ปีการศึกษา 2533

ปริญญานิพนธ์ เรื่อง

PROGRAMMABLE SCORE BOARD DISPLAY

โดย

นายยุทธนา ปานมงคล นายไพบูลย์ รวมสุวรรณ

นายสุธิสาร สิริจันทรดิลก นายสมชัย ไอเว็ล

นายไพศาล แสนสิทธ นายเรืองศักดิ์ กลยาณชาติ

อาจารย์ที่ปรึกษา

อ. ประเสริฐ วชรพูลย์

027887

12 ก.ค. 2534

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ปริญญานิพนธ์ปีการศึกษา 2533

ภาควิชา เทคโนโลยีอุตสาหกรรม (ไทรคมนาคม)

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง PROGRAMMABLE SCORE BOARD DISPLAY

ผู้จัดทำ

1. นายยุทธนา ปานมงคล
2. นายไพบูลย์ รวมสุวรรณ
3. นายสุธสาร สิรจันทรดิลก
4. นายสมชัย ไอเว็ล
5. นายไพศาล แสนสุทธ
6. นายเรืองศักดิ์ กลยาณชาติ

..... อาจารย์ที่ปรึกษา

(อ. ประดิษฐ วชรพบูลย์)

..... กรรมการ

(ผศ. วิชัย สุรพงษ์)

..... กรรมการ

(ดร. ไพศาล นาคพงษ์)

..... กรรมการ

(อ. อุทัย ศรีธรวิโรจน์)

เลขหมู่ T 3305
เลขทะเบียน 02
วัน, เดือน, ปี 12

02

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAMMABLE SCORE BOARD DISPLAY

บทคัดย่อ

PROGRAMMABLE SCORE BOARD DISPLAY เป็นอุปกรณ์ซึ่งมีหน้าที่ในการแสดงภาพ ความ ตัวอักษร ที่เป็นภาษาไทยและภาษาอังกฤษ ที่มคุณสมบัติโปรแกรมข้อความต่าง ๆ ได้ตามความการของผู้ใช้งาน ข้อความหรือตัวอักษรที่ต้องการแสดงด้วย LED สามารถที่จะเก็บไว้ในหน่วยความมีขนาดความจุ 1000 ตัวอักษร อุปกรณ์นี้จะรับข้อความหรือตัวอักษรจากคีย์บอร์ด IBM โดยการแข้อความ หรือตัวอักษรจะถูกควบคุม รูปแบบและวิธีการแสดงภาพ ด้วยไมโครโปรเซสเซอร์ (Z

หลักการทำงานของเครื่อง PROGRAMMABLE SCORE BOARD DISPLAY คือ

1. ใช้คีย์บอร์ดในการป้อนข้อความหรือตัวอักษร เข้าไปเก็บในหน่วยความจำ
2. ใช้ฟังก์ชันและคอนโทรลคีย์ในการเลือกรูปแบบ (ภาษาไทยหรือภาษาอังกฤษ) หรืวิธีการแสดงข้อความ
3. การทำงานของอุปกรณ์ควบคุมโดยไมโครโปรเซสเซอร์ Z80
4. โปรแกรมควบคุมและโปรแกรมการใช้งาน ใช้ภาษาแอสเซมบลีย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAMMABLE SCORE BOARD DISPLAY

ABSTRACT

PROGRAMMABLE SCORE BOARD DISPLAY IS THE EQUIPMENT SHOW THE PICTU OF MESSAGE OR LETTER ON THE SCREEN. THE SCREEN DISPLAY ON BOTH THAI A ENGLISH LAUGUAGE WITH PROGRAMMABLE ACCESS BY USER. ALL CHARACTER A NUMERIC LETTERS ARE LOADED FROM STANDARD KEYBOARD INTO MEMORY (RAM) WI CAPACITY OF 1,000 CHARACTERS AND DISPLAY ON SCREEN BY CONTROLLING MICROPROCESSOR (Z80) WHEN USER PRESS PROGRAM KEYS.

THE FUNCTIONS OF PROGRAMMABLE SCORE BOARD EQUIPMENT DISPLAY

1) USE KEYBOARD TO FILL MESSAGE OR LETTER INTO MEMORY.
2) USE KEY AND CONTROL KEY ON KEYBOARD TO SELECT THE PROGRAM DISPLAY.
3) ALL PROGRAM DISPLAY FUNCTIONS CONTROLLED BY MICROPROCESSOR Z80.
4) USE ASSEMBLY LANGUAGE FOR PROGRAMMING.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทที่ 1	. บทนำ	1
บทที่ 2	. ทฤษฎีการทำงานของไมโครโปรเซสเซอร์และไมโครคอมพิวเตอร์	2
	. สถาปัตยกรรมของ Z-80	
บทที่ 3	. การสร้างและการทำงานของ Programmable Score Board	47
	Display	
	- คุณสมบัติเฉพาะของอุปกรณ์	
	- วงจรและหลักการทำงาน	
	- Controller Board	
	- Key Interface Board	
	- Led Display Board	
	- Power Supply Board	
	- วงจร	
	- โปรแกรมการใช้งาน	
บทที่ 4	. การทดลองและผลการทดลอง	
	- วิธีการใช้งานและคำสั่งควบคุม	
	- Lay out ของแผ่นปริ้นท์	
	- บทสรุป	
ภาคผนวก		
	- หนังสืออ้างอิง	
	- ข้อมูล IC SUPPORT ที่ใช้งาน	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในปัจจุบันการพัฒนาทางด้านอิเลคทรอนิกส์เป็นไปอย่างรวดเร็วมาก และมีแนวโน้มว่า
พัฒนาต่อไปอีกอย่างไม่หยุดยั้ง การพัฒนาต่าง ๆ เหล่านี้ทำให้เกิดอุปกรณ์ทางอิเลคทรอนิกส์ที่เป็นประ
อย่างมหาศาลแก่มวลมนุษย์ อาทิเช่น ไมโครอิเลคทรอนิกส์ ไมโครโปรเซสเซอร์ ไมโครคอมพิวเต
เป็นต้น

เป็นที่ยอมรับว่าการโฆษณาเป็นสื่อที่มีความสำคัญมากในการทำให้ผู้บริโภคตัดสินใจเลือ
สินค้าหรือบริการต่าง ๆ Programmable Score Board Display นี้ เป็นอุปกรณ์ชนิดหนึ่งที่ช่วย
ความสะดวกคล่องตัวในการสื่อข้อความที่ต้องการได้อย่างรวดเร็วและชัดเจน

โครงงานนี้เป็นการออกแบบสร้าง Programmable Score Board Display ที่สา
แสดง ภาพ ตัวอักษร ที่เป็นภาษาอังกฤษและภาษาไทย โดยแสดงภาพหรือข้อความด้วย LED เมทริ

คุณสมบัติในการทำงาน

1. สามารถป้อนข้อความ ลบข้อความได้โดยตรงจากคีย์บอร์ด
2. สามารถแสดงภาพที่ได้โปรแกรมเก็บไว้ใน RAM ได้
3. สามารถเลือกรูปแบบและวิธีการในการแสดงข้อความได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

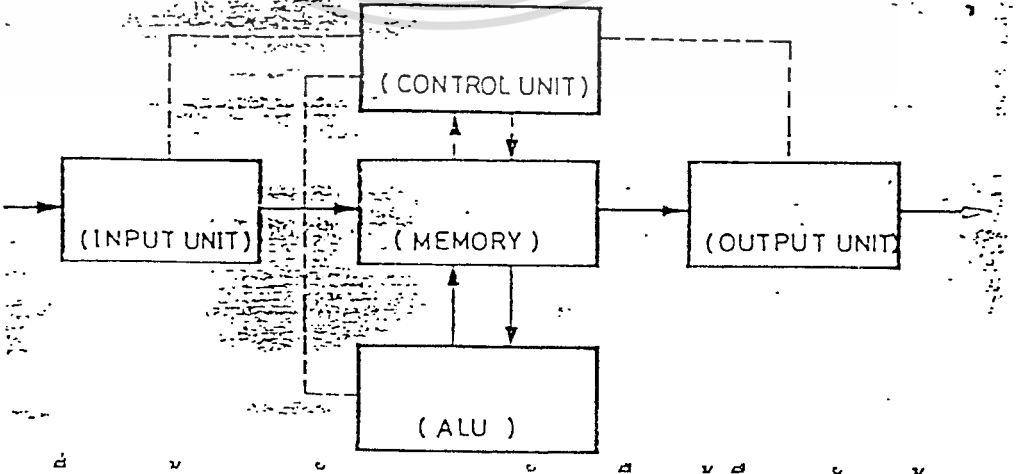
ทฤษฎีเกี่ยวกับไมโครโปรเซสเซอร์และไมโครคอมพิวเตอร์

ในระยะเวลาหลายปีที่ผ่านมา คอมพิวเตอร์ได้พัฒนาและเปลี่ยนแปลงไปอย่างมากและมีแนวโ
ถึงการเปลี่ยนแปลงใหม่ ๆ มากขึ้นในอนาคต ดิจิตอลคอมพิวเตอร์ที่มีอยู่ปัจจุบันสามารถทำงานได้เ
มีขนาดเล็กและเชื่อถือได้พร้อมทั้งราคาถูกกว่ารุ่นก่อน ๆ อีกด้วย เทคโนโลยีใหม่ ๆ และโครงสร้
ภายในมีความแตกต่างกันไปพร้อม ๆ กับอุปกรณ์อื่น ๆ เช่น หน่วยความจำ เป็นต้น ซึ่งได้พัฒนาอย่
รวดเร็ว จะทำให้คอมพิวเตอร์มีบทบาทต่อชีวิตคนเรามากขึ้นและเราก็จำเป็นจะต้องเรียนรู้เพื่อที่จะ
มัน

เครื่องคอมพิวเตอร์เป็นสมองอิเลคทรอนิคส์ที่ทำงานอย่างหนึ่ง ได้เร็ว และดีกว่าคนมาก
สามารถรับข้อมูลได้มากและแยกแยะได้เร็วกว่าคน ด้วยความเร็วนี้เอง งานชิ้นเดียวกันอาจจะต้
ใช้เวลาคนทำนานเป็นอาทิตย์เป็นเดือน แต่คอมพิวเตอร์อาจใช้ เวลาเพียงนาทีเดียวเท่านั้น ด้วยเ
นี้เราถึงใช้ความสามารถของคอมพิวเตอร์ มาประมวลข้อมูลจำนวนมาก ๆ ในเวลาสั้น ๆ และด้
ความเร็วและความสามารถสูงของคอมพิวเตอร์นี้ เราจึงนำมาใช้งานในด้านธุรกิจอุตสาหกรรม แ
ห้องทดลอง เพื่อลดรายจ่าย เช่น คอมพิวเตอร์กับการออกแบบ คอมพิวเตอร์กับนักวิทยาศาส
คอมพิวเตอร์กับอุตสาหกรรม คอมพิวเตอร์กับการพัฒนาบุคลากร เป็นต้น

ไมโครโปรเซสเซอร์และไมโครคอมพิวเตอร์

คอมพิวเตอร์โดยทั่ว ๆ ไป โครงสร้างพื้นฐานจะแบ่งออกเป็นส่วนต่าง ๆ ดังรูป 2.1



รูปที่ 2.1 เส้นประแสดงสัญญาณควบคุมและสัญญาณเลือก เส้นทึบแสดงสัญญาณข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นว่าระบบคอมพิวเตอร์ประกอบด้วย

1. หน่วยควบคุม (CONTROL UNIT)

2. หน่วยความจำ (MEMORY UNIT)

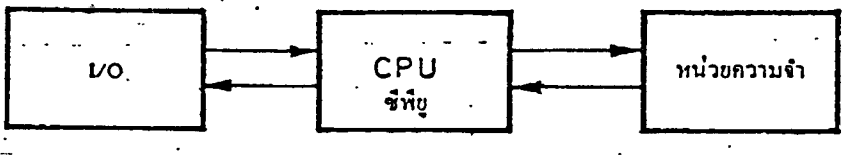
3. หน่วยคำนวณ (ARITHMATIC UNIT)

4. หน่วยส่งและรับสัญญาณ (I/O UNIT)

เครื่องจักรที่เราเรียกว่าคอมพิวเตอร์ จะต้องมีความสามารถทำการคำนวณทางคณิตศาสตร์และทำฟังก์ชั่นลอจิกได้ หน่วยที่รับผิดชอบด้านคำนวณในเครื่องคอมพิวเตอร์ คือหน่วยคำนวณ (ARITHMATIC AND LOGICAL UNIT) เมื่อหน่วยคำนวณจะสามารถฟังก์ชั่นใด ๆ ก็ต้องมีผู้บอกหรือควบคุม ซึ่งอาศัยหน่วยควบคุม (CONTROL UNIT) เมื่อทำการคำนวณก็ต้องมีข้อมูลที่จะนำมาคำนวณและคำนวณแล้วก็ต้องมีที่เก็บผล จึงจำเป็นต้องมีหน่วยความจำ (MEMORY UNIT) ซึ่งจะเป็นตัวเก็บและให้ข้อมูลแก่หน่วยคำนวณ จุดประสงค์ของการออกแบบคอมพิวเตอร์มาก็เพื่อการใช้งานซึ่งต้องติดต่อเอาข้อมูลจากภายนอกและยังต้องแสดงข้อมูลให้ภายนอกได้ด้วย จึงต้องมีหน่วยส่งและรับสัญญาณ (I/O UN

ที่กล่าวมานี้เป็นโครงสร้างทั่วไปของคอมพิวเตอร์แบ่งตามหน้าที่ เนื่องจากเทคโนโลยีด้านการผลิตอุปกรณ์อิเล็กทรอนิกส์ได้พัฒนาไปมาก ทำให้สามารถรวบรวมหน่วยต่าง ๆ ของคอมพิวเตอร์ขึ้นแก่หน่วยควบคุม (CONTROL UNIT) หน่วยความจำบางส่วนหน่วยคำนวณ (ALU) วงจรควบคุม I/O ส่วนทั้งหมดนำมาไว้ในแผ่นวงจรเดียวกัน เรียกว่า ไมโครโปรเซสเซอร์ (MICROPROCESSOR) ซึ่งสามารถทำหน้าที่ประมวลข้อมูลและควบคุมหน่วยอื่น ๆ ให้ทำงานไปด้วยกันได้ คอมพิวเตอร์ที่มีโปรเซสเซอร์เป็นหน่วยประมวลกลาง เราเรียกคอมพิวเตอร์นี้ว่า ไมโครคอมพิวเตอร์ (MICRO COMPUTE

ดังแสดงในรูป 2.2



รูป 2.2 ไมโครคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไมโครโปรเซสเซอร์เทคโนโลยี

เทคโนโลยีของแอลเอสไอ (LARGE SCALE INTEGRATION) ทำให้เราสามารถบรรจุ
วงจรอิเลคทรอนิกส์ลงไปบนแผ่นวงจรแผ่นเดียวได้ เรียกแผ่นวงจรนั้นว่า ชิพ (CHIP) เทคโนโลยี
LSI นี้ถูกนำมาในการผลิตไมโครโปรเซสเซอร์ที่จำหน่ายอยู่ในท้องตลาด โดยทั่วไปแบ่งออกตามตระกูล
หรือเทคโนโลยีที่นำมาใช้ดังต่อไปนี้

1. พวกไบโพลา พวกนี้ก็คือจักประกอบด้วยทรานซิสเตอร์ ซึ่งทำงานได้ด้วยกระแสทั้งสอง
ประเภทคือ ลบและบวก จึงเรียกว่าไบโพลา (BIPOLAR) ตระกูลนี้แบ่งเป็นพวกย่อย ๆ ตามลักษณะ
วงจรคือ

1.1 ECL (EMITTER COUPLING LOGIC)

1.2 IIL (INTEGRATED INJECTION LOGIC)

1.3 TTI (TRANSISTOR TRANSISTOR LOGIC)

1.4 STTL (SCHOTTKY TRANSISTOR TRANSISTOR LOGIC)

ทั้งสี่พวกนี้มีความแตกต่างกันเรื่องความเร็ว ภายในตระกูลไบโพลาเองความเร็วก็แตกต่างกัน แต่
โดยรวมแล้วตระกูลนี้มีความเร็วมากกว่าตระกูลอื่น

2. พวก MOS (METAL OXIDE SEMICONDUCTOR) พวกนี้ลอจิกเป็นทรานซิสเตอร์เช่นกัน แต่
เป็นทรานซิสเตอร์ชนิด Mosfet แบ่งออกเป็นพวกตามลักษณะวงจรและสารที่ทำคือ

PMOS (P-CHANNEL METAL OXIDE SEMICONDUCTOR)

NMOS (N-CHANNEL METAL OXIDE SEMICONDUTCTOR)

CMOS (COMOLEMENTARY METAL OXIDE SEMICONDUCTOR)

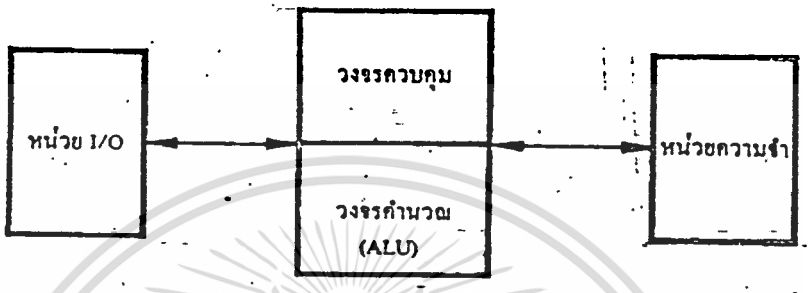
พวก MOS มีคุณสมบัติเด่นมากเรื่องกินกำลังไฟต่ำ

ในการเลือกไมโครโปรเซสเซอร์ที่เหมาะสมสำหรับงานต่าง ๆ จะต้องพิจารณาข้อมูลให้ละเอียด
จึงจะได้ระบบที่เหมาะกับงานมากที่สุด ขึ้นอยู่กับข้อจำกัดในการเลือกใช้เอง นอกจากเทคโนโลยี
ต่าง ๆ แล้วโครงสร้างภายในไมโครโปรเซสเซอร์ของแต่ละเบอร์ก็แตกต่างกันไป โครงสร้างภายใน
หมายถึง ขนาดของข้อมูล จำนวนคำสั่ง การตอบรับสัญญาณจากภายนอก ทั้งนี้จะต้องศึกษารายละเอียด
ของแต่ละเบอร์ต่าง ๆ กัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

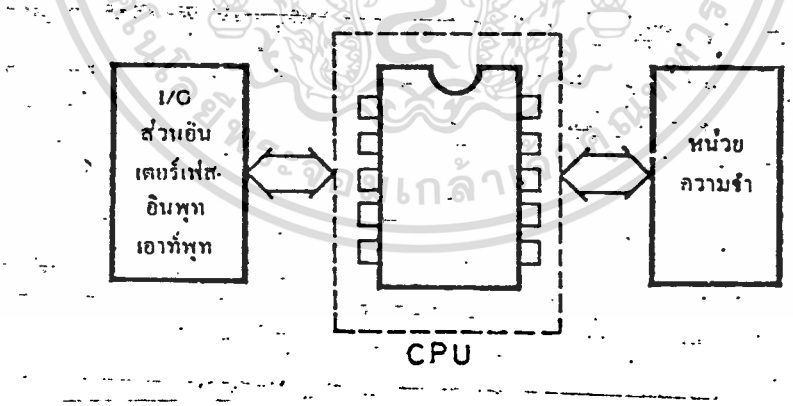
ระบบไมโครคอมพิวเตอร์

ไมโครคอมพิวเตอร์ คือ คอมพิวเตอร์ที่มีไมโครโปรเซสเซอร์ เป็นตัวประมวลผลกลาง ซึ่งมีหน้าที่ (CPU) และทำงานร่วมกับหน่วยอื่นอีก 2 หน่วยคือ หน่วยความจำ (MEMORY UNIT) หน่วยรับและสัญญาณเข้าออก (INPUT OUTPUT UNIT) เขียนเป็นบล็อคไดอะแกรมได้ดังนี้



รูป 2.3 บล็อคไดอะแกรมของไมโครคอมพิวเตอร์

หน้าที่ของ CPU นอกจากจะคำนวณทางคณิตศาสตร์แล้วยังส่งสัญญาณควบคุม หรือให้จังหวะแก่หน่วยอื่น ๆ ให้ทำงานไปพร้อม ๆ กับมันได้ด้วย ภายในตัว CPU แบ่งออกเป็นสองส่วนใหญ่ ๆ คือ วงจรควบคุม และวงจรคำนวณ เราสามารถแทนบล็อคไดอะแกรม CPU ด้วย รูปตัวจริงของไมโครโปรเซสเซอร์ได้

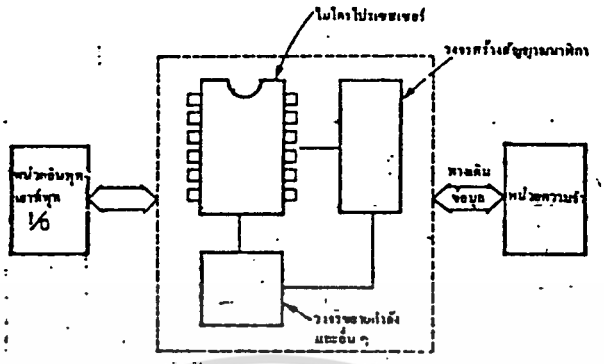


รูป 2.4

ดังรูป 2.4 เฉพาะตัวไมโครโปรเซสเซอร์จิต ๆ ก็ยังทำอะไรไม่ได้ ตัวมันเองทำงานเป็นสามขั้นจนวงจรก็จริง แต่ก็ยังต้องอาศัยวงจรส่วนประกอบอื่น ๆ ที่เพิ่มเติมเข้ามาเพื่อช่วยให้ CPU ทำงานได้ และติดต่อควบคุมหน่วยอื่น ๆ ได้ ส่วนที่เพิ่มเติมเข้ามาคือ วงจรสัญญาณนาฬิกา วงจรถอดรหัส วงจรขยายกำลัง ตามรูป 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรขยายกำลัง ตามรูป 2.5



รูป 2.5

ในระดับของวงจรต่าง ๆ ที่เกี่ยวข้องกับทางระบบ ไม่ได้จบสิ้นอยู่ที่ตัวไมโครโปรเซสเซอร์เท่านั้น วงจรประกอบอื่น ๆ ยังมีอีกมาก แต่ละวงจรมีหน้าที่สำคัญต่อระบบ เช่น เดียวกันจึงจะทำให้ไมโครโปรเซสเซอร์ติดต่อกับหน่วยควบคุมต่าง ๆ ได้ วงจรที่สำคัญที่ช่วยต่อเชื่อม CPU กับหน่วยอื่น ๆ เรียกว่าวงจรอินเตอร์เฟส (INTERFACE CIRCUIT) วงจรอินเตอร์เฟสมีตั้งแต่วงจรถอดรหัส วงจรขยายกำลังวงจรควบคุมการส่งผ่านสัญญาณ ความสลับซับซ้อนของวงจร INTERFACE จะมากเพียงใดขึ้นอยู่กับชนิดของหน่วยต่าง ๆ ที่ CPU ต้องติดต่อ

- หน่วย I/O หน่วย I/O ได้แก่ส่วนแสดงผลหรือใส่ข้อมูลที่เราใช้งาน มีมากมายหลายชนิด เช่นเครื่องพิมพ์ชนิดต่าง ๆ เครื่องเจาะเทป อ่านเทป ทีวีแสดงผลเป็นตัวหนังสือ (CRS) แป้นคีย์บอร์ด (KEY BOARD)

- หน่วยความทรงจำ เป็นส่วนเก็บคำสั่งและข้อมูลต่าง ๆ แยกแยะตามชนิดได้มากมาย เช่น หน่วยความจำแบบสารกึ่งตัวนำ (SEMICONDUCTOR MEMORY) หน่วยความจำขนาดใหญ่ ได้แก่แผ่นแม่เหล็ก (DISK) แบบอ่อนเรียกว่าฟลอบบีดิสก์ (FLOPPY DISK) แบบแข็งเรียกว่า ฮาร์ดดิสก์ (HARD DISK) เป็นต้น

จากรูป 2.5 มาเขียนแยกประเภทออกจะได้ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป 2.6 ระบบไมโครคอมพิวเตอร์

ดังกล่าวมาแล้วว่าไมโครคอมพิวเตอร์ประกอบด้วยหน่วยต่าง ๆ คือ หน่วยควบคุมหน่วยคำนวณหน่
คำนวณ หน่วยความจำ และหน่วยรับส่งข้อมูล วงจรอิเลคทรอนิคส์ที่ทำหน้าที่ของหน่วยต่าง ๆ นี้ เรียก
ฮาร์ดแวร์ (HARDWARE) วงจรเหล่านี้ทำหน้าที่ต่างกัน เช่นหน่วยควบคุมจะถูกออกแบบมาให้เข้าใจหรื
ความหมายรหัสที่อยู่ในรูปตัวเลข ชุดหนึ่งซึ่งสามารถเก็บไว้ในหน่วยความจำได้ รหัสนี้ถือว่าเป็นคำสั่ง
สำหรับคอมพิวเตอร์หรือเรียกว่า INSTRUCTION คอมพิวเตอร์เครื่องหนึ่งจะมีรหัสที่หน่วยควบคุมเข้าใ
อยู่ชุดหนึ่ง บางเครื่องมีมากกว่า 100 คำสั่ง คำสั่งหลาย ๆ คำสั่งสามารถบังคับให้คอมพิวเตอร์ทำอยู่
ก่อนหลังเป็นขั้นตอนจนได้คำตอบสุดท้ายที่เราต้องการชุดคำสั่งเราเรียกว่าโปรแกรม (PROGRAM) หรื
SOFTWARE ของเครื่องโปรแกรมสำหรับให้คอมพิวเตอร์ทำงานแต่ละงานย่อมไม่เหมือนกัน เพราะแต่ล
งานย่อมมีขั้นตอนปลีกย่อยแตกต่างกัน

หน่วยความจำ

หน่วยความจำหลักในระบบไมโครคอมพิวเตอร์เป็นความจำระบบอิเลคทรอนิคส์สามารถจำข้อมูล
ระบบอิเลคทรอนิคส์นี้ ข้อมูลหรือรหัสจะอยู่ในรูปเลขไบนารี หรือ เลขฐานสอง เลขไบนารี 1 หลัก เ
เรียกว่า 1 บิท เลขไบนารีหลาย ๆ บิท ที่คอมพิวเตอร์ถือเป็นคำสั่งหรือข้อมูลที่คอมพิวเตอร์รับเข้า
หรือส่งไป เราเรียกว่า คำ (WORD) จำนวนเลขไบนารีที่ประกอบเป็นรหัส 1 คำ คือเป็นความยาวข
คำหรือ WORD LENGTH ของคำนนคำประกอบด้วยเลข ไบนารี 8 หลักเราเรียกว่า 1 ไบท์ (BYT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเป็นขนาดของคำที่ไมโครคอมพิวเตอร์ทั่ว ๆ ไปใช้อยู่ หน่วยความจำจะสามารถเก็บรหัสไว้ได้เป็น
จำนวนมาก โดยแต่ละคำจะถูกเก็บไว้ในตำแหน่งใดตำแหน่งหนึ่งในหน่วยความจำ แต่ละตำแหน่งเป็น
อยู่ ที่สามารถอ้างอิงได้เรียกว่าที่อยู่ ตามรูปเป็นหน่วยความจำที่สามารถเก็บรหัสขนาดคำละ 8 บิ
ทั้งหมด 100 คำมีอยู่ตั้งแต่ 0 ถึง 99 สมมุติว่าระหัส M ถูกเก็บไว้ที่ แอดเดรส 20 เมื่อคอมพิวเตอร์
เรียกข้อมูลจากตำแหน่ง 20 คอมพิวเตอร์จะได้ M ภายในระบบไมโครคอมพิวเตอร์นอกจากจะมีหน่วย
คำนวณ หน่วยควบคุมและ หน่วยความจำปฐมภูมิแล้วยังมี วงจรความทรงจำ ซึ่งหน่วยคำนวณและหน่วย
ควบคุมใช้เป็นที่เก็บข้อมูลชั่วคราวอีกด้วย หน่วยความจำชั่วคราวนี้เรียกว่า รีจิสเตอร์ จำนวนรีจิสเตอ
ที่อยู่ในไมโครโปรเซสเซอร์แต่ละตัวไม่เท่ากัน โดยแต่ละรีจิสเตอร์ มีชื่อเรียกเฉพาะ เช่น รีจิสเตอร์
เป็นที่เก็บผลลัพท์ จากหน่วยคำนวนเราเรียก แอคคิวมูเลเตอร์

หน่วยคำนวณ

หน่วยคำนวณจะทำหน้าที่ทางคณิตศาสตร์ และลอจิกโดยตรง ฟังก์ชันพื้นฐานของ หน่วยคำนวนคอ
บวก ลบและฟังก์ชันทางลอจิก การบวก ลบ ตัวเลขแต่ละครั้งจะได้ผลลัพธ์ ซึ่งจะถูกเก็บไว้ในหน่วย
ความจำชั่วคราวเรียกว่าแอคคิวมูเลเตอร์ (ACCUMULATOR REGISTOR) การแสดงผลของการบวกที่
หน่วยคำนวนจะแสดงออกทางบิทแฟลก (FLAG BIT) ซึ่งก็คือ บิทใด ๆ หน่วยความจำแสดงสถานะผล
ซึ่งเราเรียกว่า STATUS REGISTOR

หน่วยควบคุม

หน่วยควบคุมจะเป็นหัวใจการดำเนินงานทั้งหมด โดยจะส่งสัญญาณควบคุมให้จังหวะแก่หน่วยอื่น
เช่น ส่งสัญญาณบังคับให้หน่วยคำนวณทำฟังก์ชันใดๆ หรือส่งสัญญาณไปอ่านคำสั่งที่ถูกเก็บไว้ในหน่วยคว
การทำงานของหน่วยควบคุมนคือ เมื่อเริ่มการทำงานหน่วยควบคุมจะรับคำสั่งจากหน่วยความจำแล้วแป
คำสั่งนั้น และส่งสัญญาณอื่นๆ เท่าที่จำเป็นไปควบคุม

