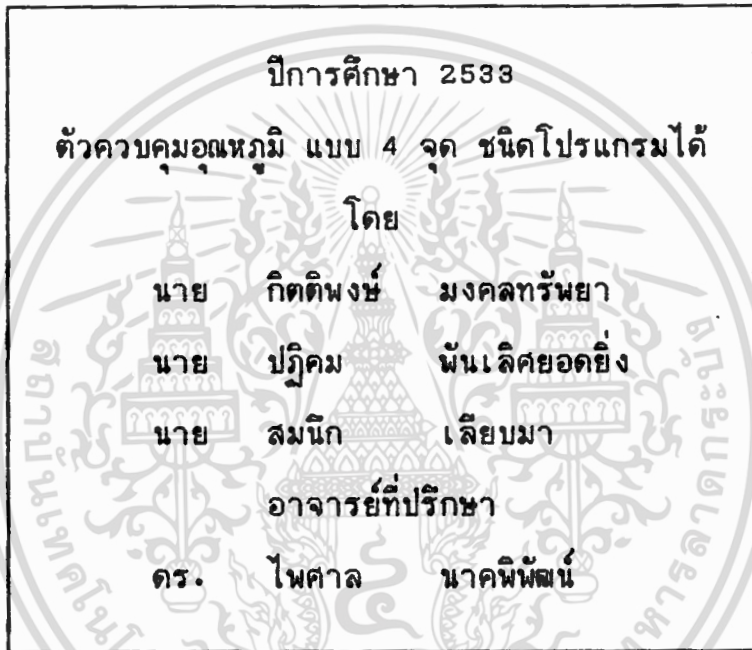




028800

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์อื่นใด
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวควบคุมอุณหภูมิ แบบ 4 จุด ชนิดโปรแกรมได้
FOUR-POINT TEMPERATURE CONTROLLER

กิตติพงษ์ มงคลทรัพย์หา

ปฏิคม พันเลิศยอดยิ่ง

สมนึก

เลียบมา

ปริญญาบัตรสำหรับปริญญาอุตสาหกรรมศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2533

ปริญญานิพนธ์ปีการศึกษา 2533

ภาควิชา เทคโนโลยีอุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

เรื่อง ตัวควบคุมอุณหภูมิ แบบ 4 จุด ชนิดโปรแกรมได้

ผู้จัดทำ

1. นาย กิตติพงษ์ มงคลทรัพย์ 323303
2. นาย ปฏิคม พันเลิศยอดยิ่ง 323315
3. นาย สมนึก เลียบมา 323332

----- อาจารย์ที่ปรึกษา
(ดร. ไพศาล นาคินวัฒน์)

----- คณะกรรมการ
(อาจารย์ วิชัย สุรพันธ์)

----- คณะกรรมการ
(อาจารย์ ประดิษฐ์ วัชรนิบูลย์)

----- คณะกรรมการ
(อาจารย์ อุทัย ศรีธีระวิโรจน์)

ตัวควบคุมอุณหภูมิแบบ 4 จุด ชนิดโปรแกรมได้

กิตติพงษ์ มงคลทรัพย์

ปฏิคม พันเลิศยอดยิ่ง

สมนึก เลียบมา

ดร.ไพศาล นาคพัฒน อาจารย์ที่ปรึกษา

ปีการศึกษา 2588

บทคัดย่อ

โครงงานฉบับนี้ เป็นการนำเอาไมโครโปรเซสเซอร์ เบอร์ 280 มาใช้ประโยชน์ในงานควบคุมอุณหภูมิโดยโครงงานนี้สามารถวัด และ ควบคุมอุณหภูมิให้อยู่ ณ ค่าที่ตั้งไว้ (Set Point) ได้ 4 จุดในเวลาเดียวกัน และมีการแสดงผลบนจอ

FOUR-POINTS TEMPERATURE CONTROLLER

KITTIPONG MONGKOLSAPPRAYA

PATIKOM PHUNLEARDYODYING

SOMNUK LIABMA

Dr. PAISAL NAKPIPAT ADVISOR

1990

ABSTRACT

This Project is the presentation of controller utilizing Z-80 microprocessor. It can measure and remote control any four temperature set points in the same time and the results can be displayed on screen.

สารบัญ

	หน้า
บทนำ	1
บทที่ 1 ทฤษฎี	2
บทที่ 2 ไมโครโปรเซสเซอร์ Z80	17
บทที่ 3 การออกแบบ	32
บทที่ 4 การทดลองและผลการทดลอง	52
บทที่ 5 สรุปและวิจารณ์	59

ภาคผนวก ก

ภาคผนวก ข

กิจกรรมประกาศ

บรรณานุกรม

บทนำ

ในปัจจุบัน วิทยาการทางเทคโนโลยีก้าวหน้าไปมาก ได้มีการนำเอา คอมพิวเตอร์มาใช้งานในด้านต่างๆมากมายได้แก่ การควบคุมการทำงานของเครื่อง ใช้ไฟฟ้าในบ้าน ไปจนถึงการควบคุมขบวนการที่ซับซ้อน ในโรงงานอุตสาหกรรม โดย ในการทำงานจะใช้ ไมโครโปรเซสเซอร์เป็นหน่วยประมวลผลกลาง และทำงาน ประกอบกับส่วนอื่น ๆ อันได้แก่ หน่วยความจำ, พอร์ตอินพุตและเอาต์พุต ส่วนที่ใช้ เปลี่ยนสัญญาณอนาลอกเป็นดิจิทัล แต่งานควบคุมบางชนิด เช่น การควบคุมการทำงาน ของเครื่องมีวัตชนิดต่างๆ การนำเอาไมโครคอมพิวเตอร์ทั้งเครื่องมาใช้ควบคุม การทำงาน อาจเกิดความไม่สะดวกในการเคลื่อนย้าย หรืออาจไม่คุ้มค่ากับการ ลงทุน จึงได้มีการนำเอาหน่วยต่าง ๆ ของ ไมโครคอมพิวเตอร์บางส่วน ๆ เช่น ไมโครโปรเซสเซอร์, หน่วยความจำ, พอร์ตอินพุต และเอาต์พุตมาใช้ในงานควบคุมด้านต่าง ๆ เป็นหลัก อันเป็นผลให้สะดวกต่อการใช้งาน

ทฤษฎีและหลักการ

1.1 หลักการวัด

เครื่องวัดอุณหภูมิมีหลายชนิดโดยเครื่องมือแต่ละชนิดอาศัยการเปลี่ยนแปลงคุณสมบัติเฉพาะของสารคือจะต้องมีการเปลี่ยนแปลงที่วัดได้ เมื่ออุณหภูมิที่วัดเปลี่ยนไปและการเปลี่ยนแปลงที่วัดได้จะต้องคงที่แน่นอนและนิสจุนได้

1. หลักการวัดอุณหภูมิโดยอาศัยการเปลี่ยนแปลงคุณสมบัติทางเชิงกล

เทอร์โมมิเตอร์	หลักการวัด	วัสดุที่ใช้	ช่วงการวัด
ไร้ช่องเหลวบรรจุในหลอดแก้วปิด (thermometer)	Δ อุณหภูมิ $\rightarrow$ เทอร์โมมิเตอร์ $\rightarrow$ Δ การขยายตัวของช่องเหลว	ปรอท แอลกอฮอล์	- 130 ถึง 315°C [- 200 ถึง 600°F]
เปลี่ยนการขยายตัวของความดัน (Filled Thermal)	Δ อุณหภูมิ $\rightarrow$ เทอร์โมมิเตอร์ แบบ Filled Thermal $\rightarrow$ Δ ความดัน $\rightarrow$ Δ ปริมาตร	ปรอท แอลกอฮอล์	- 185 ถึง 540°C [- 300 ถึง 1000°F]
แบบแถบโลหะคู่	Δ อุณหภูมิ $\rightarrow$ ตัววัด Bimetal $\rightarrow$ Δ ระยะทาง	INVAR-Ni + Fe + Cr.	- 60 ถึง 425°C [- 30 ถึง 800°F]

2. หลักการวัดอุณหภูมิโดยอาศัยการเปลี่ยนแปลงคุณสมบัติทางไฟฟ้า

เทอร์โมมิเตอร์	หลักการวัด	วัสดุที่ใช้	ช่วงการวัด
เทอร์โมคัปเปิล	Δ อุณหภูมิ $\rightarrow$ เทอร์โมคัปเปิล $\rightarrow$ Δ แรงเคลื่อนไฟฟ้า	-type B - S - R - K - E - J - T	200 ~ + 1,700°C 0 ~ + 1,600°C 0 ~ + 1,600°C -200 ~ + 1,200°C -200 ~ + 800°C -200 ~ + 800°C -200 ~ + 350°C
อาร์ทีดี	Δ อุณหภูมิ $\rightarrow$ RTD $\rightarrow$ Δ ความต้านทาน	พลาตินัม Pt นิกเกิล ทองแดง	-255 ~ 900°C -150 ~ 300°C -200 ~ 120°C
เทอร์มิสเตอร์	Δ อุณหภูมิ $\rightarrow$ เทอร์มิสเตอร์ $\rightarrow$ Δ ความต้านทาน	เทอร์มิสเตอร์	- 30 ~ 300°C

3. หลักการวัดอุณหภูมิโดยอาศัยการเปลี่ยนแปลงคุณสมบัติทางแสงและการแผ่รังสี

เทอร์โมมิเตอร์	หลักการวัด	วัสดุที่ใช้	ช่วงการวัด
ออปติคอลไพโร-มิเตอร์	ความเข้ม — เปรียบเทียบความเข้มของแสง — อุณหภูมิ	-	> 700°C

4. หลักการวัดอุณหภูมิโดยวิธีการทางเคมี

เทอร์โมมิเตอร์	หลักการวัด	วัสดุที่ใช้	ช่วงการวัด
crayon แล็กเกอร์	Δ อุณหภูมิ — แล็กเกอร์ — — วัสดุละลาย — — เปลี่ยนสี	สารเคมี	37 ถึง 150°C 100 ถึง 300°C

1.2 เทอร์โมคัปเปิล (Thermocouples)

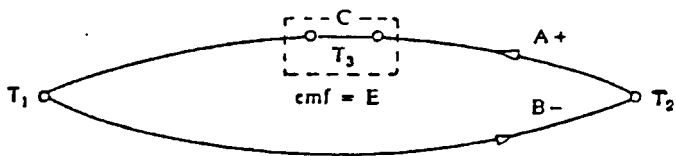
กฎของการเกิดแรงเคลื่อนไฟฟ้าเนื่องจากความร้อน (Thermoelectric laws)
สำคัญที่ควรทราบ

ก. กฎของวงจรโลหะชนิดเดียว (Law of the homogeneous circuit)

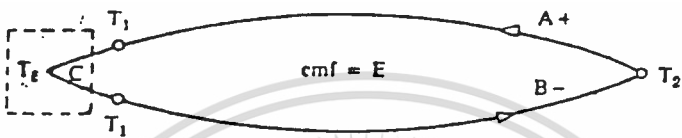
ในวงจรที่ประกอบขึ้นด้วยโลหะชนิดเดียวกัน จะไม่เกิดกระแสไฟฟ้าไหลในวงจรเมื่อเกิดความแตกต่างของอุณหภูมิ หาก ณ ปลายจุดต่อทั้งสองกระแสไฟฟ้าจะเกิดขึ้นในกรณีที่สองเป็นโลหะต่างชนิดกันแรงเคลื่อนไฟฟ้าที่เกิดขึ้น จะเห็นสัดส่วนกับผลต่างของอุณหภูมิที่ปลายจุดต่อทั้งสอง

ข. กฎของโลหะแทรก (Law of intermediate metals)

ในวงจรเทอร์โมคัปเปิลถ้ามีโลหะแทรกอยู่ในส่วนหนึ่งใดหนึ่งของวงจรหากจุดต่อปลายโลหะแทรกทั้งสองมีอุณหภูมิเท่ากันแล้วโลหะนี้จะไม่ผลต่อแรงเคลื่อนไฟฟ้าที่เกิดขึ้นในทางปฏิบัติมีอยู่เสมอ เช่น ขั้วต่อ (connectors) หรือสายทองแดงที่ใช้ต่อเข้ามิเตอร์



โลหะ C ที่ต่อแทรกอยู่ในวงจรจะไม่ผลต่อการวัดค่าอุณหภูมิที่ปลายทั้งสองเท่ากัน

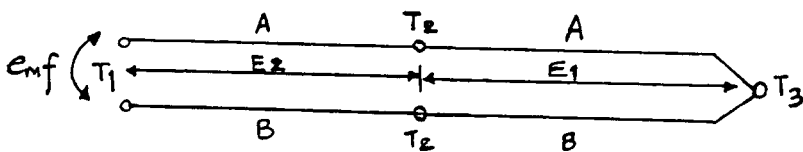


รูปที่ 1.2.1 แสดงกฎของโลหะแทรก

ค. กฎของอุณหภูมิแทรก (Law of intermediate temperature)

แรงเคลื่อนไฟฟ้าที่เกิดขึ้นโดยเทอร์โมคัปเปิลชุดใด ๆ ที่มีอุณหภูมิที่จุดต่อต่าง ๆ กันจะมีค่าเท่ากับ ผลบวกทางพีชคณิตของแรงเคลื่อนไฟฟ้าของเทอร์โมคัปเปิลชุดนั้น ตามรูปคือผลรวมของแรงเคลื่อนไฟฟ้าที่ จะมีค่าเท่ากับผลรวมของเทอร์โมคัปเปิลทั้งสองชุด

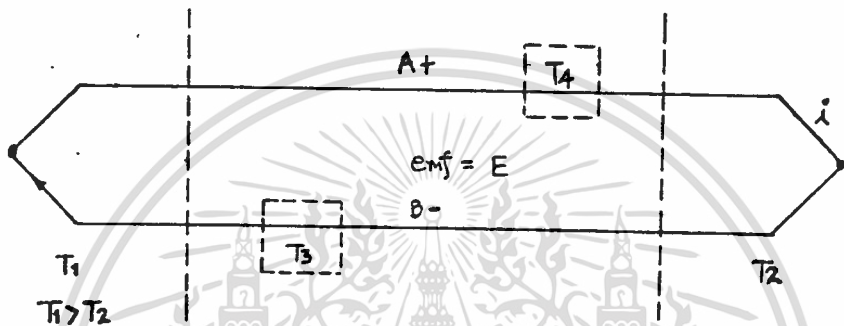
$$emf = e_1 + e_2$$



รูปที่ 1.2.2 แสดงกฎของอุณหภูมิแทรก

๑. ปรากฏการณ์ (Phenomena) ของเทอร์โมคัปเปิล

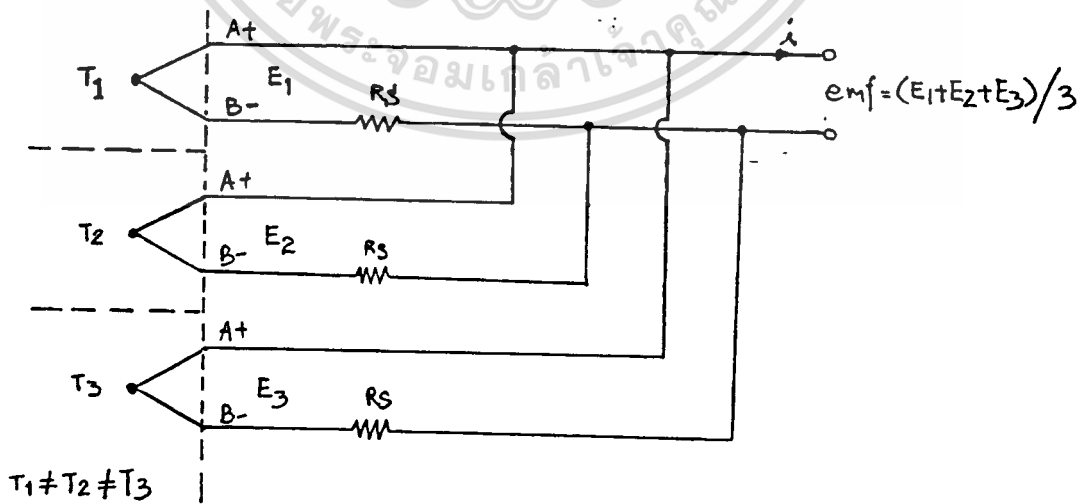
1. ในเทอร์โมคัปเปิลคู่เดียวกันอุณหภูมิที่มีผล คือ อุณหภูมิที่ปลายจุดต่อทั้งสองเท่านั้น อุณหภูมิบนจุดต่างๆ ตามสายจะไม่มีผลต่อแรงเคลื่อนไฟฟ้าที่เกิดขึ้น



รูปที่ 1.2.3 อุณหภูมิช่วงกลางจะไม่มีผลต่อการเกิดแรงเคลื่อนไฟฟ้า

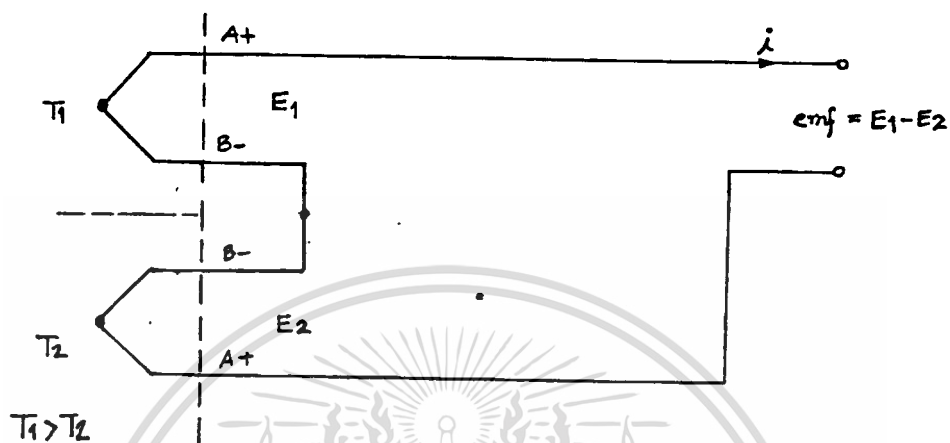
(Temperature gradients do not generate emf's in a homogeneous)

2. การต่อขนาดเทอร์โมคัปเปิลเข้าด้วยกัน ผลที่ได้คืออุณหภูมิเฉลี่ย



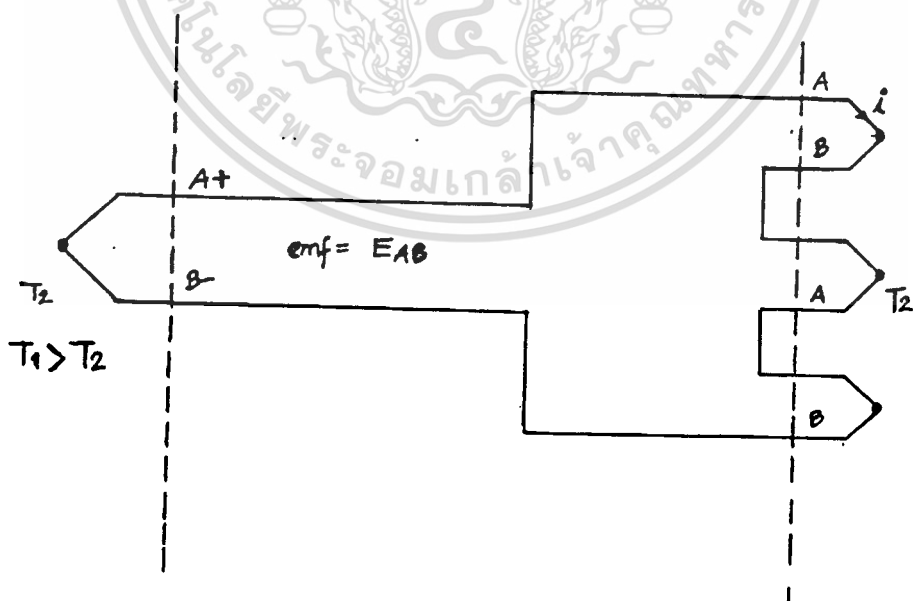
รูปที่ 1.2.4 แสดงการขนานเทอร์โมคัปเปิลเป็นการหาอุณหภูมิเฉลี่ย

3. การต่อเทอร์โมคัปเปิลสองชุดกลับหัวกันเป็นการวัดผลต่างของอุณหภูมิ



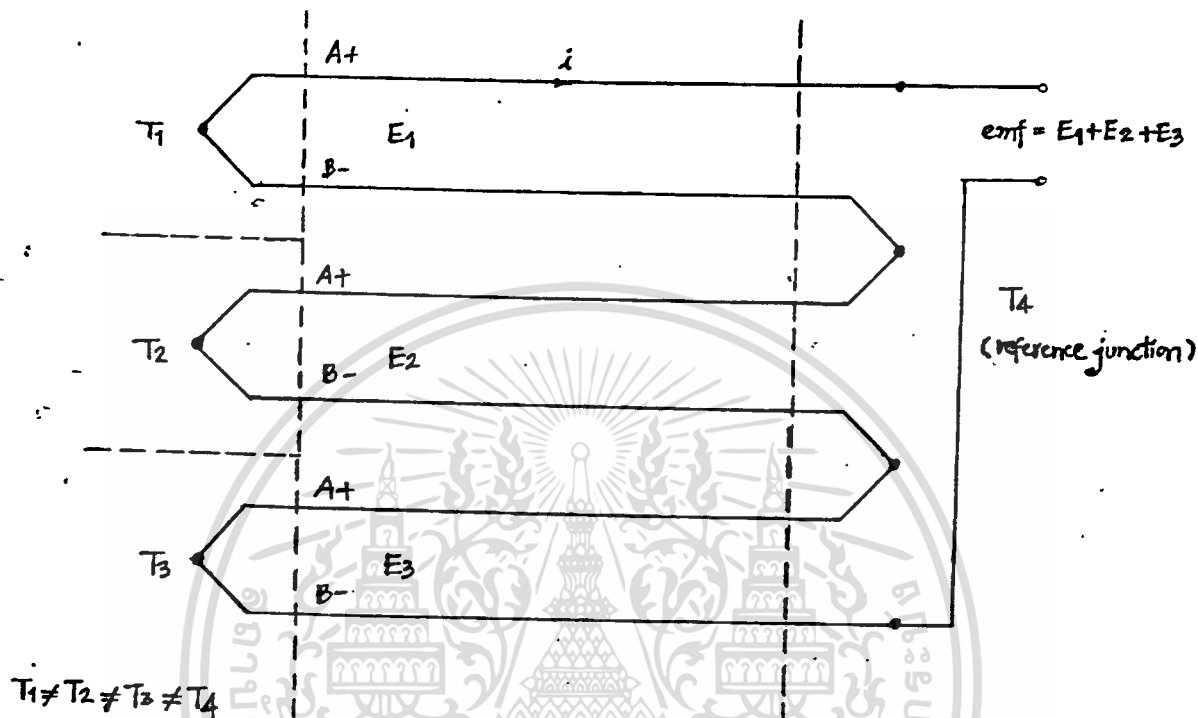
รูปที่ 1.2.5 แสดงการหาผลต่างของอุณหภูมิที่จุด T_1 และ T_2

4. การเพิ่มเทอร์โมคัปเปิลที่ T_1 หรือ T_2 ตามรูปจะไม่มีผลต่อแรงเคลื่อนที่เกิดขึ้น



รูปที่ 1.2.6 การต่อเทอร์โมคัปเปิลเพิ่มไม่มีผลต่อแรงเคลื่อนที่เกิดขึ้น

5. การต่ออนุกรมเทอร์โมคัปเปิล แรงเคลื่อนที่เกิดขึ้นจะเป็นผลรวมของเทอร์โมคัปเปิลแต่ละชุด



