จอแสดงผลทางอิเล็กทรอนิกส์

(Electronic Display)

โดย

นายสรพงษ์ อุนนาภิรักษ์

นายสุรสิทธิ์ กิจมงคลชัย

นายเหนือพงศ์ วิบูลย์เมธ

วัน เดือน ปี ๑๙ ม.ค. ๒๕๓๗

เลขทะเบียน ๐๐๓๔๘๗

เลขเรียกหนังสือ ๐๗๑๘๗ ๙๔

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมไฟฟ้า

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จอแสดงผลทางอิเล็กทรอนิกส์

(Electronic Display)

โดย

นายสรพงษ์	อุนนาภิรักษ์	34107404
นายสุรสิทธิ์	กิจมงคลชัย	34108458
นายเหนือพงศ์	วิบูลย์เมธ	34108472

อาจารย์ที่ปรึกษา

อาจารย์ปราโมทย์ วาดเขียน

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2537

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จอแสดงผลทางอิเล็กทรอนิกส์

Electronic Display

โดย นายสรพงษ์ อุนนาภิรักษ์
นายสุรสิทธ์ กิจมงคลชัย
นายเหนือพงศ์ วิบูลย์เมธ

อาจารย์ที่ปรึกษา อาจารย์ ปราโมทย์ วาคเขียน

บทคัดย่อโครงงานนี้เป็นการออกแบบและสร้างจอแสดงผลทางอิเล็กทรอนิกส์พร้อมทั้งส่วนที่ใช้ควบคุมการทำงาน จอแสดงผลนี้ประกอบด้วยดอทเมตริกซ์แอลอีดีขนาด 14 แถว 50 หลัก ซึ่งมีการทำงานโดยใช้เทคนิคการสแกนตามแนวหลักเพื่อประหยัดกระแสไฟฟ้าและจำนวนอุปกรณ์ที่ใช้ในการแลตช์ข้อมูล เมื่อเทียบกับการสแกนตามแถวโดยจะทำงานทีละ 2 หลักพร้อมกันเพื่อให้แอลอีดีมีความสว่างเพียงพอเนื่องจาก มีช่วงเวลาในการแสดงผลของแต่ละหลักต่อครั้งของการสแกนยาวกว่าเทคนิคการสแกนตามแนวหลักทีละหลัก และมีขนาดของข้อมูลที่ต้องส่งต่อครั้งไม่มากเกินไป ส่วนควบคุมจะใช้ไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ 8031 เป็นตัวประมวลผล การใช้ไมโครคอนโทรลเลอร์ควบคุมจะทำให้มีความยืดหยุ่นและความหลากหลายในรูปแบบที่แสดงออกมาเนื่องจากสามารถเปลี่ยนแปลงรูปแบบของการแสดงผลได้โดยการโปรแกรม

Abstract

This project emphasize on the design and development of electronic display and its controller. This display which consists of 14-row and 50-column dot matrix LEDs uses this technique that scans 2 collumns simultaneously per one time. The advantages of this technique is that each column than in 1-column scan techmique, and that size of data to be send per time is not too large. The controller part can process by using 8031-microcontroller in MCS51-family. Using microcontroller makes displaying flexible and various, because it can be commanded to change display pattern by program.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2537

ภาควิชา วิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง จอแสดงผลทางอิเล็กทรอนิกส์

ผู้จัดทำ

1. นายสรพงษ์	อุนนาภิรักษ์	34107404
2. นายสุรสิทธิ์	กิจมงคลชัย	34108458
3. นายเหนือพงศ์	วิบูลย์เมธ	34108472

..... อาจารย์ที่ปรึกษา

(อาจารย์ ปราโมทย์ วาดเขียน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

บทที่ 1 บทนำ	1
บทที่ 2 การทำงานของจอแสดงผลทางอิเล็กทรอนิกส์	2
บทที่ 3 ซอฟท์แวร์	10
บทที่ 4 การทดสอบชิ้นงานต้นแบบ	16
สรุป	19
แนวทางการพัฒนาต่อไป	20
บทเทรก ก. โปรแกรมควบคุมจอแสดงผลทางอิเล็กทรอนิกส์	21
บทเทรก ข. MC14017 Decade Counter	22
บทเทรก ค. MCS-51 Microcontroller	23
บทเทรก ง. 27128 Erasable PROM	24
บทเทรก จ. 62256 SRAM	25
บทเทรก ฉ. LTP-2157A Dot Matrix LED	26
บทเทรก ช. BD140 Transister	27
บทเทรก ซ. MJE270 Darlington Transister	28
บทเทรก ฌ. 74LS373,374 Octal Transparent Latch	29
บทเทรก ญ. 74LS32 Quad Two-Input OR Gate	30
บทเทรก ฏ. 4081 Quad Two-Input AND Gate	31
เอกสารอ้างอิง	32
กิตติกรรมประกาศ	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

เนื่องจากการแข่งขันทางธุรกิจในปัจจุบัน ทำให้เกิดการตื่นตัวในการเผยแพร่ข่าวสารผ่านสื่อต่างๆ จากผู้ผลิตสู่ผู้บริโภค จอแสดงผลทางอิเล็กทรอนิกส์ก็เป็นทางเลือกอย่างหนึ่งซึ่งมีผู้สนใจนำมาใช้ในช่วงไม่กี่ปีผ่านมา และยังมีแนวโน้มว่าจะเป็นที่นิยมมากขึ้นเรื่อยๆ เนื่องจากเป็นการแสดงข่าวสารผ่านสื่อขนาดใหญ่ที่มีสีสรร และสามารถเปลี่ยนแปลงรูปแบบการแสดงผลได้ตลอดเวลา ไม่ว่าจะเป็นภาพกราฟฟิกหรือตัวอักษร อีกทั้งยังเป็นตัวแทนของความทันสมัยของผู้ผลิตที่นำสื่อชนิดนี้มาใช้อีกด้วย เนื้อหาของโครงงานนี้ เป็นการปูพื้นฐานในการสร้างและออกแบบจอแสดงผลทางอิเล็กทรอนิกส์ทั้งในส่วนของฮาร์ดแวร์และซอฟท์แวร์ ซึ่งจะเป็นต้นแบบที่จะนำไปพัฒนาและประยุกต์ใช้งานต่อไปในอนาคต

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

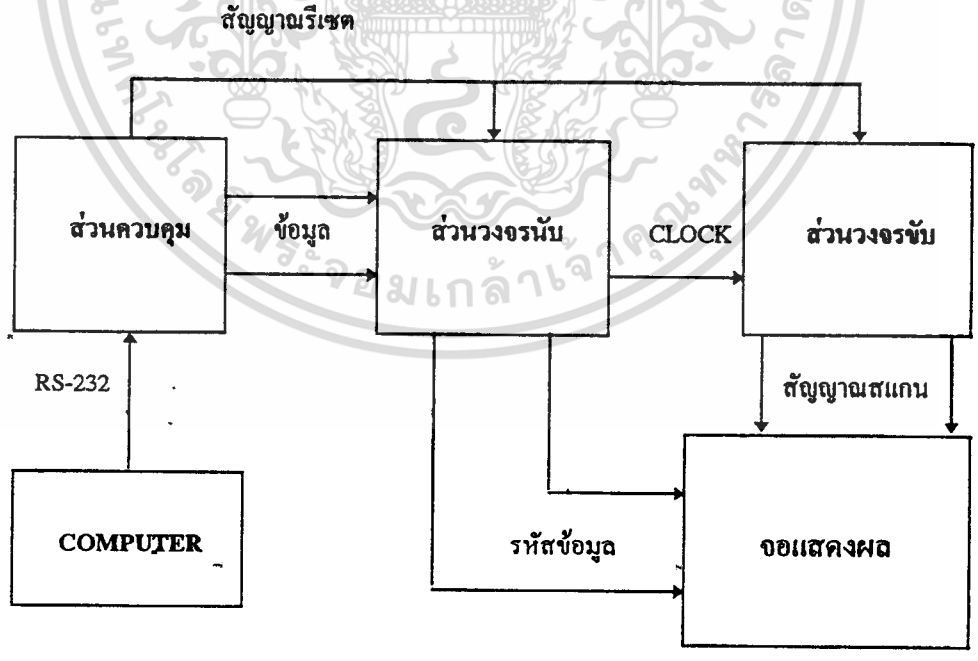
บทที่ 2
การทำงานของจอแสดงผลทางอิเล็กทรอนิกส์

การทำงานของจอแสดงผลทางอิเล็กทรอนิกส์ สามารถแบ่งได้เป็น 2 ส่วน คือ

1. ฮาร์ดแวร์
2. ซอฟท์แวร์

ฮาร์ดแวร์

ปีนวงจรส่วนที่ใช้ควบคุมการขับกระแส เพื่อป้อนให้แอลอีดีของจอแสดงผลติดสว่างตามข้อมูลที่ป้อนเข้ามา และควบคุมการเชื่อมต่อกับพอร์ทอนุกรม RS-232 ซึ่งในส่วนฮาร์ดแวร์นี้ จะแยกได้เป็น 3 ส่วน คือ ส่วนวงจรนับ ส่วนวงจรขับ และส่วนควบคุมการแสดงผล ดังแสดงไว้ในรูปที่ 2.1



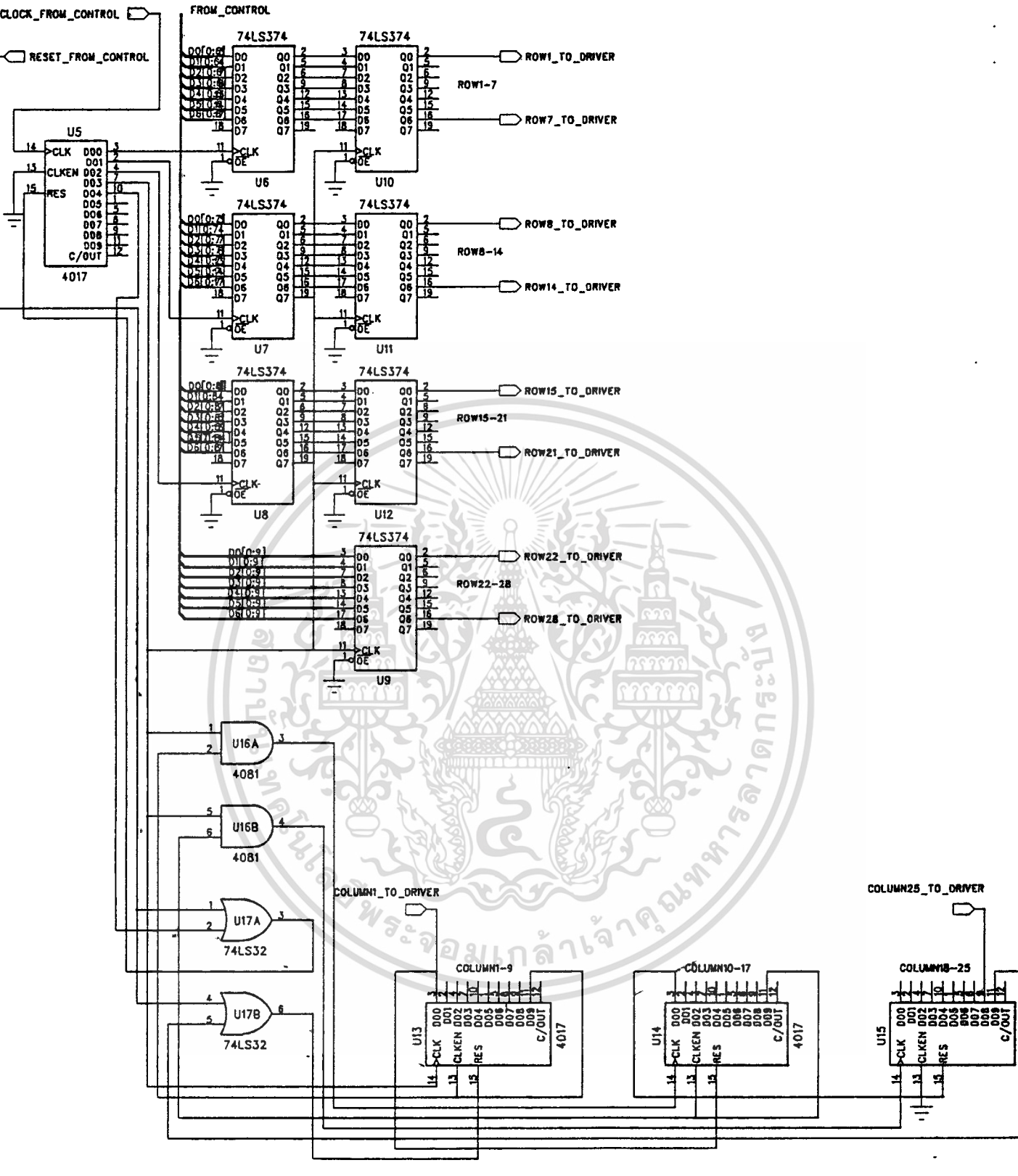
รูปที่ 2.1 บล็อกไดอะแกรมของส่วนฮาร์ดแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1. ส่วนวงจรนับ