การกระทำตั้งแต่หน่วยควบคุมเริ่มไปอ่านคำสั่งมาจากหน่วยความจำแล้ว นำมาถอดระหัสสจนถึง
คำสั่งชั่งลงไปเราเรียกว่าหนึ่งรอบคำสั่ง (INSTRUTION CYCLE) แต่ละรอบคำสั่งจึงประกอบด้วยขั้น
ย่อยๆ คือ สภาวะเฟทช์-คือขบวนการที่หน่วยควบคุมไปเอาคำสั่งมาจากหน่วยความจำแล้วถอดระหัส
คำสั่งนั้น สภาวะเอกซ์คิว คือขบวนการตั้งแต่หน่วยควบคุม เริ่มทำตามคำสั่งนั้นจนเสร็จคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่าในรอบคำสั่งหนึ่ง ๆ มีขั้นตอนรองลงมาคือรอบสถานะและรอบสภาวะเอกซ์คอท
จริงแล้วมันประกอบด้วยขั้นตอนย่อย ๆ ลงไปอีกคือ แมชีนไซเคิล (MACHINE CYCLE) ต่างๆ ในการ
บคุมหน่วยต่างๆ หน่วยควบคุมจะมีขั้นตอนพื้นฐานอยู่จำนวนหนึ่ง เช่นขบวนการอ่านข้อมูลจากหน่วย
ความจำ ขบวนการเขียนข้อมูลส่งไปหน่วยความจำขบวนการI/Oและอื่น ๆ ขบวนการย่อย ๆ นี้เรียกว่
แมชีนไซเคิล จะเห็นว่าไซเคิลการแฟทช์ เป็นการอ่านข้อมูลจากหน่วยความจำจึงต้องประกอบด้วยขบว
การพื้นฐานการอ่านข้อมูลการอ่านข้อมูลหรือ READ MACHINE CYCLE เป็นต้น

หน่วย I/O

หน่วยสุดท้ายที่จะกล่าวถึงคือหน่วย I/O ส่วนนี้เป็นส่วนที่ควบคุมจะติดต่อกับโลกภายนอก
หน่วย I/O เปรียบเสมือนตัวกลาง (BUFFER หรือกันชน) เมื่อคอมพิวเตอร์แสดงผลลัพท์เพื่อบันทึกใน
สื่อที่มนุษย์เข้าใจ หรือรับข้อมูลมาเก็บไว้ในหน่วยความจำ หน่วยควบคุมต้องส่งสัญญาณพร้อมข้อมูลมายังหน่
I/O ควบคุมอุปกรณ์ภายนอกอีกต่อหนึ่ง อุปกรณ์ภายนอกที่ใช้แสดงผลเรียกว่า อุปกรณ์ I/O

สถาปัตยกรรมของ Z-80

คำว่าไมโครคอมพิวเตอร์เป็นคำที่มความหมายถึงวงจรไมโครโปรเซสเซอร์ประกอบรวม
หน่วยความจำและอุปกรณ์อินพุท-เอาท์พุทไมโครโปรเซสเซอร์จึงเสมือนหัวใจสำคัญของระบบไมโคร
คอมพิวเตอร์ ปัจจุบันไมโครโปรเซสเซอร์มีหลายเบอร์ แต่ละเบอร์มีรายละเอียดและขีดความสามารถ
แตกต่างกันอยู่บ้างแต่โดยเนื้อแท้ของมัน เราจะพบสิ่งที่คล้ายกันในแง่ของโครงสร้างและสถาปัตยก
อย่างไรก็ตามการทำความเข้าใจกับไมโครโปรเซสเซอร์เบอร์ใดเบอร์หนึ่งให้ถ่องแท้ก็จะเป็นบันไดใน
เข้าใจไมโครโปรเซสเซอร์เบอรอื่นได้อย่างไม่ยากเย็นนัก

Z-80 ไมโครโปรเซสเซอร์มีวิวัฒนาการมาจากไมโครโปรเซสเซอร์เบอร์ 8080 กล่าว
หลังจากที่ไมโครโปรเซสเซอร์เบอร์ 8080 ได้ผลิตขึ้นและออกจำหน่ายซึ่งสร้างความตื่นเต้นต่อผู้ใช้งา
อย่างมากทำให้มีผู้ออกแบบระบบซอฟแวร์ไว้เป็นจำนวนมาก การแพร่หลายของไอซี 8080 แต่อย่างไร
ตาม 8080 มีจุดอ่อนหลายประการทั้งทางด้านฮาร์แวร์ที่วิศวกรของบริษัทไซล็อกซึ่งเป็นกลุ่มบุคคลที่เข้า
ปัญหาต่าง ๆ เหล่านี้เป็นอย่างดีเพราะเป็นวิศวกรที่แยกมาจากบริษัทอินเทล ได้ทำการออกแบบทั้งระ
ซอฟท์แวร์และฮาร์แวร์ของไมโครโปรเซสเซอร์เบอร์ใหม่ซึ่งยังคงแทนระบบซอฟท์ของ 8080 ได้และไ
ว่า Z-80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เหตุผลที่สำคัญประการหนึ่งของการนำเอา Z-80 มากล่าวถึงในโครงสร้างเพราะตัวโค

สร้างของ Z-80 เป็นโครงสร้างที่เข้าใจได้ง่ายทางฮาร์แวร์และการอินเตอร์เฟส ส่วนระบบทางซอฟ

นั้น Z-80 ก็มขีดความสามารถในการทำงานได้สูงกว่า 8080 ของอินเทลอีกมาก

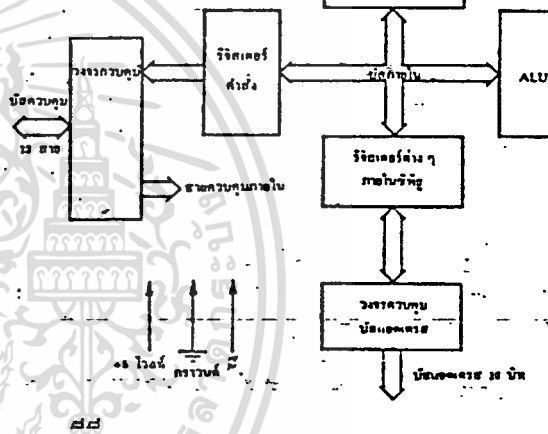
โครงสร้างของซีพียู

โครงสร้างของซีพียู Z-80 มีโครงสร้างที่พัฒนามาจาก 8080 ดังนั้นในแง่โครงสร้างฮา

จะเหมือนกับซีพียูของ 8080 แต่เนื่องจาก Z-80 มีการพัฒนามากขนทางซอฟแวร์ จึงทำให้มรายละเอ

แตกต่างเพิ่มเติมอีกหลายประการด้วยกัน บล็อกไดอะแกรมรูปที่ 2.7 เป็นไดอะแกรมแสดงให้เห็นโค

สร้างของ Z-80 โดยโครงสร้างของซีพียูจะบรรจุลงในแอลเอสไอขนาด 40 ขา

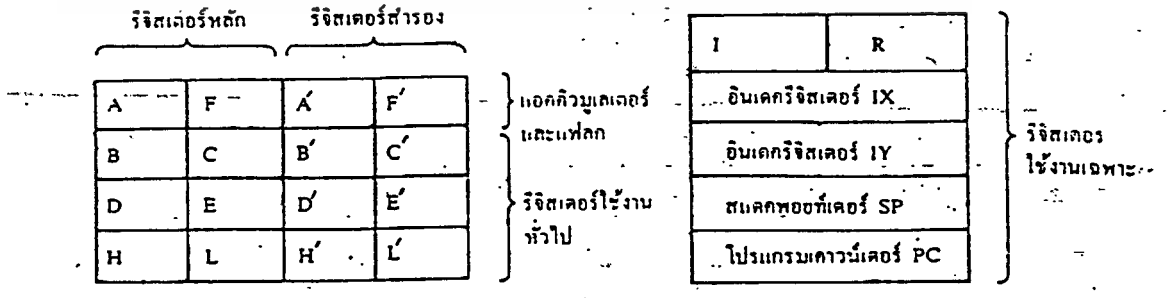


รูป 2.7 บล็อกไดอะแกรมซีพียู Z-80

โครงสร้างภายในของ Z-80 ซีพียูประกอบด้วยรจสเตอร์ภายในที่สามารถเขียนและอ่านได้ถึง 208 บ

โดยแยกเป็นกลุ่มของรจสเตอร์ขนาด 8 บิท 18 รจสเตอร์และรจสเตอร์ขนาด 16 บิทอีก 4 รจสเต

โดยมชุดรจสเตอร์แสดงไว้ดังรูป 2.8



รูป 2.8 แสดงรจสเตอร์ต่าง ๆ ที่มอยู่ใน Z-80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รจสเตอรหลกทใชงานทวไป

รจสเตอรในกลุมแรกกคอ A,F,B,C,D,E,H,L เปนรจสเตอรขนาด 8 บททใชงานโดย
ทวไปโดยรจสเตอรเหลานสามารถประกอบรวมกนเปนคูรจสเตอรไดคอ AF,BC,DE,HL โดยคูรจสเต
เหลานจะไดรบการใชงานในลกษณะของรจสเตอรขนาด 16 บท การกระทำภายในซพยูอาจจะอาศยเ
รจสเตอรเดยวหรอกระทำเปนคูรจสเตอรไดโดยท A คอแอดคมูเลเตอร แฟลกของ Z-80
จะมดวยกนทงหมด 6 ตว จงใชเพยง 6 บท แต Z-80 อาศยการเพมบทขนอก 2 บท และกลายเป
รจสเตอร F รจสเตอร F นสามารถไดรบการเซทรเซทการกระทำตามคำสงทางคณตศาสตรหรอลอจ
ได และเราสามารถใช F เหมอนรจสเตอรชงเมอรวมกนกบ A แลวจะกลายเปนรจสเตอร ขนาด 16
ได

กลุมรจสเตอรสำรอง

เปนรจสเตอรทสามารถเกบขอมลไดโดยเปนตวเกบขอมลทมาจากรจสเตอรหลก รจสเต
ชดนจงมดวยกน 8 ตว คอ A',F',B',C',D',E',H',L', รจสเตอรเหลานเปนรจสเตอรทใชกนใ
การเกบขอมลชวคราว ในการทตองการใชรจสเตอรหลกใชงานอยางอนกอน ดงนนรจสเตอรกลุมนจง
สามารถกระทำทางคณตศาสตรหรอลอจก

กลุมรจสเตอรทใชงานเฉพาะอยาง

- โปรแกรมเคาเตอร (PC-PROGRAM COUNTER) โปรแกรมเคาเตอร เปนรจสเตอรข
16 บท ทเปนตวกำหนดตำแหนงของโปรแกรมในขณะสภาวะการกระทำการเฟทช โดยขณะทำการเฟท
ทอยในโปรแกรมเคาเตอรจะไปปรากฏอยทแอดเดรสบสเมอชไปยงตำแหนงในหนวยความจำให CPU
อานคำสงมาตความหมาย ค่าทอยในโปรแกรมเคาเตอรจะเพมค่าขนได้อย่างอตโนมตหลงการกระทำ
การเฟทช แตถาหากซพยูกระทำคำสงใชขามไปยงตำแหนงอน (JUMP) ค่าแอดเดรสทจะกระโดดข้าม
จะโหลดเขามายงโปรแกรมเคาเตอรได้อย่างอตโนมต

- สแตคพอยทเตอร (SP-STACK POINTER) เปนรจสเตอรทมขนาด 16 บท ทใชสำหรบชไ
ยงแอดเดรสชนบนสดของสแตคทอยใน RAM โดยส่วนของสแตคมลกษณะโครงสรางเปนหนวยความจำแ
แบบเกบทหลงเรยกออกได้ก่อน ข้อมูลในสแตคอาจได้รับการพุทหรอพอยออกมาจากข้อมูลรจสเตอร์ภาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซีพียู ลักษณะของสแตคนี้ยังเป็นส่วนช่วยในการกระทำอินเตอร์รัพท์และการเรียกโปรแกรมย่อย กล

คือในการอินเตอร์รัพท์ค่าของโปรแกรมเคาเตอร์ จะได้รับการเก็บรักษาไว้ในชั้นของสแตค ครั้น

โปรแกรมกลับจากอินเตอร์รัพท์ไปกระทำยังโปรแกรมหลักก็นำค่าจากสแตคกลับเข้ามายังโปรแกรมเคา

เตอร์ในที่นี้ในทำนองเดียวกันการกระโดดไปยังโปรแกรมย่อย ก็เช่นเดียวกันดังนั้นการกระทำในรูปของ

อินเตอร์รัพท์หรือโปรแกรมย่อยสามารถซ้อนกันได้ไม่มีที่สิ้นสุด

- อินเด็กซ์รีจิสเตอร์ (IX, IY-INDEX REGISTOR) ซีพียู Z-80 มีอินเดดรีจิสเตอร์ขนาด

16 บิท 2 ตัว แต่ละตัวใช้ประโยชน์หลักในการทำหน้าที่เป็นตัวเก็บแอดเดรสฐาน (BASE ADDRESS)

เพื่อทำหน้าที่ยังแอดเดรสแบบอินเดดแอดเดรสซึ่ง (INDEX ADDRESSING) ในหมวดของอินเดรสนี้

คำสั่งขอมูลทดมากับคำสั่งที่เรียกว่าดิสเพลซเมนท์ (DISPLACEMENT) ซึ่งจะเก็บในรูปตัวเลข 2'S

คอมพลีเมนท์

- อินเตอร์รัพท์เพจแอดเดรสรีจิสเตอร์ (I-INTERRUPT PAPE ADDRESS REGISTER) ก

อินเตอร์รัพท์ของ Z-80 มีหลายหมวด และหมวดหนึ่งที่ทำให้การอินเตอร์รัพท์ของ Z-80 มีประสิทธิ

สูง กล่าวคือเมื่อเกิดการอินเตอร์รัพท์ในหมวดนี้ขณะนั้นสามารถอ้างแอดเดรสโดยทางอ้อมไปกระทำ

โปรแกรมในที่ใดก็ได้หน่วยความจำ โดยอาศัยค่าในรีจิสเตอร์ I รวมกับค่าที่ส่งมาจากอุปกรณ์เพอร์-

เฟอรัลอีก 8 บิท ชี้ไปยังค่าในรีจิสเตอร์ ในหน่วยความจำเพื่อนำค่านั้นมาโหลดเข้าในโปรแกรมเคา

เตอร์เพื่อกระทำต่อไป ด้วยวิธีการนี้เราจึงสามารถกระโดดเข้าไปทำที่ส่วนใดก็ได้ในหน่วยความจำ

- รีจิสเตอร์รีเฟรชหน่วยความจำ (R-MEMORY REFLESH) การต่อซีพียูกับหน่วยความจำนั้น

ปรกติจะต่อกับหน่วยความจำชนิดสแตติคได้โดยง่าย แต่อย่างไรก็ดี ชนิดไดนามิคที่ต้องการรีเฟรชมีราค

ถูกว่ามีความหนาแน่นสูงกว่า Z-80 ให้ข้อดีกว่าประการหนึ่งคือมันสามารถให้การรีเฟรชหน่วยความจำ

อย่างอัตโนมัติ โดยค่าในรีจิสเตอร์จะเพิ่มค่าขึ้นอีก 1 ทุกครั้งที่มีการกระทำการเฟทช์คำสั่งและข้อมูล

รีจิสเตอร์ R นี้จะส่งไปยังแอดเดรสบัสในส่วนบิทที่มีนัยสำคัญต่ำกว่าจังหวะของการส่งนี้จะเป็นจังหวะ

กันกับซีพียูส่งสัญญาณรีเฟรชออกมา ผู้โปรแกรมสามารถกำหนดค่าให้กับรีจิสเตอร์ R นี้ได้แต่ค่าในรีจิ

เตอร์นี้จะเรียกใช้โดยผู้โปรแกรมทางคำสั่งโดยตรงไม่ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

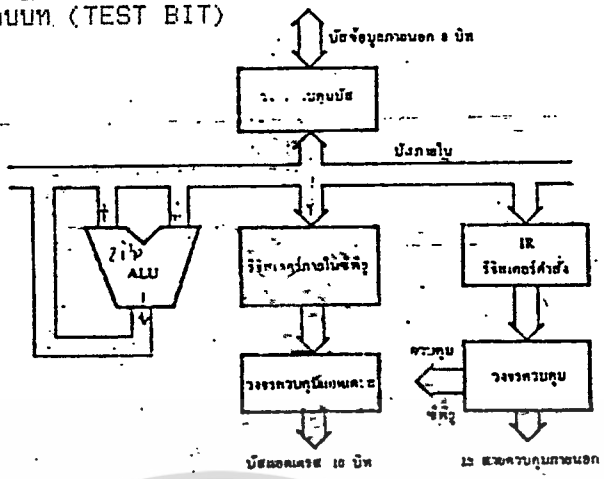
- แอคคิวมูเลเตอร์ (ACCUMULATOR) และแฟลก (FLAG) ซึ่งมีจะมีรีจิสเตอร์ที่ใช้เป็นหลักในการเป็นตัวโอเปอร์แรนด์สำหรับกระทำทางคณิตศาสตร์และลอจิก โดยรีจิสเตอร์หลักนี้จะมีเพียง 8 บิท เรียกว่าแอคคิวมูเลเตอร์ (ACCUMULATOR) การกระทำในส่วนของหน่วยคณิตศาสตร์และลอจิกย่อมเกิดเงื่อนไขได้หลายอย่างที่จะต้องแสดงสถานะภาพของเงื่อนไขเหล่านั้น เช่น เงื่อนไขผลลัพธ์เป็นศูนย์ผลลัพธ์เป็นบวกหรือลบ มีตัวทดหรือขอยืมในการกระทำทางคณิตศาสตร์แสดงเงื่อนไขพาริตี้หรือคี่ ฯลฯ สิ่งเหล่านี้จะแสดงผลลัพธ์ด้วยแฟลก (FLAG) แฟลกเป็นรีจิสเตอร์ขนาด 8 บิท ซึ่งสามารถรวมกับแอคคิวมูเลเตอร์เป็นรีจิสเตอร์ขนาด 16 บิทได้ ผู้โปรแกรมยังสามารถใช้คำสั่งในการเคลื่อนย้ายข้อมูลจากแอคคิวมูเลเตอร์ (A) และแฟลก (F) ไปเก็บไว้ใน A' และ F' ได้เพื่อให้การใช้งานของ A และ F มีประสิทธิภาพยิ่งขึ้น

- หน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU-ARITHMETIC AND LOGIC UNIT) การประมวลผลที่สำคัญของซีพียูของคอมพิวเตอร์ส่วนใหญ่จะมีหน่วยคำนวณทางคณิตศาสตร์และลอจิก (ALU) ส่วน ALU นี้จะนำข้อมูลซึ่งอาจจะนำมาจากนอกซีพียูหรือภายในซีพียูก็ได้มาประมวลผล การประมวลผลในส่วน ALU ที่สำคัญจะประกอบด้วย

  - การบวก (ADD)
  - การลบ (SUBSTRACT)
  - ลอจิก (AND)
  - ลอจิก (OR)
  - ลอจิก (EX-OR)
  - เปรียบเทียบ (COMPARE)
  - การเลื่อนไปทางซ้ายหรือขวา (SHIFT)
  - การเพิ่มค่า (INCREMENT)
  - การลดค่า (DECREMENT)
  - การเซทบิท (SET BIT)
  - การรีเซทบิท (RESET BIT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- การทดสอบบิท (TEST BIT)



รูป 2.9 แสดงการทำงาน ALU ภายใน

รจสเตอร์คำสั่งและส่วนควบคุม (INTRUCTION REGISTER AND CONTROL) ในการ

กระทำการเฟทช์ซีพียู จะอ่านคำสั่งจากหน่วยความจำเป็นส่วนของโปรแกรมโดยรอคำสั่งนั้นมาเก็บไว้ที่

IR เพื่อทำการถอดรหัสคำสั่งและส่งสัญญาณควบคุมการทำงานภายในซีพียูหรือควบคุมการทำงานของระ

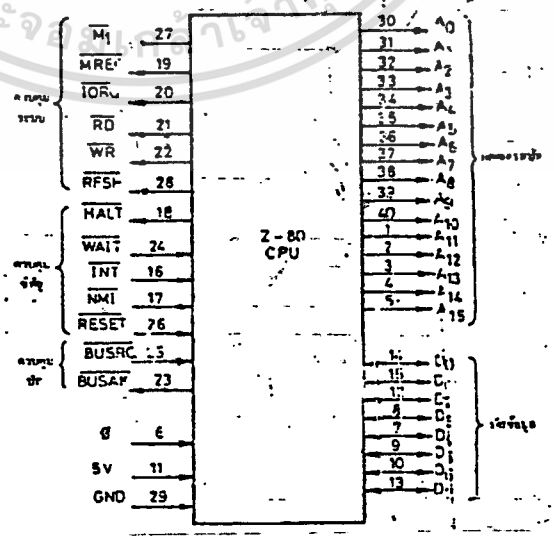
สัญญาณควบคุมเหล่านี้ออกมาในจังหวะต่าง ๆ กันเพื่อใช้ควบคุมระบบในการทำงานต่อไป

การจัดขาของ Z-80

Z-80 ซีพียูเป็นไอซีไมโครโปรเซสเซอร์ขนาดเพียง 40 ขา โดยหลักการแล้ว Z-80 เป็

ซีพียูได้โดยสมบูรณ์ กล่าวคือ Z-80 ไม่ต้องประกอบกับอุปกรณ์อื่นที่แยกการทำงานเพื่อรวมเป็นซีพียู ส่วน

ของสัญญาณจะประกอบด้วยบัสแอดเดรสบัสข้อมูลและสัญญาณควบคุมการจัดขาแสดงในรูปที่ 2.10



รูปที่ 2.10 ลักษณะของขาไอซี Z-80 ซีพียู

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของขาต่าง ๆ แสดงได้ดังต่อไปนี้

- A0-A15 บัสแอดเดรสสัญญาณที่ออกมาจากไอซีเหล่านี้จะให้แอดเดรสมีค่า HIGH โดยขาเหล่านี้เป็นเอาต์พุทแบบไตรสเตท บัสแอดเดรสมีด้วยกันทั้งหมด 16 สาย เพื่อให้ซีพียูติดต่อกับหน่วยความจำได้ถึง 2 = 64K ไบท์ นอกจากนี้ส่วนของแอดเดรสยังเป็นตัวกำหนดเบอร์พอร์ทของอุปกรณ์อินเอาต์พุท โดยขณะที่ซีพียูกระทำคำสั่งเกี่ยวกับอินพุทหรือเอาต์พุท ค่าของแอดเดรสบัสในบิทล่าง (A0-A จะแสดงค่าเบอร์พอร์ท ดังนั้น เราจึงมีอุปกรณ์อินพุทหรือเอาต์พุทได้ทั้งหมด 2 = 256 พอร์และในขณะเวลารีเฟท โดยเมื่อสัญญาณรีเฟทปรากฏขึ้นที่ตรีเฟรช (RFSH) ค่าในแอดเดรสบัส A0-A7 จะแสดงค่าแอดเดรสของหน่วยความจำที่จะได้รับการกระทำการรีเฟทช์

- D0-D7 บัสข้อมูล (DATA BUS) เป็นลักษณะบัสแบบสองทิศทาง Z-80 ซีพียูมีบัสข้อมูล 8 บัสข้อมูลเป็นเส้นทางผ่านของข้อมูลระหว่างซีพียูกับหน่วยความจำ ซีพียูกับอุปกรณ์อินพุทหรือการติดต่อระหว่างอุปกรณ์อินพุทเอาต์พุทกับหน่วยความจำ

- M1 (MACHINE CYCLE ONE) มีลักษณะเป็นแอคตีฟที่ลอจิก "0" เป็นส่วนที่จะบอกให้ทราบขณะนี้ซีพียูกำลังอยู่ในสภาวะรีเฟทช์ ในขณะที่ซีพียูรีเฟทช์คำสั่งที่มีออฟโค้ดสองไบท์ส่วนของ M1 จะสร้างขึ้นขณะรีเฟทช์ในแต่ละไบท์ลักษณะของคำสั่งที่มีออฟโค้ดสองไบท์จะขึ้นต้นด้วย CBH, DDH, EDH, FDH นอกจากนี้ M1 ยังสร้างสัญญาณร่วมกับ IORQ เพื่อบอกสถานะการตอบรับการอินเตอร์รัพท์

- MREQ (MEMORY REQUEST) เป็นลักษณะไตรสเตทในลอจิกแอคตีฟที่ "0" เป็นสัญญาณที่บอกให้ทราบซีพียูต้องการเขียนหรืออ่านหน่วยความจำตามแอดเดรสที่ปรากฏอยู่ในแอดเดรสบัส

- IORQ (INPUT OUTPUT REQUEST) เป็นเอาต์พุทลักษณะไตรสเตทให้ลอจิกแอคตีฟที่ "0" เป็นสายสัญญาณที่บอกให้ทราบว่า ซีพียูต้องการติดต่อกับอุปกรณ์อินพุท-เอาต์พุทโดยแอดเดรสบัส 8 บิทล่างจะให้แสดงค่าเบอร์พอร์ทส่วนบัสข้อมูลจะมีการส่งถ่ายระหว่างซีพียูกับ I/O นอกจากนี้ ถ้าเกิดขึ้นพร้อมสัญญาณ M1 เป็นตัวบอกถึงสถานะที่ซีพียูกำลังตอบสนองผลการอินเตอร์รัพท์ โดยขณะนี้ส่วนของบัสข้อจะมีการส่งผ่านเข้ามาด้วยค่าของอินเตอร์รัพท์เวคเตอร์

- RD (MEMORY READ) เป็นเอาต์พุทไตรสเตทและแอคตีฟขณะลอจิก "0" เป็นตัวบอกว่าขณะนี้ซีพียูต้องการผ่านข้อมูลจากหน่วยความจำหรืออุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- WR (MEMORY WRITE) เป็นขาเอาต์พุทแบบไตรสเตทและแอคตีฟขณะลอจิก "0" เป็นสัญญ

บอกว่าซีพียูต้องการเขียนข้อมูลโดยจะเขียนข้อมูลในตำแหน่งที่ ADDRESS BUS กำหนดขึ้นอาจจะเป็น

หน่วยความจำหรืออุปกรณ์ I/O ก็ได้

- RESH (REFRESH) เป็นขาเอาต์พุทแอคตีฟขณะลอจิก "0" RESH เป็นสัญญาณบอกให้ทราบ

สัญญาณในแอดเดรสบัสในส่วน A0-A6 เป็นแอดเดรสที่ใช้ในการรีเฟรชหน่วยความจำแบบไดนามิกส่วนบ

A7 จะเป็น "0" ส่วน A15-A8 จะแสดงค่าของรีจิสเตอร์ I

- HALT (HALT STATE) เป็นเอาต์พุทแอคตีฟด้วยลอจิก "0" สัญญาณจะแสดงเมื่อซีพียูได้กร

ทำคำสั่ง HALT และจะหยุดรอจนกว่าจะมีการอินเตอร์รัพท์หรือรีเซทขณะที่อยู่ในช่วง HALT ซีพียูจะเหม

กระทำคำสั่ง NOP (NO OPERATION) เพื่อให้เกิดไซเคิลในการทำงานเพื่อสัญญาณไปกระทำการรีเฟร

หน่วยความจำชนิด ไดนามิกส์

- WAIT (WAIT) เป็นขาอินพุทจะแอคตีฟด้วยลอจิก "0" WAIT เป็นตัวกำหนดแสดงเพื่อบอก

ซีพียูให้ซีพียูหยุดรอในกรณีอุปกรณ์อินพุท-เอาต์พุท หรือหน่วยความจำไม่สามารถรับหรือส่งข้อมูลได้ทัน

WAIT จะเป็นตัวทำให้ซีพียู ซึ่งคิดพอดีกับอุปกรณ์อินพุท-เอาต์พุททำงานด้วยความเร็วช้า

- INT (INTERRUPT REQUEST) เป็นขาอินพุทแอคตีฟด้วยลอจิก "0" INT เป็นสัญญาณที่สร้าง

ขึ้นมาจากอุปกรณ์อินพุท-เอาต์พุท เพื่อต้องการที่จะอินเตอร์รัพท์ซีพียูจะทำการตรวจสอบสัญญาณนี้ทุก ๆ

ครั้งที่จบการกระทำแต่ละคำสั่งการตอบสนองของการอินเตอร์รัพท์สามารถควบคุมได้โดยซอฟแวร์ด้วย

การเซทค่าอินเตอร์รัพท์ฟลิพฟลอบ (IFF) การตอบอินเตอร์รัพท์ซีพียูจะสร้างสัญญาณตอบด้วยการสร้าง

สัญญาณ IORQ ระหว่างช่วงเวลา M1 การตอบสนองต่อการอินเตอร์รัพท์มีแยกแยะให้ 3 แบบซึ่งจะ

อธิบายในรายละเอียดต่อไป

- NMI (NONMASKABLE INTERRUPT) เป็นขาอินพุทที่จะบอกซีพียูในขณะของพัลซ์ขาลงการ

อินเตอร์รัพท์ด้วยวิธี ซีพียูจะให้คำสั่งสูงกว่า INT กล่าวคือบัสมันจะตอบสนองและกระทำทันทีด้วยการ

เริ่มเอกซ์คิว คำสั่งในตำแหน่ง 0066H โดยอัตโนมัติการกระโดดไปกระทำในกรณีนี้ซีพียูจะเก็บค่า

โปรแกรมเคาน์เตอร์เดิมไว้ในสแตคเพื่อจะได้กลับไปทำงานเดิมเมื่อเสร็จสิ้นการอินเตอร์รัพท์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- RESET (RESET) เป็นขาอินพุทแอคตีฟด้วยลอจิก "0" การรีเซทในกรณีจะมีผลดังนี้

1. ค่าของ PC มีค่าเป็น "0"

2. IFF จะได้รับการ DISABLE

3. รีจิสเตอร์ I จะมีค่า 00H

4. รีจิสเตอร์ R จะมีค่า 00H

5. จะมีการเซทอินเตอร์รัพท์โหมดมาอยู่ที่โหมดที่ 0

ระหว่างการรีเซทสายแอดเดรสบัสและข้อมูลจะได้รับการกระทำให้มีค่าอิมพีแดนซ์สูงเพื่อแยกออกจากซีพียู ส่วนสายสัญญาณควบคุมจะได้รับการทำให้เป็นสัญญาณที่ไม่แอคตีฟ รีเฟรชจะไม่เกิดขึ้น

