

หน่วยที่ 5 การวัดและทดสอบวงจรใช้งานของไอซี MCS-51

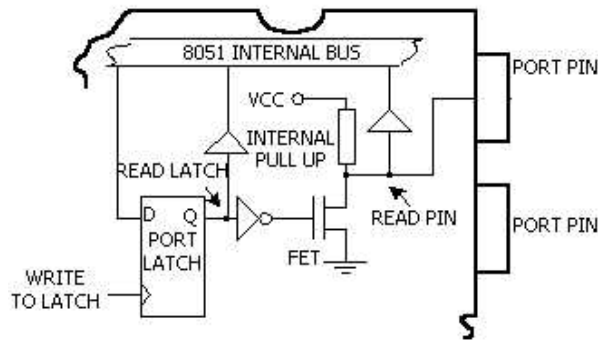
อดิศักดิ์ ชินะวงศ์

เอกสารประกอบการเรียนวิชาไมโครคอนโทรลเลอร์

เผยแพร่ที่ www.Adisak51.com

1. การติดต่อกับอุปกรณ์ภายนอกของไอซี MCS-51

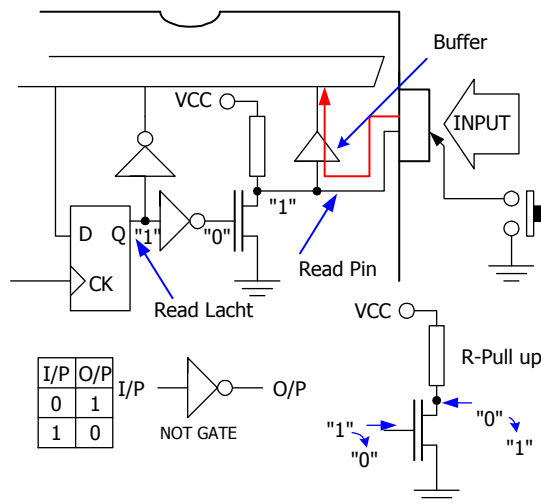
การให้ไอซี MCS-51ติดต่อกับอุปกรณ์ภายนอก เช่น การรับข้อมูลจากอุปกรณ์ภายนอก (Input) หรือการส่งข้อมูลให้กับอุปกรณ์ภายนอก (Output) ต้องติดต่อผ่านพอร์ต (Port) หรือกล่าวได้ว่าพอร์ตคือช่องทางสำหรับโอนย้ายข้อมูลระหว่างไอซี MCS-51กับอุปกรณ์ภายนอก โครงสร้างแสดงดังภาพที่ 5.1



ภาพที่ 5.1 แสดงโครงสร้างภายในของพอร์ต

1.1 การใช้งานไอซี MCS-51 ให้เป็นพอร์ตอินพุต (Input Port)

การกำหนดให้พอร์ตหรือบิตใดๆ ของไอซี MCS-51 เป็นอินพุต แสดงดังภาพที่ 5.2



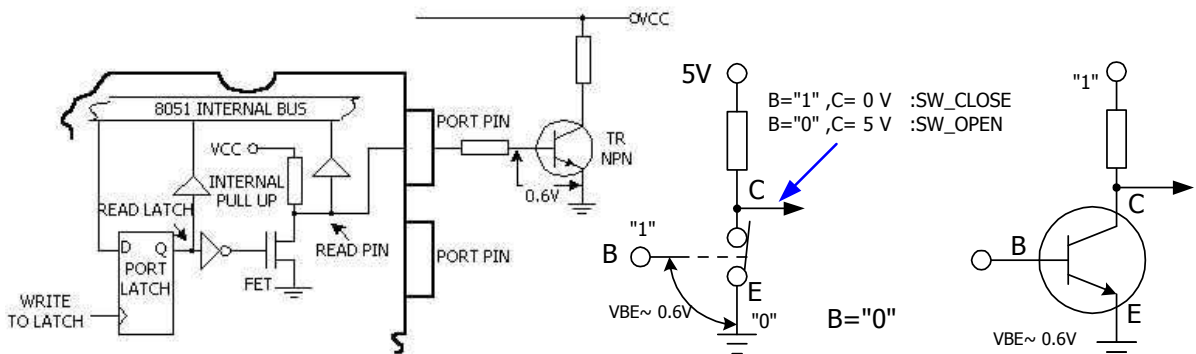
ภาพที่ 5.2 แสดงการกำหนดเป็นพอร์ตอินพุตโดยให้สถานะเป็น "1" ตำแหน่งบิตที่ต้องการ

เริ่มต้นด้วยการเขียนข้อมูลให้กับพอร์ตหรือบิตนั้น เป็นสถานะลอจิก "1" แล้วส่งไปแต่ละบิตของพอร์ตที่ต้องการใช้งาน เป็นพอร์ตอินพุต เช่นคำสั่ง MOV P3, #0FFH หรือคำสั่ง SETB P1.5 ทำให้วงจร

ส่วนของการคงสถานะข้อมูล หรือวงจรถัด (Latch) ที่สร้างมาจากวงจรถัด ฟลิป-ฟลอป (D flip-flop) ให้เอาต์พุต Q มีสถานะลอจิกเป็น “1” ผ่านไปที่วงจรถัดกลับสัญญาณ (Inverter) เพื่อหยุดการทำงานของเฟต (FET) ทำให้ขาสัญญาณของพอร์ต ถูกเชื่อมต่อเข้ากับวงจรถัดขึ้น ที่มีตัวต้านทานภายใน (Internal Pull up) โดยตรง ทำให้ขาพอร์ตมีสถานะลอจิกเป็น “1” และสามารถรับสัญญาณลอจิก “0” จากอุปกรณ์ภายนอกได้ ข้อมูลจากอุปกรณ์ภายนอกที่ต่อ ถูกส่งเข้ามาเก็บไว้ในวงจรถัดเฟลอร์ภายในพอร์ต เพื่อรอให้ซีพียูอ่านค่าเข้าไป ดังนั้นอุปกรณ์ภายนอกที่นำมาเชื่อมต่อกับพอร์ตอินพุต จึงกำหนดให้ทำงานในสถานะลอจิก “0”

1.2 การอ่านค่าลอจิกจากพอร์ต

การอ่านค่าลอจิกจากพอร์ตทำได้ 2 ลักษณะ คือ การอ่านค่าจากขาพอร์ตโดยตรง(Read Pin) และการอ่านค่าจากวงจรถัดของแต่ละพอร์ต (Read Latch) ในกรณีที่พอร์ตต่อกับขาเบสของทรานซิสเตอร์ชนิด NPN เพื่อให้ทำหน้าที่เป็นทรานซิสเตอร์สวิตช์ ขาเบสของทรานซิสเตอร์จึงต่อลงกราวด์ แสดงดังภาพที่ 5.3 ถ้าหากมีการส่งข้อมูลให้เป็นลอจิก “1” ไปยังขาเบสของทรานซิสเตอร์ ทำให้ทรานซิสเตอร์นำกระแส ส่งผลให้สถานะลอจิกที่ขาพอร์ตเป็นลอจิก “0” เพราะ เมื่อทรานซิสเตอร์นำกระแส จึงเสมือนว่าขาพอร์ตนั้นถูกต่อลงกราวด์ (VBE=0.6 V) ดังนั้น ถ้าหากอ่านค่าลอจิกที่ขาพอร์ตโดยตรง (Read Pin) ได้ผลตรงกันข้ามกับค่าลอจิกที่ส่งออกมา แต่ถ้าอ่านค่าลอจิกที่วงจรถัด (Read Latch) ได้สถานะลอจิกตรงกับค่าที่ส่งออกพอร์ตจริง ด้วยเหตุผลนี้การอ่านค่าลอจิกจากพอร์ตจึงต้องเลือกวิธีการให้เหมาะสมกับอุปกรณ์ที่นำมาต่อด้วย



ภาพที่ 5.3 แสดงสถานะของทรานซิสเตอร์ชนิด NPN เมื่อนำมาต่อเป็นทรานซิสเตอร์สวิตช์

1.3 การใช้งานไอซี MCS-51ให้เป็นพอร์ตเอาต์พุต (Output Port)

พอร์ตเอาต์พุต สามารถส่งข้อมูลเป็นลอจิก ที่ต้องการออกไปได้โดยตรง เช่นถ้าหากต้องการส่งข้อมูลเป็นสถานะลอจิก “0” ออกไปทางพอร์ต P1 ทั้ง 8 บิตเขียนโดยใช้คำสั่ง MOV P1,#00H ทำให้เอาต์พุตของวงจรถัดเป็นสถานะลอจิก “0” ซึ่งส่งไปที่วงจรถัดกลับสัญญาณ ทำให้มีสถานะลอจิกเป็น “1” แล้ว

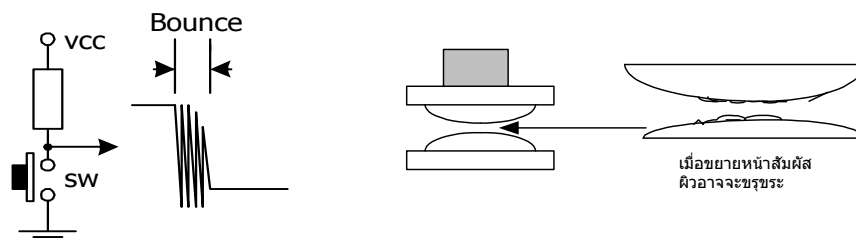
จึงส่งไปขับเฟต (FET) ให้ทำงาน ส่งผลให้ตำแหน่งของพอร์ตที่กำหนดทำงานเป็นสถานะลอจิก “0” และ ถ้าต้องการส่งข้อมูลลอจิก “1” ออกไป สามารถเขียนคำสั่งให้ส่งข้อมูลเป็น “1” ไปยังวงจรถอดสัญญาณการทำงาน เป็นผลให้ขาของพอร์ตเชื่อมต่อกับวงจรพูลอัพภายใน เป็นสถานะลอจิก “1” ที่ตำแหน่งขาพอร์ตนั้นๆ ซึ่งคล้ายกับการกำหนดให้เป็นขาอินพุต เพียงแต่แตกต่างกันที่กระบวนการเคลื่อนย้ายข้อมูล โดยถ้าเป็นอินพุต มีสัญญาณมาอ่านข้อมูลที่บัฟเฟอร์ แต่หากเป็นเอาต์พุตจะไม่มี การอ่านข้อมูลที่บัฟเฟอร์แต่อย่างใด ยกเว้นในกรณีที่ต้องการตรวจสอบข้อมูลที่ส่งออกมาทางเอาต์พุตเท่านั้น

เมื่อให้เอาต์พุตของพอร์ต มีสัญญาณออกเป็นลอจิก “0” และรับกระแสเข้ามาเรียกว่ากระแสซิงค์ (I_{OL} : Sink Current) ที่ตำแหน่งของบิต เมื่อให้เอาต์พุต มีสัญญาณออกมาเป็นลอจิก “1” มีกระแสจ่ายออกมาเรียกว่ากระแสซอร์ส (I_{OH} :Source Current) ในแต่ละบิตของพอร์ตมีความสามารถในการรับกระแส ซิงค์ได้สูงสุด 20 mA และทุกขารวมกันในแต่ละพอร์ต กรณีใช้งานพร้อมกัน สามารถรับกระแสเข้าได้รวมกันสูงสุดไม่เกิน 80 mA ดังนั้นการใช้งานให้ทุกขาเป็นพอร์ตเอาต์พุต เพื่อไม่ให้เกิดปัญหาเกี่ยวกับความสามารถในการรับกระแสเข้า จึงควรต่อวงจรบัฟเฟอร์ ทางเอาต์พุตเพื่อช่วยป้องกันไอซีเสียหาย จากการรับกระแสเข้าเกินกำหนด

1.4 สัญญาณอินพุต

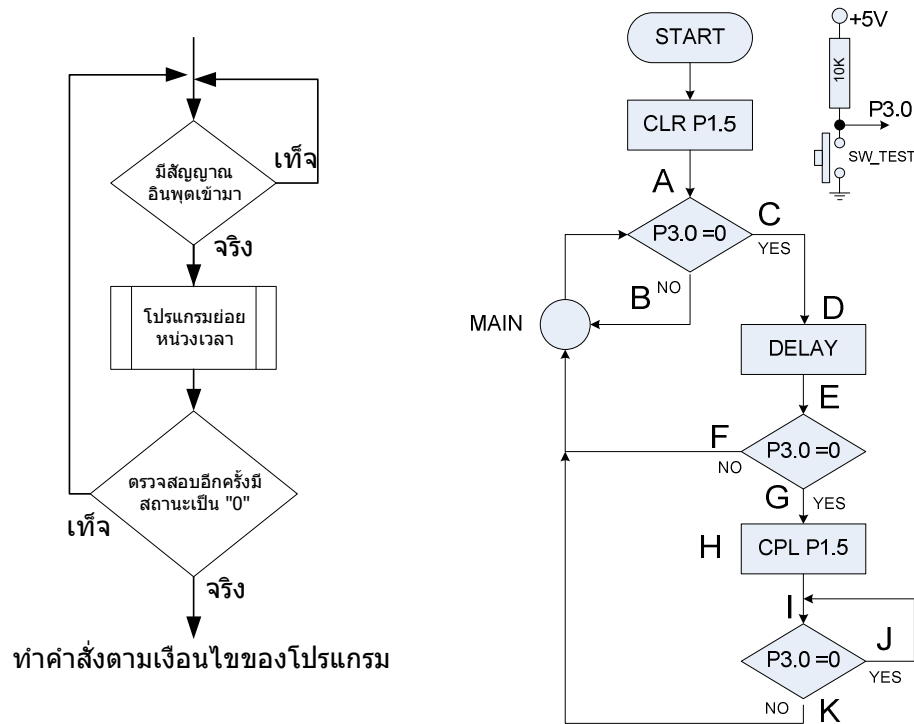
การประยุกต์ใช้งานอินพุต อาจรับสัญญาณข้อมูลมาจากหลายแหล่ง หรืออาจถูกแปลงสัญญาณมาจากแหล่งอุปกรณ์อื่นๆเช่น คีย์สวิตช์ อุปกรณ์เซนเซอร์ต่างๆ หรือสัญญาณในรูปแบบต่างๆ อินพุตจากสวิตช์ แบ่งได้หลายประเภทที่นำมาใช้งาน เช่น สวิตช์แบบโยก, สวิตช์แบบลิ้มิต, สวิตช์แบบสไลด์, สวิตช์แบบกด ฯลฯ ดังนั้นวิธีการป้องกันข้อผิดพลาด และการจำกัดเหตุการณ์ต่างๆที่อาจเกิดขึ้นในการใช้งานหลายๆ กรณี เช่น

- 1.4.1 สัญญาณการรบกวนที่เกิดจากหน้าสัมผัสขณะมีการกดสวิตช์
- 1.4.2 สัญญาณรบกวนจากภายนอกที่เกิดขึ้นในระยะสั้นๆ
- 1.4.3 การกดสวิตช์ตัวเดิมค้างไว้โดยไม่ปล่อยมือ
- 1.4.4 การกดสวิตช์หลายๆตัวพร้อมกัน
- 1.4.5 เวลาที่ใช้ในการกดสวิตช์เร็วหรือช้าเกินกำหนด



ภาพที่ 5.4 สัญญาณรบกวนที่เกิดจากหน้าสัมผัสขณะกดสวิตช์

สัญญาณรบกวนที่เกิดขึ้น จากการสั้นของหน้าสัมผัสสวิตช์ขณะทำการกดสวิตช์ (Bounce) ทำให้เกิดการแกว่งของสัญญาณ แสดงดังภาพที่ 5.4 จึงเป็นปัญหาที่ต้องแก้ไข โดยใช้โปรแกรม หรือใช้วงจรภายนอกป้องกัน ดังนั้นถ้าต้องการให้ไอซี MCS-51 สามารถอ่านค่าได้อย่างถูกต้อง หลังจากกดสวิตช์เพียงครั้งเดียว ต้องแก้ไขปัญหาจากสัญญาณรบกวน โดยใช้วิธีการตรวจสอบการกดสวิตช์ในแต่ละครั้งว่า มีการกดจริงหรือไม่ เขียนเป็นผังงานได้แสดงดังภาพที่ 5.5 ถ้าหากมีการกดสวิตช์ หรือมีสัญญาณใดๆเกิดขึ้น ส่งผลให้มีการเปลี่ยนสถานะลอจิกที่อินพุต ลำดับแรกเรียกโปรแกรมย่อยหน่วงเวลา (Delay Time) ในช่วงเวลาหนึ่ง และหลังจากนั้นจึงออกจากโปรแกรมหน่วงเวลา แล้วทำการอ่านค่าสถานะลอจิกที่อินพุตเดิมอีกครั้งหนึ่ง เพื่อตรวจสอบการเปลี่ยนสถานะลอจิกที่อินพุตจริงหรือไม่ หากตรวจสอบสถานะลอจิกที่อินพุตเดิมแล้วปรากฏมีการเปลี่ยนสถานะของลอจิกที่เกิดขึ้นจากการกดสวิตช์จริง ทำตามเงื่อนไขที่ได้กำหนดไว้ในโปรแกรม แต่หากตรวจสอบสถานะลอจิกที่อินพุตแล้ว ปรากฏยังเป็นสถานะลอจิกเดิมก่อนการกดสวิตช์ ซึ่งหมายความว่าสัญญาณที่รับเข้ามาทางอินพุต อาจเกิดจากสัญญาณการรบกวนในช่วงเวลานั้นๆ



ภาพที่ 5.5 แสดงผังงานการตรวจสอบสัญญาณอินพุต

1.5 โปรแกรมตรวจสอบการกดสวิตช์

การทำงานของผังงานในภาพที่ 5.5 เริ่มตรวจสอบการกดที่สวิตช์ทดสอบ (A) ซึ่งได้ต่ออยู่กับ บิต P3.0 ถ้าไม่มีการกดสวิตช์จะวน(B) ตรวจสอบการกดที่สวิตช์ จนกว่าสวิตช์ถูกกด (C) โปรแกรมจะตรวจสอบสัญญาณจากการเปลี่ยนสถานะลอจิกจากเหตุการณ์ต่างๆ ถ้ามีจะกระโดดไปที่โปรแกรมย่อยหน่วง

