บทที่ 2
พื้นฐานของการสร้างเครื่องมือวัดอุณหภูมิและความเข้มแสง

2.1 ส่วนประกอบของเครื่องมือวัด [1]

เครื่องมือที่ใช้ในการวัดระบบต่าง ๆ เป็นอุปกรณ์ที่มีความจำเป็นในการวัดเพื่องานทดลองทางด้านวิทยาศาสตร์ โดยเฉพาะเครื่องมือประเภทที่สามารถนำผลจากวัดมาเปรียบเทียบและรายงานได้ อย่างเที่ยงตรงเครื่องมือประเภทนี้เป็นเครื่องมือวัดที่ประกอบด้วยส่วนต่าง ๆ คือ

2.2.1 ส่วนของการรับรู้สภาวะมากน้อยของระบบ เช่น เทอร์โมมิเตอร์วัดความร้อน โซล่าเซลล์วัดความเข้มแสง

2.1.2 ส่วนของการนำสัญญาณที่ได้จากการวัดให้มีขนาดพอดีที่จะนำไปส่งให้กับส่วนที่ใช้ประมวลผล

2.2.3 ส่วนของการประมวลผล เราจะนำไมโครโปรเซสเซอร์เข้ามาใช้ในการประมวลผลเนื่องจากจะทำให้อุปกรณ์มีความกะทัดรัดสามารถที่จะพกพาไปได้สะดวกในการใช้งาน การแสดงผลเราจะแสดงผลออกมาทางจอแสดงผล

เครื่องมือวัดจะประกอบด้วยองค์ประกอบสำคัญ 3 ส่วนคือ

1. Transducer ได้แก่ เทอร์โมมิเตอร์ โซล่าเซล ฯลฯ
2. หน่วยประมวลผลสัญญาณ (Signal processing Unit)

   ในส่วนนี้จะเป็นการตัดแต่งสัญญาณที่ได้รับจาก Sensor ให้มีขนาดพอดีที่จะส่งไปบันทึกค่าแล้วแสดงผลและยังตัดสัญญาณรบกวนชนิดอื่น ๆ ออกเพื่อไม่ให้มีการผิดเพี้ยนก่อนที่จะไปสู่ส่วนแสดงผลการประมวลผลสัญญาณเชิงตัวเลข (Digital Signal Processing Unit) ประกอบด้วย

   - Multiplex (MUX).
   - Amplifier (Amp)
   - Sample Hole (S/H)
   - Analog to digital Converter (A/D)

3. ชุดบันทึกค่าและแสดงผล
   - แบบอะนาลอก เป็นการแสดงผลโดยการสเกล หรือแสดงเป็นเส้นกราฟ

- แบบดิจิตอล เป็นการแสดงผลเป็นค่าตัวเลขหรือกราฟที่มี 2 ลักษณะเท่านั้น การบันทึกค่าและแสดงผลกระทำโดยระบบไมโครคอมพิวเตอร์

2.2 องค์ประกอบของไมโครคอมพิวเตอร์ [1]

CPU	:	หน่วยประมวลผลกลาง(Central Processing Unit)
Mem	:	หน่วยความจำ(Memory)
		หน่วยความจำที่ใช้เก็บโปรแกรม(ROM)
		หน่วยความจำทั่วไปที่ใช้เก็บข้อมูลไปประมวลผล(RAM)
In	:	อินพุต
Output	:	เอาท์พุต
I/O	:	อินพุต/เอาท์พุต

2.3 ระบบ Intelligent Instrumentation [2]

คือระบบที่มีการนำไมโครคอมพิวเตอร์มาประยุกต์ใช้โดยการเชื่อมโยงระบบไมโครคอมพิวเตอร์เข้ากับส่วนของการประมวลผลสัญญาณเชิงดิจิตอล แล้วนำไปประมวลผลภายในหน่วยความจำของระบบไมโครคอมพิวเตอร์ เพื่อให้ได้ข่าวสารเพียงพอที่จะแสดงผลให้มนุษย์เข้าใจได้

อินเตอร์เฟส ทำหน้าที่ป้องกันข้อมูลที่จะส่งไปเก็บไว้ในหน่วยความจำไม่ให้สูญหายในระหว่างดำเนินการอ่านข้อมูล และช่วยให้ทำงานโดยไม่เกิดการขัดจังหวะการทำงาน

2.3.1 Transducer : ใช้ในการแปลงค่าทางฟิสิกส์เป็นสัญญาณไฟฟ้าให้ได้ตามที่ระบบยอมรับ ซึ่งค่าทางฟิสิกส์อาจจะเป็นอุณหภูมิ ความดัน ความเร็ว หรือปริมาณทางไฟฟ้าเช่น แรงดันไฟฟ้า ความต้านทาน หรือ ความถี่ของไฟฟ้า

2.3.2 Muxtiplexer(MUX) : วงจรควบคุมสัญญาณ คือ วงจรที่เลือกสัญญาณหนึ่งสัญญาณใดในหลาย ๆ สัญญาณ เพื่อจะส่งไปใช้เพียงสัญญาณเดียว

2.3.3 Sample and Hole(S/H) : การสุ่มและการคงค่า เนื่องจากวงจรอะนาลอกทูดิจิตอลต้องการเวลาในการแปลงสัญญาณ อะนาลอก ให้เป็นสัญญาณดิจิตอลที่เหมาะสม ถ้าสัญญาณอะนาลอกมีการเปลี่ยนแปลงในช่วงเวลาที่มีการแปลงสัญญาณอาท์พุทของวงจรเปลี่ยนสัญญาณอะนาลอกทูดิจิตอล จะเกิดการผิดพลาด จึงต้องมีการป้องกันด้วยวงจรสุ่มค่าและคงค่าสัญญาณ แล้วเก็บไว้ในตัวเก็บประจุ ในระหว่างช่วงเวลาการเปลี่ยนแปลง หลังจากที่การเปลี่ยนแปลงสัญญาณแสร็จสิ้นลงจึงจับสัญญาณอะนาลอกค่าใหม่มาเป็นเช่นนี้ไปเรื่อย ๆ

2.3.4 อะนาลอกทูดิจิตอล คอนเวอร์เตอร์ : วงจรสำหรับเปลี่ยนสัญญาณอะนาลอกเป็นสัญญาณดิจิตอล ซึ่งสัญญาณ เอาท์พุตอาจแสดงผลเลข หรืออาจเก็บบันทึกในรูปสัญญาณแรงดันเลขก็ได้

2.3.5 Amplifier(Amp) : ส่วนขยายสัญญาณ จะเห็นได้ว่าสัญญาณที่ออกมาจากส่วนของ Transducer มักจะมีขนาดที่เล็กมากจึงต้องได้รับการขยายสัญญาณให้มีขนาดใหญ่ขึ้นเป็นพันเท่าก่อนที่จะถูกส่งต่อไปยังส่วนการวิเคราะห์หรือแสดงผลของอุปกรณ์

2.4 โครงสร้างของ MCS-51 [3-5]

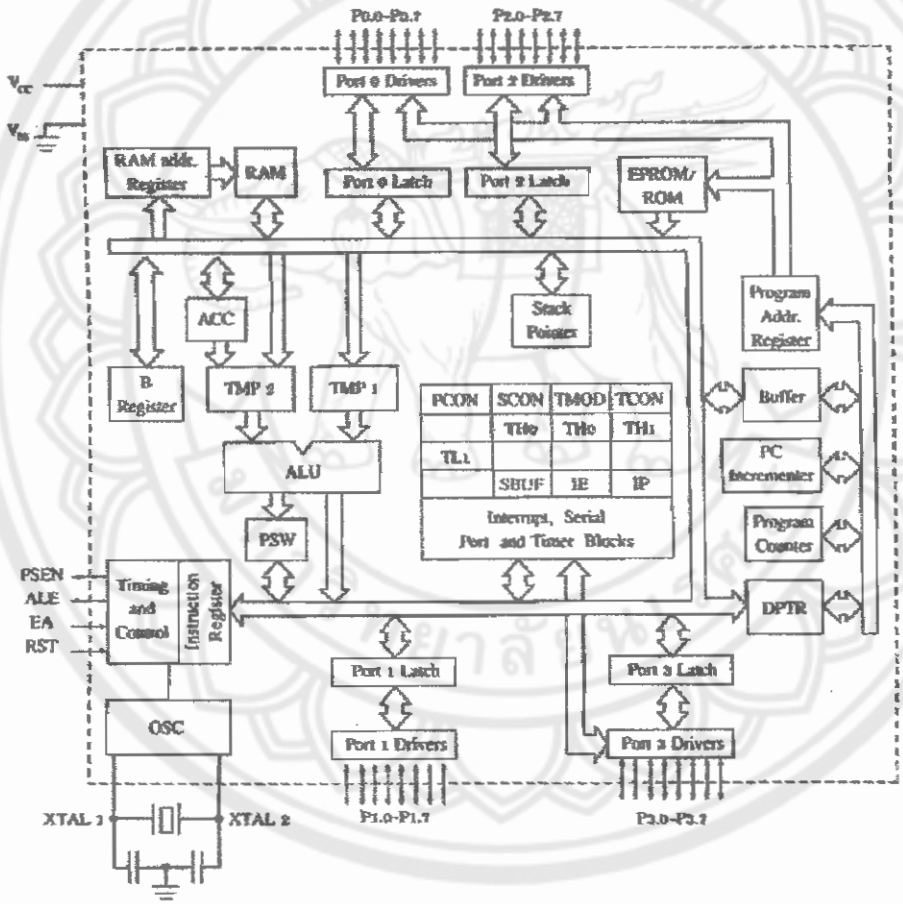
ไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีด้วยกันหลายเบอร์ขึ้นกับโครงสร้างภายในของมัน บางเบอร์จะมีหน่วยความจำภายในเป็นแบบ ROM บางเบอร์เป็นแบบ EPROM บางเบอร์มี RAM ภายใน 128 ไบต์ บางเบอร์มี 256 ไบต์ ซึ่งคุณสมบัติที่สำคัญของ MCS-51 มีดังต่อไปนี้

- มีหน่วยความจำ ROM 4 K byte
- มีหน่วยความจำ RAM 128 byte
- มีพอร์ท I/O ขนาด 8 บิต 4 พอร์ท
- มี Timer 16 บิต 2 ตัว
- สามารถอินเตอร์รัพท์ได้ 5 แหล่ง
- มีวงจรออสซิลเลเตอร์และวงจรนาฬิกาบนชิพ
- มีพอร์ทอนุกรมที่สามารถรับส่งข้อมูลแบบ Full Duplex ความเร็วสูง
- อ้างหน่วยความจำโปรแกรมภายนอกได้ 64K
- อ้างหน่วยความจำข้อมูลภายนอกได้ 64K
- สามารถประมวลผลทีละบิตได้
- สามารถอ้างหน่วยความจำแบบบิตได้ 210 ตำแหน่ง
- หนึ่งวัฏจักรคำสั่ง ใช้เวลาประมาณ 1 ไมโครวินาที ขณะทำงานด้วยสัญญาณนาฬิกา 12 MHz

ซึ่งตัวอย่างไมโครคอนโทรเลอร์ตระกูล MCS-51 และลักษณะต่าง ๆ สามารถแสดงได้ในตารางที่ 2.1

ตารางที่ 2.1 ไมโครโปรเซสเซอร์ตระกูล MCS-51 เบอร์ต่าง ๆ

เบอร์	หน่วยความจำโปรแกรมบนชิพ	หน่วยความจำข้อมูลบนชิพ	TIMERS
8051	4K ROM	128 byte	2
8031	-	128 byte	2
8751	4K EPROM	128 byte	2
8052	8K ROM	128 byte	3
8032	-	128 byte	3
8752	8K EPROM	128 byte	3



รูปที่ 2.1 โครงสร้างภายในของ MCS-51

ในส่วนของโครงสร้างภายนอกไอซี ไมโครคอนโทลเลอร์ 8051 โครงสร้างไอซี เป็นแบบ DIP มีขาทั้งหมด 40 ขา โดยขาต่าง ๆ จะใช้เป็นขาพอร์ทอินพุต เอาท์พุต ขาสัญญาณควบคุม ขาตำแหน่งหน่วยความจำ และขาข้อมูลดังรูปที่ 2.2

ความหมายของขาต่าง ๆ มีดังนี้

พอร์ท 0 (Port 0)

พอร์ท 0 ได้แก่ขาที่ 32-39 ของ MCS-51 สามารถใช้เป็นอินพุตเอาท์พุตได้นอกจากนี้ในการติดต่อกับหน่วยความจำภายนอกยังใช้เป็นขา แอดเดรสบัส และดาต้าบัสอีกด้วย

พอร์ท 1 (Port 1)

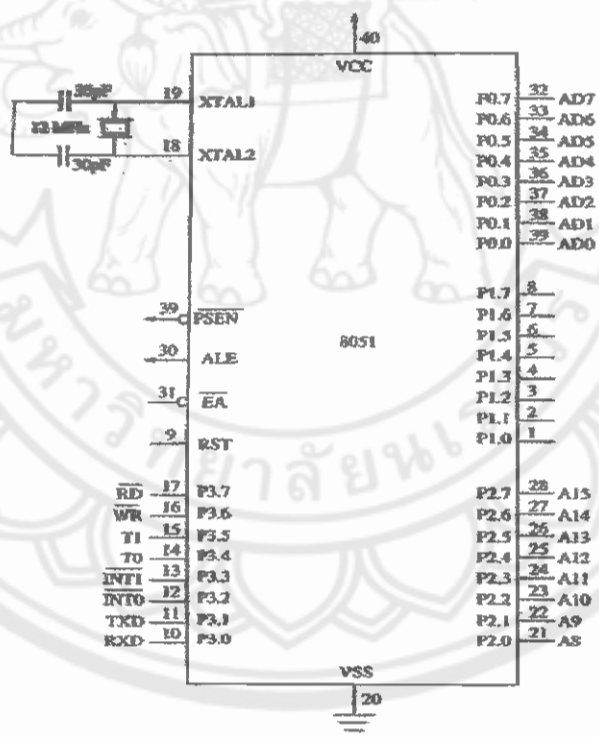
พอร์ท 1 ได้แก่ขาที่ 1-8 เป็นพอร์ท 8 บิต สามารถอ้างทีละบิตได้ คือ P1.0,P1.1,...,P1.7

พอร์ท 2 (Port 2)

พอร์ท 2 ได้แก่ขาที่ 21-28 จะใช้งาน 2 หน้าที่ คือใช้เป็นพอร์ท 8 บิต และใช้เป็นขาแอดเดรส 8 บิตในการอ้างหน่วยความจำภายนอก

พอร์ท 3 (Port 3)

พอร์ท 3 ได้แก่ขาที่ 10-17 จะใช้งานสองหน้าที่คือเป็นพอร์ทอินพุต และเอาท์พุต และใช้เป็นขาควบคุมต่าง ๆ ดังตารางที่ 2.2



รูปที่ 2.2 ขาต่าง ๆ ของ 8051

ตารางที่ 2.2 บิตและหน้าที่ต่าง ๆ ของพอร์ท 3

บิต	ชื่อ	หน้าที่พิเศษ
P3.0	RXD	ใช้รับข้อมูลทางพอร์ทอนุกรม
P3.1	TXD	ใช้ส่งข้อมูลทางพอร์ทอนุกรม
P3.2	INT0	อินเตอร์รัพท์ภายนอกหมายเลข 0
P3.3	INT1	อินเตอร์รัพท์ภายนอกหมายเลข 1
P3.4	T0	ตัวจับเวลา / ตัวนับ ตัวที่ 0
P3.5	T1	ตัวจับเวลา / ตัวนับ ตัวที่ 1
P3.6	WR	สัญญาณเขียนข้อมูลหน่วยความจำภายนอก
P3.7	RD	สัญญาณเขียนข้อมูลหน่วยความจำภายนอก

PSEN (Program Store Enable)

ขา PSEN เป็นขาที่ส่งสัญญาณออก คือขาที่ 29 ขานี้จะแอคทีฟเมื่อ MCS-51 ต้องการอ่าน Code โปรแกรมภายนอก โดยปกติถ้าหน่วยความจำภายนอกเป็น EPROM ขา PSEN จะต่อกับขา Output Enable ของ EROM

ALE (Address Latch Enable)

เนื่องจากพอร์ท 0 สามารถใช้เป็นขาอ้างตำแหน่ง และขาข้อมูล MCS-51 จะมีขา ALE ได้แก่ขา 30 ขานี้จะใช้ Multiplex สัญญาณ Address Bus ของพอร์ท 0 ในการใช้งานระบบ MCS-51 นั้น จะต้องมีอุปกรณ์มาต่อกับพอร์ท 0 ทำหน้าที่ Latch สัญญาณ Address Bus เมื่อ MCS-51 ต้องการติดต่อกับหน่วยความจำภายนอก MCS-51 จะส่งสัญญาณ Address Bus ออกมาทางพอร์ท 0 จากนั้นจะส่งสัญญาณ ALE มา Latch อุปกรณ์ภายนอก ให้เก็บค่า Address Bus ของพอร์ท 0 ไว้เพื่อใช้พอร์ท 0 เป็น Data Bus ต่อไป

EA (External Access)

ขา EA ได้แก่ขาที่ 31 ถ้าขานี้เป็นลอจิก "1" จะใช้กับเบอร์ 8051/8052 เพื่อบอกว่าให้อ่านโปรแกรมจากหน่วยความจำภายใน แต่ถ้าเป็นลอจิก "0" จะบอกว่าให้ MCS-51 ทำโปรแกรมโดยอ่านจากหน่วยความจำโปรแกรมภายนอก (ถ้าขา EA เป็น "0" ขา PSENจะแอคทีฟ) ถ้าหากเป็นเบอร์ 8031

