นาฬิกาโปรแกรมเวลา

PROGRAMMABLE CLOCK CONTROLLER

โดย

นายณรงค์ฤทธิ์ ดีรีโชติ

จ.อ.วิเชียร ปานเที่ยง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชาเทคนิคอุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2534

นาฬิกาโปรแกรมเวลา

PROGRAMMABLE CLOCK CONTROLLER

แรงค์ฤทธิ์ คีรีโชติ

วิเชียร ปานเที่ยง

ได้รับการพิจารณาอนุมัติให้นับเป็นส่วนหนึ่งของการศึกษา

ตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต

สาขาวิชา เทคโนโลยีอิเลคทรอนิคส์

คณะกรรมการตรวจสอบปริญญานิพนธ์

.....ประธานกรรมการ

( )

.....กรรมการ

( )

.....กรรมการ

( )

วันที่.....เดือน.....พ.ศ. .....

ชื่อเรื่อง ปริญญานิพนธ์	นาฬิกาโปรแกรม (PROGRAMMABLE CLOCK CONTROLLER)
ชื่อผู้เขียน	1. นาย ณรงค์ฤทธิ์ คีรีโชติ 2. จ.อ.วิเชียร ปานเที่ยง
ปริญญานิพนธ์	อุตสาหกรรมศาสตร์บัณฑิต สาขาเทคโนโลยีอิเลคทรอนิคส์ สถาบันเทคโนโลยีพระจอมเกล้าวิทยาเขต เจ้าคุณทหาร ลาดกระบัง

บทคัดย่อ

นาฬิกาโปรแกรมเวลาที่ได้จัดทำขึ้นนี้จะใช้หลักการโดยให้ IC เบอร์ MM58167 ซึ่งเป็น CHIP IC 24 ขา ซึ่งทำหน้าที่ในการบอกเวลา วัน, เดือน, ปี และเป็นการ ขัดจังหวะ (INTERUPT) ให้แก่ไมโครโปรเซสเซอร์เบอร์ Z80 เพื่อประมวลผลของเวลาไปทำการปิดเปิด CHANNALต่างๆซึ่งมีทั้งหมด 8 CHANNALในการใช้งาน Z80 นั้นในวิทยานิพนธ์เล่มนี้ผู้เขียนไม่ได้กล่าวถึงรายละเอียดมากนักเพราะทฤษฎีต่างๆของZ80 มีแพร่หลายผู้ต้องการศึกษาเพิ่มเติมสามารถค้นคว้าได้ง่าย

การใช้งานของเครื่องถูกออกแบบมาให้ใช้งานที่ง่ายเพราะใช้KEYBORAD 8 ตัว และแต่ละ MODE จะทำงานโดยใช้ KEYBORAD ตัวเดียวกันซึ่งสามารถใช้งานได้ง่าย

ประโยชน์ของเครื่องนี้สามารถเก็บโปรแกรมการปิดเปิดได้ 25 ครั้งต่อ 1 ช่อง เพราะฉะนั้นถ้า 8 ช่องก็จะเก็บได้ 200 โปรแกรมต่อ 1วันเพราะฉะนั้นการทำงานก็จะทำงานในรอบสัปดาห์คือ 7 วัน แต่การ PROGRAM นั้นสามารถ PROGRAM ได้ที่ละช่อง โดยมีหลอด LED แสดงวันและสถานะของแต่ละ CHANNAL ว่า ON หรือ OFF ซึ่งผู้ใช้สามารถมองเห็นได้ง่ายและเข้าใจวิธีการ PROGRAM ได้รวดเร็ว

การใช้งานเพียงแค่ต่อ O/P CHANNAL ซึ่งมีลักษณะเป็นขา NO และ C ของ RELAY ซึ่งมีลักษณะเป็น SWITCH ดังนั้นจึงสามารถนำไปประยุกต์การ ON อุปกรณ์ไฟฟ้าต่างๆได้มากมาย แล้วแต่ผู้ใช้จะนำไปประยุกต์

นาฬิกาโปรแกรมเวลาเครื่องนี้มีข้อดีคือ สามารถโปรแกรมปิดเปิด มีความละเอียดเป็นวินาทีดังนั้นจึงสามารถควบคุมงานที่ละเอียดและมีความเที่ยงตรงแน่นอน เพราะใช้ CHIP IC เบอร์ MM58167 ซึ่งเป็นIC REAL TIME CLOCK ของ NATIONAL SEMICONDUCTOR

Mr. NARONGRIT KERECHOT 33.131206

Mr. VICTHINT PANTHIT 33.131221

Mr. ---------- ADVISOR

(----------)

ABSTRACT

This project programmable clock controller is consis, z-80cpu main board and real time clock MM58167 interface ,the real time clock are interupt cpu with second ,It is control port 8255 operates flower program to output port next signal from port are control output relay 8 ch on-off

The project is use display time mode,program mode list mode ,eras mode and display day ,channel status

กิตติกรรมประกาศ

ในโลกที่กำลังเจริญรุดหน้าอยู่ทุกวันนี้ โลกแห่งคอมพิวเตอร์ก็เจริญก้าวหน้าไปอย่างรวดเร็ว คอมพิวเตอร์ได้เข้ามามีบทบาทต่อการพัฒนาทางเทคโนโลยี ซึ่งคอมพิวเตอร์ก็จะต้องใช้หลักการของการมีฐานเวลาที่แน่นอนถูกต้องซึ่งจะทำงานได้แม่นยำ

การทำงานประจำวันอาศัยเวลาเป็นตัววัดสิ่งต่างๆได้มากมายอีกทั้งเป็นตัวกำหนดสิ่งที่จะเกิดขึ้นในอนาคตได้ การทำงานที่ซ้ำๆกันเป็นประจำเราสามารถที่จะใช้ให้คอมพิวเตอร์ทำได้โดยการบอกเวลาที่ถูกต้องให้จากนั้นคอมพิวเตอร์ก็จะทำงานตามโปรแกรมที่เราจัดทำขึ้น

จากการสร้างเครื่องนาฬิกาโปรแกรมเวลาเพื่อใช้ประโยชน์ในการบอกเวลาและศึกษาการทำงานของ IC เบอร์ MM58167 ร่วมกับ Z80 CPU อีกทั้งยังให้ประโยชน์ในการปิดเปิดอุปกรณ์ต่างๆตามเวลาที่โปรแกรมไว้ถึง 8 ช่อง(ควบคุมอุปกรณ์ได้ 8 อย่าง)ในเวลาเดียวกันโดยแต่ละช่องสามารถทำงานได้เป็นอิสระต่อกัน

จากการที่ได้ทดลองและวิจัย ผลที่ออกมาได้รับความสำเร็จดังที่ต้องการซึ่งผลความสำเร็จเหล่านี้ได้รับความช่วยเหลือจากท่านอาจารย์ สมภพ แก้วมีชัย ซึ่งช่วยให้คำแนะนำและให้คำปรึกษาเป็นอย่างดีตลอดมา ซึ่งเป็นประโยชน์อย่างมากในการทำงาน จึงขอขอบพระคุณท่านอาจารย์มาใน ณ ที่นี้

ณรงค์ฤทธิ์ คีรีโชติ

วิเชียร ปานเที่ยง

สารบัญ

บทที่ 1

บทนำ

บทนำ

ในปัจจุบันจะเห็นว่ามีการใช้เวลากันอย่างแพร่หลายในการที่จะทำสิ่งต่างๆให้สำเร็จและเป็นไปตามเวลาที่กำหนด ซึ่งเป็นเหตุหนึ่งที่ทำให้เกิดปริญญานิพนธ์ฉบับนี้ซึ่งปริญญานิพนธ์นี้จะกล่าวถึงหลักการนำ IC REAL TIME CLOCK บอกเวลาให้แก่ไมโครโปรเซสเซอร์ โดยใช้ Z-80 เป็นหัวใจหลัก โดยที่ RTC จะเป็นตัวคอย interupt Z-80 ทุกๆ 1 วินาที และ เปรียบเทียบกับเวลาที่ตั้งในโปรแกรม จากนั้นก็นำเอาเวลาที่เปรียบเทียบแล้วตรงกับหน่วยความจำทำการประมวลผลเพื่อปิดเปิด channel แต่ละ channel ตามโปรแกรมที่ผู้ใช้ตั้งไว้

เครื่องนาฬิกาโปรแกรมเวลาเครื่องนี้ สร้างขึ้นจากแนวความคิดในการทำงานที่ซ้ำซากเป็นประจำวันหรือสัปดาห์ ดังนั้นถ้าเราใช้ส่วนที่มีค่าเวลาที่เที่ยงตรงแน่นอนไปควบคุมการทำงาน การทำงานในสิ่งต่างๆก็จะได้ประสิทธิภาพที่ดีมากขึ้น

เครื่องนาฬิกาโปรแกรมเวลาเครื่องนี้มีข้อดีที่เห็นเด่นชัดคือ การปิดเปิด channel แต่ละ channel สามารถใช้งานได้เป็นอิสระแก่กัน

จุดประสงค์ของปริญญานิพนธ์

ในการทำ Project Programmable clock controller สิ่งที่สำคัญคือการศึกษาการทำงานของ ic real time clock MM58167 ทำงานร่วมกกับ z-80โดยจะถูก interupt ตามค่าที่กำหนดใน program RTC เบอร์นี้มมีข้อดีคือสามารถต่อกับz-80 ได้ ทันที คือมีขา data 8 เส้น และ adress (A0-A4) เพื่อทำหน้าที่ในการเลือกค่า adress ภายใน MM 58167 เพื่อที่เข้าไปสู่ mode การทำงานต่างๆโดยที่ภายใน RTC จะมีหน่วยความจำ อยู่ภายใน 56 bit เพื่อทำการเก็บค่าข้อมูลเมื่อเวลาไฟดับ เช่นสามารถเก็บค่า วัน เดือน ปี เวลา ได้

จากหลักการอันนี้ เราจึงสามารถต่อ RTC ประยุกต์ใช้งานกับ z-80 ได้โดยการเพิ่มส่วนของ program ปิดเปิด channal ตามที่เราต้องการ โดยการใช้ program เป็นตัวเปรียบเทียบเวลา ในการปิดเปิดตามที่เราต้องการ

จุดประสงค์หลัก

1. เพื่อเข้าใจการทำงาน RTC 58167 และการใช้งาน z-80

2. เพื่อศึกษา Program การ interupt RTC

3. เพื่อเข้าใจการออกแบบลายวงจร

4. การประยุกต์นำไปใช้งาน

บทที่ 2

แนวความคิดและการออกแบบ

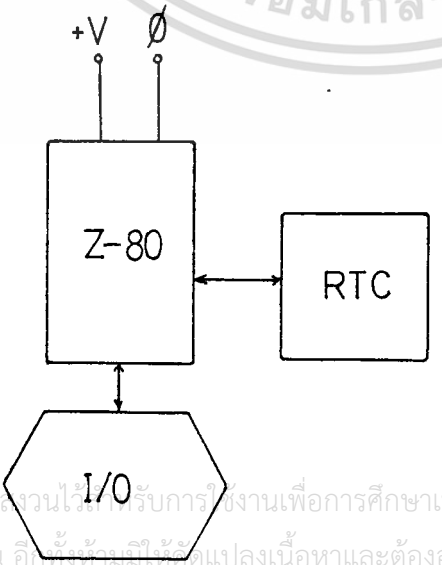
การใช้งาน Z-80 CPU

ไมโครโปรเซสเซอร์เบอร์ Z-80 เป็นไมโครโปรเซสเซอร์ขนาด 8 บิต ที่ได้พัฒนามาจากไมโครโปรเซสเซอร์เบอร์ 8080 ของบริษัทอินเทล โดยได้ทำการแก้ไขข้อบกพร่องบางอย่างของไมโครโปรเซสเซอร์เบอร์ 8080 เช่นทำให้มีคำสั่งใหม่มากขึ้นมีวิธีการเข้าถึงหน่วยความจำ (Adressing mode) ใหม่และมีระบบฮาร์ดแวร์ที่มีความสามารถ และมีความสะดวกในการใช้งานมากยิ่งขึ้น นอกจากนั้น CPU Z-80 ยังสามารถใช้ Software ของระบบที่ใช้ CPU เบอร์ 8080 ได้อีกด้วย

ในการทำนาฬิกาโปรแกรมเวลาเปิดปิดนั้น ได้นำเอา IC เบอร์ CPU Z-80 มาเป็น CPU ของเครื่องเพราะว่าง่ายต่อการใช้งานและออกแบบอีกทั้ง IC RTC เบอร์ MM58167 ก็มีความพร้อมในด้านฮาร์ดแวร์ คือมีลักษณะ DATA ที่มีขนาดเท่ากันคือ 8 Bit อีกทั้งยังง่ายต่อการเรียกค่าต่าง ๆ เช่นค่า เวลา, วัน, เดือน, ปี ที่ถูกต้อง

ดังนั้นในการเขียนวิทยานิพนธ์ชุดนี้ จึงเน้นหลักของการปฏิบัติและการสร้าง อีกทั้งการ program เวลา ON-OFF พร้อมทั้ง Mode การตั้งเวลา ส่วนทฤษฎีเนื้อหาของ Z-80 นั้น ผู้อ่านสามารถค้นคว้าเพิ่มเติมได้จากหนังสือทั่วไป

อีกอย่างหนึ่งที่จะลืมไม่ได้ยังคงไว้ซึ่งลักษณะของขารูปแบบการใช้งานเบื้องต้น เพื่อเผื่อไว้สำหรับผู้ที่ไม่มีเวลาค้นคว้า ก็จะอ่านเป็นสังเขปได้ ดังนั้นเราจึงกล่าวแค่สิ่งที่มีส่วนสำคัญในการอ้างอิงถึงทั่วไปเท่านั้น

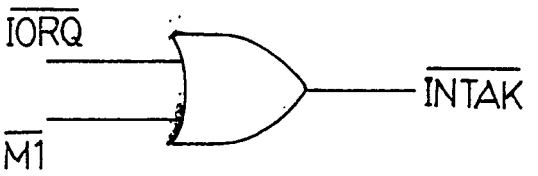


กลุ่มสัญญาณต่างๆของไมโครโปรเซสเซอร์ Z-80 แบ่งออกได้เป็น 3 กลุ่มคือ กลุ่มของสายสัญญาณเพื่อกำหนดตำแหน่ง (ADDRESS BUS) คือ A15-A0 กลุ่มสายสัญญาณส่งข้อมูล (DATA BUS) D7-D0 และกลุ่มของสายสัญญาณควบคุม (CONTROL BUS) คือสายสัญญาณทั้งหมดที่เหลือยกเว้นขาแหล่งจ่ายไฟและสัญญาณนาฬิกา หน้าที่ของขาต่างๆมีดังนี้

A15-A0 เป็นสายสัญญาณกำหนดตำแหน่ง (ADDRESS BUS) โดยที่ A0 เป็นบิทต่ำด้านต่ำขาเหล่านี้เป็นเอาท์พุทแบบสามสถานะและจะให้แอคตีฟที่ลอจิก "1" บัสนี้มีด้วยกันทั้งหมด 16 สาย ดังนั้นจึงสามารถติดต่อกับหน่วยความจำได้ถึง 65536 ตำแหน่ง (64K) นอกจากนี้ยังสามารถใช้ในการกำหนดตำแหน่งของพอร์ต อินพุท/เอาท์พุท เมื่อใช้คำสั่งกลุ่มอินพุท/เอาท์พุทได้ โดยใช้ 8 บิทด้านต่ำ เพื่อแสดงตำแหน่งของพอร์ต ดังนั้นจึงสามารถกำหนดพอร์ตอินพุทได้ 256 พอร์ท หรือกำหนดพอร์ตเอาท์พุทได้ถึง 256 พอร์ทเช่นกัน

D7-D0 เป็นสายสัญญาณข้อมูล (DATA BUS) D0 เป็นบิตทางด้านต่ำลักษณะเป็นบัสสองทิศทางแบบสามสถานะ ขนาด 8 บิท และแอคตีฟที่ลอจิก "1" ใช้เป็นเส้นทางเพื่อผ่านของข้อมูลระหว่างไมโครโปรเซสเซอร์กับหน่วยความจำหรืออุปกรณ์อินพุท/เอาท์พุทต่างๆ

$\overline{M1}$ (MACHINE CYCLE ONE) เป็นขาเอาท์พุทแอคตีฟที่ "0" เมื่อขานี้แอคตีฟ ชี้ให้เห็นว่าขณะนี้กำลังอยู่ในโปรแกรมของการเฟทช์คำสั่ง และถ้าเป็นคำสั่งที่มีรหัส 2 ไบต์ ส่วนของ $\overline{M1}$ จะถูกสร้างขึ้นขณะเฟทช์ในแต่ละไบต์ ลักษณะของคำสั่งที่มีขนาด 2 ไบต์ เช่นคำสั่งที่มีรหัสที่เริ่มต้นด้วย CBH,DDH,EDH,FDH นอกจากนั้นสัญญาณ $\overline{M1}$ นี้จะใช้ร่วมกับ $\overline{IORQ}$ เพื่อสร้างสัญญาณตอบรับการอินเตอร์รัพท์ INTERUPT ACKNOWLAGE) โดยใช้วงจร logic ดังรูป 1-2



$\overline{MREQ}$ (Memory Request) เป็น output แบบสามสถานะและ active ที่ logic "0" เมื่อสายสัญญาณนี้ active บอกให้ทราบว่าขณะนี้ไมโครโปรเซสเซอร์ต้องการติดต่อกับหน่วยความจำเพื่ออ่านหรือเขียนข้อมูลโดยที่ตำแหน่งของหน่วยความจำจะปรากฏอยู่บนบัสตำแหน่งแล้ว

$\overline{IORQ}$ (Input/output Request) เป็น output แบบสามสถานะและ active ที่ logic "0" เมื่อสายสัญญาณนี้ active บอกให้ทราบว่าขณะนี้ทางด้าน byte ต่ำ (A7-A0) ของบัสตำแหน่งบรรจุตำแหน่งของ port ที่จะส่งถ่ายข้อมูลระหว่างไมโครโปรเซสเซอร์กับอุปกรณ์อินพุท/เอาท์พุท นอกจากนี้จะใช้ร่วมกับสัญญาณ $\overline{M1}$ เพื่อตอบรับการ interupt ดังรูป 1-2 และขณะนี้ vector ของการ interupt จะส่งผ่านเข้ามาในบัสข้อมูลเพื่อกำหนดตำแหน่งของโปรแกรมบริการการ interupt

$\overline{RD}$ (Memorey Read) เป็นขา output แบบสามสถานะและ active ที่ logic "0" สัญญาณนี้เพื่อชี้ว่าขณะนี้ไมโครโปรเซสเซอร์ต้องการอ่านข้อมูลจากหน่วยความจำหรือจากอุปกรณ์อินพุท/เอาท์พุท

$\overline{WR}$ (Memomry Write) เป็นขา output แบบสามสถานะและ active ที่ logic "0" เมื่อสัญญาณนี้ active ชี้ว่าขณะนี้ไมโครโปรเซสเซอร์ต้องการเขียนข้อมูลเข้าสู่หน่วยความจำหรือเข้าอุปกรณ์อินพุท/เอาท์พุท

$\overline{REFSH}$ (Refresh) เป็นขา output active ที่ logic "0" เป็นสัญญาณเพื่อชี้ว่าขณะนี้บัสตำแหน่งทางด้านต่ำ 7 bit (A6-A0) บรรจุค่าหน่วยความจำ dynamic RAM ที่จะ refresh และสัญญาณ $\overline{MREQ}$ ในช่วงนี้จะนำไปใช้เป็นสัญญาณสำหรับอ่านเพื่อ refresh dynamic RAM ทั้งหมดที่ใช้ระบบ

$\overline{HALT}$ (Halt State) เป็นขา output active ที่ logic "0" เป็นสัญญาณเพื่อชี้ว่าขณะนี้ไมโครโปรเซสเซอร์ปฏิบัติคำสั่ง Halt จากโปรแกรม และกำลังรอสัญญาณการ interupt ชนิด Nonmaskable และ Maskable จากอุปกรณ์ภายนอก ถ้าได้รับสัญญาณการ interupt แล้วจึงจะทำงานต่อไปได้ ในขณะที่หยุดรอ (Halted) นี้ CPU จะทำคำสั่ง NOP เพื่อให้มีการ fet คำสั่ง ซึ่งจะไม่ทำให้การ refresh หยุดชงักลง

$\overline{INT}$ (Interupt request) เป็นขา input active ที่ logic "0" สัญญาณ int นี้เป็นสัญญาณที่สร้างมาจากอุปกรณ์อินพุท/เอาท์พุท เพื่อต้องการ interupt การทำงานตามปกติของไมโครโปรเซสเซอร์ สัญญาณร้องขอนี้จะถูกตรวจสอบเมื่อถึง state สุดท้ายของคำสั่งและไมโครโปรเซสเซอร์จะจดจำไว้ ถ้าหากว่าโปรแกรมกำหนดให้มีการยอมรับสัญญาณการ interupt ได้ (Enable interupt) โดย IFF1 ถูก set เป็น "1" และไม่มีการขอใช้บัสเสียก่อน คือขา bus request ต้องไม่ active เมื่อไมโครโปรเซสเซอร์รับสัญญาณ interupt มันจะตอบสนองโดยการส่งสัญญาณ $\overline{IORQ}$ ออกมาในช่วงเวลา $\overline{M1}$ เพื่อเป็นการตอบรับการ interupt ในช่วง cycle ของคำสั่งต่อมา ส่วนรายละเอียดของการขัดจังหวะจะอธิบายในเรื่องของการ interupt

$\overline{NMI}$ (Nonmaskable interupt) เป็นขา input และ active ที่ขอบ pulse ขาลง (Negative edge trigger) สัญญาณที่ขา $\overline{NMI}$ นี้มีลำดับความสำคัญสูงกว่าสัญญาณที่ขา $\overline{INT}$ ไมโครโปรเซสเซอร์จะทำการตรวจสอบขานี้ที่ state สุดท้ายของคำสั่ง เช่น เดียวกับค่า $\overline{INT}$ แต่จะไม่ขึ้นอยู่กับ IFF เมื่อไมโครโปรเซสเซอร์ได้รับสัญญาณที่ขา $\overline{NMI}$ จะทำให้เริ่มต้นการทำงานใหม่ที่ตำแหน่ง 0066H ส่วนค่าในโปรแกรม counter ที่ชี้ตำแหน่งของคำสั่งต่อไปก่อนที่ CPU จะถูก interupt จะเก็บไว้ใน stack (ที่ RAM) เพื่อที่ CPU สามารถกลับมาทำงานต่อได้หลังจากที่ทำโปรแกรมบริการการ interupt เสร็จสิ้นแล้วในขณะที่ CPU อยู่ในจังหวะ $\overline{WAIT}$ มันจะไม่รับสัญญาณ $\overline{NMI}$ นี้ สัญญาณ $\overline{NMI}$ มีลำดับความสำคัญกว่าสัญญาณ $\overline{BUSRQ}$ ดังนั้นในขณะที่ CPU กำลังทำโปรแกรมบริการการ Interupt อยู่มันสามารถรับสัญญาณ $\overline{BUSRQ}$ ได้

$\overline{\text{RESET}}$ เป็นอินพุท active เมื่อ logic "0" เมื่อไมโครโปรเซสเซอร์ได้รับสัญญาณ $\overline{\text{RESET}}$ จะทำให้ค่าในโปรแกรม counter เริ่มต้นที่ 0 และตั้งต้นการทำงานของไมโครโปรเซสเซอร์ใหม่และในส่วนอื่นๆจะเป็นดังนี้

1. จัด interupt flip flop (IFF) ให้อยู่ในสภาพที่ไม่ยอมรับการ interupt แบบ maskable (IFF1 = IFF2 =0)
2. set register I = 00H
3. register R = 00H
4. set ให้เป็นการ interupt mode 0

ในช่วงเวลาของการ reset บัสข้อมูล บัสตำแหน่ง จะอยู่ในสภาวะ impedance สูง ส่วนบัสควบคุมจะอยู่ในสภาพที่ไม่ active

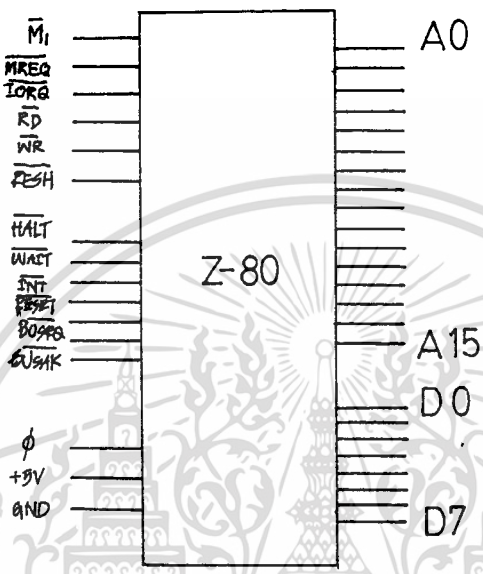
$\overline{\text{BUSRQ}}$ (bus request) เป็นขาอินพุท active ที่ logic "0" สัญญาณ $\overline{\text{BUSRQ}}$ นี้มีผลทำให้บัสตำแหน่ง บัสข้อมูล และสัญญาณควบคุมที่เป็นขา output แบบสามสถานะอยู่ในสภาวะ impedance สูง จากนั้นบัสต่างๆ จะถูกควบคุมโดยอุปกรณ์ภายนอก ไมโครโปรเซสเซอร์จะตรวจสอบสัญญาณการขอใช้บัสนี้ทุกๆ state สุดท้ายของทุกๆ machine cycle ของคำสั่ง และเมื่อพบการขอใช้บัส CPU จะตอบสนองใน cycle ถัดไป

$\overline{\text{BUSAK}}$ (Bus acknowlage) เป็นขา output active ที่ระดับ "0" สัญญาณนี้ใช้สำหรับการตอบรับการขอใช้บัส และแสดงว่าบัสตำแหน่ง บัสข้อมูลและสัญญาณควบคุมที่เป็น
output แบบสามสถานะอยู่ในสภาวะ Impedance สูงแล้วอุปปกรณ์ควบคุมภายนอกสามารถเข้ามาควบคุมบัสได้

ϕ เป็นขาที่รับสัญญาณนาฬิกาซึ่งเป็นเพียงเฟสเดียว ใช้ระดับสัญญาณแบบ TTL และต้องการตัวต้านทานเพื่อ pull up ค่า 330 ohm 1ตัวเพื่อต่อกับแหล่งจ่ายไฟ 5V Z-80 ทำงานได้ที่สัญญาณนาฬิกาไม่เกิน 2.5 MHz Z-80A ทำงานได้ไม่เกิน 4MHz และ Z-80B ทำงานได้ไม่เกิน 6 MHz

ลักษณะขาและการเชื่อมต่อ

ไมโครโปรเซสเซอร์ Z-80 บรรจุอยู่ใน IC ขนาดมาตรฐานอุตสาหกรรม (Industry standard) แบบ Dual In-Line Package (DIP) ซึ่งมีลักษณะขาดังนี้



กลุ่มสัญญาณต่าง ๆ ของไมโครโปรเซสเซอร์ตัวนี้แบ่งออกเป็น 3 กลุ่ม

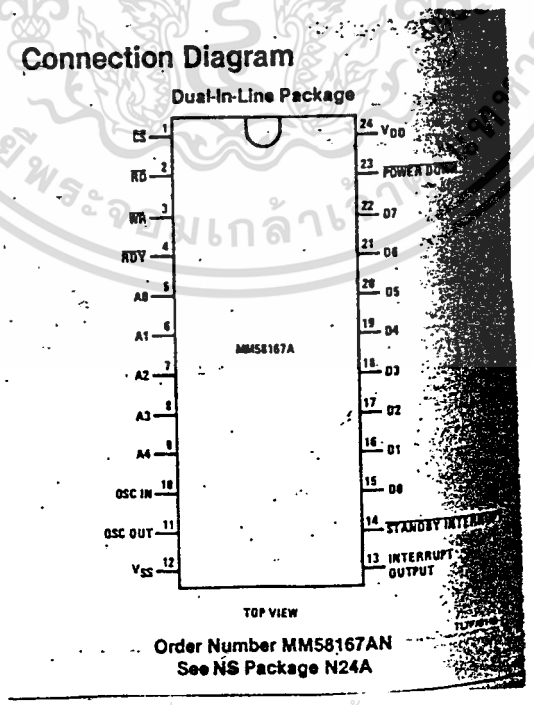
1. กลุ่มสายสัญญาณเพื่อกำหนดตำแหน่ง (Adress Bus) คือ A_{15}-A_0
2. กลุ่มสายสัญญาณส่งข้อมูล (Data Bus) คือ D_7-D_0
3. กลุ่มของสายสัญญาณควบคุม (Control Bus) คือ สายสัญญาณทั้งหมดที่เหลือยกเว้นขาแหล่งจ่ายไฟและสัญญาณนาฬิกา

ลักษณะทางด้านฮาร์ดแวร์และการออกแบบ

จากแนวความคิดที่ว่า โดยทั่วไปนาฬิกาสามารถบอกเวลาให้เราทราบได้ และนาฬิกาในทางอิเล็คทรอนิคส์นั้นก็สามารถบอกเวลาให้กับอุปกรณ์อิเล็คทรอนิคส์ให้ทราบเวลาของตัวมันได้ โดยอาศัยการนับ (counter) ส่วนของฐานเวลาซึ่งมีลักษณะที่เที่ยงตรงแน่นอน ดังนั้นถ้าเรามีไมโครโปรเซสเซอร์อยู่ในวงจร เพราะฉะนั้นไมโครโปรเซสเซอร์ก็สามารถที่จะรู้

เวลาได้ถ้าเรามีการบอกเวลาให้มันรู้ ดังนั้นแนวความคิดต่อไปก็คือว่า เมื่อ CPU รู้เวลาของตัวเองแล้วเราจะทำอย่างไรต่อไปเพื่อที่จะนำเอาค่าเวลาที่รู้ไปใช้ประโยชน์ในชีวิตประจำวัน ดังนั้นหนทางหนึ่งที่จะใช้ประโยชน์ได้ก็คือการปิดเปิดอุปกรณ์ไฟฟ้าตามเวลาที่ต้องการโดยมีความถูกต้องแม่นยำสูง อีกทั้งสามารถตั้งโปรแกรมล่วงหน้าได้ด้วย ในการปิดหรือเปิดก็จะทำตามรอบของสัปดาห์ได้อย่างถูกต้อง และความละเอียดในการปิดเปิดเป็นวินาที เช่นต้องการเปิดแอร์เวลา 8:00:00 และปิดเวลา 10:00:00 ดังนั้นเมื่ออัดโปรแกรมเหล่านี้โดยผ่านทาง Key board ทางด้านหน้าปัทม์ของเครื่องเราก็สามารถใช้ประโยชน์นาฬิกาโปรแกรมเวลาได้ ส่วนรายละเอียดทางด้านฮาร์ดแวร์ program จะกล่าวต่อไปในบทของ program และการใช้งาน

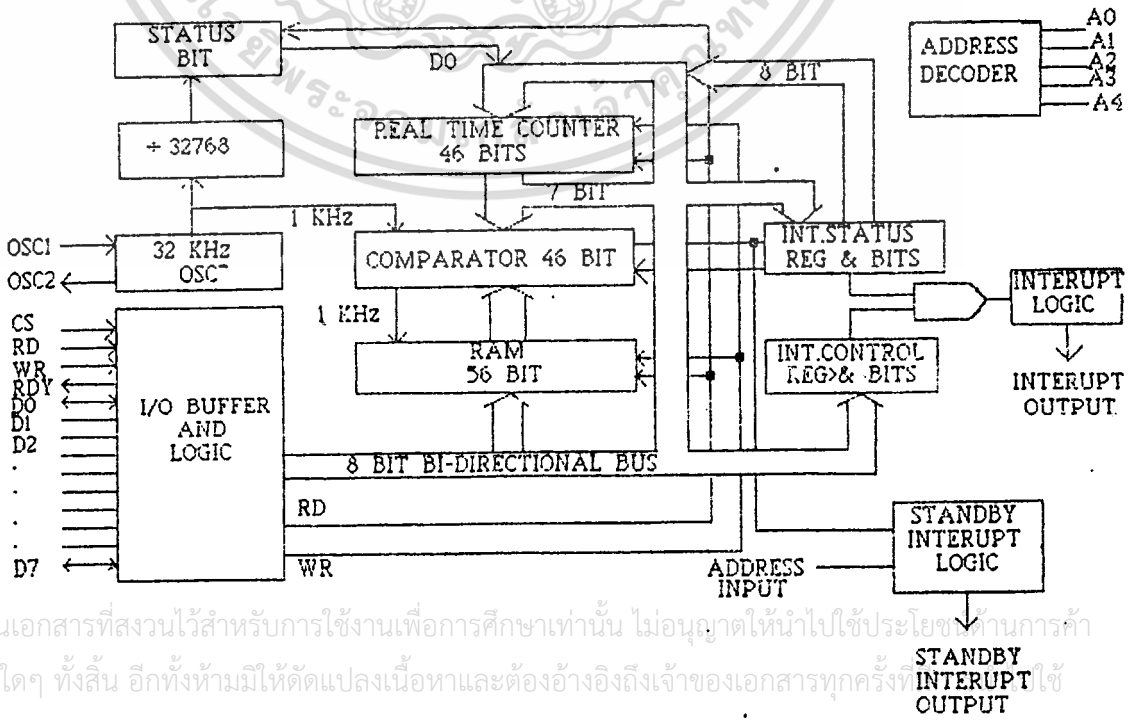
RTC (Real Time Clock) เป็นส่วนที่ทำให้เกิดฐานเวลาที่เที่ยงตรงและมี counter นับฐานเวลาอยู่ภายในที่สามารถโปรแกรมค่าการนับได้ ซึ่งทำให้สะดวกอีกทั้งมี DATA 8 BIT และมีการตั้งค่า MODE ต่าง ๆ ได้มากมาย ซึ่งดูรายละเอียดได้จากบทของเรื่อง RTC (MM 58167)



RTC (REAL TIME CLOCK)

RTC คือนาฬิกาบอกเวลาให้กับไมโครโปรเซอร์ (เวลาที่แท้จริงที่ใช้อยู่) โดยการเชื่อมต่อกับ cpu ก็สามารถทำให้ cpu สามารถรู้เวลาได้ทุกขณะโดยที่ cpu สามารถอ่านค่าเวลาจากตัว RTC ได้ทุกครั้งที่ cpu ต้องการจึงทำให้ไปทำงานอื่นๆได้ RTCเบอร์นี้กินกระแสน้อยมาก เมื่อเราต่อ battaty สำรองให้กับมันจะทำให้ใช้ RTC ทำงานอยู่ได้นานเป็นเดือน แต่ก่อนที่จะทำให้ RTC ทำงานเราจะต้องให้ cpu เขียนข้อมูลเกี่ยวกับเวลาจริงให้กับมัน เสียก่อน เมื่อเรียบร้อยแล้ว เราเลิกจ่ายไฟเลี้ยง RTC ก็ยังเป็นนาฬิกาเหมือนนาฬิกาทั่วไป (แต่ต้องต่อ battary สำรองด้วย) RTC# 58167A สามารถบอกเวลา ได้ตั้งแต่ วินาที,นาที,ชั่วโมง,วันในรอบสัปดาห์(อาทิตย์-เสาร์), เดือน และ ปี ให้แก่ไมโครโปรเซสเซอร์ได้ดีและเที่ยงตรง โดยที่ cpu เพียงแต่ติอต่อกับ RTC เหมือนก้บติดต่อกับหน่วยความจำ หรือ port เท่านั้น RTC มีด้วยกันหลายเบอร์ เช่น MM58167A ,MM58174, MM58274 .. แต่ในที่นี้ เราจะกล่าวถึง เบอร์ MM58167A ด้วยเหตุผล ที่ว่า ต่อกับ Z80 ได้ง่าย และข้อมูล ที่เขียนเข้าไปไม่สูญหายง่าย จากการเลิกจ่ายไฟ

รูปแสดงขา IC.