เวลา (D) เพื่อหน่วงเวลารอในระยะหนึ่ง เพื่อแก้ปัญหาการรบกวนจากหน้าสัมผัสของสวิตช์ แล้วกลับมาตรวจสอบสถานะ(E) ของบิต P3.0 อีกครั้ง ถ้าบิต P3.0 อยู่ในสถานะ "1" หมายถึงสัญญาณที่เข้ามาอาจเป็นสัญญาณใดๆที่เกิดขึ้นในช่วงเวลาที่น้อยกว่าเวลาที่หน่วงไว้ โปรแกรมจะวนกลับไป (F) ตรวจสอบการกดสวิตช์ใหม่อีกครั้ง แต่ถ้าหากตรวจสอบที่สวิตช์ ยังถูกกด (G) ค้าง หมายถึงมีการกดสวิตช์จริง สถานะลอจิกที่ P3.0 เป็น "0" โปรแกรมจะกำหนดให้ LED แสดงผลตรงข้ามกับสถานะเดิม (H) และทำการตรวจสอบที่ P3.0 อีกครั้ง (I) โดยสังเกตว่ามีการปล่อยมือจากการกดแล้วหรือยัง ถ้ายังจะให้โปรแกรมวนอยู่ที่คำสั่งเดิม (J) ถ้าหาก P3.0 เป็น "1" (K) หมายถึงได้ปล่อยมือจากสวิตช์แล้ว โปรแกรมจะกระโดดกลับไปเริ่มต้นใหม่ ถ้าใช้คำสั่งนี้จะทำให้ P1.5 กลับสถานะตลอด จนกว่ามีการปล่อยมือจากสวิตช์ หรือเป็นการทำงาน เป็นฟลิปฟล็อป จากผังงานเขียนเป็นโปรแกรมได้ดังนี้

```

                ORG    0000H
LED_TEST      BIT    P1.5          ; กำหนดค่า LED_TEST มีค่าแทน P1.5
SW_TEST       BIT    P3.0         ; SW_TEST มีค่าแทน P3.0
                SETB   SW_TEST     ; กำหนดให้เป็นอินพุต
                CLR    LED_TEST    ; เคลียร์ LED
MAIN:         JB     SW_TEST, $     ; P3.0 = "1" หรือไม่ ถ้าเท่าไปที่เลเบลชื่อ MAIN
                ACALL  DELAY_D     ; เรียก DELAY_D
                JB     SW_TEST, MAIN ; ตรวจสอบบิต P3.0 อีกครั้งถ้าเป็นจริงให้ไปที่ MAIN
                CPL    LED_TEST    ; กลับสถานะของบิต P1.5
                JNB   SW_TEST, $    ; ตรวจสอบการปล่อยคีย์ถ้าไม่ปล่อยให้วนที่เดิม
                SJMP  MAIN         ; กระโดดกลับไป ที่ เลเบล MAIN หากมีการปล่อยคีย์แล้ว
DELAY:        MOV    R1, #90H     ; โปรแกรมย่นหน่วงเวลา
DELAY1:       MOV    R2, #0FF
                DJNZ  R2, $
                DJNZ  R1, DELAY1
                RET
                END

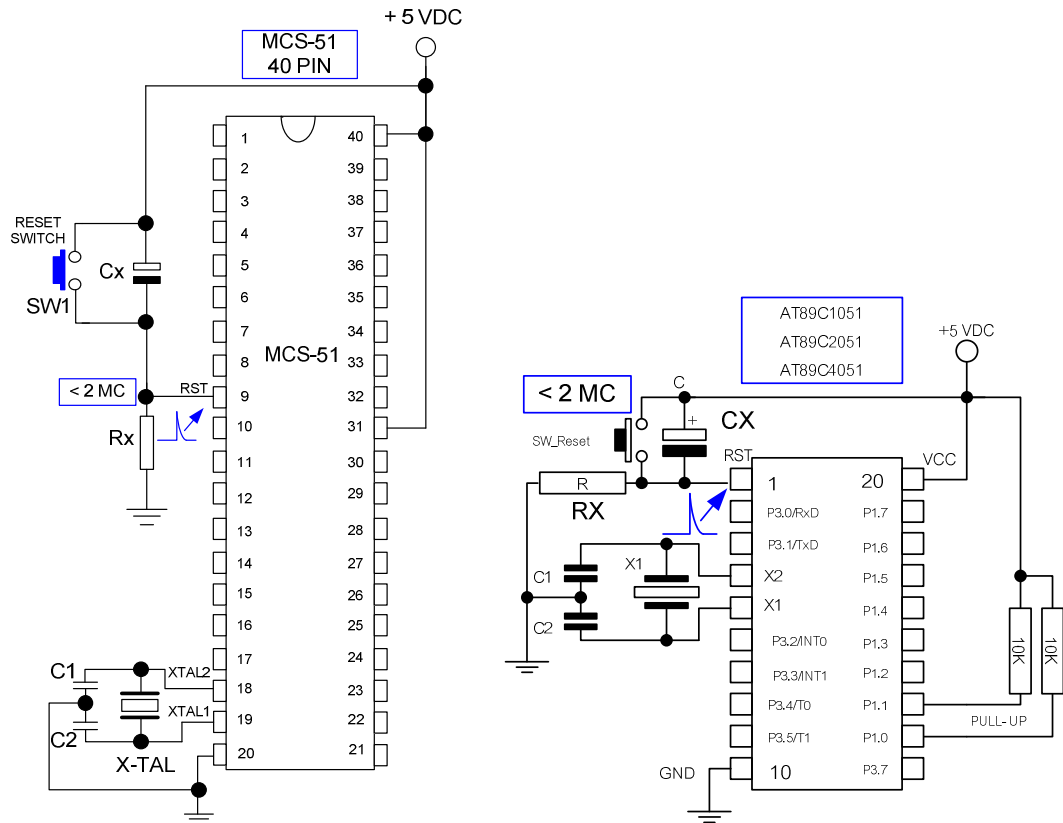
```

2. วงจรใช้งานของไอซี MCS-51

การกำหนดจุดประสงค์ของวงจรที่ต้องการสร้าง โดยใช้ไอซี MCS-51 ทำหน้าที่ควบคุมการทำงาน และกำหนดเงื่อนไขของเหตุการณ์ที่เกิดขึ้นในเวลานั้น ควรคำนึงถึงอุปกรณ์ที่เลือกนำมาต่อใช้งาน เช่น คุณสมบัติที่มีของไอซี MCS-51แต่ละเบอร์ จำนวนของพอร์ตที่ต้องนำไปใช้งาน ฯลฯ

2.1 การต่อวงจรพื้นฐานของไอซี MCS-51

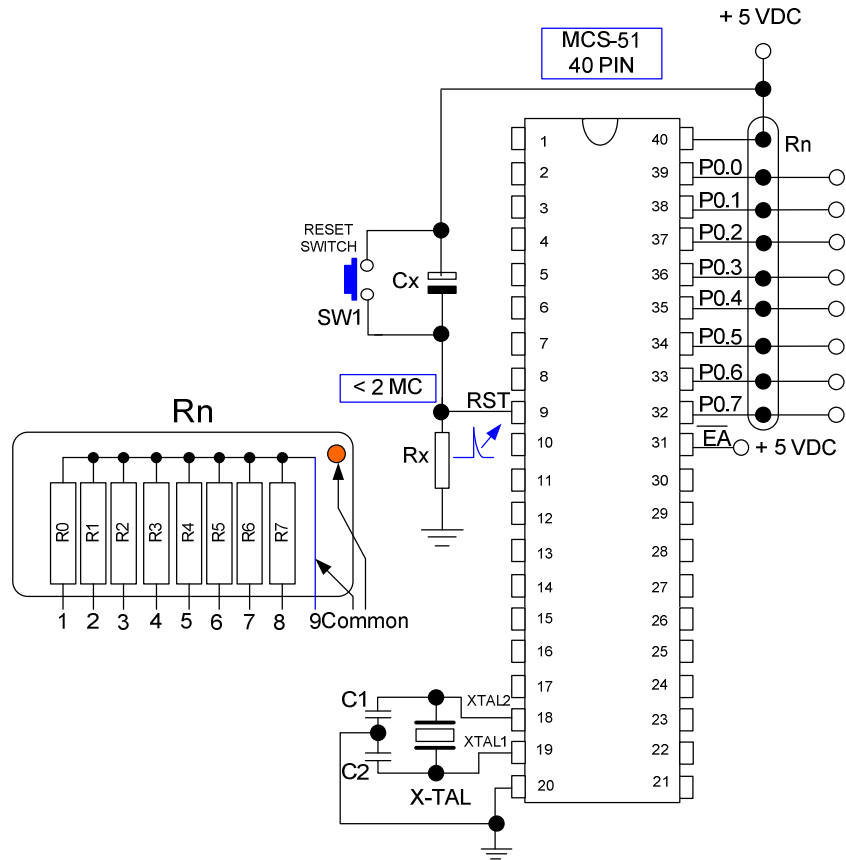
การต่อวงจรโดยใช้ไอซี MCS-51 ขนาด 20 ขา และ 40 ขา ที่มีหน่วยความจำโปรแกรมภายใน ตัวอย่างไอซีที่มีขนาด 20 ขา เช่นเบอร์ AT89C1051, AT89C2051 หรือ AT89C4051 ส่วนไอซีขนาด 40 ขา สามารถใช้ไอซีไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีหน่วยความจำโปรแกรมภายใน (ไอซี MCS-51 ในปัจจุบันจะผลิตเป็นชนิดที่มีหน่วยความจำภายใน) โดยเลือกขนาดของหน่วยความจำได้ตามคุณสมบัติของ ไอซี ดังนั้นขา \overline{EA} จึงต่อกับแหล่งจ่ายไฟ +VCC เพื่อมีสถานะลอจิก “1” เป็นการเลือกใช้หน่วยความจำภายใน แสดงดังภาพที่ 5.6



ภาพที่ 5.6 แสดงวงจรใช้งานของไอซี MCS-51ขนาด 40 ขา และ 20 ขา

ตำแหน่งพอร์ต P1.0 และ P1.1 ของไอซี MCS-51 ขนาด 20 ขา จะทำหน้าที่เป็นอินพุตของวงจรเปรียบเทียบ แบบอนาล็อก ดังนั้นจึงไม่มีตัวต้านทานพูลอัปอยู่ภายใน เมื่อใช้เป็นอินพุตพอร์ต หรือเอาต์พุตพอร์ต ต้องต่อตัวต้านทานภายนอก เพื่อทำหน้าที่ยกระดับสัญญาณ (Pull - Up) ของขาพอร์ตให้สถานะลอจิกสูง และทำหน้าที่จ่ายกระแสเมื่อมีสถานะต่ำจากภายนอก พอร์ต 0 ของไอซี MCS-51 แบบ 40 ขา ไม่มีตัวต้านทานพูลอัปอยู่ภายใน การใช้งานต้องต่อเพิ่มจากภายนอก สามารถใช้ตัวต้านทาน Rn (Resistor Network) ตัวต้านทานที่มีหลาย ๆ ตัวในแพ็คเกจ (Packet) เดียวกัน โดยมีขาต่อรวมอยู่ 1 ขา ส่วนที่เหลือมีจำนวนเท่ากับตัวต้านทานที่นำมาต่อ แสดงดังภาพที่ 5.7

ไอซีไมโครคอนโทรลเลอร์รับสัญญาณนาฬิกาจาก X-TAL, C1 และ C2 ซึ่งตัวเก็บประจุ C1, C2 แบบเซรามิกมีค่าอยู่ระหว่าง 20 pF - 40 pF ความถี่ของคริสตัลระบุไว้ที่ด้านข้าง โดยใช้ไม่เกินค่าที่ระบุไว้ในคุณสมบัติของไอซีแต่ละตัว (ข้อมูลจาก Data Sheet) วงจร SW1, Cx และ Rx ทำหน้าที่กำหนดคาบเวลาเพื่อสร้างสัญญาณการรีเซ็ตของไอซี MCS-51 ที่สถานะลอจิก "1" ในช่วงเวลาอย่างน้อย 2 แมกซ์อินไซเคิล



ภาพที่ 5.7 แสดงการต่อวงจรใช้งานพื้นฐานของไอซี MCS-51

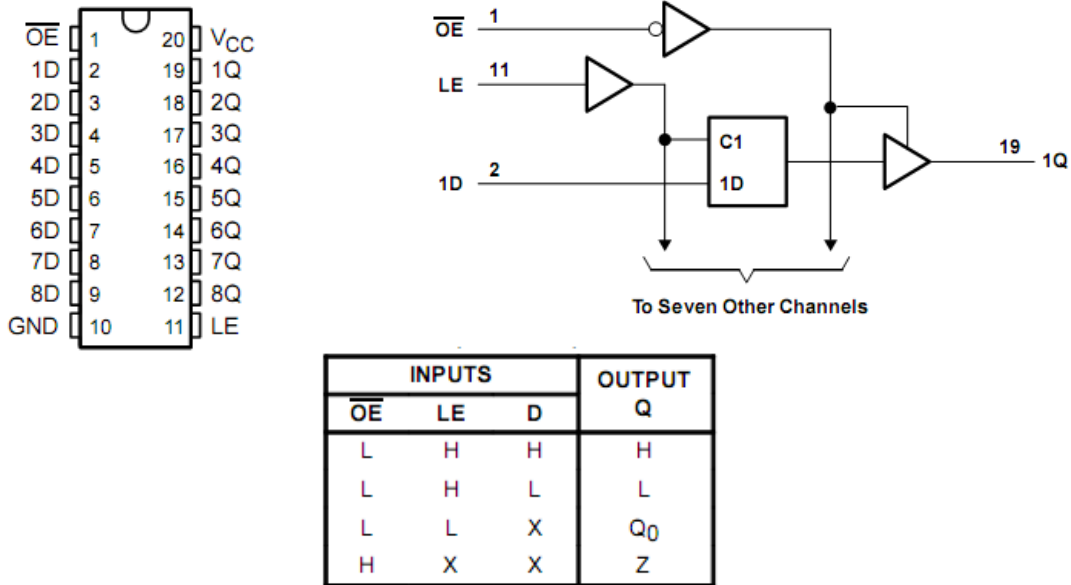
2.2 ไอซีสนับสนุน

2.2.1 ไอซี 74LS573

เป็นไอซี D-Flip - Flop แบบ 3 สถานะจำนวน 8 ตัว (Octal Tri State Flip- Flop) มีขา D0-D7 เป็นขาอินพุตและ ขา Q0 -Q7 เป็นขาเอาต์พุตโดยใช้ขา LE (Latch Enable)เป็นขาควบคุมการเก็บค่าสถานะของข้อมูล และขา OE (Output Enable) เป็นขาอินาเบิลควบคุมวงจรไตรสเตทให้ทำงานดังแสดงในภาพที่ 5.8 มีโครงสร้างภายใน และการทำงาน เช่นเดียวกับ 74LS373 ใช้สัญญาณนาฬิกา CLK ในการกำหนดการเก็บข้อมูล สำหรับการทำงานของไอซีกับระบบคือ เมื่อมีสัญญาณข้อมูลที่ต่อเข้ากับขา D0-D7 เข้ามา จะถูกเก็บค่าไว้ที่เอาต์พุต Q0-Q7 ของฟลิปฟลอป โดยการกำหนดสถานะของขา LE และขา OE ทำหน้าที่เปิดการทำงานให้กับไอซีไตรสเตท เพื่อส่งข้อมูลออกไปคงค้างไว้ที่เอาต์พุตในกรณีข้อมูลที่ขาอินพุต D0-D7 มีการเปลี่ยนค่าไปเอาต์พุตของไอซีจะยังคงเดิม และสามารถ

เปลี่ยนแปลงค่าได้ โดยกำหนดสถานะ ที่ขา LE ให้มีลอจิก “ 1” ให้มีการเก็บค่าข้อมูลอีกครั้ง และ กระตุ้นขา \overline{OE} ให้เป็นลอจิก “0” เพื่อให้ส่งข้อมูลออกไปเอาต์พุต

ไอซี 74LS573 สามารถนำไปขับพอร์ตของไอซี MCS-51 ในกรณีที่จ่ายกระแสไม่เพียงพอที่จะขับอุปกรณ์เอาต์พุตได้โดยตรง โดยทำหน้าที่ช่วยขับกระแส และยังเป็นบัฟเฟอร์ได้ ข้อดีของไอซี 74LS573 ในส่วนของฮาร์ดแวร์การวางตำแหน่งของขาอินพุต และเอาต์พุต มีการแยกทิศทางของขาไอซี ทำให้สะดวกในการออกแบบลายวงจรพิมพ์



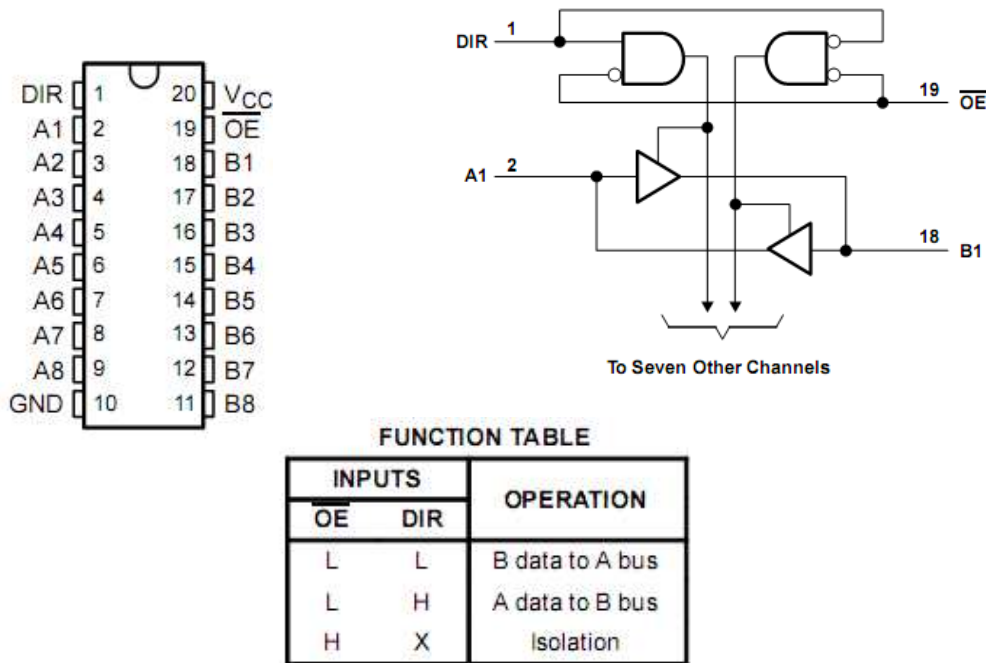
ภาพที่ 5.8 ไอซีเบอร์ 74LS573 D-Flip - Flop แบบ 3 สถานะ

(แหล่งอ้างอิง <http://www.es.co.th/Schematic/PDF/74LS573N.PDF>)

