



ปีการศึกษา 2533

ปริญญานิพนธ์



นาย นิเมพันธ์ สุชะการผดุง 323312

นาย เท็ดวงศ์ แจ่มวิหี 323314

นาย สุวิน แสงพรหมมา 323335

อาจารย์ที่ปรึกษา

ปริญญานิพนธ์ปีการศึกษา 2533

ภาควิชา เทคโนโลยีอุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องคิดเงินค่าใช้บริการโทรศัพท์

ผู้จัดทำ

1. นายธนะพันธ์ สุขะการผดุง 323312
2. นายเทิดวงศ์ แจ่มรัมย์ 323314
3. นายสุวิน แพงพรหมมา 323335

.....อาจารย์ที่ปรึกษา

(อ.อรรถสิทธิ์ หล้าสกุล)

.....กรรมการ

(ผศ.ดร.กนก เจนจิระพงศ์เวช)

.....กรรมการ

(อ.กฤตากร กล่อมการ)

เครื่องคิดเงินค่าใช้บริการโทรศัพท์

นักศึกษา:

1. นายชินะพันธ์ สุทธการผดุง
2. นายเทิดวงศ์ แจ่มรัมย์
3. นายสุวิน พวงพรหมมา

อาจารย์ที่ปรึกษา:

อาจารย์ อรรถสิทธิ์ หล้าสกุล

ปีการศึกษา 2533

บทคัดย่อ

ปฏิญานพณธ์ฉบับนี้ได้นำเสนอโดยการนำไมโครโพรเซสเซอร์ (microprocessor) มาประยุกต์ใช้กับงานทางด้านเครื่องคิดเงินค่าใช้บริการโทรศัพท์ โดยพัฒนาเป็นเครื่องต้นแบบขึ้นมาโดยสามารถคิดเงินในการใช้โทรศัพท์ในแต่ละครั้ง ของการใช้โทรศัพท์ โดยความสามารถของเครื่องสามารถรับข้อมูลทางคีย์บอร์ด(key board) ขนาด 64 คีย์และ แสดงผลทางจอ LCD แบบ 2 บรรทัดละ 20 ตัวอักษร และมี Real Time Clock ในการคิดเวลาของการใช้โทรศัพท์ในแต่ละครั้ง และ สำหรับส่วนสุดท้ายซึ่งมีความสำคัญส่วนหนึ่งคือ Detect Signals ซึ่งจะทำงานเมื่อทางด้านผู้รับยกหูโทรศัพท์เท่านั้น ซึ่ง มีความเกี่ยวข้องตรงมากในด้าน การคิดเงิน เพราะเป็นเวลาจริงๆ ของการใช้โทรศัพท์เท่านั้น

CALCULATOR AUTOMATIC TELEPHON SERVICES

STUDENT:

MR.CHINAPHAN SUKAKARNPADOONG

MR.TARDWONG JAMRUSSAMEE

MR.SUWIN PANGPROMMA

ADVISER:

MR.AUTHASIT LASAKUL

ABSTRACT

THIS PROJECT IS PRESENT BY APPLIED MICROPROCESSOR FOR CALCULATOR AUTOMATIC TELEPHON SERVICES. WHICH IS DEVELOP FOR ORIGINAL. IT CAN BE CALCULATE ANOTHER PERSON USE TELEPHON ANY TIMES. PROPERTY OF IMPLEMENT SUCH AS RECEIVES DATA FROM KEYBOARD 64 KEYS TOUCH ,DISPLAY BY LCD MODULE 20 CHARACTORS 2 LINES, REALTIME CLOCK FOR CALCULATE PERIOD OF TIMES USED TELEPHON AND FINAL PART IT IMPORTANT FOR THIS PROJECT IS DETECT SIGNALS WHICH IS WORK WHEN RECEIVER HAND SET ONLY.

สารบัญ

บทที่ 1.	บทนำ	1
บทที่ 2.	ทฤษฎีและหลักการเครื่องคิดเงินค่าใช้บริการโทรศัพท์	2
2.1	คุณสมบัติของเครื่องคิดเงินค่าใช้บริการโทรศัพท์	2
2.2	จุดประสงค์และการนำไปใช้	2
2.3	สิ่งแวดล้อมภายในชุมสายโทรศัพท์ขององค์การโทรศัพท์แห่งประเทศไทย	3
2.4	โครงสร้างของเครื่องคิดเงินค่าใช้บริการโทรศัพท์	4
2.5	ไมโครโปรเซสเซอร์ Z-80	6
2.6	โครงสร้างภายในของไมโครโปรเซสเซอร์ Z-80	7
บทที่ 3.	หลักการสร้าง	11
3.1	Z-80 กับ การถอดรหัสหน่วยความจำ	12
3.2	Z-80 กับ อุปกรณ์อินพุต/เอาต์พุต	16
3.3	ลักษณะของวงจร	18
3.4	Z-80 กับ 8279 Keyboard/Display	19
3.5	Z-80 กับ Real Time	26
3.6	Z-80 กับ 8255	41
3.7	Dot Matric LCD Display	46
3.8	การเขียนข้อมูล	47
3.9	MT 8870 Integrated DTMF Receiver	49
3.10	หลักการของวงจรดีเทคเตอร์สัญญาณโทรศัพท์	55

บทที่4 โปรแกรมควบคุมระบบและการทำงานของเครื่องคิดเงินค่าใช้บริการโทรศัพท์	61
4.1 โปรแกรมควบคุมการทำงานแสดงผลบนจอ LCD	61
4.2 โปรแกรมควบคุมการรับข้อมูลทาง Key board	61
4.3 โปรแกรมควบคุม Realtime clock	61
4.4 โปรแกรมควบคุมการใช้คการยกหูทางด้านผู้รับและคิดเงิน	61

บทที่5 บทสรุป	62
---------------	----

ภาคผนวก

ภาคผนวก ก. โปรแกรม Monitor

ภาคผนวก ข. วงจรทั้งหมด

ภาคผนวก ค. Data sheet Z-80, LCD, 8279

บทที่ 1

บทนำ

(INTRODUCTION)

การติดต่อสื่อสารในปัจจุบันนี้ จะเห็นได้ว่าโทรศัพท์ มีความจำเป็นมากสำหรับวงการธุรกิจ และ เมื่อในวงการธุรกิจมีความต้องการโทรศัพท์เป็นจำนวนมาก ก็เกิดการขาดแคลนคู่สายโทรศัพท์ขึ้น การขาดแคลนนี้เกิดจากระดับเงินกู้ ขององค์การโทรศัพท์ชนพีดาน ทำให้ องค์การโทรศัพท์ มีอาจจะกู้เงินมาเพิ่มวงจรได้อีก ก็ต้องอาศัยเอกชนมาลงทุนแทน แต่ทางแก้ปัญหาทางนี้ก็ไม่ได้ตอบสนองความต้องการได้ทันความต้องการ กราบไต่ที่ตึกสำนักงาน ใหญ่ๆ ยังใช้โทรศัพท์อย่างฟุ่มเฟือย และใช้งานได้ไม่เต็มความสามารถ

ด้วยความก้าวหน้าทางเทคโนโลยี ผู้จัดทำได้เล็งเห็นความสำคัญที่จุดนี้ก็เลยร่วมกันพัฒนาเครื่องต้นแบบของ เครื่องคิดเงินค่าใช้บริการโทรศัพท์ แต่เครื่องคิดเงินค่าใช้บริการโทรศัพท์นี้เป็นเพียงส่วนประกอบอย่างหนึ่งของอาคารสำนักงาน ใช้โทรศัพท์ให้มีประสิทธิภาพมากที่สุดเท่านั้น

สำหรับปริญญานิพนธ์ฉบับนี้ ทุกส่วนทุกระบบได้ได้ออกแบบขึ้นมาเองทั้งหมด เหตุผลที่มีได้ ำใช้งานบนเครื่องไมโครคอมพิวเตอร์ ซึ่งง่ายต่อการพัฒนานั้น เพราะเมื่อสร้างขึ้นมาแล้วจะมองให้อยู่ในการผลิตระดับอุตสาหกรรมค่อนข้างยาก เนื่องจากการอ้างอิงทางด้านตลาดก็ต้องอาศัยตลาดคอมพิวเตอร์ด้วย ซึ่งความล้ำสมัยอย่างรวดเร็ว ของเครื่องคอมพิวเตอร์ ราคาค่อนข้างแพงและห้องเก็บคอมพิวเตอร์ที่ต้องทำขึ้นโดยเฉพาะ ซึ่งทำให้สิ้นเปลือง มากกว่าการพัฒนาทั้งระบบ

โดยทางด้านเนื้อหาได้แยกออกเป็นหัวข้อ ตั้งแต่จุดเริ่มและแนวความคิดจนถึงการสร้าง พร้อมทั้งการเขียนโปรแกรมควบคุม และได้เพิ่มรายละเอียดของอุปกรณ์ที่ใช้ พร้อมทั้งการใช้งานของอุปกรณ์แต่ละตัวรวมอยู่ในภาคผนวกด้วย เพื่อสำหรับผู้ที่ต้องการพัฒนาต่อไป

บทที่ 2

ทฤษฎี และ หลักการเครื่องคิดเงินค่าใช้บริการโทรศัพท์

2.1 คุณสมบัติของเครื่องคิดเงินค่าใช้บริการโทรศัพท์

2.1.1 สามารถคิดเงินค่าบริการโทรศัพท์ภายในเขตและ ภายนอกเขตทั้งต่างจังหวัด และต่างประเทศได้ และ สามารถแยกคิดเงินตามชนิดของการใช้โทรศัพท์ ถ้าเป็นโทรศัพท์ ภายในเขต ก็คิดเงินเฉพาะต่อครั้งของการใช้โทรศัพท์ สำหรับในปัจจุบันครั้งละ 3 บาท และอาจจะรวมค่า บริการตามแต่เจ้าของกิจการจะพิจารณา ส่วนการโทรศัพท์ติดต่อภายนอก เขตต้องนำเวลามาคิดด้วยโดยให้ความละเอียดเป็นวินาทีและมีช่วงเวลา 3 ช่วงราคาแตกต่างกัน

2.1.2 การคิดเงินสามารถแสดงให้ทราบราคาได้ทั้งหน้าจอ Display และโดย Printer ซึ่งเป็น Printer ขนาดเล็ก

2.1.3 การใช้งานส่วนใหญ่จะเน้นทางด้าน function key เพราะง่ายแก่การใช้งาน

2.2 จุดประสงค์และการนำไปใช้งาน

2.2.1 รับรองการเจริญเติบโตของเศรษฐกิจ ใช้ภายในอาคารที่มีบริษัทหลายบริษัทใน อาคารเดียวกันให้มีความคุ้มค่าของคู่สายโทรศัพท์มากขึ้น คือสมมติอาคาร ก. มีบริษัทอยู่ทั้งหมด 5 บริษัทใน 5 บริษัทนี้ โดยปรกติแล้วจะมีคู่สายโทรศัพท์ใช้อย่างน้อย 1 คู่สาย ฉะนั้นใน อาคาร ก. นี้จะมีคู่สายโทรศัพท์เข้ามา 5 คู่สาย โดย 1 บริษัทใช้ได้เพียง 1 คู่สายเท่านั้น แต่ถ้าอาคารมี เครื่องคิดเงินค่าใช้บริการโทรศัพท์ โดยที่อาคารนี้มี 5 คู่สายเช่นกัน แต่ 1 บริษัทสามารถใช้คู่สายได้ถึง 5 คู่สาย โดยแต่ละบริษัทจะไม่มีปัญหาทางด้านการคิดเงินค่า ใช้โทรศัพท์เลย เพราะเครื่องคิดเงินนี้ช่วยคุณได้

2.2.2 ใช้งานด้านกิจการคอนโดมิเนียม อพาร์ทเมนต์(Apartment) การบริการที่ คล่องตัวให้กับ กิจการโรงแรม กิจการด้านอาคารชุดนี้จะมีห้องพักจำนวนมาก ในหนึ่งห้องพักจะมีโทรศัพท์ใช้ 1 เครื่อง ผู้เช่าอาศัยก็ใช้โทรศัพท์โทรหาไปแต่ละที่แต่ละเวลา อาจจะเป็นใน เขต นอกเขตหรือต่างประเทศ ปัญหาที่เกิดขึ้นกับผู้ประกอบการด้านนี้คือ เมื่อใบเสร็จโทรศัพท์ เอกสารนี้เป็นเอกสารที่ส่งวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประเภททางไกลมาก็ ต้องมานั่งจัดสรรว่า ลำดับการโทรศัพท์ของใบเสร็จโทรศัพท์นั้นเป็นของ ห้างไหนบ้าง บางทีผู้เช่าอาจจะออกไปก่อนที่จะใบเสร็จโทรศัพท์จะมา ทำให้เกิดความลำบากใจ กับผู้ประกอบการด้านนี้มาก แต่ถ้าผู้ประกอบการด้านนี้มีเครื่องคิดเงินค่าใช้โทรศัพท์ใช้ก็เพียงพอ แต่กดปุ่มไม่ก็ปุ่มก็ได้จำนวนเงิน ที่ถูกต้องได้ และบางทีอาจจะมีการรื้อไหลของเงินค่าโทรศัพท์ ได้ถ้าหาก พนักงานต่อสายโทรศัพท์รู้เห็นกับผู้เช่าโดย โทรศัพท์ฟรีก็สามารถทำได้ง่ายมาก แต่ ถ้ามีเครื่องนี้ก็ไม่สามารถที่โกงค่าบริการโทรศัพท์กับ ผู้ประกอบการด้านนี้ได้

2.3 สัญญาณภายในชุมสายโทรศัพท์ขององค์การโทรศัพท์แห่งประเทศไทย

สำหรับโครงงานนี้จำเป็นต้องอาศัย สัญญาณต่างๆ จากองค์การโทรศัพท์แห่งประเทศไทยประกอบในการทำงานของ เครื่องคิดเงินค่าใช้บริการโทรศัพท์โดยมีสัญญาณที่เกี่ยวข้องกับ โครงงานนี้ดังนี้

2.3.1 สัญญาณ Dual tone (DT) ใช้เพื่อแสดงให้ผู้ใช้โทรศัพท์ทราบว่าเริ่มต้นกด หมายเลขโทรศัพท์ของคู่สายที่ต้องการจะสนทนาด้วย ซึ่งสัญญาณนี้จะมีลักษณะ เป็นสัญญาณเสียง มีความถี่ต่อเนื่องขนาด 400 Hz มอดูเลชั่นแบบ AM กับความถี่ 50 Hz

2.3.2 สัญญาณ Busy tone (BT) ใช้เพื่อแสดงให้ผู้ใช้โทรศัพท์ทราบว่าคู่สายที่ต้องการจะสนทนาด้วยนั้นไม่ว่าง ต้องวางหูใหม่สักครู่ แล้วจึงจะยกหูโทรศัพท์ใหม่ ซึ่งเป็นสัญญาณ เสียงขนาด 400 Hz โดยมีช่วงเวลาของการส่งประมาณ 0.5 วินาที เงียบประมาณ 0.5 วินาที

2.3.3 สัญญาณ Ring back tone (RBT) ใช้เพื่อแสดงให้ผู้ใช้โทรศัพท์ทราบว่า เครื่องชุมสายโทรศัพท์ดำเนินการต่อโทรศัพท์ได้สำเร็จแล้ว ให้รอคู่สนทนาไว้ ซึ่งเป็นสัญญาณ เสียงขนาด 400 Hz โดยมีช่วงเวลาของการส่งประมาณ 1 วินาที เงียบประมาณ 4 วินาที

2.3.4 สัญญาณ Ringing tone (RGT) ใช้เพื่อแสดงเสียงกริ่ง ให้ทราบว่าผู้คนที่ ต้องการจะสนทนาด้วย ซึ่งเป็นสัญญาณเสียงขนาด 25 Hz โดยมีช่วงเวลาการส่งประมาณ 1 วินาที เงียบประมาณ 4 วินาที ซึ่งเหมือนกับสัญญาณ Ring back tone

2.4 โครงสร้างของเครื่องคิดเงินค่าใช้บริการโทรศัพท์

หลักการทำงานของเครื่องคิดเงินค่าใช้บริการโทรศัพท์สามารถแบ่งการทำงานออกเป็น ส่วนได้ดังนี้

1. Main board Z-80

2. Key board 8279

3. Timer HD146818

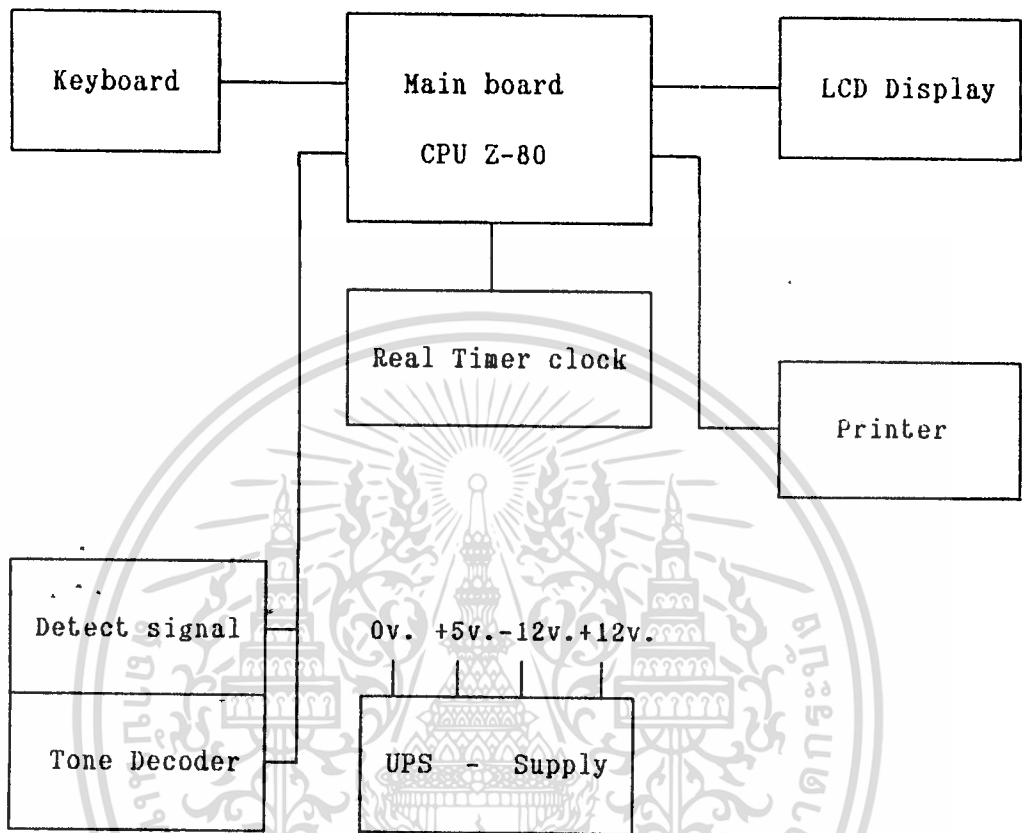
4. Detect Signal

5. LCD Display

6. Tone decoder MT8870

7. Printer

8. Supply



รูปที่ ๑. Block Diagram

โครงการนี้ใช้กับชุดสายโทรศัพท์ระบบ DTMF เมื่อผู้ใช้โทรศัพท์ต้องการที่จะใช้โทรศัพท์เพื่อที่จะติดต่อกับคู่สายอื่น โดยโครงการนี้มีการทำงานดังนี้

2.4.1 Detect Signal (สัญญาณการยกหูด้านต้นทาง)

2.4.2 รอรับการถอดรหัสเป็นรหัส BCD จาก Tone Decoder เพื่อให้ CPU เก็บไว้และคำนวณค่าบริการตามประเภท และ ระยะเวลาของการใช้โทรศัพท์ ถ้าไม่มีการใช้โทรศัพท์ (เนื่องจากไม่มีคนรับสาย) ก็จะยกเลิกตัวเลขที่ Tone Decoder ส่งมาให้ทันที

2.4.3 ตรวจสอบสัญญาณ

2.4.3.1 ถ้าสัญญาณต่อไปเป็นสัญญาณ Busy Tone ก็จะทำการ Reset ระบบ ยกเลิกตัวเลขที่ Tone Decoder ส่งมาให้ทันทีเช่นกัน เพื่อรอการทำงานเริ่มต้นใหม่ เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3.2 ถ้าสัญญาณต่อไปเป็นสัญญาณ Ring Back Tone ต้องรอนจนกว่า Ring Back Tone หายไปก็จะเริ่มจับเวลาเพื่อใช้คิดค่าบริการ

2.4.4 เมื่อมีการวางหูโทรศัพท์ ก็จะเป็นการกำหนดเวลาสิ้นสุดการใช้โทรศัพท์ที่จะได้ระยะเวลาการใช้โทรศัพท์

2.4.5 ในส่วนของโปรแกรมก็จะทำการคำนวณค่าใช้บริการ และเก็บเบอร์โทรศัพท์ปลายทาง ที่โทรศัพท์ติดต่อด้วย เพื่อที่จะใช้แสดงให้ทราบถึง การใช้โดยจะเก็บไว้ในหน่วยความจำเป็นส่วนๆในส่วนของ Display, Key board, printer มีลักษณะการทำงานดังนี้ ใช้ Key board ในการป้อนชื่อของผู้ใช้บริการในแต่ละส่วนของผู้ใช้ทุกๆ และมี Key Function ใช้ในการแสดงยอดเงินค่าบริการ หรือต้องการพิมพ์ออกทาง Printer เพื่อใช้เป็นใบเสร็จเงิน การเก็บค่าโทรศัพท์จากผู้เช่าในแต่ละส่วน

2.5 ไมโครโปรเซสเซอร์ Z-80

หลังจากหมดยุคไมโครคอมพิวเตอร์ขนาด 8 บิต แล้วนั้นในส่วนของไมโครโปรเซสเซอร์ขนาด 8 บิต ก็มีได้หมดยุคตามไปด้วย แต่จะมีความนิยมนำมาประยุกต์ใช้กับลักษณะงานที่เฉพาะกิจมากขึ้น ราคาของไมโครโปรเซสเซอร์ Z-80 นี้ก็ถูกลงมากและสามารถใช้ร่วมกับอุปกรณ์สนับสนุน (Peripheral) ได้มากมายการพัฒนาระบบก็เป็นไปได้โดยสะดวกเช่น การนำ Z-80 ไปประยุกต์ใช้กับงานในชุมสายโทรศัพท์แต่ในที่นี้จะนำ Z-80 ไปประยุกต์ใช้กับเครื่องคิดเงินค่าบริการโทรศัพท์ ฉะนั้นหนังสือเล่มนี้จะกล่าวถึงลักษณะการนำ Z-80 มาใช้กับงานเครื่องคิดเงินค่าบริการโทรศัพท์เท่านั้น โดยจะนำเสนอในลักษณะของการอินเตอร์เฟส (Interface) กับอุปกรณ์สนับสนุนทีละส่วน

2.5.1 เหตุผลที่ใช้ไมโครโปรเซสเซอร์เบอร์ Z-80 เพราะมีลักษณะที่น่าสนใจดังนี้

2.5.1.1 หือจำกัดในพัฒนาทางด้านฮาร์ดแวร์ (HARD WARE) มีน้อย

2.5.1.2 สามารถที่จะใช้อัตราของ สัญญาณนาฬิกา ได้หลายระดับ ซึ่งสามารถเลือกใช้ให้เหมาะสม กับงานได้โดยไม่ต้องเปลี่ยนระบบฮาร์ดแวร์

2.5.1.3 ต้นทุนในการผลิตถูกกว่า CPU ตัวอื่น เมื่อเทียบความสามารถแล้วใกล้เคียงกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.1.4 ในปัจจุบันนี้ยังมีการพัฒนา CPU Z-80 ต่อไปอีกโดยพัฒนาทางด้านเทคโนโลยี (Technology) CMOS ซึ่งทำให้สิ้นเปลืองกำลังงานน้อยมาก ในส่วนของ การพัฒนาความเร็ว ซึ่งสามารถใช้กับความเร็วจนถึง 20MHz แต่ใช้ชื่อใหม่ว่า Z-84C0020 โดย compatible กับโค้ดเดิมทุกอย่าง

2.5.1.5 มีบุคลากร ระดับช่างเทคนิคในท้องตลาดจำนวนมากที่มีความรู้ และ ความชำนาญในระบบไมโครคอมพิวเตอร์ที่มี ไมโครโปรเซสเซอร์เซอ์ Z-80 เป็น CPU ดังนั้น ปัญหาทางด้านการซ่อมบำรุง สามารถที่จะตัดทิ้งได้ เมื่อโครงการนี้ ได้ถูกพัฒนาเป็น อุตสาหกรรม

2.7 โครงสร้างภายในของไมโครโปรเซสเซอร์ Z-80

สามารถแยกสายสัญญาณออกเป็น 6 ส่วนใหญ่ๆ คือ

2.7.1 บัสข้อมูล (Data bus) บัสข้อมูลมี 8 เส้น ขนาด 8 บิต เป็นบัสชนิดสอง ทิศทาง ข้อมูลสามารถที่จะเข้าหรือออกจาก CPU กับอุปกรณ์ สลับสับเปลี่ยนได้ และ เป็นลักษณะ ลอจิก 3 สถานะ

2.7.2 บัสแอดเดรส (Address bus) บัสแอดเดรสมี 16 เส้น สามารถอ้าง แอดเดรสได้โดยตรงทั้งหมด 2^{16} ตำแหน่ง (65536 ตำแหน่ง) และ สามารถอ้าง แอดเดรสของอุปกรณ์สลับสับเปลี่ยน ที่เป็น อินพุต/เอาต์พุต (Input/Output) ได้อีก 2^8 ตำแหน่ง (256 ตำแหน่ง) เป็นทิศทางเดียว ลักษณะลอจิกเป็น เอาต์พุต 3 สถานะ

2.7.3 ควบคุมระบบ (Control bus) สายสัญญาณ ที่ทำหน้าที่ควบคุมระบบมี ทั้งหมด 6 เส้นคือ

- M1 เป็นสัญญาณเอาต์พุต โดยส่งสัญญาณออกมา เพื่อบอกให้ทราบว่ากำลัง อยู่ในสภาวะเฟต (Fetch) คำสั่งโดยจะแอดดที่ลอจิก 0
- MREQ เป็นสัญญาณเอาต์พุต โดยแอดดที่ลอจิก 0 เมื่อต้องการเขียนหรือ อ่านหน่วยความจำ
- IORQ เป็นสัญญาณเอาต์พุต โดยจะแอดดที่ลอจิก 0 เมื่อต้องการที่จะ เขียนหรืออ่านอุปกรณ์ที่เป็น อินพุต/เอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- RD เป็นสัญญาณเอาต์พุต เป็นการบอกอุปกรณ์ภายนอกว่า CPU ต้องการอ่านหน่วยความจำหรืออุปกรณ์อินพุต แอดเดรสที่ลอจิก 0
- WR เป็นสัญญาณเอาต์พุตเป็นการบอก CPU ต้องการที่จะเขียนลงความจำหรืออุปกรณ์เอาต์พุต แอดเดรสที่ลอจิก 0
- RFSH เป็นสัญญาณเอาต์พุตที่บอกว่าขณะนี้สายแอดเดรสจะบรรจุข้อมูลแอดเดรสสำหรับสำหรับการรีเฟต (Refetch) หน่วยความจำชนิดไดนามิก

2.7.4 ควบคุม CPU สายสัญญาณที่ควบคุมการทำงานของ CPU มีทั้งหมด 5 เส้น

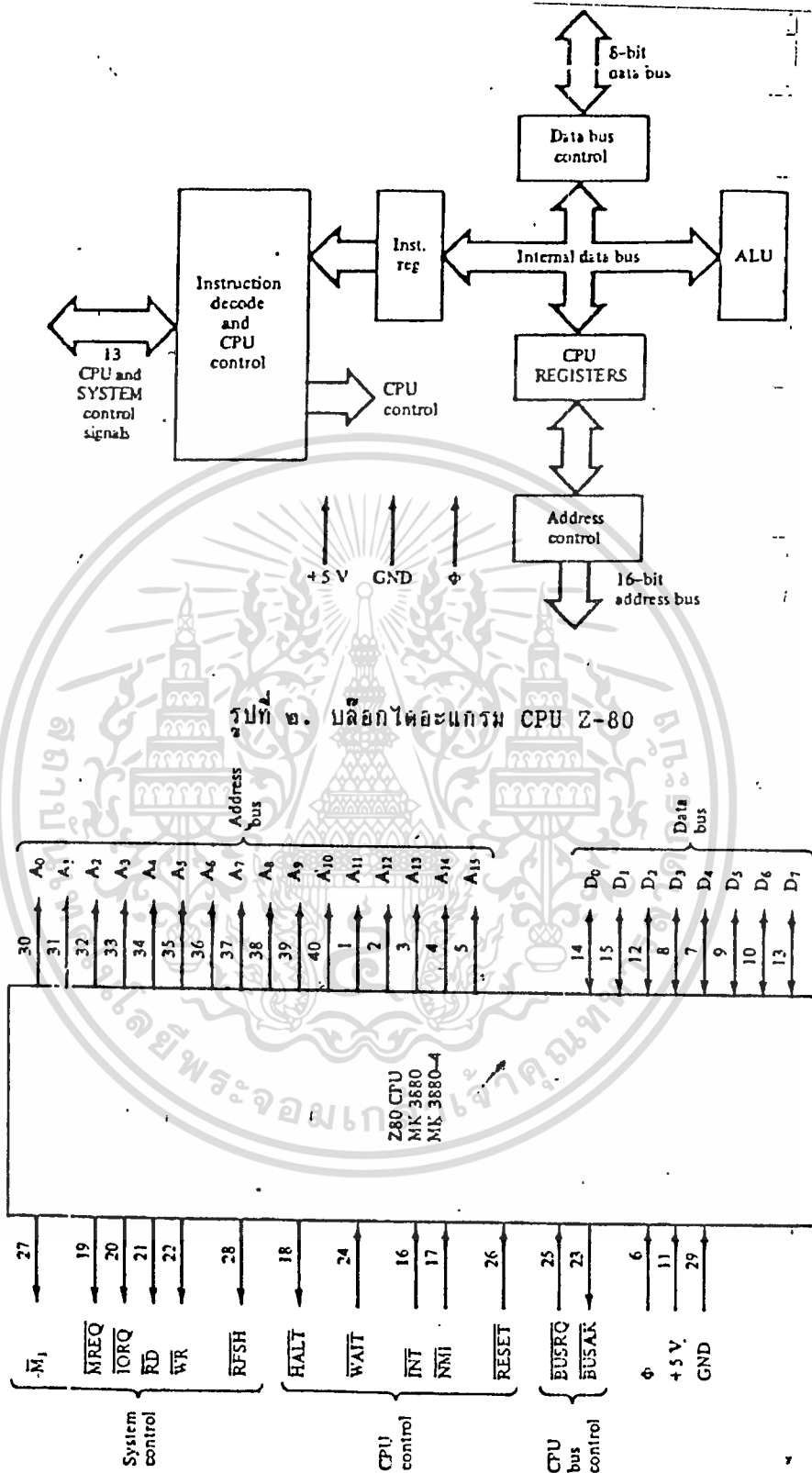
- HALT เป็นสายสัญญาณโดยจะแสดงการทำงานของ CPU เมื่อได้รับคำสั่ง HALT ซึ่งจะแอกทีฟ "0"
- WAIT เป็นสัญญาณอินพุต ซึ่งเป็นการบอก CPU ให้ทราบว่าอุปกรณ์ที่ CPU จะติดต่อด้วยนั้น ยังไม่พร้อมในการที่จะรับหรือส่งข้อมูล CPU ก็รอจนกว่า อุปกรณ์ที่ CPU ติดต่อด้วยเลิก WAIT ก็ทำงานโดยปกติต่อไป
- INT เป็นสัญญาณจากอุปกรณ์ อินพุต เอาต์พุตที่ต้องการจะอินเทอร์รัพท์ CPU ซึ่งมีอยู่ 3 โหมด คือ
 - การอินเทอร์รัพท์โหมด 0
 - การอินเทอร์รัพท์โหมด 1
 - การอินเทอร์รัพท์โหมด 2
- NMI เป็นสัญญาณจากอุปกรณ์ อินพุตเอาต์พุตเมื่อเข้ามา CPU จะทำการตอบสนองทันที โดยจะกระโดดไปกระทำยังตำแหน่งแอดเดรส 0066H ทันที และจะเก็บค่าโปรแกรมเคาน์เตอร์ เดิมไว้บนสแต็ก เมื่อจะกลับมาตำแหน่งเดิมได้โดยคำสั่ง RETN
- RESET เป็นการ reset CPU เพื่อกำหนดโปรแกรมเคาเตอร์ (Program counter) มีค่าเท่ากับ 0 คือเริ่มต้นใหม่



2.7.5 ความคับคั่ง ใช้ในการควบคุมสถานะของบัสมีทั้งหมด 2 เส้น

- BUSRQ เป็นการส่งสัญญาณเพื่อบอก CPU ว่า ขณะนี้ต้องการใช้บัสโดย ใช้หลักการลอจิก 3 สถานะเพื่อเป็นการกำหนดให้ระบบบัสแยกออกจาก CPU ให้หน่วยความจำกับ อุปกรณ์ภายนอกเคลื่อนย้ายข้อมูลระหว่างกันได้
- BUSAK เป็นสัญญาณที่ส่งออกจาก CPU เพื่อบอกอุปกรณ์ภายนอกให้ทราบว่าขณะนี้ CPU ไม่ใช้บัสแล้วให้ผู้ที่ขอใช้บัสใช้บัสได้เลย





บทที่ 3

หลักการสร้าง

การออกแบบทั้งหมดของเครื่องคิดเงินค่าใช้โทรศัพท์จำเป็นต้องกล่าวแยกเป็นส่วนๆดังนี้

- CPU กับ การถอดรหัสหน่วยความจำ และถอดรหัสพอร์ต อินพุต/ เอาท์พุต
- CPU กับ 8279 ชิพสนับสนุน (Ship support) Key board/Display
- CPU กับ Real Time
- CPU กับ port 8255 ลำโพง 8870 LCD Display
- CPU กับวงจร Detect telephone signal

3.1 Z-80 กับการถอดรหัสหน่วยความจำ

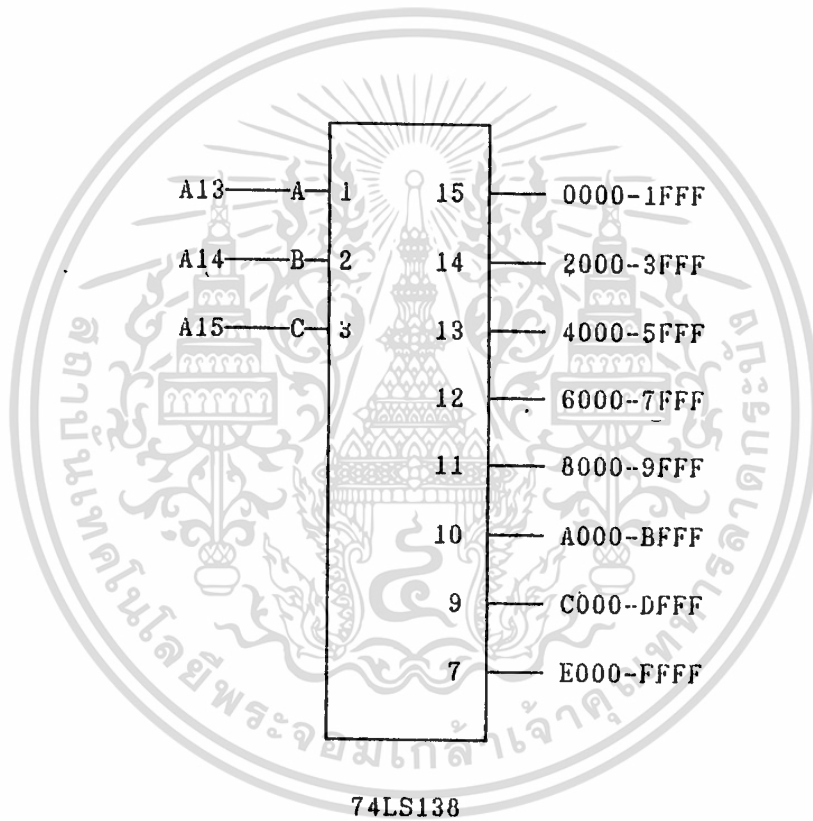
Z-80 มีสัญญาณแอดเดรสทั้งหมด 16 เส้น ฉะนั้น Z-80 สามารถอ้างแอดเดรสโดยตรงทั้งหมดได้ถึง 64K byte ตั้งแต่ตำแหน่ง 0000-FFFF ฉะนั้นจึงแบ่งการอ้างแอดเดรสออกเป็น 8 ส่วนๆละ 8K byte ตามตาราง memory map ข้างล่างนี้

ตำแหน่ง	ช่วงหน่วย	สายสัญญาณแอดเดรสที่จะนำมาถอดรหัส 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0	ชนิดหน่วย ความจำ
1	0000-1FFF	0 0 0 * * * * * * * * * * * *	ROM
2	2000-3FFF	0 0 1 * * * * * * * * * * * *	RAM
3	4000-5FFF	0 1 0 * * * * * * * * * * * *	OPTION
4	6000-7FFF	0 1 1 * * * * * * * * * * * *	OPTION
5	8000-9FFF	1 0 0 * * * * * * * * * * * *	OPTION
6	A000-BFFF	1 0 1 * * * * * * * * * * * *	OPTION
7	C000-DFFF	1 1 0 * * * * * * * * * * * *	OPTION
8	E000-FFFF	1 1 1 * * * * * * * * * * * *	RAM

ตารางที่ ๑. MEMORY MAP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

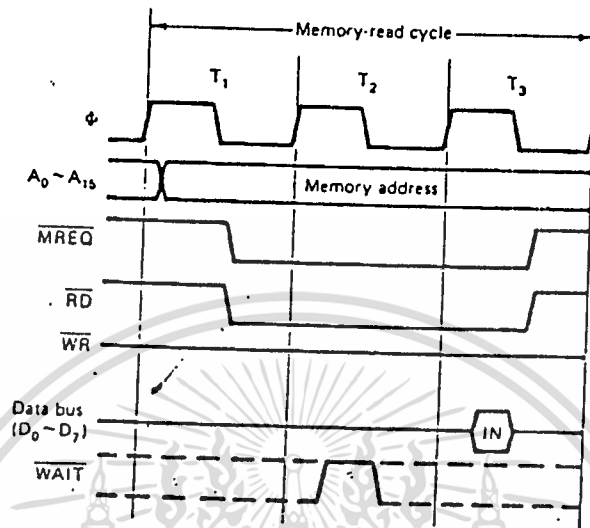
จากตารางข้างบน จะเห็นว่าเราได้แบ่งตำแหน่งการแบ่งการอ้างถึงแอดเดรสโดยตรง ออกเป็น 8 ส่วนๆละ 8K byte ในแต่ละส่วนช่วงของแอดเดรส จะมีสายสัญญาณ A13-A15 เท่านั้นที่เปลี่ยนแปลงตามช่วงของการแบ่งแอดเดรส แบบ 8 ส่วนๆละ 8K byte ซึ่งเราสามารถนำสายสัญญาณ A13-A15 เข้าวงจรถอดรหัสเพื่อเลือกตำแหน่งของหน่วยความจำ ไอซีที่ใช้ในการถอดรหัสคือ ไอซี ทิทแอล เบอร์ 74LS138 โดยมีหลักการคิดดังนี้ $2^{13}=8 \text{ kbyte}$ ฉะนั้นถ้าเริ่มนำสายแอดเดรสมาต่อเข้ากับ IC 74LS138 คือต่อ A13,A14,A15 เข้ากับ A,B,C ตามลำดับ ก็ได้ช่วงละ 8 kbyte



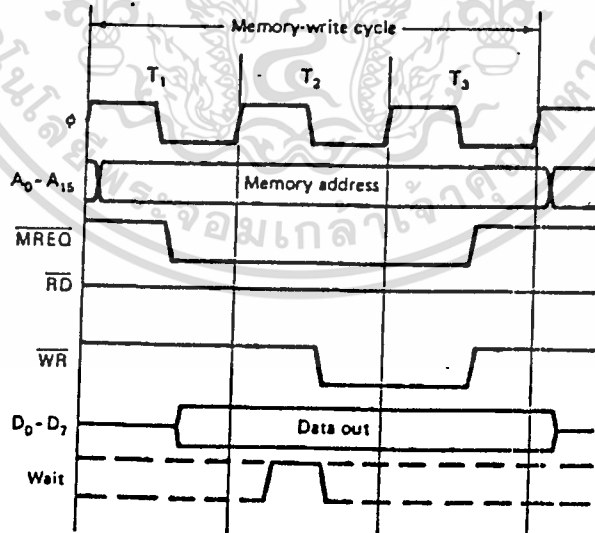
รูปที่ ๔. ตำแหน่งการถอดรหัสของ IC 74LS138

ก่อนที่จะนำ Z-80 ต่อเข้ากับหน่วยความจำ เราจำเป็นต้องทราบถึงแมชชีนไซเคิล (machine cycle) ที่เกี่ยวข้องกับหน่วยความจำเสียก่อน

Z-80 Memory-read/write Machine cycle



รูปที่ ๕. ไตอะแกรมเวลาสัญญาณที่ใช้ในการอ่านข้อมูลจากหน่วยความจำ



รูปที่ ๖. ไตอะแกรมเวลาแสดงสัญญาณที่ใช้ในการเขียนข้อมูลลงบนหน่วยความจำ

จากเมฆขึ้นไฉ่เกิดจะเห็นว่าสัญญา MERQ กับ RD จะต้องแยกที่พร้อมกัน ฉะนั้นเมื่อต้องการที่จะอ่านหน่วยความจำต้องรอจนกว่าสัญญา MREQ และ RD แยกที่พร้อมกันก่อนจึงจะออกมาจากหน่วยความจำ ส่วนการอ่านหน่วยความจำก็เช่นเดียวกันคือ เมื่อสัญญา MREQ และ WR แยกที่พร้อมกันก่อนจึงจะสามารถที่จะทำการเขียนลงบนหน่วยความจำได้



3.2 Z-80 กับอุปกรณ์อินพุต/เอาต์พุต

ในโครงงานนี้จำเป็นต้องมีการติดต่อส่งข้อมูลระหว่าง CPU กับ อุปกรณ์อินพุต/เอาต์พุต ในการรับส่งนั้นอาศัย Register A กับ Bus ข้อมูลโดยมีการควบคุมสัญญาณและการกำหนด แอดเดรสคล้ายกับหน่วยความจำ แต่เราจะเรียกอุปกรณ์อินพุต/เอาต์พุตว่าพอร์ตซึ่งสายสัญญาณที่ใช้ในการอ้างแอดเดรสเบอร์พอร์ตรหัสคือ A0-A7 ดังนั้นจึงสามารถที่จะอ้างพอร์ทได้ 256 พอร์ท คือตั้งแต่ 00-FF ซึ่งต้องอาศัยวงจรถอดรหัสเพื่อที่จะกำหนดตามที่เรากำลังต้องการ โดยจะแบ่งพอร์ท ทั้งหมดออกเป็น 8 ส่วนๆละ 32 พอร์ท

ตำแหน่ง	ช่วงเบอร์พอร์ท	สายแอดเดรสที่นำมาถอดรหัส								ชนิดของพอร์ท
		A7	A6	A5	A4	A3	A2	A1	A0	
1	00-1F	0	0	0	*	*	*	*	*	FOR FUTURE
2	20-3F	0	0	1	*	*	*	*	*	8255, 8870, DET
3	40-5F	0	1	0	*	*	*	*	*	REAL TIME
4	60-7F	0	1	1	*	*	*	*	*	KEYBOARDS

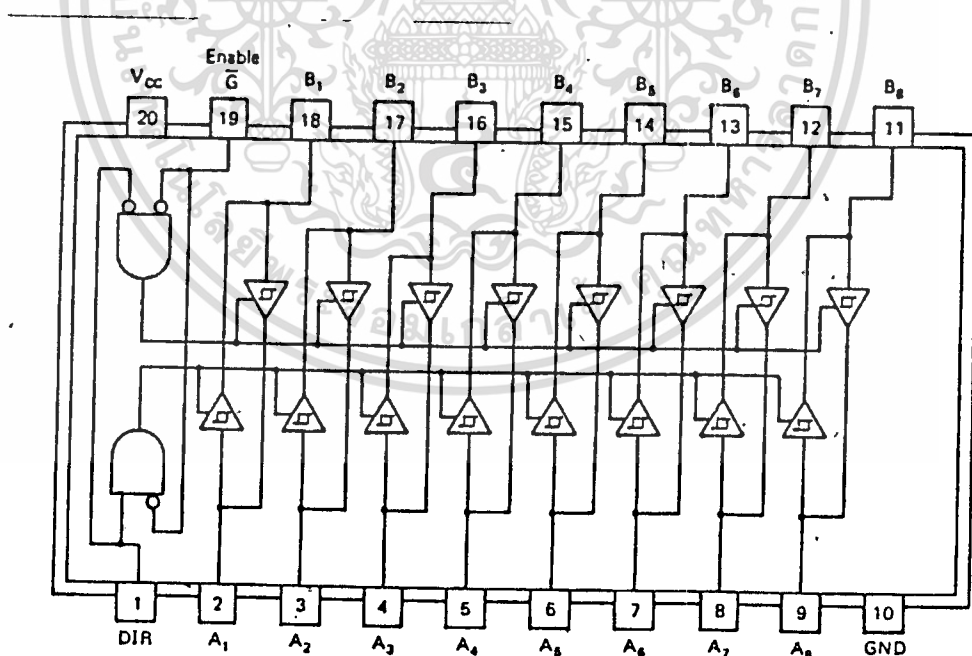
ตำแหน่ง	ช่วงเบอร์ พอร์ต	สายสัญญาณที่นำมาต่อรหัสเบอร์พอร์ต								ชนิดของพอร์ต
		A7	A6	A5	A4	A3	A2	A1	A0	
5	80-9F	1	0	0	*	*	*	*	*	FOR FUTURE
6	A0-BF	1	0	1	*	*	*	*	*	DETECT
7	C0-DF	1	1	0	*	*	*	*	*	FOR FUTURE
8	E0-FF	1	1	1	*	*	*	*	*	