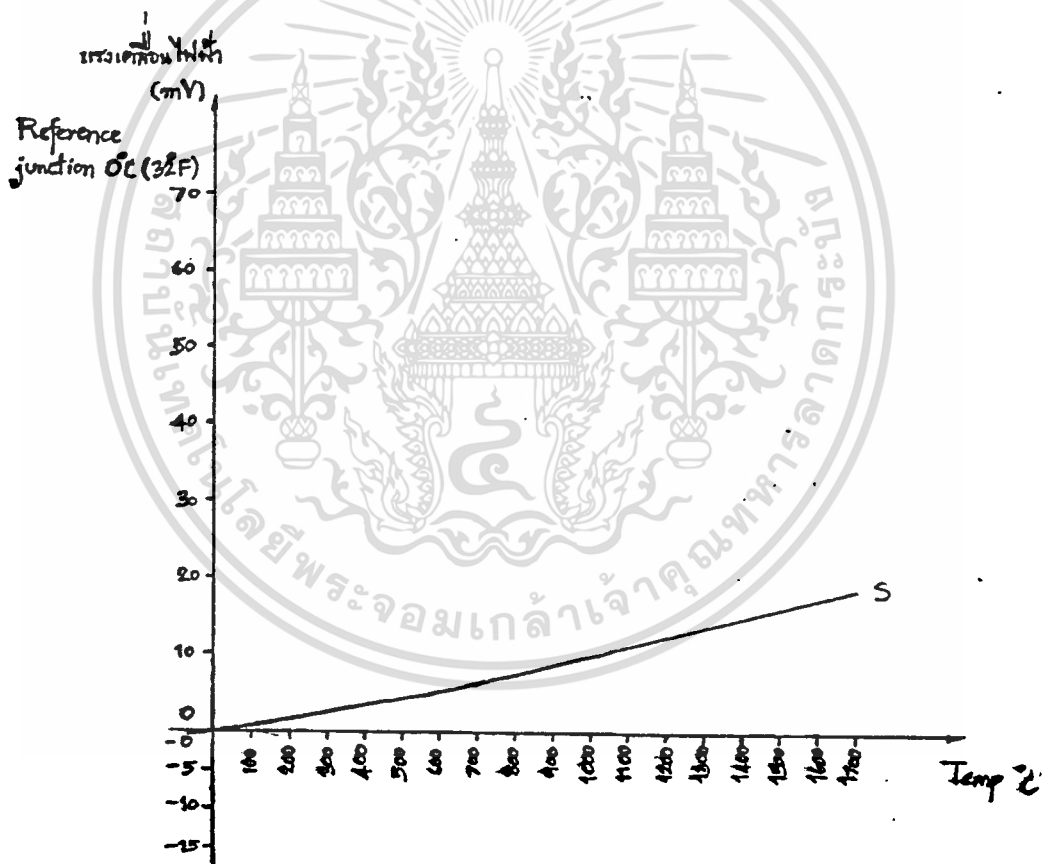
รูปที่ 1.2.7 การต่อเทอร์โมคัปเปิล emf ที่เกิดขึ้นเป็นผลรวมของเทอร์โมคัปเปิลแต่ละชุด

เทอร์โมคัปเปิลแบบมาตรฐาน

นับตั้งแต่ที่เบ็คได้ค้นพบหลักการวัดอุณหภูมิด้วยเทอร์โมคัปเปิลในปี 1821 เป็นต้นมารายละเอียดในหลักการได้ถูกพัฒนาให้ก้าวหน้าเหมาะสมที่จะนำมาใช้ในงานอุตสาหกรรมมาโดยตลอดและได้เกิดเทอร์โมคัปเปิลแบบมาตรฐานขึ้นหลายชนิด เพื่อให้เหมาะสมกับการใช้งานในลักษณะต่างๆ

เทอร์โมคัปเปิลแบบ S (type S platinum 10% rhodium v.s. platinum)

ในปี 1886 le chatelier ได้ประดิษฐ์เทอร์โมคัปเปิลโดยสายลวดทำจากแพลตินัมและสายลวดทำจากโลหะผสม 90% ของแพลตินัม + 10% ของโคบอลต์ สามารถวัดอุณหภูมิได้สูงถึง 1400 °C ซึ่งต่อมาเทอร์โมคัปเปิลแบบนี้ได้กลายเป็นแบบมาตรฐานสากลตาม IPTS 68 ระบุว่าสามารถใช้ในการสอบเทียบค่าและเปรียบเทียบเป็นค่ามาตรฐาน (calibration and comparison) ตั้งแต่จุดแข็งตัวของแอนติโมนี (630.74 °C) จนถึงจุดแข็งตัวของทอง (1064.43 °C) ตามกราฟแสดงรูปที่ 1.2.8 คุณสมบัติของเทอร์โมคัปเปิลแบบ S



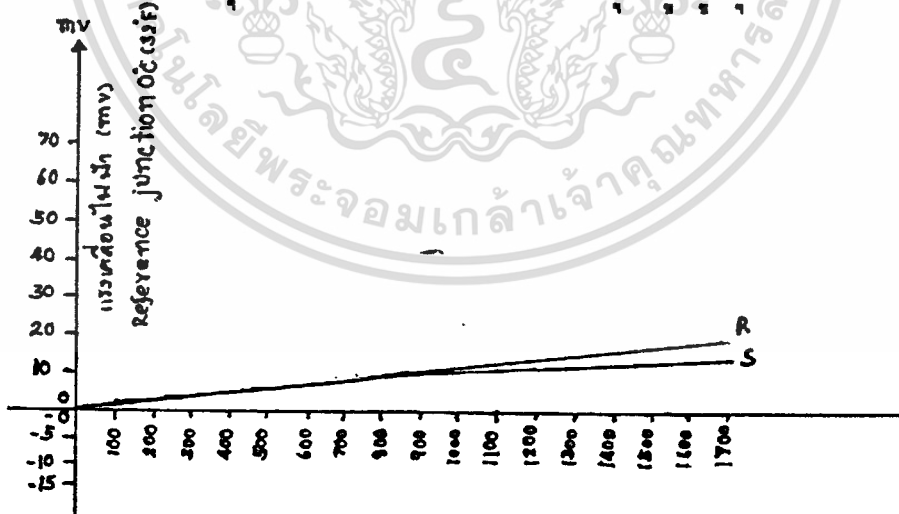
รูปที่ 1.2.8 กราฟแสดงความสัมพันธ์ของแรงเคลื่อนไฟฟ้ากับอุณหภูมิของเทอร์โมคัปเปิลแบบ S



เทอร์โมคัปเปิลแบบ ๕ นี้สามารถใช้งานในสภาวะที่เป็น oxidizing และ inert ได้ดีโดยสามารถทนอุณหภูมิได้ถึง 1400 °C หรือกับการใช้งานในระยะเวลาด้านๆ สามารถทนได้ถึง 1482 °C แต่ไม่เหมาะสำหรับสภาวะงานที่เป็นแบบ reducing, vacuum หรือสภาวะงานที่มีไอของโลหะ เช่น ตะกั่ว สังกะสี และ ไอของอโลหะ เช่น อาเซอิตซัลเฟอร์ ฟอสฟอรัสซึ่งจะทำให้มีอายุการใช้งานสั้นลงถ้าจำเป็นต้องใช้ จะต้องป้องกันด้วย protecting tube ที่เป็นแบบอโลหะ เช่น อลูมินาบริสุทธิ์ที่อุณหภูมิสูง ๆ เม็ดเกรนของพลาสติกัมจะพองตัวและพลาสติกัมก็จะเกิดสกรปรก (contamination) ได้ง่ายที่อุณหภูมิสูง ๆ ทำให้แรงเคลื่อนไฟฟ้ามีค่าต่ำลงจากการ วิเคราะห์ ส่วนผสม (composition) ภายหลังการใช้งาน 20 ปี ส่วนผสมของโรเดียมจะเปลี่ยนสภาพเป็นพลาเดียม ทำให้มีคุณสมบัติผิดไปการเปลี่ยนสภาพเช่นนี้ จะเกิดกับ เทอร์โมคัปเปิลทุกแบบที่มีส่วนผสมของโรเดียม

เทอร์โมคัปเปิลแบบ R (type r platinum 13% rhodium v.s. platinum)

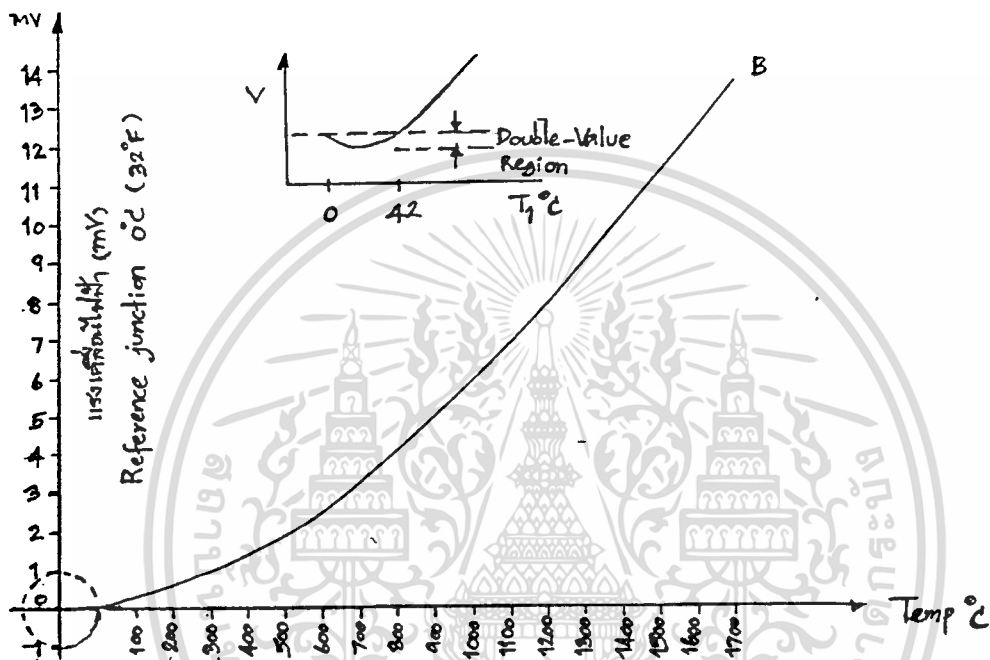
เทอร์โมคัปเปิลแบบ R ให้เอาต์พุตสูงกว่าแบบ ๕ ตามกราฟแสดงคุณสมบัติ เปรียบเทียบกับแบบ ๕ คุณสมบัติเหมือนกันแบบ ๕ ทนอุณหภูมิสูงสุดได้ 1400 °C



รูปที่ 1.2.9 กราฟแสดงความสัมพันธ์ของแรงเคลื่อนไฟฟ้ากับอุณหภูมิของเทอร์โมคัปเปิลแบบ R

เทอร์โมคัปเปิลแบบ B (type b platinum 30% rhodium/platinum 6% rhodium)

ผลิตขึ้นครั้งแรกเมื่อปี 1954 ในประเทศเยอรมันสายบวกทำจากแพลตินัม 70% + โรเดียม 6% คุณสมบัติของเทอร์โมคัปเปิลแบบ b ตามกราฟที่แสดง

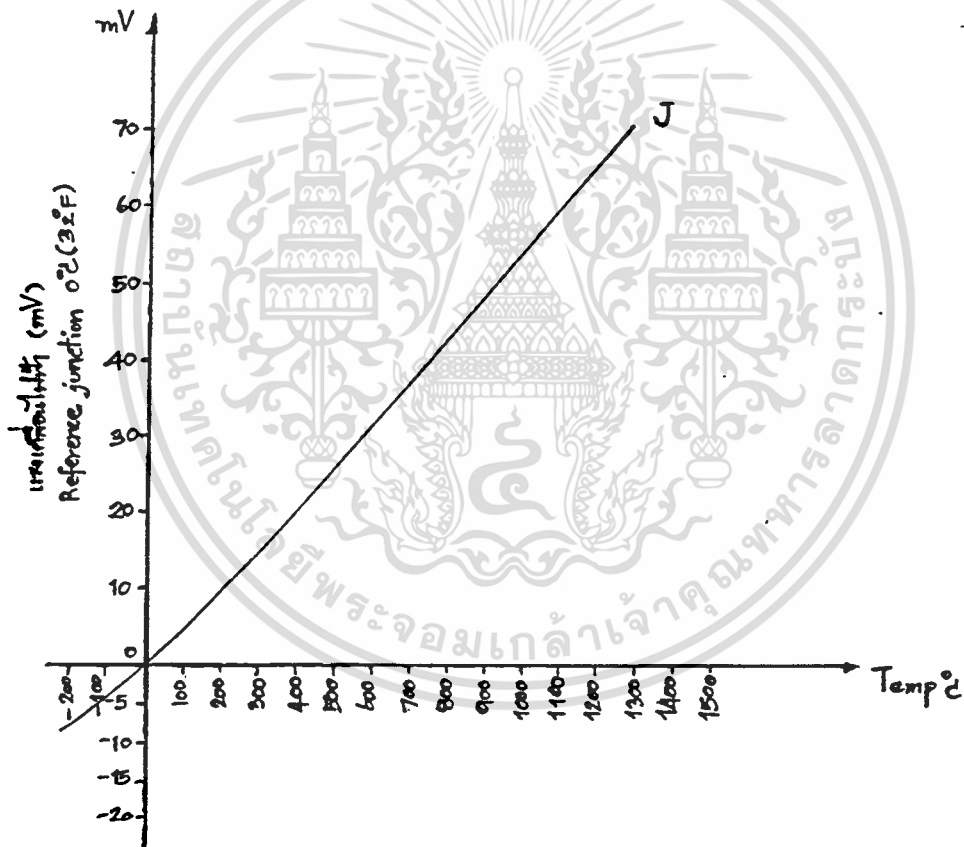


รูปที่ 1.2.10 กราฟแสดงความสัมพันธ์ของแรงเคลื่อนไฟฟ้ากับอุณหภูมิของเทอร์โมคัปเปิลแบบ b

เทอร์โมคัปเปิลแบบ b จะให้แรงเคลื่อนต่ำกว่า แบบ e. แบบ R แต่คุณสมบัติที่เด่นกว่าคือแข็งแรงและทนกว่าสามารถใช้งานที่อุณหภูมิสูงสุดได้ถึง 1704 °C (3100 °F) ในสภาวะที่เป็น oxidizing หรือ inert แต่ไม่เหมาะสมกับการใช้งานในสภาวะ reducing หรือ vacuum และใช้งานที่มีไอของโลหะและโลหะเช่นเดียวกับแบบ R และแบบ e

เทอร์โมคัปเปิลแบบ J (type j iron v.s. constantan)

เนื่องจากพลาทินัมเป็นธาตุที่มีราคาแพง เพื่อที่จะทำให้เทอร์โมคัปเปิลมีราคาถูกลงได้มีการค้นคว้าหาวัสดุที่มีราคาถูกกว่า เพื่อใช้แทนพลาทินัมวัสดุที่เริ่มทดลองใช้ เช่น เหล็ก นิเกิล นิเกิลคริสทิตี เปราะมากในสภาพงานที่เป็น oxidizing การทดลองต่อมาพบว่าโลหะผสมระหว่าง 60% ของทองแดง + 40% ของนิเกิล ที่ต่อมาเรียกว่า constantan สามารถแก้ปัญหานี้ได้เทอร์โมคัปเปิลที่ใช้สายบวกทำด้วยเหล็ก และสายลบทำด้วย constantan จึงถือกำเนิดขึ้นและต่อมากลายเป็นแบบมาตรฐาน แบบ J

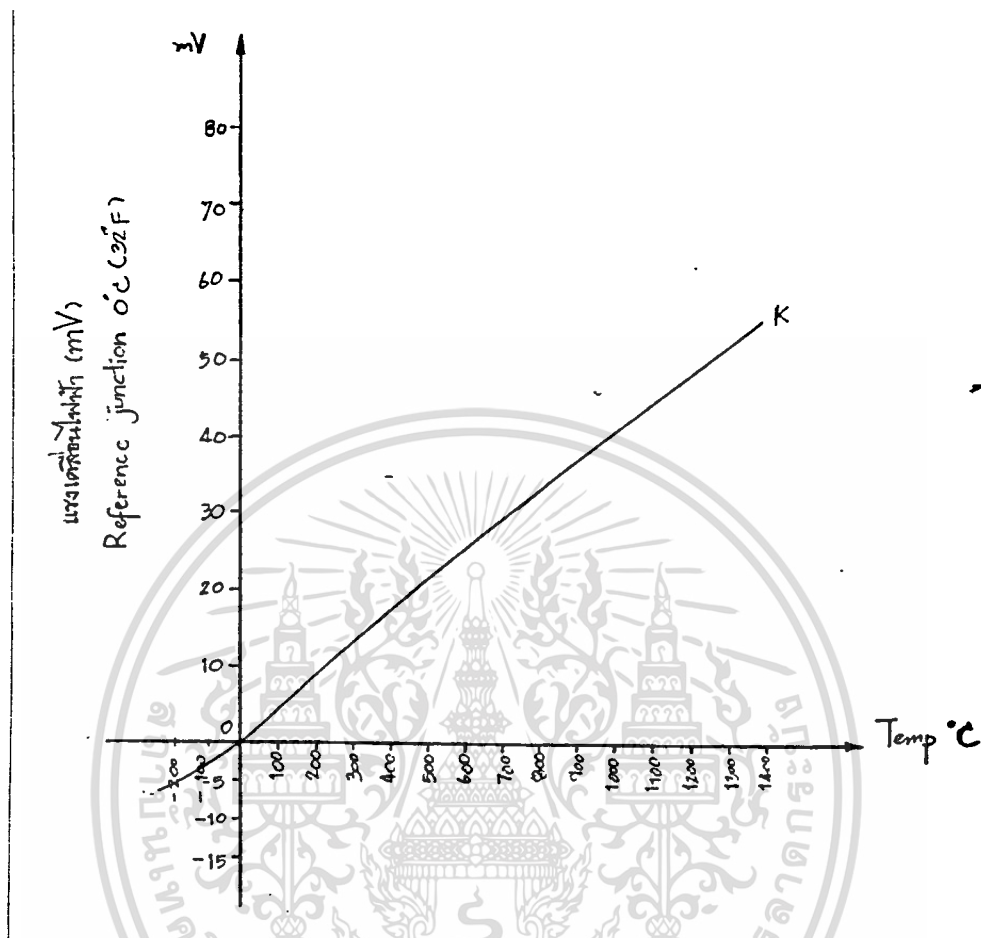


รูปที่ 1.2.11 กราฟแสดงความสัมพันธ์ของแรงเคลื่อนไฟฟ้ากับอุณหภูมิของเทอร์โมคัปเปิลแบบ J

คุณสมบัติของเทอร์โมคัปเปิลแบบ J เหมาะสำหรับสภาพงานที่เป็น vacuum, oxidizing, reducing หรือ inert ที่อุณหภูมิไม่เกิน 760 °C ไม่เหมาะสมสำหรับงานที่มีอุณหภูมิต่ำกว่า 0 °C และที่อุณหภูมิสูงกว่า 538 °C (1000 °F) สายที่เป็นเหล็กจะเกิด oxidization ด้วยอัตราที่สูงกว่าปกติมาก สำหรับอุณหภูมิที่สูงกว่า 538 °C จะต้องใช้สายเทอร์โมคัปเปิลขนาดใหญ่จะช่วยให้อายุการใช้งานยืนยาวขึ้น จากการทดลองใช้งานภายใน 20 ปีพบว่าส่วนผสมของโลหะเทอร์โมคัปเปิลจะเปลี่ยนไป 0.5% (แมงกานีส เพิ่มขึ้นในเนื้อเหล็ก)

เทอร์โมคัปเปิลแบบ K (type K chromel v.s. alumel)

เพื่อที่จะทำให้เทอร์โมคัปเปิลสามารถวัดอุณหภูมิได้สูงกว่าแบบ J และมีราคาถูกกว่าจึงได้มีผู้ประดิษฐ์เทอร์โมคัปเปิลแบบใหม่ที่ สายบวกทำจากโลหะผสมระหว่าง นิกเกิล 10% + โครเมียม 90% และสายลบทำจากโลหะผสมระหว่าง นิกเกิล 95% + 5% ของส่วนผสมระหว่างอลูมิเนียม, แมงกานีสและซิลิกอน ซึ่งต่อมาได้กลายเป็นแบบมาตรฐานแบบ K เป็นที่นิยมกันอย่างแพร่หลายมากที่สุดสามารถใช้กับสภาวะงานที่เป็น oxidizing หรือ inert ได้ดีกว่าแบบอื่นสามารถทนอุณหภูมิได้ถึง 1260 °C (2300 °F) และที่อุณหภูมิต่ำถึง -250 °C (429 °F) สภาพงานที่ต้องรับการแผ่รังสีโดยตรงจากแหล่งกำเนิดความร้อนแบบ K ก็สามารถใช้งานได้ดีเช่นกัน



รูปที่ 1.2.12 กราฟแสดงความสัมพันธ์ของแรงเคลื่อนไฟฟ้ากับอุณหภูมิของเทอร์โมคัปเปิลแบบ k

คุณสมบัติที่เด่นมากของเทอร์โมคัปเปิลแบบ k คือให้แรงเคลื่อนเอาต์พุตสูงกว่าแบบอื่นๆ

ข้อควรระวัง ไม่ควรใช้เทอร์โมคัปเปิลแบบ k สัมผัสโดยตรงกับ

1. ในสภาวะงานที่เป็น reducing หรือเป็นการใช้งานสลับกันระหว่าง oxidizing และ reducing

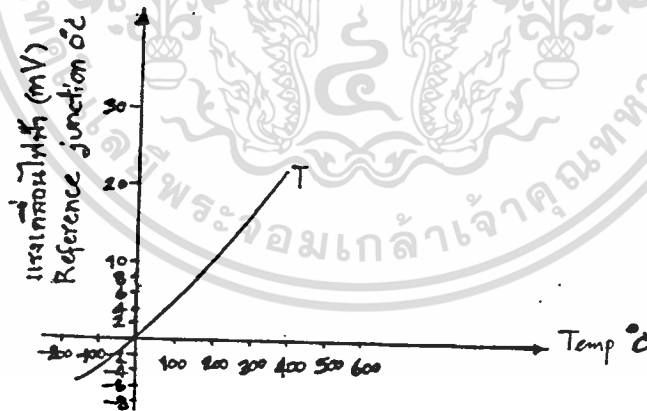
2. สภาวะงานที่มีไอของซิลิเฟอหรือเพราซซิลิเฟอจะทำลายโลหะ ทั้งคู่ของเทอร์โมคัปเปิลโดยเฉพาะสายลบจะเกิดขึ้นอย่างรวดเร็ว

3. สภาวะงานที่เป็น vacuum ยกเว้นในช่วงระยะเวลาดสั้น ๆ

ถ้าจำเป็นต้องใช้งานเทอร์โมคัปเปิลกับสภาพดังกล่าวนี้จะต้องใช้ protecting tube ป้องกันเสียก่อนอายุการใช้งานของเทอร์โมคัปเปิลแบบ K นี้ประมาณ 20 ปี เพราะหลังจากนั้นส่วนผสมจะเปลี่ยนไปโดยจะมีเหล็กเพิ่มขึ้นประมาณ 2% และส่วนผสมบางอย่าง เช่น แมงกานีส จะลดปริมาณลง

เทอร์โมคัปเปิลแบบ T (type T copper v.s. constantan)

