

หน่วยที่ 2 โครงสร้าง และสถาปัตยกรรมของไอซี MCS-51

อดิศักดิ์ ชินะวงศ์

เอกสารประกอบการเรียนวิชาไมโครคอนโทรลเลอร์

เผยแพร่ที่ www.Adisak51.com

1. ตำแหน่งขาและหน้าที่การทำงาน ของไอซี MCS-51

ไอซีไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ทุกเบอร์จะใช้แรงดันไฟเลี้ยง 5 โวลต์ในการทำงาน ส่วนกระแสไฟฟ้า จะใช้แตกต่างกันไปตามเทคโนโลยีที่ผลิต โดยเบอร์ของไอซีที่มีตัวอักษร C อยู่ตรงกลางผลิตโดยอาศัยเทคโนโลยี HMOS (High-Performance N-Channel MOS) ซึ่งประหยัดพลังงานในการทำงาน สามารถใช้การควบคุมพลังงานของตัวไอซีได้จากโปรแกรม ไอซี MCS-51 มีรายละเอียด และการทำงาน ของขา แสดงดังภาพที่ 2.1

P1.0	1	40	VCC
P1.1	2	39	P0.0 (AD0)
P1.2	3	38	P0.1 (AD1)
P1.3	4	37	P0.2 (AD2)
P1.4	5	36	P0.3 (AD3)
P1.5	6	35	P0.4 (AD4)
P1.6	7	34	P0.5 (AD5)
P1.7	8	33	P0.6 (AD6)
RST	9	32	P0.7 (AD7)
(RXD) P3.0	10	31	EA/VPP
(TXD) P3.1	11	30	ALE/PROG
(INT0) P3.2	12	29	PSEN
(INT1) P3.3	13	28	P2.0 (A15)
(T0) P3.4	14	27	P2.1 (A14)
(T1) P3.5	15	26	P2.2 (A13)
(WR) P3.6	16	25	P2.3 (A12)
(RD) P3.7	17	24	P2.4 (A11)
XTAL2	18	23	P2.5 (A10)
XTAL1	19	22	P2.6 (A9)
GND	20	21	P2.7 (A8)

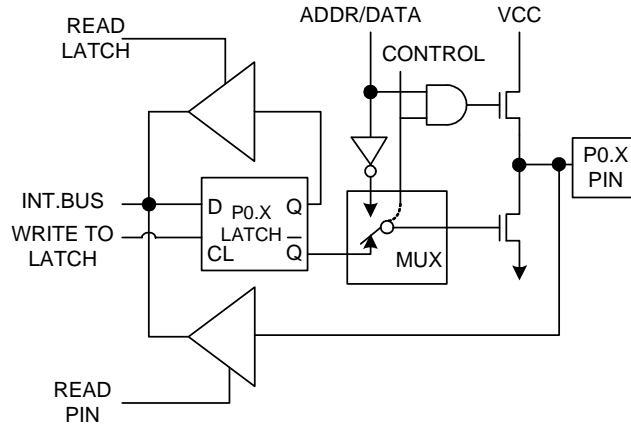
ภาพที่ 2.1 การจัดขาของไอซี MCS-51

Vcc เป็นขาสำหรับต่อแหล่งจ่ายไฟกระแสตรง +5VDC

GND เป็นขากราวด์ของไอซี

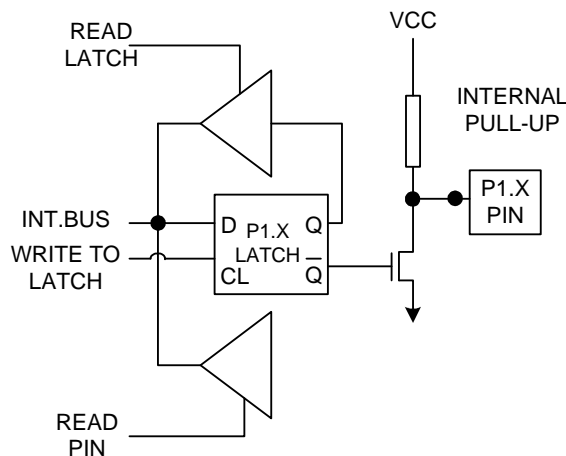
พอร์ต 0 (P0.0 -P0.7) มีจำนวน 8 ขา แต่ละขาเรียกเป็น 1 บิต ทำหน้าที่เป็นแอดเดรสบัส และดาต้าบัส (AD0-AD7) แสดงดังภาพที่ 2.2 สำหรับการต่อใช้งานกับหน่วยความจำภายนอก สามารถต่อเป็นพอร์ตอินพุตหรือพอร์ตเอาต์พุตทั่วไป ถ้าต้องการให้บิตใดเป็นอินพุต ให้เขียนข้อมูลบิตนั้นๆเป็นสถานะลอจิก “1” แล้วส่งไปแต่ละบิตของพอร์ตที่ต้องการ และที่เอาต์พุต \bar{Q} ของวงจรถัก (Latch) มีสถานะลอจิกเป็น “0” ทำให้หยุดการทำงานของ FET ตัวล่าง ส่งผลให้ขาพอร์ตจะมีสถานะลอจิกเป็น “1” ส่วนการอ่านค่าสัญญาณจากขาพอร์ตทำได้โดยการกระตุ้นที่ขา READ PIN ทำให้ Tri-state Buffer ตัวล่างทำงาน และรับ

สัญญาณลอจิกจากขาพอร์ตได้ การนำไปใช้งานต้องต่อทั้งขาของพอร์ตเข้ากับตัวต้านทานค่า 10 กิโลโอห์มไว้กับขา Vcc เพื่อทำหน้าที่เป็นตัวต้านทานพูลอัป (Pull-up Resistor)



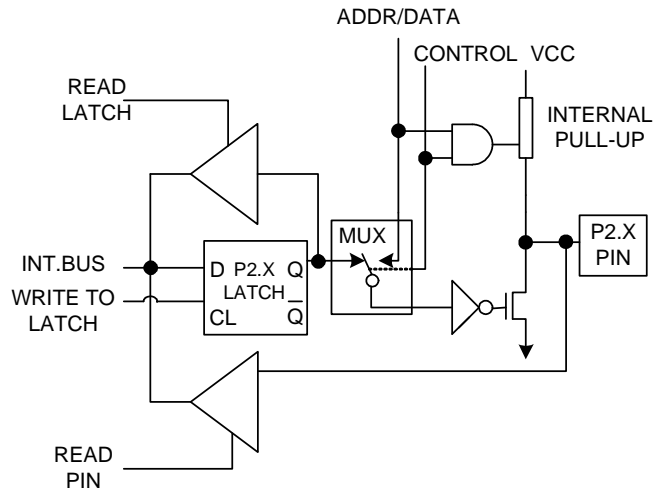
ภาพที่ 2.2 แสดงโครงสร้างพอร์ต 0 (1บิต)

พอร์ต 1 (P1.0-P1.7) มีจำนวน 8 บิต และในแต่ละบิตมีตัวต้านทานพูลอัปภายใน (Internal Pull-up Resistor) แสดงดังภาพที่ 2.3 สามารถกำหนดให้เป็นพอร์ตอินพุต หรือเป็นพอร์ตเอาต์พุตได้ สำหรับใช้งานโดยทั่วไป ถ้าให้ขาพอร์ตใดเป็นอินพุต จะต้องเขียนข้อมูลลอจิก “1” ไปแต่ละบิตของพอร์ตที่ต้องการติดต่อ พอร์ต 1 ยังมีขาสำหรับใช้ในการโปรแกรมแบบ ISP สำหรับไอซีเบอร์ AT89SXX ขา P1.5 จะเป็นขา MOSI (Master data Output Slave Data Input) ขา P1.6 จะเป็นขา MISO (Master Data Input Slave Data Output) และขา SCK (Master Clock Output)



ภาพที่ 2.3 แสดงโครงสร้างพอร์ต 1 (1บิต)

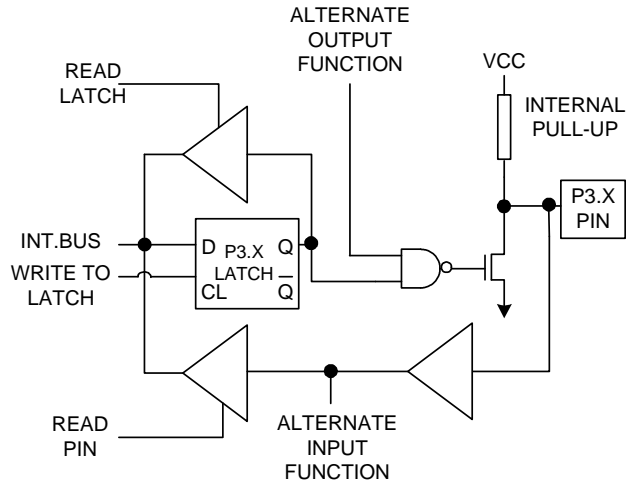
พอร์ต 2 (P2.0-P2.7) มีจำนวน 8 บิต ในแต่ละบิตจะมีตัวต้านทานพูลอัปอยู่ภายใน และจะทำงานได้สองลักษณะเช่นเดียวกับพอร์ต 0 โดยทำหน้าที่เป็นแอดเดรสบัส (A8-A15) สำหรับหน่วยความจำภายนอก และทำหน้าที่เป็นพอร์ตอินพุตหรือเอาต์พุต หากต้องการให้ขาพอร์ตใดเป็นอินพุต ให้เขียนข้อมูลที่พอร์ตหรือบิตนั้นๆ เป็นสถานะลอจิก “1” ทำให้หยุดการทำงานของ FET ตัวล่าง แสดงดังภาพที่ 2.4



ภาพที่ 2.4 แสดงโครงสร้างพอร์ต 2 (1บิต)

พอร์ต 3 (P3.0-P3.7) มีจำนวน 8 บิต ในแต่ละบิตมีตัวต้านทานพูลอัปอยู่ภายใน สามารถกำหนดให้เป็นได้ทั้งพอร์ตอินพุต และพอร์ตเอาต์พุตสำหรับการใช้งานทั่วไป หากต้องการกำหนดให้ขาของพอร์ตใดเป็นอินพุต ต้องเขียนข้อมูลให้เป็นลอจิก “1” ไปแต่ละบิตที่ต้องการติดต่อ แสดงดังภาพที่ 2.5 นอกจากนี้ขาของพอร์ต 3 ยังมีหน้าที่ใช้งานพิเศษดังนี้

