ปีการศึกษา 2532

เครื่องควบคุมการเลือกสัญญาณอินพุท-เอาท์พุท

(programmable I/O selector)

โดย

นาย กวีวุธ นุตาคม

นาย สายัณห์ แซ่ซ้ง

นาย พูนเสริม แสงทอง

อาจารย์ที่ปรึกษา

อาจารย์ ดร. ณก เจนจิระพงศ์เวช

22 พ.ย. 2533

027007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทคัดย่อ

บทที่ 1 เครื่องควบคุมการเลือกสัญญาณอินพุทเอาท์พุท 1

บทที่ 2 การออกแบบวงจร 4

บทที่ 3 การเขียนโปรแกรมควบคุมระบบ (Monitor Program) 13

บทที่ 4 การใช้งาน 31

ภาคผนวก ก โปรแกรมของระบบ

ภาคผนวก ข การใช้ 8255 กับ Z-80

ภาคผนวก ค Z-80 และ 4066

บรรณานุกรม

กิตติกรรมประกาศ

027007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมการเลือกสัญญาณอินพุท-เอาท์พุท

นายกวิวุธ นุตาคม
นายสายัณห์ แซ่ชั่ง
นายบุญเสริม แสงทอง
ดร.ภนก เจนจิระพงศ์ (อาจารย์ที่ปรึกษา)
ปี พ.ศ. 2532

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้เป็นเรื่องเกี่ยวกับการประยุกต์ใช้งานของไมโครโปรเซสเซอร์ เบอร์ Z-80 ในการควบคุมสวิทช์เลือกช่องสัญญาณขนาด 8 ช่องสัญญาณเข้าและ 8 ช่องสัญญาณออก โดยเป็นแบบแมทริกซ์สวิทช์ขนาด 8x8 ซึ่งสามารถบันทึกรูปแบบการต่อช่องสัญญาณการทำงานได้ 3 โปรแกรม หรือมากกว่านี้ขึ้นอยู่กับการเพิ่มสวิทช์และโปรแกรม การตัดต่อช่องสัญญาณนั้นอาศัยหลักการกดสวิทช์คู่อินพุท-เอาท์พุท และจะใช้อนาล็อกสวิทช์เป็นตัวตัดต่อช่องสัญญาณ หรือใช้รีเลย์ในการตัดต่อช่องสัญญาณ แต่ในปริญญานิพนธ์นี้ใช้อนาล็อกสวิทช์เป็นตัวผ่านสัญญาณ ลักษณะงานที่นำไปใช้คือการนำไปใช้ลดปัญหาความยุ่งยากในการถอดเข้าถอดออกของอุปกรณ์บ่อย ๆ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PROGRAMMABLE I/O SELECTOR

Mr.KAWEEWUT NUTAKOM

Mr.SAYAN SAECHUNG

Mr.BOONSERM SANGTHONG

Dr.KANOK JAENJIRAPONGWECH (ADVISOR)

YEAR 1989

ABSTRACT

This thesis is the application of Z-80 microprocessor for control the selector switches to selected the 8x8 channels of signal. These switches are arranged by 8x8 matrix switches,also three programs in their memory or more depend on the monitor program.The principle of this machine used pressing the input and output switches,may it used the CMOS analog switch or relay for passing the signal.This machine is designed for reduce the trouble of connection and disconnection of signal's connector.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

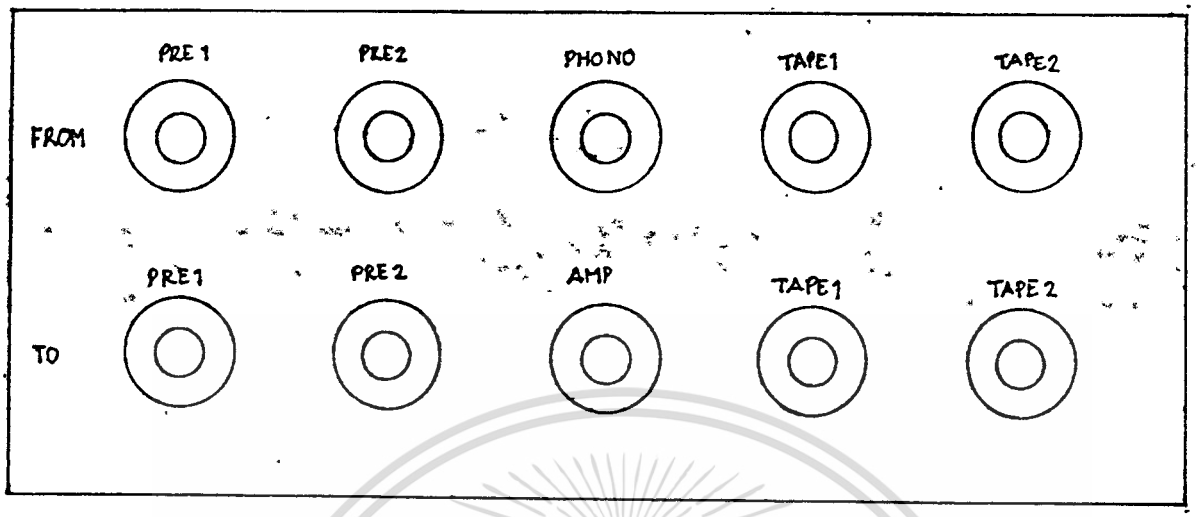
เครื่องควบคุมการเลือกสัญญาณอินพุทเอาท์พุท

(programmable i/o selector)

ในการทำงานเกี่ยวกับอุปกรณ์หลาย ๆ ชนิด และสัญญาณจากเครื่องมือเครื่องใช้ต่างๆ ในห้องปฏิบัติการ ห้อง studio ทางด้านเสียง จะต้องมีการต่ออุปกรณ์สัญญาณต่างๆ เหล่านี้เข้ากับเครื่องมือ เครื่องทดสอบ หรือเครื่องขยาย อย่างแน่นอน ซึ่งถ้ามีการใช้งานมากขึ้นเท่าใด หรือมีการใช้งานบ่อยครั้งเพียงใด ก็ยิ่งจะเกิดความยุ่งยาก สับสน และน่าเบื่อหน่ายในการที่จะตัดหรือต่อสัญญาณ จากอุปกรณ์หนึ่งไปยังอุปกรณ์หนึ่ง เราจะต้องเสี่ยงอันตรายจากอุบัติเหตุที่อาจจะเกิดขึ้นกับเครื่องมือถ้าเกิดมีการต่อผิดพลาดขึ้น และยังเป็นเรื่องยุ่งยากมากขึ้นถ้าเครื่องมือและอุปกรณ์ต่าง ๆ เหล่านั้นติดตั้งอยู่บน rack อาจจะยากมากที่จะถอดเปลี่ยน connector ด้านหลังของเครื่อง และปัญหาเหล่านี้มักจะเป็นเรื่องใหญ่สำหรับ sound studio มากกว่าห้องปฏิบัติการ เพราะว่าเครื่องมือต่างๆในห้องปฏิบัติการส่วนใหญ่จะมี connector อยู่ด้านหน้า และไม่มีการเปลี่ยนแปลงการเชื่อมต่อบ่อยนัก ตรงกันข้ามกับ sound studio ซึ่งมี connector อยู่ด้านหลังและส่วนใหญ่จะติดตั้งบน rack หรือในกรณีห้องฟังจะติดตั้งอยู่อย่างมิดชิดมาก ดังนั้นการที่จะสับเปลี่ยนอุปกรณ์ต่างๆนั้นก็จะยิ่งทำได้ลำบาก แต่อย่างไรก็ตามความพยายามที่จะแก้ปัญหาต่างๆเหล่านี้ก็ได้ทำให้มีการสร้างอุปกรณ์อำนวยความสะดวกขึ้นมาที่เรียกว่า patch bay

patch bay เป็นการแก้ปัญหาความยุ่งยากในการตัดต่อสัญญาณหรือเปลี่ยนแปลงการใช้งานโดยการใช้วิธีต่อสายนำสัญญาณเข้าและออก จากแต่ละเครื่องที่ใช้งานมาไว้บน panel ซึ่งมี connector สำหรับใช้กับ microphone ติดตั้งอยู่ และ panel นี้อาจติดตั้งไว้บน rack ก็ได้ ลักษณะการใช้งานจะคล้ายกับการใช้งานตู้สาขาโทรศัพท์แบบ manual โดยการใช้สายสัญญาณที่ปลายทั้งสองข้างมี connector microphone ชนิดตัวผู้ต่ออยู่ เราจะนำปลายข้างหนึ่งต่อกับช่องสัญญาณ output ที่ต้องการและอีกปลายข้างหนึ่งต่อกับช่อง input ของอีกเครื่องตามรูปที่ 1. นับว่าเป็นการพัฒนาการใช้งานอุปกรณ์เพื่ออำนวยความสะดวกได้ดีมาก แต่อย่างไรก็ตาม patch bay ก็ยังมีจุดบกพร่องอยู่บ้าง เช่น ความยุ่งยากและสับสนเมื่อเวลาจะเปลี่ยนแปลงช่องต่อสัญญาณ อาจทำให้ผู้ใช้งานงงและผิดพลาดได้ถ้ามีการเปลี่ยนแปลงบ่อยๆและเมื่อใช้งานนานเข้า connector แบบนี้อาจเกิด oxide ขึ้นที่บริเวณที่ใช้สัมผัสกัน ทำให้สัญญาณเดินทางไม่สะดวกเท่าที่ควร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

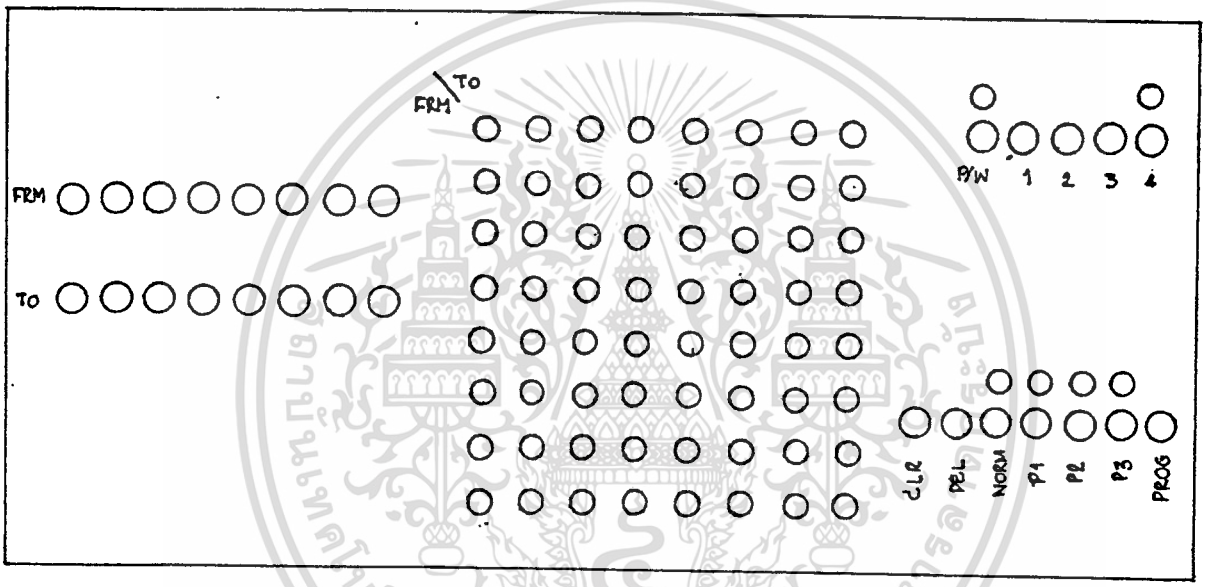


ลักษณะของ Patch Bay

เนื่องแก้ปัญหาดังกล่าวนี้ ทำให้เกิดแนวความคิดที่ว่าควรจะสร้างอุปกรณ์ Electronic มาใช้ในการควบคุมการต่อช่องสัญญาณต่างๆ และใช้ควบคุมการทำงานด้วยระบบ Microprocessor เนื่องความมุ่งต่องสะดวกรวดเร็วในการทำงานและยังสามารถสร้างให้มีการโปรแกรมระบบที่มีการใช้งานบ่อยหรือเปลี่ยนแปลงบ่อยๆเข้าไว้ในเครื่องได้ จากสาเหตุนี้จึงทำให้เกิดเครื่องควบคุมการเลือกสัญญาณ I/O หรือ Progammable I/O Selector ขึ้น ซึ่งหลักการของเครื่องก็อาศัยหลักการเช่นเดียวกับ Patch Bay คือการนำช่องสัญญาณ Input และ Output ของเครื่องมือแต่ละเครื่องต่อเข้าที่เครื่องควบคุมนี้ โดยกำหนดการเชื่อมต่อของสัญญาณเป็นแบบ Matrix คือแบ่งเป็นแถวของ Input และแถวของ Output และใช้เลือกโดยกด Switch Input กับกด Switch Output ช่องที่ต้องการ Microprocessor จะทำการตรวจสอบการกดและส่งสัญญาณไปควบคุมวงจร Electronic ให้ต่อช่องสัญญาณตามที่เราต้องการ ระบบนี้ และในเครื่องควบคุมนี้ยังมีการป้องกันการเลือกของสัญญาณ Output ซ้ำกันในการใช้งานครั้งเดียวกันได้ ซึ่งหมายความว่าเครื่องหนึ่งจะรับสัญญาณเข้าได้เพียง

เอกสารนี้เป็นเอกสารที่สร้างขึ้นสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1สัญญาณ ต่อ 1 ช่อง Input เท่านั้น แต่สัญญาณออกจากเครื่องแต่ละเครื่องสามารถที่จะจ่ายไปยังเครื่องอื่นๆ ได้หลายเครื่อง และในเครื่องควบคุมนี้ยังสามารถ Program การเลือกสัญญาณหรือการต่อช่องสัญญาณไว้ได้ 3 Program ซึ่งทำให้สะดวกมากเวลาใช้งาน โดยเพียงแต่เลือก Program ที่ต้องการ เครื่องก็จะทำการต่อระบบการใช้งานที่ต้องการให้ทันทีและยังมีความพิเศษข้อหนึ่งคือ มีการใช้รหัสผ่านเพื่อป้องกันผู้ไม่ได้รับอนุญาตเข้าไปใช้งานเครื่องควบคุมการเลือกสัญญาณ I/O นี้ก็ยังเป็นเครื่องที่ต้องการ การปรับปรุงให้ดียิ่งขึ้นต่อไปเพื่อการใช้งานที่เหมาะสม



หน้าตาของเครื่อง PROGRAMMABLE I/O SELECTOR

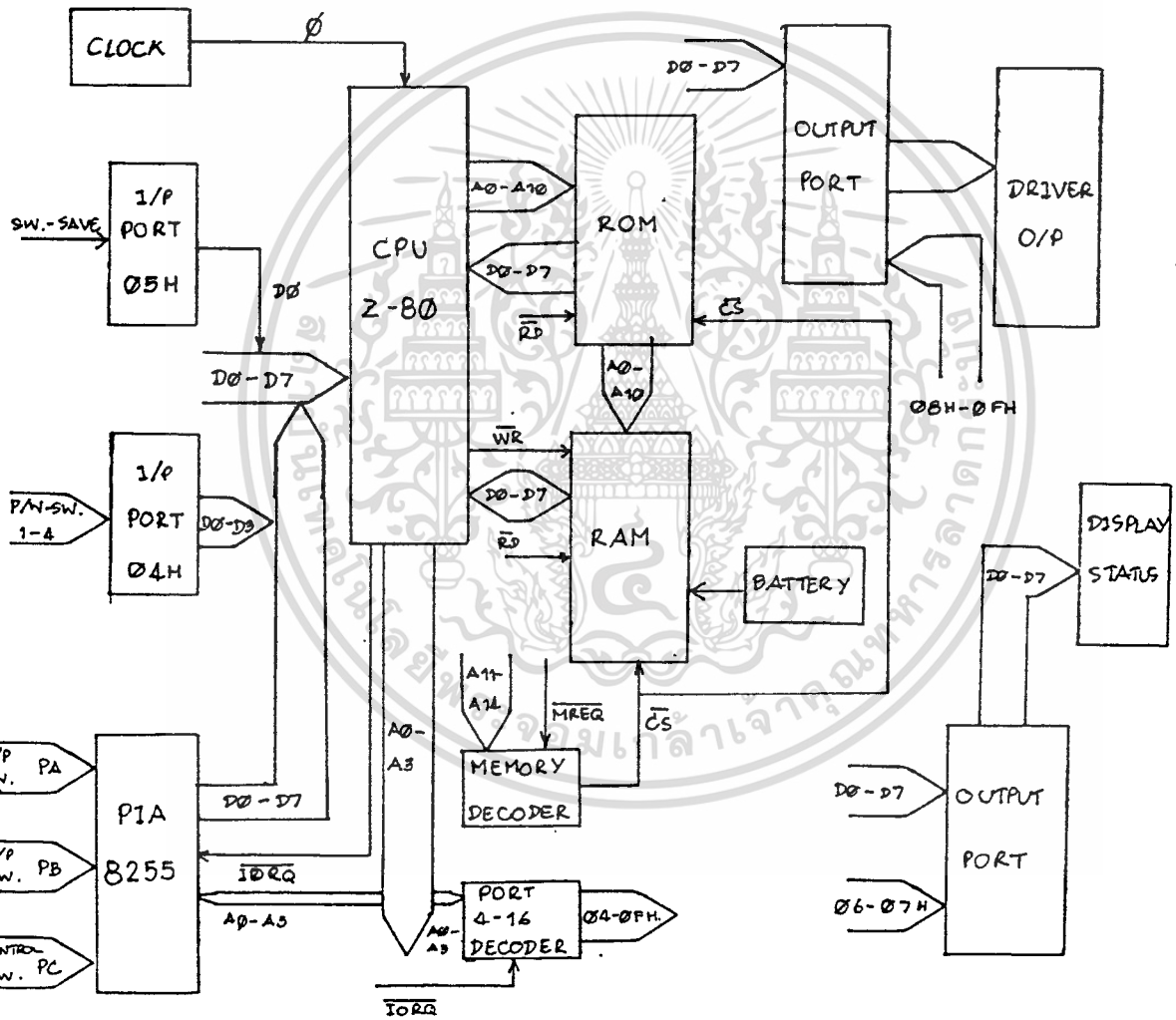
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

การออกแบบวงจร (HARD WARE)

2.1 โครงสร้างการทำงานของระบบไมโครโปรเซสเซอร์

รายละเอียดของระบบไมโครโปรเซสเซอร์แสดงได้ดังรูปข้างล่าง (วงจรจริงแสดงในภาคผนวกที่ 1)

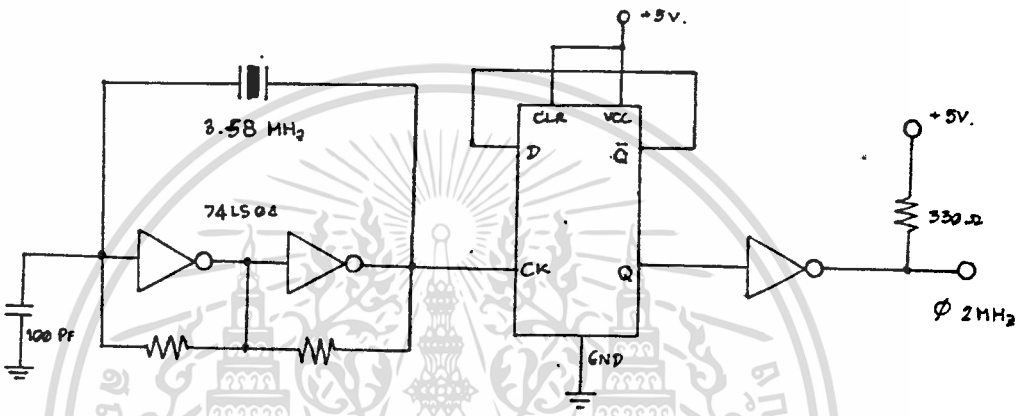


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในส่วนของระบบไมโครโปรเซสเซอร์จะประกอบด้วย

2.1.1 ตัวไมโครโปรเซสเซอร์ใช้เบอร์ Z-80 ทำหน้าที่เป็นส่วนประมวลผลกลางทั้งหมด

2.1.2 สัญญาณนาฬิกา (clock) ที่กำหนดให้ Z-80 CPU เพื่อให้ ซีพียู ทำงานได้วงจรที่ออกไว้เป็น astable mutlivibrator มีความถี่ 2 เมกกะเฮิร์ท ได้จากวงจรออสซิเลทเตอร์ (oscilator)ที่มี crystalเป็นตัวควบคุมความถี่ออสซิเลทเตอร์ที่ความถี่ 3.58เมกกะเฮิร์ท จากนั้นป้อนเข้าวงจร divider (IC 74LS74) เพื่อให้ได้ความถี่ 2 เมกกะเฮิร์ท



รูปแสดงวงจรสร้างความถี่ 2 M_{HZ}

2.1.3 หน่วยความจำ (MEMORY)

หน่วยความจำที่ออกแบบไว้ในระบบไมโปรเซสเซอร์นี้ แบ่งออกเป็น 2 ชนิดคือ

1.หน่วยความจำแบบอ่านอย่างเดียว (Read Only Memory หรือ ROM)เราจะเขียนข้อมูลเข้าไปในหน่วยความจำชนิดนี้เพียงครั้งเดียวและข้อมูลนั้นจะอยู่ตลอดไป แม้จะไม่มีไฟเลี้ยงก็ตาม เวลาใช้งาน ซีพียู จะอ่านข้อมูลออกมาเพียงอย่างเดียวใช้เป็นที่เก็บโปรแกรมหลัก (Monitor program) คือโปรแกรมควบคุมการทำงานของระบบการตัดต่อช่องสัญญาณหน่วยความจำชนิดนี้ใช้ไอซีเบอร์ EPROM 2716 ซึ่งมีความจุ 2K*8 ไปท์

2.หน่วยความจำแบบแรนดอม (Rendom Access Memory หรือ RAM)เป็นหน่วยความจำชนิดที่เก็บข้อมูลได้ และขณะเดียวกัน ซีพียู สามารถจะเปลี่ยนแปลงข้อมูลในหน่วยความนั้นได้หรือกล่าวอีกนัยหนึ่งคือเป็นหน่วยความจำที่อ่านออกมาและเขียนเข้าไปได้(Reed Write Memory) ตลอดเวลาที่ไฟเลี้ยง (supply) อยู่ถ้าเราปิดเครื่องข้อมูลเหล่านั้นจะหายไป แต่ในการออกแบบครั้งนี้ได้มีส่วนของแบตเตอรี่ (Battery)เพิ่มเติมลงไป ทำให้หน่วยความจำนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

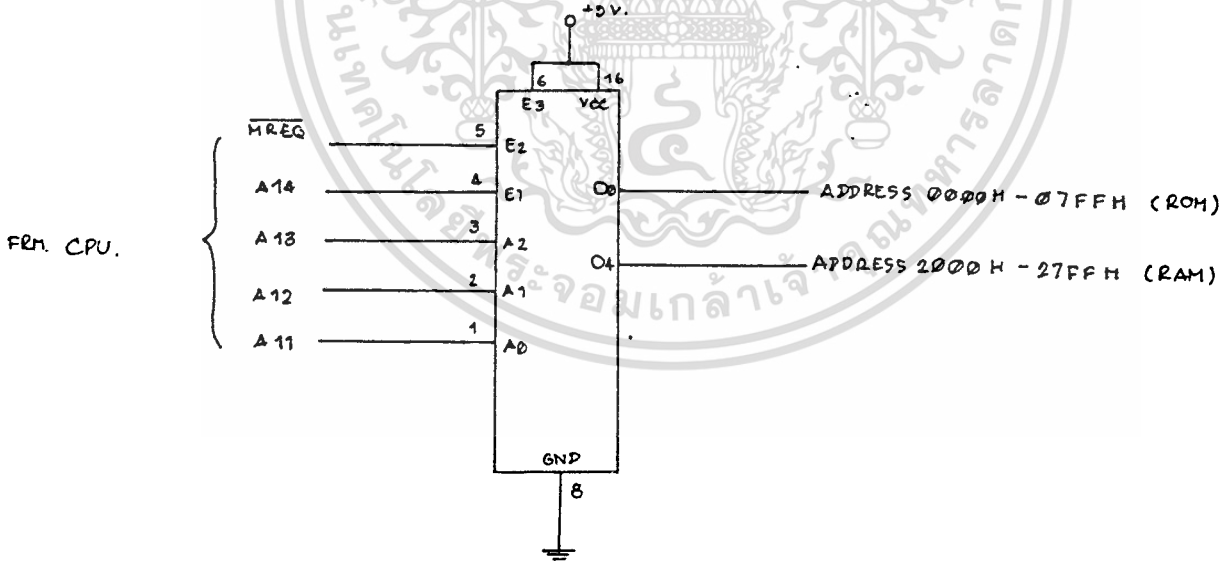
ข้อมูลจะคงอยู่ตลอดเวลาเพราะไฟเลี้ยงจากแบตเตอรี่ หน่วยความจำนี้จะใช้เก็บข้อมูลชั่วคราวบางอย่างในการทำงานของโปรแกรม และเนื้อที่ส่วนใหญ่จะใช้เก็บข้อมูลของการกด sw และข้อมูลของการเก็บสถานะการต่อช่องสัญญาณของแต่ละช่องสัญญาณนั้นไว้หน่วยความจำชนิดนี้ใช้ไอซีเบอร์ RAM 6116 ซึ่งมีความจุ 2K*8 ไปท์

2.1.4 วงจรถอดรหัสหน่วยความจำ (Memory Decoder)

เนื่องจากระบบไมโครโปรเซสเซอร์มีหน่วยความจำ 2 อย่าง จำเป็นต้องมีวงจรแยกตำแหน่งของหน่วยความจำทั้ง 2 ชนิดดังนี้

- ตำแหน่ง (Address) 0000H ถึง 07FFH เป็นตำแหน่งที่ ซีพียู ใช้ติดต่อกับรอม (ROM)
- ตำแหน่ง (Address) 2000H ถึง 27FFH เป็นตำแหน่งที่ ซีพียู ใช้ติดต่อกับแรน (RAM)

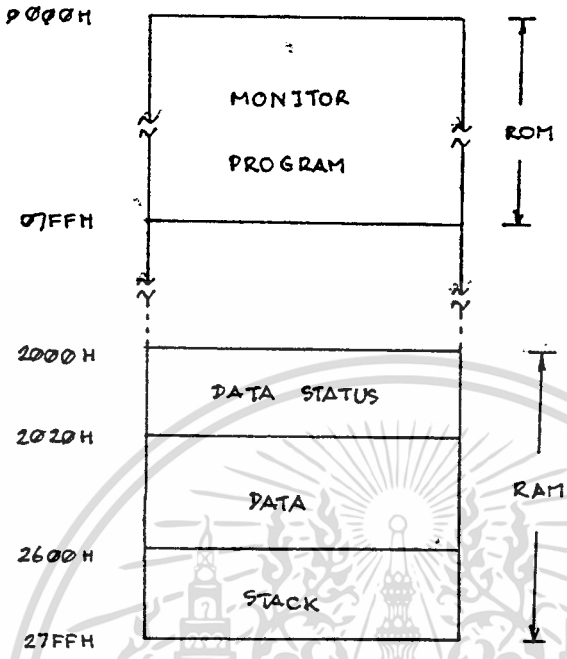
การแยกตำแหน่งนี้ใช้การควบคุมของสัญญาณจาก Z-80 CPU คือ $A_{11}, A_{12}, A_{13}, A_{14}$ และ MERQ เป็นตัวกำหนดโดยใช้ IC 74LS138 เป็นตัวแยกดังรูป



รูปแสดงวงจรถอดรหัสหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

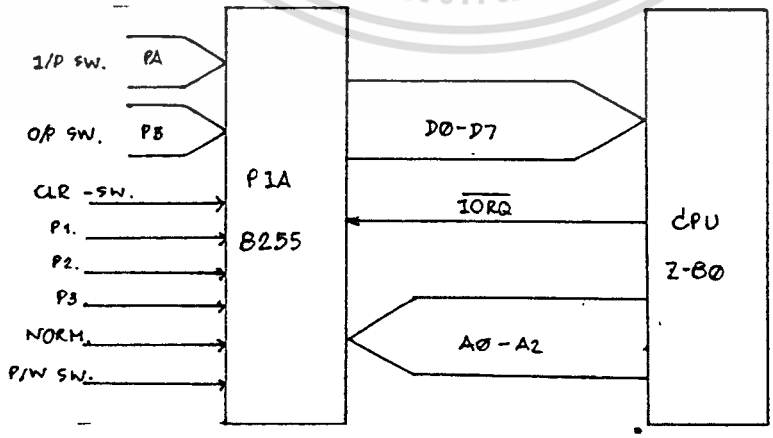
จากการกำหนดตำแหน่งของหน่วยความจำทำให้สามารถแสดงตารางการใช้งานของหน่วยความจำ (Memory Map) ดังรูป



รูปแสดงตารางการใช้งานของหน่วยความจำ

2.1.5 ส่วนเชื่อมต่อระหว่างระบบไมโครโปรเซสเซอร์กับ PIA (8255)

ส่วนนี้จะให้ทำหน้าที่ในการตรวจเช็คสวิทช์ (switch) ของอินพุทสวิทช์และของเอ้าพุทสวิทช์ และสวิทช์ควบคุมการทำงาน (Control Switch) ซึ่งแสดงดังรูป



