การควบคุมเสต็ปเปอร์มอเตอร์แบบวงปิด

CLOSED-LOOP CONTROL OF STEPPER MOTOR

โดย

นายวรากร ฐปนางกูร

นายวิชัย ณรงค์วณิชย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาวิศวกรรมระบบควบคุม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2534

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ปริญญานิพนธ์ปีการศึกษา 2534

ภาควิชา วิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การควบคุมเสต็ปเปอร์มอเตอร์แบบวงปิด

ผู้จัดทำ

1. นาย วรากร ฐปนางกูร
2. นาย วิชัย ณรงค์วณิชย์

รศ.ดร. โยธิน เปรมปราณีรัชต์ อาจารย์ที่ปรึกษา

(.....)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การควบคุมเสต็ปเปอร์มอเตอร์แบบวงปิด

วรากร ฐปนางกูร
วิชัย ณรงค์วณิชย์

รศ.ดร.โยธิน เปรมปราณีรัชต์
อาจารย์ที่ปรึกษา

ปีการศึกษา 2534

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้อธิบายถึง คุณสมบัติของเสต็ปเปอร์มอเตอร์ (STEPPER MOTOR) ลักษณะของการกระตุ้นเสต็ปเปอร์มอเตอร์ การพัฒนาการควบคุมเสต็ปเปอร์มอเตอร์ ซึ่งโดยปรกติทั่วไปการควบคุมมอเตอร์ชนิดนี้จะเป็นการควบคุมแบบวงเปิด ซึ่งถ้าเพิ่มการควบคุมแบบวงปิดเข้าไปก็จะทำให้ประสิทธิภาพในการทำงานของมอเตอร์เพิ่มขึ้นทั้งด้านความแม่นยำทางด้านตำแหน่งและความเร็วในการทำงานที่สูงขึ้น โดยใช้เครื่องคอมพิวเตอร์ (COMPUTER) เป็นตัวควบคุมโดยรับข้อมูลจากตัวเอ็นโค้ดเดอร์ (ENCODER) เข้ามาประมวลผลเพื่อที่จะส่งสัญญาณการกระตุ้นกระแสควบคุมตามจังหวะเวลา แล้วอินเตอร์เฟส (INTERPHASE) กับวงจรขับมอเตอร์ โดยวงจรอินเตอร์เฟสที่ใช้ทามเมอร์ (TIMER COUNTER) 8253 และโลจิกซีเควนเซอร์ (LOGIC SEQUENCER) 8713 ทำการส่งสัญญาณไปขับมอเตอร์โดยวงจรขับแบบ 4 เฟส จากผลการทดลองปรากฏว่าสามารถรับสัญญาณจากตัวเอ็นโค้ดเดอร์ได้ วงจรอินเตอร์เฟสให้สัญญาณถูกต้อง สามารถควบคุมให้มอเตอร์หมุนได้จำนวนขั้นตามต้องการ และสามารถทำให้มอเตอร์สามารถใช้ได้ในความถี่ใช้งานที่สูงขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CLOSED-LOOP CONTROL OF STEPPER MOTOR

WARAKORN THAPANANGKUN

WICHAI NARONGWANIT

Assc.Prof YOTHIN PREMPRANERATCH,Dr.Eng.Advisor

1991

Abstract

This thesis decribes the characteristic of stepper motor, switching angle and pulse injections, and development of stepper motor control. Closed-loop control of stepper motor can improve the performance by using positional feedback to determine the proper phase switchings with effect to the rotor position. Motor can achieve not only more accurate position control but also much higher speed and smoother speed. The encoder send the position of the motor to computer which is controller. This controller will send the control signal to programable counter (8253) and logic sequencer (8713) and these will drive the motor by the driver. The result of tests on the built system shows that the encoder can send the signal to the computer through interface circuit ,giving the correct signals ,controlling motor to the desired position and improving the efficiency of the motor.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ขอขอบคุณท่านอาจารย์ โยธิน เปรมปราณีรัชต์ ที่ได้กรุณาให้คำแนะนำและช่วยแก้ปัญหาต่าง ๆ ที่เกิดขึ้นมาโดยตลอดจนปริญญานิพนธ์นี้สำเร็จลงได้ ขอขอบคุณอาจารย์ทุก ๆ ท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้แก่ผู้เขียน ขอขอบคุณเพื่อน ๆ ที่ช่วยกันอยู่ทำงานร่วมกัน และสุดท้ายขอขอบคุณทุก ๆ สิ่งที่ทำให้มีกำลังต่อสู้จนกระทั่งสามารถทำปริญญานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยดี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อ	I
Abstract	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VI
สารบัญภาพ	VII
สัญญลักษณ์	IX
บทที่ 1 บทนำ	1
บทที่ 2 ไอบริดเสต็ปเปอร์มอเตอร์	3
บทที่ 3 คุณสมบัติของแรงบิดของมอเตอร์	8
3.1 คุณสมบัติของแรงบิดสถิตย์ของมอเตอร์	8
3.2 คุณสมบัติของแรงบิดพลวัตรของมอเตอร์	9
บทที่ 4 วงจรอินเตอร์เฟส	13
4.1 วงจรอินเตอร์เฟส	13
4.2 การทำงานของวงจรอินเตอร์เฟส	14
4.3 การจัดการขัดจังหวะ	14
4.4 การโปรแกรม 8259 ตัวควบคุมการขัดจังหวะ	15
4.5 ข้อควรระวังในการเขียนการจัดการการขัดจังหวะฮาร์ดแวร์	17
4.6 ทามเมอร์เคาเตอร์ 8253	18
4.7 โหมดการทำงานของ 8253	18
4.8 วงจรโลจิกซีเควนเซอร์	19
บทที่ 5 วงจรขับและเอ็นโค้ดเดอร์	22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
บทที่ 6 ผลการทดลอง	24
บทที่ 7 สรุปผลและวิจารณ์	26
เอกสารอ้างอิง	27
ภาคผนวก	28
ภาคผนวก ก โปรแกรมการควบคุม	29
ภาคผนวก ข โลจิกชีเควนเซอร์ 8713	37
ภาคผนวก ค การติดต่อกับคอมพิวเตอร์	41
ภาคผนวก ง ทามเมอร์เคาเตอร์	48

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

		หน้า
ตารางที่ 4.1	ตารางแสดงค่าของ IRQ	17
ตารางที่ 6.2	ตารางแสดงผลการทดลอง	25

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ

		หน้า
รูปที่ 1.1	บล็อกไดอะแกรมของการควบคุมเสต็ปเปอร์มอเตอร์แบบวงปิด	2
รูปที่ 2.1	โครงสร้างของเสต็ปเปอร์มอเตอร์แบบไฮบริด จำนวนซี่ของโรเตอร์คือ 30, จำนวนซี่ของสเตเตอร์คือ 24, ซี่ของโรเตอร์ทั้งสองชุดต่างกัน 3 องศา, ลูกศรแสดงถึงทิศทางของเส้นแรงแม่เหล็กที่เกิดขึ้นเนื่องจากกระแส I_1 และ I_2	3
รูปที่ 2.2	วงจรแม่เหล็กของมอเตอร์ ในรูปแสดงถึงทิศทางของเส้นแรงแม่เหล็กเมื่อกระตุ้นเฟสที่ 1	4
รูปที่ 2.3	ลำดับ 4 ขั้นของเสต็ปเปอร์มอเตอร์แบบไฮบริดแบบ 2 เฟส	5
รูปที่ 2.4	การพัน (a)แบบเดี่ยว (b)แบบคู่	6
รูปที่ 2.5	เสต็ปเปอร์มอเตอร์แบบไฮบริด (a) แบบขดลวดเดี่ยวใช้แหล่งจ่ายไฟสองชุด (b) แบบขดลวดคู่ใช้แหล่งจ่ายไฟชุดเดียว ลำดับการกระตุ้นเฟส (c) แบบทีละหนึ่ง (d)แบบทีละสอง (e)แบบหนึ่งสอง	7
รูปที่ 3.1	คุณสมบัติของแรงบิดสถิตย์ของมอเตอร์	8
รูปที่ 3.2	กราฟแสดงความสัมพันธ์ระหว่างคุณสมบัติของแรงบิดและความเร็ว	9
รูปที่ 3.3	ผลของกระแสกระตุ้นเมื่อความเร็วของมอเตอร์มีค่าต่าง ๆ และเมื่อเพิ่มค่าความต้านทานภายนอกให้กับวงจร	10
รูปที่ 3.4	กราฟแสดงความสัมพันธ์ ระหว่างแรงบิด และความ	11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้า

เร็ว เมื่อลดค่าเวลาคงที่

รูปที่ 3.5 การกระตุ้นกระแสของสเต็ปปิ้งมอเตอร์แบบ 3 เฟส 12

รูปที่ 4.1 วงจรอินเตอร์เฟส 21

รูปที่ 5.1 เอ็นโค้ดเดอร์ 22

รูปที่ 5.2 วงจรขับ 23

รูปที่ 6.1 ผลการทดลอง 24

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญลักษณ์

P_r : ระยะห่างระหว่างซี่ของโรเตอร์

n : จำนวนสแตกของมอเตอร์แบบไฮบริด

I_1 : กระแสกระตุ้นเฟสที่ 1

I_2 : กระแสกระตุ้นเฟสที่ 2

I_+ : กระแสเฟสบวก

I_- : กระแสเฟสลบ

I_{ph} : ค่ากระแสแต่ละเฟสของมอเตอร์

T_H : ค่าแรงบิดสูงสุดของมอเตอร์

E : ค่าความต่างศักย์ของเฟส

R : ค่าความต้านทานต่อเฟส

EP_1 : ตำแหน่งสมดุลย์ของมอเตอร์

θ_c : ตำแหน่งที่จะทำการกระตุ้นมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบันเสต็ปเปอร์มอเตอร์เป็นอุปกรณ์ชนิดหนึ่งซึ่งนิยมใช้แพร่หลาย เช่น เป็นส่วนประกอบของดิสก์ไดร์ฟ (disk drive), เครื่องพิมพ์, แขนกล ฯลฯ เนื่องจากเสต็ปเปอร์มอเตอร์มีวิธีการควบคุมที่ง่าย มีความสามารถในการเคลื่อนที่เป็นขั้น ๆ มีความเที่ยงตรงเป็นที่น่าพอใจ ซึ่งโดยทั่วไปการควบคุมเสต็ปเปอร์มอเตอร์เป็นการควบคุมแบบวงเปิด การควบคุมแบบวงเปิดเป็นการควบคุมที่ง่าย แต่การควบคุมแบบนี้จะเกิดความผิดพลาดขึ้นได้ เมื่อตัวมอเตอร์ได้รับภาระมากเกินไปโดยที่เราไม่สามารถที่จะทราบได้ เราสามารถที่จะแก้ปัญหานี้ได้โดยใช้การควบคุมแบบวงปิด

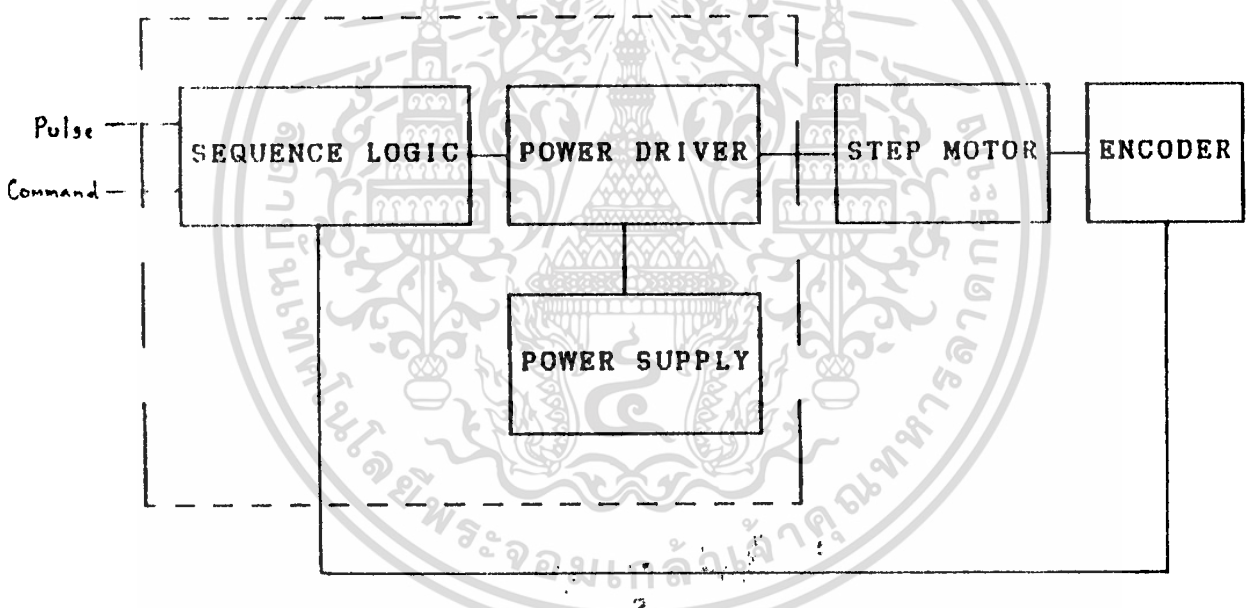
การควบคุมเสต็ปเปอร์มอเตอร์เป็นการควบคุมที่ง่าย เมื่อเราสามารถควบคุมแบบวงเปิดได้ ภายใต้การควบคุมแบบวงเปิดเสต็ปเปอร์มอเตอร์จะขึ้นอยู่กับลำดับของการกระตุ้น ซึ่งการควบคุมในลักษณะนี้เป็นที่นิยมอย่างแพร่หลายเนื่องจากความประหยัดและการควบคุมที่ไม่ยุ่งยาก เพราะไม่จำเป็นที่จะต้องอาศัยตัวเอ็นโค้ดเดอร์หรือวงจรทางอิเลคทรอนิกที่ซับซ้อน ด้วยเหตุนี้การควบคุมแบบเปิดนี้จึงเป็นที่ยอมรับโดยทั่วไป ตราบจนกว่าความสามารถในการควบคุมเสต็ปเปอร์มอเตอร์จะไม่มีความแม่นยำหรือประสิทธิภาพที่เพียงพอในการทำงาน เช่นเสต็ปเปอร์มอเตอร์ไม่สามารถที่จะตอบสนองต่อสัญญาณคำสั่งทุก ๆ คำสั่งที่เกิดขึ้นได้ ทำให้เกิดความผิดพลาดทางตำแหน่งได้

กล่าวโดยทั่วไปความสามารถของเสต็ปเปอร์มอเตอร์ขึ้นอยู่กับรูปแบบของการควบคุม เนื่องจากเป็นธรรมดาที่ความสามารถของการควบคุมเสต็ปเปอร์มอเตอร์แบบวงเปิดจะมีจำกัด เนื่องจากไม่มีสัญญาณป้อนกลับที่จะทำให้สามารถล่วงรู้ถึงความผิดพลาดที่เกิดขึ้นหรือผลตอบสนองของความเร็ว ความผิดพลาดที่เกิดขึ้นส่วนใหญ่จะเกิดขึ้นเมื่อความถี่ของสัญญาณในการทำงานมีค่ามาก ทำให้เสต็ปเปอร์มอเตอร์ไม่สามารถที่จะตอบสนองได้ครบถ้วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความสามารถของเสต็ปเปอร์มอเตอร์สามารถที่จะพัฒนาขึ้นได้อย่างมากโดยการใช้การป้อนกลับของตำแหน่งหรือความเร็วของโรเตอร์(rotor) เราจะสามารถควบคุมตำแหน่งได้อย่างแน่นอนและความเร็วในการทำงานก็จะสูงขึ้นและเรียบขึ้นได้

การควบคุมเสต็ปเปอร์มอเตอร์แบบวงปิดมีหลายชนิดเช่นระบบเซอร์โวแบบวงปิด (Closed-loop servo system) ระบบควบคุมความเร็วโดยเฟสล็อคลูป (Phase-locked loop) ส่วนระบบที่ใช้ในการทดลองนี้เป็นแบบใช้การป้อนกลับของผลตอบสนองทางตำแหน่งของโรเตอร์ ดังรูปที่ 1.1



รูปที่ 1.1 บล็อกไดอะแกรมของการควบคุมเสต็ปเปอร์มอเตอร์แบบวงปิด

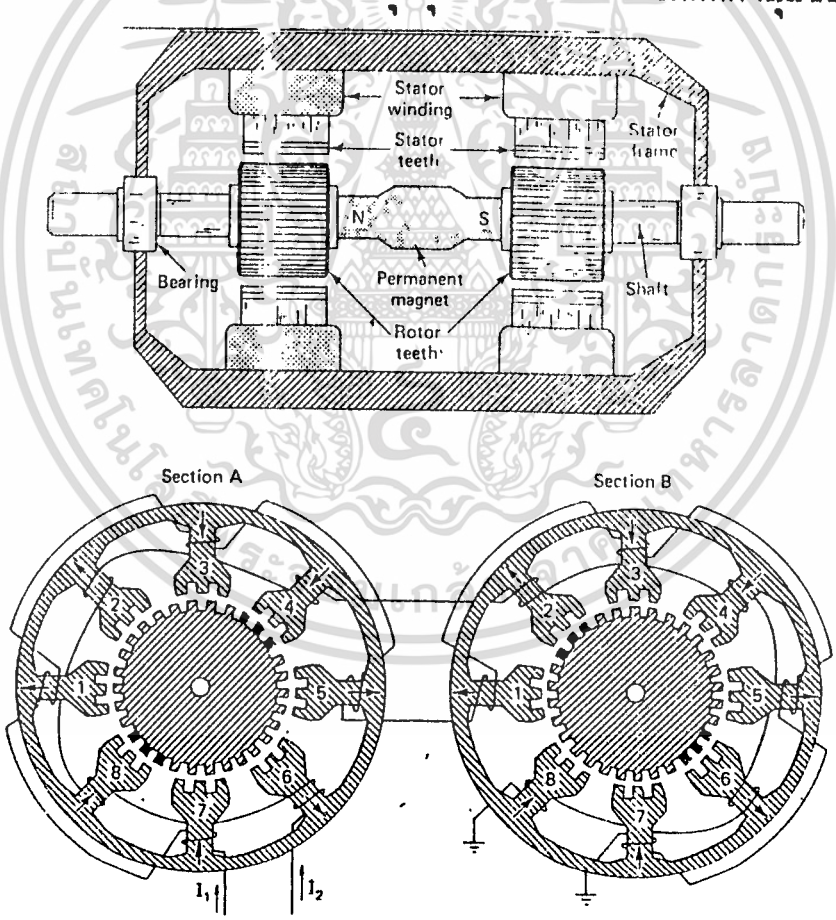
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ไฮบริดเสต็ปเปอร์มอเตอร์