2.2.2 ไอซี 74LS245

ไอซี 74LS245 เป็นไอซีบัฟเฟอร์แบบ 3 สถานะ (Tri State Buffer) ภายในประกอบด้วย บัฟเฟอร์จำนวน 8 ตัวเพื่อรองรับอินพุต 8 ช่องสัญญาณ และใช้เป็นเอาต์พุต กรณีที่ต้องการใช้เชื่อมต่อพอร์ตกับอุปกรณ์ภายนอก แสดงดังภาพที่ 5.9 ในกรณีที่ไอซีไมโครคอนโทรลเลอร์ไม่สามารถที่จ่ายกระแสมากเกินกว่า จะขับอุปกรณ์เอาต์พุตได้โดยตรง ดังนั้นจึงต้องผ่านไอซีที่ทำหน้าที่บัฟเฟอร์ก่อน ส่วนการทำงาน ให้เป็นพอร์ตอินพุตเมื่อพอร์ตไม่ถูกเลือกให้ทำงาน ไอซีพอร์ตจะอยู่ในสถานะความต้านทานสูง (High Impedance) เสมือนว่าไม่ได้ต่อเข้ากับระบบ ซึ่งเรียกว่าสถานะลอยตัว (Floating) เมื่อมีสัญญาณเลือกพอร์ตมากระตุ้นให้ทำงาน พอร์ตจึงเชื่อมต่อเข้ากับบัสข้อมูลของระบบ โดยส่งข้อมูลออกมาที่บัสข้อมูล และซีพียูจะอ่านข้อมูลเข้าไป การใช้บัฟเฟอร์เป็นพอร์ตอินพุตเปรียบเสมือนกันชนระหว่างอุปกรณ์อินพุต กับ ไอซี MCS-51 ทำให้สัญญาณทั้ง 2 ด้านไม่กระทบกัน สัญญาณในระบบคอมพิวเตอร์จึงไม่เกิดความผิดพลาด สำหรับลักษณะของ

ไอซี 74LS245 สามารถควบคุมการส่งข้อมูลได้ 2 ทิศทางโดยใช้ขา DIR ถ้าให้ขา DIR มีสถานะเป็นลอจิก 0 เป็นการส่งข้อมูล จาก A ไป B ถ้ากำหนดให้ขา DIR มีสถานะลอจิก 1 เป็นการส่งข้อมูล จาก B ไป A



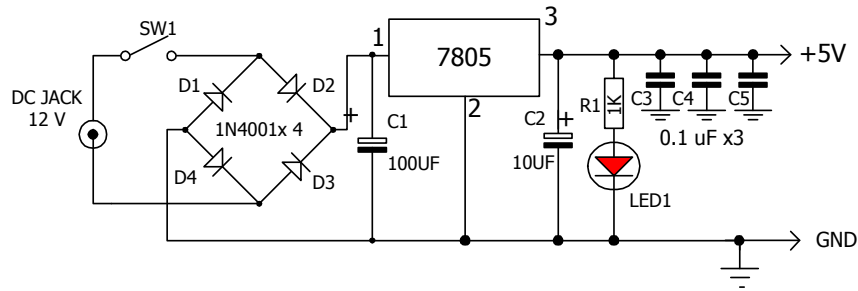
ภาพที่ 5.9 ไอซีเบอร์ 74LS245 บัฟเฟอร์แบบ 3 สถานะ
(แหล่งอ้างอิง <http://www.es.co.th/detail.asp?Prod=SN74LS245N>)

2.3 วงจรใช้งาน

จากภาพที่ 5.10 การทำงานของวงจร เริ่มจากสวิตช์ SW1 ทำหน้าที่ เปิด ปิด การจ่ายไฟให้กับวงจรทั้งหมด โดยใช้แหล่งจ่ายไฟแบบ DC หรือ AC ที่มีแรงดันไฟขนาด 9-12 โวลต์ ไดโอด D1-D4 ต่อเป็นวงจรบริดจ์ เพื่อการต่อขั้วไฟ DC จากอแดปเตอร์ที่อาจเป็นแบบขั้วไฟบวกอยู่ด้านนอกขั้วไฟลบอยู่ด้านใน หรือขั้วไฟลบอยู่ด้านนอกขั้วไฟบวกอยู่ด้านใน ก็สามารถจะใช้งานได้ หรืออาจใช้แหล่งจ่ายไฟ AC จากหม้อแปลงโดยตรง ที่มีขนาด 6-9 โวลต์เอซี วงจรไดโอดบริดจ์ก็จะทำหน้าที่ เป็นวงจรเรกติไฟเออร์ โดยทำการเปลี่ยนไฟฟ้ากระแสสลับ (AC) ให้เป็น ไฟฟ้ากระแสตรง (DC) และกลับขั้วไฟให้ถูกต้อง

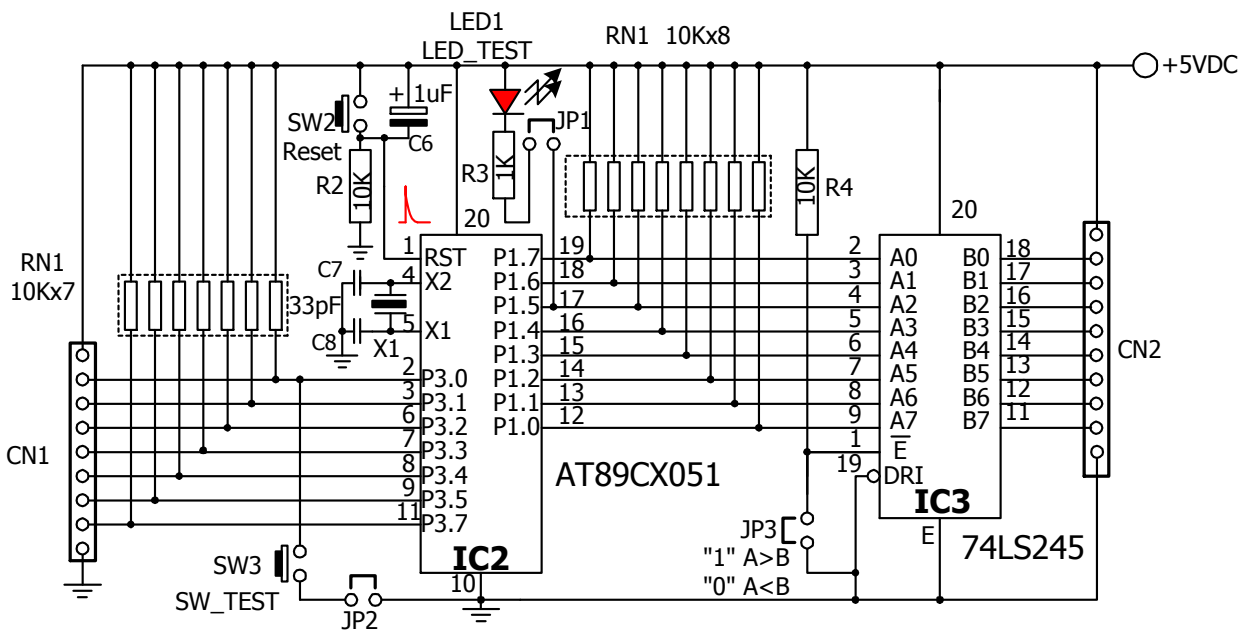
ตัวเก็บประจุ C1 ทำหน้าที่กรองแรงดันหรือฟิลเตอร์ (Filter) เพราะในการเปลี่ยนแรงดันไฟกระแสสลับให้เป็นแรงดันไฟกระแสตรง ยังมีการกระเพื่อมของแรงดันไฟตรงที่เรียกว่าริปเปิล (Ripple) ดังนั้นจึงใช้ตัวเก็บประจุเพื่อลดค่าแรงดันริปเปิลลงไป โดยการเก็บค่าประจุไว้เมื่อช่วงแรงดันสูง และจะจ่ายประจุให้กับโหลดเมื่อมีการกระเพื่อมทางด้านต่ำ ดังนั้นโหลดจะได้แรงดันที่ราบเรียบขึ้น IC1 เป็นไอซีเรกูเลเตอร์ (Regulate) ขนาด 5 โวลต์ ทำหน้าที่รักษาระดับของแรงไฟให้มีค่าคงที่ 5 โวลต์ ตัวเก็บประจุ C2 ทำหน้าที่กรองแรงดัน ที่ออกมาจากเอาต์พุตของ IC1 ส่วน C3,C4 และ C5 ทำหน้าที่กรองสัญญาณความถี่

สูงเกินไป ซึ่งอาจเกิดจากอุปกรณ์วงจรภายในของไอซี การใช้งานจะต่อระหว่างขาไฟเลี้ยงของไอซี กับขากราวด์ โดยให้ใกล้กับขาไอซีมากที่สุด ส่วนตัวต้านทาน R1 ทำหน้าที่จำกัดกระแสที่ป้อนให้กับ LED1 เพื่อกำหนดความสว่าง ถ้าหากค่าความต้านทานน้อยทำให้ LED1 สว่างมาก และกินกระแสมากขึ้น หากค่าความต้านทานน้อยเกินไปอาจทำให้ LED1 เสียหายได้ ดังนั้นการกำหนดความสว่างของ LED1 จึงให้พอดีสำหรับการแสดงสถานะการทำงานของบอร์ด ควรใช้ค่าความต้านทานที่มีค่ามาก แต่ยังให้แอลอีดีมีความสว่างสังเกตได้ จะเป็นการประหยัดแหล่งจ่ายไฟของระบบ



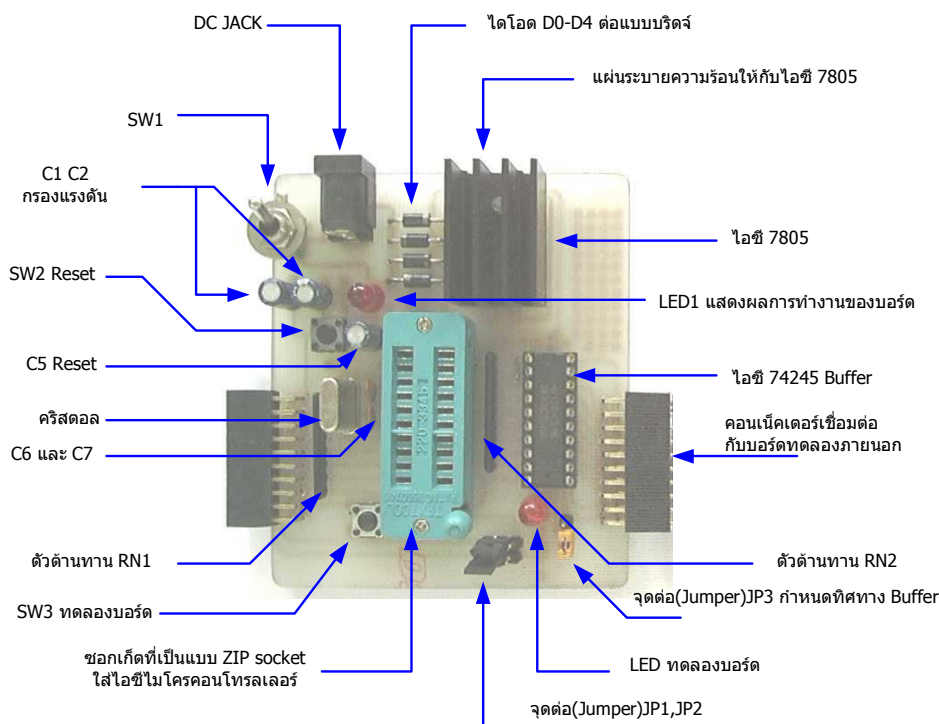
ภาพที่ 5.10 วงจรแหล่งจ่ายไฟกระแสตรง 5 โวลต์

วงจรใช้งานไอซีไมโครคอนโทรลเลอร์ขนาด 20 ขา ซึ่งจะเป็นชนิดที่มีหน่วยความจำแบบแฟลช (Flash Memory) แสดงดังภาพที่ 5.11 สามารถเลือกใช้ไอซีเบอร์ AT89C1051, AT89C2051 หรือ AT89C4051 โดยเลือกใช้งาน ตามขนาดหน่วยความจำที่ต้องการ



ภาพที่ 5.11 วงจรใช้งานไอซีไมโครคอนโทรลเลอร์ MCS-51 ขนาด 20 ขา

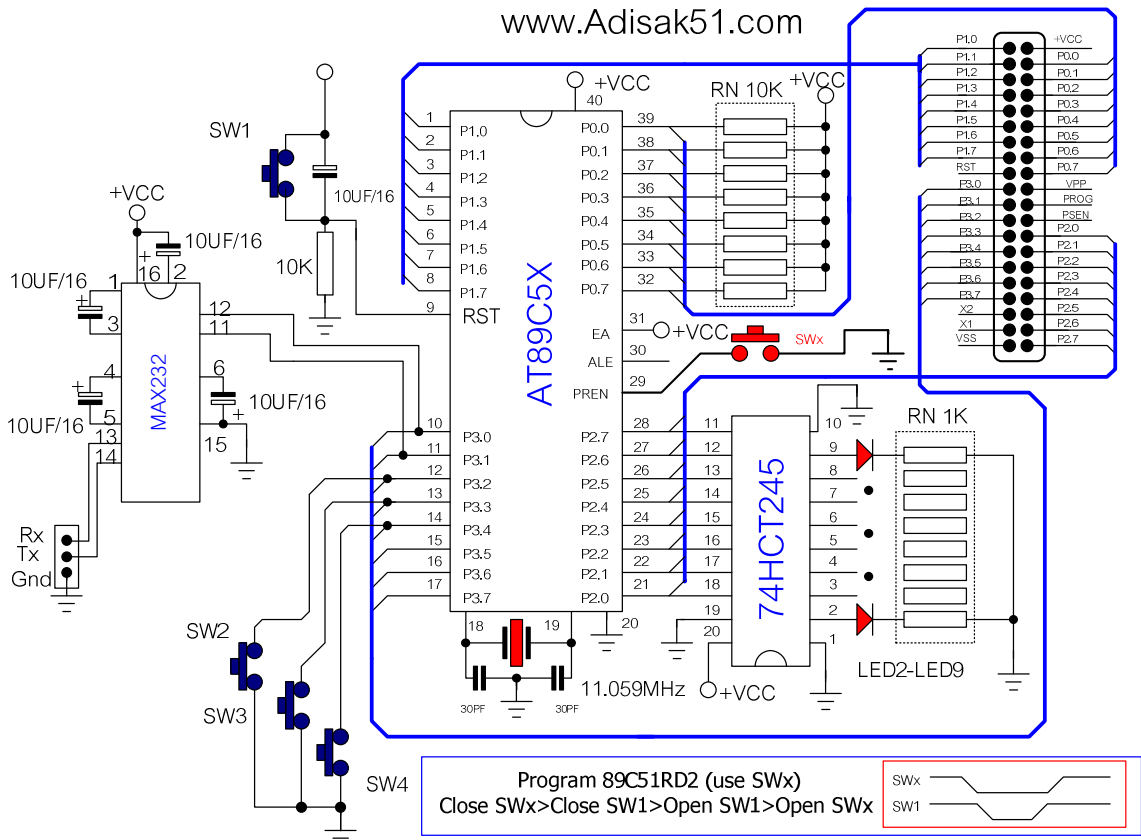
สวิตช์ SW2, C6 และ R2 จะเป็นวงจรกำหนดคาบเวลาของสัญญาณรีเซตของ IC2 ตัวต้านทาน RN1 และ RN2 นำมาต่อให้กับขาของพอร์ต P1 และ P3 ถึงแม้ว่าพอร์ต P1 และ P3 จะมีตัวต้านทานพูลอัปภายในแล้วก็ตาม (ยกเว้นขา P1.0 P1.1 จะไม่มีตัวต้านทานพูลอัปภายใน) IC2 จะได้รับสัญญาณนาฬิกาจากคริสตอล X1, C7 และ C8 ที่จุดต่อ (Jumper) JP1 มีไว้สำหรับการทดลองส่งข้อมูล กำหนดให้แสดงผลที่ LED 2 โดยให้ขาแคโทด (Cathode) ต่อกับพอร์ต P1.5 ของไอซีไมโครคอนโทรลเลอร์ ส่วนขาแอนโนด (Anode) จะต่อกับตัวต้านทานจำกัดกระแส ก่อนที่ต่อกับแรงดัน 5 โวลต์ ดังนั้นการให้แอลอีดีสว่างได้ต้องกำหนดสถานะลอจิกที่พอร์ต P1.5 เป็นลอจิก “0” (ต่ำ) LED จึงนำกระแสได้ ส่วนที่จุดต่อ JP2 มีไว้สำหรับการทดลองรับข้อมูล โดยใช้สวิตช์ SW3 ซึ่งต่อกับขาพอร์ต P3.0 ของไอซีไมโครคอนโทรลเลอร์



ภาพที่ 5.12 บอร์ดของไอซี MCS-51 ขนาด 20 ขา

IC3 เบอร์ 74LS245 จะทำหน้าที่เป็นบัฟเฟอร์ (Buffer) ข้อมูลขนาด 8 บิต ในกรณีที่ต้องการเชื่อมต่อพอร์ตกับอุปกรณ์ภายนอก ในกรณีที่ไอซีไมโครคอนโทรลเลอร์ไม่สามารถที่จ่ายกระแสมากเกินไปที่จะขับอุปกรณ์เอาต์พุตโดยตรงได้ ดังนั้นไอซีที่ทำหน้าที่บัฟเฟอร์จึงเปรียบเสมือนเป็นอุปกรณ์ป้องกันหรือตัวกันชน ให้กับไอซีไมโครคอนโทรลเลอร์ จุดต่อ JP3 ทำการเลือกทิศทางของไอซีบัฟเฟอร์ 74LS245 โดยถ้ากำหนด JP3 ต่อถึงกัน (Close) พอร์ต P1 จะรับข้อมูลจากภายนอกเป็นอินพุตเข้ามา แต่ถ้า JP3 ไม่ได้ต่อถึงกัน (Open) เป็นการเลือกให้พอร์ต P1 ส่งข้อมูลออกไปเป็นเอาต์พุตภายนอก บอร์ดของไอซี MCS-51 ขนาด 20 ขา แสดงดังภาพที่ 5.12

การใช้งานไอซีบัฟเฟอร์ 74LS245 และไอซี MAX232 ทำหน้าที่ติดต่อกับพอร์ตอนุกรมภายนอก โดยใช้มาตรฐาน RS232 ไอซี MCS-51 แบบ 40 ขา ที่พอร์ต 0 ไม่มีตัวต้านทานพูลอัปอยู่ภายใน การใช้งานต้องต่อเพิ่มจากภายนอก จึงต้องใช้ตัวต้านทาน Rn (Resistor Network) แสดงดังภาพที่ 5.13 วงจรที่ประกอบเป็นบอร์ดใช้งานแสดงดังภาพที่ 5.14