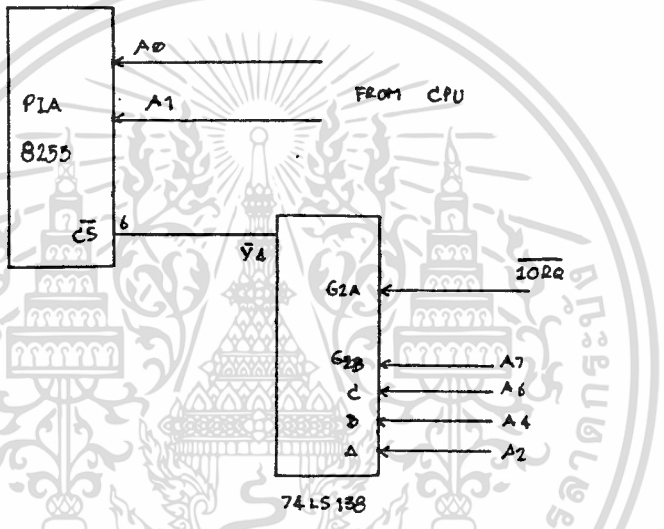
รูปแสดงส่วนของการเชื่อมต่อระหว่าง CPU กับ 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การออกแบบการต่อ 8255 เข้ากับ CPU Z-80

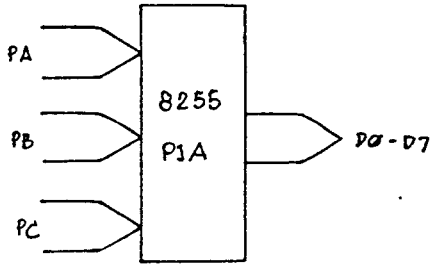
เราใช้สัญญาณต่างๆ ที่เกิดขึ้นจะเหมือนกับขบวนการติดต่อกับ I/O โดยจะต้องเอาสัญญาณ A_0-A_7 จาก Z-80 มาถอดรหัสเพื่อสร้างสัญญาณเลือกพอร์ท แต่เนื่องจาก 8255 มีขา Address Input อยู่แล้ว 2 ขา (A_0, A_1) ซึ่งโดยปกติแล้วขา A_0, A_1 จากบัสแอดเดรสและเราใช้ A_2-A_7 ในการถอดรหัสเพื่อทำสัญญาณเลือกชิพ (Chip Select) ให้แก่ 8255

ในระบบนี้เราได้ออกแบบไว้ที่ port 60H-63H การถอดรหัสจะแสดงไว้ดังรูป



รูปแสดงวงจรถอดรหัสport 60H-63H

ลักษณะการใช้งาน port ทั้ง 3 ของ 8255 กำหนดให้ทำงานอยู่ในโหมด 0 โดยให้ port ทั้ง 3 เป็นอินพุททั้งหมด



รูปแสดง MODE 0 control word = 9BH

การกำหนดค่าให้กับ Register ควบคุมของ 8255 นั้นดูได้ในภาคผนวกที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนสัญญาณเข้า (INPUT) ของ 8255 นั้น ส่วนนี้จะเป็นการตรวจสอบการกด switch ของทั้ง 3 port เพื่อส่งต่อไปยัง ซีพียู โดยจะใช้จุดสัญญาณเข้ามาควบคุมความสัมพันธ์ของข้อมูลดังรูป

กำหนด port A แทน i/p ของ sw-i/p

port B แทน i/p ของ sw-o/p

port C แทน i/p ของ sw-control

sw-I/P	DATA
SW_1	D_0
SW_2	D_1
SW_3	D_2
SW_4	D_3
SW_5	D_4
SW_6	D_5
SW_7	D_6
SW_8	D_7

sw-O/P	DATA
SW_1	D_0
SW_2	D_1
SW_3	D_2
SW_4	D_3
SW_5	D_4
SW_6	D_5
SW_7	D_6
SW_8	D_7

027007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

sw-control	DATA
sw-clear	D_0
sw-program1	D_1
sw-program2	D_2
sw-program3	D_3
sw-DEL	D_4
sw-Normal	D_5
sw-password	D_6

รูปแสดงความสัมพันธ์ของสัญญาณเข้ากับข้อมูล D_0-D_7 ของ 8255

ส่วนสัญญาณเข้า (I/P) ของ port 04H-05H ส่วนนี้จะรับการตรวจสอบการกด switch ของ switch code pass word กับ switch save ซึ่งสามารถแสดงความสัมพันธ์ระหว่างสัญญาณเข้ากับข้อมูล (DATA) ดังรูป ในส่วนนี้เราใช้ IC 74LS125

port 04H

SW_1	D_0
SW_2	D_1
SW_3	D_2
SW_4	D_3

ตารางแสดงความสัมพันธ์ sw code pass word กับ D_0-D_3

port 05H

SW_1	D_0

ตารางแสดงความสัมพันธ์ sw-save กับ D_0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.6 ส่วนเชื่อมต่อระหว่างไมโครโปรเซสเซอร์กับ port 06H-07H

1. จะใช้ในการแสดงสถานะ (status display) ของ การทำงานโดยใช้ port 06H-07H โดยแยก

- port 06H ในการแสดงสถานะดังตาราง

DATA	แสดงสถานะ
D_0	กด pass word
D_1	ไม่ผ่าน pass word

-port 07H ในการแสดงสถานะดังตาราง

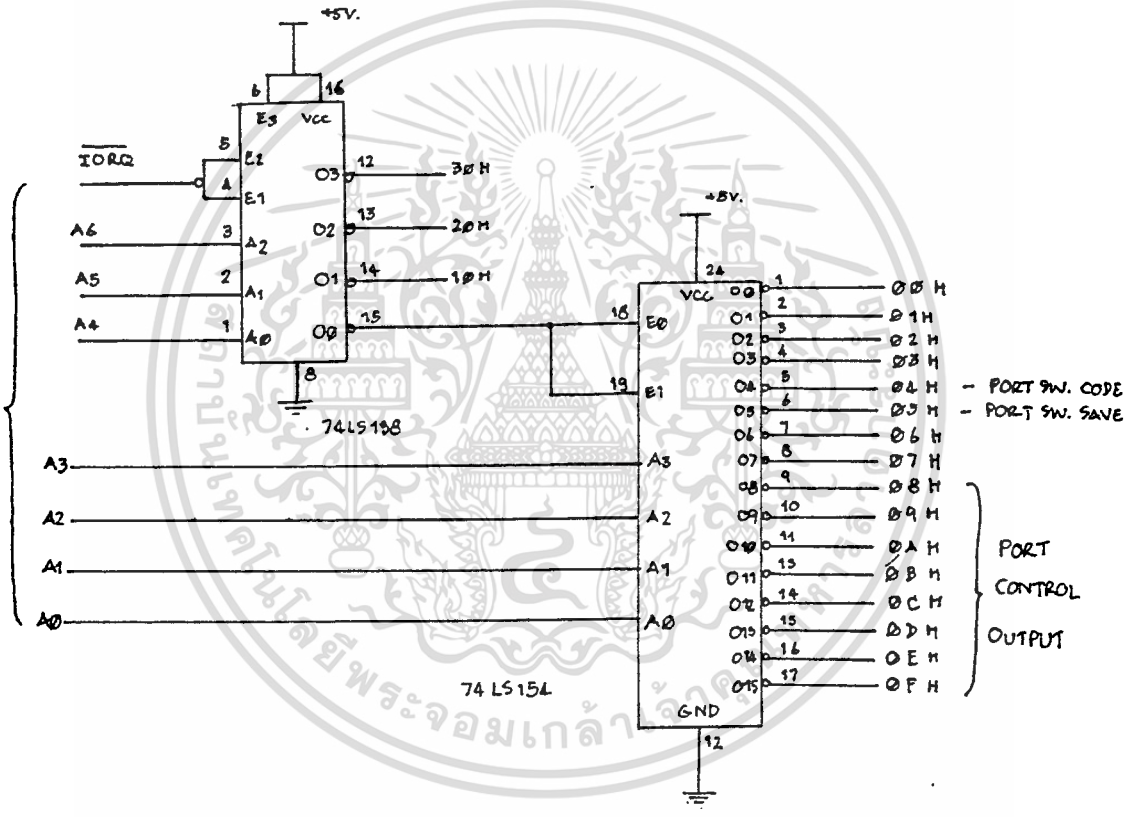
DATA	แสดงสถานะ
D_0	การ save
D_1	prom 1
D_2	prom 2
D_3	prom 3
D_4	DEL
D_5	normal

ตารางแสดงความสัมพันธ์ของสัญญาณ o/p กับ D_0-D_5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.port 08H-0FH เป็น port output

จะใช้ในการแสดงสถานะของการตัดต่อช่องสัญญาณและเป็น port o/p ที่ใช้ในการควบคุมการตัดต่อช่องสัญญาณโดยจะไปควบคุมการขับ relay อีกที่หนึ่ง การถอดรหัสของ port 06H-0FH การแบ่งแยกข้อมูลต่างๆ จะใช้จุดสัญญาณเข้าและจุดสัญญาณออกที่แตกต่างไป โดยแต่ละจุดออกจากกัน ใช้สัญญาณ A_0 - A_6 และ IORQ ทำงานร่วมกับ IC 74LS138 กับ IC 74LS154 ดังรูป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

โปรแกรมควบคุมระบบของเครื่องควบคุมการเลือกสัญญาณ I/O

(Monitor Program of Programable I/O Selector)

การใช้ไมโครโปรเซสเซอร์ในการควบคุมการเลือกสัญญาณ I/O นั้น ส่วนสำคัญที่จะทำให้เครื่องควบคุมมีคุณสมบัติการใช้งานที่ดีนั้น คือ ส่วนของโปรแกรมควบคุมระบบ (เปรียบเสมือมหัวใจในการทำงานซึ่งจะดีมากเพียงใดก็ขึ้นอยู่กับความสามารถของโปรแกรมเท่านั้น

3.1 โครงสร้างของโปรแกรม

โปรแกรมหลักที่ใช้ในเครื่องควบคุมการเลือกสัญญาณ I/O จะมีอยู่ 2 ส่วน คือ

1. โปรแกรมตรวจสอบรหัสผ่าน (check pass word program) ทำหน้าที่ควบคุมการกด switch การใช้เครื่องโดยมีการตรวจสอบรหัสผ่าน (check pass word)
2. โปรแกรมหลัก (Main Program) ทำหน้าที่ควบคุมการตรวจเช็ค switch เพื่อทำงานตามขั้นตอนที่ได้กำหนดเอาไว้

3.2 อธิบายรายละเอียดของโปรแกรม

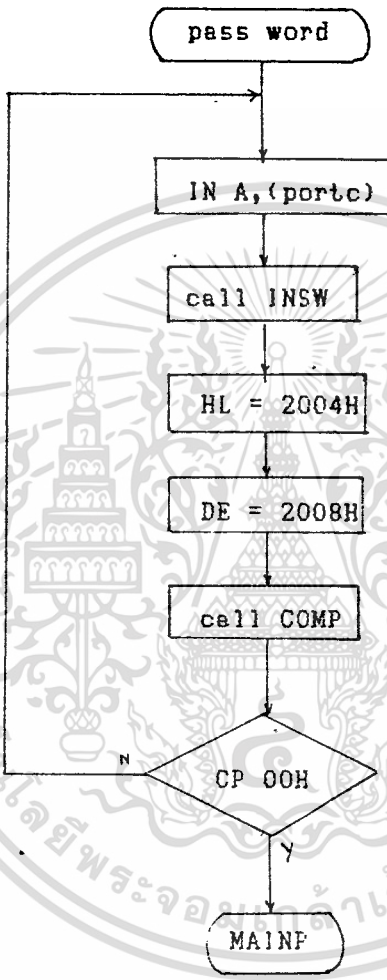
3.2.1. ขั้นตอนการทำงานของโปรแกรมในช่วงเริ่มต้นจะทำการ initial ค่าต่างๆในส่วนของ port 8255 ที่นำมาใช้ โดยกำหนด port A ,B,C เป็น port input ทั้งหมดเพราะฉะนั้นเราต้องกำหนดค่า control word ให้กับ port control ของ PIO_8255 ก่อนด้วยค่า 9BH จากนั้นทำการเคลียร์ relay ที่ output ทุกตัวให้อยู่ในสภาวะ OFF ทั้งหมดพร้อมทั้งเคลียร์สถานะของ Function Key และเคลียร์สถานะของ pass word ให้ LED อยู่ในสภาวะดับทุกตัว

3.2.2. ขั้นตอนการทำงานโปรแกรมตรวจสอบรหัสผ่าน (pass word program)

โปรแกรมตรวจสอบรหัสผ่านนี้จะทำหน้าที่ตรวจเช็คการกด sw pass word ถ้าเมื่อไรมีการกด sw pass word โปรแกรมจะแสดงผล LED pass word บนเครื่อง จากนั้นโปรแกรมจะ call subrutine INSW เพื่อทำการอ่านค่า sw code pass เข้ามาเก็บไว้ใน address 2004H - 2007H เมื่ออ่านค่าได้เรียบร้อยแล้ว โปรแกรมจะกำหนดค่า Registor HL ชี้ที่ address 2004H และ DE ชี้ที่ address 2008H เพื่อเป็นจุดเริ่มที่จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นำค่า code ใน address นั้นมาทำการเปรียบเทียบกันในส่วนโปรแกรมย่อย COMP ซึ่งผลของการ COMP นั้นไม่ถูกต้องคือ รหัส code ที่กดไม่ตรงกับที่เราตั้งไว้ในเครื่องโปรแกรมจะแสดงผลโดยให้ LED pass word ดับ แต่ถ้ากด code ถูกต้องคือ กดรหัสตรงกับเครื่องโปรแกรมก็จะแสดงผลให้ LED pass code ติด เมื่อผ่านรหัสถูกต้องแล้วเครื่องจะเข้าสถานะ NORMAL ซึ่งเป็นสภาวะพร้อมที่จะทำงานโดยผ่านไปยังโปรแกรม MAIN-PROGRAM



รูป.แสดง Flow Chart ของโปรแกรม pass word

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3. ขั้นตอนการทำงานโปรแกรมหลัก (main program)

ส่วนโปรแกรมหลักจะเริ่มต้นโดยทำการ initial สถานะของ Function key และสถานะของการกด switch จากนั้นจะ set สถานะใน status word เป็นสถานะ Normal ในบิทที่ 5 เป็น logic "1" จะทำให้เครื่องควบคุมการเลือกสัญญาณ I/O อยู่ในสถานะปกติโดยจะมี LED ตัวที่ 5 แสดงผลสภาวะ Normal ในสภาวะนี้ program จะทำการตรวจเช็คการกด Switch Function Key ว่าตัวใดถูกกดซึ่งในระบบเครื่องนี้เราได้กำหนด Switch Function Key ดังนี้.

1.) Switch Clear ทำหน้าที่ ตัดช่องสัญญาณ I/O ทุกช่องสัญญาณให้อยู่ในสภาพ open switch

2.) Switch Prog_1 ทำหน้าที่ ต่อช่องสัญญาณ I/O ตามรูปแบบที่เก็บอยู่ในโปรแกรมที่ 1

3.) Switch Prog_2 ทำหน้าที่ ต่อช่องสัญญาณ I/O ตามรูปแบบที่เก็บอยู่ในโปรแกรมที่ 2

4.) Switch Prog_3 ทำหน้าที่ ต่อช่องสัญญาณ I/O ตามรูปแบบที่เก็บอยู่ในโปรแกรมที่ 3

5.) Switch DEL ทำหน้าที่ ตัดช่องสัญญาณ I/O ที่ละช่องสัญญาณตามการกดคู่ switch i/p and switch o/p

6.) Switch Normal ทำหน้าที่ ทำให้เครื่องอยู่ในสภาวะพร้อมที่จะมีการตัดต่อช่องสัญญาณ I/O ตามการ switch i/p กับ switch o/p

7.) Switch Save ทำหน้าที่ ย้ายรูปแบบที่มีการต่อช่องสัญญาณ I/O ในสถานะ Normal ไปเก็บไว้ใน prog_1 or prog_2 or prog_3 โดยทำการกด switch prog_1 or switch prog_2 or switch prog_3 ตามหลัง

เมื่อ switch Function Key ถูกกดโปรแกรมหลักจะ ไป call subrutine ในสถานะ switch ตัวนั้น เมื่อบริการดปรแกรมย่อยเสร็จมันจะกลับมาทำการตรวจเช็ค seitch Function key ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบของ status word ของ Function Key ได้กำหนดอยู่ในรูปของข้อมูล 1 byte ในการเก็บสถานะ ซึ่งจะถูกเก็บไว้ใน address statusซึ่งชี้โดย Registor Index (IY) ที่ Address 2000H.

(IY) =	D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

โดยที่.

D_0 : แสดงสถานะของการ Save

D_1 : แสดงสถานะของ Prog_1

D_2 : แสดงสถานะของ Prog_2

D_3 : แสดงสถานะของ Prog_3

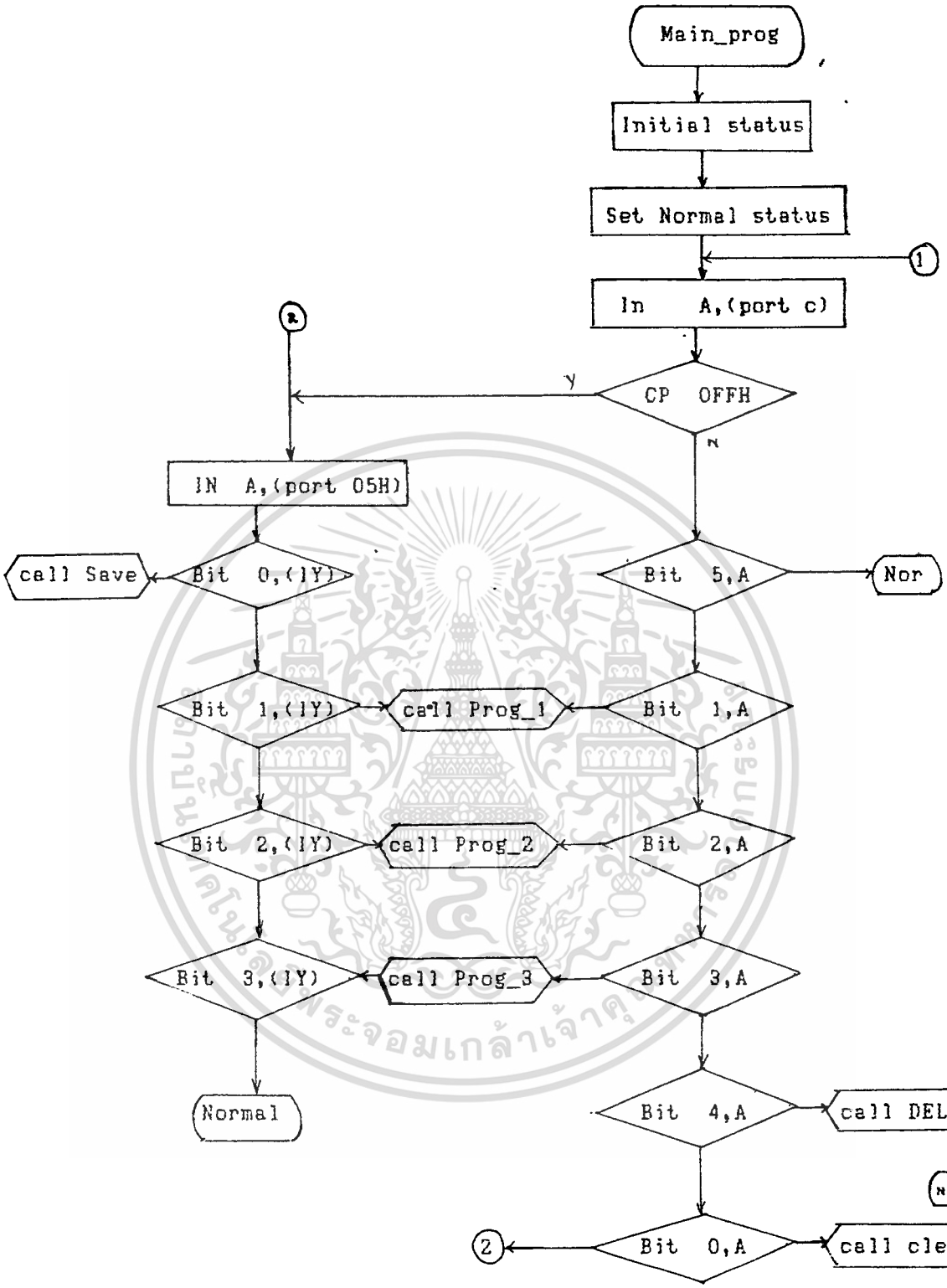
D_4 : แสดงสถานะของ DEL

D_5 : แสดงสถานะของ Normal

D_6, D_7 : ว่าง

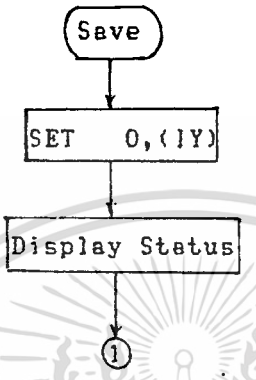
หลังจากการตรวจเช็ค switch แล้วโปรแกรมจะไปตรวจสอบสถานะใน status word ว่าเป็นอะไรจากนั้นโปรแกรมจะเข้าสู่สถานะนั้น ดังจะแสดงในรูปของ Flow Chart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



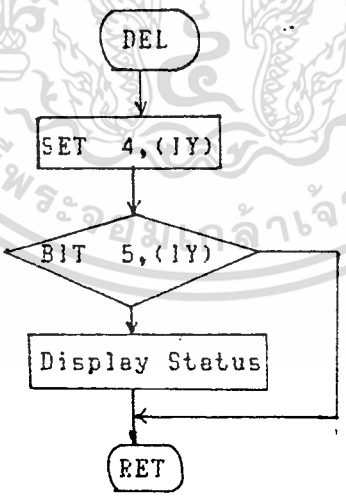
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม Save --- เมื่อเรากด switch Save โปรแกรมหลักจะ call Subrutine Save ซึ่งโปรแกรม Save จะทำการ set สถานะใน status word bit 0 เป็น logic 1 ของสถานะ Save แล้วจากนั้นจะแสดงผลทาง LED save บนเครื่องเมื่อเสร็จจากส่วนนี้แล้ว มันจะกลับไปที่โปรแกรมหลักเพื่อตรวจสอบ switch P1, P2, P3 เพื่อที่จะย้ายรูปแบบการต่อช่องสัญญาณในส่วนของ Normal ไปเก็บไว้ใน Prog_1,2,3 ตามที่ต้องการ



รูป.แสดง flow chartของโปรแกรม save

โปรแกรม DEL --- ทำหน้าที่ใช้ในส่วนที่เราต้องการตัดช่องสัญญาณ I/O ตัวใดตัวหนึ่ง การทำงานของโปรแกรม DEL จะมีการ set สถานะใน status word ให้เป็นสถานะ DELก่อนพร้อมทั้งแสดงผล LED DEL บนเครื่องให้ทราบ แต่การที่จะมีการ DEL ได้นั้น เราได้กำหนดให้ DEL ได้ในสถานะ Normal เท่านั้นถ้าอยู่ในสถานะอื่นเราจะไม่ให้เกิดการDEL

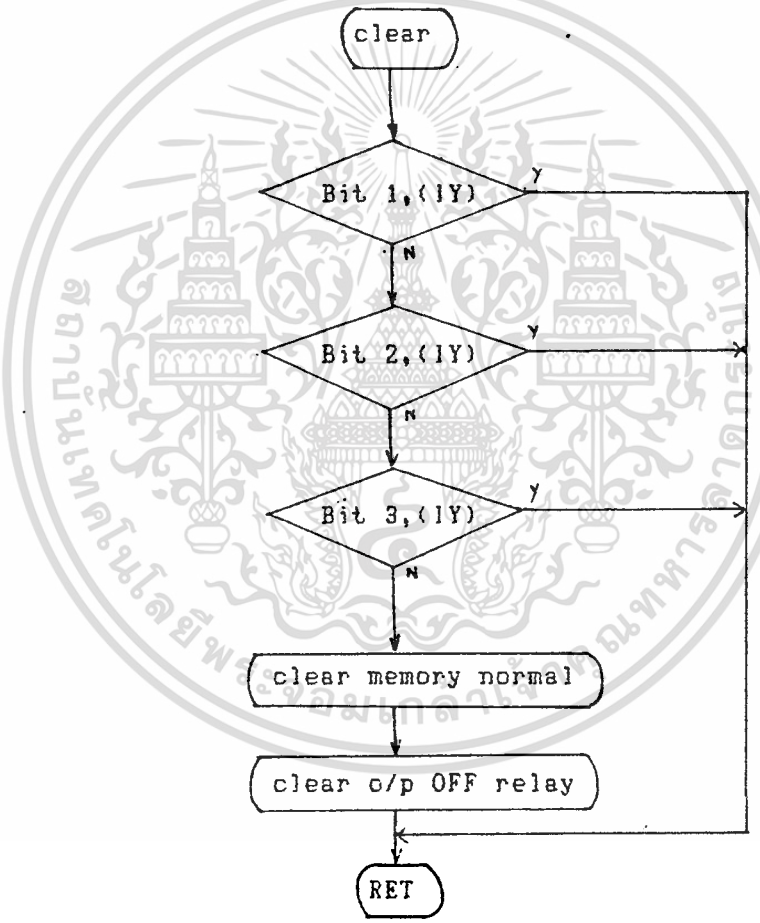


รูป.แสดง flow chart ของโปรแกรม DEL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อมีการกด sw DEL ในสถานะ Normal โปรแกรม DEL จะ set สถานะในช่วงนั้นให้เป็น DEL แล้วจะไปที่ Normal เพื่อเช็ค switch i/p และ switch o/p ว่าคู่ใดกดเพื่อที่จะไปตัดช่องสัญญาณที่ตำแหน่งนั้นพร้อมทั้งลบสถานะ DEL ไปด้วย

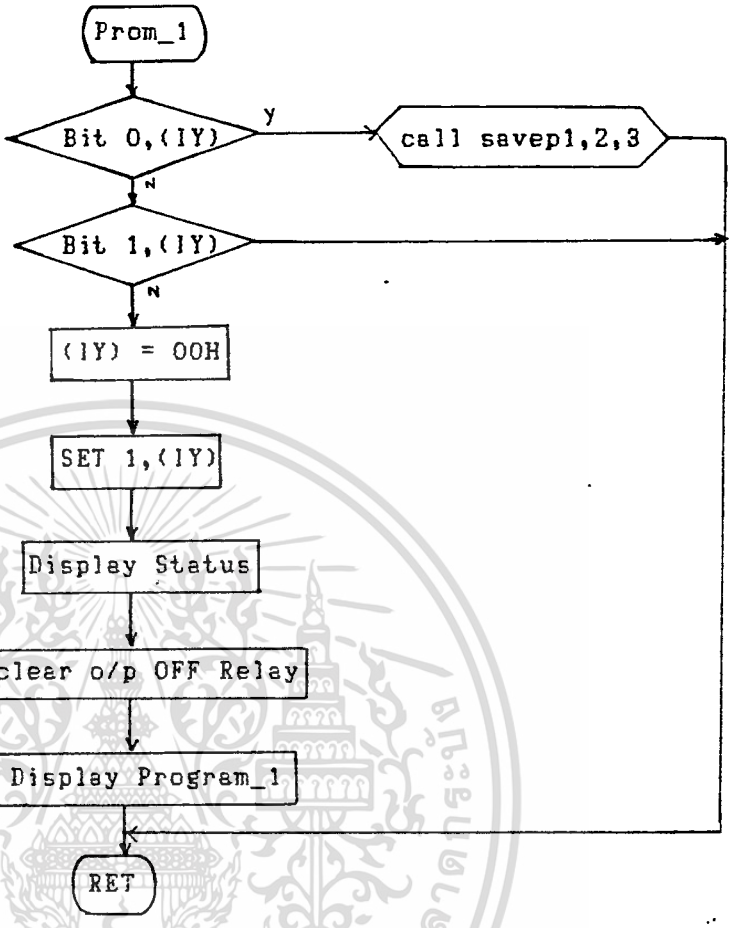
<u>โปรแกรม Clear</u>--- เป็นส่วนที่ทำหน้าที่เคลียร์ข้อมูลใน memory ของ Normal ที่ address 2020H - 2027H ให้เป็น 00H แล้วจากนั้นจะเคลียร์ O/P Relay หรือ Analog switch ให้อยู่ในสภาวะ OFF ทุกตัว แต่การที่จะทำการ clear ตามที่กล่าวมาได้นั้น เครื่องจะต้องอยู่ในสถานะของ Normal เท่านั้น ถ้าเครื่องอยู่ในสถานะการแสดงผลของ Program_1 , Program_2 or Program_3 เครื่องจะไม่ทำการ clear



รูป.แสดง Flow chart ของโปรแกรม clear

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม Prom_1---ทำหน้าที่ แสดงการต่อช่องสัญญาณ I/O ตามรูปแบบที่ได้ save เก็บไว้ใน Program_1



