลอจิกที่ขา RST เป็น 0 ก็จะไม่เกิดการรีเซ็ทขึ้นที่เวลา S5P2 เมื่อตรวจสอบสภาวะสัญญาณที่ขา RST แล้วพบว่าเป็น 1 จะต้องรอไปจนถึงเวลาS4P2 ของไซเคิลของเครื่องถัดไปจึงจะทำให้สัญญาณรีเซ็ทภายในเปลี่ยนสภาวะลอจิกจาก 0 เป็น 1 ในระหว่างเวลา S5P2 ที่ตรวจพบสัญญาณ RST มีลอจิกเป็น 1 จนถึง S4P2 ของไซเคิลเครื่องถัดไปจะยังคงมีการ Fetch คำสั่งเข้าไปทำงานอีก 2 คำสั่ง เมื่อสัญญาณรีเซ็ทภายในเปลี่ยนเป็น 1 ก็จะเริ่มการรีเซ็ท โดยการเขียนข้อมูล 0 ไปยัง Special Function Register ทุกตัวยกเว้นพอร์ท 0 ถึงพอร์ท 3 Stack Pointer และรีจิสเตอร์ SBUF ดังในตารางที่ 2.2 ระหว่างนี้ข้อมูลใน RAM ภายใน8051 จะไม่เปลี่ยนแปลง ข้อมูลในระหว่างการเขียนข้อมูลลงไปยัง SFR จะยังมีการ Fetch คำสั่งเข้ามาทำงานอีก 1 คำสั่งจนกว่าจะถึง S3P1 ของไซเคิลของที่ 2 (นับแต่ไซเคิลของเครื่องที่ตรวจพบลอจิก1 ที่ขา RST) ก็จะทำให้สภาวะลอจิกที่ขา ALE และ PSEN ค้างอยู่ที่สภาวะลอจิก 1 และจะเป็นอย่างนี้ไปจนกว่าสภาวะลอจิกที่ขา RST เป็น 0 เวลานับตั้งแต่พบสัญญาณลอจิก 1 ที่ขา RST ที่เวลา S5P2 จนถึงเวลาที่ ALE และ PSEN ค้างอยู่ที่ 1 จะเท่ากับ 19 ไซเคิลของออสซิลเลเตอร์เมื่อสัญญาณที่ขา RST ถูกเปลี่ยนกลับเป็นลอจิก 0 8051 จะรออีก 1 ถึง 2 ไซเคิลของเครื่องสัญญาณ ALE และ PSENจะเริ่มเปลี่ยนแปลงเพื่อเริ่มกระบวนการ Fetch คำสั่งเข้าไปทำงานเริ่มจากคำสั่งในหน่วยความจำสำหรับโปรแกรมตำแหน่ง 0000H

2.2 พอร์ทสื่อสารอนุกรม

พอร์ตสื่อสารเป็นส่วนสำคัญส่วนหนึ่งที่มีอยู่บนไมโครคอมพิวเตอร์ 16 บิท พอร์ตสื่อสารนี้มีชื่อเรียกอีกอย่างหนึ่งว่า คอม-พอร์ต (COM PORT) ผู้ออกแบบพอร์ตสื่อสารต้องการให้เป็นมาตรฐานการเชื่อมต่อแบบอนุกรมที่เรียกว่า RS232C พอร์ตนี้เป็นทางออกของข้อมูลที่ผู้ใช้สามารถส่งออกหรือรับกับระบบอื่นได้

พอร์ตสื่อสาร RS232C บนเครื่องไมโครคอมพิวเตอร์ 16 บิท มีโครงสร้างที่สามารถโปรแกรมด้วยการส่งรหัสคำสั่งให้กับชิพหลักได้ อย่างไรก็ตามผู้ออกแบบได้ให้ทางเลือกในการสื่อสารด้วย กระแสรอบ (CURRENTLOOP) หรือแบบแรงดันคือ RS232 โดยใช้จัมเปอร์เพื่อเลือกระบบ สำหรับตัว RS232C นี้เป็นมาตรฐานแบบอะซิงโครนัสที่โปรแกรม Start Bit,Stop Bit และ Parity Bit อัตราการส่งก็สามารถกำหนดได้ตั้งแต่ 50 บอด ถึง 9600 บอด ลักษณะพิเศษของวงจรคือของวงจรคือสามารถส่งสัญญาณมา INTERUPT ซีพียูตามเงื่อนไขได้ และยังมีโครงสร้างฮาร์ดแวร์ป้อนกลับเพื่อใช้ในการตรวจสอบระบบว่าทำงานปกติดีหรือไม่ ได้อีกด้วย วงจรพอร์ตสื่อสารของไมโครคอมพิวเตอร์ 16 บิทนี้ใช้ไอซีเบอร์ 8250 เป็นตัวสำคัญของระบบ 8250 เป็นไอซีขนาด 40 ขา มีขีดความสามารถพิเศษดังนั้น

- มีบัฟเฟอร์ในตัวเพื่อทำให้ไม่จำเป็นต้องซิงโครนัสการรับส่ง
- ใช้สัญญาณนาฬิกาอิสระต่างหากไม่ขึ้นกับสัญญาณนาฬิกาของระบบ
- มีสัญญาณตอบโต้ควบคุมโมเดมทั้ง CTS (clear to send), RTS (request to send), DSR (data set ready), DTR (data terminl ready), RC (ring indicator) และสัญญาณดีเทคตัวพาหะ (carrier detect)

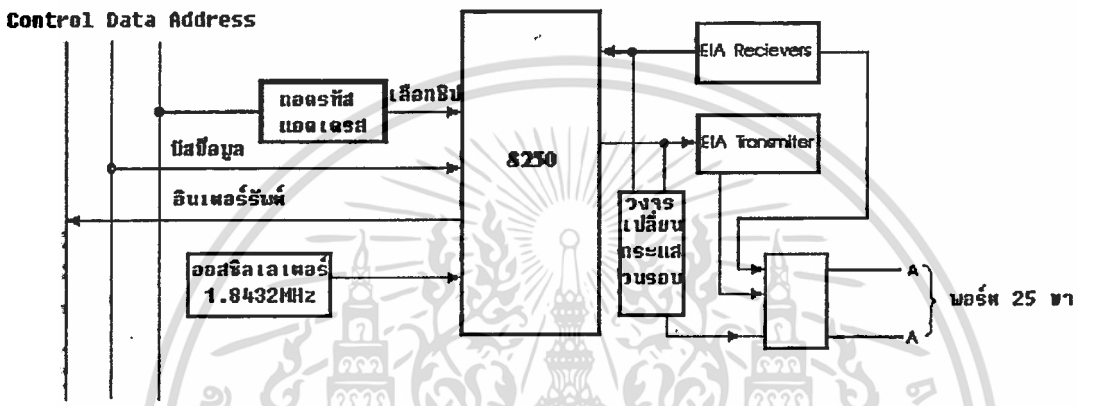
ตรวจสอบสตาร์ตบิทที่ผิดพลาด

ตรวจสอบและสร้างสัญญาณสายขาด (line break) เพื่อใช้ในการตรวจสอบการทำงานของระบบ

ชิพ 8250 มีโครงสร้างการทำงานคล้ายกับ 8251 ที่นักฮาร์ดแวร์ที่ทั่วโลกรู้จักดี การทำงานของ 8250 ต้องได้รับการโปรแกรมก่อน หลังจากนั้นจะทำงานตามรูปแบบที่โปรแกรมไว้ จนกว่าจะมีการโปรแกรมคำใหม่ อย่างไรก็ตาม 8250 มีขีดความสามารถในการทำงานที่สูงกว่า 8251 หลายอย่าง และด้วยเหตุนี้เอง 8250 จึงเป็นชิพประจำที่ใช้กับเครื่องไมโครคอมพิวเตอร์16 บิท

2.2.1 โครงสร้างของพอร์ตสื่อสาร

พอร์ตสื่อสารของเครื่องไมโครคอมพิวเตอร์ 16 บิท ที่จะกล่าวถึงนี้ เป็นระบบมาตรฐานตามแบบเครื่อง ไอบีเอมพีซีเอ็กซ์ที โครงสร้างบล็อกไดอะแกรมแสดงดังรูปที่ 2.19



รูปที่ 2.19 โครงสร้างพอร์ทสื่อสารข้อมูลที่ใช้ 8250

พิจารณาได้ว่า 8250 เป็นตัวรับข้อมูลจากบัสของระบบ ซึ่งก็คือสล็อตขนาด 31*2 (62)ขาของระบบนั้นเอง CPU ติดต่อกับ 8250 ในลักษณะพอร์ทที่เป็นอินพุทเอาต์พุท การจัดพอร์ตนี้กำหนดหมายเลขพอร์ทอย่างเจาะจง ระบบไมโครคอมพิวเตอร์ 16 บิท มีพอร์ทสื่อสาร 2 พอร์ทคือ COM_1 และ COM_2 ทั้ง COM_1 และ COM_2 มีหมายเลขอินพุทเอาต์พุทพอร์ท ดังตารางที่ 2.3

ตารางที่ 2.3 หมายเลขอินพุตเอาต์พุตพอร์ตของCOM_1 และ COM_2

อินพุตเอาต์พุตพอร์ต		เลือกรีจิสเตอร์	สถานะ DLAB
COM1	COM2		
3F8	2F8	บัฟเฟอร์ TX	DLAB = 0 (เขียน)
3F8	2F8	บัฟเฟอร์ RX	DLAB = 0 (อ่าน)
3F8	2F8	แลตช์ตัวหาร (LSB)	DLAB = 1
3F9	2F9	แลตช์ตัวหาร (MSB)	DLAB = 1
3F9	2F9	อีนาเบิลอินเตอร์รัพต์	
3FA	2FA	กำหนดอินเตอร์รัพต์	
3FB	2FB	ควบคุมสายสื่อสาร	
3FC	2FC	ควบคุมโมเดม	
3FD	2FD	แสดงสถานะสายสื่อสาร	
3FE	2FE	แสดงสถานะโมเดม	

การเลือกหมายเลขอินพุตเอาต์พุตพอร์ตแยกเป็น 2 กลุ่ม กลุ่มหนึ่ง คือ COM_1 จะกำหนดหมายเลขพอร์ตจาก 3F8-3FE อีกกลุ่มหนึ่งถ้ากำหนดหมายเลขพอร์ตเป็น 2F8-2FE ในการเลือกหมายเลขรีจิสเตอร์หมายในกำหนดด้วยแอดเดรส 3 บิท คือ A_0 ,A_1 และ A_2 สำหรับการเลือกCOM_1 และ COM_2 เราใช้แอดเดรส A_8 เป็นตัวเลือกการเลือกนี้กำหนดเป็นตารางได้ ดังตารางที่ 2.4

ตารางที่ 2.4 กำหนดแอดเดรสสำหรับหมายเลขพอร์ตต่าง ๆ

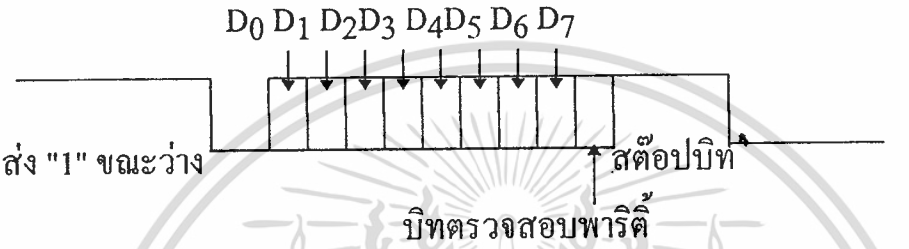
แอดแดรส 3F8-3FE และ 2F8-2FE											
A_9	A_8	A_7	A_6	A_5	A_4	A_3	A_2	A_1	A_0	DLAB	รีจิสเตอร์
1	1/0	1	1	1	1	1	x	x	x		
							0	0	0	0	บัฟเฟอร์สำหรับตัวรับข้อมูล(อ่าน) โฮลดิงสำหรับตัวส่งข้อมูล
							0	0	1	0	อีนาเบิลอินเตอร์รัพต์
							0	1	0	x	กำหนดอินเตอร์รัพต์
							0	1	1	x	ควบคุมสายสื่อสาร
							1	0	0	x	ควบคุมโมเดม
							1	0	1	x	แสดงสถานะสายสื่อสาร
							1	1	0	x	แสดงสถานะโมเดม
							1	1	1	x	ไม่ใช้
							0	0	0	1	แลตช์ตัวหาร (LSB)
							0	0	1	1	แลตช์ตัวหาร (MSB)