ภาพที่ 5.13 วงจรใช้งานไอซีไมโครคอนโทรลเลอร์ MCS-51 ขนาด 40 ขา



ภาพที่ 5.14 บอร์ดใช้งานไอซีไมโครคอนโทรลเลอร์ MCS-51 ขนาด 40 ขา

3. การรีเซ็ตไอซี MCS-51

ไอซี MCS-51 เมื่อถูกรีเซ็ต ค่าในรีจิสเตอร์ใช้งานพิเศษ จะมีค่าเป็น “0” ยกเว้นรีจิสเตอร์ P0-P3, SP และ SBUF โดยค่า P0 – P3 เป็นลอจิก “1” ทุกบิต ค่าในรีจิสเตอร์ SP จะชี้ที่แอดเดรส 07H ส่วนค่าในรีจิสเตอร์ SBUF มีค่าที่ไม่แน่นอน และค่าของหน่วยความจำข้อมูลภายในจะไม่มีการเปลี่ยนแปลง แต่ถ้าหากมีการรีเซ็ตโดยเริ่มจ่ายพลังงานให้ไอซีเป็นครั้งแรก (Power-On Reset) ค่าของหน่วยความจำภายใน จะมีค่าที่ไม่แน่นอน ส่วนค่าในรีจิสเตอร์ PC จะชี้ที่ตำแหน่ง 0000H ทำให้ ไอซี MCS-51 เริ่มต้นทำงานที่ตำแหน่งแอดเดรสแรก ค่าข้อมูลของรีจิสเตอร์เฉพาะที่เกิดขึ้นจากการรีเซ็ต ดังแสดงในตารางที่ 5.1

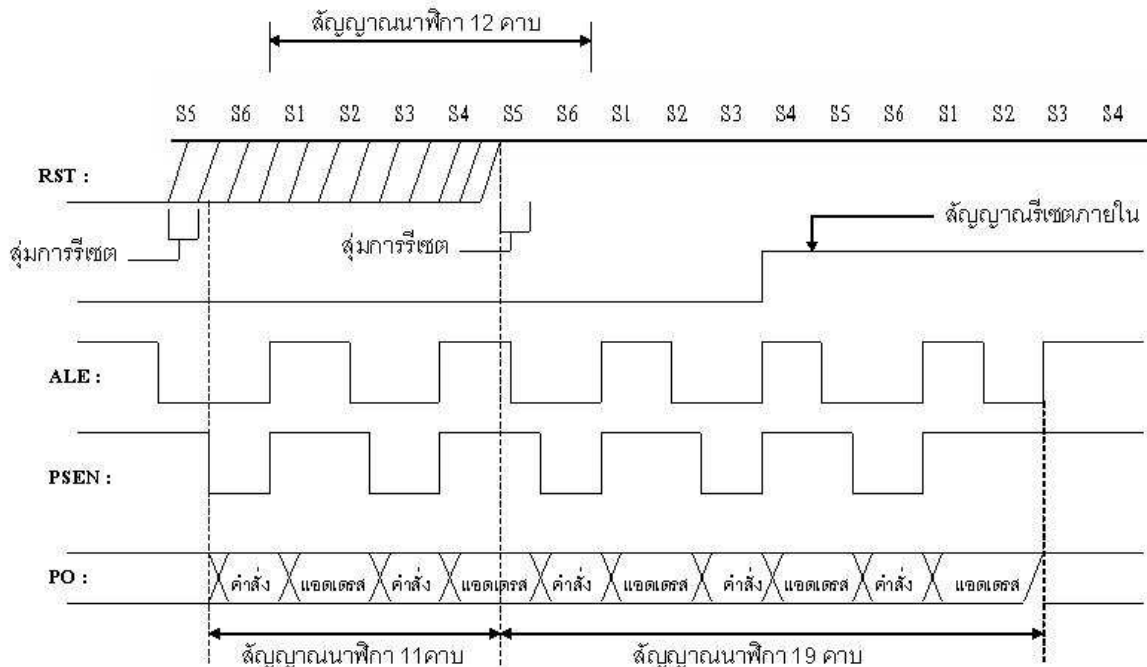
ตารางที่ 5.1 แสดงค่าข้อมูลของรีจิสเตอร์เฉพาะ ที่เกิดขึ้นจากการรีเซ็ต

SFR (Name)	Reset Value
PC	0000H
ACC	00H
B	00H
SP	07H
DPTR	0000H
P0-P3	11111111B
IP(8051)	XXX00000B
IP(8052)	XX000000B
IE(8051)	OXX00000B
IE(8052)	OX000000B
TMOD	00H
TCON	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
TH2(8052)	00H
TL2(8052)	00H
PCAP2H(8052)	00H
RCAP2L(8052)	00H
SCON	00H
SBUF	Indeterminate
PCON(HMOS)	OXXXXXXXB
PCON(CHMOS)	OXXX0000B

3.1 กระบวนการรีเซ็ตของไอซี MCS-51

สัญญาณรีเซ็ตเป็นสถานะลอจิก “1” ที่ขา RST อาจเกิดขึ้นได้ในทุกช่วงเวลา โดยไม่ต้องคำนึงถึงสัญญาณนาฬิกาของออสซิลเลเตอร์ที่ใช้ โดยจะสุ่มตรวจสอบขา RST ทุกๆ 5 สเตทที่เฟส 2 ของแต่ละแมชชีนไซเคิล และทุกพอร์ตยังทำงานต่อเนื่องที่จะปฏิบัติ อีกอย่างน้อย 19 คาบของสัญญาณนาฬิกา

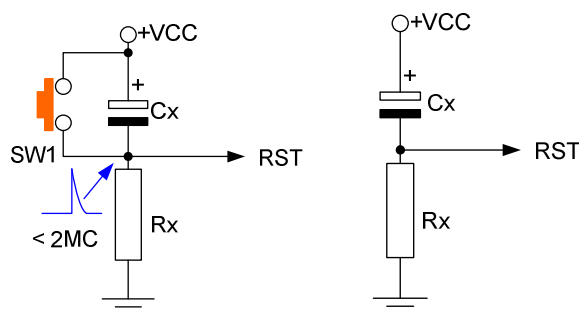
ในขณะที่ขา RST มีสถานะลอจิก “1” ขา ALE, \overline{PSEN} มีสถานะเป็น “1” เช่นเดียวกัน เมื่อขา RST มีสถานะกลับมาเป็น “0” วงจรรีเซ็ตภายในไอซี MCS-51 ต้องใช้เวลาอย่างน้อย 1 ถึง 2 แมกซีนไซเคิล ก่อนที่ขา ALE, \overline{PSEN} จะกลับมาเริ่มต้นทำงานใหม่อีกครั้ง แผนผังเวลาการรีเซ็ตแสดงดังภาพที่ 5.15



ภาพที่ 5.15 แสดงแผนผังเวลาการรีเซ็ต

3.2 สัญญาณรีเซ็ตโดยใช้วงจรร C (Resister Capacitor)

สัญญาณรีเซ็ตโดยใช้วงจรร C (Resister Capacitor) ประกอบด้วย SW1, Cx และ Rx แสดงดังภาพที่ 5.16 ยังแบ่งออกเป็น การรีเซ็ตโดยกดที่สวิตช์ (Manual Reset) และรีเซ็ตเมื่อเริ่มจ่ายพลังงานให้กับไอซี (Power-On Reset)

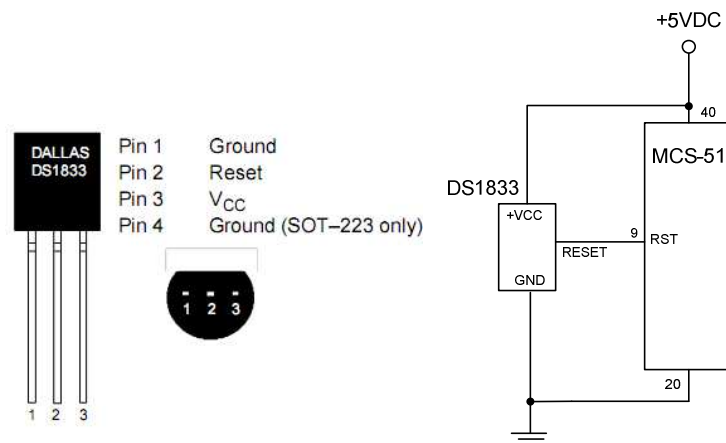


ภาพที่ 5.16 แสดงการต่อวงจรรีเซ็ตโดยสวิตช์กด และการต่อวงจรรีเซ็ตเมื่อเริ่มจ่ายพลังงานให้กับไอซี

การทำงานในภาพที่ 5.16 เมื่อเริ่มจ่ายกระแสไฟตรงขนาด 5 โวลต์ป้อนให้กับไอซี MCS-51 เป็นครั้งแรก Cx จะเริ่มเก็บประจุทันที แต่แรงดันที่ตัวเก็บประจุไม่สามารถเพิ่มในทันทีได้ เปรียบเสมือนการลัดวงจรชั่วขณะ ทำให้มีแรงดันตกคร่อมที่ตัวต้านทาน Rx ต่อกับ ขาริเซตเป็นสถานะ “1” อยู่ระยะเวลาหนึ่ง ตัวเก็บประจุ Cx เริ่มมีแรงดันเพิ่มขึ้นไปเรื่อย ๆ เมื่อมีการประจุจนเต็มทำให้ขาริเซตกลับมาเป็นสถานะลอจิก "0" (ต่ำ) ไอซี MCS-51 จึงเริ่มต้นทำงาน วิธีรีเซตโดยใช้สวิตช์กด SW1 ทำหน้าที่ลัดวงจรของตัวเก็บประจุ Cx จึงคายประจุอย่างรวดเร็ว หลังจากที่ปล่อยคีย์สวิตช์ ทำให้ Cx เริ่มทำการประจุใหม่อีกครั้ง

3.3 ไอซีสร้างสัญญาณรีเซต

วงจรรีเซตที่มีเสถียรภาพทำให้ลดผลของความไม่แน่นอน ในการใช้วงจรแบบ RC รีเซตที่เริ่มจ่ายพลังงานให้ไอซีครั้งแรก ในปัจจุบันบริษัท Dallar Semiconductor ได้ผลิตไอซีสำหรับสร้างสัญญาณรีเซตให้กับไอซี MCS-51 ขึ้นมาโดยเฉพาะ เป็นไอซีขนาด 3 ขา เบอร์ DS1833 โดยสร้างสัญญาณรีเซตลอจิกสูง (“1”) การต่อไอซีรีเซตเข้ากับ MCS-51 และการจัดขาของไอซีแสดงดังภาพที่ 5.17

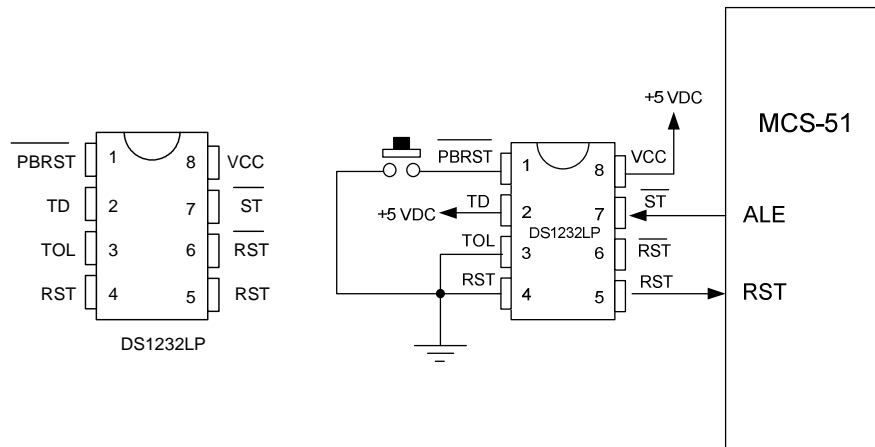


ภาพที่ 5.17 ไอซีสร้างสัญญาณรีเซตลอจิกสูง DS1833

(แหล่งอ้างอิง <http://www.dalsemi.com>)

3.4 ไอซีวอตช์ด็อกไทมเมอร์ (Watchdog Timer: WDT)

ไอซีวอตช์ด็อกไทมเมอร์เบอร์ DS1232 ทำหน้าที่ตรวจสอบการทำงานของไอซี MCS-51 ในกรณีเกิดภาวะหยุดทำงานกะทันหันโดยไม่ทราบสาเหตุ มีการทำงานอย่างต่อเนื่องแม้อยู่ในโหมดสลีพ (Sleep Mode) เมื่อสั่งให้วอตช์ด็อกไทมเมอร์ ทำงาน จะเริ่มนับค่าเวลาจนถึงที่ตั้งไว้ โดยไอซี MCS-51 ต้องส่งสัญญาณเคลียร์ค่าในวอตช์ด็อกไทมเมอร์ ก่อนที่วอตช์ด็อกไทมเมอร์จะส่งสัญญาณไปรีเซตตัวไอซี ดังนั้นจึงต้องเขียนโปรแกรมให้ซีพียูส่งพัลส์ 1 ลูกตลอดเวลา เพื่อกระตุ้นให้วอตช์ด็อกไทมเมอร์ ที่ขา \overline{ST} (Strobe Input) ของไอซี DS1232 และยังสามารถใช้ขา ALE ของไอซี MCS-51 ซึ่งมีสัญญาณส่งออกทุก 1/6 ของสัญญาณรายการการต่อวงจรแสดงดังภาพที่ 5.18 ส่วนการกำหนดค่าเวลาส่งสัญญาณรีเซตให้กับซีพียู กำหนดโดยต่อขา TD ดังแสดงในตารางที่ 5.2



ภาพที่ 5.18 ไอซีวงจรวอตช์ดีค็อกไทเมอร์เบอร์ DS1232

ขาและการทำงาน

- PBRST – ขาต่อกับสวิตซ์รีเซต
- TD – ขากำหนดค่าเวลาส่งสัญญาณรีเซต
- TOL – ขาตรวจสอบระดับแรงดันไฟ 5% หรือ 10%
- GND – ขากราวด์
- RST – ขาเอาต์พุตรีเซต ทำงานที่ลอจิกสูง Reset Output (Active High)
- RST – ขาเอาต์พุตรีเซต ทำงานที่ลอจิกต่ำ เป็นแบบวงจร Open Drain
- ST – สัญญาณอินพุตรีเซตค่าจากภายนอก
- VCC – ขาไฟ VCC +5

ตารางที่ 5.2 การกำหนดค่าเวลา โดยการต่อขา TD ตามตำแหน่ง

TD	TIME-OUT		
	MIN	TYP	MAX
ต่อลงกราวด์	62.5 ms	150 ms	250 ms
ลอยขา	250 ms	600 ms	1000 ms
ต่อกับไฟ+VCC	500 ms	1200 ms	2000 ms

3.5 วอตช์ดีค็อกไทเมอร์ภายในไอซี MCS-51 (Watchdog Timer: WDT)

ไอซี MCS-51 ที่มีวงจรวอตช์ดีค็อกไทเมอร์เพิ่มเติม โดยนำไปสร้างสัญญาณรีเซตกรณีที่ซีพียูหยุดทำงานกะทันหัน ไอซีเบอร์ AT89S8252 และ AT89S53 ของบริษัท Atmel มีวงจรวอตช์ดีค็อกไทเมอร์แบบโปรแกรมได้ โดยกำหนดค่าของบิต PS0-PS2 ในรีจิสเตอร์ WCON สำหรับเบอร์ AT89S8252 และ รีจิสเตอร์ WCON สำหรับเบอร์ AT89S53

วอตช์ด็อกไทเมอร์จะได้รับอนุญาตให้เปิดใช้งานโดยอัตโนมัติ เมื่อเริ่มต้นจ่ายไฟให้กับไอซี MCS-51เข้าสู่การทำงานในโหมดประหยัดพลังงาน แบบลดพลังงาน หรือเพาเวอร์ดาวน์ สำหรับการอนุญาตใช้งานได้โดยกระบวนการทางซอฟต์แวร์ โดยทำการเซตบิต WDTEN ในรีจิสเตอร์ WMCON สำหรับเบอร์ AT89S8252 และรีจิสเตอร์ WCON สำหรับเบอร์ AT89S53

การทำงานของวอตช์ด็อกไทเมอร์เริ่มต้น เมื่อมีการกำหนดคาบเวลา และรออนุญาตให้เปิดใช้งาน เมื่อเริ่มต้นวงจรนับภายในวอตช์ด็อกไทเมอร์ จะนับค่าเวลาตามปกติ หากชิพทำงานจะส่งสัญญาณรีเซตหรือเคลียร์ค่าในวอตช์ด็อกไทเมอร์อยู่เป็นระยะ เพื่อไม่ให้วอตช์ด็อกไทเมอร์ทำงาน จนถึงค่าที่กำหนดหรือเกิดการไทม์เอาต์ (Time Out) ขึ้น หากชิพหยุดทำงานกะทันหัน ไม่ว่าจะด้วยสาเหตุใด ยกเว้นการรีเซตและการเข้าสู่โหมดประหยัดพลังงาน แบบลดพลังงานหรือเพาเวอร์ดาวน์ จะไม่มีสัญญาณจากชิพมารีเซตหรือเคลียร์ค่าในวอตช์ด็อกไทเมอร์ ทำให้ถึงค่าเวลาที่กำหนด เกิดการไทม์เอาต์วอตช์ด็อกไทเมอร์ ไอซีวอตช์ด็อกจะสร้างสัญญาณรีเซตส่งไปยังชิพ ส่งผลให้ชิพกลับมาเริ่มต้นทำงานใหม่ได้อีกครั้ง โดยไม่จำเป็นต้องทำการรีเซตจากภายนอก หรือจ่ายไฟเลี้ยงใหม่ให้แก่ไอซี MCS-51

รีจิสเตอร์ควบคุมการทำงานของวอตช์ด็อกไทเมอร์ (Watchdog & Memory Control Register: WMCON และ Watchdog Control Register: WCON) แสดงดังภาพที่ 5.19

การทำงานของวอตช์ด็อกไทเมอร์ในไอซี MCS-51 แบบแฟลชอนุกรมเบอร์ AT89S8252 มีรีจิสเตอร์ WMCON ควบคุมการทำงาน ส่วนเบอร์ AT89S53 ใช้รีจิสเตอร์ WCON

รีจิสเตอร์ WMCON และ WCON เป็นรีจิสเตอร์ขนาด 8 บิต มีแอดเดรสอยู่ที่ 96H สามารถเข้าถึงได้ในระดับบิต มีรายละเอียดดังนี้

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0	WMCON AT89S8252
PS2	PS1	PS0	EEMWE	EEMEN	DPS	WDTRST	WDTEN	
PS2	PS1	PS0			DPS	WDTRST	WDTEN	WCON AT89S53

PS2-PS0 (Prescaler bits)			บิตพรีสเกลเลอร์			ค่าเวลาของ วอตช์ด็อกไทเมอร์
PS2	PS1	PS0	PS2	PS1	PS0	
0	0	0	0	0	0	16 ms
0	0	1	0	0	1	32 ms
0	1	0	0	1	0	64 ms
0	1	1	0	1	1	128 ms
1	0	0	1	0	0	256 ms
1	0	1	1	0	1	512 ms
1	1	0	1	1	0	1,024 ms
1	1	1	1	1	1	2,048 ms