- BUSRQ (BUS REQUEST) เป็นขาอินพุทแอคตีฟด้วยลอจิก "0" BUSRQ' เป็นสัญญาณที่ส่งบอกกับซีพียูเพื่อต้องการให้ซีพียูควบคุมบัส กล่าวคือต้องการให้ซีพียูทำให้บัสแอดเดรสและข้อมูลอยู่ในสถานะอิมพีแดนซ์สูงคือต้องการแยกซีพียูออกจากบัสนั่นเอง

- BUSAK (BUS ACKNOWLEDGE) เป็นขาเอาท์พุทแอคตีฟขณะลอจิก "0" BUSAK เป็นสัญญาณตอบจากซีพียูได้แยกตัวเองออกจากแอดเดรสบัสและบัสข้อมูลเรียบร้อยแล้ว

- Ø (CLOCK) สัญญาณนาฬิกาที่จะป้อนเข้าระบบ

คำสั่งของ Z-80

ในการพัฒนา Z-80 มีจุดมุ่งหมายหลักเพื่อใช้ชุดคำสั่งเดิมของ 8080 ได้ ดังนั้นทุกคำสั่งที่เคยใช้ได้ในไมโครโปรเซสเซอร์เบอร์ 8080 จะยังคงใช้ได้ในกรณีที่ใช้งานกับซีพียู Z-80 Z-80 มีคำสั่งได้มากถึง 158 คำสั่ง ซึ่งรวมคำสั่งของ 8080 ทั้ง 78 คำสั่งไว้ด้วย กลุ่มคำสั่งของ Z-80 พอแบ่งแยกได้เป็นกลุ่มดังนี้

1. กลุ่มการโหลดและแลกเปลี่ยนข้อมูล (LOAD AND EXCHANGE)

2. กลุ่มการย้ายและค้นหาข้อมูลเป็นบล็อค(BLOCK TRANSFER AND SEARCH)

3. กลุ่มคำสั่งทางคณิตศาสตร์และลอจิก (ARITHEMATIC AND LOGIC)

4. กลุ่มคำสั่งการเคลื่อนย้ายข้อมูลเป็นวงจรรอบและการชิฟท์ (ROTATE AND SHIFT)

5. กลุ่มคำสั่งการกระทำในส่วนบิท (BIT MANIPULATION SET RESET TEST)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6. กลุ่มคำสั่งการกระโดด การเรียกโปรแกรมย่อย การคืนสู่กลับโปรแกรมหลัก

(JUMP,CALL AND RETURN)

7. กลุ่มคำสั่งเกี่ยวกับอินพุท-เอาท์พุท (INPUT-OUTPUT)

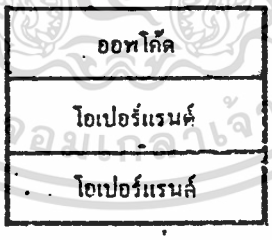
8. กลุ่มคำสั่งควบคุมซีพียู (BASIC CPU CONTROL)

การสร้างแอดเดรสของ Z-80

ภายในซีพียู Z-80 มีรจสเตอร์หลายตัวการทำงานของ Z-80 จึงคล่องตัวได้มาก การที่ให้ Z-80 เป็นระบบที่สมบูรณ์ต้องต่อกับหน่วยความจำและอุปกรณ์อินพุทเอาท์พุทในการต่อร่วมเพื่อให้ซีพียูทำงานได้สมบูรณ์ ในกรณีซีพียูจำเป็นจะต้องมีการเรียกแอดเดรสที่ต้องการอ้างแอดเดรส(ADDRESSING)ของซีพียูมีโหมด (MODE) การอ้างได้หลายวิธี ในแต่ละวิธีการอ้างแอดเดรสได้เป็นกลุ่ม ๆ คือ

1. การอ้างแอดเดรสแบบอิมมีเดียท (IMMEDIATE ADDRESSING) ในโหมดนี้คำสั่งจะประกอบด้วยออปโค้ด 1-2 ไบท์ และตามด้วยโอเปอร์แรนด์อีก 1 ไบท์ ลักษณะของคำสั่งจะเป็นดังนี้

2. การขยายตัวโอเปอร์แรนด์ในกลุ่มคำสั่งการอ้างแอดเดรสแบบอิมมีเดียท (IMMEDIATE EXTENDED) การอ้างแอดเดรสแบบนี้เหมือนกับแบบแรก แตกต่างกันเพียงส่วนของโอเปอร์แรนด์ เป็นข้อมูลขนาด 16 บิท ลักษณะรูปแบบของคำสั่งจะเป็นดังนี้



3. การอ้างแอดเดรสแบบ MODIFIED PAGE ZERO ในคำสั่ง Z-80 มีคำสั่งพิเศษอยู่ คำสั่งที่เหมือนคำสั่ง CALL คือคำสั่ง RES หรือ RESTART ลักษณะการทำงานในการอ้างแอดเดรสคือ มันสามารถกำหนดค่าให้กับโปรแกรมเคาว์เตอร์ได้โดยตรง ค่าที่มันกำหนดให้จะเป็นแอดเดรสที่อยู่ในเพจศูนย์ ลักษณะของคำสั่งนี้ประกอบขึ้นเพียงไบท์เท่านั้น จึงมีประโยชน์หลายอย่าง เช่นใช้เป็นมาร์ค (MARK) สำหรับคำสั่งให้กระทำในขณะที่มีการอินเตอร์รัพท์ ข้อมูลเดิมในโปรแกรมเคาว์เตอร์ก่อนการเปลี่ยนแปลงจะได้รับการเก็บรักษาไว้ได้อีกด้วย แต่เงื่อนไขในการกระโดดไปยังเพจศูนย์ยังมีขอบเขตจำกัด คือมันจะ

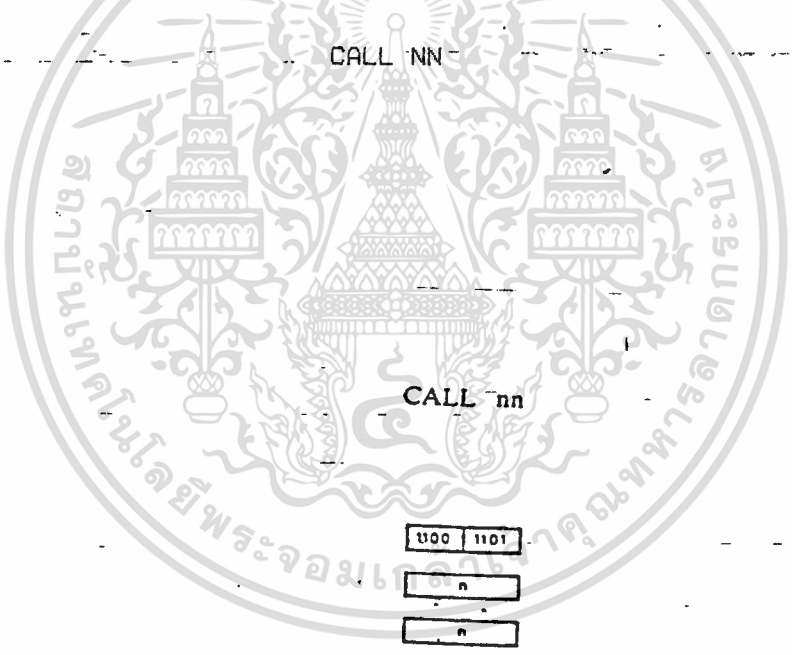
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระโดดไปได้เพียง 8 แอดเดรสเท่านั้น

4. การอ้างแอดเดรสแบบเปรียบเทียบ (RELATIVE ADDRESSING) วิธีการนี้จะใช้ข้อมูลไบ
ที่อยู่ตามหลังออปโค้ด เพื่อบอกตำแหน่งว่าแอดเดรสที่อ้างถึงอยู่ห่างจากค่าในโปรแกรมเคาว์เตอร์เท่า
เช่นเมื่อซีพียูกระทำคำสั่งนี้เสร็จ ค่าในโปรแกรมเคาว์เตอร์จะมีค่าเป็น PC+2+E นั่นคือ ค่าใหม่ที่มีน
อ้างถึงอยู่ห่างจากคำสั่งที่มันกระทำแล้วด้วยค่า E นั่นเอง ค่า E ที่ซีพียูมองเห็นจะเป็นลักษณะของตัว
2'S คอมพลีเมนท์ ดังนั้นค่าที่มีอ้างถึงได้จึงอยู่ระหว่าง + 127 กับ - 128

5. การอ้างแอดเดรสแบบ EXTENDED ADDRESSING วิธีนี้ข้อมูล 2 ไบท์ที่ตามออปโค้ดเป็น
ตัวกำหนดค่าแอดเดรส เช่นข้อมูลคือ NN ซึ่งเป็นตัวกำหนดค่าของแอดเดรสที่จะกระทำ ให้ยกตัวอย่างที่
คำสั่งคือ CALL NN หมายถึงการเรียกโปรแกรมย่อยที่ตำแหน่ง NN

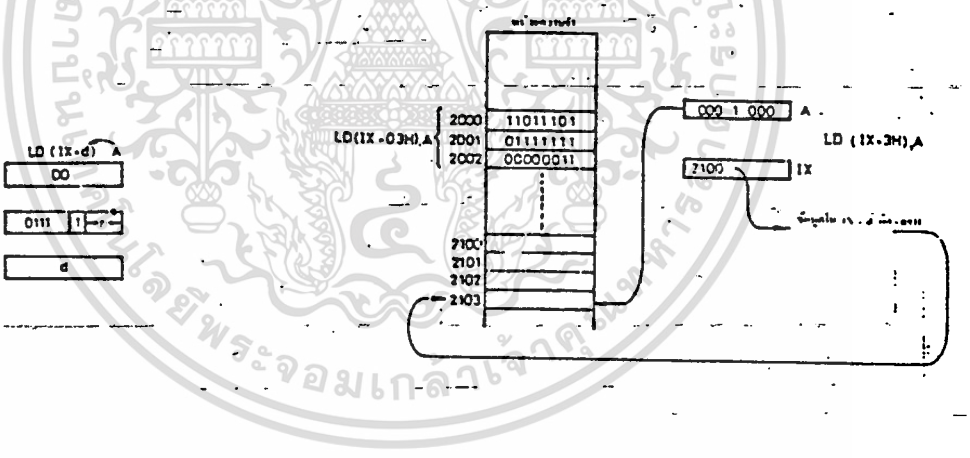
CALL NN



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งเป็นการนำเอาข้อมูลในตัวโอเปอร์แรนด์เป็นแอดเดรสนั่นเอง โดยการนำเอาข้อมูล ส่วน
โอเปอร์แรนด์ไปไว้ยังโปรแกรมเคาว์เตอร์โดยตรง ส่วนข้อมูลเดิมโปรแกรมเคาว์เตอร์จะเก็บไว้ในส่
ของสแตค

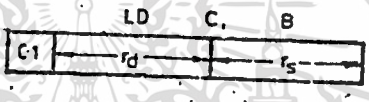
6. การอ้างแอดเดรสโดยใช้อินเด็กซ์รีจิสเตอร์ (INDEX ADDRESSING) ใน Z-80 จะมี IX
และ IY เป็นอินเด็กซ์รีจิสเตอร์ วิธีการใช้ร่วมในการอ้างแอดเดรสนั้นจะใช้ค่า IX หรือ IY เป็นฐา
เพื่อรวมกับค่าที่ตามหลังออปโค้ดมารวมกันเป็นแอดเดรสที่ต้องการ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะของคำสั่งนี้เป็นการนำเอาข้อมูลของรีจิสเตอร์ A ไปเก็บไว้ในหน่วยความจำ ที่อ้างแอดเดรสโ
ที่ค่า IX รวมกับค่า D โดยปรกติค่า D จะได้รับการรวมโดยอยู่ในรูป 2'S คอมพลีเมนท์ ดังนั้นจึงทำ
ค่า D แปรได้จากค่า (127 ถึง-128)

7. การอ้างแอดเดรสแบบใช้รีจิสเตอร์ (REGISTER ADDRESSING) การออกแบบกลุ่มคำสั่งจ
ขอบเขตของจำนวนบิทออปโค้ดที่จำกัด ต้องหาวิธีที่ให้ได้ประสิทธิภาพดีที่สุด ในกรณีของ Z-80 เป็นกา
กำหนดรหัสของรีจิสเตอร์ เช่น รีจิสเตอร์ 8 ตัว ก็ใช้รหัส 3 บิท ดังนั้นกลุ่มคำสั่งโหลดสามารถใช้คำ
เพียง 1 ไบท์ เพื่อการโหลดข้อมูลระหว่างรีจิสเตอร์ได้ เช่น

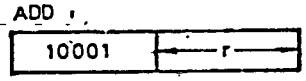


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่สามบิทใน RD คือชื่อรจสเตอร์ปลายทาง ส่วนสามบิทใน RS คือรจสเตอร์ต้นทาง

8. การอ้างแอดเดรสแบบ IMPLIED ADDRESSING ในกรณีนี้ชพยุจะต้ความหมายเองโดยตรง

รจสเตอร์ภายในชพยุตัวหนึ่งจะเป็นตัวโอเปอร์แรนด์ เช่น คำสั่ง



จากคำสั่งนจะเห็นว่าเราไม่ต้องกำหนดรหัสรจสเตอร์ A เลย แต่เครื่องจะรู้เอง ว่าเป็นคำสั่งที่ต้องก

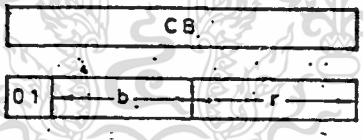
กระทำร่วมกับ รจสเตอร์ A กล่าวคือ การทำงานชพยุจะนำเอารจสเตอร์ที่กำหนดในออปโค้ดมากระท

ร่วมกับรจสเตอร์ A นั่นเอง

9. การอ้างแอดเดรสเข้าสู่บิทต่าง ๆ (BIT ADDRESSING) คำสั่งพิเศษใน Z-80 ที่สามาร

กระทำการเซทและรีเซท หรือการทดลองบิทใดบิทหนึ่งใน 8 บิท เช่น

RES b,r



ในที่นี้เราใช้รหัส B แทนโอเปอร์แรนด์ว่า เป็นบิทใดในรจสเตอร์ ถ้า B=000 หมายถึง

บิท 0 และ B=111 หมายถึงบิท 7 ดังนั้นเราสามารถทำให้บิทใดในรจสเตอร์ให้เป็น 0 ได้ เช่น RE

0,A หมายถึงทำให้บิท 0 ของรจสเตอร์ A เป็น 0

แฟลก

ใน Z-80 ชพยุ จะประกอบด้วยแฟลกจำนวน 6 แฟลก และมีบิทที่ไม่ได้แสดงเป็นแฟลกอี

2 บิท รวมเป็น 8 บิท เพื่อประกอบเป็นรจสเตอร์ F ส่วนของแฟลกแต่ละบิทสามารถที่จะเซทหรือรีเซ

ตามการกระทำของคำสั่งที่ชพยุกำลังทำงาน นอกจากนี้ชพยุยังสามารถใช้ตรวจสอบแฟลกเพื่อกระทำ

เงื่อนไขต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะการใช้แฟลกจะใช้ตัวอักษรย่อแทนแฟลก ดังนี้

C = แฟลกตัวทด

N = แฟลกแสดงการบวกหรือลบ

P/V = แฟลกแสดงพาริตและโอเวอร์โฟลว์

H = แฟลกแสดงตัวทดช่วย

Z = แฟลกแสดงค่าศูนย์

S = แฟลกเครื่องหมาย

X = ไม่ได้ใช้

ในการกระทำคำสั่งต่าง ๆ ของ Z-80 บางคำสั่งจะมผลต่อแฟลก บางคำสั่งก็ไม่มผลต่อแฟลก การทคำสั่งบางคำสั่งมผลต่อแฟลกทำให้เราสามารถตรวจสอบเงื่อนไขได้จากแฟลก เช่น

INC A (ให้เพิ่มค่ารีจิสเตอร์ A อีก 1)

ผลที่เกี่ยวกับแฟลกคือ

ถ้าผลลัพธ์เป็น 0 Z = 1

ถ้าผลลัพธ์เป็นลบหรือบิท D = 1 S = 1

ถ้ามีตัวทดในบิท 3 H = 1

ถ้าผลลัพธ์เท่ากับ 80 P/V = 1

คำสั่งตัวอย่างนี้ไม่เกี่ยวกับ N และ C แฟลก

ลักษณะของรีจิสเตอร์ F จะประกอบด้วยแฟลกแต่ละบิทเป็นดังนี้

S	Z	X	H	X	P/V	N	C
b_7	b_6	b_5	b_4	b_3	b_2	b_1	b_0

ในบิท b5 และ b3 จะเป็นบิทที่ประกอบเพิ่มขึ้นมาโดยมิได้มีความหมายในทางแฟลก แต่จะประกอบเพื่อให้รีจิสเตอร์ F ครบ 8 บิท การทำงานของซีพียูจึงสามารถโหลดข้อมูลจาก F ไปยัง A หรือโหลดจาก A กลับมายัง F ได้เป็นคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มคำสั่งเกี่ยวกับการโหลดข้อมูลและแลกเปลี่ยนข้อมูล

คำสั่งโหลดเป็นคำสั่งที่ต้องการให้ข้อมูลเคลื่อนย้ายจากสถานที่หนึ่ง (SOURCE) ไปยังอีกสถานที่หนึ่ง (DESTINATION) ซึ่งซอร์สและดิสเทนเนชั่นจะอยู่ทรจสเตอร์ภายในซีพียูหรือที่หน่วยความจำเอง ในตัวซีพียูจะมีรจสเตอร์หลายตัวจำเป็นที่จะต้องแสดงว่าที่แอดเดรสฟิต คำสั่งโหลดแบ่งออกได้เป็นสองชนิดคือ 1 ไบท์ กับ 2 ไบท์ ตัวอย่าง ถ้าต้องการโหลดข้อมูลของรจสเตอร์ H ไปที่รจสเตอร์ B ภาษาเอ็สเซ็มเบอร์เขียนได้ดังนี้

```
LD B,H
 |  |
 |  +-- ซอร์ส
 +----- ดิสเทนเนชั่น
```

ดิสเทนเนชั่นเขียนไว้หลัง LD ตอมาก็เขียนซอร์ส ออปโค้ดก็คือ 44H เมื่อซีพียูทำงานข้อมูลภายในรจสเตอร์ H ก็จะถูกส่งไปไว้ที่รจสเตอร์ B

คำสั่งอื่น ๆ นอกเหนือจากนี้ ก็มีการสลับเปลี่ยนข้อมูล

กลุ่มคำสั่งในการโหลด 8 บิท

Mnemonic	Operation	S	Z	-	H	-	P/V	N	C	76 543 210	(Hex)	จำนวนไบท์	จำนวน M Cycle	No. of T States	บิท Comments
LD r, r'	r ← r'	•	•	X	•	X	•	•	•	01 r r'		1	1	4	r, r' Reg.
LD r, n	r ← n	•	•	X	•	X	•	•	•	00 r 110 — n —		2	2	7	000 B 001 C
LD r, (HL)	r ← (HL)	•	•	X	•	X	•	•	•	01 r 110		1	2	7	010 D
LD r, (IX+d)	r ← (IX+d)	•	•	X	•	X	•	•	•	11 011 101 01 r 110 — d —	DD	3	5	19	011 E 100 H 101 L
LD r, (IY+d)	r ← (IY+d)	•	•	X	•	X	•	•	•	11 111 101 01 r 110 — d —	FD	3	5	19	111 A
LD (HL), r	(HL) ← r	•	•	X	•	X	•	•	•	01 110 r		1	2	7	
LD (IX+d), r	(IX+d) ← r	•	•	X	•	X	•	•	•	11 011 101 01 110 r — d —	DD	3	5	19	
LD (IY+d), r	(IY+d) ← r	•	•	X	•	X	•	•	•	11 111 101 01 110 r — d —	FD	3	5	19	
LD (HL), n	(HL) ← n	•	•	X	•	X	•	•	•	00 110 110 — n —	36	2	3	10	
LD (IX+d), n	(IX+d) ← n	•	•	X	•	X	•	•	•	11 011 101 00 110 110 — d — — n —	DD 36	4	5	19	
LD (IY+d), n	(IY+d) ← n	•	•	X	•	X	•	•	•	11 111 101 00 110 110 — d — — n —	FD 36	4	5	19	
LD A, (BC)	A ← (BC)	•	•	X	•	X	•	•	•	00 001 010	0A	1	2	7	
LD A, (DE)	A ← (DE)	•	•	X	•	X	•	•	•	00 011 010	1A	1	2	7	
LD A, (nn)	A ← (nn)	•	•	X	•	X	•	•	•	00 111 010 — n — — n —	3A	3	4	13	
LD (BC), A	(BC) ← A	•	•	X	•	X	•	•	•	00 000 010	02	1	2	7	
LD (DE), A	(DE) ← A	•	•	X	•	X	•	•	•	00 010 010	12	1	2	7	
LD (nn), A	(nn) ← A	•	•	X	•	X	•	•	•	00 110 010 — n — — n —	32	3	4	13	
LD A, I	A ← I	↕	↕	X	0	X	IFF	0	•	11 101 101 01 010 111	ED 57	2	2	9	
LD A, R	A ← R	↕	↕	X	0	X	IFF	0	•	11 101 101 01 011 111	ED 5F	2	2	9	
LD I, A	I ← A	•	•	X	•	X	•	•	•	11 101 101 01 000 111	ED 47	2	2	9	
LD R, A	R ← A	•	•	X	•	X	•	•	•	11 101 101 01 001 111	ED 4F	2	2	9	

r, r' หมายถึงรีจิสเตอร์ A, B, C, D, E, H, L

IFF หมายถึงข้อมูลในฟลิปฟล็อบแสดงสถานะการอินเตอร์รัพท์ได้รับการโหลดมาเก็บไว้ใน P/V แฟลก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มคำสั่งการโหลดข้อมูล 16 บิท

นีโมนิค	สัญญลักษณ์การกระทำ	แฟลก S	Z	-	H	-	P/V	N	C	ออพโค้ด 76 543 210	Hex	จำนวนไบท์	จำนวนไซเคิล	Mem T States	Comment
LD dd, nn	dd ← nn	•	•	X	•	X	•	•	•	00 dd0 001 — n — — n —		3	3	10	dd Pair 00 BC 01 DE 10 HL 11 SP
LD IX, nn	IX ← nn	•	•	X	•	X	•	•	•	11 011 101 00 100 001 — n — — n —	DD 21	4	4	14	
LD IY, nn	IY ← nn	•	•	X	•	X	•	•	•	11 111 101 00 100 001 — n — — n —	FD 21	4	4	14	
LD HL, (nn)	H ← (nn+1) L ← (nn)	•	•	X	•	X	•	•	•	00 101 010 — n — — n —	2A	3	5	16	
LD dd, (nn)	dd_H ← (nn+1) dd_L ← (nn)	•	•	X	•	X	•	•	•	11 101 101 01 dd1 011 — n — — n —	ED	4	6	20	
LD IX, (nn)	IX_H ← (nn+1) IX_L ← (nn)	•	•	X	•	X	•	•	•	11 011 101 00 101 010 — n — — n —	DD 2A	4	6	20	
LD IY, (nn)	IY_H ← (nn+1) IY_L ← (nn)	•	•	X	•	X	•	•	•	11 111 101 00 101 010 — n — — n —	FD 2A	4	6	20	
LD (nn), HL	(nn+1) ← H (nn) ← L	•	•	X	•	X	•	•	•	00 100 010 — n — — n —	22	3	5	16	
LD (nn), dd	(nn+1) ← dd_H (nn) ← dd_L	•	•	X	•	X	•	•	•	11 101 101 01 dd0 011 — n — — n —	ED	4	6	20	
LD (nn), IX	(nn+1) ← IX_H (nn) ← IX_L	•	•	X	•	X	•	•	•	11 011 101 00 100 010 — n — — n —	DD 22	4	6	20	
LD (nn), IY	(nn+1) ← IY_H (nn) ← IY_L	•	•	X	•	X	•	•	•	11 111 101 00 100 010 — n — — n —	FD 22	4	6	20	
LD SP, HL	SP ← HL	•	•	X	•	X	•	•	•	11 111 001	F9	1	1	6	
LD SP, IX	SP ← IX	•	•	X	•	X	•	•	•	11 011 101 11 111 001	DD F9	2	2	10	
LD SP, IY	SP ← IY	•	•	X	•	X	•	•	•	11 111 101 11 111 001	FD F9	2	2	10	
PUSH qq	(SP-2) ← qq_L (SP-1) ← qq_H	•	•	X	•	X	•	•	•	11 qq0 101		1	3	11	qq Pair 00 BC 01 DE 10 HL 11 AF
PUSH IX	(SP-2) ← IX_L (SP-1) ← IX_H	•	•	X	•	X	•	•	•	11 011 101 11 100 101	DD E5	2	4	15	
PUSH IY	(SP-2) ← IY_L (SP-1) ← IY_H	•	•	X	•	X	•	•	•	11 111 101 11 100 101	FD E5	2	4	15	
POP qq	qq_H ← (SP+1) qq_L ← (SP)	•	•	X	•	X	•	•	•	11 qq0 001		1	3	10	
POP IX	IX_H ← (SP+1) IX_L ← (SP)	•	•	X	•	X	•	•	•	11 011 101 11 100 001	DD E1	2	4	14	
POP IY	IY_H ← (SP+1) IY_L ← (SP)	•	•	X	•	X	•	•	•	11 111 101 11 [illegible] 001	FD E1	2	4	14	

ดังเกตุ

dd คือคู่รีจิสเตอร์ BC, DE, HL, SP

qq คือคู่รีจิสเตอร์ AF, BC, DE, HL

ตัวห้อย H หมายถึงข้อมูลไบท์ที่มีนัยสำคัญสูง

ตัวห้อย L หมายถึงข้อมูลไบท์ที่มีนัยสำคัญต่ำ

รูปที่ 2.12 แสดงกลุ่มคำสั่งในการโหลด 8 บิทและ 16 บิท

กลุ่มการเคลื่อนย้ายและค้นหาข้อมูลเป็นบล็อก

คำสั่งการส่งบล็อก หมายถึง คำสั่งที่ต้องการส่งข้อมูลภายในแอดเดรสหนึ่งไปยังอีก

แอดเดรสหนึ่งที่แสดงโดยค่าของจำนวนไบท์เท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง ถ้าต้องการส่งข้อมูลจากแอดเดรส 8280 จำนวน 128 ไบท์ไปยังแอดเดรสจากตำแ

8380 โปรแกรมที่ได้เป็นดังนี้

```
        LD      HL,8280     ชี้ซอแอดเดรส HL          8280
        LD      DE,8380     ดิวเทนนชันแอดเดรส DE     8380
        LD      BC,80       จำนวนไบท์ BC             80
        LDIR    LDIR        คำสั่งส่งบล็อก
        JP      MNTR        กระโดดไปที่มอนเตอร์ (MONITOR)
```

นิโมนิก	สัญญลักษณ์การกระทำ	แฟลก S	Z		H		P/V	N	C	ออพโค้ด 76 543 210	Hex	จำนวนไบท์	จำนวน M ไซเคิล	จำนวน T สเตท	Comments
EX DE, HL	DE←→HL	•	•		•	X	•	•	•	11 101 011	EB	1	1	4	
EX AF, AF'	AF←→AF'	•	•		•	X	•	•	•	00 001 000	08	1	1	4	
EXX	(BC←→BC' DE←→DE' HL←→HL')	•	•		•	X	•	•	•	11 011 001	D9	1	1	4	การสับเปลี่ยนค่าในรีจิสเตอร์
EX (SP), HL	H ←→(SP+1) L ←→(SP)	•	•	X	•	X	•	•	•	11 100 011	E3	1	5	19	
EX (SP), IX	IX$_H$←→(SP+1) IX$_L$←→(SP)	•	•	X	•	X	•	•	•	11 011 101 11 100 011	DD E3	2	6	23	
EX (SP), IY	IY$_H$←→(SP+1) IY$_L$←→(SP)	•	•	X	•	X	•	•	•	11 111 101 11 100 011	FD E3	2	6	23	
LDI	(DE)←(HL) DE ← DE+1 HL ← HL+1 BC ← BC-1	•	•	X	0	X	①	0	•	11 101 101 10 100 000	ED A0	2	4	16	โหลด (HL) ไปยัง (DE) เพิ่มที่เพอยท์เตอร์ และลดค่า (BC) ลง
LDIR	(DE)←(HL) DE ← DE+1 HL ← HL+1 BC ← BC-1 Repeat until BC = 0	•	•	X	0	X	0	0	•	11 101 101 10 110 000	ED B0	2 2	5 4	21 16	ถ้า BC ≠ 0 ถ้า BC = 0
LDD	(DE)←(HL) DE ← DE-1 HL ← HL-1 BC ← BC-1	•	•	X	0	X	①	0	•	11 101 101 10 101 000	ED A8	2	4	16	
LDDR	(DE)←(HL) DE ← DE-1 HL ← HL-1 BC ← BC-1 Repeat until BC = 0	•	•	X	0	X	0	0	•	11 101 101 10 111 000	ED B8	2 2	5 4	21 16	ถ้า BC ≠ 0 ถ้า BC = 0
CPI	A - (HL) HL ← HL+1 BC ← BC-1	①	②	X	①	X	①	1	•	11 101 101 10 100 001	ED A1	2	4	16	
CPIR	A - (HL) HL ← HL+1 BC ← BC-1 Repeat until A = (HL) or BC = 0	①	②	X	①	X	①	1	•	11 101 101 10 110 001	ED B1	2 2	5 4	21 16	ถ้า BC ≠ 0 และ A ≠ (HL) ถ้า BC = 0 หรือ A = (HL)
CPD	A - (HL) HL ← HL-1 BC ← BC-1	①	②	X	①	X	①	1	•	11 101 101 10 101 001	ED A9	2	4	16	
CPDR	A - (HL) HL ← HL-1 BC ← BC-1 Repeat until A = (HL) or BC = 0	①	②	X	①	X	①	1	•	11 101 101 10 111 001	ED B9	2 2	5 4	21 16	ถ้า BC ≠ 0 และ A ≠ (HL) ถ้า BC = 0 หรือ A = (HL)

