

ปริญญานิพนธ์
ชุดฝึกอบรมปฏิบัติการไมโครโปรเซสเซอร์
MICROPROCESSOR OPERATION TRAINING SYSTEM

1. นายเฉลิมพันธ์ ขสสมบัติ
2. นายนิวัฒน์ วีระกุล
3. นายสมพร ศรีวัฒนพล
4. นายสุเมศร์ ศรีคำพา



A021065

เลขหมู่.....	1297
เลขทะเบียน.....	-3 WFI 2538
วัน เดือน ปี.....	

021045

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
ภาควิชาครุศาสตร์วิศวกรรม
คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



มหาวิทยาลัยราชภัฏนครพนม
คณะวิทยาศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ใบรับรองปริญญานิพนธ์

หัวข้อปริญญานิพนธ์ ชุดฝึกอบรมปฏิบัติการไมโครโปรเซสเซอร์

MICROPROCESSOR OPERATION TRAINING SYSTEM

ชื่อนักศึกษา 1. นายเฉลิมพันธ์ ยศสมบัติ รหัสประจำตัว 36031110
2. นายนิวัฒน์ วีระกุล รหัสประจำตัว 36031120
3. นายสมพร ศรีวัฒนพล รหัสประจำตัว 36031135
4. นายสุเมษฐ์ ศรีคำพา รหัสประจำตัว 36031140

หลักสูตร วิศวกรรมศาสตรบัณฑิต สาขาวิชา วิศวกรรมโทรคมนาคม

อาจารย์ผู้ควบคุมปริญญานิพนธ์

1. อาจารย์สุชิน อางหาญ
2. อาจารย์ปิยะ จิตธรรมมาภิรมย์
3. อาจารย์โกศล ตราชู

คณะกรรมการสอบปริญญานิพนธ์	ลายมือชื่อ
1. อาจารย์สุชิน อางหาญ	
2. อาจารย์ปิยะ จิตธรรมมาภิรมย์	
3. อาจารย์โกศล ตราชู	
4. อาจารย์วิสูตร อธิพรธรรม	
5. อาจารย์พีระวุฒิ สุวรรณจันทร์	

วัน/เดือน/ปี ที่สอบ 22 มีนาคม 2538 เวลา 13.00 น.ถึงเวลา 14.00 น.

สถานที่สอบ ห้อง ค.301 คณะวิทยาศาสตร์อุตสาหกรรม

ภาควิทยารับรองแล้ว
ลงนาม.....
(ผศ.ดร.ธีระพล หาดหลุมสินธุ์ อธิการบดี)
หัวหน้าภาควิชาวิศวกรรมโทรคมนาคม
วันที่ 10 เดือน มีนาคม 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง ชุดฝึกอบรมปฏิบัติการไมโครโปรเซสเซอร์

MICROPROCESSOR OPERATION TRAINING SYSTEM

ผู้จัดทำ

1. นายเฉลิมพันธ์ ยศสมบัติ
2. นายนิวัฒน์ วีระกุล
3. นายสมพร ศรีวัฒนพล
4. นายสุเมศร์ ศรีคำพา

อาจารย์ที่ปรึกษา

ลงนาม.....
(อาจารย์สุชิน อาจหาญ)

ลงนาม.....
(อาจารย์ปิยะ จิตธรรมมาภิรมย์)

ลงนาม.....
(อาจารย์โกศล ตราชู)

หัวหน้าภาควิชาครุศาสตร์วิศวกรรม

ลงนาม.....
(ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง ชุดฝึกอบรมปฏิบัติการไมโครโปรเซสเซอร์

MICROPROCESSOR OPERATION TRAINING SYSTEM

จุดประสงค์

1. เพื่อศึกษาโครงสร้างและระบบการทำงานของไมโครโปรเซสเซอร์ เบอร์ Z80
2. เพื่อศึกษาหลักการเขียนโปรแกรมคำสั่งการทำงานของไมโครโปรเซสเซอร์ เบอร์ Z80
3. เพื่อออกแบบชุดฝึกอบรมปฏิบัติการไมโครโปรเซสเซอร์
4. เพื่อสร้างชุดฝึกอบรมปฏิบัติการไมโครโปรเซสเซอร์
5. เพื่อนำชุดฝึกอบรมปฏิบัติการไมโครโปรเซสเซอร์เป็นสื่อประกอบการเรียนการสอนได้

ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจถึงระบบโครงสร้างการทำงานของไมโครโปรเซสเซอร์
2. สามารถอธิบายถึงหลักการเขียนโปรแกรมคำสั่งการทำงาน
3. สามารถสร้างชุดฝึกอบรมปฏิบัติการไมโครโปรเซสเซอร์
4. สามารถนำชุดฝึกอบรมปฏิบัติการไมโครโปรเซสเซอร์ไปประกอบการเรียนการสอน

ชุดฝึกระบบปฏิบัติการไมโครโปรเซสเซอร์

นายเฉลิมพันธ์ ยศสมบัติ

นายนิวัฒน์ วีระกุล

นายสมพร ศรีวัฒนพล

นายสุเมศร์ ศรีคำพา

อาจารย์ที่ปรึกษา

อาจารย์สุชิน อาจหาญ

อาจารย์ปิยะ จิตธรรมมาภิรมย์

อาจารย์โกศล ตราชู

ปีการศึกษา 2537

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้สร้างขึ้นเพื่อเป็นชุดฝึกระบบปฏิบัติการ ไมโครโปรเซสเซอร์ซึ่งมีการทำงานเหมือนกับชุดทดลองซิงเกิลบอร์ด (Single Board) แต่ปริญญานิพนธ์นี้จะแสดงกระบวนการทำงานต่างๆที่ละเอียดกว่าชุดทดลองซิงเกิลบอร์ด และใช้ในการศึกษาขั้นเบื้องต้นของการเรียนวิชาไมโครโปรเซสเซอร์สำหรับการออกแบบจะเน้นให้อธิบายโครงสร้างการทำงานของไมโครโปรเซสเซอร์เบอร์ Z80 ขนาด 8 บิตเป็นหลักซึ่งสามารถนำไปเป็นเครื่องต้นแบบของชุดฝึกในการเรียนการสอนวิชาไมโครโปรเซสเซอร์ได้

MICROPROCESSOR OPERATION TRAINING SYSTEM

MR.CHALERM PUN YOTSOMBUT

MR.NIWAT VEERAGUL

MR.SOMPORN SEEWATTANAPON

MR.SUMETH SRIKHAMPHA

ADVISOR

MR.SUCHIN

ARDHAN

MR.PIYA

JITTROMMAPIROM

MR.KOSON

TRACHU

1994

ABSTRACT

This thesis was create for training the operation system of micorprocessor which all of the process as the single board training but this thesis is show the process more find than the single board training to use for educate of beginner in subject of microprocessor. Design is to explain the structure and process of 8 bit z80 microprocessor which can use as the model of the training in the educate of microprocessor subject...

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สามารถสำเร็จตามกำหนดเวลาได้นั้นก็เพราะความสามัคคีผู้ร่วมงานในกลุ่มเป็นหลักและขอขอบคุณอาจารย์ สุชิน อาจหาญ และ อาจารย์ ปิยะ จิตธรรมมาภิรมย์ ที่ได้กรุณาให้คำปรึกษาทางด้านวิชาการตลอดจนเครื่องมือ อุปกรณ์และสถานที่ทางกลุ่มผู้จัดทำจึงขอขอบคุณมา ณ.ที่นี้

นอกจากนี้ขอขอบคุณเพื่อนๆที่ให้คำปรึกษาช่วยเหลือให้มีความสุขให้กำลังใจและชี้แนวทางแก้ไขปัญหาต่างๆที่เกิดขึ้น ขอขอบคุณสโมสรนักศึกษาครุศาสตร์อุตสาหกรรมที่ให้ความเอื้อเฟื้อทางด้านสถานที่และเครื่องอำนวยความสะดวกจึงทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จได้ด้วยดี



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เรื่อง	หน้า
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	2
2.1 ไมโครโปรเซสเซอร์	2
2.1.1 หลักการทำงานของไมโครโปรเซสเซอร์	2
2.1.2 โครงสร้างภายในและขบวนการทำงานของไมโครโปรเซสเซอร์	3
2.1.3 การใช้งานไมโครโปรเซสเซอร์	7
2.2 ระบบหน่วยความจำ	9
2.2.1 รอม	10
2.2.2 แรม	11
2.3 หน่วยอินพุตและเอาต์พุต	12
2.3.1 อินพุตเอาต์พุตพอร์ตแบบขนาน	13
2.3.2 อินพุตเอาต์พุตพอร์ตแบบอนุกรม	14
2.3.3 อินเตอร์รัพท์	15
2.3.4 การเข้าถึงหน่วยความจำโดยตรง	15
บทที่ 3 การออกแบบชุดฝึกระบบปฏิบัติการไมโครโปรเซสเซอร์	16
3.1 หลักการออกแบบ	19
3.1.1 หน่วยฝึกประมวลผลทางคณิตศาสตร์	19
3.1.2 หน่วยควบคุม	19
3.1.3 หน่วยพักข้อมูล	20
3.1.4 หน่วยความจำ	20
3.1.5 หน่วยติดต่ออุปกรณ์ภายนอกระบบ	20
3.1.6 หน่วยฝึกการป้อนข้อมูล	20
3.1.7 หน่วยบริการย่อย	20
3.2 โครงสร้างการทำงาน	20
3.2.1 ฝั่งการทำงานของเครื่องโดยรวม	21
3.2.2 ฝั่งการทำงานหน่วยบัสอินพุตเอาต์พุต	21

3.2.3 ฟังก์ชันการทำงานหน่วยประมวลผลทางคณิตศาสตร์	22
3.3 การออกแบบส่วนโปรแกรม	23
3.4 การออกแบบส่วนวงจร	24
3.5 การออกแบบวงจรส่วนแอลอีดี	28
3.6 การออกแบบชุดตัวเลขแสดงผล	29
3.7 การออกแบบเมทริกคีย์	29
3.8 การออกแบบส่วนติดต่อกับภายนอก	30
บทที่ 4 การทดลองและแสดงผลการทดลอง	31
4.1 การทดลองการทำงานของหน่วยการป้อนข้อมูลตามไทม์มิงไดอะแกรม	31
4.1.1 อุปกรณ์ส่วนต่างๆที่เกี่ยวข้องกับการทำงานของหน่วยป้อนข้อมูล	31
4.1.2 ตัวอย่างการทดลองในหน่วยการป้อนข้อมูล	31
4.2 การทดลองการทำงานของหน่วยประมวลผลคำสั่งทางคณิตศาสตร์	34
4.2.1 อุปกรณ์ที่เกี่ยวข้องกับการทำงานของหน่วยประมวลผลทางคณิตศาสตร์	40
4.2.2 ตัวอย่างการทดลองในหน่วยประมวลผลคำสั่งทางคณิตศาสตร์	40
4.3 การทดลองการทำงานในหน่วยประมวลผลกลาง	41
4.3.1 อุปกรณ์ที่เกี่ยวข้องกับการทำงานในหน่วยของตัวประมวลผลกลาง	45
4.3.2 ตัวอย่างการทดลองในหน่วยประมวลผลกลาง	45
บทที่ 5 บทวิจารณ์สรุปและแนวทางการพัฒนา	46
5.1 บทสรุป	52
5.1.1 หน่วยประมวลผลกลาง	52
5.1.2 หน่วยประมวลผลทางคณิตศาสตร์	53
5.1.3 หน่วยอ่านเขียนข้อมูลกับอุปกรณ์	53
5.2 ปัญหาและอุปสรรค	53
5.3 แนวทางการพัฒนา	53
ภาคผนวก ก.วงจรที่ใช้ในชุดฝึกอบรมปฏิบัติการไมโครโปรเซสเซอร์	55
ภาคผนวก ข.โปรแกรมในชุดฝึกอบรมปฏิบัติการไมโครโปรเซสเซอร์	65
บรรณานุกรม	

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 3.1 ผลการทดลองคำสั่ง LD A,CC	25
ตารางที่ 4.1 คำสั่งประกอบการทดลอง	41
ตารางที่ 4.2 ชุดคำสั่งที่ใช้ในการทดลอง	47
ตารางที่ 4.3 ผลการทดลองจากตารางที่ 4.2	51



สารบัญภาพ

รูปภาพ	หน้า
รูปที่ 2.1 แสดงบล็อกไดอะแกรมไมโครโปรเซสเซอร์เบื้องต้น	2
รูปที่ 2.2 แสดงโครงสร้างภายในไมโครโปรเซสเซอร์	4
รูปที่ 2.3 แสดงแอดเดรสรีจิสเตอร์	5
รูปที่ 2.4 แสดงการอ่านคำสั่งที่มาจาก IR	6
รูปที่ 2.5 แสดงการใช้ระบบไมโครโปรเซสเซอร์ทำงานแทนวงจรดิจิทัล	8
รูปที่ 2.6 แสดงผังงานเมื่อทำหน้าที่เป็นโมโนสเตเบิล	9
รูปที่ 2.7 ก แสดงเซลล์พื้นฐานของแรมชนิดสแตติก	11
รูปที่ 2.7 ข แสดงการต่อเซลล์ต่างๆเข้าด้วยกันภายในตัว IC	12
รูปที่ 2.8 แสดงการต่ออุปกรณ์ภายนอกกับ CPU หรือไมโครโปรเซสเซอร์	13
รูปที่ 2.9 แสดงโครงสร้างภายในของ 8212 อินพุทเอาต์พุทพอร์ท	14
รูปที่ 2.10 แสดงพอร์ทเบอร์ 8255	15
รูปที่ 2.11 แสดงการติดต่อระหว่างอุปกรณ์ภายนอกกับหน่วยความจำโดยผ่าน ทาง CPU	16
รูปที่ 2.12 แสดงการขอแลการตอบรับสัญญาณ WAIT,HOLDและอินเตอร์รัพท์ ของ CPU 8080	17
รูปที่ 3.1 แสดงบล็อกไดอะแกรมโครงสร้างของชุดฝึกอบรมปฏิบัติการ ไมโครโปรเซสเซอร์	19
รูปที่ 3.2 แสดงแสดงผังการทำงานของเครื่องโดยรวม	21
รูปที่ 3.3 แสดงผังการทำงานหน่วยบัสนินพุทเอาต์พุท	22
รูปที่ 3.4 แสดงผังงานประมวลผลทางคณิตศาสตร์	23
รูปที่ 3.5 แสดงหน้าเครื่องเมื่อเปิดเครื่อง	24
รูปที่ 3.6 แสดงผังการทำงานของคำสั่ง 3E	27
รูปที่ 3.7 แสดงการใช้พอร์ทที่ใช้งาน 8255 ทั้ง 3 ตัว	28
รูปที่ 3.8 แสดงวงจรส่วนสวิทช์	29
รูปที่ 3.9 แสดงวงจรส่วน LED	29
รูปที่ 3.10 แสดงวงจรชุดตัวเลขแสดงผล 7 หลัก	30
รูปที่ 3.11 แสดงส่วนแมทริกกี	31
รูปที่ 3.12 แสดงวงจรส่วนติดต่ออุปกรณ์นอก	31

บทที่ 1

บทนำ

ปัจจุบัน คอมพิวเตอร์มีบทบาทเกี่ยวข้องกับชีวิตประจำวันมากมายการมีความรู้ในเรื่องของคอมพิวเตอร์นับว่าสำคัญอย่างยิ่ง โดยเฉพาะในด้านอุตสาหกรรม ซึ่งจะต้องอาศัยผู้เชี่ยวชาญในด้านการใช้งานคอมพิวเตอร์กับงานอุตสาหกรรม เพื่อให้ได้จำนวนการผลิตที่มากขึ้น และทำให้ประเทศไทยได้ก้าวทันกับเทคโนโลยีอุตสาหกรรม ผู้เชี่ยวชาญเหล่านั้นจะต้องมีความเข้าใจในระบบของคอมพิวเตอร์ รู้โครงสร้าง , กระบวนการทำงาน , โปรแกรมควบคุม เป็นต้น

วิธีการในการศึกษาเล่าเรียนที่จะให้มีความรู้มีทักษะ ทั้งหมดพบในหลักสูตร ปวช , ปวส และระดับปริญญาตรีที่เป็นวิชาที่เกี่ยวกับไมโคร โปรเซสเซอร์เพื่อเป็นแนวทางในการศึกษาระบบที่มีขนาดใหญ่ขึ้นไป

จากปัญหาที่เกิดขึ้นจากการตรวจแบบสอบถาม พบว่าผู้สอนจะต้องรู้และเข้าใจอย่างลึกซึ้งสามารถถ่ายทอดให้เกิดการจินตนาการตามได้ ส่วนผู้เรียนจะต้องมีความสนใจมีการจินตนาการตามผู้สอน มีความคิดสร้างสรรค์และประยุกต์ได้ด้วยตัวเอง

ปริญญานิพนธ์ที่น่าเสนอนี้จะทำให้ผู้สอนสามารถอธิบายโครงสร้าง,กระบวนการทำงาน, ตลอดจนความหมายของโปรแกรมการดัดแปลงประยุกต์ใช้งานของไมโคร โปรเซสเซอร์ ซึ่งจะอ้างถึง ซีพียู (CPU) เบอร์ Z80 (โดยนำโครงสร้างที่จำเป็นต้องรู้มาแสดง) จะทำให้ผู้เรียนเกิดความสนใจมากกว่าที่เป็นอยู่ สามารถมองภาพกระบวนการทำงานที่เกิดขึ้นได้เข้าใจ

องค์ประกอบของชุดฝึกจะเน้นให้เกิดทักษะดังต่อไปนี้

1. กระบวนการเขียนการอ่านของตัวประมวลผลกลาง (CPU)
2. กระบวนการทำงานแต่ละขั้นตอนของคำสั่งหนึ่งๆ
3. การทำงานของหน่วยประมวลผลทางคณิตศาสตร์ (ALU)
- 4.การประยุกต์ใช้งานคำสั่งและโปรแกรมเบื้องต้น

ในการสร้างชุดฝึกนี้ได้นำข้อมูลมาจากชุดฝึก COMPUTER TRAINER ที่ใช้ CPU 8080 ถ้านำมาทำสื่อการเรียนการสอนจะมีความเหมาะสมมากยิ่งขึ้นรวมทั้งได้ดัดแปลงเพิ่มเติมบางส่วนเพื่อให้สามารถประกอบการสอนได้ทั้งในชั้นเรียนทั้งชั่วโมงทฤษฎีและชั่วโมงปฏิบัติโดยปฏิบัติตามใบงาน

ปริญญานิพนธ์นี้จะเป็นประโยชน์อย่างยิ่งในการสร้างความรู้ , ทักษะ , ประสบการณ์แก่ผู้ศึกษาซึ่งจะเป็นพื้นฐานต่อไปในการศึกษาทำให้เกิดความชำนาญและปฏิบัติการร่วมกับเทคโนโลยีอนาคตได้

บทที่ 2

ทฤษฎีและหลักการ

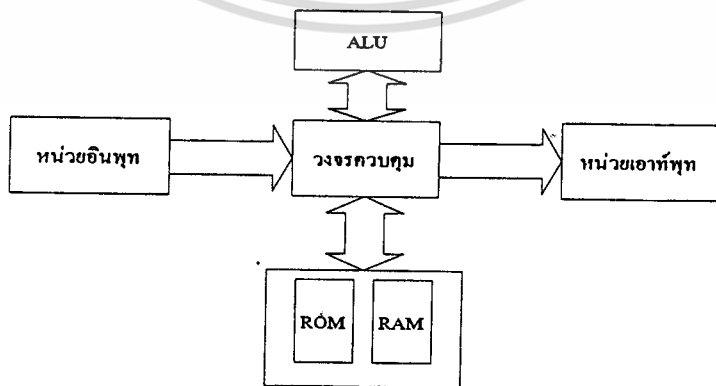
2.1 ไมโครโปรเซสเซอร์

ไมโครโปรเซสเซอร์ (Microprocessor) เป็นวิวัฒนาการทางเทคโนโลยีที่เด่นที่สุดในรอบ 30 ปีโดยกำเนิดเมื่อปีพ.ศ. 2491 เป็นต้นมา ไมโครโปรเซสเซอร์ไม่เพียงแต่จะทำให้เกิดการเปลี่ยนแปลงแนวความคิดครั้งยิ่งใหญ่ของวงจรอิเล็กทรอนิกส์เชิงเลข (Digital Electronics) เท่านั้นแต่มีผลกระทบต่อชีวิตความเป็นอยู่มนุษย์ ไมโครโปรเซสเซอร์ตัวแรกถูกสร้างขึ้น เมื่อปี พ.ศ. 2514 โดยบริษัทอินเทลคอร์ปอเรชัน (Intel Corporation) ซึ่งจัดเป็นไมโครโปรเซสเซอร์ชนิด 4 บิตชื่อว่าอินเทล 4004 จุดประสงค์ในการสร้างใช้ในเครื่องคำนวณอิเล็กทรอนิกส์ขนาดเล็ก ต่อมาบริษัทผู้ผลิตอุปกรณ์สารกึ่งตัวนำหลายบริษัททำการผลิตไมโครโปรเซสเซอร์ชนิดต่างๆ ขึ้นมาอีกจำนวนมาก โดยบริษัทผู้ผลิตส่วนใหญ่ใช้เทคโนโลยีการผลิต แอลเอสไอ (Large Scale Integration:LSI)

ไมโครโปรเซสเซอร์ทำหน้าที่เป็นหน่วยประมวลผลกลาง (Central Processing Unit:CPU) CPU เดิมทีนั้นทำจากไอซีจำนวนมากประกอบกันแต่ไมโครโปรเซสเซอร์ได้รวมไอซีจำนวนมากนั้นบรรจุลงในชิ้นส่วนเดียวกัน (1 Chip) ซึ่งทำให้ประหยัดเนื้อที่และสะดวกในการใช้งานจึงเป็นที่นิยมใช้กันในปัจจุบัน

2.1.1 หลักการทำงานของระบบไมโครโปรเซสเซอร์

สามารถแบ่งออกเป็นหน่วยใหญ่ๆ ได้ดังบล็อกไดอะแกรมต่อไปนี้



รูปที่ 2.1 แสดงบล็อกไดอะแกรมไมโครโปรเซสเซอร์เบื้องต้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทำงานของระบบไมโครโปรเซสเซอร์ภายในประกอบด้วยส่วนประกอบพื้นฐานอยู่ 5 หน่วยคือหน่วย CPU จะรวมวงจรควบคุม (Control Unit) และวงจรคำนวณทางคณิตศาสตร์และลอจิก (Arithmetic Logic Unit:ALU) เข้าไว้ด้วยกัน หน่วยความจำ (Memory Unit) หน่วยอินพุต (Input Unit) และหน่วยเอาต์พุต (Output Unit) จะทำหน้าที่ป้อนข้อมูลคำนวณทางคณิตศาสตร์และคำนวณทางลอจิกให้กับหน่วย ALU ดังแสดงตามรูป

หน่วยควบคุม วงจรจะทำหน้าที่ควบคุมการทำงานของอุปกรณ์ทุกอย่างในระบบ จะผลิตสัญญาณที่ใช้ในการซิงโครไนส์ (Synchronize) การทำงานและการส่งถ่ายข้อมูลเข้าออก ALU และภายนอก CPU โดยควบคุมผ่านบัสแอดเดรส (Address bus) และบัสข้อมูล (Data bus) และจะทำการแปลความหมายของสัญญาณต่างบนบัสควบคุม (Control bus) ซึ่งรับจากวงจรภายนอก CPU หน้าที่สำคัญของวงจรควบคุมคือ การอ่านคำสั่งที่เก็บไว้ในหน่วยความจำมาถอดรหัส (Decode) และการทำงานจะเป็นจังหวะซ้ำกันคือ จังหวะในการอ่านคำสั่งเฟตช์ (Fetch) จังหวะถอดรหัสของคำสั่งและจังหวะทำงานตามคำสั่งเอ็กซีคิว (Execute) กลุ่มคำสั่งที่ทำการเขียนไว้ในหน่วยความจำจะเรียกว่าโปรแกรม (Program)

หน่วยความจำ เป็นหน่วยข้อมูลและเก็บคำสั่ง หน่วยความจำแบ่งออกง่ายๆ เป็น 2 ชนิด ได้แก่ รม (Read Only Memory :ROM) คือหน่วยความจำที่สามารถอ่านข้อมูลได้อย่างเดียว และแรม (Random Access Memory :RAM) คือหน่วยความจำที่ใช้ได้ทั้งการอ่านข้อมูลและเขียนข้อมูล หน่วยความจำทั้ง 2 ชนิดเป็นหน่วยความจำที่ทำจากสารกึ่งตัวนำ (IC memory) แรมสะดวกตรงที่สามารถอ่านและเขียนข้อมูลได้หน่วยความจำจะเป็นที่ไว้เก็บข้อมูล 2 ชนิดคือคำสั่งและข้อมูล

หน่วยอินพุต ทำหน้าที่รับข้อมูลป้อนเข้าสู่ CPU ตัวอย่างของหน่วยอินพุตได้แก่ แป้นกดข้อมูล (Keyboard) หรือตัวตรวจวัด เช่น ตัววัดอุณหภูมิตัววัดแรงดัน ตัววัดตรวจจับแสงเป็นต้น

หน่วยเอาต์พุต ทำหน้าที่ส่งข้อมูลภายใน CPU ออกแสดงภายนอกตัวอย่างได้แก่ ไลโอดเปล่งแสง (LED) เครื่องพิมพ์ (Printer) หลอดไฟ (Lamp) ทีวีและเครื่องใช้ไฟฟ้าต่างๆเป็นต้น

หน่วยอินพุตเอาต์พุต ตามปกติจะไม่ต่อโดยตรงกับ ALU แต่จะต่อกับ CPU โดยผ่านทางบัสแอดเดรส บัสข้อมูลและบัสควบคุม การเขียนและการอ่านข้อมูลที่หน่วยอินพุต และเอาต์พุต มักจะทำผ่าน CPU โดยข้อมูลจะส่งถ่ายจากหน่วยความจำใน CPU ไปยังหน่วยเอาต์พุต หรืออินพุตจะส่งข้อมูลมายัง CPU หรือ หน่วยความจำ

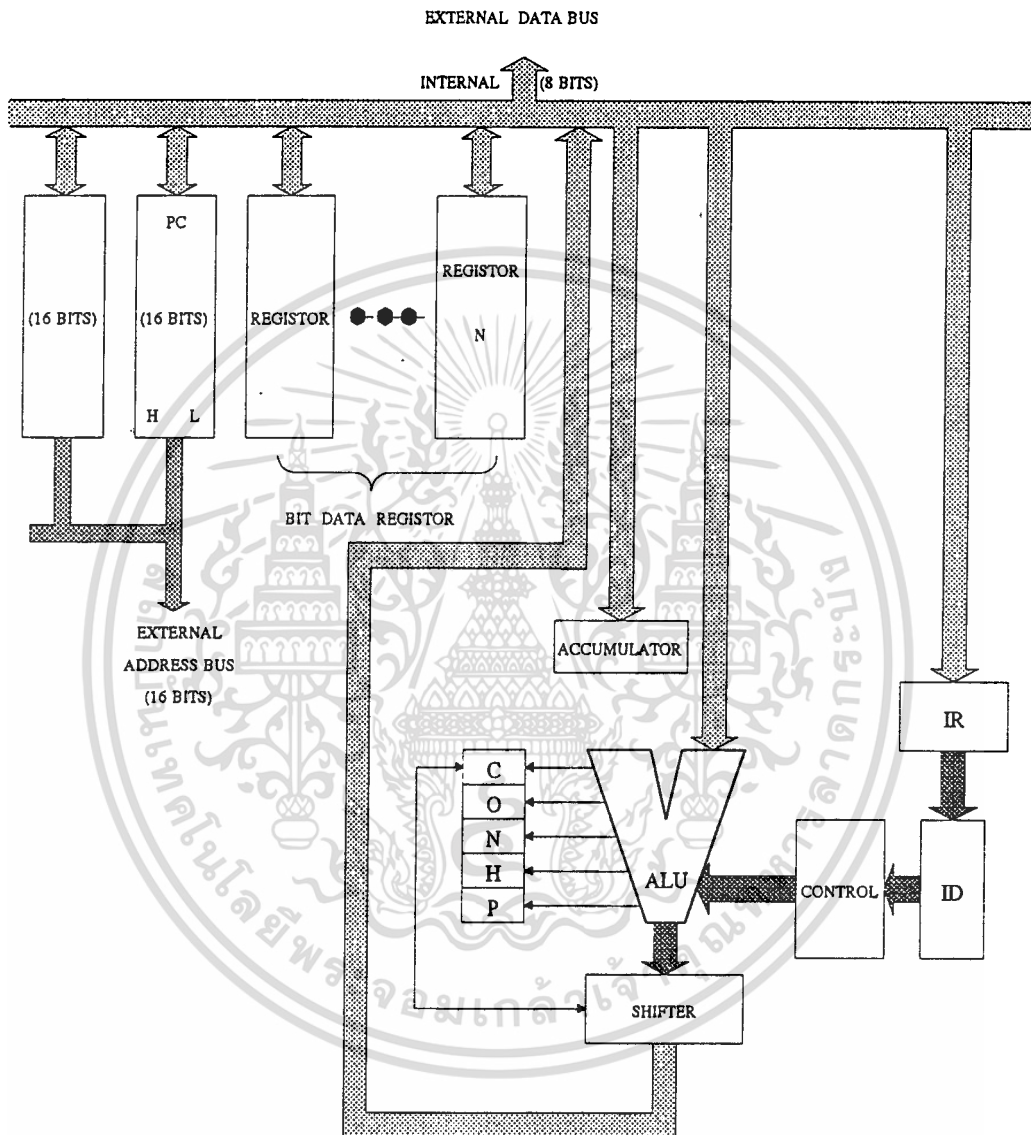
2.1.2 โครงสร้างภายในและขบวนการทำงานของไมโครโปรเซสเซอร์

โครงสร้างภายในไมโครโปรเซสเซอร์

ส่วนประกอบที่สำคัญที่สุดของไมโครโปรเซสเซอร์มี ALU รีจิสเตอร์ข้อมูลและวงจรถอด

รหัสคำสั่ง (Instruction Register : IR and Instruction Decoder :ID) วงจรควบคุม (Control Unit) เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

circuit) และรีจิสเตอร์สำหรับใช้งานส่วนประกอบที่สำคัญนี้จะติดต่อกับบัสภายใน (Internal bus) มีเส้นสัญญาณจำนวนเท่ากับจำนวนบิตของไมโครโปรเซสเซอร์ บัสภายในนี้จะต่อกับบัสภายนอกโดยผ่านบัฟเฟอร์ (Buffer) เพื่อเป็นการติดต่อภายนอกตามรูปแสดงต่อไปนี้



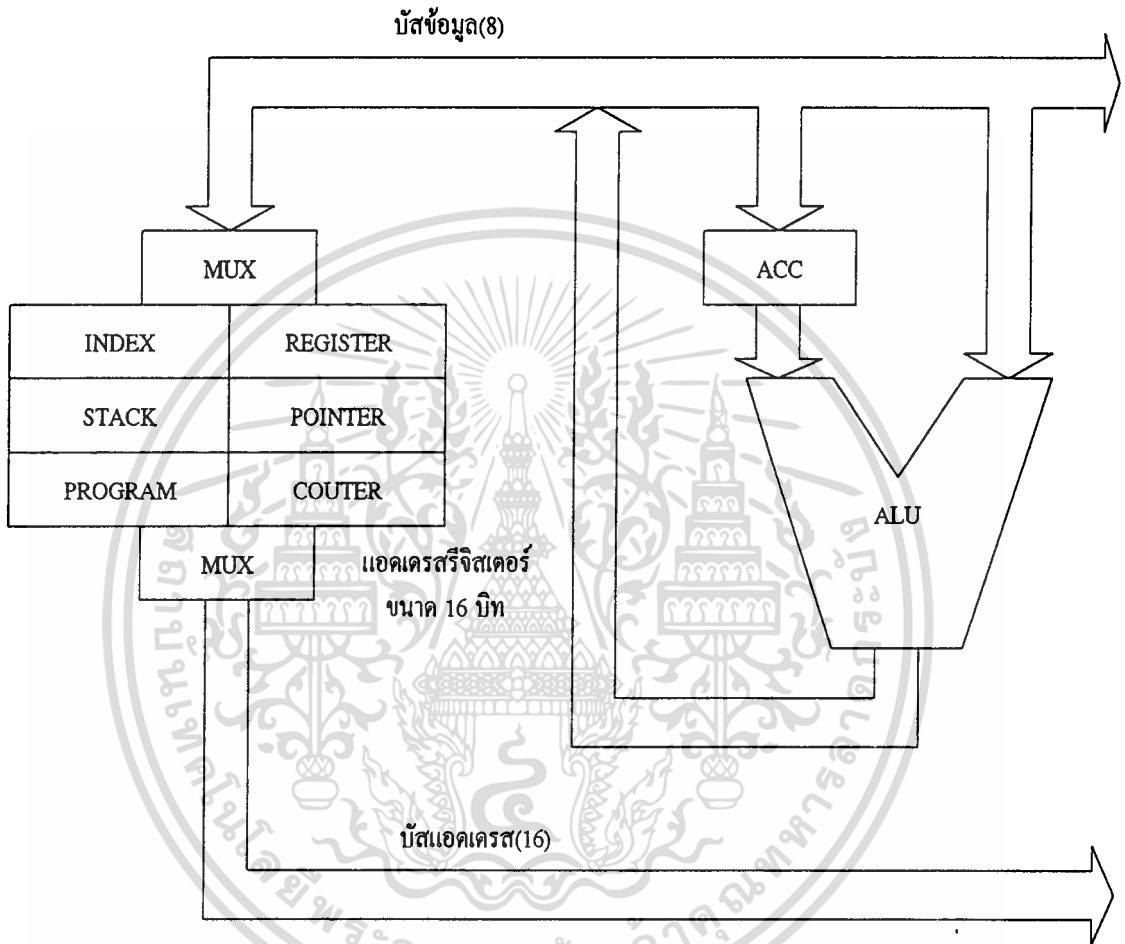
รูปที่ 2.2 แสดงโครงสร้างภายในไมโครโปรเซสเซอร์

ขบวนการทำงานไมโครโปรเซสเซอร์

ALU จะทำหน้าที่คำนวณในไมโครโปรเซสเซอร์โดยจะทำการคำนวณและประมวลผลทางคณิตศาสตร์ และ คำนวณทางลอจิกข้อมูล ที่จะคำนวณจะเข้าสู่ **ALU** ได้ทั้ง 2 ทางดังรูปที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทางด้านซ้ายจะรับข้อมูลจากรีจิสเตอร์พิเศษตัวหนึ่งเรียกว่า แอควิวูเลเตอร์ (Accumulator) ทางด้านขวาจะรับข้อมูลจากบัสข้อมูลภายในเพื่อทำการติดต่อยีจิสเตอร์ต่างๆ แอควิวูเลเตอร์จะเป็นรีจิสเตอร์ทำงานร่วมกับ ALU และเป็นทั้งอินพุตและเอาท์พุท



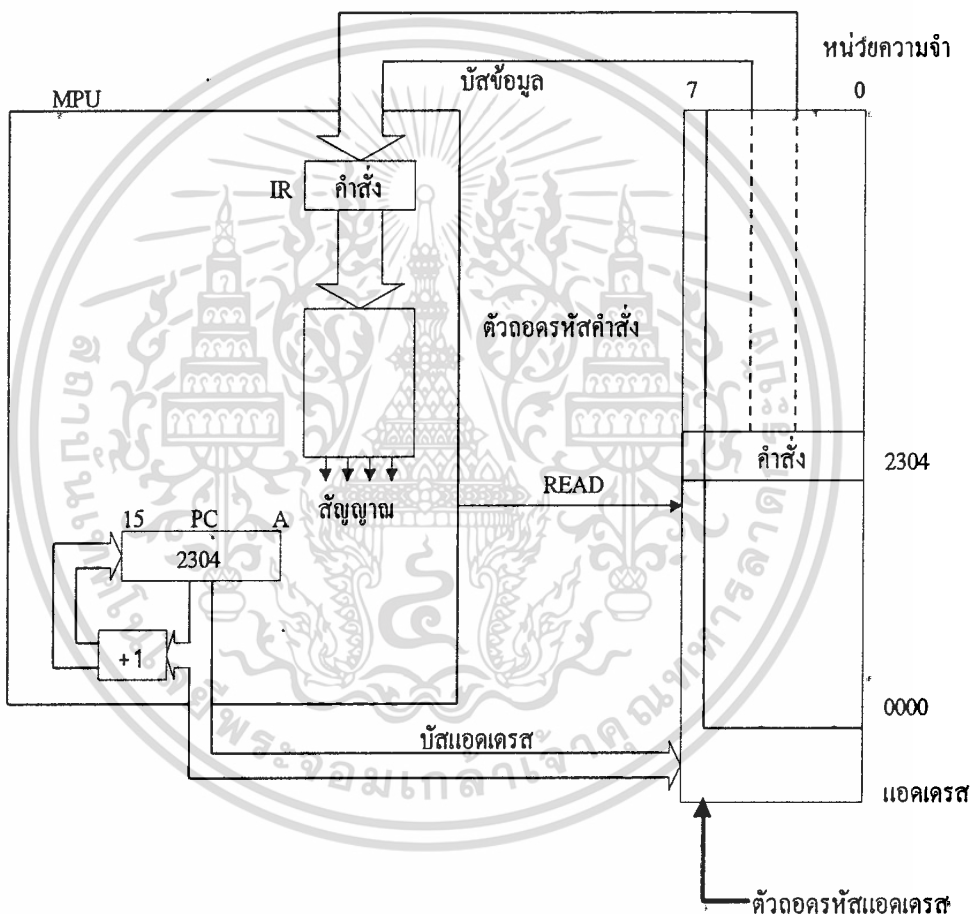
รูปที่ 2.3 แสดงแอควิวูเลเตอร์

ตัวอย่างเช่นทำการบวกข้อมูลของแอควิวูเลเตอร์กับข้อมูลในรีจิสเตอร์ใดตัวหนึ่งผลลัพธ์จะมาเก็บไว้ในแอควิวูเลเตอร์เสมอ ส่วนการคำนวณทางตรรก เช่น การเลื่อนข้อมูลที่ทำกรคำนวณกับข้อมูลในแอควิวูเลเตอร์ ก็เช่นเดียวกัน ให้เปรียบเทียบแอควิวูเลเตอร์เหมือนกระดานลูกคิดที่ทั้งตัวบวกและผลลัพธ์จะปรากฏบนกระดานเองเสมอ

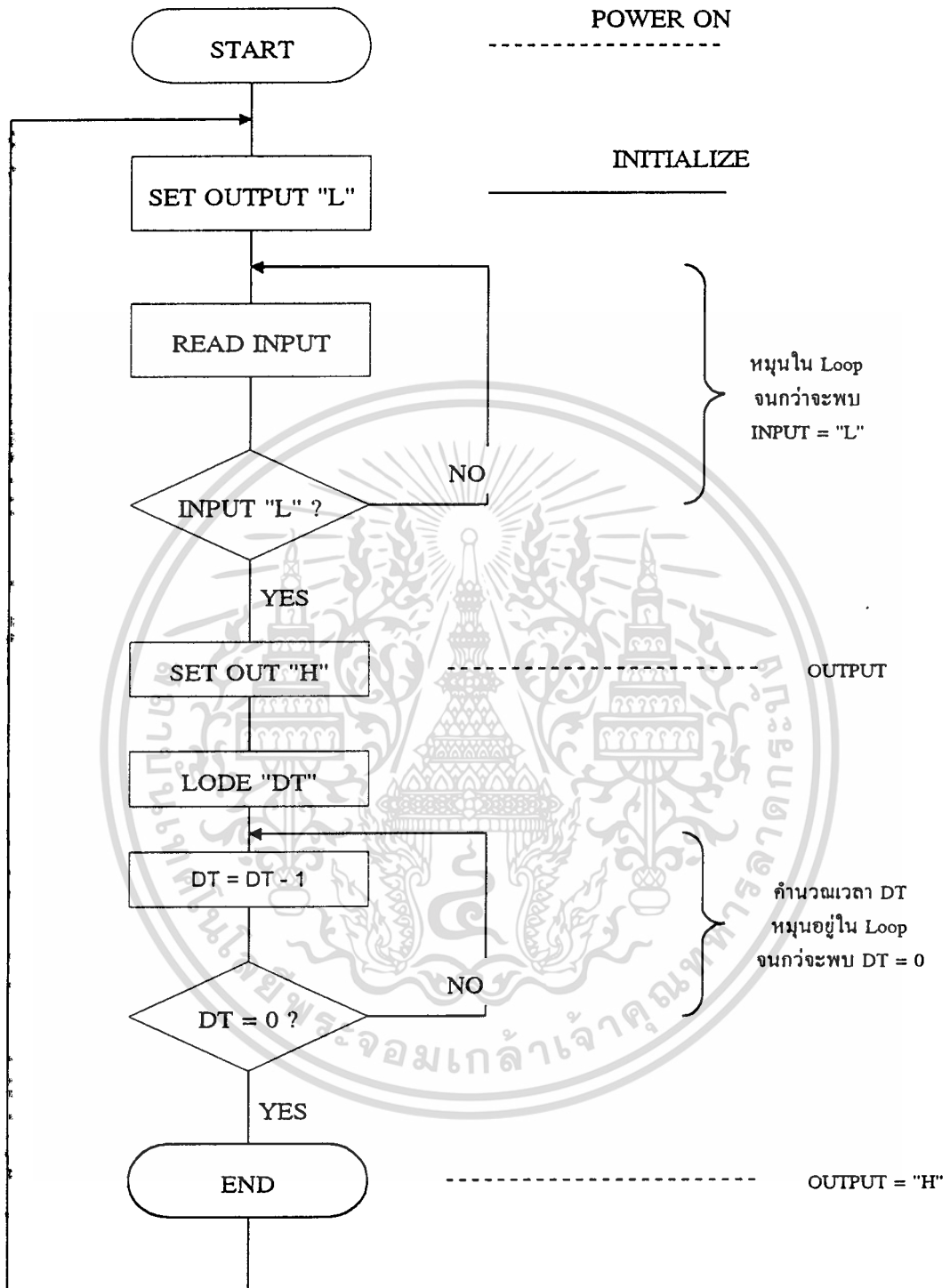
แฟลกรีจิสเตอร์ (Flag register) ทำหน้าที่แสดงสถานะการทำงานของ ALU ภายในโดยที่รีจิสเตอร์นี้แต่ละบิตจะแสดงสถานะการทำงานของ CPU ที่แตกต่างกัน

รีจิสเตอร์ข้อมูล (Data register) ทำหน้าที่เก็บข้อมูลชั่วคราวในขั้นตอนของการคำนวณ รีจิสเตอร์นี้ต่อกับบัสภายใน สามารถส่งข้อมูลเข้าออก ALU หรือแอมพลิฟายเออร์ได้อย่างรวดเร็ว ข้อมูลจากหน่วยความจำซึ่งส่งเข้ามาจากบัสภายนอก จะนำมาเก็บไว้ในรีจิสเตอร์ข้อมูลนี้เสมอ โดยจำนวนของรีจิสเตอร์จะแตกต่างกันตามชนิดของไมโครโปรเซสเซอร์

แอดเดรสรีจิสเตอร์ (Address register) ทำหน้าที่เป็นรีจิสเตอร์ใช้เก็บข้อมูลของแอดเดรส รีจิสเตอร์จะเป็นตัวชี้บอกแอดเดรสของหน่วยความจำเพื่อทำให้เกิดการอ่านและเขียนหน่วยความจำได้ปกติจะต่อเข้ากับบัสแอดเดรสเพื่อติดต่อกับวงจรภายนอกดังรูปที่ 2.3



รูปที่ 2.4 แสดงการอ่านคำสั่งจากหน่วยความจำที่มาจากที่ IR



รูปที่ 2.6 แสดงผังงานเมื่อทำหน้าที่เป็นโมโนสเตเบิล

2.2 ระบบหน่วยความจำ

ระบบหน่วยความจำในไมโครโปรเซสเซอร์จะแบ่งออกเป็น 2 ชนิดคือชนิดจำถาวรรวมเอกสาร (ROM) และชนิดชั่วคราวที่สามารถเปลี่ยนแปลงข้อมูลภายในได้โดยง่ายคือแรม (RAM) มีด้านการคำนวณว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 รอม

รอมเป็นหน่วยความจำที่ใช้เก็บข้อมูลแบบถาวรหรือกึ่งถาวรการเขียนข้อมูลทำได้ 2 วิธีคือการเขียนข้อมูลจากโรงงานผู้ผลิตตามความต้องการของลูกค้าเป็นชนิดมาส์คโปรแกรมเมटरอม (Mask Programmed Rom) ส่วนอีกแบบหนึ่งคือผู้ใช้สามารถโปรแกรมได้เองเรียกว่าเป็นชนิดพรอม (Programmed Read Only Memory :PROM) และมีบางประเภทที่ผู้ใช้สามารถเปลี่ยนข้อมูลในหน่วยความจำได้แต่ในการเขียนข้อมูลแต่ละครั้งต้องใช้เวลามากกว่าเวลาที่ใช้ในการอ่านข้อมูลจากหน่วยความจำมากประเภทนี้เรียกว่ารีโปรแกรมเมเบิลรอม (Reprogrammable ROM) ข้อดีของหน่วยความจำตระกูลนี้คือ ข้อมูลจะไม่สูญหายถึงแม้ไฟฟ้ามดับก็ตามเราเรียกลักษณะอย่างนี้ว่าไม่เปลี่ยนแปลงตามเวลา (nonvolatile) ซึ่งต่างไปจากหน่วยความจำตระกูลแรมซึ่งเป็นชนิดจะเปลี่ยนแปลงตามเวลา (volatile)

ชนิดของรอม

มาส์คโปรแกรมเมटरอม เป็นหน่วยความจำที่โปรแกรมไว้อย่างถาวรในระหว่างกระบวนการผลิตโดยการทำมาส์คเฉพาะสำหรับงานนั้นๆ รอมประเภทนี้สามารถทำได้จากสิ่งประดิษฐ์จำพวกไดโอดบีเจที (BJT) หรือมอส (MOS) ก็ได้

