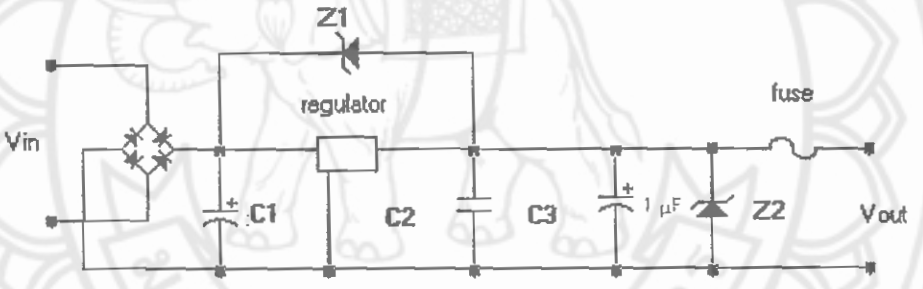
บทที่ 2

หลักการและทฤษฎี

เนื้อหาในบทนี้กล่าวถึงเรื่องพาวเวอร์ซัพพลาย วงจรเปลี่ยนสัญญาณอะนาลอกเป็นดิจิตอล คำอธิบายเกี่ยวกับบอร์ดไมโครคอนโทรลเลอร์ การใช้งานรอมอีมูเลเตอร์ หลักการเซอร์โวแอมปลิไฟร์ การทำงานของไดรเวอร์ในการขับมอเตอร์

2.1 พาวเวอร์ซัพพลาย (Power Supply)

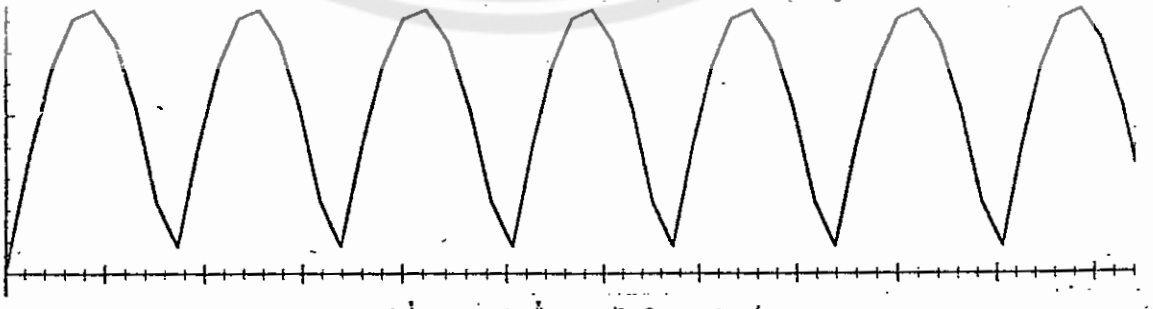
วงจรพาวเวอร์ซัพพลายเป็นวงจรที่จ่ายพลังงานให้แก่วงจรชุดอื่น ๆ วงจรที่ใช้เป็นดังนี้



รูปที่ 2.1 วงจรพาวเวอร์ซัพพลาย

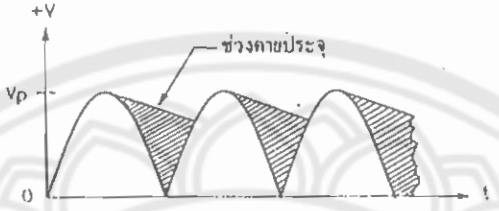
วงจรพาวเวอร์ซัพพลายประกอบไปด้วย

2.1.1 ชุดเรียงกระแสเต็มคลื่นแบบบริดจ์ (Bridge Rectifier) ซึ่งประกอบไปด้วยไดโอด 4 ตัว ไฟฟ้ากระแสตรงที่ได้ออกมาจะได้ไฟฟ้ากระแสตรงแบบกระเพื่อม ดังรูปที่ 2.2



รูปที่ 2 .2 รูปคลื่นหลังไดโอดบริดจ์

2.1.2 ตัวเก็บประจุ C1 ทำหน้าที่เป็นตัวกรองกระแส (Low Pass Filter) ทำให้รูปคลื่นไฟฟ้ากระแสตรงที่ได้นั้นราบเรียบขึ้นทำให้ใกล้เคียงกับไฟฟ้ากระแสตรงดังรูปที่ 2.3



รูปที่ 2.3 รูปคลื่นเมื่อผ่านตัวเก็บประจุ 2200 μF

2.1.3 วงจรรักษาระดับแรงดัน (Voltage Regulator) ซึ่งทำให้ได้ระดับแรงดันที่คงที่โดยใช้ไอซี 3 ขา (Three-Terminal Regulator)

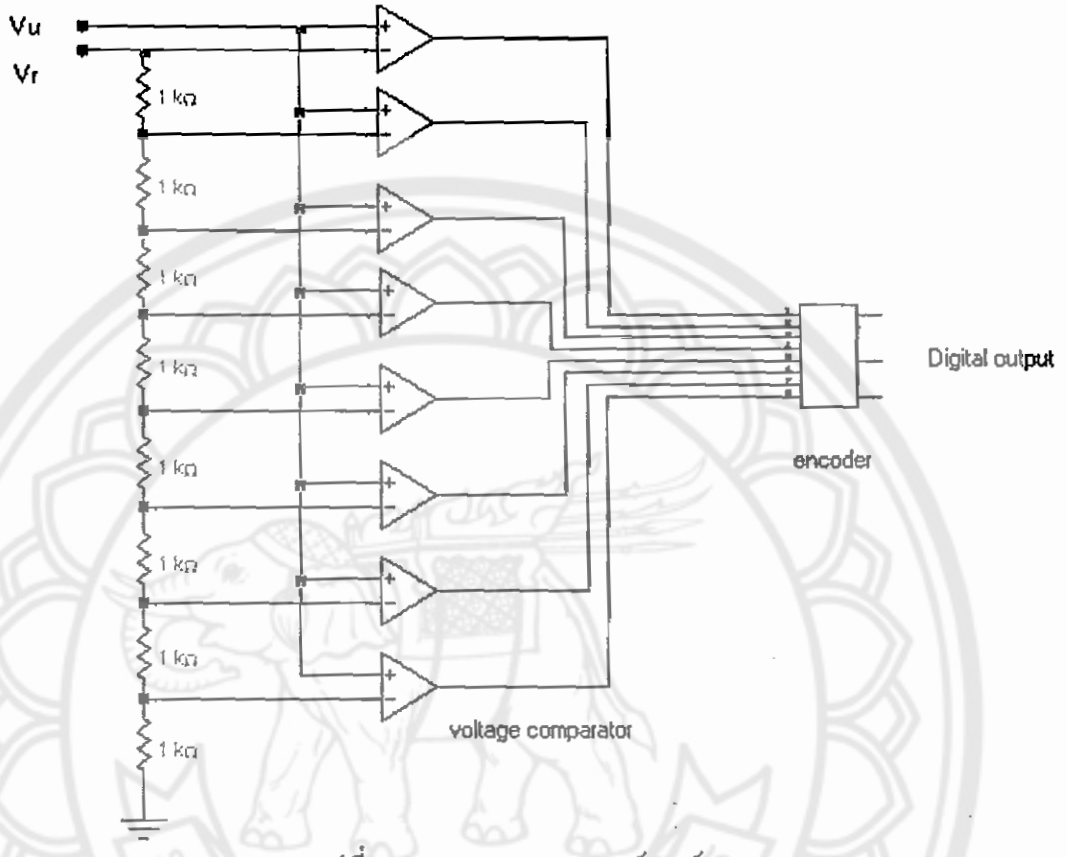
2.1.4 ตัวเก็บประจุ C2 ใช้สำหรับปรับค่าความต้านทานต่อสัญญาณรบกวนให้ดีขึ้น โดยตัวเก็บประจุเหล่านี้ใช้สำหรับกดสัญญาณรบกวนต่าง ๆ ซึ่งประกอบไปด้วยพัลส์ความถี่สูงๆ

2.1.5 ตัวเก็บประจุ C3 สำหรับตัวเก็บประจุตัวนี้เป็นตัวที่เพิ่มเติมเข้ามาแต่ขอให้มีในวงจรที่มีการเปลี่ยนแปลงของกระแสโหลดอย่างมาก ในช่วงเวลาสั้นๆ เช่น วงจรดิจิตอล C3 ทำหน้าที่ปรับค่าการตอบสนองทางด้านทรานเซียน (Transient) ของวงจรรักษาระดับแรงดัน ตัวเก็บประจุทำหน้าที่เก็บสะสมประจุเพื่อที่จะป้อนกระแสเข้าสู่โหลดในช่วงเวลาสั้นๆ ในขณะวงจรรักษาระดับแรงดันกำลังปรับตัวเองเพื่อรับกับความต้องการกระแสสูงสุด

2.1.6 ซีเนอร์ไดโอด (Z1) โดยจะทำหน้าที่เมื่อมีกระแสไหลย้อนกลับมาจากทางเอาท์พุท โดยจะป้องกันไม่ให้กระแสไหลเข้าไปในไอซีรักษาระดับแรงดัน เพื่อไม่ให้เกิดความเสียหายต่อตัวไอซีนั้น

2.1.7 ซีเนอร์ไดโอด (Z2) โดยทำหน้าที่ป้องกันระดับแรงดันเกินอันเนื่องมาจากไอซีรักษาระดับแรงดันผิดปกติ ซึ่งอาจทำให้วงจรภายนอกที่รับกระแสและแรงดันมาจากแหล่งจ่ายไฟ เกิดความเสียหายได้

2.2 วงจรเปลี่ยนสัญญาณอะนาลอกเป็นดิจิตอล (Analog-to-Digital Converter; ADC)



รูปที่ 2.4 วงจร A/D คอนเวอร์เตอร์

คอนเวอร์เตอร์ชนิดนี้เป็นชนิดที่มีความสามารถทำงานได้เร็วที่สุดโดยจะมีรูปแบบวงจรดังรูปที่ 2.4 โดยจะใช้ออปแอมป์ซึ่งทำหน้าที่เป็นตัวเปรียบเทียบแรงดัน (Voltage Comparator) ทั้งหมด $2^n - 1$ ต่อขนานกันโดยออปแอมป์แต่ละตัวจะต่ออยู่กับแรงดันอ้างอิง ส่วนแรงดันอินพุทจะต่อกับ ไบนารี่รีซิสเตอร์แลดเดอร์ (Binary Resistor Ladder) เพื่อให้แรงดันเข้าสู่ออปแอมป์สูงกว่าที่เข้าสู่ออปแอมป์ของตัวที่ต่ำกว่า 1 LSB

เมื่อสัญญาณอะนาลอกเข้าสู่ออปแอมป์ โดยออปแอมป์ตัวที่มี $VR^* < Vn$ จะเปลี่ยนสถานะเป็น 1 (High) ส่วนตัวที่ $VR^* > Vn$ จะคงสถานะ 0 (Low) เนื่องมาจากพวกออปแอมป์ชนิดนี้จะเป็นแบบแลทชิ่ง (Latching) ดังนั้นมันจะคงสถานะไว้จนกระทั่งค่าเหล่านั้นถูกส่งเข้าระบบดิจิตอลลอจิกเกต (Digital Logic Gate) ซึ่งในที่นี้ใช้ตัวเข้ารหัส (Encoder) ซึ่งจะทำหน้าที่เปลี่ยนพาราแลลคอมพาราเตอร์เวิร์ด (Parallel Comparator Word) เป็นรหัสไบนารี่ (Binary Code)

2.3 ซิงเกิลบอร์ดที่ใช้บนคอมพิวเตอร์สำหรับรถไฟฟ้าคนพิการ CP-SB31 (Single Board 31 on PC for Independent of Clipples)

CP-SB31 ถูกสร้างขึ้นเพื่อใช้ในงานควบคุม ซึ่งตรงกับหน้าที่หลักของซีพียูในตระกูล MCS 51 คือเป็นไมโครคอนโทรลเลอร์ โครงสร้างทางกายภาพของบอร์ด CP-SB31 มีดังนี้

ลักษณะของบอร์ด CP-SB31

- CPU 8031(On Board) หรือ 8032, 8052, 8751

Memory

- มี Socket ขนาด 28 PIN 2 ตัว สามารถใส่หน่วยความจำได้สูงสุด 96 kB

I/O

- 8 x 3 บิท	อินพุต/เอาท์พุต (I/O 8255)
- 8 x 1 บิท	อินพุต/เอาท์พุต (I/O Port1)
- 1 พอร์ทอนุกรม	(Serial Port RS 232)

Power

- 10 VDC	แจ็คแหล่งจ่ายไฟ (Power Suply Jack)
- 5 VDC	บอร์ดเรกูเลเตอร์ 7805 (Regulate 7805 Board)

คุณลักษณะพิเศษของ CP-SB31

- หน่วยความจำ สามารถเลือกได้ทั้งขนาด, ตำแหน่ง และลักษณะการทำงาน (Data Memory, Code Memory, Code & Data Memory) ดูในหัวข้อ การติดตั้งหน่วยความจำ

- สามารถพัฒนาโปรแกรมได้ทั้งภาษาแอสเซมบลี้ (ร่วมกับ SB31-Debugger) หรือภาษาเบสิค เพื่อใช้ 8052 AH-Basic หรือ ET-8051 EM หรือใช้กับชุดจำลองรอม (ET Eprom Emulator) ก็ได้