2.2.2 การอินเตอร์รัพต์

โครงสร้างการควบคุมอินเตอร์รัพต์ของชิพ 8259 ได้จัดลำดับการอินเตอร์รัพต์ไว้ 8 ระดับ สัญญาณรับอินเตอร์รัพต์ที่เข้าทางชิพ 8259 มี 8 เส้น คือ IRQ_0-IRQ_7 สำหรับกรณีของระบบสื่อสารอนุกรมได้ กำหนดสัญญาณการอินเตอร์รัพต์ไว้แล้ว คือ ให้IRQ_4 เป็นสัญญาณอินเตอร์รัพต์ต้องให้บิท 3 ของรีจิสเตอร์ควบคุมโดยโมเดมได้รับการเซตค่าเป็น"1" ก่อน จากนั้นข้อมูลอินเตอร์รัพต์ที่อยู่ในรีจิสเตอร์อีนาเบิลอินเตอร์รัพต์ จะเป็นตัวสั่งการอินเตอร์รัพต์

2.2.3 รูปแบบข้อมูลที่รับหรือส่ง

การสื่อสารข้อมูลของระบบนี้เป็นการสื่อสารแบบอะซิงโครนัส รูปแบบของข้อมูลจะมีสตาร์ตบิทบิตตรวจสอบพาริตี้ และสต็อบบิท โครงสร้างของข้อมูลแต่ละเฟรมเป็นดังรูปที่ 2.20



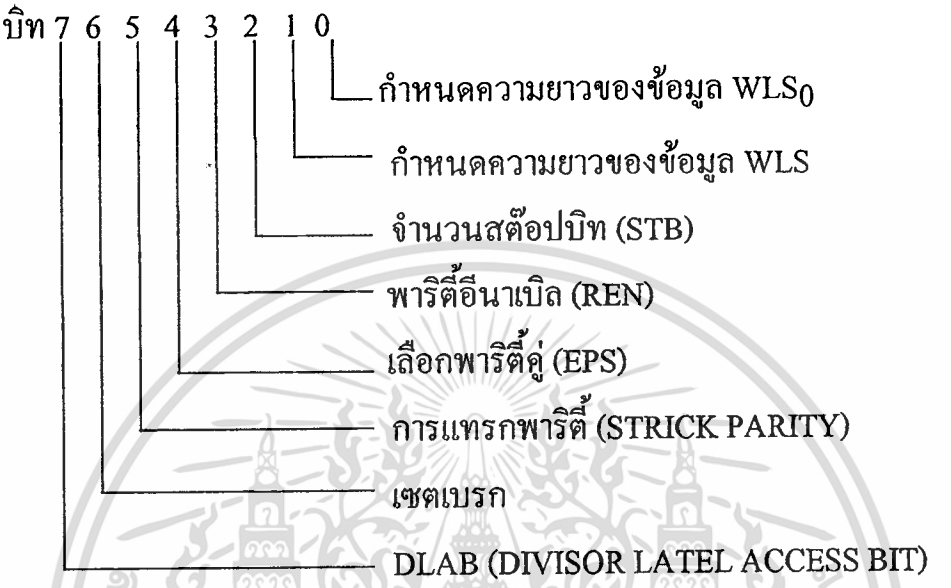
รูปที่ 2.20 รูปแบบข้อมูล 1 เฟรม

ข้อมูลบิทแรกของการส่งเป็นสตาร์ตบิท ระบบจะส่งสตาร์ตบิทก่อนหลังจากนั้นจะตามด้วยข้อมูล และแทรกด้วยบิท ตรวจสอบพาริตี้ตามด้วยสต็อปบิท ขนาดของข้อมูลมีค่าได้ตั้งแต่ 5-8 บิทแต่ทั่ว ๆ ไปใช้ 8 บิท สต็อปบิทมีได้ 1, 1.5 หรือ 2 บิท ค่าเหล่านี้สามารถกำหนดลงไปในรีจิสเตอร์ควบคุมสายสื่อสาร

2.2.4 รีจิสเตอร์ควบคุมสายสื่อสาร (line control register)

ในการควบคุมรูปแบบของข้อมูลแบบอะซิงโครนัสนั้น ผู้โปรแกรมจะต้องกำหนดค่าลงในรีจิสเตอร์ควบคุมสายสื่อสารรีจิสเตอร์ตัวนี้มี 8 บิท

พอร์ตแอดเดรสหมายเลข 3FB



รูปที่ 2.21 แสดงค่าของบิทในรีจิสเตอร์ควบคุมสายการสื่อสาร

บิท 0 และ 1 เป็นตัวกำหนดความยาวในการรับส่งโดยที่

บิท 0	บิท 1	ความหมาย
0	0	หมายถึงข้อมูลขนาด 5บิท
0	1	หมายถึงข้อมูลขนาด 6บิท
1	0	หมายถึงข้อมูลขนาด 7บิท
1	1	หมายถึงข้อมูลขนาด 8บิท

บิท 2 เป็นบิทที่ใช้ในการกำหนดจำนวนสต๊อปบิท ถ้าเป็น "0" หมายถึงใช้สต๊อปบิท 1 บุ๊ทแต่ถ้าบิท 2 เป็น "1" ในกรณีส่งแบบ 5 บิทจะมีความยาวของสต๊อปบิทเป็น 1.5 บิทแต่ถ้าส่งแบบ 6,7 หรือ 8 บิทความยาวของสต๊อปบิทจะเป็น 2

บิท 3 เป็นบิทแสดงการอีนาเบิลให้มีการตรวจสอบพาริตี้ โดยถ้าบิทนี้มีค่าเป็น 1 จะมีการเพิ่มพาริตี้

บิท 4 มีค่าเป็น "0" และบิท 3 มีค่าเป็น "1" จะมีการกำหนดเป็นพาริตี้คี่ แต่ถ้าบิทนี้มีค่าเป็น "1" จะเป็นพาริตี้คู่

บิท 5 เมื่อบิท 3 มีค่าเป็น "1" และบิท 5 มีค่าเป็น "1" และบิท 4 มีค่าเป็น "1" จะมีการแทรกหรือตรวจสอบพาริตี้ (Strick Parity) ด้วยเงื่อนไขกำหนดให้เป็น "0" และถ้าบิท 4 มีค่าเป็น "0" บิท 3 มีค่าเป็น "1" และบิท 5 มีค่าเป็น "1" จะมีการกำหนดพาริตี้เป็น "1"

บิท 6 เป็นบิทที่ควบคุมการเบรก เมื่อบิท 6 มีค่าเป็น "1" ส่วนของ SOUT จะได้รับการกำหนดให้เป็น "0" ตลอด

บิท 7 บิทนี้ทำหน้าที่เป็น DLAB บิทที่จะมีผลต่อการแลตช์ตัวหารดังที่กล่าวมาแล้ว

2.2.5 การโปรแกรมอัตราบอด (BOUD RATE GENERATOR)

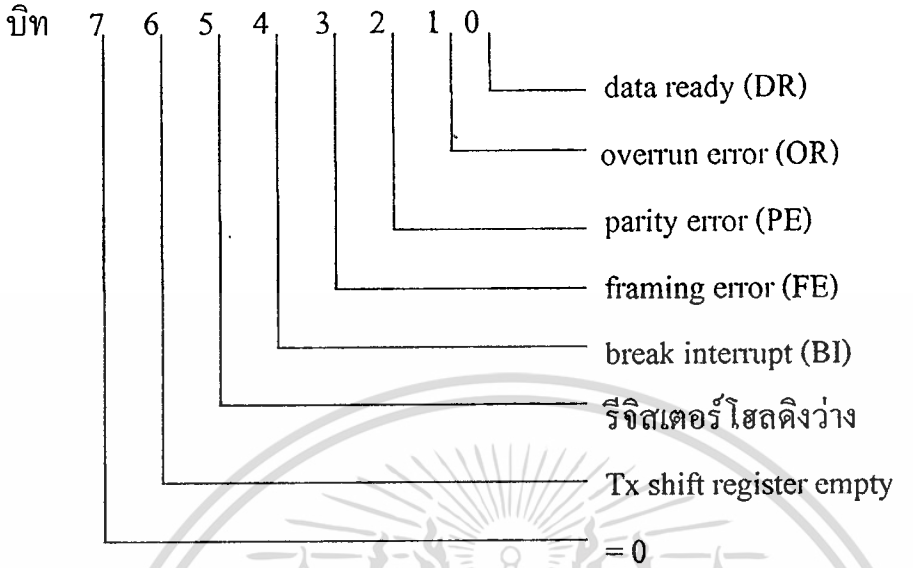
อัตราบอดได้รับการกำหนดเทียบกับสัญญาณนาฬิกา 1.8432 MH_Z และสามารถโปรแกรมตัวหารได้ตั้งแต่ 1-(2^{16}-1) ค่าความถี่เอาต์พุตของตัวกำหนดอัตราบอดมีค่าเท่ากับ 16xอัตราบอด ดังนั้นตัวหาร = ความถี่สัญญาณนาฬิกา / (อัตราบอด*16) การกำหนดอัตราบอดด้วยการกำหนดตัวหารนี้ตัวหารจึงเป็นค่าที่กำหนดใน รีจิสเตอร์ 2 ตัว ตัวหารนี้จะต้องถูกกำหนดค่าก่อน แล้วโปรแกรมลงมาในรีจิสเตอร์นี้ การกำหนด ต้องให้ DLAB = 1แล้วให้ลดลงมาในรีจิสเตอร์ $3F_8$ ซึ่งเรียงกันเป็น LSB ของตัวหารส่วน $3F_9$ เมื่อ DLAB=1 จะเป็นค่าของตัวหารMSB ค่าของตัวหารเมื่อเปรียบเทียบกับสัญญาณ 1.8432 MHz เป็นดังตารางที่ 2.5

ตารางที่ 2.5 ค่าตัวหารสำหรับการกำหนดอัตราบอด

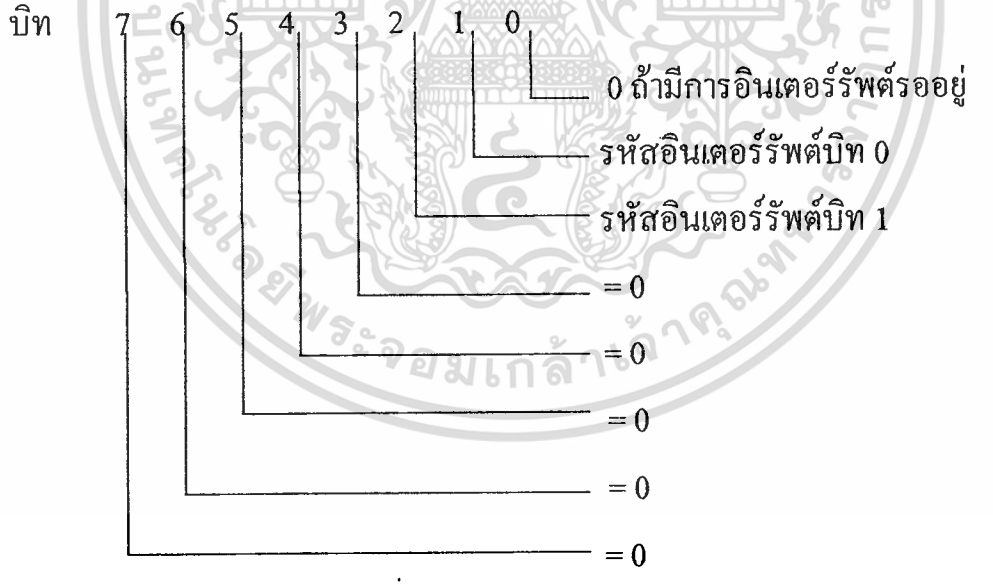
อัตราบอด	ตัวหาร		ค่าผิดพลาด
	ฐานสิบ	ฐานสิบหก	
50	2304	900	-
75	1536	600	-
110	1047	417	0.026
134.5	857	359	0.058
150	768	300	-
300	384	180	-
600	192	0C0	-
1200	96	060	-
1800	64	040	-
2000	58	03A	0.69
2400	48	030	-
3600	32	020	-
4800	24	018	-
7200	16	010	-
9600	12	00C	-