หรือ 8032 ขา EA จะเป็น "0" เสมอ เพราะว่าไม่มีโปรแกรมหน่วยความจำภายใน แต่ถ้าใช้เบอร์ 8051/8052 ซึ่งมีหน่วยความจำโปรแกรมภายในและให้ขา EA เป็น "0" ซึ่งจะ Disabled ROM ภายใน และจะอ่านโปรแกรมจาก EPROM ภายนอกแทน

RST (Reset)

ขา RST ได้แก่ขา 9 จะใช้ในการรีเซต MCS-51 โดยจะให้ขานี้เป็นลอจิก "1" อย่างน้อย 2 แมชชีนไซเคิล (Machine Cycle)

ความถี่สัญญาณนาฬิกาบนชิพ (On-Chip Oscillator Inputs)

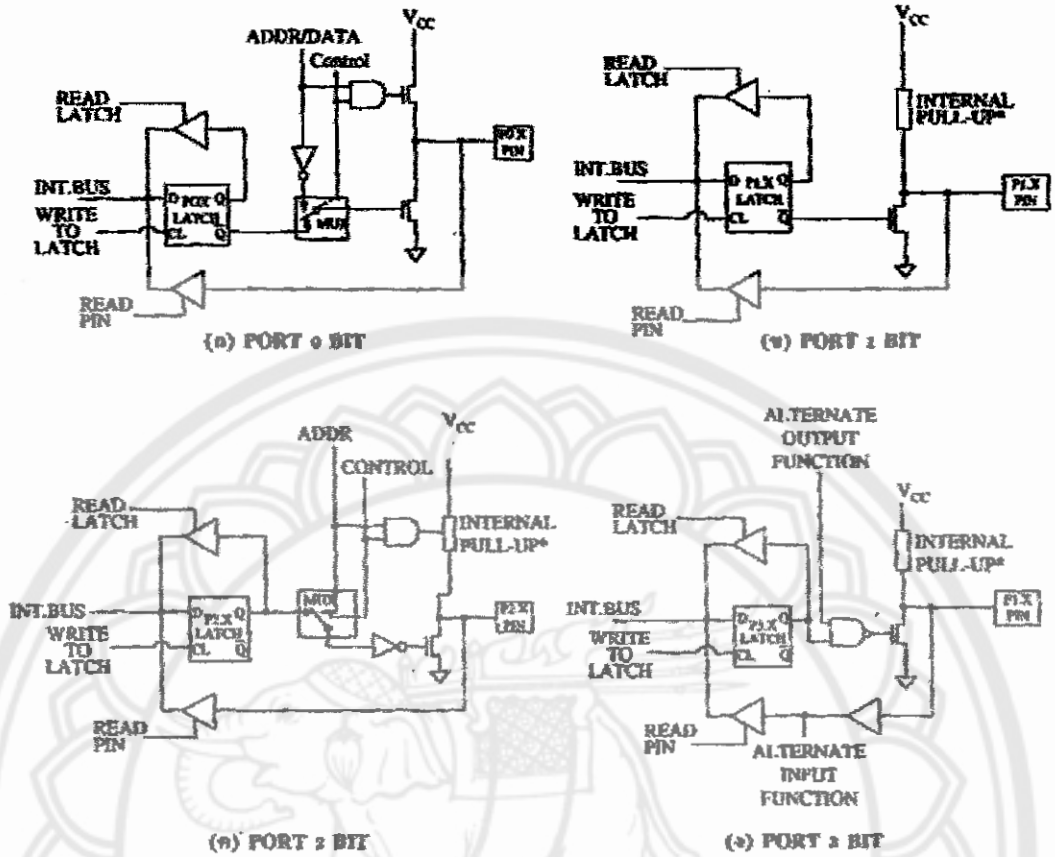
เป็นวงจรออสซิเลเตอร์ (Oscillator) บนชิพ ได้แก่ขา 18-19 โดยต่อคริสตอล (Crystal) เข้ากับขานี้โดยปกติมักจะใช้ Crystal ความถี่ 12 MHz กับตัวเก็บประจุหรืออาจใช้สัญญาณนาฬิกาจาก TTL Clock Source ต่อกับ XTAL1 และ XTAL2

Power Connections

ใน MCS-51 จะใช้แหล่งจ่ายไฟ 5 V ต่อเข้ากับขา Vcc (ขา 40) ส่วนขา Vss (ขา 20) จะต่อลงกราวน์

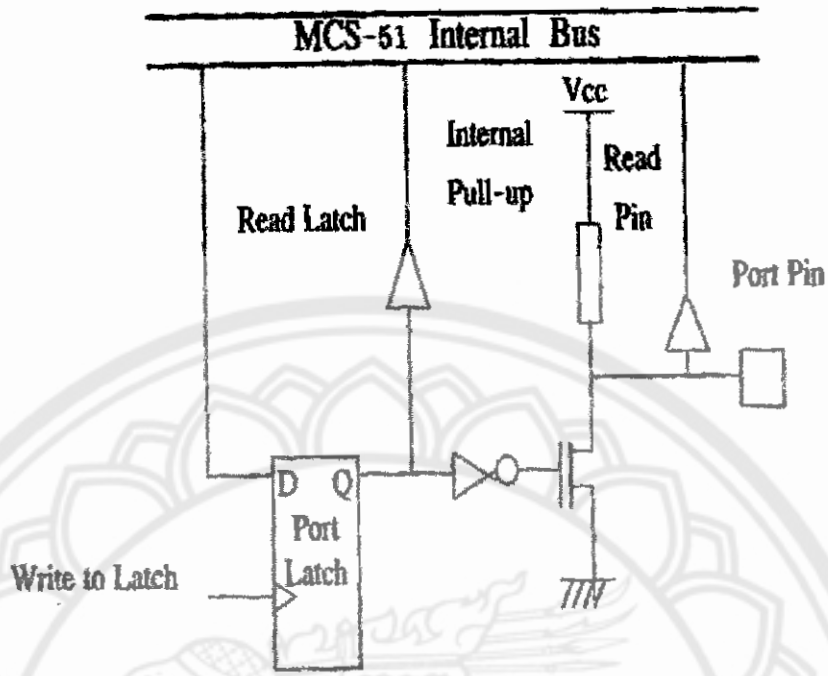
โครงสร้างของพอร์ทอินพุตเอาท์พุต (I/O Port Structure)

ขาของพอร์ทจะแสดงโครงสร้างภายในได้ ดังรูปที่ 2.3 โดยจะมีโครงสร้างเป็น Field-effect Transistor ต่ออยู่กับขาภายนอก และมีความต้านทานต่อพลูอัพอยู่ สำหรับพอร์ท 1,2,3 แต่ถ้าเป็นพอร์ท 0 จะไม่มีตัวต้านทานพลูอัพภายใน เพราะว่าต้องใช้เป็นขา Address Bus และ Data Bus



รูปที่ 2.3 โครงสร้างพอร์ททั้ง 4 ของ MCS-51

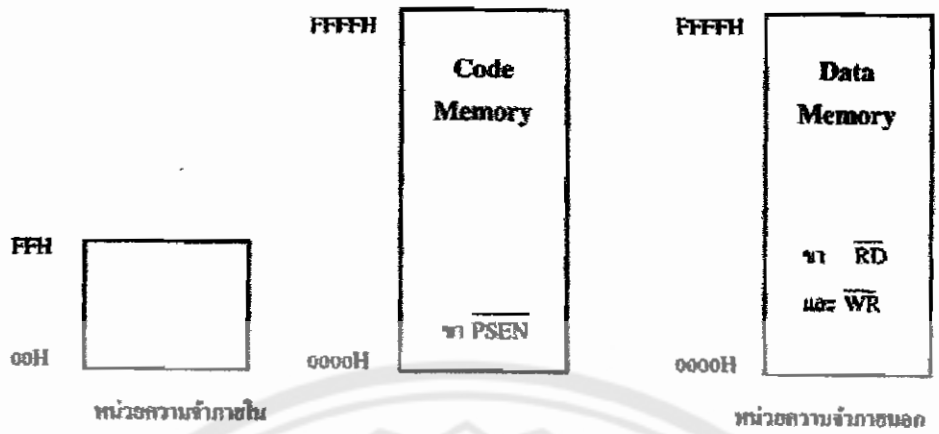
พอร์ทนี้สามารถใช้เป็นอินพุตเอาท์พุตกับอุปกรณ์ภายนอกได้ ในการอ่านข้อมูลจากพอร์ทจะอ่านสองแบบคือ Read Latch และ Read Pin โดย Read Latch หมายถึงการอ่านข้อมูลที่ถูก Latch เอาไว้เข้าสู่บัสภายในของ MCS-51 เช่นการทำคำสั่ง CPL P1.5 แต่ถ้าเป็นการ Read Pin จะเป็นการใช้พอร์ทเป็นอินพุต โดยการอ่านจากขาของไอซีเข้าสู่บัสภายในโดยการอ่านแบบ Read Latch และ Read Pin จะมีสัญญาณมาควบคุมที่บัฟเฟอร์ดังรูปที่ 2.4



รูปที่ 2.4 การต่อพอร์ทเข้ากับระบบบัสของ MCS-51

โครงสร้างหน่วยความจำ

หน่วยความจำสำหรับ MCS-51 จะมี 2 ชนิดคือ หน่วยความจำที่ใช้เก็บโปรแกรม (ROM) กับหน่วยความจำที่ใช้เก็บข้อมูลในการประมวลผล (RAM) MCS-51 บางเบอร์เช่น 8051 8052 จะมีหน่วยความจำโปรแกรมภายในชิพ และ MCS-51 ทุกเบอร์สามารถอ้างหน่วยความจำโปรแกรมภายนอกได้มากที่สุด 64 K และอ้างหน่วยความจำข้อมูลภายนอกได้มากที่สุด 64 K สำหรับหน่วยความจำ RAM ภายใน จะประกอบไปด้วยพื้นที่ใช้งานทั่วไป รีจิสเตอร์แบงค์ พื้นที่ใช้งานระดับบิต และรีจิสเตอร์ฟังก์ชันพิเศษ เราอาจเขียน ไดอะแกรมของหน่วยความจำของ 8031 ได้ดังรูปที่ 2.5 โดยในรูปจะบอกด้วยว่าขาใดจะแอดทีฟ



รูปที่ 2.5 การจัดหน่วยความจำของ MCS-51

ใน 8031 จะมีหน่วยความจำภายในตั้งแต่ตำแหน่ง 00H ถึง FFH และสามารถอ้างหน่วยความจำโปรแกรมภายนอกได้ 64 K ตำแหน่ง ถ้าอ่านข้อมูลจากหน่วยความจำโปรแกรมขา PSEN จะแอคทีฟ นอกจากนี้ 8031 สามารถอ้างหน่วยความจำข้อมูลภายนอกได้ 64 K ตำแหน่ง โดยการติดต่อกับหน่วยความจำนี้ ขา RD และ WR จะแอคทีฟ สำหรับหน่วยความจำข้อมูลภายในนั้นจะแบ่งออกได้ดังนี้

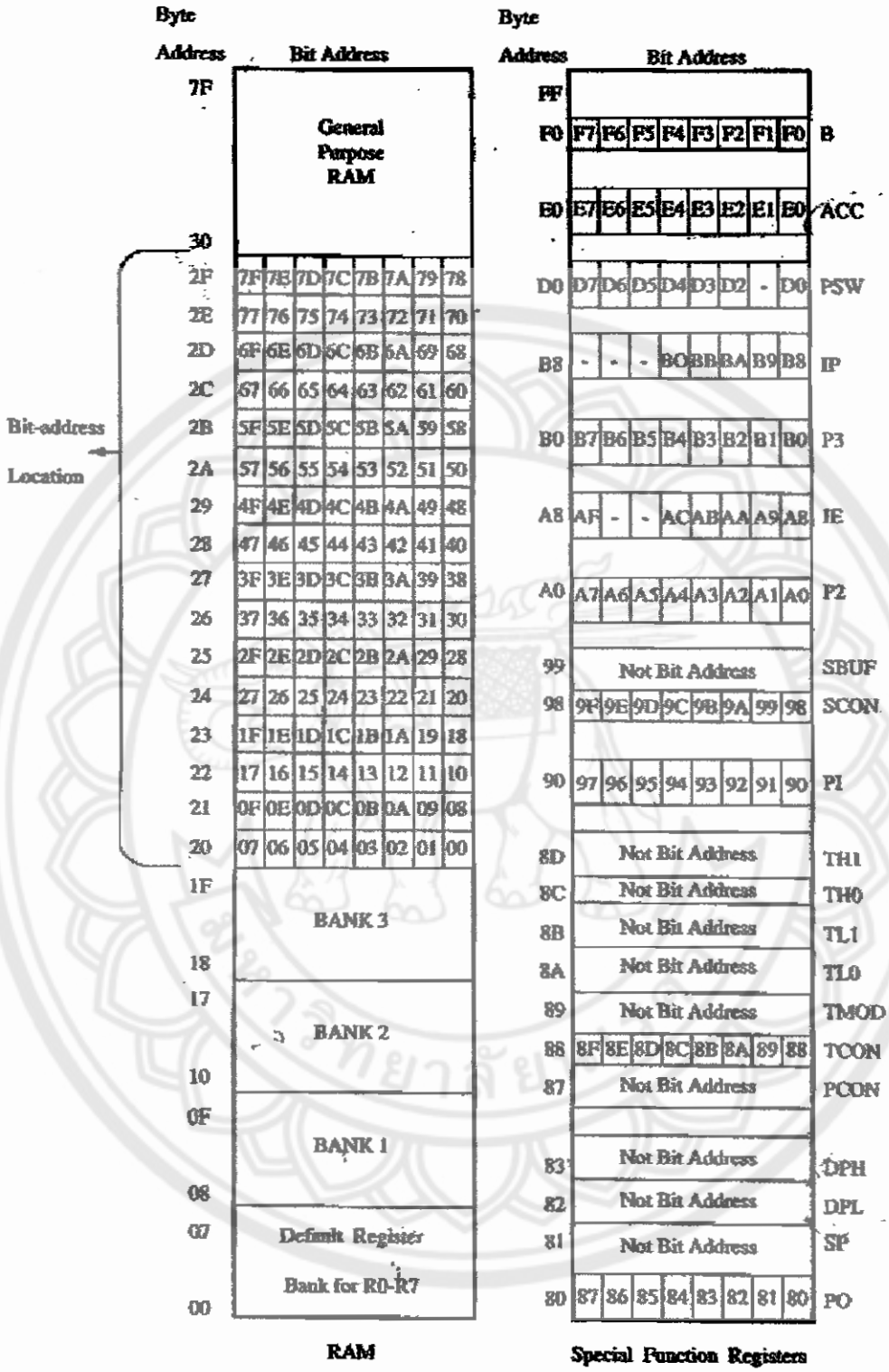
1. ชุดรีจีสเตอร์ 4 ชุด แต่ละชุดเรียกว่า รีจีสเตอร์แบงค์ ที่ตำแหน่ง 00H ถึง 1FH โดยแต่ละชุดประกอบด้วยรีจีสเตอร์ R0 ถึง R7
2. หน่วยความจำที่สามารถเข้าถึงข้อมูลระดับบิตได้ ตำแหน่ง 20H ถึง 2FH
3. หน่วยความจำใช้งานทั่วไปตำแหน่ง 30H ถึง 7FH
4. รีจีสเตอร์ฟังก์ชันพิเศษ ตำแหน่ง 80H ถึง FFH

แผนผังการจัดหน่วยความจำข้อมูลภายในแสดงได้ดังรูปที่ 2.5 จากแผนผังจะเห็นว่าการอ้างตำแหน่งหน่วยความจำภายในจะอ้างได้สองแบบ คือ การอ้างไปที่ตำแหน่งของไบต์ (เขียนหมายเลขตำแหน่งด้านนอก) หรือการอ้างไปที่ตำแหน่งของบิต โดยตำแหน่งของหน่วยความจำที่อ้างเป็นแบบบิตที่แน่นอน

หน่วยความจำใช้งานทั่วไป

จากรูปที่ 2.6 จะเห็นว่าใน 8031 จะมีหน่วยความจำ RAM สำหรับงานทั่วไปจำนวน 80 ไบต์ ตั้งแต่ตำแหน่ง 30H ถึง 7FH ตำแหน่งนี้สามารถอ้างตำแหน่งแบบ Direct Addressing Mode หรือ Indirect Addressing Mode ได้ ตัวอย่างเช่น ถ้าต้องการอ่านข้อมูลที่อยู่ในตำแหน่ง 5FH มาเก็บในรีจีสเตอร์ A สามารถเขียนคำสั่งได้เป็น

```
MOV     A,5FH
```

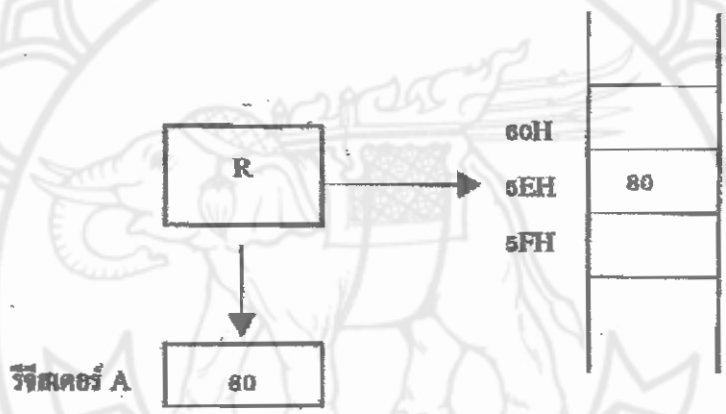


รูปที่ 2.6 ตำแหน่งของหน่วยความจำทั้งแบบไบต์และบิต

การย้ายข้อมูลแบบนี้เป็นการย้ายข้อมูลจากตำแหน่งที่เก็บโดยตรง (ตำแหน่ง 5FH) เรียกว่าการอ้างตำแหน่งแบบ Direct Addressing Mode นอกจากนี้ยังสามารถอ่านข้อมูลโดยใช้รีจีสเตอร์ R0 หรือ R1 การตัวชี้ตำแหน่งได้เรียกว่า การอ้างตำแหน่งแบบ Indirect Addressing Mode ตัวอย่างเช่น

```
MOV   R0,#5FH
MOV   A,@R0
```