ภาพที่ 5.19 รีจิสเตอร์ควบคุมการทำงานของวอตช์ด็อกไทเมอร์

EEMWE (EEPROM Data Memory Write Enable Bit) เป็นบิตที่ 4 ของรีจิสเตอร์ WCON ทำหน้าที่ อนุญาตเปิดให้เขียนข้อมูลลงในหน่วยความจำข้อมูล แบบอีพรอมภายในไอซี MCS-51 เบอร์ AT89S8252 บิตนี้จะถูกเซตเมื่อต้องการเขียนข้อมูล หลังจากเขียนข้อมูลเสร็จแล้ว ต้องเคลียร์บิตนี้เสมอ ส่วนในรีจิสเตอร์ WCON บิตนี้ไม่มีการใช้งานต้องกำหนดให้เป็นลอจิก “0”

EEMEN (Internal EEPROM Access Enable Bit) เป็นบิตที่ 3 ของรีจิสเตอร์ WCON ทำหน้าที่ อนุญาตให้เปิดการเข้าถึงหน่วยความจำข้อมูลแบบอีพรอมภายใน AT89S8252 โดยใช้คำสั่ง MOVX และใช้งานรีจิสเตอร์ DPTR ต้องเซตบิตนี้ให้เป็นลอจิก “0” เป็นการกำหนดให้ใช้คำสั่ง MOVX และรีจิสเตอร์ DPTR ติดต่อกับหน่วยความจำข้อมูลภายนอก สำหรับรีจิสเตอร์ WCON ของ AT89S53 บิตนี้ไม่ใช้งานกำหนดเป็นลอจิก “0”

DPS (Data Pointer Register Select) เป็นบิตที่ 2 ของรีจิสเตอร์ WCON และ WCON ทำหน้าที่ เลือกใช้รีจิสเตอร์ DPTR โดยถ้าเป็นลอจิก “0” เป็นการเลือกใช้รีจิสเตอร์ DPTR0 (DP0L และDP0H) ถ้ามีสถานะเป็นลอจิก “1” เป็นการเลือกใช้รีจิสเตอร์ DPTR1 (DP1L และDP1H)

WDTRST/RDY/BSY (Watchdog Timer Reset and EEPROM Ready/Busy Flag) บิตนี้มีการทำงาน 2 ส่วนคือ การรีเซตของวอตช์ด็อกไทมเมอร์ และใช้เป็นบิตแสดงสถานะของหน่วยความจำข้อมูลแบบอีพรอมภายในของ AT89S8252 การใช้เป็นวอตช์ด็อกไทมเมอร์ บิตนี้เซตโดยผู้ใช้งานด้วยกระบวนการทางซอฟต์แวร์ให้วอตช์ด็อกไทมเมอร์สร้างสัญญาณรีเซตส่งออกไป ในกรณีที่วอตช์ด็อกไทมเมอร์เกิดไทม์เอาต์ หลังจากส่งสัญญาณรีเซตออกไปแล้ว บิตจะเคลียร์ตัวเองเป็น “0” อัลดโนมัติในไซเกิลของการทำงานถัดไป ดังนั้นบิตนี้จึงเขียนได้เพียงอย่างเดียว (Write-Only)

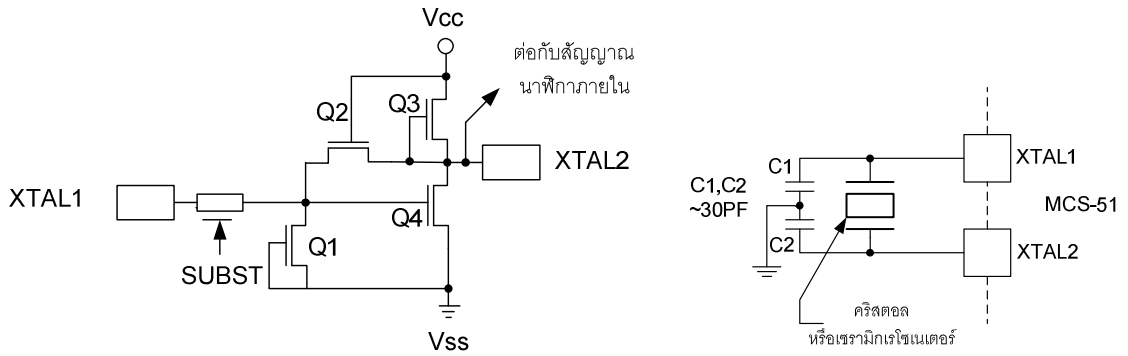
ในกรณีใช้กับหน่วยความจำข้อมูลอีพรอม บิตนี้ใช้ป้องกันการเขียนข้อมูลลงในหน่วยความจำ ถ้าเป็น “1” หน่วยความจำข้อมูลอีพรอม จะอ่านได้เพียงอย่างเดียว เมื่อต้องการเขียนข้อมูลหลังจากอนุญาตให้มีการเขียนบิต EEMWE แล้ว ต้องเคลียร์บิตนี้ให้เป็นลอจิก “0” หลังจากเขียนข้อมูลเสร็จสิ้นลงบิตนี้ จะเซตเป็น “1” อัลดโนมัติ เป็นการแจ้งเขียนข้อมูลเสร็จสิ้นลง สำหรับไอซีเบอร์ AT89S53 ใช้งานบิตนี้สร้างสัญญาณรีเซตของวอตช์ด็อกไทมเมอร์เท่านั้น

WSTEN (Watchdog Timer Enable Bit) บิตนี้ทำหน้าที่อนุญาตให้มีการทำงานของวอตช์ด็อกไทมเมอร์โดยกำหนดสถานะลอจิกของบิตเป็น “0” ไม่อนุญาตให้มีการทำงานของวอตช์ด็อกไทมเมอร์ถ้าบิตมีลอจิก “1” เป็นการอนุญาตให้วอตช์ด็อกไทมเมอร์ทำงาน

4. วงจรออสซิลเลเตอร์ของไอซี MCS-51

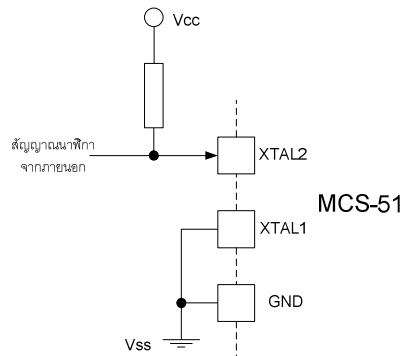
MCS-51 มีวงจรออสซิลเลเตอร์สำหรับสร้างสัญญาณนาฬิกา เพื่อกำหนดจังหวะการทำงานของซีพียู วงจรออสซิลเลเตอร์ใน MCS-51 แต่ละเบอร์ มีข้อแตกต่างกันออกไปขึ้นกับเทคโนโลยีที่ใช้ในการผลิต ซึ่งผลิตโดยเทคโนโลยี HMOS (High-Performance N-Channel MOS) และ CHMOS (Complementary High-Density MOS) วงจรออสซิลเลเตอร์ที่อยู่ภายใน MCS-51 ผลิตโดยเทคโนโลยี HMOS ดังแสดงในภาพที่ 5.20

อดิศักดิ์ ชิมะวงศ์ : www.Adisak51.com วิชาไมโครคอนโทรลเลอร์

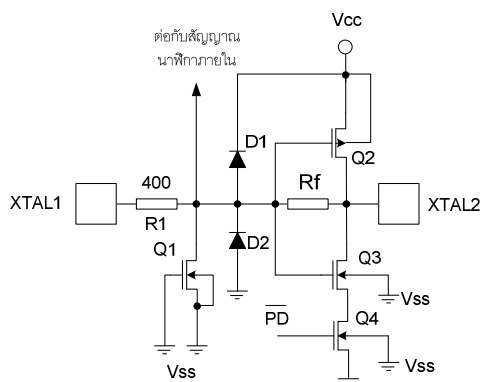


ภาพที่ 5.20 วงจรออสซิลเลเตอร์ภายใน MCS-51 แบบ HMO5 และการต่อใช้งาน

การควบคุมวงจรรออสซิลเลเตอร์ด้วยสัญญาณนาฬิกาจากภายนอก โดยต่อสัญญาณเข้าขา XTAL1 และต่อที่ขา XTAL2 ลงกราวด์ แสดงดังในภาพที่ 5.21 ที่ขา XTAL2 ควรต่อตัวต้านทานพูลอัพ เพื่อป้องกันสัญญาณรบกวน เพราะต้องการสัญญาณในระดับ TTL MCS-51 ที่ผลิตโดยเทคโนโลยีแบบ CHMOS วงจรรออสซิลเลเตอร์ภายใน แสดงดังภาพที่ 5.22



ภาพที่ 5.21 แสดงการใช้สัญญาณนาฬิกาจากภายนอกป้อนให้แก่วงจรรออสซิลเลเตอร์ใน MCS-51

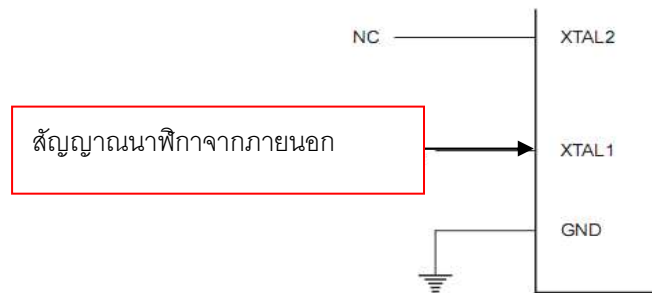


ภาพที่ 5.22 วงจรรออสซิลเลเตอร์ภายใน MCS-51 แบบ CHMOS

วงจรรอสซิชิลเลเตอร์แบบ CHMOS ไอซี MCS-51 สามารถหยุดการทำงานของวงจรรอสซิชิลเลเตอร์ภายใต้การควบคุมด้วยซอฟต์แวร์ โดยการโหลดค่า “1” ในบิต PD ของรีจิสเตอร์ PCON หากใช้สัญญาณนาฬิกาภายนอกเป็นตัวควบคุมวงจรรอสซิชิลเลเตอร์ ต้องต่อสัญญาณเข้าที่ขา XTAL1 การนำไปใช้งานเมื่อใช้คริสตอลเป็นตัวควบคุม ตัวเก็บประจุจะใช้ค่า 30 PF และเมื่อใช้วงจรเซรามิกเรโซเนเตอร์ ตัวเก็บประจุจะใช้ค่า 47 PF โดยตัวเก็บประจุสามารถเปลี่ยนค่าได้ ± 10 PF

ตัวต้านทานทำหน้าที่เป็นฟีดแบ็คซิสเตอร์ (R_f ในภาพที่ 5.22) ประกอบด้วย N และ P Channel FETs ที่ถูกควบคุมด้วยบิต \overline{PD} มีค่าเป็น “1” ตัวไดโอด D1 และ D2 ทำหน้าที่รักษาระดับโวลต์ (Clamp) เพื่อป้องกัน V_{cc} และ V_{ss} ที่จะเข้าไปทำลาย R1 FETs

หากต้องการใช้สัญญาณนาฬิกาภายนอกเป็นตัวควบคุมวงจรรอสซิชิลเลเตอร์ สัญญาณนาฬิกาต้องต่อเข้าที่ขา XTAL1 และปล่อยขา XTAL2 ลอยไว้ แสดงดังภาพที่ 5.23

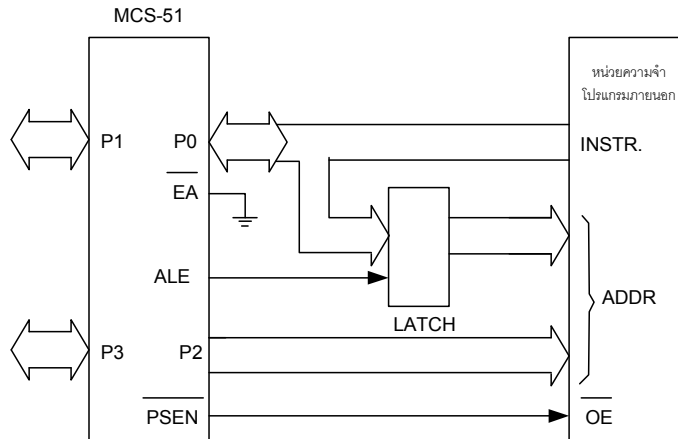


ภาพที่ 5.23 แสดงการใช้สัญญาณนาฬิกาภายนอกป้อนให้แก่วงจรรอสซิชิลเลเตอร์ใน MCS-51

5. การต่อไอซีหน่วยความจำภายนอกไอซี MCS-51

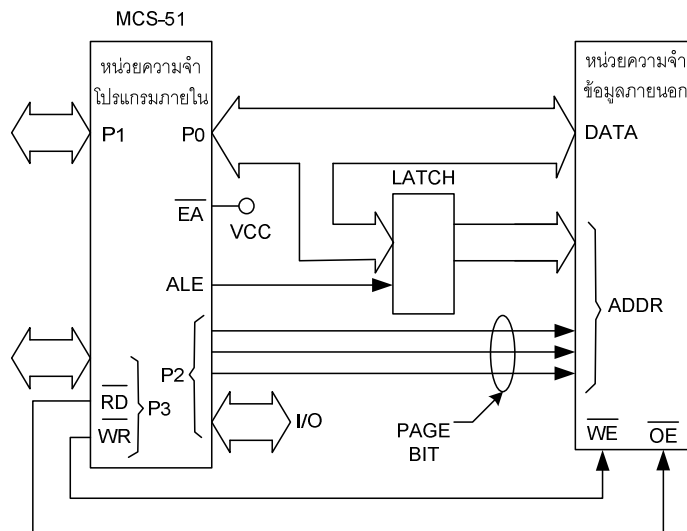
ไอซี MCS-51 สามารถติดต่อกับหน่วยความจำโปรแกรมภายนอก และหน่วยความจำข้อมูลภายนอกได้ถึง 64 กิโลไบต์โดยใช้ขาแอดเดรสจำนวน 16 เส้น ขาข้อมูล 8 เส้น และขาควบคุมอีก 3 เส้นส่งออกมาทางพอร์ต 0 และพอร์ต 2 ของไอซี MCS-51

หน่วยความจำโปรแกรมภายนอกที่นำมาต่อร่วมกับไอซี MCS-51 เป็นหน่วยความจำแบบอ่านได้เพียงอย่างเดียว หรือหน่วยความจำแบบถาวร สัญญาณจากไอซี MCS-51 ควบคุมการติดต่อกับหน่วยความจำโปรแกรมโดยใช้ขา \overline{PSEN} แสดงดังภาพที่ 5.24



ภาพที่ 5.24 สัญญาณควบคุมหน่วยความจำโปรแกรมโดยใช้ขา \overline{PSEN}

หน่วยความจำข้อมูลภายนอกที่นำมาต่อร่วมกับไอซี MCS-51 เป็นหน่วยความจำแบบอ่านและเขียนข้อมูลได้ สัญญาณที่ต่อมาจากไอซี MCS-51 เพื่อทำการอ่านข้อมูลจากหน่วยความจำ จะใช้ขา \overline{RD} ควบคุม หากเป็นการเขียนข้อมูลจะใช้ขา \overline{WR} ควบคุมแสดงดังภาพที่ 5.25



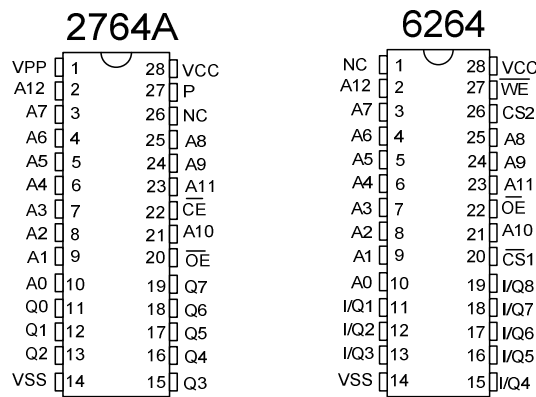
ภาพที่ 5.25 วงจรไอซี MCS-51 ที่ใช้ติดต่อหน่วยความจำข้อมูลในการอ่านและเขียนข้อมูล

5.1 ไอซีหน่วยความจำ

ไอซีหน่วยความจำแบบถาวรชนิด EPROM เบอร์ 2764 มีขนาดความจุ 8 กิโลไบต์ (A0-A12) โดยใช้ขา \overline{OE} (Output Enable) ควบคุมการอ่านค่าข้อมูลในตัวไอซี ถ้าขา \overline{OE} นี้ถูกแอกทีฟ ข้อมูลที่อยู่ในไอซี จะถูกส่งออกทางบัสคราด้า ส่วนขา \overline{CS} (Chip Enable) ใช้เลือกไอซีหน่วยความจำที่ต้องการอ่านค่าในกรณีมี

ไอซี EPROM จำนวนหลายตัว โดยใช้งานร่วมกับไอซีลอตรหัส แสดงตำแหน่งขาไอซีหน่วยความจำ 2764 แสดงดังภาพที่ 5.26

ไอซีหน่วยความจำข้อมูลแบบชั่วคราวชนิด RAM เบอร์ 6264 เป็นหน่วยความจำที่มีความจุขนาด 8 กิโลไบต์ (A0-A12) โดยใช้ขา \overline{OE} ควบคุมการอ่านค่าข้อมูลในตัวไอซี ถ้าขานี้ถูกแอกทีฟ ข้อมูลที่อยู่ในไอซีจะถูกส่งออกทางบัสค่าขา \overline{WR} ใช้สำหรับการเขียนข้อมูล ถ้าขานี้ถูกแอกทีฟข้อมูลที่ส่งมาจากขา ค่าตัวบัสจะถูกเขียนโดยเก็บค่าข้อมูลไว้ในตัวไอซีหน่วยความจำ และขา \overline{CS} ใช้สำหรับการเลือกไอซีตัวที่ต้องการอ่านค่า แสดงตำแหน่งของขาไอซีหน่วยความจำ 6264 แสดงดังภาพที่ 5.26



ภาพที่ 5.26 แสดงตำแหน่งขาของไอซีหน่วยความจำเบอร์ 2764 (EPROM) และ6264 (RAM)