จากโครงสร้างที่สำคัญที่จะกล่างถึงก็คือ REAL TIME COUNTER และ RAM

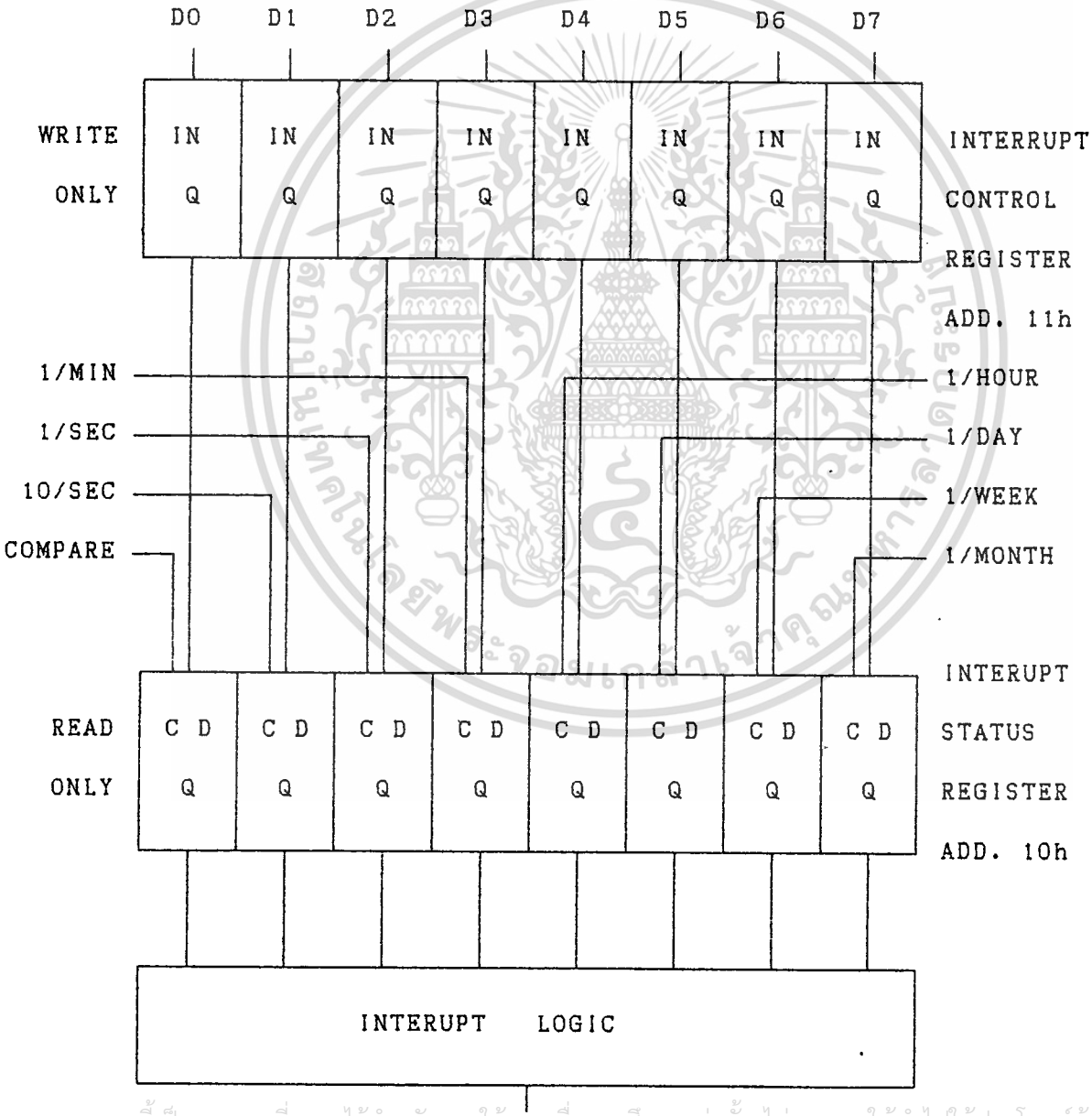
COUNTER ADDRESS	UNIT D0	D1	D2	D3	MAX BCD CODE	TENS D4	D5	D6	D7	MAX BCD CODE
1/10,000 (00H)	-	-	-	-		D4	D5	D6	D7	9
100&10 (01H)	D0	D1	D2	D3	9	D4	D5	D6	D7	9
SECOND (02H)	D0	D1	D2	D3	9	D4	D5	D6	-	5
MINUTE (03H)	D0	D1	D2	D3	9	D4	D5	D6	-	5
HOURS (04H)	D0	D1	D2	D3	9	D4	D5	-	-	2
DAY/WEEK (05H)	D0	D1	D2	-	7	-	-	-	-	0
DAY/MONTH(06H)	D0	D1	D2	D3	9	D4	D5	-	-	3
MONTH (07H)	D0	D1	D2	D3	9	D4	-	-	-	1

REAL TIME COUNTER เป็นตัวนับและจัดการเกี่ยวกับเวลาถูกแบ่งเป็นเลข BCD 2 หลัก และ BIT ไหนที่ไม่ได้ใช้จะมีค่าเป็นศูนย์เช่น DAY OF THE WEEK (วันในสัปดาห์) คือในหนึ่งสัปดาห์มีแค่ 7 วัน เพราะฉะนั้นเลขจะไม่เกินหลักหน่วยหลักสิบจึงไม่สนใจ

RAM ใช้เก็บข้อมูลเมื่อไฟตกหรือใช้เก็บข้อมูลการตั้งปลุกเพื่อที่จะนำมาเปรียบเทียบกับ REAL TIME COUNTER ซึ่ง RAM นี้ก็จะถูกจัดให้มีรูปแบบเหมือนกับ REAL TIME COUNTER ดังตราง

ตารางนี้จะมีส่วนของ ADDRESS ที่จะติดต่อด้วยอยู่ด้านซ้ายและหน้าที่ของแต่ละตำแหน่งด้านขวา เช่น เราต้องการอ่านวินาทีเข้ามาที่ CPU จากที่กล่าวมาแล้วว่าตัวกระทำการนับเวลาอยู่ที่ COUNTER ดั้งนั้นเราก็อ่าน ADDRESS 00010 COUNTER SECONDS คือ ADDRESS 02H นั่นเองก็เป็น IN A, (02H) และถ้าเราต้องการตั้งเวลาสำหรับปลุกบ้างก็เขียนเข้าไปที่ RAM เช่น กำหนดวินาทีไว้ที่ RAM ก็เป็น OUT RAM SECONDS=OUT (0AH),A

INTERRUPT และ COMPARATOR



	A4	A3	A2	A1	A0	FUNCTION
00H	0	0	0	0	0	COUNTER THOUSANDTHS OF SECOND
01H	0	0	0	0	1	COUNTER HUNDREDTHS AND TENTHS OF SEC.
02H	0	0	0	1	0	COUNTER SECONDS
03H	0	0	0	1	1	COUNTER MINUTES
04H	0	0	1	0	0	COUNTER HOURS
05H	0	0	1	0	1	COUNTER DAY OF WEEK
06H	0	0	1	1	0	COUNTER DAY OF MONTH
07H	0	0	1	1	1	COUNTER MONTH
08H	0	1	0	0	0	RAM THOUSANDTHS OF SECONDS
09H	0	1	0	0	1	RAM HUNDREDTHS AND TENTHS OF SECONDS
0AH	0	1	0	1	0	RAM SECONDS
0BH	0	1	0	1	1	RAM MINUTES
0CH	0	1	1	0	0	RAM HOURS
0DH	0	1	1	0	1	RAM DAY OF WEEK
0EH	0	1	1	1	0	RAM DAY OF MONTH
0FH	0	1	1	1	1	RAM MONTH
10H	1	0	0	0	0	INTERRUPT STATUS REGISTER
11H	1	0	0	0	1	INTERRUPT CONTROL REGISTER
12H	1	0	0	1	0	COUNTER RESET
13H	1	0	0	1	1	RAM RESET
14H	1	0	1	0	0	STATUS RESET
15H	1	0	1	0	1	GO COMMAND
16H	1	0	1	1	0	STANDBY INTERRUPT
17H	1	1	1	1	1	TEST MODE

หน้าที่ขาต่างๆ MM 58167.

CS CHIP SELECT เป็นขาเลือกให้ IC ทำการติดต่อกับ cpu หรือทำงานได้ active 0

RD active 0 สำหรับอ่านข้อมูลจาก RTC

WR active 0 สำหรับเขียนข้อมูลเข้าไปยัง RTC.

A0-A4 เป็นสาย address ติดต่อกับ RTC เพื่อให้ cpu เขียนหรืออ่านข้อมูลในตำแหน่งที่มีหน้าที่นั้นๆ

OSCIN, OUT ใช้ติดต่อกับ X-TAL 32.678 KHz หรือจะนำความถี่จากภายนอกป้อนเข้าที่ OSCIN ก็ได้

INT O/P เป็นขาสัญญาณใช้ในการ INTERRUPT โดยการ INTERRUPT สั่งได้ทาง software ว่าเป็นการ INTERRUPT ที่ เวลาเท่าไรสัญญาณออกเป็นHIGH

STANDBY INTERRUPT จะให้สัญญาณ low ก็ต่อเมื่อ RTC มีการเปรียบเทียบกันระหว่าง RAM กับตัวนับเวลาเมื่อเท่ากัน

D0-D7 สายข้อมูลที่จะทำการติดต่อกับ CPU

VSS ไฟลบ

ส่วนมากแล้วในกรณีที่เราจะให้ CPU แสดงผลเวลาออกมาวิธีที่ง่ายคือ ให้ RTC ส่งสัญญาณ INTERUPT มาทุกๆ 1 วินาทีแล้ว CPU ก็อ่านเวลาจาก RTC แล้ว OUT ออกไปที่ DISPLAY RTC ตัวนี้มีสัญญาณ INTERUPT 2 อย่างคือ INTERUPT OUTPUT โดยสามารถโปรแกรมการ INTERUPT ได้ 8 อย่างคือ 10Hz, 1Hz, 1นาที/ครั้ง, 1ชั่วโมง/ครั้ง, 1วัน/ครั้ง, 1สัปดาห์ /ครั้ง และเมื่อ RAM กับ REAL TIME COUNTER เกิดการเปรียบเทียบขึ้นก็จะเกิด STAND BY INTERUPT วิธีการจะให้เกิดการ INTERUPT ขึ้นอยู่กับการให้ ENABLE LOGIC 1 ไปยัง INTERUPT CONTROL REGISTOR (ADDRESS 11H) ใน BIT ตรงกับความถี่ที่เราต้องการ เช่นให้ INT ทุกๆ 1 วินาที ก็ดูรูป 1/SEC ตรงกับ BIT ไหน ในที่นี้คือ D2 ก็เป็น

```
LD  A, 04H
OUT (11H),A
```

และเรายังสามารถ SET การ INTERUPT ได้มากกว่า 1 BIT เช่นจะให้ INTERUPT ทุกๆ วินาทีและทุกๆชั่วโมงก็เป็น 14H

ส่วนการ STAND BY INTERUPT (หรือ INTERUPT เมื่อการเปรียบเทียบ 8 ADDRESS ระหว่าง COUNTER กับ RAM ต้องเท่ากัน) ต้องให้ BIT 1 เป็น 1 ไปที่ ADDRESS 16H

ส่วนการให้ INTERUPT DISABLE ก็ส่ง 0 ไปที่ BIT นั้น

INTERUPT STATUS REGISTOR ใช้สำหรับอ่านสัญญาณ INTERUPT ว่ามาจาก BIT ใดและจะเป็นการ DISABLE การ INTERUPT อีกด้วย

COUNTER AND RAM RESET โดยการเขียน OFFH ไปที่ ADDRESS 12H และ 13H เมื่อ COUNTER ถูก RESET ส่วนที่เปลี่ยนแปลงที่ใช้เวลาน้อยจะเป็น 0 ซึ่งขณะส่วนวันของสัปดาห์วันของเดือนและเดือนจะถูกปรับเป็น 1 แต่ RAM จะเป็น 0 ทั้ง 8 ADDRESS

GO COMMAND โดยการให้ PULSE ของการเขียนไปที่ ADDRESS 15H เป็นการกำหนดให้นาทีมีความถูกต้องก่อนที่จะเริ่มนับ COUNTER หรือเริ่ม CLOCK นั่นเอง

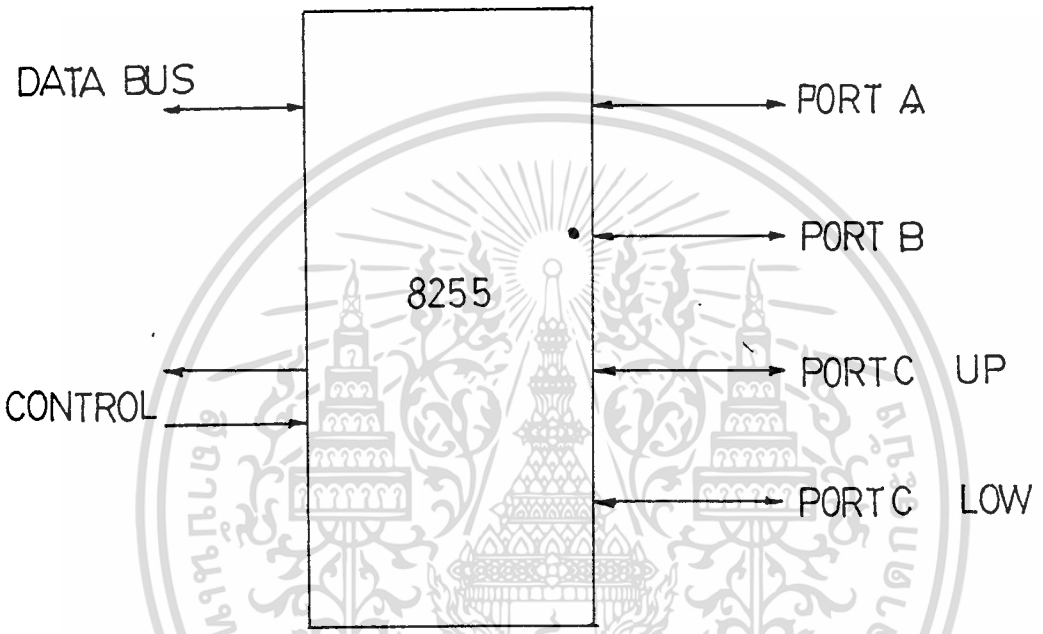
STATUS BIT

จะบอกถึงความถูกต้องของการอ่านตัวนับ COUNTER โดย LOGIC ที่อ่านได้ที่ BIT 0 จะเป็น 1 ส่วน BIT อื่นจะเป็น 0 หมด แต่ถ้าอ่านค่าจาก COUNTER มาไม่ถูกต้องที่ BIT 0 นี้จะมีค่าเป็น 0

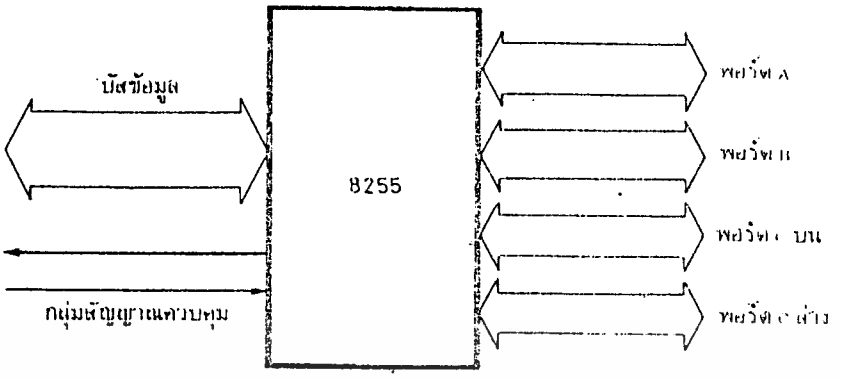
TEST MODE ใช้ทดสอบ RTC ให้ทำงานที่ความถี่สูงกว่าปกติ

การใช้งาน port 8255 เบื้องต้น

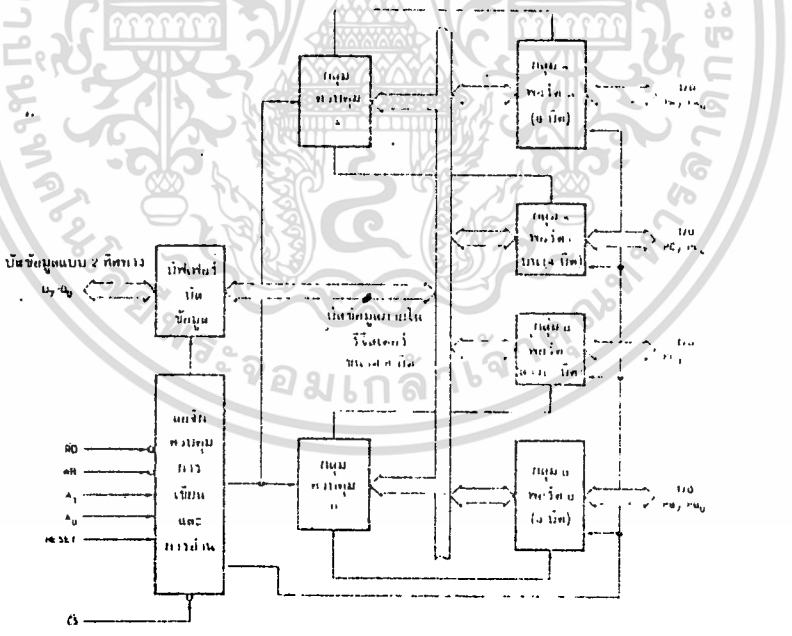
IC 8255 เป็น IC ที่มี 40 ขา ได้รับการออกแบบมาให้มีสัญญาณเพื่อเชื่อมต่อกับ 8080 แต่สัญญาณนี้พอเหมาะที่ใช้กับ Z-80 ได้ดีเช่นเดียวกัน 8255 เป็น IC ที่ต่อเป็น port ให้กับไมโครโปรเซสเซอร์ได้ 3 port โดยมีโครงสร้างพื้นฐานแสดงดังรูป



การเรียก port ของ 8255 จะเรียน port ต่าง ๆ กันว่า port A, port B, และ port C โดย port C แยกเป็น 2 ส่วน คือ port C ล่าง หรือตั้งแต่ Pc_0-Pc_3 มีจำนวน 4 บิต และ port C บน หรือตั้งแต่ Pc_4-Pc_7 ที่พิเศษคือ port ทุก port เป็นได้ทั้ง port อินพุทและเอาต์พุท



รูปที่ 7.1 แผนผังโครงสร้างของไอซี 8255



รูปแสดงภายในของการจัดขา IC 8255

จากรูปการจัดขาภายใน IC เบอร์ 8255 การทำงานของวงจรจะใช้สัญญาณควบคุมจากไมโครโปรเซสเซอร์มาควบคุมการทำงาน โดยไมโครโปรเซสเซอร์จะสั่งคำสั่งมาโปรแกรมการทำงานหรือกำหนดรูปแบบของ port โดยให้เป็น input หรือ output port ก็ได้

ลักษณะขาต่าง ๆ ของ 8255

เพื่อให้เข้าใจวิธีการต่อใช้งานระหว่าง Z-80 และ 8255 จึงจำเป็นต้องเข้าใจความหมายและตำแหน่งของขาต่าง ๆ เสียก่อน ขาทั้ง 40 ขา ของ IC ประกอบด้วย

D0-D7 เป็นขาที่ข้อมูลอินพุทเอาต์พุทจะต้องรับสัญญาณจากภายนอก เพื่อเลือก chip 8255 โดยเมื่อขานี้เป็น "0" จะทำให้ 8255 ต่อเข้ากับระบบบัสของไมโครโปรเซสเซอร์ เพื่อให้ไมโครโปรเซสเซอร์เขียนหรืออ่านข้อมูลจาก port ได้

$\overline{RD}$ เป็นสัญญาณการอ่าน เป็นขา Input ที่ต้องส่งมาจาก CPU เมื่อสัญญาณที่ขานี้เป็น "0" หรือสัญญาณ CS เป็น 0 ด้วย ไอซี 8255 จะทำตัวให้ CPU อ่านข้อมูลจาก BUS ในขณะที่เป็น port input

$\overline{CS}$ สัญญาณการเลือก chip ขานี้เป็นขา input ที่จะรับสัญญาณจากภายนอกเพื่อเลือก chip 8255 โดยเมื่อขานี้เป็น 0 จะทำให้ 8255 ต่อเข้ากับระบบ BUS ของไมโครโปรเซสเซอร์ เพื่อให้ไมโครโปรเซสเซอร์ เขียนหรืออ่านข้อมูลจาก port ได้

$\overline{WR}$ เป็นสัญญาณการเขียน จะ Active เมื่อสัญญาณ WR และสัญญาณ CS เป็น "0" สัญญาณนี้จะมาจาก CPU เพื่อต้องการเขียนข้อมูลลงบน port ที่กำหนด

A_0-A_1 สัญญาณ Adress logic ของสัญญาณทั้งสองจะถอดรหัสออกเป็น 4 รหัส เพื่อกำหนด Registor ภายในที่เชื่อมต่อกับ port Input Output ของ 8255

$\overline{RESET}$ สัญญาณ RESET เป็นสัญญาณที่ส่งจากภายนอกเข้ามาทำการ RESET 8255 เพื่อ clear สถานะต่าง ๆ ของ 8255 เมื่อ 8255 ได้รับการรีเซต ก็จะกลับเข้าสู่ Mode Input หรือ port ที่เป็น Input port

PA_0-PA_7 เป็นสายสัญญาณที่เป็น port ของ 8255 ที่ชื่อ port A การเลือก port จะเลือก โดยสัญญาณ Adress A_0-A_1

PB_0-PB_7 เป็นสายสัญญาณที่เป็น port B ของ 8255 ถูกเลือกโดยสัญญาณ Adress A_0-A_1

PC_0-PC_7 เป็นสายสัญญาณที่เป็น port C ของ 8255 การกำหนด port นี้จะได้รับการกำหนดโดยสัญญาณ Adress A_0-A_1, port C นี้แบ่งเป็น 2 กลุ่มคือ กลุ่ม PC_0-PC_3 และกลุ่ม PC_4-PC_7

การใช้งานในเครื่องนาฬิกา RTC โปรแกรมเวลานี้จะออกแบบให้ 8255 ใช้งานใน Mode 0 เท่านั้น เพราะเป็น Mode การใช้งานที่เข้าใจง่าย และง่ายต่อการ program ซึ่งเป็น Mode พื้นฐาน

ก่อนอื่นจะต้องมารู้จักสัญญาณควบคุมการทำงานของขาต่าง ๆ เบื้องต้นก่อน

Write - Active Logic 0

$\overline{RD}$	$\overline{WR}$	A_1	A_0	ความหมาย
1	0	0	0	เขียนเพอร์ท A ซึ่งเป็นข้อมูล
0	1	0	0	อ่านเพอร์ท A ซึ่งเป็นข้อมูล
1	0	0	1	เขียนเพอร์ท B ซึ่งเป็นข้อมูล
0	1	0	1	อ่านเพอร์ท B ซึ่งเป็นข้อมูล
1	0	1	0	เขียนเพอร์ท C ซึ่งเป็นข้อมูล
0	1	1	0	อ่านเพอร์ท C ซึ่งเป็นข้อมูล
1	0	1	1	เขียนข้อมูล ซึ่งเป็นเวิร์ดควบคุม
0	1	1	1	อ่านข้อมูล ซึ่งไม่มีความหมายใด

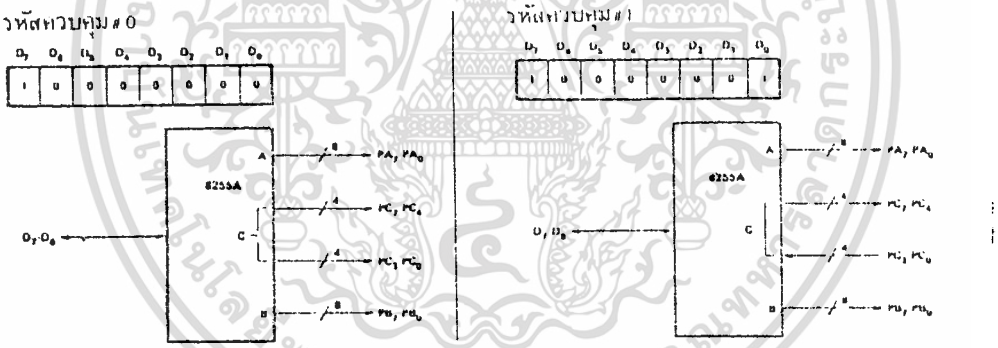
การใช้งาน 8255 จะต้องส่งรหัสควบคุม นั่นคือเราจะต้องควบคุมการทำงานของ Registor ภายใน ผู้ใช้จะต้องทำการ program 8255 ทำงานตามต้องการจากการที่ 8255 มี port ที่ Z-80 มองเห็น 4 port แต่ละ port จะเหมือนเป็น registor ที่สามารถเขียนและอ่านได้ registor แต่ละตัวนี้จึงถูกกำหนดด้วย Adress ตามที่ตั้งไว้ เช่น ในกรณีที่เป็น Adress 10H, 11H, และ 12H registor แต่ละตัวจะได้รับการกำหนดควบคู่กับสัญญาณ RD และ WR เพื่อแสดงความหมาย ตัวอย่างเช่น port 10H เป็น port A ซึ่งเมื่อเขียนที่ port นี้จะเป็นการส่งข้อมูลเอาต์พุท

และถ้าอ่าน port นี้ก็จะเป็นการ Input ข้อมูลจาก port ดังนั้นสัญญาณของขาควบคุมที่ประกอบกันจะแสดงความหมายดังตารางที่แสดงให้ดูที่ผ่านมา

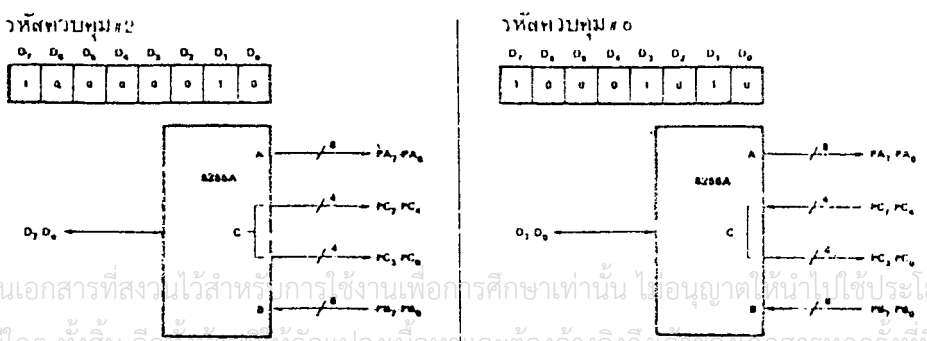
การใช้งาน 8255 จะต้องส่งรหัสควบคุม (Control Code) เข้าไปยัง port ข้อมูลควบคุมเพื่อควบคุมการทำงานของ 8255 โดยใช้สัญญาณควบคุม port หมายเลข 13H การควบคุมการทำงานของ 8255 มีหลาย Mode แต่ละ Mode คือ Mode 0, Mode 1 และ Mode 2

การใช้งาน Mode 0

การกำหนด Mode การทำงานจะต้องส่งข้อมูลคำสั่งเข้าไป program ใน port ควบคุมของ 8255 ซึ่งในที่นี้ใช้ port 80H ไปทำการอ่านค่าใน RTC เพื่อ INT เวลาเข้ามายัง CPU

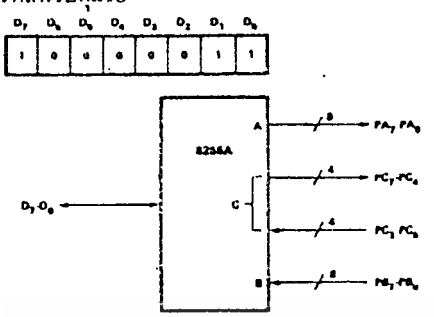


รูปที่ 7.7 ลักษณะของวงจรควบคุมแบบต่าง ๆ ในโหมด 0

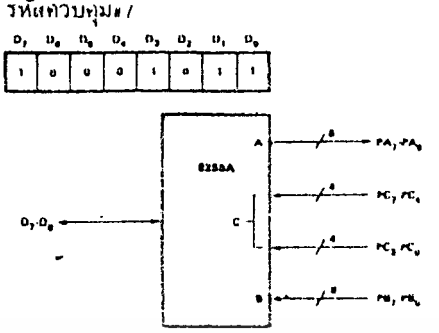


ตารางการทำงานของ Mode 0

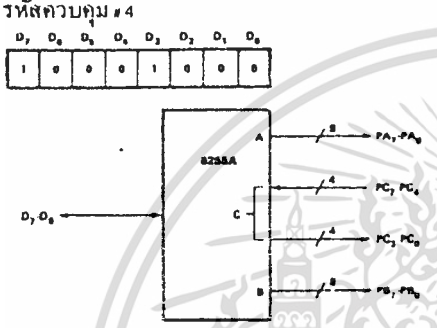
รหัสควบคุม#3



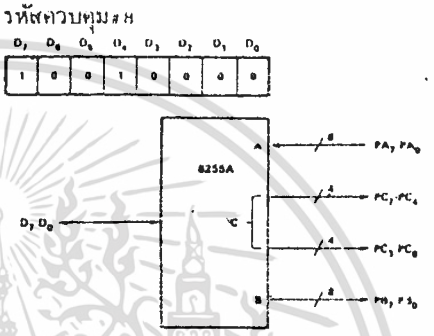
รหัสควบคุม#7



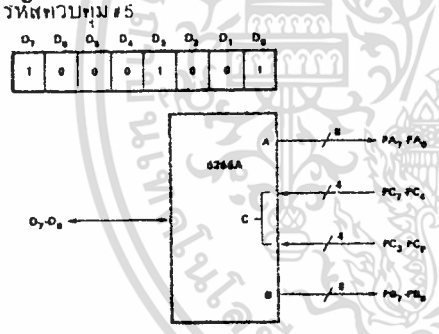
รหัสควบคุม#4



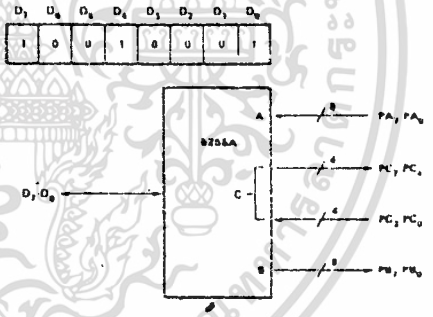
รหัสควบคุม#8



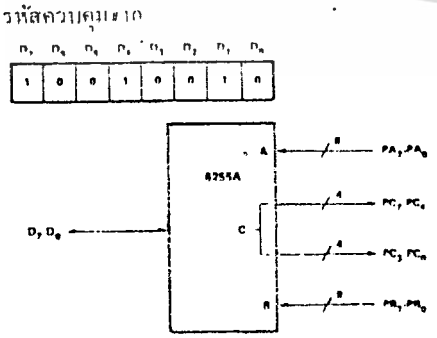
รหัสควบคุม#5



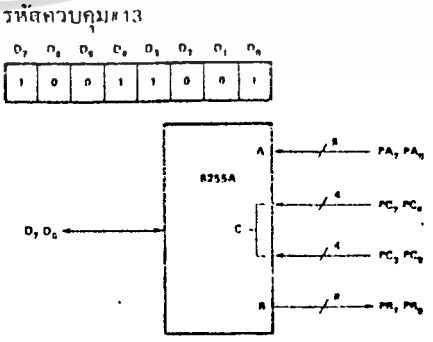
รหัสควบคุม#9



รหัสควบคุม#10



รหัสควบคุม#13



ลักษณะการทำงานของวงจร REAL TIME CLOCK

จากวงจร real time clock และ output control จะขอกล่าวส่วนของวงจร MM58167 ก่อน

ลักษณะของการต่อ MM58167 จะนำขา data D0-D7 ต่อเข้ากับ Z-80 โดยตรง ส่วน address bus จะใช้แค่ A0-A4 เพื่อเป็นตัว decode ค่า address ส่วนขา CS เป็นขา active low เมื่อขานี้ active แสดงว่า 74LS138 โดย portเบอร์ 20H-3FH active ที่ 0 โดยในที่นี้ใช้ port เบอร์ 20H ซึ่ง 74LS138 ได้รับ input A,B,C มาจากขา A5-A7 ของ Z-80 โดยที่ขา G2A,G2B ต่อด้วยกัน และนำไปใช้ต่อกับขา IORQ ของ Z-80

MM58167 จะทำการอ่านเขียนข้อมูลบนบัสข้อมูลได้ ก็จะต้องมีสัญญาณ active ที่ขา RD และ WR อย่างใดอย่างหนึ่งคือ RD,WR จะต้อง active ไม่พร้อมกัน

ส่วนของสัญญาณ INToutput MM58167 ก็จะต่อเข้าขา B ของtransistor BC 547 เพื่อเป็นตัว invertor output ไปยังขา INT ของ Z-80 เพราะขาของ Z-80 INT จะ active ที่ 0

RTC MM58167 มีขา clock คือ X1,X2 ซึ่งจะทำหน้าที่ clock 32.768KHZโดยมี c 10pf และ R 10 โอห์ม เป็นตัวทำให้เกิด oscillator จากนั้นก็จะมีทริมเมอร์ค่า 5-30pf ซึ่งเป็นตัวปรับละเอียด ทำให้ค่าของความถี่ของ oscillator มีค่าเที่ยงตรงยิ่งขึ้น

MM58167 จำเป็นจะต้องมีการ back up ข้อมูลใน registor,counter ต่างๆที่ได้ program ไว้ดังนั้นจะต้องมีถ่าน ni-cd 3.8 volt 180 ma i ก้อนเพื่อคอยจายกระแสให้กับขา 24 vcc โดยการตรวจเช็คสภาวะที่ขา 23 คือขา power down ของ MM58167

ดังนั้นเมื่อป้อนไฟเข้าเครื่อง MM58167 จะทำงานทันทีและ เมื่อเราโปรแกรมค่า registor ต่างๆเรียบร้อยแล้วแม้ไฟจะดับก็จะสามารถเก็บข้อมูลต่างๆได้ เช่น วัน,เดือน, ปี, เวลาและค่าของ control registor และ status registor ต่างๆซึ่งรายละเอียดเกี่ยวกับการ program ค่า registor ภายในได้แสดงไว้แล้ว ในตาราง การ program ค่า registor

battery ni-cd 3.6 v 180 ma จะ charge ตัวเองเมื่อไฟไม่ดับเพราะแรงดันที่จุด source 5v มากกว่า battery ดังนั้น battery จะ charge ตัวเองไปเรื่อยๆ

ภาค OUTPUT CONTROL

จากวงจร output control จะใช้หลักการง่ายๆคือ นำขา data D0-D7 มาเข้าขา data ทั้ง 8 เส้นของ IC 74LS374 ซึ่งเป็น IC ที่ทำหน้าที่เป็น output port ได้ดีโดยภายในมี flipflop อยู่ถึง 8 ตัว และใช้ขา oc เป็นตัว control ร่วมกับขา clock เพื่อให้ output ที่ Q0-Q7 active 1 ไปยังขา base ของ transister BC 549 จำนวน 8 ตัว เพื่อให้ transister เป็นตัวขับ relay ทั้ง 8 ตัว ซึ่ง relay ทั้ง 8 ตัวจะใช้ขนาด 2A voltage จุดบนขดลวดเท่ากับ 6v ส่วนหน้าสัมผัสของ relay คือ NO และ NC จะนำไปประยุกต์ใช้งานเป็น switch อย่างไรก็ได้โดยได้ถึง 8 ชุด หรือ 8 ช่องสัญญาณควบคุม