รูป.แสดง Flow chart ของโปรแกรม Prom_1

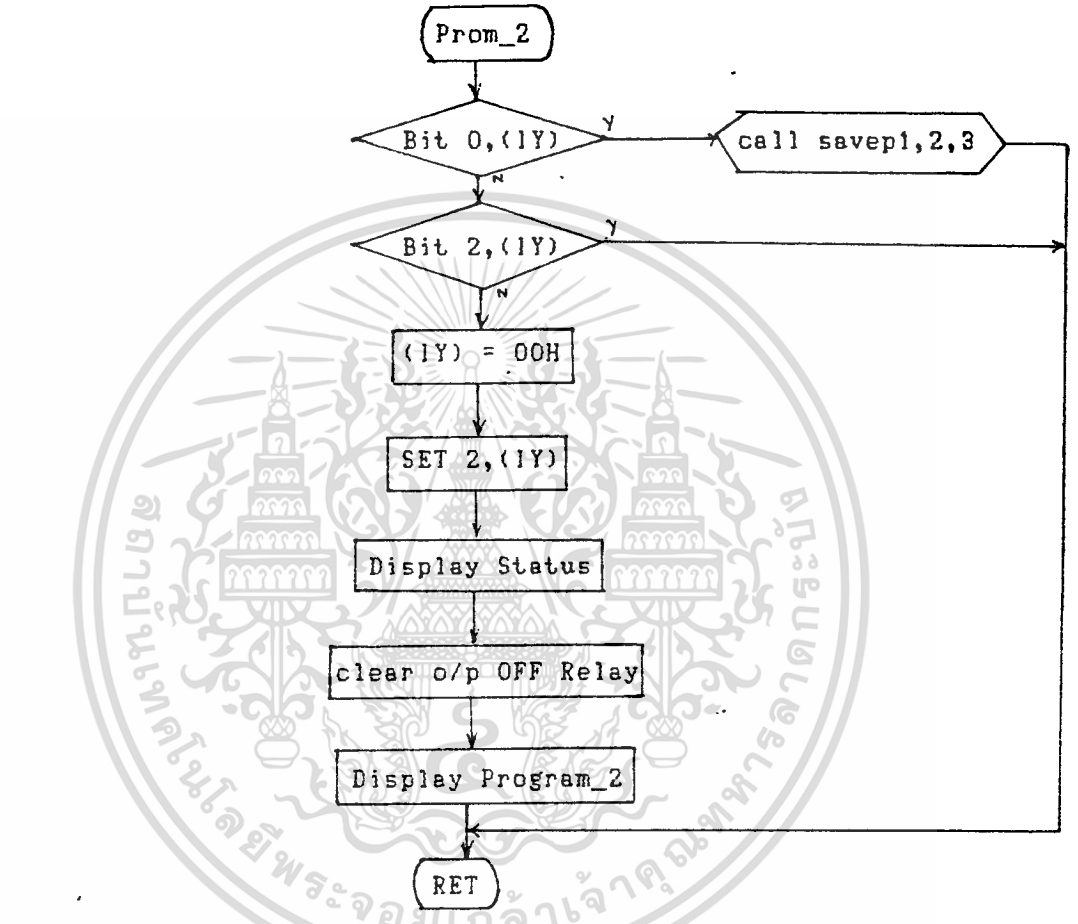
ขั้นตอนการทำงานของโปรแกรม Prom_1

ก่อนที่จะทำการต่อช่องสัญญาณ I/O ตามผลที่เก็บไว้ใน Prom_1 นั้นโปรแกรมจะมีการตรวจเช็คสถานะใน status word ดูก่อนว่าได้มีการกด save มาก่อนหรือไม่ ถ้ามีโปรแกรมก็จะทำการ call subrutine savep1,2,3 เพื่อทำการ save รูปแบบการต่อช่องสัญญาณที่ได้ Set จากสถานะของ Normal มาเก็บไว้ใน Address ของ Prom_1 แต่ถ้าไม่มีการกด save มาก่อนโปรแกรมก็จะไปเช็คดูว่าเป็นสถานะ Prom_1 อยู่ก่อนหรือไม่ ถ้าเป็นสถานะ Prom_1 อยู่แล้วโปรแกรมก็จะกลับเข้าสู่โปรแกรมหลัก ถ้าสถานะเดิมไม่ใช่สถานะ Prom_1 โปรแกรมส่วนนี้จะทำการ clear status word แล้วจะทำการ Set สถานะใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

status word ให้เป็นสถานะ Prom_1 จากนั้นจะแสดงผลทาง LED ของ Prom_1 บนเครื่อง เมื่อแสดงผลสถานะ Prom_1 แล้วโปรแกรมจะทำการเคลียร์ O/P relay หรือ Analog switch พร้อมทั้งทำการ ON Relay หรือ Analog switch ตามรูปแบบการต่อช่องสัญญาณ I/O ของ Prom_1

โปรแกรม Prom_2---ทำหน้าที่ แสดงการต่อช่องสัญญาณ I/O ตามรูปแบบที่ได้ save เก็บไว้ใน Program_2



รูป.แสดง Flow chart ของโปรแกรม Prom_2

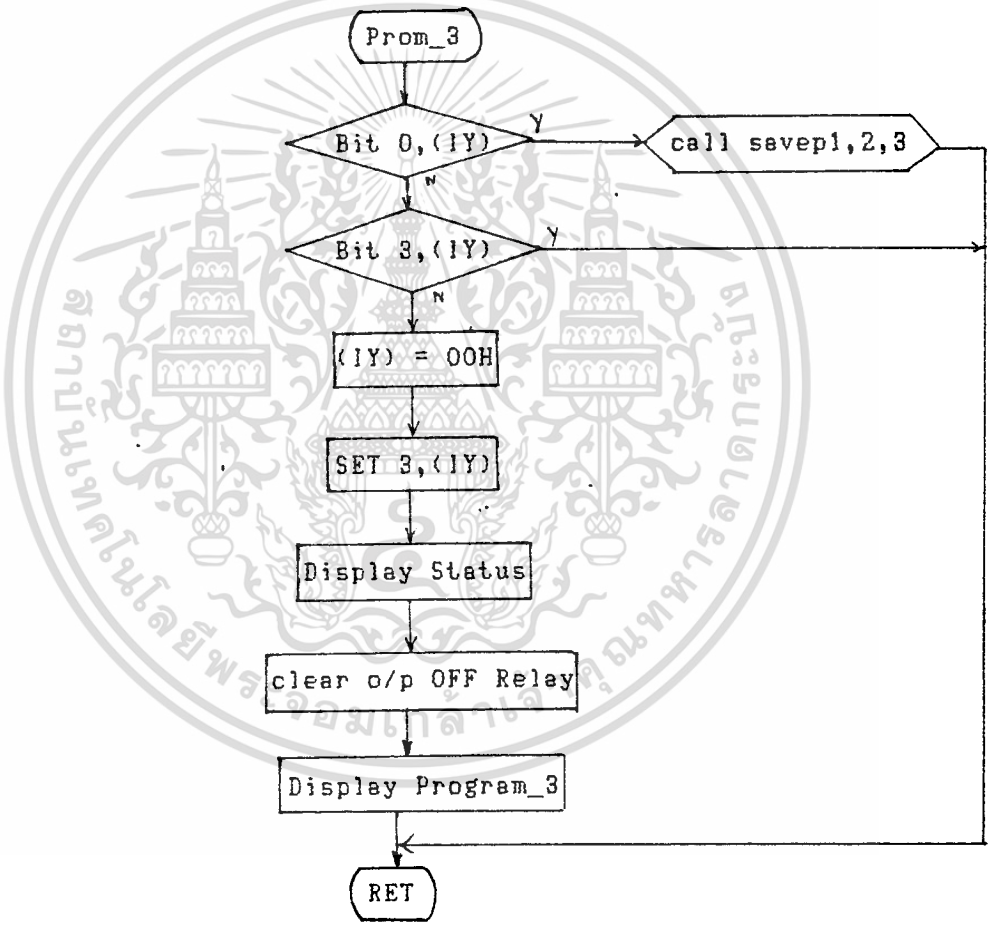
ขั้นตอนการทำงานของโปรแกรม Prom_2

ก่อนที่จะทำการต่อช่องสัญญาณ I/O ตามผลที่เก็บไว้ใน Prom_2 นั้นโปรแกรมจะมีการตรวจเช็คสถานะใน status word ดูก่อนว่าได้มีการกด save มาก่อนหรือไม่ ถ้ามีโปรแกรมก็จะทำการ call subrutine savep1,2,3 เพื่อทำการ save รูปแบบการต่อช่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่ได้ Set จากสถานะของ Normal มาเก็บไว้ใน Address ของ Prom_2 แต่ถ้าไม่มีการกด save มาก่อนโปรแกรมก็จะไปเช็คดูว่าเป็นสถานะ Prom_2 อยู่ก่อนหรือไม่ ถ้าเป็นสถานะ Prom_2 อยู่แล้วโปรแกรมก็จะกลับเข้าสู่โปรแกรมหลัก ถ้าสถานะเดิมไม่ใช่สถานะ Prom_2 โปรแกรมส่วนนี้จะทำการ clear status word แล้วจะทำการ Set สถานะใน status word ให้เป็นสถานะ Prom_2 จากนั้นจะแสดงผลทาง LED ของ Prom_2 บนเครื่อง เมื่อแสดงผลสถานะ Prom_2 แล้วโปรแกรมจะทำการเคลียร์ O/P relay หรือ Analog switch พร้อมทั้งทำการ ON Relay หรือ Analog switch ตามรูปแบบการต่อช่องสัญญาณ I/O ของ Prom_2

โปรแกรม Prom_3---ทำหน้าที่ แสดงการต่อช่องสัญญาณ I/O ตามรูปแบบที่ได้ save เก็บไว้ใน Program_3



รูป.แสดง Flow chart ของโปรแกรม Prom_3

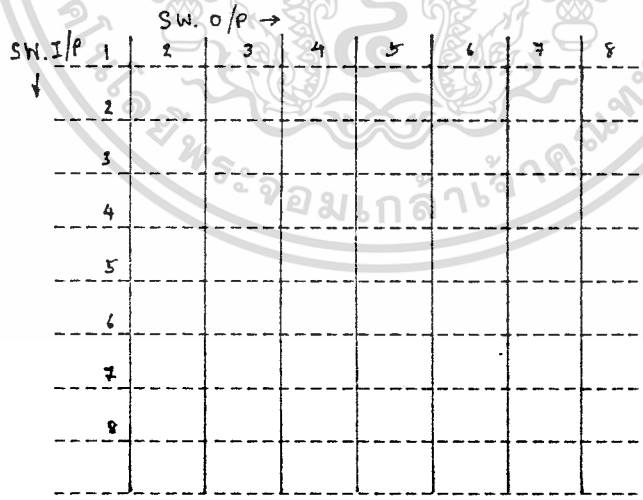
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการทำงานของโปรแกรม Prom_3

ก่อนที่จะทำการต่อช่องสัญญาณ I/O ตามผลที่เก็บไว้ใน Prom_3 นั้นโปรแกรมจะมีการตรวจเช็คสถานะใน status word ดูก่อนว่าได้มีการกด save มาก่อนหรือไม่ ถ้ามีโปรแกรมก็จะทำการ call subrutine savep1,2,3 เพื่อทำการ save รูปแบบการต่อช่องสัญญาณที่ได้ Set จากสถานะของ Normal มาเก็บไว้ใน Address ของ Prom_3 แต่ถ้าไม่มีการกด save มาก่อนโปรแกรมก็จะไปเช็คดูว่าเป็นสถานะ Prom_3 อยู่ก่อนหรือไม่ ถ้าเป็นสถานะ Prom_3 อยู่แล้วโปรแกรมก็จะกลับเข้าสู่โปรแกรมหลัก ถ้าสถานะเดิมไม่ใช่สถานะ Prom_3 โปรแกรมส่วนนี้จะทำการ clear status word แล้วจะทำการ Set สถานะใน status word ให้เป็นสถานะ Prom_3 จากนั้นจะแสดงผลทาง LED ของ Prom_3 บนเครื่อง เมื่อแสดงผลสถานะ Prom_3 แล้วโปรแกรมจะทำการเคลียร์ O/P relay หรือ Analog switch พร้อมทั้งทำการ ON Relay หรือ Analog switch ตามรูปแบบการต่อช่องสัญญาณ I/O ของ Prom_3

<u>โปรแกรม Save</u> ---ทำหน้าที่ Save รูปแบบการต่อช่องสัญญาณ I/O ในสถานะการทำงานของ Normal โดยการนำเอาข้อมูลใน Adress Normal ที่ 2020H--2027H ไปเก็บไว้ใน Adress P1 หรือ P2 หรือ P3 ตามแต่ว่าเราจะ save ใน Prom ใด.

<u>โปรแกรม Normal</u> --- คือส่วนของการตรวจเช็คการเลือกช่องติดต่อช่องสัญญาณ I/O โดยอาศัยหลักการตรวจสอบ switch I/P ก่อนแล้วทำการตรวจสอบ switch O/P ตาม ในระบบเครื่องนี้เราใช้ SW I/P 1 ถึง 8 และ SW O/P 1 ถึง 8 โดยจะกำหนดในรูปของ SW Metrix (8 x 8) แสดงดังรูปข้างล่าง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่.. SW I/P 1-8 เรากำหนดแทน ช่องสัญญาณ I/P 1-8

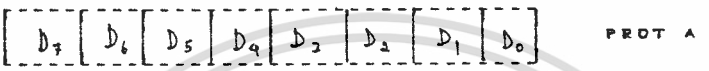
SW O/P 1-8 เรากำหนดแทน ช่องสัญญาณ O/P 1-8

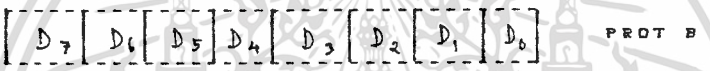
<u>หลักการทำงานของระบบ sw matrix (8 x 8)</u>

SW I/P 1-8 ถูกต่อกับ I/P PORT A ของ PIO_8255

SW O/P 1-8 ถูกต่อกับ I/P PORT B ของ PIO_8255

ซึ่งเราจะใช้ ข้อมูล 1 byte แทน SW I/P end SW O/P



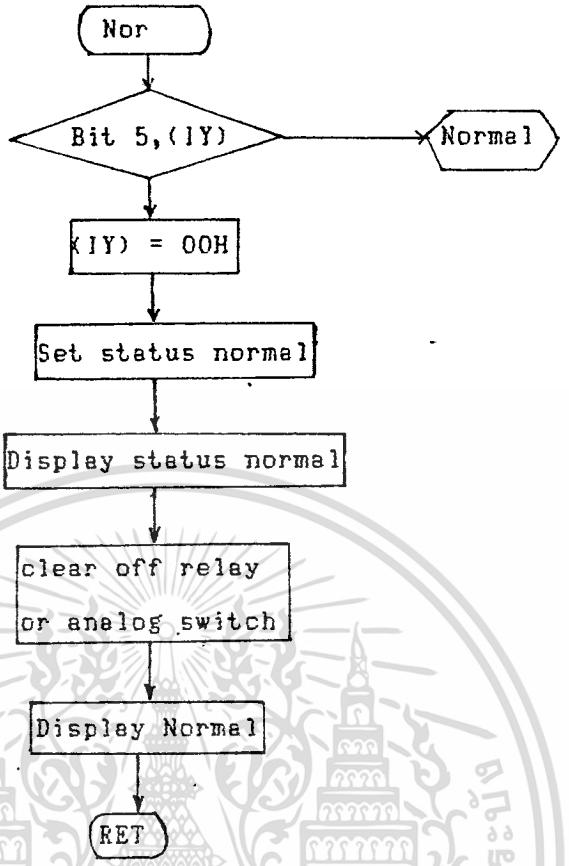


เมื่อ SW I/P หรือ SW O/P ตัวใดถูกกดจะทำให้ Bit นั้นเป็น logic "0" ส่วนตัวใดไม่ถูกกดจะมี logic "1" เพราะ SW จะต่อในลักษณะมี $+v_{cc}$ ต่ออยู่ตลอดเวลา

<u>หลักการทำงานของโปรแกรม Normal</u>

ก่อนจะเข้าสถานะ Normal นั้นโปรแกรมจะทำการเช็คสถานะจาก status word ดูก่อนว่า ตอนนี้เป็นสถานะ Normal อยู่หรือไม่ ถ้าเป็นสถานะ Noraml อยู่ก็จะไปในส่วนของ Normal แต่ถ้าไม่ใช่มันจะทำการเคลียร์ status word แล้วจึง SET status wrod ให้เป็นสถานะ Normal พร้อมทั้งแสดงผล LED Normal บนเครื่องให้ทราบจากนั้นจะทำการต่อช่องสัญญาณ I/O ของ Normal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เมื่อเข้าสถานะ Normal ในส่วนนี้จะเป็นส่วนทำการตรวจสอบ switch โดยจะเริ่มตรวจสอบ SW I/P 1-8 ถ้าไม่มีการกด SW I/P โปรแกรมจะทำการตรวจเช็คสถานะใน address 2001H ดูว่ามีสถานะการกด SW I/P มาก่อนหรือไม่ ถ้ามีโปรแกรมจะไปเริ่มเช็คที่ SW O/P1-8

IY+1 ชี้ที่ address 2001H

(IY+1) =

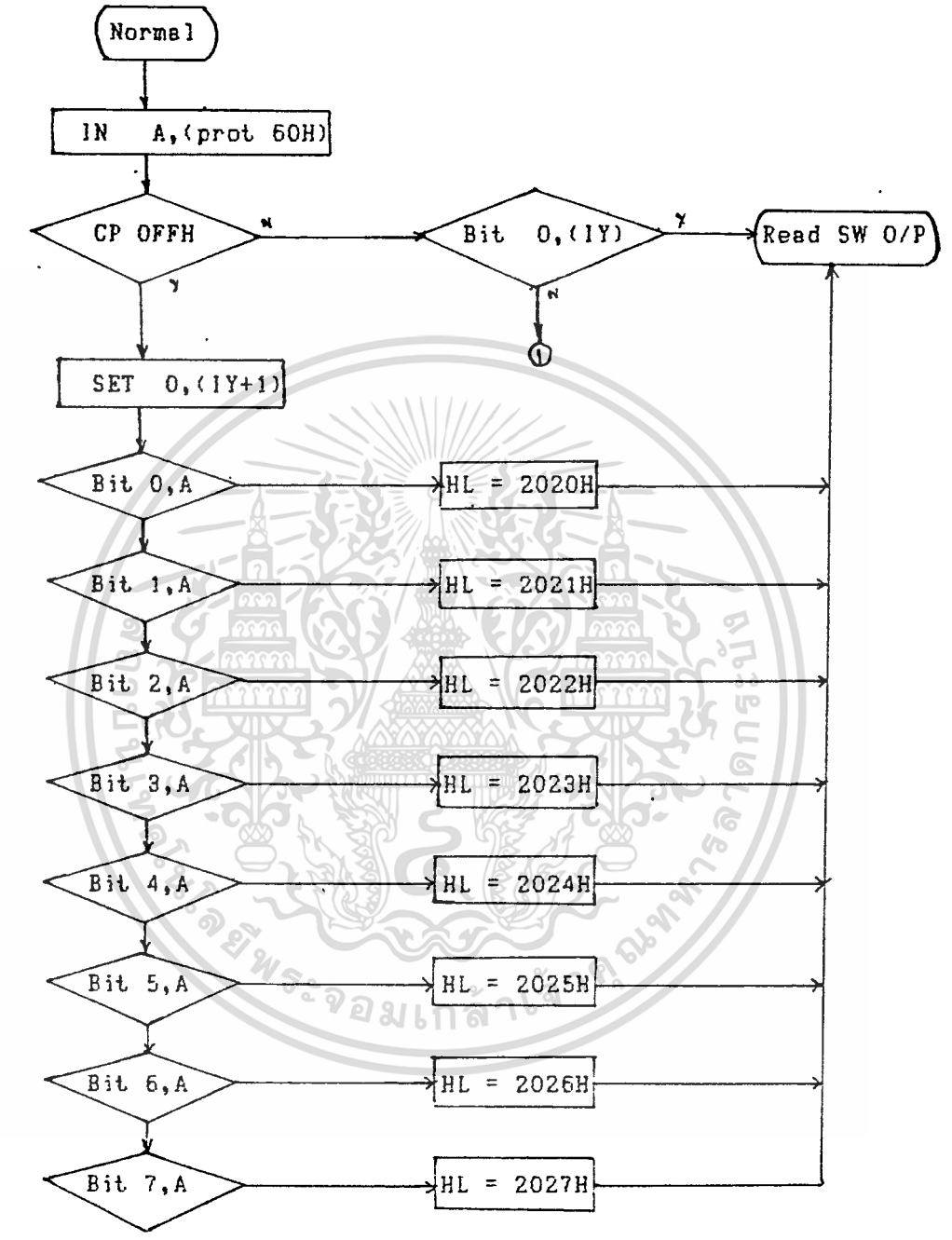
"1" : แสดงสถานะการกด SW I/P

"0" : ไม่มีสถานะการกด SW I/P

แต่ในสถานะเป็น "0" โปรแกรมจะไปเริ่มตรวจเช็ค Function Key ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป.แสดง Flow chart ของโปรแกรม Normal



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีมีการกด SW I/P โปรแกรมจะทำการ SET ข้อมูลใน Memory Address 2001H ให้เป็น 01H ทุกครั้ง จากนั้นจะตรวจเช็คดูว่า SW I/P ตัวใดกด

เมื่อ SW I/P ตัวที่ 1 กด จะกำหนดค่า HL = 2020H

"-------" 2 "----------"HL = 2021H

"-------" 3 "----------"HL = 2022H

"-------" 4 "----------"HL = 2023H

"-------" 5 "----------"HL = 2024H

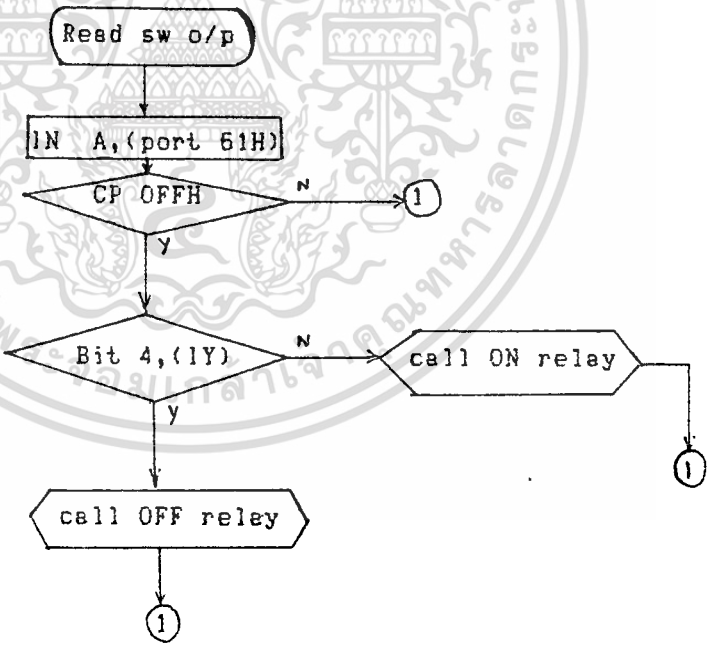
"-------" 6 "----------"HL = 2025H

"-------" 7 "----------"HL = 2026H

"-------" 8 "----------"HL = 2027H

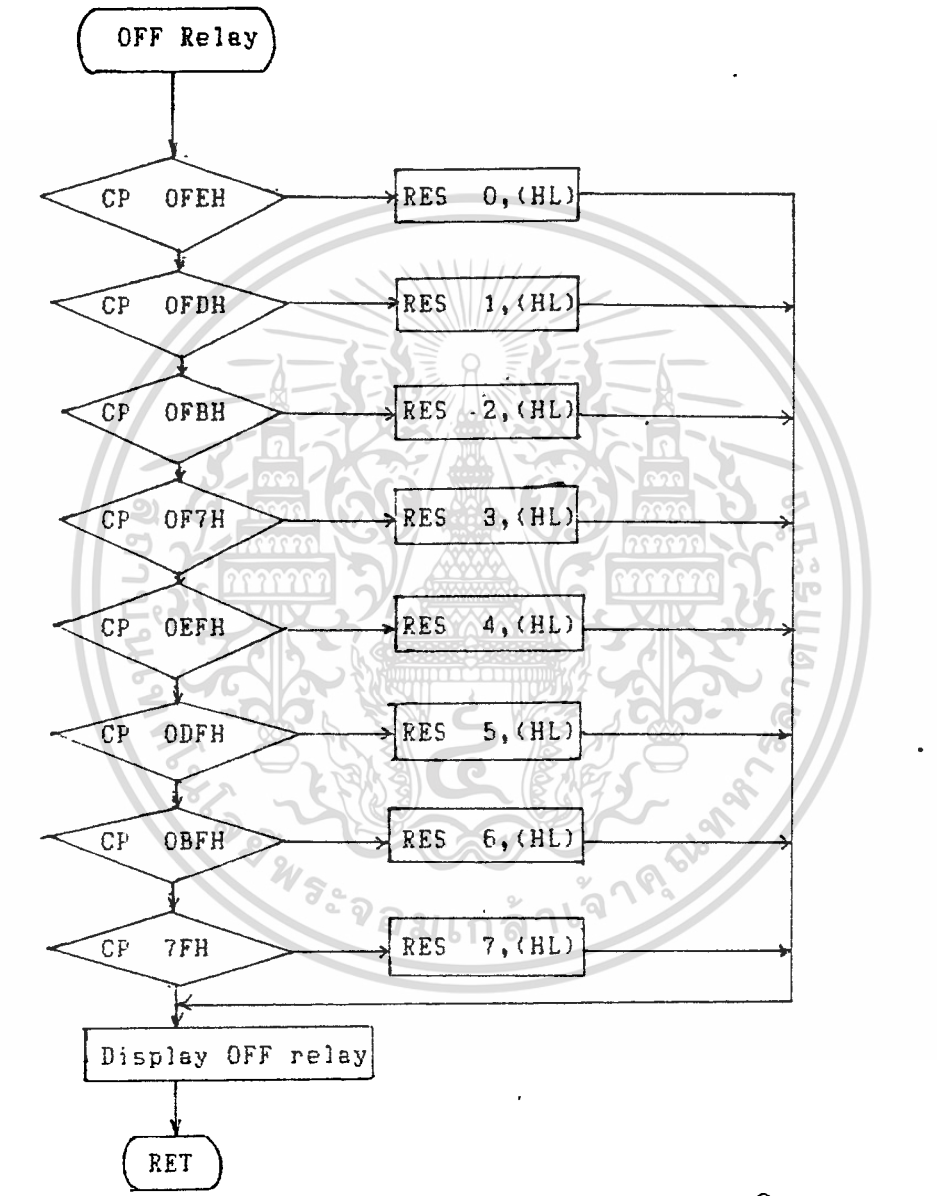
ค่า HL ที่ชี้ใน Adress นี้ จะถูกส่งไปยังโปรแกรม ON/OFF RELAY

เมื่อตรวจ SW I/P แล้วโปรแกรมจะมาทำการตรวจเช็ค SW O/P 1-8 อีกครั้งหนึ่งเพื่อเช็คดูว่า SW O/P ตัวใดกดเมื่อมีการกด SW O/P โปรแกรมจะไปตรวจสอบดูว่าเป็นการ ON RELAY หรือ OFF RELAY จากสถานะ DEL ถ้าสถานะใน DEL เป็น "0" แสดงว่า ON (ต่อช่องสัญญาณ) แต่ถ้าเป็น "1" แสดงว่า OFF (ตัดช่องสัญญาณ) ดัง Flow chart ข้างล่าง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณี OFF .. โปรแกรมจะ call subrutine OFF relay ซึ่งส่วนนี้จะมีหน้าที่ในการตัดช่องสัญญาณ I/O ตามคู่ SW I/P กับ SW O/P โดยในโปรแกรมจะนำผลการกด sw I/P กับ sw O/P มาทำการเปลี่ยนค่าใน Memory ในตำแหน่งที่กำหนดโดยการ set ค่าเป็น logic "0" แล้วจะนำค่าใน Memory ออก port ที่ 08H - OFH เพื่อทำให้ relay or analog switch ตัดช่องสัญญาณให้



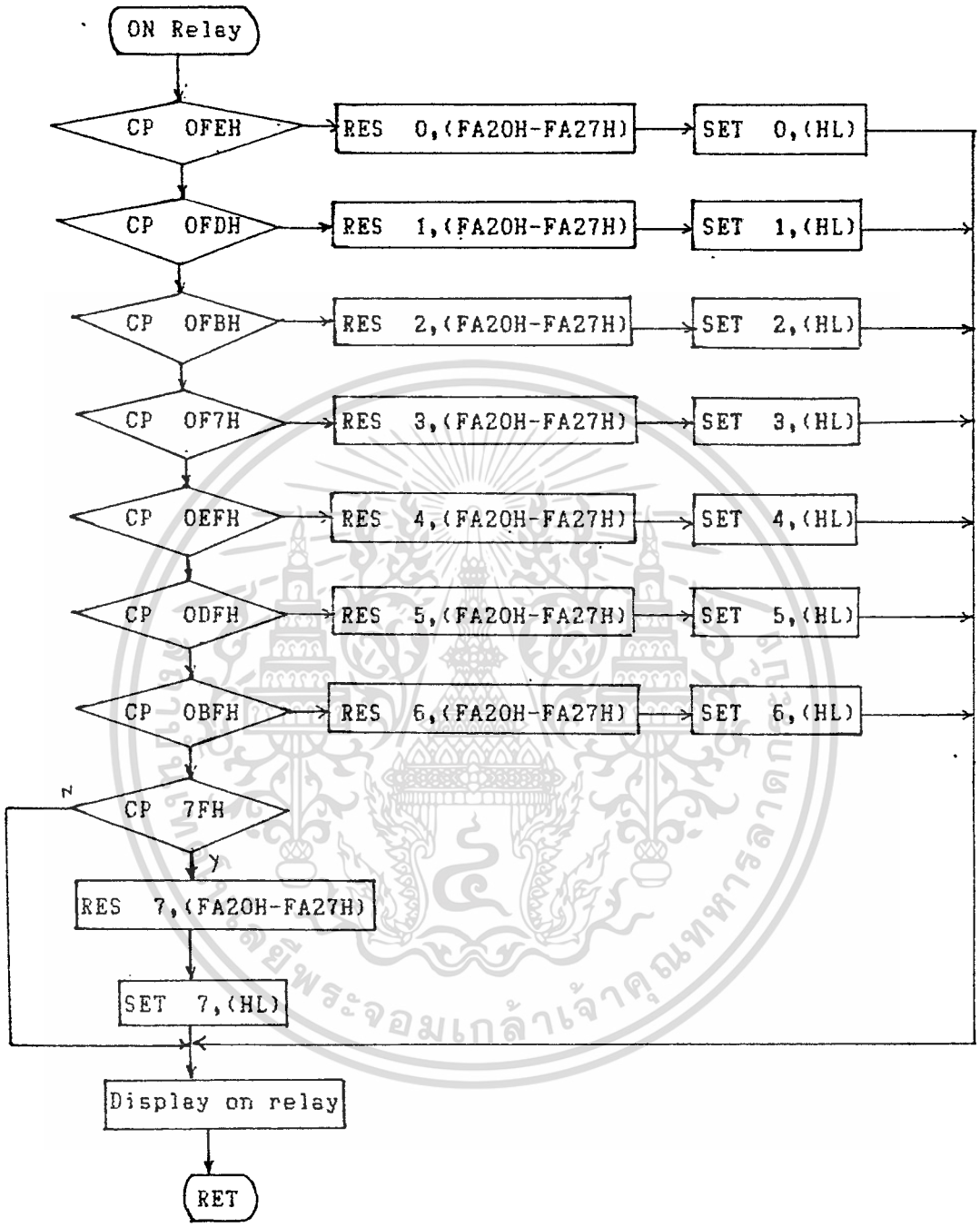
รูป.แสดง Flow Chart ของโปรแกรม OFF RELAY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

<u>ในกรณี ON</u> -- โปรแกรมจะ cell subrutine ON relay ซึ่งส่วนนี้จะมีหน้าที่ในการต่อช่องสัญญาณ I/O ตามคู่ SW I/P กับ SW O/P โดยในโปรแกรมจะนำผลการกด sw I/P กับ sw O/P มาทำการเปลี่ยนค่าใน Memory ในตำแหน่งที่กำหนดโดยการ set ค่าเป็น logic "1" แล้วจะนำค่าใน Memory ออก port ที่ 08H - OFH เพื่อทำให้ relay or analog switch ต่อช่องสัญญาณให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูป.แสดง Flow chart ของโปรแกรม ON RELAY



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การใช้เครื่องควบคุมการเลือกสัญญาณ I/O

มีขั้นตอนการใช้ดังนี้..