เป็นเทอร์โมคัปเปิลแบบ T ทำจากทองแดง และสายลวดทำมาจาก constantan ในสภาพการใช้งานปกติสามารถวัดอุณหภูมิได้ต่ำถึง -184°C แต่อุณหภูมิทางขั้ววัดได้ต่ำกว่าแบบอื่นๆ คือ ประมาณ 370°C เท่านั้น เพราะที่อุณหภูมิสูงกว่า 370°C อัตราการเกิด oxide ของโลหะเทอร์โมคัปเปิลจะเพิ่มมากขึ้นเป็นแบบที่ทนการกัดกร่อนในบรรยากาศที่มีความชื้นได้ดีเป็นพิเศษ และสามารถใช้งานในสภาวะที่เป็น vacuum, oxidizing หรือ inert ได้ดี

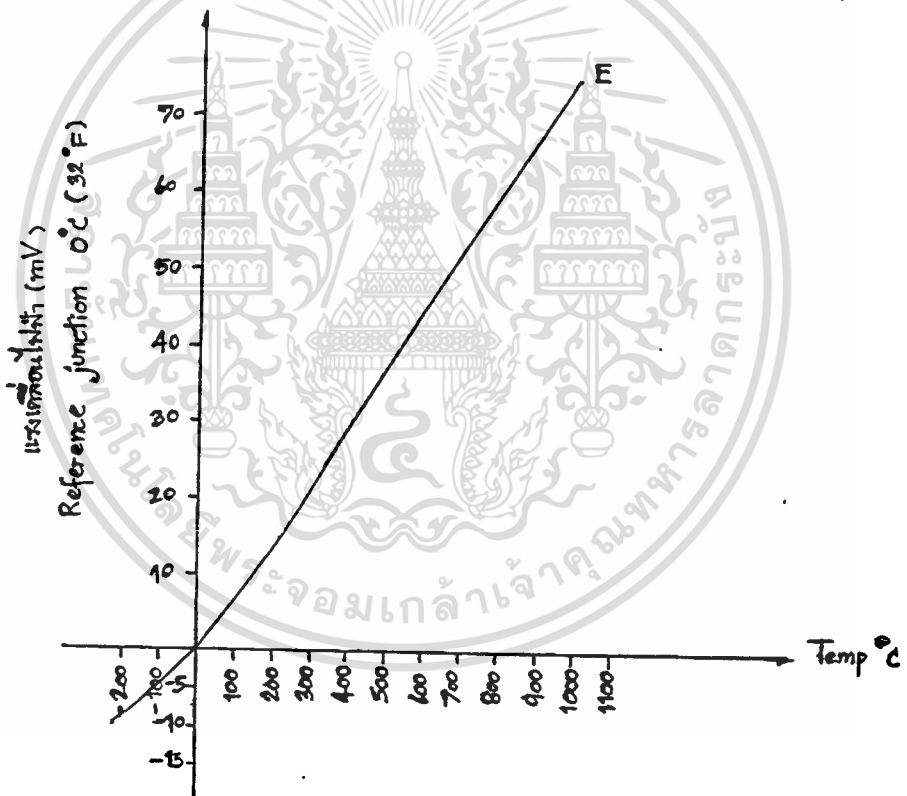


รูปที่ 1.2.13 กราฟแสดงความสัมพันธ์ของแรงเคลื่อนไฟฟ้ากับอุณหภูมิของเทอร์โมคัปเปิลแบบ T

การใช้งานที่ต้องสัมผัสการแผ่รังสีโดยตรงจะทำให้ส่วนผสมของเทอร์โมคัปเปิลเปลี่ยนไปได้จึงไม่เหมาะสมกับงานทดลองใช้งานใน 20 ปี ส่วนผสมของนิเกิล และ สังกะสี จะเพิ่มขึ้นประมาณ 10%

เทอร์โมคัปเปิลแบบ E (type E chromel v.s.s constant)

สายลวดทำจาก constantan อุณหภูมิใช้งานปกติอยู่ระหว่าง -250°C หรือ 87°C เหมาะสมกับสภาพงานที่เป็น oxidizing คุณสมบัติด้านอื่น ๆ นั้นคล้ายกับ เทอร์โมคัปเปิลแบบ K



รูปที่ 1.2.14 กราฟแสดงความสัมพันธ์ของแรงเคลื่อนไฟฟ้ากับอุณหภูมิของเทอร์โมคัปเปิลแบบ E

แบบของเทอร์โมคัปเปิลที่กล่าวมานี้เป็นแบบมาตรฐานแต่ในปัจจุบันได้มีการทดลองนำโลหะอีกหลายชนิด เช่น ทังสเตน ริเนียม (rhenium) เออิเดียม (iridium) มาประดิษฐ์เทอร์โมคัปเปิลแบบใหม่ เพื่อให้ได้คุณสมบัติที่ต่างไปจากแบบมาตรฐานที่มีอยู่ซึ่งจะได้รับการจัดเข้าเป็นแบบมาตรฐานในอนาคตต่อไป

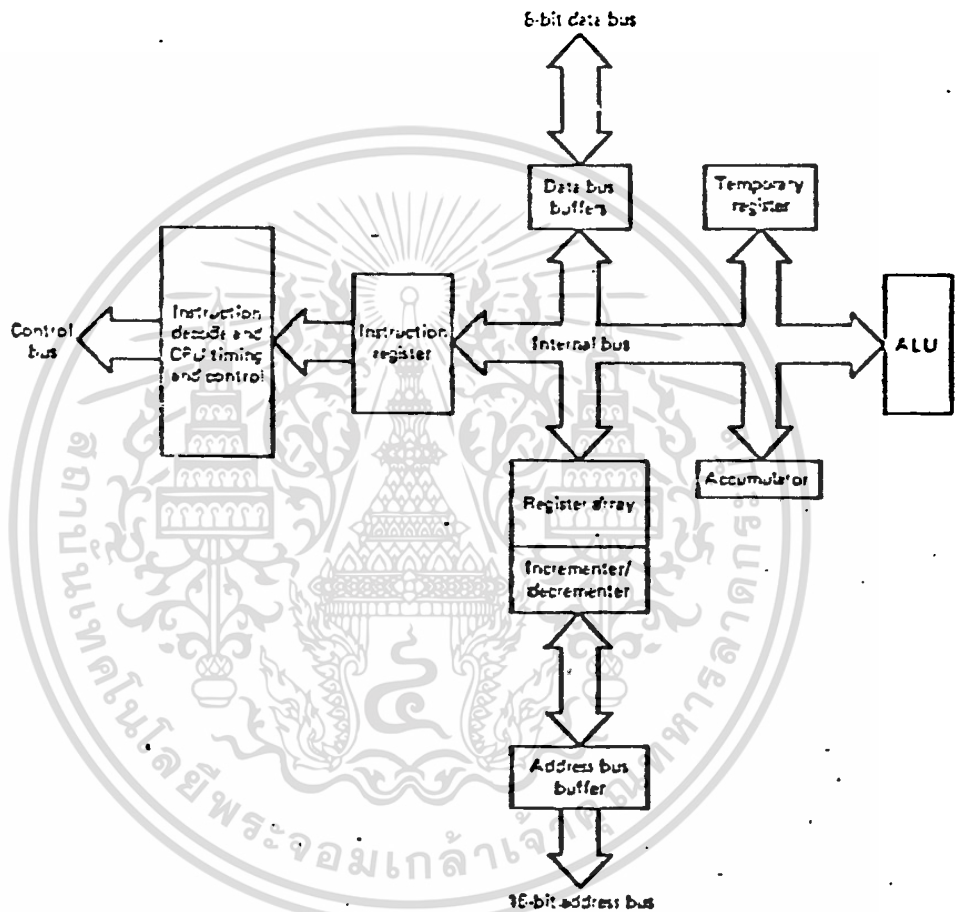


ไมโครโปรเซสเซอร์ Z-80 ซึ่งผลิตโดย บริษัท ซิล็อก (Zilog) ทำงานด้วยบัสข้อมูลขนาด 8 บิต และบัสตำแหน่งขนาด 16 บิต ต้องการแหล่งจ่ายไฟเลี้ยงงานไฟฟ้าระดับเดียวกับ +5 โวลต์ ลักษณะงานนี้ก็เพียงลักษณะเดียว ชุดคำสั่งของ Z-80 จะรวมคำสั่งทั้งหมดของ 8080 ไว้ด้วย ตลอดจนมีคำสั่งอื่น ๆ ที่ 8080 ไม่มีรวม 158 คำสั่ง ทำให้คำสั่ง Z-80 มีสมรรถนะเหนือคำสั่งของ 8080 โปรแกรมที่เขียนให้ทำงานกับ 8080 จึงสามารถนำมาใช้กับ Z-80 ได้ แต่ในทางกลับกันโปรแกรมที่เขียนให้ทำงานกับ Z-80 อาจนำมาใช้ 8080 ได้ หรือไม่ได้อีกแล้วแต่คำสั่งที่ปรากฏในโปรแกรมนั้น ๆ

2.1 โครงสร้างของไมโครโปรเซสเซอร์ Z-80

โครงสร้างของไมโครโปรเซสเซอร์ Z-80 มีโครงสร้างที่พัฒนามาจาก 8080 ดังนั้นในแง่โครงสร้างพื้นฐานจะเหมือนกับไมโครโปรเซสเซอร์ของ 8080 แต่เนื่องจาก Z-80 มีการพัฒนามามากขึ้นทางซอฟต์แวร์และฮาร์ดแวร์ จึงทำให้มีรายละเอียดแตกต่างเพิ่มเติมอีกหลายประการด้วยกัน ซิล็อกได้ออกแตรัมรูปที่ 2.1 เป็นไดอะแกรม แสดงให้เห็นโครงสร้างภายในของ Z-80 โดยโครงสร้างของไมโครโปรเซสเซอร์นี้ จะบรรจุลงใน แอลเอสไอ ขนาด 40 ขา

โครงสร้างภายในของไมโครโปรเซสเซอร์ Z-80 ประกอบด้วยรีจิสเตอร์ภายในที่สามารถเขียนและอ่านได้ถึง 208 บิต โดยแยกเป็นกลุ่มของรีจิสเตอร์ขนาด 8 บิต 18 รีจิสเตอร์ และรีจิสเตอร์ขนาด 16 บิต 4 รีจิสเตอร์ โดยมีชุดรีจิสเตอร์แสดงได้ดังรูปที่ 2.2



รูปที่ 2.1 บล็อกไดอะแกรม ไมโครโปรเซสเซอร์ Z-80

รีจิสเตอร์หลัก		รีจิสเตอร์สำรอง		แอดเดรส เลเตอร์ และ แฟล็ก
A	F	A'	F'	
B	C	B'	C'	รีจิสเตอร์ใช้งาน ทั่วไป.
D	E	D'	E'	
H	L	H'	L'	

I	R
IX	
IY	
SP	
PC	

รีจิสเตอร์ใช้งาน
เฉพาะอย่าง.

รูปที่ 2.2 แสดงรีจิสเตอร์ต่าง ๆ ที่มีอยู่ในไมโครโปรเซสเซอร์ Z-80

2.2 รีจิสเตอร์หลักที่ใช้งานทั่วไป

รีจิสเตอร์ในกลุ่มแรกคือ A, F, B, C, D, E, H, L เป็นรีจิสเตอร์ขนาด 8 บิต ที่ใช้งานทั่วไป โดยรีจิสเตอร์เหล่านี้ สามารถประกอบรวมกัน เป็นรีจิสเตอร์คู่ได้คือ AF, BC, DE, HL โดยคู่รีจิสเตอร์เหล่านี้ จะได้รับการใช้งานในลักษณะของรีจิสเตอร์ขนาด 16 บิต การกระทำภายในไมโครโปรเซสเซอร์อาจจะอาศัยเพียงรีจิสเตอร์เดี่ยวกระทำเป็นรีจิสเตอร์คู่ได้ โดยที่ A คือ แอดเดรส เลเตอร์ F คือ แฟล็ก แฟล็ก ของ Z-80 จะมีด้วยกัน 6 ตัว จึงใช้เพียง 6 บิตแต่ Z-80 อาศัยการเพิ่มบิตขึ้นอีก

2 บิต และกลายเป็นรีจิสเตอร์ F รีจิสเตอร์ F นี้สามารถได้รับการเซต รีเซต กระทำตามคำสั่งทางคณิตศาสตร์หรือลอจิกได้และเราสามารถรีจิสเตอร์ F เหมือน รีจิสเตอร์หนึ่ง ซึ่งเมื่อรวมกับกับ A แล้วจะกลายเป็นรีจิสเตอร์ 16 บิตได้

2.3 กลุ่มรีจิสเตอร์สำรอง

เป็นกลุ่มรีจิสเตอร์ที่สามารถเก็บข้อมูลได้ โดยเป็นตัวเก็บข้อมูลที่มาจากรีจิสเตอร์หลัก ซึ่งรีจิสเตอร์ชุดนี้จึงมีด้วยกัน 8 ตัวคือ A , F , B , C , D , E , H , L รีจิสเตอร์เหล่านี้เป็นรีจิสเตอร์ที่ใช้ในการเก็บข้อมูลชั่วคราว ในการที่ต้องการใช้ รีจิสเตอร์หลักทำงานอย่างอื่นก่อน ดังนั้นรีจิสเตอร์กลุ่มนี้จึงไม่สามารถกระทำ ทางคณิตศาสตร์และลอจิกได้

2.4 กลุ่มรีจิสเตอร์ที่ใช้งานเฉพาะอย่าง

โปรแกรมเคาน์เตอร์ (PC-Program counter) โปรแกรมเคาน์เตอร์ เป็นรีจิสเตอร์ขนาด 16 บิตที่เป็นตัวกำหนดตำแหน่งของโปรแกรมในขณะสภาวะการกระทำการเฟตซ์ โดยขณะทำการเฟตซ์ค่าที่อยู่ในโปรแกรมเคาน์เตอร์ จะไปปรากฏอยู่ที่แอดเดรสบัส เพื่อชี้ไปยังตำแหน่งในหน่วยความจำให้ CPU อ่านคำสั่งมาตีความหมาย ค่าที่อยู่ในโปรแกรมเคาน์เตอร์จะเพิ่มค่าขึ้นได้อย่างอัตโนมัติ หลังการกระทำ การเฟตซ์แต่ถ้าหาก CPU กระทำคำสั่งให้ข้ามไปยังตำแหน่งอื่นค่าแอดเดรสที่จะกระโดดข้ามไปนั้นจะไหลตเข้ามายังโปรแกรมเคาน์เตอร์ได้อย่างอัตโนมัติ

สแตคพอยน์เตอร์ (SP-Stack pointer) เป็นรีจิสเตอร์ที่มีขนาด 16 บิตที่ใช้สำหรับชี้ ไปยังแอดเดรสขั้นบนสุดของสแตคที่อยู่ใน RAM โดยส่วนของสแตคมีลักษณะ โครงสร้างเป็นหน่วยความจำเป็นแบบเก็บที่หลังเรียกออกได้ก่อน ข้อมูลในสแตคอาจ ได้รับ การpush หรือ pop มาจากข้อมูลรีจิสเตอร์ภายใน CPU ลักษณะของสแตคในที่นี้ยังเป็นส่วนช่วยในการกระทำอินเตอร์รัพท์ และการเรียกโปรแกรมย่อย กล่าวคือ

ในการอินเทอร์รัพท์ค่าของโปรแกรมเคาน์เตอร์จะได้รับการเก็บรักษาไว้ในชั้นสแตค
ครั้นเมื่อโปรแกรมกลับจากการอินเทอร์รัพท์ไปกระทำยังโปรแกรมหลักก็จะนำค่าจาก
สแตคกลับเข้าไปยังโปรแกรมเคาน์เตอร์ใหม่ในทำนองเดียวกัน การกระโดดไปกระทำ
โปรแกรมย่อยก็เช่นเดียวกัน ดังนั้นการกระทำในรูปของอินเทอร์รัพท์ หรือโปร
แกรมย่อย สามารถซ้อนกันได้ไม่มีที่สิ้นสุด

อินเดครีจิสเตอร์ (IX, IY-index register) CPU Z-80 มีอินเดครีจิสเตอร์
ขนาด 16 บิต 2 ตัว แต่ละตัวใช้ประโยชน์หลักในการทำหน้าที่เป็นตัวเก็บแอดเดรส
ฐาน (base address) เพื่อทำหน้าที่อ้างแอดเดรสแบบอินเดคแอดเดรสซิง (index
-addressing) ในโหมดของอินเดคแอดเดรส ซึ่งมีข้อมูลที่อยู่ในอินเดครีจิสเตอร์นี้
จะร่วมกับข้อมูลที่ติดมากับคำสั่งอีก 8 บิต เพื่อเป็นตัวกำหนดแอดเดรสให้กับคำสั่ง
ข้อมูลที่ติดมากับคำสั่งนี้เราเรียกว่า ดิสเพลซเมนต์ (displacement) ซึ่งจะเก็บใน
รูปของตัวเลข 2^๘ คอมพลีเมนต์

อินเทอร์รัพท์แอดเดรสรีจิสเตอร์ (I-Interrupt base address
register) การอินเทอร์รัพท์ของ Z-80 มีหลายโหมดและโหมดหนึ่งที่ทำให้การอิน
เทอร์รัพท์ของ Z-80 มีประสิทธิภาพสูงกล่าวคือเมื่อเกิดการอินเทอร์รัพท์ในโหมดนี้ขึ้น
มันสามารถอ้างแอดเดรส โดยทางอ้อมไปกระทำโปรแกรมในที่ใดก็ได้ในหน่วยความ
จำ โดยอาศัยค่าในรีจิสเตอร์ I รวมกับค่าที่ส่งมาจากอุปกรณ์เพอร์เฟอรัลอีก 8 บิต
ซึ่งไปยังค่าในหน่วยความจำเพื่อค่านั้นมาไหลตเข้าไปในโปรแกรมเคาน์เตอร์เพื่อกระ
ทำต่อไปด้วยวิธีการนี้เราจึงสามารถกระโดดเข้าไปทำที่ส่วนใดก็ได้ในหน่วยความจำ

รีจิสเตอร์รีเฟรชหน่วยความจำ (R-memory refresh register) การ
ต่อ CPU เข้ากับหน่วยความจำนั้น โดยปกติจะต่อกับหน่วยความจำชนิดสแตติกได้โดย
ง่าย แต่อย่างไรก็ดีชนิดไดนามิกที่ต้องการรีเฟรชมีราคาสูงกว่า มีความหนาแน่นสูง

กว่า Z-80 ให้ข้อดีกว่าประการหนึ่งคือมันสามารถให้การรีเฟรชต่อหน่วยความจำได้อย่างอัตโนมัติ โดยค่าใน R รีจิสเตอร์จะเพิ่มค่าขึ้นอีก 1 ทุกครั้งที่มีการกระทำการเฟรชคำสั่ง และข้อมูลในรีจิสเตอร์ R นี้จะส่งออกไปยังแอดเดรสบัลในส่วนบิตที่มีนัยสำคัญต่ำกว่า จังหวะของการส่งนี้จะเป็นจังหวะเดียวกันกับที่ CPU สัญญาแบริเฟรชออกมาผู้โปรแกรมสามารถกำหนดค่าให้กับรีจิสเตอร์ R นี้ได้ แต่ค่าในรีจิสเตอร์นี้จะเรียกใช้โดยผู้โปรแกรมทางคำสั่งโดยตรงไม่ได้

แอคคิวมูเลเตอร์ รีจิสเตอร์ (A-Accumulator Register) แอคคิวมูเลเตอร์เป็นรีจิสเตอร์ที่มีขนาด 8 บิต ที่สำคัญของ CPU Z-80 เพราะว่าจะใช้แอคคิวมูเลเตอร์รีจิสเตอร์ในการเก็บผลลัพธ์ที่ได้จากการคำนวณ และการกระทำทางลอจิก (Logic)

แฟลกรีจิสเตอร์ (F-Flag register) เป็นรีจิสเตอร์ ที่แสดงสถานะภาพของเงื่อนไขในการกระทำในส่วนของหน่วยคณิตศาสตร์และลอจิก เช่น เงื่อนไขผลลัพธ์เป็นศูนย์ ผลลัพธ์เป็นบวกหรือลบ มีตัวทดหรือตัวขอยืมในการกระทำทางคณิตศาสตร์ แสดงเงื่อนไขพาริตีคี่หรือคี่ ฯลฯ แฟลกเป็นรีจิสเตอร์ ขนาด 8 บิต ซึ่งสามารถรวมกับแอคคิวมูเลเตอร์รีจิสเตอร์ (A) เป็นรีจิสเตอร์ขนาด 16 บิต ผู้โปรแกรมยังสามารถใช้คำสั่งในการเคลื่อนย้ายข้อมูลจากแอคคิวมูเลเตอร์ A และแฟลก F ไปเก็บไว้ใน A' และ F' ได้เพื่อทำให้การใช้งานของ A และ F มีประสิทธิภาพดียิ่งขึ้น รูปที่ 2.3 แสดงรายละเอียดของแฟลก 1 ไบท์ สังเกตว่าจะใช้งานแค่ 6 บิต ใน 8 บิตของแฟลก ซึ่ง 6 บิตที่ใช้งานมีดังนี้

Flag Register

S	Z		H		P/V	N	C
---	---	--	---	--	-----	---	---

S = Sign

Z = Zero

H = Half-carry

P/V = Parity/overflow

N = Add/Subtract

C = Carry

รูปที่ 2.3 Z-80 แฟลกริจิสเตอร์

1. บิตเครื่องหมาย (Sign flag : S) ใช้แสดงผลของการคำนวณ หรือ การกระทำทางลอจิกว่าเป็นบวกหรือลบ ถ้าเป็น ลอจิก "1" จะมีผลเป็นเครื่องหมาย ลบ และถ้าเป็นลอจิก "0" จะมีผลเป็นเครื่องหมายบวก

2. บิตศูนย์ (Zero flag : Z) แสดงผลที่ได้รับจากการกระทำของ ALU ว่าเป็นศูนย์หรือไม่เป็นศูนย์ ถ้าเป็นลอจิก "1" ในบิตนี้แสดงว่าเป็นศูนย์ และเป็นลอจิก "0" จะแสดงว่าไม่เป็นศูนย์

3. บิตตัวทศช่วย (Half carry flag : H) จะแสดงการทดที่เกิดขึ้นระหว่างบิต D_3 และ D_4 อันเป็นผลมาจาก ALU ถ้า $H = 1$ จะเกิดการทดหรือขอยืม ระหว่างบิต D_3 และ D_4

4. บิตพาริตีหรือโอเวอร์โฟลว์แฟลก (Parity/overflow flag : P/V) จะแสดงสภาวะโอเวอร์โฟลว์ของการกระทำทางคณิตศาสตร์ และสภาวะพาริตีของการกระทำทางลอจิก กรณีที่เป็นพาริตี $P/V = 1$ หมายถึงผลลัพธ์การกระทำให้ผลพาริตีคู่ ถ้า $P/V = 0$ ผลลัพธ์จะเป็น พาริตีคี่

5. บิตการลบ (Negative flag : N) เป็นการแสดงการกระทำทางคณิตศาสตร์คือการลบหรือ การบวก ถ้า $N = 1$ หมายถึงคำสั่งก่อนหน้าเป็นคำสั่งลบ

