



CNC : COMPUTER NUMERICAL CONTROL



โดย

นายชัยฤทธิ มลิวัลย์ 34162109

นายธีรโรจน์ แซ่โจ้ว 34162117

ปริญญานิพนธ์สำหรับปริญญาอุตสาหกรรมศาสตรบัณฑิต
สาขาคอมพิวเตอร์อุตสาหกรรม
ภาควิชาเทคโนโลยีการวัดคุมทางอุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2534

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032624

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

สาขา เทคโนโลยีคอมพิวเตอร์อุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบควบคุมเชิงเลขด้วยคอมพิวเตอร์

CNC : COMPUTER NUMERICAL CONTROL

ผู้จัดทำ

1. นายชัยฤทธิ์ มลิวัลย์ 34162109
2. นายธีรโรจน์ แซ่โจ้ว 34162117



.....อาจารย์ที่ปรึกษา

(อ.ภากร หุตะสังกาศ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032624

บทคัดย่อ

ปฏิญานี้ฉบับนี้ เป็นส่วนหนึ่งของเนื้อหาการออกแบบและสร้าง ระบบควบคุมเชิงเลขด้วย คอมพิวเตอร์ และเป็นการนำเสนอใน ส่วน ของการควบคุมระบบเครื่องจักรกล ที่ใช้มอเตอร์แบบ สเตปป์ (STEPPING MOTOR) และเป็นการนำไปประยุกต์ใช้งานเกี่ยวกับระบบควบคุมอัตโนมัติ โดยใช้คอมพิวเตอร์เป็นตัวควบคุม (CNC : COMPUTER NUMERICAL CONTROL)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSTRACT

THIS THESIS IS PART OF DESCRIPTION OF THE DESIGN AND THE CONSTRUCTION IN NUMERICAL CONTROL SYSTEM BY USING COMPUTER AND PROVIDES TO DETIAL OF CONTROLLING - MACHINE SYSTEM WHICH USES STEPPING - MOTOR WHICH CAN BE INCLUDED IN AUTOMATIC - CONTROLLED APPLICATIONS USING COMPUTER CONTROL OR CNC : (COMPUTER NUMERICAL CONTROL).



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ

Abstract

บทนำ.....

บทที่ 1. สเต็ปป์มอเตอร์

โครงสร้างและการทำงาน..... 1-1

บทที่ 2. การ์ดอินพุทและเอาต์พุท (Input/Output Card)

กล่าวหน้า..... 2-1

การอินเตอร์เฟส

2.1 ขาสัญญาณที่ใช้ในการอินเตอร์เฟส..... 2-1

2.2 บัสไซเคิลของการอ่านข้อมูลจากพอร์ต อินพุทและเอาต์พุท..... 2-4

2.3 บัสไซเคิลของการเขียนข้อมูลลงบนพอร์ต อินพุทและเอาต์พุท..... 2-4

2.4 ชิพชิพพอร์ต #8255 PIA..... 2-5

2.5 วงจรของ การ์ดอินพุทและเอาต์พุท..... 2-9

บทที่ 3. ไดรฟ์เวอร์

DRIVER UNIT..... 3-1

ส่วนประกอบที่สำคัญของ Driver Unit..... 3-2

3.1 ไอซี #L297..... 3-6

3.2 ส่วนของการ Drive Stepper Motor..... 3-7

3.3 วงจรของ DRIVER UNIT..... 3-10

การทำงานของวงจรโดยรวม..... 3-11

โปรแกรมที่ใช้ในการทดลอง..... 4-1

สรุป..... 4-13

หนังสืออ้างอิง..... 4-14

บทนำ

การนำเสนอ ผลงานทางด้าน ฮาร์ดแวร์ (Hard Ware) ในส่วนของชุดควบคุมการทำงานของมอเตอร์แบบ สเต็ปป์ (Stepping Motor) นั้น มีวัตถุประสงค์เพื่อนำเอาฮาร์ดแวร์ไปประยุกต์ใช้ใน ระบบควบคุมเชิงเลขด้วยคอมพิวเตอร์ และการนำเสนอในคราวนี้ได้ถูกแบ่งออกเป็นสองส่วนด้วยกันกล่าวคือ

ในส่วนที่หนึ่งนั้น เป็น การดัดอินพุทและเอาต์พุทที่จะนำไปใช้งานอยู่ภายใน ไมโครคอมพิวเตอร์ (Microcomputer) ประกอบด้วยไอซีชิพซัพพอร์ต (Shipsupport) เบอร์ #8255 เป็นสำคัญ และเป็น การดัดอินพุทและเอาต์พุทแบบแอนะล็อกที่สามารถโปรแกรมให้รับหรือส่งข้อมูลระหว่าง CPU กับอุปกรณ์ภายนอก

ส่วนที่สองก็จะเป็นส่วนของชุด ไดรเวอร์ (Driver) เป็นส่วนที่เชื่อมต่ออยู่ภายนอก ไมโครคอมพิวเตอร์ (Microcomputer) ในส่วนนี้จะประกอบด้วย ไอซีเบอร์ #L297 ซึ่งทำหน้าที่เป็นตัวผลิตเฟสให้ควบคุมมอเตอร์ แบบสเต็ปป์ และมีวงจรมอดูเลเตอร์ (PWM) เป็นซอฟต์แวร์อยู่ภายในชิพ ในส่วนของชุด ไดรเวอร์ (Driver) นี้ จะถูกสั่งงานผ่านการดัดอินพุทและเอาต์พุท ไปควบคุมสเต็ปป์ ให้มีการทำงานแบบ ครึ่งสเต็ปหรือเต็มสเต็ป (Half/Full Step) , หมุนทวนเข็มหรือหมุนตามเข็มนาฬิกา (Clockwise/AntiClockwish Direction) รวมถึงการรีเซ็ต (Reset) และอีน่าเบิล (Enable) #L297 เฟสเอาต์พุทที่ได้จาก #L297 จะถูกส่งผ่านไปยังไดรฟ์เพาเวอร์ดาร์ลิ่งตัน (Darlington) และไปขับมอเตอร์ให้เกิดการหมุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

สแต็ปปีงมอเตอร์ (Motor Stepping)

การใช้ไมโครคอมพิวเตอร์ (Microcomputer) ในการควบคุมเชิงตัวเลข กับสแต็ปปีงมอเตอร์ (Motor Stepping) มีอยู่ด้วยกันหลายๆ งาน เช่นในเครื่องพิมพ์ ใน X-Y พล็อตเตอร์ (Plotter) ในแขนกล ในเครื่องถ่ายภาพเอกสาร ในเครื่องควบคุมการเจาะ เพราะสแต็ปปีงมอเตอร์ (Motor Stepping) มีข้อได้เปรียบมอเตอร์ (Motor) ไฟตรงแบบธรรมดา คือ สแต็ปปีงมอเตอร์ (Motor Stepping) มีการควบคุมแบบหลูปเปิด จึงทำให้ง่ายต่อการควบคุม และยังสามารถที่จะรู้ถึงตำแหน่งของมอเตอร์ (Motor) ได้อย่างตลอดเวลาอย่างแน่นอนและแม่นยำ ดังนั้นงานที่ต้องการควบคุมตำแหน่งที่แน่นอนและแม่นยำจึงมักใช้สแต็ปปีงมอเตอร์ (Motor Stepping)

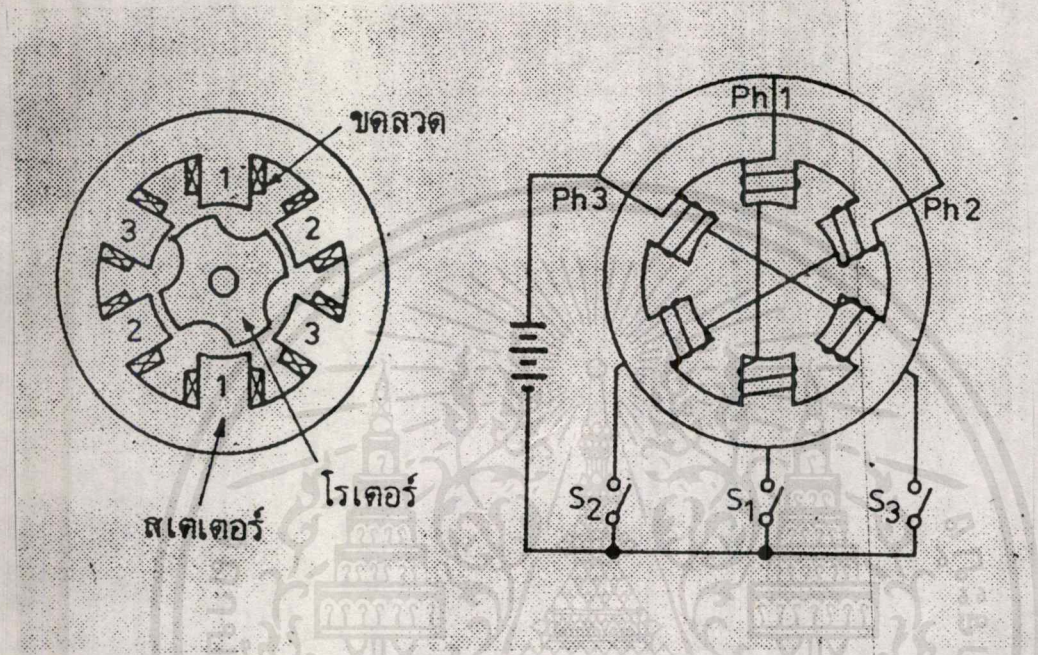
การควบคุมสแต็ปปีงมอเตอร์ (Motor Stepping) ส่วนใหญ่ใช้สองวิธีด้วยกันคือ วิธีแรกใช้ไมโครคอมพิวเตอร์ (Microcomputer) หรือ ซิงเกิ้ลบอร์ด (Single Board) ในการควบคุม ส่วนอีกแบบหนึ่ง ใช้วงจรถิจตอล (Digital) ในการควบคุม ซึ่งในวิธีแรกนั้นจะต้องทำชุดอินเตอร์เฟส (Interface) ระหว่างไมโครคอมพิวเตอร์ (Microcomputer) หรือ ซิงเกิ้ลบอร์ด (Single Board) กับสแต็ปปีงมอเตอร์ (Motor Stepping) และจะต้องมีซอฟต์แวร์ เพื่อให้ควบคุมสแต็ปปีงมอเตอร์ (Motor Stepping) ให้ทำงาน ซึ่งเป็นวิธีที่ต้องอาศัยความรู้และเทคนิคในการเขียนโปรแกรมเป็นอย่างมาก

โครงสร้างและการทำงานของสแต็ปปีงมอเตอร์ (Motor Stepping)