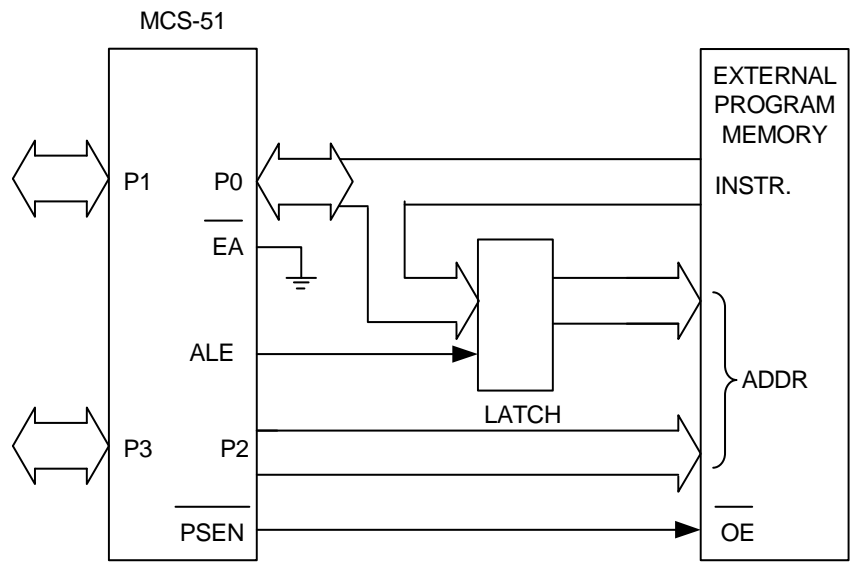
- P3.0 ใช้เป็นขาอินพุตสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD
- P3.1 ใช้เป็นขาอินพุตสำหรับส่งรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD
- P3.2 ใช้เป็นขาอินพุตรับสัญญาณอินเทอร์รัพท์จากภายนอกช่อง 0 หรือขา $\overline{INT0}$
- P3.3 ใช้เป็นขาอินพุตรับสัญญาณอินเทอร์รัพท์จากภายนอกช่อง 1 หรือขา $\overline{INT1}$
- P3.4 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทมเมอร์จากภายนอกช่อง 0 หรือขา T0
- P3.5 ใช้เป็นขาอินพุตสำหรับรับสัญญาณไทมเมอร์จากภายนอกช่อง 1 หรือขา T1
- P3.6 ใช้เป็นขาควบคุมการเขียนข้อมูลลงในหน่วยความจำภายนอก หรือขา \overline{WR}
- P3.7 ใช้เป็นขาควบคุมการอ่านข้อมูลลงในหน่วยความจำภายนอก หรือขา \overline{RD}



ภาพที่ 2.5 แสดงโครงสร้างพอร์ต 3 (1บิต)

$\overline{ALE}/\overline{PROG}$ (Address Latch Enable) เป็นขาเอาต์พุตทำหน้าที่แยกแอดเดรสไบต์ออกจากดาต้าบัสที่พอร์ต 0 โดยจะได้ค่าแอดเดรสไบต์ต่ำ (A0-A7) ถ้าหากไม่มีการติดต่อกับหน่วยความจำภายนอก จะส่งสัญญาณพัลส์ออกมาด้วยความถี่ 1/6 ของความถี่ออสซิลเลเตอร์ตลอดเวลา และยังสามารถนำไปใช้เป็นขาควบคุมการบันทึกข้อมูลลงในหน่วยความจำโปรแกรมภายในแบบ EPROM และ Flash แสดงดังภาพที่ 2.6

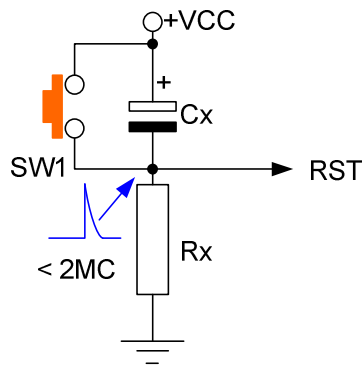
\overline{PSEN} (Program Store Enable) เป็นขาเอาต์พุต ทำหน้าที่ ส่งสัญญาณเป็นลอจิก “0” เมื่อไอซี MCS-51 ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก สัญญาณที่ขานี้จะส่งออกมาทุกๆ 2 ครั้ง ใน 1 แมกซ์ชีนไซเคิล แสดงดังภาพที่ 2.6



ภาพที่ 2.6 แสดงการใช้งานขา \overline{ALE} , \overline{PSEN} , \overline{EA}

\overline{EA}/VPP (External Access) เป็นขาอินพุตทำหน้าที่เลือกหน่วยความจำโปรแกรมภายในหรือภายนอกไอซี แสดงได้ดังภาพที่ 2.6 โดยถ้าหากให้ขามีสถานะลอจิกเป็น “0” จะเป็นการเลือกหน่วยความจำโปรแกรมภายนอก ถ้ากำหนดให้สถานะลอจิกเป็น “1” หมายถึงเลือกใช้หน่วยความจำโปรแกรมภายในไอซี นอกจากนี้ยังทำหน้าที่รับแรงดันไฟสูง เพื่อบันทึกข้อมูลให้กับหน่วยความจำโปรแกรมที่อยู่ในไอซีด้วย

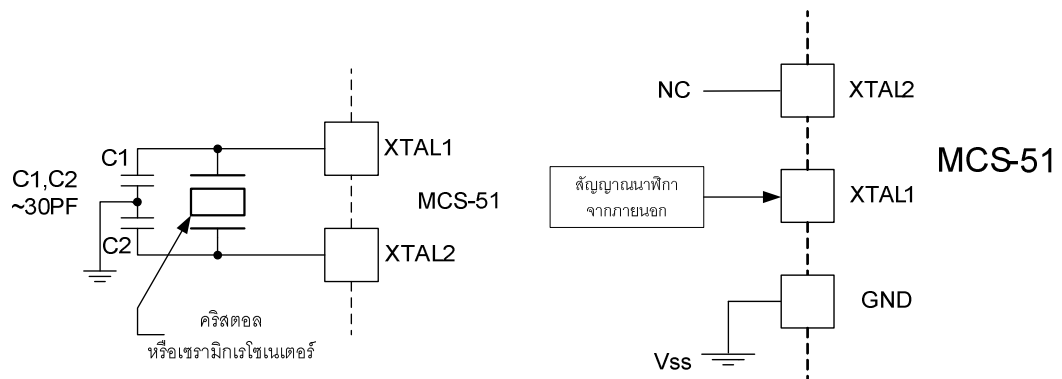
รีเซต (Reset) เป็นขาจับสัญญาณการรีเซต โดยจะใช้รีเซตวงจรทุกอย่างภายในไอซี MCS-51หรือเป็นการเริ่มต้นทำงานใหม่ในกรณีที่ไอซีมีการทำงานผิดพลาด การป้อนสัญญาณต้องทำให้สถานะของขาอยู่ในระดับลอจิก “1” (High) อย่างน้อย 2 แมกซ์ซีไนซ์เคลิล โดยที่วงจรกำเนิดสัญญาณนาฬิกา จะยังคงทำงานต่อเนื่องไปอย่างเป็นปกติ แสดงดังภาพที่ 2.7



ภาพที่ 2.7 แสดงวงจรรีเซต

XTAL1 ใช้ต่อคริสตอลภายนอก เป็นอินพุตเข้าสู่วงจรออสซิลเลเตอร์ภายใน แสดงดังภาพที่ 2.8

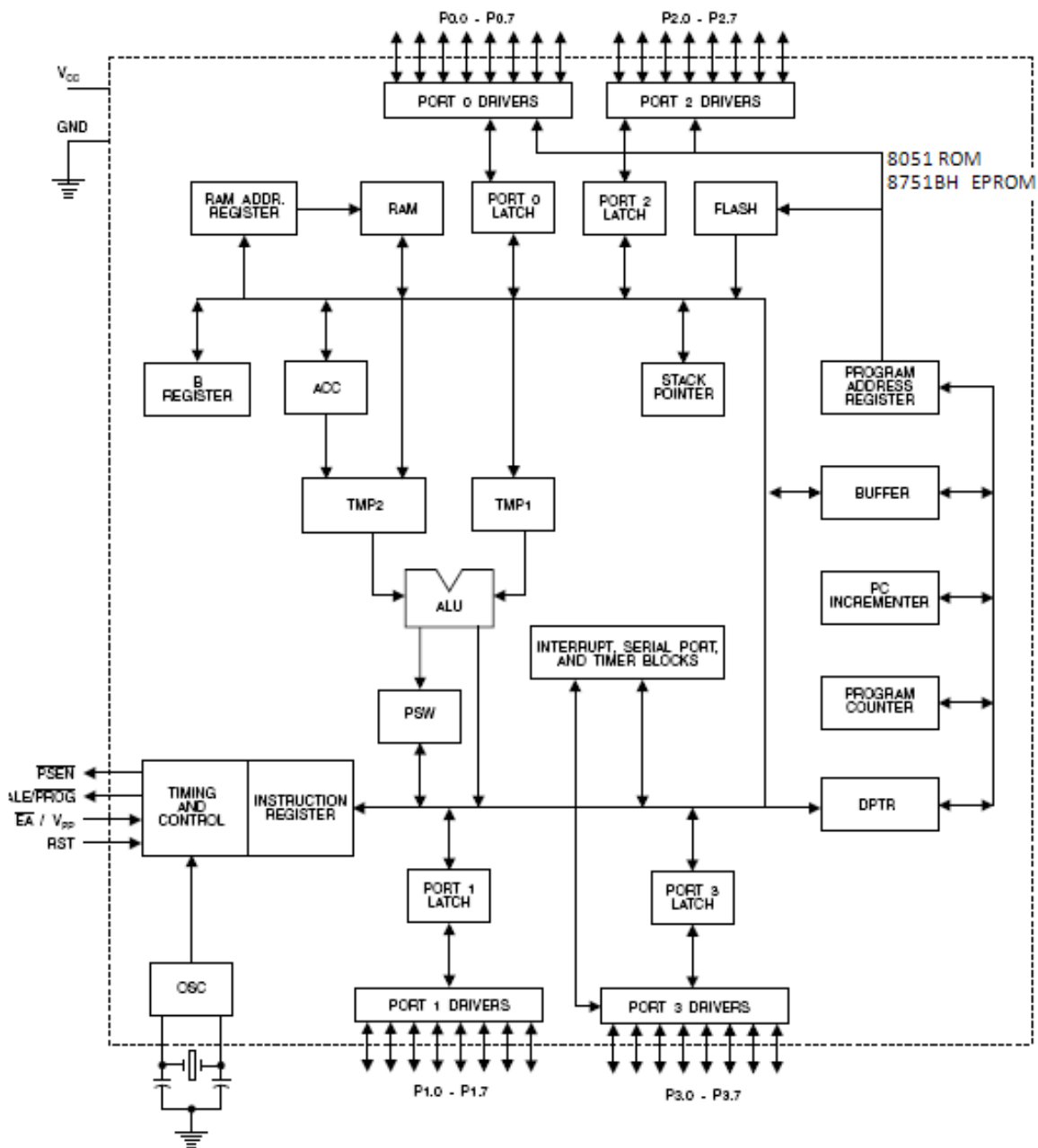
XTAL2 ใช้ต่อคริสตอลภายนอก เป็นเอาต์พุตออกจากวงจรออสซิลเลเตอร์ภายใน