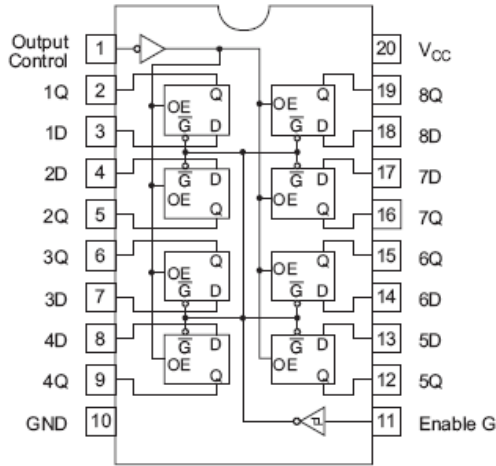
ไอซีหน่วยความจำมีให้เลือกใช้งานหลายเบอร์ มีขนาดความจุ และวิธีการบันทึกข้อมูลแตกต่างกันไป ปัจจุบันหน่วยความจำแบบ EEPROM และแบบ Flash นิยมนำมาใช้เป็นหน่วยความจำโปรแกรมภายในของไอซี MCS-51 ค่าความจุของแต่ละเบอร์ ดังแสดงในตารางที่ 5.3

ตารางที่ 5.3 แสดงค่าความจุ ของหน่วยความจำแบบ ต่างๆ

EPROM	ความจุ	EEPROM	ความจุ	Flash	ความจุ	RAM	ความจุ
2704	512 x 8						512 x 8
2708	1K x 8						1K x 8
2716	2K x 8					6116	2K x 8
2732	4K x 8					6232	4K x 8
2764	8K x 8	28C64	8K x 8	29F64		6264	8K x 8
27128	16K x 8	28C128		29F128		62128	16K x 8
27256	32K x8	28C256	32K x8	29F256		62256	32K x8
27512	64Kx8	28C512	64Kx8	29F512	64Kx8	62512	64Kx8

(แหล่งอ้างอิง <http://www.es.co.th/Schematic/PDF/>)

ไอซีเบอร์ 74LS373 เป็นไอซี Latch ข้อมูล หรือการคงสถานะข้อมูลไว้ แสดงดังภาพที่ 5.27 นำมาต่อวงจรเพื่อ ทำหน้าที่แยกสัญญาณ แอดเดรสบิต และดาต้าบิต ของพอร์ต 0 ออกจากกัน ใช้ขา ALE ของ ไอซี MCS-51 เป็นขาควบคุมการสลับ สัญญาณแอดเดรสบิต หรือดาต้าบิต เมื่อไอซี 74LS373 ได้รับสัญญาณ จากขา ALE จะทำการ Latch ข้อมูลของพอร์ต 0 โดยคงสถานะไว้เป็นค่าของแอดเดรสไบต์ต่ำ ส่วนพอร์ต 2 ส่งค่าของแอดเดรสไบต์สูง จึงทำให้เป็นการส่งแอดเดรสให้กับหน่วยความจำพร้อมกันแสดงดังภาพที่ 5.28

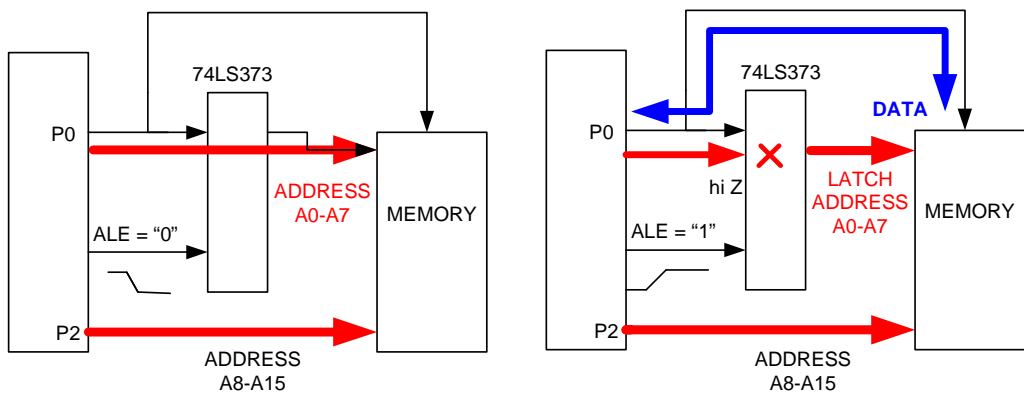


Inputs			Output
Output control	Enable G	D	Q
L	H	H	H
L	H	L	L
L	L	X	Q ₀
H	X	X	Z

Notes: H; high level, L; low level, X; irrelevant
 Q₀; level of Q before the indicated steady-state input conditions were established
 Z; off (high-impedance) state of a three-state output

ภาพที่ 5.27 แสดงตำแหน่งขาของไอซีเบอร์ 74LS373

(แหล่งอ้างอิง <http://www.es.co.th/Schemetic/PDF/SN74LS373.PDF>)

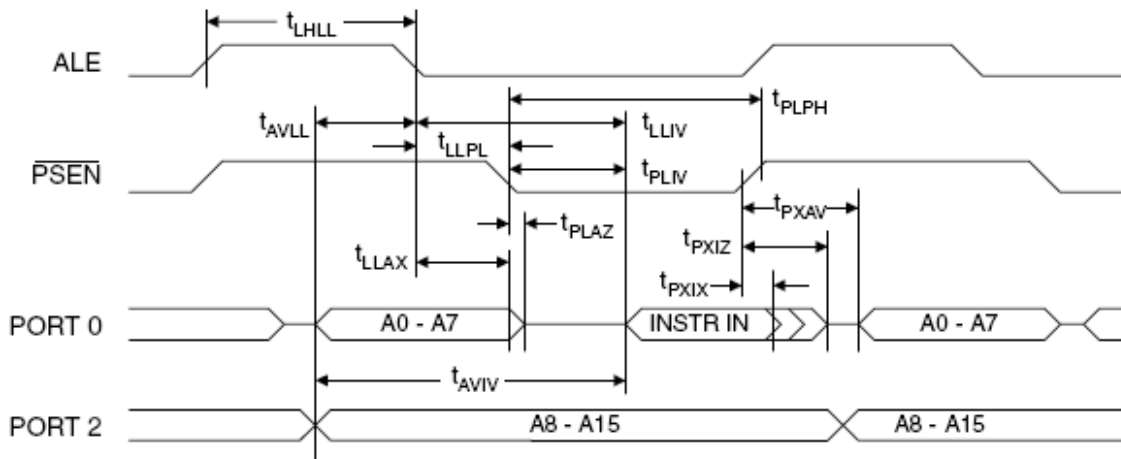


ภาพที่ 5.28 แสดงสัญญาณเวลาการสลับตำแหน่ง แอดเดรส และดาต้า ของไอซี MCS51

5.2 การต่อไอซีหน่วยความจำโปรแกรมภายนอก

หน่วยความจำโปรแกรมชนิด EPROM เบอร์ 2764 มีขนาดความจุ 8 กิโลไบต์ มีขาแอดเดรสจำนวน 13 ขา กำหนดให้เป็นขา A0-A12 มีขาข้อมูล 8 บิต กำหนดให้เป็นขา D0-D7 วงจรของ ไอซี MCS-51 จะต่อขา \overline{PSEN} เข้ากับขา \overline{OE} ของไอซี เบอร์ 2764

การทำงานของวงจรแสดงดังภาพที่ 5.29 โดยพอร์ต 0 ถูกออกแบบให้เป็นที่ทั้งบัสแอดเดรสและบัสข้อมูล (Multiplexed Address/Data Bus) ถ้าหากใช้เป็นบัสแอดเดรส A0-A7 จะถูกเป็นช่องทางส่งไบต์ล่าง (Low Byte) และรอรับข้อมูล D0-D7 จากหน่วยความจำ สำหรับพอร์ต 2 จะเป็นบัสแอดเดรส A8-A15 ซึ่งเป็นช่องทางส่งไบต์บน (High Byte) โค้ดอะแอสเวลการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกจะใช้เวลา 1 แมกซิมัซเซก



ภาพที่ 5.29 โค้ดอะแอสเวลการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก

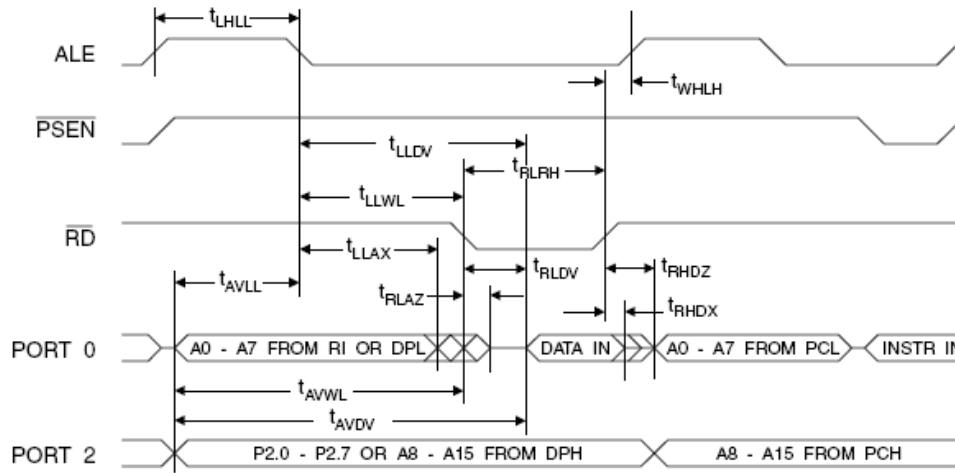
ขั้นตอนการติดต่อกับหน่วยความจำโปรแกรมภายนอกสรุปได้ดังนี้

- 1) ซีพียูส่งแอดเดรสของหน่วยความจำโปรแกรมไปยังตัว Program Counter: PC
- 2) ซีพียูส่งค่าใน Program Counter คำสั่งส่งไปยังพอร์ต 0 และพอร์ต 2 (Address A0-A15)
- 3) ซีพียูจะส่งสัญญาณลอจิก 1 ออกมาที่ขา ALE เพื่อแลตซ์ค่าแอดเดรส A0-A7 โดยให้คงสถานะไว้ที่เอาต์พุตของไอซี 74LS373
- 4) ซีพียูส่งสัญญาณลอจิก 0 ออกมาที่ขา \overline{PSEN} เพื่ออ่านข้อมูล จาก EPROM
- 5) ข้อมูลภายใน EPROM จะถูกอ่านลงสู่บัสข้อมูล D0-D7 (พอร์ต 0)
- 6) ซีพียูอ่านข้อมูลที่บัส D0-D7 เพื่อทำการแปลความหมายและปฏิบัติตามคำสั่งต่อไป

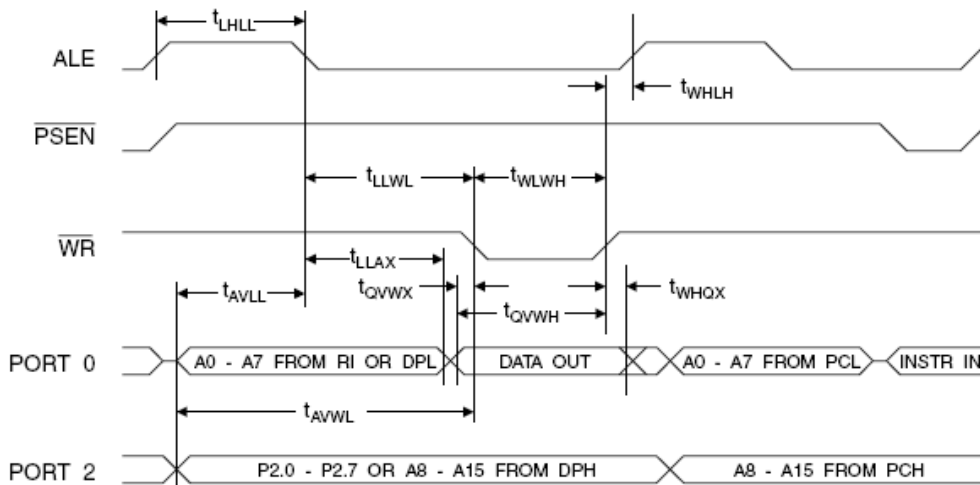
5.3 การต่อไอซีหน่วยความจำข้อมูลภายนอก

หน่วยความจำข้อมูลชนิด RAM เบอร์ 6264 มีขนาดความจุ 8 กิโลไบต์ มีขาแอดเดรสจำนวน 13 ขา กำหนดให้เป็นขา A0-A12 มีขาข้อมูล 8 บิต กำหนดให้เป็นขา D0-D7 โดยต่อสัญญาณที่ขา \overline{RD} ของไอซี

MCS-51 เข้ากับขา \overline{OE} ของหน่วยความจำข้อมูล 6264 และที่ขา \overline{WR} ของไอซี MCS-51 เข้ากับขา \overline{WE} ของไอซี 6264 การต่อกับหน่วยความจำข้อมูล มีทั้งการอ่านและเขียน ดังนั้นติดต่อโดยใช้สัญญาณ \overline{RD} เพื่ออ่าน และใช้สัญญาณ \overline{WR} เพื่อเขียน แสดงดังภาพที่ 5.30 ขั้นตอนการติดต่อกับหน่วยความจำข้อมูลสรุปได้ดังนี้



ก. ไคอะแกรมเวลาของการอ่านข้อมูล



ข. ไคอะแกรมเวลาของการเขียนข้อมูล

ภาพที่ 5.30 ไคอะแกรมเวลาของการอ่าน และเขียนข้อมูลจากหน่วยความจำข้อมูลภายนอก

ขั้นตอนการอ่านข้อมูล

- 1) ซีพียูส่งเลขที่อยู่ของหน่วยความจำข้อมูลที่ต้องการอ่านให้ตัวนับระบุตำแหน่งคำสั่ง
- 2) ค่าในตัวนับระบุตำแหน่งคำสั่งถูกส่งไปยังพอร์ต 0 และพอร์ต 2 (Address A0-A15)
- 3) ซีพียูส่งสัญญาณลอจิก 1 ออกที่ขา ALE เพื่อทำการแลตช์เลขที่อยู่ A0-A7 ไว้ที่ไอซี

- 4) ซีพียูส่งสัญญาณลอจิก 0 ออกที่ขา \overline{RD} เพื่ออ่านข้อมูลในแรม
- 5) ข้อมูลภายในแรมจะถูกอ่านลงสู่บัสข้อมูล D0-D7 (พอร์ต 0)
- 6) ซีพียูอ่านข้อมูลที่บัส D0-D7 เข้าไปเก็บที่รีจิสเตอร์ A

ขั้นตอนการเขียนข้อมูล

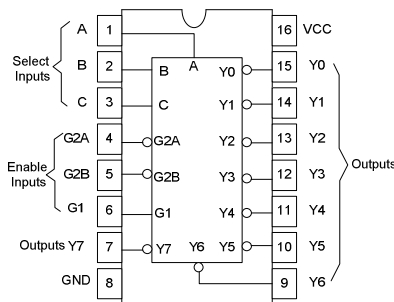
- 1) ซีพียูส่งเลขที่อยู่ของหน่วยความจำข้อมูลที่ต้องการเขียนให้ตัวนับระบุตำแหน่งคำสั่ง
- 2) ค่าในตัวนับระบุตำแหน่งคำสั่งถูกส่งไปยังพอร์ต 0 และพอร์ต 2 (Address A0-A15)
- 3) ซีพียูส่งสัญญาณ 1 ออกมาที่ขา ALE เพื่อทำการแลตช์แอดเดรส A0-A7 ไว้ที่ไอซี

74LS373

- 4) ซีพียูส่งข้อมูลที่จะเขียนลงแรม (Op Code) มาที่บัสข้อมูล D0-D7
- 5) ซีพียูส่งสัญญาณลอจิก 0 ออกมาที่ขา \overline{WR} เพื่อเขียนข้อมูลลงแรม
- 6) ข้อมูลบนบัสข้อมูล D0-D7 จะถูกเขียนลงแรมในเลขที่อยู่ A0 – A15

5.4 การต่อไอซีหน่วยความจำภายนอกหลายตัว

ไอซีเบอร์ 74LS138 หรือไอซี 1-Of-8 Decoder/De Multiplexer ทำหน้าที่ถอดรหัส โดยกำหนดที่ขา A B และ C เพื่อเลือกตำแหน่งแอดเดรสของไอซีหน่วยความจำ ใช้ในกรณีที่เลือกต่อไอซีหน่วยความจำมีเนื้อที่น้อยกว่า 64 กิโลไบต์ หรือใช้หน่วยความจำจำนวนหลายตัว แสดงดังภาพที่ 5.31



Enable		Select			Outputs							
G1	G2*	C	B	A	Y ₀	Y ₁	Y ₂	Y ₃	Y ₄	Y ₅	Y ₆	Y ₇
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

H ; high level, L ; low level, X ; irrelevant
 * ; G₂ = G_{2A} + G_{2B}

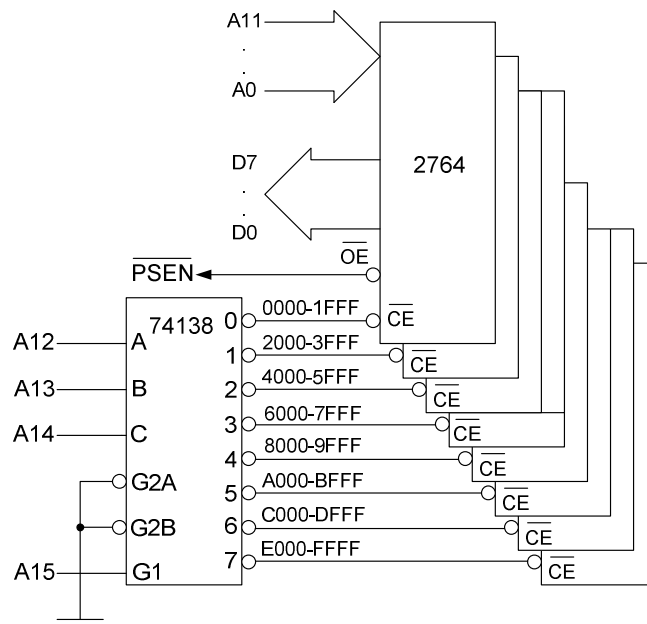
ภาพที่ 5.31 แสดงตำแหน่งขา และตารางการทำงานของไอซี 74LS138

(แหล่งอ้างอิง http://nitc.ac.in/nitc/dept/ece/public_html/student/digital/74LS138.pdf)

ไอซี MCS-51 ใช้ขาแอดเดรส 16 ขา ี้งหน่วยความจำได้ 65,536 ตำแหน่ง แต่ละตำแหน่งมีขนาด 1 ไบต์ ดังนั้น จึงสามารถ ติดต่อกับหน่วยความจำทั้งหมดได้ 64 กิโลไบต์ ถ้าใช้ไอซีหน่วยความจำความจุ 8 กิโลไบต์ จะใช้ได้ถึง 8 ตัว ซึ่งเลือกไอซีตัวใด หรือตำแหน่งแอดเดรสใด ขึ้นอยู่กับขา \overline{CE} ของไอซีตัวนั้น โดยกำหนดตำแหน่งแอดเดรสของหน่วยความจำ และขาของไอซี 74LS138 ดังแสดงในตารางที่ 5.4 และการต่อวงจรถอดรหัส กับหน่วยความจำตามตารางที่ 5.4 แสดงดังภาพที่ 5.32