- ต่อกับ แอลซีดี (LCD) ได้ทันทีโดยไม่ต้องใช้ อินพุต/เอาท์พุต (I/O) พอร์ท

- มี อินพุต/เอาท์พุต (I/O) พอร์ต ขนาด 8 บิท ถึง 4 พอร์ท

- ต่อร่วมกับอุปกรณ์สนับสนุนของบริษัทอีทีที ได้ทันที เช่น SSRAC, RTC, 72IO, ET-AD

การติดตั้งหน่วยความจำให้กับ CP-SB31

เนื่องจาก CP-SB31 ถูกสร้างมาให้เป็นอิสระในการเลือกใช้หน่วยความจำได้หลายขนาดทั้งอีพรอม (EPROM) และแรม (RAM) รวมทั้งตำแหน่งแอดเดรสของหน่วยความจำผู้ใช้ก็ยังสามารถกำหนดได้ตามต้องการซึ่งทั้งหมดนี้ขึ้นอยู่กับการใส่ตำแหน่งของจัมพ์เปอร์ต่าง ๆ ให้ถูกต้อง ซึ่ง U3 และ U4 จะถูกควบคุมด้วยจัมพ์เปอร์ซึ่งมีรายละเอียดดังนี้

U3 JP3 เลือกบอร์ดของหน่วยความจำที่ใส่อยู่บน U3 (2764 , 27256 , 27512 , 6264 , 62556)

JP4 สำหรับเลือกว่าจะให้หน่วยความจำที่ U3 เป็นหน่วยความจำข้อมูล หรือหน่วยความจำรหัส หรือเป็นทั้งหน่วยความจำรหัสและข้อมูล (Code & Data Memory)

JP7 เลือกตำแหน่งเริ่มต้นและขนาดของหน่วยความจำ U3

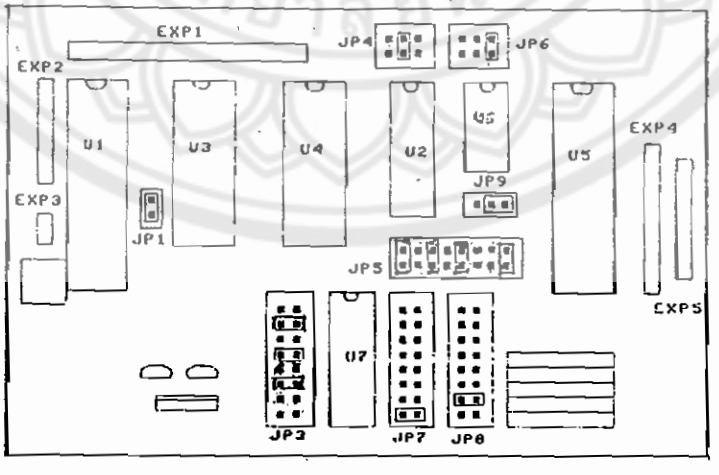
U4 JP5 เลือกบอร์ดของหน่วยความจำที่ใส่อยู่บน U4 (27266 , 6116 , 62256)

JP6 เลือกลักษณะการทำงานของ U4 ว่าจะให้เป็นหน่วยความจำข้อมูล หรือหน่วยความจำรหัส หรือเป็นทั้งสองอย่าง

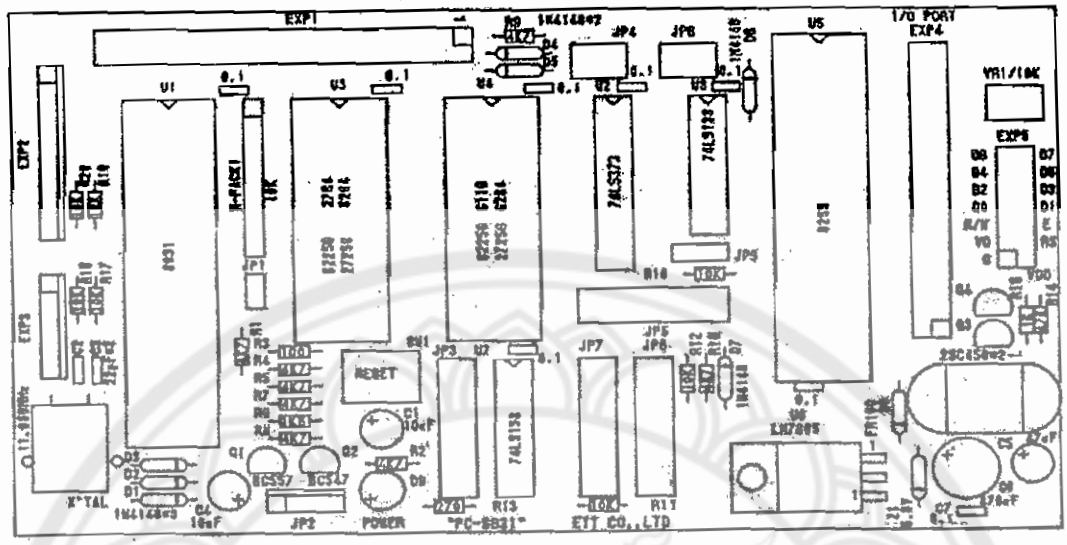
JP8 เลือกตำแหน่งเริ่มต้นและขนาดของหน่วยความจำ U4

JP9 เลือกว่าจะอนุญาตให้มีการใช้ I/O พอร์ท (8255) ภายนอกอีกหรือไม่ถ้ามีพอร์ทภายนอก U4 จะมีขนาดสูงสุดได้ถึง 32 kB

ตัวอย่างการเซตจัมพ์เปอร์และขาคอนเนคเตอร์ของพอร์ท I/O



รูปที่ 2.5 ตัวอย่างการเซตจัมพ์เปอร์และขาคอนเนคเตอร์ของพอร์ท I/O



รูปที่ 2.6 บอร์ดไมโครคอนโทรลเลอร์ CP-SB31

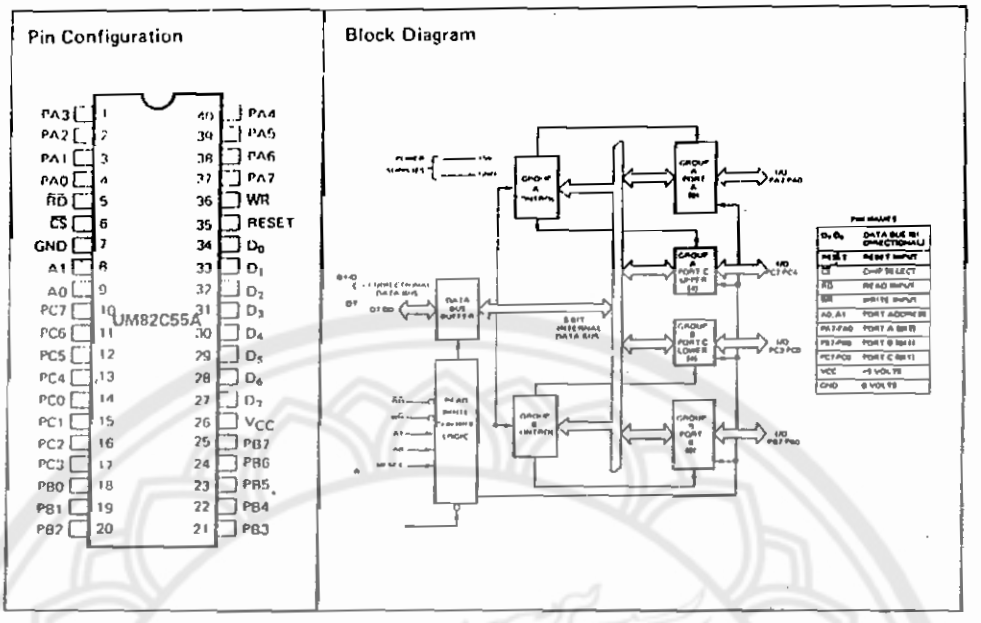
UM82C55A 8x3 บิต Input/Output (I/O)

เป็น CMOS ที่สามารถโปรแกรมการเชื่อมต่อกับภายนอกได้

ลักษณะสำคัญ (Features)

- พิน (Pin) ต่อเข้ากันกับ NMOS 8255A ได้
- มีพิน I/O 24 พิน ที่สามารถโปรแกรมได้
- เต็มไปด้วย TTL ที่เข้ากันได้
- วงจรบัสของการเชื่อมต่อทุก ๆ I/O มีตัวต้านทานยกระดับแรงดันแยกทุกวงจร
- มีความเร็วสูงไม่ต้องรอการทำงานแต่ละสถานะ (State) มีความเร็วของ 80C86 8 MHz
- มีปุ่มเซต/รีเซตกดได้โดยตรง
- มีการเพิ่มการอ่านคำที่ใช้ในการควบคุมได้ดีขึ้น
- **ใช้แหล่งจ่ายไฟ 5 V เพียงแหล่งเดียว**
- **ทุก ๆ จุดเชื่อมต่อ I/O มีเอาท์พุตขับโดยกระแสเพียง 2.5 mA**
- **กำลังงานที่ใช้ในการคงสถานะ (Standby Power) ต่ำมากเพียง 10 μA**

(-Iccsb = 10 μA)



รูปที่ 2.7 ขาบล็อกไดอะแกรมของ IC UM82C55A

การทำงาน (Modes) ความสามารถที่สูงของ UM82C55A จึงทำให้มันสามารถใช้ได้กับ 8086, 8084, 8051 (Single Board)

วงจร CMOS ของ UM82C55A ถูกออกแบบมาให้ใช้พลังงานที่ต่ำ TTL ที่ใช้ จะใช้แรงดัน V_{IH} = 2.0 โวลท์ ซึ่งมีย่านอุณหภูมิการใช้งานที่เหนือกว่า TTL ที่ใช้อุตสาหกรรม และในวงจรบัสมีการใส่ค่าความต้านทานยกระดับแรงดันไว้ด้วย

ลักษณะของพิน (Pin Description)

Data Bus Buffer

เป็น 8 บิตบัฟเฟอร์สองทิศทางการส่งข้อมูล 3 สถานะที่ UM82C55A ใช้ในการเชื่อมต่อของระบบการส่งข้อมูล ข้อมูลจะถูกส่งหรือถูกรับโดยการกระทำของบัฟเฟอร์ผ่านทางโครงสร้างอินพุตหรือโครงสร้างเอาท์พุตโดยซีพียู ข้อมูลส่วนของคำสั่งควบคุมและข้อมูลอื่น ๆ ก็จะผ่านทางบัสบัฟเฟอร์นี้ด้วย

อ่าน/เขียนและลอจิกที่ใช้ในการควบคุม (Read/Write and Control Logic)

ฟังก์ชันในกล่องนี้เป็นตัวควบคุมการส่งข้อมูลภายในและภายนอกโดยรับคำสั่งจากซีพียู

Chip Select (CS)

จะทำงานเมื่อรับอินพุตเป็น "Low" เป็นตัวสั่งการทำงานการสื่อสารระหว่าง UM82C55A กับซีพียู

Read (RD)

จะทำงานเมื่อได้รับอินพุตเป็น "Low" เป็นตัวสั่งให้ข้อมูลใน UM82C55A ไหลออกสู่บัสข้อมูล (Data Bus) และไปสั่งให้ซีพียูอ่านข้อมูลจากบันทึกข้อมูลเข้าไป

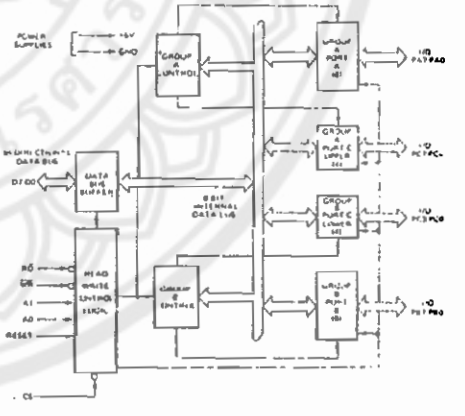
Write (WR)

จะทำงานเมื่อได้รับอินพุตเป็น "Low" เป็นตัวสั่งให้ซีพียูเขียนข้อมูลลงใน UM82C55A

พอร์ทเลือก 0 และพอร์ทเลือก 1 (A0 and A1)

นี่เป็นสัญญาณอินพุตในการเชื่อมต่อระหว่างอินพุตของ RD และ WR ซึ่งจะเป็นตัวควบคุมการเลือก 1 พอร์ทจาก 3 พอร์ท หรือเป็นตัวเก็บคำสั่งควบคุม ปกติพอร์ทนี้จะถูกควบคุมโดยบิตที่มีความสำคัญน้อยที่สุด (The Least Significant Bits) ของแอสเดรสบัส (A_0 and A_1)

A	A_1	RD	WR	CS	Input Operation (Read)
0	0	0	1	0	Port A → Data Bus
0	1	0	1	0	Port B → Data Bus
1	0	0	1	0	Port C → Data Bus
1	1	0	1	0	Control Word → Data Bus
					Output Operation (Write)
0	0	1	0	0	Data Bus → Port A
0	1	1	0	0	Data Bus → Port B
1	0	1	0	0	Data Bus → Port C
1	1	1	0	0	Data Bus → Control
					Disable Function
x	x	x	x	1	Data Bus → 3-State
x	x	1	1	0	Data Bus → 3-State



รูปที่ 2.8 ตารางการใช้งานพื้นฐานและบล๊อกไดอะแกรมของ UM82C55A

Reset

จะทำงานเมื่อรับอินพุตเป็น "High" จะทำงานโดยการส่งข้อมูลไปล้างหน่วยความจำควบคุมและพอร์ททั้งหมด (A, B, C) จะถูกเซตโดยขึ้นกับโหมดการทำงานของอินพุต "Bus Hold" เป็นอุปกรณ์ภายใน UM82C55A จะทำให้ I/O พอร์ทถือสัญญาณเป็น ลอจิก "1" โดยจะใช้กระแสใน การถือสัญญาณ 300 μA

