



ปีการศึกษา 2531

มอเตอร์ไมโครคอนโทรลเลอร์
(MOTOR MICROCONTROLLER)

โดย

นาย พรศักดิ์ เหล่าบุญชัย 27-1141

อาจารย์ที่ปรึกษา

อาจารย์ ฝึกฝน เล่าสงคราม



ปริญญาโท ประจำปีการศึกษา 2531

ภาควิชา ไฟฟ้ากำลัง

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง มอเตอร์ไมโครคอนโทรลเลอร์ (MOTOR MICROCONTROLLER)

ผู้จัดทำ

นาย พรศักดิ์ เหล่านฤชัย 27-1141

อาจารย์ที่ปรึกษา

(อาจารย์ พิเศษ เลาสงคราม)

สารบัญ

เรื่อง

หน้า

บทคัดย่อ

บทนำ..... 1-2

บทที่ 1 ทฤษฎีและหลักการควบคุมมอเตอร์เหนี่ยวนำ 3-17

1.1 ทฤษฎีการควบคุมมอเตอร์เหนี่ยวนำ..... 3-7

1.2 การมอดูเลตตามความกว้างของพัลส์..... 7-14

1.3 วงจรอินเวอร์เตอร์สามเฟสแบบสะพานเต็ม..... 14-16

1.4 วงจรขับ..... 16-18

บทที่ 2 ไมโครโปรเซสเซอร์ MCS 8048

2.1 แนะนำไมโครโปรเซสเซอร์ MCS 8048..... 18-19

2.2 หน่วยความจำ

2.2.1 หน่วยความจำข้อมูล..... 20-23

2.2.2 หน่วยความจำข้อมูลโปรแกรม..... 23-24

2.3 พอร์ต..... 25

2.4 สัญญาณนาฬิกาของระบบและสัญญาณควบคุม

2.4.1 สัญญาณควบคุมเอาต์พุต..... 26-27

2.4.2 สัญญาณควบคุมอินพุต..... 27-28

2.5 การอินเทอร์รัพท์..... 29-30

2.6 ไทเมอร์/เคาน์เตอร์ (TIMER/COUNTER)..... 30-33

2.7 ขาอินพุตสำหรับการทดสอบ..... 33

2.8 ชุดคำสั่งของ 8048..... 33-36

2.9 การต่อขยายหน่วยความจำโปรแกรมภายนอก..... 37-38

บทที่ 3 การคำนวณสร้างสัญญาณมอดูเลตตามความกว้างของพัลส์

3.1 การคำนวณหาค่าตัวแปร h ของ PWM 30 พัลส์..... 40-43

3.2 การสร้างข้อมูล PWM 30 พัลส์..... 43-45

3.3 ตารางข้อมูลสัญญาณมอดูเลตตามความกว้างของพัลส์ 30 พัลส์..... 46

3.4 ตารางข้อมูลสัญญาณมอดูเลตตามความกว้างของพัลส์ 24 พัลส์..... 47

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ตารางข้อมูลสัญญาเมอดูเลตตามความกว้างของฟิล์ม 18 ฟิล์ม.....	48
3.6 ตารางข้อมูลสัญญาเมอดูเลตตามความกว้างของฟิล์ม 12 ฟิล์ม.....	49
บทที่ 4 การทดลองและการวัดผล.....	50-52
บทที่ 5 บทวิจารณ์และสรุป.....	53-54
กิตติกรรมประกาศ.....	55
ภาคผนวก คู่มือการใช้อุปกรณ์.....	56-63
เอกสารอ้างอิง.....	64



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มอเตอร์ไมโครคอนโทรลเลอร์

พรศักดิ์ เหล่าบุญชัย

ผศ.พิพัฒน์ เล่าสงคราม อาจารย์ที่ปรึกษา

ปีการศึกษา 2531

บทคัดย่อ

บทความนี้กล่าวถึงผลการวิจัย และพัฒนาสร้างต้นแบบนาเวอร์ทรานซิสเตอร์ อินเวอร์เตอร์ เพื่อใช้ในงานควบคุมการทำงานของมอเตอร์เหนี่ยวนำ 3 เฟส การทำงานของวงจรอินเวอร์เตอร์ถูกควบคุมด้วยระบบไมโครโปรเซสเซอร์ 8 บิต ตระกูล MCS 8048 โดยใช้หลักการ pulse width modulation (PWM) วงจรที่ได้พัฒนาขึ้น ใช้ทรานซิสเตอร์เบอร์ 4DH 07 เป็นอุปกรณ์สวิตซ์กำลัง โดยออกแบบให้สามารถทำงานที่แรงดัน 55 โวลต์ จากการทดลองใช้ขับเคลื่อนมอเตอร์เหนี่ยวนำ 3 เฟส แรงดันพิกัด 220 โวลต์ โดยการให้หม้อแปลงเพิ่มไฟแรงดันขาออกของอินเวอร์เตอร์เป็น 4 เท่า ในสภาวะไร้อภาระ สามารถขับเคลื่อนมอเตอร์ให้หมุนได้ในช่วงความเร็ว 429-2400 รอบต่อนาที หรือสามารถปรับความถี่ที่อินเวอร์เตอร์สร้างออกมาในช่วง 14.3-80 Hz

MOTOR MICROCONTROLLER

Pornsak Laonoonchai.

Pipat Laohasongkram Advisor

1987

Abstract

This paper describes a result of a design and development of power transistor for 3-phase induction motor control. The inverter is controlled by 8 bit microprocessor, Intel MCS 8048 using pulse width modulation method. With readily available transistor type 4 DH 07 as power switching device, the inverter is rated at 55 volt. Test result indicate that the inverter is capable of driving a 3-phase induction at rated voltage in condition of no-load by using a 3-phase transformer to step up voltage for rated voltage of a 3-phase induction. The inverter can variable in a range of speed 420 to 2400 rpm or in a range of frequency 14.3 to 80 Hz.

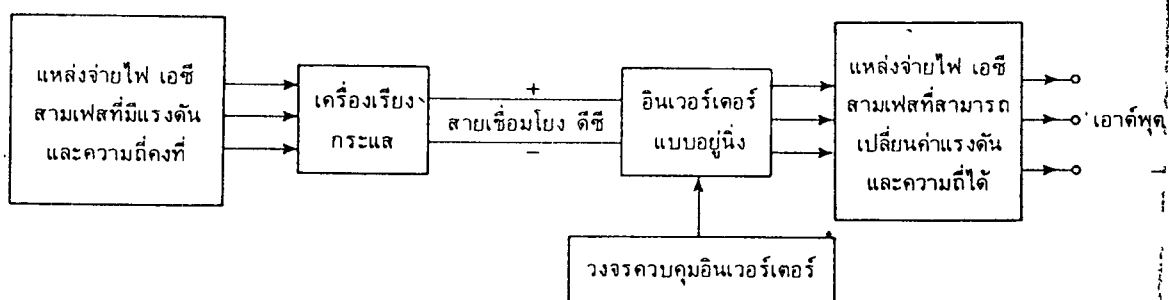
บทนำ

เนื่องจากในปัจจุบันนี้เราสามารถสร้างมอเตอร์เอซีให้มีกำลังมาสูงๆได้ จึงเป็นที่นิยมใช้ในโรงงานอุตสาหกรรมขนาดใหญ่ ซึ่งมอเตอร์ที่เรานิยมใช้กันมากเป็นมอเตอร์เหนี่ยวนำแบบกรงกระรอก ซึ่งมีวิธีสตาร์ทและการควบคุมอัตราเร็วหลายวิธีตามที่ได้เคยศึกษามา ทั้งการสตาร์ทและการควบคุมอัตราเร็วเราสามารถใช้อิเล็กทรอนิกส์เข้ามาควบคุมได้ โดยอาศัยการปรับค่าแรงดันเฉลี่ยและความถี่ที่ป้อนเข้ามอเตอร์จากแหล่งจ่ายไฟดิซีหรือเอซีที่มีแรงดันและความถี่คงที่

วิธีการสตาร์ทและการควบคุมอัตราเร็วส่วนใหญ่ใช้หลักการของเครื่องเปลี่ยนความถี่ ซึ่งเราได้เคยใช้เครื่องเปลี่ยนความถี่แบบหมุน (rotating frequency converters) โดยเป็นชุดของเครื่องกำเนิดไฟฟ้ากระแสสลับกับมอเตอร์เอซี ที่สามารถเปลี่ยนอัตราเร็วเป็นเครื่องฉุดให้หมุน ซึ่งเราสามารถเปลี่ยนค่าแรงดันได้ด้วยการเปลี่ยนค่าฟิลต์ของเครื่องกำเนิดไฟฟ้า และสามารถปรับค่าความถี่ได้ด้วยการเปลี่ยนอัตราเร็วของมอเตอร์ แต่ชุดของมอเตอร์และเครื่องกำเนิดไฟฟ้าจะมีขนาดใหญ่มาก และมีราคาแพงจึงไม่นิยมใช้ ในปัจจุบันเราได้หันมาใช้เครื่องเปลี่ยนความถี่แบบสถิต (static frequency converters) ซึ่งประดิษฐ์ขึ้นจากทรานซิสเตอร์เป็นชิ้นส่วนสำคัญของวงจร

เครื่องเปลี่ยนความถี่แบบสถิตชนิดพื้นฐานมี 2 ชนิด คือ ไซโคลคอนเวอร์เตอร์ และคอนเวอร์เตอร์เชื่อมโยงด้วยดิซี (DC link converters)

สำหรับโครงการนี้เครื่องเปลี่ยนความถี่สถิตที่ใช้จะเป็นเครื่องเปลี่ยนความถี่แบบคอนเวอร์เตอร์เชื่อมโยงด้วยดิซีมีหลักการเปลี่ยนกำลังไฟฟ้า 2 ครั้ง ซึ่งจะเปลี่ยนแหล่งจ่ายไฟเอซีให้เป็นไฟฟ้าดิซีก่อนด้วยวงจรเรียงกระแสทั่วๆไป แล้วเปลี่ยนกลับเป็นไฟฟ้าเอซีที่สามารถเปลี่ยนค่าแรงดันเฉลี่ยและความถี่ได้ด้วยเครื่องอินเวอร์เตอร์แบบอยู่หนึ่ง ซึ่งวิธีนี้จะแตกต่างกับวิธีของไซโคลคอนเวอร์เตอร์ คือวิธีของไซโคลคอนเวอร์เตอร์ไม่มีการเชื่อมโยงด้วยดิซี แต่เป็นการเปลี่ยนไฟฟ้าเอซีเป็นไฟฟ้าเอซีโดยตรง เรายินยมใช้เครื่องคอนเวอร์เตอร์แบบนี้ในการหมุนมอเตอร์หลายตัวพร้อมๆ กัน และสามารถปรับให้มีอัตราส่วนของแรงดันต่อความถี่ให้คงที่ได้



เครื่องเปลี่ยนความถี่แบบอยู่หนึ่งมีข้อดีเหนือกว่า เครื่องเปลี่ยนความถี่แบบหมุนหลายอย่าง ถึงแม้ราคาในการผลิตเริ่มต้นอาจสูง แต่ในอนาคตข้างหน้าอาจมีราคาถูกลงก็ได้ คือ -

1. มีราคาในการติดตั้งต่ำเนื่องจากไม่ต้องการฐานราก และวิธีการติดตั้งเครื่องที่อยู่ ยาก ต้องการที่ติดตั้งเล็กและส่งเสียงออกมารบกวนเล็กน้อย
2. ต้องการบำรุงรักษาบ่อย และถ้าเกิดเสียขึ้นมาจะใช้เวลาซ่อมบ่อย
3. มีค่าใช้จ่ายในระหว่างเดินเครื่องน้อยเนื่องจากมีประสิทธิภาพสูง(ประมาณ 85 ถึง 95%) และเนื่องจากไม่มีชิ้นส่วนใดเคลื่อนไหวจึงมีแต่การเสื่อมตามอายุใช้งานซึ่งจะต้องเปลี่ยนชิ้น ส่วนใหม่ในเวลาที่นานมาก
4. เนื่องจากเราสามารถควบคุมเครื่องเปลี่ยนความถี่แบบอยู่หนึ่งได้ง่ายจึงสามารถปรับ ค่าแรงดันและความถี่เอาท์พุทได้โดยมีพิสัยกว้างและปรับได้อย่างอิสระ สามารถใช้วิธีการป้อนกลับ ได้ จึงมีความแม่นยำสูงมีเสถียรภาพที่ดีและทำงานไว้วางใจได้
5. สามารถปรับแรงดันเอาท์พุทต่อความถี่ ได้เหมาะกับการเปลี่ยนอัตราเร็วของมอ เตอร์และสามารถปรับให้มีแรงหมุนต่อนสตาร์ทมีค่าสูงได้
6. สามารถกลับทิศการหมุนของมอเตอร์ได้ง่ายๆ ด้วยการควบคุมกลับจังหวะการบัง คับให้ไทรซิสเตอร์นำกระแสในวงจรอินเวอร์เตอร์แบบอยู่หนึ่งหรือในวงจรไซโคลคอนเวอร์เตอร์โดย ไม่ต้องสลับสายไฟจากแหล่งจ่ายไฟ

จากแผนภูมิในหน้า 1 สำหรับโครงการนี้ส่วนของเครื่องเรียงกระแส เราจะใช้เป็นวงจร บริดจ์ (bridge rectifier) ส่วนอินเวอร์เตอร์แบบอยู่หนึ่งนั้นจะเป็นวงจรอินเวอร์เตอร์สาม เฟสแบบสะพานเต็ม (3 phase full wave bridge inverter) โดยใช้พาวเวอร์ทรานซิส เตอร์เป็นอุปกรณ์สวิตชิง ส่วนวงจรควบคุมอินเวอร์เตอร์นั้นจะเป็นวงจรของไมโครโปรเซสเซอร์ MCS 8048 ของ Intel ซึ่งจะกล่าวถึงในบทต่อไป

บทที่ 1.

ทฤษฎีและหลักการควบคุมมอเตอร์เหนี่ยวนำ

มอเตอร์เหนี่ยวนำที่ได้รับความนิยมใช้มากในอุตสาหกรรมจะเป็นมอเตอร์เอชไอชนิดเหนี่ยวนำแบบกรงกระรอก (squirrel-cage induction motor) และมอเตอร์ที่ได้รับความนิยมรองลงไปคือมอเตอร์ชนิดรีลัคแตนซ์ (reluctance motor) มอเตอร์ทั้งสองมีลักษณะร่วมกันอย่างหนึ่งที่สามารถปรับเปลี่ยนอัตราเร็วได้ด้วยการเปลี่ยนความถี่ของแหล่งจ่ายไฟเอชไอซึ่งเมื่อเรากล่าวถึงมอเตอร์เหนี่ยวนำในโครงการนี้จะหมายความถึงมอเตอร์เหนี่ยวนำแบบกรงกระรอก

1.1 ทฤษฎีการควบคุมมอเตอร์เหนี่ยวนำ

มอเตอร์เหนี่ยวนำมีส่วนอยู่หนึ่งและส่วนหมุนคล้ายมอเตอร์ทั่วๆไป โดยส่วนอยู่หนึ่งมีขดลวดพันไว้และจัดให้เป็นแบบสองเฟสหรือสามเฟส ตามแต่จะออกแบบไว้ซึ่งไม่ต้องการวางแหวนสลิปหรือคอมมิวเตเตอร์ซึ่งสามารถสร้างได้ง่าย มีโครงสร้างแข็งแรงทนทานต่อการใช้งานและราคาถูก ส่วนหมุนอาจเป็นแบบใช้ลวดพันหรือแบบกรงกระรอก ถ้าหากมอเตอร์เป็นแบบสองเฟสส่วนหมุนมักจะเป็นแบบกรงกระรอก ส่วนหมุนทำหน้าที่รับพลังงานจากแหล่งจ่ายไฟฟ้าเอชไอผ่านส่วนอยู่หนึ่งโดยอาศัยหลักการเหนี่ยวนำ ซึ่งขดลวดของส่วนอยู่หนึ่งทำหน้าที่สร้างสนามแม่เหล็กที่มีค่าคงที่ขึ้นในช่องอากาศระหว่างส่วนอยู่หนึ่งกับส่วนหมุน และหมุนด้วยอัตราเร็วเชิงโคโรนัล

$$\begin{aligned} \omega_{\text{u}} &= 120f/P && \text{รอบต่อนาที} \\ &= 4\pi f/P && \text{เรเดียนต่อนาที} \end{aligned} \quad (1.1.1)$$

เมื่อ ω_{u} เป็นอัตราเร็วเชิงโคโรนัลของสนามแม่เหล็ก

f เป็นค่าความถี่ของแหล่งจ่ายไฟเอชไอ

P เป็นจำนวนโพลที่เกิดขึ้นจากการพันขดลวดบนส่วนอยู่หนึ่ง

โดยการพิจารณาที่ (1.1.1) เราสามารถเปลี่ยนอัตราเร็วของมอเตอร์เอชไอได้ด้วยการเปลี่ยนค่าความถี่ของแหล่งจ่ายไฟฟ้าเอชไอและการเปลี่ยนจำนวนโพล

ในการควบคุมอัตราเร็วของมอเตอร์เอชไอด้วยการเปลี่ยนค่าความถี่จะเป็นผลทำให้สนามแม่เหล็กหรือฟลักซ์ (flux) แม่เหล็กเปลี่ยนค่าไปตามสมการ

$$V = k \phi f$$

เมื่อ V : เป็นค่าแรงดันอาร์เอ็มบีอนเข้าที่ขั้วของมอเตอร์

K : เป็นค่าคงที่ซึ่งมีค่าขึ้นอยู่กับตัวประกอบรูปแบบ (form factor) ตัวประกอบ

ขดลวด (winding factor) และจำนวนรอบของขดลวดที่พันไว้ในมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการคำนวณเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

๐ : เป็นค่าฟลักซ์สูงสุดต่อหนึ่ง โพลของมอเตอร์

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามปกติมอเตอร์จะถูกออกแบบให้ฟลักซ์มีค่าอยู่ตรงส่วนเข้าโค้งของเส้นโค้งลักษณะสมบัติของการทำให้เป็นแม่เหล็ก (magnetization curve) ซึ่งถ้าเราลดความถี่ลง ฟลักซ์จะมีค่าสูงขึ้นและเข้าไปอยู่ในอาณาบริเวณอ้อมตัวทำให้มีกำลังสูญเสียในเหล็ก (iron losses) มีค่ามากขึ้น ทำให้มอเตอร์กินกระแสเพิ่มมากขึ้น ทำให้มีกำลังสูญเสียในทองแดง (copper losses) เพิ่มค่ามากขึ้น เป็นผลทำให้ประสิทธิภาพของมอเตอร์ลดต่ำลง ดังนั้นเราต้องลดค่าแรงดันให้เป็นสัดส่วนกับความถี่ที่ลดลง เพื่อให้ฟลักซ์ยังคงมีค่าอยู่ตรงส่วนเข้าโค้งของเส้นโค้งของการทำให้เป็นแม่เหล็ก กลับกันกับข้อความที่กล่าวข้างบน ถ้าเราเพิ่มความถี่ให้สูงขึ้น จะทำให้ฟลักซ์และแรงหมุนมีค่าคงเดิม ซึ่งข้อความนี้เท่ากับมีความหมายว่า เพื่อรักษาให้ฟลักซ์มีค่าคงที่เราต้องปรับค่าแรงดันให้เป็นสัดส่วนโดยตรงกับความถี่ หรือรักษาอัตราส่วนแรงดันต่อความถี่ให้คงที่วิธีการแบบนี้เรียกว่า "วิธีปฏิบัติการแรงดันต่อ Hz คงที่" หรือ "วิธีโวลต์ต่อเฮิรตซ์คงที่"

โดยการอินทิเกรตกฎของ Faraday

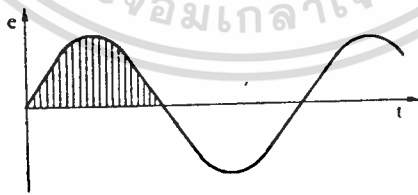
$$e = -d\phi / dt \quad (1.1.3)$$

เราจะได้

$$|\phi| = \int e dt \quad (1.1.4)$$

สมการนี้แสดงว่า พื้นที่ใต้เส้นโค้งรูปคลื่นแรงดันที่ได้เขียนเส้นแรงเงาไว้ดังแสดงในรูปที่

1.1.1 เป็นค่าฟลักซ์แม่เหล็กรวม ทำให้เราสรุปได้ว่า "ไม่ว่าความถี่จะมีค่าเท่าไรขอให้เพียงแต่พื้นที่ใต้เส้นโค้งแรงดันไฟฟ้ามีค่าคงที่อยู่เสมอแล้วมอเตอร์เหนี่ยวนำจะสามารถทำงานที่แรงหมุนสูงเหมาะสมที่สุด" ซึ่งเท่ากับมีความหมายว่า "โวลต์ต่อเฮิรตซ์มีค่าคงที่"



รูปที่ 1.1.1 แรงดันเพียงวงจรรูปคลื่นไซน์

ในกรณีที่ความต้านทานของขดลวดของมอเตอร์มีค่าน้อยมากสามารถตัดทิ้งได้ คักย์ตกคร่อมความต้านทานของขดลวดจะมีค่าเท่ากับศูนย์โดยประมาณ แรงดันเอซีที่ป้อนเข้ามอเตอร์จะมีค่าเท่ากับแรงเคลื่อนไฟฟ้าที่เกิดขึ้นในมอเตอร์โดยการเหนี่ยวนำ แต่ในกรณีที่ความต้านทานของขดลวดมีค่าใหญ่กว่าศูนย์ และถ้าความถี่มีค่าลดลงด้วย คักย์ตกคร่อมขดลวดจะมีขนาดใหญ่ขึ้น ซึ่งเป็นสาเหตุไปทำให้ฟลักซ์ในช่องอากาศมีค่าลดลงและเป็นผลทำให้มอเตอร์มีแรงหมุนลดลงด้วย ดังเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา -4- ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

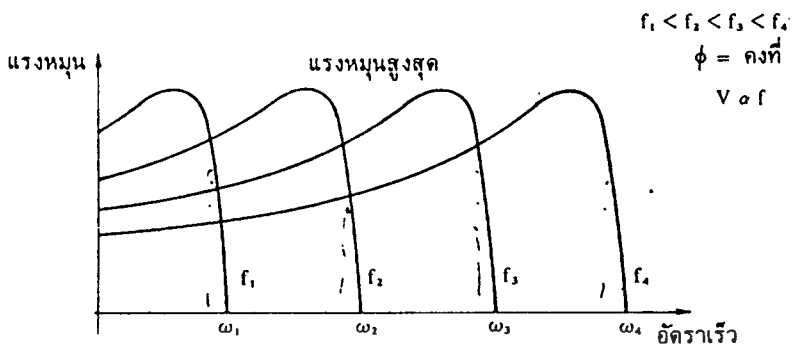
นั้นในปฏิบัติสำหรับความถี่ที่ต่ำกว่า 20 Hz โดยประมาณแล้วเราต้องรักษาแรงหมุนที่อัตราเร็วต่ำไว้ให้ดีด้วยการเพิ่มค่าโวลต์ต่อเฮิรตซ์" ให้มีค่าสูงกว่าปกติ

ถ้าเราป้อนกระแสไฟฟ้าเข้ามอเตอร์เหนี่ยวนำแบบกรงกระรอกขณะอยู่หนึ่ง โดยทั่วๆไปจะมีกระแสไหลเข้ามอเตอร์ประมาณ 5 หรือ 6 เท่าของอัตรากระแสปกติแต่แรงหมุนตอนเริ่มสตาร์ทมีค่าน้อยเนื่องจากส่วนหมุนมีตัวประกอบกำลัง (power factor) มีค่าต่ำที่ความถี่ของส่วนหมุนมีค่าสูง ด้วยเหตุนี้ในระบบที่สามารถเปลี่ยนความถี่ได้ขณะสตาร์ทมอเตอร์เหนี่ยวนำเราจึงต้องลดความถี่ลง เพื่อให้ทำให้ตัวประกอบกำลังของส่วนหมุนมีค่าสูงขึ้นและเป็นการเพิ่มค่าแรงหมุนขณะสตาร์ทต่อแอมแปร์ให้สูงขึ้นโดยการปรับค่าปลั๊กซ์ให้สูงขึ้นด้วยวิธีการ "โวลต์ต่อเฮิรตซ์คงที่" หลักการเช่นนี้ ทำให้เราสามารถปรับค่าแรงหมุนขณะมอเตอร์อยู่หนึ่งให้มีค่าใหญ่กว่าอัตราแรงหมุนปกติหลายเท่า และสามารถควบคุมมอเตอร์ให้หมุนเร็วขึ้นได้อย่างรวดเร็วด้วยการเพิ่มความถี่ของแหล่งจ่ายไฟฟ้าโดยวิธีการแบบนี้ เราสามารถหลีกเลี่ยงอันตรายที่เกิดขึ้นในมอเตอร์จากการสตาร์ทมอเตอร์ด้วยแหล่งจ่ายไฟฟ้าเอชไอความถี่คงที่ค่าต่ำซึ่งมอเตอร์จะหมุนขึ้นอย่างช้าๆ และกินกระแสไฟฟ้ามาก

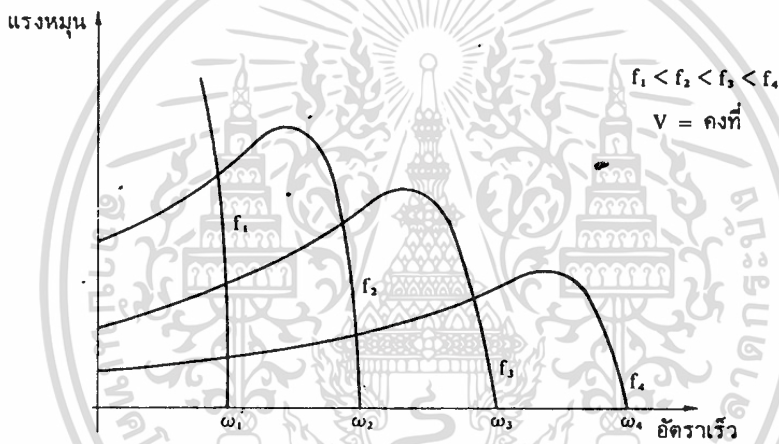
รูปที่ 1.1.2 (ก) แสดงลักษณะสมบัติของแรงหมุนของมอเตอร์ที่ความถี่ต่างๆ โดยรักษาแรงหมุนนิ่งทลายให้คงที่ด้วยการปรับค่า "โวลต์ต่อเฮิรตซ์" ให้คงที่ ลักษณะสมบัติเช่นนี้ทำให้มอเตอร์เหนี่ยวนำเหมาะสมกับงานขับเคลื่อนภาระที่มีแรงหมุนคงที่ให้มีอัตราเร็วต่าง ๆ

ถ้าเรารักษาแรงดันในส่วนหมุนให้มีค่าคงที่ขณะที่มีการเปลี่ยนค่าความถี่ ปลั๊กซ์ในช่องอากาศและแรงหมุนนิ่งทลายจะลดค่าลงเมื่อเราเพิ่มความถี่ให้สูงขึ้นดังแสดงในรูปที่ 1.1.2 (ข) ลักษณะสมบัติเช่นนี้ทำให้มอเตอร์เหนี่ยวนำเหมาะสมกับการใช้งานจุดลาก ซึ่งงานนี้ต้องการแรงหมุนสูงขณะสตาร์ทหรือขณะอัตราเร็วต่ำและต้องการแรงหมุนค่าน้อยที่อัตราเร็วสูง

ขอให้เราสังเกตว่าจากรูปที่ 1.1.2 ในการขับเคลื่อนมอเตอร์เหนี่ยวนำเราควรใช้ความถี่ต่ำในตอนสตาร์ทเพื่อให้มอเตอร์มีแรงหมุนค่าสูงและกินกระแสต่ำ ต่อไปจึงค่อย ๆ เพิ่มความถี่ขึ้นทีละน้อยเพื่อให้มอเตอร์เพิ่มอัตราเร็วสูงขึ้น และเมื่อมอเตอร์ได้หมุนเข้าสู่อัตราเร็วปกติแล้วก็ให้เรารักษาความถี่ไว้ให้คงที่ นอกจากนี้แรงหมุนตอนสตาร์ทมีค่าขึ้นอยู่กับค่าความต้านทานของส่วนหมุน แต่ค่าความต้านทานค่าใหญ่ของส่วนหมุนจะทำให้มีกำลังสูญเสียใหญ่มากขึ้นเป็นเหตุให้มอเตอร์มีประสิทธิภาพต่ำขณะทำงานตามปกติ ถ้าความต้านทานของส่วนหมุนมีค่าต่ำจะทำให้แรงหมุนตอนสตาร์ทต่ำแต่อาจทำให้สูงขึ้นได้ด้วยการใช้ความถี่ต่ำลง ซึ่งในขณะที่ทำงานตามปกติมอเตอร์จะมีประสิทธิภาพดีขึ้นและช่วยลดกำลังสูญเสียอาร์โมนิคต่าง ๆ ให้น้อยลงเมื่อแหล่งจ่ายไฟฟ้ามีรูปคลื่นแรงดันไม่ใช่คลื่นไซน์



(ก)



(ข)

รูปที่ 1.1.2 เส้นโค้งลักษณะสมบัตินี้ระหว่างแรงหมุนกับอัตราเร็วของมอเตอร์เหนี่ยวนำที่ความถี่ต่างของแหล่งจ่ายไฟ

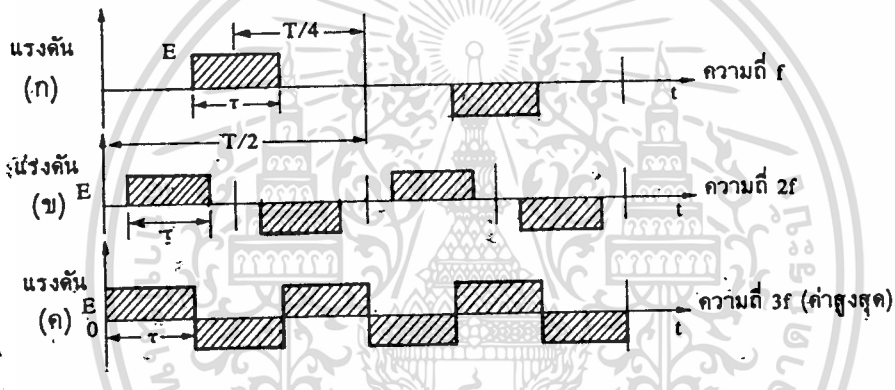
(ก) เมื่อพหุคูณในช่องอากาศมีค่าคงที่

(ข) เมื่อแรงดันที่ป้อนเข้ามอเตอร์มีค่าคงที่

โดยทั่วไป ถ้าให้มอเตอร์เหนี่ยวนำทำงานโดยมีความถี่สูงในส่วนหมุนที่ไม่เกินความถี่ขดลวด มอเตอร์น่าจะทำงานด้วยประสิทธิภาพสูง แต่ถ้าให้มอเตอร์ทำงานนั้นเกินจากจุดขดลวดขดลวดมอเตอร์จะกินกระแสตัวประกอบกำลังมีค่าต่ำ และแรงหมุนต่อแอมแปร์มีค่าต่ำ เป็นผลให้ประสิทธิภาพต่ำมาก ซึ่งเราอาจแก้ไขโดยใช้มอเตอร์เหนี่ยวนำทำงานด้วยระบบป้อนกลับวงปิด โดยวิธีนี้เราสามารถควบคุมความถี่ของส่วนหมุนให้มีค่าต่ำกว่าความถี่ขดลวดอยู่เสมอได้ เป็นผลให้มอเตอร์เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา -6- ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหนี่ยวนำที่ทำงานแบบนี้มีลักษณะสมบัติของการควบคุมอัตราเร็ว และผลตอบสนองชั่วคราวมีคุณสมบัติดีกว่าลักษณะสมบัติของมอเตอร์ดีซีที่ทำงานแบบระบบ Ward-Leonard หรือที่ป้อนด้วยแรงดันจากคอนเวอร์เตอร์อิเล็กทรอนิกส์