2.2.6 รีจิสเตอร์แสดงสถานะสายสื่อสาร(linestatusregister)

รีจิสเตอร์ตัวนี้เป็นรีจิสเตอร์ที่จะให้ข้อมูลแก่ซีพียูที่เกี่ยวกับการสื่อสารข้อมูลในสายสื่อสารค่าของบิทต่าง ๆ ในรีจิสเตอร์นี้ เป็นดังนี้



รูปที่ 2.22 แสดงค่าของบิทในรีจิสเตอร์แสดงสถานะสายสื่อสาร



รูปที่ 2.23 แสดงค่าของบิทในรีจิสเตอร์กำหนดอินเตอร์รัพต์

บิท 0 บิทนี้เป็นบิทที่บอกสถานะการรับข้อมูล ถ้าบิทนี้เป็น "1" แสดงว่าการรับข้อมูลเข้ามาในบัฟเฟอร์ได้ครบทุกบิทแล้ว บิทนี้จะได้รับการรีเซตให้เป็น"0" เมื่อซีพียูได้อ่านข้อมูลในบัฟเฟอร์ไปแล้ว หรือจะให้ซีพียูเขียนข้อมูลกลับมายังรีจิสเตอร์ก็ได้

บิท 1 บิทนี้ถ้ามีค่าเป็น "1" แสดงว่าเกิด OVERRUN ERROR (OR) กล่าวคือขณะที่มีข้อมูลที่บัฟเฟอร์แต่ CPU ยังไม่ได้อ่านไป ปรากฏว่ามีข้อมูลชุดใหม่มาเขียนทับบนบัฟเฟอร์นี้ บิทนี้จะรีเซตโดย CPU เมื่อ CPU อ่านค่าจาก รีจิสเตอร์นี้ไปแล้ว

บิท 2 บิทนี้ถ้ามีค่าเป็น "1" แสดงว่าเกิด Parity Error (PE) กล่าวคือถ้ามีการตรวจสอบบิทพาริตี้แล้วไม่เป็นไปตามที่กำหนดไว้ บิทนี้จะได้รับการรีเซตโดยซีพียูอ่านค่าจากรีจิสเตอร์ไปแล้ว

บิท 3 บิทนี้ถ้ามีค่าเป็น "1" แสดงว่าเฟรมของข้อมูลไม่เป็นไปตามที่กำหนด เช่น ตรวจสอบจำนวนบิทโดยดูที่พาริตี้ และสต๊อปบิทไม่เป็นไปตามที่กำหนด

บิท 4 บิทนี้เรียกว่า break interrupt (BI) บิทนี้จะได้รับการเซตให้มีค่าเป็น "1" ถ้าหากว่ารับข้อมูลอินพุตเป็น "0" เป็นเวลายาวนานกว่าเวิร์ดของการสื่อสาร

บิท 5 บิทนี้เป็นบิทที่บอกว่า 8250 พร้อมที่จะรับข้อมูลจากสายสื่อสาร บิทนี้จะได้รับการเซตให้มีค่าเป็น "1" บิทนี้ยังคงสร้างสัญญาณอินเตอร์รัพต์เพื่อส่งไปบอกซีพียูด้วย บิทนี้จะมีสถานะเซตเมื่อต้องการส่งถ่ายข้อมูลจากโฮลด์รีจิสเตอร์ไปยังชิพรีจิสเตอร์เพื่อพร้อมที่จะส่ง

บิท 6 เป็นบิทที่จะบอกว่าชิพรีจิสเตอร์ว่างหรือเปล่า บิทนี้จะได้รับการเซตให้มีค่าเป็น "1"เพื่อบอกว่าพร้อมส่งแล้ว

บิท 7 จะเป็นบิท "0" ตลอด

2.2.7 รีจิสเตอร์กำหนดอินเตอร์รัพต์ (IIR-Interrupt Identification Register)

ไอซี 8250 มีขีดความสามารถในการส่งอินเตอร์รัพต์ภายในชิพ เพื่อว่าให้การทำงานระหว่าง 8250 กับซีพียูเป็นไปอย่างมีประสิทธิภาพสูง เพื่อให้ผู้เขียนซอฟต์แวร์สามารถเขียนซอฟต์แวร์ได้ง่าย และสั้นลงได้มาก 8250 กำหนดความ

สำคัญของอินเตอร์รัพต์ไว้ 4 ระดับ คือ

ระดับแรก - สถานะการับข้อมูลจากสายสื่อสาร

ระดับที่สอง - การพร้อมรับข้อมูล

ระดับที่สาม - ขณะรีจิสเตอร์โฮลดิงสำหรับส่งข่าว

ระดับที่สี่ - สัญญาณสถานะโมเดม

ในขณะที่มีความต้องการอินเตอร์รัพต์หลายระดับพร้อมกัน 8250 จะให้ระดับที่มีความสำคัญน้อยกว่ารอไว้ก่อน โดยเก็บสถานะการอินเตอร์รัพต์นี้ไว้ในรีจิส เตอร์กำหนดอินเตอร์รัพต์ความหมายของรีจิสเตอร์นี้มีดังนี้

บิท 0 เป็นบิทที่ใช้แสดงว่ามีอินเตอร์รัพต์เกิดหรือไม่ ซึ่งสามารถให้ซีพียูตรวจสอบดูด้วยวิธีการ POLLING ได้ ถ้าบิทนี้เป็น "1" หมายถึง ไม่มีอินเตอร์รัพต์เกิดขึ้น

บิท 1-2 เป็นบิทที่แสดงความหมายบอกว่าการอินเตอร์รัพต์ที่เกิดขึ้นนั้นมาจากอินเตอร์รัพต์ฟังก์ชันใด

บิท 3-7 มีค่าเป็น 0

2.2.8 รีจิสเตอร์อีนาเบิลอินเตอร์รัพต์ (INTRRT - interrupt enable register)

ในCOM$_1$ เมื่อให้ DLAB = 0 พอร์ต3F9 จะเป็นรีจิสเตอร์อีนาเบิลอินเตอร์รัพต์ ผู้ใช้สามารถกำหนดให้เกิดอินเตอร์รัพต์หรือไม่ก็ได้ โดยการกำหนดค่าลงในรีจิสเตอร์นี้ จากที่กล่าวแล้วว่า การอินเตอร์รัพต์ของ 8250 นี้มี 4 แบบ ดังนั้นจึงต้องกำหนดการอินนาเบิลได้ทั้ง 4แบบ โดยการใช้ข้อมูลแต่ละบิทของรีจิสเตอร์นี้เพื่อการกำหนดการอีนาเบิล ข้อมูลที่อยู่ในรีจิสเตอร์นี้มีความหมายดังนี้

กำหนดพอร์ต 3F9 DLAB = 0

รูปที่ 2.24 ค่าของบิทในรีจิสเตอร์อีนาเบิลอินเตอร์รัพต์

ตารางที่ 2.6 ฟังก์ชั่นการอินเตอร์รัพต์รหัสอินเตอร์รัพต์

บิท 2	บิท 1	บิท 0	ระดับความสำคัญ	ชนิดของการอินเตอร์รัพต์	แหล่งเกิดอินเตอร์รัพต์	การรีเซตควบคุมอินเตอร์รัพต์
0	0	1	สูงสุด	ไม่เกิดสถานะการรับข้อมูลจากสายส่งสื่อสาร	ไม่เกิด overrun error framming error break error parity error	อ่านข้อมูลจากรีจิสเตอร์สถานะ สายสื่อสาร
1	0	0	ที่สอง	การพร้อมรับข้อมูลโฮลดิงรีจิสเตอร์สำหรับส่งข่าว	มีข้อมูลที่ตัวรับโฮลดิงรีจิสเตอร์สำหรับส่งข่าว	การอ่านข้อมูลจากบัฟเฟอร์ อ่านรีจิสเตอร์กำหนดอินเตอร์รัพต์ IRR หรือเขียนลงไปยังโฮลดิงรีจิสเตอร์สำหรับส่งข่าว
0	1	0	ที่สาม			

บิท 0 บิทนี้ได้รับการเซตเป็น "1" เมื่อต้องการอีนาเบิลอินเตอร์รัพต์การพร้อมรับข้อมูล

บิท 1 บิทนี้ได้รับการเซตเป็น "1" เมื่อต้องการอีนาเบิลอินเตอร์รัพต์โฮลดิงรีจิสเตอร์ว่าง

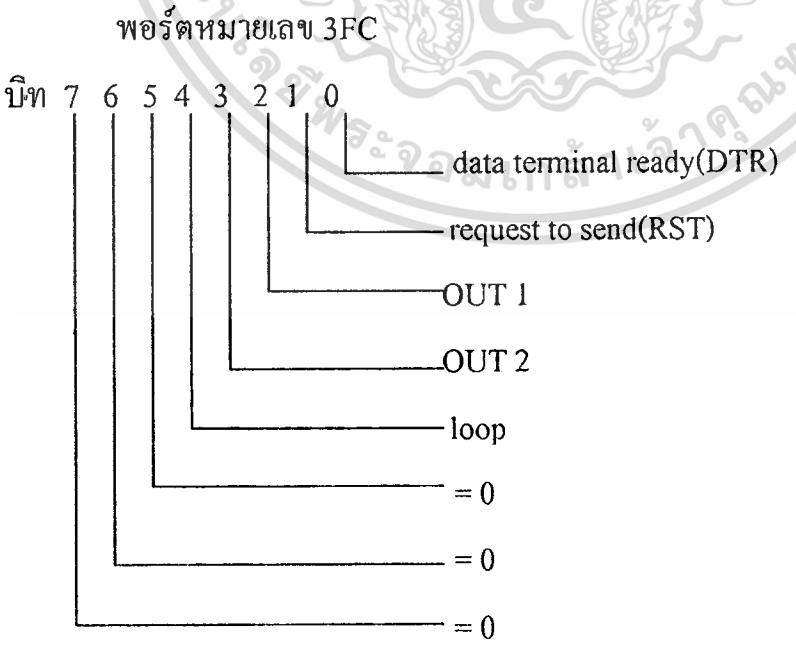
บิท 2 บิทนี้ได้รับการเซตเป็น "1" เมื่อต้องการอีนาเบิลอินเตอร์รัพต์จากสถานะการรับข้อมูลการสื่อสาร

บิท 3 บิทนี้ได้รับการเซตเป็น "1" เมื่อต้องการอีนาเบิลอินเตอร์รัพต์จากสถานะโมเดม

บิท *4-7* ได้รับการกำหนดเป็น 0 เสมอ

2.2.9 รีจิสเตอร์ควบคุมโมเดม (modem control register)

รีจิสเตอร์ตัวนี้มีไว้ให้ซีพียูส่งผ่านข้อมูลมาเก็บเพื่อเป็นรหัสสำหรับการควบคุมการทำงานของโมเดม การกำหนดพอร์ตของรีจิสเตอร์ตัวนี้คือ 3FC ข้อมูลต่างๆ ที่มีในรีจิสเตอร์ตัวนี้มีความหมายดังนี้ บิท 0 บิทนี้มีความหมายถึงการควบคุมสัญญาณ DTR เมื่อบิทนี้มีค่าเป็น "1" เอาต์พุตที่ DTRจะได้รับการกำหนดให้เป็น "0" และถ้าบิทนี้มีค่าเป็น "0" เอาต์พุตที่ DTR จะได้รับการกำหนดเป็น "1"



