

ภาควิชาครุศาสตร์วิศวกรรม
คณะครุศาสตร์อุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญานิพนธ์

ปริญญานิพนธ์ เครื่องโปรแกรมอุปกรณ์ลอจิกแบบโปรแกรมได้

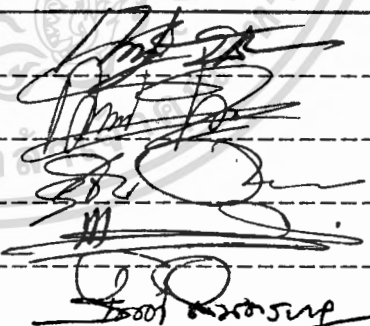
PROGRAMMABLE LOGIC DEVICE PROGRAMMER

ชื่อนักศึกษา 1. นายวิฑูร ฉายกะพันธ์ รหัสประจำตัว 37031321
2. นายสุรพงษ์ ชาสุวรรณ รหัสประจำตัว 37031327
3. นายอนุสิทธิ์ บุญมาก รหัสประจำตัว 37031330
4. นางสาวอรทัย ใจคำลือ รหัสประจำตัว 37031332

หลักสูตร ครุศาสตร์อุตสาหกรรมบัณฑิต สาขาวิชา อิเล็กทรอนิกส์และคอมพิวเตอร์

อาจารย์ผู้ควบคุมปริญญานิพนธ์

1. อาจารย์กิตติพงศ์ มะโน
2. อาจารย์ ดร.สุรสิทธิ์ ราษฎร์
3. อาจารย์สุชิน อาจหาญ

คณะกรรมการสอบปริญญานิพนธ์	ลายมือชื่อ
1. อาจารย์กิตติพงศ์ มะโน	
2. อาจารย์วรวีทย์ สมหา	
3. อาจารย์สุชิน อาจหาญ	
4. อาจารย์พีระวุฒิ สุวรรณจันทร์	
5. อาจารย์สันติ ดันตระกูล	

วันเดือนปีที่สอบ วันที่ 8 ธันวาคม 2538 เวลา 17.00 ถึง 19.00 น.

สถานที่สอบ ห้อง ค.303 คณะครุศาสตร์อุตสาหกรรม



ใบรับรองแล้ว
(ศาสตราจารย์ ดร.สุจินต์ ฌ. อรุณยา)
คณบดี คณะครุศาสตร์อุตสาหกรรม

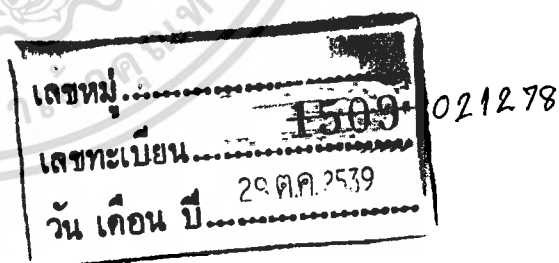
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษา วันที่ 17 เดือน 20 พ.ศ. 77

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งในการนำไปใช้

ปริญญานิพนธ์
เครื่องโปรแกรมอุปกรณ์ลอจิกแบบโปรแกรมได้
Programmable Logic Device Programmer



นายวิฑูรย์ ฉายกะพันธ์
นายสุรพงษ์ ชาสุวรรณ
นายอนุสิทธิ์ บุญมาก
นางสาวอรทัย ใจคำลือ



ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรครุศาสตรบัณฑิต
สาขาอิเล็กทรอนิกส์และคอมพิวเตอร์
ภาควิชาครุศาสตร์วิศวกรรม คณะครุศาสตรอุตสาหกรรม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์

เรื่อง เครื่องโปรแกรมอุปกรณ์ลอจิกแบบโปรแกรมได้
Programmable Logic Device Programmer

ผู้จัดทำ

1. นายวิฑูร ฉายกะพันธ์
2. นายสุรพงษ์ ชาสุวรรณ
3. นายอนุสิทธิ์ บุญมาก
4. นางสาวอรทัย ใจคำลือ

อาจารย์ที่ปรึกษา

ลงนาม
(อาจารย์กิตติพงศ์ มะโน)

ลงนาม
(อาจารย์ ดร.สุรสิทธิ์ รัตรี)

ลงนาม
(อาจารย์สุชิน อาจหาญ)

หัวหน้าภาควิชา

ลงนาม
(ผศ.ดร.ธีระพล เทพหัสดิน ณ อยุธยา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์
เครื่องโปรแกรมอุปกรณ์ลอจิกแบบโปรแกรมได้
Programmable Logic Device Programmer

จุดประสงค์

1. เพื่อศึกษาคุณสมบัติ ลักษณะโครงสร้างของอุปกรณ์ประเภท PLD
2. เพื่อออกแบบเครื่องโปรแกรมอุปกรณ์ลอจิกแบบโปรแกรมได้
3. เพื่อสร้างเครื่องโปรแกรมอุปกรณ์ลอจิกแบบโปรแกรมได้พร้อมชุดทดสอบ
4. เพื่อนำเครื่องโปรแกรมอุปกรณ์ลอจิกแบบโปรแกรมได้ไปใช้ในการออกแบบวงจรลอจิกต่าง ๆ
5. เพื่อนำเครื่องโปรแกรมอุปกรณ์ลอจิกแบบโปรแกรมได้มาทดสอบ และใช้งานจริง

ประโยชน์ที่คาดว่าจะได้รับ

1. มีความรู้เกี่ยวกับคุณสมบัติ ลักษณะโครงสร้างของอุปกรณ์ประเภท PLD
2. สามารถออกแบบเครื่องโปรแกรมอุปกรณ์ลอจิกแบบโปรแกรมได้
3. สามารถสร้างเครื่องโปรแกรมอุปกรณ์ลอจิกแบบโปรแกรมได้
4. สามารถนำเครื่องโปรแกรมอุปกรณ์ลอจิกแบบโปรแกรมได้ไปใช้ในการออกแบบวงจรลอจิกต่าง ๆ
5. สามารถนำอุปกรณ์ที่ออกแบบจากเครื่องโปรแกรมอุปกรณ์ลอจิกแบบโปรแกรมได้ไปใช้งานจริงได้

เครื่องโปรแกรมอุปกรณ์ลอจิกแบบโปรแกรมได้

นายวิฑูร จายกะพันธ์
 นายสุรพงษ์ ชาสุวรรณ
 นายอนุสิทธิ์ บุญมาก
 นางสาวอรทัย ใจคำลือ

อาจารย์ที่ปรึกษา

อาจารย์กิติพงศ์ มะโน
 คร.สุรสิทธิ์ ราชตรี
 อาจารย์สุชิน ออาจหาญ
 ปีการศึกษา 2538

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ได้กล่าวถึงการออกแบบและการสร้างเครื่องโปรแกรม PLD ที่สามารถติดต่อกับไมโครคอมพิวเตอร์ IBM Compatible ได้ และ เขียนซอฟต์แวร์ภาษาแอสเซมบลี MCS-51 ที่สามารถโปรแกรมอุปกรณ์ PAL พร้อมทั้งเขียนซอฟต์แวร์ภาษาซี ที่สามารถติดต่อและควบคุมเครื่องโปรแกรม PLD ได้ สำหรับข้อมูลที่จะทำการโปรแกรม จะเป็นแฟ้มข้อมูลแบบ JEDEC ซึ่งแฟ้มข้อมูลแบบ JEDEC นี้สามารถเขียนและคอมไพล์ได้จากโปรแกรม OPAL และ OrCAD / PLD ซึ่งให้ผลได้ตามที่คาดหมายไว้

PROGRAMMABLE LOGIC DEVICE PROGRAMMER

MR.WITOON CHAYKAPHUN

MR.SURAPONG CHASUWAN

MR.ANUSIT BOONMAK

MISS.ORATHAI JAIKAMLUE

ADVISORS

MR.KITIPHONG MANO

DR.SURASIT RATREE

MR.SUCHIN ADHAN

1995

ABSTRACT

In this thesis, presents the design and implementation of the PLD programmer . It is controlled by IBM PC compatible microcomputer and software , that is written by assembly of MCS-51 and C language. The C language is used to attach between MCS-51 board and jedec file , which is writed by OPAL and OrCAD / PLD software. That complete in the objectives.

กิติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงมาได้เพราะ พระคุณของ คุณพ่อ คุณแม่ และความกรุณาของคณะอาจารย์ภาควิชาครุศาสตร์วิศวรกรรมทุกท่าน โดยเฉพาะอย่างยิ่งท่าน อาจารย์กิติพงศ์ มะโน ที่ได้ให้ความช่วยเหลือ และสนับสนุนคณะผู้จัดทำมาโดยตลอด ทั้งให้คำแนะนำ และให้การสนับสนุนทางด้านอุปกรณ์ต่าง ๆ ที่ใช้ในการทำปริญญานิพนธ์ ซึ่งในโอกาสนี้คณะผู้จัดทำ รู้สึกซาบซึ้งในความกรุณาของทุก ๆ ท่าน จึงขอขอบพระคุณเป็นอย่างยิ่งมา ณ โอกาสนี้ด้วย