1.2 การมอดูเลตตามความกว้างของพัลส์ วิธีการมอดูเลตตามความกว้างของพัลส์ (pulse width modulation=PWM) อาศัยหลักการจากสมการที่ (1.14) ที่ว่า "พื้นที่ใต้เส้นโค้งแรงดันในแต่ละครึ่งวัฏจักรมีค่าคงที่ จะทำให้พัลส์ของมอเตอร์มีค่าคงที่" ซึ่งเราอาจทำขึ้นได้ด้วยการให้อินเวอร์เตอร์สร้างรูปคลื่นสี่เหลี่ยมขึ้นมาจากระดับดีซีเลี้ยงวงจรให้มีความสูง และความกว้างคงที่ และอาศัยหลักการสำคัญที่ว่า วงจรอิเล็กทรอนิกส์ที่ควบคุมสามารถทำการไบแอสพาวเวอร์ทรานซิสเตอร์



รูปที่ 1.2.1 การปรับค่าพัลส์ให้คงที่โดยใช้มอดูเลตตามความกว้างของพัลส์

รูปที่ 1.2.1 แสดงการมอดูเลตตามความกว้างของพัลส์สำหรับความถี่แตกต่างกันสามกรณี ในแต่ละกรณีรูปคลื่นแรงดันสี่เหลี่ยมที่มีความสูง E และความกว้างคงที่ในครึ่งวัฏจักร ซึ่งในแต่ละกรณีจะให้พัลส์มีค่าคงที่ คือ

$$\int_0^{T/k} E dt = E\tau = \text{คงที่} \quad (1.2.1)$$

เมื่อ k เป็นค่าคงที่ในการจัดความกว้างของพัลส์

τ เป็นความกว้างของพัลส์

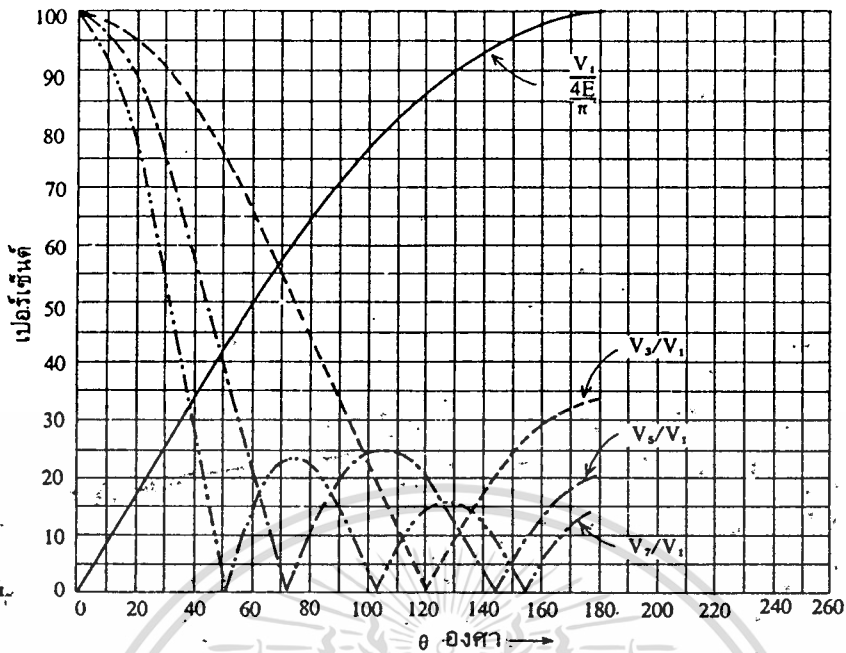
T เป็นเวลาหนึ่งคาบของความถี่สูงสุดที่อาจจัดได้ f_n ซึ่งมีความสัมพันธ์

$$f_n = 1/T = 1/2\tau \quad (1.2.2)$$

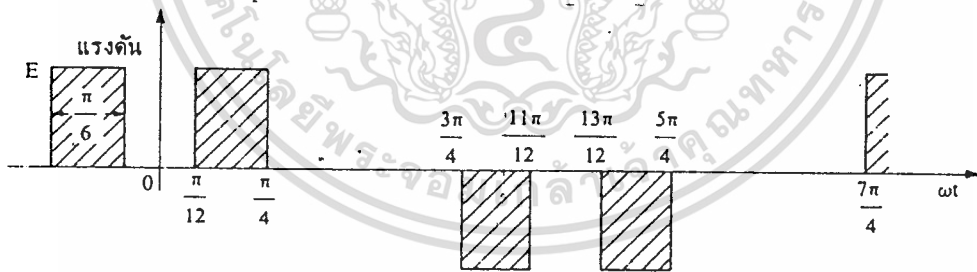
โดยความถี่สูงสุดมีค่าขึ้นอยู่กับความกว้างของพัลส์ ซึ่งอาจจัดได้ดังแสดงในรูปที่ 1.2.1

เพื่อความสะดวกขอให้เราตั้งแกนตั้งให้ผ่านกึ่งกลางของรูปพัลส์ค่าบวก ดังแสดงในรูปที่

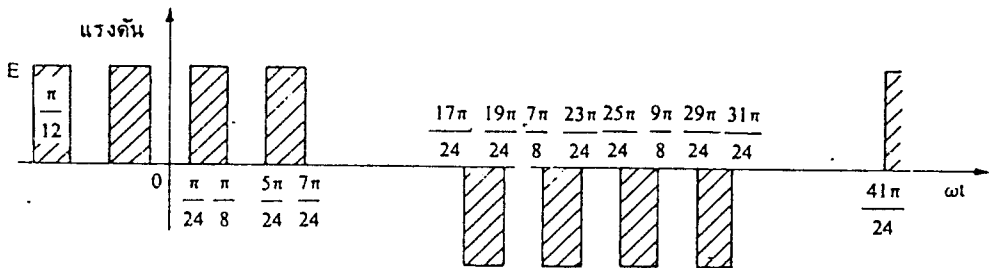
1.2.2 ทำให้พัลส์นี้ฟังก์ชันคู่ซึ่งเขียนเป็นสมการได้ว่า



ลบ จะให้แรงหมุนโต้กลับเป็นผลให้มอเตอร์มีลักษณะสมยัตติของแรงหมุนเลวลง เราจึงจำเป็นต้อง
 ขจัดออกไป พิจารณารูปคลื่นแรงดันรูปสี่เหลี่ยมรูปเตี้ยหรือพัลส์เตี้ยดังแสดงในรูปที่ 1.2.2 และ
 สมการที่ 1.2.8 เราจะเห็นว่า รูปคลื่นแรงดันพัลส์เตี้ยนี้มีอาร์โมนิคลำดับที่สามประกอบอยู่ด้วย
 โดยจะมีขนาดใหญ่มากเมื่อมีค่าน้อยและจะมีขนาดเท่ากับศูนย์เมื่อ $\omega = 120$ แต่ถ้าเราเพิ่มจำนวน



(ก)



(ข)

รูปที่ 1.2.3 (ก) รูปคลื่นพัลส์สี่เหลี่ยมในครึ่งวัฏจักรถูกแบ่งออกเป็นสองพัลส์ หรือ พัลส์คู่

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์เพียงบางส่วนในครึ่งวัฏจักรถูกแบ่งออกเป็นพัลส์ย่อยสี่พัลส์โดยขึ้นด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้เพื่อการนำไปใช้

24636

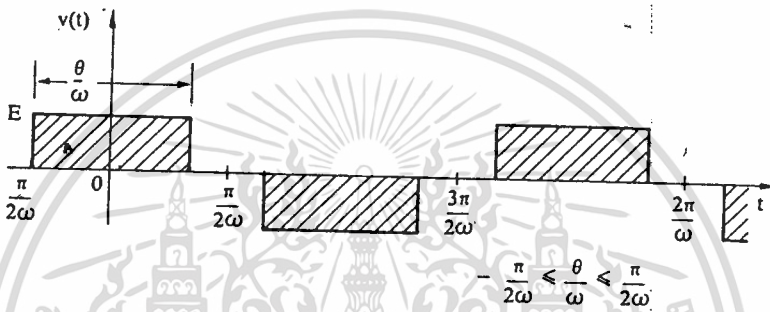
$$\begin{aligned}
 V(t) &= E \cdot \text{สำหรับ } -\frac{\theta}{2\omega} \leq t \leq \frac{\theta}{2\omega} \\
 &= 0 \text{ สำหรับที่อื่นในระยะเวลาครึ่งคาบ}
 \end{aligned}
 \tag{1.2.3}$$

เมื่อใช้อนุกรม Fourier ทำการวิเคราะห์ เราจะได้

$$b_n = 0 \tag{1.2.4}$$

เนื่องจากรูปคลื่นของฟังก์ชันนี้สมมาตรกันระหว่างข้างบนของแกนนอน ดังนั้นในการกระจายนี้ไม่มีองค์ประกอบดิซิปประกอบอยู่ด้วย คือ

$$a_0 = 0 \tag{1.2.5}$$



รูปที่ 1.2.2 รูปคลื่นแรงดันเอาต์พุทของอินเวอร์เตอร์

และเพราะว่าฟังก์ชันนี้มีรูปคลื่นสมมาตรกันระหว่างครึ่งวัฏจักรบวกกับครึ่งวัฏจักรลบ เพราะฉะนั้น ฟังก์ชันนี้จึงเป็นฟังก์ชันฮาร์โมนิกคี่ (odd harmonic frequency) คือ ฟังก์ชันนี้มีฮาร์โมนิกคี่เป็นคี่ประกอบอยู่เท่านั้น เพราะฉะนั้นเราจะได้

$$a_2 = a_4 = a_6 = \dots = 0 \tag{1.2.6}$$

ส่วนสัมประสิทธิ์ของฮาร์โมนิกคี่มีค่า

$$\begin{aligned}
 a_n &= 4E / (\pi / \omega) \int_0^{\theta/2\omega} \cos n\omega t \, dt \\
 &= 4E / (n\pi) \sin (n\theta/2) , n = 1, 3, 5, \dots
 \end{aligned}
 \tag{1.2.7}$$

เพราะฉะนั้นเราสามารถเขียนฟังก์ชันของแรงดันเอาต์พุตได้เป็น

$$V(t) = \sum_{n=1}^{\infty} \frac{4E}{n\pi} \left(\sin \frac{n\theta}{2} \right) (\cos n\omega t) \tag{1.2.8}$$

โดยค่าแรงดันเอาต์พุตที่ความถี่พื้นฐานมีค่าสูงสุดเกิดขึ้นเมื่อมีเงื่อนไข

$$\theta = \pi \tag{1.2.9}$$

ถ้าเราใช้ค่า $4E/\pi$ เป็นค่าหลักในการคิดค่าเปลี่ยนแปลงของฮาร์โมนิกตามค่ามุมของเทียบกับแรงดันที่ความถี่พื้นฐานในหน่วยเปอร์เซ็นต์ โดยเราสามารถเขียนเป็นเส้นโค้งของฮาร์โมนิกต่าง ๆ ได้ดังแสดงรูปที่ 1.2.3

ขอให้เราสังเกตว่า หลักการของอินเวอร์เตอร์ที่ใช้วิธีการที่ทำให้หลักมีค่าคงที่สำหรับครึ่งวัฏจักรเป็นหลักการที่ง่าย แต่เราต้องขจัดฮาร์โมนิกคี่ออก โดยฮาร์โมนิกคี่ที่ 5, 11 เป็นเฟสอันดับไม่เท่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พัลส์ในเครื่องวิทยุจักรให้ เป็นสองพัลส์ ดังแสดงในรูป 1.2.3 (ก) เราจะเขียนสมการของแรงดันสำหรับเศษหนึ่งส่วนสี่ของวัฏจักรได้ดังนี้

$$v(t) = \begin{cases} E & \text{สำหรับ } \pi/12 \leq \omega t \leq \pi/4 \\ 0 & \text{สำหรับ } 0 \leq \omega t \leq \pi/12 \text{ และ } \pi/4 \leq \omega t \leq \pi/2 \end{cases} \quad (1.2.8)$$

โดยคุณสมบัติที่รูปคลื่นนี้เป็นฟังก์ชันคู่และเป็นฟังก์ชันฮาร์โมนิกคี่ เราจะได้

$$a_0 = a_2 = a_4 = \dots = 0 \text{ และ } b_n = 0 \quad (1.2.9)$$

และเราจะคำนวณหาค่าสัมประสิทธิ์ที่เหลืออยู่ได้ดังนี้

$$\begin{aligned} a_n &= 4E/\pi \int_{\pi/12}^{\pi/4} \cos n\omega t \, d(\omega t) \\ &= 4E/n\pi [\sin n\pi/4 - \sin n\pi/12] ; n = 1, 3, 5, 7, \dots \end{aligned} \quad (1.2.10)$$

ดังนั้นเราจะได้ผลลัพธ์เป็น

$$V(t) = 0.571E(\cos \omega t - 0.746 \cos 5\omega t - 0.533 \cos 7\omega t + 0.091 \cos 11\omega t + \dots) \quad (1.2.11)$$

ในการทำงานเดียวกันกับรูปที่ 1.2.3 (ก) เราจะเขียนสมการของแรงดันสำหรับกรณีที่เราเพิ่มจำนวนพัลส์ในเครื่องวิทยุจักรให้เป็นสี่พัลส์ ดังแสดงในรูปที่ 1.2.3 (ข) ได้ดังนี้

$$V(t) = \begin{cases} E & \text{สำหรับ } \pi/24 \leq \omega t \leq \pi/8 \text{ และ } 5\pi/24 \leq \omega t \leq 7\pi/24 \\ 0 & \text{สำหรับ } 0 \leq \omega t < \pi/24 \text{ และ } 3\pi/24 < \omega t < 5\pi/24 \\ & \text{และ } 7\pi/24 < \omega t < \pi/2 \end{cases} \quad (1.2.12)$$

ด้วยเหตุผลเช่นเดียวกับข้างบน เราจะได้

$$a_0 = a_2 = a_4 = \dots = 0 \text{ และ } b_n = 0 \quad (1.2.13)$$

และเราจะคำนวณหาค่าสัมประสิทธิ์ที่เหลืออยู่ได้ดังนี้

$$\begin{aligned} a_n &= 4E/\pi \left[\int_{\pi/24}^{\pi/8} \cos n\omega t \, d(\omega t) + \int_{5\pi/24}^{7\pi/24} \cos n\omega t \, d(\omega t) \right] \\ &= 4E/n\pi [\sin 3n\pi/24 - \sin n\pi/24 + \sin 7n\pi/24 - \sin 5n\pi/24] \end{aligned} \quad (1.2.14)$$

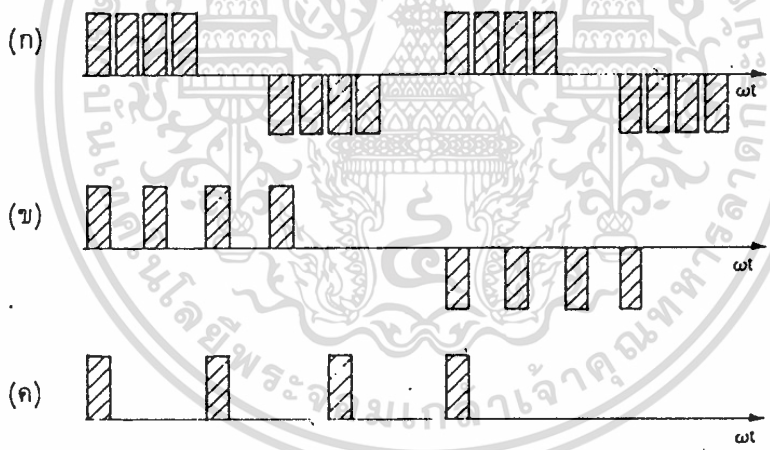
ดังนั้นเราจะได้ผลลัพธ์เป็น

$$V(t) = 0.556E(\cos \omega t - 0.249 \cos 5\omega t + 0.129 \cos 7\omega t - 0.69 \cos 11\omega t + \dots) \quad (1.2.15)$$

โดยการสังเกตสมการที่ 1.2.11 กับ 1.2.15 เปรียบเทียบกัน เราจะเห็นว่า ไม่มีฮาร์โมนิคลำดับที่สามอยู่ในสมการทั้งสอง ฮาร์โมนิคลำดับที่ห้าและที่เจ็ด (ซึ่งเป็นฮาร์โมนิคที่มีผลกระทบที่สำคัญ) ไม่ถูกรณณาใดๆทั้งสิ้น อีกทั้งห่ามมีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อลักษณะสมบัติของมอเตอร์) ยังคงมีอยู่ในสมการทั้งสองแต่มีขนาดลดค่าลงเมื่อเพิ่มจำนวนพัลส์ต่อครั้งวัฏจักรให้มากขึ้นซึ่งจะมีค่าลดลงเข้าสู่ค่าของรูปคลื่นบันไดหกขั้น (สมการที่ 1.2.10) แต่เราไม่สามารถลดค่าลงให้ต่ำกว่าค่าของรูปคลื่นบันไดหกขั้นได้ถึงแม้ว่าเราจะได้เพิ่มจำนวนพัลส์ต่อครั้งวัฏจักรให้สูงขึ้นต่อไปอีก

ข้อได้เปรียบของวิธีการนี้ คือ สามารถปรับค่า "โวลต์ต่อเฮิรตซ์" ให้คงที่ได้ จึงสามารถนำไปควบคุมอัตราเร็วของมอเตอร์เอชไอได้โดยหลักการที่ว่า "โดยการรักษาพื้นที่ (ในหน่วยโวลต์-วินาที) ภายในเครื่องวัฏจักรของรูปคลื่นแรงดันเอาท์พุทให้เท่ากันเสมอที่ทุก ๆ ความถี่" ซึ่งในการปรับอัตราเร็วของมอเตอร์ให้ลดต่ำลงโดยลดความถี่ลง เราจะได้ทำได้ด้วยการรักษาความกว้างของพัลส์ให้คงที่แต่ขยายระยะห่างให้ห่างออกเป็นสัดส่วนกับความถี่ที่ลดลง ดังแสดงในรูปที่ 1.2.4 ก็จะทำให้เราได้ "โวลต์ต่อเฮิรตซ์" คงที่ ตามต้องการในวิธีการเช่นนี้ ความถี่ของพัลส์จะมีค่าลดลงเป็นสัดส่วนโดยตรงกับการลดค่าลงของความถี่พื้นฐานเอาท์พุท แต่ถ้าเราต้องการเพิ่มแรงดันให้สูงขึ้นด้วยเราจะได้ทำได้ด้วยการเพิ่มความกว้างของพัลส์ให้กว้างออก



รูปที่ 1.2.4 รูปคลื่นแรงเอาท์พุทของการปรับค่า "โวลต์ต่อเฮิรตซ์" ของอินเวอร์เตอร์

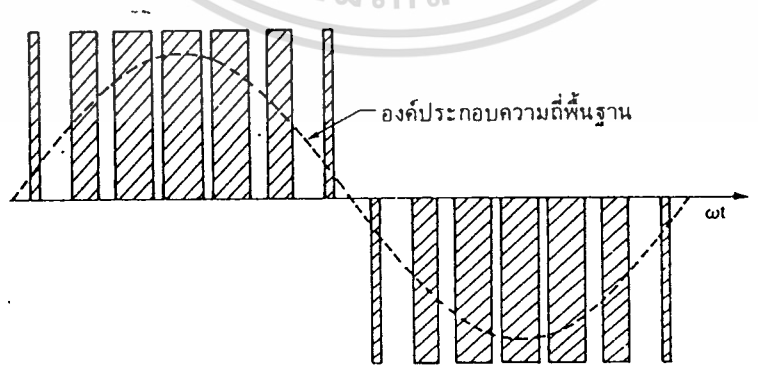
แบบที่ใช้วิธีการแบ่งความกว้างของรูปคลื่นออกเป็นสัดส่วนหลายพัลส์

- (ก) ที่ความถี่สูงสุด
- (ข) ที่ความถี่ลดลงครึ่งหนึ่ง
- (ค) ที่ความถี่ลดลงเหลือเพียงหนึ่งส่วนสี่

ข้อเสียเปรียบของการปรับแรงดันด้วยวิธีนี้ จะเกิดขึ้นที่ความต่ำ เนื่องจากในการที่ควบคุมให้ระยะห่างของพัลส์ห่างกันมากขึ้น จำทำให้รูปคลื่นแรงดันเอาท์พุทเสื่อมเสียมากขึ้น ค่าความถี่ของฮาร์โมนิคก็ลดลงตามค่าความถี่พื้นฐาน ค่ารีแอคแตนซ์ของมอเตอร์จะมีค่าน้อยลงไม่เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา -11-ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

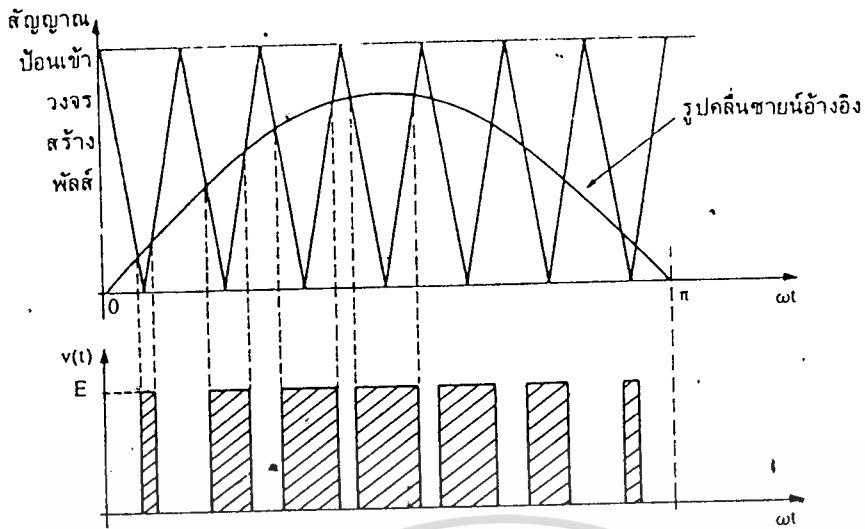
สามารถรองฮาร์โมนิคออกได้มากเหมือนเดิม แต่เราอาจปรับปรุงแก้ไขการทำงานของมอเตอร์ให้ดีขึ้นได้ด้วยการต่อรีแอ็คแตนซ์ภายนอกเข้าเป็นอนุกรมกับขดลวดของส่วนอยู่นิ่งของมอเตอร์ถึงแม้ว่าวิธีการเช่นนี้จะทำให้แรงหมุนที่ความถี่พื้นฐานเสียไปบ้าง แต่รีแอ็คแตนซ์ จะมีผลต่อต้านกับฮาร์โมนิคมากกว่าการต่อต้านที่ความถี่พื้นฐาน จึงเป็นผลทำให้กระแสติดเฟี้ยนและกำลังสูญเสียเนื่องจากฮาร์โมนิคมีค่าลดลงมากชดเชยคุ้มค่ากับการที่แรงหมุนเสียไปที่ความถี่พื้นฐาน นอกจากนี้การใช้หม้อแปลงออโตลด์ไฟลง (step-down autotransformer) เข้าต่อทางด้านเอาท์พุทของอินเวอร์เตอร์ก็จะสามารถช่วยลดความผิดเฟี้ยนที่รุนแรงของแรงดันขณะที่แรงดันเอาท์พุทมีค่าต่ำได้ด้วยการให้อินเวอร์เตอร์ทำงานที่แรงดันสูงเต็มที่ แล้วใช้หม้อแปลงออโตลด์ไฟลงให้ได้แรงดันเอาท์พุทมีค่าต่ำตามต้องการแล้วจะ ได้รูปคลื่นแรงดันเอาท์พุทที่ดีขึ้น โดยการใช้หม้อแปลงอิตผสมกับการต่อรีแอ็คเตอร์แบบอนุกรมจะช่วยปรับปรุงการทำงานของมอเตอร์เอชไอ (ที่ทำงานด้วยรูปคลื่นแบบแบ่งออกเป็นสองพัลส์หรือสี่พัลส์ต่อครึ่งวัฏจักร) ให้ดีขึ้น เกือบเท่ากับการทำงานด้วยรูปคลื่นขั้นบันไดหกขั้น

วิธีการแบ่งรูปคลื่นออกเป็นพัลส์หลายพัลส์ที่ดีกว่าข้างบน คือ การแบ่งความกว้างของพัลส์แต่ละพัลส์ให้แปรค่าในครึ่งวัฏจักรไปตามลักษณะคลื่นไซน์ หรือ พูตง่าย ๆ ก็คือ การแบ่งความกว้างของพัลส์แต่ละพัลส์ให้มีค่า เป็นสัดส่วนกับพื้นที่ใต้รูปคลื่นไซน์ที่มีตำแหน่งสมนัยกันในครึ่งวัฏจักร ดังแสดงในรูปที่ 1.2.4 วิธีการนี้จะคล้ายคลึงกับวิธีการที่ทำให้เกิดรูปคลื่นขั้นบันไดที่มีความสูงของแต่ละขั้นเป็นสัดส่วนกับพื้นที่ใต้รูปคลื่นไซน์ที่มีตำแหน่งสมนัยกัน ในวิธีการแบ่งรูปคลื่นออกเป็นพัลส์หลายพัลส์ให้มีความกว้าง เป็นสัดส่วนกับรูปคลื่นไซน์ จะให้ฮาร์โมนิคลำดับต่ำสุดมีความถี่ขึ้นอยู่กับค่าความถี่ของพัลส์ ถ้าความถี่ของฮาร์โมนิคลำดับต่ำสุดใหญ่กว่าความถี่พื้นฐานมาก เป็นผลให้ค่าเหนี่ยวนำของมอเตอร์ใหญ่พอ เพียงที่จะกรองเอาฮาร์โมนิคที่ปะปนออกได้

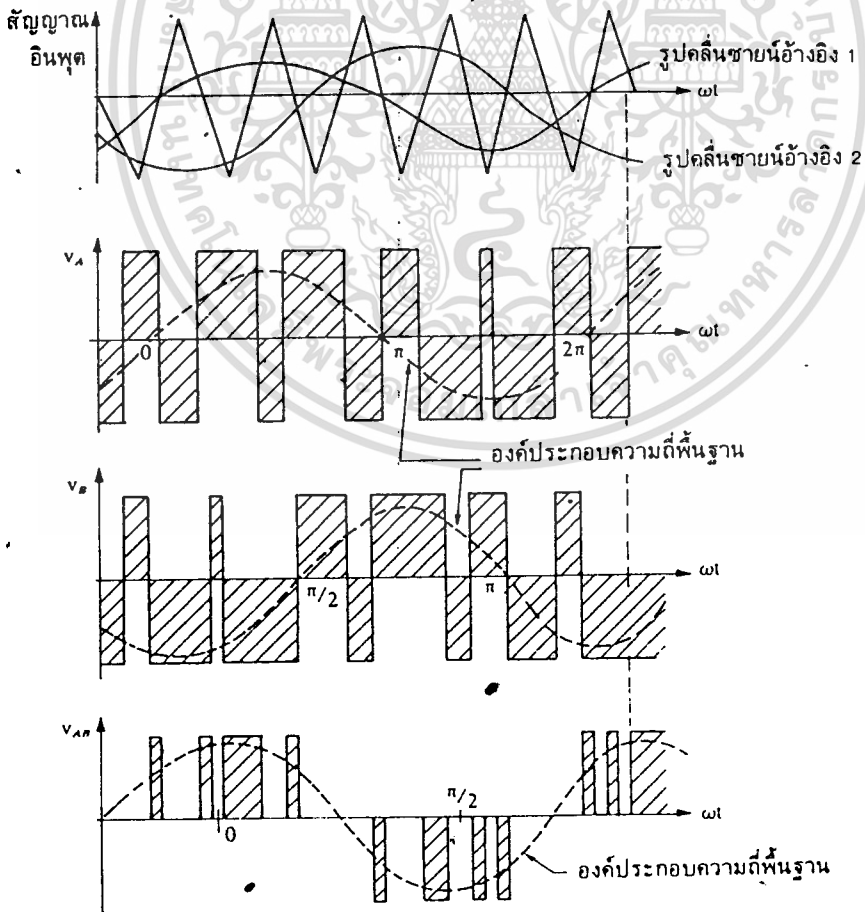


รูปที่ 1.2.4 รูปคลื่นแรงดันแรงเอาท์พุทของวิธีการแบ่งรูปคลื่นออกเป็นพัลส์หลายพัลส์

ใช้มีความกว้างเป็นสัดส่วนกับพื้นที่ภายใต้



รูปที่ 1.2.5 รูปคลื่นของวงจรควบคุมสำหรับอินเวอร์เตอร์แบบรูปคลื่นแรงดันเอาต์พุตถูกแบ่งออกเป็นพัลส์หลายพัลส์



รูปที่ 1.2.6 รูปคลื่นของวงจรควบคุมที่ใช้รูปคลื่นขาหนี้อ่างอิงสองรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา-13-ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