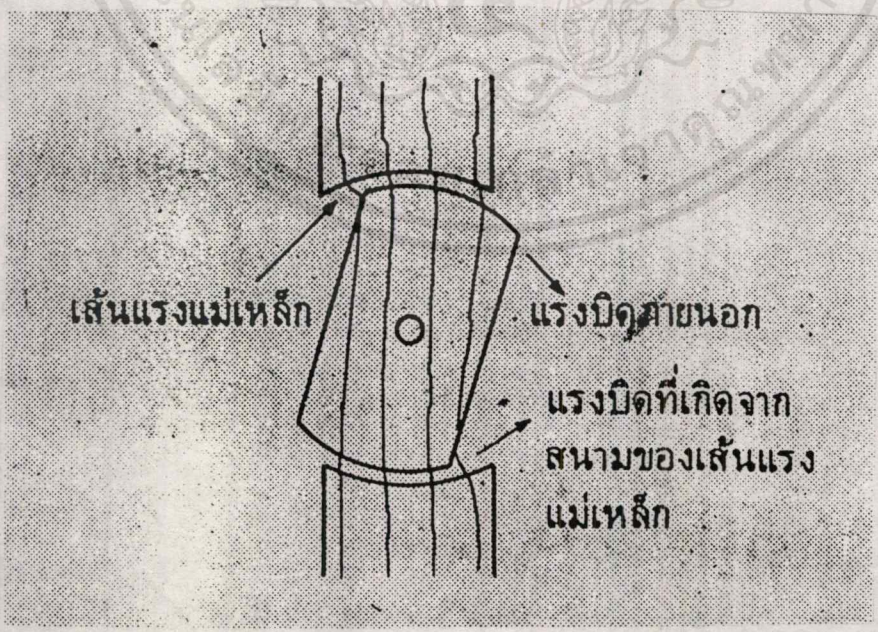
ภายในสแต็ปปีงมอเตอร์ (Motor Stepping) จะประกอบด้วย สเตเตอร์, โรเตอร์, และขดลวดประกอบเข้าด้วยกันดังรูปที่ 1.1 (สมมุติเป็นมอเตอร์แบบ 3 เฟส) และเนื่องจากสแต็ปปีงมอเตอร์ (Motor Stepping) แบบนี้มีโรเตอร์เป็นเหล็กอ่อน ซึ่งมีคุณสมบัติพยายามปรับตัวเองให้อยู่ในแนวที่เส้นแรงแม่เหล็กผ่านมากที่สุดดังรูปที่ 1.2 เมื่อเกิดเส้นแรงแม่เหล็กขึ้นที่สเตเตอร์ตัดผ่านโรเตอร์ ตัวโรเตอร์ก็จะพยายามปรับตัวเองให้เส้นแรงแม่เหล็กตัดผ่านตัวเองมากที่สุด โดยการหมุนตัวเอง จะทำให้เกิดมุมของการหมุนขึ้น และจะหยุดหมุนเมื่อเส้นแรงแม่เหล็กที่ตัดผ่านตัวมันถึงจุดที่มากที่สุด การทำให้สแต็ปปีงมอเตอร์ (Motor Stepping) หมุนก็ทำได้โดยอาศัยหลักการนี้ แต่ต้องให้เส้นแรงแม่เหล็กที่เกิดขึ้น โดยรับช่วงต่อกันไปเรื่อยๆ ดังรูปที่ 1.3ก, ข, และ ค. ซึ่งแสดงถึงการหมุนมอเตอร์ โดยทิศทางนั้นจะขึ้นอยู่กับกรับกระแสเข้าขดลวดว่าจะให้ไปทางไหน และเมื่อต้องเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การให้มอเตอร์หยุด ก็ให้หยุดการขับเคลื่อนแกโรเตอร์ (Rotor) มอเตอร์ก็จะหยุด ณ ตำแหน่ง
 สุดท้ายที่มีการขับที่สเตเตอร์ (Stator) ดังนั้นจึงสามารถรู้ถึงตำแหน่งของมอเตอร์ได้ โดยการนับจํ
 นวนพัลส์ (Pluse) ที่ป้อนให้มอเตอร์โดยใช้สูตร

$$\text{มุมที่เปลี่ยนไป} = \text{ค่ามุมแต่ละสเตป} \times \text{จำนวนพัลส์ที่ป้อน}$$

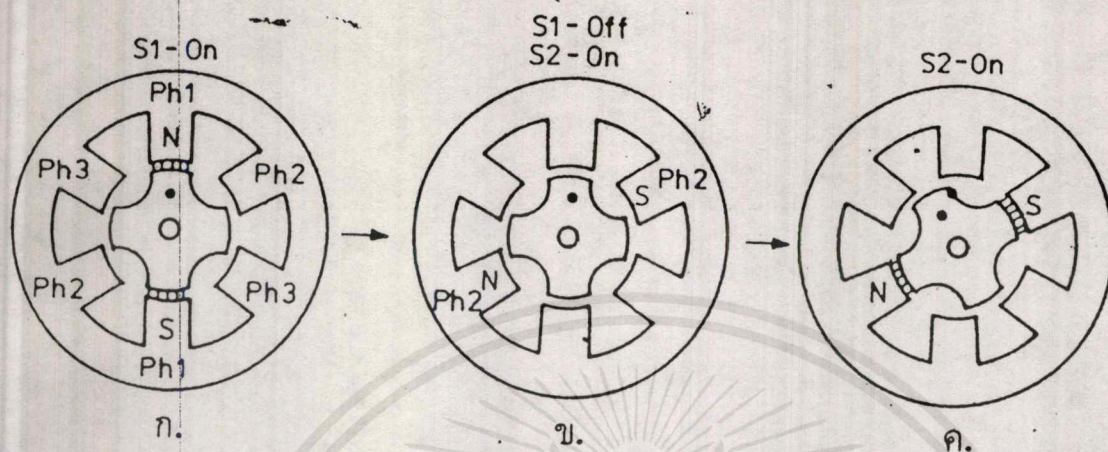


รูปที่ 1.1 ภาพหน้าตัดของสเต็ปมิ่งมอเตอร์

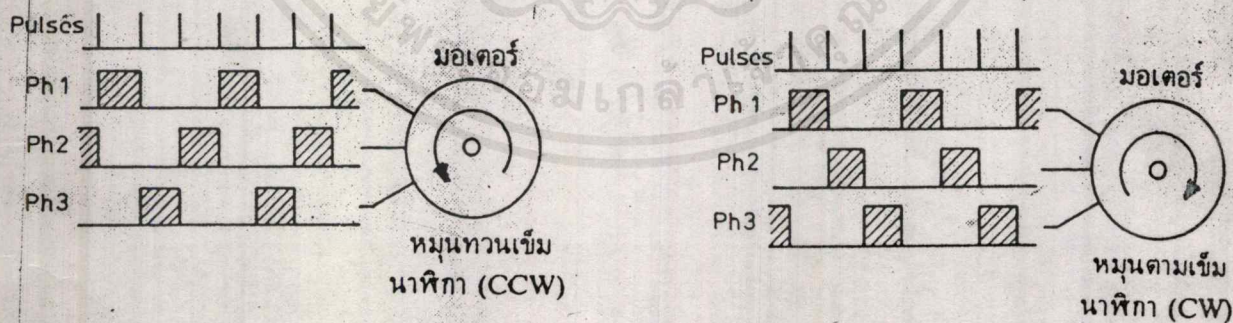


รูปที่ 1.2 เส้นแรงแม่เหล็กที่ทำให้เกิดแรงบิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

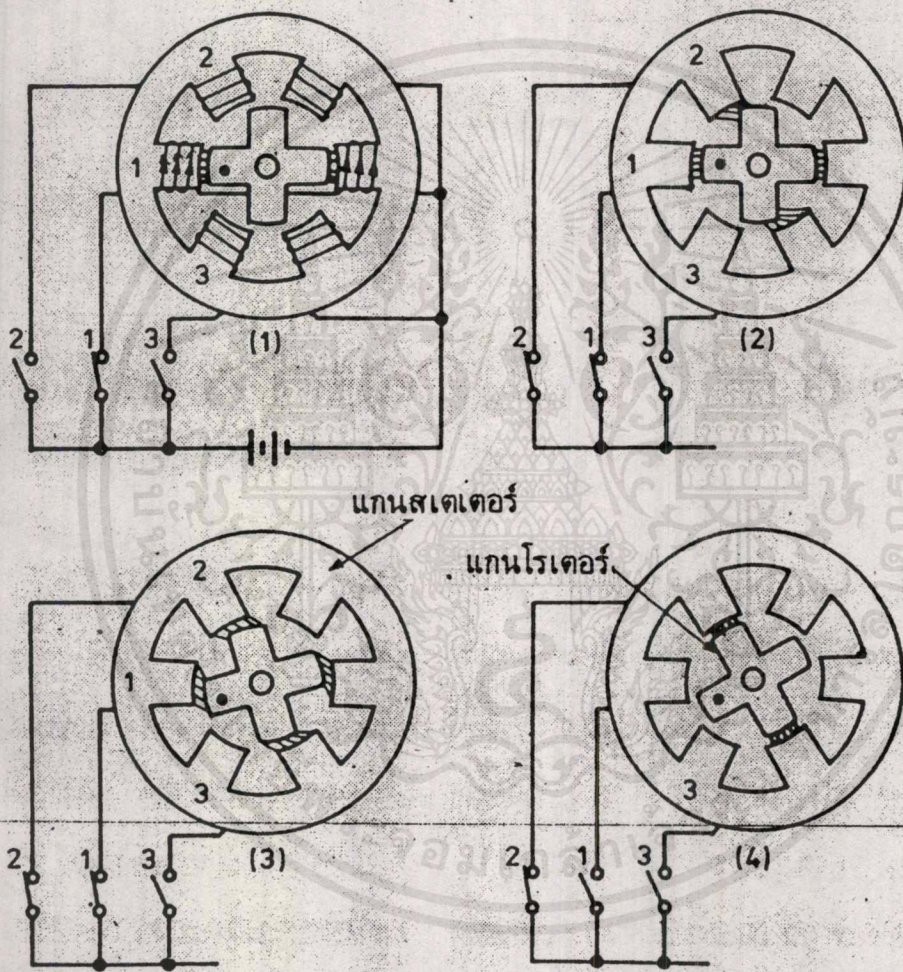


รูปที่ 1.3 แสดงการเคลื่อนที่ทีละสเตป เมื่อกระตุ้นเฟส 1 เฟส 2



รูปที่ 4 แสดงการกระตุ้น แบบเดินหน้าและถอยหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.5 แสดงการทำงานของสเต็ปป์มอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การกระตุ้นเฟสขดลวดสเตเตอร์

จากที่กล่าวมาการที่จะทำให้สเตปมอเตอร์ (Stepping Motor) หมุนได้นั้น จะต้องทำการกระตุ้นเฟสของขดลวดสเตเตอร์ให้เรียงกันไปเรื่อยๆทางใดทางหนึ่งถ้าต้องการให้หมุนกลับ การจะต้องกระตุ้นเฟสในทิศทางที่กลับกัน ซึ่งการกระตุ้นเฟสของสเตออร์มีอยู่ด้วยกัน 3 แบบ คือ

1>. การกระตุ้นเฟสเดี่ยว Single Phase Excitation

สัญญาณนาฬิกา	R	1	2	3	4	5	6	7	8
เฟสที่ 1	X			X			X		
เฟสที่ 2		X			X			X	
เฟสที่ 3			X			X			X

รูปที่ 1.6 แสดงการกระตุ้นแบบเฟสเดี่ยว

2>. การกระตุ้นแบบสองเฟส เรียกกันว่า Two Phase Excitation

สัญญาณนาฬิกา	R	1	2	3	4	5	6	7	8
เฟสที่ 1	X	X		X	X		X	X	
เฟสที่ 2		X	X		X	X		X	X
เฟสที่ 3			X	X		X	X		X

รูปที่ 1.7 แสดงการกระตุ้นแบบสองเฟส

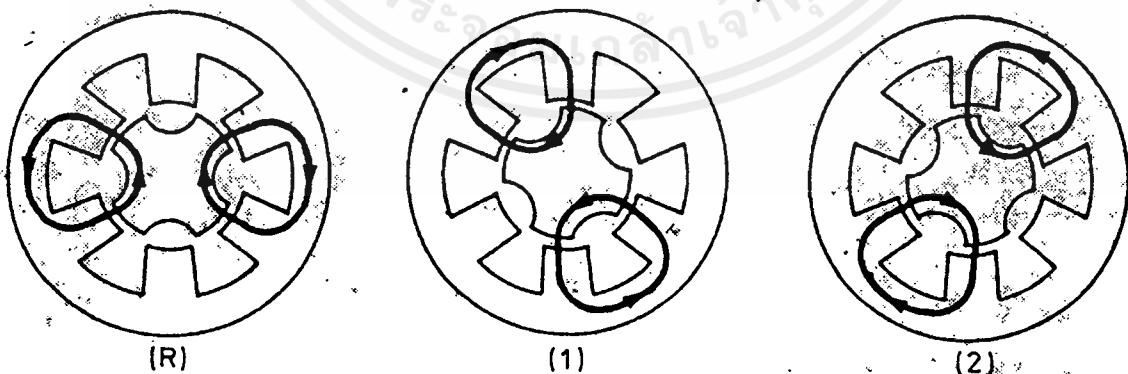
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3>. การกระตุ้นโดยใช้แบบที่หนึ่ง และแบบที่สองสลับกันไป เรียกว่าแบบ One - Two Phase Excitation หรือแบบ Half Step Operation