ภาพที่ 2.8 แสดงการต่อคริสตอลเพื่อสร้างสัญญาณนาฬิกา

2. สถาปัตยกรรมภายในของไอซี MCS-51

สถาปัตยกรรมภายในของไอซีไมโครคอนโทรเลอร์ MCS-51 จะแบ่งการทำงานออกได้เป็น บล็อกการทำงานประกอบไปด้วย หน่วยความจำข้อมูล หน่วยความจำโปรแกรม วงจรส่วนที่ใช้ในการควบคุมการทำงาน หน่วยคำนวณทางคณิตศาสตร์ และลอจิก (ALU: Arithmetic and Logic Unit) รีจิสเตอร์ (Register) วงจรออสซิลเลเตอร์ (Oscillator) และพอร์ตต่างๆ โดยแต่ละบล็อกการทำงาน จะถูกเชื่อมต่อกับ บัสข้อมูล และบัสแอดเดรส แสดงดังภาพที่ 2.9



ภาพที่ 2.9 สถาปัตยกรรมภายในของไมโครคอนโทรเลอร์ 8051

(แหล่งอ้างอิง http://atmel.com/dyn/products/datasheets.asp?family_id=604)

3. การจัดหน่วยความจำของไอซี MCS-51

หน่วยความจำทำหน้าที่เก็บ โปรแกรมคำสั่ง และเก็บค่าข้อมูลต่างๆ ให้กับไอซี MCS-51 จากตารางที่ 2.1 แสดงขนาดหน่วยความจำไอซี MCS-51 เบอร์ต่างๆ ของบริษัท Atmel

ตารางที่ 2.1 แสดงขนาดหน่วยความจำของไอซี MCS-51 ของบริษัท Atmel

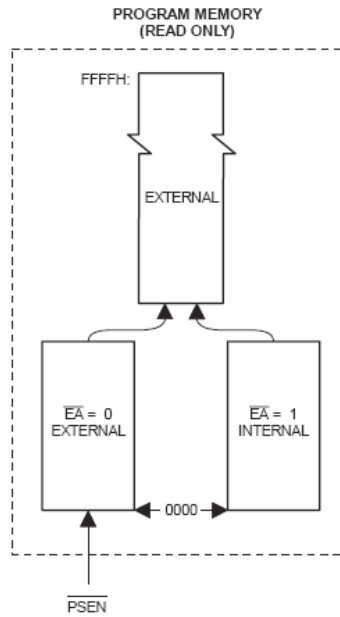
เบอร์ไอซี	หน่วยความจำโปรแกรม	หน่วยความจำข้อมูล	เทคโนโลยีในการผลิต
AT89C2051	2K Flash	128 RAM	HMOC
AT89C51	4K Flash	128 RAM	HMOC
AT89C52	8K Flash	256 RAM	HMOC
AT89C55	20K Flash	256 RAM	HMOC
AT89S8252	8K Flash	256 RAM 2 K EEPROM	HMOC
AT89S53	12K Flash	256 RAM	HMOC
AT98C51RD2	64 K	64K RAM 1K EEPROM	HMOC

(แหล่งอ้างอิง <http://www.atmel.com>)

ไอซี MCS-51 ได้แบ่งหน่วยความจำออกเป็น 2 ชนิด ดังนี้

3.1 หน่วยความจำสำหรับเก็บโปรแกรม (Program Memory)

ทำหน้าที่เก็บรหัสคำสั่ง(Code Memory) เพื่อให้ไมโครคอนโทรลเลอร์ได้ปฏิบัติตามคำสั่งนั้น ไอซี MCS-51 มีหน่วยความจำโปรแกรมภายใน (Internal Program Memory) เป็นแบบ ROM, EPROM และ Flash ตามขนาดต่างๆ ยังสามารถต่อหน่วยความจำภายนอก (External Program Memory) ได้ถึง 64 กิโลไบต์ การเลือกใช้งานหน่วยความจำโปรแกรมที่ต่อภายนอก หรือภายในไอซี กำหนดที่สถานะลอจิกของขา \overline{EA} โครงสร้างของหน่วยความจำโปรแกรม แสดงได้ดังภาพที่ 2.10 การเลือกใช้งานขนาดความจุของหน่วยความจำโปรแกรม ขึ้นอยู่กับจำนวนรหัสคำสั่งที่ใช้งาน ตัวอย่างเช่น การเขียนโปรแกรมควบคุมงานต้องการ เนื้อที่สำหรับการเขียนโปรแกรมจำนวน 3,100 ไบต์ ดังนั้นควรเลือกใช้ไอซีที่มีหน่วยความจำขนาด 4 กิโลไบต์ เช่นเบอร์ AT89C51 และเมื่อมีการรีเซตระบบ ไอซี MCS-51 จะเริ่มทำงาน โดยอ่านรหัสคำสั่ง จากตำแหน่งแอดเดรส ที่ 0000H ของหน่วยความจำเป็นลำดับแรก

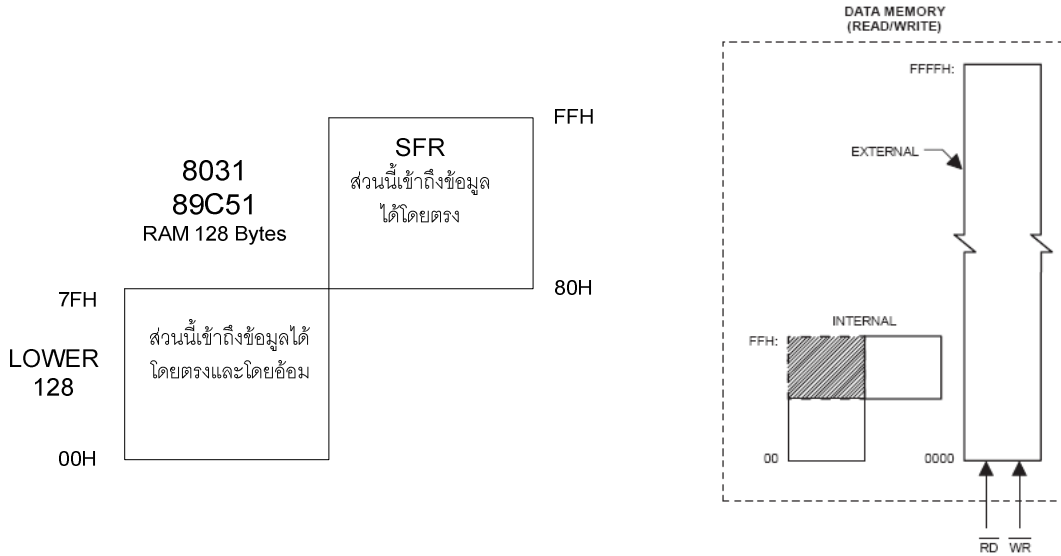


แอดเดรส (Adress)																ข้อมูล (data)								
เลขฐาน16 หลักที่ 4				เลขฐาน16 หลักที่ 3				เลขฐาน16 หลักที่ 2				เลขฐาน16 หลักที่ 1				เลขฐาน16 หลักที่ 2				เลขฐาน16 หลักที่ 1				
A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	D7	D6	D5	D4	D3	D2	D1	D0	
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0									
X	X	X	X	X	X	X	X	← 256 bytes 00H - FFH →								← 8 บิต หรือ 1 ไบต์ →								
X	X	X	X	X	X	X	X	← 1 kbytes 0000H - 03FFH →																
X	X	X	X	X	X	X	X	← 2 kbytes 0000H - 07FFH →																
X	X	X	X	X	X	X	X	← 4 kbytes 0000H - 0FFFH →																
← 64kbytes 0000H - FFFFH →																								
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1									

ภาพที่ 2.10 โครงสร้างหน่วยความจำสำหรับเก็บโปรแกรม

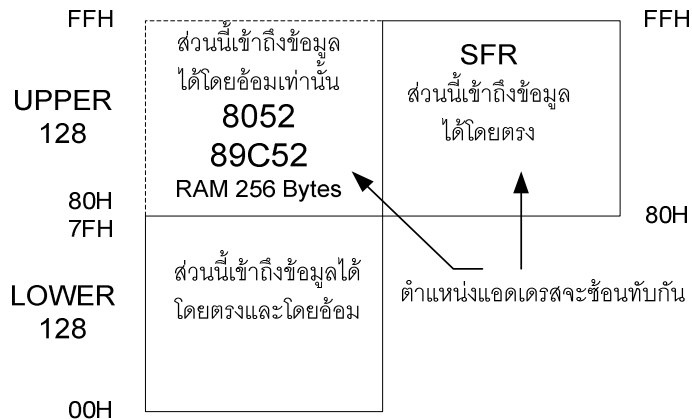
3.2 หน่วยความจำสำหรับเก็บข้อมูล (Data Memory)

ทำหน้าที่เก็บข้อมูลที่ต้องเปลี่ยนแปลงค่าได้ ในไอซี MCS-51 จะมีหน่วยความจำข้อมูลภายใน (Internal Data Memory) มีเนื้อที่ตามเบอร์ไอซีที่เลือก สามารถต่อหน่วยความจำข้อมูลภายนอก (External Data Memory) ได้ถึง 64 กิโลไบต์ ยังแบ่งการใช้งานออกได้เป็น 2 ส่วนคือ หน่วยความจำข้อมูล 128 ไบต์แรก เป็นหน่วยความจำใช้งานทั่วไป (General Ram) อยู่ที่ตำแหน่งแอดเดรส 00H - 7FH และหน่วยความจำ 128 ไบต์ในลำดับถัดไป อยู่ในตำแหน่งแอดเดรส 80H - FFH ซึ่งจะเป็นส่วนของรีจิสเตอร์ใช้งานพิเศษ (Special Function Register) แสดงได้ดังภาพที่ 2.11



ภาพที่ 2.11 หน่วยความจำข้อมูลภายในของไอซีที่มีขนาด 128 ไบต์

ไอซี MCS-51 ที่มีหน่วยความจำข้อมูลภายในมากกว่า 128 ไบต์ ดังนั้นส่วนที่เกิน 128 ไบต์แรก แอดเดรสจะซ้ำ และทับกับหน่วยความจำที่เป็นตำแหน่งของรีจิสเตอร์ใช้งานพิเศษ แสดงได้ดังภาพที่ 2.12 ดังนั้นในการเข้าถึงข้อมูล จะใช้คำสั่งแบบ โดยอ้อมเท่านั้น (Indirect Addressing Mode)