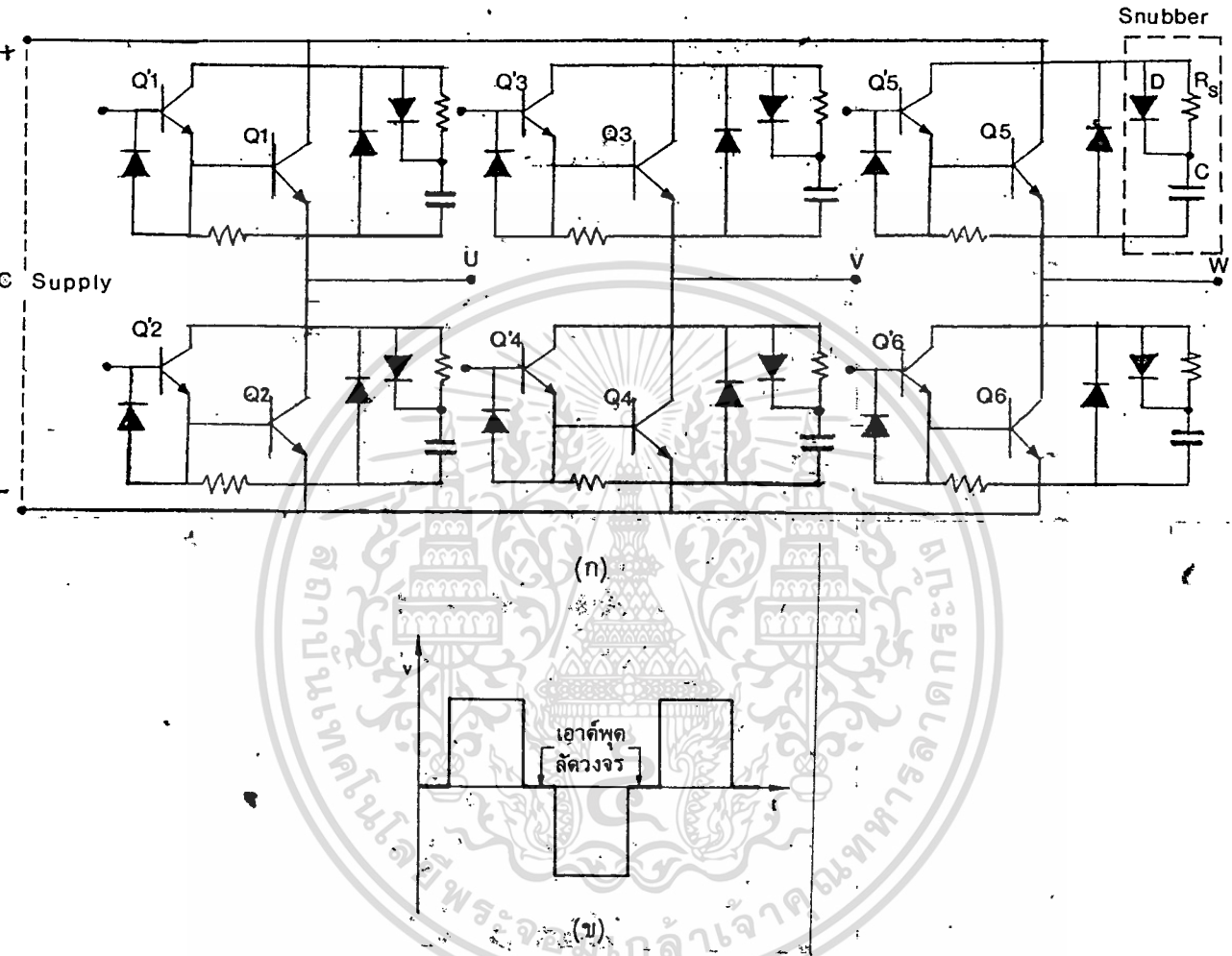
ในการที่เราจะได้รูปคลื่นดังแสดงในรูปที่ 1.2.4 เราต้องใช้วงจรควบคุมไบแอสพาวเวอร์ทรานซิสเตอร์ โดยอาศัยหลักการเปรียบเทียบรูปคลื่นสามเหลี่ยมความถี่สูงกับรูปคลื่นไซน์อ้างอิงดังแสดงในรูปที่ 1.2.5 โดยอาศัยไอซีที่ทำหน้าที่เป็นวงจรเปรียบเทียบ ในโครงการนี้จะใช้ไมโครโปรเซสเซอร์ MCS 8048 ของ Intel เป็นตัวสร้างสัญญาณมอดูเลตแทนการใช้ ไอซีทำหน้าที่เป็นวงจรเปรียบเทียบ เราจะได้สัญญาณจังหวะเวลาที่จะนำไปสร้างพัลส์เพื่อไบแอสพาวเวอร์ทรานซิสเตอร์ของวงจรอินเวอร์เตอร์ซึ่งจำทำให้ได้แรงดันเอาท์พุทที่มีรูปคลื่น ดังแสดงในรูปที่ 1.2.4 ตามต้องการ วิธีการสร้างพัลส์ได้ดีกว่าวิธีข้างบน คือ วิธีที่ใช้รูปคลื่นไซน์อ้างอิงสองคลื่นพร้อมกับการใช้การแปรค่ามูเฟสให้เคลื่อนไปดังอธิบายด้วยรูปที่ 1.2.6 แต่ทั้งสองวิธีจะต้องอยู่บนหลักการที่ว่า พื้นที่รวมทั้งหมดของพัลส์ต่อครึ่งวัฏจักร $(\int v(t) dt)$ จะต้องมีค่าคงที่เสมอ

อินเวอร์เตอร์แบบมอดูเลตตามความกว้างของพัลส์ ที่ใช้การแบ่งรูปคลื่นออกเป็นพัลส์หลายพัลส์จะต้องการขึ้นเพิ่มเติมน้อยขึ้นเช่น ไดโอดราคาถูกซึ่งใช้ในวงจรเรียงกระแสแบบสะพาน เป็นต้น สำหรับหม้อแปลงเอาท์พุทไม่จำเป็นต้องใช้นอกจากจะต้องการชจัดอาร์โมนิค โดยนำเอาอินเวอร์เตอร์หลายวงจรมาต่อผลมกันจึง จะต้องการหม้อแปลงเอาท์พุทที่ต้องการขึ้นส่วนอุปกรณ์เพิ่มเติมเล็กน้อยจะทำให้เกิดข้อเสียเปรียบที่ว่า ต้องการวงจรควบคุมที่ยุ่งยากในการไบแอสพาวเวอร์ทรานซิสเตอร์ของวงจรอินเวอร์เตอร์ แต่วงจรที่ยุ่งยากนี้สามารถทำให้ควบคุมอัตราเร็วของมอเตอร์ได้เป็นพัลส์กว้าง 20 ต่อ 1 ของอัตราเร็วเต็มที่และยังมีผลตอบสนองได้รวดเร็ว

ในการเลือกใช้ความถี่ของอินเวอร์เตอร์แบบมอดูเลตตามความกว้างของพัลส์นี้ เราต้องเลือกความถี่อย่างประนีประนอม โดยการใช้ความถี่ของพัลส์สูงทำให้ช่วยลดอาร์โมนิคลำดับต่ำให้มีค่าลดน้อยลงได้ เป็นผลให้เราสามารถออกแบบวงจรกรองคลื่นได้ง่าย แต่เราเลือกใช้เช่นนั้นจะทำให้อินเวอร์เตอร์มีกำลังสูง ในการปิดกระแสมีค่าเพิ่มขึ้นเป็นสัดส่วนโดยตรงกับความถี่ของพัลส์ เนื่องจากกำลังสูงของการปิดกระแสสำหรับอินเวอร์เตอร์มีค่าขึ้นอยู่กับจำนวนการปิดกระแสต่อวินาที ในการปฏิบัติการเลือกใช้ความถี่ของพัลส์ขึ้นอยู่กับการพิจารณาที่ว่า " การที่มีกำลังสูงเพิ่มขึ้นเนื่องจากการปิดกระแสของอินเวอร์เตอร์นั้นจะต้องสมดุลกับกำลังสูงลดลงในมอเตอร์เอซีเนื่องจากมีอาร์โมนิคลดลงในแรงดันป้อนเข้ามอเตอร์ "

1.3 วงจรอินเวอร์เตอร์สามเฟสแบบสะพานเต็ม ในวงจรอินเวอร์เตอร์เฟสเดียวแบบสะพานเต็มมีข้อเสียเปรียบที่มองเห็นได้ คือ วงจรแบบนี้ต้องการพาวเวอร์ทรานซิสเตอร์ ซึ่งวงจรเฟสเดียวแบบอื่นอีกสองแบบต้องการพาวเวอร์ทรานซิสเตอร์เพียงสองตัวเท่านั้น แต่มีข้อได้เปรียบที่วงจรแบบนี้ไม่ต้องการแท่งกลางของแหล่งจ่ายไฟฟ้าหรือของหม้อแปลงที่ต่อเข้ากับภาระ และโดย

การเพิ่มพาวเวอร์ทรานซิสเตอร์ขึ้นอีกสองตัวมาประกอบเป็นขาของสะพานอีกขาหนึ่งของวงจรก็สามารถกลายเป็นวงจรแหล่งจ่ายไฟฟ้าเอชไอสามเฟสได้ดังแสดงในรูปที่ 1.3.1 (ก) (ซึ่งได้ต่อไดโอดขนานกับพาวเวอร์ทรานซิสเตอร์ไว้ด้วย)



รูปที่ 1.3.1 (ก) วงจรอินเวอร์เตอร์สามเฟสแบบสะพานเต็ม

(ข) รูปคลื่นแรงดันเอาต์พุตของ (ก) เพียงหนึ่งเฟส

พาวเวอร์ทรานซิสเตอร์ Q_1, Q_2, Q_3, Q_4, Q_5 และ Q_6 เป็นพาวเวอร์ทรานซิสเตอร์หลัก ส่วน $Q'_1, Q'_2, Q'_3, Q'_4, Q'_5$ และ Q'_6 จากรูป 1.3.1 (ก) นั้น เป็นทรานซิสเตอร์ที่ช่วยขยายกระแสจากชุดวงจรขับ ซึ่งวิธีนี้จะทำให้ช่วยลดกระแสลัดที่วงจรขับให้น้อยลงได้ โดยดึงกระแสที่จะนำไปขับพาวเวอร์ทรานซิสเตอร์หลัก $Q_1 - Q_6$ จากแหล่งจ่ายไฟดิซีทำให้สามารถใช้ทรานซิสเตอร์ในวงจรขับขนาดเล็กลงได้จึงเป็นการประหยัดค่าอุปกรณ์ในส่วนวงจรขับ ส่วนไดโอดที่ต่อคร่อมระหว่างขา B ของ Q' และ ขา E ของ Q นั้นเป็นส่วนจำกัดกระแสในช่วงแรงดันกลับ (Reverse voltage) ให้ไหลผ่านไดโอดตัวนี้เพื่อป้องกันไม่ให้พาวเวอร์ทรานซิสเตอร์หลัก Q และ Q' ถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ป้อนแรงดันกลับในขณะต้องการให้ปิดนำกระแสมากเกินไปจนเกินการ พังทะลุที่รอยต่อ BE ของ พาวเวอร์ทรานซิสเตอร์ Q และ Q' ได้

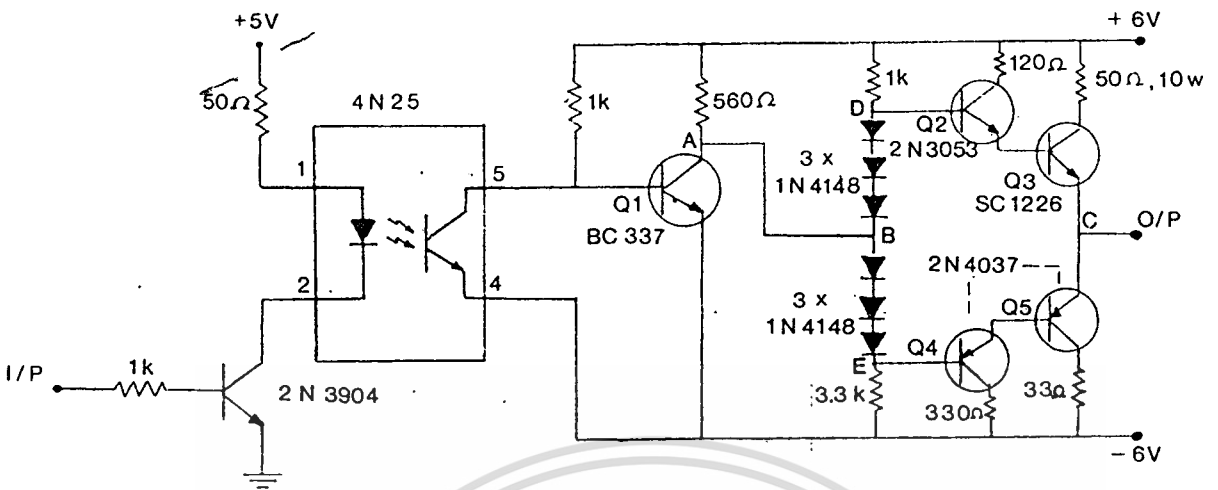
ส่วนของวงจรถับเบอร์ (SNUBBER) ในรูป 1.3.1 ก นั้น ใช้ไดโอดฟื้นตัวกลับคืนเร็ว (fast recovery diode) D เข้าต่อขนานกับ $R_s = 100$ เมื่อแรงดันกระชอกเข้ามา D จะเปิดกระแสเข้าประจุ $C = 0.33 \mu\text{f}$ และ C จะทำหน้าที่คล้ายสปริงคือมีพฤติกรรมเหมือนกับมีการลัดวงจรเกิดขึ้น แรงดันของวงจรถับเบอร์จะลดค่าลงพร้อมกันนั้นก็เบี่ยงกระแสให้ผ่านเข้าไปใน C แทนที่จะผ่านเข้าพาวเวอร์ทรานซิสเตอร์ ครั้นเมื่อ C ได้รับประจุแล้วก็จะมีความดันตกคร่อมมากขึ้นกระแสก็ไหลเบี่ยงกลับไปพาวเวอร์ทรานซิสเตอร์ต่อเมื่อ C ได้รับประจุเต็มที่แล้ว กระแสจะหยุดไหลผ่าน C และไปไหลผ่านพาวเวอร์ทรานซิสเตอร์ทั้งหมด เมื่อ D ได้ฟื้นตัวกลับคืนแล้ว กระแสที่ไหลผ่าน C นั้นจะไหลผ่าน R_s ด้วย และ R_s ทำหน้าที่ลดขนาดการกวัดแกว่งค่าเล็กที่เกิดขึ้น และจำกัดกระแสในช่วงพาวเวอร์ทรานซิสเตอร์เปิดกระแสซึ่งในตอนเริ่มแรกของช่วงนี้ C จะจ่ายประจุออกและไหลผ่านพาวเวอร์ทรานซิสเตอร์ ส่วน D ใช้ทำหน้าที่ลดการเปลี่ยนแปลงชั่วครู่ของ พานะที่ค้างอยู่ฟื้นตัวกลับคืน

รูปที่ 1.3.1 (ข) แสดงรูปคลื่นแรงดันของเอาท์พุทเฟืองเฟลเดี่ยวโดยเป็นรูปคลื่นสี่เหลี่ยม ที่มีค่าแรงดันเท่ากับศูนย์ในบางส่วนของวัฏจักร ซึ่งเกิดขึ้นขณะที่มีกระแสเหนี่ยวนำไหลอยู่ในภาวะ ผ่านไดโอดที่ต่อขนานกับพาวเวอร์ทรานซิสเตอร์ตัวที่ปิดกระแสแล้วไหลผ่านพาวเวอร์ทรานซิสเตอร์ ที่เปิดกระแส จึงเป็นผลทำให้เกิดการลัดวงจรภาวะและมีแรงดันเอาท์พุทเป็นศูนย์ ซึ่งเราอาจปรับ ช่วงเวลาของแรงดันเท่ากับศูนย์ได้พอสมควร จึงมีประโยชน์ในการควบคุมเรกูเลชันของแรงดัน และควบคุมอาร์โมนิคส์ของแรงดันเอาท์พุท ซึ่งนับได้ว่าเป็นข้อได้เปรียบอย่างหนึ่งที่เราอาจมองไม่ ค่อยเห็น (วงจรถับเบอร์เฟลเดี่ยวอีกสองแบบไม่มีข้อได้เปรียบแบบนี้)

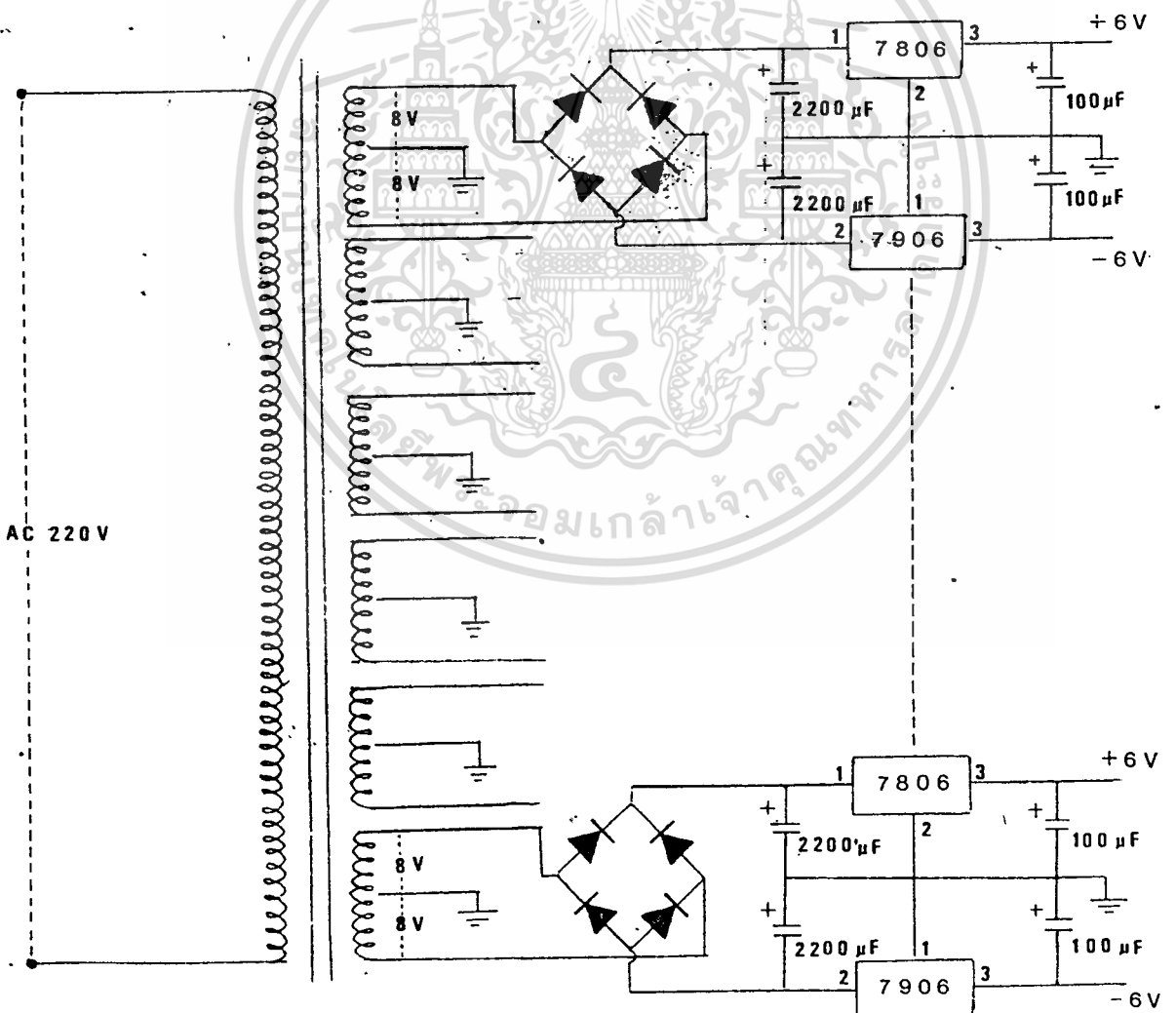
1.4 วงจรถับ (Base Drive Circuit)

ในส่วนของวงจรถับนั้นจะรวมถึงทำหน้าที่ในการแยก (Isolated) ระหว่างวงจรมอเตอร์ โพรเซสเซอร์ กับส่วนวงจรถับดังรูป 1.4.1

การแยกระหว่างวงจรมอเตอร์โพรเซสเซอร์และวงจรถับออกจากกันนั้น จะใช้ไอซีเชื่อมโยง ทางแสง (Opto couple) เบอร์ 4N25 โดยจะมีเอกสารประกอบการใช้ (Specification Document) อยู่ในส่วนของภาคผนวก ทรานซิสเตอร์ Q1 (BC337) นั้นจะทำหน้าที่สวิสซึ่งไฟ +6V ที่และ -6V ที่จุด A (ขาคอลเลคเตอร์ของ Q1) จุด A จะมีไฟ +6V ต่ออยู่กับที่ต่อเมื่อมีล่อ เอกสาร ฝึกค้นย์ออกมาจาก 4N25 ที่ขา 5 ทำให้ Q1 เกิดสถานะนำกระแส จุด A จึงต่อเข้ากับไฟ -6V ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.4.1 วงจรขับหลอดไฟแอลอีดี



รูปที่ 1.4.2 วงจรจ่ายไฟตรงต้น +6V และ -6V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา-17-ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยตรง จะเห็นว่าส่วนนี้จะกลับสัญญาณ (Inverting) ที่ออกมาจากไมโครโพรเซสเซอร์แต่เมื่อผ่าน Q_2, Q_3 , และ Q_4, Q_5 แล้วทำการกลับสัญญาณอีกครั้ง โดยเมื่อจุด A, มีค่าเป็น $-6V$ โดยตรงจะทำให้ Q_4, Q_5 หยุดนำกระแสมีกระแสไหลจาก $+6V$ มาที่จุด B ไอโอดที่ต่อเรียงกันสามตัวทำให้ที่จุด D มีแรงดันสูงพอที่จะทำการไบแอสให้ Q_2, Q_3 นำกระแส ดังนั้นในขณะที่จุด C มีไฟ $+6V$ ต่ออยู่ และเมื่อจุด A มีค่าเป็น $+6V$ จะทำให้ Q_2, Q_3 หยุดนำกระแสและ Q_4, Q_5 นำกระแสแทนเนื่องจากมีกระแสไหลจาก B ไป $-6V$ และแรงดันที่จุด E สูงพอที่จะไบแอสให้ Q_4, Q_5 ได้ทำให้จุด C ต่อเข้ากับไฟ $-6V$ แทน จะเห็นว่ามีการกลับสัญญาณอีกครั้ง ทำให้สัญญาณที่ได้ไม่ผิดไปจากสัญญาณที่ออกมาจากไมโครโพรเซสเซอร์

การสร้างไฟแรงดัน $+6V$ และ $-6V$ นั้น ทำได้โดยการผันหม้อแปลงลดไฟ (step down) จากขดปฐมภูมิ $220V$ มาเป็นขดทุติยภูมิ $8V, 0V$ และ $8V$ โดยค่า $0V$ สร้างจากการเทพหม้อแปลง เมื่อถึงจำนวนรอบที่ทำให้ได้ค่า $8V$ ขึ้น ซึ่งจะต้องพันขดทุติยภูมิถึง 6 ขดด้วยกัน เพราะจุด $0V$ นั้นจะเป็นกราวด์ของชุดวงจรขับแต่ละชุด (สัญญาณขับพาวเวอร์ทุกตัวจะมีกราวด์แยกกัน) ดังรูปที่ 1.4.2.

บทที่ 2

ไมโครโปรเซสเซอร์ MCS 8048

2.1 แนะนำไมโครโปรเซสเซอร์ MCS 8048

ไอซีตระกูล 8048 นี้จัดว่าเป็นชิปที่ค่อนข้างจะมีความพร้อม หรือ ความสมบูรณ์อยู่ในตัวมันเองแล้ว คือไม่เพียงแต่จะประกอบด้วยส่วนชิปที่อยู่อย่างเดี่ยวแต่ยังมีส่วนของหน่วยความจำ, ส่วนของพอร์ต, ส่วนไทมเมอร์, เคาน์เตอร์, และอื่นๆ อีกมากมาย ซึ่งทั้งหมดนี้รวมอยู่ในชิปเพียงชิปเดียวเท่านั้น

ขาสัญญาณต่างๆ ของ 8048 ที่แสดงในรูปที่ 2.1.1 นั้นมีหน้าที่ต่างกันไปดังจะกล่าวรวมๆ อย่างย่อๆ ดังนี้

- 1 T_0 เป็นขาอินพุตทดสอบ หรือใช้เป็นขาเอาต์พุตสัญญาณนาฬิกาของระบบ
- 2 $XTAL_1$ เป็นขาอินพุตของสัญญาณนาฬิกาที่ใช้เป็นฐานเวลาของระบบ หรืออาจจะต่อคริสตัลเข้ากับขา $XTAL_1$ และ $XTAL_2$ ก็ได้
- 3 $XTAL_2$ ต่อคริสตัลกับขาที่ 1 เพื่อให้วงจรออสซิลเลเตอร์ภายในทำงานได้
- 4 RESET เป็นสัญญาณอินพุตแอกทีฟลูนีย์ ใช้รีเซตระบบในตัว 8048
- 5 SS เป็นสัญญาณอินพุตแอกทีฟลูนีย์ ที่ทำงานร่วมกับ ALE เวลาตรวจสอบโปรแกรมด้วยวิธี SINGLE STEP
- 6 INT เป็นอินพุตของสัญญาณอินเตอร์รัพท์ หรือใช้เป็นขาอินพุตธรรมดาก็ได้
- 7 EA เป็นสัญญาณอินพุตเพื่อเลือกให้ 8048 รันโปรแกรมในหน่วยความจำโปรแกรมภายใน หรือหน่วยความจำภายนอก
- 8 RD เป็นสัญญาณเอาต์พุตที่แอกทีฟลูนีย์ เมื่อ 8048 อ่านข้อมูลจากหน่วยความจำข้อมูลที่อยู่ภายนอก
- 9 PSEN เป็นสัญญาณเอาต์พุตที่แอกทีฟลูนีย์ เมื่อ 8048 เฟตซ์คำสั่งจากหน่วยความจำโปรแกรมที่อยู่ภายนอก
- 10 WR เป็นสัญญาณเอาต์พุตที่แอกทีฟลูนีย์ เมื่อ 8048 เขียนข้อมูลลงในหน่วยความจำข้อมูลภายนอก
- 11 ALE เป็นสัญญาณที่ใช้แลตซ์แอดเดรส ที่มัลติเพล็กซ์มาทางดาต้าบัสและใช้แลตซ์ข้อมูลทางพอร์ต 2 ด้วย
- 12-19 DB_0-DB_7 ดาต้าบัส เป็นทางผ่านเข้าออกของข้อมูล และแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ของระบบ 8048 การศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
20 v... กราวนด์ของระบบ 8048
ไม่หวังกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

21-24 P₂₀-P₂₃ 4 บิตล่างของพอร์ท 2

25 PROG เป็นสัญญาณ STROBE สำหรับการต่อขยายพอร์ทด้วย 8243

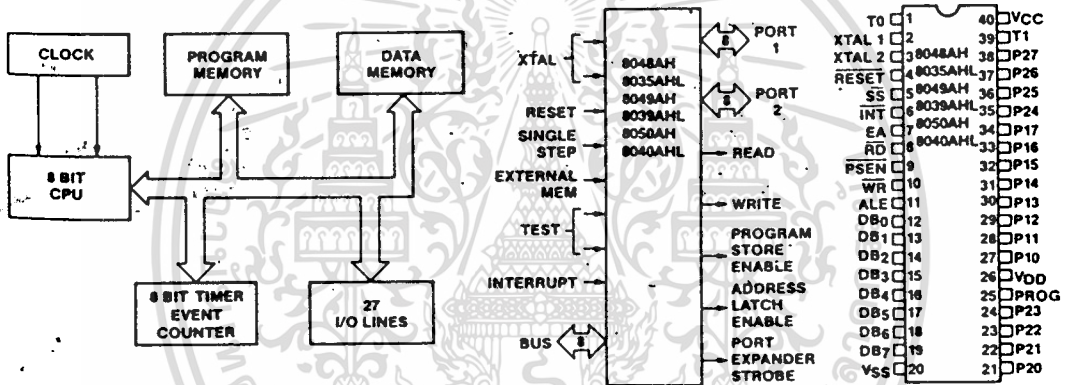
26 V_{DD} ต่อกับแบตเตอรี่ขนาดเล็ก เพื่อเลี้ยงแรมภายใน ไม่ให้ข้อมูลสูญหายเมื่อไม่มีไฟเลี้ยง V_{CC}

27-34 P₁₀-P₁₇ พอร์ท 1

35-38 P₂₄-P₂₇ 4 บิตบนของพอร์ท 2

39 T₁ เป็นขาอินพุททดสอบ หรือใช้เป็นขาอินพุทของสัญญาณนาฬิกาที่ป้อนให้แก่ไทเมอร์/เคาน์เตอร์ ภายในตัว 8048

40 V_{CC} ต่อกับไฟเลี้ยง +5 โวลต์



รูปที่ 2.1.1 การจัดการของไอซี MCS 8048

2.2 หน่วยความจำ

ในระบบของ 8048 นี้ได้มีการแบ่งหน่วยความจำออกเป็นสองลักษณะตามชนิดของข้อมูลที่เก็บ คือ

1. หน่วยความจำข้อมูล
2. หน่วยความจำโปรแกรม

ถ้าจะพูดกันอย่างง่ายแล้ว หน่วยความจำข้อมูลหมายถึง หน่วยความจำส่วนที่เป็นแรม (RAM) ที่เราสามารถเขียน หรืออ่านข้อมูลจากหน่วยความจำส่วนนี้ได้แต่ไม่สามารถจะรันโปรแกรม บนหน่วยความจำนี้ได้

หน่วยความจำทั้งสองประเภทนี้ถูกแบ่งแยกออกจากกันด้วยคำสั่งทางซอฟต์แวร์ และลักษณะการติดต่อทางฮาร์ดแวร์ด้วย กล่าวคือ มีคำสั่งเฉพาะสำหรับติดต่อกับหน่วยความจำชนิดใดชนิดหนึ่ง

และมีการจัดสัญญาณ STROBE ในการติดต่อกับหน่วยความจำแต่ละชนิด แยกต่างหากกันด้วย

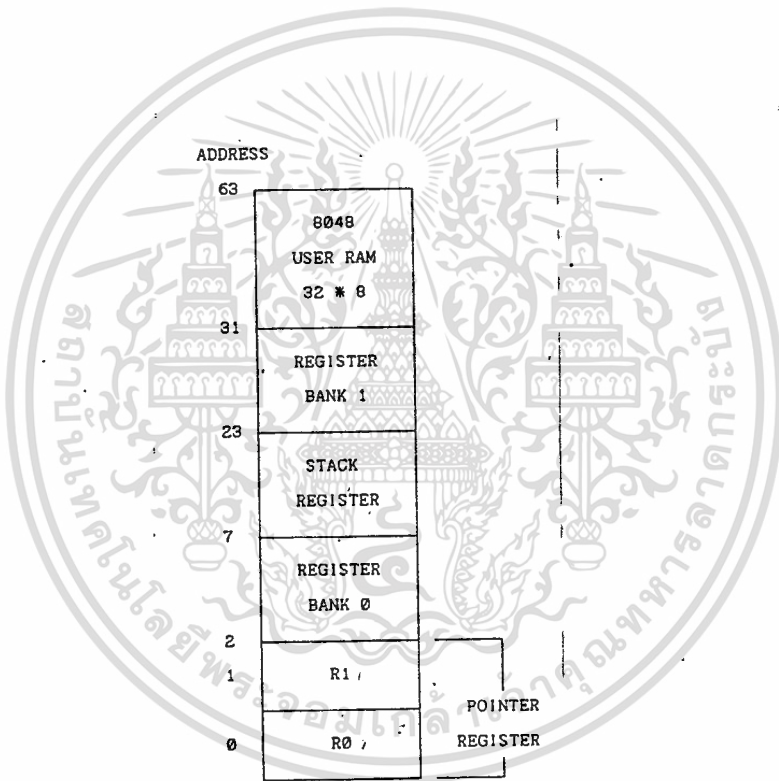
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา -19- ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 หน่วยความจำข้อมูล