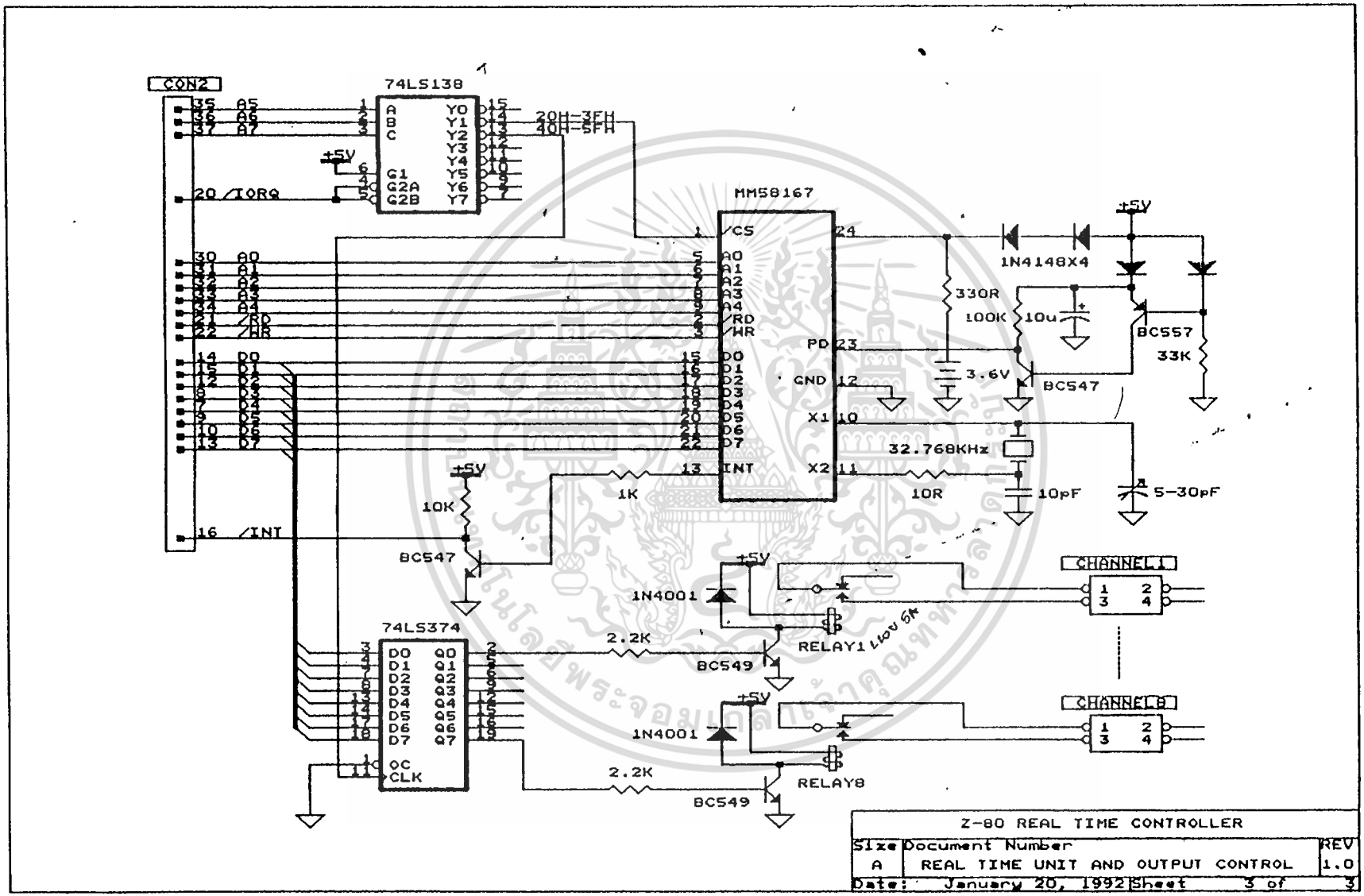
การที่ใช้งาน output control เป็นแบบ relay ก็เพราะว่าสะดวกในการประยุกต์ใช้งาน แต่มีข้อเสียที่ว่าถ้ากระแสมากจะมีขนาดใหญ่ ด้านในการใช้งานที่ต้องการขับ load ที่มีขนาดกระแสมาก เช่น motor ก็ทำได้โดยการต่อพ่วงกับ mahetic relay อีกทีหนึ่ง

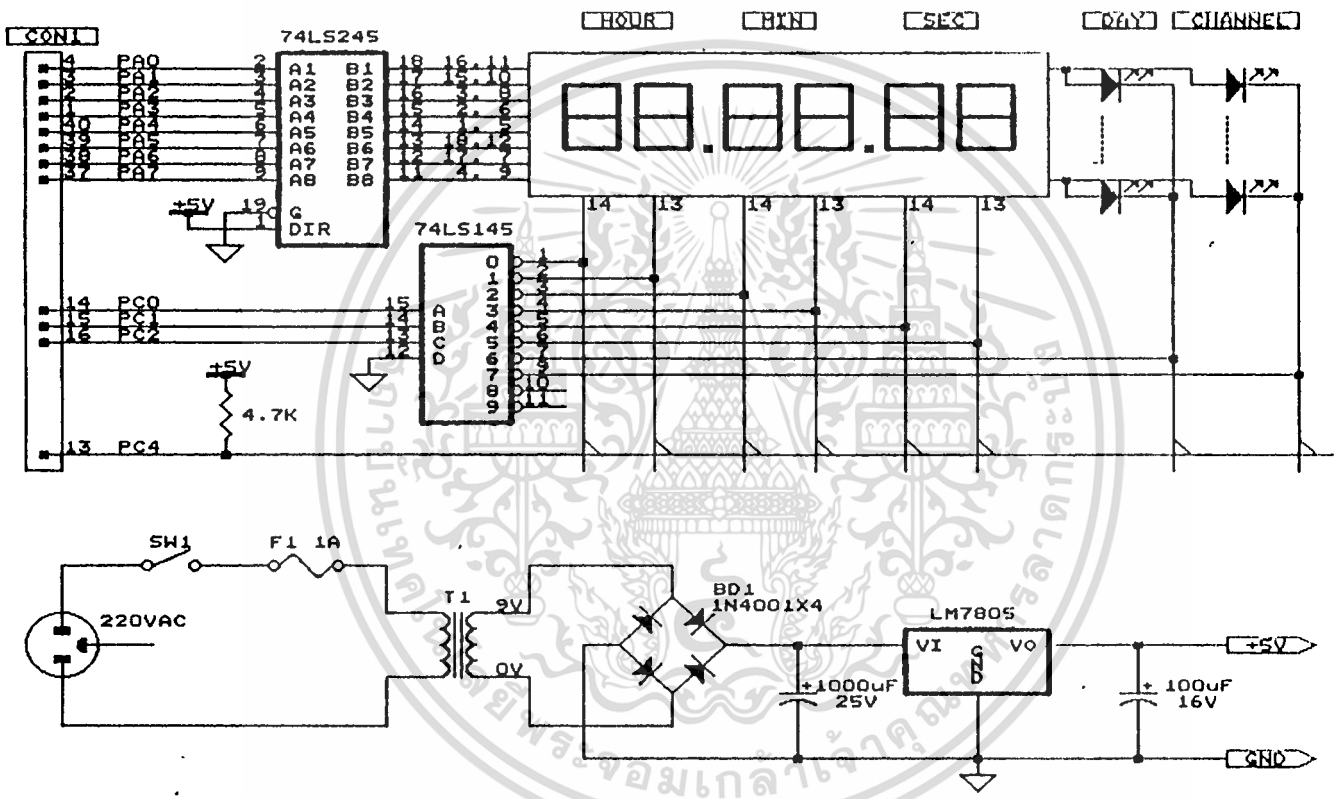
ลักษณะการทำงานของส่วน DISPLAY และ KEYBOARD

ส่วนของ port PA0-PA7 ของ 8255 จะนำมาต่อกับ 8 bit latch เบอร์ 74LS245 ซึ่งภายในเป็น tri state ซึ่ง 74LS245 จะทำหน้าที่เป็น output port เพื่อใช้ในการส่งค่าของอักษรและตัวเลขต่างๆ โดยที่ IC 74LS145 เป็นตัว scan ทางแนวตั้ง ซึ่ง 74LS145 เป็น IC input เป็น bcd และออกเป็น decimal

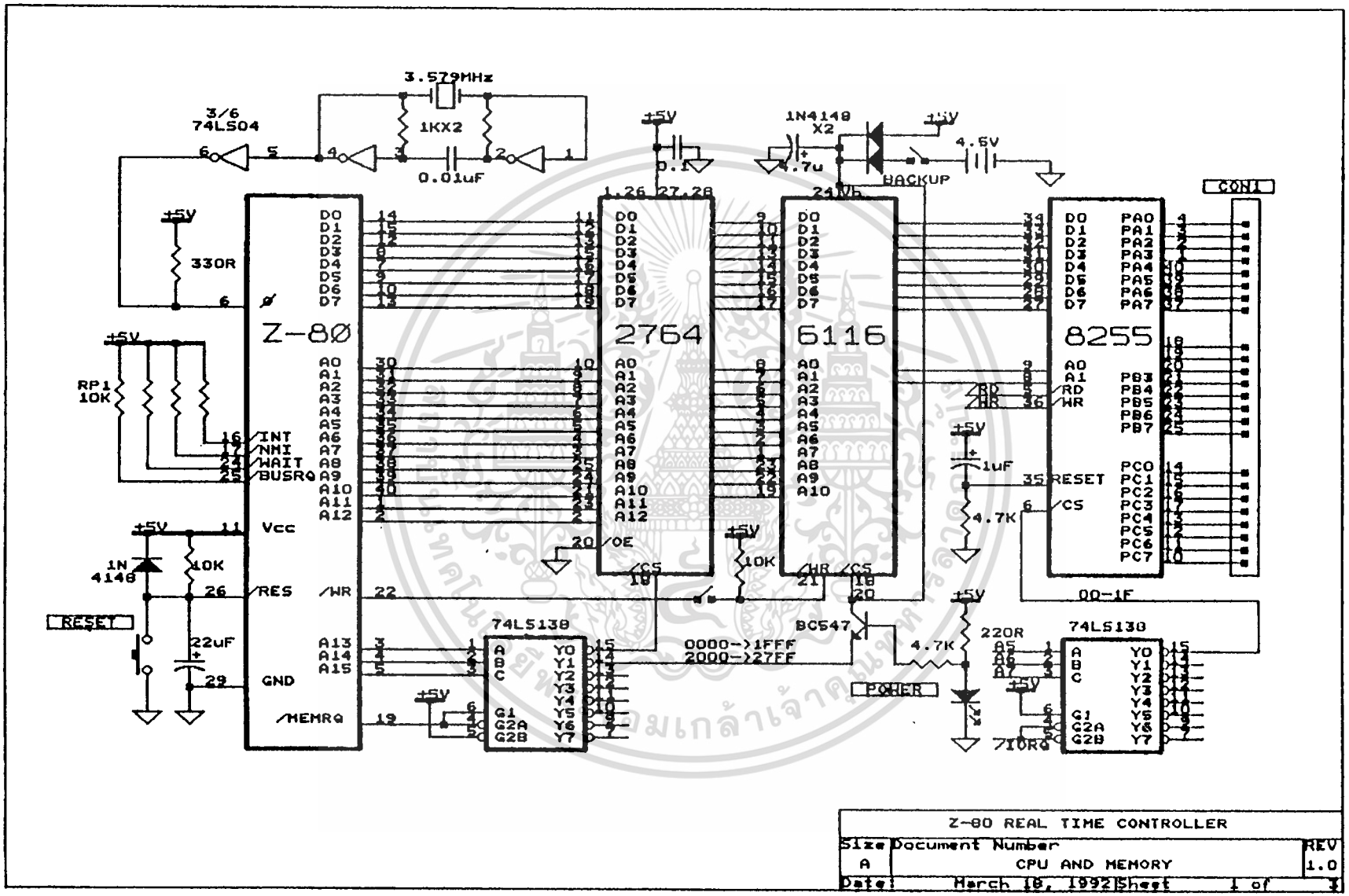
การ PROGRAM PORT 8255

1. ให้ port A PA0-PA7 เป็น output ทำหน้าที่ส่ง data scan ทาง row เพื่อ scan seven segment และ LED แสดงวันและ channel
2. ให้ port PC0-PC2 เป็น output port ทำหน้าที่ในการ scan ทาง colum
3. ให้ PC4 เป็น input port รับข้อมูลจากการกด keyboard


CON2
74LS138
20H-3FH
40H-5FH
+5V
20 /IORQ
MM58167
/CS
PD
GND
X1
X2
INT
/RD
/WR
1N4148X4
330R
100K
10u
BC557
33K
3.6V
BC547
32.768KHz
10pF
5-30pF
1K
10R
10K
16 /INT
74LS374
OC
CLK
1N4001
2.2K
RELAY1
RELAY8
CHANNEL1
CHANNEL8
Z-80 REAL TIME CONTROLLER
Size
A
Document Number
REAL TIME UNIT AND OUTPUT CONTROL
REV
1.0
Date: January 20, 1992
Sheet 3 of 3

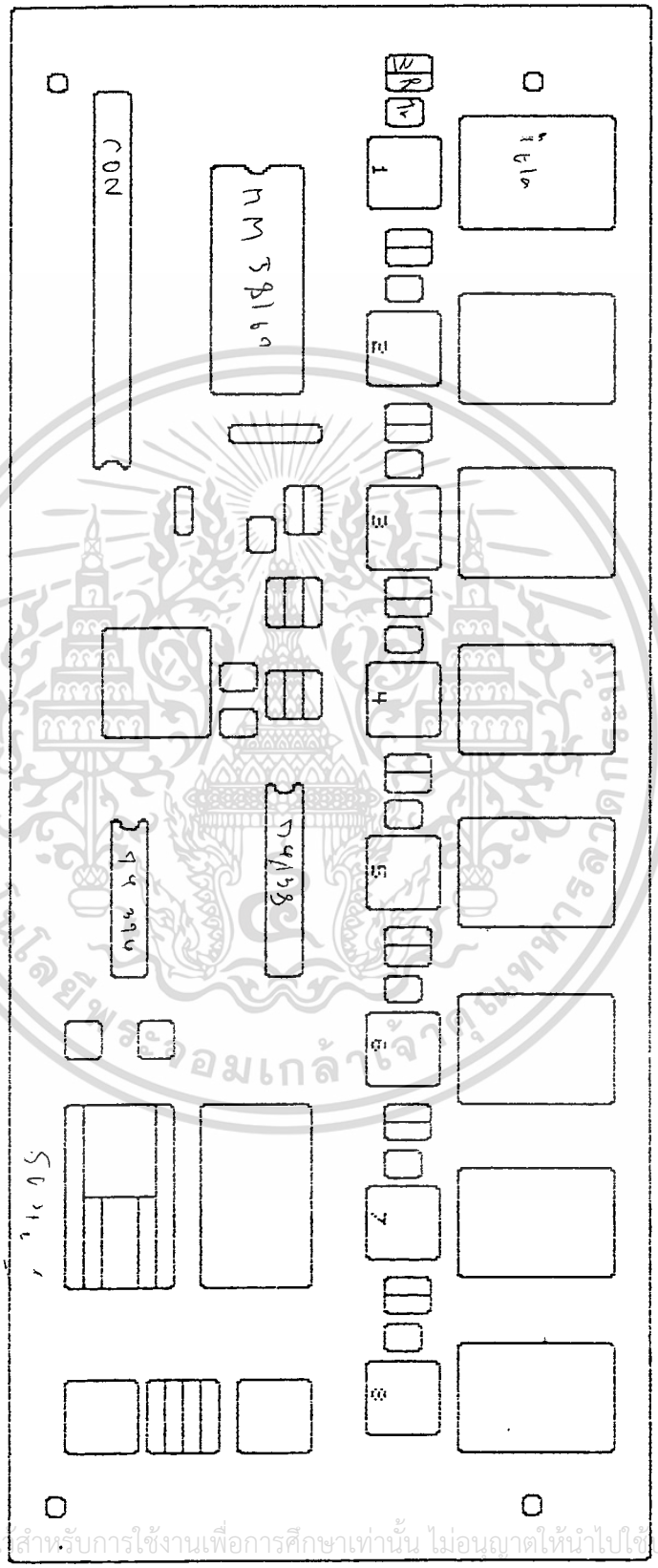


Z-80 REAL TIME CONTROLLER		
Size: A	Document Number: DISPLAY, KEYBOARD & POWER SUPPLY	REV: 1.0
Date: February 5, 1992	Sheet 2 of 3	

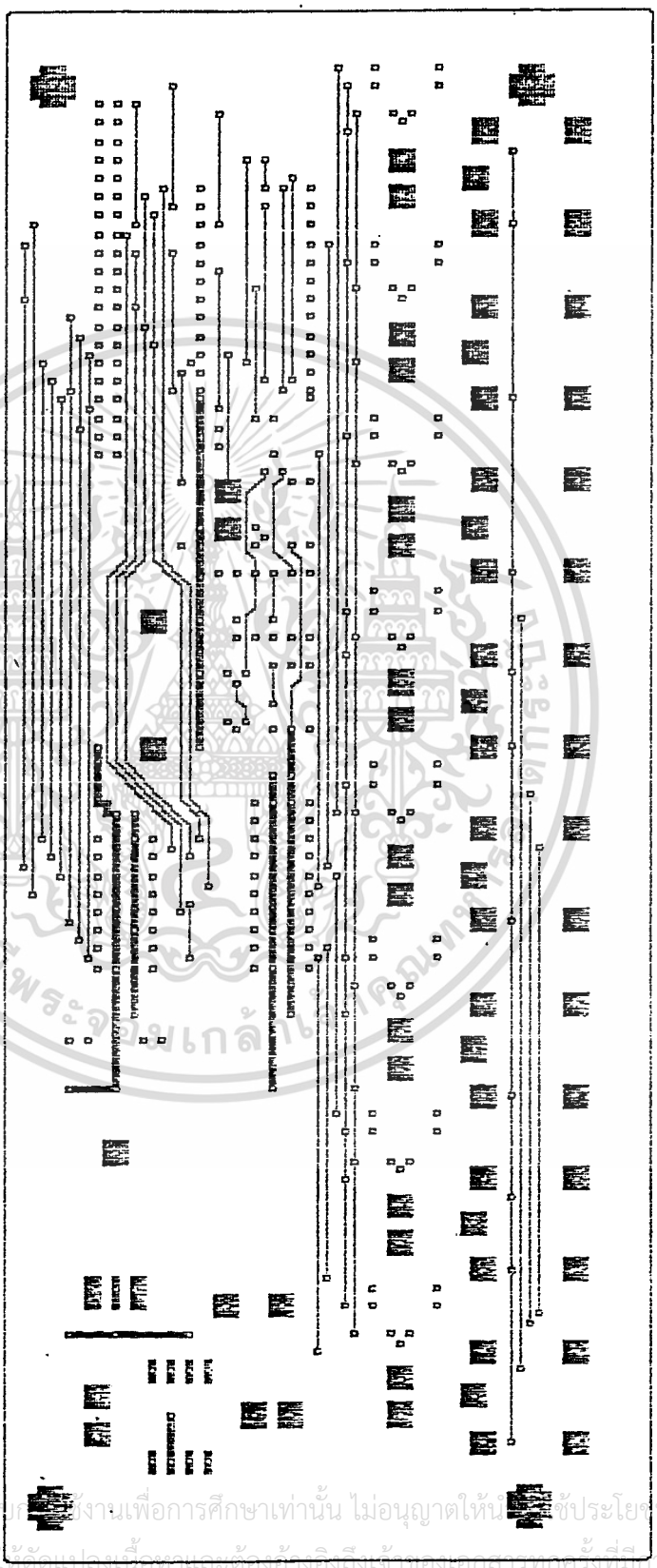


Z-80 REAL TIME CONTROLLER		
Size: A	Document Number: CPU AND MEMORY	REV: 1.0
Date: March 10, 1992	Sheet 1 of 1	

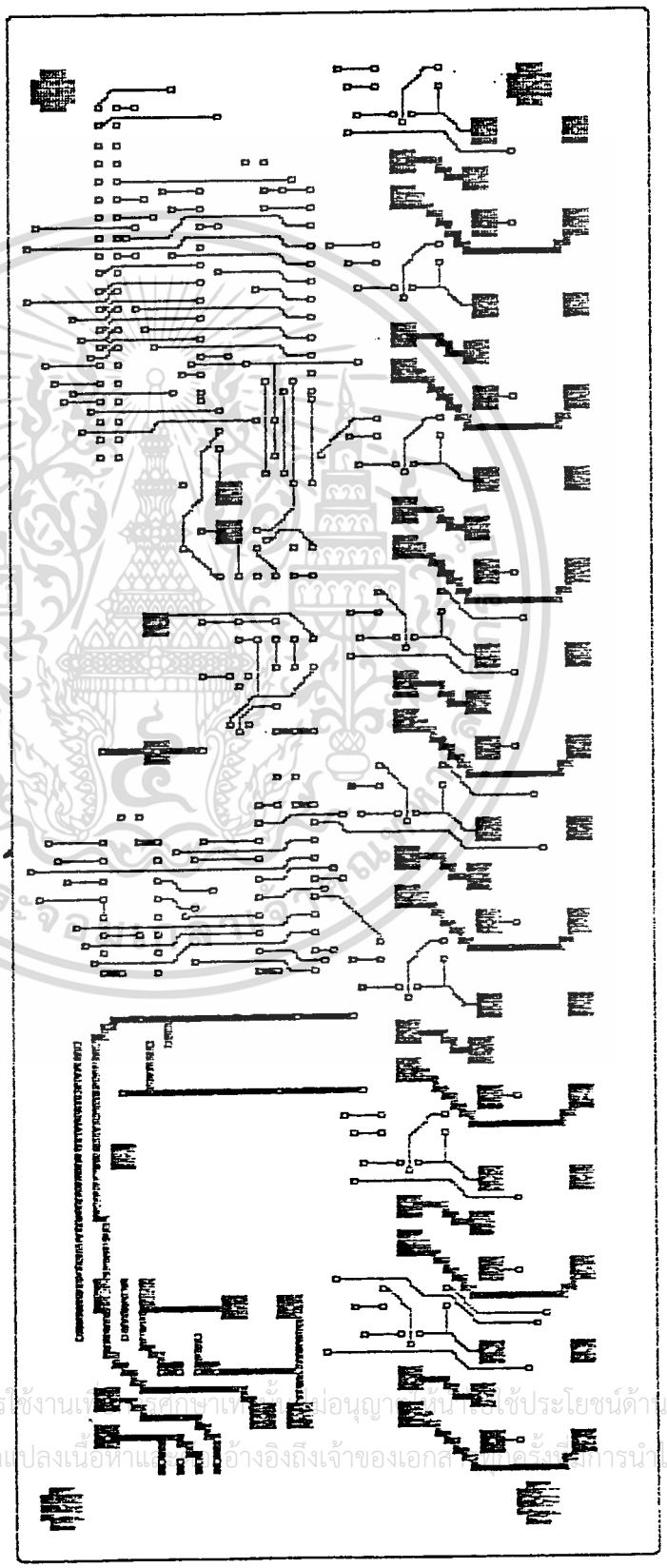
1X checkplot 18 Mar 1992 18:56:50
project2.pcb
v1.2 r3 holes: 420 silkscreen
approximate size: 8.50 by 3.55 inches



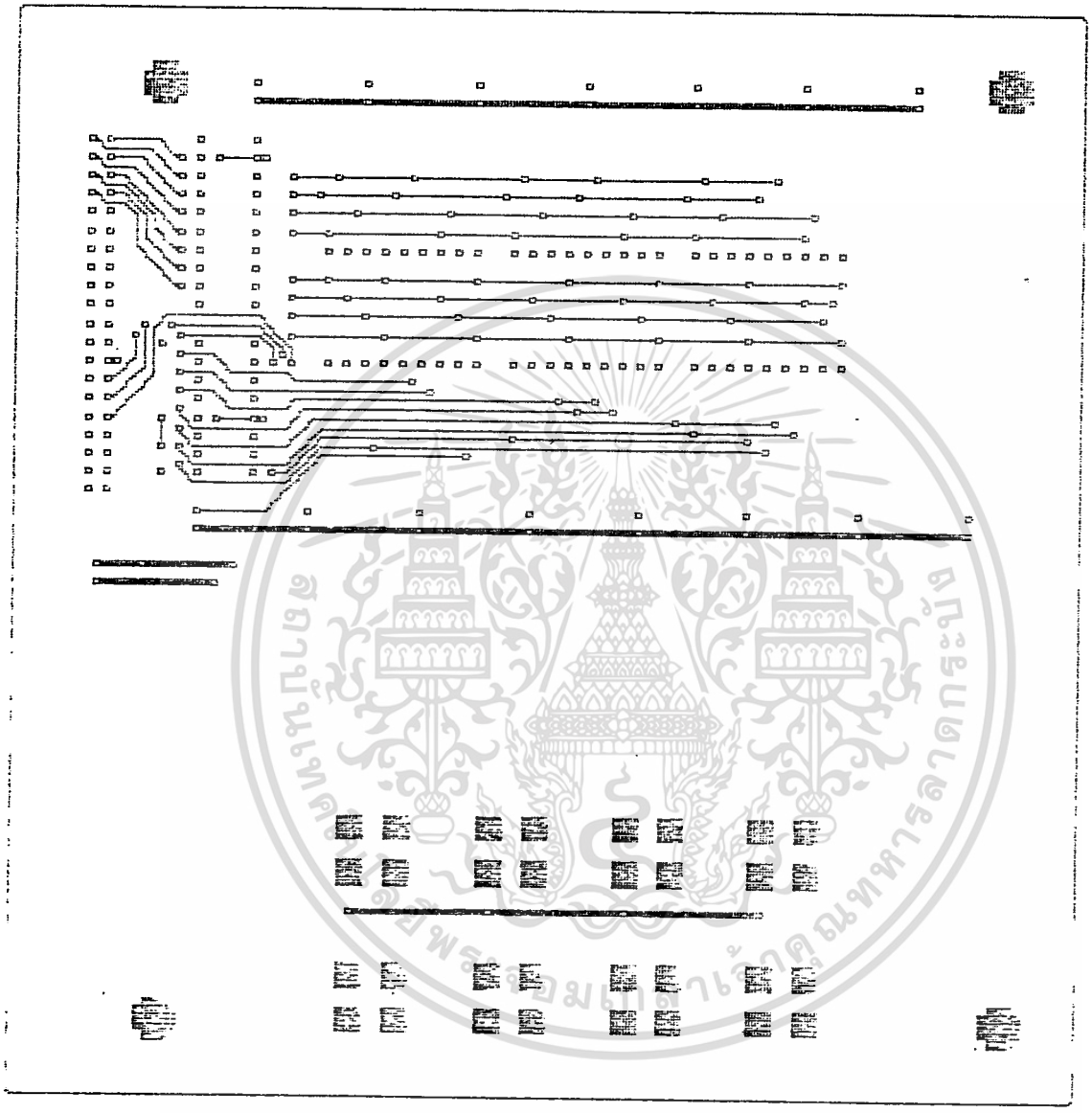
1X checkplot 18 Mar 1992 18:55:22
project2.pcb
v1.2 r3 holes: 420 component side
approximate size: 8.50 by 3.55 inches



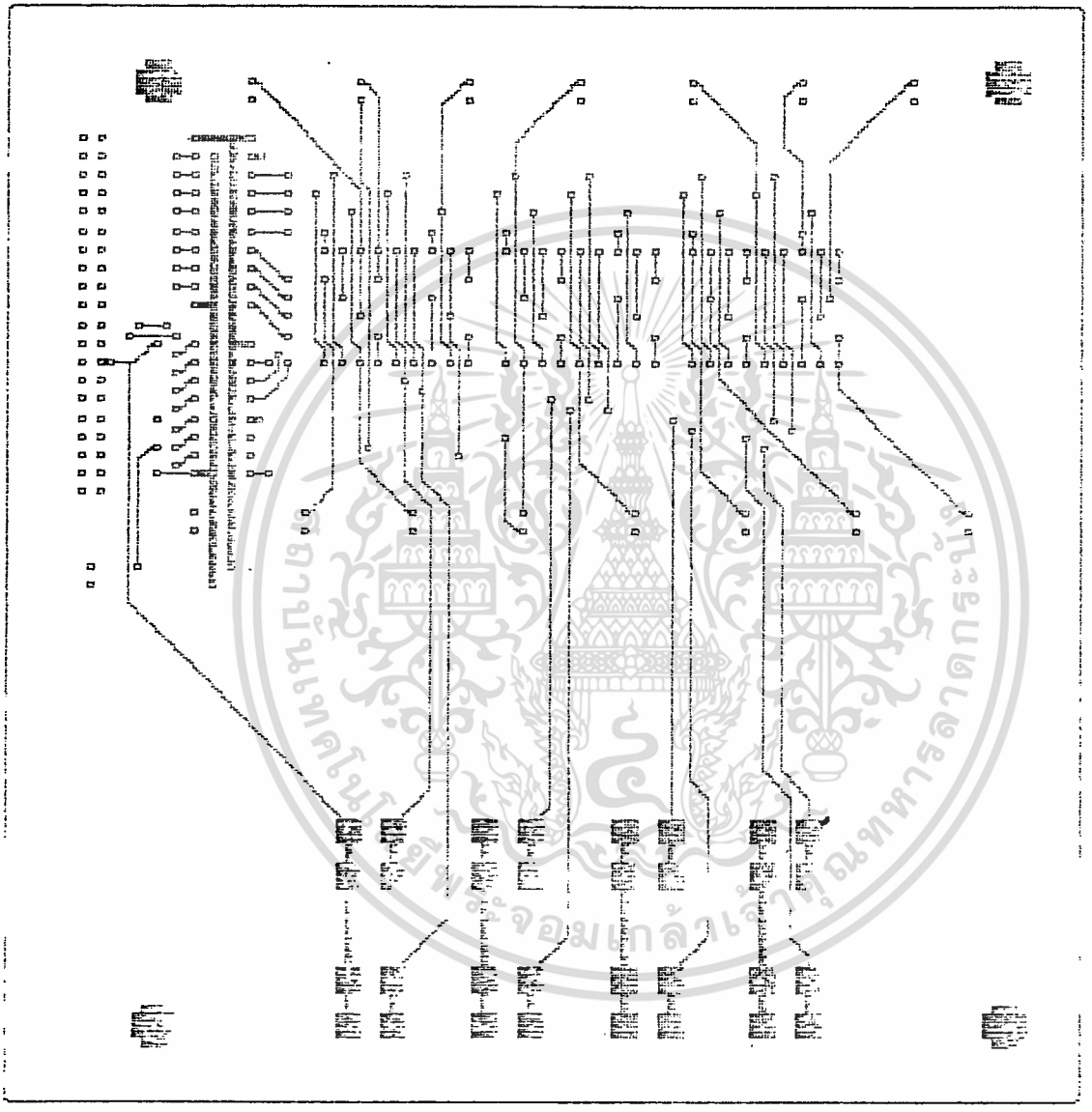
1X checkplot 18 Mar 1992 18:53:50
project2.pcb
v1.2 r3 holes: 420 solder side
approximate size: 8.50 by 3.55 inches



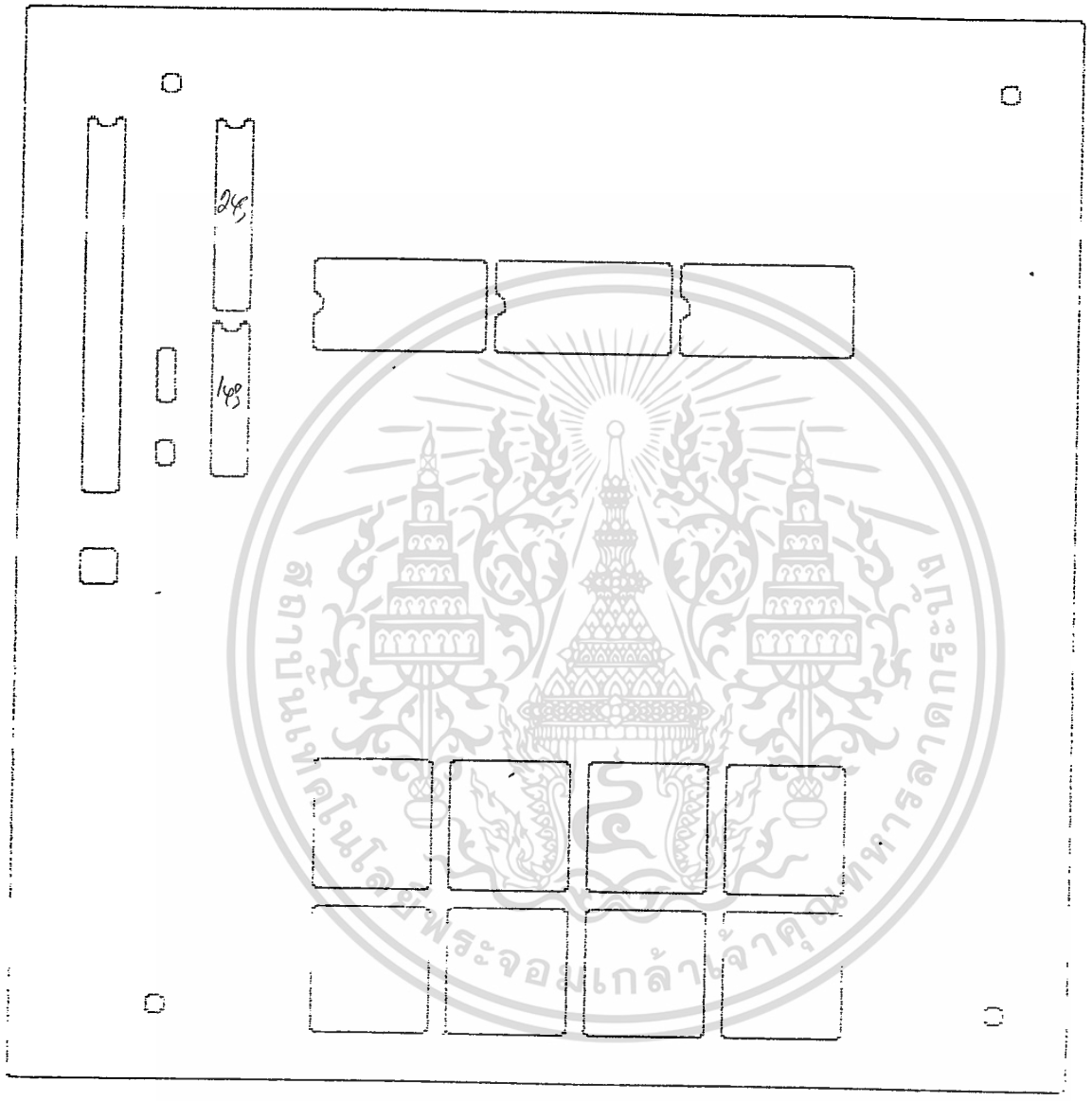
1X checkplot 18 Mar 1992 18:48:49
project1.pcb
v1.2 r3 holes: 310 component side
approximate size: 5.85 by 5.90 inches



1X checkplot 18 Mar 1992 18:47:18
project1.pcb
v1.2 r3 holes: 310 solder side
approximate size: 5.85 by 5.90 inches



1X checkplot 18 Mar 1992 18:50:28
project1.pcb
v1.2 r3 holes: 310 silkscreen
approximate size: 5.85 by 5.90 inches



บทที่ 3

การใช้งานของเครื่องและการ SET PROGRAM

การตั้งเวลา

1 — กด key TIME ที่ display จะแสดงผล DAY

_ป้อนค่าวันให้ถูกต้อง (อาทิตย์-เสาร์)โดย shift LED

-กด TIME เครื่องแสดง ค่า ช.ม.

