

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 เซนเซอร์ชนิดใช้แสง [1]

เซนเซอร์ชนิดใช้แสง (Optical Sensor) หรือ Photo Sensor โดยทั่วไปใช้ในการตรวจจับ การเคลื่อนไหว การตรวจจับวัตถุ และการตรวจสอบขนาดรูปร่างของวัตถุ เซนเซอร์ชนิดนี้ทำงาน โดยอาศัยหลักการส่งและรับแสง มีส่วนประกอบสำคัญ 2 ส่วนคือ ตัวส่งแสง (Emitter) และตัวรับ แสง (Receiver) ลักษณะการตรวจจับเกิดจากการที่ลำแสงจากตัวส่งแสง ส่งไปสะท้อนกับวัตถุหรือ ถูกขวางกั้นด้วยวัตถุ ส่งผลให้ตัวรับแสงรู้สถานะที่เกิดขึ้นและเปลี่ยนแปลงสถานะของสัญญาณ ทางด้านเอาต์พุตเพื่อนำไปใช้งานต่อไปอุปกรณ์ที่เป็นตัวรับแสงส่วนใหญ่นิยมใช้โฟโตไดโอด (Photo diode) หรือโฟโต-ทรานซิสเตอร์ (Photo Transistor) ส่วนตัวส่งแสงนั้นโดยทั่วไปใช้ LED (Light Emitting Diode) เนื่องจากการต่อใช้งานร่วมกับวงจรอิเล็กทรอนิกส์ทำได้ง่าย สะดวกในการ บำรุงรักษา ใช้กระแสไฟฟ้าต่ำ และไม่ได้รับผลกระทบจากสภาวะรอบข้างไม่ว่าจะเป็น สุนัขแม่เหล็ก ความถี่ ความร้อน ความชื้น หรือการสั่นสะเทือน

2.1.1. แบ่งประเภทของ LED ตามความยาวคลื่นของแสง

1. LED แบบแสงอินฟราเรด มีความยาวคลื่นอยู่ในช่วง 910-950 nm ไม่สามารถ มองเห็นได้ด้วยตาเปล่า ให้ความเข้มของแสงสูงและระยะส่งไกล แต่ไม่สามารถ แยกแยะความแตกต่างของสีได้
2. LED แบบแสงสีแดง มีความยาวคลื่นประมาณ 650 nm มองเห็นได้ด้วยตาเปล่า ให้ความเข้มของแสงอยู่ในระดับปานกลาง สามารถตรวจจับพื้นผิวที่มีสีดำ สีน้ำ เงินและสีเขียวบนพื้นสีขาวได้ดี
3. LED แบบแสงสีเขียว มีความยาวคลื่นประมาณ 560 nm ให้ความเข้มของแสงต่ำ มีระยะการตรวจจับที่ไม่ไกล สามารถตรวจจับพื้นที่สีแดงบนพื้นสีขาวได้ดี

2.1.2. เซนเซอร์ชนิดใช้แสงประเภทตรวจจับโดยตรง (Diffuse-Reflective Optical Sensor)

เซนเซอร์ชนิดใช้แสงประเภทตรวจจับโดยตรง (Diffuse-Reflective Optical Sensor)
เซนเซอร์ชนิดใช้แสง (Optical Sensor/Photo Sensor) ประเภทนี้ตัวส่งแสงและตัวรับแสง

ติดตั้งรวมอยู่ในตัวเดียวกัน ตรวจสอบโดยการสะท้อนลำแสงโดยตรงกับตัววัตถุ และใช้วัตถุนั้นเป็นตัวสะท้อนลำแสงกลับมายังตัวรับแสงโดยไม่ต้องมีการปรับแต่งทิศทางลำแสง ระยะการตรวจจับไกลที่สุดเมื่อเปรียบเทียบกับแบบอื่น ๆ มีระยะการตรวจจับประมาณ 1 เมตร เซนเซอร์ประเภทนี้นิยมใช้มากที่สุด เนื่องจากสะดวกและง่ายในการติดตั้ง ใช้เนื้อที่ในการติดตั้งน้อย ราคาถูก ไม่ต้องใช้อุปกรณ์อื่นเพิ่มเติมเพราะมีทั้งตัวส่งและตัวรับอยู่ในตัวเดียวกัน สามารถตรวจจับวัตถุได้เกือบทุกชนิด ยกเว้น วัตถุที่มีลักษณะพื้นผิวด้านหรือดูดกลืนแสงและวัตถุที่โปร่งแสง เหมาะสำหรับการตรวจจับวัตถุที่มีลักษณะพื้นผิวเรียบเป็นมันวาวและทึบแสง ระยะการตรวจจับขึ้นอยู่กับลักษณะของวัตถุที่ต้องการตรวจจับ เช่น สี และความเรียบมันของวัตถุ โดยระยะการตรวจจับจะลดลงอย่างมากถ้าเป็นวัตถุสีดำ

2.1.3. เซนเซอร์ชนิดใช้แสงประเภทลำแสงสะท้อนกลับ (Retro-Reflective Optical

Sensor)

เซนเซอร์ชนิดใช้แสงประเภทลำแสงสะท้อนกลับ (Retro-Reflective Optical Sensor) เป็นเซนเซอร์แสง (Optical Sensor/Photo Sensor) ที่อาศัยหลักการสะท้อนกลับของลำแสง โดยตัวส่งแสงและตัวรับแสงติดตั้งรวมอยู่ในตัวเดียวกันเช่นเดียวกับแบบตรวจจับโดยตรง ต่างกันที่เซนเซอร์ประเภทนี้ต้องใช้งานร่วมกับแผ่นสะท้อนแสง (Reflector) เพื่อประโยชน์ในการเพิ่มความเข้มของแสงให้มากขึ้น มีผลทำให้ระยะทางในการตรวจจับเพิ่มมากขึ้นด้วย มีระยะการตรวจจับประมาณ 3-10 เมตร ข้อดีของเซนเซอร์ใช้แสง (Optical Sensor/Photo Sensor) ประเภทนี้ คือ การปรับแต่งทิศทางลำแสงทำได้ง่าย ติดตั้งง่าย มีระยะการตรวจจับปานกลาง การตรวจจับและระยะการตรวจจับไม่ขึ้นกับสีของวัตถุ แต่ไม่สามารถตรวจจับวัตถุที่มีผิวมันเงาและโปร่งแสงได้ เหมาะกับวัตถุที่มีลักษณะพื้นผิวที่ดูดกลืนแสงและไม่สามารถสะท้อนแสงกลับมายังตัวรับ หรืออาจใช้กับวัตถุที่มีลักษณะพื้นผิวขรุขระก็ได้เช่นกัน