เสต็ปเปอร์มอเตอร์แบบไฮบริดเป็นการนำคุณสมบัติของมอเตอร์แบบแม่เหล็กถาวรและแบบวาริเอเบิลรีลักแตน (variable reluctance) มารวมกัน โครงสร้างของมอเตอร์แบบนี้ประกอบด้วยแกนของโรเตอร์ที่เป็นแม่เหล็กถาวร อยู่ระหว่างซี่ของโรเตอร์สองชุดหรือมากกว่า ดังที่แสดงให้เห็นในรูปที่ 2.1

จำนวนซี่ของโรเตอร์แต่ละชุด จะมีค่าแตกต่างจากจำนวนที่มีอยู่บนสเตเตอร์ ซึ่งแตกต่างกับมอเตอร์แบบมัลติแสตกวาริเอเบิลรีลักแตน(multistack variable reluctance) ที่ซี่ของแสตกบนโรเตอร์ทุกชุดมีค่าเท่ากัน โดยที่ค่ามุมของซี่ของโร



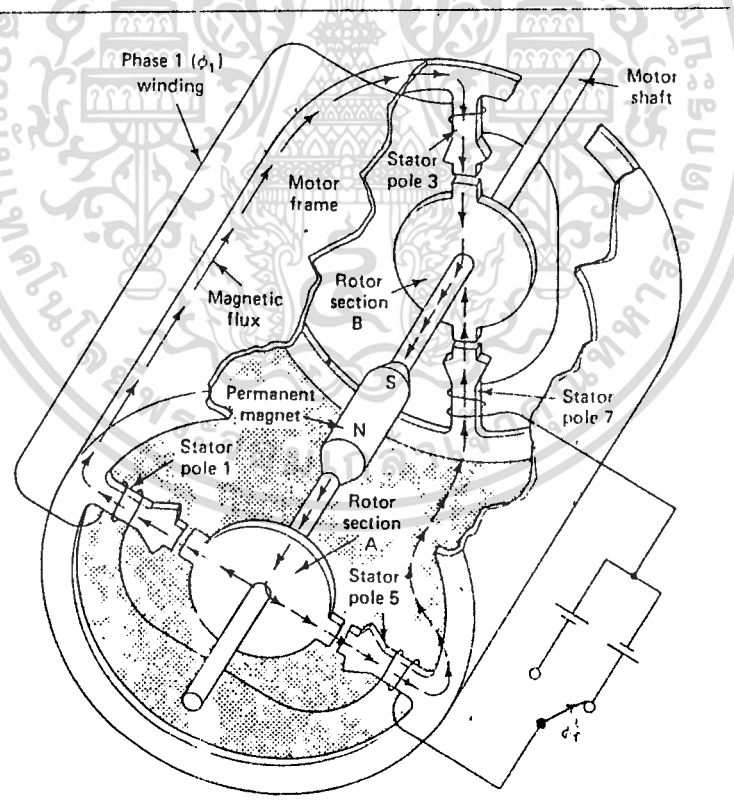
รูปที่ 2.1 โครงสร้างของเสต็ปเปอร์มอเตอร์แบบไฮบริด จำนวนซี่ของโรเตอร์คือ 30, จำนวนซี่ของสเตเตอร์คือ 24, ซี่ของโรเตอร์ทั้งสองชุดต่างกัน 3 องศา, ลูกศรแสดงถึงทิศทางของเส้นแรงแม่เหล็กที่เกิดขึ้นเนื่องจากกระแส I_1 และ I_2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เตอร์ทั้งสองแสตกมีค่าเบี่ยงเบนไปP_r/nโดยที่ n คือจำนวนแสตก จากรูป ส่วน A และ B แสดงถึงโครงสร้างภายในของมอเตอร์

ขดลวดสเตเตอร์แต่ละเฟสจะถูกแบ่งไปพันแต่ละโพล(pole)ตามรูป โดยเฟสที่ 1 พันอยู่บนโพลที่ 1,3,5,และ 7 ของส่วน A และ B ส่วนโพลที่เหลือก็พันโดยเฟสที่ 2 ตามรูป

แกนแม่เหล็กถาวรมีขั้วเหนืออยู่ที่ส่วน A และมีขั้วใต้อยู่ที่ส่วน B เราพิจารณามอเตอร์ในขณะที่เฟสที่ 1 ถูกกระตุ้นโดยกระแส I_1 ตามทิศทางดังรูป ขั้วของโรเตอร์ในส่วน A จะถูกดึงดูดโดยขั้วสเตเตอร์ของโพล 1 และ 5 ส่วนส่วน B ก็จะดูดกับโพล 3 และ 7 ดังรูปที่ 2 วงจรแม่เหล็กมีทิศทางจากขั้วเหนือของแม่เหล็กถาวรผ่านเข้าโรเตอร์ของส่วน A ผ่านออกไปทางโพลที่ 1 และ 5 แล้ว

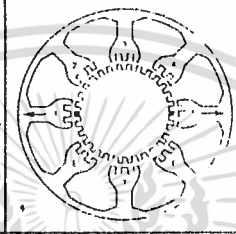
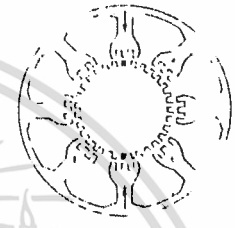
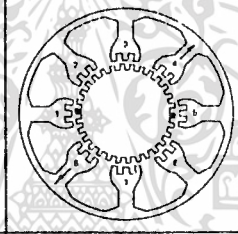
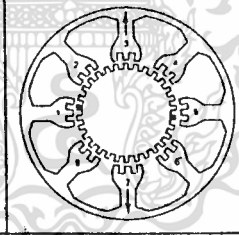
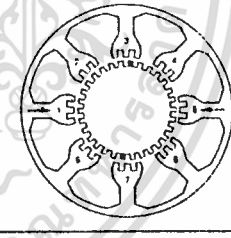
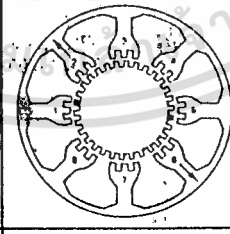
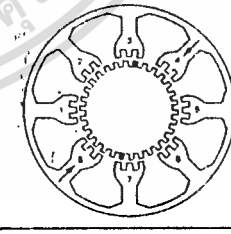
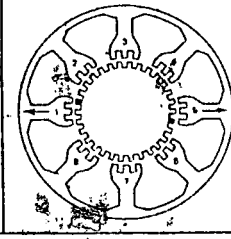
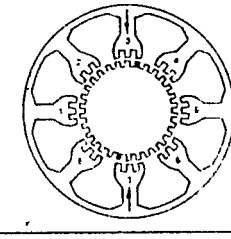


รูปที่ 2.2 วงจรแม่เหล็กของมอเตอร์ ในรูปแสดงถึงทิศทางของเส้นแรงแม่เหล็กเมื่อกระตุ้นเฟสที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านเฟรมของสเตเตอร์เข้าโรเตอร์ส่วน B ทางโพลที่ 3 และ 7 แล้วไปสิ้นสุดที่แม่เหล็กถาวรขั้วใต้

เพื่อที่จะขับเคลื่อนแกนของมอเตอร์ไปหนึ่งเสต็ปในทิศทางตามเข็มนาฬิกาเราจะต้องตัดกระแส I_1 แล้วกระตุ้นกระแส I_2 โดยที่เพื่อให้เห็นชัดเจนจะเห็นว่าใน

Step	ϕ_1 I_1	ϕ_2 I_2	Flux out sec. A pole nos.	Flux in sec. B pole nos.	Section A	Section B
1	+		1, 5	3, 7		
2		−	4, 8	2, 6		
3	−		3, 7	1, 5		
4		+	2, 6	4, 8		
1	+		1, 5	3, 7		

รูปที่ 2.3 ลำดับ 4 ขั้นของเสต็ปเปอร์มอเตอร์แบบไฮบริดแบบ 2 เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

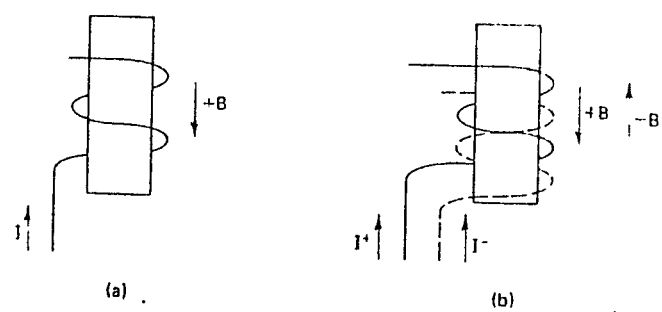
ส่วนของโรเตอร์ที่เป็นซี่สีดำในรูปที่ 1 จะอยู่ใกล้โพลมากที่สุด คือมีค่าเท่ากับมุมหนึ่งเสต็ปพอดี ในการที่เราจะกระตุ้นกระแส I_2 จากรูปเราจะเห็นว่าเราจะต้องกระตุ้นให้เกิด I_2 ในทิศทางที่ตรงกันข้ามกับลูกศร นั่นคือถ้าเราต้องการให้มอเตอร์หมุนในทิศทางตามเข็มนาฬิกาเราจะต้องทำการกระตุ้นตามลำดับคือ 1^+ 2^- 1^- 2^+ 1^+ โดยสำหรับการหมุนในทิศทางทวนเข็มนาฬิกาก็เรียงลำดับตรงกันข้ามกัน

จากที่ได้แสดงมาแล้วจะเห็นว่า กระแสที่จะใช้ในการกระตุ้นมอเตอร์ให้ทำงานในแต่ละเฟสจะต้องมีทิศทางสองทาง นั่นคือจำเป็นที่จะต้องใช้แหล่งจ่ายไฟสองชุด ซึ่งไม่เป็นที่นิยมเพราะสิ้นเปลืองค่าใช้จ่าย โดยทั่วไปเราจะใช้แหล่งจ่ายไฟเพียงตัวเดียวในการทำงานมากกว่า

ความแตกต่างของมอเตอร์ที่ใช้แหล่งจ่ายไฟหนึ่งตัวและสองตัวก็คือโครงสร้างของขดลวดที่พันอยู่ภายในสเตเตอร์ สเตเตอร์ที่มีขดลวดพันอยู่แบบชุดเดี่ยวจะต้องใช้แหล่งจ่ายไฟสองชุด ส่วนแหล่งจ่ายไฟชุดเดียวสามารถใช่ร่วมกับมอเตอร์ที่ใช้การพันแบบขดลวดคู่

การพันขดลวดในรูปที่ 1 นั้นเหมือนกันกับรูปที่ 4a นั่นคือทิศทางของกระแสต้องเปลี่ยนเพื่อที่จะเปลี่ยนทิศทางของเส้นแรงแม่เหล็ก

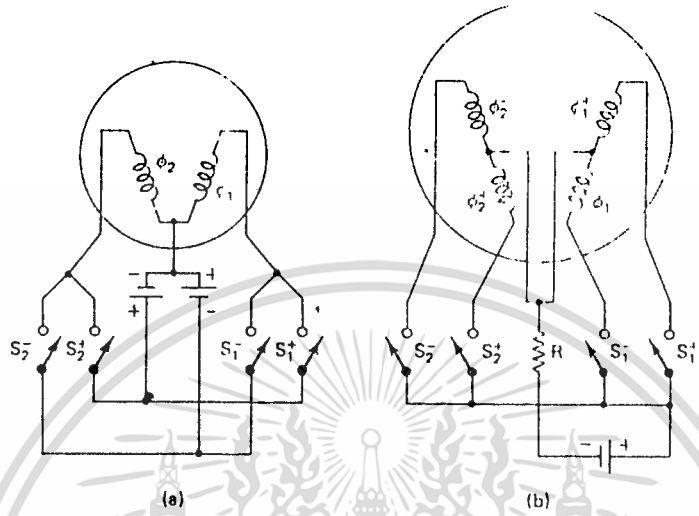
ส่วนในรูปที่ 4b เป็นการพันแบบคู่ จะเห็นว่าเราไม่จำเป็นที่จะต้องเปลี่ยนทิศทางของกระแส เราเพียงใช้กระแสจากแหล่งจ่ายไฟตัวเดิมในการกระตุ้นกระแส I^+ I^- ก็จะเกิดเส้นแรงแม่เหล็กในทิศทางตรงข้ามกันตามที่ต้องการ



รูปที่ 2.4 การพัน (a)แบบเดี่ยว (b)แบบคู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.5 แสดงถึงวงจรในการกระตุ้นและลำดับในการกระตุ้นมอเตอร์ในแบบต่าง ๆ โดยสมมติว่ามอเตอร์ในรูปที่ 1.1 มีการพันแบบคู่



Step	S_1^+	S_1^-	S_2^+	S_2^-
1	X			
2				X
3		X		
4			X	
1	X			

(c)

Step	S_1^+	S_1^-	S_2^+	S_2^-
1	X			X
2		X		X
3		X	X	
4	X		X	
1	X			X

(d)

Step	S_1^+	S_1^-	S_2^+	S_2^-
1	X			X
2				X
3		X		X
4		X		
5		X	X	
6			X	
7	X		X	
8	X			
1	X			X

(e)

รูปที่ 2.5 เสต็ปเปอร์มอเตอร์แบบไฮบริด (a)แบบขดลวดเดี่ยวใช้แหล่งจ่ายไฟสองชุด (b)แบบขดลวดคู่ใช้แหล่งจ่ายไฟชุดเดียว ลำดับการกระตุ้นเฟส (c)แบบทีละหนึ่ง (d)แบบทีละสอง (e)แบบหนึ่งสอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

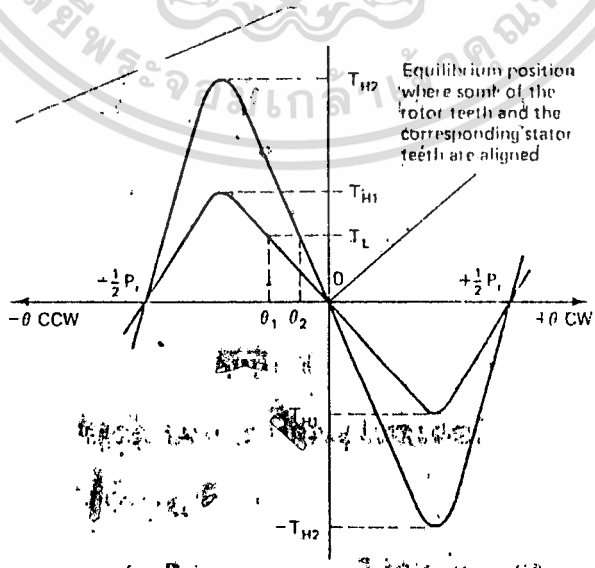
คุณสมบัติของแรงบิดของมอเตอร์

3.1 คุณสมบัติของแรงบิดสถิตย์ของมอเตอร์

ในขณะที่มอเตอร์อยู่ในสภาวะหยุดนิ่ง ซี่ของตัวโรเตอร์จำนวนหนึ่งจะดูดอยู่กับซี่ของสเตเตอร์ นั่นคือว่ามอเตอร์อยู่ในสภาวะสมดุลย์ จากรูปที่ 3.1 มอเตอร์จะไม่มีแรงบิดเลยในขณะที่อยู่ที่ตำแหน่ง 0 เมื่อมอเตอร์ถูกทำให้เปลี่ยนตำแหน่งไปในทิศทางทวนเข็มนาฬิกาเป็นมุม θ โดยภาระภายนอก มอเตอร์จะสร้างแรงบิดในขนาดที่เท่ากันในทิศทางตรงกันข้ามเพื่อที่จะรักษาสมดุลย์ โดยที่ความสัมพันธ์ระหว่างค่าของแรงบิดและตำแหน่งจะปรากฏอยู่ในรูปคลื่นซายน์ โดยลักษณะจริง ๆ จะขึ้นอยู่กับโครงสร้างภายในของขดลวด โดยมีคาบระยะทางคือหนึ่งซี่โรเตอร์

ค่าสูงสุดของแรงบิดจะขึ้นอยู่กับค่าของกระแส ซึ่งในที่นี้คือ T_H แต่ถึงแม้ว่าโดยทั่วไปแล้วค่าของแรงบิดจะขึ้นอยู่กับค่าของกระแส ค่าแรงบิดจะถึงจุดอิ่มตัวเมื่อค่าของกระแสถึงจุดอิ่มตัว คือค่าของเส้นแรงแม่เหล็กอิ่มตัว ค่าแรงบิดของมอเตอร์ก็จะไม่เพิ่มขึ้นตามค่าของกระแส

ค่าของแรงบิดจะมีค่าเป็นศูนย์เมื่อตำแหน่งของโรเตอร์อยู่ที่ระยะครึ่งหนึ่งของระยะซี่โรเตอร์ มอเตอร์มีโอกาสที่จะเกิดการลื่นไหลไปยังตำแหน่งสมดุลย์อื่นได้



รูปที่ 3.1 คุณสมบัติของแรงบิดสถิตย์ของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นั่นคือเราจะถือว่าจุดนี้เป็นจุดวิกฤต เมื่อเราออกแบบเพื่อให้มอเตอร์มีค่าตำแหน่งที่ถูกต้อง เราจำเป็นที่จะต้องใช้มอเตอร์ที่มีค่าแรงบิดสูงสุดมากกว่าค่าแรงบิดของค่าภาระภายนอกของมอเตอร์

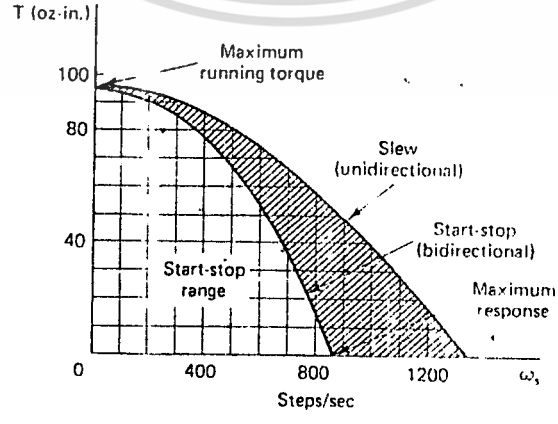
3.2 คุณสมบัติของแรงบิดพลวัตรของมอเตอร์

แรงบิดสถิตย์ของมอเตอร์ที่ได้กล่าวไปแล้วในหัวข้อที่ผ่านมา คือขณะที่มอเตอร์อยู่นิ่ง ในทางปฏิบัติแล้วโรเตอร์จะต้องมีการหมุนเพื่อขับเคลื่อนภาระภายนอก โดยมอเตอร์จำเป็นที่จะต้องใช้แรงบิดจำนวนมาก เพื่อที่จะเอกชนะแรงเสียดทานและเร่งการขับเคลื่อน ซึ่งถ้าหากว่ามอเตอร์ไม่สามารถที่จะเอาชนะภาระภายนอกได้ จะทำให้เกิดการผิดพลาดของตำแหน่งได้