ดังเกต 1) P/V แฟลกเป็น 0 ถ้าผลลัพธ์ของ BC - 1 = 0 นอกนั้น

2) Z แฟลกเป็น 1 ถ้า A = (HL) นอกนั้น Z = 0

รูปที่ 2-13 แสดงกลุ่มคำสั่งในการเคลื่อนย้ายและค้นหาข้อมูลเป็นกลุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มคำสั่งการกระทำทางคณิตศาสตร์ของตัวเลข 16 บิท

นีโมนิค	สัญญลักษณ์การกระทำ	แฟลก S	Z		H		P/V	N	C	ออพโค้ด 76 543 210	Hex	จำนวนไบท์	จำนวนแบ่งไซเคิล	จำนวนสเตท	หมายเหตุ
ADD HL,ss	HL ← HL+ss	•	•	X	X	X	•	0	↕	00 ss1 001		1	3	11	ss Reg. 00 BC 01 DE 10 HL 11 SP
ADC HL,ss	HL ← HL+ss+CY	↕	↕	X	X	X	V	0	↕	11 101 101 01 ss1 010	ED	2	4		
SBC HL,ss	HL ← HL-ss-CY	↕	↕	X	X	X	V	1	↕	11 101 101 01 ss0 010	ED	2	4	15	
ADD IX,pp	IX ← IX+pp	•	•	X	X	X	•	0	↕	11 011 101 00 pp1 001	DD	2	4	15	pp Reg. 00 BC 01 DE 10 IX 11 SP
ADD IY,rr	IY ← IY+rr	•	•	X	X	X	•	0	↕	11 111 101 00 rr1 001	FD	2	4	15	rr Reg. 00 BC 01 DE 10 IY 11 SP
INC m	m ← m+1	•	•	X	•	X	•	•	•	00 m0 011		1	1	6	
INC IX	IX ← IX+1	•	•	X	•	X	•	•	•	11 011 101 00 100 011	DD 23	2	2	10	
INC IY	IY ← IY+1	•	•	X	•	X	•	•	•	11 111 101 00 100 011	FD 23	2	2	10	
DEC m	m ← m-1	•	•	X	•	X	•	•	•	00 m1 011		1	1	6	
DEC IX	IX ← IX-1	•	•	X	•	X	•	•	•	11 011 101 00 101 011	DD 2B	2	2	10	
DEC IY	IY ← IY-1	•	•	X	•	X	•	•	•	11 111 101 00 101 011	FD 2B	2	2	10	

สังเกตุ SS เป็นคู่รจสเตอร์ BC,DE,HL,SP

PP เป็นคู่รจสเตอร์ BC,DE,IX,SKP

RR เป็นคู่รจสเตอร์ BC,D,IY,SP

กลุ่มคำสั่งทางคณิตศาสตร์และลอจิก

คำสั่งคำนวณมี 2 ชนิด คือ คำสั่งคำนวณเลขลอจิกและคำสั่งคำนวณเลขธรรมดา ในกรณี

คำสั่งการคำนวณ 8 บิท ดสเทนนชนไดถูกกำหนดที่เอกคมูเลเตอร์ (A) หรือในกรณีการคำนวณเลข

16 บิทใช้รจสเตอร์ HL,IX,IY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มคำสั่งการกระทำทางคณิตศาสตร์และลอจิก

นีโมนิค	สัญญลักษณ์การกระทำ	แฟลก S	Z		H		P/V	N	C	ออพโค้ด		จำนวนไบท์	จำนวน M ไซเคิล	จำนวน สเตท	หมายเหตุ
ADD A, r	A ← A+r	↕	↕	X	↕	X	V	0	↕	10 000 r		1	1	4	r Reg.
ADD A, n	A ← A+n	↕	↕	X	↕	X	V	0	↕	11 000 110 – n –		2	2	7	000 B 001 C 010 D
ADD A, (HL)	A ← A+(HL)	↕	↕	X	↕	X	V	0	↕	10 000 110		1	2	7	011 E
ADD A, (IX+d)	A ← A+(IX+d)	↕	↕	X	↕	X	V	0	↕	11 011 101 10 000 110 – d –	DD	3	5	19	100 H 101 L 111 A
ADD A, (IY+d)	A ← A+(IY+d)	↕	↕	X	↕	X	V	0	↕	11 111 101 10 000 110 – d –	FD	3	5	19	
ADC A, s	A ← A+s+CY	↕	↕	X	↕	X	V	0	↕	001					S คือรีจิสเตอร์ใด ๆ r, n,
SUB s	A ← A-s	↕	↕	X	↕	X	V	1	↕	010					(HL), (IX+d),
SBC A, s	A ← A-s-CY	↕	↕	X	↕	X	V	1	↕	011					(IY+d) ดังที่อยู่ในคำสั่ง
AND s	A ← A ∧ s	↕	↕	X	1	X	P	0	0	100					ADD ถ้าเป็นคำสั่งอื่น
OR s	A ← A ∨ s	↕	↕	X	0	X	P	0	0	110					ให้เปลี่ยน 000 ในคำสั่ง
XOR s	A ← A ⊕ s	↕	↕	X	0	X	P	0	0	101					ADD ให้เป็นไปตามคำสั่งนั้น
CP s	A - s	↕	↕	Y	↕	X	V	1	↕	111					
INC r	r ← r+1	↕	↕	X	↕	X	V	0	•	00 r 100		1	1	4	
INC (HL)	(HL) ← (HL)+1	↕	↕	X	↕	X	V	0	•	00 110 100		1	3	11	
INC (IX+d)	(IX+d) ← (IX+d)+1	↕	↕	X	↕	X	V	0	•	11 011 101 00 110 100 – d –	DD	3	6	23	
INC (IY+d)	(IY+d) ← (IY+d)+1	↕	↕	X	↕	X	V	0	•	11 111 101 00 110 100 – d –	FD	3	6	23	
DEC s	s ← s-1	↕	↕	X	↕	X	V	1	•	101					S คือรีจิสเตอร์ใด ๆ r, (HL), (IX+d), (IY+d) ดังที่อยู่ในคำสั่ง INC ถ้าเป็นการสั่งงาน DEC ก็เพียงแต่เปลี่ยน 100 ด้วย 101

สังเกตว่า แฟลก P/V ถ้า P หมายถึง การแสดงพาริตี้ และ V หมายถึงการแสดงการโอเวอร์โฟลว

0 แฟลกไม่เปลี่ยน 0 = แฟลกรีเซท 1 = แฟลกเซท

X เป็นอะไรก็ได้ = แฟลกจะเป็นไปตามเงื่อนไข

กลุ่มคำสั่งการกระทำทางคณิตศาสตร์โดยเฉพาะบางอย่างและการควบคุมซีพียู

นีโมนิค	สัญญลักษณ์การกระทำ	แฟลก S	Z		H		P/V	N	C	ออพโค้ด		จำนวนไบท์	จำนวน M ไซเคิล	จำนวน สเตท	หมายเหตุ
DAA	เปลี่ยนข้อมูลในแอคคิวมูเลเตอร์ที่เป็นตัวเลข BCD โดยใช้มาพลังคำสั่ง ADD หรือ SUB เพื่อบวกหรือลบเลข BCD	↕	↕	X	↕	X	P	•	↕	00 100 111	27	1	1	4	การปรับข้อมูลในแอคคิวมูเลเตอร์เป็น BCD
CPL	A ← $\bar{A}$	•	•	X	1	X	•	1	•	00 101 111	2F	1	1	4	ทำการคอมพลีเมนท์แอคคิวมูเลเตอร์
NEG	A ← $\bar{A}$+1	↕	↕	X	↕	X	V	1	↕	11 101 101 01 000 100	ED 44	2	2	8	ทำแอคคิวมูเลเตอร์ให้เป็นเลขลบหรือ 2's คอมพลีเมนท์
CCF	CY ← $\overline{CY}$	•	•	X	X	X	•	0	↕	00 111 111	3F	1	1	4	
SCF	CY ← 1	•	•	X	0	X	•	0	1	00 110 111	37	1	1	4	
NOP	No operation	•	•	X	•	X	•	•	•	00 000 000	00	1	1	4	
HALT	CPU halted	•	•	X	•	X	•	•	•	01 110 110	76	1	1	4	
DI*	IFF ← 0	•	•	X	•	X	•	•	•	1x 110 011	F3	1	1	4	
EI*	IFF ← 1	•	•	X	•	X	•	•	•	11 111 011	FB	1	1	4	
IM 0	Set interrupt mode 0	•	•	X	•	X	•	•	•	11 101 101 01 000 110	ED 46	2	2	8	
IM 1	Set interrupt mode 1	•	•	X	•	X	•	•	•	11 101 101 01 010 110	ED 56	2	2	8	
IM 2	Set interrupt mode 2	•	•	X	•	X	•	•	•	11 101 101 01 011 110	ED 5E	2	2	8	

สังเกต IFF แสดงผลของอินเตอร์รัพท์ฟลิบฟลอบ

CY แสดงลักษณะของฟลิบฟลอบตัวทด

รูปที่ 2.14 แสดงกลุ่มคำสั่งทางคณิตศาสตร์และลอจิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มคำสั่งการเคลื่อนย้ายข้อมูลเป็นวงรอบและการชิฟท์

คำสั่งที่ต้องการชิฟท์ข้อมูลภายในรีจิสเตอร์ หรือข้อมูลที่ชี้แสดงภายในหน่วยความจำจะให้คำสั่งอ

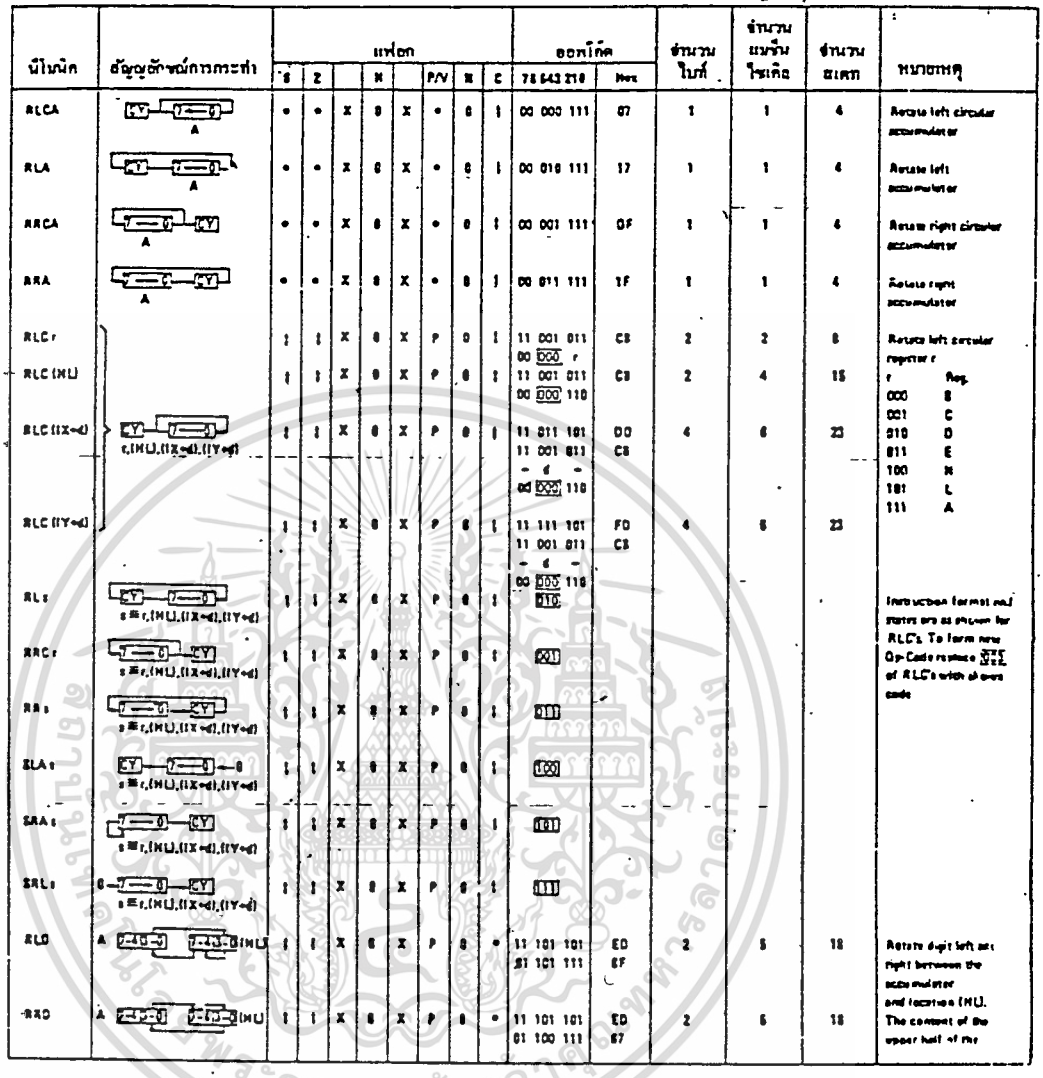
หลักการจะใช้การควบคุมในระบบชิฟแบบอนุกรม ซึ่งจะใช้เวลาเข้ามาเกี่ยวข้องด้วย

ตัวอย่าง RLC (ROTATE LEFT CIRCUIT)

ข้อมูลที่ชี้แสดงโดยซ็อซจะถูกชิฟท์ไปทางซ้าย 1 บิท ซึ่งหลังจากคำสั่งนบิทตำแหน่งสูงสุด (MSB) จะถูกย้ายไปที่ตำแหน่งของบิทต่ำสุด (LSB) และบิทสูงสุด (MSB) ก็จะไปเข้าที่แคร์แฟลก รูปแสดงการทำงานเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มคำสั่งในการเลื่อนบิทและหมุนบิท



นีโมนิก	สัญญลักษณ์การกระทำ	แฟลก S	Z		H		P/V	N	C	ออปโค้ด 76 543 210	Hex	จำนวนไบท์	จำนวนแมชีนไซเคิล	จำนวนสเตท	หมายเหตุ
RLCA		•	•	X	0	X	•	0	↕	00 000 111	07	1	1	4	Rotate left circular accumulator
RLA		•	•	X	0	X	•	0	↕	00 010 111	17	1	1	4	Rotate left accumulator
RRCA		•	•	X	0	X	•	0	↕	00 001 111	0F	1	1	4	Rotate right circular accumulator
RRA		•	•	X	0	X	•	0	↕	00 011 111	1F	1	1	4	Rotate right accumulator
RLC r		↕	↕	X	0	X	P	0	↕	11 001 011 00 000 r	CB	2	2	8	Rotate left circular register r
RLC (HL)		↕	↕	X	0	X	P	0	↕	11 001 011 00 000 110	CB	2	4	15	r Reg. 000 B 001 C 010 D 011 E 100 H 101 L 111 A
RLC (IX+d)	c,(IX+d)	↕	↕	X	0	X	P	0	↕	11 011 101 11 001 011 d 00 000 110	DD CB	4	6	23	
RLC (IY+d)		↕	↕	X	0	X	P	0	↕	11 111 101 11 001 011 d 00 000 110	FD CB	4	6	23	
RL s	s=r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	010					Instruction format and states are as shown for RLC's. To form new Op-Code replace 000 of RLC's with shown code
RRC s	s=r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	001					
RR s	s=r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	011					
SLA s	s=r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	100					
SRA s	s=r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	101					
SRL s	s=r,(HL),(IX+d),(IY+d)	↕	↕	X	0	X	P	0	↕	111					
RLD	(HL)	↕	↕	X	0	X	P	0	•	11 101 101 01 101 111	ED 6F	2	5	18	Rotate digit left and right between the accumulator and location (HL). The content of the upper half of the
RRD	(HL)	↕	↕	X	0	X	P	0	•	11 101 101 01 100 111	ED 67	2	5	18	

รูปที่ 2.15 แสดงกลุ่มคำสั่งการเคลื่อนย้ายข้อมูลเป็นวงรอบและการชิฟท์

กลุ่มคำสั่งการกระทำในส่วนบิท

ในการเขียนโปรแกรมบางที่ต้องการที่จะกำหนดให้ตำแหน่งของบิททำงาน ตัวอย่างเช่นใ
กรณีที่ต้องการให้ข้อมูลจาก I/O อ่านเข้าและภายในข้อมูลมี 1 บท เท่านนำมาตรวจสอบหรือว่าต้อง
ที่จะเซทบิทใดบิทหนึ่ง คำสั่งของการกระทำงานในสถานะนี้จะใช้คำว่า BIT