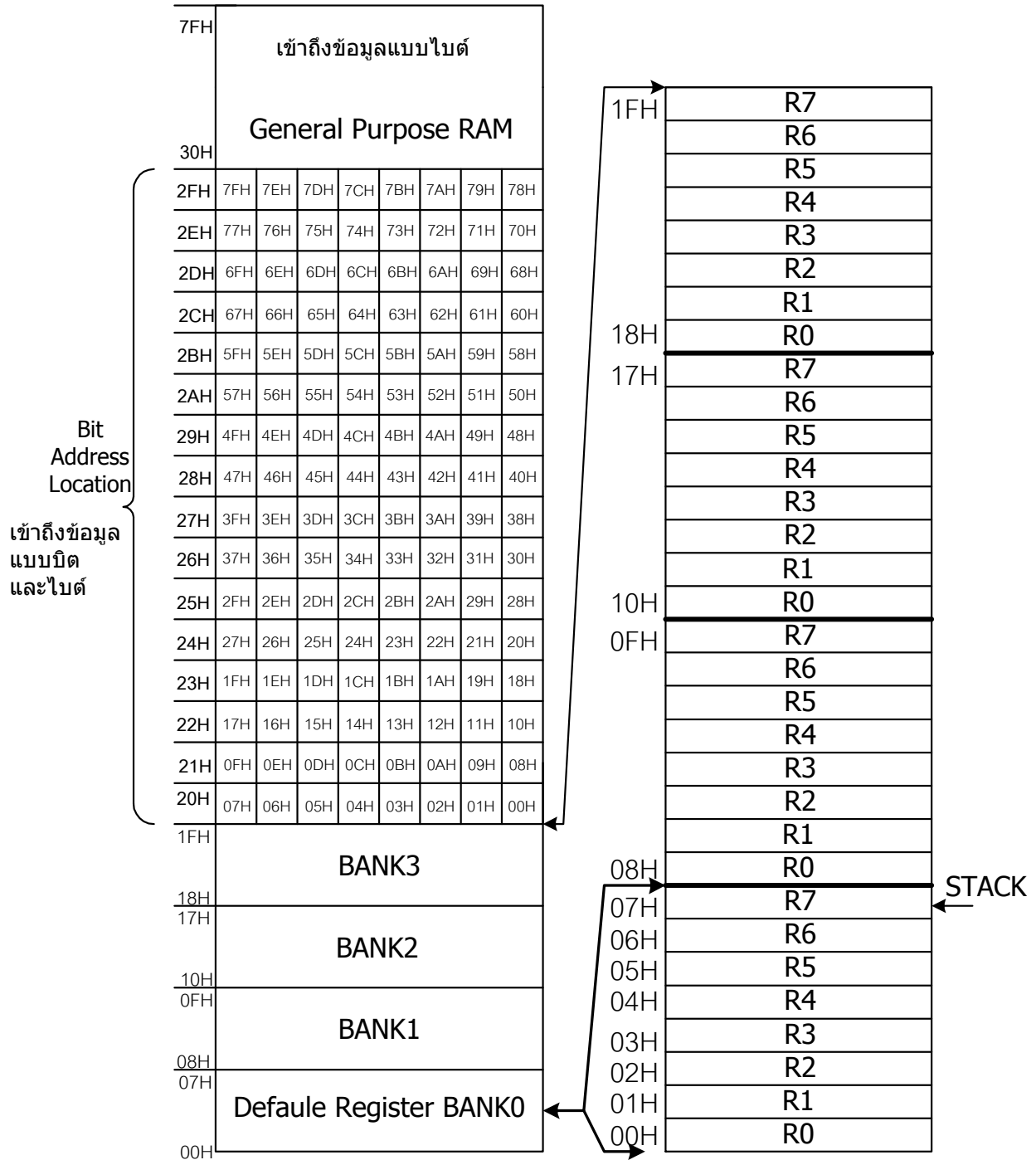


ภาพที่ 2.12 การแบ่งพื้นที่ 256 ไบต์ ของหน่วยความจำข้อมูลภายในไอซี MCS-51

พื้นที่ของหน่วยความจำข้อมูลภายในไอซี ตำแหน่งแอดเดรส 00H - 7FH ยังสามารถแบ่งหน้าที่การใช้งานออกเป็นส่วนย่อยได้อีก 3 ส่วน ดังแสดงในภาพที่ 2.13 มีรายละเอียดดังนี้

3.2.1 พื้นที่ในหน่วยความจำข้อมูลตำแหน่ง 00H-1FH จำนวน 32 ไบต์ยังแบ่งออกเป็น 4 กลุ่ม เรียกว่า แบงก์ (Bank) ในแต่ละแบงก์จะมีขนาด 8 ไบต์ ถูกใช้เป็นที่รีจิสเตอร์ใช้งานทั่วไปโดยรีจิสเตอร์ R0 อยู่

ในตำแหน่งแรกของแต่ละแบงก์ ส่วนรีจิสเตอร์ R7 อยู่ตำแหน่งสุดท้ายของแบงก์ การใช้งานเลือกรีจิสเตอร์ R0-R7 ได้เพียงแบงก์เดียว ซึ่งกำหนดข้อมูลในรีจิสเตอร์ PSW หากไม่ทำการกำหนดค่า หลังจากรีเซตระบบให้กับไอซี MCS-51 จะถูกกำหนดให้เริ่มต้นใช้งานตำแหน่งของแบงก์ 0



ภาพที่ 2.13 ตำแหน่งแอดเดรส 00H-7FH ของหน่วยความจำข้อมูลภายใน

3.2.2 พื้นที่ในหน่วยความจำข้อมูลภายใน ที่ตำแหน่งแอดเดรส 20H - 7FH จำนวน 16 ไบต์ สามารถใช้งานในลักษณะการเข้าข้อมูลแบบไบต์ หรือแบบบิต และอ้างตำแหน่งแบบบิตได้โดยตรง โดยระบุตำแหน่ง หรือชื่อของบิตนั้น จำนวนทั้งหมด 128 บิต ในแต่ละบิตจะมีหมายเลขตำแหน่ง 00H - 7FH ซึ่งตำแหน่งบิต 00H คือข้อมูลของบิตต่ำสุด ในตำแหน่งแอดเดรส 20H หรือเรียกว่า 20H.1 ส่วนตำแหน่งของบิตที่ 7FH คือข้อมูลบิตสูงสุดในตำแหน่งแอดเดรสที่ 7FH หรือ 20H.7 การอ้างตำแหน่งแบบบิตทำให้โปรแกรมทำงานได้รวดเร็วขึ้น ส่วนการเข้าถึงข้อมูลแบบไบต์ คือการเปลี่ยนแปลงหรือกำหนดข้อมูล โดยทำครั้งละ 8 บิตพร้อมกัน ตัวอย่างเช่น มีข้อมูล 1111110B อยู่ในหน่วยความจำข้อมูลแอดเดรสที่ 30H ซึ่งเข้าถึงได้แบบไบต์ หากต้องการเปลี่ยนข้อมูลในตำแหน่งบิตที่ 0 ของแอดเดรส 30H ให้เป็นสถานะ “1” ต้องเปลี่ยนทั้งหมดตั้งแต่ข้อมูลในบิตที่ 0 ถึงบิตที่ 7 จำนวนครั้งละ 8 บิตพร้อมกัน ให้เป็น 11111111 แต่หากข้อมูลที่ต้องการเปลี่ยนอยู่ในตำแหน่งแอดเดรส 20H ซึ่งเป็นหน่วยความจำที่มีความสามารถเข้าถึงข้อมูลแบบบิตได้ให้ระบุตำแหน่งของบิตที่ต้องการเปลี่ยนข้อมูลโดยตรง ที่ตำแหน่ง 00H หรือ 20H.0 โดยใช้คำสั่งเซตบิตให้เป็นสถานะ “1” ที่บิต 0 ของแอดเดรส 20H เพียงแค่บิตเดียว

3.2.3 พื้นที่บริเวณหน่วยความจำข้อมูลตำแหน่งที่ 30H - 7FH เป็นหน่วยความจำใช้งานทั่วไป การติดต่อกับข้อมูลในตำแหน่งต่างๆ ของส่วนนี้ จะอ้างตำแหน่งข้อมูลได้ในลักษณะของแบบไบต์เท่านั้น

4. รีจิสเตอร์ใช้งานพิเศษ

รีจิสเตอร์พิเศษ หรือรีจิสเตอร์เฉพาะ(Special Function Register) ทำหน้าที่ควบคุมการทำงานของไอซี MCS-51อยู่ในหน่วยความจำตำแหน่งแอดเดรสที่ 80H - FFH สามารถเรียกชื่อของรีจิสเตอร์ได้โดยตรง หรือเรียกชื่อตามตำแหน่งแอดเดรสได้ ดังแสดงในภาพที่ 2.14 ประกอบด้วยรีจิสเตอร์ต่างๆ ดังนี้

4.1 รีจิสเตอร์ B

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ตำแหน่ง F0H ใช้กระทำในคำสั่งคูณหรือหารข้อมูลร่วมกับรีจิสเตอร์ A ทำหน้าที่เก็บผลลัพธ์เป็นเศษของการหาร และผลลัพธ์ของค่าผลคูณไปต่อบนนอกจากนั้นยังใช้เก็บข้อมูลทั่วไป กรณีไม่ได้ทำคำสั่งในการคูณหรือหาร

4.2 รีจิสเตอร์ A หรือ ACC (Accumulator)

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสตำแหน่ง E0H เข้าถึงข้อมูลได้ในระดับบิต ใช้กระทำทางคณิตศาสตร์ เช่น การบวก ลบ คูณ หาร และทำหน้าที่เก็บผลลัพธ์ จากการคำนวณทางคณิตศาสตร์ รีจิสเตอร์ A จะถูกนำมาใช้งานมาก ในการเขียนโปรแกรม

4.3 PSW (Program Status Word)

เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่แสดงสถานะการทำงานของโปรแกรมจะใช้เลือกตำแหน่งแบงก์ของรีจิสเตอร์ R0-R7 หลังทำงานในคำสั่งผลการเปลี่ยนแปลงสถานะของบิตต่างๆ นำไปเป็นเงื่อนไข

การกระโดด (Jump) ค่าของบิตต่างๆ ใน PSW สามารถจะเซตบิต หรือเคลียร์บิต ด้วยคำสั่งทางซอฟต์แวร์ได้ รายละเอียดของรีจิสเตอร์ PSW ดังแสดงในภาพที่ 2.15

