

หน่วยที่ 7 ไทเมอร์ เคาน์เตอร์

อดิศักดิ์ ชินะวงษ์

เอกสารประกอบการเรียนวิชาไมโครคอนโทรลเลอร์

เผยแพร่ที่ www.Adisak51.com

1. หน้าที่ของไทมเมอร์ เคา์เตอร์

ไทมเมอร์ เคา์เตอร์ (Timer/Counter) หมายถึงความสามารถในการนำไปใช้เป็นการนับค่า หรือเป็นการตั้งเวลา เพื่อกำหนดค่าที่ต้องการ ส่วนของไทมเมอร์ เคา์เตอร์อยู่ในรีจิสเตอร์ TMOD เป็นรีจิสเตอร์ใช้งานพิเศษ ทำหน้าที่ควบคุมโหมดการทำงาน ประกอบด้วยไทมเมอร์ เคา์เตอร์ จำนวน 2 ตัว คือไทมเมอร์0 และไทมเมอร์1 การใช้งานของไทมเมอร์ ทั้ง 2 ตัว สามารถแยกการทำงานออกจากกันได้อย่างอิสระโดยเก็บค่าที่กำหนดการนับ หรือตั้งเวลาไว้ในข้อมูลของรีจิสเตอร์ TLx และ THx ซึ่งเป็นรีจิสเตอร์ขนาด 8 บิต อยู่ในรีจิสเตอร์ใช้งานพิเศษ มีทั้งหมด 4 ตัว ประกอบด้วย TH0, TL0, TH1 และ TL1 สามารถ ใช้ร่วมกันเป็นรีจิสเตอร์ขนาด 16 บิตได้ โดยนำ TH0 รวมกับ TL0 และ TH1 รวมกับ TL1

การใช้งานสามารถกำหนดที่บิต C/T ของรีจิสเตอร์ TMOD โดยการเลือกฟังก์ชันให้กับเคา์เตอร์ หรือไทมเมอร์ ในกรณีที่เลือกให้เป็นไทมเมอร์ ใช้ความถี่ภายในของไอซี MCS-51 ได้มาจากสัญญาณนาฬิกาที่ป้อนให้กับไอซี MCS-51 โดยใช้การนับขึ้นจากค่าข้อมูลในรีจิสเตอร์ TLx และ รีจิสเตอร์ THx จนทำให้มีค่าเป็น “1” ทุกบิต ทั้ง 16 บิต (TL0 = FFH ,TH0= FFH) หลังจากนั้นให้นับต่ออีกหนึ่งค่าทำให้ข้อมูลรีจิสเตอร์ TLx และ THx เป็น ”0” ทุกบิต (TL0 = 00H ,TH0= 00H) ทำให้บิตแสดงสถานะ TFX (TF0 และ TF1 อยู่ในรีจิสเตอร์ TCON) ทำหน้าที่แสดงผลการเกิดค่านับเกิน (Overflow) ถูกเซตเป็น “1” คือการครบกำหนดค่าที่ตั้งไว้นั่นเอง กรณีเดียวกันการทำให้เป็นตัวกำหนด เคา์เตอร์ ต้องกำหนดสถานะที่บิต C/T ของรีจิสเตอร์ TMOD ให้อยู่ที่ฟังก์ชันของการนับ และนับค่าเมื่อมีสัญญาณ จากภายนอกป้อนเข้ามาที่ขา Tx (T0 และ T1) ของไอซี MCS-51 และต้องเป็นสัญญาณที่มีการเปลี่ยนแปลงสถานะจากลอจิก “1” เป็นลอจิก “0” จึงทำให้รีจิสเตอร์ THx และ รีจิสเตอร์ TLx นับค่าขึ้นทีละหนึ่งค่า ต่อจากค่าที่กำหนดไว้ก่อนหน้า โดยให้นับต่อจนเกิดค่านับเกิน เช่นเดียวกันกับการกำหนดค่าไทมเมอร์ หลังจากนั้นทำให้บิต TFX แสดงผลค่านับเกินถูกเซตให้เป็น “1” หมายถึงการนับสัญญาณจากภายนอก ครบจำนวนที่ได้กำหนดไว้แล้ว

2. รีจิสเตอร์ที่ใช้งานร่วมกับ ไทมเมอร์ เคา์เตอร์

2.1 รีจิสเตอร์ไทมเมอร์

เป็นรีจิสเตอร์ขนาด 8 บิต อยู่ในรีจิสเตอร์ใช้งานพิเศษมี 4 ตัวคือ รีจิสเตอร์ TL0 อยู่ที่ตำแหน่งแอดเดรส 8AH, รีจิสเตอร์ TH0 อยู่ที่ตำแหน่งแอดเดรส 8CH, รีจิสเตอร์ TL1อยู่ที่ตำแหน่งแอดเดรส 8BH และรีจิสเตอร์ TH1อยู่ที่ตำแหน่งแอดเดรส8DH ในการใช้งานรีจิสเตอร์ทั้ง 4 ตัว สามารถใช้งานเป็นคู่เพื่อให้ได้รีจิสเตอร์ที่มีขนาด 16 บิต คือ TL0 กับ TH0 และ TL1 กับ TH1 โดยข้อมูลในรีจิสเตอร์ของ TL0 และ TL1 เก็บค่าข้อมูลของ 8 บิตล่าง ส่วน TH0 และ TH1 เก็บค่าข้อมูล 8 บิตบน เมื่อนำมารวมกันสามารถนับค่าได้ถึง 65,536 ค่าหรือ 0000H-FFFFH การใช้งานกำหนดให้มีการนับค่าให้เกิน 65,536 เพื่อให้เกิดค่านับเกิน โดยที่บิต TF0 และ TF1 เซตบิตทำให้มีสถานะลอจิกเป็น “1” แสดงการนับเกินค่าสูงสุดแล้ว ดังนั้นวิธีกำหนดค่าข้อมูลของรีจิสเตอร์ไทมเมอร์ จึงต้องนำค่าสูงสุดที่นับได้ (FFFFH) ไปลบออกจาก

ค่าเวลาที่ต้องการ และค่าผลลัพธ์กำหนดให้เป็นข้อมูลของรีจิสเตอร์ TLx และ THx ซึ่งเป็นค่าเริ่มต้นการนับ

2.2 รีจิสเตอร์ TCON (Timer Counter Control Register)

ทำหน้าที่ควบคุมสัญญาณการอินเตอร์รัปต์ของไทมเมอร์ และทำหน้าที่ตรวจสอบค่าการนับของไทมเมอร์ เมื่อเกิดค่านับเกิน ซึ่งเซต และเคลียร์บิตด้วยวิธีทางซอฟต์แวร์ หรือใช้คำสั่งในการโอนย้ายข้อมูล รีจิสเตอร์ TCONเป็นส่วนของรีจิสเตอร์ใช้งานพิเศษ อยู่ในตำแหน่งแอดเดรสที่ 88H และมีขนาด 8 บิต สามารถเข้าถึงในระดับบิตได้ ในสภาวะเริ่มต้นของไอซี MCS-51หลังการรีเซต กำหนดให้ ค่าของรีจิสเตอร์ TCON มีค่า 00000000B เป็นค่าเริ่มต้น แสดงดังภาพที่ 7.1

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 7.1 รีจิสเตอร์ควบคุมสัญญาณการอินเตอร์รัปต์ของไทมเมอร์ (TCON)

TF1: TCON.7 (Timer1 Overflow Flag) ทำหน้าที่เป็นแฟล็ก แสดงสถานะลอจิกเป็น “1” เมื่อเกิดค่า นับเกินที่ไทมเมอร์1 และกระโดดไปทำโปรแกรมการตอบสนองการอินเตอร์รัปต์ตำแหน่งเริ่มต้นที่แอดเดรส 001BH ของหน่วยความจำโปรแกรม และบิตนี้ถูกเคลียร์ให้เป็น “0” อัตโนมัติโดยทางฮาร์ดแวร์

TR1: TCON.6 (Timer1 Run Control Bit) เป็นบิตควบคุมการทำงานของไทมเมอร์1 โดยถ้าเซตที่บิตนี้ให้มีสถานะลอจิกเป็น “1” ควบคุมให้ไทมเมอร์1 ให้เริ่มทำงาน และถ้าเคลียร์ เป็นสถานะลอจิก “0” ทำการควบคุมให้ไทมเมอร์ 1 หยุดทำงาน

TF0: TCON.5 (Timer0 Overflow Flag) ทำหน้าที่เป็นแฟล็ก แสดงสถานะลอจิกเป็น “1” เมื่อเกิดค่านับเกินที่ไทมเมอร์1 และกระโดดไปทำโปรแกรมการตอบสนองการอินเตอร์รัปต์ตำแหน่งเริ่มต้นที่แอดเดรส 001BH ของหน่วยความจำโปรแกรม และบิตนี้ถูกเคลียร์ให้เป็น “0” อัตโนมัติโดยทางฮาร์ดแวร์

TR0: TCON.4 (Timer0 Run Control Bit) เป็นบิตควบคุมการทำงานของไทมเมอร์ 0 โดยถ้าเซตที่บิตนี้ให้มีสถานะลอจิกเป็น “1” ควบคุมให้ไทมเมอร์ 0 ให้เริ่มทำงาน และถ้าเคลียร์เป็นสถานะลอจิก“0” ทำการควบคุมให้ไทมเมอร์ 0 หยุดทำงาน

IE1: TCON.3 (External Interrupt1 Edge Flag) เป็นแฟล็กร้องขอการอินเตอร์รัปต์ภายนอกของสัญญาณ $\overline{INT1}$ บิตนี้ถูกเซตด้วยฮาร์ดแวร์ เมื่อมีสัญญาณอินเตอร์รัปต์ที่ขา $\overline{INT1}$ และถูกเคลียร์ อัตโนมัติเมื่อไอซี MCS-51กระโดดไปทำโปรแกรมตอบสนองการอินเตอร์รัปต์ของสัญญาณ $\overline{INT1}$ ซึ่งอยู่ในตำแหน่งเริ่มต้นที่แอดเดรส 0013H ของหน่วยความจำโปรแกรม

IT1: TCON.2 (Interrupt1 Type Control Bit) เป็นบิตควบคุมการเลือกรูปแบบการแอกทีฟของสัญญาณอินเตอร์รัปต์ จากภายนอกที่ขา $\overline{INT1}$ สามารถเซตหรือเคลียร์ด้วยซอฟต์แวร์ ถ้ามีสถานะเป็น “1”