1. กด sw pass word ก่อน ซึ่งเมื่อกด switch แล้ว LED pass word จะติด
2. ทำการกด sw code pass word ซึ่งมีอยู่ด้วยกัน 4 ตัว เราจะต้องทำการกด switch ตาม code ที่ได้กำหนดมา เช่น เครื่องตั้งที่ sw1, sw3, sw2, sw4 เพราะฉะนั้นเวลากดก็ต้องกดตาม code นั้นด้วยถ้ากดไม่ตรง LED pass word จะดับ เมื่อไม่ผ่านเราจะต้องกลับไปทำตามขั้นตอนที่ 1 ใหม่แต่ถ้ากด switch ตรง เครื่องจะแสดงผล LED pass word ติด พร้อมทั้งจะแสดงผล LED normal ให้รู้ว่าเครื่องพร้อมที่จะทำงาน
3. เมื่ออยู่ในสถานะ normal เราต้องการที่จะทำการเลือกช่องสัญญาณ I/P ช่อง1 ถึง ช่อง 8 ต่อกับช่องสัญญาณ O/P ช่อง1 ถึงช่อง8 ได้โดยการกด sw I/P ตัวหนึ่งตัวใดแล้วจึงกด sw o/p ตาม เช่นเรากด sw i/p1 กับ sw o/p1 จะทำให้relay or analogswitchตัวที่1 ต่อช่องสัญญาณให้ ในเครื่องนี้เราจะกำหนดให้ช่องสัญญาณ o/p สามารถต่อช่องสัญญาณ i/p ได้ 1 ช่อง แต่ช่องสัญญาณ i/p สามารถต่อช่องสัญญาณ o/p ได้หลายช่อง
4. ในกรณีเราต้องตัดช่องสัญญาณ I/O ช่องใดนั้นกระทำได้โดยการกด sw DELแล้วตามด้วยกด sw i/p กับ sw o/p คู่นั้น
5. ถ้าต้องการตัดช่องสัญญาณ I/O ทั้งหมด เราจะใช้กด sw clear
6. เมื่อต้องการเก็บรูปแบบการต่อช่องสัญญาณ I/O เราจะเริ่มทำการกด sw save ก่อนแล้วตามด้วยกด sw prog-1 หรือ sw prog-2 หรือ sw prog-3 ตัวใดตัวหนึ่ง ตามที่เราต้องการ
7. เมื่ออยู่ในสถานะ normal เราสามารถที่จะนำรูปแบบการต่อช่องสัญญาณที่เก็บอยู่ในโปรแกรมมาแสดงผลการต่อได้โดยการกด sw prog-1 or 2 or 3 ตามที่เราต้องการ
8. ในกรณีที่เครื่องแสดงผลการต่อช่องสัญญาณตาม prog-1,2,3 อยู่นั้นเราจะทำการกด sw clear หรือ sw save หรือ sw DEL จะไม่มีผลต่อเครื่อง แต่ถ้าเราจะไปเริ่ม set การต่อช่องสัญญาณใหม่ได้โดยการกด sw normal ซึ่งใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สถานะนี้ถ้าเรา set การต่อช่องสัญญาณ I/O ใหม่ แล้วทำการ save ใหม่ ได้โดยกด sw save แล้วตามด้วย prog-1 จะทำให้รูปแบบการต่อช่องสัญญาณใน porg-1 เปลี่ยนตามไปด้วย

9. เมื่อทำการปิดเครื่องและจะมีการเริ่มเล่นเครื่องใหม่นั้นจะต้องเริ่มต้นที่ขั้นตอนที่ 1 ใหม่ก่อนทุกครั้ง ส่วนในขั้นตอนที่ 3 ไปจนถึงขั้นตอนที่ 8 ไม่จำเป็นต้องเรียงลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MONITOR-PROGRAM
OF
PROGRAMABLE I/O SELECTOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MACRO-80 3.36   17-Mar-80      PAGE   1


                                .Z80
0000                            ASEG
                                ORG     0000H
2600                    STACK   EQU     2600H
0063                    PORTC   EQU     63H
0062                    PORTSF  EQU     62H
0060                    PORTSI  EQU     60H
0061                    PORTSO  EQU     61H
0005                    PORTSS  EQU     05H
0007                    PORTDS  EQU     07H
0004                    PORTS4  EQU     04H
0006                    PORTSP  EQU     06H
                        ;
                        ;---------------------------------------
                        ;       MONITOR-PROGRAM
                        ;---------------------------------------
0000    31 2600                 LD      SP,STACK        ;STACK POINT AT ADDRESS 2600H
0003    3E 9B                   LD      A,9BH           ;CONTROL WORD  8255
0005    D3 63                   OUT     (PORTC),A       ;OUT CONTROL PORTC
0007    CD 0363                 CALL    CLEARO
000A    CD 035B                 CALL    CLEARM
000D    AF                      XOR     A
000E    D3 06                   OUT     (PORTSP),A      ;CLEAR STATUS FUNTION
0010    D3 07                   OUT     (PORTDS),A      ;CLEAR STATUS PASS_WORD
0012    FD 21 2000              LD      IY,2000H        ;ADDRESS STATUS
0016    DD 21 2008              LD      IX,2008H
001A    DD 36 00 FE             LD      (IX),0FEH
001E    DD 36 01 FB             LD      (IX+1),0FBH     ;CORD PASS WORD
0022    DD 36 02 FD             LD      (IX+2),0FDH
0026    DD 36 03 F7             LD      (IX+3),0F7H
002A    DB 62           PASS:   IN      A,(PORTSF)      ;READ PASS_WORD
002C    CB 77                   BIT     6,A             ;CHECK SW PASS_WORD
002E    20 FA                   JR      NZ,PASS
0030    3E 01                   LD      A,01H
0032    D3 06                   OUT     (PORTSP),A
0034    CD 0061                 CALL    INSW
0037    21 2004                 LD      HL,2004H        ;ADDRESS IN SW 1-4
003A    11 2008                 LD      DE,2008H        ;ADDRESS IN MEMORY
003D    CD 004A                 CALL    COMP
0040    FE 00                   CP      00h
0042    28 32                   JR      Z,NO-INP
0044    AF                      XOR     A
0045    D3 06                   OUT     (PORTSP),A      ;NO PASS WORD
0047    C3 002A                 JP      PASS
                        ;---------------------------------------
004A    06 04           COMP:   LD      B,04H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
004C  1A          COM:  LD    A,(DE)
004D  96                SUB   (HL)
004E  20 04             JR    NZ,LN
0050  23                INC   HL
0051  13                INC   DE
0052  10 F8             DJNZ  COM
0054  C9          LN:   RET
                  ;-------------------------------------
0055  11 09FF     DELAY: LD    DE,09FFH
0058  1B          DLY:  DEC   DE
0059  7B                LD    A,E
005A  B2                OR    D
005B  20 FB             JR    NZ,DLY
005D  0D                DEC   C
005E  20 F5             JR    NZ,DELAY
0060  C9                RET
                  ;-------------------------------------
0061  06 04       INSW: LD    B,04H
0063  21 2004           LD    HL,2004H
0066  DB 04       SW:   IN    A,(PORTS4)
0068  FE FF             CP    0FFH
006A  28 FA             JR    Z,SW
006C  77                LD    (HL),A
006D  23                INC   HL
006E  0E 1F             LD    C,1FH
0070  CD 0055           CALL  DELAY
0073  10 F1             DJNZ  SW
0075  C9                RET
                  ;-------------------------------------
                  ;---   MAIN-PROGRAM                ---
                  ;-------------------------------------
0076  3E 02       MAIN: LD    A,02H
0078  D3 08             OUT   (PORTSP),A    ;YES PASS WORD
007A  CD 0296           CALL  ONOFFR        ;ON ANALOG SW O/P
007D  AF                XOR   A
007E  FD 77 00          LD    (IY),A        ;CLEAR STATUS FUNTION KEY
0081  FD 77 01          LD    (IY+1),A      ;CLEAR STATUS SW I/P
0084  FD CB 00 EE       SET   5,(IY)        ;SET STATUS NORMAL
0088  CD 0380           CALL  DIS_S         ;DISPLAY STATUS
008B  DB A2       READF: IN    A,(PORTSP)    ;READ SW FUNTION KEY
008D  FE FF             CP    0FFH          ;CHECK SW FUNTION KEY
008F  28 1B             JR    Z,READS
0091  CB 6F             BIT   5,A           ;CHECK SW NORMAL
0093  28 61             JR    Z,NOR
0095  CB 4F             BIT   1,A           ;CHECK SW PROM_1
0097  28 29             JR    Z,PROM_1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0099  CB 57                    BIT     2,A             ;CHECK SW PROM_2
009B  28 2D                    JR      Z,PROM_2
009D  CB 5F                    BIT     3,A             ;CHECK SW PROM_3
009F  28 2F                    JR      Z,PROM_3
00A1  CB 67                    BIT     4,A             ;CHECK SW DEL
00A3  28 31                    JR      Z,DEL
00A5  CB 47                    BIT     0,A             ;CHECK SW CLEAR
00A7  28 3D                    JR      Z,CLEAR
00A9  DB 05             READS: IN      A,(PORTSS)      ;READ SW SAVE
00AB  CB 47                    BIT     0,A             ;CHECK SW SAVE
00AD  28 3D                    JR      Z,SAVE
00AF  FD CB 00 4E              BIT     1,(IY)          ;CHECK STATUS PROM_1
00B3  20 0F                    JR      NZ,PROM_1
00B5  FD CB 00 56              BIT     2,(IY)          ;CHECK STATUS PROM_2
00B9  20 0F                    JR      NZ,PROM_2
00BB  FD CB 00 5E              BIT     3,(IY)          ;CHECK STATUS PROM_3
00BF  20 0F                    JR      NZ,PROM_3
00C1  C3 0110                  JP      NORMAL
00C4  CD 02D2           PROM_1: CALL   INTP1           ;DISPLAY PROM_1
00C7  C3 00BB                  JP      RDKEYF
00CA  CD 02FC           PROM_2: CALL   INTP2           ;DISPLAY PROM_2
00CD  C3 00BB                  JP      RDKEYF
00D0  CD 0326           PROM_3: CALL   INTP3           ;DISPLAY PROM_3
00D3  C3 00BB                  JP      RDKEYF
00D6  FD CB 00 E6       DEL:   SET     4,(IY)          ;SET STATUS DEL
00DA  FD CB 00 6E              BIT     5,(IY)
00DE  28 03                    JR      Z,OUTDEL
00E0  CD  38                   CALL    DIS_S           ;DISPLAY STATUS
00E3  C3 0110           OUTDEL: JP     NORMAL
00E6  CD 02AF           CLEAR: CALL    INTC            ;CLEAR MEMORY AND CLEAR RL_O/P
  E9  C3 0110                  JP      NORMAL
00EC  FD CB 00 C6       SAVE:  SET     0,(IY)          ;SET STATUS SAVE
00F0  CD 0360                  CALL    DIS_S           ;DISPLAY STATUS
00F3  C3 00BB                  JP      RDKEYF
                        ;--------------------------------
00F6  FD CB 00 6E       KEY    BIT     5,(IY)          ;CHECK STATUS NORMAL
00FA  20 14                    JR      NZ,NORMAL
00FC  AF                       XOR     A
00FD  FD                       LD      (IY),A          ;CLEAR STATUS
0101  FD CB 00 EE              SET     5,(IY)          ;SET STATUS NORMAL
0105  CD  39                   CALL    DIS_S           ;DISPLAY STATUS
0108  CD 0313                  CALL    CLEARD          ;CLEAR ANALOG SW O/P
010B  21 2020                  LD      HL,2020H
010E  CD 0340                  CALL    DIS_RL          ;ON ANALOG SW OF NORMAL
                        ;--------------------------------
0111  DB 05             NORMAL: IN     A,(PORTSI)      ;READ SW I/P
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0112  FE 00                    CP      00H             ;CHECK SW I/P
0114  20 0A                    JR      NZ,SETSSW
0116  FD CB 01 46              BIT     0,(IY+1)        ;CHECK STATUS SW I/P
011A  C2 0170                  JP      NZ,RSW_DP
011D  C3 00BB                  JP      RDKEYF
                        ;--------------------------------------
0120  FD CB 01 C6       SETSSW: SET     0,(IY+1)        ;SET STATUS SW I/P
0124  CB 47                    BIT     0,A             ;CHECK SW I/P_1
0126  20 1B                    JR      NZ,SW_1
0128  CB 4F                    BIT     1,A             ;CHECK SW I/P_2
012A  20 1D                    JR      NZ,SW_2
012C  CB 57                    BIT     2,A             ;CHECK SW I/P_3
012E  20 1F                    JR      NZ,SW_3
0130  CB 5F                    BIT     3,A             ;CHECK SW I/P_4
0132  20 21                    JR      NZ,SW_4
0134  CB 67                    BIT     4,A             ;CHECK SW I/P_5
0136  20 23                    JR      NZ,SW_5
0138  CB 6F                    BIT     5,A             ;CHECK SW I/P_6
013A  20 25                    JR      NZ,SW_6
013C  CB 77                    BIT     6,A             ;CHECK SW I/P_7
013E  20 27                    JR      NZ,SW_7
0140  C3 018D                  JP      SW_8            ;CHECK SW I/P_8
0143  21 2020          SW_1:   LD      HL,2020H
0146  C3 0170                  JP      RSW_DP
0149  21 2021          SW_2:   LD      HL,2021H
014C  C3 0170                  JP      RSW_DP
014F  21 2022          SW_3:   LD      HL,2022H
0152  C3 0170                  JP      RSW_DP
0155  21 2027          SW_4:   LD      HL,2027H
0158  C3 0170                  JP      RSW_DP
015B  21 2024          SW_5:   LD      HL,2024H
015E  C3 0170                  JP      RSW_DP
0161  21 2026          SW_6:   LD      HL,2026H
0164  C3 0170                  JP      RSW_DP
0167  21 2025          SW_7:   LD      HL,2025H
016A  C3 0170                  JP      RSW_DP
                        ;--------------------------------------
016D  CB 7F             SW_8:   BIT     7,A             ;REAT D/P
016F  28 A1                    JR      Z,JTP           ;CHECK SW D/P
0171  20 0E                    JR      NZ,DISPLAY
0173  C3 00BB                  JP      RDKEYF
0176  FD 77             DISPLAY: LD     (IY+2),A        ;STORE SW DATA IN MEMOR
0179  FD CB 01 A6               RES     4,(IY+1)        ;RESET STATUS DE
017D  3A 0D20                   LD      A,(2020H)
0180  32 0E20                   LD      (2020H+1),A
0183  3A 0D21                   LD      A,(2021H)
0186  32 0E21                   LD      (2021H+1),A
0189  3A 0D22                   LD      A,(2022H)
018C  32 0E22                   LD      (2022H+1),A
018F  3A 0D27                   LD      A,(2027H)
0192  32 0E27                   LD      (2027H+1),A
0195  3A 0D24                   LD      A,(2024H)
0198  32 0E24                   LD      (2024H+1),A
019B  3A 0D26                   LD      A,(2026H)
019E  32 0E26                   LD      (2026H+1),A
01A1  3A 0D25                   LD      A,(2025H)
01A4  32 0E25                   LD      (2025H+1),A
01A7  C3 0000                   JP      0000H           ;RESTART SW
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
018C   C3  49F                    JP      RDKEYF
018F   CD  A35            DELB :  CALL    OFFRL           ;CALL OFF ANALOG_SW
0192   C3  49                     JP      RDKEYF
                          ;-------------------------------------
0195   FD 7E 0A           ONRL:   LD      A,(IY+2)        ;LOAD SW O/P FROM MEMORY TO A
0198   06 0B                      LD      B,0BH
019A   DD 21 2020                 LD      IX,2020H
019E   FE FE                      CP      0FEH            ;CHECK SW O/P_1
01A0   20 10                      JR      NZ,CH2
01A2   DD CB 00 86         CBIT_0: RES     0,(IX)          ;RESET BIT_0 IN (IX)
01A6   DD 23                      INC     IX
01A8   10 FB                      DJNZ    CBIT_0
01AA   CB C6                      SET     0,(HL)          ;SET BIT_0 IN (HL)
01AC   CD 0296                    CALL    ONOFFR          ;ON O/P ANALOG_SW
01AF   C3 0234                    JP      OUTON
01B2   FE FD               CH2:   CP      0FDH            ;CHECK SW O/P_2
01B4   20 10                      JR      NZ,CH3
01B6   DD CB 00 8E         CBIT_1: RES     1,(IX)          ;RESET BIT_1 IN (IX)
01BA   DD 23                      INC     IX
01BC   10 FB                      DJNZ    CBIT_1
01BE   CB CE                      SET     1,(HL)          ;SET BIT_1 IN (HL)
01C0   CD 0296                    CALL    ONOFFR          ;ON O/P ANALOG_SW
01C3   C3  234                    JP      OUTON
01C6   FE FB               CH3:   CP      0FBH            ;CHECK SW O/P_3
01C8   20 10                      JR      NZ,CH4
01CA   DD CB 00 96         CBIT_2: RES     2,(IX)          ;RESET BIT_2 IN (IX)
01CE   DD 23                      INC     IX
01D0      FB                      DJNZ    CBIT_2
01D2   CB D6                      SET     2,(HL)          ;SET BIT_2 IN (HL)
01D4      02                      CALL    ONOFFR          ;ON O/P ANALOG_SW
01D7      C3                      JP      OUTON
01DA   FE F7               CH4:   CP      0F7H            ;CHECK SW O/P_4
01DC   20 10                      JR      NZ,CH5
01DE   DD CB 00 9E         CBIT_3: RES     3,(IX)          ;RESET BIT_3 IN (IX)
01E2   DD 23                      INC     IX
01E4   10 FB                      DJNZ    CBIT_3
01E6   CB DE                      SET     3,(HL)          ;SET BIT_3 IN (HL)
01E8   CD 0296                    CALL    ONOFFR          ;ON O/P ANALOG_SW
01EB   C3  234                    JP      OUTON
01EE   FE EF               CH5:   CP      0EFH            ;CHECK SW O/P_5
01F0   20 10                      JR      NZ,CH6
01F2   DD CB 00 A6         CBIT_4: RES     4,(IX)          ;RESET BIT_4 IN (IX)
01F6   DD 23                      INC     IX
01F8      FB                      DJNZ    CBIT_4
01FA   CB E6                      SET     4,(HL)          ;SET BIT_4 IN (HL)
01FC   CD  296                    CALL    ONOFFR          ;ON O/P ANALOG_SW
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
01F8  C3 0234                    JP      OUTON
01FB  FE DF              CH6:    CP      0DFH          ;CHECK SW D/P_6
01FD  20 10                      JR      NZ,CH7
01FF  DD CB 00 AE        CBIT_5: RES     5,(IX)        ;RESET BIT_5 IN (IX)
0203  DD 23                      INC     IX
0205  10 F8                      DJNZ    CBIT_5
0207  CB EE                      SET     5,(HL)        ;SET BIT_5 IN (HL)
0209  CD 0296                    CALL    ONOFFR        ;ON O/P ANALOG_SW
020C  C3 0234                    JP      OUTON
020F  FE BF              CH7:    CP      0BFH          ;CHECK SW D/P_7
0211  20 10                      JR      NZ,CH8
0213  DD CB 00 B6        CBIT_6: RES     6,(IX)        ;RESET BIT_6 IN (IX)
0217  DD 23                      INC     IX
0219  10 F8                      DJNZ    CBIT_6
021B  CB F6                      SET     6,(HL)        ;SET BIT_6 IN (HL)
021D  CD 0296                    CALL    ONOFFR        ;ON O/P ANALOG_SW
0220  C3 0234                    JP      OUTON
0223  FE 7F              CH8:    CP      07FH          ;CHECK SW D/P_8
0225  20 0C                      JR      NZ,OUTON
0227  DD CB 00 BE        CBIT_7: RES     7,(IX)        ;RESET BIT_7 IN (IX)
022B  DD 23                      INC     IX
022D  10 F8                      DJNZ    CBIT_7
022F  CB FE                      SET     7,(HL)        ;SET BIT_7 IN (HL)
0231  CD 0296                    CALL    ONOFFR        ;ON O/P ANALOG_SW
0234  C9                OUTON:  RET
                         ;-------------------------------------
0235  FD 7E 04           OFFRL:  LD      A,(IY+2)      ;LOAD SW O/P FROM MEMORY TO A
0238  FE FE                      CP      0FEH          ;CHECK SW D/P_1
023A  20 08                      JR      NZ,CH2
023C  CB B6                      RES     6,(HL)        ;DEL BIT_0 IN (HL)
023E  CD 02A6                    CALL    OFFOFFR       ;OFF O/P ANALOG_SW
0241  C3 0295                    JP      OUTOFF
0244  FE FD              CH2:    CP      0FDH          ;CHECK SW D/P_2
0246  20 0B                      JR      NZ,CH3
0248  CB BE                      RES     7,(HL)        ;DEL BIT_1 IN (HL)
024A  CD 02A6                    CALL    OFFOFFR       ;OFF O/P ANALOG_SW
024D  C3 0295                    JP      OUTOFF
0250  FE FB              CH3:    CP      0FBH          ;CHECK SW D/P_3
0252  20 0C                      JR      NZ,CH4
0254  CB 86                      RES     0,(HL)        ;DEL BIT_2 IN (HL)
0256  CD 02A6                    CALL    OFFOFFR       ;OFF O/P ANALOG_SW
0259  C3 0295                    JP      OUTOFF
025C  FE F7              CH4:    CP      0F7H          ;CHECK SW D/P_4
025E  20 0C                      JR      NZ,CH5
0260  CB 8E                      RES     1,(HL)        ;DEL BIT_3 IN (HL)
0262  CD 02A6                    CALL    OFFOFFR       ;OFF O/P ANALOG_SW
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0265  C3 0295                    JP      OUTOFF
0268  FE EF              DH5:    CP      0EFH          ;CHECK SW O/P_5
026A  20 06                     JR      NZ,DH6
026C  CB A6                     RES     4,(HL)        ;DEL BIT_4 IN (HL)
026E  CD 0296                   CALL    ONOFFR        ;OFF O/P ANALOG_SW
0271  C3 0295                   JP      OUTOFF
0274  FE DF              DH6:    CP      0DFH          ;CHECK SW O/P_6
0276  20 08                     JR      NZ,DH7
0278  CB AE                     RES     5,(HL)        ;DEL BIT_5 IN (HL)
027A  CD 0296                   CALL    ONOFFR        ;OFF O/P ANALOG_SW
027D  C3 0295                   JP      OUTOFF
0280  FE BF              DH7:    CP      0BFH          ;CHECK SW O/P_7
0282  20 08                     JR      NZ,DH8
0284  CB B6                     RES     6,(HL)        ;DEL BIT_6 IN (HL)
0286  CD 0296                   CALL    ONOFFR        ;OFF O/P ANALOG_SW
0289  C3 0295                   JP      OUTOFF
028C  FE 7F              DH8:    CP      7FH           ;CHECK SW O/P_8
028E  20 0E                     JR      NZ,OUTOFF
0290  CB BE                     RES     7,(HL)        ;DEL BIT_7 IN (HL)
0292  CD 0296                   CALL    ONOFFR        ;OFF O/P ANALOG_SW
0295  C9                OUTOFF: RET
                                ;-------------------------------------
0296  21 2020            ONOFFR: LD      HL,2020H
0299  06 08                     LD      B,08H
029B  48                        LD      C,B
029C  7E                  DIS_OP: LD      A,(HL)        ;DISPLAY O/P ON_OFF ANALOG_SW
029D  ED 79                     OUT     (C),A
029F  0C                        INC     C
02A0  23                        INC     HL
02A1  10 F9                     DJNZ    DIS_OP
02A3  FD CB 00 A6               RES     4,(IY)        ;RESET STATUS DEL
02A7  CD 02E2                   CALL    DIS_S         ;DISPLAY STATUS
02AA  FD CB 01 86               RES     0,(IY+1)      ;RESET STATUS SW I/P
02AE  C9                        RET
                                ;-------------------------------------
02AF  FD CB 00 4E         INT0: BIT     1,(IY         ;CHECK STATUS PROM_1
02B3  20 2E                     JR      NZ,OUTO
02B5  FD CB 00 56               BIT     2,(IY)        ;CHECK STATUS PROM_2
02B9  20 28                     JR      NZ,OUTO
02BB  FD CB 00 5E               BIT     3,(IY)        ;CHECK STATUS PROM_3
02BF  20 22                     JR      NZ,OUTO
02C1  FD CB 00 86               RES     0,(IY)
02C5  CD 02E2                   CALL    DIS_S
02C8  21 2020                   LD      HL,2020H
02CB  CD 0358                   CALL    CLEARM        ;CLEAR MEMORY OF NORMAL
02CE  CD 0365                   CALL    CLEARO        ;CLEAR O/P RELAY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
02D1  C9                  OUTC:   RET
                          ;--------------------------------------
02D2  FD CB 00 46         INTP1:  BIT     0,(IY)          ;CHECK STAUSW SAVE
02D6  20 1D               JR      NZ,SAVEP1
02D8  FD CB 00 4E         BIT     1,(IY)          ;CHECK STATUS PROM_1
02DC  20 1D               JR      NZ,OUTP1
02DE  AF                  XOR     A
02DF  FD 77 00            LD      (IY),A          ;CLEAR STATUS
02E2  FD CB 00 CE         SET     1,(IY)          ;SET STATUS PROM_1
02E6  CD 0380             CALL    DIS_S           ;DISPLAY STATUS
02E9  CD 0363             CALL    CLEAR0          ;CLEAR O/P ANALOG_SW
02EC  21 202B             LD      HL,202BH
02EF  CD 0350             CALL    DIS_RL          ;ON PROM_1
02F2  C3 02F8             JP      OUTP1
02F5  11 2028             SAVEP1: LD      DE,202SH
02F8  CD 036D             CALL    SAVEPT          ;SAVE PROM_1
02FB  C9                  OUTP1:  RET
                          ;--------------------------------------
02FC  FD CB 00 46         INTP2:  BIT     0,(IY)          ;CHECK STAUSW SAVE
0300  20 1D               JR      NZ,SAVEP2
0302  FD CB 00 56         BIT     2,(IY)          ;CHECK STATUS PROM_2
0306  20 1D               JR      NZ,OUTP2
0308  AF                  XOR     A
0309  FD 77 00            LD      (IY),A          ;CLEAR STATUS
030C  FD CB 00 D6         SET     2,(IY)          ;SET STATUS PROM_2
0310  CD 0380             CALL    DIS_S           ;DISPLAY STATUS
0313  CD 0363             CALL    CLEAR0          ;CLEAR O/P ANALOG_SW
0316  21 2030             LD      HL,2030H
0319  CD 0350             CALL    DIS_RL          ;ON PROM_2
031C  C3 0325             JP      OUTP2
031F  11 2030             SAVEP2: LD      DE,2030H
0322  CD 036D             CALL    SAVEPT          ;SAVE PROM_2
0325  C9                  OUTP2:  RET
                          ;--------------------------------------
0326  FD CB 00 46         INTP3:  BIT     0,(IY)          ;CHECK STAUSW SAVE
032A  20 1D               JR      NZ,SAVEP3
032C  FD CB 00 5E         BIT     3,(IY)          ;CHECK STATUS PROM_3
0330  20 1D               JR      NZ,OUTP3
0332  AF                  XOR     A
0333  FD 77 00            LD      (IY),A          ;CLEAR STATUS
0336  FD CB 00 DE         SET     3,(IY)          ;SET STATUS PROM_3
033A  CD 0380             CALL    DIS_S           ;DISPLAY STATUS
033D  CD 0363             CALL    CLEAR0          ;CLEAR O/P ANALOG_SW
0340  21 2038             LD      HL,2038H
0343  CD 0350             CALL    DIS_RL          ;ON PROM_3
0346  C3 034F             JP      OUTP3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
0349  11 2038          SAVEP3:  LD      DE,2038H
034C  CD 036D                  CALL    SAVEP7          ;SAVE PROM_3
034F  C9               OUTP3:   RET
                       ;---------------------------------------
0350  06 08            DIS_BL:  LD      B,08H           ;DISPLAY O/PO ANALOG_SW PROM_1_2_3
0352  48                       LD      C,B
0353  7E               DISP:    LD      A,(HL)
0354  ED 79                    OUT     (C),A
0356  0C                       INC     C
0357  23                       INC     HL
0358  10 F9                    DJNZ    DISP
035A  C9                       RET
                       ;---------------------------------------
035B  06 08            CLEARM:  LD      B,08H           ;CLEAR MEMORY
035D  AF                       XOR     A
035E  77               CL:      LD      (HL),A
035F  23                       INC     HL
0360  10 FC                    DJNZ    CL
0362  C9                       RET
                       ;---------------------------------------
0363  06 08            CLEARO:  LD      B,08H           ;CLEAR O/P ANALOG_SW
0365  48                       LD      C,B
0366  AF                       XOR     A
0367  ED 79            LOO:     OUT     (C),A
0369  0C                       INC     C
036A  10 FB                    DJNZ    LOO
036C  C9                       RET
                       ;---------------------------------------
036D  06 2E            SAVEP7:  LD      B,08H           ;SAVE PROM_1_2_3
036F  21 2020                  LD      HL,2020H
0372  7E               LSP:     LD      A,(HL)
0373  12                       LD      (DE),A
0374  23                       INC     HL
0375  13                       INC     DE
0376  10 FA                    DJNZ    LSP
0378  FD CB 06 86              RES     0,(IY
037C  CD 0350                  CALL    DIS_B
037F  C9                       RET
                       ;---------------------------------------
0380  FD 7E 00         DIS_B    LD      A,(IY           ;DISPLAY STATUS
0383  D3 FF                    OUT     (0FFH),A
0385  C9                       RET
                       ;---------------------------------------
                               END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MACROS:

5 obsolete:

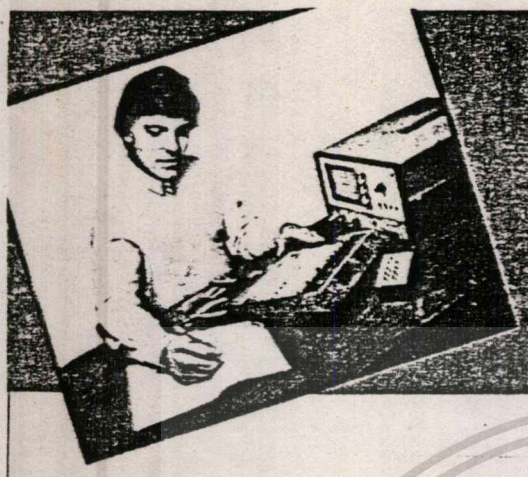
CBIT_0	0196	CBIT_1	01AF	CBIT_2	01C5	CBIT_3	01E7
CBIT_4	01EB	CBIT_5	01FF	CBIT_6	0213	CBIT_7	0227
CH2	01AB	CH3	01BF	CH4	01D3	CH5	01E7
CH6	01FB	CH7	020F	CH8	0223	CH8KEY	0179
CL	^35E	CLEAR	0CEE	CLEARM	025E	CLEARO	0363
COM	004C	COMF	0044	DEL	0^D6	DELAY	005E
DELAL	01BB	DH2	0244	DH3	0259	DH4	025C
DH5	0269	DH6	0274	DH7	028^	DH8	028C
DISP	0353	DIS_OF	029L	DIS_RL	0350	DIS_S	03BC
DLY	005B	INSM	0061	INTC	01AF	INTP1	02D2
INTP2	02EC	INTP3	0326	LDB	036^	LN	0054
LSP	0372	MAINP	0076	NOR	0^F6	NORMAL	011C
OFFRL	0235	ONOFFR	0296	ONSL	01BE	OUTC	02D1
OUTDEL	^0E3	OUTOFF	0295	OUTBN	0134	OUTP1	02E9
OUTP2	0315	OUTP3	034F	PASE	0C2A	PORTC	0063
PORTDS	0007	PORTSA	0004	PORTSB	0062	PORTSI	0066
PORTSO	00a1	PORTSP	0002	PORTSS	0C05	PROM_1	0004
PROM_2	0JLA	PROM_3	0080	RDACYC	0 E9	RESDS	0085
RSW_OF	0170	S-VE	005C	SAVEP1	02F5	SAVEP2	031F
SAVEP3	0349	SAVEP4	036C	SETSSA	0121	STAC	2602
SW	0066	SW_1	0143	SW_2	0^4^	SW_3	014F
SW_4	015B	SW_5	0167	SW_6	0181	SW_7	01A7
SW_8	1EF						

No fatal errors

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Z80 กับ 8255

การใช้งานไมโครโปรเซสเซอร์ส่วนใหญ่ จะต้องเชื่อมโยงกับอุปกรณ์ภายนอก เช่น สวิตช์, รีเลย์ หรือตัวตรวจจับอื่น ๆ การเชื่อมต่อในลักษณะดังกล่าวจะเชื่อมโยงกับพอร์ตอินพุต–เอาต์พุต เพื่อให้ไมโครโปรเซสเซอร์ส่งสัญญาณควบคุมไปยังอุปกรณ์ต่าง ๆ ตามเงื่อนไขที่เกิดขึ้นและตรวจสอบได้ด้วยไมโคร–โปรเซสเซอร์เอง

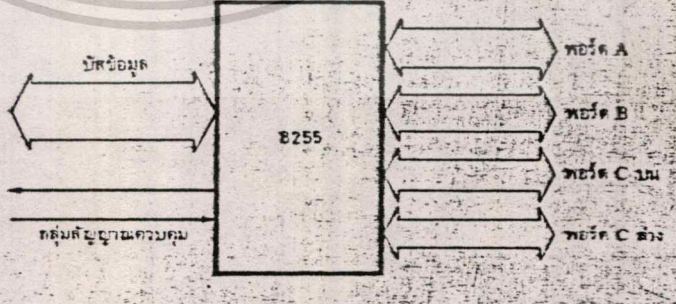
การเชื่อมต่อกับพอร์ตอินพุตในลักษณะที่ง่ายที่สุด คือ การเชื่อมโดยใช้เกตลอจิกสามสถานะ โดยสัญญาณควบคุมพอร์ตอินพุตจะเป็นตัวไปเปิดเกตให้ข้อมูลเข้าสู่บัสและไมโครโปรเซสเซอร์จะอ่านเข้าไป แต่สำหรับพอร์ตเอาต์พุต จะใช้แลตช์ฟลิปฟลอปทำหน้าที่รับสัญญาณข้อมูลจากไมโครโปรเซสเซอร์ที่ส่งเข้าไปในบัสและได้รับการจับไว้ที่พอร์ตในขณะที่มีสัญญาณควบคุมพอร์ตทริกมาที่ขาแลตช์

พอร์ตอินพุต–เอาต์พุตที่ใช้เกตขนาดเล็กดังกล่าว ยังมีจุดอ่อนในเรื่องของจำนวนไอซี ซึ่งอาจต้องใช้หลายชิพ (ถ้าต้องการหลายพอร์ต) และยากที่จะกำหนดลักษณะการทำงานให้แตกต่างไปจากวงจรเดิมที่ออกแบบไว้ บริษัทผู้ออกแบบไมโครโปรเซสเซอร์ส่วนใหญ่จึงออกแบบ LSI ชิพ เพื่อทำหน้าที่เป็นอินพุต–เอาต์พุตของระบบ ซึ่งมีข้อดีในเรื่องการใช้งานได้ง่าย ในตอนนี้จะได้กล่าวถึงการประยุกต์ใช้ไอซี LSI ที่ทำหน้าที่เป็นพอร์ตอินพุต–เอาต์พุต ตัวที่รู้จักกันดีมากที่สุด มีราคาถูก หาได้ง่าย ไอซี LSI ดังกล่าวคือ 8255 ของบริษัทอินเทล

8255 เป็นไอซีในตระกูลของ 8080 เพราะอินเทลได้ออกแบบมาให้ใช้งานร่วมกับซีพียู 8080 แต่อย่างไรก็ตาม เราสามารถนำมาประยุกต์ใช้กับ Z80 ได้ง่ายเช่นกัน

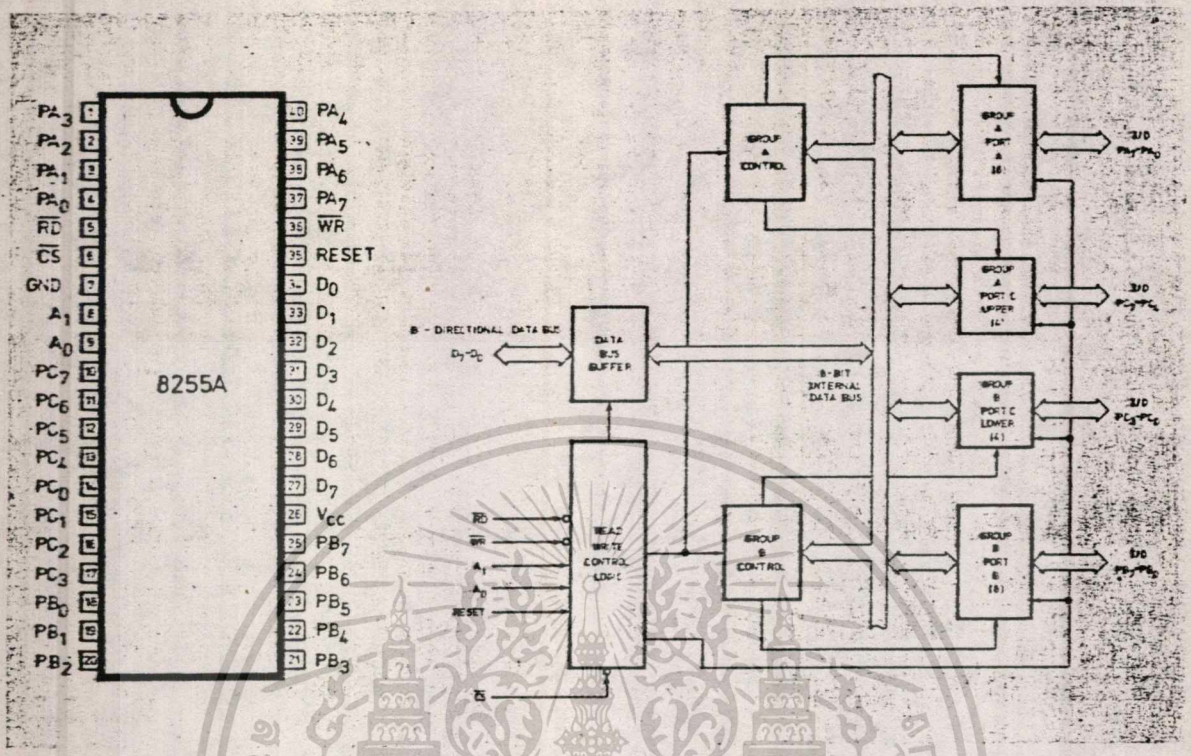
รู้จักกับ 8255

8255 เป็นไอซี 40 ขา ตัวไอซี 8255 ได้รับการออกแบบมาให้มีสัญญาณเชื่อมโยงกับ 8080 แต่สัญญาณนี้พอเหมาะที่ใช้กับ Z80 ได้ดีเช่นเดียวกัน ตัว 8255 เป็นไอซีที่ต่อเป็นพอร์ตให้ไมโครโปร–เซสเซอร์ได้ 3 พอร์ต โดยมีโครงสร้างพื้นฐานแสดงด้วยบล็อกไดอะแกรมได้ดังรูปที่ 1



รูปที่ 1 ไดอะแกรมของไอซี 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2 แสดงการจัดขาไอซี 8255 และแผนผังวงจรภายใน

การเรียงพอร์ตของ 8255 จะเรียกพอร์ตว่า พอร์ต A, B และ C โดยพอร์ต C แยกเป็น 2 ส่วน คือ PC_0-PC_3 เรียกว่า พอร์ต C ล่าง จำนวน 4 บิต และพอร์ต C บน คือ PC_4-PC_7 ที่พิเศษคือ พอร์ตทุกพอร์ตเป็นได้ทั้งพอร์ตอินพุตและพอร์ตเอาต์พุต

รูปที่ 2 เป็นแผนผังภายในตัวไอซี และการจัดวางขาของไอซี 8255 โดยนำมาจากคู่มือไอซีที่บริษัทผู้ผลิตพิมพ์แจกจ่าย

การทำงานของวงจรจะใช้สัญญาณควบคุมจากไมโครโปรเซสเซอร์มาควบคุมการทำงาน โดยไมโครโปรเซสเซอร์ส่งคำสั่งมาโปรแกรมการทำงานหรือกำหนดรูปแบบของพอร์ตให้เป็นอินพุตหรือเอาต์พุตได้

ขาต่าง ๆ ของ 8255

เพื่อให้เข้าใจวิธีการต่อใช้งานระหว่าง Z80 กับ 8255 จึงจำเป็นต้องเข้าใจความหมายและตำแหน่งของขาต่าง ๆ เสียก่อน ขาทั้ง 40 ขาของไอซีประกอบด้วย

D_0-D_7 เป็นขาข้อมูลอินพุต–เอาต์พุตที่จะต้องผ่านเข้าออกจากส่วนนี้ D_0-D_7 จึงต่อเข้ากับระบบบัสของไมโครโปรเซสเซอร์ เพื่อให้ไมโครโปรเซสเซอร์สามารถอ่านหรือเขียนข้อมูลจากพอร์ตผ่านทางบัสนี้

$\overline{CS}$ (เลือกชิพ) ขานี้เป็นขาอินพุตที่จะรับสัญญาณจากภายนอกเพื่อเลือกชิพ 8255 นี้ โดยเมื่อขานี้เป็นลอจิก "0" จะทำให้ตัว 8255 ต่อเข้ากับระบบบัสของไมโครโปรเซสเซอร์ เพื่อให้ไมโครโปรเซสเซอร์เขียนหรืออ่านข้อมูลจากพอร์ตได้

$\overline{RD}$ ขาสัญญาณการอ่าน เป็นสัญญาณอินพุตที่ต้องส่งมาจาก CPU เมื่อสัญญาณที่ขานี้เป็น "0" และ $\overline{CS}$ เป็น "0" ด้วย ตัว 8255 จะทำตัวให้ซีพียูอ่านข้อมูลจากบัสในขณะเป็นพอร์ตอินพุต

$\overline{WR}$ ขาสัญญาณการเขียน จะแอคทีฟเมื่อสัญญาณ $\overline{WR}$ เป็น "0" และ $\overline{CS}$ เป็น "0" สัญญาณนี้จะมาจาก CPU เมื่อต้องการเขียนข้อมูลลงบนพอร์ตที่กำหนด

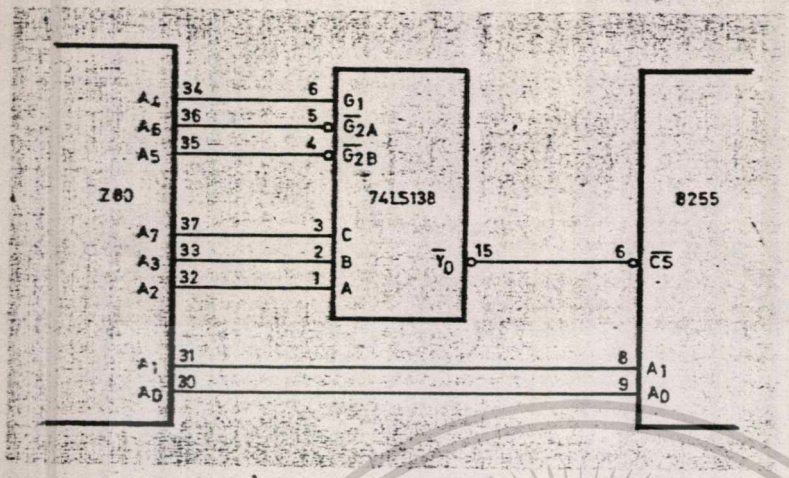
A_0-A_1 ขาแอดเดรส ลอจิกของขาทั้งสองจะถอดรหัสออกเป็น 4 เพื่อกำหนดรีจิสเตอร์ภายในที่เชื่อมต่อกับพอร์ตอินพุต–เอาต์พุตของ 8255

$\overline{RESET}$ ขารีเซต เป็นสัญญาณที่ส่งจากภายนอกเข้ามาทำการรีเซต 8255 เพื่อเคลียร์สถานะต่าง ๆ ของ 8255 เมื่อ 8255 ได้รับการรีเซต มันจะกลับเข้าสู่โหมดอินพุต หรือ ทุกพอร์ตเป็นพอร์ตอินพุต

PA_0-PA_7 เป็นสายสัญญาณที่เป็นพอร์ตของ 8255 ที่ชื่อพอร์ต A การเลือกพอร์ตจะเลือกโดยขาแอดเดรส A_0-A_1

PB_0-PB_7 เป็นสายสัญญาณที่เป็นพอร์ต B ของ 8255 ที่ถูกเลือกโดยขาแอดเดรส A_0-A_1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 การกำหนดแอดเดรสให้กับ 8255

PC_0-PC_3 เป็นสายสัญญาณที่เป็นพอร์ต C ของ 8255 การกำหนดพอร์ตนี้จะได้รับการกำหนดโดยขาแอดเดรส A_0-A_1 พอร์ต C นี้แบ่งเป็น 2 กลุ่ม คือ กลุ่ม PC_0-PC_3 และกลุ่ม PC_4-PC_7

การเชื่อมต่อ 8255 กับ Z80

หากพิจารณาจากขาต่าง ๆ ของ 8255 จะเห็นว่าส่วนขาควบคุมที่จะเชื่อมโยงเข้ากับบัสของไมโครโปรเซสเซอร์นั้น สามารถเชื่อมโยงกับบัสได้โดยไม่ยาก ในที่นี้จะลองต่อ 8255 เป็นพอร์ตให้กับ Z80 ในที่นี้สมมติว่าต้องการให้ Z80 มองเห็น 8255 เป็นพอร์ตหมายเลข 10H, 11H, 12H และ 13H เราจะเชื่อมต่อสายสัญญาณการเลือกแอดเดรสของพอร์ตได้ดังรูปที่ 3

สังเกตว่า ขณะ $\overline{CS}$ แอคทีฟนั้น ขา A_7, A_6, A_5, A_4, A_3, A_2 จะต้องมีข้อมูล 000100 และเมื่อรวมกับ A_1, A_0 จะเป็น 000100XX พอร์ตที่เกิดขึ้นเมื่อ A_0, A_1 เป็น 00 คือพอร์ต 10H และถ้า A_1, A_0 เป็น 11 พอร์ตจะเป็น 13H การกำหนดพอร์ตของ Z80 จะใช้ข้อมูลบนบัสแอดเดรส 8 เส้น คือ A_0-A_7 เท่านั้น

สัญญาณที่จะควบคุม 8255 อีกชุดหนึ่ง คือ สัญญาณควบคุมการเขียนและอ่าน หากสัญญาณ $\overline{WR}$ แอคทีฟเป็น "0" จะหมายถึง การเขียนพอร์ต หรือส่งข้อมูลให้พอร์ตเอาต์พุตนั่นเอง แต่หากสัญญาณ $\overline{RD}$ แอคทีฟเป็น "0" จะหมายถึง การอ่านพอร์ตหรือรับข้อมูลอินพุตนั่นเอง

เพื่อให้แยกกันระหว่างการเขียน/อ่านหน่วยความจำและเขียน/อ่านพอร์ต-อินพุต-เอาต์พุต จึงต้องใช้ขา $\overline{IORQ}$ ร่วมด้วย กล่าวคือ ถ้าสัญญาณ $\overline{WR}$ เกิดขึ้นพร้อม $\overline{IORQ}$ จะหมายถึง $\overline{IOW}$ หรือสัญญาณเขียนพอร์ต และถ้าให้ $\overline{IORQ}$ แอคทีฟพร้อมกับ $\overline{RD}$ จะมีความหมาย $\overline{IOR}$ หรือสัญญาณอ่านพอร์ต

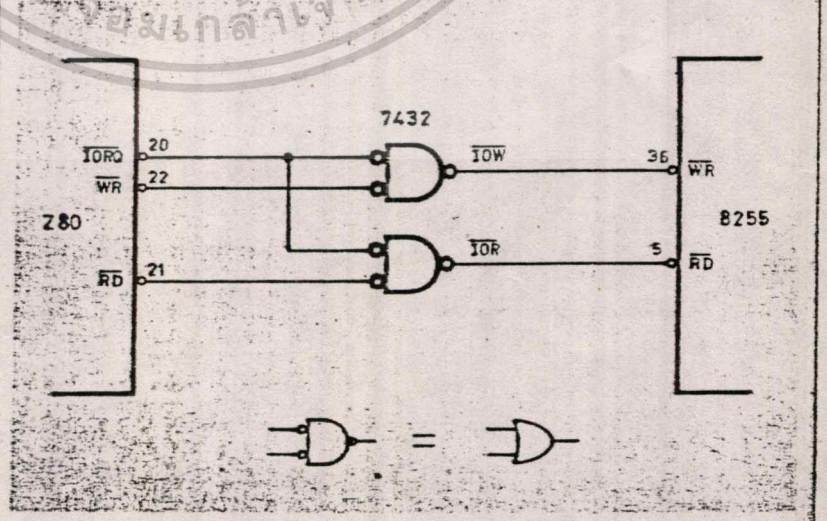
วงจรการเชื่อมต่อของสัญญาณส่วนควบคุมแสดงได้ดังรูปที่ 4

เมื่อเชื่อมต่อเป็นระบบ เราต้องเชื่อมสายสัญญาณ $\overline{RESET}$ ของ Z80 มายังขา RESET ของ 8255 การรีเซตของ 8255 ใช้ลอจิก "1" ซึ่งตรงข้ามกับ Z80 ดังนั้นจำเป็นต้องมีอินเวอร์เตอร์เปลี่ยนลอจิกก่อน การที่ต้องรีเซต 8255 พร้อมกับ Z80 ก็เนื่องจากว่า ขณะที่ Z80 รีเซตแล้ว เราจะเริ่มจากให้พอร์ตทุกพอร์ตของ 8255 เป็นอินพุต เพื่อว่ามิฉะนั้น อาจมีข้อมูลออกไปยังพอร์ตเอาต์พุตที่เรายังไม่พึงประสงค์ ซึ่งอาจจะทำให้ระบบอินเตอร์เฟสภายนอกมีปัญหาได้ เพราะเราไม่รู้สถานะที่แน่นอนของ 8255 ก่อนการโปรแกรมโหมดการทำงาน

ระบบการเชื่อมต่อของ 8255 กับ Z80 ทั้งระบบแสดงดังรูปที่ 5

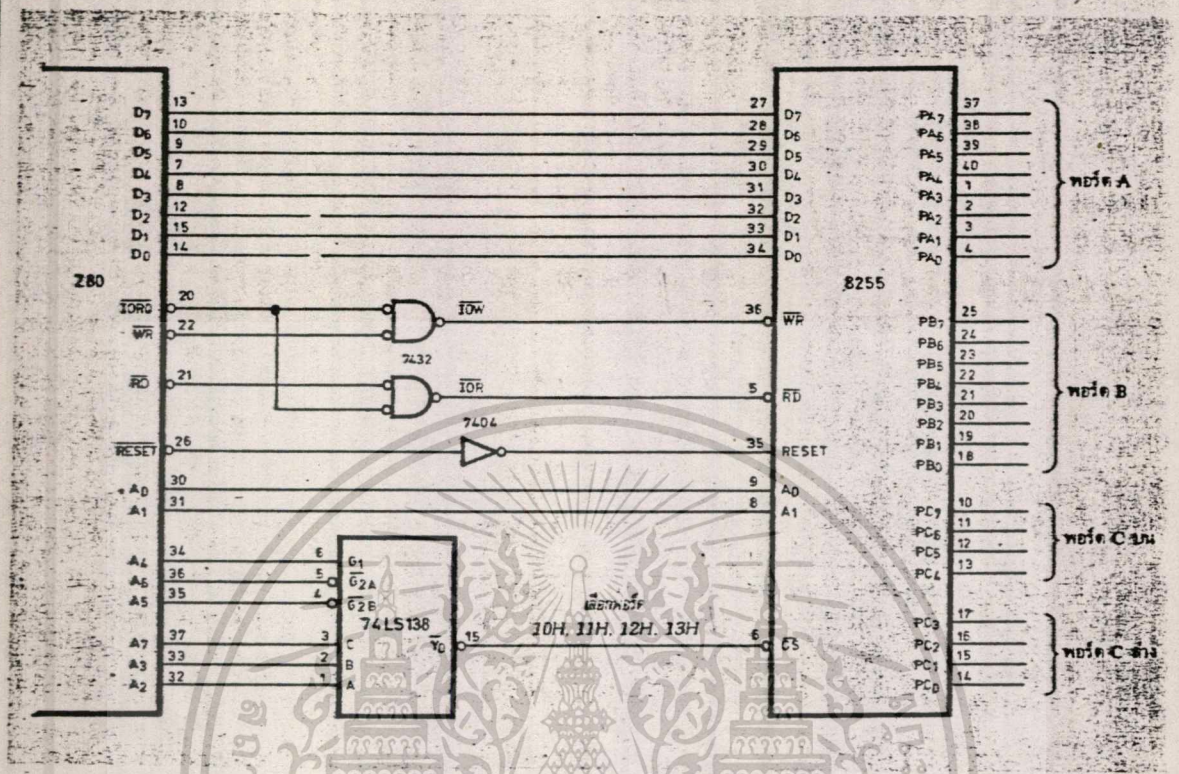
รีจิสเตอร์ภายในของ 8255

เมื่อเราต่อ 8255 เข้ากับ Z80 ได้แล้ว สิ่งที่ผู้ใช้จะต้องทำคือ การโปรแกรมให้ 8255 ทำงานตามที่ต้องการ จากการที่ 8255 มีพอร์ตที่ Z80 มองเห็น 4 พอร์ต แต่ละพอร์ตจะเห็นเสมือนเป็นรีจิสเตอร์ที่เขียน/อ่านได้ รีจิสเตอร์แต่ละตัวนี้จึงถูกกำหนดด้วยแอดเดรสตามที่ตั้งไว้ เช่น



รูปที่ 4 วงจรการเชื่อมต่อสายสัญญาณควบคุมการเขียน/อ่าน 8255

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5 การเชื่อมต่อ 8255 กับ Z80 ทั้งระบบ

ในกรณีนี้เป็นแอดเดรส 10H, 11H, 12H และ 13H รีจิสเตอร์แต่ละตัวจะได้รับการกำหนดควบคู่กับ $\overline{RD}$ และ $\overline{WR}$ เพื่อแสดงความหมาย ดังตัวอย่างเช่น พอร์ต 10H เป็นพอร์ต A ซึ่งเมื่อเขียนที่พอร์ตนี้ จะเป็นการส่งข้อมูลเอาต์พุต และถ้าอ่านพอร์ตนี้ก็จะเป็นการอินพุตจากพอร์ตนี้ ดังนั้นสัญญาณของขาควบคุมที่ประกอบกันจะแสดงความหมายดังตารางที่ 1

การใช้งาน 8255 จะต้องส่งรหัสควบคุม (control byte) เข้าไปยังพอร์ตข้อมูลควบคุม เพื่อควบคุมการทำงานของ 8255 โดยใช้สัญญาณควบคุมพอร์ตหมายเลข 13H การควบคุมการทำงานของ 8255 มีหลายโหมด แต่ละโหมดจะแตกต่างกันออกไป การโปรแกรมให้ 8255 ทำงานจะทำได้ 3 โหมด คือ โหมด 0 โหมด 1 และโหมด 2

ตารางที่ 1 สัญญาณควบคุมการกระทำของ 8255

$\overline{RD}$	$\overline{WR}$	A1	A0	ความหมาย
1	0	0	0	เขียนพอร์ต A ซึ่งเป็นข้อมูล
0	1	0	0	อ่านพอร์ต A ซึ่งเป็นข้อมูล
1	0	0	1	เขียนพอร์ต B ซึ่งเป็นข้อมูล
0	1	0	1	อ่านพอร์ต B ซึ่งเป็นข้อมูล
1	0	1	0	เขียนพอร์ต C ซึ่งเป็นข้อมูล
0	1	1	0	อ่านพอร์ต C ซึ่งเป็นข้อมูล
1	0	1	1	เขียนข้อมูล ซึ่งเป็นรหัสควบคุม
0	1	1	1	อ่านเข้ามาซึ่งไม่มีความหมายใด

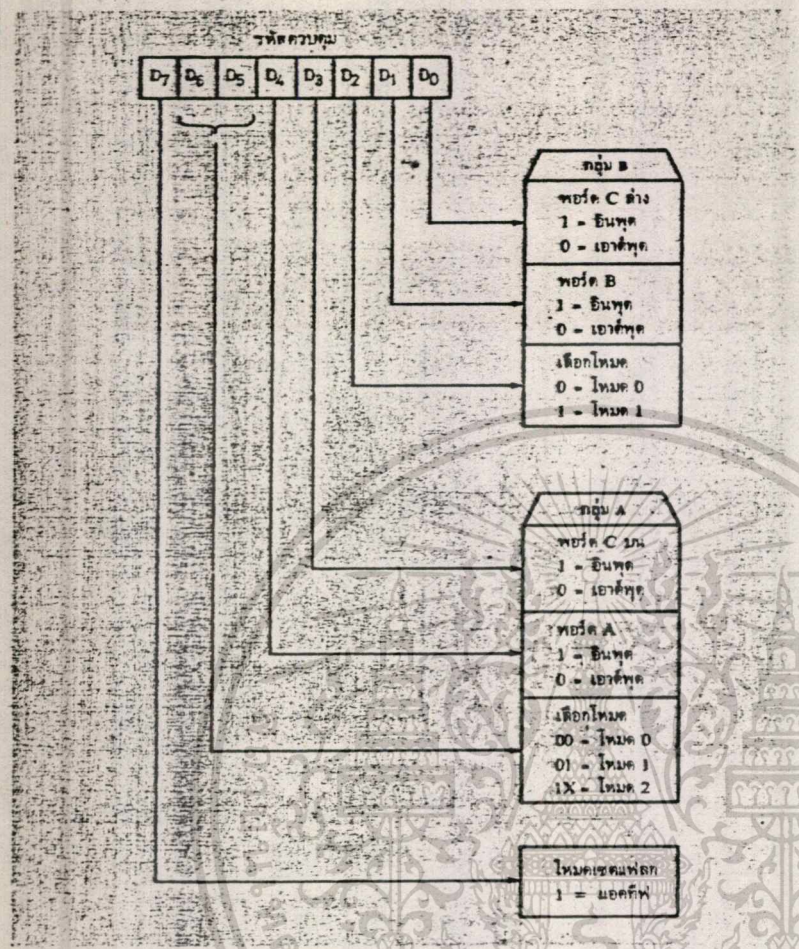
โหมด 0 : อินพุต-เอาต์พุต แบบพื้นฐาน

การกำหนดโหมดการทำงาน จะต้องส่งข้อมูลคำสั่งเข้าไปโปรแกรมในพอร์ตควบคุมของ 8255 ซึ่งในที่นี้ใช้พอร์ตหมายเลข 13H (ตามรูปที่ 5) แต่ละบิตของข้อมูลที่ส่งไป จะมีความหมายในตัวเอง ลักษณะความหมายของแต่ละบิตของรหัสควบคุมแสดงได้ดังรูปที่ 6