รูปที่ 2.25 ค่าของบิทในรีจิสเตอร์ควบคุมโมเดม

บิท 1 บิทนี้มีความหมายถึงสัญญาณ RTS ซึ่งมีผลเหมือนกับบิท 0 ในกรณีของ DTR

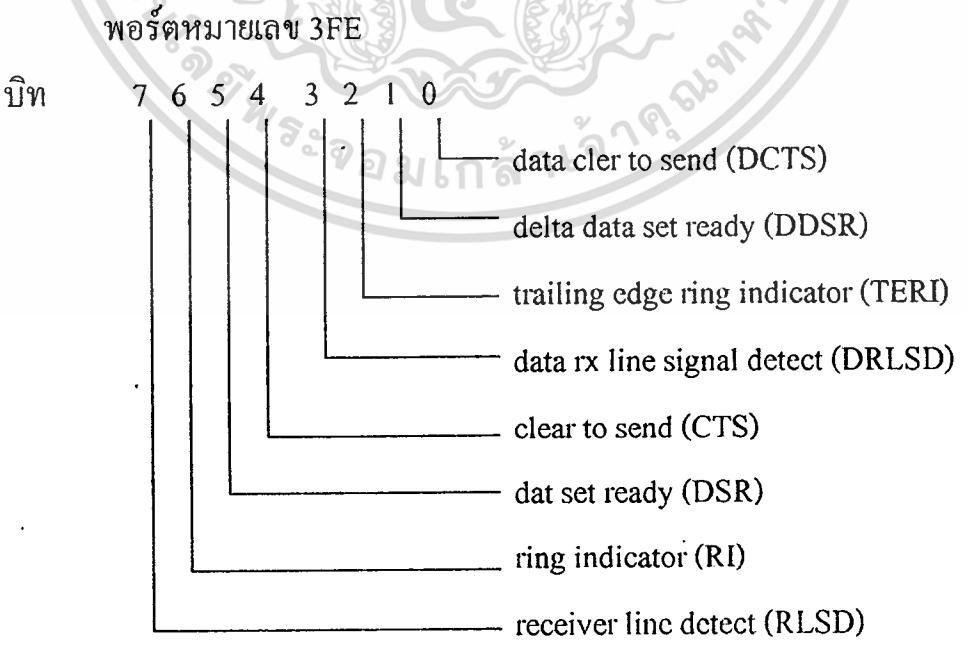
บิท 2 บิทนี้ใช้ควบคุมขาเอาต์พุท 1 (OUT 1) ซึ่งจะมีผลเหมือนบิท 0

บิท 3 บิทนี้ใช้ควบคุมขาเอาต์พุท 1 (OUT 1) ซึ่งจะมีผลเหมือนบิท 0

บิท 4 บิทนี้จะใช้สำหรับการกำหนดวงรอบสำหรับการตรวจสอบ 8250 เมื่อบิท 4 นี้ได้รับการเซตเป็น "1" สิ่งที่จะเกิดขึ้นเป็นดังนี้ ข้อมูลที่ SOUT จะได้รับการเซตเป็นลอจิก "1" ขาข้อมูลอินพุทSIN จะได้รับการแยกตัวออก ข้อมูลของเอาต์พุทชิพรีจิสเตอร์จะได้รับการป้อนกลับมารีจิสเตอร์ข้อมูลอินพุท ส่วนสัญญาณ CTS, DSR, RLSD และ RI จะได้รับการแยกออกจากระบบแต่สัญญาณควบคุมโมเดม คือ DTR, RTS, OUT1 และ OUT 2 จะต่อเข้ากับสัญญาณทั้งสี่ที่เป็นอินพุท ดังนั้นจึงตรวจสอบระบบการทำงานได้

2.2.10 รีจิสเตอร์แสดงสถานะโมเดม

รีจิสเตอร์ตัวนี้จะเป็นตัวที่รับสถานะจากโมเดมมาเก็บไว้เพื่อให้ซีพียูสามารถอ่านตรวจสอบดูได้ สถานะของข้อมูลจะแอกทีฟเมื่อมีข้อมูลเป็น "1" และจะได้รับการรีเซตเมื่อซีพียูอ่านข้อมูลในรีจิสเตอร์นี้ไป พอร์ตที่ใช้กำหนดเป็นพอร์ตหมายเลข 3FE ข้อมูลภายในรีจิสเตอร์นี้เป็นดังนี้



รูปที่ 2.26 ค่าของบิทในรีจิสเตอร์แสดงสถานะโมเดม

บิท 0 บิทนี้ใช้สำหรับแสดงการเปลี่ยนแปลงของสัญญาณ CTS กล่าวคือ เมื่อขา CTS ของ 8250 ได้เปลี่ยนสถานะหลังจากที่ซีพียูได้อ่านสถานะนี้ไปแล้ว บิทนี้จะบอกด้วยเซตและเมื่อซีพียูอ่านก็จะได้รับการรีเซต "0" และจะได้รับการเซต "1" เมื่อมีการเปลี่ยนสถานะที่ขา CTS

บิท 1 เหมือนบิท 0 แต่ เป็นบิทที่แสดงสถานะการเปลี่ยนแปลงของขา DSR

บิท 2 บิทนี้เป็นบิทที่แสดงว่าสัญญาณ RI ซึ่งเป็นอินพุตของ 8250 ได้รับการเปลี่ยนจากออน "1" มาเป็นออฟ "0"

บิท 3 บิทที่เหมือนบิท 0 แต่เป็นบิทแสดงสถานะแสดงการเปลี่ยนแปลงของ line signaldetector ซึ่งเป็นขาอินพุต RLSD

บิท 4 บิทนี้เก็บสัญญาณคอมพลีเมนต์กับสัญญาณที่ขา CTS

บิท 5 บิทนี้เก็บสัญญาณคอมพลีเมนต์กับสัญญาณที่ขา DSR

บิท 6 บิทนี้เก็บสัญญาณคอมพลีเมนต์กับสัญญาณที่ขา RI

บิท 7 บิทนี้เก็บสัญญาณคอมพลีเมนต์กับสัญญาณที่ขา RLSD

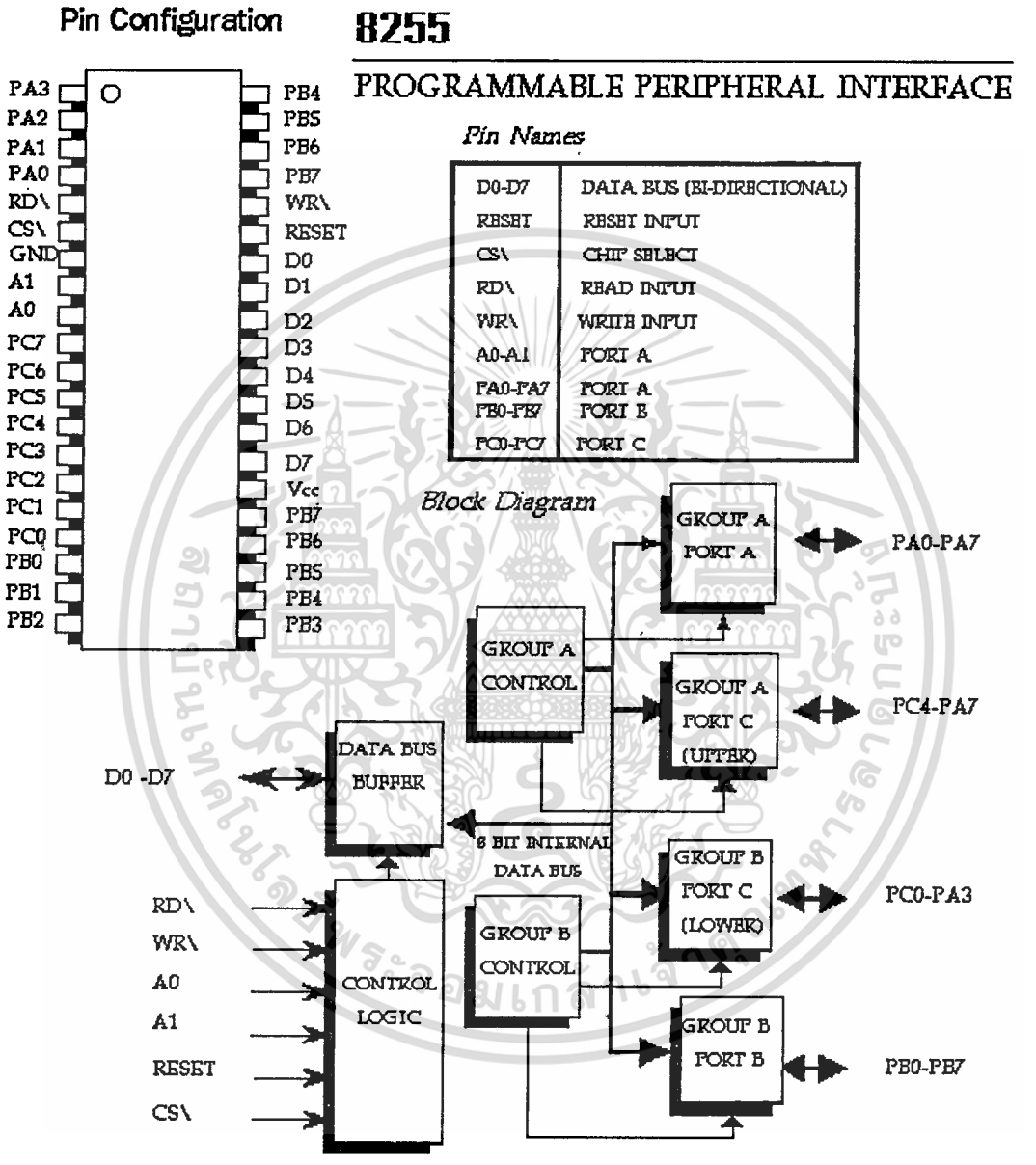
อนึ่งถ้าบิท 4 ของ MCR ไดรับการเซตหรือให้ทำลูป(loop) ตรวจสอบข้อมูลในบิท 4 จะเหมือนกับ RTS ใน MCR ข้อมูลในบิท 5 จะเหมือนกับ DTR ใน MCR ข้อมูลในบิท 6 จะเหมือนกับ OUT_1 ใน MCR ข้อมูลในบิท 7 จะเหมือนกับ OUT_2 ใน MCR

2.2.11 รีจิสเตอร์บัฟเฟอร์สำหรับตัวรับข้อมูล(receiver buffer register)

เป็นรีจิสเตอร์สำหรับการรับข้อมูลที่มาจากสายสื่อสารสัญญาณ พอร์ตที่กำหนดคือหมายเลขแอดเดส 3F8 ขณะที่ DLAB = 0 หากซีพียูอ่านข้อมูลที่รีจิสเตอร์นี้หมายถึงได้อ่านข้อมูลที่มาจากสายสัญญาณสื่อสารนั้นเอง 2.2.12 รีจิสเตอร์โฮลดิงสำหรับตัวส่งข้อมูล(transmitter holding register)

เป็นรีจิสเตอร์บัฟเฟอร์สำหรับส่งข้อมูล รีจิสเตอร์นี้จะรับข้อมูลจากซีพียู โดยที่กำหนดพอร์ตเป็นหมายเลข 3F8 เมื่อ DLAB = 0 ข้อมูลของซีพียูที่เอาต์พุทที่พอร์ตนี้ก็จะส่งต่อออกไปยังสายสารข้อมูล

2.4 การใช้งาน 8255



รูปที่ 2.27 แผนภาพแบบบล็อกภายในและขาสัญญาณของไอซีเบอร์ 8255

จากแผนภาพในรูปที่ 2.27 จะเห็นว่า 8255 ประกอบด้วยบล็อกการทำงาน โดยบล็อก 4 บล็อกทางขวามือจะเป็นส่วนที่เชื่อมต่อกับอุปกรณ์ภายนอก โดยตรงโดยผ่านทางเส้นสัญญาณที่ระบุชื่อว่า PA0-PA7 (พอร์ต A),PB0-PB7 (พอร์ต B)และ

PC0-PC7 (พอร์ต C) สำหรับบล็อกถัดเข้ามาบริเวณส่วนกลางที่มีชื่อว่า GROUP A CONTROL และ GROUP B CONTROL ทำหน้าที่กำหนดการทำงานของพอร์ตทั้ง 3 สำหรับบล็อกการทำงานทางด้านซ้ายที่มีชื่อว่า Data bus buffer และ read/write control logic ทำหน้าที่เชื่อมต่อระหว่างระบบบัสของไมโครคอนโทรลเลอร์ กับ 8255 เพื่อรับหรือส่งข้อมูลระหว่างกันตามระดับลอจิกของขาสัญญาณ RDและ WRตามลำดับ

2.3.1 การจำแนกกลุ่มของพอร์ต 8255

ในพอร์ตต่าง ๆ ของ 8255 จะเป็นพอร์ตแบบขนานที่ประกอบด้วยเส้นสัญญาณ 8 เส้น นอกจากนี้ยังสามารถอ้างถึงแต่ละบิทของเส้นสัญญาณได้โดยอิสระ อย่างไรก็ตาม 8255 ได้จัดกลุ่มของพอร์ตเหล่านี้ออกเป็น 2 กลุ่ม คือ GROUP A และ GROUP B เพื่อประโยชน์ในการกำหนดรูปแบบการทำงานของพอร์ตดังตารางที่ 2.7