สารบัญ

เรื่อง	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญภาพ	VIII
บทที่ 1 บทนำ	1
บทที่ 2 ทฤษฎีและหลักการ	4
2.1 สัญลักษณ์ลอจิกภายใน PLD	4
2.2 โครงสร้างภายในของ PLD	6
2.3 การโปรแกรมทางด้านแอนด	10
2.3.1 อุปกรณ์ PAL	10
2.3.2 อุปกรณ์ EPLD	23
2.3.3 อุปกรณ์ GAL	25
2.3.4 อุปกรณ์ PLD เบอร์ HY16CV8	27
2.4 การโปรแกรมทางด้านออร์	33
2.4.1 อุปกรณ์ PROM	33
2.5 การโปรแกรมด้านแอนดและออร์	34
2.5.1 อุปกรณ์ FPLA	34
2.6 ข้อดีและข้อเสียของการใช้ PLD	36
2.7 การโปรแกรมระเบิดฟิวส์	36
2.8 การโปรแกรม PAL	38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เรื่อง	หน้า
บทที่ 3 การออกแบบและการสร้าง	43
3.1 คุณสมบัติของบอร์ด MCS-51	43
3.2 การจัดหน่วยความจำบอร์ด MCS-51	45
3.3 การเลือกใช้ไอซีบัฟเฟอร์	46
3.3.1 ไอซีบัฟเฟอร์ที่ใช้กับวงจรเปิด / ปิด	46
3.3.2 ไอซีบัฟเฟอร์ที่ใช้ในการอ่านอุปกรณ์ PAL	46
3.4 การออกแบบวงจรเปิด / ปิดทรานซิสเตอร์	47
3.4.1 การออกแบบวงจรตามตารางความจริงของตำแหน่งอินพุตทางแนวตั้ง	47
3.4.2 การนำวงจรบัฟเฟอร์มาต่อกับวงจรเปิด / ปิดทรานซิสเตอร์	49
3.4.3 การออกแบบวงจรตามตารางความจริงของตำแหน่งอินพุตทางแนวนอน	50
3.4.4 การนำวงจรบัฟเฟอร์มาต่อกับวงจรเปิด / ปิดทรานซิสเตอร์	51
3.5 การรวมวงจรทรานซิสเตอร์	51
3.6 การออกแบบสร้างโปรแกรม	55
3.6.1 ส่วนของภาษาซี	55
3.6.2 ส่วนของภาษาแอสเซมบลี	62
บทที่ 4 ผลการทดลองและทดสอบ	69
4.1 การทดลองวงจรทรานซิสเตอร์ควบคุมระดับแรงไฟ	69
4.2 การทดลองวงจรร่วมขนาดใหญ่	71
4.3 ทดสอบการระเบิดฟิวส์ครั้งละ 1 ตำแหน่ง	74
4.4 ทดสอบการระเบิดฟิวส์ครั้งละ 1 เอาต์พุต	76
4.4.1 วงจรถอดรหัสฐานสิบหกแสดงผลออก 7 เซกเมนต์	81
4.4.2 วงจรถอดรหัส	83
บทที่ 5 สรุปอภิปรายและข้อเสนอแนะ	87

เรื่อง	หน้า
5.2 คุณสมบัติของเครื่องโปรแกรม PAL	88
5.3 ปัญหาและแนวทางแก้ไข	88
5.4 ข้อเสนอแนะและแนวทางการพัฒนา	89
ภาคผนวก ก. คู่มือการใช้งานเครื่องโปรแกรม PAL	90
ภาคผนวก ข. PLD SOFTWARE	94
ภาคผนวก ค. โปรแกรมที่ใช้ร่วมกับเครื่อง PAL	102
ภาคผนวก ง. วงจรเครื่องโปรแกรม PAL	253
บรรณานุกรม	258

สารบัญตาราง

ตาราง	หน้า
ตารางที่ 2.1 ตารางรวมคุณสมบัติอุปกรณ์ PLD	6
ตารางที่ 2.2 คุณสมบัติของ PAL แต่ละเบอร์	22
ตารางที่ 2.3 คุณสมบัติของ HY18CV8 ที่สามารถใช้แทน PAL	32
ตารางที่ 2.4 แถวลำดับของรีจิสเตอร์	33
ตารางที่ 2.5 การป้อนสัญญาณต่าง ๆ เข้าที่ขา PAL ในการเลือกตำแหน่งที่ต้องการทางแนวตั้ง	40
ตารางที่ 2.6 การป้อนสัญญาณต่าง ๆ เพื่อกำหนดตำแหน่งทางแนวนอน ทั้งช่วงสูง และ ช่วงต่ำ	41
ตารางที่ 4.1 ผลการทดลองวงจรไฟ 3 ระดับ	70
ตารางที่ 4.2 ผลการทดลองวงจรไฟ 2 ระดับ	70
ตารางที่ 4.3 ผลการทดสอบบอร์ดโดยใช้โปรแกรม TEST.ASM เทียบกับค่าที่วัดได้	77
ตารางที่ 4.4 ผลการทดสอบอุปกรณ์หลังการโปรแกรมไฟล์ GATES.JED	80
ตารางที่ 4.5 ผลการทดสอบอุปกรณ์หลังจากโปรแกรมไฟล์ HEX-SEG.JED	82
ตารางที่ 4.6 ผลการทดสอบอุปกรณ์หลังการโปรแกรมไฟล์ DECODER.JED	85
ตารางที่ 4.7 ผลการทดสอบการใช้งาน	86

สารบัญภาพ

ภาพ	หน้า
รูปที่ 2.1 สัญลักษณ์ภายในของ PLD	5
รูปที่ 2.2 ก. โครงสร้างของ PLD แบบโปรแกรมทางด้านแอนด	7
รูปที่ 2.2 ข. โครงสร้างของ PLD แบบโปรแกรมทางด้านออร์	8
รูปที่ 2.2 ค. โครงสร้างของ PLD แบบโปรแกรมทางด้านแอนดและด้านออร์	10
รูปที่ 2.3 การอธิบายพื้นฐานภายในโครงสร้าง PAL	10
รูปที่ 2.4 ตัวอย่างวงจรลอจิกภายในโครงสร้างของ PAL	11
รูปที่ 2.5 โครงสร้างของ PAL 2 อินพุต 1 เอาต์พุต	12
รูปที่ 2.6 การเปลี่ยนแปลงลอจิกโดยใช้ PAL	13
รูปที่ 2.7 สัญลักษณ์โครงสร้างของ PAL 2 อินพุต 1 เอาต์พุต	13
รูปที่ 2.8 ตัวอย่างวงจรลอจิกพื้นฐาน $Output = I_1\bar{I}_2 + \bar{I}_1I_2$	14
รูปที่ 2.9 การสร้างลอจิกจากรูปที่ 2.8 โดยใช้ PAL	14
รูปที่ 2.10 การกำหนดหมายเลขประจำตัว PAL	15
รูปที่ 2.11 วงจรทางด้านเอาต์พุตของ PAL ชนิด L	15
รูปที่ 2.12 วงจรทางด้านเอาต์พุตของ PAL ชนิด L ที่มีการป้อนกลับ	16
รูปที่ 2.13 วงจรทางด้านเอาต์พุตของ PAL ชนิด R	17
รูปที่ 2.14 วงจรทางด้านเอาต์พุตของ PAL ชนิด X	17
รูปที่ 2.15 วงจรทางด้านเอาต์พุตของ PAL ชนิด R	18
รูปที่ 2.16 แผนผังคาร์นอของสมการที่เกิดจากการป้อนกลับ	18
รูปที่ 2.17 ลักษณะวงจรที่ PAL สามารถนำค่ามาโปรแกรมใหม่ 16 สมการ	18
รูปที่ 2.18 โครงสร้างภายในของ PAL ที่เป็นแบบเมโครเซล	21
รูปที่ 2.19 วงจรภายในอุปกรณ์ EPLD ใช้มอสทรานซิสเตอร์แทนฟิวส์	24
รูปที่ 2.20 โครงสร้างภายใน GAL	26
รูปที่ 2.21 เมโครเซลของ GAL	26

ภาพ	หน้า
รูปที่ 2.23 แมโครเซลล์ของ HY18CV8	31
รูปที่ 2.24 โครงสร้างภายในของ PROM	34
รูปที่ 2.25 โครงสร้างภายในของ FPLA	35
รูปที่ 2.26 การใช้ไมโครคอมพิวเตอร์ร่วมกับเครื่องโปรแกรมอุปกรณ์ PLD	37
รูปที่ 2.27 PAL เมื่ออยู่ในช่วงการโปรแกรมขาสัญญาณต่าง ๆ จะเปลี่ยนไปขึ้นอยู่กับทางเลือกตำแหน่งทางแนวตั้ง และแนวนอน	39
รูปที่ 2.28 แผนผังเวลาในการโปรแกรมและตรวจสอบ PAL	42
รูปที่ 3.1 ผังการทำงานของเครื่องโปรแกรม PAL	43
รูปที่ 3.2 บอร์ด MCS-51	44
รูปที่ 3.3 ชุดแสดงผล DOT Matrix LCD ขนาด 20 ตัว * 20 แถว	45
รูปที่ 3.4 การจัดหน่วยความจำของระบบ	45
รูปที่ 3.5 โครงสร้างภายในของไอซีเบอร์ 7406	46
รูปที่ 3.6 โครงสร้างภายใน 1 เกท ของไอซีเบอร์ 7406	46
รูปที่ 3.7 โครงสร้างภายในของไอซีเบอร์ 74245	47
รูปที่ 3.8 วงจรควบคุมระดับแรงดัน H กับ L และตารางการทำงาน	47
รูปที่ 3.9 วงจรควบคุมระดับแรงดัน HH กับ L และตารางการทำงาน	48
รูปที่ 3.10 วงจรควบคุมระดับแรงดัน HH , H , L และตารางการทำงาน	49
รูปที่ 3.11 วงจรควบคุมระดับแรงดัน HH , H , L ที่มี 7406 ต่อร่วม และตารางการทำงาน	50
รูปที่ 3.12 วงจรควบคุมระดับแรงดัน HH กับ Z และตารางการทำงาน	51
รูปที่ 3.13 วงจรควบคุมระดับแรงดัน HH กับ Z ที่มี 7406 ต่อร่วม และตารางการทำงาน	51
รูปที่ 3.14 การต่อวงจรที่ใช้โปรแกรมอุปกรณ์ PAL	51
รูปที่ 3.15 บอร์ดที่ใช้ในการทดลอง	53
รูปที่ 3.16 บอร์ดที่ออกแบบใช้งานจริง	54
รูปที่ 3.17 ตำแหน่งการวางอุปกรณ์ภายในเครื่องโปรแกรม PAL	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพ	หน้า
รูปที่ 3.18 เครื่องโปรแกรม PAL	55
รูปที่ 3.19 โครงสร้างของซอฟต์แวร์	56
รูปที่ 3.20 ลักษณะของหน้าจอที่ติดต่อกับผู้ใช้งาน	58
รูปที่ 3.21 ผังงานฟังก์ชันการทำงานของ “Load”	59
รูปที่ 3.22 ผังงานฟังก์ชันการทำงานของ “Chang Dir”	60
รูปที่ 3.23 ผังงานฟังก์ชันการทำงานของ “Dos Shell”	61
รูปที่ 3.24 ผังงานฟังก์ชันการทำงานของ “Edit / Buffer”	62
รูปที่ 3.25 ผังงานการเก็บข้อมูล JEDEC ไฟล์ลงในโปรแกรม	63
รูปที่ 3.26 ผังงานการแสดงผลของ JEDEC ไฟล์	66
รูปที่ 3.27 ผังงานการโปรแกรมระเบิดฟิวส์ขนาด 1 ตำแหน่ง	67
รูปที่ 3.28 ผังงานการโปรแกรมระเบิดฟิวส์แบบชุดทั้งตัว	68
รูปที่ 4.1 วงจรไฟ 3 ระดับ	69
รูปที่ 4.2 วงจรไฟ 2 ระดับ	70
รูปที่ 4.3 วงจรร่วมขนาดใหญ่ที่ใช้ในการทดลอง	71
รูปที่ 4.4 บอร์ดวงจรร่วมขนาดใหญ่ที่ใช้ในการทดลอง	74
รูปที่ 4.5 การทดสอบอุปกรณ์ PAL ที่ผ่านการโปรแกรมโดยใช้เครื่อง EXPRO - 60	75
รูปที่ 4.6 ผลการทดลองการระเบิดฟิวส์ 5 ตำแหน่ง	75
รูปที่ 4.7 ลักษณะโครงสร้างของอุปกรณ์ PAL เมื่อผ่านการโปรแกรม จาก JEDEC ไฟล์	79
รูปที่ 4.8 วงจรทดสอบหลังจากการโปรแกรมไฟล์ GATES.JED	80
รูปที่ 4.9 วงจรทดสอบอุปกรณ์หลังจากการโปรแกรมไฟล์ HEX-SEG.JED	82
รูปที่ 4.10 วงจรทดสอบอุปกรณ์หลังจากการโปรแกรมไฟล์ DECODER.JED	85