จอแสดงผลทางอิเล็กทรอนิกส์ ประกอบด้วยแผงแสดงผลแอลอีดี ซึ่งเป็นแบบคอมมอนคาโธด ขนาด 7 แถว 5 หลัก จำนวน 20 แผง มาต่อรวมกัน โดยจะแบ่งเป็น 14 แถว 50 หลัก ซึ่งจะได้หลอดแอลอีดีทั้งหมดจำนวน 700 ดวง และเนื่องจากการที่จะทำให้หลอดแอลอีดีของจอแสดงผลติดสว่างพร้อมกันทุกดวงตลอดเวลา จะทำให้สิ้นเปลืองกระแสมาก (จากการคำนวน แอลอีดีแต่ละดวงจะกินไฟประมาณ 20 mA การที่จะทำให้แอลอีดีทั้งหมด 700 ดวงติด ต้องใช้กระแสถึง 14 A) ดังนั้นการขับกระแสจึงต้องใช้เทคนิคการสแกนเข้าช่วย ในที่นี้เลือกใช้การสแกนตามแนวหลัก (จากการทดสอบจากวงจรต้นแบบที่ออกแบบขึ้นจะใช้กระแสเพียง 3 A เท่านั้นสำหรับการเร่งความสว่างสูงสุด) และเพื่อทำให้ประหยัดตัวแลตช์ข้อมูล ซึ่งในกรณีนี้จะใช้กับข้อมูลเพียง 14 ตัว (สำหรับการสแกนตามแนวหลัก ใช้ข้อมูลป้อนเข้า 14 แถวพร้อมกันต่อการสแกน 1 ครั้ง) ในขณะที่ถ้าเลือกใช้การสแกนตามแนวแถวจะต้องป้อนข้อมูลเข้าพร้อมกันถึง 50 ตัวต่อการสแกนในแต่ละครั้ง ซึ่งก็หมายถึงจำนวนของตัวแลตช์ข้อมูลที่ต้องใช้เพิ่มขึ้นด้วย นอกจากนี้ ยังทำให้ง่ายในการออกแบบวงจรขับ โดยทรานซิสเตอร์จะโหลดกระแสต่ำกว่าวิธีการสแกนตามแนวแถวหลายเท่าตัว สำหรับวิธีการสแกนตามแนวหลักนั้น การสแกนจะเลื่อนให้แอลอีดีติดทีละหลัก (หลักละ 14 ดวง) โดยให้แอลอีดีของจอแสดงผลแต่ละหลักกระพริบด้วยความถี่สูง (50 เฮิร์ตขึ้นไป) ซึ่งจะอยู่ในระดับที่สายตาไม่สามารถแยกออกได้ว่าเกิดการกระพริบอยู่ และจากการทดลอง การที่ทำให้เกิดการสแกนติดต่อกัน 50 หลัก จะทำให้คาบเวลาในการติดของหลอดแอลอีดีแต่ละดวงแคบเกินไป ทำให้ได้ความสว่างไม่มากเท่าที่ควร ดังนั้นจึงทำการแก้ไขโดยการใช้การสแกนทีละ 2 หลักไปพร้อมๆกัน คือ จากหลักที่ 1 ถึง 25 และจากหลักที่ 26 ถึง 50 เช่น ในการสแกนครั้งแรกหลักที่ 1 จะติดพร้อมกับหลักที่ 26 โดยจะมีข้อมูลคนละชุดกัน ซึ่งจะทำให้ความสว่างอยู่ในระดับที่น่าพอใจ ในกรณีนี้จะต้องเพิ่มตัวแลตช์ข้อมูลเข้าไปเพื่อให้สามารถเก็บข้อมูลได้ถึง 28 บิท (หลักละ 14 บิท) แต่อย่างไรก็ตามก็ยังคงประหยัดกว่าการใช้การสแกนตามแนวแถวอยู่มาก ส่วนของวงจรนับจะใช้ IC เบอร์ 4017 เป็นหลัก โดยมีวงจรรวมแสดงดังรูปที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



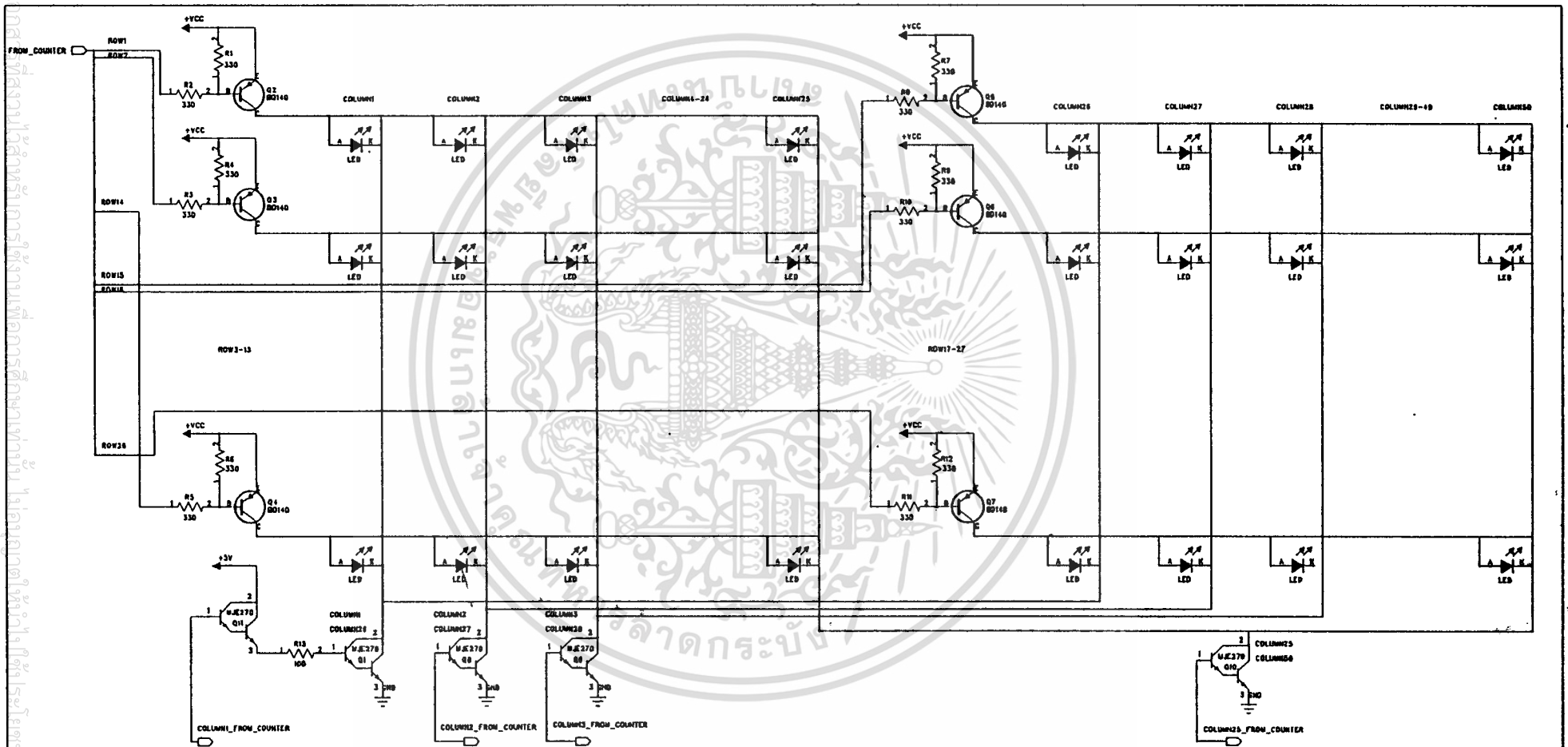
รูปที่ 2.2 แสดงวงจรของส่วนวงจรนับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2. ส่วนของวงจรขับ

จะใช้ทรานซิสเตอร์เบอร์ BD140 จำนวน 28 ตัว สำหรับการควบคุมการติดของหลอดแอลอีดีตามแนวนอน (จำนวน 14 แถว) และทรานซิสเตอร์เบอร์ MJE270 จำนวน 25 ตัว สำหรับควบคุมการสแกนในแนวหลัก (จำนวน 50 หลัก) ซึ่งในส่วนของวงจรขับนี้ ทรานซิสเตอร์ทั้ง 2 เบอร์ จะทำหน้าที่เป็นสวิตช์ในการควบคุมการติดดับของหลอดแอลอีดี ในการทำงาน BD140 จะถูกควบคุมด้วยข้อมูลที่ส่งเข้ามาและเนื่องจากแผงแสดงผลเป็นแบบคอมมอนคาโธด ดังนั้นข้อมูลทั้ง 14 แถวที่ส่งเข้ามาจะถูกเชื่อมต่อถึงกันหมดในแต่ละหลักทั้ง 25 หลัก (จำนวน 2 ชุด) และแต่ละหลักจะถูกสวิตช์โดย MJE270 เพื่อให้หลอดแอลอีดีในแต่ละหลักติดในจังหวะที่มีข้อมูลสำหรับหลักนั้นๆเข้ามา สำหรับทรานซิสเตอร์ MJE270 จะเป็นทรานซิสเตอร์แบบดาร์ลิงตัน ซึ่งสามารถขับกระแสได้สูง ทำให้สามารถรักษาระดับความสว่างของแอลอีดีแต่ละดวงให้มีค่าคงที่ แม้ว่าจำนวนการติดของแอลอีดีในแต่ละครั้งจะไม่เท่ากัน ส่วนของวงจรขับแสดงดังรูปที่ 2.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.3 แสดงวงจรของส่วนวงจรขับ

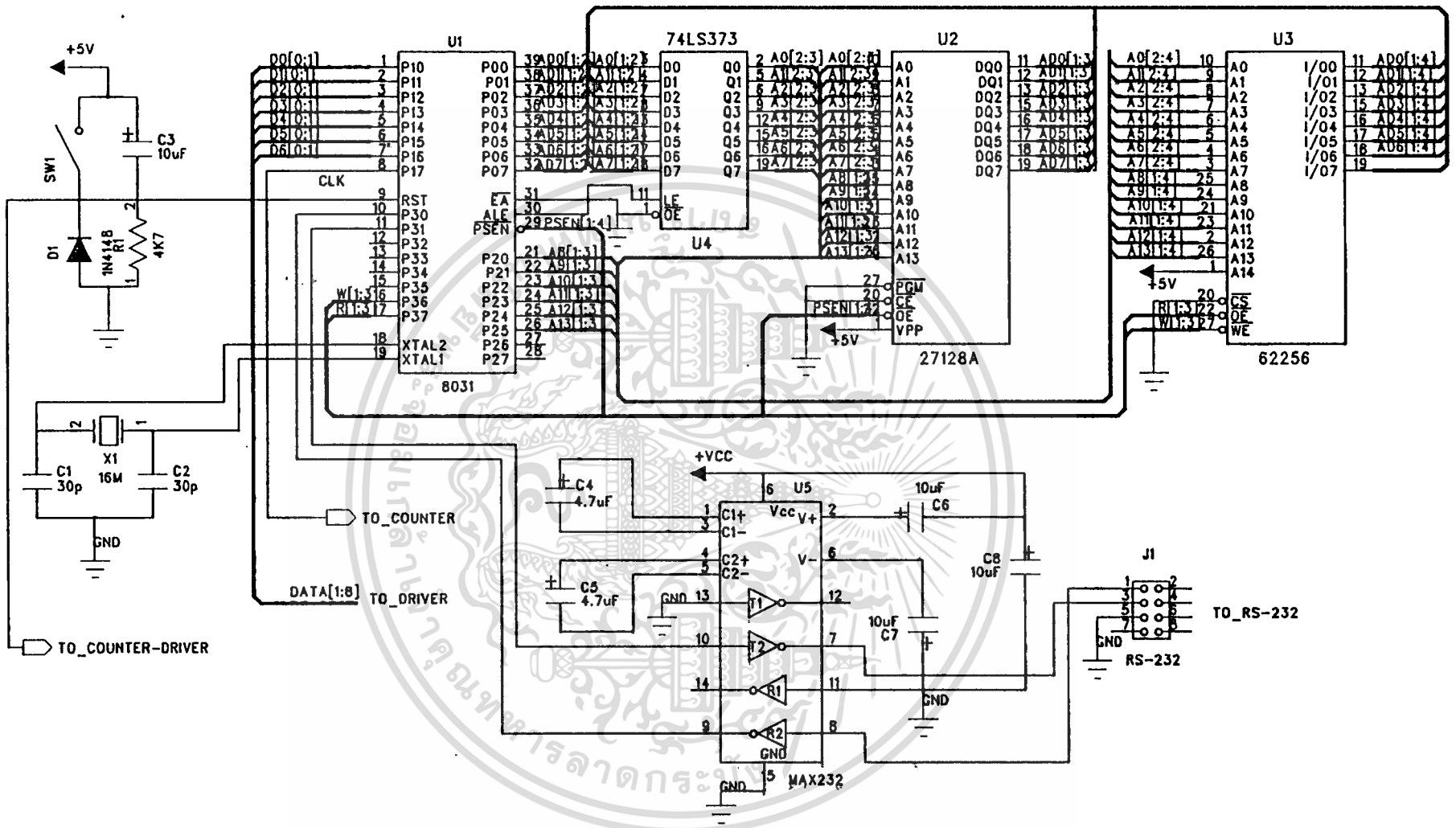
2.3. ส่วนควบคุมการแสดงผล

จะเป็นการต่อเชื่อมไมโครคอนโทรลเลอร์ 8031 เข้ากับ EPROM, RAM, ส่วนวงจรนับและส่วนวงจรขับ

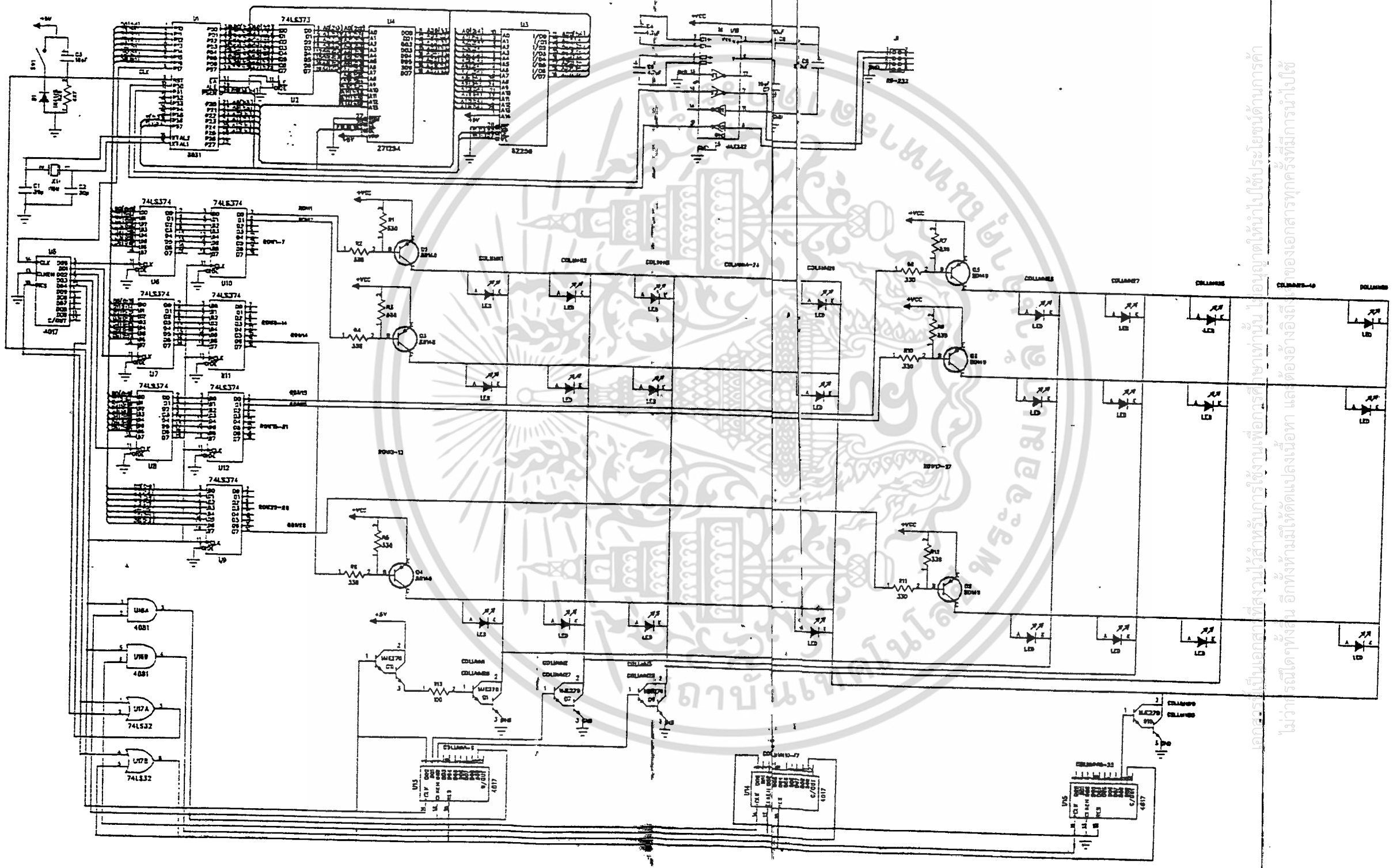
ในการทำงานส่วนของโปรแกรมและข้อมูลจะถูกเก็บไว้ใน EPROM ซึ่งจะถูกอ่านเข้ามาทำการประมวลผลที่ 8031 ข้อมูลที่ได้จะถูกอ่านออกทางพอร์ท 1 ของ 8031 ทีละ 8 บิท โดยบิทที่ 8 จะถูกเซตให้เป็น 1 ทุกครั้งที่มีการส่งข้อมูล และจะมีการส่งข้อมูลควบคุม (ข้อมูลควบคุมจะเซตบิท 8 เป็น 0) ตามไปอีก 1 ไบต์ทุกครั้ง เพื่อสร้างเป็นสัญญาณนาฬิกาให้กับส่วนวงจรนับเก็บไว้ในและส่วนอินเตอร์เฟส เพื่อให้สามารถที่จะทำการเชื่อมต่อกันได้อย่างสมบูรณ์ ข้อมูลที่ออกจาก 8031 จะถูกส่งเข้ามาตัวแลตช์ข้อมูล 74LS374 จำนวน 4 ตัว ซึ่งสัญญาณที่ใช้ในการเลือก 74LS374 แต่ละตัวจะมาจาก 4017 ที่ใช้สัญญาณนาฬิกาจากบิทที่ 8 ของข้อมูลแต่ละชุดที่ส่งมา เมื่อข้อมูลถูกเก็บลงใน 74LS374 ครบทั้ง 4 ตัวแล้ว สัญญาณการเลือกของตัวที่ 4 จะถูกส่งไปเข้าที่ขา Clock ของ 4017 ของวงจรนับและขา CP ของ 74LS374 ซึ่ง Clock ที่ส่งมานี้จะไปทำการเซตให้ตัวแลตช์ชุดที่ 2 ทำการแลตช์ข้อมูลเข้ามาและทำการส่งออกไปในเวลาเดียวกัน เพื่อให้ข้อมูลทั้งหมดถูกส่งออกไปพร้อมกันทั้ง 4 ชุด ซึ่งจะเป็นการป้องกันไม่ให้ข้อมูลที่ถูกส่งมาก่อนถูกส่งออกไปทับข้อมูลที่ถูกส่งออกไปก่อนหน้า ทำให้เกิดความผิดพลาด