การเขียนโปรแกรมด้านบนหมายความว่า เก็บค่า 5FH ไว้ใน R0 จากนั้นอ่านค่าที่ R0 ชี้อยู่คือตำแหน่ง 5FH มาเก็บไว้ที่รีจีสเตอร์ A ถ้าในตำแหน่ง 5FH มี 80 จะถูกเก็บใน A



รูปที่ 2.7 ขั้นตอนในการอ่านข้อมูล

Bit-Addressable RAM

ใน MCS-51 จะมีหน่วยความจำที่สามารถอ้างข้อมูลในระดับบิตได้ตั้งแต่ตำแหน่ง 20H ถึง 2FH รวม 16 ไบต์ โดยสามารถ SET,CLEAR,AND,OR ทางลอจิกได้ จำนวนบิตที่ใช้งานได้ทั้งหมดมีจำนวน 128 บิต (8บิต x 16 ไบต์) ถ้าต้องการเซตบิตตำแหน่งที่ 67H สามารถเขียนคำสั่งได้ดังนี้

```
SETB   67H
```

Register Banks

หน่วยความจำข้อมูลภายในที่ใช้เป็นชุดรีจีสเตอร์มีทั้งหมด 32 ตำแหน่ง โดยจะมี 4 ชุด แต่ละชุดมีรีจีสเตอร์ 8 ตัว คือ R0 ถึง R7 โดยชุดแรกจะอยู่ในตำแหน่ง 00H-07H ถ้าหากจะอ่านค่าจากตำแหน่ง 05H มาเก็บไว้ในรีจีสเตอร์ A จะเขียนโปรแกรมได้ดังนี้

```
MOV   A,R5
```

การอ้างตำแหน่งจะใช้แบบ Register Addressing ซึ่งขนาดของรหัสคำสั่งจะมีขนาด 1 ไบต์ แต่ถ้าเขียนคำสั่งเป็น MOV A,05H ผลที่ได้จะเหมือนกันแต่การเขียนแบบนี้ถ้าแปลงเป็นรหัสคำสั่งจะมีขนาด 2 ไบต์ ซึ่งจะทำให้โปรแกรมยาวกว่าแบบแรก ในการติดต่อกับ Register Bank นั้น เราสามารถเลือกให้ Bank ใด ๆ แอคทีฟได้โดยเขียนข้อมูลไปที่ Program Status Word ซึ่งอยู่ในส่วนของรีจีสเตอร์ฟังก์ชันพิเศษ เช่น ถ้าโปรแกรมให้ Bank 3 แอคทีฟ จะย้ายข้อมูลจากรีจีสเตอร์ A ไปที่ตำแหน่ง 18H ได้ดังนี้

```
MOV   R0,A
```

ถ้าไม่มีการเลือก Bank จะเป็นการติดต่อกับรีจีสเตอร์ Bank แรกเสมอ

รีจีสเตอร์ฟังก์ชันพิเศษ (Special Function Register)

ใน MCS-51 รีจีสเตอร์จะใช้หน่วยความจำ RAM ภายในชิพ โดยส่วนหนึ่งเป็น รีจีสเตอร์พิเศษ (Special Function Register : SFR) ซึ่งมีทั้งหมด 21 ตัว โดยรีจีสเตอร์พิเศษต่าง ๆ จะเริ่มที่หน่วยความจำตั้งแต่ 80H ถึง FFH ซึ่งมีทั้งหมด 128 ตำแหน่ง แต่จะเป็นรีจีสเตอร์ฟังก์ชันพิเศษเพียง 21 ตำแหน่ง แต่ถ้าเป็น 8032 / 8051 จะใช้ 26 ตำแหน่ง หรือมี SFR 26ตัว

จากรูปที่ 2.6 จะแสดงตำแหน่งหน่วยความจำของรีจีสเตอร์ บางตัวสามารถเข้าถึงข้อมูลแบบบิตได้อีกด้วย เช่น ถ้าเขียนโปรแกรม เป็น

```
SETB   0E0H
```

จะเป็นการเซตบิต 0 ของแอกคิวมูเลเตอร์ เนื่องจากตำแหน่ง E0H เป็นตำแหน่งของรีจีสเตอร์ A และเป็นบิตแอดเดรส บิตแรกของรีจีสเตอร์ A ด้วยคำสั่ง SETB (Set Bit) จะมีผลต่อบิตเท่านั้น จะไม่มีผลต่อไบต์ ถ้าหากต้องการติดต่อกับพอร์ท 1 ซึ่งไบต์ของพอร์ท 1 อยู่ที่ตำแหน่ง 90H แต่ตำแหน่งของระดับบิตจะอยู่ที่ตำแหน่ง 90H ถึง 97H รีจีสเตอร์ในกลุ่ม Special Function Register มีดังนี้

Program Status Word (PWS)

รีจีสเตอร์ตัวนี้เรียกย่อ ๆ ว่า PSW จะอยู่ที่ตำแหน่ง D0H ซึ่งสามารถเข้าถึงข้อมูลระดับบิตได้ โดยรีจีสเตอร์นี้จะเป็นตัวบอกสภาวะต่าง ๆ ของไมโครคอนโทรลเลอร์ ความหมายของแต่ละบิตแสดงได้ดังตารางที่ 2.3

ตารางที่ 2.3 ตำแหน่งบิตและหน้าที่ต่าง ๆ ใน PSW

บิต	ชื่อบิต	ตำแหน่ง	ความหมาย
PSW.7	CY	D7H	Carry Flag
PSW.6	AC	D6H	Auxiliary Carry Flag
PSW.5	F0	D5H	Flag 0
PSW.4	RS1	D4H	บิตสำหรับเลือก Register Bank 1
PSW.3	RS0	D3H	บิตสำหรับเลือก Register Bank 0
PSW.2	OV	D2H	Overflow Flag
PSW.1	-	D1H	Reserved
PSW.0	P	D0H	Even Parity Flag

1. แฟลกตัวทด (CY) บิตนี้เป็นบิตที่ 7 ของ PSW บิตนี้จะมีความสำคัญหากมีการกระทำทางคณิตศาสตร์ โดยบิตนี้จะเซต เมื่อเกิดการทดของบิตที่ 7 ขณะทำการบวกเลข หรือเซตเมื่อเกิดการยืมของบิตที่ 7 เมื่อเกิดการลบเลข
2. แฟลกตัวช่วยทด (AC) เมื่อมีการบวกแบบ BCD บิต AC จะถูกเซต เมื่อมีการทดจากบิตที่ 3 ไปบิตที่ 4 หรือถ้าใน Lower Nibble มีค่าระหว่าง 0AH-0FH เนื่องจากรหัส BCD นี้มีค่าได้มากที่สุดแค่ 9 หากมีการบวกเลขแบบ BCD จะต้องตามด้วยคำสั่ง DAA เพื่อปรับค่าที่มีค่าเกิน 9 โดยบวกเลข 6 เข้าไปจะทำให้เป็นรหัส BCD ที่แทนเลขฐานสิบได้
3. แฟลกศูนย์ (F0) เป็นแฟลกที่ผู้ใช้สามารถใช้งานทั่วไปได้
4. **บิตเลือกรีจีสเตอร์แบงก์ (RS) เป็นตัวเลือกว่าจะให้รีจีสเตอร์ตัวใดแอคทีฟ โดยกำหนดได้ในบิต RS1 และ RS2 ของ PSW และจะ Clear ตัวเองเมื่อระบบถูกรีเซต**
5. **แฟลกโอเวอร์โฟลว์(OV) แฟลกโอเวอร์โฟลว์จะถูกเซตหลังเกิดการกระทำทางคณิตศาสตร์แล้วเกิดโอเวอร์โฟลว์ คือจำนวนที่เกิดจากการบวกหรือการลบมีค่าเกินกว่าที่จำนวนนี้จะเกิดการเซตบิต OV ขึ้นใน PSW**

Hex : 0F Dec : 15

+7F +127

8E 142

1. บิตพาริตี้ (P) เป็นบิตที่บอกค่าพาริตี้ ของรีจีสเตอร์ Accumulator ซึ่งอาจเป็นตัวตรวจสอบความถูกต้องของข้อมูลได้ โดยจะเซตหรือเคลียร์ ขึ้นกับผลที่เกิดขึ้นกับแอคคิวมูเลเตอร์ เช่น ถ้าแอคคิวมูเลเตอร์มีค่าเป็น 10101101B บิต P จะเป็น "1"

รีจีสเตอร์ B (B Register)

รีจีสเตอร์ B จะอยู่ตำแหน่ง F0H ของหน่วยความจำข้อมูลภายใน โดยทั่วไปรีจีสเตอร์นี้จะใช้คูณหรือหารกับรีจีสเตอร์แอคคิวมูเลเตอร์ โดยการคูณจะใช้คำสั่ง MUL AB แล้วผลลัพธ์ 8 บิตล่างจะถูกเก็บไว้ที่รีจีสเตอร์ A ส่วน 8 บิตบน จะถูกเก็บที่รีจีสเตอร์ B การหารจะใช้คำสั่ง DIV AB โดยค่าใน A จะถูกหารด้วย B รีจีสเตอร์ B นี้ สามารถเข้าถึงข้อมูลระดับบิตได้ โดยอ้างอิงตำแหน่ง F0H ถึง F7H

ตัวชี้สแตก (Stack Pointer)

เป็นรีจีสเตอร์ขนาด 8 บิต อยู่ที่ตำแหน่ง 81H การเขียนค่าเข้าไปในตำแหน่งที่ SP ชี้อยู่นี้ เรียกว่า "Pushing" สำหรับการอ่านนั้นเรียกว่า "Popping" ค่าของ SP จะเพิ่มขึ้นหนึ่งก่อนที่จะเขียนข้อมูลลงไป และจะลดลงหนึ่งเมื่ออ่านข้อมูลออกมาแล้ว หากโปรแกรมทำคำสั่ง CALL จะใช้รีจีสเตอร์สแตคนี้เก็บค่าตำแหน่งเดิมของโปรแกรม ก่อนที่จะทำโปรแกรมย่อยเมื่อทำโปรแกรมย่อยเสร็จแล้วจะคืนค่าในสแตคให้กับ PC ตามเดิม โดยปกติค่า PC จะกำหนดให้อยู่ใน RAM ภายใน ถ้าต้องการให้มี SP เริ่มที่ตำแหน่ง 60H จะต้องเขียนคำสั่งดังนี้

```
MOV   SP,#5FH
```

การเขียนคำสั่งข้างบนถ้าใช้กับเบอร์ 8031 / 8051 จะเก็บค่าสแตคได้ 32 ไบต์ เพราะหน่วยความจำของ RAM ภายในจะสิ้นสุดที่ 7FH แต่เรากำหนดให้ SP มีค่าเท่ากับ 5FH ซึ่งจะเริ่มใช้งานที่ตำแหน่ง 60H ถ้าหาก MCS-51 ถูกรีเซต ค่า SP จะถูกกำหนดเองเป็น 07H ซึ่งจะเห็นได้ว่าทับกับ Register Bank 1 ถ้าหากงานที่ออกแบบมากขึ้น จะต้องใช้ Register Bank 2 ด้วยควรกำหนดค่า SP เสียก่อน

รีจีสเตอร์ Data Pointer (DPTR)

รีจีสเตอร์นี้ใช้สำหรับชี้ตำแหน่งรหัสโปรแกรมหรือข้อมูลในหน่วยความจำ โดยเป็นรีจีสเตอร์ขนาด 16 บิต ซึ่งประกอบด้วยรีจีสเตอร์ 2 ตัว คือ DPL ทำหน้าที่เก็บ 8 บิตต่ำ และ DPH ทำหน้าที่เก็บ 8

บิตสูง รีจีสเตอร์ทั้งสองตัวนี้จะรวมกันกลายเป็นรีจีสเตอร์ 16 บิต ถ้าหากต้องการเก็บค่า 55H ไปยังตำแหน่งของหน่วยความจำข้อมูลภายนอกตำแหน่งที่ 1000H จะเขียนโปรแกรมได้ดังนี้

```
MOV   A,#55H
MOV   DPTR,1000H
MOVX  @DPTR,A
```

ในบรรทัดแรกจะเป็นการอ้างตำแหน่งแบบ Immediate Addressing ซึ่งจะเก็บค่า 55H ลงในรีจีสเตอร์ A ต่อมาจะเก็บค่า 1000H ลงในรีจีสเตอร์ 16 บิต DPTR เพื่อชี้ไปที่ตำแหน่งหน่วยความจำ บรรทัดที่ 3 จะเป็นการอ้างตำแหน่งแบบ Indirect Addressing ซึ่งจะเก็บค่าใน A คือ 55H ลงในตำแหน่งที่ DPTR ชี้อยู่คือ ตำแหน่ง 1000H

รีจีสเตอร์พอร์ท (Port Registers)

ใน MCS-51 ค่าของพอร์ทจะหมายถึงค่าของหน่วยความจำด้วย หากต้องการส่งข้อมูลออกไปที่พอร์ท ก็เพียงแต่เขียน ข้อมูลไปที่หน่วยความจำที่พอร์ทนั้นอยู่ ใน MCS-51 พอร์ท 0 จะอยู่ที่ตำแหน่ง 80H พอร์ท 1 จะอยู่ที่ตำแหน่ง 90H พอร์ท 2 จะอยู่ที่ตำแหน่ง A0H และ พอร์ท 3 จะอยู่ที่ตำแหน่ง B0H พอร์ท 0,2 และ 3 โดยทั่วไปจะไม่ใช้ถ้าหากมีการติดต่อกับหน่วยความจำภายนอก หรือใช้เป็น พอร์ทพิเศษ โดยปกติแล้วจะใช้พอร์ท 1 ในการติดต่อกับอุปกรณ์ภายนอก พอร์ททุกพอร์ทสามารถอ้างข้อมูลในระดับบิตได้ ตัวอย่างเช่น ถ้าพอร์ท 1 บิต 7 ต่อกับหลอดไฟ การเปิดปิดหลอดไฟทำได้โดยการเซต หรือ Clear บิต 7 ของพอร์ท 1 นี้

การติดต่อกับพอร์ทในระดับนี้ อาจใช้คำสั่งในการอ้างข้อมูลระดับบิตได้ เช่น บิต 7 ของพอร์ทตรงกับตำแหน่งระดับบิตคือ ตำแหน่ง 97H อาจเขียนคำสั่งได้ดังนี้

```
CLR   97H
```

รีจีสเตอร์เวลา (Timer Registers)

ใน MCS-51 เบอร์ 8051จะมีรีจีสเตอร์ที่ใช้นับและจับเวลาขนาด 16 บิต 2 ตัว คือ Timer 0 อยู่ที่ตำแหน่ง 8AH และ 8CH โดยตำแหน่ง 8AH หมายถึง TL0 ซึ่งจะเป็น 8 ไบต์ และ 8CH หมายถึง 8 ไบต์สูง TH0 รีจีสเตอร์ อีกตัวคือ Timer 1 โดยแบ่งเป็น TL1 อยู่ที่ตำแหน่ง 8BH เป็นไบต์ต่ำ และ TH1 อยู่ที่ตำแหน่ง 8DH เป็นไบต์สูง การใช้ Timer จะต้องกำหนดการทำงานในรีจีสเตอร์ TMOD (Timer / Counter Mode Control Register) ซึ่งอยู่ที่ตำแหน่ง 88H เสียก่อน

รีจีสเตอร์พอร์ทอนุกรม (Serial Port Registers)

MCS-51 จะมีพอร์ทสื่อสารอนุกรม อยู่ภายในชิพ ซึ่งสามารถจะรับหรือส่งข้อมูลได้โดยติดต่อผ่านรีจีสเตอร์ SBUF ซึ่งอยู่ที่ตำแหน่ง 99H โดยถ้าต้องการส่งข้อมูลแบบอนุกรมให้เขียนข้อมูลไปที่รีจีสเตอร์นี้ ตัวพอร์ทสื่อสารอนุกรม สามารถโปรแกรมให้ทำงานได้ 4 โหมด โดยโปรแกรมผ่านรีจีสเตอร์ SCON ตำแหน่ง B8H

รีจีสเตอร์อินเตอร์รัพท์ (Interrupt Port Registers)

MCS-51 สามารถอินเตอร์รัพท์ได้ 5 ตำแหน่ง โดยมี 2-Priority ตัวอินเตอร์รัพท์นี้จะถูก Disable หลังจากระบบถูกรีเซต และจะ Enabled หลังจากเราเขียนข้อมูลไปที่รีจีสเตอร์ IE หรือตำแหน่ง A8H ลำดับความสำคัญสามารถเซตได้ที่รีจีสเตอร์ IP หรือตำแหน่ง B8H