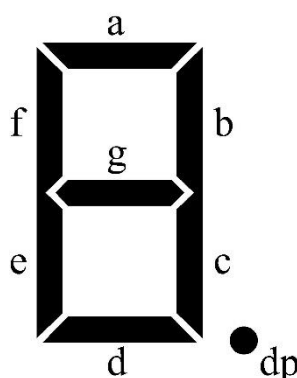
2.1.4. เซนเซอร์ชนิดใช้แสงประเภทลำแสงผ่านตลอด (Through-Beam Optical Sensor)

เซนเซอร์ชนิดใช้แสงประเภทลำแสงผ่านตลอด (Through-Beam Optical Sensor) เป็นเซนเซอร์แสง (Optical Sensor/Photo Sensor) ที่อาศัยวิธีการตัดต่อลำแสงเมื่อมีวัตถุที่ต้องการตรวจจับเคลื่อนที่ผ่านระหว่างตัวรับและตัวส่ง เซนเซอร์ประเภทนี้ติดตั้งตัวส่งแสงกับตัวรับแสงแยกกัน และต้องจ่ายไฟให้ทั้งตัวส่งและตัวรับ และเป็นเซนเซอร์ชนิดใช้แสงที่มีระยะการตรวจจับไกลที่สุด อย่างไรก็ตาม เซนเซอร์ประเภทนี้ติดตั้งยาก และมีราคาสูง

กว่าประเภทตรวจจับโดยตรงและลำแสงผ่านตลอด โดยสีและความมันวาวของวัตถุไม่มีผลต่อการตรวจจับ สามารถตรวจจับได้เฉพาะวัตถุที่มีลักษณะพื้นผิวทึบแสงเท่านั้น ไม่สามารถตรวจจับวัตถุที่มีสภาพผิวโปร่งแสงได้ ข้อควรระวังสำหรับการติดตั้งเซนเซอร์แสงประเภทนี้ คือ ต้องปรับตั้งศูนย์ของตัวรับและตัวส่งให้ตรงกันเสมอ

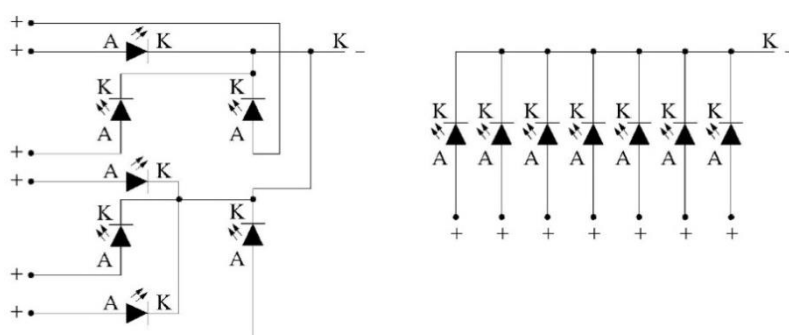
2.2 ตัวแสดงผล 7 ส่วน (7 Segment) [2]

ตัวแสดงผล 7 ส่วน หรือที่เราเรียกว่า 7 Segment เป็นอุปกรณ์อิเล็กทรอนิกส์ประเภท Display เช่นเดียวกับไดโอดเปล่งแสง หรือ LED ตัว 7 Segment เองนั้นภายในก็คือ LED 7ตัว(หรือมากกว่า) มาต่อกันเป็นรูปตัวเลข 8 ที่ตัว ส่วนแสดงผล 7 Segment จะมีชื่อกำกับอยู่โดยจะไล่จาก a,b, c, d, e, f, g และจุด เป็นดังนี้

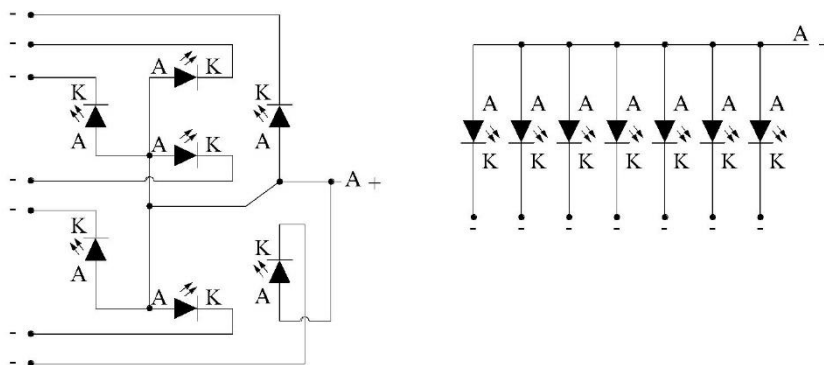


ภาพที่ 2.1 ตำแหน่งส่วนแสดงผล a- g

การต่อ LED ภายในตัว 7 Segment นั้นจะมีอยู่ 2 คอมมอนหลักๆ คือ แบบคอมมอน A (อานโท) และแบบคอมมอน K (คาโทท) การต่อแบบคอมมอน A เราจะใช้ขั้วลบ (-) ป้อนให้ที่ขา A - G ส่วนไฟบวก (+) จะมาป้อนที่จุดรวมของขา A การต่อแบบคอมมอน K เราจะใช้ ขั้วบวก (+) ป้อนให้ที่ขา A - G ส่วนไฟลบ (-) จะมาป้อนที่จุดรวมของขา K