ในช่วงเวลาการสแกนแต่ละครั้ง ข้อมูลทั้ง 28 บิท จะถูกส่งเข้าไปยังส่วนของวงจรขับ ในจังหวะเดียวกับที่หลักที่ 1 ถึง 25 และหลักที่ 26 ถึง 50 ถูกสแกนพร้อมๆกัน เพื่อให้การแสดงผลของข้อมูลแต่ละชุดตรงกับแต่ละหลักที่ต้องการ วงจรของส่วนควบคุมแสดงดังรูปที่ 2.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 แสดงวงจรของส่วนควบคุม



รูปที่ 2.5 แสดงวงจรรวมของจอแสดงผลทางอิเล็กทรอนิกส์

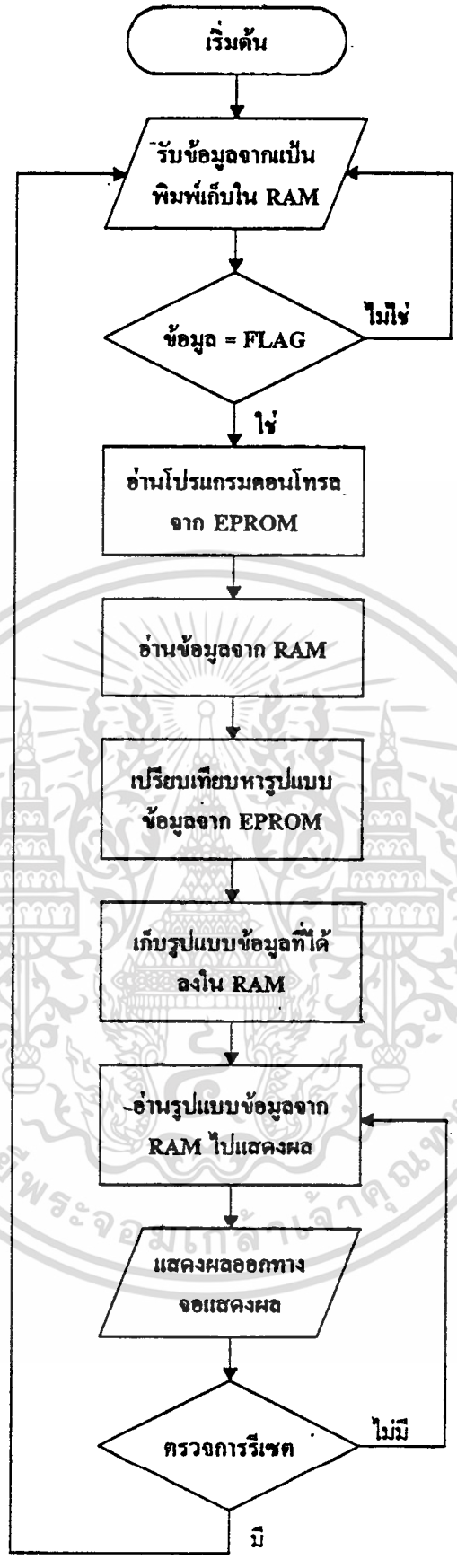
บทที่ 3

ซอฟท์แวร์

เป็นส่วนของโปรแกรมที่สั่งงานไมโครคอนโทรลเลอร์ 8031 เพื่อให้สามารถแสดงข้อมูลออกทางจอแสดงผลได้ตามที่ต้องการ โดยในขั้นแรกนี้จะแสดงผลเฉพาะตัวอักษรซึ่งแต่ละตัวจะมีขนาด 14 x 8 จุด โดยรหัสของตัวอักษรและสัญลักษณ์แต่ละตัวจะถูกเก็บไว้ใน EPROM การอ่านข้อมูลเข้ามาแสดงผล ทำได้ 2 วิธี คือ อ่านโดยตรงจากข้อมูลที่เก็บไว้ในEPROM หรือ อ่านรหัสข้อมูลผ่านพอร์ทอนุกรม RS-232 ซึ่งในกรณีหลังนี้จะเป็นการรับข้อมูลโดยตรงจากแป้นพิมพ์ของเครื่อง PC เข้ามาเก็บใน RAM ซึ่งมีการทำงานดังนี้

ขั้นตอนการทำงานของตัวโปรแกรมเริ่มจากการอ่านรหัสของตัวอักษรจาก RAM จากนั้นก็นำรหัสเหล่านี้ไปเปรียบเทียบกับรหัสที่เก็บไว้ใน EPROM เพื่อทำการหารูปแบบของข้อมูลในแต่ละหลักที่ต้องการส่งออกทางพอร์ท โดยจากโปรแกรมที่เขียนขึ้นจะเป็นการแสดงผลโดยให้ตัวอักษรวิ่งจากขวาไปซ้าย โดยเริ่มจากหลักที่ 50 แล้วจึงเลื่อนไปทางซ้ายเรื่อยๆก็จะสามารถส่งข้อมูลได้ครบทุกตัวอักษร โฟล์วชาร์ตการทำงานของส่วนซอฟท์แวร์ แสดงดังรูปที่ 3.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
034887



รูปที่ 2.1 โฟลว์ชาร์ตการทำงานของส่วนซอร์ฟแวร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1 ลักษณะตัวอักษร

ตัวอักษรที่ใช้ มีขนาด 10x8 ดอท โดยแบ่งเป็น 2 ส่วน ส่วนบนและล่าง แต่ละส่วนมีขนาด 5x8 ดอท

ตัวอักษรทุกตัวจะมีรหัสข้อมูลประจำตัว สำหรับตัวอักษรที่ใช้ในการทดสอบจะมีค่ารหัสดังนี้

A...Z = "1...26"

0...9 = "27...36"

" " = "37"

" . " = "38"

" , " = "39"

" : " = "40"

3.2 หน่วยความจำ

เมื่อการรีเซตเริ่มขึ้น 8031 จะรอรับรหัสของตัวอักษรที่ส่งมาจาก PC นำรหัสที่รับได้ไปเก็บไว้ใน RAM จากนั้นนำค่ารหัสนั้นไปคำนวนประมวลผล หาค่าพอยเตอร์สำหรับใช้ชี้แอดเดรสของ EPROM นำข้อมูลที่แสดงถึงรหัสที่ได้นั้นมาเก็บไว้ใน RAM เพื่อให้ 8031 อ่านข้อมูลส่งต่อไปยังตัวแลดช์ หน่วยความจำที่ใช้ในวงจร ประกอบไปด้วยหน่วยความจำ 2 ชนิด ดังนี้

3.2.1 สแตติกแรม

สแตติกแรมเป็นส่วนที่ใช้สำหรับพักข้อมูล เพื่อให้ 8031 ใช้ในการอ่านข้อมูลส่งออกไปแสดงผล

ลักษณะของการจัดการหน่วยความจำสแตติกแรมเป็นดังนี้

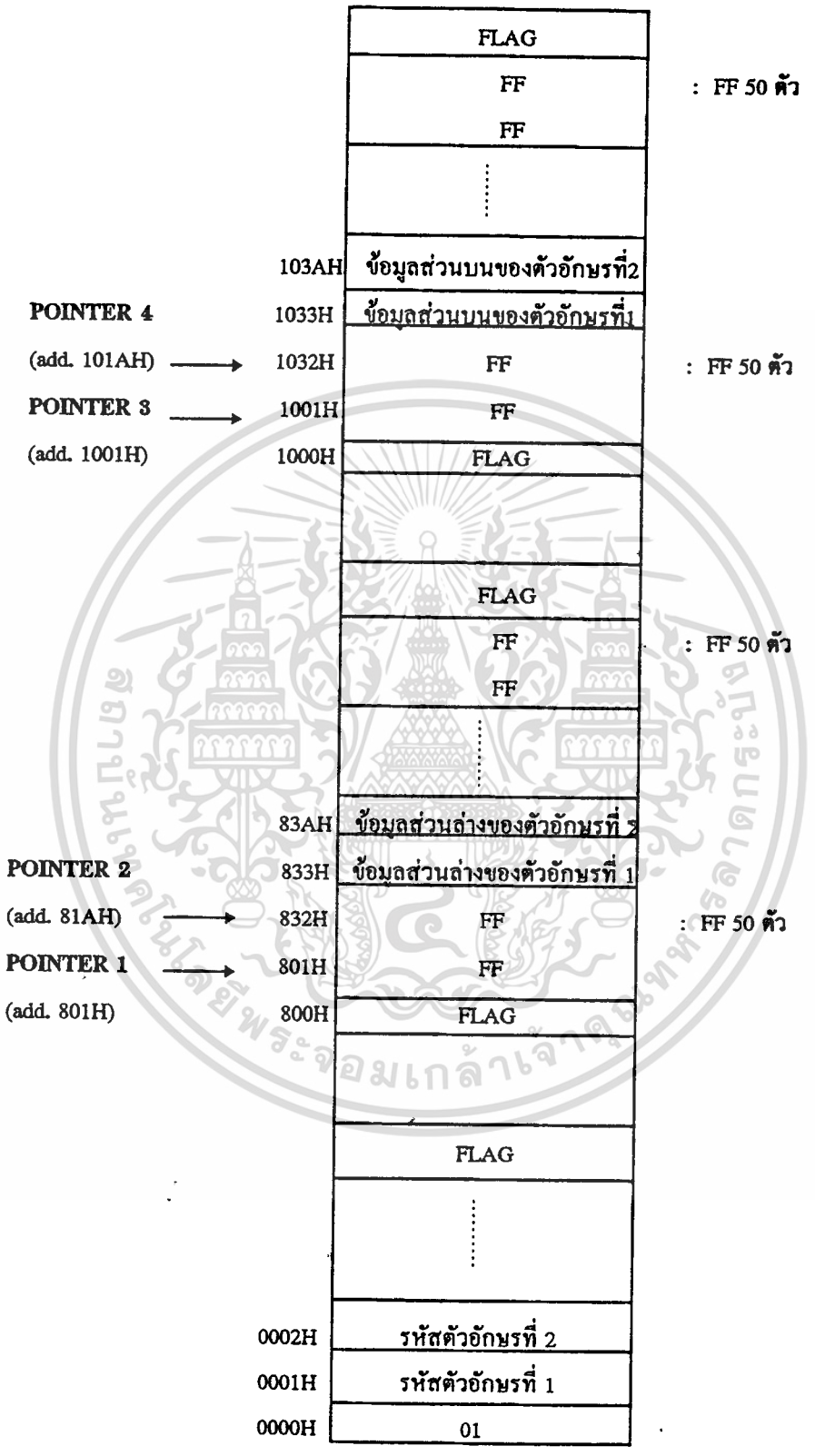
ส่วนแรก แอดเดรสที่ 00-FFH เป็นส่วนที่เก็บค่ารหัสตัวอักษรที่ PC ส่งมาให้ รหัสตัวสุดท้ายจะเป็น FLAG ซึ่งเป็นตัวใช้บอกการสิ้นสุดของข้อมูล

ส่วนที่สอง แอดเดรสที่ 800-0FFFH เป็นส่วนที่เก็บข้อมูลส่วนล่างของตัวอักษรทุกตัวซึ่งเรียงต่อกันตามลำดับรหัสตัวอักษรที่ได้รับมาจาก PC และปิดท้ายด้วย FLAG

ส่วนที่สาม แอดเดรสที่ 1000-17FFH เป็นส่วนที่เก็ บข้อมูลส่วนบนของตัวอักษรทุกตัวซึ่งเรียงต่อกันตามลำดับรหัสตัวอักษรที่ได้รับมาจาก PC และปิดท้ายด้วย FLAG

สำหรับส่วนที่สองและสามที่แอดเดรส 50 แอดเดรสแรกและ 50 แอดเดรสสุดท้าย จะเก็บค่า FFH ไว้ ลักษณะการจัดหน่วยความจำจะเป็นดังรูป 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 แสดงลักษณะการจัดการหน่วยความจำสแตติกแรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 EPROM

EPROM เป็นส่วนที่ใช้เก็บข้อมูลตัวอักษร เพื่อใช้สำหรับเป็นข้อมูลอ้างอิงรูปแบบตัวอักษร