โดยปรกติแล้วความสามารถในการสร้างแรงบิดของมอเตอร์นั้น ทางผู้ผลิตจะเป็นผู้บอกมา ดังตัวอย่างในรูปที่ 7 จากรูปจะเห็นว่าคุณสมบัติของมอเตอร์แบ่งออกเป็น 2 ส่วน คือ ส่วนเริ่มและหยุด และส่วนไม่สามารถควบคุมทิศทางได้

ส่วนเริ่มและหยุดจะเป็นส่วนที่ในระหว่างการขับเคลื่อนมอเตอร์อยู่สามารถที่จะหยุดหรือเปลี่ยนทิศทางไปในทิศทางตรงข้ามได้โดยที่ไม่มีข้อผิดพลาดทางตำแหน่งเกิดขึ้น

ส่วนไม่สามารถควบคุมทิศทางได้เป็นส่วนที่มอเตอร์มีความเร็วในการหมุนมาก



รูปที่ 3.2 กราฟแสดงความสัมพันธ์ระหว่างคุณสมบัติของแรงบิดและความเร็ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
007682

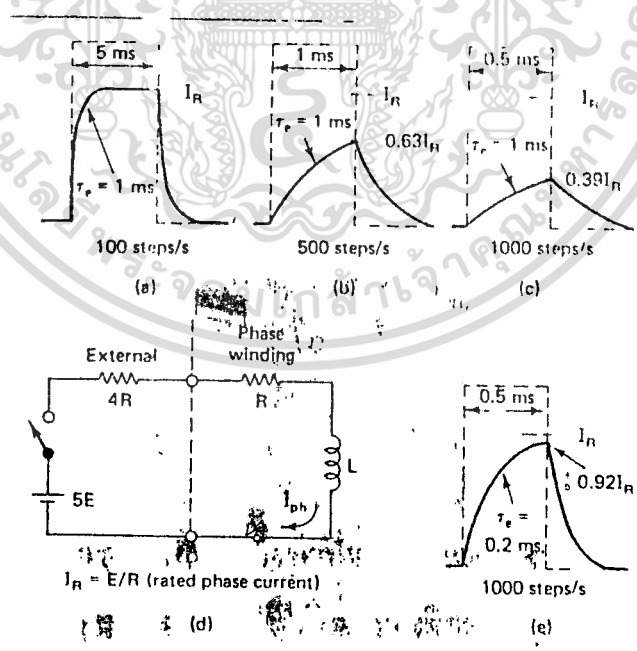
กว่าส่วนเริ่มและหยุดในภาระภายนอกที่เท่ากัน แต่ไม่สามารถที่จะเปลี่ยนทิศทางหรือหยุดในทันทีทันใดได้ เพราะจะทำให้เกิดความผิดพลาดทางด้านตำแหน่งได้

แรงบิดที่มอเตอร์สร้างขึ้นจากรูปที่ 7 เรียกว่า แรงบิดฉุด แรงบิดฉุดนี้จะมีค่ามากสุดเมื่อ ภาระภายนอกมีความสัมพันธ์กับซี่โรเตอร์และขดลวดที่พันอยู่กับสเตเตอร์

จากรูปที่ 3.2 แรงบิดฉุดจะมีค่าลดลงเมื่อความเร็วเพิ่มขึ้น เนื่องจากทฤษฎีพื้นฐานที่ว่า

$$I_{ph} = (E/R)(1 - e^{-t/\tau})$$

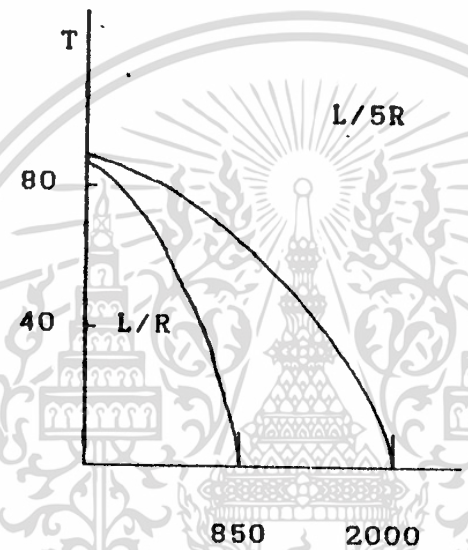
โดยที่ E/R เป็นค่ากระแสสูงสุดของแต่ละเฟส และ $\tau = L/R$ คือค่าเวลาคงที่ของขดลวดที่พันอยู่ในแต่ละเฟส จากรูปที่ 3.3(a) จะเห็นว่าเมื่อมอเตอร์หมุนด้วยความเร็ว100ขั้นต่อวินาทีจะมีคาบเวลาเท่ากับ5ms ส่วนในรูปที่ 3.3(b) และ 3.3(c) มีคาบเวลาเป็น 1 ms และ 0.5 ms ตามลำดับ โดยที่ค่าเวลาคงที่ มีค่าเท่ากับ 1 ms ค่าพลังงานที่ได้จะมีค่ามาก เมื่อคาบเวลาแต่ละ



รูปที่ 3.3 ผลของกระแสกระตุ้นเมื่อความเร็วของมอเตอร์มีค่าต่าง ๆ และเมื่อเพิ่มค่าความต้านทานภายนอกให้กับวงจร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นของมอเตอร์มีค่ามาก เราสามารถที่จะเพิ่มค่ากำลังงานของมอเตอร์ได้โดยเพิ่มค่าความต้านทานภายนอก ดังรูปที่ 3.3(d) แต่เราก็ต้องเพิ่มค่าแหล่งจ่ายไฟตามไปด้วย

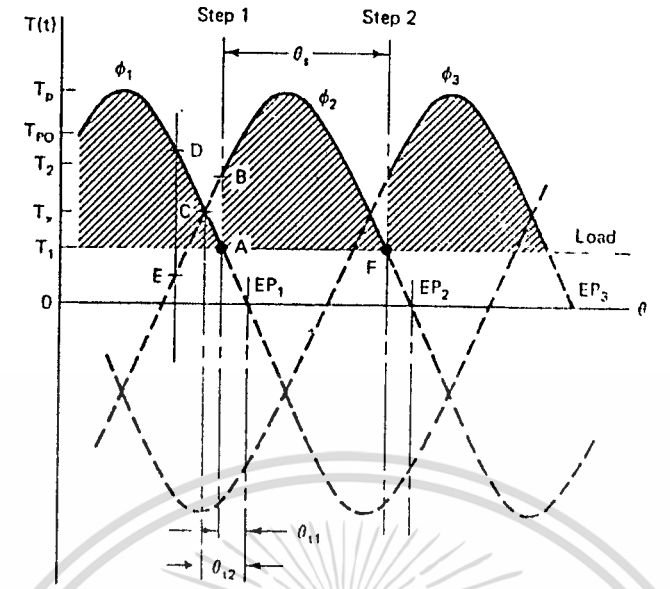


รูปที่ 3.4 กราฟแสดงความสัมพันธ์ระหว่างแรงบิดและความเร็วเมื่อลดค่าเวลาคงที่

พิจารณารูปที่ 3.5 เมื่อเฟสที่ 1 ถูกกระตุ้น และโรเตอร์จะเคลื่อนที่ไปยังตำแหน่งสมดุลย์ของเฟสที่ 1 ที่ EP_1 แต่ก่อนที่จะถึงตำแหน่ง EP_1 เฟสที่ 2 ก็จะถูกกระตุ้น ณ จุด A ทำให้แรงบิดที่เกิดขึ้นจากเฟสที่ 2 เริ่มต้น ณ จุด B ซึ่งจะเร่งการขับภาระ เพื่อที่จะขับเคลื่อนไปยังตำแหน่งต่อไปจนถึงจุด F ซึ่งเฟสที่ 3 ก็จะถูกกระตุ้นวนไปเรื่อย ๆ

ในการที่เราจะกำหนดตำแหน่งที่จะกระตุ้นเฟสต่อไป (θ_c) จะมีส่วนสำคัญในการที่จะแสดงถึงประสิทธิภาพในการผลิตแรงบิดของมอเตอร์ โดยค่ามุมที่มีความเหมาะสมจะทำให้ประสิทธิภาพในการใช้งานของมอเตอร์มีค่ามากขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 การกระตุ้นกระแสของสเต็ปปิ้งมอเตอร์แบบ 3 เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

วงจรอินเตอร์เฟส

4.1 วงจรอินเตอร์เฟส

วงจรอินเตอร์เฟสประกอบไปด้วย วงจรถอดรหัส วงจรแลทช์(latch) วงจรสร้างสัญญาณนาฬิกา วงจรนับ และวงจรโมโนสเตเบิล(monostable)

วงจรถอดรหัส เราใช้พอร์ต(port) หมายเลข 300H ถึง 304H ซึ่งมีรายละเอียดดังนี้

300H	ตัวนับ 0 ของ 8253
301H	ตัวนับ 1 ของ 8253
302H	ตัวนับ 2 ของ 8253
303H	รหัสตัวนับของ 8253
304H	วงจรแลทช์

ในวงจรถอดรหัสนี้เราใช้แอดเดรสบัส (adressbus) จากเครื่องคอมพิวเตอร์มาใช้ในการถอดรหัส เพื่อที่จะทำให้ 74LS138 เลือกพอร์ตต่าง ๆ ได้อย่างถูกต้อง

วงจรแลทช์เราใช้ 74LS374 เพื่อที่จะเก็บค่าแฟลก (flag) ต่าง ๆ ของ 8713 โลจิกซีเควนเซอร์ และสัญญาณเกท(gate) ของ 8253 ทามเมอร์เคาเตอร์

วงจรสร้างสัญญาณนาฬิกาเราใช้ 74LS393 ในการหารสัญญาณจากสัญญาณนาฬิกามาตราฐานจากเครื่องคอมพิวเตอร์ให้เหลือความที่ 55 kHz เพื่อเข้าขาสัญญาณนาฬิกาของตัวนับ 0 ของ 8253

วงจรนับเราใช้ 8253 ทามเมอร์เคาเตอร์ในการนับและสร้างสัญญาณ เพื่อป้อนให้กับเครื่องคอมพิวเตอร์ และนับสัญญาณป้อนกลับเพื่อที่จะนำไปใช้ในการควบคุมมุมนำ

วงจรโมโนสเตเบิล เราใช้ 74LS123 ในการสร้างสัญญาณนาฬิกาจากสัญญาณออกจากตัวเอ็นโค้ดเดอร์ โดยจะตรวจสอบสัญญาณทั้งขาขึ้นและขาลงของสัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ญาณออกของเอ็นโค้ดเดอร์ทั้ง 2 เฟส เพื่อที่จะให้ได้สัญญาณนาฬิกาที่มีความละเอียดขึ้น 4 เท่า ในวงจรนี้เราจำเป็นที่จะต้องใช้ตัวต้านทานและตัวเก็บประจุต่อเข้าไปเพื่อให้ได้ความกว้างของสัญญาณที่สร้างขึ้นตามที่ต้องการ

4.2 การทำงานของวงจรอินเตอร์เฟส

ในการควบคุมแบบเปิด วงจรจะทำงานโดยนำค่าความถี่เข้าไปยังตัวนับ 0 ของ 8253 รวมทั้งสัญญาณเกทที่ได้รับจากเอาท์พุท (output) ของ 74LS374 8253 จะทำงานในแบบที่ 3 คือ ซิกแนลเจนเนอเรชัน (signal generation) คือจะสร้างสัญญาณที่มีความถี่คงที่ออกมา จากนั้นจะนำสัญญาณที่สร้างได้มาเข้า IRQ4 ของเครื่องคอมพิวเตอร์ และจะทำเช่นนี้จนกระทั่งมอเตอร์หมุนไปตามต้องการแล้วเครื่องจะสั่งให้ 74LS374 ให้สัญญาณเกท ของตัวนับ 0 เป็น 0 เพื่อที่จะทำให้ 8253 หยุดนับ

ส่วนในการควบคุมแบบปิด ตอนแรกวงจรจะทำงานเพื่อให้ตัวนับ 0 ของ 8253 สร้างสัญญาณออกมาหนึ่งครั้งก่อน จากนั้นจะทำการนับสัญญาณที่ได้รับจากวงจรโมโนสเตเบิล โดยผ่านทางตัวนับ 2 ของ 8253 เพื่อตรวจสอบดูว่าได้ค่าตามที่ต้องการแล้วหรือยัง ถ้าได้แล้ววงจรจะทำการนำค่า 4 เท่าของ pulse/step ของตัวเอ็นโค้ดเดอร์ เข้าไปยัง 8253 จากนั้นให้ตัวนับที่ 2 ของ 8253 ทำงานในแบบที่ 3 ส่วนสัญญาณขัดจังหวะที่ได้จากตัวนับ 2 มาทาง IRQ3 จะนำมาทำการให้ตัวนับที่ 0 สร้างเอาท์พุทออกไปเพื่อให้มอเตอร์หมุนเป็นจังหวะ ทำจนได้จำนวนขั้นตามที่ต้องการ

4.3 การจัดการขัดจังหวะ

การขัดจังหวะเป็นรูทีน (routine) ที่มีอยู่แล้ว ซึ่งคอมพิวเตอร์เรียกใช้ในงานทั่วไป ซึ่งมีทั้งฮาร์ดแวร์ (hardware) และซอฟท์แวร์ (software) การขัดจังหวะฮาร์ดแวร์ เป็นสิ่งที่สามารถถูกทำให้เกิดขึ้นได้โดยฮาร์ดแวร์ ไม่ว่าจะมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากบนแผ่นของระบบ หรือแผ่นที่เสียบในช่อง I/O ในเมื่อมีการขัดจังหวะทางฮาร์ดแวร์ CPU จะหยุดการทำงานที่มันกำลังทำอยู่ แล้วค่อยมาทำการขัดจังหวะแทน และกลับไปสู่งานเดิมเมื่อได้ทำงานในรูทีนการขัดจังหวะเสร็จแล้ว ดังนั้นเพื่อที่จะสามารถกลับไปทำงานต่อที่เดิมได้ ค่าแอดเดรส (CS:IP) จะถูกผลักลงบนแสตกรวมทั้งตัวแฟลกรีจิสเตอร์ (flag register) ด้วย จากนั้นแอดเดรสของรูทีนการขัดจังหวะจะถูกส่งให้ CS:IP เพื่อที่จะกระทำรูทีนนั้น ๆ ซึ่งส่วนมากเราเรียกรูทีนการขัดจังหวะว่า "การจัดการขัดจังหวะ"

โดยทั่วไปการจัดการขัดจังหวะ จะจบด้วยคำสั่ง IRET ซึ่งจะทำการคืนค่า CS:IP และ แฟลกของการทำงานก่อนเข้าการขัดจังหวะ เพื่อที่จะทำให้โปรแกรมกลับไปทำงานเดิมได้ถูกที่และถูกต้อง

แอดเดรสของการขัดจังหวะถูกเรียกว่า "เวคเตอร์" (Vector) แต่ละเวคเตอร์จะมีขนาด 2 เวิร์ด(word) เวิร์ดแรกจะเก็บค่า IP เวิร์ดหลังจะเก็บค่า CS โดยที่เวกเตอร์เหล่านี้จะเก็บไว้ที่หน่วยความจำ 1024 ไบต์ (byte) ล่างสุด ดังนั้นจึงมีเวกเตอร์ทั้งหมดรวมเป็น 256 เวคเตอร์ ซึ่งรวมกันแล้วเรียกตารางอินเทอร์รัพท์เวคเตอร์ (interupt vector) สำหรับ INT 0H อยู่ที่ 0000:0000, INT 1H อยู่ที่ 0000:0004, INT 2H อยู่ที่ 0000:0008 ถ้าต้องการหาว่า INT 8H จะทำงานที่ไหน จะต้องไปหาว่าที่แอดเดรส 0000:0020 เก็บค่าอะไร ถ้าเก็บค่า A5FE00F0 แสดงว่า CS:IP ของ INT 8H คือ F000:FEA5

4.4 การโปรแกรม 8259 ตัวควบคุมการขัดจังหวะ

อินเทล(Intel) 8259 ซึ่งเป็นตัวควบคุมการขัดจังหวะที่โปรแกรมได้ ปรกติใช้ในคอมพิวเตอร์ เพื่อจะจัดการการขัดจังหวะฮาร์ดแวร์ เพราะว่าการเรียกเข้าการขัดจังหวะ เป็นไปได้ที่จะมีการเรียกเข้ามาพร้อมกันมากกว่า 1 การขัดจังหวะ ดังนั้นจึงจะต้องใช้ 8259 ในการลำดับความสำคัญ ซึ่งใน IBM PC/XT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะมีระดับความสำคัญอยู่ 8 ระดับความสำคัญ ซึ่งมี IRQ0-IRQ7 ซึ่งแทนการเรียกเข้าของการขัดจังหวะ ซึ่งลำดับความสำคัญเป็นดังนี้

IRQ	0	TIMER
	1	KEYBOARD
	2	I/O CHANNEL
	3	COM 1
	4	COM 2
	5	FIXED DISK
	6	DISKETTE CONTROLLER
	7	LPT 1

การที่การขัดจังหวะของวัน เวลา ถูกให้มีลำดับความสำคัญสูงสุด เนื่องจากว่าจะทำให้เวลาของระบบไม่มีความผิดพลาดเกิดขึ้นเนื่องจากการขัดจังหวะของตัวอื่น ๆ