ตัวอย่าง ถ้าต้องการตรวจสอบข้อมูลที่แสดงทรจสเตอร์ HL บิทที่ 4 ว่ามีค่าเป็น 0 ห

1 คำสั่งเป็นดังนี้

```
BIT 4, (HL)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากการทำงานของคำสั่งน ถ้าค่าของข้อมูลบทที่ 4 มีค่า 0.Z แฟลกจะถูกเซท ถ้าม

ค่า 1.Z แฟลกจะถูกรีเซท ข้อมูลที่แสดงที่แอดเดรส (HL) จะไม่เปลี่ยนแปลง

กลุ่มคำสั่งในการเซท รีเซทและทดสอบบิท

มีโนมิค	สัญญลักษณ์	แฟลก S	Z		H		P/V	N	C	ออฟโค้ด 76 543 210	Hex	จำนวนไบท์	จำนวนแมชชีนไซเคิล	จำนวน T อัตรา
BIT b, r	$Z \leftarrow \overline{r_b}$	X	‡	X	1	X	X	0	•	11 001 011 01 b r	CB	2	2	8
BIT b, (HL)	$Z \leftarrow \overline{(HL)_b}$	X	‡	X	1	X	X	0	•	11 001 011 01 b 110	CB	2	3	12
BIT b, (IX+d)	$Z \leftarrow \overline{(IX+d)_b}$	X	‡	X	1	X	X	0	•	11 011 101 11 001 011 – d – 01 b 110	DD CB	4	5	20
BIT b, (IY+d)	$Z \leftarrow \overline{(IY+d)_b}$	X	‡	X	1	X	X	0	•	11 111 101 11 001 011 – d – 01 b 110	FD CB	4	5	20
SET b, r	$r_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 001 011 11 b r	CB	2	2	8
SET b, (HL)	$(HL)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 001 011 11 b 110	CB	2	4	15
SET b, (IX+d)	$(IX+d)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 011 101 11 001 011 – d – 11 b 110	DD CB	4	6	23
SET b, (IY+d)	$(IY+d)_b \leftarrow 1$	•	•	X	•	X	•	•	•	11 111 101 11 001 011 – d – 11 b 110	FD CB	4	6	23
RES b, s	$s_b \leftarrow 0$ s=r, (HL), (IX+d), (IY+d)	•	•	X	•	X	•	•	•	10				

r	Reg.
000	B
001	C
010	D
011	E
100	H
101	L
111	A

b	Bit Tested
000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

ในการสร้างออฟโค้ดตัวใหม่ ให้เปลี่ยน 11 ในกลุ่มคำสั่ง SET ด้วยค่า 10 แฟลกและช่วงเวลาการทำงานเหมือนกับคำสั่งเซท

สังเกต สัญญลักษณ์ SB แทนบิท 6 (0 ถึง 7) ของตำแหน่งรจสเตอร์ S หรือแอดเดรส S

รูปที่ 2.16 แสดงกลุ่มคำสั่งการกระทำในส่วนบิท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มคำสั่งการกระโดด

คำสั่ง JUMP เป็นคำสั่งที่ทำการเปลี่ยนแปลงลำดับการทำงานของโปรแกรม โปรแกรมตามธรรมดาหลังจากเมื่อถูกรีเซทแล้ว การทำงานจะเริ่มจากแอดเดรส 0 แต่ถ้าในกรณีที่ รับการควบคุมอย่างใดอย่างหนึ่งตาม ค่าผลลัพธ์ที่ได้จะให้การทำงานแตกต่างกันออกไป ในกรณีนี้ การทำงานของโปรแกรมจะต้องเปลี่ยนไปด้วย ซึ่งจำเป็นจะต้องเปลี่ยนไปด้วย ซึ่งจำเป็นจะต้องใช้ คำสั่ง JUMP

ตัวอย่าง JP 8100H

เมื่อคำสั่งนี้ทำงาน การทำงานจะกระโดดไปที่แอดเดรส 8100 อย่างไม่มีเงื่อนไข

กลุ่มคำสั่งการกระโดด

นีโมนิค	สัญญลักษณ์การกระทำ	แฟล็ก S	Z		H		P/V	N	C	ออพโค้ด 76 543 210	Hex	จำนวนไบท์	จำนวนแมชีนไซเคิล	จำนวน T สเตท	หมายเหตุ
JP aa	PC ← aa	•	•	X	•	X	•	•	•	11 000 011 – a – – a –	C3	3	3	10	
JP cc, aa	If condition cc is true PC ← aa, otherwise continue	•	•	X	•	X	•	•	•	11 cc 010 – a – – a –		3	3	10	cc เงื่อนไข 000 NZ ไม่เซ็ตศูนย์ (Z แฟล็ก = 0) 001 Z (Z = 1) 010 NC (C = 0) 011 C (C = 1) 100 PO (P = 0) 101 PE (P = 1) 110 P (B = 0) 111 M (S = 1)
JR e	PC ← PC + e	•	•	X	•	X	•	•	•	00 011 000 – e-2 –	18	2	3	12	
JR C, e	If C = 0, continue If C = 1, PC ← PC+e	•	•	X	•	X	•	•	•	00 111 000 – e-2 –	38	2 2	2 3	7 12	ถ้าไม่เป็นไปตามเงื่อนไข
JR NC, e	If C = 1, continue If C = 0, PC ← PC+e	•	•	X	•	X	•	•	•	00 110 000 – e-2 –	30	2 2	2 3	7 12	ถ้าไม่เป็นไปตามเงื่อนไข
JR Z, e	If Z = 0 continue If Z = 1, PC ← PC+e	•	•	X	•	X	•	•	•	00 101 000 – e-2 –	28	2 2	2 3	7 12	ถ้าไม่เป็นไปตามเงื่อนไข
JR NZ, e	If Z = 1, continue If Z = 0, PC ← PC+e	•	•	X	•	X	•	•	•	00 100 000 – e-2 –	20	2 2	2 3	7 12	ถ้าไม่เป็นไปตามเงื่อนไข
JP (HL)	PC ← HL	•	•	X	•	X	•	•	•	11 101 001	E9	1	1	4	
JP (IX)	PC ← IX	•	•	X	•	X	•	•	•	11 011 101 11 101 001	DD E9	2	2	8	
JP (IY)	PC ← IY	•	•	X	•	X	•	•	•	11 111 101 11 101 001	FD E9	2	2	8	
DJNZ, e	B ← B-1 If B = 0, continue If B ≠ 0, PC ← PC+e	•	•	X	•	X	•	•	•	00 010 000 – e-2 –	10	2 2	2 3	8 13	ถ้า B ≠ 0 ถ้า B ≠ 0

สังเกต e แทนข้อมูลที่จะใช้อ้างอิง

e เป็นเลข 2'S คอมพลีเมนท์ มีค่าจาก -126, 129

e-2 เป็นออปโค้ดเพื่อให้ค่าแอดเดรสของ PC + e เป็นค่า PC ใหม่

รูปที่ 2.17 แสดงกลุ่มคำสั่งการกระโดด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นิโมนิก	สัญญลักษณ์การกระทำ	แฟลก S	Z		H		P/V	N	C	ออพโค้ด 76 543 210	Hex	จำนวนไบท์	จำนวนแมชชีนไซเกิล	จำนวน T สเตท	หมายเหตุ
CALL nn	(SP-1) ← PC_H (SP-2) ← PC_L PC ← nn	•	•	X	•	X	•	•	•	11 001 101 – n – – n –	CD	3	5	17	
CALL cc, nn	ถ้าเงื่อนไข CC ไม่เป็นจริง โปรแกรมจะกระทำต่อไป แต่ถ้าเป็นจริง ทำสั่งนี้จะเหมือน CALL nn	•	•	X	•	X	•	•	•	11 cc 100 – n – – n –		3 3	3 5	10 17	ถ้า CC ไม่เป็นจริง
RET	PC_L ← (SP) PC_H ← (SP+1)	•	•	X	•	X	•	•	•	11 001 001	C9	1	3	10	
RET cc	ถ้าเงื่อนไข CC ไม่เป็นจริง โปรแกรมจะกระทำต่อไป แต่ถ้าเป็นจริง ทำสั่งนี้จะเหมือนกับ RET	•	•	X	•	X	•	•	•	11 cc 000		1 1	1 3	5 11	ถ้า CC ไม่เป็นจริง
RETI	เป็นการกลับเข้าสู่โปรแกรมหลักหลังจากถูกอินเตอร์รัพท์	•	•	X	•	X	•	•	•	11 101 101 01 001 101	ED 4D	2	4	14	
RETN[1]	เป็นการกลับเข้าสู่โปรแกรมหลักหลังจากได้รับการอินเตอร์รัพท์แบบนอนมาสเคเบิ้ล	•	•	X	•	X	•	•	•	11 101 101 01 000 101	ED 45	2	4	14	
RST p	(SP-1) ← PC_H (SP-2) ← PC_L PC_H ← 0 PC_L ← p	•	•	X	•	X	•	•	•	11 t 111		1	3	11	

CC	เงื่อนไข
000	NZ (Z = 0)
001	Z (Z = 1)
010	NC (C = 1)
011	C (C = 1)
100	PO (P = 0)
101	PE (P = 1)
110	P (S = 0)
111	M (S = 1)

t	p
000	00H
001	08H
010	10H
011	18H
100	20H
101	28H
110	30H
111	38H

[1] RETN จะกระทำการโหลด IFF_2 IFF_1

รูปที่ 2.18 (ต่อ) แสดงกลุ่มคำสั่งการถูกกระโดด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กลุ่มคำสั่งเกี่ยวกับอินพุท เอาท์พุท

จากที่ได้กล่าวมาแล้ว แอดเดรสที่ใช้อยู่ในช่วงแอดเดรสจาก 0 ถึง 64K (65532) สำหรับ Z-80 ซีพียูจะมีแอดเดรสพิเศษสำหรับ I/O (อินพุท/เอาท์พุท) ซึ่งจะให้ข้อมูลติดต่อกับภายนอ
ได้ ช่วงของ I/O แอดเดรสอยู่ในตำแหน่งจาก 00 ถึง FFH

คำสั่ง IN คำสั่งนี้ใช้สำหรับข้อมูลจากภายนอกถูกให้เขียนเข้าไปที่ซีพียู

ตัวอย่าง IN A, C

ข้อมูลของ I/O แอดเดรสที่แสดงที่รีจิสเตอร์ C ถูกเขียนเข้าไปที่แอคคูมูเลเตอร์

นีโมนิก	สัญญลักษณ์การกระทำ	แฟลก S	Z		H		P/V	N	C	ออพโค้ด 76 543 210	Hex	จำนวนไบท์	จำนวนแมชีนไซเกิล	จำนวน T อเดท	หมายเหตุ
IN A, (n)	A ← (n)	•	•	X	•	X	•	•	•	11 011 011 n	DB	2	3	11	n to $A_0 \sim A_7$ Acc to $A_8 \sim A_{15}$
IN r, (C)	r ← (C) ถ้า r = 110 เท่านั้น แฟลก จึงจะมีผล	‡	‡	X	‡	X	P	0	•	11 101 101 01 r 000	ED	2	3	12	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INI	(HL) ← (C) B ← B-1 HL ← HL+1	X	① ‡	X	X	X	X	1	•	11 101 101 10 100 010	ED A2	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INIR	(HL) ← (C) B ← B-1 HL ← HL+1 Repeat until B = 0	X	1	X	X	X	X	1	•	11 101 101 10 110 010	ED B2	2 (If B ≠ 0) 2 (If B = 0)	5 4	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
IND	(HL) ← (C) B ← B-1 HL ← HL-1	X	① ‡	X	X	X	X	1	•	11 101 101 10 101 010	ED AA	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
INDR	(HL) ← (C) B ← B-1 HL ← HL-1 ทำซ้ำจนกระทั่ง B = 0	X	1	X	X	X	X	1	•	11 101 101 10 111 010	ED BA	2 (If B ≠ 0) 2 (If B = 0)	5 4	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUT (n), A	(n) ← A	•	•	X	•	X	•	•	•	11 010 011 n	D3	2	3	11	n to $A_0 \sim A_7$ Acc to $A_8 \sim A_{15}$
OUT (C), r	(C) ← r	•	•	X	•	X	•	•	•	11 101 101 01 r 001	ED	2	3	12	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUTI	(C) ← (HL) B ← B-1 HL ← HL+1	X	① ‡	X	X	X	X	1	•	11 101 101 10 100 011	ED A3	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OTIR	(C) ← (HL) B ← B-1 HL ← HL+1 ทำซ้ำจนกระทั่ง B = 0	X	1	X	X	X	X	1	•	11 101 101 10 110 011	ED B3	2 (If B ≠ 0) 2 (If B = 0)	5 4	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OUTD	(C) ← (HL) B ← B-1 HL ← HL-1	X	① ‡	X	X	X	X	1	•	11 101 101 10 101 011	ED AB	2	4	16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$
OTDR	(C) ← (HL) B ← B-1 HL ← HL-1 ทำซ้ำจนกระทั่ง B = 0	X	1	X	X	X	X	1	•	11 101 101 10 111 011	ED BB	2 (If B ≠ 0) 2 (If B = 0)	5 4	21 16	C to $A_0 \sim A_7$ B to $A_8 \sim A_{15}$

สังเกต 1. ถ้าผลลัพธ์ B-1 เป็นศูนย์

Z แฟลก จะได้รับการเซท นอกนั้นจะได้รับการรีเซท

รูปที่ 2-19 แสดงกลุ่มคำสั่งเกี่ยวกับอินพุท-เอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง OUT

คำสั่งนี้จะกลับกันกับคำสั่ง IN ข้อมูลของรีจิสเตอร์ที่แสดงด้วยค่าโอเปอร์แรนด์จะให้ข้อมูลจาก

แอคคิวมูเลเตอร์อ่านออกไปที่ I/O แอดเดรสที่แสดงด้วยค่าโอเปอร์แรนด์

ตัวอย่าง OUT (C), D

ข้อมูลของ D รีจิสเตอร์จะถูกให้อ่านไปที่ I/O แอดเดรสที่แสดงที่รีจิสเตอร์ C -

กลุ่มคำสั่งควบคุมซีพียู

การควบคุมการทำงานมีหลายคำสั่ง เช่น

NOP (NO OPERATION) เป็นคำสั่งที่ไม่ให้ซีพียูทำงานแต่โปรแกรมเคาน์เตอร์ถูกเพิ่มด้วยค่า

ข้อมูลภายในรีจิสเตอร์ และแฟลกต่าง ๆ จะไม่เปลี่ยนแปลง คำสั่งนี้จะใช้ในการปรับแต่งช่วงเวลาใ

ในกรณีที่ใช้ทำโปรแกรมของไทม์เมอร์

HALT เป็นคำสั่งให้ซีพียูหยุดทำงาน เพื่อให้สถานะการทำงานของคำสั่งนี้หยุดลงในขณะที่การ

อินเตอร์รัพรับภายนอกเข้าหรือทำการรีเซท

นอกจากนี้คำสั่งเกี่ยวกับการอินเตอร์รัพ เช่น EI, DI, IM0, IM1, IM2

MEMORY AND MEMORY DECODER

หน่วยความจำหลักในระบบไมโครคอมพิวเตอร์ เป็นความจำระบบอิเล็คทรอนิคส์ สามา

ที่จะจำข้อมูลในรูปของตัวเลขซึ่งอยู่ในรูปของเลขไบนารี (เลขฐาน) คือเลข "0" กับ "1" ตัวเล

ไบนารี 1 หลักเรียกว่า 1 บิท (BIT) เลขไบนารีหลายบิทที่คอมพิวเตอร์ถือว่าเป็นคำสั่งหรือข้อมูลที่

คอมพิวเตอร์อ่าน (READ) เข้ามาหรือเขียน (WRITE) ออกไปเรียกว่าคำ (WORD) จำนวนเลข

ไบนารีที่ประกอบเป็นรหัส 1 คำ ถือว่าเป็นความยาวของคำ (WORD LENGHT) ซึ่งมีคำเรียกตามคว

หมายของคำนั้น เช่น ถ้าเป็นความยาว 4 บิทเรียกว่า NIBBLE คำที่มีความยาว 8 บิทเรียกว่า 1

(BYTE) ซึ่งขนาดของคำในไมโครคอมพิวเตอร์ต่าง ๆ ไปใช้กันอยู่ หน่วยความจำจะสามารถเก็บคำสั่ง

ข้อมูลเหล่านี้ไว้ในตำแหน่งใดตำแหน่งหนึ่ง ในหน่วยความจำ แต่ละตำแหน่งที่ซีพียูสามารถที่จะอ้างถึงได

เรียกว่า WORD POSITION หรือแอดเดรส (ADDRESS)

การนำข้อมูลจากภายนอกเข้าไปเก็บในหน่วยความจำที่แอดเดรสใด ๆ เรียกการเขียน

(MEMORY WRITE OPERATION) ถ้านำข้อมูลออกจากหน่วยความจำเรียกว่าการอ่าน (MEMORY RE

OPERATION).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการค้นหาข้อมูลหรือเข้าไปหาข้อมูลในหน่วยความจำที่แอดเซส (ACCESS) ซึ่งแบ่งต

ลักษณะได้ 2 แบบใหญ่ ๆ คือแบบแรกเรียกว่า แบบแรนดอม (RANDOM ACCESS) ซึ่งสามารถที่จะเ

ไปถึง ณ แอดเดรสใด ๆ ได้โดยตรง จึงทำให้ใช้เวลาเท่ากันในการอ้างแอดเดรสและแบบที่สองคือ

ซีแควนเชียล (SEQUENTIAL ACCESS) เราใช้เวลาเข้าถึงข้อมูลที่แอดเดรสใด ๆ ไม่เท่ากัน จะ

ขึ้นอยู่กับตำแหน่งของข้อมูลนั้น ๆ เช่นที่แอดเดรส 100 จะใช้เวลาน้อยกว่าที่แอดเดรส 150 เพราะ

การเข้าถึงข้อมูลจะทำได้โดยลำดับก่อนหลัง ไม่สามารถอ้างถึงได้โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชนิดของหน่วยความจำ

หน่วยความจำซึ่งมีความจุมาก ๆ ปัจจุบันถูกสร้างให้อยู่ในรูปของไอซีชิพเดียวซึ่งมีขนาดและราค
ถูก การนำเอาไปใช้งานก็สะดวก ระบบความจำที่มีความจุมาก ๆ สามารถนำเอาหน่วยความจำทาง
อิเลคโทรนิคส์หลาย ๆ ชิพมาประกอบกัน เพื่อให้ความจุตามต้องการ

โครงสร้างของชิพหน่วยความจำโดยทั่วไปจะมีสายสัญญาณใช้งานต่าง ๆ เกี่ยวข้องดังต่อไปนี้

1. สายแอดเดรส (ADDRESS LINE) เป็นสัญญาณที่ใช้เลือกตำแหน่งที่จะทำการอ่านหรื
เขียน สายส่วนนี้จะต่อโดยตรงกับแอดเดรสบัส (ADDRESS BUS) ของซีพียู

2. สายดาต้า (DATA IN, DATA OUT) ซึ่งจะเป็นบัสแบบสองทิศทาง
(BID DIRECTION BUS) คือ ใช้ได้เป็น ทั้งเป็นทางเดินของข้อมูลเข้าและข้อมูลออกในเส้นเดียวกัน

3. ชิพอีนาเบิล (CHIP ENABLE) เป็นสายสัญญาณควบคุมให้หน่วยความจำชิพใด ๆ
สามารถทำการอ่านหรือเขียนกับซีพียูได้

4. เขียนหรืออ่าน (WRITE/READ) คือสัญญาณให้ซีพียูสามารถอ่านหรือเขียนกับหน่วย
ความจำที่ถูกเลือก

หน่วยความจำแบ่งออกเป็น 2 ประเภท คือ RAM และ ROM

RAM (RANDOM ACCESS MEMORY)

เป็นหน่วยความจำที่สามารถทำการอ่านหรือเขียนได้ทั้งสองอย่าง ข้อมูลที่เก็บไว้ในหน่วย
ความจำแบบนี้ จะอยู่ได้ตราบใดที่ยังมีไฟเลี้ยงจ่ายให้กับชิพอยู่ และข้อมูลจะหายไปเมื่อไม่มีไฟเลี้ยง
หน่วยความจำแบบนี้เราเรียกว่า STATIC RAM และมี RAM อีกประเภทหนึ่งเรียกว่า DYNAMIC RAM
เป็นหน่วยความจำที่สามารถเก็บข้อมูลของตัวเองได้ในระยะเวลาสั้น ข้อมูลจะสูญหายไป ถึงแม้จะมีไฟ
เลี้ยงอยู่ก็ตาม ดังนั้นจึงต้องมีการ รีเฟรช (REFRESHING) อยู่ทุกระยะข้อมูลจึงจะไม่สูญหาย แต่ถึง
แม้ว่าหน่วยความจำแบบนี้ จะต้องอาศัยขบวนการรีเฟรชซึ่งยุ่งยากแต่ก็มีข้อดีที่คุ้มค่า คือ สามารถสร้า
เป็นหน่วยความจำที่มีความจุสูง ๆ ได้

ROM (READ ONLY MEMORY)

เป็นหน่วยความจำแบบที่สามารถใช้อ่านได้อย่างเดียว กล่าวคือ เรานำเอาข้อมูลเข้าไป
ไว้ซึ่งเรียกว่าการโปรแกรมรอม แล้วข้อมูลจะไม่สูญหายไปถึงแม้จะไม่มีไฟเลี้ยงบ้อนให้ซีพียูก็ตาม ดังนั้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลที่เรานำเข้าไปเก็บไว้จะเป็นข้อมูลที่มีความสำคัญ และต้องใช้งานอยู่เป็นประจำ เช่น โปรแกรม

ควบคุมหรือสั่งทำงานต่าง ๆ เป็นต้น

รอมยังแบ่งออกได้เป็นหลายประเภทดังนี้

ROM โปรแกรมจะเขียนมาแล้วจากโรงงานผู้ผลิตซึ่งมีราคาแพง

PROM (PROGRAMMABLE READ-ONLY MEMORY) เป็นรอมที่ผู้ใช้งานสามารถโปรแกรม

ใช้งานได้ตามความต้องการ แต่ไม่สามารถจะเปลี่ยนแปลงหรือลบข้อมูลได้ กล่าวคือ โปรแกรมได้

เพียงครั้งเดียว

EPRROM (ERASABLE PROGRAMMABLE READ-ONLY MEMORY) เป็นรอมที่ใช้งาน

สามารถทำการโปรแกรมใช้งานเองได้และยังสามารถเปลี่ยนแปลงข้อมูลได้ กล่าวคือลบล้างข้อมูลเก่า

ได้และยังเขียนใหม่ได้อีกหลายครั้ง โดยการฉายแสงอุลตร้าไวโอเลทลงตรงช่องปิดชิพ

EAROM (ELEXTRICALLY ALTERRABLE READ-ONLY MEMORY) การโปรแกรมทำได้

โดยผู้ใช้งานเหมือน EPROM แต่การลบล้างข้อมูลต้องใช้กระแสไฟฟ้า

เมื่อไมโครโปรเซสเซอร์จะอ่านหรือเขียนข้อมูลกับหน่วยความจำ หน่วยควบคุมจะส่งสัญญ

การอ้างแอดเดรสที่แอดเดรส โดยทั่วไปรวมทั้ง Z-80 ซีพียู จะมี 16 เส้น ดังนั้นมันสามารถที่จะอ้าง

แอดเดรสได้ถึง 2 ก.เท่ากับ 65,536 ตำแหน่ง ซึ่งโดยทั่วไป เรานิยมเรียกเป็นหน่วยของพันคือ 102

ตำแหน่ง (ไบท์) เท่ากับ 1 กิโลไบท์ ดังนั้นสายแอดเดรส 16 จะอ้างได้ถึง 64K

ถ้าชิพหน่วยความจำมีความจุ 4K ดังนั้นจะต้องใช้แอดเดรส 12 เส้นคือ A0 - A11

ในกรณี 2K ก็ต้องใช้แอดเดรส 11 เส้น A0 - A10

จากที่ได้กล่าวมาแล้วในตอนต้น การที่จะสร้างระบบหน่วยความจำให้มีขนาดความจุมาก

จำเป็นจะต้องใช้ชิพหน่วยความจำ ซึ่งอาจจะเป็นทั้งรอมและแรมที่มีความจุจำกัดมาประกอบกันเข้าเพื่อ

ให้ได้ความจุตามต้องการ ตัวอย่างเช่นระบบความจำขนาด 10K อาจจะใช้ชิพขนาด 2K จำนวน 5

ตัวประกอบ หรืออาจจะใช้ชิพขนาด 4K-2 ตัวและ 2K อีกหนึ่งตัวมาประกอบก็ได้ ขึ้นอยู่กับการออกแ

และความเหมาะสมของงาน

เมื่อมีชิพหน่วยความจำหลาย ๆ ชิพมาต่อกันดังกล่าว จึงจำเป็นอย่างยิ่งที่จะต้องมีการถอ

รหัสเข้าช่วย เพราะข้อมูลที่อ้างถึงอาจจะอยู่ที่ชิพใด ชิพหนึ่งในระบบการถอดรหัส (DECODE) จะเป็

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การให้สัญญาณอีนาเบิล (ENABLE) แก่ CS ของชิพ ซึ่งจะทำให้ชิพนั้นสามารถเข้าถึงแอดเดรสที่อยู่บน

ชิพนั้นได้ กล่าวคือในการออกแบบฮาร์ดแวร์ แต่ละชิพในระบบความจำจะมีตำแหน่งแอดเดรสที่แน่นอนข

ตัวเอง เช่นชิพที่ 1 ขนาด 2 ให้เป็นแอดเดรส 0000H ถึง 07FF H ชิพที่ 2 ให้เป็นแอดเดรสที่

0800 H ถึง DFFH (มีขนาด 2K) เป็นต้น

8255 I/O INTERFACE

เป็นส่วนที่เชื่อมต่ออุปกรณ์อินพุท เอาต์พุทเข้ากับชิพนี้ โดยวงจรของเครื่อง AUTOMATIC

VIDIO SWITCHER นี้ใช้ไอซี 8255 PPI (PROGRAMMABLE PERIPHERAL INTERFACE) ซึ่งเป็

ไอซีทบรษทอนุเทลผลตขนมาสำหรับสนับสนุนการใช้งานของชิพนี้ 8080 โดยเฉพาะ แต่ก็ยังสามารถ

ที่จะใช้กับชิพนี้ Z-80 และชิพตัวอื่นมีลักษณะของสัญญาณการติดต่อควบคุมต่าง ๆ กันเกือบกันได้ โดย

มีการดัดแปลงแก้ไขนิดหน่อยเท่านั้นเอง

มีลักษณะพิเศษคือ เป็นชิพอินพุท เอาต์พุท ซึ่งสามารถโปรแกรมกำหนดลักษณะการทำงาน

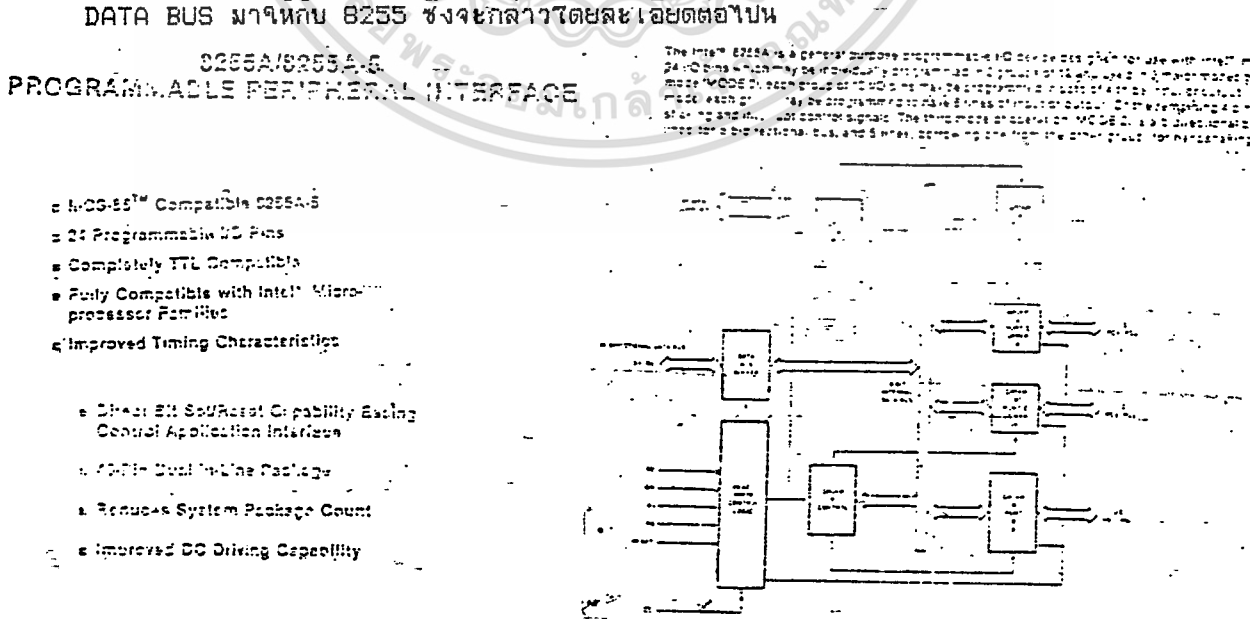
ได้โดย SOFTWARE 8255 เป็นไอซีขนาด 40 ขา โดยจะมีลักษณะของพอร์ทแบบขนานถึง 3 พอร์ท

พอร์ทละ 8 บิท ให้ใช้งานรวมเป็น 24 ขาด้วยกัน การติดต่อกับชิพนี้ทำได้โดยต่อกับ DATA BUS

8 ขา ต่อกับสัญญาณควบคุมอีก 6 ขาและอีก 2 ขาเป็นแหล่งจ่ายไฟกับกราวน์

ชิพนี้สามารถที่จะเลือก FUNCTION การทำงานได้โดยส่งข้อมูล 8 บิทซึ่งเรียกว่าผ่าน

DATA BUS มาที่กับ 8255 ซึ่งจะกล่าวโดยละเอียดต่อไปนี้



รูปที่ 2.20 แสดงตำแหน่งขาและโครงสร้างภายในของ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งขาต่างและโครงสร้างภายในแสดงให้เห็นได้ดังรูปที่ 4.1 ซึ่งอธิบายหลักการทำงานได

DATA BUS BUFFER มีโครงสร้างการทำงานเพื่อจะต่อกับ DATA BUS ของระบบซึ่งเราจะต่อ
โดยตรงกับ DATA BUS ของซีพยู ดังนั้น BUFFER ตัวนี้จะเป็นตัวกักหรือส่งข้อมูลแบบ 2 ทิศทาง จึง
วงจรลอจิกสามสถานะ (TRI STATE) ข้อมูลที่ผ่าน DATA BUS BUFFER นี้จะกระทำได้โดยคำสั่ง
หรือคำสั่ง OUT ซึ่งเป็นคำสั่งมาจากซีพยูและรวมถึงการเลือก MODE (CONTROL WORD และ STATU

READ UNIT CONTROL LOGIC ส่วนนี้จะเป็นส่วนควบคุมและจัดการเกี่ยวกับการเคลื่อนย้ายข้
ของ INTERNAL DATA BUS และ EXTERNAL DATA BUS ควบคุมทั้งสัญญาณที่เป็นข้อมูลและสัญญาณ
ควบคุมเองด้วย ส่วนนี้จะรับสัญญาณ ADDRESS และสัญญาณควบคุมจากซีพยูอีกทีหนึ่งซึ่งตัวมันเองจะทำ
ถอดรหัสเพื่อควบคุมการทำงานของ GROUP A CONTROL และ GROUP B CONTROL

CS CHIP SELECT INPUT ที่ขานี้รับลอจิก (LOGIC) "0" หมายความว่าซีพยู สามาร
ติดต่อ (อ่านหรือเขียน) กับ 8255 ได้

RD READ INPUT ถ้าขานี้รับลอจิก "0" แล้ว 8255 จะทำการส่งข้อมูลจากพอร์ทหรือข้อ
ที่แสดงสถานะการทำงานไปยังซีพยูโดยผ่าน DATA BUS หรือกล่าวได้ว่าซีพยูจะทำการอ่านข้อมูลจาก
8255 นั่นเอง

WR WRITE INPUT เมื่อ 8255 รับสัญญาณเข้ามาที่ขานี้เป็นลอจิก "0" ซีพยูจะสามารถทำกา
ส่ง (เขียน) ข้อมูลซึ่งอาจจะเป็น DATA ไปที่พอร์ทหรืออาจเป็นข้อมูลที่เป็นคำสั่งไปยัง 8255

A0 - A1 (ADDESS INPUT) สัญญาณที่ขานี้จะทำงานพร้อมกับ RD และ WR ทำให้
สามารถเลือกพอร์ทได้ 3 พอร์ทโดยเหลืออีกหนึ่งพอร์ทจะเป็นพอร์ททรับคำสั่งควบคุมที่เรียกว่า
"CONTROL WORD" ซึ่งพอร์ทนี้เราจะเรียกว่า "CONTROLPORT REGISTER" โดยทั่วไปขา A0, A1
นี้จะต่ออยู่กับบิตล่างสุดของ ADDRESS BUS ซึ่งก็คือ A0, A1 ADDRESS BUS นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการเลือกพอร์ทแสดงให้เห็นในตาราง 2.1

A1	A1	RD	WR	CS	Function การทำงาน	
0	0	0	1	0	READ เป็นอินพุท	อ่านข้อมูลเข้าพอร์ท A
0	1	0	1	0		อ่านข้อมูลเข้าพอร์ท B
1	0	0	1	0		อ่านข้อมูลเข้าพอร์ท C
1	1	0	1	0		ไม่ใช้งาน
0	0	1	0	0	WRITE เป็นเอาท์พุท	ส่งข้อมูลออกพอร์ท A
0	1	1	0	0		ส่งข้อมูลออกพอร์ท B
1	0	1	0	0		ส่งข้อมูลออกพอร์ท C
1	1	1	0	0		พอร์ทควบคุมการทำงาน
X	X	X	X	1	ยังไม่มีข้อมูลที่ใช้งานเข้ามา	

ตารางที่ 2.1 แสดงลักษณะการเลือกพอร์ท

RESET เมื่อขาขาอินพุทเป็น "0" ในทางลอจิก CONTROL WORD REGISTER จะถูกเคลียร์ ทำให้ทุกพอร์ทถูกเซทให้เป็นอินพุท โหมด (INPUT MODE)

PA0-PA7 ขาทั้ง 8 ขานี้ถูกใช้เป็นอินพุท เอาท์พุทพอร์ทสำหรับพอร์ท A ซึ่งสามารถที่จะต่อกับอุปกรณ์ภายนอกได้

PB0-PC7 เหมือนกับ PA0-PA7 และ PB0-PB7 นอกจากนี้ยังสามารถแบ่งเป็นอินพุทพอร์ท เอาท์พุทพอร์ทขนาด 4 บิท ได้สองจุดซึ่งสามารถใช้งานได้อย่างอิสระ กล่าวคือแบ่งออกเป็น PORT C UPPER และ PORT C LOWER และยังสามารถใช้เป็นสายควบคุมของ PA0-PA7 และ PB0-PB7 ในการทำงานโหมดอื่นด้วย

GROUP A และ GROUP B CONTROL

การควบคุมการทำงานของ 8255 จะถูกสั่งงานโดยการส่งคำสั่งมาจากซีพียู โดยที่เรา สามารถจะควบคุมได้สองกลุ่มคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GROUP A จะควบคุมไดคือ PORT A และ PORT C UPPER (PC7-PC4)

GROUP B จะควบคุม PORT B และ PORT C LOWER PC3-PC0

PORT A,B,C

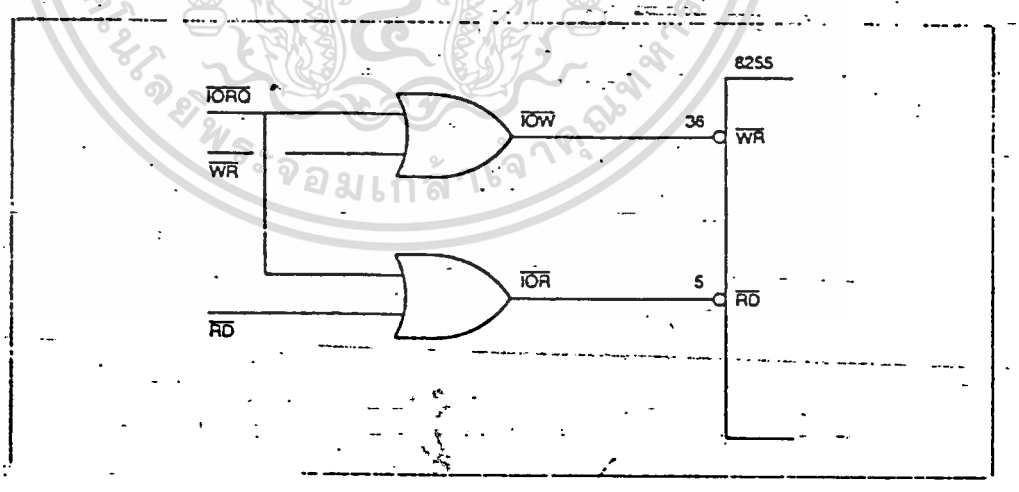
8255 ประกอบด้วยพอร์ท 3 พอร์ทซึ่งจะทำหน้าที่เป็น อินพุท เอาท์พุท

PORT A มี 8 บิททำหน้าที่เป็นเอาท์พุท แลทช์ (OPUPUT LATCH) และอินพุท แลทช์ INPUT LATCH

PORT B มี 8 บิทเป็นอินพุท หรือเอาท์พุท

PORT C อาจจะใช้งานทีละ 8 บิทหรือแบ่งเป็น 4 บิท UPPER และ 4 บิท LOWER นอกจากนี้ยังสามารถเอาไปควบคุม PORT A และ PORT B ได้อีกด้วยซึ่งจะเป็นการใช้งานในโหมด 1 2

จากที่ได้กล่าวมาแล้ว 8255 นั้นสร้างเพื่อสนับสนุนการใช้งานของ 8080 ดังนั้นการต่อสัญญาณควบคุม RD และ WR เข้ากับ IOR และ LOW ของ Z-80 จึงไม่สามารถกระทำได้โดยตรง ซึ่งในทางปฏิบัติเราต้องแปลงให้ต่อเข้าด้วยกันได้ดังรูป 2.21

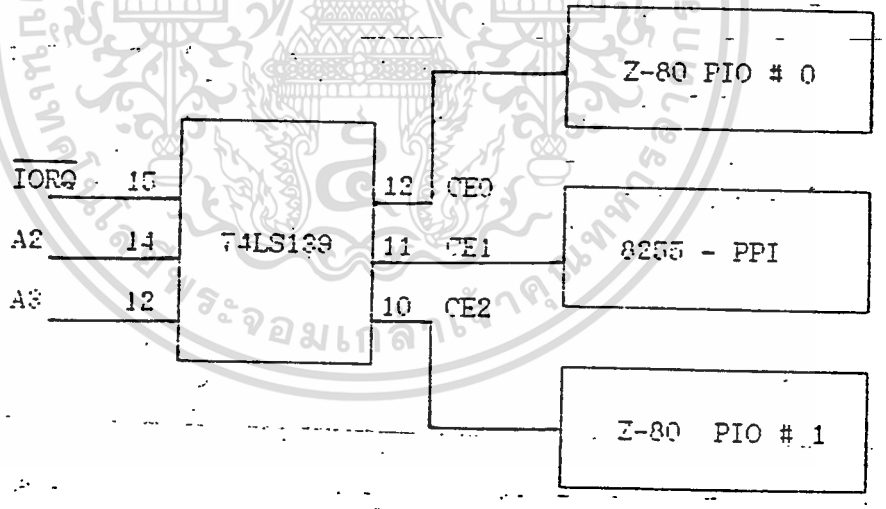


รูป 2.21 แสดงการต่อสัญญาณควบคุมของ Z-80 เข้ากับ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นอกจากนี้ยังมีขาสัญญาณควบคุมอีกเส้นหนึ่งที่ต้องพิจารณาคือสัญญาณ RESET ซึ่งสัญญาณนี้จะ
RESET เครื่อง ในวงจรเครื่องนี้ได้สัญญาณมาจาก POWER SUPPLY ซึ่ง 8255 จะ RESET ที่ไลิก "1"
แต่ Z-80 จะ RESET ที่ไลิก "0" ซึ่งจะเห็นว่าการทำงานไม่เหมือนกัน เพื่อที่จะให้วงจรสามา
RESET ได้โดยใช้สัญญาณ RESET จาก POWER SUPPLY อันเดียวกันคือ RESET ที่ไลิก "0" ดังนั้นสัญญาณ
ที่จะป้อนให้กับ 8255 ต้องผ่าน NOT GATE ก่อนเพื่อให้ 8255 ได้รับสัญญาณ RESET เป็นไลิก "1"
วงจรนี้ใช้ ไอซีเบอร์ 74LS04 เข้าช่วย

จากที่ได้กล่าวมาแล้วขา ขา A0,A1 นั้นต่อโดยตรงกับแอดเดรสบัสของ Z-80 และเป็
กำหนดให้ Z-80 ติดต่อกับ REGISTER ภายในพอร์ทของ 8255 เมื่อพิจารณาร่วมกับสิ่งได้มาจาก
ถอดรหัส (DECODE) โดยในวงจรเครื่องนี้ใช้ไอซีเบอร์ 74LS139 เป็นตัวถอดรหัส (DECODE
ดังวงจรในรูปที่ 4.3 โดยจะมีการเลือก Z-80 PIO อีก 2 ตัว ซึ่งจะกล่าวในบทต่อไป



รูปที่ 2.23 แสดงการจัดวงจรถอดรหัสเลือก CHIP SELECT

แอดเดรสที่นำมากำหนดเงื่อนไขการเลือกคือ A2-A3 การเลือกเป็นตามตารางที่ 2.21

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$\overline{IORQ}$	A3	A2	เลือก
0	0	0	Z-80 PIO,0
0	1	0	Z-80 PIO,1
0	0	1	8255 PPI

ตาราง 2.2 แสดงการเลือก I/O

จะเห็นว่า CS CE1 ของ 8255 เป็น ACTIVE เมื่อ A3-A0 เป็น 01XX ซึ่ง 2 บิท สุดท้ายที่เป็น DON'T CARE จะเป็นบิทที่กำหนด I/O ADDRESS 04H,05H,06H,07H จากตาราง 8.1 ซึ่งเป็นการแสดงเลือกพอร์ทภายใน 8255 จะได้ I/O แอดเดรสคุมพอร์ท คือ PORT A(04H), PORT B (05H), PORT C (06H) ส่วน I/O แอ็ดเดรส 07H จะเป็น CONTROLWORD REGISTER ใช้เป็นตัวกำหนดหน้าที่ของพอร์ทต่าง ๆ ที่กล่าวมาว่าให้เป็นอินพุทพอร์ท หรือเอาท์พุทพอร์ท

การกำหนดทำได้โดยการส่ง CONTROL WORD ขนาด 8 บิท มาให้ CONTROL WORD REGISTER จากสาเหตุนี้เองทำให้เราสามารถโปรแกรมให้ 8255 ทำงานในโหมดต่าง ๆ ได้ 3 โหมดที่แตกต่างกันโดยการกำหนด BIT DEFINITION ของ CONTROL WORD การโปรแกรมการทำงานของ 8255 ทำได้ 3 โหมด คือ

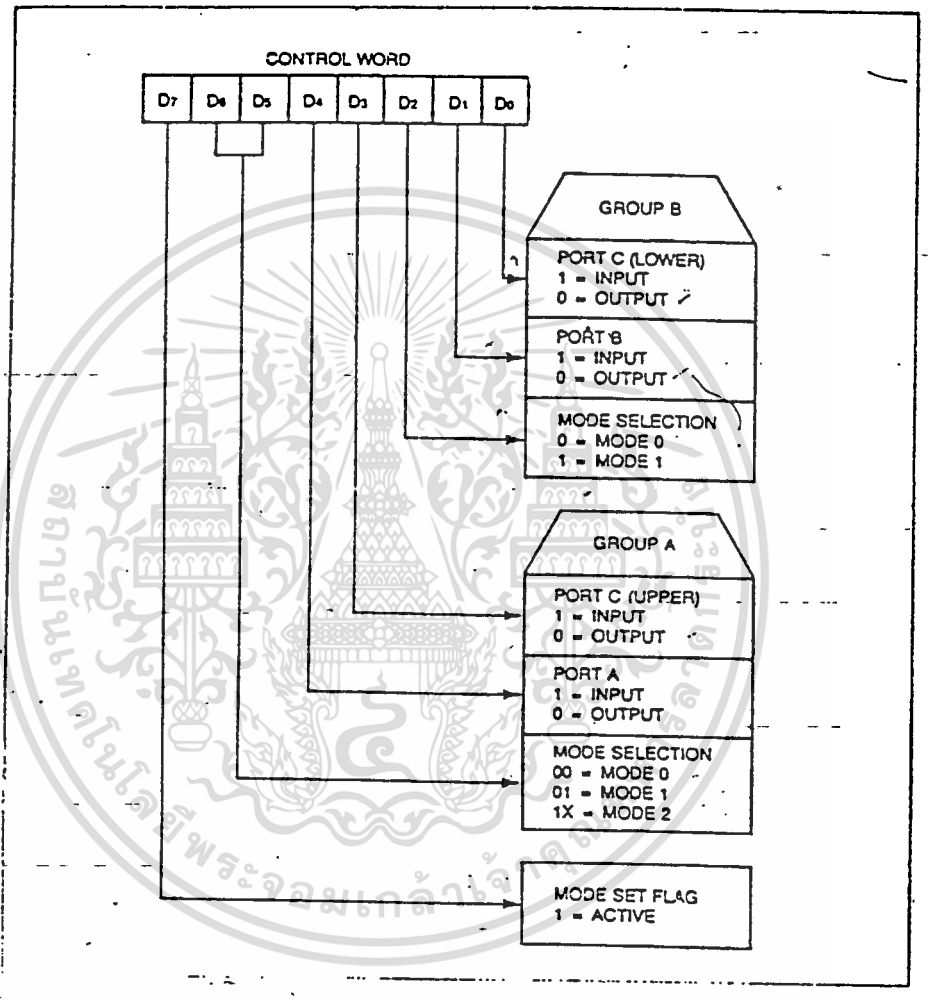
โหมด 0 BASIC REGISTER INPUT/OUTPUT

โหมด 1 STROBE -INPUT/OUTPUT

โหมด 2 BIDIRECTIONAL I/O PORT

ในการทำงานของ I/O INTERFACE นี้ ต้องการให้ทำงานใน โหมด 0 โดยกำหนดให้ PORT A,B เป็นเอาท์พุทพอร์ท, PORT C UPPER (PC7-PC4) เป็นเอาท์พุทพอร์ทและให้ PORT C LOWER (PC3-PC0) เป็นอินพุทพอร์ท โดยการให้แต่ละบิทของ CONTROL WORD เป็นไปตามเงื่อนไขของ IT DEFINITION ดังแสดงในรูป 2.24 ตามการกำหนดค่าแต่ละบิทจะได้ค่าของ CONTROL WORD เป็นตัวเลขออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 2.24 แสดง BIT DEFINITION ของ CONTROL WORD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.24 การกำหนดให้ 8255 ทำงานตามความต้องการในวงจรนั้นแต่ละบิทจะมี
ค่าดังนี้

D7 กำหนดโหมดการทำงาน ถ้าเป็นลอจิก "1" คือแอคทีฟ จะกำหนดเป็น CONTROL
WORD

D6,D5 เป็นบิทกำหนดการเลือกโหมด ถ้าเป็น 00 จะทำให้ 8255 ทำงานในโหมด 0

D4 เป็นลอจิก "0" กำหนดใน PORT A เป็นเอาต์พุทพอร์ท

D3 เป็นลอจิก "0" กำหนด PORT C UPPER (PC7-PC3) เป็นเอาต์พุทพอร์ท

D2 เป็นลอจิก "0" เลือกโหมดการทำงานเป็นโหมด 0

D1 เป็นลอจิก "0" กำหนดให้ PORT B เป็นเอาต์พุทพอร์ท

เป็นลอจิก "1" กำหนดให้ PORT C LOWER (PC3-PC0) เป็นอินพุทพอร์ท

จากการกำหนดแต่ละบิทตามนี้จะได้ค่า CONTROL WORD เป็น 81H การเซ็ท 8255 ทำ
โปรแกรมดังนี้