ลักษณะการจัดการหน่วยความจำ EPROM

ส่วนแรก แอดเดรสที่ 0000-1007H เป็นส่วนโปรแกรมคอนโทรล 8031 ซึ่งเป็นโปรแกรมที่ใช้เลื่อนตัวอักษรที่แสดงบนจอแสดงผลจากขวาไปซ้าย

ส่วนที่สอง แอดเดรสที่ 1008H-2007H เป็นส่วนที่ใช้เก็บรูปแบบส่วนบนของตัวอักษร

ส่วนที่สาม แอดเดรสที่ 2008H-3007H เป็นส่วนที่ใช้เก็บรูปแบบส่วนล่างของตัวอักษร

รูปที่ 3.3 แสดงการจัดการหน่วยความจำ EPROM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การอ่าน RAM

โปรแกรมที่เก็บไว้ในแอดเดรส 0000-1007H นั้น ได้ถูกเขียนขึ้นเพื่อนำข้อมูลไปแสดงที่จอแสดงผลนั้น จะมีพอยเตอร์ 4 ตัว สำหรับ BANK 4 BANK ใช้ชี้แอดเดรสของ RAM ดังแสดงไว้ในรูป 2.2 สำหรับการสแกน 1 หลัก จะอ่าน RAM ทั้งหมด 4 ครั้ง เพื่อไปเก็บไว้ในตัวแลตช์ 4 ตัว เมื่อจะอ่าน RAM ครั้งต่อไปเพื่อสแกนหลักต่อไป จะต้องทำการเพิ่มค่าพอยเตอร์เหล่านั้นอีก 1 ค่า เป็นเช่นนี้เรื่อยๆไป

การอ่าน RAM จะสิ้นสุดเมื่อพอยเตอร์ที่ใช้ชี้ RAM นั้นชี้แอดเดรส RAM ที่เก็บ FLAG ซึ่งมีค่า 00 และใช้สำหรับเป็นรหัสบอกการสิ้นสุดของข้อมูล

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดสอบชิ้นงานต้นแบบ

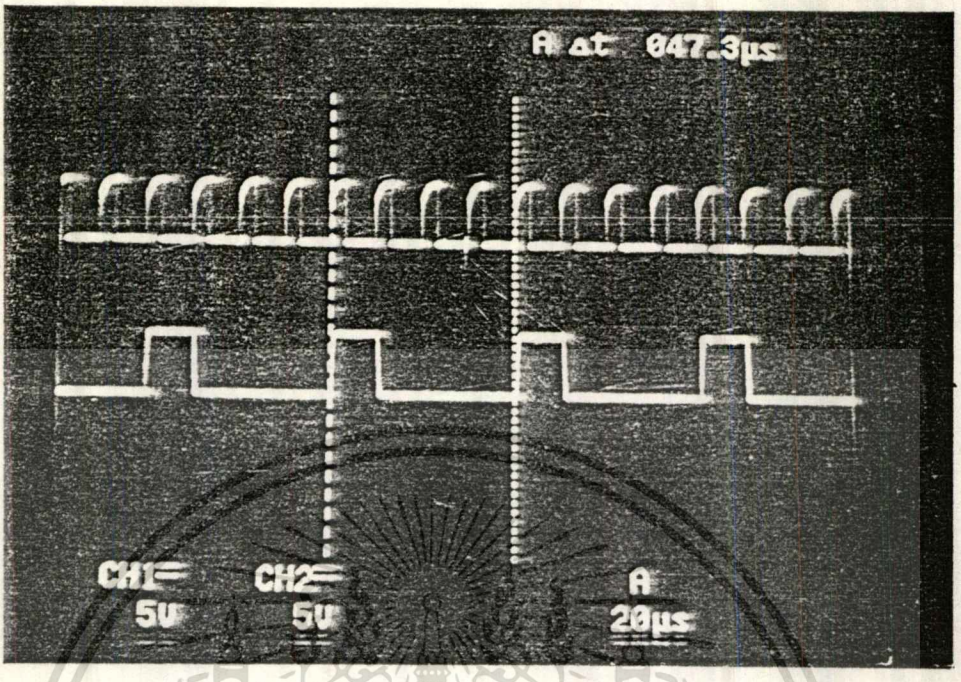
สำหรับชิ้นงานต้นแบบที่ได้ออกแบบสร้างขึ้นนี้ ได้ทำขึ้นด้วยวัตถุประสงค์ที่จะได้ชิ้นงานขนาดเล็ก สร้างง่าย (ใช้อุปกรณ์พื้นฐานที่สามารถหาซื้อได้ง่ายและราคาถูก)กินกระแสต่ำและสามารถพัฒนาขีดความสามารถในการทำงานให้มีประสิทธิภาพสูงขึ้นโดยสามารถปรับปรุงแก้ไขในส่วนของซอฟท์แวร์

ในการออกแบบลายวงจรในส่วนของฮาร์ดแวร์ ได้มีการนำซอฟท์แวร์ PADS PERFORM มาใช้ในการออกแบบ เนื่องจากเป็นซอฟท์แวร์ที่มีประสิทธิภาพสูง จึงทำให้ได้ขนาดชิ้นงานที่เล็กมาก นอกจากนี้ ยังมีการใช้เทคนิคการสแกนทางหลักทีละ 2 หลัก ซึ่งทำให้ช่วยลดจำนวนอุปกรณ์ให้น้อยลงอย่างมากและที่สำคัญคือทำให้วงจรใช้กระแสในการทำงานต่ำมาก (ประมาณ 1 A สำหรับการแสดงผลเป็นตัวอักษรในรูปแบบปกติ)

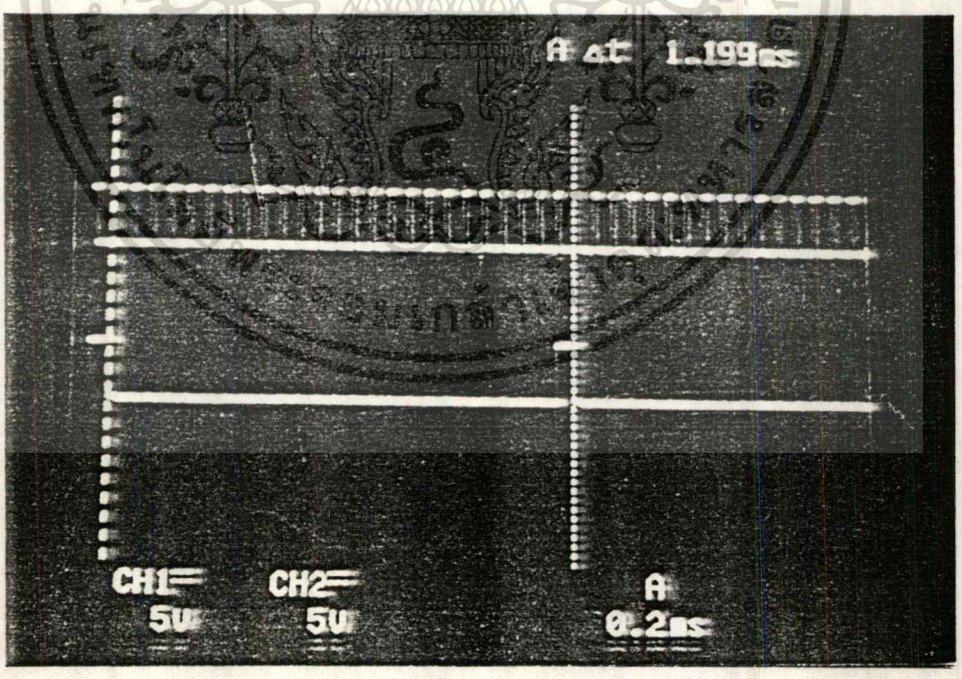
สำหรับในส่วนของซอฟท์แวร์ ได้ทำการออกแบบให้สามารถทำการเชื่อมต่อเข้ากับพอร์ทอนุกรม RS-232 ซึ่งทำให้สามารถทำการรับข้อมูลได้โดยตรงจากแป้นพิมพ์ของเครื่อง PC IBM COMPATIBLE จึงเกิดความคล่องตัวอย่างมากในการนำไปใช้งานและสามารถนำไปพัฒนาเทคนิคใหม่ๆเพิ่มขึ้นได้ง่าย

จากการทดสอบชิ้นงานต้นแบบสามารถทำงานได้ตามที่คำนวนไว้ (ใช้ความถี่ในการกระพริบของแอลอีดีตั้งแต่ 50 เฮิร์ตขึ้นไป) และจากการทดสอบถ้าใช้ความถี่สูงเกินไปมากๆจะทำให้เกิดความผิดพลาดในการแสดงผล เนื่องจากทรานซิสเตอร์ที่ใช้ในการสแกนตามแนวหลัก (MJE270) จะไม่สามารถทำการสวิตช์การตัดกระแสได้ทัน และถ้าหากใช้ความถี่ต่ำเกินไป ก็จะทำให้ตัวอักษรที่ทำการแสดงผลเกิดการพริ้วขึ้นได้ ทั้งนี้จากการทดลองป้อนความถี่ให้สูงขึ้นประมาณ 10 เท่าของค่าที่คำนวนไว้ (ประมาณ 500 เฮิร์ต) วงจรก็ยังสามารถทำงานได้อย่างมีประสิทธิภาพ และสามารถจับสัญญาณออกมาได้อย่างสมบูรณ์ ดังรูปที่ 4.1, 4.2 และ 4.3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

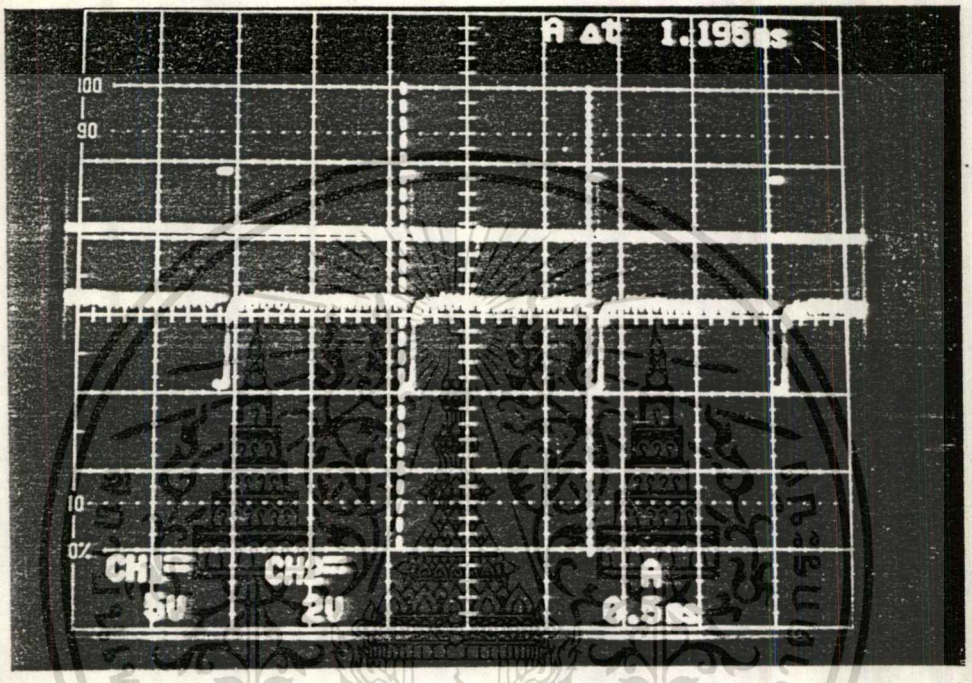


รูปที่ 4.1 แสดงการเปรียบเทียบสัญญาณระหว่าง CLOCK ที่เข้ามากับวงจรเลือกตัวแลตช์ข้อมูล (ใน CLOCK 4 ลูก จะมีการส่งข้อมูล 1 ชุด)



รูป 4.2 แสดงการเปรียบเทียบสัญญาณระหว่างการส่งข้อมูลแต่ละชุดของวงจรเลือกตัวแลตช์ข้อมูลกับสัญญาณการสแกนตามแนวหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูป 4.3 แสดงการเปรียบเทียบสัญญาณระหว่างสัญญาณการสแกนตามแนวหลัก แต่ละครั้งกับการสวิตช์ของทรานซิสเตอร์ MJE 270

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุป

จอแสดงผลทางอิเล็กทรอนิกส์ซึ่งใช้คอดเมตริกซ์แอลอีดีขนาด 14 แถว 50 หลัก สามารถแสดงได้ทั้งรูปภาพและตัวอักษร การควบคุมทำได้โดยการโปรแกรมตัวไมโครคอนโทรเลอร์ 8031 ซึ่งอยู่ในส่วนควบคุมการแสดงผล จากเทคนิคการสแกนแบบ 2 หลักพร้อมกันที่นำมาใช้ทำให้แอลอีดีแต่ละดวงมีระดับความสว่างพอดี ภาพที่ปรากฏไม่สั่นและที่สำคัญคือจะประหยัดกระแสที่ใช้ได้มากเมื่อเทียบกับการจ่ายไฟโดยไม่ใช้เทคนิคนี้

สำหรับตัวโปรแกรมนี้สามารถแสดงผลเป็นตัวอักษรได้โดยสมบูรณ์แล้ว ซึ่งสามารถที่จะนำไปพัฒนาต่อเพื่อเพิ่มประสิทธิภาพให้ดียิ่งขึ้น

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนวทางการพัฒนาต่อไป

การแสดงผลของจอแสดงผลทางอิเล็กทรอนิกส์ สามารถเปลี่ยนแปลงรูปแบบได้โดยอาศัยการโปรแกรมตัวไมโครคอนโทรลเลอร์ซึ่งเป็นส่วนควบคุม แม้ว่าจะมีความยืดหยุ่นในการใช้งานมาก แต่การเปลี่ยนแปลงแก้ไขโปรแกรมแต่ละครั้งอาจเป็นการยุ่งยาก เนื่องจากต้องทำการล้างและบันทึกโปรแกรมในตัว EPROM ดังนั้นเพื่อให้การเปลี่ยนแปลงทำได้ง่ายขึ้นควรจะมีส่วนที่สามารถรับข้อมูลจากผู้ใช้เพื่อให้สามารถแสดงรูปแบบต่างๆตามที่ผู้ใช้ต้องการได้ทันที

การรับข้อมูลจากผู้ใช้รูปแบบหนึ่งที่น่าจะเป็นประโยชน์คือการใช้การรับส่งข้อมูลระยะไกลโดยผ่านทางคลื่นวิทยุหรือแสงอินฟาเรด อันจะนำมาซึ่งความสะดวก รวดเร็วในการควบคุมจอแสดงผลนี้

นอกจากนี้การนำดอตเมตริกซ์แอลอีดีแบบ 3 สีมาใช้ จะทำให้จอแสดงผลมีความน่าสนใจต่อผู้พบเห็นมากยิ่งขึ้น รวมถึงการขยายขนาดของจอแสดงผลจะมีส่วนช่วยดึงดูดความสนใจได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*************
;receive.asm *
;for eprom   *
;*************