คุณสมบัติของรอมชนิดนี้มีดังต่อไปนี้คือ เก็บข้อมูลได้ตลอดกาลแม้ไฟเลี้ยงวงจรหรือไม่ก็ตาม ผู้ผลิตเป็นผู้โปรแกรมข้อมูลในหน่วยความจำ เราไม่สามารถเปลี่ยนแปลงข้อมูลนี้ได้ ข้อดีข้อเสียก็คล้ายคลึงกับของรอมชนิดอื่นแต่ต่างกันบ้างก็ตรงที่ขนาดของหน่วยความจำจำนวนเวิร์ด คุณกับจำนวนบิตต่อ 1 เวิร์ด อาจเล็กหรือใหญ่ตามต้องการได้รอมชนิดนี้มีประโยชน์เฉพาะกรณีที่มีความต้องการสูงมีฉะนั้นจะไม่คุ้มค่ากับการลงทุนในการผลิต

พรอม ได้ที่จริงแล้วมีลักษณะวงจรภายในจะคล้ายคลึงกับพวก มาส์คโปรแกรมเมटरอมภายในรอม จุดตัดทุกจุดในตารางกริดจะมีสิ่งประดิษฐ์สารกึ่งตัวนำต่ออยู่แต่การโปรแกรมให้ข้อมูลแต่ละบิตเป็น 0 หรือ 1 นั้นใช้วิธีปล่อยกระแสสูงๆไหลผ่านฟิวส์ที่มีต่อร่วมกับสิ่งประดิษฐ์เพื่อให้ฟิวส์หลอมขาด

คุณสมบัติของพรอมก็คือเป็นชนิดไม่เปลี่ยนแปลงตามเวลาซึ่งโปรแกรมข้อมูลได้ แต่ไม่สามารถเปลี่ยนแปลงข้อมูลหลังจากที่โปรแกรมไปแล้วได้เพราะฟิวส์ที่เคยต่ออยู่ขาดไปแล้วไม่สามารถที่จะต่อกลับคืนดังเดิมได้อีก

- รีโปรแกรมเมเบิล ทำมาจากมอสเฟตโดยที่สภาวะ 0 หรือ 1 แทนการนำกระแสหรือไม่นำกระแสของมอสเฟตโดยการควบคุมแรงดันที่ขาทะในมอสเฟตขาทะถูกแยกจากส่วนที่นำกระแสอย่างเด็ดขาดดังนั้นแรงดันที่ขาทะจึงขึ้นอยู่กับประจุที่ขาทะลักษณะ โครงสร้างของขาทะ

กับส่วนนำกระแสของมอเฟทนั้นเปรียบเสมือนตัวเก็บประจุที่มีชั้นออกไซด์เป็นตัวไดอิเล็กทริกเป็นแนวคิดในการพัฒนารีโปรแกรมเมเบิลรอมขึ้น

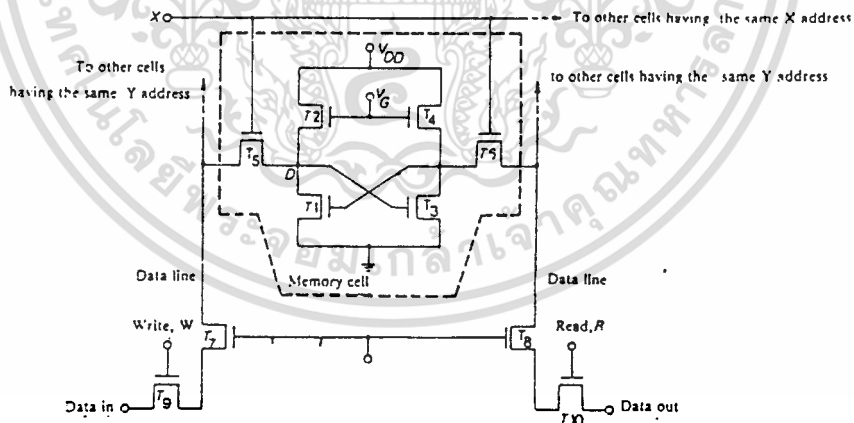
ข้อดีของรีโปรแกรมเมเบิลรอมอยู่ที่ความสามารถในการเปลี่ยนแปลงข้อมูลในหน่วยความจำได้ ซึ่งสะดวกมากสำหรับการพัฒนาทางซอฟต์แวร์ (Software) ซึ่งการเปลี่ยนแปลงข้อมูลภายในรอมชนิดนี้จะใช้เครื่องโปรแกรมรอม

2.2.2 แรม

แรมเป็นหน่วยความจำที่ต่างไปจากรอมทั้งในลักษณะโครงสร้างและการทำงานเราสามารถเขียนหรืออ่านข้อมูลจากแรมโดยใช้เวลาพอกัน แรมทำจากสิ่งประดิษฐ์จำพวกบิเจทหรือมอสเฟทก็ได้ โดยมากแรมที่ทำมาจากบิเจทมักมีความเร็วสูงกว่าพวกที่ทำมาจากมอสเฟท

การเก็บข้อมูลในเซลล์หน่วยความจำแต่ละเซลล์ของแรม มี 2 ลักษณะคือ

1. การใช้ลักษณะวงจรฟลิปฟล็อปเป็นเซลล์พื้นฐานในการจำข้อมูล เราใช้ลักษณะนี้มากในวงจรบิเจทแรม และวงจรมอสแรมบางชนิด
2. การใช้การประจุตัวเก็บประจุที่ขาคาทของมอเฟทเป็นการเก็บข้อมูล นิยมใช้มากในวงจรมอสแรม



รูปที่ 2.7 ก แสดงเซลล์พื้นฐานของ แรม ชนิด STATIC

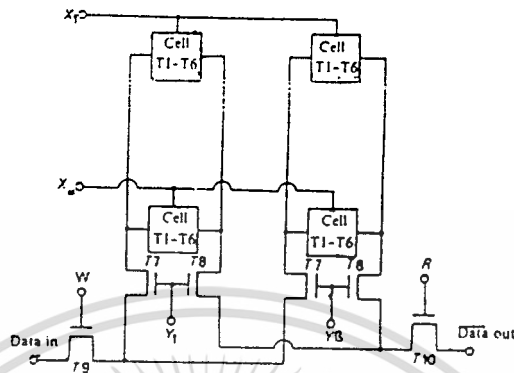
ชนิดของ แรม

1. แรม ชนิดสถิตยชนิดนี้ใช้ลักษณะวงจรฟลิปฟล็อปเป็นวงจรพื้นฐานสำหรับเซลล์หน่วยความจำแต่ละเซลล์ตามปกติวงจรพื้นฐานก็มักไม่แตกต่างกันมากนักและโดยมากจะเป็นดังใน

เอกสารรูปที่ 2.7 ก T1-T6 คือทรานซิสเตอร์ที่ประกอบเป็นวงจรฟลิปฟล็อปในเซลล์พื้นฐานส่วน T7-T10

คำว่ากรณีใดๆทั้งนั้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นเกทที่เปิดให้ข้อมูลไหลผ่านไปได้ เป็นวงจรรวมของเซลล์หน่วยความจำหลายๆเซลล์ ดังนั้นเราจึงเรียกเซลล์หน่วยความจำนี้ว่าเซลล์หน่วยความจำชนิดสแตติกแบบทรานซิสเตอร์ 6 ตัว



รูปที่ 2.7 ข แสดงการต่อเซลล์ต่างๆเข้าด้วยกันภายในตัว IC

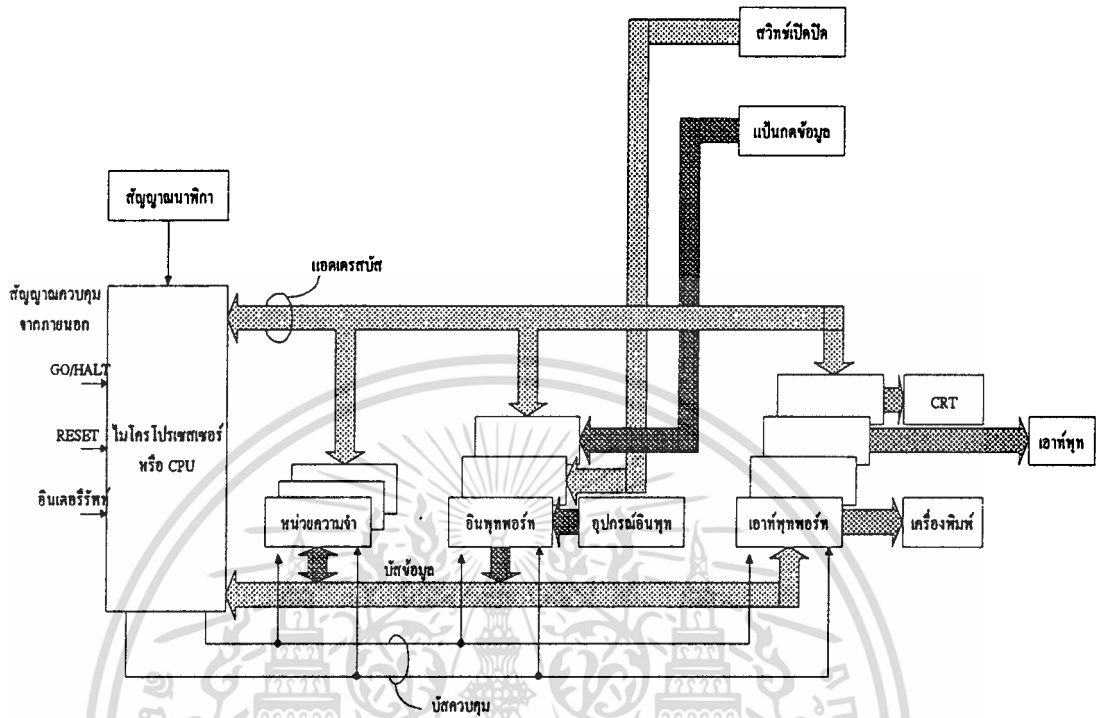
ลักษณะการทำงานเป็นดังนี้ T1,T2 และ T3,T4 ประกอบกันเป็นวงจรกลับเฟสที่ต่อโยงไขว้กันอยู่ ทำหน้าที่เป็นฟลิปฟล็อปโดย T5,T6 เป็นเกทเปิดหรือปิดตามแต่ละระดับแรงดันในสาย X T7,T8 เป็นเกทเปิดหรือปิด ตามแต่ละระดับแรงดันในสาย Y T9,T10 เลือกว่าจะให้เขียนข้อมูลเข้าไปหรืออ่านข้อมูลออกมาเมื่อต้องการเลือกเซลล์ใดก็ให้สาย X,Y ของเซลล์นั้นจะได้ค่าเป็น 1 ทำให้สายข้อมูล และสายข้อมูลต่อกับเซลล์ได้ และยังต่อออกมาที่สาย Data in หรือสาย Data out ก็ได้ด้วยการให้ W หรือ R เป็น 1 ตามลำดับ

2. แรม ชนิดไดนามิกใช้การเก็บประจุในตัวเก็บประจุที่ขาทของมอสเฟตแทนการเก็บข้อมูลวงจรพื้นฐานของแรมชนิดไดนามิกก็ได้หลายรูปแบบและใช้จำนวนทรานซิสเตอร์แตกต่างกันไปแรมชนิดไดนามิกก็ได้มีข้อดีเหนือกว่าแรมชนิดสแตติกมากนั่นคือการมีเวลาเข้าถึงหน่วยความจำน้อยกว่าแรมชนิดสแตติกเล็กน้อย แต่ต้องเสียเวลาในการรีเฟรชช่วยความจำทั้งหมด การมีการสิ้นเปลืองกำลังไฟฟ้าต่ำกว่าของแรมชนิดสแตติกอื่นเนื่องจากกระแสไหลผ่านวงจรขณะเขียนหรืออ่านรีเฟรชเท่านั้นแต่สิ่งที่น่าสนใจอย่างยิ่งคือการใช้จำนวนทรานซิสเตอร์ต่อ 1 เซลล์น้อยกว่าของแรมชนิดสแตติกค่อนข้างมากทำให้เนื้อที่ 1 ตารางหน่วย สามารถบรรจุแรมชนิดไดนามิกได้จำนวนเซลล์มากกว่าของแรมชนิดสแตติกค่อนข้างมาก

2.3 หน่วยอินพุตและเอาต์พุต

เรื่องของอินพุตและเอาต์พุตจำเป็นต้องรู้โครงสร้างของระบบว่าเป็นอย่างไรโดยมีรูปประกอบที่ รูปที่ 2.8 แสดงถึงการต่ออุปกรณ์ต่างๆ เข้ากับ CPU การส่งถ่ายข้อมูลนั้น จะส่งผ่านบัสที่ใช้ร่วมกัน CPU และหน่วยความจำการส่งถ่ายข้อมูลจากอุปกรณ์ภายนอกเข้าบัสข้อมูลของCPU

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อการเรียนการสอนเพื่อให้นักศึกษาได้เรียนรู้และเข้าใจเนื้อหาโดยไม่ต้องเสียค่าใช้จ่ายในการดาวน์โหลดเอกสารนี้ หากท่านใดต้องการนำเอกสารนี้ไปใช้โดยไม่ผ่านการอนุญาตจากข้าพเจ้าถือว่าผิดกฎหมายและต้องรับผิดชอบต่อเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แสดงการต่ออุปกรณ์ภายนอกกับ CPU หรือ ไมโครโปรเซสเซอร์

จะมีอินพุทเอาท์พุทอินเตอร์เฟส (I/O interface) เป็นตัวกลาง มีแอดเดรสบัสเป็นตัวเลือกพอร์ท (Port) และบัสควบคุมจะส่งสัญญาณมาควบคุมการส่งถ่ายข้อมูล การจะส่งข้อมูลเข้าหรือออกที่อุปกรณ์ตัวใด ทำโดยใช้คำสั่งอินพุทข้อมูล เช่น IN หรือ OUT ตามปกติคำสั่งอินพุทหรือเอาท์พุทจะมีส่วนหนึ่งของคำสั่งสำหรับบอกหมายเลขอุปกรณ์ที่จะติดต่อกับ CPU หมายเลขนี้จะถูกถอดรหัสโดยส่วนควบคุมการเลือกอุปกรณ์ที่จะติดต่อ การส่งเลขหมายอาจใช้แอดเดรสบัสเป็นตัวเลือกอุปกรณ์โดยตรงได้ เช่น ตัวอย่างคำสั่งต่อไปนี้

IN 03 (Acc) ----- (ข้อมูลจากอุปกรณ์ 03)

คำสั่งนี้เป็นการนำข้อมูลจากอุปกรณ์ผ่านอินพุทเอาท์พุทพอร์ทหมายเลข 03 ทำการเข้าไปเก็บในแอดเดรสแอมคิวมูลเตอร์ของ CPU ข้อมูลที่ผ่านจากพอร์ทที่จะนำไปทำอะไรต่อนั้นก็ขึ้นอยู่กับคำสั่งต่างๆที่ตามมา

อินพุตและเอาต์พุตพอร์ท

อุปกรณ์ภายนอกที่ต้องการส่งถ่ายข้อมูลกับ CPU จะต้องอาศัยพอร์ทเป็นตัวกลาง โดยพอร์ทจะทำหน้าที่เป็นช่องทางส่งผ่านข้อมูลสำหรับอุปกรณ์แต่ละตัว อินพุตเอาต์พุตพอร์ทจะประกอบด้วยวงจรที่สามารถสลับเปลี่ยนข้อมูลเข้าออก CPU ได้ พอร์ทเหล่านี้จะถูกเลือกก็ต่อเมื่อถูกเอ็กซีคิวต์คำสั่งเกี่ยวกับอินพุตและเอาต์พุต อินพุตเอาต์พุตพอร์ทมีทั้งสำหรับเป็นอินพุตพอร์ทและเอาต์พุตพอร์ท

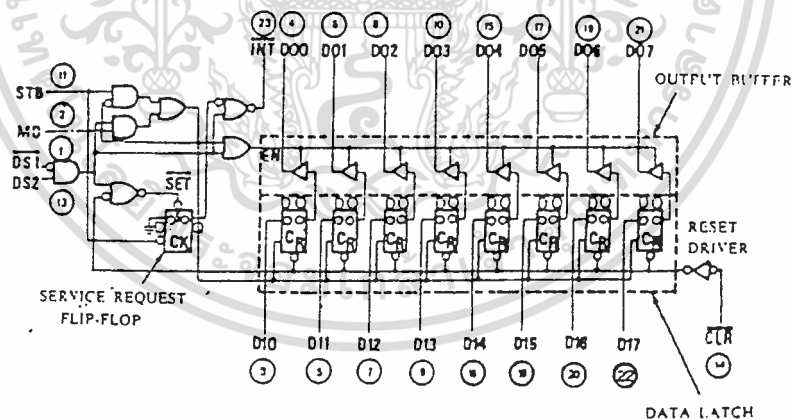
อินพุตพอร์ทเป็นช่องทางสำหรับการนำข้อมูลของ CPU

เอาต์พุตพอร์ทเป็นช่องทางสำหรับนำเอาต์พุตออกจาก CPU

ปกติอินพุตเอาต์พุตพอร์ทมีแบบการส่งถ่ายข้อมูล 2 แบบคือ การส่งถ่ายข้อมูลแบบขนาน และการส่งถ่ายข้อมูลแบบอนุกรม ด้านหนึ่งของพอร์ทจะต่อกับบัสข้อมูลของ CPU ส่วนอีกด้านสำหรับต่อเข้ากับอุปกรณ์ภายนอก

2.3.1 อินพุตและเอาต์พุตพอร์ทแบบขนาน (Parallel I/O port)

ตัวอย่างรูปที่ 2.9 เป็นอินพุตเอาต์พุตพอร์ทแบบขนานชนิด 8 บิต เบอร์ 8212 ซึ่งใช้เป็นอินพุตเอาต์พุตพอร์ทสำหรับการส่งข้อมูลแบบขนานอย่างธรรมดา



รูปที่ 2.9 แสดงโครงสร้างภายในของ 8212 อินพุตเอาต์พุตพอร์ท

ช่องทางการส่งผ่านข้อมูลแบ่งเป็น 2 ส่วน คือด้านรับข้อมูลเข้า $DI_0 - DI_7$ ด้านส่งข้อมูลออก $DO_0 - DO_7$ ด้านรับข้อมูลเข้าประกอบด้วย 8 บิต แลทช์ (Latch) เป็นพวก D ฟลิปฟลอปด้านส่งข้อมูลออกประกอบด้วย 8 บิตเอาต์พุตบัฟเฟอร์ เป็นพวกไตรสเททบัฟเฟอร์นอกจากช่องทางส่งผ่านข้อมูลแล้ว ยังมีสายควบคุมซึ่งใช้ควบคุมการเคลื่อนย้ายข้อมูลต่างๆอีก MD เป็นขั้วสัญญาณเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



สำหรับการเลือกโหมด (Mode) อยู่ที่ว่าการทำงานจะเป็นอินพุทโหมดก็มีสัญญาณที่เป็นลอจิก “1” หรือ เอาท์พุทโหมด MD ก็จะมีสัญญาณลอจิกเป็น “0” ซึ่งก็จะทำงานได้

2.3.2 อินพุทและเอาท์พุทแบบอนุกรม (Serial I/O port)

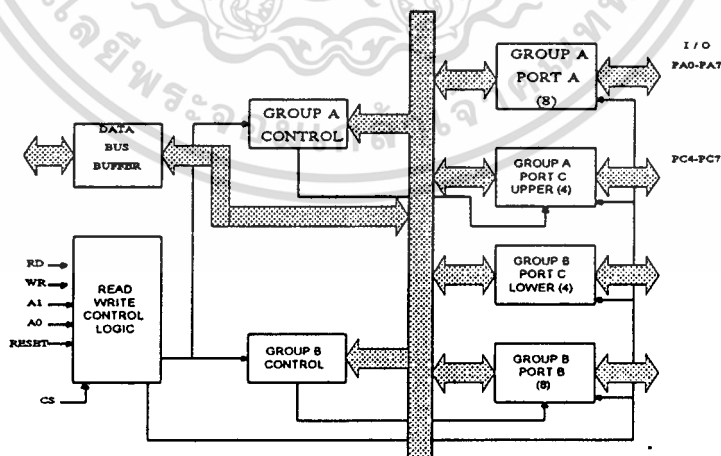
ในที่นี้จะยกตัวอย่างอินพุทเอาท์พุทพอร์ตแบบอนุกรม เบอร์ 8255 ซึ่งเป็นเบอร์ที่นิยมใช้กับไมโครโปรเซสเซอร์มากในปัจจุบัน โดยมีโครงสร้างดังรูปต่อไปนี้

ภายในตัวไอซีที่ใช้ในการรับส่งข้อมูลแบบอนุกรมมีส่วนประกอบสำคัญๆ ดังนี้

1. วงจรส่งข้อมูลออกแบบอนุกรมและรับข้อมูลแบบอนุกรม
2. การกำหนดแอดเดรสด้านรับและส่งจะมีขั้วกำหนดการทำงานของพอร์ตโดยรับสัญญาณรหัสจากบัสแอดเดรสหรือตัวถอดรหัสในแอดเดรส
3. บัสควบคุมพอร์ตรับส่งข้อมูลแบบอนุกรม

2.3.3 อินเตอร์รัพท์ (Interrupt)

อินเตอร์รัพท์ หมายถึง การขัดจังหวะการทำงานของ CPU เพื่อให้ CPU หยุดการทำงานของโปรแกรมที่กำลังอยู่เมื่อได้รับการอินเตอร์รัพท์จากอุปกรณ์ภายนอก CPU จะทำงานในคำสั่งที่ทำอยู่จนจบคำสั่งนั้นเสียก่อนแล้ว CPU จึงไปบริการให้กับงานที่ ส่งสัญญาณ อินเตอร์รัพท์เข้ามา เมื่อบริการอินเตอร์รัพท์เสร็จแล้วก็กลับไปทำงานที่โปรแกรมเดิมต่อไป



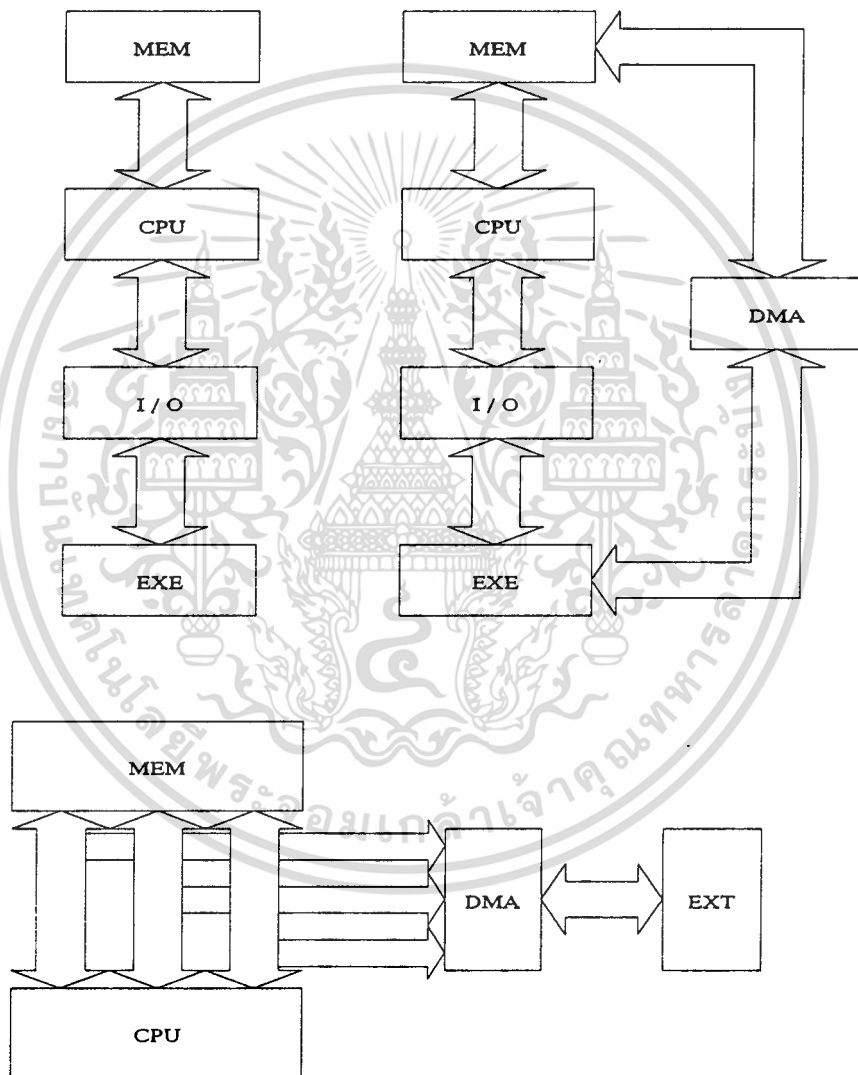
รูปที่ 2.10 แสดงพอร์ตเบอร์ 8255

ในการอินเตอร์รัพท์มีลักษณะคล้ายคลึงกับการที่ CPU กระโดดไปทำงานที่โปรแกรมย่อย เอกสารนี้เป็นเอกสารที่ลงนามไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า จะแตกต่างกันก็คือ การกระโดดภายในโปรแกรมจะกระโดดไปที่ใดเมื่อมีคำสั่งจัมพ์ (JUMP) แต่ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามเผยแพร่ข้อมูลนี้ และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอินเทอร์รัพท์จะกระโดดไปทำงานโปรแกรมหนึ่งเมื่อ CPU ได้รับสัญญาณอินเทอร์รัพท์จากอุปกรณ์ภายนอก

2.3.4 การเข้าถึงหน่วยความจำโดยตรง (Direct Memory Access : DMA)

คำจำกัดความ DMA คือการส่งข้อมูลเข้าไปเก็บในหน่วยความจำ หรือการนำข้อมูลที่เก็บไว้ในหน่วยความจำไปยังอุปกรณ์ภายนอกโดยไม่ผ่าน CPU

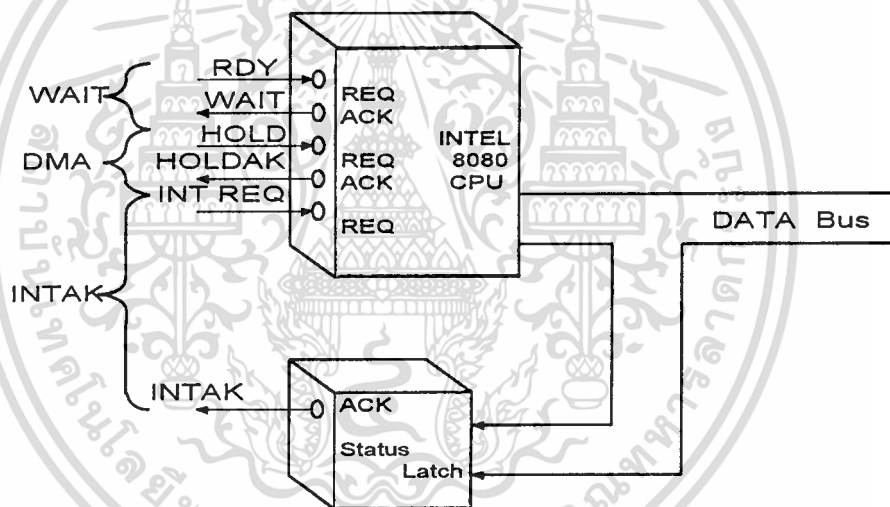


รูปที่ 2.11 แสดงการติดต่อระหว่างอุปกรณ์ภายนอกกับหน่วยความจำโดยผ่านทาง CPU

เหตุที่ต้องใช้ DMA ในกรณีที่ไมโครโปรเซสเซอร์กำลังจัดการข้อมูลอยู่นั้น มักจะอ่านข้อมูลจากอุปกรณ์ภายนอกเข้าไปเก็บในหน่วยความจำ และนำข้อมูลจากหน่วยความจำไปให้แก่เอกสารนี้เป็นเอกสารที่สว่นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ภายนอกวิธีการหนึ่งที่ไม่โครโปรเซสเซอร์จะสามารถกระทำดังกล่าวได้ก็โดยการอินเทอร์รัพท์ตามที่ได้กล่าวมา แต่การอินเทอร์รัพท์เป็นการโอนย้ายข้อมูลผ่านตัว CPU ประการแรกเป็นการใช้ความสามารถส่วนน้อยมากของ CPU ซึ่งมีความสามารถมากกว่าการโอนย้ายข้อมูล ประการที่สองต้องเสียเวลาในการย้ายข้อมูลเข้าไปใน CPU แล้วย้ายออก CPU ทำให้อัตราการโอนย้ายตามที่ไมโครโปรเซสเซอร์ต้องการจะต้องปรับพอยเตอร์ (Pointers) รีจิสเตอร์ (Registers) และตรวจสอบสถานะบิต (statusbit) สำหรับการโอนย้ายแต่ละครั้งอัตราการโอนย้ายจะยิ่งต่ำลงไปอีก ในไมโครโปรเซสเซอร์ทั่วไปการโอนย้ายข้อมูลหนึ่งไบต์จะใช้เวลา 20 ถึง 50 ไมโครวินาที ดังนั้นอัตราการโอนย้ายข้อมูลจะต่ำมากประมาณ 10 กิโลไบต์ต่อวินาทีถึง 50 กิโลไบต์ต่อวินาทีซึ่งนับได้ว่าช้ามาก

รูปแบบโครงสร้าง DMA ของไมโครโปรเซสเซอร์เบอร์ 8080



รูป 2.12 แสดงการขอและการตอบรับสัญญาณ WAIT , HOLD และอินเทอร์รัพท์ของ CPU 8080

การทำงาน DMA ไมโครโปรเซสเซอร์ จะทบทวนการทำเวท (WAIT) และอินเทอร์รัพท์เป็นการเปรียบเทียบกับสัญญาณของการรีเควสท์ (Request) นั้นมาจากวงจรภายนอก และรับสัญญาณแอคโคโนลิจ (Acknowledge) ดังรูปที่ 2.12 สำหรับไมโครโปรเซสเซอร์การทำงานแต่ละคำสั่งจะประกอบด้วย 5 ไชเคิลเพื่อโอนย้ายข้อมูลระหว่าง CPU กับระบบภายนอก บิตของสเตตัสที่ได้ก่อนแต่ละไชเคิลจะบอกหน้าที่ที่จะต้องทำในไชเคิลนั้น แต่ละไชเคิลจะประกอบด้วย 3 ถึง 5 สเตต (State) เช่นสภาพแสดงแอดเดรส (Address presentation state) สภาพสุ่ม (Sample state)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สภาพการโอนย้ายข้อมูล (Data transfer state) สภาพไมโครโปรโพรเซสข้อมูลภายใน (Internal dataprocessing state) ซึ่งแต่ละชนิดต้องมีการเลือกใช้ที่เหมาะสม



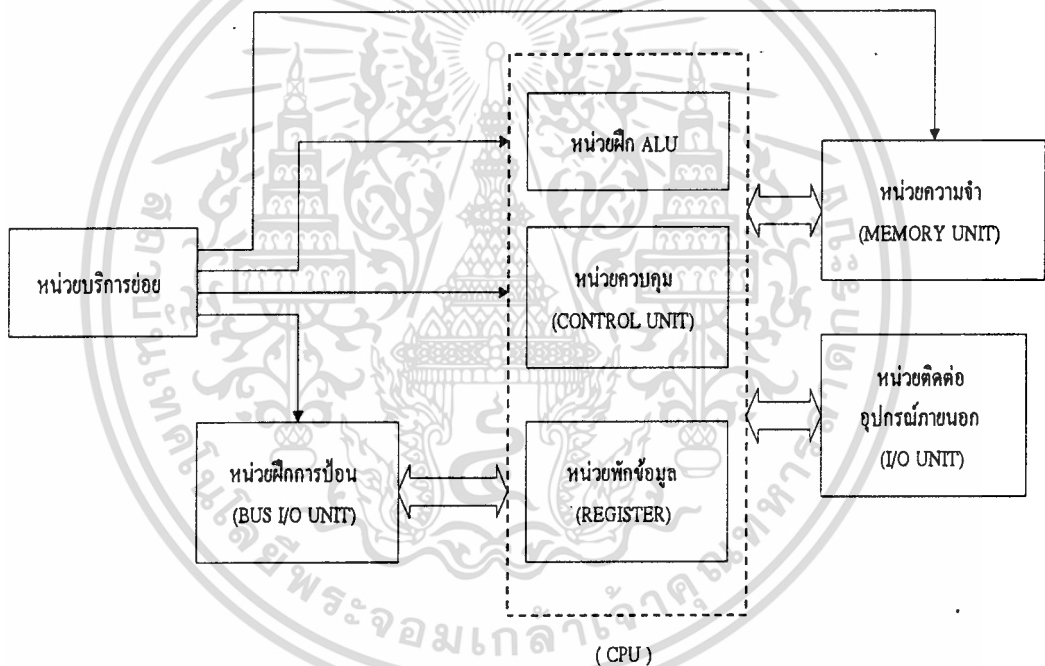
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบชุดฝึกระบบปฏิบัติการไมโครโปรเซสเซอร์

3.1 หลักการออกแบบ

ชุดฝึกระบบปฏิบัติการไมโครโปรเซสเซอร์จะทำให้ผู้ที่เรียนสามารถมองเห็นภาพพื้นฐานของระบบไมโครโปรเซสเซอร์ สามารถฝึกการทำงานของตัวไมโครโปรเซสเซอร์จำลองนี้ได้ด้วยตัวเองโดยแยกฝึกเป็นหน่วยต่างๆเพื่อให้เข้าใจเกิดทักษะในการเข้าถึงองค์ประกอบของแต่ละหน่วยที่เป็นพื้นฐานสำคัญ บล็อกไดอะแกรมในรูปที่ 3.1 แสดงองค์ประกอบพื้นฐานของชุดฝึกนี้



รูปที่ 3.1 แสดงบล็อกไดอะแกรมโครงสร้างของชุดฝึกระบบปฏิบัติการไมโครโปรเซสเซอร์

3.1.1 หน่วยฝึกประมวลผลทางคณิตศาสตร์

ในการออกแบบเพื่อให้ผู้ใช้งานสามารถรู้ความหมายของหน่วยนี้ โดยจะทำการฝึกดังนี้

- การเขียนข้อมูลลงแอดเดรสและเดอเรก
- การทำงานของคำสั่งคณิตศาสตร์ และการทำงานทางลอจิกพื้นฐาน
- การวิเคราะห์ผลที่เกิดขึ้นจากการทดลอง

โดยมีส่วนประกอบดังนี้

- สวิตช์ (SW) S_0, S_1, S_2, WRA, EXE
- LED $S_0, S_1, S_2, WRA, EXE, RDA$

3.1.2 หน่วยควบคุม

เป็นหน่วยที่ทำหน้าที่ควบคุมกระบวนการของคำสั่ง, ขั้นตอนการประมวลผลโดยสามารถควบคุมได้จากสวิตช์ (Switch) กดติดปล่อยดับและ LED แสดงผลของกระบวนการที่จะเกิดขึ้นในการใช้งานจะมีส่วนหนึ่งที่ต้องใช้ร่วมกับหน่วยฝึกการป้อนข้อมูล (BUS I/O UNIT)

3.1.3 หน่วยพักข้อมูล

เป็นรีจิสเตอร์สำหรับเก็บค่าการทำงานของคำสั่งที่เกิดขึ้นโดยมีรีจิสเตอร์ A, F, B, SP, PC , ดาต้าบัฟเฟอร์ (Data Buffer), แอดเดรสบัฟเฟอร์ (Address Buffer), IR ซึ่งรีจิสเตอร์ทั้งหมดที่กล่าวมานี้เป็นรีจิสเตอร์ขนาด 8 บิต

3.1.4 หน่วยความจำ

มีขนาดของหน่วยความจำในการเขียนอ่านข้อมูลแรมเท่ากับ 256 ไบต์ (00H ถึง FFH) โดยมีสวิตช์เลือกตำแหน่งหน่วยความจำว่าจะเป็นรอมหรือแรม

3.1.5 หน่วยติดต่ออุปกรณ์ภายนอก

ใช้ฝึกการส่งข้อมูลหรือรับข้อมูลกับอุปกรณ์ภายนอกโดยมีคอนเน็คเตอร์ต่อขนาด 8 บิต ทั้งอินพุตพอร์ท (8 ตัว) และเอาต์พุตพอร์ท (8 ตัว)

3.1.6 หน่วยฝึกการป้อนข้อมูล

ใช้ฝึกกระบวนการติดต่อข้อมูลผ่านทางสายข้อมูล (Data Bus) โดยสามารถที่จะติดต่อกับ หน่วยความจำ, หน่วยอินพุตเอาต์พุตตามไทม์ทิงไดอะแกรม (Timing diagram) ประกอบด้วย

- สวิตช์คอนโทรลสัญญาณ (RD, WR, IORQ, INT, NMI, CLK, WAIT)
- สวิตช์แอดเดรส A_0-A_7 และสวิตช์ข้อมูล D_0-D_7

3.1.7 หน่วยบริการย่อย

เป็นชุดบริการอำนวยความสะดวกกับหน่วยที่เกี่ยวข้องโดยประกอบด้วย

- คีย์บอร์ดแมทริก
- สวิตช์ RUN, STEP และ SPEED

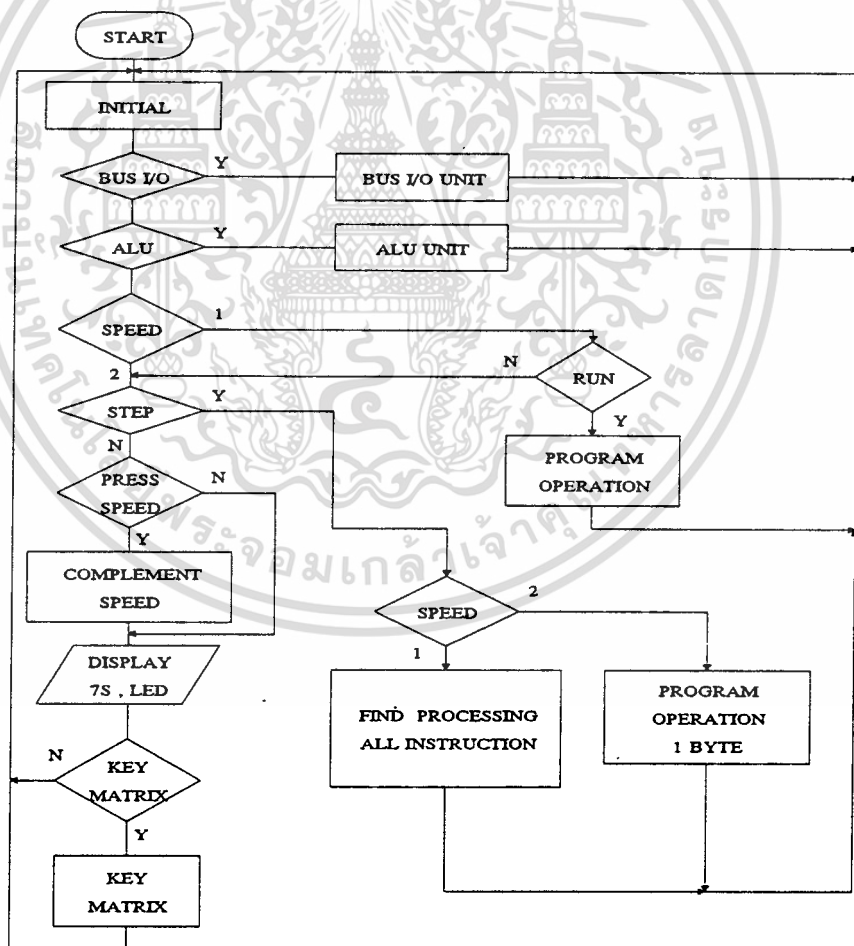
- สวิตช์เลือกใช้งานหน่วย ALU, ฝึกการป้อนข้อมูล

สวิตช์ SPEED จะเป็นตัวเลือกระบบการประมวลผลอย่างหยาบหรือละเอียดหากเป็นแบบหยาบ หมายถึงประมวลผลทีละ 1 ไบต์ เพื่อผลลัพธ์ที่สำคัญโดยรวม ส่วนแบบละเอียดหมายถึงจะกระจายใน 1 ไบต์คำสั่งว่าประกอบด้วยกระบวนการอะไรบ้าง ซึ่งมีจำนวนคำสั่งจำกัดเฉพาะในโปรแกรมการทดลอง (น้อยกว่าการประมวลผลอย่างหยาบ) สามารถใช้ได้เฉพาะสวิตช์ STEP เท่านั้น

3.2 โครงสร้างการทำงาน

จากการอธิบายโครงสร้างที่กำหนดขึ้นจึงได้ผังงานซึ่งจะเป็นตัวกำหนดรูปแบบของโปรแกรมดังต่อไปนี้

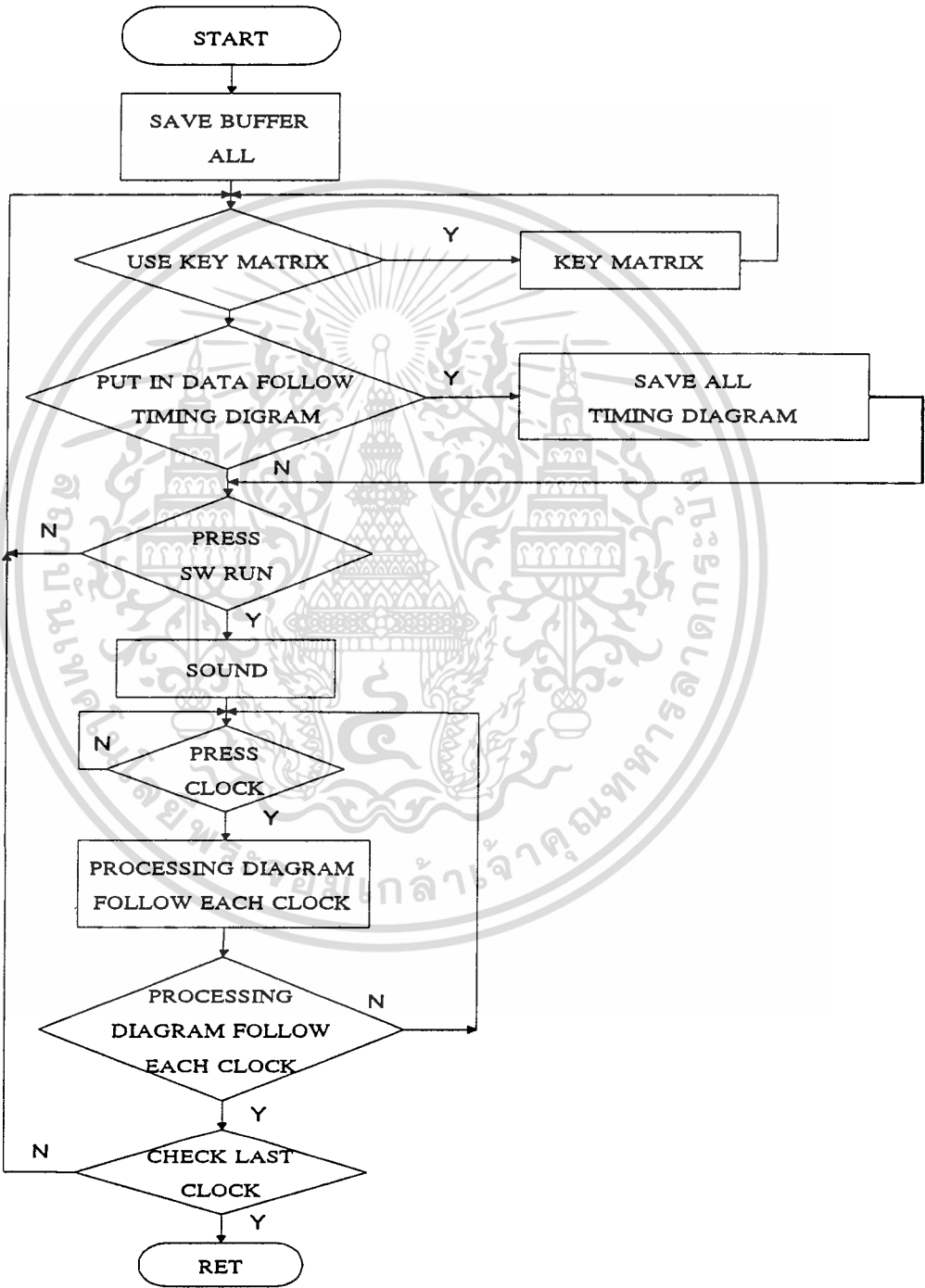
3.2.1 ผังการทำงานของเครื่องโดยรวม



รูปที่ 3.2 แสดงผังการทำงานของเครื่องโดยรวม

ในขณะที่เปิดเครื่องจะอยู่ในส่วนการประมวลผลกลาง คือพร้อมที่จะแปลความหมายคำสั่งที่หน่วยความจำเก็บไว้พร้อมที่จะรับสวิตช์ RUN และสวิตช์ STEP ทันที