Group A and Group B Controls

พอร์ทภายนอกอื่นๆจะถูกเขียนโปรแกรมโดยซอฟท์แวร์ระบบหรืออีกนัยหนึ่งซีพียู "Out put" จะส่งประโยคควบคุมไปยัง UM82C55A ประโยคควบคุม (Control Word) จะประกอบไปด้วยข่าวสารต่างๆ ตัวอย่างเช่น "Mode" , "Bit set" , "Bit Reset" นี่คือค่าเริ่มต้นของลักษณะภายนอกของ UM82C55A

นอกเหนือจาก กล่องควบคุม (Control Blocks) กลุ่ม A และ กลุ่ม B (Group A and Group B) จะได้รับคำสั่งอ่าน/เขียน (Read/Write) ซึ่งเป็นลอจิกควบคุมและจะได้รับประโยคควบคุมจากบัสข้อมูลภายใน และข้อมูลก็จะไหลไปยังพอร์ทต่างๆที่เกี่ยวข้องกับการทำงานนี้

กล่องควบคุม A - พอร์ท A และพอร์ท C (C_7-C_4)

กล่องควบคุม B - พอร์ท B และพอร์ท C (C_3-C_0)

หน่วยความจำประโยคควบคุม สามารถเขียนและอ่านได้ทั้ง 2 อย่าง โดยจะแสดงในตารางการใช้งานพื้นฐาน ในรูปที่ 2.9 รูปแบบการใช้งานของประโยคควบคุมสำหรับทั้งการอ่านและการเขียนข้อมูลต่าง ๆ เมื่อประโยคควบคุมทำการสั่งการอ่าน บิตที่ D_7 จะเป็น ลอจิก 1 ตลอดเวลา

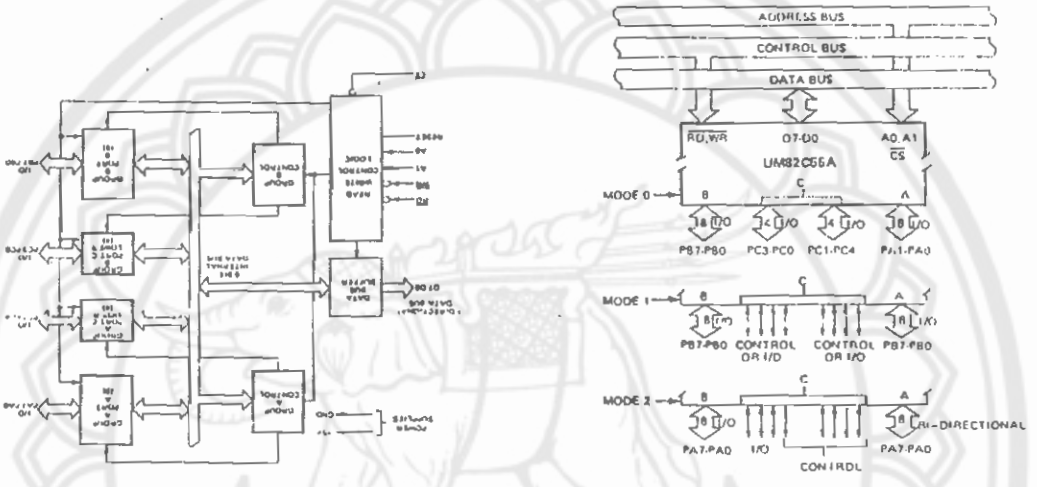
พอร์ท A, B และ C

ใน UM82C55A ประกอบด้วย 8 บิต 3 พอร์ท (A, B และ C) โดยจะใช้งานได้หลากหลายโดยการสั่งการของซอฟท์แวร์ระบบ แต่มีคุณสมบัติพิเศษของตัวมันเอง หรือ การใช้งานของบุคคลในการจัดการเกี่ยวกับพลังงานและการปรับแต่งของ UM82C55A

พอร์ท A เป็นพอร์ทข้อมูลเอาท์พุต 8 บิต แบบแลตช์ / บัฟเฟอร์(Latch / Buffer) 1 พอร์ท และเป็นพอร์ทข้อมูลอินพุตแบบ 8 บิต แลตช์ (Latch) อีก 1 พอร์ททั้ง 2 พอร์ท มีอุปกรณ์ในการถือสัญญาณบัสโดยเป็น "Pull-Up" และ "Pull-Down"

พอร์ท B เป็นพอร์ทข้อมูล อินพุต / เอาท์พุต (1 / O) แลตช์ / บัฟเฟอร์ แบบ 8 บิต และเป็นพอร์ทข้อมูลอินพุต 8 บิต แบบบัฟเฟอร์ (Buffer)

<u>พอร์ท C</u> เป็นพอร์ทข้อมูลเอาท์พุต 8 บิต แบบแลตช์ / บัฟเฟอร์ 8 บิต 1 พอร์ทและเป็นพอร์ทข้อมูลอินพุตแบบบัฟเฟอร์อีก 1 พอร์ท พอร์ท C นี้สามารถแบ่งเป็นพอร์ท 4 บิต 2 พอร์ทได้ โดยอยู่ภายใต้การควบคุมของโหมดการทำงานและยังมีพอร์ทอื่นอีก 1 พอร์ทที่เป็น 4 บิต พอร์ทเป็นแบบแลตช์และมันสามารถใช้ในการควบคุมสัญญาณเอาท์พุตและสัญญาณอินพุต ในการเชื่อมต่อกับพอร์ท A และพอร์ท B ด้วย



รูปที่ 2.9 รูปกลุ่ม A และกลุ่ม B และข้อจำกัดของโหมดของ UM82C55A

<u>คำอธิบายการทำงาน</u>

<u>การเลือกโหมด</u> **(Mode Selection)**

มีอยู่ 3 โหมด ขึ้นอยู่กับผู้ใช้จะเลือก โดยเลือกจากซอฟท์แวร์ระบบ

Mode 0 - อินพุต / เอาท์พุต พื้นฐาน

Mode 1 - อินพุต / เอาท์พุต ไฟฟ้า (Strobed)

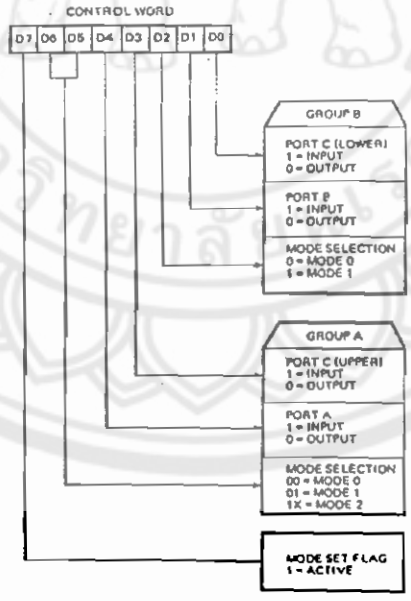
Mode 2 - ระบบบัส 2 ทิศทาง

เมื่อรีเซตให้อินพุตเป็น "High" ทุกพอร์ทจะเซตตัวเองให้อยู่ในอินพุตโหมด และทั้ง 24 พอร์ทจะถือสัญญาณของอุปกรณ์บัสภายใน หลังจากการรีเซตถูกย้ายแล้ว UM82C55A จะยังคงเป็นอินพุตโหมดอยู่ เมื่อไม่มีความต้องการสั่งให้มีการเปลี่ยนค่าเริ่มต้น การกำจัดสถานการณ์อย่างนี้ทำได้โดย ต้องมีตัวต้านทาน "Pull Up" หรือ "Pull Down" ในการออกแบบ CM05 ทุกตัว ระหว่างกระบวนการของโปรแกรมระบบ โหมดการทำงานจะถูกเลือกเพียงโครงสร้างเอาท์พุตโหมดเดียว

สำหรับ UM82C55A เป็นอุปกรณ์ภายนอก ทำหน้าที่หลากหลายกับซอฟท์แวร์ และทำหน้าที่บำรุงรักษาระบบเป็นประจำ

โหมดการทำงานของ พอร์ท A และ พอร์ท B สามารถแยกได้อย่างแน่นอน แต่การทำงานของ พอร์ท C ถูกกำหนดโดย ความต้องการของ พอร์ท A และ พอร์ท B หน่วยความจำเอาท์พุตทั้งหมด เป็นการรวมของฟลิปฟลอป จะรีเซตก็ต่อเมื่อโหมดถูกเปลี่ยนแปลงและโหมดต้องเหมาะสมกัน ข้อจำกัดของฟังก์ชันของ ทุก ๆโครงสร้าง I / O โดยกลุ่ม B สามารถถูกโปรแกรมในโหมด D ในการควบคุม การแสดงผลทางจอภาพหรือการแสดงผลการคำนวณทางเครื่องแสดงผล กลุ่ม A สามารถถูกโปรแกรมในโหมด 1 โดยการรับข้อมูลจากคีย์บอร์ดหรือการบันทึกข้อมูลด้วยเทป บนการขับโดยจังหวะพื้นฐาน(Base Interrupt)

ข้อจำกัดของโหมดและปัญหาการแยกโหมด ที่ควรคำนึงถึงเป็นประการแรก แต่ต้องขึ้นอยู่กับผู้ใช้และความสมบูรณ์ของอุปกรณ์ที่มาประกอบกัน การออกแบบของ UM82C55A ควรมีการทดสอบประสิทธิภาพบน PC ก่อน และข้อจำกัดในการควบคุม กับความสามารถของ PC และความสมบูรณ์ของฟังก์ชันต่าง ๆ ที่เหมาะสมกันทุก ๆ อุปกรณ์ที่อยู่ภายนอกและอุปกรณ์ที่อยู่ภายใน ดังนั้นการออกแบบควรให้เกิดประโยชน์สูงสุดกับผู้ใช้และออกแบบให้เหมาะสมกับอุปกรณ์รอบข้างอีกด้วย



รูปที่ 2.10 รูปแบบข้อจำกัดของโหมด

การเซตบิตเดี่ยว / ลักษณะสำคัญของการรีเซต (Single Bit Set / Reset Feature)

ทุกๆ บิตของพอร์ท C สามารถเซตหรือรีเซตโดยให้โครงสร้างของเอาท์พุต ซึ่งเป็นลักษณะสำคัญของการควบคุมโดยคำสั่งตามความต้องการของผู้ใช้ที่เหมาะสมกับงานต่าง ๆ

เมื่อพอร์ท C ถูกใช้ในการควบคุม พอร์ท A และพอร์ท B ในที่นี้บิตสามารถเซตหรือรีเซต ได้โดยผู้ใช้ต้องทำเองถ้ามีพอร์ทเอาท์พุตข้อมูล

เมื่อ UM82C55A ถูกโปรแกรมโดยผู้ใช้ในโหมด 1 หรือโหมด 2 สัญญาณจะถูกควบคุม เมื่อเราส่งความต้องการที่จะอินเทอร์รัปท์ (Interrupt) ไปยังอินพุตของซีพียู สัญญาณที่ต้องการจะอินเทอร์รัปท์ จะถูกกำหนดโดยการเซตหรือรีเซตความสัมพันธ์ของ INTE ฟลิปฟลอปที่ใช้ ฟังก์ชันการเซตหรือรีเซตของพอร์ท C

ที่ฟังก์ชันนี้อนุญาตให้ผู้เขียนโปรแกรมสั่งใช้งาน (Enable) หรือสั่งให้หยุดทำงาน (Disable) การอินเทอร์รัปท์ของซีพียูโดยอุปกรณ์ I/O พิเศษกับผลของอุปกรณ์ที่ใช้ในโครงสร้างการอินเทอร์รัปท์

ข้อจำกัดของไอเอ็นทีอีฟลิปฟลอป INTE – Flip – Flop Definition

- การเซตบิต (Bit – Set) – INTE ถูกเซต – อินเทอร์รัปท์ถูกสั่งใช้งาน (Interrupt Enable)
- การรีเซตบิต (Bit – Reset) – INTE ถูกรีเซต – อินเทอร์รัปท์ถูกสั่งไม่ให้ทำงาน (Interrupt Disable)

หมายเหตุ

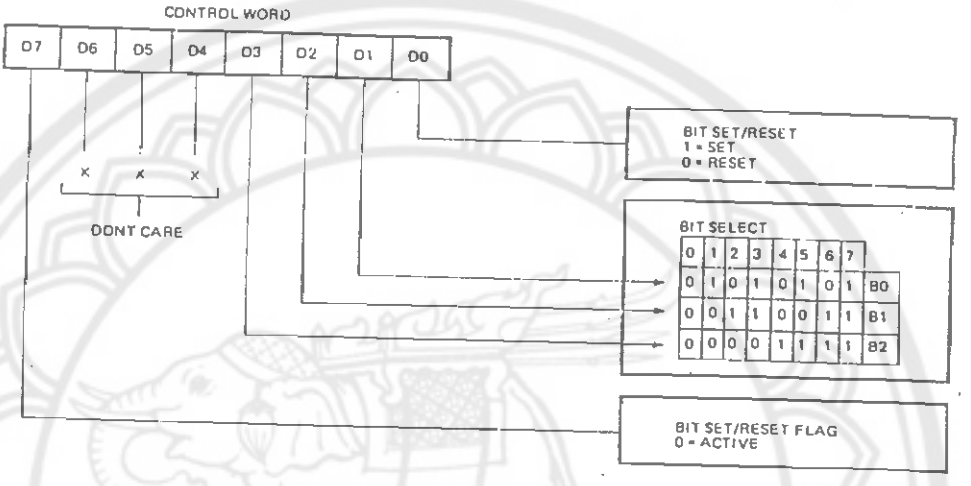
ทุกฟลิปฟลอปจะรีเซตโดยอัตโนมัติระหว่างการเลือกโหมดและอุปกรณ์รีเซต