```
LD A,81H
OUT(07H),A
```

ก็จะได้ 8255 ทำงานตามความต้องการในวงจร

สาเหตุที่กำหนด 8255 ให้ทำงานดังนี้ก็เพื่อจะใช้ PORT A, B เป็นส่วนแสดงผล
(DISPLAY) ซึ่งเป็นแบบเซคเมนท์ (SEGMENT) และสามารถป้อนข้อมูลแบบ ASCII ได้เลยซึ่ง
รายละเอียดจะกล่าวในหัวข้อเรื่องส่วนแสดงผล (DISPLAY) ส่วน PORT C ใช้เป็น คีย์บอร์ด
(KEYBOARD)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การแสดงและการทำงานของ Programmable Score Board Display คุณสมบัติของอุปกรณ์

SPECIFICATIONS :

-	ขนาดจอ DISPLAY	ROW x COLUMN	16 CM. x 95 CM.
-	RESOLUTION LED	ROW x COLUMN	16 x 95 DOTS.
-	จำนวนตัวอักษร		16 CHARACTERS
-	ภาษาที่ใช้		THAI/ENGLISH
-	STORAGE CHARACTER		1,000 CHARACTERS
-	POWER		20 WATTS.
-	SUPPLY		220 V. 50 HZ.
-	KEYBOARD		KEY IBM COMPATIB
-	SPECIAL COMMAND		25 COMMANDS
-	PATTERN ENGLISH		4 PATTERNS
-	PATTERN THAI		2 PATTERNS

FEATURE

-	CPU	Z80A	
-	CLOCK	4 MHZ	
-	MEMORY	ROM (MONITOR)	8 KBYTES.
		RAM (DATA)	2 KBYTES.
		CHARACTER GEN.	4 KBYTES.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PORT

  SHIFT REGISTER 74164

  DECODER 74154

  PARALLEL PORT 8255

- DISPLAY LED ROW 16 PCS.

  COLUMN 95 PCS.

- POWER SUPPLY

  LOGIC 5 V. (IC 7805) 2 PCS.

  DRIVE LED 25 V. (TRANSISTOR MJ 2955)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรและหลักการทำงาน

PROGRAMMABLE SCORE BOARD DISPLAY สามารถแบ่งการทำงานได้ออกเป็น 4 ส
คือ

1. CONTROLLER BOARD
2. KEYBOARD AND KEY INTERFACE BOARD
3. LED DISPLAY BOARD
4. POWER SUPPLY BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. CONTROLLER BOARD เป็นวงจรควบคุมการทำงานทั้งหมดของระบบทุก ๆ ส่วน ไมโครโปรเซสเซอร์เป็นตัวควบคุม และกำหนดตามโปรแกรม คือ CPU 'Z80' เมื่อเริ่มต้นจ่าย SUPPLLY ให้แก่วงจร CPU จะเริ่มต้นผ่านโปรแกรมจาก ROM (2764) ซึ่งทำหน้าที่เป็น MONIT PROGRAM หรือ OS ของระบบ โดยภายในโปรแกรม จะทำการ INITIALIZED ข้อมูล SET ให้แก่ PORT 8255 และควบคุมการ DISPLAY LED เบื้องต้น เมื่อมีการป้อนข้อมูลเข้าทางคีย์บอ CPU จะเก็บข้อมูลไว้ใน RAM (6116) การแสดงผลออกทาง LED DISPLAY CPU จะมาอ่านข้อ DISPLAY CHARACTER GENERATOR (2732) เพื่อแปลงจากข้อมูลรหัส ASCII ไปเป็น DISPL CODE (เพื่อไปออกแสดงผล) เมื่อได้ DISPLAY CODE แล้วก็จะส่งไปให้วงจร DECODE (7415 เพื่อแปลงเป็นสัญญาณควบคุม ออกทาง JP20 เพื่อไปให้วงจร LED DISPLAY BOARD ไปแสดง ไป

การเลือก ADDRESS สามารถเลือกได้โดยวงจร DECODER (74138) ส่วนการเลือ PORT ใช้ DECODER (74138) เช่นเดียวกัน

เมื่อ CPU ต้องการนำข้อมูลใน RAM ออกแสดงผล CPU จะอ่านข้อมูลจาก RAM เข้า แล้วทำการเปิด TABLE ของโค๊ด ASCII แปลงไปเป็น DISPLAY CODE โดยอ่านค่าตารางจ DISPLAY CHAR GEN เมื่อได้โค๊ดแล้วจึงส่งออกไปที่วงจร LATCH (74374) ที่ LED DISPL BOARD ต่อไปส่วนรูปแบบการวิ่ง PATTERN DISPLAY จะควบคุมด้วย DECODE 4 TO16 (7415 แล้วส่งไปที่ LED DISPLAY BOARD ต่อไป เพื่อทำการป้อนเข้าวงจร 1. to 8 SHIFT REGIST

การรับข้อมูลจาก KEYBORAD CPU จะส่ง CS เข้า B255 ที่ KEY INTERFACE BO อ่านข้อมูลทาง KEYBOARD ได้เป็น SCAN CODE และส่ง CS แก่ 74374 เพื่ออ่านข้อมูที่แปลงจาก SCAN CODE มาเป็น ASCII CODE เข้ามาทาง JP16 แล้วนำข้อมูลมาเก็บไว้ใน RAM และโปร LED DISPLAY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. KEY INTERFACE BOARD เป็นวงจร INTERFACE ระหว่าง CONTROLLER B

กับ KEYBOARD ภายในประกอบด้วย PORT 8255 ซึ่งโปรแกรมให้อยู่ในโหมด 0 มี PORT A เป็น I

PORT C UPPER เป็น INPUT LOWER เป็น OUTPUT โดยการทำงานรับ KEY จะเป็นไปดังนี้

การกดคีย์เกิดขึ้น จะมีสัญญาณจากขา KBD CLK และขา KBD DATA ซึ่งจะส่งไปยัง 74322 ทำหน้

การแปลงข้อมูลจาก SERIAL ไปเป็นแบบ PARALLEL แล้วจึงส่งเข้า PORT A ของ 8255

เวลา KEYBOARD ส่งข้อมูลครบ 1 CHARACTER จะมีสัญญาณ INTERRUPT เข้า INPUT

ของ8255 เมื่อ CPU อ่านพบก็จะ OUT '0' ออกที่ PC0 เพื่อหยุดสัญญาณ KBD CLK จากนั้นก็จะท

านข้อมูลทาง PORT A ของ 8255 แล้วส่งข้อมูลนี้ไป LATCH ที่ 74374 โดยควบคุมที่ #PS2 ข

จาก LATCH ก็จะไปเปิด TABLE ที่ KEY GENERATOR เพื่อแปลงรหัส SCAN CODE เป็น

ASCII CODE แล้วส่งไป LATCH ที่ 374 อีกตัวหนึ่งเพื่อรอให้ CPU รออ่านเข้าไปอีกต่อไป

ดังนั้นวงจร KEY INTERFACE จึงมีหน้าที่ในการรับข้อมูล KEYBOARD แล้วแปลงข้

ของ KEYBOARD จากเดิมเป็น SCAN CODE มาเป็น ASCII CODE เพื่อให้ CPU สามารถอ่านเข้

โปรเซสได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. วงจร LED DISPLAY BOARD จะรับข้อมูลสัญญาณการแสดงผลออก LED MATI

โดยสัญญาณเหล่านี้ จะเป็นสัญญาณควบคุม LED ทางด้าน ROW กับ COLUMN ซึ่งจะให้ LED ติดส

และกระพริบ ตามต้องการภายในวงจร ประกอบด้วย 8 PARALLEL OUT SHIFT REGIST

(74164) ทำหน้าที่รับข้อมูลจาก สัญญาณข้อมูลควบคุมทาง COLUMN เข้า 1 มาแปลงเป็นสัญ

SHIFT OUT ออก 8 จุดโดยจังหวะการ SHIFT จะควบคุมทาง COLUMN CLOCK ส่วนสัญญาณ SH

OUT เหล่านี้ จะผ่าน DRIVER (UN2003) แล้วไปควบคุม LED ทางด้าน COLUMN ต่อไป เพ

ฉะนั้นมสัญญาณจาก JP20 เป็นสัญญาณควบคุม COLUMN 12 เส้น เมื่อป้อนเข้า SHIFT REGISTER

ตัว ก็จะกระจายเพิ่มการควบคุม COLUMN ได้อีก ตัวละ 8 เส้น ดังนั้นก็จะควบคุม COLUMN เท่

8X12 = 96 COLUMNS ส่วนการควบคุม LED ทางแนว ROW จะรับสัญญาณ D0-D7 ผ่านวงจร LA

(74374) 2 ตัว เพื่อนำข้อมูลมา LATCH ออกเป็นสัญญาณ DISPLAY ROW โดยแต่ละตัวจะ L

ตัวละ 8 BITSโดยมีสัญญาณควบคุม A/B เป็นตัวเลือกสัญญาณที่จะไป OUT ออกที่ 74374 ได้ทีละตัว

ทำให้มสัญญาณ DISPLAY ROW เป็น 16 BITS ได้ จากนั้นสัญญาณเหล่านี้จะผ่านเข้าไป DRIVE

POWER TRANSISTOR ที่จะไปรับ LED ทาง ROW ทั้ง 16 แถว

จากวงจรจะเห็น LED ต่อ CROSS ระหว่าง ROW กับ COLUMN โดย ANODE จะต่

COLUMN และ CATHODE จะต่อกับ ROW วิธีการควบคุมให้ LED แต่ละดวงดับและมีการวงเป็นตัวอ

ต่าง ๆ สามารถทำได้โดยการส่งให้ CPU OUT ข้อมูลมาที่ PORT #PS4+0 เป็นการ LATCH 8 B

ล่างแล้วส่งให้ OUT PORT PC4+1 เป็นการ LATCH 8 BITS บน ซึ่งจะได้สัญญาณควบคุม ROW

16 BITS ไป DRIVE LED 'ROW' ตามข้อมูลที่ส่งมา ส่วนการติดสว่างขึ้นอยู่กับทาง COLUMN ถ้า

ที่ ROW เป็น '0' ส่วนทาง CQLUMN เป็น '1' LED ตำแหน่งนั้นก็จะติดสว่าง แต่ถ้า COLUMN

' LED ก็จะดับเพราะไม่ได้รับ BIAS เพียงพอที่จะติด แต่ถ้าให้ ROW เป็น '1' ไม่ว่า COLUMN

เป็น '0' หรือ '1' LED ก็จะดับเสมอ เพราะถูก REVERSE BIAS โดยพิจารณาดูจากรูปจะเห็น

ที่ COL1, ROW1 จะติด และ COL3, RO1 จะติดตามสัญญาณควบคุม ROW กับ COLUMN ที่ให้ม

FORWARD BIAS แก่ LED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการที่ทำให้ LED DISPLAY สามารถแสดงเป็นอักษรวิ่ง โดยการให้สัญญาณที่ COL

SHIFT ไปตามลำดับไล่จาก COLLUMN 0 SHIFT ไปทีละบิทไป COLUMN 1 แล้วไล่ไปเรื่อยจ

COLUMN 7 จากนั้นจึงเลื่อนไปที่ COLUMN 8 จนกระทั่งถึง COLUMN 95 ซึ่งเป็นการ SH

ไปตลอดแถวทุก ๆ COLUMN จังหวะการ SHIFT จะขึ้นอยู่กับสัญญาณ COLUMN CLOCK จากว

CONTROLLER BOARD โดยความเร็วของการวิ่งจะขึ้นอยู่กับความถี่ของสัญญาณนั้น ถ้าให้ความถี่สูง ก

ของตัวอักษรก็จะเร็วขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. วงจร POWER SUPPLY BOARD นี้จะรับไฟ AC ที่ผ่านการ STEP DOWN มาจ

TRANSFORMER ที่แปลงจากไฟ AC 220 V ไฟ AC ที่ได้จะผ่านวงจร RECTIFIER มาจาก DIO

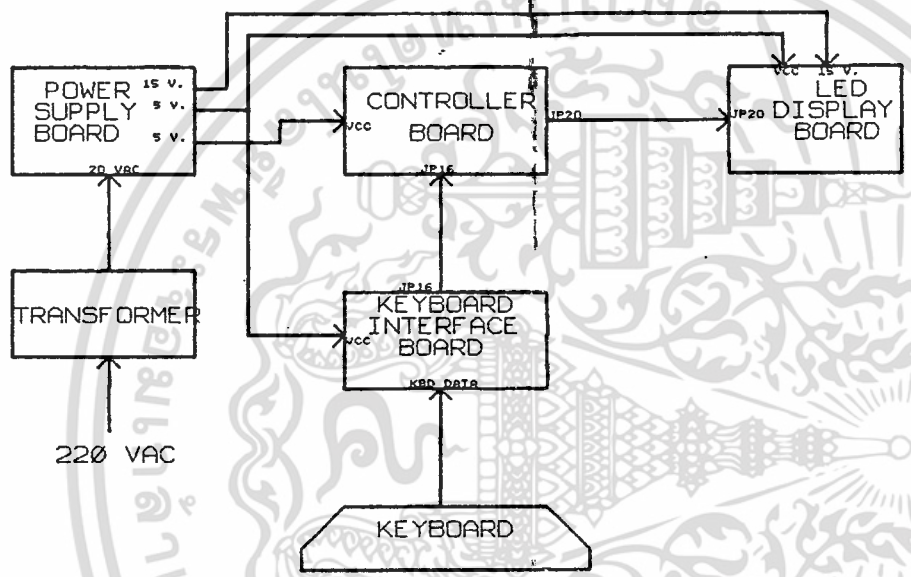
และมี CAPACITOR ทำการ FILTER ให้ได้ DC ที่ 25 V. จากนั้นผ่านเข้าวงจร REGULAT

TRANSISTOR เพื่อ REGULATE ไฟที่ 15 V. ซึ่งจะไป DRIVE LED ให้สว่างทั่ง BOARD และจ

DC นี้จะผ่าน REGULATOR IC 7805 2 ตัว ซึ่งจะให้แรงดันลงที่ 5 V. ป้อนให้วงจร CONTROL

BOARD, KEYBOARD INTERFACE BOARD และวงจร LED DISPLAY BOARD ดังวงจรที่แสดงต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



BLOCK DIAGRAM "PROGRAMMABLE SCORE BOARD DISPLAY"

BLOCK DIAGRAM OF SYSTEM

Size	Document Number	REV
C		
Date:	October 25, 1990 Sheet	of

ศูนย์วิทยทรัพยากร
จุฬาลงกรณ์มหาวิทยาลัย

MAIN CPU
Z80

ROM MONITOR

ROM CHARACTOR GEN.

RAM
DATA MEMORY

SPEED CONTROL

WRITE RAM PROTECT

COLUMN CLOCK

FILENAME CONTROL.SCH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INTERFACE KEYBOARD

VCC. = +5 V.
GND. = 0V.

KEYBOARD CONECTOR

KBD VCC.
KBD CLK.
KBD DATA
KBD RESET
KBD GND.

HEADER 16

XTAL 6 MHz.

100PF.

R10K

R10K

KEY CLOCK

VCC. PIN 14
GND. PIN 7

R1 10K
R2 10K
R3 10K

8255

74322

7474

74175

7404

7407

74LS374

2732

7432

LATCH

KEY GENERATOR

KEY INTERFACE BOARD	
Size: C	Document Number:
Date: October 25, 1990	Sheet 4 of 5

REV

FILENAME KEY.SCH

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

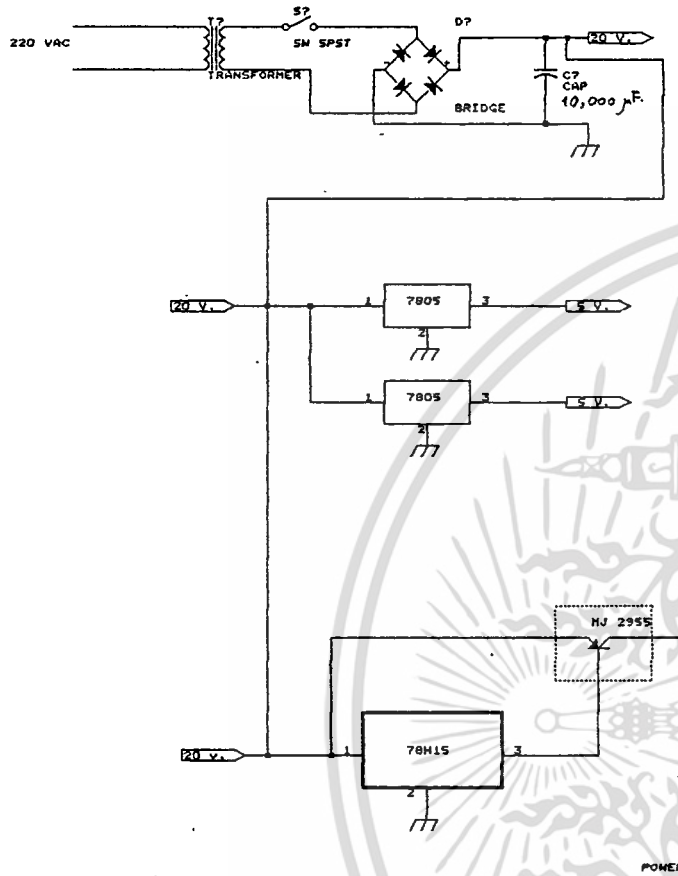
CONTROL COLUMN

CONTROL ROW

LED MATRIX DISPLAY 16 x 95

FILE. DISPLAY_SCH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

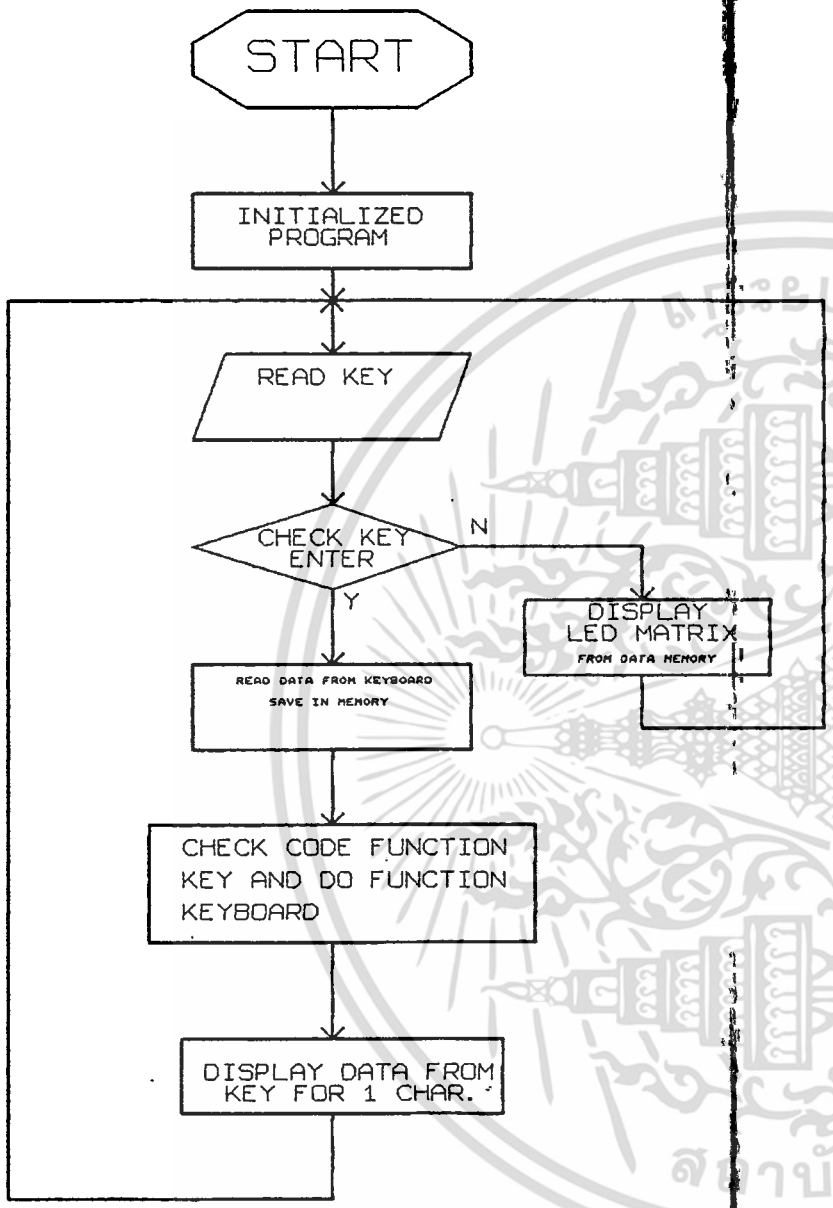


POWER SUPPLY BOARD

Size	Document Number	REV
C		
Date: October 25, 1990	Sheet of	

FINAME SUPPLY.SCH

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



FLOW CHART PROGRAM

FILNAME . . FLOW. SCH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
2500 A.D. Z80 CROSS ASSEMBLER   -   VERSION 3.00b
-------------------------------------------------

        INPUT  FILENAME :  PROGRAM.Z80
        OUTPUT FILENAME :  PROGRAM.OBJ



                ;*****************************************************************************
                ;**  PROGRAM OPERATION SYSTEM FOR RUN IN CONTROLLER BOARD                   **
                ;**  USING AREA FOR 8 KBYTES. (RESERVE FOR EXTEND IN FUTURE)                **
                ;**  THIS IS THE MAIN PROGRAM AT THE FIST STARTING TO EXECUTE               **
                ;**  PROGRAM AND CONTROL LED DISPLAY, READ DATA FROM KEYBOARD.              **
                ;*****************************************************************************

                ;**********  USING AREA  **********
00 00           ROMAREA       EQU     0000H
00 20           CHARGENAREA   EQU     2000H
00 40           RAMAREA       EQU     4000H
FF 47           SYSSTACK      EQU     47FFH

                ;****************************************
                ; DEFINE DATA TO USING ALL      *******
                ;****************************************
                ;
                ;  USING PORT


00 20           ETBL          EQU     2000H
00 28           TTBL          EQU     2800H
00 00           KBDPPI        EQU     00H
20 00           LATCHSCAN     EQU     20H
40 00           LATCHASCII    EQU     40H
60 00           COLOUT        EQU     60H
80 00           ROWOUT        EQU     80H
80 00           PPIMODE       EQU     80H
05 00           CECODE        EQU     05H
15 00           ESC           EQU     15H
70 00           CAP           EQU     70H
71 00           LARROW        EQU     71H
72 00           RARROW        EQU     72H
73 00           UPARROW       EQU     73H
74 00           DNARROW       EQU     74H
75 00           F1            EQU     75H
76 00           CTRL          EQU     76H
0D 00           ENTER         EQU     0DH
```

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        ;            M A I N   P R O G R A M
                        ;*************************************************
0000                                ORG     ROMAREA

0000    AF              MAIN:       XOR     A                   ;INITIALIZED PROGRAM
0001    D3 60                       OUT     (COLOUT),A
0003    D3 80                       OUT     (ROWOUT),A
0005    D3 81                       OUT     (ROWOUT+1),A
0007    32 04 40                    LD      (HROWBIT),A
000A    32 03 40                    LD      (LROWBIT),A
000D    32 02 40                    LD      (COLBIT),A
0010    3E 80                       LD      A,PPIMODE
0012    D3 03                       OUT     (KBDPPI+3),A
0014    31 FF 47                    LD      SP,SYSSTACK
0017    21 00 08                    LD      HL,0800H
001A    22 07 40                    LD      (KEYCNT),HL