บทที่ 1

บทนำ

นับตั้งแต่จวบจรวม หรือเรียกย่อๆ ว่า ไอซี (Integrated Circuit : IC) ประเภทดิจิทัล (Digital) ได้ถูกผลิตขึ้นมาเมื่อประมาณ 20 กว่าปีแล้ว ถือได้ว่าเป็นความสำเร็จขั้นสุดยอดที่ทำให้วงการอิเล็กทรอนิกส์ทั่วโลกในยุคนั้นถูกเปลี่ยนโฉม เข้าสู่ยุคการพัฒนาอย่างรวดเร็ว ประกอบกับเทคโนโลยีการผลิตที่ช่วยให้ขนาดของตัวไอซีเล็กลง จนสามารถนำไอซีนับร้อยนับพันตัวมารวมอยู่บนชิพ(Chip) เดียวกันได้และเราก็ได้รู้จักดีในชื่อของ“ไมโครโปรเซสเซอร์ (Microprocessor)” หรือ วงจรอิเล็กทรอนิกส์ที่ทำงานด้วยโปรแกรม

แต่ถึงจะก้าวล้ำหน้าไปขนาดไหนก็ตามไอซีในยุคต้นๆ ก็ยังนิยมกันอยู่ในปัจจุบัน ด้วยเหตุผลที่ว่า โครงสร้างที่ไม่ซับซ้อนและเข้าใจได้ง่ายสามารถนำมาต่อรวมกันหลาย ๆ ตัวเพื่อให้งานตามระบบที่ออกแบบไว้โดยไม่มีขีดจำกัด จึงช่วยให้นักอิเล็กทรอนิกส์สมัครเล่น หรือ ผู้ผลิตรายย่อย ๆ มีโอกาสสร้างสรรค์สิ่งประดิษฐ์ หรือสินค้าออกมาใช้งานหรือจำหน่ายได้ ในการที่จะเลือกใช้ไอซีลอจิก (Logic) ต่างๆ ในปัจจุบันมีหนทางเลือกอยู่มากมาย ยิ่งจำนวนลอจิกเกต (Logic Gate) ภายในตัวไอซี มากเท่าใด ก็จะทำให้การออกแบบวงจรทำได้ง่ายขึ้นตามไปด้วย

ในยุคเริ่มต้นของไอซี ที่ผลิตออกมาในลักษณะการค้าเป็นประเภท SSI (Small Scale Integrated) เช่น ไอซี TTL ตระกูล 7400 หรือ CMOS ตระกูล 4000 ซึ่งเป็น ลอจิกเกตง่าย ๆ ต่อมาเทคโนโลยีการผลิตสูงขึ้นสามารถเพิ่มจำนวนลอจิกเกต ลงในชิพ ได้มากขึ้นเลยผลิตออกมาเป็น MSI (Medium Scale Integrated) เช่น เคาน์เตอร์ (Counter) หรือ ไอซีทำหน้าที่คำนวณซึ่งไอซี ทั้ง 2 นี้มีราคาถูกมากและนักอิเล็กทรอนิกส์ทั่วไปมีโอกาสได้ใช้มาเป็นเวลานานแล้ว

PLD (Programmable Logic Devices) เป็น ไอซีที่ผู้ใช้สามารถนำไปโปรแกรมได้เอง โดยขบวนการทางไฟฟ้าโดยไม่ยุ่งยากผู้ใช้สามารถออกแบบ และ กำหนดเลือกชนิดของเกตและการต่อเกตได้ตามความต้องการแต่ก็มีขีดจำกัดที่จำนวนเกต และความซับซ้อนของการต่อวงจร ข้อดีของการใช้อุปกรณ์ PLD แทนไอซีดิจิทัลประเภทกรรมดามีอยู่หลายข้อด้วยกัน ได้แก่ การลดต้นทุนการผลิต สามารถลดเวลาและขั้นตอนการออกแบบ รวมไปถึง

ถึงการลดขนาดของวงจรพิมพ์ให้เล็กลง จึงทำให้ PLD มีโอกาสที่จะเข้ามาแทนไอซีธรรมดา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในเรื่องของคุณภาพ PLD จะได้เปรียบกว่ามากในแง่ของความเชื่อได้ของวงจร เพราะจำนวนไอซี และจุดบัดกรีจะน้อยลง เป็นการลดปัญหาที่จะเกิดขึ้นภายหลัง และ การสิ้นเปลืองกระแสไฟฟ้าก็จะน้อยกว่าด้วย ทำให้ระบบระบายความร้อนมีขนาดเล็กลงตามไปด้วย ข้อดีอีกประการหนึ่งซึ่งอาจมองไม่เห็นได้ในขณะนี้ คือ จำนวนสต็อก หรือเบอร์ไอซี สำหรับร้านค้าจะลดน้อยลงไปด้วย เพียง PLD แค่ 20-30 เบอร์ สามารถที่จะแทนหน้าที่เดียวกันของไอซี SSI หรือ LSI ได้ถึงหลายร้อยเบอร์ ในอนาคตร้านค้าอาจจะใช้วิธีโปรแกรมหน้าที่การทำงานของ PLD ให้ลูกค้าได้เลยโดยเพียงแค่เสียบแผ่นดิสก์เข้ากับเครื่องคอมพิวเตอร์แล้ว กดคีย์เลือกเท่านั้น และต่อไปในอนาคตอุปกรณ์ PLD จะมีบทบาทต่อการอิเล็กทรอนิกส์ของไทย แต่ในปัจจุบันเครื่องโปรแกรมอุปกรณ์ยังมีราคาค่อนข้างแพงอยู่ทำให้มีแนวคิดที่จะสร้างเครื่องโปรแกรม PLD ขึ้นมา โดยได้กำหนดเป้าหมายในการทำปริญญานิพนธ์มีรายละเอียดดังนี้ คือ สร้างเครื่องโปรแกรมอุปกรณ์ PLD ที่สามารถโปรแกรมอุปกรณ์ PAL ได้ และสามารถนำอุปกรณ์ที่ผ่านการโปรแกรมมาทดสอบกับชุดทดลองได้

ในปริญญานิพนธ์ฉบับนี้กล่าวถึงรายละเอียดต่าง ๆ มีหัวข้อที่สำคัญดังต่อไปนี้

บทที่ 2 กล่าวถึงทฤษฎีและหลักการ ของอุปกรณ์ PLD ได้แก่ สัญลักษณ์ที่ใช้ของอุปกรณ์ PLD , โครงสร้างของ PLD แต่ละชนิด , ข้อดีและข้อเสียของการใช้ PLD , การโปรแกรมการระเบิดฟิวส์ , การโปรแกรม PAL

บทที่ 3 การออกแบบและการสร้าง กล่าวถึงอุปกรณ์ที่ใช้ การออกแบบสร้างเครื่องโปรแกรมแต่ละภาค และการเขียนโปรแกรม มีรายละเอียดดังนี้ , คุณสมบัติของบอร์ด MCS - 51 , การจัดหน่วยความจำบอร์ด MCS - 51 , การเลือกใช้ไอซีบัฟเฟอร์ , การออกแบบวงจรเปิด / ปิดทรานซิสเตอร์ , การรวมวงจรทรานซิสเตอร์ และ การออกแบบสร้างโปรแกรม

บทที่ 4 ผลการทดลองและทดสอบ ในบทนี้จะกล่าวถึงการทดลองวงจรในส่วนต่างๆ ที่ได้ออกแบบไว้ในบทที่ 3 และทดสอบการทำงานของเครื่องโปรแกรมอุปกรณ์ PLD ที่สร้างขึ้น การทดลองและทดสอบมีรายละเอียดดังนี้ , การทดลองวงจรทรานซิสเตอร์ควบคุมระดับแรงดันไฟ , การทดลองวงจรรวมขนาดใหญ่ , ทดสอบการระเบิดฟิวส์ครั้งละ 1 ตำแหน่ง ทดสอบการระเบิดฟิวส์ครั้งละ 1 เอาต์พุต

บทที่ 5 สรุปอภิปรายและข้อเสนอแนะ ในบทนี้จะกล่าวถึงรายละเอียดต่าง ๆ ดังนี้

บทสรุป , คุณสมบัติของเครื่องโปรแกรม PAL , ปัญหาและแนวทางแก้ปัญหา , ข้อเสนอแนะและแนวทางการพัฒนา