ตัว 8259 มี รีจิสเตอร์ขนาด 1 ไบต์ จำนวน 8 ตัว เพื่อใช้ทำการควบคุมการขัดจังหวะฮาร์ดแวร์ "อินเทอร์รัพท์ รีเควสท์ รีจิสเตอร์" (Interupt Request Register) จะเปลี่ยนบิต (bit) เป็น 1 ต่อเมื่อมีการขัดจังหวะของเส้นนั้นเข้ามา ตัวชิพ (chip) จะตรวจสอบอย่างอัตโนมัติว่า ได้มีการขัดจังหวะตัวอื่น ๆ เข้ามาด้วยหรือไม่ โดยที่มันจะเข้าไปดูที่ "อินเซอร์วิส รีจิสเตอร์" (ISR) สำหรับข้อมูลนี้ และก่อนที่จะทำการขัดจังหวะในขณะนั้น "อินเทอร์รัพท์ แมส รีจิสเตอร์" (Interupt Mask Register,IMR) จะถูกตรวจดูว่าการขัดจังหวะลำดับนั้นในปัจจุบันได้รับการยอมรับหรือไม่ ซึ่งการติดต่อกับ IMR จะติดต่อผ่านทางพอร์ต 21H ซึ่ง IMR จะมีลักษณะดังตารางที่ 4.1 นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B7	B6	B5	B4	B3	B2	B1	B0
IRQ7	IRQ6	IRQ5	IRQ4	IRQ3	IRQ2	IRQ1	IRQ0

ตารางที่ 4.1 ตารางแสดงค่าของ IRQ

ซึ่งถ้าเราเซต (set) บิตไหนจะทำให้ IRQ ที่บิตนั้นไม่ถูกยอมรับ ดังตัวอย่าง

```
MOV  AL,01000000B
OUT  21H,AL
```

จะเป็นการทำให้ IRQ6 ไม่ถูกยอมรับ

และหลังจากทำการขัดจังหวะฮาร์ดแวร์การจัดการขัดจังหวะและ เราต้องส่งค่าให้ 8259 รู้ว่าเราได้ทำการขัดจังหวะเสร็จแล้ว โดยส่งค่าออกทางพอร์ต 20H ด้วยค่า 20H ซึ่งโปรมแกรมโดย

```
MOV  AL,20H
OUT  20H,AL
```

ซึ่งถ้าไม่ทำเช่นนี้แล้ว 8259 จะไม่เคลียร์ (clear) ISR ซึ่งจะทำให้มันทำการขัดจังหวะของการขัดจังหวะในระดับต่ำกว่ามัน

4.5 ข้อควรระวังในการเขียนการจัดการการขัดจังหวะฮาร์ดแวร์

- เนื่องจาก MS-DOS ไม่สามารถเรียกเข้าตัวเองได้ การจัดการขัดจังหวะจะต้องไม่เรียกรูปแบบคำสั่งของ MS-DOS ระหว่างการทำการขัดจังหวะ

- เนื่องจากเมื่อเกิดการขัดจังหวะขึ้นจะมีการทำให้การขัดจังหวะอื่นไม่สามารถทำงานได้ ดังนั้นเมื่อเข้าการจัดการขัดจังหวะแล้วจะต้องรีบทำให้การขัดจังหวะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อื่นสามารถทำงานได้ เพื่อที่จะทำให้การขัดจังหวะฮาร์ดแวร์ที่มีลำดับสูงกว่าทำงานได้

- การโปรแกรม 8259 ควรทำด้วยความระมัดระวังและก่อนออกจากการจัดการขัดจังหวะต้องส่งค่า EOI (20H) ไปยัง 8259 ด้วยมิฉะนั้นจะทำให้การจัดการขัดจังหวะในลำดับที่ต่ำกว่าและลำดับเดียวกันไม่สามารถทำงานได้

4.6 ทามเมอร์เคาเตอร์ 8253

ไอซี 8253 เป็นไอซีที่ทำหน้าที่เป็นตัวนับและตัวสร้างสัญญาณพัลส์ที่สามารถโปรแกรมได้ โดยมีตัวนับและตัวสร้างสัญญาณพัลส์อยู่ภายในจำนวน 3 ตัว การกำหนดให้ทำหน้าที่ตัวนับหรือตัวสร้างสัญญาณพัลส์ สามารถทำได้โดยสั่งผ่านทางรีจิสเตอร์ควบคุม (control register) ตัวไอซี 8253 ใช้แอดเดรสจำนวน 4 ตำแหน่ง โดย 3 ตำแหน่งแรกเป็นการค่านับให้กับตัวนับที่ 0, 1 และ 2 ตามลำดับ ส่วนตำแหน่งที่ 4 เป็นตำแหน่งของรีจิสเตอร์ควบคุม

4.7 โหมดการทำงานของ 8253

การใช้งาน 8253 ในโหมดต่าง ๆ มีดังนี้

1. โหมด 0 การนับเพื่อสร้างสัญญาณอินเทอร์รัพท์

การทำงานแบบนับลงตามความถี่สัญญาณนาฬิกาที่ป้อนเข้ามา เมื่อนับถึง 0 ก็จะให้เอาท์พุทที่ขาเอาท์เป็น 1 และจะกลับเป็น 0 เมื่อป้อนค่านับใหม่ให้กับตัวนับ

2. โหมด 1 การสร้างพัลส์ที่โปรแกรมช่วงเวลาได้

การทำงานจะสร้างพัลส์ที่สามารถกำหนดความกว้างได้ โดยเมื่อมีการกระตุ้นที่ขาเกทก็จะให้เอาท์พุทเป็น 0 เท่ากับจำนวนพัลส์ของสัญญาณนาฬิกา ซึ่งเราสามารถโปรแกรมจำนวนพัลส์ได้

3. โหมด 2 ตัวกำเนิดสัญญาณ

การทำงานจะหารสัญญาณนาฬิกาด้วย N ที่โปรแกรมเข้ามา โดยรูปแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอาท์พุทจะมี 0 อยู่ 1 พัลส์ต่อ 1 จำนวน N พัลส์

4. โหมด 3 ตัวกำเนิดสัญญาณรูปสี่เหลี่ยม

การทำงานคล้ายกับโหมด 2 แต่รูปแบบเอาท์พุทจะมีลักษณะที่ 0 และ 1 ที่สมมาตรกัน

5. โหมด 4 การหน่วงเวลาด้วยซอฟต์แวร์

การทำงานในโหมดนี้ตัวนับจะให้เอาท์พุทจำนวน 1 พัลส์ โดยที่มีการหน่วงเวลาออกไปเท่ากับจำนวนพัลส์ที่โปรแกรมไว้

6. โหมด 5 การหน่วงเวลาด้วยฮาร์ดแวร์

การทำงานเหมือนกับโหมด 4 แต่การกระตุ้นให้เริ่มหน่วงเวลาต้องทำโดยป้อนสัญญาณอินพุทขาขึ้นที่ขาเกท ก็จะเริ่มนับ และเมื่อนับครบแล้วก็จะให้เอาท์พุทเป็น 0 จำนวน 1 พัลส์

4.8 วงจรลอจิกซีเควนเซอร์

วงจรลอจิกซีเควนเซอร์ มีหน้าที่ในการควบคุมการส่งพัลส์ออกไปขับขดลวดสเตปปิ้งมอเตอร์แต่ละเฟสอย่างสัมพันธ์กัน ให้เป็นไปตามรูปแบบการขับและทิศทางการหมุนที่ต้องการ การควบคุมความเร็วโดยความถี่ของสัญญาณพัลส์ที่ป้อนให้วงจร โดยทั่วไปประกอบด้วย รีจิสเตอร์(register)เลื่อนข้อมูลและลอจิกเกท(logic gate)

ส่วนวงจรลอจิกซีเควนเซอร์ที่ใช้ในโครงงานนี้ ใช้ไอซีลอจิกซีเควนเซอร์สำเร็จรูป คือ PMM8713 ของ บริษัทซันโย เดนกิ จำกัดแห่งประเทศญี่ปุ่น โดยเป็นไอซีประเภท ซีมอส โมโนลิทิค(CMOS Monolithic) สามารถขับสเตปปิ้งมอเตอร์ได้ทั้งแบบ 3 เฟส และ 4 เฟส และสามารถขับได้ทั้งแบบกระตุ้น 1 เฟส, กระตุ้น 2 เฟส และกระตุ้น 1-2 เฟส

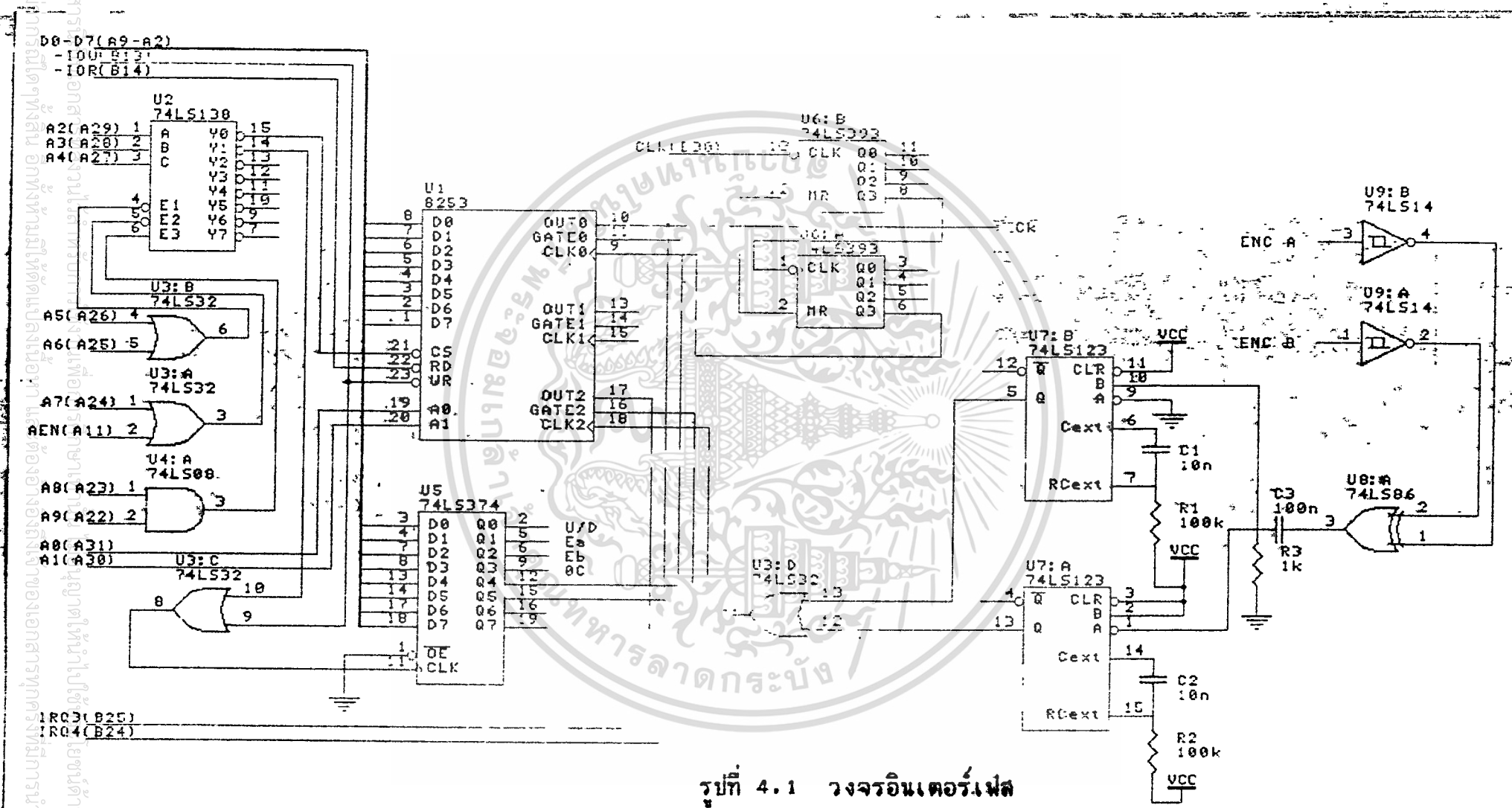
การนำไปใช้งานทำได้ง่ายมาก เพียงแต่ต่อทรานซิสเตอร์ที่เอาท์พุทของ 8713 ก็จะสามารถจ่ายกระแสที่อัตราสูงให้สเตปปิ้งมอเตอร์ได้ เพียงแต่ให้ความ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถี่ที่ต้องการขับ รูปแบบการขับ และทิศทางการขับ

ในวงจรทดลองที่ได้สร้างขึ้นนั้น เราใช้สัญญาณนาฬิกาที่มาจาก 8253 ในการกระตุ้นสัญญาณให้กับ 8713 เพื่อที่จะควบคุมการทำงานของมอเตอร์ สำหรับรายละเอียด และลักษณะทั่วไปของ 8713 ดูเพิ่มเติมได้จากภาคผนวก ข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

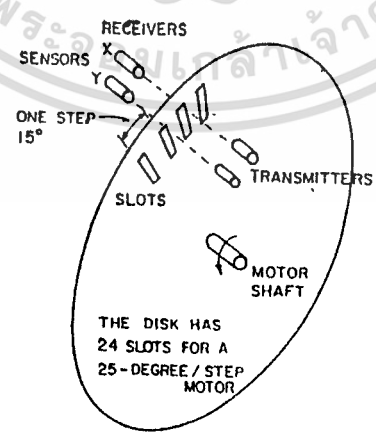


รูปที่ 4.1 วงจรอินเตอร์เฟส

บทที่ 5
วงจรขับและเอ็นโค้ดเดอร์

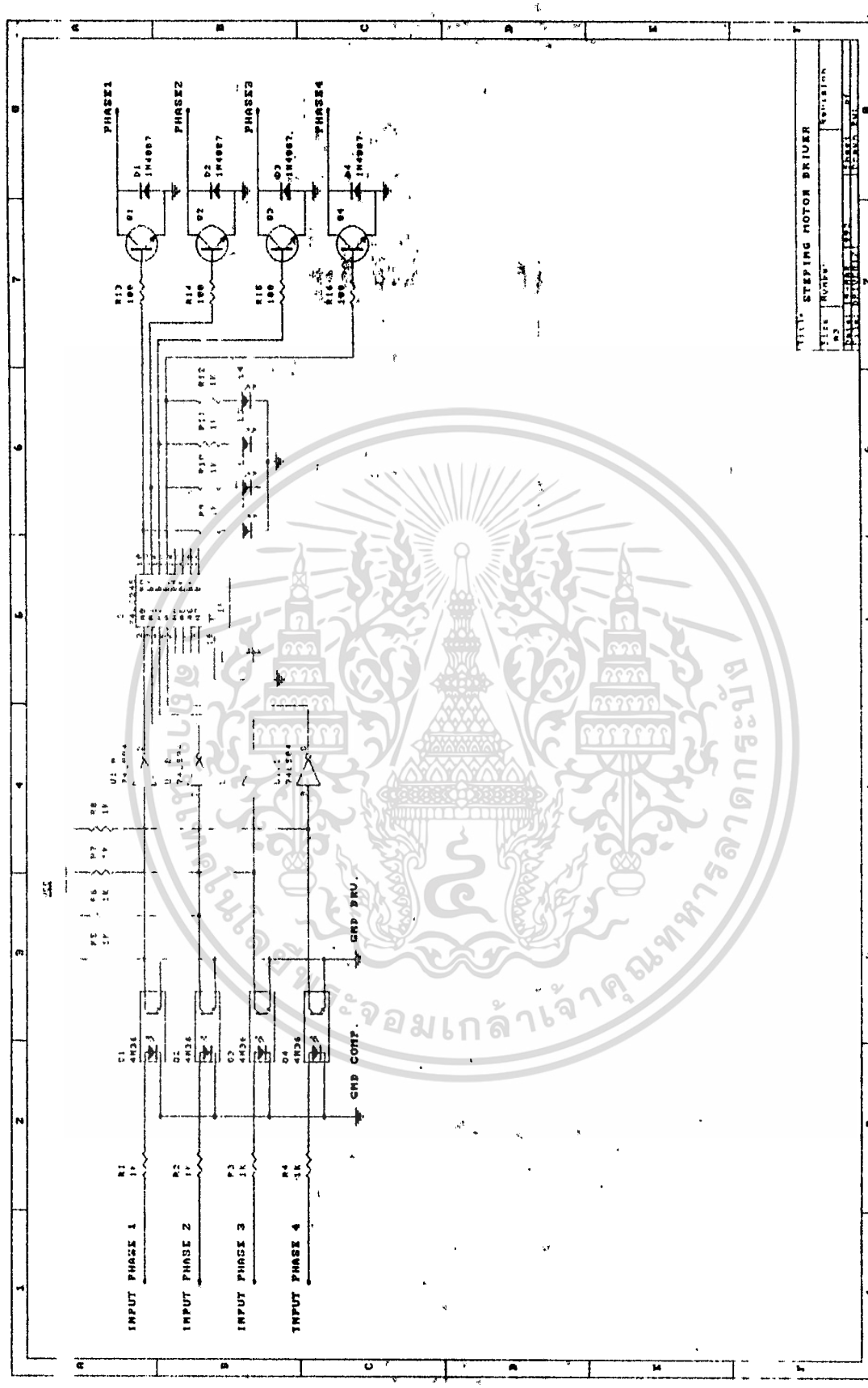
วงจรขับใช้ TIP 3055 จำนวน 4 ตัวเป็นตัวจ่ายกระแสโดยรับสัญญาณจากวงจรอินเตอร์เฟส ซึ่งมีทั้งสิ้น 4 เฟสผ่านตัวแยกสัญญาณ 4N36 เพื่อที่จะป้องกันมิให้เครื่องคอมพิวเตอร์ต้องจ่ายกระแสมากจนเกินไป

เอ็นโค้ดเดอร์เป็นตัววัดสัญญาณตำแหน่งของมอเตอร์โดยใช้แสงส่องผ่านช่องว่างทำให้เกิดสัญญาณไฟฟ้าขึ้น โดยจะใช้ตัวจับสัญญาณ 2 ตัว ดังรูปที่ 5.1