CPU     "8051.TBL"
HOF     "INT8"
;
;main of filling ram
;
ORG     0000H
SP:     EQU     081H
B:      EQU     0F0H
P0:     EQU     080H
P1:     EQU     090H
P2:     EQU     0A0H
P3:     EQU     0B0H
DPL:    EQU     082H
DPH:    EQU     083H
TMOD:   EQU     089H
TCON:   EQU     088H
SCON:   EQU     098H
SBUF:   EQU     099H
PSW:    EQU     0D0H
PCON:   EQU     087H
R0:     EQU     000H
R1:     EQU     001H
R2:     EQU     002H
R3:     EQU     003H
R4:     EQU     004H
R5:     EQU     005H
R6:     EQU     006H
R7:     EQU     007H
R8:     EQU     008H
R9:     EQU     009H
R10:    EQU     00AH
R11:    EQU     00BH
R12:    EQU     00CH
R13:    EQU     00DH
R14:    EQU     00EH
R15:    EQU     00FH
R16:    EQU     010H
R17:    EQU     011H
R18:    EQU     012H
R19:    EQU     013H
R20:    EQU     014H
A:      EQU     0E0H
TL0:    EQU     08AH
TL1:    EQU     08BH
TH0:    EQU     08CH
TH1:    EQU     08DH
IE:     EQU     0A8H
IP:     EQU     0B8H
TR1:    EQU     08EH
TI:     EQU     099H
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
RI:       EQU      098H
BANK3_4:SETL      001H
FLAG:    SETL      000H
BANK1_2:SETL      000H
PSW5:    SETL      0D5H
DPH_BANK1:    EQU   01H
DPH_BANK3:    EQU   02H
DPL_B1_2:     EQU   03H
DPL_B3_4:     EQU   04H
DPH_BANK2:    EQU   05H
DPH_BANK4:    EQU   06H
DELAY_2_SEC:  EQU   09H
COLUMN25:     EQU   12H
P1.8:         EQU   0E7H
;************************
;* control serial port  *
;************************

        MOV       DPTR,#0
BEGIN:  MOV       SCON,#01010010b ;
        MOV       TMOD,#00100000b ; look for 8051
        MOV       PCON,#00000000b ;  data book
        MOV       TCON,#01000000b ;
        MOV       IE,#00000000b   ;
        MOV       TH1,#152   ; define baudrate = 150 baute
        JNB       RI,$

REVDATA: MOV      A,SBUF             ; receive data
        MOVX      @DPTR,A            ;from computer
        INC       DPTR               ;to keep at RAM
        CJNE    A,#FLAG,BEGIN

MAIN:   LCALL FILLRAM
        LCALL     INIT_H.W
        MOV       DPTR,#0
        MOVX      A,@DPTR
        ANL       A,#0FH
        CJNE      A,#1,CONT2

LOP:    LCALL     MODE1   ;STILL
        LJMP LOP
CONT2:  LCALL MODE2       ;<<<===========
        LJMP      CONT2

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

        ;****************************
        ;subrotine INITIAL HARDWARE *
        ;****************************
INIT_H.W:MOV      R2,#95
 AGAIN:  MOV      A,R3
         CLR      0E7H
         MOV      P1, A
         MOV      R1,#040H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DJNZ    R1,$
        SETB    097H
        MOV     R1,#040H
        DJNZ    R1,$
        DJNZ    R2,AGAIN
        RET

        ;**********************
        ;subroutine FILL RAM *
        ;**********************
        ;fill ram for easy to define frequency
        ; of each dot
FILLRAM:MOV     A,      #FLAG   ;begin of char
        MOV     DPTR,   #1000H
        MOVX    @DPTR,  A
        MOV     DPTR,   #800H
        MOVX    @DPTR,  A
        ;bank 1&2
        MOV     R1,     #10H
        MOV     R2,     #01
        LCALL   F80
        ;
        MOV     R7,     #0      ;USE TO CHK IF BANK 1&2 OR 3
        MOV     R1,     #10H
        MOV     R2,     #51
        MOV     R0,     #1
        LCALL   START

        LCALL   F80
        MOV     A,      DPH     ;save address for mode 2
        MOV     R11,    A
        MOV     A,      DPL
        MOV     R13,    A
        CLR     0D7H
        SUBB    A,      #24
        MOV     R14,    A
        MOV     A,      #FLAG   ;
        MOVX    @DPTR,  A
        ;
        ;bank 3&4
        MOV     R1,     #08
        MOV     R2,     #01
        LCALL   F80
        ;
        MOV     R7,     #1      ;USE TO CHK IF MODE 3&4
        MOV     R1,     #08
        MOV     R2,     #51
        MOV     R0,     #1
        LCALL   START
        ;
        LCALL   F80
        MOV     A,      DPH     ;save address for mode 2
        MOV     R12,    A
        MOV     A,      #FLAG
        MOVX    @DPTR,  A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RET

        ;*******
        ; F80  *
        ;*******
        ;fill #ff 50 time
F80:    MOV     R3,     #32H    ; 50 times
        MOV     A,      R1
        MOV     DPH,    A
        MOV     A,      R2
        MOV     DPL,    A
FILL:   MOV     A,      #0FFH
        MOVX    @DPTR,  A
        INC     DPTR
        DJNZ    R3,     FILL
        RET

        ;******************
        ;subroutine START;*
        ;******************
        ;read data from EPROM add. 2008h and 1008h
        ;to fill RAm at add.800h,add1000h
        ;

START:  MOV     R4,     #8      ;1 code = 8 subcode
        MOV     B,      #8
        MOV     DPH,    #0
        MOV     A,      R0
        MOV     DPL,    A
        MOVX    A,      @DPTR
        MUL     AB
        MOV     DPL,    A
        MOV     A,      B
        MOV     DPH,    A
        ;chk bank
        CJNE    R7,     #BANK3_4,     R7ZERO
        MOV     A,      #20H
R7ZERO: CJNE    R7,     #BANK1_2,     GO_ON
        MOV     A,      #10H

GO_ON:  ADD     A,      DPH
        MOV     DPH,    A
        MOV     A,      DPH     ;R5&R6 are
        MOV     R5,     A       ;readed rom pointer
        MOV     A,      DPL     ;
        MOV     R6,     A
        ;
R4TEST: MOV     A,      #0
        MOVC    A,      @A+DPTR
        PUSH    A
        INC     DPTR
        MOV     A,      DPH
        MOV     R5,     A
        MOV     A,      DPL
        MOV     R6,     A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ;
        MOV     A,      R1      ;R1&R2 are
        MOV     DPH,    A       ;writed ram
        MOV     A,      R2      ;pointer
        MOV     DPL,    A
        POP     A
        MOVX    @DPTR,  A
        INC     DPTR
        ;
        MOV     A,      DPH     ;push writed ram
        MOV     R1,     A       ; dptr
        MOV     A,      DPL
        MOV     R2,     A
        ;
        MOV     A,      R5
        MOV     DPH,    A
        MOV     A,      R6
        MOV     DPL,    A
        DJNZ    R4,     R4TEST
        ;
        INC     R0
        MOV     A,      R0
        MOV     DPL,    A
        MOV     DPH,    #0      ;chk if code ff mean
        MOVX    A,      @DPTR   ;end of char
        CJNE    A,      #FLAG,     START   ;RETURN TO MAIN I
        RET
        ;
        ;end of filling ram
        ;
    ;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

        ;********************
        ; subroutine MODE2 *
        ;********************
        ;read data from RAM add.800h and add 1000h
        ;to display

        ;   <<<<<<<<------------
MODE2:  MOV     DPH_BANK1,#10H
        MOV     DPH_BANK2,#08
        MOV     DPL_B1_2,#01
        ;--------------------
        MOV     DPL_B3_4,#25
        MOV     DPH_BANK3,#10H
        MOV     DPH_BANK4,#08

VWPRINT3:       MOV     R0,#1           ;DELAY1_1
VWPRINT2:       MOV     COLUMN25,#25
                PUSH    DPH_BANK1
                PUSH    DPH_BANK2
                PUSH    DPL_B1_2
                PUSH    DPL_B3_4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            PUSH    DPH_BANK3
            PUSH    DPH_BANK4
            ;
PRINT2:     LCALL   BANK
            LCALL   INC_DPTR
            DJNZ    COLUMN25,PRINT2
            ;
            POP     DPH_BANK4
            POP     DPH_BANK3
            POP     DPL_B3_4
            POP     DPL_B1_2
            POP     DPH_BANK2
            POP     DPH_BANK1

            DJNZ    R0,VWPRINT2
            ;
            LCALL   INC_DPTR
            ;
            LCALL   DETECT_FLAG
            CJNE    A,#FLAG,GO
            SJMP    GORET
GO:         LJMP    VWPRINT3
GORET:      RET

            ;*************************
            ;subroutine DETECT-FLAG *
            ;*************************
            ;flag mean end of data to send
            ; out at display

DETECT_FLAG:MOV     A,#24
            CLR     0D7H
            ADD     A,DPL_B3_4
            MOV     DPL,A
            MOV     A,#0
            ADDC    A,DPH_BANK1
            MOV     DPH,A
            MOVX    A,@DPTR
            RET


            ;******************
            ;subroutine bank **
            ;******************
            ;there are 4 bank at the display board
            ; so you must send data to each bank

BANK:       ;bank 1
            MOV     A,DPH_BANK1
            MOV     DPH,A
            MOV     A,DPL_B1_2
            MOV     DPL,A
            MOVX    A,@DPTR
            CLR     0E7H    ;clear a.8
            MOV     P1,A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     R20,#010h;DELAY
        DJNZ    R20,$
        SETB    097H    ;set p1.8
        DJNZ    R20,$

        ;bank 3
        MOV     A,DPH_BANK2
        MOV     DPH,A
        MOV     A,DPL_B1_2
        MOV     DPL,A
        MOVX    A,@DPTR
        CLR     0E7H    ;clear a.8
        MOV     P1,A
        MOV     R20,#010h;DELAY
        DJNZ    R20,$
        SETB    097H    ;set p1.8
        MOV     R20,#020H
        DJNZ    R20,$

        ;bank 4
        MOV     A,DPH_BANK3
        MOV     DPH,A
        MOV     A,DPL_B3_4
        MOV     DPL,A
        MOVX    A,@DPTR
        CLR     0E7H    ;clear a.8
        MOV     P1,A
        MOV     R20,#010h;DELAY
        DJNZ    R20,$
        SETB    097H    ;set p1.8
        MOV     R20,#020H
        DJNZ    R20,$

        ;bank 1
        MOV     A,DPH_BANK4
        MOV     DPH,A
        MOV     A,DPL_B3_4
        MOV     DPL,A
        MOVX    A,@DPTR
        CLR     0E7H    ;clear a.8
        MOV     P1,A
        MOV     R20,#010h;DELAY
        DJNZ    R20,$
        SETB    097H    ;set p1.8
        MOV     R20,#020H
        DJNZ    R20,$
        RET

        ;********************
        ;subroutine increl *
        ;********************

INC_DPTR: INC   DPL_B1_2
        MOV     R3,DPL_B1_2
        CJNE    R3,#0,NOCY1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                INC     DPH_BANK1
                INC     DPH_BANK2
                ;
        NOCY1:  INC     DPL_B3_4
                MOV     R4,DPL_B3_4
                CJNE    R4,#0,NOCY2
                INC     DPH_BANK3
                INC     DPH_BANK4
        NOCY2:  RET

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
                ;******************
                ;subroutine mode1 *
                ;******************

        MODE1:  MOV     DPH_BANK1,#10H
                MOV     DPH_BANK2,#08
                MOV     DPL_B1_2,#48
                ;------------------
                MOV     DPL_B3_4,#72
                MOV     DPH_BANK3,#10H
                MOV     DPH_BANK4,#08
                ;------------------
ALL_25COLUMN:   MOV     COLUMN25,#25
                PUSH    DPH_BANK1
                PUSH    DPH_BANK2
                PUSH    DPL_B1_2
                PUSH    DPL_B3_4
                PUSH    DPH_BANK3
                PUSH    DPH_BANK4
                ;
   CONT_PRINT:  LCALL   BANK
                LCALL   INC_DPTR
                DJNZ    COLUMN25,CONT_PRINT
                ;
                POP     DPH_BANK4
                POP     DPH_BANK3
                POP     DPL_B3_4
                POP     DPL_B1_2
                POP     DPH_BANK2
                POP     DPH_BANK1                                          ;

                LJMP    ALL_25COLUMN
                ;
       GORET2:  RET


;**********************************
                ; code data for each character
                ; keep at eprom add. 1008h and 2008h

                ORG     1008H
     FILLROM:   DFB     08FH,0F7H,0FBH,0FBH,0FBH,0F7H,08FH,0FFH  ;A 1
                DFB     083H,0BBH,0BBH,0BBH,0BBH,0C7H,0FFH,0FFH  ;B 2
                DFB     08FH,0F7H,0FBH,0FBH,0FBH,0F7H,0EFH,0FFH  ;C 3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DFB     083H,0FBH,0FBH,0FBH,0F7H,0EFH,09FH,0FFH ;D 4
DFB     083H,0BBH,0BBH,0BBH,0BBH,0FBH,0F3H,0FFH ;E 5
DFB     083H,0BBH,0BBH,0BBH,0BBH,0FBH,0F3H,0FFH ;F 6
DFB     08FH,0F7H,0FBH,0FBH,0FBH,0F7H,0EFH,0FFH ;G 7
DFB     083H,0BFH,0BFH,0BFH,0BFH,0BFH,083H,0FFH ;H 8
DFB     0FFH,0FBH,0FBH,083H,0FBH,0FBH,0FFH,0FFH ;I 9
DFB     0FFH,0FFH,0FBH,0FBH,083H,0FBH,0FFH,0FFH ;J 1
DFB     083H,0BFH,0DFH,0EFH,0F7H,0FBH,0FFH,0FFH ;K 1
DFB     083H,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH ;L 1
DFB     083H,0EFH,0DFH,0BFH,0DFH,0EFH,083H,0FFH ;M 1
DFB     083H,0EFH,0DFH,0BFH,0FFH,0FFH,083H,0FFH ;N 1
DFB     08FH,077H,07BH,07BH,07BH,077H,08FH,0FFH ;0 1
DFB     083H,0BBH,0FBH,0FBH,0FBH,0FBH,087H,0FFH ;P 1
DFB     08FH,0F7H,0FBH,0FBH,0FBH,0F7H,08FH,0FFH ;Q 1
DFB     083H,0BBH,0FBH,0FBH,0FBH,0FBH,087H,0FFH ;R 1
DFB     0EFH,0D7H,0BBH,0BBH,0BBH,0F7H,0EFH,0FFH ;S 1
DFB     0FBH,0FBH,0FBH,083H,0FBH,0FBH,0F3H,0FFH ;T 2
DFB     083H,0FFH,0FFH,0FFH,0FFH,0FFH,083H,0FFH ;U 2
DFB     083H,0FFH,0FFH,0FFH,0FFH,0FFH,083H,0FFH ;V 2
DFB     083H,0FFH,0FFH,0FFH,0FFH,0FFH,083H,0FFH ;W 2
DFB     0F3H,0EFH,0DFH,0BFH,0DFH,0EFH,0F3H,0FFH ;X 2
DFB     0E3H,0DFH,0BFH,0FFH,0BFH,0DFH,0E3H,0FFH ;Y 2
DFB     0F3H,0FBH,0DBH,0BBH,0BBH,0DBH,0E3H,0FFH ;Z 2
DFB     08FH,0F7H,0FBH,0FBH,0BBH,0D7H,08FH,0FFH ;0 2
DFB     0FFH,0EFH,0F7H,083H,0FFH,0FFH,0FFH,0FFH ;1 2
DFB     0CFH,0F7H,0FBH,0FBH,0FBH,0B7H,0CFH,0FFH ;2 2
DFB     0CFH,0F7H,0FBH,0FBH,0FBH,0B7H,0CFH,0FFH ;3 3
DFB     083H,0BFH,0BFH,0BFH,0BFH,0BFH,083H,0FFH ;4 3
DFB     083H,0DBH,0EBH,0EBH,0EBH,0DBH,0BBH,0FFH ;5 3
DFB     08FH,0F7H,0BBH,0BBH,0BBH,0F7H,0EFH,0FFH ;6 3
DFB     0FBH,0FBH,0FBH,0BBH,0DBH,0EBH,0F3H,0FFH ;7 3
DFB     0CFH,0B7H,0BBH,0BBH,0BBH,0B7H,0CFH,0FFH ;8 3
DFB     0CFH,0B7H,0FBH,0FBH,0FBH,0B7H,08FH,0FFH ;9 3
DFB     0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH ;  3
DFB     0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH ;. 3
```