สำหรับหน่วยความจำข้อมูลภายในตัว 8048 หน่วยความจำข้อมูลจะแบ่งออกเป็น 2 ส่วน ดังแสดงในรูป 2.2.1

Device	Internal Memory		RAM Standby
8050AH	4K × 8 ROM	256 × 8 RAM	yes
8049AH	2K × 8 ROM	128 × 8 RAM	yes
8048AH	1K × 8 ROM	64 × 8 RAM	yes
8040AHL	none	256 × 8 RAM	yes
8039AHL	none	128 × 8 RAM	yes
8035AHL ¹	none	64 × 8 RAM	yes



รูปที่ 2.2.1 แสดงการแบ่งหน่วยความจำข้อมูลภายในไมโครโปรเซสเซอร์

ส่วนรีจิสเตอร์ ได้แก่ หน่วยความจำแอดเดรส 00-07 ซึ่งเราเรียกว่ารีจิสเตอร์แบงก์ 1 (BANK1) แต่ละแบงก์จะมีรีจิสเตอร์อยู่ 8 ตัว ซึ่งมีชื่อเรียกว่า R₀-R₇ ในการใช้งานรีจิสเตอร์เหล่านี้ เราจะสามารถใช้งานได้ทีละแบงก์เท่านั้น โดยใช้คำสั่ง SEL RB₀ หรือ SEL RB₁ ในการเลือกแบงก์ของรีจิสเตอร์

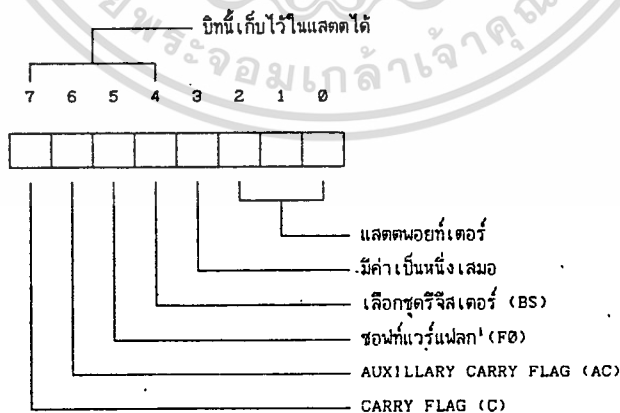
รีจิสเตอร์ทั้ง 8 ตัวนี้เป็นรีจิสเตอร์สำหรับใช้งานทั่วไป มีคำสั่งต่างๆ ที่กระทำร่วมกับรีจิสเตอร์ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยามให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาข้อมูลต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เตอร์เหล่านี้ครบแต่จะมีรีจิสเตอร์ R_0 และ R_1 สองตัวเท่านั้นที่มีหน้าที่พิเศษ คือ สามารถใช้เป็นตัวอ้างอิงหน่วยความจำข้อมูลได้โดยค่าแอดเดรสจะมีขนาดเพียง 8 บิตเท่านั้น เป็นการจำกัดหน่วยความจำข้อมูลให้มีเพียง 256 ไบต์ ซึ่งจะคลุมพื้นที่ส่วนที่เป็นรีจิสเตอร์ และส่วนที่เป็นแอสตคด้วยทำให้เราสามารถติดต่อกับรีจิสเตอร์ หรือ แอสตค ในลักษณะที่เป็นหน่วยความจำข้อมูลได้อีกวิธีหนึ่ง การติดต่อดังวิธีนี้ส่วนใหญ่จะใช้กับแอสตคมากกว่าส่วนอื่น เพราะว่าคำสั่งต่างๆ ของ 8048 ที่เกี่ยวข้องกับแอสตค โดยตรงมีเพียงคำสั่ง CALL กับคำสั่ง RET(RETR) เท่านั้น

ส่วนของแอสตค ประกอบด้วยแอสตคทั้งหมด 8 ชั้น แต่ละชั้นประกอบด้วยหน่วยความจำข้อมูลสองไบต์ เพื่อให้เพียงพอกับการเก็บค่าแอดเดรสและเก็บสถานะของโปรแกรม การที่มีแอสตคเพียง 8 ชั้นนี้ทำให้มีขีดจำกัดในการเรียกโปรแกรมย่อย (SUBROUTINE) ซ้อนๆ กันได้ไม่เกิน 8 ชั้น ถ้าหากเกิน แอสตคพอยเตอร์ก็จะย้อนกลับไปชี้ที่แอสตคชั้นแรกใหม่

โดยปกติแล้วแอสตคพอยเตอร์จะชี้ที่ S_0 เป็นแอสตคชั้นแรกเมื่อมีการ CALL หรือการอินเตอร์รั้นท์ ก็จะเก็บค่าแอดเดรส และสถานะของโปรแกรม (PROGRAM STATUS WORD) ลงในแอสตค แล้วเพิ่มค่าแอสตคพอยเตอร์ให้ชี้ไปที่ S_1 และเมื่อเอ็กซ์ิต์คำสั่ง RET(RETR) ก็จะลดค่าแอสตคพอยเตอร์ลงไปที่ S_0 แล้วนำค่าแอดเดรส กับสถานะของโปรแกรมที่เก็บไว้ในแอสตค S_0 กลับคืนที่เดิม

PSW เป็นรีจิสเตอร์ตัวหนึ่งที่ทำหน้าที่เก็บค่าของแอสตคพอยเตอร์ และสถานะของโปรแกรม (สถานะของโปรแกรม ก็คือ ค่าของแฟลกต่างๆ) แต่ละบิตของ PSW มีความดังรูปที่ 2.2.2



รูปที่ 2.2.2 แสดงความหมายของแต่ละบิตใน PSW

C หมายถึง แฟลกตัวทศที่ได้จากการบวก หรือโดยการเลื่อนบิต

AC หมายถึง แฟลกตัวทศที่ได้จากการบวกบิตที่ 3 ของตัวเลข BCD จะใช้ทำงานร่วมกับค่า

สั่ง DA (DECIMAL ADJUST)

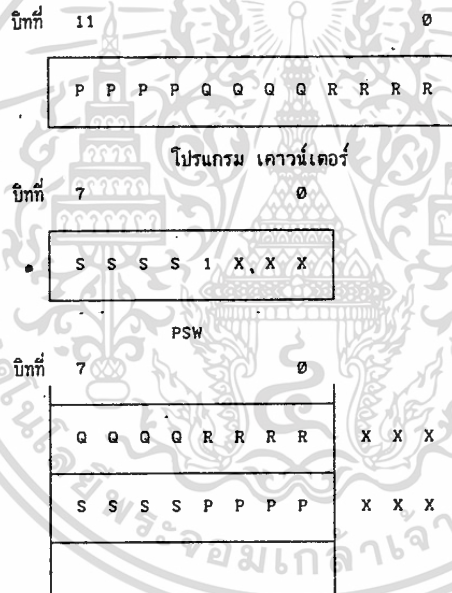
FO หมายถึง ซอฟต์แวร์แฟลกสามารถใช้งานได้ทั่วไป คำสั่งที่มีผลต่อแฟลกนี้ได้แก่ คำสั่ง CLR FO และ CPL FO

BS หมายถึง แฟลกที่แสดงการเลือกเบงค์ของรีจิสเตอร์

นอกจากแฟลก FO แล้วยังมีซอฟต์แวร์แฟลก F1 อีกตัวหนึ่ง ที่มีลักษณะเช่นเดียวกับแฟลก FO ต่างกันเพียงแต่ว่าแฟลก F1 ไม่ได้เก็บไว้ใน PSW ดัวย

บิต 0 - บิต 2 ของ PSW จะเก็บค่าของแอสตคนอยเตอร์เอาไว้

เมื่อมีการเรียกโปรแกรมย่อยไม่ว่าจะโดยการ CALL หรือ โดยการเรียกอินเตอร์รั้นท์จะมีการเก็บค่าแอดเดรสจากโปรแกรมเคาน์เตอร์และค่าของ PSW ลงในแอสตค ในลักษณะการจัดเรียงดังรูปที่ 2.2.3



รูปที่ 2.2.3 แสดงการจัดเก็บค่าโปรแกรมเคาน์เตอร์ และ PSW

แต่สำหรับการ RETURN นั้น จะแบ่งออกเป็นสองแบบคือ คำสั่ง RET ใช้สำหรับออกจากโปรแกรมย่อย จะเอาเฉพาะค่าแอดเดรสใส่คืนให้แก่โปรแกรมเคาน์เตอร์เท่านั้น ส่วนคำสั่ง RETR นั้นใช้สำหรับออกจาก INTERRUPT SERVICE ROUTINE (ISR) ซึ่งจะนำเอาทั้งค่า แอดเดรส และสถานะของโปรแกรม ใส่กลับคืนที่เดิม

นอกจากหน่วยความจำข้อมูลภายในตัว 8048 แล้ว ยังสามารถต่อขยายหน่วยความจำแรมภายนอกได้อีก 256 ไบต์ สำหรับการติดต่อกับหน่วยความจำข้อมูลภายนอก จะยังคงใช้รีจิสเตอร์ R₀ และ R₁ ในการอ้างถึงแอดเดรสเช่นเดิม จะต่างกันตรงที่มีคำสั่งในการติดต่อกับหน่วยความจำ เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา -22- อังอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำข้อมูล แยกจากกันคือ ใช้คำสั่ง MOV ในการติดต่อกับหน่วยความจำข้อมูลภายใน และใช้คำสั่ง MOVX ในการติดต่อกับหน่วยความจำข้อมูลภายนอก

2.2.2 หน่วยความจำโปรแกรม

เนื่องจากเราไม่สามารถรันโปรแกรมใดๆ ในพื้นที่หน่วยความจำข้อมูลตั้งนั้น 8048 จึงจัดหน่วยความจำขนาด 4 KBYTE ไว้อีกส่วนหนึ่งเพื่อทำหน้าที่เก็บโปรแกรม หรือค่าคงที่ ซึ่งเราเรียกหน่วยความจำนี้ว่า หน่วยความจำโปรแกรม

หน่วยความจำโปรแกรม แบ่งออกเป็น 2 ส่วนคือ หน่วยความจำโปรแกรมภายใน และหน่วยความจำโปรแกรมภายนอก หน่วยความจำโปรแกรมภายในก็คือ หน่วยความจำประเภท ROM หรือ EPROM ที่อยู่ในตัว 8048 ซึ่งส่วนใหญ่มีขนาดไม่เกิน 2 KBYTE ส่วนหน่วยความจำโปรแกรมภายนอกนั้น เป็นหน่วยความจำที่เราต่ออยู่ภายนอกตัว 8048 โดยอาจจะเป็นการต่อเพื่อขยายหน่วยความจำร่วมกับหน่วยความจำภายใน ให้มีขนาดครบ 4 KBYTE หรือ อาจต่อเป็นหน่วยความจำภายนอกทั้งหมด 4 KBYTE เลยก็ได้ เช่นในกรณีของ 8035 หรือ 8039 นั้น ไม่มีหน่วยความจำโปรแกรมอยู่ภายในเลย จึงต้องต่อหน่วยความจำโปรแกรมทั้ง 4 KBYTE ไว้ภายนอกทั้งหมด

8048 สามารถเลือกที่จะรันโปรแกรมในหน่วยความจำโปรแกรมภายนอก หรือภายในก็ได้ โดยการส่งสัญญาณเลือกไปที่ ขาคควบคุม EA (EXTERNAL ACCESS)

ถ้าขา EA ได้รับลอจิกศูนย์ 8048 จะรันโปรแกรมในหน่วยความจำภายในแต่เนื่องจากมีพื้นที่หน่วยความจำเพียง 1-2KBYTE เท่านั้น จึงอาจจะรันโปรแกรมต่อที่หน่วยความจำภายนอกได้ แต่ถ้าหาก EA ได้รับลอจิกหนึ่ง 8048 ก็จจะรันโปรแกรมในหน่วยความจำโปรแกรมภายนอกทั้งหมด

ลักษณะพิเศษของหน่วยความจำโปรแกรมอีกอย่างหนึ่งก็คือ การแบ่งพื้นที่หน่วยความจำออกเป็นส่วนย่อยๆ หลายๆ ส่วนดังนี้

แบ่งหน่วยความจำโปรแกรมขนาด 4KBYTE ออกเป็น 2 แบลงค์ (BLANK) แบลงค์ละ 2 KBYTE และในแต่ละแบลงค์ยังแบ่งออกเป็น 8 เพจ (PAGE) ย่อยๆ เพจละ 256 ไบท์ การแบ่งหน่วยความจำออกเป็นส่วนๆ เช่นนี้ก็เพื่อให้มีการอ้างถึงแอดเดรสได้ง่ายในลักษณะ PAGE ADDRESSING MODE ทำให้คำสั่งมีขนาดสั้นลง นั่นคือ สามารถประหยัดเนื้อที่หน่วยความจำ (ซึ่งมีน้อยอยู่แล้ว) และสามารถทำงานได้รวดเร็วยิ่งขึ้น

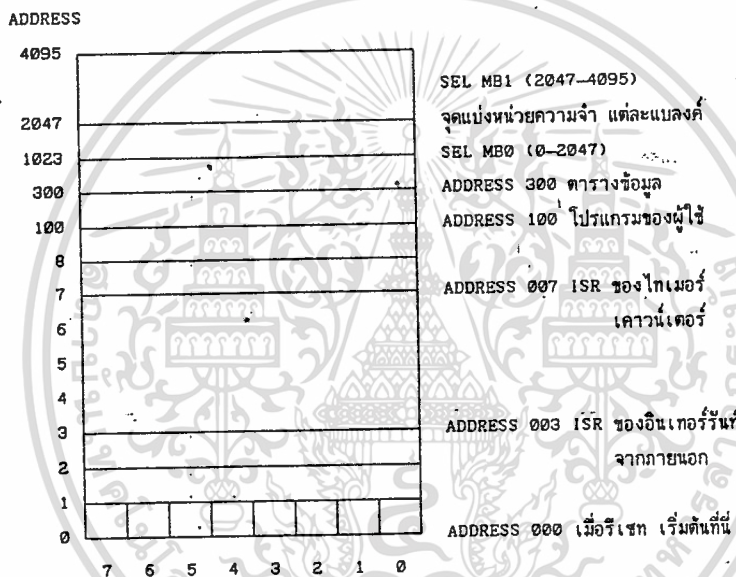
คำสั่งที่มี PAGE ADDRESSING MODE เช่น คำสั่ง JMPP @A หรือ คำสั่ง MOVP A,@A จะไม่สามารถทำงานออกนอกเพจไปได้เลย แต่สำหรับคำสั่งที่มี ADDRESSING MODE เป็นอย่างอื่นเช่นคำสั่ง JMP หรือ CALL นั้นสามารถกระโดดข้ามไปยังเพจใดๆ ก็ได้ แต่อย่างไรก็ตามก็ยัง ไม่สามารถกระโดดข้ามเซตแบลงค์ไปได้อยู่ดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับอาจารย์และบุคลากรในมหาวิทยาลัยเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจะข้ามจากแบลงค์หนึ่งไปยังอีกแบลงค์หนึ่งนั้น ก็คือ การเซท หรือ รีเซท A_{11} ของโปรแกรมเคอร์เนลเตอร์นั่นเอง คำสั่งที่หน้าที่นี้ก็คือ คำสั่ง SEL MB₀ หรือ SEL MB₁ ซึ่งจะต้องตามหลังด้วยคำสั่ง JMP หรือ CALL

หน่วยความจำโปรแกรมจะมีการแบ่งพื้นที่ของหน่วยความจำโปรแกรมไปใช้ในงานต่างๆ กันด้วย ดังรูปที่ 2.2.4

1. แอดเดรส ๓๐๐๐๐ เป็นจุดเริ่มต้นของโปรแกรมหลังจากการรีเซท
2. แอดเดรส ๓๐๐๐๓ เป็นจุดเริ่มต้นของ INTERRUPT SERVICE ROUTINE (ISR) ของการอินเทอร์รัพท์ จากภายนอกที่เข้ามาทางขา JNT



รูปที่ 2.2.4 แสดงการจัดแบ่งพื้นที่ หน่วยความจำโปรแกรม

3. แอดเดรส ๓๐๐๐๗ เป็นจุดเริ่มต้นของ ISR ของการอินเทอร์รัพท์ จากตัวไมโครเคอร์เนล/เคอร์เนลที่อยู่ภายใน 8048

แอดเดรส ๓๐๑๐๐ เป็นจุดเริ่มต้นของ USER PROGRAM แต่อาจจะไม่จำเป็นต้องเริ่มที่แอดเดรส ๓๐๑๐๐ ก็ได้

แอดเดรส ๓๐๓๐๐-๓๐๓FF เป็นที่เก็บข้อมูลค่าคงที่ต่างๆ จำนวน 256 ไบต์ เราสามารถอ่านข้อมูลมาใช้ได้โดยสะดวก ด้วยคำสั่งที่ออกแบบมาใช้กับหน่วยความจำส่วนนี้ โดยเฉพาะคำสั่ง MOVP3 A,@A

2.3 พอร์ต (PORT)

นอกจาก 8048 จะมีหน่วยความจำภายในตัวมันเองแล้ว ยังมีพอร์ตเอาไว้ให้เราใช้งานอีก 3 พอร์ตด้วยกัน พอร์ตทั้งสามได้แก่ พอร์ต 1, พอร์ต 2 และ ดาต้าบัสนั่นเอง

พอร์ต 1 และ พอร์ต 2 นั้นมีลักษณะพิเศษสามารถเป็นได้ทั้งอินพุตและเอาต์พุต เมื่อเราเอาต์พุตค่าตัวเลขนั้นก็จะถูกแลตซ์ค้างอยู่ที่พอร์ตนั้น ถ้ามีข้อมูลใหม่เข้ามา ก็จะทับข้อมูลเก่าไป ส่วนการใช้งานเป็นอินพุตพอร์ตนั้นก็มีวิธีการเพิ่มขึ้นมาอีกชนิด คือ เมื่อเราต้องการให้บิตใดเป็นอินพุต เราก็จะต้องเอาต์พุตลอจิกหนึ่งออกมาที่บิตนั้น จากนั้นก็อ่านค่าจากพอร์ตเข้าไปวงจรภายนอกสามารถที่จะดึง ลอจิกหนึ่งที่บิตนั้นให้กลายเป็นลอจิกศูนย์ตั้งนั้นการตรวจสอบว่าวงจรภายนอกส่งสัญญาณอินพุตเข้าหรือไม่ ก็ทำได้โดยการตรวจสอบลอจิกศูนย์ที่อินพุตบิตนั้นเอง

ส่วนบิตใดที่เราเอาต์พุตลอจิกศูนย์ออกไปนั้น เมื่ออ่านกลับมาก็ยังคงเป็นลอจิกศูนย์อยู่ดี วงจรภายนอกไม่สามารถดึงให้กลายเป็นลอจิกหนึ่งได้

ข้อแตกต่างระหว่าง พอร์ต 1 และพอร์ 2 คือ 4 บิตล่างของพอร์ต 2 ($P_{20}-P_{23}$) ถูกนำไปใช้งานได้สองลักษณะคือ มีการมัลติเพล็กซ์ค่าแอดเดรส 4 บิตสูงของโปรแกรมเคาน์เตอร์ (A_8-A_{11}) กับค่าของพอร์ต ดังนั้นในการใช้งานส่วนใหญ่แล้ว จะมีความสะดวกที่จะใช้ $P_{20}-P_{23}$ เป็นเอาต์พุตพอร์ต มากกว่าเป็นอินพุตพอร์ตโดยการหาตัวเลขขนาด 4 บิต เช่น 74LS175 มาแลตซ์ค่าเอาต์พุตเอาไว้ได้

คำสั่งอินพุต เอาต์พุต ที่สามารถใช้กับพอร์ตทั้งสามนี้ ได้แก่

IN A, P_n และ OUT P_n, A เมื่อ n เป็นเบอร์พอร์ต

ดาต้าบัส นั้นจัดว่าเป็นพอร์ตๆ หนึ่งเหมือนกัน แต่ลักษณะของดาต้าบัสไม่เหมือนกับพอร์ต 1 หรือ พอร์ต 2 เท่าใดนัก โดยปกติดาต้าบัสจะทำหน้าที่ส่งแอดเดรส และรับส่งข้อมูลสลับกันแบบมัลติเพล็กซ์ตลอดเวลา การใช้งานดาต้าบัสในฐานะเป็นพอร์ตนั้น มีตำราหลายเล่มกล่าวเอาไว้ว่าจะใช้บัสเป็นพอร์ตได้ก็ต่อเมื่อ ไม่มีการต่อหน่วยความจำภายนอกเลย และคำสั่งที่ใช้คือ คำสั่ง INS A, BUS และ OUTL BUS, A เป็นต้น แต่โดยทั่วไปแล้ว เรามักจะต่อหน่วยความจำภายนอกกับ 8048 เสมอ การใช้งานดาต้าบัสเป็นพอร์ จึงไม่ค่อยมีให้เห็นกันนัก

นอกจากลักษณะพิเศษของพอร์ และคำสั่งอินพุต เอาต์พุตธรรมดาแล้ว 8048 ยังมีคำสั่งอีกสองแบบ ที่ใช้ติดต่อกับพอร์ตคือ

คำสั่ง ORL $PORT, DATA$ เป็นการนำค่าตัวเลข $DATA$ ไป AND กับข้อมูลในพอร์ต

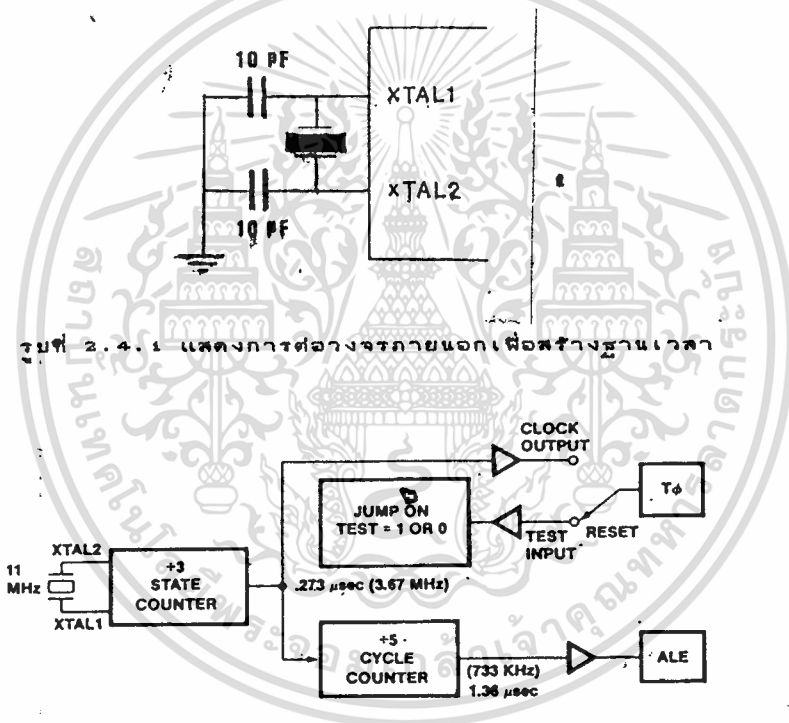
คำสั่ง ORL $PORT, DATA$ เป็นการนำค่าตัวเลข $DATA$ ไป OR กับข้อมูลในพอร์ต

2.4 สัญญาณนาฬิกาของระบบและสัญญาณควบคุม

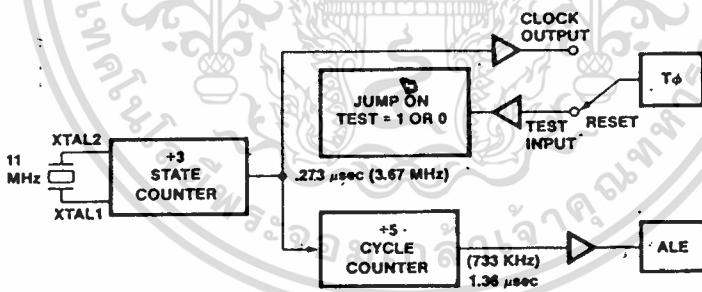
ขอเริ่มจากสัญญาณนาฬิกาที่เป็นฐานเวลาของระบบกันก่อน สัญญาณนาฬิกานี้อาจจะสร้างจากวงจรออสซิลเลเตอร์ภายนอก แล้วป้อนให้แก่ 8048 ทางขา XTAL₁ แต่ภายใน 8048 เองก็มีวงจรออสซิลเลเตอร์อยู่ภายในอยู่แล้วเพียงแต่ต่อคริสตอล และตัวเก็บประจุอีก 2 ตัว ดังรูปที่ 2.4.1 ก็จะทำให้มีสัญญาณนาฬิกาขึ้นกับความถี่ของคริสตอล ซึ่งมีให้เลือก 2 รุ่นคือ

คริสตอลขนาด 1 MHz - 6 MHz โดยมากจะใช้ความถี่ 6 MHz กับชิปไมโครกรม 8048 ทัวๆ ไป และจะใช้คริสตอลขนาด 4 MHz - 11 MHz กับไอซี 8039 และ 8049

สัญญาณนาฬิกาที่ได้จากคริสตอลถูกหารความถี่ด้วย 3 ได้เป็นสัญญาณนาฬิกาของระบบ นำสัญญาณนี้ไปหารความถี่ด้วย 5 จะได้เป็นสัญญาณ ALE ดังรูปที่ 2.4.2



รูปที่ 2.4.1 แสดงการต่อวงจรภายนอกเพื่อสร้างฐานเวลา



รูปที่ 2.4.2 แสดงระบบสัญญาณนาฬิกาภายใน 8048

นอกจากสัญญาณนาฬิกาที่ใช้เป็นหลักฐานเวลาของระบบแล้ว 8048 ก็ยังต้องการสัญญาณจากภายนอก เพื่อมาควบคุมการทำงานเช่น สัญญาณ RESET, EA, INT, SS เป็นต้น และในทางตรงกันข้าม ก็ต้องส่งสัญญาณควบคุม ไปยังอุปกรณ์ภายนอกเช่นกัน สัญญาณเหล่านี้ ได้แก่ ALE, PSEN, RD เป็นต้น เราจะมาทำความรู้จักกับสัญญาณควบคุมเหล่านี้กันเลย โดยเริ่มจาก

2.4.1 สัญญาณควบคุมเอาท์พุท ได้แก่ สัญญาณ ALE, PSEN, WR, RD

ALE (ADDRESS LATCH ENABLE) เป็นสัญญาณนาฬิกาที่ออกมาทุกๆ แมกซีนไซเคิลหรือทุกๆ สัญญาณ CLK 5 ลูกตามที่กล่าวมาแล้ว เราจะใช้ขอบขาลงของสัญญาณ ALE ในการแลทช์เอกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าของแอดเดรสไบท์ต่ำ ที่มีลติเพิล็กซ์มาทางด้านดาต้าบัส

PSEN (PROGRAM STORE ENABLE) เป็นสัญญาณที่แอกทีฟศูนย์ และจะแอกทีฟเฉพาะเมื่อ 8048 เฟทซ์คำสั่งจากหน่วยความจำโปรแกรมภายนอกเท่านั้น

ในการต่อหน่วยความจำโปรแกรมภายนอกเข้ากับ 8048 นั้น เราจำเป็นต้องแลทซ์ค่าแอดเดรสที่มีลติเพิล็กซ์มาทางด้านดาต้าบัสเสียก่อน โดยใช้สัญญาณ ALE เป็น STROBE (ไอซีที่เรานิยมใช้แลทซ์ค่าแอดเดรสนี้เป็นไอซี 74LS373) ค่าแอดเดรส 8 บิตที่ได้จากการแลทซ์ เมื่อนำมารวมกับแอดเดรส 4 บิตสูง ($A_0-A_{3,1}$) ซึ่งได้จากพอร์ท 2 $P_{20}-P_{23}$ ก็จะได้แอดเดรสขนาด 12 บิตที่จะใช้อ้างอิงหน่วยความจำโปรแกรมได้ครบ 4 KBYTE จากนั้น เราใช้สัญญาณ PSEN เป็น STROBE ในการเฟทซ์คำสั่งจากหน่วยความจำเข้าไปยัง 8048 ได้

WR และ RD เป็นสัญญาณที่แอกทีฟศูนย์เช่นกัน แต่จะแอกทีฟเมื่อ 8048 มีการติดต่อกับหน่วยความจำข้อมูลภายนอก หรือเมื่อเราใช้ดาต้าบัสมาทำหน้าที่เป็นพอร์ท เราก็ใช้สัญญาณ WR หรือ RD ที่แอกทีฟมาเป็น STROBE ในการเขียน หรืออ่านข้อมูลกับทางบัสพอร์ทได้ด้วย

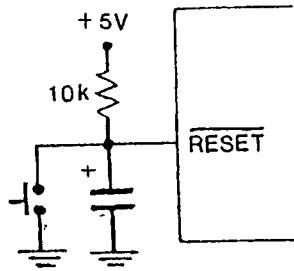
จากที่กล่าวมาแล้ว พอจะสรุปเรื่องราวของสัญญาณ PSEN, WR, RD ได้ว่า สัญญาณ STROBE ทั้งสามนี้ จะสามารถแอกทีฟได้เพียงครั้งละ หนึ่งสัญญาณเท่านั้น เช่น ในช่วงการเฟทซ์คำสั่ง สัญญาณ PSEN ทั้งสามนี้ จะสามารถแอกทีฟได้เพียงครั้งละ หนึ่งสัญญาณเท่านั้น เช่น ในช่วงการเฟทซ์คำสั่งสัญญาณ PSEN จะแอกทีฟ ส่วน WR และ RD จะไม่แอกทีฟ เป็นต้น นอกจากคำสั่งที่จะทำให้สัญญาณแต่ละเส้นแอกทีฟได้ แสดงไว้ในตาราง

สัญญาณ	สัญญาณจะแอกทีฟเมื่อ
RD	MOVX A,@R และ INS A,BUS
WR	MOVX @R,A และ OUTL BUS,A
PSEN	การเฟทซ์คำสั่งจากภายนอก MOVP,MOVP3

2.4.2 สัญญาณควบคุมอินพุท ได้แก่ สัญญาณ RESET,SS,EA และสัญญาณ INT รายละเอียดของสัญญาณต่างๆ มีดังนี้

RESET เป็นสัญญาณอินพุทที่ยังคัยให้ชิพริเริ่มทำงานใหม่ โดยการป้อนลอจิกศูนย์ที่ขาอินพุทนี้