Byte Address	Bit Address								
FFH									
F0H	F7H	F6H	F5H	F4H	F3H	F2H	F1H	F0H	B
E0H	E7H	E6H	E5H	E4H	E3H	E2H	E1H	E0H	ACC
D0H	CY	AC	F0	RS1	RS0	OV	F1	P	PSW
	D7H	D6H	D5H	D4H	D3H	D2H	D1H	D0H	
B8H	BFH	BEH	BDH	BCH	BBH	BAH	B9H	B8H	IP
B0H	B7H	B6H	B5H	B4H	B3H	B2H	B1H	B0H	P3
A8H	EA	ET2	ES	ET1	EX1	ET0	EX0		IE
	AFH	AEH	ADH	ACH	ABH	AAH	A9H	A8H	
A0H	A7H	A6H	A5H	A4H	A3H	A2H	A1H	A0H	P2
99H	ไม่สามารถเข้าถึงได้ในระดับบิต								SBUF
98H	SM0	SM1	SM2	REN	TB8	TB8	T1	R1	SCON
	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H	
90H	97H	96H	95H	94H	93H	92H	91H	90H	PI
8DH	ไม่สามารถเข้าถึงได้ในระดับบิต								TH1
8CH	ไม่สามารถเข้าถึงได้ในระดับบิต								TH0
8BH	ไม่สามารถเข้าถึงได้ในระดับบิต								TL1
8AH	ไม่สามารถเข้าถึงได้ในระดับบิต								TL0
89H	ไม่สามารถเข้าถึงได้ในระดับบิต								TMOD
88H	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H	TCON
87H	ไม่สามารถเข้าถึงได้ในระดับบิต								PCON
83H	ไม่สามารถเข้าถึงได้ในระดับบิต								DPH
82H	ไม่สามารถเข้าถึงได้ในระดับบิต								DPL
81H	ไม่สามารถเข้าถึงได้ในระดับบิต								SP
80H	87H	86H	85H	84H	83H	82H	81H	80H	P0

ภาพที่ 2.14 แสดงตำแหน่งของรีจิสเตอร์ใช้งานพิเศษ (Special Function Register)

PSW7	PSW6	PSW5	PSW4	PSW3	PSW2	PSW1	PSW0
CY	AC	F0	RS1	RS0	OV	-	P

ภาพที่ 2.15 แสดงตำแหน่งบิตต่างๆ ของรีจิสเตอร์แสดงสถานะการทำงานของโปรแกรม (PSW)

CY (Carry Flag) ทำหน้าที่แสดงสถานะของตัวทด เช่นหากนำเลขขนาด 8 บิต 2 จำนวนบวกกัน ผลบวกที่ได้ จะมีค่ามากกว่า 8 บิต ทำให้สถานะของบิต CY ถูกเซตเป็น “1” แต่ถ้าหากผลบวกไม่เกิน 8 บิต สถานะที่บิต CY จะยังเป็น “0” แสดงได้ดังภาพที่ 2.16 บิตนี้ยังทำหน้าที่เป็นตัวยืมในการลบ ใช้เป็นตัวร่วมกับแอดคิวมูเลเตอร์คำสั่งหมุนบิต และยังใช้ค่าของ CY เป็นเงื่อนไขในการเขียนโปรแกรมกระโดด (Jump) ได้

ตัวทด CY = 1 ผลลัพท์ 1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
	1	0	0	0	0	0	0	0
	1	0	0	0	1	0	0	0
	0	0	0	0	1	0	0	0

ภาพที่ 2.16 แสดงสถานะของ Carry flag ที่เกิดจากการบวกเลข 8 บิต 2 จำนวนเมื่อมีตัวทด

AC (Auxiliary Carry Flag) เป็นแฟลกตัวทศช่วย ในกรณีมีการบวกเลขสองจำนวนและมีการทศระหว่างบิตที่ 3 ไปบิตที่ 4 ทำให้มีการเซตค่าที่บิต AC เป็น “1” แสดงได้ดังภาพที่ 2.17

ตัวทด CY = 1 ตัวทศช่วย AC = "1" ผลลัพท์ 1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
	1	0	0	0	1	0	0	0
	1	0	0	0	1	0	0	0
	0	0	0	1	0	0	0	0

ภาพที่ 2.17 แสดงสถานะของ AC หลังจากบวกเลขสองจำนวน แล้วมีการทศระหว่างบิตที่ 3 ไปบิตที่ 4

F0 (Flag 0) เป็นบิตใช้งานทั่วไป ใช้เป็นแฟลกสถานะ ของโปรแกรมโดยการเซต หรือรีเซต ด้วยคำสั่งทางซอฟต์แวร์ (กันไว้สำหรับผู้ใช้)

RS1-RS0 (Register Bank Select) เป็นบิตกำหนดการเลือกพื้นที่ใช้งาน ของกลุ่มรีจิสเตอร์ R0 - R7 ในแบงก์ต่างๆ โดยการกำหนดสถานะที่บิต RS0 และ RS1 ตามตารางที่ 2.2

ตารางที่ 2.2 แสดงการเลือกใช้งานรีจิสเตอร์แบ่งก็โดยกำหนดสถานะที่บิต RS0 และ RS1

Register	RS0	RS1	ตำแหน่งหน่วยความจำ
BANK 0	0	0	00H - 07H
BANK 1	0	1	08H - 0FH
BANK 2	1	0	10H - 17H
BANK 3	1	1	18H - 1FH

OV (Overflow Flag) เป็นบิตแสดงสถานะค่านับเกิน จะถูกเซตหรือเคลียร์จากการทำงานของคำสั่งทางคณิตศาสตร์ เมื่อมีการทดข้ามจากบิตที่ 6 ไปยังบิตที่ 7 เช่นในการนำเลขสองจำนวนรวมกัน แล้วได้ผลลัพธ์มากกว่า +127 (ฐานสิบ) หรือต่ำกว่า -128 (ฐานสิบ) ในบิตที่ 7 (ซ้ายมือสุด) จะแสดงเป็นบิตสถานะของค่าบวก หรือลบโดยถ้าสถานะเป็น 1 มีค่าบวก ถ้าสถานะเป็น 0 มีค่าลบ ดังนั้นเมื่อมีการเกิดค่า นับเกิน จะทำให้แฟล็ก OV ถูกเซตเป็น “1”

P (Parity Flag) เป็นบิตแสดงสถานะโดยตรวจสอบจำนวนบิตที่เป็น “1” ในข้อมูลของรีจิสเตอร์ ACC บิต P จะถูกเซตเป็น “1” เมื่อสถานะทั้ง 8 บิตมีเลข 1 เป็นจำนวนคี่ (Odd) และจะเคลียร์เป็น “0” เมื่อสถานะของทั้ง 8 บิตในรีจิสเตอร์ ACC มีจำนวนเลข 1 เป็นจำนวนคู่ (Even)

4.4 รีจิสเตอร์ IP (Interrupt Priority Register)

ทำหน้าที่จัดลำดับความสำคัญของการอินเตอร์รัพท์ อยู่ตำแหน่งแอดเดรสที่ B8H มีขนาด 8 บิต เข้าถึงได้ในระดับบิต แสดงได้ดังภาพที่ 2.18 เซตค่าในรีจิสเตอร์ควบคุมในกรณีไม่มีการจัดลำดับความสำคัญ หรือจัดให้มีความสำคัญในระดับเดียวกันไอซี MCS-51 จะจัดตามลำดับความสำคัญการอินเตอร์รัพท์จากสูงไปต่ำดังแสดงในตารางที่ 2.3 เพื่อแก้ปัญหการร้องขออินเตอร์รัพท์ในระดับเดียวกัน และเกิดขึ้นพร้อมกัน สัญญาณในการร้องขออินเตอร์รัพท์ ที่มีระดับความสำคัญสูงกว่า สามารถร้องขอการอินเตอร์รัพท์ซ้อนได้ ในขณะที่ไอซี MCS-51 กำลังปฏิบัติตามโปรแกรมตอบสนองการอินเตอร์รัพท์ของสัญญาณ ที่มีความสำคัญต่ำกว่าได้

IP.7	IP.6	IP.5	IP.4	IP.3	IP.2	IP.1	IP.0
-	-	PT2	PS	PT1	PX1	PT0	PX0

ภาพที่ 2.18 รีจิสเตอร์จัดลำดับความสำคัญการอินเตอร์รัพท์ (IP)

PX0 การกำหนดลำดับความสำคัญของสัญญาณอินเตอร์รัพท์ภายนอก 0

PT0 การกำหนดลำดับความสำคัญของสัญญาณอินเตอร์รัพท์ Timer 0

PX1 การกำหนดลำดับความสำคัญของสัญญาณอินเตอร์รัพท์ภายนอก 1

- PT1 การกำหนดลำดับความสำคัญของสัญญาณอินเทอร์รัพท์ Timer 1
 PS การกำหนดลำดับความสำคัญของสัญญาณอินเทอร์รัพท์พอร์ตอนุกรม
 PT2 การกำหนดลำดับความสำคัญของสัญญาณอินเทอร์รัพท์ Timer2

ตารางที่ 2.3 แสดงตามลำดับความสำคัญจากต่ำ (EXF2) ไปสูง (IE0)

IE0	External Interrupt 0
TF0	Timer 0
IE1	External Interrupt 1
TF1	Timer 1
RI หรือ TI	Serial Port
TF2 หรือ EXF2	Timer 2

4.5 รีจิสเตอร์ IE (Interrupt Enable Register)

ทำหน้าที่ควบคุมการตอบรับการอินเทอร์รัพท์ที่อยู่ในตำแหน่งแอดเดรส A8H มีขนาด 8 บิต และสามารถเข้าถึงในระดับบิตได้ แสดงได้ดังภาพที่ 2.19 ประกอบด้วยบิตต่างๆ ดังนี้

IE.7	IE.6	IE.5	IE.4	IE.3	IE.2	IE.1	IE.0
EA	-	ET2	ES	ET1	EX1	ET0	EX0

ภาพที่ 2.19 รีจิสเตอร์ควบคุมการตอบรับการอินเทอร์รัพท์ (IE)

บิต EA หรือ IE.7 ทำหน้าที่ตอบรับการอินเทอร์รัพท์ทั้งหมดทุกแหล่ง

บิต IE.6 สำรองไว้ใช้ในไอซีเบอร์ใหม่ ๆ

บิต ET2 หรือ IE.5 ทำหน้าที่ควบคุมให้ตอบรับการอินเทอร์รัพท์ TF2 เมื่อเกิดโอเวอร์โฟลว์