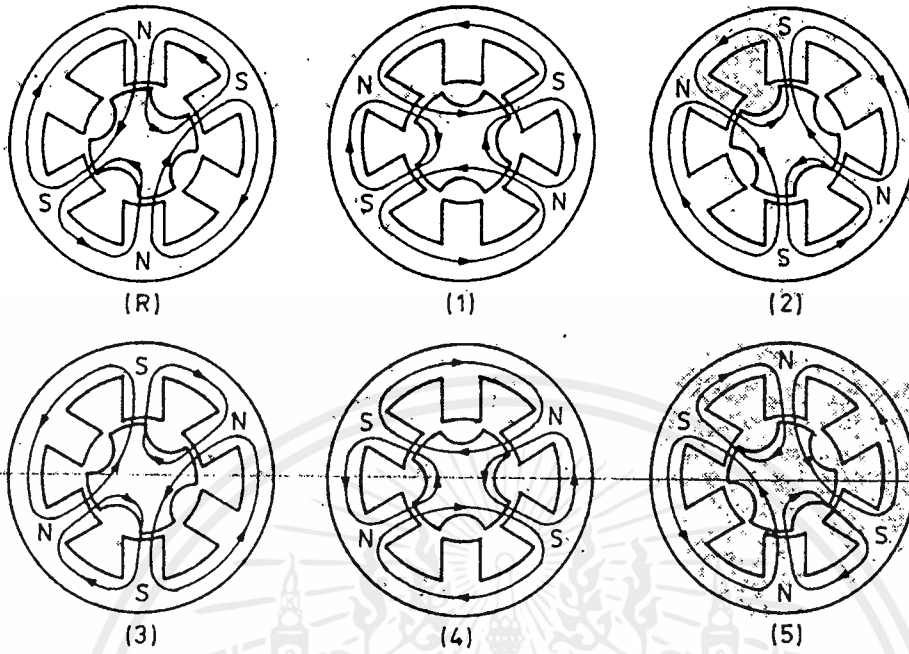
สัญญาณนาฬิกา	R	1	2	3	4	5	6	7	8
เฟสที่ 1	X	X					X	X	X
เฟสที่ 2		X	X	X					X
เฟสที่ 3				X	X	X			

รูปที่ 1.8 แสดงการกระตุ้นแบบครึ่งสเต็ป

การขับแบบกระตุ้นสองเฟส เส้นแรงแม่เหล็ก จะไม่ผ่านแกนเหล็กเป็นเส้นตรงเลยที่เดียวเหมือนแบบกระตุ้นเฟสเดียวแต่จะวกกลับเข้ามาสู่แกนทางด้านข้าง ดังรูปที่ 1.9 และเส้นแรงแม่เหล็กส่วนหนึ่งมาจากแกนตรงข้าม ดังรูปที่ 1.10



เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ให้บุคคลหรือหน่วยงานใด ๆ ที่ได้รับอนุญาตให้ใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.10 ลักษณะเส้นแรงแม่เหล็กเมื่อขับแบบสองเฟส

การขับสแต็ปมอเตอร์ แบบกระตุ้น 2 เฟส จะมีลักษณะเดียวกันกับการขับแบบกระตุ้นเฟสเดียวแต่ในการกระตุ้นแต่ละครั้งนั้น จะกระตุ้นที่เดียวพร้อมกันทั้ง 2 เฟส จะต่างก็ตรงที่ การขับแบบ 2 เฟส เข้าตำแหน่งแต่ละสแต็ปได้เร็วกว่าแบบเฟสเดียว และแรงบิดมากกว่าการขับแบบเฟสเดียวด้วย

การเกิดฮิสเทรีซิสของสแต็ปมอเตอร์

ความเร็วของการหมุนของสแต็ปมอเตอร์เป็นแบบลิเนียร์ กับความถี่ที่ป้อนให้สแต็ปมอเตอร์ แต่เมื่อป้อนความถี่ขึ้นเรื่อยๆ จนถึงความถี่ช่วงหนึ่งสแต็ปมอเตอร์ก็จะหยุดหมุน เนื่องจากการที่โรเตอร์หมุนตามพัลส์แม่เหล็กไม่ทัน เรียกว่ามอเตอร์เกิดการฮิสเทรีซิส ซึ่งมอเตอร์แต่ละตัวจะเกิดการฮิสเทรีซิสที่ความถี่ต่างๆ กันไป โดยทั่วไปแล้วนั้นสแต็ปมอเตอร์จะฮิสเทรีซิสที่ความถี่ประมาณ 500 Hz ในการขับแบบกระตุ้น 1 หรือ 2 เฟส และจะมีความถี่ประมาณ 1 KHz แต่เมื่อลดความถี่ให้ต่ำลงสแต็ปมอเตอร์จะไม่หมุนทันที และเมื่อความถี่ลดลงจนถึงความถี่หนึ่งสแต็ปมอเตอร์ จึงจะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มหมุนอีกครึ่งหนึ่ง นั่นก็คือสแต๊ปมอเตอร์มี ฮีสเทอรีซิส ซึ่งสแต๊ปมอเตอร์จะเริ่มหมุนที่ความถี่ประมาณ 200 Hz ในการขับแบบ 1 หรือ 2 เฟส และมีความถี่ประมาณ 150 Hz เมื่อขับแบบครึ่งสแต๊ป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



การดีอินพุทและเอาต์พุท

INPUT/OUTPUT CARD

การใช้เครื่อง ไมโครคอมพิวเตอร์ (Microcomputer) ในการควบคุมการเคลื่อนที่หรือในการเคลื่อนย้ายวัตถุ อุปกรณ์ที่สามารถควบคุมตำแหน่งและการหมุนแกน (Shaft) อย่างละเอียดและแม่นยำ รวมทั้งการควบคุมทิศทาง การหมุนและความเร็วคือ สเตปปีงมอเตอร์ (Stepping Motor) ก็จำเป็นต้องทำการอินเตอร์เฟส ซึ่งการอินเตอร์เฟสจะออกแบบและจัดให้ใช้ควบคุมและตรวจสอบ (Sense) ข้อมูล จะเป็นการอินเตอร์เฟส ที่ใช้สัญญาณดิจิทัลเป็นหลัก สัญญาณพวกนี้จะถูกส่งและรับมาจากรีจิสเตอร์ อินพุท/เอาต์พุท, เป็นการที่ใช้สัญญาณดิจิทัลที่มีระดับสัญญาณแบบ TTL ในการอินเตอร์เฟส ดังนั้นถ้าเรานำอุปกรณ์ที่ใช้ระดับสัญญาณแบบ TTL ในการอินเตอร์เฟสด้วยกันมาทำการอินเตอร์เฟสเข้ากับเครื่อง ไมโครคอมพิวเตอร์ (Microcomputer) จะทำให้ไมโครคอมพิวเตอร์ (Microcomputer) สามารถติดต่อกับอุปกรณ์บริการภายนอกได้

การอินเตอร์เฟสโดยการสร้าง การดีอินพุทและเอาต์พุท (Input/Output Card) และชุดของไดร์ฟเวอร์ (Driver) นั้นจำเป็นที่จะต้องมียุทธศาสตร์ต่างๆ เพื่อทำการติดต่อกับอุปกรณ์ภายนอก

2.1 ขาสัญญาณที่ใช้ในการอินเตอร์เฟส บนไมโครคอมพิวเตอร์ (Microcomputer)

2.1.1 A0-A15 (Address Bus)

เป็นขาของสัญญาณ เอาต์พุท ซึ่งใช้สำหรับกำหนดแอดเดรสของอุปกรณ์อินพุทและเอาต์พุท ที่ต้องการติดต่อกับ โดยที่สัญญาณ A0 จะมีนัยสำคัญต่ำที่สุด (Least Significant Bit) และ A15 จะมีนัยสำคัญสูงสุด (Most Significant Bit) สำหรับการอ้างแอดเดรสของ พอร์ต อินพุทและเอาต์พุท นั้น จะใช้แอดเดรสเพียง 16 เส้นคือ A0-A15 ซึ่งจะทำให้อ้างแอดเดรสของพอร์ตได้ 64K พอร์ต โดยผ่านทางชุดคำสั่ง IN และ OUT (ในภาษา Assambley) อย่างไรก็ตามภายใน IBM/PC จะใช้แอดเดรสในการอ้างแอดเดรสของพอร์ตเพียง 10 เส้น คือจาก A0-A9 และค่าแอดเดรสที่ใช้งานจะอยู่ในช่วง 0200H จนถึง 03FFH เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032624

2.1.2 DO-D7 (Data Bus)

ขาสัญญาณนี้เป็นขาสัญญาณแบบ Bi-Directional ซึ่งต่อกับบัสข้อมูลของระบบ เพื่อทำหน้าที่ในการส่งผ่านข้อมูล ระหว่างพอร์ตอินพุตและเอาต์พุต กับ IBM/PC โดยบิต DO จะมีนัยสำคัญต่ำที่สุด และบิต D7 จะมีนัยสำคัญสูงที่สุด

สำหรับในบัสไซเคิลของการเขียนข้อมูลที่สร้างขึ้นโดย 8088 นั้น ข้อมูลจะถูกส่งมายังบัสข้อมูล ก่อนที่สัญญาณ IOW (ในกรณีที่ต้องการส่งข้อมูลให้กับพอร์ต) ซึ่งโดยทั่วไปขอบเขตของสัญญาณ IOW จะถูกใช้พอร์ต อินพุตและเอาต์พุต ที่มีแอดเดรสตรงกับค่าแอดเดรสนั้น เพื่อรับข้อมูลเก็บไว้

สำหรับในบัสไซเคิลของการอ่านข้อมูลที่สร้างขึ้นโดย 8088 นั้น พอร์ตอินพุตและเอาต์พุต ที่ถูกอ้างถึงจะต้องส่งข้อมูลออกมาบนบัสข้อมูล ก่อนที่สัญญาณ IOR (ในกรณีที่ต้องการอ่านข้อมูลจากพอร์ต)

2.1.3 IOR (อินพุตและเอาต์พุต พอร์ต Read)

ขาสัญญาณนี้เป็นเอาต์พุตแอดดิฟที่ลอจิก "0" ที่สร้างขึ้นโดย 8288 Bus Controller เพื่อใช้ในการแสดงว่าบัสไซเคิลที่เกิดขึ้นนี้เป็นบัสไซเคิลของการอ่านข้อมูลจาก พอร์ตอินพุตและเอาต์พุต เพื่อให้ พอร์ต อินพุตและเอาต์พุต ที่มีแอดเดรสตรงกับแอดเดรสบนบัสแอดเดรสนั้นส่งข้อมูลออกมาบนบัสข้อมูล โดยข้อมูลจะต้องถูกส่งออกมาบนบัสข้อมูลก่อนขอบเขตของสัญญาณ IOR ประมาณ 30 NanoSec. เพื่อให้มั่นใจได้ว่า 8088 สามารถรับข้อมูลได้ถูกต้อง

2.1.4 IOW (อินพุตและเอาต์พุต พอร์ต Write)

ขาสัญญาณนี้เป็นเอาต์พุตแอดดิฟที่ลอจิก "0" ที่สร้างขึ้นโดย 8288 Bus Controller เพื่อใช้ในการแสดงว่าบัสไซเคิลที่เกิดขึ้นนี้เป็นบัสไซเคิลของการเขียนข้อมูลลงบน พอร์ต พอร์ตอินพุตและเอาต์พุต เพื่อให้ พอร์ต อินพุตและเอาต์พุต ที่มีแอดเดรสตรงกับแอดเดรส บนบัสแอดเดรสนั้นรับข้อมูลออกมาบนบัสข้อมูลไปเก็บไว้ อย่างก็ตามเนื่องจากในช่วงเวลาที่สัญญาณ IOW นี้แอดดิฟ (ลอจิก "0") นั้นข้อมูลบนบัสข้อมูลนั้นอาจจะยังไม่สมบูรณ์ดังนั้นในการออกแบบจึงควรใช้ขอบเขตของสัญญาณ IOW แทนขอบเขตในการทำให้พอร์ตอินพุตและเอาต์พุต ที่เกี่ยวข้องรับข้อมูลไปเก็บไว้ เพื่อให้ข้อมูลบนบัสข้อมูลสมบูรณ์เสียก่อน

2.1.5 AEN (Address Enable)

สัญญาณนี้เป็น เอาต์พุต ที่ใช้ในการแสดงว่าบัสไซเคิลที่เกิดขึ้นในช่วงเวลาที่สัญญาณ AEN แอดดิฟ (ลอจิก "1") นั้นเป็นบัสไซเคิลของขบวนการ DMA