การโปรแกรม 8255 คือ การให้ค่ารหัสบิตต่าง ๆ เข้าไปในรหัสควบคุม แล้วส่งไปยังรีจิสเตอร์ของพอร์ตควบคุม ลองดูความหมายของบิตต่าง ๆ ดังนี้

บิต D_7 เป็นบิตแสดงรหัสคำสั่งควบคุม ถ้าบิตนี้เป็น "1" จะหมายถึง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6 ความหมายของบิตต่างๆ ของรหัสควบคุม (controlbyte)

รหัสควบคุมนี้จะมีผลต่อการเปลี่ยนแปลงการเซตโหมดต่าง ๆ ของ 8255

บิต D_6 และ D_5 เป็นการเลือกโหมดของพอร์ต A ซึ่งมี 3 โหมด คือ โหมด 0, 1 และ 2 ตามค่าในรูปที่ 6

บิต D_4 ถ้ามีค่าทำกับ "0" มีความหมายถึงการกำหนดพอร์ต A เป็นเอาต์พุต ถ้ามีค่าเป็น "1" จะกำหนดให้พอร์ต A เป็นอินพุต

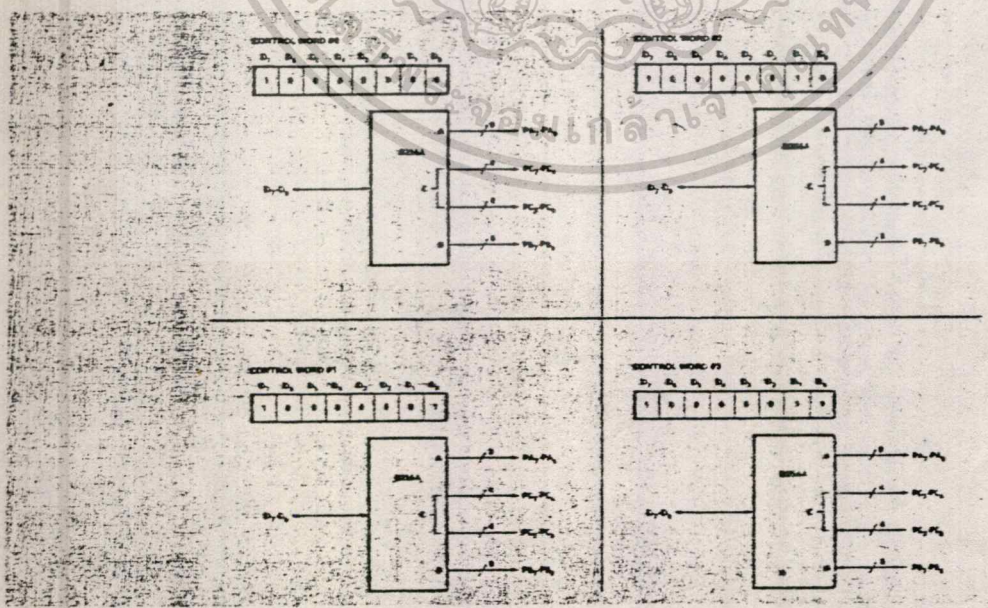
บิต D_3 เป็นบิตที่บอกการเซตของพอร์ต C บน ถ้าเป็น "0" จะทำให้พอร์ต C บนเป็นเอาต์พุต

บิต D_2 เป็นบิตที่บอกการเซตโหมดของพอร์ต B ถ้าเป็น "0" หมายถึง การเลือกพอร์ต B เป็นโหมด 0 และถ้าเป็น "1" คือ การเลือกโหมด 1

บิต D_1 เป็นการกำหนดอินพุต–เอาต์พุตของพอร์ต B ถ้าเป็น "0" คือ เอาต์พุต แต่ถ้าเป็น "1" คืออินพุต

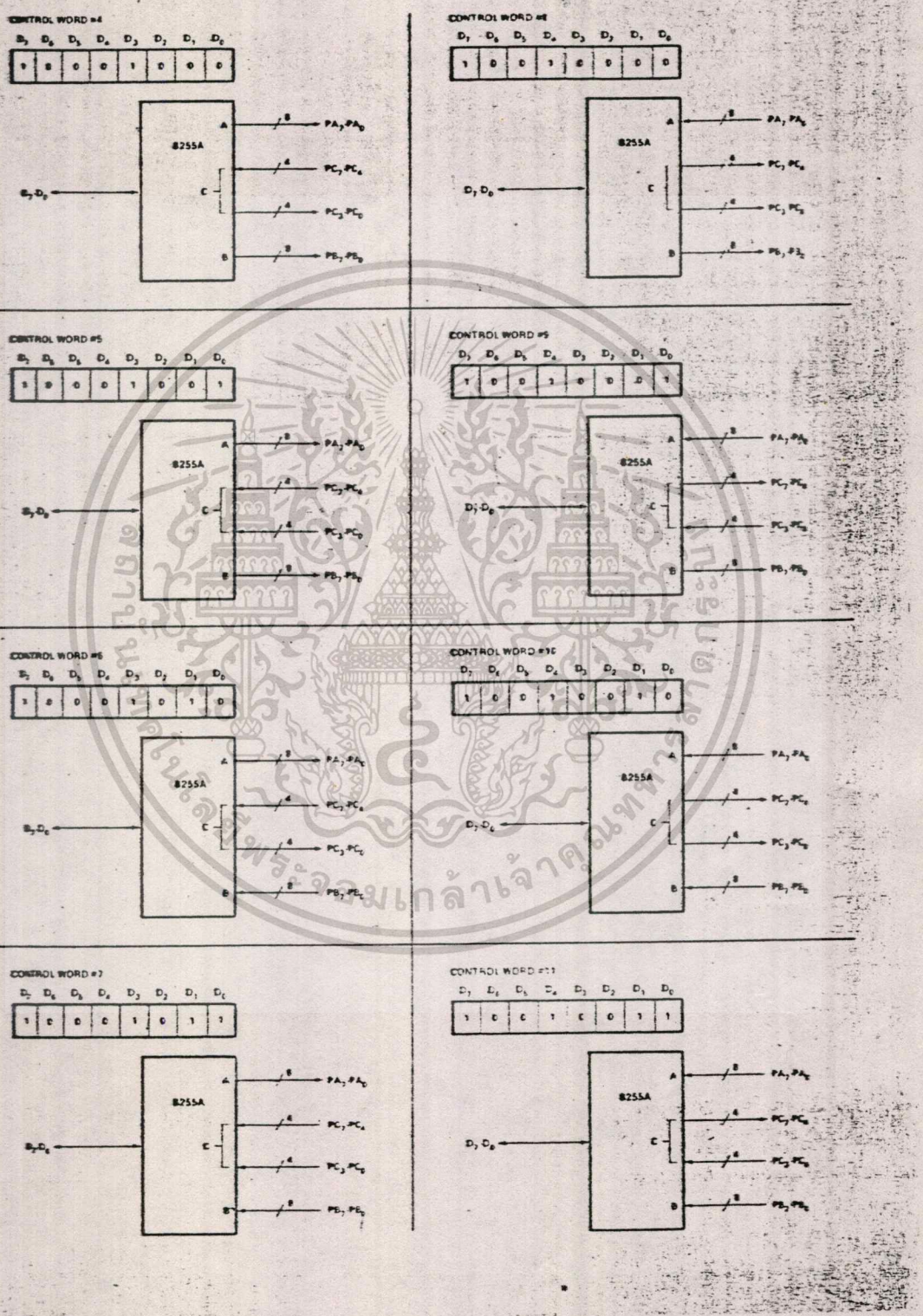
บิต D_0 เป็นการกำหนดอินพุต–เอาต์พุตของพอร์ต C ล่าง ถ้าเป็น "0" คือเอาต์พุต และ "1" คืออินพุต

การโปรแกรม 8255 จะเริ่มจากการเซตค่าที่ต้องการแล้วเอาต์พุตไปยัง



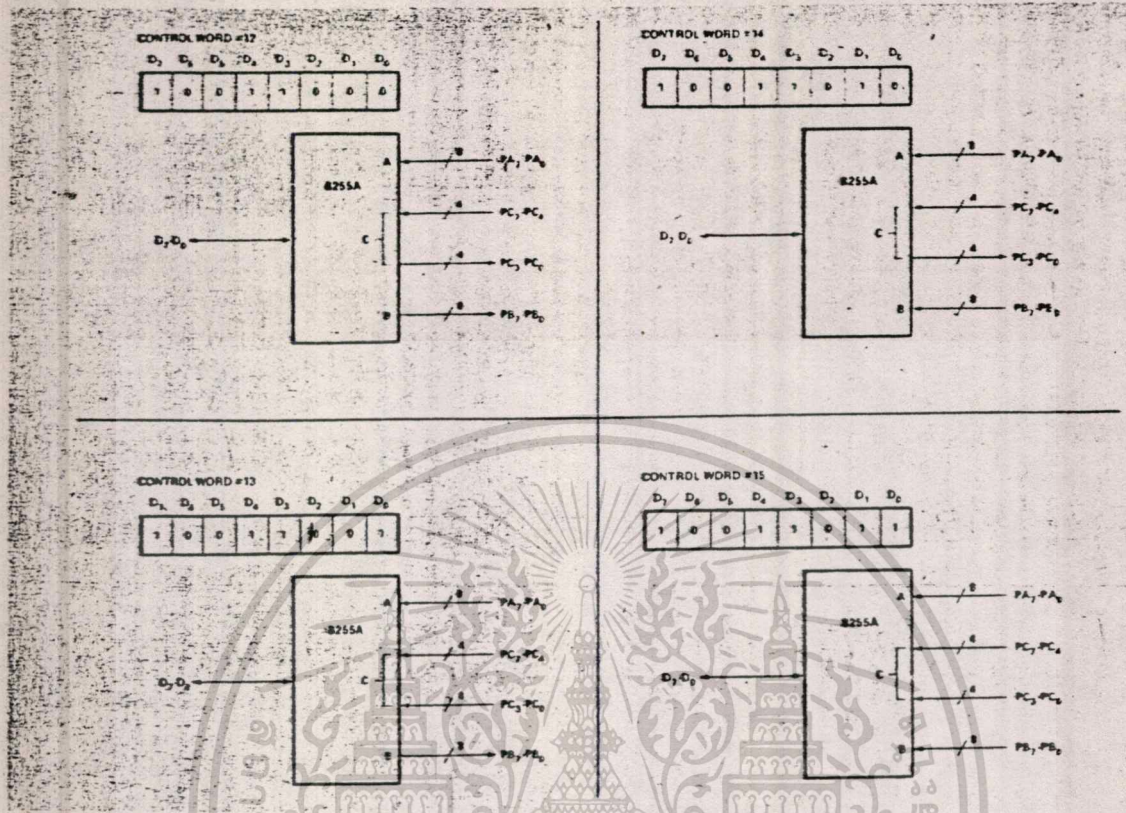
รูปที่ 7 ลักษณะของรหัสควบคุมที่เป็นไปได้ในโหมด 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7 ลักษณะของรหัสควบคุมที่เป็นไปได้ในโหมด 0 (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7 ลักษณะของรหัสควบคุมที่เป็นไปได้ในโหมด 0 (ต่อ)

พอร์ตควบคุม เช่น ต้องการโปรแกรมให้พอร์ต A, B และ C ทั้งสามพอร์ตเป็นเอาต์พุตหมด เราจะเลือก 8255 ให้อยู่ในโหมด 0 โดยมีรหัสควบคุมเป็น 10000000 หรือ 80H ดังนั้นจึงเขียนเป็นคำสั่งได้เป็น

```
LD A, 80H     ; กำหนดรหัสควบคุม
OUT (13H), A  ; ส่งไปยังพอร์ตควบคุม
```

หลังจากคำสั่ง OUT นี้ผ่านไปแล้ว พอร์ต A, B และ C จะเป็นพอร์ตเอาต์พุตหมด ซึ่งก็จะส่งข้อมูลจาก CPU ไปยังพอร์ตต่าง ๆ ได้ เช่น ถ้าต้องการส่งข้อมูล 8AH ไปยังพอร์ต A, 41H ไปยังพอร์ต B และ 25H ไปยังพอร์ต C คำสั่งที่ใช้จะเป็นดังนี้

```
LD A, 8AH     ; เลือกค่า 8AH
OUT (10H), A  ; ส่งให้พอร์ต A
LD A, 41H     ; เลือกค่า 41H
OUT (11H), A  ; ส่งให้พอร์ต B
LD A, 25H     ; เลือกค่า 25H
OUT (12H), A  ; ส่งให้พอร์ต C
```

เนื่องจากมีพอร์ตที่รับส่งข้อมูล 3 พอร์ต แต่พอร์ต C ซึ่งแยกออกเป็น 2 ส่วน คือ C ล่าง และ C บน ซึ่งเราสามารถโปรแกรมให้ทั้ง 4 พอร์ตนี้ เป็นอินพุตหรือเอาต์พุตก็ได้ เช่น ถ้าให้รหัสควบคุมเป็น 82H จะทำให้พอร์ต B เป็นอินพุตและพอร์ต A และ C เป็นเอาต์พุต

การทำงานในโหมด 0

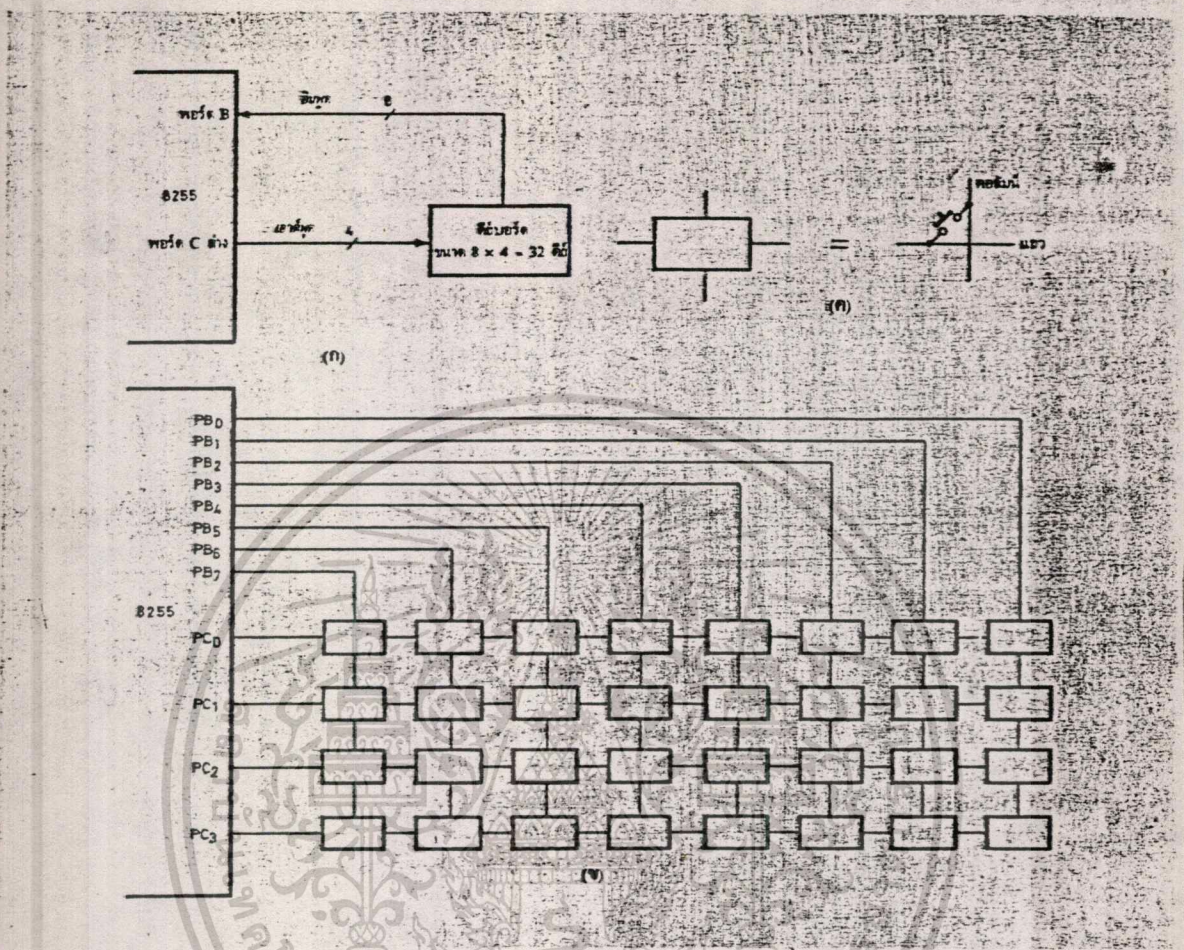
โหมด 0 เป็นโหมดที่กำหนดให้พอร์ตทุกพอร์ตบนตัว 8255 เป็นพอร์ตอินพุต–เอาต์พุตแบบพื้นฐาน รูปแบบความเป็นไปได้จึงมีทั้งสิ้น 16 รูปแบบ ตามลักษณะของพอร์ต A, B, C บนและ C ล่าง ลักษณะของรหัสควบคุมแต่ละแบบจะเป็นดังรูปที่ 7

ตัวอย่างการใช้งานของ 8255 ในโหมด 0 นี้ เช่น เมื่อต้องการให้พอร์ต B เป็นอินพุตและพอร์ต C ล่าง เป็นเอาต์พุต เพื่อรับรู้การกดคีย์บอร์ดและหาดูว่ามีคีย์ใดกด ซึ่งเราสามารถใช้การจัดคีย์บอร์ดในรูปแบบแมทริกซ์ได้ ดังรูปที่ 8

การต่อคีย์จำนวนมากที่เรานิยมต่อกันแบบแมทริกซ์ เพราะเป็นโครงสร้างทางฮาร์ดแวร์แบบประหยัด และลดข้อยุ่งยากไปได้ แต่ก็ต้องใช้ซอฟต์แวร์ในการควบคุมหรือตรวจสอบว่า คีย์ใดกด ในรูปที่ 8 เป็นการต่อแบบแมทริกซ์มีจำนวน 4 แถวและคอลัมน์ 8 คอลัมน์ ซึ่งได้จำนวนคีย์ทั้งสิ้น 32 คีย์

หลักการทำงานทั่วไปคือ เราจะทำการสแกนดู กล่าวคือ กำหนดให้แต่ละแถว (ซึ่งเป็นพอร์ตเอาต์พุต) เป็น "0" หรือ "1" ในเวลาต่างกัน เช่นแถวแรกเป็น "0" แถวอื่นเป็น "1" หมด แล้วทำการอ่านข้อมูลที่พอร์ตอินพุตดูว่า มีบิตใดบิตหนึ่งทางคอลัมน์เป็น "0" หรือไม่ ถ้ามีก็ทราบได้ว่าคีย์ในตำแหน่งแถวแรกคอลัมน์ที่เท่าไร เป็นคีย์ที่ได้รับการกด แต่ถ้าไม่มี CPU ก็จะสแกนไปยังแถว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 8 รูป(ก) การกำหนดพอร์ต B และ C ต่อกับคีย์บอร์ด 24 คีย์แบบแมทริกซ์ รูป(ข) แสดงการจัดวงจำแหน่งคีย์บอร์ดและรูป(ค) แสดงลักษณะการต่อคีย์บอร์ดแต่ละตัว

ถัดไปวนรอบไปเรื่อย ๆ ตลอดเวลา ดังนั้น การตรวจสอบคีย์กดจะทราบในลักษณะแถวที่เท่าไร และคอลัมน์ที่เท่าไรเป็นคีย์ที่ได้รับการกด

คราวนี้ลองพิจารณาวงจรในรูปที่ 8 ซึ่งต่อคีย์ในลักษณะ 4×8 เราจะสแกนทีละแถวโดยเริ่มจากแถว 0 โดยการให้เอาต์พุตที่บิต "1" เป็น "0" แล้ววนไปเรื่อย ๆ โดยใช้ค่าในรีจิสเตอร์ C เป็นค่ากำหนดแถว โปรแกรมที่เขียนขึ้นจะได้ดังโปรแกรมที่ 1

โปรแกรมที่กล่าวถึงโปรแกรมแรกนี้ เป็นการตรวจสอบคีย์ว่า ทุกคีย์ขณะนี้อยู่ในสภาพปล่อยหมดหรือไม่ โดยมีการเรียกโปรแกรมย่อย SCAN เพื่อส่งค่าสแกน-

โปรแกรมที่ 1 ส่วนตรวจเช็คการปล่อยคีย์

```
; RELEASE CHECK THAT KEY HAS BEEN RELEASE
RELEASE LD    C, 00H
        CALL  SCAN          ; เรียกโปรแกรมสแกนคีย์
        IN    A, 11H
        XOR   0FFH          ; ไม่กด = FFH
        JR    NZ, PRESS
        INC   C
        LD    A, 04H
        CP    C             ; ครบ 4 แถว?
        RET   C             ; ถ้าทุกคีย์ปล่อยหมด ให้ RET
        JR    RELEASE + 2
PRESS   CALL  DLY10*
        JR    RELEASE + 5
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คีย์ทีละแถวให้ตรวจสอบ โดยมีโปรแกรมย่อย คือ SCAN และหากถ้าพบว่ายังมีคีย์กดก็จะหน่วงเวลา 10 ms แล้วตรวจสอบใหม่ การใช้ RELEASE + 5 หมายถึง กระโดดไปยังตำแหน่ง RELEASE บวกอีก 5 ไบต์ และ $ + 4 หมายถึงกระโดดจากคำสั่งของตัวเองไปอีก 4 ไบต์

ในที่นี้โปรแกรมย่อยการสแกนคีย์จะเป็นดังโปรแกรมที่ 2

ส่วนโปรแกรมหน่วงเวลา 10 ms ซึ่งเราใช้โปรแกรมวนลูป จะมีลักษณะดังโปรแกรมที่ 3

ผลของโปรแกรมสแกน จะบอกสถานะการสแกนว่าสแกนมาถึง แถวใด ซึ่งจะอยู่ในหน่วยความจำ LINE นั่นก็คือเราจะทราบได้ว่า แถวใดกดโดยดูจาก LINE และจะตรวจสอบการกดคีย์ว่าเป็นคอลัมน์ใดได้ โดยทำโปรแกรมรับคีย์เข้าทางพอร์ตอินพุตและเก็บไว้ในรีจิสเตอร์ A ดังตัวอย่างในโปรแกรมที่ 4

จากกรณีนี้ จะได้ค่าคีย์ตามแนวคอลัมน์อยู่ในรีจิสเตอร์ A และแถวอยู่ในตัวแปร LINE ซึ่งเราจะนำไปใช้ในการถอดรหัสว่าเป็นคีย์อะไรได้ สังเกตว่าเรามีการตรวจสอบเพื่อ debounce คีย์ด้วย และมีการหน่วงเวลา 10 ms เพื่อป้องกันการอ่านค่าคีย์ผิดพลาดได้ และสิ่งที่สำคัญคือการทำงานของ CPU เร็วมาก ดังนั้นจึงต้องมีการตรวจสอบการปล่อยคีย์ (release) ก่อนแล้วจึงสแกนหาคอลัมน์ และแถว เพื่อกำหนดตำแหน่งคีย์

การทำงานของ 8255 ในโหมด 1

การทำงานของ 8255 ในโหมด 1 เป็นโหมดที่ทำให้อินพุต-เอาต์พุตมีการตรวจสอบสัญญาณ (handshaking) โดยใช้อินพุต-เอาต์พุตของพอร์ต A และ B เป็นหลัก และใช้พอร์ต C บนเป็นสัญญาณ handshake ของพอร์ต A ส่วนพอร์ต C ล่างเป็นสัญญาณ handshake ของพอร์ต B

โปรแกรมที่ 2 ส่วนการสแกนคีย์

```
; SCAN KEYBOARD BY OUTPUT TO ROW SCAN
SCAN    LD      HL, LINE
        LD      A, (HL)
        RLCA                    ; เลื่อนไปแถวถัดไป
        CP      10H             ; เกินแถวที่ 4?
        JR      NZ, $ + 4
        LD      A, 01H          ; เริ่มต้นแถวที่ 1 ใหม่
        LD      (HL), A
        OUT     10H, A
        OUT     12H, A
        RET
LINE    DB      01H
```

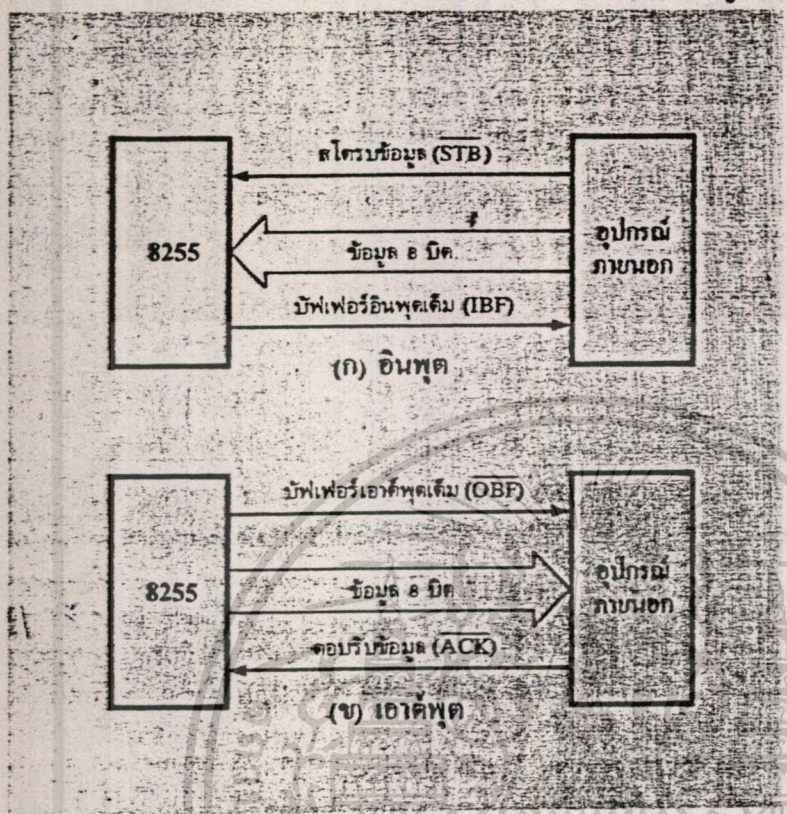
โปรแกรมที่ 3 ส่วนหน่วงเวลา 10 ms

```
; DELAY LOOP 10 MILLISECOND
DLY10   PUSH    DE              ; SAVE DE
        LD      DE, 1247 H      ; จำนวนลูป
        DEC     DE
        JR      NZ, $-1         ; LOOP ให้วน
        POP     DE
        RET
```

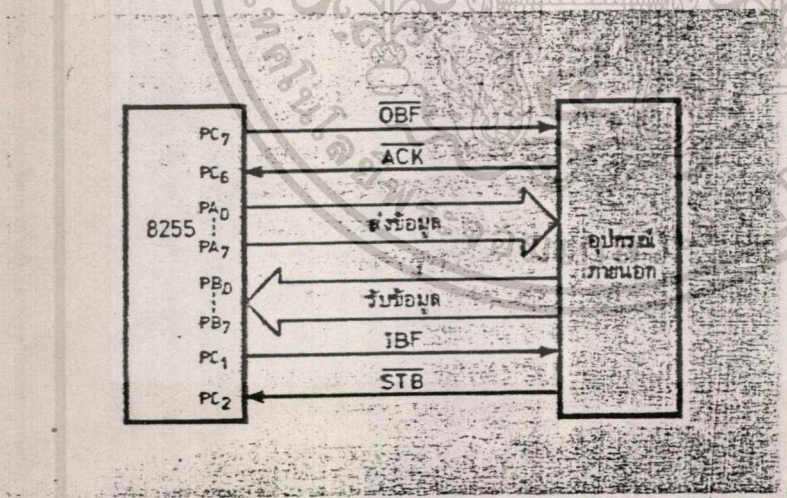
โปรแกรมที่ 4 ส่วนการรับคีย์และ debounce

```
; CHECK THAT KEY WAS PRESSED AND DEBOUNCE IT
KEYIN   CALL    SCAN
        IN      A, 11H
        XOR     0FFH
        JR      Z, KEYIN
DEBOUN  LD      B, A
        CALL    DLY10           ; หน่วงเวลา 10 ms
        IN      A, 11H          ; อ่านค่าคอลัมน์
        XOR     0FFH            ; กลับค่า
        CALL    Z, DLY10
        JR      Z, KEYIN + 3
        CP      B               ; ยังกดตัวเดิมอยู่?
        JR      NZ, DEBOUN      ; ให้วนอีกครั้ง
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9 โครงสร้างการจัดสัญญาณ handshake



รูปที่ 10 การต่อ 8255 ในโหมด 1

แนวความคิดของการใช้อินพุต-เอาต์พุตแบบ handshake ก็เพื่อให้มีการซิงโครไนซ์ระหว่างอุปกรณ์ภายนอกที่ทำงานได้ช้ากับการทำงานของคอมพิวเตอร์ที่ทำงานได้เร็ว เช่น เครื่องพิมพ์ทำงานได้ช้า เมื่อคอมพิวเตอร์ส่งตัวอักษรตัวแรกมาพิมพ์ เครื่องพิมพ์รับตัวอักษรและกำลังจะพิมพ์ คอมพิวเตอร์ก็ส่งตัวที่ 2 ตัวที่ 3 ตามมาทำให้การประมวลผลของอุปกรณ์เครื่องพิมพ์ทำงานไม่ทัน