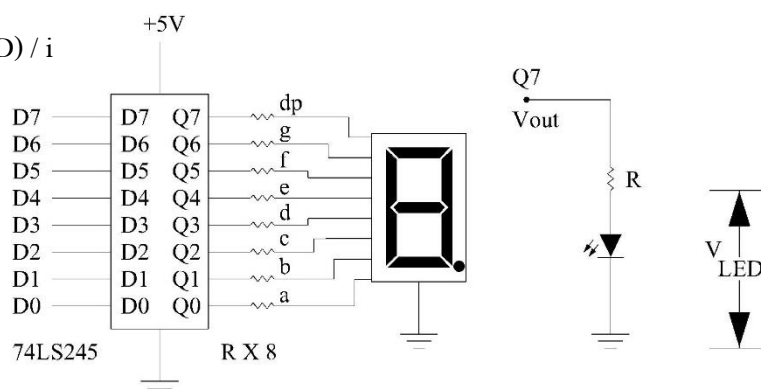


ภาพที่ 2.2 การนำเอา LED มาต่อกัน แบบคอมมอน K



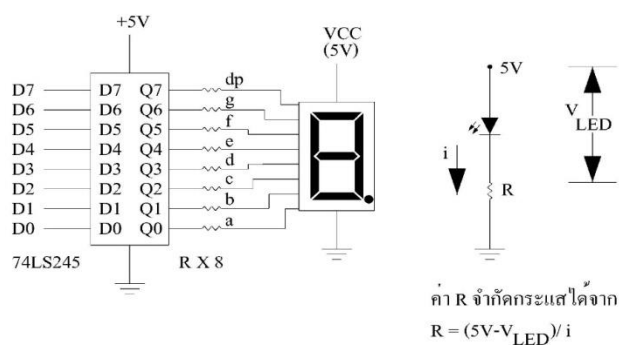
ภาพที่ 2.3 การนำเอา LED มาต่อกัน แบบคอมมอน A

การนำ LED ไปใช้แสดงผลแบบ Common Cathode ในแต่ละ Segment จะต้องมีตัวต้านทานจำกัดกระแส โดยผู้ออกแบบจะต้องกำหนดว่าในแต่ละ Segment จะให้กระแสไฟฟ้าไหลได้เท่าใดและเอาท์พุทที่ขับจะต้องให้ลอจิก “1” และ V_{out} ของไอซีที่ขับขณะเป็นลอจิก “1” จะต้องเป็นเท่าใด $R = (V_{out} - V_{LED}) / i$



ภาพที่ 2.4 แบบ Common Cathode ในแต่ละ Segment

แบบ Common Anode LED ตัวเลข 7 ส่วนแบบ Common Anode จะต้องขับด้วยการให้ลอจิก “0” แก่ Anode โดยมีตัวต้านทาน R จำกัดกระแส



ภาพที่ 2.5 แบบ Common Anode ในแต่ละ Segment

2.3 Microcontroller MCS51 [3]

คุณสมบัติทั่วไปที่สำคัญของไมโครคอนโทรลเลอร์ตระกูล MCS-51 มีดังนี้

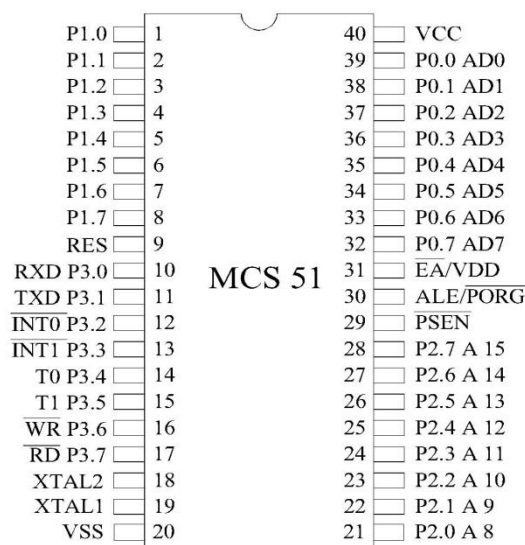
- เป็นไมโครคอนโทรลเลอร์ขนาด 8 บิต
- มีวงจรออสซิลเลเตอร์และวงจรผลิตสัญญาณนาฬิกาภายในไอซี
- มีขาสัญญาณอินพุตเอาต์พุตจำนวน 32 บิต
- สามารถเชื่อมต่อหน่วยความจำข้อมูลภายนอก (External Data Memory) โดยอ้างตำแหน่งแอดเดรสได้ถึง 64
- สามารถเชื่อมต่อหน่วยความจำโปรแกรมภายนอก (External Program Memory) โดยอ้างตำแหน่งแอดเดรสได้ถึง 64 K
- มีหน่วยความจำภายในตัว (On-Chip Program Memory) ขนาด 4 K โดยเฉพาะเบอร์ 8052 จะมีหน่วยความจำในส่วนนี้ถึง 8 K
- มีหน่วยความจำข้อมูลภายในตัว (On-Chip Data Memory) ขนาด 128 ไบต์
- หน่วยความจำข้อมูลภายในบางส่วนสามารถเข้าถึงข้อมูลระดับบิตได้ด้วย ทำให้การควบคุมหรือการตรวจสอบสถานะบิตทำได้ง่าย ส่งผลให้การเขียนโปรแกรมทำได้ง่ายมากขึ้น
- มีไทม์เมอร์/เคาน์เตอร์ (Timer/Counters) ขนาด 16 บิต จำนวน 2 ตัว โดยเฉพาะเบอร์ 8032 หรือ 8052 จะมีไทม์เมอร์/เคาน์เตอร์จำนวน 3 ตัว
- การอินเตอร์รัปต์สามารถทำได้จาก 5 แหล่งกำเนิด โดยเฉพาะเบอร์ 8032 และ 8052 จะทำการอินเตอร์รัปต์ได้จาก 6 แหล่งกำเนิด โดยการอินเตอร์รัปต์ยังสามารถจัดระดับความสำคัญได้เป็น 2 ระดับ
- มีพอร์ตสื่อสารอนุกรมภายในตัวเอง ซึ่งทำงานเป็นแบบฟูลดูเพล็กซ์ (Full Duplex)
- มีคำสั่งในการคำนวณทางคณิตศาสตร์และทางตรรกศาสตร์
- คำสั่งโดยส่วนใหญ่ใช้เวลาการทำงานเพียง 1 ไมโครวินาที เมื่อใช้คริสตอลความถี่ 12 เมกะเฮิร์ตซ์
- ต้องการแหล่งจ่ายไฟ 5 โวลต์ เพียงชุดเดียว

2.3.1 โครงสร้างภายนอกของ MCS-51

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ทุกเบอร์จะมีตำแหน่งขาพื้นที่เหมือนกัน ดังแสดงในภาพที่ 2.6 สำหรับหน้าที่การใช้งานของแต่ละขามีดังนี้