ส่วนรายละเอียดต่าง ๆ ของวงจร อุปกรณ์ที่ใช้ และส่วนของโปรแกรม รวมทั้งวิธีการใช้งานได้แสดงไว้ในภาคผนวก



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

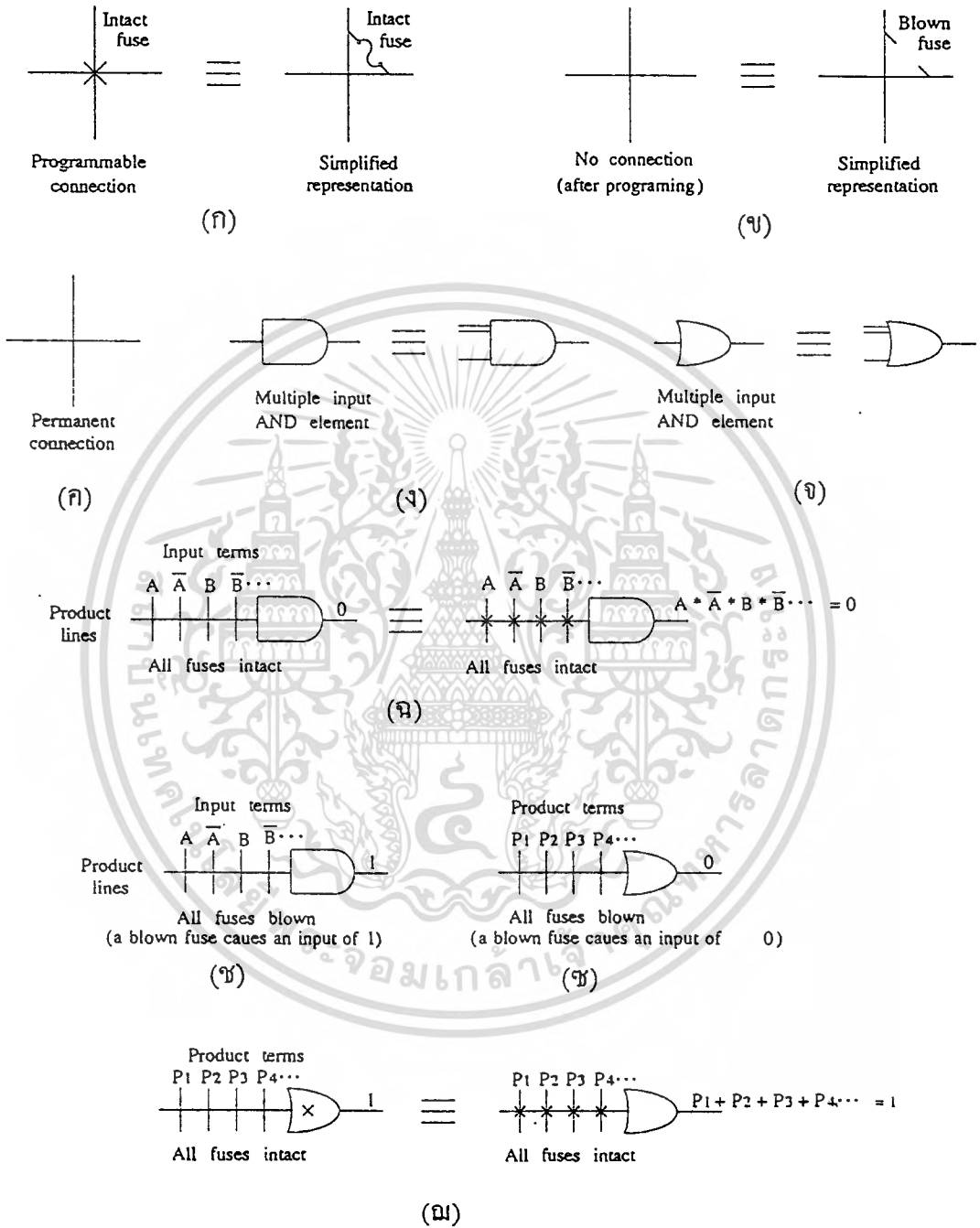
ทฤษฎีและหลักการ

PLD (Programmable Logic Device) คือ อุปกรณ์อิเล็กทรอนิกส์ชนิดหนึ่ง ที่สามารถโปรแกรมการทำงานภายในเป็นวงจรต่าง ๆ ทางดิจิทัล และ คอมพิวเตอร์ได้ตามต้องการ โดยใช้คุณสมบัติของแอนด์เกต (AND Gate) และ ออร์เกต (OR Gate) เป็นตัวกำหนดการทำงาน ภายใน PLD จะประกอบด้วยส่วนของแอนด์ (AND Element) และส่วนของออร์ (OR Element) เป็นส่วนสำคัญซึ่งจะอาศัยสมการบูลีน ช่วยในการออกแบบการโปรแกรม และ เมื่อทำการโปรแกรมใช้งานเรียบร้อยแล้ว ยังสามารถโปรแกรมปรับปรุงการทำงานในบางส่วนได้อีกด้วย ปัจจุบันอุปกรณ์ PLD ได้เข้ามามีบทบาทเป็นอย่างมากในวงการอิเล็กทรอนิกส์ และคอมพิวเตอร์ จึงจำเป็นอย่างยิ่งที่จะต้องศึกษาถึงการทำงานภายในของอุปกรณ์ตัวนี้

2.1 สัญญลักษณ์ลอจิก (Logic) ภายใน PLD

สัญญลักษณ์ลอจิกที่ถูกใช้แทนลอจิกภายในของอุปกรณ์ PLD แสดงไว้ในรูปที่ 2.1 จากรูปที่ 2.1 (ก) จะใช้เครื่องหมายกากบาท “X” ทับบนเส้นตรง 2 เส้นตัดกันแทนฟิวส์ที่ยังไม่ถูกกระเปิด (Intact Fuse) ก่อนจะถูกโปรแกรม จากรูปที่ 2.1 (ข) จะใช้เส้นตรง 2 เส้นตัดกันแทนฟิวส์ที่ถูกกระเปิด (Blown Fuse) หลังจากที่ถูกโปรแกรมแล้ว รูปที่ 2.1 (ค) จะใช้แทนจุดต่อคงที่ (Fixed) ซึ่งไม่สามารถโปรแกรมได้ รูปที่ 2.1 (ง) และรูปที่ 2.1 (จ) จะใช้รูปส่วนของแอนด์ และส่วนของออร์ซึ่งทางด้านซ้ายมือมีเพียง 1 อินพุตแทนแอนด์เกต และ ออร์เกต ทางด้านขวามือจะมีหลายอินพุต (Input) เพื่อความสะดวกในการเขียนรูปลอจิก ปกติจะใช้แทนจำนวนอินพุต 32 อินพุตหรือมากกว่าในอุปกรณ์ PLD บางชนิด รูปส่วนของแอนด์ และส่วนของออร์ ทางด้านซ้ายมือของรูปที่ 2.1 (ฉ) และรูปที่ 2.1 (ช) จะใช้แทนสมการแอนด์ ของรูปขวามือ ซึ่งมีค่าเท่ากับ “0” และใช้แทนสมการออร์ของรูปขวามือ ซึ่งมีค่าเท่ากับ “1” ตามลำดับ จะเห็นว่าในรูปได้เขียนเครื่องหมายกากบาท “X” ไว้ข้างในส่วนของแอนด์ และ ส่วนของออร์ ซึ่งหมายถึงอินพุตทั้งหมดยังมีฟิวส์ต่อใช้งานอยู่นั่นเอง ส่วนรูปที่ 2.1 (ซ) และรูปที่ 2.1 (ฌ) จะใช้แทนสมการแอนด์ ซึ่งมีค่าเท่ากับ “1” และใช้แทนสมการออร์ ซึ่งมีค่าเท่ากับ “0” ตามลำดับในกรณีที่ฟิวส์ถูกโปรแกรมให้ระเบิดไปแล้ว

เอกสารนี้เป็นเอกสารของสำนักงานคณะกรรมการการอุดมศึกษา กระทรวงศึกษาธิการ
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 สัญลักษณ์ภายในของ PLD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 โครงสร้างภายในของ PLD

โครงสร้างภายในของ PLD แบ่งประเภทของการโปรแกรม PLD เป็นดังนี้

ก. การโปรแกรมทางด้านแอนด์ (Programmable AND-fixed OR) ได้แก่

- อุปกรณ์ PAL (Programmable Array Logic)
- อุปกรณ์ EPLD (Erasable Programmable Logic Device)
- อุปกรณ์ GAL (Gate Array Logic)

ข. การโปรแกรมทางด้านออร์ (Fixed AND-programmable OR) ได้แก่

- อุปกรณ์ PROM (Programmable Read Only Memory)

ค. การโปรแกรมทั้งด้านแอนด์ และ ด้านออร์ (Programmable AND - Programmable OR) ได้แก่

- อุปกรณ์ FPLA (Field Programmable Logic Array)

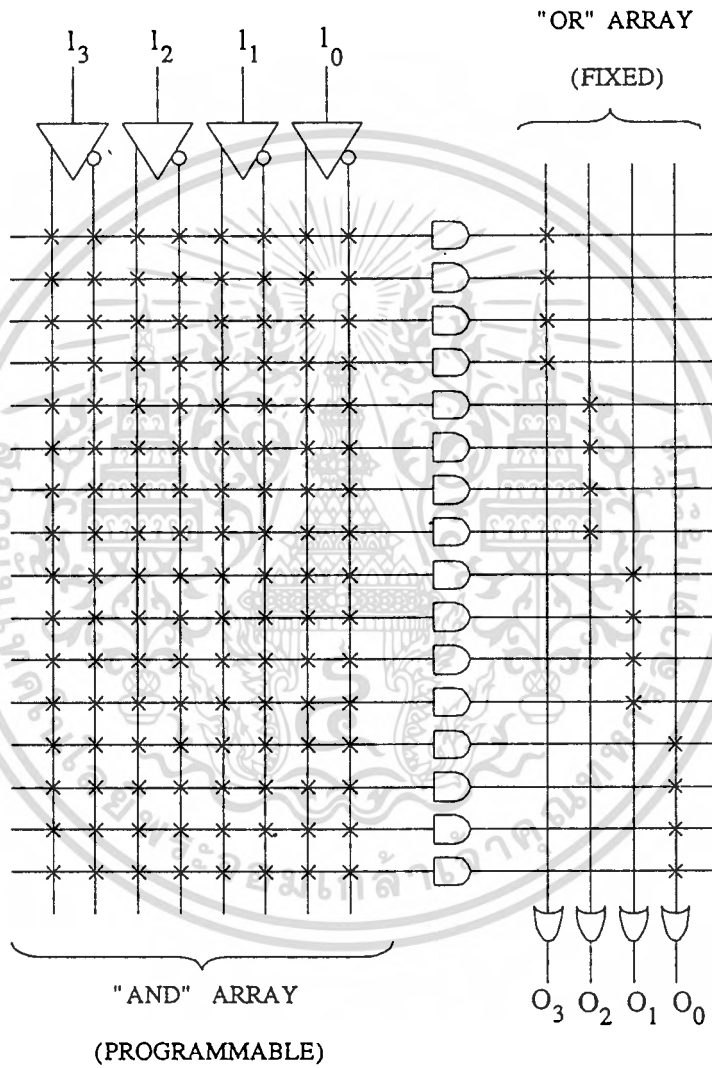
โครงสร้างภายใน และตารางคุณสมบัติของ PLD แต่ละประเภท แสดงในรูปที่ 2.2 (ก), (ข) และ (ค) และตารางที่ 2.1 ตามลำดับ