ตารางที่ ๒. PORT INPUT/OUTPUT MAP

3.3 ลักษณะของวงจร

สัญญาณนาฬิกาใช้ขนาด 4 MHz ให้กับ Z-80A ใช้ ROM ขนาด 8K เบอร์ 2764 1 ตัว และ RAM 8k 2 ตัว โดย ROM จะอยู่ที่ตำแหน่ง 0000-1FFF ส่วน RAM อยู่ที่ตำแหน่ง 2000-3FFF ตัวหนึ่งและที่ตำแหน่ง E000-FFFF อีกตัวหนึ่ง เราจะกำหนดพอร์ตเบอร์ 1 สำหรับเลือก Display, D/A, Detect Signal พอร์ตเบอร์ 2 สำหรับ Real Time พอร์ตเบอร์ 3 สำหรับ Key Board ส่วนพอร์ตเบอร์ 4 ทำไว้เพื่อพัฒนาเครื่องโดยใช้ในการ Load Data จากเครื่อง PC XT/AT และสามารถที่จะเก็บข้อมูลจากเครื่องเข้าไปไว้ในเครื่อง PC XT/AT ได้ถ้าหาก Data มีขนาดใหญ่เกินกว่าที่เครื่องนี้จะสามารถที่จะเก็บไว้ได้

สำหรับการต่อ 74LS245 เป็นคาตาบัฟเฟอร์นี้ใช้สายสัญญาณ Read ของ CPU เป็น 0 ในการกำหนดทิศทางการไหลของข้อมูล จาก B1 ไปทางด้าน A1 ในทางกลับกันถ้าเป็น 1 ก็เป็นการไหลจาก A1 ไป B1 ฉะนั้นเราจึงต่อสายสัญญาณ Read เข้ากับขา Dir ของไอซี 74LS245 ส่วนขา Enable จะต่อเข้ากับสายสัญญาณ BUSAK ของ CPU ที่ต่อมานี้เพราะเพื่อให้เมื่อต้องการขยายระบบอาจจะเชื่อมกับคิส์ไทร์ ก็สามารถที่จะต่อได้เลยกับระบบนี้



รูปที่ ๗. ลักษณะการต่อสัญญาณควบคุมทิศทางของ IC 74LS245 Buffer Data

3.4 Z-80 กับ 8279 Keyboard/display

3.4.1 8279 PROGRAMABLE KEYBOARD/DISPLAY INTERFACE

คุณสมบัติที่สำคัญ

- สามารถ Scankey /Display LCD ได้
- Display 16 ตัวเลข เข้าทางซ้ายหรือทางขวาก็ได้
- โปรแกรมโหมดการทำงานได้โดย CPU
- สามารถ Interrupt CPU เมื่อมีการกด KEY

Software Operater

โหมด Keyboard

code: 000**KKK ** for display

*2 Key lockout 000= Encode scan keyboard

2 Key lockout 001= Decode scan keyboard

N key rollover 010= Encode scan keyboard

N key rollover 011= Decode scan keyboard

Dipswitch 100= Encode scan matrix

Dipswitch 101= Decode scan matrix

Dipswitch 110= Strobe input enable display scan

Dipswitch 111= Strobe input decode display scan

Program Clock

code: 0001PPPPP

10100 For 2 MHz จะถูกหารด้วย 20 ซึ่งจะมีสัดส่วนในการทำ

งานคือ 100KHz

Read FIFO/Senser RAM

Code: 010AI*AAA

โดยการเขียนคำสั่งนี้ในการ Scankey

AI:Auto-increment

ADDRESS ของแรมอยู่ที่ AAA

A0=1 เป็นคำสั่ง

A0=0 เป็นDATA

เมื่อใช้ใน Mode senser matrix address ของแรมอยู่ที่ AAA ถูกเลือก 1/8 ของแถวของ Senser ram ถ้า AI ถูกปรับเป็น 1 ของแต่ละช่วงของการอ่านจากแต่ละแถวของ Senser ram

คำสั่งในการยอมให้อ่าน Ram display

เป็นคำสั่งที่ CPU จะไปอ่านข้อมูลใน Display buffer ตามลักษณะข้างล่าง

Read Display Ram : 0 1 1 AI A A A A

AAAA คือตำแหน่งในการอ่าน ส่วน AI เป็นการเพิ่มตำแหน่งของแรมโดยอัตโนมัติ (AAAA=AAAA+1) คำสั่งในการเขียนข้อมูลลงใน Display Ram มีลักษณะเหมือนกับการอ่านในเรื่องของ ตำแหน่งและ AI

Write Display Ram : 1 0 0 AI A A A A

คำสั่งนี้ไม่มีผลต่อแหล่งข้อมูลที่ตามมา โดย CPU จะอ่านจาก แหล่งข้อมูลใดก็ตาม (FIFO/Senser หรือ Display Ram) ที่เพิ่งถูกเลือกผ่านไป ถ้าแหล่งข้อมูลที่ถูกเลือกผ่านไป เป็น Display Ram คำสั่งในการเขียนก็จะเปลี่ยนตำแหน่งของตำแหน่งถัดไป

คำสั่ง Display inhibit /blanking เป็นคำสั่งที่ใช้ในการห้ามการเขียนลงไป ใน Ram Display หรือให้ Display ที่ว่าง คำสั่งนี้ใช้ 4 bit port ในการห้ามการเขียน สามารถจะทำบนแต่ละ 2 port อย่างอิสระ

Display inhibit : 1 1 0 * IW_A IW_B BL_A BL_B

IW bits ใช้ในการห้ามเขียนเป็น port A หรือ B ถ้าเราใช้ BCD decoding สำหรับเราใช้ 4bit 2 ชุด แยกกัน เราสามารถห้ามการเขียนอันหนึ่งขณะที่เราเขียนข้อมูลอีกอันหนึ่งได้ โดย port B เป็น bit นัยต่ำสุดของการแสดงการเขียนข้อมูล ซึ่ง port เป็นนัยสูงสุด BL bits แสดงถูกให้เป็นคำสั่งการเคลีย เมื่อเราใช้ Display เป็น 8bit โดยรวม port A กับ port B เข้าด้วยกัน bit BL จะต้องตั้งให้เป็น blank เมื่อแสดงออกมา

กำหนดคำสั่งการเคลีย ส่งรหัสไป Display เมื่อให้เป็นที่ว่าง และสามารถทำการเคลีย Ram และ reset ด้วย

Clear : 1 1 0 E_n C_d C_e C_n

ประกอบไปด้วยส่วนการ enable 3 bit C_d ถูกกำหนดเป็น C_d C_e ที่ว่างสำหรับการแสดง

C_d	C_e	Blanking code
0	*	all zero
1	0	hex 20
1	1	all ones

E_n bit ถ้าเป็น "1" จะยอมให้มีการ clear display ได้ ในระหว่างการเคลีย จะไม่สามารถเขียนลงไปใน ram display ได้ สถานะยังคงอยู่จนกว่าจะสิ้นการเคลีย bit CF ถ้าเป็น 1 จะทำการเคลียสถานะด้วย และ ตัวชี้ senser ram ก็จะถูกกลับไปอยู่ที่ 0

bit CA จะถูก set เมื่อรวมผลของ EN และ CF ในการรวมกันมันจะทำการชิงโทรไลน์กันใหม่ ภายในเวลาที่ต่อเนื่อง

คำสั่ง the end interrupt/error mode หรือ interrupt request line จะเป็น 0 เมื่อต้องการเขียนลงไปในแรม คำสั่งนี้จะถูกเซต เมื่อเกิดการ error สำหรับ N-KEY rollover

1 1 1 E * * * *

End interrupt/error mode set

ในส่วนของ NNN จะเป็นตัวชี้จำนวนของตัวหนังสือที่ป้อนเข้าไปใน FIFO ในโหมดอินพุต ในส่วนนี้จะไม่มีผลต่อโหมด scan sensor

bit F จะเป็นตัวชี้ว่า FIFO เต็มแล้วและจำเป็นที่จะอ่านไปก่อนเพื่อให้ตัวอื่นเข้ามาแทน และไม่มีผลต่อโหมด scanned sensor

bit U จะบ่งชี้ว่าได้เกิดการผิดพลาดในการทำงาน คือ CPU อ่าน FIFO ที่ว่างเปล่า จะไม่มีผลในโหมด scanned sensor

bit 0 จะบ่งชี้ว่าได้เกิดการผิดพลาดในการทำงาน คือ ตัวหนังสือเข้าไปเก็บใน FIFO จนเต็มแล้ว

bit s/n มี 2 บิตที่กำหนดโดยการกำหนดโหมดการทำงานของ 8279 คือ

- Sanned sensor mode บิตนี้จะถูกเซตเมื่อแสดงเมื่อถูกบรรจุใน sensor ram

- N - Key rollover mode บิต special error mode เซต bit นี้ เมื่อเกิดการผิดพลาดเกิดขึ้น

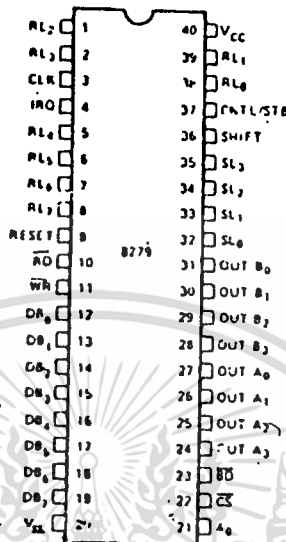
bit DU บ่งชี้ไม่เปลี่ยนแปลงการแสดงผล เพราะว่าจะเคลียร์การแสดงผลหรือคำสั่งทั้งหมดที่ไม่สมบูรณ์

keyboard data : C S SC SC SC RL RL RL

บิตนี้จะแทนตำแหน่งของ switch matrix กับ หรือ control และ shift โดย RL และ SC จะแทนที่ตำแหน่งของ matrix ของการกด switch SC ในส่วนของการแสดงผลจะแสดงเมื่อคีย์ถูกกด และจะตรวจรับ RL เป็น line นำเข้าเมื่อมีการกด สวิตช์ bit S และ C จะใช้ในการ shift และ control

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIN CONFIGURATION



PIN NAMES

Pin	I/O	DATA BUS/BI-DIRECTIONAL
CLK	I	CLOCK INPUT
RESET	I	RESET INPUT
CS	I	CHIP SELECT
AL	I	READ INPUT
WR	I	WRITE INPUT
A0	I	BUFFER ADDRESS
IRQ	O	INTERUPT REQUEST OUTPUT
SL03	O	SCAN LINES
PL03	I	RETURN LINES
SHIFT	I	SHIFT INPUT
CNTL/STB	I	CONTROL/STROBE INPUT
OUT A03	O	DISPLAY A0 OUTPUTS
OUT B03	O	DISPLAY B0 OUTPUTS
B5	O	BLANK DISPLAY OUTPUT

รูปที่ ๘. แสดงตำแหน่งขาของ Intel 8279 keyboard

Intel 8279 เป็นชิพซีพพอร์ท Programable Keyboard Display Controler ซึ่งสามารถที่จะโปรแกรมโหมดการทำงานได้ตามลักษณะงานที่ต้องการ เราสามารถที่จะต่อ 8279 เข้ากับ Z-80 ได้โดย 8279 ต้องการ A0 จาก Z-80 ต่อเข้ากับ A0 ของ 8279 โดยทำให้ A0 เป็น "1" แสดงว่าเราต้องการที่จะโปรแกรมโหมดการทำงานให้กับ 8279 และถ้าเป็น "0" แล้วจะเป็นการรับส่งข้อมูลระหว่าง Z-80 กับ 8279

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

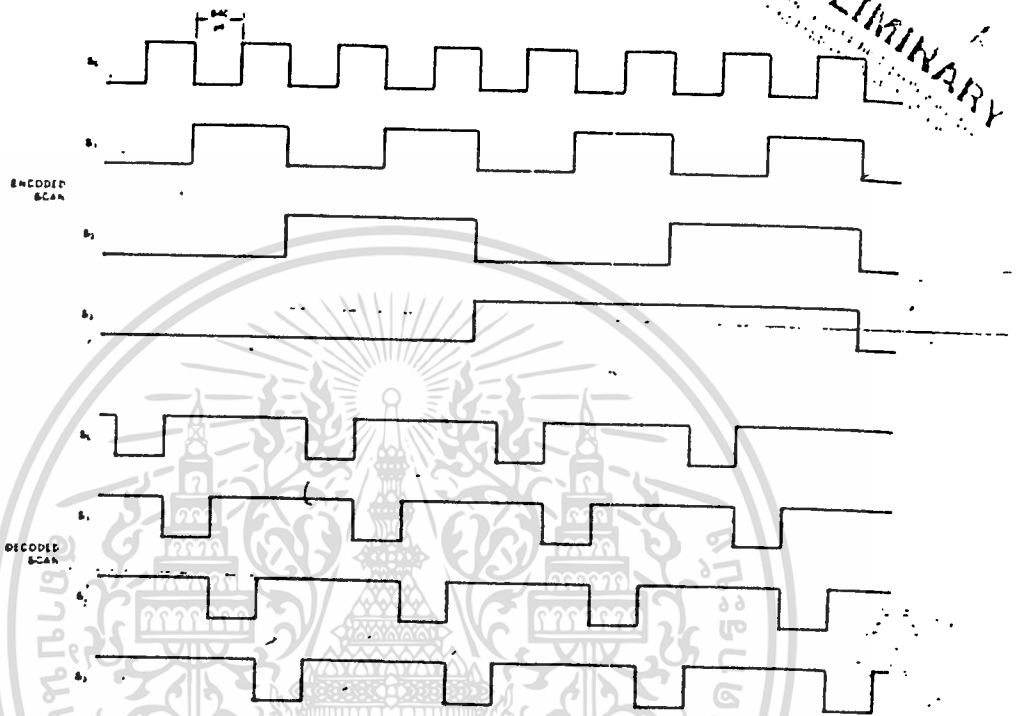
8279 มีขาข้อมูล 8 ขา โดยสามารถต่อโดยตรงกับ Z-80 ส่วนสัญญาณการควบคุมการทำงานคือสัญญาณ RD WR เป็นการกำหนดการอ่านและเขียนลงบน 8279 ตามลำดับดังตาราง

WR	RD	IO/M	การทำงาน
0	1	1	I/O WRITE
1	0	1	I/O READ
0	1	0	MEMORY READ
1	0	0	MEMORY WRITE

ตารางที่ ๓. สัญญาณควบคุมการทำงาน

โดย 8279 นี้จะทำการสแกน Keyboard เองแทนการที่จะสแกนด้วย CPU เป็นการแยกกันอย่างอิสระโดยจะต่อเชื่อมกันเฉพาะ บัสข้อมูลและบัสควบคุมเท่านั้น และ 8279 ยังสามารถที่จะสแกน Keyboard ได้ถึง 64 Key และมีสายสัญญาณ Shift, Cntl ซึ่งทำให้เพิ่มคีย์ได้อีก 2 เท้า แต่สายสัญญาณการสแกนมีเพียง 3 เส้นจึงต้องใช้ IC 74LS138 เพื่อ Decode ออกเป็น 8 เส้น และส่งคำสั่งการควบคุมโหมดแบบ Encode ดัง Timing ข้างล่างนี้

SCAN WAVEFORMS

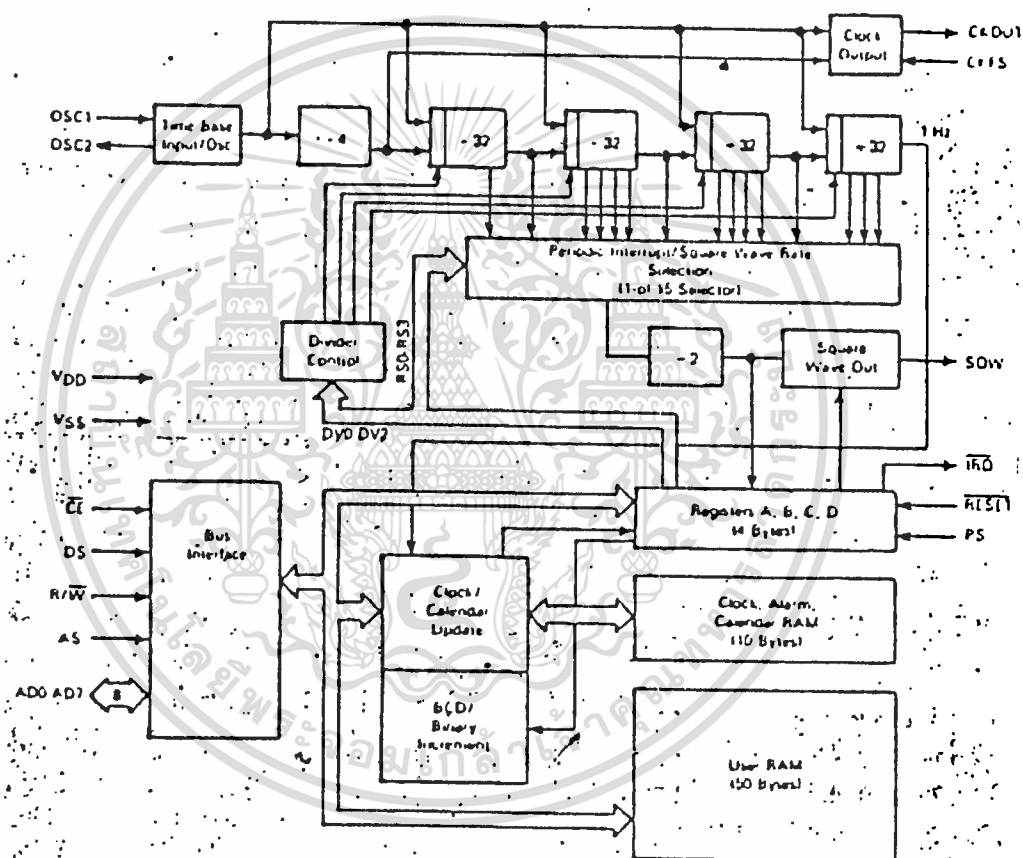


รูปที่ ๙. 8279 Scan Timing Diagram

จาก Timing ข้างบนนี้เป็นการเปรียบเทียบ Timing ทั้ง 2 แบบถ้าต้องการใช้เพียง 24 คีย์ก็ไม่ต้องต่อ IC 74LS138 แต่ต้องกำหนดโหมดการทำงานสแกนด้วยแบบ Decode แต่ถ้าหากต้องการใช้ 64 คีย์ก็ต้องต่อกับ IC 74LS138 แล้วกำหนดโหมดการทำงานสแกนด้วยแบบ Encode ซึ่งเป็นการ decode ภายนอกในการใช้งานโหมด Encode จะใช้สายสัญญาณการสแกนเพียง 3 เส้น คือ SL_0 - SL_2 และ IC 8279 ได้เตรียมสายสัญญาณ shift และ control คีย์ไว้ด้วย แต่มิได้เข้าไปควบคุม IC 8279 โดยตรง แต่มันจะนำสถานะของคีย์ เข้าไปเก็บไว้ใน FIFO พร้อมกับคีย์ที่กดพร้อมกัน ซึ่งสามารถมีคีย์ได้ 256 รหัส

3.5 Z-80 กับ REALTIME

ในส่วนนี้เราจะกล่าวถึงวงจรส่วนที่ทำหน้าที่เป็นนาฬิกาหรือที่เรียกว่า RTC(real time) เป็นนาฬิกาบอกเวลาที่แท้จริงให้กับ ไมโครคอมพิวเตอร์ โดยต่อเชื่อมเข้ากับ CPU ก็สามารภที่จะอ่านเวลาจากตัว RTC ได้ทุกครั้งที่ CPU ต้องการเพื่อใช้เป็นตัวที่จะบอกเวลาให้กับระบบ เพื่อใช้เป็นฐานเวลาในการใช้คิดค่าใช้บริการเช่น เวลาเริ่มต้นติดต่อโดยผ่านเครื่องคิดเงินค่าใช้บริการโทรศัพท์ และเวลาที่หยุดการติดต่อ วงจร RTC นี้โดยทั่วไปแล้วมีไอซีที่ทำหน้าที่นี้มีอยู่หลายเบอร์ แต่ในโครงงานนี้ เราจะใช้ RTC เบอร์ MC146818 ซึ่งมีประสิทธิภาพสูงกว่าเบอร์อื่น และสามารถหาซื้อได้จากทั่วไป MC146818 เป็นไอซีชนิด CMOS ซึ่งกินกระแสไฟน้อยมาก ดังนั้นเราจึงสามารถสำรองแหล่งจ่ายไฟด้วยแบตเตอรี่ขนาดเล็ก เพื่อกรณีที่ไฟดับหรือปิดเครื่อง โดยที่ RTC ยังคงสามารถที่จะทำงานได้ ข้อมูลใน RTC ยังคงอยู่และเวลาใน RTC ยังคงสามารถเดินต่อไปด้วยเวลาที่ถูกต้อง และที่สำคัญคือ กรณีที่เดือนที่ลงท้ายด้วย "ยน" ซึ่งมี 30 วัน และเดือนที่ลงท้ายด้วย "คม" ซึ่งมี 31 วันหรือเดือนกุมภาพันธ์ซึ่งมี 28 วัน และในปีอธิกสุรทินจะมี 29 วัน RTC ก็ยังสามารถที่จะเดินได้ถูกต้องทุกประการ และยังมีพิเศษกว่านั้นคือ MC146818 ยังมีฟังก์ชันที่สามารถปรับว่าจะให้แสดงเวลาในประเทศไทยหรือประเทศอเมริกา เพราะเวลาในประเทศอเมริกาจะมีการปรับเวลาให้เร็วขึ้น 1 ชั่วโมงในปลายเดือนเมษายน และปรับเวลาให้ช้าลง 1 ชั่วโมงในปลายเดือนตุลาคม



รูปที่ ๑๐. แสดงโครงสร้างภายใน RTC เบอร์ MC146818

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.1 คุณสมบัติที่สำคัญของ MC146818

โครงสร้างภายในของ MC146818 แสดงในรูปที่ 1 ซึ่งมีคุณสมบัติดังนี้

3.5.1.1 กินกระแสต่ำ เพราะเป็นไอซี CMOS โดยจะกินกระแสประมาณ 250uA ที่ความถี่ 32.768KHz และประมาณ 750uA ที่ 4.194304MHz สามารถสำรองไฟเลี้ยงด้วย แบตเตอรี่ขนาดเล็กได้

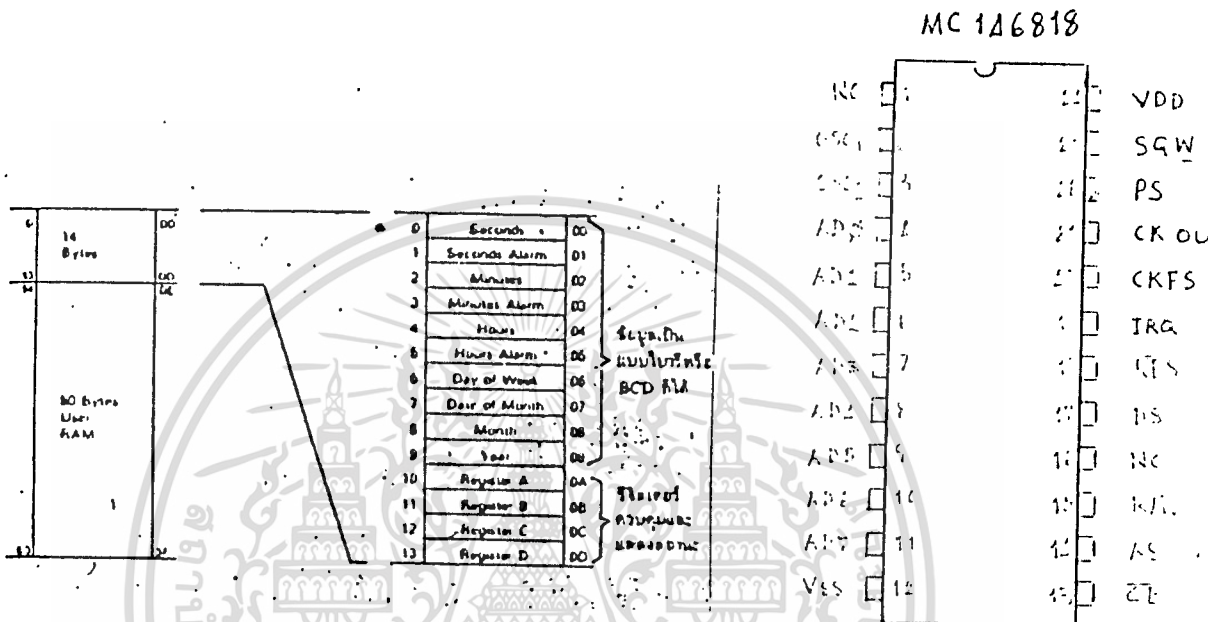
3.5.1.2 ใช้กับคริสตัล ได้ 3 ความถี่คือ 32.768KHz, 1.048576MHz, 4.194304MHz ซึ่งสามารถที่จะเลือกได้โดยทางซอฟต์แวร์

3.5.1.3 มีหน่วยความจำรวม 64byte โดยแบ่ง 10byte แรกเป็นที่เก็บเวลาและเวลาปลุก 4byte หลังเป็นรีจิสเตอร์ควบคุมและแสดงสถานะ (รีจิสเตอร์ A, B, C, D) และอีก 50byte เป็นแรมที่สามารถที่จะใช้ได้ตามประสงค์

3.5.1.4 การนับเวลา นับได้ทั้งแบบ 12 ชั่วโมงและ 24 ชั่วโมงและยังสามารถที่จะแสดงด้วยรหัสไบนารีหรือ BCD

3.5.1.5 กำหนดการอินเตอร์รัพต์ได้ 3 ลักษณะคือ

- interrupt ตามเวลาปลุก
- interrupt ตามความถี่ที่ตั้งไว้
- interrupt ทุกๆวินาทีหลังการปรับเวลาในตัว RTC



รูปที่ ๑๑. การจัดแอดเดรสของหน่วยความจำ

รูปที่ ๑๒. การจัดขาของ MC146818

3.5.2 การจัดหาและความหมาย

3.5.2.1 OSC1, OSC2 ต่อเข้ากับตัวคริสตัลได้โดยตรง หรือ บ่อนความถี่จากภายนอกเข้าที่ขา osc1 โดยตรง

3.5.2.2 AD0-AD7 (ADDRESS/DATA BUS) เป็นบัสแอดเดรสและดาต้าบัสรวมกัน โดยจะทำงานในลักษณะมัลติเพลก โดยจะเป็นแอดเดรสบัสก่อน เมื่อ AS active และ จะเป็น DATA BUS เมื่อ DS หรือ R/W active

3.5.2.3 CE (CHIP ENABLE) เป็นขาควบคุมการทำงานของบัสและสัญญาณอื่นๆ

เอกสารนี้จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.2.4 AS (ADDRESS STROBE) ทำหน้าที่แลตซ์ขา ADO-AD7 เอาไว้เพื่อแสดงเป็นขาแอดแตรส

3.5.2.5 R/W ทำหน้าที่กำหนดการเขียนข้อมูลในหน่วยความจำ แลคตป์ที่ลอจิก "0"

3.5.2.6 DS (DATA STROBE) ทำหน้าที่กำหนดการอ่านข้อมูลในหน่วยความจำทั้ง 64byte แลคตป์ที่ลอจิก "0"

3.5.2.7 SQW เป็นขาสัญญาณคลื่นสี่เหลี่ยมเอาก์พุท ที่สามารถที่จะกำหนดความถี่ได้โดยกำหนดจากรีจิสเตอร์ควบคุม

3.5.2.8 RESET เป็นขาที่ใช้รีเซทระบบ เมื่อมีการรีเซทจะมีผลดังนี้

- แฟล็กสำหรับอินทิเอร์รัพท์ ทั้งสาม (PIE, UIE, AIE) จะถูกรีเซทเป็น "0"
- แฟล็กแสดงการขออินทิเอร์รัพท์ทั้งสาม (PE, UF, AF) จะถูกเคลียเป็น "0"
- ไม่สามารถที่จะเขียนหรืออ่านเวลาได้
- ขาสัญญาณขออินทิเอร์รัพท์ (IRQ) เป็น high impedance
- แฟล็กอินทิเอร์รัพท์คลื่นสี่เหลี่ยมถูกเคลียเป็น "0"

3.5.2.9 IRQ (INTERRUPT REQUEST) เป็นขาสัญญาณเอาก์พุท สำหรับการขออินทิเอร์รัพท์กับ CPU active ที่ ลอจิก "0" สามารถที่จะขออินทิเอร์รัพท์ได้สามลักษณะ

- Interrupt เป็นจังหวะตามค่าความถี่ที่ขา SQW
- interrupt ทุกวินาทีหลังจากผ่านการปรับเวลา
- interrupt เมื่อถึงเวลาปลุก

3.5.2.10 CSFS (CLOCK OUT FREQUENCY SELECT) เป็นขาสัญญาณอินพุท สำหรับการหาความถี่ที่ตัวคริสตอล แล้วส่งออกทางขา CKOUT ถ้าขานี้เป็น "0" หมายถึงการทำการหารด้วย 4 แต่ถ้าเป็น "1" หมายถึงไม่มีการหาร

3.5.2.11. CKOUT (CLOCK OUT) สัญญาณเอาก์พุท ความถี่ของระบบเพื่อนำความถี่ของระบบไปใช้งานอื่น

3.5.2.12 PS (POWER SENSE) เป็นขาสัญญาณอินพุท สำหรับควบคุมการแสดงผลสถานะของหน่วยความจำทั้ง 64byte ว่าเป็นข้อมูลที่ถูกต้องหรือไม่ ซึ่งสัญญาณจากขานี้จะแสดงสภาวะของการป้อนแรงดันเร้าตัว RTC ขณะเริ่มต้นโดยการไปแสดงผลที่แฟล็ก VRT ในรีจิสเตอร์ D เมื่อขาเป็น "0" อยู่แฟล็ก VRT จะถูกเคลียให้เป็น "0" ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5.3 รีจิสเตอร์ควบคุมและแสดงสถานะ

รีจิสเตอร์ควบคุมและบอกสถานะมีอยู่ 4byte ด้วยกันคือ รีจิสเตอร์ A,B,C,D

3.5.3.1 รีจิสเตอร์ A (เขียน/อ่านได้ยกเว้น UIP)

- UIP(UPDATE IN PROCESS) เมื่อบิตนี้เป็น "1"แสดงว่ากำลังใกล้หรืออยู่ระหว่างการปรับเวลา แต่ถ้าเป็น "0"แสดงว่าอยู่ในการทำงานปกติ บิตนี้อ่านได้อย่างเดียว ไม่มีผลต่อการรีเซต

- DV2,DV1,DV0(DIVIDER SELECTION) กำหนดค่าเลือกคริสตัลหรือฐานความถี่

-000 เลือกฐานความถี่ 4.194304MHz

-001 เลือกฐานความถี่ 1.048576MHz

-010 เลือกฐานความถี่ 32.768KHz

นอกเหนือจากนี้ไม่สามารถทำงานได้

- RS3,RS2,RS1,RS0 (RATE SELECTION) ทั้งสี่บิตเป็นตัวกำหนดความถี่ออกทาง SQW และกำหนดจังหวะการอินเทอร์รัพตามตารางที่ 1

3.5.3.2 รีจิสเตอร์ B(เขียน/อ่านได้)

- set ทำหน้าที่ในการตั้งเวลาเมื่อเขียนให้บิตนี้เป็น "0"จะควบคุมให้นาฬิกาเดินตามปกติ คือมีการปรับเวลาทุกๆ 1วินาที เมื่อเขียนให้บิตนี้เป็นหนึ่งนาฬิกาจะหยุดเดินเพื่อให้ทำการตั้งเวลาได้ บิตนี้จะไม่มีผลต่อการรีเซต

- PIE(PERIODIC INTERRUPT ENABLE) ทำหน้าที่อานาเบิ้ลอินเตอร์รัพแบบเป็นจังหวะ(ตามช่วงเวลาในตารางที่ 1) เมื่อเขียนให้บิตนี้เป็น "1" จะทำให้แฟลคที่แสดงการอินเตอร์รัพเป็นจังหวะ(PF)ถูกเซตซึ่งจะมีผลให้ขา IRQ เป็นลอจิก "0" ได้เมื่อเกิดการอินเตอร์รัพในโหมดนี้ แต่ยังคงทำให้แฟลค PF เปลี่ยนแปลงเมื่อถึงเวลาอินเตอร์รัพ บิต PIE นี้จะถูเคลียเมื่อเกิดการรีเซต

- AIE(ALAM INTERRUPT ENABLE) ทำหน้าที่อานาเบิ้ลการอินเตอร์รัพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า เมื่อถึงเวลาปลูก เมื่อเขียนให้บิตนี้เป็น "1" และจะทำให้แฟลคแสดงการอินเตอร์รัพในโหมดนี้ ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาปลูก(AF) ซึ่งอยู่ในรีจิสเตอร์ C มีผลต่อขา IRQ แต่ถ้าเขียนให้บิตนี้เป็น "0" จะไม่มีผลต่อขา IRQ ขา AIE จะถูกเคลียร์เมื่อเกิดการรีเซ็ต

- UIE(UPDATE ENDED INTERRUPT ENABLE) ทำให้มีการรีเซ็ตการอินเตอร์รัพท์ทุกครั้งที่มีการปรับเวลาหรือทุกวินาที โดยบิตนี้เป็น "1"จะทำให้มีการอินเตอร์รัพท์โดยมีแฟล็ก UF ในรีจิสเตอร์ C แสดงสภาวะการอินเตอร์รัพท์ ในโหมดนี้เมื่อเกิดการรีเซ็ตหรือมีการเขียนบิตเซตให้เป็น "1" จะทำให้บิตนี้ถูกเคลียร์

- SQWE (SQUARE ENABLE) ทำหน้าที่รีเซ็ตความถี่ของ สัญญาณคลื่นสี่เหลี่ยมที่ออกมาทางขา SQW ตามค่าของบิต RS3-RS0 ซึ่งแสดงไว้ในตารางที่ 1 ถ้าเขียนให้บิตนี้เป็น "1" จะยอมให้ความถี่ที่ขา SQW แต่ถ้าเขียนให้บิตนี้เป็น "0"ขา SQW เป็นศูนย์ตลอด บิต SQWE จะถูกเคลียร์เมื่อเกิดการรีเซ็ต

- DM (DATA MODE) ทำหน้าที่กำหนดการอ่านหรือเขียนข้อมูลในหน่วยความจำแสดงเวลาว่าจะให้อยู่ในลักษณะเลขไบนารี(DM="1")หรือ BCD(DM="0")ดังแสดงตัวอย่างไว้ในตารางที่ 2 บิต DM นี้ไม่มีผลต่อการรีเซ็ต

- 24/12 ทำหน้าที่เลือกการแสดงผลเวลาในลักษณะ 24 ชั่วโมง(บิตนี้เป็น "1") หรือ 12 ชั่วโมง(บิตนี้เป็น "0") ซึ่งถ้าเป็น 12 ชั่วโมง จะมีการแสดงผลเวลาด้วยรหัส AM หรือ PM ด้วยบิตที่ 7 ของหน่วยความจำเก็บชั่วโมง(address 04H และ 05H)บิตนี้ไม่มีผลต่อการรีเซ็ต

- DSE นี้ใช้สำหรับแสดงเวลาในประเทศสหรัฐอเมริกาหรือไทย เพราะเวลาในอเมริกาจะมีการปรับเวลาเร็วขึ้น 1 ชั่วโมงในปลายเดือนเมษายน และปรับเวลาช้าลงในปลายเดือนตุลาคม แต่เมื่อใช้ในไทยจะให้บิตนี้เป็น "0" บิตนี้จะไม่มีผลต่อการรีเซ็ต

3.5.3.3 รีจิสเตอร์ C (อ่านได้อย่างเดียว)

ในรีจิสเตอร์ตัวนี้มีบิตที่ใช้งานเพียง 4บิต เท่านั้น คือบิตที่7-4 ทั้ง 4 บิต ทำหน้าที่เป็นแฟล็กแสดงการอินเตอร์รัพท์แต่อ่านได้เพียงอย่างเดียวเท่านั้น แต่ เขียนลงไปไม่ได้ คือแสดงสถานะของตัว RTC เพียงอย่างเดียว

3.5.3.1 IRQF (INTERRUPT REQUEST FLAG) เป็นแฟล็กแสดงการอิน

เตอร์รัพท์ขึ้นแล้วจากโหมดใดโหมดหนึ่งในสามโหมด ซึ่งบิตนี้จะถูกเซตให้เป็นหนึ่งเมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- PF และ PIE เป็น "1" ทั้งคู่หรือ
- AF และ AIE เป็น "1" ทั้งคู่หรือ
- UF และ UIE เป็น "1" ทั้งคู่

เมื่อแฟลก IRQF นี้เป็นหนึ่งจะทำให้ขา IRQ เป็น "0" เพื่อขอการอินเตอร์รัพและแฟลกทั้ง 4 ตัวนี้จะถูกเคลียให้เป็น "0" เมื่อ CPU มาอ่านข้อมูลในรีจิสเตอร์ C นี้ไปหรือเมื่อเกิดการรีเซ็ต

3.5.3.2 PF (PERIODIC INTERRUPT FLAG) เป็นแฟลกที่แสดงการอินเตอร์รัพเมื่อถึงจังหวะการอินเตอร์รัพโดยไมขึ้นอยู่กับบิต PIE เมื่อแฟลกนี้เป็น "1" จะทำให้ขา IORQ เป็น "0" และแฟลก IRQF เป็น "1" ด้วย แฟลก PF นี้จะถูกเคลียให้เป็น "0" โดยฮาร์ดแวร์เมื่อ CPU มาอ่านรีจิสเตอร์ C หรือมีการรีเซ็ต

3.5.3.3 AF (ALARM INTERRUPT FLAG) เป็นแฟลกแสดงการอินเตอร์รัพเมื่อนาฬิกาเดินมาถึงเวลาปลูกที่ตั้งไว้ (ไม่รวมวัน, วันที่, เดือน, ปี) จะทำให้แฟลก AF เป็น "1" และถ้าบิต AIE เป็น "1" อยู่แล้วก็จะทำให้แฟลก IRQF เป็น "1" ไปด้วย และยังทำให้ขา IRQ เป็น "0" เพื่อขออินเตอร์รัพ แฟลก AF นี้จะถูกเคลียโดยฮาร์ดแวร์เมื่อ CPU มาอ่านหรือมีการรีเซ็ต

3.5.3.4 UF (UPDATED-ENDED INTERRUPT FLAG) เป็นแฟลกแสดงการอินเตอร์รัพเมื่อถึงช่วงการเปลี่ยนเวลาทุกวินาที โดยจะเป็น "1" และทำให้เกิดการอินเตอร์รัพเมื่อบิต UIE เป็น "1" บิต UF นี้จะถูกเคลียเมื่อ CPU มาอ่านหรือมีการรีเซ็ต

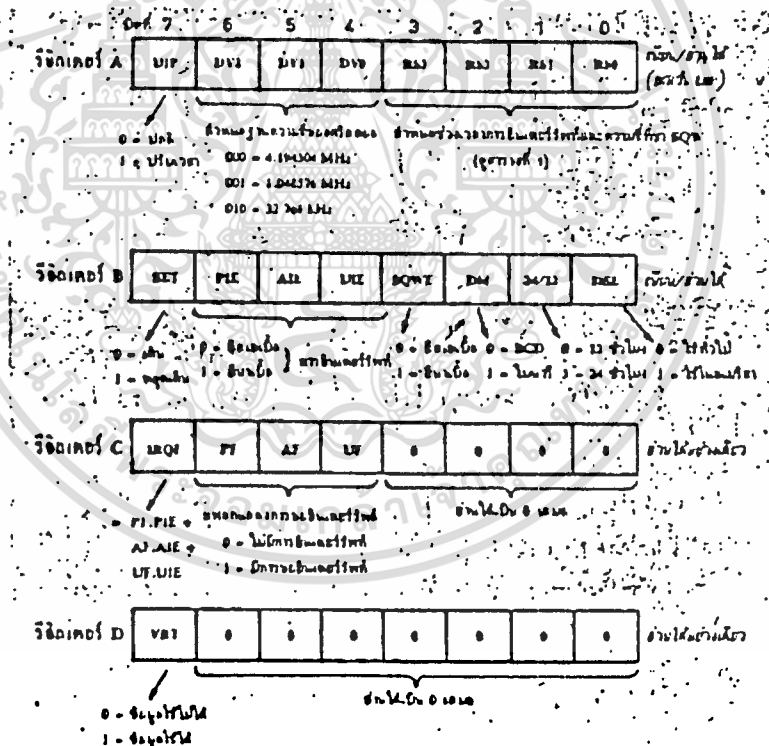
3.5.4 รีจิสเตอร์ D (อ่านได้อย่างเดียว)

รีจิสเตอร์นี้มีบิตที่ใช้งานเพียงบิตเดียวคือบิตที่ 7 อีก 7 บิตไม่มีการใช้งานเมื่ออ่านจะให้ค่า "0" เสมอ

3.5.4.1 VRT (Valid RAM and time) เป็นตัวแสดงสถานะของข้อมูลใน RAM ทั้ง 64 บิต ว่ายังใช้ได้หรือไม่ ซึ่งอาจจะใช้ไม่ได้กรณีที่แรงดันที่ป้อนให้ตัวมันมีค่าต่ำลงจน RTC ไม่สามารถทำงานได้และข้อมูลใน RAM นั้นสูญหายไปแล้วด้วยเหตุนี้จึงมีการใช้ขาไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ps แสดงสถานะของข้อมูลใน RAM โดยต่อขา ps เข้ากับแรงดันไฟเลี้ยงเมื่อขา ps มีแรงดันต่ำมากหรือเมื่อเริ่มป้อนแรงดันไฟเลี้ยงบิต VRT จะมีค่าเป็น '0' เป็นการบอกว่าข้อมูลใน RAM ไม่ถูกต้องแล้ว

ดังนั้นเวลาใช้งาน หลังจาก que CPU ตั้งเวลาเรียบร้อยแล้วจะต้องทำให้บิต VRT เป็น '1' ด้วยการอ่านรีจิสเตอร์ D นี้ ซึ่งจะเป็น '1' ใ้ก็ต่อเมื่อขา ps มีแรงดันสูงแล้ว และถ้าขา ps มีแรงดันต่ำลงอีก บิต VRT ก็เป็น '0' ทำให้ CPU สามารถตรวจสอบสถานะของ RAM ได้และ บิต VRT จะไม่มีผลจากการรีเซ็ต



รูปที่ ๑๓. รายละเอียดและความหมายในแต่ละบิตของรีจิสเตอร์ควบคุม

บิตต่าง ๆ ในรีจิสเตอร์ A				คริสตอลความถี่ 4.194304 MHz หรือ 1.048576 MHz		คริสตอลความถี่ 32.768 kHz	
RS3	RS2	RS1	RS0	ช่วงเวลาการอินเตอร์รัพท์	ความถี่เอาต์พุตจาก SQW	ช่วงเวลาการอินเตอร์รัพท์	ความถี่เอาต์พุตจาก SQW
0	0	0	0	None	None	None	None
0	0	0	1	30 517 μ s	32 768 kHz	3 90625 ms	256 Hz
0	0	1	0	61 035 μ s	16 384 kHz	7 8125 ms	128 Hz
0	0	1	1	122 070 μ s	8 192 kHz	122 070 μ s	8 192 kHz
0	1	0	0	244 141 μ s	4 096 kHz	244 141 μ s	4 096 kHz
0	1	0	1	488 281 μ s	2 048 kHz	488 281 μ s	2 048 kHz
0	1	1	0	976 562 μ s	1 024 kHz	976 562 μ s	1 024 kHz
0	1	1	1	1 953 125 μ s	512 kHz	1 953 125 μ s	512 kHz
1	0	0	0	3 906 250 μ s	256 Hz	3 906 250 μ s	256 Hz
1	0	0	1	7 8125 ms	128 Hz	7 8125 ms	128 Hz
1	0	1	0	15 625 ms	64 Hz	15 625 ms	64 Hz
1	0	1	1	31 25 ms	32 Hz	31 25 ms	32 Hz
1	1	0	0	62 5 ms	16 Hz	62 5 ms	16 Hz
1	1	0	1	125 ms	8 Hz	125 ms	8 Hz
1	1	1	0	250 ms	4 Hz	250 ms	4 Hz
1	1	1	1	500 ms	2 Hz	500 ms	2 Hz

ตารางที่ ๕. แสดงช่วงเวลาการอินเตอร์รัพท์เป็นจังหวะและความถี่เอาต์พุต

ด้านหนึ่ง แยกแยะ	แสดงถึง	ช่วงการ เปลี่ยน แปลง (ฐานสิบ)	ช่วงการเปลี่ยนแปลง		ตัวอย่าง*	
			โหมดไบนารี	โหมด BCD	โหมด ไบนารี	โหมด BCD
0	Seconds	0-59	\$00-13B	\$00-159	15	21
1	Seconds Alarm	0-59	\$00-13B	\$00-159	15	21
2	Minutes	0-59	\$00-13B	\$00-159	3A	58
3	Minutes Alarm	0-59	\$00-13B	\$00-159	3A	58
4	Hours (12 Hour Model)	1-12	\$01-10C (AM) and \$81-18C (PM)	\$01-112 (AM) and \$81-192 (PM)	06	06
	Hours (24 Hour Model)	0-23	\$00-117	\$00-123	06	06
5	Hours Alarm (12 Hour Model)	1-12	\$01-10C (AM) and \$81-18C (PM)	\$01-112 (AM) and \$81-192 (PM)	06	06
	Hours Alarm (24 Hour Model)	0-23	\$00-117	\$00-123	06	06
6	Day of the Week Sunday = 1	1-7	\$01-107	\$01-107	06	06
7	Date of the Month	1-31	\$01-11F	\$01-131	0F	15
8	Month	1-12	\$01-10C	\$01-112	02	02
9	Year	0-99	\$00-163	\$00-199	4F	79

* ตัวอย่างคือเวลา 9 : 58 : 21 วันอังคารที่ 15 กุมภาพันธ์ ปี 1976 (เป็นเวลาในช่วงเช้า: AM)