รูปที่ 5.1 เอ็นโค้ดเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



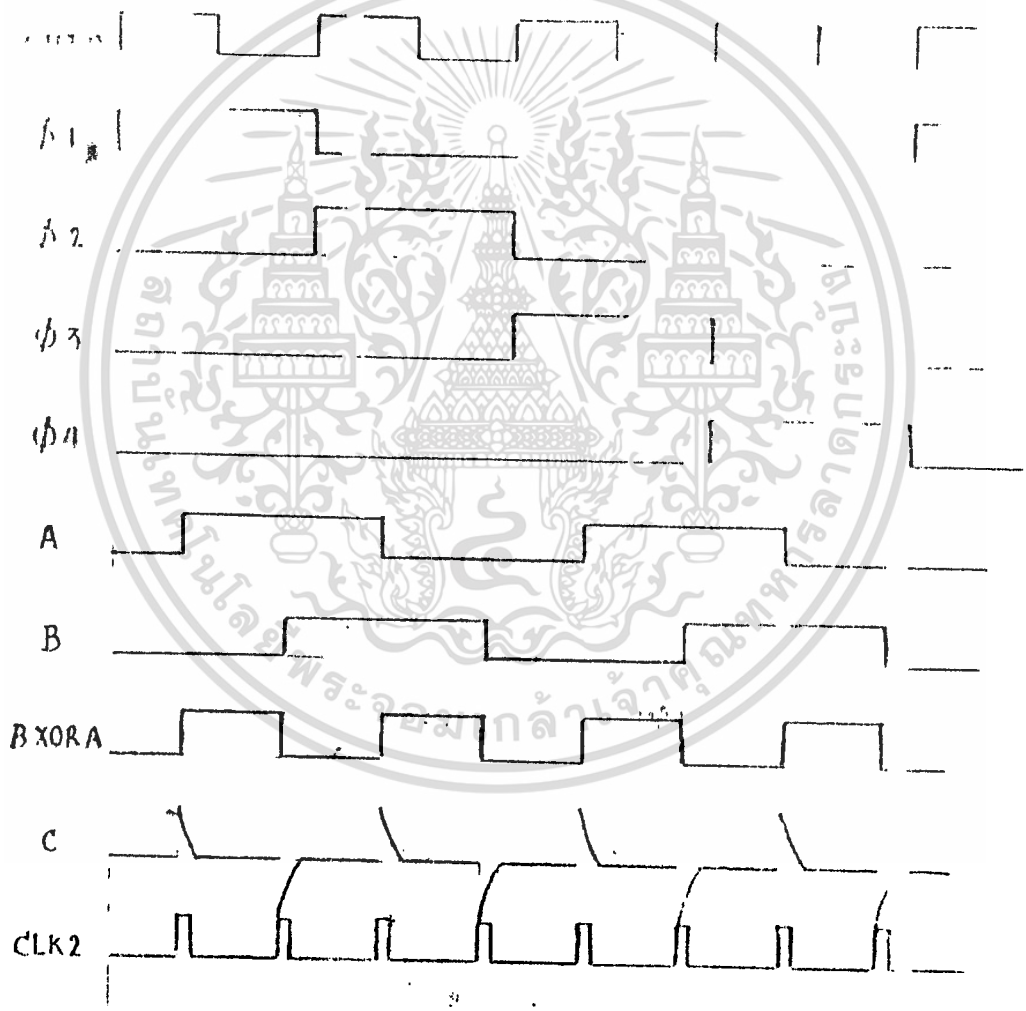
รูปที่ 5.2 วงจรขับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

บทที่ 6

ผลการทดลอง

จากการทดลองได้กำหนดความถี่ในการทำงาน แล้ววัดค่าสัญญาณที่ได้ออกมา ดังรูปที่ 6.1



รูปที่ 6.1 ผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการทดลองเราได้ลองให้มอเตอร์ทำงานที่ความถี่ต่าง ๆ กัน ในขณะที่จำนวนขั้นในการหมุนของมอเตอร์คงที่ เราจะได้ว่าในการทำงานแบบวงปิดนั้น มอเตอร์สามารถที่จะทำงานได้จนถึงความถี่ที่ 800 Hz ซึ่งมากกว่าในการทำงานแบบเปิด

ความถี่(Hz)	การควบคุม	เวลา(ms)	กระแส (A)
100	แบบเปิด	507	0.6
200	แบบเปิด	253	0.4
350	แบบเปิด	146	0.1
450	แบบเปิด	114	0.07
600	แบบเปิด	ไม่หมุน	0.0
450	แบบปิด	109	0.3
600	แบบปิด	81	0.18
700	แบบปิด	68	0.13
800	แบบปิด	66	0.12

ตารางที่ 6.1 ตารางแสดงผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
สรุปผลและวิจารณ์

จากผลการทดลอง แสดงให้เห็นว่า วงจรอินเตอร์เฟสทำงานได้ถูกต้องตามการออกแบบ ส่วนสัญญาณขัดจังหวะนั้นทำงานได้ทั้งสองสัญญาณ ซึ่งในการทำงานแบบระบบเปิดสามารถทำงานได้โดยไม่มีปัญหา ส่วนการทำงานแบบระบบปิดการทำงานยังไม่เป็นไปตามความต้องการ ซึ่งอาจมีสาเหตุจากตัวคอมพิวเตอร์ที่ใช้ในการทดลองมีความเร็วต่ำ จึงทำให้การตอบสนองการควบคุมได้ไม่ดีพอ หรืออาจเป็นเพราะ ตัวเอ็นโค้ดเดอร์มีความผิดพลาด โดยที่ความเร็วต่ำ ถ้าเราให้สเตปมอเตอร์หมุนไปเพียงเล็กน้อย จะได้จำนวนพัลส์จากเอ็นโค้ดเดอร์มากผิดปกติ แต่ถ้าทำที่ความเร็วประมาณ 300 Hz จะให้ผลการควบคุมตามต้องการ แต่ถ้าขับมอเตอร์ที่ความเร็วมากกว่า 300 Hz จะทำให้จำนวนพัลส์ที่ได้รับจากเอ็นโค้ดเดอร์ผิดพลาดไปประมาณ 5 พัลส์ต่อการเคลื่อนที่ของเพลามอเตอร์ 30000 สเตป ในการวัดความเร็วของการทำงานแบบระบบปิดและระบบเปิด วัดเวลาในการทำงานในช่วงสภาวะคงตัว เพื่อที่จะเปรียบเทียบว่า การควบคุมแบบระบบเปิดกับระบบปิดแตกต่างกันอย่างไร ส่วนการเปรียบเทียบแรงบิดของการควบคุมแบบระบบปิดกับระบบเปิด จะใช้กระแสที่ไหลเข้ามอเตอร์เป็นพารามิเตอร์ในการพิจารณาค่าแรงบิด

การพัฒนาสามารถทำได้โดยการปรับปรุงให้เอ็นโค้ดเดอร์ที่มีความเที่ยงตรงและละเอียดขึ้น และใช้คอมพิวเตอร์ที่มีความเร็วสูงขึ้นเพื่อที่จะสามารถรับสัญญาณขัดจังหวะจากระบบมอเตอร์ได้เร็วขึ้น ซึ่งอาจทำให้การทำงานแบบระบบปิดมีผลเป็นที่หน้าพอใจ และสามารถนำไปใช้ควบคุมในระบบที่ต้องการความเที่ยงตรงทางตำแหน่งและต้องการแรงบิดสูงต่อไปได้ ตัวอย่างเช่น ระบบควบคุมการเคลื่อนที่ของแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

[1] ธนินทร์ ถาวรศาสนวงศ์ และ ทินกร ดุ๊ก, "การอินเทอร์เฟส IBM PC", ฟิสิกส์เซ็นเตอร์การพิมพ์.

[2] Gayakwad Ramakant and Leonard Sokoloff, "ANALOG AND DIGITAL CONTROL SYSTEMS", Prentice-Hall International, Inc., 1983

[3] B. Kuo, "ANALOG AND DIGITAL CONTROL", Prentice-Hall International, Inc., 1986.

[4] Takashi Kenjo,"STEPPING MOTORS AND THEIR MICROPROCESSOR CONTROLS", Oxford Science Publications, 1986.

[5] James W. Coffron, "Z80 APPLICATION", SYBEX Inc.,1983.

[6] Douglas V. Hall, "MICROPROCESSORS AND INTERFACING", McGraw-Hill Book Co.,1986.

[7] "PC/AT TECHNICAL REFERENCE", 1984.

[8] "MICROSOFT QuickC", Microsoft, 1987.

[9] "TURBO C 2.0 Reference and User's Guide", Borland.

[10] Kent Porter, "STRETCHING TURBO C", Californin, U.S., 1987.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

```
/* openloop */

#include <dos.h>


#define counter0_8253 0x300

#define counter2_82530x302

#define control_word  0x303

#define status_8713     0x304

#define imr 82590x21

#define [illegible]55c3

#define [illegible]1/* four phase only */

#define [illegible]0/* two excite only */

#define [illegible]1.0

#define n2


/* Global variable */

int step, sto;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*  function */
void interrupt (*oldvect0c)(void);
void interrupt (*oldvect0b)(void);
void interrupt extern openloop(void);
void interrupt extern closeloop(void);
void set8253 c0_m0_f(void);
void set8253 c0_m3_f(void);
void set8253 c2_m0_1(void);
void set8253 c2_m3_1(void);
int  flag(int g0, int g2, int ud);
void set8713(int setv);
void out50hz(int setv);
void install(void);
void restore(void);
void stop(void);


void set8253 c0 m0_f(void)
{
   outportb(control_word, 0x30);       /* mode 0 counter 0 */
}


void set8253 c0_m3_f(void)
{
   outportb(control_word, 0x36);       /* mode 3 counter 0 */
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void set8253_c2_m0_1(void)
{
  outportb(control_word, 0x90); /* mode 0 counter 2  half */
}

void set8253_c2_m3_1(void)
{
  outportb(control_word, 0x96); /* mode 3 counter 2 half */
}

int  flag(int g0, int g2, int dir)
{
  int var;
  var = 32*g2 + 16*g0 + 8*zc + 4*eb + 2*ea + dir;
  return(var);
}

void set8713(int setv)
{
  outportb(status_8713, setv);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void out50hz(int setv)
{
  init_8253_c0_m0_f();
  outportb(counter0_8253, 0x4c);   /* for 50 Hz */
  outportb(counter0_8253, 0x04);
  outportb(imr_8259,sto && 0xef);
  out8713(setv);
}

void install(void)
{
  oldvect0c = getvect(0x0c);
  oldvect0b = getvect(0x0b);
  setvect(0x0c,openloop);
  setvect(0x0b,closeloop);
}

void restore(void)
{
  setvect(0x0c,oldvect0c);
  setvect(0x0b,oldvect0b);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void stop(void)
{
  outport(status_8713, 0);
}

  main(void)
{
  int ud, set, gate0, gate2;
  int ans, freq, count;
  int step_end, step_steady, delay_ang, tstep;
  int outlow, outhigh;
  char ansc;
  gate0 = 0; gate2 = 0; tstep = 0;ansc = ' ';
  printf("\n direction (CW = 0, CCW = 1) :");
  scanf("%d",&ans);
  ud = ans == 0 ? 0 : 1;
  printf("\n frequency max :");
  scanf("%d",&freq);
  count = F / freq;
  printf("\n step :");
  scanf("%d",&step_end);
  printf("\n Control mode ( Openloop: 0, Closeloop: 1) :");
  ans  = (char)getch();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if (ansc == 'c' || ansc == 'C')
    {
      printf("\n delay angle : ");
      scanf("%d",&delay_ang);
    }

    outlow = count % 256;
    outhigh = count / 256;
    step_steady = step_end - 2;
    slo = inportb(imr_8259);
    install();

/* start up interval */
    gate0 = 1; gate2 = 0; step = 0;
    set = flag(gate0, gate2, ud);
    outb(0x61, set);
    while(step == 0);
    step = step;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้ ... ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้าม ... ถึงเจ้าของเอกสารก่อนครั้งที่จะนำเอกสารนี้ไปใช้

```
/* steady state interval */
  if(ansc == 'o' || ansc == 'c')          /* closeloop control */
  {
    gate0 = 1; gate2 = 1; step = 0;
    set = flag(gate0, gate2, ud);
    set8253_c2_m0_1();
    outportb(counter2_8253, delay_ang);
    set8253_c0_m0_1();
    outportb(counter0_8253,outlow);
    outportb(counter0_8253,outhigh);
    outportb(imr_8259,sto && 0xf7);       /* enable IRQ 3 */
    set8713(set);
    while(step == 0);                     /* waiting for interrupt */
    set8253_c2_m3_1();
    outportb(counter2_8253, n);
  while(step < step_steady)
    printf("c");
}
else                                      /* openloop control */
{
  step = 0;
  set8713(step);
  outportb(imr_8259, sto && 0xef);
  set8253_c0_m3_1();
  outportb(counter0_8253, outlow);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        outportb(counter0_8253, outhigh);
        set8713(set);
        while(step < step_steady)
          printf("o");
      }
      step += step;