	AND	OR	OUTPUT OPTIONS
PROM	Fixed	Prog	TS, OC
FPLA	Prog	Prog	TS, OC, Fusible Polarity
FPGA	Prog	None	TS, OC, Fusible Polarity
FPLS	Prog	Prog	TS, Registered Feedback, I/O
PAL	Prog	Fixed	TS, Registered Feedback, I/O

ตารางที่ 2.1 ตารางรวมคุณสมบัติอุปกรณ์ PLD

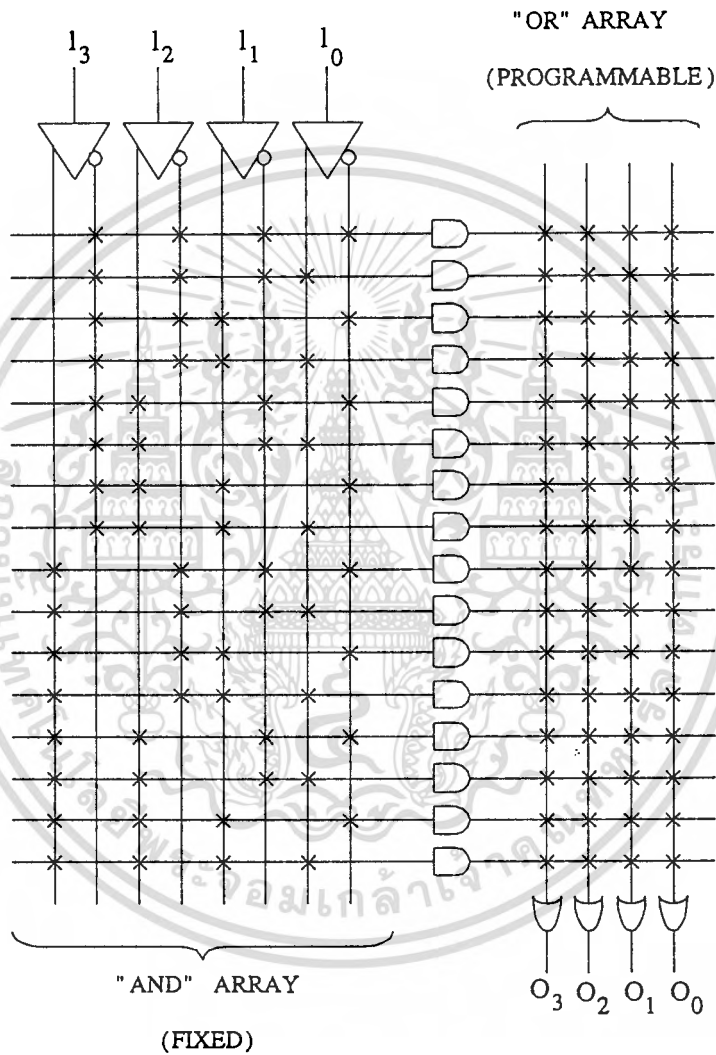
PAL

4 In : 4 Out : 16 Product



รูปที่ 2.2 (ก) โครงสร้างของ PLD แบบโปรแกรมทางด้านแอนด์

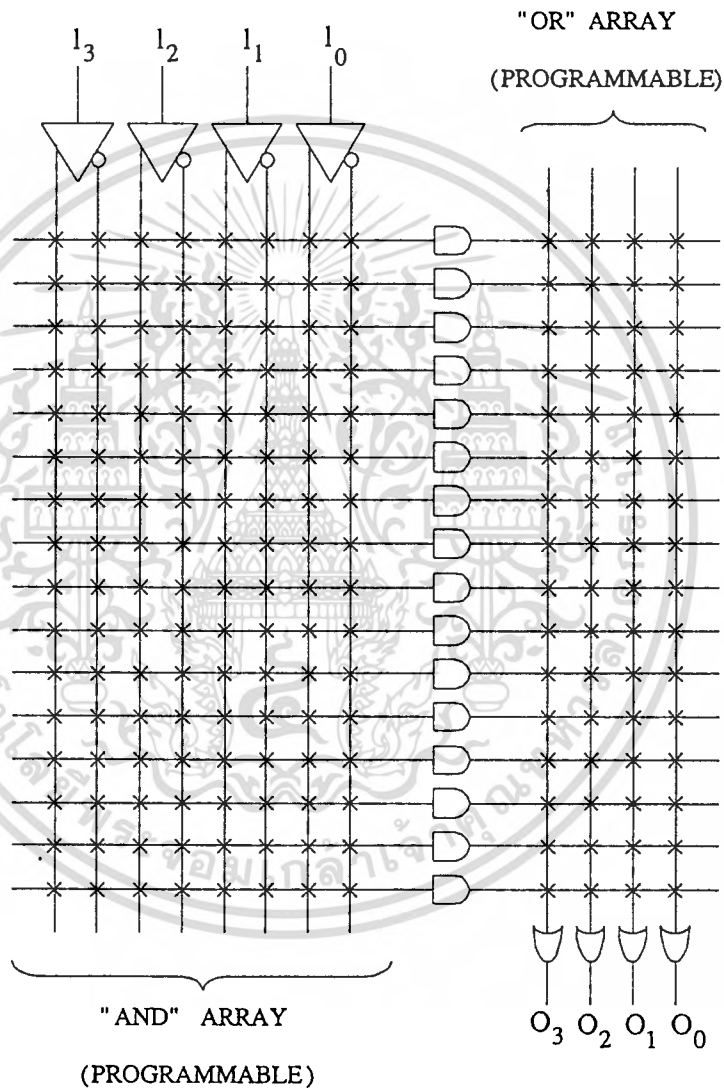
PROM

16 Words $\times$ 4 Bits

รูปที่ 2.2 (ข) โครงสร้างของ PLD แบบโปรแกรมทางด้านอินพุต

FPLA

4 In : 4 Out : 16 Products



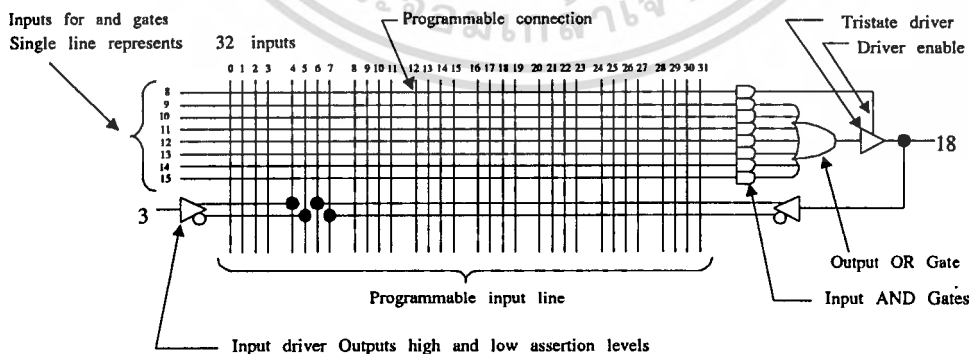
รูปที่ 2.2 (ค) โครงสร้างของ PLD แบบโปรแกรมทางด้านแอนด์ และ ทางด้านออร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