โหมดใช้งาน (Operating Modes)

โหมด 0 (อินพุต / เอาท์พุตพื้นฐาน) นี่คือฟังก์ชันภายนอกที่ให้ตัวอย่างของอินพุตและเอาท์พุตโดยผู้ใช้ สำหรับพอร์ททั้ง 3 พอร์ท ข้อมูลจะถูกเขียนหรืออ่านจากพอร์ทเฉพาะเท่านั้น

ข้อจำกัดของฟังก์ชันพื้นฐานของโหมด 0 (Mode 0 Basic Functional Definitions)

- **มี 8 บิตพอร์ท 2 พอร์ท และ 4 บิตพอร์ท 2 พอร์ท**
- **ทุกพอร์ทสามารถเป็นทั้งอินพุตและเอาท์พุต**
- **เอาท์พุตเป็นแบบแลตช์ (Latch)**
- **อินพุตไม่เป็นแลตช์**
- **16 อินพุต / เอาท์พุต ที่แตกต่างตามองค์ประกอบภายนอกที่ใช้งาน**



รูปที่ 2.11 รูปแบบการเซตและรีเซตบิต

2.4 อีที – อีเอ็มพลัส เครื่องจำลองอีพรอม (ET – EM Plus EPROM Emulator)

อีที – อีเอ็มพลัส (ET - EM Plus) คือ บอร์ดวงจรที่ถูกสร้างขึ้นมาเพื่อใช้แทนส่วนของตัวอีพรอม (EPROM) หรือแรม (RAM) ซึ่งจะมีความง่ายและสะดวกกว่าการใช้ อีพรอมหรือแรมจริงๆ ทำให้เหมาะกับการพัฒนาระบบไมโครต่างๆ หรือในการแก้ไขเปลี่ยนแปลงโปรแกรมการทำงานของเครื่องเป็นไปโดยง่าย

เราสามารถเขียนข้อมูลเข้าไปในส่วนของ อีที – อีเอ็มพลัส ได้โดยตรงซึ่งต่างกับการที่เราต้องนำ อีพรอม นั้นออกมาเขียนเปลี่ยนแปลงแล้วจึงนำกลับเข้าไปใส่ในวงจรใหม่

<u>งานที่นำไปใช้</u>

อีที – อีเอ็มพลัส สามารถนำไปใช้งานได้หลายรูปแบบขึ้นอยู่กับผู้ใช้จะนำไปใช้งาน ตัวอย่างเช่น

1. เครื่องช่วยควบคุมรอม (ROM Monitor) ใช้ในการเขียนโปรแกรมในส่วนของโปรแกรมการควบคุม (Monitor Program) ของเครื่องไมโครต่างๆ เพราะในการเริ่มต้นพัฒนาระบบไมโครอะไรก็ตาม ไมโครนั้นยังไม่มีส่วนของโปรแกรมในการเขียนอ่านข้อมูลจากภายนอก ซึ่งถ้านำ อีที – อีเอ็มพลัส มาใช้ก็จะสะดวกกว่าใช้ อีพรอม เป็นเครื่องช่วยควบคุม (Monitor) จริง ๆ มาก

2. คุณลักษณะเป็นแหล่งกำเนิด (Charactor Generator) เราสามารถใช้ อีที – อีเอ็มพลัส ในการทำเป็น Charactor Generator ในการเปลี่ยนแปลงหรือเพิ่มเติมตัวอักษรเข้าไป เช่น ทำภาษาไทยในเครื่องพิมพ์ ซึ่งเราสามารถทดสอบเขียนแล้วทดสอบทำงานได้โดยตรง

3. ไบออส (Bios) เป็นรอมไบออส (ROM Bios) ในเครื่อง PC ใช้ทดสอบหรือดีบัก (Debug) ตัวโปรแกรมไบออส ของเครื่องหรือใช้ทดสอบเขียนโปรแกรมตรวจสอบการทำงานเป็นส่วน ๆ ของเครื่องได้ ฯลฯ

<u>คุณสมบัติของ อีที – อีเอ็มพลัส</u>

1. สามารถส่งผ่านข้อมูลจากเครื่อง PC ได้ทางพอร์ทเครื่องพิมพ์ (Printer Port) ที่ให้ง่ายในการหาเครื่อง PC มาใช้ เพราะเครื่อง PC ต่างๆนั้นจะมีส่วนของพอร์ทเครื่องพิมพ์ อยู่แล้ว

2. ใช้กับไฟล์ (File) ได้หลายรูปแบบ เช่น ไฟล์ฐานสอง (Binary File), ไฟล์ฐานสิบหกของอินเทล (Intel Hex File), ไฟล์ของโมโตโลล่า (Motolora File ;S Format)

3. สามารถเปลี่ยนแปลงข้อมูลใน อีที – อีเอ็มพลัส ทั้งหมดหรือบางไบท์ (Byte) ก็ได้โดยไม่จำเป็นต้องโหลดไฟล์ใหม่หมด

4. สามารถตั้งการเริ่มต้น (Offset) คือ ค่าตำแหน่งที่อยู่ของข้อมูลจากไฟล์ได้โดยตรงอย่างอิสระ

5. สามารถเลือกรูปแบบการส่งข้อมูลจากไฟล์ได้หลายแบบ เช่น ส่งข้อมูลทุกไบท์, ส่งข้อมูลเฉพาะไบท์คู่, ส่งข้อมูลเฉพาะไบท์คี่ ซึ่งทำให้เราสามารถนำ อีที – อีเอ็มพลัส 2 ตัวมาต่อกับพอร์ทเครื่องพิมพ์ 2 ชุด ใช้พัฒนาระบบเครื่องที่จำเป็นต้องใช้ 16 บิตข้อมูล (16 Bit Data)หรือที่ต้องใช้ อีพรอม 2 ตัว ของเครื่อง 16 บิตที่ใช้เป็นไบออส ของเครื่อง

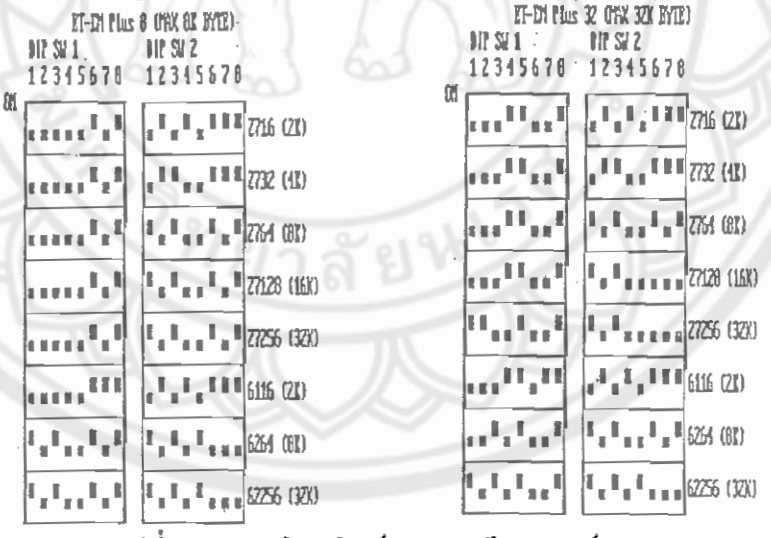
6. สามารถต่อกับพอร์ทเครื่องพิมพ์ได้ทั้ง LPT1, LPT2 หรือโมโนการ์ด (Mono Card) ได้โดยเป็นอิสระต่อกัน

7. มีความเร็วในการส่งผ่านข้อมูลสูงมาก

8. สามารถรีเซตซีพียู (Reset CPU) ได้เมื่อโหลดข้อมูล (Load Data) แล้วโดยมีแบบ รีเซตโลว์ (Reset Low) และรีเซตไฮ (Reset High)

<u>การนำไปใช้งาน</u>

1. ปรับดิพสวิทช์ (Dip Switch) ของ อีที – อีเอ็มพลัส ให้ตรงกับส่วนของวงจรที่จะนำไปใช้งาน ดังรูปที่ 2.12 (หรือจะดูจาก ET EM Help ก็ได้)



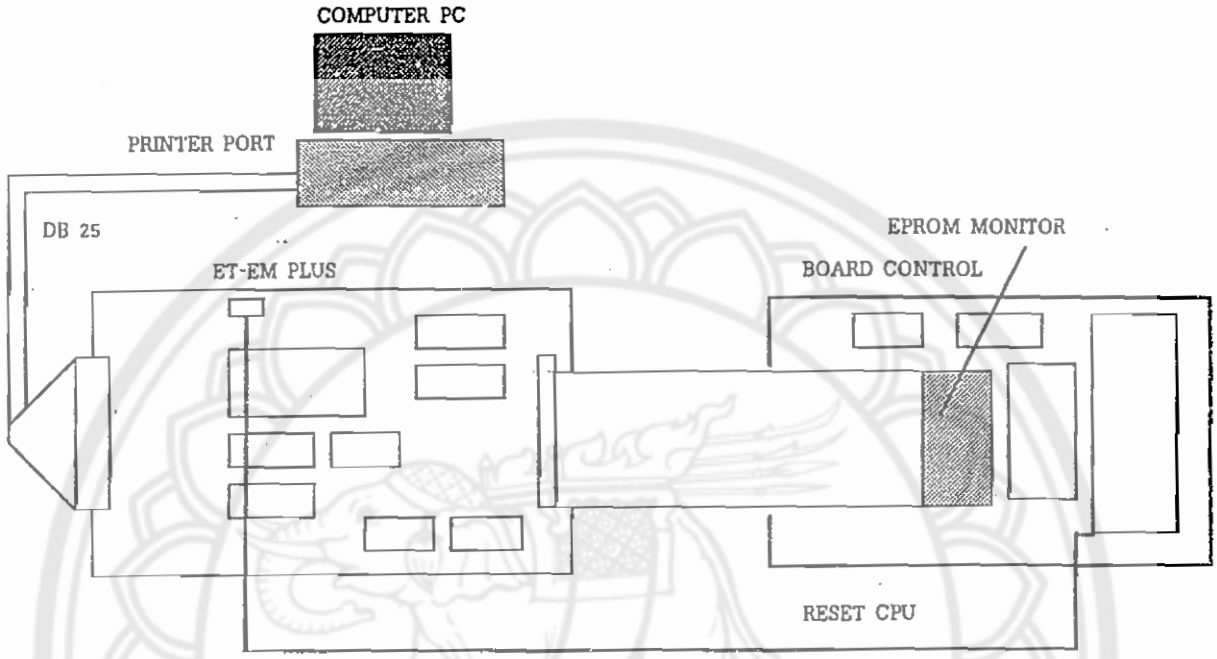
รูปที่ 2.12 การดิพสวิทช์ของรอมอีมูเลเตอร์

- ปรับ Jumper Vcc (O, I) ว่าต้องการให้บอร์ดอีเอ็ม (Board EM) ใช้แหล่งจ่ายไฟ +5VDC จากที่ใด

I = จาก Dip Jumper 28 Pin

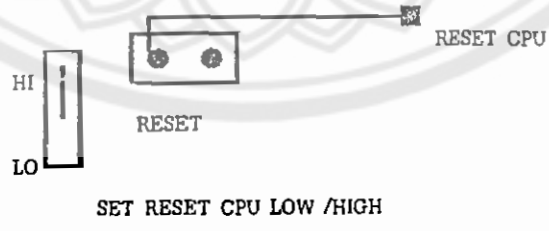
O = จาก Connector Power

- ปรับ Jumper OE ว่าต้องการต่อขา OE (Dip 28 Pin EM) ลงกราวด์ (GND) หรือต่อเข้ากับระบบ



รูปที่ 2.13 การเชื่อมของรอมอีมูเลเตอร์กับบอร์ด PC-SB31

2. ต่ออีที – อีเอ็มพลัส เข้ากับทางพอร์ทเครื่องพิมพ์ของเครื่อง PC โดยใช้สาย DB 25 พินที่มีให้



รูปที่ 2.14 การเซตและการรีเซต CPU (Low, High)

3. ใช้โปรแกรมในแผ่นดิสก์ (Disk) ที่ให้โดยเราสามารถเลือกใช้ได้ 3 โปรแกรมตรงกับพอร์ทเครื่องพิมพ์ที่เรามีอยู่กับเครื่อง เช่น

EM1P: ใช้กับการแสดงการเคลือบโลหะเดี่ยว (Monochrome Display) และตัวแปลงเครื่องพิมพ์ (Printer Adapter)

EM2P: ใช้กับตัวแปลงเครื่องพิมพ์ LPT1

EM3P: ใช้กับตัวแปลงเครื่องพิมพ์ LPT2

<u>รูปแบบการคีย์โปรแกรม ET – EM Plus</u>

EM1P [/ , /p] [SOURCEFILE] [H , O , S] [E , O]

[/] = เข้าสู่โหมดแก้ไข (Edit Mode) ในการต้องการคีย์ข้อมูลเป็นไบท์ หรือใช้ไฟล์ข้อมูล

เช่น EM1P/

[/P] = เป็นการตั้งค่าเริ่มต้นของพอร์ทเครื่องพิมพ์ (Initialize Printer Port) เพื่อนำพอร์ทเครื่องพิมพ์นั้นกลับไปให้กับเครื่องพิมพ์ และเป็นการตรวจว่าเครื่องมีการ์ดเครื่องพิมพ์ (Printer Card) อะไรต่ออยู่