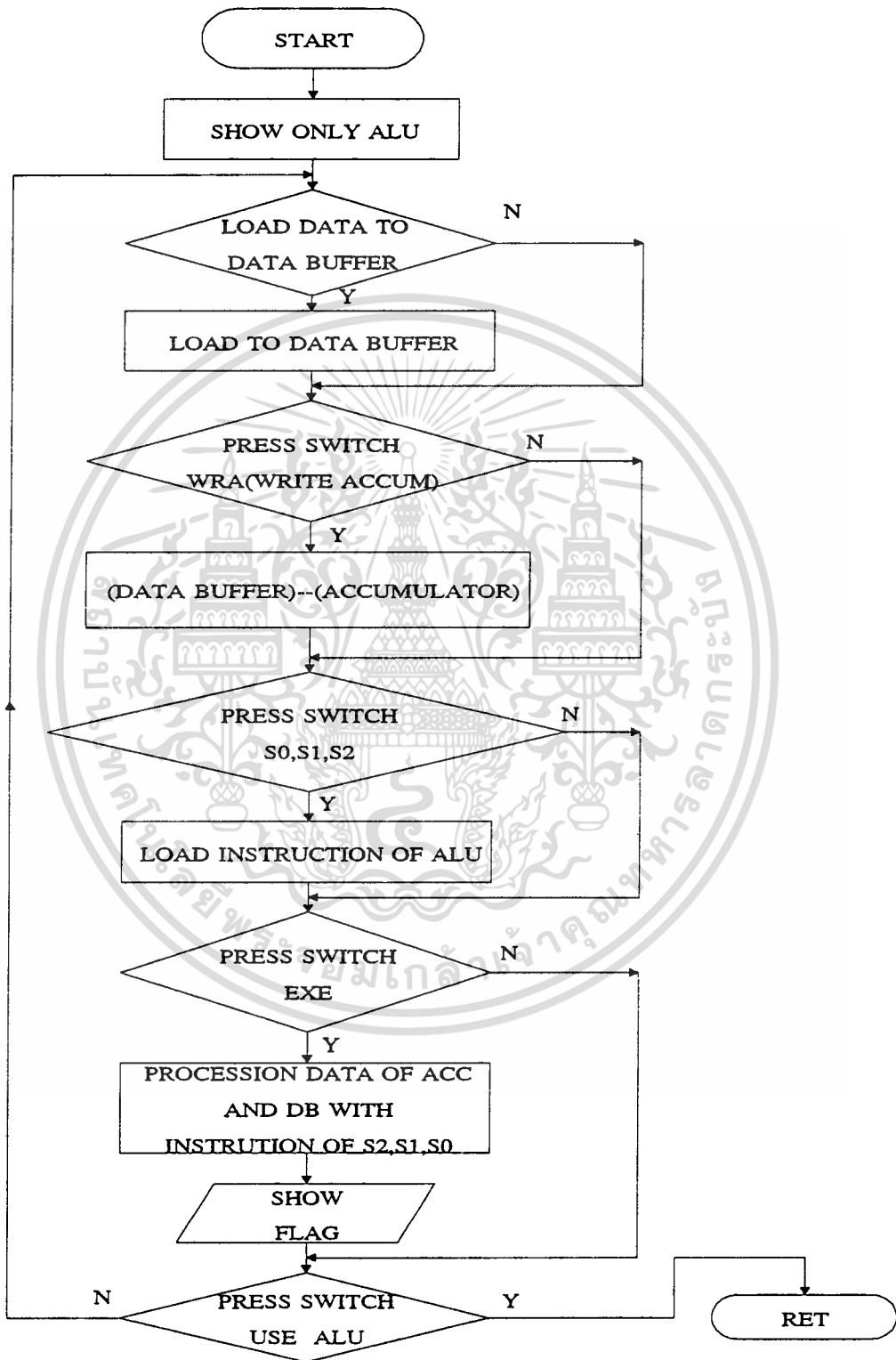
3.2.2 ฟังก์การทำงานหน่วยบัสอินพุตเอาต์พุต



รูปที่ 3.3 แสดงฟังก์การทำงานหน่วยบัสอินพุตเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 ฟังก์ชันหน่วยประมวลผลทางคณิตศาสตร์



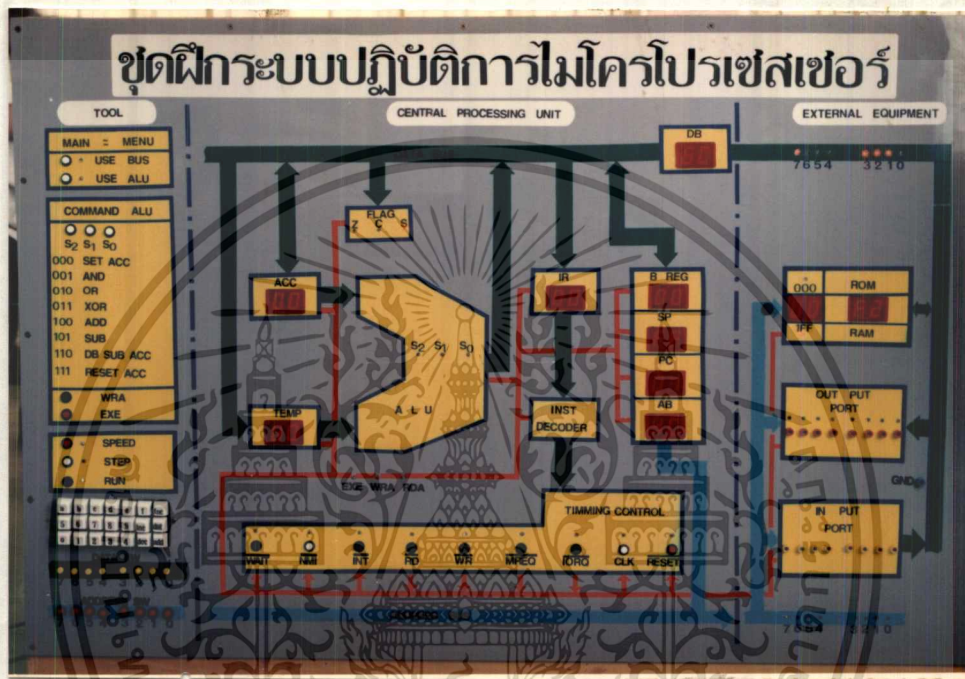
รูปที่ 3.4 แสดงฟังก์ชันประมวลผลทางคณิตศาสตร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การออกแบบส่วนโปรแกรม

จากที่กล่าวถึงสภาพทั่วไปของเครื่อง จากนั้นไปจะกล่าวถึงขั้นตอนในการออกแบบของส่วนโปรแกรมโดยใช้ผังการทำงานของเครื่องโดยรวมอ้างอิงในการอธิบาย

ขณะเริ่มเปิดเครื่องนั้นการทำงานของโปรแกรมจะอยู่ในส่วนพร้อมที่จะประมวลผลตามโปรแกรมเข้าสู่หน่วยบัสนิพพุตเอาต์พุต, เข้าสู่หน่วย ALU , พร้อมทั้งจะรับคีย์ป้อนข้อมูล



รูปที่ 3.5 แสดงหน้าเครื่องเมื่อเปิดเครื่อง

หากโปรแกรมจะอยู่ที่เลเบล (Label) ชื่อ "START_MAIN" ในการประมวลผลตามโปรแกรม หรือ หน่วยประมวลผลกลาง เครื่องพร้อมที่จะรับสวิตช์ SPEED , STEP และ RUN โดยแบ่งการประมวลผลเป็น 4 แบบคือ

แบบที่ 1 ประมวลผลทีละไบท์จะประมวลผลคำสั่งต่างๆ ให้สามารถมองเห็นถึงการเพิ่มของโปรแกรมเคาท์เตอร์ (Program Counter ; PC) ,แอดเดรสบัฟเฟอร์ และหน่วยแสดงผลอื่นที่ประกอบด้วยดาต้าบัฟเฟอร์,แอดคิวมูลเตอร์ ,เทมปี (Temp) , รีจิสเตอร์ B , แสตคพอยท์เตอร์และอินสตรักชันรีจิสเตอร์

แบบที่ 2 เป็นการประมวลผลทีละไบท์แบบต่อเนื่อง คือไม่จำเป็นต้องรอการกดสวิตช์ STEP เหมือนแบบแรก แบบต่อเนื่องจะทำงานโดยการกดสวิตช์ RUN ค่าของโปรแกรมเคาท์เตอร์

จะมีการเปลี่ยนแปลงไปตลอดประโยชน์ในรูปแบบนี้คือจะทำให้ทราบผลการทำงานของโปรแกรมที่ป้อนที่ต้องการความต่อเนื่อง

แบบที่3 เป็นการประมวลผลคำสั่งแบบละเอียดทีละขั้นตอน ซึ่งจะทำให้มองเห็น สถานะของการอ่านคำสั่ง,แปลความหมาย (สถานะ Fetch) และการปฏิบัติตามคำสั่งที่แปลความหมายมา (สถานะ Execute)

เช่น คำสั่ง 3E n = LD A,n มีขั้นตอนดังนี้

STEP	1	2	3	4	5	6	7	8	9
DB	XX	XX	3E	3E	3E	3E	CC	CC	CC
PC	01	01	01	01	02	02	02	02	02
AB	00	00	00	01	01	01	01	01	02
RD									
WRA									
D0-D7									
A0-A7									

ตารางที่ 3.1 ผลการทดลองคำสั่ง LD A,CC

ช่วงเฟิร์ทคือตั้งแต่สเต็ปที่ 1 ไปจนถึงสเต็ปที่ 4 ค่า AB และ PC จะเท่ากัน ค่าใน IR จะมีรหัสคำสั่ง 3EH เก็บอยู่และแปลความหมายออกมา หลังจากนั้นจะนำข้อมูลที่ตำแหน่งแอดเดรสถัดไปเข้ามาเขียนลงในแอดเดรสสำหรับทางโปรแกรมสามารถเขียนได้ดังนี้

I3EN:

CALL I3E3

;MOV A,n EXECUIT

I3E8:

LD BC,I3E8+6

JP STP2

LD A,(B_DB)

LD (B_SW_DATA),A

```

        PUSH    AF            ;SAVE (B_DB)---/
        LD      HL,B_SW_COMM2
        SET     5,(HL)        ;WRA=1
I3E9:   LD      BC,I3E9+6
        JP      STP2
        POP     AF            ;MOV (B_DB)---\
        LD      (B_ACC),A      ;** MOV A,n
        CALL    DEC_7S
        LD      A,0           ;COMMAND CLR 'WRA'
;FIND COMMAND OF (STP2_COM)
;AND OPERATE COMMAND
        CALL    STP2_COMD     ;OPERATE COMMAND
        JP      CPC_I         ;<< END 3E >>
;((( ***** )))
I3E3:   LD      BC,I3E3+6
        JP      STP2
        LD      A,(B_PC)       ;<<< PC=AB >>>
        LD      (B_AB),A
I3E4:   LD      BC,I3E4+6
        JP      STP2
        LD      A,(B_PC)
        INC     A              ;<<< PC+1 >>>
        LD      (B_PC),A
        LD      HL,SYS_STP2
        SET     0,(HL)         ;INSTR. 2 BYTE
I3E7:   LD      DE,I3E7+6
        JP      RDI_PDC
        RET

```

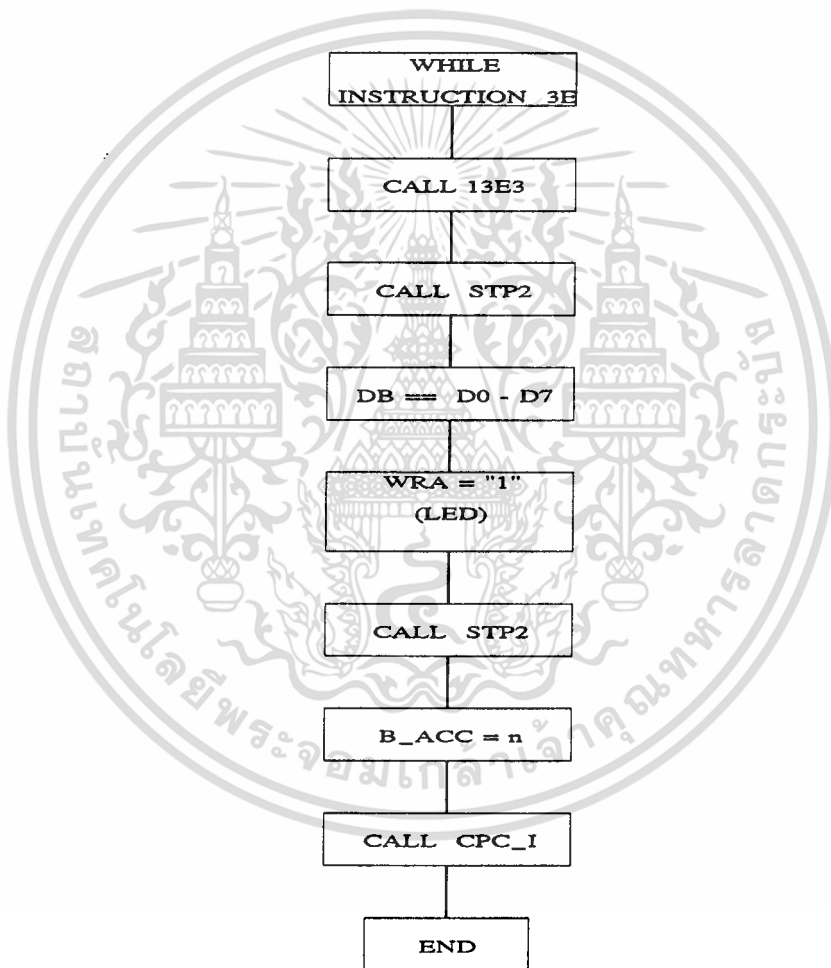
ในการที่จะจับพีไปหาโปรแกรมย่อยของคำสั่ง เช่น I3EN (LD A,n) ทุกคำสั่งจะต้องผ่าน
 เอกสาร RDI_PDC (Read Data Instruction Push Data To DB and Clear LED D₀-D₇, A₀-A₇)
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CONTROL) โดยรอสวิตช์ STEP จากนั้นจะอ่านข้อมูลจากหน่วยความจำที่เข้ามานำเก็บไปที่ DB ขั้นตอนต่อมาถ้ารอรับสวิตช์ STEP จากนั้นข้อมูลจาก DB ไปเก็บที่ IR

I3E3 คือโปรแกรมย่อยทำงานในช่วงอ่านคำสั่งที่เข้ามาโดยจะต้องกดสวิตช์ STEP 2 ครั้ง จึงจะออกจากโปรแกรม

STEP ที่ 2 คือถ้าหากกดสวิตช์ STEP จึงจะออกจากโปรแกรม

CPC_I คือรอกการกดสวิตช์และทำการนำค่า PC ไปใส่ AB รอกกดสวิตช์ STEP เพื่อทำการเพิ่มค่า $PC = PC + 1$



รูปที่ 3.6 แสดงผังการทำงานของคำสั่ง 3E

แบบสุดท้ายเป็นการประมวลผลแบบละเอียดอย่างต่อเนื่องมีหลักการเขียน, การออกแบบ

เหมือนกับแบบละเอียดแต่ไม่ต้องรอกการกดสวิตช์ STEP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

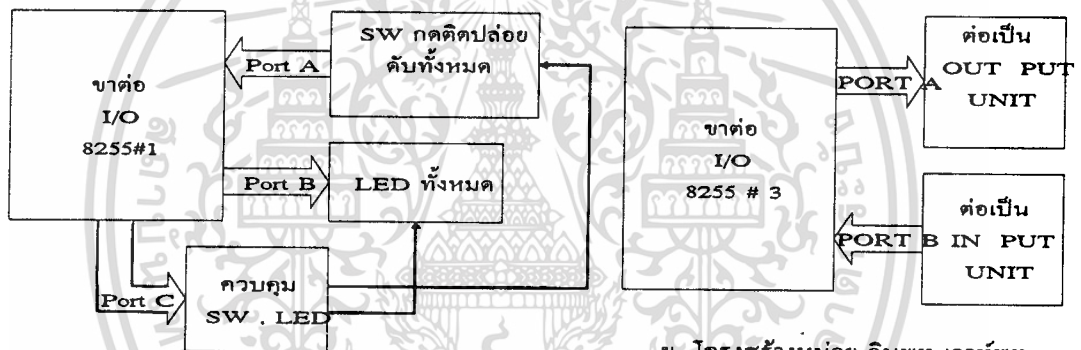
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบส่วนวงจร

เลือกใช้ COTROL PACK CP-Z80 V3 ของบริษัท ETT เนื่องจากว่าผู้ทำหน้าที่เขียนโปรแกรมมีความคุ้นเคยกับการใช้งาน Single Board ชุดนี้

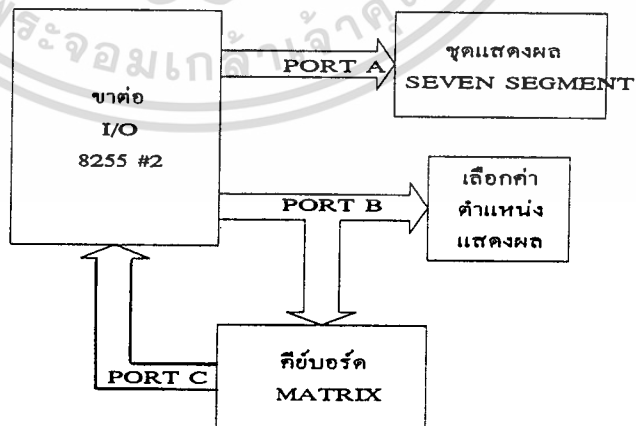
จากวงจรภายในบล็อก CP-Z80 V3 นำที่ใช้งานมาดังนี้

- ส่วนตัว CPU เบอร์ Z80
- ความถี่ที่ใช้งาน 3.579 เมกกะเฮิร์ซ
- หน่วยความจำ รวม 32 กิโลไบต์
- หน่วยความจำ แรม 8 กิโลไบต์
- พอร์ตที่ใช้งาน 8255 3 ตัว



ข. โครงสร้างหน่วย อินพุต เอาท์พุต

ก. โครงสร้างส่วยสวิทซ์ใช้งาน

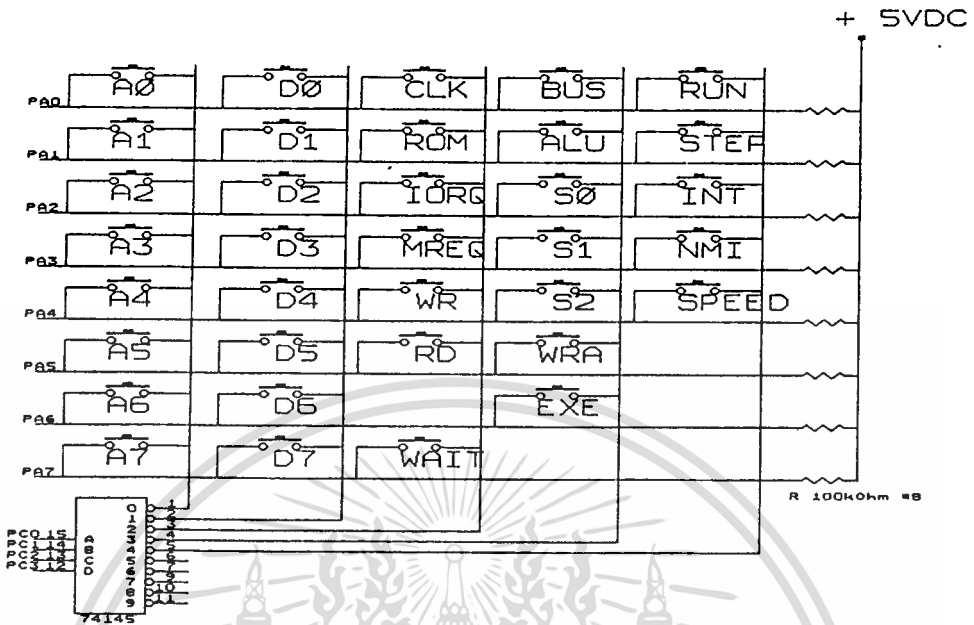


ค. โครงสร้างส่วนแสดงผลตัวเลขและคีย์บอร์ดป้อนข้อมูล

รูปที่ 3.7 แสดงการใช้พอร์ตที่ใช้งาน 8255 ทั้ง 3 ตัว

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ การเผยแพร่โดยไม่ได้รับอนุญาตถือว่าผิดกฎหมาย ผู้ที่นำเอกสารนี้ไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

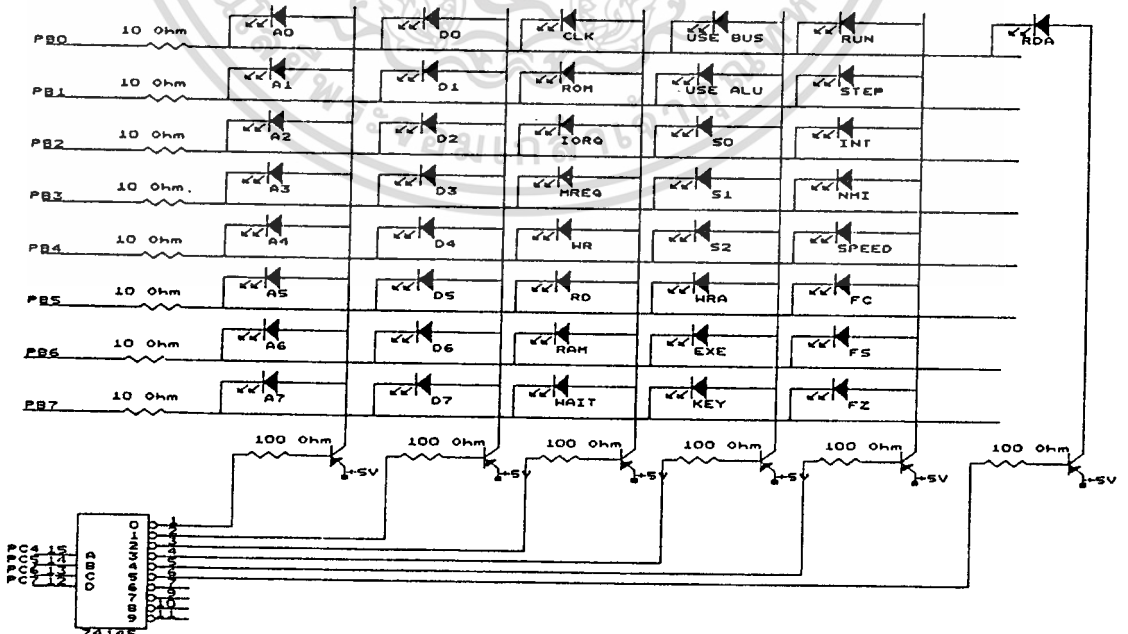
3.5 การออกแบบวงจรส่วนสวิทช์



รูปที่ 3.8 แสดงวงจรส่วนสวิทช์

ใช้หลักการสแกน ในการอ่านข้อมูลพอร์ท A และส่วนคอยล์น้สวิทช์จะใช้พอร์ท C การจัดวางจะให้ตำแหน่งสวิทช์ตรงกับตำแหน่ง วงจรส่วน LED เพื่อสะดวกในการเขียนโปรแกรม

3.6 การออกแบบวงจรส่วน LED



รูปที่ 3.9 แสดงวงจรส่วน LED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ในเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

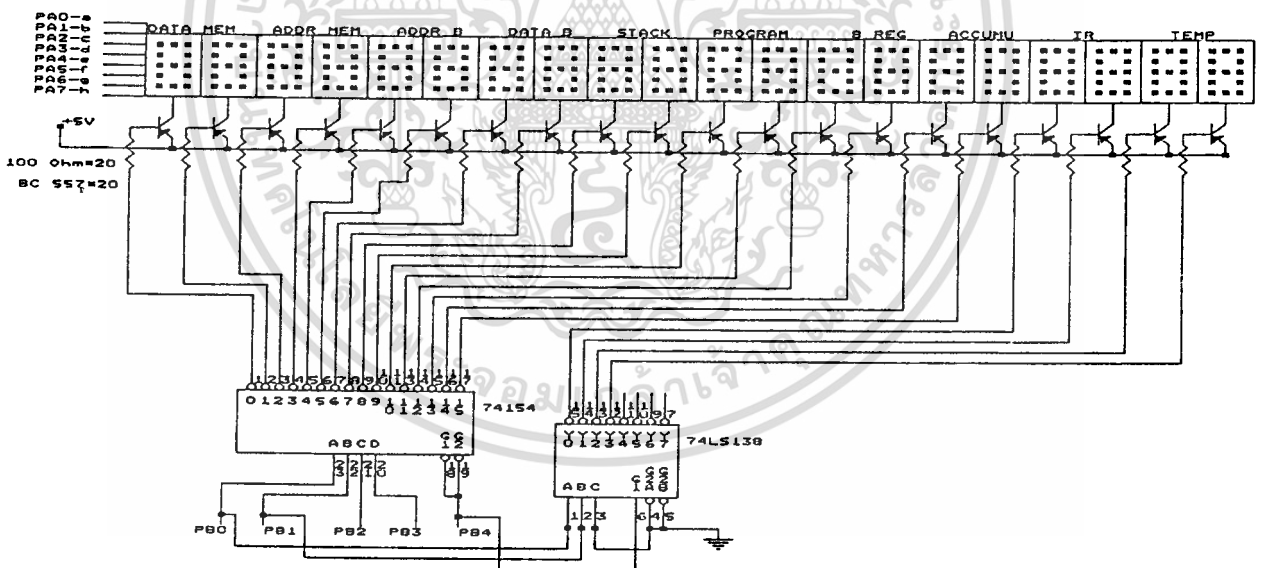
-ส่วนแสดงผล LED มีลักษณะเหมือนกับส่วนสวิทช์โดยจะใช้ TR#BC557 มาทำการขยายสัญญาณ (DRIVE)

-ใช้ขาจากพอร์ท C ด้านบน มาทำการถอดรหัสโดย 74LS145 ดังรูปที่ 3.9 ทำให้วงจรสามารถขยาย LED ได้อีก 39 ดวง จากเดิมที่มีอยู่แล้ว 41 ดวง

-ลักษณะการจัดไบอัสให้ TR#BC557 อิ่มตัว (SATURATE) คือจ่ายกระแสเต็มๆ เพื่อให้มีแรงดันไปขับ LED ให้สว่างเพียงพอกับการสแกนที่รวดเร็ว

3.4.8 การออกแบบชุดตัวเลขแสดงผล 7 หลัก (Seven segment)

-ในการออกแบบชุดตัวเลขแสดงผลจากรูปใช้ตัวเลขแสดงผลเป็นชนิดคอมมอนคาโทดและใช้ 74LS154, 74LS138 มาช่วยในการถอดรหัสสัญญาณให้ออกมาควบคุมมากกว่า 16 คอลัมน์ โดยในแต่ละคอลัมน์ใช้ TR#BC557 เป็นตัวขยายโดยใช้ขา PB4 จากพอร์ท B 8255 (ตัวที่2) เป็นตัวควบคุม 74LS154, 74LS138 ในส่วนของโปรแกรม SEVEN SEGMENT แต่ละชุด (2หลัก) จะมีบัพเฟออร์รองรับทั้งหมด 10 ตัว ใช้สำหรับปรับเปลี่ยนข้อมูล เช่น B-ACC, B-BR, B-DB เป็นต้น

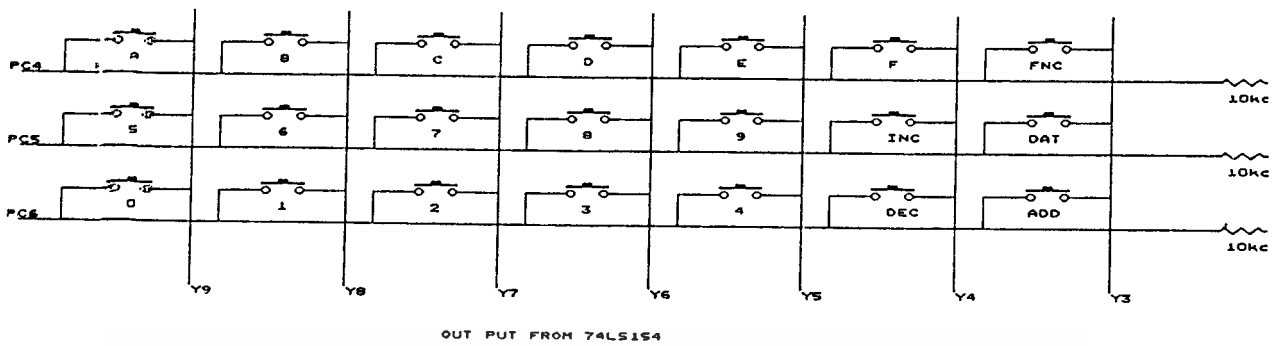


รูปที่ 3.10 แสดงวงจรส่วนชุดตัวเลขแสดงผล 7 หลัก

3.7 การออกแบบเมทริกคีย์

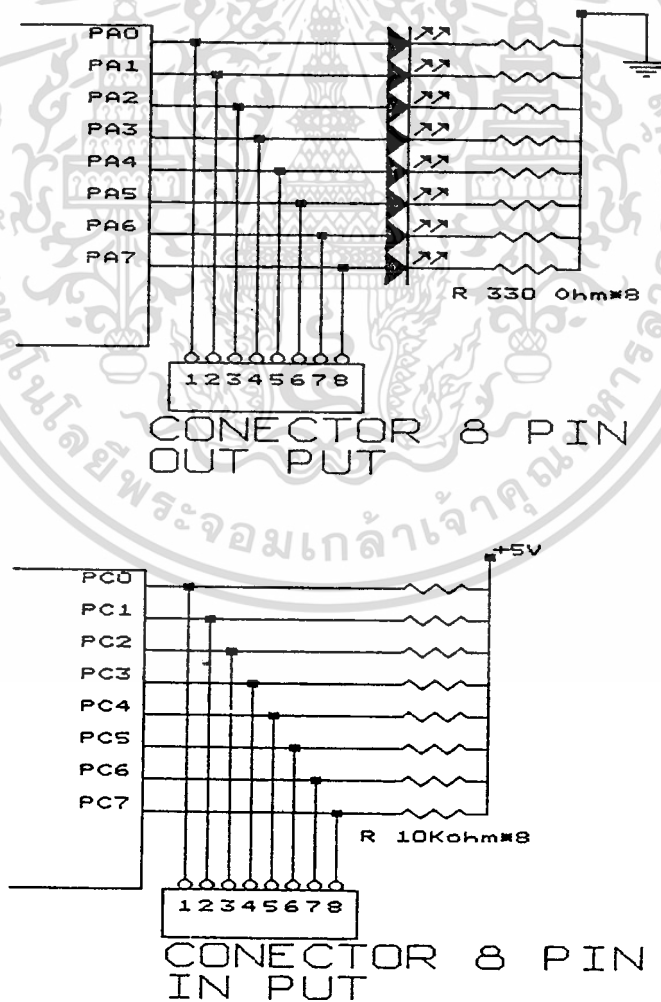
-ส่วนของเมทริกคีย์ใช้ประโยชน์จาก O/P ของ 74154 คือ Y3-Y9 ทำหน้าที่สแกนทางคอลัมน์และ PORT C (PC4, PC5, PC6) ของ 8255(2) เป็นส่วนของ PORT INPUT KEY ที่ใช้งานมีทั้งหมด 21 KEY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงวงจรส่วนเมทริกซ์

3.8 ส่วนติดต่ออุปกรณ์ภายนอก



รูปที่ 3.12 แสดงวงจรส่วนติดต่ออุปกรณ์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-ส่วนของเอาท์พุทจะใช้พอร์ท A โดยที่อุปกรณ์ที่จะมาติดต่อสามารถใช้สายต่อและนำไปใช้งานได้ ส่วนของพอร์ทอินพุทจะใช้พอร์ท C ทำหน้าที่อ่านข้อมูล



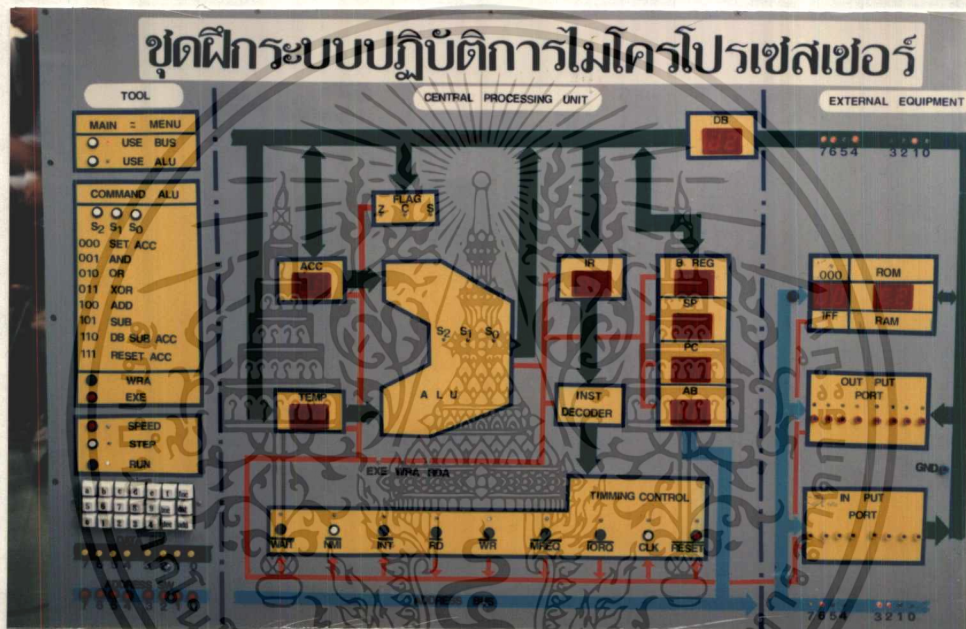
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 การทดลองการทำงานของหน่วยการป้อนข้อมูลตามไทม์มิงโคอะแกรม

4.1.1 อุปกรณ์ส่วนต่างๆที่เกี่ยวข้องกับการทำงานของหน่วยการป้อนข้อมูล



รูปที่ 4.1 แสดงอุปกรณ์ส่วนต่างๆของหน่วยการป้อนข้อมูล

1. สวิตช์ของเมนเมนู (Main Menu) จะทำการเลือก USE BUS เพื่อให้เห็นการทำงานของหน่วยป้อนข้อมูล
2. สวิตช์ในบล็อกของไทม์มิง TIMING CONTROL นี้จะมีการยกเว้นการอินเตอร์รัพท์แบบไม่มีเงื่อนไข (Non-Maskable Interrupt: NMI) ,และ อินเตอร์รัพท์ (Interrupt: INT) โดยการป้อนตามไทม์มิงโคอะแกรม ของคำสั่งโดยการป้อนตามขั้นตอนในใบงาน
3. สวิตช์ค่าด้าบัส (Data bus: DB) ใช้ในการป้อนข้อมูลตามต้องการ
4. ใช้หน่วยแสดงผล 7 ส่วน AB , DB และส่วนแสดงข้อมูลของข้อมูล และแอดเดรส
5. สวิตช์เลือกใช้รอมหรือแรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6.บล็อคของเอาต์พุตพอร์ทเป็นส่วนที่แสดงข้อมูลที่เราจะติดต่อกับอุปกรณ์ภายนอกหรือแสดงข้อมูลที่เราส่งออกมา

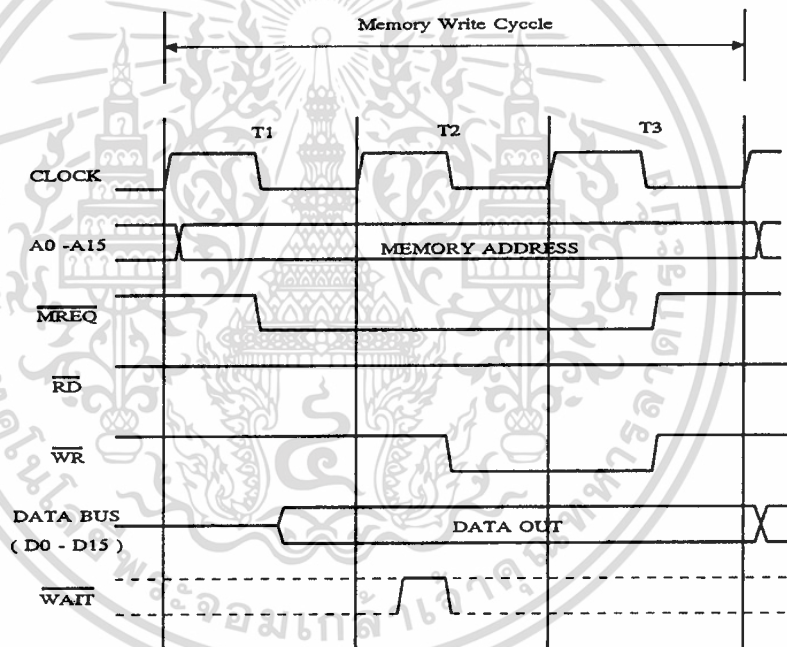
7.บล็อคของอินพุตพอร์ทแสดงข้อมูลที่ได้มีการติดต่อมาจากภายนอก

8.เมมทริกคีย์

4.1.2 ตัวอย่างการทดลองในหน่วยการป้อนข้อมูลบัสดำเนินการไมโครแอสเซมบลีและการทดลอง

กระบวนการอ่านและเขียนข้อมูล

การเขียนข้อมูลลงหน่วยความจำ



รูปที่ 4.2 แสดงไทม์มิงไดอะแกรมของการเขียนข้อมูลลงหน่วยความจำ

การอ่านหน่วยความจำนั้น CPU จะใช้เวลาทั้งสิ้น 3 ไชเกิลสัญญาณนาฬิกา T_1 , T_2 , T_3 โดย CPU จะส่งแอดเดรสที่ต้องการอ่านเข้าสู่แอดเดรสบัสก่อนในตอนแรกของเมชีนไชเกิล หลังจากนั้นต่อมาก็จะส่งสัญญาณ $\overline{MREQ}$ และ $\overline{RD}$ ออกมาและทุกครั้งของขาลงในช่วงจังหวะ T_2 CPU จะตรวจสอบว่า $\overline{WAIT}$ ว่าเป็นลอจิก 0 หรือไม่ ถ้าไม่ CPU ก็จะทำงานตามปกติ แต่ถ้าเป็น 0 ก็จะสร้าง T_w เพิ่มขึ้น ช่วงจังหวะที่แอดเดรสและสัญญาณ $\overline{MREQ}$, $\overline{RD}$ ไปยังหน่วยความจำจะมีข้อมูลออกจากหน่วยความจำลงในบัสดำเนินการแล้วแต่ CPU ยังไม่อ่านจังหวะการอ่านของ CPU จะเกิดขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

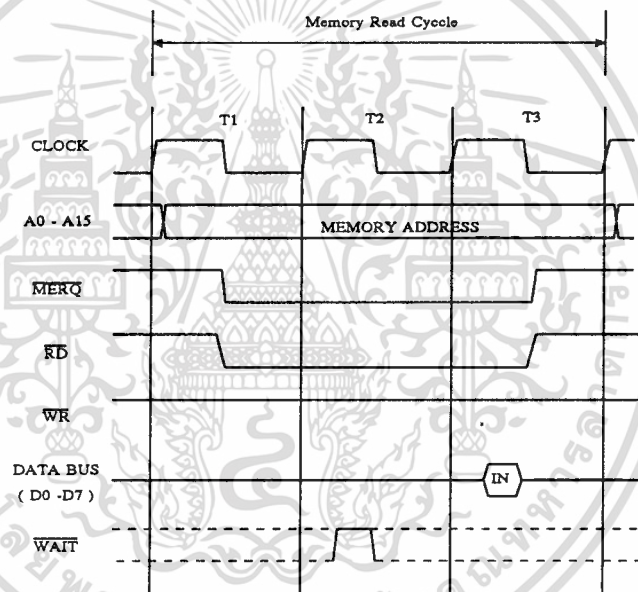
ถ้ามีคำสั่ง WAIT CPU จะรอจนกว่าจะออกจากกระบวนการ WAIT เสียก่อนจึงจะทำงานตามปกติ

10. กคสวิตช์ CLK ลูกที่ 4 (T2 ช่วงลบ) ทุกอย่างจะค้างที่สถานะเดิมยกเว้น $\overline{WR}$ จะพบว่ามีการกระพริบแสดงพร้อมที่จะเขียนข้อมูลลงไป

11. กคสวิตช์ CLK ลูกที่ 5 (T3 ช่วงบวก) ทุกอย่างค้างที่สถานะเดิม

12. กคสวิตช์ CLK ลูกที่ 6 (T3 ช่วงลบ) แรมแอดเดรสที่ AB ซึ่งตำแหน่งจะถูกเขียนลงไปด้วยข้อมูลใน DB เป็นอันจบกระบวนการเขียนแรม

การอ่านข้อมูลจากหน่วยความจำ



รูปที่ 4.8 แสดงไทม์มิงไคอะแกรมของการอ่านข้อมูลจากหน่วยความจำ

ลำดับขั้นตอนการทดลอง

1. จากรูปที่ 4.3 กคสวิตช์ USE BUS เพื่อใช้ระบบบัสของชุดฝึก
2. กำหนดสวิตช์ที่ต้องใช้ทั้ง 2 ชุดคือสวิตช์แอดเดรสและสวิตช์TIMING CONTROL โดยการกำหนดแอดเดรสที่ต้องการจะอ่านจากหน่วยความจำ
3. กำหนดชุดสวิตช์ TIMING CONTROL ที่ต้องใช้โดยดูไทม์มิงไคอะแกรมประกอบการกำหนดถ้าสวิตช์ตัวใดต้องการใช้งานให้กคสวิตช์ตัวนั้น
4. กำหนดสวิตช์ ROM/RAM ตามต้องการโดยจะอ่านข้อมูลจากรอมหรือแรม ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในจังหวะกลางของ T_3 ช่วงการอ่านของ CPU จะใช้เวลาสั้นๆ เพื่อป้องกันการอ่านข้อมูลผิดพลาด คือผิวดำแหน่งแอดเดรสหรืออ่านได้หลายแอดเดรสนั่นเอง

ในทำนองเดียวกันจังหวะของ MW ก็เช่นกันประกอบด้วย T_1 , T_2 และ T_3 (กรณีไม่มี T_w) ในที่นี้ CPU จะสร้างสัญญาณ $\overline{MREQ}$ กับ $\overline{WR}$ ให้ทำงานอย่างไรก็ตาม CPU จะส่งข้อมูลออกไปรอไว้ที่บัซข้อมูลก่อนและยาวนานกว่าพัลส์การเขียน การเขียนข้อมูลจะใช้เวลาเพียงสั้นๆ ในจังหวะที่ข้อมูลมารออยู่ที่บัซเรียบร้อยแล้ว

ในกรณีที่ต้องการหน่วยเวลาการทำงานในแมชีนไซเคิลนี้สามารถทำได้ด้วยการสร้างขา $\overline{WAIT}$ ให้มีค่าเป็น 0 ในจังหวะกลางของของ T_2 ซึ่ง CPU จะสร้าง T_w ขึ้นตรงกับเท่าที่ขา $\overline{WAIT}$ ยังคงเป็น 0 อยู่

ลำดับขั้นตอนการทดลอง

1. กดสวิตช์ USE BUS เพื่อใช้ระบบบัซของชุดฝึก
2. กำหนดสวิตช์ต่างๆที่ต้องใช้ทั้ง 3 ชุดซึ่งได้แก่ชุดข้อมูล, แอดเดรสและชุดสวิตช์ TIMING CONTROL โดยการใส่ข้อมูลและแอดเดรสที่ต้องการจะเขียนลงไป ในหน่วยความจำ
3. กำหนดสวิตช์ TIMING CONTROL ที่ต้องใช้ โดยดูไทม์มิงไคอะแกรมประกอบกับการกำหนดถ้าสวิตช์ตัวใดต้องการใช้งานให้กดสวิตช์ตัวนั้น
4. กำหนดสวิตช์ ROM/RAM ให้อยู่ที่แรม เพราะเป็นการเขียนข้อมูลลงในหน่วยความจำซึ่งจะเขียนลงไป ในรอมไม่ได้
5. กดสวิตช์ RUN
6. ขณะนี้ชุดแสดงผลของ AB และ DB จะแสดงผลข้อมูลของข้อมูลและแอดเดรสที่ต้องการเขียนลงสู่แรม LED จะแสดงผลที่ USE BUS , ใช้แรม
7. กดสวิตช์ CLK 1 ครั้งเมื่อ

ที่ CLK ลูกที่ 1 (T_1 ช่วงบวก) CPU จะทำการส่งค่าแอดเดรสออกไปที่ตำแหน่งแอดเดรสบัซ

AB จะมีค่าตามที่กำหนดสวิตช์

DB จะมีค่าตามที่กำหนดสวิตช์

แรมจะถูกชี้ตำแหน่งตามที่ AB แสดง

8. กดสวิตช์ CLK ลูกที่ 2 (T_1 ช่วงลบ)

$\overline{MREQ}$ จะกระพริบแสดงว่ามีการขอใช้หน่วยความจำ

DATA BUS จะกระพริบแสดงว่ามีการส่งข้อมูลออกไป

9. กดสวิตช์ CLK ลูกที่ 3 (T_2 ช่วงบวก) ทุกอย่างค้างสภาวะเดิมตามรูปไทม์มิงไคอะแกรม

5. กคสวิตช์ RUN

6. ขณะนี้ชุดแสดงผลจะแสดงค่าแอดเดรสที่เราต้องการอ่านส่วนของข้อมูล จะแสดงข้อมูลตามที่สวิตช์ข้อมูลที่ถูกกำหนดไว้ แต่จะไม่มีผลอะไรต่อการทำงานเพราะเป็นการอ่านข้อมูลออกมา ไม่ใช่การเขียนข้อมูลลงไปสวิตช์ ROM/RAM จะแสดงผลที่ใดก็ได้

7. กคสวิตช์ CLK ลูกที่ 1 (T1 ช่วงบวก) แอดเดรสที่ต้องการของรอมหรือแรมจะถูกส่งเข้าสู่แอดเดรสบัส

8. กคสวิตช์ CLK ลูกที่ 2 (T1 ช่วงลบ) $\overline{MREQ}$ และ $\overline{RD}$ จะกระพริบแสดงว่ามีการขอใช้หน่วยความจำโดยให้มีการอ่านหน่วยความจำ

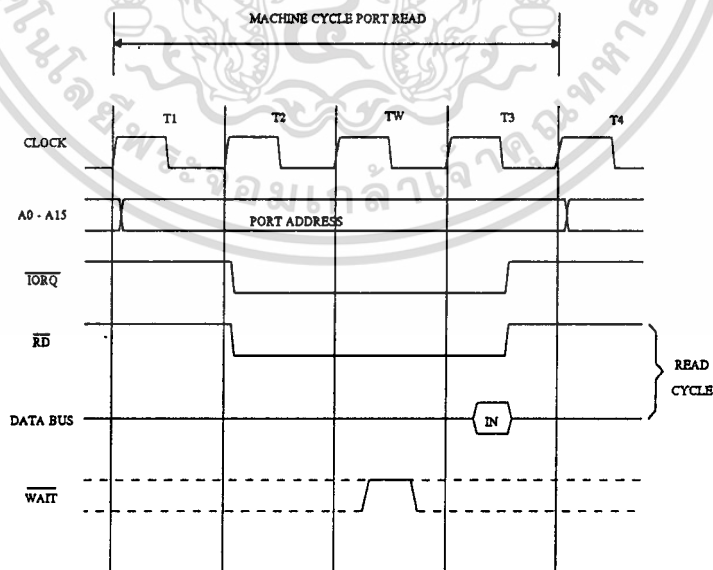
9. กคสวิตช์ CLK ลูกที่ 3 (T2 ช่วงบวก) ทุกอย่างค้างที่สถานะเดิม