บิต ES หรือ IE.4 ทำหน้าที่ควบคุมการตอบรับการอินเทอร์รัพท์จากพอร์ตสื่อสารอนุกรม

บิต ET1 หรือ IE.3 ทำหน้าที่ควบคุมการตอบรับการอินเทอร์รัพท์ TF1 เมื่อเกิดโอเวอร์โฟลว์

บิต EX1 หรือ IE.2 ทำหน้าที่ควบคุมการตอบรับการอินเทอร์รัพท์จากภายนอกที่ขา $\overline{INT1}$

บิต ET0 หรือ IE.1 ทำหน้าที่ควบคุมการตอบรับการอินเทอร์รัพท์ TF0 เมื่อเกิดโอเวอร์โฟลว์

บิต EX0 หรือ IE.0 ทำหน้าที่ควบคุมการตอบรับการอินเทอร์รัพท์จากภายนอก $\overline{INT0}$

* ถ้ากำหนดให้บิตเป็นสถานะลอจิก “1” จะเป็นการตอบรับการอินเทอร์รัพท์

* ถ้ากำหนดให้บิตเป็นสถานะลอจิก “0” จะไม่ตอบรับการอินเทอร์รัพท์

4.6 รีจิสเตอร์ SBUF (Serial data Buffer Register)

เป็นรีจิสเตอร์ขนาด 8 บิตหรือ 1 ไบต์ มีแอดเดรสอยู่ตำแหน่งที่ 99H เข้าถึงข้อมูลแบบไบต์ได้
 อย่างเดียว ทำหน้าที่รับส่งข้อมูลออกไป ทางพอร์ตอนุกรมของ ไอซี MCS-51

4.7 รีจิสเตอร์ SCON (Serial Port Control Register)

ทำหน้าที่ควบคุมการทำงานของพอร์ตอนุกรม การเลือกโหมดการทำงาน และเก็บข้อมูลในบิต
 ที่ 9 (โดยปกติข้อมูลจะมี 8 บิต อยู่ในรีจิสเตอร์ SBUF) ของการรับข้อมูล (RB8) และส่งข้อมูล (TB8) อยู่
 ในตำแหน่งแอดเดรสที่ 98H เข้าถึงข้อมูล แบบไบต์ และแบบบิตได้ แสดงดังภาพที่ 2.20

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

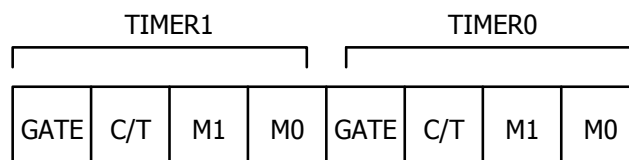
ภาพที่ 2.20 รีจิสเตอร์ควบคุมการทำงานของพอร์ตอนุกรม (SCON)

4.8 รีจิสเตอร์ไทมเมอร์ (Timer) TH1, TH0, TL1, TL0

เป็นรีจิสเตอร์ขนาด 8 บิต มีจำนวน 4 ตัวประกอบด้วยรีจิสเตอร์ TL0 อยู่ที่แอดเดรส 8AH
 รีจิสเตอร์ TH0 อยู่ที่แอดเดรส 8CH รีจิสเตอร์ TL1 อยู่ที่แอดเดรส 8BH และรีจิสเตอร์ TH1 อยู่ที่แอดเดรส
 8DH การใช้งานสามารถใช้งานตัวเดียว 8 บิต หรือใช้งานร่วมกันเป็นคู่ เพื่อให้ได้รีจิสเตอร์ขนาด 16 บิต คือ
 TL0 และ TH0 เพื่อใช้กำหนดค่าการนับของไทมเมอร์ 0 ที่มีขนาด 16 บิต และทำนองเดียวกัน TL1 และ TH1
 รวมกันเพื่อใช้กำหนดค่าการนับของไทมเมอร์ 1 มีขนาด 16 บิตได้

4.9 รีจิสเตอร์ TMOD (Timer/Counter Mode Control Register)

ทำหน้าที่ควบคุมโหมดการทำงาน และควบคุมสัญญาณการทำงานของไทมเมอร์ 0 และไทมเมอร์ 1
 อยู่ตำแหน่งแอดเดรสที่ 89H ไม่สามารถอ้างถึงแอดเดรสในระดับบิตได้ ต้องใช้วิธีการโอนย้ายข้อมูลขนาด
 8 บิตโดยตรง แสดงดังภาพที่ 2.21



ภาพที่ 2.21 รีจิสเตอร์ควบคุมโหมด และควบคุมสัญญาณทำงานของไทมเมอร์ (TMOD)

4.10 รีจิสเตอร์ TCON (Timer Control Register)

ทำหน้าที่ควบคุมสัญญาณการอินเทอร์รัปต์ของไทมเมอร์ และทำหน้าที่ตรวจสอบค่าการนับของไทมเมอร์ เมื่อเกิดค่านับเกิน จะทำการเซตและเคลียร์บิตด้วยวิธีทางซอฟต์แวร์ หรือใช้คำสั่งในการโอนย้ายข้อมูลอยู่ตำแหน่งแอดเดรส 88 H ดังแสดงได้ในภาพที่ 2.22

TCON.7	TCON.6	TCON.5	TCON.4	TCON.3	TCON.2	TCON.1	TCON.0
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

ภาพที่ 2.22 รีจิสเตอร์ควบคุมสัญญาณการอินเทอร์รัปต์ของไทมเมอร์ (TCON)

TF1: TCON.7 (Timer1 Overflow Flag) ทำหน้าที่เป็นแฟล็กแสดงสถานะเป็น “1” เมื่อเกิดค่า นับเกินที่ไทมเมอร์1 และจะกระโดดไปทำโปรแกรมการตอบสนองการอินเทอร์รัปต์ตำแหน่งเริ่มต้นที่ แอดเดรส 001BH ของหน่วยความจำโปรแกรม และบิตนี้จะถูกเคลียร์ให้เป็น “0” อัตโนมัติโดยทาง ฮาร์ดแวร์

TR1: TCON.6 (Timer1 Run Control bit) เป็นบิตควบคุมการทำงานของไทมเมอร์1 ถ้าเซตที่ บิตนี้ให้มีสถานะเป็น “1” จะควบคุมให้ไทมเมอร์1 เริ่มทำงาน และถ้าเคลียร์ เป็นสถานะ “0” ทำให้ไทมเมอร์1 หยุดทำงาน

TF0: TCON.5 (Timer0 Overflow Flag) ทำหน้าที่เป็นแฟล็กแสดงสถานะเป็น “1” เมื่อเกิดค่า นับเกินที่ไทมเมอร์0 และจะกระโดดไปทำโปรแกรมการตอบสนองการอินเทอร์รัปต์ตำแหน่งเริ่มต้นที่ แอดเดรส 001BH ของหน่วยความจำโปรแกรม และบิตนี้จะถูกเคลียร์ให้เป็น “0” อัตโนมัติโดยทาง ฮาร์ดแวร์

TR0: TCON.4 (Timer0 Run Control Bit) เป็นบิตควบคุมการทำงานของไทมเมอร์0 ถ้าเซตที่ บิตนี้มีสถานะเป็น “1” จะควบคุมไทมเมอร์0 ให้เริ่มทำงาน และถ้าเคลียร์ เป็นสถานะ “0” ทำให้ไทมเมอร์0 หยุดทำงาน

IE1: TCON.3 (External Interrupt1 Edge Flag) เป็นแฟลกร็องขอการอินเทอร์รัปต์ภายนอก ของสัญญาณ $\overline{INT1}$ บิตนี้จะถูกเซตด้วยฮาร์ดแวร์ เมื่อมีสัญญาณอินเทอร์รัปต์ที่ขา $\overline{INT1}$ และจะถูกเคลียร์ อัตโนมัติ เมื่อไมโครคอนโทรลเลอร์กระโดดไปทำโปรแกรมตอบสนองการอินเทอร์รัปต์ของสัญญาณ $\overline{INT1}$ ซึ่งอยู่ในตำแหน่งเริ่มต้นที่แอดเดรส 0013H ของหน่วยความจำ โปรแกรม

IT1: TCON.2 (Interrupt1 Type Control Bit) เป็นการเลือกรูปแบบการแอกทีฟของสัญญาณ อินเทอร์รัปต์ จากภายนอกที่ขา $\overline{INT1}$ โดยสามารถเซต หรือเคลียร์ด้วยซอฟต์แวร์ ถ้ามีสถานะ “1” เลือกใช้

สัญญาณอินเทอร์รัปต์ $\overline{\text{INT1}}$ แอคทีฟ ที่การเปลี่ยนจาก “1” ไป “0” (ขอบขาลง) ถ้ามีสถานะเป็น “0” เลือกใช้สัญญาณอินเทอร์รัปต์ $\overline{\text{INT1}}$ แอคทีฟที่ลอจิก “0” (Level)

IE0: TCON.1 (External Interrupt0 Edge Flag) เป็นแฟลกร็องของอินเทอร์รัปต์จากสัญญาณ $\overline{\text{INT0}}$ ถูกเซตด้วยฮาร์ดแวร์ เมื่อมีสัญญาณอินเทอร์รัปต์เข้ามาที่ขา $\overline{\text{INT0}}$ และจะถูกเคลียร์โดยฮาร์ดแวร์เมื่อไมโครคอนโทรลเลอร์ กระโดดไปทำโปรแกรมตอบสนองการอินเทอร์รัปต์ของสัญญาณ $\overline{\text{INT0}}$ ในตำแหน่งแอดเดรส 0003H ของหน่วยความจำโปรแกรม

IT0: TCON.0 (Interrupt0 Type Control bit) เป็นบิตควบคุมการเลือกรูปแบบการแอคทีฟของสัญญาณอินเทอร์รัปต์จากภายนอกที่ขา $\overline{\text{INT0}}$ สามารถเซต หรือเคลียร์ด้วยซอฟต์แวร์ ถ้ามีสถานะลอจิก “1” เลือกใช้สัญญาณอินเทอร์รัปต์ $\overline{\text{INT0}}$ แอคทีฟที่การเปลี่ยนจาก “1” ไป “0” (ขอบขาลง) ถ้ามีสถานะลอจิก “0” เลือกใช้สัญญาณอินเทอร์รัปต์ $\overline{\text{INT0}}$ แอคทีฟที่ลอจิก “0” (Level)

4.11 รีจิสเตอร์ PCON (Power Control)

ทำหน้าที่หยุดการทำงานของไอซี MCS-51 โดยหยุดจ่ายสัญญาณนาฬิกาให้กับระบบ ทำให้ข้อมูลต่างๆ ภายในไอซีไม่มีการเปลี่ยนแปลง นอกจากนั้นยังทำหน้าที่ลดพลังงานที่จ่ายให้กับไอซี MCS-51 ลงด้วยมีแอดเดรสอยู่ตำแหน่งที่ 87H เข้าถึงข้อมูลได้แบบไบต์