[SOURCEFILE] = ชื่อไฟล์ที่เราจะโหลดข้อมูลเข้าอีที – อีเอ็มพลัส

เช่น EM1P TEST.HEX H

[H] = ใช้กับไฟล์ที่เป็นไฟล์ฐานสิบหกของอินเทล (Intel Hex File)

[O] = ใช้กับไฟล์ที่เป็นไฟล์ฐานสอง (Binary File)

[S] = ใช้กับไฟล์ที่เป็นไฟล์ของโมโตโลล่า (Motolora File ; S File)

เช่น ส่งข้อมูลแบบฐานสอง (Binary)

EM1P TEST.OBJ O

[OFFEST] = ค่าตำแหน่งแอดเดรส (Address) ที่เราจะโหลดหรือบวกเข้าไปในกรณีใช้กับ ไฟล์ฐานสิบหกของอินเทลโดยคีย์เป็นเลขฐาน 16 ขนาด 4 หลัก

เช่น ส่งข้อมูลแบบอินเทล (Intel) บวก และตั้งค่าเริ่มต้น (Offset) อีก 100H

EM1P TEST.HEX H 0100

[E] = ใช้ส่งข้อมูลแบบไบท์คู่

[O] = ใช้ส่งข้อมูลแบบไบท์คี่

เช่น ส่งข้อมูลแบบอินเทลบวก และตั้งค่าเริ่มต้นเข้าไปอีก 200H แบบไบท์คู่

EM1P TEST.HEX H 0200 0

ในขณะที่ส่งข้อมูลจะทำให้แอลอีดี (LED) บนบอร์ด (Board) ติดสลับกันทำงานและเมื่อส่งข้อมูลเรียบร้อยแล้วจะดัดตัวเองกลับไปยังบอร์ดที่ต่อโดยอัตโนมัติโดยแอลอีดี PC จะติดเมื่อบอร์ด

ทำงานรับข้อมูลและแอลอีดี ROM จะดับและเมื่อ PC ส่งข้อมูลเรียบร้อยแล้วแอลอีดี PC จะดับแอลอีดี ROM จะติด

*<u>ข้อควรระวัง</u> *

*ในบางครั้งไฟล์ฐานสิบหกของอินเทล หรือไฟล์ฐานสองที่ได้อาจจะโตกว่าความจุในแรมของ EM เช่นมากกว่า 8 กิโลไบท์จะทำให้ข้อมูลเกินกลับมาทับข้อมูลเดิมที่แอดเดรส 0000H ได้

*ในเครื่อง PC ความเร็วสูงๆ อาจเซตโหมดของเครื่องพิมพ์ (Set Mode Printer) ได้หลายรูปแบบในการเซตไบออส (Bios Setup) ในกรณีใช้ อีที – อีเอ็มพลัส ให้เซตไบออสโหมดของเครื่องพิมพ์ (Setup Bios Printer Mode) เป็นแบบ ECP Mode ในกรณีต้องการดูเมนูช่วย (Help Menu) เราก็สามารถคีย์ได้โดยการคีย์ เช่น EM1P โดยจะมีคำอธิบายการทำและการปรับดิพสวิทช์ขึ้นแสดงบนจอ

<u>เมนูแก้ไข (Edit Menu)</u>

เราสามารถเข้าสู่โหมดแก้ไขได้โดยการคีย์ เช่น EM2P โดยมีการทำงาน 2 ลักษณะในโหมด (Mode) ดังนี้

1. การนำข้อมูลเข้า (Enter Data) เราสามารถคีย์ข้อมูลเข้าไปในตำแหน่งของอีที – อีเอ็มพลัส (ET – EM Plus) ได้โดยตรงโดยการคีย์ เช่น ET – EM > E 0010
โดยค่าแอดเดรส ต้องเป็นค่า 4 หลัก ฐาน 16 แล้ว ให้เราคีย์ข้อมูลเข้าไปได้โดยคีย์เลขจำนวน 2 หลัก ฐาน 16 โดยจะรับคีย์ครบ 2 หลัก แล้วจะเพิ่มค่าแอดเดรสขึ้นเป็นอัตโนมัติ และถ้าไม่ต้องการเปลี่ยนค่าข้อมูลในตำแหน่งนั้นๆ ให้กดคีย์ " SPACES BAR " เข้าไป และถ้าต้องการออกจากการแก้ไขของการนำเข้า (Enter) ให้คีย์ "."

2. ไฟล์ข้อมูล (File Data) เราสามารถเปลี่ยนข้อมูลในอีที – อีเอ็มพลัส ให้เป็นช่วง เช่น ต้องการให้ตำแหน่ง 0100H ถึง 01FFH มีข้อมูลเป็น A5H เราทำได้โดยคีย์

ET – EM PLUS > F 0100 01FF A5

และถ้าต้องการออกจากแก้ไข (Edit) ให้คีย์ "."

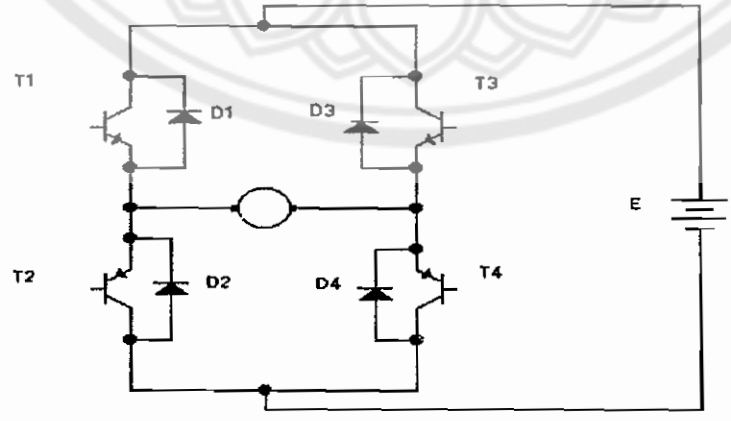
2.5 เซอร์โวแอมปลิไฟร์ (Servo Amplifier)

เซอร์โวแอมปลิไฟร์เป็นการเรียกวงจรเพาเวอร์อิเล็กทรอนิกส์ที่ออกแบบขึ้นเพื่อใช้ขับเซอร์โวมอเตอร์ โดยการควบคุมความเร็วหรือตำแหน่ง ถูกนำไปใช้มากในอุตสาหกรรมหุ่นยนต์ และอุปกรณ์ควบคุมเชิงเลข (Numerically Controlled Equipment) เซอร์โวแอมปลิไฟร์แบ่งออกได้เป็น 2 ชนิดตามวิธีใช้ขับอุปกรณ์โซลิดสเตต (Solid State) ชนิดแรกเรียกว่า เซอร์โวแอมปลิไฟร์ชนิดเชิงเส้น (Lenear Servo Amplifier) ซึ่งขับให้ทรานซิสเตอร์ทำงานอยู่ในย่านที่เป็นเชิงเส้น อีกชนิดหนึ่งเรียกว่า เซอร์โวแอมปลิไฟร์แบบพัลส์วิดธ์มอดูเลชั่น (PWM) ซึ่งจะขับทรานซิสเตอร์หรือมอสเฟตให้ทำงานแบบสวิตช์ คือ ทำงาน (ON) และไม่ทำงาน (OFF) เหมือนกับใช้ในคอนเวอร์เตอร์แต่ในที่นี้จะกล่าวถึงการทำงานแบบ เซอร์โวแอมปลิไฟร์แบบพัลส์วิดธ์มอดูเลชั่น

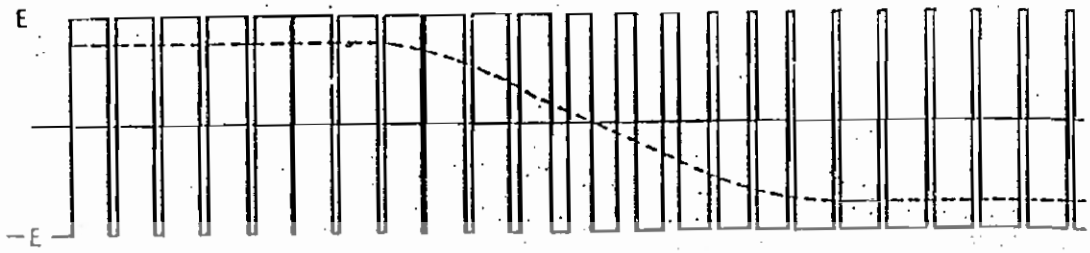
<u>เซอร์โวแอมปลิไฟร์ชนิด PWM (Servo Amplifier Type PWM)</u>

เพื่อเป็นการลดความสูญเสียทางความร้อนในทรานซิสเตอร์และเพิ่มประสิทธิภาพให้กับเซอร์โวแอมปลิไฟร์ และเนื่องจากเซอร์โวแอมปลิไฟร์ชนิดเชิงเส้นมีกำลังงานสูญเสียมากกว่าเซอร์โวแอมปลิไฟร์แบบพัลส์วิดธ์มอดูเลชั่นจึงมีการนำเอาวิธีการพัลส์วิดธ์มอดูเลชั่นหรือ PWM มาใช้กับเซอร์โวแอมปลิไฟร์ โดยให้ทรานซิสเตอร์ทำงานเพียงสองสภาวะคือ สภาวะทำงาน และไม่ทำงาน หลักการทำงานของเซอร์โวแอมปลิไฟร์ชนิด PWM มีกันอยู่ 3 ชนิดมีด้วยกัน ดังนี้

1. เซอร์โวแอมปลิไฟร์ชนิด PWM แบบควบคุมแรงดัน
2. เซอร์โวแอมปลิไฟร์ชนิด PWM แบบควบคุมกระแส
3. เซอร์โวแอมปลิไฟร์ชนิด PWM แบบบริดจ์



รูปที่ 2.15 วงจรพื้นฐานของเซอร์โวแอมปลิไฟร์ชนิด PWM แบบบริดจ์



รูปที่ 2.16 รูปคลื่น PWM ที่สวิงระหว่าง +E ถึง –E เมื่อใช้ทรานซิสเตอร์ทั้งสี่ตัวทำงานแบบบริดจ์

รูปที่ 2.17 รูปคลื่น PWM ที่สวิงระหว่าง +E ถึง ศูนย์ หรือ –E ถึง ศูนย์ เมื่อใช้ทรานซิสเตอร์สองตัวทำงานแบบบริดจ์

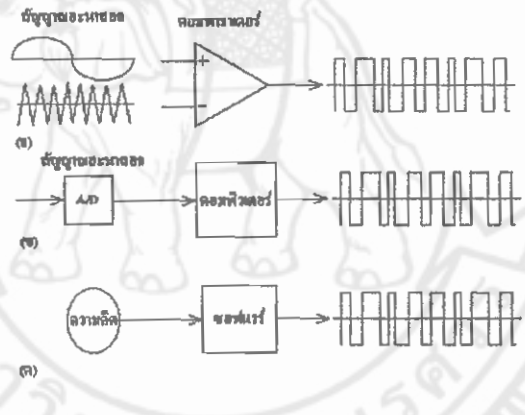
แต่ตอนนี้จะกล่าวถึงหลักการทำงานเซอร์โวแอมปลิไฟร์ชนิด PWM แบบบริดจ์ รูปที่ 2.15 เป็นวงจรเบื้องต้นของเซอร์โวแอมปลิไฟร์ชนิด PWM แบบบริดจ์ ใช้ทรานซิสเตอร์ทั้งหมดสี่ตัวควบคุมแรงดันที่จ่ายให้ดีซีมอเตอร์ การใช้ทรานซิสเตอร์ทั้งสี่ตัวให้ทำงานมีอยู่สองวิธีในเบื้องต้นคือ วิธีแรกใช้ทรานซิสเตอร์ทุกตัวเป็นสวิตช์ จะได้แรงดันตามรูปที่ 2.16 ซึ่งแรงดันชั่วขณะจะสวิงอยู่ระหว่าง +E ถึง -E และค่าแรงดันเฉลี่ยแสดงอยู่ในรูปเดียวกันตามแนวเส้นประ อีกวิธีหนึ่งจะสร้างรูปคลื่นดังในรูปที่ 2.17

<u>การพิจารณาวิธีการพัลส์วิดธ์มอดูเลชั่น</u>

พัลส์วิดธ์มอดูเลชั่นหรือ PWM คือวิธีการทำให้สัญญาณอะนาลอกเปลี่ยนไปอยู่ในรูปของสัญญาณพัลส์ที่เปลี่ยนแปลงช่วงเวลาในหนึ่งคาบเวลาได้ การพิจารณาเราจะศึกษาถึงวิธีการสร้างและการดีมอดูเลชั่นสัญญาณ PWM วิธีการสร้างสัญญาณ PWM ทำได้สามอย่างคือ ใช้วิธีดิจิตอล ไฮบริดจ์ และอะนาลอก จากรูปที่ 2.18(ก) รูปคลื่นสามเหลี่ยมจะถูกป้อนเข้าที่ขาหนึ่งของคอมพาราเตอร์ส่วนขาที่เหลือจะป้อนสัญญาณอะนาลอกเข้าไป ก็จะได้สัญญาณ PWM ที่เอาท์พุตของคอมพาราเตอร์โดยสัญญาณอะนาลอกเป็นสัญญาณที่จะถูกมอดูเลต ส่วน