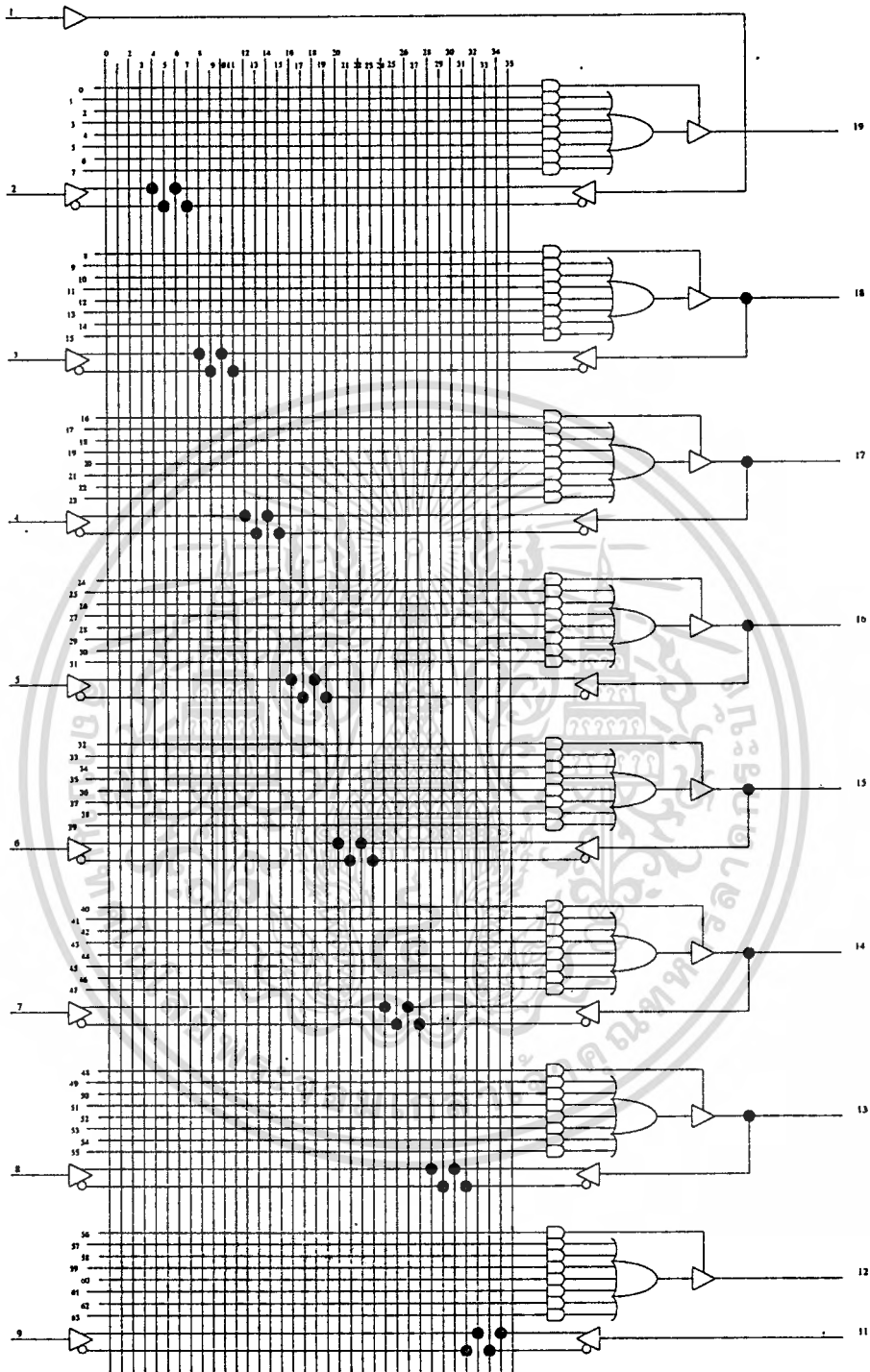
2.3 การโปรแกรมทางด้านแอนด์

2.3.1 อุปกรณ์ PAL

บริษัท MMI (Monolithic Memories Inc.) เป็นต้นกำเนิดในการผลิต อุปกรณ์ที่เรียกว่า PAL (Programmable Array Logic) โดย John Birkner ในปี ค.ศ. 1976 โครงสร้างภายในของ PAL จะทำการโปรแกรมทางด้านแอนด์เกต แต่จะคงส่วนทางด้านออร์เกต มีลักษณะที่เป็นทั้งแบบทีทีแอล (TTL) และ ซีเอ็มอส (CMOS) สำหรับเอาต์พุต (Output) ที่มาจาก PAL จะมีทั้งแบบแอคทีฟ “1” (Active High) , แอคทีฟ “0” (Active Low) , แบบเติมเต็ม (Complementary), แบบมีการป้อนกลับ และแบบมีรีจิสเตอร์ (Register) รูปที่ 2.3 การอธิบายรูปพื้นฐานภายในโครงสร้าง PAL โดยที่ภายในจะมีการต่อถึงกันในแต่ละเส้น เพื่อทำการโปรแกรมระหว่างเส้นในแนวดิ่ง 32 เส้น ซึ่งเราเรียกว่าเส้นอินพุต (Input Line) และ เส้นที่อยู่ในแนวระดับของอินพุตจะต่อเข้ากับแอนด์เกต และเส้นที่ต่อเข้ากับแอนด์เกต ทั้งหมดจะต่อเข้ากับออร์เกต สำหรับขาอินพุตที่ใช้ จะไปต่อกับวงจรภายนอกเรียกว่าเป็นตัวขับ (Driver) โดยที่ขาอินพุตนี้จะถูกกำหนดให้เป็นได้ทั้งระดับ “1” และ ระดับ “0” และกรณีขาเอาต์พุตจากรูปจะถูกใช้ให้เป็นได้ทั้งอินพุต และเอาต์พุต เพราะตัวขับเอาต์พุตเป็นแบบ 3 สถานะ (Tristate) โดยการโปรแกรมผ่าน แอนด์เกต ของทางด้านอินพุตของ PAL ถ้าตัวขับเอาต์พุตเปิดทาง (Enable) แล้วทางด้านเอาต์พุตจะทำหน้าที่เป็นเอาต์พุต ถ้าตัวขับเอาต์พุตปิดทาง (Disable) แล้วขาเอาต์พุตจะทำหน้าที่เป็นอินพุตผ่านทางตัวขับทางด้านอินพุต



รูปที่ 2.3 การอธิบายพื้นฐานภายในโครงสร้าง PAL



รูปที่ 2.4 ตัวอย่างวงจรลอจิกภายในโครงสร้างของ PAL

จากรูปที่ 2.4 เป็นตัวอย่างวงจรลอจิกภายในโครงสร้างของ PAL ประกอบด้วย

8 อินพุต 8 เอาต์พุตจะประกอบด้วยอินพุตและเอาต์พุต โดยที่มีขา 2 เป็นอินพุตเหมือนกับการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ในการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบาย ในรูปที่ 2.3 รวมทั้งขา 1 ก็จะเป็นอินพุตด้วย และให้ผลเอาต์พุตผ่านทาง ขา 19 และในกรณีของขา 9 และ 11 ก็จะแสดงเป็นอินพุต และส่งเอาต์พุตผ่าน ทางขา 12 สรุปแล้ว PAL นี้มีจำนวนของอินพุตทั้งหมด 10 ตัวเป็นการติดต่อกับภายนอกให้ทำหน้าที่เป็นได้ทั้งอินพุตและเอาต์พุต 8 ตัว

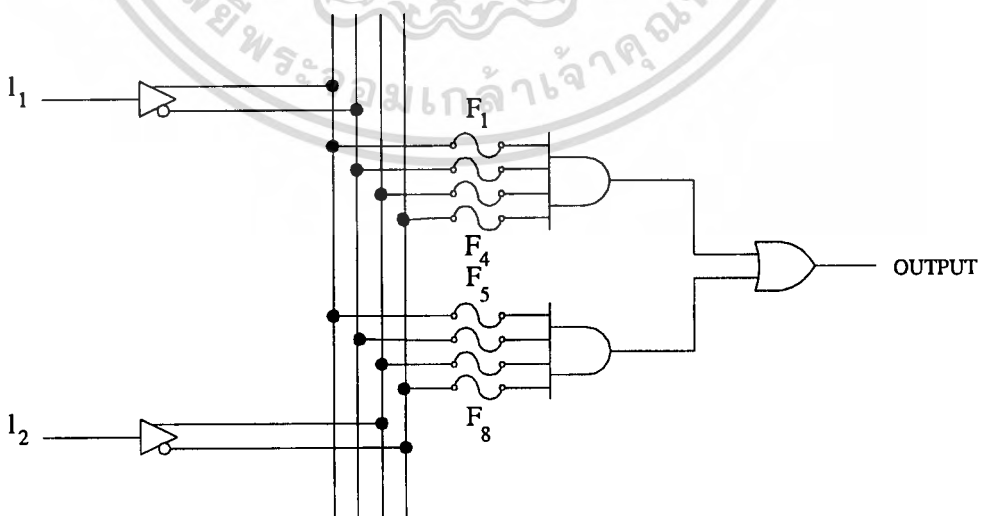
PAL จะอยู่ในตระกูลที่ใช้รูปแบบผลบวกของผลคูณ (Sum of Product) ซึ่งจะใช้งานโดยแถวแอนด์ (And Array) ก็สามารถโปรแกรมได้ ซึ่งกลุ่มของเอาต์พุตจะต่อไปยังออร์เกตต์ตั้งแต่ได้มีการใช้รูปแบบผลบวกของผลคูณ จึงทำให้สามารถนำสมการบูลีนมาทำการเปลี่ยนรูปให้สามารถ ใช้งานในงาน PAL ได้ วงจรภายในของ PAL จะใช้งานอย่างจำกัดตามจำนวนของแถวแอนด์-แถวออร์ อุปกรณ์ PAL จะมีจำนวนของลอจิกที่แตกต่างกันเพื่อ จะให้ผู้ใช้สามารถเลือกใช้ตามขนาดใหญ่-เล็ก ของวงจรที่ออกแบบมา รูปที่ 2.5 แสดง โครงสร้างพื้นฐานของ PAL สำหรับ 2 อินพุต 1 เอาต์พุตสมการลอจิก

$$\text{Output} = (I_1 + \bar{I}_1)(\bar{I}_1 + \bar{I}_2)(I_2 + \bar{I}_3)(\bar{I}_2 + \bar{I}_4) + (I_1 + \bar{I}_5)(\bar{I}_1 + \bar{I}_6)(I_2 + \bar{I}_7)(\bar{I}_2 + \bar{I}_8)$$

โดยที่ f แสดงสถานะของฟิวส์ที่ใช้เชื่อมต่อใน PAL แอนด์อาร์เรย์ฟิวส์ที่ยังไม่ได้ระเบิดจะแสดงด้วยลอจิก '1' ดังนั้นจึงได้ว่า