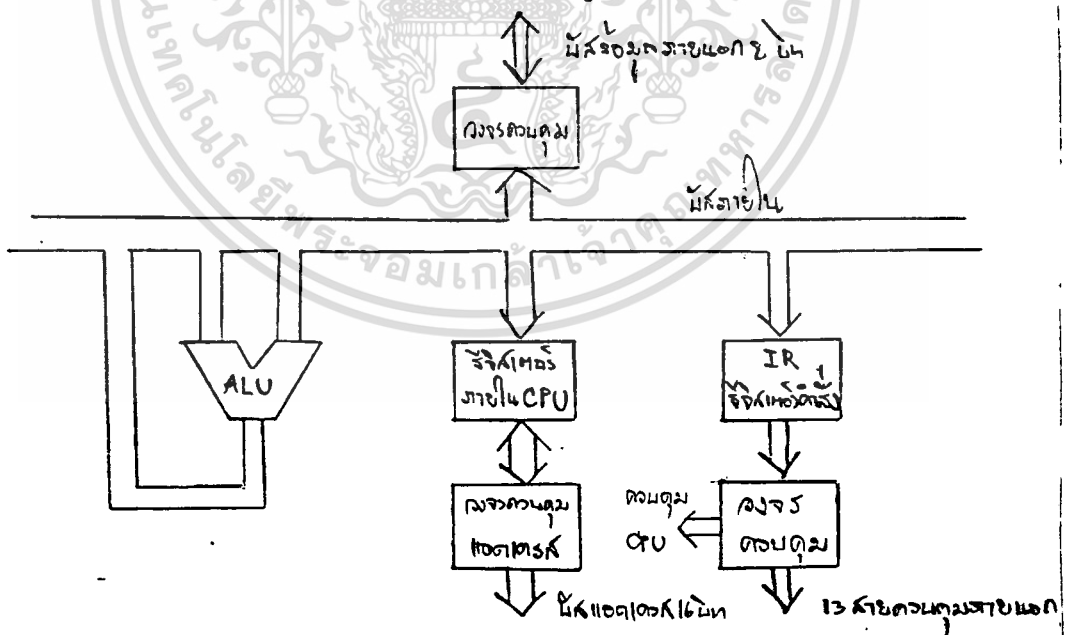
6. บิตตัวทด (Carry flag : C) จะแสดงตัวทดที่เกิดขึ้นจากบิตที่มีนัยสำคัญสูงสุดของ ALU ภายหลังจากการบวก และมีการยืมภายหลังจากการลบ

หน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU-arithmetic & logic unit)

การประมวลผลที่สำคัญของ CPU ของคอมพิวเตอร์ยังขึ้นอยู่กับหน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU) ส่วนนี้จะนำข้อมูลซึ่งอาจจะนำมาจากภายนอก CPU หรือภายใน CPU ก็ได้มาประมวลผลการประมวลผลในส่วน ALU ที่สำคัญประกอบด้วย

การบวก (add)	การเลื่อนบิตทางซ้ายหรือขวา
การลบ (subtract)	การเพิ่มค่า (increment)
ลอจิก AND	การลดค่า (decrement)
ลอจิก OR	การเซตบิต (set bit)
ลอจิก EX-OR	การรีเซตบิต (reset bit)
เปรียบเทียบ (compare)	การทดสอบบิต (test bit)

ดังได้แสดงการทำงานของ ALU ไว้ในรูปที่ 2.4



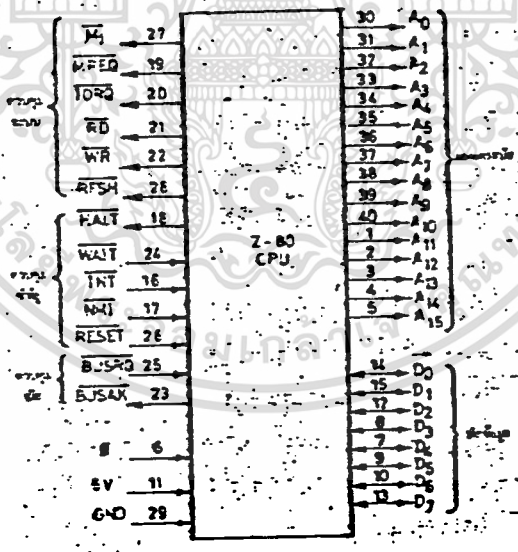
รูปที่ 2.4 แสดงการทำงานของ ALU

รีจิสเตอร์คำสั่งและส่วนควบคุม (Instruction register and control)

ในการกระทำการเฟรช CPU จะอ่านคำสั่งจากหน่วยความจำที่เป็นส่วนของโปรแกรม โดยรอ่านคำสั่งนั้นมาเก็บไว้ใน IR เพื่อทำการถอดรหัสคำสั่งและส่งสัญญาณควบคุมการทำงานภายใน CPU หรือควบคุมการทำงานของระบบ สัญญาณควบคุมเหล่านี้จะออกมาในจังหวะต่างๆ กันเพื่อใช้ควบคุมระบบในการทำงานต่อไป

2.5 การจัดขาของ Z-80

Z-80 CPU เป็น IC ไมโครโปรเซสเซอร์ที่มีขาเพียง 40 ขาโดยหลักการแล้ว Z-80 เป็น CPU ได้โดยสมบูรณ์ กล่าวคือ Z-80 ไม่ต้องประกอบกับอุปกรณ์ประกอบอื่นที่จะแยกการทำงานเพื่อรวมเป็น CPU ส่วนของสัญญาณจะประกอบด้วยบัสแอดเดรส บัสข้อมูลและสัญญาณควบคุม การจัดวางขาแสดงดังรูปที่ 2.5



รูปที่ 2.5 ลักษณะของขา ไมโครโปรเซสเซอร์ Z-80

รายละเอียดของขาต่าง ๆ แสดงได้ดังนี้

$A_{10} - A_{15}$ บัสแอดเดรส สัญญาณที่ออกมาจากขา IC เหล่านี้ จะให้แอดคิฟ ณะ high โดยขาเหล่านี้เป็นเอาต์พุตแบบไตรสเตท บัสแอดเดรสมีด้วยกัน 16 สาย เพื่อให้ CPU ติดต่อกับหน่วยความจำได้มากถึง $2^{16} = 64K$ ไบต์ นอกจากนี้ส่วนของแอดเดรสยังเป็นตัวกำหนดเบอร์พอร์ทของอุปกรณ์อินพุต-เอาต์พุตโดยขณะที่ CPU การกระทำคำสั่งเกี่ยวกับอินพุต หรือเอาต์พุตค่าของแอดเดรสบัสใน 8 บิตล่าง ($A_0 - A_7$) จะแสดง ค่าเบอร์พอร์ท ดังนั้น เราจึงมีอุปกรณ์อินพุตหรือเอาต์พุตได้ทั้งหมด $2^8 = 256$ พอร์ท และในขณะช่วงเวลารีเฟรชโดยเมื่อสัญญาณรีเฟรชปรากฏที่ขา รีเฟรช (RFSH) ค่าในแอดเดรสบัส $A_0 - A_7$ จะแสดงค่าแอดเดรสของหน่วยความจำที่จะได้รับการกระทำการรีเฟรช

$D_0 - D_7$ บัสข้อมูล (data bus) เป็นลักษณะบัสแบบสองทิศทาง Z-80 CPU มีบัสข้อมูลเส้น บัสข้อมูลเป็นเส้นทางผ่านของข้อมูลระหว่าง CPU กับหน่วยความจำ CPU กับอุปกรณ์อินพุต-เอาต์พุต หรือ การติดต่อระหว่างอุปกรณ์อินพุต-เอาต์พุตกับหน่วยความจำ

$\overline{M_1}$ (machine cycle one) มีลักษณะเป็นแอดคิฟที่ลอจิก "0" M_1 เป็นส่วนที่จะบอกให้ทราบว่าขณะนี้ CPU กำลังอยู่ในสถานะเฟรช ในขณะที่ CPU เฟรชคำสั่งที่มีอนโค็ด 2 ไบต์ ส่วนของ M_1 จะสร้างขึ้นสถานะเฟรชในแต่ละไบต์ ลักษณะของคำสั่งที่มีอนโค็ดสองไบต์ จะขึ้นต้นด้วย CBH, DDH, EDH, FDH (H ต่อท้ายหมายถึงตัวเลขฐานสิบหก) นอกจากนี้ M_1 ยังสร้างสัญญาณร่วมกับ IORQ เพื่อบอกสถานะการตอบรับการอินเตอร์รวันท์

$\overline{MREQ}$ (memory request) เป็นเอาต์พุตลักษณะไตรสเตทให้ลอจิกแอดคิฟที่ "0" เป็นสัญญาณที่บอกให้ทราบว่า CPU ต้องการเขียนหรืออ่านหน่วยความจำตามแอดเดรสที่ปรากฏอยู่ในแอดเดรสบัส

$\overline{IORQ}$ (input output request) เป็นเอาต์พุตลักษณะไตรสเททให้ลอจิก แอคทีฟที่ "0" เป็นสัญญาณที่บอกให้ทราบว่า CPU ต้องการติดต่อกับอุปกรณ์อินพุต เอาต์พุต โดยแอดเดรสบัส 8 บิตล่างจะให้แสดงค่าเบอร์พอร์ท ส่วนบัสข้อมูลจะ แสดงข้อมูลที่จะมีการส่งถ่ายระหว่าง CPU กับ I/O นอกจากนี้ $\overline{IORQ}$ ถ้าเกิดขึ้น พร้อมกับสัญญาณ M_1 เป็นตัวบอกถึงสถานะที่ CPU กำลังตอบสนองผลการอินเตอร์รัพท์ โดยขณะนั้นส่วนของบัสข้อมูลจะมีการส่งผ่านเข้ามาด้วยค่าของอินเตอร์รัพท์ เวกเตอร์

$\overline{RD}$ (memory read) เป็นเอาต์พุตที่ไตรสเททและแอคทีฟขณะลอจิก "0" $\overline{RD}$ เป็นตัวบอกว่าขณะนี้ CPU ต้องการอ่านข้อมูลจากหน่วยความจำหรืออุปกรณ์ I/O

$\overline{WR}$ (memory write) เป็นเอาต์พุตแบบไตรสเทท และ แอคทีฟขณะลอจิก "0" $\overline{WR}$ เป็นสัญญาณบอกว่า CPU ต้องการเขียนข้อมูลโดยจะเขียนข้อมูลในตำแหน่ง ที่แอดเดรสบัสกำหนดขึ้น อาจเป็นหน่วยความจำหรืออุปกรณ์ I/O ก็ได้

$\overline{RFSH}$ (refresh) เป็นเอาต์พุตแอคทีฟขณะลอจิก "0" $\overline{RFSH}$ เป็นสัญญาณ ที่จะบอกให้ทราบว่าสัญญาณในแอดเดรสบัส ในส่วน $A_0 - A_6$ เป็นแอดเดรสที่จะ ใช้ ในการรีเฟรชหน่วยความจำชนิดไดนามิกส์ส่วนบิต A_7 จะเป็น "0" ส่วนบิต $A_8 - A_{10}$ จะแสดงค่าของรีจิสเตอร์ I

$\overline{HALT}$ (halt state) เป็นเอาต์พุตที่แอคทีฟด้วยลอจิก "0" สัญญาณ $\overline{HALT}$ จะแสดงเมื่อ CPU ได้กระทำคำสั่ง $\overline{HALT}$ และจะหยุดรอจนกว่าจะมีการอินเตอร์รัพท์ หรือรีเซทขณะที่อยู่ในช่วง $\overline{HALT}$ CPU จะเสมือนทำคำสั่ง NOP (no operation) เพื่อให้เกิดไขเคลในการทำงานเพื่อส่งสัญญาณไปกระทำ การรีเฟรชหน่วยความจำ ชนิดไดนามิกส์

$\overline{\text{WAIT}}$ (wait) เป็นขาอินพุตจะแอกทีฟด้วยลอจิก "0" $\overline{\text{WAIT}}$ เป็นตัวกำหนดแสดง เพื่อบอก CPU ให้ CPU หยุดรอ ในกรณีที่อุปกรณ์อินพุต-เอาต์พุต หรือหน่วยความจำไม่สามารถรับหรือส่งข้อมูลได้ทัน $\overline{\text{WAIT}}$ จะเป็นตัวทำให้ CPU ซึ่งได้พอดีกับอุปกรณ์อินพุต-เอาต์พุต ที่ทำงานด้วยความเร็วช้า ๆ

$\overline{\text{INT}}$ (interrupt request) เป็นขาอินพุตจะแอกทีฟด้วยลอจิก "0" $\overline{\text{INT}}$ เป็นสัญญาณที่สร้างขึ้นมาจาก อุปกรณ์อินพุต-เอาต์พุต เพื่อต้องการที่จะอินเทอร์รัพต์ CPU จะทำการตรวจสอบสัญญาณนี้ทุก ๆ ครั้งที่จบการกระทำแต่ละคำสั่ง การตอบสนองของตัวการอินเทอร์รัพต์ สามารถควบคุม ได้ด้วยซอฟต์แวร์ ด้วยการเซตค่าอินเทอร์รัพท์ฟลิปฟล็อป (IFF) การตอบสนอง อินเทอร์รัพต์ จะเกิดได้ยังต้องให้ $\overline{\text{BUSRD}}$ ไม่แอกทีฟเมื่อ CPU ตอบสนองต่อการอินเทอร์รัพต์ CPU จะสร้างสัญญาณตอบด้วยการสร้างสัญญาณ $\overline{\text{IORQ}}$ ระหว่างช่วงเวลา M_1 การตอบสนองต่อการอินเทอร์รัพต์ที่มีแยกแยะได้ 3 แบบ ซึ่งจะอธิบายในรายละเอียดต่อไป

$\overline{\text{NMI}}$ (nonmaskable interrupt) เป็นขาอินพุต ที่จะตรึงนอก CPU ในขณะของบัลซ์ขาลง การอินเทอร์รัพต์ด้วยวิธีนี้ CPU จะให้ความสำคัญสูงกว่า $\overline{\text{INT}}$ กล่าวคือมันจะตอบสนอง และกระทำทันทีด้วยการเริ่มเอ็กซีคิวต์ คำสั่งในตำแหน่ง 0066H โดยอัตโนมัติ การกระโดดไปกระทำในกรณีนี้ CPU จะเก็บค่าโปรแกรมเคาน์เตอร์เดิมไว้ในสแตค เพื่อจะได้กลับไปทำงานเดิมเมื่อเสร็จสิ้นการอินเทอร์รัพต์ แล้ว

$\overline{\text{RESET}}$ (reset) เป็นขาอินพุตที่แอกทีฟด้วยลอจิก "0" การรีเซตในกรณีนี้จะมียุทธวิธีดังนี้

1. ค่าของ PC มีค่าเป็น "0"
2. IFF จะได้รับการ Disable

3. รีจิสเตอร์ I จะมีค่า 00H

4. รีจิสเตอร์ R จะมีค่า 00H

5. จะมีการรีเซตอินเตอร์รัพท์ใหม่ตามอยู่ที่โหมด 0

ระหว่างการรีเซตสายแอดเดรสและบัสข้อมูลได้รับการกระทำ ให้มีค่าอิมพีแดนซ์สูง เพื่อแยกออกจาก CPU ส่วนสายสัญญาณควบคุมการได้รับการทำให้เป็นสัญญาณที่ไม่แอคทีฟ การรีเฟรชจะไม่เกิดขึ้น

$\overline{\text{BUSERQ}}$ (bus request) เป็นขาอินพุตที่แอคทีฟด้วยลอจิก "0" $\overline{\text{BUSERQ}}$ เป็นสัญญาณที่ส่งบอกกับ CPU เพื่อต้องการให้ CPU ควบคุมบัสกล่าวคือต้องการให้ CPU ทำให้บัสแอดเดรสและบัสข้อมูลอยู่ในสถานะอิมพีแดนซ์สูง คือต้องการแยก CPU ออกจากบัสนั่นเอง

$\overline{\text{BUSAK}}$ (bus acknowledge) เป็นขาเอาต์พุต แอคทีฟด้วยลอจิก "0" $\overline{\text{BUSAK}}$ เป็นสัญญาณตอบจาก CPU ว่า CPU ได้แยกตัวเองออกจากแอดเดรสและบัสข้อมูลเรียบร้อยแล้ว (clock) สัญญาณนาฬิกาที่จะป้อนเข้าระบบ

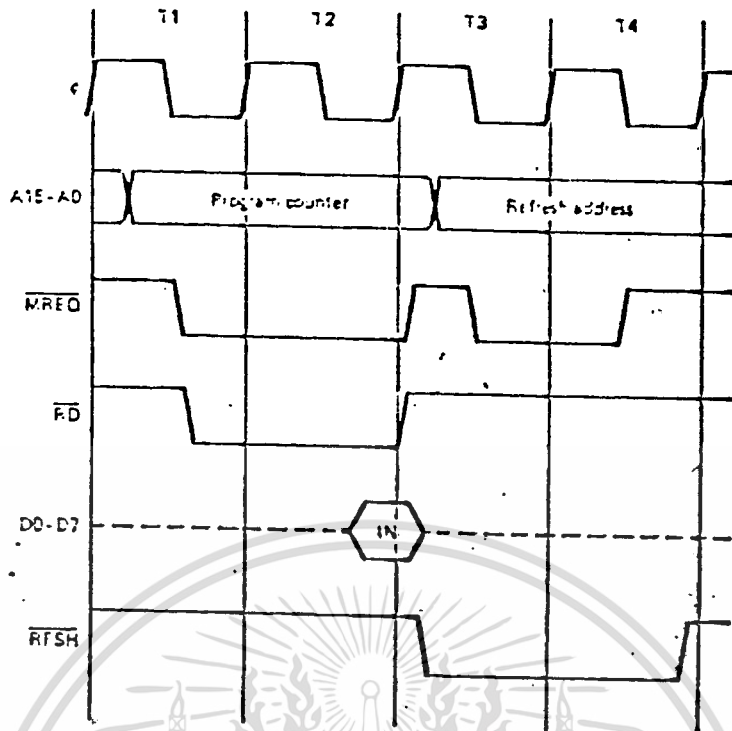
2.6 ไทม์มิ่งการอ่านและเขียนของ Z-80

ไทม์มิ่งของไมโครโปรเซสเซอร์ Z-80 เป็นสิ่งสำคัญต่อหน่วยความจำภายนอก (External memory) และหน่วยอินพุต/เอาต์พุต ซึ่งจะทำให้เข้าใจเกี่ยวกับโครงสร้างการกระทำตามคำสั่งไทม์มิ่งเป็นส่วนสำคัญในการเลือกหน่วยความจำและหน่วยอินพุต/เอาต์พุตที่เป็นส่วนประกอบของ Z-80 ให้สอดคล้องกัน

ไทม์มิ่งการอ่านเบื้องต้น

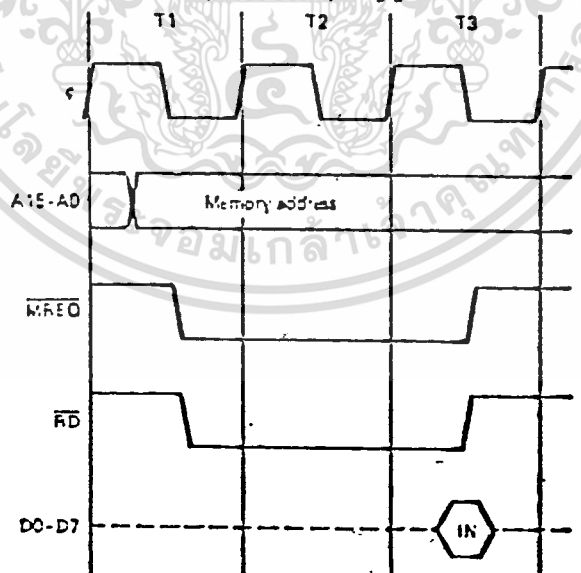
ไมโครโปรเซสเซอร์ Z-80 มีการอ่านหน่วยความจำ 2 ช่วงคือ

1. ออพโค้ดเฟตซ์ รูปที่ 2.6



รูปที่ 2.6 ออฟเค็ตเฟรช (M_1) ไทม์มิ่งของไมโครโปรเซสเซอร์ Z-80

2. การอ่านหน่วยความจำและอินพุท/เอาต์พุต รูปที่ 2.7

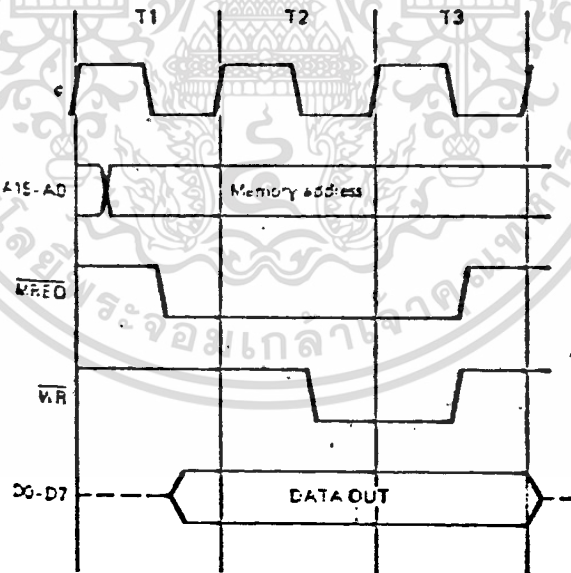


รูปที่ 2.7 ช่วงการอ่านหน่วยความจำของ Z-80

ถ้าไทม์มิงของทั้งสองเปรียบเทียบกันจะสังเกตได้ว่าข้อมูล (DATA) ที่จะอ่านจากหน่วยความจำมีจุดที่แตกต่างกัน ที่ M_1 หรืออินพุตไทม์มิงไคอะแกรมข้อมูล ที่อ่านจะเริ่มที่ขอบขาขึ้นของคล็อก T_1 ในการอ่านหน่วยความจำหรืออินพุต/เอาต์พุต ไทม์มิงไคอะแกรมข้อมูลจะอ่านที่ขอบขาลงของคล็อก ในช่วงกลาง T_2 ดังนั้นหน่วยความจำ จะมีหน้าที่ทั้งในช่วง M_1 และช่วงการหน่วยความจำหรืออินพุต/เอาต์พุต

ไทม์มิงการเขียนเบื้องต้น

สังเกตว่าจะมีลักษณะคล้ายกับไทม์มิงการอ่านในรูป 2.7 ไทม์มิงการเขียนจะต่างจากไทม์มิงการอ่านอยู่ 2 ประการเท่านั้นไมโครโปรเซสเซอร์จะอ้างสัญญาณ WR เป็นสัญญาณ RD และส่งข้อมูลออกจากบัลข้อมูล ขณะที่การอ่านข้อมูลออกจากตัวหน่วยความจำ ข้อมูลที่เขียนลงไปหน่วยความจำ หรืออุปกรณ์อินพุต/เอาต์พุต เมื่อ WR มีสัญญาณกลับเป็นลอจิก "1" ผลที่ได้รับในไทม์มิง นี้จะเกิดหลังถึงกลางของสัญญาณนาฬิกา T_2



รูปที่ 2.8 แสดงการเขียนไทม์มิงของไมโครโปรเซสเซอร์ Z-80

หลักการงานและการออกแบบวงจร

3.1 หลักการงาน

การควบคุมอุณหภูมิในขบวนการนี้ จัดเป็นการควบคุมแบบดิจิตอลโดยตรง ประกอบด้วย
ไปด้วยส่วนสำคัญ 2 ส่วนใหญ่ ๆ คือ

1. ฮาร์ดแวร์ (Hardware)

ประกอบด้วยส่วนของการวัดค่าอุณหภูมิจากขบวนการ และแปลงเป็นค่าแรงดันไฟฟ้า
และส่วนแปลงค่าแรงดันไฟฟ้าเป็นความถี่ และความถี่เป็นแรงดันไฟฟ้า เพื่อผลของ
การลดสัญญาณรบกวน (Noise) ในระบบส่วนของตัวควบคุมและมีการแปลงสัญญาณ
อนาลอกเป็นดิจิตอล

2. ซอฟต์แวร์ (Software)

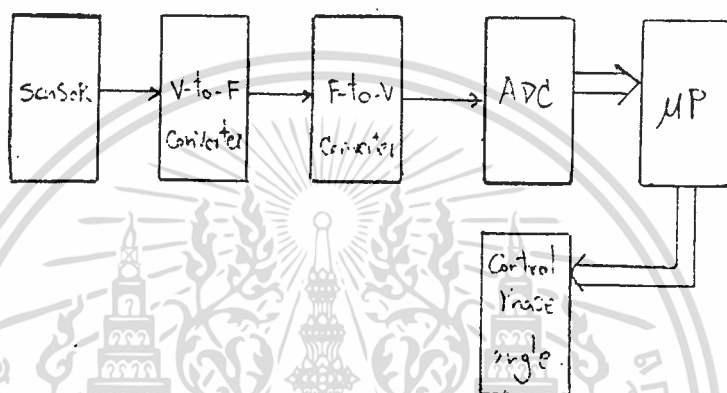
ประกอบด้วย โปรแกรมการทำงานของตัวควบคุม ซึ่งจะกำหนดการทำงานของ
การควบคุม