4.12 DPTR (Data Pointer)

เป็นตัวชี้ตำแหน่งแอดเดรสของหน่วยความจำโปรแกรม หรือตำแหน่งของอุปกรณ์อินพุตเอาต์พุต ที่ไอซี MCS-51 ต้องการจะติดต่อด้วย และใช้กำหนดตำแหน่งเริ่มต้นของตารางการทำงานแบบ Lookup Table เป็นรีจิสเตอร์ขนาด 16 บิตมีแอดเดรสที่ตำแหน่ง 82H, 83H ประกอบด้วย รีจิสเตอร์ขนาด 8 บิต 2 ตัว คือ DPL และ DPH เลือกการใช้งานในลักษณะ 8 บิต 2 ตัว หรือ 16 บิต 1 ตัวได้

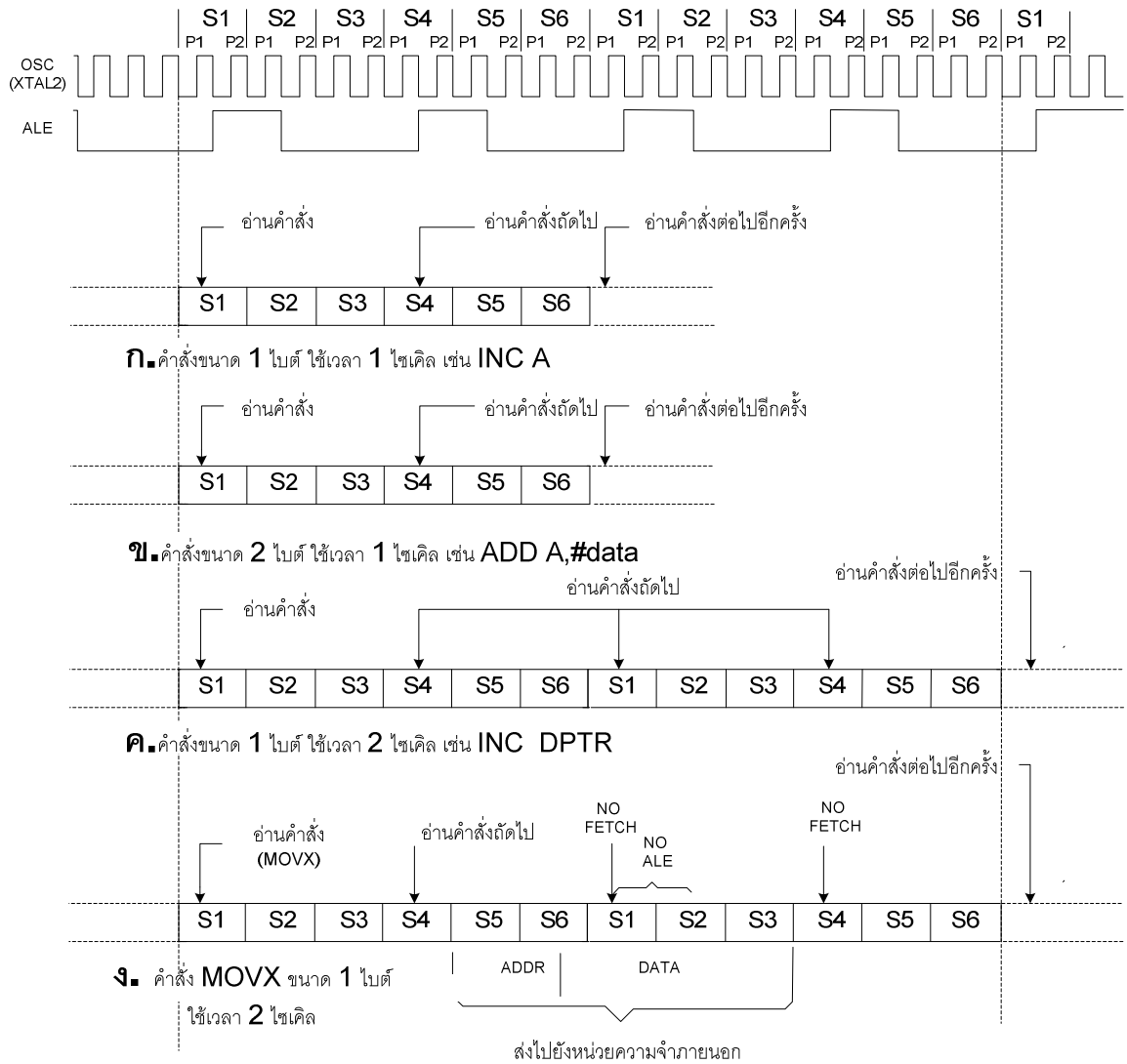
4.13 ตัวชี้สแต็ก SP (Stack Pointer)

เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ตำแหน่ง 81H ใช้เก็บค่าข้อมูลในรีจิสเตอร์ และค่าของโปรแกรมเคาน์เตอร์ ในขณะที่มีการเรียกใช้งานโปรแกรมย่อยหรือโปรแกรมย่อยบริการอินเทอร์รัปต์ หลังจากไอซี MCS-51 กระทำคำสั่งในส่วนของโปรแกรมย่อยเสร็จแล้ว สแต็กจะคืนค่าข้อมูลเดิมให้กับรีจิสเตอร์ และโปรแกรมเคาน์เตอร์ที่เก็บไว้ เมื่อมีการรีเซตระบบตัวชี้สแต็กจะถูกกำหนดให้เริ่มต้นชี้ตำแหน่งแอดเดรส 07H

5. ลักษณะสัญญาณและกระบวนการทำงาน

การทำงานในคำสั่งต่างๆของไอซี MCS-51 ประกอบไปด้วยรอบการทำงาน (Machine Cycle) ซึ่งแต่ละรอบประกอบไปด้วยสถานะเฟตช์ (Fetch) และเอ็กซีคิว (Execute) ในแต่ละคำสั่ง อาจต้องใช้จำนวนรอบการทำงานต่างกันไป การหาค่าเวลาที่ปฏิบัติตามคำสั่งนั้น สามารถคำนวณได้จากผลรวม ของจำนวนรอบการทำงานทั้งหมดได้ ดังนั้นหากกำหนดเวลาเพื่อให้ได้ค่าตามต้องการไอซี MCS-51 จะต้องกระทำ

ตามคำสั่ง เดิมๆ ซ้ำกันเรียกว่า “การหน่วงเวลา” วิธีสร้างโปรแกรมหน่วงเวลา จะกำหนดค่าให้กับรีจิสเตอร์ หรือหน่วยความจำในตำแหน่งหนึ่งๆ ให้ได้จำนวนรอบการทำงานตามที่กำหนด หลังจากนั้นจึงเขียนคำสั่ง เพื่อลดค่ารีจิสเตอร์ หรือหน่วยความจำลงทีละ 1 ค่า แล้วทำการตรวจสอบข้อมูลจนกว่าจะได้ค่าเท่ากับ 0 จึงออกจากโปรแกรมหน่วงเวลาได้



ภาพที่ 2.23 แสดงโครงสร้าง รอบการทำงานของ MCS-51

(แหล่งอ้างอิง http://atmel.com/dyn/products/datasheets.asp?family_id=604)

1 แมชชีนไซเคิล ประกอบด้วย 6 สเตท คือ สเตทที่ 1 ถึงสเตทที่ 6 ในแต่ละสเตทใช้ 2 เฟส (P1, P2) สัญญาณออสซิลเลเตอร์ (Oscillator Period) ดังนั้นใน 1 แมชชีนใช้เวลา 12 คาบสัญญาณออสซิลเลเตอร์ การกำหนดเวลาทำงานของแต่ละคำสั่งใช้เวลาประมวลผล ที่แตกต่างกันโดยนับจำนวนของรอบการทำงาน หรือแมชชีนไซเคิล สามารถคำนวณได้จากสูตร $T = MC \times 12/f_{xtal}$

- เมื่อ T : คือค่าเวลาที่ใช้ในการประมวลผลคำสั่ง
 MC : คือจำนวนแมชชีนไซเคิล
 f-xtal : คือค่าความถี่ของคริสตอลที่ใช้
 12 : คือใน 1 แมชชีนไซเคิล CPU จะใช้ สัญญาณนาฬิกา 12 ลูก

ตัวอย่าง การหาค่าเวลา

ถ้ากำหนดค่าความถี่ของคริสตอลเท่ากับ 12 MHz ดังนั้นใน 1 แมชชีนไซเคิลไอซี MCS-51 จะใช้เวลาเท่ากับ $1 \times (12/12 \times 10^6) = 1 \mu\text{sec}$

ถ้ากำหนดค่าความถี่ของคริสตอลเท่ากับ 11.059 MHz ดังนั้นใน 1 แมชชีนไซเคิลไอซี MCS-51 จะใช้เวลาเท่ากับ $1 \times (12/11.059 \times 10^6) = 1.085 \mu\text{sec}$

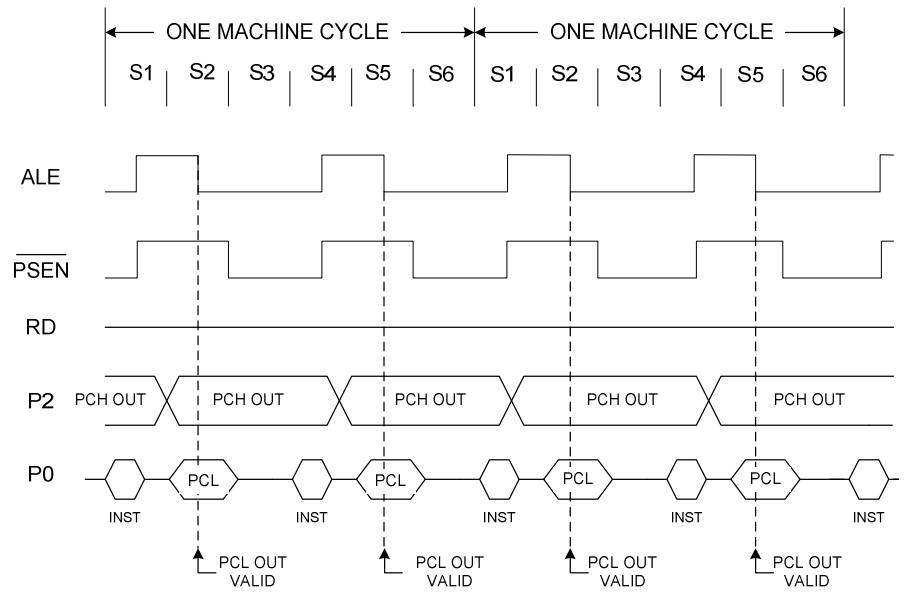
การประมวลผลคำสั่งของซีพียูประกอบด้วย การเฟตซ์ เป็นการเรียกคำสั่งจากหน่วยความจำโปรแกรม หลังจากนั้นทำการแปลรหัสคำสั่งให้เป็นภาษาเครื่อง และการเอ็กซิกิวต์ เป็นการกระทำตามคำสั่ง ที่กำหนด หรือการกระทำตามรหัสคำสั่งของการเฟตซ์ โดยปกติไอซี MCS-51 จะเฟตซ์คำสั่ง 2 ครั้งในแต่ละรอบการทำงาน แม้ว่าคำสั่งที่กำลังทำงานไม่ต้องเฟตซ์เพิ่มเติมอีก แต่ถ้าคำสั่งที่กำลังทำงานไม่ต้องการรหัสคำสั่ง หรือโอเปอเรนด์ของคำสั่งเพิ่ม ซีพียูจะไม่สนใจเฟตซ์คำสั่งที่เกินเข้ามา ค่าในโปรแกรมเคาน์เตอร์จึงไม่ถูกเพิ่มค่า