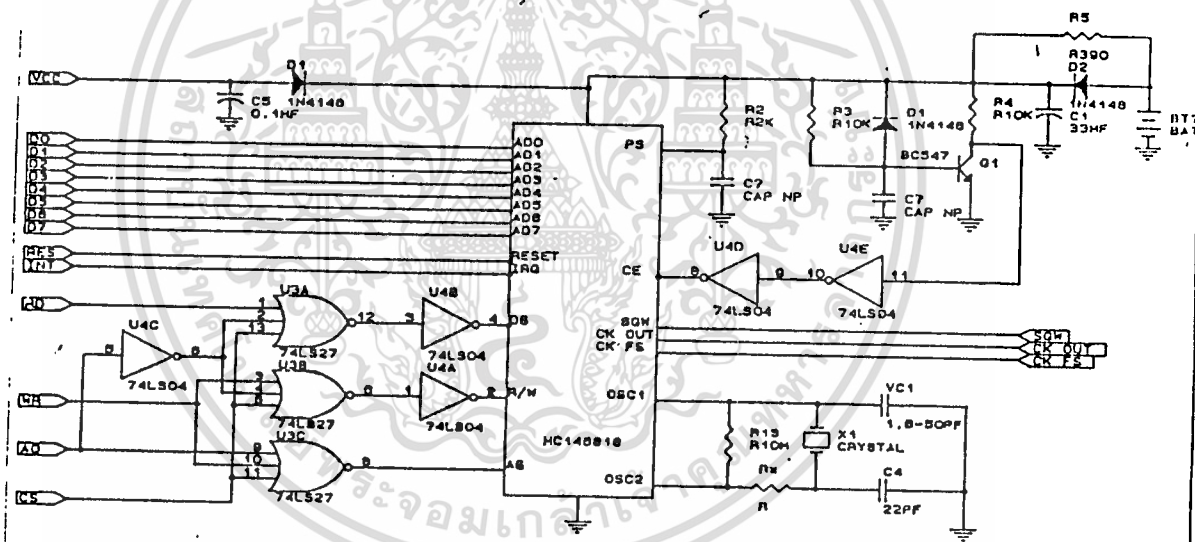
ตารางที่ ๕. แสดงช่วงของข้อมูลที่อ่าน/เขียนได้จากหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครู/อาจารย์เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

3.5.4 การอินเตอร์รัพท์เข้ากับ CPU Z-80

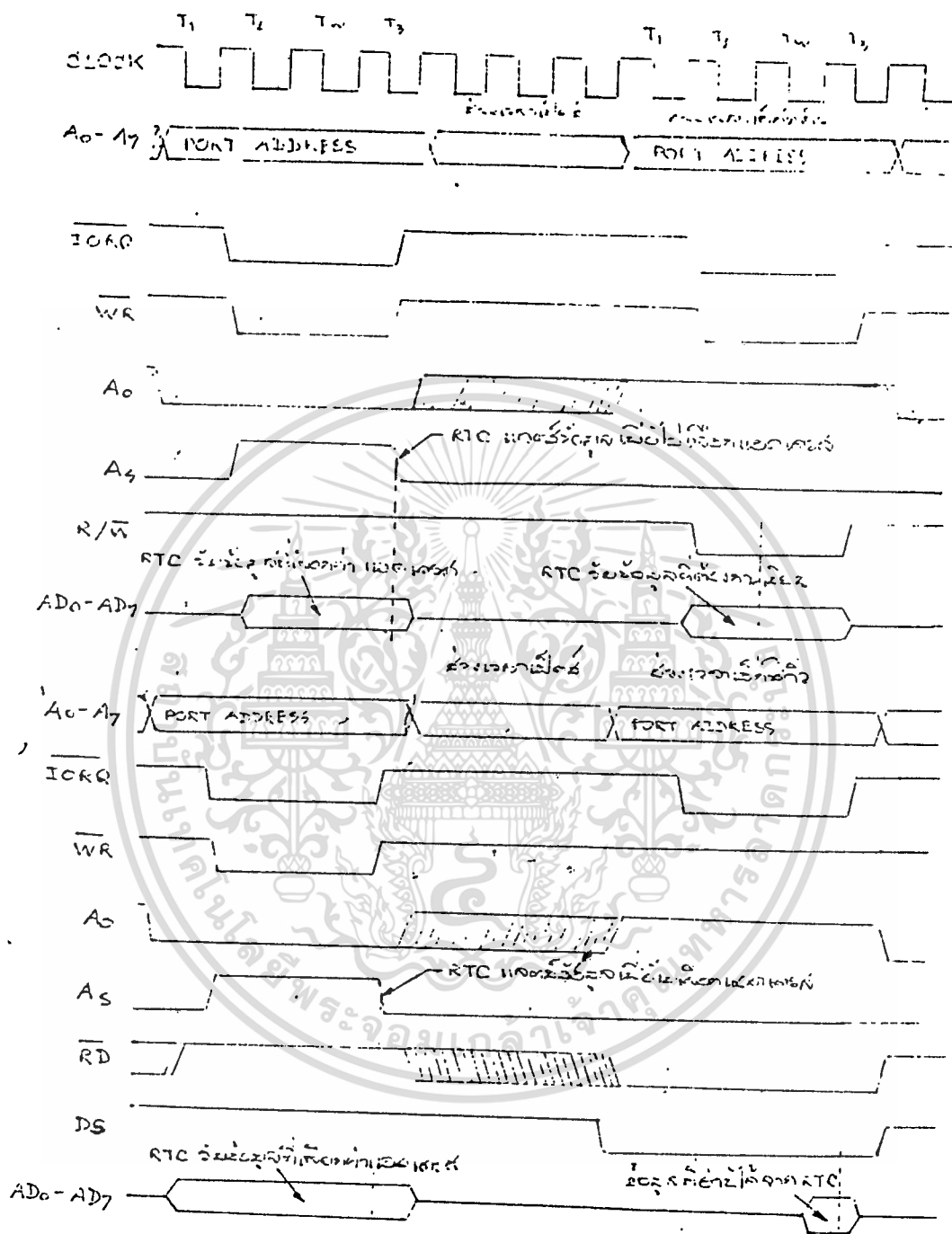
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากว่า MC 146818 ไม่ได้ออกแบบมาเพื่ออินเตอร์เฟสกับ Z-80 โดยตรง ดังนั้นในการที่จะให้ CPU Z-80 สามารถติดต่อ (access) กับ RTC ได้ก็จะต้องอาศัยเทคนิคพิเศษ โดยการใช้ขาสัญญาณอื่นๆ และคำสั่งเข้าช่วย จึงจะสามารถ access ตัว RTC ได้สำเร็จ โดยคำสั่งแรกเป็นการส่งค่า address เพื่อตำแหน่งของ RAM ตัวใดตัวหนึ่งใน 64 byte ตามด้วยคำสั่งที่ 2 ซึ่งจะเป็นคำสั่งเพื่ออ่านหรือเขียนข้อมูลได้ตามที่ต้องการ



รูปที่ ๑๘. วงจรการอินเตอร์เฟส กับ CPU Z-80

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ ๑๕. Timing Diagram

วงจรในรูปแบบการต่อ RTC เพื่อให้ CPU สามารถติดต่อกับ RTC ได้จะใช้ IC อีก 2 ตัวคือ 74LS27 และ 74HC04 สัญญาณ Rd และ Wr ของ Z-80 จำเป็นต้องใช้ IC_{3/1} กับ IC_{4/1} และ IC_{3/2} กับ IC_{4/2} ทำงานเป็น or-gate 3 อินพุตเข้าช่วยเพื่อให้สามารถเข้าถึงตัว RTC ได้โดยผ่านการควบคุมจากสัญญาณ IORQ และ Ao อีกที ในการใช้ Ao เข้าช่วยก็เพื่อให้เกิดการอ้างแอดเดรสถึงตัว RTC 2 ครั้งถึง จะเขียนหรืออ่าน ข้อมูลจาก RTC นี้ ได้โดยการอ้างแอดเดรสครั้งแรก เราจะให้ขา Ao เป็น '0' เพื่อให้เกิดสัญญาณที่ขา As ทำให้ RTC แลatchesค่าแอดเดรสที่ใช้เลือก RAM ในตัว RTC ก่อน แล้วจึงอ้างแอดเดรสครั้งที่ 2 ซึ่งจะทำให้ขา Ds หรือ R/W ตามลำดับ เพื่อให้เกิดการอ่านหรือเขียนข้อมูลที่ต้องการได้

ในวงจรนอกจากจะประกอบด้วยส่วนที่ต่อเพื่อติดต่อกับ CPU แล้วยังมีส่วนที่สนับสนุนการทำงานของ RTC ให้ถูกต้องมากยิ่งขึ้น ประกอบด้วย Rx, C4, Vc1, R13 ทำหน้าที่เป็นตัวปรับแต่งความถี่ให้ถูกต้องเพื่อให้เวลาผิดพลาดน้อยที่สุด ส่วนสวิตช์สัญญาณ CE เพื่อให้ RTC อยู่ในโหมด active เมื่อมีแรงดันไฟ +5 v. ป้อนเข้าตัว RTC และอยู่ในโหมด stand by เมื่อ RTC ใช้แรงดันไฟเลี้ยงจาก battery ซึ่งช่วยให้ประหยัดกระแสไฟเลี้ยงจากแบตเตอรี่ ส่วนนี้ประกอบด้วย Q1, D3, C3, R3, R4 โดยมี IC_{4/4} และ IC_{4/5} เป็นบัฟเฟอร์ช่วยอีกชั้นหนึ่ง เมื่อมีแรงดัน +5 V. จาก sk1 ตัว C3 จะประจุเพื่อรอเวลาสักครู่ที่จะทำให้ Q1 นำกระแสและขา CE เป็นลอจิก '0' ทำนองเดียวกันถ้า +5 V. ถูกตัดออก C3 จะหน่วงเวลานี้เพื่อป้องกันไม่ให้ลอจิกที่ขา CE เปลี่ยนแปลงอย่างกะทันหันจากการเปิด ปิด แรงดัน +5 V. อันอาจทำให้ข้อมูลภายใน RTC เกิดผิดพลาดได้

ส่วนจ่ายแรงดันสำรองจากแบตเตอรี่ ในที่นี้ใช้แบตเตอรี่ ชนิด นิกเกิล-แคดเมียม เพื่อให้มีการประจุกระแสใหม่ได้ เมื่อมีแรงดัน +5 V. โดย R1 และ เมื่อแรงดัน +5 V. ถูกตัดออก แบตเตอรี่ B1 จะจ่ายแรงดันแทนโดยผ่าน D2 เนื่องจากตัว RTC จะใช้กระแสน้อยมากเมื่ออยู่ในโหมด stand by (ขา CE เป็น '1' จึงสามารถให้แบตเตอรี่ขนาด 4.8 V. ความจุ 50-100 mA/H ก็เพียงพอที่จะสำรองให้ RTC ได้หลายวัน

3.5.5 การติดต่อ กับ RTC

เพื่อให้เข้าใจการทำงานของวงจร RTC มากยิ่งขึ้น เราจะยกตัวอย่างการติดต่อกับ RTC โดยในที่นี้จะให้ RTC เป็น PORT โดยกำหนดให้อินพุตพอร์ทเป็นพอร์ทเบอร์ E1 และเอาต์พุตพอร์ทเบอร์ E0 คำสั่งแรกที่ใช้คือ

OUT (PORT E0),ค่าแอดเดรส

คำสั่งนี้จะทำให้สัญญาณ IORQ และ AO ACTIVE เป็น "0" ทำการเปิดเกต IC 3/3 ให้สัญญาณ WR ผ่านเข้าไปที่ขา AS ทำให้ RTC แลทช์ข้อมูลที่เขียนเข้ามา แล้วนำไปเลือกตำแหน่งแอดเดรสของแรมที่แสดงเวลา (7 byte) หรือ (3 byte) หรือ รีจิสเตอร์ควบคุม (4 byte) หรือ แรมข้อมูล (50 byte) และคำสั่งที่ 2 คือ

OUT (PORT E1),DATA

จะเริ่มด้วยช่วงเวลาเฟิร์สของคำสั่งนี้ก่อน เมื่อถึงช่วงเวลาเอ็กซ์ชีวซ์ สัญญาณ IORQ และ AO active เป็น "1" เป็นการเปิดเกต IC 3/1 และ 3/2 ให้สัญญาณ RD และ WR ผ่านไปยังขา DS และ R/W ได้ แต่เนื่องจากคำสั่งนี้เป็นการเขียนข้อมูล สัญญาณ WR จึงแอตตีป ทำให้เกิดการเขียนข้อมูลลงในแอดเดรสที่ถูกกำหนดโดยคำสั่งแรก

การอ่านข้อมูลออกจาก RTC ก็เหมือนกับการเขียนโดยจะต้องอ้างแอดเดรสของแรมที่ต้องการอ่าน แล้วจึงตามด้วยคำสั่งอ่านพอร์ท

IN A, (PORT E1)

คำสั่งนี้จะทำให้ IC 3/1 และ 3/2 ถูกบังคับให้ผ่านสัญญาณ RD และ WR เข้าสู่ขา DS หรือ R/W แต่เนื่องจากเป็นคำสั่งให้อ่านข้อมูล ขา RD เป็น "0" ทำให้ DS เป็น "0" ส่วนขา R/W เป็น "1" จึงได้ข้อมูลจาก RTC ส่งไปสู่ CPU ได้

ตัวอย่างโปรแกรมย่อยสำหรับเขียนและอ่านข้อมูลจาก RTC

```

READ      LD  A,ADD      ; กำหนดแอดดเรสในตัว RTC โดยผ่านรีจิสเตอร์ A
          OUT (EO),A      ; เขียนค่าแอดดเรสให้ RTC
          IN  A,(E1)      ; อ่านข้อมูลที่ได้จาก RTC ไปเก็บไว้ในรีจิสเตอร์ A
          LD  (ADD),A      ; ย้ายข้อมูลจากรีจิสเตอร์ไปเก็บในที่อื่น
          RET

WRITE     LD  A,ADD      ; กำหนดแอดดเรสในตัว RTC โดยผ่านรีจิสเตอร์ A
          OUT (EO),A      ; เขียนค่าแอดดเรสให้ RTC
          LD  A,DATA      ; กำหนดข้อมูลที่ต้องการเขียนโดยผ่านรีจิสเตอร์ A
          OUT (E1),A      ; เขียนข้อมูลให้ RTC
          RET
  
```

3.6 Z-80 กับ 8255

ไอซี 8255 เป็นไอซีต่อเป็นพอร์ตได้ 3 พอร์ต ดังนี้

- PA7 - PA0 เป็นพอร์ต A มีขนาด 8 เส้น สามารถกำหนดให้เป็นได้ทั้งพอร์ตอินพุต และ เอาท์พุท
- PB7 - PB0 เป็นพอร์ต B มีขนาด 8 เส้น สามารถกำหนดให้เป็นได้ทั้งพอร์ตอินพุต และ เอาท์พุท
- PC7 - PC0 เป็นพอร์ต C โดยแยกออกเป็น 2 เส้นคือ พอร์ต c ล่าง PC0 - PC3 มีจำนวน 4 บิตและพอร์ต C บน PC4 - PC7 โดยที่พอร์ต C สามารถกำหนด ได้ว่าให้เป็นอินพุตหรือเอาท์พุทที่แตกต่างกันได้เช่นต้องการ PC0 - PC3 เป็นอินพุต PC4 - PC7 เป็นเอาท์พุท

3.6.1 ความหมายต่างๆของ 8255

D_7-D_0 เป็นขาข้อมูลในการติดต่อกับ CPU ฉะนั้น การเคลื่อนย้ายข้อมูล ระหว่าง Z-80 กับ 8255 ต้องผ่านทางขานี้

CS เป็นสัญญาณอินพุตในการให้ CPU บอกกับ 8255 ว่าตอนนี้ CPU ต้องการติดต่อกับ 8255 ทำให้ CPU สามารถอ่าน และ เขียนจากพอร์ตที่กำหนดของ 8255 ได้

RD เป็นสัญญาณอินพุต เมื่อ CPU ต้องการจะอ่านข้อมูลจากพอร์ตที่กำหนดโดย A_0-A_1 ของ 8255 โดยผ่านทางขา D_7-D_0 .

WR เป็นสัญญาณอินพุตเมื่อ CPU ต้องการจะเขียนข้อมูลลงบนพอร์ตที่กำหนดได้โดย A_0-A_1 ของ 8255 โดยผ่านทางขา D_7-D_0 .

A_0-A_1 เป็นสัญญาณแอดเดรสโดยจะถอดรหัสออกเป็น 4 รหัสเพื่อกำหนด รีจิสเตอร์ ภายใน ที่เชื่อมกับพอร์ต A, B หรือ C ของ 8255

RESET เป็นสัญญาณที่ส่งมาทำการรีเซ็ต 8255 เพื่อเคลียร์สถานะต่างๆของ 8255 เมื่อได้รับการรีเซ็ตก็จะกลับเข้าสู่โหมดอินพุต หรือทุกพอร์ตเป็นพอร์ตอินพุต

$P_{A0}-P_{A7}$ เป็นสายสัญญาณพอร์ตที่เป็นได้ทั้งอินพุต และ เอาท์พุต เป็นพอร์ต A สามารถเลือกพอร์ตนี้ได้โดยกำหนดที่ A_0-A_1

$P_{B0}-P_{B7}$ เป็นสายสัญญาณพอร์ตที่เป็นได้ทั้งอินพุต และ เอาท์พุต เป็นพอร์ต B สามารถเลือก พอร์ตนี้ได้โดยกำหนดที่ A_0-A_1

$P_{C0}-P_{C7}$ เป็นสายสัญญาณพอร์ตที่เป็นได้ทั้งอินพุต และ เอาท์พุต เป็นพอร์ต C สามารถเลือก พอร์ตนี้ได้โดยกำหนดที่ A_0-A_1



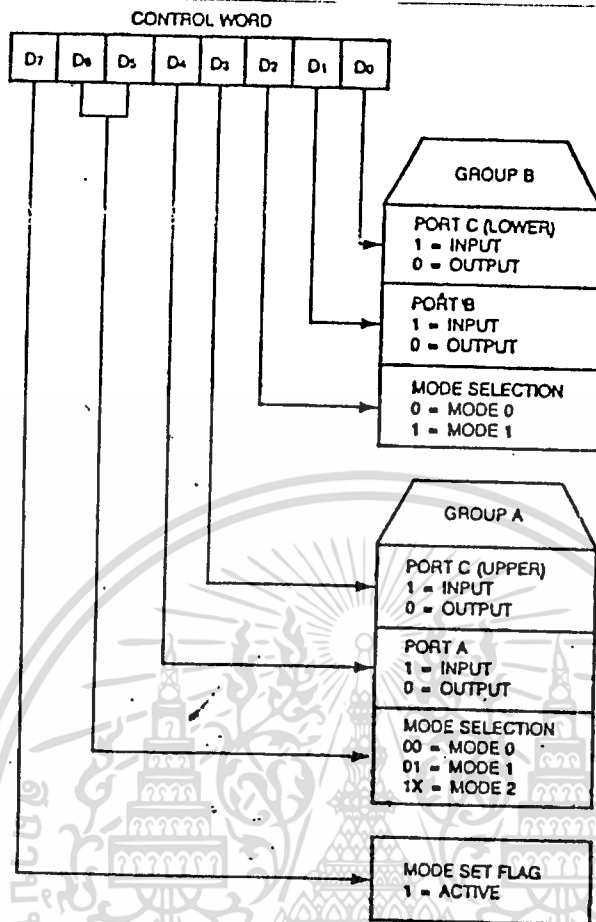
รูปที่ ๑๖. ลักษณะต่างๆของ 8255

จากรูปที่ ๑๑. จะเห็นว่า Z-80 มอง 8255 เป็นพอร์ตหมายเลข 20H-3FH ฉะนั้นเมื่อ Z-80 ต้องการติดต่อกับ 8255 ก็โดยนำสัญญาณที่ถอดรหัสแล้วต่อกับ ขา CS ของ 8255 ส่วนสัญญาณที่ใช้ควบคุมเพื่อเขียนข้อมูลลงบน 8255 คือ WR และสัญญาณที่ใช้ในการอ่านข้อมูลจาก 8255 คือ RD ซึ่งเป็นสัญญาณเส้นเดียวกับที่ใช้กับ การอ่าน การเขียน ข้อมูลกับหน่วยความจำ แต่จะแยกได้ด้วยสัญญาณ IORQ ฉะนั้น IORQ กับ WR ต้องแยกก็ฟพร้อมกันจึงสามารถเขียนข้อมูลลงบนพอร์ตได้ และ IORQ กับ RD ต้องแยกก็ฟพร้อมกันจึงสามารถอ่านข้อมูลจากพอร์ตได้

จากโครงงานนี้เมื่อต่อ Z-80 กับ 8255 แล้วและกำหนดให้พอร์ต ของ 8255 มีลักษณะทำงานดังนี้

- Port A เป็นพอร์ตเอาต์พุตและ เชื่อมกับขาข้อมูลของ LCD
- Port B เป็นพอร์ตเอาต์พุต และ เชื่อมกับขาควบคุมการทำงานของ LCD ขนาด 3 บิตด้านบน และ 5 บิต ด้านล่างที่เหลือจะต่อกับ วงจรเปลี่ยนสัญญาณดิจิทัลเป็นอนาล็อก ออกเป็นสัญญาณเสียง
- Port C เป็นพอร์ตอินพุต และเชื่อมต่อกับตัวถอดรหัสสัญญาณโทรศัพท์เพื่อเก็บในหน่วยความจำ

การใช้งาน 8255 จะต้องส่งรหัสควบคุมเข้าไปยังพอร์ตข้อมูลควบคุมเพื่อควบคุมการทำงานของ 8255 โดยใช้สัญญาณควบคุมพอร์ตหมายเลข 23H และควบคุมในโหมด 0 แต่ละบิตของข้อมูลที่ส่งไปจะมีความหมายในตัวเองดังนี้



รูปที่ ๑๓. แสดงรายละเอียดแต่ละบิตของรีจิสเตอร์ควบคุมของ 8255

จากรูปที่ ๑๓. สามารถอธิบายได้ดังนี้

-บิต D_7 เป็นบิตแสดงรหัสคำสั่งควบคุม ถ้าบิตนี้เป็น "1" หมายถึงต้องการเซทโหมดต่างๆ ของ 8255

-บิต D_6-D_5 เป็นการเลือกโหมดของพอร์ต A ซึ่งมี 3 โหมดคือ 0, 1, 2

-บิต D_4 ถ้ามีค่าเป็น "0" หมายถึงการกำหนดพอร์ต A เป็น เอาต์พุตถ้ามีค่าเป็น "1" จะหมายถึงการกำหนดให้พอร์ต A เป็นอินพุต

-บิต D_3 เป็นบิตที่บอกถึงการเซตของพอร์ต C บน ถ้าเป็น "0" จะทำให้พอร์ต C บนเป็น เอาต์พุต

-บิต D_2 เป็นบิตที่บอกถึงการเซตของพอร์ต B ถ้าเป็น "0" การเลือกพอร์ต B เป็น โหมด 0 และถ้าเป็น "1" หมายถึงการเลือกโหมด 1

-บิต D_1 เป็นการกำหนดอินพุตเอาต์พุตของพอร์ต B ถ้าเป็น "0" หมายถึง เอาต์พุตถ้าเป็น "1" หมายถึงอินพุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

-บิต D_0 เป็นการกำหนด อินพุตเอาต์พุตของพอร์ต C ล่าง ถ้าเป็น "0" หมายถึงเอาต์พุตไปใช้ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ติดเบสลงในบิต 1 และต้องโยงไปยังบิต 0 ในเอาต์พุตถ้าเป็น "1" หมายถึง อินพุต

3.7 DOT MATRIC LCD DISPLAY

เราสามารถที่จะแบ่ง DOT MATRIC LCD MODULE ออกเป็นพวกๆได้ดังนี้

- CHARACTER LCD MODULE
- GRAHPIC LCD MODULE
- SEGMENT DISPLAY TYPE LCD MODULE

โดยในโครงงานนี้เราใช้ GRAPHIC LCD DISPLAY ส่วนประกอบใหญ่ๆสามารถแบ่งได้เป็น

3.7.1 DOT MATRIC LCD เป็นตัวแสดงผลให้เรามองเห็นในลักษณะการปิดเปิดตัวเองกับแสงก็คือ ส่วนของที่เป็นตัวกระจกบรรจุหลัก

3.7.2 DRIVER เป็นเป็นตัวรับสัญญาณจากตัวควบคุมมาขับพลิก LCD อีกทีหนึ่ง โดยมีเบอร์ที่นิยมใช้ใน LCD MODULE เช่น HD44100H, MSM5259

3.7.3 CONTROLLER เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาและจัดการควบคุม LCD MODULE ให้ทำงานแสดงผลต่างๆเช่น การลบจอภาพ การเกิดตัวอักษร เป็นต้น โดยมีเบอร์ IC ที่นิยมใช้กันคือ HD44780 ซึ่งใช้ใน GRAHPIC LCD MODULE

โดยทั่วไปในการทำงานและการใช้งาน LCD MODULE นั้นทำได้โดยการทำความเข้าใจในส่วน CONTROLLER เพราะCONTROLLER หลักการทำงานคล้ายๆกันและใน LCD MODULE แต่ละขนาดจำนวนตัวอักษรหรือจำนวนบรรทัดก็มีหลักการทำงานเหมือนกันหมด IC ที่นิยมมากที่สุดตัวหนึ่งที่เป็น CONTROLLER LCD คือ เบอร์ HD44780 โดยรูปแบบการทำงานของมันได้เป็นมาตรฐานให้กับ CONTROLLER LCD ตัวอื่นๆด้วย

HD44780 เป็นไอซี LSI ตัวหนึ่งใช้ควบคุม LCD โดยแสดงผลในรูปแบบตัวอักษรหรือสัญลักษณ์ต่างๆตัวมันเองสามารถต่อใช้งานแบบ 4บิต หรือ 8บิต ก็ได้ โดยถ้าเราต่อแบบ 4บิต จะต่อใช้งานที่ DB7-DB4 เท่านั้นโดยข้อมูลครั้งแรกที่ส่งนั้น HD44780 จะถือเป็นข้อมูล 4บิต บน และข้อมูลต่อมานั้นเป็นข้อมูล 4บิต ล่างในรูปที่ 1 แสดงถึงส่วนประกอบภายในของ HD44780 ส่วนในรูปที่ 2 แสดง TIMING DIAGRAM การส่งข้อมูลขนาด 4บิต และ ในรูปที่ 3 แสดง TIMING DIAGRAM การ CHECK BUSY FLAG

3.8 การเขียนข้อมูล CHARACTER GENERATER

เราสามารถที่จะเขียนข้อมูลได้โดยกำหนดแอดเดรสของ CG RAM ก่อนโดยเขียนได้ 64 ตำแหน่ง บิต 5 - บิต 0 และเมื่อกำหนดแอดเดรสแล้วก็จะทำการเขียนข้อมูลลงใน CG RAM โดยเป็นลักษณะบิตต่อบิตบนของ 1 ตัวอักษร คือ ขนาด 5×7 นั้นจะใช้ข้อมูล บิต 4 ถึง บิต 0 ต่อ 1 ไบท์เท่านั้น ซึ่ง 1 ตัว อักษรจะใช้ข้อมูล 8 byte ด้วยกัน และเมื่อเขียนข้อมูลลงใน CG RAM เมื่อจะใช้งานก็ทำการเขียนข้อมูลลงใน DD RAM คือ ข้อมูลตำแหน่งในตาราง CHARACTER ที่ตำแหน่ง 00H-07H

3.8.1 ขาดังๆในการต่อใช้งาน HD44780

- RS(REGISTER SELECTION) จะเป็นขาที่ใช้เลือก رجิสเตอร์ภายในที่มีอยู่ 2 ตัว คือ INSTRUCTION REGISTER (IR) และ DATA REGISTER (DR) โดยถ้าเป็น 1 จะเป็นการเลือกข้อมูล และ ถ้าเป็น 0 จะเป็นการเลือกคำสั่ง
- R/W(READ/WRITE) เป็นตัวเลือกว่าจะเขียนหรืออ่านข้อมูลจากตัวไอซี โดยถ้าอ่านข้อมูล =1 และถ้าเขียนข้อมูล =0
- E(ENABLE SIGNAL) เป็นขากำหนดสภาพการเขียนอ่านข้อมูล
- DB0-DB7 เป็นขารับข้อมูลจากไอซี
- VDD ไฟเลี้ยงตัวไอซี
- VSS เป็นขา GND
- VO เป็นขารับแรงดันขับ LCD ให้สว่างหรือมืด

3.8.2 การต่อ DOT MATRIC LCD DISPLAY เข้ากับ CPU

การต่อ DOT MATRIC LCD DISPLAY เข้ากับระบบได้โดยการกำหนดเป็นเอาต์พุตพอร์ตตัวหนึ่ง จากรูปเราจะต่อ 8255 ให้เข้ากับ LCD สัญญาณต่างๆที่ให้การควบคุมจะถูกจำลองขึ้นมาโดยการใช้พอร์ต A และ พอร์ต B ซึ่งพอร์ต A นั้นให้ค่าตัวพอร์ท และพอร์ต B นั้นเราให้เป็นสัญญาณควบคุม เมื่อเริ่มป้อนไฟให้ HD44780 ก็จะมีการรีเซ็ตตัวเองโดยใช้เวลาประมาณ 10ms หลังจากไฟ V_{DD} ถึง 4.5V แล้ว หลังจากนั้นจะเซตตัวเองดังนี้

- DISPLAY CLEAR ทำการลบข้อมูลจอภาพ LCD

- FUNCTION SET ทำการเซตค่าภายในดังนี้

DL = 1:เป็นการเซตให้เป็นการติดต่อแบบ 8 บิต

N = 0:เซตให้เป็น 1 บรรทัดการแสดงผล

F = 0:จำนวนจุดต่อหนึ่งตัวอักษรเท่ากับ 5*7

- DISPLAY ON/OFF

D = 0 :display off

C = 0 :cursor off

B = 0 :blink off

- ENTRY MODE SET

I/D = 1:+1 (เพิ่มค่าเคาน์เตอร์ขึ้น 1)

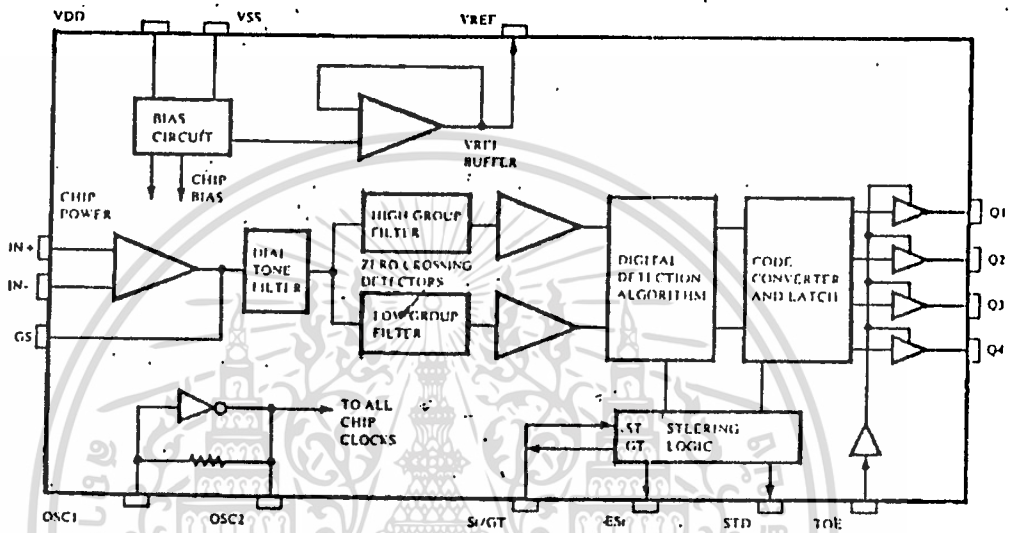
S = 0: no shift

เมื่อเริ่มเปิดเครื่องทำงานแล้วก็ต้องส่งคำสั่งควบคุมให้มันเริ่มทำงานตั้งนั้นขั้นตอนในรูปที่

4 ส่วนในตารางที่ 1 นั้นเป็นการรวมคำสั่งที่ให้การควบคุมการทำงานของ DOT MATRIC LCD DISPLAY ให้เป็นไปตามที่ต้องการได้ โดยมีทั้งหมด 11 คำสั่งด้วยกัน

3.9 MT 8870 INTEGRATED DTMF RECEIVER

3.9 MT 8870 INTEGRATED DTMF RECEIVER



รูปที่ ๑๘. แสดงโครงสร้างภายในของ MT8870

3.9.1 คุณสมบัติ ของ MT8870

- เป็นตัวรับและถอดรหัสความถี่
- กินไฟน้อย ใช้ไฟเลี้ยงระดับเดียวกับ TTL
- สามารถตั้งอัตราขยายภายในตัวไอซีได้
- สามารถปรับการคัทโคมได้
- เป็นไอซีคุณภาพสูง

ชนิดคูปุ่ม (DTMF) ให้เป็นตัวเลข BCD ขนาด 4 บิต เริ่มตั้งแต่ 0000-1111 ให้รหัสตามตารางข้างล่าง โดยให้ร่วมกับคริสตอล 3.579 MHz เท่านั้น

F_{low}	F_{high}	No.	TOE	Q_4	Q_3	Q_2	Q_1
697	1209	1	H	0	0	0	1
697	1336	2	H	0	0	1	0
697	1477	3	H	0	0	1	1
770	1209	4	H	0	1	0	0
770	1336	5	H	0	1	0	1
770	1477	6	H	0	1	1	0
852	1209	7	H	0	1	1	1
852	1336	8	H	1	0	0	0
852	1477	9	H	1	0	0	1
941	1339	0	H	1	0	1	0

F_{low}	F_{high}	No.	TOE	Q_4	Q_3	Q_2	Q_1
941	1209	*	H	1	0	1	1
941	1477	#	H	1	1	0	0
697	1633	A	H	1	1	0	1
770	1633	B	H	1	1	1	0
852	1633	C	H	1	1	1	1
941	1633	D	H	0	0	0	0
-	-	ANY	L	Z	Z	Z	Z

ตารางที่ ๖. แสดงค่าที่ถอดรหัสได้จากความถี่ต่างๆ

หมายเหตุ

TOE เป็นอินพุท ขา 10 ของ IC MT8870

Z High Impedance

F_{low} ความถี่ต่ำ

F_{high} ความถี่สูง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

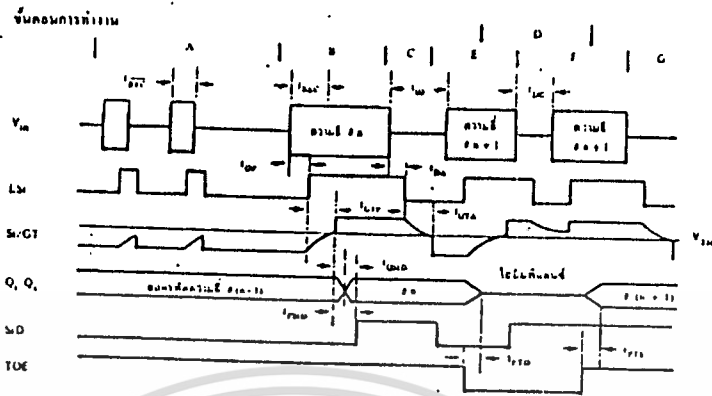
3.9.2 ฟังก์ชันการทำงานภายในสามารถแบ่งได้ 5 ส่วนสำคัญ

- ภาคกรองความถี่ (filter section)
- ภาคถอดรหัส (decoder section)
- ภาคตรวจสอบสัญญาณ (sterring circuit)
- ภาคขยายสัญญาณความแตกต่าง (differential input)
- ภาคกำเนิดความถี่ (oscillator)

ภาคกรองความถี่ โดยภาคนี้อจะเป็นตัวแยก สัญญาณ DTMF ที่เข้ามาออกเป็น 2 กลุ่มของความถี่ คือ กลุ่มความถี่สูงและกลุ่มความถี่ต่ำ โดยใช้วงจรกรองแถบความถี่อันดับ 6 ชนิดสวิดคาปาซิเตอร์ (six-order switched capacitor band pass filter)

ภาคถอดรหัส ความถี่ DTMF ที่ถูกกรองเรียบร้อยแล้วจะผ่านเข้าวงจรถอดรหัสความถี่ออกเป็นตัวเลขโดยใช้เทคนิคการนับแบบดิจิทัล

ภาคตรวจสอบสัญญาณ เป็นการตรวจสอบความถี่ที่เข้ามาว่ามีระยะเวลาตามที่กำหนดหรือไม่โดยสังเกตจากระยะเวลาการกดปุ่มโทรศัพท์ซึ่งต้องกดปุ่มให้มีความถี่ออกมาเป็นช่วงเวลาพอสมควรมิฉะนั้นวงจรส่วนนี้จะไม่รับโดยถือว่าสัญญาณนั้นไม่ถูกต้อง ซึ่งช่วงเวลาสามารถตั้งโดยเปลี่ยน ค่า RC สัญญาณที่ขา EST จะเป็น "1" นานใกล้เคียงกับระยะเวลาที่มีความถี่ DTMF เข้ามา



อธิบายขั้นตอนการทำงาน

- A - ตรวจพบความถี่เข้ามา แต่กลับเวลาไม่ถูกต้อง เอาต์พุตไม่เปลี่ยน
- B - ความถี่ # n ถูกตรวจพบและมีคาบเวลาที่ถูกต้อง ความถี่ถูกถอดรหัส และแสดงไว้ที่เอาต์พุต
- C - จนความถี่ # n ช่วงห่างถูกต้อง เอาต์พุตยังคงแสดงอยู่จนกว่าจะได้รับความถี่ที่สูงกว่า
- D - เอาต์พุตเปลี่ยนเป็นไอเอ็มพีแดนซ์
- E - ความถี่ # n + 1 ถูกตรวจพบ คาบเวลาถูกต้อง ความถี่ถูกถอดรหัสและแสดงไว้ที่เอาต์พุต
- F - ความถี่ # n + 1 หายไป ช่วงห่างไม่ถูกต้อง เอาต์พุตยังคงแสดงอยู่
- G - จนความถี่ # n + 1 ช่วงห่างถูกต้อง เอาต์พุตยังคงแสดงอยู่จนกว่าจะได้รับความถี่ที่สูงกว่า

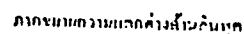
รูปที่ ๑๔. timing diagram

ภาคขยายสัญญาณความแตกต่าง วงจรทางด้านอินพุตของ IC เบอร์ MT8870 เป็นภาคขยายออปแอมป์ ที่สามารถปรับอัตราขยายออปแอมป์ที่สามารถปรับอัตราขยาย โดยต่อวงจรภายนอกเพิ่มเข้าไป ซึ่งค่าต่างๆที่จะนำมาต่อนั้นสามารถหาได้ดังนี้

$$\text{อัตราขยาย (AV diff)} = R_2/R_1$$

$$\text{อินพุตอิมพีแดนซ์ (Zin diff)} = 2 \sqrt{(R_1^2 + (1/wc)^2)}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



$$C_1 = C_2 = 10 \text{ nF}$$

$$R_1 = R_2 = R_3 = 100\text{K}\Omega \quad \text{ค่าผิดพลาด} \pm 1\%$$

$R_1 = 60\text{ K}\Omega$, $R_2 = 17.5\text{ K}\Omega$ ค่าผิดพลาด $\pm 5\%$

$$R_3 = \frac{R_2 R_1}{R_1 + R_2}$$

$$\text{อัตราผลตอบแทน (A_v diff)} = \frac{R_s}{R_s}$$

อัมพุกฉิมทิศแดนขั้ว

$$(Z_{\text{INDIFF}}) = 2 \sqrt{z_1^2 + \left(\frac{1}{\omega C}\right)^2}$$

รูปที่ ๒๐. ลักษณะการต่อทางด้านอินพุต

อัตราขยายแรงดัน = $100\text{K}/100\text{K}$

$$f = 697\text{Hz} - 1633\text{Hz}$$

$$W = 2\pi f C$$

$$W_1 = 0.43 \cdot 10^{-3}$$

$$wh = 1.026 \times 10^{-3}$$

$$Z_{in,f} = 200.052 \text{ K}$$

$$Z_{in}, f_p = 200.009 \text{ K}$$

จะเห็นว่าเมื่อเราเลือกอัตราขยายเท่ากับ 1 ใช้ค่า $R1 = 100K$ คิดในช่วงความถี่ตั้งแต่ 697 ถึง 1633 Hz ได้ค่าอินพุตอิมพีแดนซ์ที่ใกล้เคียงกันถือว่าเท่ากัน จึงเลือกใช้งานจริงนี้ เพราะมีค่าอินพุตอิมพีแดนซ์สูงและเกือบคงที่ เพื่อป้องกันวงจรที่จะนำไปพ่วงนี้ดึงกระแสใน line ขององค์การโทรศัพท์มากเกินไป ซึ่งจะก่อให้เกิดผลกระทบต่อองค์การโทรศัพท์ด้วย

3.10 หลักการของวงจรดีเทคสัญญาณโทรศัพท์ .