/*    low down interval */
      outportb(imr_8259,sto | 0x18);/* disable IRQ3, IRQ4 */
      step = 0; gate2 = 0; step = 0;
      set = flag(gate0, gate2, ud);
      out50hz(set);
      while(step == 0);
      tstep += step;
      stop();
      restore();
      outportb(imr_8259,sto);/* restore IMR of 8259 */
      printf("\n Total step : %d\n",tstep);
}
```

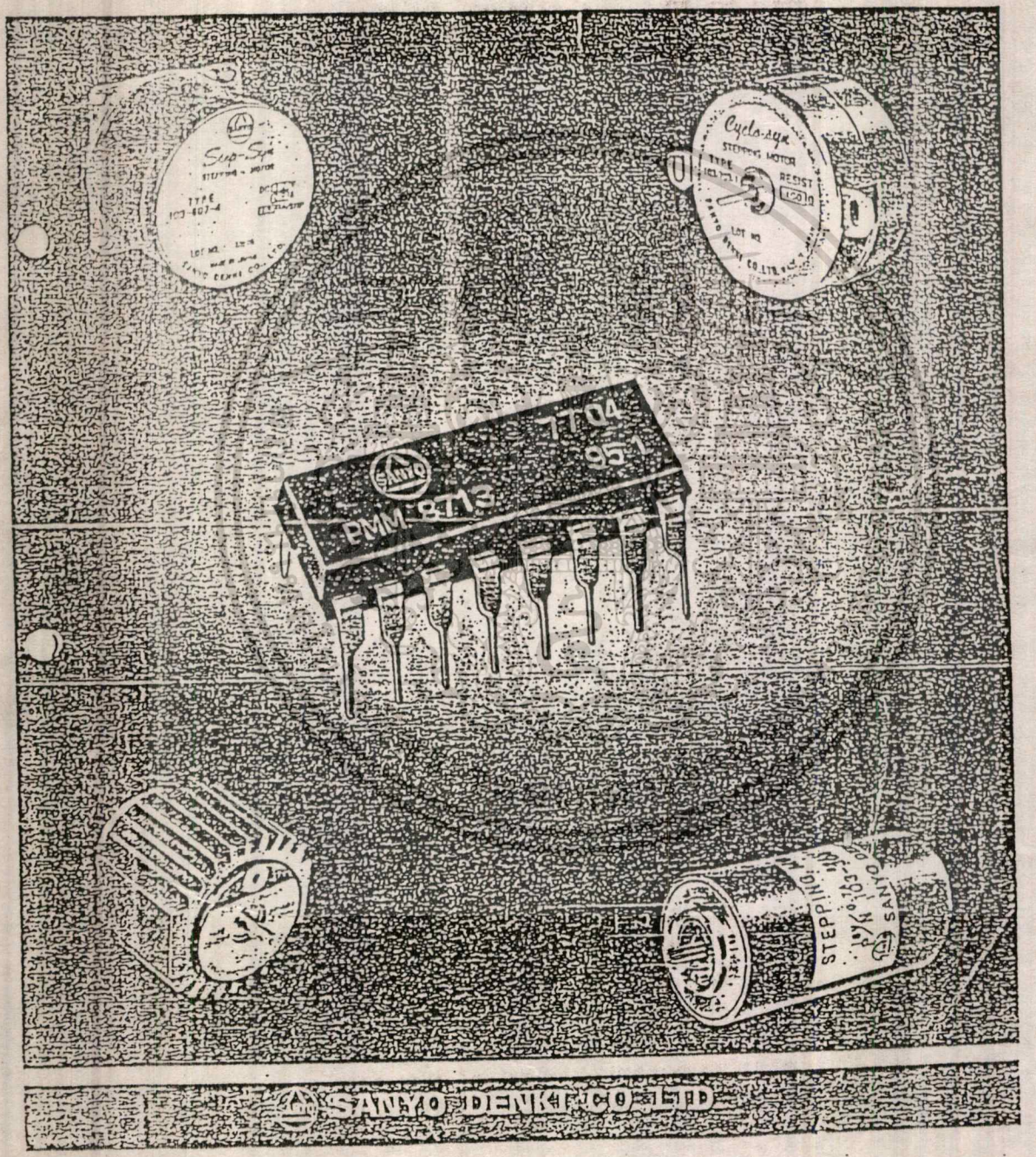
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

UNIVERSAL CONTROLLER

for Stepping Motor Driving

PMM8713



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

① GENERAL

The universal controller PMM8713 for driving SANYO stepping motor is a C-MOS monolithic type IC designed for controlling stepping motors of three-phase and four-phase type.
The Sanyo's PMM8713 is designed to control either of 3 phase and 4 phase stepping motors in any mode of 1 phase excitation, 2 phase excitation or 1–2 phase excitation.
In addition to that, when darlington type transistors are only inserted in the output lines, it can work on switching of higher current ratings of stepping motors.
PMM8713 has been developed for the purpose of simplifying the utilization of stepping motors and it permits to easily compose a stepping motor drive unit by preparing a pulse oscillator (pulse input signal), power switching transistors and DC power supply.
It is chief advantages that it affords cost savings in parts inventory, purchasing, handling, assembly labor, circuit test and design.

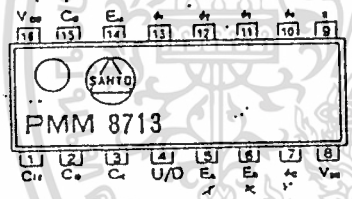
② FEATURES

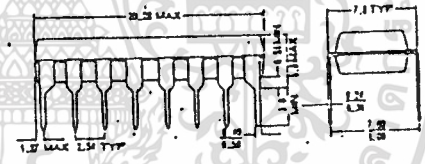
★ Universal controller: The following 6 types of mode can be selected by means of a excitation mode change-over terminal:
- 4 phase 1 excitation
- 4 phase 2 excitation
- 4 phase 1–2 excitation
- 3 phase 1 excitation
- 3 phase 2 excitation
- 3 phase 1–2 excitation

★ Wider range of power supply voltage: V_{DD} = +4V ~ +18V
★ High output current: 20mA min. for both sink and source
★ High noise margin: All input terminals have built-in schmitt circuit
★ 2 types of pulse input: 2 input terminal system / 1 input, 1 change-over terminal system — either one selectable
★ Excitation detecting monitor: Outputs monitor signal showing operating condition of controller

③ PIN CONNECTION (16 LEAD PLASTIC DUAL IN LINE PACKAGE)

(Fig.-1)





PIN NO.		
1	C_U	Input pulse, UP clock
2	C_D	Input pulse, DOWN clock
3	C_X	Input pulse, clock
4	U/D	Change-over of rotation direction "0" DOWN, "1" UP
5	E_A	Exitation-mode change over (See Table-1)
6	E_B	Exitation-mode change over (See Table-1)
7	ϕ_C	3, 4 phase-change (See Table-1)
8	V_{SS}	(GND)
9	$\bar{R}$	Reset
10	ϕ_4	Output (4 phase)
11	ϕ_3	Output (4 phase)
12	ϕ_2	Output (3 phase / 4 phase)
13	ϕ_1	Output (3 phase / 4 phase)
14	E_M	Excitation moniter
15	C_O	Input pulse monitor
16	V_{DD}	(+4 ~ +18V)

④ FUNCTION TABLE

(Table 1)

Excitation Mode		Input ϕ_C	E_A	E_B	C_U	C_D	C_X	U/D	$\bar{R}$	Output C_O	E_M	ϕ_1	ϕ_2	ϕ_3	ϕ_4
3 phase	1–2 mode	0	1	1	×	×	×	×	0	—	1	1	0	1	0
	2 mode	0	0	0	×	×	×	×	0	—	1	1	0	1	0
	1 mode	0	0/1 OR 1/0		×	×	×	×	0	—	0	1	0	0	0
4 phase	1–2 mode	1	1	1	×	×	×	×	0	—	1	1	0	0	1
	2 mode	1	0	0	×	×	×	×	0	—	1	1	0	0	1
	1 mode	1	0/1 OR 1/0		×	×	×	×	0	—	0	1	0	0	0

× : Don't care — : Indefinite

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

⑯ EXCITATION SEQUENCE (Table-2)

4 Phase 2 Excitation (a)

	0 (Reset)	1	2	3	4
ϕ_1	1	1	0	0	1
ϕ_2	0	1	1	0	0
ϕ_3	0	0	1	1	0
ϕ_4	1	0	0	1	1
UP					
DOWN					

3 Phase 2 Excitation (b)

	0 (Reset)	1	2	3
ϕ_1	1	1	0	1
ϕ_2	0	1	1	0
ϕ_3	1	0	1	1
UP				
DOWN				

4 Phase 1-2 Excitation (c)

	0 (Reset)	1	2	3	4	5	6	7	8
ϕ_1	1	1	1	0	0	0	0	0	1
ϕ_2	0	0	1	1	1	0	0	0	0
ϕ_3	0	0	0	0	1	1	1	0	0
ϕ_4	1	0	0	0	0	0	1	1	1
UP									
DOWN									

3 Phase 1-2 Excitation (d)

	0 (Reset)	1	2	3	4	5	6
ϕ_1	1	1	1	0	0	0	1
ϕ_2	0	0	1	1	1	0	0
ϕ_3	1	0	0	0	1	1	1
UP							
DOWN							

4 Phase 1 Excitation (e)

	0 (Reset)	1	2	3	4
ϕ_1	1	0	0	0	1
ϕ_2	0	1	0	0	0
ϕ_3	0	0	1	0	0
ϕ_4	0	0	0	1	0
UP					
DOWN					

3 Phase 1 Excitation (f)

	0 (Reset)	1	2	3
ϕ_1	1	0	0	1
ϕ_2	0	1	0	0
ϕ_3	0	0	1	0
UP				
DOWN				

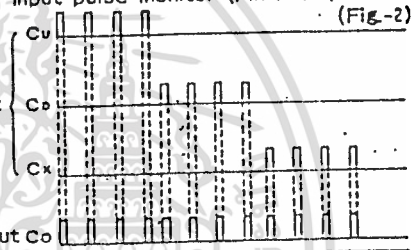
⑰ FUNCTION OF EXCITATION MONITOR (PIN NO.14) AND INPUT PULSE MONITOR (PIN NO.15)

Excitation monitor (Pin No.14) $-E_M$ (Table-3)

Mode \ Input Pulse	0 (Reset)	1	2	3	4	5	6	7	8
3, 4 phase 1 excitation	0	0	0	0	0	0	0	0	0
3, 4 phase 2 excitation	1	1	1	1	1	1	1	1	1
3, 4 phase 1-2 excitation	1	0	1	0	1	0	1	0	1

That is, transmits "1" level when output mode is set at 2 excitation, and transmits "0" level when 1 excitation.

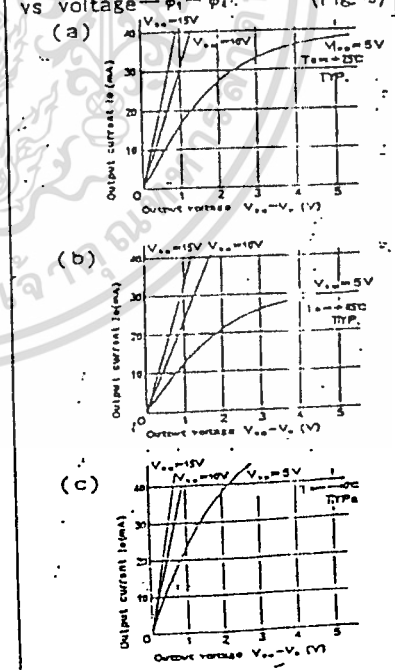
Input pulse monitor (Pin No.15)——Co (Fig-2)



⑰ MAXIMUM RATINGS (Table-4)

Parameter	Symbol	Ratings	Units
Supply Voltage	V_{DD}	$-0.5 \sim +18$	V
Input Voltage	V_{IN}	$-0.5 \sim V_{DD}+0.5$	V
Operating temperature range	T_A	$-40 \sim +85$	°C
Storage temperature range	T_{STG}	$-65 \sim +150$	°C
Power Dissipation	P_D	280 (T_A = -40°C $\sim +60$°C) / 160 (T_A = $+60$°C $\sim +85$°C)	mW
Output current	I_O	35	mA

Characteristic of output current vs voltage — $\phi_1 \sim \phi_4$ (Fig-3)



⑱ ELECTRICAL CHARACTERISTICS

DC characteristics T_A = -40°C $\sim +85$°C (Table-5)

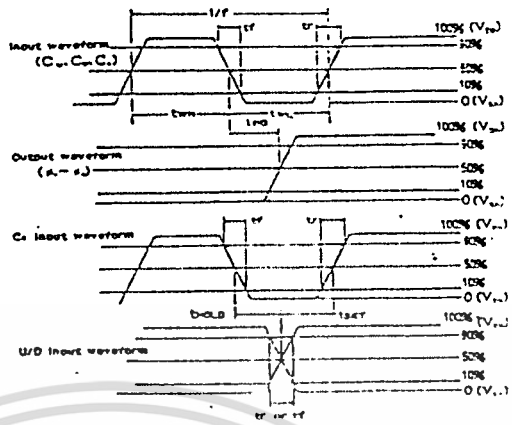
Parameter		Symbol		Test Conditions	Limits Min.	Limits Typ.	Limits Max.	Units
Input voltage	"H" level	V_{IH}	ST		3.3			V
			LT		11			
	"L" level	V_{IL}	ST				1.5	
			LT				4.0	
Output voltage	"H" level	V_{OH}	ST	V_{DD}=5V, I_{OH}=−1mA	4.9			V
			LT	V_{DD}=10V, I_{OH}=−2mA	10.0			
	"L" level	V_{OL}	ST	V_{DD}=5V, I_{OL}=1mA			0.1	
			LT	V_{DD}=10V, I_{OL}=2mA			0.2	
Output current (Co, Cx Output)	"H" level	I_{OH}	ST	V_{DD}=5V, V_{OH}=4.6V	−2			mA
			LT	V_{DD}=10V, V_{OH}=9.5V	−8			
	"L" level	I_{OL}	ST	V_{DD}=5V, V_{OL}=0.4V	2			
			LT	V_{DD}=10V, V_{OL}=0.5V	8			
Output current (Cu Output)	"H" level	I_{OH}	ST	V_{DD}=5V, V_{OH}=4.6V	−0.8			
			LT	V_{DD}=10V, V_{OH}=9.5V	−1.6			
	"L" level	I_{OL}	ST	V_{DD}=5V, V_{OL}=0.4V	1.6			
			LT	V_{DD}=10V, V_{OL}=0.5V	3.6			
Input current		I_I	LT	V_{IN}=0V			10	μA
Stand-by current		I_{DD}	LT	V_{DD}=10V, V_{IN}=0V			1.0	mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC characteristics (Table-6)

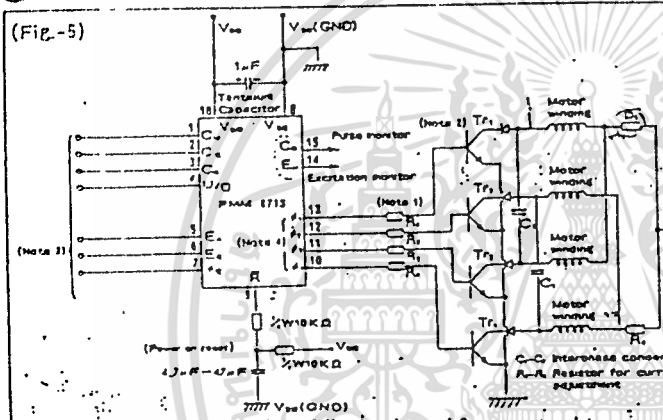
Parameter	Symbol	Test Condition V_{DD}		Limits Min.	Typ.	Max.	Units
Max. clock frequency	f_{max}	5 V	$t_r = t_f = 20ns$	1			MHz
		15V	$C_L = 50PF$	2			
Min. clock pulse width	t_{wc}	5 V	$t_r = t_f = 20ns$			500	ns
	t_{wc}	15V	$C_L = 50PF$			250	
Min. reset pulse width	t_{wr}	5 V	$t_r = t_f = 20ns$			1000	
	t_{wr}	15V	$C_L = 50PF$			500	
Transmission delay time (from clock input to Q output)	t_{PD}	5 V	$t_r = t_f = 20ns$			2000	
	t_{PD}	15V	$C_L = 50PF$			1000	
Set up time	t_{SET}	5 V	$t_r = t_f = 20ns$	0			
	t_{SET}	15V	$C_L = 50PF$	0			
Hold time	t_{HOLD}	5 V	$t_r = t_f = 20ns$	250			
	t_{HOLD}	15V	$C_L = 50PF$	125			

Input, output waveforms (Fig.-4)



APPLICATION EXAMPLE

(Fig.-5)

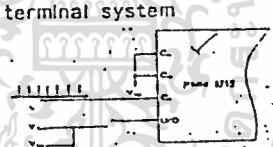


(Fig.-6)

2 input terminal system

(Fig.-7)

1 input, 1 change-over terminal system



(Note 1) Recommended the followings be used for output resistors

R_1 to R_4 : ½W 100Ω ±5% when V_{DD} = 5V
1 W 300Ω ±5% when V_{DD} = 10V
1 W 500Ω ±5% when V_{DD} = 15V

The output current at this time will be about I_{OH} = 15 ~ 27mA. (with Tr $V_{BE(SAT)}$ = 1.2V).
In case I_{OH} larger than this value is required, please design a unit in accordance with Fig. 3-(a), 3-(b) and 3-(c).

(Note 2) It is recommended that darlington type transistors ($V_{BE(SAT)}$ is less than 2V, h_{FE} is more than 500) be used as switching transistors (Tr_1 ~ Tr_4).

(Note 3) Please select proper input terminals depend on the type of mode to be used and input system and connect them in accordance with this catalog (Table-1, Fig.-6, Fig.-7) for use.

(Note 4) As for the excitation sequence of φ1 to φ4, please refer to Par. 5, excitation sequence, of this catalog.

This element has a built-in protective circuit at its input to prevent it from being damaged by high voltage and static electricity. However, it is very high in impedance, therefore, it should be used with great care exercized not to apply a voltage higher than its maximum rating. Especially in operation, it is desired to be kept within such a range as $V_{SS} < (V_{IN}, V_{OUT}) < V_{DD}$.
Input terminals not in use should be connected to V_{SS} or V_{DD}.

SANYO DENKI CO.,LTD.
No. 15-1, 1-chome, Kitaotsuka
Toshima-ku, Tokyo, JAPAN
TEL:TOKYO (03)917-5151-9
TLX:2722175 SANYOT J

Catalog No. 148-3, 79.9.3CB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

Hex Range	Usage
000-00F	DMA Chip 8237A-5
020-021	Interrupt 8259A
040-043	Timer 8253-5
060-063	PPI 8255A-5
080-083	DMA Page Registers
0AX*	NMI Mask Register
0CX	Reserved
0EX	Reserved
200-20F	Game Control
210-217	Expansion Unit
220-24F	Reserved
278-27F	Reserved
2F0-2F7	Reserved
2F8-2FF	Asynchronous Communications (Secondary)
300-31F	Prototype Card
320-32F	Fixed Disk
378-37F	Printer
380-38C**	SDLC Communications
380-389**	Binary Synchronous Communications (Secondary)
3A0-3A9	Binary Synchronous Communications (Primary)
380-38F	IBM Monochrome Display/Printer
3C0-3CF	Reserved
3D0-3DF	Color/Graphics
3E0-3E7	Reserved
3F0-3F7	Diskette
3F8-3FF	Asynchronous Communications (Primary)

* At power-on time, the Non Mask Interrupt into the 8088 is masked off. This mask bit can be set and reset through system software as follows:

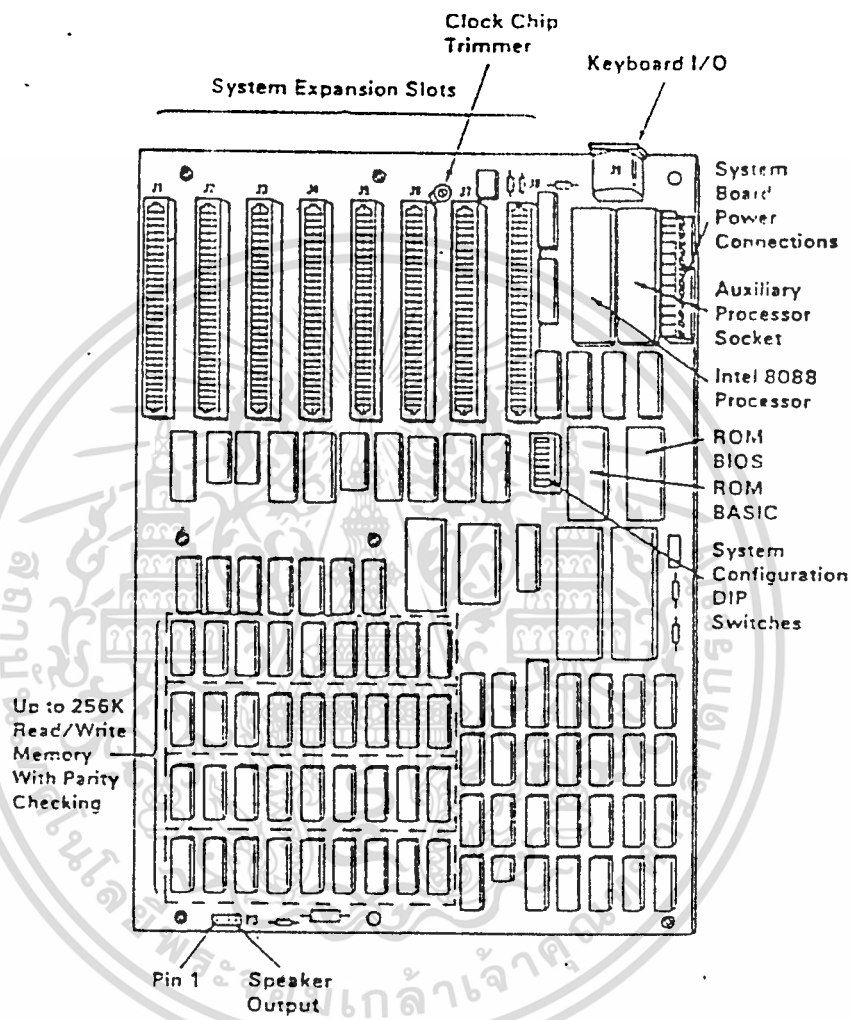
Set mask: Write hex 80 to I/O Address hex A0 (enable NMI)

Clear mask: Write hex 00 to I/O Address hex A0 (disable NMI)

** SDLC Communications and Secondary Binary Synchronous Communications cannot be used together because their hex addresses overlap.

I/O Address Map

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

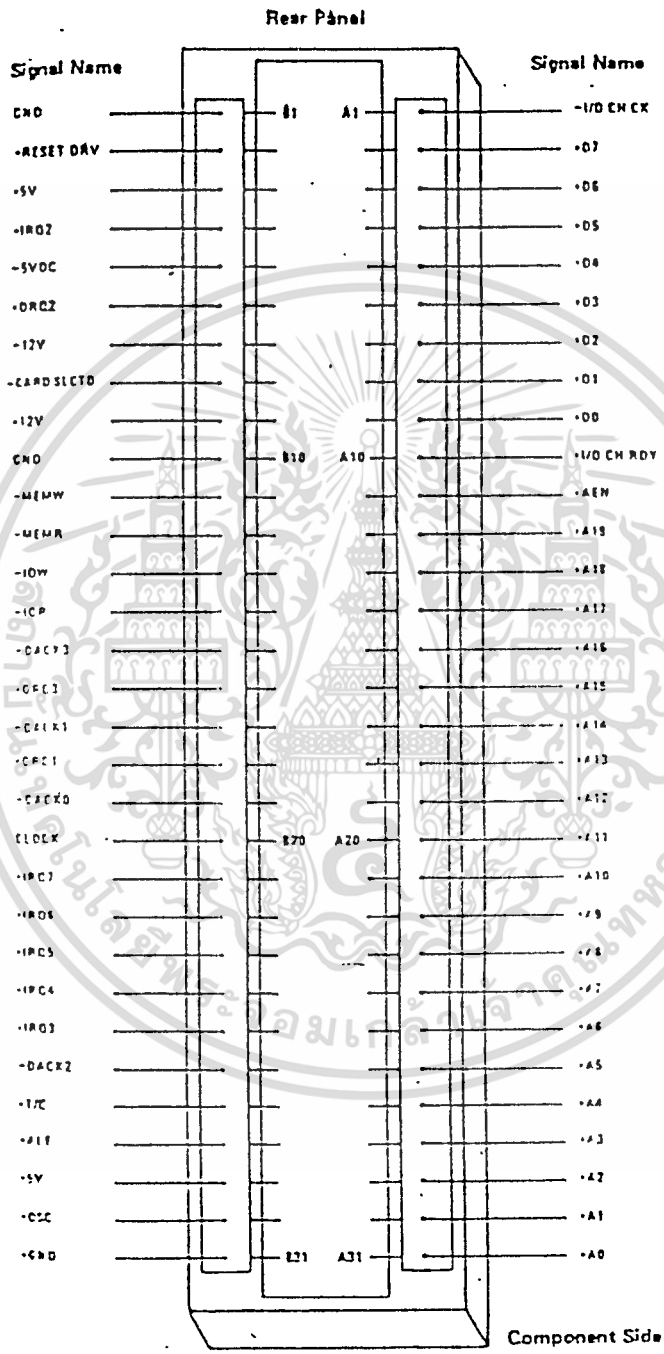


System Board Component Diagram

System Board Switch Settings

All system board switch settings for total system memory, number of diskette drives, and type of display are located in "Appendix G: Switch Settings."

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



I/O Channel Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

I/O Channel Description

The following is a description of the IBM Personal Computer XT I/O Channel. All lines are TTL-compatible.

Signal	I/O	Description
OSC	O	Oscillator: High-speed clock with a 70-ns period (14.31818 MHz). It has a 50% duty cycle.
CLK	O	System clock: It is a divide-by-three of the oscillator and has a period of 210 ns (4.77 MHz). The clock has a 33% duty cycle.
RESET DRV	O	This line is used to reset or initialize system logic upon power-up or during a low line voltage outage. This signal is synchronized to the falling edge of clock and is active high.
A0-A19	O	Address bits 0 to 19: These lines are used to address memory and I/O devices within the system. The 20 address lines allow access of up to 1 megabyte of memory. A0 is the least significant bit (LSB) and A19 is the most significant bit (MSB). These lines are generated by either the processor or DMA controller. They are active high.
D0-D7	I/O	Data Bits 0 to 7: These lines provide data bus bits 0 to 7 for the processor, memory, and I/O devices. D0 is the least significant bit (LSB) and D7 is the most significant bit (MSB). These lines are active high.
ALE	O	Address Latch Enable: This line is provided by the 8288 Bus Controller and is used on the system board to latch valid addresses from the processor. It is available to the I/O channel as an indicator of a valid processor address (when used with AEN). Processor addresses are latched with the failing edge of ALE.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Signal	I/O	Description
$\overline{\text{I/O CH CK}}$	I	-I/O Channel Check: This line provides the processor with parity (error) information on memory or devices in the I/O channel. When this signal is active low, a parity error is indicated.
I/O CH RDY	I	I/O Channel Ready: This line, normally high (ready), is pulled low (not ready) by a memory or I/O device to lengthen I/O or memory cycles. It allows slower devices to attach to the I/O channel with a minimum of difficulty. Any slow device using this line should drive it low immediately upon detecting a valid address and a read or write command. This line should never be held low longer than 10 clock cycles. Machine cycles (I/O or memory) are extended by an integral number of CLK cycles (210 ns).