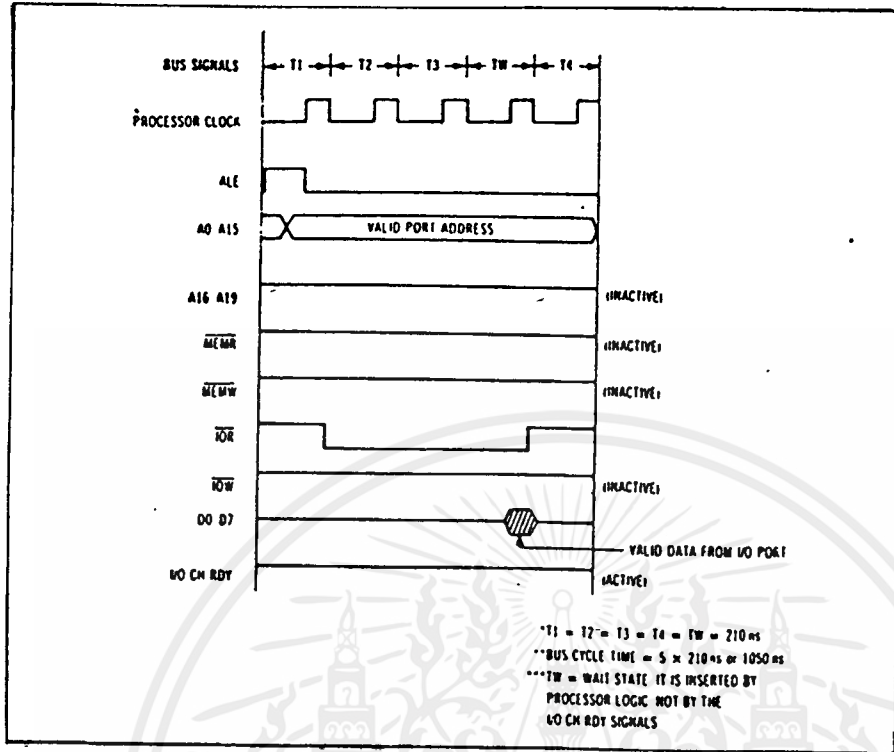
สำหรับ Main Board ของ IBM/PC นั้น จะใช้สัญญาณในการดิสแอกสาร์นี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้ ไม่นับเข้าเป็นไปใช้ประโยชน์ด้านใด ๆ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอเบิล (Disable) 8288 Bus Controller และจะใช้ดิสเอเบิลพอร์ต อินพุทและเอาต์พุท ต่างๆ ที่ไม่เกี่ยวข้องกับขบวนการ DMA ที่เกิดขึ้นนี้ ที่จำเป็นต้องทำการเช่นนี้ก็เพราะว่า ในระหว่างขบวนการ DMA นั้น 8237A-5 จะส่งแอดเดรสแอดเดรสของหน่วยความจำออกมาบนบัสแอดเดรส และจะทำให้สัญญาณ IOR หรือ IOW แอดตีฟด้วย ดังนั้นถ้าไม่ทำการ ดิสเอเบิลพอร์ต อินพุทและเอาต์พุท ที่ไม่เกี่ยวข้องไว้ก็อาจทำให้พอร์ต อินพุทและเอาต์พุท ที่มีแอดเดรสตรงกับค่าแอดเดรสบนบัสแอดเดรส (ซึ่งเป็นแอดเดรสแอดเดรสของหน่วยความจำ) นั้นทำการอ่านหรือส่งข้อมูลออกมาบนบัสข้อมูลทำให้เกิดความผิดพลาดขึ้นได้

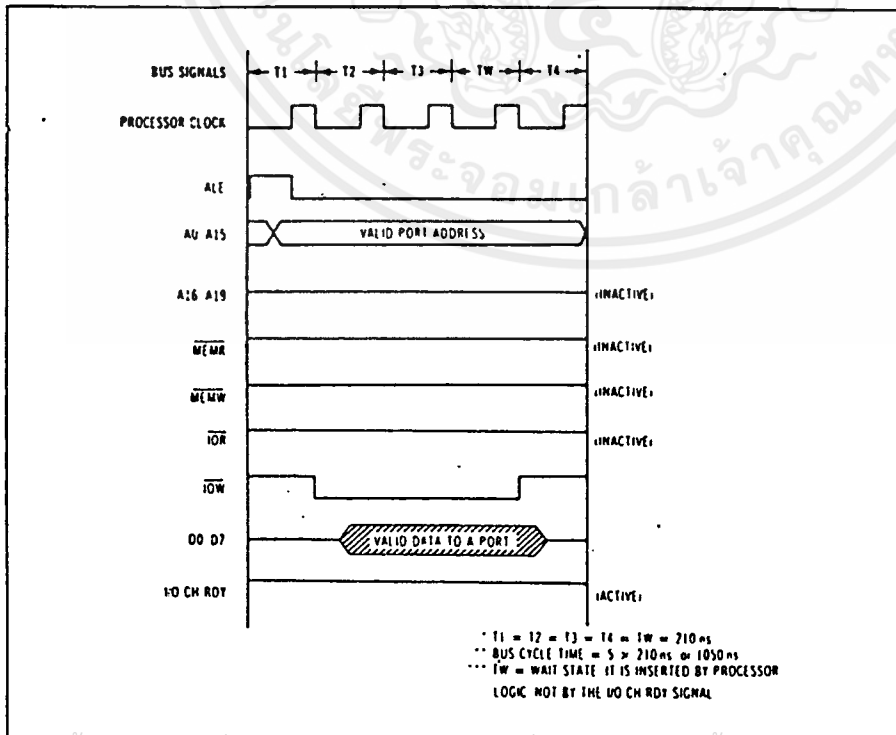


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 บัสไซเคิลของการอ่านข้อมูลจากพอร์ต อินพุตและเอาต์พุต



2.3 บัสไซเคิลของการเขียนข้อมูลลงบนพอร์ต อินพุตและเอาต์พุต

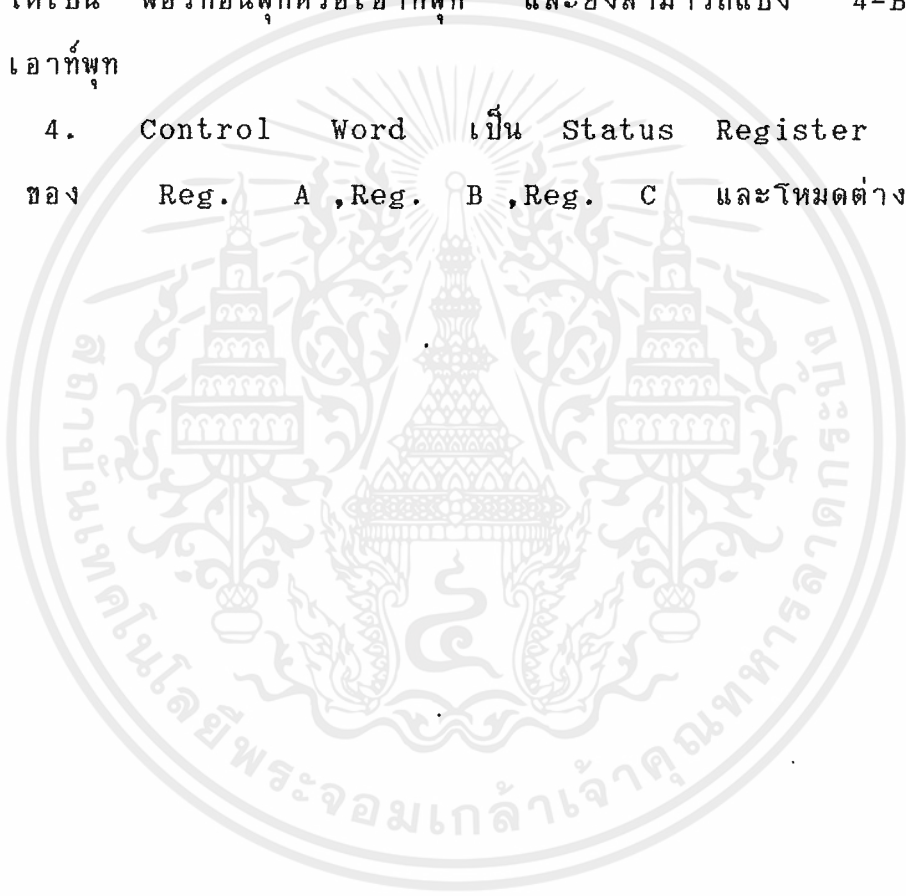


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ชิพพีพอร์ท #8255 PIA (Programmable Interface Adapter).

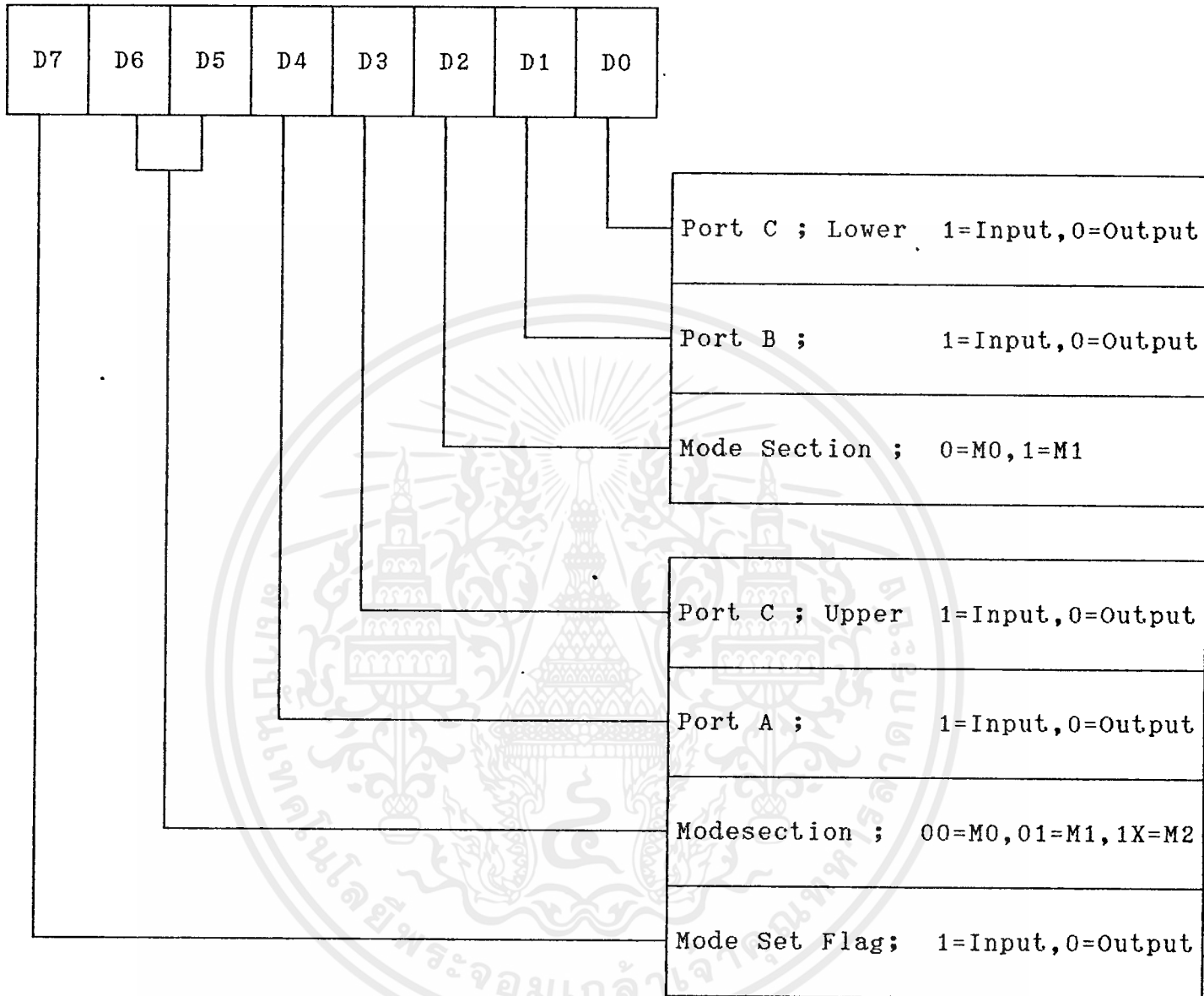
2.4.1 รีจิสเตอร์ภายใน ได้แบ่งออกเป็น 4 ส่วนด้วยกันคือ