พิจารณาภาพที่ 2.23 (ก.) และ (ข.) การทำคำสั่งที่ใช้เวลาทำงานใน 1 ไซเคิล เริ่มระหว่างสเตตที่ 1 ของรอบการทำงาน เมื่อรหัสคำสั่งถูกแลตซ์เข้าไปในรีจิสเตอร์คำสั่ง (IR: Instruction Register) การเฟตซ์ครั้งที่ 2 จะเริ่มระหว่างสเตตที่ 4 ของแมชชีนไซเคิลเดียวกัน และทำคำสั่งจนเสร็จสมบูรณ์ ขณะสิ้นสุดสเตตที่ 6 ของ แมชชีนไซเคิลนี้ คำสั่ง MOVX ใช้เวลา 2 แมชชีนไซเคิลในการทำคำสั่ง และไม่มีการเฟตซ์คำสั่งที่เกิดขึ้นในไซเคิลที่ 2 ของคำสั่ง MOVX เป็นเวลาเดียวที่การเฟตซ์คำสั่งได้ถูกข้ามไปลำดับการเฟตซ์และกระทำคำสั่ง MOVX แสดงได้ดังภาพที่ 2.23 (ค.)

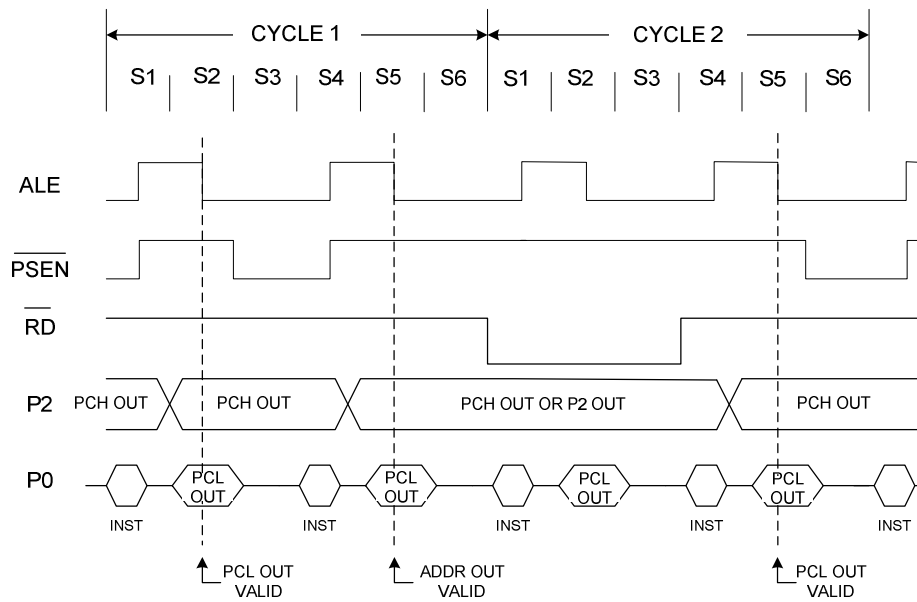
ในกรณีอ่านค่าจากหน่วยความจำข้อมูลภายนอก ขาสัญญาณ \overline{PSEN} จะถูกแอกทีฟ 2 ครั้งต่อ 1 รอบ การทำงาน แสดงได้ในภาพที่ 2.24 (ก) ถ้ามีการติดต่อกับหน่วยความจำภายนอกไอซี MCS-51 สัญญาณที่ขา \overline{PSEN} จำนวน 2 ลูก จะถูกข้ามไปเพราะว่าแอดเดรสบัส และดาต้าบัส กำลังถูกใช้งานในการติดต่อกับหน่วยความจำสำหรับเก็บข้อมูลภายนอกไอซี แสดงได้ในภาพที่ 2.24 (ข)

การติดต่อกับหน่วยความจำข้อมูลภายนอกไอซี (Data Memory Bus Cycle) จะใช้เวลา 2 เท่าของช่วงติดต่อกับหน่วยความจำโปรแกรม (Program Memory Bus Cycle) ภาพที่ 2.24 แสดงการเปรียบเทียบเวลาการทำงานของแอดเดรสที่ส่งออกมาที่พอร์ต 0 และพอร์ต 2 กับ เวลาทำงานของสัญญาณ ALE และ \overline{PSEN} ถูกใช้แลตซ์ค่าแอดเดรสไปตั่ว จากพอร์ต 0 ไปที่วงจรแลตซ์ภายนอก เมื่อซีพียูกำลังกระทำคำสั่งจากหน่วยความจำ สำหรับเก็บโปรแกรมภายในไอซี สัญญาณ \overline{PSEN} จะไม่แอกทีฟ และค่าตำแหน่งหน่วยความจำของโปรแกรม (Program Address) จะไม่ถูกส่งออกมาภายนอก แต่สัญญาณ ALE จะยังคง

แอกทีฟ 2 ครั้งต่อ 1 แมชชีนไซเคิล และสามารถใช้เป็นสัญญาณนาฬิกาที่มีความถี่ก็ได้ แต่สังเกตว่า สัญญาณ ALE จะถูกข้ามไประหว่างการทำคำสั่ง MOVX



ก. ไม่กระทำคำสั่ง MOVX



ข. กระทำคำสั่ง MOVX

ภาพที่ 2.24 แสดงลำดับสัญญาณเมื่อมีการเฟตซ์คำสั่ง จากหน่วยความจำสำหรับ โปรแกรมภายนอก (แหล่งอ้างอิง http://atmel.com/dyn/products/datasheets.asp?family_id=604)

สรุป

ไอซี MCS-51 ได้ถูกผลิตขึ้นเป็นจำนวนมาก ทำให้มีประสิทธิภาพในด้านต่างๆ เพิ่มมากขึ้น เช่น มี วงจรไทมเมอร์และเคาน์เตอร์ขนาด 16 บิต มีหน่วยความจำโปรแกรมภายในแบบแฟลช สามารถโปรแกรม ข้อมูลลงหน่วยความจำโปรแกรมภายในแบบ ISP ทำงานที่ความเร็วได้ถึง 33 MHz มีหน่วยความจำข้อมูล ภายใน 1 กิโลไบต์ มีหน่วยความจำแบบ EEPROM ภายในไอซี มีวงจร PWM มีวงจร WDT ไอซี MCS-51 ประกอบด้วยพอร์ตจำนวน 4 พอร์ตโดยพอร์ต 3 ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษอีกทางหนึ่งด้วย

การรีเซตวงจรทุกอย่างภายในไอซี MCS-51 หรือเริ่มต้นทำงานใหม่ ในกรณีที่ไอซีมีการทำงานผิดปกติ อนุญาตรีเซตต้องให้สถานะอยู่ในระดับลอจิก “1” (High) อย่างน้อย 2 แมกซ์ซีซีเคล

สถาปัตยกรรมภายในของไมโครคอนโทรลเลอร์ MCS-51 ใช้เทคโนโลยีการผลิตแบบ NMOS และ แบบ HMOS ประกอบด้วย วงจรควบคุมต่างๆ หน่วยความจำข้อมูล หน่วยความจำโปรแกรมภายใน หน่วย คณิตศาสตร์และลอจิก รีจิสเตอร์ วงจรออสซิลเลเตอร์ และพอร์ตที่ใช้ติดต่อกับอุปกรณ์ภายนอก ซึ่งแต่ละ ส่วนจะถูกเชื่อมต่อกันด้วย บัสข้อมูล และบัสแอดเดรส

หน่วยความจำทำหน้าที่เก็บโปรแกรมคำสั่ง และค่าข้อมูลต่างๆ ให้กับไอซี MCS-51 การจัด หน่วยความจำ จะแบ่งออกเป็น 2 ชนิดคือหน่วยความจำสำหรับเก็บโปรแกรม และหน่วยความจำสำหรับ เก็บข้อมูล

หน่วยความจำสำหรับเก็บข้อมูลภายในแบ่งพื้นที่เป็น 3 ส่วน (1) เป็นหน่วยความจำข้อมูลตำแหน่ง 00H-1FH จำนวน 32 ไบต์ แบ่งออกเป็น 4 แบนก์ แต่ละแบนก์มี 8 ไบต์ (2) พื้นที่หน่วยความจำข้อมูลภายใน ตำแหน่งแอดเดรสที่ 20H-2FH จำนวน 16 ไบต์ เป็นการเข้าถึงข้อมูลแบบไบต์หรือแบบบิต (3) พื้นที่บริเวณ หน่วยความจำข้อมูลในตำแหน่งที่ 30H-7FH อ้างตำแหน่งข้อมูลได้ในลักษณะของแบบไบต์เท่านั้น

รีจิสเตอร์พิเศษ ทำหน้าที่ควบคุมการทำงานของไอซี MCS-51 อยู่ในหน่วยความจำตำแหน่ง แอดเดรสที่ 80H-FFH การใช้งานสามารถใช้ชื่อของรีจิสเตอร์ หรือตำแหน่งแอดเดรสก็ได้

การทำงานคำสั่งต่างๆ ของไอซี MCS-51 ประกอบด้วย รอบการทำงานหลายรอบ ในแต่ละรอบ ประกอบไปด้วยสถานะเฟตซ์ และเอ็กซีคิวต์ การทำงานของแต่ละคำสั่งจะใช้เวลาประมวลผลที่แตกต่างกัน โดยนับจำนวนของรอบการทำงานหรือค่าแมกซ์ซีซีเคล