```
ORG     2008H
DFB     0E0H,0FEH,0FEH,0FEH,0FEH,0FEH,0E0H,0FFH ;A
DFB     0E0H,0EFH,0EFH,0EFH,0EFH,0EEH,0F1H,0FFH ;B
DFB     0F8H,0F7H,0EFH,0EFH,0EFH,0F7H,0F9H,0FFH ;C
DFB     0E0H,0EFH,0EFH,0EFH,0EFH,0F7H,0F8H,0FFH ;D
DFB     0E0H,0EFH,0EFH,0EFH,0EFH,0EFH,0E7H,0FFH ;E
DFB     0E0H,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH ;F
DFB     0F8H,0F7H,0EFH,0EEH,0EEH,0F6H,0F8H,0FFH ;G
DFB     0E0H,0FFH,0FFH,0FFH,0FFH,0FFH,0E0H,0FFH ;H
DFB     0FFH,0EFH,0EFH,0E0H,0EFH,0EFH,0FFH,0FFH ;I
DFB     0F3H,0EFH,0EFH,0EFH,0F0H,0FFH,0FFH,0FFH ;J
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DFB     0E0H,0FFH,0FEH,0FDH,0FBH,0F7H,0EFH,0FFH ;K
DFB     0E0H,0EFH,0EFH,0EFH,0EFH,0EFH,0E7H,0FFH ;L
DFB     0E0H,0FFH,0FFH,0FFH,0FFH,0FFH,0E0H,0FFH ;M
DFB     0E0H,0FFH,0FFH,0FFH,0FEH,0FDH,0E0H,0FFH ;N
DFB     0F8H,0F7H,0EFH,0EFH,0EFH,0F7H,0F8H,0FFH ;O
DFB     0E0H,0FEH,0FEH,0FEH,0FEH,0FEH,0FFH,0FFH ;P
DFB     0F8H,0F7H,0EFH,0EFH,0EBH,0F7H,0E8H,0FFH ;Q
DFB     0E0H,0FEH,0FEH,0FCH,0FAH,0F6H,0EFH,0FFH ;R
DFB     0F3H,0EFH,0EFH,0EFH,0EFH,0F6H,0F9H,0FFH ;S
DFB     0FFH,0FFH,0FFH,0E0H,0FFH,0FFH,0FFH,0FFH ;T
DFB     0F8H,0F7H,0EFH,0EFH,0EFH,0F7H,0F8H,0FFH ;U
DFB     0FCH,0FBH,0F7H,0EFH,0F7H,0FBH,0FCH,0FFH ;V
DFB     0E0H,0FBH,0FDH,0FEH,0FDH,0FBH,0E0H,0FFH ;W
DFB     0E3H,0FDH,0FEH,0FFH,0FEH,0FDH,0E3H,0FFH ;X
DFB     0FFH,0FFH,0FFH,0E0H,0FFH,0FFH,0FFH,0FFH ;Y
DFB     0E3H,0EDH,0EEH,0EEH,0EDH,0EFH,0E7H,0FFH ;Z
DFB     0F8H,0F3H,0EDH,0EEH,0EFH,0F7H,0F8H,0FFH ;0
DFB     0FFH,0EFH,0EFH,0E0H,0EFH,0EFH,0FFH,0FFH ;1
DFB     0EFH,0E7H,0EBH,0EDH,0EEH,0EFH,0EFH,0FFH ;2
DFB     0F9H,0F7H,0EFH,0EFH,0EFH,0F6H,0F9H,0FFH ;3
DFB     0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0E0H,0FFH ;4
DFB     0FBH,0F7H,0EFH,0EFH,0EFH,0F7H,0F8H,0FFH ;5
DFB     0F8H,0F6H,0EFH,0EFH,0EFH,0F6H,0F9H,0FFH ;6
DFB     0E3H,0FDH,0FEH,0FFH,0FFH,0FFH,0FFH,0FFH ;7
DFB     0F9H,0F6H,0EFH,0EFH,0EFH,0F6H,0F9H,0FFH ;8
DFB     0FBH,0F7H,0EEH,0EEH,0EEH,0F7H,0F8H,0FFH ;9
DFB     0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH,0FFH ;
DFB     0FFH,0FFH,0E7H,0E7H,0FFH,0FFH,0FFH,0FFH ;:
DFB     0FFH,0FFH,0F7H,0E7H,0FFH,0FFH,0FFH,0FFH ;;
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*****************************************/
/* file  program.c for receive character */
/* from keyboard then transform to code  */
/* and send it to 8031 microcontroller   */
/*****************************************/
#include <stdio.h>
#include <process.h>
#include <dos.h>
#include <conio.h>
#include <stdlib.h>
#define FLAG 0
#define mode2 1
#define mode1 0
void saved_file(char *,int);
void send_file();
void receivebyte(char *);
void symbol(char *,char *);
void del(int *,char *);

main()
    {
    char SYMBOL[50];
    int MODE,i;
    char ch[80];
    clrscr() ;
    system("mode com1:150,n,8,1");
    printf("\n\n\n\n\n");
    printf("ENTER YOUR MASSAGE TO DISPLAY => ' ");
    receivebyte(ch);
    symbol(SYMBOL,ch);
    saved_file(SYMBOL,++i);
    send_file();
    }


/**** FUNCTION RECEIVE BYTE FROM KEYBOARD  ****/

void receivebyte(char *ch)
{
int c,i;

i=0;
do
    {
get: ch[i]=toupper(getch());
    c=ch[i];
    if ((c == 0)||(ch[0]=='\r'))
        {
        printf ("\007");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    if ((c>64 && c<91 )||(c>47 && c< 58)||(c==' ')||(c=='.'))
        {
        printf("%c",c);
        i++;
        }
    if (c=='\b')
        {
        del(&i,ch);
        goto get;
        }
    }while(c!='\r');
    ch[i]='\0';
    printf(" '\n");
}


/**** Funtion symbol: transform character to code , example: A,a=cod

void symbol(char *SYMBOL,char *ch)
{
int i,c;
i=1;
c=0;
do
    {
    if (ch[i-1]>='A' && ch[i-1]<='Z')   /*character*/
        {
        SYMBOL[i]=ch[i-1];
        SYMBOL[i]-=64;
        }
    if (ch[i-1]>47 && ch[i-1]<58) /* digit*/
        {
         SYMBOL[i]=ch[i-1];
         SYMBOL[i]-=48;
         SYMBOL[i]+=27;
         }
    if (ch[i-1]==' ')
        SYMBOL[i]=37;
    if (ch[i-1]== '.')
        SYMBOL[i]=38;
    i++;
}while (ch[i-1]);
SYMBOL[i]=FLAG; /** for detect the last char**/
printf(" %d",SYMBOL[i]);
SYMBOL[0]=MODE2;
}


/****Function del key*****/

void del(int *i,char ch[])
{
int x,y;
if(*i>0)
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      x=wherex();
      y=wherey();
      gotoxy(x-1,y);
      printf("%c",' ');
      gotoxy(x-1,y);
      ch[*i]='\0';
      *i-=1;
      }
}

/****SAVE FILE*******/
void saved_file(char SYMBOL[],int x)
{
FILE *fp;
int i,f[4];
char filename[20]="project.dat",a[20];
if ((fp = fopen(filename,"wb")) == NULL)
      {
      printf("Error in open file\n");
      exit(1);
      }
fwrite(SYMBOL,x,1,fp);
if (ferror(fp))
      {
      printf("Error in writing file\n");
      exit(1);
      }

printf(" OK! file is saved  \007\n");
printf(" Now Send file\n" );
fclose(fp);

}



/*************SEND FILE ***************/
void send_file()
#define COM1    0
/*YOU MUST DEFINE BAUD RATE 150 AT PORT COM1*/
{
char filename[20]="project.dat";
int y[16],i;
unsigned char ch;
FILE *fp;

if ((fp = fopen(filename,"rb")) == NULL)
      {
      printf("Can't Open File to Send !");
      exit(1);
      }
else
      {
      /*bioscom(0, SETTINGS, COM1);*/
      ch=getc(fp);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        printf("%d",ch);
        while(!feof(fp))
            {
        bioscom(1,ch,COM1);              /*NOW SEND EACH CHARACTER*/
            ch=getc(fp);
            printf(" %d ",ch);
            }
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข. MC14017 Decade Counter

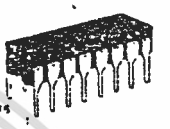
MOTOROLA

MC14017B

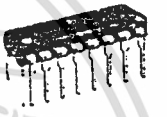
DECADE COUNTER

The MC14017B is a five-stage Johnson decade counter with built-in code converter. High speed operation and spike-free outputs are obtained by use of a Johnson decade counter design. The ten decoded outputs are normally low, and go high only at their appropriate decimal time period. The output changes occur on the positive-going edge of the clock pulse. This part can be used in frequency division applications as well as decade counter or decimal decode display applications.

- Fully Static Operation
- DC Clock Input Circuit Allows Slow Rise Times
- Carry Out Output for Cascading
- Divide-by-N Counting
- Supply Voltage Range = 3.0 Vdc to 18 Vdc
- Capable of Driving Two Low-power TTL Loads or One Low-power Schottky TTL Load Over the Rated Temperature Range
- Pin-for-Pin Replacement for CD4017B
- Triple Diode Protection on All Inputs



L SUFFIX
CERAMIC
CASE 620



P SUFFIX
PLASTIC
CASE 648

D SUFFIX
SOIC
CASE 751B

ORDERING INFORMATION

MC14XXXBCP Plastic
MC14XXXBCL Ceramic
MC14XXXBD SOIC

T_A = −55° to 125°C for all packages.

MAXIMUM RATINGS* (Voltages Referenced to V_{SS})

Symbol	Parameter	Value	Unit
V_{DD}	DC Supply Voltage	−0.5 to +18.0	V
V_{in}, V_{out}	Input or Output Voltage (DC or Transient)	−0.5 to V_{DD} +0.5	V
I_{in}, I_{out}	Input or Output Current (DC or Transient), per Pin	±10	mA
P_D	Power Dissipation, per Package†	500	mW
T_{stg}	Storage Temperature	−65 to +150	°C
T_L	Lead Temperature (8-Second Soldering)	260	°C

*Maximum Ratings are those values beyond which damage to the device may occur.
†Temperature Derating: Plastic "P and D/DW" Packages: − 7.0 mW/°C From 65°C To 125°C
Ceramic "L" Packages: − 12 mW/°C From 100°C To 125°C

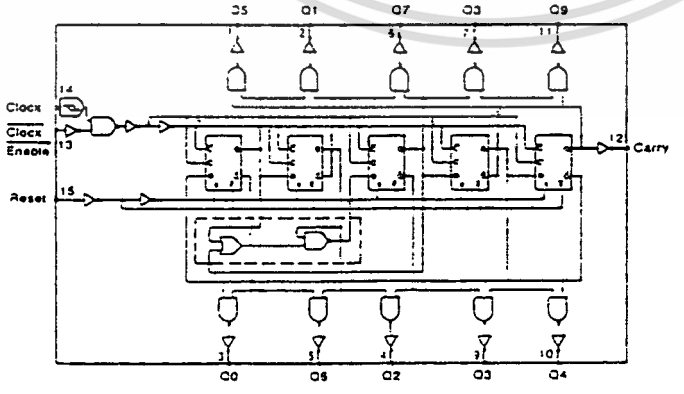
FUNCTIONAL TRUTH TABLE

(Positive Logic)

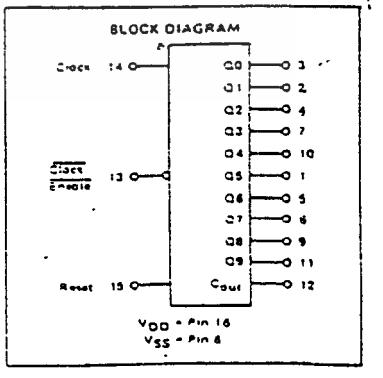
CLOCK	CLOCK ENABLE	RESET	DECODE OUTPUT n
0	X	0	n
X	1	0	n
↑	↓	0	n
0	↓	0	n+1
X	1	0	n
↑	0	0	n
1	↓	0	n+1

X = Don't Care n = 5 Carry = "1" Otherwise = "0"

LOGIC DIAGRAM



BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ELECTRICAL CHARACTERISTICS (Voltages Referenced to V_{SS})