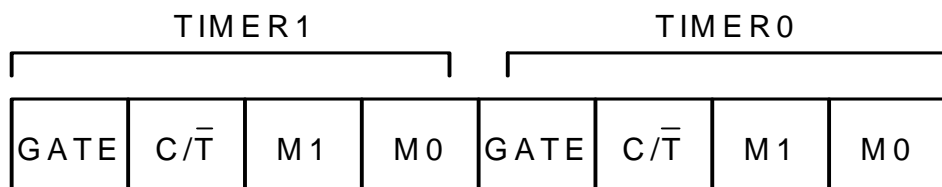
เป็นการเลือกใช้สัญญาณอินเทอร์รัปต์ $\overline{INT1}$ แอคทีฟ ที่การเปลี่ยนจาก “1” ไป “0” (ขอบขาลง) ถ้ามีสถานะเป็น “0” เป็นการเลือกใช้สัญญาณอินเทอร์รัปต์ $\overline{INT1}$ แอคทีฟที่ลอจิก “0” (Level)

IE0: TCON.1 (External Interrupt0 Edge Flag) เป็นแฟลกร็องขออินเทอร์รัปต์จากสัญญาณ $\overline{INT0}$ บิตนี้ ถูกเซตด้วยฮาร์ดแวร์เมื่อมีสัญญาณอินเทอร์รัปต์เข้ามาที่ขา $\overline{INT0}$ และถูกเคลียร์อัตโนมัติเมื่อไอซี MCS-51 กระโดดไปทำโปรแกรมตอบสนองการอินเทอร์รัปต์ของสัญญาณ $\overline{INT0}$ ในตำแหน่งแอดเดรส 0003H ของหน่วยความจำโปรแกรม

IT0: TCON.0 (Interrupt0 Type Control Bit) เป็นบิตควบคุมการเลือกรูปแบบการแอคทีฟของสัญญาณอินเทอร์รัปต์จากภายนอกที่ขา $\overline{INT0}$ สามารถเซต หรือเคลียร์ด้วยซอฟต์แวร์ ถ้ามีสถานะลอจิกเป็น “1” เป็นการเลือกใช้สัญญาณอินเทอร์รัปต์ $\overline{INT0}$ แอคทีฟที่การเปลี่ยนจาก “1” ไป “0” (ขอบขาลง) ถ้ามีสถานะลอจิกเป็น “0” เป็นการเลือกใช้สัญญาณอินเทอร์รัปต์ $\overline{INT0}$ แอคทีฟที่ลอจิก “0” (Level)

2.3 รีจิสเตอร์ TMOD (Timer/Counter Mode Control Register)

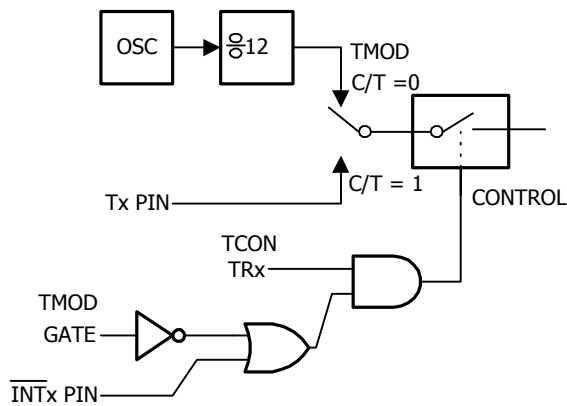
เป็นรีจิสเตอร์ควบคุมโหมดการทำงาน อยู่ในส่วนของรีจิสเตอร์ใช้งานพิเศษในตำแหน่งแอดเดรส 89H เป็นรีจิสเตอร์ขนาด 8 บิต ไม่สามารถอ้างถึงในระดับบิตได้ ดังนั้นจึงต้องใช้วิธีการโอนย้ายข้อมูลขนาด 8 บิต โดยตรง ในสถานะเริ่มต้นของไอซี MCS-51 หลังการรีเซต กำหนดให้ค่าของรีจิสเตอร์ TMOD มีค่า 00000000B เป็นค่าเริ่มต้น บิตต่างๆ แสดงดังภาพที่ 7.2 ใช้งานในการควบคุมโหมดการทำงาน และสัญญาณการควบคุมการทำงานของ ไทเมอร์ 0 และ ไทเมอร์ 1



ภาพที่ 7.2 แสดงรีจิสเตอร์ควบคุมโหมดการทำงาน (TMOD)

ความหมายแต่ละบิตของ ไทเมอร์ 0 และ ไทเมอร์ 1 เหมือนกัน แต่การใช้งานสามารถแยกออกจากกันได้อย่างอิสระ

GATE: (Gate Control When Set) เป็นบิตเลือกลักษณะควบคุมการทำงานในส่วนของ ไทเมอร์ 0 และ ไทเมอร์ 1 โดยการกำหนดสถานะของบิตถ้า GATE = “0” ให้ ไทเมอร์ 0 (ไทเมอร์ 1) ทำงาน โดยถูกควบคุมจากบิต TR0 (TR1) ในรีจิสเตอร์ TCON เป็นการควบคุมทางซอฟต์แวร์ (Software) ถ้ากำหนดให้ GATE = “1” ให้ ไทเมอร์ 0 (ไทเมอร์ 1) ทำงานโดยถูกควบคุมจากภายนอกที่ขาอินเทอร์รัปต์ $\overline{INT0}$ ($\overline{INT1}$) ของไอซีเป็นการควบคุมทางฮาร์ดแวร์ (Hardware)



ภาพที่ 7.3 แสดงวงจรการทำงานของไทมเมอร์ในส่วนองขา GATE \overline{INT} และ T1

จากภาพที่ 7.3 ถ้าให้สถานะลอจิกขา GATE เป็น “1” เมื่อผ่านอินเวอร์เตอร์ ทำให้เอาต์พุตของอินเวอร์เตอร์เป็นสถานะลอจิก “0” และถ้าให้ที่อินพุตของขาอินเตอร์รัปต์ $\overline{INT0}$ ($\overline{INT1}$) เป็นลอจิก “1” เอาต์พุตของออร์เกทเป็นสถานะลอจิก “1” ด้วย หากกำหนดให้ขา TR1 (TR0) เป็นสถานะลอจิก “1” ส่งผลให้เอาต์พุตของแอนด์เกทเป็นสถานะลอจิก “1” ตามไปด้วย ทำให้สวิทช์ในส่วนของ CONTROL ปิดวงจร ดังนั้นถ้าหากสถานะลอจิกที่ขา $\overline{INT0}$ เป็นสถานะลอจิก “0” เอาต์พุตของออร์เกทเป็นลอจิก “0” ส่งผลให้สวิทช์ Control เปิดวงจรออก และต้องกำหนดให้สถานะลอจิกของบิต TR1 (TR0) มีสถานะลอจิกเป็น “1” ด้วย เพื่อเป็นการเริ่มต้น และหยุดการนับโดยการกำหนดทางซอฟต์แวร์

C/T: (Timer or Counter Selector) เป็นการกำหนดฟังก์ชันการทำงานของไทมเมอร์ โดยใช้สถานะลอจิกของบิตนี้เลือกให้เป็นเคาน์เตอร์ หรือไทมเมอร์ หากกำหนดให้ลอจิกของบิตนี้เป็น “0” เป็นการเลือกให้เป็นไทมเมอร์ โดยใช้สัญญาณนาฬิกาจากภายในตัวไอซี MCS-51 เอง ถ้าลอจิกของบิตนี้เป็น “1” กำหนดเป็นเคาน์เตอร์รับสัญญาณอินพุตภายนอกที่ขา T0 และ T1

M1, M0: (Mode Selector Bit) เป็นการกำหนดโหมดทำงานของ เคาน์เตอร์และไทมเมอร์ สามารถแบ่งออกได้เป็น 4 โหมดคือ โหมด 0, โหมด 1, โหมด 2 และโหมด 3 ดังตารางที่ 7.1

ตารางที่ 7.1 แสดงการกำหนดค่าในบิต M0 และ M1 เพื่อเลือกโหมดการทำงาน

M1	M0	โหมด	การทำงาน
0	0	0	13 บิต ไทมเมอร์ เคาน์เตอร์
0	1	1	16 บิต ไทมเมอร์ เคาน์เตอร์
1	0	2	8 บิตแบบโหลดซ้ำ (Auto-reload Timer Counter)
1	1	3	8 บิต ไทมเมอร์ เคาน์เตอร์ โดยใช้ TLO

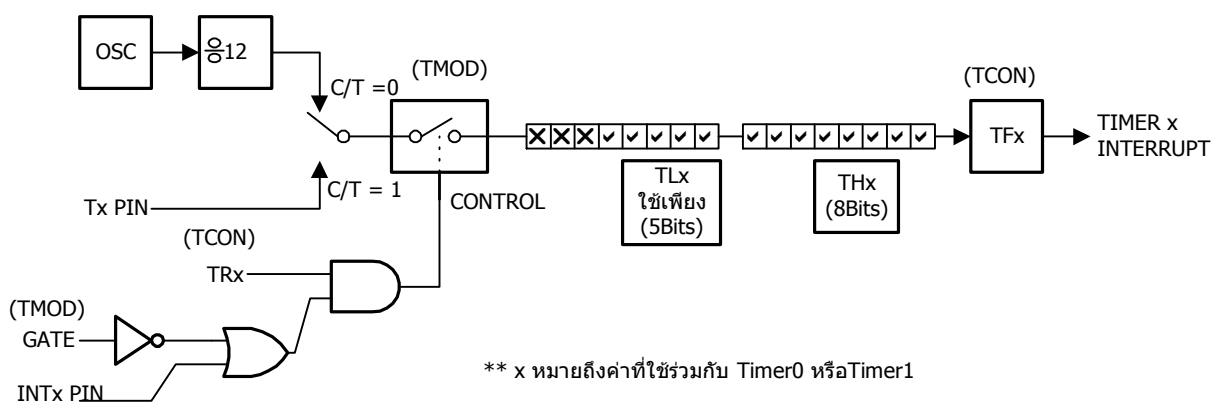
3. โหมดการทำงานของไทเมอร์ เคา์เตอร์

การกำหนดใช้งานไทเมอร์ 0 และ ไทเมอร์ 1 กระทำได้โดยกำหนดค่าที่ต้องการไว้ในรีจิสเตอร์ใช้งานเฉพาะ TMOD ดังแสดงในตารางที่ 7.1 ในแต่ละตารางให้ไทเมอร์ 0 หรือ ไทเมอร์ 1 ถูกใช้งานเพียงตัวเดียวเท่านั้น หากต้องการใช้งานไทเมอร์ 0 และไทเมอร์ 1 ให้ทำงานเป็นไทเมอร์ พร้อมกัน หรืออาจทำต่างหน้าที่ เช่น ให้ไทเมอร์ 0 เป็นเคา์เตอร์ ส่วนไทเมอร์ 1 ให้เป็นไทเมอร์ นำค่าในตารางของไทเมอร์ 0 มาทำการ OR กับค่าในตารางของไทเมอร์ 1 หลังจากนั้นนำมากำหนดค่าให้กับรีจิสเตอร์ TMOD