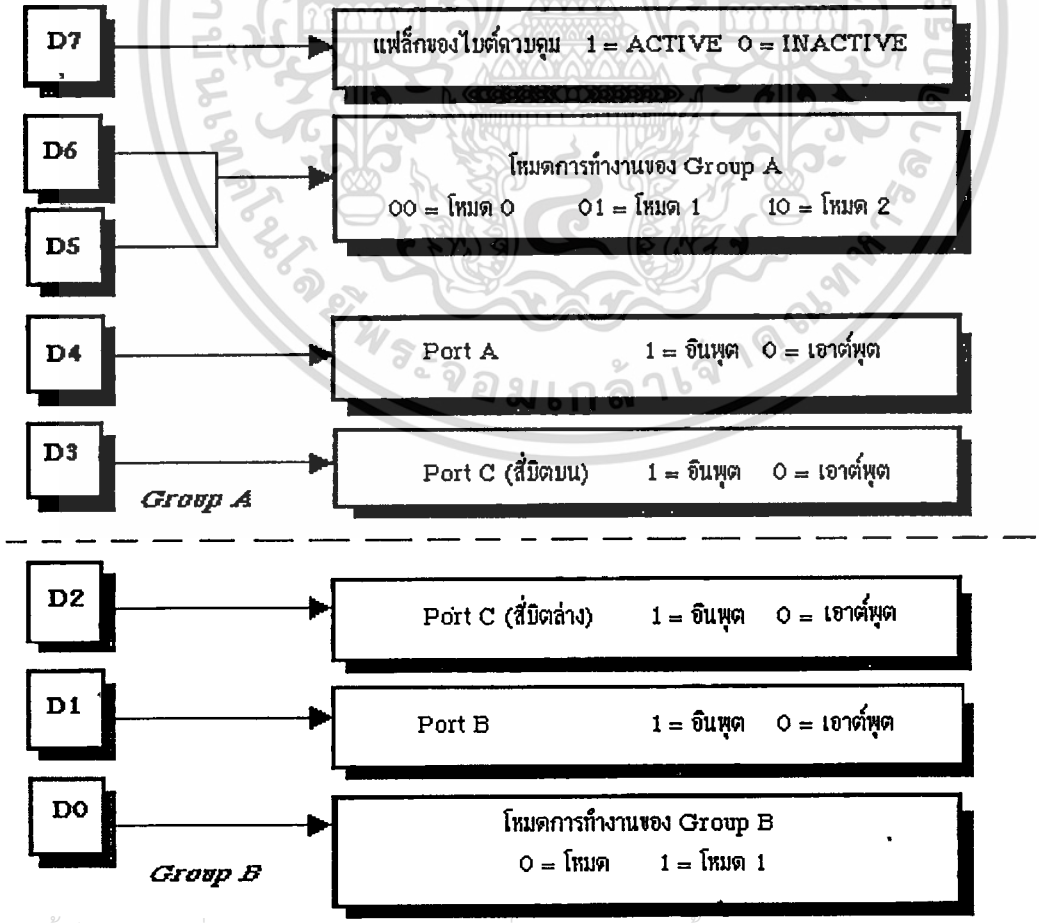
ตารางที่ 2.7 แสดงการจำแนกกลุ่มของพอร์ต 8255

ชื่อกลุ่ม	ลักษณะ
GROUP A	พอร์ต A จำนวน 8 บิท (ทุกบิตของพอร์ต) พอร์ต C จำนวน 4 บิท (เฉพาะ 4 บิตบนของพอร์ต)
GROUP B	พอร์ต B จำนวน 8 บิท (ทุกบิตของพอร์ต) พอร์ต C จำนวน 4 บิท (เฉพาะ 4 บิตล่างของพอร์ต)

ตารางที่ 2.8 แสดงหน้าที่การทำงานของขาสัญญาณไอซี 8255

สัญญาณ	ความหมาย
D0-D7	กลุ่มของเส้นสัญญาณข้อมูลของ 8255 เมื่อมีการเขียนหรืออ่าน
CS\	สัญญาณเลือกอุปกรณ์ เมื่อขาสัญญาณนี้เป็นระดับลอจิกต่ำ ซีพียูก็สามารถเขียนหรืออ่าน ข้อมูลจาก 8255 ได้
RD\	สัญญาณบอกสถานะต้องการอ่านข้อมูลจาก 8255
WR\	สัญญาณบอกสถานะต้องการเขียนข้อมูลจาก 8255
A0-A1	สัญญาณระบุตำแหน่งรีจิสเตอร์ภายใน 8255 ที่ต้องการ
RESET	สัญญาณการรีเซตวงจรทำงานภายใน 8255 เพื่อเริ่มต้นใหม่
PA0-PA7	กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับ พอร์ต A
PB0-PB7	กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับ พอร์ต B
PC0-PC7	กลุ่มของสัญญาณ 8 เส้น เมื่อทำการติดต่อกับ พอร์ต C

2.3.2 รูปแบบคำสั่งเพื่อกำหนดการทำงานของ 8255



รูปที่ 2.33 ความหมายของบิตภายในไบต์ข้อมูลควบคุมสำหรับ 8255

การกำหนดให้พอร์ตทั้งสามของ 8255 ทำงานในลักษณะต่าง ๆ กันหรือที่เรียกว่า โหมดการทำงาน (MODE) จะเริ่มด้วยการส่งค่าข้อมูลไบต์หนึ่งให้กับรีจิสเตอร์ควบคุมการทำงานภายใน 8255 ข้อมูลนี้เรียกว่า ไบต์ข้อมูลควบคุม (Control Word) โดยแต่ละบิตของข้อมูลนี้จะมีความหมายที่ระบุถึงความต้องการต่าง ๆ ไปดังแสดงดังรูปที่ 2.28 การส่งข้อมูลไบต์นี้จะต้องเริ่มต้นเป็นลำดับแรกก่อนที่จะได้มีการดำเนินการใด ๆ กัน 8255 ทั้งสิ้น

ตามความต้องการของบิตภายในตารางของรูปที่ 2.28 จะเห็นว่าการเลือกให้พอร์ตใดทำหน้าที่เป็นพอร์ตอินพุตก็เพียงแต่กำหนดค่าของข้อมูล 1 ให้กับบิตที่เกี่ยวข้องกับพอร์ตนั้น หรือกรณีตรงกันข้ามสำหรับการเอาต์พุตก็เพียงการกำหนดค่าข้อมูล 0 เท่านั้น อย่างไรก็ตามการกำหนดให้ไบต์ข้อมูลควบคุมนั้นมีผลอย่างถูกต้อง ก็จะต้องทำการกำหนดให้บิต D7 มีค่าเป็น 1 เสมอ สำหรับบิตที่บอกถึงโหมดการทำงาน (บิต D6 - D5 และ D2) นั้นจะได้กล่าวถึงในหัวข้อถัดไป

2.3.3 การเชื่อมต่อ 8255 กับ 8051

เมื่อพิจารณาแผนภาพของ 8255 จะเห็นว่ามีขาสัญญาณแอดเดรสจำนวน 2 เส้น คือ A0 และ A1 ทำให้ตำแหน่งของแอดสเดรสที่จะอ้างถึงได้มีค่าเป็น 2^2 หรือเท่ากับ 4 ตำแหน่งซึ่งแต่ละตำแหน่งจะมีความหมายถึงการระบุรีจิสเตอร์ หรือ พอร์ตภายใน 8255 ดังตารางที่ 2.9

ตารางที่ 2.9 แสดงการเชื่อมต่อ 8255 กับ 8051

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

เมื่อพิจารณาค่าของแอดเดรสเหล่านี้ร่วมกับระดับลอจิกของสัญญาณ RD และ WR จะเป็นการอ่าน หรือ เขียนข้อมูลทางขาสัญญาณ D0 - D7 ให้กับรีจิสเตอร์นั้นดังตารางที่ 2.10

ตารางที่ 2.10 แสดงความหมายโดยเมื่อรวมกับสัญญาณ RD และ WR

RD\	WR\	A1	A0	ความหมาย
0	1	0	0	ส่ง (หรือเขียน) ข้อมูลให้กับพอร์ต A
1	0	0	0	รับ (หรืออ่าน) ข้อมูลจากพอร์ต A
0	1	0	1	ส่ง (หรือเขียน) ข้อมูลให้กับพอร์ต B
1	0	0	1	รับ (หรืออ่าน) ข้อมูลจากพอร์ต B
0	1	1	0	ส่ง (หรือเขียน) ข้อมูลให้กับพอร์ต C
1	0	1	0	รับ (หรืออ่าน) ข้อมูลจากพอร์ต C
0	1	1	1	ส่ง (หรือเขียน) ข้อมูลให้กับรีจิสเตอร์ควบคุม
1	0	1	1	เป็นสภาวะที่ไม่ถูกต้อง

2.3.4 การทำงานโหมด 0 ของ 8255

การทำงานในโหมดนี้จะทำให้พอร์ตต่าง ๆ มีหน้าที่เป็นพอร์ตอินพุตหรือเอาต์พุทได้เพียงลักษณะเดียวเท่านั้น รูปแบบการทำงานในโหมดนี้จะมีการกำหนดบิต เมื่อต้องการให้พอร์ต A,B และ C ทำหน้าที่เป็นพอร์ตเอาต์พุตทั้งหมดดังตารางที่ 2.11

ตารางที่ 2.11 แสดงค่าของข้อมูลที่ใช้ควบคุม 8255

ตำแหน่งบิท	ค่าข้อมูล	ความหมาย
D7	1	ระบุให้ทราบว่าเป็นไบต์ข้อมูลควบคุม
D6 และ D5	00	กำหนดโหมดการทำงานให้กับพอร์ต A เป็นโหมด 0
D4	0	ระบุว่าพอร์ต A เป็นการเอาต์พุต (output) ข้อมูล
D3	0	กำหนดให้เส้นสัญญาณสี่บิทบนของพอร์ต C เป็นการเอาต์พุต
D0	0	กำหนดโหมดการทำงานให้กับพอร์ต B เป็นโหมด 0
D1	0	ระบุว่าพอร์ต B เป็นการเอาต์พุตข้อมูล
D2	0	กำหนดให้เส้นสัญญาณสี่บิทล่างของพอร์ต C เป็นการเอาต์พุต

2.3.5 การทำงานโหมด 1 ของ 8255

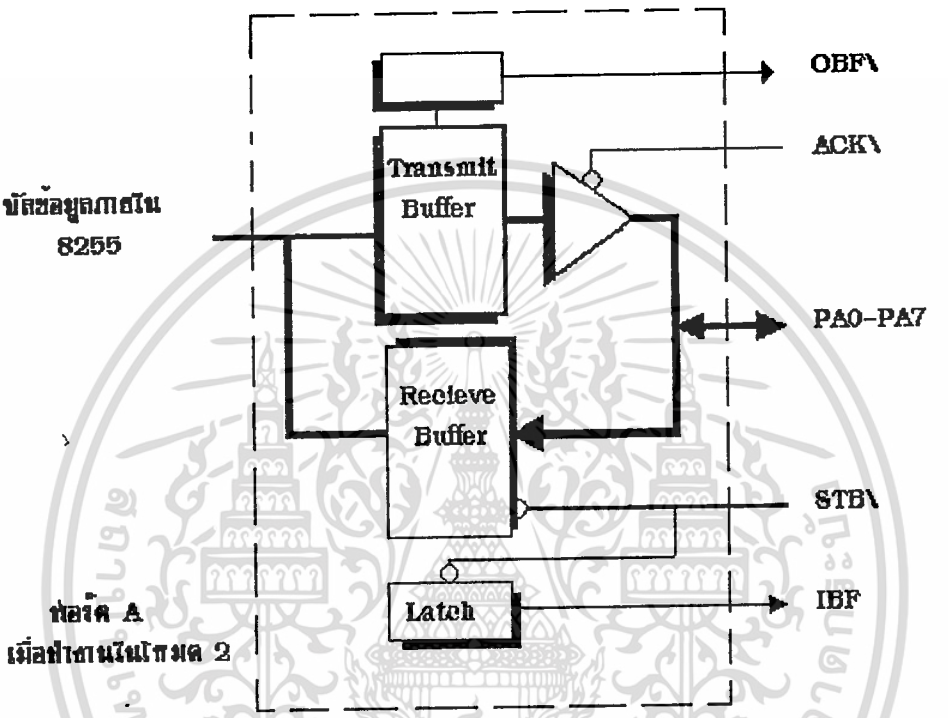
การทำงานในโหมด 1 จะทำให้พอร์ต A,B ใช้งานเป็นอินพุตหรือเอาท์พุตในลักษณะเดียวกับโหมด 0 เพียงแต่พอร์ต C จะถูกนำไปใช้เป็นพอร์ตสำหรับการบอกสภาวะการติดต่อเท่านั้น โดยเส้นสัญญาณสี่บิตบน (PC7-PC4) จะใช้งานร่วมกับพอร์ต A เส้นสัญญาณสี่บิตล่าง(PC3-PC0) จะใช้งานร่วมกับพอร์ต B