ในการใช้งานทั่วไปแล้ว จะมีสวิทช์กดติดปล่อยดับทำหน้าที่ส่งสัญญาณรีเซทซึ่งมีวงจรถังรูปที่ 2.4.3 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่สามารถตีพิมพ์ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4.3 การต่อวงจรภายนอกเพื่อรีเซ็ต 8048

เมื่อ 8048 ถูกรีเซ็ต จะทำให้

1. โปรแกรมเคาน์เตอร์ และแอสตคพอยเตอร์ ถูกรีเซ็ตเป็นศูนย์
2. เลือกหน่วยความจำแปลงค่าศูนย์ และ รีจิสเตอร์แบ่งค่าศูนย์
3. อิตนเตอร์รัท ถูกติสเอบีล
4. แฟล็ก F_0 และ F_1 และไทเมอร์แฟล็ก ถูกเคลียร์
5. T_0 จะกลายเป็นขาอินพุท

SS' (SINGLE STEP) เป็นสัญญาณอินพุท ที่สามารถควบคุมให้ 8048 ทำงานทีละคำสั่งตามจังหวะที่เราต้องการคือ ถ้าเราส่งสัญญาณ SS ให้มีจังหวะคล้องจองกับสัญญาณ ALE ก็จะสามารถบังคับให้ 8048 หยุดการทำงานของมันได้ตรงเท่าที่สัญญาณ SS ยังคงแอกทีฟอยู่ และในระหว่างที่ 8048 นี้หยุดทำงาน ตัวมันเองจะแสดงค่าแอดเดรสออกมาทางดาต้าบัส และ 4 บิตล่างของพอร์ท 2 ($P_{20}-P_{23}$) จากลักษณะนี้เองจึงเป็นการอำนวยความสะดวกแก่ผู้ใช้ในการติดตามการทำงานของ 8048 ว่า เป็นไปตามโปรแกรมที่เราต้องการหรือไม่

EA (EXTERNAL ACCESS) ตามที่กล่าวมาแล้วว่า 8048 นั้นแบ่งหน่วยความจำโปรแกรมออกเป็น ส่วนภายนอกกับส่วนภายใน 8048 เอง การให้ลลอจิกแก่ขา EA เป็นการบอกให้ 8048 รันโปรแกรมในหน่วยความจำส่วนใด

ถ้า EA มีลลอจิกหนึ่ง 8048 จะรันโปรแกรมในหน่วยความจำภายนอกทั้งหมด (ในกรณีของ 8035 และ 8039 หรือเบอร์อื่นๆ ที่ไม่มีหน่วยความจำโปรแกรมภายใน เราจำเป็นต้องต่อขา EA นี้กับลลอจิกหนึ่งเสมอ)

แต่ถ้าหากให้ EA มีลลอจิกศูนย์จะเป็นการสั่งให้รันโปรแกรมจากหน่วยความจำภายใน และอาจจะรันโปรแกรมต่อมาถึงหน่วยความจำภายนอกที่เราต่อเพิ่มก็ได้

8048 จะยอมรับลลอจิกของสัญญาณ EA ในช่วงที่มีการรีเซ็ตเท่านั้นการเปลี่ยนแปลงลลอจิกของ EA เวลาอื่นจะไม่มีผลเกิดขึ้น

2.5 การอินเทอร์รัทท์

อินเทอร์รัทท์เป็นกระบวนการอย่างหนึ่งที่ ระบบไมโครคอมพิวเตอร์ทั้งหลายจำเป็นต้องมีไว้ แม้แต่ระบบเล็กๆ อย่าง 8048 เองก็ยังมีระบบอินเทอร์รัทท์สองอย่างด้วยกัน คือ

1. อินเทอร์รัทท์จากอุปกรณ์ภายนอก ที่ส่งสัญญาณอินเทอร์รัทท์มาทางขา INT
2. อินเทอร์รัทท์จากภายใน เป็นการอินเทอร์รัทท์โดย ไทเมอร์, เควนเตอร์ที่อยู่ภายใน 8048

เมื่อ 8048 ถูกอินเทอร์รัทท์ ไม่ว่าจะโดยอุปกรณ์ชนิดใดก็ตาม 8048 จะตรวจสอบสถานะการอินเทอร์รัทท์ของตัวเองเสียก่อน ว่าถูกอินาเบิ้ลแล้วหรือยัง ถ้าได้รับการอินาเบิ้ลไว้แล้ว ก็ 8048 จะหยุดทำงานที่กำลังดำเนินอยู่เดิม และก้าวไปตอบรับอุปกรณ์ที่ส่งสัญญาณอินเทอร์รัทท์มา โดยการเก็บค่าของโปรแกรมเคาน์เตอร์และสถานะของโปรแกรม (PSW) เข้าไปไว้ในแอสตและกระโดดไปยัง ISR ซึ่งมีอยู่สองแห่งด้วยกันตามประเภทของอินเทอร์รัทท์

การขออินเทอร์รัทท์โดยอุปกรณ์ภายนอก ด้วยการส่งสัญญาณมาทางขา INT นั้นจะทำให้ 8048 กระโดดไปยัง ISR ที่ตำแหน่ง ๑๐๐๐๓

แต่เมื่อ 8048 ถูกอินเทอร์รัทท์ด้วยไทเมอร์/เควนเตอร์ ที่อยู่ภายใน จะทำให้ 8048 กระโดดไปยัง ISR อีกโปรแกรมหนึ่งที่ตำแหน่งแอดเดรส ๑๐๐๐๗ สำหรับรายละเอียดเกี่ยวกับตัวไทเมอร์/เควนเตอร์นี้ จะกล่าวถึงครั้งในภายหลัง

ถ้าจะเปรียบเทียบไปแล้ว ISR ก็มีลักษณะคล้ายกับ โปรแกรมย่อย (SUBROUTINE) ทั่วๆ ไป ยกเว้นอยู่จุดหนึ่ง คือ ที่ตอนจบของ ISR นั้น จะต้องลงท้ายด้วยคำสั่ง RETURN พิเศษที่เรียกว่า RETR ซึ่งต่างจากคำสั่ง RET ธรรมดาตรงที่ เมื่อทำคำสั่ง RETR นั้นนอกจากจะนำค่าแอดเดรสจากแอสตไปเก็บไว้ในโปรแกรมเคาน์เตอร์แล้วยังจะเอาค่า PSW ที่เก็บในแอสต คืนกลับที่เดิมของมันด้วย

ในขณะที่ 8048 กำลังรันโปรแกรมใน ISR อยู่ นั้น จะดีสเอเบิ้ลอินเทอร์รัทท์โดยอัตโนมัติ จึงทำให้ไม่สามารถมีการเรียกอินเทอร์รัทท์ซ้ำได้อีก จนกระทั่งเมื่อ 8048 พบกับคำสั่ง RETR ซึ่งเป็นคำสั่งปิดท้ายโปรแกรม ISR แล้ว 8048 จะได้รับการอินาเบิ้ลอินเทอร์รัทท์ อีกครั้งหนึ่ง หลังจากเสร็จ RETR แล้ว 8048 จะได้รับการอินาเบิ้ลอินเทอร์รัทท์อีกครั้งและก็จะกลับไปทำงานของตนตามปกติต่อไป ในระหว่างนี้อาจจะเกิดปัญหาขึ้นได้ เมื่อสัญญาณอินเทอร์รัทท์ที่ส่งมาในตอนแรกนั้นยังคงแอกทีฟอยู่ แม้จะจบโปรแกรม ISR แล้วก็ตาม จะทำให้ 8048 ย้อนกลับไปทำโปรแกรม ISR อีกครั้งหนึ่งทันที ที่จบคำสั่ง RETR อันเป็นความผิดพลาดที่เราไม่ต้องการ

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. คอยตรวจสอบดูก่อนว่า สัญญาณอินเทอร์รัพท์ยังคงแอกทีฟอยู่หรือไม่ ถ้าขา INT ยังมีลอจิกศูนย์ ก็จะทำให้รออยู่ภายใน ISR ก่อน จนกระทั่งเมื่อขา INT เป็นลอจิกหนึ่งจึงจะออกจาก ISR ไปได้ การตรวจสอบสถานะลอจิกที่ขา INT นั้น เราสามารถทำได้ด้วยคำสั่งกระโดดแบบมีเงื่อนไข JMI ซึ่งจะกระโดดเมื่อขา INT เป็นลอจิกศูนย์การที่เราสามารถใช้คำสั่ง JMI นี้ได้ เพราะ ขาสัญญาณ INT นั้นสามารถทำหน้าที่ได้สองลักษณะ คือ เมื่อ 8048 ได้รับการอินาเบิลอินเทอร์รัพท์ ขา INT จะใช้เป็นอินพุทของการอินเทอร์รัพท์ เช่นเมื่ออยู่ใน ISR เราจะสามารถใช้ขา INT เป็นขาอินพุทธรรมดาเช่นเดียวกับ ขา T₀ หรือ T₁ ได้

2. การส่งสัญญาณตอบรับ (ACKNOWLEDGE) จาก 8048 ย้อนกลับไปบอกอุปกรณ์ที่ขออินเทอร์รัพท์ว่าขณะนี้ 8048 รับรู้ไว้แล้ว ให้ถอนอินเทอร์รัพท์ออกไปได้ เนื่องจาก 8048 ไม่มีขาสัญญาณสำหรับการตอบรับอินเทอร์รัพท์ที่อยู่เลย ดังนั้นเราจำเป็นต้องเป็นผู้จัดให้เองโดยอาจจะกำหนดเอาบิตใดบิตหนึ่งของพอร์ต 1 หรือพอร์ต 2 มาใช้แทนก็ได้

3. การออกแบบให้วงจรภายนอกส่งสัญญาณอินเทอร์รัพท์ในลักษณะพัลส์ โดยที่ช่วงแอกทีฟของสัญญาณจะต้องไม่เกินกว่าเวลาที่ใช้ในการเอ็ทซีคิวท์ โปรแกรม ISR จนจบ และในขณะเดียวกัน ก็จะต้องมีช่วงกว้างไม่น้อยกว่า 2 แมกซีไซเคิลเพราะ 8048 จะตรวจสอบสัญญาณลอจิกที่ขา INT ทุก ๆ แมกซีไซเคิล ในช่วงที่ ALE แอกทีฟ

การอินาเบิลอินเทอร์รัพท์ก็คือ การยอมให้ 8048 รับรู้อินเทอร์รัพท์ต่างๆ ได้เราสามารถส่งการอินาเบิลตัวมันเองได้โดยตรงด้วยคำสั่ง ENI นอกจากนี้ 8048 ก็ยังสามารถอินาเบิลตัวมันเองโดยอัตโนมัติ เมื่อกระทำคำสั่ง RETR ด้วยเช่นกัน ดังที่กล่าวมาแล้ว

ส่วนการดีสเอเบิลอินเทอร์รัพท์นั้น ทำได้สามวิธีด้วยกัน เป็นแบบอัตโนมัติเสียสองวิธี ดังนี้

1. ใช้คำสั่ง DIS I ในการดีสเอเบิลโดยตรง
2. โดยการรีเซท 8048 จะทำให้อินเทอร์รัพท์ ถูกดีสเอเบิลโดยอัตโนมัติ
3. ในขณะที่ 8048 กำลังอยู่ในโปรแกรม ISR 8048 จะถูกดีสเอเบิลอินเทอร์รัพท์เช่นกัน เพื่อป้องกันการอินเทอร์รัพท์ซ้ำซ้อน

2.6 ไทเมอร์/เคาน์เตอร์ (TIMER/COUNTER)

ลักษณะของตัวเคาน์เตอร์ เป็นแบบนับขึ้น มีขนาด 8 บิต สามารถตั้งโปรแกรมให้เริ่มนับที่ค่าใดค่าหนึ่งก็ได้ โดยค่าตัวเลขเริ่มต้นจะถูกเก็บไว้ในรีจิสเตอร์ที่มีชื่อว่า T เราสามารถเปลี่ยนแปลงค่าในรีจิสเตอร์ T หรือนำค่าใน T ออกมาคูณก็ได้ โดยใช้คำสั่ง MOV T,A และ MOV A,T

ตามลำดับ

เมื่อเราสั่งให้ไทเมอร์/เคาน์เตอร์เริ่มทำงาน โดยใช้คำสั่ง START ค่าในรีจิสเตอร์ T จะถูกโหลดเข้าไปในไทเมอร์/เคาน์เตอร์ (T/CNT) แล้วเริ่มการนับสัญญาณ (สัญญาณนาฬิกาที่ใช้นับนั้นอาจมาจากภายในหรือภายนอกก็ได้ และเพิ่มค่าใน T/CNT เรื่อยๆ จนเมื่อนับจากค่า ๑๐๐ T/CNT ก็ส่งสัญญาณอินเทอร์รัพท์ภายในไปยังซีพียู และขณะเดียวกันก็จะเซตค่าไทเมอร์แฟลกให้เป็นลอจิกหนึ่ง

สำหรับไทเมอร์แฟลกนั้น เราสามารถตรวจสอบได้ด้วยคำสั่งกระโดดอย่างมีเงื่อนไข JTF และเมื่อซีพียูพบกับคำสั่งแล้วซีพียูก็จะเคลียร์แฟลกนี้ให้เป็นศูนย์ นอกจากนี้การรีเซทก็จะทำให้ไทเมอร์แฟลกเป็นศูนย์ได้เช่นกัน

หลังจากที่ T/CNT นับถึง ๑๐๐ แล้วก็ตาม แต่ T/CNT ก็จะไม่หยุดหน้าที่ของตน กล่าวคือ จะเริ่มต้นนับใหม่ โดยการโหลดเอาค่าเริ่มต้นจากรีจิสเตอร์ T อีกเช่นเคย การนับจะเป็นอย่างต่อเนื่องเช่นนี้ จนกว่า ซีพียูจะพบกับคำสั่ง STOP หรือเมื่อ 8๐48 ถูกรีเซทใหม่ การนับจึงจะหยุดลงได้ และเป็นเพียงการหยุดนับเท่านั้นค่าในรีจิสเตอร์ T จะไม่เปลี่ยนแปลง เราสามารถจะให้เริ่มนับได้โดยใช้คำสั่ง START ได้เช่นเดิม

เมื่อ T/CNT ส่งสัญญาณไปอินเทอร์รัพท์ซีพียู ซีพียูจะต้องตรวจสอบตัวเองก่อนว่า อินาเบิ้ลอินเทอร์รัพท์ไว้แล้วหรือยัง และขณะนั้นจะต้องไม่มีการขออินเทอร์รัพท์จากภายนอก (ทางขา INT) ด้วย ซีพียูจึงจะสามารถตอบรับการเรียกจาก T/CNT ได้ แต่ถ้าเกิดมีอินเทอร์รัพท์พร้อมกันทั้งภายในและภายนอก ซีพียูได้ถูกออกแบบให้ถือการอินเทอร์รัพท์จากภายนอกมีความสำคัญมากกว่าอินเทอร์รัพท์จาก T/CNT ดังนั้นซีพียูจะกระโดดไปยัง ISR ที่แอดเดรส ๑๐๐๐3 เสียก่อน เมื่อเสร็จสิ้นแล้วจึงค่อยกระโดดไปยัง ISR ของ T/CNT ซึ่งเริ่มต้นที่แอดเดรส ๑๐๐๐7

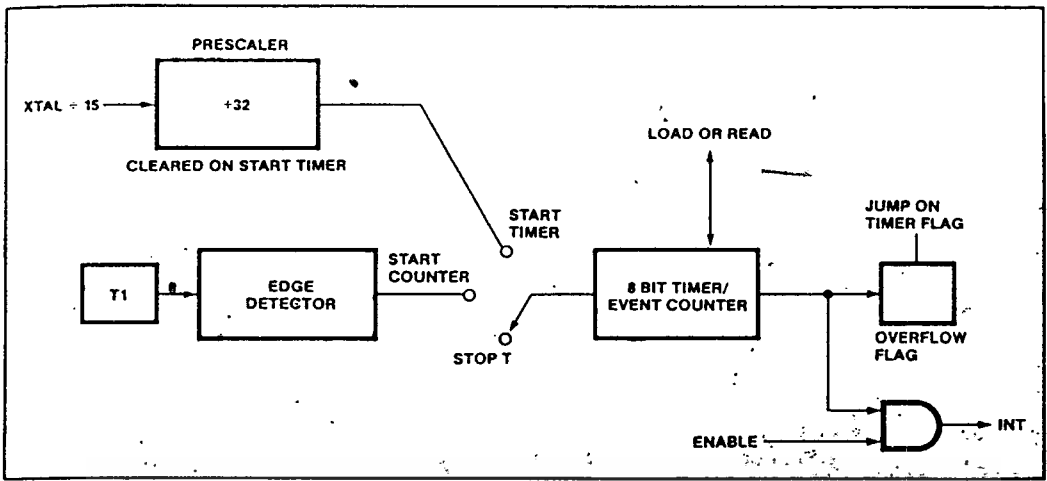
การอินาเบิ้ล และดิสเอเบิ้ลอินเทอร์รัพท์สามารถกระทำได้โดยใช้คำสั่ง EN TCNT1 และ DIS TCNT1 ตามลำดับซึ่งจะเห็นว่า ใช้คำสั่งแยกจากอินเทอร์รัพท์จากภายนอก แต่สำหรับการอินาเบิ้ล และดิสเอเบิ้ล โดยอัตโนมัติในยังคงเหมือนกันคือ

1. เมื่อซีพียูพบกับคำสั่ง RETR ก็จะได้รับการอินาเบิ้ล อินเทอร์รัพท์
2. เมื่อมีการรีเซทหรือขณะอยู่ในโปรแกรม ISR ซีพียูจะถูกดิสเอเบิ้ลอินเทอร์รัพท์โดยอัตโนมัติ

การทำงานของ T/CNT แบ่งออกเป็น สองแบบด้วยกัน คือ

1. ไทเมอร์ โหมด
2. เคาน์เตอร์ โหมด

การเลือกให้ T/CNT เป็นตัวจับเวลา (TIMER) ทำได้โดยใช้คำสั่ง STRT T ซึ่งซีพียูจะ
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และ-31-อ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6.1 แสดงแผนผังของระบบ ไทเมอร์/เดาท์เตอร์

จัดสัญญาณนาฬิกาที่มีความถี่คงที่ป้อนให้แก่ T/CNT สัญญาณดังกล่าว จะมีความถี่เท่ากับ ความถี่ของ คริสตัลหารด้วย 480 เช่น ถ้าใช้คริสตัลที่มีความถี่ 6 MHz เราก็จะได้สัญญาณนาฬิกาที่ป้อนให้ แก่ T/CNT มีความถี่เท่ากับ 12.5 KHz หรือค่าของ T/CNT จะนับเพิ่มขึ้นทุกๆ 80 μ S นั่นเอง

ช่วงเวลาที่ T/CNT สามารถนับได้จะอยู่ในช่วง 80 μ S ถึง 20 ms ถ้าหากเราต้องการ ช่วงเวลาที่ยาวนานกว่านี้ ก็เห็นจะต้องเขียนโปรแกรมให้ช่วยในการนับอีกทอดหนึ่ง เช่น ถ้าเรา ต้องการออกแบบให้ 8048 เป็นนาฬิกา ซึ่งต้องอาศัยฐานเวลาขนาด 1 Hz หรือมีคาบเวลา 1 วินาที เราสามารถออกแบบให้ T/CNT นับเวลาได้นานที่สุด 20 ms ถ้าหากเราเขียนโปรแกรมให้ซีพียูนับจำนวนครั้งที่มันถูกอินเตอร์รัพท์โดย T/CNT เราก็จะทราบว่า เมื่อซีพียูถูกอิน เตอร์รัพท์ครบ 50 ครั้ง จะเป็นเวลาประมาณ 1 วินาที เป็นต้น

ช่วงเวลาที่น้อยที่สุดที่ไทมเมอร์จะทำงานได้ โดยการโหลดค่า FF ใส่ลงในรีจิสเตอร์ T แล้วใช้คำสั่ง STRT T หลังจากนั้นอีก 80 μ S ไทมเมอร์ก็จะส่งสัญญาณมาอินเตอร์รัพท์ซีพียู ถ้า ช่วงเวลา 80 μ s นั้นมากเกินไป ต้องการให้มีความละเอียดมากขึ้นก็สามารถทำได้โดยการป้อน สัญญาณนาฬิกาที่มีความถี่สูงขึ้นเข้ามาทางขา T1 แล้วใช้คำสั่ง STRT CNT สั่งให้ T/CNT ทำหน้าที่เป็นเคาน์เตอร์แทน

ในโหมดของเคาน์เตอร์นั้น เราสามารถนำสัญญาณนาฬิกาจากภายนอกมาป้อนให้ที่อินพุท ของเคาน์เตอร์ได้ และสัญญาณดังกล่าวไม่จำเป็นต้องมี คาบเวลา หรือความถี่แน่นอนเหมือนใน กรณีไทมเมอร์โหมด เราอาศัยลักษณะขอบขาลง (HIGH TO LOW TRANSITION) ของสัญญาณ อินพุท T₁ เป็นตัวเพิ่มค่าของเคาน์เตอร์ สัญญาณดังกล่าวนี้ จะต้องมีลักษณะสำคัญ 2 อย่าง คือ มีช่วงเวลาที่เป็นลอจิกหนึ่งนานมากกว่า 500 ns และจะต้องมีความถี่น้อยกว่าความถี่ของ ALE หารด้วย 3 นั่นคือ ถ้าเราใช้คริสตัลขนาด 6 MHz จะได้ความถี่ของ ALE เป็น 400 KHz และ

สัญญาณที่ป้อนให้แก่แคว้นเตอร์ต้องมีความถี่ไม่เกิน 133 KHz

T/CNT ใน 8048 นั้น ทำงานแบบต่อเนื่องไม่ว่าจะอยู่ในโหมดใด ด้วยเหตุนี้จึงสามารถส่งสัญญาณไปอินเตอร์รัฟท์ซึ่งมีอยู่ได้อย่างสม่ำเสมอ จนกว่าเราจะสั่งให้ T/CNT หยุดทำงาน โดยใช้คำสั่ง STOP TCNT หรือโดยการรีเซทระบบของ 8048 ก็ได้

2.7 ขาอินพุตสำหรับการทดสอบ

ในตัวของ 8048 นั้น นอกจากจะมีพอร์ท 2 พอร์ทสำหรับติดต่อกับภายนอกแล้ว ยังมีขาสัญญาณบางขาที่ถูกออกแบบให้เป็นอินพุตขนาดหนึ่งบิตได้ ขาสัญญาณเหล่านี้ได้แก่ T_0 , T_1 และ INT ขา T_0 สามารถเป็นได้ทั้งอินพุตและเอาต์พุต โดยปกติแล้ว ขา T_0 จะเป็นอินพุต ที่เราสามารถตรวจสอบลอจิกของ T_0 ได้ด้วยคำสั่ง JNT0 หรือ JTO เมื่อเราต้องการให้ T_0 เป็นเอาต์พุตจะใช้คำสั่ง ENT0 CLK เป็นการสั่ง 8048 ให้เอาต์พุตสัญญาณนาฬิกาที่เป็นฐานเวลาของระบบ (มีความถี่เท่ากับ ความถี่ของคริสตัลหารด้วย 8) ออกมาทางขา T_0 นี้

เมื่อ T_0 กลายเป็นเอาต์พุตไปแล้วเราจะไม่สามารถใช้คำสั่งใดๆ เพื่อเปลี่ยน T_0 กลายเป็นเอาต์พุตไปแล้ว เราจะไม่สามารถใช้คำสั่งใดๆ เพื่อเปลี่ยน T_0 ให้กลายเป็นอินพุตอีกครั้งหนึ่ง นอกจากจะอาศัยการรีเซท 8048 เท่านั้น

ขา T_1 เราใช้เป็นขาอินพุตของสัญญาณนาฬิกาที่ป้อนให้แก่แคว้นเตอร์ตามที่กล่าวมาแล้ว นอกจากนั้น เรายังสามารถใช้ T_1 เป็นขาอินพุตธรรมดาได้เช่นกัน และสามารถตรวจสอบลอจิกของสัญญาณที่ขา T_1 ได้ ด้วยคำสั่ง JNT1 และ JT1

ขา INT เป็นขาสัญญาณอินพุตเท่านั้น แต่สามารถจะเลือกใช้ขานี้ในฐานะที่เป็นขาอินเตอร์รัฟท์ หรือขาอินพุตทดสอบธรรมดา

ในกรณีที่อินเตอร์รัฟท์ถูกอินาเบิ้ล ขา INT ก็จะเป็นขาอินพุตสำหรับการขออินเตอร์รัฟท์ตรงกันข้ามถ้าอินเตอร์รัฟท์ถูกดีสเอเบิ้ล ขา INT ก็จะเป็นขาอินพุตธรรมดาที่เราสามารถตรวจสอบลอจิกที่อินพุตนี้ได้ด้วยคำสั่ง JNI

2.8 ชุดคำสั่งของ 8048

แบ่งชุดคำสั่งออกเป็นกลุ่ม ๆ เพื่อให้ง่ายแก่การศึกษา ดังนี้

1. กลุ่มคำสั่ง ควบคุม
2. กลุ่มคำสั่ง เคลื่อนย้ายข้อมูล
3. กลุ่มคำสั่ง อินพุต เอาต์พุต
4. กลุ่มคำสั่ง เกี่ยวกับแอดเด็ควูลเตอเรอร์

5. กลุ่มคำสั่ง เกี่ยวกับโปรแกรมย่อย
6. กลุ่มคำสั่ง เกี่ยวกับการกระโดด
7. กลุ่มคำสั่ง เกี่ยวกับโปรแกรมย่อย
8. กลุ่มคำสั่ง เกี่ยวกับแฟล็ก

กลุ่มคำสั่งควบคุม ได้แก่

- คำสั่งฮีนาเบิ้ล และดิสเอเบิ้ลอินเตอร์รัพท์ (EN I และ DIS I)
- คำสั่งเลือกแบริจของรีจิสเตอร์ (SEL RB₀ หรือ SEL RB₁)
- คำสั่งเลือกแบริจของหน่วยความจำ. (SEL MB₀ หรือ SEL MB₁)
- คำสั่งเอาที่พุกสัญญาณนาฬิกาของระบบออกมาทางขา T₀ (ENT0 CLK)
- คำสั่ง NO OPERATION (NOP)

กลุ่มคำสั่งเคลื่อนย้ายข้อมูล ได้แก่

- คำสั่งเคลื่อนย้ายข้อมูลระหว่างรีจิสเตอร์ หรือ หน่วยความจำข้อมูลภายในกับแอดคิวมูลเตอร์ (MOV)
- คำสั่งเคลื่อนย้ายข้อมูลระหว่างแอดคิวมูลเตอร์กับหน่วยความจำข้อมูลภายนอก (MOVX)
- คำสั่งอ่านข้อมูลจากหน่วยความจำโปรแกรมที่อยู่ในเพจเดียวกัน (MOVP) และ คำสั่งอ่านข้อมูลจากหน่วยความจำเพจ 3 (MOVP3)
- คำสั่งแลกเปลี่ยนข้อมูลระหว่างแอดคิวมูลเตอร์กับรีจิสเตอร์ หรือกับหน่วยความจำข้อมูลภายใน (XCH) และคำสั่งแลกเปลี่ยนข้อมูลระหว่างแอดคิวมูลเตอร์กับหน่วยความจำข้อมูลภายใน โดยแลกเปลี่ยนเฉพาะข้อมูล 4 บิตล่างเท่านั้น (XCHD)

กลุ่มคำสั่งเคลื่อนย้ายข้อมูล ได้แก่

- คำสั่งอ่านข้อมูลหรือเขียนข้อมูลระหว่าง แอดคิวมูลเตอร์กับพอร์ท 1 หรือ พอร์ท 2 (IN และ OUT)
- คำสั่งอ่าน, เขียนข้อมูลระหว่าง แอดคิวมูลเตอร์ กับ ดาต้าบัส (INS และ OUTL)
- คำสั่งเกี่ยวกับการ AND หรือการ OR ข้อมูลกับข้อมูลเดิมในพอร์ท 1 หรือ พอร์ท 2 หรือ ดาต้าบัส (ANL และ ORL)
- คำสั่งเกี่ยวกับพอร์ทส่วนขยายที่ใช้ไอซีเบอร์ 8243 อันได้แก่ คำสั่งอ่าน, เขียนข้อมูลกับพอร์ทนี้ และคำสั่ง AND, OR ข้อมูลกับพอร์ทนี้ (MOVD, ANLD, ORLD)

ตัวอักษร D ที่ห้อยท้ายนั้นหมายถึง การกระทำเกี่ยวกับข้อมูลเฉพาะ 4 บิตล่างเท่านั้น

กลุ่มคำสั่งเกี่ยวกับแอดคิวมูลเตอร์ ได้แก่

- คำสั่งเกี่ยวกับการบวกตัวเลขแบบไม่คิดตัวทดและแบบที่คิดตัวทดด้วย (ADD และ ADDC)
- คำสั่งเกี่ยวกับการกระทำทางลอจิกคือ การ AND, OR, EXCLUSIVE OR, COMPLEMENT และ CLEAR (ANL, ORL, XRL, CPL และ CLR ตามลำดับ)
- คำสั่งเกี่ยวกับการหมุน และการเลื่อนบิต
- คำสั่งเพิ่มหรือลดค่าของแอดคิวมูลเตอร์ หรือรีจิสเตอร์ หรือหน่วยความจำข้อมูลภายใน (INC และ DEC)
- คำสั่งปรับค่าตัวเลขในฐานะสิบ DECIMAL ADJUST (DA)
- คำสั่งสลับค่าข้อมูลในแอดคิวมูลเตอร์ระหว่าง 4 บิตล่าง กับ 4 บิตบน (SWAP)

กลุ่มคำสั่งเกี่ยวกับ ไทเมอร์/เคาน์เตอร์ ได้แก่

- คำสั่งเคลื่อนย้ายข้อมูลระหว่างแอดคิวมูลเตอร์ กับรีจิสเตอร์ T (MOV)
- คำสั่งเริ่ม หรือ หยุดการทำงานของ ไทเมอร์/เคาน์เตอร์ (STRT และ STOP)
- คำสั่งอินทเอบีล และดิสเอบีล ไทเมอร์อินเตอร์รัทท์ (EN TCNT1 และ DIS TCNT1)

กลุ่มคำสั่งเกี่ยวกับการกระโดด ได้แก่

- คำสั่งกระโดด (JMP)
- คำสั่งกระโดดทางอ้อม ภายในเพจเดียวกัน (JMPP)
- คำสั่งตรวจสอบขาอินพุททดสอบ T₀, T₁, INT (JTO, JNTO, JT1, JNT1 และ JN1 ตามลำดับ)
- คำสั่งตรวจสอบแฟลกต่าง เช่น แฟลกตัวทด, ซอฟท์แวร์แฟลก (JC, JNC, JFO และ JF1)
- คำสั่งทดสอบค่าในแอดคิวมูลเตอร์ว่าเป็นศูนย์หรือไม่ (JZ และ JNZ)
- คำสั่งเกี่ยวกับการวนลูป โดยการลดค่ารีจิสเตอร์ลงหนึ่ง แล้วตรวจสอบว่าเป็นศูนย์หรือไม่ ถ้าค่าในรีจิสเตอร์ไม่เท่ากับศูนย์ ก็จะกระโดดไป (DJNZ)