ตัวอย่างเช่นให้ไทเมอร์ 0 เป็นไทเมอร์ในโหมด 2 และควบคุมด้วยขา INT0 (External Control) ซึ่งในตารางได้ค่าข้อมูลเป็น 0AH ส่วนในไทเมอร์ 1 กำหนดให้เป็นเคา์เตอร์ในโหมด 2 และควบคุมด้วยซอฟต์แวร์ภายใน (Internal Control) ในตารางได้ค่าข้อมูลเป็น 60H เมื่อนำมารวมกันได้ค่าเท่ากับ 0A60H ซึ่งเป็นการกำหนดค่าข้อมูลในรีจิสเตอร์ TMOD และการใช้ไทเมอร์ หรือเคา์เตอร์ โดยควบคุมจากโปรแกรม ต้องทำการเซตที่บิต TRx ในรีจิสเตอร์ใช้งานเฉพาะ TCON เพื่อให้ไทเมอร์เริ่มทำงานนั่นเอง

3.1 การทำงานของ ไทเมอร์ เคา์เตอร์ ในโหมด 0

การทำงานของไทเมอร์ เคา์เตอร์ในโหมด 0 ค่าของการนับใช้งานเพียง 13 บิต โดยใช้ข้อมูลในรีจิสเตอร์ TLx เพียง 5 บิต และข้อมูลในรีจิสเตอร์ THx อีก 8 บิต รวมเป็น 13 บิต การนับอยู่ในช่วง 0000H - 1FFFFH (TLx = 00 - 1FH, THx = 00 - FFH) หรือ 0 – 8,191 วิธีนับให้เกินค่า 8,192 ทำให้บิตแสดงสถานะเกิน TF0,TF1 เกิดการเซตบิต ไอซี MCS-51 ทราบว่าเกิดการนับครบค่าที่กำหนดแล้ว ในวิธีการนับเริ่มที่รีจิสเตอร์ TLx โดยทำในส่วนของ 5 บิตล่าง (LSB) ให้ครบ 31 ครั้ง (2^5 หรือ XXX00000B - XXX11111B) หลังจากนั้นเมื่อทำการนับครั้งที่ 32 เพิ่มค่าที่ THx ขึ้นอีกหนึ่งค่า ดังนั้นอัตราส่วนระหว่าง THx และ TLx จึงเป็น 1:32

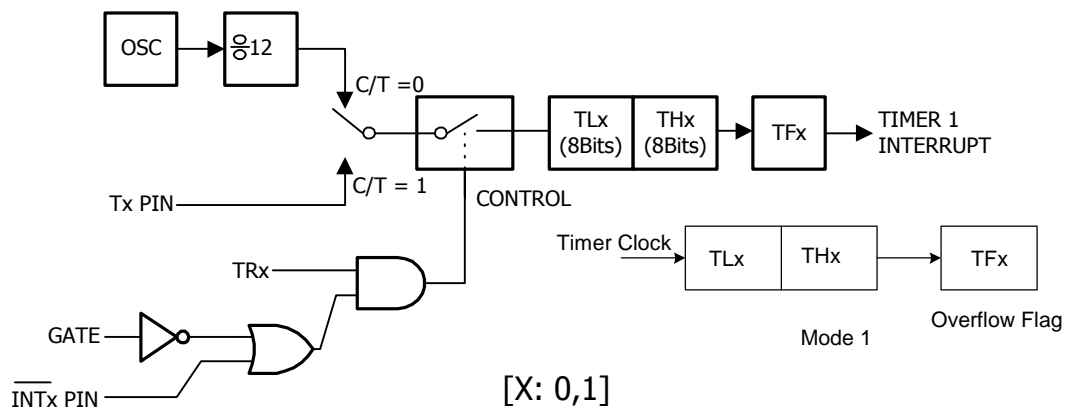


ภาพที่ 7.4 แสดงการทำงานของ ไทเมอร์ เคา์เตอร์ ในโหมดที่ 0

จากภาพที่ 7.4 การกำหนดสถานะของบิต GATE C/T และ TRx (TR0 หรือ TR1) โดยกำหนดจากซอฟต์แวร์ เป็นการเลือกใช้ควบคุมไทมเมอร์ หรือการทำงานในส่วนของฮาร์ดแวร์โดยกำหนดสถานะลอจิกที่ขา INTO หรือ INT1 เป็นการควบคุมการปิด – เปิดสวิตช์ ที่รับสัญญาณอินพุต เพื่อเข้าสู่การนับที่รีจิสเตอร์ TLx สัญญาณที่รับเข้ามามี 2 ทางโดยเลือกได้จากบิต C/T ทางแรกรับมาจากสัญญาณความถี่ภายในของไอซี MCS-51 ทารด้วย 12 ส่วนอีกทางรับสัญญาณมาจากภายนอกที่ขา T0 หรือ T1 ของไอซี MCS-51 โดยใช้บิต TRx ควบคุมการเริ่มต้น และหยุดการนับโดยคำสั่งทางซอฟต์แวร์

3.2 การทำงานของ ไทมเมอร์ เคาท์เตอร์ ในโหมด 1

การทำงานของโหมด1 มีลักษณะการทำงานคล้ายกับโหมด 0 เพียงแต่ค่าของการนับในโหมด 1 มีได้ถึง 16 บิต โดยนำเอาข้อมูลในรีจิสเตอร์ TL0 กับ TH0 หรือรีจิสเตอร์ TL1 กับ TH1 ขนาด 8 บิตมารวมกัน เพื่อให้ได้รีจิสเตอร์ขนาด 16 บิต โดยค่าของการนับอยู่ในช่วง 0000H - FFFFH หรือ 65,536 ค่าวิธีการใช้งานให้นับไปจนครบรอบโดยให้ค่าของการนับเปลี่ยนจาก FFFFH เป็น 0000H ทำให้บิต TFX ทำการเซตบิตเป็นลอจิก “1” ไอซี MCS-51 รู้ว่ามีการนับครบกำหนดแล้ว แสดงการทำงานของไทมเมอร์ เคาท์เตอร์ ในโหมด 1 (16 บิต) แสดงดังภาพที่ 7.5

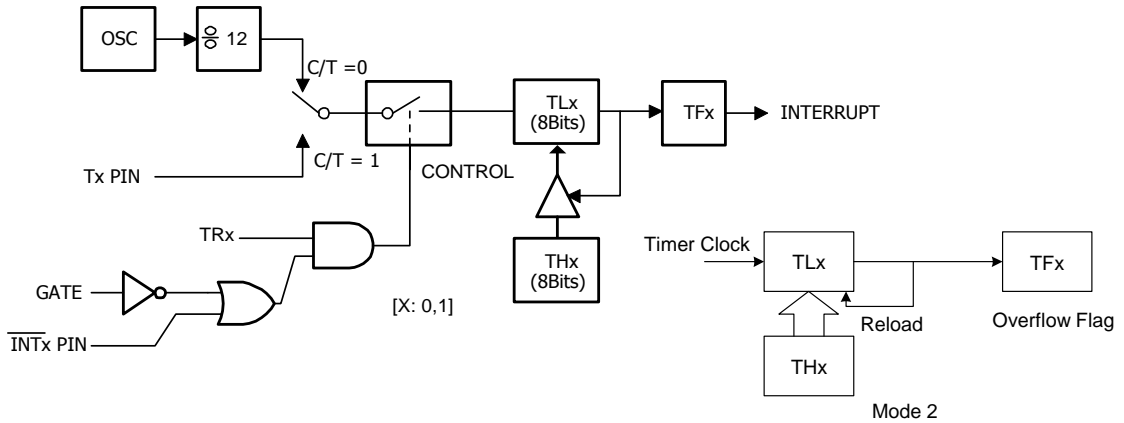


ภาพที่ 7.5 แสดงการทำงานของไทมเมอร์ เคาท์เตอร์ในโหมด 1 (16 บิต)

3.3 การทำงานของ ไทมเมอร์ เคาท์เตอร์ ในโหมด 2

โหมด 2 ทำงานเป็นแบบโหลดค่าข้อมูลใหม่ในการเริ่มนับ หรือโหมดการตั้งค่าใหม่แบบอัตโนมัติ (Auto Reloads Mode) การทำงานของไทมเมอร์ เคาท์เตอร์ในโหมด 2 แสดงดังภาพที่ 7.6 ในโหมดนี้แยกรีจิสเตอร์ไทมเมอร์ ออกเป็น 2 ตัว มีขนาดตัวละ 8 บิต วิธีการเก็บค่าเริ่มต้นการนับกำหนดอยู่ในข้อมูลของรีจิสเตอร์ THx โดยค่าการนับสูงสุด 256 ครั้ง (THx = 00) ขณะที่ไทมเมอร์ ทำงาน รีจิสเตอร์ TLx ซึ่งมีขนาด 8 บิต ทำหน้าที่เป็นคา운เตอร์ โดยนำค่าเริ่มต้นที่ต้องการนับมาจากรีจิสเตอร์ THx หลังจากนั้นค่าข้อมูลที่อยู่ในรีจิสเตอร์ TLx ถูกเพิ่มขึ้นทุกแมกซ์ซินไซเคิล จนมีค่าข้อมูลถึง FF เมื่อนับต่ออีก 1 แมกซ์ซิน

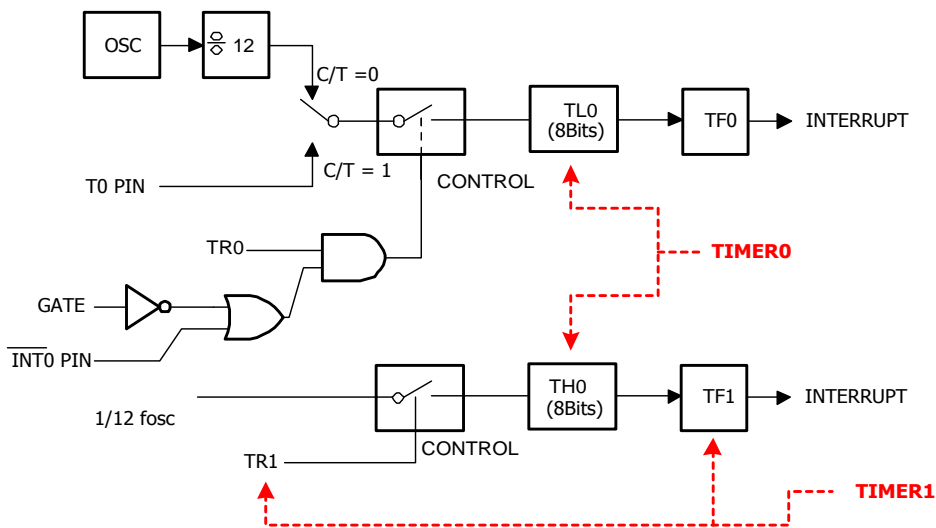
ไซเคิล วนกลับไป 00 อีกครั้ง ซึ่งคือค่านับเกิน ทำให้แฟล็ก TF0 (หรือTF1) ถูกเซตเป็นลอจิก “1” หลังจากนั้นค่าข้อมูลที่ถูกกำหนดไว้ในรีจิสเตอร์ THx นำไปเป็นข้อมูลในรีจิสเตอร์ TLx ใหม่อีกครั้ง เพื่อเริ่มต้นการนับในรอบต่อไปอีกครั้ง การทำงานจะเป็นเช่นนี้ตลอดไป (Auto Reload) ส่วนของแฟล็ก TFx ถูกเคลียร์โดยอัตโนมัติ เมื่อไอซี MCS-51 กระโดดไปทำโปรแกรมตอบสนองอินเทอร์รัปต์ของไทเมอร์ ในแต่ละรอบ