1. Reg. A หรือ พอร์ท A เป็นแบบ 8-Bit ซึ่งสามารถโปรแกรมให้เป็น พอร์ทอินพุทหรือเอาต์พุท
2. Reg. B หรือ พอร์ท B เป็นแบบ 8-Bit ซึ่งสามารถโปรแกรมให้เป็น พอร์ทอินพุทหรือเอาต์พุท
3. Reg. C หรือ พอร์ท C เป็นแบบ 8-Bit หรือ 4-Bit ซึ่งสามารถโปรแกรมให้เป็น พอร์ทอินพุทหรือเอาต์พุท และยังสามารถแบ่ง 4-Bit บนหรือ 4-Bit ล่างเป็นอินพุทหรือเอาต์พุท
4. Control Word เป็น Status Register ที่กำหนดการเป็น อินพุทหรือเอาต์พุท ของ Reg. A ,Reg. B ,Reg. C และโหมดต่างๆ ของการทำงานซึ่งได้ถูกจัดไว้มีดังนี้



REGISTER ที่ใช้ในการควบคุมโหมดต่างๆ ของการทำงานซึ่งได้ถูกจัดไว้มีดังนี้

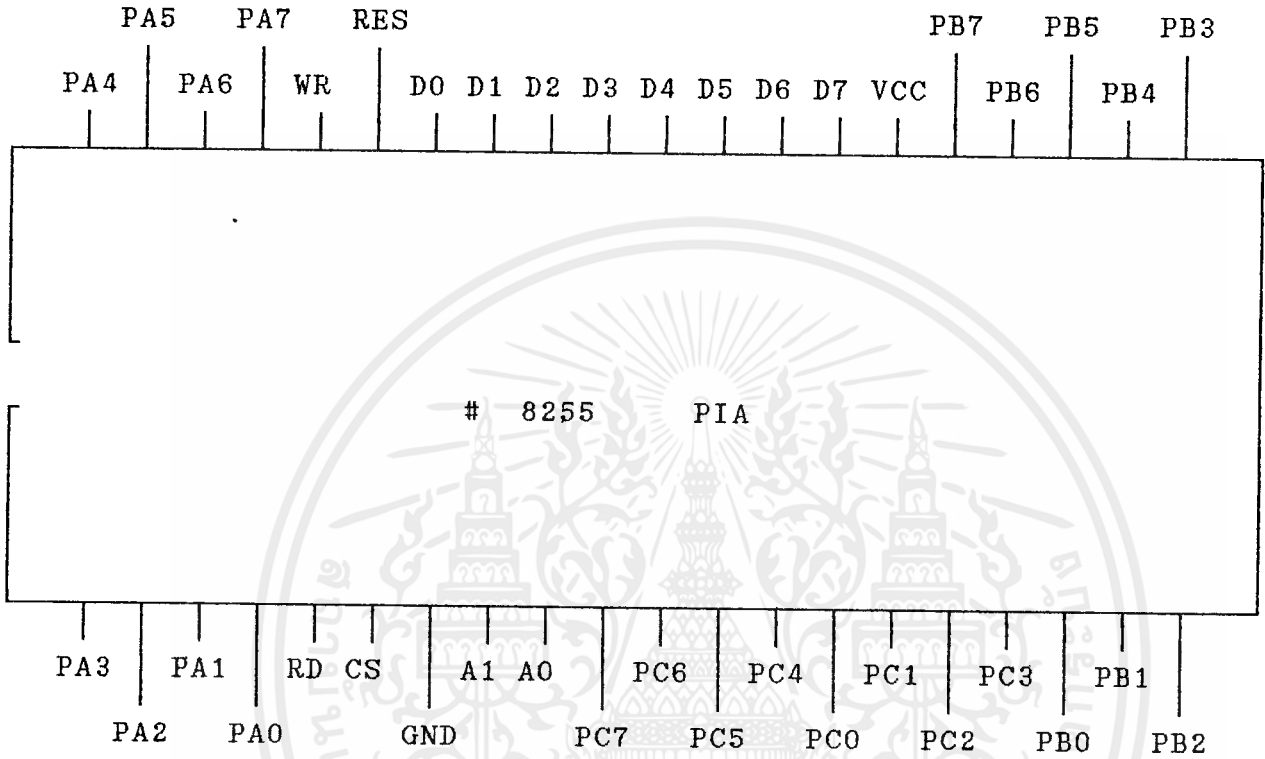
(Control Word Register)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.1 รายละเอียดของสัญญาณบน #8255 PIA

ขาสัญญาณบน #8255 PIA



DO-D7 เป็นสัญญาณอินพุตและเอาต์พุต แบบ Bi-Directional Bus จะเป็นทางผ่านของข้อมูลระหว่างพอร์ทต่างๆ ของ #8255 กับบัสข้อมูลภายในไมโครคอมพิวเตอร์ (Microcomputer) PC.

CS (Chip Select . Input) เป็นขาสัญญาณการเลือกการติดต่อ # 8255

RD (Read Input) เป็นขาสัญญาณการอ่าน #8255

WR (Write Input) เป็นขาสัญญาณการเขียนข้อมูลลงบน #8255

A0-A1 (Address Input) เป็นขาสัญญาณการเลือก Reg. ภายใน #8255

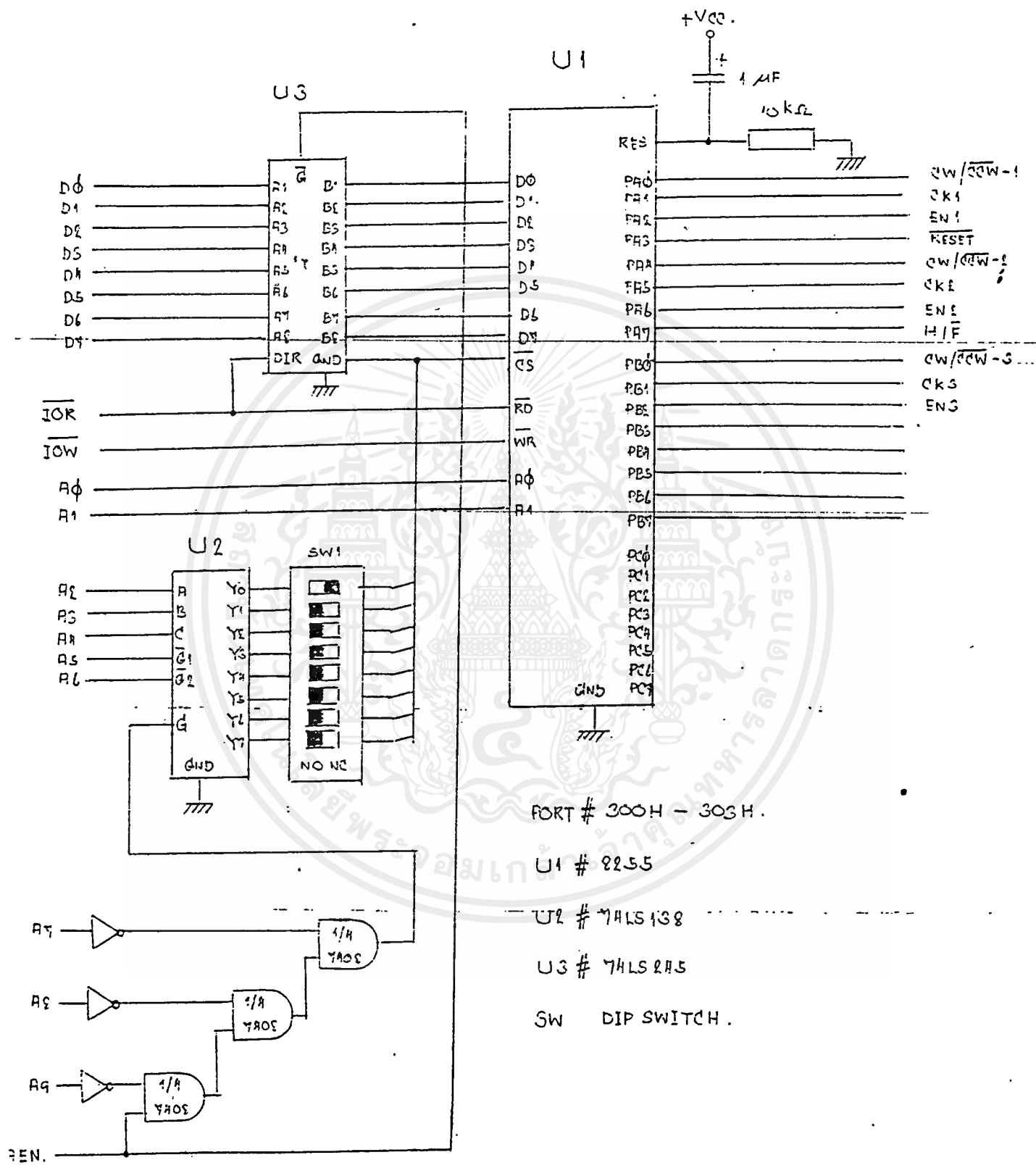
Res เป็นขาสัญญาณการ RESET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5 วงจรของ การ์ดอินพุตและเอาต์พุต (Input/Output Card Circuit)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าวิธีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

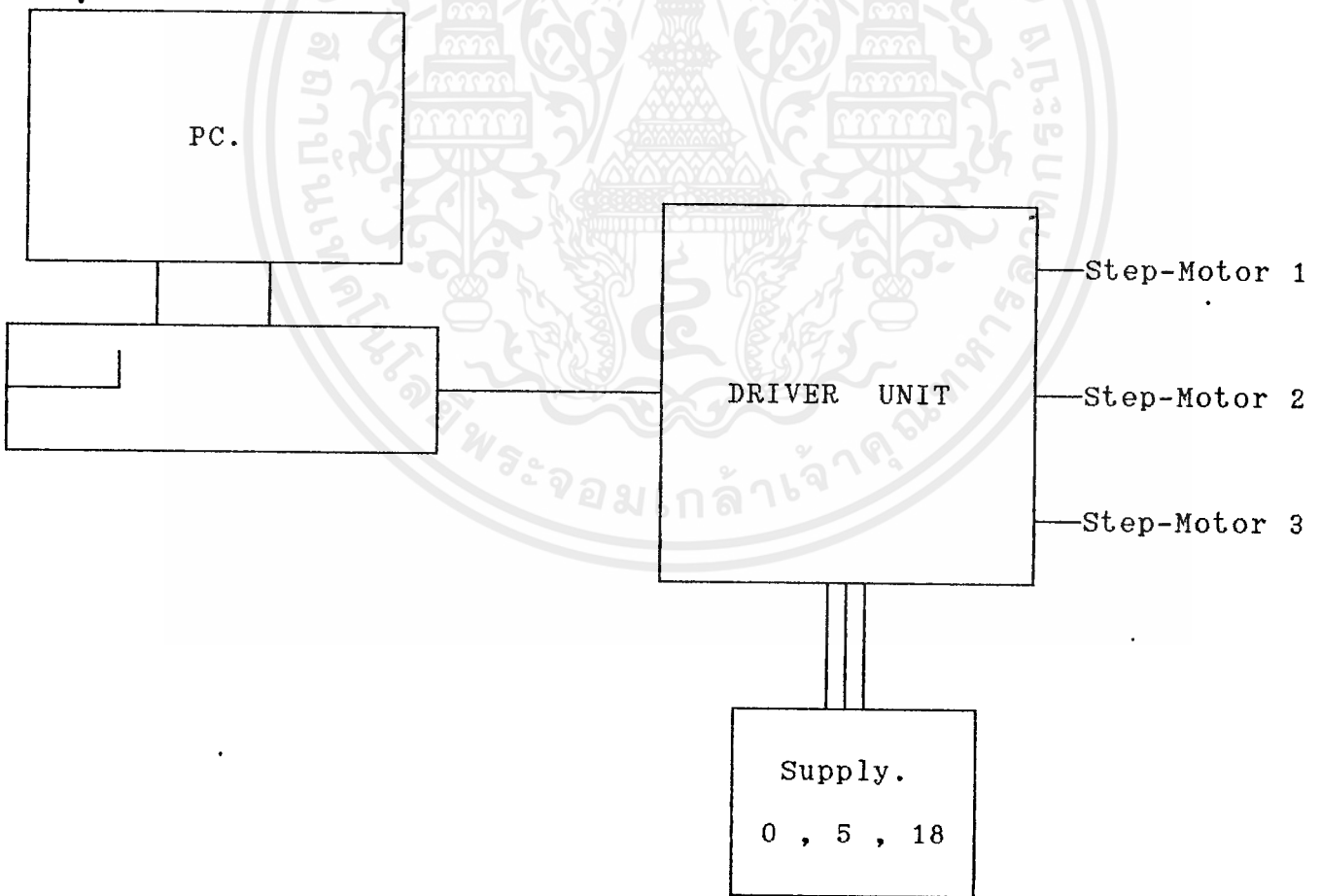
บทที่ 3

ไดรฟ์เวอร์

DRIVER UNIT

ในส่วนของ ไดรฟ์เวอร์ (Driver Unit) เป็นส่วนที่ต้องใช้พลังงานมาก จึงไม่สามารถนำไปเก็บไว้ใน ไมโครคอมพิวเตอร์ (Microcomputer) ได้ เพราะมีขนาดใหญ่และความร้อนที่เกิดขึ้น ในขณะที่ Driver Unit ทำงานค่อนข้างสูง จึงนำมาใช้งานอยู่ภายนอกไมโครคอมพิวเตอร์ (Microcomputer)