1. ขา Vcc เป็นขาป้อนแรงดันไฟเลี้ยง + 5 โวลต์
2. ขา Vss เป็นขากราวด์

3. ขาพอร์ต 0 (Port 0) มี 8 ขา ได้แก่ขา P0.0 – P0.7 เป็นขาพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทางสำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้ขาพอร์ตเหล่านั้นอยู่ในสถานะปล่อยลอย ซึ่งในสถานะนี้เองที่สามารถนำมาใช้ เป็นพอร์ตอินพุตอิมพีแดนซ์ได้ นอกจากพอร์ตนี้ จะใช้งานเป็นพอร์ตอินพุตเอาต์พุตแล้วมันยังถูกใช้งานในการติดต่อกับหน่วยความจำภายนอกด้วย โดยทำหน้าที่ในการกำหนดตำแหน่งแอดเดรสไบต์ต่ำ (A0-A7) ซึ่งจะใช้งานเป็น แบบมัลติเพล็กซ์กับการรับส่งข้อมูลขนาด 8 บิต (D0-D7)
4. ขาพอร์ต 1 (Port 1) มี 8 ขา ได้แก่ขา P1.0-P1.7 เป็นขาพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทาง สำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้เป็นพอร์ตอินพุต นอกจากนี้สำหรับเบอร์ 8032 และ 8052 ขาพอร์ต P1.0 และ P1.1 จะถูกนำมาใช้งานเป็นขา T2 และ T2EX ตามลำดับด้วย



ภาพที่ 2.6 การจัดตำแหน่งขาต่าง ๆ ของไมโครคอนโทรลเลอร์ MCS-51

5. ขาพอร์ต 2 (Port 2) มี 8 ขา ได้แก่ขา P2.0-P2.7 เป็นขาพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทางสำหรับใช้งานทั่วไป โดยถ้าใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้เป็นพอร์ตอินพุต นอกจากพอร์ตนี้ จะใช้งานเป็นพอร์ตอินพุตเอาต์พุตแล้วมันยังถูกใช้งานในการติดต่อกับหน่วยความจำภายนอกด้วย โดยทำหน้าที่ในการกำหนดตำแหน่งแอดเดรสไบต์สูง (A8-A15)

6. ขาพอร์ต 3 (Port 3) มี 8 ขาได้แก่ขา P3.0-P3.7 เป็นขาพอร์ตอินพุตเอาต์พุตแบบ 2 ทิศทางสำหรับใช้งานทั่วไปโดยถ้าใช้งานเป็นอินพุตพอร์ตต้องทำการเขียนค่า 1 ไปยังแต่ละบิตของพอร์ต เพื่อกำหนดให้เป็นพอร์ตอินพุต นอกจากพอร์ตนี้จะใช้งานเป็นพอร์ตอินพุตเอาต์พุตแล้วมันยังถูกใช้งานในหน้าที่พิเศษต่าง ๆ ดังแสดงในตารางที่ 2.1
7. ขารีสต (RST) ใช้สำหรับการรีเซ็ตการทำงานของไมโครคอนโทรลเลอร์ โดยการรีเซ็ตต้องคงสถานะเป็น 1 อย่างน้อยนาน 2 แมกซ์ไซเคิล ในขณะที่ออสซิลเลเตอร์ยังทำงานอยู่
8. ขา $\overline{\text{ALE/PROG}}$ เป็นขาสัญญาณเพื่อทำหน้าที่ควบคุมการแลตช์ (Latch) ค่าตำแหน่งแอดเดรสไบต์ต่ำ (Address Latch Enable) เมื่อต้องการติดต่อกับหน่วยความจำภายนอก นอกจากนี้ขานี้ยังทำหน้าที่เป็นอินพุตรับพัลส์ในการโปรแกรม (Program Pulse Input) ในส่วนของหน่วยความจำ EPROM สำหรับไมโครคอนโทรลเลอร์ในตระกูล MCS-51 ที่มีหน่วยความจำโปรแกรมภายในเป็น EPROM

ตารางที่ 2.1_หน้าที่พิเศษของแต่ละขาของพอร์ต P3

ขาพอร์ต	หน้าที่พิเศษ
P _{3.0}	EXD (Serial input port)
P _{3.1}	$\overline{\text{TXD}}$ (Serial output port)
P _{3.2}	$\overline{\text{INT0}}$ (external interrupt 0)
P _{3.3}	INT1 (external interrupt 1)
P _{3.4}	TO (Timer 0 external input)
P _{3.5}	$\overline{\text{T1}}$ (Timer 1 external input)
P _{3.6}	$\overline{\text{WR}}$ (external data memory write strobe)
P _{3.7}	$\overline{\text{RD}}$ (external data memory read strobe)

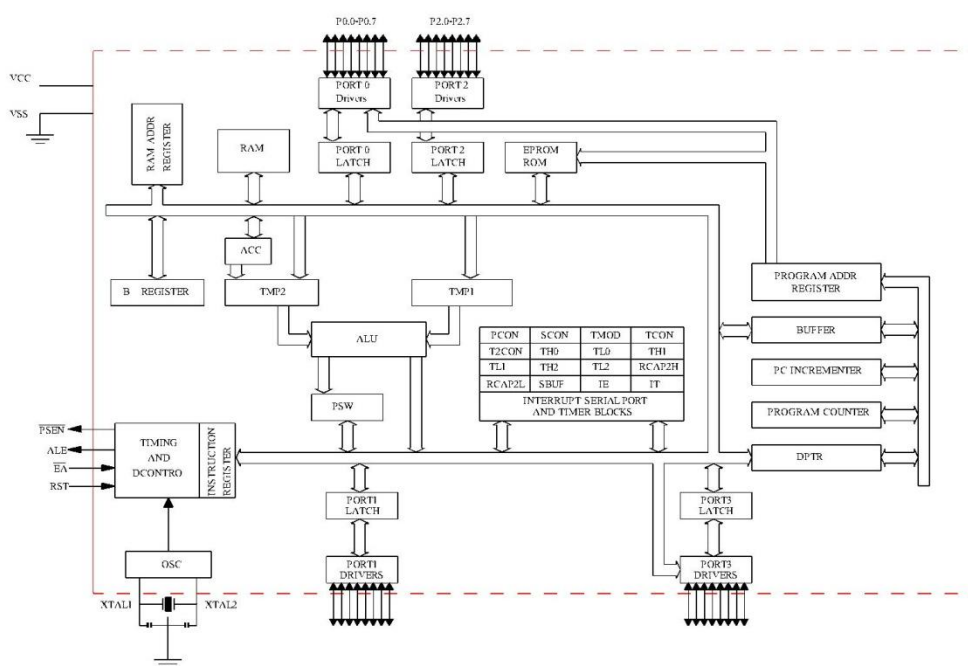
9. ขา $\overline{\text{PSEN}}$ (Program Store Enable) ทำหน้าที่เป็นสัญญาณสโตรบเพื่ออ่านคำสั่งจากหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ประมวลผลคำสั่งจากหน่วยความจำภายนอก ขานี้จะส่งสัญญาณสโตรบ