10. กคสวิตช์ CLK ลูกที่ 4 (T2 ช่วงลบ) CPU จะเช็คว่ามี $\overline{WAIT}$ หรือไม่ ถ้ามีจะเข้าสู่กระบวนการ WAIT จนกว่าจะออกจากกระบวนการ WAIT จึงจะทำงานต่อ

11. กคสวิตช์ CLK ลูกที่ 5 (T3 ช่วงบวก) ทุกอย่างค้างที่สถานะเดิม

12. กคสวิตช์ CLK ลูกที่ 6 (T3 ช่วงลบ) ข้อมูลที่อยู่ในรอม/แรม ตำแหน่งที่ต้องการจะอ่านจะถูกส่งผ่านทาง DATA BUS ก็จะออกมายังชุดแสดงผลตามไทม์มิงโคอะแกรมเป็นอันจบสิ้นกระบวนการอ่านข้อมูล

การอ่านข้อมูลจากพอร์ทเพื่อนำข้อมูลเข้า



รูปที่ 4.4 แสดงไทม์มิงโคอะแกรมของการอ่านข้อมูลจากพอร์ทเพื่อนำข้อมูลเข้า

ในการเขียนเอาต์พุตและอ่านอินพุตจากพอร์ตก็นำคลั่งกับการเขียนและอ่านหน่วยความจำ โดยการเปลี่ยนสัญญาณ $\overline{MREQ}$ เป็น $\overline{IORQ}$ เท่านั้นไคอะแกรมการเขียนเอาต์พุตและอ่านอินพุตแสดงดังรูปที่ 4.4

ข้อแตกต่างที่เห็นได้ชัดเมื่อเทียบกับการเขียนและอ่านหน่วยความจำอีกข้อหนึ่งคือในการทำงานโดยปกติ CPU จะสร้าง T_w แทรกเข้ามา 1 ลูก โดยอัตโนมัติแม้จะไม่มี $\overline{WAIT}$ ทั้งนี้เพราะช่วงจังหวะ $\overline{IORQ}$ นั้น CPU สร้างในขณะช่วง T_2 และ CPU จะไม่สามารถตรวจสอบจังหวะ $\overline{WAIT}$ ในขณะกลางของ T_2 ได้ การตรวจสอบว่า $\overline{WAIT}$ จะเกิดขึ้นในจังหวะของ T_w แทนนั่นเอง และถ้าหากมีการ $\overline{WAIT}$ CPU ก็จะสร้างสัญญาณ T_w ต่อมาอีก

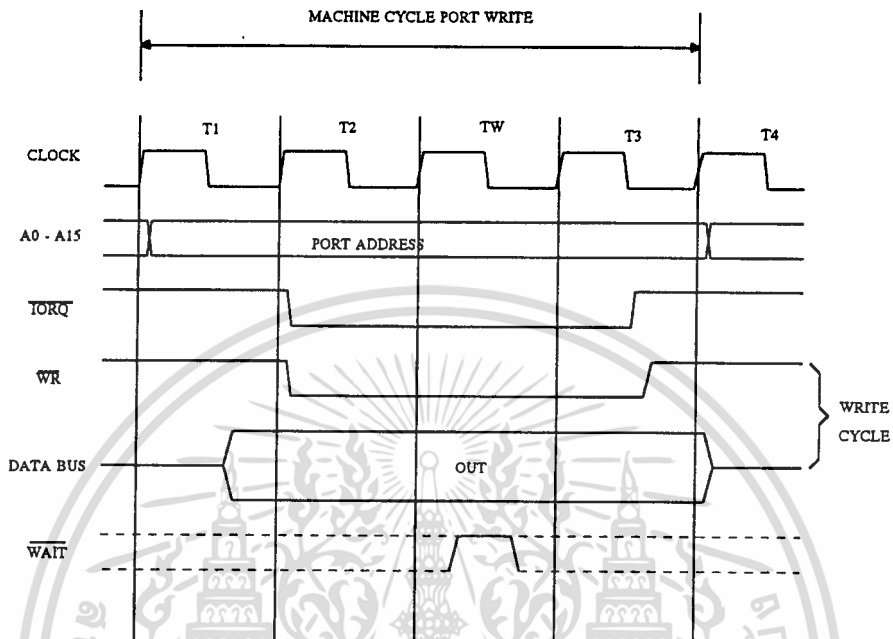
ลำดับขั้นตอนการทดลอง

1. กดสวิตช์ USE BUS เพื่อใช้ระบบบัสของชุดฝึก
2. กำหนดสวิตช์ต่างๆที่ต้องการใช้ทั้ง 3 ชุด ซึ่งได้แก่ชุดพอร์ท , แอดเดรส และชุดสวิตช์ TIMING CONTROL โดยแอดเดรสที่จะใช้ต้องเป็น A2 เท่านั้นเพราะในชุดฝึกใช้พอร์ท A2 เป็นอินพุตพอร์ท
3. กำหนดสวิตช์ TIMING CONTROL ที่ต้องใช้ในการอินพุตโดยดูไทม์มิงไคอะแกรมประกอบ
4. กดสวิตช์ RUN
5. ขณะนี้ชุดแสดงผลของ AB จะแสดงแอดเดรสที่เราต้องการอินพุตเข้ามาคือ A2
6. กดสวิตช์ CLK ลูกที่ 1 (T1 ช่วงบวก) ข้อมูลใน AB จะถูกส่งเข้าสู่แอดเดรสบัส
7. กดสวิตช์ CLK ลูกที่ 2 (T1 ช่วงลบ) ทุกอย่างค้างที่สถานะเดิม
8. กดสวิตช์ CLK ลูกที่ 3 (T2 ช่วงบวก) $\overline{IORQ}$ และ $\overline{RD}$ มีการอ่านข้อมูลเข้ามาที่พอร์ท
9. กดสวิตช์ CLK ลูกที่ 4 (T2 ช่วงลบ) CPU จะเช็คว่ามี $\overline{WAIT}$ หรือไม่ ถ้ามีจะเข้าสู่กระบวนการ WAIT จนกว่าจะออกจากกระบวนการ WAIT จึงจะทำงานต่อ
10. กดสวิตช์ CLK ลูกที่ 5 (T3 ช่วงบวก) ทุกอย่างค้างที่สถานะเดิม
11. กดสวิตช์ CLK ลูกที่ 6 (T3 ช่วงลบ) ข้อมูลที่อ่านเข้ามาจะถูกส่งเข้าไปที่ DATA BUS และออกไปแสดงผลที่ DB เป็นอันจบกระบวนการอินพุต

การส่งข้อมูลออกพอร์ท

วิธีการรับส่งข้อมูลโดยใช้คำสั่งในไมโครโปรเซสเซอร์ ปกติจะมีคำสั่งเฉพาะสำหรับการส่งข้อมูลเมื่อ CPU ทำการ เอ็กซีคิวทีฟจะมีสัญญาณออกจากระบบควบคุมโดยตรง สัญญาณนี้เรียกว่าสัญญาณไอโซเลตอินพุตเอาต์พุต (Isolated I/O) นอกจากนี้ยังมีวิธีของการใช้คำสั่ง เพื่อติดต่อกับอุปกรณ์คือใช้คำสั่งเกี่ยวกับการอ่านเขียนหน่วยความจำซึ่ง CPU จะให้สัญญาณการควบคุมการไม่ว่าการณใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อ่านและการเขียนข้อมูลที่หน่วยความจำสัญญาณนี้สามารถนำมาดัดแปลงใช้เป็นสัญญาณควบคุมอุปกรณ์ได้



รูปที่ 4.5 แสดงไทม์มิงไดอะแกรมของการส่งข้อมูลออกพอร์ต

ลำดับขั้นตอนการทดลอง

- 1.จากรูปที่ 4.5 กดสวิทช์ USE BUS เพื่อใช้ระบบบัสของชุดฝึก
- 2.กำหนดสวิทช์ต่างๆที่ต้องใช้ทั้ง 3 ชุดซึ่งได้แก่ชุดพอร์ตแอดเดรสและชุดสวิทช์ TIMING CONTROL โดยแอดเดรสที่จะใช้ต้องเป็น A0 เท่านั้นเพราะในชุดฝึกใช้พอร์ต A0 เป็นเอาต์พุตพอร์ต
- 3.กำหนดสวิทช์ TIMING CONTROL ที่ต้องใช้การเอาต์พอร์ตโดยดูไทม์มิงไดอะแกรม
- 4.กดสวิทช์ RUN
- 5.ขณะนี้ชุดแสดงผลของ AB จะแสดงแอดเดรสที่ต้องการเอาต์พอร์ตออกไปคือ A0 ส่วน DB จะแสดงข้อมูลที่เรากำลังส่งออกพอร์ต
- 6.กดสวิทช์ CLK ลูกที่ 1 (T1 ช่วงบวก) ABจะส่งค่าแอดเดรสลง แอดเดรสบัส ดังนั้นที่แอดเดรสบัสมีค่าของแอดเดรสของพอร์ตที่เราต้องการส่งข้อมูลออกคือ A0
- 7.กดสวิทช์ CLK ลูกที่ 2 (T1 ช่วงลบ) ข้อมูลในDBจะถูกส่งเข้าสู่DATA BUS
- 8.กดสวิทช์ CLK ลูกที่ 3 (T2 ช่วงบวก) $\overline{WR}$ และ $\overline{TORQ}$ จะกระพริบแสดงว่าจะมีการส่งข้อมูลออกพอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

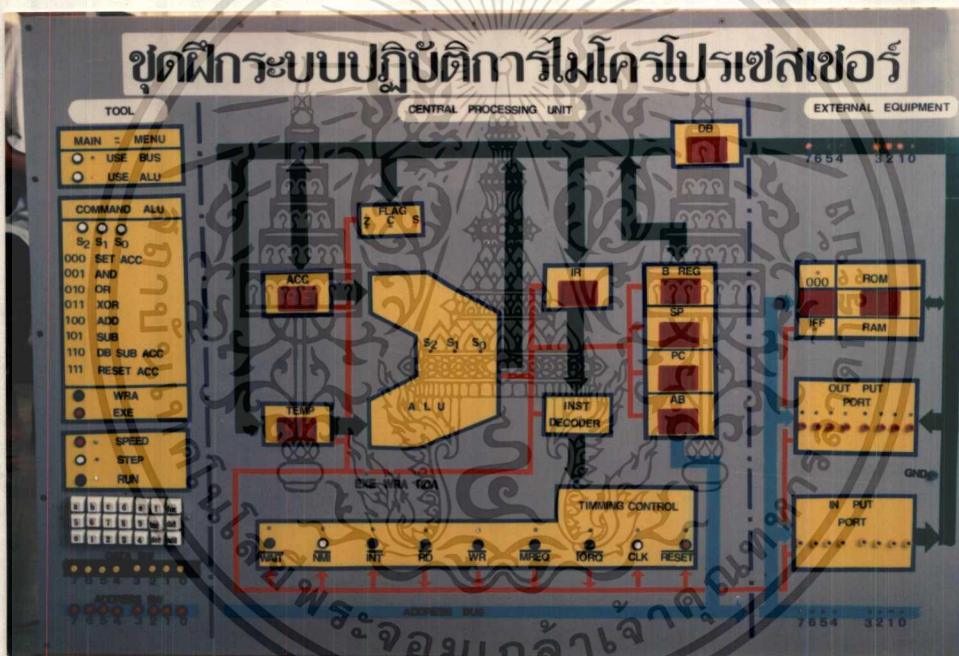
9.กดสวิตช์ CLK ลูกที่ 4 (T2 ช่วงลบ) CPU จะเช็คว่ามี $\overline{\text{WAIT}}$ หรือไม่ถ้ามีจะเข้าสู่กระบวนการ WAIT จนกว่าจะออกจากกระบวนการ WAIT จึงจะทำงานต่อ

10.กดสวิตช์ CLK ลูกที่ 5 (T3 ช่วงบวก) ทุกอย่างค้างที่สถานะเดิม

11.กดสวิตช์ CLK ลูกที่ 6 (T3 ช่วงลบ) ข้อมูลจาก DATA BUS จะส่งออกพอร์ตไปแสดงผลเป็น LED เป็นอันจบกระบวนการการทำงานเอาต์พอร์ท

4.2 การทดลองการทำงานของหน่วยประมวลผลคำสั่งทางคณิตศาสตร์

4.2.1 อุปกรณ์ที่เกี่ยวข้องกับการทำงานของหน่วยประมวลผลคำสั่งทางคณิตศาสตร์



รูปที่ 4.6 แสดงอุปกรณ์ส่วนต่างๆของหน่วยประมวลผลทางคณิตศาสตร์

1. สวิตช์ในบล็อกของ COMMAND ALU โดยที่ S0 , S1 , S2 เป็นฟังก์ชันเลือกการทำงานทางคณิตศาสตร์เช่น Add,Xor สวิตช์ WRA กดเพื่อให้อ่านข้อมูลจาก DB นำค่าไปเก็บไว้ที่ ACC สวิตช์ EXE กดเพื่อให้คำสั่งนั้นๆทำงาน

2.กดสวิตช์ DATA ป้อนข้อมูลตามต้องการ

3.หน่วยแสดงผล 7 ส่วนที่ TEMP , ACC และ DB แสดงผลการทำงานคำสั่งคณิตศาสตร์

4. LED FLAG ใช้แสดงสถานะของ ACC

เอกสารนี้เป็นเอกสารที่ส่ง LED Z ติดแสดงว่า ACC เท่ากับ 0 เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LED C แสดงว่า ACC มีการข้ม

LED S แสดงว่า ตัวเลขใน ACC เป็นลบ

5. LED S0 , S1 , S2 เป็นตัวแปลความหมายของคำสั่งทางคณิตศาสตร์

6. แมททริกซ์

4.2.2 ตัวอย่างการทดลองในหน่วยประมวลผลคำสั่งทางคณิตศาสตร์ (ALU)

ในขั้นพื้นฐานโอเปอร์เรนด์ของคำสั่งที่ถูกปฏิบัติงานจะอยู่ที่แอดคิวมูลเตอร์และหลังจากที่การทำงานสิ้นสุดลงผลลัพธ์ก็จะถูกโหลดเข้าสู่แอดคิวมูลเตอร์รีจิสเตอร์ ต่อมาที่ถูกใช้งานซึ่งต่ออยู่กับ ALU คือแฟลกรีจิสเตอร์แต่ละบิตของ รีจิสเตอร์นี้จะถูกกำหนดตามผลลัพธ์

ในชุดฝึกแฟลกรีจิสเตอร์ที่มีอยู่คือ S,Z และ CY

ซายน์ (Sign:S) บิตถูกกำหนดเมื่อผลลัพธ์การทำงานทำให้บิตที่ 7 ของแอดคิวมูลเตอร์ถูกกำหนดซึ่งปกติจะเกิดขึ้นเสมอเมื่อมีการทำคำสั่งลบแล้ว ได้ผลลัพธ์มาเป็นค่าติดลบ

ซีโร (Zero:Z) บิตจะถูกกำหนดหลังจากการทำงานที่ให้ข้อมูลในแอดคิวมูลเตอร์เป็น 0

แครี่ (Cary:CY) บิตจะเป็นการแสดงผลลัพธ์ของการทำงานเมื่อข้อมูลมีค่าเกินกว่าความจุของแอดคิวมูลเตอร์ เพราะตัวทศมีค่ามากกว่าบิตแสดงผลหรือในการลบมีการข้มเกิดขึ้น

การทดลองที่ 1

อธิบายการเปลี่ยนแปลงที่เกิดขึ้นกับแฟลค CY และ แฟลค Z โดยให้เปรียบเทียบกับคำสั่งที่ให้มา

Addr	Binary	Hex	Mnemonic	Commentary
00	0011 1110	3E	LD A , 04	Load 04 to ACC
01	0000 0100	04		
02	1100 0110	C6	ADD 05	Add 05 to ACC
03	0000 0101	05		
04	0000 0000	00	NOP	No operation
05	1100 0110	C6	ADD F7H	Add F7H to ACC
06	1111 0111	F7		

ตารางที่ 4.1 คำสั่งประกอบการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1 การทดลองโดยใช้เมทริกซ์

ลำดับขั้นการทดลอง

1. กดสวิทช์ RESET เพื่อให้เครื่องพร้อมที่จะทำงาน
2. กดสวิทช์ CLK เพื่อเข้าสู่โหมดการใช้เมทริกซ์ให้ทำการสังเกต LED เมทริกซ์จะติดชุดแสดงผลจะแสดงผลแรมที่แอดเดรส 00 และแสดงข้อมูลภายในแอดเดรส 00 ออกมา
3. ป้อนคำสั่งตามใบงาน โดยจะใช้เมทริกซ์เมื่อใส่ข้อมูลลงไปแอดเดรสจะเลื่อนให้เอง โดยอัตโนมัติระหว่างที่ป้อนข้อมูลสามารถใช้สวิทช์ INC และ DEC เลื่อนไปยังแอดเดรสที่ต้องการได้

4. หลังจากป้อนคำสั่งจนหมดแล้วกดสวิทช์ CLK หรือ RESET 1 ครั้งเพื่อออกจากระบบการใช้เมทริกซ์ หลังจากที่เกิดแล้วชุดแสดงผลทั้งหมดจะแสดงผล

5. กดสวิทช์ STEP เพื่อทำงานในโหมดซิงเกิลสเตป

5.1 ในช่วงนี้ให้เทียบผลกับตารางในใบงานสิ่งที่เกิดขึ้นคือ

DB จะนำค่าในแอดเดรสเริ่มต้น (00) มาใส่ซึ่งมีค่าเป็น 3E

IR จะนำคำสั่ง 3E มาใส่เพราะ 3E เป็นคำสั่ง

PC จะมีค่าเท่ากับ $PC + 1 = 01 + 1 = 02$

AB จะมีค่าเท่ากับ $AB = 00$

ACC จะมีค่าเท่ากับ 00 เพราะคำสั่งยังไม่ประมวลผล

5.2 กดสวิทช์ STEP ครั้งที่ 2

DB จะนำค่าในแอดเดรสบวก 1 (01) มาใส่ซึ่งมีค่าเป็น 04

IR ยังคงเป็นคำสั่ง 3E เพราะ 04 ไม่เป็นคำสั่งแต่เป็นข้อมูล

PC จะมีค่าเท่ากับ $PC + 1 = 02 + 1 = 03$

AB จะมีค่าเท่ากับ $AB + 1 = 00 + 1 = 01$

ACC จะมีค่าเท่ากับ 04 เพราะคำสั่งเกิดการประมวลผลแล้ว

5.3 กดสวิทช์ STEP ครั้งที่ 3

DB จะนำค่าในแอดเดรสบวก 1 (02) มาใส่ ซึ่งมีค่าเป็น C6

IR เปลี่ยนเป็น C6 เพราะ C6 เป็นคำสั่ง

PC จะมีค่าเท่ากับ $PC + 1 = 03 + 1 = 04$

AB จะมีค่าเท่ากับ $AB + 1 = 01 + 1 = 02$

ACC ยังคงมีค่าเท่ากับ 04

5.4 กคสวิทช์ STEP ครั้งที่ 4

DB จะนำค่าในแอดเดรสบวก 1 (03) มาใส่ ซึ่งมีค่าเป็น 05

IR ยังคงเป็นเป็น C6 เพราะ 05 ไม่เป็นคำสั่ง

PC จะมีค่าเท่ากับ $PC + 1 = 04 + 1 = 05$

AB จะมีค่าเท่ากับ $AB + 1 = 03 + 1 = 04$

ACC จะมีค่าเท่ากับ $ACC + 05 = 04 + 05 = 09$

5.5 กคสวิทช์ STEP ครั้งที่ 5

DB จะนำค่าในแอดเดรสบวก 1 (04) มาใส่ซึ่งมีค่าเป็น 00

IR เปลี่ยนเป็น 00 เพราะ 00 เป็นคำสั่ง

PC จะมีค่าเท่ากับ $PC + 1 = 05 + 1 = 06$

AB จะมีค่าเท่ากับ $AB + 1 = 04 + 1 = 05$

ACC ยังคงมีค่าเท่ากับ 09 เพราะคำสั่งนี้ไม่มีผลในการทำงาน

5.6 กคสวิทช์ STEP ครั้งที่ 6

DB จะนำค่าในแอดเดรสบวก 1 (05) มาใส่ ซึ่งมีค่าเป็น C6

IR ถูกนำค่า C6 มาใส่เพราะ C6 เป็นคำสั่ง

PC จะมีค่าเท่ากับ $PC + 1 = 06 + 1 = 07$

AB จะมีค่าเท่ากับ $AB + 1 = 05 + 1 = 06$

ACC ยังคงมีค่าเท่ากับ 09

5.7 กคสวิทช์ STEP ครั้งที่ 7

DB จะนำค่าในแอดเดรสบวก 1 (06) มาใส่ ซึ่งมีค่าเป็น F7

IR มีค่าเป็น C6 เพราะ F7 ไม่เป็นคำสั่ง

PC จะมีค่าเท่ากับ $PC + 1 = 07 + 1 = 08$

AB จะมีค่าเท่ากับ $AB + 1 = 06 + 1 = 07$

ACC จะมีค่าเท่ากับ $ACC + F7 = 09 + F7 = 100$

ACC เท่ากับ 00

มีผลต่อแฟลกคือ $Z = 1, C = 1$

1.2 การทดลองโดยใช้ชุด COMMAND ALU

ลำดับขั้นการทดลอง

1.กคสวิทช์ RESET เพื่อให้เครื่องพร้อมที่จะทำงาน

เอกสารนี้เป็นเอกสารทรัพย์สินทางปัญญาของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. กศวิทย์ USE ALU เพื่อใช้คำสั่งใน ALU

3. ทำงานคำสั่งแรกคือ LD A, 04

3.1 กำหนดสวิตช์ข้อมูลให้มีค่าตามต้องการคือ 04 โดยดู DB ประกอบ DB จะเปลี่ยนแปลงไปตามการกศวิทย์

3.2 กศวิทย์ WRA สังเกตที่ ACC จะนำค่า 04 จาก DB มาเก็บไว้

4. ทำงานคำสั่งที่ 2 คือ ADD 05 โดย

4.1 กำหนดสวิตช์ข้อมูลให้มีค่าตามต้องการคือ 05 โดยดู DB ประกอบ

4.2 กำหนดสวิตช์ ALU โดยใช้สวิตช์ S_0, S_1, S_2 ให้ตรงกับคำสั่ง ADD คือ 100

4.3 กศวิทย์ EXE เพื่อประมวลผลคำสั่ง

5. ผลลัพธ์ที่ได้คือ

ACC เท่ากับ $ACC + 05 = 04 + 05 = 09$

DB จะนำค่าจาก ACC มาเก็บไว้เช่นกันคือเท่ากับ 09

TEMP จะนำค่าข้อมูลที่ใช้งานครั้งสุดท้ายสุดมาเก็บไว้คือเท่ากับ 05

6. คำสั่งต่อมาคือ NOP ซึ่งคำสั่งนี้ไม่มีผลต่อการทำงานใดๆ

7. ทำคำสั่ง ADD F7 ทำได้โดย

7.1 กำหนดสวิตช์ข้อมูลให้มีค่าเท่ากับ F7

7.2 กำหนดสวิตช์ COMMAND ALU ให้มีค่าเป็นคำสั่ง ADD คือ 100

7.3 กศวิทย์ EXE จะได้ผลลัพธ์ดังนี้

ACC เท่ากับ $ACC + F7 = 09 + F7 = 100$

ACC เท่ากับ 00 , แครี่เท่ากับ 1 , แฟล็ก Z เท่ากับ 1

DB เท่ากับ ACC เท่ากับ 00

TEMP เท่ากับ F7

สรุปผลการทดลอง

1. คำสั่งบวกที่ 1 แฟล็กไม่เปลี่ยนแปลงอะไรทุกแฟล็กเป็น 0

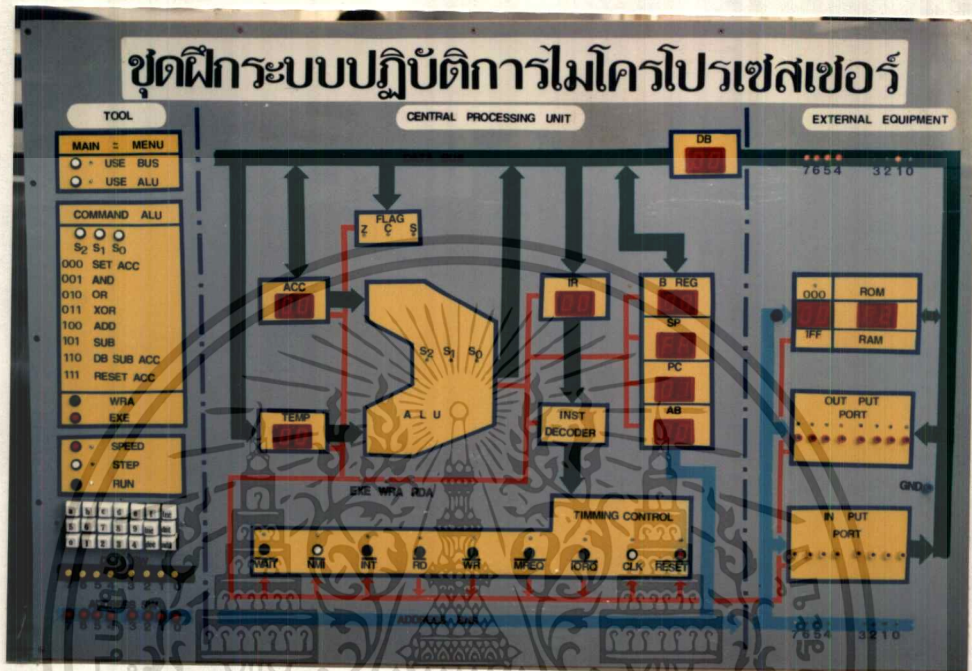
2. คำสั่งบวกที่ 2 แฟล็ก Z และแฟล็ก CY จะถูกกำหนดเป็น 1

หลังจากการบวกครั้งแรกผลลัพธ์ยังคงมีค่าน้อยกว่าตัวเลขสูงสุดแอสคิโมเตเตอร์สามารถเก็บไว้ได้ดังนั้นทุกแฟล็กก็จะกลับสู่สภาพเดิม

การบวกครั้งที่ 2 ผลลัพธ์ในแครี่มีค่ามากกว่าที่จะเก็บไว้ได้ดังนั้นแฟล็ก CY จะถูกกำหนดเมื่อค่าใน แอสคิโมเตเตอร์ เป็น 0 แฟล็ก Z จะถูกกำหนดด้วย

4.3 การทดลองการทำงานในหน่วยประมวลผลกลาง

4.3.1 อุปกรณ์ที่เกี่ยวข้องกับการทำงานในหน่วยของตัวประมวลผลกลาง



รูปที่ 4.7 อุปกรณ์ส่วนต่างๆของหน่วยประมวลผลกลาง

1. กดสวิทช์ RUN เมื่อกดเครื่องจะทำการประมวลผลโดยอัตโนมัติตามขั้นตอน
2. กดสวิทช์ STEP เมื่อกดเครื่องจะทำการประมวลผลทีละไบต์ ตามการกดที่ทำการส่งโดยเครื่องจะทำงานตามขั้นตอนที่เรากำหนด
3. กดสวิทช์ SPEED มี 2 ระดับคือ
 - ลักษณะแรกเป็นความเร็วธรรมดาคือจะมีการประมวลผลธรรมดาทีละไบต์
 - ลักษณะถัดมาใช้ร่วมกับสวิทช์ STEP จะมีการทำงานโดยละเอียดทีละขั้นตอน
4. LED S0 , S1 , S2 เป็นตัวแปรคำสั่งทางคณิตศาสตร์
5. หน่วยแสดงผล 7 ส่วนทุกตัวจะแสดงผลการประมวลผลตามหน้าที่ของตัวเองตามโปรแกรม
6. กดสวิทช์ CLK เพื่อเลือกที่จะใช้เมทริกซ์
7. เมทริกซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 ตัวอย่างการทดลองในหน่วยประมวลผลกลาง

โครงสร้างของไมโครโปรเซสเซอร์สามารถแบ่งเป็นส่วนได้อย่างชัดเจนแต่ละส่วนจะมีหน้าที่ในกระบวนการของคำสั่งโดยแบ่งออกเป็น

1.ระบบต่อเพื่อโอนย้ายข้อมูล (Bus System)

การโอนย้ายสัญญาณจะโอนย้ายจากภายนอกหรือภายในเข้าสู่ไมโครโปรเซสเซอร์ผ่านระบบของสายโดยทั่วไปสวิตช์ USE BUS เป็นตัวอ้างอิงซึ่งประกอบไปด้วย 3 ส่วนพื้นฐานคือ ข้อมูล, แอดเดรส และ TIMING CONTROL

DATA BUS เป็นระบบสายซึ่งจะใช้ในการโอนย้ายข้อมูลระหว่างไมโครโปรเซสเซอร์และหน่วยอื่นๆของระบบในทิศทางใดก็ได้ (2 ทิศทาง)

แอดเดรสบัสจะประกอบไปด้วยจำนวนของสายซึ่งจะโอนย้ายข้อมูลไปยังหน่วยที่ถูกใช้งานชนิดของข้อมูลจะใช้ในการอ้างแอดเดรสโดยจะเป็นตัวชี้จำนวนสายของแอดเดรสบัสซึ่งบัสนี้จะถูกควบคุมโดยไมโครโปรเซสเซอร์เท่านั้นในทิศทางเดียว (Uni Directional)

การทำงานร่วมกันของข้อมูล, แอดเดรสและไทม์มิงสัญญาณภายในไมโครโปรเซสเซอร์และไมโครคอมพิวเตอร์ที่สมบูรณ์จะถูกควบคุมโดยสัญญาณจาก TIMING CONTROL BUS

2.หน่วยควบคุม

หน่วยควบคุมของไมโครโปรเซสเซอร์ประกอบไปด้วย 2 ส่วนหลักคือชุดลอจิกคำสั่งกับระบบของไทม์มิง TIMING CONTROL โดยชุดลอจิกคำสั่งจะลอจิกคำสั่งควบคุมซึ่งบรรจุอยู่ในไบท์แรกของแต่ละคำสั่ง เมื่อลอจิกคำสั่งเสร็จก็จะเข้าสู่ไทม์มิง TIMING CONTROL ซึ่งจะไปรวมกันเป็นคำสั่งที่สมบูรณ์ซึ่งก็ขึ้นอยู่กับคำสั่งควบคุมอื่นๆจาก TIMING CONTROL BUS ภายนอก

3.หน่วยประมวลผลกลาง

หน่วยประมวลผลของไมโครโปรเซสเซอร์ประกอบไปด้วย ALU , แอคคิวมูเลเตอร์ และแฟลกริजิสเตอร์โดย ALU เป็นตัวหลักในการบวกจะกระทำกับรีจิสเตอร์เฉพาะเช่นแอคคิวมูเลเตอร์สามารถที่จะกระทำกับการคำนวณ (Arithmetic) และลอจิกขึ้นอยู่กับการควบคุมแต่ละบิต ถูกกำหนดในแฟลกริजิสเตอร์ซึ่งจะถูกใช้ตามการทำของ ALU

4 รีจิสเตอร์

รีจิสเตอร์ของไมโครโปรเซสเซอร์จะเป็นตัวบอกจำนวนของตำแหน่งของความจำซึ่งรวมขึ้นเพื่อให้เหมาะสมกับข้อมูลหรือแอดเดรสในไมโครโปรเซสเซอร์ (8 หรือ 16 บิต)

ตำแหน่งของความจำเหล่านี้ถูกใช้ในการเป็นตัวกลางในการเก็บข้อมูลโดยการอ้างถึง

ตำแหน่งแอดเดรส ซึ่งจะกำหนดจากการเข้าแอดเดรสภายในรีจิสเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- AB แอดเดรสบัส
- DB ดาต้าบัส
- PC โปรแกรมเคาท์เตอร์
- SP สแตคพอยท์เตอร์
- ACC แอควิวมูลเตอร์
- D₀-D₇ ดาต้า
- A₀-A₇ แอดเดรส
- S₀-S₃ ไทม์มิ่งคอนโทรล

Addr	Binary	Hex	Mnemonic	Commentary
00	0011 1110	3E	LD A , n	Load แอควิวมูลเตอร์
01	1111 1111	FF		Constant , n
02	1100 0110	C6	ADD A,n	Add constant to
				the contents of the
				accumulator
03	0000 0001	01		Constant
04	0011 1110	3E	LD A,n	Load accumulator
05	1010 1010	AA		Constant
06	0010 1110	2E	LD B,n	Load B register
07	0101 0101	55		Constant
08	1010 0101	A0	AND B	And constant with the
				constant of the
				accumulator
09	0000 0000	00	NOP	No Operate

ตารางที่ 4.2 ชุดคำสั่งที่ใช้ในการทดลอง

ลำดับขั้นการทดลอง

1.กดสวิตช์ RESET เพื่อทำการเริ่มต้น

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้เพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.กดสวิทช์ CLK เพื่อเข้าสู่โหมดการใช้แมททริกคีย์สังเกต LED ที่แมททริกคีย์จะติดชุด
แสดงผลจะแสดงผลแรมแอดเดรส 00 และแสดงข้อมูลภายในแอดเดรส 00 ออกมาด้วย

3. ป้อนคำสั่งตามใบงานโดยใช้เมาท์ทริกคีย์เมื่อใส่ข้อมูลลงไปแอคเคสจะเลื่อนให้เองโดยอัตโนมัติโดยระหว่างที่ป้อนข้อมูลสามารถใช้สวิทช์ INC และ DEC เลื่อนแอคเคสไปที่ต้องการ

4. หลังจากป้อนคำสั่งจนหมดแล้วกดสวิทช์ CLK หรือ RESET 1 ครั้งเพื่อออกจากระบบ
การใช้เมทริกคีย์หลังจากที่กดแล้วชุดแสดงผลทั้งหมดจะแสดงผล

5. กคสวิทย์ STEP เพื่อทำงานในโหมคซึ่งเกิลสเทป

6.เทียบผลการทดลองกับตารางในใบงานซึ่งจะได้ผลดังนี้

DB จะนำค่าในแอตเตริสแรกมาใส่ไว้ในค่าแอตเตริส 00 ซึ่งมีค่าเท่ากับ 3E

IR จะนำค่า 3E ซึ่งเป็นคำสั่งมาเก็บ

PC จะนำค่าเท่ากับ $PC+1 = 01+1 = 02$

AB จะมีค่าเท่ากับ $AB+1 = 00+1 = 01$

ACC ยังคงเป็น 00 แพลก S,Z,C จะไม่มีผลอะไรเพราะคำสั่งยังไม่ทำงาน

รีจิสเตอร์ B ยังไม่มีการเปลี่ยนแปลง

7.กวดวิชา STEP ครั้งที่ 2 มีผลดังนี้

DB นำค่าในแอดเดรสเท่ากับแอดเดรสบวก 1 มาใส่เท่ากับ $00+01 = 01$ ซึ่งค่าในแอดเดรสที่ 01 คือ FF ดังนั้น DB จะแสดงค่า FF

IR จะมีค่าเดิมคือ 3F เพราะ FF ไม่ใช่ค่าตั้ง ดังนั้น IR จะไม่นำมาแสดงผล

PC จะมีค่าเท่ากับ $PC+1 = 02+1 = 03$

AB จะมีค่าเท่ากับ $AB+1 = 01+1 = 02$

ACC จะถูกใส่ลงไปด้วย FF ตามคำสั่ง LD A, FF

แฟลก Z , C และ S จะเป็น 0 ไม่มีการเปลี่ยนแปลงเพราะการใช้คำสั่ง LD จะไม่มีผลต่อแฟลกถึงแม้ข้อมูลที่โอนย้ายเข้าไปจะทำให้ค่าใน ACC เป็น 0 หรือเป็นคำสั่งลบก็ตาม

รีจิสเตอร์ B ยังไม่มีการเปลี่ยนแปลง

8.กคสวितซ์ STEP เป็นครั้งที่ 3

DB นำค่าในแอดเดรสเท่ากับแอดเดรสบวก 1 มาใส่เท่ากับ $01+01 = 02$ ซึ่งค่าในแอดเดรสที่ 02 คือ C6 ดังนั้น DB จะแสดงค่า C6

IR จะมีค่า C6 เพราะ C6 เป็นกำลังคั้งนั้น IR จะนำมาแสดงผล

PC จะมีค่าเท่ากับ $PC+1 = 03+1 = 04$

AB จะมีค่าเท่ากับ $AB+1 = 02+1 = 03$

ACC ยังคงมีค่าเป็น FF

แฟลก Z , C และ S จะเป็น 0 ไม่มีการเปลี่ยนแปลง

รีจิสเตอร์ B ยังไม่มีการเปลี่ยนแปลง

9. กคสวิทช์ STEP เป็นครั้งที่ 4

DB นำค่าในแอดเดรสเท่ากับแอดเดรสบวก 1 มาใส่เท่ากับ $02+01 = 03$ ซึ่งค่าในแอดเดรสที่ 03 คือ 01 ดังนั้น DB จะแสดงค่า 01

IR จะมีค่าเท่ากับ C6

PC จะมีค่าเท่ากับ $PC+1 = 04+1 = 05$

AB จะมีค่าเท่ากับ $AB+1 = 03+1 = 04$

ACC ยังคงมีค่าเป็นเท่ากับ $ACC + 1 = 100$

จากผลการบวกทำให้แฟลก Z และ C เป็น 1 และ S จะเป็น 0

รีจิสเตอร์ B ยังไม่มีการเปลี่ยนแปลง

10. กคสวิทช์ STEP เป็นครั้งที่ 5

DB นำค่าในแอดเดรสเท่ากับ แอดเดรสบวก 1 มาใส่เท่ากับ $03+01 = 04$ ซึ่งค่าในแอดเดรส ที่ 04 คือ 3E ดังนั้น DB จะแสดงค่า 3E

IR จะมีค่า 3E เพราะ 3E เป็นคำสั่งดังนั้น IR จะนำมาแสดงผล

PC จะมีค่าเท่ากับ $PC+1 = 05+1 = 06$

AB จะมีค่าเท่ากับ $AB+1 = 04+1 = 05$

ACC ยังคงมีค่าเป็น 00

แฟลก Z และ C ยังคงเป็น 1 ไม่มีการเปลี่ยนแปลง

รีจิสเตอร์ B ยังไม่มีการเปลี่ยนแปลง

11. กคสวิทช์ STEP เป็นครั้งที่ 6

DB นำค่าในแอดเดรสเท่ากับแอดเดรสบวก 1 มาใส่เท่ากับ $04+01 = 05$ ซึ่งค่าในแอดเดรสที่ 05 คือ AA ดังนั้น DB จะแสดงค่า AA

IR จะมีค่า 3E เพราะ 3E เป็นคำสั่งดังนั้น IR จะนำมาแสดงผล

PC จะมีค่าเท่ากับ $PC+1 = 06+1 = 07$

AB จะมีค่าเท่ากับ $AB+1 = 05+1 = 06$

ACC ยังคงมีค่าเป็น AA ตามการทำงานของคำสั่ง LD A , AA

แฟลก Z และ C จะเป็น 1 ไม่มีการเปลี่ยนแปลงแม้ว่า ACC จะเกิดการเปลี่ยนแปลงเพราะคำสั่ง LD เนื่องจากคำสั่ง LD ไม่มีผลต่อแฟลคนั่นเอง

รีจิสเตอร์ B ยังไม่มีการเปลี่ยนแปลง

12. กคสวิตช์ STEP เป็นครั้งที่ 7

DB นำค่าในแอดเดรสเท่ากับแอดเดรสบวก 1 มาใส่เท่ากับ $05+01 = 06$ ซึ่งค่าในแอดเดรส ที่ 06 คือ 2E ดังนั้น DB จะแสดงค่า 2E

IR จะมีค่า 2E อยู่เพราะ 2E เป็นคำสั่งดังนั้น IR จะนำมาแสดงผล

PC จะมีค่าเท่ากับ $PC+1 = 07+1 = 08$

AB จะมีค่าเท่ากับ $AB+1 = 06+1 = 07$

ACC ยังคงมีค่าเป็น AA

แฟลก Z และ C จะเป็น 1 ไม่มีการเปลี่ยนแปลง

รีจิสเตอร์ B ยังไม่มีการเปลี่ยนแปลง

13. กคสวิตช์ STEP เป็นครั้งที่ 8

DB นำค่าในแอดเดรสเท่ากับแอดเดรสบวก 1 มาใส่เท่ากับ $06+01 = 07$ ซึ่งค่าในแอดเดรสที่ 07 คือ 55 ดังนั้น DB จะแสดงค่า 55

IR จะมีค่า 2E ค้างอยู่

PC จะมีค่าเท่ากับ $PC+1 = 08 + 1 = 09$

AB จะมีค่าเท่ากับ $AB+1 = 07 + 1 = 08$

ACC ยังคงมีค่าเป็น AA

แฟลก Z และ C จะเป็น 1 ไม่มีการเปลี่ยนแปลง

รีจิสเตอร์ B จะถูกนำข้อมูล 55 มาใส่ตามคำสั่ง LD B , 55

14. กคสวิตช์ STEP เป็นครั้งที่ 9

DB นำค่าแอดเดรสเท่ากับแอดเดรสบวก 1 มาใส่เท่ากับ $07+01 = 08$ ค่าแอดเดรสที่ 08 คือ A5 ดังนั้น DB จะแสดงค่า A5

IR จะมีค่า A5 เพราะ A5 เป็นคำสั่งดังนั้น IR จะนำมาแสดงผล

PC จะมีค่าเท่ากับ $PC+1 = 09+1 = 0A$

AB จะมีค่าเท่ากับ $AB+1 = 08+1 = 09$

ACC จะมีค่าเป็นเท่ากับ ACC And 55 = AA And 55 = 00

แฟลก C จะถูกรีเซ็ตให้เป็น 0 เสมอเมื่อมีการกระทำคำสั่ง AND

แฟลก Z เปลี่ยนแปลงไปตามค่าใน ACC ซึ่งในใบงานนี้ $ACC = 0$ ดังนั้นแฟลก Z เท่ากับ 1

รีจิสเตอร์ B ยังไม่มีการเปลี่ยนแปลงจะมีค่าเป็น 55 ดังเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

15.กคสวิทช์ STEP ครั้งที่ 10

เป็นการทำคำสั่ง NOP คือไม่มีการทำงานใดๆดังนั้นผลจะมีค่าเหมือนกับการกค STEP ครั้งที่ 9 เพียงแต่ว่าค่า PC และ AB จะเพิ่มขึ้นมาอีก 1 เท่านั้นส่วน IR จะนำรหัสคำสั่ง NOP (00) มาเก็บไว้

STEP	1	2	3	4	5	6	7	8	9	10	11
DB	xx	3E	FF	C6	01	3E	AA	2E	55	55	00
PC	01	02	03	04	05	06	07	08	09	0A	0B
ACC	00	00	FF	FF	00	00	AA	AA	AA	00	00
S - FLAG	0	0	0	0	0	0	0	0	0	0	0
Z - FLAG	0	0	0	0	1	1	1	1	1	1	1
CY - FLAG	0	0	0	0	1	1	1	1	1	0	0
B - REG	xx	xx	xx	xx	xx	xx	xx	xx	55	55	55

ตารางที่ 4.3 ผลการทดลองจากตารางที่ 4.2

อธิบายกระบวนการของคำสั่งเหล่านี้เมื่อมีการทำงาน
ชุดคำสั่งนี้แสดงการทำงานของหน่วยประมวลผลซึ่งอยู่ในไมโครโปรเซสเซอร์ค่าคงที่ จะถูกโหลดเข้าสู่แอดคิวมูลเตอร์ในชุดคำสั่งที่ 1

ค่าคงที่อื่นๆ (01) จะถูกบวกเข้ากับแอดคิวมูลเตอร์ (FE) ด้วยคำสั่งที่ 2 เมื่อความจำของ แอดคิวมูลเตอร์ไม่พอเพียงที่จะเก็บผลลัพธ์ของการบวกนี้แครี่บิท (CY) ในแฟลกริजิสเตอร์ก็จะถูก กำหนดให้เป็น 1 เช่นกันด้วยผลลัพธ์นี้ข้อมูลในแอดคิวมูลเตอร์ก็จะเป็น 00 ดังนั้นซีโรแฟล็กก็ จะถูกกำหนด

ในคำสั่งชุดที่ 2 การแทนที่ของการทำงานทางคณิตศาสตร์จะถูกแทนการทำงานด้วยลอจิก (AND) ระหว่างข้อมูลในแอดคิวมูลเตอร์และรีจิสเตอร์ B เมื่อข้อมูล 2 บิตถูกนำมารวมกันผลลัพธ์ ของการ AND จะเป็น 0 ถ้าในแอดคิวมูลเตอร์จะถูกแทน ที่ด้วยผลลัพธ์ของการทำงานด้วยลอจิก

บทที่ 5

บทวิจารณ์สรุปและแนวทางพัฒนา

5.1 บทสรุป

ปฏิญานิพนธ์ที่จัดทำขึ้นนี้ จะทำให้ผู้ศึกษามีทักษะสามารถจินตนาการได้ โดยรูปแบบของโครงสร้างหน่วยประมวลผลกลางที่สมบูรณ์นั้น การมองภาพแยกแยะการทำงานจะมีข้อยุ่งยากมากมายหากสามารถจำลองโดยการนำเฉพาะส่วนสำคัญออกมาให้เป็นพื้นฐานหลักๆ ได้ศึกษาได้ทดลอง ก็จะเป็นพื้นฐานของระบบที่ซับซ้อนขึ้นไปอีก