สัญญาณรูปสามเหลี่ยมเป็นคลื่นพาห์ในกระบวนการมอดูเลชั่น ถ้าใช้อุปกรณ์ดิจิตอลหรือไมโครโปรเซสเซอร์ แทนคอมพาราเตอร์ที่เป็นอุปกรณ์อะนาลอกตามรูปที่ 2.18(ข) สัญญาณอะนาลอก จะถูกเปลี่ยนเป็นข้อมูลตัวเลขดิจิตอลป้อนเข้าไมโครคอมพิวเตอร์ และข้อมูลนั้นจะผ่านโปรแกรมที่มีฟังก์ชั่นเหมือนกับคอมพาราเตอร์

ในบางครั้ง อาจจะเป็นไปได้ที่จะไม่ต้องมีสัญญาณอะนาลอกป้อนเข้ามาดังรูปที่ 2.18(ค) สัญญาณที่จะถูกมอดูเลต เกิดขึ้นในความคิดของผู้ออกแบบหรือโปรแกรมเมอร์ การสร้างสัญญาณ PWM จะอยู่ในหน่วยความจำของคอมพิวเตอร์ อันเป็นวิธีทางดิจิตอลอย่างสมบูรณ์ รูปที่ 2.19 แสดงการขับดีซีมอเตอร์ด้วยสัญญาณ PWM การขับแบบแรกตามรูปที่ 2.19(ก) ซึ่งเป็นภาคขับกำลังของมอเตอร์ให้หมุนในทิศทางเดียว เมื่อต่อแบบคอมมอนอีมิเตอร์จะใช้สัญญาณ PWM ที่มีขนาด 0 ถึงE ได้ แต่แบบคอมมอนคอลเลคเตอร์ต้องใช้สัญญาณ PWM ทั้งบวกและลบ และแรงดันสัญญาณอินพุตที่จะมอดูเลตต้องอยู่ในย่าน -E ถึง +E ด้วยเช่นกัน



รูปที่ 2.18 วิธีการสร้าง PWM
(ก) การใช้คอมพาราเตอร์เป็นวิธีทางอะนาลอก
(ข) วิธีทางอะนาลอกผสมกับดิจิตอลเรียกว่าไฮบริดจ์
(ค) วิธีการทางดิจิตอลที่สมบูรณ์

รูปที่ 19(ข) เป็นการขับแบบสองทิศทางใช้แหล่งจ่ายกำลัง +E ถึง -E ส่วนวิธีที่สามในรูปที่ 19(ค) ใช้แหล่งจ่ายเพียงแหล่งเดียวต่อทรานซิสเตอร์แบบไฮบริดจ์และมอเตอร์หมุนได้สองทิศทาง

ป
TF
456
J137
2543

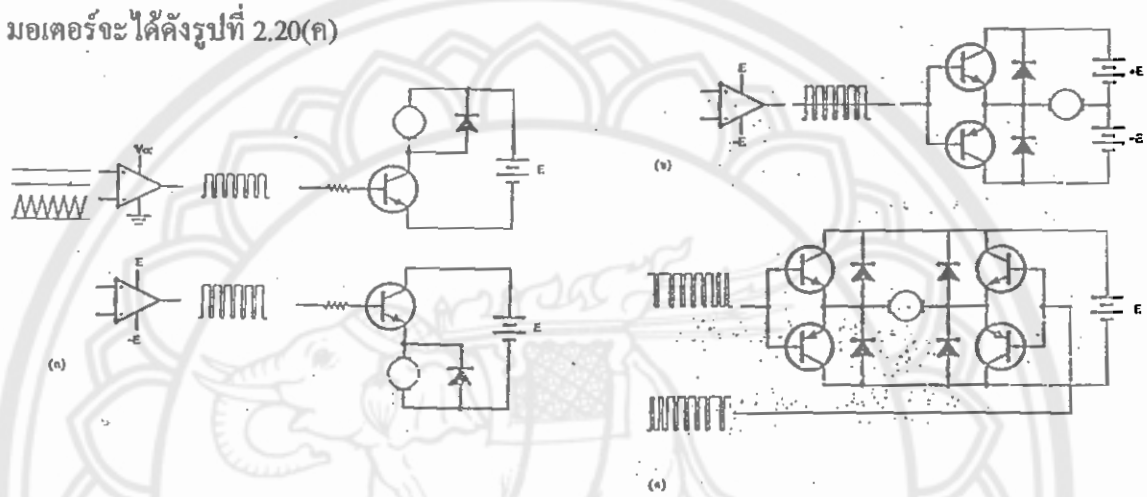
- 9 พ.ค. 2544

4440104

สำนักหอสมุด

การดีมอดูเลชั่นสัญญาณ PWM

การดีมอดูเลชั่นสัญญาณ PWM เป็นการนำเอารูปคลื่นสัญญาณอะนาลอกกลับคืนมาจากรูปแบบสัญญาณที่เป็นพัลส์ ในกรณีของดีซีมอเตอร์ ถ้าดูจากวงจรสมมูลตามรูปที่ 20(ก) หลักการในการดีมอดูเลชั่นแสดงให้เห็นชัดเจนที่สุดเมื่อมอเตอร์หยุดนิ่ง ตัวเก็บประจุ (แทนโรเตอร์) ขณะเริ่มต้นจะลัดวงจรความต้านทาน Rd (กำลังงานกลที่เอาท์พุต) ซึ่งจะเป็นศูนย์เมื่อมอเตอร์หยุดนิ่ง ซึ่งจะได้วงจรสมมูลในขณะหยุดนิ่ง ดังรูปที่ 20(ข) ความสัมพันธ์ระหว่างสัญญาณ PWM กับกระแสในมอเตอร์จะได้ดังรูปที่ 2.20(ค)

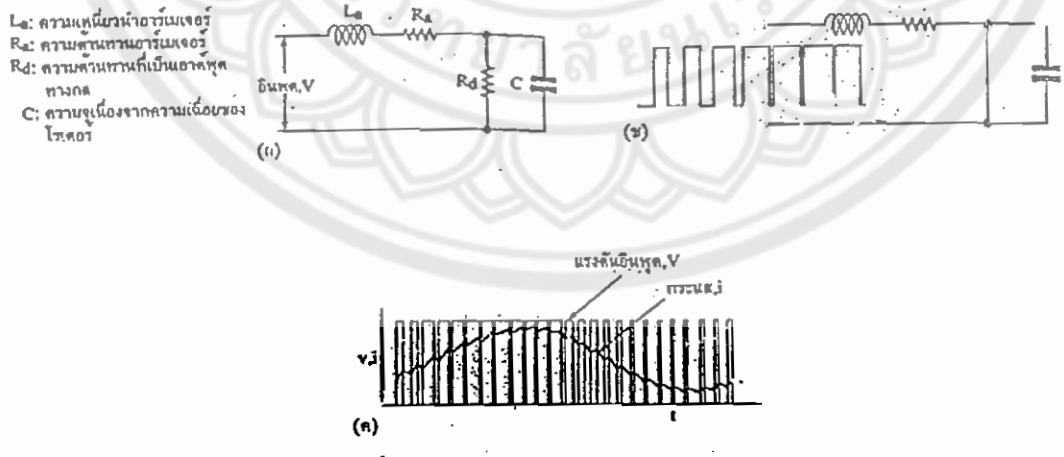


รูปที่ 2.19 การขับดีซีมอเตอร์ด้วยวิธี PWM

(ก) ต่อแบบคอมมอนอีมิตเตอร์และคอมมอนคอลเลคเตอร์ขับแบบทิศทางเดียว

(ข) ขับแบบสองทิศทางใช้แหล่งจ่ายคู่

(ค) ขับแบบสองทิศทางใช้แหล่งจ่ายเดี่ยวต่อแบบไฮบริดจ์



รูปที่ 2.20 วงจรสมมูลของดีซีมอเตอร์

(ก) วงจรสมมูลทั่วไป

(ข) วงจรสมมูลขณะมอเตอร์หยุดนิ่ง

(ค) รูปคลื่นกระแส

ถ้าหากไม่นำริปเปิ้ลของกระแสมาพิจารณา รูปคลื่นของกระแส (เส้นขีดประ) จะเหมือนกับกระแสที่ได้จากแหล่งจ่ายอะนาลอกที่จ่ายให้กับดีซีมอเตอร์ ความถี่ของการมอดูเลตที่สูงก็ช่วยลดริปเปิ้ลได้อีกทางหนึ่งโดยเฉพาะเมื่อคาบเวลาของพัลส์น้อยกว่าค่าคงที่เวลา (Time Constant) ในกรณีนี้คือ La/Ra ริปเปิ้ลจะมีค่าน้อยจนอาจไม่ต้องนำมาคิด และถ้าต้องการให้การเปลี่ยนแปลงของกระแสเป็นไปอย่างราบเรียบ ก็ให้คาบเวลาของพัลส์น้อยกว่า ค่าคงที่ของเวลาทางกล (Machanical Time Constant) มาก ๆ ในกรณีนี้ไม่จำเป็นต้องได้รูปคลื่นของแรงดันอะนาลอกกลับคืนมาจากสัญญาณ PWM เนื่องจากในกรณีของมอเตอร์สัญญาณ PWM ที่ผ่านการดีมอดูเลชั่นจะอยู่ในรูปของกระแส ซึ่งเป็นพารามิเตอร์ที่จำเป็นที่เราต้องการเพื่อสร้างแรงบิด

2.6 การทำงานของบอร์ดไดรเวอร์ (Driver)

จากวงจรการทำงานของไดรเวอร์ (Driver) สัญญาณที่รับมาจากบอร์ด PWM จะถูกไอโซเลท (Isolate) ด้วยออปโต 4N25 (OPTO 4N25) สัญญาณเอนนาเบิล (Enable) จะเป็นตัวกำหนดให้ IC 74LS138 ทำงาน คือสัญญาณเอนนาเบิล L (Enable-L) ที่ต่อกับขา G_{2A} ของ 74LS138 จะต้องมีระดับสัญญาณลอจิกต่ำ (Low) ฉะนั้นสัญญาณเอนนาเบิลออปโต (Enable-OPTO) จะต้องมีระดับสัญญาณลอจิกสูง (High) นั้นคือออปโต (OPTO) จะต้องทำงาน เพราะฉะนั้นเอนนาเบิลที่ส่งมายังบอร์ด PWM ต้องมีระดับสัญญาณลอจิกต่ำ

สัญญาณไดเร็กชัน (Direction) เป็นตัวกำหนดทิศทางของเอาต์พุต (Output) ว่าเป็น A หรือ B คือถ้าสัญญาณไดเร็กชัน L (Direction-L) ที่ต่อกับขา 1 ของ 74LS138 มีระดับสัญญาณลอจิกต่ำ เอาต์พุตของ 74LS138 ก็จะกระตุ้นที่ Y0 ทำให้ทิศทางคือ A สัญญาณไดเร็กชันออปโต้ (Direction-OPTO) จะต้องมีระดับสัญญาณลอจิกต่ำ นั้นคือออปโตไม่ทำงานฉะนั้นสัญญาณไดเร็กชันที่ส่งมายังบอร์ด PWM ก็จะต้องมีระดับสัญญาณลอจิกสูงในทางกลับกันถ้าต้องการให้ทิศทางเป็น B เอาต์พุตของ 74LS138 ก็ต้องกระตุ้นที่ Y1 สัญญาณไดเร็กชัน L ต้องมีระดับสัญญาณลอจิกสูง ไดเร็กชันออปโต้ ก็ต้องมีระดับสัญญาณลอจิกสูงด้วย นั้นคือออปโต้ทำงานตามไดเร็กชันที่ส่งมายังบอร์ด PWM จะต้องมีระดับสัญญาณลอจิกต่ำสัญญาณ PWM ที่เข้ามาก็มีขั้นตอนการทำงานเช่นเดียวกับสัญญาณทั้งสองข้างต้น โดยสัญญาณ PWM-L ที่ต่อกับขา G1 ของ 74LS138 จะมีระดับสัญญาณลอจิกสูงหรือต่ำตามพัลส์วิดธ์ (Pulse Width) ตามนั้น สัญญาณทั้ง 3 ข้างต้นจะมีการแสดงผลโดยแอลอีดี (LED) โดยผ่าน 74LS04 ก่อนเพื่อกลับเฟสให้สัมพันธ์กับการทำงานและมีแอลอีดีแสดงทิศทางว่าเป็น A หรือ B ด้วย

นอกจากนั้นยังมีการตรวจเช็คกระแสเกินโดยมีตัวต้านทานวัตต์สูง ๆ มาต่ออนุกรม แล้วดึงสัญญาณที่ได้มาเข้าขาบวกของออปแอมป์ LM358 (OP-AMP LM358) โดยผ่าน R, C ค่าน้อย ๆ เพื่อกรองสัญญาณรบกวนขาลบของออปแอมป์ ตั้ง V_{ref} ไว้ ในการทำงานปกติ ถ้าไม่เกิดกระแส**แรงดัน (Voltage) ที่ขาบวกจะน้อยกว่าขาลบ ฉะนั้นเอาต์พุตพี (Output P) จะมีระดับสัญญาณลอจิกต่ำไปต่อเข้ากับขา 2 ของ 74LS138 และทำงานตามปกติ แต่เมื่อเกิดกระแสเกิน หรือบางครั้งจะเป็นภาวะที่ไม่แน่นอน เอาต์พุตพีจะเกิดเป็นพัลส์ (Pluse) เล็ก ๆ ขึ้น จะใช้ IC 74LS123 เป็นตัวตรวจเช็คดังนี้**

ตารางการทำงานของ 74LS123 (Function Table of 74LS123)