จำนวน 2 ครั้งในแต่ละแมชชีนไซเคิล แต่ในขณะที่ติดต่อกับหน่วยความจำ ข้อมูลภายนอกจะไม่มีการส่งสัญญาณ สโตรบแต่อย่างใด

10. ขา \overline{EA} / VPP (External Access Enable/VPP) เป็นขาสำหรับการเลือกใช้หน่วยความจำโปรแกรมจากภายในหรือจากภายนอก โดยถ้ามีสถานะเป็น 0 จะหมายถึงให้ไมโครคอนโทรลเลอร์รับคำสั่งจากหน่วยความจำภายนอกที่ตำแหน่งแอดเดรส 0-FFFFH (ถ้าเป็นเบอร์ 8052) อย่างไรก็ตามถ้าบิตป้องกัน (Security Bit) ในหน่วยความจำ EPROM ถูกโปรแกรมไว้ ไมโครคอนโทรลเลอร์จะไม่รับคำสั่งจากหน่วยความจำภายนอกเลย นอกจากนี้ขานี้ยังทำหน้าที่รับแรงดันไฟสำหรับการโปรแกรม (Vpp) ขนาด 21 โวลต์ เพื่อใช้ในระหว่างการโปรแกรม EPROM
11. ขา XTAL₁ และขา XTAL₂ เป็นขาอินพุตและเอาต์พุตของวงจรอินเวอร์ตออสซิลเลเตอร์แอมพลิไฟเออร์ (Inverting Oscillator Amplifier) สำหรับใช้ต่อร่วมกับคริสตัลภายนอก

2.3.2 โครงสร้างภายในของ MCS-51

โครงสร้างภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51 แสดงดังในภาพที่ 2.7

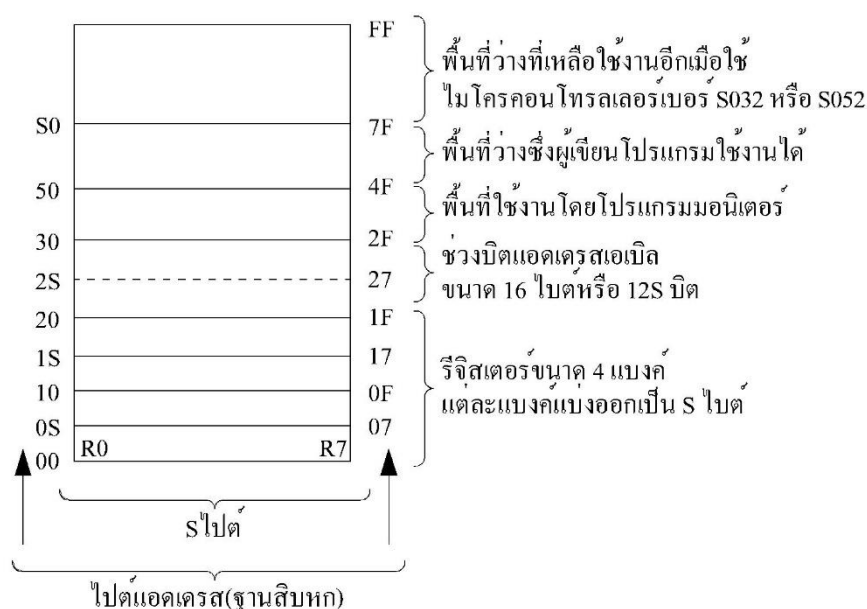


ภาพที่ 2.7 โครงสร้างภายในของไมโครคอนโทรลเลอร์ตระกูล MCS-51

2.3.3 หน่วยความจำสำหรับเก็บโปรแกรม

หน่วยความจำสำหรับเก็บโปรแกรมสามารถขยายได้สูง 64 กิโลไบต์ มีหน้าที่เก็บคำสั่งต่างๆ สำหรับไมโครคอนโทรลเลอร์ มันสามารถใช้เก็บตารางข้อมูลและค่าคงที่ได้ในการใช้งานไมโครคอนโทรลเลอร์โดยใช้หน่วยความจำโปรแกรมภายนอกเท่านั้น ที่ขา 31 หรือค่า EA (External Access Enable) จึงถูกต่อลงกราวด์ไว้เพื่อกำหนดให้ไมโครคอนโทรลเลอร์ไม่ใช้งานหน่วยความจำโปรแกรมภายในที่มีอยู่แล้ว และเมื่อไมโครคอนโทรลเลอร์ต้องการติดต่อกับหน่วยความจำโปรแกรมภายนอกมันจะส่งสัญญาณลอจิก Low ที่ขา 29 หรือขา $\overline{\text{PSEN}}$ ออกมา

หน่วยความจำโปรแกรมไม่จำเป็นต้องเป็นรอมหรืออีพรอม เช่นเดียวกับตำแหน่งแอดเดรสที่ว่างแต่ละแอดเดรส อาจอยู่ในรูปของหน่วยความจำหรือเป็นตำแหน่งของพอร์ตอินพุตเอาต์พุตก็ได้ หน่วยความจำโปรแกรมในที่นี้ถูกแบ่งออกเป็น 2 ช่วงดังนี้ คือช่วงแอดเดรสต่ำ 00000H ถึง 04000H เป็นส่วนของอีพรอม IC₅ และช่วงแอดเดรสจาก 04000H ถึง 08000H เป็นหน่วยความจำแรม IC₆ ของระบบ คำสั่งต่างๆ จะถูกป้อนให้ไปเก็บไว้และทำการประมวลผลจากที่แรมนี้ ภาพที่ 2.8 แผนผังที่แสดงการจัดแบ่งช่วงแอดเดรสใช้งานของแรมภายใน ซึ่งมีขนาด 128 ไบต์ หรือ 256 ไบต์



