

บทที่ 10

กระบวนการอินเทอร์รัพท์

10.1 บทนำ

การอินเทอร์รัพท์คือขั้นตอนของการขัดจังหวะการทำงานของโปรแกรมหลักที่กำลังทำงานอยู่ เพื่อให้มาทำงานในส่วนของโปรแกรมที่ได้กำหนดไว้ในอินเทอร์รัพท์ วิธีการนี้จะช่วยให้ประหยัดเวลาในการทำงานของโปรแกรมและลดโอกาสความผิดพลาดในการตรวจสอบเงื่อนไขการทำงานของโปรแกรมหลัก เพราะไม่ต้องไปคอยตรวจสอบเงื่อนไขใดเงื่อนไขหนึ่งตลอดเวลาขณะที่โปรแกรมกำลังทำงานอยู่ โดยสามารถกำหนดหน้าที่การตรวจสอบเงื่อนไขนี้ให้กับการอินเทอร์รัพท์ทำหน้าที่แทน เนื้อหาในบทนี้ได้อธิบายการทำงานของกระบวนการอินเทอร์รัพท์สำหรับไมโครคอนโทรลเลอร์ MCS-51

10.2 พื้นฐานการทำงานของอินเทอร์รัพท์

การอินเทอร์รัพท์คือการขัดจังหวะการทำงานของซีพียู เพื่อให้ซีพียูหยุดการประมวลผลในปัจจุบัน เอาไว้ชั่วคราวก่อน จากนั้นกระโดดไปทำงานในฟังก์ชันหรือชุดคำสั่งของอินเทอร์รัพท์ ซึ่งฟังก์ชันหรือชุดคำสั่งที่ซีพียูกระโดดไปประมวลผลเมื่อเกิดการอินเทอร์รัพท์จะเรียกว่าโปรแกรมสำหรับบริการอินเทอร์รัพท์ (ISR: Interrupt service routine) เมื่อซีพียูประมวลผลชุดคำสั่งในโปรแกรมสำหรับบริการอินเทอร์รัพท์เสร็จสิ้นซีพียูจะกลับไปประมวลผลโปรแกรมที่ได้หยุดไว้ก่อนหน้าที่จะมีการอินเทอร์รัพท์ การเกิดการอินเทอร์รัพท์ของไมโครคอนโทรลเลอร์ MCS-51 โดยทั่วไปจะเกิดขึ้นได้จาก 6 แหล่งกำเนิดสัญญาณอินเทอร์รัพท์ แต่มีเวกเตอร์ (Vector) ของการอินเทอร์รัพท์ 5 เวกเตอร์ เนื่องจากการอินเทอร์รัพท์ที่เกิดจากพอร์ตอนุกรมทั้งการรับข้อมูลและการส่งข้อมูลจะใช้เวกเตอร์ร่วมกัน การอินเทอร์รัพท์เกิดขึ้นได้ 2 ลักษณะ คือการอินเทอร์รัพท์จากภายนอกและการอินเทอร์รัพท์จากภายใน ซึ่งจะได้อธิบายรายละเอียดของการอินเทอร์รัพท์ต่างๆ ในหัวข้อถัดไป

10.3 เรจิสเตอร์ที่เกี่ยวข้องกับกระบวนการอินเทอร์รัพท์

กระบวนการอินเทอร์รัพท์มีเรจิสเตอร์ที่ควบคุมการทำงานที่สำคัญ ซึ่งมีการทำงานแตกต่างกันออกไป ดังนี้

10.3.1 เรจิสเตอร์ IE

เรจิสเตอร์ IE (Interrupt enable register) เป็นเรจิสเตอร์ฟังก์ชันพิเศษมีขนาด 8 บิต อยู่ที่แอดเดรส 0A8H สามารถเข้าถึงในระดับบิตได้ ใช้ในการควบคุมการเปิด/ปิด หรือผลการตอบสนองของการอินเทอร์รัพท์ต่างๆ รายละเอียดของแต่ละบิตแสดงดังภาพที่ 10.1 ซึ่งในสถานะเริ่มต้นของไมโครคอนโทรลเลอร์หลังการรีเซต จะกำหนดให้ค่าของเรจิสเตอร์ IE มีค่า 0000H เป็นค่าเริ่มต้น