ฮาร์ดแวร์ (Hardware)

ฮาร์ดแวร์มีโครงสร้างส่วนประกอบต่าง ๆ ดังรูปที่ 3.1

จากขบวนการค่าอุณหภูมิจะถูกแปลงเป็นระดับแรงดันไฟฟ้า โดยตัวตรวจจับอุณหภูมิ
ซึ่งในโครงการนี้ จะใช้ตัวเทอร์โมคัปเปิลแบบ K แรงดันไฟฟ้าจะผ่านการขยายให้
มีอัตราค่าการเปลี่ยนของแรงดันไฟฟ้าต่อการเปลี่ยนแปลงอุณหภูมิอยู่ในระดับ 10 mV/°C
จากนั้น ค่าแรงดันไฟฟ้าจะถูกเปลี่ยนเป็นความถี่ โดยมีอัตราค่าการเปลี่ยนแปลง
ของความถี่ ต่อการเปลี่ยนแปลงของระดับแรงดันไฟฟ้าเป็น 1 kHz/V จากนั้นก็จะ

ถูกเปลี่ยนกลับเป็นค่าแรงดันไฟฟ้าอีกครั้ง การที่ทำเช่นนี้ เพื่อกำจัดผลของสัญญาณรบกวน เมื่อได้แรงดันไฟฟ้าจากส่วนเปลี่ยนความถี่เป็นแรงดันไฟฟ้า จะถูกทำการแปลงให้เป็นข้อมูลทางดิจิทัล เพื่อป้อนข้อมูลทางดิจิทัลให้ไมโครโปรเซสเซอร์ทำการควบคุมต่อไป



รูป 3.1 แสดงส่วนประกอบต่างของโครงงานนี้

Software

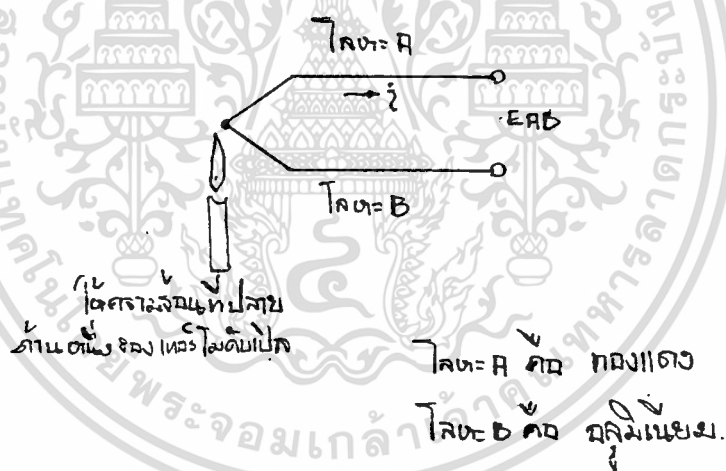
ส่วนนี้เปรียบเสมือนหัวใจในการทำงานของตัวควบคุมการเปรียบเทียบค่า และการประมวลผลในตัวควบคุม ก็จะถูกกำหนดในส่วนของซอฟต์แวร์นี้ ส่วนของซอฟต์แวร์นี้จะประกอบไปด้วย โปรแกรมการทำงานที่ควบคุม การทำงานของตัวควบคุมเอง คือ รับค่าอุณหภูมิจากผู้ใช้ การแสดงผลการควบคุม และการโปรแกรมการทำงานในการควบคุมอุณหภูมิในขบวนการ

3.2 การออกแบบวงจร

ส่วนวัดอุณหภูมิ และแปลงสัญญาณไฟฟ้า ประกอบไปด้วย ตัววัดอุณหภูมิ, วงจรขยาย
วงจรชดเชยอุณหภูมิ

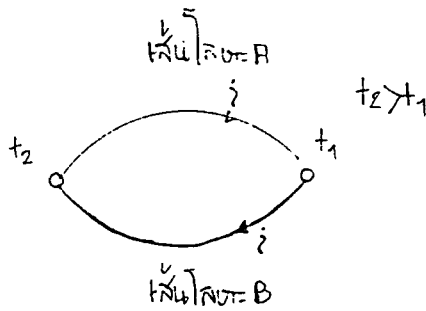
1. ตัววัดอุณหภูมิ (Temperature Transducer)

ใช้เทอร์โมคัปเปิลเป็นตัวตรวจวัดอุณหภูมิของกระบวนการ และแปลงเป็น
แรงดันไฟฟ้า หลักการทำงานของเทอร์โมคัปเปิลมีอยู่ว่า เมื่อนำเส้นลวดโลหะต่างชนิดกัน
2 ชนิด มาเชื่อมเข้าด้วยกันที่ปลายทั้งสองข้าง และให้ความร้อนที่ปลายด้านหนึ่ง
จะเกิดกระแสไหลในเส้นลวดทั้งสองเส้นนั้น ดังรูปที่ 3.2



รูปที่ 3.2 แสดงหลักการทำงานของเทอร์โมคัปเปิล

และจากรูป 3.2 ถ้าเราเปิดวงจรที่จุดต่อ 1 ก็จะทำให้เกิดค่าความต่างศักย์ EAB ขึ้น
เรียกว่าความต่างศักย์ซีเบค (Seebeck Voltage) ดังรูปที่ 3.4



$$\varepsilon_{AB} = \alpha (T_2 - T_1)$$

รูปที่ 3.3 แสดงการเกิดความต่างศักย์ซีเบค

ค่าความต่างศักย์ซีเบคนี้ จะมีค่าขึ้นอยู่กับอุณหภูมิของจุดต่อและชนิดของเส้นลวดโลหะที่ใช้ทำเทอร์โมคัปเปิล สำหรับโครงงานนี้ เราใช้เทอร์โมคัปเปิลแบบ K ซึ่งมีค่าสัมประสิทธิ์ ซีเบคเท่ากับ $40.7 \mu\text{V}/^\circ\text{C}$ โดยค่า ε_{AB} ที่วัดได้เป็นไปตามสมการ

$$\varepsilon_{AB} = \alpha (T_2 - T_1)$$

เมื่อ

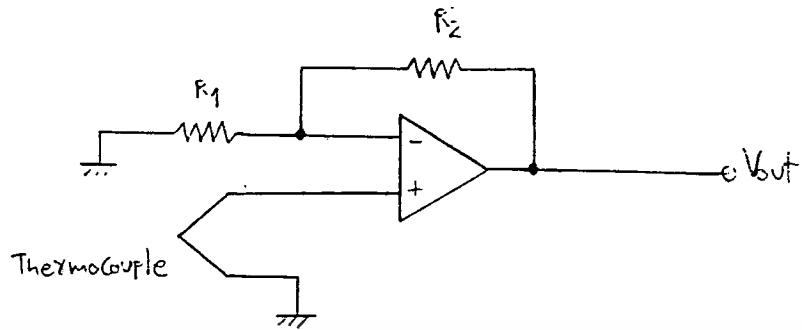
$$\alpha = \text{สัมประสิทธิ์ซีเบค}$$

$$T_1 = \text{อุณหภูมิที่ปลายข้างหนึ่ง}$$

$$T_2 = \text{อุณหภูมิที่ปลายอีกข้างหนึ่ง}$$

2. วงจรขยายสัญญาณ

การเปลี่ยนแปลงของระดับแรงดันจากเทอร์โมคัปเปิลมีค่าน้อยมาก ประมาณ $40 \mu\text{V}/^\circ\text{C}$ จึงต้องทำการขยายสัญญาณเป็น $10 \text{ mV}/^\circ\text{C}$ วงจรขยายสัญญาณที่ใช้เป็นแบบ นอนอินเวอร์ตติ้งแอมป์ (Non-Inverting Amplifier) ดังรูปที่ 3.4



รูปที่ 3.4 แสดงวงจรขยายแบบอินเวอร์ตติ้งแอมป์

การออกแบบ

$$\text{Gain} = \frac{V_{out}}{V_{in}} = \frac{R_2 + 1}{R_1}$$

แรงดันที่เอาต์พุตของวงจรขยายเท่ากับ 10 mV/ °C

แรงดันอินพุตจากเทอร์โมคัปเปิลเท่ากับ 40.7 μV/ °C

$$\text{Gain} = \frac{10 \text{ mV}}{40.7 \text{ μV}} = 245.7$$

47 μV

สำหรับความต้านทานของเทอร์โมคัปเปิลแบบ K มีค่าประมาณ 422 โอห์ม ดังนั้น เราจึงให้ค่า $R_1 = 422$ ดังนั้น $R_2 = 103.685$ กิโลโอห์ม

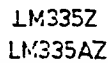
3. วงจรชดเชยอุณหภูมิ

ค่าความต่างศักย์ซีเบคจากเทอร์โมคัปเบิลนี้เป็นค่าที่เกิดจากความแตกต่างระหว่างอุณหภูมิ T2 และ T1 ตามปกติในการใช้เทอร์โมคัปเบิลจึงต้องมีการรักษาอุณหภูมิ T1 ไว้ที่ศูนย์องศาเซลเซียส เพื่อให้ค่าความต่างศักย์ที่วัดได้เป็นค่าที่อ้างอิงกับอุณหภูมิศูนย์องศาเซลเซียส แต่การรักษาอุณหภูมิศูนย์องศานั้นทำได้ยากมาก ดังนั้นจึงได้นำเทคนิคทางไฟฟ้ามาใช้ในการชดเชยด้วยระบบแรงดันไฟฟ้า การชดเชยระดับแรงดันไฟฟ้านี้ จะทำการบวกค่าแรงดันไฟฟ้าเข้ากับค่าแรงดันที่วัดได้จากเทอร์โมคัปเบิล ซึ่งเกิดจากค่าผลต่างของอุณหภูมิของขบวนการและอุณหภูมิห้อง โดย แรงดันที่จะบวกเพิ่มเข้าไปนี้มีค่าเท่ากับแรงดันที่เกิดจากเทอร์โมคัปเบิลที่วัดอุณหภูมิห้องเทียบกับศูนย์องศาเซลเซียส

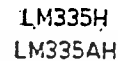
วงจรที่ใช้ชดเชยแรงดันไฟฟ้านี้ จะต้องให้ค่าแรงดันเอาต์พุตของมันแปรไปตามอุณหภูมิ ห้องที่เปลี่ยนไปด้วย ดังนั้นวงจรนี้จึงใช้ไอซีเบอร์ LM 335 ซึ่งเป็นไอซีชนิด Precision Temperature Sensor ซึ่งมีคุณสมบัติที่สำคัญดังนี้

LM 335 Precision Temperature

Output	+10 mV/ °C from absolute
Output Voltage	2.92 - 3.04 Volt (at 1 mA)
Operating Current	400 mA - 5 mA
Dynamic Impedance	< 1
Temperature Range	0 - 100 °C
Temperature error with 25 °C Cal	1-2 °C
Break down	> 0.5 mA
Non-Linear	0.3 - 1.5 °C

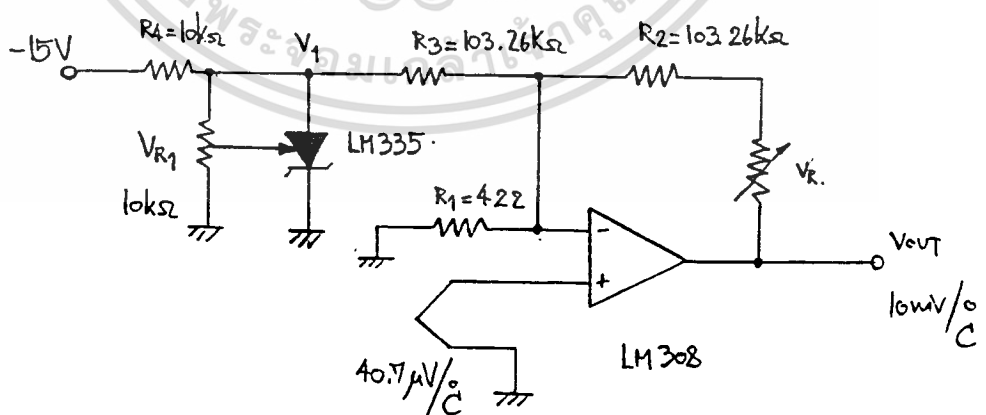


(က)



(v)

จากคุณสมบัติของ LM 335 มีข้อมูลที่น่าสนใจอยู่ 2 อย่างคือเอาต์พุต (Output) เปลี่ยนแปลงตามอุณหภูมิด้วยค่า $10 \text{ mV}/^{\circ}\text{C}$ จากค่าสมบูรณ์ (Absolute) และ มีการใช้งานตั้งแต่ $0-100^{\circ}\text{C}$ ซึ่งครอบคลุมอุณหภูมิห้องได้อย่างเพียงพอเราจึงใช้ LM 335 นี้สร้างแรงดันไฟฟ้า V_1 ขึ้นมาเพื่อชดเชยอุณหภูมิห้อง และสามารถเปลี่ยนค่าไปตาม การเปลี่ยนแปลงของอุณหภูมิห้องด้วยค่า $+10 \text{ mV}/^{\circ}\text{C}$ จากค่าสมบูรณ์ (Absolute)



รูปที่ 3.6 แสดงวงจรชุดขยายเพนมิ

ในรูปที่ 3.6 แรงเคลื่อน V_1 เอาไปรวมกับเอาต์พุตจากรอยต่อเย็น (Cold junction) ของเทอร์โมคัปเปิล และแรงเคลื่อน V_1 ป้อนเข้าสู่วงจรขยายด้วยลักษณะวงจรแบบอินเวอร์ตติ้งแอมป์ (Inverting Amplifier) ดังนั้นจึงได้เกน (Gain) ที่จะขยายแรงเคลื่อน V_1 เป็น $-R_2/R_3$ และต้องมีค่าเท่ากับ -1 ดังนั้น

$$R_3 = R_2 = 103.2 \text{ กิโลโอห์ม}$$

วงจรที่มีการชดเชยดังรูปที่ 3.6 ที่อุณหภูมิห้องประมาณ 25°C จะได้แรงเคลื่อน V_1 มีค่าประมาณ

$$\begin{aligned} V_1 &= 10 \text{ mV}/^\circ\text{C} \times (25 - (-273)) \\ &= (25 + 273) \times 10 \text{ mV} \\ &= 2980 \text{ mV} \\ &= 2.98 \text{ V} \end{aligned}$$

จากรูปที่ 3.6 เมื่อต้องการให้กระแสไหลผ่าน 1.2 mA ดังนั้น R_4 จึงมีค่าเท่ากับ

$$R_4 = (15 - 3) / 1.2 \text{ Ohm}$$

$$= 10 \text{ Kohm}$$

The diagram shows a precision rectifier circuit. The input is connected to a -15V supply through a resistor $R_4 = 10k$. The input signal V_{K1} is applied to the non-inverting input of the op-amp through a resistor $R_1 = 422$. The op-amp's output is connected to the anode of diode $LM1329Z$ through a resistor $R_7 = 4.7k$. The cathode of $LM1329Z$ is connected to ground. The output of the op-amp is also connected to the anode of diode $LM1335$ through a resistor $R_3 = 103.26k$. The cathode of $LM1335$ is connected to ground. The output of the circuit is aV_{out} with a sensitivity of $10mV/C$. A note indicates the op-amp is a TYPE K with a sensitivity of $40.7 \mu V/C$.

4. ส่วนแปลงแรงเคลื่อนไฟฟ้าเป็นความถี่ (Voltage To Frequency)

$$f_0 = 1 \frac{\text{kHz}}{V_{in}} \times V_{in}$$

เมื่อแรงเคลื่อนที่อินพุตเต็มสเกล(Full Scale)เท่ากับ 10 โวลต์ ค่าความถี่ที่เอาต์พุตของ LM 331 จะคำนวณจากสมการ

$$f_o = \frac{V_{in} \times R_5}{2V \times R_L \times R_t C_t}$$

ตัวต้านทาน(R_t)และตัวเก็บประจุ(C_t) จะใช้ในการกำหนดช่วงเวลาที่เอาต์พุตเป็นลอจิกศูนย์โดยจะคำนวณจากสมการ

$$t_{low} = 1.1 R_t C_t$$

โดยปกติเรามักจะกำหนดให้คาบเวลาน้อยกว่าคาบเวลาของความถี่สูงสุดที่เอาต์พุตของ Lm 331 โดยกำหนดให้น้อยกว่า 80% ของคาบเวลาน้อยที่สุดจากสมการ

$$T_{min} = \frac{1}{f_{max}}$$

กำหนดให้ความถี่สูงสุดเท่ากับ 10 khz

$$T_{min} = \frac{1}{10 \text{ khz}}$$

$$= 100 \mu S$$

$$= 0.1 \text{ ms}$$

ปกติมักให้ความกว้างของพัลส์ประมาณ 80 % ของคาบเวลาต่ำสุด

$$t_{low} = 0.8 \times 100 \mu s$$

$$= 80 \mu s$$

กำหนดให้ค่าตัวเก็บประจุ (C_t) เท่ากับ $0.01 \mu F$ เพราะฉะนั้นตัวต้านทาน (R_t) จะหาได้จาก

$$R_t = \frac{80 \mu s}{1.1 \times 0.01 \mu F}$$

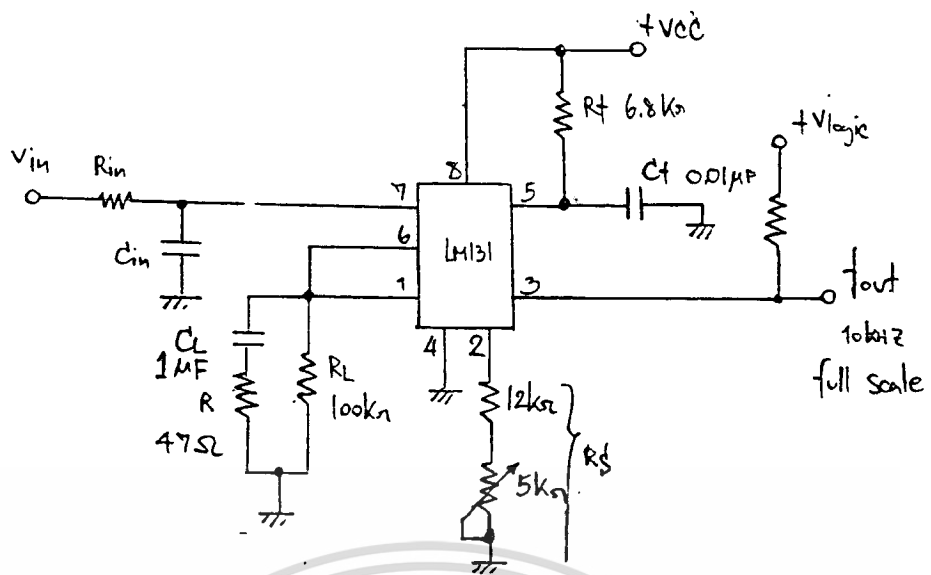
เลือกใช้ตัวต้านทาน (R_t) เท่ากับ $6.8 K\Omega$

เพราะฉะนั้นค่าความต้านทาน (R_s) จะมีค่าเท่ากับ

$$R_s = \frac{(2V)(10KHz)(10K\Omega)(6.8K\Omega)(0.01\mu F)}{5V}$$

$$= 27.2 K\Omega$$

เพื่อให้สามารถทำการปรับค่าต่าง ๆ ได้ เราจึงใช้ตัวต้านทานแบบปรับค่าได้เท่ากับ $10 K\Omega$ ต่ออนุกรมกับ ตัวต้านทาน $18 K\Omega$



รูปที่ 3.8 แสดงวงจรเปลี่ยนแรงเคลื่อนไฟฟ้าเป็นความถี่

5. ส่วนแปลงความถี่เป็นแรงเคลื่อน (Frequency To Voltage Conversion)

การออกแบบจะเริ่มที่เราต้องการกระแสที่ไหลไปที่เอาต์พุตไปยังโหลดด้วยช่วงเวลาเท่าใดสามารถกำหนดได้ด้วยสมการ

$$t = 1.1 R_t C_t$$

ในโครงการนี้เรากำหนดไว้เท่ากับ $80 \mu s$ และเลือกค่าตัวเก็บประจุ (C_t) เท่ากับ $0.01 \mu F$ ดังนั้น

$$R_t = 7.2 K\Omega$$

แต่เราเลือกใช้ตัวต้านทาน (R_t) เท่ากับ $6.8 K\Omega$

สำหรับกระแสจะกำหนดโดยสมการ

$$i = \frac{2v}{R_s} < 200 \mu A$$

ดังนั้นจะได้ตัวต้านทาน (R_s) เท่ากับ 10 Kohm และ กระแสเฉลี่ยที่เอาต์พุตจะ
คำนวณจากสมการ

แทนค่า

เมื่อ

$$i_{ave} = \frac{i_t}{T}$$

$$i_{ave} = \frac{2v \times 1.1R_t C_t}{R_s T}$$

$$T_{min} = \frac{1}{f_{max}}$$

$$= \frac{1}{10Khz}$$

$$= 100 \mu s$$

$$= 0.1 ms$$

จะได้

$$I_{ave} = \frac{(2v)(1.1)(6.8Kohm)(0.01\mu F)}{(10Kohm)(100\mu S)}$$

$$= 149.6 \mu A$$

เพราะฉะนั้น

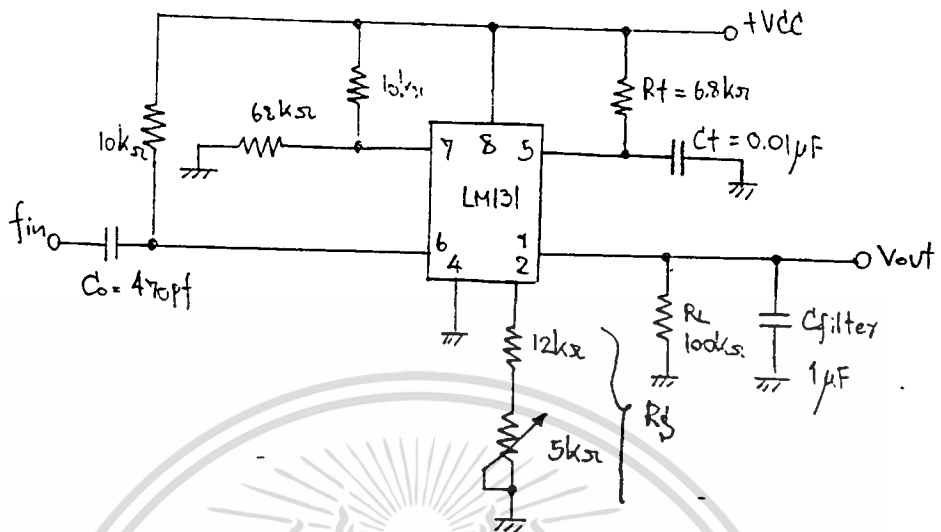
$$V_{ave} = I_{ave} R_L$$

เมื่อต้องการให้แรงเคลื่อนนี้เท่ากับแรงเคลื่อนของแหล่งจ่าย (VCC) เมื่อ VCC=15v

จะได้

$$R_L = \frac{15V}{149.6\mu A} = 100 Kohm$$

แต่เอาท์พุทของวงจรนี้ยังคงมีริปเปิล (ripple) อยู่ซึ่งกำจัดโดยการให้ตัวเก็บประจุ ต่อพร้อมกับตัวต้านทาน (RL) ที่ขา 1 ของไอซี LM 331