ภาพที่ 2.8 แผนผังที่แสดงการจัดแบ่งช่วงแอดเดรสใช้งานของแรมภายใน ซึ่งมีขนาด 128 ไบต์ หรือ 256 ไบต์ขึ้นอยู่กับเบอร์ของไมโครคอนโทรลเลอร์

2.3.4 การเขียนโปรแกรมเพื่ออินเตอร์เฟซผ่านพอร์ตอนุกรม

ไมโครคอนโทรลเลอร์ตระกูล MCS-51 ได้บรรจุวงจรในส่วนการอินเตอร์เฟซแบบอนุกรมไว้ในตัวด้วยซึ่งในส่วนนี้นับได้ว่าเป็นอีกภาคหนึ่งที่มีความซับซ้อนมากถ้าจะกล่าวกันจนหมดเปลือกจริงคงต้องยืดยาวมาก อีกทั้งฮาร์ดแวร์หรือบอร์ด MCS-51 ก็ได้ออกแบบวงจรในส่วนนี้ค่อนข้างพื้นฐานมากสำหรับการใช้งาน และในโปรแกรมมอนิเตอร์เองก็มีรูทีนสำหรับเรียกใช้งานไว้เสร็จสรรพแล้วดังนั้นคงจะกล่าวได้แต่เพียงการทำงานพื้นฐานและการเขียนโปรแกรมใช้งานกับ MCS-51 บอร์ดนี้เฉพาะโหมดที่ใช้งานกันเท่านั้น ผู้อ่านท่านใดอยากจะศึกษาอย่างลึกซึ้งคงต้องหาอ่านจากหนังสือเชิงทฤษฎีประกอบด้วย เพื่อเพิ่มเติมความรู้

การอินเตอร์เฟซผ่านพอร์ตอนุกรมสามารถเลือกทำงานได้หลายโหมด ซึ่งในบางโหมคนั้นอาจไม่ต้องให้ความสนใจเลยก็ได้ เพราะมันได้ถูกเลือกก็ต่อเมื่อทำงานเป็นระบบไมโครคอนโทรลเลอร์หลายตัวเชื่อมกันเป็นเน็ตเวิร์ก เราจะเริ่มต้นเรียนรู้กันที่โหมดการทำงานพื้นฐานที่ก่อนคือโหมดที่ทำการส่งและรับข้อมูล 8 บิต แบบอะซิงโครนัส (Asynchronously) โดยมีบิตเริ่มต้น (Start) และบิตสิ้นสุด (Stop Bit) กำหนดให้ขนาด 1 บิต และไม่มีการตรวจสอบพาริตี (Parity) และโหมดนี้เองที่ใช้ติดต่อสื่อสารกับเครื่องพีซีที่ใช้งานโดยตลอด ยิ่งกว่านั้นโหมดนี้ยังถูกงานในการส่งข้อมูลแบบ MIDI (Musical Instrument Digital Interface)

การอินเตอร์เฟซผ่านพอร์ตอนุกรมมีรีจิสเตอร์พิเศษ (SFR) ควบคุมการทำงานคือรีจิสเตอร์ SCC (Serial Control) อยู่ที่ตำแหน่งแอดเดรส 098H ทำหน้าที่หลักคือควบคุมการอินเตอร์เฟซแบบอนุกรมในตัวไมโครคอนโทรลเลอร์และเป็นตัวกำหนดโหมดการทำงาน ซึ่งแสดงหน้าที่ของแต่ละตำแหน่งบิตของรีจิสเตอร์ SCC ในตารางที่ 2.2

ตารางที่ 2.2 หน้าที่การทำงานของบิตต่าง ๆ ในรีจิสเตอร์ SCON

(MSB)

SM0	SM1	SM2	REN	TB8	RB8	T1	R1
-----	-----	-----	-----	-----	-----	----	----

*บิต SM0 และ SM1 ทำหน้าที่กำหนดโหมดการทำงานของพอร์ตสื่อสารแบบอนุกรม

SM0	SM1	โหมด	ลักษณะการทำงาน	อัตราบอดเรต
0	0	0	ชิพรีจิสเตอร์	$F_{osc} / 12$
0	1	1	8 บิต UART	กำหนดค่าเองได้
1	0	2	9 บิต UART	$F_{osc} / 32$ หรือ $F_{osc} / 64$
1	1	3	9 บิต UART	กำหนดค่าเองได้

UART = Universal Asynchronous Receiver/Transmitter

* บิต SM2 จะถูกอีนابلเมื่อทำการติดต่อสื่อสารระหว่างโปรเซสเซอร์หลายตัว (Multiprocessor) ในโหมด 2 และ 3 โดยถ้า SM2 ถูกเซตนั้นคือบิต RI ไม่ถูกเซตหรือข้อมูลบิตที่ 9 ที่เก็บใน RB8 มีค่าเป็น “0” และสำหรับในโหมด 1 ถ้าบิต SM2 ถูกเซตนั้นคือบิต RI ไม่ถูกเซตหรือแสดงว่าบิตสิ้นสุดไม่ได้รับเข้า และสำหรับในโหมด 0 บิตนี้จะมีค่าเป็น “0”

* บิต REN ทำหน้าที่อีนابلการรับข้อมูลผ่านพอร์ตอนุกรม สามารถเซตเพื่ออีนابلและเคลียร์เพื่อคิเอสอเบิลการรับข้อมูลได้

* บิต TB8 เป็นสถานะของบิตที่ 9 สำหรับการส่งข้อมูลในโหมด 2 และ 3 สามารถเซตและเคลียร์ได้จากซอฟต์แวร์

* บิต TI ทำหน้าที่เป็นอินเตอร์รัปแฟลกในการส่งข้อมูล จะถูกเซตโดยฮาร์ดแวร์เมื่อสิ้นสุดการรับส่งข้อมูลบิตที่ 8 ในการทำงานโหมด 0 หรือถูกเซตที่จุดเริ่มต้นของบิตสิ้นสุดในการทำงานโหมดอื่นๆ และบิตนี้ต้องถูกเคลียร์จากซอฟต์แวร์