IRQ2-IRQ7	I	Interrupt Request 2 to 7: These lines are used to signal the processor that an I/O device requires attention. They are prioritized with IRQ2 as the highest priority and IRQ7 as the lowest. An Interrupt Request is generated by raising an IRQ line (low to high) and holding it high until it is acknowledged by the processor (interrupt service routine).
$\overline{\text{IOR}}$	O	-I/O Read Command: This command line instructs an I/O device to drive its data onto the data bus. It may be driven by the processor or the DMA controller. This signal is active low.
$\overline{\text{IOW}}$	O	-I/O Write Command: This command line instructs an I/O device to read the data on the data bus. It may be driven by the processor or the DMA controller. This signal is active low.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Signal	I/O	Description
$\overline{\text{MEMR}}$	O	Memory Read Command: This command line instructs the memory to drive its data onto the data bus. It may be driven by the processor or the DMA controller. This signal is active low.
$\overline{\text{MEMW}}$	O	Memory Write Command: This command line instructs the memory to store the data present on the data bus. It may be driven by the processor or the DMA controller. This signal is active low.
DRQ1-DRQ3	I	DMA Request 1 to 3: These lines are asynchronous channel requests used by peripheral devices to gain DMA service. They are prioritized with DRQ3 being the lowest and DRQ1 being the highest. A request is generated by bringing a DRQ line to an active level (high). A DRQ line must be held high until the corresponding DACK line goes active.
$\overline{\text{DACK0-}}$ $\overline{\text{DACK3}}$	O	-DMA Acknowledge 0 to 3: These lines are used to acknowledge DMA requests (DRQ1-DRQ3) and to refresh system dynamic memory (DACK0). They are active low.
AEN	O	Address Enable: This line is used to de-gate the processor and other devices from the I/O channel to allow DMA transfers to take place. When this line is active (high), the DMA controller has control of the address bus, data bus, read command lines (memory and I/O), and the write command lines (memory and I/O).
T/C	O	Terminal Count: This line provides a pulse when the terminal count for any DMA channel is reached. This signal is active high.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Signal	I/O	Description
$\overline{\text{CARD SLCTD}}$	I	-Card Selected: This line is activated by cards in expansion slot J8. It signals the system board that the card has been selected and that appropriate drivers on the system board should be directed to either read from, or write to, expansion slot J8. Connectors J1 through J8 are tied together at this pin, but the system board does not use their signal. This line should be driven by an open collector device.

The following voltages are available on the system board I/O channel:

+5 Vdc ±5%, located on 2 connector pins
−5 Vdc ±10%, located on 1 connector pin
+12 Vdc ±5%, located on 1 connector pin
−12 Vdc ±10%, located on 1 connector pin
GND (Ground), located on 3 connector pins

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

intel

8253/8253-5
PROGRAMMABLE INTERVAL TIMER

- MCS-85™ Compatible 8253-5
- 3 Independent 16-Bit Counters
- DC to 2.6 MHz
- Programmable Counter Modes
- Count Binary or BCD
- Single +5V Supply
- Available in EXPRESS
  - —Standard Temperature Range
  - —Extended Temperature Range

The Intel® 8253 is a programmable counter/timer device designed for use as an Intel microcomputer peripheral. It uses nMOS technology with a single +5V supply and is packaged in a 24-pin plastic DIP.

It is organized as 3 independent 16-bit counters, each with a count rate of up to 2.6 MHz. All modes of operation are software programmable.

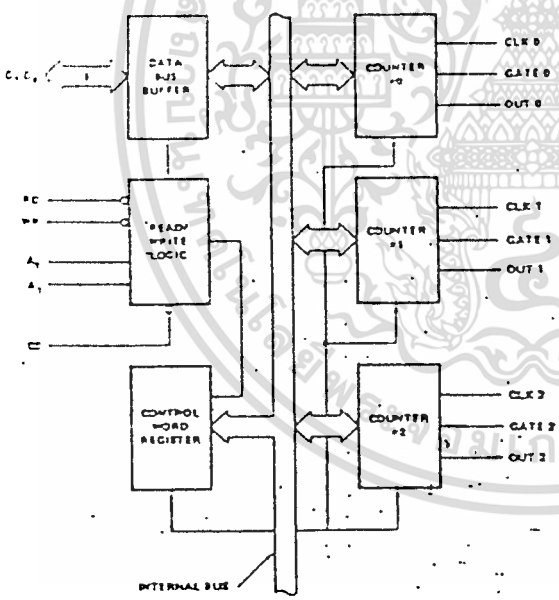


Figure 1. Block Diagram

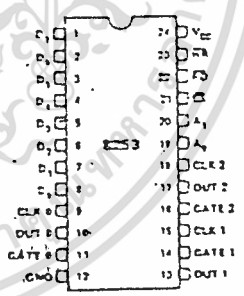


Figure 2. Pin Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FUNCTIONAL DESCRIPTION

General

The 8253 is a programmable interval timer/counter specifically designed for use with the Intel™ Microcomputer systems. Its function is that of a general purpose, multi-timing element that can be treated as an array of I/O ports in the system software.

The 8253 solves one of the most common problems in any microcomputer system, the generation of accurate time delays under software control. Instead of setting up timing loops in systems software, the programmer configures the 8253 to match his requirements, initializes one of the counters of the 8253 with the desired quantity, then upon command the 8253 will count out the delay and interrupt the CPU when it has completed its tasks. It is easy to see that the software overhead is minimal and that multiple delays can easily be maintained by assignment of priority levels.

Other counter/timer functions that are non-delay in nature but also common to most microcomputers can be implemented with the 8253.

- Programmable Rate Generator
- Event Counter
- Binary Rate Multiplier
- Real Time Clock
- Digital One-Shot
- Complex Motor Controller

Data Bus Buffer

This 3-state, bi-directional, 8-bit buffer is used to interface the 8253 to the system data bus. Data is transmitted or received by the buffer upon execution of INput or OUTput CPU instructions. The Data Bus Buffer has three basic functions.

1. Programming the MODES of the 8253.
2. Loading the count registers.
3. Reading the count values.

Read/Write Logic

The Read/Write Logic accepts inputs from the system bus and in turn generates control signals for overall device operation. It is enabled or disabled by CS so that no operation can occur to change the function unless the device has been selected by the system logic.

$\overline{RD}$ (Read)

A "low" on this input informs the 8253 that the CPU is inputting data in the form of a counters value.

$\overline{WR}$ (Write)

A "low" on this input informs the 8253 that the CPU is outputting data in the form of mode information or loading counters.

A0, A1

These inputs are normally connected to the address bus. Their function is to select one of the three counters to be operated on and to address the control word register for mode selection.

$\overline{CS}$ (Chip Select)

A "low" on this input enables the 8253. No reading or writing will occur unless the device is selected. The $\overline{CS}$ input has no effect upon the actual operation of the counters.

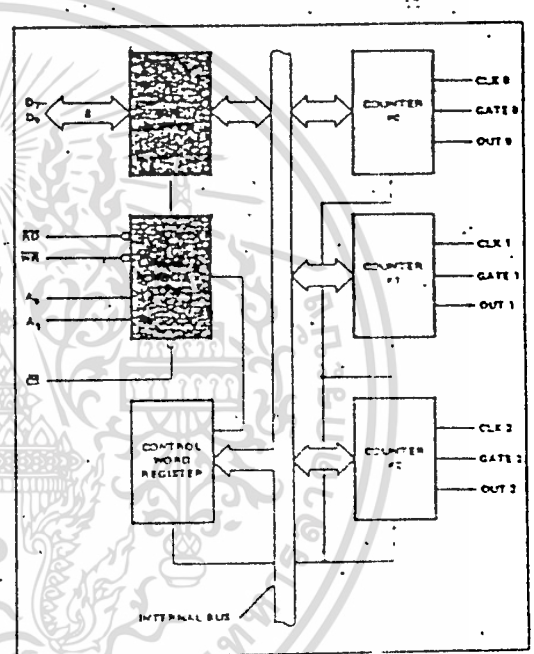


Figure 3. Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_1	A_0	
0	1	0	0	0	Load Counter No. 0
0	1	0	0	1	Load Counter No. 1
0	1	0	1	0	Load Counter No. 2
0	1	0	1	1	Write Mode Word
0	0	1	0	0	Read Counter No. 0
0	0	1	0	1	Read Counter No. 1
0	0	1	1	0	Read Counter No. 2
0	0	1	1	1	No-Operation 3-State
1	X	X	X	X	Disable 3-State
0	1	1	X	X	No-Operation 3-State

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Control Word Register

The Control Word Register is selected when A0, A1 are 11. It then accepts information from the data bus buffer and stores it in a register. The information stored in this register controls the operational MODE of each counter, selection of binary or BCD counting and the loading of each count register.

The Control Word Register can only be written into; no read operation of its contents is available.

Counter #0, Counter #1, Counter #2

These three functional blocks are identical in operation so only a single Counter will be described. Each Counter consists of a single, 16-bit, pre-settable, DOWN counter. The counter can operate in either binary or BCD and its input, gate and output are configured by the selection of MODES stored in the Control Word Register.

The counters are fully independent and each can have separate Mode configuration and counting operation, binary or BCD. Also, there are special features in the control word that handle the loading of the count value so that software overhead can be minimized for these functions.

The reading of the contents of each counter is available to the programmer with simple READ operations for event counting applications and special commands and logic are included in the 8253 so that the contents of each counter can be read "on the fly" without having to inhibit the clock input.

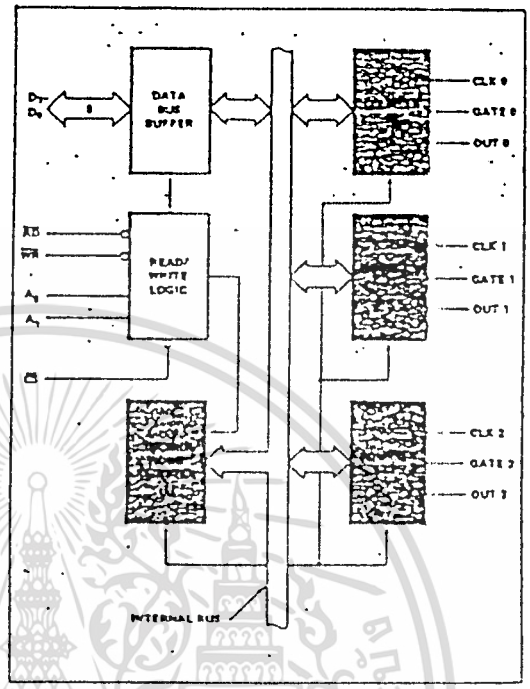


Figure 4. Block Diagram Showing Control Word Register and Counter Functions

8253 SYSTEM INTERFACE

The 8253 is a component of the Intel™ Microcomputer Systems and interfaces in the same manner as all other peripherals of the family. It is treated by the system's software as an array of peripheral I/O ports; three are counters and the fourth is a control register for MODE programming.

Basically, the select inputs A0, A1 connect to the A0, A1 address bus signals of the CPU. The $\overline{CS}$ can be derived directly from the address bus using a linear select method. Or it can be connected to the output of a decoder, such as an Intel® 8205 for larger systems.

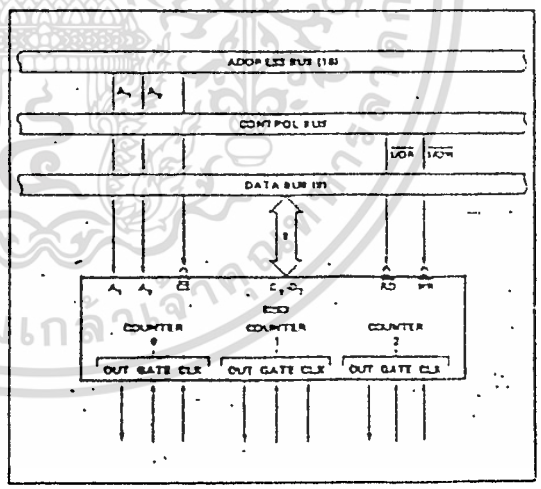


Figure 5. 8253 System Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OPERATIONAL DESCRIPTION

General

The complete functional definition of the 8253 is programmed by the systems software. A set of control words must be sent out by the CPU to initialize each counter of the 8253 with the desired MODE and quantity information. Prior to initialization, the MODE, count, and output of all counters is undefined. These control words program the MODE, Loading sequence and selection of binary or BCD counting.

Once programmed, the 8253 is ready to perform whatever timing tasks it is assigned to accomplish.

The actual counting operation of each counter is completely independent and additional logic is provided on-chip so that the usual problems associated with efficient monitoring and management of external, asynchronous events or rates to the microcomputer system have been eliminated.

Programming the 8253

All of the MODES for each counter are programmed by the systems software by simple I/O operations.

Each counter of the 8253 is individually programmed by [illegible] control word into the Control Word Register.

[illegible]

Control Word Format

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC	SC0	RL1	RL0	M2	M1	M0	BCD

Definition of Control

SC — Select Counter:

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Illegal

RL — Read/Load:

RL1	RL0	
0	0	Counter Latching operation (see READ/WRITE Procedure Section)
1	0	Read/Load most significant byte only.
0	1	Read/Load least significant byte only.
1	1	Read/Load least significant byte first, then most significant byte.

M — MODE:

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

BCD:

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

Counter Loading

The count register is not loaded until the count value is written (one or two bytes, depending on the mode selected by the RL bits), followed by a rising edge and a falling edge of the clock. Any read of the counter prior to that falling clock edge may yield invalid data.

MODE Definition

MODE 0: Interrupt on Terminal Count. The output will be initially low after the mode set operation. After the count is loaded into the selected count register, the output will remain low and the counter will count. When terminal count is reached the output will go high and remain high until the selected count register is reloaded with the mode or a new count is loaded. The counter continues to decrement after terminal count has been reached.

Rewriting a counter register during counting results in the following:

(1) Write 1st byte stops the current counting.
(2) Write 2nd byte starts the new count.

MODE 1: Programmable One-Shot. The output will go low on the count following the rising edge of the gate input.