รูปที่ 3.9 แสดงวงจรเปลี่ยนความถี่เป็นแรงเคลื่อน

5. วงจรเปลี่ยนสัญญาณแอนะล็อกเป็นดิจิทัล (A/D conversion)

การออกแบบจะเริ่มคำนวณจากสมการ

$$\frac{V_{in}}{(V_{FS} - V_Z)} = \frac{D_x}{(D_{max} - D_{min})}$$

เมื่อ

V_{in} = แรงเคลื่อนอินพุต

V_{FS} = แรงเคลื่อนเต็มสเกลของไอซี ADC 0809

V_Z = แรงเคลื่อนซีโรวของไอซี ADC 0809

D_{max} = ค่าข้อมูลดิจิทัลสูงสุด

D_{min} = ค่าข้อมูลดิจิทัลต่ำสุด

D_x = ค่าข้อมูลที่แปลงได้

ในไอซี ADC 0809 นั้นเป็นไอซี A/D ชนิด 8 บิต ดังนั้นสามารถแบ่งได้เป็น 256 ระดับนั้นคือเริ่มจาก 0-255 เพราะฉะนั้น $D_{max} = 255$, $D_{min} = 0$ สำหรับแรงเคลื่อนเต็มสเกลกำหนดไว้เท่ากับ 5 โวลต์ ดังนั้นเมื่อแรงดันอินพุตเท่ากับ 5 V ค่าข้อมูลที่แปลงได้จะเท่ากับ

$$\frac{5}{(5-0)} = \frac{D_x}{(D_{max}-D_{min})}$$

ดังนั้นค่าข้อมูลดิจิตอลที่แปลงได้ (D_x) เท่ากับ 255

เพราะฉะนั้นแรงเคลื่อนที่อินพุตที่สูงที่สุดที่สามารถป้อนได้จะเท่ากับ 5 โวลต์

6. การออกแบบวงจรมิกด้วยดิจิตอล (Digital control phase)

จากวงจรเพื่อให้วงจรทำงานได้พร้อมกับสัญญาณไฟสลับ 50Hz โดยให้ทำงานครึ่งไซเคิลของไฟสลับทำงานได้ 256 ระดับดังนั้นความถี่ที่วงจรอะอสเตเบิลต้องผลิตจะเท่ากับ $100 \times 256 = 25.6 \text{ KHz}$

จากวงจรอะอสเตเบิลความถี่ของวงจรจะคำนวณได้จาก

$$f_{out} = \frac{1.44}{(R_1 + 2R_2)C_1}$$

กำหนดให้ตัวเก็บประจุ (C_1) เท่ากับ 0.01 μF และตัวต้านทาน R_2 เท่ากับ 1 K Ω m ดังนั้นตัวต้านทาน R_1 จะเท่ากับ

$$R1 = \frac{1.44}{-2R2}$$

foutxc1

$$= \frac{1.44}{-2 \text{ Kohm}}$$

$$25.6 \text{ Khz} \times 0.01 \mu\text{F}$$

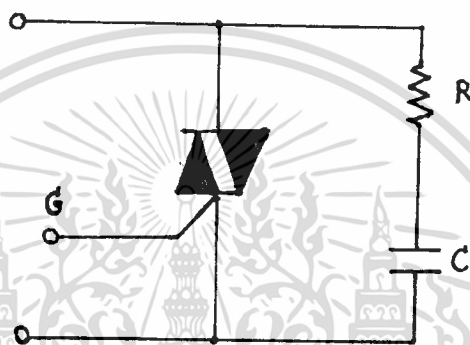
$$= 3625 \text{ Ohm}$$

$$= 3.625 \text{ Kohm}$$

เลือกใช้ตัวต้านทานแบบปรับค่าได้ ๖R1 เท่ากับ 5 กิโลโอห์ม

7. การออกแบบวงจรสับเบิล

วงจรสับเบิลเป็นวงจรที่ใช้ในการป้องกันแรงดันกระชอกหรือแรงดันเปลี่ยนแปลงชั่วครู่ ซึ่งเป็นแรงดันกระชอกที่เป็นแรงดันชั่วครู่ที่ทำให้ไทรสเตอร์นำกระแสได้ด้วยวิธี dv/dt อันเป็นวิธีที่เราไม่ต้องการให้เกิดขึ้นดังนั้นเราจึง ต่อวงจรอาร์ซี (Rc circuit) คร่อมไทรสเตอร์ ดังรูปที่ 3.11



รูปที่ 3.11 แสดงการต่อวงจรสับเบิล

การออกแบบ

$$\text{เวลาคงที่} = \frac{0.632 \times \text{อัตราแรงดันปกติของไทรสเตอร์}}{dv/dt}$$

เมื่อแรงดันปกติของไทรสเตอร์เท่ากับ 400 โวลต์และ dv/dt จากผู้ผลิต $= 10 \text{ v}/\mu\text{s}$

$$\text{เวลาคงที่} = \frac{0.632 \times 400 \text{ V}}{10 \text{ v}/\mu\text{s}}$$

สำหรับการคำนวณหาค่าตัวต้านทาน (R) จะหาได้จาก

$$R = \frac{10 \text{ Vline(peak)}}{I_{\text{tcm}} - I_{\text{load}}}$$

เมื่อแรงดันของสายเท่ากับ 310 โวลต์และกระแส $I_{\text{sc}} = 100 \text{ Amp}$ และกระแส
ที่โหลด (I_{load}) เท่ากับ 10 Amp ดังนั้น

$$R = \frac{(10)(310)}{(100-10)}$$

$$= 34.4 \text{ Ohm}$$

ดังนั้นเลือกใช้ตัวต้านทาน (R) เท่ากับ 39 โอห์ม
สำหรับตัวเก็บประจุ (C) หาได้จากเวลาคงที่ $\leq R \times C$

$$C = \frac{25.2 \text{ } \mu\text{s}}{39 \text{ Ohm}}$$

$$= 0.6 \text{ } \mu\text{F}$$

เราเลือกใช้ตัวเก็บประจุ (C) เท่ากับ 0.01 μF

การทดลองและผลการทดลอง

ในบทนี้ ประกอบด้วย การทดสอบผลการทำงานของโครงงานนี้ 2 ส่วนใหญ่ๆ คือ ส่วนการวัดอุณหภูมิ และแปลงเป็นสัญญาณดิจิตอล กับส่วนของตัวควบคุม

4.1 ส่วนการวัดอุณหภูมิและแปลงเป็นสัญญาณดิจิตอล

แบ่งการทดลองออกเป็นขั้นตอนต่าง ๆ ดังนี้

ขั้นที่ 1 ทดสอบและปรับค่าวงจรแปลงระดับแรงดันเป็นความถี่ให้ได้อัตราการแปลงความถี่ต่อการเปลี่ยนแปลงของระดับแรงดัน เท่ากับ 1 กิโลเฮิรตซ์ ต่อโวลต์ จากรูปที่ 3.8 ความสัมพันธ์ระหว่างความถี่เอาต์พุต กับระดับแรงดันอินพุตของวงจรเป็นไปตามสมการ

$$F_{out} = \frac{-V_{in} \times R_0 \times 1}{2.09 \times R_{in} \times R_t \times C_t}$$

เราใช้ค่า

$$R_t = 6.8 \text{ K} \quad C_t = 0.01 \text{ uF}$$

$$R_0 = 12.1 \text{ K (คงที่)} + 5 \text{ K (ปรับค่าได้)}$$

$$R_{in} = 100 \text{ K}$$

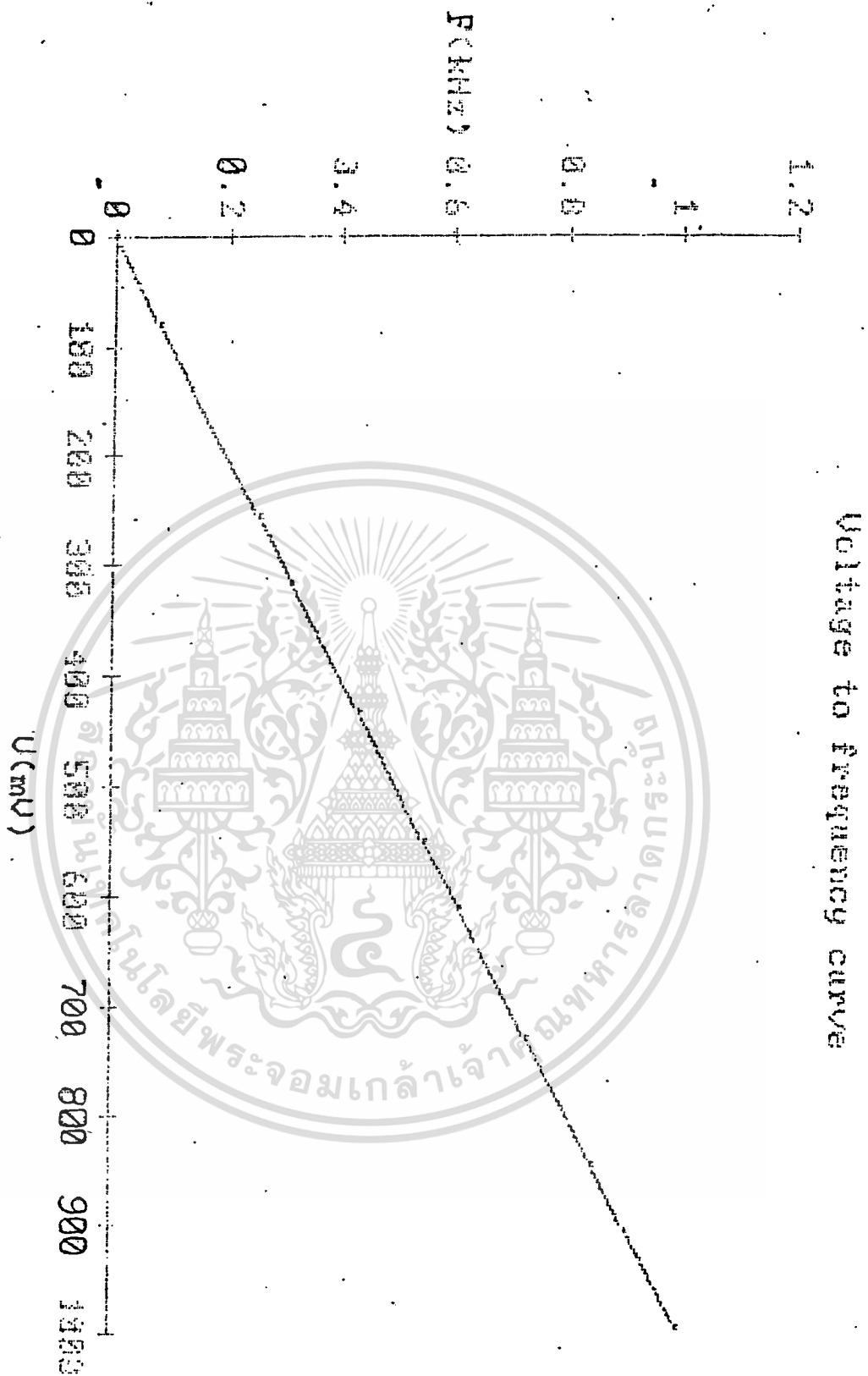
จากนั้น ทำการปรับค่าความต้านทาน R_0 และปรับค่าออฟเซตในวงจรอินทิเกรเตอร์ให้ได้ อัตราการเปลี่ยนแปลงความถี่ต่อการเปลี่ยนแปลงของ ระดับแรงดัน เท่ากับ 1 กิโลเฮิรตซ์ ต่อ โวลต์ ซึ่งได้ผลการทดลองดังตารางที่ 4.1 และ นำตารางมาเขียนกราฟได้ดังรูปที่ 4.2

ระดับแรงดัน (mV)	ความถี่ (KHz)	ความผิดพลาด (%)
0	0	0
80	.078	-2.5
138	.138	0
256	.258	0.78
316	.318	0.63
433	.438	1.15
552	.556	0.72
612	.619	1.14
732	.741	1.23
849	.861	1.41
909	.921	1.32
999	1.011	1.20

รูปที่ 4.1

ตารางผลการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อ 53 ศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.2

ขั้นที่ 2 ในขั้นนี้จะทำการปรับค่าวงจรชดเชยอุณหภูมิ ให้ได้อัตราการเปลี่ยนแปลงของระดับแรงดันต่ออุณหภูมิเป็น $10 \text{ mV/}^{\circ}\text{C}$

การทดลองทำโดย

ก) นำเทอร์โมคัปเบิลออก แล้วต่อกราวด์เป็นดินพุทของวงจรขยาย เพื่อให้แรงดันของวงจรส่วนขยายของเทอร์โมคัปเบิลมีค่าเป็นศูนย์ (อาจมีค่าไม่เป็นศูนย์ แต่ใกล้ ๆ ศูนย์ เนื่องจากออฟเซ็ทของออปแอมป์ ค่าออฟเซ็ทนี้ สามารถชดเชยได้จากการปรับค่าศูนย์)

ข) ใช้วงจรชดเชยอุณหภูมิ ทำการวัดค่าอุณหภูมิ จากการต้มน้ำ แล้ววัดค่าความถี่เอาท์พุทของวงจรแปลงแรงดันเป็นความถี่ (ความถี่ 0.01 กิโลเฮิรตซ์ แทนอุณหภูมิ 1°C) เทียบกับอุณหภูมิที่วัดได้จากเทอร์โมมิเตอร์

ค) ทำการปรับแต่ง (calibration) ค่า VR1 และ VR2 เพื่อให้อุณหภูมิที่วัดได้จากวงจรชดเชยอุณหภูมินี้ มีค่าใกล้เคียงกับเทอร์โมมิเตอร์มากที่สุด การปรับ VR1 นี้ คือการปรับสเกล (span) และการปรับ VR2 คือการปรับค่าศูนย์ (zero) ของวงจร

จากการปรับแต่งวงจรชดเชยอุณหภูมิ ได้ผลการทดลองออกมา ดังตารางรูปที่ 4.8

อุณหภูมิ (C)	ความถี่ (KHz)	ความผิดพลาด (%)
70	.70	0.0
69	.69	0.0
68	.68	0.0
67	.67	0.0
66	.65	-1.5
65	.64	-1.5
64	.63	-1.5
63	.62	-1.6
62	.62	0.0
61	.61	0.0
60	.60	0.0
59	.59	0.0
58	.58	0.0
57	.57	0.0
56	.55	-1.8
55	.54	-1.8
54	.52	-3.7

รูปที่ 4.3

ตารางผลการทดลองของวงจรต.เซย์อุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการ56เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นที่ 3 หลังจากผ่าน 2 ขั้นตอนแรกมาแล้ว ในขั้นตอนนี้ จะนำเอาเทอร์โมคัปเบิลต่อเข้าไปเป็นอินพุทของวงจรขยายสัญญาณ แล้วนำเทอร์โมคัปเบิลวัดอุณหภูมิจากการต้มน้ำ ส่วนวงจรชดเชยอุณหภูมิที่ปรับแต่งแล้วจะวัดอุณหภูมิห้อง จากนั้นทำการปรับแต่งค่าการขยายของวงจรขยาย และ ค่า VR2 แล้วเปรียบเทียบความถี่จากวงจรแปลงแรงดันเป็นความถี่ กับ อุณหภูมิที่วัดจากเทอร์โมมิเตอร์ ปรับให้ได้ค่าใกล้เคียงกับเทอร์โมมิเตอร์มากที่สุด

ผลการวัดอุณหภูมิจากเทอร์โมคัปเบิล ซึ่งมีการชดเชยอุณหภูมิแล้ว แสดงดังตารางรูปที่ 4.4

อุณหภูมิ (C)	ความถี่ (KHz)	ความผิดพลาด (%)
70	.70	0.0
69	.70	-1.4
68	.69	-1.5
67	.67	0.0
66	.66	0.0
65	.65	0.0
64	.65	-1.6
63	.63	0.0
62	.61	-1.6
61	.60	-1.6
60	.59	-1.7

รูปที่ 4.4

ตารางแสดงการวัดอุณหภูมิของทั้งหมด

4.2 ส่วนของตัวควบคุมแบบดิจิทัล

ในส่วนนี้นับว่าเป็นส่วนที่ทำการทดลองยากที่สุด เนื่องจากเราต้องทำเขียนโปรแกรมโดยใช้ภาษาแอสเซมบลี (Assembly) ซึ่งเราทราบถึงข้อดีของภาษานี้คือ ทำงานได้รวดเร็ว แต่มีข้อเสีย คือ เมื่อเกิดการผิดพลาดจะตรวจสอบได้ยากและเสียเวลา เพราะ เราต้องทำการป้อนลงในแรมแพค (Ram pack) แล้วจึงนำข้อมูลในแรมแพคใส่ลงในบอร์ด Z80 เพื่อทำการเอ็กคิวต์ (Execute) โปรแกรม ถ้าการเอ็กคิวต์โปรแกรมแล้วได้ผลไม่เป็นไปตามที่ต้องการ เราก็ต้องทำการแก้ไขใหม่แล้ว นำใส่ในแรมแพคใหม่อีกครั้ง ทำเช่นนั้นจนกระทั่งได้ผลตามต้องการ สำหรับการเอ็กคิวต์โปรแกรมในโครงงานนี้ ผลปรากฏว่า ส่วนรับข้อมูลจากคีย์บอร์ดและส่วนแสดงผลทำงานได้ตามต้องการ แต่สำหรับส่วนควบคุมนั้นยังทำงานไม่สมบูรณ์นัก ยังมีข้อผิดพลาดอยู่

บทที่ 5

บทวิจารณ์ และ บทสรุป

บทวิจารณ์

ปัญหาต่าง ๆ ที่พบในการทำโครงงานนี้ ได้แก่ การหาอุปกรณ์ต่างๆ ในท้องตลาด เช่น ตัวต้านทาน , ตัวเก็บประจุ ซึ่งต้องใช้ค่าใกล้เคียงกับที่หาได้จากคำนวณ และการหาอุปกรณ์ที่ใช้ชดเชยเกี่ยวกับอุณหภูมิ คือ ไอซีเบอร์ LM 329 ซึ่งหาได้ยาก ดังนั้นเราจึงใช้อุปกรณ์ที่มี สัมประสิทธิ์การเปลี่ยนตามอุณหภูมิ คือ ซีเนอร์ไดโอด แทน ทำให้การชดเชยอุณหภูมิทำได้ไม่สมบูรณ์

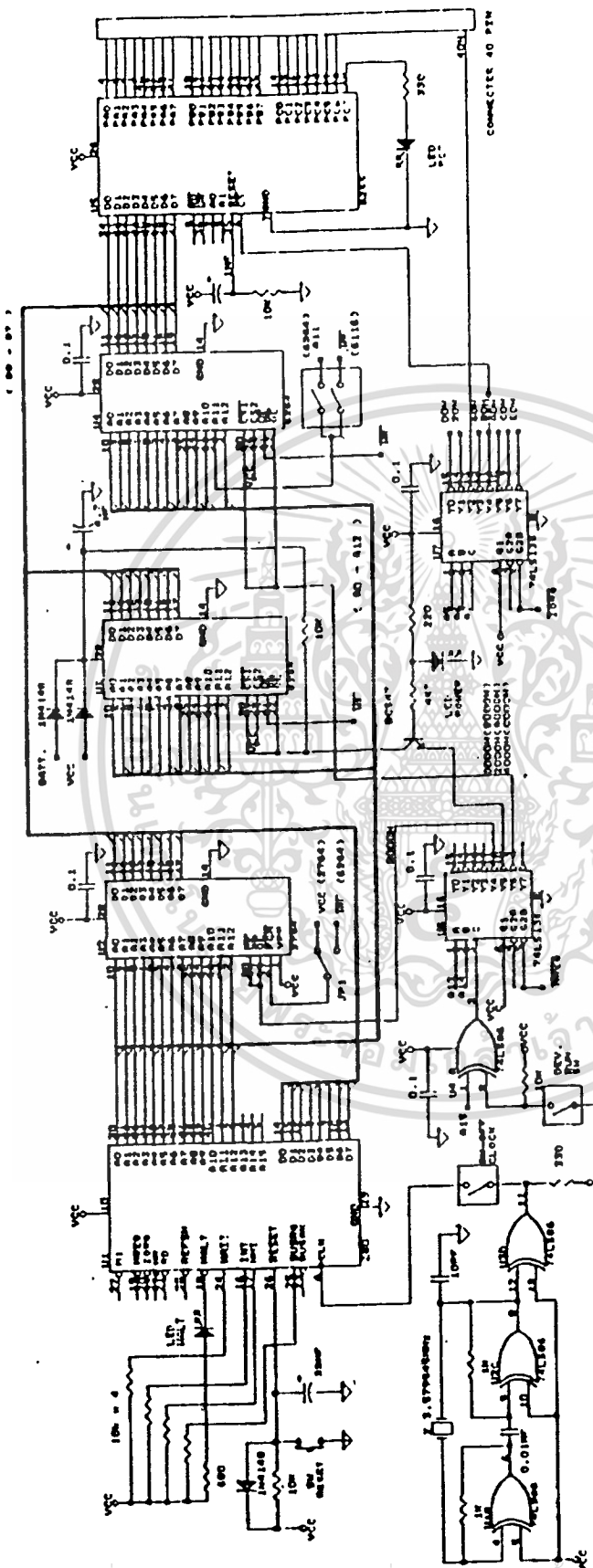
สำหรับปัญหาต่อมา คือ สัญญาณรบกวน (Noise) จากภายนอกซึ่งก็แก้ไขได้โดยใช้วงจรโลว์พาสฟิลเตอร์ (Low pass filter) แต่ ก็ไม่สามารถกรองสัญญาณรบกวนได้หมด และ ปัญหาเกี่ยวกับการชดเชยเทอร์โมคัปเปิล ณ อุณหภูมิห้อง ซึ่งในการออกแบบเราออกแบบ ณ อุณหภูมิห้อง 25 องศาเซลเซียส แต่ในทางปฏิบัติการปรับแต่งทำได้ยากมาก ต้องใช้เครื่องมือวัดทางดิจิตอล คือ ดิจิตอลโวลต์มิเตอร์ ซึ่งมีย่านการวัดต่ำ ๆ เป็นมิลลิโวลต์

บทสรุป

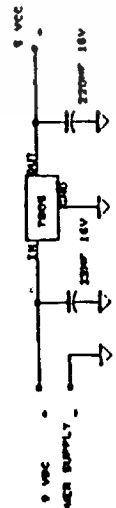
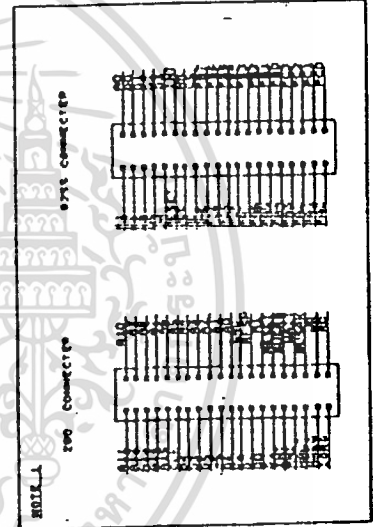
โครงงานนี้สามารถพัฒนาต่อไปได้อีก ให้ทำงานได้โดยถูกต้อง โดยเริ่มแรกการปรับปรุงในส่วนของวงจรขยาย ส่วนตรวจจับ (sensor amp) ซึ่งทำได้โดยการใช้อิซีที่ใช้ในงานอุตสาหกรรม คือ ไอซีเบอร์ AD 564 , AD 565 ซึ่งเป็นไอซีที่ใช้งานกับเทอร์โมคัปเปิล type j, type k สำหรับการพัฒนาดำเนินไป อาจนำ ไมโครคอมพิวเตอร์ PC มาใช้งาน