Characteristic	Symbol	V_{DD} Vdc	−55°C Min	−55°C Max	25°C Min	25°C Typ #	25°C Max	125°C Min	125°C Max	Unit
Output Voltage "0" Level V_{in} = V_{DD} or 0	V_{OL}	5.0	—	0.05	—	0	0.05	—	0.05	Vdc
		10	—	0.05	—	0	0.05	—	0.05	
		15	—	0.05	—	0	0.05	—	0.05	
"1" Level V_{in} = 0 or V_{DD}	V_{OH}	5.0	4.95	—	4.95	5.0	—	4.95	—	Vdc
		10	9.95	—	9.95	10	—	9.95	—	
		15	14.95	—	14.95	15	—	14.95	—	
Input Voltage "0" Level (V_O = 4.5 or 0.5 Vdc)	V_{IL}	5.0	—	1.5	—	2.25	1.5	—	1.5	Vdc
(V_O = 9.0 or 1.0 Vdc)		10	—	3.0	—	4.50	3.0	—	3.0	
(V_O = 13.5 or 1.5 Vdc)		15	—	4.0	—	6.75	4.0	—	4.0	
"1" Level (V_O = 0.5 or 4.5 Vdc)	V_{IH}	5.0	3.5	—	3.5	2.75	—	3.5	—	Vdc
(V_O = 1.0 or 9.0 Vdc)		10	7.0	—	7.0	5.50	—	7.0	—	
(V_O = 1.5 or 13.5 Vdc)		15	11	—	11	8.25	—	11	—	
Output Drive Current (V_{OH} = 2.5 Vdc) Source	I_{OH}	5.0	−3.0	—	−2.4	−4.2	—	−1.7	—	mAdc
(V_{OH} = 4.6 Vdc)		5.0	−0.64	—	−0.51	−0.88	—	−0.36	—	
(V_{OH} = 9.5 Vdc)		10	−1.6	—	−1.3	−2.25	—	−0.9	—	
(V_{OH} = 13.5 Vdc)		15	−4.2	—	−3.4	−8.8	—	−2.4	—	
(V_{OL} = 0.4 Vdc) Sink	I_{OL}	5.0	0.64	—	0.51	0.88	—	0.36	—	mAdc
(V_{OL} = 0.5 Vdc)		10	1.6	—	1.3	2.25	—	0.9	—	
(V_{OL} = 1.5 Vdc)		15	4.2	—	3.4	8.8	—	2.4	—	
Input Current	I_{in}	15	—	±0.1	—	±0.00001	±0.1	—	±1.0	µAdc
Input Capacitance (V_{in} = 0)	C_{in}	—	—	—	—	5.0	7.5	—	—	pF
Quiescent Current (Per Package)	I_{DD}	5.0	—	5.0	—	0.005	5.0	—	150	µAdc
		10	—	10	—	0.010	10	—	300	
		15	—	20	—	0.015	20	—	600	
Total Supply Current**† (Dynamic plus Quiescent, Per Package) (C_L = 50 pF on all outputs, all buffers switching)	I_T	5.0	I_T = (0.27 µA/kHz) f + I_{DD}							µAdc
		10	I_T = (0.55 µA/kHz) f + I_{DD}							
		15	I_T = (0.83 µA/kHz) f + I_{DD}							

#Data labelled "Typ" is not to be used for design purposes but is intended as an indication of the IC's potential performance.

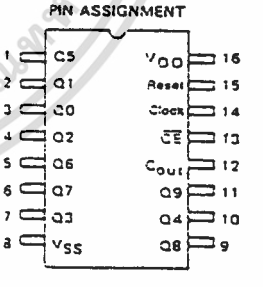
**The formulas given are for the typical characteristics only at 25°C.

†To calculate total supply current at loads other than 50 pF:

$$I_T(C_L) = I_T(50\ pF) + (C_L - 50)\ Vfk$$

where: I_T is in µA (per package), C_L in pF, V = $(V_{DD} - V_{SS})$ in volts, f in kHz is input frequency, and k = 0.0011.

PIN ASSIGNMENT



This device contains protection circuitry to guard against damage due to high static voltages or electric fields. However, precautions must be taken to avoid applications of any voltage higher than maximum rated voltages to this high-impedance circuit. For proper operation, V_{in} and V_{out} should be constrained to the range $V_{SS} \le (V_{in}$ or $V_{out}) \le V_{DD}$.

Unused inputs must always be tied to an appropriate logic voltage level (e.g., either V_{SS} or V_{DD}). Unused outputs must be left open.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

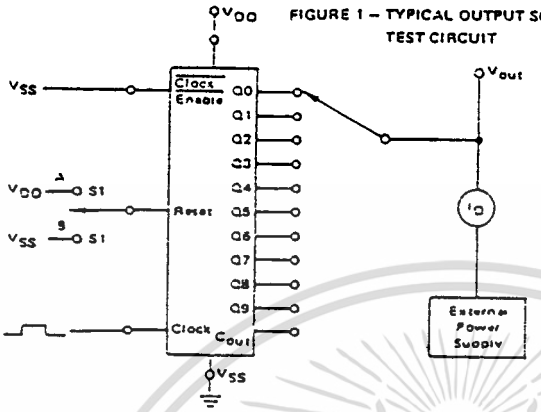
Characteristic	Symbol	V_{DD} Vdc	Min	Typ #	Max	Unit
Output Rise and Fall Time	t_{TLH}, t_{THL}					ns
t_{TLH}, t_{THL} = (1.5 ns/pF) C_L + 25 ns		5.0	—	100	200	
t_{TLH}, t_{THL} = (0.75 ns/pF) C_L + 12.5 ns		10	—	50	100	
t_{TLH}, t_{THL} = (0.55 ns/pF) C_L + 9.5 ns		15	—	40	80	
Propagation Delay Time	t_{PLH}, t_{PHL}					ns
Reset to Decade Output						
t_{PLH}, t_{PHL} = (1.7 ns/pF) C_L + 415 ns		5.0	—	500	1000	
t_{PLH}, t_{PHL} = (0.66 ns/pF) C_L + 197 ns		10	—	230	460	
t_{PLH}, t_{PHL} = (0.5 ns/pF) C_L + 150 ns		15	—	175	350	
Propagation Delay Time	t_{PLH}, t_{PHL}					ns
Clock to C_{out}						
t_{PLH}, t_{PHL} = (1.7 ns/pF) C_L + 315 ns		5.0	—	400	800	
t_{PLH}, t_{PHL} = (0.66 ns/pF) C_L + 142 ns		10	—	175	350	
t_{PLH}, t_{PHL} = (0.5 ns/pF) C_L + 100 ns		15	—	125	250	
Propagation Delay Time	t_{PLH}, t_{PHL}					ns
Clock to Decode Output						
t_{PLH}, t_{PHL} = (1.7 ns/pF) C_L + 415 ns		5.0	—	500	1000	
t_{PLH}, t_{PHL} = (0.66 ns/pF) C_L + 197 ns		10	—	230	460	
t_{PLH}, t_{PHL} = (0.5 ns/pF) C_L + 150 ns		15	—	175	350	
Turn-Off Delay Time	t_{PLH}					ns
Reset to C_{out}						
t_{PLH} = (1.7 ns/pF) C_L + 315 ns		5.0	—	400	800	
t_{PLH} = (0.66 ns/pF) C_L + 142 ns		10	—	175	350	
t_{PLH} = (0.5 ns/pF) C_L + 100 ns		15	—	125	250	
Clock Pulse Width	$t_{w(H)}$	5.0	250	125	—	ns
		10	100	50	—	
		15	75	35	—	
Clock Frequency	f_{cl}	5.0	—	5.0	2.0	MHz
		10	—	12	5.0	
		15	—	16	6.7	
Reset Pulse Width	$t_{w(H)}$	5.0	500	250	—	ns
		10	250	125	—	
		15	190	95	—	
Reset Removal Time	t_{rem}	5.0	750	375	—	ns
		10	275	135	—	
		15	210	105	—	
Clock Input Rise and Fall Time	t_{TLH}, t_{THL}	5.0	No Limit			—
		10				
		15				
Clock Enable Setup Time	t_{su}	5.0	350	175	—	ns
		10	150	75	—	
		15	115	52	—	
Clock Enable Removal Time	t_{rem}	5.0	420	250	—	ns
		10	200	100	—	
		15	140	70	—	

*The formulas given are for the typical characteristics only at 25°C.

#Data labelled "Typ" is not to be used for design purposes but is intended as an indication of the IC's potential performance.

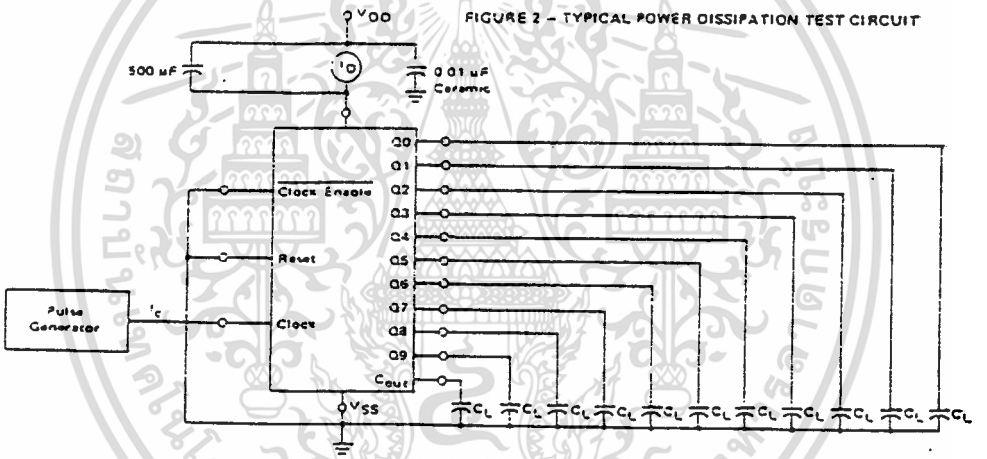
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 1 – TYPICAL OUTPUT SOURCE AND OUTPUT SINK CHARACTERISTICS TEST CIRCUIT



	OUTPUT SINK DRIVE	OUTPUT SOURCE DRIVE
DECODE OUTPUTS	(S1 to A)	Clock to desired outputs (S1 to B)
Carry	Clock to 5 thru 9 (S1 to B)	S1 to A
V_{OS} *	V_{DD}	$-V_{DD}$
V_{OS} *	V_{out}	$V_{out}-V_{DD}$

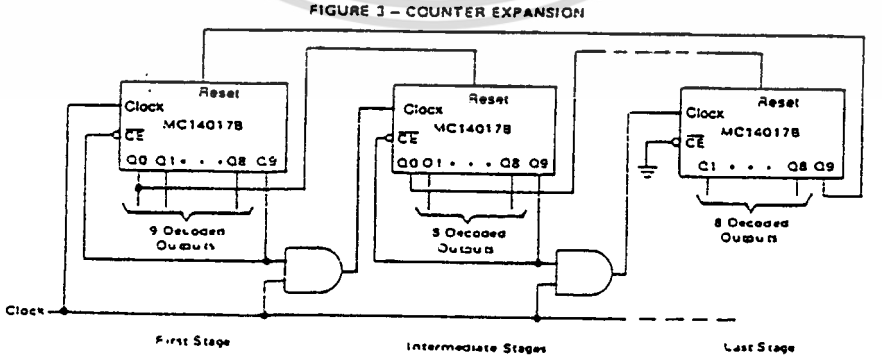
FIGURE 2 – TYPICAL POWER DISSIPATION TEST CIRCUIT



APPLICATIONS INFORMATION

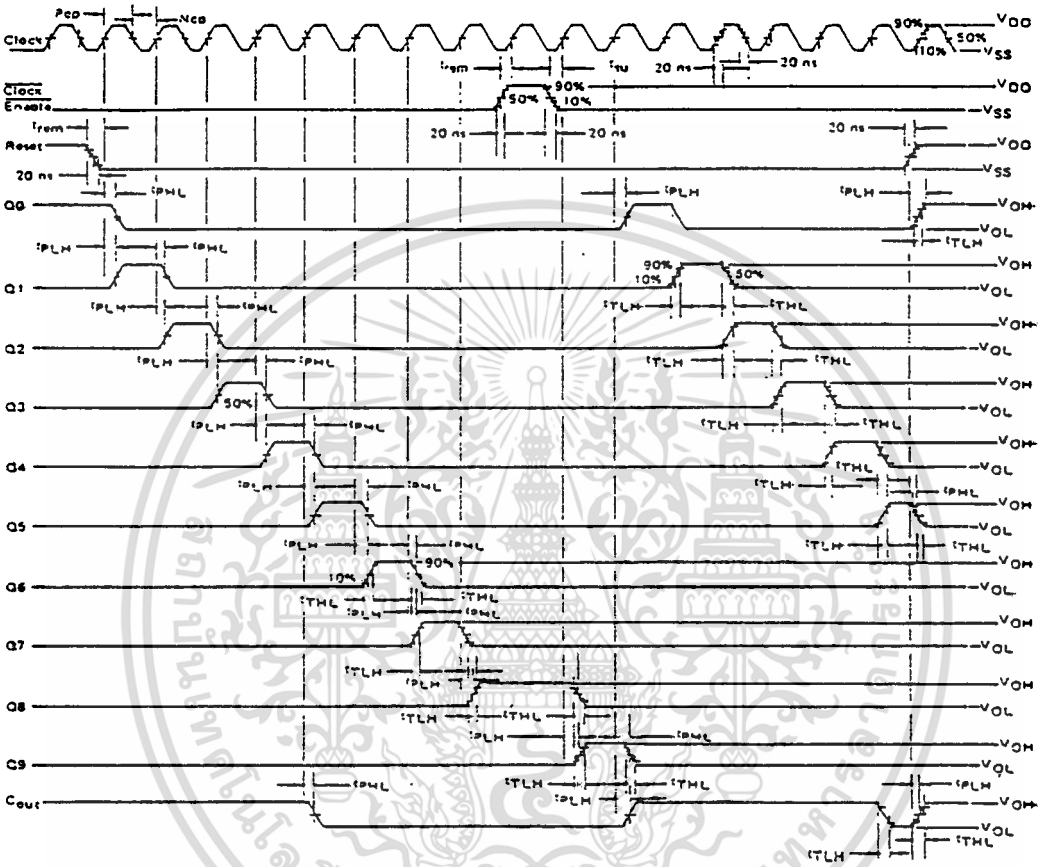
Figure 3 shows a technique for extending the number of decoded output states for the MC140178. Decoded outputs are sequential within each stage and from stage to stage, with no dead time (except propagation delay).

FIGURE 3 – COUNTER EXPANSION



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FIGURE 4 – AC MEASUREMENT DEFINITION AND FUNCTIONAL WAVEFORMS



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

intel®

MCS®-51
8-BIT CONTROL-ORIENTED MICROCOMPUTERS

8031/8051
8031AH/8051AH
8032AH/8052AH
8751H/8751H-12

- High Performance HMOS Process
- Internal Timers/Event Counters
- 2-Level Interrupt Priority Structure
- 32 I/O Lines (Four 8-Bit Ports)
- 64K Program Memory Space
- Boolean Processor
- Bit-Addressable RAM
- Programmable Full Duplex Serial Channel
- 111 Instructions (64 Single-Cycle)
- 64K Data Memory Space
- Security Feature Protects EPROM Parts Against Software Piracy

The MCS®-51 products are optimized for control applications. Byte-processing and numerical operations on small data structures are facilitated by a variety of fast addressing modes for accessing the internal RAM. The instruction set provides a convenient menu of 8-bit arithmetic instructions, including multiply and divide instructions. Extensive on-chip support is provided for one-bit variables as a separate data type, allowing direct bit manipulation and testing in control and logic systems that require Boolean processing.