_กด INC และ DEC เพื่อตั้งค่า ช.ม.ที่ถูกต้อง

_กด TIME หลัก นาทีจะแสดงผล

_กด INC และ DEC เพื่อตั้งค่านาที ที่ถูกต้อง

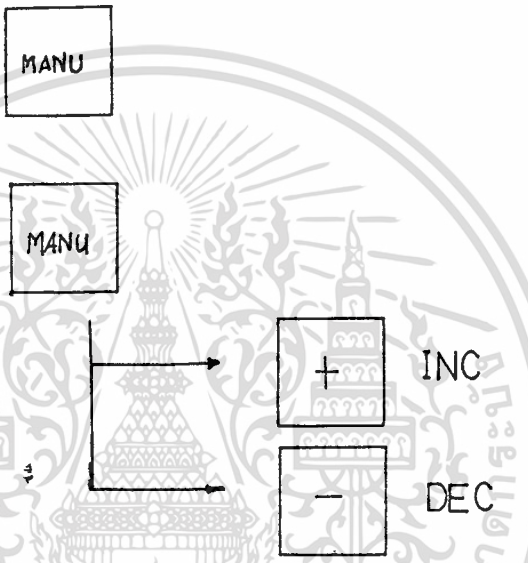
_กด TIME หลักวินาทีจะเริ่มนับจาก 00 วินาที

2 — หลังจากทำตาม ข้อ 1 นาฬิกาจะเริ่มเดิน

_ถ้าต้องการตั้งเวลาใหม่ก็ทำได้โดยการกด SET

การควบคุมอุปกรณ์เอาท์พุทโดยตรงจากคีย์บอร์ด

1 — กดคีย์ "MANU" เข้าสู่โหมดการควบคุมอุปกรณ์เอาท์พุท โดยตรงจากคีย์บอร์ด (DIRECT MODE) ที่ตัวเลขแสดงผลจะแสดง "ON" หรือ "OFF" ตามสภาวะ ON หรือ OFF ของเอาท์พุทที่ 1



2 — เลือกหมายเลขช่องเอาท์พุท โดยใช้คีย์ "MANU" เพื่อเปลี่ยนหมายเลขช่องเอาท์พุทที่ต้องการควบคุมตัวเลขหลักที่สองจะแสดงหมายเลขเอาท์พุทส่วนตัวเลขสองหลักสุดท้ายจะแสดงสภาวะปัจจุบันของเอาท์พุท

การออกจากโหดนี้ ให้กดคีย์โหดอื่นที่ต้องการให้เครื่องไปทำงาน เช่น กดคีย์ "TIME" เครื่องจะกลับไปทำงานในโหมดแสดงเวลาทันที

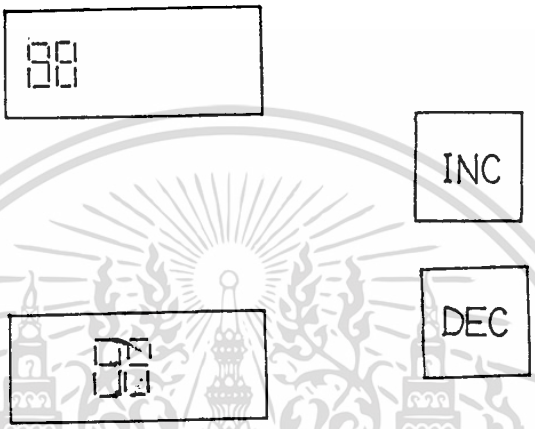
3 — กด key "PROG" ที่ส่วนแสดงผลจะปรากฏตัวอักษรคำว่า "day-" และที่หลอดไฟสามารถกด คีย์ INC และ DEC เพื่อค่าของวันได้(LED SHIFT DAY)

INC

DEC

4 — ตั้งแบบของวัน โดยใช้ key INC และ DEC เพื่อเลือกแบบของวัน ที่จะให้โปรแกรมนี้ทำงานตามต้องการ แบบของวันมีไว้เพื่อทำให้โปรแกรมที่ตั้งไว้ 1 โปรแกรม สามารถควบคุมอุปกรณ์ output ให้ทำงานได้มากกว่า 1 วัน โดยสามารถกำหนดรูปแบบของของวันได้ 50 แบบ LED แสดงวัน

5— กด key "PROG" ที่ตัวเลขแสดงผลจะปรากฏตัวแรกแสดงเวลา ที่จะให้program ทำงานเป็นชั่วโมงและนาที และวินาทีโดยตัวเลขแสดงชั่วโมง นาทีและวินาทีจะติดเรียงกันไป



6— ตั้งชั่วโมง ที่จะให้โปรแกรมนี้ทำงานโดยใช้ key INC และ DECเพื่อเปลี่ยนค่าชั่วโมงตามต้องการ (ค่า 0-23) ตัวเลขแสดงชั่วโมงจะเปลี่ยนไป

7 — กด key "PROG" ที่ตัวเลขแสดงชั่วโมงจะหยุดและตัวเลขนาทีจะแสดงออกมา

8 — ตั้งนาทีที่จะให้โปรแกรมนี้ทำงานโดยใช้ key INC and DEC เพื่อเปลี่ยนค่านาที ตามต้องการ (ค่า 0-59) ตัวเลขแสดงนาทีจะเปลี่ยนแปลง

9— กด key "PROG" ที่ตัวเลขแสดงผลจะปรากฏตัวอักษร "on" โดยที่ตัวเลขแสดง

10— ตั้งลักษณะควบคุมการปิด/เปิด อุปกรณ์ output (type) use INCลักษณะการควบคุม output ได้2แบบด้วยกัน คือโปรแกรม ON (อุปกรณ์ output ทำงาน) โปรแกรม off (อุปกรณ์ output หยุดทำงาน) โดยตั้งละเอียดได้เป็นวินาที

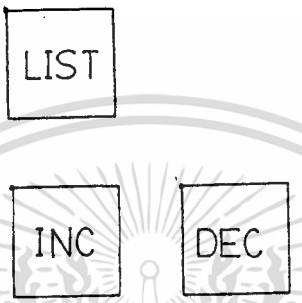
1 1 — PROGRAM ก็จะถูกเก็บในหน่วยความจำ เมื่อกด PROG

การเรียกดูโปรแกรมและการลบโปรแกรม

1 — กด key "LIST" เข้าสู่โหมดการเรียกดูโปรแกรม (LIST MODE)
ในกรณีที่มี program อยู่ในหน่วยความจำของเครื่อง ที่หลอด LED จะแสดงวันในรอบสัปดาห์จะใช้แสดงแบบของวันของโปรแกรม ส่วนตัวเลขแสดงผลจะใช้แสดงข้อมูล 2 ชุดโดยแสดงผลสลับไปมา (ON,OFF)

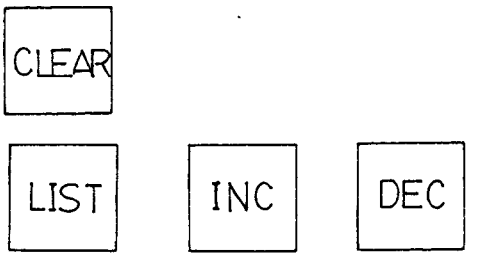
ข้อมูลชุดแรก ที่ตัวเลขหลักแรกจะแสดงหมายเลขช่องเอาท์พุท (ค่า CH1-8) และสองหลักสุดท้ายจะแสดงลักษณะของการควบคุมเอาท์พุท (ON,OFF หรือเป็นค่าการปิดเปิดมีความละเอียดเป็นวินาที)

และกด key "LIST" อีกครั้งเพื่อแสดงโปรแกรมถัดไป โดยถ้าแสดงโปรแกรมจนหมดแล้วการแสดงผลก็จะกลับไปแสดงเวลาดังเดิมในกรณีที่ไม่มีโปรแกรมอยู่ในหน่วยความจำของเครื่องที่ตัวเลขแสดงผลจะปรากฏเครื่องหมาย"----"



2— ถ้าต้องการลบโปรแกรมออกจากหน่วยความจำของเครื่อง ให้ทำตามขั้นที่ 1 เพื่อแสดงโปรแกรมที่ต้องการลบให้ปรากฏที่ส่วนแสดงผล กดคีย์ "CLEAR" เพื่อลบโปแกรมออกจากหน่วยความจำตัวเลขแสดงผลจะปรากฏคำว่า "CONF"(CONFIRM) เพื่อถามว่าต้องการจะลบโปรแกรมนี้หรือไม่

ถ้าต้องการลบโปรแกรม ให้กดคีย์ "CLEAR" อีกครั้ง หนึ่งครั้งเพื่อบอกให้ทราบว่าเครื่องได้ทำการลบโปรแกรมนั้นออกจากหน่วยความจำเรียบร้อยแล้ว ถ้าโปรแกรมที่ลบนั้นไม่ใช่โปรแกรมสุดท้าย ที่ส่วนแสดงผลจะแสดงโปรแกรมและเมื่อกด TIME ผลจะกลับไปแสดงผลจะกลับไปแสดงเวลาตามเดิม

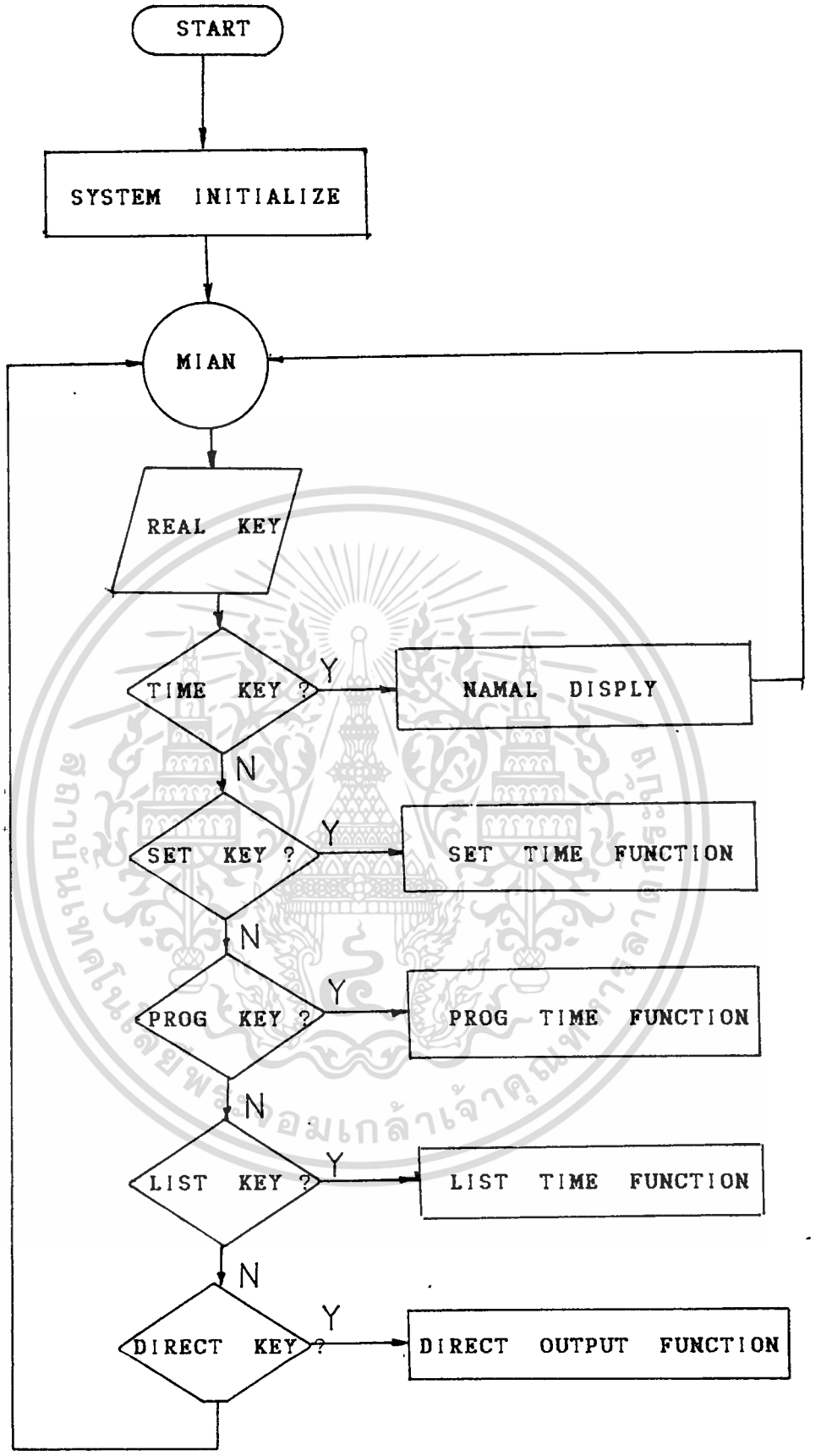


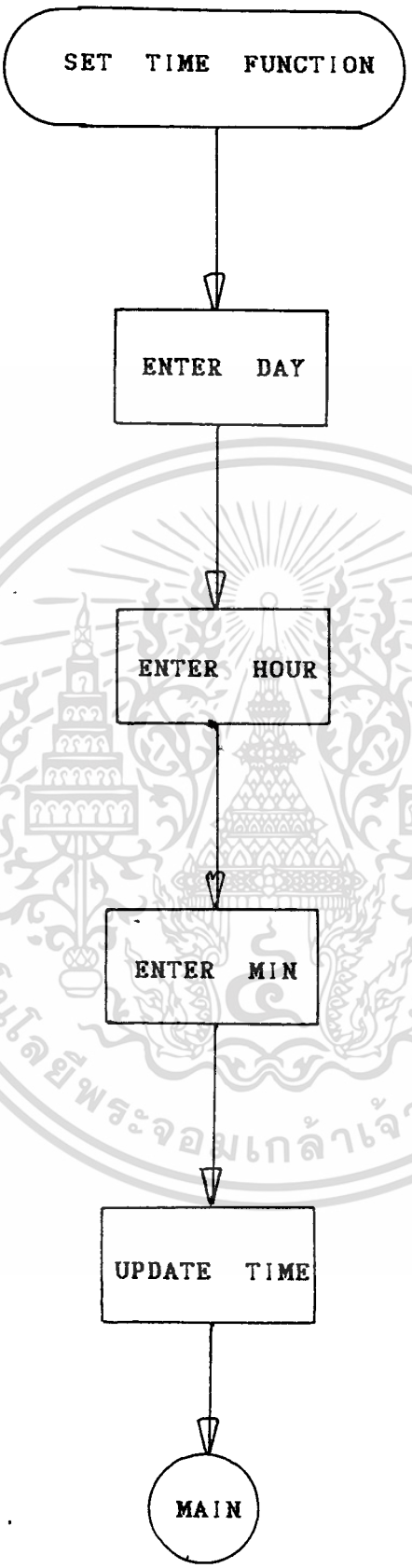
ถ้าไม่ต้องการลบโปรแกรม ให้กลับไปทำตามขั้นตอนที่ 1. เครื่องจะกลับไปแสดงโปรแกรมเดิม

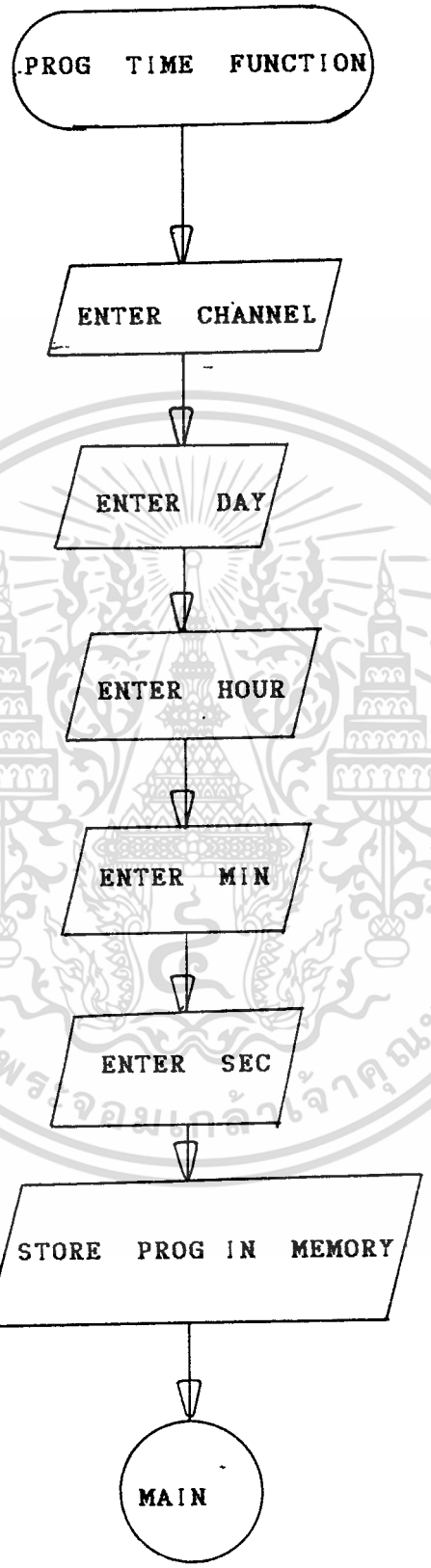
บทที่ 4

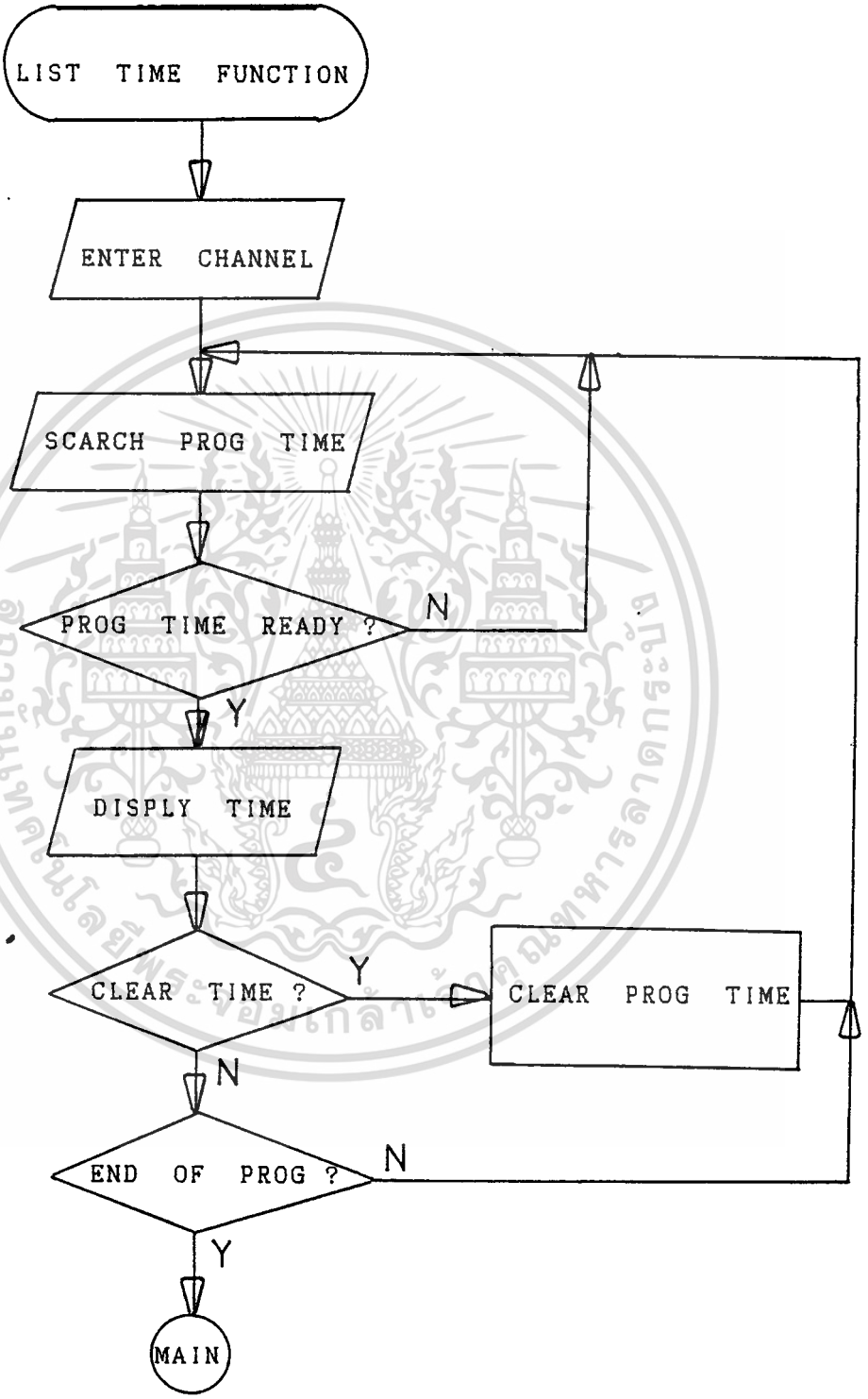
การทำงานของ PROGRAM และ FLOW CHART

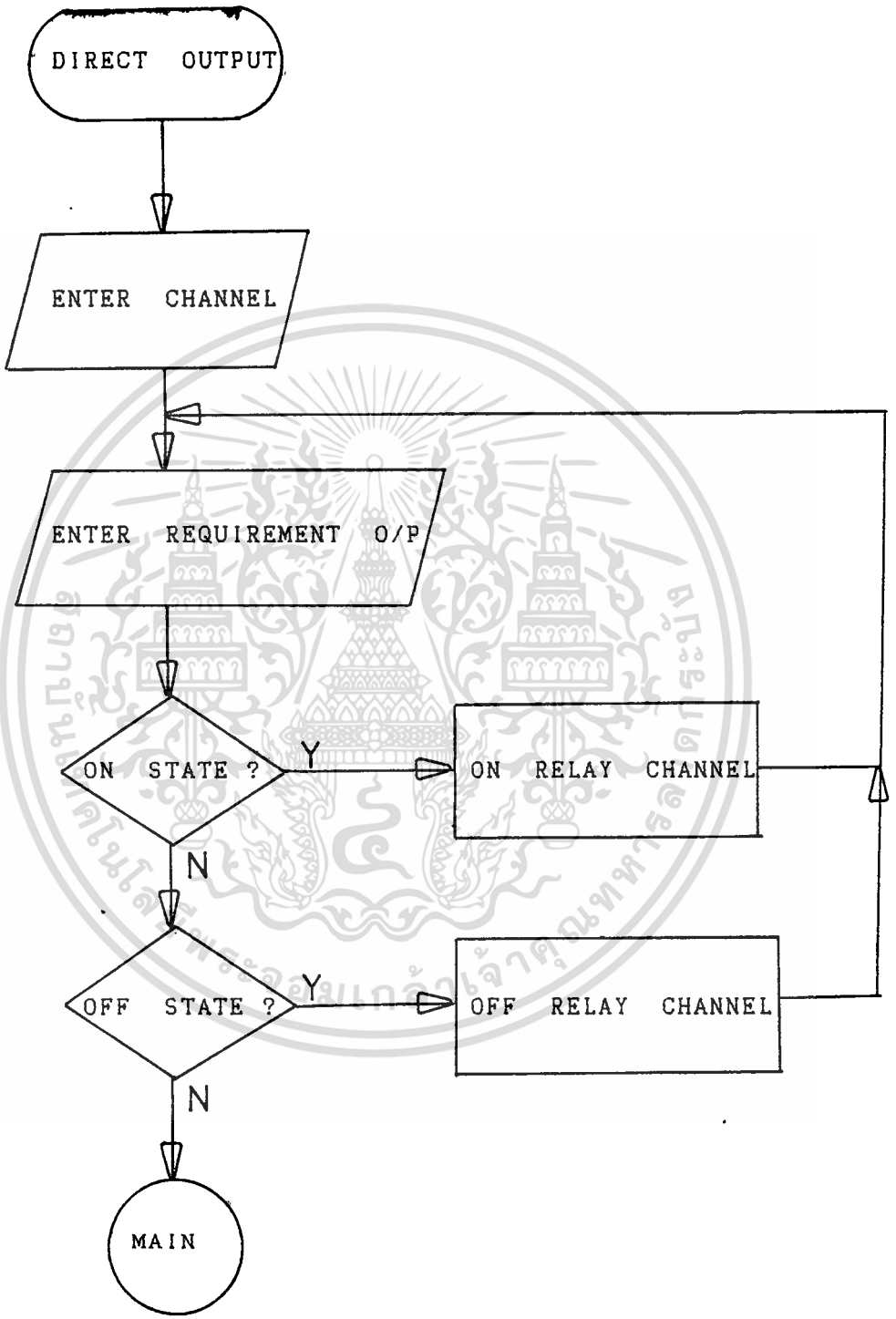
FLOW CHART



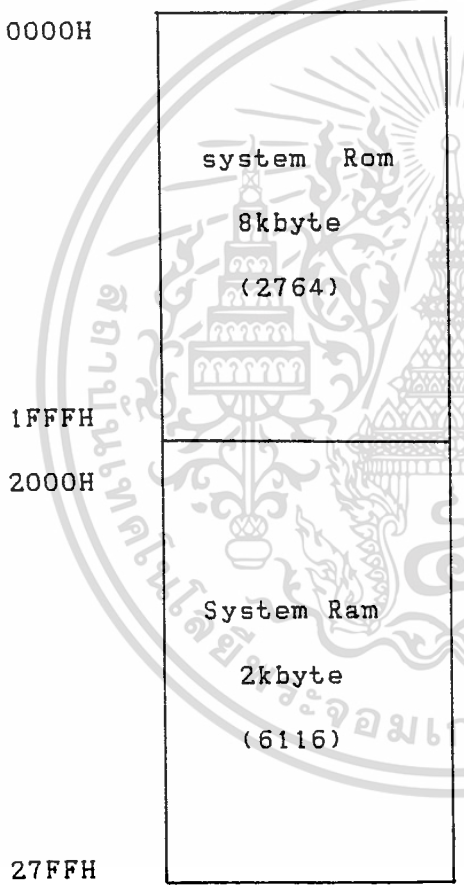




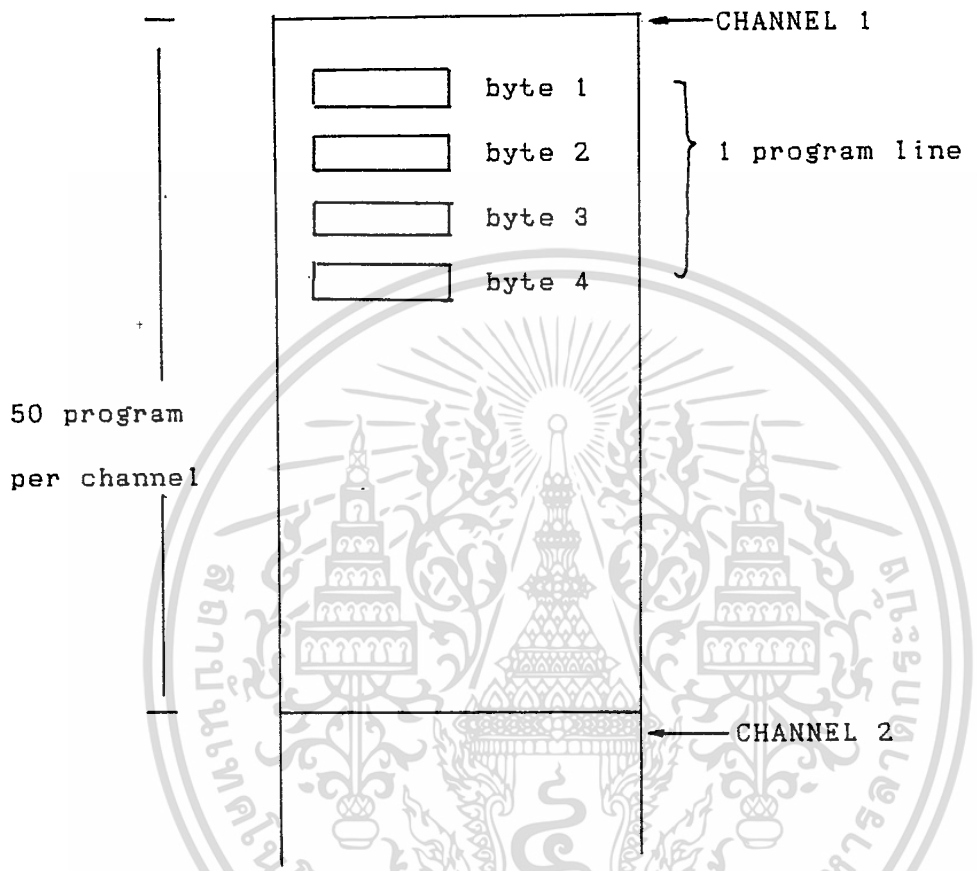




MEMMORY MAP

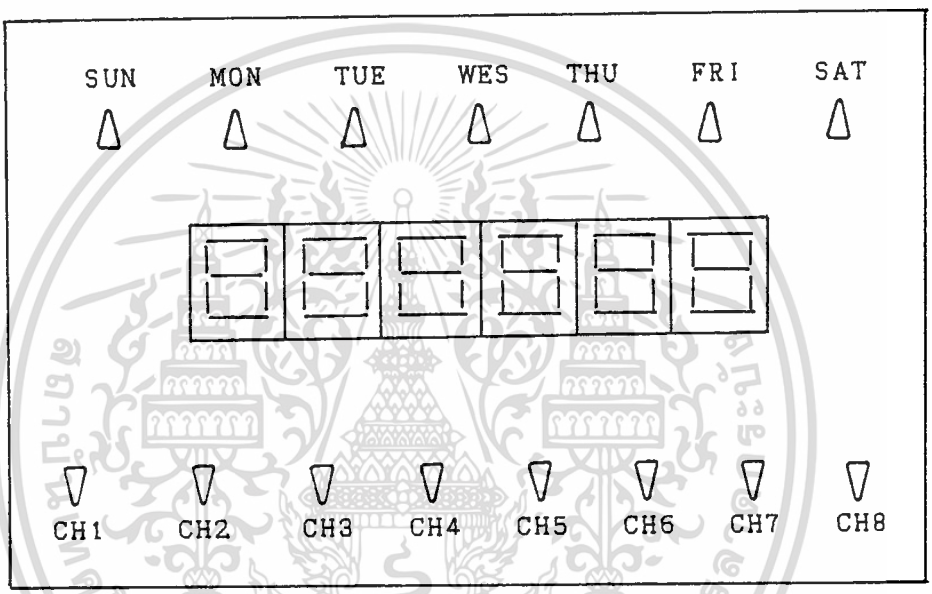


system byte



ภาพแสดงกล่องภายนอก

PROGRAMMABLE CLOCK CONTROLLER



TIME	PROG	MANU	INC/ON
SET	LIST	CLEAR	DEC/OFF

คุณสมบัติทางเทคนิค

1) X-tal	ความถี่ 32.675 KHz
2) Real time clock	Ic เบอร์ MM58167A
3) CPU	Ic Z-80A
4) Program	ความจุ 50 โปรแกรม/ช่อง
5) Keyboard	8 Key
6) Display output	LED and Seven segment
7) Control Unit	Relay 8 channel Max 2A
8) Power consumtion	7 w (approximate)
9) Data Retention Time	6 month
10) DC Power regulation	+5 v

บทสรุปและวิจารณ์ผลการทำวิทยานิพนธ์

จากการที่ได้ทดลองนำ real time clock เบอร์MM58167 มาใช้งานกับ z-80 ผลคือใช้งานร่วมกันได้ดี และการเขียนprogram interupt ก็สามารถกระทำได้ไม่ยากนัก อีกทั้งการใช้นาฬิกาโปรแกรมเวลา ก็สามารถต่อกับ load ได้ถึง 8 channel ในเวลาเดียวกัน การทำงานก็ออกแบบให้มีความละเอียดเป็นค่าวินาที ซึ่งมีประโยชน์ในการควบคุมงานที่ต้องการความเที่ยงตรงและมีความละเอียดสูง

ปัญหาที่เกิดขึ้นในการทำโปรแกรมซึ่งมีความยุ่งพอสมควรในด้านการเปรียบและการ scan key และการ compare program ซึ่งมีความยาวมาก อีกทั้งหน่วยความจำ Rom ที่ออกแบบมาในแผ่นปริ้นของจริงนั้น เป็นเบอร์ 2732 ความจุ 4 kbyte แต่พอเขียน program จริงปรากฏว่ามากกว่า 4kbyte ดังนั้นจึงต้องเปลี่ยนมา เป็น 2764 ขนาด 8kbyte อีกประการหนึ่งการ scan display จะมีการกระพริบเล็กน้อยซึ่งเกิดจากวนของ program scan display

ในด้านการใช้งานนั้นมีความเที่ยงตรงในการบอกเวลา จึงขอฝากให้ผู้ที่สนใจนำไปประยุกต์ใช้งานต่อไป

หนังสืออ้างอิง

ยืน ภู่วรวรรณ "ทฤษฎีการใช้งานไมโครโปรเซสเซอร์ Z80",พิมพ์ครั้งที่ 1,2532

วิบูลย์ ชื่นแขก "ไมโครโปรเซสเซอร์ และ ไมโครคอมพิวเตอร์",สถาบันเทคโนโลยีพระจอมกล้าพระนครเหนือ

บริษัท ETT จำกัด "REAL TIME CLOCK"

ZILOG .,INC "Z80 TECHNICAL MANUAL",1977

RODNAY ZAKS "PROGRAM THE Z80",SYBEX.,INC,1980

ภาคผนวก 1

MONITOR PROGRAM

TURBO80. Z-80 Assembler Ver 1.00

```
Line  Addr  Obj

   1                        ;FILE NAME      RTC-CONT.ASM
   2                        ;DESCRIPTION    REAL TIME CONTROLLER
   3                        ;HARDWARE       Z-80 MICRO CONTROLLER BOARD
   4                        ;ASSEMBLER      TURBO 80 ON MS-DOS
   5
   6                        ;****************************************
   7                        ;*              VARIABLE SET            *
   8                        ;****************************************
   9  0000                  PPI       EQU   00H          ;8255 PPI
  10  0000                  SEG       EQU   PPI          ;SEGMENT PORT
  11  0002                  DIGIT     EQU   PPI+2        ;DIGIT PORT
  12  0002                  KEYIN     EQU   PPI+2        ;KEY IN PORT
  13  0020                  RTC       EQU   20H          ;REAL TIME CHIP
  14  0022                  SEC       EQU   RTC+2
  15  0023                  MIN       EQU   RTC+3
  16  0024                  HOUR      EQU   RTC+4
  17  0025                  DAY       EQU   RTC+5
  18  0040                  RELAY     EQU   40H          ;RELAY PORT
  19
  20                        ;****************************************
  21                        ;*          SYSTEM WORKING AREA         *
  22                        ;****************************************
  23  2000                            ORG   2000H
  24                        RESFLAG:  DS    1            ;RESET FLAG
  25                        TIMEFLAG: DS    1            ;TIME SET FLAG
  26                        PROGFLAG: DS    1            ;TIME PROGRAM FLAG
  27                        COUNT:    DS    1            ;TEMP.COUNTER
  28                        IDBYTE:   DS    1            ;ID.PROG.BYTE
  29                        OUTBUF:   DS    1            ;CH.OUTPUT BUFFER
  30                        OUTBIT:   DS    1            ;CH.OUTPUT BIT
  31                        CHCOUNT:  DS    1            ;CH.OUTPUT COUNT
  32
  33                        KEYCRRT:  DS    1            ;KEYBOARD CURRENT
  34                        KEYIND:   DS    2            ;KEYBOARD INDEX
  35                        KEYBUF:   DS    8            ;KEYBOARD BUFFER
  36
  37                        TIMEBUF:  DS    5            ;TIME/CHANNEL BUFF
  38                        DIGCRRT:  DS    1            ;DIGIT CURRENT
  39                        DISPBUF:  DS    8            ;DISPLAY BUFFER
  40
  41  2080                  TMPIND1   EQU   2080H        ;TEMP INDEX1
  42  2082                  TMPIND2   EQU   2082H        ;TEMP INDEX2
  43  2100                  ALARM1    EQU   2100H        ;PROG FOR CH-1
  44  21C8                  ALARM2    EQU   ALARM1+200   ;PROG FOR CH-2
  45  2290                  ALARM3    EQU   ALARM1+400   ;PROG FOR CH-3
  46  2358                  ALARM4    EQU   ALARM1+600   ;PROG FOR CH-4
  47  2420                  ALARM5    EQU   ALARM1+800   ;PROG FOR CH-5
  48  24E8                  ALARM6    EQU   ALARM1+1000  ;PROG FOR CH-6
  49  25B0                  ALARM7    EQU   ALARM1+1200  ;PROG FOR CH-7
  50  2678                  ALARM8    EQU   ALARM1+1400  ;PROG FOR CH-8
```