ตารางที่ 2 หน้าที่ของขาต่าง ๆ ของพอร์ต C ในการทำงานเป็นสัญญาณ handshake เมื่อ 8255 ทำงานในโหมด 1

ขา	กรณีอินพุต	กรณีเอาต์พุต
PC_0	$INTR_B$	$INTR_B$
PC_1	IBF_B	$\overline{OBF}_B$
PC_2	$\overline{STB}_B$	$\overline{ACK}_B$
PC_3	$INTR_A$	$INTR_A$
PC_4	$\overline{STB}_A$	I/O
PC_5	IBF_A	I/O
PC_6	I/O	$\overline{ACK}_A$
PC_7	I/O	$\overline{OBF}_A$

อาจทำให้ข้อมูลสูญหาย ดังนั้น เครื่องพิมพ์จึงส่งสัญญาณบอกคอมพิวเตอร์ว่า อย่าเพิ่งส่งมาเพราะยังไม่พร้อมรับ

ลักษณะของการรับส่งข้อมูลอินพุต-เอาต์พุตแบบมี handshake แสดงได้ดังรูปที่ 9 ในกรณีนี้ เราจะใช้ PA_0-PA_7 เป็นเอาต์พุต และ PB_0-PB_7 เป็นอินพุต โดยมีพอร์ต C เป็นสัญญาณ handshake ดังแผนผังในรูปที่ 10

เมื่อโปรแกรม 8255 เป็นโหมด 1 แล้ว ตัว 8255 จะให้พอร์ต C เป็นสัญญาณควบคุมโดยแต่ละบิตของพอร์ต C เป็นไปตามที่กำหนดไว้แล้วดังตารางที่ 2

โดยปกติ 8255 ยังให้สัญญาณอินเตอร์รัพต์ไปบอก CPU ด้วย สัญญาณอินเตอร์รัพต์ของ 8255 จะเกิดขึ้นที่ขา PC_0 และ PC_3 โดยที่เมื่อบัฟเฟอร์พร้อมแล้วและต้องการให้ CPU ส่งอินพุตหรือเอาต์พุตมาที่บัฟเฟอร์ สัญญาณอินเตอร์รัพต์ก็จะเกิดขึ้น สังเกตว่า สัญญาณอินเตอร์รัพต์เป็นสัญญาณแอคทีฟลอจิก "1" ซึ่งตรงกับของ 8080 แต่เมื่อใช้กับ Z80 สัญญาณ $\overline{INT}$ ของ Z80 จะรับด้วยลอจิก "0"

โครงสร้างการ handshake ของ 8255 แสดงด้วยสัญญาณทางไฟฟ้าได้ดัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 11 สังเกตว่า การทำงานของ 8255 เกี่ยวกับสัญญาณ $\overline{RD}$ และ $\overline{WR}$ ซึ่งจะทำให้สัญญาณควบคุมเปลี่ยนแปลงไป

การตรวจสอบสัญญาณซึ่งกันและกันนี้ เป็นวิธีการรับส่งที่มีประสิทธิภาพ เช่นในกรณีอินพุต เมื่ออุปกรณ์ภายนอกต้องการส่งข้อมูลให้ CPU ก็จะส่งข้อมูลแบบขนานเข้ามา พร้อมทั้งสโตรบ ($\overline{STB}$) บอก 8255 ตัว 8255 จะนำข้อมูลนั้นไปเก็บไว้ในรีจิสเตอร์ภายในก่อนแล้วส่งสัญญาณตอบบอกว่า *บัฟเฟอร์ยังเต็มอยู่นะ (IBF) อย่าเพิ่งส่งมาอีก* ครั้นเมื่อ CPU อ่านข้อมูลจากรีจิสเตอร์ไปแล้ว ส่วนของสัญญาณบัฟเฟอร์อินพุต (IBF) ก็จะบอกว่า *ว่างแล้วส่งมาได้* อุปกรณ์ภายนอกก็จะส่งมาให้อีก

ทำนองเดียวกัน สำหรับพอร์ตเอาต์พุต เมื่อ CPU ส่งข้อมูลออกทางพอร์ตเอาต์พุตให้กับ 8255 ตัว 8255 ก็จะรับไว้ในรีจิสเตอร์ภายใน พร้อมทั้งส่งสัญญาณออกไปบอกอุปกรณ์ภายนอกว่า *เอาต์พุตบัฟเฟอร์ของฉันมีข้อมูลนะ ($\overline{OBF}$) มาอ่านเอาไปซิ* อุปกรณ์ภายนอกเมื่อทราบและพร้อมจะอ่านก็จะส่งสัญญาณตอบรับ ($\overline{ACK}$) พร้อมกับอ่านข้อมูลไป โดยสัญญาณ $\overline{ACK}$ จะมีความหมายว่า *ฉันอ่านข้อมูลของเธอไปแล้วนะ* ตัว 8255 ก็จะตอบกลับว่า *บัฟเฟอร์ฉันว่างแล้ว เธอรอก่อนนะ จะมีข้อมูลใหม่ส่งมาให้อีก*

ในการที่จะโปรแกรมโหมด 1 นี้นั้น เราจะใช้รหัสควบคุมเป็น 101(I/O)01(I/O)0 ในส่วน (I/O) หมายถึง ถ้าเป็นอินพุต ก็เป็น "1" เอาต์พุตก็เป็น "0" โดย I/O ตัวแรกเป็นของพอร์ต A ตัวที่ 2 คือ พอร์ต B เช่นถ้าต้องการให้พอร์ต A เป็นเอาต์พุต และพอร์ต B เป็นอินพุต เราจะใช้รหัสควบคุมเป็น 10100110 หรือ A6H

หากจะพิจารณาการทำงานของ CPU จะเห็นว่า ทำอย่างไรจึงจะเขียนหรืออ่านพอร์ตได้ถูกต้อง วิธีง่ายวิธีหนึ่งคือ CPU จะคอยตรวจสอบสัญญาณของ 8255 เช่นกรณีเอาต์พุต CPU จะคอยอ่านพอร์ต C แล้วตรวจสอบบิต 7 ($\overline{OBF}$) หลังจากที่ส่งข้อมูลไปแล้ว ถ้าบิต 7 ยังเป็น "0" แสดงว่ายังไม่ได้รับการสโตรบ แต่ถ้าเป็น "1" แล้ว แสดงว่าอุปกรณ์ภายนอกรับข้อมูลไปแล้ว สำหรับกรณีอินพุตก็คอยตรวจสอบจากสัญญาณ IBF ได้เช่นกันว่า มีข้อมูลใหม่เข้ามาหรือยัง คือ ตรวจสอบบิต PC_1 ของพอร์ต C

8255 ทำงานในโหมด 2

8255 ยังมีโหมดการทำงานอีกโหมดหนึ่งคือ โหมด 2 ซึ่งทำได้เฉพาะพอร์ต A ในโหมดนี้ 8255 จะใช้พอร์ต A ทำหน้าที่เป็นพอร์ตแบบสองทิศทาง คือ

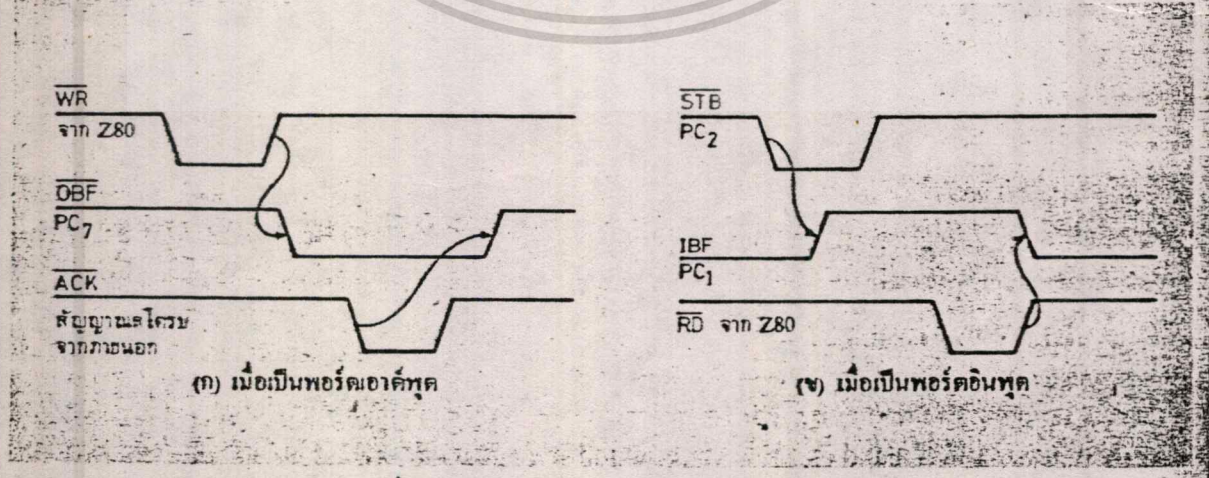
ตารางที่ 3 หน้าที่ของพอร์ต C ในโหมด 2

พอร์ต C	ความหมาย
PC_0	I/O
PC_1	I/O
PC_2	I/O
PC_3	$INTR_A$
PC_4	$\overline{STB}_A$
PC_5	IBF_A
PC_6	$\overline{ACK}_A$
PC_7	$\overline{OBF}_A$

สามารถเป็นได้ทั้งอินพุตและเอาต์พุต โดยโครงสร้างของพอร์ต A ทั้งอินพุตและเอาต์พุตมี handshake ทั้งคู่ ส่วนพอร์ต C ที่ใช้จะทำหน้าที่เป็นสัญญาณตรวจสอบ โดยมีสัญญาณแต่ละขาดังตารางที่ 3

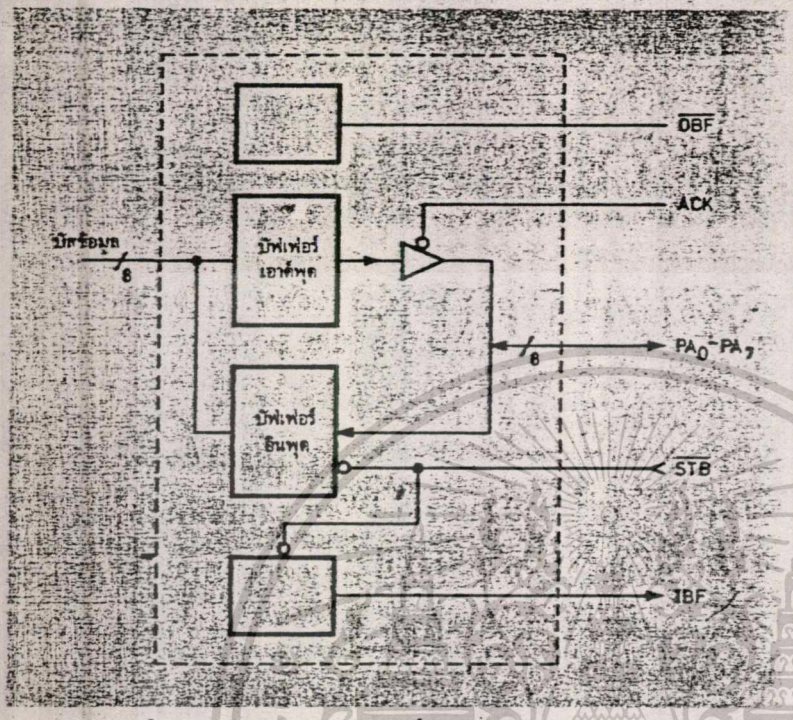
โครงสร้างของพอร์ต A ที่ทำงานแบบสองทิศทาง เป็นดังรูปที่ 12

สังเกตว่าเมื่อโปรแกรมพอร์ต A เป็นโหมด 2 แล้ว พอร์ต B จะต้องโปรแกรมเป็นโหมด 0 หรือโหมด 1 ก็ได้ ซึ่งก็ทำงานได้แยกอิสระอีก ในการใช้งานพอร์ตแบบ 2 ทิศทางนี้มีใช้ได้กับงานบางประเภทเช่น ใช้ในการรับส่งข้อมูลของพอร์ตมาตรฐานบางประเภทเช่น IEEE 488 หรือใช้เชื่อมโยงระหว่างคอม-



รูปที่ 11 แผนผังเวลาการรับ/ส่งข้อมูลแบบ handshake

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 12 โครงสร้างของพอร์ต A ที่ทำงานแบบพอร์ต 2 ทิศทาง

พิวเตอร์กับคอมพิวเตอร์ในการรับส่งข้อมูลสลับกันไปกลับ

ส่งท้าย

การใช้ 8255 เป็นการใช้ที่เราพบเห็นในวงจรไมโครโปรเซสเซอร์ทั่วไป เพราะใช้งานได้ง่ายและมีความคล่องตัว ผู้ออกแบบที่ต้องใช้พอร์ตขนานมักเลือกใช้ 8255 เป็นตัวอินเตอร์เฟสกับอุปกรณ์ภายนอก การทำความเข้าใจกับ 8255 จึงเป็นเรื่องที่ควรแก่การศึกษาอย่างยิ่ง

เอกสารอ้างอิงสำหรับบทความนี้

1. Intel "8085 reference manual" Intel 1980
2. Jame W. Coffron "Z80 application" Zybex Inc., 1983
3. คู่มือไอซีไมโครซัพพอร์ต และหน่วยความจำ จัดพิมพ์โดย บ.ซีเอ็ดยูเคชั่น จก.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Z8400
Z80® CPU Central Processing Unit

Zilog

Product Specification

April 1985

FEATURES

- The instruction set contains 158 instructions. The 78 instructions of the 8080A are included as a subset; 8080A software compatibility is maintained.
- Eight MHz, 6 MHz, 4 MHz, and 2.5 MHz clocks for the Z80H, Z80B, Z80A, and Z80 CPU result in rapid instruction execution with consequent high data throughput.
- The extensive instruction set includes string, bit, byte, and word operations. Block searches and block transfers, together with indexed and relative addressing, result in the most powerful data handling capabilities in the microcomputer industry.
- The Z80 microprocessors and associated family of peripheral controllers are linked by a vectored interrupt system. This system may be daisy-chained to allow implementation of a priority interrupt scheme. Little, if any, additional logic is required for daisy-chaining.
- Duplicate sets of both general-purpose and flag registers are provided, easing the design and operation of system software through single-context switching, background-foreground programming, and single-level interrupt processing. In addition, two 16-bit index registers facilitate program processing of tables and arrays.
- There are three modes of high speed interrupt processing: 8080 similar, non-Z80 peripheral device and Z80 Family peripheral with or without daisy chain.
- On-chip dynamic memory refresh counter.

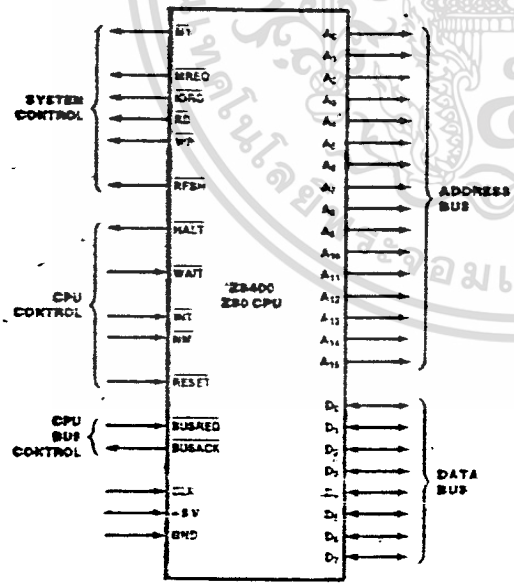


Fig. 1. Pin Functions

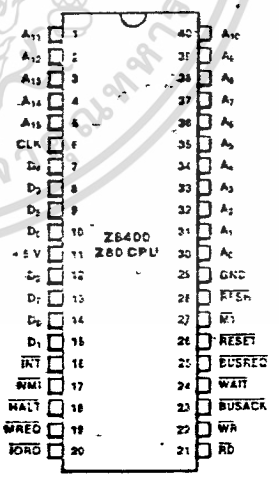


Figure 2a. 40-Pin Dual-In-Line Package (DIP)
Pin Assignments

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

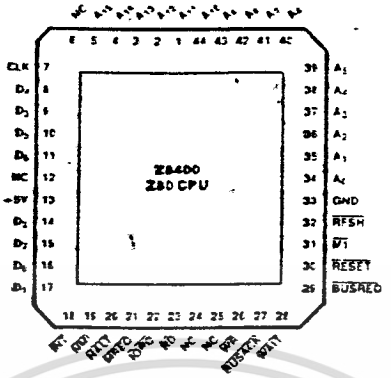


Figure 2b. 44-Pin Chip Carrier Pin Assignments

GENERAL DESCRIPTION

The Z80, Z80A, Z80B, and Z80H CPUs are third-generation single-chip microprocessors with exceptional computational power. They offer higher system throughput and more efficient memory utilization than comparable second- and third-generation microprocessors. The internal registers contain 208 bits of read/write memory that are accessible to the programmer. These registers include two sets of six general-purpose registers which may be used individually as either 8-bit registers or as 16-bit register pairs. In addition, there are two sets of accumulator and flag registers. A group of "Exchange" instructions makes either set of main or alternate registers accessible to the programmer. The alternate set allows operation in foreground-background mode or it may be reserved for very fast interrupt response.

The Z80 also contains a Stack Pointer, Program Counter, two index registers, a Refresh register (counter), and an Interrupt register. The CPU is easy to incorporate into a system since it requires only a single + 5V power source. All output signals are fully decoded and timed to control standard memory or peripheral circuits; the CPU is supported by an extensive family of peripheral controllers. The internal block diagram (Figure 3) shows the primary functions of the Z80 processors. Subsequent text provides more detail on the Z80 I/O controller family, registers, instruction set, interrupts and daisy chaining, and CPU timing.

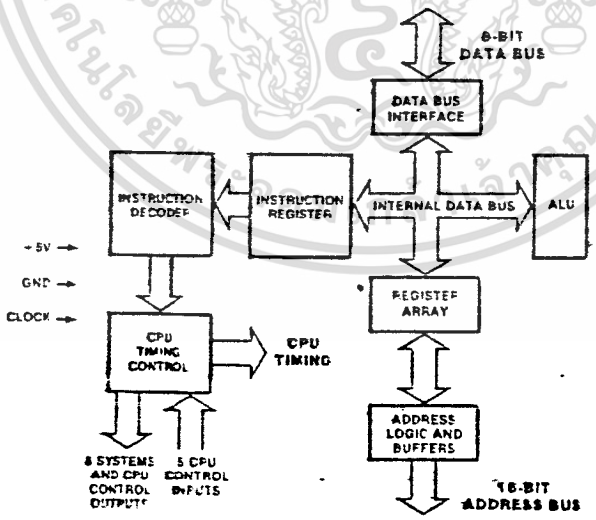


Figure 3. Z80 CPU Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Z80 MICROPROCESSOR FAMILY

The Zilog Z80 microprocessor is the central element of a comprehensive microprocessor product family. This family works together in most applications with minimum requirements for additional logic, facilitating the design of efficient and cost-effective microcomputer-based systems.

Zilog has designed five components to provide extensive support for the Z80 microprocessor. These are:

- The PIO (Parallel Input/Output) operates in both data-byte I/O transfer mode (with handshaking) and in bit mode (without handshaking). The PIO may be configured to interface with standard parallel peripheral devices such as printers, tape punches, and keyboards.
- The CTC (Counter/Timer Circuit) features four programmable 8-bit counter/timers, each of which has an 8-bit prescaler. Each of the four channels may be configured to operate in either counter or timer mode.
- The DMA (Direct Memory Access) controller provides dual port data transfer operations and the ability to terminate data transfer as a result of a pattern match.
- The SIO (Serial Input/Output) controller offers two channels. It is capable of operating in a variety of programmable modes for both synchronous and asynchronous communication, including Bi-Synch and SDLC.
- The DART (Dual Asynchronous Receiver/Transmitter) device provides low cost asynchronous serial communication. It has two channels and a full modem control interface.

Z80 CPU REGISTERS

Figure 4 shows three groups of registers within the Z80 CPU. The first group consists of duplicate sets of 8-bit registers: a principal set and an alternate set (designated by ' [prime], e.g., A'). Both sets consist of the Accumulator Register, the Flag Register, and six general-purpose registers. Transfer of data between these duplicate sets of registers is accomplished by use of "Exchange" instructions. The result is faster response to interrupts and easy, efficient implementation of such versatile programming techniques as background-foreground data processing. The second set of registers consists of six registers with assigned functions. These are the I (Interrupt Register), the R (Refresh Register), the IX and IY (Index Registers), the SP (Stack Pointer), and the PC (Program Counter). The third group consists of two interrupt status flip-flops, plus an additional pair of flip-flops which assists in identifying the interrupt mode at any particular time. Table 1 provides further information on these registers.

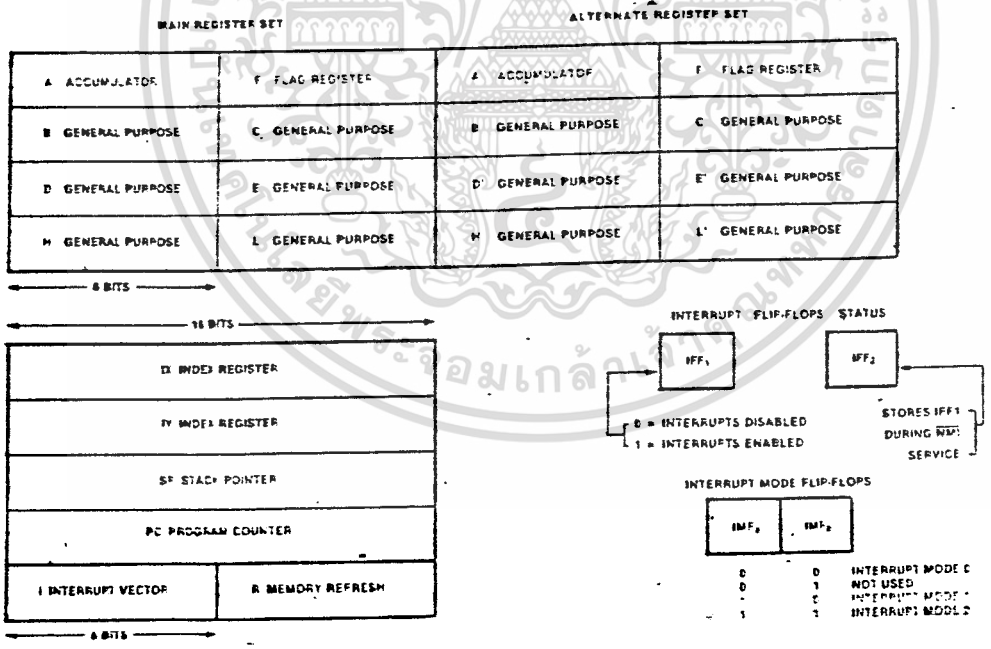


Figure 4. CPU Registers

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Z80 CPU REGISTERS (Continued)

Table 1. Z80 CPU Registers

Register		Size (Bits)	Remarks
A, A'	Accumulator	8	Stores an operand or the results of an operation.
F, F'	Flags	8	See Instruction Set.
B, B'	General Purpose	8	Can be used separately or as a 16-bit register with C.
C, C'	General Purpose	8	See B, above
D, D'	General Purpose	8	Can be used seperately or as a 16-bit register with E.
E, E'	General Purpose	8	See D, above.
H, H'	General Purpose	8	Can be used separately or as a 16-bit register with L.
L, L'	General Purpose	8	See H, above.
			Note: The (B,C), (D,E), and (H,L) sets are combined as follows: B — High byte C — Low byte D — High byte E — Low byte H — High byte L — Low byte
I	Interrupt Register	8	Stores upper eight bits of memory address for vectored interrupt processing.
R	Refresh Register	8	Provides user-transparent dynamic memory refresh. Automatically incremented and placed on the address bus during each instruction fetch cycle.
IX	Index Register	16	Used for indexed addressing.
IY	Index Register	16	Used for indexed addressing
SP	Stack Pointer	16	Holds address of the top of the stack. See Push or Pop in instruction set.
PC	Program Counter	16	Holds address of next instruction.
IFF_1-IFF_2	Interrupt Enable	Flip-Flops	Set or reset to indicate interrupt status (see Figure 4)
IM$_2$-IM$_0$	Interrupt Mode	Flip-Flops	Reflect interrupt mode (see Figure 4).

INTERRUPTS: GENERAL OPERATION

The CPU accepts two interrupt input signals: $\overline{NMI}$ and $\overline{INT}$. The $\overline{NMI}$ is a non-maskable interrupt and has the highest priority. $\overline{INT}$ is a lower priority interrupt and it requires that interrupts be enabled in software in order to operate. $\overline{INT}$ can be connected to multiple peripheral devices in a wired-OR configuration.

The Z80 has a single response mode for interrupt service for the non-maskable interrupt. The maskable interrupt, $\overline{INT}$, has three programmable response modes available. These are:

- Mode 0 — similar to the 8080 microprocessor.
- Mode 1 — Peripheral Interrupt service, for use with non-8080/Z80 systems.
- Mode 2 — a vectored interrupt scheme, usually daisy-chained, for use with Z80 Family and compatible peripheral devices.

The CPU services interrupts by sampling the $\overline{NMI}$ and $\overline{INT}$ signals at the rising edge of the last clock of an instruction. Further interrupt service processing depends upon the type of interrupt that was detected. Details on interrupt responses are shown in the CPU Timing Section.

Non-Maskable Interrupt ($\overline{NMI}$). The nonmaskable interrupt cannot be disabled by program control and therefore will be accepted at all times by the CPU. $\overline{NMI}$ is usually reserved for servicing only the highest priority type interrupts, such as that for orderly shutdown after power failure has been detected. After recognition of the $\overline{NMI}$ signal (providing $\overline{BUSREQ}$ is not active), the CPU jumps to restart location 0066H. Normally, software starting at this address contains the interrupt service routine.

Maskable Interrupt ($\overline{INT}$). Regardless of the interrupt mode set by the user, the Z80 response to a maskable interrupt input follows a common timing cycle. After the

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

interrupt has been detected by the CPU (provided that interrupts are enabled and $\overline{BUSREQ}$ is not active) a special interrupt processing cycle begins. This is a special fetch ($\overline{M1}$) cycle in which $\overline{IORQ}$ becomes active rather than $\overline{MREQ}$, as in a normal M1 cycle. In addition, this special M1 cycle is automatically extended by two WAIT states, to allow for the time required to acknowledge the interrupt request.

Mode 0 Interrupt Operation. This mode is similar to the 8080 microprocessor interrupt service procedures. The interrupting device places an instruction on the data bus. This is normally a Restart instruction, which will initiate a call to the selected one of eight restart locations in page zero of memory. Unlike the 8080, the Z80 CPU responds to the Call instruction with only one interrupt acknowledge cycle followed by two memory read cycles.

Mode 1 Interrupt Operation. Mode 1 operation is very similar to that for the $\overline{NMI}$. The principal difference is that the Mode 1 interrupt has only one restart location, 0038H.

Mode 2 Interrupt Operation. This interrupt mode has been designed to utilize most effectively the capabilities of the Z80 microprocessor and its associated peripheral family. The interrupting peripheral device selects the starting address of the interrupt service routine. It does this by placing an 8-bit vector on the data bus during the interrupt acknowledge cycle. The CPU forms a pointer using this byte as the lower 8 bits and the contents of the I register as the upper 8 bits. This points to an entry in a table of addresses for interrupt service routines. The CPU then jumps to the routine at that address. This flexibility in selecting the interrupt service routine address allows the peripheral device to use several different types of service routines. These routines may be located at any available location in memory. Since the interrupting device supplies the low-order byte of the 2-byte vector, bit 0 (A_0) must be a zero.

Interrupt Priority (Daisy Chaining and Nested Interrupts). The interrupt priority of each peripheral device is determined by its physical location within a daisy-chain configuration. Each device in the chain has an interrupt enable input line (IEI) and an interrupt enable output line (IEO), which is fed to the next lower priority device. The first device in the daisy chain has its IEI input hardwired to a High level. The first device has highest priority, while each succeeding device has a corresponding lower priority. This arrangement permits the CPU to select the highest priority interrupt from several simultaneously interrupting peripherals.

The interrupting device disables its IEO line to the next lower priority peripheral until it has been serviced. After servicing, its IEO line is raised, allowing lower priority peripherals to demand interrupt servicing.

The Z80 CPU will nest (queue) any pending interrupts or interrupts received while a selected peripheral is being serviced.

Interrupt Enable/Disable Operation. Two flip-flops, IFF_1 and IFF_2, referred to in the register description, are used to signal the CPU interrupt status. Operation of the two flip-flops is described in Table 2. For more details, refer to the *Z80 CPU Technical Manual* (03-0029-01) and *Z80 Assembly Language Programming Manual* (03-0002-01).

Table 2. State of Flip-Flops

Action	IFF_1	IFF_2	Comments
CPU Reset	0	0	Maskable interrupt INT disabled
DI instruction execution	0	0	Maskable interrupt INT disabled
EI instruction execution	1	1	Maskable interrupt INT enabled
LD A,I instruction execution	•	•	IFF_2 → Parity flag
LD A,R instruction execution	•	•	IFF_2 → Parity flag
Accept NMI	0	IFF_1	IFF_1 → IFF_2 (Maskable interrupt INT disabled)
RETN instruction execution	IFF_2	•	IFF_2 → IFF_1 at completion of an NMI service routine.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INSTRUCTION SET

The Z80 microprocessor has one of the most powerful and versatile instruction sets available in any 8-bit microprocessor. It includes such unique operations as a block move for fast, efficient data transfers within memory, or between memory and I/O. It also allows operations on any bit in any location in memory.