                        ;CHECK INPUT DATA FROM KEYBOARD
001D    CD 88 02        MAINDO:     CALL    READKEY     ;READ ENTER KEYBOARD
0020    3A 00 40                    LD      A,(ASCIICODE)
0023    FE 0D                       CP      ENTER
0025    20 05                       JR      NZ,READDATA
0027    CD 29 02                    CALL    DISPLAY
002A    18 F1                       JR      MAINDO
002C    CD 88 02        READDATA:   CALL    READKEY     ;READ KEY DATA SAVE IN MEMORY
002F    3A 00 40                    LD      A,(ASCIICODE)
0032    2A 05 40                    LD      HL,(KEYBUFF)  ;INCREMENT POINTER BUFFER
0035    77                          LD      (HL),A
0036    23                          INC     HL
0037    22 05 40                    LD      (KEYBUFF),HL
003A    ED 5B 07 40                 LD      DE,(KEYCNT)
003E    1B                          DEC     DE
003F    ED 53 07 40                 LD      (KEYCNT),DE
0043    7A                          LD      A,D
0044    B3                          OR      E
0045    C2 4E 00                    JP      NZ,NBUFFULL ;IF KEYCNT = 0 THEN STOP
0048    CD AC 02                    CALL    BUFFULL
004B    C3 1D 00                    JP      MAINDO
004E    3A 00 40        NBUFFULL:   LD      A,(ASCIICODE)
0051    FE 54                       CP      'T'     ;PRESS KEY 'T'
0053    C2 62 00                    JP      NZ,OTHER1
0056    21 00 28                    LD      HL,TTBL         ;LOAD FONT THAI
0059    C6 10                       ADD     A,16            ;PLUS DATA TO 16 ADDRESS
005B    85                          ADD     A,L             ;PLUS POINTER VALUE
005C    CD 46 02                    CALL    OUTCHAR  ;OUT LED DISPLAY FOR 1 CHARACTOR
005F    C3 1D 00                    JP      MAINDO
0062    FE 76           OTHER1:     CP      CTRL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0067  3A 0A 40                  LD      A,(CTLFLG)
006A  CB C7                     SET     0,A
006C  32 0A 40                  LD      (CTLFLG),A
006F  C3 1D 00                  JP      MAINDO
0072  FE 45       OTHER2:       CP      'E'     ;CHECK KEY 'E'
0074  C2 95 00                  JP      NZ,OTHER3
0077  47                        LD      B,A     ;SAVE OLD DATA
0078  3A 0A 40                  LD      A,(CTLFLG)
007B  A7                        AND     A
007C  CA 8D 00                  JP      Z,NOCTRL
007F  21 00 20                  LD      HL,ETBL    ;LOAD FONT ENGLISH
0082  3E 05                     LD      A,CECODE
0084  C6 10                     ADD     A,16
0086  85                        ADD     A,L
0087  CD 46 02                  CALL    OUTCHAR
008A  C3 1D 00                  JP      MAINDO
008D  78           NOCTRL:      LD      A,B
008E  85                        ADD     A,L
008F  CD 46 02                  CALL    OUTCHAR
0092  C3 1D 00                  JP      MAINDO
0095  FE 70       OTHER3:       CP      CAP     ;CHECK KEY CAPLOCK
0097  C2 AD 00                  JP      NZ,OTHER4
009A  47                        LD      B,A
009B  3A 0B 40                  LD      A,(CAPFLG)
009E  CB C7                     SET     0,A
00A0  32 0B 40                  LD      (CAPFLG),A
00A3  78                        LD      A,B
00A4  C6 10                     ADD     A,16
00A6  85                        ADD     A,L
00A7  CD 46 02                  CALL    OUTCHAR
00AA  C3 1D 00                  JP      MAINDO
00AD  FE 71       OTHER4:       CP      LARROW      ;CHK LEFT ARROW KEY
00AF  C2 BE 00                  JP      NZ,OTHER5
00B2  ED 4B 05 40               LD      BC,(KEYBUFF)
00B6  0D                        DEC     C           ;DECREMENT POINTER
00B7  ED 43 05 40               LD      (KEYBUFF),BC
00BB  C3 1D 00                  JP      MAINDO
00BE  FE 72       OTHER5:       CP      RARROW      ;CHK RIGHT ARROW KEY
00C0  C2 CF 00                  JP      NZ,OTHER6
00C3  ED 4B 05 40               LD      BC,(KEYBUFF)
00C7  0C                        INC     C
00C8  ED 43 05 40               LD      (KEYBUFF),BC  ;INCREMENT POINTER
00CC  C3 1D 00                  JP      MAINDO
00CF  FE 15       OTHER6:       CP      ESC         ;MOVE CHAR. TO LEFT AND DELETE 1 CHAR.
00D1  C2 E9 00                  JP      NZ,OTHER7
00D4  ED 4B 05 40               LD      BC,(KEYBUFF)
00D8  0D                        DEC     C
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
00DD   ED 4B 07 40             LD      BC,(KEYCNT)
00E1   0D                      DEC     C
00E2   ED 43 07 40             LD      (KEYCNT),BC
00E6   C3 1D 00                JP      MAINDO
                    ;DISPLAY PATTERN CODE FOR CONTROL MOVEMENT OF LED DISPLAY
00E9   FE 50        OTHER7:    CP      'P'             ;CHECK KEY P
00EB   C2 0C 01                JP      NZ,OTHER8
00EE   47                      LD      B,A
00EF   3A 0A 40                LD      A,(CTLFLG)
00F2   CB 47                   BIT     0,A
00F4   C2 02 01                JP      NZ,NO_FUNC
00F7   3A 0C 40                LD      A,(CPFLG)
00FA   CB C7                   SET     0,A
00FC   32 0C 40                LD      (CPFLG),A
00FF   C3 1D 00                JP      MAINDO
0102   78          NO_FUNC:   LD      A,B
0103   C6 10                   ADD     A,16
0105   85                      ADD     A,L
0106   CD 46 02                CALL    OUTCHAR
0109   C3 1D 00                JP      MAINDO
010C   FE 71        OTHER8:    CP      LARROW    ;CHK LEFT ARROW KEY
010E   C2 47 01                JP      NZ,OTHER9
0111   3A 0C 40                LD      A,(CPFLG)
0114   CB 47                   BIT     0,A     ;CHK CONTROL+P KEY
0116   CA 2B 01                JP      Z,NO_C_P
0119   06 0C                   LD      B,12
011B   3A 02 40     SHIFT:     LD      A,(COLBIT)
011E   CB 27                   SLA     A
0120   32 02 40                LD      (COLBIT),A
0123   CD 62 02                CALL    OUTWORD
0126   10 F3                   DJNZ    SHIFT
0128   C3 1D 00                JP      MAINDO
012B   CB 4F        NO_C_P:    BIT     1,A
012D   CA 1D 00                JP      Z,MAINDO    ;CHECK F1+6+P KEYS
0130   06 0C                   LD      B,12
0132   3A 02 40     SHIFTL:    LD      A,(COLBIT)
0135   CB 27                   SLA     A
0137   32 02 40                LD      (COLBIT),A
013A   D3 60                   OUT     (COLOUT),A
013C   CD 62 02                CALL    OUTWORD
013F   CD 29 02                CALL    DISPLAY
0142   10 EE                   DJNZ    SHIFTL
0144   C3 1D 00                JP      MAINDO
0147   FE 72        OTHER9:    CP      RARROW      ;CHK RIGHT ARROW KEY
0149   C2 84 01                JP      NZ,OTHER10
014C   3A 0C 40                LD      A,(CPFLG)
014F   CB 47                   BIT     0,A
```

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0154   08 0C              LD      B,12
0156   3A 02 40    RSHIFT: LD      A,(COLBIT)
0159   CB 27               SLA     A
015B   32 02 40            LD      (COLBIT),A
015E   D3 60               OUT     (COLOUT),A
0160   CD 62 02            CALL    OUTWORD
0163   10 F1               DJNZ    RSHIFT
0165   C3 1D 00            JP      MAINDO
0168   CB 4F       NOT_C_R: BIT    1,A
016A   CA 1D 00            JP      Z,MAINDO
016D   06 0C               LD      B,12
016F   3A 02 40    SHIFTRR: LD     A,(COLBIT)
0172   CB 27               SLA     A
0174   32 02 40            LD      (COLBIT),A
0177   D3 60               OUT     (COLOUT),A
0179   CD 62 02            CALL    OUTWORD
017C   CD 29 02            CALL    DISPLAY
017F   10 EE               DJNZ    SHIFTRR
0181   C3 1D 00            JP      MAINDO
0184   FE 53       OTHER10: CP     'S'         ; CHECK KEY 'S'
0186   C2 A7 01            JP      NZ,OTHER11
0189   47                  LD      B,A
018A   3A 0A 40            LD      A,(CTLFLG)
018D   CB 47                BIT     0,A
018F   C2 9D 01            JP      NZ,NOT_FUNC
0192   3A 0D 40            LD      A,(CSFLG)
0195   CB C7               SET     0,A
0197   32 0D 40            LD      (CSFLG),A
019A   C3 1D 00            JP      MAINDO
019D   78           NOT_FUNC: LD    A,B
019E   C6 10                ADD     A,16
01A0   85                   ADD     A,L
01A1   CD 46 02             CALL    OUTCHAR
01A4   C3 1D 00             JP      MAINDO
01A7   FE 73        OTHER11: CP     UPARROW     ;KEY UP ARROW
01A9   C2 DA 01             JP      NZ,OTHER12
01AC   3A 0D 40             LD      A,(CSFLG)
01AF   CB 47                BIT     0,A
01B1   CA 1D 00             JP      Z,MAINDO
01B4   06 10                LD      B,16        ;SCRPLL UP FUNCTION
01B6   1E 00        USHIFT: LD      E,00H
01B8   3A 04 40             LD      A,(HROWBIT)
01BB   CB 27                SLA     A
01BD   32 04 40             LD      (HROWBIT),A
01C0   D3 81                OUT     (ROWOUT+1),A
01C2   D2 C7 01             JP      NC,NOCBIT
01C5   1E 01                LD      E,01H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
01CA   CB 27                    SLA     A
01CC   B3                       OR      E
01CD   32 03 40                 LD      (LROWBIT),A
01D0   D3 80                    OUT     (ROWOUT),A
01D2   CD 62 02                 CALL    OUTWORD
01D5   10 DF                    DJNZ    USHIFT
01D7   C3 1D 00                 JP      MAINDO
01DA   FE 74            OTHER12:CP      DNARROW   ;CHK DOWN ARROW
01DC   C2 0D 02                 JP      NZ,OTHER13
01DF   3A 0D 40                 LD      A,(CSFLG)
01E2   CB 47                    BIT     0,A
01E4   CA 1D 00                 JP      Z,MAINDO
01E7   06 10                    LD      B,16
01E9   1E 00            DNSHIFT:LD      E,00H     ;DO SCROLL DOWN FUNCTION
01EB   3A 03 40                 LD      A,(LROWBIT)
01EE   CB 2F                    SRA     A
01F0   32 03 40                 LD      (LROWBIT),A
01F3   D3 80                    OUT     (ROWOUT),A
01F5   D2 FA 01                 JP      NC,NOTCBIT
01F8   1E 01                    LD      E,01H
01FA   3A 04 40         NOTCBIT:LD      A,(HROWBIT)
01FD   CB 2F                    SRA     A
01FF   B3                       OR      E
0200   32 04 40                 LD      (HROWBIT),A
0203   D3 81                    OUT     (ROWOUT+1),A
0205   CD 62 02                 CALL    OUTWORD
0208   10 DF                    DJNZ    DNSHIFT
020A   C3 1D 00                 JP      MAINDO
020D   FE 75            OTHER13:CP      F1    ;DO F1 FUNCTION KEY
020F   C2 1D 02                 JP      NZ,OTHER14
0212   3A 0C 40                 LD      A,(CPFLG)
0215   CB CF                    SET     1,A
0217   32 0C 40                 LD      (CPFLG),A
021A   C3 1D 00                 JP      MAINDO
021D   3A 00 40         OTHER14:LD      A,(ASCIICODE)  ;NORMAL CHARACTOR DISPLAY
0220   C6 10                    ADD     A,16
0222   85                       ADD     A,L
0223   CD 46 02                 CALL    OUTCHAR
0226   C3 1D 00                 JP      MAINDO

                        ;END MAIN PROGRAM


                        ;*************************************************************
                        ;SUBROUTINE DISPLAY DATA IN MEMORY TO LED DISPLAY        ******
                        ;*************************************************************
0229   F5               DISPLAY:PUSH    AF
```

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
022B   ED 4B 05 40            LD      BC,(KEYBUFF)
022F   ED 5B 07 40            LD      DE,(KEYCNT)
0233   3A 05 40      DO_DISP:  LD      A,(KEYBUFF)
0236   C6 10                   ADD     A,16
0238   85                      ADD     A,L
0239   CD 46 02                CALL    OUTCHAR
023C   03                      INC     BC
023D   1B                      DEC     DE
023E   7A                      LD      A,D
023F   B3                      OR      E
0240   C2 33 02                JP      NZ,DO_DISP
0243   E1                      POP     HL
0244   F1                      POP     AF
0245   C9                      RET
                ;*************************************************
                ;SUBROUTINE OUT CHARACTOR TO LED DISPLAY      ********
                ;*************************************************
0246   F5       OUTCHAR:       PUSH    AF
0247   E5                      PUSH    HL
0248   C5                      PUSH    BC
0249   06 08                   LD      B,8
024B   7E       LOOP:          LD      A,(HL)
024C   32 03 40                LD      (LROWBIT),A
024F   23                      INC     HL
0250   7E                      LD      A,(HL)
0251   32 04 40                LD      (HROWBIT),A
0254   D3 81                   OUT     (ROWOUT+1),A
0256   3A 03 40                LD      A,(LROWBIT)
0259   D3 80                   OUT     (ROWOUT),A
025B   23                      INC     HL
025C   10 ED                   DJNZ    LOOP
025E   C1                      POP     BC
025F   E1                      POP     HL
0260   F1                      POP     AF
0261   C9                      RET
                ;*************************************************
                ;SUBROUTINE OUT WORD CHARACTORS TO LED DISPLAY ********
                ;*************************************************
0262   F5       OUTWORD:       PUSH    AF
0263   E5                      PUSH    HL
0264   C5                      PUSH    BC
0265   06 0C                   LD      B,12
0267   7E       WLOOP:         LD      A,(HL)
0268   32 03 40                LD      (LROWBIT),A
026B   23                      INC     HL
026C   7E                      LD      A,(HL)
026D   32 04 40                LD      (HROWBIT),A
```

```
0272   3A 03 40          LD      A,(ROWBIT)
0275   D3 80             OUT     (ROWOUT),A
0277   23                INC     HL
0278   3A 02 40          LD      A,(COLBIT)
027B   CB 27             SLA     A
027D   32 02 40          LD      (COLBIT),A
0280   D3 60             OUT     (COLOUT),A
0282   10 E3             DJNZ    WLOOP
0284   C1                POP     BC
0285   E1                POP     HL
0286   F1                POP     AF
0287   C9                RET
                ;*****************************************************
                ;SUBROUTINE TO READ DATA FROM KEYBOARD       ******
                ;*****************************************************
0288   F5       READKEY: PUSH    AF
0289   DB 02    READSTA: IN      A,(KBDPPI+2)
028B   CB 7F             BIT     7,A
028D   CA 93 02          JP      Z,READY
0290   C3 89 02          JP      READSTA
0293   3E 01    READY:   LD      A,01H
0295   D3 02             OUT     (KBDPPI+2),A   ;BREAKE KBD CLOCK
0297   DB 00             IN      A,(KBDPPI+0)   ;READ KBD DATA
0299   32 01 40          LD      (SCANCODE),A   ;SAVE SCAN CODE
029C   3E 02             LD      A,02H          ;CLEAR KEYBOARD
029E   D3 02             OUT     (KBDPPI+2),A
02A0   3A 01 40          LD      A,(SCANCODE)
02A3   D3 20             OUT     (LATCHSCAN),A  ;LATCH SCAN CODE
02A5   DB 40             IN      A,(LATCHASCII) ;READ ASCII CODE FROM KEY GEN.
02A7   32 00 40          LD      (ASCIICODE),A
02AA   F1                POP     AF
02AB   C9                RET
                ;*****************************************************
                ;SUB  DISPLAY DATA BUFFER FULL               ******
                ;*****************************************************
02AC   F5       BUFFULL: PUSH    AF             ;DISPLAY LED WARNNING
02AD   3E FF             LD      A,0FFH
02AF   32 04 40          LD      (HROWBIT),A
02B2   D3 80             OUT     (ROWOUT),A
02B4   D3 81             OUT     (ROWOUT+1),A
02B6   D3 60             OUT     (COLOUT),A
02B8   F1                POP     AF
02B9   C9                RET


                ;*****************************************************
```

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
4000                    ORG     RAMAREA

4000            ASCIICODE:      DEFS    1
4001            SCANCODE:       DEFS    1
4002            COLBIT:         DEFS    1
4003            LROWBIT:        DEFS    1
4004            HROWBIT:        DEFS    1
4005            KEYBUFF:        DEFS    2
4007            KEYCNT:         DEFS    2
4009            COLCNT:         DEFS    1
400A            CTLFLG:         DEFS    1
400B            CAPFLG:         DEFS    1
400C            CPFLG:          DEFS    1
400D            CSFLG:          DEFS    1
400E            DATABUFF:       DEFS    1000

43F6                            END
```

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ASCIICODE    4000    BUFFULL      02AC    CAP        = 0070    CAPFLG       400B
CECODE     = 0005    CHARGENARE=  2000    COLBIT       4002    COLCNT       4009
COLOUT     = 0060    CPFLG        400C    CSFLG        400D    CTLFLG       400A
CTRL       = 0076    DATABUFF     400E    DISPLAY      0229    DNARROW    = 0074
DNSHIFT      01E9    DO_DISP      0233    ENTER      = 000D    ESC        = 001B
ETBL       = 2000    F1         = 0075    HROWBIT      4004    KBDPPI     = 0000
KEYBUFF      4005    KEYCNT       4007    LARROW     = 0071    LATCHASCII = 0040
LATCHSCAN  = 0020    LOOP         024B    LROWBIT      4003    MAIN         0000
MAINDO       001D    NBUFFULL     004E    NOCBIT       01C7    NOCTRL       008D
NOTCBIT      01FA    NOT_C_P      0168    NOT_FUNC     019D    NO_C_P       012B
NO_FUNC      0102    OTHER1       0062    OTHER10      0184    OTHER11      01A7
OTHER12      01DA    OTHER13      020D    OTHER14      021D    OTHER2       0072
OTHER3       0095    OTHER4       00AD    OTHER5       00BE    OTHER6       00CF
OTHER7       00E9    OTHER8       010C    OTHER9       0147    OUTCHAR      0246
OUTWORD      0262    PPIMODE    = 0080    RAMAREA    = 4000    RARROW     = 0072
READDATA     002C    READKEY      0288    READSTA      0289    READY        0293
ROMAREA    = 0000    ROWOUT     = 0080    RSHIFT       0156    SCANCODE     4001
SHIFT        011B    SHIFTL       0132    SHIFTRR      016F    SYSSTACK   = 47FF
TTBL       = 2800    UPARROW    = 0073    USHIFT       01B6    WLOOP        0267