ภาพที่ 7.6 แสดงการทำงานของ ไทเมอร์ เคาท์เตอร์ ในโหมด 2 (8 bit Auto reload)

3.4 การทำงานของไทเมอร์ เคาท์เตอร์ ในโหมด 3

โหมด 3 แบ่งการใช้งานของไทเมอร์ 0 และไทเมอร์ 1 ออกจากกันโดยไทเมอร์ 0 แยกรีจิสเตอร์ TH0 และ TLO ทำให้เกิดเป็นเคาน์เตอร์ 2 ตัวทำงานอิสระต่อกัน และใช้งานได้เพียง 8 บิตเท่านั้นที่รีจิสเตอร์ TLO สามารถเลือกการทำงานได้ทั้งโหมดไทเมอร์ และโหมดเคาน์เตอร์ แสดงดังภาพที่ 7.7



ภาพที่ 7.7 แสดงการทำงานของไทเมอร์ 0 ในโหมด 3

เคาน์เตอร์ ที่รีจิสเตอร์ TL0 ใช้บิตควบคุมการทำงานของไทมเมอร์ 0 โดยกำหนดที่บิต C/T, Gate, TR0 INTO และแฟล็ก TF0 เป็นตัวควบคุมการทำงานขณะเกิดค่านับเกิน การทำงานของไทมเมอร์ 1 ไม่มีโหมด 3 แต่ใช้งานบิต TF1 และบิต TR1 ควบคุมการทำงานร่วมกับรีจิสเตอร์ TH0 (ไทมเมอร์ 0) แทนแสดงดังภาพที่ 7.7 กำหนดให้เป็นส่วนของไทมเมอร์ เท่านั้น (นับค่าของ แมชชีน ไซเคิลจากความถี่ภายในตัวไอซี) โดยกำหนดให้บิต TR1 มีค่าเป็นลอจิก “1” เคาน์เตอร์ TH0 เริ่มทำการนับ และเมื่อค่าข้อมูลของ TH0 เกิดค่านับเกิน (ค่าใน TH0 กลับไปเป็นค่า 00H อีกครั้ง) ทำให้บิต TF1 เซตค่าเป็นสถานะลอจิก “1” ซึ่งเป็นการร้องขออินเตอร์รัปต์ต่อไอซี MCS-51

ดังนั้นการใช้งานเคาน์เตอร์ ไทมเมอร์ ในโหมด 3 จึงเหมือนกับไอซีไมโครคอนโทรลเลอร์ 8051 มีเคาน์เตอร์ไทมเมอร์ อยู่ถึง 3 ชุด และถ้าหากเป็น ไอซีไมโครคอนโทรลเลอร์ 8052 เหมือนว่ามีถึง 4 ชุด

4. การกำหนดค่าการนับให้กับรีจิสเตอร์ TLx และ THx

ค่าข้อมูลที่อยู่ในรีจิสเตอร์ THx, TLx (THx = TH0, TH1 และ TLx = TL0, TL1) เพิ่มขึ้นทีละหนึ่งค่าจนเป็น “1” ทุกบิต (THx = FFH, TLx = FFH) ถ้ามีการนับอีกครั้งเกิดการนับเกินหมายถึงครบกำหนดของเวลาที่ตั้งไว้เช่น ต้องการให้ไทมเมอร์ นับจำนวน 200 ครั้ง ต้องกำหนดให้ค่าที่นับได้สูงสุดคือ 65,536 โดยนำไปลบออกจากค่าที่ต้องการนับคือ $65,536 - 200 = 65,336$ ถ้าแปลงให้เป็นเลขฐาน 16 มีค่าเท่ากับ FF38H ดังนั้นต้องกำหนดค่าข้อมูลในรีจิสเตอร์ THx ให้มีค่าเท่ากับ FFH และกำหนดค่าข้อมูลในรีจิสเตอร์ TLx ให้มีค่าเท่ากับ 38H โดยเริ่มต้นนับที่ค่า FF38H เพิ่มไปที่ละหนึ่งค่า จนครบ 200 ครั้งจึงเกิดค่านับเกิน

การกำหนดค่าข้อมูลให้กับรีจิสเตอร์ THx, TLx อีกวิธีหนึ่งโดยนำเอาจำนวนค่าที่ต้องการนับมาหาค่า 2'S Complement โดยวิธีการดังนี้

ขั้นตอนที่ 1 ให้แปลงค่า 200 เป็นเลขฐาน 16 ได้ค่าเท่ากับ	00C8H
ขั้นตอนที่ 2 แปลงเป็นเลขฐาน 2	= 0000 0000 1100 1000 B
ขั้นตอนที่ 3 ทำให้เป็น 1'S Complement	= 1111 1111 0011 0111B
(กลับข้อมูลจากขั้นตอนที่ 2 จาก “1” เป็น “0” และ “0” เป็น “1”)	
ขั้นตอนที่ 4 ทำเป็น 2'S Complement	= 1111 1111 0011 1000B
(บวกค่าข้อมูลในขั้นตอนที่ 3 อีก 1 ค่า)	
ขั้นตอนที่ 5 แปลงเป็นเลขฐาน 16	= FF38H

ผลลัพธ์ที่ได้คือค่าข้อมูลในรีจิสเตอร์ THx = FFH และค่าข้อมูลในรีจิสเตอร์ TLx = 38H

5. การหาค่าแมชชีน ไซเคิลของ ไทมเมอร์ เคาน์เตอร์ ในโหมด 1

ในโหมด 1 นับค่าได้สูงสุด 65,536 แมชชีน ไซเคิล ดังนั้นการเริ่มต้นค่าการนับต้องกำหนดค่า TH0 และ TL0 ค่าที่นับได้สูงสุด (0000H - FFFFH) การหาจำนวนของแมชชีน ไซเคิลมีสูตรคำนวณดังนี้

จำนวนแมชชีนไซเคิล = (2^{complement} ของข้อมูลในรีจิสเตอร์ TH0 -1) 256¹ + (2^{complement} ของข้อมูลในรีจิสเตอร์ TL0) 256⁰

การกำหนดค่าในรีจิสเตอร์ TH0 และ TL0 โดยใช้ไทมเมอร์ โหมด 1 ให้ได้ค่าเวลามากที่สุดโดยเริ่มนับจาก 0000H จนถึง FFFFH ดังนั้นจึงกำหนดให้ TH0 = 00H และ TL0 = 00H

แทนค่าในสูตร

$$\begin{aligned} \text{จำนวนแมชชีนไซเคิล} &= [(+256)-1] 256^1 + [(+256)] 256^0 \\ &= [255] 256^1 + [256] 256^0 \\ &= [255] 256 + [256] 1 \\ &= 65,280+256 \end{aligned}$$

$$\text{จำนวนแมชชีนไซเคิล} = 65,536$$

ถ้าใช้ X-TAL 11.0592 MHz ได้เวลาเท่ากับ $65,536 \times 1.085 = 71,106.56 \mu\text{S}$ หรือประมาณ 71 msec

ถ้าใช้ X-TAL 12 MHz ได้เวลาเท่ากับ $65,536 \times 1 = 65,536 \mu\text{S}$ หรือประมาณ 65.5 msec

การกำหนดค่าในรีจิสเตอร์ TH0 และ TL0 โดยใช้ไทมเมอร์ โหมด 1 ให้ได้ค่าเวลาน้อยที่สุด กำหนดให้ TH0 = FFH และ TL0 = FFH

แทนค่าในสูตร

$$\begin{aligned} \text{จำนวนแมชชีนไซเคิล} &= [(+1)-1] 256^1 + [(+1)] 256^0 \\ &= [0] 256^1 + [1] 256^0 \\ &= [0] 256 + [1] 1 \\ &= 0+1 \end{aligned}$$

$$\text{จำนวนแมชชีนไซเคิล} = 1$$

ถ้าใช้ X-TAL 11.0592 MHz ได้เวลาเท่ากับ $1 \times 1.085 = 1.085 \mu\text{S}$

ถ้าใช้ X-TAL 12 MHz ได้เวลาเท่ากับ $1 \times 1 = 1 \mu\text{S}$

6. ไทมเมอร์ เคนต์เตอร์ 2 ของไอซี MCS -51

ไอซี MCS-51 เบอร์ 8052, AT89C52 และอนุกรม AT89Sxx มีไทมเมอร์เคนต์เตอร์ขนาด 16 บิตเพิ่มอีก 1 ตัวคือไทมเมอร์ เคนต์เตอร์ 2 (Timer/Counter 2) โดยสามารถใช้งานได้เหมือนกับไทมเมอร์เคนต์เตอร์ 0 และไทมเมอร์ เคนต์เตอร์ 1 ผู้ใช้สามารถเลือกการทำงานได้ด้วยบิต C/T ในรีจิสเตอร์ T2CON ส่วนโหมดการทำงานของไทมเมอร์ เคนต์เตอร์ 2 มี 3 โหมดคือ โหมดแคปเจอร์หรือตรวจจับสัญญาณ (Capture) , โหมดตั้งค่าอัตโนมัติ (Auto-Reload) และโหมดกำเนิดอัตราเร็วในการสื่อสารข้อมูลอนุกรมหรืออัตราบอด (Baud Rate Generator)

6.1 รีจิสเตอร์ใช้งานพิเศษที่เกี่ยวข้องกับไทมเมอร์ เคนต์เตอร์ 2

6.1.1 รีจิสเตอร์ไทมเมอร์

ประกอบด้วยกัน 2 ตัวคือ TL2 มีแอดเดรสอยู่ที่ CCH และ TH2 มีแอดเดรสอยู่ที่ CDH รีจิสเตอร์ทั้ง 2 ตัวอยู่ในพื้นที่ของรีจิสเตอร์ใช้งานพิเศษ แต่ละตัวมีขนาด 8 บิตในการใช้งานโดยทั่วไปจะใช้รวมกันเป็นรีจิสเตอร์ไทมเมอร์ 2 ขนาด 16 บิต โดยใน TL2 จะใช้เก็บข้อมูล 8 บิตล่าง ส่วน TH2 จะเก็บข้อมูล 8 บิตบน เมื่อนำมาใช้งานร่วมกันสามารถเก็บค่าของการนับได้สูงสุด 65,536 ค่าหรือ FFFFH เมื่อนับเกินค่านี้จะวนไปเริ่มนับ 0000H ใหม่ และเกิดค่านับเกินมีการเซตที่บิต TF2 ในรีจิสเตอร์ T2CON เพื่อแสดงให้ทราบว่าเกิดการนับเกินค่าสูงสุด