* บิต RI ทำหน้าที่เป็นอินเตอร์รัปแฟลกในการรับข้อมูล จะถูกเซตโดยฮาร์ดแวร์เมื่อสิ้นสุดการรับข้อมูลบิตที่ 8 ในการทำงานโหมด 0 หรือถูกเซตในช่วงระหว่างกลางของบิตสิ้นสุดในโหมดอื่นๆ (ยกเว้นบางกรณีสัมพันธ์กับบิต SM2) และบิตนี้ต้องถูกเคลียร์จากซอฟต์แวร์

จากภาพที่เรากำหนดกล่าวถึงนี้ก็คือโหมด 1 หรือเซตบิตโดยการโปรแกรมให้บิต SMO = 0 และ SM1 = 1 โหมดนี้นอกจากเป็นโหมดที่ง่ายที่สุดแล้ว อีกประการหนึ่งคือบิต SM2 อาจนำมาประยุกต์เพื่อตรวจสอบการติดต่อสื่อสารว่าสมบูรณ์หรือเกิดการผิดพลาดขึ้นหรือไม่ได้ด้วย

2.3.5 บัฟเฟอร์สำหรับการรับและส่งข้อมูล

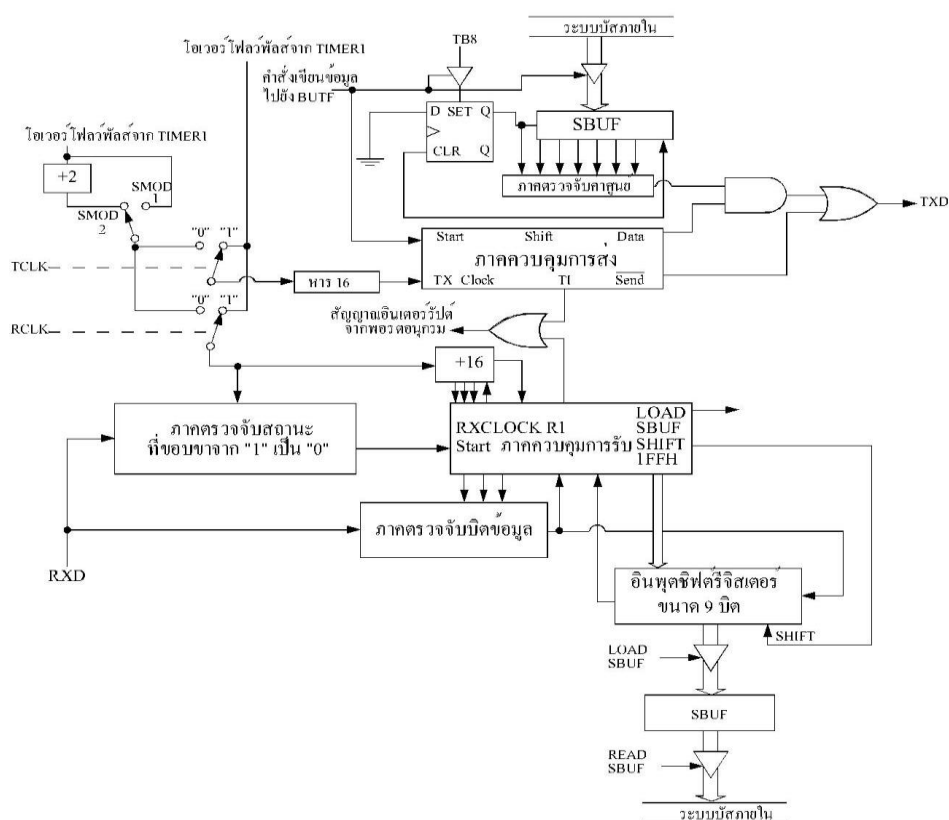
ภายในตัวไมโครคอนโทรลเลอร์บิตข้อมูลที่ต้องการทำการส่งและรับผ่านพอร์ตอนุกรมจะถูกส่งและรับลักษณะแบบชิฟต์ (Shift) เช่นเดียวกับการทำงานของชิฟต์รีจิสเตอร์ในกรณีที่ทำการส่งข้อมูล บิตข้อมูลทั้งหมดจะถูกโหลดลงสู่บัฟเฟอร์ ซึ่งทำงานเป็นชิฟต์รีจิสเตอร์แบบขนาน หลังจากนั้นจึงทำการชิฟต์ข้อมูลออกโดยเลื่อนไปที่ละบิตด้วยบิตเรต (Bit Rate) หรือบอดเรต (Baud Rate) ที่กำหนดไว้แล้วในโปรแกรม ในกรณีที่ทำการรับข้อมูลจะมีการทำงานลักษณะตรงข้ามกัน กล่าวคือ บิตข้อมูลที่รับได้จะถูกนำมารวบรวมใหม่ด้วยการชิฟต์บิตข้อมูลเข้าที่ละบิตไปยังชิฟต์รีจิสเตอร์เมื่อได้รับครบแล้ว จึงทำการอ่านข้อมูลนั้นไปใช้งานแบบขนานเข้าสู่ระบบ

ชิฟต์รีจิสเตอร์ที่กล่าวถึงนี้ทำหน้าที่เป็นเช่นเดียวกับบัฟเฟอร์ (Buffer) เพื่อพักข้อมูลในการส่งและรับข้อมูลซึ่งไมโครคอนโทรลเลอร์ก็ได้จัดเตรียมไว้แล้ว นั่นคือรีจิสเตอร์พิเศษ (SFR) ชื่อ SBUF อยู่

ที่ตำแหน่งแอดเดรส 099H มันจะทำหน้าที่เป็นได้ทั้งบัฟเฟอร์รับข้อมูลในกรณีระบบทำการรับข้อมูลจากภายนอก และเป็นบัฟเฟอร์เพื่อพักข้อมูลก่อนส่งออกสู่ระบบในกรณีระบบทำการส่งข้อมูลซึ่งผู้เขียนโปรแกรมต้องกำหนดการใช้คำสั่งอ่านหรือเขียนข้อมูลไปยังรีจิสเตอร์ SBUF ตามรูปแบบของการติดต่อที่ต้องการในขณะนั้นและตรงกับกรณีด้วย