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
EA	-	ET2	ES	ET1	EX1	ET0	EX0

ภาพที่ 10.1 รายละเอียดบิตต่างๆ ของเรจิสเตอร์ IE

ที่มา: Architecture and Programming of 8051 MCUS, 2014

การกำหนดค่าในบิตต่างๆ ของเรจิสเตอร์ IE สามารถแสดงการทำงานของแต่ละบิตได้ ดังนี้

1. EA (Enable/Disable all interrupt) เป็นบิตที่ทำหน้าที่ตอบรับการอินเทอร์รัพท์ โดยที่

- 1) ลอจิก 0 ไม่มีการตอบรับอินเทอร์รัพท์ทั้งหมด
- 2) ลอจิก 1 ให้มีการตอบรับอินเทอร์รัพท์

2. ET2 (Timer 2 interrupt enable) ใช้ในการเปิด/ปิดการอินเทอร์รัพท์จาก ไทม์เมอร์ 2 ที่เกิดจากการเกิดโอเวอร์โฟลว์ของไทม์เมอร์ 2 มีการนับขึ้นจนเกิดการโอเวอร์โฟลว์ หรือมีการแคปเจอร์เกิดขึ้น บิตนี้สามารถเข้าถึงเพื่อเซตหรือเคลียร์ค่าได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น โดยที่

- 1) ลอจิก 0 เป็นการปิดการอินเทอร์รัพท์จากไทม์เมอร์ 2
- 2) ลอจิก 1 เป็นการเปิดการอินเทอร์รัพท์จากไทม์เมอร์ 2

3. ES (Serial port interrupt enable) ใช้ในการเปิด/ปิดการอินเทอร์รัพท์จาก พอร์ตอนุกรมในกระบวนการรับส่งข้อมูล บิตนี้สามารถเข้าถึงเพื่อเซตหรือเคลียร์ค่าได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น โดยที่

- 1) ลอจิก 0 เป็นการปิดการอินเทอร์รัพท์จากพอร์ตอนุกรม
- 2) ลอจิก 1 เป็นการเปิดการอินเทอร์รัพท์จากพอร์ตอนุกรม

4. ET1 (Timer 1 interrupt enable) ใช้ในการเปิด/ปิดการอินเทอร์รัพท์จาก ไทม์เมอร์ 1 ที่เกิดจากการเกิดโอเวอร์โฟลว์ของไทม์เมอร์ 1 บิตนี้สามารถเข้าถึงเพื่อเซตหรือเคลียร์ค่าได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น โดยที่

- 1) ลอจิก 0 เป็นการปิดการอินเทอร์รัพท์จากไทม์เมอร์ 1
 - 2) ลอจิก 1 เป็นการเปิดการอินเทอร์รัพท์จากไทม์เมอร์ 1
5. EX1 เป็นบิตที่ทำหน้าที่ควบคุมให้ทำการตอบรับการอินเทอร์รัพท์จากภายนอก INT1 โดยที่
- 1) ลอจิก 0 ไม่ตอบรับการอินเทอร์รัพท์ที่ INT1
 - 2) ลอจิก 1 ตอบรับการอินเทอร์รัพท์ที่ INT1
6. ET0 (Timer 0 interrupt enable) ใช้ในการเปิด/ปิดการอินเทอร์รัพท์จากไทม์เมอร์ 0 ที่เกิดจากการเกิดโอเวอร์โฟลว์ของไทม์เมอร์ 0 บิตนี้สามารถเข้าถึงเพื่อเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น โดยที่
- 1) ลอจิก 0 เป็นการปิดการอินเทอร์รัพท์จากไทม์เมอร์ 0
 - 2) ลอจิก 1 เป็นการเปิดการอินเทอร์รัพท์จากไทม์เมอร์ 0
7. EX0 เป็นบิตที่ทำหน้าที่ควบคุมให้ทำการตอบรับการอินเทอร์รัพท์จากภายนอก INTO โดยที่
- 1) ลอจิก 0 ไม่ตอบรับการอินเทอร์รัพท์ที่ INTO
 - 2) ลอจิก 1 ตอบรับการอินเทอร์รัพท์ที่ INTO