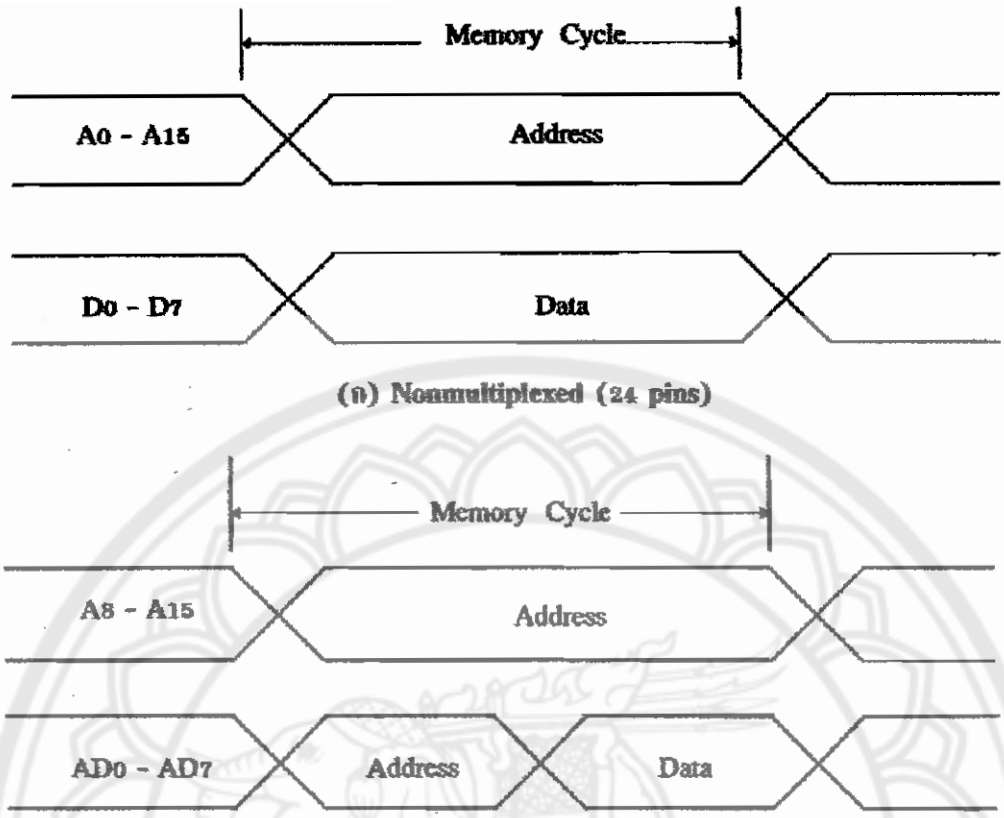
Power Control Register (PCON)

รีจีสเตอร์ PCON อยู่ที่ตำแหน่ง 87H ใช้หยุดการทำงานของ MCS-51 โดยจะหยุดจ่ายสัญญาณนาฬิกาให้ระบบ ทำให้ข้อมูลต่าง ๆ ภายใน MCS-51 ไม่มีการเปลี่ยนแปลง นอกจากนี้ยังลดพลังงานไฟฟ้าที่จ่ายให้ MCS-51 ลงด้วย

หน่วยความจำภายนอก (External Memory)

MCS-51 สามารถอ้างหน่วยความจำภายนอกได้ 64 K และอ้างหน่วยความจำโปรแกรมภายนอกได้ 64 K MCS-51 จะใช้พอร์ท 0 ในการอ้างตำแหน่งหน่วยความจำ 8 บิตล่าง และใช้พอร์ท 0 เป็นพอร์ทข้อมูลด้วย โดยใช้ขา ALE มาเป็น Latch ข้อมูลพอร์ท 0 และใช้พอร์ท 2 เป็นขาอ้างตำแหน่ง 8 บิตบน

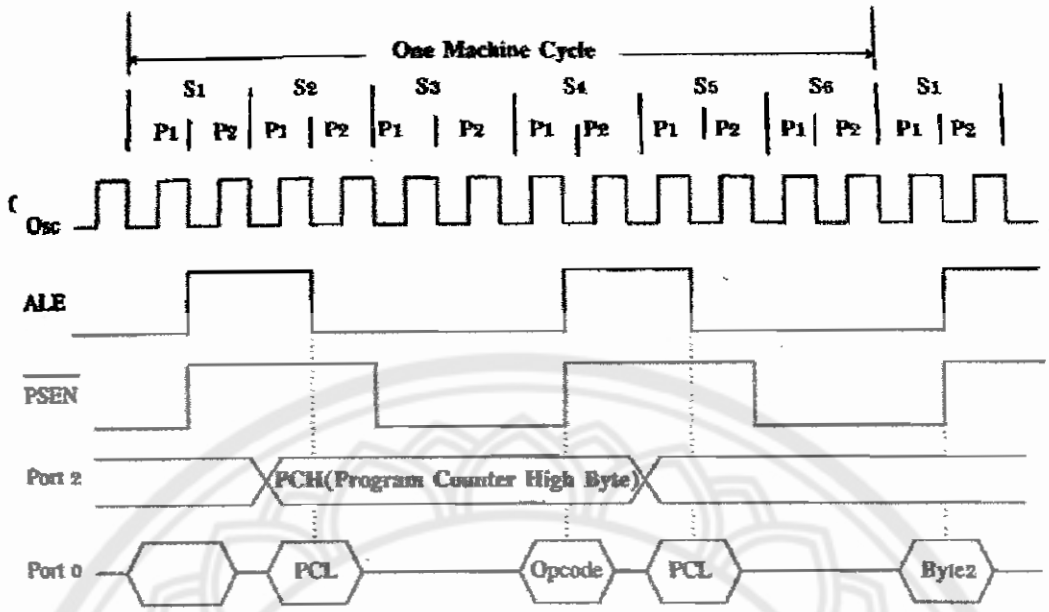
เนื่องจากพอร์ท 0 จะใช้งาน 2 หน้าที่ ในการติดต่อกับหน่วยความจำ จะใช้วิธี มัลติเพลกเซอร์ระหว่างแอดเดรส กับดาต้า พิจารณารูปที่ 2.8 ถ้าต้องการติดต่อกับหน่วยความจำที่เก็บข้อมูล 8 บิต และเก็บได้ 64 K จะต้องใช้สายสัญญาณ 24 เส้น คือ เป็นขาแอดเดรส 16 เส้น และขาข้อมูล 8 เส้น ดังรูปที่ 2.8(ก) แต่ถ้าใช้วิธีมัลติเพลกเซอร์ คือใช้ขา A0-A7 เป็นขาข้อมูลด้วยคือ D0-D7 จะใช้สายสัญญาณเพียง 16 เส้นเท่านั้น จากรูปที่ 2.8(ข) จะเห็นว่าเมื่อต้องการติดต่อกับหน่วยความจำจะส่งสัญญาณแอดเดรส A0-A15 ออกมาก่อน 16 เส้น และเวลาต่อมาขา A0-A7 จะถูกเปลี่ยนเป็น D0-D7



รูปที่ 2.8 ไดอะแกรมกลุ่มสัญญาณที่ใช้อ่านข้อมูล

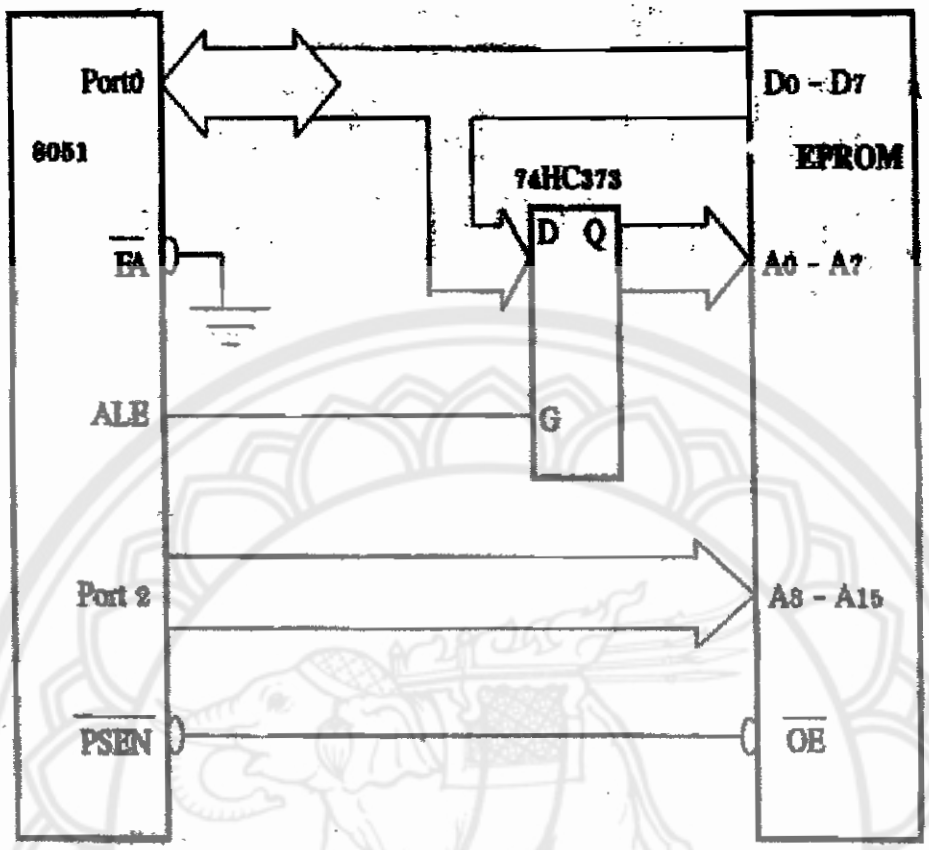
การติดต่อกับหน่วยความจำโปรแกรมภายนอก

ในการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก MCS-51 จะส่งค่าตำแหน่งของหน่วยความจำออกไปก่อน ซึ่งค่าตำแหน่งจะเก็บอยู่ใน PC โดยส่งออกไปทางพอร์ท 0 และพอร์ท 2 จากนั้นเวลาต่อมาจะส่งขา ALE ให้เป็นลอจิก "0" เพื่อ Latch ขาแอดเดรสของ 8 บิตต่ำ คือพอร์ท 0 จากนั้นจะส่งสัญญาณทางขา PSEN ให้เป็นลอจิก "0" เพื่ออ่านข้อมูลซึ่งจะได้ Opcode เข้าไปทางขา Data Bus คือพอร์ท 0 ไดอะแกรมเวลาอ่านข้อมูลจากหน่วยความจำภายนอกแสดงดังรูปที่ 2.9



รูปที่ 2.9 ไดอะแกรมอ่านข้อมูลจากหน่วยความจำภายนอก

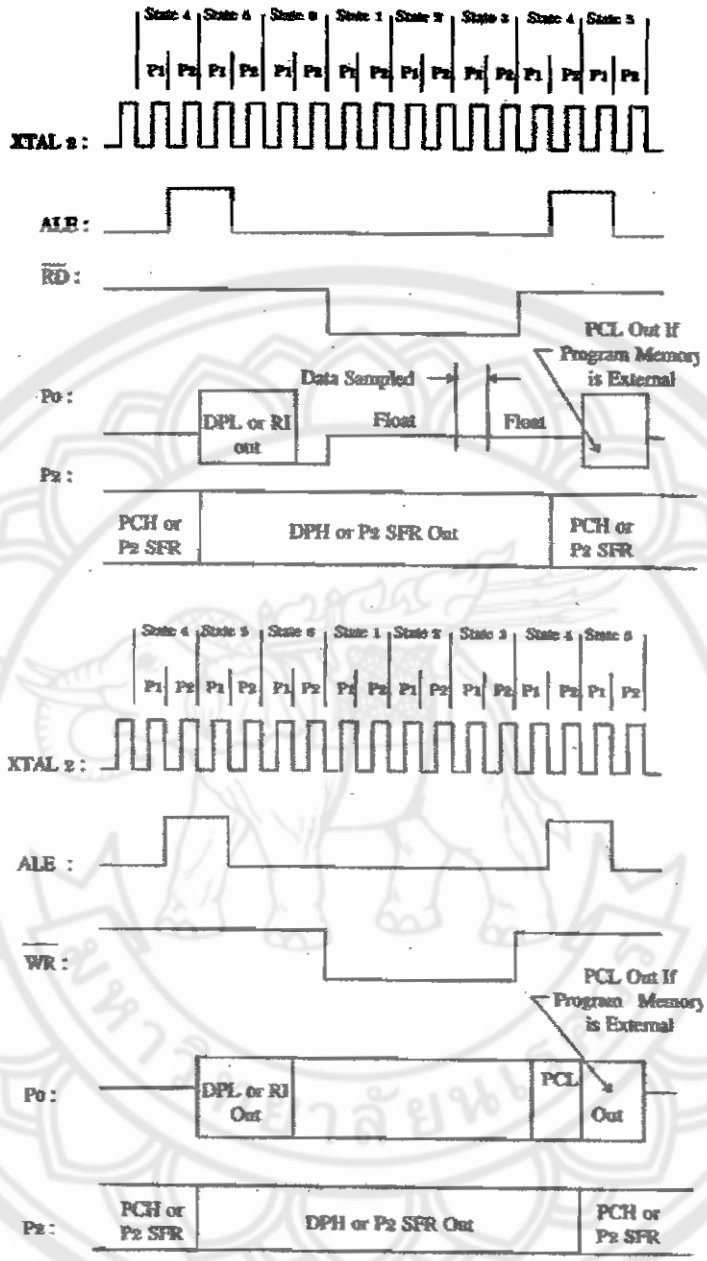
สำหรับการต่อหน่วยความจำกับ MCS-51 แสดงได้ดังรูปที่ 2.10 โดยขา EA จะต่อเป็น "0" เพื่อบอก MCS-51 ว่าให้อ่านหน่วยความจำภายนอก สำหรับการมัลติเพลกเซอร์จะใช้ฟลิปฟลอป 8 ตัว เบอร์ 74373 เก็บค่าตำแหน่ง 8 บิตต่ำเอาไว้ เมื่อ MCS-51 ส่งค่าตำแหน่งพอร์ทออกไป เวลาต่อมาจะส่งขา ALE ให้เป็น "0" ซึ่งจะใช้ขานี้ต่อกับ 74373 เพื่อให้ Latch ข้อมูลสำหรับขา PSEN จะต่อกับขา Output Enable (OE) ของหน่วยความจำ



รูปที่ 2.10 การต่อ MCS-51 กับหน่วยความจำภายนอก

การติดต่อกับหน่วยความจำข้อมูลภายนอก

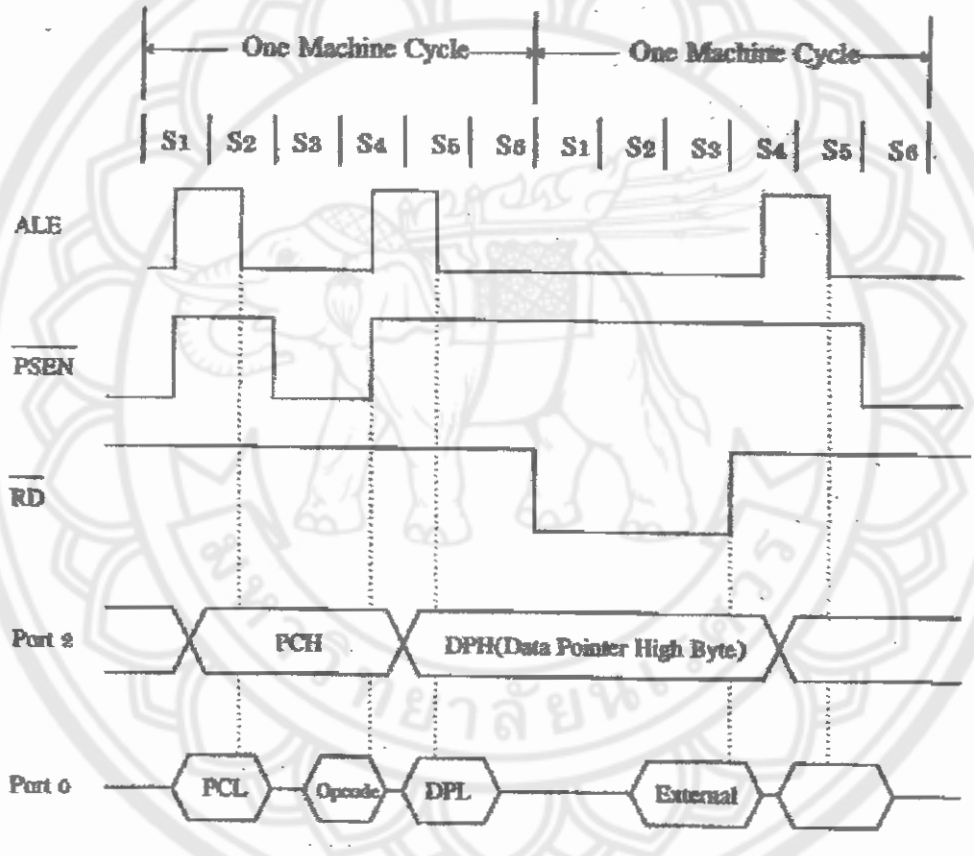
หน่วยความจำข้อมูลภายนอก MCS-51 สามารถอ่านและเขียนได้ในการติดต่อกับหน่วยความจำข้อมูลภายนอก MCS-51 ส่งแอดเดรสไปทางพอร์ท 0 และพอร์ท 2 จากนั้นจะส่งขา ALE เพื่อไป Latch Address 8 บิตต่ำ โดยการอ่านเขียนข้อมูลนั้นจะใช้ขา RD หรือ P3.7 และขา WR หรือ P3.6 ตามลำดับ **ไดอะแกรมเวลาการอ่านและเขียนข้อมูลกับหน่วยความจำข้อมูลภายนอกแสดงดังรูปที่ 2.11**