TURBO80. Z-80 Assembler Ver 1.00

```
Line  Addr  Obj

  51
  52  0000                          ORG   0000H
  53  0000  ED 56                   IM    1             ;INT MODE 1
  54  0002  31 FF 27                LD    SP,27FFH      ;SET STACK
  55  0005  3E 88           INIT:   LD    A,88H         ;CONTROL BYTE
  56  0007  D3 03                   OUT   (PPI+3),A
  57  0009  3A 00 20                LD    A,(RESFLAG);GET RESET FLAG
  58  000C  FE 55                   CP    55H
  59  000E  C2 2C 01                JP    NZ,COLDRES
  60  0011  C3 50 01                JP    WARMRES
  61                          ;*************************************
  62                          ;*      RTC INTERRUPT SERVICE        *
  63                          ;*************************************
  64  0038                          ORG   0038H
  65  0038  F5              INT:    PUSH  AF            ;SAVE AF
  66  0039  C5                      PUSH  BC            ;SAVE BC
  67  003A  D5                      PUSH  DE            ;SAVE DE
  68  003B  E5                      PUSH  HL            ;SAVE HL
  69  003C  DB 30                   IN    A,(RTC+10H);INT RESET
  70  003E  11 19 20                LD    DE,DISPBUF  ;DISPLAY INDEX
  71  0041  DB 24                   IN    A,(HOUR)    ;GET HOUR
  72  0043  32 15 20                LD    (TIMEBUF+2),A
  73  0046  CD 31 07                CALL  HCONV       ;SEGMENT CONV.
  74  0049  12                      LD    (DE),A
  75  004A  13                      INC   DE          ;NEXT DIGIT
  76  004B  DB 24                   IN    A,(HOUR)
  77  004D  CD 35 07                CALL  LCONV
  78  0050  F6 80                   OR    80H         ;WITH DOT
  79  0052  12                      LD    (DE),A
  80  0053  13                      INC   DE          ;NEXT DIGIT
  81  0054  DB 23                   IN    A,(MIN)     ;GET MINUTE
  82  0056  32 14 20                LD    (TIMEBUF+1),A
  83  0059  CD 31 07                CALL  HCONV       ;SEGMENT CONV.
  84  005C  12                      LD    (DE),A
  85  005D  13                      INC   DE          ;NEXT DIGIT
  86  005E  DB 23                   IN    A,(MIN)
  87  0060  CD 35 07                CALL  LCONV
  88  0063  F6 80                   OR    80H         ;WITH DOT
  89  0065  12                      LD    (DE),A
  90  0066  13                      INC   DE          ;NEXT DIGIT
  91  0067  DB 22                   IN    A,(SEC)     ;GET SECOND
  92  0069  32 13 20                LD    (TIMEBUF),A
  93  006C  CD 31 07                CALL  HCONV       ;SEGMENT CONV.
  94  006F  12                      LD    (DE),A
  95  0070  13                      INC   DE          ;NEXT DIGIT
  96  0071  DB 22                   IN    A,(SEC)
  97  0073  CD 35 07                CALL  LCONV
  98  0076  12                      LD    (DE),A
  99  0077  13                      INC   DE          ;NEXT DIGIT
 100  0078  DB 25                   IN    A,(DAY)     ;GET DAY
```

```
Line  Addr  Obj
 101  007A  32 16 20                    LD    (TIMEBUF+3),A
 102  007D  47                          LD    B,A
 103  007E  0E 00                       LD    C,00H
 104  0080  37                          SCF                 ;FIRST DAY
 105  0081  CB 11            DAYADJ:    RL    C
 106  0083  10 FC                       DJNZ  DAYADJ
 107  0085  79                          LD    A,C
 108  0086  12                          LD    (DE),A
 109
 110  0087  3E 01            CHANbit:   LD    A,01H          ;OUTPUT BIT
 111  0089  32 06 20                    LD    (OUTBIT),A
 112  008C  3E 08                       LD    A,8            ;8 O/P MAX
 113  008E  32 07 20                    LD    (CHCOUNT),A
 114  0091  21 00 21                    LD    HL,ALARM1      ;FIRST PROG.
 115  0094  3E 32            TIMECHK1:  LD    A,50           ;50 PROG.MAX
 116  0096  32 03 20                    LD    (COUNT),A
 117  0099  22 80 20                    LD    (TMPIND1),HL
 118  009C  22 82 20                    LD    (TMPIND2),HL
 119  009F  CD 9D 06                    CALL  SCDISP         ;SCAN DISPLAY
 120  00A2  2A 82 20            TIMECHK2:  LD    HL,(TMPIND2)
 121  00A5  7E                          LD    A,(HL)         ;GET ID.BYTE
 122  00A6  CB 7F                       BIT   7,A            ;PROG.BIT CHECK
 123  00A8  20 47                       JR    NZ,NOTEQ1
 124  00AA  7E                          LD    A,(HL)         ;GET DAY
 125  00AB  32 04 20                    LD    (IDBYTE),A     ;SAVE ID.BYTE
 126  00AE  E6 07                       AND   07H            ;USE BIT 0->2
•127  00B0  47                          LD    B,A
 128  00B1  3A 16 20                    LD    A,(TIMEBUF+3)
 129  00B4  B8                          CP    B              ;DAY COMPARE
 130  00B5  20 3A                       JR    NZ,NOTEQ1
 131  00B7  23                          INC   HL
 132  00B8  7E                          LD    A,(HL)         ;GET HOUR
 133  00B9  47                          LD    B,A
 134  00BA  3A 15 20                    LD    A,(TIMEBUF+2)
 135  00BD  B8                          CP    B              ;HOUR COMPARE
 136  00BE  20 31                       JR    NZ,NOTEQ1
 137  00C0  23                          INC   HL
 138  00C1  7E                          LD    A,(HL)         ;GET MIN.
 139  00C2  47                          LD    B,A
 140  00C3  3A 14 20                    LD    A,(TIMEBUF+1)
 141  00C6  B8                          CP    B              ;MIN.COMPARE
 142  00C7  20 28                       JR    NZ,NOTEQ1
 143  00C9  23                          INC   HL
 144  00CA  7E                          LD    A,(HL)         ;GET SEC.
 145  00CB  47                          LD    B,A
 146  00CC  3A 13 20                    LD    A,(TIMEBUF)
 147  00CF  B8                          CP    B              ;SEC.COMPARE
 148  00D0  20 1F                       JR    NZ,NOTEQ1
 149  00D2  3A 04 20            ALARMEQ:   LD    A,(IDBYTE)  ;GET ID.BYTE
 150  00D5  CB 77                       BIT   6,A
```

```
Line  Addr  Obj

 151  00D7  20 0B                        JR    NZ,SETON
 152  00D9  3A 06 20          SETZERO:   LD    A,(OUTBIT)
 153  00DC  2F                           CPL
 154  00DD  47                           LD    B,A
 155  00DE  3A 05 20                     LD    A,(OUTBUF)
 156  00E1  A0                           AND   B
 157  00E2  18 08                        JR    CMPSKIP1
 158  00E4  3A 06 20          SETON:     LD    A,(OUTBIT)
 159  00E7  47                           LD    B,A
 160  00E8  3A 05 20                     LD    A,(OUTBUF)
 161  00EB  B0                           OR    B
 162  00EC  32 20 20          CMPSKIP1:  LD    (DISPBUF+7),A
 163  00EF  D3 40                        OUT   (RELAY),A
 164  00F1  11 04 00          NOTEQ1:    LD    DE,4          ;BYTE COUNT
 165  00F4  2A 82 20                     LD    HL,(TMPIND2)
 166  00F7  19                           ADD   HL,DE
 167  00F8  22 82 20                     LD    (TMPIND2),HL
 168  00FB  3A 03 20                     LD    A,(COUNT)
 169  00FE  3D                           DEC   A
 170  00FF  32 03 20                     LD    (COUNT),A
 171  0102  C2 A2 00                     JP    NZ,TIMECHK2
 172
 173  0105  3A 06 20                     LD    A,(OUTBIT)
 174  0108  07                           RLCA                ;NEXT CHAN.
 175  0109  32 06 20                     LD    (OUTBIT),A
 176  010C  3E 32                        LD    A,50          ;50 PROG.MAX
 177  010E  32 03 20                     LD    (COUNT),A
 178  0111  11 C8 00                     LD    DE,200        ;PROG.LENGHT
 179  0114  2A 80 20                     LD    HL,(TMPIND1)
 180  0117  19                           ADD   HL,DE
 181  0118  22 80 20                     LD    (TMPIND1),HL
 182  011B  3A 07 20                     LD    A,(CHCOUNT)
 183  011E  3D                           DEC   A
 184  011F  32 07 20                     LD    (CHCOUNT),A
 185  0122  C2 94 00                     JP    NZ,TIMECHK1
 186  0125  E1                           POP   HL            ;RESTORE HL
 187  0126  D1                           POP   DE            ;RESTORE DE
 188  0127  C1                           POP   BC            ;RESTORE BC
 189  0128  F1                           POP   AF            ;RESTORE AF
 190  0129  FB                           EI                  ;RTC ENABLE
 191  012A  ED 4D                        RETI
 192                          ;*****************************************
 193                          ;*        BEFORE SET TIME RESET          *
 194                          ;*****************************************
 195  012C  3E 55             COLDRES:   LD    A,55H         ;RESET BYTE
 196  012E  32 00 20                     LD    (RESFLAG),A
 197  0131  01 40 06                     LD    BC,8*200      ;PROG.ARRAY
 198  0134  21 00 21                     LD    HL,ALARM1
 199  0137  36 00             FILLNXT:   LD    (HL),00H      ;CLEAR RAM
 200  0139  23                           INC   HL
```

```
Line  Addr  Obj

 201  013A  0B                        DEC   BC          ;DEC.COUNTER
 202  013B  78                        LD    A,B
 203  013C  B1                        OR    C
 204  013D  20 F8                     JR    NZ,FILLNXT
 205  013F  AF                        XOR   A
 206  0140  32 20 20                  LD    (DISPBUF+7),A
 207  0143  32 05 20                  LD    (OUTBUF),A
 208  0146  D3 40                     OUT   (RELAY),A   ;RELAY "OFF"
 209  0148  21 61 07                  LD    HL,MSSG1    ;DISPLAY "SET-L
 210  014B  CD 44 07                  CALL  SETMSSG
 211  014E  18 08                     JR    MAIN
 212                          ;*****************************************
 213                          ;*        AFTER SET TIME RESET           *
 214                          ;*****************************************
 215  0150  3A 05 20          WARMRES: LD    A,(OUTBUF) ;OUTPUT STAT
 216  0153  32 20 20                  LD    (DISPBUF+7),A
 217  0156  D3 40                     OUT   (RELAY),A
 218  0158  AF                MAIN:   XOR   A           ;FIRST DIGIT
 219  0159  32 18 20                  LD    (DIGCRRT),A
 220  015C  FB                        EI                ;RTC ENABLE
 221  015D  CD 9D 06          MAIN1:  CALL  SCDISP
 222  0160  CD 5A 06                  CALL  KEYREAD
 223  0163  FE 01                     CP    01H         ;TIME  KEY?
 224  0165  CA 7E 01                  JP    Z,TIMERUN
 225  0168  FE 02                     CP    02H         ;SET   KEY?
 226  016A  CA 8D 01                  JP    Z,TIMESET
 227  016D  FE 03                     CP    03H
 228  016F  CA 98 02                  JP    Z,TIMEPROG  ;PROG  KEY?
 229  0172  FE 04                     CP    04H
 230  0174  CA 9E 04                  JP    Z,TIMELIST  ;LIST  KEY?
 231  0177  FE 05                     CP    05H
 232  0179  CA B9 05                  JP    Z,DIRECT    ;DIRT  KEY?
 233  017C  18 DF                     JR    MAIN1
 234                          ;*****************************************
 235                          ;*        TIME RUN SUBMAIN               *
 236                          ;*****************************************
 237  017E  3A 01 20          TIMERUN: LD    A,(TIMEFLAG)
 238  0181  FE 55                     CP    55H         ;TIME SET FLAG
 239  0183  C2 5D 01                  JP    NZ,MAIN1
 240  0186  3E 04                     LD    A,04H       ;INT AT 1 SEC.
 241  0188  D3 31                     OUT   (RTC+11H),A
 242  018A  FB                        EI                ;RTC ENABLE
 243  018B  18 D0                     JR    MAIN1       ;GOTO MAIN
 244                          ;*****************************************
 245                          ;*        TIME SETING SUBMAIN            *
 246                          ;*****************************************
 247  018D  F3                TIMESET: DI                ;RTC DISABLE
 248  018E  21 67 07                  LD    HL,MSSG2    ;DISPLAY "DAY-"
 249  0191  CD 44 07                  CALL  SETMSSG
 250  0194  3A 01 20                  LD    A,(TIMEFLAG)
```

```
Line  Addr  Obj

 251  0197  FE 55                    CP   55H          ;TIME SET FLAG
 252  0199  28 04                    JR   Z,DAYSKIP1
 253  019B  3E 01                    LD   A,01H        ;"SUN" FIRST
 254  019D  18 02                    JR   DAYSKIP2
 255  019F  DB 25           DAYSKIP1: IN   A,(RTC+5)   ;GET DAY
 256  01A1  32 16 20        DAYSKIP2: LD   (TIMEBUF+3),A
 257  01A4  CD E9 06                  CALL DAYCONV      ;DAY TO DISPLAY
 258                        ;*****************************************
 259                        ;*        DAY OF THE WEEK SETTING        *
 260                        ;*****************************************
 261  01A7  CD 9D 06        DAYSET1:  CALL SCDISP       ;SCAN DISPLAY
 262  01AA  CD 5A 06                  CALL KEYREAD
 263  01AD  FE 01                     CP   01H          ;TIME KEY?
 264  01AF  CA 95 02                  JP   Z,SETEND
 265  01B2  FE 02                     CP   02H          ;SET KEY?
 266  01B4  28 2E                     JR   Z,TOHRSET
 267  01B6  FE 07                     CP   07H          ;+ KEY?
 268  01B8  28 06                     JR   Z,DAYINC
 269  01BA  FE 08                     CP   08H          ;- KEY?
 270  01BC  28 14                     JR   Z,DAYDEC
 271  01BE  18 E7                     JR   DAYSET1
 272  01C0  3A 16 20        DAYINC:   LD   A,(TIMEBUF+3)
 273  01C3  3C                        INC  A
 274  01C4  FE 08                     CP   08H          ;LAST DAY+1?
 275  01C6  20 02                     JR   NZ,DAYINC1
 276  01C8  3E 01                     LD   A,01H
 277  01CA  32 16 20        DAYINC1:  LD   (TIMEBUF+3),A
 278  01CD  CD E9 06                  CALL DAYCONV
 279  01D0  18 D5                     JR   DAYSET1
 280  01D2  3A 16 20        DAYDEC:   LD   A,(TIMEBUF+3)
 281  01D5  3D                        DEC  A
 282  01D6  FE 00                     CP   00H          ;FIRST DAY-1?
 283  01D8  20 02                     JR   NZ,DAYDEC1
 284  01DA  3E 07                     LD   A,07H
 285  01DC  32 16 20        DAYDEC1:  LD   (TIMEBUF+3),A
 286  01DF  CD E9 06                  CALL DAYCONV
 287  01E2  18 C3                     JR   DAYSET1
 288                        ;*****************************************
 289                        ;*        HOUR SETTING                   *
 290                        ;*****************************************
 291  01E4  AF              TOHRSET:  XOR  A            ;MIN,SEC CLEAR
 292  01E5  32 1B 20                  LD   (DISPBUF+2),A
 293  01E8  32 1C 20                  LD   (DISPBUF+3),A
 294  01EB  32 1D 20                  LD   (DISPBUF+4),A
 295  01EE  32 1E 20                  LD   (DISPBUF+5),A
 296  01F1  3A 01 20                  LD   A,(TIMEFLAG)
 297  01F4  FE 55                     CP   55H          ;TIME SET FLAG
 298  01F6  28 04                     JR   Z,HRSKIP1
 299  01F8  3E 12                     LD   A,12H        ;HOUR=12
 300  01FA  18 02                     JR   HRSKIP2
```

```
Line  Addr  Obj

 301  01FC  DB 24          HRSKIP1:  IN    A,(RTC+4)   ;GET HOUR
 302  01FE  32 15 20       HRSKIP2:  LD    (TIMEBUF+2),A
 303  0201  CD F7 06                 CALL  HRCONV      ;HOUR TO DISPLAY
 304  0204  CD 9D 06       HRSET1:   CALL  SCDISP      ;SCAN DISPLAY
 305  0207  CD 5A 06                 CALL  KEYREAD
 306  020A  FE 01                    CP    01H         ;TIME KEY?
 307  020C  CA 95 02                 JP    Z,SETEND
 308  020F  FE 02                    CP    02H         ;SET KEY?
 309  0211  28 30                    JR    Z,TOMINSET
 310  0213  FE 07                    CP    07H         ;+ KEY?
 311  0215  28 06                    JR    Z,HRINC
 312  0217  FE 08                    CP    08H         ;- KEY?
 313  0219  28 15                    JR    Z,HRDEC
 314  021B  18 E7                    JR    HRSET1
 315  021D  3A 15 20       HRINC:    LD    A,(TIMEBUF+2)
 316  0220  3C                       INC   A
 317  0221  27                       DAA               ;TO DECIMAL
 318  0222  FE 25                    CP    25H         ;LAST HOUR+1?
 319  0224  20 02                    JR    NZ,HRINC1
 320  0226  3E 01                    LD    A,01H
 321  0228  32 15 20       HRINC1:   LD    (TIMEBUF+2),A
 322  022B  CD F7 06                 CALL  HRCONV
 323  022E  18 D4                    JR    HRSET1
 324  0230  3A 15 20       HRDEC:    LD    A,(TIMEBUF+2)
 325  0233  3D                       DEC   A
 326  0234  27                       DAA               ;TO DECIMAL
 327  0235  FE 00                    CP    00H         ;FIRST HOUR-1?
 328  0237  20 02                    JR    NZ,HRDEC1
 329  0239  3E 24                    LD    A,24H
 330  023B  32 15 20       HRDEC1:   LD    (TIMEBUF+2),A
 331  023E  CD F7 06                 CALL  HRCONV
 332  0241  18 C1                    JR    HRSET1
 333                       ;*****************************************
 334                       ;*        MINUTE SETTING                 *
 335                       ;*****************************************
 336  0243  3A 01 20       TOMINSET: LD    A,(TIMEFLAG)
 337  0246  FE 55                    CP    55H         ;TIME SET FLAG
 338  0248  28 04                    JR    Z,MINSKIP1
 339  024A  3E 34                    LD    A,34H       ;MIN=34
 340  024C  18 02                    JR    MINSKIP2
 341  024E  DB 23          MINSKIP1: IN    A,(RTC+3)   ;GET MINUTE
 342  0250  32 14 20       MINSKIP2: LD    (TIMEBUF+1),A
 343  0253  CD 0B 07                 CALL  MINCONV     ;MIN TO DISPLAY
 344  0256  CD 9D 06       MINSET1:  CALL  SCDISP      ;SCAN DISPLAY
 345  0259  CD 5A 06                 CALL  KEYREAD
 346  025C  FE 01                    CP    01H         ;TIME KEY?
 347  025E  CA 95 02                 JP    Z,SETEND
 348  0261  FE 02                    CP    02H         ;SET KEY?
 349  0263  28 30                    JR    Z,SETEND
 350  0265  FE 07                    CP    07H         ;+ KEY?
```

```
Line  Addr  Obj

 351  0267  28 06                     JR    Z,MININC
 352  0269  FE 08                     CP    08H          ;- KEY?
 353  026B  28 15                     JR    Z,MINDEC
 354  026D  18 E7                     JR    MINSET1
 355  026F  3A 14 20          MININC: LD    A,(TIMEBUF+1)
 356  0272  3C                        INC   A
 357  0273  27                        DAA                ;TO DECIMAL
 358  0274  FE 60                     CP    60H          ;LAST MIN+1?
 359  0276  20 02                     JR    NZ,MININC1
 360  0278  3E 00                     LD    A,00H
 361  027A  32 14 20          MININC1: LD   (TIMEBUF+1),A
 362  027D  CD 0B 07                  CALL  MINCONV
 363  0280  18 D4                     JR    MINSET1
 364  0282  3A 14 20          MINDEC: LD    A,(TIMEBUF+1)
 365  0285  3D                        DEC   A
 366  0286  27                        DAA                ;TO DECIMAL
 367  0287  FE 99                     CP    99H          ;FIRST MIN-1?
 368  0289  20 02                     JR    NZ,MINDEC1
 369  028B  3E 59                     LD    A,59H
 370  028D  32 14 20          MINDEC1: LD   (TIMEBUF+1),A
 371  0290  CD 0B 07                  CALL  MINCONV
 372  0293  18 C1                     JR    MINSET1
 373  0295  C3 3F 06          SETEND: JP    TIMERTC
 374                          ;*****************************************
 375                          ;*        PROGRAM TIME SUBMAIN           *
 376                          ;*****************************************
 377  0298  F3                TIMEPROG: DI                ;RTC DISABLE
 378  0299  21 6D 07                  LD    HL,MSSG3     ;DISPLAY "CH-"
 379  029C  CD 44 07                  CALL  SETMSSG
 380  029F  AF                        XOR   A            ;DAY CLEAR
 381  02A0  32 1F 20                  LD    (DISPBUF+6),A
 382  02A3  3E 01                     LD    A,01H        ;"CH-" FIRST
 383  02A5  32 17 20                  LD    (TIMEBUF+4),A
 384  02A8  CD D2 06                  CALL  CHANCONV     ;CHAN.TO DISPLAY
 385  02AB  11 C8 00                  LD    DE,0200      ;PROG LENGHT
 386  02AE  21 00 21                  LD    HL,ALARM1    ;FIRST PROG.
 387  02B1  22 80 20                  LD    (TMPIND1),HL
 388                          ;*****************************************
 389                          ;*        OUTPUT CHANNEL SETTING         *
 390                          ;*****************************************
 391  02B4  CD 9D 06          PRGCHSET1:CALL SCDISP      ;SCAN DISPLAY
 392  02B7  CD 5A 06                  CALL  KEYREAD
 393  02BA  FE 01                     CP    01H          ;TIME KEY?
 394  02BC  CA 94 04                  JP    Z,PROGEND
 395  02BF  FE 03                     CP    03H          ;PROG KEY?
 396  02C1  28 3E                     JR    Z,PRDAYSET
 397  02C3  FE 07                     CP    07H          ;+ KEY?
 398  02C5  28 06                     JR    Z,CHANINC
 399  02C7  FE 08                     CP    08H          ;- KEY?
 400  02C9  28 1F                     JR    Z,CHANDEC
```

```
Line  Addr  Obj

 401  02CB  18 E7                    JR    PRGCHSET1
 402  02CD  3A 17 20       CHANINC:  LD    A,(TIMEBUF+4)
 403  02D0  3C                       INC   A
 404  02D1  19                       ADD   HL,DE
 405  02D2  FE 09                    CP    09H          ;LAST CH.+1?
 406  02D4  20 05                    JR    NZ,CHANINC1
 407  02D6  3E 01                    LD    A,01H
 408  02D8  21 00 21                 LD    HL,ALARM1    ;FIRST PROG.
 409  02DB  22 80 20       CHANINC1: LD    (TMPIND1),HL
 410  02DE  32 17 20                 LD    (TIMEBUF+4),A
 411  02E1  CD D2 06                 CALL  CHANCONV
 412  02E4  18 CE                    JR    PRGCHSET1
 413  02E6  3A 17 20       CHANDEC:  LD    A,(TIMEBUF+4)
 414  02E9  3D                       DEC   A
 415  02EA  B7                       OR    A            ;CARRY=0
 416  02EB  ED 52                    SBC   HL,DE
 417  02ED  FE 00                    CP    00H          ;FIRST CH.-1?
 418  02EF  20 05                    JR    NZ,CHANDEC1
 419  02F1  3E 08                    LD    A,08H
 420  02F3  21 78 26                 LD    HL,ALARM8    ;LAST PROG.
 421  02F6  22 80 20       CHANDEC1: LD    (TMPIND1),HL
 422  02F9  32 17 20                 LD    (TIMEBUF+4),A
 423  02FC  CD D2 06                 CALL  CHANCONV
 424  02FF  18 B3                    JR    PRGCHSET1
 425                                ;**************************************
 426                                ;*          PROGRAM DAY SETING        *
 427                                ;**************************************
 428  0301  21 67 07       PRDAYSET: LD    HL,MSSG2     ;DISPLAY "DAY-"
 429  0304  CD 44 07                 CALL  SETMSSG
 430  0307  3E 01                    LD    A,01H        ;"SUN" FIRST
 431  0309  32 16 20                 LD    (TIMEBUF+3),A
 432  030C  CD E9 06                 CALL  DAYCONV      ;DAY TO DISPLAY
 433  030F  CD 9D 06       PRDAYSET1:CALL  SCDISP       ;SCAN DISPLAY
 434  0312  CD 5A 06                 CALL  KEYREAD
 435  0315  FE 01                    CP    01H          ;TIME KEY?
 436  0317  CA 94 04                 JP    Z,PROGEND
 437  031A  FE 03                    CP    03H          ;PROG KEY?
 438  031C  28 2E                    JR    Z,PRHRSET
 439  031E  FE 07                    CP    07H          ;+ KEY?
 440  0320  28 06                    JR    Z,PRDAYINC
 441  0322  FE 08                    CP    08H          ;- KEY?
 442  0324  28 14                    JR    Z,PRDAYDEC
 443  0326  18 E7                    JR    PRDAYSET1
 444  0328  3A 16 20       PRDAYINC: LD    A,(TIMEBUF+3)
 445  032B  3C                       INC   A
 446  032C  FE 08                    CP    08H          ;LAST DAY+1?
 447  032E  20 02                    JR    NZ,PRDAYINC1
 448  0330  3E 01                    LD    A,01H
 449  0332  32 16 20       PRDAYINC1:LD    (TIMEBUF+3),A
 450  0335  CD E9 06                 CALL  DAYCONV
```

```
Line  Addr  Obj

 451  0338  18 D5                    JR    PRDAYSET1
 452  033A  3A 16 20       PRDAYDEC: LD    A,(TIMEBUF+3)
 453  033D  3D                      DEC   A
 454  033E  FE 00                   CP    00H         ;FIRST DAY-1?
 455  0340  20 02                   JR    NZ,PRDAYDEC1
 456  0342  3E 07                   LD    A,07H
 457  0344  32 16 20       PRDAYDEC1:LD    (TIMEBUF+3),A
 458  0347  CD E9 06                CALL  DAYCONV
 459  034A  18 C3                   JR    PRDAYSET1
 460                        ;*****************************************
 461                        ;*          PROGRAM HOUR SETTING         *
 462                        ;*****************************************
 463  034C  AF              PRHRSET:  XOR   A           ;MIN,SEC CLEAR
 464  034D  32 1B 20                LD    (DISPBUF+2),A
 465  0350  32 1C 20                LD    (DISPBUF+3),A
 466  0353  32 1D 20                LD    (DISPBUF+4),A
 467  0356  32 1E 20                LD    (DISPBUF+5),A
 468  0359  3A 01 20                LD    A,(TIMEFLAG)
 469  035C  3E 12                   LD    A,12H       ;HOUR=12
 470  035E  32 15 20                LD    (TIMEBUF+2),A
 471  0361  CD F7 06                CALL  HRCONV      ;HOUR TO DISPLAY
 472  0364  CD 9D 06       PRHRSET1: CALL  SCDISP    ;SCAN DISPLAY
 473  0367  CD 5A 06                CALL  KEYREAD
 474  036A  FE 01                   CP    01H         ;TIME KEY?
 475  036C  CA 94 04                JP    Z,PROGEND
 476  036F  FE 03                   CP    03H         ;PROG KEY?
 477  0371  28 30                   JR    Z,PRMINSET
 478  0373  FE 07                   CP    07H         ;+ KEY?
 479  0375  28 06                   JR    Z,PRHRINC
 480  0377  FE 08                   CP    08H         ;- KEY?
 481  0379  28 15                   JR    Z,PRHRDEC
 482  037B  18 E7                   JR    PRHRSET1
 483  037D  3A 15 20       PRHRINC:  LD    A,(TIMEBUF+2)
 484  0380  3C                      INC   A
 485  0381  27                      DAA               ;TO DECIMAL
 486  0382  FE 25                   CP    25H         ;LAST HOUR+1?
 487  0384  20 02                   JR    NZ,PRHRINC1
 488  0386  3E 01                   LD    A,01H
 489  0388  32 15 20       PRHRINC1: LD    (TIMEBUF+2),A
 490  038B  CD F7 06                CALL  HRCONV
 491  038E  18 D4                   JR    PRHRSET1
 492  0390  3A 15 20       PRHRDEC:  LD    A,(TIMEBUF+2)
 493  0393  3D                      DEC   A
 494  0394  27                      DAA               ;TO DECIMAL
 495  0395  FE 00                   CP    00H         ;FIRST HOUR-1?
 496  0397  20 02                   JR    NZ,PRHRDEC1
 497  0399  3E 24                   LD    A,24H
 498  039B  32 15 20       PRHRDEC1: LD    (TIMEBUF+2),A
 499  039E  CD F7 06                CALL  HRCONV
 500  03A1  18 C1                   JR    PRHRSET1
```

```
Line  Addr  Obj

 501                        ;****************************************
 502                        ;*        PROGRAM MINUTE SETTING        *
 503                        ;****************************************
 504  03A3  3E 34           PRMINSET: LD   A,34H       ;MIN=34
 505  03A5  32 14 20                  LD   (TIMEBUF+1),A
 506  03A8  CD 0B 07                  CALL MINCONV     ;MIN TO DISPLAY
 507  03AB  CD 9D 06          PRMINSET1:CALL SCDISP     ;SCAN DISPLAY
 508  03AE  CD 5A 06                  CALL KEYREAD
 509  03B1  FE 01                     CP   01H         ;TIME KEY?
 510  03B3  CA 94 04                  JP   Z,PROGEND
 511  03B6  FE 03                     CP   03H         ;PROG KEY?
 512  03B8  28 30                     JR   Z,PRSECSET
 513  03BA  FE 07                     CP   07H         ;+ KEY?
 514  03BC  28 06                     JR   Z,PRMININC
 515  03BE  FE 08                     CP   08H         ;- KEY?
 516  03C0  28 15                     JR   Z,PRMINDEC
 517  03C2  18 E7                     JR   PRMINSET1
 518  03C4  3A 14 20          PRMININC: LD   A,(TIMEBUF+1)
 519  03C7  3C                        INC  A
 520  03C8  27                        DAA              ;TO DECIMAL
 521  03C9  FE 60                     CP   60H         ;LAST MIN+1?
 522  03CB  20 02                     JR   NZ,PRMININC1
 523  03CD  3E 00                     LD   A,00H
 524  03CF  32 14 20          PRMININC1:LD   (TIMEBUF+1),A
 525  03D2  CD 0B 07                  CALL MINCONV
 526  03D5  18 D4                     JR   PRMINSET1
 527  03D7  3A 14 20          PRMINDEC: LD   A,(TIMEBUF+1)
 528  03DA  3D                        DEC  A
 529  03DB  27                        DAA              ;TO DECIMAL
 530  03DC  FE 99                     CP   99H         ;FIRST MIN-1?
 531  03DE  20 02                     JR   NZ,PRMINDEC1
 532  03E0  3E 59                     LD   A,59H
 533  03E2  32 14 20          PRMINDEC1:LD   (TIMEBUF+1),A
 534  03E5  CD 0B 07                  CALL MINCONV
 535  03E8  18 C1                     JR   PRMINSET1
 536                        ;****************************************
 537                        ;*        PROGRAM SECOND SETTING        *
 538                        ;****************************************
 539  03EA  AF              PRSECSET: XOR  A           ;SEC=00
 540  03EB  32 13 20                  LD   (TIMEBUF),A
 541  03EE  CD 1F 07                  CALL SECCONV     ;SEC TO DISPLAY
 542  03F1  CD 9D 06          PRSECSET1:CALL SCDISP     ;SCAN DISPLAY
 543  03F4  CD 5A 06                  CALL KEYREAD
 544  03F7  FE 01                     CP   01H         ;TIME KEY?
 545  03F9  CA 94 04                  JP   Z,PROGEND
 546  03FC  FE 03                     CP   03H         ;PROG KEY?
 547  03FE  CA 31 04                  JP   Z,OPSTAT
 548  0401  FE 07                     CP   07H         ;+ KEY?
 549  0403  28 06                     JR   Z,PRSECINC
 550  0405  FE 08                     CP   08H         ;- KEY?
```

```
Line   Addr  Obj

 551   0407  28 15                    JR    Z,PRSECDEC
 552   0409  18 E6                    JR    PRSECSET1
 553   040B  3A 13 20       PRSECINC: LD    A,(TIMEBUF)
 554   040E  3C                       INC   A
 555   040F  27                       DAA                 ;TO DECIMAL
 556   0410  FE 60                    CP    60H           ;LAST SEC+1?
 557   0412  20 02                    JR    NZ,PRSECINC1
 558   0414  3E 00                    LD    A,00H
 559   0416  32 13 20      PRSECINC1:LD    (TIMEBUF),A
 560   0419  CD 1F 07                 CALL  SECCONV
 561   041C  18 D3                    JR    PRSECSET1
 562   041E  3A 13 20       PRSECDEC: LD    A,(TIMEBUF)
 563   0421  3D                       DEC   A
 564   0422  27                       DAA                 ;TO DECIMAL
 565   0423  FE 99                    CP    99H           ;FIRST SEC-1?
 566   0425  20 02                    JR    NZ,PRSECDEC1
 567   0427  3E 59                    LD    A,59H
 568   0429  32 13 20      PRSECDEC1:LD    (TIMEBUF),A
 569   042C  CD 1F 07                 CALL  SECCONV
 570   042F  18 C0                    JR    PRSECSET1
 571                       ;*****************************************
 572                       ;*          PROGRAM ON/OFF SETTING       *
 573                       ;*****************************************
 574   0431  0E 40         OPSTAT:   LD    C,40H         ;OFF STATE
 575   0433  21 73 07                 LD    HL,MSSG4      ;DISPLAY "ON"
 576   0436  CD 44 07                 CALL  SETMSSG
 577   0439  AF                       XOR   A
 578   043A  32 1D 20                 LD    (DISPBUF+4),A
 579   043D  32 1E 20                 LD    (DISPBUF+5),A
 580   0440  32 1F 20                 LD    (DISPBUF+6),A
 581   0443  CD 9D 06       OPSTAT1:  CALL  SCDISP        ;SCAN DISPLAY
 582   0446  CD 5A 06                 CALL  KEYREAD
 583   0449  FE 01                    CP    01H           ;TIME KEY?
 584   044B  CA 94 04                 JP    Z,PROGEND
 585   044E  FE 03                    CP    03H           ;PROG KEY?
 586   0450  28 1E                    JR    Z,PRENTER
 587   0452  FE 07                    CP    07H           ;+ KEY?
 588   0454  28 06                    JR    Z,SETOPON
 589   0456  FE 08                    CP    08H           ;- KEY?
 590   0458  28 0C                    JR    Z,SETOPOFF
 591   045A  18 E7                    JR    OPSTAT1
 592   045C  CB F1          SETOPON:  SET   6,C           ;SET O/P "ON"
 593   045E  21 73 07                 LD    HL,MSSG4      ;DISPLAY "ON"
 594   0461  CD 44 07                 CALL  SETMSSG
 595   0464  18 DD                    JR    OPSTAT1
 596   0466  CB B1          SETOPOFF: RES   6,C           ;SET O/P "OFF"
 597   0468  21 79 07                 LD    HL,MSSG5      ;DISPLAY "OFF"
 598   046B  CD 44 07                 CALL  SETMSSG
 599   046E  18 D3                    JR    OPSTAT1
 600   0470  11 04 00       PRENTER:  LD    DE,4          ;BYTE LENGHT
```

```
Line  Addr  Obj

 601  0473  2A 80 20                  LD   HL,(TMPIND1)
 602  0476  7E              PRENTNXT: LD   A,(HL)      ;GET ID.BIT
 603  0477  CB 7F                     BIT  7,A         ;EMTRY LOCATE?
 604  0479  28 03                     JR   Z,PRENTER2
 605  047B  19                        ADD  HL,DE       ;INDEX ADJUST
 606  047C  18 F8                     JR   PRENTNXT
 607  047E  3A 16 20          PRENTER2: LD   A,(TIMEBUF+3)
 608  0481  F6 80                     OR   80H         ;PROG.BIT
 609  0483  B1                        OR   C
 610  0484  77                        LD   (HL),A      ;SAVE OP/DAY
 611  0485  23                        INC  HL
 612  0486  3A 15 20                  LD   A,(TIMEBUF+2)
 613  0489  77                        LD   (HL),A      ;SAVE HOUR
 614  048A  23                        INC  HL
 615  048B  3A 14 20                  LD   A,(TIMEBUF+1)
 616  048E  77                        LD   (HL),A      ;SAVE MIN.
 617  048F  23                        INC  HL
 618  0490  3A 13 20                  LD   A,(TIMEBUF)
 619  0493  77                        LD   (HL),A      ;SAVE SEC.
 620  0494  3A 05 20          PROGEND: LD   A,(OUTBUF)  ;OUTPUT STAT
 621  0497  32 20 20                  LD   (DISPBUF+7),A
 622  049A  FB                        EI               ;RTC ENABLE
 623  049B  C3 5D 01                  JP   MAIN1
 624                        ;*************************************
 625                        ;*        PROGRAM TIME LISTING       *
 626                        ;*************************************
 627  049E  F3              TIMELIST: DI               ;RTC DISABLE
 628  049F  21 6D 07                  LD   HL,MSSG3    ;DISPLAY "CH-"
 629  04A2  CD 44 07                  CALL SETMSSG
 630  04A5  AF                        XOR  A           ;DAY CLEAR
 631  04A6  32 1F 20                  LD   (DISPBUF+6),A
 632  04A9  3E 01                     LD   A,01H       ;"CH-" FIRST
 633  04AB  32 17 20                  LD   (TIMEBUF+4),A
 634  04AE  11 C8 00                  LD   DE,200      ;PROG LENGHT
 635  04B1  21 00 21                  LD   HL,ALARM1   ;FIRST PROG.
 636  04B4  22 80 20                  LD   (TMPIND1),HL
 637  04B7  CD D2 06                  CALL CHANCONV    ;CHAN.TO DISPLAY
 638                        ;*************************************
 639                        ;*        OUTPUT CHANNEL SETTING     *
 640                        ;*************************************
 641  04BA  CD 9D 06          CHLIST1: CALL SCDISP      ;SCAN DISPLAY
 642  04BD  CD 5A 06                   CALL KEYREAD
 643  04C0  FE 01                      CP   01H         ;TIME KEY?
 644  04C2  CA 92 05                   JP   Z,LISTEND
 645  04C5  FE 04                      CP   04H         ;LIST KEY?
 646  04C7  28 3E                      JR   Z,TIMELIST1
 647  04C9  FE 07                      CP   07H         ;+ KEY?
 648  04CB  28 06                      JR   Z,LISTINC
 649  04CD  FE 08                      CP   08H         ;- KEY?
 650  04CF  28 1B                      JR   Z,LISTDEC
```

```
Line  Addr  Obj

 651  04D1  18 E7              JR    CHLIST1
 652  04D3  3A 17 20  LISTINC: LD    A,(TIMEBUF+4)
 653  04D6  3C                 INC   A
 654  04D7  19                 ADD   HL,DE
 655  04D8  FE 09              CP    09H          ;LAST CH.+1?
 656  04DA  20 05              JR    NZ,LISTINC1
 657  04DC  3E 01              LD    A,01H
 658  04DE  21 00 21           LD    HL,ALARM1    ;FIRST PROG.
 659  04E1  22 80 20  LISTINC1: LD   (TMPIND1),HL
 660  04E4  32 17 20           LD    (TIMEBUF+4),A
 661  04E7  CD D2 06           CALL  CHANCONV
 662  04EA  18 CE              JR    CHLIST1
 663  04EC  3A 17 20  LISTDEC: LD    A,(TIMEBUF+4)
 664  04EF  3D                 DEC   A
 665  04F0  B7                 OR    A            ;CARRY=0
 666  04F1  ED 52              SBC   HL,DE
 667  04F3  FE 00              CP    00H          ;FIRST CH.-1?
 668  04F5  20 05              JR    NZ,LISTDEC1
 669  04F7  3E 08              LD    A,08H
 670  04F9  21 78 26           LD    HL,ALARM8    ;LAST PROG.
 671  04FC  22 80 20  LISTDEC1: LD   (TMPIND1),HL
 672  04FF  32 17 20           LD    (TIMEBUF+4),A
 673  0502  CD D2 06           CALL  CHANCONV
 674  0505  18 B3              JR    CHLIST1
 675  0507  3E 32  TIMELIST1:LD    A,50         ;50 PROG.MAX
 676  0509  32 03 20           LD    (COUNT),A
 677  050C  11 04 00           LD    DE,4         ;BYTE LENGHT
 678  050F  3A 03 20  TLISTNXT: LD   A,(COUNT)
 679  0512  47                 LD    B,A
 680  0513  2A 80 20  TLISTNXT1:LD   HL,(TMPIND1)
 681  0516  22 82 20           LD    (TMPIND2),HL
 682  0519  7E                 LD    A,(HL)       ;GET ID.BIT
 683  051A  CB 7F              BIT   7,A          ;EMTRY BYTE?
 684  051C  20 08              JR    NZ,TIMELIST2
 685  051E  19                 ADD   HL,DE        ;INDEX ADJUST
 686  051F  22 80 20           LD    (TMPIND1),HL
 687  0522  10 EF              DJNZ  TLISTNXT1
 688  0524  18 6C              JR    LISTEND
 689  0526  78        TIMELIST2:LD   A,B          ;SAVE COUNTER
 690  0527  32 03 20           LD    (COUNT),A
 691  052A  7E                 LD    A,(HL)       ;GET OP/DAY
 692  052B  CB 77              BIT   6,A          ;OP STAT CHECK
 693  052D  C2 33 05           JP    NZ,CHSETON
 694  0530  AF                 XOR   A
 695  0531  18 06              JR    LISTSKIP
 696  0533  3A 17 20  CHSETON: LD    A,(TIMEBUF+4)
 697  0536  CD DB 06           CALL  CHANCONV1
 698  0539  32 20 20  LISTSKIP: LD   (DISPBUF+7),A
 699  053C  7E                 LD    A,(HL)       ;GET DAY
 700  053D  E6 07              AND   07H          ;USE BIT0->2
```

```
Line  Addr  Obj

 701  053F  CD EC 06                    CALL  DAYCONV1
 702  0542  23                         INC   HL
 703  0543  7E                         LD    A,(HL)      ;GET HOUR
 704  0544  CD FA 06                    CALL  HRCONV1
 705  0547  23                         INC   HL
 706  0548  7E                         LD    A,(HL)      ;GET MIN.
 707  0549  CD 0E 07                    CALL  MINCONV1
 708  054C  23                         INC   HL
 709  054D  7E                         LD    A,(HL)      ;GET SEC.
 710  054E  CD 22 07                    CALL  SECCONV1
 711  0551  23                         INC   HL
 712  0552  22 80 20                    LD    (TMPIND1),HL
 713  0555  CD 9D 06          LISTWAIT: CALL  SCDISP      ;SCAN DISPLAY
 714  0558  CD 5A 06                    CALL  KEYREAD
 715  055B  FE 01                      CP    01H         ;TIME KEY?
 716  055D  28 33                      JR    Z,LISTEND
 717  055F  FE 04                      CP    04H         ;LIST KEY?
 718  0561  28 AC                      JR    Z,TLISTNXT
 719  0563  FE 06                      CP    06H         ;CLEAR
 720  0565  28 02                      JR    Z,PROGCLR
 721  0567  18 EC                      JR    LISTWAIT
 722  0569  21 7F 07          PROGCLR:  LD    HL,MSSG6    ;DISPLAY "ConF"
 723  056C  CD 44 07                    CALL  SETMSSG
 724  056F  AF                         XOR   A
 725  0570  32 1F 20                    LD    (DISPBUF+6),A
 726  0573  32 20 20                    LD    (DISPBUF+7),A
 727  0576  CD 9D 06          CLRCONF:  CALL  SCDISP      ;SCAN DISPLAY
 728  0579  CD 5A 06                    CALL  KEYREAD
 729  057C  FE 01                      CP    01H         ;TIME KEY?
 730  057E  28 12                      JR    Z,LISTEND
 731  0580  FE 04                      CP    04H         ;LIST KEY?
 732  0582  28 8B                      JR    Z,TLISTNXT
 733  0584  FE 06                      CP    06H         ;CLEAR KEY?
 734  0586  28 02                      JR    Z,CLRPROG
 735  0588  18 EC                      JR    CLRCONF
 736  058A  2A 82 20          CLRPROG:  LD    HL,(TMPIND2)
 737  058D  CB BE                      RES   7,(HL)      ;CLEAR BIT
 738  058F  C3 0F 05                    JP    TLISTNXT
 739  0592  21 85 07          LISTEND:  LD    HL,MSSG7    ;"------"
 740  0595  CD 44 07                    CALL  SETMSSG
 741  0598  AF                         XOR   A
 742  0599  32 1F 20                    LD    (DISPBUF+6),A
 743  059C  32 20 20                    LD    (DISPBUF+7),A
 744  059F  CD 9D 06          LISTEND1: CALL  SCDISP      ;SCAN DISPLAY
 745  05A2  CD 5A 06                    CALL  KEYREAD
 746  05A5  FE 01                      CP    01H         ;TIME KEY?
 747  05A7  28 06                      JR    Z,LISTEND2
 748  05A9  FE 04                      CP    04H         ;LIST KEY?
 749  05AB  28 02                      JR    Z,LISTEND2
 750  05AD  18 F0                      JR    LISTEND1
```

```
Line  Addr  Obj

 751  05AF  3A 05 20        LISTEND2: LD   A,(OUTBUF)
 752  05B2  32 20 20                  LD   (DISPBUF+7),A
 753  05B5  FB                        EI                 ;RTC ENABLE
 754  05B6  C3 5D 01                  JP   MAIN1
 755                         ;***************************************
 756                         ;*         SET DIRECT OUTPUT           *
 757                         ;***************************************
 758  05B9  F3              DIRECT:   DI                 ;RTC DISABLE
 759  05BA  21 6D 07                  LD   HL,MSSG3      ;DISPLAY "CH-"
 760  05BD  CD 44 07                  CALL SETMSSG
 761  05C0  AF                        XOR  A             ;DAY CLEAR
 762  05C1  32 1F 20                  LD   (DISPBUF+6),A
 763  05C4  3E 01                     LD   A,01H         ;"CH-" FIRST
 764  05C6  32 17 20                  LD   (TIMEBUF+4),A
 765  05C9  CD D2 06                  CALL CHANCONV      ;CHAN.TO DISPLAY
 766  05CC  CD 9D 06        DRTCH1:   CALL SCDISP        ;SCAN DISPLAY
 767  05CF  CD 5A 06                  CALL KEYREAD
 768  05D2  FE 01                     CP   01H           ;TIME KEY?
 769  05D4  28 5F                     JR   Z,DRTEND
 770  05D6  FE 05                     CP   05H           ;DIRT KEY?
 771  05D8  28 2E                     JR   Z,DRTOUT1
 772  05DA  FE 07                     CP   07H           ;+ KEY?
 773  05DC  28 06                     JR   Z,DRTCHINC
 774  05DE  FE 08                     CP   08H           ;- KEY?
 775  05E0  28 14                     JR   Z,DRTCHDEC
 776  05E2  18 E8                     JR   DRTCH1
 777  05E4  3A 17 20        DRTCHINC: LD   A,(TIMEBUF+4)
 778  05E7  3C                        INC  A
 779  05E8  FE 09                     CP   09H           ;LAST CH.+1?
 780  05EA  20 02                     JR   NZ,DRTCHINC1
 781  05EC  3E 01                     LD   A,01H
 782  05EE  32 17 20        DRTCHINC1:LD   (TIMEBUF+4).A
 783  05F1  CD D2 06                  CALL CHANCONV
 784  05F4  18 D6                     JR   DRTCH1
 785  05F6  3A 17 20        DRTCHDEC: LD   A,(TIMEBUF+4)
 786  05F9  3D                        DEC  A
 787  05FA  FE 00                     CP   00H           ;FIRST CH.-1?
 788  05FC  20 02                     JR   NZ,DRTCHDEC1
 789  05FE  3E 08                     LD   A,08H
 790  0600  32 17 20        DRTCHDEC1:LD   (TIMEBUF+4),A
 791  0603  CD D2 06                  CALL CHANCONV
 792  0606  18 C4                     JR   DRTCH1
 793  0608  CD 9D 06        DRTOUT1:  CALL SCDISP        ;SCAN DISPLAY
 794  060B  CD 5A 06                  CALL KEYREAD
 795  060E  FE 01                     CP   01H           ;TIME KEY?
 796  0610  28 23                     JR   Z,DRTEND
 797  0612  FE 05                     CP   05H           ;DIRT KEY?
 798  0614  28 B6                     JR   Z,DRTCH1
 799  0616  FE 07                     CP   07H           ;+ KEY?
 800  0618  28 06                     JR   Z,DRTON
```

```
Line  Addr  Obj
 801  061A  FE 08                    CP   08H          ;- KEY?
 802  061C  28 0F                    JR   Z,DRTOFF
 803  061E  18 E8                    JR   DRTOUT1
 804  0620  3A 17 20          DRTON: LD   A,(TIMEBUF+4)
 805  0623  CD DB 06                 CALL CHANCONV1
 806  0626  32 20 20                 LD   (DISPBUF+7),A
 807  0629  D3 40                    OUT  (RELAY),A
 808  062B  18 DB                    JR   DRTOUT1
 809  062D  AF                DRTOFF: XOR  A
 810  062E  32 20 20                 LD   (DISPBUF+7),A
 811  0631  D3 40                    OUT  (RELAY),A
 812  0633  18 D3                    JR   DRTOUT1
 813  0635  3A 05 20          DRTEND: LD   A,(OUTBUF)
 814  0638  32 20 20                 LD   (DISPBUF+7),A
 815  063B  FB                       EI                ;RTC ENABLE
 816  063C  C3 5D 01                 JP   MAIN1
 817                          ;*****************************************
 818                          ;*      TIME BUFFER TO RTC               *
 819                          ;*****************************************
 820  063F  AF                TIMERTC: XOR  A           ;SEC=00
 821  0640  D3 22                    OUT  (RTC+2),A
 822  0642  3A 14 20                 LD   A,(TIMEBUF+1)
 823  0645  D3 23                    OUT  (RTC+3),A
 824  0647  3A 15 20                 LD   A,(TIMEBUF+2)
 825  064A  D3 24                    OUT  (RTC+4),A
 826  064C  3A 16 20                 LD   A,(TIMEBUF+3)
 827  064F  D3 25                    OUT  (RTC+5),A
 828  0651  3E 55                    LD   A,55H        ;TIME SET FLAG
 829  0653  32 01 20                 LD   (TIMEFLAG),A
 830  0656  FB                       EI                ;RTC ENABLE
 831  0657  C3 5D 01                 JP   MAIN1
 832                          ;*****************************************
 833                          ;*      KEY BOARD READ SUBMAIN           *
 834                          ;*****************************************
 835  065A  E5                KEYREAD: PUSH HL          ;SAVE HL
 836  065B  CD 7E 06                 CALL KEYTEST
 837  065E  FE 01                    CP   01H          ;ONE KEY?
 838  0660  20 19                    JR   NZ,KEYINVD
 839  0662  CD 9D 06          DEPCHK: CALL SCDISP       ;SCAN DISPLAY
 840  0665  CD 9D 06                 CALL SCDISP
 841  0668  2A 09 20                 LD   HL,(KEYIND);GET KEY INDEX
 842  066B  7E                       LD   A,(HL)       ;GET KEY CRRT
 843  066C  FE FF                    CP   0FFH
 844  066E  20 F2                    JR   NZ,DEPCHK    ;NOT DEPRESS
 845  0670  CD 9D 06                 CALL SCDISP       ;DEBOUNCING
 846  0673  CD 9D 06                 CALL SCDISP
 847  0676  3A 08 20                 LD   A,(KEYCRRT)
 848  0679  18 01                    JR   KEYEND
 849  067B  AF                KEYINVD: XOR  A
 850  067C  E1                KEYEND: POP  HL           ;RESTORE HL
```

```
Line  Addr  Obj

 851  067D  C9                        RET
 852                          ;****************************************
 853                          ;*        No.OF KEY CHECK               *
 854                          ;****************************************
 855  067E  C5                  KEYTEST:  PUSH BC          ;SAVE BC
 856  067F  E5                            PUSH HL          ;SAVE HL
 857  0680  01 00 08                      LD   BC,0800H    ;SET COUNTER
 858  0683  21 0B 20                      LD   HL,KEYBUF   ;SET INDEX
 859  0686  7E                  KTESTLP1: LD   A,(HL)      ;GET KEY IN
 860  0687  FE FF                         CP   0FFH
 861  0689  20 05                         JR   NZ,KEYCHK2
 862  068B  23                  KEYCHK1:  INC  HL          ;NEXT DATA
 863  068C  10 F8                         DJNZ KTESTLP1
 864  068E  18 09                         JR   KEYCHK3
 865  0690  0C                  KEYCHK2:  INC  C           ;ADD COUNT
 866  0691  32 08 20                      LD   (KEYCRRT),A;SAVE KEY CRRT
 867  0694  22 09 20                      LD   (KEYIND),HL;SAVE KEY INDEX
 868  0697  18 F2                         JR   KEYCHK1
 869  0699  79                  KEYCHK3:  LD   A,C
 870  069A  E1                            POP  HL          ;RESTORE HL
 871  069B  C1                            POP  BC          ;RESTORE BC
 872  069C  C9                            RET
 873                          ;****************************************
 874                          ;*        DISPLAY SCAN SUBMAIN          *
 875                          ;****************************************
 876  069D  F5                  SCDISP:   PUSH AF          ;SAVE AF
 877  069E  C5                            PUSH BC          ;SAVE BC
 878  069F  E5                            PUSH HL          ;SAVE HL
 879  06A0  AF                            XOR  A
 880  06A1  D3 00                         OUT  (SEG),A     ;SEGMENT OFF
 881  06A3  3A 18 20                      LD   A,(DIGCRRT);GET DIGIT
 882  06A6  D3 02                         OUT  (DIGIT),A   ;DIGIT ACTIVE
 883  06A8  4F                            LD   C,A         ;TO BASE INDEX
 884  06A9  06 00                         LD   B,00H
 885  06AB  C5                            PUSH BC          ;SAVE CURRENT
 886  06AC  21 0B 20                      LD   HL,KEYBUF   ;KEYBUF INDEX
 887  06AF  09                            ADD  HL,BC       ;ADJ.INDEX
 888  06B0  3E FF                         LD   A,0FFH      ;BLANK KEY
 889  06B2  77                            LD   (HL),A      ;KEY CLEAR
 890  06B3  DB 02                         IN   A,(KEYIN)   ;READ KEY
 891  06B5  CB 67                         BIT  4,A         ;KEY IN?
 892  06B7  20 03                         JR   NZ,NOKEYIN
 893  06B9  79                            LD   A,C         ;GET DIGIT
 894  06BA  3C                            INC  A
 895  06BB  77                            LD   (HL),A      ;SAVE KEY IN
 896  06BC  C1                  NOKEYIN:  POP  BC          ;RESTORE CRRT
 897  06BD  21 19 20                      LD   HL,DISPBUF
 898  06C0  09                            ADD  HL,BC       ;ADJ.INDEX
 899  06C1  7E                            LD   A,(HL)      ;GET SEGMENT
 900  06C2  D3 00                         OUT  (SEG),A
```

```
Line  Addr  Obj
 901  06C4  79                        LD   A,C          ;GET DIGIT
 902  06C5  3C                        INC  A
 903  06C6  FE 08                     CP   08H          ;LAST DIGIT?
 904  06C8  20 01                     JR   NZ,SETDIGIT
 905  06CA  AF                        XOR  A            ;FIRST DIGIT
 906  06CB  32 18 20          SETDIGIT: LD  (DIGCRRT),A
 907  06CE  E1                        POP  HL           ;RESTORE HL
 908  06CF  C1                        POP  BC           ;RESTORE BC
 909  06D0  F1                        POP  AF           ;RESTORE AF
 910  06D1  C9                        RET
 911                          ;****************************************
 912                          ;*        TIME TO DISPLAY               *
 913                          ;****************************************
 914  06D2  3A 17 20          CHANCONV: LD  A,(TIMEBUF+4)
 915  06D5  CD 35 07                  CALL LCONV        ;TO SEGMENT
 916  06D8  32 1D 20                  LD   (DISPBUF+4),A
 917  06DB  3A 17 20          CHANCONV1:LD  A,(TIMEBUF+4)
 918  06DE  3D                        DEC  A
 919  06DF  47                        LD   B,A
 920  06E0  3E 01                     LD   A,01H        ;"CH-1" FIRST
 921  06E2  07                CHCNV1:  RLCA             ;SHIFT CHAN.
 922  06E3  10 FD                     DJNZ CHCNV1
 923  06E5  32 20 20                  LD   (DISPBUF+7),A
 924  06E8  C9                        RET
 925  06E9  3A 16 20          DAYCONV:  LD  A,(TIMEBUF+3)
 926  06EC  3D                DAYCONV1: DEC A
 927  06ED  47                        LD   B,A
 928  06EE  3E 01                     LD   A,01H        ;"SUN" FIRST
 929  06F0  07                DYCNV1:  RLCA             ;SHIFT DAY
 930  06F1  10 FD                     DJNZ DYCNV1
 931  06F3  32 1F 20                  LD   (DISPBUF+6),A
 932  06F6  C9                        RET
 933  06F7  3A 15 20          HRCONV:   LD  A,(TIMEBUF+2)
 934  06FA  F5                HRCONV1:  PUSH AF         ;SAVE HOUR
 935  06FB  CD 31 07                  CALL HCONV
 936  06FE  32 19 20                  LD   (DISPBUF),A
 937  0701  F1                        POP  AF           ;RESTORE HR
 938  0702  CD 35 07                  CALL LCONV
 939  0705  F6 80                     OR   80H          ;WITH DOT
 940  0707  32 1A 20                  LD   (DISPBUF+1),A
 941  070A  C9                        RET
 942  070B  3A 14 20          MINCONV:  LD  A,(TIMEBUF+1)
 943  070E  F5                MINCONV1: PUSH AF         ;SAVE MIN.
 944  070F  CD 31 07                  CALL HCONV
 945  0712  32 1B 20                  LD   (DISPBUF+2),A
 946  0715  F1                        POP  AF           ;RESTORE MIN.
 947  0716  CD 35 07                  CALL LCONV
 948  0719  F6 80                     OR   80H          ;WITH DOT
 949  071B  32 1C 20                  LD   (DISPBUF+3),A
 950  071E  C9                        RET
```

```
Line  Addr  Obj

 951  071F  3A 13 20        SECCONV:  LD   A,(TIMEBUF)
 952  0722  F5              SECCONV1: PUSH AF          ;SAVE SEC.
 953  0723  CD 31 07                  CALL HCONV
 954  0726  32 1D 20                  LD   (DISPBUF+4),A
 955  0729  F1                        POP  AF          ;RESTORE SEC.
 956  072A  CD 35 07                  CALL LCONV
 957  072D  32 1E 20                  LD   (DISPBUF+5),A
 958  0730  C9                        RET
 959                        ;****************************************
 960                        ;*        HI NIBBLE TO SEGMENT          *
 961                        ;****************************************
 962  0731  0F              HCONV:    RRCA             ;SHIFT TO LOW
 963  0732  0F                        RRCA
 964  0733  0F                        RRCA
 965  0734  0F                        RRCA
 966                        ;****************************************
 967                        ;*        LO NIBBLE TO SEGMENT          *
 968                        ;****************************************
 969  0735  C5              LCONV:    PUSH BC          ;SAVE BC
 970  0736  E5                        PUSH HL          ;SAVE HL
 971  0737  E6 0F                     AND  0FH         ;LOW NIBBLE
 972  0739  4F                        LD   C,A
 973  073A  06 00                     LD   B,00H
 974  073C  21 51 07                  LD   HL,SEGTAB   ;SEG.INDEX
 975  073F  09                        ADD  HL,BC       ;ADJUST INDEX
 976  0740  7E                        LD   A,(HL)      ;GET SEGMENT
 977  0741  E1                        POP  HL          ;RESTORE HL
 978  0742  C1                        POP  BC          ;RESTORE BC
 979  0743  C9                        RET
 980                        ;****************************************
 981                        ;*        SET MESSAGE TO DISPLAY        *
 982                        ;****************************************
 983  0744  C5              SETMSSG:  PUSH BC          ;SAVE BC
 984  0745  D5                        PUSH DE          ;SAVE DE
 985  0746  01 06 00                  LD   BC,06H      ;DIGIT COUNT
 986  0749  11 19 20                  LD   DE,DISPBUF  ;SET DEST INDEX
 987  074C  ED B0                     LDIR             ;REPEAT MOVE
 988  074E  D1                        POP  DE          ;RESTORE DE
 989  074F  C1                        POP  BC          ;RESTORE BC
 990  0750  C9                        RET
 991                        ;****************************************
 992                        ;*        SEGMENT CODE TABLE            *
 993                        ;****************************************
 994  0751  3F 06 5B 4F     SEGTAB:   DB   3FH,06H,5BH,4FH
 995  0755  66 6D 7D 07               DB   66H,6DH,7DH,07H
 996  0759  7F 6F 77 7C               DB   7FH,6FH,77H,7CH
 997  075D  39 5E 79 71               DB   39H,5EH,79H,71H
 998                        ;        "SET-UP"
 999  0761  6D 79 78 40     MSSG1:    DB   6DH,79H,78H,40H,3EH,73H
1000  0765  3E 73
```

```
Line  Addr  Obj

1001                              ;             "DAY-"
1002  0767  00 5E 77 6E      MSSG2:    DB    00H,5EH,77H,6EH,40H,00H
1003  076B  40 00
1004                              ;             "CH-1"
1005  076D  00 39 74 40      MSSG3:    DB    00H,39H,74H,40H,00H,00H
1006  0771  00 00
1007                              ;             "ON
1008  0773  00 3F 54 00      MSSG4:    DB    00H,3FH,54H,00H,00H,00H
1009  0777  00 00
1010                              ;             "OFF"
1011  0779  00 3F 71 71      MSSG5:    DB    00H,3FH,71H,71H,00H,00H
1012  077D  00 00
1013                              ;             "ConF"
1014  077F  00 39 5C 54      MSSG6:    DB    00H,39H,5CH,54H,71H,00H
1015  0783  71 00
1016                              ;             "-------"
1017  0785  40 40 40 40      MSSG7:    DB    40H,40H,40H,40H,40H,40H
1018  0789  40 40
1019                              END
```