กลุ่มคำสั่งเกี่ยวกับโปรแกรมย่อย ได้แก่

- คำสั่งเรียกโปรแกรมย่อย (CALL)
- คำสั่ง RETURN มีสองคำสั่งด้วยกันคือ RET และ RETR

กลุ่มคำสั่งเกี่ยวกับแฟลก ได้แก่

- คำสั่ง CLEAR และ COMPLEMENT แฟลกตัวทด และ ซอฟท์แวร์แฟลก F₀, F₁ (CLR และ CPL)

เอกสารตำราลงในรูปที่ 2.8 นี้ ใช้สำหรับแปลงรหัสนิโมนิกของ 8048 ให้เป็นภาษาเครื่อง
ไม่ทำกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MNEMONIC	HEX	MNEMONIC	HEX	MNEMONIC	HEX	MNEMONIC	HEX
CONTROL		ACCUMULATOR		IN/OUT		TIMER	
EN	I 05	ADD	A,Rr 6X	IN	A,P1 09	MOV	A,T 42
DIS	I 15	ADD	A,@R0 60	OUTL	P1,A 39	MOV	T,A 62
SEL	RB0 C5	ADD	A,@R1 61	ANL	P1,DATA 99	STRT	T 55
SEL	RB1 D5	ADD	A,DATA 03	ORL	P1,DATA 89	STRT	CNT 45
SEL	MB0 E5	ADDC	A,Rr 7X			STOP	TNCT 55
SEL	MB1 F5	ADDC	A,@R0 70	IN	A,P2 0A	EN	TCNT1 25
ENT0	CLK 75	ADDC	A,@R1 71	OUTL	P2,A 3A	DIS	TCNT1 35
		ADDC	A,DATA 13	ANL	P2,DATA 9A		
DATA MOVE		ANL	A,Rr 5X	ORL	P2,DATA 8A	BRANCH	
		ANL	A,@R0 50				
MOV	A,Rr FX	ANL	A,@R1 51	INS	A,BUS 08	JMP	addr P4
MOV	A,@R0 F0	ANL	A,DATA 53	OUTL	BUS,A 02	JMPP	@A B3
MOV	A,@R1 F1	ORL	A,Rr 4X	ANL	BUS,DATA 98	DJNZ	Rr,addr EX
MOV	A,DATA 23	ORL	A,@R0 40	ORL	BUS,DATA 88	JC	addr F6
MOV	Rr,A AX	ORL	A,@R1 41			JNC	addr E6
MOV	@R0,A AD	ORL	A,DATA 43	MOVD	A,Pp 0R	JZ	addr C6
MOV	@R1,A A1	XRL	A,Rr DX	MOVD	Pp,A 3R	JNZ	addr 96
MOV	Rr,DATA EX	XRL	A,@R0 D0	ANLD	Pp,A 9R	JT0	addr 36
MOV	@R0,DATA E0	XRL	A,@R1 D1	ORLD	Pp,A 8R	JNT0	addr 26
XCH	A,Rr 2X	XRL	A,DATA D3			JT1	addr 56
XCH	A,@R0 20	INC	A 17	REGISTER		JNT1	addr 46
XCH	A,@R1 21	DEC	A 07			JF0	addr B6
XCHD	A,@R0 30	CLR	A 27	INC	Rr 1X	JF1	addr 76
XCHD	A,@R1 31	CPL	A 37	DEC	Rr CX	JT1	addr 16
MOV	A,PSW C7	RL	A E7	INC	@R0 10	JNI	addr 86
MOV	PSW,A D7	RLC	A F7	INC	@R1 11	JB0	addr 12
MOVX	A,@R0 80	RR	A 77			JB1	addr 32
MOVX	A,@R1 81	RRC	A 67	SUBROUTINE		JB2	addr 52
MOVX	@R0,A 90	DA	A 57			JB3	addr 72
MOVX	@R1,A 91	SWAP	A 47	CALL	addr P4	JB4	addr 92
MOVVP3	A,@A E3			RET	83	JB5	addr 82
MVOP	A,@A A3	NOP	00	RETR	93	JB6	addr D2
						JB7	addr F2
FLAG							
CLR	C 97	CPL	C A7	CLR	F0 85	CPL	F0 95
CLR	F1 A5	CPL	F1 E5				

X = 8-F FOR REGISTER R0-R7

R = C-F FOR EXPAND PORT P4-P7

P = 0-F FOR INDICATE THE PAGE TO JMP OR CALL TO

PAGE 0 1 2 3 4 5 6 7

JMP P = 0 2 4 6 8 A C E

CALL P = 1 3 5 7 9 B D F

2.9 การต่อขยายหน่วยความจำโปรแกรมภายนอก

ไอซี 74LS373 ใช้สัญญาณ ALE ในการแลทช์ค่าของแอดเดรส ให้แยกออกมาจากดาต้าบัสได้เป็นแอดเดรสบัส A_0-A_7 ส่วนแอดเดรสบัส A_8-A_{11} นั้นได้มาจาก 4 บิตล่างของพอร์ท 2 ($P_{20}-P_{23}$)

ส่วนสัญญาณ PSEN นั้นเป็นสัญญาณที่แสดงให้รู้ว่า 8048 กำลังเพทช์คำสั่งจากหน่วยความจำโปรแกรมภายนอก เราอาจใช้สัญญาณนี้เป็นตัวอินาเบิล (CHIP ENABLE) หน่วยความจำโปรแกรมภายนอกก็ได้

ในส่วนของวงจรไมโครโปรเซสเซอร์ MCS8048 จะรวมส่วนของวงจรแปลงอนาลอกเป็นดิจิทัล (AD Converter) โดยใช้ IC ADC0809 ส่วนนี้จะทำหน้าที่ปรับความถี่โดยการปรับที่ตัวความต้านทาน R ซึ่งจะทำให้สะดวกมากในการปรับความถี่ และ วงจรหาร 10 โดยใช้ IC7490 ในส่วนวงจรหาร 10 นี้ เพื่อที่จะลดความถี่นาฬิกา (CLOCK) ที่ป้อนให้ ADC0809 เพราะความถี่นาฬิกาที่ออกมาจากขา T_0 โดยใช้คำสั่ง ENT_0 CLK นี้ จะเท่ากับความถี่ในหนึ่งไซเคิลการทำงานซึ่งเท่ากับ ความถี่ของคริสตอลหาร 3 ($f_c = f_{xtal}/3$) ในที่นี้จะใช้ $f_{xtal} = 6\text{MHz}$ ดังนั้น $f_c = 2\text{MHz}$ นั้นจะมีค่ามากกว่าความถี่สูงสุดที่ ADC0809 จะรับได้ ดังนั้นเมื่อผ่านวงจรหาร 10 แล้ว $f_c = 200\text{ KHz}$ ดังนั้น ADC0809 ก็จะสามารถรับความถี่นี้ได้

อีกส่วนหนึ่งที่สำคัญหลังจากต่อขยายระบบ MCS 8048 แล้วก็คือ ส่วนของแลทช์ (LATCH) ข้อมูลที่ส่งออกมายังพอร์ท 2 ทาง 4 บิตล่าง ซึ่งจำเป็นอย่างยิ่งที่จะต้องใช้อุปกรณ์แลทช์ข้อมูลไว้ เพราะข้อมูลที่ออกมาจากพอร์ท 2 ทาง 4 บิตล่าง นั้น จะถูกมัลติเพลกซ์กับสัญญาณแอดเดรส A_8-A_{11} ในวงจรนี้จะใช้ IC74174 และที่สำคัญคือ นำสัญญาณรีเซทไปต่อที่ขาเคลียร์ของ IC74174

บทที่ 3

การคำนวณสร้างสัญญาณมอดูเลตตามความกว้างของพัลส์ (PWM)

การสร้างสัญญาณที่มอดูเลตตามความกว้างของพัลส์ (PWM) ในโครงงานนี้ ใช้วิธีการควบคุมอัตราส่วนของค่าความต่างศักย์ต่อความถี่ให้คงที่ (V/f คงที่) ซึ่งได้จากข้อสรุปในบทที่ 1 ที่ว่า "ไม่ว่าความถี่จะมีค่าเท่าไร ขอเพียงแต่ให้พื้นที่ใต้เส้นโค้งแรงดันไฟฟ้ามีค่าคงที่อยู่เสมอแล้วมอเตอร์เหนี่ยวนำจะสามารถทำงานที่แรงหมุนสูงเหมาะสมที่สุด" การสร้างสัญญาณมอดูเลตตามความกว้างของพัลส์ (PWM) ใช้ CPU เบอร์ MCS 8048 เป็นตัวสร้างสัญญาณโดยให้ค่าพื้นที่ใต้เส้นโค้งแรงดันนั้นคงที่ แล้วเพิ่มค่าความกว้างของพัลส์ โดยเพิ่มความยาวในส่วนที่แรงดันเป็นศูนย์ เพื่อเป็นการปรับความถี่ ดังรูป 3.1

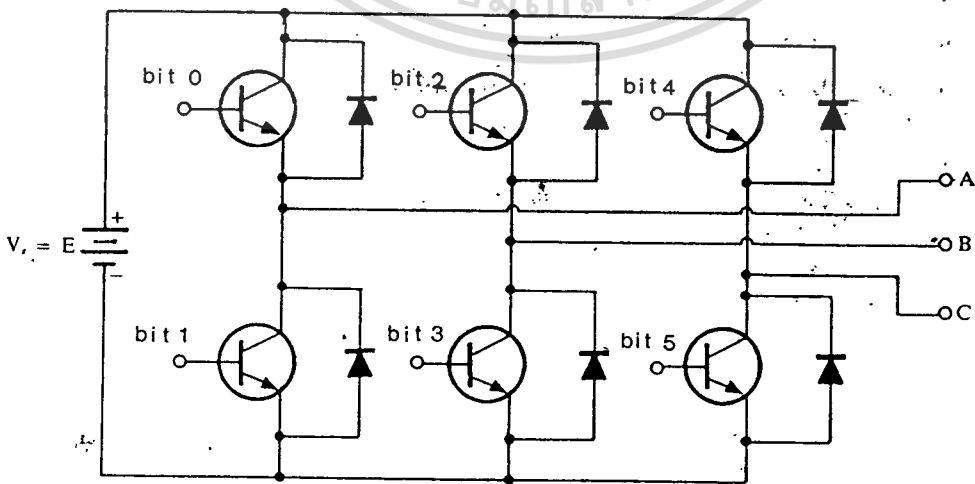


รูปที่ 3.1 สัญญาณมอดูเลตตามความกว้างของพัลส์

h : เป็นค่าคงที่ หนึ่งช่วงเวลาเพื่อให้เกิดพัลส์ที่มีพื้นที่ใต้แรงดันคงที่

x : ตัวแปรเปลี่ยนความถี่ มีค่าตั้งแต่ 1-256

ความกว้างของพัลส์จะถูกเพิ่มในส่วนที่เป็นค่า x คือการเพิ่มความยาวในส่วนที่แรงดันที่เป็นศูนย์นั่นเอง สัญญาณพัลส์ที่สร้างขึ้นมีทั้งหมด 6 สัญญาณ จึงจัดสัญญาณแต่ละสัญญาณออกเป็นแต่ละบิตแยกจากกัน ซึ่งได้จาก port 2 ของ MCS 8048 ซึ่งมีถึง 8 บิต โดยให้สัญญาณแต่ละบิตเป็นสัญญาณต่าง ๆ ดังนี้



รูปที่ 3.2 การนำสัญญาณแต่ละบิตที่ออกจากไมโครโปรเซสเซอร์มาไบออสควาว์เวอร์ที่สร้างขั้วมอเตอร์

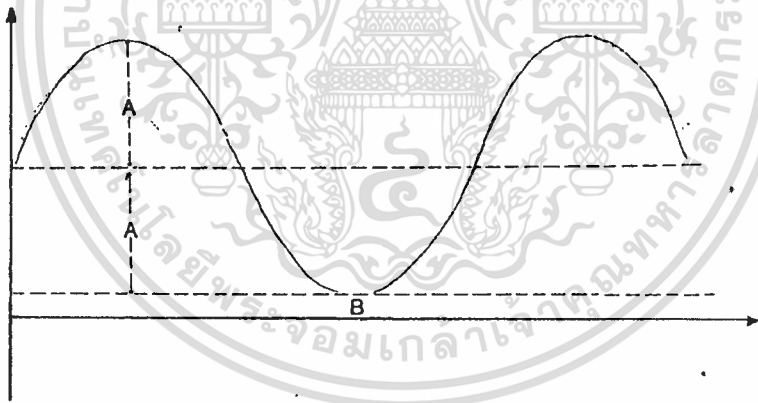
อีกสำรณเป็นอีกสำรณหนึ่งหรือการเรียงในทิศทางที่ขอ แต่เพิ่ม เมื่อผู้จัดทำหนังสือฉบับนี้ขึ้นการคำ

ไม่จำกัดใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- บิต 0 เป็นสัญญาณของเฟส U
- บิต 1 เป็นสัญญาณตรงข้ามของเฟส U
- บิต 2 เป็นสัญญาณของเฟส V
- บิต 3 เป็นสัญญาณตรงข้ามของเฟส V
- บิต 4 เป็นสัญญาณของเฟส W
- บิต 5 เป็นสัญญาณตรงข้ามของเฟส W

สัญญาณแต่ละบิตนำไปไบแอสให้ที่ขา B ของพาวเวอร์ทรานซิสเตอร์ของวงจร Inverter ดังรูป 3.2 ดังนั้นจากหลักการของการสวิทชิงทรานซิสเตอร์ของอินเวอร์เตอร์นั้น เราสามารถนำมาสร้างสัญญาณพัลส์ทั้ง 6 บิตได้ โดยการสร้างสัญญาณพัลส์จะอธิบายในช่วงหลังอย่างละเอียด แต่ก่อนจะสร้างสัญญาณพัลส์นั้น เราจะต้องยกระดับโวลท์เตจของไฟเฟส U, V และ W เป็น

$$\begin{aligned}
 U &= A \sin(\omega t) + A + B \\
 V &= A \sin(\omega t + 120^\circ) + A + B \\
 W &= A \sin(\omega t + 240^\circ) + A + B
 \end{aligned}$$



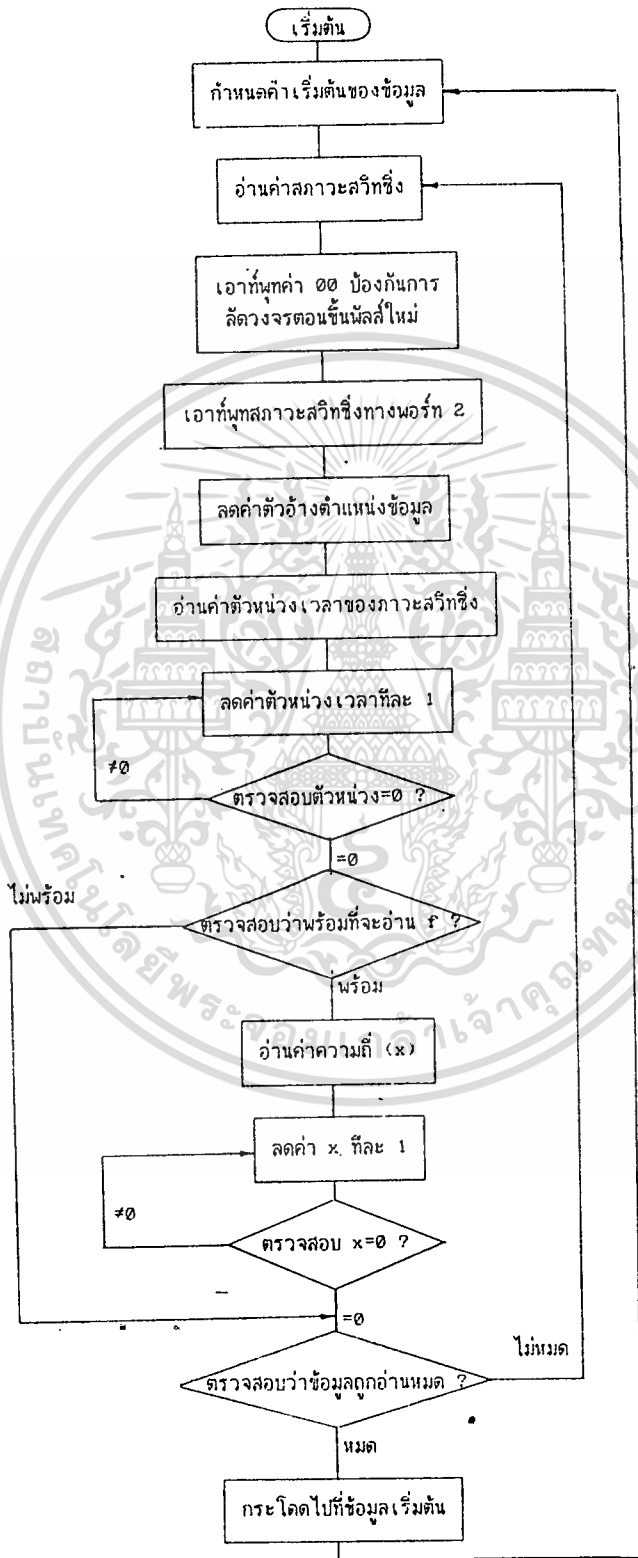
รูปที่ 3.3 ลักษณะรูปคลื่นสามเฟสยกระดับแล้ว

แล้วค่า A และ B ก็หาได้จาก การหาค่า h เสียก่อน ซึ่งจะอธิบายต่อไป ในโครงการนี้ ได้ทำการสร้างสัญญาณที่มีอูเลตตามความกว้างของพัลส์ไว้ถึง 4 แบบ คือ แบบที่มีจำนวนพัลส์ 30 พัลส์, 24 พัลส์, 18 พัลส์ และ 12 พัลส์ใน 1 ไซเคิล

3.1 การคำนวณค่าตัวแปร h ของ PWM 30 พัลส์

การคำนวณหาค่า h หรือค่าตั้งหน่วงเวลาเพื่อประโยชน์ในการสร้างข้อมูลต่อไปนั้นสามารถคำนวณได้จากการนับจำนวนครั้งของแต่ละคำสั่ง (Instruction) ที่ CPU ได้ทำการประมวลผลในการสร้างสัญญาณ PWM ออกมา 1 พัลส์ โดยที่ตัดตัวแปร x, h ไว้ (x : ตัวแปรเปลี่ยนความถี่, h : ตัวแปรหน่วงเพื่อให้เกิดสัญญาณพัลส์ที่มีพื้นที่ใต้แรงดันคงที่) เมื่อนับจำนวนครั้งของการคำนวณได้ค่า h แล้วก็นำค่า h ไปหารด้วยค่า x จะได้ค่า h/x ซึ่งเป็นค่าตั้งหน่วงเวลาที่ต้องการใช้ ในการสร้างสัญญาณ PWM

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประมวลผลของแต่ละคำสั่งได้แล้ว จึงนำมาคูณกับจำนวนไซเคิลของแต่ละคำสั่ง (แต่ละคำสั่งจำนวนไซเคิลในการทำงานไม่เท่ากันมี 1-2 ไซเคิล) จะได้เป็นจำนวนไซเคิลทั้งหมดที่ใช้สร้างพัลส์ 1 พัลส์ จากนั้น เราก็คูณด้วย 30 ซึ่งเป็นจำนวนพัลส์ที่เราต้องการสร้างภายใน 1 คาบ (โดยการออกแบบเราจะใช้ $f_{xt} = 6 \text{ MHz}$ ให้ f_{max} (ความถี่ของพัลส์ที่สร้างสูงสุด) = 80 Hz f_{min} เมื่อ $x = 1$, f_{min} เมื่อ $x = 256$) ก็จะสามารถหาค่า h ได้ดังตัวอย่างการนับครั้งของคำสั่งที่ทำไปภายในช่วงเวลาที่สร้างพัลส์ 1 พัลส์

ADDRESS	LABEL	INSTRUCTION	CYCLE	จำนวนครั้ง ใน 1 พัลส์	จำนวน CYCLE ใน 1 พัลส์
100		MOV R6,04H	2		
102	ADD5	MOV R0,FOH	2		
104	ADD1	MOV 4,R0	1	4	4
105		MOV R3,A,@A	2	4	8
106		MOV R5,A	1	4	4
107		ANL P2,00H	2	4	8
109		MOV A,R5	1	4	4
10A		OUTL P2,A	2	4	8
10B		DEC R0	1	4	4
10C		MOV A,R0	1	4	4
10D		MOV R3,A,@A	2	4	8
10E	ADD2	DEC A	1	h	h
10F		JNZ ADD2	2	h	2h
111		DJNZ R6,ADD3	2	4	8
113		MOV R6,04H	2	1	2
115		ORL P1,FFH	2	1	2
117		IN A,P1	2	1	2
118	ADD4	DEC A	1	x	x
119		JNZ ADD4	2	x	2x
11B	ADD3	DJNZ R0,ADD1	2	4	8
11D		JMP1 ADD5	2		

LOOP อ่าน
ความถี่ และหน่วยเวลา

$$\text{ผลรวมของจำนวนไซเคิลภายใน 1 พัลส์} = 3x + 3h + 74$$

$$\text{ผลรวมของจำนวนไซเคิลภายใน 1 คาบ} = 30(3x + 3h + 74) + 4$$

หมายเหตุ ที่บวก 4 เพิ่มเข้าไปเนื่องจากในกรณีที่ดาต้าอ่านหมดแล้วจะวนกลับมาอ่านใหม่ที่แอด

เดรส F0 โดยเข้าคำสั่ง JMPP ADD5 และคำสั่ง MOV R0,FOH จึงเพิ่มเข้าไป 4 ไซเคิลในการดำเนินการ
ไม่ทำกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้จำนวนไซเคิลการทำงานของ CPU MCS 8048 ครบทั้งหมดภายใน 1 คลื่นมอดูเลต ตามความกว้างของพัลส์ PWM 30 พัลส์ ซึ่งมีค่าเท่ากับ $90(x+h) + 2224$ จะสามารถคำนวณ หาค่าความถี่ของสัญญาณมอดูเลตตามความกว้างของพัลส์ PWM 30 พัลส์ โดย

กำหนดให้ f_c : ความถี่ของสัญญาณนาฬิกา (CLOCK) = ความถี่ของคริสตอล/3

f_{15} : ความถี่ของไซเคิลการทำงานของ CPU MCS 8048 1 ไซเคิล = ความถี่คริสตอล/15

f : ความถี่ของสัญญาณมอดูเลตตามความกว้างของพัลส์ (PWM)

$$f = f_c / (90(x+h) + 2224) \quad (3.1.1)$$

หรือ $f = \text{ความถี่ของคริสตอล} / [15(90(x+h) + 2224)] \quad (3.1.2)$

ต่อไปจึงทำการคำนวณหาค่าตัวแปร h เนื่องจากความถี่ของคริสตอลที่ใช้เป็น 6 MHz และ ต้องการให้ความถี่สูงสุดของสัญญาณมอดูเลตตามความกว้างของพัลส์ (PWM) เป็น 80 Hz เมื่อค่า $x = 1$ ด้วยการแทนค่าในสมการข้างบน จะได้ค่า h และให้ $x = 256$ จะได้

$$f_{\min} = 6 \times 10^6 / [15(90(256 + 29.844) + 2224)]$$

$$f_{\min} = 14.3$$

สัญญาณพัลส์ที่มีความถี่สูงสุดของ PWM 30 พัลส์ เมื่อ $x = 1$, $h = 30$

$$f_{\max} = \text{ความถี่ของคริสตอล} / [15(3(x+h) + 24)]$$

$$= 6 \times 10^6 / [15(3(1+30) + 24)]$$

$$= 2395 \text{ Hz (ประมาณ 2.4 KHz)}$$

การสร้างสัญญาณมอดูเลตตามความกว้างของพัลส์โดยใช้ CPU MCS 8048 ความถี่คริสตอลที่ใช้คือ 6 MHz สามารถปรับความถี่ได้ในช่วง 14.3 Hz - 80 Hz หรือปรับความเร็วรอบมอเตอร์ 4 โพลได้ในช่วงความเร็ว 429-2400 รอบ/นาที เลือกใช้ฟาวเวอร์ทรานซิสเตอร์ที่นำมาเป็นวงจรรีเลย์มอเตอร์ที่มีความถี่ f_{\max} ในการสวิตซ์ ซึ่ง มากกว่า 2.4 KHz

ส่วนการคำนวณค่าตัวแปร h, f_{\min}, f_{\max} ของสัญญาณมอดูเลตตามความกว้างของพัลส์ ที่มีจำนวนพัลส์ 24 ลูก ก็มีวิธีคำนวณ เช่นเดียวกับ สัญญาณมอดูเลตตามความกว้างของพัลส์ที่มีจำนวนพัลส์ 30 ลูก ซึ่งจะได้ค่า $h = 43.7$ (ประมาณ 44) สามารถปรับความถี่ในช่วง 17-80 Hz ถ้านำไปขับมอเตอร์จะสามารถปรับความเร็วรอบได้ในช่วง 513-2400 รอบต่อนาที เลือกใช้ฟาวเวอร์ทรานซิสเตอร์ ที่มีความถี่สูงสุด f_{\max} ในการสวิตซ์ซึ่งมากกว่า 1.9 KHz

ส่วนสัญญาณมอดูเลตตามความกว้างของพัลส์ที่มีจำนวนพัลส์ 18 ลูก นั้นจะได้ค่า $h = 66.85$ (ประมาณค่า 67) สามารถปรับความถี่ได้ในช่วง 21.3 - 80 Hz หรือปรับความเร็วรอบมอเตอร์ 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่จำกัดสิทธิ์ในสิ่งอื่น อีกทั้งห้ามมิให้ดัดแปลงเนื้อที่-42- ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพลได้ในช่วง 639 - 2400 รอบต่อวินาที และใช้พาวเวอร์ทรานซิสเตอร์ที่มีความถี่ในการสวิตชิ่ง f_{sw} มากกว่า 1.4 KHz

ส่วนสัญญาณมอดูเลตตามความกว้างของพัลส์ที่มีจำนวนพัลส์ 12 ลูก นั้นจะได้ค่า $h = 114$ สามารถปรับความถี่ได้ในช่วง 28.2 - 80 Hz หรือปรับความเร็วมอเตอร์ 4 โพล ในช่วงความเร็ว 846 - 2400 รอบต่อวินาที และใช้พาวเวอร์ทรานซิสเตอร์ที่มีความถี่ในการสวิตชิ่ง F_T มากกว่า 0.95 KHz

3.2 การสร้างข้อมล PWM 30 พัลส์

จากกรอกแบบการสร้าง PWM 30 พัลส์ โดยใช้ MCS 8048 โดยปรับความถี่ได้ในช่วง 14.3 - 80 Hz (ปรับความเร็วมอเตอร์ 4 โพล ได้ในช่วง 429 - 2400 รอบต่อวินาที) นั้นจากการคำนวณได้ค่า h (ประมาณค่า 30)

จากสมการไฟ สามเฟส $U = A \sin(\omega t) + A + B$

$$V = A \sin(\omega t + 120) + A + B$$

$$W = A \sin(\omega t - 120) + A + B$$

ค่าแรงดันสูงสุดของ $x = A \sin(90) + A + B$

$$= 2A + B$$

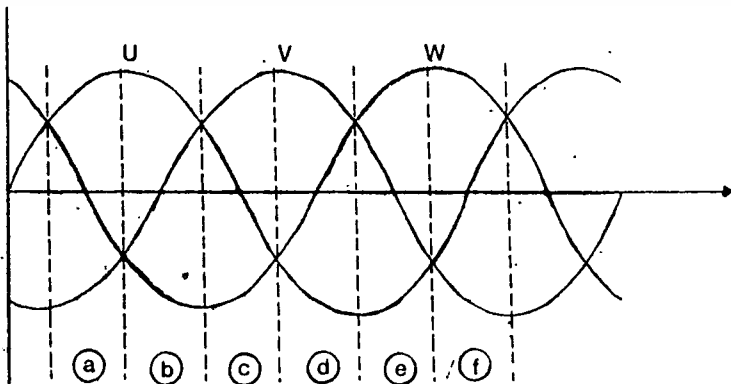
ค่าแรงดันสูงสุดของพัลส์ = $h + 1$ (1 มาจากค่าตัวเลขที่นำมาหาช่วงเวลาเพื่อสร้างความถี่สูงสุด) สามารถคำนวณหา A และ B ได้

$$2A + B = h + 1$$

$$2A + B = 44 + 1$$

ค่าที่เหมาะสม $A = 22$, $B = 1$

หนึ่งคาบเวลาของลูกคลื่น PWM 24 พัลส์ = 360 เพราะฉะนั้นหนึ่งพัลส์จะมีความกว้าง = $360/30 = 6$ แล้วนำรูปชายนี่ 3 เฟสมาแบ่งเป็น 6 ช่วง