รูปที่ 2.11 ไดอะแกรมเวลาการอ่านและเขียนข้อมูลกับหน่วยความจำภายนอก

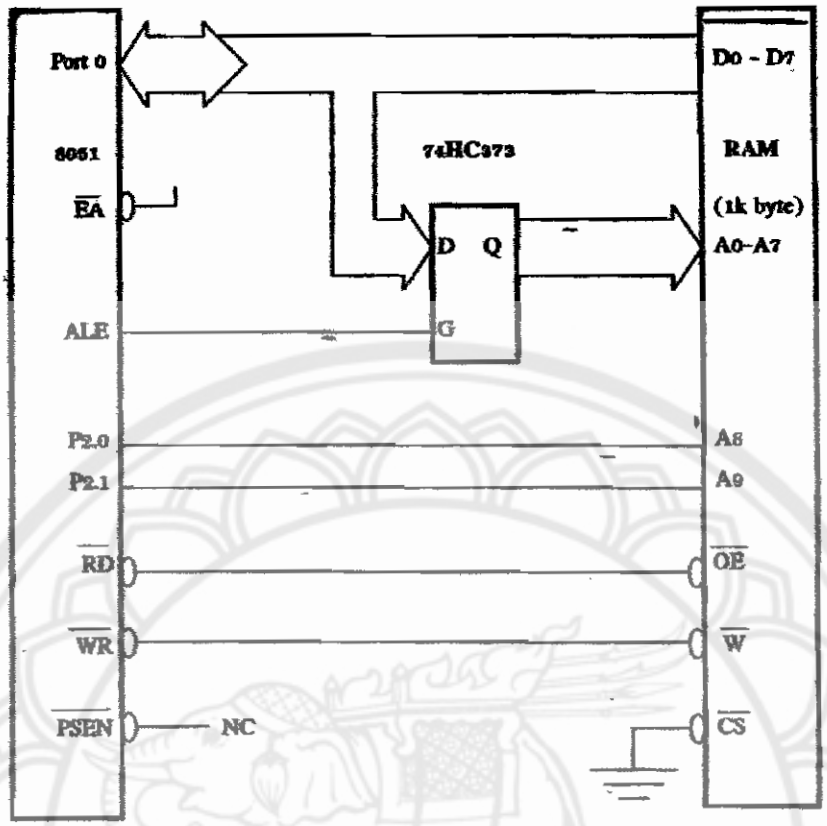
เนื่องจากตำแหน่งของหน่วยความจำภายนอกมีได้ถึง 64 K รีจีสเตอร์ที่ใช้เก็บค่าตำแหน่งของหน่วยความจำภายนอก จะใช้รีจีสเตอร์ 16 บิต คือ DPTR นอกจากนี้ยังใช้รีจีสเตอร์ 8 บิต ได้ 2 ตัวคือ R0 และ R1 ในการติดต่อกับหน่วยความจำ ภายนอกจะใช้คำสั่ง MOVX

ถ้าหาก MCS-51 ทำคำสั่ง MOVX A,@DPTR ซึ่งหมายความว่าให้อ่านค่าจากตำแหน่งที่ DPTR ชี้อยู่มาเก็บในรีจีสเตอร์ A ไดอะแกรมเวลาจะเป็นดังรูปที่ 2.11 โดยแมชชีนไซเคิลแรก จะเป็นการอ่านค่า Opcode ของโปรแกรมให้รู้ว่าทำคำสั่ง MOVX A,@DPTR ซึ่งในการอ่านค่าจากโปรแกรมจะได้ Opcode เข้ามาและตีความ จากนั้น MCS-51 จะรู้ว่าต้องอ่านข้อมูลจากตำแหน่งที่ DPTR ชี้อยู่ในแมชชีนไซเคิล ต่อไปก็จะนำค่า DPTR ส่งออกเป็นค่าแอดเดรส โดย DPH จะส่งไปทางพอร์ท 2 และ DPL จะส่งไปทางพอร์ท 0 จากนั้นขา ALE จะเป็น "0" เพื่อ Latch ข้อมูลแอดเดรส 8 บิตต่ำแวลาต่อมาขา RD จะเป็น "0" จากนั้นข้อมูลจะถูกอ่านเข้ามาทาง Data Bus คือพอร์ท 0 ไดอะแกรมเวลาการทำงานแสดงได้ดังรูปที่ 2.12



รูปที่ 2.12 สัญญาณต่าง ๆ ที่เกิดขึ้นขณะทำคำสั่ง MOVX

สำหรับการเชื่อมต่อหน่วยความจำข้อมูลกับ MCS-51 โดยให้ 8051 ทำงานกับหน่วยความจำแสดงได้ดังรูปที่ 2.13 ซึ่งจะเป็นการเชื่อมต่อ RAM ขนาด 1 K byte ซึ่งจะใช้ขาแอดเดรสเพียง 10 เส้น ดังนั้น A8 และ A9 จะต่อกับ P2.0 และ P2.1 ส่วนขา EA จะต่อกับลอจิก "1" เพื่อบอกว่าให้อ่านโปรแกรมจาก ROM ภายใน และขา PSEN จะไม่ใช้เพราะไม่ได้ต่อ ROM แสดงได้ดังรูปที่ 2.13

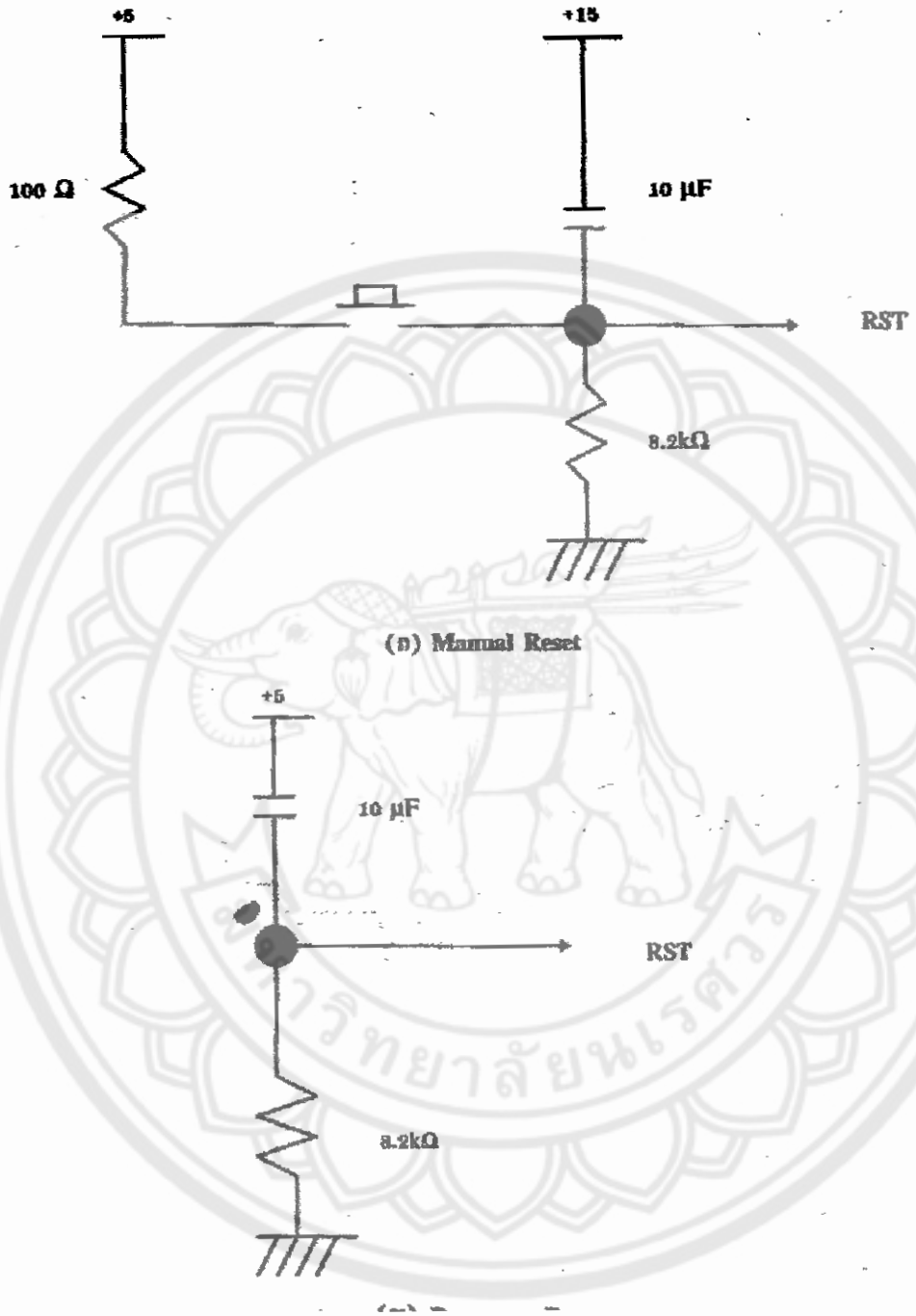


รูปที่ 2.13 การต่อหน่วยความจำโปรแกรมกับ MCS-51

ข้อสังเกต ขนาดที่ MCS-51 ติดต่อกับหน่วยความจำโปรแกรม หรือหน่วยความจำข้อมูลภายนอก จะใช้ขาแอดเดรสเหมือนกัน แต่จะต่างกันตรงที่ถ้าติดต่อกับหน่วยความจำโปรแกรมขา PSEN จะแอคทีฟ ถ้าติดต่อกับหน่วยความจำข้อมูลขา WR,RD จะแอคทีฟ และ MCS-51 จะติดต่อกับหน่วยความจำโปรแกรมด้วยคำสั่ง MOVC และติดต่อกับหน่วยความจำข้อมูลด้วยคำสั่ง MOVX

Reset Operation

การรีเซตหรือเริ่มทำงานใหม่ของ MCS-51 จะต้องให้ลอจิก "1" ที่ขา RST เป็นเวลา 2 แมชชีนไซเคิล จากนั้นให้กลับเป็นลอจิก "0" การรีเซตอาจทำได้โดยใช้ สวิตช์กด หรือใช้วิธี Power-up โดยใช้ตัว R-C ต่อเป็นวงจรดังรูปที่ 2.14



รูปที่ 2.14 การต่อวงจรรีเซต

ตารางที่ 2.4 ค่าต่าง ๆ ที่เกิดหลังการรีเซต

Register (S)	Counter
Program Counter	0000H
Accumulator	00H
B Register	00H
PSW	00H
SP	07H
DPTR	0000H
Ports 0-3	FFH
IP (8031 / 8051)	XXX00000B
IP (8032 / 8052)	XX000000B
IE (8031 / 8051)	0XX00000B
IE (8032 / 8052)	0X000000B
Timer Registers	00H
SCON	00H
SBUF	00H
PCON (HMOS)	0XXXXXXXB
PCON (CMOS)	0XXX0000B

เมื่อ MCS-51 ถูกรีเซต ค่ารีจิสเตอร์ต่าง ๆ จะถูกกำหนดค่าดังตารางที่ 2.4 โดย PC จะชี้ไปอยู่ที่ตำแหน่งเริ่มต้น คือ 0000H เมื่อขา RST กลับเป็น "0" MCS-51 จะเริ่มทำโปรแกรมที่ตำแหน่งแรก

ระบบอินเตอร์รัพท์ของ 8051

การติดต่อระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก มักจะทำโดยการตรวจสอบสภาวะของสัญญาณติดต่อระหว่างกัน การอินเตอร์รัพท์เป็นวิธีการหนึ่งที่นิยมนำมาใช้กับไมโครคอนโทรลเลอร์ เพื่อสามารถจัดการตอบรับหรือบริการกับอุปกรณ์ต่าง ๆ ให้เป็นไปได้ อย่างรวดเร็ว ความสามารถในการดำเนินการจัดการสัญญาณอินเตอร์รัพท์ จากแหล่งกำเนิดสัญญาณหลายประเภทของ 8051 ถือได้ว่าเป็นลักษณะเด่นประการหนึ่งซึ่งหากว่าได้มีการนำมาใช้ในการออกแบบแล้วก็จะส่งผลให้ระบบสามารถตอบสนองต่อเหตุการณ์ภายนอกที่เกิดขึ้นได้ดียิ่งขึ้น

การอินเตอร์รัพท์

ลักษณะการอินเตอร์รัพท์เป็นการขัดจังหวะการทำงานอย่างใดอย่างหนึ่งซึ่งกำลังดำเนินอยู่ เหตุการณ์คล้ายกับการที่เรากำลังคุยโทรศัพท์กับเพื่อนของเราอยู่ แล้วมีเสียงกริ่งโทรศัพท์ดังขึ้นมาซึ่งจะมีผลให้เกิดการหันเหความสนใจของเราในการสนทนาไปยังเครื่องโทรศัพท์แทนการจัดการต่อเหตุการณ์นี้กระทำได้ในหลายลักษณะ เช่น

1. ไม่สนใจกับเสียงกริ่งโทรศัพท์ และยังคงดำเนินการสนทนาต่อไปเช่นเดิม

2. หยุดพักการสนทนากับเพื่อนได้ชั่วขณะ และยกหูโทรศัพท์ขึ้นสนทนา

3. ขอจบการสนทนากับเพื่อนโดยทันที และยกหูโทรศัพท์ขึ้นสนทนา

ตัวอย่างข้างต้นแสดงให้เห็นถึงลักษณะของการอินเตอร์รัพท์ได้เป็นอย่างดี โดยเสียงกริ่งโทรศัพท์ ทำหน้าที่เป็นสัญญาณอินเตอร์รัพท์ติดต่อเข้ามาหาเรา ซึ่งกำลังดำนินงานอื่นอยู่ และแม้ว่าเราจะจัดการตอบสนองการขออินเตอร์รัพท์นั้นในลักษณะใดก็ตามจนเสร็จสิ้นแล้ว ก็จะต้องกลับมาดำเนินงานที่ค้างต่อไปเช่นเดิม

ประเภทของการอินเตอร์รัพท์

ไมโครคอนโทรลเลอร์ 8051 สามารถกิดการอินเตอร์รัพท์นี้ จะสามารถกำหนดให้มีการตรวจสอบในลักษณะเมื่อได้มีการเปลี่ยนแปลงระดับสัญญาณอินเตอร์รัพท์นั้น ได้แก่

1. สัญญาณอินเตอร์รัพท์ภายนอก (External Interrupt)

   การตรวจสอบสัญญาณที่เข้ามาอินเตอร์รัพท์นี้ จะสามารถกำหนดให้มีการตรวจสอบในลักษณะเมื่อได้มีการเปลี่ยนแปลงระดับสัญญาณ (Level-sensitive) ไปแล้ว หรือในช่วงเวลาขณะเริ่มมีการเปลี่ยนแปลงสัญญาณจากลอจิกสูงไปลอจิกต่ำ (Edge-sensitive)

2. สัญญาณอินเตอร์รัพท์ภายใน (Internal Interrupt)

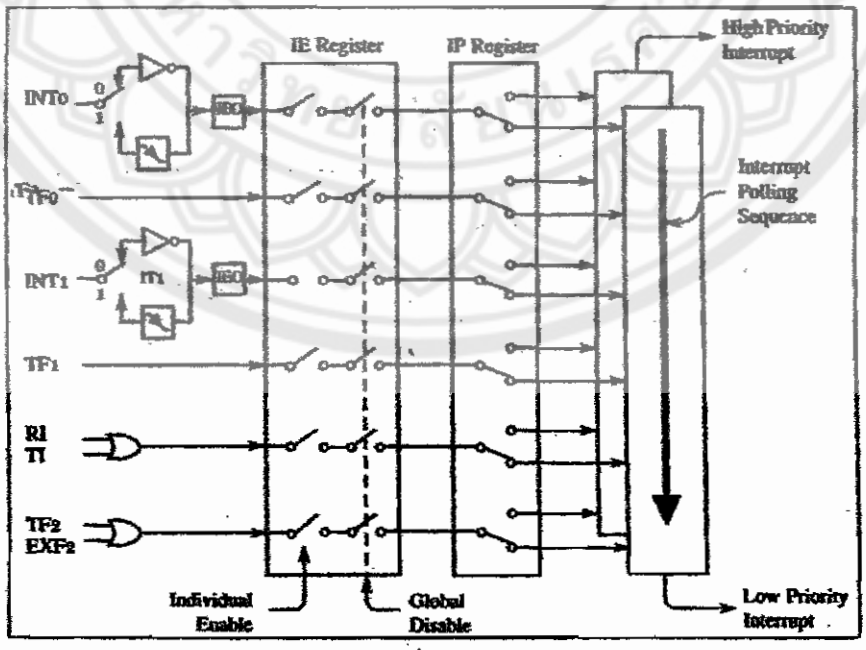
   แหล่งกำเนิดของสัญญาณนี้ จะเป็นวงจรภายในของไมโครคอนโทรลเลอร์เอง เช่น วงจรนับ / จับเวลา วงจรเชื่อมต่อสัญญาณอนุกรม เป็นต้น

โครงสร้างการอินเตอร์รัพท์

ตารางที่ 2.5 สัญญาณที่เข้ามาทำการอินเตอร์รัพท์ 8051

สัญญาณ	ความหมาย
INT0	สัญญาณอินเตอร์รัพท์จากภายนอกทางนอก ทางขาสัญญาณ P3.2 โดย 8051 จะทำการสุ่มตัวอย่างเมื่อสิ้นสุดทุกแมชชีนไซเคิล
INT2	สัญญาณอินเตอร์รัพท์จากภายนอกทางขาสัญญาณ P3.3 โดย 8051 จะทำการสุ่มตัวอย่างเมื่อสิ้นสุดทุกแมชชีนไซเคิล
TIMER 0	สัญญาณการเกิดโอเวอร์โฟลว์ของ Timer 0
TIMER 1	สัญญาณการเกิดโอเวอร์โฟลว์ของ Timer 1
พอร์ทอนุกรม	การเกิดอินเตอร์รัพท์ที่เกิดขึ้นจากการรับส่งข้อมูลอนุกรม ทำให้มีผลต่อแฟล็กอินเตอร์รัพท์ RI และ TI ตามลำดับ