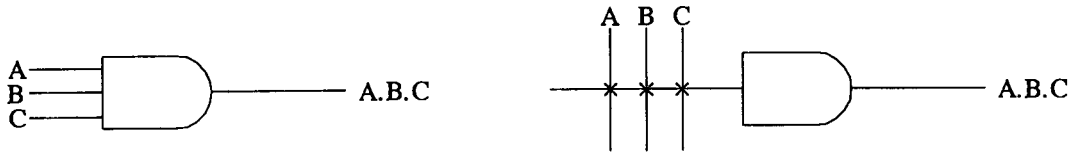
ฟิวส์ระเบิด $f = 0$

ฟิวส์ยังไม่ถูกระเบิด $f = 1$

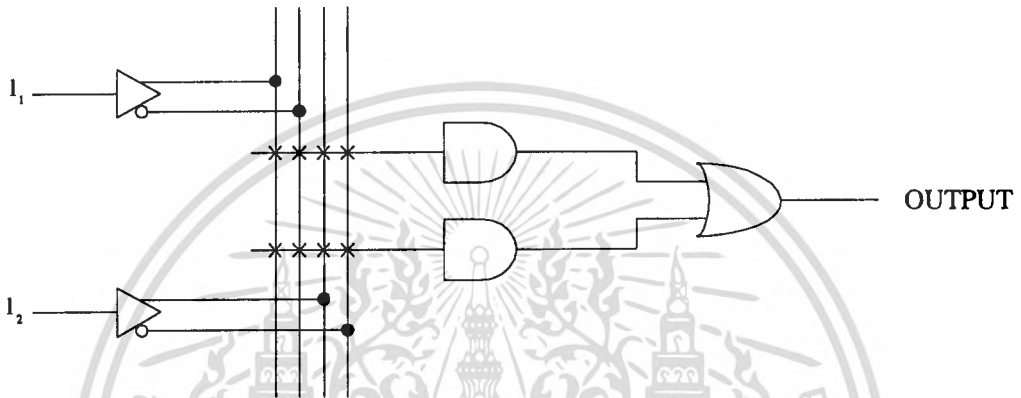


รูปที่ 2.5 โครงสร้างของ PAL 2 อินพุต 1 เอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 การเปลี่ยนแปลงลอจิกโดยใช้ PAL

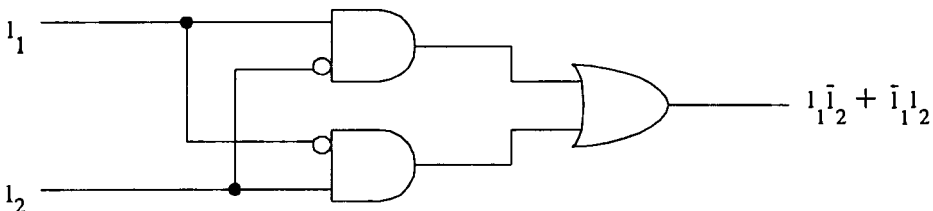


รูปที่ 2.7 สัญลักษณ์โครงสร้างของ PAL 2 อินพุต 1 เอาต์พุต

การโปรแกรมเล็กลงจกเป็นการส่งสัญญาณกระแสไฟฟ้าเข้าไปเพื่อตัดฟิวส์ที่ไม่ต้องการต่อเข้ากับแอนด์เกต ออกให้เหลือเฉพาะสัญญาณที่ต้องการใช้

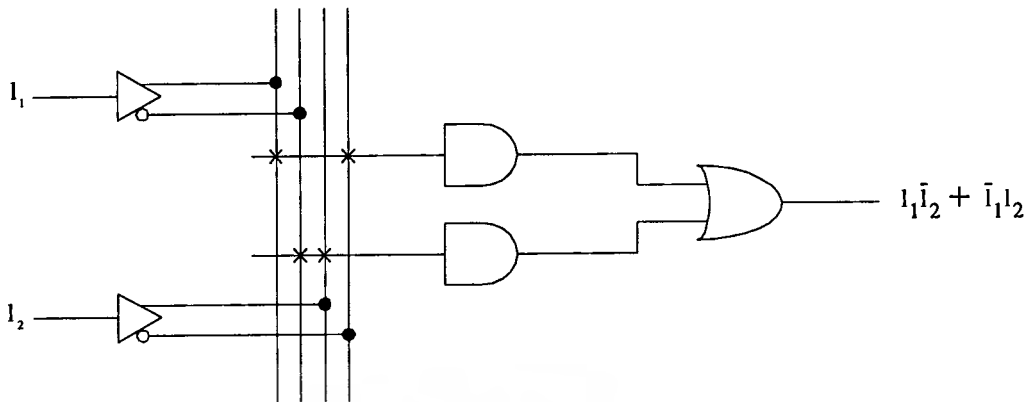
จากตัวอย่างของ PAL พิจารณาถึงส่วนของผลรวมของเอาต์พุตที่ได้รูปวงจรถลอจิกพื้นฐานที่นำมาประกอบการอธิบายดังรูปที่ 12.8 ยังสามารถเขียนให้อยู่ในรูปของวงจรมื่อนำมาสร้างโดยใช้ PAL จะได้ดังรูปที่ 2.9

$$\text{Output} = I_1 \bar{I}_2 + \bar{I}_1 I_2$$



รูปที่ 2.8 ตัวอย่างวงจรถลอจิกพื้นฐาน $\text{Output} = I_1 \bar{I}_2 + \bar{I}_1 I_2$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

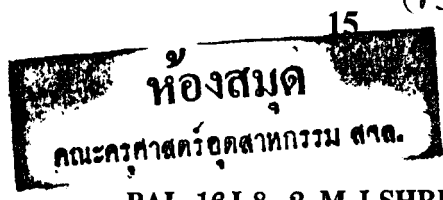


รูปที่ 2.9 การสร้างลอจิกจากรูปที่ 2.8 โดยใช้ PAL

ลักษณะวงจรทางด้านเอาต์พุต

PAL มีลักษณะวงจรทางเอาต์พุตที่ต่างกันหลายแบบ หมายเลขประจำตัวของ PAL แต่ละตัวจะบอกถึงจำนวนอินพุตชนิดของ PAL จำนวนเอาต์พุตและความหมายอื่น ๆ ในรูปที่ 2.10 แสดงความหมายของหมายเลขประจำตัว PAL ที่บริษัท MMI กำหนดขึ้นชนิดต่าง ๆ ทางด้านเอาต์พุตของ PAL มีดังนี้

- H - แอกทีฟที่ "1" เอาต์พุตของ PAL ชนิด H จะเป็นบัฟเฟอร์(buffer) ที่เป็นสภาวะลอจิก ตรงกับออร์เกต
- L - แอกทีฟที่ "0" เอาต์พุตของ PAL จะเป็นอินเวอร์ท 3 สถานะ (inverter tristate) ที่ให้สภาวะลอจิกตรงข้ามกับออร์เกต
- R - เป็นรีจิสเตอร์ หมายถึงรักษาสถานะอินพุตเอาต์พุตให้ค้างไว้ด้วยฟลิปฟลอป และเอาต์พุตยังป้อนย้อนกลับมาที่อินพุตของ AND ได้ แสดงดังรูปที่ 2.13
- X - เป็นรีจิสเตอร์แบบเอกซ์คลูซีฟออร์ (Exclusive-or Register) จะมี เอกซ์คลูซีฟออร์เกต(Exclusive-or Gate) ต่อจากเอาต์พุตของออร์เกต 2 ตัว และเอาต์พุตของเอกซ์คลูซีฟออร์ต่อเข้ากับรีจิสเตอร์ ดังรูปที่ 2.14
- A - เป็นรีจิสเตอร์ที่สามารถคำนวณทางคณิตศาสตร์ได้ (Arithmetic Registered) คือรีจิสเตอร์ของเอกซ์คลูซีฟออร์ที่มีวงจรเพิ่มสำหรับฟังก์ชันการบวก ลบ และเปรียบเทียบค่าอินพุต กับ สัญญาณเอาต์พุตจากรีจิสเตอร์ ดังรูปที่ 2.15



PAL 16L8 - 2 M J SHRP P01234

PAL = Programmable Family
HAL = Hard Array Famil

NUMBER OF ARRAY INPUTS

OUTPUT TYPE
H = Active High
L = Active Lo
C = Complementary
P = Programmable
R = Register
RA = Registered Asynchronous
S = Shared
X = Exclusive OR Registered
A = Arithmetic Registered

NUMBER OF OUTPUTS

SPEED / POWER
A = High Speed
-2 = 1 / 2 Power
-4 = 1 / 4 Power
A - 2 = High Speed and 1 / 2 Power
A - 4 = High Speed and 1 / 4 Power

BIT PATTERN NUMBER

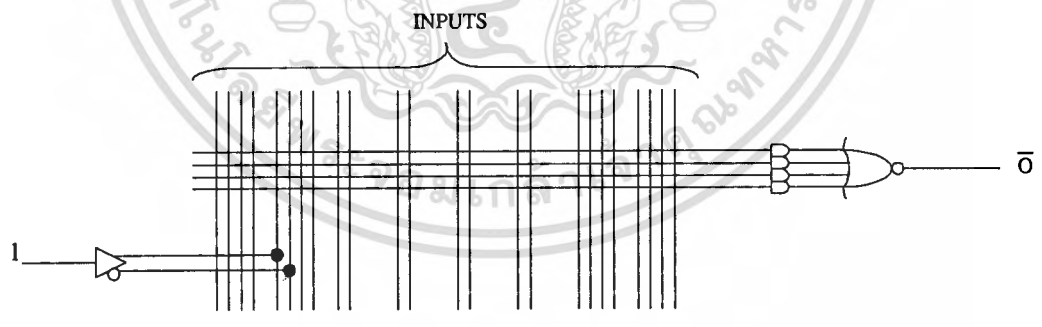
OPTIONAL PROCESSIN
SHRP = Commercial
Reliability Enhance
XXXX = Hi - Rel

PACKAGE
N = Plastic Di
J = Ceramic Dip
F = Flat Pack
NL = Plastic Leadless
JS = Ceramic SKINNYDIP
L = Leadless Chip Carrier
P = Pin Grid Arra

TEMPERATURE COD
C = Commercial
M = Military

รูปที่ 2.10 การกำหนดหมายเลขประจำตัว PAL

แถวเกิดของ PAL จะมีตั้งแต่ 12*2 ซึ่งจะมีทั้งเอาต์พุตทำงานที่ลอจิก “1” และลอจิก “0” ดังรูปที่ 2.11 เป็นวงจรของ PAL ชนิด L PAL นำมาใช้งานได้มากมายตามจำนวนของอินพุต และเอาต์พุต ซึ่งสามารถเป็นตัวกำหนดขนาดของ PAL เพื่อนำไปใช้งาน