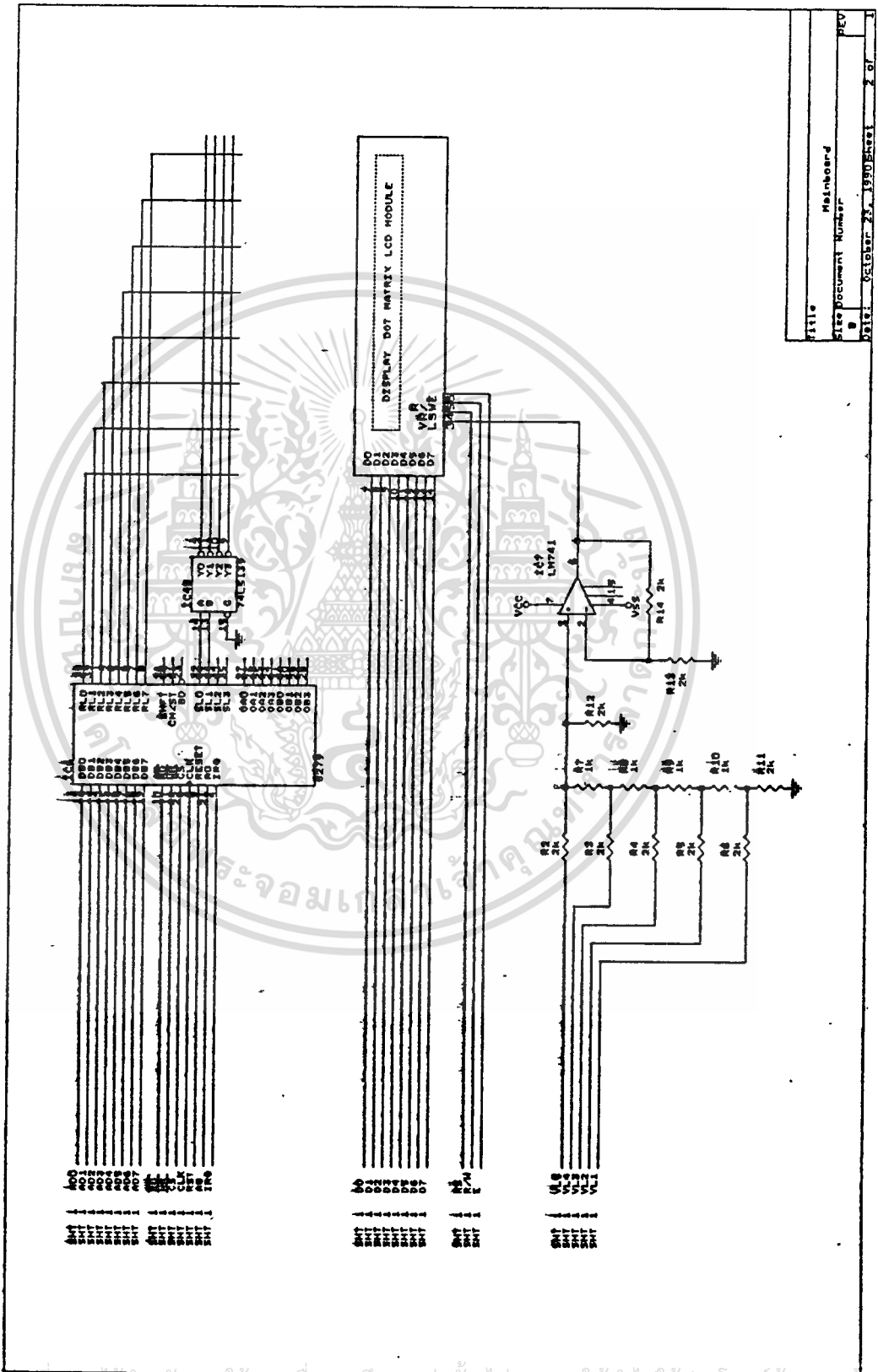


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



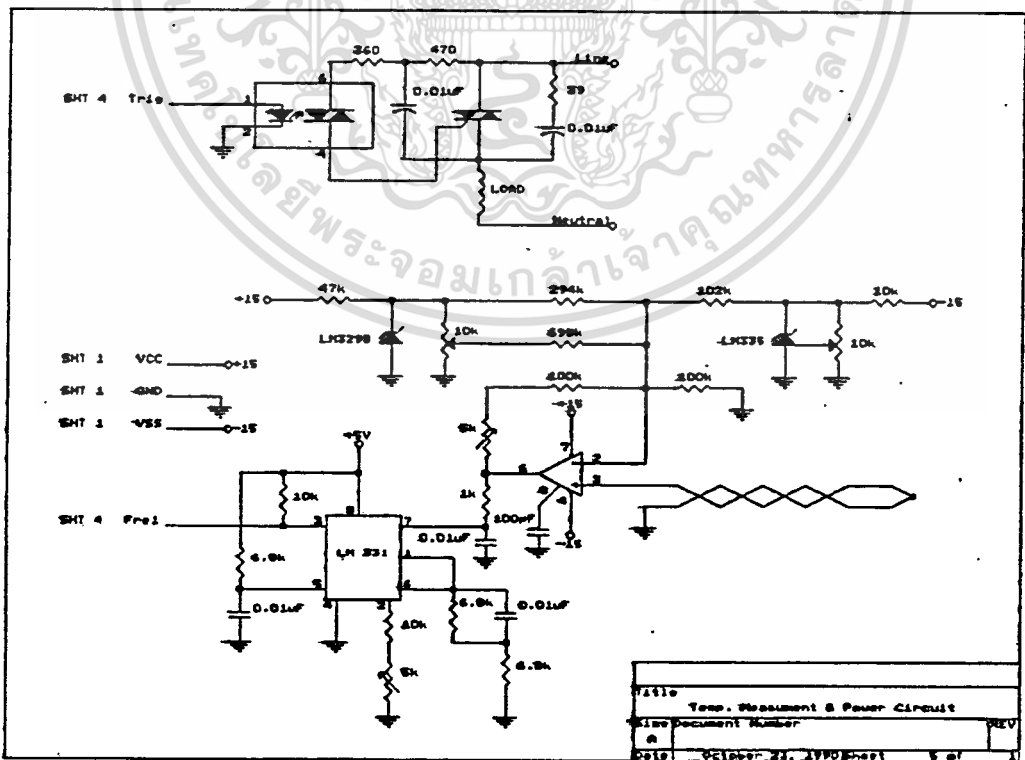
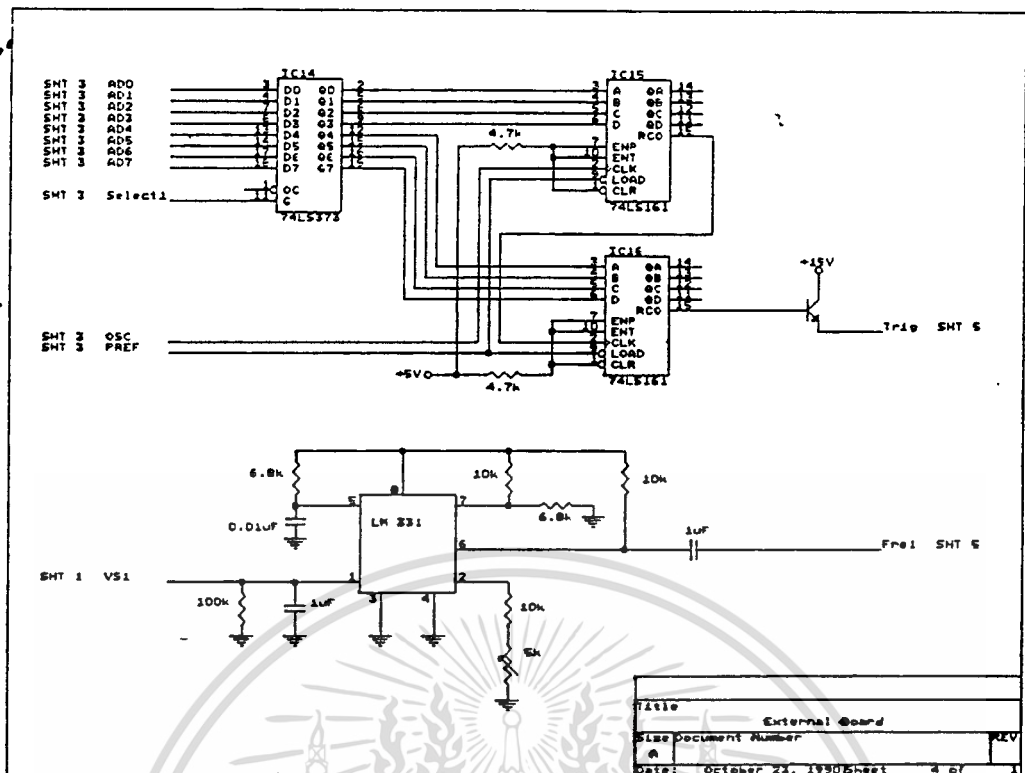
VCC 5 VCC
 U2 74181 ALU
 U3 74180 MAJORITY GATE
 U4 74184 BUS TRANSCEIVER
 U5 74185 PARITY GENERATOR



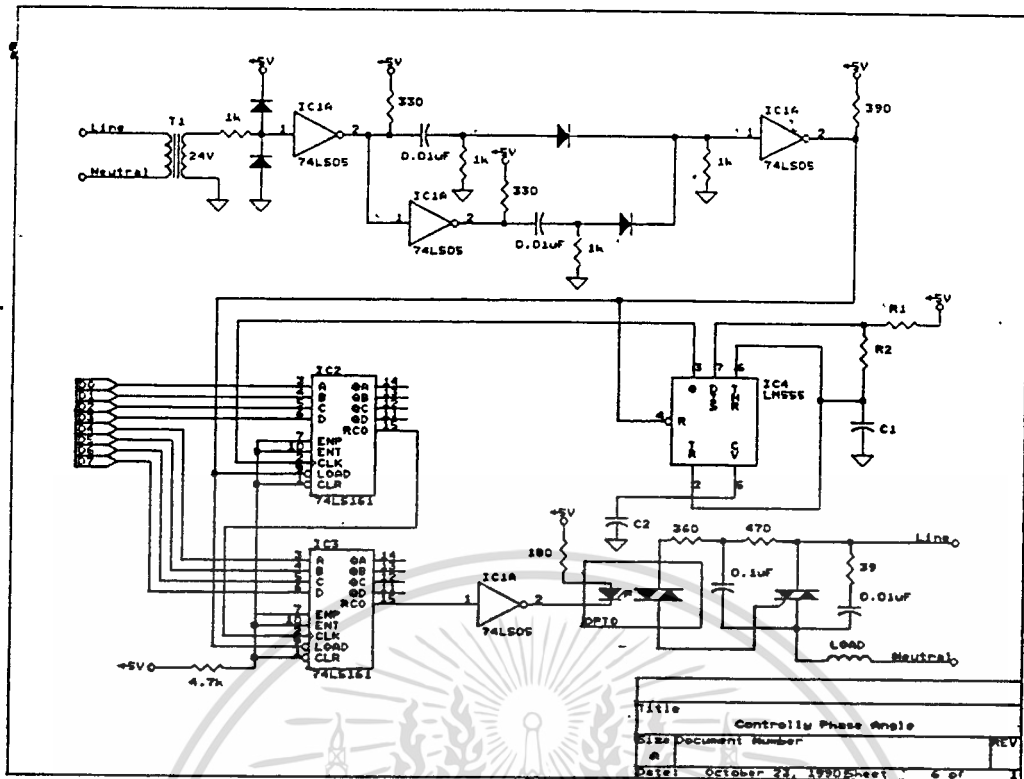


File	Mainboard
Site Document Number	0
REV	0
Date	October 23, 1990
Sheet	2 of 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้เผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Z80-CPU
Absolute Maximum Ratings

Table with 3 columns: Parameter, Specified operating range, and Comment. Parameters include Temperature Under Bias, Storage Temperature, Voltage On Any Pin, Power Dissipation, and various voltage and current limits.

Note: For Z80-CPU all AC and DC characteristics remain the same for the military grade parts except Icc. Icc = 100 mA

Z80-CPU D.C. Characteristics

TA = 0°C to 70°C, Vcc = 5V ± 5% unless otherwise specified

Table with 6 columns: Symbol, Parameter, Min, Typ, Max, Unit, Test Condition. Parameters include VILC, VIHc, VIL, VIH, VOL, VOH, ICC, ILI, ILOH, ILOL, ILD.

Capacitance

TA = 25°C, f = 1 MHz, unmeasured pins returned to ground

Table with 4 columns: Symbol, Parameter, Max, Unit. Parameters include Cc, CIN, COUT.

Z80-CPU Ordering Information

- C - Ceramic
P - Plastic
S - Standard 5V ± 5% 0° to 70°C
E - Extended 5V ± 5% -40° to 85°C
M - Military 5V ± 10% -55° to 125°C

Z80A-CPU D.C. Characteristics

TA = 0°C to 70°C, Vcc = 5V ± 5% unless otherwise specified

Table with 6 columns: Symbol, Parameter, Min, Typ, Max, Unit, Test Condition. Parameters include VILC, VIHc, VIL, VIH, VOL, VOH, ICC, ILI, ILOH, ILOL, ILD.

Capacitance

TA = 25°C, f = 1 MHz, unmeasured pins returned to ground

Table with 4 columns: Symbol, Parameter, Max, Unit. Parameters include Cc, CIN, COUT.

Z80A-CPU Ordering Information

- C - Ceramic
P - Plastic
S - Standard 5V ± 5% 0° to 70°C

We reprint data sheets on pages 7-D2 through 7-D13 by permission of Zilog, Incorporated.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Z80A-CPU

A.C. Characteristics

$T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = +5V \pm 5\%$, Unless Otherwise Noted.

Signal	Symbol	Parameter	Min	Max	Unit	Test Condition
ϕ	t_c (ϕ H)	Clock Period	.25	112	µsec	
	t_{wH} (ϕ H)	Clock Pulse Width, Clock High	110	1E	µsec	
	t_{wL} (ϕ L)	Clock Pulse Width, Clock Low	110	2000	µsec	
	$t_{f,1}$	Clock Rise and Fall Time		10	ns	
A_{0-15}	$t_{D(AD)}$	Address Output Delay		110	ns	$C_L = 50\text{pF}$
	$t_{F(AD)}$	Delay to Float		90	ns	
	t_{acm}	Address Stable Prior to $\overline{\text{MREQ}}$ (Memory Cycle)	111		ns	
	t_{aci}	Address Stable Prior to IORQ , RD or $\overline{\text{WR}}$ (I/O Cycle)	121		ns	
	t_{cst}	Address Stable from RD , $\overline{\text{WR}}$, IORQ or $\overline{\text{MREQ}}$	131		ns	
D_{0-7}	$t_{D(D)}$	Data Output Delay		150	ns	$C_L = 50\text{pF}$
	$t_{F(D)}$	Delay to Float During Write Cycle		90	ns	
	$t_{Sd(D)}$	Data Setup Time to Rising Edge of Clock During M1 Cycle	35		ns	
	$t_{Sf(D)}$	Data Setup Time to Falling Edge of Clock During M2 to M5	30		ns	
	t_{dcm}	Data Stable Prior to $\overline{\text{WR}}$ (Memory Cycle)	131		ns	
	t_{dci}	Data Stable Prior to RD (I/O Cycle)	101		ns	
	t_{cdf}	Data Stable From $\overline{\text{WR}}$	171		ns	
	t_H	Any Hold Time for Setup Time		0	ns	
$\overline{\text{MREQ}}$	$t_{DL\phi}(\overline{\text{M}})$	$\overline{\text{MREQ}}$ Delay From Falling Edge of Clock, $\overline{\text{MREQ}}$ Low		85	ns	$C_L = 50\text{pF}$
	$t_{DH\phi}(\overline{\text{M}})$	$\overline{\text{MREQ}}$ Delay From Rising Edge of Clock, $\overline{\text{MREQ}}$ High		85	ns	
	$t_{DL\phi}(\overline{\text{M}})$	$\overline{\text{MREQ}}$ Delay From Falling Edge of Clock, $\overline{\text{MREQ}}$ High		85	ns	
	$t_{w}(\overline{\text{MRL}})$	Pulse Width, $\overline{\text{MREQ}}$ Low	181		ns	
	$t_{w}(\overline{\text{MRH}})$	Pulse Width, $\overline{\text{MREQ}}$ High	191		ns	
IORQ	$t_{DL\phi}(\text{I})$	IORQ Delay From Rising Edge of Clock, IORQ Low		75	ns	$C_L = 50\text{pF}$
	$t_{DH\phi}(\text{I})$	IORQ Delay From Rising Edge of Clock, IORQ High		85	ns	
	$t_{DL\phi}(\text{I})$	IORQ Delay From Rising Edge of Clock, IORQ High		85	ns	
	$t_{DH\phi}(\text{I})$	IORQ Delay From Falling Edge of Clock, IORQ High		85	ns	
RD	$t_{DL\phi}(\text{R})$	RD Delay From Rising Edge of Clock, RD Low		85	ns	$C_L = 50\text{pF}$
	$t_{DL\phi}(\text{R})$	RD Delay From Falling Edge of Clock, RD Low		95	ns	
	$t_{DL\phi}(\text{R})$	RD Delay From Rising Edge of Clock, RD High		85	ns	
	$t_{DH\phi}(\text{R})$	RD Delay From Falling Edge of Clock, RD High		85	ns	
$\overline{\text{WR}}$	$t_{DL\phi}(\overline{\text{W}})$	$\overline{\text{WR}}$ Delay From Rising Edge of Clock, $\overline{\text{WR}}$ Low		65	ns	$C_L = 50\text{pF}$
	$t_{DL\phi}(\overline{\text{W}})$	$\overline{\text{WR}}$ Delay From Rising Edge of Clock, $\overline{\text{WR}}$ Low		80	ns	
	$t_{DL\phi}(\overline{\text{W}})$	$\overline{\text{WR}}$ Delay From Falling Edge of Clock, $\overline{\text{WR}}$ High		80	ns	
	$t_{w}(\overline{\text{WRL}})$	Pulse Width, $\overline{\text{WR}}$ Low	1101		ns	
$\overline{\text{M}}1$	$t_{DL}(\overline{\text{M}}1)$	$\overline{\text{M}}1$ Delay From Rising Edge of Clock, $\overline{\text{M}}1$ Low		100	ns	$C_L = 50\text{pF}$
	$t_{DH}(\overline{\text{M}}1)$	$\overline{\text{M}}1$ Delay From Rising Edge of Clock, $\overline{\text{M}}1$ High		100	ns	
RFSH	$t_{DL}(\text{R})$	RFSH Delay From Rising Edge of Clock, RFSH Low		130	ns	$C_L = 50\text{pF}$
	$t_{DH}(\text{R})$	RFSH Delay From Rising Edge of Clock, RFSH High		120	ns	
WAIT	$t_s(\text{WT})$	WAIT Setup Time to Falling Edge of Clock	98		ns	
HALT	$t_D(\text{HT})$	HALT Delay Time From Falling Edge of Clock		300	ns	$C_L = 50\text{pF}$
INT	$t_s(\text{IT})$	INT Setup Time to Rising Edge of Clock	80		ns	
$\overline{\text{NM}}1$	$t_w(\overline{\text{N}}1)$	Pulse Width, $\overline{\text{NM}}1$ Low	80		ns	
BUSRQ	$t_s(\text{BQ})$	BUSRQ Setup Time to Rising Edge of Clock	50		ns	
BUSAK	$t_{DL}(\text{BA})$	BUSAK Delay From Rising Edge of Clock, BUSAK Low		100	ns	$C_L = 50\text{pF}$
	$t_{DH}(\text{BA})$	BUSAK Delay From Falling Edge of Clock, BUSAK High		100	ns	
RESET	$t_s(\text{RS})$	RESET Setup Time to Rising Edge of Clock	60		ns	
	$t_F(\text{C})$	Delay to Float ($\overline{\text{MREQ}}$, IORQ , RD and $\overline{\text{WR}}$)		40	ns	
	t_{su}	$\overline{\text{M}}1$ Stable Prior to IORQ (Interrupt Ack.)	111		ns	

$$[12] t_c = t_{wH}(\phi H) + t_{wL}(\phi L) + t_f + t_r$$

$$[1] t_{acm} = t_{wH}(\phi H) + t_f + 85$$

$$[2] t_{aci} = t_c - 70$$

$$[3] t_{cst} = t_{wH}(\phi H) + t_f + 50$$

$$[4] t_{cdf} = t_{wH}(\phi H) + t_f + 45$$

$$[5] t_{dcm} = t_c - 170$$

$$[6] t_{dci} = t_{wH}(\phi H) + t_f + 170$$

$$[7] t_{cdf} = t_{wH}(\phi H) + t_f + 70$$

$$[8] t_w(\overline{\text{MRL}}) = t_c - 30$$

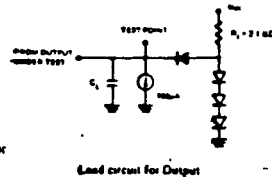
$$[9] t_w(\overline{\text{MRH}}) = t_{wH}(\phi H) + t_f + 20$$

$$[10] t_w(\overline{\text{WRL}}) = t_c - 30$$

$$[11] t_{su} = 2t_c + t_{wH}(\phi H) + t_f + 65$$

NOTES

- Data should be enabled onto the CPU data bus when RD is active. During interrupt acknowledge data should be enabled when $\overline{\text{M}}1$ and IORQ are both active.
- All control signals are internally synchronized, so they may be totally asynchronous with respect to the clock.
- The RESET signal must be active for a maximum of 3 clock cycles.
- Output Delay vs. Load Capacitance
 $T_A = 70^\circ\text{C}$, $V_{CC} = +5V \pm 5\%$
Add 10ns delay for each 50pF increase in load up to maximum of 200pF for data bus and 100pF for address & control lines.
- Although static by design, testing guarantees $t_{wH}(\phi H)$ of 300 psec maximum.



Z80A-CPU

A.C. Characteristics

$T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = +5V \pm 5\%$. Unless Otherwise Noted.