ตารางที่ 5.4 แสดงตำแหน่งแอดเดรส และการ ต่อกับขา \overline{CE} ของไอซีหน่วยความจำแต่ละตัว

หน่วยความจำ ตัวที่	ตำแหน่งแอดเดรส	ขาที่ต่อกับ CE (Chip select) ของไอซีหน่วยความจำ	สถานะของขา		
			C =A15	B=A14	A=A13
1	0000H-1FFFFH	Y0	0	0	0
2	2000H-3FFFFH	Y1	0	0	1
3	4000H-5FFFFH	Y2	0	1	0
4	6000H-7FFFFH	Y3	0	1	1
5	8000H-9FFFFH	Y4	1	0	0
6	A000H-BFFFFH	Y5	1	0	1
7	C000H-DFFFFH	Y6	1	1	0
8	E000H-FFFFFFH	Y7	1	1	1



ภาพที่ 5.32 แสดงการต่อวงจรถอดรหัส สำหรับหน่วยความจำตามตารางที่ 5.4

6. การเขียนโปรแกรมหน่วงเวลาจากคำสั่ง

ระบบของไมโครคอนโทรลเลอร์ในแต่ละคำสั่งต้องใช้จำนวนรอบการทำงานแตกต่างกันไป การหาค่าเวลาที่ปฏิบัติตามคำสั่งนั้น คำนวณได้จากผลรวม ของรอบการทำงานทั้งหมดวิธีสร้างโปรแกรมหน่วงเวลา โดยกำหนดค่าให้กับรีจิสเตอร์ หรือหน่วยความจำ ให้ได้จำนวนรอบการทำงานตามต้องการ หลังจากนั้นจึงทำการลดค่าในรีจิสเตอร์ หรือหน่วยความจำลง ไปครั้งละ 1 ค่า แล้วตรวจสอบข้อมูลในรีจิสเตอร์จนกว่าจะมีค่าเท่ากับ 0 จึงออกจากโปรแกรมได้

การทำงานของแต่ละคำสั่งจะใช้เวลาในการประมวลผลที่แตกต่างกันไป โดยนับจำนวนรอบการทำงานหรือ แมกซ์ซินไซเคิล หาได้จากสูตร $T = MC \times 12/f_{xtal}$

โดยถ้าใช้คริสตอลค่า 12 MHz ใน 1 แมกซ์ซินไซเคิลไอซี MCS-51 ใช้สัญญาณนาฬิกา 12 คาบ ดังนั้นจึงใช้เวลาเท่ากับ $1 \times (12/12 \times 10^6) = 1 \mu\text{s}$ ถ้าใช้คริสตอลค่า 11.0592 MHz ใน 1 แมกซ์ซินไซเคิลจึงใช้เวลาเท่ากับ $1 \times (12/11.0592 \times 10^6) = 1.085 \mu\text{s}$

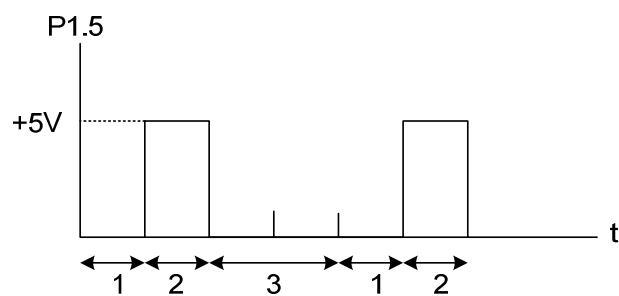
ตัวอย่างที่ 1 ให้หาค่าจำนวนแมกซ์ซินไซเคิล โดยแสดงเป็น Timing Diagram แสดงดังภาพที่ 5.33

```

MAIN: (1)  SETB  P1.5   จำนวนแมกซ์ซินไซเคิล = 1 MC
        (2)  CLR   P1.5   จำนวนแมกซ์ซินไซเคิล = 1 MC
        (3)  SJMP  MAIN  จำนวนแมกซ์ซินไซเคิล = 2 MC
        END
  
```

โปรแกรมทำงานดังนี้

$MC = (1)+(2)+(3)+(1)+(2)+(3)+\dots$ เขียนเป็น Timing Diagram ได้ดังนี้ (P1.5 มีค่าเริ่มต้นที่ LOW)



ภาพที่ 5.33 Timing Diagram ของคำสั่ง

6.1 การหาค่าจำนวนแมกซ์ซินไซเคิลใน 1 LOOP

```

DELAY:  MOV  R0, #N      ;จำนวนแมกซ์ซินไซเคิล = 1
        DJNZ R0, $      ;จำนวนแมกซ์ซินไซเคิล = 2
        RET              ;จำนวนแมกซ์ซินไซเคิล = 2
  
```

จากโปรแกรม

$$MC = \begin{array}{ccc} \downarrow 1MC & & \downarrow 2MC & & \downarrow 2MC \\ ((MOV R0, \#N) 1\text{ ครั้ง}) & + & ((DJNZ R0, \$) N \text{ ครั้ง}) & + & ((RET) 1\text{ ครั้ง}) \end{array}$$

สามารถสรุปเป็นสูตรได้ดังนี้

$$MC = 1+(2N)+2$$

ตัวอย่างที่ 2 กำหนดให้ $N = 05H$ ให้หาค่าเวลาจากจำนวนของแมชชีนไซเคิล

จากสูตร $MC = 1+(2N)+2$

แทนค่าในสูตร $MC = 1+(2 \times 5)+2$

$$MC = 13$$

ถ้าใช้คริสตอลค่า 12 MHz จะใช้เวลาเท่ากับ $13 \times (12/12 \times 10^6) = 13 \times 1 = 13 \mu S$

ถ้าใช้คริสตอลค่า 11.0592 MHz จะใช้เวลาเท่ากับ $13 \times (12/11.059 \times 10^6)$

$$13 \times 1.085 = 14.105 \mu S$$

ตัวอย่างที่ 3 ถ้ากำหนดให้ $N = 00H$ ให้หาค่าเวลาจากจำนวนของแมชชีนไซเคิล

จากสูตร $MC = 1+(2N)+2$

แทนค่าในสูตร $MC = 1+(2 \times 256)+2$

$$MC = 515$$

ถ้าใช้คริสตอลค่า 12 MHz และจะใช้เวลาเท่ากับ $515 \times (12/12 \times 10^6) = 515 \mu S$

ถ้าใช้คริสตอลค่า 11.0592 MHz และจะใช้เวลาเท่ากับ $515 \times (12/11.0592 \times 10^6) = 558.77 \mu S$

ข้อสังเกต การใช้ค่าคงที่ 00H ให้กับรีจิสเตอร์ R0 ในคำสั่ง DJNZ จะทำการลดค่าข้อมูลก่อนตรวจสอบ ดังนั้นค่า 00H จะลดค่าลง 1 ค่าเท่ากับ FFH ในรอบต่อไป จึงทำการลดค่าข้อมูลลงไปอีก จนกว่าข้อมูลจะมีค่าเป็น 00H อีกรอบ ดังนั้นจึงทำให้นับได้ถึง 256 ค่า

ตัวอย่างที่ 4 ต้องการความกว้างของพัลส์เท่ากับ 500 Hz ให้หาค่า N โดยกำหนดค่าคริสตอลเท่ากับ 12 MHz

จากสูตร $t = 1/f$

แทนค่าในสูตร $t = 1/500$

$$= 0.002 \text{ sec}$$

$$= 20 \mu S$$

ถ้าใช้คริสตอลค่า 12 MHz ($1MC = 1\mu S$) ต้องได้ค่าแมชชีนไซเคิลเท่ากับ 20 MC

จากสูตร $MC = 1+(2N)+2$

แทนค่าในสูตร $20 = 3+2N$

$$2N = 20-3$$

$$2N = 17$$

$$N = 7.5$$

คำสั่งครึ่งแมชชีนไซเคิลไม่มีใน MCS-51 เมื่อนำมาแทนค่าจึงไม่สามารถทำให้ลงตัวได้ ดังนั้นจึงเพิ่มคำสั่ง NOP อีก 1 คำสั่ง (No Operand) เป็นคำสั่งให้ซีพียูทำงาน 1 แมชชีนไซเคิลโดยไม่มีผลกระทบต่อซีพียูจึงแทนค่าและเขียนโปรแกรมได้ดังนี้

```

DELAY:   MOV  R0, #N      ; จำนวนแมชชีนไซเคิล = 1
         DJNZ R0, $      ; จำนวนแมชชีนไซเคิล = 2
         NOP              ; จำนวนแมชชีนไซเคิล = 1
         RET              ; จำนวนแมชชีนไซเคิล = 2

```

เขียนเป็นสมการได้ $MC = 1 + (2N) + 2 + 1(NOP)$

$$2N = 20 - 4$$

$$2N = 16$$

$$N = 8$$

ดังนั้นจึงต้องกำหนดค่าคงที่ให้กับรีจิสเตอร์ R0 เท่ากับ **08H**

6.2 การหาค่าจำนวนแมชชีนไซเคิลใน 2 LOOP

คำนวณจากจำนวนของ LOOP ในคูณด้วยจำนวนของ LOOP นอก

ตัวอย่างที่ 5

```

DELAY:   MOV  R0, #02      ; จำนวนแมชชีนไซเคิล = 1
DELAY1:  MOV  R1, #03      ; จำนวนแมชชีนไซเคิล = 1 x R0
         DJNZ R1, $        ; จำนวนแมชชีนไซเคิล = 2 x R1 x R0
         DJNZ R0, DELAY1   ; จำนวนแมชชีนไซเคิล = 2 x R0
         RET              ; จำนวนแมชชีนไซเคิล = 2

```

$$MC = 1 + R0 + 2R0R1 + 2R0 + 2$$

$$MC = 3 + R0 + 2R1R0 + 2R0$$

$$MC = 3 + R0 + 2R1R0$$

$$MC = 3 + R0 (3 + 2R1)$$

แทนค่า R0 ด้วย N1 และ R1 ด้วย N2 สามารถสรุปเป็นสูตรได้ดังนี้

$$MC = 3 + N1 (3 + 2N2)$$

ตัวอย่างที่ 6 กำหนดให้ $N2 = 0FFH$ และ $N1 = 02H$ คริสตอลความถี่ 12 MHz ให้หาค่าจำนวนแมชชีนไซเคิล และค่าของเวลารวมทั้งหมดในการทำคำสั่ง

โจทย์กำหนดให้ $N1 = 0FFH, N2 = 02H$ $f\text{-xtal} = 12\text{MHz}$

จากสูตร $MC = 3 + N1 (3 + 2N2)$

แทนค่าในสูตร $MC = 3 + 2 (3 + (2 \times 255))$

$MC = 3 + 2 (513)$

จำนวนแมชชีนไซเคิล $MC = 1029$

จากสูตร $T = MC \times 12 / f\text{-xtal}$

แทนค่าในสูตร $T = 1029 \times 12 / 12 \times 10^6 \text{ sec}$

ค่าเวลารวมทั้งหมด $T = 1.029 \text{ milliseconds}$

6.3 การหาค่าจำนวนแมชชีนไซเคิลใน 3 LOOP

การห้วงเวลาแบบ 3 LOOP จะสามารถแทนค่าได้จากสูตร

DELAY: MOV R0, N3 ; จำนวนแมชชีนไซเคิล = 1

DELAY1: MOV R1, N2 ; จำนวนแมชชีนไซเคิล = 1

DELAY2: MOV R2, N1 ; จำนวนแมชชีนไซเคิล = 1

DJNZ R2, \$; จำนวนแมชชีนไซเคิล = 2

DJNZ R1, DELAY2 ; จำนวนแมชชีนไซเคิล = 2

DJNZ R0, DELAY1 ; จำนวนแมชชีนไซเคิล = 2

RET ; จำนวนแมชชีนไซเคิล = 2

สามารถสรุปเป็นสูตรได้ดังนี้

$$MC = 1 + N3 [1 + N2 (1 + N1 (2) + 2) + 2]$$

7. การประยุกต์ใช้งาน บอร์ดควบคุมเครื่องใช้ไฟฟ้า 4 ช่องแบบกดเปิด กดปิด

7.1 คุณสมบัติการใช้งาน

การนำความรู้เรื่องการวัดและทดสอบวงจรใช้งานมาประยุกต์ใช้งานเป็นบอร์ดควบคุมเครื่องใช้ไฟฟ้าแบบกดเปิด กดปิด 4 ช่อง มีคุณสมบัติดังนี้

7.1.1 มีสวิตช์กดเพื่อควบคุมการทำงานของเครื่องใช้ไฟฟ้าได้ 4 ช่องอย่างอิสระ

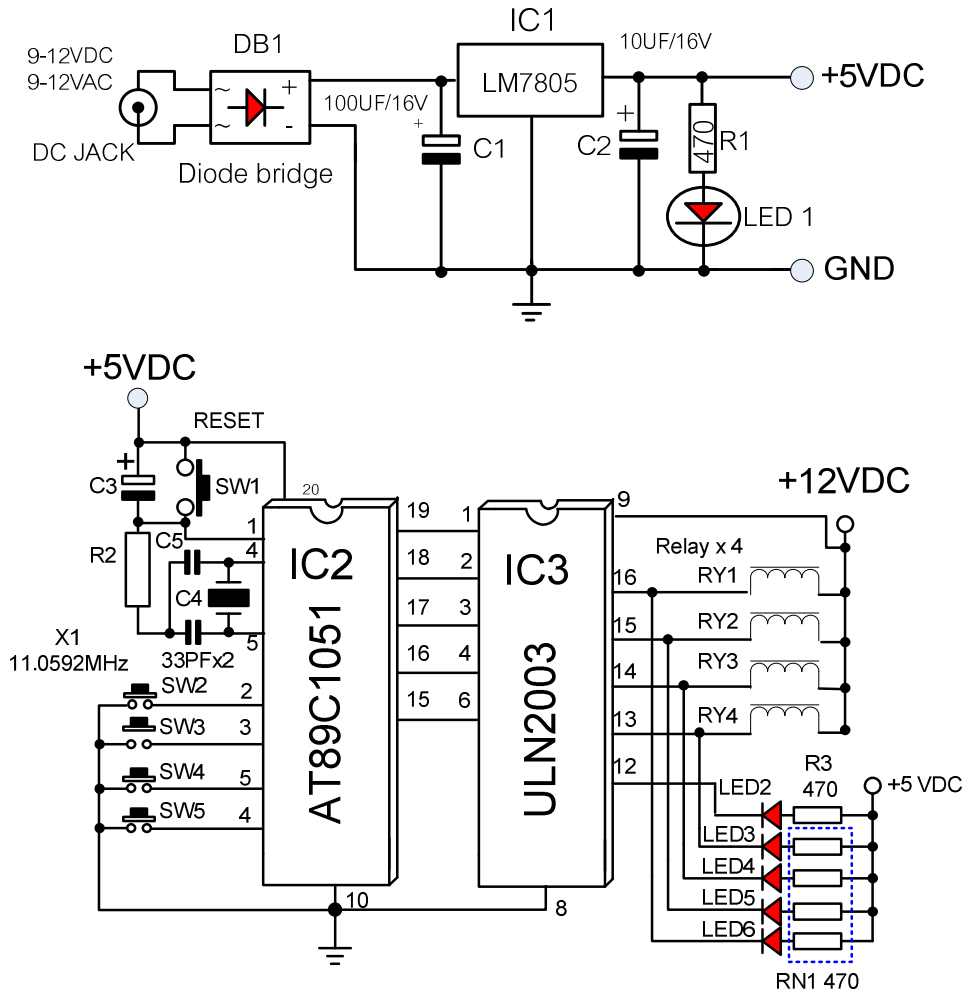
7.1.2 การเปลี่ยนช่อง ทำได้โดยกดสวิตช์ควบคุมของแต่ละช่อง

7.1.3 การกดเลือกช่องครั้งแรกจะเป็นการเปิด และเมื่อกดเลือกครั้งที่สองจะเป็นการปิด

7.1.4 มีแอลอีดีกระพริบเป็นสัญญาณแสดงการกด หรือเปลี่ยนช่องทุกครั้ง

7.1.5 มีแอลอีดีแสดงสถานะของการทำงานในแต่ละช่อง

7.2 การทำงานของวงจร



ภาพที่ 5.34 แสดงวงจรควบคุมเครื่องใช้ไฟฟ้าแบบกดเปิด กดปิด 4 ช่อง

การทำงานของวงจรในภาพที่ 5.34 เริ่มการทำงานจากแหล่งจ่ายไฟ ได้มาจากหม้อแปลงไฟ กระแสสลับ (AC) 9 โวลต์ หรือแหล่งจ่ายไฟกระแสตรง (DC) 12 โวลต์ โดยผ่านไดโอด DB1 เป็นวงจร บริดจ์เรกติไฟเออร์ เพื่อแปลงให้เป็นไฟกระแสตรง และยังใช้สำหรับจัดขั้วของแหล่งจ่ายไฟแบบกระแสตรง ในกรณีที่ไม่ทราบขั้วของแหล่งจ่ายไฟที่นำมาใช้งาน โดยปรับให้ขั้วที่ถูกต้อง ต่อเข้ากับ IC1 เบอร์ LM7805 ให้มีแรงดันเอาต์พุตคงที่ 5 โวลต์ ป้อนเป็นไฟเลี้ยงให้กับวงจร โดยมี LED1 เป็นตัวแสดงสถานะ ของการทำงาน คอนเดนเซอร์ C1, C2, C6 ทำหน้าที่กรองแรงดัน ส่วนแหล่งจ่ายอีกชุดหนึ่งจะมีแรงดัน ตามขนาดของรีเลย์ที่นำมาต่อ โดยนำไปป้อนให้กับรีเลย์ เพื่อควบคุมอุปกรณ์ไฟฟ้าจากภายนอก