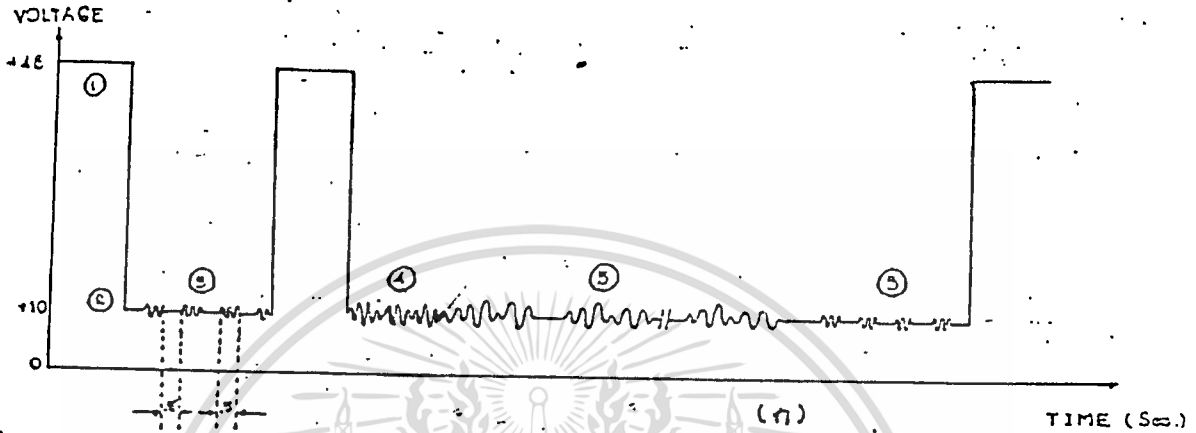
ระบบการต่อของเครื่องชุมสายโทรศัพท์และสัญญาณที่คู่สายโทรศัพท์

-ระบบต่อด้านผู้เรียก

เมื่อผู้เรียกยกหูขึ้นเพื่อที่จะทำการเรียกจะทำให้ DC VOLTAGE ที่คู่สายโทรศัพท์ เปลี่ยนจาก 48V(1) เป็น 10V(2) ดังรูปที่ 21 เครื่องชุมสายโทรศัพท์จะรู้ว่าเป็นการเริ่มต้นการเรียกก็จะส่งสัญญาณให้หมุนหรือกดปุ่ม DT(DUAL TONE)(4) ไปยังผู้เรียก (กรณี OGT: OUT GOING TRUCK ถ้าไม่ว่างก็จะส่งสัญญาณไม่ว่าง :BT BUSY TONE(A) ไปยังผู้เรียกทำให้ผู้เรียกวางหูและเริ่มทำการเรียกใหม่) เมื่อผู้เรียกได้ยินสัญญาณให้หมุน DT ก็จะทำการกดหมายเลขเพื่อส่งสัญญาณ DTMF=DUAL TONE MULTI FREQUENCY(5) เมื่อรับสัญญาณตัวแรกชุมสายก็จะทำการตัดสัญญาณ DT ทันทีเมื่อชุมสายได้รับสัญญาณครบ วงจรคู่สายโทรศัพท์ก็จะทำการแปลงรหัสตัวเลขระบุปลายทางจากรหัสชุมสายที่กดหมายเลขมา เมื่อรู้ตำแหน่งของผู้รับแล้วเครื่องชุมสายโทรศัพท์ก็จะทำการจองคู่สายของผู้เรียก (OGT) และผู้รับ (ICT) : INCOMING TRUNK แล้ว ส่งสัญญาณกริ่งเรียกกลับ :RGT (RING BACK TONE) ไปยังผู้เรียก และในขณะเดียวกันวงจรคู่สายส่งสัญญาณกริ่งเรียก : RGT ไปยังผู้รับ [กรณีผู้รับกำลังใช้โทรศัพท์ หรือ ICT ไม่ว่าง เครื่องชุมสายโทรศัพท์ก็จะส่งสัญญาณไม่ว่าง BT (BUSY TONE) ไปยังผู้เรียกเพื่อให้ทำการวางหู และเริ่มทำการเรียกใหม่ (ก)]

ในกรณีที่ปลายทางว่าง ผู้เรียกจะได้รับสัญญาณ RBT ในเงื่อนไขนี้เรานำเอาสัญญาณนี้ มาผ่านวงจร DETECT เพื่อตรวจสอบ RBT ถ้าเมื่อใดที่สัญญาณ RBT หยุด ให้ทำการคิดเงินทันที จะกล่าวถึงรายละเอียดต่อไป

เมื่อผู้รับมาตอบเรียก สัญญาณตอบรับจะถูกส่งไปยังเครื่องชุมสายโทรศัพท์ จะตัดสัญญาณกริ่งเรียก RGT ด้านผู้รับหรือ ICT และ ยกเลิกสัญญาณเรียกกลับ RBT ด้านผู้เรียก (OGT) และ ทำให้ทางผ่านระหว่าง RBT และ ผู้เรียกว่าง ขณะเดียวกันก็จะสร้างทางผ่านด้านผู้รับ หรือ ICT การสนทนาจึงจะสามารถเริ่มต้นได้ (2) รูปที่ 22 (สถานะว่างหูของผู้เรียก)



รูปที่ ๒๐ สัญญาณเมื่อผู้เรียกฯไม่สำเร็จ

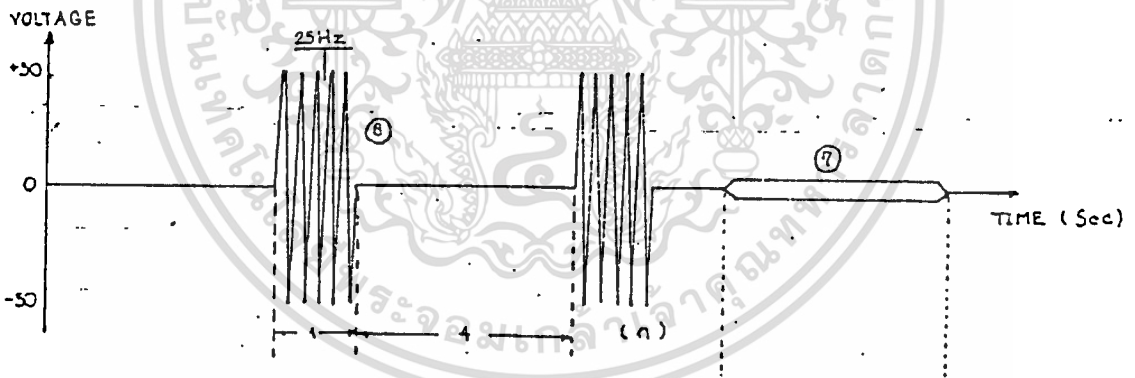


รูปที่ ๒๑ สัญญาณเมื่อผู้เรียกฯสำเร็จ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-ระบบการเรียกด้านผู้รับ

เมื่อผู้รับถูกเรียกจากผู้เรียก เครื่องชุมสายโทรศัพท์จะส่งสัญญาณกริ่งเรียก RGT (3) ขนาด 100 V_{AC} ไปรับผู้รับเพื่อทำการเรียกผู้รับ รูปที่ 23 (ถ้าผู้รับไม่ตอบการรับ สัญญาณกริ่งเรียกก็จะดำเนินอยู่จนกว่าวงจรคู่สายจะทำการตัดสัญญาณเอง เมื่อไม่มีการตอบรับ และจะส่งสัญญาณ BT ไปให้ผู้เรียกเพื่อที่จะให้รู้ว่าปลายทางไม่รับสายให้ทำการเรียกใหม่) เมื่อผู้รับตอบรับการเรียกจะทำให้ DC VOLTAGE เปลี่ยนจาก $48\text{V}(1)$ เป็น $10\text{V}(2)$ ทำให้วงจรคู่สายตัดสัญญาณกริ่งเรียก RGT หรือ ICT และ สัญญาณเรียกกลับด้านผู้เรียก RBT (RING BACK TONE) พร้อมทั้งสร้างทางผ่านระหว่างผู้รับหรือ ICT และ ผู้เรียก OGT การสนทนาจึงสามารถเริ่มต้นได้(7) รูปที่ 24 สถานะวางหูของผู้เรียก จะเลิกทางเสียงผู้พูดผ่านและทำให้ผู้รับวางหูตามวงจรคู่สายจะตรวจรู้ว่าเป็นการเลิกสนทนาและเลิกติดต่อ



รูปที่ ๒๓ สัญญาณทางด้าน AC



รูปที่ ๑๘ สัญญาณทางด้าน DC

3.10.1 หลักการทำงานของวงจร Detector

เนื่องจากคู่สายโทรศัพท์ในระบบ DTMF ในกรณีที่คู่สายปลายทางยกหูโทรศัพท์จะไม่ทำการ revers line มาให้ ดังนั้นจึงจำเป็นต้องสร้างวงจร detect เพื่อทำการตรวจสอบให้รู้ว่า เมื่อใดที่คู่สายปลายทางยกหูโทรศัพท์แล้ว

จากวงจรดังรูป sw1 ทำหน้าที่เป็น switch ปิด/เปิด (handset) ในโทรศัพท์ D1 ทำหน้าที่เป็น Bridge diode ทำหน้าที่เรียงกระแส D_{z3} D_{z4} ทำหน้าที่ จำกัด Voltage ตกคร่อม ให้มีค่า 66V เพื่อป้องกัน ความเสียหายอันเกิดจากไฟเกิน

อุปกรณ์ Q_1 , R_2 , R_3 , D_3 , U_{9A} , U_{9F} ทำหน้าที่เป็นวงจรตรวจสอบ handset ถ้ามีการยกหู output ของ U_{9A} จะเป็น high แต่ถ้าวางหูโทรศัพท์จะเป็น low สัญญาณการยกจะใช้เป็น สัญญาณควบคุม IC 8870 ถ้าเป็น high IC8870 จะยอมรับ tone แต่ถ้าเป็น low IC8870 ไม่ยอมรับ tone

U_{9A} , U_{9B} , U_{7A} , U_{7B} , R , C ที่ต่ออยู่กับขา 7 และ 6 ทำหน้าที่เป็น วงจร mono-stable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

U_{2C} ทำหน้าที่เป็นวงจร band pass filter 2 order $f_0=400\text{Hz}$

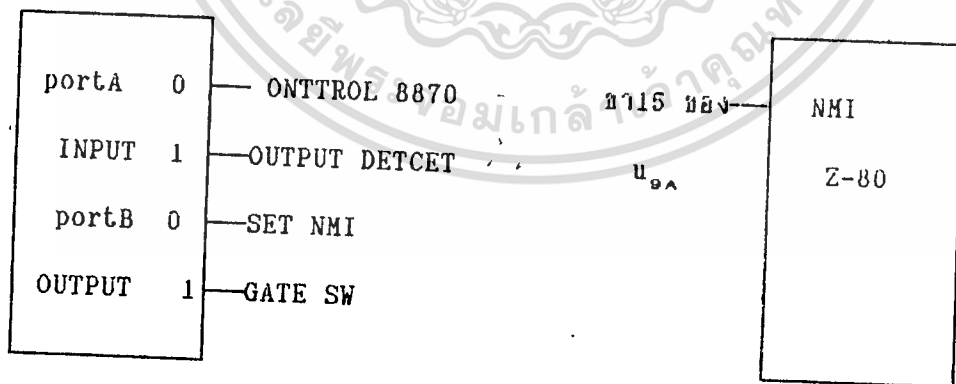
U_{5A}, U_{5B}, U_{9A} (JK-FIPFLOP)

UIC ทำหน้าที่เป็น electronics switch ทำหน้าที่ปิดเปิด สัญญาณจากคู่สาย เข้าสู่วงจร filter และส่วน detect ring back tone

U_{6A}, U_{6B} = or gate และ U_{4B} =not gate

เริ่มการทำงาน ผู้เรียกเริ่มยกโทรศัพท์ sw1 on ระดับ voltage ในคู่สายจะลดลง จาก 48v เป็น 10v D_2 จะ cut off และ Q_1 off ,input u_{9A} เป็นระดับ high และส่งผลให้ U_{9A} ที่ OUTPUT เป็นระดับ high =1 เป็นผลให้ยกเลิกการ clear ของ mono-stable, flipflop และ 8870 พร้อมทั้งจะรับหมายเลขจากคู่สายโทรศัพท์แล้ว ในช่วงเวลาต่อจากนี้ ผู้เรียกจะทำการกดหมายเลขโทรศัพท์จนครบ ทุกครั้งที่มีการกดหมายเลข จะทำให้เกิดการ ทริก U_{9A} ทุกครั้งเพื่อทำให้ output (NMI) ของ Z-80 active และ เมื่อมีการ active NMI ก็จะไปทำงานตามโปรแกรมบริการ และ เมื่อสิ้นสุดโปรแกรมก็จะทำการถอนการ NMI เพื่อให้รอรับการ interrupt ต่อไป

การนำเอาสัญญาณต่าง ๆ ของวงจร detect ไปใช้งานมีการต่อดังรูป โดยการใช้ 8255 PORT A ,PORT B ตำแหน่ง A0H-A3H



สภาวะเริ่มแรกก่อนเรียกใช้คู่สาย port A = 00 เมื่อผู้เรียกยกหูจะเปลี่ยนจาก 00 เป็น 01 เมื่อทำการกดหมายเลข NMI จะแอกติฟ 0 โปรแกรมก็จะทำการทดสอบ port A ว่าเป็น 01 หรือไม่ ถ้าใช่ก็จะทำการเก็บหมายเลข และจะถอน NMI และการทำงานจะเป็นไปตาม loop ข้างต้นไปจนครบจำนวนหมายเลข แล้วโปรแกรมจะทำการเปิด gate switch พร้อมทั้งถอน NMI ในช่วงนี้สัญญาณจะสามารถผ่านเข้าสู่ Filter, phase loop, JK- Flipflop และ mono-stable ทั้งหมดนี้ จะทำหน้าที่เป็นวงจร check ring back tone เมื่อใดที่ ring back tone หด ซึ่งหมายถึงผู้เรียกยกหูโทรศัพท์ จะเกิด NMI และที่ port A จะเปลี่ยน 01 เป็น 11 ซึ่งจะเป็นการบอกว่าเริ่มคิดเงิน เมื่อทำการคิดเงินแล้ว ก็จะถอน NMI และออกจากโปรแกรมไป

แต่หากก่อนที่ ring back tone จะหยุด(ยังไม่เกิด สภาวะ 11) หรือก่อนที่จะกดหมายเลขครบ ผู้เรียกวางหูโทรศัพท์เสียก่อน โปรแกรมจะทำการ reset counter memory, memory ที่ใช้เก็บหมายเลขโทรศัพท์รวมทั้ง memory ที่ใช้เก็บสภาวะที่แสดงการผ่านขั้นตอน ring back tone การทำงานใน memory นี้คือ หากเกิดสภาวะ 11 ก่อน จะมีค่าเท่ากับ 01 ถ้าหากไม่เกิดสภาวะ 11 คือ ring back tone ไม่หยุดเนื่องจาก ผู้เรียกไม่ยกหูโทรศัพท์ แล้วเกิดสภาวะ 00 คือ สภาวะวางหูโทรศัพท์ของผู้เรียก จะเท่ากับ 00 โปรแกรมจะทำการ reset memory ที่ใช้บันทึกหมายเลขและข้อมูลที่ใช้ในการคิดเงิน

แต่ถ้าหากผ่านสภาวะ 01→11 และสภาวะที่ตามมาคือ 00 นั้นหมายถึงผู้เรียกยกหูโทรศัพท์และกดหมายเลขครบ ซึ่งโปรแกรมจะทำการเปิด gate ให้ switch ring back tone ผ่าน detect จนผู้รับยกหูโทรศัพท์ สภาวะจะเปลี่ยนจาก 11→00 สภาวะ 00 นี้คือ สภาวะ stop ในสภาวะนี้ โปรแกรมจะทำการเก็บค่าเวลาแรก แล้วนำจำนวนเวลาการใช้โทรศัพท์นำไปคิดเงินโดย rate ค่าบริการ (บาท/นาที) แล้วจะให้จำนวนเงินทั้งหมดออกมา จากนั้นก็จะนำไปบันทึกเก็บไว้ใน memory

โปรแกรมควบคุมระบบ และ การทำงานของเครื่องคิดเงินค่าให้บริการโทรศัพท์

4.1 โปรแกรมควบคุมการแสดงผลบนจอ LCD

ก่อนที่จะเขียนโปรแกรมสำหรับควบคุมการทำงานของจอ LCD ก็จะต้องให้ค่าเริ่มต้น LCD ก่อน ซึ่งมีลำดับ ดังตารางข้างล่าง

เมื่อให้ค่าเริ่มต้นแก่ LCD แล้วก็ทำการเขียนโปรแกรมตามลำดับดังนี้

โปรแกรม DISPLAY

INPUT=HL (ตำแหน่ง ADDRESS ที่จะให้แสดง)

การให้ LD HL, ADDRESS

CALL DISPLAY

4.2 โปรแกรมควบคุมการรับข้อมูลทาง KEY BOARD

การทำงานของ Key board จะอาศัยอินเตอร์รัพท์ โหมด 1 โดยเมื่อมีสัญญาณมา Active เข้ามา INT ของ Z-80 ทำให้ โปรแกรมกระโดดไปทำงานตำแหน่ง 038H เพราะฉะนั้นทุกครั้งที่กด Key Keyboard ก็จะส่งสัญญาณไปอินเตอร์รัพท์ Z-80

OUTPUT=(KEYIN) , KEYIN เป็นที่เก็บรหัสที่ส่งมาจาก keyboard
=(DATAKEY) , DATAKEY เป็นที่เก็บรหัส ASCII

4.3 โปรแกรมควบคุม REAL TIME CLOCK

การควบคุมการทำงานของ Real time clock อาศัย Port เบอร์ 2 (40H-5FH) เป็นตัวควบคุมการทำงานดังตารางข้างล่าง

4.4 โปรแกรมควบคุมการเช็คสัญญาณด้านผู้รับ และ คิดเงิน

การทำงานของตัว Detect signal จะกระทำร่วมกับ IC 8255 พอร์ทเบอร์ 5 (A0H-B0H) โดยให้บริการอินเตอร์รัพท์แบบนอนมาร์คเคเบิล เมื่อ Active จะกระโดดไปที่ตำแหน่ง 066H มีลำดับการทำงานดังโฟลชาร์ทข้างล่าง

บทที่ 5

บทสรุป

สำหรับโครงงานนี้แบ่งออกเป็นสองส่วนใหญ่ๆคือ

1. ฮาร์ดแวร์
2. ซอฟต์แวร์

โดยในภาคแรกได้ดำเนินงานในส่วนของฮาร์ดแวร์ ซึ่งในขณะนี้ได้ดำเนินงานได้เสร็จ แล้วดังนี้

-ภาคควบคุมและประมวลผล

-ภาค DETECT SIGNALS

-ภาครับ KEYBOARD

-ภาคแสดงผล

-ภาคจ่ายพลังงาน @

ซึ่งในช่วงนี้กำลังอยู่ในขั้นตอนการทำงานของซอฟต์แวร์สำหรับควบคุมการทำงานของระบบฮาร์ดแวร์ เพื่อแสดงผลการคิดเงินในการโทรศัพท์แต่ละครั้ง

ปัญหาที่เกิดขึ้น

-เกิดจากการผิดพลาดจากการ WIREWRAP

-เกิดจากการรบกวนของสัญญาณความถี่สูงอันเกิดจากภายในระบบเองและจากภายนอก

แนวทางพัฒนาต่อไป

-พัฒนาทางด้านซอฟต์แวร์ ทั้งซอฟต์แวร์ควบคุมระบบและซอฟต์แวร์ด้านบริการต่างๆ

-ทำการออกแบบลายวงจรของวงจรทั้งระบบ แทนการ WIREWRAP เพื่อแก้ปัญหาการรบกวนของความถี่สูง

-ควรจะมีการพัฒนาต่อไปโดยออกแบบให้สามารถติดต่อกับเครื่องพิมพ์ได้ เพื่อที่จะ

นำข้อมูลออกมาเป็นหลักฐานในการเก็บค่าบริการได้ X

ภาคผนวก

ภาคผนวก ก. โปรแกรม MONITOR

ภาคผนวก ข. วงจรรวมทั้งหมด

ภาคผนวก ค. DATA SHEET Z-80, LCD, 8279



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

000000          CPU "Z80.TBL" ;defind table for z80

000000          HOF "BIN16"    ;fomat. of file

000000          ORG 0000H

000000 AF      POWER:      XOR A          ;delay for ready all component

000001 3D      POWERI:     DEC A          ;

000002 00          NOP          ;

000003 20FC          JR NZ,POWERI  ;

000005 31FFFF          LD SP,OFFFHH ;defind local for push pop stack

000008 06FF          LD B,OFFH      ;

00000A 10FE      LOOP:     DJNZ LOOP    ;delay again

00000C C39E01          JP START      ; go to main program for running

00000F 00000000000000BLANK: DFB 0,0,0,0,0,0,0,0,0,0 ;defind blank for

000019 0000000000000000 DFB 0,0,0,0,0,0,0,0,0,0 ;load data from

000023 0000000000000000 DFB 0,0,0,0,0,0,0,0,0,0 ;pc to et board

00002D 0000000000000000 DFB 0,0,0,0,0,0,0,0,0,0

000038          ORG 0038H ;location for interrupt keyboard to

000038 CDEF02          CALL KEYBOARD ;receipt data for keyboard to Re

00003B 57          LD D,A          ;move data in Reg a to Reg d for

00003C 79          LD A,C          ;ADD DD RAM is defind Reg c

00003D CDC802          CALL GOTO    ;set the DD Ram,DD Ram is sent a

000040 CDB702          CALL EPLUSE  ;call program enable display LCD

000043 CDD502          CALL WRBYTE  ;write data to LCD size byte

000046 0C          INC C          ;increase DD RAM for rewrite

000047 C9          RET

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

000048 00000000000000BLANK1:      DFB 0,0,0,0,0,0,0,0,0,0
000052 0000000000000000      DFB 0,0,0,0,0,0,0,0,0,0
00005C 0000000000000000      DFB 0,0,0,0,0,0,0,0,0,0
000066                                ORG 0066H
000066 F5                        PUSH AF
000067 C5                        PUSH BC
000068 D5                        PUSH DE
000069 E5                        PUSH HL

;*****
;SUBROUTINE INTERRUPT SERVICE
;*****
00006A DBA0      SERVICE:      IN A,(0A0H)      ;PROT PHONE DETECT
00006C E603                        AND 11B
00006E FE01                        CP 01B
000070 C2E800      JP NZ,START1      ;START TO CALCULATE SERVICE C

;*****
;SUBROUTINE INKEY FROM #8870 AND DETECT FAR OR NEAR
;*****
000073 2190E1      INKEY1:      LD HL,COUNTER      ;COUNTER RAM
000076 7E                        LD A,(HL)
000077 47                        LD B,A
000078 FEFF                        CP OFFH      ;DETECT FIRST KEY
00007A C29A00      JP NZ,S1
00007D CD6202      CALL CLRLCD
000080 21CC08      LD HL,TAB54      ;load TAB54" TEL....."

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000083	CD8B02		CALL DISPLAY
000086	CD5701		CALL KEYNUMBER ;DATA FROM #8870
000089	FE30		CP 30H
00008B	CA9300		JP Z,S2
00008E	0607		LD B,07H ;NEAR KEY
000090	C39500		JP S3
000093	0609	S2:	LD B,09H ;FAR KEY
000095	2190E1	S3:	LD HL,COUNTER
000098	70		LD (HL),B ;COUNTER RAM
000099	78		LD A,B
00009A	FE00	S1:	CP 0
00009C	CADA00		JP Z,S4 ;IS NOT KEY
00009F	FE01		CP 01
0000A1	200B		JR NZ,S5
0000A3	3E02		LD A,02
0000A5	D3A1		OUT (0A1H),A
0000A7	CDE002		CALL DELAY
0000AA	3E03		LD A,03
0000AC	D3A1		OUT (0A1H),A
0000AE		S5:	
0000AE	C5		PUSH BC
0000AF	CD7801		CALL TABLENUM
0000B2	C1		POP BC
0000B3	3E09		LD A,09
0000B5	90		SUB B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0000B6 83	ADD A,E
0000B7 5F	LD E,A
0000B8 3E00	LD A,0
0000BA 8A	ADC A,D
0000BB 57	LD D,A
0000BC CD5701	CALL KEYNUMBER ;DATA FROM #8870
0000BF 12	LD (DE),A ;KEEP DATA IN TABLE8
0000C0 219AE1	LD HL,DDRAM ;POSITION ON DISPLAY
0000C3 4E	LD C,(HL)
0000C4 CD6B01	CALL WRITELCD
0000C7 79	LD A,C
0000C8 FE14	CP 20
0000CA 2002	JR NZ,SETADDR
0000CC 0E00	LD C,0
0000CE 71	SETADDR: LD (HL),C
0000CF 05	DEC B
0000D0 2190E1	LD HL,COUNTER
0000D3 70	LD (HL),B
0000D4 78	LD A,B
0000D5 FE00	CP 0
0000D7 C24001	JP NZ, RESETNMI
0000DA 3E00	S4: LD A,00
0000DC D3A1	OUT (0A1H),A
0000DE CDE002	CALL DELAY

```

0000E1 3E03          LD A,03
0000E3 D3A1          OUT (0A1H),A
0000E5 C34B01        JP STOPS1

```

;SUBROUTINE TO READ START TIME

```

0000E8 DBA0          START1:  IN A,(0A0H)      ;PROT PHONE DETECT
0000EA E603          AND 11B
0000EC FE03          CP 011B
0000EE C21D01        JP NZ,STP1
0000F1 3E01          LD A,01H
0000F3 2195E1        LD HL,SEQUE
0000F6 77            LD (HL),A
0000F7 3A02E2        LD A,(NUMTEL)
0000FA 3C            INC A
0000FB 3202E2        LD (NUMTEL),A
0000FE 1600          LD D,0
000100 5F            LD E,A
000101 3E03          LD A,3
000103 CD8F01        CALL MUL
000106 114409        LD DE,TABLE9 ;ld a,(monney)
000109 7D            LD A,L
00010A 83            ADD A,E
00010B 5F            LD E,A

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

00010C 3E00          LD A,0H

00010E 8A            ADC A,D

00010F 57            LD D,A

000110 EB            EX DE,HL

000111 1104E2        LD DE,MONEY

000114 010300        LD BC,3

000117 CD8D04        CALL MOVES

00011A C34001        JP RESETNMI

;*****

;SUBROUTINE READ STOPTIME AND CALCULATE MONEY

;*****

00011D 2195E1        STP1: LD HL,SEQUE

000120 7E            LD A,(HL)

000121 FE01          CP 01H

000123 CA2E01        JP Z,EN2

000126 CD7801        EN1: call tablenum ; input in HL =start address B =

000129 0614          ld b,20 ;

00012B CD7D04        call clear ;

00012E 2195E1        EN2: LD HL,SEQUE

000131 3600          LD (HL),0

000133 2190E1        LD HL,COUNTER ;PUSH 7 TO COUNTER RAM

000136 36FF          LD (HL),OFFH ;INIT FOR FRIST NMI

000138 3E07          LD A,7

00013A 329AE1        LD (DDRAM),A

00013D CD6202        CALL CLRLCD

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

000140 3E00      RESETNMI:    LD A,00H
000142 D3A1                      OUT (0A1H),A
000144 CDE002                      CALL DELAY
000147 3E01                      LD A,01H
000149 D3A1                      OUT (0A1H),A
00014B E1        STOPS1:      POP HL
00014C D1                      POP DE
00014D C1                      POP BC
00014E F1                      POP AF
00014F CD6F04                  CALL DELAY1
000152 CD6F04                  CALL DELAY1
000155 ED45                      RETN

```

```

; *****KEYNUMBER*****
; OUTPUT = KEYTEL ;ASCII CODE
;           = Reg A ;ASCII CODE
; *****

```

```

000157 DB22      KEYNUMBER:   IN A,(22H)
000159 E60F                      AND OFH
00015B D5                      PUSH DE
00015C 118E08                  LD DE,TAB3
00015F 83                      ADD A,E
000160 5F                      LD E,A
000161 3E00                      LD A,0H
000163 8A                      ADC A,D

```

```

000164 57          LD D,A
000165 1A          LD A,(DE)
000166 3298E1      LD (KEYTEL),A
000169 D1          POP DE
00016A C9          RET

```

*****WRITE LCD*****

```

; INPUT = A ,DATA
;          = C ,ADDRESS IN DD RAM

```

```

00016B 57      WRITELCD: LD D,A
00016C 79      LD A,C
00016D CDC802  CALL GOTO
000170 CDB702  CALL EPLUSE
000173 CBD502  CALL WRBYTE
000176 0C      INC C
000177 C9      RET

```

*****OPENTABLE FOR PUSH DATA*****

```

;          INPUT = DE ;ADRESS BEGIN
;          = (NUMTEL) ; NUMBERS OF HANDSET
;          OUTPUT= DE ,HL

```

```

000178      TABLENUM:
000178 2102E2  LD HL,NUMTEL
00017B 7E      LD A,(HL)
00017C 2100E8  LD HL,TABLE8

```

```

00017F FE00          CP 0

000181 2809          JR Z,N1

000183 111400      N:    LD DE,20

000186 19           ADD HL,DE

000187 3D           DEC A

000188 FE00          CP 0

00018A 20F7          JR NZ,N

00018C 54           N1:   LD D,H

00018D 5D           LD E,L

00018E C9           RET

;*****SUBROUTINE MULTIPLIER*****
;USE REG DE,HL,AF,BC
;INPUT DE=MULTIPLIER,A=MULTIPLICAND
;OUTPUT HL=RESOULT
;*****

00018F C5           MUL:  PUSH BC

000190 010008        LD BC,0800H ;COUNTER BIT

000193 61           LD H,C

000194 29           M1:   ADD HL,HL ;

000195 07           RLCA   ;ROTATE MULTIPLICAND IN CARRY

000196 3002          JR NC,M2 ;CARRY RESET JUMP TO M2

000198 19           ADD HL,DE

000199 89           ADC A,C ;A=A+C+CY

00019A 10F8          M2:   DJNZ M1 \

```

00019C C1

POP BC

00019D C9

RET

*****MAIN PROGRAM*****

;* 1.Display demo project *

;* 2.set time set date *

;* 3.work *

00019E CD3702

START:

CALL INITIAL ;Set all Micro system

0001A1 214107

AGAIN:

LD HL,TAB11 ;hl is index add for display den

0001A4 CD8B02

CALL DISPLAY

0001A7 CD6F04

CALL DELAY1

0001AA CD6F04

CALL DELAY1

0001AD CD6F04

CALL DELAY1

0001B0 216907

LD HL,TAB12 ;hl is index add for display den

0001B3 CD8B02

CALL DISPLAY

0001B6 CD6F04

CALL DELAY1

0001B9 CD6F04

CALL DELAY1

0001BC CD6F04

CALL DELAY1

0001BF 219107

LD HL,TAB13 ;hl is index add for display den

0001C2 CD8B02

CALL DISPLAY

0001C5 CD6F04

CALL DELAY1

0001C8 CD6F04

CALL DELAY1

0001CB CD6F04

CALL DELAY1

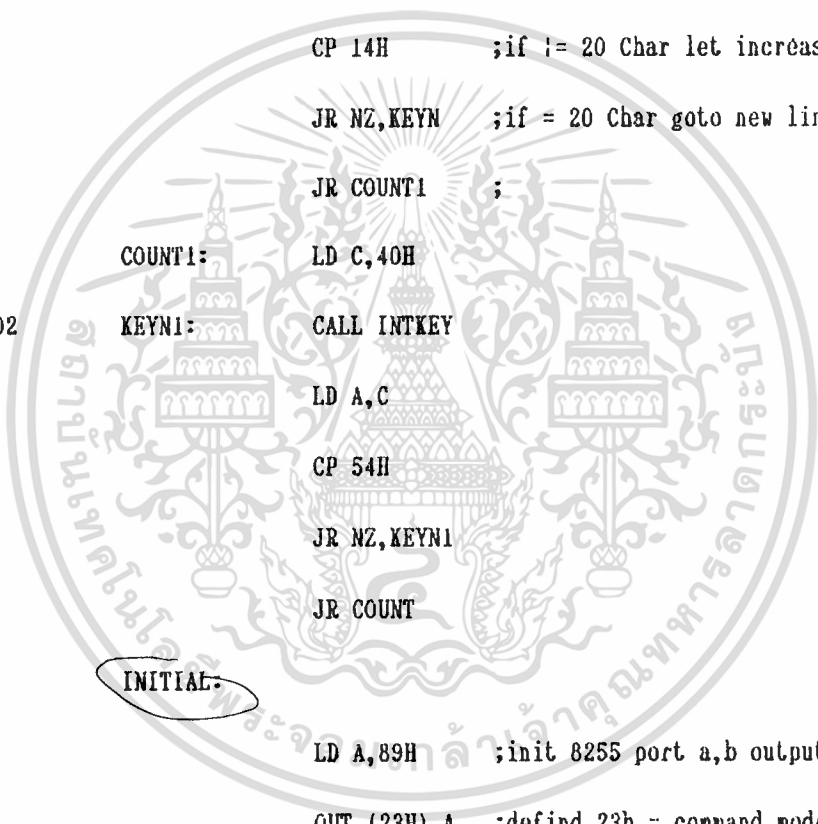
0001CE 21BD07

LD HL,TAB14 ;hl is index add for display den

0001D1 CD8B02

CALL DISPLAY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

000217 2104E2          LD HL,MONNEY

00021A 3630      NEY:    LD (HL),30H    ;clear position monney

00021C 23                INC HL

00021D 10FB                DJNZ NEY

00021F 0E00      COUNT:  LD C,00        ;Reg C index begin line in DD Ra

000221 CDE902      KEYN:   CALL INTKEY   ;Enable interrupt keyboard

000224 79                LD A,C        ;check location DD Ram = 1 first

000225 FE14                CP 14H        ;if != 20 Char let increase DD R

000227 20F8                JR NZ,KEYN    ;if = 20 Char goto new line 40h

000229 1800                JR COUNT1    ;

00022B 0E40      COUNT1:  LD C,40H

00022D CDE902      KEYN1:  CALL INTKEY

000230 79                LD A,C

000231 FE54                CP 54H

000233 20F8                JR NZ,KEYN1

000235 18E8                JR COUNT

000237      INITIAL:

000237 3E89                LD A,89H    ;init 8255 port a,b output port

000239 D323                OUT (23H),A    ;defind 23h = command mode 8255

00023B 3E90                LD A,90H    ;init 8255 port a,input portb,c

00023D D3A3                OUT (0A3H),A    ;defind a3h = command mode 8255

00023F 3E00                LD A,00        ;init for ready detect system po

000241 D3A1                OUT (0A1H),A

000243 CDE002            CALL DELAY

000246 3E01                LD A,01        ;clear NMI PORT B=01

```

```

000217 2104E2          LD HL,MONNEY

00021A 3630      NEY:      LD (HL),30H      ;clear position monney

00021C 23          INC HL

00021D 10FB          DJNZ NEY

00021F 0E00      COUNT:    LD C,00          ;Reg C index begin line in DD Ra

000221 CDE902      KEYN:     CALL INTKEY      ;Enable interrupt keyboard

000224 79          LD A,C          ;check location DD Ram = 1 first

000225 FE14          CP 14H          ;if != 20 Char let increase DD R

000227 20F8          JR NZ,KEYN      ;if = 20 Char goto new line 40h

000229 1800          JR COUNT1      ;

00022B 0E40      COUNT1:    LD C,40H

00022D CDE902      KEYN1:     CALL INTKEY

000230 79          LD A,C

000231 FE54          CP 54H

000233 20F8          JR NZ,KEYN1

000235 18E8          JR COUNT

000237      INITIAL:

000237 3E89          LD A,89H      ;init 8255 port a,b output port

000239 D323          OUT (23H),A      ;defind 23h = command mode 8255

00023B 3E90          LD A,90H      ;init 8255 port a,input portb,c

00023D D3A3          OUT (0A3H),A      ;defind a3h = command mode 8255

00023F 3E00          LD A,00          ;init for ready detect system po

000241 D3A1          OUT (0A1H),A

000243 CDE002      CALL DELAY

000246 3E01          LD A,01          ;clear NMI PORT B=01

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

000282 CDB702          CALL EPLUSE

000285 CDE002          CALL DELAY

000288 ED56            IM 1

00028A C9              RET

                        ;*****DISPLAY LCD MODULE*****

                        ; INPUT = HL START ADDR FOR DISPLAY

                        ;*****

00028B 3E01      DISPLAY: LD A,1

00028D CD9602    CALL WRLINE

000290 3E02      LD A,2

000292 CD9602    CALL WRLINE

000295 C9        RET

000296 FE01      WRLINE: CP 1

000298 2B05      JR Z,WRL1

00029A FE02      CP 2

00029C 2B08      JR Z,WRL2

00029E C9        RET

00029F 3E00      WRL1: LD A,00H

0002A1 CDC802    CALL GOTO

0002A4 1B05      JR WRLM

0002A6 3E40      WRL2: LD A,40H

0002A8 CDC802    CALL GOTO

0002AB 0614      WRLM: LD B,20

0002AD 56        WRL: LD D,(HL)

0002AE C5        PUSH BC

```

0002AF CDD502	CALL WRBYTE
0002B2 C1	POP BC
0002B3 23	INC HL
0002B4 10F7	DJNZ WRL
0002B6 C9	RET
0002B7 DB21	EPLUSE: IN A, (21H)
0002B9 CBD7	SET 2,A
0002BB D321	OUT (21H),A
0002BD C5	PUSH BC
0002BE 0600	LD B,00H
0002C0 10FE	ED1: DJNZ ED1
0002C2 C1	POP BC
0002C3 CB97	RES 2,A
0002C5 D321	OUT (21H),A
0002C7 C9	RET
0002C8 C5	GOTO: PUSH BC
0002C9 CBFF	SET 7,A
0002CB D320	OUT (20H),A
0002CD AF	XOR A
0002CE D321	OUT (21H),A
0002D0 CDB702	CALL EPLUSE
0002D3 C1	POP BC
0002D4 C9	RET
0002D5 3E01	WRBYTE: LD A,00000001B
0002D7 D321	OUT (21H),A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0002D9 7A		LD A,D
0002DA D320		OUT (20H),A
0002DC CDB702		CALL EPLUSE
0002DF C9		RET
0002E0 C5	DELAY:	PUSH BC
0002E1 0600		LD B,0
0002E3 00	DE1:	NOP
0002E4 00		NOP
0002E5 10FC		DJNZ DE1
0002E7 C1		POP BC
0002E8 C9		RET
		*****ENABLE KEYBOARD*****
		; OUTPUT = (DATAKEY) ,DATA ASCII CODE
		; = (KEYIN) .KEYCODE FROM IC 8279

0002E9 FB	INTKEY:	EI
0002EA 00		NOP
0002EB 00		NOP
0002EC 00		NOP
0002ED F3		DI
0002EE C9		RET
0002EF DB80	KEYBOARD:	IN A,(80H)
0002F1 47		LD B,A
0002F2 3200E2		LD (KEYIN),A
0002F5 FE15		CP 15H

0002F7 CA5503	JP Z,BACKTAB
0002FA FE07	CP 07H
0002FC CA9004	JP Z,TIMEDATE
0002FF FE0F	CP 0FH
000301 CA5F03	JP Z,NAME
000304 FE17	CP 17H
000306 CA4B03	JP Z,PRINT1 .
000309 FE1F	CP 1FH
00030B CA5303	JP Z,PRINT2
00030E FE27	CP 27H
000310 CAAD03	JP Z,LINEFEED
000313 FE2F	CP 2FH
000315 CAAF03	JP Z,ALIST
000318 FE3F	CP 3FH
00031A CA9E01	JP Z,START
00031D 17	RLA
00031E DA2403	JP C,KEYFUNCTION
000321 C35E04	JP KEYDATA
000324 CB3F	KEYFUNCTION: SRL A
000326 FE32	CP 32H
000328 CA9004	JP Z,TIMEDATE
00032B FE12	CP 12H
00032D CA5F03	JP Z,NAME
000330 FE22	CP 22H
000332 CA4B03	JP Z,PRINT1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000335 FE0C	CP 0CH
000337 CA5303	JP Z,PRINT2
00033A FE1C	CP 1CH
00033C CAAD03	JP Z,LINEFEED
00033F FE2C	CP 2CH
000341 CAAF03	JP Z,ALIST
000344 FE04	CP 04H
000346 CA9E01	JP Z,START
000349 C9	RET
00034A C9	STARTTDATE: RET
00034B CD6202	PRINT1: CALL CLRLCD
00034E C31F02	JP COUNT
000351 C9	RET
000352 C9	STARTPRINT1: RET
000353 C9	PRINT2: RET
000354 C9	STARTPRINT2: RET
000355 CD6202	BACKTAB: CALL CLRLCD
000358 C31F02	JP COUNT ; CLEAR KEY CMD
00035B CD5E04	CALL KEYDATA ;
00035E C9	RET ;
00035F 2140E0	NAME: LD HL,0E040H ;CLEAR BUFFER NAME
000362 0614	LD B,20
000364 CD7D04	CALL CLEAR
000367 219F08	LD HL,TAB51 ;DEMO ENTER NAME
00036A CD8B02	CALL DISPLAY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00036D 0E45		LD C,45H	;POSITION ENTER NAME
00036F CDE902	BENAME:	CALL INTKEY	
000372 D5		PUSH DE	
000373 F5		PUSH AF	
000374 1100E0		LD DE,TEMPO	;BUFFER TEMPOLARY
000377 79		LD A,C	
000378 83		ADD A,E	
000379 5F		LD E,A	
00037A 3A0FE2		LD A,(DATAKEY)	
00037D 12		LD (DE),A	
00037E F1		POP AF	
00037F D1		POP DE	
000380 3A00E2		LD A,(KEYIN)	
000383 FE10		CP 10H	
000385 CA9203		JP Z,GONAME	
000388 79		LD A,C	
000389 FE55		CP 55H	
00038B CA5F03		JP Z,NAME	
00038E C36F03		JP BENAME	
000391 C9		RET	
000392 0E12	GONAME:	LD C,12H	;KEY CODE FROM ENTER NAME FU
000394 CD1804		CALL RETURN	
000397 21B308	STARTNAME:	LD HL,TAB52	;MOVE "NAME:" TO DISPLAY
00039A 1140E0		LD DE,0E040H	
00039D 010600		LD BC,6	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0003A0 CD8D04	CALL MOVES
0003A3 2140E0	LD HL,0E040H ;DISPLAY "NAME:-----"
0003A6 CD8B02	CALL DISPLAY
0003A9 C31F02	JP COUNT
0003AC C9	RET
0003AD C9	LINEFEED: RET
0003AE C9	STARTLINEFEED: RET
0003AF 2140E0	ALIST: LD HL,0E040H ;MOVE NAME FROM NAMEBUFFER TO RAM
0003B2 1100E1	LD DE,RAMDISPLAY ;0E100H
0003B5 011400	LD BC,20
0003B8 CD8D04	CALL MOVES
0003BB 21B808	LD HL,TAB53
0003BE 1114E1	LD DE,0E114H
0003C1 011400	LD BC,20
0003C4 CD8D04	CALL MOVES
0003C7 2104E2	LD HL,MONNEY ;MOVE NUMBER TEL..BUFF TO RAMDISP
0003CA 1118E1	LD DE,0E118H
0003CD 010300	LD BC,3
0003D0 CD8D04	CALL MOVES
0003D3 2100E1	LD HL,RAMDISPLAY ;0E100H
0003D6 CD8B02	CALL DISPLAY
0003D9 CD6F04	CALL DELAY1
0003DC CD6F04	CALL DELAY1
0003DF CD6F04	CALL DELAY1
0003E2 CD6F04	CALL DELAY1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0003E5 2102E2		LD HL,NUMTEL
0003E8 46		LD B,(HL)
0003E9 CB38		SRL B
0003EB 04		INC B
0003EC 2100E8		LD HL,TABLE8
0003EF C5	A1:	PUSH BC
0003F0 E5		PUSH HL
0003F1 CD8B02		CALL DISPLAY
0003F4 CD6F04		CALL DELAY1
0003F7 CD6F04		CALL DELAY1
0003FA CD6F04		CALL DELAY1
0003FD CD6F04		CALL DELAY1
000400 011400		LD BC,20
000403 E1		POP HL
000404 09		ADD HL,BC
000405 C1		POP BC
000406 78		LD A,B
000407 05		DEC B
000408 FE00		CP 0
00040A C2EF03		JP NZ,A1
00040D CD6202		CALL CLRLCD
000410 C31F02		JP COUNT
000413 C9		RET
000414 C31F02	STARTALIST:	JP COUNT
000417 C9		RET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000418 79	RETURN:	LD A,C ;LOAD CODE FUNCTION IN Reg C BEFORE R
000419 FE32		CP 32H
00041B CA0905		JP Z,T3
00041E FE33		CP 33H
000420 CA8205		JP Z,Q2
000423 FE07		CP 07H
000425 CA0905		JP Z,T3
000428 FE12		CP 12H
00042A CA9703		JP Z,STARTNAME
00042D FE0F		CP 0FH
00042F CA9703		JP Z,STARTNAME
000432 FE22		CP 22H
000434 CA5203		JP Z,STARTPRINT1
000437 FE17		CP 17H
000439 CA5203		JP Z,STARTPRINT1
00043C FE0C		CP 0CH
00043E CA5403		JP Z,STARTPRINT2
000441 FE1F		CP 1FH
000443 CA5403		JP Z,STARTPRINT2
000446 FE1C		CP 1CH
000448 CAAE03		JP Z,STARTLINEFEED
00044B FE27		CP 27H
00044D CAAE03		JP Z,STARTLINEFEED
000450 FE2C		CP 2CH
000452 CA1404		JP Z,STARTALIST

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000455	FE2E	CP 2EH
000457	CA1404	JP Z,STARTALIST
00045A	C39E01	JP START
00045D	C9	RET
00045E	D5	KEYDATA: PUSH DE
00045F	78	LD A,B
000460	110D08	LD DE,TAB2
000463	83	ADD A,E
000464	5F	LD E,A
000465	3E00	LD A,0H
000467	8A	ADC A,D
000462	57	LD D,A
000469	1A	LD A,(DE)
00046A	320FE2	LD (DATAKEY),A
00046D	D1	POP DE
00046E	C9	RET
		*****DELAY LONGTIME *****
00046F	F5	DELAY1: PUSH AF
000470	E5	PUSH HL
000471	21FFFF	LD HL,0FFFFH
000474	2B	LOOP1: DEC HL
000475	7C	LD A,H
000476	B5	OR L
000477	C27404	JP NZ,LOOP1
00047A	E1	POP HL

```

00047B F1          POP AF

00047C C9          RET

;*****CLEAR RAM DISPLAY*****

;          INPUT = HL START ADDR

;          = B LENGTH

;*****

00047D 00          CLEAR:      NOP

00047E 3620        NEWCLR:     LD (HL),20H

000480 23          INC HL

000481 10FB        DJNZ NEWCLR

000483 C9          RET

;*****CLEAR RAM*****

;          IN = HL START ADDRESS

;          = BC LENGTH

;*****

000484 3620        CLR:        LD (HL),20H

000486 E5          PUSH HL

000487 D1          POP DE

000488 13          INC DE

000489 0B          DEC BC

00048A EDB0        LDIR

00048C C9          RET

```

*****MOVES DATA*****

; INPUT = DE DESTINATION

; = HL START

; = BC LENGTH

00048D MOVES:

00048D EDB0 LDIR

00048F C9 RET

*****SUB PROGRAM REAL TIME CLOCK****

*****USE REGISTER=A,B,HL,IY,DE,*****

*****READ DATA(BCD) FROM RTC AND COVERT TO ASCII****

*****MOVE DATA DATE(ASCII)TABLE7 TO DISPLAYTABLE****

000490 F5 TIMEDATE: PUSH AF

000491 C5 PUSH BC

000492 D5 PUSH DE

000493 E5 PUSH HL

000494 CDOA07 CALL READDATE

*****MOVE DATA TO RTC DISPLSY BUFFER*****

000497 2100E7 Z2: LD HL,TABLE7 ;DISPLAY TABLE

00049A 1100E1 LD DE,RAMDISPLAY ;LCD DISPLAY BUFFER

00049D 061C LD B,28

00049F EDB0 LDIR ;MOVE DATA FROM DISTABLE TO

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0004A1 2100E1		LD HL,RAMDISPLAY ;HL INDEX POSITION FOR DISPL
0004A4 CD8B02		CALL DISPLAY
0004A7 3EFF		LD A,OFFH
0004A9 3200E2		LD (KEYIN),A
0004AC 111D00		LD DE,001DH
0004AF 2100E7		LD HL,TABLE7
0004B2 19		ADD HL,DE
0004B3 0E49	NEWLOOP1:	LD C,49H
0004B5 CDE902	NEWSET1:	CALL INTKEY
0004B8 3A00E2		LD A,(KEYIN)
0004BB FEFF		CP OFFH
0004BD CAB504		JP Z,NEWSET1
0004C0 FE10		CP 10H
0004C2 CAD804		JP Z,GORETURN1
0004C5 3A0FE2		LD A,(DATAKEY)
0004C8 77		LD (HL),A
0004C9 23		INC HL
0004CA 3EFF		LD A,OFFH
0004CC 3200E2		LD (KEYIN),A
0004CF 79		LD A,C
0004D0 FE52		CP 052H
0004D2 C2B504		JP NZ,NEWSET1
0004D5 C3DD04		JP Q1
0004D8 0E32	GORETURN1:	LD C,32H
0004DA C31804		JP RETURN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

;* CONVERT DATA ASCII(TABLE62)TO BCD AND MOVE INTO RTCBUFFER(

0004DD DD2100E7 Q1: LD IX, TABLE7 ;DISPLAY TABLE POINTER
0004E1 FD2100E3 LD IY, TABLE6 ;RTC BUFFER POINTER
0004E5 DD561D LD D, (IX+1DH) ;GATE DATE FROM DISPLAY TABLE
0004E8 DD5E1E LD E, (IX+1EH) ;
0004EB CD0006 CALL CON2 ;CONVERT DATE(ASCII) TO BCD
0004EE FD7707 LD (IY+7H), A ;KEEP DATE(BCD) IN RTC BUFFER
0004F1 DD5620 LD D, (IX+20H) ;GATE MONTH FROM DISPLAY TABLE
0004F4 DD5E21 LD E, (IX+21H) ;
0004F7 CD0006 CALL CON2 ;CONVERT MONTH(ASCII) TO BCD
0004FA FD7708 LD (IY+8H), A ;KEEP MONTH(BCD) IN RTC BUFFER
0004FD DD5623 LD D, (IX+23H) ;GATE YEAR FROM DISPLAY TABLE
000500 DD5E24 LD E, (IX+24H) ;
000503 CD0006 CALL CON2 ;CONVERT YEAR(ASCII) TO BCD
000506 FD7709 LD (IY+9H), A ;KEEP YEAR(BCD) IN RTC BUFFER

;***** DISPLAY OLD TIME *****

000509 CDD706 T3: CALL READTIME
00050C 2100E7 Y2: LD HL, TABLE7 ;DISPLY TABLE POINTER
00050F 1100E1 LD DE, RAMDISPLAY ;LCD DISPLAY BUFFER POINTER
000512 012800 LD BC, 0028H
000515 EDB0 LD IR
000517 2100E1 LD HL, RAMDISPLAY

00051A	CD8B02	CALL DISPLAY
00051D	3EFF	LD A,OFFH
00051F	3200E2	LD (KEYIN),A
000522	111D00	LD DE,001DH
000525	2100E7	LD HL,TABLE7
000528	19	ADD HL,DE
000529	0E49	NEWLOOP2: LD C,49H
00052B	CDE902	NEWSET2: CALL INTKEY
00052E	3A00E2	LD A,(KEYIN)
000531	FEFF	CP OFFH ;CHECK PRESSKEY
000533	CA2B05	JP Z,NEWSET2 ;JUMP IF NO KEYPRESS
000536	FE10	CP 10H ;CHECK ENTER KEY
000538	CA5105	JP Z,GORETURN2;JUMP TO SET DATE/TIME
00053B	3A0FE2	LD A,(DATAKEY)
00053E	77	LD (HL),A
00053F	23	INC HL
000540	3EFF	LD A,OFFH
000542	3200E2	LD (KEYIN),A
000545	79	LD A,C
000546	FE52	CP 52H
000548	C22B05	JP NZ,NEWSET2
00054B	C36C06	JP TESTTIME
00054E	C35605	JP Q21
000551	0E33	GORETURN2: LD C,33H
000553	C31804	JP RETURN

;** CONVERT DATETIME(TABLE63)AND MOVE INTO RTCBUFFER(TABLE6)

000556 DD2100E7 Q21: LD IX, TABLE7 ;BUFFER TABLE POINTER

00055A FD2100E3 LD IY, TABLE6 ;DISPLAY TABLE POINTER

00055E DD561D LD D, (IX+1DH) ;HIGH BYTE HOUR(ASCII) FROM DISP

000561 DD5E1E LD E, (IX+1EH) ;LOW BYTE HOUR(ASCII) FROM DISPL

000564 CD0006 CALL CON2 ;CONVERT ASCII TO BCD

000567 FD7704 LD (IY+4H), A ;KEEP HOUR(BCD) IN RTC BUFFER

00056A DD5620 LD D, (IX+20H) ;HIGH BYTE OF MINUTE(ASCII) FROM

00056D DD5E21 LD E, (IX+21H) ;LOW BYTE OF MINUTE(ASCII) FROM

000570 CD0006 CALL CON2 ;CONVERT ASCII TO BCD

000573 FD7702 LD (IY+2H), A ;KEEP MINUTE(BCD) IN RTC BUFFER

000576 DD5623 LD D, (IX+23H) ;HIGH BYTE OF SECOND(ASCII) FROM

000579 DD5E24 LD E, (IX+24H) ;LOW BYTE OF SECOND(ASCII) FROM

00057C CD0006 CALL CON2 ;CONVERT ASCII TO BCD

00057F FD7700 LD (IY+0H), A ;KEEP SECOND(BCD) IN RTC BUFFER

000582 CDB406 Q2: CALL SET ;SET DATE/TIME TO RTC

000585 E1 POP HL

000586 D1 POP DE

000587 C1 POP BC

000588 F1 POP AF

000589 CD6202 CALL CLRLCD

00058C C31F02 JP COUNT

;*****SUBROUTINE READ TIME/DATE*****

;*****INPUT HL=TABLE6,OUT=TABLE61*****

;*****RTC->TABLE6->TABLE60->TABLE61*****

;*****USE REGISTER B,DE,HL*****

;*****READ DATA(BCD) FROM RTC KEEP INTO RTC BUFFER(TABL

00058F F5 READTD: PUSH AF
000590 C5 PUSH BC
000591 D5 PUSH DE
000592 E5 PUSH HL
000593 060E LD B,0EH ;
000595 1600 LD D,00H
000597 2100E3 LD HL,TABLE6 ;RTC BUFFER POINTER
00059A CDD605 READ1: CALL RDbyte ;CALL SUBREAD DATA FROM RTC
00059D 73 LD (HL),E ;KEEP DATA TO RTCBUFFER
00059E 14 INC D
00059F 23 INC HL
0005A0 10F8 DJNZ READ1

;*****MOVE DATA(BCD) FROM RTCBUFFER(TABLE 6) INTO BCD TABL

0005A2 DD2100E3 LD IX,TABLE6 ;RTC BUFFER POINTER
0005A6 2100E4 LD HL,TABLE60 ;
0005A9 DD7E04 LD A,(IX+04H) ;HOUR FROM RTC BUFFER

0005AC 77	LD (HL),A	;KEEP TO TABLE
0005AD 23	INC HL	
0005AE DD7E02	LD A,(IX+02H)	;MINUTE FROM RTC BUFFER
0005B1 77	LD (HL),A	;KEEP TO TABLE
0005B2 23	INC HL	
0005B3 DD7E00	LD A,(IX+0H)	;SECOND FROM RTC BUFFER
0005B6 77	LD (HL),A	;KEEP TO TABLE
0005B7 23	INC HL	
0005B8 DD7E07	LD A,(IX+07H)	;DATE FROM RTC BUFFER
0005BB 77	LD (HL),A	;KEEP TO TABLE
0005BC 23	INC HL	
0005BD DD7E08	LD A,(IX+08H)	;MONTH FROM RTC BUFFER
0005C0 77	LD (HL),A	;KEEP TO TABLE
0005C1 23	INC HL	
0005C2 DD7E09	LD A,(IX+09H)	;YEAR FROM RTC BUFFER
0005C5 77	LD (HL),A	
;***CONVERT DATA(BCD)TABLE60 TO ASCII KEEP INTO ASCII TABLE		
0005C6 2100E4	LD HL,TABLE60	;RTC BUFFER POINTER
0005C9 1100E5	LD DE,TABLE61	
0005CC 0607	LD B,07H	;BUFFER TABLE POINTER
0005CE CDDD05	CALL CON1	;CONVER DATA BCD TO ASCII KEEP
		;IN BUFFER TABLE 61
0005D1 E1	POP HL	

0005D2 D1	POP DE
0005D3 C1	POP BC
0005D4 F1	POP AF
0005D5 C9	RET

*****SUB ROUTINE READ*****

*****IN D=ADD,OUT E=DATA*****

*****USE REGISTER=A,DE,*****

0005D6 7A	RDbyte:	LD A,D
0005D7 D340		OUT (40H),A
0005D9 DB41		IN A,(41H)
0005DB 5F		LD E,A
0005DC C9		RET

*****SUB ROUTINE CONVERT BCD TO ASCII*****

*****IN HL=RTC BUFFER,OUT DE=BUFFER*****

*****USE REGISTER=A,C,DE,HL,*****

0005DD 7E	CON1:	LD A,(HL)	;BCD FROM RTC BUFFER
0005DE 4F		LD C,A	
0005DF E6F0		AND 0F0H	;BLANK 4 HIGHER BIT
0005E1 1F		RRA	
0005E2 1F		RRA	
0005E3 1F		RRA	
0005E4 1F		RRA	
0005E5 CDF605		CALL NASCII	