Signal	Symbol	Parameter	Min	Max	Unit	Test Condition
ϕ	t_c (PH)	Clock Period	.25	(12)	ns	
	t_w (PH)	Clock Pulse Width, Clock High	110	(13)	ns	
	t_w (PL)	Clock Pulse Width, Clock Low	110	2000	ns	
	t_r, f	Clock Rise and Fall Time		10	ns	
A_{0-15}	t_D (AD)	Address Output Delay		110	ns	$C_L = 50\text{pF}$
	t_F (AD)	Delay to Float		90	ns	
	t_{acm}	Address Stable Prior to MREQ (Memory Cycle)	111		ns	
	t_{ao}	Address Stable Prior to IORQ, RD or WR (I/O Cycle)	121		ns	
	t_{af}	Address Stable From RD, WR, IORQ or MREQ	131		ns	
D_{0-7}	t_D (D)	Data Output Delay		150	ns	$C_L = 50\text{pF}$
	t_F (D)	Delay to Float During Write Cycle		90	ns	
	t_{SD} (D)	Data Setup Time to Rising Edge of Clock During M1 Cycle	35		ns	
	t_{FD} (D)	Data Setup Time to Falling Edge of Clock During M2 to M5	50		ns	
	t_{acm}	Data Stable Prior to WR (Memory Cycle)	151		ns	
	t_{ao}	Data Stable Prior to WR (I/O Cycle)	161		ns	
	t_{df}	Data Stable From WR	171		ns	
t_H		Any Hold Time for Setup Time		0	ns	
MREQ	$t_{DL\bar{L}}$ (MR)	MREQ Delay From Falling Edge of Clock, MREQ Low		85	ns	$C_L = 50\text{pF}$
	$t_{DH\bar{H}}$ (MR)	MREQ Delay From Rising Edge of Clock, MREQ High		85	ns	
	$t_{DL\bar{H}}$ (MR)	MREQ Delay From Falling Edge of Clock, MREQ High		85	ns	
	t_w (MRL)	Pulse Width, MREQ Low	181		ns	
	t_w (MRH)	Pulse Width, MREQ High	191		ns	
IORQ	$t_{DL\bar{L}}$ (IR)	IORQ Delay From Rising Edge of Clock, IORQ Low		75	ns	$C_L = 50\text{pF}$
	$t_{DL\bar{H}}$ (IR)	IORQ Delay From Falling Edge of Clock, IORQ Low		85	ns	
	$t_{DH\bar{H}}$ (IR)	IORQ Delay From Rising Edge of Clock, IORQ High		85	ns	
	$t_{DL\bar{H}}$ (IR)	IORQ Delay From Falling Edge of Clock, IORQ High		85	ns	
	$t_{DH\bar{L}}$ (IR)	IORQ Delay From Falling Edge of Clock, IORQ High		85	ns	
RD	$t_{DL\bar{L}}$ (RD)	RD Delay From Rising Edge of Clock, RD Low		85	ns	$C_L = 50\text{pF}$
	$t_{DL\bar{H}}$ (RD)	RD Delay From Falling Edge of Clock, RD Low		95	ns	
	$t_{DH\bar{H}}$ (RD)	RD Delay From Rising Edge of Clock, RD High		85	ns	
	$t_{DL\bar{H}}$ (RD)	RD Delay From Falling Edge of Clock, RD High		85	ns	
	$t_{DH\bar{L}}$ (RD)	RD Delay From Falling Edge of Clock, RD High		85	ns	
WR	$t_{DL\bar{L}}$ (WR)	WR Delay From Rising Edge of Clock, WR Low		65	ns	$C_L = 50\text{pF}$
	$t_{DL\bar{H}}$ (WR)	WR Delay From Falling Edge of Clock, WR Low		80	ns	
	$t_{DH\bar{H}}$ (WR)	WR Delay From Rising Edge of Clock, WR High		60	ns	
	t_w (WRL)	Pulse Width, WR Low	1101		ns	
	t_w (WRH)	Pulse Width, WR High			ns	
INT	t_{DL} (INT)	INT Delay From Rising Edge of Clock, INT Low		100	ns	$C_L = 50\text{pF}$
	t_{DH} (INT)	INT Delay From Rising Edge of Clock, INT High		100	ns	
NFSH	t_{DL} (NFSH)	NFSH Delay From Rising Edge of Clock, NFSH Low		130	ns	$C_L = 50\text{pF}$
	t_{DH} (NFSH)	NFSH Delay From Rising Edge of Clock, NFSH High		120	ns	
WAIT	t_s (WAIT)	WAIT Setup Time to Falling Edge of Clock		70	ns	
HALT	t_D (HALT)	HALT Delay Time From Falling Edge of Clock		300	ns	$C_L = 50\text{pF}$
INT	t_s (INT)	INT Setup Time to Rising Edge of Clock		80	ns	
NMI	t_w (NMI)	Pulse Width, NMI Low		80	ns	
BUSRQ	t_s (BQ)	BUSRQ Setup Time to Rising Edge of Clock		50	ns	
BUSAK	t_{DL} (BA)	BUSAK Delay From Rising Edge of Clock, BUSAK Low		100	ns	$C_L = 50\text{pF}$
	t_{DH} (BA)	BUSAK Delay From Falling Edge of Clock, BUSAK High		100	ns	
RESET	t_s (RS)	RESET Setup Time to Rising Edge of Clock		60	ns	
	t_F (C)	Delay to Float (MREQ, IORQ, RD and WR)		80	ns	
	t_{su}	(M1 Stable Prior to IORQ (Interrupt Ack.))	(11)		ns	

$$[12] t_c = t_w(\text{PH}) + t_w(\text{PL}) + t_r + t_f$$

$$[1] t_{acm} = t_w(\text{PH}) + t_r - 65$$

$$[2] t_{ao} = t_c - 70$$

$$[3] t_{SD} = t_w(\text{PL}) + t_r - 50$$

$$[4] t_{af} = t_w(\text{PL}) + t_r - 45$$

$$[5] t_{dcn} = t_c - 170$$

$$[6] t_{dci} = t_w(\text{PL}) + t_r - 170$$

$$[7] t_{cdf} = t_w(\text{PL}) + t_r - 70$$

$$[8] t_w(\text{MRL}) = t_c - 30$$

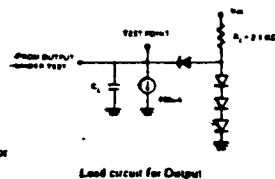
$$[9] t_w(\text{MRH}) = t_w(\text{PH}) + t_r - 20$$

$$[10] t_w(\text{WRL}) = t_c - 30$$

$$[11] t_{su} = 2t_c + t_w(\text{PH}) + t_r - 65$$

NOTES

- Data should be enabled onto the CPU data bus when RD is active. During interrupt acknowledge data should be enabled when INT and IORQ are both active.
- All control signals are internally synchronized, so they may be totally asynchronous with respect to the clock.
- The RESET signal must be active for a minimum of 3 clock cycles.
- Output Delay vs. Loaded Capacitance
 $T_A = 70^\circ\text{C}$ $V_{CC} = +5V \pm 5\%$
 Add 10ns delay for each 50pF increase in load up to maximum of 200pF for data bus and 100pF for address & control lines.
- Although not by design, testing guarantees $t_w(\text{PH})$ of 200 psec maximum.



A.C Timing Diagram

	-1-	0
CLOCK	V _{CC} -6V	45V
ON INPUT	2.0V	8V
INPUT	2.0V	8V
FLUENT	ΔV	±0.5V

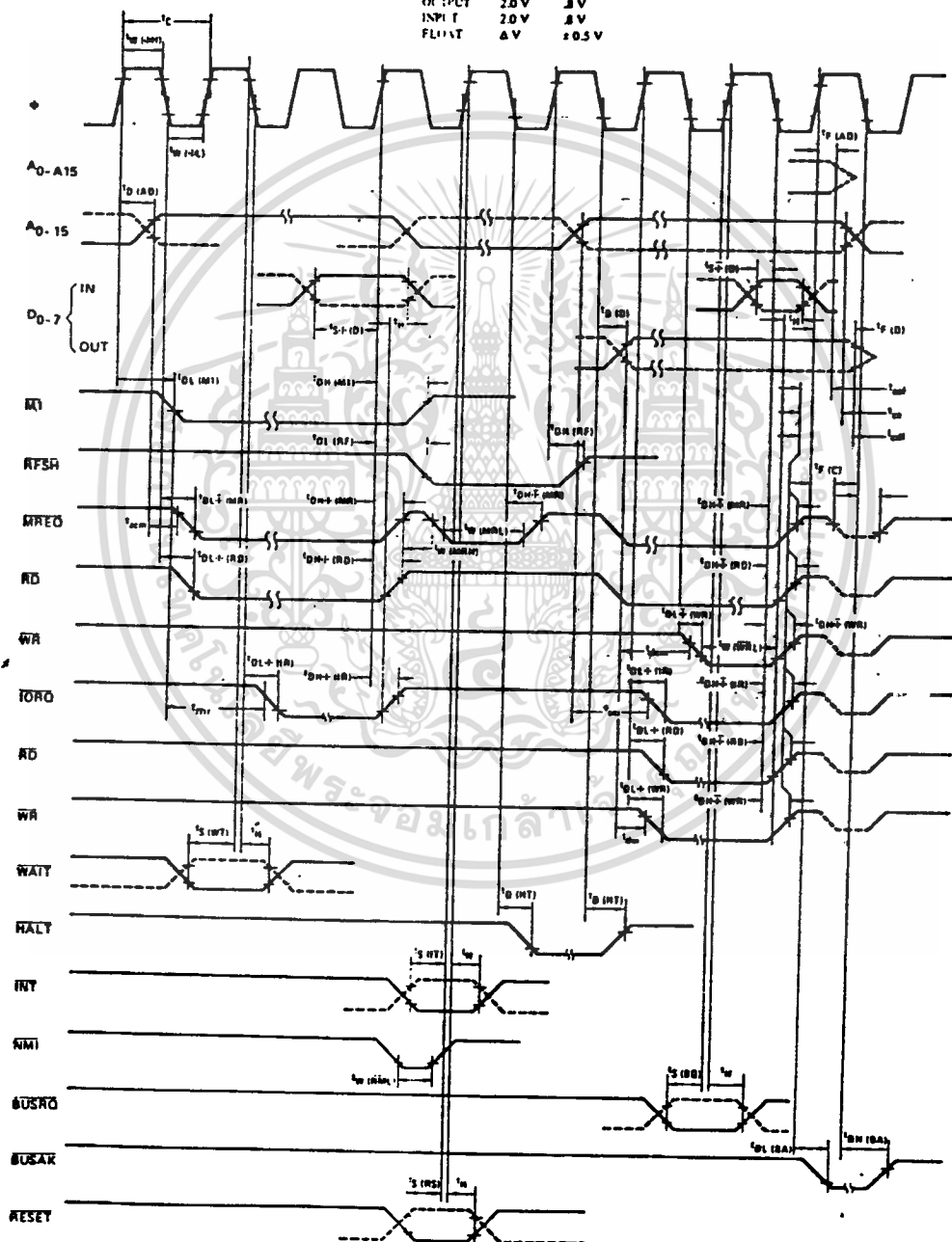


Table 7-3 A Summary of Instruction Object Codes and Execution Cycles with 8080A Mnemonics for Identical Instructions

INSTRUCTION	OBJECT CODE	BYTES	CLOCK PERIODS	8080A MNEMONIC	8080A CLOCK PERIODS
ADC data	CE yy	2	7	ACI data	7
ADC (HL)	8E	1	7	ADC M	7
ADC HL, d	ED 01xx1010	2	15		
ADC (IX + disp)	DD 8E yy	3	19		
ADC (IY + disp)	FD 8E yy	3	19		
ADC reg	10001xxx	1	4	ADC reg	4
ADD data	C6 yy	2	7	ADI data	7
ADD (HL)	86	1	7	ADD M	7
ADD HL, r	00xx1001	1	11	DAD r	10
ADD (IX + disp)	DD 86 yy	3	19		
ADD (IX, r)	DD 00xx1001	2	15		
ADD (IY + disp)	FD 86 yy	3	19		
ADD IY, r	FD 00xx1001	2	15		
ADD reg	10000xxx	1	4	ADD reg	4
AND data	E6 yy	2	7	ANI data	7
AND (HL)	A6	1	7	ANA M	7
AND (IX + disp)	DD A6 yy	3	19		
AND (IY + disp)	FD A6 yy	3	19		
AND reg	10100xxx	1	4	ANA reg	4
BIT b, r	CB	2	12		
	01bbb110				
BIT r, (IX + disp)	DD CB yy	4	20		
	01bbb110				
BIT r, (IY + disp)	FD CB yy	4	20		
	01bbb110				
BIT r, reg	CB	2	8		
	01bbbxxx				
CALL label	CD pppq	3	17	CALL label	17
CALL C, label	DC pppq	3	10/17	CC label	11/17
CALL A, label	FC pppq	3	10/17	CM label	11/17
CALL M, label	D4 pppq	3	10/17	CNC label	11/17
CALL NZ, label	C4 pppq	3	10/17	CNZ label	11/17
CALL P, label	F4 pppq	3	10/17	CP label	11/17
CALL PE, label	EC pppq	3	10/17	CPE label	11/17
CALL PO, label	E4 pppq	3	10/17	CPO label	11/17
CALL Z, label	CC pppq	3	10/17	CZ label	11/17
CMC	3F	1	4	CMC	4
CP data	FE yy	2	7	CPI data	7
CP (HL)	8E	1	7	CMP M	7
CP (IX + disp)	DD 8E yy	3	19		
CP (IY + disp)	FD 8E yy	3	19		
CP reg	10111xxx	1	4	CMP reg	19
CPI	ED A9	2	16		
CPIR	ED 89	2	21/16*		
CPI	ED A1	2	16		
CPIR	ED B1	2	21/16*		
CPL	2F	1	4	CMA	4
DAA	27	1	4	DAA	4
DEC (HL)	35	1	11	DCR M	10
DEC (IX)	DD 2B	2	10		
DEC (IX + disp)	DD 35 yy	3	23		
DEC IY	FD 2B	2	10		
DEC (IY + disp)	FD 35 yy	3	23		
DEC r	00xx1011	1	6	DCX r	5
DEC r	00xx1011	1	4	DCR reg	5
DI	F3	1	4	DI	4
DJNZ reg	10 yy	2	8/13		
EI	FB	1	4	EI	4
EX AF, AF'	0B	1	4		
EX AF, HL	EB	1	4	XCHG	4
FX (SP, HL)	E3	1	19	XTHL	18
EX (SP, IX)	DD E3	2	23		

Table 7-3. A Summary of Instruction Object Codes and Execution Cycles with 8080A Mnemonics
 *for Identical Instructions (Continued)

INSTRUCTION	OBJECT CODE	BYTES	CLOCK PERIODS	8080A MNEMONIC	8080A CLOCK PERIODS
EX CS,LY	FD E3	2	23		
EXX	D9	1	4		
HALT	76	1	4	HLT	4
IN 0	ED 46	2	8		
IN 1	ED 56	2	8		
IN 2	ED 5E	2	8		
IN A, port	D8 yy	2	10	IN port	10
IN reg,IC	ED	2	11		
	01ddd000				
INC (HL)	34	1	11	INR M	10
INC DX	DD 23	2	10		
INC (IX + disp)	DD 34 yy	3	23		
INC IY	FD 23	2	10		
INC (IY + disp)	FD 34 yy	3	23		
INC rp	00xx0011	1	6	INX rp	5
INC reg	00xx100	1	4	INR /reg	5
INR	ED AA	2	15		
INR	ED BA	2	20/15		
INR	ED A2	2	15		
INR	ED B2	2	20/15		
JP label	C3 ppqq	3	10	JMP label	10
JP C,label	DA ppqq	3	10	JC label	10
JP (HL)	E9	1	4	PCHL	5
JP (IX)	DD E9	2	8		
JP (IY)	FD E9	2	8		
JP M,label	FA ppqq	3	10	JM label	10
JP MC,label	D2 ppqq	3	10	JNC label	10
JP MZ,label	C2 ppqq	3	10	JNZ label	10
JP P,label	F2 ppqq	3	10	JP label	10
JP PE,label	EA ppqq	3	10	JPE label	10
JP PO,label	E2 ppqq	3	10	JPO label	10
JP Z,label	CA ppqq	3	10	JZ label	10
JR C,disp	38 yy	2	7/12		
JR disp	18 yy	2	12		
JR NC,disp	30 yy	2	7/12		
JR NZ,disp	20 yy	2	7/12		
JR Z,disp	28 yy	2	7/12		
LD (A,addr)	2A ppqq	3	13	LDA addr	13
LD (A,BC)	0A	1	7	LDAX B	7
LD (A,DE)	1A	1	7	LDAX D	7
LD A,J	ED 57	2	9		
LD A,R	ED 5F	2	9		
LD (addr),A	32 ppqq	3	13	STA addr	13
LD (addr),BC	ED 43 ppqq	4	20		
LD (addr),DE	ED 53 ppqq	4	20		
LD (addr),HL	22 ppqq	3	16	SHLD addr	16
LD (addr),IX	DD 22 ppqq	4	20		
LD (addr),IY	FD 22 ppqq	4	20		
LD (addr),SP	ED 73 ppqq	4	20		
LD (BC),A	02	1	7	STAX B	7
LD (DE),A	12	1	7	STAX D	7
LD (HL,addr)	2A ppqq	3	16	LHLD addr	16
LD (HL),data	38 yy	2	10	MVI M,data	10
LD (HL),reg	01110ass	1	7	MOV M,reg	
LD LA	ED 47	2	9		
LD IX(addr)	DD 2A ppqq	4	20		
LD IX,data16	DD 21 yyyy	4	14		
LD (IX + disp),data	DD 36 yy yy	4	19		
LD (IX + disp),reg	DD 01110ass	3	19		
	yy				
LD IY(addr)	FD 2A ppqq	4	20		
LD IY,data16	FD 21 yyyy	4	4		

Table 7-3. A Summary of Instruction Object Codes and Execution Cycles with 8080A Mnemonics for Identical Instructions (Continued)

INSTRUCTION	OBJECT CODE	BYTES	CLOCK PERIODS	8080A MNEMONIC	8080A CLOCK PERIODS
LD (IY + disp),data	FD 36 yyy	4	19		
LD (IY + disp),reg	FD 01110sss yy	3	19		
LD R,A	ED 4F	2	9		
LD reg,data	00ddd110 yy	2	7	MOV reg,data	7
LD reg,(HL)	01ddd110	1	7		
LD reg,(IX + disp)	DO 01ddd110	3	19	MOV reg,M	7
LD reg,(IY + disp)	yy FD 01ddd110	3	19		
LD reg,reg	01dddsss	1	4	MOV reg,reg	5
LD rp,(addr)	ED 01xx1011 ppqq	4	20		
LD rp,data16	00xx0001 yyyy	3	10	LXI rp,data16	10
LD SP,HL	F9	1	6	SPHL	5
LD SP,IX	DD F9	2	10		
LD SP,IY	FD F9	2	10		
LDD	ED A8	2	16		
LDDR	ED B8	2	21/16*		
LDI	ED A0	2	16		
LDIR	ED B0	2	21/16*		
NEG	ED 44	2	8		
NOP	00	1	4	NOP	4
ORA (HL)	86	1	7	ORA data	7
ORA (IX + disp)	DD 86 yy	3	19	ORA M	7
ORA (IY + disp)	FD 86 yy	3	19		
ORA reg	10110xxx	1	4	ORA reg	5
OTDR	ED B8	2	20/15*		
OTIR	ED B3	2	20/15*		
OUT (C),reg	ED 01sss001	2	12		
OUT port,A	D3 yy	2	11	OUT port	10
OUTD	ED A8	2	15		
OUTI	ED A3	2	15		
POP IX	DD E1	2	14		
POP IY	FD E1	2	14		
POP pr	11xx0001	1	10	POP rp	10
PUSH IX	DD E5	2	15		
PUSH IY	FD E5	2	15		
PUSH pr	11xx0101	1	11	PUSH rp	11
RES b,(HL)	CB 10bbb110	2	15		
RES b,(IX + disp)	DD CB yy 10bbb110	4	23		
RES b,(IY + disp)	FD CB yy 10bbb110	4	23		
RES b,reg	CB 10bbbxxx	2	8		
RET C	C9	1	10	RET	10
RET M	D8	1	5/11	RC	5/11
RET NC	F8	1	5/11	RM	5/11
RET NZ	DD	1	5/11	RNC	5/11
RET P	CD	1	5/11	RNZ	5/11
RET PE	FD	1	5/11	RP	5/11
RET PO	EB	1	5/11	RPE	5/11
RET Z	ED	1	5/11	RPO	5/11
RETI	CB ED 4D	2	14	RZ	5/11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Table 7-3 A Summary of Instruction Object Codes and Execution Cycles with 8080A Mnemonics for Identical Instructions (Continued)

INSTRUCTION	OBJECT CODE	BYTES	CLOCK PERIODS	8080A MNEMONIC	8080A CLOCK PERIODS
RETN	ED 45	2	14		
RL (HL)	CB 16	2	15		
RL (IX + disp)	DD CB yy 16	4	23		
RL (IY + disp)	FD CB yy 16	4	23		
RL reg	CB	2	8		
RLA	00010xxx				
RLC (HL)	17	1	4	RAL	4
RLC (IX + disp)	CB 06	2	15		
RLC (IY + disp)	DD CB yy 06	4	23		
RLC reg	FD CB yy 06	4	23		
RLCA	CB	2	8		
RLD	00000xxx				
RRC (HL)	07	1	4	RLC	4
RRC (IX + disp)	ED 6F	2	18		
RRC (IY + disp)	CB 1E	2	15		
RRC reg	DD CB yy 1E	4	23		
RRA	FD CB yy 1E	4	23		
RRC	CB	2	8		
RRA	00011xxx				
RRC (HL)	1F	1	4	RAR	4
RRC (IX + disp)	CB 0E	2	15		
RRC (IY + disp)	DD CB yy 0E	4	23		
RRC reg	FD CB yy 0E	4	23		
RRC	CB	2	8		
RRC	00001xxx				
RRC	0F	1	4	RRC	4
RRC	ED 67	2	18		
RST n	11xxx111	1	11	RST n	11
SBC data	DE yy	2	7	SBI data	7
SBC (HL)	9E	1	7	SBB m	7
SBC (IX + disp)	ED 01xx0010	2	15		
SBC (IY + disp)	DD 9E yy	3	19		
SBC reg	FD 9E yy	3	19		
SCF	10011xxx	1	4	SBB reg	4
SET b,HL	37	1	4	STC	4
SET b,IX + disp	CB	2	15		
SET b,IY + disp	11bbb110				
SET b,reg	DD CB yy	4	23		
SLA (HL)	11bbb110				
SLA (IX + disp)	FD CB yy	4	23		
SLA (IY + disp)	11bbb110				
SLA reg	FD CB yy	4	23		
SRA (HL)	11bbb110				
SRA (IX + disp)	CB	2	8		
SRA (IY + disp)	11bbbxxx				
SRA reg	CB 26	2	15		
SRL (HL)	DD CB yy 26	4	23		
SRL (IX + disp)	FD CB yy 26	4	23		
SRL (IY + disp)	CB 00100xxx	2	8		
SRL reg	CB 2E	2	15		
SUB data	DD CB yy 2E	4	23		
SUB (HL)	FD CB yy 2E	4	23		
SUB (IX + disp)	CB 00101xxx	2	8		
SUB (IY + disp)	CB 3E	2	15		
SUB reg	DD CB yy 3E	4	23		
SUB	FD CB yy 3E	4	23		
SUB	CB 00111xxx	2	8		
SUB	D6 yy	2	7	SUI data	7
SUB	96	1	7	SUB M	7
SUB	DD 96 yy	3	19		
SUB	FD 96 yy	3	19		
SUB	80010xxx	1	4	SUB reg	4
XOR data	EE yy	2	7	XRI data	7
XOR (HL)	AE	1	7	XRA M	7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กติการวมประกาศ

ขอขอบคุณคณาจารย์ทุกท่านที่ให้คำแนะนำปรึกษาข้อสงสัยต่างๆ และชี้แนะการแก้ปัญหาต่างๆที่เกิดขึ้น และ ขอขอบคุณเพื่อนร่วมงานที่ให้ความร่วมมือในการวิจัยผลงานนี้เป็นอย่างดี

ความดีของวิทยานิพนธ์เล่มนี้ขอมอบให้สถาบันแห่งนี้ที่ให้ความรู้และประสบการณ์ต่างๆแก่คณะผู้จัดทำวิทยานิพนธ์ฉบับนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

ยีน ภู่วรรณา วัฒนา เชียงกุล. "ไมโครโปรเซสเซอร์และไมโครคอมพิวเตอร์ Z80"
บริษัท ซีเอ็ดดูเคชั่น จำกัด. กรุงเทพฯ. พ.ศ. 2529

ซีเอ็ดดูเคชั่น "คู่มือ/เทียบเบอร์ ไอซีทีทีแอล" . บริษัท ซีเอ็ดดูเคชั่น จำกัด

ซีเอ็ดดูเคชั่น "คู่มือชุดวิชาไอซี". บริษัท ซีเอ็ดดูเคชั่น จำกัด

Barry B. Brey, "THE Z-80 Microprocessor Hardware ,Software"
Prentice-Hall New Jersey ,1988

J.MICHAEL JACOB, "INDUSTIAL CONTROL ELECTRONICS"
Prentice Hall, New Jersey, 1980