10.3.2 เรจิสเตอร์ IP

เรจิสเตอร์ IP (Interrupt priority register) เป็นเรจิสเตอร์ฟังก์ชันพิเศษ มีขนาด 8 บิต อยู่ที่แอดเดรส 0B8H สามารถเข้าถึงในระดับบิตได้ ใช้ในการควบคุมลำดับความสำคัญของการอินเทอร์รัพท์ต่างๆ เนื่องจากในบางครั้งการอินเทอร์รัพท์จะเกิดขึ้นพร้อมกันหรือซ้อนกัน ดังนั้นจึงต้องมีการจัดลำดับความสำคัญของการอินเทอร์รัพท์เพราะเมื่อมีการอินเทอร์รัพท์เกิดขึ้นซ้อนกันซีพียูจะไปประมวลผลในโปรแกรมบริการการอินเทอร์รัพท์ที่มีความสำคัญสูงกว่าก่อน หากต้องการให้การอินเทอร์รัพท์จากแหล่งกำเนิดใดสูงกว่าสามารถทำได้โดยการให้บิตประจำตำแหน่งของการอินเทอร์รัพท์นั้นเป็นลอจิก 1 รายละเอียดของแต่ละบิตแสดงดังภาพที่ 10.2

bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
-	-	PT2	PS	PT1	PX1	PT0	PX0

ภาพที่ 10.2 รายละเอียดบิตต่างๆ ของเรจิสเตอร์ IP

ที่มา: Architecture and Programming of 8051 MCUS, 2014

การกำหนดค่าในบิตต่างๆ ของเรจิสเตอร์ IP จะมีผลต่อการทำงานของลำดับความสำคัญของการอินเทอร์รัพท์ที่แตกต่างกันออกไป สามารถแสดงการทำงานของแต่ละบิตได้ดังนี้

1. PT2 (Timer 2 interrupt priority) ใช้ในการกำหนดลำดับความสำคัญของการอินเทอร์รัพท์ที่เกิดขึ้นจากไทม์เมอร์ 2 บิตนี้สามารถเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์
2. PS (Serial port interrupt priority) ใช้ในการกำหนดลำดับความสำคัญของการอินเทอร์รัพท์ที่เกิดจากพอร์ตอนุกรม ซึ่งสามารถเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น
3. PT1 (Timer 1 interrupt priority) ใช้ในการกำหนดลำดับความสำคัญของการอินเทอร์รัพท์ที่เกิดจากไทม์เมอร์ 1 ซึ่งสามารถเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น
4. PX1 (External interrupt1 priority) ใช้ในการกำหนดลำดับความสำคัญของการอินเทอร์รัพท์จากภายนอก 1 ซึ่งสามารถเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น
5. PT0 (Timer 0 interrupt priority) ใช้ในการกำหนดลำดับความสำคัญของการอินเทอร์รัพท์ที่เกิดขึ้นจากไทม์เมอร์ 0 ซึ่งสามารถเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์
6. PX0 (External interrupt1 priority) ใช้ในการกำหนดลำดับความสำคัญของการอินเทอร์รัพท์จากภายนอก 0 ซึ่งสามารถเซตหรือเคลียร์ได้โดยกระบวนการทางซอฟต์แวร์เท่านั้น