Driver Unit จะรับข้อมูลจาก การ์ดอินพุท/เอาต์พุท โดยผ่านทางสายนำสัญญาณขนาด 25 Line ขั้วต่อ (Connecton) แบบ DB-25 มีรูปแบบดังนี้



รูปแบบการรับส่งข้อมูลระหว่าง PC. กับ DRIVER UNIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนประกอบที่สำคัญของ Driver Unit นั้นพอจะแบ่งออกได้เป็นส่วนๆ ได้ดังนี้

3.1 ไอซี #L297 เป็นประเภท Stepper Motor Controller ที่มีคุณสมบัติหลายๆ อย่างอยู่ภายในตัว และเป็นส่วนที่สำคัญที่สุดในส่วนนี้ ซึ่งได้ถูกออกแบบมาเพื่อใช้ในงานทางด้านนี้ โดยเฉพาะ และมีคุณสมบัติโดยทั่วไปกล่าวคือ

คุณสมบัติโดยทั่วไป

1. Normal/Wave Drive
2. Half/Full Step Mode
3. Clockwise/Anticlockwise Direction
4. Programmable Load Current
5. Few External Components
6. Reset Input & Home Output
7. Enable Input
8. Step Pulse Doubler (L297A Only)

ขาการทำงานของ #L297 จะประกอบไปด้วย

SYNC	ซึ่งเป็นขาแสดง Synchronized ของ Chopper Oscillator
GND	Ground Connector
HOME	เป็นขาแสดงสภาวะการเริ่มต้นของเฟส
A	Motor Phase A Drive Signal For Power Step
INH1	Chopper Control
B	Motor Phase B Drive Signal For Power Step
C	Motor Phase C Drive Signal For Power Step
INH2	Chopper Control
D	Motor Phase D Drive Signal For Power Step
Enable	ขาสัญญาณของการเลือกชีพ
Control	ควบคุมการทำงาน ของ Chopper
Vs	Supply
SENS2	ควบคุมกระแสโดยการทำ Shopper
SENS1	ควบคุมกระแสโดยการทำ Shopper

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์และสงวนไว้เพื่อใช้ภายในเท่านั้น กรุณาอย่าได้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Vref	แรงดันอ้างอิงในการทำ Shopper
OSC	สร้างควมถี่ Shopper ($f=1/0.69 RC$)
CW/CCW	ขาสัญญาณของการควบคุมการหมุนทวนเข็มนาฬิกา
CLOCK	ขาสัญญาณของการ Step
HALF/FULL	ขาสัญญาณของการควบคุมโหมด
RESET	ขาสัญญาณของการ Reset

Motor Driving Phase Sequences

เมื่อป้อนสัญญาณ Clock เข้าที่ขา CLOCK ของ #L297 และทำการ SET ในโหมดต่างๆ แล้วก็จะได้ความสัมพันธ์ดังนี้

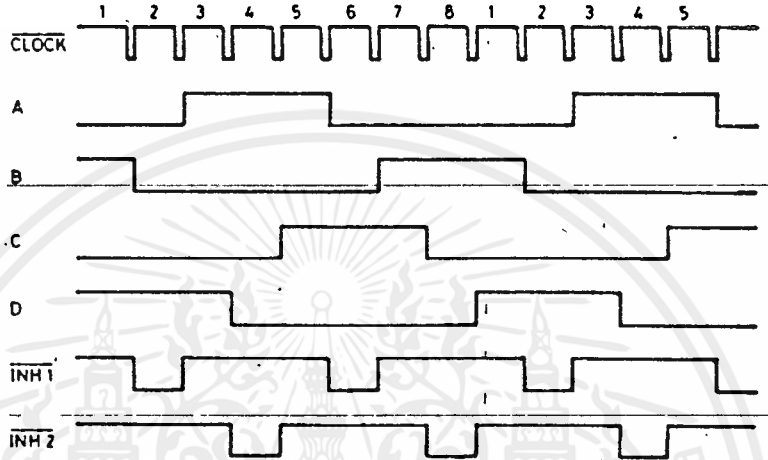
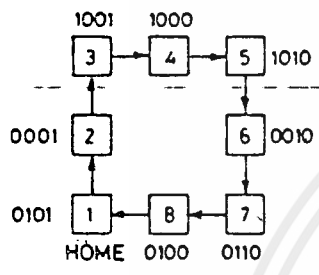
HALF STEP MODE

NORMAL DRIVE MODE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Half step mode

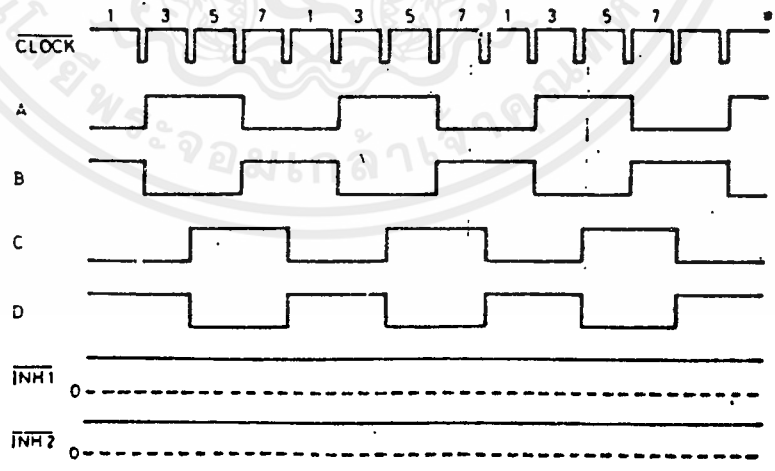
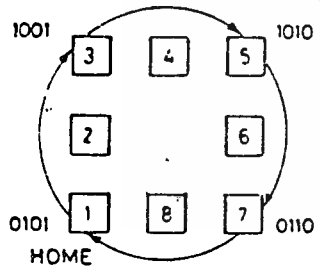
Half step mode is selected by a high level on the HALF/FULL input,



5-5841

Normal drive mode

Normal drive mode (also called "two-phase-on" drive) is selected by a low level on the HALF/FULL input when the translator is at an odd numbered state (1, 3, 5 or 7). In this mode the INH1 and INH2 outputs remain high throughout.