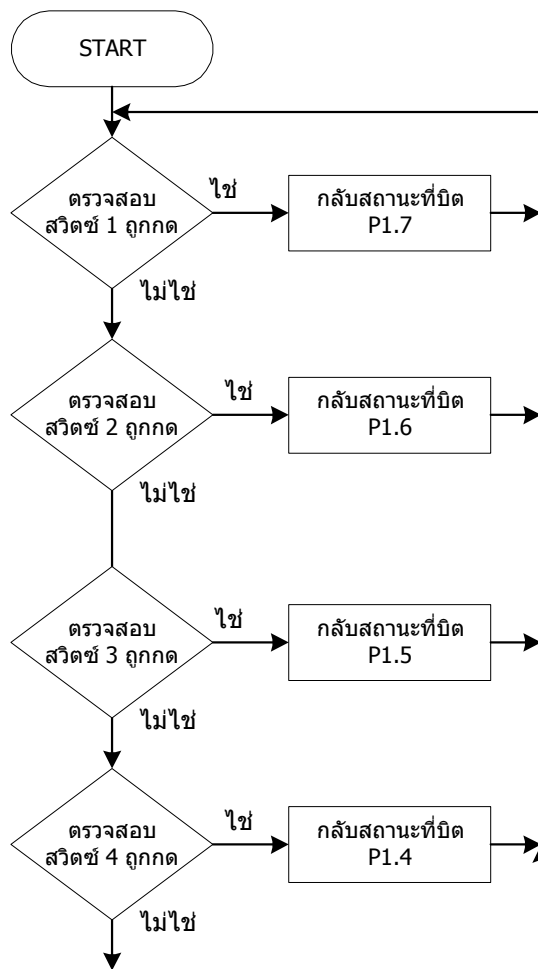
IC2 เป็นไอซีไมโครคอนโทรลเลอร์เบอร์ AT89C1051 ของ Atmel มีการโปรแกรมข้อมูล ขั้นตอนการทำงานไว้ภายใน C3, R2 และ SW1 เป็นส่วนของวงจรสร้างสัญญาณรีเซต และ C4, C5, X1 เป็นการสร้างสัญญาณนาฬิกาให้กับไอซี โดย IC2 จะทำหน้าที่ควบคุม และสั่งงานทั้งหมด สวิตช์ SW2, SW3,

SW4 และ SW5 ใช้กดเลือกการควบคุมในแต่ละช่อง โดยมีแอลอีดี LED3- LED5 ทำหน้าที่แสดงสถานะ การทำงานของแต่ละช่อง และ LED 6 แสดงผลเป็นการกระพริบในแต่ละครั้งที่กดสวิตช์

IC3 เบอร์ ULN2003 เป็นไอซีไดรเวอร์โดยภายในบรรจุอินเวอร์เตอร์เกต แบบคอลเล็กเตอร์เปิด จำนวน 7 ตัว สามารถใช้แรงดันสูงสุดได้ถึง 50 โวลต์ และให้กระแสเอาต์พุตได้สูงสุดในแต่ละขา 500 mA นอกจากนั้นยังมีไดโอดป้องกันไว้ที่ทุกขาของเอาต์พุต ทำให้สามารถต่อกับโหลดที่เป็นขดลวดได้โดยทำหน้าที่ขั้วรีเลย์ และ LED3 – LED6 ซึ่งทำหน้าที่แสดงผลการใช้งานของแต่ละช่อง

7.3 การเขียนโปรแกรม

โปรแกรมควบคุมเครื่องใช้ไฟฟ้าแบบกดเปิด กดปิด 4 ช่องแสดงเป็นผังงาน แสดงดังภาพที่ 5.35



ภาพที่ 5.35 แสดงผังงานของโปรแกรมควบคุมเครื่องใช้ไฟฟ้าแบบกดเปิด กดปิด 4 ช่อง

*** สวิตช์ควบคุมเครื่องใช้ไฟฟ้า 4 ช่อง *****

OUT1	BIT	P1.7	; เป็นไคเร็กทีฟที่ใช้กำหนดค่า OUT1 แทน P1.7
OUT2	BIT	P1.6	; เป็นไคเร็กทีฟที่ใช้กำหนดค่า OUT2 แทน P1.6
OUT3	BIT	P1.5	; เป็นไคเร็กทีฟที่ใช้กำหนดค่า OUT3 แทน P1.5
OUT4	BIT	P1.4	; เป็นไคเร็กทีฟที่ใช้กำหนดค่า OUT4 แทน P1.4
STROBE	BIT	P1.3	; เป็นไคเร็กทีฟที่ใช้กำหนดค่า STROBE แทน P1.3
SW1	BIT	P3.0	; เป็นไคเร็กทีฟที่ใช้กำหนดค่า SW1 แทน P3.0
SW2	BIT	P3.1	; เป็นไคเร็กทีฟที่ใช้กำหนดค่า SW2 แทน P3.1
SW3	BIT	P3.5	; เป็นไคเร็กทีฟที่ใช้กำหนดค่า SW3 แทน P3.5
SW4	BIT	P3.4	; เป็นไคเร็กทีฟที่ใช้กำหนดค่า SW4 แทน P3.4

BIT Symbol Name Bit Expression

เป็นไคเร็กทีฟที่ใช้กำหนดค่าแอดเดรสของตำแหน่งหน่วยความจำที่เข้าได้แบบบิต ให้กับสัญลักษณ์กรณีที่มีค่าอยู่ในหน่วยความจำตำแหน่งแอดเดรส 20H – 2FH หรือตำแหน่งของรีจิสเตอร์ที่เข้าได้แบบบิต

```
START:  ORG      0000H      ; กำหนดแอดเดรสเริ่มต้นของโปรแกรม
        MOV     P3,#0FFH   ; กำหนดให้พอร์ต P1 เป็นอินพุต กำหนดค่าเป็น "1" ทุกบิต
        MOV     P1,#00H    ; พอร์ต P1 มีค่า 00 เพื่อขณะเริ่มต้นทำให้ LED ดับทุกดวง
```

*** ตรวจสอบการเลือกกดคีย์สวิตช์ควบคุมเครื่องใช้ไฟฟ้า 4 ช่อง ***

```
SCAN:   JNB     SW1, RELAY1 ; ถ้ามีการกดสวิตช์ (P3.0 = "1") กระโดดไปที่เลเบล RELAY1
        JNB     SW2, RELAY2 ; ถ้ามีการกดสวิตช์ (P3.1 = "1") กระโดดไปที่เลเบล RELAY2
        JNB     SW3, RELAY3 ; ถ้ามีการกดสวิตช์ (P3.5 = "1") ให้กระโดดที่เลเบล RELAY3
        JNB     SW4, RELAY4 ; ถ้ามีการกดสวิตช์ (P3.4 = "1") ให้กระโดดที่เลเบล RELAY4
        SJMP    SCAN       ; ถ้าไม่มีการกดสวิตช์ใดๆ ให้กระโดดที่เลเบล SCAN ใหม่
```

JNB เป็นคำสั่งตรวจสอบการกดสวิตช์ของแต่ละตัว โดยมีเงื่อนไขเมื่อกดสวิตช์ หรือทำให้สถานะลอจิกของบิตที่กำหนดเป็นลอจิก “0” ให้กระโดดไปทำที่ตำแหน่งเลเบลที่กำหนด แต่ถ้าไม่มีการกดสวิตช์บิตใดๆ ให้กระโดดกลับไปทำการตรวจสอบสวิตช์ทุกตัวใหม่ สามารถเพิ่มจำนวนสวิตช์ให้สัมพันธ์กับรีเลย์ได้

*** ตรวจสอบการกดคีย์สวิตช์ เรียกใช้สัญญาณแสดงผลการกด และกลับสถานะของบิต **

```
RELAY1: ACALL    DELAY_D    ; เรียกโปรแกรม DELAY_D เพื่อหน่วงเวลาทดสอบการกด
        JB      SW1, SCAN   ; ตรวจสอบซ้ำถ้าไม่มีการกดสวิตช์ (P3.0 = "1") ให้ทำที่ SCAN
        ACALL   STROBE_SIG ; เรียกโปรแกรมย่อยแสดงผลการเปลี่ยนช่องที่ LED_STROBE
        JNB    SW1, $       ; ตรวจสอบการปล่อยคีย์ (P3.0 = "0") ให้ทำที่คำสั่งในบรรทัดเดิม
        CPL    OUT1        ; กลับค่าสถานะที่บิต OUT1
        SJMP   SCAN        ; กระโดดไปทำคำสั่งที่เลเบล SCAN ใหม่
```

```
RELAY2: ACALL    DELAY_D
        JB      SW2, SCAN
        ACALL   STROBE_SIG
        JNB    SW2, $
        CPL    OUT2
        SJMP   SCAN
```

```
RELAY3: ACALL    DELAY_D
        JB      SW3, SCAN
        ACALL   STROBE_SIG
        JNB    SW3, $
        CPL    OUT3
        SJMP   SCAN
```

```
RELAY4: ACALL    DELAY_D
        JB      SW4, SCAN
        ACALL   STROBE_SIG
        JNB    SW4, $
        CPL    OUT4
        SJMP   SCAN
```

ถ้าต้องการช่องควบคุมมากกว่า 4 ช่อง ให้ทำโปรแกรมในส่วนนี้เพิ่มเติมไปอีกได้ถึง 8 ช่อง (RELAY8 :)

```

;*****
;***   โปรแกรมย่อยแสดงการกระพริบที่แอลอีดี   ***
;*****

STROBE_SIG: SETB  STROBE      ; ให้สถานะของบิต STROBE เป็นลอจิก “1”
              ACALLDELAY      ; เรียกโปรแกรมย่อย DELAY เพื่อหน่วงเวลาการแสดงผล
              CLR   STROBE      ; ให้สถานะของบิต STROBE เป็นลอจิก “0”
              RET                ; ออกจากโปรแกรมย่อย

;*****
;***   โปรแกรมย่อยหน่วงเวลา แก้ปัญหาการรบกวนจากหน้าสัมผัสของสวิตช์   *****
;*****

DELAY_D:  MOV  R7,#0FFH      ; นำค่าคงที่ FFH ไปเก็บไว้เป็นข้อมูลของรีจิสเตอร์ R7
           DJNZ R0,$          ;
           RET                ; ออกจากโปรแกรมย่อยหน่วงเวลา

;*****
;***   โปรแกรมย่อยหน่วงเวลา   *****
;*****

DELAY:    MOV  R1,#0FFH      ; นำค่าคงที่ FFH ไปเก็บไว้เป็นข้อมูลของรีจิสเตอร์ R1
DELAY1:   MOV  R2,#0FFH      ; นำค่าคงที่ FFH ไปเก็บไว้เป็นข้อมูลของรีจิสเตอร์ R2
           DJNZ R2,$          ;
           DJNZ R1, DELAY1    ;
           RET                ; ออกจากโปรแกรมย่อยหน่วงเวลา

           END

```

7.4 การทดสอบ และใช้งาน

เมื่อเปิดเครื่องเพื่อใช้งานในครั้งแรกรีเลย์ทั้ง 4 ตัวยังไม่ทำงานใดๆ หลังจากนั้นให้กดสวิตช์ SW2 SW3, SW4 และ SW5 ทีละตัว สังเกตที่ LED6 จะกระพริบทุกครั้งที่เกิดสวิตช์ และกดสวิตช์ครั้งแรก รีเลย์ จะทำงาน หลังจากนั้นเมื่อกดสวิตช์ตัวเดิมอีกครั้ง ทำให้รีเลย์หยุดทำงาน สามารถสังเกตได้จากที่ LED2 - LED5 จะแสดงผลตามตำแหน่งของช่องที่ถูกเลือกใช้

7.5 การตรวจสอบ และแก้ไข

7.5.1 ถอด IC2 เบอร์ AT89C2051 และ IC3เบอร์ ULN2003 ออกจากบอร์ด

7.5.2 ต่อไฟ AC หรือ DC เข้าสู่วงจร สังเกตที่ LED1 จะต้องสว่าง แสดงถึงมีไฟ +VCC 5 โวลต์ ในวงจร ถ้า LED1 ไม่สว่างให้ตรวจสอบแรงดันที่ป้อนจากอแดปเตอร์, สวิตช์เปิด ปิด, ขั้วของไดโอดบริดจ์, ขั้วของตัวเก็บประจุ, ไอซี 7805 และ ขั้วของ LED1 ตามลำดับ

7.5.3 ใช้สายไฟขนาดเล็ก (สายโทรศัพท์) เสียบที่ขา 8 (Ground) ของบอร์ด IC3 จากนั้นให้นำปลายสายอีกด้านหนึ่งไปแตะกับขาที่ 12 ถึงขาที่ 16 จนครบทุกขา สังเกตที่ LED 2- LED6 ต้องสว่าง ถ้าไม่สว่างให้ตรวจสอบความถูกต้องของขั้ว LED และรีเลย์ต้องทำงานมีเสียงดังคลิก

7.5.4 ใส่ IC3 แล้วใช้สายไฟขนาดเล็กเสียบที่ขา 20 (+VCC) ของบอร์ด IC2 จากนั้นให้นำปลายสายอีกข้างหนึ่งไปแตะกับขาที่ 15 (P1.3) ถึงขาที่ 19 (P1.7) ของบอร์ด IC2 จนครบทุกขา สังเกต LED 2 - LED6 ต้องสว่าง ถ้าไม่สว่างให้ตรวจสอบความถูกต้องของขั้ว LED และรีเลย์ต้องทำงานมีเสียงดังคลิก ถ้าไม่ทำงานให้ตรวจสอบการต่อวงจรของไอซี ULN2003

7.5.5 หลังจากแก้ไขเรียบร้อยแล้ว ให้ใส่ IC2 แล้วทดลองตามขั้นตอนการทดสอบและใช้งานต่อไป การประยุกต์ใช้งาน แสดงดังภาพที่ 5.36 โดยการนำวงจรใส่ไว้ในชั้นไม้กลุเพื่อปิด เปิดระบบไฟฟ้าในห้องทำงาน หรือสามารถค้นคว้าวิธีการสร้างเพิ่มเติมได้ที่ <http://www.adisak51/project01.html>



ภาพที่ 5.36 การนำวงจรควบคุมเครื่องใช้ไฟฟ้าแบบกดเปิดกดปิด 4 ช่องประยุกต์ใช้งาน

สรุป

ไอซี MCS-51 ติดต่อกับอุปกรณ์ภายนอก เช่น การรับข้อมูลจากอุปกรณ์ภายนอก (Input) หรือการส่งข้อมูลให้กับอุปกรณ์ภายนอก (Output) ต้องติดต่อกันผ่านพอร์ต (Port) หรือกล่าวได้ว่าพอร์ตคือช่องทางสำหรับโอนย้ายข้อมูลระหว่างไอซี MCS-51 กับอุปกรณ์ภายนอก การอ่านค่าลอจิกจากพอร์ตทำได้ 2 ลักษณะด้วยกัน คือ การอ่านค่าจากขาพอร์ตโดยตรง (Read Pin) และการอ่านค่าจากวงจรถักค่าของแต่ละพอร์ต (Read Latch)

พอร์ตเอาต์พุต สามารถส่งข้อมูลเป็นลอจิก ที่ต้องการออกไปได้ เมื่อให้เอาต์พุตของพอร์ต มีสัญญาณออกเป็นลอจิก “0” และรับกระแสเข้ามาเรียกว่ากระแสซิงค์ ที่ตำแหน่งของบิต เมื่อให้เอาต์พุตมีสัญญาณออกเป็นลอจิก “1” มีกระแส จ่ายออกมาเรียกว่ากระแสซอร์ส

สัญญาณรบกวนที่เกิดขึ้น จากการสั้นของหน้าสัมผัสสวิตช์ขณะทำการกดสวิตช์ ทำให้เกิดการแกว่งของสัญญาณ เป็นปัญหาที่ต้องแก้ไขโดยใช้โปรแกรม หรือใช้วงจรมีการป้องกัน

การต่อวงจรโดยใช้ไอซีขนาด 40 ขา สามารถใช้ไอซีไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีหน่วยความจำโปรแกรมภายใน โดยเลือกขนาดของหน่วยความจำได้ตามคุณสมบัติของไอซี ดังนั้นขา \overline{EA} จึงต่อกับแหล่งจ่ายไฟ +VCC เพื่อมีสถานะลอจิก “1” เป็นการเลือกใช้หน่วยความจำภายใน

ไอซี MCS-51 เมื่อถูกรีเซ็ต ค่าในรีจิสเตอร์ใช้งานพิเศษ จะมีค่าเป็น “0” ยกเว้นรีจิสเตอร์ P0-P3, SP และ SBUF โดยค่า P0 – P3 เป็นลอจิก “1” ทุกบิต ค่าในรีจิสเตอร์ SP จะชี้ที่แอดเดรส 07H ส่วนค่าในรีจิสเตอร์ SBUF มีค่าที่ไม่แน่นอน และค่าของหน่วยความจำข้อมูลภายในจะไม่มีการเปลี่ยนแปลง แต่ถ้าหากมีการรีเซ็ตโดยเริ่มจ่ายพลังงานให้ไอซีเป็นครั้งแรก (Power-On Reset) ค่าของหน่วยความจำภายในจะมีค่าไม่แน่นอน ส่วนค่าในรีจิสเตอร์ PC จะชี้ ที่ตำแหน่ง 0000H สัญญาณรีเซ็ตโดยใช้วงจร โดยการรีเซ็ตโดยกดที่สวิตช์ (Manual Reset) และรีเซ็ตเมื่อเริ่มจ่ายพลังงานให้กับไอซี (Power-On Reset)

MCS-51 มีวงจรออสซิลเลเตอร์สำหรับสร้างสัญญาณนาฬิกา เพื่อกำหนดจังหวะการทำงานของซีพียู วงจรออสซิลเลเตอร์ใน MCS-51 แต่ละเบอร์ มีข้อแตกต่างกันออกไปขึ้นกับเทคโนโลยีที่ใช้ในการผลิต

ไอซี MCS-51 สามารถติดต่อกับหน่วยความจำโปรแกรมภายนอก และหน่วยความจำข้อมูลภายนอกได้ถึง 64 กิโลไบต์โดยใช้ขาแอดเดรสจำนวน 16 เส้น ขาข้อมูล 8 เส้น และขาควบคุมอีก 3 เส้นส่งออกมาทางพอร์ต 0 และพอร์ต 2 ของไอซี MCS-51 การทำงานของวงจรโดยพอร์ต 0 ถูกออกแบบให้เป็นได้ทั้งบัสแอดเดรส และบัสข้อมูล ถ้าใช้เป็นบัสแอดเดรส A0-A7 จะเป็นช่องทางส่งไบต์ต่ำ (Low Byte) และรองรับข้อมูล D0-D7 จากหน่วยความจำ สำหรับพอร์ต 2 เป็นบัสแอดเดรส A8-A15 ซึ่งเป็นช่องทางส่งไบต์บน

ระบบของไมโครคอนโทรลเลอร์ ในแต่ละคำสั่งต้องใช้จำนวนรอบการทำงานแตกต่างกันไป การหาค่าเวลาที่ปฏิบัติตามคำสั่งนั้น คำนวณได้จากผลรวม ของรอบการทำงานทั้งหมด วิธีสร้างโปรแกรมหน่วยเวลา โดยกำหนดค่าให้กับรีจิสเตอร์ หรือหน่วยความจำ ให้ได้จำนวนรอบการทำงานตามต้องการ หลังจากนั้นจึงทำการลดค่าในรีจิสเตอร์ หรือหน่วยความจำลงไปครั้งละ 1 ค่า แล้วตรวจสอบข้อมูลในรีจิสเตอร์จนกว่ามีค่าเท่ากับ 0 จึงออกจากโปรแกรมได้