10.4 การอินเทอร์รัพท์จากภายนอก

การอินเทอร์รัพท์จากภายนอกจะเกิดขึ้นเมื่อได้รับจากสัญญาณอินเทอร์รัพท์ที่ส่งเข้ามาจากภายนอกผ่านทางขารับสัญญาณอินเทอร์รัพท์ โดยทั่วไปสามารถเกิดขึ้นได้ 2 แหล่งกำเนิด โดยมีรายละเอียดของการอินเทอร์รัพท์จากภายนอกดังนี้

10.4.1 การเกิดอินเทอร์รัพท์จากภายนอก 0

การเกิดอินเทอร์รัพท์จากภายนอก 0 จะเกิดขึ้นเมื่อมีสัญญาณเข้ามาที่ขา INTO การอินเทอร์รัพท์นี้จะเกิดขึ้นได้ 2 ลักษณะคือเมื่อขา INTO ได้รับสัญญาณลอจิก 0 และเมื่อขา INTO ได้รับสัญญาณขอบขาลง การอินเทอร์รัพท์จากภายนอก 0 มีเวกเตอร์ของการอินเทอร์รัพท์อยู่ที่แอดเดรส 0003H

10.4.2 การเกิดอินเทอร์รัพท์จากภายนอก 1

การเกิดอินเทอร์รัพท์จากภายนอก 1 จะเกิดขึ้นเมื่อมีสัญญาณเข้ามาที่ขา INT1 การอินเทอร์รัพท์นี้จะเกิดขึ้นได้ 2 ลักษณะคือเมื่อขา INT1 ได้รับสัญญาณลอจิก 0 และเมื่อขา INT1 ได้รับสัญญาณขอบขาลง การอินเทอร์รัพท์จากภายนอก 1 มีเวกเตอร์ของการอินเทอร์รัพท์อยู่ที่แอดเดรส 0013H

10.5 การอินเทอร์รัพท์จากภายใน

การอินเทอร์รัพท์จากภายในจะเกิดขึ้นจากสัญญาณอินเทอร์รัพท์ของวงจรมายในที่มีอยู่ในตัวไมโครคอนโทรลเลอร์ เช่น ไทม์เมอร์หรือพอร์ตอนุกรม เป็นต้น

10.5.1 การอินเทอร์รัพท์จากไทม์เมอร์ 0

ไทม์เมอร์ 0 สามารถสร้างสัญญาณอินเทอร์รัพท์ให้เกิดขึ้นได้เมื่อมีการนับขึ้นจนเกิดการโอเวอร์โฟลว์ เมื่อไทม์เมอร์ 0 นับขึ้นจนเกิดการโอเวอร์โฟลว์ จะมีผลทำให้บิต TF0 ของเรจิสเตอร์ TCON เกิดการเซตเป็นลอจิก 1 และในเวลาเดียวกันนี้เองหากมีการอนุญาตให้มีการอินเทอร์รัพท์ไว้ (EA = 1 และ ET0 = 1) ซีพียูจะกระโดดไปประมวลผลยังโปรแกรมบริการการอินเทอร์รัพท์ของไทม์เมอร์ 0 โดยการอินเทอร์รัพท์จากไทม์เมอร์ 0 มีแอดเดรสของการอินเทอร์รัพท์อยู่ที่แอดเดรส 000BH