The output will go high on the terminal count. If a new count value is loaded while the output is low it will not affect the duration of the one-shot pulse until the succeeding trigger. The current count can be read at any time without affecting the one-shot pulse.

The one-shot is retriggerable, hence the output will remain low for the full count after any rising edge of the gate input.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODE 2: Rate Generator. Divide by N counter. The output will be low for one period of the input clock. The period from one output pulse to the next equals the number of input counts in the count register. If the count register is reloaded between output pulses the present period will not be affected, but the subsequent period will reflect the new value.

The gate input, when low, will force the output high. When the gate input goes high, the counter will start from the initial count. Thus, the gate input can be used to synchronize the counter.

When this mode is set, the output will remain high until after the count register is loaded. The output then can also be synchronized by software.

MODE 3: Square Wave Rate Generator. Similar to MODE 2 except that the output will remain high until one half the count has been completed (for even numbers) and go low for the other half of the count. This is accomplished by decrementing the counter by two on the falling edge of each clock pulse. When the counter reaches terminal count, the state of the output is changed and the counter is reloaded with the full count and the whole process is repeated.

If the count is odd and the output is high, the first clock pulse (after the count is loaded) decrements the count by 1. Subsequent clock pulses decrement the clock by 2. After timeout, the output goes low and the full count is reloaded. The first clock pulse (following the reload) decrements the counter by 3. Subsequent clock pulses decrement the count by 2 until timeout. Then the whole process is repeated. In this way, if the count is odd, the output will be high for $(N+1)/2$ counts and low for $(N-1)/2$ counts.

In Modes 2 and 3, if a CLK source other than the system clock is used, GATE should be pulsed immediately following $\overline{WR}$ of a new count value.

MODE 4: Software Triggered Strobe. After the mode is set, the output will be high. When the count is loaded, the counter will begin counting. On terminal count, the output will go low for one input clock period, then will go high again.

If the count register is reloaded during counting, the new count will be loaded on the next CLK pulse. The count will be inhibited while the GATE input is low.

MODE 5: Hardware Triggered Strobe. The counter will start counting after the rising edge of the trigger input and will go low for one clock period when the terminal count is reached. The counter is retriggerable. The output will not go low until the full count after the rising edge of any trigger.

Modes \ Signal Status	Low Or Going Low	Rising	High
0	Disables counting	—	Enables counting
1	—	1) Initiates counting 2) Resets output after next clock	—
2	1) Disables counting 2) Sets output immediately high	1) Reloads counter 2) Initiates counting	Enables counting
3	1) Disables counting 2) Sets output immediately high	1) Reloads counter 2) Initiates counting	Enables counting
4	Disables counting	—	Enables counting
5	—	Initiates counting	—

Figure 6. Gate Pin Operations Summary

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

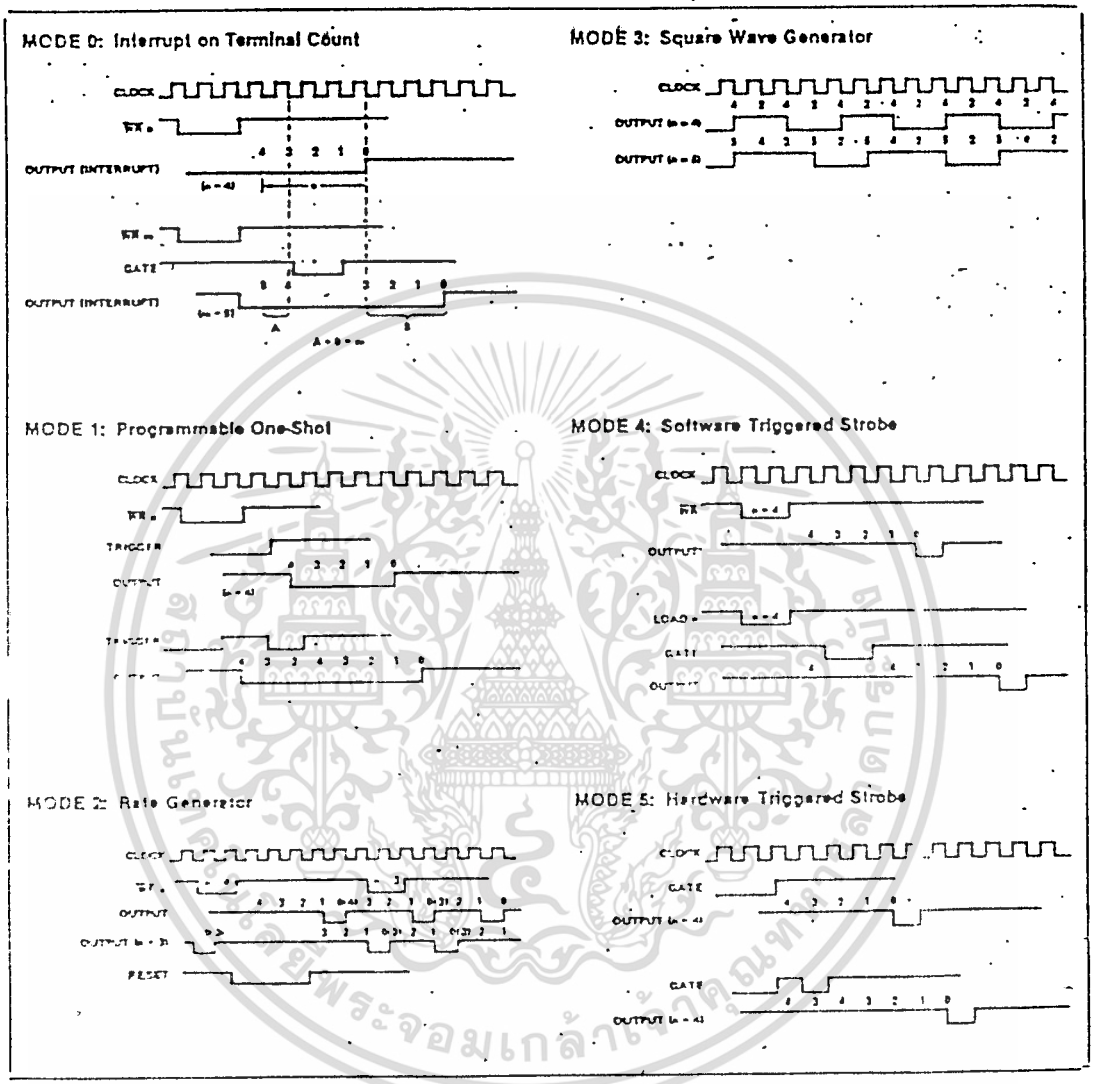


Figure 7. 8253 Timing Diagrams

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8253 READ/WRITE PROCEDURE

Write Operations

The systems software must program each counter of the 8253 with the mode and quantity desired. The programmer must write out to the 8253 a MODE control word and the programmed number of count register bytes (1 or 2) prior to actually using the selected counter.

The actual order of the programming is quite flexible. Writing out of the MODE control word can be in any sequence of counter selection, e.g., counter #0 does not have to be first or counter #2 last. Each counter's MODE control word register has a separate address so that its loading is completely sequence independent. (SC0, SC1)

The loading of the Count Register with the actual count value, however, must be done in exactly the sequence programmed in the MODE control word (RL0, RL1). This loading of the counter's count register is still sequence independent like the MODE control word loading, but when a selected count register is to be loaded it must be loaded with the number of bytes programmed in the MODE control word (RL0, RL1). The one or two bytes to be loaded in the count register do not have to follow the associated MODE control word. They can be programmed at any time following the MODE control word loading as long as the correct number of bytes is loaded in order.

All counters are down counters. Thus, the value loaded into the count register will actually be decremented. Loading all zeroes into a count register will result in the maximum count (2^{16} for Binary or 10^4 for BCD). In MODE 0 the new count will not restart until the load has been completed. It will accept one of two bytes depending on how the MODE control words (RL0, RL1) are programmed. Then proceed with the restart operation.

	MODE Control Word Counter n
LSB	Count Register Byte Counter n
MSB	Count Register Byte Counter n

Note: Format shown is a simple example of loading the 8253 and does not imply that it is the only format that can be used.

Figure 8. Programming Format

			A1	A0
No. 1		MODE Control Word Counter 0	1	1
No. 2		MODE Control Word Counter 1	1	1
No. 3		MODE Control Word Counter 2	1	1
No. 4	LSB	Count Register Byte Counter 1	0	1
No. 5	MSB	Count Register Byte Counter 1	0	1
No. 6	LSB	Count Register Byte Counter 2	1	0
No. 7	MSB	Count Register Byte Counter 2	1	0
No. 8	LSB	Count Register Byte Counter 0	0	0
No. 9	MSB	Count Register Byte Counter 0	0	0

Note: The exclusive addresses of each counter's count register make the task of programming the 8253 a very simple matter, and maximum effective use of the device will result if this feature is fully utilized.

Figure 9. Alternate Programming Formats

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Read Operations

In most counter applications it becomes necessary to read the value of the count in progress and make a computational decision based on this quantity. Event counters are probably the most common application that uses this function. The 8253 contains logic that will allow the programmer to easily read the contents of any of the three counters without disturbing the actual count in progress.

There are two methods that the programmer can use to read the value of the counters. The first method involves the use of simple I/O read operations of the selected counter. By controlling the A0, A1 inputs to the 8253 the programmer can select the counter to be read (remember that no read operation of the mode register is allowed A0, A1 11). The only requirement with this method is that in order to assure a stable count reading the actual operation of the selected counter must be inhibited either by controlling the Gate input or by external logic that inhibits the clock input. The contents of the counter selected will be available as follows:

first I/O Read contains the least significant byte (LSB).

second I/O Read contains the most significant byte (MSB).

Due to the internal logic of the 8253 it is absolutely necessary to complete the entire reading procedure. If two bytes are programmed to be read then two bytes must be read before any loading WR command can be sent to the same counter.

Read Operation Chart

A1	A0	RD	
0	0	0	Read Counter No. 0
0	1	0	Read Counter No. 1
1	0	0	Read Counter No. 2
1	1	0	Illegal

Reading While Counting

In order for the programmer to read the contents of any counter without effecting or disturbing the counting operation the 8253 has special internal logic that can be accessed using simple WR commands to the MODE register. Basically, when the programmer wishes to read the contents of a selected counter "on the fly" he loads the MODE register with a special code which latches the present count value into a storage register so that its contents contain an accurate stable quantity. The programmer then issues a normal read command to the selected counter and the contents of the latched register is available.

MODE Register for Latching Count

A0, A1 = 11

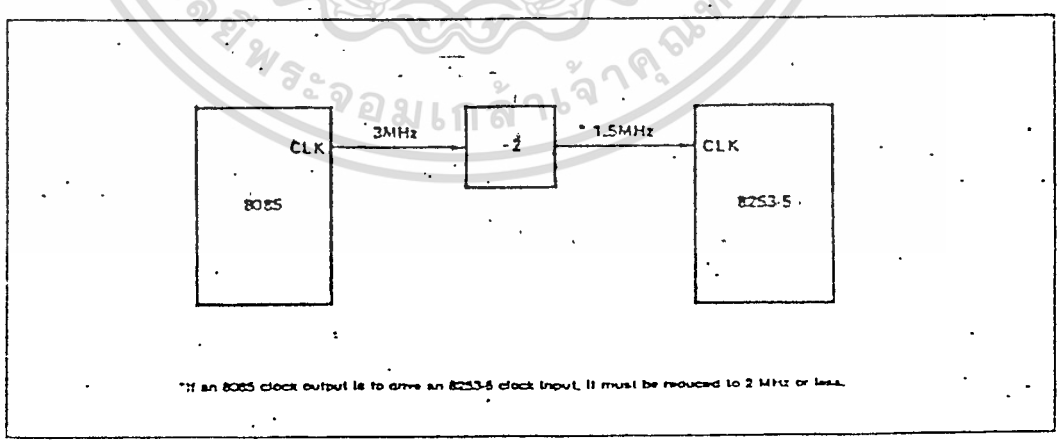
D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	0	0	X	X	X	X

SC1.SC0— specify counter to be latched.

D5.D4 — 00 designates counter latching operation.

X — don't care.

The same limitation applies to this mode of reading the counter as the previous method. That is, it is mandatory to complete the entire read operation as programmed. This command has no effect on the counter's mode.



*If an 8085 clock output is to drive an 8253-5 clock input, it must be reduced to 2 MHz or less.

Figure 10. MCS-85™ Clock Interface*

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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 ..... −0.5 V to +7 V
Power Dissipation ..... 1 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 the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

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

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.2	V_{CC}+.5V	V	
V_{OL}	Output Low Voltage		0.45	V	Note 1
V_{OH}	Output High Voltage	2.4		V	Note 2
I_{IL}	Input Load Current		±10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} to .45V
I_{CC}	V_{CC} Supply Current		140	mA	

CAPACITANCE (T_A = 25°C, V_{CC} = GND = 0V)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance			10	pF	fc = 1 MHz
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to V_{SS}

A.C. CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = 5.0V ± 10%, GND = 0V)

Bus Parameters (Note 3)

READ CYCLE

Symbol	Parameter	8253		8253-5		Unit
		Min.	Max.	Min.	Max.	
t_{AR}	Address Stable Before $\overline{READ}$	50		30		ns
t_{RA}	Address Hold Time for $\overline{READ}$	5		5		ns
t_{RR}	$\overline{READ}$ Pulse Width	400		300		ns
t_{RD}	Data Delay From $\overline{READ}$[4]		300		200	ns
t_{DF}	$\overline{READ}$ to Data Floating	25	125	25	100	ns
t_{RV}	Recovery Time Between READ and Any Other Control Signal	1		1		μs

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A.C. CHARACTERISTICS (Continued)

WRITE CYCLE

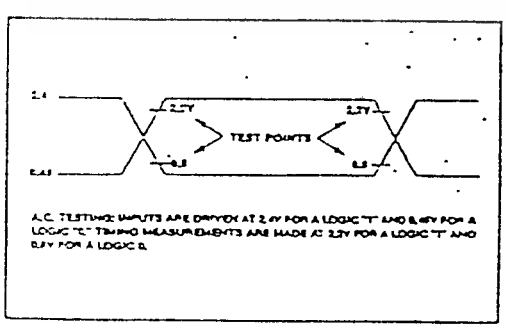
Symbol	Parameter	8253 Min.	8253 Max.	8253-5 Min.	8253-5 Max.	Unit
t_{AW}	Address Stable Before $\overline{WRITE}$	50		30		ns
t_{WA}	Address Hold Time for $\overline{WRITE}$	30		30		ns
t_{WW}	$\overline{WRITE}$ Pulse Width	400		300		ns
t_{DW}	Data Set Up Time for $\overline{WRITE}$	300		250		ns
t_{WD}	Data Hold Time for $\overline{WRITE}$	40		30		ns
t_{RV}	Recovery Time Between $\overline{WRITE}$ and Any Other Control Signal	1		1		μs

CLOCK AND GATE TIMING

Symbol	Parameter	8253 Min.	8253 Max.	8253-5 Min.	8253-5 Max.	Unit
t_{CLK}	Clock Period	380	dc	380	dc	ns
t_{PWH}	High Pulse Width	230		230		ns
t_{PWL}	Low Pulse Width	150		150		ns
t_{GW}	Gate Width High	150		150		ns
t_{GL}	Gate Width Low	100		100		ns
t_{GS}	Gate Set Up Time to CLK↑	100		100		ns
t_{GH}	Gate Hold Time After CLK↑	50		50		ns
t_{OD}	Output Delay From CLK↓[4]		400		400	ns
t_{ODG}	Output Delay From Gate↓[4]		300		300	ns

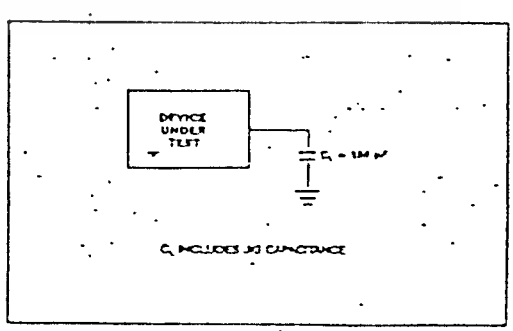
NOTES:
1. I_F = 2.2 mA
2. I_{OH} = −400 μA
3. A.C. timings measured at V_{OH} 2.2, V_{OL} = 0.8
4. C_L = 150pF.
* For Extended Temperature EXPRESS, use M8253 electrical parameters.

A.C. TESTING INPUT, OUTPUT WAVEFORM



A.C. TESTING INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0". TIMING MEASUREMENTS ARE MADE AT 2.2V FOR A LOGIC "1" AND 0.8V FOR A LOGIC 0.

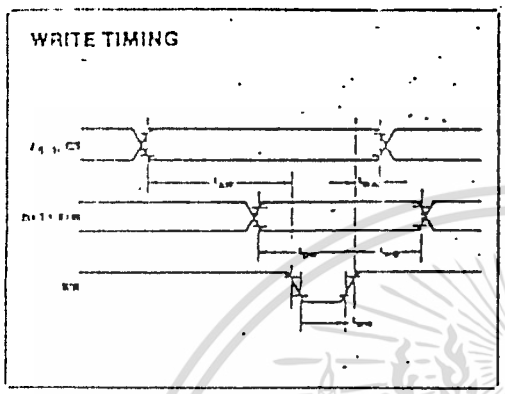
A.C. TESTING LOAD CIRCUIT

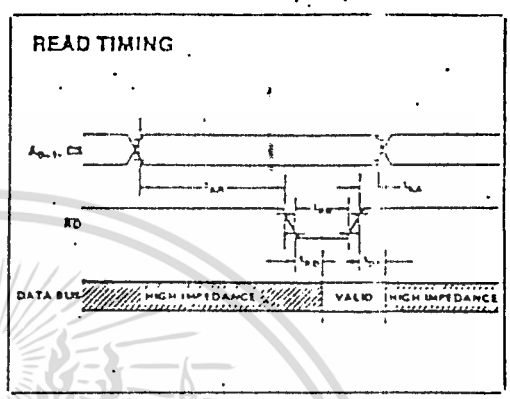


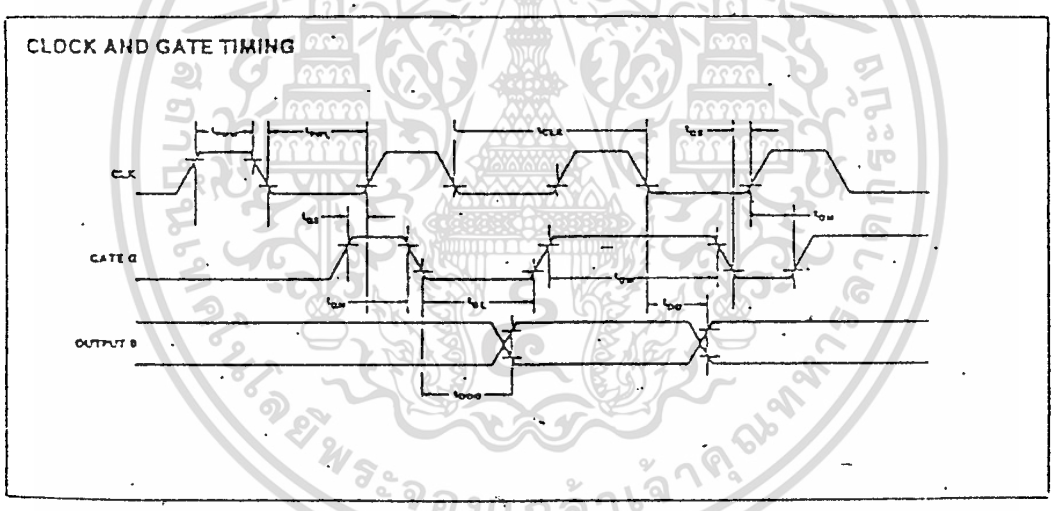
C_L INCLUDES JIG CAPACITANCE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAVEFORMS







เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้