```

0005E8 12          LD (DE),A      ;ASCII OF 4 HIGHER BIT
0005E9 13          INC DE
0005EA 79          LD A,C
0005EB E60F        AND 0FH
0005ED CDF605      CALL NASCII
0005F0 12          LD (DE),A
0005F1 13          INC DE
0005F2 23          INC HL
0005F3 10E8        DJNZ CON1
0005F5 C9          RET
0005F6 FE0A        NASCII: CP 10
0005F8 DAFD05      JP C,NASI
0005FB C607        ADD A,07
0005FD C630        NASI:  ADD A,30H
0005FF C9          RET

;***** SUBROUTINE CONVERT ASCII TO BCD*****
;*****IN DE=DATA,OUT=A*****
;*****USE REGISTER=A,DE,C*****

000600 7B          CON2:  LD A,E      ;HIGHER BYTE
000601 CD1306      CALL A2HEX      ;CONVERT TO HEX VALUE
000604 4F          LD C,A
000605 7A          LD A,D      ;LOWER BYTE
000606 CD1306      CALL A2HEX      ;CONVERT TO HEX VALUE
000609 CB27        SLA A
00060B CB27        SLA A

```

```

00060D CB27          SLA A

00060F CB27          SLA A

000611 B1            OR C          ;OR A WITH LOW HEX VALUE

000612 C9            RET

000613 D630          A2HEX:        SUB 30H

000615 FE0A          CP 10

000617 DA1C06        JP C,A2HEX1

00061A D607          SUB 07

00061C C9            A2HEX1:      RET

                                ;*****SUB ROUTINE TEST DATE FROM*****
                                ;*****USE REGISTER=A,B,IX,*****
00061D DD2100E7      TESTDATE:    LD IX,TABLE7;DISPLAY DATE TABLE
000621 DD7E1D          LD A,(IX+10H) ;HIGHER BYTE OF DATE
000624 FE33          CP 33H
000626 CA4006        JP Z,LBD      ;EQU ASCII "3" JUMP TO LBD
000629 0633          LD B,33H
00062B 90            SUB B
00062C D25A06        JP NC,P0      ;IF LAST THEN ASCII"3" JUMP TO P0


00062F DD7E20          HBM:        LD A,(IX+20H) ;HIGHER BYTE OF MONTH
000632 FE31          CP 31H
000634 CA4D06        JP Z,LBM      ;EQU ASCII"1" JUMP TO LBM
000637 0631          LD B,31H
000639 90            SUB B

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00063A D25A06		JP NC,PO	;HIGHER BYTE OF MONTH IF
			;LAST THEN ASCII"1" JUMP TO PO(ER
00063D C34D06		JP LBM	
000640 DD7E1E	LBD:	LD A,(IX+1EH)	;LOWER BUTE OF DATE
000643 47		LD B,A	
000644 3E31		LD A,31H	;A=ASCII"1"
000646 90		SUB B	
000647 DA5A06		JP C,PO	;IF LAST THEN ASCII"2" JUMP TO PO
00064A C32F06		JP HBM	
00064D DD7E21	LBM:	LD A,(IX+21H)	;LOWER BYTE OF MONTH
000650 47		LD B,A	
000651 3E32		LD A,32H	;A=ASCII"2"
000653 90		SUB B	
000654 DA5A06		JP C,PO	;IF LAST THEN ASCII"2" JUMP TO PO
000657 C3DD04	X31:	JP Q1	;RETURN TO MAIN
00065A 2100E7	PO:	LD HL,TABLE7	;DISPLAY TABLE POINTER
00065D 111D00		LD DE,1DH	
000660 19		ADD HL,DE	
000661 3E20		LD A,20H	
000663 0608		LD B,08H	
000665 77	Z1:	LD (HL),A	
000666 23		INC HL	
000667 10FC		DJNZ Z1	
000669 C39704		JP Z2	;JUMP TO DISPLAY D/M/Y AGAIN

;*****SUB ROUTINE TEST TIME*****

;*****USE REGISTER=A,B,IX,*****

```
00066C DD2100E7    TESTTIME:    LD IX, TABLE7 ;DISPLAYTIME TABLE
000670 DD7E1D      LD A, (IX+1DH) ;HIGHER BYTE OF HOUR
000673 FE32        CP 32H
000675 CA9506      JP Z, LBH      ;EQU ASCII"2" JUMP TO J1
000678 0632        LD B, 32H      ;B=ASCII"2"
00067A 90          SUB B
00067B D2A206      JP NC, P00      ;IF LAST THEN ASCII"2" JUMP TO P0
00067E DD7E20      HBN:          LD A, (IX+20H) ;HIGH BYTE OF MINUTE
000681 47          LD B, A
000682 3E35        LD A, 35H      ;A=ASCII"5"
000684 90          SUB B
000685 DAA206      JP C, P00      ;IF LAST THEN ASCII"5" JUMP TO P0
000688 DD7E23      LD A, (IX+23H) ;HIGHER BYTE OF SECOND
00068B 47          LD B, A
00068C 3E35        LD A, 35H      ;ASCII "5"
00068E 90          SUB B
00068F DAA206      JP C, P00      ;IF LAST THEN ASCII"5" JUMP TO P0
000692 C35605      X32:          JP Q21
000695 DD7E1E      LBH:          LD A, (IX+1EH) ;LOWER BYTE OF HOUR
000698 47          LD B, A
000699 3E33        LD A, 33H      ;A=ASCII"3"
00069B 90          SUB B
00069C DAA206      JP C, P00      ;IF LAST THEN ASCII"3" JUMP TO P0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00069F C37E06		JP HBN
0006A2 2100E7	P00:	LD HL, TABLE7 ;DISPLAY TABLE POINTER
0006A5 111D00		LD DE, 1DH
0006A8 19		ADD HL, DE
0006A9 3E20		LD A, 20H
0006AB 0609		LD B, 09H
0006AD 77	Y1:	LD (HL), A
0006AE 23		INC HL
0006AF 10FC		DJNZ Y1
0006B1 C30C05		JP Y2 ;JUMP TO DISPLLY H:M:S AGAIN
		*****SUB ROUTINE SET TIME/DATE*****
		***** IN D=ADD, HL= RCTBUFF POINTER, E=DATA***
		*****USE REGISTER=A, DE, HL, *****
0006B4 160B	SET:	LD D, 0BH ;ADD OF REGISTER B
0006B6 CDD605		CALL RDbyte ;READ REGISTER B FROM RTC
0006B9 CBF8		SET 7, E
0006BB 160B		LD D, 0BH
0006BD CDD006		CALL WRbyte1
0006C0 060E	WRITE:	LD B, 0EH ;VALUE OF DATA
0006C2 1600		LD D, 0H
0006C4 2100E3		LD HL, TABLE6 ;RTC BUFFER POINTER
0006C7 5E	WRITE1:	LD E, (HL)
0006C8 CDD006		CALL WRbyte1
0006CB 23		INC HL
0006CC 14		INC D

```

0006CD 10F8          DJNZ WRITE1

0006CF C9            RET

;*****SUB ROUTINE WRITE BYTE*****
;*****IN D=ADD,E=DATA*****
;***** USE REGISTER=A,DE,*****

0006D0 7A          WRbyte1:  LD A,D          ;REGISTER ADD

0006D1 D340          OUT (40H),A      ;WRITE ADD TO RTC

0006D3 7B          LD A,E          ;DATA WANT TO WRITE

0006D4 D341          OUT (41H),A      ;WRITE DATA TO RTC

0006D6 C9            RET

;*****SUBROUTINE READTIME*****
;***IN IX=TABLE 61,IY=TABLE 7*****
;*****OUTPUT TBLE 7*****
;*****USE REGISTER A,IX,IY*****

0006D7 CD8F05        READTIME:  CALL READTD    ;READ DATA FROM RTC AND CONVERT
;BCD TO ASCII

0006DA 211C09        LD HL,TABLE63

0006DD 1100E7        LD DE,TABLE7

0006E0 012800        LD BC,28H

0006E3 EDB0          LDIR

0006E5 2100E5        LD HL,TABLE61 ;TABLE ASCII

0006E8 FD2100E7      LD IY,TABLE7 ;TIME/DATE TABLE

0006EC 7E          LD A,(HL)      ;HIGH BYTE OF HOUR

0006ED FD7709        LD (IY+09H),A ;

```

```

0006F0 23          INC HL
0006F1 7E          LD A,(HL)      ;LOW BYTE OF HOUR
0006F2 FD770A      LD (IY+0AH),A ;
0006F5 23          INC HL
0006F6 7E          LD A,(HL)      ;HIGH BYTE OF MINUTE
0006F7 FD770C      LD (IY+0CH),A
0006FA 23          INC HL
0006FB 7E          LD A,(HL)      ;LOW BYTE OF MINUTE
0006FC FD770D      LD (IY+0DH),A
0006FF 23          INC HL
000700 7E          LD A,(HL)
000701 FD770F      LD (IY+0FH),A ;
000704 23          INC HL
000705 7E          LD A,(HL)      ;LOW BYTE OF SECOND
000706 FD7710      LD (IY+010H),A
000709 C9          RET
;*****SUBROUTINE READ DATE*****
;**IN IX=TABLE 61,IY=TABLE 7,OUT=TABLE7***
;*****USE REGISTER A,IX,IY*****

00070A CD8F05      READDATE:      CALL READTD  ;READ DATA FROM RTC CONVERT
                                           ; BCD TO ASCII

00070D 21F408      LD HL,TABLE62
000710 1100E7      LD DE,TABLE7
000713 012800      LD BC,28H
000716 ED80        LDIR

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000718 2100E5	LD HL, TABLE61 ;ASCII TABLE TABLE POINTER
00071B 110600	LD DE, 0006H
00071E 19	ADD HL, DE
00071F FD2100E7	LD IY, TABLE7
000723 7E	LD A, (HL) ;HIGH BYTE OF DATE OF MONTH
000724 FD7709	LD (IY+09H), A ;
000727 23	INC HL
000728 7E	LD A, (HL) ;LOW BYTE OF DATE OF MONTH
000729 FD770A	LD (IY+0AH), A ;
00072C 23	INC HL
00072D 7E	LD A, (HL) ;HIGH BYTE OF MONTH
00072E FD770C	LD (IY+0CH), A ;
000731 23	INC HL
000732 7E	LD A, (HL) ;LOW BYTE OF MONTH
000733 FD770D	LD (IY+0DH), A
000736 23	INC HL
000737 7E	LD A, (HL) ;HIGH BYTE OF YEAR
000738 FD770F	LD (IY+0FH), A ;
00073B 23	INC HL
00073C 7E	LD A, (HL) ;LOW BYTE OF YEAR
00073D FD7710	LD (IY+010H), A ;
000740 C9	RET

*****TABLE IN BUFF..MEMORY*****

0000E000 = TEMPO: EQU 0E000H ;BUFFER TEMPOLARY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0000E100	=	RAMDISELAY:	EQU	0E100H	;BUFFER FOR DISPLAY
0000E190	=	COUNTER:	EQU	0E190H	;BUFFER NUMBER TELEPHON
0000E195	=	SEQUE:	EQU	0E195H	;PASS WORD STEP PART
0000E198	=	KEYTEL:	EQU	0E198H	;DATA KEYTELEPHON
0000E19A	=	DDRAM:	EQU	0E19AH	;ADDRESS DD RAM
0000E200	=	KEYIN:	EQU	0E200H	;DATA KEYCODE
0000E202	=	NUMTEL:	EQU	0E202H	;NUMBER TIMES FOR CALL
0000E204	=	MONNEY:	EQU	0E204H	;MONNEY BUFFER
0000E20F	=	DATAKEY:	EQU	0E20FH	;DATA KEY ASCII CODE
0000E300	=	TABLE6:	EQU	0E300H	;RTC BUFFER TABLE
0000E400	=	TABLE60:	EQU	0E400H	;BCD BUFFER TABLE
0000E500	=	TABLE61:	EQU	0E500H	;ASCII BUFFER TABLE
0000E700	=	TABLE7:	EQU	0E700H	;TIME/DATE BUFFER TABLE
0000E800	=	TABLE8:	EQU	0E800H	;SERVICE & NUMBER TELEPHONE
;*****TABLE IN THE ROM*****					
000741 2A54484953		TAB11:	DFB"	*THIS"	;TABLE DEMO
000746 204953			DFB"	IS"	
000749 2050524F4A			DFB"	PROJ"	
00074E 454354			DFB"	ECT"	
000751 2032202A			DFB"	2 *	
000755 20205445			DFB"	TE"	
000759 4C455048			DFB"	LEPH"	
00075D 4F4E2053			DFB"	ON S"	
000761 45525649			DFB"	ERVI"	
000765 43452020			DFB"	CE "	

000769 20202020	TAB12:	DFB"	"
00076D 20202020		DFB"	"
000771 42592020		DFB"BY	"
000775 20202020		DFB"	"
000779 20202020		DFB"	"
00077D 20312E20		DFB"	1. "
000781 532E2043		DFB"S. C"	
000785 48494E41		DFB"HINA"	
000789 5048414E		DFB"PHAN"	
00078D 20202020		DFB"	"
000791 20202020	TAB13:	DFB"	"
000795 20322E20		DFB"	2. "
000799 502E2053		DFB"P. S"	
00079D 5557494E		DFB"UWIN"	
0007A1 20202020		DFB"	"
0007A5 20202020		DFB"	"
0007A9 20332E20		DFB"	3. "
0007AD 4A2E2054		DFB"J. T"	
0007B1 41524457		DFB"ARDW"	
0007B5 4F4E4720		DFB"ONG	"
0007B9 20202020		DFB"	"
0007BD 20202020	TAB14:	DFB"	"
0007C1 20202041		DFB"	A"
0007C5 44564953		DFB"DVIS"	
0007C9 45522020		DFB"ER	"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0007CD 20202020	DFB" "
0007D1 4D722E41	DFB"Mr.A"
0007D5 52545441	DFB"RTTA"
0007D9 53495420	DFB"SIT "
0007DD 4C415341	DFB"LASA"
0007E1 4B554C20	DFB"KUL "
0007E5 20202020	TAB15: DFB" "
0007E9 204B204D	DFB" K M"
0007ED 20492054	DFB" I T"
0007F1 204C2020	DFB" L "
0007F5 20202020	DFB" "
0007F9 496E6475	DFB"Indu"
0007FD 73747261	DFB"stra"
000801 696C5465	DFB"ilTe"
000805 63686E6F	DFB"chmo"
000809 6C6F6779	DFB"logy"
00080D 206C2F7437	TAB2: DFB" l/t7" ;TABLE KEYBOARD
000812 2064	DFB" d"
000814 2020692C71	DFB" i,q"
000819 3479	DFB"4y"
00081B 6120206F32	DFB"a o2"
000820 7720	DFB"w "
000822 2067203B6A	DFB" g ;j"
000827 2D72	DFB"-r"
000829 357A622020	DFB"5zb "

00082E 7033	DFB"p3"
000830 7820206820	DFB"x h "
000835 3D6B	DFB"=k"
000837 2E73362063	DFB".s6 c"
00083C 2020	DFB" "
00083E 6E31763920	DFB"n1v9 "
000843 6620	DFB"f "
000845 206D307538	DFB" m0u8"
00084A 206520	DFB" e "
00084D 204C3F5426	DFB" L?T&"
000852 204420	DFB" D "
000855 16	DFB 22
000856 49	DFB"l"
000857 3C51245941	DFB"<QsYA"
00085C 20204F4057	DFB" 06W"
000861 202047203A	DFB" G :"
000866 4A5F52255A	DFB"J-R%Z"
00086B 4220205023	DFB"B P#"
000870 5820204820	DFB"x H "
000875 2B4B3E535E	DFB"+K>S^"
00087A 204320204E	DFB" C N".
00087F 2156282046	DFB"!V(F"
000884 20204D2955	DFB" M)U"
000889 2A20452020	DFB"* E "
00088E 2031323334	TAB3: DFB" 1234"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000893	3536373839		DFB"56789"
000898	302A234142		DFB"0*#AB"
00089D	4344		DFB"CD"
00089F	2A454E5445	TAB51:	DFB"*ENTE" ;ENTER NAME
0008A4	52204E414D		DFB"R NAM"
0008A9	452054454E		DFB"E TEN"
0008AE	414E54202A		DFB"ANT #"
0008B3	4E414D453A	TAB52:	DFB"NAME:"
0008B8	2020202020	TAB53:	DFB" "
0008BD	2020202020		DFB" "
0008C2	2020202020		DFB" "
0008C7	426174682E		DFB"Bath."
0008CC	54454C2E2E	TAB54:	DFB"TEL.."
0008D1	2020202020		DFB" "
0008D6	2020202020		DFB" "
0008DB	2020202020		DFB" "
0008E0	2020202020		DFB" "
0008E5	2020202020		DFB" "
0008EA	2020202020		DFB" "
0008EF	2020202020		DFB" "
0008F4	4F4C442044	TABLE62:	DFB"OLD D" ;DISPLAY BUFFER TABLE
0008F9	4154452020		DFB"ATE "
0008FE	202F20202F		DFB" / /"
000903	2020202020		DFB" "
000908	4E45572044		DFB"NEW D"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

00090D 4154452020	DFB"ATE "
000912 2020202020	DFB" "
000917 2020202020	DFB" "
00091C 4F4C442054	TABLE63: DFB"OLD T"
000921 494D452020	DFB"IME "
000926 203A20203A	DFB" : :"
00092B 2020202020	DFB" "
000930 4E45572054	DFB"NEW T"
000935 494D452020	DFB"IME "
00093A 2020202020	DFB" "
00093F 2020202020	DFB" "
000944 202030202033	TABLE9: DFB" 0 3"
00094A 202036202039	DFB" 6 9"
000950 203132203135	DFB" 12 15"
000956 203138203231	DFB" 18 21"
00095C 203234203237	DFB" 24 27"
000962 203330203333	DFB" 30 33"
000968 203336203339	DFB" 36 39"
00096E 203432203435	DFB" 42 45"
000974 203438203531	DFB" 48 51"
00097A 203534203537	DFB" 54 57"
000980 203630203633	DFB" 60 63"
000986 203636203639	DFB" 66 69"
00098C 203732203735	DFB" 72 75"
000992 203738203731	DFB" 78 71"

000998 203734203737	DFB" 74 77"
00099E 203830203833	DFB" 80 83"
0009A4 203836203839	DFB" 86 89"
0009AA 203932203935	DFB" 92 95"
0009B0 203938313031	DFB" 98101"
0009B6 313034313037	DFB"104107"
0009BC 313130313133	DFB"110113"
0009C2 313136313139	DFB"116119"
0009C8 313232313235	DFB"122125"
0009CE 313238313331	DFB"128131"
0009D4 313334313337	DFB"134137"
0009DA 313430313433	DFB"140143"
0009E0 313436313439	DFB"146149"
0009E6 313532313535	DFB"152155"
0009EC 313538313631	DFB"158161"
0009F2 313634313637	DFB"164167"
0009F8 313730313733	DFB"170173"
0009FE 313736313739	DFB"176179"
000A04 313832313835	DFB"182185"
000A0A 313838313931	DFB"188191"
000A10 313934313937	DFB"194197"
000A16 323030323033	DFB"200203"
000A1C 323036323039	DFB"206209"
000A22 323132323135	DFB"212215"
000A28 323138323231	DFB"218221"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000A2E 323233323236	DFB"223226"
000A34 323239323331	DFB"229231"
000A3A 323334323337	DFB"234237"
000A40 323430323433	DFB"240243"
000A46 323436323439	DFB"246249"
000A4C 323532323535	DFB"252255"
000A52 323538323631	DFB"258261"
000A58 323634323637	DFB"264267"
000A5E 323730323733	DFB"270273"
000A64 323736323739	DFB"276279"
000A6A 323831323834	DFB"281284"
000A70 323837323930	DFB"287290"
000A76 323933323936	DFB"293296"
000A7C 323939333032	DFB"299302"
000A82 333035333038	DFB"305308"
000A88 333131333134	DFB"311314"
000A8E 333137333230	DFB"317320"
000A94 333233333236	DFB"323326"
000A9A 333239333332	DFB"329332"
000AA0 333335333338	DFB"335338"
000AA6 333431333434	DFB"341344"
000AAC 333437333530	DFB"347350"
000AB2 333533333536	DFB"353356"
000AB8 333539333632	DFB"359362"
000ABE 333635333638	DFB"365368"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000AC4 333731333734	DFB"371374"
000ACA 333737333830	DFB"377380"
000AD0 333833333836	DFB"383386"
000AD6 333839333932	DFB"389392"
000ADC 333935333938	DFB"395398"
000AE2 343031343034	DFB"401404"
000AE8 343037343130	DFB"407410"
000AEE 343133343136	DFB"413416"
000AF4 343139343232	DFB"419422"
000AFA 343235343238	DFB"425428"
000B00 343331343334	DFB"431434"
000B06 343337343430	DFB"437440"
000B0C 343433343436	DFB"443446"
000B12 343439343532	DFB"449452"
000B18 343535343538	DFB"455458"
000B1E 343631343634	DFB"461464"
000B24 343637343530	DFB"467450"
000B2A 343533343536	DFB"453456"
000B30 343539343632	DFB"459462"
000B36 343635343638	DFB"465468"
000B3C 343731343734	DFB"471474"
000B42 343737343830	DFB"477480"
000B48 343833343836	DFB"483486"
000B4E 343839343932	DFB"489492"
000B54 343935343938	DFB"495498"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000B5A 353031353034	DFB"501504"
000B60 353037353130	DFB"507510"
000B66 353133353136	DFB"513516"
000B6C 353139353232	DFB"519522"
000B72 353235353238	DFB"525528"
000B78 353331353334	DFB"531534"
000B7E 353337353430	DFB"537540"
000B84 353433353436	DFB"543546"
000B8A 353439353532	DFB"549552"
000B90 353535353538	DFB"555558"
000B96 353631353634	DFB"561564"
000B9C 353637353730	DFB"567570"
000BA2 353733353736	DFB"573576"
000BA8 353739353832	DFB"579582"
000BAE 353835353838	DFB"585588"
000BB4 353931353934	DFB"591594"
000BBA 353937363030	DFB"597600"
000BC0 363033363036	DFB"603606"
000BC6 363039363132	DFB"609612"
000BCC 363135363138	DFB"615618"
000BD2 363231363234	DFB"621624"
000BD8 363237363330	DFB"627630"
000BDE 363333363336	DFB"633636"
000BE4 363339363432	DFB"639642"
000BEA 363435363438	DFB"645648"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000BF0 363531363534	DFB"651654"
000BF6 363537363630	DFB"657660"
000BFC 363633363636	DFB"663666"
000C02 363639363732	DFB"669672"
000C08 363735363738	DFB"675678"
000C0E 363831363834	DFB"681684"
000C14 363837363930	DFB"687690"
000C1A 363933363936	DFB"693696"
000C20 363939373032	DFB"699702"
000C26 373035373038	DFB"705708"
000C2C 373131373134	DFB"711714"
000C32 373137373230	DFB"717720"
000C38 373233373236	DFB"723726"
000C3E 373239373332	DFB"729732"
000C44 373335373338	DFB"735738"
000C4A 373431373434	DFB"741744"
000C50 373437373530	DFB"747750"
000C56 373533373536	DFB"753756"
000C5C 373539373632	DFB"759762"
000C62 373635373638	DFB"765768"
000C68 373731373734	DFB"771774"
000C6E 373737373830	DFB"777780"
000C74 373833373836	DFB"783786"
000C7A 373839373932	DFB"789792"
000C80 373935373938	DFB"795798"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000C86 383031383034	DFB"801804"
000C8C 383037383130	DFB"807810"
000C92 383133383136	DFB"813816"
000C98 383139383232	DFB"819822"
000C9E 383235383238	DFB"825828"
000CA4 383331383334	DFB"831834"
000CAA 383337383430	DFB"837840"
000CB0 383433383436	DFB"843846"
000CB6 383439383532	DFB"849852"
000CBC 383535383538	DFB"855858"
000CC2 383631383634	DFB"861864"
000CC8 383637383730	DFB"867870"
000CCE 383733383736	DFB"873876"
000CD4 383739383832	DFB"879882"
000CDA 383835383838	DFB"885888"
000CEO 383931383934	DFB"891894"
000CE6 383937393030	DFB"897900"
000CEC 393033393036	DFB"903906"
000CF2 393039393132	DFB"909912"
000CF8 393135393138	DFB"915918",
000CFE 393231393234	DFB"921924"
000D04 393237393330	DFB"927930"
000DOA 393333393336	DFB"933936"
000D10 393339393432	DFB"939942"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

000D16 393435393438	DFB"945948"
000D1C 393531393534	DFB"951954"
000D22 393537393630	DFB"957960"
000D28 393633393636	DFB"963966"
000D2E 393639393732	DFB"969972"
000D34 393735393738	DFB"975978"
000D3A 393831393834	DFB"981984"
000D40 393837393930	DFB"987990"
000D46 393933393936	DFB"993996"
000D4C 393939	DFB"999"



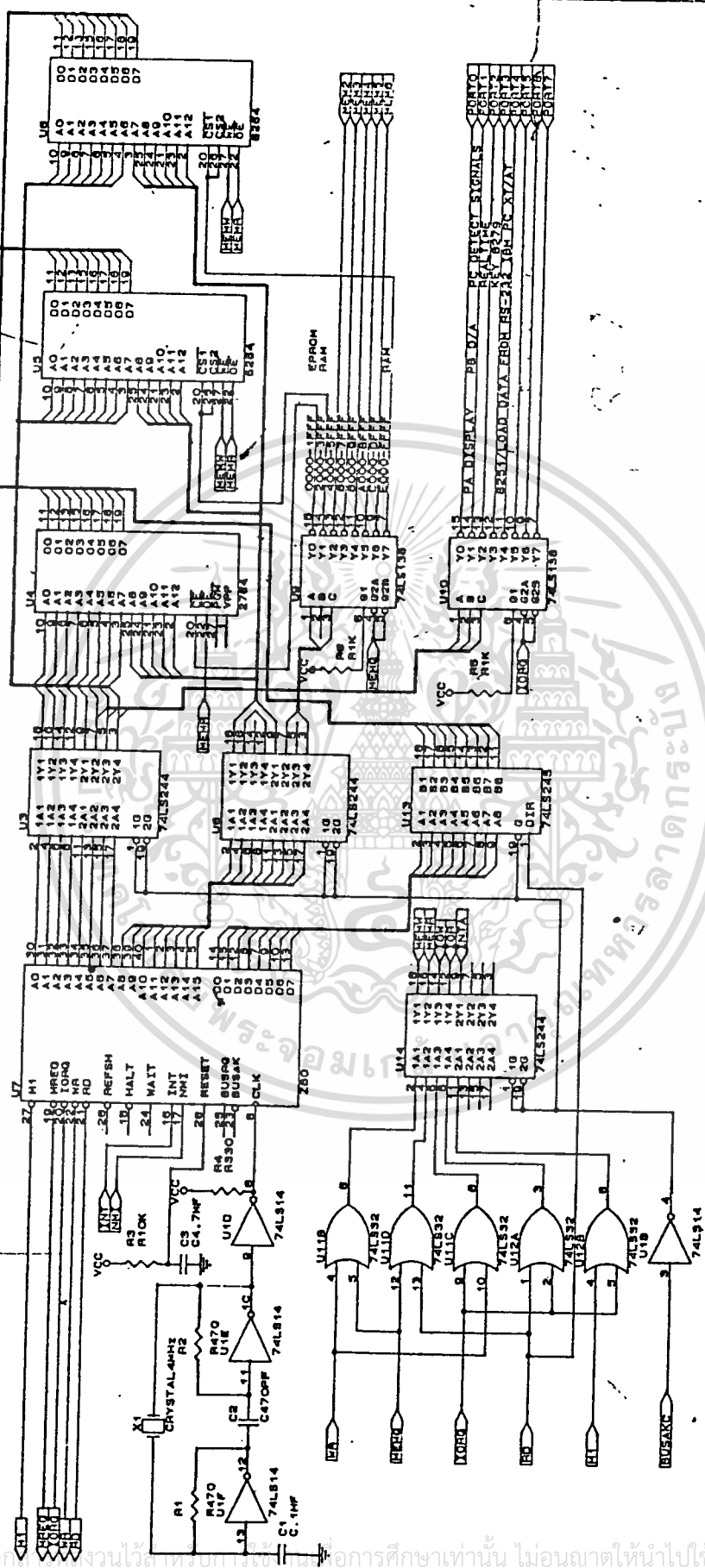
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

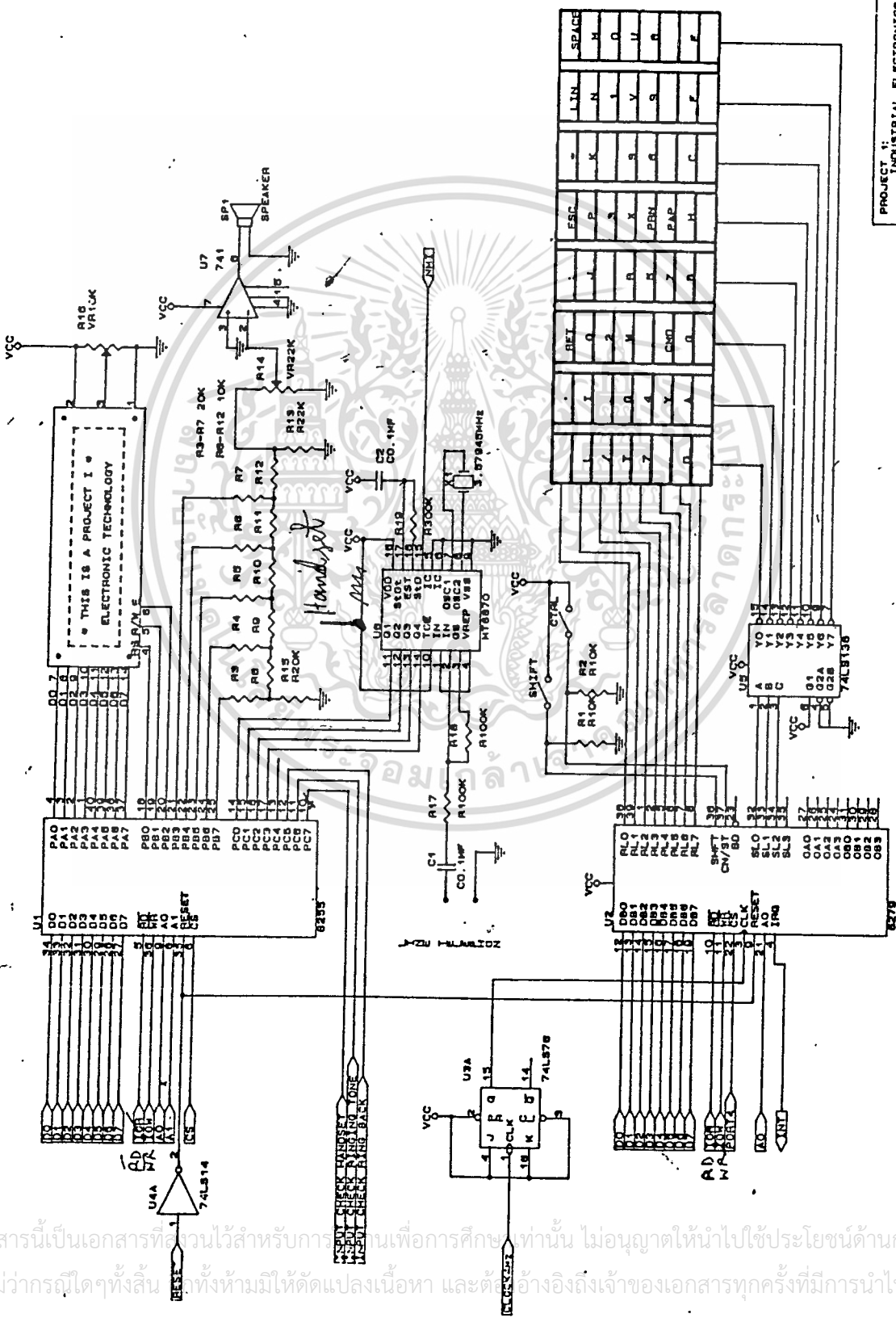
000003EF	A1	00000613	A2HEX	0000061C	A2HEX1
000001A1	AGAIN	000003AF	ALIST	00000355	BACKTAB
0000036F	BENAME	0000000F	BLANK	00000048	BLANK1
0000047D	CLEAR	00000262	CLRLCD	00000484	CLRRAM
000005DD	CON1	00000600	CON2	0000021F	COUNT
0000022B	COUNT1	0000E190	COUNTER	0000E20F	DATAKEY
0000E19A	DDRAM	000002E3	DE1	000002E0	DELAY
0000046F	DELAY1	0000028B	DISPLAY	000002C0	ED1
00000126	EN1	0000012E	EN2	000002B7	EPLUSE
00000392	GONAME	000004D8	GORETURN1	00000551	GORETURN2
000002C8	GOTO	0000062F	HBM	0000067E	HBN
00000237	INITIAL	00000073	INKEY1	000002E9	INTKEY
000002EF	KEYBOARD	0000045E	KEYDATA	00000324	KEYFUNCTION
0000E200	KEYIN	00000221	KEYN	0000022D	KEYN1
00000157	KEYNUMBER	0000E198	KEYTEL	00000640	LBD
00000695	LBN	0000064D	LBM	000003AD	LINEFEED
0000000A	LOOP	00000474	LOOP1	00000194	M1
0000019A	M2	0000E204	MONNEY	0000048D	MOVES
0000018F	MUL	00000183	N	0000018C	N1
0000035F	NAME	000005F6	NASCII	000005FD	NASI
0000047E	NEWCLR	000004B3	NEWLOOP1	00000529	NEWLOOP2
000004B5	NEWSET1	0000052B	NEWSET2	0000021A	NEY
0000E202	NUMTEL	0000065A	PO	000006A2	POG
00000000	POWER	00000001	POWERI	0000034B	PRINT1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

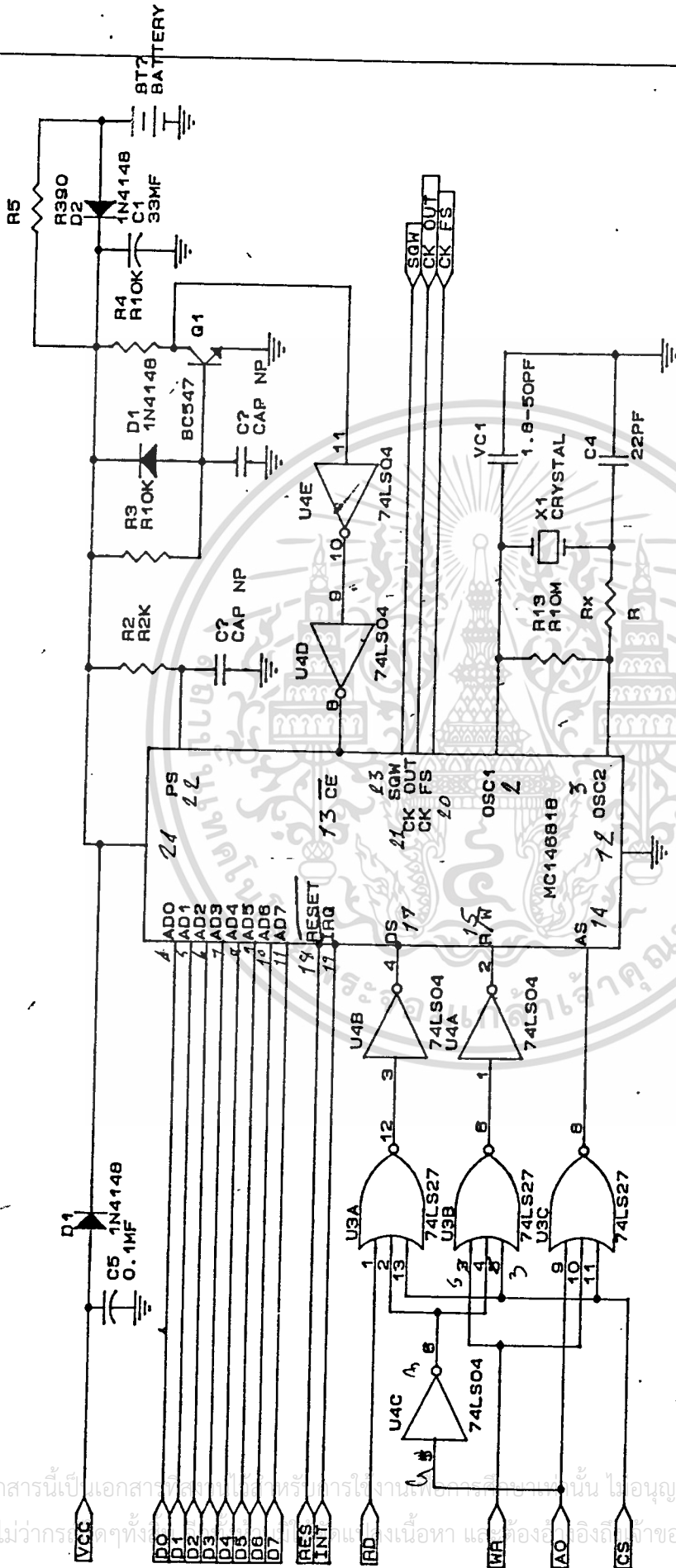
00000353	PRINT2	000004DD	Q1	00000582	Q2
00000556	Q21	0000E100	RAMDISPLAY	000005D6	RDBYTE
0000059A	READ1	0000070A	READDATE	0000058F	READTD
000006D7	READTIME	00000140	RESETNMI	00000418	RETURN
0000009A	S1	00000093	S2	00000095	S3
000000DA	S4	000000AE	S5	0000E195	SEQUE
0000006A	SERVICE	000006B4	SET	000000CE	SETADDR
0000019E	START	000000E8	START1	00000414	STARTALIST
000003AE	STARTLINEFEED	00000397	STARTNAME	00000352	STARTPRINT1
00000354	STARTPRINT2	0000034A	STARTTIMDATE	0000014B	STOPS1
0000011D	STP1	00000509	T3	00000741	TAB11
00000769	TAB12	00000791	TAB13	000007BD	TAB14
000007E5	TAB15	0000080D	TAB2	0000088E	TAB3
0000089F	TAB51	000008B3	TAB52	000008B8	TAB53
000008CC	TAB54	0000E300	TABLE6	0000E400	TABLE60
0000E500	TABLE61	000008F4	TABLE62	0000091C	TABLE63
0000E700	TABLE7	0000E800	TABLE8	00000944	TABLE9
00000178	TABLENUM	0000E000	TEMPO	0000061D	TESTDATE
0000066C	TESTTIME	00000490	TIMEDATE	000002D5	WRBYTE
000006D0	WRBYTE1	000006C0	WRITE	000006C7	WRITE1
0000016B	WRITELCD	000002AD	WRL	0000029F	WRL1
000002A6	WRL2	00000296	WRLINE	000002AB	WRLM
00000657	X31	00000692	X32	000006AD	Y1
0000050C	Y2	00000665	Z1	00000497	Z2

Handwritten notes and annotations on the left side of the schematic, including a large arrow pointing right labeled "26" and "45V", and a smaller arrow pointing left labeled "15V".



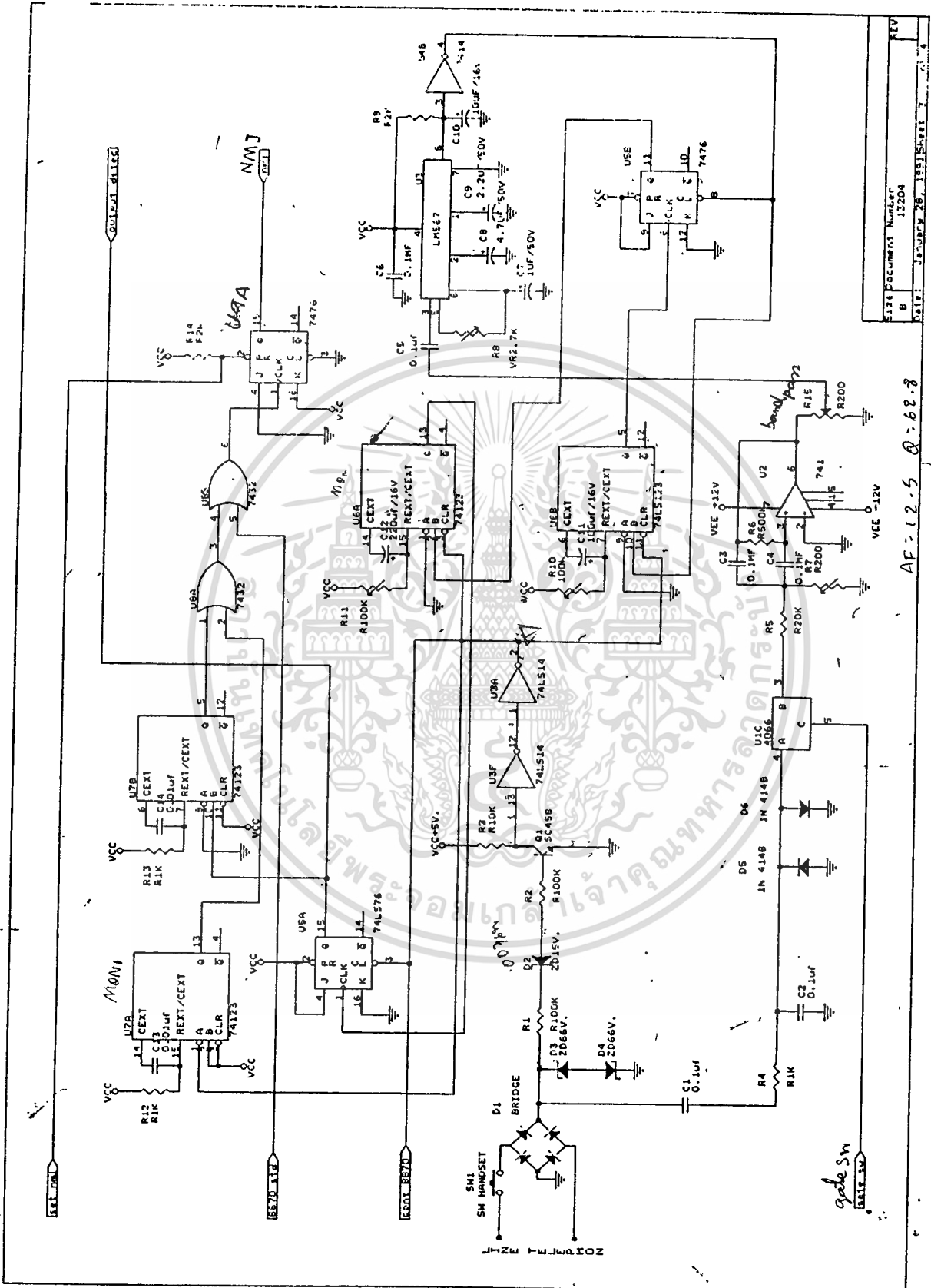


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น หากท่านมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



PROJECT 1:	
INDUSTRIAL ELECTRONICS TECHNOLOGY	
Size Document Number 13204	
A	REV
Date: January 5, 1980	Sheet 4 of 4

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานที่ออกจำหน่ายเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าการคัดลอกทั้งฉบับหรือบางส่วนโดยไม่ได้รับอนุญาตจากเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



REV	DATE	SHEET	OF
1	1985	1	1

AF-12-5 Q-68.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวกที่ 2

DATA SHEET Z-80,8279,LCD



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings

Temperature Under Bias
Storage Temperature
Voltage On Any Pin
with Respect to Ground
Power Dissipation

Specified operating range:
-55°C to +150°C
-0.5V to +7V

1.5W

*Comment

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

Note For Z80-CPU all AC and DC characteristics remain the same for the military grade parts except I_{CC} .