ตารางที่ 2.1 การทำงานของ 74LS123

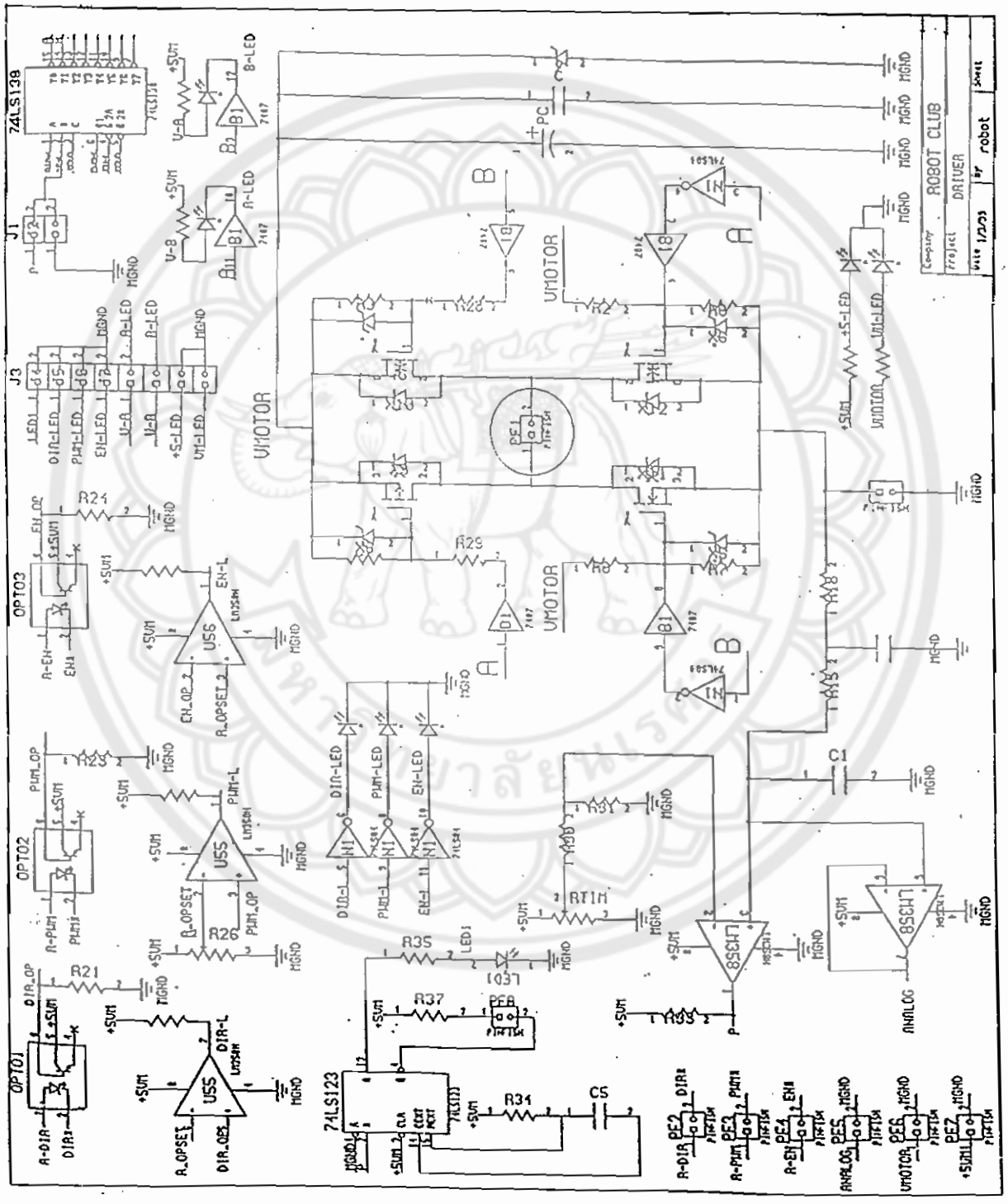
INPUT			OUTPUT	
CLEAR	A	B	Q	Q
L	X	X	L	H
X	H	X	L	H
X	X	L	L	H
H	L	↑	⎍	⎍
H	↓	H	⎍	⎍
↑	L	H	⎍	⎍

จากตารางการทำงานของ 74LS123 ในสภาวะปกติ ขา A และขา B (ซึ่งต่อกับขา P) มีระดับสัญญาณลอจิกต่ำ ฉะนั้น Q จึงมีระดับสัญญาณลอจิกต่ำด้วย แอลอีดีจึงไม่สว่าง แต่เมื่อเกิดกระแสเกินหรือสภาวะที่ไม่แน่นอน ขา B เปลี่ยนจากระดับสัญญาณลอจิกต่ำมาเป็นระดับลอจิกสูง ทำให้ขา Q ซึ่งมีระดับสัญญาณลอจิกต่ำอยู่ เกิดพัลส์มีความกว้างขึ้นอยู่กับค่า R และ C ที่ต่อเข้ากับขา C_{EXT} และ R_{EXT} ซึ่งถ้าเป็นสภาวะที่ไม่แน่นอน ขา B จะเปลี่ยนระดับสัญญาณลอจิกต่ำเป็นสูงอยู่ตลอดเวลา จึงทำให้ขา Q มีระดับสัญญาณสูงอยู่ตลอดเวลาด้วย ทำให้แอลอีดีสว่างแสดงให้รู้ว่าการทำงานผิดปกติหรือมีกระแสเกิน

บอร์ดไดรเวอร์ (Driver)

อุปกรณ์อิเล็กทรอนิกส์ที่ใช้มีดังต่อไปนี้

มอสเฟท MTP12P10	**2 ตัว**
MTP15N06	2 ตัว
ออปโต้4N25 (OPTO 4N25)	3 ตัว
IC 74LS04	1 ตัว
IC 74LS07	1 ตัว
IC 74LS138	1 ตัว
IC 74LS123	1 ตัว
IC LM358	3 ตัว
ซีเนอร์ไดโอด (Zener Diode) 1N4735A	4 ตัว
ไดโอด (Diode) 1N4007	4 ตัว
ตัวต้านทาน 1.5 กิโลโอห์ม 0.5 วัตต์	4 ตัว
ตัวต้านทาน 1 กิโลโอห์ม 0.5 วัตต์	4 ตัว
ตัวต้านทาน 1 กิโลโอห์ม 0.25 วัตต์	11 ตัว
ตัวต้านทาน 10 กิโลโอห์ม 0.25 วัตต์	4 ตัว
ตัวต้านทาน 5 กิโลโอห์ม 0.25 วัตต์	1 ตัว
ตัวต้านทาน 10 โอห์ม 0.25 วัตต์	2 ตัว
ตัวต้านทาน 0.2 โอห์ม 5 วัตต์	1 ตัว
ตัวเก็บประจุอิเล็คทรอไลต์ 2200 µF	1 ตัว
ตัวเก็บประจุอิเล็คทรอไลต์ 100 µF	1 ตัว
ตัวเก็บประจุอิเล็คทรอไลต์ 47 µF	1 ตัว
ตัวเก็บประจุอิเล็คทรอไลต์ 10 µF	1 ตัว
ตัวต้านทานปรับค่าได้ 1 KΩ	**2 ตัว**
จัมพ์เปอร์ 2 ขา (Jumper 2 Pins)	**2 ตัว**
คอนเนคเตอร์ 16 ขา (Connecter 16 Pins)	**1 ตัว**
แอลอีดี (LED)	**8 ตัว**
ขั้วต่อ 2 ขา (Terminal 2 pin)	**8 ตัว**

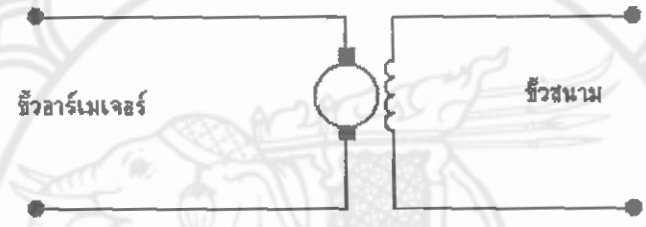


รูปที่ 2.21 วงจรไดรเวอร์

2.7 มอเตอร์กระแสตรง

มอเตอร์กระแสตรง นั้นสามารถจำแนกออกไปได้อีกหลายประเภท ซึ่งขึ้นอยู่กับวิธีการสร้าง จึงขอกล่าวรวม ๆ ที่รู้จักกันมาเป็นส่วนใหญ่ในปัจจุบันนั่นก็คือ มอเตอร์กระแสตรงแบบขนาน (Shunt DC Motor), แบบอนุกรม (Series), แบบผสม (Compound) และแบบแม่เหล็กถาวร (Permanent Magnet Motor)

ลักษณะโครงสร้างของมอเตอร์กระแสตรงแบบขนาน



รูปที่ 2.22 ลักษณะโครงสร้างของมอเตอร์กระแสตรงแบบขนาน

มอเตอร์แบบนี้สามารถปรับเส้นแรงแม่เหล็กได้อย่างอิสระต่อกระแสของอาร์เมเจอร์ เป็นผลทำให้สามารถควบคุมพารามิเตอร์ให้มีค่าคงที่ได้ตลอดช่วงพิสัยที่กว้าง มอเตอร์ชนิดนี้จึงมักใช้ในงานระบบควบคุมการเคลื่อนที่ ที่ต้องการแรงบิดสูง

ลักษณะโครงสร้างของมอเตอร์กระแสตรงแบบอนุกรม

รูปที่ 2.23 ลักษณะโครงสร้างของมอเตอร์กระแสตรงแบบอนุกรม

มอเตอร์แบบนี้จะมีเส้นแรงแม่เหล็กเป็นสัดส่วนกับกระแสดังนั้นเส้นแรงของสนามแม่เหล็กจึงสามารถปรับค่าได้ ซึ่งเราจะได้ความสัมพันธ์ ระหว่างความเร็วและแรงบิดเป็นแบบไม่เชิงเส้น จึงเหมาะไปใช้งานในภาวะเฉพาะคือ เมื่อต้องการแรงบิดสูงที่ความเร็วต่ำ และแรงบิดต่ำที่ความเร็วสูง เช่น ระบบการขับเคลื่อนของรถลาก

ลักษณะโครงสร้างของมอเตอร์กระแสตรงแบบแม่เหล็กถาวร

รูปที่ 2.24 ลักษณะโครงสร้างของมอเตอร์กระแสตรงแบบแม่เหล็กถาวร

มอเตอร์แบบนี้จะใช้การกระตุ้นสนามแม่เหล็กของมอเตอร์เป็นแม่เหล็กถาวรซึ่งต่างจากที่กล่าวมาข้างต้นที่ใช้ขดลวดซึ่งแบบนี้จะให้เส้นแรงของสนามแม่เหล็กมีค่าคงที่ ดังนั้น อัตราส่วนระหว่างกระแสอาร์เมเจอร์และแรงบิดจะมีค่าคงที่ด้วย ซึ่งมีข้อดีคือ ไม่มีกำลังสูญเสียในสนามแม่เหล็ก มีประสิทธิภาพสูงกว่าและมีขนาดเล็กกว่า เมื่อเทียบกับมอเตอร์แบบใช้ขดลวดในการกระตุ้นที่มีขนาดแรงม้าเท่ากัน จึงเหมาะกับงานที่ต้องการแรงบิดของโหลดสูง

ระบบการควบคุมมอเตอร์กระแสตรงแบบพื้นฐาน

รูปที่ 2.25 ระบบควบคุมมอเตอร์กระแสตรงพื้นฐาน

ตัวควบคุม(Controller)

เป็นส่วนของระบบที่ทำให้เกิดสัญญาณควบคุมไปบังคับมอเตอร์กระแสตรงและโหลดซึ่งอาจจะเป็นอะนาลอกหรือดิจิตอลก็ได้

หรือส่วนไดรเวอร์ทำหน้าที่ปรับปรุงและขยายสัญญาณให้เหมาะสมก่อนที่จะป้อนไปขับมอเตอร์กระแสตรงซึ่งอาจแบ่งแยกเป็นเซอร์โวแอมปลิไฟร์ชนิดเชิงเส้นและพัลส์วิดธ์มอดูเลชั่น

เซอร์โวแอมปลิไฟร์ชนิดเชิงเส้น

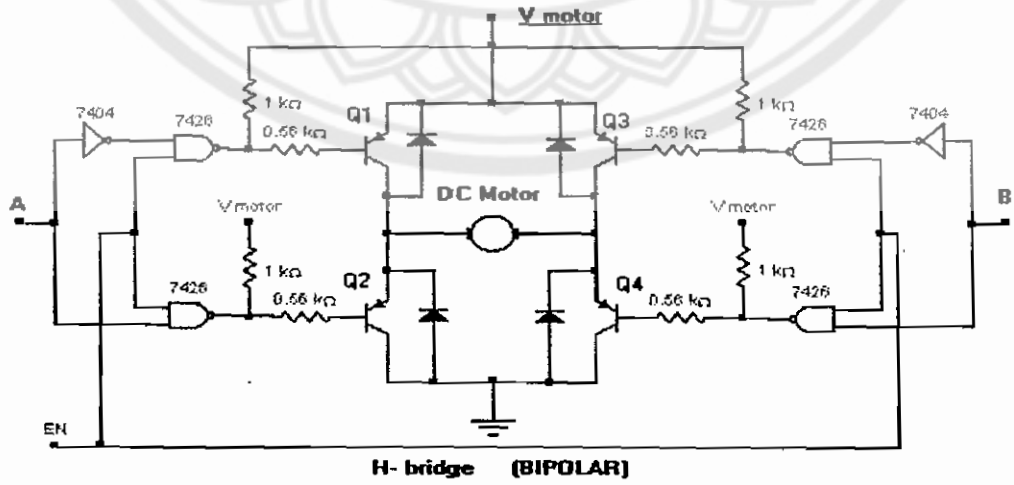
เป็นการควบคุมมอเตอร์แบบต่อเนื่องแต่จะมีความสูญเสียทางเพาเวอร์สูงเนื่องจากกำลังส่วนใหญ่จะสูญเสียในเอาต์พุตของทรานซิสเตอร์เป็นจำนวนมากเพราะขณะมอเตอร์ไม่ทำงานทรานซิสเตอร์ส่วนนี้ก็ต้องแบกภาระเนื่องจากมีกระแสไหลผ่านตัวมัน