จากแผนภาพโครงสร้างระบบอินเตอร์รัพท์ของ 8051 ในรูปที่ 2.15 จะเห็นว่าเมื่อเกิดการอินเตอร์รัพท์สัญญาณต่าง ๆ ขึ้น จะส่งผลให้มีการควบคุมเพื่อสั่งไมโครโปรเซสเซอร์กระโดดไปทำงานที่ตำแหน่งแอดเดรสต่าง ๆ ตามประเภทของแหล่งกำเนิดสัญญาณอินเตอร์รัพท์ที่เกิดขึ้น ซึ่งปกติแล้วควรจะต้องมีการสร้างโปรแกรมที่ตำแหน่งเหล่านี้ไว้ เพื่อทำหน้าที่เป็นโปรแกรมย่อยบริการอินเตอร์รัพท์



รูปที่ 2.15 แผนภาพแสดงโครงสร้างระบบการอินเตอร์รัพท์ของ 8051

การกำหนดให้ 8051 สามารถตอบรับการอินเตอร์รัพท์แต่ละประเภท ทำได้โดยกำหนดบิตของข้อมูลที่เกี่ยวข้อง ซึ่งมักจะอยู่ภายในรีจีสเตอร์ TCON และ SCON หากว่าได้มีการกำหนดค่าของบิตซึ่งอยู่ภายในรีจีสเตอร์ IE (Interrupt Enable Register) ด้วยแล้ว ก็จะสามารถตอบรับการอินเตอร์รัพท์ของสัญญาณนั้น ๆ ได้ นอกจากนี้แล้วตามแผนภาพในรูปที่ 2.15 ยังแสดงให้เห็นว่าสัญญาณอินเตอร์รัพท์แต่ละประเภท ยังสามารถกำหนดความสำคัญ (Priority) ของการอินเตอร์รัพท์ได้สองลักษณะ คือ ระดับความสำคัญสูงหรือต่ำ (High Low Priority) กล่าวคือ ขณะที่กำลังประมวลผลอยู่ภายในส่วนของโปรแกรมย่อยบริการอินเตอร์รัพท์ของสัญญาณที่มีระดับความสำคัญต่ำอยู่ ก็อาจจะถูกขัดจังหวะให้ไปประมวลผลของสัญญาณอินเตอร์รัพท์ที่มีระดับความสูงกว่าได้ แต่หากว่าเป็นสัญญาณอินเตอร์รัพท์ที่มีระดับความสำคัญต่ำเช่นเดียวกันแล้วก็จะต้องรอให้เสร็จสิ้นการประมวลผลที่ดำเนินการอยู่ก่อน

การควบคุมอินเตอร์รัพท์

ตามโครงสร้างด้านการจัดการอินเตอร์รัพท์ของ 8051 สามารถกำหนดเลือกเพื่อยินยอมหรือไม่ยินยอม (Enable/Disable) ให้มีการอินเตอร์รัพท์ของแต่ละสัญญาณได้ โดยใช้วิธีการกำหนดค่าของบิตภายในรีจีสเตอร์ IE ซึ่งจะมีทั้งแบบที่ระบุถึงอินเตอร์รัพท์โดยรวมทั้งหมด (บิตที่ 7) และอินเตอร์รัพท์แต่ละประเภทได้ ในกรณีที่กำหนดค่าข้อมูลเป็น 1 ให้กับบิตจะมีความหมายถึงการยินยอมให้การอินเตอร์รัพท์เกิดขึ้นได้ และจะเป็นกรณีตรงข้ามกันสำหรับการกำหนดค่าข้อมูลที่เป็น 0 หากลองย้อนกลับไปพิจารณาแผนภาพรูปที่ 2.15 อีกครั้ง จะเห็นว่าจะต้องทำการกำหนดให้ยินยอมการอินเตอร์รัพท์ทั้งหมดให้เกิดขึ้นก่อน จึงจะมีผลทำให้การกำหนดบิตเพื่อยินยอมของแต่ละอินเตอร์รัพท์มีผลขึ้นได้

ตารางที่ 2.6 บิตต่าง ๆ ภายในรีจีสเตอร์ IE (Interrupt Enable)

ชื่อบิต	ตำแหน่ง	ความหมาย
EA	IE.7	อีนาเบิล/ดิสเอเบิลการเกิดอินเตอร์รัพท์โดยรวม
-	IE.6	-
ET2	IE.5	อีนาเบิล/ดิสเอเบิลการเกิดอินเตอร์รัพท์ Timer 2
ES	IE.4	อีนาเบิล/ดิสเอเบิลการเกิดอินเตอร์รัพท์พอร์ทอนุกรม
ET1	IE.3	อีนาเบิล/ดิสเอเบิลการเกิดอินเตอร์รัพท์ Timer 1
EX1	IE.2	อีนาเบิล/ดิสเอเบิลการเกิดอินเตอร์รัพท์ INT 1
ET0	IE.1	อีนาเบิล/ดิสเอเบิลการเกิดอินเตอร์รัพท์ Timer 0
EX0	IE.0	อีนาเบิล/ดิสเอเบิลการเกิดอินเตอร์รัพท์ INT 0

ระดับความสำคัญของการอินเตอร์รัพท์

การกำหนดระดับความสำคัญให้กับสัญญาณอินเตอร์รัพท์แต่ละประเภทนั้น สามารถทำได้โดยการกำหนดข้อมูลที่มีค่าเป็น 1 หรือ 0 ให้กับบิตภายในรีจิสเตอร์ IP (Interrupt Priority) ดังแสดงในตารางที่ 2.18 โดยหากว่ามีค่าเป็น 1 ก็จะทำให้สัญญาณอินเตอร์รัพท์นั้น ๆ มีระดับความสำคัญสูง แบะในกรณีตรงข้ามกันสำหรับการกำหนดค่าเป็น 0

กรณีที่สัญญาณนั้นเข้ามาอินเตอร์รัพท์มีระดับความสำคัญเดียวกันเกิดขึ้นพร้อมกัน ก็อาจจะทำให้เกิดปัญหาขึ้นได้ แต่อย่างไรก็ตาม 8051 ก็มีโครงสร้างทางด้านฮาร์ดแวร์ในการพิจารณาตามระดับของตารางที่ 2.7

ตารางที่ 2.7 สัญญาณขาต่าง ๆ ของ MCS-51 และความหมาย

ระดับความสำคัญ	สัญญาณ	ความหมาย
1.	IE0	อินเตอร์รัพท์ภายนอก 0
2.	TF0	วงจรนับ/จับเวลา 0
3.	IE1	อินเตอร์รัพท์ภายนอก 1
4.	TF1	วงจรนับ/จับเวลา 1
5.	RI หรือ RT	วงจรรับ/ส่งข้อมูลอนุกรม

ตารางที่ 2.8 บิตต่าง ๆ ภายในรีจิสเตอร์ IP (Interrupt Priority)

ชื่อบิต	ตำแหน่ง	ความหมาย
	IP.7	
	IP.6	
PT2	IP.5	ระดับความสำคัญของ Timer 2
PS	IP.4	ระดับความสำคัญของพอร์ทอนุกรม
PT1	IP.3	ระดับความสำคัญของ Timer 1
PX1	IP.2	ระดับความสำคัญของ INT1
PT0	**IP.1**	**ระดับความสำคัญของ Timer 0**
PX0	**IP.0**	**ระดับความสำคัญของ INT0**

การจัดการอินเตอร์รัพท์

เมื่อมีการอินเตอร์รัพท์เกิดขึ้น ไมโครคอนโทรลเลอร์จะทำคำสั่งที่กำลังดำเนินการอยู่ให้แล้วเสร็จ จากนั้นจึงทำการเก็บค่าตำแหน่งแอดเดรสของคำสั่งที่จะทำงานต่อไปได้ยังบริเวณของหน่วยความจำที่ถูกกำหนดได้ให้เป็นสแต็ก (Stack) และกระโดดไปยังตำแหน่งแอดเดรสที่ได้มีการกำหนดได้แน่นอนตำแหน่งหนึ่งโดยอัตโนมัติตำแหน่งนี้เรียกว่า แอดเดรสของอินเตอร์รัพท์เวอร์เตอร์ (Interrupt Vector Address) ซึ่งผู้ใช้จะต้องทำการเขียนโปรแกรมย่อย (Subroutine) ยังตำแหน่งแอดเดรสเหล่านี้ไว้ ซึ่งเรียกว่า โปรแกรมย่อยบริการอินเตอร์รัพท์ (Interrupt Service Routine) ตำแหน่งของแอดเดรสเหล่านี้ได้แก่

ตารางที่ 2.9 แหล่งกำเนิดสัญญาณและแอดเดรส

แหล่งกำเนิดสัญญาณ	สัญญาณ	ตำแหน่งแอดเดรส (Hex)
IE0	อินเตอร์รัพท์ภายนอก 0	0003
TF0	วงจรนับ/จับเวลา 0	000B
IE1	อินเตอร์รัพท์ภายนอก 1	0013
TF1	วงจรนับ/จับเวลา 1	001B
RI หรือ TI	วงจรรับ/ส่งข้อมูลอนุกรม	0023

สิ่งที่ควรคำนึงถึงในการเขียนโปรแกรมย่อยบริการอินเตอร์รัพท์

1. ส่วนเริ่มต้นของโปรแกรมย่อย ควรจะมีการเก็บค่าของรีจีสเตอร์หรือแฟล็กสถานะต่าง ๆ ที่จะต้องนำไปใช้ในโปรแกรมย่อย มิฉะนั้นอาจมีผลทำให้โปรกรมปกติที่ทำอยู่ก่อนหน้าการมาทำงานโปรแกรมย่อยตอบสนองอินเตอร์รัพท์ทำงานผิดพลาดไปได้
2. บรรทัดสุดท้ายของโปรแกรมย่อยจะต้องสิ้นสุดด้วยคำสั่ง RETI เพื่อสั่งให้มีการนำค่าที่ได้เก็บไว้ก่อนหน้าการกระโดดมายังโปรแกรมย่อยบริการอินเตอร์รัพท์นี้ออกจากสแต็ก และ**กับไปทำงานเดิมต่อไป นอกจากนี้แล้วยังมีผลทำให้แฟล็กสถานะที่เกี่ยวข้องกับการอินเตอร์รัพท์นั้น ๆ ถูกรีเซตกลับไปเป็นค่าปกติเพื่อรอรับการอินเตอร์รัพท์ครั้งใหม่ต่อไปด้วย**

ขอให้ดูตัวอย่างโปรแกรมของโปรแกรมบริการอินเตอร์รัพท์ ซึ่งเริ่มต้นที่ตำแหน่งแอดเดรส 0003H ในที่นี้สมมุติว่าภายในส่วนการทำงานของโปรแกรมนั้น ต้องการนำข้อมูลจากรีจีสเตอร์หลายตัวมาคำนินการด้วย ได้แก่ รีจีสเตอร์ R0, R1 และรีจีสเตอร์ A แต่ไม่ต้องการที่จะทำให้ค่าของรีจีสเตอร์นี้เปลี่ยนแปลงไป ภายหลังจากที่สิ้นสุดการทำงานในโปรแกรมย่อยนี้ ดังนั้นจึงจำเป็นต้องเก็บสถานะของข้อมูลต่าง ๆ ของรีจีสเตอร์เหล่านี้ไว้ก่อน ด้วยการใช้คำสั่ง PUSH 00,01 และACC ตามลำดับ จากนั้นเมื่อเสร็จสิ้นการทำงานแล้วจึงค่อยนำค่ากลับคืนมา อีกครั้งหนึ่งด้วยคำสั่ง POP ซึ่งจะทำตามลำดับที่ตรงข้ามกัน กับเมื่อใช้คำสั่ง PUSH สำหรับการเก็บค่าของแฟล็กสถานะต่าง ๆ นั้นก็สามารถทำในลักษณะเดียวกันโดยใช้รีจีสเตอร์ PSW แทน

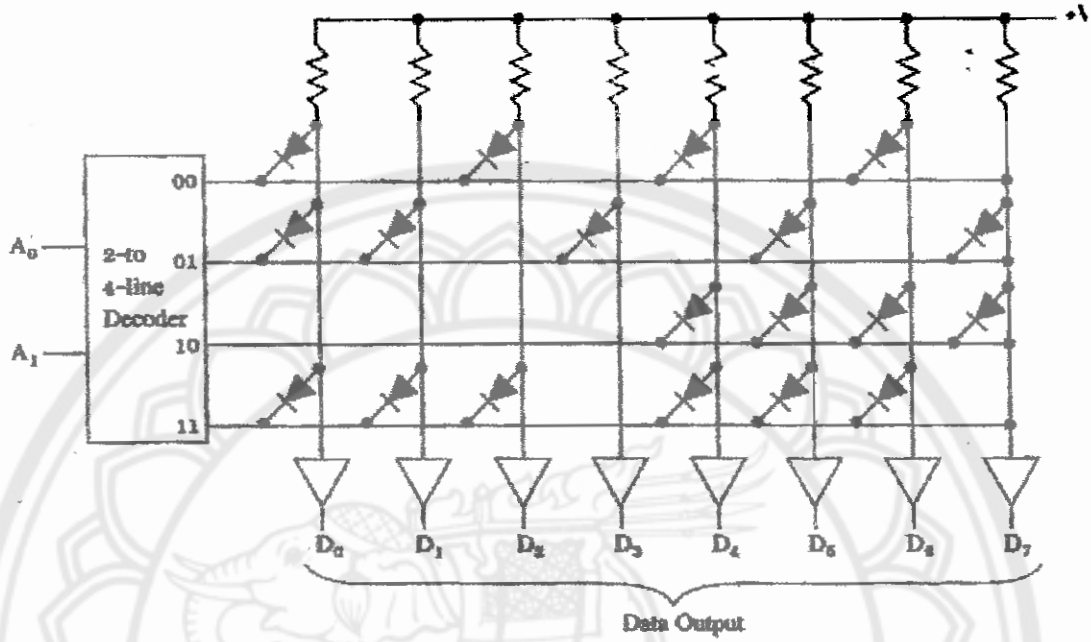
2.5หน่วยความจำและการเชื่อมต่อกับ MCS-51

1. หน่วยความจำโปรแกรม(Program Memory)

เป็นหน่วยความจำที่สามารถเก็บข้อมูลได้แม้ว่าไม่มีไฟเลี้ยง ข้อมูลก็ยังปรากฏอยู่ หน่วยความจำประเภทนี้มักใช้เก็บโปรแกรมระบบที่ไม่มีการเปลี่ยนแปลงอีกแล้ว เช่น โปรแกรมBIOS ในคอมพิวเตอร์ในบอร์ดควบคุมต่าง ๆ หน่วยความจำประเภทนี้แบ่งได้หลายชนิดแต่รียกรวมว่า ROM

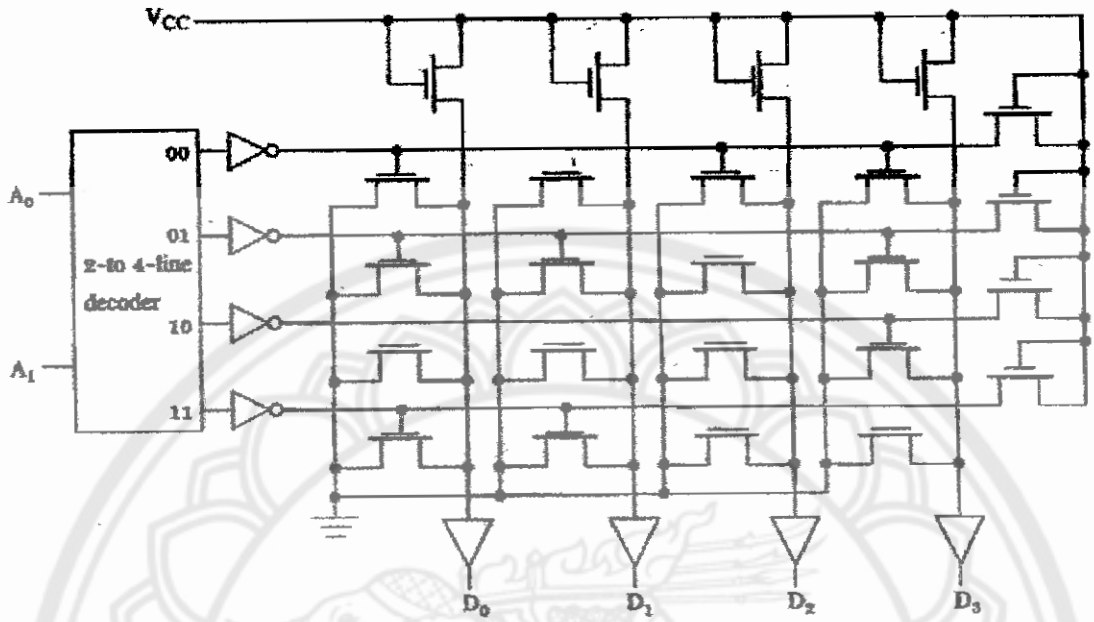
1.1 ROM (Read Only Memmory)

ข้อมูลในหน่วยความจำประเภทนี้ CPU จะอ่านได้อย่างเดียว ข้อมูลทั้งหมดที่อยู่ROMจะถูกโปรแกรมโดย โรงงานผู้ผลิตเรา ไม่สามารถนำมาโปรแกรมได้องหน่วยความจำประเภทนี้บางครั้งจะเรียกว่า Mask-programmed Rom โดยโครงสร้างง่าย ๆ แสดงได้ดังรูปที่ 2.16 โดยภายในไอซี ROM จะถูกสร้างเป็นวงจรถอดรหัส และไดโอดต่อกันอยู่ภายในชิพ จากรูปจะเป็นตัวอย่าง ROM ที่เก็บข้อมูลได้ 4 ไบต์ ในกาอ่านข้อมูลจาก ROM จะอ้างตำแหน่งแอดเดรส A0,A1 ซึ่งจะทำให้สัญญาณที่ผ่านวงจรถอดรหัสเป็น 0 ตำแหน่งใดที่มีไดโอดต่ออยู่จะทำให้ข้อมูลที่ลอจิกนั้นเป็น 0 ดังรูปที่ 2.16