6.1.2 รีจิสเตอร์ T2CON (Timer/Counter 2 Control Register)

เป็นรีจิสเตอร์ขนาด 8 บิต แสดงดังภาพที่ 7.8 มีแอดเดรสอยู่ที่ C8H ในพื้นที่ของรีจิสเตอร์ใช้งานพิเศษ โดยทำหน้าที่ควบคุมการทำงานของไทมเมอร์ เคนต์เตอร์ 2 เข้าถึงได้ในระดับบิต มีรายละเอียดการทำงานดังนี้

T2CON.7	T2CON.6	T2CON.5	T2CON.4	T2CON.3	T2CON.2	T2CON.1	T2CON.0
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	$C/\overline{T2}$	$CP/\overline{RL2}$

ภาพที่ 7.8 รีจิสเตอร์ควบคุมการทำงานของไทมเมอร์ เคนต์เตอร์ 2 (T2CON)

TF2: T2CON7 (Timer2 Overflow Flag) บิตแสดงการเกิดค่านับเกินของไทมเมอร์ 2 จะเซตเมื่อค่ารีจิสเตอร์ Timer 2 เกิดการนับเกิน การเคลียร์บิตนี้ทำได้ด้วยกระบวนการทางซอฟต์แวร์ และไม่เกิดการเซตถ้าบิต RCLK หรือบิต TCLK เป็น “1” เมื่อใช้ไทมเมอร์ 2 ในการกำหนดอัตรารับส่งข้อมูลในการสื่อสารแบบอนุกรม

EXF2: T2CON6 (Timer2 External Flag) บิตแสดงการเกิดอินเตอร์รัปต์ในไทมเมอร์ 2 เกิดการเซตเมื่อเกิดคาบแคปเจอร์หรือรีโวลด์ขึ้น ซึ่งเกิดจากการเปลี่ยนแปลงระดับลอจิกจาก “1” เป็น “0” ที่ขาอินพุต T2EX และบิต EXEN2 เป็น “1” ในกรณีที่อนุญาตให้มีการอินเตอร์รัปต์ และบิต EXF2 ถูกเซตซีพียูจะถูกย้ายไปทำที่โปรแกรมย่อยการอินเตอร์รัปต์ของไทมเมอร์ 2 ทั้งนี้ บิตนี้สามารถเคลียร์ด้วยกระบวนการทางซอฟต์แวร์เท่านั้น

RCLK: T2CON5 (Receive Clock Flag) บิตเลือกใช้ไทมเมอร์ 2 เป็นตัวกำหนดอัตรารับข้อมูล จะเกิดการเซตทำให่วงจรพอร์ตอนุกรมภายในไอซี MCS-51 ใช้พัลส์จากค่านับเกินของไทมเมอร์ 2 เป็นสัญญาณพิกการรับข้อมูลในโหมด 1 และ 3 ถ้าบิตนี้เป็น “0” จะใช้พัลส์จากไทมเมอร์ 1

TCLK: T2CON4 (Transmit Clock Flag) บิตเลือกใช้ไทมเมอร์ 2 เป็นตัวกำหนดอัตราส่ง

ข้อมูล จะเกิดการเซตทำให้วงจรพอร์ตอนุกรมภายในไอซี MCS-51 ใช้พัลส์จากค่านับเกินของไทมเมอร์ 2 เป็นสัญญาณนาฬิกา การส่งข้อมูลในโหมด 1 และ 3 ถ้าบิตนี้เป็น “0” จะใช้พัลส์จากไทมเมอร์ 1

EXEN2: T2CON3 (Timer2 External Enable Flag) บิตเลือกให้ไทมเมอร์ 2 ถูกควบคุมการทำงานจากสัญญาณที่ขา T2EX โดยเมื่อมีการเซตเป็น “1” จะอนุญาตให้มีการแคปเจอร์หรือรีโหลดที่ขา T2EX ให้เกิดขึ้น โดยไทมเมอร์ 2 ต้องไม่ถูกนำไปใช้ในการสื่อสารพอร์ตอนุกรม หรือบิต RCLK หรือบิต TCLK เป็น “1” ถ้าบิตนี้เป็น “0” ไทมเมอร์ 2 จะไม่สนใจเหตุการณ์ที่ขา T2EX

TR2: T2CON2 (Timer2 Run Control Bit) บิตควบคุมการเริ่มต้น และหยุดการทำงานของ ไทมเมอร์ เป็นการควบคุมทางซอฟต์แวร์ ถ้าบิตนี้มีลอจิกเป็น “0” หยุดการทำงานของไทมเมอร์ 2 และถ้ามีลอจิกเป็น “1” เริ่มต้นการทำงานของไทมเมอร์ 2

$C/\overline{T2}$: T2CON1 (Timer or Counter Selector) บิตเลือกลักษณะการทำงานของไทมเมอร์ เคาท์เตอร์ 2 ถ้าบิตนี้มีลอจิกเป็น “0” เลือกให้ทำงานเป็นไทมเมอร์ ถ้าเป็น “1” เลือกให้ทำงานเป็นเคาท์เตอร์

$CP/\overline{RL2}$: T2CON0 (Capture/Reload Flag) บิตแสดงสถานการณ์ทำงานของไทมเมอร์ 2 ถ้าบิตนี้มีลอจิกเป็น “0” เลือกให้ไทมเมอร์ 2 ทำงานโหลดค่าของการนับแบบอัตโนมัติ ที่มีการโหลดค่าจากรีจิสเตอร์ RCAP2L, RCAP2H เมื่อ ไทมเมอร์ 2 เกิดค่านับเกิน หรือเกิดการเปลี่ยนระดับลอจิก (เปลี่ยนจาก “1” เป็น “0”) ที่ขา T2EX ในกรณีที่บิต EXEN2 เป็น “1” และบิต RCLK หรือ TCLK เป็น “1” จะยอมให้การรีโหลดแบบอัตโนมัติเกิดขึ้นเมื่อไทมเมอร์ 2 เกิด ค่านับเกินเท่านั้น ถ้ามีสถานะลอจิกเป็น “1” มีการแคปเจอร์เกิดขึ้น เมื่อเกิดการเปลี่ยนระดับลอจิก (จาก “1” เป็น “0”) ที่ขา T2EX ในกรณีบิต EXEN2 เป็น “1”

6.1.3 T2MOD (Timer/Counter 2 Mode Control Register)

เป็นรีจิสเตอร์ขนาด 8 บิต อยู่แอดเดรสที่ C9H ของรีจิสเตอร์ใช้งานเฉพาะ ไม่สามารถเข้าถึงได้ในระดับบิต ทำหน้าที่เลือกโหมดการทำงานของไทมเมอร์ เคาท์เตอร์ 2 แสดงดังภาพที่ 7.9 มีบิตสำหรับกำหนดการทำงานเพียง 2 บิตคือ

บิต7	บิต6	บิต5	บิต4	บิต3	บิต2	บิต1	บิต0
-	-	-	-	-	-	T2OE	DCEN

ภาพที่ 7.9 รีจิสเตอร์โหมดการทำงานของไทมเมอร์ เคาท์เตอร์ 2 (T2MOD)

T2OE (Timer2 Output Enable Bit) ใช้อุญาตการทำงานที่เอาต์พุตของไทมเมอร์ 2 ถ้ากำหนดให้บิตนี้มีลอจิกเป็น “0” ไม่อนุญาตให้มีการทำงาน ถ้ามีลอจิกเป็น “1” อนุญาตให้มีการทำงาน

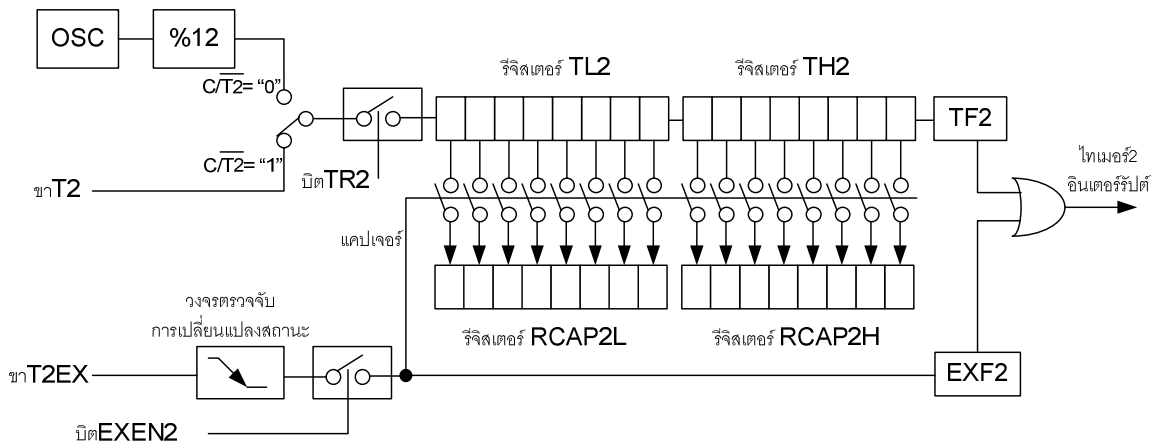
DCEN (Timer2 Counter Enable Bit) กำหนดให้ไทมเมอร์ 2 ทำงานเป็นตัวนับหรือเคาท์เตอร์แบบขึ้นและลง ถ้ากำหนดให้บิตนี้มีลอจิกเป็น “0” ไม่อนุญาตให้มีการทำงาน ถ้ามีลอจิกเป็น “1” อนุญาตให้มีการทำงานเป็นตัวนับหรือเคาท์เตอร์แบบขึ้นลง

6.2 โหมดการทำงานของไทมเมอร์ เคน์เตอร์ 2

ไทมเมอร์ เคน์เตอร์ 2 เลือกการทำงานได้ 3 โหมด โดยกำหนดที่บิต RCLK+TCLK, CP/ $\overline{RL2}$ และ TR2 ในรีจิสเตอร์ T2CON แสดงดังในตารางที่ 7.2 โดยมีรายละเอียดดังต่อไปนี้

ตารางที่ 7.2 การเลือกโหมดทำงานของไทมเมอร์ เคน์เตอร์ 2

RCLK+TCLK	CP/ $\overline{RL2}$	TR2	โหมด
0	0	1	16 บิต ตั้งค่าการนับอัตโนมัติ (Auto-Reload)
0	1	1	แคปเจอร์หรือตรวจจับ(Capture)
1	X	1	กำหนดอัตราบอด (Baud Rate Generator)
X	X	0	ปิดการใช้งาน(Off)