ในชุดฝึกนี้จะมีความคล้ายคลึงกับซิงเกิลบอร์ดแต่จะมีข้อแตกต่างกันที่ชัดเจนดังนี้

1. ซิงเกิลบอร์ดไม่สามารถอธิบายกระบวนการในการทำงานของไมโครโปรเซสเซอร์โดยละเอียดได้ต้องอาศัยจินตนาการ แต่ในชุดฝึกนี้จะสามารถอธิบายถึงกระบวนการในการทำงานของไมโครโปรเซสเซอร์โดยละเอียดทำให้มองเห็นโครงสร้างและการทำงานได้ง่ายขึ้น

2. ซิงเกิลบอร์ดไม่สามารถแยกส่วนการทำงานแต่ละหน่วยเช่น ALU ,BUS ,CPU ให้ผู้ศึกษามองเห็นได้อย่างชัดเจนแต่ในชุดฝึกนี้มีการแยกส่วนการทำงานต่างๆอย่างชัดเจนทำให้ผู้ศึกษาวิเคราะห์การทำงานของส่วนต่างๆได้ง่ายขึ้น

3. ซิงเกิลบอร์ดมีความเหมาะสมสำหรับใช้งานจริงซึ่งมีการทำงานที่รวดเร็วเหมาะกับผู้ที่มีความรู้ขั้นพื้นฐานทางโครงสร้าง , การเขียนโปรแกรมและการทำงานของหน่วยต่างๆแล้วส่วนของชุดฝึกจะเหมาะสำหรับผู้เริ่มต้นศึกษาทางด้านนี้เท่านั้น

ข้อเสียของชุดฝึกนี้

1. ไม่สามารถประมวลผลคำสั่งแบบละเอียดได้ครบเท่ากับการประมวลผลคำสั่งที่ละไบต์
2. ไม่สามารถป้อนโปรแกรมที่มีความยาวเกิน 256 คำแหน่งได้
3. มีความเร็วในการทำงานต่ำกว่าซิงเกิลบอร์ดอย่างมากถึงแม้จะอยู่ในโหมดการทำงานที่มีความเร็วสูงสุดแล้วก็ตาม
4. ไม่สามารถขยายอุปกรณ์เพิ่มเติมได้อีกเช่น หน่วยความจำ , อุปกรณ์อินพุทเอาต์พุท
5. มีขนาดหน่วยความจำน้อย

5.1.1. หน่วยประมวลผลกลาง

ทำให้ทราบถึงขั้นตอนกระบวนการในการทำงานคำสั่งที่บรรจุอยู่ในหน่วยความจำว่ามี การกระทำที่สำคัญกับตัวแปรใดบ้าง เช่น คำสั่ง LD A , n จะมีช่วงแปลความหมายคำสั่งพีทซ์และ ช่วงปฏิบัติตามคำสั่งเอ็กซีคิว

5.1.2. หน่วยประมวลผลทางคณิตศาสตร์

ทำให้ทราบถึงขั้นตอนกระบวนการในการทำงานของคำสั่งทางคณิตศาสตร์และทราบผลที่ เกิดขึ้นของข้อมูล ซึ่งแสดงค่าสถานะข้อมูลแฟลก

5.1.3. หน่วยอ่านเขียนข้อมูลกับอุปกรณ์

ทำให้ทราบขั้นตอนในการส่งข้อมูลขั้นตอนการรับข้อมูลกับอุปกรณ์ภายนอกที่มาติดต่อ ซึ่งจะทำให้เกิดทักษะพื้นฐานถึงหลักการการติดต่อซึ่งมีการระบุตำแหน่งของอุปกรณ์,การระบุ ข้อมูลที่ใช้,การเลือกสัญญาณควบคุมเพื่อแบ่งชนิดของอุปกรณ์ เช่น ติดต่อหน่วยความจำเพื่ออ่าน จะต้องใช้สัญญาณควบคุม

นอกจากนี้ยังสามารถเชื่อมต่ออินเตอร์เฟซกับอุปกรณ์ภายนอกได้โดยผ่านทางพอร์ทที่มีให้ คืออินพุทพอร์ทและเอาต์พุทพอร์ท

5.2 ปัญหาและอุปสรรค

1.ทางด้านวงจรปัญหาที่พบคือ จากการใช้หลักการสแกนคีย์และสแกนชุดแสดงผล เมื่อมี การใช้งานส่วนของวงจรเสี่ยงการแสดงผลจะดับลงจนกว่าไม่มีการใช้ส่วนเสี่ยงจึงสามารถแสดงผล ได้ตามปกติ

2.ทางด้านโปรแกรมปัญหาที่พบคือ

-ในการออกแบบโปรแกรมส่วนการทำงานแบบละเอียด เนื่องจากข้อมูลของคำสั่งมีน้อยจึง ออกแบบโปรแกรมอธิบายขั้นตอนคำสั่งได้ไม่ครบทุกคำสั่งที่มีในชุดฝึกนี้

-ปัญหาเกี่ยวกับการใช้สแตคในคำสั่งที่มีผลต่อสแตคเมื่อป้อนเข้าสู่หน่วยความจำและมีการ ประมวลผลหากมีคำสั่งนี้มากจะทำให้เครื่องหยุดทำงานการแก้ไขต้องกรีเซทใหม่

5.3 แนวทางการพัฒนา

ส่วนของปัญหาที่เกิดขึ้นหากสามารถหาข้อมูลของคำสั่งอย่างละเอียดจะทำให้มีความ สมบูรณ์มากขึ้นได้

-ในหน่วยของการฝึกการอ่านเขียน อุปกรณ์ที่เกี่ยวข้องสามารถที่จะเพิ่มไดอะแกรมเวลาได้ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-รูปร่างของหน้าปัทม์และวงจร หากสามารถทำขึ้นใหม่โดยแยกแยกเป็นส่วนๆถึงเวลาจะฝึกในหน่วยหนึ่งก็นำส่วนย่อยที่จำเป็นมาประกอบกัน จะทำให้ผู้ศึกษามองภาพได้เข้าใจมากขึ้น

-ส่วนโปรแกรมมอนิเตอร์ขณะใช้ไม่เกิน 8 กิโลไบต์ซึ่งสามารถหาเพิ่มได้ถึง 32 กิโลไบต์ และหน่วยความจำที่เขียนได้มีขนาด 8 กิโลไบต์ฉะนั้นจึงสามารถที่จะเพิ่มโปรแกรมอำนวยความสะดวกต่างๆเพิ่มขึ้นได้

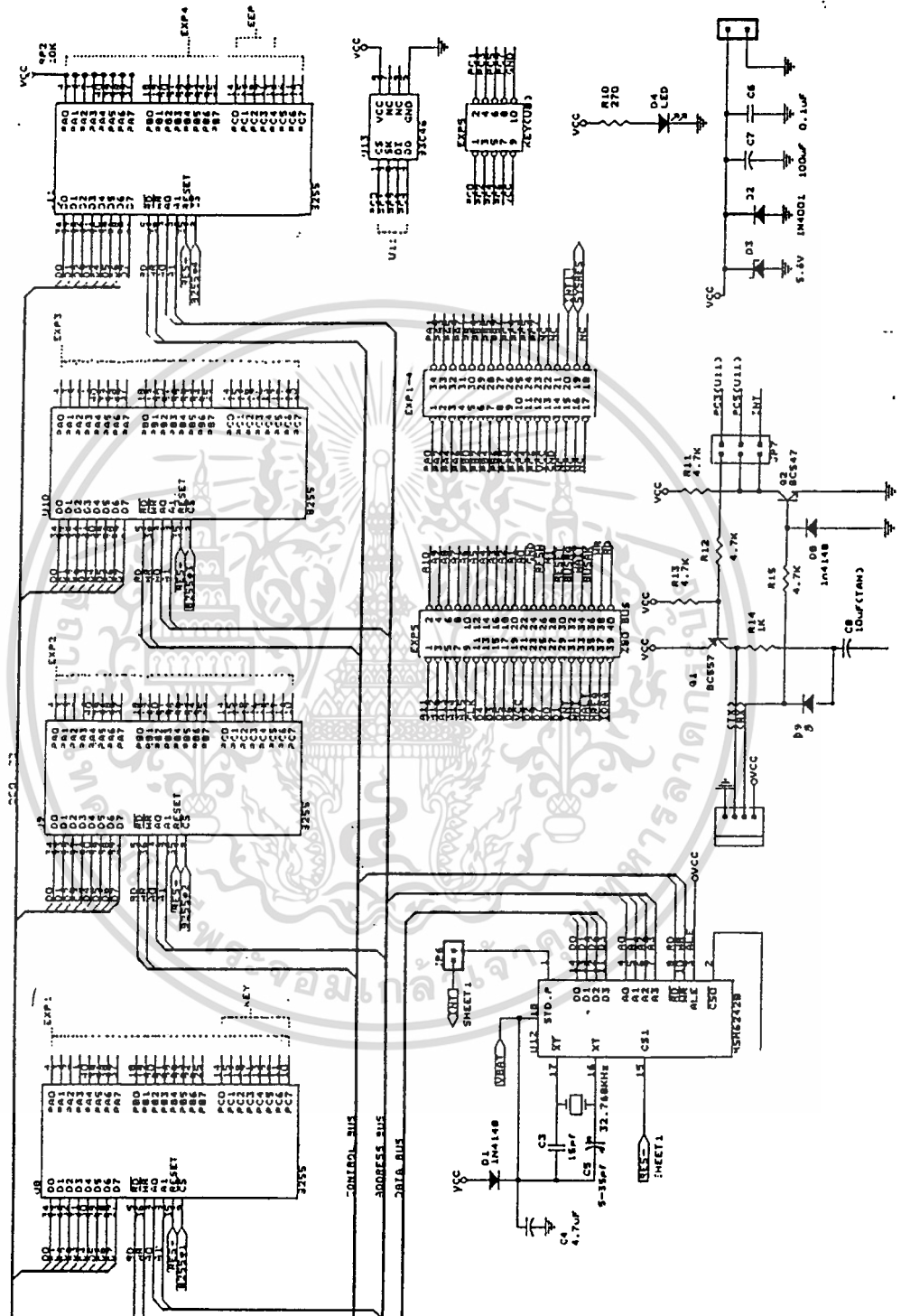
-ชุดฝึกนี้เน้นถึงพื้นฐานในเบื้องต้นซึ่งการที่ทำให้ผู้ศึกษาสามารถที่จะมองเห็นถึงการใช้งานได้แล้วจะทำให้ผู้เขียนมีความอยากศึกษาเพิ่มมากขึ้นอีก ฉะนั้นหากมีการเพิ่มชุดเชื่อมต่อภายนอกเข้าไปอีก เช่น ชุดควบคุมอุปกรณ์ไฟฟ้าโดยใช้ลอจิกควบคุมชุดควบคุมสเต็ปมอเตอร์และชุดอื่นที่จะทำให้ทราบถึงพื้นฐานการควบคุมอุปกรณ์ภายนอก



ภาคผนวก ก.วงจรที่ใช้ในชุดฝึกอบรมปฏิบัติการไมโครโปรเซสเซอร์

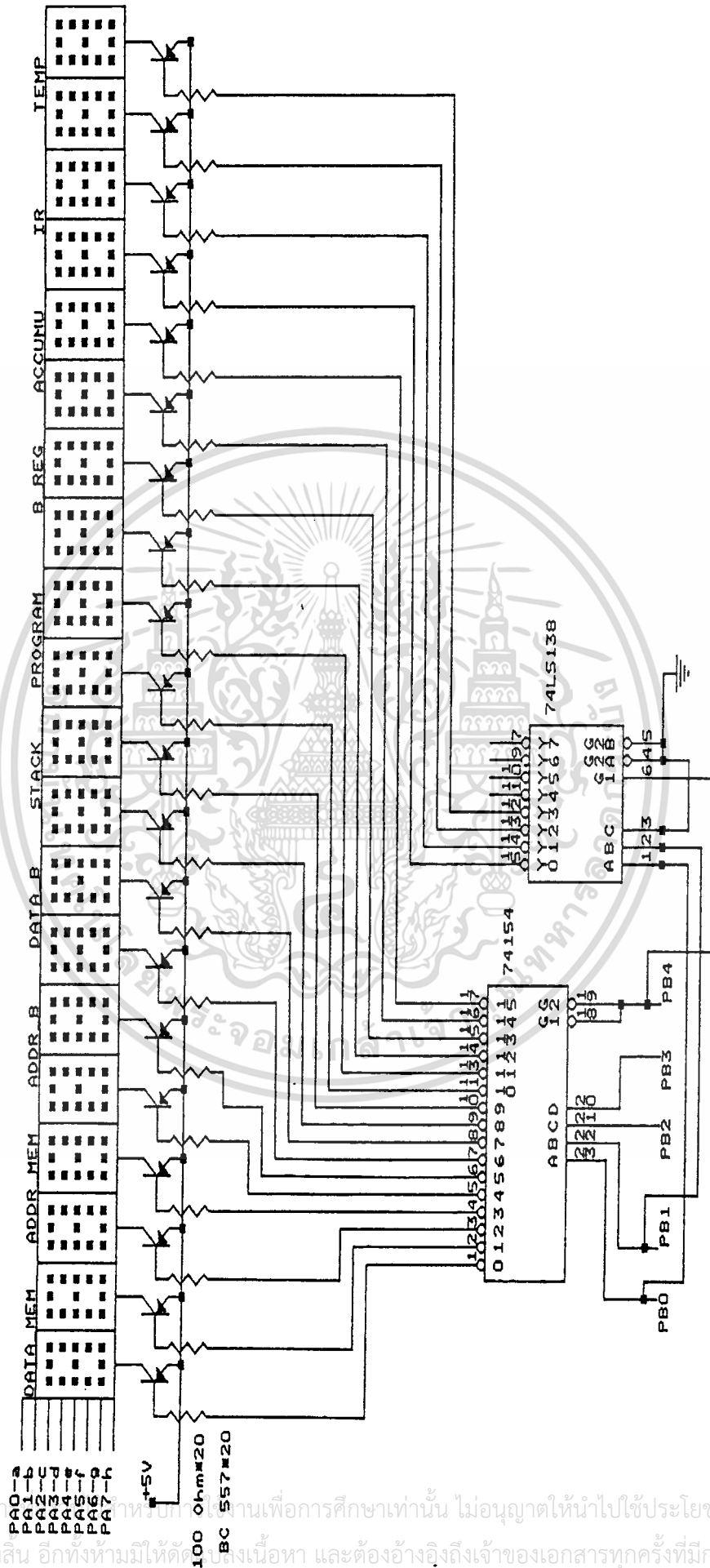


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

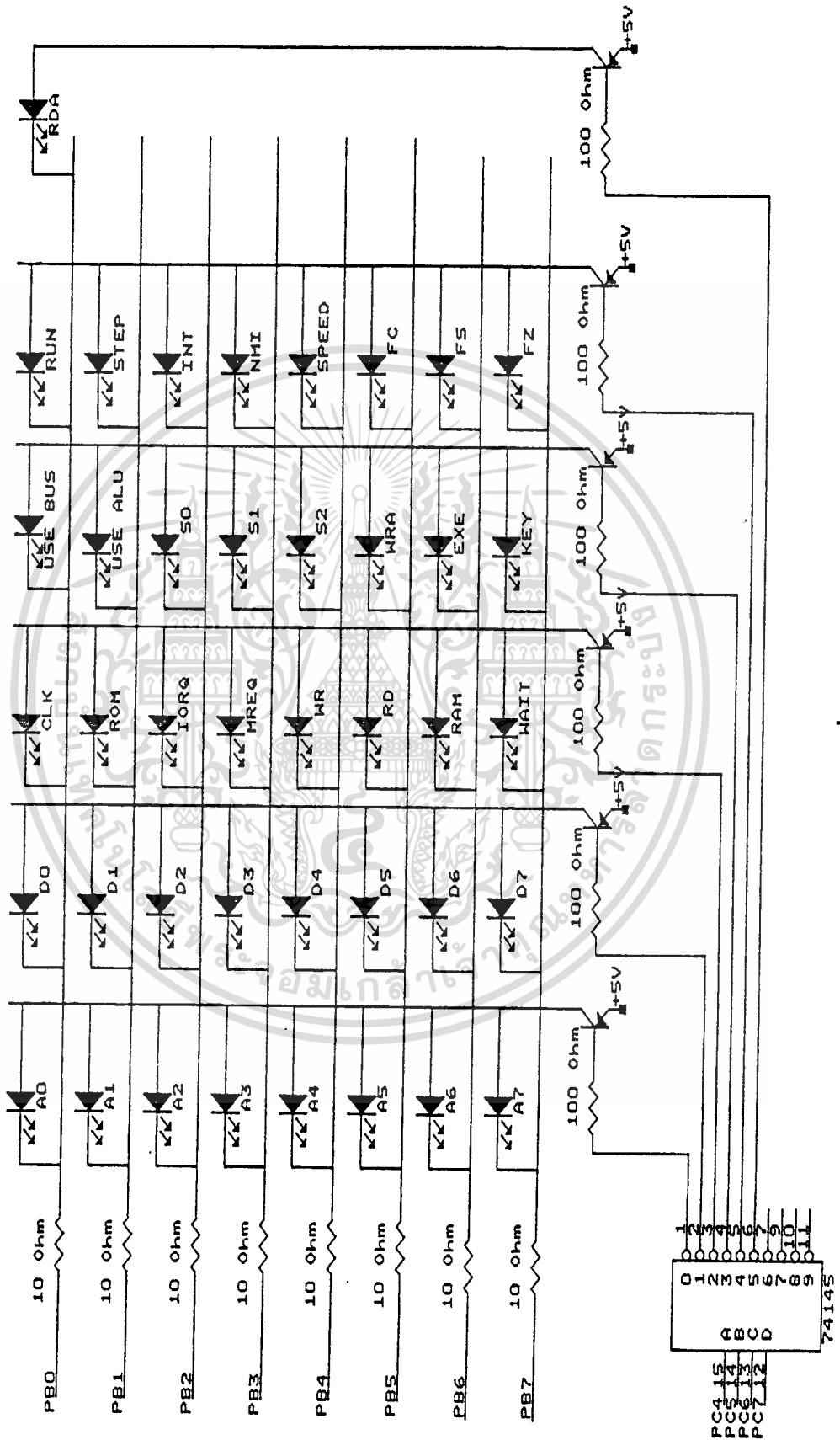


แสดงวงจร PORT I/O 5288

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

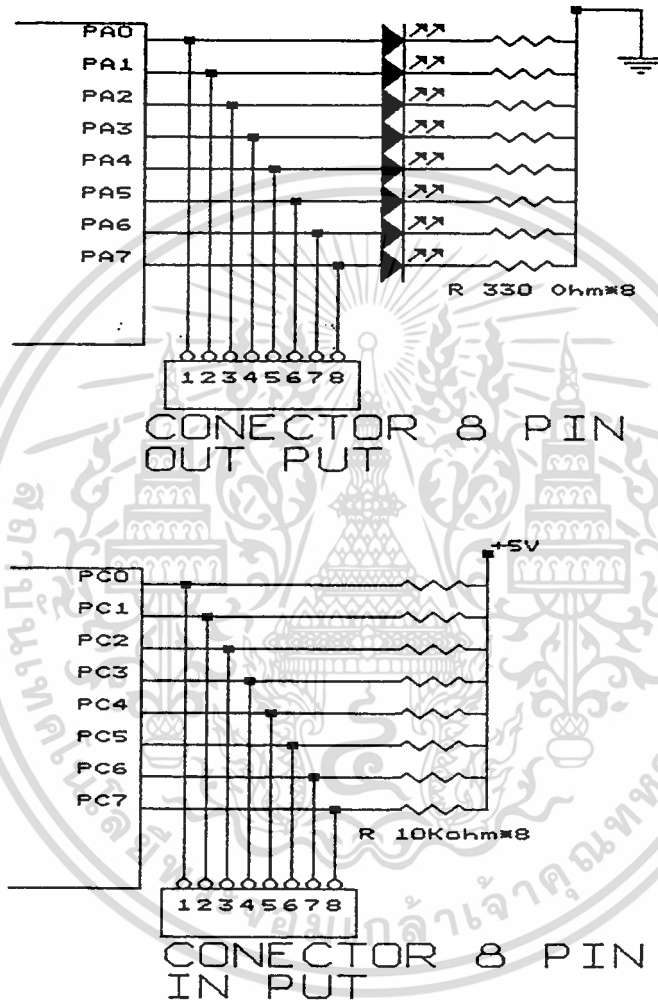


แสดงวงจรส่วนชุดแสดงผล 7 หลัก



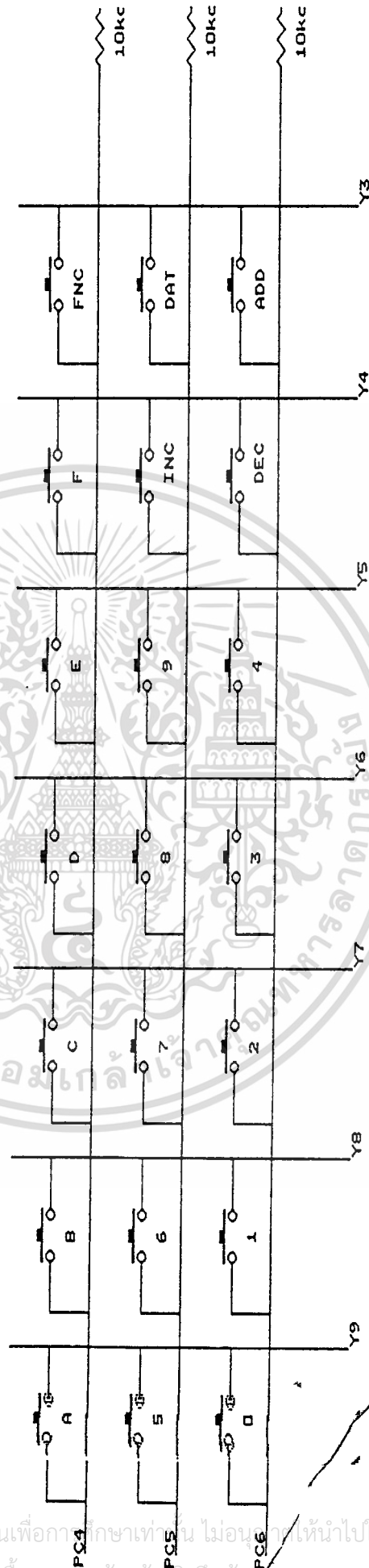
แสดงวงจรส่วน LED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



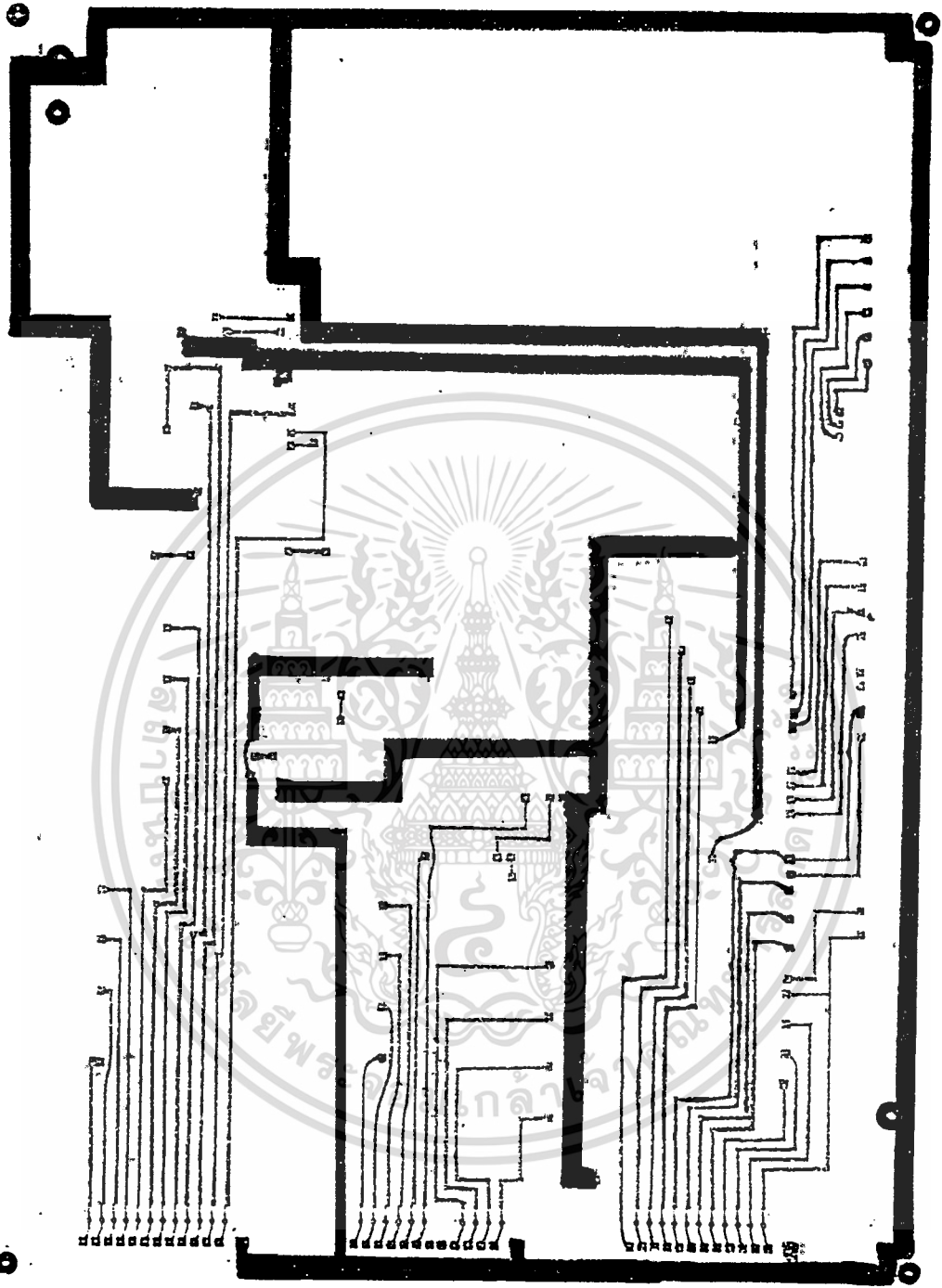
แสดงวงจรส่วนติดต่ออุปกรณ์ภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



OUT PUT FROM 74LS154

แสดงวงจรส่วนแมทริกซ์



แสดงฉายจริงของชุดฝึกด้านบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

27	0040	LED_COMM3:	EQU	40H	
28	0050	LED_COMM4:	EQU	50H	
29	00A1	SPEAK:	EQU	0A1H	
30	0000	SONG_P0:	EQU	0	
31	0001	SONG_P1:	EQU	1	
32	0002	SONG_P2:	EQU	2	
33	0003	SONG_ERR:	EQU	3	
34	0004	SONG_MAT:	EQU	4	
35	0005	SONG_RD:	EQU	5	
36	0006	SONG_WR:	EQU	6	
37	0007	SONG_WAI:	EQU	7	;WAIT
38	0008	SONG1_K:	EQU	8	
39	0009	SONG_INT:	EQU	9	
40	000A	SONG_NMI:	EQU	10	
41	000B	SONG_STP:	EQU	11	;STEP
42	000C	SONG_ALU:	EQU	12	
43	000D	SONG_EXEA:	EQU	13	
44	0000 AF	AUTO:	XOR	A	;POWER
45	0001 3D		DEC	A	
46	0002 20 FD		JR	NZ,AUTO+1	
47	0004 C3 AD 00		JP	CHK_REBOOT	
48	0007	SET_INIT:			
49	0007 AF		XOR	A	;RESET CONT ALL
50	0008 32 59 80		LD	(B_ON_SONG),A	;ON SONG
51	000B 32 24 80		LD	(B_SW_COMM3),A	
52	000E 3E 10		LD	A,10H	
53	0010 32 54 80		LD	(B_SPEED_RUN),A	;SPEED RUN
54	0013 3E 01		LD	A,1	
55	0015 32 53 80		LD	(B_SPEED_INT),A	;SET SPEED INT
56	0018	CONT_P:			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

57 0018 3E 4A	LD	A,4AH	
58 001A 32 5A 80	LD	(B_REBOOT),A	
59 001D 31 FF 9F	LD	SP,STACK	
60 0020 3E 90	LD	A,90H	
61 0022 D3 83	OUT	(P_SW+3),A	
62 0024 3E 89	LD	A,89H	
63 0026 D3 93	OUT	(P_SEL_7S+2),A	
64 0028 3E 89	LD	A,10001001B	
65 002A D3 A3	OUT	(0A3H),A	;A=O,B=O,C=I
66 002C 3E 01	LD	A,01	
67 002E D3 A1	OUT	(SPEAK),A	;OSCILLATION
68 0030 3E 02	LD	A,02	
69 0032 4F	LD	C,A	
70 0033 D3 82	OUT	(P_SEL_SW),A	
71 0035 DB 80	IN	A,(P_SW)	
72 0037 47	LD	B,A	
73 0038 CB 47	BIT	0,A	;CLK =REBOOT
74 003A 20 08	JR	NZ,COTIN	
75 003C 3E 00	LD	A,0	
76 003E 32 5A 80	LD	(B_REBOOT),A	
77 0041 C3 07 00	JP	SET_INIT	
78 0044 CD 27 03	COTIN: CALL	ST_REG	
79 0047 CD 2A 10	CALL	DEC_7S	
80 004A 3E 40	LD	A,01000000B	;SET IFF1
81 004C 32 2F 80	LD	(SYS_STP2),A	; SEL 66H,38H
82 004F 3E 01	LD	A,00000001B	; NMI SW REPEAT
83 0051 32 30 80	LD	(SYS_NMI),A	
84 0054 3E 02	LD	A,00000010B	;SET RAM ENABLE
85 0056 32 22 80	LD	(B_SW_COMM1),A	
86			;SET DATA OF ROM ADDRESS 038H,066H


```

117 009C DB 80          IN    A,(P_SW)
118 009E 47             LD     B,A
119 009F CB 47          BIT    0,A          ; MATRIX KEY
120 00A1 CC 98 01       CALL   Z,KEY_MAT1
121 00A4 CD EB 0F       CALL   DISP_7S
122 00A7 CD 1F 11       CALL   DISP_L
123 00AA C3 6D 00       JP     ST_MAIN
124 00AD                CHK_REBOOT:
125 00AD 3A 5A 80       LD     A,(B_REBOOT)
126 00B0 FE 4A         CP     4AH
127 00B2 CA 18 00       JP     Z,CONT_P
128 00B5 C3 07 00       JP     SET_INIT
129                ;CHK IFF1 NMI,INT JUMP SERVICE ROUTINE
130 00B8 21 2E 80       NI_SUB: LD     HL,SYSFLX
131 00BB CB 6E          BIT     5,(HL)      ;IF LAST STATE
132 00BD C0             RET     NZ          ;CYCLE
133 00BE 21 2F 80       LD     HL,SYS_STP2
134 00C1 CB 6E          BIT     5,(HL)
135 00C3 C8             RET     Z           ;IFF1=0 'RETURN'
136 00C4 CB B6          RES     6,(HL)      ;NMI FORM
137 00C6 21 24 80       LD     HL,B_SW_COMM3
138 00C9 CB 5E          BIT     3,(HL)      ;LED 'NMI=1'
139 00CB C2 70 01       JP     NZ,R_INTS
140 00CE CB 56          BIT     2,(HL)      ;LED 'INT=1'
141 00D0 C8             RET     Z
142 00D1 21 24 80       LD     HL,B_SW_COMM3
143 00D4 CD E0 03       CALL   DELAY_INT
144 00D7 CB 96          RES     2,(HL)
145 00D9 21 2F 80       LD     HL,SYS_STP2
146 00DC CB F6          SET     6,(HL)

```

```

147                                ;PROCESS INTERRUPT SERVICE ROUTINE
148                                ;SAVE PC TO (SP)
149 00DE                            R_INT:
150 00DE DD 21 16 80                LD    IX,B_DB
151 00E2 DD CB 19 AE                RES    5,(IX+25)    ;SET IFF1='0'
152 00E6 DD 35 02                  DEC    (IX+2)    ;SP=SP-1
153 00E9 CD E0 03                  CALL   DELAY_INT
154 00EC DD 7E 02                  LD     A,(IX+2)
155 00EF DD 77 01                  LD     (IX+1),A    ;AB<=SP
156 00F2 CD E0 03                  CALL   DELAY_INT
157 00F5 DD 7E 03                  LD     A,(IX+3)
158 00F8 DD 77 00                  LD     (IX),A      ;DB<=PC
159 00FB CD E0 03                  CALL   DELAY_INT
160 00FE DD 4E 00                  LD     C,(IX)      ;rC=DB
161 0101 21 00 86                  LD     HL,RAM
162 0104 DD 7E 01                  LD     A,(IX+1)
163 0107 6F                        LD     L,A
164 0108 71                        LD     (HL),C      ;(AB)<=DB
165 0109 DD 71 0B                  LD     (IX+11),C   ;D0-D7<=DB
166 010C 3A 17 80                  LD     A,(B_AB)
167 010F DD 77 0A                  LD     (IX+10),A   ;A0-A7<=AB
168 0112 DD 36 0C 1A              LD     (IX+12),00011010B ;WR,MREQ
169 0116 CD E0 03                  CALL   DELAY_INT
170 0119 DD 34 01                  INC     (IX+1)      ;AB=AB+1
171 011C AF                        XOR     A
172 011D DD 77 0A                  LD     (IX+10),A   ;A0-A7=0
173 0120 DD 77 0B                  LD     (IX+11),A   ;D0-D7=0
174 0123 DD 77 0C                  LD     (IX+12),A   ;CLR COMM1
175 0126 CD E0 03                  CALL   DELAY_INT
176 0129 DD 36 00 66              LD     (IX),66H

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

177 012D DD CB 0C F6      SET    6,(IX+12)      ;LED 'ROM' ON
178 0131 DD CB 19 76      BIT     6,(IX+25)      ;IF=0=NMI,66H
179 0135 28 04            JR      Z,DB_NMI
180 0137 DD 36 00 38      LD      (IX),38H      ;DB=38H IF INT
181 013B 3A 16 80      DB_NMI: LD      A,(B_DB)
182 013E DD 77 0B      LD      (IX+11),A      ;D0-D7<=DB
183 0141 F5            PUSH    AF
184 0142 CD E0 03      CALL    DELAY_INT
185 0145 F1            POP     AF
186 0146 DD 77 03      LD      (IX+3),A      ;PC<=DB
187 0149 DD 36 0C 40      LD      (IX+12),01000000B
188 014D DD 36 0B 00      LD      (IX+11),0      ;D0-D7=0
189 0151 CD E0 03      CALL    DELAY_INT
190 0154 3A 19 80      LD      A,(B_PC)
191 0157 DD 77 01      LD      (IX+1),A      ;AB<=PC
192 015A CD E0 03      CALL    DELAY_INT
193 015D DD 34 03      INC     (IX+3)      ;PC=PC+1
194 0160 CD E0 03      CALL    DELAY_INT
195 0163 CD 1C 08      CALL    RUN
196 0166 CD E0 03      CALL    DELAY_INT
197 0169 CD 1C 08      CALL    RUN
198 016C CD E0 03      CALL    DELAY_INT
199 016F C9            RET
200 0170 FD 21 30 80      R_INTS: LD      IY,SYS_NMI
201 0174 FD CB 00 46      BIT     0,(IY)      ;CAN'T PRESSED
202 0178 C8            RET     Z      ;NMI SW
203 0179 FD CB 00 86      RES     0,(IY)
204 017D C3 DE 00      JP      R_INT
205                                ;IF INTERRUPT WHEN CHANGE RAM TO ROM
206                                ;CHANGE RAM ROM

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

207 0180 21 00 86      CH_RA_RO: LD    HL, RAM
208 0183 3A 22 80      LD    A, (B_SW_COMM1)
209 0186 CB 77         BIT    6, A
210 0188 C8           RET    Z           ;IF RAM
211 0189 21 00 85      LD    HL, ROM
212 018C C9           RET
213                   ;SHOW DATA FROM ADDR AT MEMORY (RAM)
214 018D CD 57 12      CUR_MEM: CALL  P_MEM
215 0190 7E           LD    A, (HL)
216 0191 32 14 80      LD    (B_DA_M), A
217 0194 CD 2A 10      CALL  DEC_7S
218 0197 C9           RET
219 0198 CD E6 08      KEY_MAT1: CALL  SBA           ;SAVE BUFFER ALL
220 019B 06 0A         LD    B, 10
221 019D CD EA 03      CALL  DELAY_CPU
222 01A0 CD 48 11      CALL  KEY_MAT
223 01A3 CD F2 08      CALL  MBC           ;MOVE BUFFER
COMEBACK
224 01A6 CD 8D 01      CALL  CUR_MEM
225 01A9 C9           RET
226                   ;<<<<<<<<< ARITHIMATRIX UNIT >>>>>>>>>;
227 01AA              M_ALU:
228 01AA 21 23 80      LD    HL, B_SW_COMM2
229 01AD CB CE         SET    1, (HL)
230 01AF 2B           DEC    HL
231 01B0 CB 8E         RES    1, (HL)           ;CLEAR RAM LED
232 01B2 C5           PUSH   BC
233 01B3 CD E6 08      CALL  SBA           ;SAVE BUFFER ALL
234 01B6 3E 0C         LD    A, SONG_ALU       ;SOUND ALU

```

TITLE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

235 01B8 CD C0 10      CALL SONG
236 01BB 06 B7         LD B,10110111B ;SEL ACC,DB,TEMP
237 01BD CD 6E 10      CALL MODE_7S
238 01C0 C1            POP BC
239 01C1 CD CC 0F      CALL REP_SW ;CHECK REPEAT
SW USE ALU
240                      ; START ALU
241 01C4 CD EB 0F      M_ALU1: CALL DISP_7S
242 01C7 CD 1F 11      CALL DISP_L
243 01CA 3E FF         LD A,0FFH ;SHOW 7S
244 01CC 32 2B 80      LD (MODE_SW),A
245 01CF 0E 01         LD C,01
246 01D1 21 21 80      LD HL,B_SW_DATA
247 01D4 CD 95 0F      CALL SWITCH_P
248 01D7 CD 15 02      CALL MOV_DB
249 01DA 3E 1C         LD A,00011100B ;SET S0,S1,S2
250 01DC 32 2B 80      LD (MODE_SW),A ;ALU
251 01DF 0E 03         LD C,03
252 01E1 21 23 80      LD HL,B_SW_COMM2
253 01E4 CD 95 0F      CALL SWITCH_P
254 01E7 79            LD A,C ;PRESS WRITE ACC
255 01E8 D3 82         OUT (P_SEL_SW),A ;PRESS EXEQUIT
256 01EA DB 80         IN A,(P_SW)
257 01EC 47            LD B,A
258 01ED CB 6F         BIT 5,A ;BIT W_ACC
259 01EF CC 24 02      CALL Z,WR_ACC
260 01F2 CB 77         BIT 6,A ;BIT EXE_A
261 01F4 CC 3F 02      CALL Z,EXE_ACC
262 01F7 CB 4F         BIT 1,A ;BIT USE ALU
263 01F9 C2 C4 01      JP NZ,M_ALU1 ;LOOP IF NOT END

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

264 01FC C5          PUSH BC
265 01FD CD F2 08     CALL MBC          ;BUFFER COMBACK
266 0200 3E 0C        LD A,SONG_ALU ;SOUND TITLE
267 0202 CD C0 10     CALL SONG
268 0205 CD 2A 10     CALL DEC_7S
269 0208 C1          POP BC
270 0209 CD CC 0F     CALL REP_SW ;SW OFF ALU
271 020C 21 23 80     LD HL,B_SW_COMM2
272 020F CB 8E        RES 1,(HL)
273 0211 2B          DEC HL
274 0212 CB CE        SET 1,(HL) ;SET RAM LED
275 0214 C9          RET
276                  ;<<<<< END ALU >>>>>
277 0215 3A 21 80     MOV_DB: LD A,(B_SW_DATA) ;
278 0218 32 16 80     LD (B_DB),A
279 021B CD 2A 10     CALL DEC_7S ;DECODE BINARY
280 021E 06 B7        LD B,10110111B ;SHOW ACC,DB
281 0220 CD 6E 10     CALL MODE_7S
282 0223 C9          RET
283 0224 F5          WR_ACC: PUSH AF
284 0225 3E 00        LD A,SONG_P0
285 0227 CD C0 10     CALL SONG
286 022A 21 23 80     LD HL,B_SW_COMM2 ;SHOW W_ACC
287 022D CB EE        SET 5,(HL)
288 022F CD CC 0F     CALL REP_SW
289 0232 3A 16 80     LD A,(B_DB) ;MOV (B_ACC),(B_DB)
290 0235 32 1B 80     LD (B_ACC),A
291 0238 21 23 80     LD HL,B_SW_COMM2
292 023B CB AE        RES 5,(HL) ;OFF W_ACC LED
293 023D F1          POP AF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

294 023E C9                                RET
295 023F F5                                EXE_ACC:  PUSH  AF
296 0240 CD CC 0F                          CALL  REP_SW      ;WAIT STATE
297 0243 3E 0D                          LD    A,SONG_EXEA
298 0245 CD C0 10                          CALL  SONG
299 0248 21 23 80                          LD    HL,B_SW_COMM2
300 024B CB F6                          SET   6,(HL)      ;LED 'EXE' ON
301 024D 7E                          LD    A,(HL)
302 024E E6 1C                          AND   00011100B
303 0250 0F                          RRCA              ; COMMAND ALU 2
304 0251 21 5B 14                          LD    HL,T_ALU
305 0254 CD 93 08                          CALL  R_ALU
306 0257 F1                          POP   AF
307 0258 C9                          RET
308                                ;FOR SPEED2 WITCH ALU INSTR.
309 0259 21 23 80      C_EXE:  LD    HL,B_SW_COMM2 ; EXECUTE
310 025C CB F6                          SET   6,(HL)      ;LED 'EXE' ON
311 025E 7E                          LD    A,(HL)
312 025F E6 1C                          AND   00011100B
313 0261 0F                          RRCA
314 0262 21 5B 14                          LD    HL,T_ALU
315 0265 CD 93 08                          CALL  R_ALU
316 0268 C9                          RET
317 0269 3E FF      ALU0:  LD    A,0FFH      ;**SET ACC=0FFH
318 026B C3 F1 02                          JP    ALUS
319 026E CD AB 02      ALU1:  CALL  INI_ALU      ;**ACC and DB
320 0271 A0                          AND   A,B
321 0272 C3 B3 02                          JP    ALUS2
322 0275 CD AB 02      ALU2:  CALL  INI_ALU      ;**ACC or DB
323 0278 B0                          OR    A,B

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


```

354 02B3          ALUS2:
355              ;(B_DB)=>(B_TEM)+DELAY+(B_ACC)=>(B_DB)
356 02B3  F5          PUSH  AF
357 02B4  E1          POP   HL
358 02B5  78          LD    A,B
359 02B6  32 1D 80     LD    (B_TEM),A
360 02B9  CD E3 02     CALL  A_DE_DI      ;ALU DELAY
361 02BC  7C          LD    A,H
362 02BD  32 21 80     LD    (B_SW_DATA),A
363 02C0  32 16 80     LD    (B_DB),A
364 02C3  7D          LD    A,L
365 02C4  32 1E 80     LD    (B_FLX),A
366 02C7  21 23 80     LD    HL,B_SW_COMM2
367 02CA  CB EE       SET   5,(HL)      ;LED 'WRA' ON
368 02CC  CD E3 02     CALL  A_DE_DI
369 02CF  3A 21 80     LD    A,(B_SW_DATA)
370 02D2  32 1B 80     LD    (B_ACC),A
371 02D5  CD E3 02     CALL  A_DE_DI
372 02D8  21 23 80     LD    HL,B_SW_COMM2  ;'WRA' OFF
373 02DB  CB AE       RES   5,(HL)
374 02DD  3A 1E 80     LD    A,(B_FLX)
375 02E0  C3 FA 02     JP    ALUS3S
376 02E3  CD 2A 10     A_DE_DI: CALL  DEC_7S
377 02E6  06 B7       LD    B,10110111B
378 02E8  CD 6E 10     CALL  MODE_7S
379 02EB  06 32       LD    B,50
380 02ED  CD EA 03     CALL  DELAY_CPU
381 02F0  C9          RET
382 02F1  F5          ALUS:  PUSH  AF
383 02F2  E1          POP   HL

```

384	02F3	32 1B 80		LD	(B_ACC),A	
385	02F6	7D	ALUS3:	LD	A,L	
386	02F7	32 1E 80		LD	(B_FLX),A	
387	02FA	21 2E 80	ALUS3S:	LD	HL,SYSFLX	
388	02FD	CB D6		SET	2,(HL)	;SET DATA VARY
389	02FF	47		LD	B,A	
390	0300	CB 18		RR	B	
391	0302	3E 00		LD	A,0	
392	0304	CB 1F		RR	A	
393	0306	CB 10		RL	B	
394	0308	CB 10		RL	B	
395	030A	CB 1F		RR	A	
396	030C	CB 10		RL	B	
397	030E	CB 1F		RR	A	
398	0310	47		LD	B,A	
399	0311	3A 24 80		LD	A,(B_SW_COMM3)	
400	0314	E6 1F		AND	0001111B	;CLR OLD FLAX
401	0316	B0		OR	B	
402	0317	32 24 80		LD	(B_SW_COMM3),A	
403	031A	06 0A		LD	B,10	
404	031C	CD EA 03		CALL	DELAY_CPU	
405	031F	21 23 80		LD	HL,B_SW_COMM2	
406	0322	7E		LD	A,(HL)	
407	0323	E6 02		AND	A,00000010B	;CLR S0,S1,S2
408	0325	77		LD	(HL),A	;LED 'EXE' OFF
409	0326	C9		RET		
410	0327	AF	ST_REG:	XOR	A	
411	0328	21 00 80		LD	HL,B_7S	
412	032B	11 01 80		LD	DE,B_7S+1	
413	032E	01 2F 00		LD	BC,SYSFLX-B_7S+1	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