0000 ASSEMBLY ERRORS
```

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

ในบทนี้จะแสดงส่วนต่าง ๆ ของวงจรและ PRINTED CIRCUIT BOARD ซึ่งอุปกรณ์ที่
สามารถหาซื้อได้ในท้องตลาดทั่วไป ส่วนการทำงานของภาคต่าง ๆ ของอุปกรณ์แต่ละภาคได้อธิบาย
ที่ผ่านมาแล้ว

ในบทนี้จะอธิบายวิธีการใช้งาน และคำสั่งควบคุมการทำงานต่าง ๆ ของ PROGRAMM
SCORE BOARD DISPLAY

นอกจากนี้แล้วจะแสดงส่วนต่าง ๆ ของวงจรและ PRINTED CIRCUIT BOARD ซึ่งอุ
ที่ใช้นี้สามารถหาซื้อได้ในท้องตลาดทั่วไป ส่วนการทำงานของภาคต่าง ๆ ได้อธิบายในบทที่ผ่านม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการใช้งาน (OPERATION MANUAL)

PROGRAMMABLE SCORE BOARD DISPLAY เป็นวงจรไฟวิ่งแบบโปรแกรมเองได้ โดยรับข้อมูลที่ต้องการทางคีย์บอร์ด ซึ่งใช้ KEYBOARD ของเครื่อง IBM/PC XT COMPATIBLE แล้วนำข้อมูลที่ป้อนเข้ามานี้ นำออกมาแสดงผลทาง LED DISPLAY ขนาด DOT 16 x 95 โดย สามารถแสดงการวิ่งของตัวอักษรได้หลายรูปแบบ ตาม PATTERN ที่กำหนดไว้ เช่น จะให้วิ่งไป ทางซ้าย, ขวา ดึงข้อความไปทางขวาและกลับมาสู่ข้อความเดิมอีกครั้ง (PULL RIGHT) เป็นต้น เมื่อเริ่มต้นการทำงาน โปรแกรม MONITOR จะสั่งงานให้ CPU ทำงาน DISPLAY ข้อความที่ ป้อนเก็บไว้จาก KEYBOARD ซึ่งเก็บไว้ใน RAM ออกมาแสดงตาม PATTERN การวิ่งต่าง ๆ กัน จนจบแล้วก็จะวนกลับไปแสดงตอนเริ่มต้นอีกครั้ง วนเช่นนี้ไปเรื่อย ๆ

วิธีการโปรแกรม สามารถทำได้โดยการกดคีย์ "ENTER" ตัวไฟวิ่งก็จะหยุดวิ่ง โปรแกรมจะถูก RUN ที่ EDITOR PROGRAM เพื่อทำการรับข้อมูลทาง KEYBOARD แล้วนำแสดง ออกให้เห็นทาง DISPLAY LED และเก็บข้อมูลใน RAM อีกด้วย ในโหมดนี้จะสามารถแก้ไข ข้อมูลได้ โดยเลื่อน ตำแหน่งตัวอักษรไปทางซ้าย หรือทางขวา ทำการลบข้อความตัวถัดไปก่อน หน้านี้ เมื่อทำการป้อนข้อมูลให้เรียบร้อยแบบการวิ่ง DISPLAY PATTERN ก่อน แล้วตามด้วย ข้อมูลตามที่ต้องการให้แสดงออกมาโดยจำนวนข้อความวิ่งจรทั้งรูปแบบการวิ่ง รวมทั้งหมดจะต้อง ไม่เกิน 1,000 ตัวอักษร

คำสั่งต่าง ๆ ที่ใช้การควบคุมทางคีย์บอร์ด มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้อักษรไฟว์งคอมพิวเตอร์

เมื่อต้องการเริ่มต้นข้อมูล ให้กด "ENTER" เสมอและเครื่องหมายที่ปรากฏบนจอภา

เมื่อต้องการป้อนข้อมูลจะเป็น T เสมอ ซึ่งหมายถึงภาษาไทย ฉะนั้นหากท่านต้องการภาษาอังกฤ

ท่านต้องกดคำสั่ง CTRL + E ก่อนจึงจะสามารถคีย์ข้อมูลที่ท่านต้องการได้ทันที การป้อนข้อมูลต้อ

กด CAPS LOCK ก่อนเสมอ (ให้ไฟสีเขียวติด) เฉพาะภาษาไทย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รหัสคำสั่ง

←---- ใช้สำหรับเลื่อนข้อความครั้งละ 1 ตัวอักษร มาข้างหน้าข้อความจะปรากฏ
จากขอบด้านขวามือ

----→ ใช้สำหรับเลื่อนข้อความรหัสคำสั่งถอยหลัง ข้อความจะถอยครั้งละ 1 ตัวอักษ

ESC - ใช้ลบตัวอักษร, รหัสคำสั่งที่อยู่ทางขวามือ ในขณะเดียวกันจะเลื่อนถอย
หลัง 1 ครั้ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปนี้เป็นคำสั่งพิเศษ ซึ่งท่านต้องกด NUM LOCK (ให้ไฟเขียวติดเสมอ) จึงกดคำสั่งต่อไปนี้

1. CTRL X P + ←--- ดึงข้อความไปทางซ้ายและกลับมาสู่จอด้วยข้อความเดิมอีกครั้งหนึ่ง

(เหมาะสำหรับเน้นข้อความสำคัญ) (PULL LEFT)

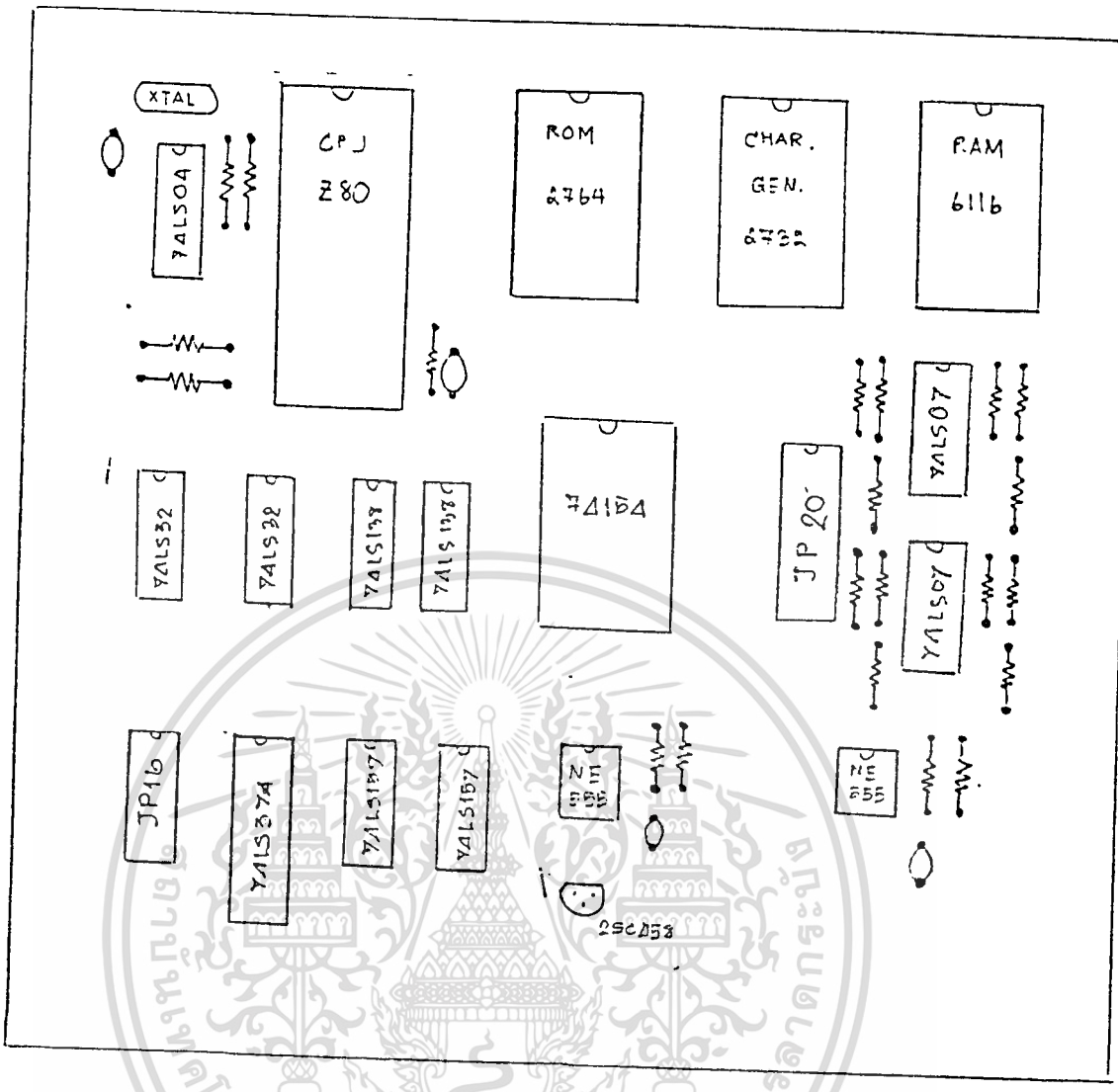
2. CTRL X P + ---→ ดึงข้อความไปทางขวาและกลับมาสู่ข้อความเดิมอีกครั้งหนึ่ง

(PULL RIGHT)

3. CTRL X S + เลื่อนข้อความขึ้น (SCROLL UP)

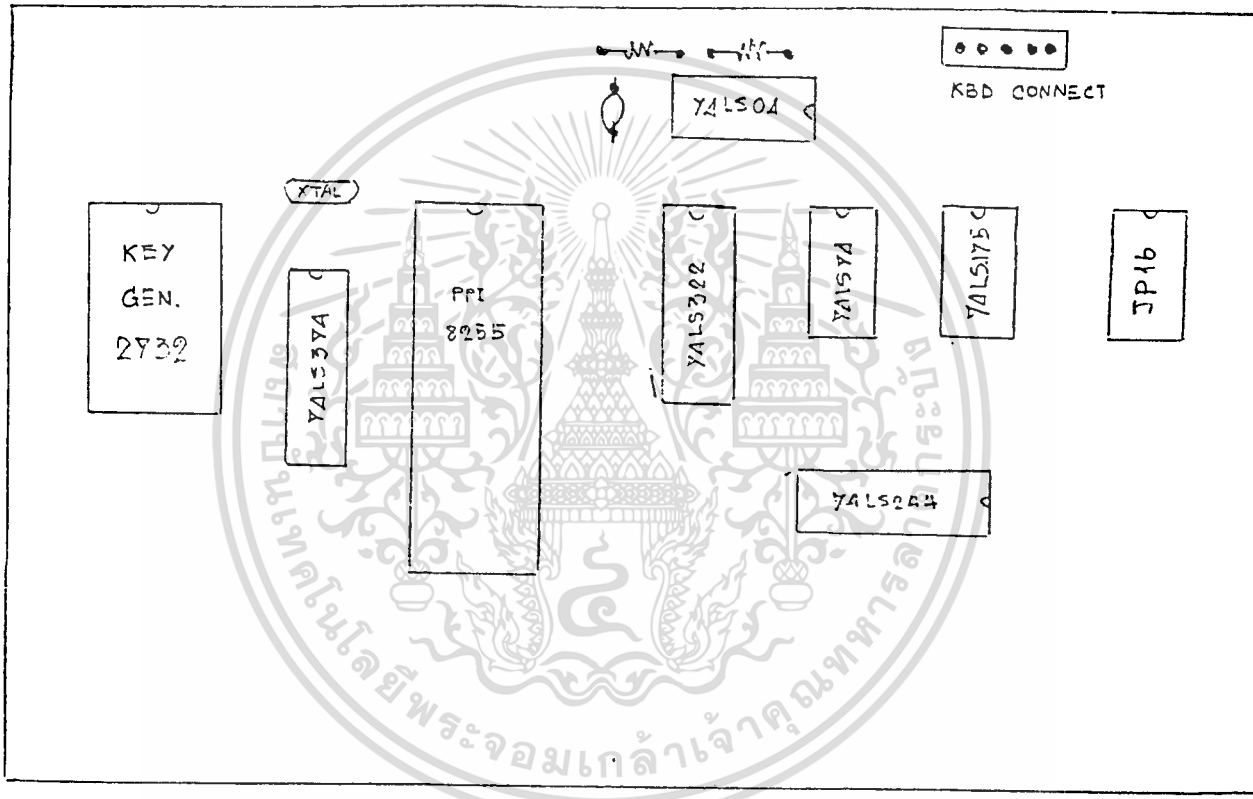
4. CTRL X S + เลื่อนข้อความลง (SCROLL DOWN)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



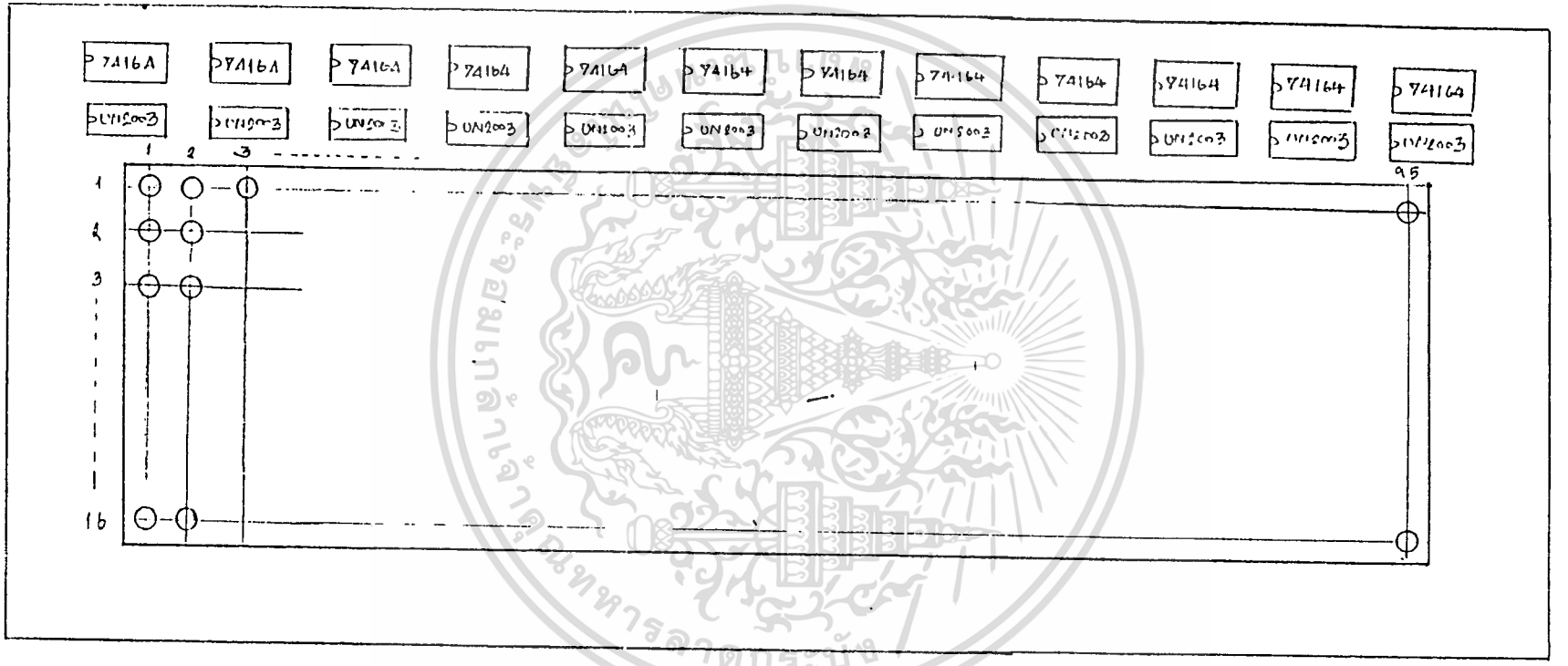
CONTROLLER BOARD..

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

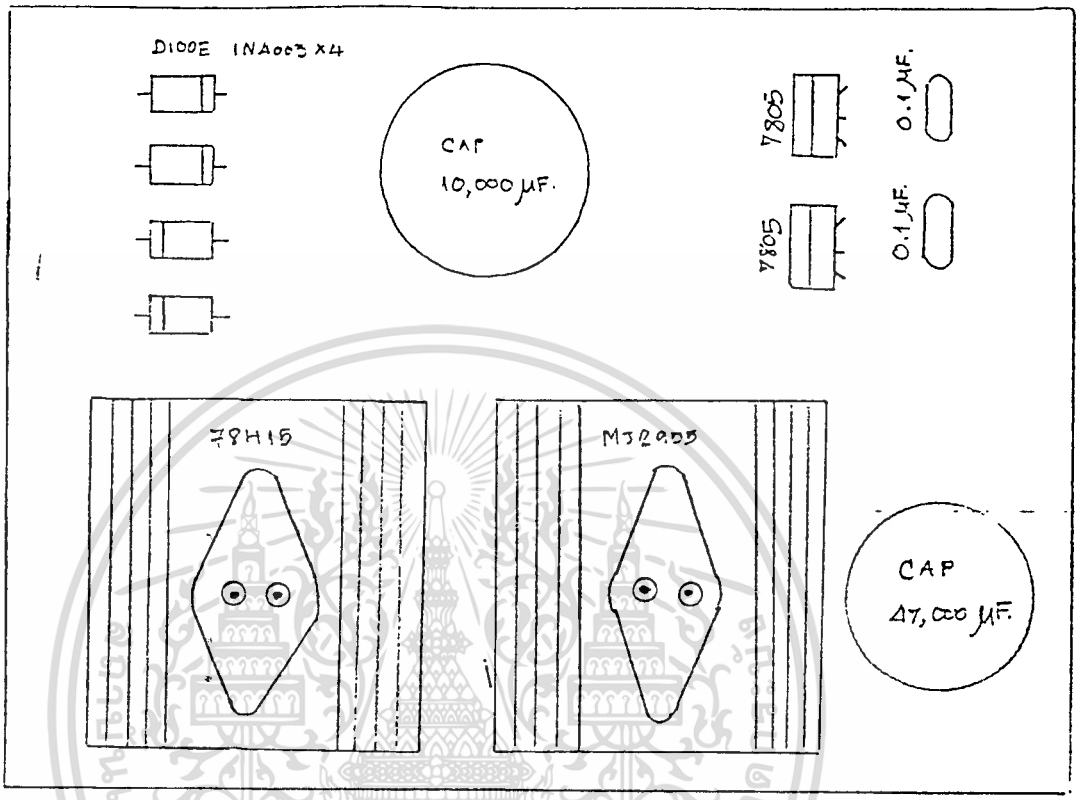


KEYBOARD INTERFAC BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



LED DISPLAY BOARD



POWER SUPPLY BOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทสรุป

ในการออกแบบและสร้าง PROGRAMMABLE SCORE BOARD DISPLAY นี้ ผู้ทำโครง
ได้มีความรู้และความเข้าใจในการพัฒนาและใช้งานไมโครโปรเซสเซอร์ ให้มีขีดความสามารถสนอ
ความต้องการในการใช้งาน ดังนั้น การศึกษา ค้นคว้า เรียนรู้และการออกแบบอุปกรณ์ให้ทำงานได้
ประสิทธิภาพตามที่เราต้องการนั้น จำเป็นต้องมพื้นความรู้ทาง HARDWARE และ SOFTWARE ประก
การศึกษาและออกแบบป้ายแสดงอักษรวิ่งแบบโปรแกรมได้นี้ เป็นส่วนในการเริ่มให้รู้จักการประยุกต์
งาน MICROPROCESSOR รวมทั้งการออกแบบโปรแกรมการใช้งานด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. "PROGRAMMING THE Z80", RODNAY ZAKS, SYBEX INC. USA, 1980

2. "Z80" ASSEMBLY LANGUAGE PROGRAMMING", LANCE A. LEVENTHAL

3. "Z80" APPLICATIONS", JAMES W. COFFRON, SYBEX INC. USA, 1983

5. "ไมโครโปรเซสเซอร์ ไมโครคอมพิวเตอร์ (Z-80 MICROPROCESSOR)
ยืน ภู่วรวรรณ และ วัฒนา เชี่ยงกุล, 2528

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

54154/74154 4-Line-to-16-Line Decoder/Demultiplexer

	Schottky TTL			High-Speed TTL			Low-Power Schottky TTL			Standard TTL			Low-Po
	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Device Type
T.I.																
FAIRCHILD																
MOTOROLA																
N.S.C.																
PHILIPS																
SIGNETICS																
SIEMENS																
FUJITSU																
HITACHI																
MITSUBISHI																
NEC																
TOSHIBA																

Electrical Characteristics SN54154/SN74154

absolute maximum ratings over operating free-air temperature range

Supply voltage, V_{CC}	7V	Operating free-air temperature range	SN54*	−55°C to 125°C
Input voltage	5.5V		SN74*	0°C to 70°C
		Storage temperature range		−55°C to 150°C

recommended operating conditions

	SN54154 MIN	NOM	MAX	SN74154 MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			−400			−400	μA
Low-level output current, I_{OL}			16			16	mA
Operating free-air temperature, T_A	−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range

PARAMETER†		TEST CONDITIONS†		MIN	TYP‡	MAX	UNIT
V_{IH}	High-level input voltage			2			V
V_{IL}	Low-level input voltage					0.8	V
V_I	Input clamp voltage	V_{CC}=MIN, I_I=−12mA				−1.5	V
V_{OH}	High-level output voltage	V_{CC}=MIN, V_{IH}=2V, V_{IL}=0.8V, I_{OH}=−800μA		2.4	3.4		V
V_{OL}	Low-level output voltage	V_{CC}=MIN, V_{IH}=2V, V_{IL}=0.8V, I_{OL}=16mA			0.2	0.4	V
I_I	Input current at maximum input voltage	V_{CC}=MAX, V_I=5.5V				1	mA
I_{IH}	High-level input current	V_{CC}=MAX, V_I=2.4V				40	μA
I_{IL}	Low-level input current	V_{CC}=MAX, V_I=0.4V				−1.6	mA
I_{OS}	Short-circuit output current*	V_{CC}=MAX,	SN54*	−20		−55	mA
			SN74*	−18		−57	
I_{CC}	Supply current	V_{CC}=MAX, See Note	SN54*		34	49	mA
			SN74*		34	56	
t_{PLH}	from A, B, C, or D inputs through 3 levels of logic	V_{CC}=5V, T_A=25°C, C_L=15pF, R_L=400Ω			24	36	ns
t_{PHL}	from A, B, C, or D inputs through 3 levels of logic				22	33	ns
t_{PLH}	from either strobe input				20	30	ns
t_{PHL}	from either strobe input				18	27	ns

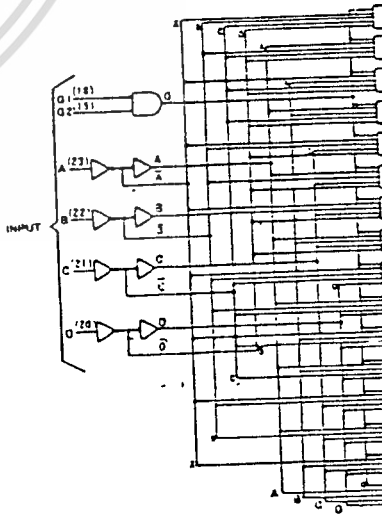
Function Table

'154, 'L154

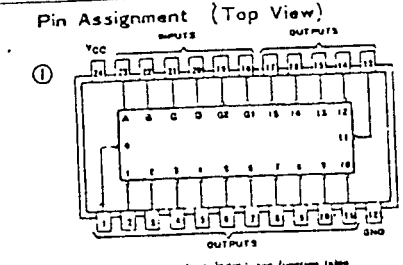
INPUTS G1	G2	D	C	B	A	OUTPUTS 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	L	L	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H
L	L	H	L	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H
L	L	H	H	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H
L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L
H	X	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
X	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

H = high level, L = low level, X = irrelevant

Functional Block Diagram



Pin Assignment (Top View)



positive logic: see function table

'154, 'L154 4-LINE-TO-16-LINE DECODER/DEM

NOTE: I_{CC} is measured with all inputs grounded and all outputs

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable type.
‡ All typical values are at V_{CC}=5V, T_A=25°C.
* Not more than one output should be shorted at a time.
t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output

เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า กรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(RESET)

Reset. A "high" on this input clears the control register and all ports (A, B, C) are set to the input mode.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 8255A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A – Port A and Port C upper (C7-C4)
Control Group B – Port B and Port C lower (C3-C0)

The Control Word Register can Only be written into. No Read operation of the Control Word Register is allowed.

Ports A, B, and C

The 8255A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255A.

Port A. One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B. One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.

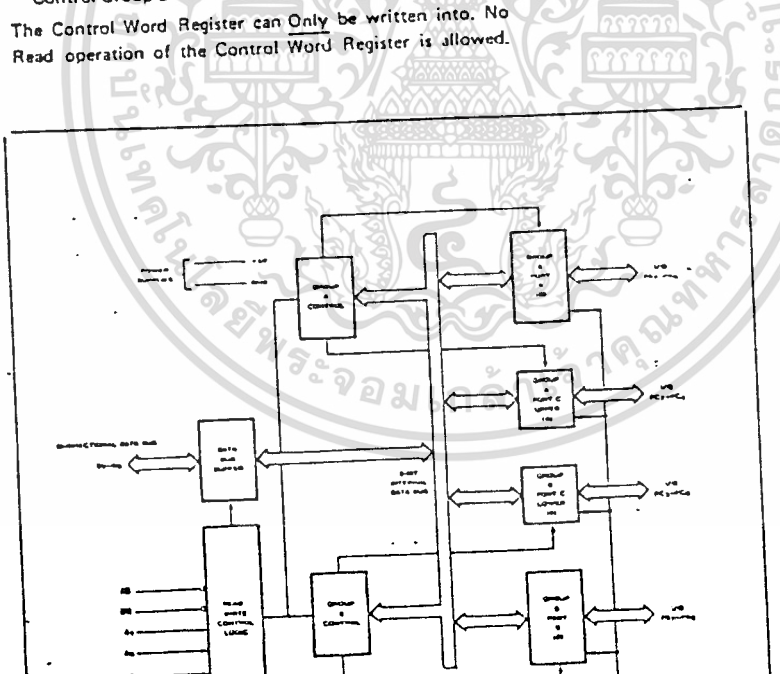
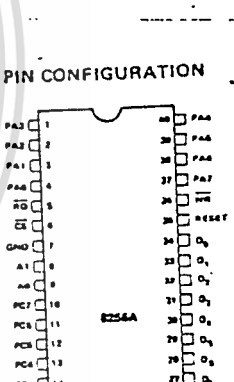


Figure 4. 8225A Block Diagram Showing Group A and Group B Control Functions

PIN CONFIGURATION



PIN NAMES

D_7-D_0	DATA BUS (BI DIRECTIONAL)
RESET	RESET INPUT
$\overline{CS}$	CHIP SELECT
$\overline{RD}$	READ INPUT
$\overline{WR}$	WRITE INPUT
A0, A1	PORT ADDRESS
PA7-PA0	PORT A (BIT)
PB7-PB0	PORT B (BIT)
PC7-PC0	PORT C (BIT)
V_{CC}	+5 VOLTS
GND	0 VOLTS

231308-6

นี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า กรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

54138/74138 3-Line-to-8-Line Decoder

	Schottky TTL		High-Speed TTL		Low-Power Schottky TTL		Standard TTL	
	Device Type	Package (C I P I MICF)	Device Type	Package (C I P I MICF)	Device Type	Package (C I P I MICF)	Device Type	Package (C I P I MICF)
T.I.	SN54S138	J W |			SN54LS138	J W |		
	SN74S138	J N |			SN74LS138	J D N |		
FAIRCHLD	[illegible]	J D |			[illegible]	J D |		
	[illegible]	J D |			[illegible]	J D |		
MOTOROLA	[illegible]				[illegible]			
N.S.C.					DM74LS138	J D |		
	DM74S138	J D |			DM54LS138	J D |		
PHILIPS								
	N74S138	F D |			N74LS138	F D |		
SIGNETICS	S54S138	F C B D | W |						
	N74S138	F C B |			N74LS138	F A D |		
SIEMENS								
FUJITSU					74LS138	M D |		
HITACH					HD74LS138	P D |		
MITSUBISHI	M74S138	P |			M74LS138	P |		
NEC					74LS138	C D |		
TOSHIBA								

Electrical Characteristics SN54LS138/SN74LS138

absolute maximum ratings over operating free-air temperature range

Supply voltage, V_{CC}	7V	Operating free-air temperature range	SN54LS	−55°C to 125°C
Input voltage	7V		SN74LS	0°C to 70°C
		Storage temperature range		−65°C to 150°C

recommended operating conditions

	SN54LS138			SN74LS138			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			−400			−400	μA
Low-level output current, I_{OL}			4			8	mA
Operating free-air temperature, T_A	−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range

PARAMETER*		TEST CONDITIONS†		MIN	TYP‡	MAX	UNIT
V_{IH}	High-level input voltage			2			V
V_{IL}	Low-level input voltage					0.8	V
V_I	Input clamp voltage	V_{CC} = MIN, I_I = −18mA				−1.5	V
V_{OH}	High-level output voltage	V_{CC} = MIN, V_{IH} = 2V, V_{IL} = 0.8V, I_{OH} = −400μA	SN54LS	2.5	3.4		V
			SN74LS	2.7	3.4		
V_{OL}	Low-level output voltage	V_{CC} = MIN, V_{IH} = 2V, V_{IL} = 0.8V, I_{OL} = 8mA			0.35	0.5	V
I_I	Input current at maximum input voltage	V_{CC} = MAX, V_I = 7V				0.1	mA
I_{IH}	High-level input current	V_{CC} = MAX, V_I = 2.7V				20	μA
I_{IL}	Low-level input current	V_{CC} = MAX, V_I = 0.4V				−0.4	mA
I_{OS}	Short-circuit output current*	V_{CC} = MAX		−20		−100	mA
I_{CC}	Supply current	V_{CC} = MAX, Outputs enabled and open			6.3	10	mA
t_{PLH}	from Binary select to Any output	V_{CC} = 5V, T_A = 25°C, C_L = 15pF, R_L = 2kΩ	2 levels of delay		13	20	ns
t_{PHL}					27	41	
t_{PLH}			3 levels of delay		18	27	ns
t_{PHL}					26	39	
t_{PLH}	from Enable to Any output		2 levels of delay		12	18	ns
t_{PHL}					21	32	
t_{PLH}			3 levels of delay		17	26	ns
t_{PHL}					25	38	

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable device type

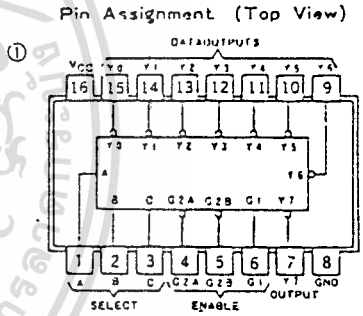
‡ All typical values are at V_{CC} = 5V, T_A = 25°C

* Not more than one output should be shorted at a time, and duration of the short-circuit test should not exceed one second.

t_{PLH} = propagation delay time, low-to-high-level output

t_{PHL} = propagation delay time, high-to-low-level output

Pin Assignment (Top View)



Function Table

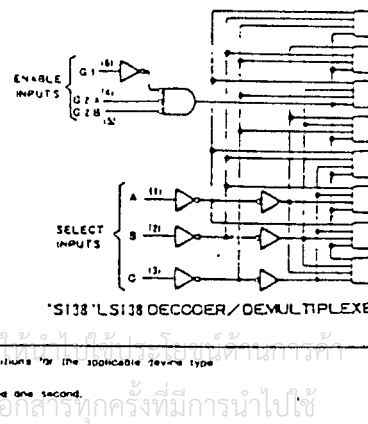
'S138, 'LS138

INPUTS					OUTPUTS							
ENABLE		SELECT										
G1	G2*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

* $\overline{G2} = \overline{G2A} + \overline{G2B}$.

H = high level, L = low level, X = irrelevant

Functional Block Diagram



'S138, 'LS138 DECODER/DEMULTIPLEXER

เอกสารนี้เป็นเอกสารที่ [illegible] เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ในกรณีใดๆทั้งสิ้น [illegible] และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

54322/74322 8-Bit Shift Registers with Sign Extend

	Schottky TTL				High-Speed TTL				Low-Power Schottky TTL				Standard TTL				Low-Power T			
	Device Type	Package			Device Type	Package			Device Type	Package			Device Type	Package			Device Type	Pac		
		C	F	M		C	F	M		C	F	M		C	F	M		C	F	
T. I.									SN54LS322											
									SN74LS322											
FAIRCHILD																				
MOTOROLA																				
N. S. C.																				
PHILIPS																				
SIGNETICS																				
SIEMENS																				
FUJITSU																				
HITACHI																				
MITSUBISHI																				
NEC																				
TOSHIBA																				

Electrical Characteristics SN54LS322/SN74LS322

absolute maximum ratings over operating free-air temperature range

Supply voltage, V_{CC}	7 V	Operating free-air temperature range	SN54LS	−55°C to	125°C
Input voltage	7 V		SN74LS	0°C to	70°C
		Storage temperature range		−65°C to	150°C

recommended operating conditions

		SN54LS322			SN74LS322			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}		4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}	Q_A thru Q_H			−1			−2.6	mA
	Q_H'			−0.4			−0.4	
Low-level output current, I_{OL}	Q_A thru Q_H			12			24	mA
	Q_H'			4			8	
Clock frequency, f_{clock}		0		35	0		35	MHz
Width of clock pulse, t_w(clock)	Clock high	18			14			ns
	Clock low	14			14			
Width of clear pulse, t_w(clear)	Clear low	20			20			ns
Setup time, t_{su}	Data select	10†			10†			ns
	High-level data ◇	20†			20†			
	Low-level data ◇	20†			20†			
	Clear inactive-state	20†			20†			
Hold time, t_h	Data select	10†			10†			ns
	Data ◇	0†			0†			
Operating free-air temperature, T_A		−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER			TEST CONDITIONS‡		SN74LS322 MIN	TYP§	MAX	UNIT
V_{IH}	High-level input voltage				2			V
V_{IL}	Low-level input voltage						0.8	V
V_{IK}	Input clamp voltage		V_{CC} = MIN,	I_I = −18 mA			−1.5	V
V_{OH}	High-level output voltage	Q_A thru Q_H	V_{CC} = MIN, V_{IH} = 2V, V_{IL} = V_{IL} max,	I_{OH} = MAX	2.4	3.1		V
		Q_H'			2.7	3.4		
V_{OL}	Low-level output voltage	Q_A thru Q_H	V_{CC} = MIN, V_{IH} = 2V, V_{IL} = V_{IL} max,	I_{OL} = 24 mA		0.35	0.5	V
		Q_H		I_{OL} = 4 mA		0.25	0.4	
I_{OZH}	Off-state output current, high-level voltage applied	Q_A thru Q_H	V_{CC} = MAX, V_O = 2.7V	V_{IH} = 2V			40	μA
I_{OZL}	Off-state output current, low-level voltage applied	Q_A thru Q_H	V_{CC} = MAX, V_O = 0.4V	V_{IH} = 2V			−400	μA
I_I	Input current at maximum input voltage	A thru H	V_{CC} = MAX	V_I = 5.5V			0.1	mA
		Data select		V_I = 7V			0.2	
		Sign extend		V_I = 7V			0.3	
		Any other		V_I = 7V			0.1	
I_{IH}	High-level input current	A thru H, OS	V_{CC} = MAX,	V_I = 2.7V			40	μA
		Sign extend					60	
		Any other					20	
I_{IL}	Low-level input current	Data select	V_{CC} = MAX,	V_I = 0.4V			−0.8	mA
		Sign extend					−1.2	
		Any other					−0.4	
I_{OS}	Short-circuit output current ¶	Q_A thru Q_H	V_{CC} = MAX		−30		−130	mA
		Q_H'			−20		−100	
I_{CC}	Supply current		V_{CC} = MAX			35	60	mA

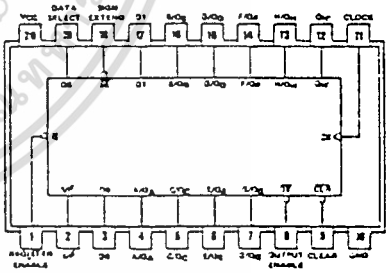
◇Data includes the two serial inputs and the eight input/output data lines.
†The arrow indicates that the rising edge of the clock pulse is used for reference.
‡For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
§All typical values are at V_{CC} = 5V, T_A = 25°C.
¶Not more than one output should be shorted at a time and duration of the short-circuit should not exceed one second.

switching characteristics, V_{CC} = 5V, T_A = 25°C

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MA
f_{max}			See Note 2	25	50	
t_{PLH}	Clock	Q_H'	C_L = 15pF, R_L = 2kΩ, See Note 2		22	
t_{PHL}					26	
t_{PHL}	Clear	Q_H'			27	
t_{PLH}	Clock	Q_A thru Q_H	C_L = 45pF, R_L = 667Ω, See Note 2		16	
t_{PHL}					22	
t_{PHL}	Clear	Q_A thru Q_H			22	
t_{PZH}	Output enable	Q_A thru Q_H			15	
t_{PZL}					15	
t_{PHZ}	Output enable	Q_A thru Q_H	C_L = 5pF, R_L = 667Ω, See Note 2		15	
t_{PLZ}					15	

Pin Assignments (Top View)

SN54LS322 . . . . J PACKAGE
SN74LS322 . . . . J OR N PACKAGE



FUNCTION TABLE

OPERATION	INPUTS								INPUTS/OUTPUTS				OUTPUTS
	CLEAR	CLOCK†	SR	SL	DATA SELECT	OUTPUT ENABLE	G1	G2	A/Q_A	B/Q_B	C/Q_C	D/Q_D	Q_H
Clear	L	X	X	X	X	X	X	X	L	L	L	L	L
Hold	H	L	X	X	X	X	X	X	Q_{A0}	Q_{B0}	Q_{C0}	Q_{D0}	Q_{H0}
Shift Right	H	↑	H	X	L	L	X	X	H	Q_{An}	Q_{Bn}	Q_{Cn}	Q_{Dn}
	H	↑	L	X	L	L	X	X	L	Q_{An}	Q_{Bn}	Q_{Cn}	Q_{Dn}
Shift Left	H	↑	X	H	H	L	X	X	Q_{Bn}	Q_{Cn}	Q_{Dn}	H	Q_{Hn}
	H	↑	X	L	H	L	X	X	Q_{Bn}	Q_{Cn}	Q_{Dn}	L	Q_{Hn}
Load Register	H	↑	X	X	H	L	L	L	a	b	c	d	h
Output Disable	H	↑	X	X	X	H	X	X	Z	Z	Z	Z	Z

H = high level (steady steady state)
L = low level (steady state)
X = irrelevant (any input, including transitions)
↑ = transition from low to high level
Q_{A0} ... Q_{H0} = the level of Q_A through Q_H, respectively, before the indicated steady-state conditions were established
Q_{An} ... Q_{Hn} = the level of Q_A through Q_H, respectively, before the most recent ↑ transition of the clock
G0, G1 = the level of steady-state inputs at inputs G0 and G1 respectively
a ... h = the level of steady-state inputs at inputs A through H respectively

f_{max} = maximum clock frequency
t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output
t_{PZH} = output enable time to high level
t_{PZL} = output enable time to low level
t_{PHZ} = output disable time from high level
t_{PLZ} = output disable time from low level

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

54374/74374 Octal D-Type Transparent Latches and Edge-Triggered Flip-Flops

	Schottky TTL		High-Speed TTL		Low-Power Schottky TTL		Standard TTL		Low-Power TTL	
	Device Type	Package	Device Type	Package	Device Type	Package	Device Type	Package	Device Type	Package
		C P M ICF		C P M ICF		C P M ICF		C P M ICF		C P M ICF
T.I.	SN54S374 J SN74S374 J, N				SN54LS374 J SN74LS374 J, N					
FAIRCHILD										
MOTOROLA										
N.S.C.										
PHILIPS										
SIGNETICS										
SIEMENS										
FUJITSU										
HITACHI										
MITSUBISHI										
NEC										
TOSHIBA										

Electrical Characteristics SN54LS374/SN74LS374

absolute maximum ratings over operating free-air temperature range

Supply voltage, V_{CC}	7V	Operating free-air temperature range	SN54LS	−55°C to 125°C
Input voltage	7V		SN74LS	0°C to 70°C
		Storage temperature range		−65°C to 150°C

recommend operating conditions

		SN54LS374			SN74LS374			UNIT
		MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage V_{CC}		4.5	5	5.5	4.75	5	5.25	V
High-level output voltage, V_{OH}				−420			−400	μA
High-level output current, I_{OH}				16			16	mA
Width of clock enabling pulse, t_w	High	15			15			ns
	Low	15			15			
Data hold time, t_{hold}		0†			0†			ns
Setup time, t_{setup}		20†			20†			ns
Operating free-air temperature, T_A		−55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range (unless otherwise noted)

PARAMETER		TEST CONDITIONS‡		MIN	TYP‡	MAX	UNIT
V_{IH}	High-level input voltage			2			V
V_{IL}	Low-level input voltage					0.8	V
V_{IK}	Input clamp voltage	V_{CC} = MIN, I_I = −18mA				−1.5	V
V_{OH}	High-level output voltage	V_{CC} = MIN, V_{IH} = 2V, V_{IL} = V_{IL} max, I_{OH} = MAX		2.4	3.1		V
V_{OL}	Low-level output voltage	V_{CC} = MIN, V_{IH} = 2V, V_{IL} = V_{IL} max	I_{OL} = 24mA		0.35	0.5	V
I_{OZH}	Off-state output current, high-level voltage applied	V_{CC} = MAX, V_{IH} = 2V, V_O = 2.7V				20	μA
I_{OZL}	Off-state output current, low-level voltage applied	V_{CC} = MAX, V_{IH} = 2V, V_O = 0.4V				−20	μA
I_I	Input current at maximum input voltage	V_{CC} = MAX, V_I = 7V				0.1	mA
I_{IH}	High-level input current	V_{CC} = MAX, V_I = 2.7V				20	μA
I_{IL}	Low-level input current	V_{CC} = MAX, V_I = 0.4V				−0.1	mA
I_{OS}	Short-circuit output current §	V_{CC} = MAX		−30		−130	mA
I_{CC}	Supply current	V_{CC} = MAX, Output control at 4.5V	LS374		27	40	mA

switching characteristics, V_{CC} = 5V, T_A = 25°C

PARAMETER	FROM (INPUT)	TO (OUTPUT)	TEST CONDITIONS	MIN	TYP	MAX	UNIT
f_{max}			C_L = 45pF, R_L = 667Ω, See Notes 2 and 3	35	50		MHz
t_{PLH}	Data	Any Q					
t_{PHL}					15	23	ns
t_{PLH}	Clock or enable	Any Q			19	29	
t_{PHL}					20	28	ns
t_{PZH}	Output Control	Any Q			21	28	
t_{PZL}					12	20	ns
t_{PHZ}	Output Control	Any Q	C_L = 5pF, R_L = 667Ω, See Note 3		14	25	
t_{PLZ}							ns

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at V_{CC} = 5V, T_A = 25°C

§ Not more than one output should be shorted at a time, and duration of the short circuit should not exceed one second.

NOTES 2. Maximum clock frequency is tested with all outputs loaded.

3. See load circuits and waveforms on page 3-11.

f_{max} = maximum clock frequency

t_{PLH} = propagation delay time, low-to-high-level output

t_{PHL} = propagation delay time, high-to-low-level output

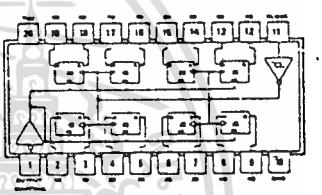
t_{PZH} = output enable time to high level

t_{PZL} = output enable time to low level

t_{PHZ} = output disable time from high

t_{PLZ} = output disable time from low

Pin Assignment (Top View)



SN54LS374 (J) SN74LS374 (J, N)
SN54S374 (J) SN74S374 (J, N)

'LS374, 'S374
FUNCTION TABLE

OUTPUT CONTROL	CLOCK	D	OUTPUT
L	↑	H	H
L	↑	L	L
L	L	X	Q_0
H	X	X	Z

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้