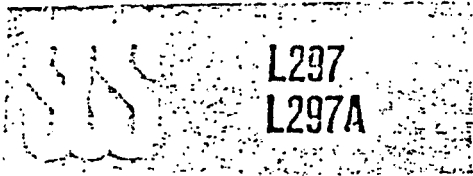


5-5842

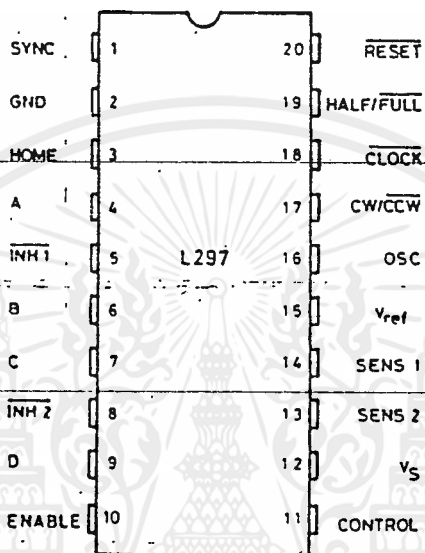
BLOCK DIAGRAM OF #L297



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

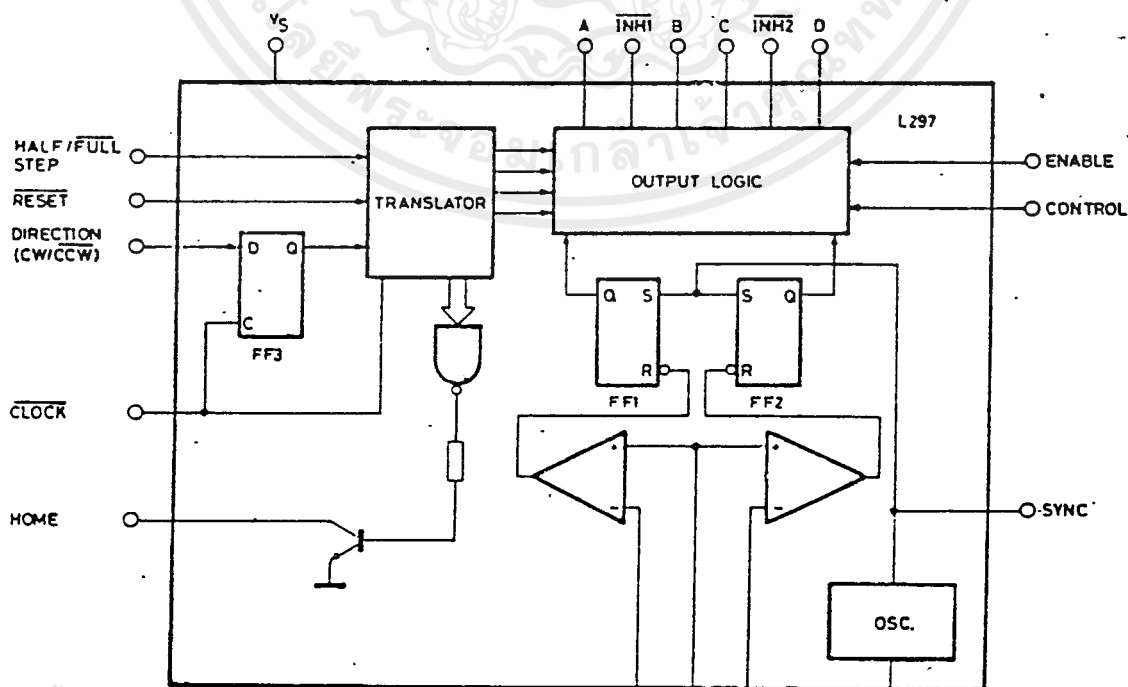


CONNECTION DIAGRAM



S-5839

BLOCK DIAGRAM



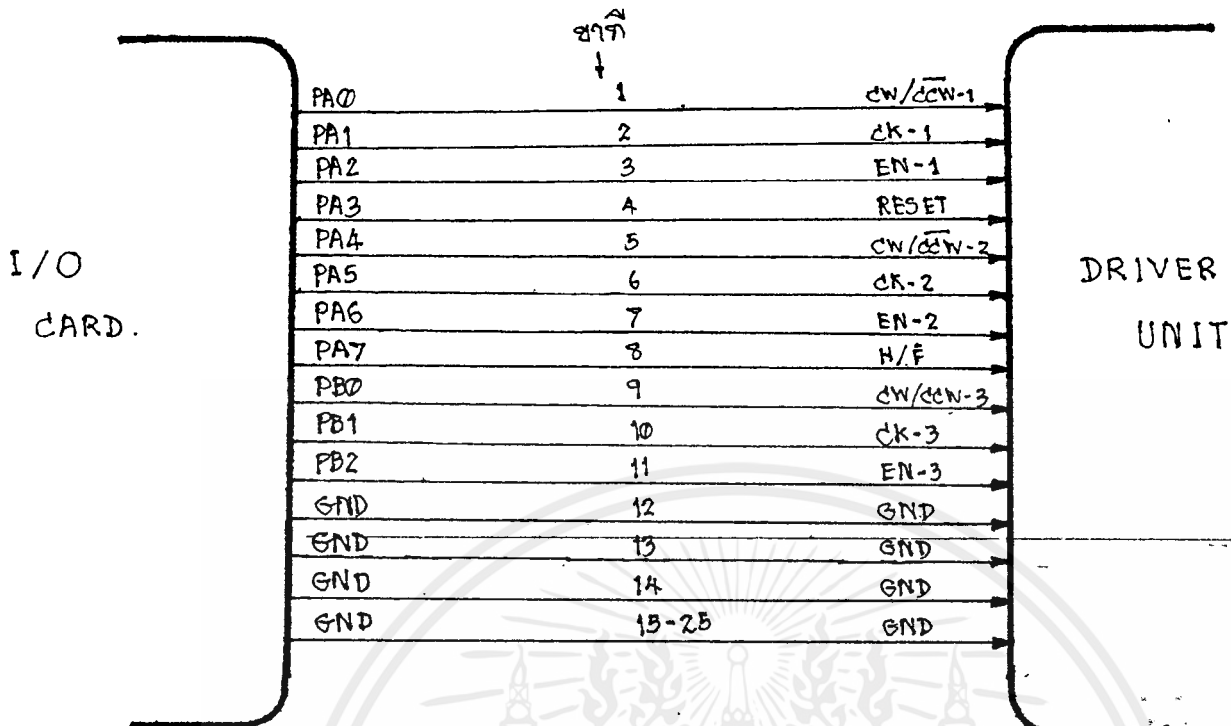
S-5838

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

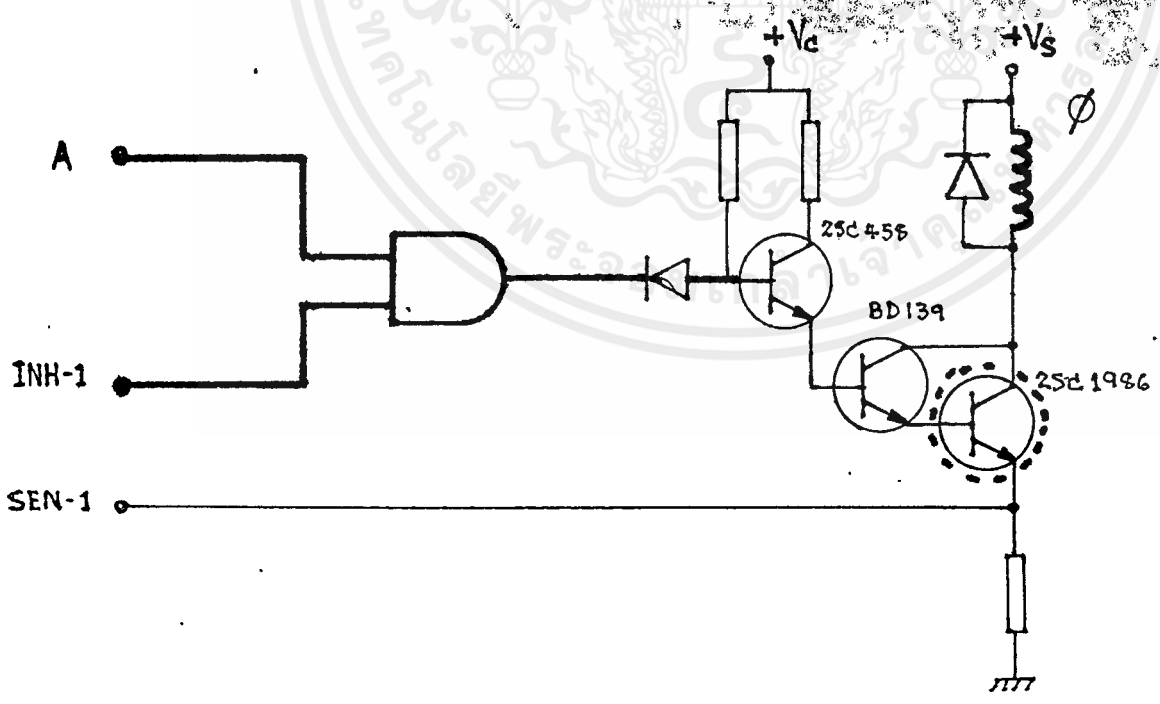
3.2 ส่วนของการ Drive Stepper Motor เป็นส่วนที่มีความสำคัญลดลงมาจากส่วนแรก
การ Drive จะกระทำได้ตั้งวงจรด้านล่างนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดงการเชื่อมต่อวงจร I/O กับ DRIVER



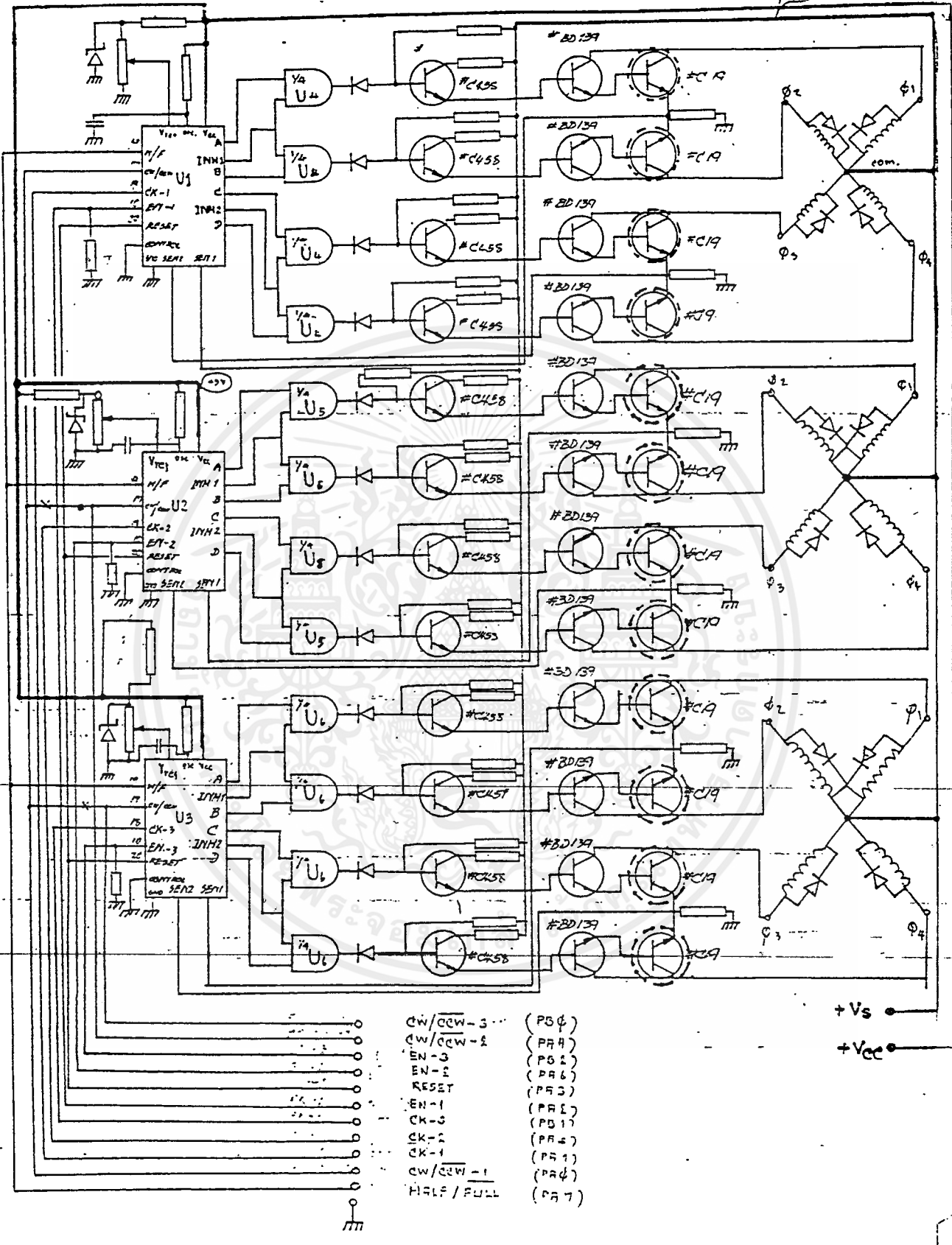
รูปแสดงวงจรขับกระแส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 วงจรของ DRIVER UNIT



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของวงจรวจรโดยรวม

** เริ่มจากการป้อนไฟเข้าเครื่อง (ทั้งไมโครคอมพิวเตอร์ และ DRIVER UNIT) ในขณะนี้จะไม่มีการแสไปที่ MOTOR เลยเป็นเพราะ R-Enable เป็นตัว Dis-Enable **

** CLOCK ที่ใช้ไม่ควรต่ำกว่า 500Hz ในโหมด FULL - Step จะทำให้ DRIVER UNIT เกิดความเสียหาย และไม่ควรใช้สูงเกิน 1670Hz (จุดสูงสุดของ MOTOR ที่ยังหมุนได้) ในโหมด HALF - Step **

การทำงาน

ทำการ SET ค่าบน Control Word ให้เป็น Port OUT ทั้งหมด โดยการส่งข้อมูล 80H ออกไปที่ Control Word (ดูจากการกำหนดค่าบน #8255) แล้วก็ให้ทำการ SET Bit บน Port ใดๆ Port หนึ่ง ที่จะทำการติดต่อกับ #L297 ตัวนั้นให้มีความตามโหมดที่เราต้องการ ให้ #L297 ตัวนั้นทำงาน ขณะนี้ให้ทำ Dis - Enable ไว้ก่อน (เพื่อป้องกัน TR #C1986 ทำงานมากเกินไปจนความจำเป็น)

เมื่อพร้อมใช้งาน ให้ทำการ Enable และป้อน Clock ผ่านทาง Bit ของ Port ที่ได้กำหนดให้เป็นสัญญาณ Clock ก็จะมีสัญญาณออกทางเอาต์พุตของ #L297 และสัญญาณที่ได้จะผ่านไป 7408 (AND Gate) เพื่อทำ Chopper แล้วจึงส่งสัญญาณที่ได้ทำ Choper แล้วไป (กระแสประมาณ 1 mA.) ขยาย ในส่วนของการขยายกระแสต่ำที่วงจรถอง TR #C458 กระแสที่ถูกขยายแล้ว จะมีค่าประมาณ 20 mA. เป็นกระแสที่เพียงพอในการขับ TRANSistor กำลังคือ #C139 กับ #C1986 ซึ่งขยายแบบ Darlington และสามารถขับ LOAD ที่ต้องการกระแสไม่ควรเกิน 8 AMP.

Choper เป็นส่วนหนึ่งที่ช่วยป้องกันความเสียหายแก่ Driver Unit คือ จะมีการ Choper ในขณะที่มีแรงดันไฟเลี้ยงวงจรรีบที่มีค่าสูงกว่าความต้องการของ Stepping Motor - Coil การขับ LOAD ของวงจรมันจะขับ LOAD ที่เป็น MOTOR แบบ Stepping Motor 4 - PHASE ถ้าต้องการให้หมุนกี่ Step ก็ให้ทำการป้อน Clock ตามจำนวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Step ที่เราต้องการ แต่ต้องทำการ Dis - Enable ทุกครั้งที่ไม่ใช้งาน (เพื่อป้องกัน TR
#C1986 ทำงานมากเกินไปจนความจำเป็น)