414	0331	77	LD	(HL),A	
415	0332	ED B0	LDIR		;CLEAR ALL BUFFER=>0
416	0334	3E FF	LD	A,0FFH	
417	0336	32 18 80	LD	(B_SP),A	
418	0339	06 14	LD	B,20	
419	033B	CD EA 03	CALL	DELAY_CPU	
420	033E	AF	XOR	A	
421	033F	32 17 80	LD	(B_AB),A	
422	0342	3C	INC	A	
423	0343	32 19 80	LD	(B_PC),A	
424	0346	CD C9 08	CALL	CUR_D_A	
425	0349	C9	RET		
426	034A	3A 24 80	CHE_CPU_SW:	LD	A,(B_SW_COMM3)
427	034D	CB 67	BIT	4,A	;SPEED
428	034F	C2 5D 03	JP	NZ,CSSTP	;CHECK SW STEP
429	0352	3E 04	LD	A,4	;CHK RUN PRESS
430	0354	D3 82	OUT	(P_SEL_SW),A	
431	0356	DB 80	IN	A,(P_SW)	
432	0358	CB 47	BIT	0,A	
433	035A	CA 8A 03	JP	Z,R_CPU	;GOTO RUN CPU
434	035D	3E 04	CSSTP:	LD	A,4
435	035F	4F	LD	C,A	
436	0360	D3 82	OUT	(P_SEL_SW),A	
437	0362	DB 80	IN	A,(P_SW)	
438	0364	47	LD	B,A	
439	0365	CB 4F	BIT	1,A	
440	0367	CA B8 03	JP	Z,S_CPU	;GOTO STEP CPU
441	036A	3E 10	LD	A,00010000B	;SPEED ONLY
442	036C	32 2B 80	LD	(MODE_SW),A	
443	036F	21 24 80	LD	HL,B_SW_COMM3	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

444 0372 0E 04          LD    C,04
445 0374 CD 95 0F          CALL SWITCH_P
446 0377 3A 24 80          LD    A,(B_SW_COMM3)
447 037A CB 67            BIT    4,A
448 037C 28 08            JR     Z,CSSTP1
449 037E 3E 00            LD     A,0
450 0380 32 20 80          LD     (B_SW_ADDR),A
451 0383 32 21 80          LD     (B_SW_DATA),A
452 0386 CD 05 04          CSSTP1: CALL NMICHE      ;CHK INT,NMI
453 0389 C9              RET
454 038A                R_CPU:
455 038A 21 2E 80          LD     HL,SYSFLX
456 038D CB FE            SET    7,(HL)
457 038F 21 24 80          LD     HL,B_SW_COMM3
458 0392 CB C6            SET    0,(HL)      ;LED 'RUN' ON
459 0394                R_CPU1:
460 0394 3E 00            LD     A,0
461 0396 32 23 80          LD     (B_SW_COMM2),A
462 0399 32 25 80          LD     (B_SW_COMM4),A
463 039C CD FF 03          CALL  DELAY_CPUS
464 039F CD 1C 08          CALL  RUN
465                        ;SHOW ADDRESS MEMORY & DATA MEMORT
466                        ;CURRENT DATA MEMORY BY 'B_AB'=>'B_AD_M'
467 03A2 CD C9 08          CALL  CUR_D_A
468 03A5 CD 05 04          CALL  NMICHE
469 03A8 CD B8 00          CALL  NI_SUB
470 03AB 21 2E 80          LD     HL,SYSFLX
471 03AE CB 7E            BIT    7,(HL)
472 03B0 20 E2            JR     NZ,R_CPU1
473 03B2 21 24 80          LD     HL,B_SW_COMM3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

474 03B5 CB 86 RES 0,(HL) ;LED 'RUN' OFF
475 03B7 C9 RET
476 03B8 S_CPU:
477 03B8 21 24 80 LD HL,B_SW_COMM3
478 03BB CB 66 BIT 4,(HL)
479 03BD C2 22 04 JP NZ,S2_CPU ;STEP OF SPEED 2
480 03C0 CB CE SET 1,(HL)
481 03C2 C5 PUSH BC ;STEP OF SPEED 1
482 03C3 06 0A LD B,10
483 03C5 CD EA 03 CALL DELAY_CPU
484 03C8 3E 0B LD A,SONG_STP
485 03CA CD C0 10 CALL SONG
486 03CD CD 1C 08 CALL RUN
487 03D0 CD C9 08 CALL CUR_D_A
488 03D3 C1 POP BC
489 03D4 CD CC 0F CALL REP_SW
490 03D7 21 24 80 LD HL,B_SW_COMM3
491 03DA CB 8E RES 1,(HL)
492 03DC CD 05 04 CALL NMICHE
493 03DF C9 RET
494 03E0 DELAY_INT:
495 03E0 CD C9 08 CALL CUR_D_A
496 03E3 3A 53 80 LD A,(B_SPEED_INT)
497 03E6 47 LD B,A
498 03E7 CD 2A 10 DELAY_CPU1: CALL DEC_7S
499 03EA CD 1F 11 DELAY_CPU: CALL DISP_L
500 03ED CD EB 0F CALL DISP_7S
501 03F0 E5 PUSH HL
502 03F1 21 24 80 LD HL,B_SW_COMM3
503 03F4 CB 46 BIT 0,(HL)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[illegible]

```

533 0422          S2_CPU:
534          ;          LD  A,2
535          ;          CALL STP2_COMD
536 0422 21 2F 80          LD  HL,SYS_STP2
537 0425 CB 86          RES  0,(HL)      ;SET INSTR. 2 BYTE
538 0427 11 2D 04          LD  DE,S21_CPU
539 042A C3 DF 06          JP   RDI_PDC      ;RECI DATA INSTR.
540 042D 3A 1C 80          S21_CPU: LD  A,(B_IR)
541 0430 CD BE 08          CALL SRCHIN      ;FIND INSTR.
542 0433 21 3A 04          LD  HL,T_CPU2      ;TABLE SPEED2
543 0436 78          LD  A,B
544 0437 C3 92 08          JP   R_CPUS2      ;JMP INSTR.
545          ;TABLE ROUTINE INSTRUCTION OF CPU SPEED 2
546 043A BA04 2D05          T_CPU2: DW  I00,I7D      ;0,1
547 043E BD04 2E05          DW  I7E,I6F      ;2,3
548 0442 BE04 5305          DW  I6E,IF5      ;4,5
549 0446 BF04 5405          DW  IE5,IED      ;6,7
550 044A C004 5505          DW  IF1,IE1      ;8,9
551 044E C104 5605          DW  IFD,I3C      ;A,B
552 0452 C204 5705          DW  I2C,I3D      ;C,D
553 0456 C304 5805          DW  I2D,I87      ;E,F
554 045A C404 5905          DW  I85,I86      ;10,11
555 045E EB04 5A05          DW  I97,I95      ;12,13
556 0462 EC04 5B05          DW  I96,IBF      ;14,15
557 0466 ED04 5C05          DW  IBD,IBE      ;16,17
558 046A EE04 5D05          DW  IA7,IA5      ;18,19
559 046E EF04 5E05          DW  IA6,IB7      ;1A,1B
560 0472 F004 5F05          DW  IB5,IB6      ;1C,1D
561 0476 F104 6005          DW  IAF,IAD      ;1E,1F
562 047A F204 6105          DW  IAE,IC9      ;20,21

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

563	047E	F304 8605	DW	I76,IFB	;22,23
564	0482	F404 8705	DW	IF3,I2F	;24,25
565	0486	F704 F604	DW	IDF,I4D	;26S1,26S2
566	048A	F504 8805	DW	I45,I3EN	;26,27
567	048E	F804 C705	DW	I3AN,I32N	;28,29
568	0492	F904 3406	DW	I2EN,I31N	;2A,2B
569	0496	0B05 3506	DW	IDBN,ID3N	;2C,2D
570	049A	0C05 3606	DW	IC6N,ID6N	;2E,2F
571	049E	0D05 3706	DW	IFEN,IE6N	;30,31
572	04A2	2705 3806	DW	IF6N,IEEN	;32,33
573	04A6	2805 3906	DW	IC3N,ICAN	;34,35
574	04AA	2905 3A06	DW	IC2N,IDAN	;36,37
575	04AE	2A05 3B06	DW	ID2N,ICDN	;38,39
576	04B2	2B05 BB06	DW	ICCN,IC4N	;3A,3B
577	04B6	2C05 BC06	DW	IDCN,ID4N	;3C,3D
578	04BA	C3 BD 06	I00:	JP	CPC_I
579	04BD	C9	I7E:	RET	
580	04BE	C9	I6E:	RET	
581	04BF	C9	IE5:	RET	
582	04C0	C9	IF1:	RET	
583	04C1	C9	IFD:	RET	
584	04C2	C9	I2C:	RET	
585	04C3	C9	I2D:	RET	
586	04C4	01 CA 04	I85:	LD	BC,I851
587	04C7	C3 2F 07		JP	STP2
588	04CA	3A 1A 80	I851:	LD	A,(B_BR) ;Br=>DB
589	04CD	32 16 80		LD	(B_DB),A
590	04D0	01 D6 04		LD	BC,I852
591	04D3	C3 2F 07		JP	STP2
592	04D6	3E 12	I852:	LD	A,00010010B ;COMM 'ADD'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

593	04D8	21 23 80	LD	HL,B_SW_COMM2	
594	04DB	77	LD	(HL),A	
595	04DC	CD 59 02	CALL	C_EXE	;CPU EXECUTE
596	04DF	3A 1B 80	LD	A,(B_ACC)	
597	04E2	32 21 80	LD	(B_SW_DATA),A	;ACC=>D0-D7
598	04E5	32 16 80	LD	(B_DB),A	
599	04E8	C3 BD 06	JP	CPC_I	
600	04EB	C9	I97:	RET	
601	04EC	C9	I96:	RET	
602	04ED	C9	IBD:	RET	
603	04EE	C9	IA7:	RET	
604	04EF	C9	IA6:	RET	
605	04F0	C9	IB5:	RET	
606	04F1	C9	IAF:	RET	
607	04F2	C9	IAE:	RET	
608	04F3	C9	I76:	RET	
609	04F4	C9	IF3:	RET	
610	04F5	C9	I45:	RET	
611	04F6	C9	I4D:	RET	
612	04F7	C9	IDF:	RET	
613	04F8	C9	I3AN:	RET	
614			I2EN:	;<<<LD B,n;>>>	
615	04F9	CD A2 05	CALL	I3E3	
616	04FC	01 02 05	LD	BC,I2E1	
617	04FF	C3 2F 07	JP	STP2	
618	0502	3A 16 80	I2E1: LD	A,(B_DB)	;**MOV B,n
619	0505	32 1A 80	LD	(B_BR),A	
620	0508	C3 BD 06	JP	CPC_I	
621	050B	C9	IDBN:	RET	
622	050C	C9	IC6N:	RET	

```

623                                IFEN:                                ;** CP A,n
624 050D CD A2 05                CALL I3E3
625 0510 01 16 05                LD BC,IFE1
626 0513 C3 2F 07                JP STP2
627 0516 3E 16                IFE1: LD A,00010110B ;COMMAND '
SUB',USE ALU
628 0518 21 2F 80                LD HL,SYS_STP2
629 051B CB E6                SET 4,(HL) ;SET USE 'CP' INSTR.
630 051D 21 23 80                LD HL,B_SW_COMM2
631 0520 77                LD (HL),A
632 0521 CD 59 02                CALL C_EXE ;CPU EXECUTE
633 0524 C3 BD 06                JP CPC_I
634 0527 C9                IF6N: RET
635 0528 C9                IC3N: RET
636 0529 C9                IC2N: RET
637 052A C9                ID2N: RET
638 052B C9                ICCN: RET
639 052C C9                IDCN: RET
640 052D C9                I7D: RET
641                                I6F: ;<<< LD B,A >>>;
642 052E 01 34 05                LD BC,I6F1
643 0531 C3 2F 07                JP STP2
644 0534 3A 1B 80                I6F1: LD A,(B_ACC)
645 0537 32 21 80                LD (B_SW_DATA),A ;D0-D7=(B_ACC)
646 053A 21 25 80                LD HL,B_SW_COMM4
647 053D CB C6                SET 0,(HL) ;RDA=1
648 053F 01 45 05                LD BC,I6F2
649 0542 C3 2F 07                JP STP2
650 0545 3A 1B 80                I6F2: LD A,(B_ACC) ;**MOV B,A
651 0548 32 1A 80                LD (B_BR),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

652	054B	3E 01	LD	A,1	;CLR RDA,D0-S7
653	054D	CD 5A 07	CALL	STP2_COMD	
654	0550	C3 BD 06	JP	CPC_I	
655	0553	C9	IF5:	RET	
656	0554	C9	IED:	RET	
657	0555	C9	IE1:	RET	
658	0556	C9	I3C:	RET	
659	0557	C9	I3D:	RET	
660	0558	C9	I87:	RET	
661	0559	C9	I86:	RET	
662	055A	C9	I95:	RET	
663	055B	C9	IBF:	RET	
664	055C	C9	IBE:	RET	
665	055D	C9	IA5:	RET	
666	055E	C9	IB7:	RET	
667	055F	C9	IB6:	RET	
668	0560	C9	IAD:	RET	
669	0561	01 67 05	IC9:	LD	BC,IC91
670	0564	C3 2F 07	JP	STP2	
671	0567	3A 18 80	IC91:	LD	A,(B_SP)
672	056A	32 17 80	LD	(B_AB),A	;SP=>AB
673	056D	01 73 05	LD	BC,IC92	
674	0570	C3 2F 07	JP	STP2	
675	0573	21 18 80	IC92:	LD	HL,B_SP ;SP=SP+1
676	0576	34	INC	(HL)	
677	0577	11 7D 05	LD	DE,IC93	
678	057A	C3 DF 06	JP	RDI_PDC	
679	057D	3A 16 80	IC93:	LD	A,(B_DB) ;DB=>PC
680	0580	32 19 80	LD	(B_PC),A	
681	0583	C3 BD 06	JP	CPC_I	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

682 0586 C9          IFB:      RET
683 0587 C9          I2F:      RET
684 0588 CD A2 05     I3EN:      CALL  I3E3
685                  ;MOV A,n EXECUIT
686 058B 01 91 05     I3E8: LD    BC,I3E8+6
687 058E C3 2F 07     JP      STP2
688 0591 3A 16 80     LD      A,(B_DB)
689 0594 32 21 80     LD      (B_SW_DATA),A
690 0597 21 23 80     LD      HL,B_SW_COMM2
691 059A CB EE        SET     5,(HL)      ;WRA=1
692 059C 32 1B 80     LD      (B_ACC),A      ;** MOV A,n
693 059F C3 BD 06     JP      CPC_I      ;<<< END 3E >>>
694                  ;((( ***** )))
695 05A2 01 A8 05     I3E3: LD    BC,I3E3+6
696 05A5 C3 2F 07     JP      STP2
697 05A8 3A 19 80     LD      A,(B_PC)      ;<<< PC=AB >>>
698 05AB 32 17 80     LD      (B_AB),A
699 05AE 01 B4 05     I3E4: LD    BC,I3E4+6
700 05B1 C3 2F 07     JP      STP2
701 05B4 3A 19 80     LD      A,(B_PC)
702 05B7 3C          INC     A      ;<<< PC+1 >>>
703 05B8 32 19 80     LD      (B_PC),A
704 05BB 21 2F 80     LD      HL,SYS_STP2
705 05BE CB C6        SET     0,(HL)      ;INSTR. 2 BYTE
706 05C0 11 C6 05     I3E7: LD    DE,I3E7+6
707 05C3 C3 DF 06     JP      RDI_PDC
708 05C6 C9          RET
709 05C7 01 CD 05     I32N: LD    BC,I321
710 05CA C3 2F 07     JP      STP2
711 05CD 3A 19 80     I321: LD    A,(B_PC)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

712 05D0 32 17 80	LD (B_AB),A ;<< AB=PC >>
713 05D3 01 D9 05	LD BC,I322
714 05D6 C3 2F 07	JP STP2
715 05D9 3A 19 80	I322: LD A,(B_PC)
716 05DC 3C	INC A
717 05DD 32 19 80	LD (B_PC),A ;<< PC=PC+1 >>
718 05E0 CD 2A 10	CALL DEC_7S
719 05E3 11 E9 05	LD DE,I323
720 05E6 C3 DF 06	JP RDI_PDC
721 05E9 01 EF 05	I323: LD BC,I324
722 05EC C3 2F 07	JP STP2
723 05EF 3A 16 80	I324: LD A,(B_DB)
724 05F2 32 17 80	LD (B_AB),A ;<< AB<=DB >>
725 05F5 01 FB 05	LD BC,I325
726 05F8 C3 2F 07	JP STP2
727 05FB 3A 1B 80	I325: LD A,(B_ACC) ; D0-D7<= ACC
728 05FE 32 21 80	LD (B_SW_DATA),A
729 0601 21 25 80	LD HL,B_SW_COMM4
730 0604 CB C6	SET 0,(HL) ; RDA = 1
731 0606 01 0C 06	LD BC,I326
732 0609 C3 2F 07	JP STP2
733 060C 21 20 80	I326: LD HL,B_SW_ADDR
734 060F 3A 19 80	LD A,(B_PC) ;<< A0-A7<=PC >>
735 0612 77	LD (HL),A
736 0613 23	INC HL
737 0614 23	INC HL ;HL=B_SW_COMM1
738 0615 CB E6	SET 4,(HL) ;WR = 1
739 0617 CB DE	SET 3,(HL) ;MREQ = 1
740 0619 21 25 80	LD HL,B_SW_COMM4
741 061C CB 86	RES 0,(HL) ;RDA = 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

742 061E 21 00 86          LD    HL, RAM
743 0621 3A 16 80          LD    A, (B_DB)      ;POINT ADDRESS
744 0624 CD 5D 12          CALL  SRCHAD
745 0627 3A 1B 80          LD    A, (B_ACC)
746 062A 77              LD    (HL), A
747 062B 32 16 80          LD    (B_DB), A
748 062E CD 8D 01          CALL  CUR_MEM      ;DATA CURRENT
749 0631 C3 BD 06          JP    CPC_I
750                      ;((((((( ***** ))))))))
751 0634 C9              I31N:  RET
752 0635 C9              ID3N:  RET
753 0636 C9              ID6N:  RET
754 0637 C9              IE6N:  RET
755 0638 C9              IEEN:  RET
756 0639 C9              ICAN:  RET
757 063A C9              IDAN:  RET
758 063B              ICDN:
759 063B 01 41 06          LD    BC, ICD1
760 063E C3 2F 07          JP    STP2
761 0641 21 18 80          ICD1: LD    HL, B_SP
762 0644 35              DEC    (HL)      ;SP=SP-1
763 0645 01 4B 06          LD    BC, ICD2
764 0648 C3 2F 07          JP    STP2
765 064B 21 18 80          ICD2: LD    HL, B_SP
766 064E 7E              LD    A, (HL)
767 064F 32 17 80          LD    (B_AB), A
768 0652 01 58 06          LD    BC, ICD3
769 0655 C3 2F 07          JP    STP2
770 0658 3A 19 80          ICD3: LD    A, (B_PC)
771 065B 3C              INC    A      ;DB=PC+1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

772	065C	32 16 80	LD	(B_DB),A	
773	065F	01 65 06	LD	BC,ICD4	
774	0662	C3 2F 07	JP	STP2	
775	0665	21 16 80	ICD4: LD	HL,B_DB	
776	0668	7E	LD	A,(HL)	;DB => D0-D7
777	0669	32 21 80	LD	(B_SW_DATA),A	
778	066C	23	INC	HL	;AB => A0-A7
779	066D	7E	LD	A,(HL)	
780	066E	32 20 80	LD	(B_SW_ADDR),A	
781	0671	3A 22 80	LD	A,(B_SW_COMM1)	
782	0674	F6 1A	OR	00011010B	;WR,MREQ='1'
783	0676	32 22 80	LD	(B_SW_COMM1),A	
784			;PUSH (B_DB) TO STACK		
785	0679	21 00 86	LD	HL,RAM	
786	067C	3A 17 80	LD	A,(B_AB)	
787	067F	6F	LD	L,A	
788	0680	3A 16 80	LD	A,(B_DB)	
789	0683	77	LD	(HL),A	;HL=STACK
790	0684	01 8A 06	LD	BC,ICD5	
791	0687	C3 2F 07	JP	STP2	
792	068A	AF	ICD5: XOR	A	
793	068B	32 22 80	LD	(B_SW_COMM1),A	;
794	068E	32 20 80	LD	(B_SW_ADDR),A	
795	0691	32 21 80	LD	(B_SW_DATA),A	
796	0694	3A 19 80	LD	A,(B_PC)	; PC=>AB
797	0697	32 17 80	LD	(B_AB),A	
798	069A	01 A0 06	LD	BC,ICD6	
799	069D	C3 2F 07	JP	STP2	
800	06A0	11 AC 06	ICD6: LD	DE,ICD7	
801	06A3	3A 16 80	LD	A,(B_DB)	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

802 06A6 32 19 80      LD      (B_PC),A      ;DB=>PC
803 06A9 C3 DF 06      JP      RDI_PDC
804 06AC 01 B2 06      ICD7: LD      BC,ICD8
805 06AF C3 2F 07      JP      STP2
806 06B2 3A 16 80      ICD8: LD      A,(B_DB)      ;DB=>PC
807 06B5 32 19 80      LD      (B_PC),A
808 06B8 C3 BD 06      JP      CPC_I
809 06BB C9            IC4N: RET
810 06BC C9            ID4N: RET
811                    ;SET (B_AB) <= (B_PC)
812                    ;WAIT PUSH STEP
813                    ;SET (B_PC) <= PC+1
814 06BD                    CPC_I:
815 06BD 01 C3 06      LD      BC,CPC_I1
816 06C0 C3 2F 07      JP      STP2
817 06C3 3A 19 80      CPC_I1: LD      A,(B_PC)
818 06C6 32 17 80      LD      (B_AB),A
819 06C9 3E 02      LD      A,2      ;CLEAR BUS ALL
820 06CB CD 5A 07      CALL   STP2_COMD
821 06CE 01 D4 06      LD      BC,CPC_I2
822 06D1 C3 2F 07      JP      STP2
823 06D4 3A 19 80      CPC_I2: LD      A,(B_PC)
824 06D7 3C            INC      A
825 06D8 32 19 80      LD      (B_PC),A
826 06DB CD 2A 10      CALL   DEC_7S
827 06DE C9            RET
828                    ;USE (B_AB) POINT DATA
829                    ;READ DATA INSTRUCTION AND PUSH DATA
830                    ;TO B_DB AND CLEAR LED BUS DATA,ADDR,COMM1

```

831 06DF RDI_PDC:

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

832
833 06DF D5          PUSH  DE
834 06E0 01 E6 06    LD     BC,RDI11_1    ;FOR RET---BC1
835 06E3 C3 2F 07    JP     STP2
836 06E6 21 20 80    RDI11_1: LD     HL,B_SW_ADDR
837 06E9 3A 17 80    LD     A,(B_AB)
838 06EC 77          LD     (HL),A      ;LOAD TO A0-A7
839 06ED E5          PUSH  HL
840 06EE 21 00 86    LD     HL,RAM
841 06F1 CD 5D 12    CALL  SRCHAD
842 06F4 7E          LD     A,(HL)      ;A=DATA OF (B_AB)
843 06F5 E1          POP     HL
844 06F6 23          INC     HL      ;HL=B_SW_DATA
845 06F7 77          LD     (HL),A
846 06F8 23          INC     HL      ;HL=B_SW_COMM1
847 06F9 3E 2A      LD     A,00101010B    ;'RD','MREQ' =1
848 06FB 77          LD     (HL),A
849 06FC 01 02 07    PDC1: LD     BC,PDC1+6    ;FOR RET---BC2
850 06FF C3 2F 07    JP     STP2
851 0702 3A 21 80    LD     A,(B_SW_DATA)
852 0705 32 16 80    LD     (B_DB),A      ;D0-D7=> B_DB >>
853 0708 3A 2F 80    LD     A,(SYS_STP2)
854 070B CB 47      BIT     0,A
855 070D 20 13      JR     NZ,PDC2      ;CHECK 2 BYTE
856                                     ;INSTR. 2 BYTE
857 070F CD 2A 10    CALL  DEC_7S
858 0712 06 0A      LD     B,10
859 0714 CD EA 03    CALL  DELAY_CPU
860 0717 3A 16 80    LD     A,(B_DB)      ;(B_DB)=>(B_IR)
861 071A 32 1C 80    LD     (B_IR),A

```

```

862 071D 21 2F 80          LD    HL,SYS_STP2
863 0720 CB C6             SET    0,(HL)      ;CLR BIT 2 BYTE
864 0722                  PDC2:
865 0722 21 2F 80          LD    HL,SYS_STP2
866 0725 AF               XOR    A      ;CLEAR
867 0726 21 20 80          LD    HL,B_SW_ADDR
868 0729 77               LD    (HL),A
869 072A 23               INC    HL
870 072B 77               LD    (HL),A
871 072C 23               INC    HL
872 072D 77               LD    (HL),A
873 072E C9               RET
874                      ;REPEAT UNTIL PRESS STEP SW.
875 072F                  STP2:
876 072F C5               PUSH    BC
877 0730 CD 2A 10          CALL    DEC_7S
878 0733 0E 04            STP2S: LD    C,4
879 0735 79               LD    A,C
880 0736 D3 82            OUT    (P_SEL_SW),A
881 0738 DB 80            IN     A,(P_SW)
882 073A 47               LD    B,A
883 073B CB 4F            BIT     1,A
884 073D 20 12            JR     NZ,LOP_STP ;EXIT PRESS STEP
885 073F 3E 01            LD     A,SONG_P1
886 0741 CD C0 10          CALL    SONG
887 0744 21 24 80          LD     HL,B_SW_COMM3
888 0747 E5               PUSH    HL
889 0748 CB CE            SET     1,(HL)
890 074A CD CC 0F          CALL    REP_SW
891 074D E1               POP     HL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

892 074E CB 8E RES 1,(HL)
893 0750 C9 RET
894 0751 CD 1F 11 LOP_STP: CALL DISP_L
895 0754 CD EB 0F CALL DISP_7S
896 0757 C3 33 07 JP STP2S
897 ;**** OPERATE FOR A COMMAND
898 075A STP2_COMD:
899 075A 21 7E 07 LD HL,T_COMS2
900 075D C3 92 08 JP R_CPUS2
901 0760 3E 00 C_WA_BD: LD A,0 ;CLR WRA,D0-D7
902 0762 32 21 80 LD (B_SW_DATA),A
903 0765 21 23 80 LD HL,B_SW_COMM2
904 0768 CB AE RES 5,(HL)
905 076A C9 RET
906 076B 3E 00 C_RA_BD: LD A,0 ;
907 076D 32 21 80 LD (B_SW_DATA),A
908 0770 21 25 80 LD HL,B_SW_COMM4
909 0773 CB 86 RES 0,(HL)
910 0775 3E 00 C_BS_AL: LD A,0
911 0777 21 20 80 LD HL,B_SW_ADDR
912 077A 77 LD (HL),A
913 077B 23 INC HL
914 077C 77 LD (HL),A
915 077D C9 RET
916 ;TABLE COMMAND OPERATION
917 077E 6007 T_COMS2: DW C_WA_BD
918 0780 6B07 DW C_RA_BD
919 0782 7507 DW C_BS_AL
920 0784 B513 DW P_SP ;PUSH 'SP'-'AD_M'
921 0786 C213 DW P_RN ;PUSH 'm'-'AD_M'

```

922 0788 9E07 AC07 B007 DW WARD,WARDM,RAWRM,RDWR
078E B907

923 0790 BD07 C607 CA07 DW RAWR,RDMWR,WAIRD,RAIWR
0796 D407

924 0798 E112 DW C_RAM

925 079A B112 DW F_NSHOW

926 079C 5613 DW O_F_SONG

927 079E WARD:

928 079E 3E 21 LD A,00100001B ;RD

929 07A0 21 22 80 WARDS: LD HL,B_SW_COMM1

930 07A3 B6 OR (HL)

931 07A4 77 LD (HL),A

932 07A5 23 INC HL

933 07A6 3E 20 LD A,00100000B ;WRA

934 07A8 77 LD (HL),A

935 07A9 C3 DE 07 JP DE_SWP

936 07AC WARDM:

937 07AC 3E 29 LD A,00101001B ;RD,MREQ

938 07AE 18 F0 JR WARDS

939 07B0 RAWRM:

940 07B0 3E 01 LD A,1 ;RDA

941 07B2 32 25 80 LD (B_SW_COMM4),A

942 07B5 3E 19 LD A,00011001B ;WR,MREQ

943 07B7 18 E7 JR WARDS

944 07B9 RDWR:

945 07B9 3E 31 LD A,00110001B ;WR,RD

946 07BB 18 E3 JR WARDS

947 07BD RAWR:

948 07BD 3E 01 LD A,1 ;RDA

949 07BF 32 25 80 LD (B_SW_COMM4),A ;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

950 07C2 3E 11          LD    A,00010001B    ;WR
951 07C4 18 DA          JR     WARDS
952 07C6                RDMWR:
953 07C6 3E 39          LD    A,00111001B    ;RD,WR,MREQ
954 07C8 18 D6          JR     WARDS
955 07CA                WAIRD:
956 07CA 3E 20          LD    A,00100000B    ;WRA
957 07CC 32 23 80       WAIRDS: LD    (B_SW_COMM2),A
958 07CF 3E 25          LD    A,00100101B    ;RD,IORQ
959 07D1 C3 A0 07       JP     WARDS
960 07D4                RAIWR:
961 07D4 3E 01          LD    A,00000001B    ;RDA
962 07D6 32 25 80       LD    (B_SW_COMM4),A
963 07D9 3E 15          LD    A,00010101B    ;IORQ,WR
964 07DB C3 A0 07       JP     WARDS
965 07DE                DE_SWP:
966 07DE D9             EXX
967 07DF 0E 01          LD    C,01
968 07E1 C5             DE_SWPS: PUSH    BC
969 07E2 06 01          LD    B,1
970 07E4 CD EA 03       CALL    DELAY_CPU
971 07E7 21 22 80       LD    HL,B_SW_COMM1
972 07EA 16 02          LD    D,00000010B    ;LED RAM 'ON'
973 07EC 7E             LD    A,(HL)
974 07ED 72             LD    (HL),D
975 07EE 23             INC    HL
976 07EF 32 55 80       LD    (B_COMM1),A
977 07F2 7E             LD    A,(HL)
978 07F3 72             LD    (HL),D
979 07F4 23             INC    HL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

980 07F5 23          INC    HL
981 07F6 32 56 80    LD     (B_COMM2),A
982 07F9 7E          LD     A,(HL)
983 07FA 72          LD     (HL),D
984 07FB 32 58 80    LD     (B_COMM4),A
985 07FE 06 01       LD     B,1
986 0800 CD EA 03    CALL   DELAY_CPU
987 0803 21 55 80    LD     HL,B_COMM1
988 0806 7E          LD     A,(HL)
989 0807 32 22 80    LD     (B_SW_COMM1),A
990 080A 23          INC     HL
991 080B 7E          LD     A,(HL)
992 080C 32 23 80    LD     (B_SW_COMM2),A
993 080F 23          INC     HL
994 0810 23          INC     HL
995 0811 7E          LD     A,(HL)
996 0812 32 25 80    LD     (B_SW_COMM4),A
997 0815 C1          POP     BC
998 0816 0D          DEC     C
999 0817 C2 E1 07    JP      NZ,DE_SWPS
1000 081A D9          EXX
1001 081B C9          RET

1002                ;//////// %%%% //////////
1003                ;[ ---- CPU STEP 1 BYTE --- ]
1004                ; NO HAVE INSTIUTION END
1005 081C             RUN:
1006                ;CLEAR TIMMING CONTROL
1007 081C 3A 22 80    LD     A,(B_SW_COMM1)
1008 081F E6 42       AND     01000010B    ;ROM,RAM
1009 0821 32 22 80    LD     (B_SW_COMM1),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1010
1011           ;          CALL  CUR_D_A
1012 0824 21 2E 80      LD    HL,SYSFLX
1013 0827 CB E6         SET    4,(HL)          ;NEW STEP KEY PUSH
1014 0829 CB 6E         BIT    5,(HL)
1015 082B C2 73 08      JP     NZ,RINST        ;JP IF OPERAND
1016 082E CD 80 01      CALL   CH_RA_RO
1017 0831 3A 17 80      LD     A,(B_AB)        ;[(B_AB)]=>(B_DB)
1018 0834 CD 5D 12      CALL   SRCHAD
1019 0837 7E           LD     A,(HL)
1020 0838 32 16 80      LD     (B_DB),A
1021           ;CONTROL ACTIVE
1022 083B 3A 22 80      LD     A,(B_SW_COMM1) ;RD,MREQ
1023 083E E6 6A         AND     01101010B
1024 0840 F6 28         OR      00101000B
1025 0842 32 22 80      LD     (B_SW_COMM1),A
1026 0845 CD 9B 08      CALL   DE_CLK          ;'LED' CLK SWAP
1027 0848 3A 22 80      LD     A,(B_SW_COMM1) ;RD,MREQ
1028 084B E6 42         AND     01000010B
1029 084D 32 22 80      LD     (B_SW_COMM1),A
1030           ;END CONTROL ACTIVE
1031 0850 3A 16 80      LD     A,(B_DB)
1032 0853 32 1C 80      LD     (B_IR),A
1033 0856 3A 19 80      LD     A,(B_PC)          ;(B_PC)=(B_AB)
1034 0859 32 17 80      LD     (B_AB),A
1035 085C 3C           INC     A                ;(B_PC)=(B_PC+1)
1036 085D 32 19 80      LD     (B_PC),A
1037 0860 7E           LD     A,(HL)
1038 0861 CD BE 08      CALL   SRCHIN
1039 0864 78           LD     A,B              ; INSTRUCTION

```

1040	0865	FE 29	CP	TINSTM-TINSTS
1041	0867	DA 8F 08	JP	C,RINST1 ;JMP IF INSTI 1BYTE
1042	086A	21 2E 80	LD	HL,SYSFLX
1043	086D	CB EE	SET	5,(HL) ;COMMAND 2 BYTE
1044	086F	32 1F 80	LD	(B_IRS),A
1045	0872	C9	RET	
1046	0873	CB AE	RINST: RES	5,(HL)
1047	0875	CD 80 01	CALL	CH_RA_RO
1048	0878	3A 17 80	LD	A,(B_AB)
1049	087B	CD 5D 12	CALL	SRCHAD
1050	087E	7E	LD	A,(HL)
1051	087F	32 16 80	LD	(B_DB),A
1052	0882	3A 19 80	LD	A,(B_PC)
1053	0885	32 17 80	LD	(B_AB),A
1054	0888	3C	INC	A
1055	0889	32 19 80	LD	(B_PC),A
1056	088C	3A 1F 80	LD	A,(B_IRS) ;INSTIUTION
1057	088F	21 5F 0C	RINST1: LD	HL,CPUTAB
1058	0892	87	R_CPUS2: ADD	A,A
1059	0893	CD 5D 12	R_ALU: CALL	SRCHAD
1060	0896	5E	LD	E,(HL)
1061	0897	23	INC	HL
1062	0898	56	LD	D,(HL)
1063	0899	D5	PUSH	DE
1064	089A	C9	RET	
1065			;DELAY AND DISPLAY SWAP CLOCK LED.	
1066	089B		DE_CLK:	
1067	089B	C5	PUSH	BC
1068	089C	E5	PUSH	HL
1069	089D	F5	PUSH	AF

```

1070 089E 06 02          LD    B,2
1071 08A0 21 22 80      LD    HL,B_SW_COMM1
1072 08A3 CB C6          DE_CLK2: ; SET    0,(HL)
1073 08A5 3E 03          LD    A,3
1074 08A7 3D            DE_CLK1: DEC    A
1075 08A8 CD 1F 11      CALL  DISP_L
1076 08AB CD EB 0F      CALL  DISP_7S
1077 08AE 20 F7          JR     NZ,DE_CLK1
1078 08B0 CB 86          RES    0,(HL)
1079 08B2 CD 1F 11      CALL  DISP_L
1080 08B5 CD EB 0F      CALL  DISP_7S
1081 08B8 10 E9          DJNZ  DE_CLK2
1082 08BA F1            POP    AF
1083 08BB E1            POP    HL
1084 08BC C1            POP    BC
1085 08BD C9            RET
1086 08BE 06 3F          SRCHIN: LD    B,TINSTE-TINSTS
1087 08C0 21 5E 0C      LD    HL,TINSTE
1088 08C3 BE          SRCHLP: CP    (HL)
1089 08C4 C8            RET    Z
1090 08C5 2B            DEC    HL
1091 08C6 10 FB          DJNZ  SRCHLP
1092 08C8 C9            RET
1093                      ;B_AB+1->B_AD_M->B_SW_ADDR,
1094                      ;[B_AB+1]->B_DA_M->B_SW_DATA
1095                      ;FOR SHOW CURRENT
1096 08C9              CUR_D_A:
1097 08C9 3A 17 80      LD    A,(B_AB)
1098                      ;      INC    A
1099 08CC 00            NOP

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1100 08CD 32 15 80      CUR_D_A1: LD    (B_AD_M),A
1101 08D0 32 20 80      LD    (B_SW_ADDR),A
1102 08D3 CD 57 12      CALL  P_MEMS
1103 08D6 7E            LD    A,(HL)
1104 08D7 32 14 80      LD    (B_DA_M),A
1105 08DA 32 21 80      LD    (B_SW_DATA),A
1106 08DD CD 2A 10      CALL  DEC_7S
1107 08E0 C9            RET
1108 08E1 3A 17 80      CUR_D_AS:LD    A,(B_AB)
1109 08E4 18 E7          JR     CUR_D_A1
1110                      ;*** SAVE BUFEI ALL ***;
1111 08E6 21 00 80      SBA:    LD    HL,B_7S
1112 08E9 11 32 80      LD    DE,R_BUF
1113 08EC 01 20 00      LD    BC,B_SW_ADDR-B_7S
1114 08EF ED B0          LDIR
1115 08F1 C9            RET
1116                      ;*** MOV BUFEI COMEBACK ***;
1117 08F2 21 32 80      MBC:    LD    HL,R_BUF
1118 08F5 11 00 80      LD    DE,B_7S
1119 08F8 01 20 00      LD    BC,B_SW_ADDR-B_7S
1120 08FB ED B0          LDIR
1121 08FD C9            RET
1122                      ;
1123                      ;////////////////////////%;
1124                      ;----- SUB ROUTINE uP TIAINEI Z80s -----;
1125 08FE                      R00:
1126 08FE C9            RET                      ;NOP
1127 08FF                      R7D:
1128 08FF 3E 05          LD    A,0+5
1129 0901 CD 5A 07      CALL  STP2_COMD

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1130	0904	3A 1A 80	LD	A,(B_BR)	;LD A,B
1131	0907	32 1B 80	LD	(B_ACC),A	
1132	090A	C9	RET		
1133	090B		R7E:		
1134	090B	3E 06	LD	A,1+5	
1135	090D	CD 5A 07	CALL	STP2_COMD	
1136	0910	3A 1A 80	LD	A,(B_BR)	;LD A,(B)
1137	0913	21 00 86	LD	HL,RAM	
1138	0916	CD 5D 12	CALL	SRCHAD	
1139	0919	7E	LD	A,(HL)	
1140	091A	32 1B 80	LD	(B_ACC),A	
1141	091D	C9	RET		
1142	091E		R6F:		
1143	091E	3E 09	LD	A,4+5	
1144	0920	CD 5A 07	CALL	STP2_COMD	
1145	0923	3A 1B 80	LD	A,(B_ACC)	;LD B,A
1146	0926	32 1A 80	LD	(B_BR),A	
1147	0929	C9	RET		
1148	092A		R6E:		
1149	092A	3E 0A	LD	A,5+5	
1150	092C	CD 5A 07	CALL	STP2_COMD	
1151	092F	3A 1A 80	LD	A,(B_BR)	;LD B,(B)
1152	0932	21 00 86	LD	HL,RAM	
1153	0935	CD 5D 12	CALL	SRCHAD	
1154	0938	7E	LD	A,(HL)	
1155	0939	32 1A 80	LD	(B_BR),A	
1156	093C	C9	RET		
1157	093D		RF5:		
1158	093D	3E 07	LD	A,2+5	
1159	093F	CD 5A 07	CALL	STP2_COMD	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1160	0942	CD E5 0A	CALL R765	;PUSH A
1161	0945	3A 1B 80	LD A,(B_ACC)	
1162	0948	77	LD (HL),A	
1163	0949	C9	RET	
1164	094A		RE5:	
1165	094A	3E 0A	LD A,5+5	
1166	094C	CD 5A 07	CALL STP2_COMD	
1167	094F	CD E5 0A	CALL R765	;PUSH B
1168	0952	3A 1A 80	LD A,(B_BR)	
1169	0955	77	LD (HL),A	
1170	0956	C9	RET	
1171	0957		RED:	
1172	0957	3E 0A	LD A,5+5	
1173	0959	CD 5A 07	CALL STP2_COMD	
1174	095C	CD E5 0A	CALL R765	;PUSH F
1175	095F	3A 1E 80	LD A,(B_FLX)	
1176	0962	77	LD (HL),A	
1177	0963	C9	RET	
1178	0964		RF1:	
1179	0964	3E 06	LD A,1+5	
1180	0966	CD 5A 07	CALL STP2_COMD	
1181	0969	CD F3 0A	CALL R89A	;POP A
1182	096C	32 1B 80	LD (B_ACC),A	
1183	096F	7A	LD A,D	
1184	0970	32 18 80	LD (B_SP),A	
1185	0973	C9	RET	
1186	0974		RE1:	
1187	0974	3E 0A	LD A,5+5	
1188	0976	CD 5A 07	CALL STP2_COMD	
1189	0979	CD F3 0A	CALL R89A	;POP B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1190	097C	32 1A 80		LD	(B_BR),A	
1191	097F	7A		LD	A,D	
1192	0980	32 18 80		LD	(B_SP),A	
1193	0983	C9		RET		
1194	0984		RFD:			
1195	0984	3E 0A		LD	A,5+5	
1196	0986	CD 5A 07		CALL	STP2_COMD	
1197	0989	CD F3 0A		CALL	R89A	;POP F
1198	098C	32 1E 80		LD	(B_FLX),A	
1199	098F	7A		LD	A,D	
1200	0990	32 18 80		LD	(B_SP),A	
1201	0993	C9		RET		
1202	0994	21 F1 02	R3C:	LD	HL,ALUS	;RNC A
1203	0997	E5		PUSH	HL	
1204	0998	CD 00 0B	R3CR:	CALL	RAF	;R3C=SUB PROGIAM
1205	099B	3C		INC	A	
1206	099C	C9		RET		
1207	099D	3A 1B 80	R2C:	LD	A,(B_ACC)	;INC B
1208	09A0	5F		LD	E,A	;SAVE A Reg
1209	09A1	CD FF 08		CALL	R7D	;CALL LD A,B
1210	09A4	CD 94 09		CALL	R3C	;CALL FOR INC B
1211	09A7	3A 1B 80		LD	A,(B_ACC)	
1212	09AA	32 1A 80		LD	(B_BR),A	
1213	09AD	7B		LD	A,E	
1214	09AE	C3 F1 02		JP	ALUS	
1215	09B1	21 F1 02	R3D:	LD	HL,ALUS	;DEC A
1216	09B4	E5		PUSH	HL	
1217	09B5	CD 00 0B	R3DR:	CALL	RAF	;SUB PROGIAM
1218	09B8	3D		DEC	A	
1219	09B9	C9		RET		