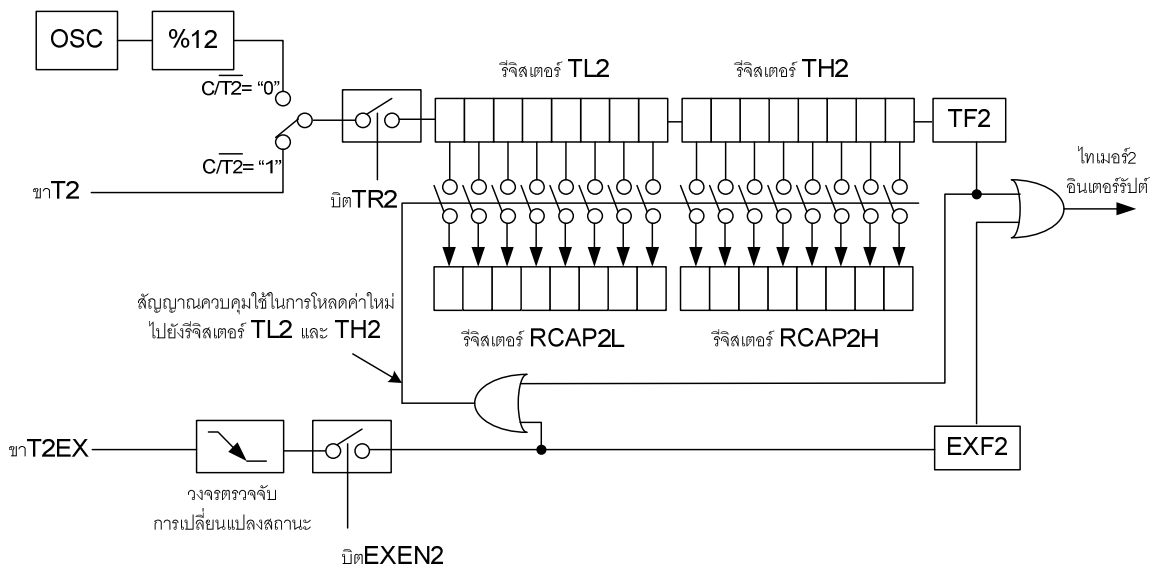
ภาพที่ 7.10 การทำงานในโหมดแคปเจอร์ หรือตรวจจับสัญญาณของไทมเมอร์เคน์เตอร์ 2

6.2.1 การทำงานในโหมดแคปเจอร์หรือตรวจจับสัญญาณ (Capture)

การทำงานในโหมดนี้แบ่งได้เป็น 2 ลักษณะเมื่อกำหนดสถานะของบิต EXEN2 ดังนี้ กำหนดให้บิต EXEN2 เป็น “0” เมื่อไทมเมอร์ 2 ใช้งานเป็นไทมเมอร์หรือเคน์เตอร์ ขนาด 16 บิตเมื่อนับค่าเกินรอบแล้ว จะเกิดการเซตบิต TF2 และทำให้เกิดการอินเตอร์รัปต์ กำหนดให้บิต EXEN2 เป็น “1” เมื่อไทมเมอร์ 2 ใช้งานในโหมดแคปเจอร์ โดยใช้ ตรวจจับที่ขา T2EX เป็นอินพุต (ขา P1.1) ที่ขา T2EX มีวงจรตรวจจับการเปลี่ยนแปลงระดับลอจิกจาก “1” เป็น “0” รีจิสเตอร์ TL2 และ TH2 ทำการนับค่าเพิ่มขึ้นไปเรื่อย ๆ จนกว่าที่ขา T2EX จะตรวจจับการเปลี่ยนแปลงได้ หลังจากนั้นค่าของรีจิสเตอร์ TL2 และ TH2 ถูกส่งไปยังรีจิสเตอร์ RCAP2L และ RCAP2H และยังเซตบิต EXF2 ให้เป็น “1” เพื่อสร้างสัญญาณอินเตอร์รัปต์ให้เกิดขึ้น การทำงานของไทมเมอร์ 2 ในโหมดแคปเจอร์ แสดงดังภาพที่ 7.10

6.2.2 การทำงานในโหมดตั้งค่าการนับอัตโนมัติ

การทำงานในโหมดนี้ มี 2 ลักษณะโดยเลือกที่บิต EXEN2 ในกรณีที่บิต EXEN2 เป็น “0” เมื่อไทมเมอร์ 2 ทำการนับค่าเกินรอบแล้ว จะเกิดการเซตที่บิต TF2 และ ทำการเรียกค่าการนับใหม่จากรีจิสเตอร์ RCAP2L และ RCAP2H โดยอัตโนมัติ การกำหนดค่าของรีจิสเตอร์ RCAP2L และ RCAP2H สามารถกระทำได้ด้วยกระบวนการทางซอฟต์แวร์เหมือนการทำงานในโหมดตั้งค่าการนับอัตโนมัติในไทมเมอร์ 0 และไทมเมอร์ 1 ในกรณีที่บิต EXEN2 มีสถานะเป็น “1” ไทมเมอร์ 2 นับเหมือนเดิม แต่เงื่อนไขตั้งค่าการนับใหม่ โดยอัตโนมัติ จากรีจิสเตอร์ RCAP2L และ RCAP2H จะใช้การเปลี่ยนแปลงของระดับลอจิกจาก “1” เป็น “0” ที่ขา T2EX เป็นตัวกำหนด และทำการเซตบิต EXF2 เพื่อสร้างสัญญาณอินเตอร์รัปต์แทนบิต TF2 แสดงดังภาพที่ 7.11



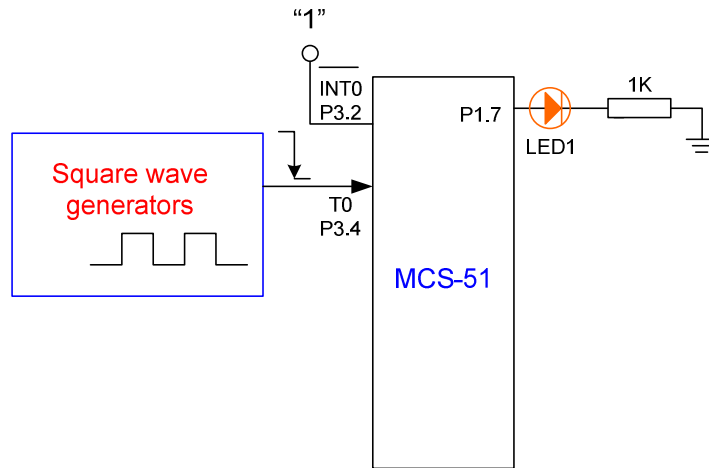
ภาพที่ 7.11 การทำงานในโหมดตั้งค่าการนับอัตโนมัติ ของไทมเมอร์ เคนเตอร์ 2

6.2.3 การทำงานในโหมดกำเนิดอัตราเร็วในการสื่อสารข้อมูลอนุกรมหรืออัตราบอด

การทำงานในโหมดนี้เกิดขึ้นเมื่อเซตบิต RCLK และ/หรือบิต TCLK ในรีจิสเตอร์ T2CON ให้เป็น “1” โดยบิต TF2 จะไม่เกิดการเซต และไม่มีการสร้างสัญญาณอินเตอร์รัปต์ เมื่ออยู่ในโหมดนี้ และบิต EXEN2 ถูกเซต การเปลี่ยนแปลงระดับลอจิกจาก “1” เป็น “0” ที่ขา T2EX จะส่งผลให้บิต EXF2 เกิดการเซต แต่ไม่มีการเรียกค่าจากรีจิสเตอร์ RCAP2L และ RCAP2H ดังนั้นจึงทำให้ขา T2EX สามารถใช้เป็นค่าอินพุต สำหรับการอินเตอร์รัปต์ได้เป็นกรณีพิเศษ ในโหมดนี้ใช้ไทมเมอร์ 2 เป็นตัวกำหนดอัตราความเร็วในการส่งข้อมูล

7. การเขียนโปรแกรมใช้งานไทมเมอร์ เคน์เตอร์

ตัวอย่างที่ 1 กำหนดการทำงานเป็นเคน์เตอร์โดยใช้ไทมเมอร์ 0 ในโหมด 0 ให้รับสัญญาณอินพุตจากภายนอก ป้อนเข้าที่ขา T0 (P3.4) ของไอซี MCS-51 เป็นสัญญาณพัลส์จำนวน 1000 ลูก หลังจากมีการนับจนครบให้แอลอีดีแสดงผลเป็นไฟกระพริบที่ P1.7 จำนวน 5 ครั้ง วงจรแสดงดังภาพที่ 7.12 การเขียนโปรแกรมมีขั้นตอนการกำหนดค่าดังนี้



ภาพที่ 7.12 แสดงวงจรของตัวอย่างที่ 1

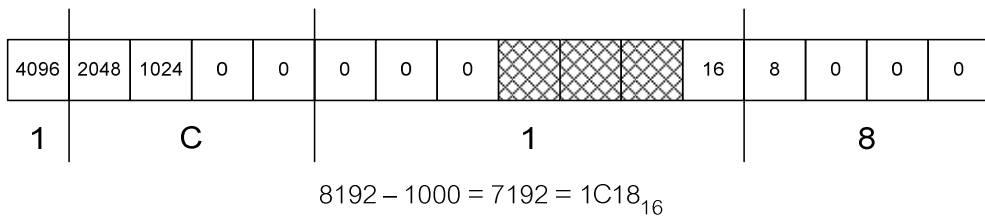
ขั้นตอนที่ 1 กำหนดค่าของบิตในรีจิสเตอร์ TMOD โดยให้ GATE = “1” C/T = “1”

ขั้นตอนที่ 2 กำหนดค่าเริ่มต้นการนับที่รีจิสเตอร์ (Count Up Register) TL0 และ TH0 ดังนั้นค่าข้อมูลในรีจิสเตอร์ $8,192 - 1,000 = 7,192 = 1C18H$ กำหนดให้ TH0 = 1CH และ TL0 = 18H แสดงการหาค่า TH0 และ TL0 แสดงดังภาพที่ 7.13

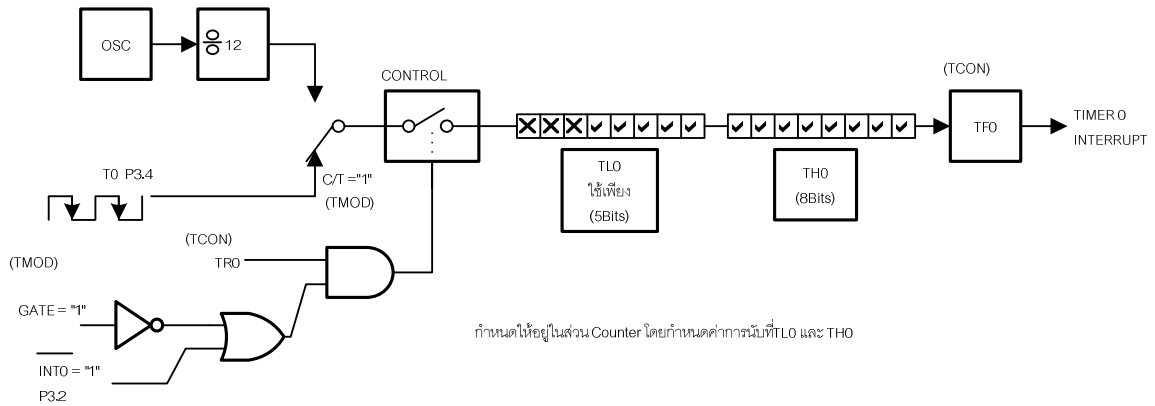
ขั้นตอนที่ 3 เริ่มต้นการนับโดยกำหนดค่าที่บิต TR0 = “1” โดยคำสั่ง SETB TR0