ตำแหน่งแอดเดรส	ข้อมูลเลขฐานสอง								ข้อมูลเลขฐานสิบหก
	D0	D1	D2	D3	D4	D5	D6	D7	
0 0	0	1	0	1	0	1	0	1	55
0 1	0	0	1	0	1	0	1	0	2A
1 0	1	1	1	1	0	0	0	0	F0
1 1	0	0	0	1	0	0	0	1	11

รูปที่ 2.16 โครงสร้างของหน่วยความจำ ROM



ตำแหน่งแอดเดรส	ข้อมูลเอาต์พุตแสดง				ข้อมูลเอาต์พุตฐานสิบหก
	D0	D1	D2	D3	
00	0	1	0	0	4
10	0	0	1	0	2
01	1	1	1	0	E
11	0	0	1	1	3

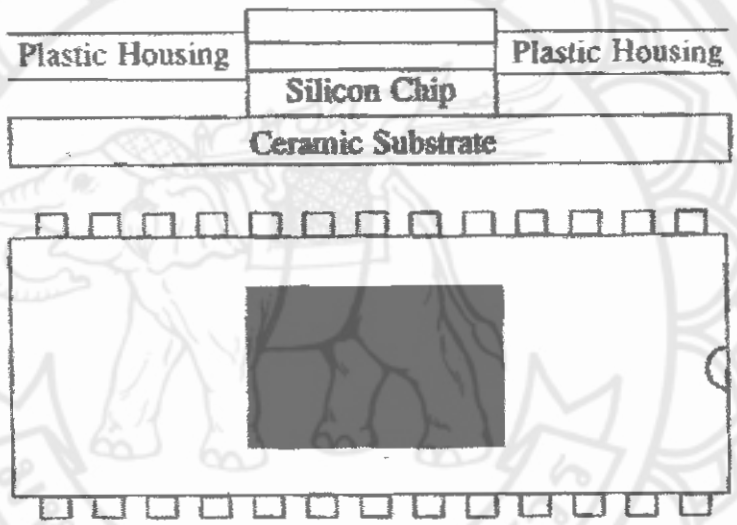
รูปที่ 2.17 เทคโนโลยีแบบ MOS-Transistor ในหน่วยความจำ ROM

1.2 PROM(Programmable Read Only Memmory)

เป็นหน่วยความจำ ROM ที่ผู้ใช้สามารถโปรแกรมข้อมูลเองได้โดยการป้อนพัลส์ที่มีแรงดันสูงทำให้โครงสร้างภายในตัวไอซีขาดออกจากกัน ทำให้เกิดลอจิก "1" หรือ "0" ตรงตามตำแหน่งที่กำหนดเมื่อ PROM ถูกโปรแกรมแล้วข้อมูลภายในไม่สามารถเปลี่ยนแปลงได้อีก

1.3 EPROM (Erasable Programmable Rom)

เรียกอีกอย่างหนึ่งว่า PROM ที่สามารถลบได้ ข้อมูลภายในหน่วยความจำชนิดนี้ผู้ใช้สามารถโปรแกรมเองได้ แต่ข้อมูลภายในสามารถเปลี่ยนแปลงได้โดยการลบข้อมูลออกให้หมดแล้วโปรแกรมไปใหม่การลบข้อมูลทำได้โดยการฉายแสงอุลตราไวโอเลตเข้าไปในตัวไอซีโดยตัวไอซีจะมีช่องพลาสติกไว้สำหรับฉายแสงเมื่อโปรแกรมลงบนไอซีชนิดนี้แล้วควรใช้วัตถุทึบแสงปิดเอาไว้ หน่วยความจำชนิดนี้เหมาะที่จะใช้งานที่มีโอกาสปรับปรุงโปรแกรมใหม่และเป็นหน่วยความจำโปรแกรมใหม่และเป็นหน่วยความจำโปรแรมที่นิยมมากเป็นอันดับหนึ่ง ลักษณะของ EPROM แสดงได้ดังรูปที่ 2.18 โดยด้านบนของชิพจะมีช่องพลาสติกเจาะอยู่เพื่อฉายแสง UV ลงไป สำหรับล้างข้อมูล



รูปที่ 2.18 ลักษณะของ EPROM

1.4 EEROM(Electrically Erasable Rom)

หน่วยความจำชนิดนี้จะคล้ายกับ EPROM แต่สามารถลบได้โดยใช้กระแสไฟฟ้าโดยไม่ต้องใช้แสง UV

2. หน่วยความจำข้อมูล

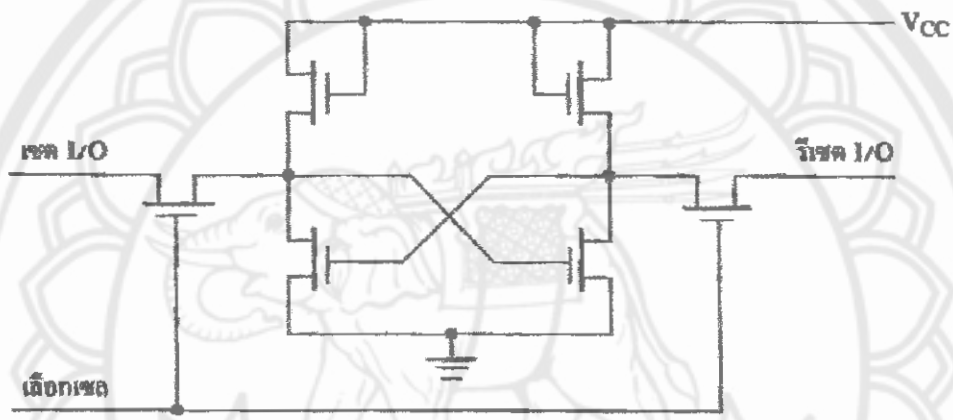
หน่วยความจำชนิดนี้ CPU สามารถจะอ่านและเขียนข้อมูลได้เรียกว่า RAM(Random Access Memmory) เป็นหน่วยเก็บความจำที่ใช้เก็บข้อมูลจากการประมวลผลของ CPU ข้อมูลภายใน

RAM จะคงอยู่ตลอดเวลาที่มีแหล่งจ่ายไฟต่อกับหน่วยความจำ ตัวหน่วยความจำ RAM นี้แบ่งออกได้เป็นสองชนิดใหญ่ ๆ คือ

1. สแตติกแรม (Static Ram)
2. ไดนามิกแรม (Dynamic Ram)

2.1 สแตติกแรม (Static Ram)

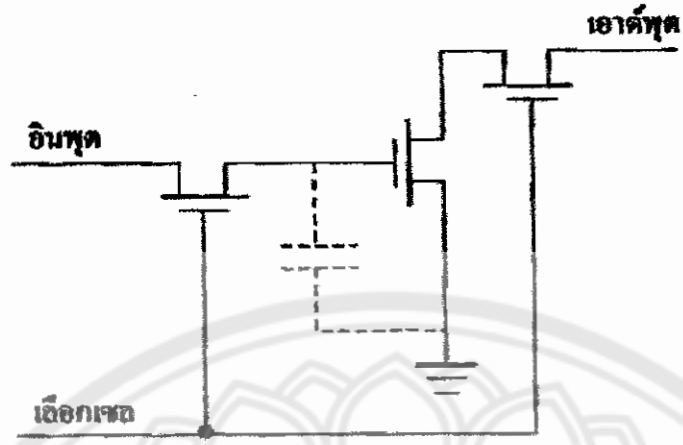
เป็นหน่วยความจำ RAM ที่สามารถอ่านเขียนข้อมูลได้ และจะคงอยู่ตลอดไปถ้ามีไฟเลี้ยง โครงสร้างภายในการเก็บข้อมูลแต่ละบิตจะสร้างฟลิปฟลอป ทำให้การเก็บข้อมูลแต่ละบิตจะสร้างจากทรานซิสเตอร์ออกมาหลายตัวดังรูปที่ 2.19แสดงโครงสร้างของสแตติกในการเก็บข้อมูล 1 บิต



รูปที่ 2.19 การเก็บข้อมูลแต่ละบิตของสแตติกแรม

2.2 ไดนามิกแรม (Dynamic Ram)

เป็นหน่วยความจำที่ข้อมูลจะคงอยู่ตลอดไปถ้ามีไฟเลี้ยง แต่จะต้องมีการกระตุ้นหรือการเขียนข้อมูลซ้ำ ตลอดเวลาด้วยวงจรพิเศษ เนื่องจากโครงสร้างภายในของไดนามิกแรมจะสร้างเป็นตัวเก็บประจุดังนั้น ในการเก็บข้อมูล 1 บิตจึงต้องมีการเขียนข้อมูลซ้ำเพื่อให้ประจุคงอยู่ และเนื่องจากโครงสร้างภายในตัวเก็บประจุทำให้การเก็บข้อมูล 14 บิตจะสร้างทรานซิสเตอร์ไม่กี่ตัว ซึ่งจะทำให้ข้อมูลต่อชิพสูงกว่าหน่วยความจำแบบสแตติกแรม โครงสร้างภายในแสดงได้ดังรูปที่ 2.20



รูปที่ 2.20 โครงสร้างภายในของสแตติกแรม

3. การอ่านและการเขียนข้อมูลหน่วยความจำ

ไอซีหน่วยความจำโดยทั่วไปจะประกอบด้วยกลุ่มสัญญาณ 3 กลุ่ม คือแอดเดรสบัส ข้อมูลบัสควบคุม โดยแอดเดรสบัสจะใช้ในการอ้างตำแหน่ง ถ้ามีแอดเดรสบัส 10 เส้นคือ A0-A9 สามารถอ้างตำแหน่งได้ 1 กิโลไบต์ บัสข้อมูลจะเป็นตัวบอกว่าในแต่ละตำแหน่งจะจะเก็บข้อมูลได้กี่บิต เช่น ถ้าข้อมูล 8 เส้นคือ D0-D7 จะสามารถเก็บข้อมูลได้ตำแหน่งละ 1 ไบต์ส่วนควบคุมจะใช้ในการอ่านเขียนหน่วยความจำ ระบบบัสทั้ง 3ระบบ

โดยทั่วไปไอซีหน่วยความจำจะมีขา CE (Chip Enable) ไว้สำหรับเลือกให้ไอซีถอดรหัสสำหรับการใช้งาน ในกรณีหน่วยความจำหลาย ๆ ตัวจะใช้กับไอซีถอดรหัสสำหรับออ่านข้อมูลถ้าขานี้แอคทีฟ ข้อมูลที่อยู่ในไอซีหน่วยความจำจะส่งออกมาทางบัสข้อมูล

การอ่านข้อมูลจากหน่วยความจำ มีลำดับขั้นดังนี้

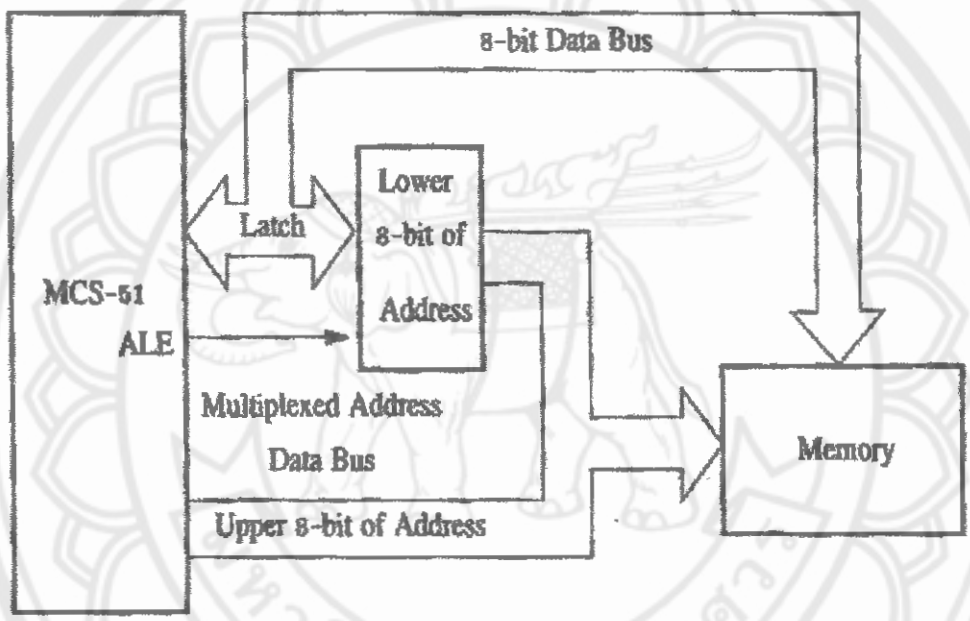
1. ส่งตำแหน่งที่อ่านไปก่อนทางแอดเดรสบัส
2. ส่งสัญญาณควบคุมว่าต้องการอ่าน
3. ข้อมูลในไอซีหน่วยความจำถูกส่งออกมาทางบัสข้อมูลได้

การเขียนข้อมูลลงในหน่วยความจำมีลำดับขั้นดังนี้

1. **ส่งตำแหน่งที่จะเขียนข้อมูลออกไปก่อนทางแอดเดรสบัส**
2. **ส่งตำแหน่งที่จะเขียนไปทางบัสข้อมูล**
3. **ส่งสัญญาณเขียนข้อมูล**

4. การเชื่อมต่อหน่วยความจำกับ MCS-51

ในการเชื่อมต่อกับหน่วยความจำจะต้องมีอุปกรณ์ภายนอกมา LATCH สัญญาณแอดเดรสบัสที่ได้จากพอร์ท 0 เพื่อที่จะใช้พอร์ท 0 เป็นบัสข้อมูลต่อไป จากรูปที่ 2.21 จะแสดงการต่อหน่วยความจำประเภท ROM และRAM กับ MCS-51 โดยมีอุปกรณ์ภายนอกมา Latch ค่าแอดเดรสบัสไบต์ต่ำเรียกว่า Address Latching



รูปที่ 2.21 การต่อ MCS-51 กับหน่วยความจำ

อุปกรณ์ที่นิยมใช้ได้แก่ ไอซี TTL เบอร์74LS373 โดยสัญญาณที่ใช้ LATCH คือสัญญาณ ALE จาก MCS-51

ในการอ่านหน่วยความจำโปรแกรมและการอ่านเขียนหน่วยความจำข้อมูลการเชื่อมต่อขาแอดเดรสบัสและข้อมูลจะเหมือนกันถ้าหน่วยความจำทั้งสองอยู่ตำแหน่งเดียวกันแต่จะแยกกันด้วยสัญญาณควบคุมซึ่งแสดงค่าลอจิกของสัญญาณได้ดังนี้

5. การเชื่อมต่อ MCS-51 กับหน่วยความจำโปรแกรมภายนอก

การอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกถ้า MCS-51 ต้องการอ่านข้อมูลจากหน่วยความจำข้อมูลภายนอกช่วงแรกจะส่งสัญญาณ ALE ให้เป็น 1 และ PSEN จะอยู่ในภาวะปกติคือเป็น 1จากนั้นพอร์ท 0 และพอร์ท 2 จะส่งค่าตำแหน่งของหน่วยความจำออกมา จากนั้น ALE จะกลับมาเป็นลอจิก 0 อุปกรณ์ภายนอกจะ LATCH ข้อมูลพอร์ท 0 ไว้ เพราะต้องใช้พอร์ท เป็นพอร์ทข้อมูล จากนั้นเวลาต่อมาพอร์ท 0 จะอยู่ในช่วงความต้านทานสูง ต่อจากนั้น PSEN จะเอกทีฟเป็นลอจิก 0 หมาย ความว่า MCS-51 ต้องการอ่านคำสั่ง จากนั้นข้อมูลคำสั่งจากหน่วยความจำโปรแกรมถูกเก็บเข้าไปใน Instruction Register ภายใน MCS-51 จากนั้น ALEและ PSEN จะกลับเป็น 1 ตามเดิม

จากไดอะแกรมเวลา การอ่านข้อมูลหน่วยความจำภายนอก จะเห็นว่าสัญญาณที่สำคัญในการเชื่อมต่อหน่วยความจำโปรแกรมกับ MCS-51 ได้แก่ ALE, PSEN, PORT0, PORT1 และอุปกรณ์ที่ใช้ Latch ข้อมูลของพอร์ท 0 ในการเชื่อมต่อ MCS-51 กับหน่วยความจำภายนอกจะใช้สัญญาณที่สำคัญ

การเชื่อมต่อ MCS-51 กับหน่วยความจำภายนอกนั้นสามารถติดต่อกับหน่วยความจำภายนอก 64 กิโลไบต์ จะมีไอซีหน่วยความจำขนาด 64 กิโลไบต์สามารถต่อได้โดยตรงเลย ซึ่งขาแอดแดรสของหน่วยความจำขนาด 64 กิโลไบต์ จะมี 16 เส้นก็นำขา A8-A15 ต่อโดยตรงกับพอร์ท 2 ของ MCS-51 ได้เลยแต่ถ้าหากมีโปรแกรมไม่ยาวมาก เช่นยาวประมาณ 64 กิโลไบต์ เราสามารถเลือกไอซีหน่วยความจำขนาด 8 กิโลไบต์ ได้ดังตารางที่ 2.10