1220	09BA	3A 1B 80	R2D:	LD	A,(B_ACC)	;DEC B
1221	09BD	5F		LD	E,A	
1222	09BE	CD FF 08		CALL	R7D	;LD A,B
1223	09C1	CD B5 09		CALL	R3DR	;DEC A FOR DEC B
1224	09C4	32 1A 80		LD	(B_BR),A	
1225	09C7	7B		LD	A,E	
1226	09C8	C3 F1 02		JP	ALUS	
1227	09CB	CD 00 0B	R87:	CALL	RAF	;ADD A,A
1228	09CE	87		ADD	A,A	;A+A=A
1229	09CF	21 23 80		LD	HL,B_SW_COMM2	
1230	09D2	36 F0		LD	(HL),11110000B	;SET COMD ALU
1231	09D4	C3 F1 02		JP	ALUS	
1232	09D7	3A 1A 80	R85:	LD	A,(B_BR)	;ADD A,B
1233	09DA	57		LD	D,A	
1234	09DB	CD 00 0B	R85R:	CALL	RAF	
1235	09DE	82		ADD	A,D	
1236	09DF	21 23 80		LD	HL,B_SW_COMM2	
1237	09E2	36 F0		LD	(HL),11110000B	;SET COMD ALU
1238	09E4	C3 F1 02		JP	ALUS	
1239	09E7	CD 0B 0B	R86:	CALL	D_HL	;ADD A,(B)
1240	09EA	C3 DB 09		JP	R85R	;PROCESSING
1241	09ED	CD 00 0B	R97:	CALL	RAF	;SUB A,A
1242	09F0	97		SUB	A	
1243	09F1	21 23 80		LD	HL,B_SW_COMM2	
1244	09F4	36 F4		LD	(HL),11110100B	;SET COMD ALU
1245	09F6	C3 F1 02		JP	ALUS	
1246	09F9	3A 1A 80	R95	LD	A,(B_BR)	;SUB A,B
1247	09FC	57		LD	D,A	
1248	09FD	CD 00 0B	R95R:	CALL	RAF	
1249	0A00	92		SUB	D	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1250	0A01	21 23 80		LD	HL,B_SW_COMM2
1251	0A04	36 F4		LD	(HL),11110100B ;SET COMD ALU
1252	0A06	C3 F1 02		JP	ALUS
1253	0A09	CD 0B 0B	R96:	CALL	D_HL ;SUB A,(B)
1254	0A0C	C3 FD 09		JP	R95R
1255	0A0F	CD 00 0B	RBF:	CALL	RAF ;CP A
1256	0A12	BF		CP	A
1257	0A13	21 23 80		LD	HL,B_SW_COMM2
1258	0A16	36 F4		LD	(HL),11110100B ;SET COMD ALU
1259	0A18	C3 F1 02		JP	ALUS
1260	0A1B	3A 1A 80	RBD:	LD	A,(B_BR) ;CP B
1261	0A1E	57		LD	D,A
1262	0A1F	CD 00 0B	RBDR:	CALL	RAF
1263	0A22	BA		CP	D
1264	0A23	21 23 80		LD	HL,B_SW_COMM2
1265	0A26	36 F4		LD	(HL),11110100B ;SET COMD ALU
1266	0A28	C3 F1 02		JP	ALUS
1267	0A2B	CD 0B 0B	RBE:	CALL	D_HL ;CP (B)
1268	0A2E	C3 1F 0A		JP	RBDR
1269	0A31	CD 00 0B	RA7:	CALL	RAF ;AND A,A
1270	0A34	A7		AND	A
1271	0A35	21 23 80		LD	HL,B_SW_COMM2
1272	0A38	36 E4		LD	(HL),111100100B ;SET COMD ALU
1273	0A3A	C3 F1 02		JP	ALUS
1274	0A3D	3A 1A 80	RA5:	LD	A,(B_BR) ;AND A,B
1275	0A40	57		LD	D,A
1276	0A41	CD 00 0B	RA5R:	CALL	RAF
1277	0A44	A2		AND	D
1278	0A45	21 23 80		LD	HL,B_SW_COMM2
1279	0A48	36 E4		LD	(HL),111100100B ;SET COMD ALU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1280	0A4A	C3 F1 02		JP	ALUS	
1281	0A4D	CD 0B 0B	RA6:	CALL	D_HL	;AND A,(B)
1282	0A50	C3 41 0A		JP	RA5R	
1283	0A53	CD 00 0B	RB7:	CALL	RAF	;OR A,A
1284	0A56	B7		OR	A	
1285	0A57	21 23 80		LD	HL,B_SW_COMM2	
1286	0A5A	36 E8		LD	(HL),11101000B	;SET COMD ALU
1287	0A5C	C3 F1 02		JP	ALUS	
1288	0A5F	3A 1A 80	RB5:	LD	A,(B_BR)	;OR A,B
1289	0A62	57		LD	D,A	
1290	0A63	CD 00 0B	RB5R:	CALL	RAF	
1291	0A66	B2		OR	D	
1292	0A67	21 23 80		LD	HL,B_SW_COMM2	
1293	0A6A	36 E8		LD	(HL),11101000B	;SET COMD ALU
1294	0A6C	C3 F1 02		JP	ALUS	
1295	0A6F	CD 0B 0B	RB6:	CALL	D_HL	;OR A,(B)
1296	0A72	C3 63 0A		JP	RB5R	
1297	0A75	CD 00 0B	XAF:	CALL	RAF	;XOR A
1298	0A78	AF		XOR	A	
1299	0A79	21 23 80		LD	HL,B_SW_COMM2	
1300	0A7C	36 EC		LD	(HL),11101100B	;SET COMD ALU
1301	0A7E	C3 F1 02		JP	ALUS	
1302	0A81	3A 1A 80	RAD:	LD	A,(B_BR)	;XOR A,(B)
1303	0A84	57		LD	D,A	
1304	0A85	CD 00 0B	RADR:	CALL	RAF	
1305	0A88	AA		XOR	D	
1306	0A89	21 23 80		LD	HL,B_SW_COMM2	
1307	0A8C	36 EC		LD	(HL),11101100B	;SET COMD ALU
1308	0A8E	C3 F1 02		JP	ALUS	
1309	0A91	CD 0B 0B	RAE:	CALL	D_HL	;XOR A,(B)

1310	0A94	C3 85 0A		JP	RADR	
1311	0A97	E5	R45:	PUSH	HL	;RETN
1312	0A98	21 2F 80		LD	HL,SYS_STP2	
1313	0A9B	CB AE		RES	5,(HL)	;IFF1='0'CAN 'NMI'
1314	0A9D	21 24 80		LD	HL,B_SW_COMM3	
1315	0AA0	CB 9E		RES	3,(HL)	;LED NMI 'OFF'
1316	0AA2	21 30 80		LD	HL,SYS_NMI	
1317	0AA5	CB C6		SET	0,(HL)	;SET REPEAT 'NMI'
1318	0AA7	E1		POP	HL	
1319	0AA8		RC9:			
1320	0AA8	3E 08		LD	A,8	
1321	0AAA	CD 5A 07		CALL	STP2_COMD	
1322	0AAD	CD F3 0A		CALL	R89A	;RET
1323	0AB0	32 19 80		LD	(B_PC),A	
1324	0AB3	3D		DEC	A	
1325	0AB4	32 17 80		LD	(B_AB),A	
1326	0AB7	7A		LD	A,D	
1327	0AB8	32 18 80		LD	(B_SP),A	
1328	0ABB	C9		RET		
1329	0ABC	E5	R4D:	PUSH	HL	;RETI 'INT'
1330	0ABD	21 2F 80		LD	HL,SYS_STP2	
1331	0AC0	CB AE		RES	5,(HL)	;IFF='0' CAN 'INT'
1332	0AC2	21 24 80		LD	HL,B_SW_COMM3	
1333	0AC5	CB 96		RES	2,(HL)	;LED INT 'OFF'
1334	0AC7	E1		POP	HL	
1335	0AC8	C3 A8 0A		JP	RC9	
1336	0ACB	76	R76:	HALT		;HALT
1337	0ACC		RFB:			
1338	0ACC	21 2F 80		LD	HL,SYS_STP2	;EI
1339	0ACF	CB EE		SET	5,(HL)	;IFF1='1'

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1340 0AD1 C9                                RET
1341 0AD2                                RF3:
1342 0AD2 21 2F 80                        LD    HL,SYS_STP2    ;DI
1343 0AD5 CB AE                            RES    5,(HL)        ;IFF1='0'
1344 0AD7 C9                                RET
1345 0AD8 CD 00 0B      R2F:              CALL   RAF        ;A=1'S A
1346 0ADB 2F                                CPL
1347 0ADC C3 F1 02                        JP     ALUS
1348 0ADF 21 2E 80      RDF:              LD     HL,SYSFLX
1349 0AE2 CB BE                            RES    7,(HL)        ;INSTIUTION END.
1350 0AE4 C9                                RET
1351                                     ;
1352                                     ;////////// SUB R765,R89A,RAF,D_HL //////////
1353                                     ;
1354 0AE5 3A 18 80      R765:              LD     A,(B_SP)      ;SEND SP>A
1355 0AE8 3D                                DEC     A            ;DEC SP
1356 0AE9 32 18 80      LD     (B_SP),A    ;SAVE SP
1357 0AEC 21 00 86      LD     HL,RAM      ;ADDRESS OF RAM
1358 0AEF CD 5D 12      CALL   SRCHAD      ;FIND
1359                                     ;CALL SRCHAD
1360                                     ;I/P HL,A
1361                                     ;O/P HL AT A
1362 0AF2 C9                                RET
1363 0AF3 21 00 86      R89A:              LD     HL,RAM      ;ADDRESS START
1364 0AF6 3A 18 80      LD     A,(B_SP)    ;SEND SP
1365 0AF9 57            LD     D,A          ;SADC SP
1366 0AFA 14            INC     D           ;SP=SP+1
1367 0AFB CD 5D 12      CALL   SRCHAD      ;FIND ADR DATA
1368 0AFE 7E            LD     A,(HL)      ;SEND DATA OF RAM
1369 0AFF C9                                RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1370 0B00          RAF:
1371 0B00 3A 1E 80          LD    A,(B_FLX)
1372 0B03 6F          LD    L,A          ;READ AF AT BUF
1373 0B04 3A 1B 80          LD    A,(B_ACC)
1374 0B07 67          LD    H,A
1375 0B08 E5          PUSH   HL          ;SAVE TO SP
1376 0B09 F1          POP    AF          ;COPY RELAY
1377 0B0A C9          RET
1378 0B0B 3A 1A 80      D_HL: LD    A,(B_BR)      ;
1379 0B0E 21 00 86      LD    HL,RAM          ;ADDR START
1380 0B11 CD 5D 12      CALL   SRCHAD          ;FIND ADDR(B)
1381 0B14 56          LD    D,(HL)          ;D=(B)
1382 0B15 C9          RET
1383          ;////////// END SUB SERVICE ////////////
1384          ;..... OF ONE BYTE .....
1385 0B16          R3EN:
1386 0B16 3E 05          LD    A,0+5
1387 0B18 CD 5A 07      CALL   STP2_COMD
1388 0B1B 3A 16 80      LD    A,(B_DB)          ;**LD A,N
1389 0B1E 32 1B 80      LD    (B_ACC),A          ;AReg=N
1390 0B21 C9          RET
1391 0B22          R3AN:
1392 0B22 3E 06          LD    A,1+5
1393 0B24 CD 5A 07      CALL   STP2_COMD
1394 0B27 3A 16 80      LD    A,(B_DB)          ;**LD A,(N)
1395 0B2A 21 00 86      LD    HL,RAM
1396 0B2D CD 5D 12      CALL   SRCHAD
1397 0B30 7E          LD    A,(HL)          ;A=(N)
1398 0B31 32 1B 80      LD    (B_ACC),A          ;AReg=(N)
1399 0B34 C9          RET

```

1400	0B35		R32N:	
1401	0B35	3E 07		LD A,2+5
1402	0B37	CD 5A 07		CALL STP2_COMD
1403	0B3A	3A 16 80		LD A,(B_DB) ;**LD (N),A
1404	0B3D	21 00 86		LD HL,RAM
1405	0B40	CD 5D 12		CALL SRCHAD
1406	0B43	3A 1B 80		LD A,(B_ACC) ;A=AReg
1407	0B46	77		LD (HL),A ;(N)=AReg
1408	0B47	C9		RET
1409	0B48		R2EN:	
1410	0B48	3E 08		LD A,3+5
1411	0B4A	CD 5A 07		CALL STP2_COMD
1412	0B4D	3A 16 80		LD A,(B_DB) ;**LD B,n
1413	0B50	32 1A 80		LD (B_BR),A ;BReg=n
1414	0B53	C9		RET
1415	0B54		R31N:	
1416	0B54	3E 08		LD A,3+5
1417	0B56	CD 5A 07		CALL STP2_COMD
1418	0B59	3A 16 80		LD A,(B_DB) ;**LD SP,n
1419	0B5C	32 18 80		LD (B_SP),A ;SP=n
1420	0B5F	C9		RET
1421	0B60		RDBN:	
1422	0B60	3E 0B		LD A,6+5
1423	0B62	CD 5A 07		CALL STP2_COMD
1424	0B65	3A 16 80		LD A,(B_DB) ;**IN A,(n)
1425	0B68	4F		LD C,A ;C=n
1426	0B69	ED 78		IN A,(C)
1427	0B6B	32 1B 80		LD (B_ACC),A
1428	0B6E	C9		RET
1429	0B6F		RD3N:	

```

1430 0B6F 3E 0C          LD    A,7+5
1431 0B71 CD 5A 07      CALL  STP2_COMD
1432 0B74 3A 16 80      LD    A,(B_DB)      ;**OUT (n),A
1433 0B77 4F           LD    C,A
1434 0B78 3A 1B 80      LD    A,(B_ACC)
1435 0B7B ED 79        OUT   (C),A
1436 0B7D C9           RET
1437 0B7E 3A 16 80      RC6N: LD    A,(B_DB)      ;**ADD A,n
1438 0B81 57           LD    D,A
1439 0B82 C3 DB 09      JP    R85R      ;GO TO ADD
1440 0B85 3A 16 80      RD6N: LD    A,(B_DB)      ;**SUB A,n
1441 0B88 57           LD    D,A      ;D=n
1442 0B89 C3 FD 09      JP    R95R      ;GO TO SUB
1443 0B8C 3A 16 80      RFEN: LD    A,(B_DB)      ;**CP A,n
1444 0B8F 57           LD    D,A      ;D=n
1445 0B90 C3 1F 0A      JP    RBDR      ;GO COMPARE
1446 0B93 3A 16 80      RE6N: LD    A,(B_DB)      ;**AND A,n
1447 0B96 57           LD    D,A
1448 0B97 C3 41 0A      JP    RA5R      ;GOTO AND
1449 0B9A 3A 16 80      RF6N: LD    A,(B_DB)      ;**OR A,n
1450 0B9D 57           LD    D,A
1451 0B9E C3 63 0A      JP    RB5R
1452 0BA1 3A 16 80      REEN: LD    A,(B_DB)      ;**XOR A,n
1453 0BA4 57           LD    D,A      ;D=n
1454 0BA5 C3 85 0A      JP    RADR      ;GOTO XOR
1455                    ;
1456                    ;////////// JUMP //////////
1457 0BA8 3A 16 80      RC3N: LD    A,(B_DB)      ;**JP n
1458 0BAB 32 19 80      LD    (B_PC),A      ;PC=n
1459 0BAE E5           PUSH  HL

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1460 0BAF 21 22 80      LD    HL,B_SW_COMM1
1461 0BB2 CB B6         RES    6,(HL)      ;ROM ='0'
1462 0BB4 CB CE         SET    1,(HL)
1463 0BB6 E1           POP    HL
1464 0BB7 CD BB 0B      CALL   JMP_R
1465 0BBA C9           RET
1466
1467 0BBB                JMP_R:
1468 0BBB CD FF 03      CALL   DELAY_CPUS
1469 0BBE 3A 19 80      LD     A,(B_PC)      ;PC=>AB
1470 0BC1 32 17 80      LD     (B_AB),A
1471 0BC4 3C           INC     A
1472 0BC5 32 19 80      LD     (B_PC),A      ;PC=PC+1
1473 0BC8 CD FF 03      CALL   DELAY_CPUS
1474 0BCB C9           RET
1475
1476 0BCC 3A 1E 80      RCAN:   LD     A,(B_FLX)      ;**JP Z,n
1477 0BCF CB 77         BIT     6,A      ;Z FLAX
1478 0BD1 C2 A8 0B      JP     NZ,RC3N      ;JMP IF Z='1'
1479 0BD4 C9           RET
1480 0BD5 3A 1E 80      RC2N:   LD     A,(B_FLX)      ;**JP NZ,n
1481 0BD8 CB 77         BIT     6,A      ;Z
1482 0BDA CA A8 0B      JP     Z,RC3N      ;JMP IF Z='0'
1483 0BDD C9           RET
1484 0BDE 3A 1E 80      RDAN:   LD     A,(B_FLX)      ;**JP C,n
1485 0BE1 CB 47         BIT     0,A      ;C
1486 0BE3 C2 A8 0B      JP     NZ,RC3N      ;JMP IF C='1'
1487 0BE6 C9           RET
1488 0BE7 3A 1E 80      RD2N:   LD     A,(B_FLX)      ;**JP NC,n
1489 0BEA CB 47         BIT     0,A      ;C

```

```

1490 0BEC CA A8 0B      JP    Z,RC3N      ;JMP IF C='0'
1491 0BEF C9              RET
1492                      ;////////// CALL //////////
1493 0BF0 CD E5 0A      RCDN:   CALL    R765      ;SP=SP-1
1494 0BF3 3A 19 80      LD     A,(B_PC)
1495 0BF6 3D              DEC     A
1496 0BF7 77              LD     (HL),A      ;PC TO (SP)
1497 0BF8 C3 A8 0B      JP     RC3N      ;PC=n
1498 0BFB 3A 1E 80      RCCN:   LD     A,(B_FLX)    ;**CALL Z,n
1499 0BFE CB 77          BIT     6,A      ;Z
1500 0C00 CA F0 0B      JP     Z,RCDN      ;JMP IF Z=1
1501 0C03 C9              RET
1502 0C04 3A 1E 80      RC4N:   LD     A,(B_FLX)    ;**CALL NZ,n
1503 0C07 CB 77          BIT     6,A      ;Z
1504 0C09 CA F0 0B      JP     Z,RCDN      ;JMP IF Z=0
1505 0C0C C9              RET
1506 0C0D 3A 1E 80      RDCN:   LD     A,(B_FLX)    ;**CALL C,n
1507 0C10 CB 47          BIT     0,A      ;C
1508 0C12 C2 F0 0B      JP     NZ,RCDN      ;JMP IF C=1
1509 0C15 C9              RET
1510 0C16 3A 1E 80      RD4N:   LD     A,(B_FLX)    ;**CALL NC.N
1511 0C19 CB 47          BIT     0,A      ;C
1512 0C1B CA F0 0B      JP     Z,RCDN      ;JMP IF C=0
1513 0C1E C9              RET
1514                      ;
1515                      ;////////// TABLE uP TRAINER Z80s //////////
1516                      ;
1517                      ;..... TABLE INSTRUCTION OF TRAINER.....;
1518 0C1F 00 7D 7E 6F 6E TINSTS: DB 0,7DH,7EH,6FH,6EH      ;0-4
1519 0C24 F5 E5 ED F1      DB 0F5H,0E5H,0EDH,0F1H      ;5-8

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1550	0C7B	BA09 CB09	DW	R2D,R87	;E,F
1551	0C7F	D709 E709	DW	R85,R86	;10,11
1552	0C83	ED09 F909	DW	R97,R95	;12,13
1553	0C87	090A 0F0A	DW	R96,RBF	;14,15
1554	0C8B	1B0A 2B0A	DW	RBD,RBE	;16,17
1555	0C8F	310A 3D0A	DW	RA7,RA5	;18,19
1556	0C93	4D0A 530A	DW	RA6,RB7	;1A,1B
1557	0C97	5F0A 6F0A	DW	RB5,RB6	;1C,1D
1558	0C9B	750A 810A	DW	XAF,RAD	;1E,1F
1559	0C9F	910A A80A	DW	RAE,RC9	;20,21
1560	0CA3	CB0A CC0A	DW	R76,RFB	;22,23
1561	0CA7	D20A D80A	DW	RF3,R2F	;24,25
1562	0CAB	DF0A BC0A	DW	RDF,R4D	;26S1,S6S2
1563	0CAF	970A 160B	DW	R45,R3EN	;26,27
1564	0CB3	220B 350B	DW	R3AN,R32N	;28,29
1565	0CB7	480B 540B	DW	R2EN,R31N	;2A,2B
1566	0CBB	600B 6F0B	DW	RDBN,RD3N	;2C,2D
1567	0CBF	7E0B 850B	DW	RC6N,RD6N	;2E,2F
1568	0CC3	8C0B 930B	DW	RFEN,RE6N	;30,31
1569	0CC7	9A0B A10B	DW	RF6N,REEN	;32,33
1570	0CCB	A80B CC0B	DW	RC3N,RCAN	;34,35
1571	0CCF	D50B DE0B	DW	RC2N,RDAN	;36,37
1572	0CD3	E70B F00B	DW	RD2N,RCDN	;38,39
1573	0CD7	FB0B 040C	DW	RCCN,RC4N	;3A,3B
1574	0CDB	0D0C 160C	DW	RDCN,RD4N	;3C,3D

1 5 7 5

;=====;

1576 ; BUS I/O UNIT

1577 ;TIMMING PROCESS -READ MEMORY DIAGRAM

1578 ; -WRITE MEMORY DIAGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1579          ;          -READ PORT DIAGRAM
1580          ;          -WRITE PORT DIAGRAM
1581          ; WAIT STATE CONTROL ALL DIAGRAM! BY USER
1582          ;BUFFER COMMAND 'B_COMM' USE FOR SET BY
USER
1583          ;
1584          ;{main bus i/o}
1585 0CDF          M_BUS_IO:
1586 0CDF C5          PUSH BC
1587 0CE0 21 23 80      LD HL,B_SW_COMM2
1588 0CE3 CB C6          SET 0,(HL) ;SET USE BUS_IO
1589 0CE5 2B          DEC HL
1590 0CE6 CB CE          SET 1,(HL) ;SET RAM LED
1591 0CE8 CD 30 14      CALL SONG_BUS
1592 0CEB 06 E1          LD B,11100001B ;ENABLE
ADDRESS,DATA MEMORY
1593 0CED CD 6E 10      CALL MODE_7S
1594 0CF0 C1          POP BC
1595 0CF1 CD CC 0F      CALL REP_SW
1596 0CF4 CD 31 0D      LOOP: CALL S_BUS_IO
1597          ;COLUMN 2
1598 0CF7 3E 02          LD A,02
1599 0CF9 4F          LD C,A
1600 0CFA D3 82          OUT (P_SEL_SW),A
1601 0CFC DB 80          IN A,(P_SW)
1602 0CFE 47          LD B,A
1603 0CFF CB 47          BIT 0,A
1604 0D01 CC 48 11      CALL Z,KEY_MAT ;MATRIX KEY
1605          ;COLUMN 3
1606 0D04 3E 03          LD A,03

```

```

1607 0D06 4F          LD    C,A
1608 0D07 D3 82       OUT   (P_SEL_SW),A
1609 0D09 DB 80       IN    A,(P_SW)
1610 0D0B 47          LD    B,A
1611 0D0C CB 47       BIT    0,A          ;PRESS USE BUS = EXIT
1612 0D0E 28 0D       JR     Z,E_BUS
1613                  ;COLUMN 4
1614 0D10 3E 04       LD    A,4
1615 0D12 D3 82       OUT   (P_SEL_SW),A
1616 0D14 DB 80       IN    A,(P_SW)
1617 0D16 CB 47       BIT    0,A
1618 0D18 CC 7D 0D     CALL   Z,DIAGRAM      ;RUN DIAGRAM
1619 0D1B 18 D7       JR     LOOP
1620 0D1D             E_BUS:
1621 0D1D C5           PUSH   BC
1622 0D1E CD C3 0D     CALL   CLR_BUS
1623 0D21 21 23 80     LD     HL,B_SW_COMM2
1624 0D24 CB 86       RES    0,(HL)
1625 0D26 CD 30 14     CALL   SONG_BUS
1626 0D29 C1          POP    BC
1627 0D2A CD CC 0F     CALL   REP_SW
1628 0D2D CD 2A 10     CALL   DEC_7S
1629 0D30 C9          RET
1630 0D31             S_BUS_IO:
1631 0D31 21 20 80     LD     HL,B_SW_ADDR
1632 0D34 11 79 0D     LD     DE,T_MODE_SW
1633 0D37 0E 00       LD     C,00
1634 0D39             START1:
1635 0D39 CD 1F 11     CALL   DISP_L
1636 0D3C CD EB 0F     CALL   DISP_7S

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1637 0D3F 1A          LD    A,(DE)
1638 0D40 32 2B 80    LD    (MODE_SW),A
1639 0D43 79          LD    A,C
1640 0D44 CD 95 0F    CALL SWITCH_P
1641 0D47 23          INC    HL
1642                  ;CHANG ROM<->RAM LED AND BUFFER
1643 0D48 E5          PUSH   HL
1644 0D49 21 22 80    LD    HL,B_SW_COMM1
1645 0D4C CB 76        BIT    6,(HL)
1646 0D4E 20 04        JR     NZ,CHANG_A
1647 0D50 CB CE        SET    1,(HL)
1648 0D52 18 02        JR     E_CHANG
1649 0D54 CB 8E        CHANG_A: RES    1,(HL)
1650 0D56 E1          E_CHANG: POP    HL
1651 0D57 13          INC    DE
1652 0D58 0C          INC    C
1653 0D59 3E 03        LD    A,03
1654 0D5B B9          CP     C
1655 0D5C 20 DB        JR     NZ,START1
1656                  ;*****
1657                  ;PUSH DATA TO B_SW_DATA->B_DB
1658                  ; B_SW_ADDR->B_AB
1659                  ;*****
1660 0D5E 21 20 80    LD    HL,B_SW_ADDR
1661 0D61 7E          LD    A,(HL)
1662 0D62 32 17 80    LD    (B_AB),A
1663 0D65 23          INC    HL          ;B_SW_DATA
1664 0D66 7E          LD    A,(HL)
1665 0D67 32 16 80    LD    (B_DB),A
1666 0D6A CD 2A 10    CALL DEC_7S

```

```

1667 0D6D 06 E1          LD    B,11100001B
1668 0D6F CD 6E 10       CALL  MODE_7S
1669 0D72 CD 1F 11       CALL  DISP_L
1670 0D75 CD EB 0F       CALL  DISP_7S
1671                      ;*****
1672 0D78 C9             RET
1673 0D79 FF FF FE       T_MODE_SW: DB    0FFH,0FFH,1111110B
1674 0D7C 01             DB    1H
1675                      ;comm2,comm3,POIT
1676                      ;Finding a diagram flom (b_sw_comm)
1677                      ;when found otheI,will opeIateds and Ietulen to bus_i/o
1678 0D7D                DIAGRAM:
1679 0D7D 21 45 14       LD    HL,TAB_DIA1
1680 0D80 3A 22 80       LD    A,(B_SW_COMM1)
1681 0D83 E6 7C          AND    A,01111100B
1682 0D85 47             LD    B,A
1683 0D86 0E 00          LD    C,0 ; POINTER
1684 0D88 7E             NO_ERROR: LD    A,(HL)
1685 0D89 B8             CP    B
1686 0D8A 28 11          JR    Z,EXECUIT_DIA ; RUN DIAGRAM !
1687 0D8C 0C             INC    C
1688 0D8D 23             INC    HL
1689 0D8E 79             LD    A,C
1690 0D8F FE 07          CP    TAB_DIA2-TAB_DIA1
1691 0D91 20 F5          JR    NZ,NO_ERROR
1692                      ;.....
1693                      ; SUB ERROR SELECT PROGRAM BY YOU!
1694                      ;.....
1695 0D93                ERROR:
1696 0D93 3E 03          LD    A,SONG_ERR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1697 0D95 CD C0 10          CALL SONG
1698 0D98 3E 01             LD A,I
1699 0D9A D3 40             OUT (40H),A
1700 0D9C C9                RET

1701                        ;.....
1702                        ; FIND DIAGRAM ADDRESS AND JUMP TO THAT
1703                        ;.....
1704 0D9D EXECUIT_DIA:
1705 0D9D 06 E7              LD B,11100111B
1706 0D9F CD 6E 10          CALL MODE_7S ;NOT SHOW 7S'
1707 0DA2 CD AE 0D          CALL SAVE_BS
1708 0DA5 79                LD A,C
1709 0DA6 21 4D 14          LD HL,TAB_EXE_DIA
1710 0DA9 CD DD 10          CALL FIND_TA
1711 0DAC D5                PUSH DE
1712 0DAD C9                RET
1713 0DAE SAVE_BS:
1714 0DAE E5                PUSH HL
1715 0DAF C5                PUSH BC
1716 0DB0 11 27 80          LD DE,B2_SW_AD
1717 0DB3 21 20 80          LD HL,B_SW_ADDR
1718 0DB6 7E                LD A,(HL)
1719 0DB7 12                LD (DE),A
1720 0DB8 AF                XOR A
1721 0DB9 77                LD (HL),A
1722 0DBA 23                INC HL
1723 0DBB 13                INC DE
1724 0DBC 7E                LD A,(HL)
1725 0DBD 12                LD (DE),A
1726 0DBE AF                XOR A

```

```

1727 0DBF 77          LD    (HL),A
1728 0DC0 C1          POP    BC
1729 0DC1 E1          POP    HL
1730 0DC2 C9          RET

1731                  ;
1732                  ;..... SUB CLEAR OF BUS_IO .....;
1733                  ;

1734 0DC3 AF          CLR_BUS: XOR    A
1735 0DC4 21 20 80     LD    HL,B_SW_ADDR
1736 0DC7 77          LD    (HL),A
1737 0DC8 23          INC    HL          ;B_SW_DATA
1738 0DC9 77          LD    (HL),A
1739 0DCA 23          INC    HL
1740 0DCB 3E 02       LD    A,00000010B ;SET RAM LED
1741 0DCD 77          LD    (HL),A ;B_SW_COMM1
1742 0DCE C9          RET

1743                  ;.....
1744                  ;SUB OPERATING BY DIAGRAM MICROPROCESSOR
1745                  ;.....
1746                  ;*****
1747 0DCF             READ_ROM:
1748                  ;*****
1749                  ;*****
1750 0DCF             READ_RAM:
1751                  ;*****
1752                  ;INITIAL;
1753                  ;CLEAR LED COMMAND

1754 0DCF CD D1 0E     CALL    INIT
1755 0DD2 3E 05       LD    A,SONG_RD
1756 0DD4 CD C0 10     CALL    SONG

```

```

1757 0DD7 CD E7 0E          CALL CLK1_UD
1758 0DDA 16 28             LD D,00101000B
1759 0DDC CD 3D 0F          CALL CLK_U_D1
1760 0DDF                  CHEC_W_RD:
1761 0DDF CB 7E             BIT 7,(HL) ;BIT WAIT
1762 0DE1 28 06             JR Z,N_WAI_RD
1763 0DE3 11 80 29          LD DE,0010100110000000B
1764 0DE6 CD 7A 0F          CALL WAIT
1765 0DE9                  N_WAI_RD:
1766 0DE9 16 28             LD D,00101000B
1767 0DEB CD 3D 0F          CALL CLK_U_D1
1768                      ;PROCESS OF READ ROM WITH SHOW ALL
1769 0DEE 11 00 00          LD DE,0
1770 0DF1 CB 76             BIT 6,(HL) ;CHECK ROM?
1771 0DF3 20 03             JR NZ,RD_ROM
1772 0DF5 11 00 86          LD DE,RAM ;RAM MUST E=00 !
1773 0DF8                  RD_ROM:
1774 0DF8 3A 15 80          LD A,(B_AD_M)
1775 0DFB 5F                LD E,A
1776 0DFC 1A                LD A,(DE) ;RD MEMDB,DA_M
1777 0DFD 32 14 80          LD (B_DA_M),A
1778 0E00 32 21 80          LD (B_SW_DATA),A
1779 0E03 21 14 80          LD HL,B_DA_M
1780 0E06 11 21 80          LD DE,B_SW_DATA
1781 0E09 CD 2E 0F          CALL ROTAT_L
1782 0E0C 32 16 80          LD (B_DB),A
1783 0E0F                  CLK41S:
1784 0E0F 06 E1             LD B,11100001B
1785 0E11 CD 2A 10          CLK42S: CALL DEC_7S
1786 0E14 CD 6E 10          CALL MODE_7S

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1787 0E17                                CLK4S:
1788 0E17 06 50                          LD    B,50H
1789 0E19 CD EA 03                        CALL   DELAY_CPU
1790 0E1C CD D1 0E                        CALL   INIT
1791 0E1F C9                              RET
1792                                     ; END OF READ MEMORY
1793                                     ;.....
1794
1795 0E20                                WRITE_RAM:
1796 0E20 CD D1 0E                        CALL   INIT
1797 0E23 3E 06                          LD     A,SONG_WR
1798 0E25 CD C0 10                        CALL   SONG
1799 0E28 CD E7 0E                        CALL   CLK1_UD
1800 0E2B 21 28 80                      LD     HL,B2_SW_DA
1801 0E2E 11 21 80                      LD     DE,B_SW_DATA
1802 0E31 CD 2E 0F                        CALL   ROTAT_L
1803 0E34 16 08                          LD     D,00001000B ;MREQ='1'
1804 0E36 CD 3D 0F                        CALL   CLK_U_D1
1805 0E39 21 22 80                      LD     HL,B_SW_COMM1
1806 0E3C                                CHEC_W_WR:
1807 0E3C CB 7E                          BIT    7,(HL)
1808 0E3E 28 06                          JR     Z,N_WAI_WR
1809 0E40 11 80 19                      LD     DE,0001100110000000B
1810 0E43 CD 7A 0F                        CALL   WAIT
1811 0E46                                N_WAI_WR:
1812 0E46 16 18                          LD     D,00011000B
1813 0E48 CD 3D 0F                        CALL   CLK_U_D1
1814 0E4B 21 00 86                      LD     HL,RAM ; DATA TO MEM
1815 0E4E 3A 16 80                      LD     A,(B_DB)
1816 0E51 32 14 80                      LD     (B_DA_M),A

```

```

1817 0E54 47 LD B,A
1818 0E55 3A 20 80 LD A,(B_SW_ADDR)
1819 0E58 6F LD L,A
1820 0E59 70 LD (HL),B ;B=DATA=>(HL=MEM)
1821 0E5A C3 0F 0E JP CLK41S
1822 ,*****
1823 0E5D WRITE_PORT:
1824 ,*****
1825 0E5D CD D1 0E CALL INIT
1826 0E60 3E 06 LD A,SONG_WR
1827 0E62 CD C0 10 CALL SONG
1828 0E65 CD E7 0E CALL CLK1_UD
1829 0E68 21 28 80 LD HL,B2_SW_DA
1830 0E6B 11 21 80 LD DE,B_SW_DATA
1831 0E6E CD 2E 0F CALL ROTAT_L
1832 0E71 16 00 LD D,0
1833 0E73 21 2F 80 LD HL,SYS_STP2
1834 0E76 CB D6 SET 2,(HL) ;WR,IORQ='1'
1835 0E78 CD 3D 0F CALL CLK_U_D1
1836 0E7B CHEC_W_WRO:
1837 0E7B 11 80 14 LD DE,000101001000000B
1838 0E7E CD 87 0F CALL TW_WAIT
1839 0E81 NO_WAI_WRO:
1840 0E81 16 14 LD D,00010100B
1841 0E83 21 22 80 LD HL,B_SW_COMM1
1842 0E86 CD 3D 0F CALL CLK_U_D1
1843 0E89 21 20 80 LD HL,B_SW_ADDR
1844 0E8C 4E LD C,(HL)
1845 0E8D 23 INC HL ;B_SW_DATA
1846 0E8E 7E LD A,(HL)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1847 0E8F ED 79                                OUT    (C),A
1848 0E91 06 E7                                END_RP: LD    B,11100111B ;SHOW AB,DB †
1849 0E93 C3 11 0E                                JP     CLK42S
1850
1851                                ;*****
1852 0E96                                READ_PORT:
1853                                ;*****
1854 0E96 CD D1 0E                                CALL   INIT
1855 0E99 3E 06                                LD     A,SONG_WR
1856 0E9B CD C0 10                                CALL   SONG
1857 0E9E CD E7 0E                                CALL   CLK1_UD
1858 0EA1 21 2F 80                                LD     HL,SYS_STP2
1859 0EA4 CB DE                                SET    3,(HL) ;RD,IORQ='1'
1860 0EA6 16 00                                LD     D,0.
1861 0EA8 CD 3D 0F                                CALL   CLK_U_D1
1862 0EAB 11 80 24                                LD     DE,0010010010000000B
1863 0EAE CD 87 0F                                CALL   TW_WAIT
1864 0EB1 16 24                                LD     D,00100100B
1865 0EB3 21 22 80                                LD     HL,B_SW_COMM1
1866 0EB6 CD 3D 0F                                CALL   CLK_U_D1
1867 0EB9 21 20 80                                LD     HL,B_SW_ADDR
1868 0EBC 4E                                LD     C,(HL)
1869 0EBD 21 28 80                                LD     HL,B2_SW_DA
1870 0EC0 11 21 80                                LD     DE,B_SW_DATA
1871 0EC3 CD 2E 0F                                CALL   ROTAT_L
1872 0EC6 ED 78                                IN     A,(C)
1873 0EC8 32 21 80                                LD     (B_SW_DATA),A
1874 0ECB 32 16 80                                LD     (B_DB),A
1875 0ECE C3 91 0E                                JP     END_RP
1876

```

```

1877                                ;INITIAL OF ALL DIAGRAM
1878 0ED1                        INIT:
1879 0ED1 21 22 80                LD    HL,B_SW_COMM1
1880 0ED4 3E C2                    LD    A,11000010B ;
1881 0ED6 A6                        AND    (HL)
1882 0ED7 77                        LD    (HL),A
1883 0ED8 C9                        RET
1884
1885
1886 0ED9                        BIT_CLK:
1887                                ; USE 'A' COMD SWITCH. AND DISPLAY COMD
1888                                ;AND CHECK BIT -WAIT- IN 'C'AND -CLOCK-
1889                                ; A,C,HL
1890                                ;HL=B_SW_COMM
1891 0ED9 21 22 80                LD    HL,B_SW_COMM1
1892 0EDC 32 2B 80                LD    (MODE_SW),A
1893 0EDF 0E 02                    LD    C,02 ; SELECT COMMAND
1894 0EE1 CD 95 0F                CALL SWITCH_P
1895 0EE4 CB 46                    BIT    0,(HL)
1896 0EE6 C9                        RET
1897                                ;
1898                                ;USE CLK TWO PUSH POSI AGE AND NEG AGE AND
1899                                ;DISPLAY ADDRESS
1900                                ;USE HL,A,BC,DE
1901
1902 0EE7                        CLK1_UD:
1903 0EE7 CD EB 0F                CALL DISP_7S
1904 0EEA CD 1F 11                CALL DISP_L
1905 0EED 3E 81                    LD    A,10000001B ;
1906 0EEF CD D9 0E                CALL BIT_CLK

```

```

1907 0EF2 28 F3          JR    Z,CLK1_UD
1908 0EF4 21 2F 80       LD     HL,SYS_STP2
1909 0EF7 CB 4E          BIT    1,(HL)      ;IF WRITE_MEM,PORT
1910 0EF9 28 0E          JR     Z,N_B_D7      ;(B_DB)=>(D0-D7)
1911 0EFB 21 28 80       LD     HL,B2_SW_DA
1912 0EFE 11 21 80       LD     DE,B_SW_DATA
1913 0F01 CD 2E 0F       CALL  ROTAT_L
1914 0F04 21 2F 80       LD     HL,SYS_STP2
1915 0F07 CB 8E          RES    1,(HL)      ;CLR (D0-D7)
1916 0F09                N_B_D7:
1917 0F09 21 27 80       LD     HL,B2_SW_AD
1918 0F0C D5             PUSH  DE
1919 0F0D 11 20 80       LD     DE,B_SW_ADDR
1920 0F10 CD 2E 0F       CALL  ROTAT_L
1921 0F13 D1             POP   DE
1922 0F14 7E             LD     A,(HL)
1923 0F15 32 15 80       LD     (B_AD_M),A
1924 0F18 CD 2A 10       CALL  DEC_7S
1925 0F1B 06 E3          LD     B,11100011B ; SHOW ADDRESS
1926 0F1D CD 6E 10       CALL  MODE_7S
1927 0F20 CD EB 0F       LK1D: CALL DISP_7S
1928 0F23 CD 1F 11       CALL  DISP_L
1929 0F26 3E 81          LD     A,10000001B ; CAN USE CLK,WAIT
1930 0F28 CD D9 0E       CALL  BIT_CLK
1931 0F2B C8             RET     Z
1932 0F2C 18 F2          JR     LK1D
1933                    ;
1934                    ;ROTATE B_SW_??? OF BUS
1935                    ;FOR SHOW BUS BUSY!
1936                    ; HL=SORCE BUFFER

```

```

1937                                ; DE=TRANGET BUFFER
1938                                ;USE BC,A
1939 0F2E                                ROTAT_L:
1940 0F2E 3E 50                        LD A,050H
1941 0F30 12                        ROT1: LD (DE),A
1942 0F31 CD 1F 11                    CALL DISP_L
1943 0F34 CD EB 0F                    CALL DISP_7S
1944 0F37 3D                        DEC A
1945 0F38 20 F6                        JR NZ,ROT1
1946 0F3A 7E                        LD A,(HL)
1947 0F3B 12                        LD (DE),A
1948 0F3C C9                        RET
1949                                ;PUSH CLOCK TWO PUSH AND DISPLAY
1950                                ;7S,SONG,L_COMM
1951                                ;USE 'D' FOR SWAP
1952                                ;D,HL,E
1953 0F3D                        CLK_U_D1:
1954 0F3D 1E 81                        LD E,10000001B ;USE CLK WAIT
1955 0F3F                        CLK2U:
1956 0F3F CD 62 0F                    CALL SWAP_LCOMM1 ;
1957 0F42 28 FB                        JR Z,CLK2U
1958 0F44 21 2F 80                    LD HL,SYS_STP2
1959 0F47 CB 56                        BIT 2,(HL)
1960 0F49 28 06                        JR Z,CLK2DS
1961 0F4B CB 96                        RES 2,(HL)
1962 0F4D 16 14                        LD D,00010100B ;WR,IORQ
1963 0F4F 18 08                        JR CLK2D
1964 0F51 CB 5E                        CLK2DS: BIT 3,(HL)
1965 0F53 28 04                        JR Z,CLK2D
1966 0F55 CB 9E                        RES 3,(HL)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1967 0F57 16 24          LD    D,00100100B ;RD,IORQ
1968 0F59                CLK2D:
1969 0F59 21 22 80        LD    HL,B_SW_COMM1
1970 0F5C CD 62 0F        CALL  SWAP_LCOMM1 ;
1971 0F5F 20 F8          JR    NZ,CLK2D
1972 0F61 C9             RET
1973                      ;
1974                      ; DISP SWAP BY 'D' AND 'E' FOR ENABLE PRESS SW
1975                      ; USE HL=B_SW_COMM
1976 0F62                SWAP_LCOMM1:
1977 0F62 7E             LD    A,(HL)
1978 0F63 AA             XOR    D
1979 0F64 77             LD    (HL),A
1980 0F65 06 0A          LD    B,10
1981 0F67 CD 6B 0F        CALL  DELAY_S
1982 0F6A C9             RET
1983 0F6B                DELAY_S:
1984 0F6B CD EB 0F        CALL  DISP_7S
1985 0F6E CD 1F 11        CALL  DISP_L
1986 0F71 10 F8          DJNZ  DELAY_S
1987 0F73 D5             PUSH  DE
1988 0F74 7B             LD    A,E
1989 0F75 CD D9 0E        CALL  BIT_CLK
1990 0F78 D1             POP    DE
1991 0F79 C9             RET
1992                      ;
1993                      ;GENERATE WAIT (TW) AND SWAP DISPLAY 'D'
1994                      ;WAIT SW THEN RET
1995                      ;USE HL=B_SW_COMM,'D'
1996                      ;BC,DE,HL,AF

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1997 0F7A          WAIT:
1998 0F7A  CD 62 0F          CALL  SWAP_LCOMM1
1999 0F7D  CB 7E            BIT   7,(HL)
2000 0F7F  C8              RET   Z
2001 0F80  3E 03            LD    A,SONG_ERR
2002 0F82  CD C0 10          CALL  SONG
2003 0F85  18 F3            JR    WAIT
2004
2005                ;
2006                ;GENERATE WAIT (TW) 0->1->0 CHECK SW WAIT
2007                ;WHEN PRESS WAIT THEN QUIT
2008                ;USE D,E=0!,HL=B_SW_COMM
2009 0F87          TW_WAIT:
2010 0F87  3E 07            LD    A,SONG_WAI
2011 0F89  CD C0 10          CALL  SONG
2012 0F8C  06 08            LD    B,8
2013 0F8E  CD 6B 0F          CALL  DELAY_S
2014 0F91  CD 7A 0F          CALL  WAIT
2015 0F94  C9              RET
2016
2017
2018 0F95          SWITCH_P:
2019 0F95  79              LD    A,C
2020 0F96  D3 82            OUT   (P_SEL_SW),A
2021 0F98  DB 80            IN    A,(P_SW)
2022 0F9A  47              SWITCH_P1: LD    B,A
2023 0F9B  2F              CPL
2024 0F9C  B7              OR    A
2025 0F9D  C8              RET   Z
2026 0F9E  F5              PUSH  AF

```

2027	0F9F	C5	PUSH	BC
2028	0FA0	E5	PUSH	HL
2029	0FA1	21 AB 15	LD	HL,TAB_SW_CHANG
2030	0FA4	47	ONEKEY1: LD	B,A
2031	0FA5	BE	CP	(HL)
2032	0FA6	28 09	JR	Z,COMPLEMENT
2033	0FA8	7E	LD	A,(HL)
2034	0FA9	FE FF	CP	0FFH
2035	0FAB	28 1B	JR	Z,RETURN ;
2036	0FAD	23	INC	HL
2037	0FAE	78	LD	A,B
2038	0FAF	18 F3	JR	ONEKEY1
2039	0FB1		COMPLEMENT:	
2040	0FB1	47	LD	B,A
2041	0FB2	3A 2B 80	LD	A,(MODE_SW)
2042	0FB5	A0	AND	B
2043	0FB6	28 10	JR	Z,RETURN
2044	0FB8	E1	POP	HL
2045	0FB9	AE	XOR	(HL)
2046	0FBA	77	LD	(HL),A
2047	0FBB	E5	PUSH	HL
2048	0FBC	3E 00	LD	A,SONG_P0
2049	0FBE	CD C0 10	CALL	SONG
2050	0FC1	E1	POP	HL
2051	0FC2	C1	POP	BC
2052	0FC3	F1	POP	AF
2053	0FC4	CD CC 0F	CALL	REP_SW
2054	0FC7	C9	RET	
2055	0FC8	E1	RETURN: POP	HL
2056	0FC9	C1	POP	BC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2057 0FCA F1                POP    AF
2058 0FCB C9                RET
2059
2060                ;...UNREPEAT KEY..
2061                ;USE C=SELECT COLUMN
2062                ; B=KEY 'FIRST PRESS'
2063 0FCC                REP_SW:
2064 0FCC CD 1F 11          CALL    DISP_L
2065 0FCF CD EB 0F          CALL    DISP_7S
2066 0FD2 21 2A 80         LD      HL,CHECK
2067 0FD5 CB C6             SET     0,(HL)
2068 0FD7 CB 46             KCHECK: BIT    0,(HL)
2069 0FD9 28 0C             JR      Z,NOREP
2070 0FDB 79               LD      A,C
2071 0FDC D3 82             OUT     (P_SEL_SW),A
2072 0FDE DB 80             IN      A,(P_SW)
2073 0FE0 B8               CP      B
2074 0FE1 28 E9             JR      Z,REP_SW
2075 0FE3 CB 86             RES     0,(HL)
2076 0FE5 18 F0             JR      KCHECK
2077 0FE7                NOREP:
2078 0FE7 21 22 80         LD      HL,B_SW_COMM1
2079 0FEA C9                RET
2080 0FEB                DISP_7S:
2081 0FEB F5                PUSH    AF
2082 0FEC E5                PUSH    HL
2083 0FED C5                PUSH    BC
2084 0FEE D5                PUSH    DE
2085 0FEF 21 13 80         LD      HL,B_7S+19 ; displAy lAst to fiRst
2086                ; Ac->BR->sp->pc->Ab->Db->ad_B>da_B

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2087                                ;->IR->TEM
2088 0FF2 11 15 10                LD    DE,T_COL_7S
2089 0FF5 1A                      LD    A,(DE)
2090 0FF6 D3 91                    DIS1: OUT  (P_SEL_7S),A
2091 0FF8 7E                      LD    A,(HL)
2092 0FF9 2F                      CPL
2093 0FFA D3 90                    OUT  (P_DA_7S),A
2094 0FFC 2B                      DEC    HL
2095 0FFD 3E 82                    LD    A,130
2096 0FFF 3D                      DIS3: DEC    A
2097 1000 20 FD                    JR     NZ,DIS3
2098 1002 2F                      CPL
2099 1003 D3 90                    OUT  (P_DA_7S),A
2100 1005 3E 32                    LD    A,50
2101 1007 3D                      DIS4: DEC    A
2102 1008 20 FD                    JR     NZ,DIS4
2103 100A 13                      INC    DE
2104 100B 1A                      LD    A,(DE)
2105 100C FE 14                    CP     20
2106 100E 20 E6                    JR     NZ,DIS1
2107 1010 D1                      POP    DE
2108 1011 C1                      POP    BC
2109 1012 E1                      POP    HL
2110 1013 F1                      POP    AF
2111 1014 C9                      RET
2112 1015 01 00 03 02 05    T_COL_7S: DB    1,0,3,2,5,4,7,6,9,8
101A 04 07 06 09 08
2113 101F 0B 0A 0D 0C 0F                    DB    11,10,13,12,15,14,17
1024 0E 11
2114 1026 10 13 12 14                    DB    16,19,18,20