Device	Internal Memory		Timers/ Event Counters	Interrupts
	Program	Data		
8052AH	8K × 8 ROM	256 × 8 RAM	3 × 16-Bit	6
8051AH	4K × 8 ROM	128 × 8 RAM	2 × 16-Bit	5
8051	4K × 8 ROM	128 × 8 RAM	2 × 16-Bit	5
8032AH	none	256 × 8 RAM	3 × 16-Bit	6
8031AH	none	128 × 8 RAM	2 × 16-Bit	5
8031	none	128 × 8 RAM	2 × 16-Bit	5
8751H	4K × 8 EPROM	128 × 8 RAM	2 × 16-Bit	5
8751H-12	4K × 8 EPROM	128 × 8 RAM	2 × 16-Bit	5

The 8751H is an EPROM version of the 8051AH; that is, the on-chip Program Memory can be electrically programmed, and can be erased by exposure to ultraviolet light. It is fully compatible with its predecessor, the 8751-8, but incorporates two new features: a Program Memory Security bit that can be used to protect the EPROM against unauthorized read-out, and a programmable baud rate modification bit (SMOD). SMOD is not present in the 8751H-12.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

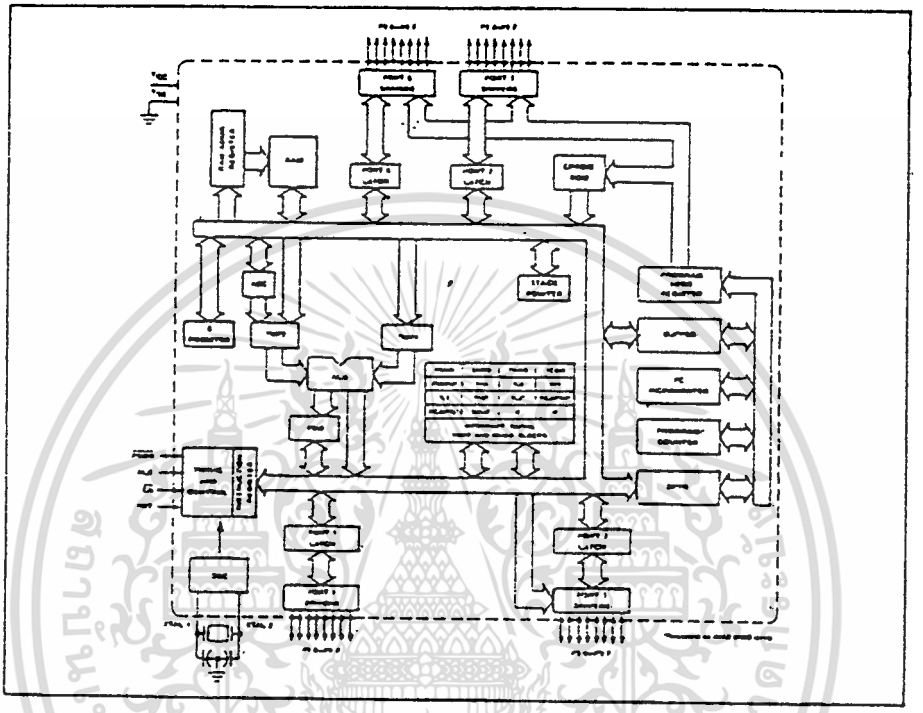


Figure 1. MCS®-51 Block Diagram

PIN DESCRIPTIONS

VCC

Supply voltage.

VSS

Circuit ground.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink 8 LS TTL inputs. Port 0 pins that have 1s written to them float, and in that state can be used as high-impedance inputs.

Port 0 is also the multiplexed low-order address and data bus during accesses to external Program and Data Memory. In this application it uses strong internal pullups when emitting 1s, and can source and sink 8 LS TTL inputs.

Port 0 also receives the code bytes during programming of the EPROM parts, and outputs the code bytes during program verification of the ROM and EPROM parts. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source 4 LS TTL inputs. Port 1 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}, on the data sheet) because of the internal pullups.

Port 1 also receives the low-order address bytes during programming of the EPROM parts and during program verification of the ROM and EPROM parts.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

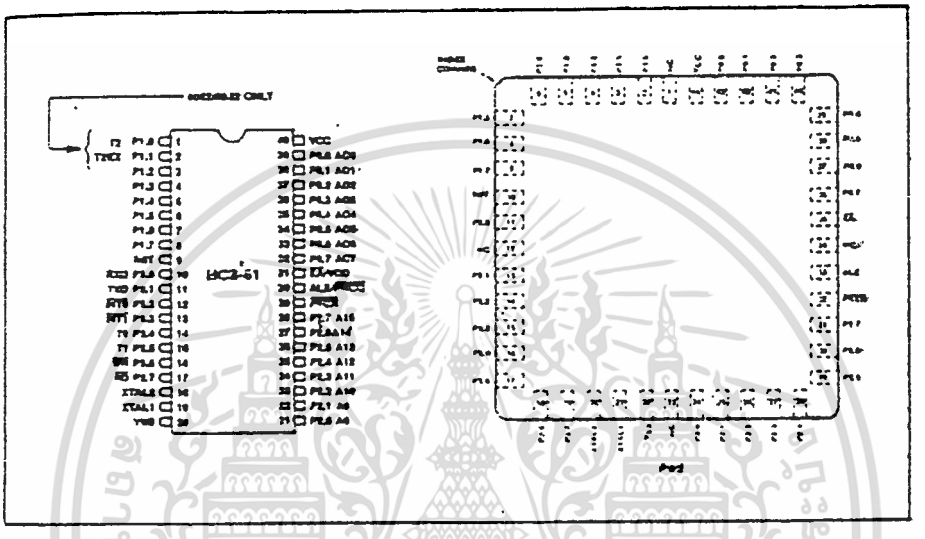


Figure 2. MCS®-51 Pin Connections

In the 8032AH and 8052AH, Port 1 pins P1.0 and P1.1 also serve the T2 and T2EX functions, respectively.

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source 4 LS TTL inputs. Port 2 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external Program Memory and during accesses to external Data Memory that use 16-bit addresses (MOVX @DPTR). In this application it uses strong internal pullups when emitting 1s. During accesses to external Data Memory that use 8-bit addresses (MOVX @RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits during programming of the EPROM parts and during program verification of the ROM and EPROM parts.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source 4 LS TTL inputs. Port 3 pins that have 1s written to them are pulled high by the internal pullups, and in that state can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (IIL, on the data sheet) because of the pullups.

Port 3 also serves the functions of various special features of the MCS-51 Family, as listed below:

Port Pin	Alternative Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (Timer 0 external input)
P3.5	T1 (Timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. ALE can drive 8 LS TTL inputs. This pin is also the program pulse input ($\overline{PROG}$) during programming of the EPROM parts.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

$\overline{PSEN}$

Program Store Enable is the read strobe to external Program Memory. $\overline{PSEN}$ can drive 8 LS TTL inputs.

When the device is executing code from external Program Memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external Data Memory.

$\overline{EA}$/VPP

External Access enable $\overline{EA}$ must be externally held low in order to enable any MCS-51 device to fetch code from external Program Memory locations 0 to 0FFFH (0 to 1FFFH, in the 8032AH and 8052AH).

Note, however, that if the Security Bit in the EPROM devices is programmed, the device will not fetch code from any location in external Program Memory.

This pin also receives the 21V programming supply voltage (VPP) during programming of the EPROM parts.

XTAL1

Input to the inverting oscillator amplifier.

XTAL2

Output from the inverting oscillator amplifier.

OSCILLATOR CHARACTERISTICS

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 3. Either a quartz crystal or ceramic resonator may be used. More detailed information concerning the use of the on-chip oscillator is available in Application Note AP-155, "Oscillators for Microcontrollers."

To drive the device from an external clock source, XTAL1 should be grounded, while XTAL2 is driven, as shown in Figure 4. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum high and low times specified on the Data Sheet must be observed.

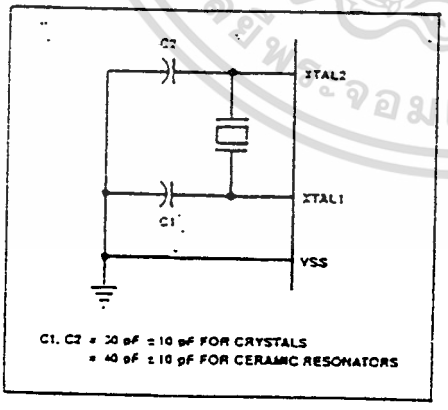


Figure 3. Oscillator Connections

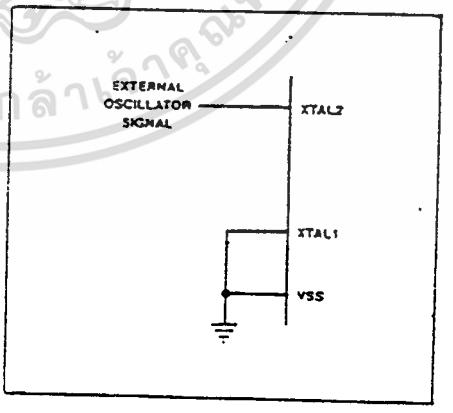


Figure 4. External Drive Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทแทรก ง. 27128 Erasable PROM

intel®

27128A
ADVANCED 128K (16K x 8) PRODUCTION AND UV ERASABLE PROMs

- Fast 200 nsec Access Time
  — HMOS* II-E Technology
- Low Power
  — 100 mA Maximum Active
  — 40 mA Maximum Standby
- Int$_e$ligent Identifier™ Mode
  — Automated Programming Operations
- Compatible with 2764A, 27128, 27256
- New Quick-Pulse Programming™ Algorithm
  — Used on Plastic DIP
  — Int$_e$ligent Programming™ Algorithm Compatible
- ±10% V_{CC} Tolerance Available
- Available in 28-Pin Cerdip and Plastic Packages
  (See Packaging Spec, Order #231369)

The Intel 27128A is a 5V only, 131,072-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The 27128A is an advanced high speed version of the 27128 and is fabricated with Intel's HMOSII-E technology which significantly reduces die size and greatly improves the device's performance, reliability and manufacturability.

The 27128A is currently available in two different package types. CERDIP packages provide flexibility in prototyping and R&D environments where reprogrammability is required. Plastic DIP EPROMs provide optimum cost effectiveness in production environments.

Intel's new Quick-Pulse Programming Algorithm enables these Plastic EPROMs to be programmed within two seconds. Programming equipment that takes advantage of this innovation will electronically identify the EPROM with the help of the int$_e$ligent Identifier and rapidly program it using a superior programming method. The int$_e$ligent Programming Algorithm may be utilized in the absence of such equipment and is used to program CERDIP devices.

The 27128A is available in fast access times including 200 ns (27128A-2). This ensures compatibility with high-performance microprocessors, such as Intel's 8 MHz 80186 allowing full speed operation without the addition of WAIT states. The 27128A is also directly compatible with the 12 MHz 8051 family. For access times down to 110 ns, refer to the 27128B data sheet.

*HMOS is a patented process of Intel Corporation.

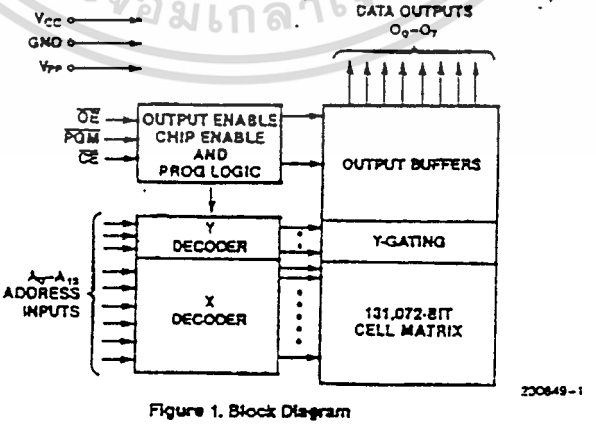


Figure 1. Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Names

A_0–A_{15}	Addresses
$\overline{CE}$	Chip Enable
$\overline{OE}/V_{PP}$	Output Enable/V_{PP}
O_0–O_7	Outputs
O.U.	Don't Use

27512	27256 27C256 87C256	27128 27128A	2764 2764A : C64- 27C64
O.U.	V_{PP}	V_{PP}	V_{PP}
A_{12}	A_{12}	A_{12}	A_{12}
A_7	A_7	A_7	A_7
A_6	A_6	A_6	A_6
A_5	A_5	A_5	A_5
A_4	A_4	A_4	A_4
A_3	A_3	A_3	A_3
A_2	A_2	A_2	A_2
A_1	A_1	A_1	A_1
A_0	A_0	A_0	A_0
O_0/O_8	O_0	O_0	O_0
O_1/O_9	O_1	O_1	O_1
O_2	O_2	O_2	O_2
Gnd	Gnd	Gnd	Gnd

231058-2

2764- 2764A 27C64 87C64	27128 27128A-	27256- 27C256- 27C256	27513
V_{CC}	V_{CC}	V_{CC}	V_{CC}
PGM	PGM	A_{14}	$\overline{PE}$
N.C.	A_{13}	A_{13}	A_{13}
A_8	A_8	A_8	A_8
A_9	A_9	A_9	A_9
A_{11}	A_{11}	A_{11}	A_{11}
$\overline{OE}$	$\overline{OE}$	$\overline{OE}$	$\overline{OE}/V_{PP}$
A_{10}	A_{10}	A_{10}	A_{10}
$\overline{CE}$ ALE/$\overline{CE}$	$\overline{CE}$	$\overline{CE}$ ALE/$\overline{CE}$	$\overline{CE}$
O_7	O_7	O_7	O_7
O_6	O_6	O_6	O_6
O_5	O_5	O_5	O_5
O_4	O_4	O_4	O_4
O_3	O_3	O_3	O_3

Figure 2. Pin Configurations

EXTENDED TEMPERATURE (EXPRESS) EPROMs

The Intel EXPRESS EPROM family is a series of electrically programmable read only memories which have received additional processing to enhance product characteristics. EXPRESS processing is available for several densities of EPROM, allowing the choice of appropriate memory size to match system applications. EXPRESS EPROM products are available with 168 ±8 hours, 125°C dynamic burn-in using Intel's standard bias configuration. This process exceeds or meets most industry specifications of burn-in. The standard EXPRESS EPROM operating temperature range is 0°C to 70°C. Extended operating temperature range (−40°C to +85°C) EXPRESS products are available. Like all Intel EPROMs, the EXPRESS EPROM family is inspected to 0.1% electrical AQL. This may allow the user to reduce or eliminate incoming inspection testing.

EXPRESS EPROM PRODUCT FAMILY

PRODUCT DEFINITIONS

Type	Operating Temperature	Burn-In 125°C (hr)
O	0°C to +70°C	168 ±8
T	−40°C to +85°C	None
L	−40°C to +85°C	168 ±8

EXPRESS OPTIONS

27512 VERSIONS

Packaging Options	
Speed Versions	Cerdip
-170V05	O
-200V05, -200V10	O, L
-STD, -25, -250V10	O, T, L
-3, -30	O, T, L

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้