ตารางที่ 2.12 แสดงหน้าที่ของเส้นสัญญาณภายในพอร์ต C เมื่อทำงานในโหมด 1

เส้นสัญญาณ	สภาวะติดต่อสำหรับการอินพุต	สภาวะติดต่อสำหรับการเอาต์พุต
PC0	สัญญาณ INTR ของพอร์ต B	สัญญาณ INTR ของพอร์ต B
PC1	สัญญาณ IBF ของพอร์ต B	สัญญาณ OBF\ ของพอร์ต B
PC2	สัญญาณ STB\ ของพอร์ต B	สัญญาณ ACK\ ของพอร์ต B
PC3	สัญญาณ INTR ของพอร์ต A	สัญญาณ INTR ของพอร์ต A
PC4	สัญญาณ STB\ ของพอร์ต A	การอินพุต/เอาต์พุตตามปกติ
PC5	สัญญาณ IBF ของพอร์ต A	การอินพุต/เอาต์พุตตามปกติ
PC6	การอินพุต/เอาต์พุตตามปกติ	สัญญาณ ACK\ ของพอร์ต A
PC7	การอินพุต/เอาต์พุตตามปกติ	สัญญาณ OBF\ ของพอร์ต A

2.3.6 การทำงานโหมด 2 ของ 8255

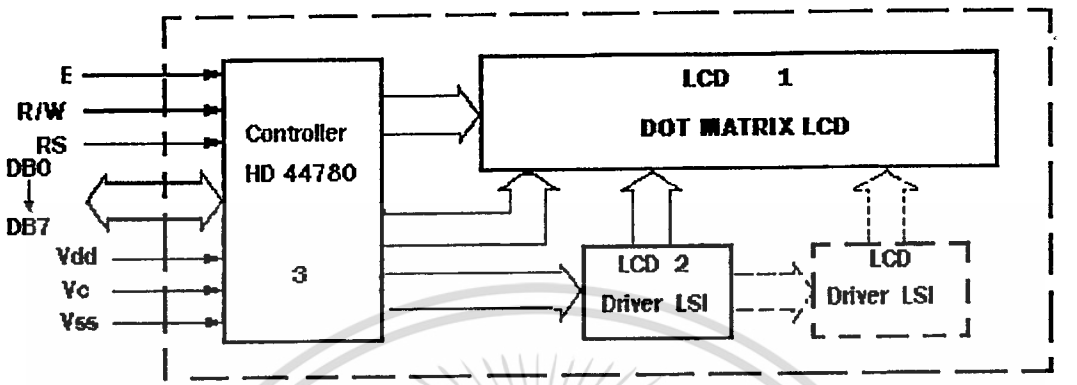
เมื่อ 8255 ได้รับการกำหนดให้ทำงานในโหมด 2 ซึ่งกำหนดไว้ใช้เฉพาะกับการทำงานของพอร์ต A เท่านั้น โดยจะมีลักษณะเป็นพอร์ตข้อมูลแบบ 2 ทิศทาง



รูปที่ 2.29 แผนภาพแสดงหลักการทำงานของ 8255 เมื่อได้รับการกำหนดให้ทำงานในโหมด 2

จากรูปที่ 2.29 จะเห็นว่า เส้นสัญญาณ PA0-PA7 จะถูกเชื่อมต่อกับบล็อกของวงจรแล็ตทั้งหมด โดยแล็ตสำหรับข้อมูลส่งออก มีหน้าที่สำหรับการค้างข้อมูลที่ 8051 ทำการเขียนมายังพอร์ต A และรอคอยให้อุปกรณ์ภายนอกมาอ่านข้อมูลนี้ไปจาก 8255 ส่วนแล็ตสำหรับรับข้อมูลเข้า ทำหน้าที่สำหรับเก็บข้อมูลที่อุปกรณ์ภายนอกส่งมาให้กับพอร์ต A

2.4 LCD Module



รูปที่ 2.30 แสดงโครงสร้างทั่วไปของ LCD module

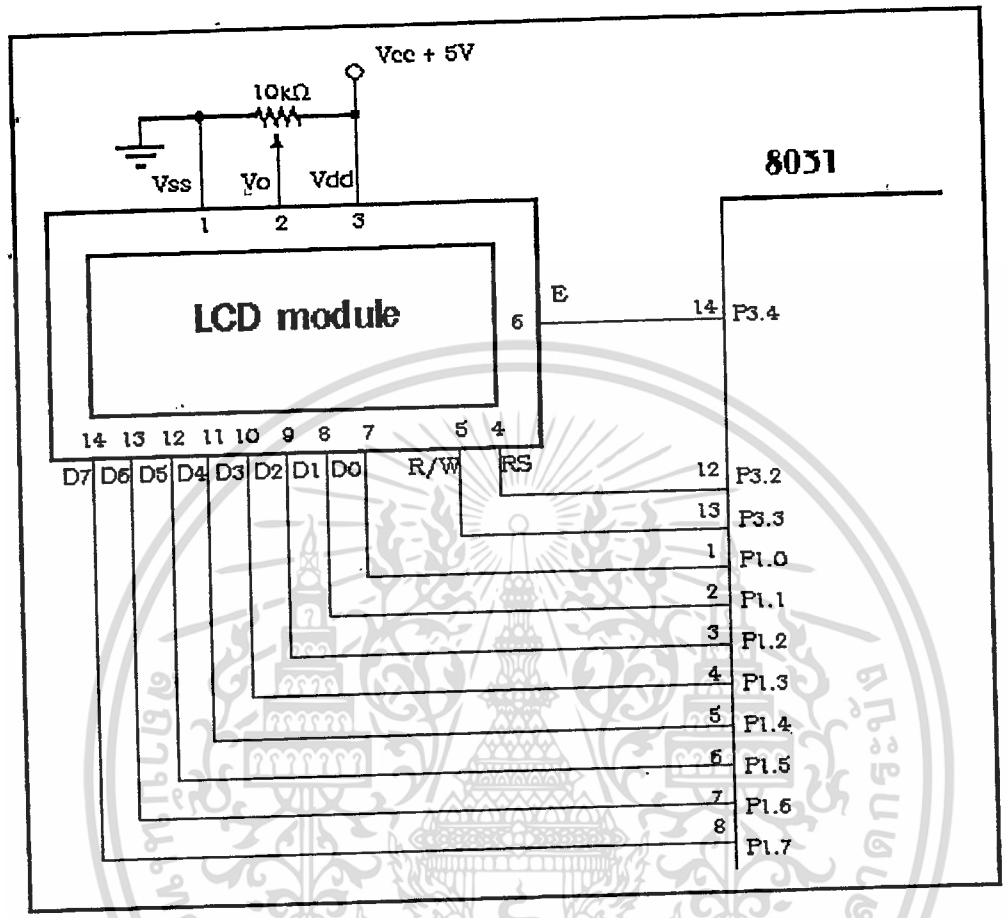
LCD Module มีส่วนประกอบที่สำคัญดังนี้

1) Dot Matrix LCD : เป็นส่วนที่ทำหน้าที่แสดงผล ซึ่งใช้หลักการหักเหของแสงผ่านผลึกโดยจะประกอบไปด้วยจุด (pixel) จำนวนมากที่สามารถบังคับให้ติดหรือดับได้ทุกจุด

2) Driver : เป็นวงจรที่ใช้ขับ LCD ส่วนใหญ่จะใช้ชิปเบอร์ HD44110H

3) Controller : เป็นส่วนที่ใช้ควบคุมการทำงานทั้งหมดของ LCD Module โดยจะรับข้อมูลจากภายนอกมาจัดการให้ LCD แสดงผลในรูปแบบต่าง ๆ ส่วนใหญ่จะใช้ชิปเบอร์ HD44780 ซึ่งมีใช้งานในแบบ Character LCD Module

ชิปคอนโทรลเลอร์เบอร์ HD44780 สามารถต่อใช้งานได้ทั้งแบบ 4 Bit 2 Operation หรือ แบบ 8 Bit 1 Operation เนื่องจาก ไมโครคอนโทรเลอร์ตระกูล MCS-51 มีดาต้าบัสขนาด 8 บิต ดังนั้นก็จะกล่าวถึงเฉพาะการติดต่อในแบบ 8 Bit 1 Operation เท่านั้น ตัวอย่างวงจรการอินเตอร์เฟส MCS-51 กับ LCD Module มีดังรูปที่ 2.31



รูปที่ 2.31 ตัวอย่างการอินเตอร์เฟส MCS-51 กับ LCM

จากรูปจะเห็นว่า LCD Module ติดต่อกับ MCS-51 โดย

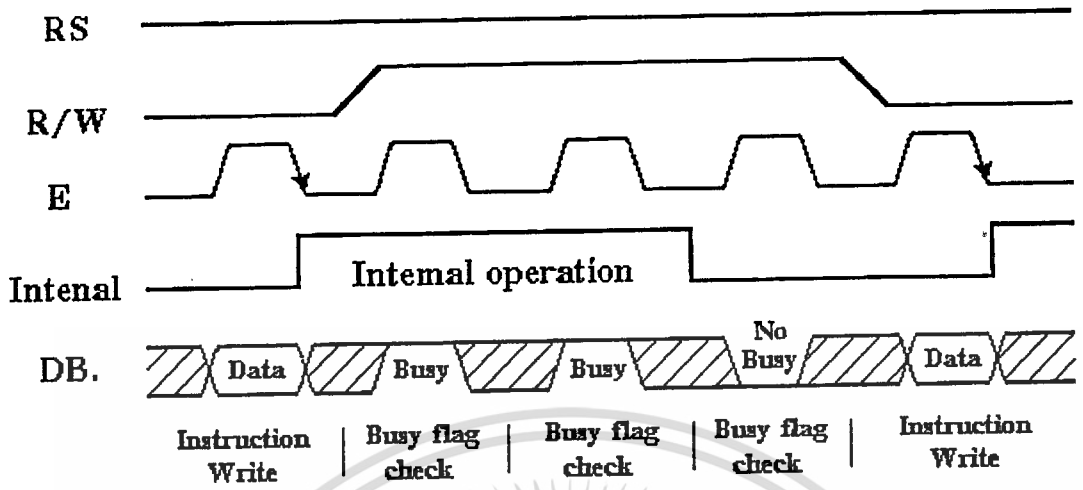
-ใช้ขา P1.0-P1.7 เป็นดาต้าบัส (DB0-DB7) ในการติดต่อ

-ใช้ขา P3.2 เป็นสัญญาณ RS

-ใช้ขา P3.3 เป็นสัญญาณ R/W

-ใช้ขา P3.4 เป็นสัญญาณ EN (E)

แผนผังเวลาในการติดต่อกับ LCD Module มีแสดงดังรูปที่ 2.37



Example of flag check timing sequence

รูปที่ 2.32 แสดงแผนผังเวลาในการติดต่อกับ LCM

จากในรูปที่ 2.37 มีรายละเอียดของแต่ละสัญญาณดังนี้

1) RS : เนื่องจากในชิปไมโครคอนโทรเลอร์มีรีจิสเตอร์อยู่ 2 ประเภทคือ Instruction registe และ data register โดยรีจิสเตอร์ทั้ง 2 จะถูกเลือกโดยสัญญาณ RS ดังนี้