 $I_{CC} = 200 \text{ mA}$

Z80-CPU D.C. Characteristics

$T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5\text{V} \pm 5\%$ unless otherwise specified

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	-0.3		0.45	V	
V_{IHC}	Clock Input High Voltage	$V_{CC} - 0.5$		$V_{CC} + 0.3$	V	
V_{IL}	Input Low Voltage	-0.3		0.3	V	
V_{IH}	Input High Voltage	2.0		V_{CC}	V	
V_{OL}	Output Low Voltage			0.4	V	$I_{OL} = 1.3 \text{ mA}$
V_{OH}	Output High Voltage	2.4			V	$I_{OH} = -250 \mu\text{A}$
I_{CC}	Power Supply Current			150	mA	
I_{LI}	Input Leakage Current			10	μA	$V_{IN} = 0 \text{ to } V_{CC}$
I_{LOH}	Tri-State Output Leakage Current in Float			10	μA	$V_{OUT} = 2.4 \text{ to } V_{CC}$
I_{LOL}	Tri-State Output Leakage Current in Float			-10	μA	$V_{OUT} = 0.4 \text{ V}$
I_{LD}	Data Bus Leakage Current in Input Mode			± 10	μA	$0 < V_{IN} < V_{CC}$

Capacitance

$T_A = 25^\circ\text{C}$, $f = 1 \text{ MHz}$.

unmeasured pins returned to ground

Symbol	Parameter	Max.	Unit
C_ϕ	Clock Capacitance	35	pF
C_{IN}	Input Capacitance	5	pF
C_{OUT}	Output Capacitance	10	pF

Z80-CPU

Ordering Information

C - Ceramic

P - Plastic

S - Standard $5\text{V} \pm 5\%$ 0° to 70°C E - Extended $5\text{V} \pm 5\%$ -40° to 85°C M - Military $5\text{V} \pm 10\%$ -55° to 125°C

Z80A-CPU D.C. Characteristics

$T_A = 0^\circ\text{C}$ to 70°C , $V_{CC} = 5\text{V} \pm 5\%$ unless otherwise specified

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Condition
V_{ILC}	Clock Input Low Voltage	-0.3		0.45	V	
V_{IHC}	Clock Input High Voltage	$V_{CC} - 0.5$		$V_{CC} + 0.3$	V	
V_{IL}	Input Low Voltage	-0.3		0.3	V	
V_{IH}	Input High Voltage	2.0		V_{CC}	V	
V_{OL}	Output Low Voltage			0.4	V	$I_{OL} = 1.3 \text{ mA}$
V_{OH}	Output High Voltage	2.4			V	$I_{OH} = -250 \mu\text{A}$
I_{CC}	Power Supply Current		90	200	mA	
I_{LI}	Input Leakage Current			10	μA	$V_{IN} = 0 \text{ to } V_{CC}$
I_{LOH}	Tri-State Output Leakage Current in Float			10	μA	$V_{OUT} = 2.4 \text{ to } V_{CC}$
I_{LOL}	Tri-State Output Leakage Current in Float			-10	μA	$V_{OUT} = 0.4 \text{ V}$
I_{LD}	Data Bus Leakage Current in Input Mode			± 10	μA	$0 < V_{IN} < V_{CC}$

Capacitance

$T_A = 25^\circ\text{C}$, $f = 1 \text{ MHz}$.

unmeasured pins returned to ground

Symbol	Parameter	Max.	Unit
C_ϕ	Clock Capacitance	35	pF
C_{IN}	Input Capacitance	4	pF
C_{OUT}	Output Capacitance	10	pF

Z80A-CPU

Ordering Information

C - Ceramic

P - Plastic

S - Standard $5\text{V} \pm 5\%$ 0° to 70°C

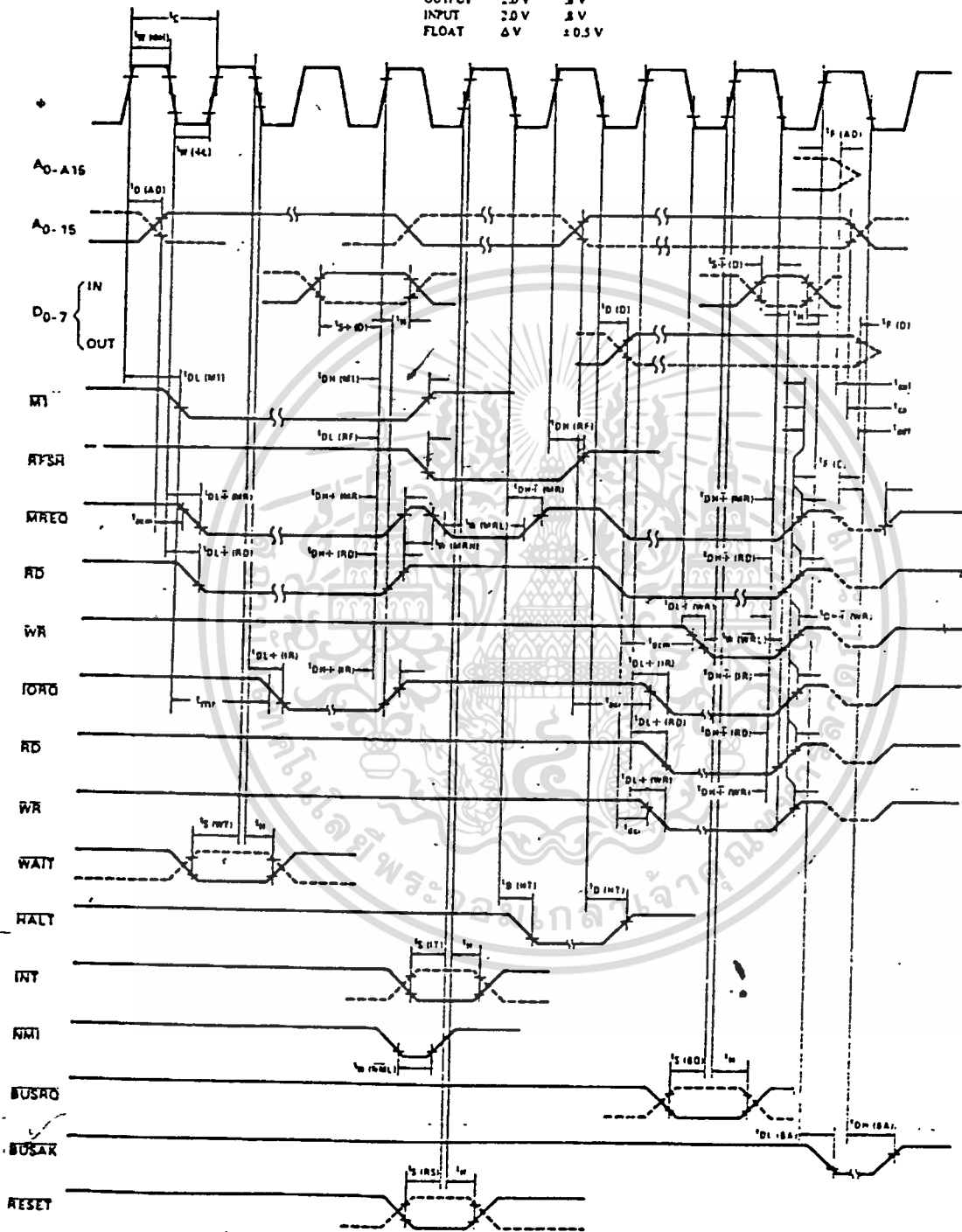
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A.C. Timing Diagram

Timing measurements are made at the following voltages, unless otherwise specified:

	-1"	-0"
CLOCK	V _{CC} -0.6V	.45V
OUTPUT	2.0V	.8V
INPUT	2.0V	.8V
FLOAT	ΔV	±0.5V



A.C. Characteristics

Z80-CPU

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

Signal	Symbol	Parameter	Min	Max	Unit	Test Condition
ϕ	t_{CPH} t_{CPL} t_{CH}	Clock Period Clock Pulse Width, Clock High Clock Pulse Width, Clock Low Clock Rise and Fall Time	4 180 180	1121 [5] 2000	nsec nsec nsec	
A_{0-15}	$t_{D(AD)}$ $t_F(AD)$ t_{ACM} t_{AS} t_{AS}	Address Output Delay Delay to Float Address Stable Prior to $MREQ$ (Memory Cycle) Address Stable Prior to $IORQ$, RD or WR (IO Cycle) Address Stable from RD , WR , $IORQ$ or $MREQ$ Address Stable from RD or WR During Float		145 110 [11] [12] [13] [14]	nsec nsec nsec nsec nsec nsec	$C_L = 50pF$
D_{0-7}	$t_{D(D)}$ $t_F(D)$ $t_{SD}(D)$ $t_{SD}(D)$ t_{dcm} t_{dcl} t_{dfl}	Data Output Delay Delay to Float During Write Cycle Data Setup Time to Rising Edge of Clock During M1 Cycle Data Setup Time to Falling Edge of Clock During M2 to M5 Data Stable Prior to WR (Memory Cycle) Data Stable Prior to WR (IO Cycle) Data Stable from WR		230 60 50 60 151 161 171	nsec nsec nsec nsec nsec nsec nsec	$C_L = 50pF$
	t_H	Any Hold Time for Setup Time	0		nsec	
$MREQ$	$t_{DL}(MRL)$ $t_{DH}(MRL)$ $t_{DL}(MRL)$ t_{dcm} t_{dfl}	$MREQ$ Delay From Falling Edge of Clock, $MREQ$ Low $MREQ$ Delay From Rising Edge of Clock, $MREQ$ High $MREQ$ Delay From Falling Edge of Clock, $MREQ$ High Pulse Width, $MREQ$ Low Pulse Width, $MREQ$ High		100 100 100 [15] [16]	nsec nsec nsec nsec nsec	$C_L = 50pF$
$IORQ$	$t_{DL}(IRL)$ $t_{DH}(IRL)$ $t_{DL}(IRL)$ $t_{DH}(IRL)$	$IORQ$ Delay From Rising Edge of Clock, $IORQ$ Low $IORQ$ Delay From Falling Edge of Clock, $IORQ$ Low $IORQ$ Delay From Rising Edge of Clock, $IORQ$ High $IORQ$ Delay From Falling Edge of Clock, $IORQ$ High		90 110 100 110	nsec nsec nsec nsec	$C_L = 50pF$
RD	$t_{DL}(RDL)$ $t_{DH}(RDL)$ $t_{DL}(RDL)$ $t_{DH}(RDL)$	RD Delay From Rising Edge of Clock, RD Low RD Delay From Falling Edge of Clock, RD Low RD Delay From Rising Edge of Clock, RD High RD Delay From Falling Edge of Clock, RD High		100 110 100 110	nsec nsec nsec nsec	$C_L = 50pF$
WR	$t_{DL}(WRL)$ $t_{DH}(WRL)$ $t_{DL}(WRL)$ t_{dcm} t_{dfl}	WR Delay From Rising Edge of Clock, WR Low WR Delay From Falling Edge of Clock, WR Low WR Delay From Rising Edge of Clock, WR High Pulse Width, WR Low		80 90 100 [10]	nsec nsec nsec nsec	$C_L = 50pF$
$M1$	$t_{DL}(M1L)$ $t_{DH}(M1L)$	$M1$ Delay From Rising Edge of Clock, $M1$ Low $M1$ Delay From Rising Edge of Clock, $M1$ High		130 130	nsec nsec	$C_L = 50pF$
$RFSH$	$t_{DL}(RFL)$ $t_{DH}(RFL)$	$RFSH$ Delay From Rising Edge of Clock, $RFSH$ Low $RFSH$ Delay From Rising Edge of Clock, $RFSH$ High		180 150	nsec nsec	$C_L = 50pF$
$WAIT$	$t_s(WT)$	$WAIT$ Setup Time to Falling Edge of Clock	70		nsec	
$HALT$	$t_D(HT)$	$HALT$ Delay Time From Falling Edge of Clock		300	nsec	$C_L = 50pF$
INT	$t_s(INT)$	INT Setup Time to Rising Edge of Clock	80		nsec	
NMI	$t_w(NML)$	Pulse Width, NMI Low	80		nsec	
$BUSRQ$	$t_s(BQ)$	$BUSRQ$ Setup Time to Rising Edge of Clock	80		nsec	
$BUSAK$	$t_{DL}(BAL)$ $t_{DH}(BAL)$	$BUSAK$ Delay From Rising Edge of Clock, $BUSAK$ Low $BUSAK$ Delay From Falling Edge of Clock, $BUSAK$ High		120 110	nsec nsec	$C_L = 50pF$
$RFSH$	$t_s(RS)$	$RFSH$ Setup Time to Rising Edge of Clock	90		nsec	
	$t_F(C)$	Delay to Float ($MREQ$, $IORQ$, RD and WR)		100	nsec	
	t_{MV}	$M1$ Stable Prior to $IORQ$ (Interrupt Ack)	1111		nsec	

$$[12] t_c = t_w(\phi H) + t_w(\phi L) + t_f + t_r$$

$$[11] t_{acm} = t_w(\phi H) + t_f + t_r$$

$$[2] t_{dcl} = t_c - 80$$

$$[3] t_{ca} = t_w(\phi L) + t_f + t_r - 40$$

$$[4] t_{dfl} = t_w(\phi L) + t_f + t_r - 40$$

$$[5] t_{dcm} = t_c - 210$$

$$[6] t_{ddl} = t_w(\phi L) + t_f + t_r - 210$$

$$[7] t_{cdf} = t_w(\phi L) + t_f + t_r - 80$$

$$[8] t_w(MRL) = t_c - 40$$

$$[9] t_w(MRH) = t_w(\phi H) + t_f + t_r - 30$$

$$[10] t_w(WRL) = t_c - 40$$

$$[11] t_{mv} = 2t_c + t_w(\phi H) + t_f + t_r - 80$$

NOTES

A Data should be enabled when the CPU data bus when RD is active. During interrupt acknowledge data should be enabled when $M1$ and $IORQ$ are both active.

B All control signals are internally synchronized, so they may be mutually asynchronous with respect to the clock.

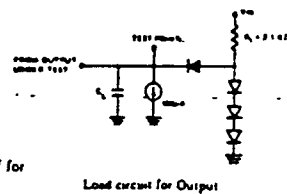
C The $RESET$ signal must be active for a minimum of 3 clock cycles.

D Output Delay vs. Load Capacitance

$T_A = 70^\circ\text{C}$, $V_{CC} = +5V \pm 5\%$.

Add 10nsec delay for each 50pF increase in load up to a maximum of 200pF for the data bus & 100pF for address & control lines.

E Although static, by design, testing parameters $t_w(\phi H)$ of 200nsec maximum.



A.C. Characteristics

Z80-CPU

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

Signal	Symbol	Parameter	Min	Max	Unit	Test Condition
ϕ	$t_{CH}(\phi H)$ $t_{WH}(\phi H)$ $t_{CL}(\phi L)$ t_{FL}	Clock Period Clock Pulse Width, Clock High Clock Pulse Width, Clock Low Clock Rise and Fall Time	4	1121	μsec	
A_{0-15}	$t_{D(AD)}$ $t_{F(AD)}$ t_{ACM} t_{AS} t_{AH} t_{AL}	Address Output Delay Delay to Flast Address Stable Prior to MREQ (Memory Cycle) Address Stable Prior to IORQ, RD, or WR (I/O Cycle) Address Stable from RD, WR, IORQ or MREQ Address Stable from RD or WR During Flast		145 110 111 121 131 141	nsec	$C_L = 50\text{pF}$
D_{0-7}	$t_{D(D)}$ $t_{F(D)}$ $t_{SD}(D)$ $t_{SD}(D)$ t_{ACM} t_{AS} t_{AH} t_{AL}	Data Output Delay Delay to Flast During Write Cycle Data Setup Time to Rising Edge of Clock During M1 Cycle Data Setup Time to Falling Edge of Clock During M2 to M5 Data Stable Prior to WR (Memory Cycle) Data Stable Prior to WR (I/O Cycle) Data Stable from WR		230 50 50 60 131 161 171	nsec	$C_L = 50\text{pF}$
$\overline{H}$		Any Hold Time for Setup Time	0		nsec	
$\overline{MREQ}$	$t_{DL}(\overline{MREQ})$ $t_{DH}(\overline{MREQ})$ $t_{DHL}(\overline{MREQ})$ $t_{WH}(\overline{MREQ})$ $t_{WL}(\overline{MREQ})$	$\overline{MREQ}$ Delay From Falling Edge of Clock, $\overline{MREQ}$ Low $\overline{MREQ}$ Delay From Rising Edge of Clock, $\overline{MREQ}$ High $\overline{MREQ}$ Delay From Falling Edge of Clock, $\overline{MREQ}$ High Pulse Width, $\overline{MREQ}$ Low		100 100 100 181 191	nsec	$C_L = 50\text{pF}$
$\overline{IORQ}$	$t_{DL}(\overline{IORQ})$ $t_{DH}(\overline{IORQ})$ $t_{DHL}(\overline{IORQ})$ $t_{WH}(\overline{IORQ})$	$\overline{IORQ}$ Delay From Rising Edge of Clock, $\overline{IORQ}$ Low $\overline{IORQ}$ Delay From Falling Edge of Clock, $\overline{IORQ}$ High $\overline{IORQ}$ Delay From Rising Edge of Clock, $\overline{IORQ}$ High $\overline{IORQ}$ Delay From Falling Edge of Clock, $\overline{IORQ}$ High		90 110 110 110	nsec	$C_L = 50\text{pF}$
$\overline{RD}$	$t_{DL}(\overline{RD})$ $t_{DH}(\overline{RD})$ $t_{DHL}(\overline{RD})$ $t_{WH}(\overline{RD})$	$\overline{RD}$ Delay From Rising Edge of Clock, $\overline{RD}$ Low $\overline{RD}$ Delay From Falling Edge of Clock, $\overline{RD}$ Low $\overline{RD}$ Delay From Rising Edge of Clock, $\overline{RD}$ High $\overline{RD}$ Delay From Falling Edge of Clock, $\overline{RD}$ High		100 130 100 110	nsec	$C_L = 50\text{pF}$
$\overline{WR}$	$t_{DL}(\overline{WR})$ $t_{DH}(\overline{WR})$ $t_{DHL}(\overline{WR})$ $t_{WH}(\overline{WR})$ $t_{WL}(\overline{WR})$	$\overline{WR}$ Delay From Rising Edge of Clock, $\overline{WR}$ Low $\overline{WR}$ Delay From Falling Edge of Clock, $\overline{WR}$ Low $\overline{WR}$ Delay From Rising Edge of Clock, $\overline{WR}$ High Pulse Width, $\overline{WR}$ Low		80 90 100 110	nsec	$C_L = 50\text{pF}$
$\overline{M1}$	$t_{DL}(\overline{M1})$ $t_{DH}(\overline{M1})$	$\overline{M1}$ Delay From Rising Edge of Clock, $\overline{M1}$ Low $\overline{M1}$ Delay From Rising Edge of Clock, $\overline{M1}$ High		130 130	nsec	$C_L = 50\text{pF}$
$\overline{RFSH}$	$t_{DL}(\overline{RFSH})$ $t_{DH}(\overline{RFSH})$	$\overline{RFSH}$ Delay From Rising Edge of Clock, $\overline{RFSH}$ Low $\overline{RFSH}$ Delay From Rising Edge of Clock, $\overline{RFSH}$ High		180 130	nsec	$C_L = 50\text{pF}$
$\overline{WAIT}$	$t_s(\overline{WAIT})$	$\overline{WAIT}$ Setup Time to Falling Edge of Clock	70		nsec	
$\overline{HALT}$	$t_D(\overline{HALT})$	$\overline{HALT}$ Delay Time From Falling Edge of Clock		300	nsec	$C_L = 50\text{pF}$
$\overline{INT}$	$t_s(\overline{INT})$	$\overline{INT}$ Setup Time to Rising Edge of Clock	80		nsec	
$\overline{NMI}$	$t_s(\overline{NMI})$	Pulse Width, $\overline{NMI}$ Low	80		nsec	
$\overline{BUSRQ}$	$t_s(\overline{BUSRQ})$	$\overline{BUSRQ}$ Setup Time to Rising Edge of Clock	80		nsec	
$\overline{BUSAK}$	$t_{DL}(\overline{BUSAK})$ $t_{DH}(\overline{BUSAK})$	$\overline{BUSAK}$ Delay From Rising Edge of Clock, $\overline{BUSAK}$ Low $\overline{BUSAK}$ Delay From Falling Edge of Clock, $\overline{BUSAK}$ High		120 110	nsec	$C_L = 50\text{pF}$
$\overline{RESET}$	$t_s(\overline{RESET})$	$\overline{RESET}$ Setup Time to Rising Edge of Clock	90		nsec	
	$t_F(C)$	Delay to Flast ($\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$ and $\overline{WR}$)		100	nsec	
	t_{MV}	$\overline{M1}$ Stable Prior to $\overline{IORQ}$ (Interrupt Ack.)	1111		nsec	

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

$$[11] t_{ACM} = t_{WH}(\phi H) + t_f - 75$$

$$[12] t_{AS} = t_c - 40$$

$$[13] t_{AH} = t_{WH}(\phi L) + t_f - 40$$

$$[14] t_{AL} = t_{WH}(\phi L) + t_f - 40$$

$$[15] t_{dcm} = t_c - 210$$

$$[16] t_{dcl} = t_{WH}(\phi L) + t_f - 210$$

$$[17] t_{cdf} = t_{WH}(\phi L) + t_f - 80$$

$$[18] t_{WH}(\overline{MREQ}) = t_c - 40$$

$$[19] t_{WH}(\overline{MREQ}) = t_{WH}(\phi H) + t_f - 30$$

$$[10] t_{WH}(\overline{WR}) = t_c - 40$$

$$[11] t_{MV} = 2t_c + t_{WH}(\phi H) + t_f - 80$$

NOTES

A. Data should be enabled until the CPU data bus when $\overline{RD}$ is active. During interrupt acknowledge data should be enabled when $\overline{M1}$ and $\overline{IORQ}$ are both active.

B. All control signals are internally synchronized so they may be totally asynchronous with respect to the clock.

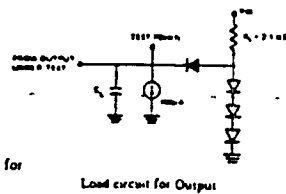
C. The $\overline{RESET}$ signal must be active for a minimum of 3 clock cycles.

D. Output Delay vs. Load Capacitance

$T_A = 70^\circ\text{C}$, $V_{CC} = +5V \pm 5\%$

Add 10nsec delay for each 50pf increase in load up to a maximum of 200pf for the data bus & 100pf for address & control lines.

E. Although stated by design, timing guarantees $t_{WH}(\phi H)$ of 200nsec minimum.



A.C. Characteristics

Z80A-CPU

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

Signal	Symbol	Parameter	Min	Max	Unit	Test Condition
ϕ	t_c $t_{w(\phi H)}$ $t_{w(\phi L)}$ t_r	Clock Period Clock Pulse Width, Clock High Clock Pulse Width, Clock Low Clock Rise and Fall Time	.25 110 110	1121 121 2000	μsec ns ns	
A_{0-15}	$t_D(\text{AD})$ $t_F(\text{AD})$ t_{acm} t_{pc} t_{ca} t_{cdf}	Address Output Delay Delay to Float Address Stable Prior to $\overline{\text{MREQ}}$ (Memory Cycle) Address Stable Prior to IORQ , RD or $\overline{\text{WR}}$ (I/O Cycle) Address Stable From RD , $\overline{\text{WR}}$, IORQ or $\overline{\text{MREQ}}$ Address Stable From RD or $\overline{\text{WR}}$ During Float		110 90 111 121 131 141	ns ns ns ns ns ns	$C_L = 50\text{pF}$
D_{0-7}	$t_D(\text{D})$ $t_F(\text{D})$ $t_{SD}(\text{D})$ $t_{SD}(\text{D})$ t_{dcm} t_{dci} t_{cdf}	Data Output Delay Delay to Float During Write Cycle Data Setup Time to Rising Edge of Clock During M1 Cycle Data Setup Time to Falling Edge of Clock During M2 to M5 Data Stable Prior to $\overline{\text{WR}}$ (Memory Cycle) Data Stable Prior to $\overline{\text{WR}}$ (I/O Cycle) Data Stable From $\overline{\text{WR}}$		150 90 35 50 131 161 171	ns ns ns ns ns ns ns	$C_L = 50\text{pF}$
	t_H	Any Hold Time for Setup Time		0	ns	
$\overline{\text{MREQ}}$	$t_{DL}(\overline{\text{MREQ}})$ $t_{DH}(\overline{\text{MREQ}})$ $t_{DHL}(\overline{\text{MREQ}})$ $t_w(\overline{\text{MREQ}})$ $t_w(\overline{\text{MREQ}})$	$\overline{\text{MREQ}}$ Delay From Falling Edge of Clock, $\overline{\text{MREQ}}$ Low $\overline{\text{MREQ}}$ Delay From Rising Edge of Clock, $\overline{\text{MREQ}}$ High $\overline{\text{MREQ}}$ Delay From Falling Edge of Clock, $\overline{\text{MREQ}}$ High Pulse Width, $\overline{\text{MREQ}}$ Low Pulse Width, $\overline{\text{MREQ}}$ High		85 85 85 181 191	ns ns ns ns ns	$C_L = 50\text{pF}$
IORQ	$t_{DL}(\text{IORQ})$ $t_{DH}(\text{IORQ})$ $t_{DHL}(\text{IORQ})$ $t_{DHL}(\text{IORQ})$	IORQ Delay From Rising Edge of Clock, IORQ Low IORQ Delay From Falling Edge of Clock, IORQ Low IORQ Delay From Rising Edge of Clock, IORQ High IORQ Delay From Falling Edge of Clock, IORQ High		75 85 85 85	ns ns ns ns	$C_L = 50\text{pF}$
RD	$t_{DL}(\text{RD})$ $t_{DH}(\text{RD})$ $t_{DHL}(\text{RD})$ $t_{DHL}(\text{RD})$	RD Delay From Rising Edge of Clock, RD Low RD Delay From Falling Edge of Clock, RD Low RD Delay From Rising Edge of Clock, RD High RD Delay From Falling Edge of Clock, RD High		85 95 85 85	ns ns ns ns	$C_L = 50\text{pF}$
$\overline{\text{WR}}$	$t_{DL}(\overline{\text{WR}})$ $t_{DH}(\overline{\text{WR}})$ $t_{DHL}(\overline{\text{WR}})$ $t_w(\overline{\text{WR}})$	$\overline{\text{WR}}$ Delay From Rising Edge of Clock, $\overline{\text{WR}}$ Low $\overline{\text{WR}}$ Delay From Falling Edge of Clock, $\overline{\text{WR}}$ Low $\overline{\text{WR}}$ Delay From Rising Edge of Clock, $\overline{\text{WR}}$ High Pulse Width, $\overline{\text{WR}}$ Low		65 80 80 1101	ns ns ns ns	$C_L = 50\text{pF}$
M1	$t_{DL}(\text{M1})$ $t_{DH}(\text{M1})$	M1 Delay From Rising Edge of Clock, M1 Low M1 Delay From Rising Edge of Clock, M1 High		100 100	ns ns	$C_L = 50\text{pF}$
RFSH	$t_{DL}(\text{RFSH})$ $t_{DH}(\text{RFSH})$	RFSH Delay From Rising Edge of Clock, RFSH Low RFSH Delay From Rising Edge of Clock, RFSH High		130 120	ns ns	$C_L = 50\text{pF}$
WAIT	$t_s(\text{WAIT})$	WAIT Setup Time to Falling Edge of Clock	70		ns	
HALT	$t_D(\text{HALT})$	HALT Delay Time From Falling Edge of Clock		300	ns	$C_L = 50\text{pF}$
INT	$t_s(\text{INT})$	INT Setup Time to Rising Edge of Clock	80		ns	
NMI	$t_w(\text{NMI})$	Pulse Width, NMI Low	80		ns	
BUSRQ	$t_s(\text{BUSRQ})$	BUSRQ Setup Time to Rising Edge of Clock	50		ns	
BUSAK	$t_{DL}(\text{BUSAK})$ $t_{DH}(\text{BUSAK})$	BUSAK Delay From Rising Edge of Clock, BUSAK Low BUSAK Delay From Falling Edge of Clock, BUSAK High		100 100	ns ns	$C_L = 50\text{pF}$
RESET	$t_s(\text{RESET})$	RESET Setup Time to Rising Edge of Clock	60		ns	
	$t_F(\text{C})$	Delay to Float ($\overline{\text{MREQ}}$, IORQ , RD and $\overline{\text{WR}}$)		80	ns	
	t_{su}	M1 Stable Prior to IORQ (Interrupt Ack)	1111		ns	

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

$$[11] t_{acm} = t_{w(\phi H)} + t_f - 65$$

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

$$[3] t_{ca} = t_{w(\phi L)} + t_f - 50$$

$$[4] t_{cdf} = t_{w(\phi L)} + t_f - 45$$

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

$$[6] t_{dci} = t_{w(\phi L)} + t_f - 170$$

$$[7] t_{cdf} = t_{w(\phi L)} + t_f - 70$$

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

$$[9] t_w(\overline{\text{MREQ}}) = t_{w(\phi H)} + t_f - 20$$

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

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

NOTES

- Data should be enabled onto the CPU data bus when RD is active. During interrupt acknowledge data should be enabled when M1 and IORQ are both active.
- All control signals are internally synchronized, so they may be totally asynchronous with respect to the clock.
- The RESET signal must be active for a maximum of 3 clock cycles.
- Output Delay vs. Loaded Capacitance
 $T_A = 70^\circ\text{C}$, $V_{CC} = +5V \pm 5\%$
Add 10ns delay for each 50pF increase in load up to maximum of 200pF for data bus and 100pF for address & control lines.
- Although static by design, testing parameters $t_{w(\phi H)}$ of 200 μsec maximum

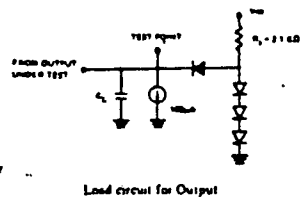


Table 4.3. Z80 Instruction Set

EXCHANGE GROUP AND BLOCK TRANSFER AND SEARCH GROUP

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210				
EX DE, HL	DE $\leftrightarrow$ HL	•	•	•	•	•	•	11	101	011	1	1	4	Register bank and auxiliary register bank exchange
EX AF, AF'	AF $\leftrightarrow$ AF'	•	•	•	•	•	•	00	001	000	1	1	4	
EXX	(BC) $\leftrightarrow$ (DE) (DE) $\leftrightarrow$ (HL)	•	•	•	•	•	•	11	011	001	1	1	4	
EX (SP), HL	H $\leftarrow$ (SP+1) 1 $\leftarrow$ (SP)	•	•	•	•	•	•	11	100	011	1	5	19	
EX (SP), IX	IX _H $\leftarrow$ (SP+1)	•	•	•	•	•	•	11	011	101	2	6	23	
	IX _L $\leftarrow$ (SP)	•	•	•	•	•	•	11	100	011	2	6	23	
FX (SP), IY	IY _H $\leftarrow$ (SP+1)	•	•	•	•	•	•	11	111	101	2	6	23	
	IY _L $\leftarrow$ (SP)	•	•	•	•	•	•	11	100	011	2	6	23	
LDI	(DE) $\leftarrow$ (HL)	•	•	1	•	0	0	11	101	101	2	4	16	Load (HL) into (DE), increment the pointers and decrement the byte counter (BC)
	DE $\leftarrow$ DE+1	•	•	•	•	•	•	10	100	000	2	4	16	
	HL $\leftarrow$ HL+1	•	•	•	•	•	•	•	•	•	•	•	•	
	BC $\leftarrow$ BC-1	•	•	•	•	•	•	•	•	•	•	•	•	
LDIR	(DE) $\leftarrow$ (HL)	•	•	0	•	0	0	11	101	101	2	5	21	If BC $\neq$ 0
	DE $\leftarrow$ DE+1	•	•	•	•	•	•	10	110	000	2	4	16	
	HL $\leftarrow$ HL+1	•	•	•	•	•	•	•	•	•	•	•	•	
	BC $\leftarrow$ BC-1	•	•	•	•	•	•	•	•	•	•	•	•	
LDI	(DE) $\leftarrow$ (HL)	•	•	1	•	0	0	11	101	101	2	4	16	If BC $\neq$ 0
	DE $\leftarrow$ DE-1	•	•	•	•	•	•	10	101	000	2	4	16	
	HL $\leftarrow$ HL-1	•	•	•	•	•	•	•	•	•	•	•	•	
	BC $\leftarrow$ BC-1	•	•	•	•	•	•	•	•	•	•	•	•	
LDOR	(DE) $\leftarrow$ (HL)	•	•	0	•	0	0	11	101	101	2	5	21	If BC $\neq$ 0
	DE $\leftarrow$ DE-1	•	•	•	•	•	•	10	111	000	2	4	16	
	HL $\leftarrow$ HL-1	•	•	•	•	•	•	•	•	•	•	•	•	
	BC $\leftarrow$ BC-1	•	•	•	•	•	•	•	•	•	•	•	•	
CPI	A $\leftarrow$ (HL)	•	1	1	1	1	1	11	101	101	2	4	16	If BC $\neq$ 0 and A $\neq$ (HL) If BC = 0 or A = (HL)
	HL $\leftarrow$ HL+1	•	•	•	•	•	•	10	100	001	2	4	16	
	BC $\leftarrow$ BC-1	•	•	•	•	•	•	•	•	•	•	•	•	
	Repeat until BC = 0	•	•	•	•	•	•	•	•	•	•	•	•	
CMR	A $\leftarrow$ (HL)	•	1	1	1	1	1	11	101	101	2	5	21	If BC $\neq$ 0 and A $\neq$ (HL) If BC = 0 or A = (HL)
	HL $\leftarrow$ HL+1	•	•	•	•	•	•	10	110	001	2	4	16	
	BC $\leftarrow$ BC-1	•	•	•	•	•	•	•	•	•	•	•	•	
	Repeat until A = (HL) or BC = 0	•	•	•	•	•	•	•	•	•	•	•	•	

Table 4.3. Continued

Mnemonic	Symbolic Operation	Flags						Op-Code				No. of Bytes	N. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210					
'CPD	A ← (HL)	•	1	1	1	1	1	11	101	101	2	4	16		
	HL ← HL-1							10	101	001					
	BC ← BC-1														
'CPDR	A ← (HL)	•	1	1	1	1	1	11	101	101	2	5	21	If BC ≠ 0 and A ≠ (HL)	
	HL ← HL-1							10	111	001	2	4	16	If BC = 0 or A = (HL)	
	BC ← BC-1														
	Repeat until A = (HL) or BC = 0														

Notes: ① P/V flag is 0 if the result of BC-1 = 0, otherwise P/V = 1

② Z flag is 1 if A = (HL), otherwise Z = 0

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,

1 = flag is affected according to the result of the operation.

8-BIT ARITHMETIC AND LOGICAL GROUP

Mnemonic	Symbolic Operation	Flags						Op-Code				No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210					
ADD A, r	A ← A + r	1	1	V	1	0	1	10	000	r	1	1	4	r	Reg.
ADD A, n	A ← A + n	1	1	V	1	0	1	11	000	110	2	2	7	000 001 010 011	B C D E
ADD A, (HL)	A ← A + (HL)	1	1	V	1	0	1	10	000	110	1	2	7	100 101 111	H L A
ADD A, (IX+d)	A ← A + (IX+d)	1	1	V	1	0	1	11	011	101	3	5	19	100 101 111	H L A
ADD A, (IY+d)	A ← A + (IY+d)	1	1	V	1	0	1	11	111	101	3	5	19		
ADC A, s	A ← A + s + CY	1	1	V	1	0	1		001						s is any of r, n, (HL), (IX+d), (IY+d) as shown for ADD instruction
SUB s	A ← A - s	1	1	V	1	1	1		010						
SBC A, s	A ← A - s - CY	1	1	V	1	1	1		011						
AND s	A ← A ∧ s	0	1	P	1	0	1		100						
OR s	A ← A ∨ s	0	1	P	1	0	0		110						The indicated bits replace the 000 in the ADD set above.
XOR s	A ← A ⊕ s	0	1	P	1	0	0		101						
CP s	A - s	1	1	V	1	1	1		111						
INC r	r ← r + 1	•	1	V	1	0	1	00	r	100	1	1	4		
INC (HL)	(HL) ← (HL) + 1	•	1	V	1	0	1	00	110	100	1	3	11		
INC (IX+d)	(IX+d) ← (IX+d) + 1	•	1	V	1	0	1	11	011	101	3	6	23		
								00	110	100					
								-	d	-					

Table 4.3. Continued

Table 4.3. Continued															
Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments	
		C	Z	P/V	S	N	H	76	543	210					
INC (IY+d)	(IY+d) ← (IY+d) + 1	•	•	V	•	•	•	11	111	101	3	6	23		
								00	110	100					
								-	d	-					
DEC m	m ← m - 1	•	•	V	•	•	•			101					
														m is any of r, (HL), (IX+d), (IY+d) as shown for INC. Same format and states as INC. Replace 100 with 101 in OP code.	

Notes: The V symbol in the P/V flag column indicates that the P/V flag contains the overflow of the result of the operation. Similarly the P symbol indicates parity. V = 1 means overflow, V = 0 means not overflow. P = 1 means parity of the result is even, P = 0 means parity of the result is odd.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown.
 • = flag is affected according to the result of the operation.

8-BIT LOAD GROUP

8-BIT LOAD GROUP															
Mnemonic	Symbolic Operation	Flags						OP-Code			No. of Bytes	No. of M Cycles	No. of T Cycles	Comments	
		C	Z	P/V	S	N	H	76	543	210				r, r'	Reg.
LD r, r'	r ← r'	•	•	•	•	•	•	01	r	r'	1	1	4	r, r'	Reg.
LD r, n	r ← n	•	•	•	•	•	•	00	r	110	2	2	7	000	B
								—	n	—				001	C
LD r, (HL)	r ← (HL)	•	•	•	•	•	•	01	r	110	1	2	7	010	D
LD r, (IX+d)	r ← (IX+d)	•	•	•	•	•	•	11	011	101	3	5	19	011	E
								01	r	110				100	H
								—	d	—				101	L
LD r, (IY+d)	r ← (IY+d)	•	•	•	•	•	•	11	111	101	3	5	19	111	A
								01	r	110					
								—	d	—					
LD (HL), r	(HL) ← r	•	•	•	•	•	•	01	110	r	1	2	7		
LD (IX+d), r	(IX+d) ← r	•	•	•	•	•	•	11	011	101	3	5	19		
								01	110	r					
								—	d	—					
LD (IY+d), r	(IY+d) ← r	•	•	•	•	•	•	11	111	101	3	5	19		
								01	110	r					
								—	d	—					
LD (HL), n	(HL) ← n	•	•	•	•	•	•	00	110	110	2	3	10		
								—	n	—					
LD (IX+d), n	(IX+d) ← n	•	•	•	•	•	•	11	011	101	4	5	19		
								00	110	110					
								—	d	—					
								—	n	—					
LD (IY+d), n	(IY+d) ← n	•	•	•	•	•	•	11	111	101	4	5	19		
								00	110	110					
								—	d	—					
								—	n	—					

Table 4.3. Continued

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T Cycles	Comments
		C	Z	P/V	S	N	H	76	543	210				
LD A, (BC)	A ← (BC)	•	•	•	•	•	•	00	001	010	1	2	7	
LD A, (DE)	A ← (DE)	•	•	•	•	•	•	00	011	010	1	2	7	
LD A, (nn)	A ← (nn)	•	•	•	•	•	•	00	111	010	3	4	13	
								—	n	—				
								—	n	—				
LD (BC), A	(BC) ← A	•	•	•	•	•	•	00	000	010	1	2	7	
LD (DE), A	(DE) ← A	•	•	•	•	•	•	00	010	010	1	2	7	
LD (nn), A	(nn) ← A	•	•	•	•	•	•	00	110	010	3	4	13	
								—	n	—				
								—	n	—				
LD A, I	A ← I	•	‡	IFF	‡	0	0	11	101	101	2	2	9	
								01	010	111				
LD A, R	A ← R	•	‡	IFF	‡	0	0	11	101	101	2	2	9	
								01	011	111				
LD I, A	I ← A	•	•	•	•	•	•	11	101	101	2	2	9	
								01	000	111				
LD R, A	R ← A	•	•	•	•	•	•	11	101	101	2	2	9	
								01	001	111				

Notes: r, r' means any of the registers A, B, C, D, E, H, L

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

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,

‡ = flag is affected according to the result of the operation.

16-BIT LOAD GROUP

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T Cycles	Comments
		C	Z	P/V	S	N	H	76	543	210				
LD dd, nn	dd ← nn	•	•	•	•	•	•	00	dd0	001	3	3	10	dd Pair
								—	n	—				00 BC
								—	n	—				01 DE
LD IX, nn	IX ← nn	•	•	•	•	•	•	11	011	101	4	4	14	10 HL
								00	100	001				11 SP
								—	n	—				
								—	n	—				
LD IY, nn	IY ← nn	•	•	•	•	•	•	11	111	101	4	4	14	
								00	100	001				
								—	n	—				
								—	n	—				
LD HL, (nn)	H ← (nn+1) L ← (nn)	•	•	•	•	•	•	00	101	010	3	3	16	
								—	n	—				
								—	n	—				
LD dd, (nn)	dd _H ← (nn+1) dd _L ← (nn)	•	•	•	•	•	•	11	101	101	4	6	20	
								01	dd1	011				
								—	n	—				
								—	n	—				
LD IX, (nn)	IX _H ← (nn+1) IX _L ← (nn)	•	•	•	•	•	•	11	011	101	4	6	20	
								00	101	010				
								—	n	—				
								—	n	—				

Table 4.3. Continued

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	V	S	N	H	76	543	210				
LD Y_L , (nn)	$Y_H - (nn+1)$	•	•	•	•	•	•	11	111	101	4	6	20	
	$Y_L - (nn)$							00	101	010				
								—	—	—				
								—	—	—				
LD (nn), HL	$(nn+1) - H$	•	•	•	•	•	•	00	100	010	3	5	16	
	$(nn) - L$							—	—	—				
								—	—	—				
								—	—	—				
LD (nn), dd	$(nn+1) - dd_H$	•	•	•	•	•	•	11	101	101	4	6	20	
	$(nn) - dd_L$							01	440	011				
								—	—	—				
								—	—	—				
LD (nn), IX	$(nn+1) - IX_H$	•	•	•	•	•	•	11	011	101	4	6	20	
	$(nn) - IX_L$							00	100	010				
								—	—	—				
								—	—	—				
LD (nn), IY	$(nn+1) - IY_H$	•	•	•	•	•	•	11	111	101	4	6	20	
	$(nn) - IY_L$							00	100	010				
								—	—	—				
								—	—	—				
LD SP, HL	SP - HL	•	•	•	•	•	•	11	111	001	1	1	6	
LD SP, IX	SP - IX	•	•	•	•	•	•	11	011	101	2	2	10	
								11	111	001				
LD SP, IY	SP - IY	•	•	•	•	•	•	11	111	101	2	2	10	
								11	111	001				
PUSH qq	$(SP-2) - qq_L$	•	•	•	•	•	•	11	qq0	101	1	3	11	qq Pair
	$(SP-1) - qq_H$													00 BC
PUSH IX	$(SP-2) - IX_L$	•	•	•	•	•	•	11	011	101	2	4	15	01 DE
	$(SP-1) - IX_H$							11	100	101				10 HL
PUSH IY	$(SP-2) - IY_L$	•	•	•	•	•	•	11	111	101	2	4	15	11 AF
	$(SP-1) - IY_H$							11	100	101				
POP qq	$qq_H - (SP+1)$	•	•	•	•	•	•	11	qq0	001	1	3	10	
	$qq_L - (SP)$													
POP IX	$IX_H - (SP+1)$	•	•	•	•	•	•	11	011	101	2	4	14	
	$IX_L - (SP)$							11	100	001				
POP IY	$IY_H - (SP+1)$	•	•	•	•	•	•	11	111	101	2	4	14	
	$IY_L - (SP)$							11	100	001				

Notes: dd is any of the register pairs BC, DE, HL, SP
 qq is any of the register pairs AF, BC, DE, HL
 (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

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
 ‡ flag is affected according to the result of the operation.