Sequence	Symbol name	Value
27	ALARM1	= 2100
28	ALARM2	= 21C8
29	ALARM3	= 2290
30	ALARM4	= 2358
31	ALARM5	= 2420
32	ALARM6	= 24E8
33	ALARM7	= 25B0
34	ALARM8	= 2678
47	ALARMEQ	00D2
42	CHANBIT	0087
92	CHANCONV	06D2
145	CHANCONV1	06DB
97	CHANDEC	02E6
99	CHANDEC1	02F6
96	CHANINC	02CD
98	CHANINC1	02DB
177	CHCNV1	06E2
18	CHCOUNT	2007
133	CHLIST1	04BA
143	CHSETON	0533
153	CLRCONF	0576
154	CLRPROG	058A
50	CMPSKIP1	00EC
36	COLDRES	012C
14	COUNT	2003
9	DAY	= 0025
41	DAYADJ	0081
65	DAYCONV	06E9
146	DAYCONV1	06EC
70	DAYDEC	01D2
72	DAYDEC1	01DC
69	DAYINC	01C0
71	DAYINC1	01CA
66	DAYSET1	01A7
63	DAYSKIP1	019F
64	DAYSKIP2	01A1
169	DEPCHK	0662
23	DIGCRRT	2018
3	DIGIT	= 0002
61	DIRECT	05B9
24	DISPBUF	2019
158	DRTCH1	05CC
162	DRTCHDEC	05F6
164	DRTCHDEC1	0600
161	DRTCHINC	05E4
163	DRTCHINC1	05EE
159	DRTEND	0635
166	DRTOFF	062D
165	DRTON	0620
160	DRTOUT1	0608

Sequence	Symbol name	Value
178	DYCNV1	06F0
51	FILLNXT	0137
39	HCONV	0731
8	HOUR	= 0024
75	HRCONV	06F7
147	HRCONV1	06FA
79	HRDEC	0230
81	HRDEC1	023B
78	HRINC	021D
80	HRINC1	0228
76	HRSET1	0204
73	HRSKIP1	01FC
74	HRSKIP2	01FE
15	IDBYTE	2004
35	INIT	0005
38	INT	0038
21	KEYBUF	200B
173	KEYCHK1	068B
172	KEYCHK2	0690
174	KEYCHK3	0699
19	KEYCRRT	2008
170	KEYEND	067C
4	KEYIN	= 0002
20	KEYIND	2009
168	KEYINVD	067B
56	KEYREAD	065A
167	KEYTEST	067E
171	KTESTLP1	0686
40	LCONV	0735
137	LISTDEC	04EC
139	LISTDEC1	04FC
134	LISTEND	0592
156	LISTEND1	059F
157	LISTEND2	05AF
136	LISTINC	04D3
138	LISTINC1	04E1
144	LISTSKIP	0539
150	LISTWAIT	0555
54	MAIN	0158
55	MAIN1	015D
7	MIN	= 0023
84	MINCONV	070B
148	MINCONV1	070E
87	MINDEC	0282
89	MINDEC1	028D
86	MININC	026F
88	MININC1	027A
85	MINSET1	0256
82	MINSKIP1	024E
83	MINSKIP2	0250

Sequence	Symbol name		Value
52	MSSG1		0761
62	MSSG2		0767
91	MSSG3		076D
125	MSSG4		0773
130	MSSG5		0779
152	MSSG6		077F
155	MSSG7		0785
175	NOKEYIN		06BC
46	NOTEQ1		00F1
120	OPSTAT		0431
126	OPSTAT1		0443
17	OUTBIT		2006
16	OUTBUF		2005
1	PPI	=	0000
103	PRDAYDEC		033A
105	PRDAYDEC1		0344
102	PRDAYINC		0328
104	PRDAYINC1		0332
95	PRDAYSET		0301
100	PRDAYSET1		030F
127	PRENTER		0470
132	PRENTER2		047E
131	PRENTNXT		0476
93	PRGCHSET1		02B4
109	PRHRDEC		0390
111	PRHRDEC1		039B
108	PRHRINC		037D
110	PRHRINC1		0388
101	PRHRSET		034C
106	PRHRSET1		0364
115	PRMINDEC		03D7
117	PRMINDEC1		03E2
114	PRMININC		03C4
116	PRMININC1		03CF
107	PRMINSET		03A3
112	PRMINSET1		03AB
151	PROGCLR		0569
94	PROGEND		0494
13	PROGFLAG		2002
122	PRSECDEC		041E
124	PRSECDEC1		0429
121	PRSECINC		040B
123	PRSECINC1		0416
113	PRSECSET		03EA
119	PRSECSET1		03F1
10	RELAY	=	0040
11	RESFLAG		2000
5	RTC	=	0020
44	SCDISP		069D
6	SEC	=	0022

Sequence	Symbol name		Value
118	SECCONV		071F
149	SECCONV1		0722
2	SEG	=	0000
179	SEGTAB		0751
176	SETDIGIT		06CB
67	SETEND		0295
53	SETMSSG		0744
48	SETON		00E4
129	SETOPOFF		0466
128	SETOPON		045C
49	SETZERO		00D9
22	TIMEBUF		2013
43	TIMECHK1		0094
45	TIMECHK2		00A2
12	TIMEFLAG		2001
60	TIMELIST		049E
135	TIMELIST1		0507
142	TIMELIST2		0526
59	TIMEPROG		0298
90	TIMERTC		063F
57	TIMERUN		017E
58	TIMESET		018D
140	TLISTNXT		050F
141	TLISTNXT1		0513
25	TMPIND1	=	2080
26	TMPIND2	=	2082
68	TOHRSET		01E4
77	TOMINSET		0243
37	WARMRES		0150

ภาคผนวก 2

DATA IC

รายการอุปกรณ์

- real time unit and output control

1	MM58167	1-
2	74LS138	1-
3	74LS374	1-
4	X-TAL 32.768 KHz	1-
5	TR BC547	2-
6	TR BC549	8-
7	TR BC557	1-
8	C 10 PF,10UF,	1-
9	R 10,100K,330,33K,10K,2.2K	10-
10	VC 5-30 PF	1-
11	DIODE 1N4001	12-
12	RELAY 6 V 2A	8-

รายการอุปกรณ์

-ส่วน display , keyboard ,power supply

1	74LS245	1-
2	74LS145	1-
3	LED SEVEN SEGMENT	6-
4	LED	15-
5	R 4.7 K	1-
6	CON1 40 PIN	1-
7	TRANSFORMER 0-9 V 2 A	1-
8	DIODE 1N4001	4-
9	C 1000 UF 25 V ,100 UF	1-
10	KEYBOARD	8-
11	IC LM 7805 3A	1-
12	AC POWER SW 250V 2A	1-
13	FUSE 1A	1-

รายการอุปกรณ์

-ส่วน cpu and memory

1	CPU Z80A	1-
2	EPROM 2764	1-
3	STATIC RAM 6116	1-
4	8255	1-
5	X-TAL 3.579 MHz	1-
6	74LS138	2-
7	74LS04	1-
8	TRANSISTOR BC547	1-
9	DIODE 1N4001	3-
10	LED	1-
11	CAPACITOR 4.7UF	1-
12	CAPACITOR 22 UF ,1 UF	1-
13	CAPACITOR 0.01,0.1 UF	4-
14	RESISTOR 10 K ,1 K	6-
15	RESISTOR 4.7 K ,330 ,220	2-
16	SWITCH RESET	1-
17	BATARY AA 1.5 V	3
18	DIP SWITCH	1-
19	CON1 40 PIN	1-

National Semiconductor

MM58167A Microprocessor Real Time Clock

General Description

The MM58167A is a low threshold metal gate CMOS circuit that functions as a real time clock in bus oriented microprocessor systems. The device includes an addressable real time counter, 56 bits of RAM, and two interrupt outputs. A $\overline{\text{POWER DOWN}}$ input allows the chip to be disabled from the rest of the system for standby low power operation. The time base is a 32,768 Hz crystal oscillator.