10.5.2 การอินเทอร์รัพท์จากไทม์เมอร์ 1

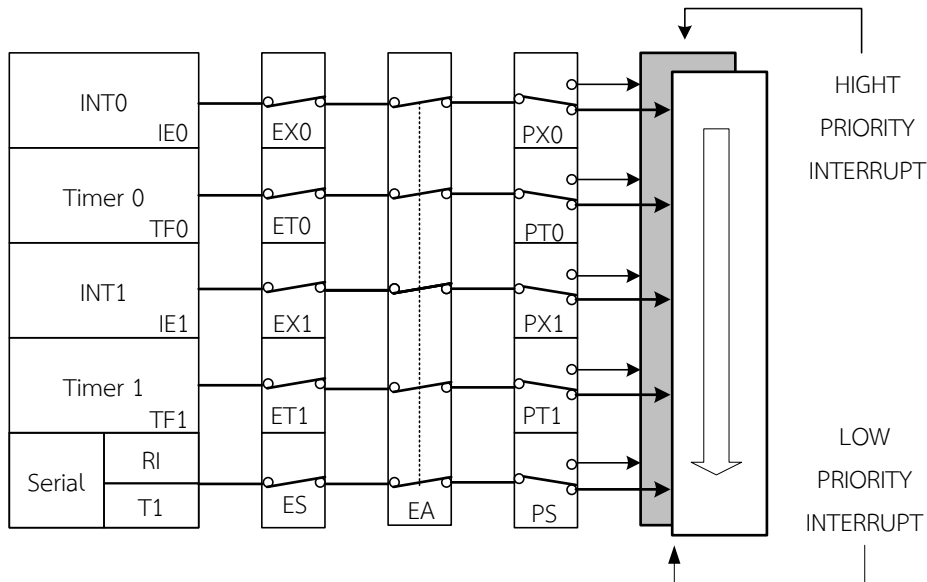
การอินเทอร์รัพท์จากไทม์เมอร์ 1 ไม่ได้มีความแตกต่างไปจากการอินเทอร์รัพท์จากไทม์เมอร์ 0 นั่นคือ ไทม์เมอร์ 1 สามารถสร้างสัญญาณอินเทอร์รัพท์ให้เกิดขึ้นได้เมื่อมีการนับขึ้นจนเกิดการโอเวอร์โฟลว์ ซึ่งจะมีผลทำให้บิต TF1 ของเรจิสเตอร์ TCON เกิดการเซตเป็นลอจิก 1 และในเวลาเดียวกันนี้เองหากมีการอนุญาตให้มีการอินเทอร์รัพท์ไว้ (EA = 1 และ ET1 = 1) ซีพียูจะกระโดดไปประมวลผลยังโปรแกรมบริการการอินเทอร์รัพท์ของไทม์เมอร์ 1 โดยมีแอดเดรสของการอินเทอร์รัพท์อยู่ที่แอดเดรส 001BH

10.6 ลำดับความสำคัญของการอินเทอร์รัพท์

กระบวนการอินเทอร์รัพท์ต่างๆ ของไมโครคอนโทรลเลอร์สามารถเกิดขึ้นได้พร้อมกัน แต่ซีพียูไม่สามารถประมวลผลโปรแกรมบริการการอินเทอร์รัพท์ต่างๆ ที่เกิดขึ้นพร้อมกันได้ในเวลาเดียวกัน ด้วยเหตุนี้เองการจัดลำดับจึงเข้ามามีส่วนเกี่ยวข้องในการเขียนโปรแกรมที่มีการใช้ความสามารถในการอินเทอร์รัพท์มากกว่า 1 อินเทอร์รัพท์ การจัดลำดับความสำคัญของการอินเทอร์รัพท์จะต้องกำหนดโดยผู้ออกแบบระบบเองว่าเมื่อมีสัญญาณอินเทอร์รัพท์เกิดขึ้นพร้อมกันจะให้ทำโปรแกรมบริการการอินเทอร์รัพท์ส่วนใดก่อนหรือหลัง โดยการกำหนดค่าลำดับความสำคัญในเรจิสเตอร์ IP ดังภาพที่ 10.3 นั่นคือหากต้องการใช้อินเทอร์รัพท์จากแหล่งกำเนิดใดสำคัญกว่าก็ให้เซตค่าเรจิสเตอร์ IP ในบิตนั้นให้เป็นลอจิก 1

สัญญาณต่าง ๆ ของการร้องขออินเทอร์รัพท์ทั้งหมดสามารถเซตหรือเคลียร์ได้ด้วยซอฟต์แวร์ เช่นเดียวกับฮาร์ดแวร์ ดังนั้นการร้องขออินเทอร์รัพท์จึงสามารถสร้างหรือยกเลิกได้ด้วยซอฟต์แวร์ อย่างไรก็ตามแหล่งกำเนิดสัญญาณอินเทอร์รัพท์ทั้งหมด สามารถควบคุมให้ทำการร้องขออินเทอร์รัพท์