GENERAL PURPOSE ARITHMETIC AND CPU CONTROL GROUPS

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210				
DAA	Converts acc. content into packed BCD following add or subtract with packed BCD operands	‡	‡	P	‡	•	‡	00	100	111	1	1	4	Decimal adjust accumulator

Table 4.3. Continued

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210				
CPL	$A \leftarrow \bar{A}$	•	•	•	•	1	1	00	101	111	1	1	4	Complement accumulator (one's complement)
NEG	$A \leftarrow 0 - A$	‡	‡	V	‡	1	‡	11	101	101	2	2	8	Negate acc. (two's complement)
CCF	$CY \leftarrow \bar{CY}$	‡	•	•	•	0	X	00	111	111	1	1	4	Complement carry flag
SCF	$CY \leftarrow 1$	1	•	•	•	0	0	00	110	111	1	1	4	Set carry flag
NOP	No operation	•	•	•	•	•	•	00	000	000	1	1	4	
HALT	CPU halted	•	•	•	•	•	•	01	110	110	1	1	4	
DI	IFF $\leftarrow 0$	•	•	•	•	•	•	11	110	011	1	1	4	
EI	IFF $\leftarrow 1$	•	•	•	•	•	•	11	111	011	1	1	4	
IM 0	Set interrupt mode 0	•	•	•	•	•	•	11	101	101	2	2	8	
IM 1	Set interrupt mode 1	•	•	•	•	•	•	11	101	101	2	2	8	
IM 2	Set interrupt mode 2	•	•	•	•	•	•	11	101	101	2	2	8	

Notes: IFF indicates the interrupt enable flip-flop.
CY indicates the carry flip-flop.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown,
‡ = flag is affected according to the result of the operation.

16-BIT ARITHMETIC GROUP

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210				
ADD HL, ss	$HL \leftarrow HL + ss$	‡	•	•	•	0	X	00	ss1	001	1	3	11	ss Reg. 00 BC 01 DE 10 HL 11 SP
ADC HL, ss	$HL \leftarrow HL + ss + CY$	‡	•	V	‡	0	X	11	101	101	2	4	15	
SBC HL, ss	$HL \leftarrow HL - ss - CY$	‡	•	V	‡	1	X	11	101	101	2	4	15	
ADD IX, pp	$IX \leftarrow IX + pp$	‡	•	•	•	0	X	11	011	101	2	4	15	pp Reg. 00 BC 01 DE 10 IX 11 SP
ADD IY, rr	$IY \leftarrow IY + rr$	‡	•	•	•	0	X	11	111	101	2	4	15	rr Reg. 00 BC 01 DE 10 IY 11 SP

Table 4.3. Continued

Mnemonic	Symbolic Operation	Flags						Op-Code	No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H					
INC ss	ss ← ss + 1	.	.	.	.	.	.	00 ss0 011	1	1	6	
INC IX	IX ← IX + 1	.	.	.	.	.	.	11 011 101	2	2	10	
INC IY	IY ← IY + 1	.	.	.	.	.	.	00 100 011				
		.	.	.	.	.	.	11 111 101	2	2	10	
DEC ss	ss ← ss - 1	.	.	.	.	.	.	00 100 011				
		.	.	.	.	.	.	00 ss1 011	1	1	6	
DEC IX	IX ← IX - 1	.	.	.	.	.	.	11 011 101	2	2	10	
		.	.	.	.	.	.	00 101 011				
DEC IY	IY ← IY - 1	.	.	.	.	.	.	11 111 101	2	2	10	
		.	.	.	.	.	.	00 101 011				

Notes: ss is any of the register pairs BC, DE, HL, SP
 pp is any of the register pairs BC, DE, IX, SP
 rr is any of the register pairs BC, DE, IY, SP.

Flag Notation: . = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown.
 ‡ = flag is affected according to the result of the operation.

ROTATE AND SHIFT GROUP

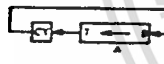
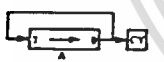
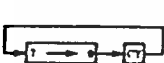
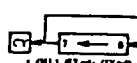
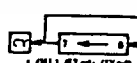
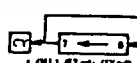
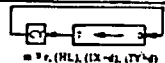
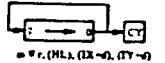
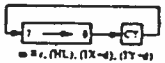
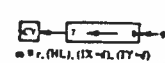
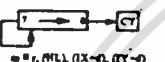
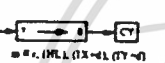
Mnemonic	Symbolic Operation	Flags						Op-Code	No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H					
RLCA		‡	.	.	.	0	0	00 000 111	1	1	4	Rotate left circular accumulator
RLA		‡	.	.	.	0	0	00 010 111	1	1	4	Rotate left accumulator
RRCA		‡	.	.	.	0	0	00 001 111	1	1	4	Rotate right circular accumulator
RRA		‡	.	.	.	0	0	00 011 111	1	1	4	Rotate right accumulator
RLC r		‡	‡	P	‡	0	0	11 001 011	2	2	8	Rotate left circular register r
RLC (HL)		‡	‡	P	‡	0	0	00 000 r				
		‡	‡	P	‡	0	0	11 001 011	2	4	15	r
		‡	‡	P	‡	0	0	00 000 110				Reg.
		‡	‡	P	‡	0	0	11 011 101	4	6	23	000 B
RLC (IX+d)		‡	‡	P	‡	0	0	11 001 011				001 C
		‡	‡	P	‡	0	0	11 011 101	4	6	23	010 D
		‡	‡	P	‡	0	0	11 001 011				011 E
		‡	‡	P	‡	0	0	← d →				100 H
		‡	‡	P	‡	0	0	00 000 110				101 L
RLC (IY+d)		‡	‡	P	‡	0	0	11 111 101	4	6	23	111 A
		‡	‡	P	‡	0	0	12 001 011				
		‡	‡	P	‡	0	0	← d →				
		‡	‡	P	‡	0	0	00 000 110				
		‡	‡	P	‡	0	0	12 001 011				

Table 4.3. Continued

Mnemonic	Symbolic Operation	Flags						Op-Code	No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H					
RL m	 $\Rightarrow r, (HL), (IX+d), (IY+d)$	*	*	P	*	0	0	010				Instruction format and states are as shown for RLC,m. To form new OP-code replace 000 of RLC,m with shown code
RRC m	 $\Rightarrow r, (HL), (IX+d), (IY+d)$	*	*	P	*	0	0	001				
RR m	 $\Rightarrow r, (HL), (IX+d), (IY+d)$	*	*	P	*	0	0	011				
SLA m	 $\Rightarrow r, (HL), (IX+d), (IY+d)$	*	*	P	*	0	0	100				
SRA m	 $\Rightarrow r, (HL), (IX+d), (IY+d)$	*	*	P	*	0	0	101				
SRL m	 $\Rightarrow r, (HL), (IX+d), (IY+d)$	*	*	P	*	0	0	111				
RLD	 $\Rightarrow r, (HL), (IX+d), (IY+d)$	*	*	P	*	0	0	11 101 101 01 101 111	2	5	18	Rotate digit left and right between the accumulator and location (HL). The content of the upper half of the accumulator is unaffected
RRD	 $\Rightarrow r, (HL), (IX+d), (IY+d)$	*	*	P	*	0	0	11 101 101 01 100 111	2	5	18	

Flag Notation: * = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, * = flag is affected according to the result of the operation.

BIT SET, RESET AND TEST GROUP

Mnemonic	Symbolic Operation	Flags						Op-Code	No. of Bytes	No. of M Cycles	No. of T States	Comments	
		C	Z	P/V	S	N	H						
BIT b, r	$Z \leftarrow \overline{T_b}$	*	*	X	X	0	1	11 001 011 01 b r	2	2	8	r	Reg.
BIT b, (HL)	$Z \leftarrow \overline{(HL)_b}$	*	*	X	X	0	1	11 001 011	2	3	12	000	B
								01 b r				001	C
								01 b 11Q				010	D
								11 001 011				011	E
BIT b, (EX+d)	$Z \leftarrow \overline{(EX+d)_b}$	*	*	X	X	0	1	11 011 101	4	5	20	100	H
								11 001 011				101	L
								- d -				111	A
								01 b 110					
BIT b, (IY+d)	$Z \leftarrow \overline{(IY+d)_b}$	*	*	X	X	0	1	11 111 101	4	5	20	b	Bit Tested
								11 001 011				000	0
								- d -				001	1
								01 b 110				010	2
												011	3
												100	4
												101	5
												110	6
												111	7

Table 4.3. Continued

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	$\overline{P}$ V	S	N	H	76	543	210				
SET b, r	$r_b \leftarrow 1$	•	•	•	•	•	•	11	001	011	2	2	8	
SET b, (HL)	$(HL)_b \leftarrow 1$	•	•	•	•	•	•	11	b	r	2	4	15	
SET b, (IX+d)	$(IX+d)_b \leftarrow 1$	•	•	•	•	•	•	11	b	110	4	6	23	
SET b, (IY+d)	$(IY+d)_b \leftarrow 1$	•	•	•	•	•	•	11	001	011	4	6	23	
								-	d	-				
								11	b	110				
								11	111	101	4	6	23	
								11	001	011				
								-	d	-				
								11	b	110				
RES b, m	$s_b \leftarrow 0$ $m \equiv r, (HL),$ $(IX+d),$ $(IY+d)$							10						To form new OP-code replace 11 of SET b,m with 10. Flags and time states for SET instruction

Notes: The notation s_b indicates bit b (0 to 7) or location s.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, $\dagger$ = flag is affected according to the result of the operation.

CALL AND RETURN GROUP

Mnemonic	Symbolic Operation	Flags						Op-Code				No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	$\overline{P}$ V	S	N	H	76	543	210					
CALL nn	(SP-1)→PC _H (SP-2)→PC _L PC←nn	•	•	•	•	•	•	11	001	101	3	5	17		
								←	n	←					
								←	n	←					
CALL cc, nn	If condition cc is false continue, otherwise same as CALL nn	•	•	•	•	•	•	11	cc	100	3	3	10	If cc is false	
								←	n	←					
								←	n	←	3	5	17	If cc is true	
RET	PC _L ←(SP) PC _H ←(SP+1)	•	•	•	•	•	•	11	001	001	1	3	10		

Table 4.3. Continued

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210				
RET cc	If condition cc is false continue, otherwise same as RET	•	•	•	•	•	•	11	cc	000	1	1	5	If cc is false
											1	3	11	If cc is true
														cc Condition
														000 NZ non zero
														001 Z zero
														010 NC non carry
														011 C carry
														100 PO parity odd
														101 PE parity even
														110 P sign positive
														111 M sign negative
RETl	Return from Interrupt	•	•	•	•	•	•	11	101	101	2	4	14	
RETn	Return from non maskable Interrupt	•	•	•	•	•	•	01	001	101	2	4	14	
RST p	(SP-1)-PC _H (SP-2)-PC _L PC _H ← 0 PC _L ← P	•	•	•	•	•	•	11	t	111	1	3	11	
														t P
														000 00H
														001 08H
														010 10H
														011 18H
														100 20H
														101 28H
														110 30H
														111 38H

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown
: = flag is affected according to the result of the operation.

JUMP GROUP

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210				
JP nn	PC ← nn	•	•	•	•	•	•	11	000	011	3	3	10	
								-	n	-				
								-	n	-				
JP cc, nn	If condition cc is true PC ← nn, otherwise continue	•	•	•	•	•	•	11	cc	010	3	3	10	cc Condition
								-	n	-				000 NZ non zero
								-	n	-				001 Z zero
								-	n	-				010 NC non carry
								-	n	-				011 C carry
														100 PO parity odd
														101 PE parity even
														110 P sign positive
														111 M sign negative
JR e	PC ← PC + e	•	•	•	•	•	•	00	011	000	2	3	12	
								-	e-2	-				
JR C, e	If C = 0, continue	•	•	•	•	•	•	00	111	000	2	2	7	If condition not met
								-	e-2	-				
	If C = 1, PC ← PC + e										2	3	12	If condition is met

Table 4.3. Continued

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210				
JR NC, e	If C = 1, continue	•	•	•	•	•	•	00	110	000	2	2	7	If condition not met
	If C = 0, PC ← PC + e							—	e-2	—	2	3	12	If condition is met
JR Z, e	If Z = 0 continue	•	•	•	•	•	•	00	101	000	2	2	7	If condition not met
	If Z = 1, PC ← PC + e							—	e-2	—	2	3	12	If condition is met
JR NZ, e	If Z = 1, continue	•	•	•	•	•	•	00	100	000	2	2	7	If condition not met
	If Z = 0, PC ← PC + e							—	e-2	—	2	3	12	If condition met
JP (HL)	PC ← HL	•	•	•	•	•	•	11	101	001	1	1	4	
JP (IX)	PC ← IX	•	•	•	•	•	•	11	011	101	2	2	8	
								11	101	001				
JP (IY)	PC ← IY	•	•	•	•	•	•	11	111	101	2	2	8	
								11	101	001				
DJNZ, e	B ← B-1	•	•	•	•	•	•	00	010	000	2	2	8	If B = 0
	If B = 0, continue							—	e-2	—	2	3	13	If B ≠ 0
	If B ≠ 0, PC ← PC + e													

Notes: e represents the extension in the relative addressing mode.

e is a signed two's complement number in the range <-126, 129>

e-2 in the op-code provides an effective address of pc + e as PC is incremented by 2 prior to the addition of e.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, § = flag is affected according to the result of the operation.

INPUT AND OUTPUT GROUP

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210				
IN A, (n)	A ← (n)	•	•	•	•	•	•	11	011	011	2	3	11	n to A ₀ ~ A ₇
IN r, (C)	r ← (C)	•	•	P	•	0	•	11	101	101	2	3	12	Acc to A ₈ ~ A ₁₅
	if r = 110 only the flags will be affected							01	r	000				C to A ₀ ~ A ₇ B to A ₈ ~ A ₁₅

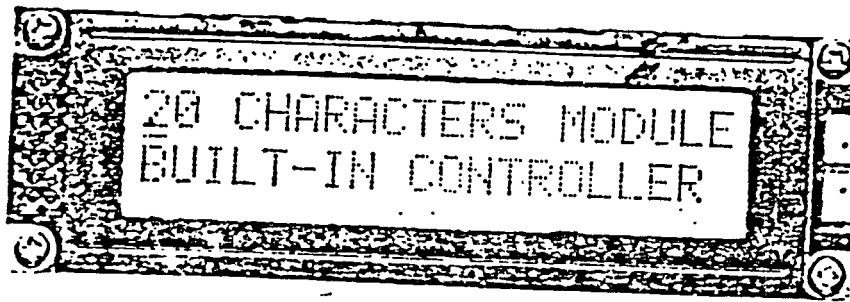
Table 4.3. Continued

Mnemonic	Symbolic Operation	Flags						Op-Code			No. of Bytes	No. of M Cycles	No. of T States	Comments
		C	Z	P/V	S	N	H	76	543	210				
INI	(HL) ← (C)	X	⊡	X	X	1	X	11	101	101	2	4	16	C to A ₀ - A ₇ B to A ₈ - A ₁₅
	B ← B - 1							10	100	010				
INIR	HL ← HL + 1													C to A ₀ - A ₇ B to A ₈ - A ₁₅
	(HL) ← (C)	X	1	X	X	1	X	11	101	101	2	5 (If B ≠ 0)	21	
	B ← B - 1							10	110	010				
	HL ← HL + 1										2	4 (If B = 0)	16	
IND	Repeat until B = 0													
IND	(HL) ← (C)	X	⊡	X	X	1	X	11	101	101	2	4	16	C to A ₀ - A ₇ B to A ₈ - A ₁₅
	B ← B - 1							10	101	010				
INDR	HL ← HL - 1													C to A ₀ - A ₇ B to A ₈ - A ₁₅
	(HL) ← (C)	X	1	X	X	1	X	11	101	101	2	5 (If B ≠ 0)	21	
	B ← B - 1							10	111	010				
	HL ← HL - 1										2	4 (If B = 0)	16	
OUT (n), A	Repeat until B = 0													
OUT (C), r	(n) ← A							11	010	011	2	3	11	n to A ₀ - A ₇ Acc to A ₈ - A ₁₅
	(C) ← r							← n →	11	101	101	2	3	12
OUTI								01	r	001				C to A ₀ - A ₇ B to A ₈ - A ₁₅
OUTI	(C) ← (HL)	X	⊡	X	X	1	X	11	101	101	2	4	16	C to A ₀ - A ₇ B to A ₈ - A ₁₅
	B ← B - 1							10	100	011				
OTIR	HL ← HL + 1													C to A ₀ - A ₇ B to A ₈ - A ₁₅
	(C) ← (HL)	X	1	X	X	1	X	11	101	101	2	5 (If B ≠ 0)	21	
	B ← B - 1							10	110	011				
	HL ← HL + 1										2	4 (If B = 0)	16	
OUTD	Repeat until B = 0													
OUTD	(C) ← (HL)	X	⊡	X	X	1	X	11	101	101	2	4	16	C to A ₀ - A ₇ B to A ₈ - A ₁₅
	B ← B - 1							10	101	011				
OTDR	HL ← HL - 1													C to A ₀ - A ₇ B to A ₈ - A ₁₅
	(C) ← (HL)	X	1	X	X	1	X	11	101	101	2	5 (If B ≠ 0)	21	
	B ← B - 1							10	111	011				
	HL ← HL - 1										2	4 (If B = 0)	16	
OTDR	Repeat until B = 0													

Notes: ① If the result of B - 1 is zero the Z flag is set, otherwise it is reset.

Flag Notation: • = flag not affected, 0 = flag reset, 1 = flag set, X = flag is unknown, ⊡ = flag is affected according to the result of the operation.

Source: Copyright © 1982 Advanced Micro Devices, Inc. Reproduced by permission.



CHARACTERS X 2 LINES MODULE

PHYSICAL DATA

Module size	120.0W x 36.0H x 10.0D mm
View area	88.2W x 20.0H mm
Character construction	5 x 7 dots
Character size	3.4W x 5.15H mm
Character pitch	4.2 mm
Dot size	0.6W x 0.65H mm
Weight	about 28g

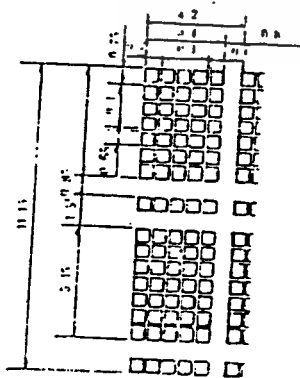
ABSOLUTE MAXIMUM RATINGS

	min	max
Power supply for logic ($V_{DD} - V_{SS}$)	0	7.0 V
Power supply for LCD ($V_{DD} - V_L$)	0	13.5 V
Input voltage (V_{IN})	V_{SS}	V_{DD} V
Operation temperature (T_{OP})	0°C	50°C
Storage temperature (T_{STG})	-20°C	70°C

TRICAL CHARACTERISTICS ($V_{DD} = 5.0 \pm 0.25V$ 25°C)

Input high voltage (V_{IH})	2.2V min.
Input low voltage (V_{IL})	0.6V max.
Output high voltage (V_{OH}) ($I_{OH}=0.205mA$)	2.4V min.
Output low voltage (V_{OL}) ($I_{OL}=1.2mA$)	0.4V max.
Power supply current (I_{DD}) ($V_{DD}=5.0V$)	1.2mA typ. 2.3mA max.
Refresh method	1/16 Duty
Power supply for LCD drive ($V_{DD} - V_L$)	
$T_a = 0^\circ C$	4.6V typ.
$T_a = 25^\circ C$	4.4V typ.
$T_a = 50^\circ C$	3.6V typ.

DISPLAY PATTERN

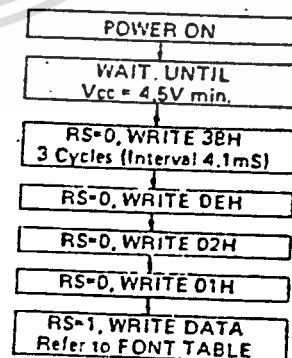


PIN CONNECTIONS

Pin No.	Symbol	Level	Function
1	V_{SS}	—	0 V
2	V_{DD}	—	5 V
3	V_L	—	—
4	RS	H/L	H: Data input L: Instruction data input
5	R/W	H/L	H: Data read L: Data write
6	E	H, H+L	Enable signal
7	D0	H/L	Data bus line*
8	D1	H/L	
9	D2	H/L	
10	D3	H/L	
11	D4	H/L	
12	D5	H/L	
13	D6	H/L	
14	D7	H/L	

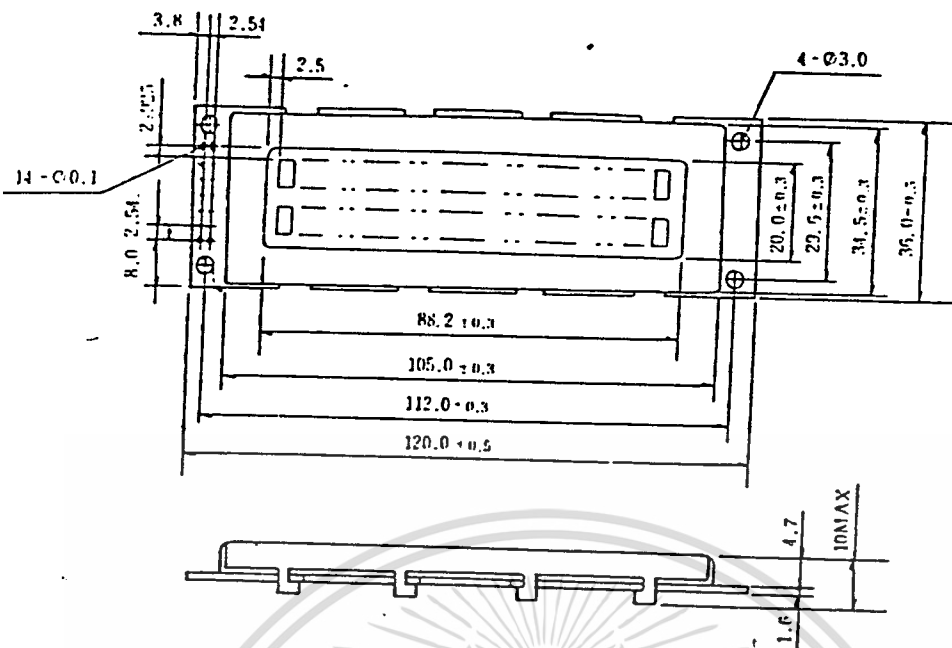
* In case of 4 bits instruction, data is transferred by twice using only 4 buses of D4-D7, and D0-D3 are not used; first operation is higher order 4 bits and second is lower 4 bits of 8 bits, but in case of 8 bits instruction, data is transferred by data bus of D0-D7.

TEST PROCEDURE

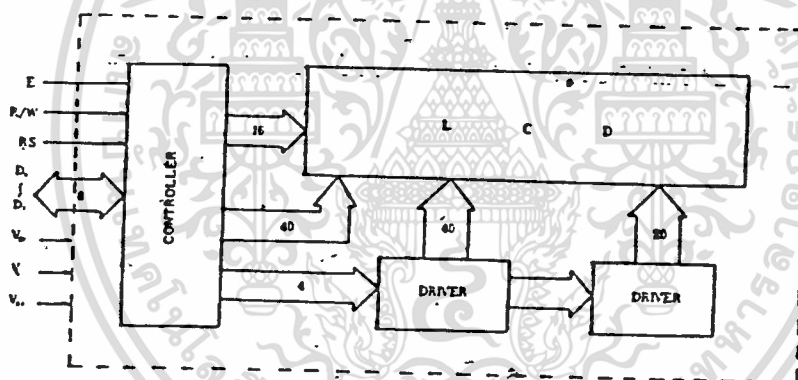


If above instruction is executed by 8 bits, 5 x 7 dots character will be displayed from left side of upper line and cursor moves to right.

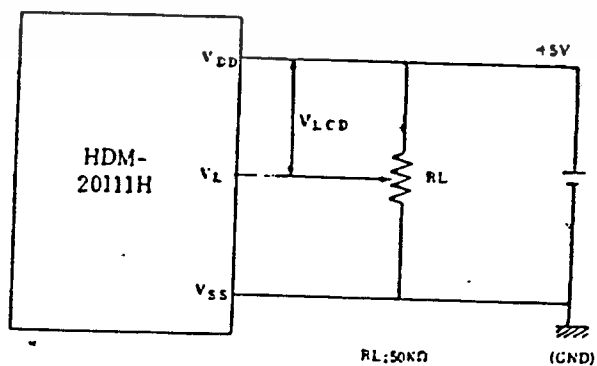
VAL DIMENSIONS



DIAGRAM

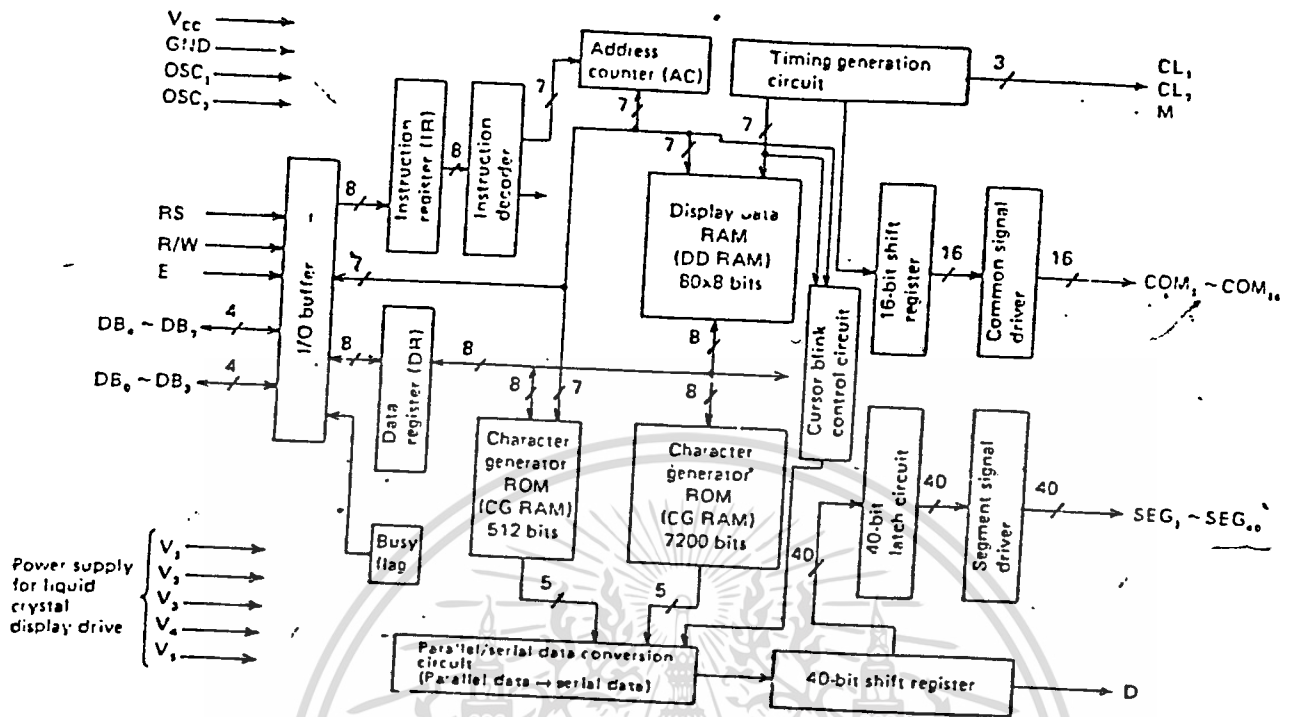


SUPPLY

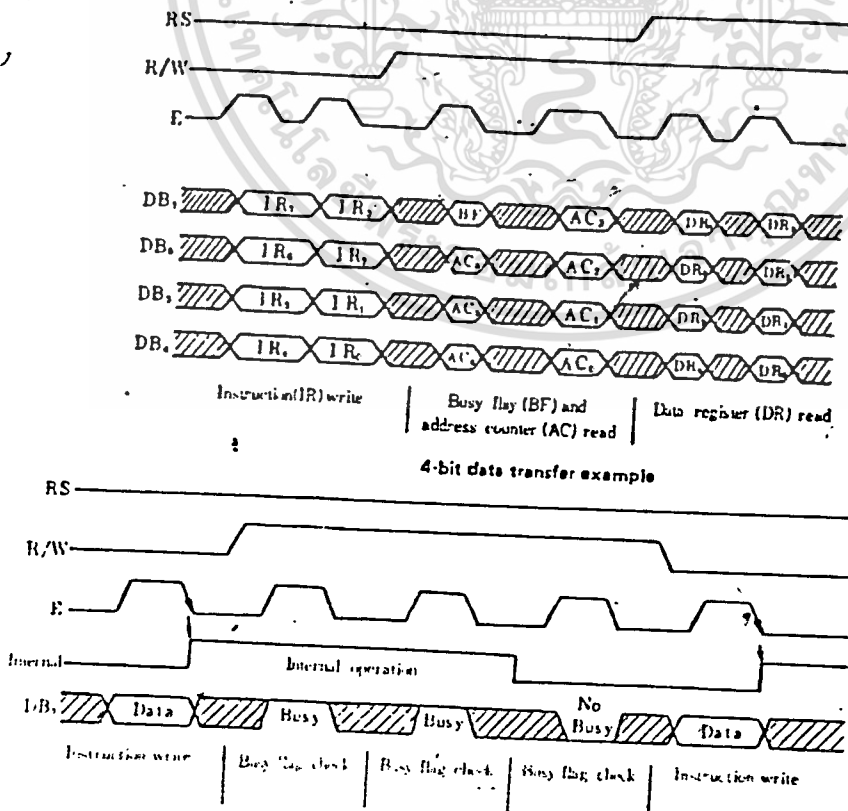


© 1997 NOK

Block diagram of HD44780 interior



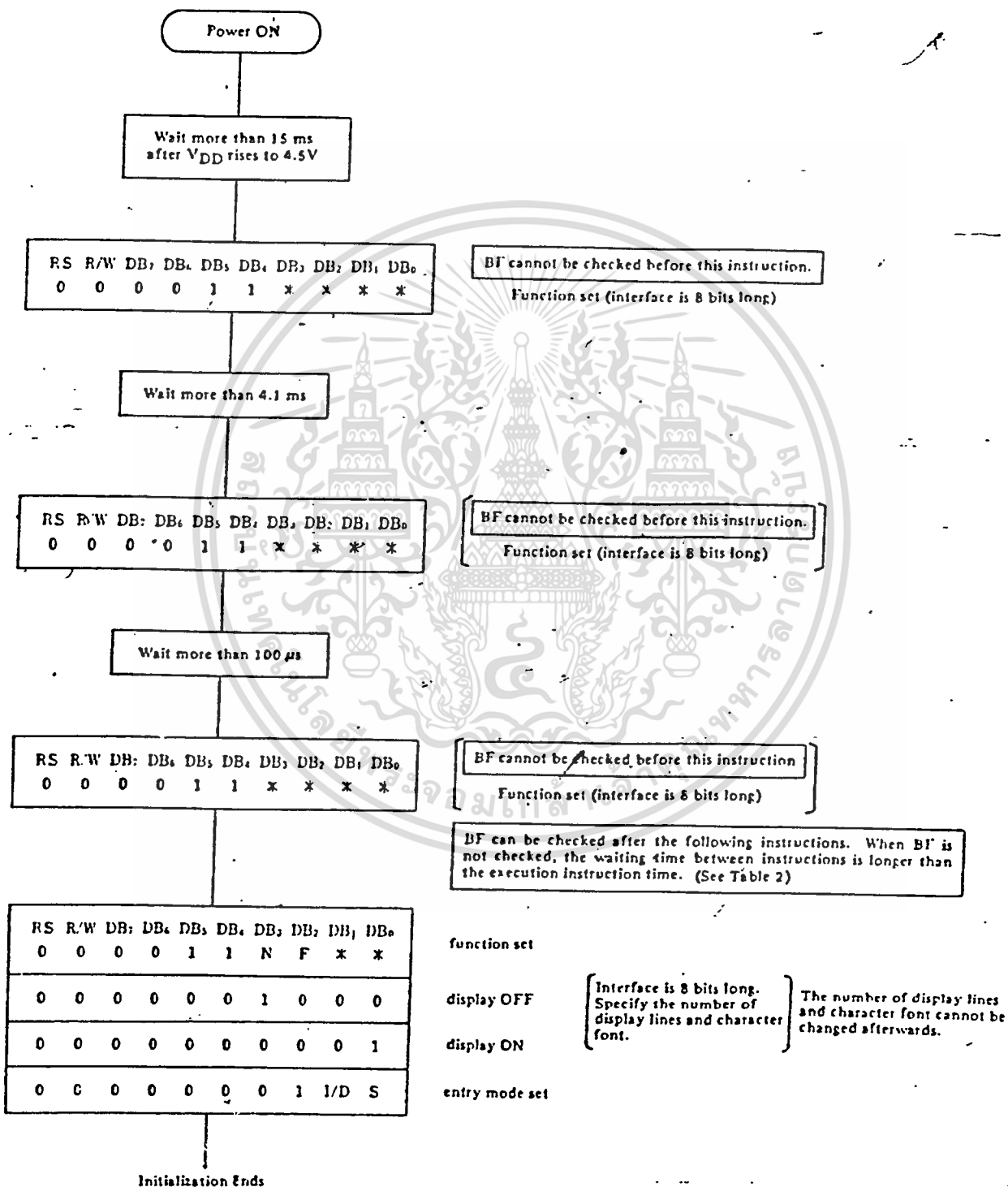
เท่านั้นโดยข้อมูลทั้งหมดที่ส่งให้ HD44780 จะถือเป็นข้อมูล 4 BIT หนึ่ง และข้อมูลทั้งหมดที่ส่งต่อมาก็เป็นข้อมูล 4 BIT ว่าง



Example of connection with LCM being used as a part of memories on the determined address.

4. ENTRY MODE SET $I/D = 1 : +1$ (เพิ่มค่า COUNTER ขึ้น 1)
 $S = 0 : NO SHIFT$

เมื่อการเขียนในบิตร็องทำงานแล้วก็จะต้องรอเพื่อให้หน่วยไมโครเริ่มทำงานถึงตาราง



	Code										Description	Execution time (when fosc is 250 kHz) Note 1	Execution time (when fosc is 160 kHz) Note 2		
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0					
Clear Display	0	0	0	0	0	0	0	0	0	1	Clears all display and returns the cursor to the home position (Address 0).	82 μ s ~ 1.64 ms	120 μ s ~ 4.9 ms		
Return Cursor	0	0	0	0	0	0	0	0	1	.	Returns the cursor to the home position (Address 0). Also returns the display being shifted to the original position. DD RAM contents remain unchanged.	40 μ s ~ 1.6 ms	120 μ s ~ 4.8 ms		
Cursor Move	0	0	0	0	0	0	0	1	I/D	S	Sets the cursor move direction and specifies or not to shift the display. These operations are performed during data write and read.	40 μ s	120 μ s		
Display ON/OFF	0	0	0	0	0	0	1	D	C	B	Sets ON/OFF of all display (D), cursor ON/OFF (C), and blink of cursor position character (B).	40 μ s	120 μ s		
Cursor and Display Shift	0	0	0	0	0	1	S/C	R/L	.	.	Moves the cursor and shifts the display without changing DD RAM contents	40 μ s	120 μ s		
Interface Set	0	0	0	0	1	DL	N	F	.	.	Sets interface data length (DL), number of display lines (L) and character font (F).	40 μ s	120 μ s		
CG RAM Address	0	0	0	1	ACG						Sets the CG RAM address. CG RAM data is sent and received after this setting.	40 μ s	120 μ s		
DD RAM Address	0	0	1	ADD						Sets the DD RAM address. DD RAM data is sent and received after this setting.	40 μ s	120 μ s			
Read Busy & address	0	1	BF	AC						Reads Busy flag (BF) indicating internal operation is being performed and reads address counter contents.	1 μ s	1 μ s			
Write Data CG or RAM	1	0	Write Data										Writes data into DD RAM or CG RAM.	40 μ s	120 μ s
Read Data CG or RAM	1	1	Read Data										Reads data from DD RAM or CG RAM.	40 μ s	120 μ s
I/D = 1: Increment (+1) I/D = 0: Decrement (-1) S = 1: Accompanies display shift. S/C = 1: Display shift S/C = 0: Cursor move R/L = 1: Shift to the right. R/L = 0: Shift to the left. DL = 1: 8 bits DL = 0: 4 bits N = 1: 2 lines N = 0: 1 line F = 1: 5 x 10 dots F = 0: 5 x 7 dots ef = 1: Internally operating ef = 0: Can accept instruction DD RAM: Display data RAM CG RAM: Character generator RAM ACG: CG RAM address ADD: DD RAM address Corresponds to cursor address. AC: Address counter used for both of DD and CG RAM address.											Execution time changes when frequency changes. (Example) When fosc is 270 kHz: $40 \mu s \times \frac{250}{270} = 37 \mu s$				

effect

1. Applied to models driven by 1/8 duty or 1/11 duty.
 2. Applied to models driven by 1/16 duty.

1. CLEAR DISPLAY

Clear display

	RS	R/W	DB ₇							DB ₀
Code	0	0	0	0	0	0	0	0	0	1

คำสั่งนี้จะเป็นการเขียนช่องว่างหรือ SPACE (ASCII 20H) เข้าไปใน DD RAM ทั้งหมดและทำการ SET DD RAM ADDRESSER เป็นศูนย์ ตัว CURSOR จะกลับไปอยู่ตำแหน่งบนสุดซ้ายมือของจอภาพ SET I/D = 1, S ไม่มีการเปลี่ยน

2. RETURN HOME

Return home

	RS	R/W	DB ₇							DB ₀
Code	0	0	0	0	0	0	0	0	1	.

* No effect

คำสั่งนี้จะทำการ SET DD RAM ADDRESSER เป็นศูนย์ ตัว CURSOR จะกลับไปอยู่ตำแหน่งบนสุดซ้ายมือของจอภาพข้อมูลในจอภาพไม่เปลี่ยน

3. ENTRY MODE SET

Entry mode set

	RS	R/W	DB ₇							DB ₀
Code	0	0	0	0	0	0	0	1	I/D	S

BIT I/D : โดยจะเป็นตัวกำหนดให้ว่า เมื่อ เขียนหรืออ่านข้อมูลแล้วจะทำให้ DD RAM ADDRESS เริ่มขึ้นที่ขึ้นหรือลดลงทั้งนี้ โดย 1 = เพิ่มขึ้น
0 : ลดลงทั้งนี้

BIT S : เป็นตัวกำหนดแสดงผลโดยถ้า $S = 1$ จะเป็นการใช้ข้อมูลแล้วตัว
 CURSOR อยู่กับที่ข้อมูลจะถูกดันไปทางซ้าย ถ้า $S = 0$ ข้อมูลจะ
 อยู่กับที่ตัว CURSOR จะถูกดันไปทางขวามือ

4. DISPLAY ON/OFF CONTROL

Display ON/OFF control

	RS	R/W	DB ₇							DB ₀
Code	0	0	0	0	0	0	1	D	C	B

BIT D : เป็น BIT ให้เปิดปิดหน้าจอโดยถ้า $D = 1$ จะ ON และ
 $D = 0$ จะ OFF

BIT C : จะให้แสดง CURSOR ให้ BIT $C = 1$ และถ้าไม่ต้องการ
 แสดง CURSOR BIT $C = 0$ โดยตัว CURSOR จะอยู่ที่ LINE
 ที่ 8 ในโหมด 5X7 DOT และจะอยู่ที่ LINE ที่ 11 ในโหมด
 5X10 DOT

BIT B : เป็น BIT SET การกะพริบของ CURSOR โดย $B = 1$
 มีการกะพริบ $B = 0$ ไม่มีการกะพริบ โดยมีระยะเวลาการ
 กะพริบประมาณ 379.2 ms



(a) Cursor Display Example

(b) Blink Display Example

5. CURSOR OR DISPLAY SHIFT

Cursor or display shift

	RS	R/W	DB ₇							DB ₀
Code	0	0	0	0	0	1	S/C	R/L	.	.

* No effect

เป็นคำสั่งกำหนดให้ตำแหน่ง CURSOR หรือข้อมูลไปเกิดทางซ้ายหรือขวาโดยไม่ต้องใช้
คำสั่งเขียนหรืออ่าน โดย

S/C R/L

0 0 ทำการเขียน CURSOR ไปจากตำแหน่งเดิมไปซ้ายมือ 1 ตำแหน่ง

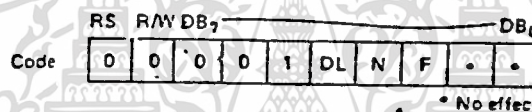
0 1 ทำการเขียน CURSOR ไปจากตำแหน่งเดิมไปขวามือ 1 ตำแหน่ง

1 0 เป็นการลบตัวอักษรที่เกิดไปทางซ้าย

1 1 เป็นการลบตัวอักษรที่เกิดไปทางขวามือ

6. FUNCTION SET

Function set



BIT DL : เป็นการ SET การติดต่อกับจอให้เป็นแบบ 8 BIT หรือ 4 BIT
โดยถ้าต้องการติดต่อ 4 BIT DL = 0 และ 8 BIT DL = 1

N : เป็นการ SET บรรทัดการแสดงผล N = 0 แสดง 1 บรรทัด
N = 1 แสดง 2 บรรทัด หมายความว่ามากกว่า 2 บรรทัด ก็ให้
SET N = 1

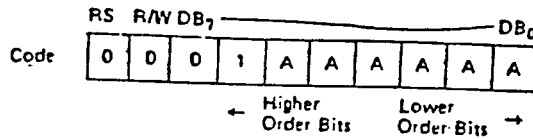
F : เป็นการ SET ขนาด DOT การแสดงผล 5X7 หรือ 5X10 โดย
F = 0 เป็นแบบ 5X7 และ F = 1 เป็นแบบ 5X10

N F	No. of display lines	Character font	Duty factor	Remarks
0 0	1	5 x 7 dots	1/8	
0 1	1	5 x 10 dots	1/11	
1 .	2	5 x 7 dots	1/16	Cannot display 2 lines with 5 x 10 dot character font.

* No effect

7. SET CG RAM ADDRESS

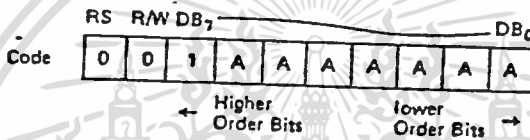
Set CG RAM address



ใน HD44780 นั้นจะมีหน่วยความจำอยู่ 2 ชุด คือ DISPLAY DATA RAM (DD RAM) จำนวน 80X8 BIT และ CHARACTER GENERATOR ROM CG RAM จำนวน 512 BIT และ 7200 BIT ถ้าสิ่งนี้จะเป็นการ SET ADDRESS ใน CG RAM โดยต้องทำการ SET ADDRESS ก่อนเขียนหรืออ่านข้อมูลจาก CG RAM ด้วย

8. SET DD RAM ADDRESS

Set DD RAM address



เป็นคำสั่ง SET ค่า ADDRESS ใน DD RAM ในการเขียนหรืออ่านค่าจาก DD RAM (DD RAM คือ ส่วนที่จะแสดงผลที่จอ LCD) โดยจำนวน ADDRESS ที่จะเกิดขึ้นบนจอ LCD จะอยู่กับ SET ค่า N ด้วย

ถ้า N = 0 (1 บรรทัด) ADDRESS จะอยู่ 00H-4FH

ถ้า N = 1 (2 บรรทัด) ADDRESS จะอยู่ 00H-27H สำหรับบรรทัดที่ 1 และ 40H-67H สำหรับบรรทัดที่ 2

ตัวอย่างการจัด ADDRESS ของ DD RAM ที่จอ LCD แบบ 16 ตัวอักษร 4 บรรทัด และ 20 ตัวอักษร 2 บรรทัด HDM-16416H, HDM-20216H

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
1-line	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	← display position
2-line	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	← DD RAM address
3-line	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	
4-line	50	51	52	53	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	

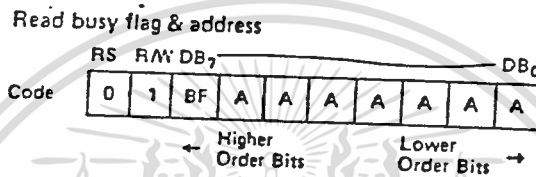
HDM-16416H

	15	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	← display position
1-line	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	10	11	12	13	← DD RAM address
2-line	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F	50	51	52	53	
3-line	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23	24	25	26	27	
4-line	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63	64	65	66	67	

(Note) Shift display is as same as 2-line type.