ขั้นตอนที่ 4 ตรวจสอบสถานะของบิตนับเกิน TF0 โดยคำสั่ง JNB TF0, \$

จากขั้นตอนทั้งหมดสามารถสรุป การกำหนดค่าของรีจิสเตอร์ต่างๆ แสดงดังภาพที่ 7.14



ภาพที่ 7.13 แสดงการหาค่า TH0 และ TL0 ไทมเมอร์ 0 โหมด 0



ภาพที่ 7.14 การกำหนดค่าของรีจิสเตอร์โดยใช้ไทมเมอร์ 0 โหมด 0

```

ORG 0000H ; เริ่มต้นที่หน่วยความจำ โปรแกรมแอดเดรส 0000H
MAIN: CLR P1.7 ; เคลียร์ค่าที่พอร์ต P1.7
      SETB P3.2 ; ให้พอร์ต P3.2 มีสถานะลอจิก "1"
      SETB P3.4 ; ให้พอร์ต P3.4 เป็นขา T0 มีสถานะลอจิก "1"
      CLR TF0 ; ทำการเคลียร์ข้อมูลของบิตแสดงสถานะเกินให้เป็น "0"
      MOV R7, #05 ; กำหนดการแสดงผลกระพริบที่แอลอีดีจำนวน 5 ครั้ง
      MOV TMOD, #00001100B ; กำหนดค่าบิต GATE และ C/T = "1" เลือการทำงานโหมด 0
      MOV TH0, #1CH ; กำหนดค่าการนับโดยให้ TH0
      MOV TL0, #18H ; กำหนดค่าการนับโดยให้ TL0
      SETB TR0 ; เริ่มต้นการนับ
      JNB TF0, $ ; คำนับเกินไม่ถูกเซต ให้อ่านทำงานที่บรรทัดเดิม
;*** โปรแกรมแสดงการกระพริบที่แอลอีดี ***
LOOP: CPL P1.7 ; กลับสถานะของแอลอีดี
      CALL DELAY ; เรียกโปรแกรมย่อยหน่วงเวลา
      DJNZ R7, LOOP ; กำหนดกระพริบที่แอลอีดี 5 รอบ
      SJMP $ ; กำหนดให้ทำงานที่บรรทัดเดิม
;*** โปรแกรมย่อยหน่วงเวลา ***
DELAY: MOV R1, #0FFH
DELAY1: MOV R2, #0FFH
        DJNZ R2, $
        DJNZ R1, DELAY1

```


RET

END

ข้อสังเกต ที่ได้จากโปรแกรมคำสั่งในตัวอย่างที่ 1 สรุปได้เป็นหัวข้อดังนี้

ข้อที่ 1 เริ่มต้นป้อนสัญญาณที่ขา T0 แอลอีดีที่ต่อกับพอร์ต P1.7 ไม่มีการแสดงผล หลังจากป้อนสัญญาณ และเริ่มต้นทำการนับค่าข้อมูล (SETB TR0) จนครบ 1000 ครั้ง แอลอีดี P1.7 จะกระพริบจำนวน 5 ครั้ง

ข้อที่ 2 การกำหนดค่าการนับในโหมด 0 มีเพียง 13 บิต คือ TH0 = 8 บิต ที่ TL0 = 5 บิต โดย TL0 สามารถกำหนดเพียง 5 บิต 00000B -11111B หรือ 32_{10} โหมดนี้จะนับได้สูงสุดเท่ากับ $(256)(32) = 8,192_{10}$

ข้อที่ 3 ตรวจสอบบิต TF0 ใช้คำสั่ง JNB TF0, \$ หมายถึงถ้าบิตนับเกิน (TF0) ไม่ถูกเซต (TF0 = 0) ให้ทำคำสั่งในบรรทัดเดิม ถ้ามีการเซตที่บิตนับเกิน (TF0 = 1) ให้ทำคำสั่งในบรรทัดถัดไป

ตัวอย่างที่ 2 จากโปรแกรมใช้ไทมเมอร์ 0 โหมด 1 ทำงานเป็นเคาน์เตอร์ กำหนดตั้งค่าการนับจำนวน 256 ครั้ง หลังจากนั้นให้กลับสถานะที่บิต P3.0 ให้หาค่าของจำนวนแมชชีนไซเคิล

	ORG	0000H	;1 เริ่มต้นที่หน่วยความจำโปรแกรมแอดเดรส 0000H
MAIN:	MOV	TMOD, #0000101B	;2 [2] เลือกการทำงานที่โหมด 1 (Gate = 0, C/T = 1)
LOOP:	MOV	TH0, #0FFH	;3 [2] กำหนดค่าการนับโดยให้ TH0 = 0FH
	MOV	TL0, #00H	;4 [2] กำหนดค่าการนับโดยให้ TL0 = 00H (นับ 256)
	SETB	TR0	;5 [1] เริ่มต้นการนับที่ ไทมเมอร์ 0
	JNB	TF0, \$;6 [2] ไม่เกิดค่านับเกิน ใหวนทำงานที่บรรทัดเดิม
	CLR	TR0	;7 [1] หยุดการนับที่ไทมเมอร์ 0
	CLR	TF0	;8 [1] เคลียร์ข้อมูลของบิตแสดงสถานะเกินให้เป็น "0"
	CPL	P3.0	;9 [1] กลับค่าข้อมูลที่พอร์ต P3.0
	SJMP	LOOP	;10 [2] วนกลับไปทำใหม่ที่เลเบล LOOP
	END		

[x] : โดย x เป็นจำนวนแมชชีนไซเคิลของแต่ละคำสั่ง

จากโปรแกรมคำสั่งในตัวอย่างที่ 2 หาผลลัพธ์ของจำนวนแมชชีนไซเคิล และเวลาได้ดังนี้

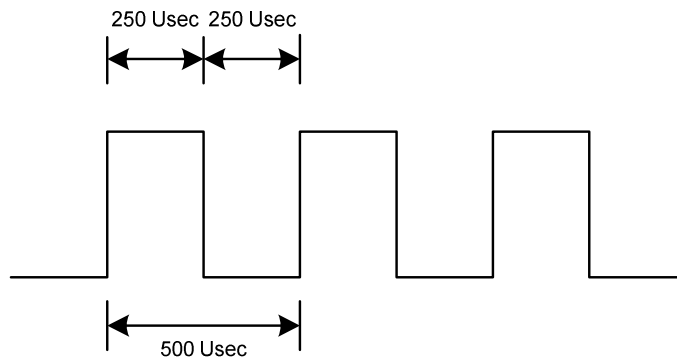
คำสั่ง JNB TF0, \$ บรรทัดที่ 6 การวนตรวจสอบในแต่ละรอบจะใช้เวลา 2 แมชชีนไซเคิล โดยจำนวนรอบที่วนขึ้นกับค่าที่โหลดให้กับ TH0 และ TL0 เพื่อนับค่าเพิ่มขึ้น

การกำหนดค่าให้รีจิสเตอร์ TH0 = FFH และ TL0 = 00H ให้นับเท่ากับ 256 แมชชีนไซเคิล เป็นค่าที่กำหนดให้นับจำนวน 256 ครั้ง รวมกับค่าเวลาที่สูญเสียขณะที่ไอซี MCS-51 ปฏิบัติตามคำสั่งตั้งแต่บรรทัดที่ 3-9 อีก 10 แมชชีนไซเคิล (จำนวนแมชชีนไซเคิลที่ใช้ในโปรแกรม ไม่รวมคำสั่งของ JNB TF0, \$ ที่ใช้ในการนับขึ้นของรีจิสเตอร์ TH0 และ TL0) รวมค่าการนับเป็น $256 + 10 = 266$ แมชชีนไซเคิล

ตัวอย่างที่ 3 เขียนโปรแกรมกำหนดให้ใช้ไทมเมอร์ 0 โหมด 2 สร้างสัญญาณพัลส์ขนาด 500 μ S โดยใช้คริสตอลขนาด 12 MHz โดยมีเอาต์พุตอยู่ที่ P3.0 แสดงคังภาพที่ 7.15 โดยคิดเฉพาะค่าที่นับใน TL0

กำหนดค่าให้กับรีจิสเตอร์ TH0 เพื่อเก็บค่าเริ่มต้นการนับ โดยรีจิสเตอร์ TL0 เพิ่มค่าขึ้นทีละหนึ่งค่าในทุกๆ แมกซ์ซินไซเคิล การคำนวณค่าระยะเวลาของหนึ่งแมกซ์ซินไซเคิล ใช้เวลานานเท่ากับคาบเวลาของออสซิลเลเตอร์จำนวน 12 คาบ คิดเป็นค่าอัตราการนับในแต่ละครั้งใช้เวลาเท่ากับ $1/12$ เท่าของความถี่ออสซิลเลเตอร์ หรือ $T = MC \times 12/f\text{-xtal}$

เมื่อใช้คริสตอลขนาด 12 MHz ใน 1 แมกซ์ซินไซเคิลจึงนับค่าได้ 1 μ S ดังนั้นค่าเริ่มต้นของการนับที่เก็บไว้ในรีจิสเตอร์ TH0 คือค่า 2'Complement ของ 250 มีค่าเท่ากับ 06H



ภาพที่ 7.15 แสดงสัญญาณที่ได้จากขาพอร์ต P3.0

```

ORG    0000H                ; เริ่มต้นที่แอดเดรส 0000H
MAIN:  MOV    TMOD,#00000010B ; กำหนดให้เป็นไทมเมอร์ 0 โหมด 2
        MOV    TH0,# 06H      ; เก็บค่าเริ่มต้นในการนับไว้ในรีจิสเตอร์ TH0
        MOV    TL0,# 06H      ; ค่าที่ทำการนับเก็บไว้ในรีจิสเตอร์ TL0
        SETB   TR0            ; กำหนดให้ไทมเมอร์ 0 เซตที่บิต TR0 ให้เริ่มทำการ RUN
LOOP:  JNB    TF0, LOOP        ; ไม่เกิดค่านับเกิน ให้วนทำงานที่บรรทัดเดิม
        CLR    TF0            ; เคลียร์สถานะของบิต TF0 เมื่อมีค่านับเกิน
        CPL    P3.0           ; กลับค่าสถานะที่บิต P3.0
        SJMP  LOOP            ; วนกลับไปทำใหม่ที่เลเบล LOOP
END