2.3.6 โครงสร้างและการทำงาน

จากภาพที่ 2.9 โครงสร้างและการทำงานภายในของส่วนอินเตอร์เฟสที่บรรจุอยู่ในไมโครคอนโทรลเลอร์ MCS-51 บริเวณมุมซ้ายด้านบนของรูปแสดงในส่วนของภาคกำหนดบอดเรต (Baud Rate Generator) โดยจากรูปบอดเรตถูกกำหนดขึ้นจากพัลส์ที่แสดงการโอเวอร์โฟลว์ที่ได้จากไทม์เมอร์ 1 หรือไทม์เมอร์ 2 (มีเฉพาะ 8052 และ 8032) มาใช้เป็นสัญญาณนาฬิกาป้อนให้กับชิพรีจิสเตอร์โอเวอร์โฟลว์ที่ได้จากไทม์เมอร์ 1 สามารถกำหนดให้ทำการหารด้วย 2 ก่อนหรือไม่ได้ด้วยโดยการกำหนดสถานะที่บิต SMOD หรือ PCON.7 ซึ่งอยู่ในรีจิสเตอร์พิเศษ PCON ที่แอดเดรส 087H



ภาพที่ 2.9 โครงสร้างและการทำงานภายในของส่วนอินเตอร์เฟสแบบอนุกรมใน MCS-51

นอกจากนี้ในกรณีที่ใช้ไมโครคอนโทรลเลอร์เบอร์ 8052 และ 8032 ผู้เขียนโปรแกรมสามารถเลือกรับโอเวอร์โวลต์พัลส์จากไทม์เมอร์ 1 หรือไทม์เมอร์ 2 ก็ได้ โดยอาจแบ่งรับจากแหล่งใดก็ได้แยกอิสระกันด้วยการควบคุมจากสถานะของบิต RCLK และ TCLK หรือบิต T2CON.5 และ T2CON.4 ตามลำดับ ซึ่งอยู่ในรีจิสเตอร์ T2CON ที่แอดเดรส 0C8H โอเวอร์โวลต์พัลส์หรือสัญญาณนาฬิกาที่ได้ถูกหารด้วย 16 ก่อนป้อนให้กับภาคควบคุมการส่งข้อมูล (TXCONTROL) และภาคควบคุมการรับข้อมูล (RX-CONTROL)

จากรูปทางด้านบนเป็นส่วนของการส่งข้อมูลออกโดยมีขาเอาต์พุตคือขา TXD หรือขาที่ 11 ของไมโครคอนโทรลเลอร์ การทำงานของระบบเริ่มจากบิตข้อมูลที่จะทำการส่งข้อมูลถูกป้อนเข้าสู่รีจิสเตอร์ SBUS แบบขนาน หลังจากนั้นจึงเริ่มทำการส่งข้อมูลออกทีละบิต ภาคตรวจจับข้อมูลเป็นศูนย์ (Zerodetector) จะทำหน้าที่ตรวจจับข้อมูลที่เก็บอยู่ใน SBUS โดยถ้าข้อมูลใน SBUS เป็น “0” ทั้งหมดแสดงว่าส่งข้อมูลครบแล้ว มันจะส่งสัญญาณกลับไปยังภาคควบคุมการส่งข้อมูลและหลังจากนั้นระบบจะแสดงผลด้วยการเซตบิต TI (บิต SCON.1) ซึ่งสถานะของบิตนี้ สามารถนำไปใช้ในการอินเตอร์รัปต์จากพอร์ตอนุกรม (Serial Port Interrupt) ได้ทางด้านล่างของภาพที่ 2.9 แสดงส่วนของการรับข้อมูล การทำงานของระบบเริ่มจากบิตข้อมูลที่ถูกส่งมาแบบอนุกรมผ่านเข้ามาทางขา RXD หรือขาที่ 10 ของไมโครคอนโทรลเลอร์ บิตข้อมูลแต่ละบิตจะถูกรวบรวมใหม่โดยการชิฟต์ข้อมูลทีละบิตเข้าสู่อินพุตชิฟต์รีจิสเตอร์ การทำงานนี้จะเริ่มต้นก็ต่อเมื่อมีการเปลี่ยนแปลงสถานะจาก “1” เป็น “0” ที่ขา RXD ซึ่งนั่นก็คือบิตเริ่มต้นของขบวนบิตข้อมูลที่จะตามมาต่อไป โดยระบบจะตรวจจับการเปลี่ยนแปลงนี้ที่ภาคตรวจจับขอบขาลง (Transition Detector) และทำการแจ้งจังหวะกันกับบิตข้อมูลที่ได้รับเข้ามา

บิตข้อมูลจริงจะถูกรับเข้าสู่ระบบหลังจากที่ปรากฏบิตเริ่มต้นที่ขา RXD แล้ว โดยส่งไปยังชิฟต์รีจิสเตอร์ที่จัดเตรียมไว้แล้ว จนกระทั่งถึงบิตสุดท้ายนั่นคือบิตสิ้นสุดซึ่งถ้านับแล้วบิตนี้คือบิตที่ 9 ต่อท้ายจากบิตข้อมูลจำนวน 8 บิต (ชิฟต์รีจิสเตอร์รับข้อมูลมีขนาด 9 บิต) หลังจากนั้นบิตข้อมูลทั้งหมด (ขนาด 8 บิต) ถูกโหลดไปยังรีจิสเตอร์ SBUF และนั่นแสดงว่ากระบวนการรับข้อมูลสมบูรณ์แล้ว ระบบจะแสดงให้เห็นทราบด้วยการเซตบิต RI (บิต SCON.0) อย่างไรก็ตามผลของการเซตบิต RI ยังขึ้นกับเงื่อนไขอื่นด้วยซึ่งแสดงความหมายต่างกัน

เมื่อบิต RI ถูกเซตอยู่แล้วอาจหมายถึงข้อมูลที่รับมาเดิมก่อนหน้านี้ยังไม่ได้เฟตช์ (Fetch) ไปใช้งาน บิต RI จะถูกเซตในกรณีที่บิต SM2 เป็น “1” และไม่มีบิตสิ้นสุดปรากฏเป็นค่า “1” ในรูปแบบนี้จะมีใช้งานกัน เมื่อผู้เขียนโปรแกรมต้องการตรวจสอบความถูกต้องและป้องกันการเกิดข้อมูลผิดพลาด โดยพิจารณาจากเกิดการผิดปกติของบิตสิ้นสุดขึ้น ซึ่งผู้เขียนโปรแกรมต้องกำหนดบิต SM2 เป็น “1” ในรูปแบบนี้จะมีใช้งานกัน เมื่อผู้เขียนโปรแกรมต้องการตรวจสอบความถูกต้องและป้องกันการเกิด