The following is a summary of the Z80 instruction set which shows the assembly language mnemonic, the operation, the flag status, and gives comments on each instruction. For an explanation of flag notations and symbols for mnemonic tables, see the Symbolic Notations section which follows these tables. The *Z80 CPU Technical Manual* (03-0029-01), the *Programmer's Reference Guide* (03-0012-03), and *Assembly Language Programming Manual* (03-0002-01) contain significantly more details for programming use.

The instructions are divided into the following categories:

- 8-bit loads
- 16-bit loads
- Exchanges, block transfers, and searches
- 8-bit arithmetic and logic operations
- General-purpose arithmetic and CPU control
- 16-bit arithmetic operations
- Rotates and shifts
- Bit set, reset, and test operations
- Jumps
- Calls, returns, and restarts
- Input and output operations

A variety of addressing modes are implemented to permit efficient and fast data transfer between various registers, memory locations, and input/output devices. These addressing modes include:

- Immediate
- Immediate extended
- Modified page zero
- Relative
- Extended
- Indexed
- Register
- Register indirect
- Implied
- Bit

8-BIT LOAD GROUP

Mnemonic	Symbolic Operation	Flags S	Z	H	P/V	N	C	Opcode 76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments	
LD r, r'	r ← r'	•	•	X	•	X	•	•	•	01	r	r'		1	1	4	r, r'	Reg.
LD r, n	r ← n	•	•	X	•	X	•	•	•	00	r	110		2	2	7	000	B
											← n →						001	C
LD r, (HL)	r ← (HL)	•	•	X	•	X	•	•	•	01	r	110		1	2	7	010	D
LD r, (IX + d)	r ← (IX + d)	•	•	X	•	X	•	•	•	11	011	101	DD	3	5	19	011	E
										01	r	110					100	H
											← d →						101	L
LD r, (IY + d)	r ← (IY + d)	•	•	X	•	X	•	•	•	11	111	101	FD	3	5	19	111	A
										01	r	110						
											← d →							
LD (HL), r	(HL) ← r	•	•	X	•	X	•	•	•	01	110	r		1	2	7		
LD (IX + d), r	(IX + d) ← r	•	•	X	•	X	•	•	•	11	011	101	DD	3	5	19		
										01	110	r						
											← d →							
LD (IY + d), r	(IY + d) ← r	•	•	X	•	X	•	•	•	11	111	101	FD	3	5	19		
										01	110	r						
											← d →							
LD (HL), n	(HL) ← n	•	•	X	•	X	•	•	•	00	110	110	36	2	3	10		
											← n →							
LD (IY + d), n	(IX + d) ← n	•	•	X	•	X	•	•	•	11	011	101	DD	4	5	19		
										00	110	110	36					
											← d →							
											← n →							

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8-BIT LOAD GROUP (Continued)

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD (IY+d), n	(IY+d)←n	•	•	X	X	•	•	11	111	101	FD	4	5	19	
								00	110	110	36				
									←d→						
									←n→						
LD A, (BC)	A←(BC)	•	•	X	X	•	•	00	001	010	0A	1	2	7	
LD A, (DE)	A←(DE)	•	•	X	X	•	•	00	011	010	1A	1	2	7	
LD A, (nn)	A←(nn)	•	•	X	X	•	•	00	111	010	3A	3	4	13	
									←n→						
									←n→						
LD (BC), A	(BC)←A	•	•	X	X	•	•	00	000	010	02	1	2	7	
LD (DE), A	(DE)←A	•	•	X	X	•	•	00	010	010	12	1	2	7	
LD (nn), A	(nn)←A	•	•	X	X	•	•	00	110	010	32	3	4	13	
									←n→						
									←n→						
LD A, I	A←I	‡	‡	X	0	X	IFF	0	•	11	101	101	ED	2	2	9	
										01	010	111	57				
LD A, R	A←R	‡	‡	X	0	X	IFF	0	•	11	101	101	ED	2	2	9	
										01	011	111	5F				
LD I, A	I←A	•	•	X	•	X	•	•	•	11	101	101	ED	2	2	9	
										01	000	111	47				
LD R, A	R←A	•	•	X	•	X	•	•	•	11	101	101	ED	2	2	9	
										01	001	111	4F				

NOTE: IFF, the content of the interrupt enable flip-flop, (IFF_2), is copied into the P/V flag

16-BIT LOAD GROUP

Mnemonic	Symbolic Operation	S	Z	H	P/V	N	C	76	543	210	Hex	No. of Bytes	No. of M Cycles	No. of T States	Comments
LD dd, nn	dd←nn	•	•	X	X	•	•	00	dd0	001		3	3	10	dd Pair
									←n→						00 BC
									←n→						01 DE
LD IX, nn	IX←nn	•	•	X	X	•	•	11	011	101	DD	4	4	14	10 HL
								00	100	001	21				11 SP
									←n→						
									←n→						
LD IY, nn	IY←nn	•	•	X	X	•	•	11	111	101	FD	4	4	14	
								00	100	001	21				
									←n→						
									←n→						
LD HL, (nn)	H←(nn+1)	•	•	X	X	•	•	00	101	010	2A	3	5	16	
	L←(nn)								←n→						
									←n→						
LD dd, (nn)	dd_H←(nn+1)	•	•	X	X	•	•	11	101	101	ED	4	6	20	
	dd_L←(nn)							01	dd1	011					
									←n→						
									←n→						

NOTE: $(PAIR)_H$, $(PAIR)_L$ refer to high order and low order eight bits of the register pair respectively e.g., BC_L = C, AF_H = A

Z80 CPU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

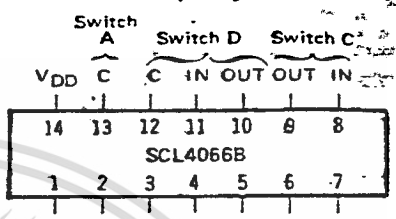
SCL4066B

CMOS QUAD ANALOG SWITCH

FEATURES

- Transmission or Multiplexing of Analog or Digital Signals
- 80Ω Typical ON-Resistance for 15-Volt operation
- Switch ON-Resistance Matched to within 5Ω over 15-Volt Signal-Input Range
- ON-Resistance Flat over Full Peak-to-Peak Signal Range
- High Degree of Linearity:
  ≤0.5% Distortion (typ) @ f_{is} = 1kHz,
  V_{is} = 5V_{p-p}, V_{DD}-V_{SS} ≥ 10V, R_L = 10kΩ
- Extremely Low OFF switch Leakage Resulting in very Low Offset Current and High Effective OFF Resistance:
  10pA (typ) @ V_{DD}-V_{SS} = 10V, T_A = 25°C
- Extremely High Control Input Impedance (Control Circuit Isolated from Signal Circuit):
  10^{12}Ω (typ)
- Low Crosstalk between Switches:
  -50dB (typ) @ f_{is} = 0.9MHz, R_L = 1kΩ
- Matched Control-Input to Signal-Output Capacitance Reduces Output Signal Transients
- Frequency Response, Switch ON = 40MHz (typ)

CONNECTION DIAGRAM
(all packages)



Add suffix for package:

C	14-pin Cerdip	F	14-pin Flat
D	14-pin Ceramic	H	Chip
E	14-pin Epoxy		

RECOMMENDED OPERATING CONDITIONS

For maximum reliability:

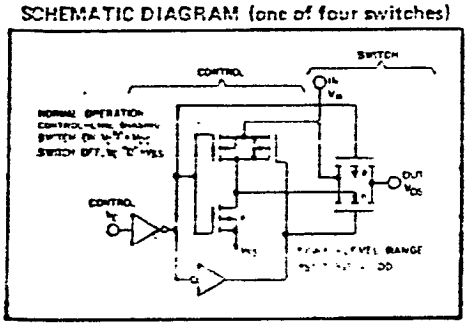
DC Supply Voltage	V_{DD} - V_{SS}	3 to 15	Vdc
Operating Temperature	T_A		
C, D, F, H Device		-55 to +125	°C
E Device		-40 to +85	°C

DESCRIPTION

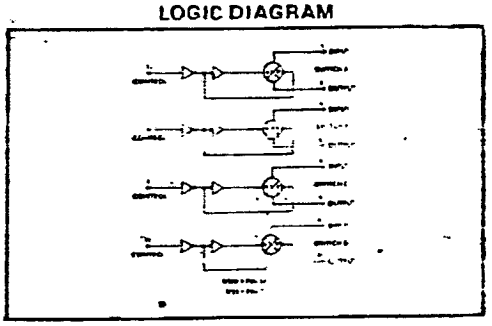
The SCL4066B is a Quad Bilateral Switch intended for the transmission or multiplexing of analog or digital signals. It is pin-for-pin compatible with the SCL4016B, but exhibits a much lower ON-resistance. In addition, the ON-resistance is relatively constant over the full input signal range. The SCL4066 consists of four independent bilateral switches. A single control signal is required per switch. Both the P and the N device in a given switch are biased ON or OFF simultaneously by the control signal. As shown below, the well of the N-channel device on each switch is either tied to the input when the switch is ON or to V_{SS} when the switch is OFF. This configuration minimizes the variation of the switch-transistor threshold voltage with input-signal, and thus keeps the ON-resistance low over the full operating range.

The advantages over single-channel switches include peak input-signal voltage swings equal to the full supply voltage, and more constant ON-impedance over the input-signal range. For sample-and-hold applications, however, the SCL4016 is recommended.

SCHEMATIC DIAGRAM (one of four switches)



LOGIC DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS

STATIC CHARACTERISTICS [1,3]

PARAMETER		CONDITIONS	V_{SS} (Vdc)	V_{DD} (Vdc)	T_{LOW} [2] Min.	T_{LOW} [2] Max.	25°C Min.	25°C Typ.	25°C Max.	T_{HIGH} [2] Min.	T_{HIGH} [2] Max.	Units
QUIESCENT DEVICE CURRENT	I_{DD}	$V_{IN} = V_{SS}$ or V_{DD}	0	5	–	0.05	–	0.0005	0.05	–	1.5	µAdc
		All valid input combinations	0	10	–	0.1	–	0.001	0.1	–	3.0	
			0	15	–	0.2	–	0.002	0.2	–	6.0	
MINIMUM INPUT HIGH VOLTAGE (Control Input)	V_{IH}	$V_{IS} = V_{SS}$	0	5	–	4.0	–	2.75	4.0	–	4.0	Vdc
		$V_{OS} = V_{DD}$	0	10	–	8.0	–	5.5	8.0	–	8.0	
		$I_{OS} = 10\mu A$	0	15	–	12.0	–	8.25	12.0	–	12.0	
MAXIMUM INPUT LOW VOLTAGE (Control Input)	V_{IL}	$V_{IS} = V_{SS}$	0	5	1.0	–	1.0	2.25	–	1.0	–	Vdc
		$V_{OS} = V_{DD}$	0	10	2.0	–	2.0	4.5	–	2.0	–	
		$I_{OS} = 10\mu A$	0	15	3.0	–	3.0	6.75	–	3.0	–	
SWITCH INPUT/OUTPUT LEAKAGE	I_{OFF}	$V_C = V_{SS}$ $V_{IS} = \pm 7.5$Vdc	-7.5	+7.5	–	±100	–	±0.01	±100	–	±200	nAdc
ON-RESISTANCE C,D,F,H device	R_{ON}	$V_C = V_{DD}$ $V_{SS} \leq V_{IS} \leq V_{DD}$ $R_L = 10k\Omega$	-7.5	+7.5	–	220	–	80	260	–	320	Ω
			0	+15								
			-5	+5	–	310	–	120	400	–	550	Ω
			0	+10								
			-2.5	+2.5	–	2000	–	270	2500	–	3500	Ω
			0	+5								
E device	R_{ON}	$V_C = V_{DD}$ $V_{SS} \leq V_{IS} \leq V_{DD}$ $R_L = 10k\Omega$	-7.5	+7.5	–	250	–	80	280	–	300	Ω
			0	+15								
			-5	+5	–	330	–	120	400	–	520	Ω
			0	+10								
			-2.5	+2.5	–	2100	–	270	2500	–	3200	Ω
			0	+5								
ON-RESISTANCE MATCH (Same package)	ΔR_{ON}	$V_C = V_{DD}$ $V_{SS} \leq V_{IS} \leq V_{DD}$ $R_L = 10k\Omega$	-7.5	+7.5	–	–	–	5	–	–	–	Ω
			0	+15								
			-5	+5	–	–	–	10	–	–		Ω
			0	+10								
			-2.5	+2.5	–	–	–	10	–	–		Ω
			0	+5								

NOTES:
1. Remaining Static Electrical Characteristics are listed under "SCL4000B Series Family Specifications"
2. T_{LOW} = -55°C for C, D, F, H device.
   = -40°C for E device.
   T_{HIGH} = +125°C for C, D, F, H device.
   = + 85°C for E device.
3. This device has been designed for balanced output drive current specifications. Consult Family Specifications.

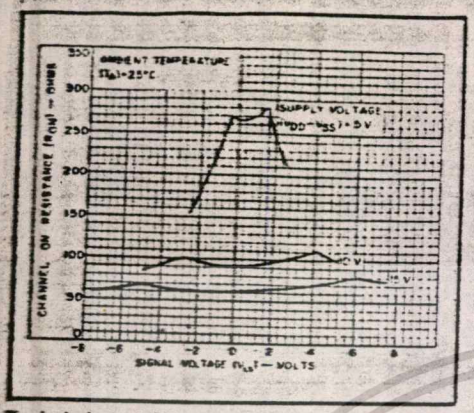
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS (Continued)

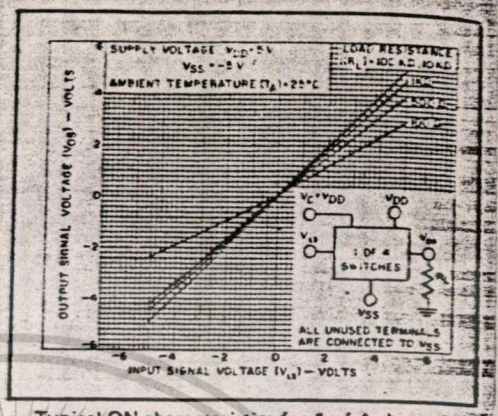
DYNAMIC CHARACTERISTICS (C_L = 50pF, T_A = 25°C)

PARAMETER		CONDITIONS	V_{SS} (Vdc)	V_{DD} (Vdc)	Min.	Typ.	Max.	Units
SIGNAL INPUTS (V_{is}) AND OUTPUTS (V_{os})								
PROPAGATION DELAY TIME Signal Input to Signal Output	t_{PLH}, t_{PHL}	V_c=V_{DD} V_{is}=Square Wave R_L = 10kΩ	0 0 0	5 10 15	– – –	20 10 7.5	40 20 15	ns
BANDWIDTH (-3dB) (Sine Wave)	BW	V_c=V_{DD} R_L V_{is}=5V_{p-p} 1kΩ centered 10kΩ @ 0.0Vdc 100kΩ 1MΩ	-5	+5	– – – –	54 40 38 37	– – – –	MHz
INSERTION LOSS $(=20\ \log_{10}\frac{V_{os}}{V_{is}})$		V_c=V_{DD} R_L V_{is}=5V_{p-p} 1kΩ centered 10kΩ @ 0.0Vdc 100kΩ 1MΩ	-5	+5	– – – –	2.3 0.2 0.1 0.05	– – – –	dB
SIGNAL DISTORTION (Sine Wave)		V_c=V_{DD} V_{is}=5V_{p-p} centered @ 0.0Vdc f_{is}=1.0kHz R_L = 10kΩ	-5	+5	–	0.16	–	%
FEEDTHROUGH (-50dB)		V_c=V_{SS} R_L V_{is}=5V_{p-p} 1kΩ centered 10kΩ @ 0.0Vdc 100kΩ 1MΩ	-5	+5	– – – –	1250 140 18 2	– – – –	kHz
CROSSTALK (-50dB) Between two switches		V_c(A)=V_{DD} V_c(B)=V_{SS} V_{is}(A)=5V_{p-p} centered @ 0.0Vdc R_L = 10kΩ	-5	+5	–	0.9	–	MHz
CAPACITANCE Input Output Feedthrough	C_{is} C_{os} C_{ios}	V_C = V_{SS}	-5	+5	– – –	8 8 0.5	– – –	pF pF pF
CONTROL INPUT (V_C)								
PROPAGATION DELAY TIME Turn on	t_{PC}	V_{SS}≤V_{is}≤V_{DD} R_L = 10kΩ	0 0 0	5 10 15	– – –	50 25 20	100 50 40	ns
MAXIMUM INPUT FREQUENCY	f_c	V_{SS}≤V_{is}≤V_{DD} R_L = 1.0kΩ	0 0 0	5 10 15	– – –	5 10 12	– – –	MHz
CROSSTALK (To signal port)		V_c = Square Wave R_L = 10kΩ R_L = 1.0kΩ	0 0 0	5 10 15	– – –	30 50 100	– – –	mV

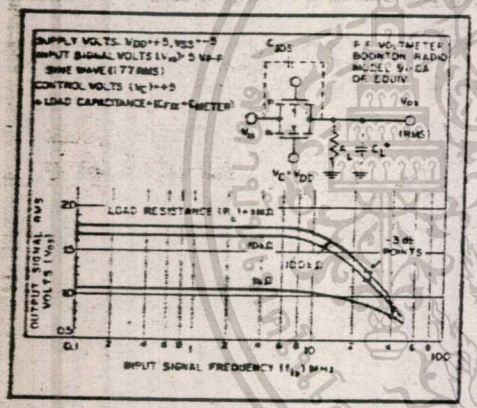
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



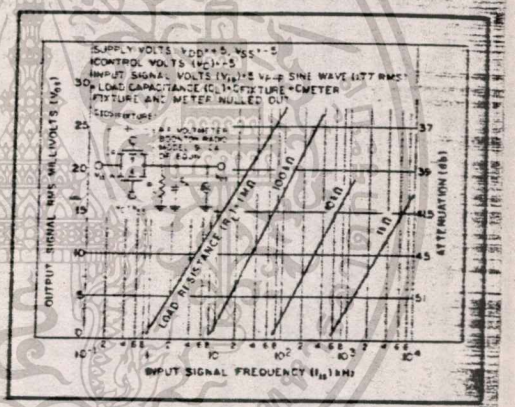
Typical channel ON resistance vs. signal voltage for three values of supply voltage (V_{DD}-V_{SS})



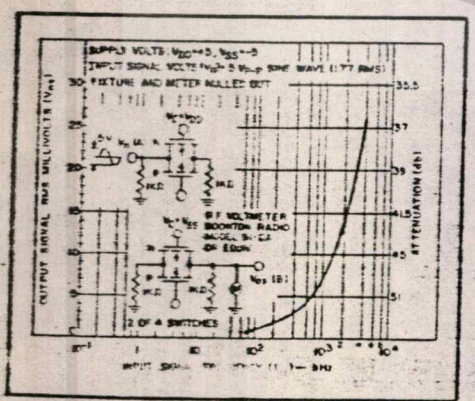
Typical ON characteristics for 1 of 4 channels.



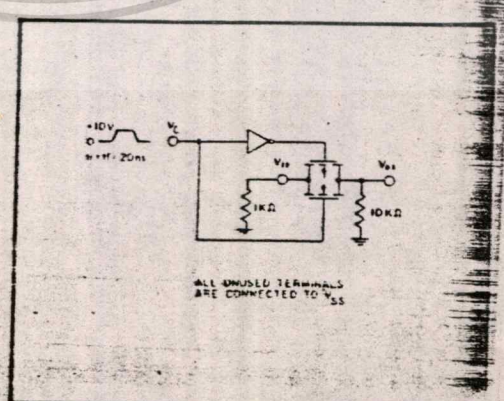
Typ. switch frequency response - switch "ON"



Typ. feedthru vs. freq. — switch "OFF"



Typ. crosstalk between switch circuits in the same package



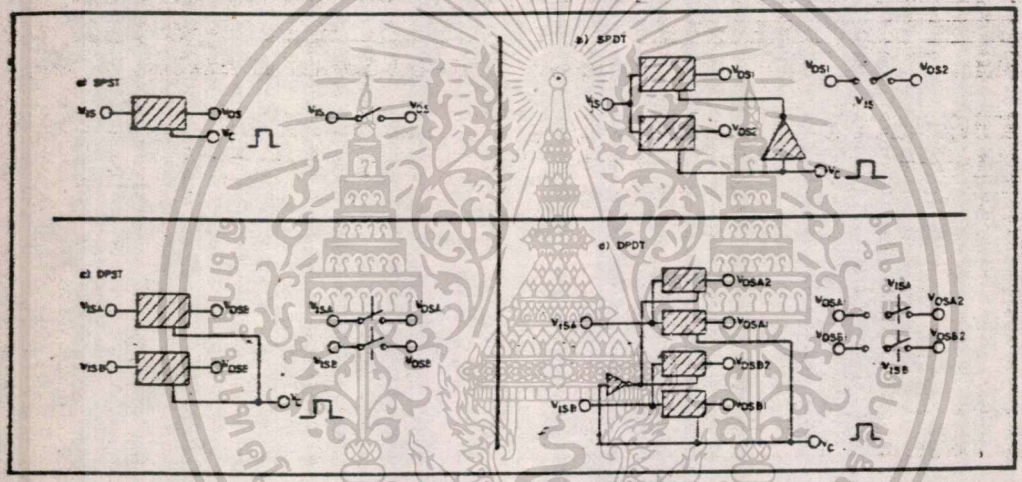
Test circuit, crosstalk-control input to signal output

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

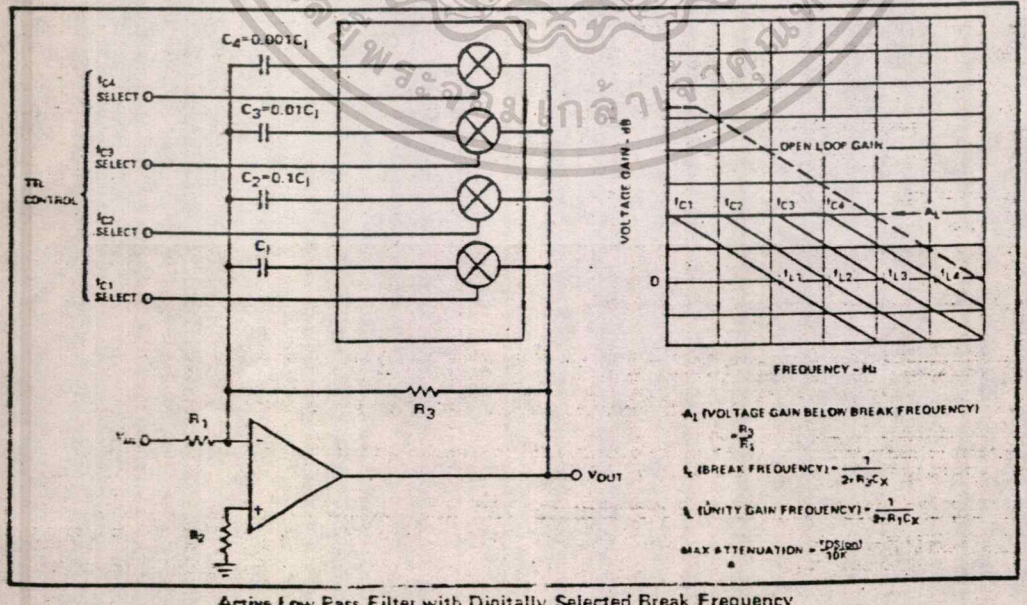
SPECIAL CONSIDERATIONS – SCL4066B

1. In applications where separate power sources are used to drive V_{DD} and the signal inputs, the V_{DD} current capability should exceed V_{DD}/R_L (R_L = effective external load of the 4 SCL4066B bilateral switches). This provision avoids any permanent current flow or clamp action on the V_{DD} supply when power is applied or removed from SCL4066B.
2. In certain applications, the external load-resistor current may include both V_{DD} and signal-line components. To avoid drawing V_{DD} current when switch current flows into terminals 1, 4, 8, or 11, the voltage drop across the bidirectional switch must not exceed 0.8 volt (calculated from R_{ON} values shown).
   No V_{DD} current will flow through R_L if the switch current flows into terminals 2, 3, 9, or 10. Failure to observe this condition may result in distortion of the signal.

APPLICATIONS INFORMATION

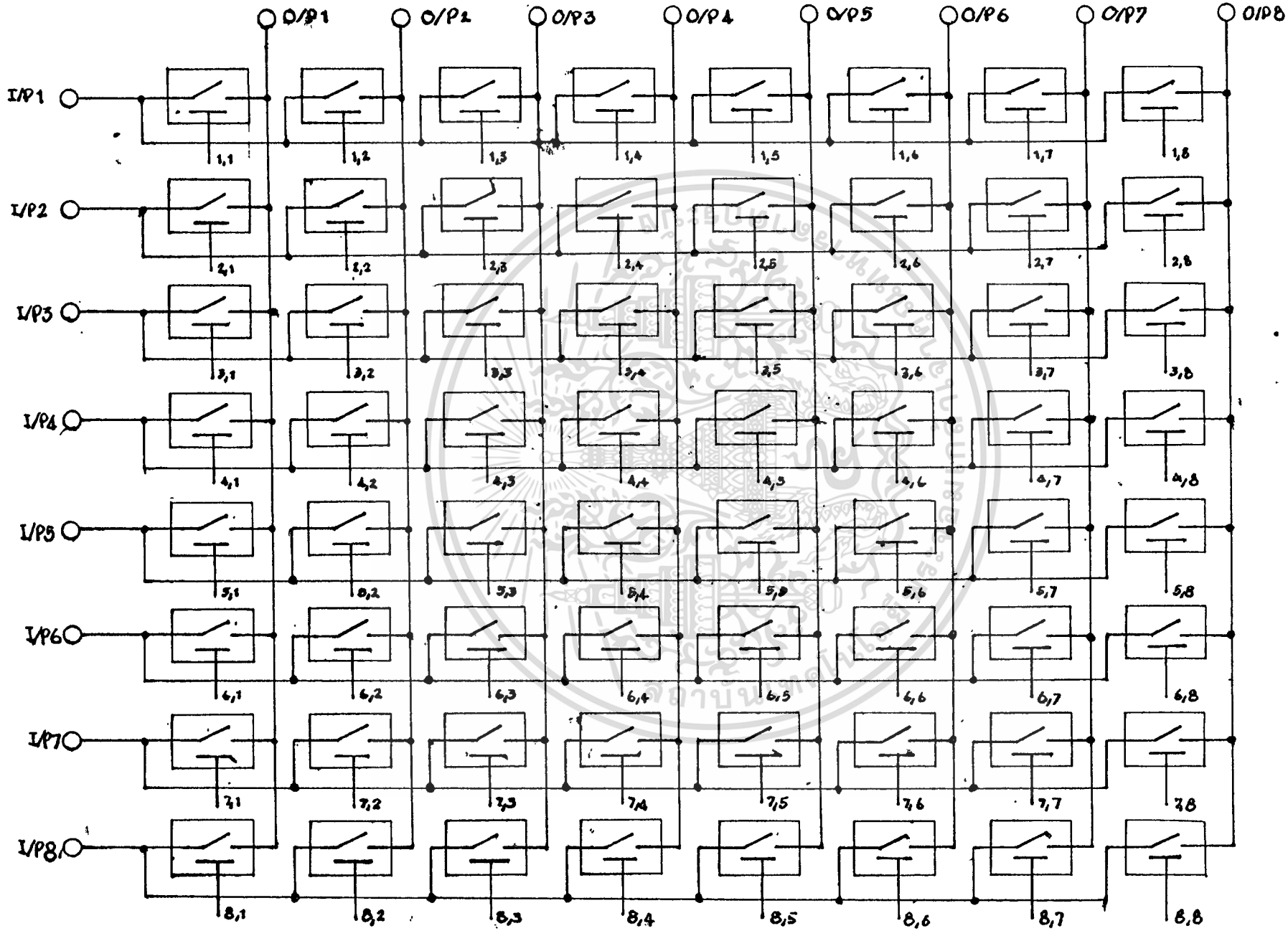


Basic Switch Functions using the SCL4066B



Active Low Pass Filter with Digitally Selected Break Frequency

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

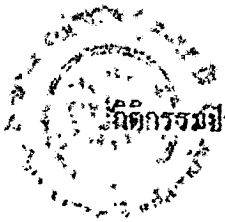


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

1. ผศ.ประทีป บรรณัดินพรัตน์ "การเขียนโปรแกรม ภาษาแอซเซมบลี Z-80" คณะวิศวกรรมศาสตร สถาบันเทคโนโลยีพระจอมเกล้า วิทยาเขตเจ้าคุณทหารลาดกระบัง พศ.2524
2. Ciracia S. "Build your own Z-80 Microcomputer" Peter Borough New hamshire Byte book McGrawhill 1981
3. "คู่มือไอซี CMOS 4000 Series" บริษัท ซีเอ็ดยูเคชั่น จำกัด พศ.2528
4. "คู่มือไมโครโปรเซสเซอร์" บริษัท ซีเอ็ดยูเคชั่น จำกัด พศ.2529
5. "คู่มือไอซี ชิพซัพพอร์ทและหน่วยความจำ" บริษัท ซีเอ็ดยูเคชั่น จำกัด พศ.2529
6. วัฒนา เชียงกูล,ยืน ภู่วรวรรณ "ไมโครโปรเซสเซอร์ ไมโครคอมพิวเตอร์" บริษัท ซีเอ็ดยูเคชั่น จำกัด พศ.2534

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



กิตติกรรมประกาศ

ขอขอบพระคุณคณาจารย์ทุกท่านในสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง ที่ประสิทธิประสาทวิชาความรู้แก่พวกกระผมให้ได้มีวิชาความรู้ความสามารถเพียงพอ ที่จะออกไปทำงานรับใช้สังคมต่อไป และขอขอบพระคุณบุพการี ผู้มีพระคุณส่งเสียให้ได้ศึกษาเล่าเรียนจนจบหลักสูตร และเพื่อน ๆ ที่ช่วยเหลือเอื้อเฟื้อในการทำปริญญานิพนธ์จนสำเร็จลุล่วงไปด้วยดี.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้