Features

- Microprocessor compatible (8-bit data bus)
- Milliseconds through month counters
- 56 bits of RAM with comparator to compare the real time counter to the RAM data
- 2 INTERRUPT OUTPUTS with 8 possible interrupt signals
- $\overline{\text{POWER DOWN}}$ input that disables all inputs and outputs except for one of the interrupts
- Status bit to indicate rollover during a read
- 32,768 Hz crystal oscillator
- Four-year calendar (no leap year)
- 24-hour clock

Functional Description

Real Time Counter

The real time counter is divided into 4-bit digits with 2 digits being accessed during any read or write cycle. Each digit represents a BCD number and is defined in Table I. Any unused bits are held at a logical zero during a read and ignored during a write. An unused bit is any bit not necessary to provide a full BCD number. For example tens of hours cannot legally exceed the number 2, thus only 2 bits are necessary to define the tens of hours. The other 2 bits in the tens of hours digit are unused. The unused bits are designated in Table I as dashes.

The addressable portion of the counter is from milliseconds to months. The counter itself is a ripple counter. The ripple delay is less than 60 μs above 4.0V and 300 μs at 2.0V.

RAM

56 bits of RAM are contained on-chip. These can be used for any necessary power down storage or as an alarm latch for comparison to the real time counter. The data in the RAM can be compared to the real time counter on a digit basis. The only digits that are not compared are the unit ten thousandths of seconds and tens of days of the week (these are unused in the real time counter). If the two most significant bits of any RAM digit are ones, then this RAM location will always compare.

The RAM is formatted the same as the real time counter, 4 bits per digit, 14 digits, however there are no unused bits. The unused bits in the real time counter will compare only to zeros in the RAM.

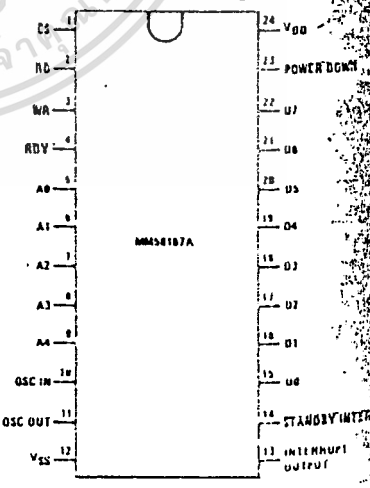
Interrupts and Comparator

There are two interrupt outputs. The first and most flexible is the INTERRUPT OUTPUT (a true high signal). This output can be programmed to provide 8 different frequency signals. They are: 10 Hz, 1 Hz, once per minute, once per hour, once a day, once a week, once a month, and when a RAM/real time counter comparison occurs. To enable the output a one is written into the interrupt control register in the bit location corresponding to the desired interrupt frequency (*Figure 1*). Once one or more bits have been enabled in the interrupt control register, the corresponding counter's rollover to its reset state will clock the interrupt status register and cause the interrupt output to go high. To reset the interrupt and to identify which frequency caused the interrupt, the interrupt status register is read. Reading this register places the contents of the status register on the data bus. The interrupting frequency will be identified by a one in the respective bit position. Remember the read will reset the interrupt.

The second interrupt is the $\overline{\text{STANDBY INTERRUPT}}$ (open drain output, active low). This interrupt occurs when enabled and when a RAM/real time counter comparison occurs. The $\overline{\text{STANDBY INTERRUPT}}$ is enabled by writing a one on the D0 line at address 1Bh, or disabled by writing a zero on the D0 line. This interrupt is not triggered by the edge of the compare signal, but rather by the level. Thus if the compare is enabled when the $\overline{\text{STANDBY INTERRUPT}}$ is enabled, the interrupt will turn on immediately.

Connection Diagram

Dual-In-Line Package



TOP VIEW

Order Number MM58167AN
See NS Package N24A

...ute Maximum Ratings

...d All Pins	$V_{SS}-0.3V$ to $V_{DD}+0.3V$	$V_{DD}-V_{SS}$	6.0V
...g Temperature	0°C to 70°C	Lead Temperature (Soldering, 10 seconds)	300°C
... Temperature	−65°C to 150°C		

...rical Characteristics $V_{SS}=0V$, $0°C \le T_A \le 70°C$

Parameter	Conditions	Min	Max	Units
... Voltage				
...	Outputs Enabled	4.0	5.5	V
...	POWER DOWN Mode	2.0	5.5	V
... Current				
...Static	Outputs TRI-STATE®			
...	f_{IN} = DC, V_{DD} = 5.5V		10	μA
...namic	Outputs TRI-STATE			
	f_{IN} = 32 kHz, V_{DD} = 5.5V			
	$V_{IH} \ge V_{DD}-0.3V$			
	$V_{IL} \le V_{SS}+0.3V$		20	μA
...namic	Outputs TRI-STATE			
	f_{IN} = 32 kHz, V_{DD} = 5.5V			
	V_{IH} = 2.0V, V_{IL} = 0.8V		5	mA
...oltage				
...cal Low		0.0	0.8	V
...cal High		2.0	V_{DD}	V
... Leakage Current	$V_{SS} \le V_{IN} \le V_{DD}$	−1	1	μA
... Impedance	I/O and INTERRUPT OUT			
...cal Low	V_{DD} = 4.5V, I_{OL} = 1.6 mA		0.4	V
...cal High	V_{DD} = 4.5V, I_{OH} = −400 μA	2.4		V
...	I_{OH} = −10 μA	$0.8\ V_{DD}$		V
...STATE	$V_{SS} \le V_{OUT} \le V_{DD}$	−1	1	μA
... Impedance	RDY and STANDBY INTERRUPT			
	(Open Drain Devices)			
...cal Low, Sink	V_{DD} = 4.5V, I_{OL} = 1.6 mA		0.4	V
...cal High, Leakage	$V_{OUT} \le V_{DD}$		10	μA

...ctional Description (Continued)

TABLE I. Real Time Counter Format

...ter Addressed		Units D0	D1	D2	D3	Max BCD Code	Tens D4	D5	D6	D7	Max BCD Code
...00 of Seconds	(00_H)	-	-	-	-	0	D4	D5	D6	D7	9
...dredths and Tenths Sec	(01_H)	D0	D1	D2	D3	9	D4	D5	D6	D7	9
...onds	(02_H)	D0	D1	D2	D3	9	D4	D5	D6	-	5
...utes	(03_H)	D0	D1	D2	D3	9	D4	D5	D6	-	5
...urs	(04_H)	D0	D1	D2	D3	9	D4	D5	-	-	2
...of the Week	(05_H)	D0	D1	D2	-	7	-	-	-	-	0
...of the Month	(06_H)	D0	D1	D2	D3	9	D4	D5	-	-	3
...nth	(07_H)	D0	D1	D2	D3	9	D4	-	-	-	1

...dicates unused bits

Functional Description (Continued)

TABLE II. Address Codes and Functions

A4	A3	A2	A1	A0	Function
0	0	0	0	0	Counter—Ten Thousandths of Seconds
0	0	0	0	1	Counter—Hundredths and Tenths of Seconds
0	0	0	1	0	Counter—Seconds
0	0	0	1	1	Counter—Minutes
0	0	1	0	0	Counter—Hours
0	0	1	0	1	Counter—Day of Week
0	0	1	1	0	Counter—Day of Month
0	0	1	1	1	Counter—Month
0	1	0	0	0	RAM—Ten Thousandths of Seconds
0	1	0	0	1	RAM—Hundredths and Tenths of Seconds
0	1	0	1	0	RAM—Seconds
0	1	0	1	1	RAM—Minutes
0	1	1	0	0	RAM—Hours
0	1	1	0	1	RAM—Day of Week
0	1	1	1	0	RAM—Day of Month
0	1	1	1	1	RAM—Months
1	0	0	0	0	Interrupt Status Register
1	0	0	0	1	Interrupt Control Register
1	0	0	1	0	Counters Reset
1	0	0	1	1	RAM Reset
1	0	1	0	0	Status Bit
1	0	1	0	1	GO Command
1	0	1	1	0	STANDBY INTERRUPT
1	1	1	1	1	Test Mode

All others unused

ภาคผนวก

สรุปลักษณะคำสั่งของ Z-80

บัสแอดเดรส : A_0-A_7 : [C]

A_8-A_{15} : [B]

ตารางที่ 1 คำสั่ง I/O

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
คำสั่ง I/O	IN	A, PORT	2		[A] ← [PORT] อินพุตข้อมูลเข้าสู่เอกคูมิวเลเตอร์จากพอร์ตที่กำหนดแอดเดรสโดยตรง บัสแอดเดรส A_0-A_7 : PORT A_8-A_{15} : [A]
	IN	REG. (C)	2	X X P X 0	[REG] ← [C] อินพุตข้อมูลเข้าสู่รีจิสเตอร์จากพอร์ต I/O โดยกำหนดหมายเลขพอร์ตไว้ในรีจิสเตอร์ C ซึ่งถ้าไบต์ที่ 2 เป็น 70_{16} ก็จะมีผลต่อแฟล็กอย่างเดียวเท่านั้น
	INIR			1 ? ? ? 1	ทำซ้ำจนกระทั่ง [B] = 0 [HL] ← [C] B ← B − 1 HL ← HL + 1 เคลื่อนย้ายข้อมูลเป็นกลุ่มจากพอร์ต I/O โดยกำหนดหมายเลขพอร์ต

ตารางที่ 1 (ต่อ) คำสั่ง I/O

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C	Z	S	P/O	A_c	N	การทำงาน
คำสั่ง I/O	INDR		2	?	?	?	?	?	1	ไว้ในรีจิสเตอร์ C และใช้ HL ชี้ที่ตำแหน่งแอดเดรสของหน่วยความจำ โดยเลื่อนจากตำแหน่งแอดเดรสต่ำไปที่แอดเดรสสูง สำหรับจำนวนไบต์ของข้อมูลที่มีการเคลื่อนย้ายให้เก็บไว้ในรีจิสเตอร์ B ทำซ้ำจนกระทั่ง (B) = 0 (HL) ← (C) B ← B − 1 HL ← HL − 1 เคลื่อนย้ายข้อมูลเป็นกลุ่มจากพอร์ต I/O โดยกำหนดหมายเลขพอร์ตไว้ในรีจิสเตอร์ C และใช้ HL ชี้ที่ตำแหน่งแอดเดรสของหน่วยความจำ โดยเลื่อนจากตำแหน่งแอดเดรสสูงไปที่แอดเดรสต่ำ สำหรับจำนวนไบต์ของข้อมูลที่มีการเคลื่อนย้ายให้เก็บไว้ในรีจิสเตอร์ B
	INI		2	X	?	?	?	?	1	(HL) ← (C) B ← B − 1 HL ← HL + 1 เคลื่อนย้ายข้อมูลที่มีขนาด 1 ไบต์จากพอร์ต I/O ที่ถูกกำหนดแอดเดรสโดยข้อมูลในรีจิสเตอร์ C ไปไว้ยังหน่วยความจำที่ชี้โดยข้อมูลที่อยู่ใน HL ลดค่าไบต์จำนวนนับในรีจิสเตอร์ B ลง 1 และเพิ่มค่า HL อีก 1
	IND		2		?	?	?	?	1	(HL) ← (C) B ← B − 1 HL ← HL − 1 เคลื่อนย้ายข้อมูลที่มีขนาด 1 ไบต์จากพอร์ต I/O ที่ถูกกำหนดแอดเดรส

ตารางที่ 1 (ต่อ) คำสั่ง I/O

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
คำสั่ง I/O	OUT	PORT,A	2		โดยข้อมูลในรีจิสเตอร์ C ไปไว้ยังหน่วยความจำที่ชี้โดยข้อมูลที่อยู่ใน HL ซึ่งจะลดค่าทั้งตัวนับไบต์ในรีจิสเตอร์ B และค่าใน HL ลง 1 [PORT] ← [A] เอาต์พุตจากแอกคูมิวเลเตอร์ไปยังพอร์ต I/O โดยตรง บัสแอดเดรส : A_0–A_7 : PORT A_8–A_{15} : A
	OUT	(C),REG	2		[[C]] ← [REG] เอาต์พุตจากรีจิสเตอร์ไปยังพอร์ต I/O ที่ถูกกำหนดแอดเดรสโดยรีจิสเตอร์ C
	OUTIR		2	1 ? ? ? 1	ทำซ้ำจนกระทั่ง [B] = 0 [[C]] ← [HL] [B] ← [B] – 1 [HL] ← [HL] + 1 เคลื่อนย้ายข้อมูลเป็นกลุ่มจากหน่วยความจำที่ชี้แอดเดรสโดยข้อมูลที่อยู่ในรีจิสเตอร์ HL ไปยังพอร์ต I/O ที่ถูกกำหนดแอดเดรสโดยข้อมูลในรีจิสเตอร์ C โดยเลื่อนแอดเดรสของหน่วยความจำจากแอดเดรสต่ำไปยังแอดเดรสสูง ข้อมูลในรีจิสเตอร์ B เป็นจำนวนไบต์ที่ใช้ในการเคลื่อนย้าย
	OUTDR		2	1 ? ? ? 1	ทำซ้ำจนกระทั่ง [B] = 0 [[C]] ← [HL] [B] ← [B] – 1 [HL] ← [HL] – 1 เคลื่อนย้ายข้อมูลเป็นกลุ่มจากหน่วยความจำที่ชี้แอดเดรสโดยข้อมูล

ตารางที่ 1 (ต่อ) คำสั่ง I/O

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
คำสั่ง I/O	OUTI		2	X ? ? ? ? 1	ที่อยู่ในรีจิสเตอร์ HL ไปยังพอร์ต I/O ที่ถูกกำหนดหมายเลขพอร์ตโดยข้อมูลที่อยู่ในรีจิสเตอร์ C โดยเลื่อนแอดเดรสของหน่วยความจำ จากแอดเดรสสูงไปยังแอดเดรสต่ำ ข้อมูลในรีจิสเตอร์ B เป็นจำนวนไบต์ที่ใช้ในการเคลื่อนย้าย [(C)] ← [HL] B ← B − 1 HL ← HL − 1 เคลื่อนย้ายข้อมูลที่มีขนาด 1 ไบต์ จากหน่วยความจำในตำแหน่งแอดเดรสที่ชี้โดยข้อมูลในรีจิสเตอร์ HL ไปยังพอร์ต I/O ที่กำหนดแอดเดรสโดยข้อมูลในรีจิสเตอร์ C ทำให้ลดค่าจำนวนไบต์ลง 1 และเพิ่มค่าแอดเดรสที่ต้นทางอีก 1
	OUTD		2	X ? ? ? ? 1	C ← HL B ← B − 1 HL ← HL − 1 เคลื่อนย้ายข้อมูลที่มีขนาด 1 ไบต์ จากหน่วยความจำในตำแหน่งแอดเดรสที่ชี้โดยข้อมูลในรีจิสเตอร์ HL ไปยังพอร์ต I/O ที่กำหนดแอดเดรสโดยข้อมูลในรีจิสเตอร์ C ทำให้ลดค่าจำนวนไบต์ลง 1 และลดค่าแอดเดรสที่ต้นทางอีก 1

ตารางที่ 2 คำสั่งอ้างอิงหน่วยความจำพื้นฐาน

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
คำสั่งอ้างอิงหน่วยความจำพื้นฐาน	LD	A,(ADDR)	3		[A]←[ADDR] โหลดแอกคูมิวเลเตอร์จากหน่วยความจำที่มีการกำหนดแอดเดรสโดยตรง
	LD	HL,(ADDR)	3		[H]←[ADDR+1], [L]←[ADDR] โหลด HL จากหน่วยความจำที่มีการกำหนดแอดเดรสโดยตรง
	LD	RP,(ADDR)	4		[RP(HI)]←[ADDR+1], [RP(LO)]←[ADDR] หรือ
		IX,(ADDR)			[IX(HI)]←[ADDR+1], [IX(LO)]←[ADDR] หรือ
		IY,(ADDR)			[IY(HI)]←[ADDR+1],[IY(LO)]←[ADDR] โหลดคู่รีจิสเตอร์หรืออินเด็กซ์รีจิสเตอร์จากหน่วยความจำที่มีการกำหนดแอดเดรสโดยตรง
	LD	(ADDR),A	3		[ADDR]←A นำค่าจากแอกคูมิวเลเตอร์ไปไว้ยังหน่วยความจำที่มีการกำหนดแอดเดรสโดยตรง
	LD	(ADDR),HL	3		[ADDR+1]←H, หรือ [ADDR]←L นำค่าจากคู่รีจิสเตอร์ HL ไปไว้ยังหน่วยความจำที่มีการกำหนดแอดเดรสโดยตรง
	LD	(ADDR),RP	4		[ADDR+1]←[RP(HI)], [ADDR]←RP(LO)] หรือ
		(ADDR),IX			[ADDR+1]←[IX(HI)], [ADDR]←[IX(LO)] หรือ
		(ADDR), IY			[ADDR+1]←[IY(HI)], [ADDR]←[IY(LO)] นำค่าจากคู่รีจิสเตอร์หรืออินเด็กซ์รีจิสเตอร์ไปไว้ยังหน่วยความจำที่มีการกำหนดแอดเดรสโดยตรง

ตารางที่ 2 (ต่อ) คำสั่งอ้างอิงหน่วยความจำพื้นฐาน

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
คำสั่งอ้างถึงหน่วยความจำพื้นฐาน	LD	A,(BC) A,(DE)	1		[A]←[[BC]] หรือ [A]←[[DE]] โหลดค่าลงในแอกคูมิวเลเตอร์จากหน่วยความจำที่มีการกำหนดแอดเดรสโดยข้อมูลที่อยู่ในคูรีจิสเตอร์ที่กำหนด
	LD	REG,(HL)	1		[REG]←[[HL]] โหลดรีจิสเตอร์จากหน่วยความจำที่มีการกำหนดแอดเดรสโดยรีจิสเตอร์ HL
	LD	(BC),A (DE),A	1 1		[[BC]]←[A] หรือ [[DE]]←[A] นำข้อมูลจากแอกคูมิวเลเตอร์ไปไว้ยังหน่วยความจำที่กำหนดแอดเดรสโดยคูรีจิสเตอร์
	LD	(HL),REG	1		[[HL]]←[REG] นำข้อมูลจากรีจิสเตอร์ไปไว้ยังหน่วยความจำที่กำหนดแอดเดรสโดยรีจิสเตอร์ HL
	LD	REG,(IX+DISP) REG,(IY+DISP)	3		[REG]←[[IX]+DISP] หรือ [REG]←[[IY]+DISP] ทำการโหลดข้อมูลจากหน่วยความจำไปไว้ยังรีจิสเตอร์ โดยกำหนดแอดเดรสแบบเบสสัมพัทธ์ (base relative)
	LD	(IX+DISP),REG (IY+DISP),REG	3		[IX+DISP]←[REG] หรือ [IY+DISP]←[REG] นำข้อมูลจากรีจิสเตอร์ไปไว้ยังหน่วยความจำที่มีการกำหนดแอดเดรส โดยอ้างอิงกับข้อมูลในอินเด็กซ์รีจิสเตอร์

ตารางที่ 3 คำสั่งค้นหาและเคลื่อนย้ายกลุ่มข้อมูล

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
คำสั่งค้นหาและเคลื่อนย้ายกลุ่มข้อมูล	LDIR		2	0 0 0	ทำซ้ำจนกระทั่ง [BC] = 0 [[DE]]←[[HL]] [DE]←[DE]+1 [HL]←[HL]+1 [BC]←[BC]−1 เคลื่อนย้ายข้อมูลเป็นบล็อกจากหน่วยความจำที่กำหนดแอดเดรสโดยข้อมูลในรีจิสเตอร์ HL ไปไว้ยังหน่วยความจำที่กำหนดแอดเดรสโดยข้อมูลในรีจิสเตอร์ DE โดยเริ่มจากแอดเดรสต่ำไปยังแอดเดรสสูง ข้อมูลในรีจิสเตอร์ BC เป็นตัวนับจำนวนไบต์ที่ใช้ในการเคลื่อนย้ายข้อมูล
	LDDR		2	0 0 0	ทำซ้ำจนกระทั่ง [BC] = 0 [[DE]]←[[HL]] [DE]←[DE]−1 [HL]←[HL]−1 [BC]←[BC]−1 เคลื่อนย้ายข้อมูลเป็นบล็อกจากหน่วยความจำที่กำหนดแอดเดรสโดยข้อมูลในรีจิสเตอร์ HL ไปไว้ยังหน่วยความจำที่กำหนดแอดเดรสโดยข้อมูลในรีจิสเตอร์ DE โดยเริ่มจากแอดเดรสสูงไปยังแอดเดรสต่ำ ข้อมูลในรีจิสเตอร์ BC เป็นตัวนับจำนวนไบต์ที่ใช้ในการเคลื่อนย้ายข้อมูล
	LDI		2	X 0 0	[[DE]]←[[HL]] [DE]←[DE]−1 [HL]←[HL]−1 [BC]←[BC]−1 เคลื่อนย้ายข้อมูลที่มีขนาด 1 ไบต์จากหน่วยความจำที่กำหนดแอดเดรส

ตารางที่ 3 (ต่อ) คำสั่งค้นหาและเคลื่อนย้ายกลุ่มข้อมูล

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
คำสั่งค้นหาและเคลื่อนย้ายกลุ่มข้อมูล	LDD		2	X 0 0	โดยข้อมูลในรีจิสเตอร์ HL ไปไว้ยังหน่วยความจำที่กำหนดแอดเดรสโดยข้อมูลในรีจิสเตอร์ DE แล้วเพิ่มค่าที่แอดเดรสต้นทางและปลายทางอย่างละ 1 และลดค่าที่ตัวนับจำนวนไบต์ลง 1 (DE) ← (HL) DE ← DE + 1 HL ← HL + 1 BC ← BC − 1
	LDDR		2	X 0 0	เคลื่อนย้ายข้อมูลที่มีขนาด 1 ไบต์จากหน่วยความจำที่กำหนดแอดเดรสโดยข้อมูลในรีจิสเตอร์ HL ไปไว้ยังหน่วยความจำที่กำหนดแอดเดรสโดยข้อมูลในรีจิสเตอร์ DE แล้วลดค่าที่แอดเดรสต้นทางและปลายทางอย่างละ 1 และลดค่าที่ตัวนับจำนวนไบต์ลง 1 ทำซ้ำจนกระทั่ง (A) = (HL) หรือ BC = 0 : (A) : (HL) HL ← HL + 1 BC ← BC − 1
	CPIR		2	X X X X 1	เปรียบเทียบกลุ่มข้อมูลจากแอกคูมิวเลเตอร์กับกลุ่มข้อมูลในหน่วยความจำที่กำหนดแอดเดรสโดยข้อมูลในรีจิสเตอร์ HL การทำงานจะเริ่มจากแอดเดรสต่ำไปยังแอดเดรสสูง และจะหยุดทำงานเมื่อการเปรียบเทียบตรงกัน หรือตัวนับจำนวนไบต์เป็น 0
	CPDR		2	X X X X 1	ทำซ้ำจนกระทั่ง (A) = (HL) หรือ BC = 0

ตารางที่ 3 (ต่อ) คำสั่งค้นหาและเคลื่อนย้ายกลุ่มข้อมูล

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
คำสั่งค้นหาและเคลื่อนย้ายกลุ่มข้อมูล					[A] : [[HL]] [HL]←[HL]−1 [BC]←[BC]−1 เปรียบเทียบกลุ่มข้อมูลจากแอกคูมิวเลเตอร์กับกลุ่มข้อมูลในหน่วยความจำที่กำหนดแอดเดรสโดยข้อมูลในรีจิสเตอร์ HL การทำงานของกลุ่มข้อมูลจะเริ่มจากแอดเดรสสูงไปยังแอดเดรสต่ำ และจะหยุดทำงานเมื่อการเปรียบเทียบตรงกัน หรือตัวนับจำนวนไบต์เป็น 0
	CPI		2	X X X X 1	[A] : [[HL]] [HL]←[HL]+1 [BC]←[BC]−1 เปรียบเทียบข้อมูลจากแอกคูมิวเลเตอร์กับข้อมูลในหน่วยความจำที่กำหนดแอดเดรสโดยข้อมูลในรีจิสเตอร์ HL ซึ่งจะทำให้เพิ่มค่าแอดเดรสขึ้น 1 และลดค่าตัวนับจำนวนไบต์ลง 1
	CPD		2	X X X X 1	[A] : [[HL]] [HL]←[HL]−1 [BC]←[BC]−1 เปรียบเทียบข้อมูลจากแอกคูมิวเลเตอร์กับข้อมูลในหน่วยความจำที่กำหนดแอดเดรสโดยข้อมูลในรีจิสเตอร์ HL ซึ่งจะทำให้ลดค่าแอดเดรสลง 1 และลดค่าตัวนับจำนวนไบต์ลง 1

ตารางที่ 4 คำสั่งอ้างอิงกับหน่วยความจำแบบทุติยภูมิ

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก						การทำงาน
				C	Z	S	P/O	A_c	N	
คำสั่งอ้างอิงกับหน่วยความจำแบบทุติยภูมิ	ADD	(HL)	1	X	X	X	0	X	0	[A]←[A]+[[HL]] บวกข้อมูลจากหน่วยความจำกับข้อมูลในแอกคูมิวเลเตอร์ ผลลัพธ์ที่ได้เก็บไว้ในแอกคูมิวเลเตอร์
	ADD	(IX+DISP) (IY+DISP)	3	X	X	X	0	X	0	[A]←[A]+[[IX]+DISP] หรือ [A]←[A]+[[IY]+DISP] บวกข้อมูลจากหน่วยความจำโดยใช้การกำหนดแอดเดรสแบบเบสอินเด็กซ์ ผลลัพธ์ที่ได้เก็บไว้ในแอกคูมิวเลเตอร์
	ADC	(HL)	1	X	X	X	0	X	0	[A]←[A]+[[HL]]+C บวกข้อมูลจากหน่วยความจำ โดยการรวมบิตตัวทดด้วย
	ADC	(IX+DISP) (IY+DISP)	3	X	X	X	0	X	0	[A]←[A]+[[IX]+DISP]+C หรือ [A]←[A]+[[IY]+DISP]+C บวกข้อมูลจากหน่วยความจำโดยการรวมบิตตัวทดด้วย และใช้วิธีกำหนดแอดเดรสแบบเบสอินเด็กซ์
	SUB	(HL)	1	X	X	X	0	X	1	[A]←[A]−[[HL]] ลบข้อมูลจากแอกคูมิวเลเตอร์ด้วยข้อมูลจากหน่วยความจำ
	SUB	(IX+DISP) (IY+DISP)	3	X	X	X	0	X	1	[A]←[A]−[[IX]+DISP] หรือ [A]←[A]−[IY+DISP] ลบข้อมูลจากแอกคูมิวเลเตอร์ด้วยข้อมูลจากหน่วยความจำที่กำหนดแอดเดรสแบบเบสอินเด็กซ์
	SBC	(HL)	1	X	X	X	0	X	1	[A]←[A]−[[HL]]−C ลบข้อมูลจากแอกคูมิวเลเตอร์ด้วยข้อมูลจากหน่วยความจำ โดยคิดการขอยืมด้วย
	SBC	(IX+DISP) (IY+DISP)	3	X	X	X	0	X	1	[A]←[A]−[IX+DISP]−C หรือ [A]←[A]−[IY+DISP]−C

ตารางที่ 4 (ต่อ) คำสั่งอ้างอิงกับหน่วยความจำแบบทุติยภูมิ

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
คำสั่งอ้างอิงกับหน่วยความจำแบบทุติยภูมิ	AND	(HL)	1	0 X X P X 1	ลบข้อมูลจากแอกคูมิวเลเตอร์ด้วยข้อมูลจากหน่วยความจำที่กำหนดแอดเดรสแบบเบสสัมพัทธ์ โดยคิดการขอยืมด้วย [A]←[A] ∧ [(HL)] AND ข้อมูลจากหน่วยความจำกับแอกคูมิวเลเตอร์
	AND	(IX+DISP) (IY+DISP)	3	0 X X P X 1	[A]←[A] ∧ [(IX)+DISP] หรือ [A]←[A] ∧ [(IY)+DISP] AND ข้อมูลจากหน่วยความจำที่มีการกำหนดแอดเดรสแบบเบสสัมพัทธ์กับแอกคูมิวเลเตอร์
	OR	(HL)	1	0 X X P X 0	[A]←[A] ∨ [(HL)] OR ข้อมูลจากหน่วยความจำกับแอกคูมิวเลเตอร์
	OR	(IX+DISP) (IY+DISP)	3	0 X X P X 0	[A]←[A] ∨ [(IX)+DISP] หรือ [A]←[A] ∨ [(IY)+DISP] OR ข้อมูลจากหน่วยความจำที่มีการกำหนดแอดเดรสแบบเบสสัมพัทธ์กับแอกคูมิวเลเตอร์
	XOR	(HL)	1	0 X X P X 0	[A]←[A]↔[(HL)] EX-OR ข้อมูลจากหน่วยความจำกับแอกคูมิวเลเตอร์
	XOR	(IX+DISP) (IY+DISP)	3	0 X X P X 0	[A]←[A]↔[(IX)+DISP] หรือ [A]←[A]↔[(IY)+DISP] EX-OR ข้อมูลจากหน่วยความจำที่มีการกำหนดแอดเดรสแบบเบสสัมพัทธ์กับแอกคูมิวเลเตอร์
	CP	(HL)	1	0 X X 0 X 1	[A]−[(HL)] เปรียบเทียบข้อมูลจากหน่วยความจำกับแอกคูมิวเลเตอร์

ตารางที่ 4 (ต่อ) คำสั่งอ้างอิงกับหน่วยความจำแบบทุติยภูมิ

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C	Z	S	P/O	A_c	N	การทำงาน
คำสั่งอ้างอิงกับหน่วยความจำแบบทุติยภูมิ	CP	(IX+DISP) (IY+DISP)	3	0	X	X	0	X	1	[A] : [[IX]+DISP] หรือ [A] : [[IY]+DISP] เปรียบเทียบข้อมูลจากหน่วยความจำที่มีการกำหนดแอดเดรสแบบเบสสัมพัทธ์กับแอกคูมิวเลเตอร์
	INC	(HL)	1		X	X	0	X	0	[[HL]] ← [[HL]] + 1 เพิ่มค่าข้อมูลในหน่วยความจำขึ้นอีก 1
	INC	(IX+DISP) (IY+DISP)	3		X	X	0	X	0	[[IX]+DISP] ← [[IX]+DISP] + 1 หรือ [[IY]+DISP] ← [[IY]+DISP] + 1 เพิ่มค่าข้อมูลในหน่วยความจำที่มีการกำหนดแอดเดรสแบบเบสสัมพัทธ์
	DEC	(HL)	1		X	X	0	X	1	[[HL]] ← [[HL]] − 1 ลดค่าข้อมูลในหน่วยความจำโดยใช้การอ้างอิงแอดเดรสแบบอิมไพล์
	DEC	(IX+DISP) (IY+DISP)	3		X	X	0	X	1	[[IX+DISP]] ← [[IX+DISP]] − 1 หรือ [[IY+DISP]] ← [[IY+DISP]] − 1 ลดค่าข้อมูลในหน่วยความจำที่มีการกำหนดแอดเดรสแบบเบสสัมพัทธ์

ตารางที่ 5 คำสั่งอิมมิเดียต

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C	Z	S	P/O	A_c	N	การทำงาน
คำสั่งอิมมิเดียต	LD	REG,DATA	2							REG ← DATA โหลดข้อมูลลงในรีจิสเตอร์โดยวิธีการอิมมิเดียต
	LD	RP, DATA16	3							RP ← DATA 16 โหลดข้อมูลที่มีขนาด 16 บิตลงในคู่รีจิสเตอร์ โดยวิธีการอิมมิเดียต

ตารางที่ 5 (ต่อ) คำสั่งอิมมีเดียต

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
คำสั่งอิมมีเดียต	LD	IX,DATA 16 IY,DATA 16	4		(IX)← DATA 16 หรือ (IY)← DATA 16 โหลดข้อมูลที่มีขนาด 16 บิตลงในอินเด็กซ์รีจิสเตอร์ โดยวิธีการอิมมีเดียต
	LD	(HL),DATA	2		((HL))← DATA โหลดข้อมูลลงในหน่วยความจำโดยใช้การอ้างอิงแอดเดรสแบบอิมไพล์
	LD	(IX+DISP),DATA (IY+DISP),DATA	4		((IX+DISP))← DATA หรือ ((IY+DISP))← DATA โหลดข้อมูลลงในหน่วยความจำโดยใช้การอ้างอิงแอดเดรสแบบเบสสัมพัทธ์

ตารางที่ 6 คำสั่งกระโดด

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
คำสั่งกระโดด	JP	LABEL	3		(PC)← LABEL คำสั่งให้กระโดดไปยังตำแหน่ง LABEL
	JR	DISP	2		PC ← PC + 2 + DISP คำสั่งกระโดดแบบอ้างอิงแอดเดรส
	JP	(HL)	1		PC ← HL กระโดดไปยังแอดเดรสที่กำหนดไว้ในรีจิสเตอร์ HL
	JP	(IX (IY	2		PC ← IX หรือ PC ← IY กระโดดไปยังแอดเดรสที่กำหนดไว้ในอินเด็กซ์รีจิสเตอร์

ตารางที่ 7 คำสั่งเรียกโปรแกรมย่อยและรีเทิร์น

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
คำสั่งเรียกโปรแกรมย่อยและรีเทิร์น	CALL	LABEL	3		[SP]−1]←[PC(HI)] [[SP]−2]←[PC(LO)] [SP]←[SP]−2 [PC]←LABEL กระโดดไปยังโปรแกรมย่อยที่ตำแหน่ง LABEL
	CALL	COND,LABEL	3		กระโดดไปยังโปรแกรมย่อยเมื่อเงื่อนไขเป็นจริง ถ้าเงื่อนไขไม่เป็นจริง จะกระทำคำสั่งถัดไป
	RET		1		[PC(LO)]←[[SP]] [PC(HI)]←[[SP + 1]] [SP]←[SP]+2 รีเทิร์นจากโปรแกรมย่อย
	RET	COND	1		รีเทิร์นจากโปรแกรมย่อยเมื่อเงื่อนไขเป็นจริง ถ้าไม่เป็นจริงจะกระทำคำสั่งถัดไป

ตารางที่ 8 คำสั่งทำงานแบบอิมมีเดียต

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
คำสั่งทำงานแบบอิมมีเดียต	ADD	DATA	2	X X X 0 X 0	A←A+DATA บวกข้อมูลแบบอิมมีเดียตกับแอกคูมิวเลเตอร์
	ADC	DATA	2	X X X 0 X 0	A←A+DATA+C บวกข้อมูลแบบอิมมีเดียตโดยคิดบิตทดด้วย
	SUB	DATA	2	X X X 0 X 1	A←A−DATA ลบข้อมูลจากแอกคูมิวเลเตอร์แบบอิมมีเดียต