HDM-20216H

9. READ BUSY FLAG AND ADDRESS



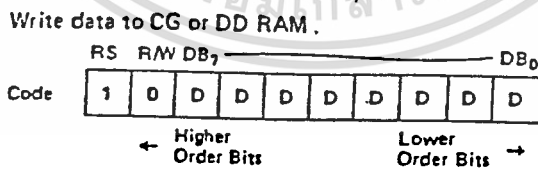
เป็นคำสั่งอ่านค่า BUSY FLAG ซึ่งจะ เป็นตัวบอกว่าตัว HD44780 นี้อยู่ในโหมดการ
ทำงานภายในอยู่หรืออยู่ในสภาวะพร้อมจะรับข้อมูล โดย

BF = 1 อยู่โหมดการทำงานภายในไม่ว่าพร้อมจะรับข้อมูลหรือไม่

BF = 0 พร้อมจะรับข้อมูลหรือคำสั่งได้

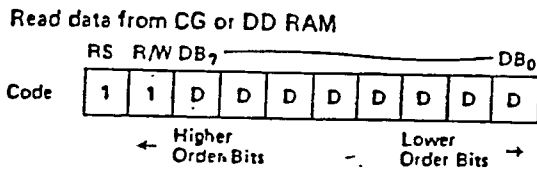
และนอกจากนี้ยังเป็นคำสั่งอ่านค่าข้อมูล ADDRESS ของ CG RAM หรือ DD RAM ด้วย

10. WRITE DATA TO CG หรือ DD RAM



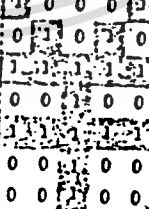
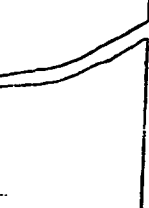
เป็นคำสั่งเขียนข้อมูลเข้าไปใน CG หรือ DD RAM โดยเมื่อเขียนข้อมูลและ ADDRESS
จะเริ่มหรือลดโดยอัตโนมัติตามคำสั่งที่ SET ใน ENTRY MODE ข้อกำหนดที่จะรู้ว่าเป็นการ
เขียนข้อมูลของ CG RAM หรือ DD RAM ทำได้โดยการ SET ADDRESS ของ CG RAM หรือ DD
RAM ขึ้นมาก่อนจะเขียนข้อมูล

11. READ DATA FROM CG OR DD RAM



เป็นคำสั่งอ่านค่าข้อมูลจาก CG RAM หรือ DD RAM โดยก่อนอ่านค่าจาก DD RAM หรือ CG RAM นี้ควรจะใช้คำสั่ง SET ADDRESS ก่อนเพื่อให้รู้ว่าข้อมูลที่อ่านได้นั้นเป็น DD หรือ CG RAM

จากตารางการทำงานจะเห็นว่าการใช้งาน LCD MODULE นั้นง่ายเพียงแค่เราส่งคำสั่งเริ่มแรกและ SET ความต้องการในขนาดตัวอักษร, CURSOR หลังจากนั้นก็เราสามารถเขียนตัวอักษรเข้าไปใน DD RAM ตามตารางตัวอักษรที่นำมาตั้งก็จะเกิดอักษรในจอภาพ LCD เรายังสามารถกำหนดตำแหน่งตัวอักษรที่จะให้เกิดผลได้โดยการ SET DD RAM ADDRESS ตามตารางที่นำมาให้หรือ SET DD RAM ADDRESS ขอให้ทดสอบทำความเข้าใจกับตัวโปรแกรมที่ใช้กับ ET-BOARD V3.0 นี้ที่นำมาจะเห็นว่ามีส่วนเริ่มต้นคือ ส่วนการ INITIAL LCD เพื่อกำหนดหน้าที่การทำงานต่างๆ

Character Codes (DD RAM Data)								CG RAM Address								Character Patterns (CG RAM Data)								For 5 x 7 dot character pattern																										
7	6	5	4	3	2	1	0	5	4	3	2	1	0	7	6	5	4	3	2	1	0	7	6	5	4	3	2	1	0																					
Higher				Lower				Higher				Lower				Higher				Lower																														
0 0 0 0 x 0 0 0								0 0 0				0	0	0	x	x	x	1	1	1	1	0									Character Pattern Example (1)																			
												0	0	1	x	x	x	0	0	0	0	0																												
												0	1	0	x	x	x	0	0	0	0	0																												
												0	1	1	x	x	x	0	0	0	0	0																												
												0 0 0 0 x 0 0 1								0 0 1				0	0	0	x	x	x	1	0	0	0	0									Character Pattern Example (2)							
																								0	0	1	x	x	x	0	1	0	1	0																
																								0	1	0	x	x	x	0	0	1	0	0																
																								0	1	1	x	x	x	0	0	1	0	0																
0 0 0 0 x 1 1 1								1 1 1																0	0	0	x	x	x	0	0	0	0	0									Character Pattern Example (3)							
																								0	0	1	x	x	x	0	0	0	0	0																
																								1	0	0	x	x	x	0	0	0	0	0																
																								1	0	1	x	x	x	0	0	0	0	0																
																								1	1	1	x	x	x	0	0	0	0	0																

* No effect

13-

* No effect

CHARACTER FONT TABLE

Higher bits 4b1	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111
ADDR 0000	CG RAM (1)												
ADDR 0001	(2)												
ADDR 0010	(3)												
ADDR 0011	(4)												
ADDR 0100	(5)												
ADDR 0101	(6)												
ADDR 0110	(7)												
ADDR 0111	(8)												
ADDR 1000	(9)												
ADDR 1001	(2)												
ADDR 1010	(3)												
ADDR 1011	(4)												
ADDR 1100	(5)												
ADDR 1101	(6)												
ADDR 1110	(7)												
ADDR 1111	(8)												

NOTE: CGRAM is a CHARACTER GENERATOR RAM having a storage function of character pattern which enable to change freely by user's program.

HANDOK

8279/8279-5 (initial version)

PROGRAMMABLE KEYBOARD/DISPLAY INTERFACE

- Simultaneous Keyboard, Display Operations
- Scanned Keyboard Mode
- Scanned Sensor Mode
- Strobed Input Entry Mode
- 6-Character Keyboard FIFO
- 2-Key Lockout or N-Key rollover with Contact Debounce
- Dual 6- or 16-Numerical Display
- Single 16-Character Display
- Right or Left Entry 16-Byte Display RAM
- Mode Programmable from CPU
- Programmable Scan Timing
- Interrupt Output on Key Entry
- Available in EXPRESS
 - Standard Temperature Range
 - Extended Temperature Range

The Intel® 8279 is a general purpose programmable keyboard and display I/O interface device designed for use with Intel® microprocessors. The keyboard portion can provide a scanned interface to a 64-contact key matrix. The keyboard portion will also interface to an array of sensors or a strobed interface keyboard, such as the hall effect and ferrite variety. Key depression can be 2-key lockout or N-key rollover. Keyboard entries are debounced and strobed in an 8-character FIFO. If more than 8 characters are entered, overrun status is set. Key entries set the interrupt output line to the CPU.

The display portion provides a scanned display interface for LED, incandescent, and other popular display technologies. Both numeric and alphanumeric segment displays may be used as well as simple indicators. The 8279 has 16x8 display RAM which can be organized into dual 16x4. The RAM can be loaded or interrogated by the CPU. Both right entry, calculator and left entry typewriter display formats are possible. Both read and write of the display RAM can be done with auto-increment of the display RAM address.

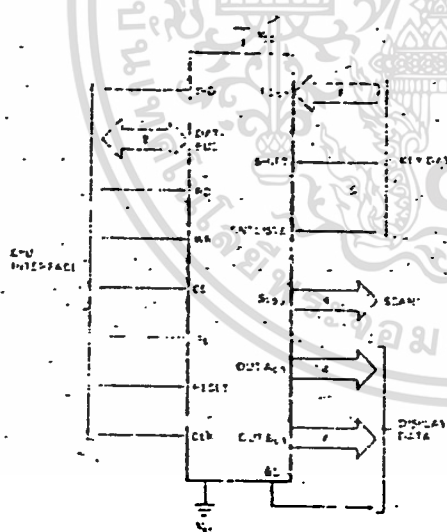


Figure 1. Logic Symbol

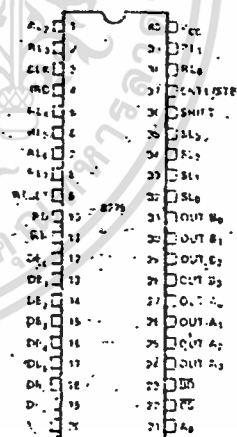


Figure 2. Pin Configuration

HARDWARE DESCRIPTION

The 8279 is packaged in a 40 pin DIP. The following is a functional description of each pin.

Table 1. Pin Descriptions

Symbol	Pin No.	Name and Function	Symbol	Pin No.	Name and Function
AD ₀ -DB ₇	15-22	Bi-directional data bus: All data and commands between the CPU and the 8279 are transmitted on these lines.	SHIFT	31	Shift: The shift input status is stored along with the key position on key closure in the 8279 in keyboard modes. It has an active internal pullup to keep it high until a switch closure pulls it low.
CLK	3	Clock: Clock from system used to generate internal timing.	CNTL	37	Control/Strobed Input Mode: For keyboard modes this line is used as a control input and stored like status on a key closure. The line is also the strobe line that enters the data into the FIFO in the Strobed Input mode. (Rising Edge) it has an active internal pullup to keep it high until a switch closure pulls it low.
RESET	9	Reset: A high signal on this pin resets the 8279. After being reset the 8279 is placed in the following mode: 1) 16 8-bit character display — left entry 2) Encoded scan keyboard — 2 key lockout Along with this the program clock prescaler is set to 31.	OUT ₀ -OUT ₁ A ₀ OUT ₂ -OUT ₃ B ₀	39-40	Outputs: These two ports are the outputs for the 16 x 4 display refresh registers. The data from these outputs is synchronized to the 8279 clock (CLK-SL) for multiplexed digit displays. The two 4 bit ports may be blanked independently. These two ports may also be considered as one 8-bit port.
CS	72	Chip Select: A low on this pin enables the interface functions to receive or transmit.	BLANK	25	Blank Display: This output is used to blank the display during digit switching or by a display blanking command.
RA	21	Buffer Address: A high on this line indicates the signal, in or out are interpreted as a command or status. A low indicates that they are data.			
RD, WR	10-11	Input/Output Read and Write: These signals enable the data buffers to either send data to the external bus or receive it from the external bus.			
IRQ	4	Interrupt Request: In a keyboard mode, the interrupt line is high when there is data in the FIFO/Sensor RAM. The interrupt line goes low with each FIFO/Sensor RAM read and returns high if there is still information in the RAM. In a sensor mode, the interrupt line goes high whenever a change in a sensor is detected.			
V _{cc} , V _{ee}	27-40	Ground and power supply pins.			
SL ₀ -SL ₃	22-25	Scan Lines: Scan lines which are used to scan the key switch or sensor matrix and the display digits. These lines can be either encoded (1 of 15) or decoded (1 of 4).			
RL ₀ -RL ₃	38, 39, 5-8	Return Line: Return line inputs which are connected to the scan lines through the keys or sensor switches. They have active internal pullups to keep them high until a switch closure pulls one low. They also serve as an 8-bit input in the Strobed Input mode.			

FUNCTIONAL DESCRIPTION

Since data input and display are an integral part of many microprocessor designs, the system designer needs an interface that can control these functions without placing a large load on the CPU. The 8279 provides this function for 8-bit microprocessors.

The 8279 has two sections: keyboard and display. The keyboard section can interface to regular typewriter style keyboards or random toggle or thumb switches. The display section drives alphanumeric displays or a bank of indicator lights. Thus the CPU is relieved from scanning the keyboard or refreshing the display.

The 8279 is designed to directly connect to the microprocessor bus. The CPU can program all operating modes for the 8279. These modes include:

Input Modes

- Scanned Keyboard — with encoded (8 x 8 key keyboard) or decoded (4 x 8 key keyboard) scan lines. A key depression generates a 6-bit encoding of key position. Position and shift and control status are stored in the FIFO. Keys are automatically debounced with 2-key lockout or N-key rollover.
- Scanned Sensor Matrix — with encoded (8 x 8 matrix switches) or decoder (4 x 8 matrix switches) scan lines. Key status (open or closed) stored in RAM addressable by CPU.
- Strobed Input — Data on return lines during control line strobe is transferred to FIFO.

Output Modes

- 8 or 16 character multiplexed displays that can be organized as dual 4-bit or single 8-bit ($D_0 = D_0, A_3 = D_7$).
- Right entry or left entry display formats.

Other features of the 8279 include:

- Mode programming from the CPU.
- Clock Prescaler
- Interrupt output to signal CPU when there is keyboard or sensor data available.
- An 8 byte FIFO to store keyboard information.
- 16 byte internal Display RAM for display refresh. This RAM can also be read by the CPU.

PRINCIPLES OF OPERATION

The following is a description of the major elements of the 8279 Programmable Keyboard/Display interface device. Refer to the block diagram in Figure 3.

I/O Control and Data Buffers

The I/O Control section uses the $\overline{CS}$, A_0 , $\overline{RD}$ and $\overline{WR}$ lines to control data flow to and from the various internal registers and buffers. All data flow to and from the 8279 is enabled by $\overline{CS}$. The direction of the information, given or decided by the CPU, is identified by A_0 . A logic one means the information is a command or status. A logic zero means the information is data. $\overline{RD}$ and $\overline{WR}$ determine the direction of data flow through the Data Buffers. The Data Buffers are bi-directional buffers that connect the internal bus to the external bus. When the chip is not selected ($\overline{CS} = 1$), the device is in a high impedance state. The drivers input during $\overline{WR} \cdot \overline{CS}$ and output during $\overline{RD} \cdot \overline{CS}$.

Control and Timing Registers and Timing Control

These registers store the keyboard and display modes and other operating conditions programmed by the CPU. The modes are programmed by presenting the proper command on the data lines with $A_0 = 1$ and then sending a $\overline{WR}$. The command is latched on the rising edge of $\overline{WR}$.

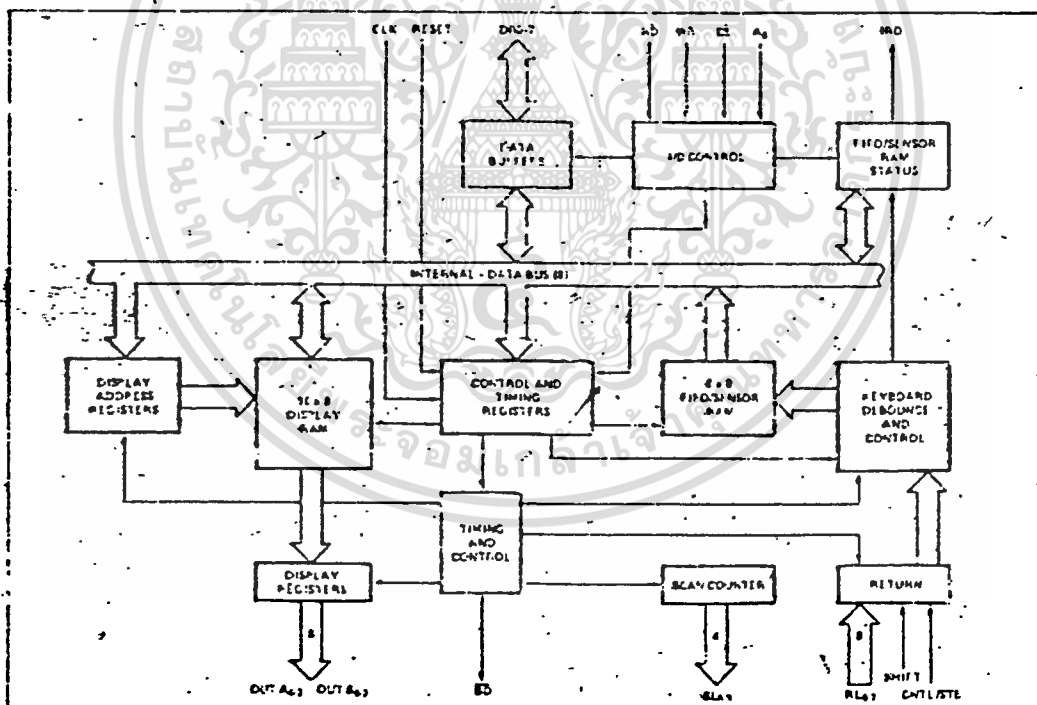


Figure 2. Internal Block Diagram

This command is then decoded and the appropriate function is set. The timing control contains the timing counter chain. The first counter is a 4-bit prescaler that can be programmed to yield an internal frequency of 100 kHz which gives a 5.1 ms keyboard scan time and a 10.5 ms debounce time. The other counters divide down the basic internal frequency to provide the proper key scan, row scan, keyboard matrix scan, and display scan times.

Scan Counter

The scan counter has two modes. In the encoded mode, the counter provides a binary count that must be externally decoded to provide the scan lines for the keyboard or display. In the decoded mode, the scan counter decodes the least significant 2 bits and provides a decoded 1 of 4 scan. Note that when the keyboard is in decoded scan, so is the display. This means that only the first 4 characters in the Display RAM are displayed.

In the encoded mode, the scan lines are active high. In the decoded mode, the scan lines are active low outputs.

Return Buffers and Keyboard Debounce and Control

The return lines are buffered and latched by the Return Buffers. In the keyboard mode, these lines are scanned looking for key closures in that row. If the detector circuit detects a closed switch, it waits about 10 ms to check if the switch remains closed. If it does, the status of the switch in the matrix plus the status of SHIFT and CONTROL are transferred to the FIFO. In the scanned Sensor Matrix mode, the contents of the return lines are directly transferred to the corresponding row of the Sensor RAM (FIFO) each key scan time. In Strobed Input mode, the contents of the return lines are transferred to the FIFO on the rising edge of the CNTLSTB line pulse.

FIFO/Sensor RAM and Status

This block is a dual function 8 x 8 RAM. In Keyboard or Strobed Input modes, it is a FIFO. Each new entry is written to successive RAM positions and each is then read in order of entry. FIFO status keeps track of the number of characters in the FIFO and whether it is full or empty. Too many reads or writes will be recognized as an error. The status can be read by an RD with CS low and A high. The status logic also provides an IRO signal when the FIFO is not empty. In Scanned Sensor Matrix mode, the memory is a Sensor RAM. Each row of the Sensor RAM is loaded with the status of the corresponding row of sensor in the sensor matrix. In this mode, IRO is high if a change in a sensor is detected.

Display Address Registers and Display RAM

The Display Address Registers hold the address of the word currently being written or read by the CPU and the two 4-bit nibbles being displayed. The read/write addresses are programmed by CPU commands. They also can be set to auto-increment after each read or write. The Display RAM can be directly read by the CPU after the correct mode and address is set. The 16 addresses for the A and B nibbles are automatically generated by the E279 to match data entry by the CPU. The A and B nibbles can be generated independently or as one word, according to the mode that is set by the CPU. Data entry to the display can be set to either left or right entry. See Interface Considerations for details.

SOFTWARE OPERATION

8279 commands

The following commands program the 8279 operating modes. The commands are sent on the Data Bus with CS low and A₀ high and are loaded on the rising edge of WR.

Keyboard/Display Mode Set

	MSB							LSB
Code:	0	0	0	0	0	0	0	0

00000000 is the Display Mode and 0000 is the Keyboard Mode.

DD

- 0 0 8-bit character display — Left entry
- 0 1 16-bit character display — Left entry
- 1 0 8-bit character display — Right entry
- 1 1 16-bit character display — Right entry

For description of right and left entry, see Interface Considerations. Note that when decoded scan is set in Keyboard mode, the display is reduced to 4 characters independent of display mode set.

KKK

- 0 0 0 Encoded Scan Keyboard — 2 Key Lockout *
- 0 0 1 Encoded Scan Keyboard — 3-Key Lockout
- 0 1 0 Encoded Scan Keyboard — N-Key Lockout *
- 0 1 1 Decoded Scan Keyboard — N-Key Lockout
- 1 0 0 Encoded Scan Sensor Matrix
- 1 0 1 Decoded Scan Sensor Matrix
- 1 1 0 Strobed Input, Encoded Display Scan
- 1 1 1 Strobed Input, Decoded Display Scan

Display Clock

Code:	0	0	1	P	P	P	P	P
-------	---	---	---	---	---	---	---	---

All timing and multiplexing signals for the 8279 are generated by an internal prescaler. This prescaler divides the external clock (pin 3) by a programmable integer. Bits P₄P₃P₂P₁P₀ determine the value of this integer which ranges from 2 to 31. Choosing a divisor that yields 100 kHz will give the specified scan and debounce times. For instance, if Pin 3 of the 8279 is being clocked by a 2 MHz signal, P₄P₃P₂P₁P₀ should be set to 10100 to divide the clock by 20 to yield the proper 100 kHz operating frequency.

Read FIFO/Sensor RAM

Code:	0	1	0	A	I	X	A	A	A
-------	---	---	---	---	---	---	---	---	---

The CPU sends up the 8279 command of the FIFO/Sensor RAM by first writing this command. In the Scan Key-

Interfacing section.

board Mode, the Auto-Increment flag (AI) and the RAM address bits (AAA) are irrelevant. The 8279 will automatically drive the data bus for each subsequent read ($A_0 = 0$) in the same sequence in which the data first entered the FIFO. All subsequent reads will be from the FIFO until another command is issued.

In the Sensor Matrix Mode, the RAM address bits AAA select one of the 8 rows of the Sensor RAM. If the AI flag is set ($AI = 1$), each successive read will be from the subsequent row of the sensor RAM.

Read Display RAM

Code:

0	1	1	AI	A	A	A	A
---	---	---	----	---	---	---	---

The CPU sets up the 8279 for a read of the Display RAM by first writing this command. The address bits AAAA select one of the 16 rows of the Display RAM. If the AI flag is set ($AI = 1$), this row address will be incremented after each following read or write to the Display RAM. Since the same counter is used for both reading and writing, this command sets the next read or write address and the sense of the Auto-Increment mode for both operations.

Write Display RAM

Code:

1	0	0	AI	A	A	A	A
---	---	---	----	---	---	---	---

The CPU sets up the 8279 for a write to the Display RAM by first writing this command. After writing the command with $A_0 = 1$, all subsequent writes with $A_0 = 0$ will be to the Display RAM. The addressing and Auto-Increment functions are identical to those for the Read Display RAM. However, this command does not affect the source of subsequent Data Reads; the CPU will read from whichever RAM (Display or FIFO/Sensor) which was last specified. If, indeed, the Display RAM was last specified, the Write Display RAM will, nevertheless, change the next Read location.

Display Write Inhibit/Blanking

Code:

1	0	1	X	IW	IW	BL	BL
---	---	---	---	----	----	----	----

The IW Bits can be used to mask nibble A and nibble B in applications requiring separate 4-bit display ports. By setting the IW flag ($IW = 1$) for one of the ports, the port becomes masked so that entries to the Display RAM from the CPU do not affect that port. Thus, if each nibble is input to a BCD decoder, the CPU may write a digit to the Display RAM without affecting the other digit being displayed. It is important to note that bit E_0 corresponds to bit D_0 on the CPU bus, and that bit A_3 corresponds to bit D_7 .

If the user wishes to blank the display, the BL flags are available for each nibble. The last Clear command issued determines the code to be used as a "blank." This code defaults to all zeros after a reset. Note that both BL flags must be set to blank a display formatted with a single 8-bit port.

Clear

Code:

1	1	0	C_D	C_D	C_D	C_F	C_A
---	---	---	-------	-------	-------	-------	-------

The C_D bits are available in this command to clear all rows of the Display RAM to a selectable blanking code as follows:

C_D	C_D	C_D	
0	X		All Zeros (X = Don't Care)
1	0		AR = Hex 20 (0010 0000)
1	1		All Ones

Enable clear display (i.e., $C_F = 1$)

During the time the Display RAM is being cleared (~160 μ s), it may not be written to. The most significant bit of the FIFO status word is set during this time. When the Display RAM becomes available again, it automatically resets.

If the C_F bit is asserted ($C_F = 1$), the FIFO status is cleared and the interrupt output line is reset. Also, the Sensor RAM pointer is set to row 0.

C_A , the Clear All bit, has the combined effect of C_D and C_F . It uses the C_D clearing code on the Display RAM and also clears FIFO status. Furthermore, it resynchronizes the internal timing chain.

End Interrupt/Error Mode Set

Code:

1	1	1	E	X	X	X	X
---	---	---	---	---	---	---	---

 X = Don't care

For the sensor matrix modes this command lowers the IRQ line and enables further writing into RAM. (The IRQ line would have been raised upon the detection of a change in a sensor value. This would have also inhibited further writing into the RAM until reset.)

For the N-key rollover mode — if the E bit is programmed to "1" the chip will operate in the special Error mode. (For further details, see Interface Considerations Section.)

Status Word

The status word contains the FIFO status, error, and display unavailable signals. This word is read by the CPU when A_0 is high and $\overline{CS}$ and $\overline{RD}$ are low. See Interface Considerations for more detail on status word.

Data Read

Data is read when A_0 , $\overline{CS}$ and $\overline{RD}$ are all low. The source of the data is specified by the Read FIFO or Read Display commands. The trailing edge of $\overline{RD}$ will cause the address of the RAM being read to be incremented if the Auto-Increment flag is set. FIFO reads always increment (if no error occurs) independent of AI.

Data Write

Data that is written with A_0 , $\overline{CS}$ and $\overline{WR}$ low is always written to the Display RAM. The address is specified by the latest Read Display or Write Display command. Auto-Incrementing on the rising edge of $\overline{WR}$ occurs if AI set by the latest display command.

INTERFACE CONSIDERATIONS

Scanned Keyboard Mode, 2-Key Lockout

There are three possible combinations of conditions that can occur during debounce scanning. When a key is depressed, the debounce logic is set. Other depressions are looked for during the next two scans. If no keys are encountered, it is a single key depression, and the key position is entered into the FIFO along with the status of CNTL and SHIFT lines. If the FIFO was empty, IRQ will be set to signal the CPU that there is an entry in the FIFO. If the FIFO was full, the key will not be entered and the error flag will be set. If another closed switch is encountered, no entry to the FIFO can occur. If all other keys are released before this one, then it will be entered to the FIFO. If this key is released before any other, it will be entirely ignored. A key is entered to the FIFO only once per depression, no matter how many keys were pressed along with it or in what order they were released. If two keys are depressed within the debounce cycle, it is a simultaneous depression. Neither key will be recognized until one key remains depressed alone. The last key will be treated as a single key depression.

Scanned Keyboard Mode, N-Key Rollover

With N-key Rollover each key depression is treated independently from all others. When a key is depressed, the debounce circuit waits for keyboard scans and then checks to see if the key is still down. If it is, the key is entered into the FIFO. Any number of keys can be depressed and another can be recognized and entered into the FIFO. If a simultaneous depression occurs, the keys are recognized and entered according to the order the keyboard scan found them.

Scanned Keyboard — Special Error Modes

For N-key rollover mode the user can program a special error mode. This is done by the "End Interrupt/Error Mode Set" command. The debounce cycle and key-validity check are as in normal N-key mode. If during a single debounce cycle, two keys are found depressed, this is considered a simultaneous multiple depression, and sets an error flag. This flag will prevent any further writing into the FIFO and will set interrupt (if not yet set). The error flag could be read in this mode by reading the FIFO STATUS word. (See "FIFO STATUS" for further details.) The error flag is reset by sending the normal CLEAR command with Cr = 7.

Sensor Matrix Mode

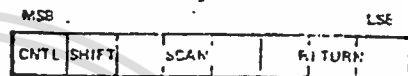
In Sensor Matrix mode, the debounce logic is inhibited. The status of the sensor switch is inputted directly to the Sensor RAM. In this way the Sensor RAM keeps an image of the state of the switches in the sensor matrix. Although debounce timing is not provided, this mode has the advantage that the CPU knows how long the sensor was closed and when it was released. A keyboard mode can only indicate a validated closure. To make the software easier, the designer should functionally group the sensors by row since this is the format in which the CPU will read them. The IRQ line goes high if any sensor value change is detected at the end of a sensor matrix scan. The IRQ line is cleared by the first data read operation if the Auto-

increment flag is set to zero, or by the End Interrupt command if the Auto-Increment flag is set to one.

Note: Multiple changes in the matrix Addressed by (SLC = 0) may cause multiple interrupts. (SLC = 0 in the Decoded Mode). Reset may cause the 8279 to see multiple changes.

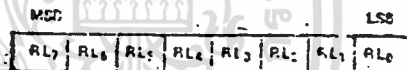
Data Format

In the Scanned Keyboard mode, the character entered into the FIFO corresponds to the position of the switch in the keyboard plus the status of the CNTL and SHIFT lines (non-inverted). CNTL is the status of the character and SHIFT is the next most significant bit. The next three bits are from the scan counter and indicate the row the key was found in. The last three bits are from the column counter and indicate to which return line the key was connected.

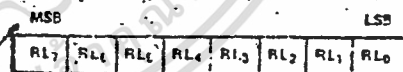


SCANNED KEYBOARD DATA FORMAT

In Sensor Matrix mode, the data on the return lines is entered directly in the row of the Sensor RAM that corresponds to the row in the matrix being scanned. Therefore, each switch position maps directly to a Sensor RAM position. The SHIFT and CNTL inputs are ignored in this mode. Note that switches are not necessarily the only thing that can be connected to the return lines in this mode. Any logic that can be triggered by the scan lines can enter data to the return line inputs. Eight multiplexed input ports could be tied to the return lines and scanned by the 8279.



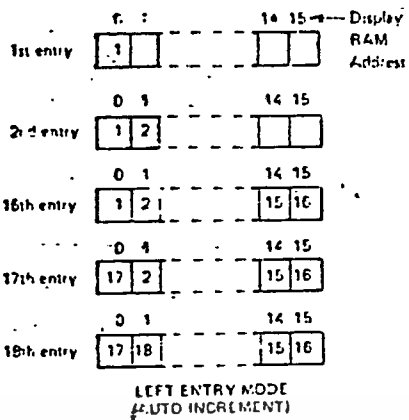
In Strobed Input mode, the data is also entered to the FIFO from the return lines. The data is entered by the rising edge of a CNTL/STB line pulse. Data can come from either an encoded keyboard or simple switch matrix. The return lines can also be used as a general purpose strobed input.



Display

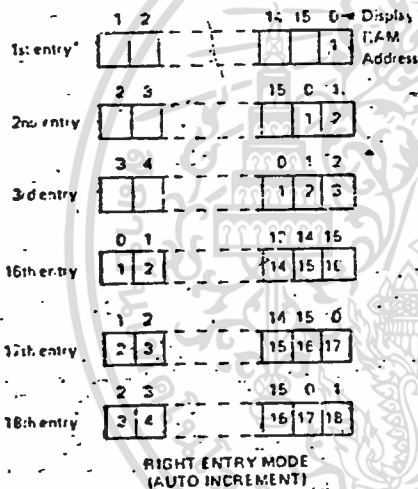
Left Entry

Left Entry mode is the simplest display format in that each display position directly corresponds to a byte (0-n) in the Display RAM. Address 0 in the RAM is the left most display character and address 15 (or address 7 in 8 character display) is the right most display character. Entering characters from position zero causes the display to fill from the left. The 17th (5th) character is entered back in the left most position and filling again proceeds from there.



Right Entry

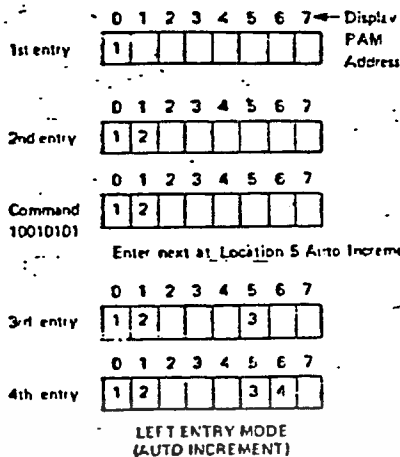
Right entry is the method used by most electronic calculators. The first entry is placed in the right most display character. The next entry is also placed in the right most character after the display is shifted left one character. The left most character is shifted off the end and is lost.



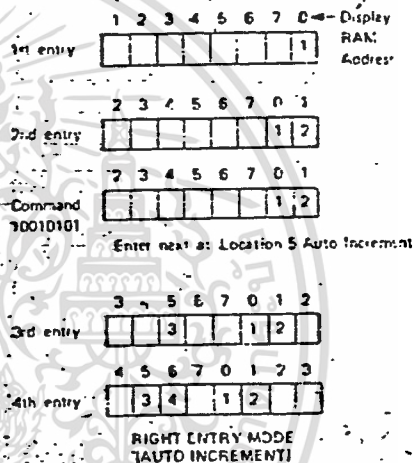
Note that now the display position and register address do not correspond. Consequently, entering a character to an arbitrary position in the Auto Increment mode may have unexpected results. Entry starting at Display RAM address 0 with sequential entry is recommended.

Auto Increment

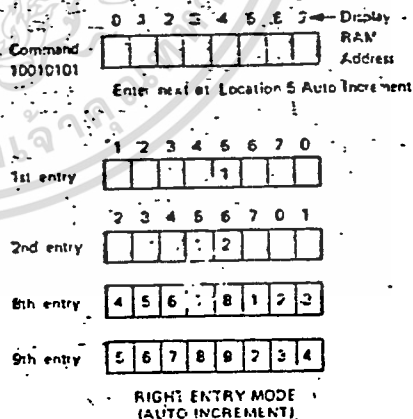
In the Left Entry mode, Auto Incrementing causes the address where the CPU will next write to be incremented by one and the character appears in the next location. With non-Auto Incrementing the entry is both to the same RAM address and display position. Entry to an arbitrary address in the Auto Increment mode has no undesirable side effects and the result is predictable:



In the Right Entry mode, Auto Incrementing and non Incrementing have the same effect as in the Left Entry except if the address sequence is interrupted:



Starting at an arbitrary location operates as shown below:



Entry appears to be from the initial entry point.

8/16 Character Display Formats

If the display mode is set to an 8 character display, the on duty-cycle is double what it would be for a 16 character display (e.g., 5.1 ms scan time for 8 characters vs. 10.3 ms for 16 characters with 100 kHz internal frequency).

G. FIFO Status

FIFO status is used in the Keyboard and Strobed Input modes to indicate the number of characters in the FIFO and to indicate whether an error has occurred. There are two types of errors possible, overrun and underrun. Overrun occurs when the entry of another character into a full FIFO is attempted. Underrun occurs when the CPU tries to read an empty FIFO.

The FIFO status word also has a bit to indicate that the Display RAM was unavailable because a Clear Display or Clear All command had not completed its clearing operation.

In a Sensor Matrix mode, a bit is set in the FIFO status word to indicate that at least one sensor closure indication is contained in the Sensor RAM.

In Special Error Mode the S/E bit is showing the error flag and serves as an indication to whether a simultaneous multiple closure error has occurred.

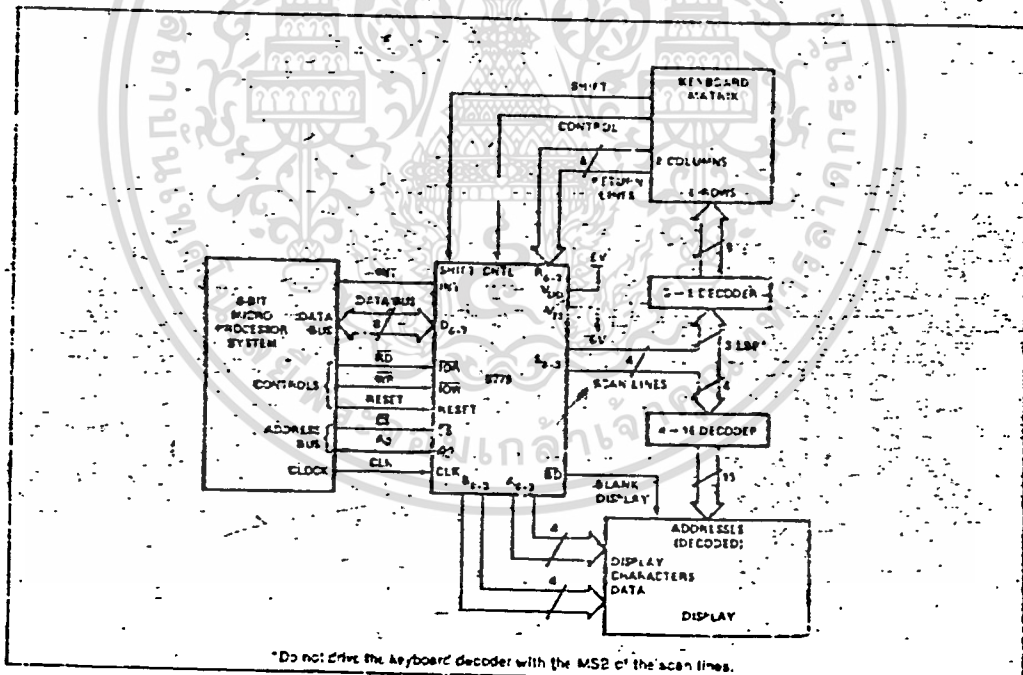
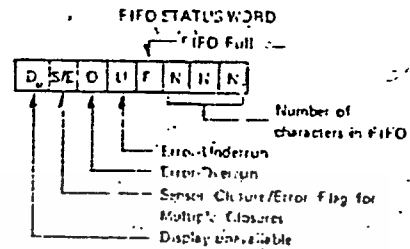


Figure 4. System Block Diagram

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature	0°C to 70°C
Storage Temperature	-65°C to 125°C
Voltage on any Pin with Respect to Ground	-0.5V to +7V
Power Dissipation	1 Watt

*NOTICE: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. CHARACTERISTICS [$T_A = 0^\circ\text{C}$ to 70°C , $V_{SS} = 0\text{V}$, (NOTE 3)]

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{IL1}	Input Low Voltage for Return Lines	-0.5	1.4	V	
V_{IL2}	Input Low Voltage for All Others	-0.5	0.8	V	
V_{IH1}	Input High Voltage for Return Lines	2.2		V	
V_{IH2}	Input High Voltage for All Others	2.0		V	
V_{OL}	Output Low Voltage		0.45	V	Note 1
V_{OH1}	Output High Voltage on Interrupt Line	3.5		V	Note 2
V_{OH2}	Other Outputs	2.4			$I_{OH} = -400\ \mu\text{A}$ 8279-5 $-100\ \mu\text{A}$ 8279
I_{IL1}	Input Current on Shift, Control and Return Lines		+10 -100	μA	$V_{IN} = V_{CC}$ $V_{IN} = 0\text{V}$
I_{IL2}	Input Leakage Current on All Others		±10	μA	$V_{IN} = V_{CC}$ to 0V
I_{OFL}	Output Float Leakage		±10	μA	$V_{OUT} = V_{CC}$ to 0.45V
I_{CC}	Power Supply Current		120	mA	

CAPACITANCE

Symbol	Parameter	Typ.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance	5	10	pF	$f_C = 1\text{ MHz}$ Unmeasured pins returned to V_{SS}
C_{OUT}	Output Capacitance	10	20	pF	

A.C. CHARACTERISTICS [$T_A = 0^\circ\text{C}$ to 70°C , $V_{SS} = 0\text{V}$, (Note 3)]

Bus Parameters

READ CYCLE

Symbol	Parameter	8279		8279-5		Unit
		Min.	Max.	Min.	Max.	
t_{AR}	Address Stable Before READ	50		0		ns
t_{RA}	Address Hold Time for READ	5		0		ns
t_{ER}	READ Pulse Width	420		250		ns
$t_{RD}^{(4)}$	Data Delay from READ		300		150	ns
$t_{AD}^{(4)}$	Address to Data Valid		450		250	ns
t_{DF}	READ to Data Floating	10	100	10	100	ns
t_{RCY}	Read Cycle Time	1		1		μs

A.C. CHARACTERISTICS (Continued)

WRITE CYCLE

Symbol	Parameter	8278		8278-5		Unit
		Min.	Max.	Min.	Max.	
t_{AW}	Address Stable Before WRITE	50		0		ns
t_{WA}	Address Hold Time for WRITE	20		0		ns
t_{WW}	WRITE Pulse Width	400		250		ns
t_{DW}	Data Set Up Time for WRITE	300		150		ns
t_{WD}	Data Hold Time for WRITE	40		0		ns
t_{WCY}	Write Cycle Time	1		1		µs

OTHER TIMINGS

Symbol	Parameter	8278		8278-5		Unit
		Min.	Max.	Min.	Max.	
t_{CW}	Clock Pulse Width	230		120		nsec
t_{CY}	Clock Period	500		420		nsec

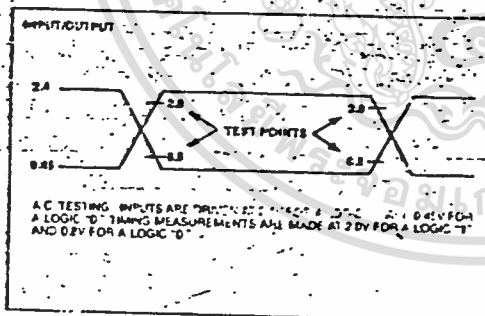
Keyboard Scan Time	5.1 msec	Digit-on Time	50 µsec
Keyboard Debounce Time	10.3 msec	Blanking Time	160 µsec
Key Scan Time	80 µsec	Internal Clock Cycle(s)	10 µsec
Display Scan Time	10.3 msec		

NOTES:

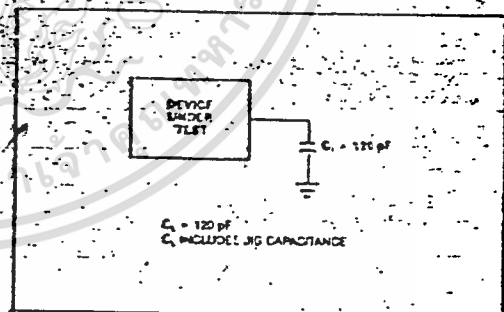
1. 8278, $I_{OL} = 1.5mA$; 8278-5, $I_{OL} = 2.2mA$.
2. $I_{OH} = -100\mu A$.
3. 8278, $V_{CC} = +5V \pm 5\%$; 8278-5, $V_{CC} = +5V \pm 10\%$.
4. 8278, $C_L = 100pF$; 8278-5, $C_L = 150pF$.
5. The Prescaler should be programmed to provide a 10 µs internal clock cycle.

* For Extended Temperature EXPRESS, use M8278A electrical parameters.

A.C. TESTING INPUT, OUTPUT WAVEFORM

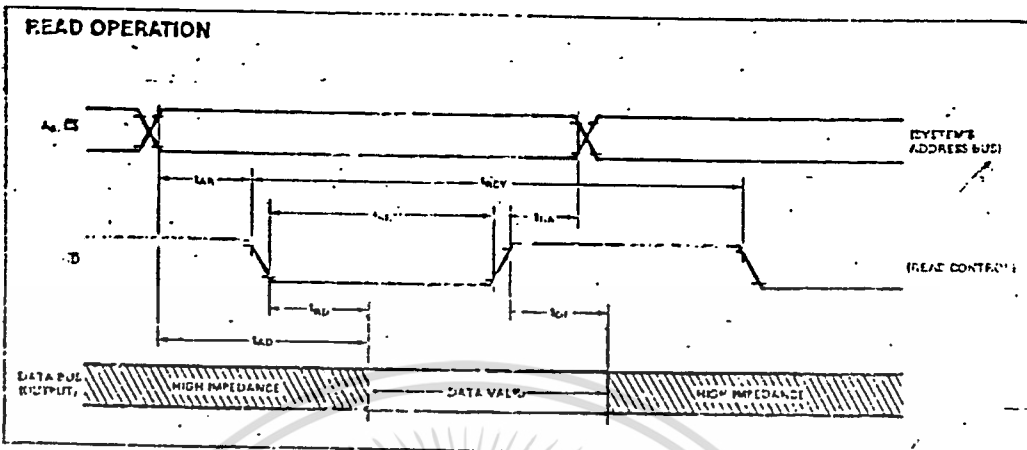


A.C. TESTING LOAD CIRCUIT

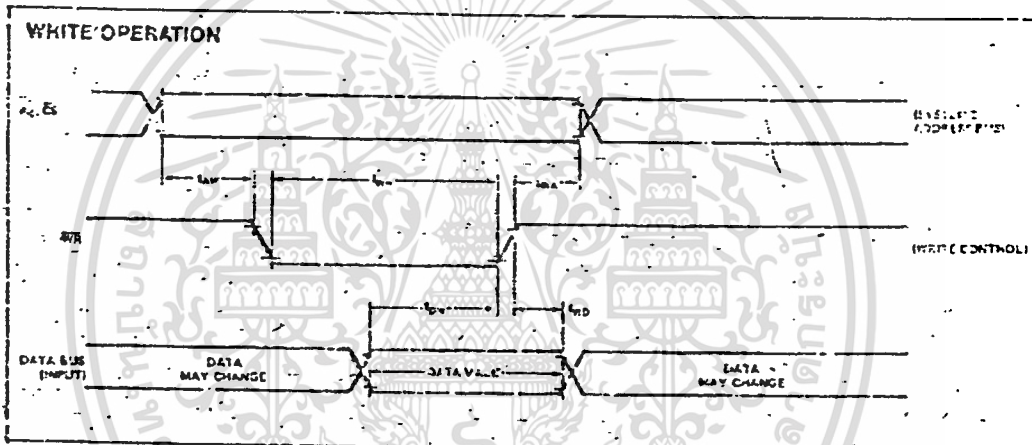


WAVEFORMS

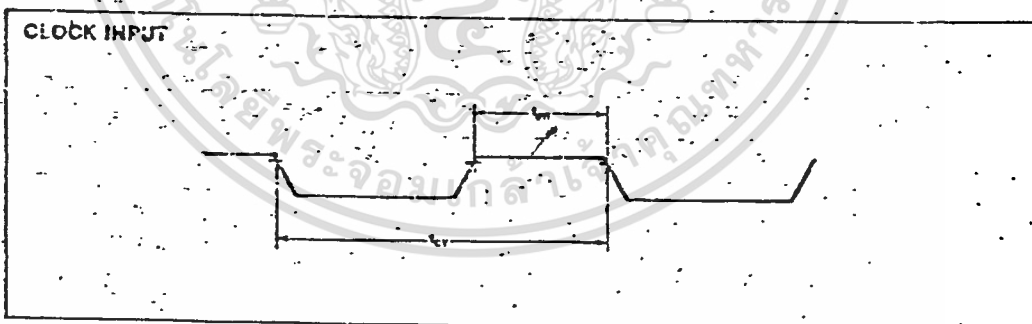
READ OPERATION



WRITE OPERATION



CLOCK INPUT



SCAN



S_1, S_2 ARE NOT KNOWN BUT THEY ARE SIMPLY S_1 DIVIDED BY 2 AND 1

กิตติกรรมประกาศ

วิทยานิพนธ์เรื่องนี้ได้รับความช่วยเหลือทางด้านอุปกรณ์ และ เครื่องมือรวมทั้งสถานที่ศึกษาจาก อาจารย์ อรรถสิทธิ์ หล้าสกุล จึงขอขอบพระคุณอย่างสูงมา ณ.ที่นี้

ขอขอบคุณ บริษัท GLOBAL SOLUTION จำกัด และ คุณ รุ่งฤดี วิทยะ ให้อิมเครื่องคอมพิวเตอร์

ขอขอบคุณ คุณดุสิต ไกรวรรณ ให้อิมเครื่อง ET Board

ขอขอบคุณเพื่อนร่วมสถาบันทุกคนที่ให้อำลัใจในการทำงานและให้ความรู้ลึกๆตลอดมา



หนังสืออ้างอิง

1. ยืน ภู่วรวรรณ "ทฤษฎีและการประยุกต์ไมโครโปรเซสเซอร์ Z-80"
บริษัท ซีเอ็ดดูเคชั่น จำกัด กรุงเทพฯ พ.ศ. 2532
2. Fredrick W. Hughes., "OP Amp Handbook"
Prentice - Hall Inc., U.S.A., 1981
3. HAND BOOK " TTL Databook "
Texas Instruments Inc., U.S.A., 1981
4. James Rignell Robert Donoven "Z 80 Microprocesssor Technology"
Delmar Publishers Inc., 1986
5. เอกสารองค์การโทรศัพท์แห่งประเทศไทย
6. ทีมงาน ETT " RTC นาฬิกาบอกเวลาให้ไมโครโปรเซสเซอร์"
เซมิคอนดักเตอร์ ฉบับที่ 79 (มิย.-กค.) หน้า 143-152 พ.ศ.2530
7. ทีมงาน ETT "จออักษร LCD สำหรับซิงเกิ้ลบอร์ด"
เซมิคอนดักเตอร์ ฉบับที่ 99 (เมษ.-พค.) หน้า 231-240 พ.ศ.2533

.....