ตารางที่ 2.10 การเก็บข้อมูล

A15	A14	A13	A12——————A0
X	X	X	1——————1

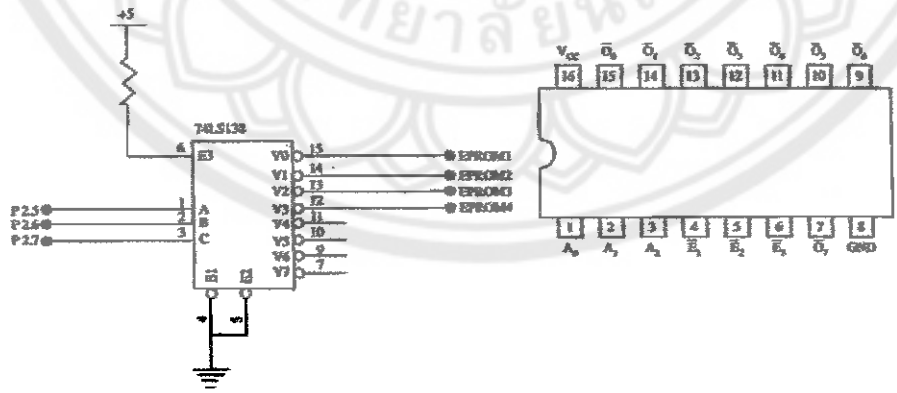
6. การเชื่อมต่อกับหน่วยความจำข้อมูลหลาย ๆ ตัว

เราสามารถนำหน่วยความจำหลาย ๆ ตัวมาต่อกันได้โดยเราต้องกำหนดโครงสร้างผังหน่วยความจำ (Memory Map)ออกมาก่อนว่าจะให้ EPROM แต่ละตัวอยู่ที่หน่วยความจำใดซึ่งแสดงลักษณะการทำงานดังตารางที่และดังรูปที่ 2.11

ตารางที่ 2.11 ตำแหน่งแอดเดรสใน EPROM

EPROM	ตำแหน่ง	A15	A14	A13
1	0000H - 1FFFH	0	0	0
2	2000H- 3FFFH	0	0	0
3	4000H- 5FFFH	0	1	0
4	6000H- 7FFFH	0	1	1
5	8000H- 9FFFH	1	0	0
6	A000H- BFFFH	1	0	1
7	C000H- DFFFH	1	1	0
8	E000H- FFFFH	1	1	1

EPROM	ตำแหน่ง	A15	A14	A13
1	0000H-1FFFH	0	0	0
2	2000H-3FFFH	0	0	1
3	4000H-5FFFH	0	1	0
4	6000H-7FFFH	0	1	1
5	8000H-9FFFH	1	0	0
6	A000H-BFFFH	1	0	1
7	C000H-DFFFH	1	1	0
8	E000H-FFFFH	1	1	1

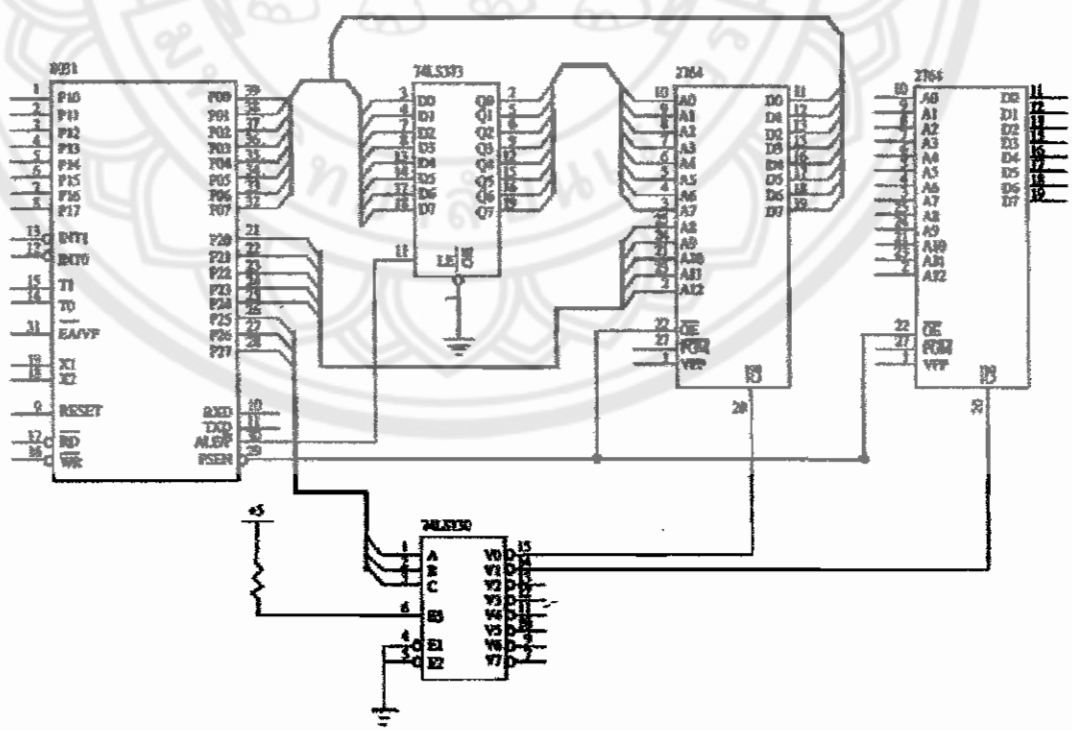


รูปที่ 2.22 การทำงานของไอซีเบอร์ 74LS138

ตารางที่ 2.12 สัญญาณลอจิก

อินพุต						เอาต์พุต							
E1	E2	E3	A0	A1	A2	00	01	02	03	04	05	06	07
X	X	X	X	X	X	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H
L	L	L	X	X	X	H	H	H	H	H	H	H	H
L	L	H	L	L	L	L	H	H	H	H	H	H	H
L	L	H	H	L	L	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	L	H	H	H	H	H
L	L	H	H	H	L	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	L	H	H
L	L	H	L	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	L

การทำให้ไอซีเบอร์ 74LS138 ทำงานได้ G2Aและ G2B จะต้องเป็นลอจิก 0 ถ้า P2.5,P2.6,P2.7เป็น0 หมดขา15ของ74LS138 จะทำงานดังนั้น EPROM ตัวแรกจะถูกเลือกให้ทำงานเรา สามารถออกแบบระบบได้ดังรูปที่ 2.23



รูปที่ 2.23 การเชื่อมต่อ MCS-51 กับหน่วยความจำหลาย ๆ ตัว

7. การเชื่อมต่อหน่วยความจำข้อมูล

การเชื่อมต่อหน่วยความจำข้อมูลภายนอกกับ MCS-51 จะเป็นทำนองเดียวกับการเชื่อมต่อกับหน่วยความจำโปรแกรมภายนอก ไอซีหน่วยความจำที่ใช้จะใช้แบบสแตติก(Static Ram) จากไดอะแกรมเวลาจะเห็นว่า MCS-51 ต้องการติดต่อกับหน่วยความจำภายนอกสัญญาณที่ต้องใช้คือ พอร์ท 0และพอร์ท 2 สำหรับเป็นแอดเดรสบัส สัญญาณ ALE สำหรับ Latch สัญญาณ พอร์ท 0 สัญญาณ RD และสัญญาณ WR สำหรับอ่านและเขียนข้อมูลบนหน่วยความจำตามลำดับการเชื่อมต่อหส่วยความจำข้อมูลภายนอกกับ MCS-51 แสดงได้ดังรูปที่

หากต้องการเชื่อมต่อหน่วยความจำข้อมูลภายนอกหลาย ๆ ตัวกับ MCS-51จะต้องมีวงจรถอดรหัสเพื่อเลือกให้หน่วยความจำทำงานในลักษณะเดียวกับการเชื่อมต่อกับหน่วยความจำโปรแกรม

2.6 การต่อเชื่อม 8255กับ MCS-51

2.6.1 ขาต่าง ๆ ของ 8255

D0-D7 เป็นขาข้อมูลอินพุตและเอาต์พุตที่จะต้องผ่านเข้าออกจากส่วนนี้ D0-D7 จึงเป็นส่วนที่จะต่อเข้ากับระบบบัสของไมโครคอนโทรลเลอร์เพื่อให้ไมโครคอนเตทรลเลอร์สามารถอ่านหรือเขียนข้อมูลจากพอร์ทผ่านทางบัสนี้

CS ขานี้เป็นขาอินพุตที่จะรับสัญญาณจากภายนอกเพื่อเลือกชิพ 8255 โดยเมื่อขานี้เป็นลอจิก 0 จะทำให้ 8255 ต่อเข้ากับระบบบัสของไมโครคอนโทรลเลอร์เพื่อให้ไมโครคอนโทรลเลอร์เขียนหรืออ่านข้อมูลจากพอร์ทได้

RD ขาสัญญาณการอ่านเป็นสัญญาณอินพุตที่ส่งมาจาก CPU เมือสัญญาณนี้เป็น 0 และ CS เป็น 0 ด้วยตัว 8255 จะทำให้ตัว CPU อ่านข้อมูลจากบัสในขณะที่เป็นพอร์ทอินพุต

WR เป็นขาสัญญาณการเขียน จะแอคทีฟเมื่อสัญญาณ WR เป็น 0 และ CS เป็น 0 สัญญาณนี้มาจาก CPU เมื่อต้องการเขียนข้อมูลลงบนพอร์ทที่กำหนด

A0-A1ขาแอดเดรส ลอจิกทั้งสองขานี้จะถอดรสออกมาได้ 4 ค่าเพื่อกำหนดค่ารีจีสเตอร์ภายในที่เชื่อมต่อเข้ากับพอร์ทอินพุตและเอาต์พุตของ8255 ชื่อพอร์ท A การเลือกพอร์ทจะเลือกโดยขา A0-A1

RESET ขารีเซต เป็นขาสัญญารที่ส่งมาจากภายนอกเพื่อทำการรีเซต 8255 เพื่อเคลียร์สถานะต่าง ๆ ของ 8255 เมื่อ 8255 ได้รับการรีเซตในจะกลับเข้าสู่โหมดอินพุตหรือทุกพอร์ทเป็นพอร์ทเป็นพอร์ทอินพุต

PA0-PA7 เป็นสายสัญญาณที่เป็นพอร์ทของ 8255ชื่อพอร์ท A การเลือกพอร์ทจะเลือกโดยขา A0-A1

PB0-PB7 เป็นสายสัญญาณที่เป็นพอร์ทของ 8255ชื่อพอร์ท B การเลือกพอร์ทจะเลือกโดยขา A0-A1

PC0-PC7 เป็นสายสัญญาณที่เป็นพอร์ทของ 8255ชื่อพอร์ท C การกำหนดพอร์ทนี้จะได้รับการกำหนดโดยขาแอดเดรส A0-A1 พอร์ท C นี้แบ่งออได้สองกลุ่มคือกลุ่ม PC0-PC3 และกลุ่ม PC4-PC7

ถ้าต้องการให้ 8255ทำงานจะต้องทำให้ขา CS แอคทีฟจากนั้นเลือกพอร์ทที่จะติดต่อโดยรีจีสเตอร์แต่ละตัวใน 8255จะได้รับการกำหนดค่าควบคู่กับสัญญาณ RD และ WR เพื่อแสดงการทำงานต่าง ๆ ดังนี้สัญญาณต่าง ๆ ของขาควบคุมที่นำมาประกอบกันจะมีความหมายดังตารางที่ 2.13

ตารางที่ 2.13 สัญญาณของขาควบคุม

RD	WR	A0	A1	ความหมาย
1	0	0	0	ส่งข้อมูลไปที่พอร์ท A
0	1	0	0	อ่านข้อมูลจากพอร์ท A
1	0	0	1	ส่งข้อมูลไปที่พอร์ท B
0	1	0	1	อ่านข้อมูลจากพอร์ท B
1	0	1	0	ส่งข้อมูลไปที่พอร์ท C
0	1	1	0	อ่านข้อมูลจากพอร์ท C
1	0	1	1	เขียนข้อมูลเป็นรหัสควบคุม
0	1	1	1	ไม่ใช้

การทำงานของไอซีเบอร์ 8255 จะทำงานได้ 3 โหมดคือ

1. Mode 0 :Basic I/O
2. Mode 1 :Strobe I/O
3. Mode 2 :Bidirectional Bus

การให้ 8255ทำงานในแต่ละโหมดจะเลือกได้โดยการโปรแกรมให้กับ 8255 คำสั่งที่โปรแกรมจะมี 8บิต แต่ละบิตมีความหมายดังนี้

บิต D7 เป็นบิตแสดงรหัสคำสั่งควบคุมถ้าบิตนี้เป็น1 จะหมายถึงรหัสควบคุมนี้มีผลต่อการเปลี่ยนแปลงการเซตโหมดต่าง ๆ ของ 8255

บิต D6 และ D5เป็นโหมดของพอร์ท A ซึ่งมีการทำงาน 3 โหมดคือ โหมด 0 โหมด 1 และโหมด 2

บิต D4 ถ้ามีค่าเท่ากับ 0 หมายถึงการกำหนดพอร์ท A เป็นเอาต์พุตและถ้ามีค่าเป็น 1จะหมายถึงกำหนดให้พอร์ทอินพุต

บิต D3 เป็นบิตที่บอกการเซตพอร์ท C ถ้าเป็น 0 จะทำให้พอร์ท C บนเป็นเอาต์พุต

บิต D2 เป็นบิตที่กำหนดการเซตโหมดของพอร์ท Bถ้าเป็น 0หมายถึงเลือกพอร์ท B เป็นโหมด 0 และถ้าเป็น1 คือการเลือกโหมด 1

บิต D1 เป็นการกำหนดอินพุตและเอาต์พุทของพอร์ท B ถ้าเป็น 0 คือพอร์ทเอาต์พุต และถ้าเป็น1 คือพอร์ทอินพุต

บิต D0 เป็นบิตที่บอกการเซตพอร์ท C ล่างถ้าเป็น 0 จะให้พอร์ท ล่างเป็นเอาต์พุต

ตารางที่ 2.14 ความหมายบิตควบคุม

บิตที่	กลุ่ม	ความหมาย
D0	B	พอร์ท C ล่าง 1= อินพุต 2=เอาต์พุต
D1	B	พอร์ท B 1= อินพุต 2=เอาต์พุต
D2	B	เลือกโหมด 1= อินพุต 2=เอาต์พุต
D3	A	พอร์ท C บน 1= อินพุต 2=เอาต์พุต
D4	A	พอร์ท A 1= อินพุต 2=เอาต์พุต

กรณีอินพุตโหมด 1

PC3,PC4 และPC5 จะใช้เป็น Handshaking สำหรับพอร์ท A ส่วน PC0,PC1 และPC2 จะใช้กับพอร์ทB ส่วน PC6 และ PC7 สามารถโปรแกรมให้เป็นอินพุตหรือเอาต์พุตก็ได้โดยปกติอุปกรณ์ภายนอกจะส่งขอมูลมาทาง PA0-PA7 เมื่อสัญญาณ Active Low จากสัญญาณ Strob จะ Set B มาที่ PC4 สำหรับพอร์ท A หรือมาที่ PC2 สำหรับพอร์ท B สัญญาณ Strobe จะอ่านข้อมูลมาทางอินพุต เมื่อข้อมูลเข้ามาเต็มอินพุต บัฟเฟอร์จะกำหนดสัญญาณ Active High IBF บน PC5 สำหรับพอร์ทA หรือ PC1

สำหรับพอร์ท B ถ้า IBF เป็น 1 ไมโครคอนโทรลเลอร์จะอ่านข้อมูลจากพอร์ท A หรือ พอร์ทB เมื่ออ่านข้อมูลเรียบร้อยแล้วสัญญาณจะกลับเป็น 0 กรณีนี้ 8255 จะบอก CPU ให้อ่านข้อมูลจากพอร์ท

กรณีเอาต์พุตโหมด 1

PC7,PC6และPC3 จะใช้เป็น Handshaking สำหรับพอร์ท A ส่วน PC0,PC1 และPC2 จะใช้กับพอร์ทB ส่วน PC4 และ PC5 สามารถโปรแกรมให้เป็นอินพุตหรือเอาต์พุตก็ได้เมื่อ CPU เขียนข้อมูลลงไปที่พอร์ท A หรือ พอร์ท B เมื่อเอาต์พุตเต็มแล้ว สัญญาณ IBF จะกลายเป็น 0 เพื่อบอกอุปกรณ์ภายนอกว่ามีข้อมูลแล้ว เมื่อข้อมูลเอาต์พุตส่งไปแล้วจะมีสัญญาณตอบรับเพื่อที่จะรับข้อมูลใหม่จากอุปกรณ์ภายนอกคือ จะแอคทีฟ Low

การออกแบบให้ MCS-51 ติดต่อกับ 8255 ก็เหมือนกับการออกแบบให้ติดต่อกับพอร์ท ทั่วไป คือมอง 8255เหมือนกับหน่วยความจำข้อมูลภายนอกซึ่งเป็นหน่วยความจำขนาด 4 ไบต์ คือ พอร์ทA พอร์ท B พอร์ท C และพอร์ทควบคุมโดยใช้สัญญาณที่ได้จากวงจรถอดรหัส พอร์ทจะนำมาต่อกับขา CS ส่วนขา RD,WR,A0,A1ของ 8255 สามารถต่อโดยตรงกับ MCS-51 โดยการอ้างขาของ A0 และ A1 ถ้าสามารถต่อวงจรตามรูปพอร์ทต่าง ๆของ 8255จะมีหมายเลขดังตารางที่ 2.15

ตารางที่ 2.15 การทำงานของขาพอร์ท

พอร์ท	A1	A0	หมายเลขพอร์ท
A	0	0	000H
B	0	1	0001H
C	1	0	0002H
Control	1	1	0003H