สัญญาณ RS = 0 หมายถึงเลือกใช้ Data Register

สัญญาณ RS = 1 หมายถึงเลือกใช้ Instruction Register

2) R/W (Read/Write) เป็นสัญญาณที่ใช้เลือกว่าจะทำการเขียนหรืออ่านข้อมูลจาก LCD Module โดย

สัญญาณ R/W = 0 หมายถึงต้องการอ่านข้อมูลจาก LCD Module

สัญญาณ R/W = 1 หมายถึงต้องการเขียนข้อมูลไปยัง LCD Module

3) E (Enable) มีรายละเอียดดังแสดงในตารางที่ 2.13

ตารางที่ 2.13 แสดงการทำงานของสัญญาณ E

RS	R/W	E	OPERATION
0	0		write instruction code
0	1		read busy flag and address counter
1	0		write data
1	1		read dara

จากแผนผังเวลา ในการตรวจสอบ Busy Flag และจากตารางจะเห็นว่าในการเขียนรหัสคำสั่ง (instruction code) ทุกครั้ง

RS และ RW ต้องมีค่าเป็น 0 และส่งข้อมูลไปในขณะที่สัญญาณ E เปลี่ยนจาก 1 เป็น 0 ในการเขียนข้อมูลทุกครั้ง

RS = 1 และ RW = 0 และส่งข้อมูลไปในขณะที่สัญญาณ E เปลี่ยนจาก 1 เป็น 0 ในการอ่าน busy flag และ address counter ทุกครั้ง

RS = 0 และ RW = 1 และรับข้อมูลเข้ามาในขณะที่สัญญาณ E เป็น 1ในการอ่านข้อมูลทุกครั้ง

RS = 1 และ RW = 1 และรับข้อมูลเข้ามาในขณะที่สัญญาณ E เป็น 1

จากแผนผังเวลา(ในรูปที่ 2.32) จะเห็นว่า DB0-DB7 จะมีสถานะเป็น high impedance เมื่อสัญญาณ E มีสถานะเป็น 0 ดังนั้นในการใช้งานจริง เมื่อเราติดต่อกับ LCD Module ควรจัดการส่งสัญญาณ E ให้มีค่าเป็น 0 เพื่อให้ P1.0-P1.7 ของ MCS-51 มีสถานะเป็น high impedance ด้วย ทั้งนี้เพื่อเราจะได้ใช้งาน P1.0-P1.7 อย่างอื่นได้ด้วย

บทที่ 3
สถาปัตยกรรมโครงงาน

จากบทที่แล้วเราได้ศึกษาเกี่ยวกับสถาปัตยกรรมของ 8051 และการสื่อสารด้วย RS-232 ไปแล้ว ในบทนี้เราจะมาทำการออกแบบการทำงานทางด้าน Hardware และ Software ซึ่งเป็นส่วนที่มีความสำคัญมาก ถ้าหากมีการออกแบบที่ดีแล้ว ก็จะทำให้โครงงานนั้นมีประสิทธิภาพ เนื่องจากมีการวางแผนที่เป็นระบบหากเมื่อเกิดปัญญาก็สามารถที่จะแก้ไขและตรวจเช็คได้ในที่สุด

3.1 การออกแบบทางด้าน Hardware

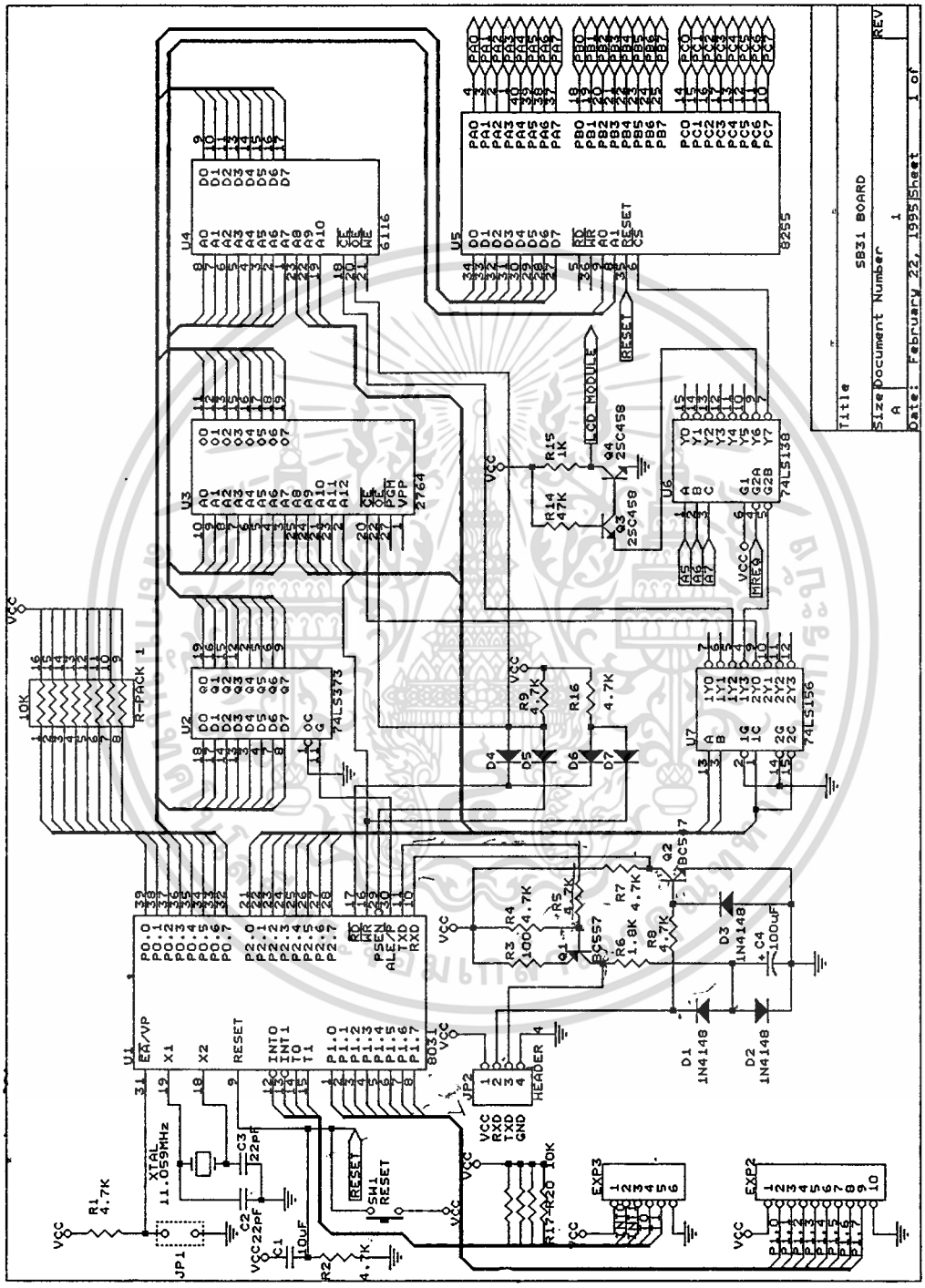
ในภาค Hardware นั้นแบ่งออกเป็น 2 ส่วนใหญ่ๆ คือ

1) Control Circuit Part (ส่วนของการคอนโทรลระบบ)

2) Data Communication for PC & μcontroller (การสื่อสารข้อมูลระหว่างเครื่องคอมพิวเตอร์และบอร์ดควบคุม)

3.1.1 Control Circuit Part (ส่วนควบคุมระบบ)

ในการดักค่าข้อมูลระหว่าง PC และ Hardlock นั้นเราใช้ไมโครคอนโทรลเลอร์ตระกูล 51 (MCS-51) เบอร์ 8051 เป็นตัวดักข้อมูล โดยใช้ Port ขนานของ IC # 8255 ทั้ง 3 port เป็นทางผ่านข้อมูล และใช้ Serial port เป็นพอร์ทป้อนข้อมูลกลับ computer แสดงดังรูปที่ 3.1



รูปที่ 3.1 วงจรภาคคอนโทรลโดยใช้ 8051 เป็นตัวควบคุม

3.1.2 Data Communication for PC & μcontroller

(การสื่อสารข้อมูลระหว่างเครื่องคอมพิวเตอร์และบอร์ดควบคุม)

ในการที่จะให้มีการติดต่อระหว่างเครื่องให้มีประสิทธิภาพนั้น จะต้องทำการออกแบบการสื่อสารให้เหมาะสมกับโครงงานนั้นๆ ด้วย ในโครงงานนี้ได้มีการออกแบบการสื่อสารดังนี้

1) MCS-51 การรับ-ส่งข้อมูลทางพอร์ทอนุกรมได้ออกแบบให้มีการทำงานในโหมด 1 โดยใน 1 ขบวนข้อมูลประกอบด้วย 1 Start bit , 8 Data bit , 1 Stop bit และจะใช้ข้อมูลในการส่ง 1 ครั้งต่อ 3 ขบวนข้อมูล ซึ่งจะได้กล่าวถึงรายละเอียดต่อไป เราใช้ความเร็วในการสื่อสารคือ 9600 BPS

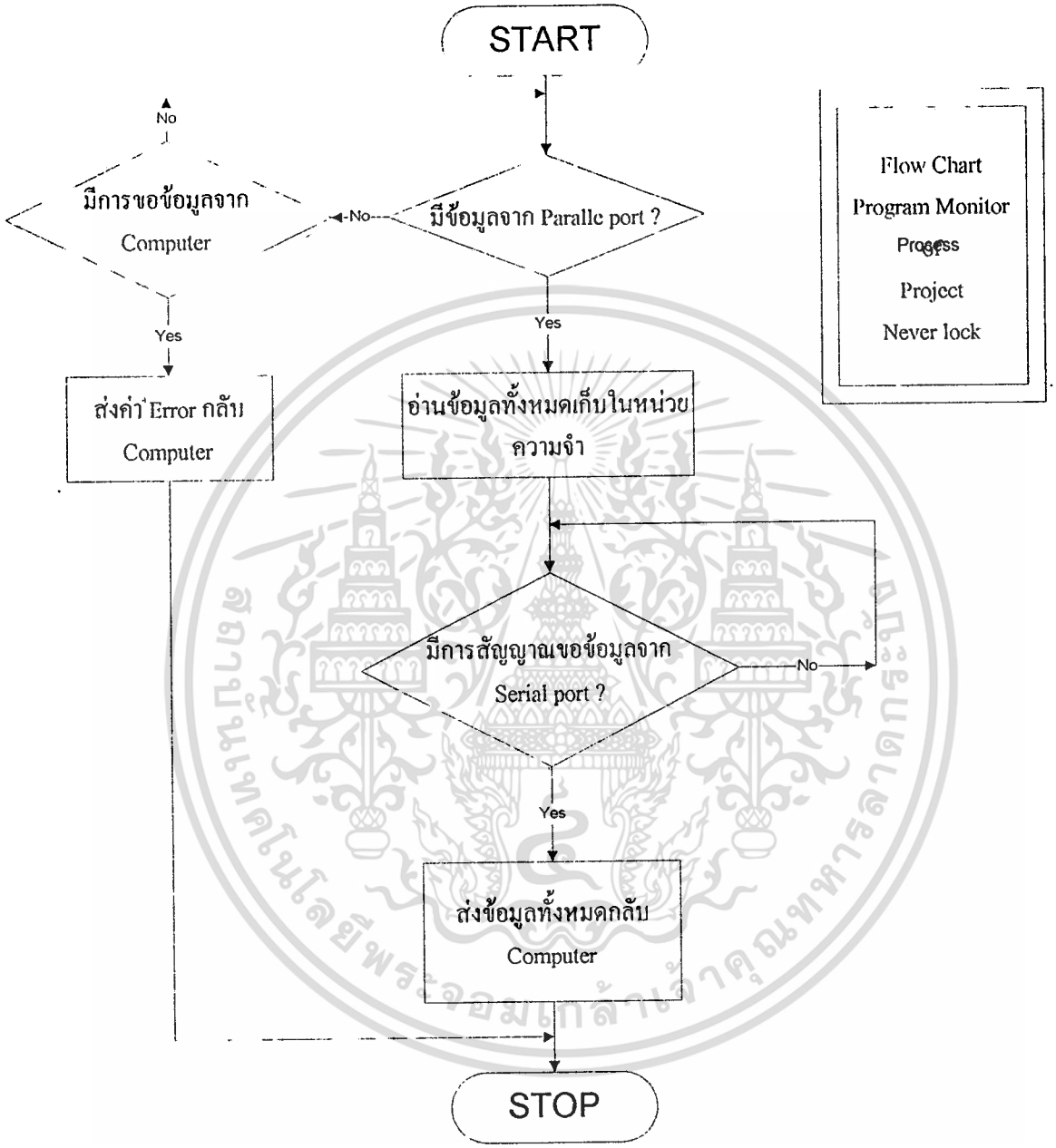
2) ลักษณะข้อมูลที่ใช้ในการสื่อสาร แบ่งออกเป็นข้อมูล 3 ขบวนดาต้า โดยทั้ง 3 ชุดนั้นเป็นค่าข้อมูลที่ดักได้ หากในการดักข้อมูลเกิดการ error ขึ้นจะส่งข้อมูลเป็น 1 ทุกบิท เพียง 1 ชุดเท่านั้น