ตารางที่ 8 (ต่อ) คำสั่งทำงานแบบอิมมีเดียต

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก						การทำงาน
				C	Z	S	P/O	A_c	N	
คำสั่งทำงานแบบอิมมีเดียต	SBC	DATA	2	X	X	X	0	X	1	[A]←[A]−DATA−C ลบข้อมูลแบบอิมมีเดียตโดยคิดบิตทดด้วย
	AND	DATA	2	0	X	X	P	X	1	[A]←[A] ∧ DATA AND ข้อมูลแบบอิมมีเดียตกับแอกคูมิวเลเตอร์
	OR	DATA	2	0	X	X	P	X	0	[A]←[A] ∨ DATA OR ข้อมูลแบบอิมมีเดียตกับแอกคูมิวเลเตอร์
	XOR	DATA	2	0	X	X	P	X	0	[A]←[A] ⊻ DATA EX-OR ข้อมูลแบบอิมมีเดียตกับแอกคูมิวเลเตอร์
	CP	DATA	2	X	X	X	0	X	1	[A] : DATA เปรียบเทียบข้อมูลแบบอิมมีเดียตกับแอกคูมิวเลเตอร์

ตารางที่ 9 คำสั่งกระโดดแบบมีเงื่อนไข

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก						การทำงาน
				C	Z	S	P/O	A_c	N	
คำสั่งกระโดดแบบมีเงื่อนไข	JP	COND,LABEL	3							ถ้า COND ดังนั้น [PC]←LABEL ถ้าเงื่อนไขเป็นจริงจะกระโดดไปยังแอดเดรสที่กำหนดด้วย LABEL
	JR	C,DISP	2							ถ้า C = 1 ดังนั้น [PC]←[PC]+2+DISP ถ้าเงื่อนไขในแฟล็กบิตทดมีค่าเป็น 1 จะกระโดดไปยังแอดเดรสที่อ้างอิงกับค่า PC เดิม
	JR	NC,DISP	2							ถ้า C = 0 ดังนั้น [PC]←[PC]+2+DISP ถ้าเงื่อนไขในแฟล็กบิตทดมีค่าเป็น 0 จะกระโดดไปยังแอดเดรสที่อ้างอิงกับค่า PC เดิม
	JR	Z,DISP	2							ถ้า Z = 1 ดังนั้น [PC]←[PC]+2+DISP

ตารางที่ 9 (ต่อ) คำสั่งกระโดดแบบมีเงื่อนไข

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
คำสั่งกระโดดแบบมีเงื่อนไข	JR	NZ,DISP	2		ถ้าเงื่อนไขในแฟล็กบิตศูนย์มีค่าเป็น 1 จะกระโดดไปยังแอดเดรสที่อ้างอิงกับค่า PC เดิม ถ้า Z = 0 ดังนั้น [PC]←[PC]+2+DISP
	DJNZ	DISP	2		ถ้าเงื่อนไขในแฟล็กบิตศูนย์มีค่าเป็น 0 จะกระโดดไปยังแอดเดรสที่อ้างอิงกับค่า PC เดิม [B]←[B]−1 ถ้า [B] ≠ 0 ดังนั้น [PC]←[PC]+2+DISP ลดค่าในรีจิสเตอร์ B ลง 1 และตรวจสอบดูว่าเป็นศูนย์หรือไม่ ถ้าเป็นศูนย์ให้กระโดดไปยังแอดเดรสที่อ้างอิงกับตัว PC

ตารางที่ 10 คำสั่งเคลื่อนย้ายระหว่างรีจิสเตอร์

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
คำสั่งเคลื่อนย้ายระหว่างรีจิสเตอร์	LD	DST,SRC			DST ← SRC เคลื่อนย้ายข้อมูลจากรีจิสเตอร์ SRC ไปยังรีจิสเตอร์ DST รีจิสเตอร์ที่ใช้อาจเป็น A, B, C, D, E, H หรือ L
	LD	A,IV	2	X X I 0 0	A ← IV เคลื่อนย้ายข้อมูลจากอินเตอร์รัพต์เวกเตอร์ไปไว้ที่แอกคูมิวเลเตอร์
	LD	A,R	2	X X I 0 0	A ← R เคลื่อนย้ายข้อมูลจากรีเฟรชรีจิสเตอร์ไปไว้ที่แอกคูมิวเลเตอร์
	LD	IV,A	2		IV ← A เคลื่อนย้ายข้อมูลจากแอกคูมิวเลเตอร์มาใส่ในอินเตอร์รัพต์เวกเตอร์

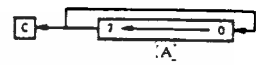
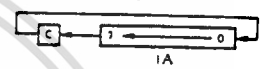
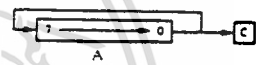
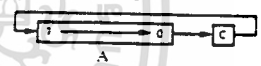
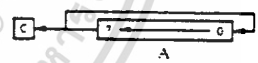
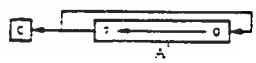
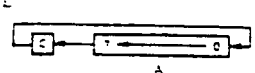
ตารางที่ 11 (ต่อ) คำสั่งการทำงานระหว่างรีจิสเตอร์

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก						การทำงาน
				C	Z	S	P/O	A_c	N	
	ADD	IY, RR	2	X				?	0	[IY]←[IY]+[RR] บวกคู่รีจิสเตอร์โดยกระทำระหว่างคู่รีจิสเตอร์กับอินเด็กซ์รีจิสเตอร์ IY

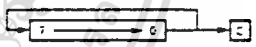
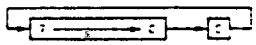
ตารางที่ 12 คำสั่งการทำงานกับรีจิสเตอร์

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก						การทำงาน
				C	Z	S	P/O	A_c	N	
การทำงานกับรีจิสเตอร์	DAA		1	X	X	X	P	X		ปรับค่าตัวเลขในแอกคูมิวเลเตอร์ให้เป็นตัวเลขฐานสิบ (ตัวเลข BCD) โดยสมมติให้ค่าในแอกคูมิวเลเตอร์ได้ผ่านการบวกหรือลบแบบตัวเลข BCD
	CPL		1					1	1	A←Ā คอมพลีเมนต์แอกคูมิวเลเตอร์
	NEG		2	X	X	X	0	X	1	A←A + 1 การทำให้ตัวเลขในแอกคูมิวเลเตอร์เป็นลบ
	INC	REG	1		X	X	0	X	0	REG←REG + 1 เพิ่มค่าในรีจิสเตอร์อีก 1
	INC	RP	1							RP←RP + 1 เพิ่มค่าในคู่รีจิสเตอร์อีก 1
	INC	IX	2							IX←IX + 1 หรือ
		IY								IY←IY + 1 เพิ่มค่าในอินเด็กซ์รีจิสเตอร์อีก 1
	DEC	REG	1		X	X	0	X	1	REG←REG − 1 ลดค่าในรีจิสเตอร์ลง 1
	DEC	RP	1							RP←RP − 1 ลดค่าในคู่รีจิสเตอร์ลง 1

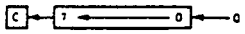
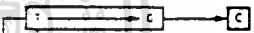
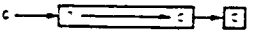
ตารางที่ 12 (ต่อ) คำสั่งการทำงานกับรีจิสเตอร์

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C	Z	S	P/O	A_c	N	การทำงาน
การทำงานกับรีจิสเตอร์	DEC	IX IY	2							(IX) ← (IX) − 1 หรือ (IY) ← (IY) − 1
	RICA		1	X				0	0	ลดค่าอินเด็กซ์รีจิสเตอร์ลง 1 
	RLA		1	X				0	0	หมุนค่าในแอกคูมิวเลเตอร์ไปทางซ้ายและเลื่อนเข้าสู่บิตตัวทดด้วย 
	RRCA		1	X				0	0	หมุนค่าในแอกคูมิวเลเตอร์ไปทางซ้าย โดยรวมบิตตัวทดด้วย 
	RRA		1	X				0	0	หมุนค่าในแอกคูมิวเลเตอร์ไปทางขวา และเลื่อนเข้าสู่บิตตัวทดด้วย 
	RLC	REG (HL)	2	X	X	X	P	0	0	หมุนค่าในแอกคูมิวเลเตอร์ไปทางขวา โดยรวมบิตตัวทดด้วย  (REG) หรือ (HL) หมุนค่าในรีจิสเตอร์หรือค่าในหน่วยความจำไปทางซ้ายและเลื่อนเข้าสู่บิตตัวทดด้วย โดยใช้การอ้างอิงแอดเดรสแบบอิมไพล์
	RLC	(IX−DISP) (IY−DISP)	4	X	X	X	P	0	0	 (IX − DISP) หรือ (IY − DISP) หมุนค่าในหน่วยความจำที่มีการกำหนดแอดเดรสแบบเบสสัมพัทธ์ไปทางซ้าย และเลื่อนเข้าสู่บิตตัวทดด้วย
	RL	REG (HL)	2	X	X	X	P	0	0	 (REG) หรือ (HL)

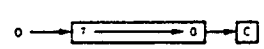
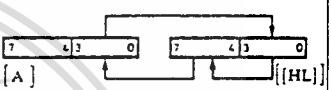
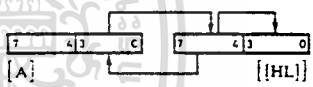
ตารางที่ 12 (ต่อ) คำสั่งการทำงานกับรีจิสเตอร์

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
คำสั่งการทำงานกับรีจิสเตอร์	RL	(IX + DISP) (IY + DISP)	4	X X X P 0 0	หมุนค่าในรีจิสเตอร์หรือค่าในหน่วยความจำที่มีการกำหนดแอดเดรสแบบเบสสัมพัทธ์ไปทางซ้าย และรวมบิตตัวทดด้วย  (IIX+DISP) หรือ (IIY+DISP)
	RRC	REG (HL)	2	X X X P 0 0	หมุนค่าในหน่วยความจำที่มีการกำหนดแอดเดรสแบบเบสสัมพัทธ์ไปทางซ้าย โดยรวมบิตตัวทดด้วย  [REG] หรือ [HL]
	RRC	(IX + DISP) (IY + DISP)	4	X X X P 0 0	หมุนค่าในรีจิสเตอร์หรือค่าในหน่วยความจำไปทางขวาโดยใช้การอ้างอิงแอดเดรสแบบอิมไพล์ และเลื่อนเข้าสู่บิตตัวทด  [IX + DISP] หรือ [IY + DISP]
	RR	REG (HL)	2	X X X P 0 0	หมุนค่าในหน่วยความจำที่มีการกำหนดแอดเดรสแบบเบสสัมพัทธ์ไปทางขวา โดยเลื่อนเข้าสู่บิตตัวทด  [REG] หรือ [HL]
	RR	(IX + DISP) (IY + DISP)	4	X X X P 0 0	หมุนค่าในรีจิสเตอร์หรือค่าในหน่วยความจำไปทางขวาโดยใช้การอ้างอิงแอดเดรสแบบอิมไพล์ และรวมบิตตัวทดด้วย  [IX + DISP] หรือ [IY + DISP]

ตารางที่ 12 (ต่อ) คำสั่งการทำงานกับรีจิสเตอร์

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
คำสั่งการทำงานกับรีจิสเตอร์	SLA	REG (HL)	2	X X X P 0 0	 [REG] หรือ [(HL)] เลื่อนบิตของรีจิสเตอร์หรือหน่วยความจำไปทางซ้ายโดยใช้การอ้างอิงแอดเดรสแบบอิมไพล์ และเลื่อนเข้าสู่บิตตัวทด
	SLA	(IX+DISP) (IY+DISP)	4	X X X P 0 0	 [(IX+DISP)] หรือ [(IY+DISP)] เลื่อนบิตของหน่วยความจำที่มีการกำหนดแอดเดรสแบบเบสสัมพัทธ์ไปทางซ้าย โดยเลื่อนเข้าสู่บิตตัวทดด้วย
	SRA	REG (HL)	2	X X X P 0 0	 [REG] หรือ [(HL)] เลื่อนบิตทางคณิตศาสตร์ไปทางขวาของรีจิสเตอร์หรือหน่วยความจำโดยใช้การอ้างอิงแอดเดรสแบบอิมไพล์
	SRA	(IX+DISP) (IY+DISP)	4	X X X P 0 0	 [(IX+DISP)] หรือ [(IY+DISP)] เลื่อนบิตทางคณิตศาสตร์ไปทางขวาของหน่วยความจำที่มีการกำหนดแอดเดรสแบบเบสสัมพัทธ์
	SRL	REG (HL)	2	X X X P 0 0	 [REG] หรือ [(HL)] เลื่อนบิตของรีจิสเตอร์หรือหน่วยความจำที่มีการอ้างอิงแอดเดรสแบบอิมไพล์ไปทางขวา โดยเลื่อนเข้าสู่บิตตัวทดด้วย

ตารางที่ 12 (ต่อ) คำสั่งการทำงานกับรีจิสเตอร์

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C	Z	S	P/O	A_c	N	การทำงาน
คำสั่งการทำงานกับรีจิสเตอร์	SRL	(IX+DISP) (IY+DISP)	4	X	X	X	P	0	0	 [(IX)+DISP] หรือ [(IY)+DISP] เลื่อนบิตของหน่วยความจำที่มีการกำหนดแอดเดรสแบบเบสสัมพัทธ์ไปทางขวา โดยเลื่อนเข้าสู่บิตตัวทด ด้วย
	RLD		2		X	X	P	0	0	 [A] [(HL)] หมุนตัวเลขแบบ BCD ไปทางซ้าย โดยอยู่ระหว่างแอกคูมิวเลเตอร์กับหน่วยความจำที่มีการอ้างอิงแอดเดรสแบบอิมไพล์
	RRD		2		X	X	P	0	0	 [A] [(HL)] หมุนตัวเลขแบบ BCD ไปทางขวา โดยอยู่ระหว่างแอกคูมิวเลเตอร์กับหน่วยความจำที่มีการอ้างอิงแอดเดรสแบบอิมไพล์

ตารางที่ 13 คำสั่งการทำงานเป็นบิต

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C	Z	S	P/O	A_c	N	การทำงาน
คำสั่งการทำงานเป็นบิต	BIT	B,REG	2		X	?	?	1	0	Z ← REG(B) นำข้อมูลบิตที่กำหนดในรีจิสเตอร์มาคอมพลีเมนต์ แล้วใส่ในแฟล็กศูนย์
	BIT	B,(HL)	2		X	?	?	1	0	Z ← [(HL)](B) นำข้อมูลบิตที่อยู่ในหน่วยความจำที่มีการอ้างอิงแอดเดรสแบบอิมไพล์มาคอมพลีเมนต์ แล้วใส่ในแฟล็กศูนย์

ตารางที่ 13 (ต่อ) คำสั่งการทำงานเป็นบิต

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C	Z	S	P/O	A_c	N	การทำงาน
คำสั่งการทำงานเป็นบิต	BIT	B,(IX + DISP) B,(IY + DISP)	4		X	?	?	1	0	Z←[(IX) + DISP] (B) หรือ Z←[(IY) + DISP] (B) นำข้อมูลบิตที่อยู่ในหน่วยความจำที่มีการกำหนดแอดเดรสแบบเบส-สัมพัทธ์มาคอมพลีเมนต์ แล้วใส่ในแฟล็กศูนย์
	SET	B,REG	2							REG (B)←1 เซตค่าบิตในรีจิสเตอร์
	SET	B,(HL)	2							[(HL)] (B)←1 เซตค่าบิตในหน่วยความจำที่มีการอ้างอิงแอดเดรสแบบอิมไพล์
	SET	B,(IX + DISP) B,(IY + DISP)	4							[(IX) + DISP] (B)←1 หรือ [(IY) + DISP] (B)←1 เซตค่าบิตในหน่วยความจำที่มีการกำหนดแอดเดรสแบบเบสสัมพัทธ์
	RES	B,REG	2							REG (B)←0 รีเซตค่าบิตในรีจิสเตอร์
	RES	B,(HL)	2							[(HL)] (B)←0 รีเซตค่าบิตในหน่วยความจำที่มีการอ้างอิงแอดเดรสแบบอิมไพล์
	RES	B,(IX + DISP) B,(IY + DISP)	4							[(IX) + DISP] (B)←0 หรือ [(IY) + DISP] (B)←0 รีเซตค่าบิตในหน่วยความจำที่มีการกำหนดแอดเดรสแบบเบสสัมพัทธ์

ตารางที่ 14 สแตก

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C	Z	S	P/O	A_c	N	การทำงาน
สแตก	PUSH	PR	1							[(SP − 1)] ←[PR(HI)] [(SP − 2)] ←[PR(LO)] (SP) ←(SP) +2

ตารางที่ 14 (ต่อ) สแตก

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
สแตก	PUSH	IX IY	2		นำค่าของคู่รีจิสเตอร์ใส่ลงในสแตกและลดค่าของสแตกพอยน์เตอร์ลง [[SP]−1]←[IX(HI)] หรือ [[SP]−1]←[IY(HI)] [[SP]−2]←[IX(LO)] หรือ [[SP]−2]←[IY(LO)] [SP]←[SP]−2
	POP	PR	1		นำค่าของอินเด็กซ์รีจิสเตอร์ใส่ลงในสแตกและลดค่าของสแตกพอยน์เตอร์ลง [PR(LO)]←[[SP]] [PR(HI)]←[[SP+1]] [SP]←[SP+2]
	POP	IX IY	2		นำค่าจากสแตกใส่ลงในคู่รีจิสเตอร์ และเพิ่มค่าของสแตกพอยน์เตอร์ [IX(LO)]←[[SP]] หรือ [IY(LO)]←[[SP]] [IX(HI)]←[[SP+1]] หรือ [IY(HI)]←[[SP+1]] [SP]←[SP+2]
	EX	(SP),HL	1		นำค่าจากสแตกใส่ลงในอินเด็กซ์รีจิสเตอร์และเพิ่มค่าของสแตกพอยน์เตอร์ แลกเปลี่ยนค่าระหว่างรีจิสเตอร์ HL กับค่าที่อยู่ตอนบนของสแตก [H]↔[[SP+1]] [L]↔[[SP]]
	EX	(SP),IX (SP),IY	2		[IX(HI)]↔[[SP+1]] หรือ [IY(HI)]↔[[SP+1]] [IX(LO)]↔[[SP]] หรือ [IY(LO)]↔[[SP]]

ตารางที่ 14 (ต่อ) สแต็ก

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
สแต็ก					แลกเปลี่ยนค่าระหว่างอินเด็กซ์-รีจิสเตอร์กับค่าที่อยู่ตอนบนของสแต็ก

ตารางที่ 15 อินเตอร์รัพต์

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
อินเตอร์รัพต์	DI		1		ดิสเอเบิลอินเตอร์รัพต์
	EI		1		อินาเบิลอินเตอร์รัพต์
	RST	N	1		PUSH PC, $(PC) \leftarrow (8 \cdot N)_{16}$ กระโดดไปเริ่มที่ค่ากำหนด
	RETI		2		รีเทิร์นจากอินเตอร์รัพต์
	RETN		2		รีเทิร์นจากนอนมาสเคเบิลอินเตอร์รัพต์
	IM	0	2		เซตอินเตอร์รัพต์โหมด 0
		1			เซตอินเตอร์รัพต์โหมด 1
		2			เซตอินเตอร์รัพต์โหมด 2

ตารางที่ 16 สถานะ

ชนิด	นีโมนิก	โอเปอแรนด์ (S)	จำนวนไบต์	สถานะแฟล็ก C Z S P/O A_c N	การทำงาน
สถานะ	SCF		1	1 0 0	$C \leftarrow 1$ เซตบิตตัวทด
	CCF		1	X ? 0	$C \leftarrow \bar{C}$ คอมพลีเมนต์บิตตัวทด
	NOP		1		ไม่มีการดีดต่อ แต่ยังคงรีเฟรชหน่วยความจำ
	HALT		1		ซีพียูเลิกทำงาน

สรุปคำสั่งที่อยู่ในรูปนีโมนิก รหัสภาษาเครื่องและช่วงเวลาในการประมวลผล (ต่อ)

คำสั่ง	ออบเจ็กต์โค้ด	จำนวนไบต์	สัญญาณนาฬิกา	รหัสนีโมนิก 8080A	จำนวนสัญญาณนาฬิกา 8080A
ADC DATA	CE YY	2	7	ACI DATA	7
ADC (HL)	8E	1	7	ADC M	7
ADC HL, RP	ED 01xx1010	2	15		
ADC (IX+DISP)	DD 8E YY	3	19		
ADC (IY+DISP)	FD 8E YY	3	19		
ADC REG	10001xxx	1	4	ADC REG	4
ADD DATA	C6 YY	2	7	ADI DATA	7
ADD (HL)	86	1	7	ADD M	7
ADD HL,RP	00xx1001	1	11	DAD RB	10
ADD (IX+DISP)	DD 86 YY	3	19		
ADD IX,PP	DD 00xx1001	2	15		
ADD (IY+DISP)	FD 86 YY	3	19		
ADD IY,RR	FD 00xx1001	2	15		
ADD REG	10000xxx	1	4	ADD REG	4
AND DATA	E6 YY	2	7	ANI DATA	7
AND (HL)	A6	1	7	ANA M	7
AND (IX+DISP)	DD A6 YY	3	19		
AND (IY+DISP)	FD A6 YY	3	19		
AND REG	10100xxx	1	4	ANA REG	4
BIT B,(HL)	CB 01bbb110	2	12		
BIT B,(IX+DISP)	DD CB YY 01bbb110	4	20		
BIT B,(IY+DISP)	FD CB YY 01bbb110	4	20		
BIT B,REG	CB 01bbbxxx	2	8		
CALL LABEL	CD ppqq	3	17	CALL LABEL	17
CALL C,LABEL	DC ppqq	3	10/17	CC LABEL	11/17
CALL M,LABEL	FC ppqq	3	10/17	CM LABEL	11/17
CALL NC,LABEL	D4 ppqq	3	10/17	CNC LABEL	11/17
CALL NZ,LABEL	C4 ppqq	3	10/17	CNZ LABEL	11/17
CALL P,LABEL	F4 ppqq	3	10/17	CP LABEL	11/17
CALL PE,LABEL	EC ppqq	3	10/17	CPE LABEL	11/17
CALL PO,LABEL	E4 ppqq	3	10/17	CPO LABEL	11/17
CALL Z,LABEL	CC ppqq	3	10/17	CZ LABEL	11/17
CCF	3F	1	4	CMC	4
CP DATA	FE YY	2	7	CPI DATA	7
CP (HL)	BE	1	7	CMP M	7
CP (IX+DISP)	DD BE YY	3	19		
CP (IY+DISP)	FD BE YY	3	19	CMP REG	19
CP REG	10111xxx	1	4		
CPD	ED A9	2	16		
CPDR	ED B9	2	21/16		
CPI	ED A1	2	16		
CPIR	ED B1	2	21/16		
CPL	2F	1	4	CMA	4
DAA	27	1	4	DAA	4
DEC (HL)	35	1	11	DCR M	10
DEC IX	DD 2B	2	10		
DEC (IX+DISP)	DD 35 YY	3	23		
DEC IY	FD 2B	2	10		
DEC (IY+DISP)	FD 35 YY	3	23		
DEC RP	00xx1011	1	6	DCX RP	5
DEC REG	00xxx101	1	4	DCR REG	5
DI	F3	1	4	DI	4
DJNZ DISP	10 YY	2	8/13		

สรุปคำสั่งที่อยู่ในรูปนีโมนิก รหัสภาษาเครื่องและช่วงเวลาในการประมวลผล (ต่อ)

คำสั่ง		ออบเจ็กต์โค้ด	จำนวนไบต์	สัญญาณนาฬิกา	รหัสนีโมนิก 8080A	จำนวนสัญญาณนาฬิกา 8080A
EI		FB	1	4	EI	4
EX	AF, AF'	08	1	4		
EX	DE,HL	EB	1	4	XCHG	4
EX	(SP),HL	E3	1	19	XTHL	18
EX	(SP),IX	DD E3	2	23		
EX	(SP),IY	FD E3	2	23		
EXX		D9	1	4		
HALT		76	1	4	HLT	4
IM	0	ED 46	2	8		
IM	1	ED 56	2	8		
IM	2	ED 5E	2	8		
IN	A,PORT	DB YY	2	10	IN PORT	10
IN	REG,(C)	ED	2	11		
		01ddd000				
INC	(HL)	34	1	11	IMR M	10
INC	IX	DD 23	2	10		
INC	(IX+DISP)	DD 34 YY	3	23		
INC	IY	FD 23	2	10		
INC	(IY+DISP)	FD 34 YY	3	23		
INC	RP	00xx0011	1	6	INX RP	5
INC	REG	00xxx100	1	4	INR REG	5
IND		ED AA	2	15		
INDR		ED BA	2	20/15		•
INI		ED A2	2	15		
INIR		ED B2	2	20/15		•
JP	LABEL	C3 ppqq	3	10	JMP LABEL	10
JP	C,LABEL	DA ppqq	3	10	JC LABEL	10
JP	(HL)	E9	1	4	PCHL	5
JP	(IX)	DD E9	2	8		
JP	(IY)	FD E9	2	8		
JP	M,LABEL	FA ppqq	3	10	JM LABEL	10
JP	NC,LABEL	D2 ppqq	3	10	JNC LABEL	10
JP	NZ,LABEL	C2 ppqq	3	10	JNZ LABEL	10
JP	P,LABEL	F2 ppqq	3	10	JP LABEL	10
JP	PE,LABEL	EA ppqq	3	10	JPE LABEL	10
JP	PO,LABEL	E2 ppqq	3	10	JPO LABEL	10
JP	Z,LABEL	CA ppqq	3	10	JZ LABEL	10
JR	C,DISP	38 YY	2	7/12		
JR	DISP	18 YY	2	12		
JR	NC,DISP	30 YY	2	7/12		
JR	NZ,DISP	20 YY	2	7/12		
JR	Z,DISP	28 YY	2	7/12		
LD	A,(ADDR)	3A ppqq	3	13	LDA ADDR	13
LD	A,(BC)	0A	1	7	LDAX B	7
LD	A,(DE)	1A	1	7	LDAX D	7
LD	A,I	ED 57	2	9		
LD	A,R	ED 5F	2	9		
LD	(ADDR),A	32 ppqq	3	13	STA ADDR	13
LD	(ADDR),BC	ED 43 ppqq	4	20		
LD	(ADDR),DE	ED 53 ppqq	4	20		
LD	(ADDR),HL	22 ppqq	3	16	SHLD ADDR	16
LD	(ADDR),IX	DD 22 ppqq	4	20		
LD	(ADDR),IY	FD 22 ppqq	4	20		
LD	(ADDR),SP	ED 73 ppqq	4	20		
LD	(BC),A	02	1	7	STAX B	7
LD	(DE),A	12	1	7	STAX D	7
LD	HL,(ADDR)	2A ppqq	3	16	LHLD ADDR	16
LD	(HL),DATA	36 YY	2	10	MVI M,DATA	10

สรุปคำสั่งที่อยู่ในรูปนีโมนิก รหัสภาษาเครื่องและช่วงเวลาในการประมวลผล (ต่อ)

คำสั่ง	ออบเจ็กต์โค้ด	จำนวนไบต์	สัญญาณนาฬิกา	รหัสนีโมนิก 8080A	จำนวนสัญญาณนาฬิกา 8080A
LD (HL), REG	01110sss	1	7	MOV M,REG	7
LD I,A	ED 47	2	9		
LD IX,(ADDR)	DD 2A ppqq	4	20		
LD IX,DATA16	DD 21 YYYY	4	14		
LD (IX+DISP),DATA	DD 36 YY YY	4	19		
LD (IX+DISP),REG	DD 01110sss YY	3	19		
LD IY,(ADDR)	FD 2A ppqq	4	20		
LD IY,DATA16	FD 21 YYYY	4	14		
LD (IY+DISP),DATA	FD 36 YYYY	4	19		
LD (IY+DISP),REG	FD 01110sss YY	3	19		
LD R,A	ED 4F	2	9		
LD REG,DATA	00ddd110 YY	2	7	MVI REG,DATA	7
LD REG,(HL)	01ddd110	1	7	MOV REG,M	7
LD REG,(IX+DISP)	DD 01ddd110 YY	3	19		
LD REG,(IY+DISP)	FD 01ddd110 YY	3	19		
LD REG,REG	01dddsss	1	4	MOV REG,REG	5
LD RP,(ADDR)	ED 01xx1011 ppqq	4	20		
LD RP,DATA16	00xx0001 YYYY	3	10	LXI RP,DATA16	10
LD SP,HL	F9	1	6	SPHL	5
LD SP,IX	DD F9	2	10		
LD SP,IY	FD F9	2	10		
LDD	ED A8	2	16		
LDDR	ED B8	2	21/16		•
LDI	ED A0	2	16		
LDIR	ED B0	2	21/16		•
NEG	ED 44	2	8		
NOP	00	1	4	NOP	4
OR DATA	F6 YY	2	7	ORI DATA	7
OR (HL)	B6	1	7	ORA M	7
OR (IX+DISP)	DD B6 YY	3	19		
OR (IY+DISP)	FD B6 YY	3	19		
OR REG	10110xxx	1	4	ORA REG	5
OUT (C),REG	ED 01sss001	2	12		
OUT PORT,A	D3 YY	2	11	OUT PORT	10
OUTD	ED AB	2	15		
OUTDR	ED BB	2	20/15		•
OUTI	ED A3	2	15		
OUTIR	ED B3	2	20/15		•
POP IX	DD E1	2	14		
POP IY	FD E1	2	14		
POP PR	11xx0001	1	10	POP RP	10
PUSH IX	DD E5	2	15		
PUSH IY	FD E5	2	15		
PUSH PR	11xx0101	1	11	PUSH RP	11
RES B,(HL)	CB 10bbb110	2	15		
RES B,(IX+DISP)	DD CB YY 10bbb110	4	23		

สรุปคำสั่งที่อยู่ในรูปนีโมนิก รหัสภาษาเครื่องและช่วงเวลาในการประมวลผล (ต่อ)

คำสั่ง		ออบเจ็กต์โค้ด	จำนวนไบต์	สัญญาณนาฬิกา	รหัสนีโมนิก 8080A	จำนวนสัญญาณนาฬิกา 8080A
RES	B,(IY+DISP)	FD CB YY 10bbb110	4	23		
RES	B,REG	CB 10bbbxxx	2	8		
RET		C9	1	10	RET	10
RET	C	D8	1	5/11	RC	5/11
RET	M	F8	1	5/11	RM	5/11
RET	NC	D0	1	5/11	RNC	5/11
RET	NZ	C0	1	5/11	RNZ	5/11
RET	P	F0	1	5/11	RP	5/11
RET	PE	E8	1	5/11	RPE	5/11
RET	PO	E0	1	5/11	RPO	5/11
RET	Z	C8	1	5/11	RZ	5/11
RETI		ED 4D	2	14		
RETN		ED 45	2	14		
RL	(HL)	CB 16	2	15		
RL	(IX+DISP)	DD CB YY 16	4	23		
RL	(IY+DISP)	FD CB YY 16	4	23		
RL	REG	CB 00010xxx	2	8		
RLA		17	1	4	RAL	4
RLC	(HL)	CB 06	2	15		
RLC	(IX+DISP)	DD CB YY 06	4	23		
RLC	(IY+DISP)	FD CB YY 06	4	23		
RLC	REG	CB 00000xxx	2	8		
RLCA		07	1	4	RLC	4
RLD		ED 6F	2	18		
RR	(HL)	CB 1E	2	15		
RR	(IX+DISP)	DD CB YY 1E	4	23		
RR	(IY+DISP)	FD CB YY 1E	4	23		
RR	REG	CB 00011xxx	2	8		
RRA		1F	1	4	RAR	4
RRC	(HL)	CB 0E	2	15		
RRC	(IX+DISP)	DD CB YY 0E	4	23		
RRC	(IY+DISP)	FD CB YY 0E	4	23		
RRC	REG	CB 00001xxx	2	8		
RRCA		0F	1	4	RRC	4
RRD		ED 67	2	18		
RST	N	11xxx111	1	11	RST N	11
SBC	DATA	DE YY	2	7	SBI DATA	7
SBC	(HL)	9E	1	7	SBB M	7
SBC	HL,RP	ED 01xx0010	2	15		
SBC	(IX+DISP)	DD 9E YY	3	19		
SBC	(IY+DISP)	FD 9E YY	3	19		
SBC	REG	10011xxx	1	4	SBB REG	4
SCF		37	1	4	STC	4
SET	B,(HL)	CB 11bbb110	2	15		
SET	B,(IX+DISP)	DD CB YY 11bbb110	4	23		
SET	B,(IY+DISP)	FD CB YY 11bbb110	4	23		
SET	B,REG	CB 11bbbxxx	2	8		

สรุปคำสั่งที่อยู่ในรูปนีโมนิก รหัสภาษาเครื่องและช่วงเวลาในการประมวลผล (ต่อ)

คำสั่ง		ออบเจ็กต์โค้ด	จำนวนไบต์	สัญญาณนาฬิกา	รหัสนีโมนิก 8080A	จำนวนสัญญาณนาฬิกา 8080A
SLA	(HL)	CB 26	2	15		
SLA	(IX+DISP)	DD CB YY 26	4	23		
SLA	(IY+DISP)	FD CB YY 26	4	23		
SLA	REG	CB 00100xxx	2	8		
SRA	(HL)	CB 2E	2	15		
SRA	(IX+DISP)	DD CB YY 2E	4	23		
SRA	(IY+DISP)	FD CB YY 2E	4	23		
SRA	REG	CB 00101xxx	2	8		
SRL	(HL)	CB 3E	2	15		
SRL	(IX+DISP)	DD CB YY 3E	4	23		
SRL	(IY+DISP)	FD CB YY 3E	4	23		
SRL	REG	CB 00111xxx	2	8		
SUB	DATA	D6 YY	2	7	SUI DATA	7
SUB	(HL)	96	1	7	SUB M	7
SUB	(IX+DISP)	DD 96 YY	3	19		
SUB	(IY+DISP)	FD 96 YY	3	19		
SUB	REG	10010xxx	1	4	SUB REG	4
XOR	DATA	EE YY	2	7	XRI DATA	7
XOR	(HL)	AE	1	7	XRA M	7
XOR	(IX+DISP)	DD AE YY	3	19		
XOR	(IY+DISP)	FD AE YY	3	19		
XOR	REG	10101xxx	1	4	XRA REG	4

หมายเหตุ

- x แทนข้อมูลที่เป็นบิต
- bbb แทนตัวเลขแบบ BCD ที่ใช้กำหนดตำแหน่งบิตในรีจิสเตอร์หรือหน่วยความจำ
- ddd แทนตัวเลขแบบ BCD ที่ใช้กำหนดรีจิสเตอร์ปลายทาง
- sss แทนตัวเลขแบบ BCD ที่ใช้กำหนดรีจิสเตอร์ต้นทาง
- pppqqq แทนตัวเลขฐานสิบหก 4 ตัว สำหรับแอดเดรสของหน่วยความจำ
- yy แทนตัวเลขฐานสิบหก 2 ตัว สำหรับข้อมูล
- yyyy แทนตัวเลขฐานสิบหก 4 ตัว สำหรับข้อมูล

เมื่อปรากฏตัวเลขของการทำงาน 2 ค่า เช่น 5/11 ในกรณีนี้ขึ้นอยู่กับเงื่อนไขในแฟล็ก

*ช่วงเวลาที่กำหนดนี้เป็นช่วงเวลาการทำงานเพียง 1 รอบ