หรือไม่ก็ได้ โดยการเซตหรือเคลียร์บิตต่างๆที่อยู่ใน เรจิสเตอร์ IE ซึ่งภายในเรจิสเตอร์ IE มีบิต EA ที่ทำหน้าที่ควบคุมให้ไมโครคอนโทรลเลอร์ตอบรับการร้องขออินเทอร์รัพท์หรือไม่ตอบรับทั้งหมด หากกำหนดให้บิต EA เป็นลอจิก 0 ไมโครคอนโทรลเลอร์จะไม่ตอบรับการร้องขออินเทอร์รัพท์ทั้งหมด



ภาพที่ 10.3 แผนภาพบล็อกแสดงการจัดลำดับความสำคัญของการอินเทอร์รัพท์

ที่มา: Architecture and Programming of 8051 MCUS, 2014

10.7 สรุป

ในบทนี้ผู้เรียนได้ทำความเข้าใจเกี่ยวกับวิธีการอินเทอร์รัพท์ โดยในขณะที่ไมโครคอนโทรลเลอร์กำลังทำงานในส่วนของโปรแกรมหลักอยู่นั้น ไมโครคอนโทรลเลอร์จะใช้วิธีการรับสัญญาณการร้องขอที่ส่งมาจากอุปกรณ์ภายนอกหรือสัญญาณจากภายในของตัวไมโครคอนโทรลเลอร์เอง มาทำการอินเทอร์รัพท์และเมื่อไมโครคอนโทรลเลอร์ได้รับสัญญาณจากการอินเทอร์รัพท์ก็จะทำการตรวจสอบในส่วนของ การติดต่อจากเหตุการณ์ในส่วนนั้นๆ และให้บริการการอินเทอร์รัพท์ โปรแกรมการบริการการอินเทอร์รัพท์ในตัวไมโครคอนโทรลเลอร์มีแหล่งสัญญาณที่เป็นารอินเทอร์รัพท์หลายแหล่งด้วยกันทั้งภายนอกและภายในของตัวไอซีไมโครคอนโทรลเลอร์ อย่างไรก็ตามการเขียนโปรแกรมในการควบคุมที่มีการประมวลผลกับสัญญาณที่มีช่วงเวลามากๆ ไม่จำเป็นต้องจัดความสำคัญของการอินเทอร์รัพท์ เนื่องจากความเร็วในการทำงานของไมโครคอนโทรลเลอร์มีมากพออยู่แล้ว แต่ทั้งนี้ต้องมั่นใจก่อนว่าสัญญาณที่ต้องการตรวจจับนั้น เกิดขึ้นนานมากกว่าเวลาที่ใช้ในการประมวลผลของโปรแกรมบริการการอินเทอร์รัพท์ที่มีความสำคัญทางฮาร์ดแวร์สูงกว่า ไม่เช่นนั้นแล้วอาจทำให้เกิดความผิดพลาดในการตรวจจับสัญญาณโดยเฉพาะอย่างยิ่งเมื่อนำสัญญาณตัวนี้ไปนับ เช่น ในเครื่องวัดความถี่ เป็นต้น

10.8 แบบฝึกหัดท้ายบท

แบบฝึกหัดมีทั้งหมด 3 ข้อ ให้นักศึกษาทำแบบฝึกหัดทุกข้อ

1. จงอธิบายความแตกต่างระหว่างการเขียนโปรแกรมแบบทั่วไปกับการเขียนโปรแกรมที่รองรับกระบวนการอินเทอร์เน็ต
2. จงอธิบายการอินเทอร์เน็ตจากสัญญาณภายนอก พร้อมทั้งยกตัวอย่างการประยุกต์ใช้งาน
3. จงอธิบายการอินเทอร์เน็ตจากภายใน พร้อมทั้งยกตัวอย่างการประยุกต์ใช้งาน