3.2 <u>การออกแบบทางด้าน Software</u>

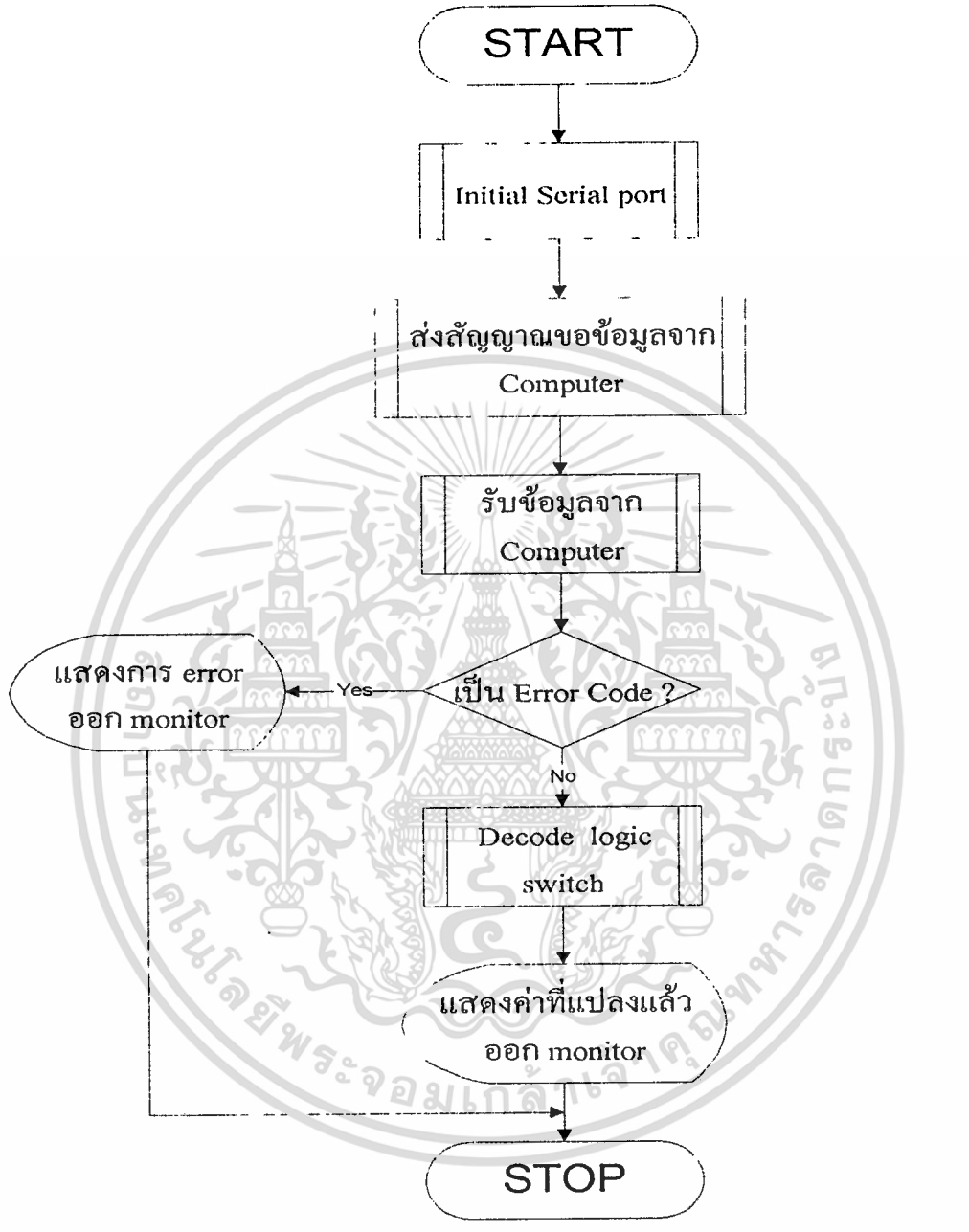
เป็นการออกแบบลักษณะการทำงานของ Program Monitor ที่ใช้ในการดักข้อมูล ดังรูปที่ 3.2

จากรูปที่ 3.2 จะเห็นว่าเมื่อเริ่มทำงานจะทำการวนตรวจสอบว่ามีการส่งข้อมูลผ่าน Paralle port หรือไม่ หากมีก็จะทำการเก็บข้อมูลเอาไว้ แล้วรอให้มีการขอข้อมูลจาก computer จึงจะส่งข้อมูลที่ดักได้กลับไป decode

ส่วนรูปที่ 3.3 เป็น Flow chart ของ Program decode

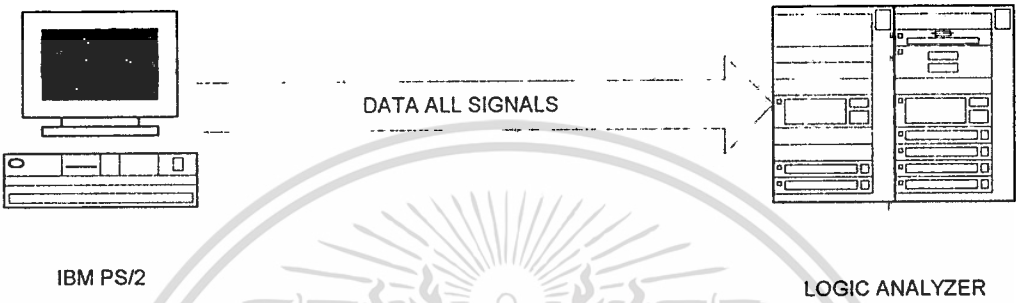


รูปที่ 3.2 แสดง Flow Chart ของ Program monitor



รูปที่ 3.3 แสดง Flow Chart ของ Program Decode

อีกวิธีหนึ่งที่สามารถทำได้คือ การใช้ logic analyzer จับสัญญาณทั้งหมดที่ใช้ในการติดต่อ แล้วค่อยนำสัญญาณเหล่านั้นมาทำสมการความสัมพันธ์อีกทีหนึ่ง



รูปที่ 3.4 แสดงการต่อ logic analyzer เข้ากับ Computer

บทที่ 4
สรุปผลการทดลอง

4.1 สรุปผลการทดลอง

จากการทดลองทำให้เห็นว่าฮาร์ดล็อกที่มีอยู่ในปัจจุบันนั้นมีทั้งที่ต่อกับ serial port และ paralle port แต่ลักษณะการส่งข้อมูลยังคงใช้ส่งข้อมูลแบบ serial

จากการที่คาดว่า Hardlock จะใช้ EEPROM หรือ EPROM ในการเก็บ Code ของโปรแกรมแต่ในไม่ได้ใช้เพียงอุปกรณ์ประเภท Program logic array แต่ยังคงมีการใช้ IC แบบใหม่ซึ่งไม่สามารถถอด code ของโปรแกรมได้เลย

4.2 ปัญหาที่เกิดขึ้น

จากการทดลองปัญหาที่เกิดขึ้นนั้นเกิดจากสิ่งที่คาดคิดไว้ทั้งหมด ไม่ตรงกับความเป็นจริง สามารถแจงออกเป็นข้อๆ ได้ดังนี้

1. เราคาดว่าข้อมูลที่ใช้ในการสื่อสารจะเป็นแบบขนานขนาด 1 byte ทั้งรับและส่ง แต่ในความเป็นจริงแล้วใช้การส่งแบบอนุกรม อีกทั้งจำนวนข้อมูลที่ใช้ในการส่งก็ไม่เท่ากันอีกด้วย
2. ขาสัญญาณที่ใช้ในการติดต่อนั้น Hardlock แต่ละตัวใช้ขาสัญญาณไม่ตรงกัน
3. เวลาการ sampling สัญญาณนั้นสุ่มไม่ทันกับ computer เนื่องจาก computer ในปัจจุบันมีขีดความเร็วมากกว่า μcontroller ทั้งนั้น

4.3 ข้อเสนอแนะ

จากการทดลองนี้ สามารถแยกการแกะ Hardlock ออกเป็น 2 แบบหลักๆ

4.3.1 แกะโดยหา Code ของ Hardlock ขึ้นมาแล้วสร้าง Hardlock จำลอง

4.3.2 แกะโดยแก้ที่ software คือการ patch code ไปที่โปรแกรมใช้งาน

แต่ไม่ว่าจะใช้วิธีไหนก็ตามต้องขึ้นอยู่กับความแน่นหน่าของทั้ง Hardlock และ software ว่าอย่าง protect มาดีกว่ากัน

เอกสารอ้างอิง

[1] PC Programmer's Guide to Low-Level Functions and Interrupts

[2] PC Intern System programming , ABACUS , Michael Tischer

[3] http://www.numega.com/WWW/numegal/si3.html

[4] Microprocessors and Interfaceing , Mc Graw Hill , Douglas V. Hall

[5] Designing With Programmable Array Logic , Monolithic Memories

[6] Borland C++ Handbook , Mc Graw Hill , Chris H. Pappas

[7] ยืน ภู่วรวรรณ , เทคโนโลยีฮาร์ดแวร์ IBM PC , ซีเอ็ดยูเคชั่น จำกัด

[8] ทินกร ดุ๊ก , การอินเตอร์เฟส IBM PC , ฟิสิกส์เซ็นเตอร์ การพิมพ์

กิตติกรรมประกาศ

ขอขอบพระคุณ อาจารย์ไพศาล สิทธิโยภาสกุล และอาจารย์บุญย์ชนะ ภู่ระหงษ์ ที่ได้ให้การประสิทธ์แระสาทวิชา ให้คำแนะนำปรึกษาในเรื่องต่างๆ แก่ผู้เขียน และขอบพระคุณ Staff Computer Lab ศูนย์วิจัยและบริการคอมพิวเตอร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ทุกท่านที่ได้ให้คำแนะนำในการทำโครงงานตั้งแต่ต้นตลอดมา

ภาคผนวก

```
// Source program scan code hardlock
#include <dos.h>
#include <conio.h>
#include <stdio.h>
void main()
{
        FILE *f;
        int in,out;
        clrscr();
        f=fopen("data.dat","w");
        for(out=0;out<0x100;out++) {
                outp(0x378,out);
            //  delay(10);
                in=inp(0x379);
                fprintf(f,"%d = %d\n",out,in);
        }
        fclose(f);
}
```

ตัวอย่างการแกะฮาร์ดล็อก

Hardlock ภาษาไทยของบริษัท R&D

จากการดักค่าข้อมูลพบว่า Software ตัวนี้ มีการตรวจสอบ Code ทั้งหมด 2000 กว่าครั้ง แล้วนำข้อมูลมาเข้ารหัสออกมาเป็นข้อมูล **32 bytes** โดยจะมีการตรวจสอบว่าข้อมูลแต่ละ byte เป็นอะไร โดยมีข้อมูลตามนี้

0F	**0F**	**0F**	**0F**	**0F**	**0F**	**0F**	**0F**
F0	**F0**	**F0**	**F0**	**F0**	**F0**	**F0**	**F0**
00	**00**	**00**	**00**	**F0**	**F0**	**F0**	**F0**
FF	**FF**	**FF**	**FF**	**FF**	**FF**	**FF**	**FF**

โดยข้อมูลทั้งหมดจะฝากไว้ที่หน่วยความจำตำแหน่งเริ่มต้นที่ **088Fh** แล้วมีการตรวจสอบผลรวมทั้งหมดอีกที่หนึ่ง ซึ่งผลรวมมีค่า **13B0h**