```

ตัวอย่างที่ 4 ป้อนสัญญาณ 100 KHz เข้ามาทางขา T0 ของไอซี MCS-51 และเขียนโปรแกรมการทำงานเป็น เคนเตอร์โหมด ทำงานโหมด 0 โดยกำหนดค่าที่ TH0 = FFH และ TL0=FFH ให้วาดรูปสัญญาณเอาต์พุตที่ขา P1.7 โดยใช้ความถี่คริสตอล 12 MHz มีโปรแกรมดังต่อไปนี้

```

ORG    0000H                ; 1 เริ่มต้นที่หน่วยความจำโปรแกรมแอดเดรส 0000H

```

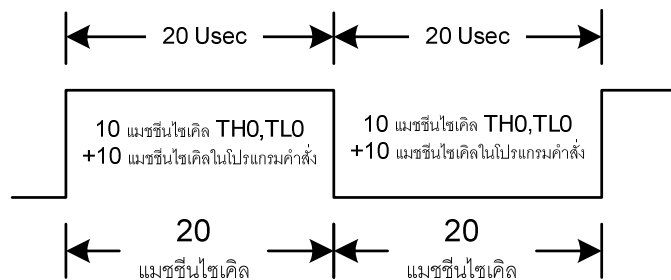
```

MAIN:      MOV  TMOD, #00000100B ; [2] 2 เลือกการทำงาน Counter โหมด 0
LOOP:     MOV  TH0, #0FFH ; [2] 3 กำหนดค่าการนับโดยให้ TH0 = FFH
          MOV  TL0, #00H ; [2] 4 กำหนดค่าการนับโดยให้ TL0 = FFH (นับ 256)
          SETB TR0 ; [1] 5 เริ่มต้นการนับ
          JNB  TF0, $ ; [2] 6 ไม่เกิดค่านับเกิน ให้งานทำงานที่บรรทัดเดิม
          CLR  TR0 ; [1] 7 หยุดการนับ
          CLR  TF0 ; [1] 8 เคลียร์ข้อมูลของบิตแสดงสถานะเกินให้เป็น "0"
          CPL  P3.0 ; [1] 9 กลับค่าข้อมูลที่พอร์ต P3.0
          SJMP LOOP ; [2] 10 วนกลับไปทำใหม่ที่เลเบล LOOP
          END

```

จากโปรแกรมคำสั่งหาผลลัพธ์ของจำนวนแมชชีนไซเคิล และเวลาได้ดังนี้

- 1) $T = MC \times 12 / f_{xtal}$ เมื่อใช้คริสตอลขนาด 12 MHz ใน 1 แมชชีนไซเคิลจึงนับค่าได้ $1 \mu S$
- 2) สัญญาณพัลส์ที่ป้อนเข้ามาความถี่ 100 KHz จะสูญเสียเวลาพัลส์ 1 ลูก เท่ากับ $10 \mu S$
- 3) ค่าที่โหลดเข้ามาที่ TH0 และ TL0 = FFFFH เมื่อมีพัลส์เข้ามาเพียงลูกเดียวจะเกิดค่านับเกิน จึงทำให้เสียเวลาไป 10 แมชชีนไซเคิล เป็นค่าของการนับขึ้นที่ TH0, TL0
- 4) จำนวนรอบคำสั่งขณะที่ไอซี MCS-51 ปฏิบัติตามคำสั่งตั้งแต่บรรทัดที่ 3-9 SJMP LOOP ถึง LOOP เท่ากับ $2+2+1+1+1+1+2$ จำนวน 10 แมชชีนไซเคิล (ไม่รวมคำสั่งของ JNB TF0, WAIT ที่ใช้ในการนับขึ้นของรีจิสเตอร์ TH0 และ TL0) เป็นค่าของคำสั่งในโปรแกรม
- 5) ในโปรแกรมมีคำสั่งกลับสถานะบิต P3.0 ด้วยคำสั่ง CPL P3.0 เพียงครั้งเดียว ดังนั้นค่าจากการนับของ TH0, TL0 + ค่าของคำสั่ง LOOP เท่ากับ $10+10 = 20$ แมชชีนไซเคิล สัญญาณแสดงดังภาพที่ 7.16



ภาพที่ 7.16 แสดงรูปสัญญาณและรอบเวลาในการทำงาน

สรุป

ไทมเมอร์ เคนต์เตอร์ หมายถึงความสามารถในการนับค่า หรือเป็นการตั้งเวลา อยู่ในรีจิสเตอร์ TMOD ซึ่งเป็นรีจิสเตอร์ใช้งานพิเศษ ทำหน้าที่ควบคุมโหมดการทำงาน ประกอบด้วยไทมเมอร์ เคนต์เตอร์ จำนวน 2 ตัวคือไทมเมอร์ 0 และไทมเมอร์ 1 การใช้งานของไทมเมอร์ ทั้ง 2 ตัว สามารถแยกการทำงานออกจากกันได้อย่างอิสระโดยเก็บค่าที่กำหนดการนับหรือตั้งเวลาไว้ในข้อมูลของรีจิสเตอร์ TLx และ THx

รีจิสเตอร์ไทมเมอร์ เป็นรีจิสเตอร์ขนาด 8 บิต อยู่ในรีจิสเตอร์ใช้งานพิเศษมี 4 ตัวคือ รีจิสเตอร์ TL0 รีจิสเตอร์ TH0, รีจิสเตอร์ TL1 และรีจิสเตอร์ TH1 สามารถใช้งานเป็นคู่เพื่อให้ได้รีจิสเตอร์ที่มีขนาด 16 บิต คือ TL0 กับ TH0 และ TL1 กับ TH1 เมื่อนำมารวมกันสามารถนับค่าได้ถึง 65,536 จิสเตอร์ TLx และ THx เพิ่มขึ้นทีละหนึ่ง ค่าจนเป็น “1” ทุกบิต ถ้ามีการนับอีกครั้งจะเกิดการนับเกิน หมายถึงครบกำหนดของเวลาที่ตั้งไว้ การกำหนดค่าในรีจิสเตอร์ TLx และ THx ให้นำค่านับได้สูงสุดไปลบออกจากค่าที่ต้องการนับ ผลลัพธ์ที่ได้ในรีจิสเตอร์ THx มีค่าเท่ากับ 8 บิตบน และรีจิสเตอร์ TLx มีค่าเท่ากับ 8 บิตล่าง หรืออาจใช้วิธีกำหนดค่าข้อมูลให้กับรีจิสเตอร์ THx, TLx โดยนำเอาจำนวนที่ต้องการนับมาหาค่า 2'S Complement

รีจิสเตอร์ TCON หน้าทีควบคุมสัญญาณการอินเตอร์รัปต์ของไทมเมอร์ และทำหน้าที่ตรวจสอบค่าการนับของไทมเมอร์ เมื่อเกิดค่านับเกิน เซต และเคลียร์บิต TFX ด้วยวิธีทางซอฟต์แวร์ หรือใช้คำสั่ง ไอออนย้ายข้อมูล

บิต C/T อยู่ในรีจิสเตอร์ TMOD เป็นการกำหนดฟังก์ชันทำงานของไทมเมอร์ โดยใช้สถานะลอจิกของบิตนี้เลือกให้เป็นเคนต์เตอร์ หรือไทมเมอร์ หากกำหนดให้ลอจิกของบิตนี้เป็น “0” เป็นการเลือกให้เป็นไทมเมอร์ โดยใช้สัญญาณนาฬิกาจากภายในตัวไอซี MCS-51เอง ถ้าลอจิกของบิตนี้เป็น “1” กำหนดให้เป็นเคนต์เตอร์รับสัญญาณจากอินพุตภายนอกที่ขา T0 และ T1

การทำงานของไทมเมอร์ เคนต์เตอร์ในโหมด 0 ค่าของการนับใช้งานเพียง 13 บิต โดยใช้ข้อมูล ในรีจิสเตอร์ TLx เพียง 5 บิต และข้อมูลในรีจิสเตอร์ THx อีก 8 บิต รวมเป็น 13 บิต การนับอยู่ในช่วง 0000H - 1FFFH (TLx = 00 - 1FH, THx = 00 - FFH) หรือ 0 - 8,191

การทำงานของโหมด 1 การนับมีได้ถึง 16 บิต โดยนำเอาข้อมูลในรีจิสเตอร์ TL0 กับ TH0 หรือรีจิสเตอร์ TL1 กับ TH1 ขนาด 8 บิตมารวมกัน เพื่อให้ได้รีจิสเตอร์ขนาด 16 บิต โดยค่าของการนับอยู่ในช่วง 0000H - FFFFH หรือ 65,536 ค่า

การทำงานของโหมด 2 ทำงานเป็นแบบโหลดค่าข้อมูลใหม่ในการเริ่มนับ หรือโหมดการตั้งค่าใหม่แบบอัตโนมัติ (Auto Reloads Mode) แยกรีจิสเตอร์ไทมเมอร์ ออกเป็น 2 ตัว มีขนาดตัวละ 8 บิต วิธีการเก็บค่าเริ่มต้นการนับกำหนดอยู่ในข้อมูลของรีจิสเตอร์ THx โดยค่าการนับสูงสุด 256 ครั้ง (THx = 00)

โหมด 3 แบ่งการใช้งานของไทมเมอร์ 0 และไทมเมอร์ 1 ออกจากกันโดยไทมเมอร์ 0 แยกรีจิสเตอร์ TH0 และ TL0 ทำให้เกิดเป็นเคนต์เตอร์ 2 ตัวทำงานอิสระต่อกัน และใช้งานได้เพียง 8 บิตเท่านั้น

ไทมเมอร์ เคนต์เตอร์ 2 ของไอซี MCS -51 มีขนาด 16 บิต ใช้งานได้เหมือนกับไทมเมอร์ เคนต์เตอร์ 0 และไทมเมอร์ เคนต์เตอร์ 1 ผู้ใช้สามารถเลือกการทำงานได้ด้วยบิต C/T ในรีจิสเตอร์ T2CON ส่วนโหมดการทำงานของไทมเมอร์ เคนต์เตอร์ 2 มี 3 โหมดคือ โหมดแคปเจอร์หรือตรวจจับสัญญาณ (Capture) โหมดตั้งค่าอัตโนมัติ (Auto-Reload) และโหมดกำเนิดอัตราเร็วในการสื่อสารข้อมูลอนุกรมหรืออัตราบอด (Baud Rate Generator) มีรีจิสเตอร์ใช้งานพิเศษที่เกี่ยวข้องกับไทมเมอร์ เคนต์เตอร์ 2 รีจิสเตอร์ไทมเมอร์ รีจิสเตอร์ T2CON ทำหน้าที่ควบคุมการทำงานของไทมเมอร์ เคนต์เตอร์ 2 รีจิสเตอร์ T2MOD ทำหน้าที่เลือกโหมดการทำงาน ของไทมเมอร์ เคนต์เตอร์ 2