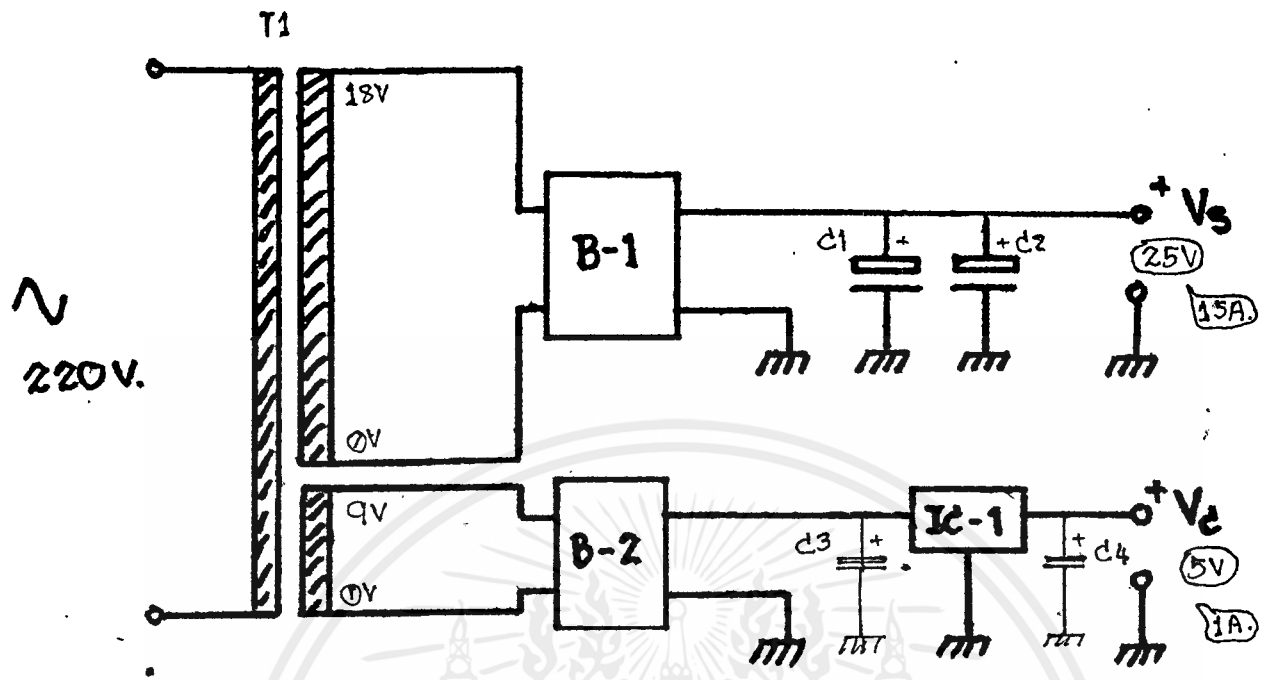
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOR No. (RUN)	PORT-A								PORT-B		
	H/F PA7	EN-2 PA6	CK-2 PA5	CW/CCW-2 PA4	RESET PA3	EN-1 PA2	CK-1 PA1	CW/CCW-1 PA0	EN-3 PB2	CK-3 PB1	CW/CCW-3 PB0
# 1	↓	L	L	L	H	H	↓	↓	L	L	L
# 2	↓	H	↓	↓	H	L	L	L	L	L	L
# 3	↓	L	L	L	H	L	L	L	H	↓	↓

ตารางแสดงการทำงานของ MOTOR

- * ↓ ; การเปลี่ยนขั้ว
- L ; สภาวะ LOGIC - "0"
- H ; สภาวะ LOGIC - "1"
- H/F ; HALF / FULL
- EN-x ; ENABLE # x
- CK-x ; CLOCK # x
- CW/CCW-x ; CLOCK WISE / COUNTERCLOCKWISE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรแหล่ง POWER SUPPLY

โปรแกรมที่ใช้ในการทดลอง

```
Program Stepping-Motor-Test (INPUT-OUTPUT);
```

```
Uses Crt,Dos;
```

```
Const   PortA  = $300;      Enable1 = $04;
        PortB  = $301;      Enable2 = $40;
        PortC  = $302;      Enable3 = $04;
        CWord  = $303;      Cw_Ccw1  = $01;
        Clock1 = $02;       Cw_Ccw2  = $10;
        Clock2 = $20;       Cw_Ccw3  = $01;
        Clock3 = $02;       Half_Full = $80;
        Reset  = $08;       Control_W = $80;
        Loop   = 18;

Var     Key   : integer;    Sig,Signal: integer;
        Clock : integer;    ST       : String;
        Enable : integer;   EN        : String;
        Halffull : integer; HF        : String;
        Cw_Ccw : integer;   CW        : String;
        Step   : integer;   AM        : integer;
        C,Counter : integer; AM1       : integer;
```

```
Procedure Port_Cword;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{ Set SYSTEM }
```

```
Begin
```

```
Port[Cword]:=Control_W;
gotoXY(27,13);write('CNC : HardWare & Test');
sound(1200);
delay(200);
nosound;
delay(200);
sound(1200);
delay(400);
nosound;
delay(2000);
clrscr;
```

```
end;
```

```
Procedure Screen;
```

```
Begin
```

```
gotoXY(50,19);write('Controller & FORMAT');
gotoXY(57,21);write(ST);
gotoXY(57,22);write(HF);
gotoXY(57,23);write(CW);
```

```
end;
```

```
Procedure Screen1;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Begin

```
gotoXY(36,11);write('MENU');
    gotoXY(24,14);write('F1 - Step Motor #1 Controller');
    gotoXY(24,15);write('F2 - Step Motor #2 Controller');
    gotoXY(24,16);write('F3 - Step Motor #3 Controller');
    gotoXY(1,25);write('Esc to EXIT');
```

end;

Procedure Screen2;

begin

```
clrscr;
gotoXY(30,11);write('RUNning FORMAT');
    gotoXY(30,14);write('F1 - Half - Step');
    gotoXY(30,15);write('F2 - Full - Step');
gotoXY(1,25);write('Esc to EXIT');
```

end;

Procedure Screen3;

begin

```
clrscr;
gotoXY(30,11);write('RUNning FORMAT');
    gotoXY(34,14);write('F1 - CCW');
    gotoXY(34,15);write('F2 - CW');
```

```
gotoXY(1,25);write('Esc to EXIT');
```

end;

Procedure Screen4;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
  clrscr;Screen;
  gotoXY(30,11);write('RUNning  FORMAT');
  gotoXY(27,14);write('Amonth  Of  Step  = ');read(Step);

  gotoXY(1,25);write('Esc to EXIT');

```

```
end;
```

```
Procedure  Read_Key;
```

```
Var  Regs  :  registers;
```

```
Begin
```

```
  Regs.AH  :=  $0;
```

```
  Intr  ($016,Regs);
```

```
  Key  :=  Regs.Ah;
```

```
  sound(1100);delay(100);nosound;
```

```
end;
```

```
Procedure  Code1;
```

```
  begin
```

```
    Repeat
```

```
      Read_Key;
```

```
      If  Key=59  Then
```

```
        begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Clock := Clock1;
    Enable := Enable1;
    ST := ('MOTOR #1');
    Exit;
end;

```

```

If Key=60 Then

```

```

    begin
        Clock := Clock2;
        Enable := Enable2;
        ST := ('MOTOR #2');
        Exit;
    end;

```

```

If Key=61 Then

```

```

    begin
        Clock := Clock3;
        Enable := Enable3;
        ST := ('MOTOR #3');
        Exit;
    end;

```

```

Until Key=01;

```

```

end;

```

Procedure Code2;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
  Repeat
    Read_Key;
    If Key=59 Then begin
      Halffull :=Half_Full;
      HF:='Half - Step';
      Counter:=Loop;
      EXIT;
    end;

    If Key=60 Then begin
      Halffull :=0;
      HF:='Full - Step';
      Counter:=Loop *4;
      EXIT;
    end;
  Until Key=01;

end;

Procedure Code3;
begin
  Repeat
    Read_Key;
    If Key=59 Then
      begin
        IF ST='MOTOR #1' then
          begin
            Cw_Ccw:=Cw_Ccw1;
            CW:='CCW';

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EXIT;

end;

IF ST='MOTOR #2' then
begin
    Cw_Ccw:=Cw_Ccw2;
    CW:='CCW';
    EXIT;
end;

IF ST='MOTOR #3' then
begin
    Cw_Ccw:=Cw_Ccw3;
    CW:='CCW';
    EXIT;
end;
end;

If Key=60 Then
begin
begin
IF ST='MOTOR #1' then
begin
    Cw_Ccw:=0;
    CW:='CW';
    EXIT;
end;
IF ST='MOTOR #2' then
begin
    Cw_Ccw:=0;
    CW:='CW';

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

EXIT;

end;

IF ST='MOTOR #3' then

begin

    Cw_Ccw:=0;

    CW:='CW';

    EXIT;

end;

end;

end;

Until Key=01;

end;

Procedure Code4;

begin

Screen4;

clrscr;

gotoXY(30,11);write('RUNning  FORMAT');

gotoXY(34,14);write(' R - RUN');

gotoXY(34,15);write('Esc - EXIT');

Screen;

AM:=Step;

gotoXY(57,24);write('Step-Counter = ',AM);

end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{ MAIN }
```

```
Begin
```

```
Repeat
```

```
  Clrscr;ST:='';HF:=ST;CW:=HF;
```

```
  Clock:=0;Enable:=0;Halffull:=0;Cw_Ccw:=0;
```

```
  Port_Cword;
```

```
  Screen1;
```

```
  Code1;
```

```
  IF Key = 01 then EXIT;
```

```
  Screen2;
```

```
  Screen;
```

```
  Code2;IF Key = 01 then EXIT;
```

```
  Screen3;
```

```
  Screen;
```

```
  Code3;IF Key = 01 then EXIT;
```

```
  Screen;
```

```
  code4;IF Key = 01 then EXIT;
```

```
Signal:=Clock + Halffull + Enable + Cw_Ccw + Reset;
```

```
IF ST='MOTOR #3' then
```

```
  Begin
```

```
    Sig:=Halffull + Reset;
```

```
    Port[PortA]:=Sig;
```

```
    Signal:=Enable + Cw_Ccw + Clock;
```

```
  end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Repeat

 Read_Key;

 IF Key=01 then EXIT;

Until Key = 19;

 AM:=Step;

 FOR AM1:=1 to AM do

Begin

 FOR c:=0 to Counter do Begin end;

 IF ST='MOTOR #3' then

 Begin

 Port[PortA]:=Sig;

 Signal:=Signal - Clock;

 Port[PortB]:=Signal;

 end

 Else

 Begin

 Signal:=Signal - Clock;

 Port[PortA]:=Signal;

 end;

 FOR c:=0 to Counter do Begin end;

 IF ST='MOTOR #3' then

 Begin

 Signal:=Signal + Clock;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Port[PortB]:=Signal;
```

```
end
```

```
Else
```

```
Begin
```

```
Signal:=Signal + Clock;
```

```
Port[PortA]:=Signal;
```

```
end;
```

```
AM:=AM-1;
```

```
gotoXY(57,24);write('Step-Counter = ',AM,' ');
```

```
end;
```

```
Port[PortA]:=0;
```

```
Port[PortB]:=0;
```

```
delay(100);
```

```
gotoXY(1,25);Write('Press Any Key To Continue Esc To EXIT');
```

```
Read_Key;
```

```
delay(1000)
```

```
Until Key=01
```

```
End.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุป

สรุปผลการทดลอง

หลังจากได้ทำการทดลอง แล้วจะพบว่า STEPPER MOTOR นั้นสามารถหมุนด้วยความเร็วที่ไม่มาก คือ CLOCK ที่ใช้ไม่ควรต่ำกว่า 500Hz ในโหมด FULL - Step จะทำให้ DRIVER UNIT เกิดความเสียหาย และไม่ควรใช้สูงเกิน 1670Hz (จุดสูงสุดของ MOTOR) ที่ยังหมุนได้) ในโหมด HALF - Step

ถ้าหมุนตาม CW แล้วกลับมาหมุนตาม CCW ในทันทีก็จะมีผลทำให้ STEPPER MOTOR หยุดหมุนทันที จะต้องพักไว้สักคู่หนึ่งแล้วจึงเปลี่ยนทิศทาง

ปัญหาที่เกิดขึ้นและข้อเสนอแนะ

การหมุน STEPPER MOTOR ให้มีความเร็วและแรงบิดสูง จำเป็นที่จะต้องใช้แหล่งจ่ายพลังงานที่มีค่ากระแสที่สูงพอสมควร ซึ่งก็ควรที่จะสูงกว่า LOAD โดยประมาณสามเท่าถึงจะทำให้การทำงานเกิดความสมบูรณ์

- | | | |
|--------------------------------------|-------------------|----------------|
| 1. แอดวานซ์เทอร์โบปาสคาล Version 4.0 | สุรศักดิ์ | สงวนพงษ์ |
| 2. ภาษาปาสคาล 4.0 - 5.0 | บุญเลิศ | เอี่ยมทัศนาศนา |
| 3. เทคโนโลยีคอมพิวเตอร์ 16 BIT | ศจ.ดร. ยืน | ภูววรรณ |
| | ดร. ชัยยงค์ | วงศ์ชัยสุวัฒน์ |
| | ดร. ไพศาล | สงวนหมู่ |
| 4. การอินเตอร์เฟส. IBM PC | ชานินทร์ | ถาวรศาสตร์วงศ์ |
| | ทินกร | ตุ๊ก |
| 5. คู่มือซีพียูพอร์ทและหน่วยความจำ | บ. ชีเอ็ดยูเคชั่น | |
| 6. LINEAR INTEGRATED CIRCUIT | MOTOROLA | LINEAR APP. |
| 7. TTL DATA BOOK | บ. ชีเอ็ดยูเคชั่น | |
| 8. วารสารทางคอมพิวเตอร์ | | |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้