พัลส์วิดธ์มอดูเลชั่น PWM

เป็นสวิทชิ่งแอมปลิไฟร์คือ การควบคุมแรงดันของมอเตอร์โดยการปรับดิวตี้ไซเคิล (Duty Cycle) ของแรงดันที่จะจ่ายให้กับมอเตอร์และให้มันทำงานทุกๆ ภาวะอิ่มตัว (ON) หรือภาวะไม่นำกระแส (OFF) ด้วยเหตุนี้กำลังสูญเสียน้อยเนื่องจากเมื่อทรานซิสเตอร์นำกระแส แรงดันตกคร่อมตัวมันจะน้อยจนตัดทิ้งได้ และเมื่อหยุดนำกระแส แรงดันตกคร่อมจะประมาณ VCC ดังนั้นกระแสไหลผ่านจึงน้อยมากประมาณศูนย์ แต่จะใช้กับความถี่สูงมากได้ไม่ดี และความถี่ต้องคงที่ถ้าไม่เช่นนั้นอาจเกิดการออสซิเลทขึ้นได้

มอเตอร์กระแสตรงและโหลด

คือระบบที่ถูกควบคุมหรือส่วนที่ออกแรงทำงานซึ่งจะเป็นเครื่องจักรกล



รูปที่ 2.26 วงจรไฮบริดจ์ (H – Bridge Bipolar)

ทรานซิสเตอร์ทุกตัวจะทำงานที่ ลอจิก 0

ถ้า Enable = 0 มอเตอร์จะ OFF

ถ้า Enable = 1 มอเตอร์จะ ON

ถ้า A = 0 และ B = 1 Q1 และ Q4 จะ ON ทำให้มีกระแสไหลผ่านมอเตอร์

ถ้า A = 1 และ B = 0 Q2 และ Q3 จะ ON ทำให้มีกระแสไหลผ่านมอเตอร์อีกทิศทางหนึ่ง

หมายเหตุ A และ B ห้ามเป็นลอจิกตามกัน เพราะจะทำให้วงจรเกิดการช็อต (Short Through) แล้วทรานซิสเตอร์จะไหม้

การถอดรหัส

หรือฟีดแบ็คทรานดิวเซอร์ใช้รับรู้หรือป้องกันสัญญาณเอาต์พุตที่ต้องการโดยไม่มีผลของการโหลดสัญญาณที่ป้องกันได้นี้จะป้อนกลับไปเปรียบเทียบกับสัญญาณอ้างอิงเพื่อควบคุมมอเตอร์อีกที ฟีดแบ็คทรานดิวเซอร์แบ่งแบบอะนาลอกและดิจิตอล

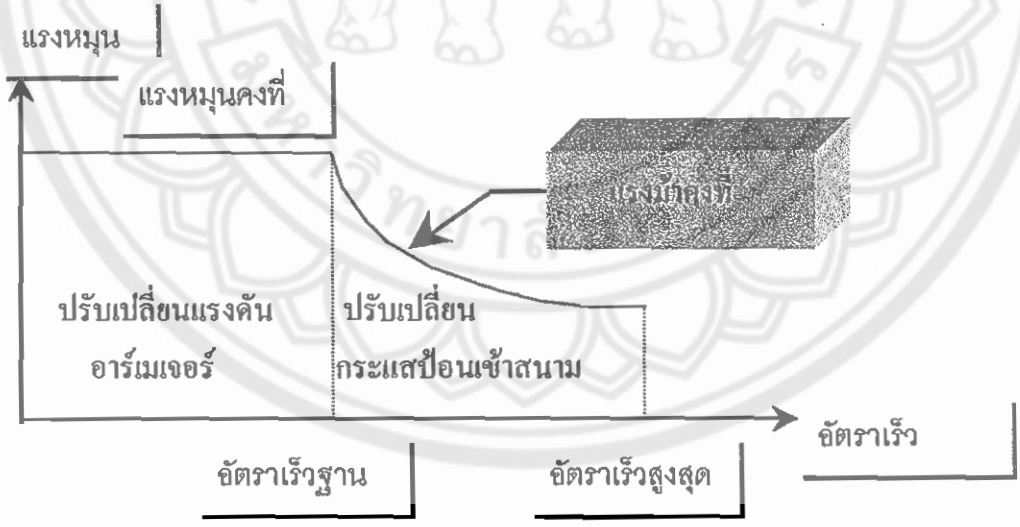
การควบคุมความเร็วมอเตอร์กระแสตรง

ทำได้ 2 วิธีคือ

1. การควบคุมแรงดันไฟตรงของอาร์เมเจอร์ (Armature Voltage Control)
2. การควบคุมความเข้มของสนามแม่เหล็ก (Field Control)

การควบคุมแรงดันไฟตรงของอาร์เมเจอร์

เนื่องจากความเร็วของมอเตอร์กระแสตรงจะแปรผันตรงกับแรงดันที่ใส่ให้กับขดลวดอาร์เมเจอร์ ดังนั้นเราจึงสามารถควบคุมความเร็วของมอเตอร์โดยการควบคุมแรงดันของอาร์เมเจอร์วิธีการนี้จะใช้ในช่วงความเร็วที่ต่ำกว่าความเร็วที่กำหนด (Base Speed) การควบคุมความเร็วของมอเตอร์กระแสตรงโดยทั่วไปจะใช้วิธีนี้เพราะให้แรงบิทสูง

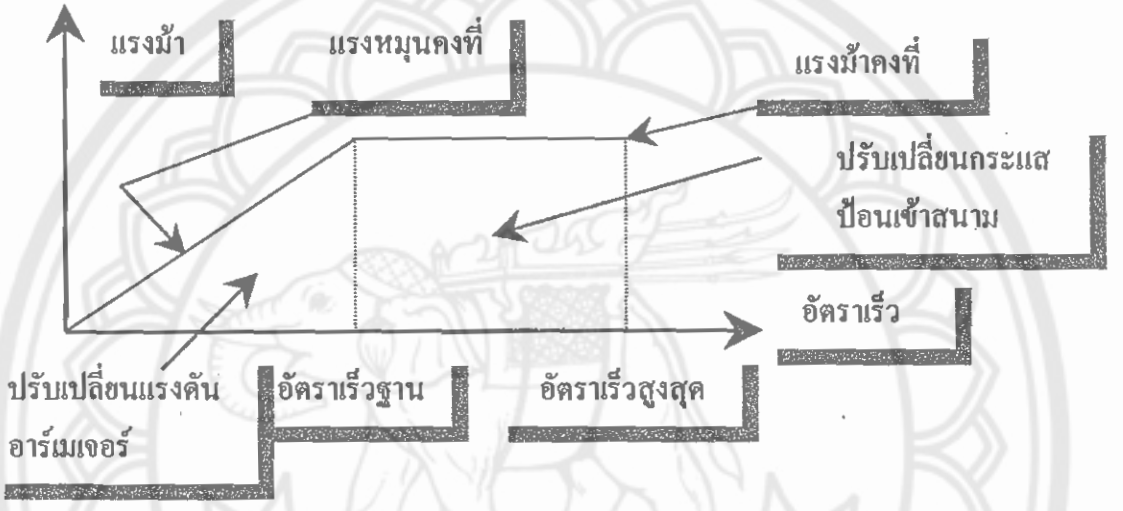


กราฟที่ 2.1 ความสัมพันธ์ระหว่างอัตราเร็วกับแรงหมุนของมอเตอร์

การควบคุมความเข้มของสนามแม่เหล็ก

การควบคุมความเร็วของมอเตอร์กระแสตรงในย่านความเร็วที่สูงกว่าความเร็วที่กำหนดจะทำได้โดยการควบคุมกระแสของขดลวดสร้างสนามแม่เหล็กของมอเตอร์ เมื่อเราต้องการเพิ่ม

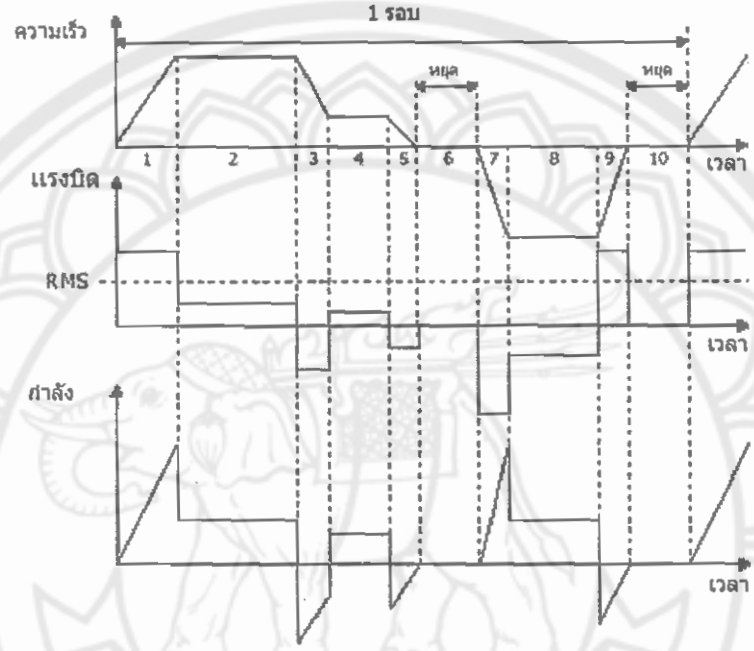
ความเร็ว เราจะต้องลดขนาดของกระแสของขดลวดลง การลดความเข้มของสนามแม่เหล็กลงของมอเตอร์จะมีผลทำให้แรงบิดสูงสุดของมอเตอร์ลดลง ขณะที่กำลังออกสูงสุดของมอเตอร์จะไม่เปลี่ยนแปลงดังแสดงในกราฟที่ 2.2 วิธีนี้จะใช้กับโหลดที่ต้องการความเร็วสูงโดยที่แรงบิดของโหลดจะต้องลดลงเมื่อความเร็วเพิ่มขึ้น ไม่เช่นนั้นมอเตอร์จะเกิดการรับโหลดมากเกินไป (Over Load Motor)



กราฟที่ 2.2 แสดงความสัมพันธ์ระหว่างแรงม้ากับอัตราเร็วของมอเตอร์

ความสัมพันธ์ระหว่างความเร็ว,แรงบิด,กำลัง

ขนาดแรงดันและกระแสพิกัดของมอเตอร์ขึ้นอยู่กับความเร็วและแรงบิดสูงสุด ขณะที่แรงบิดสูงสุดและแรงบิดอาร์เอ็มเอส (RMS Torque) จำเป็นต้องคำนวณไว้ก่อนขณะเริ่มออกแบบ



กราฟที่ 2.3 แสดงความสัมพันธ์ของความเร็ว-เวลา, แรงบิด-เวลา, กำลัง-เวลา

แรงบิดอาร์เอ็มเอส (RMS Torque)

คำนวณดังสูตรต่อไปนี้

$$T_{rms} = \sqrt{\frac{\sum T_i^2 * t_i}{\sum t_i}}$$

T_i คือ ทอร์ค

t_i คือ เวลาในแต่ละ i

$$V_t = I_m * R_m + E$$

$$E = K_e * S_m$$

$T = K_t * I_m$ สำหรับมอเตอร์แบบโรตารี (Rotary Motor) หรือ

$F = K_f * I_m$ สำหรับมอเตอร์เชิงเส้น (Linear Motor)

$$Tsh = \frac{9.55\ Pout}{N} \quad N\text{-}m$$

คำอธิบาย

Vt	แรงดันที่ปลายขั้วมอเตอร์ [V]
Im	กระแสเข้ามอเตอร์ [A]
Rm	ความต้านทานของขดลวดมอเตอร์ [Ω]
E	แรงเคลื่อนต่อต้าน (Back – EMF Voltage) [V]
T	ทอร์คมอเตอร์ [Nm or lb.-in]
F	แรงมอเตอร์ [N or lb.]
Kt	ค่าคงที่ของทอร์คมอเตอร์ [Nm/A or lb.-in/A]
Kr	ค่าคงที่ของแรงมอเตอร์ [N/A or lb./A]
Ke	ค่าคงที่ของแรงดัน [V/K_{rpm} or V/m/s]
Sm	ความเร็วของมอเตอร์ [rpm or m/s]
N	ความเร็วรอบ [rpm]

แรงดันสูงสุดและกระแสสูงสุดจะคำนวณได้จากสมการข้างต้น

ตัวอย่าง มอเตอร์มีค่า Ke = 10 V/Krpm จะมีความเร็วรอบ 3000 rpm ที่ 30 V

กระแสสูงสุด คือ ทอร์คสูงสุดหารด้วยค่า Kt

ตัวอย่าง มอเตอร์ที่มีค่า Kt = 0.5 Nm/A และมีทอร์คสูงสุดที่ 5 Nm ควรมีกระแสที่ต้องการคือ 10 A