รูปที่ 3.2.1 รูปคลื่นชายนี่แบ่งออกเป็น 6 ช่วง

ช่วง a $U > V > W$

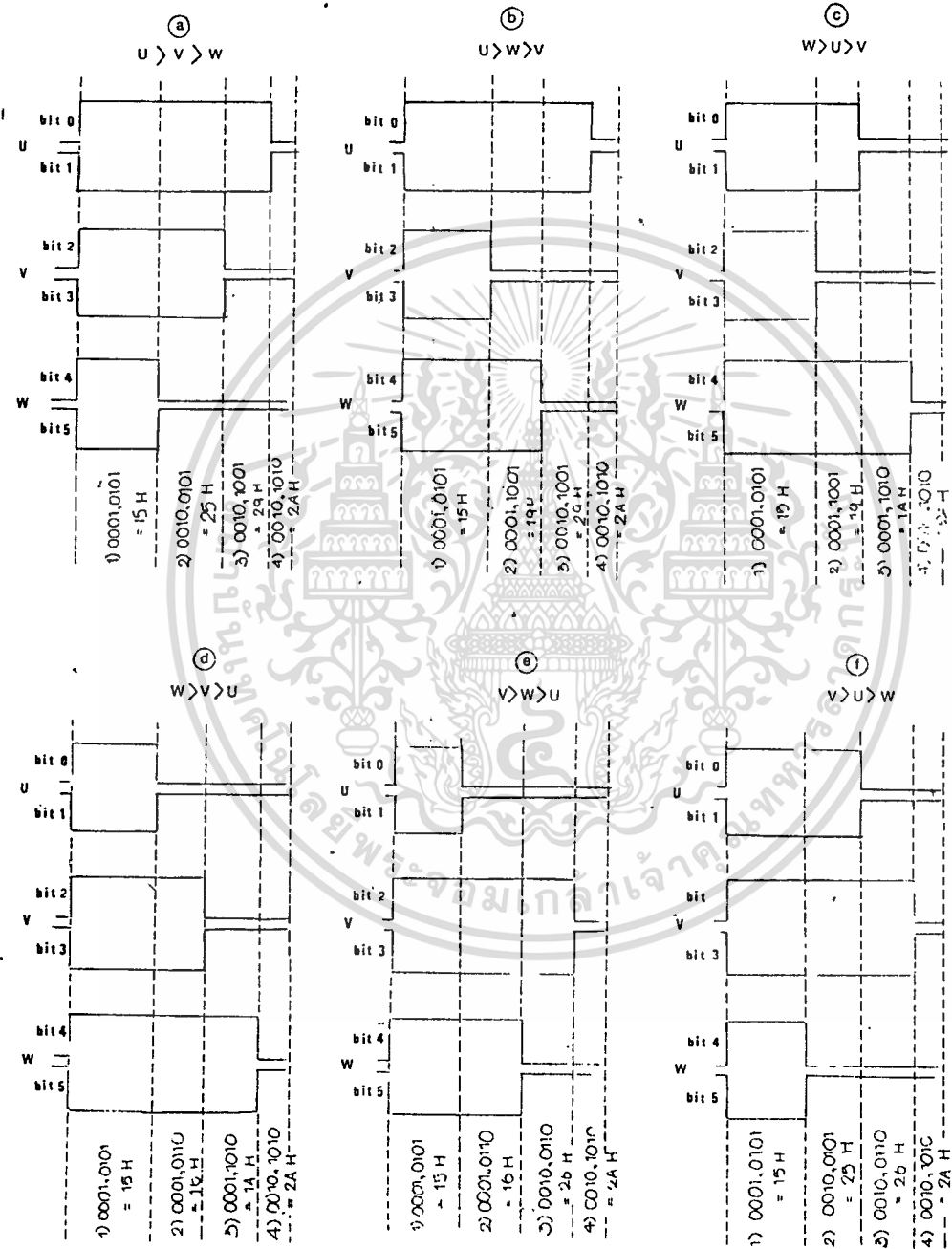
ช่วง b $U > W > V$

ช่วง c $W > U > V$

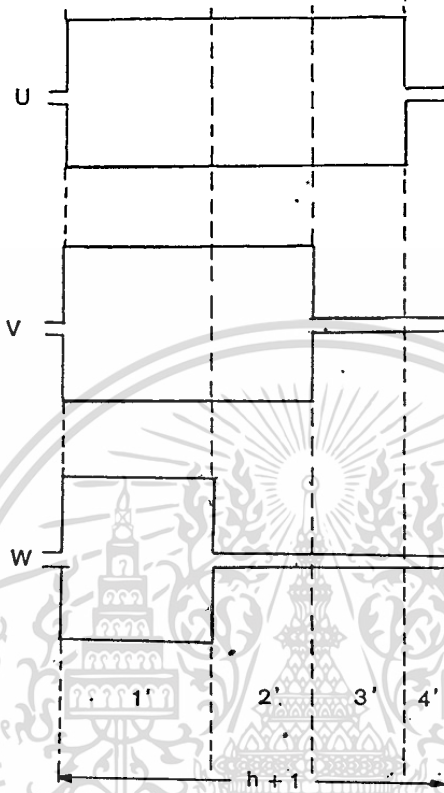
ช่วง d $W > V > U$

ช่วง e $V > W > U$

ช่วง f $V > U > W$



จากรูปแสดงภาวะสวิตซ์ซึ่ง ทั้ง 6 ช่วง สามารถหาค่าตัวหนังสือเวลา $1', 2', 3', 4'$ ได้
 ดังตัวอย่างในรูปที่ 3.2.3 จะได้ว่า



รูปที่ 3.2.3 ภาวะสวิตซ์ในช่วง a) $U > V > W$

ช่วง a) $U > V > W$

$$\begin{aligned} 1' &= W \\ 2' &= V - W \\ 3' &= U - V \\ 4' &= h + 1 - U \end{aligned}$$

ช่วง c) $W > U > V$

$$\begin{aligned} 1' &= V \\ 2' &= U - V \\ 3' &= W - U \\ 4' &= h + 1 - W \end{aligned}$$

ช่วง e) $V > W > U$

$$\begin{aligned} 1' &= U \\ 2' &= W - U \\ 3' &= V - W \\ 4' &= h + 1 - V \end{aligned}$$

ช่วง b) $U > W > V$

$$\begin{aligned} 1' &= V \\ 2' &= W - V \\ 3' &= U - W \\ 4' &= h + 1 - U \end{aligned}$$

ช่วง d) $W > V > U$

$$\begin{aligned} 1' &= U \\ 2' &= U - V \\ 3' &= W - V \\ 4' &= h + 1 - W \end{aligned}$$

ช่วง f) $V > U > W$

$$\begin{aligned} 1' &= W \\ 2' &= U - W \\ 3' &= V - U \\ 4' &= h + 1 - V \end{aligned}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 ตารางข้อมูลสัญญาณมอดูเลตตามความกว้างของพัลส์ 30 พัลส์⁴¹

จากการคำนวณได้ ค่า $h = 30$ แล้ว $A = 15, B = 1$

$$U = 15 \sin(\omega t) + 15 + 1$$

$$V = 15 \sin(\omega t + 120) + 15 + 1$$

$$W = 15 \sin(\omega t - 120) + 15 + 1$$

มุมมองค่า	U		V		W		ภาวะสวิทชิง (SWITCHING STATE)	1'		2'		3'		4'	
	D	H	D	H	D	H		D	H	D	H	D	H	D	H
42	26	1A	21	15	1	01	1) 0001,0101=15H	1	01	20	14	5	05	5	05
54	28	1C	18	12	2	02	2) 0010,0101=25H	2	02	16	10	10	0A	3	03
66	29	1D	14	0E	4	04	3) 0010,1001=29H	4	04	10	0A	15	0F	2	02
78	30	1E	11	0B	6	06	4) 0010,1010=2AH	6	06	5	05	19	13	1	01
90	30	1E	9	09	9	09		9	09	1	01	21	15	1	01
102	30	1E	6	06	11	0E	1) 0001,0101=15H	6	06	5	05	19	13	1	01
114	29	1D	4	04	14	0E	2) 0001,1001=19H	4	04	10	0A	15	0F	2	02
126	28	1C	2	02	18	1E	3) 0010,1001=29H	2	02	16	10	10	0A	3	03
138	26	1A	1	01	21	15	4) 0010,1010=2AH	1	01	20	14	5	05	5	05
150	24	18	1	01	24	18		1	01	23	17	1	01	7	07
162	21	15	1	01	26	1A	1) 0001,0101=15H	1	01	20	14	5	05	5	05
174	18	12	2	02	28	1C	2) 0001,1001=19H	2	02	16	10	10	0A	3	03
186	14	0E	4	04	29	1D	3) 0001,1010=1AH	4	04	10	0A	15	0F	2	02
198	11	0B	6	06	30	1E	4) 0010,1010=2AH	6	06	5	05	19	13	1	01
210	9	09	9	09	30	1E		9	09	1	01	21	15	1	01
222	6	06	11	0B	30	1E	1) 0001,0101=15H	6	06	5	05	19	13	1	01
234	4	04	14	0E	29	1D	2) 0001,0110=16H	4	04	10	0A	15	0F	2	02
246	2	02	18	12	28	1C	3) 0001,1010=1AH	2	02	16	10	10	0A	3	03
258	1	01	21	15	26	1A	4) 0010,1010=2AH	1	01	20	14	5	05	5	05
270	1	01	24	18	24	18		1	01	23	17	1	01	7	07
282	1	01	26	1A	21	15	1) 0001,0101=15H	1	01	20	14	5	05	5	05
294	2	02	28	1C	18	12	2) 0001,0110=16H	2	02	16	10	10	0A	3	03
306	4	04	29	1D	14	0E	3) 0010,0110=26H	4	04	10	0A	15	0F	2	02
318	6	06	30	1E	11	0B	4) 0010,1010=2AH	6	06	5	05	19	13	1	01
330	9	09	30	1E	9	09		9	09	1	01	21	15	1	01
342	11	0B	30	1E	6	06	1) 0001,0101=15H	6	06	5	05	19	13	1	01
354	14	0E	29	1D	4	04	2) 0010,0101=25H	4	04	10	0A	15	0F	2	02
6	18	12	28	1C	2	02	3) 0010,0110=26H	2	02	16	10	10	0A	3	03
18	21	15	26	1A	1	01	4) 0010,1010=2AH	1	01	20	14	5	05	5	05
30	24	18	24	18	1	01		1	01	23	17	1	01	7	07

3.4 ตารางข้อมูลสัญญาณมอดูเลตตามความกว้างของพัลส์ 24 บิตส์

จากการคำนวณได้ ค่า $h = 44$ แล้ว $A = 22$, $B = 1$

$$U = 22 \sin(\omega t) + 22 + 1$$

$$V = 22 \sin(\omega t + 120) + 22 + 1$$

$$W = 22 \sin(\omega t - 120) + 22 + 1$$

มุมมองค่า	U		V		W		ภาวะสวิตกึ่ง (SWITCHING STATE)	1'		2'		3'		4'	
	D	H	D	H	D	H		D	H	D	H	D	H	D	H
37.5	36	24	31	1F	1	01	1) 0001,0101=15H	1	01	30	1E	5	05	9	09
52.5	40	28	26	1A	3	03	2) 0010,0101=25H	3	03	23	17	14	0E	5	05
67.5	43	2B	20	14	6	06	3) 0010,1001=29H	6	06	14	0E	23	17	2	02
82.5	44	2C	15	0F	10	0A	4) 0010,1010=2AH	10	0A	5	05	29	1D	1	01
97.5	44	2C	10	0A	15	0F	1) 0001,0101=15H	10	0A	5	05	29	1D	1	01
112.5	43	2B	6	06	20	14	2) 0001,1001=19H	6	06	14	0E	23	17	2	02
127.5	40	28	3	03	26	1A	3) 0010,1001=29H	3	03	23	17	14	0E	5	05
142.5	36	24	1	01	31	1F	4) 0010,1010=2AH	1	01	30	1E	5	05	9	09
157.5	31	1F	1	01	36	24	1) 0001,0101=15H	1	01	30	1E	5	05	9	09
172.5	26	1A	3	03	40	28	2) 0001,1001=19H	3	03	23	17	14	0E	5	05
187.5	20	14	6	06	43	2B	3) 0001,1010=1AH	6	06	14	0E	23	17	2	02
202.5	15	0F	10	0A	44	2C	4) 0010,1010=2AH	10	0A	5	05	29	1D	1	01
217.5	10	0A	15	0F	44	2C	1) 0001,0101=15H	10	0A	5	05	29	1D	1	01
232.5	6	06	20	14	43	2B	2) 0001,0110=16H	6	06	14	0E	23	17	2	02
247.5	3	03	26	1A	40	28	3) 0001,1010=1AH	3	03	23	17	14	0E	5	05
262.5	1	01	31	1F	36	24	4) 0010,1010=2AH	1	01	30	1E	5	05	9	09
277.5	1	01	36	24	31	1F	1) 0001,0101=15H	1	01	30	1E	5	05	9	09
292.5	3	03	40	28	26	1A	2) 0001,0110=16H	3	03	23	17	14	0E	5	05
307.5	6	06	43	2B	20	14	3) 0010,0110=26H	6	06	14	0E	23	17	2	02
322.5	10	0A	44	2C	15	0F	4) 0010,1010=2AH	10	0A	5	05	29	1D	1	01
337.5	15	0F	44	2C	10	0A	1) 0001,0101=15H	10	0A	5	05	29	1D	1	01
352.5	20	14	43	2B	6	06	2) 0010,0101=25H	6	06	14	0E	23	17	2	02
7.5	26	1A	40	28	3	03	3) 0010,0110=26H	3	03	23	17	14	0E	5	05
22.5	31	1F	36	24	1	01	4) 0010,1010=2AH	1	01	30	1E	5	05	9	09

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา -47- ต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 ตารางข้อมูลสัญญาณมอดูเลตตามความกว้างของพัลส์ 18 บิตส์

จากการคำนวณได้ ค่า $h = 67$ แล้ว $A = 15$, $B = 1$

$$U = 33 \sin(\omega t) + 33 + 1$$

$$V = 33 \sin(\omega t + 120) + 33 + 1$$

$$W = 33 \sin(\omega t - 120) + 33 + 1$$

มุมมองค่า	U		V		W		ภาวีสวิตซิ่ง (SWITCHING STATE)	1'		2'		3'		4'	
	D	H	D	H	D	H		D	H	D	H	D	H	D	H
50	59	3B	40	28	3	03	1) 0001,0101=15H	3	03	37	25	19	13	9	09
70	65	49	28	1C	9	09	2) 0010,0101=25H	9	09	19	13	37	25	3	03
90	67	4B	18	12	18	12	3) 0010,1001=29H 4) 0010,1010=2AH	18	12	1	01	49	31	1	01
110	65	49	9	09	28	1C	1) 0001,0101=15H	9	09	19	13	37	25	3	03
130	59	3B	3	03	40	28	2) 0001,1001=19H	3	03	37	25	19	13	9	09
150	51	33	1	01	51	33	3) 0010,1001=29H 4) 0010,1010=2AH	1	01	50	32	1	01	17	11
170	40	28	3	03	59	3B	1) 0001,0101=15H	3	03	37	25	19	13	9	09
190	28	1C	9	09	65	49	2) 0001,1001=19H	9	09	19	13	37	25	3	03
210	18	12	18	12	67	4B	3) 0001,1010=1AH 4) 0010,1010=2AH	18	12	1	01	49	31	1	01
230	9	09	28	1C	65	49	1) 0001,0101=15H	9	09	19	13	37	25	3	03
250	3	03	40	28	59	3B	2) 0001,0110=16H	3	03	37	25	19	13	9	09
270	1	01	51	33	51	33	3) 0001,1010=1AH 4) 0010,1010=2AH	1	01	50	32	1	01	17	11
290	3	03	59	3B	40	28	1) 0001,0101=15H	3	03	37	25	19	13	9	09
310	9	09	65	49	28	1C	2) 0001,0110=16H	9	09	19	13	37	25	3	03
330	18	12	67	4B	18	12	3) 0010,0110=26H 4) 0010,1010=2AH	18	12	1	01	49	31	1	01
350	28	1C	65	49	9	09	1) 0001,0101=15H	9	09	19	13	37	25	3	03
10	40	28	59	3B	3	03	2) 0010,0101=25H	3	03	37	25	19	13	9	09
30	51	3B	51	33	1	01	3) 0010,0110=26H 4) 0010,1010=2AH	1	01	50	32	1	01	17	11

3.6 ตารางข้อมูลสัญญาณมอดูเลตตามความกว้างของพัลส์ 12 พัลส์

จากการคำนวณได้ ค่า $h = 114$ แล้ว $A = 57$, $B = 1$

$$U = 57 \sin(\omega t) + 57 + 1$$

$$V = 57 \sin(\omega t + 120) + 57 + 1$$

$$W = 57 \sin(\omega t - 120) + 57 + 1$$

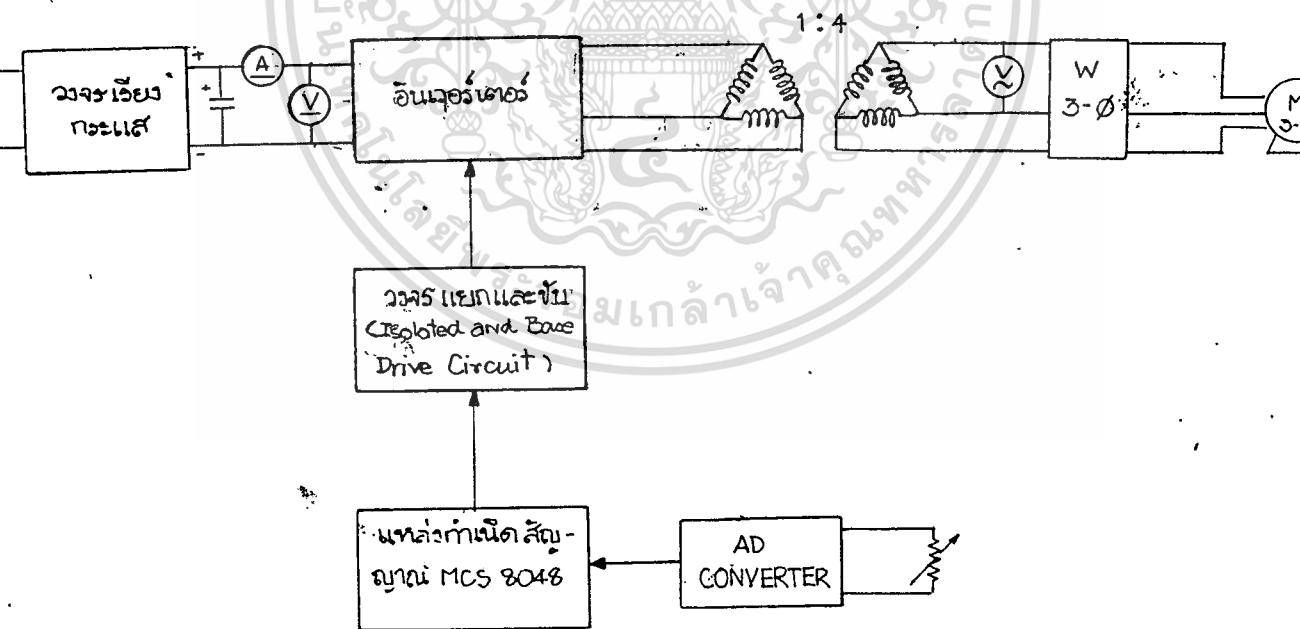
มุมมองค่า	U		V		W		ภาวะสวิตทิ่ง (SWITCHING STATE)	1'		2'		3'		4'	
	D	H	D	H	D	H		D	H	D	H	D	H	D	H
45 75	98 113	62 71	73 43	49 2B	3 18	03 12	1) 0001,0101=15H 2) 0010,0101=25H 3) 0010,1001=29H 4) 0010,1010=2AH	3 18	03 12	70 25	46 19	25 70	19 46	16 1	10 01
105 135	113 98	71 62	18 3	12 03	43 73	2B 49	1) 0001,0101=15H 2) 0001,1001=19H 3) 0010,1001=29H 4) 0010,1010=2AH	18 3	12 03	25 70	19 46	70 25	46 19	1 16	01 10
165 195	73 43	49 2B	3 18	03 12	98 113	62 71	1) 0001,0101=15H 2) 0001,1001=19H 3) 0001,1010=1AH 4) 0010,1010=2AH	3 18	03 12	70 25	46 19	25 70	19 46	16 1	10 01
225 255	18 3	12 03	43 73	2B 49	113 98	71 62	1) 0001,0101=15H 2) 0001,0110=16H 3) 0001,1010=1AH 4) 0010,1010=2AH	18 3	12 03	25 70	19 46	70 25	46 19	1 16	01 10
285 315	3 18	03 12	98 113	62 71	73 43	49 2B	1) 0001,0101=15H 2) 0001,0110=16H 3) 0010,0110=26H 4) 0010,1010=2AH	3 18	03 12	70 25	46 19	25 70	19 46	16 1	10 01
345 15	43 73	2B 49	113 98	71 62	18 3	12 03	1) 0001,0101=15H 2) 0010,0101=25H 3) 0010,0110=26H 4) 0010,1010=2AH	18 3	12 03	25 70	19 46	70 25	46 19	1 16	01 10

บทที่ 4

การทดลองและการวัดผล

ในบทนี้จะนำหลักการที่กล่าวไว้ในบทที่ 1, 2 และ 3 มาใช้ประกอบกันเป็นวงจรเพื่อที่จะทำการควบคุมมอเตอร์ และได้ติดตั้งอุปกรณ์วัดผลตามแผนภูมิข้างล่างนี้ เพื่อที่จะทำการวัดผลอัตราส่วนของโวลท์เตจต่อความถี่ ดังได้แสดงในตารางหน้าต่อไป

การทดลองจะคำนึงถึงแรงดันและความถี่ที่พิกัดของมอเตอร์เป็นหลัก จากการค่อยๆ เพิ่มค่าแรงดันดีซี จะได้ว่าค่าแรงดันดีซี 135V ทำให้แรงดันที่ออกจากขดลวดขั้วของหม้อแปลงสามเฟสมีค่าเท่ากับ แรงดันพิกัดของมอเตอร์ (220V ต่อแบบเดลต้า) ค่าความถี่ 50 Hz นั้นได้จากการปรับค่า x เท่ากับ $1010, 1010 = 22H$ ในบทนี้จะทำการทดลองวัดข้อมูลต่างๆ โดยใช้สัญญาณมอดูเลตตามความกว้างของพัลส์ 18 พัลส์เท่านั้น



ตารางบันทึกผลเมื่อจำนวนพัลส์เป็น 18 พัลส์

$f_{\text{ทฤษฎี}}$ (Hz)	x ค่าตัวหน้าเวลา	$f_{\text{จริง}}$ (Hz)	DC (V)	DC (A)	AC_{LL} (V)	WATT (W)	SPEED (rpm)	ประสิทธิภาพ (%)
14.3	1111.1111 = FFH	14.3	135	1.6	62	56	425	25.90
24.3	1000.0000 = 80H	24.5	135	2.0	107	63	735	23.33
33.8	0100.1101 = 4DH	34.1	135	2.5	152	74	1021	22.22
42.1	0011.0011 = 33H	42.4	135	2.7	189	77	1275	21.12
50	0010.0010 = 22H	50.1	135	3.0	220	82	1503	20.30
60	0001.0100 = 14H	60.3	135	3.2	266	85	1810	15.04

$f_{\text{ทฤษฎี}}$: ความถี่จากทฤษฎีคำนวณได้จากสมการที่ 3.1.1 หรือ 3.1.2

$f_{\text{จริง}}$: ความถี่จริงที่วัดได้จากการอ่านจากสโคป

x : ค่าตัวหน้าเวลาในรูปอ่านความถี่ สามารถอ่านได้จากหลอดไฟ LED

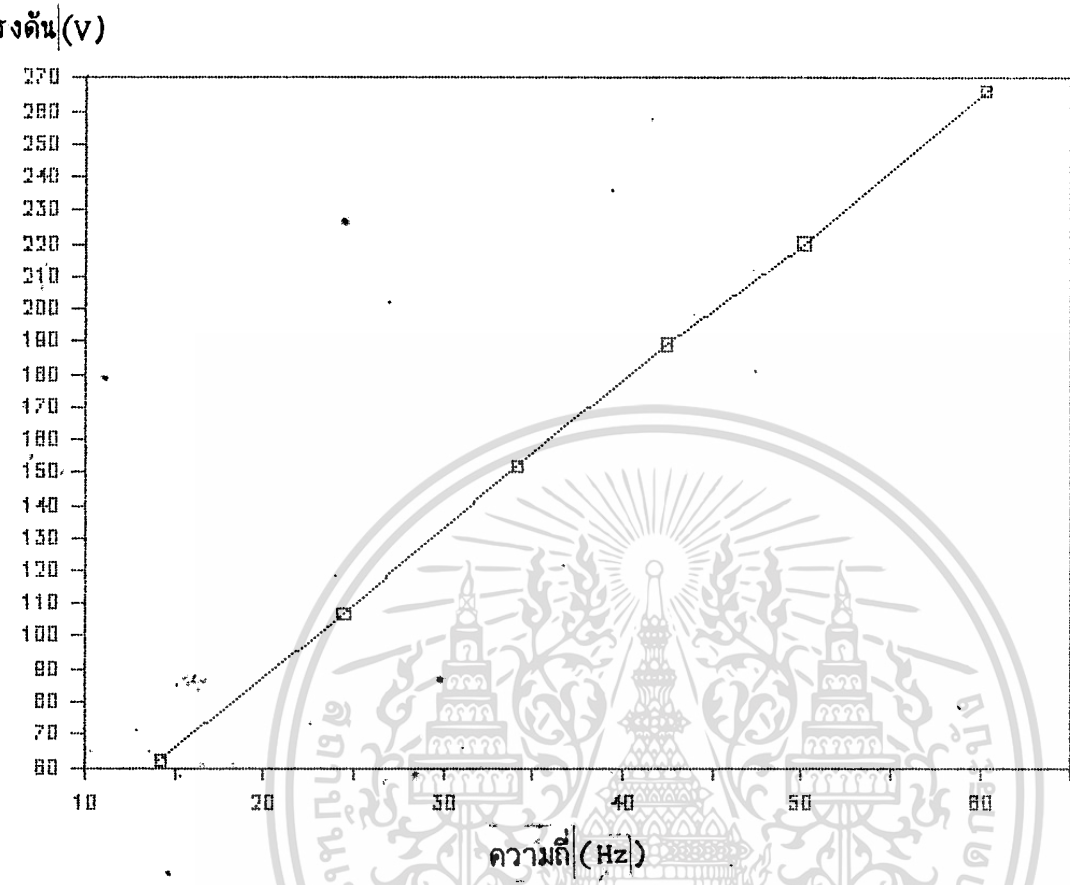
DC (V) : ค่าแรงดันไฟดีซีทางด้านอินพุทของวงจรถ่ายอินเวอร์เตอร์

DC (A) : ค่ากระแสไฟดีซีทางด้านอินพุทของวงจรถ่ายอินเวอร์เตอร์

AC_{LL} : ค่าแรงดันไฟเอซีทางด้านเอาต์พุทของขดลวดขั้วของหม้อแปลงสามเฟส (V)

WATT : ค่ากำลังทางด้านเอาต์พุทของขดลวดขั้วของหม้อแปลงสามเฟส (W)

η (%) : ค่าประสิทธิภาพของอินเวอร์เตอร์คิดเป็นเปอร์เซ็นต์



อัตราส่วนของค่าความต่างศักย์ต่อความถี่

บทที่ 5

บทวิจารณ์และสรุป

ในโครงการนี้ มุ่งเน้นในการนำเอาไมโครโพลีเซสเซอร์ MCS 8048 มาทำการสร้างสัญญาณควบคุมวงจรรีเลย์เตอร์เป็นหลัก เพื่อเป็นการแสดงให้เห็นว่า MCS 8048 มีความสามารถในการควบคุมสูงในงานควบคุมของระบบควบคุมเล็ก ๆ อย่างเช่นโครงการนี้ และยังเป็นการประหยัดอุปกรณ์ภายนอก (Peripheral Device) ที่นำมาอินเตอร์เฟส (Interfacing) ซึ่งต่างจากการใช้ไมโครโพลีเซสเซอร์ เบอร์ Z 80 เป็น CPU ในการกำเนิดสัญญาณ

บทวิจารณ์

แต่อย่างไรก็ตามโครงการนี้ก็ยังมีส่วนที่ต้องทำการพัฒนาต่อไปอีก เช่น ระบบของการควบคุมในโครงการนี้เป็นระบบเปิดทำให้การต่อภาระที่มอเตอร์ทำได้ในย่านแคบ ถ้าระบบได้มีการพัฒนาเป็นระบบปิด (closed loop) จะทำให้การต่อภาระที่มอเตอร์สามารถทำได้ในย่านกว้างขึ้น และส่วนที่จะต้องมีการทำขึ้นมาใหม่ก็คือ อุปกรณ์ป้องกัน (Protection Device) ได้แก่วงจรป้องกันกระแสเกินอาจทำได้ 2 วิธี คือ

1. วิธีการทางซอฟต์แวร์ โดยใช้สัญญาณในการ INT ให้เป็นประโยชน์โดยจะต้องมีอุปกรณ์ตรวจจับกระแส อาจจะใช้ตัวความต้านทาน หรือ โมดูลวัดกระแสที่สามารถตั้งกระแสที่ต้องการให้ตัดได้ แล้วนำสัญญาณที่ได้จากอุปกรณ์ตรวจจับกระแสมาส่งให้ขาสัญญาณ INT ซึ่งภายในหน่วยจำโปรแกรมจะต้องเก็บโปรแกรมอินเตอร์รัพท์เซอร์วิสรูทีน (Interrupt Service Routine) ไว้ว่าจะสั่งให้เอาท์พุทค่า ๒๕๕ มาทางพอร์เอาท์พุท ($P_{20} - P_{27}$) แล้วให้วนลูปจนกระทั่งสัญญาณ INT จะไม่แอกทีฟ เป็นต้น

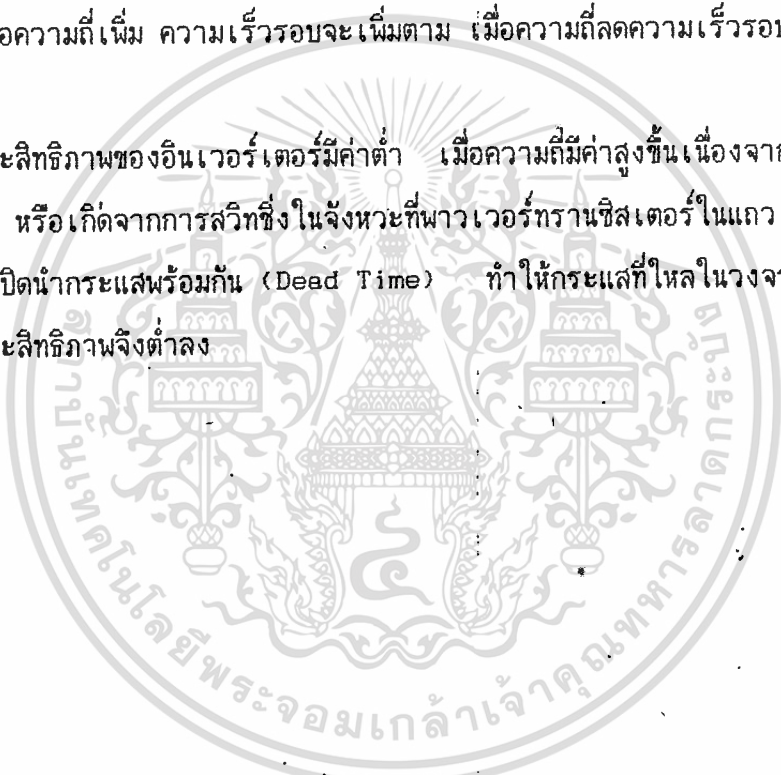
2. วิธีการทางฮาร์ดแวร์ สามารถทำได้โดยการสร้างวงจรถ่วงเบรกเกอร์อิเล็กทรอนิกส์ที่มีความไวในการติดต่อวงจรสูงมากเมื่อกระแสเกิดขีดจำกัด