```

```

2115 102A          DEC_7S:
2116 102A  E5          PUSH  HL
2117 102B  C5          PUSH  BC
2118 102C  F5          PUSH  AF
2119 102D  11 13 80    LD     DE,B_7S+13H ;R displAy 7s
2120 1030  21 14 80    LD     HL,B_DA_M
2121 1033  06 0A          LD     B,0AH      ;decode20 locAte
2122 1035  0E 02          DEC1_7S: LD     C,2
2123 1037  AF          XOR     A
2124 1038  CD 48 10    DEC2_7S: CALL  DEC3_7S
2125 103B  1B          DEC     DE
2126 103C  0D          DEC     C
2127 103D  20 F9          JR     NZ,DEC2_7S
2128 103F  ED 6F          RLD
2129 1041  23          INC     HL
2130 1042  10 F1          DJNZ   DEC1_7S
2131 1044  F1          POP     AF
2132 1045  C1          POP     BC
2133 1046  E1          POP     HL
2134 1047  C9          RET
2135 1048  ED 6F          DEC3_7S: RLD
2136 104A  F5          PUSH  AF
2137 104B  E5          PUSH  HL
2138 104C  21 5E 10    LD     HL,TAB_7S
2139 104F  CD 57 10    CALL  FIND_COD_7S
2140 1052  7E          LD     A,(HL)
2141 1053  12          LD     (DE),A
2142 1054  E1          POP     HL
2143 1055  F1          POP     AF
2144 1056  C9          Ret

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

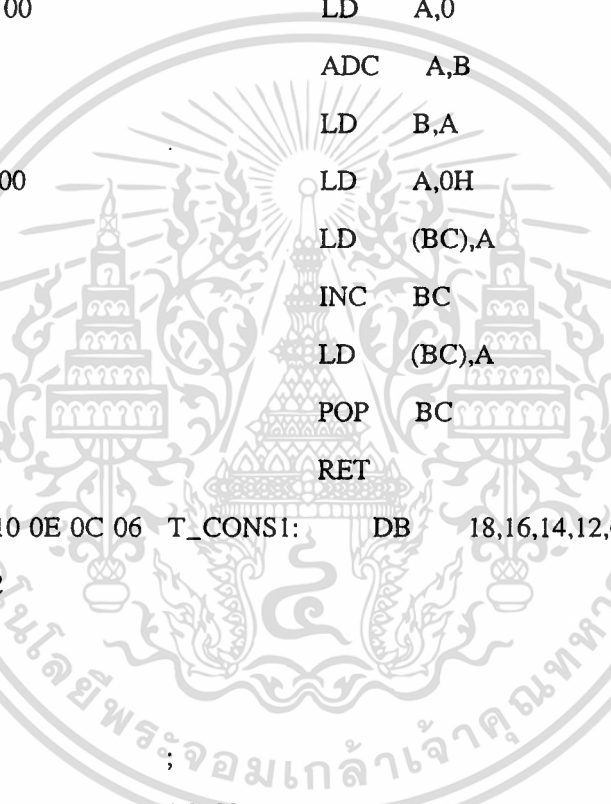
```

2145 1057          FIND_COD_7S:
2146 1057 85          ADD    A,L
2147 1058 6F          LD     L,A
2148 1059 3E 00       LD     A,0
2149 105B 8C          ADC    A,H
2150 105C 67          LD     H,A
2151 105D C9          RET
2152                ;*****<< tAble of code displAy 7s >>*****;
2153 105E 3F 06 5B 4F   TAB_7S:  db   3fh,06h,5bh,4fh ;0,1,2,3
2154 1062 66 6D 7D 07   db   66h,6dh,7dh,07h ;4,5,6,7
2155 1066 7F 6F 77 7C   db   7fh,6fh,77h,7ch ;8,9,A,b
2156 106A 39 5E 79 71   db   39h,5eh,79h,71h ;c,d,e,f
2157
2158
2159                ;..... sub select segment by 'B'.....;
2160                ; '0' = {SP,PC}, '1' = DA_M, '2' = AD_M, '3' = DB, '4' = AB
2161                ; '5' = BR, '6' = {ACC,TEMP}, '7' = IR
2162                ;USE B, '1' = DISABLE DISPLAY
2163 106E          MODE_7S:
2164 106E E5          PUSH    HL
2165 106F D5          PUSH    DE
2166 1070 C5          PUSH    BC
2167 1071 21 AC 15       LD     HL,T_CHK
2168 1074 11 B9 10       LD     DE,T_CONS1
2169 1077 48          LD     C,B
2170 1078 CB 40         BIT     0,B
2171 107A 28 0D         JR     Z,CHK_C1
2172 107C C5          PUSH    BC
2173 107D 01 08 80       LD     BC,B_7S+8
2174 1080 AF          XOR     A

```

2175	1081	02	LD	(BC),A
2176	1082	03	INC	BC
2177	1083	02	LD	(BC),A
2178	1084	03	INC	BC
2179	1085	02	LD	(BC),A
2180	1086	03	INC	BC
2181	1087	02	LD	(BC),A
2182	1088	C1	POP	BC
2183	1089	CB 70	CHK_C1: BIT	6,B
2184	108B	28 09	JR	Z,CHK_C
2185	108D	C5	PUSH	BC
2186	108E	01 00 80	LD	BC,B_7S
2187	1091	AF	XOR	A
2188	1092	02	LD	(BC),A
2189	1093	03	INC	BC
2190	1094	02	LD	(BC),A
2191	1095	C1	POP	BC
2192	1096		CHK_C:	
2193	1096	79	LD	A,C
2194	1097	A6	AND	(HL)
2195	1098	BE	CP	(HL)
2196	1099	CC A7 10	CALL	Z,CLR_7S
2197	109C	23	INC	HL
2198	109D	13	INC	DE
2199	109E	7E	LD	A,(HL)
2200	109F	FE FF	CP	0FFH
2201	10A1	20 F3	JR	NZ,CHK_C
2202	10A3	C1	POP	BC
2203	10A4	D1	POP	DE
2204	10A5	E1	POP	HL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

2205 10A6 C9 RET
2206 10A7 CLR_7S:
2207 10A7 C5 PUSH BC
2208 10A8 01 00 80 LD BC,B_7S
2209 10AB 1A LD A,(DE)
2210 10AC 81 ADD A,C
2211 10AD 4F LD C,A
2212 10AE 3E 00 LD A,0
2213 10B0 88 ADC A,B
2214 10B1 47 LD B,A
2215 10B2 3E 00 LD A,0H
2216 10B4 02 LD (BC),A
2217 10B5 03 INC BC
2218 10B6 02 LD (BC),A
2219 10B7 C1 POP BC
2220 10B8 C9 RET
2221 10B9 12 10 0E 0C 06 T_CONSI: DB 18,16,14,12,6,4,2
10BE 04 02
2222
2223
2224 ;
2225 10C0 SONG:
2226 10C0 F5 PUSH AF
2227 10C1 E5 PUSH HL
2228 10C2 D5 PUSH DE
2229 10C3 C5 PUSH BC
2230 10C4 47 LD B,A
2231 10C5 3A 59 80 LD A,(B_ON_SONG)
2232 10C8 FE 00 CP 0
2233 10CA 20 0C JR NZ,EN_SONGS

```

```

2234 10CC 78          LD    A,B
2235 10CD 21 6B 14    LD    HL,TAB_SONG
2236 10D0 CD DD 10    CALL  FIND_TA
2237 10D3 D5          PUSH  DE
2238 10D4 E1          POP   HL
2239 10D5 CD E6 10    CALL  SOUND
2240 10D8 C1          EN_SONGS: POP  BC
2241 10D9 D1          POP  DE
2242 10DA E1          POP  HL
2243 10DB F1          POP  AF
2244 10DC C9          RET
2245                  ;..... SUB FOR FIND TABLE WORD .....;
2246                  ;....USE 'A' FOR FIND,'HL' TABLE IN AND OUTPUT ...;
2247 10DD          FIND_TA:
2248 10DD 87          ADD    A,A
2249 10DE 16 00       LD     D,0
2250 10E0 5F          LD     E,A
2251 10E1 19          ADD    HL,DE
2252 10E2 5E          LD     E,(HL)
2253 10E3 23          INC    HL
2254 10E4 56          LD     D,(HL)
2255 10E5 C9          RET
2256                  ;.....SUB FOR SONG GENERATE.....;
2257                  ;....USE HL OF SONG TABLE
2258 10E6          SOUND:
2259 10E6 7E          LD     A,(HL)
2260 10E7 FE FF       CP     0FFH
2261 10E9 28 13      JR     Z,ESOU
2262 10EB 4F          LD     C,A
2263 10EC 23          INC    HL

```

```

2264 10ED 7E          LD    A,(HL)
2265 10EE E5          PUSH  HL
2266 10EF 21 7F 15     LD    HL,TAB_TIM
2267 10F2 CD DD 10     CALL  FIND_TA
2268 10F5 D5          PUSH  DE
2269 10F6 E1          POP   HL
2270 10F7 CD 03 11     CALL  SOUND1
2271 10FA E1          POP   HL
2272 10FB 23          INC   HL
2273 10FC 18 E8        JR    SOUND
2274 10FE 3E 01        ESOU: LD    A,01      ;'1' STOP OSCILLAT
2275 1100 D3 A1        OUT   (SPEAK),A
2276 1102 C9          RET
2277                  SOUND1: ;..... USE C =Frequency, HL =Time .....;
2278 1103 16 00        SOUND11: LD  D,0
2279 1105 3E 01        LD    A,00000001B
2280 1107 CD 12 11     CALL  SOUND12
2281 110A AF          XOR    A
2282 110B CD 12 11     CALL  SOUND12
2283 110E 15          DEC    D
2284 110F 20 F2        JR    NZ,SOUND11
2285 1111 C9          RET
2286 1112 D3 A1        SOUND12: OUT  (SPEAK),A
2287 1114 59          LD    E,C
2288 1115 2B          SOUND13: DEC  HL
2289 1116 7C          LD    A,H
2290 1117 B5          OR     L
2291 1118 20 01        JR    NZ,SOUND14
2292 111A 14          INC    D
2293 111B              SOUND14:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2294 111B 1D          DEC    E
2295 111C 20 F7       JR     NZ,SOUND13
2296 111E C9          RET
2297                  ;
2298                  ;
2299 111F              DISP_L:
2300 111F F5          PUSH   AF
2301 1120 E5          PUSH   HL
2302 1121 D5          PUSH   DE
2303 1122 21 3E 14    LD     HL,T_SEL_L
2304 1125 11 20 80    LD     DE,B_SW_ADDR
2305 1128              LDISP_L:
2306 1128 7E          LD     A,(HL)
2307 1129 FE FF       CP     0FFH
2308 112B 28 17       JR     Z,END_L
2309 112D D3 82       OUT    (P_SEL_L),A
2310 112F 1A          LD     A,(DE)
2311 1130 2F          CPL
2312 1131 D3 81       OUT    (P_LED),A
2313 1133 3E C8       LD     A,200
2314 1135 3D          L1_DL: DEC    A
2315 1136 20 FD       JR     NZ,L1_DL
2316 1138 2F          CPL
2317 1139 D3 81       OUT    (P_LED),A
2318 113B 3E 14       LD     A,20
2319 113D 3D          L2_DL: DEC    A
2320 113E 20 FD       JR     NZ,L2_DL
2321 1140 23          INC     HL
2322 1141 13          INC     DE
2323 1142 18 E4       JR     LDISP_L

```

```

2324 1144 D1          END_L: POP  DE
2325 1145 E1          POP  HL
2326 1146 F1          POP  AF
2327 1147 C9          RET
2328                ;....SUB KEY MATRIX....;
2329                ; USE MODE_ADDR,MODE_DATA,MODE_FNC
2330                ; MUST 'CALL' INSTRUCTION ONLY!
2331 1148                KEY_MAT:
2332 1148 21 23 80      LD    HL,B_SW_COMM2
2333 114B CB FE        SET    7,(HL)
2334 114D 2B          DEC    HL
2335 114E C5          PUSH   BC
2336 114F CB B6       RES    6,(HL) ;CLR ROM LED
2337 1151 3E 04      LD     A,SONG_MAT
2338 1153 CD C0 10    CALL   SONG
2339 1156 3E A0      LD     A,10100000B ;SET BIT DATA
2340 1158 32 2C 80    LD     (MODE_K),A
2341 115B CD 2A 12    CALL   MODE_DATA
2342 115E C1          POP    BC
2343 115F CD CC 0F    CALL   REP_SW
2344 1162                KEY11:
2345 1162 21 2D 80      LD     HL,B_K
2346 1165 7E          LD     A,(HL)
2347 1166 FE A1      CP     0A1H ;IF PLESS 'INC'->
2348                ;B_AD_M+1
2349 1168 20 0C      JR     NZ,KEY2
2350 116A 32 2C 80    LD     (MODE_K),A
2351 116D 3A 15 80    LD     A,(B_AD_M)
2352 1170 3C          INC    A
2353 1171 32 15 80    LD     (B_AD_M),A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2354 1174 18 18          JR    NO_SET_M
2355 1176                KEY2:
2356 1176 FE A2          CP    0A2H      ;IF PRESS 'DEC'->
2357 1178 20 0D          JR    NZ,KEY3      ;B_AD_M-1
2358 117A 3D            DEC    A
2359 117B 32 2C 80       LD    (MODE_K),A
2360 117E 3A 15 80       LD    A,(B_AD_M)
2361 1181 3D            DEC    A
2362 1182 32 15 80       LD    (B_AD_M),A
2363 1185 18 07          JR    NO_SET_M
2364 1187
2365 1187 FE 80          KEY3: CP    80H
2366 1189 38 03          JR    C,NO_SET_M      ;IF
FNC,ADDR,DATA
2367                                ; THEN
PUSH
2368 118B 32 2C 80       LD    (MODE_K),A      ;(MODE_K)<-(B_K)
2369                                NO_SET_M:      ;FNC =11000000
2370                                ;DATA =10100000
2371                                ;
2372                                ;OPERATION KEY UNIT;
2373                                ;.....;
2374 118E 3A 2C 80       LD    A,(MODE_K)
2375 1191 F5              PUSH   AF
2376 1192 CB 77          BIT    6,A
2377 1194 C4 96 12       CALL   NZ,MODE_FNC
2378 1197 F1              POP    AF
2379 1198 F5              PUSH   AF
2380 1199 CB 6F          BIT    5,A
2381 119B C4 2A 12       CALL   NZ,MODE_DATA

```

```

2382 119E F1                POP    AF
2383 119F F5                PUSH   AF
2384 11A0 CB 67             BIT     4,A
2385 11A2 C4 D1 11         CALL    NZ,MODE_ADDR
2386 11A5 F1                POP     AF
2387 11A6 3E 02            LD      A,2
2388 11A8 4F               LD      C,A
2389 11A9 D3 82            OUT     (P_SEL_SW),A
2390 11AB DB 80            IN      A,(P_SW)
2391 11AD 47               LD      B,A
2392 11AE CB 47             BIT     0,A
2393 11B0 C2 62 11         JP      NZ,KEY11
2394 11B3 3E 04            LD      A,SONG_MAT ;END MATRIX
2395 11B5 CD C0 10         CALL    SONG
2396 11B8 CD CC 0F         CALL    REP_SW
2397 11BB 21 23 80         LD      HL,B_SW_COMM2
2398 11BE CB BE            RES     7,(HL) ;CLR LED USE KEY
2399 11C0 C9               RET
2400 11C1                DISP_CUR:
2401                ;SHOW 7S OF ADDRESS & DATA FROM CURRENT!
2402 11C1 CD 57 12         CALL    P_MEM
2403 11C4 7E               LD      A,(HL)
2404 11C5 32 14 80         LD      (B_DA_M),A
2405 11C8 CD 2A 10         CALL    DEC_7S
2406 11CB 06 F9            LD      B,11111001B
2407 11CD CD 6E 10         CALL    MODE_7S
2408 11D0 C9               RET
2409 11D1                MODE_ADDR:
2410                ;.....;
2411 11D1 AF                XOR     A

```

```

2412 11D2 32 15 80          LD   (B_AD_M),A
2413 11D5 CD 2A 10          CALL DEC_7S
2414 11D8 06 FB            LD   B,11111011B
2415 11DA CD 6E 10          CALL MODE_7S
2416 11DD                  ADD1:
2417 11DD CD CF 13          CALL SCAN_K
2418 11E0 FE 80            CP   80H
2419 11E2 D0                RET   NC           ;WHEN

```

A=FNC,DATA,ADDR

```

2420 11E3 3A 2C 80          LD   A,(MODE_K)
2421 11E6 CB 47             BIT   0,A
2422 11E8 20 1D            JR   NZ,RED_A
2423 11EA CB C7            SET   0,A
2424 11EC 32 2C 80          LD   (MODE_K),A
2425 11EF 3E 00            LD   A,0
2426 11F1 ED 6F            RLD           ;SAVE & ROTATE
2427 11F3 7E              LD   A,(HL)
2428 11F4 32 15 80          LD   (B_AD_M),A
2429 11F7 CD 2A 10          CALL DEC_7S
2430 11FA 06 FB            LD   B,11111011B
2431 11FC CD 6E 10          CALL MODE_7S
2432 11FF 3E 80            LD   A,80H
2433 1201 32 10 80          LD   (B_7S+10H),A
2434 1204 C3 DD 11          JP   ADD1
2435 1207                  RED_A:
2436 1207 21 15 80          LD   HL,B_AD_M
2437 120A CD 24 12          CALL P_BUF
2438 120D 3A 2C 80          LD   A,(MODE_K)
2439 1210 CB 87            RES   0,A
2440 1212 32 2C 80          LD   (MODE_K),A

```

2441	1215	CD 2A 10		CALL	DEC_7S
2442	1218	06 FB		LD	B,11111011B
2443	121A	CD 6E 10		CALL	MODE_7S
2444	121D	C3 DD 11		JP	ADD1
2445	1220	C3 CF 13	SCAN_K1:	JP	SCAN_K
2446	1223	C9		RET	
2447	1224		P_BUF:		
2448	1224	3A 2D 80		LD	A,(B_K)
2449	1227	B6		OR	(HL)
2450	1228	77		LD	(HL),A
2451	1229	C9		RET	
2452					
2453					
2454	122A		MODE_DATA:		
2455			;.....;		
2456	122A	CD C1 11		CALL	DISP_CUR
2457	122D	CD CF 13	M1_DA:	CALL	SCAN_K
2458	1230	FE 80		CP	80H
2459	1232	D0		RET	NC
2460	1233	21 2C 80		LD	HL,MODE_K
2461	1236	CB 4E		BIT	1,(HL)
2462	1238	20 2A		JR	NZ,RED_D
2463	123A	CB CE		SET	1,(HL)
2464	123C	21 2D 80		LD	HL,B_K
2465	123F	3E 00		LD	A,0
2466	1241	ED 6F		RLD	
2467	1243	7E		LD	A,(HL)
2468	1244	32 14 80		LD	(B_DA_M),A
2469	1247	CD 2A 10		CALL	DEC_7S
2470	124A	06 F9		LD	B,11111001B

```

2471 124C CD 6E 10          CALL  MODE_7S
2472 124F 3E 80            LD    A,80H
2473 1251 32 12 80         LD    (B_7S+12H),A
2474 1254 C3 2D 12         JP    M1_DA
2475 1257                  P_MEM:
2476 1257 CD 80 01         P_MEMS: CALL  CH_RA_RO
2477 125A 3A 15 80         LD    A,(B_AD_M)
2478 125D 85              SRCHAD: ADD  A,L
2479 125E 6F              LD    L,A
2480 125F 3E 00           LD    A,0
2481 1261 8C              ADC    A,H
2482 1262 67              LD    H,A
2483 1263 C9              RET
2484 1264                  RED_D:
2485 1264 21 14 80         LD    HL,B_DA_M
2486 1267 CD 24 12         CALL  P_BUF
2487 126A 3A 2C 80         LD    A,(MODE_K)
2488 126D CB 8F           RES    1,A
2489 126F 32 2C 80         LD    (MODE_K),A
2490 1272 CD 57 12         CALL  P_MEM
2491 1275 3A 14 80         LD    A,(B_DA_M)
2492 1278 77              LD    (HL),A
2493 1279 CD 2A 10         CALL  DEC_7S
2494 127C 06 F9           LD    B,11111001B
2495 127E CD 6E 10         CALL  MODE_7S
2496                      ;.....
2497                      ;INC B_AD_M AND DELAY SHOW
2498                      ;.....
2499 1281 3E 14            LD    A,20
2500 1283 CD EB 0F         RDS:   CALL  DISP_7S

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2501 1286 CD 1F 11          CALL DISP_L
2502 1289 3D              DEC A
2503 128A 20 F7          JR NZ,RDS
2504 128C 3A 15 80       LD A,(B_AD_M)
2505 128F 3C              INC A
2506 1290 32 15 80       LD (B_AD_M),A
2507 1293 C3 2A 12       JP MODE_DATA
2508                      ;.....
2509 1296                MODE_FNC:
2510                      ;.....
2511                      ;DISPLAY 'Fn' AT 'AD_M'
2512 1296 3E 0E          LD A,FUNCTION
2513 1298 32 52 80       LD (B_SEL_SHO),A
2514 129B CD AF 13       CALL SH_AD
2515 129E CD CF 13       CALL SCAN_K
2516 12A1 FE 80          CP 80H
2517 12A3 D0             RET NC
2518 12A4 21 E6 15       LD HL,TAB_FNC
2519 12A7 CD DD 10       CALL FIND_TA
2520 12AA 3E A0          LD A,10100000B ;SET MODE DATA
2521 12AC 32 2C 80       LD (MODE_K),A
2522 12AF D5             PUSH DE
2523 12B0 C9             RET
2524 12B1 DD 21 00 80    F_NSHOW: LD IX,B_7S
2525 12B5 DD 36 11 71    LD (IX+17),01110001B ;F
2526 12B9 DD 36 10 54    LD (IX+16),01010100B ;n
2527 12BD DD 36 13 40    LD (IX+19),01000000B ;-
2528 12C1 DD 36 12 40    LD (IX+18),01000000B ;-
2529 12C5 C9             RET
2530 12C6                FN0:

```

2531	12C6	3E 0D	LD	A,CLEAR_RAM	
2532	12C8	32 52 80	LD	(B_SEL_SHO),A	
2533	12CB	CD AF 13	CALL	SH_AD	
2534	12CE	06 0F	LD	B,15	
2535	12D0	CD EA 03	CALL	DELAY_CPU	
2536	12D3	21 00 86	LD	HL,RAM	
2537	12D6	11 01 86	LD	DE,RAM+1	
2538	12D9	01 00 01	LD	BC,256D	
2539	12DC	AF	XOR	A	
2540	12DD	77	LD	(HL),A	
2541	12DE	ED B0	LDIR		
2542	12E0	C9	RET		
2543	12E1	DD 21 00 80	C_RAM: LD	IX,B_7S	
2544	12E5	DD 36 11 58	LD	(IX+17),01011000B ;c	
2545	12E9	DD 36 10 38	LD	(IX+16),00111000B ;L	
2546	12ED	DD 36 13 79	LD	(IX+19),01111001B ;E	
2547	12F1	DD 36 12 50	LD	(IX+18),01010000B ;r	
2548	12F5	C9	RET		
2549	12F6		FN1:		
2550				;ADJUST VELOCITY OF INTERRUPT	
2551				;PUSH DATA '01'= MAX SPEED	
2552				;DATA '00'=MIN SPEED	
2553	12F6	3E 03	LD	A,SPEED_INT	
2554	12F8	32 52 80	LD	(B_SEL_SHO),A	
2555	12FB	3A 53 80	LD	A,(B_SPEED_INT)	
2556	12FE	CD 6F 13	CALL	P_D_2	
2557	1301	3A 14 80	LD	A,(B_DA_M)	
2558	1304	32 53 80	LD	(B_SPEED_INT),A	
2559	1307	CD 8D 01	FN2S: CALL	CUR_MEM	;CURRENT

DATA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2560 130A 06 F9          LD    B,11111001B
2561 130C CD 6E 10      CALL  MODE_7S
2562 130F C9           RET
2563 1310              FN2:
2564              ;ADJUST VELOCITY OF RUN
2565              ;PUSH DATA '01'=MAX SPEED
2566 1310 3E 04          LD    A,SPEED_RUN
2567 1312 32 52 80      LD    (B_SEL_SHO),A
2568 1315 3A 54 80      LD    A,(B_SPEED_RUN)
2569 1318 CD 6F 13      CALL  P_D_2
2570 131B 3A 14 80      LD    A,(B_DA_M)
2571 131E 32 54 80      LD    (B_SPEED_RUN),A
2572 1321 C3 07 13      JP    FN2S
2573 1324              FN3:
2574 1324 3E 0F          LD    A,ON_OFFSONG
2575 1326 CD AF 13      CALL  SH_AD
2576 1329 3A 59 80      LD    A,(B_ON_SONG)
2577 132C FE 00          CP    0
2578 132E 28 11          JR    Z,OFF_SONG
2579 1330 3E 3F          LD    A,3FH          ;O
2580 1332 32 13 80      LD    (B_7S+19),A
2581 1335 3E 54          LD    A,54H          ;n
2582 1337 32 12 80      LD    (B_7S+18),A
2583 133A 3E 00          LD    A,0H
2584 133C 32 59 80      LD    (B_ON_SONG),A
2585 133F 18 0F          JR    FN3_SHOW
2586 1341              OFF_SONG:
2587 1341 3E 3F          LD    A,3FH          ;O
2588 1343 32 13 80      LD    (B_7S+19),A
2589 1346 3E 71          LD    A,71H          ;F

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2590 1348 32 12 80          LD    (B_7S+18),A
2591 134B 3E FF             LD    A,0FFH
2592 134D 32 59 80          LD    (B_ON_SONG),A
2593 1350 06 0F             FN3_SHOW: LD    B,15
2594 1352 CD EA 03          CALL  DELAY_CPU
2595 1355 C9                RET
2596 1356 DD 21 00 80        O_F_SONG: LD    IX,B_7S
2597 135A DD 36 11 6D          LD    (IX+17),01101101B ;S
2598 135E DD 36 10 3F          LD    (IX+16),00111111B ;O
2599 1362 C9                RET
2600 1363 C9                FN4:    RET
2601 1364 C9                FN5:    RET
2602 1365 C9                FN6:    RET
2603 1366 C9                FN7:    RET
2604 1367 C9                FN8:    RET
2605 1368 C9                FN9:    RET
2606 1369 C9                FNA:    RET
2607 136A C9                FNB:    RET
2608 136B C9                FNC:    RET
2609 136C C9                FND:    RET
2610 136D C9                FNE:    RET
2611 136E C9                FNF:    RET
2612 136F                    P_D_2:
2613                        ;PUSH 2 TIME OF DATA AND EXIT BY 'INC.'
2614                        ;AND SHOW AT (B_DA_M)
2615 136F 32 14 80          LD    (B_DA_M),A
2616 1372 CD 2A 10          CALL  DEC_7S
2617 1375 06 FD             LD    B,11111101B
2618 1377 CD 6E 10          CALL  MODE_7S
2619 137A CD AF 13          CALL  SH_AD

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2620 137D  CD CF 13          CALL  SCAN_K
2621 1380  FE 80             CP    80H          ;NOT EXIT '0-F'
2622 1382  D0                RET    NC
2623 1383  21 2D 80          LD     HL,B_K          ;1'st PRESSED
2624 1386  3E 00             LD     A,0
2625 1388  ED 6F             RLD
2626 138A  7E                LD     A,(HL)
2627 138B  32 14 80          LD     (B_DA_M),A      ;SAVE DATA

```

UPPER

```

2628 138E  CD 2A 10          CALL  DEC_7S
2629 1391  06 FD             LD     B,11111101B    ;SHOW DATA!
2630 1393  CD 6E 10          CALL  MODE_7S
2631 1396  3E 80             LD     A,80H
2632 1398  32 12 80          LD     (B_7S+12H),A
2633 139B  CD AF 13          CALL  SH_AD
2634 139E  CD CF 13          CALL  SCAN_K
2635 13A1  FE 80             CP    80H          ;2'nd PRESSED
2636 13A3  D0                RET    NC          ;
2637 13A4  21 14 80          LD     HL,B_DA_M
2638 13A7  3A 2D 80          LD     A,(B_K)
2639 13AA  B6                OR     (HL)
2640 13AB  77                LD     (HL),A
2641 13AC  C3 6F 13          JP     P_D_2

```

2642

2643 ;SHOW B_AD_M BY SELECT 'B_SEL_SHO'

2644 ;BY PUSH DATA TO 'AD_M'

2645 13AF SH_AD:

2646 13AF 3A 52 80 LD A,(B_SEL_SHO)

2647 13B2 C3 5A 07 JP STP2_COMD

2648 ;PUSH 'SP' TO 'AD_M' FOR SHOW

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2649 13B5 DD 21 00 80 P_SP: LD IX,B_7S
2650 13B9 DD 36 11 06 LD (IX+17),00000110B ;'T'
2651 13BD DD 36 10 54 LD (IX+16),01010100B ;'n'
2652 13C1 C9 RET
2653 13C2 DD 21 00 80 P_RN: LD IX,B_7S
2654 13C6 DD 36 11 50 LD (IX+17),01010000B ;'r'
2655 13CA DD 36 10 54 LD (IX+16),01010100B ;'n'
2656 13CE C9 RET
2657 ;.....;
2658 13CF SCAN_K:
2659 ;.....;
2660
2661 13CF 3E 02 LD A,2
2662 13D1 D3 82 OUT (P_SEL_SW),A
2663 13D3 DB 80 IN A,(P_SW)
2664 13D5 CB 47 BIT 0,A
2665 13D7 C8 RET Z
2666 13D8 11 B4 15 NO_E_K: LD DE,T_K
2667 13DB SC1_K:
2668 13DB 1A LD A,(DE)
2669 13DC FE FF CP 0FFH
2670 13DE 28 EF JR Z,SCAN_K
2671 13E0 D3 91 OUT (P_SEL_7S),A
2672 13E2 DB 92 IN A,(P_SEL_7S+1)
2673 13E4 E6 F0 AND 0F0H
2674 13E6 47 LD B,A
2675 13E7 1A LD A,(DE)
2676 13E8 4F LD C,A
2677 13E9 13 INC DE
2678 13EA DB 92 IN A,(P_SEL_7S+1)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2679 13EC E6 F0          AND    0F0H
2680 13EE CD 1F 11       CALL   DISP_L
2681 13F1 CD EB 0F       CALL   DISP_7S
2682 13F4 78            LD     A,B
2683 13F5 FE F0          CP     0F0H
2684 13F7 28 E2          JR     Z,SC1_K
2685 13F9 3E 08          LD     A,SONG1_K
2686 13FB CD C0 10       CALL   SONG
2687 13FE                REP_K:
2688 13FE CD EB 0F       CALL   DISP_7S
2689 1401 CD 1F 11       CALL   DISP_L
2690 1404 21 2A 80       LD     HL,CHECK
2691 1407 CB C6          SET    0,(HL)
2692 1409                R1_K:
2693 1409 CB 46          BIT     0,(HL)
2694 140B 28 0E          JR     Z,NO_K
2695 140D 79            LD     A,C
2696 140E D3 91          OUT    (P_SEL_7S),A
2697 1410 DB 92          IN     A,(P_SEL_7S+1)
2698 1412 E6 F0          AND    0F0H
2699 1414 B8            CP     B
2700 1415 28 E7          JR     Z,REP_K
2701 1417 CB 86          E_R_K: RES    0,(HL)
2702 1419 18 EE          JR     R1_K
2703 141B                NO_K:
2704 141B 79            LD     A,C
2705 141C E6 0F          AND    0FH
2706 141E B0            OR     B
2707                    ;'A' = KEY CODE
2708 141F 21 BB 15       LD     HL,T_K1-1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2709 1422 11 D0 15          LD    DE,T_K2-1
2710 1425                  FR_K:
2711 1425 23              INC    HL
2712 1426 13              INC    DE
2713 1427 BE              CP     (HL)
2714 1428 20 FB          JR     NZ,FR_K
2715                      ;(DE)=KEY DECODE
2716 142A 1A              LD     A,(DE)
2717 142B 21 2D 80        LD     HL,B_K
2718 142E 77              LD     (HL),A
2719 142F C9              RET
2720                      ;....SUB SONG BUS I/O UNIT....;
2721                      ;
2722 1430 3E 00          SONG_BUS: LD     A,0
2723 1432 CD C0 10        CALL    SONG
2724 1435 3C              INC     A
2725 1436 CD C0 10        CALL    SONG
2726 1439 3C              INC     A
2727 143A CD C0 10        CALL    SONG
2728 143D C9              RET
2729 143E 00 10 20 30 40  T_SEL_L: DB     00,10H,20H,30H,40H,50H,0FFH
1443 50 FF
2730                      ;
2731                      ;****<< TABLE OF DIAGRAM >>****;
2732                      ;          |WAI|ROM|IRDI|WRIM|REI|IOR|KEY|CLK|
2733 1445 68          TAB_DIA1: DB     01101000B ;READ MEMORY (ROM)
2734 1446 28          DB     00101000B ;READ MEMORY (RAM)
2735 1447 18          DB     00011000B ;WRITE MEMORY (RAM)
2736 1448 24          DB     00100100B ;READ PORT DIAGRAM
2737 1449 64          DB     01100100B ;READ PORT DIAGRAM

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2738 144A 14 DB 00010100B ;WRITE PORT DIAGRAM
2739 144B 54 DB 01010100B ;WRITE PORT DIAGRAM
2740 144C 00 TAB_DIA2:DB 00000000B ;NO OPERATION
2741 ;*****<< TABLE EXECUIT DIAGRAM >>*****;
2742 144D CF0D CF0D 200E TAB_EXE_DIA:
DW RDAD_ROM,READ_RAM,WRITE_RAM
2743 1453 960E 960E 5D0E DW READ_PORT,READ_PORT,WRITE_PORT
2744 1459 5D0E DW WRITE_PORT
2745 ;*****<< TABLE ALU >>*****;
2746 145B 6902 6E02 7502 T_ALU: DW ALU0,ALU1,ALU2,ALU3
1461 7C02
2747 1463 8302 8A02 9E02 DW ALU4,ALU5,ALU6,ALU7
1469 A702
2748
2749 ;TABLE SOUND OF TIME OR FREQUENCY ;
2750 146B TAB_SONG:
2751 146B 8714 9414 A714 DW SONG_0,SONG_1,
SONG_2,SONG_ERRO
1471 BA14
2752 1473 C914 DE14 F314 DW SONG_K_MAT,
SONG_RD0,SONG_WR0
2753 1479 0815 1D15 2A15 DW SONG_WA,SONG_K1,
SON_INT,SON_NMI
147F 3915
2754 1481 4815 5715 6E15 DW SON_STP,SON_ALU,SON_EXE
2755 ;<< DATA OF SONG_1 - SONG_n >>;
2756 ; FiRst Byte ARe ffrequency, Second Byte ARe time
2757 ; FRequency=0FFh ALe end of song
2758
2759 1487 0B 01 FF SONG_0: DB 0BH,1,0FFH

```

2760 148A 00 00 00 00 00 DB 00,00,00,00,00,00,00,00,00,00
 148F 00 00 00 00 00
 2761 1494 10 02 20 03 40 SONG_1: DB 10H,2,20H,3,40H,4,9,3,0FFH
 1499 04 09 03 FF
 2762 149D 00 00 00 00 00 DB 00,00,00,00,00,00,00,00,00,00
 14A2 00 00 00 00 00
 2763 14A7 15 05 64 09 01 SONG_2: DB 21,5,100,9,1,7,11,3,0FFH
 14AC 07 0B 03 FF
 2764 14B0 00 00 00 00 00 DB 00,00,00,00,00,00,00,00,00,00
 14B5 00 00 00 00 00
 2765 14BA A0 05 50 04 FF SONG_ERRO: DB 0A0H,5,80,4,0FFH
 2766 14BF 00 00 00 00 00 DB 00,00,00,00,00,00,00,00,00,00
 14C4 00 00 00 00 00
 2767 14C9 D0 05 01 03 90 SONG_K_MAT: DB 0D0H,5,01,3,90H,2,01,3,90H,2,0FFH
 14CE 02 01 03 90 02
 14D3 FF
 2768 14D4 00 00 00 00 00 DB 00,00,00,00,00,00,00,00,00,00
 14D9 00 00 00 00 00
 2769 14DE 01 09 70 10 01 SONG_RD0: DB 01H,9,070H,16,01H,9,090H,16,01H,9
 14E3 09 90 10 01 09
 2770 14E8 FF DB 0FFH
 2771 14E9 00 00 00 00 00 DB 00,00,00,00,00,00,00,00,00,00
 14EE 00 00 00 00 00
 2772 14F3 10 05 01 07 20 SONG_WR0:DB 10H,5,01,7,20H,5,01,7,10H,5,0FFH
 14F8 05 01 07 10 05
 14FD FF
 2773 14FE 00 00 00 00 00 DB 00,00,00,00,00,00,00,00,00,00
 1503 00 00 00 00 00
 2774 1508 90 02 01 03 90 SONG_WA: DB 90H,2,01,3,90H,2,01,3,90H,2,0FFH
 150D 02 01 03 90 02

1512 FF

2775 1513 00 00 00 00 00 DB 00,00,00,00,00,00,00,00,00,00

1518 00 00 00 00 00

2776 151D 20 02 FF SONG_K1: DB 20H,2,0FFH

2777 1520 00 00 00 00 00 DB 00,00,00,00,00,00,00,00,00,00

1525 00 00 00 00 00

2778 152A 30 01 09 03 FF SON_INT: DB 30H,1,09,3,0FFH

2779 152F 00 00 00 00 00 DB 00,00,00,00,00,00,00,00,00,00

1534 00 00 00 00 00

2780 1539 09 03 30 01 FF SON_NMI: DB 09,3,30H,1,0FFH

2781 153E 00 00 00 00 00 DB 00,00,00,00,00,00,00,00,00,00

1543 00 00 00 00 00

2782 1548 09 02 15 04 FF SON_STP: DB 09,2,15H,4,0FFH

2783 154D 00 00 00 00 00 DB 00,00,00,00,00,00,00,00,00,00

1552 00 00 00 00 00

2784 1557 46 01 05 02 08 SON_ALU: DB 70,1,05,2,08,3,15H,4,30H,4,05,2,0FFH

155C 03 15 04 30 04

1561 05 02 FF

2785 1564 00 00 00 00 00 DB 00,00,00,00,00,00,00,00,00,00

1569 00 00 00 00 00

2786 156E 04 01,00 06 0A SON_EXE: DB 04,1,00,6,10,3,0FFH

1573 03 FF

2787 1575 00 00 00 00 00 DB 00,00,00,00,00,00,00,00,00,00

157A 00 00 00 00 00

2788 ;<< TIME SONG FROM SONG_1 - SONG_n >>;

2789 157F 0011 0013 0015 TAB_TIM: DW 1100H,1300H,1500H,1700H,1900H

1585 0017 0019

2790 1589 001B 001D 001F DW 1B00H,1D00H,1F00H,2100H,2300H

158F 0021 0023

2791 1593 0040 0050 0060 DW 4000H,5000H,6000H,7000H,8000H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1599 0070 0080
2792 159D 0090 00A0 00B0          DW  9000H,0A000H,0B000H,0C000H,0D000H
15A3 00C0 00D0
2793 15A7 00E0 00F0          DW  0E000H,0F000H
2794                                ;***** TABLE FIND ONEKEY*****;
2795
2796 15AB 01          TAB_SW_CHANG: DB  01H
2797 15AC 02 04 08    T_CHK:          DB  02H,04H,08H
2798 15AF 10 20 40 80          DB  10H,20H,40H,80H
2799 15B3 FF          DB  0FFH
2800                                ;*** TABLE SCAN KEY MATRIX ***;
2801 15B4 03 04 05 06    T_K:          DB  03H,04H,05H,06H
2802 15B8 07 08 09 FF          DB  07H,08H,09H,0FFH
2803 15BC E3 E4 E5 E6 E7 T_K1:        DB  0E3H,0E4H,0E5H,0E6H
                                         0E7H,0E8H,0E9H
15C1 E8 E9
2804 15C3 D3 D4 D5 D6 D7          DB  0D3H,0D4H,0D5H,0D6H
                                         0D7H,0D8H,0D9H
15C8 D8 D9
2805 15CA B3 B4 B5 B6 B7          DB  0B3H,0B4H,0B5H,0B6H
                                         0B7H,0B8H,0B9H
15CF B8 B9
2806 15D1 C0 0F 0E 0D 0C T_K2:        DB  0C0H,0FH,0EH,0DH,0CH,0BH,0AH
15D6 0B 0A
2807 15D8 A0 A1 09 08 07          DB  0A0H,0A1H,09H,08H,07H,06H,05H
15DD 06 05
2808 15DF 90 A2 04 03 02          DB  090H,0A2H,04H,03H,02H,01H,00H
15E4 01 00
2809 15E6 C612 F612 1013 TAB_FNC: DW  FN0,FN1,FN2,FN3,FN4,FN5
15EC 2413 6313 6413

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

2810 15F2 6513 6613 6713          DW    FN6,FN7,FN8,FN9,FNA,FNB
      15F8 6813 6913 6A13
2811 15FE 6B13 6C13 6D13          DW    FNC,FND,FNE,FNF
      1604 6E13

2812                                ;.....BUFFER.....;
2813                                ;must be to RAM only!!!
2814                                ;.....;

2815 8000                          ORG AREA_B
2816 8000      B_7S:      DS 20  ; 10S,9S=TEM
2817                                ; 8S,7S=IR
2818                                ; 15,14=AccumulAtoR
2819                                ; 13,12RegisteR B
2820                                ; 11,10pRog
2821                                ; 9,8stAck poRnteR
2822                                ; 7,6AddResS buffeR
2823                                ; 5,4dAtA buffeR
2824                                ; 3,2addResS memoRy
2825                                ; 1,0data memoRy
2826 8014      B_DA_M:      DS 1
2827 8015      B_AD_M:      DS 1
2828 8016      B_DB:      DS 1
2829 8017      B_AB:      DS 1
2830 8018      B_SP:      DS 1
2831 8019      B_PC:      DS 1
2832 801A      B_BR:      DS 1
2833 801B      B_ACC:      DS 1
2834 801C      B_IR:      DS 1
2835 801D      B_TEM:      DS 1
2836 801E      B_FLX      DS 1

```

DS 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2838	8020	B_SW_ADDR:	DS 1	
2839	8021	B_SW_DATA:	DS 1	
2840	8022	B_SW_COMM1:	DS 1	
2841	8023	B_SW_COMM2:	DS 1	
2842	8024	B_SW_COMM3:	DS 1	
2843	8025	B_SW_COMM4:	DS 1	
2844	8026	B_SW_DECO:	DS 1	
2845	8027	B2_SW_AD:	DS 1	
2846	8028	B2_SW_DA	DS 1	
2847	8029	B2_COMM:	DS 1	
2848	802A	CHECK:	DS 1	
2849	802B	MODE_SW:	DS 1	
2850	802C	MODE_K:	DS 1	
2851	802D	B_K:	DS 1	
2852	802E	SYSFLX:	DS 1	;FUNCTION OF BIT
2853				; 0 = INSTRUCTION OF ALU
2854				; 1 = WRITE DATA
2855				; 2 = DATA VARY
2856				; 3 = WRITE ACCUMULATER
2857				; 4 = STEP CHECK
2858				; 5 = INSTRUCTION 2 BYTE
2859				; 6 = IFF "0" =DI,'1'=EI
2860				; 7 = INSTRUCTION END
2861	802F	SYS_STP2:	DS 1	
2862				; 0 = INST 2 BYTE='1'
2863	8030	SYS_NMI:	DS 1	
2864	8031	STP2_COM:	DS 1	;SAV COMD OF
OPERATING				
2865				;SPEED 2 (00-FF COMMAND)
2866	8032	R_BUF:	DS B_SW_ADDR-B_7S	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2867	8052	B_SEL_SHO:	DS 1
2868	8053	B_SPEED_INT:	DS 1
2869	8054	B_SPEED_RUN:	DS 1
2870	8055	B_COMM1:	DS 1
2871	8056	B_COMM2:	DS 1
2872	8057	B_COMM3:	DS 1
2873	8058	B_COMM4:	DS 1
2874	8059	B_ON_SONG:	DS 1
2875	805A	B_REBOOT:	DS 1



บรรณานุกรม

1. กฤษดา วิชาชีรานนท์ , ยืนภู่วรรณ “ไมโครโปรเซสเซอร์”

กรุงเทพมหานคร สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น) ,2534

2. ยืน ภู่วรรณ , วัฒนา เชียงกุล “ไมโครโปรเซสเซอร์ไมโครคอมพิวเตอร์”

กรุงเทพมหานคร บริษัท ซีเอ็ดดูเคชั่น จำกัด ,2532

3. ชม กิมปาน “ดิจิทัล อิเล็กทรอนิกส์และการออกแบบ”

กรุงเทพมหานคร บริษัท อิเล็กทรอนิกส์เวิลด์ จำกัด