ซึ่งวิธีที่เหมาะสมก็คือ วิธีการทางซอฟต์แวร์ เนื่องจากเราสามารถแก้ไขโปรแกรมที่อยู่ในอีพรอม (EPROM) ได้ และอาศัยอุปกรณ์เพิ่มเติมอีกเล็กน้อย และยังมีบางสิ่งซึ่งจะต้องได้รับการทดสอบอีก เช่น ผลเมื่อต่อภาระให้มอเตอร์ ซึ่งยังไม่ได้ทำการทดสอบ จากการทดลองจะพบว่าประสิทธิภาพของอินเวอร์เตอร์นั้นยังมีค่าต่ำ เกิดจากการสวิตชิงในจังหวะที่พาวเวอร์ทรานซิสเตอร์ในแถวเดียวกัน เช่น Q_1 กับ Q_2 เปิดนำกระแสพร้อมกัน (Dead Time) ทำให้กระแสที่ไหลในวงจรอินเวอร์เตอร์มีค่าสูงขึ้น, ประสิทธิภาพจึงต่ำลง สาเหตุอีกอย่างที่น่าจะทำให้ประสิทธิภาพของอินเวอร์เตอร์ต่ำอาจเกิดจากในส่วนของสแน็บเบอรันั้นมีความไม่เหมาะสม ทำให้กำลังสูญเสียในส่วนของสแน็บเบอรัมีค่ามาก สิ่งต่างๆ เหล่านี้จำเป็นต้องได้รับการพัฒนาต่อไป

บทสรุป

จากการทดลองวัดผลและนำมาเขียนกราฟอัตราส่วนระหว่าง โวลต์ เตจต่อความถี่ จะเห็นว่า อัตราส่วนระหว่าง โวลต์ เตจต่อความถี่นั้นเป็นเชิงเส้น และค่าความชันของกราฟก็คือค่าอัตราส่วน โวลต์ เตจต่อความถี่นั่นเองจากการทดลองจะได้ว่า

1. ที่ความถี่ต่ำ ๆ เส้นกราฟจะไม่เป็นเชิงเส้น
2. ที่ความถี่สูง เส้นกราฟจะเป็นเส้นตรง
3. ค่าอัตราส่วนระหว่าง โวลต์ เตจต่อความถี่คงที่ คือ ถ้าความถี่เพิ่ม โวลต์ เตจจะเพิ่มด้วย ถ้าความถี่ลด โวลต์ เตจจะลดลงด้วย
4. เมื่อความถี่เพิ่ม ความเร็วรอบจะเพิ่มตาม เมื่อความถี่ลดความเร็วรอบจะลดตามด้วยเช่นกัน
5. ประสิทธิภาพของอินเวอร์เตอร์มีค่าต่ำ เมื่อความถี่มีค่าสูงขึ้นเนื่องจากความถี่ในการสวิตช์มีค่าสูง หรือเกิดจากการสวิตช์ในจังหวะที่พาวเวอร์ทรานซิสเตอร์ในแถวเดียวกัน เช่น Q_1 กับ Q_2 เปิดนำกระแสพร้อมกัน (Dead Time) ทำให้กระแสที่ไหลในวงจรอินเวอร์เตอร์มีค่าสูงขึ้น * ประสิทธิภาพจึงต่ำลง



กิตติกรรมประกาศ

โครงการและปริญญาบัตร ฉบับนี้ สำเร็จลุล่วงลงได้ โดยความเอื้อเฟื้อกรุณาของอาจารย์ นิพนธ์ เลหาสงคราม และอาจารย์ สุวงศ์ ที่ได้ให้คำแนะนำ และอุปการะในการทำโครงการ หากขาดทั้งสองท่านนี้ โครงการคงจะไม่สำเร็จลุล่วงลงได้ จึงขอขอบคุณท่านทั้งสองในกิตติกรรมประกาศฉบับนี้ และสิ่งสำคัญที่สุด คือ ขอขอบพระคุณ คุณพ่อ และ คุณแม่ ของผู้จัดทำ ที่ได้ส่งเสริมและสนับสนุนทั้งทางด้านกำลังทรัพย์, กำลังใจ ในการทำปริญญาบัตร ฉบับนี้จนสำเร็จได้



ภาคผนวก
คู่มือ การใช้อุปกรณ์

intel®

8048AH/8035AHL/8049AH
8039AHL/8050AH/8040AHL

PRELIMINARY

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias ... 0°C to 70°C
Storage Temperature -65°C to +150°C
Voltage On Any Pin With Respect
to Ground -0.5V to +7V
Power Dissipation 1.5 Watt

*NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. CHARACTERISTICS: (T_A = 0°C to 70°C; V_{CC} = V_{DD} = 5V ± 10%; V_{SS} = 0V)

Symbol	Parameter	Limits			Unit	Test Conditions	Device
		Min	Typ	Max			
V _{IL}	Input Low Voltage (All Except RESET, X1, X2)	-5		.8	V		All
V _{IL1}	Input Low Voltage (RESET, X1, X2)	-5		.6	V		All
V _{IH}	Input High Voltage (All Except XTAL1, XTAL2, RESET)	2.0		V _{CC}	V		All
V _{IH1}	Input High Voltage (X1, X2, RESET)	3.8		V _{CC}	V		All
V _{OL}	Output Low Voltage (BUS)			.45	V	I _{OL} = 2.0 mA	All
V _{OL1}	Output Low Voltage (RD, WR, PSEN, ALE)			.45	V	I _{OL} = 1.8 mA	All
V _{OL2}	Output Low Voltage (PROG)			.45	V	I _{OL} = 1.0 mA	All
V _{OL3}	Output Low Voltage (All Other Outputs)			.45	V	I _{OL} = 1.6 mA	All
V _{OH}	Output High Voltage (BUS)	2.4			V	I _{OH} = -400 μA	All
V _{OH1}	Output High Voltage (RD, WR, PSEN, ALE)	2.4			V	I _{OH} = -100 μA	All
V _{OH2}	Output High Voltage (All Other Outputs)	2.4			V	I _{OH} = -40 μA	All



8048AH/8035AHL/8049AH
8039AHL/8050AH/8040AHL

PRELIMINARY

D.C. CHARACTERISTICS: ($T_A = 0^\circ\text{C}$ to 70°C ; $V_{CC} = V_{DD} = 5V \pm 10\%$; $V_{SS} = 0V$) (Continued)

Symbol	Parameter	Limits			Unit	Test Conditions	Device
		Min	Typ	Max			
I_{L1}	Leakage Current (T1, INT)			± 10	μA	$V_{SS} \leq V_{IN} \leq V_{CC}$	All
I_{LI1}	Input Leakage Current (P10-P17, P20-P27, EA, SS)			-500	μA	$V_{SS} + .45 \leq V_{IN} \leq V_{CC}$	All
I_{LI2}	Input Leakage Current $\overline{\text{RESET}}$	20		300	μA	$V_{SS} \leq V_{IN} \leq 3.8V$	All
I_{L0}	Leakage Current (BUS, T0) (High Impedance State)			± 10	μA	$V_{SS} \leq V_{IN} \leq V_{CC}$	All
I_{DD}	VDD Supply Current (RAM Standby)		3	5	mA		8048AH 8035AHL
			4	7	mA		8049AH 8039AHL
			5	10	mA		8050AH 8040AHL
$I_{DD} + I_{CC}$	Total Supply Current*		30	65	mA		8048AH 8035AHL
			35	70	mA		8049AH 8039AHL
			40	80	mA		8050AH 8040AHL
V_{DD}	RAM Standby Voltage	2.2		5.5	V	Standby Mode Reset $\leq V_{IL1}$	All

* $I_{CC} + I_{DD}$ is measured with all outputs disconnected; $\overline{\text{SS}}$, $\overline{\text{RESET}}$, and $\overline{\text{INT}}$ equal to V_{CC} ; EA equal to V_{SS} .



8048AH/8035AHL/8049AH
8039AHL/8050AHL/8040AHL

PRELIMINARY

A.C. CHARACTERISTICS: ($T_A = 0^\circ\text{C}$ to 70°C ; $V_{CC} = V_{DD} = 5V \pm 10\%$; $V_{SS} = 0V$)

Symbol	Parameter	f (t) (Note 3)	11 MHz		Unit	Conditions (Note 1)
			Min	Max		
t	Clock Period	1/xtal freq	90.9	1000	ns	(Note 3)
t _{LL}	ALE Pulse Width	3.5t-170	150		ns	
t _{AL}	Addr Setup to ALE	2t-110	70		ns	(Note 2)
t _{LA}	Addr Hold from ALE	t-40	50		ns	
t _{CC1}	Control Pulse Width (\overline{RD} , \overline{WR})	7.5t-200	480		ns	
t _{CC2}	Control Pulse Width (\overline{PSEN})	6t-200	350		ns	
t _{DW}	Data Setup before \overline{WR}	6.5t-200	390		ns	
t _{WD}	Data Hold after \overline{WR}	t-50	40		ns	
t _{DR}	Data Hold (\overline{RD} , \overline{PSEN})	1.5t-30	0	110	ns	
t _{RD1}	\overline{RD} to Data in	6t-170		375	ns	
t _{RD2}	\overline{PSEN} to Data in	4.5t-170		240	ns	
t _{AW}	Addr Setup to \overline{WR}	5t-150	300		ns	
t _{AD1}	Addr Setup to Data (\overline{RD})	10.5t-220		730	ns	
t _{AD2}	Addr Setup to Data (\overline{PSEN})	7.5t-200		460	ns	
t _{AFC1}	Addr Float to \overline{RD} , \overline{WR}	2t-40	140		ns	(Note 2)
t _{AFC2}	Addr Float to \overline{PSEN}	.5t-40	10		ns	(Note 2)
t _{L AFC1}	ALE to Control (\overline{RD} , \overline{WR})	3t-75	200		ns	
t _{L AFC2}	ALE to Control (\overline{PSEN})	1.5t-75	60		ns	
t _{CA1}	Control to ALE (\overline{RD} , \overline{WR} , \overline{PROG})	t-65	25		ns	
t _{CA2}	Control to ALE (\overline{PSEN})	4t-70	290		ns	
t _{CP}	Port Control Setup to \overline{PROG}	1.5t-80	50		ns	
t _{PC}	Port Control Hold to \overline{PROG}	4t-260	100		ns	
t _{PR}	\overline{PROG} to P2 Input Valid	8.5t-120		650	ns	
t _{PF}	Input Data Hold from \overline{PROG}	1.5t	0	140	ns	
t _{DP}	Output Data Setup	6t-290	250		ns	
t _{PD}	Output Data Hold	1.5t-90	40		ns	
t _{PP}	\overline{PROG} Pulse Width	10.5t-250	700		ns	
t _{PL}	Port 2 I/O Setup to ALE	4t-200	160		ns	
t _{LP}	Port 2 I/O Hold to ALE	.5t-30	15		ns	
t _{PV}	Port Output from ALE	4.5t+100		5:0	ns	
t _{OPRR}	T0 Rep Rate	3t	270		ns	
t _{CY}	Cycle Time	15t	1.36	15.0	μs	

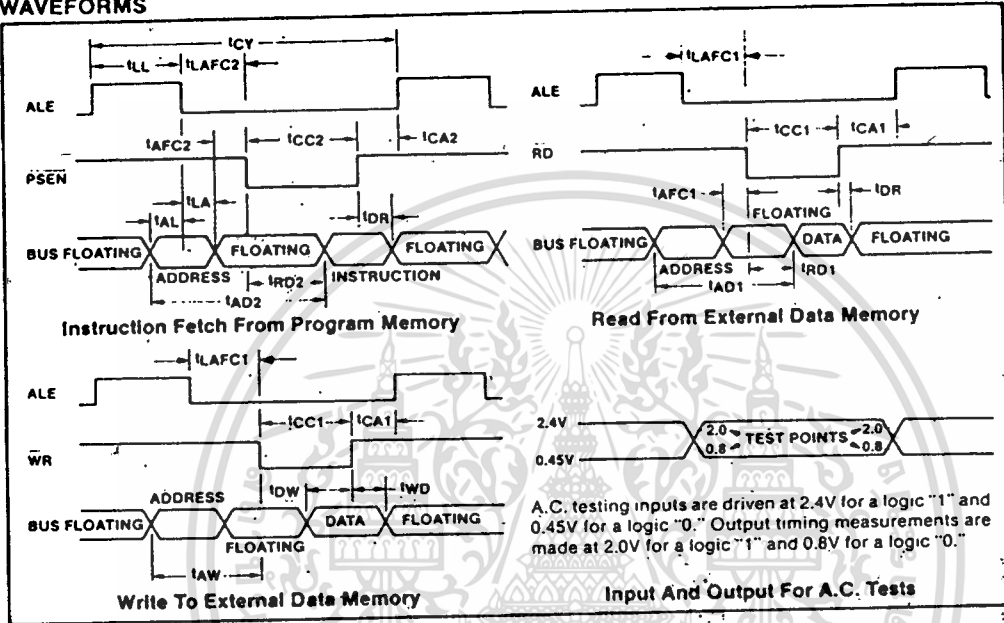
Notes:

1 Control Outputs $C_L = 80\text{pF}$
BUS Outputs $C_L = 150\text{pF}$

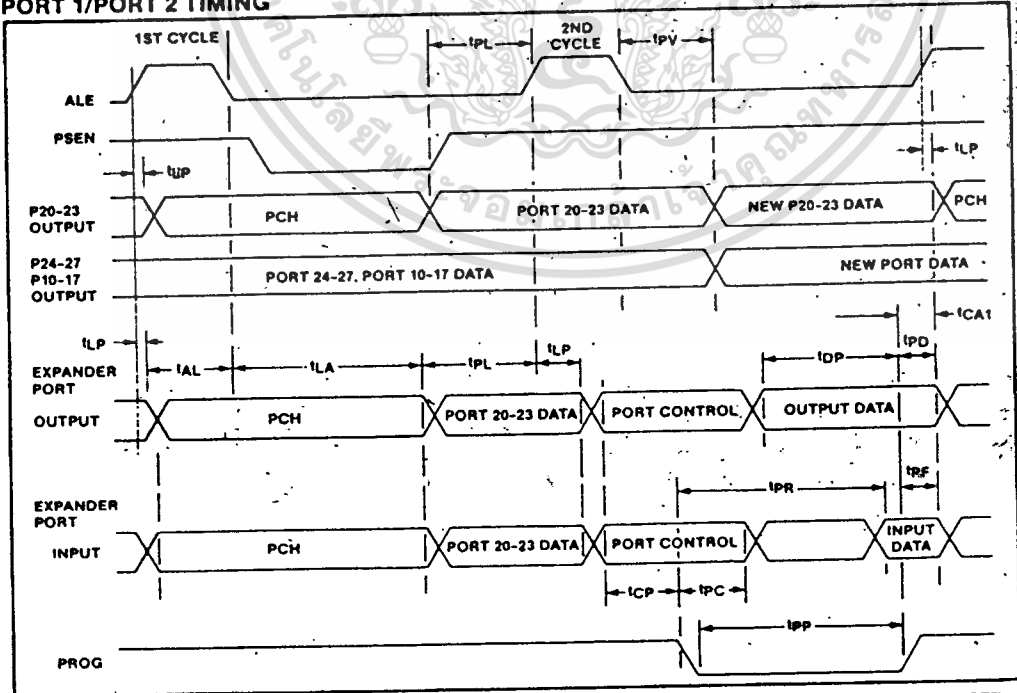
2. BUS High Impedance
Load 20pF

3. f(t) assumes 50% duty cycle on X1, X2. Max
clock period is for a 1 MHz crystal input.

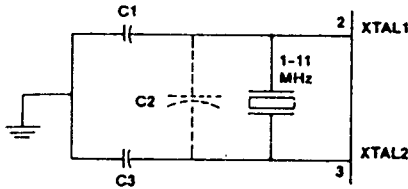
WAVEFORMS



PORT 1/PORT 2 TIMING



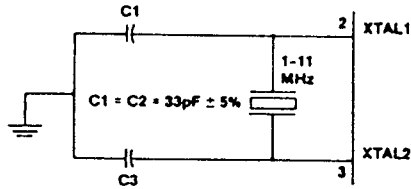
CRYSTAL OSCILLATOR MODE



$C1 = 5pF \pm 1/2pF + (STRAY < 5pF)$
 $C2 = (CRYSTAL + STRAY) < 8pF$
 $C3 = 20pF \pm 1pF + (STRAY < 5pF)$

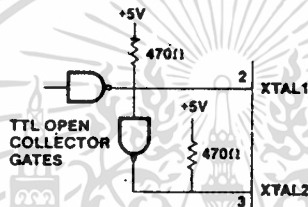
Crystal series resistance should be less than 30Ω at 11 MHz;
less than 75Ω at 6 MHz; less than 180Ω at 3.6 MHz.

CERAMIC RESONATOR MODE



$C1 = C2 = 33pF \pm 5\%$

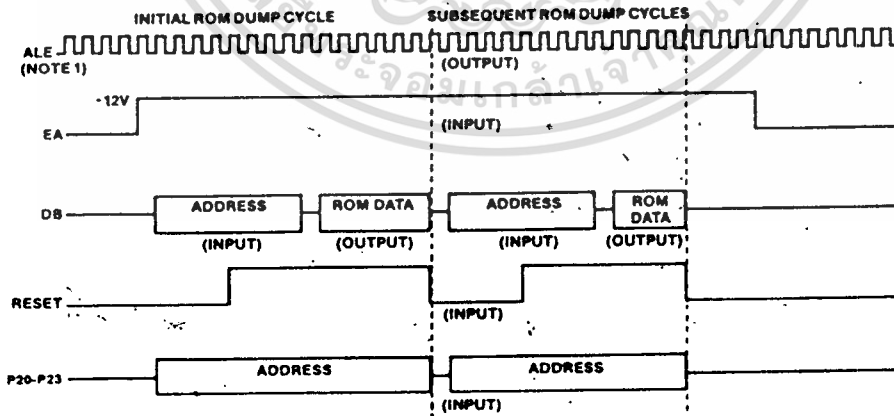
DRIVING FROM EXTERNAL SOURCE



For XTAL1 and XTAL2 define "high" as voltages above 1.6V and "low" as voltages below 1.6V. The duty cycle requirements for externally driving XTAL1 and XTAL2 using the

circuit shown above are as follows: XTAL1 must be high 35-65% of the period and XTAL2 must be high 36-65% of the period. Rise and fall times must be faster than 20 nS.

SUGGESTED ROM VERIFICATION ALGORITHM FOR H-MOS DEVICE ONLY



	48H	49H	50H
A10	0	ADDR	ADDR
A11	0	0	ADDR

VCC = VDD = 5V

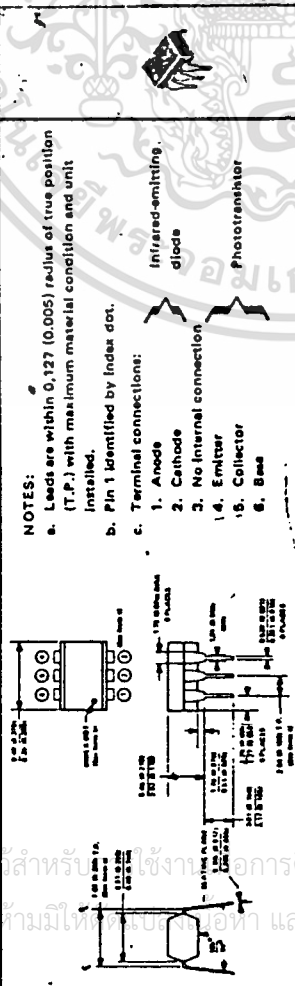
VSS = 0V

NOTE: ALE is function of X1, X2 inputs.

- Gallium Arsenide Diode Infrared Source Optically Coupled to a Silicon N-P-N Phototransistor
- High Direct-Current Transfer Ratio
- High-Voltage Electrical Isolation . . . 2.5-kV, 1.5-kV, or 0.5-kV Rating
- Plastic Dual-In-Line Package
- High-Speed Switching . . . $t_r = 2 \mu s$, $t_f = 2 \mu s$ Typical

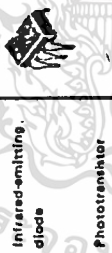
Mechanical data

The package consists of a gallium arsenide infrared-emitting diode and an n-p-n silicon phototransistor mounted on a 6-lead frame encapsulated within an electrically nonconductive plastic compound. The case will withstand soldering temperature with no deformation and device performance characteristics remain stable when operated in high-humidity conditions. Unit weight is approximately 0.52 grams.



NOTES:

- Leads are within 0.127 (0.005) radius of true position (T.P.) with maximum material condition and unit installed.
- Pin 1 identified by index dot.
- Terminal connections:
 - Anode
 - Cathode
 - No internal connection
 - Emitter
 - Collector
 - Base



ABSOLUTE MAXIMUM RATINGS AT 25°C free-air temperature (unless otherwise noted)

Peak Input-to-Output Voltage: 4N25	±2.5 kV
4N26, 4N27	±1.5 kV
4N28	±0.5 kV
Collector-Base Voltage	30 V
Collector-Emitter Voltage (See Note 1)	70 V
Emitter-Collector Voltage	7 V
Emitter-Base Voltage	7 V
Input-Diode Reverse Voltage	3 V
Input-Diode Continuous Forward Current at (or below) 25°C Free-Air Temperature (See Note 2)	80 mA
Input-Diode Peak Forward Current ($t_w = 300 \mu s$, duty cycle = 2%)	3 A
Continuous Power Dissipation at (or below) 25°C Free-Air Temperature:	
Infrared-Emitting Diode (See Note 3)	150 mW
Phototransistor (See Note 3)	150 mW
Total, Infrared-Emitting Diode plus Phototransistor (See Note 4)	250 mW
Storage Temperature Range	-55°C to 150°C
Lead Temperature 1.6 mm (1/16 inch) from Case for 10 Seconds	260°C

JEDEC registered data. This data sheet contains all applicable JEDEC-registered data in effect at the time of publication.

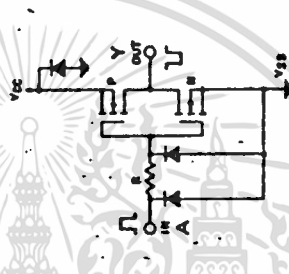
- NOTES:**
- This value applies when the base-emitter diode is open-circuited.
 - Derate linearly to 100°C free-air temperature at the rate of 1.33 mW/°C.
 - Derate linearly to 100°C free-air temperature at the rate of 2 mW/°C.
 - Derate linearly to 100°C free-air temperature at the rate of 3.33 mW/°C.

DESCRIPTION

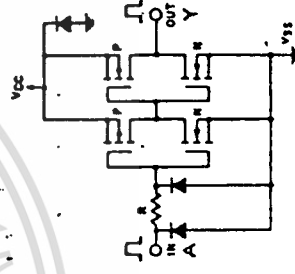
The SCL4049UB and SCL4050B are inverting and Non-Inverting Hex Buffers, respectively, and feature logic-level conversion using only one supply voltage (V_{CC}). The input-signal high level (V_{IH}) can exceed the V_{CC} supply voltage when these devices are used for logic-level conversions. These devices are intended for use as CMOS-to-DTL/TTL converters and can drive directly two DTL/TTL loads.

The SCL4049UB and SCL4050B are interchangeable with SCL4009UB and SCL4010B devices, respectively. In these applications the SCL4049UB and SCL4050B are pin-compatible with the SCL4009UB and SCL4010B, respectively, and can be substituted for these devices in existing as well as in new designs. Terminal No. 16 is not connected internally on the SCL4049UB or SCL4050B; therefore, connection to this terminal is of no consequence to circuit operation.

SCHEMATIC DIAGRAMS



SCL4049UB



SCL4050B

NC	6Y	6A	NC	5Y	5A	4Y	4A
16	15	14	13	12	11	10	9
SCL4049UB SCL4050B							
1	2	3	4	5	6	7	8
VCC	1Y	1A	2Y	2A	3Y	3A	VSS

Add suffix for package:

- C - 16-pin Cerdip
- D - 16-pin Ceramic
- E - 16-pin Epoxy
- F - 16-pin Flat
- H - Chip

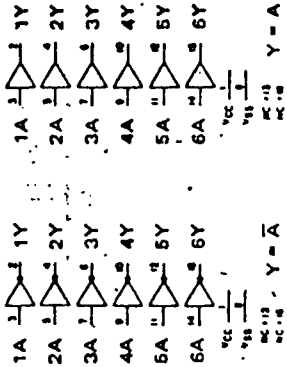
RECOMMENDED OPERATING CONDITIONS

For maximum reliability:

- DC Supply Voltage $V_{CC} - V_{SS}$ 3 to 15 Vd
- Operating Temperature TA -55 to +125 °C
- C, D, F, H Device -40 to +85 °C
- E Device

Note: These devices contain input protection networks to V_{SS} only. Therefore, V_{IH} (max) may exceed V_{CC} without damage, (subject to absolute maximum ratings).

LOGIC DIAGRAMS



SCL4049UB

SCL4050B

ADC0808, ADC0809 data acquisition component is a monolithic CMOS device with an 8-bit analog-to-digital converter, 8-channel multiplexer and microprocessor compatible control logic. The 8-bit A/D converter uses successive approximation as the conversion technique. The converter features a high impedance chopper stabilized comparator, a 256R voltage divider with analog switch tree and successive approximation register. The 8-channel multiplexer can directly access any of 8 single-ended analog signals.

The device eliminates the need for external zero and full-scale adjustments. Easy interfacing to microprocessors is provided by the latched and decoded multiplexer address inputs and latched TRI-STATE outputs.

The design of the ADC0808, ADC0809 has been optimized incorporating the most desirable aspects of several conversion techniques. The ADC0808, ADC0809 offers high speed, high accuracy, minimal temperature dependence, excellent long-term accuracy and repeatability and consumes minimal power. These features make the device ideally suited to applications from process and machine control to consumer and automotive applications. For 16-channel multiplexer with common output (sample/hold port) see ADC0816 data sheet. (See AN-247 for more information.)

- Resolution — 8-bits
- Total unadjusted error — $\pm 1/2$ LSB and ± 1 LSB
- No missing codes
- Conversion time — 100 μ s
- Single supply — 5 Vdc
- Operates ratiometrically or with 5 Vdc or analog span adjusted voltage reference
- 8-channel multiplexer with latched control logic
- Easy interface to all microprocessors, or operates "stand alone"
- Outputs meet T_L voltage level specifications
- 0V to 5V analog input voltage range with single 5V supply
- No zero or full-scale adjust required
- Standard hermetic or molded 28-pin DIP package
- Temperature range —40°C to +85°C or —55°C to +125°C
- Low power consumption — 15 mW
- Latched TRI-STATE output

Monotonicity is particularly important in closed loop feedback control systems. A non-monotonic relationship can cause oscillations that will be catastrophic for the system. Additionally, the 256R network does not cause load variations on the reference voltage.

The bottom resistor and the top resistor of the ladder network in Figure 1 are not the same value as the remainder of the network. The difference in these resistors causes the output characteristic to be symmetrical with the zero and full-scale points of the transfer curve. The first output transition occurs when the analog signal has reached $+1/2$ LSB and succeeding output transitions occur every 1 LSB later up to full-scale.

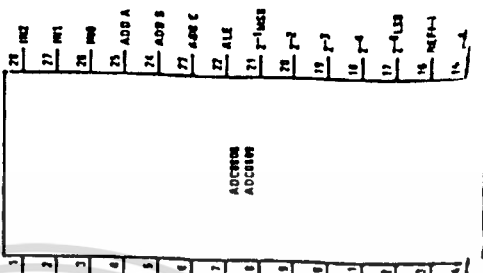
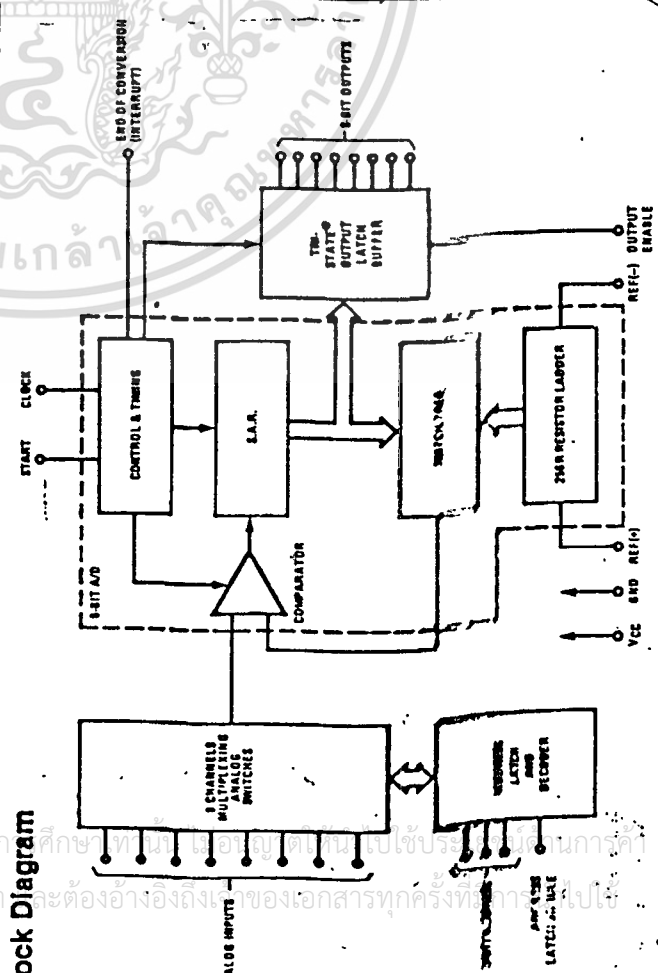
The successive approximation register (SAR) performs 8 iterations to approximate the input voltage. For any SAR type converter, n-iterations are required for an n-bit converter. Figure 2 shows a typical example of a 3-bit converter. In the ADC0808, ADC0809, the approximation technique is extended to 8 bits using the 256R network.

ANALOG CHANNEL		C	B	A
IN0		L	L	L
IN1		L	L	H
IN2		L	H	L
IN3		L	H	H
IN4		H	L	L
IN5		H	L	H
IN6		H	H	L
IN7		H	H	H

CONVERTER CHARACTERISTICS

The Converter
The heart of this single chip data acquisition system is its 8-bit analog-to-digital converter. The converter is designed

Connection Diagram



เอกสารอ้างอิง

1. BOWES, S.R. (1975), "NEW SINUSOIDAL PULSE WIDTH MODULATED INVERTER", Proc. IEEE, Vol. 122, No. 11, Nov. 1975, PP. 1279-1285
2. TAKAHASHI, MIYAIRI, "THE RELATION OF GATE CONTROL SIGNAL AND OUTPUT WAVEFORM OF PWM INVERTER", IEE of Japan, Vol. 95, No. 2, Feb. 1976, PP. 25-31
3. M.A. RAHMAN, J.E. QUAINCOE, M.A. CHOUDHURY, "A COMPARATIVE STUDY OF DELTA AND SINE PULSE-WIDTH MODULATION INVERTERS", Pro. EPE, Vol. 1, Oct. 1985, PP. 1.163-1.167
4. วิชัย ศังขจันทรานนท์, "ควบคุมเครื่องกลไฟฟ้าด้วยอิเล็กทรอนิกส์กำลัง 1", สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 379 หน้า, 2528
5. วิชัย ศังขจันทรานนท์, "ควบคุมเครื่องกลไฟฟ้าด้วยอิเล็กทรอนิกส์กำลัง 2", สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 385 หน้า, 2528
6. บริษัท ซีเอ็ดดูเคชั่น จำกัด, "คู่มือไอซี ไมโครโปรเซสเซอร์", บริษัท ซีเอ็ดดูเคชั่น จำกัด, 753 หน้า, 2529