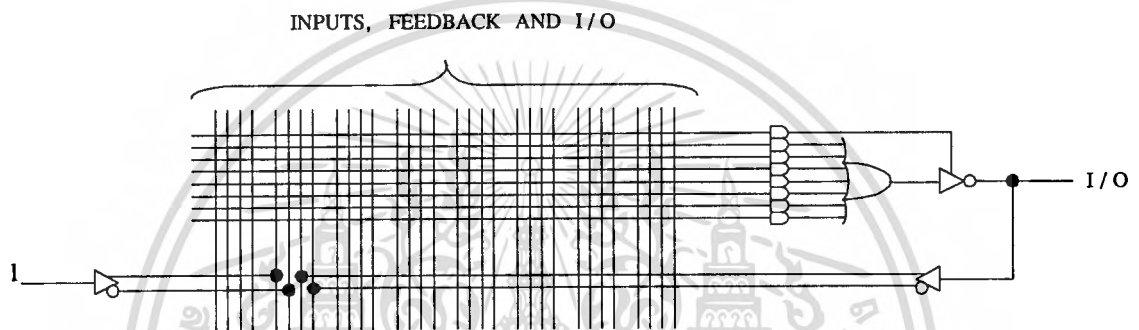
รูปที่ 2.11 วงจรทางด้านเอาต์พุตของ PAL ชนิด L

การโปรแกรมอินพุต และเอาต์พุต

จุดสูงสุดของไอซี ตระกูล PAL คือ สามารถที่จะโปรแกรมอินพุตได้ ซึ่งวิธีการนี้จะทำให้สามารถควบคุมเอาต์พุตของ PAL ได้ดังรูปที่ 2.12 วงจรทางด้านเอาต์พุตของ PAL ชนิด L ซึ่งเอาต์พุตสามารถป้อนกลับเข้าไปยังอินพุตได้ สายเส้นหนึ่งใช้ควบคุมการไหลของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

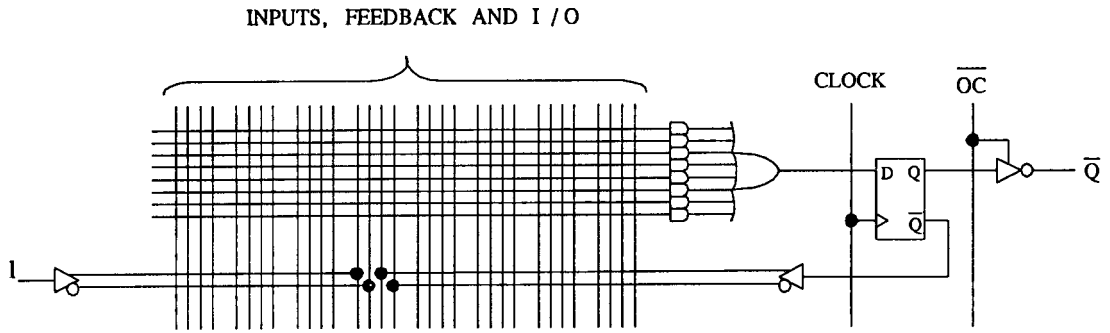
กระแสของบัฟเฟอร์แบบ 3 สถานะ ซึ่งเป็นตัวทำให้เปลี่ยนเป็นขาเอาต์พุตทางขาเอาต์พุตป้อนกลับมายังขาอินพุต คุณสมบัติข้อนี้ให้ประโยชน์ในการออกแบบเป็นตัวเลื่อนข้อมูล (Shift Register) หรือให้เกิดการวนของข้อมูล (Data Loop) และเมื่อควบคุมให้เอาต์พุตหยุดทำงาน หรือมีความต้านทานสูง จะสามารถใช้เป็นขาอินพุตได้อีกด้วย ดังนั้นอุปกรณ์ PAL สามารถที่จะเปลี่ยนแปลงอินพุตได้โดยทำการเปิดลอจิกแบบ 3 สถานะจึงทำให้ขาเป็นได้ทั้งอินพุตและเอาต์พุต



รูปที่ 2.12 วงจรทางด้านเอาต์พุตของ PAL ชนิด L ที่มีการป้อนกลับ

การป้อนกลับของหน่วยความจำเอาต์พุต

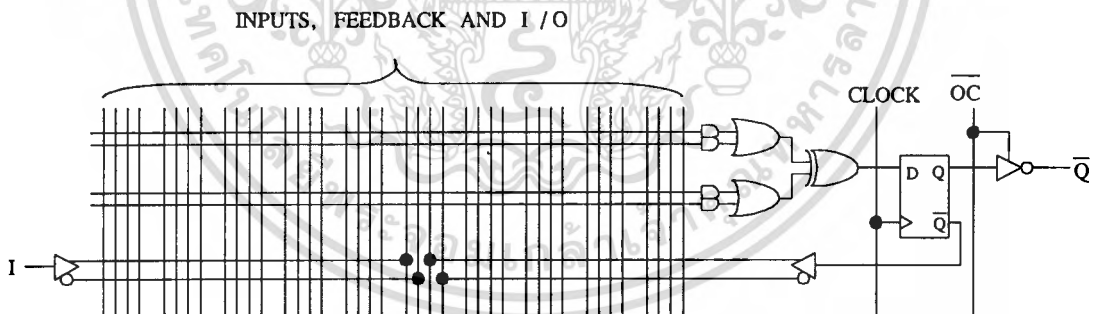
จุดสูงสุดของไอซีตระกูล PAL อีกอย่างหนึ่งคือ เอาต์พุตจะมีหน่วยความจำป้อนกลับ ซึ่งค่าก็จะถูกเก็บในดีฟลิปฟล็อป ข้อมูลจะส่งมาเมื่อมีขอบขาขึ้นของสัญญาณนาฬิกาดังรูปที่ 2.13 ขาเอาต์พุตของ D ฟลิปฟล็อป สามารถเป็นเอาต์พุตได้โดยการเปิด เกตแบบ 3 สถานะ เมื่อทำการเปิดแล้วก็จะให้ค่าของ ดีฟลิปฟล็อป ส่งไปเป็นเอาต์พุตขา Q จะป้อนกลับไปยังทางด้านอินพุต วิธีการนี้จะทำให้ PAL สามารถจำสถานะของวงจรได้ ซึ่งจะช่วยให้ผู้ออกแบบสะดวกในการใช้ออกแบบสำหรับการนับขึ้น นับลง หรือเลื่อน ข้อมูล ซึ่งหน่วยความจำใน PAL จะทำงานในอัตราสูงถึง 25 MHz ทางด้านเอาต์พุต Q ของดีฟลิปฟล็อป ถูกกั้นด้วยอินเวอร์เตอร์ (Inverter) ซึ่งสามารถควบคุมการผ่านสัญญาณออกมาที่เอาต์พุตได้ โดยที่อินเวอร์เตอร์ตัวนี้จะมีความต้านทานสูงเมื่อขา $\overline{OC}$ หยุดการอินาเบิล



รูปที่ 2.13 วงจรทางด้านเอาต์พุตของ PAL ชนิด R

PAL ชนิดรีจิสเตอร์แบบเอกซ์คลูซีฟ

อุปกรณ์ PAL ที่มีรีจิสเตอร์แบบเอกซ์คลูซีฟ ผลบวกของผลคูณคือ ตัวชี้ค่า ซึ่งเอกซ์คลูซีฟจะอยู่ที่อินพุตของ D ฟลิปฟลอป เพื่อรวมผลของออร์เกตทั้งสองตัว ดังรูปที่ 2.14 หน่วยความจำจะแปลงอยู่ในรูปของเอกซ์คลูซีฟ ซึ่งจะทำได้สามารถจำสถานะการทำงาน ซึ่งจะใช้มากในวงจรนับ และวงจรซีเควนเชียลในสถานะต่างๆ



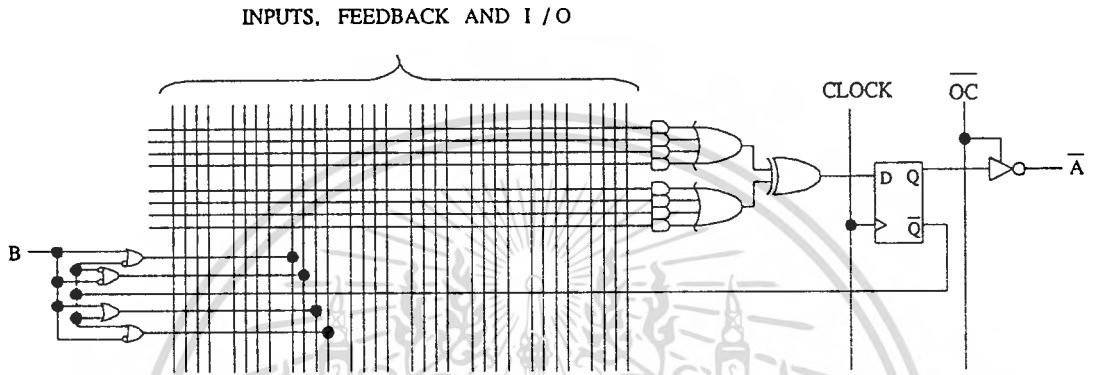
รูปที่ 2.14 วงจรทางด้านเอาต์พุตของ PAL ชนิด X

PAL ชนิดรีจิสเตอร์ที่สามารถคำนวณทางคณิตศาสตร์

เอาต์พุตชนิด A จะซับซ้อนกว่า ชนิด X ขึ้นไปอีกดังรูปที่ 2.15 ซึ่งสามารถนำมาใช้ในการคำนวณได้ โดยมีชุดเกตอยู่หลายตัวทางอินพุต ซึ่งรับสัญญาณจากฟลิปฟลอปด้วยการจัดวงจรเช่นนี้ จึงสามารถบวกอินพุตเข้ากับเอาต์พุตได้ เช่น $I+Q$, $I+Q$, $\bar{I}+Q$, $\bar{I}+Q$

ซึ่งผลการบวกจะป้อนเข้าทางอินพุตอีกครั้งเกตที่ใช้ในการป้อนกลับจะทำให้เกิดสมการบูลีนไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขึ้นมาอีก 16 สมการ ซึ่งจะจัดรูปแบบให้เป็นแบบแผนผังคานอร์ (Karnaugh map) ดังรูปที่ 2.16 ส่วนลักษณะวงจรจะอยู่ในรูปที่ 2.17 ซึ่งจะแสดงถึงวิธีที่ PAL สามารถนำค่ามาโปรแกรมให้ได้การกระทำใหม่อีก 16 สมการ ซึ่งการกระทำนี้มีความจำเป็นในขั้นตอนกระทำทางคณิตศาสตร์

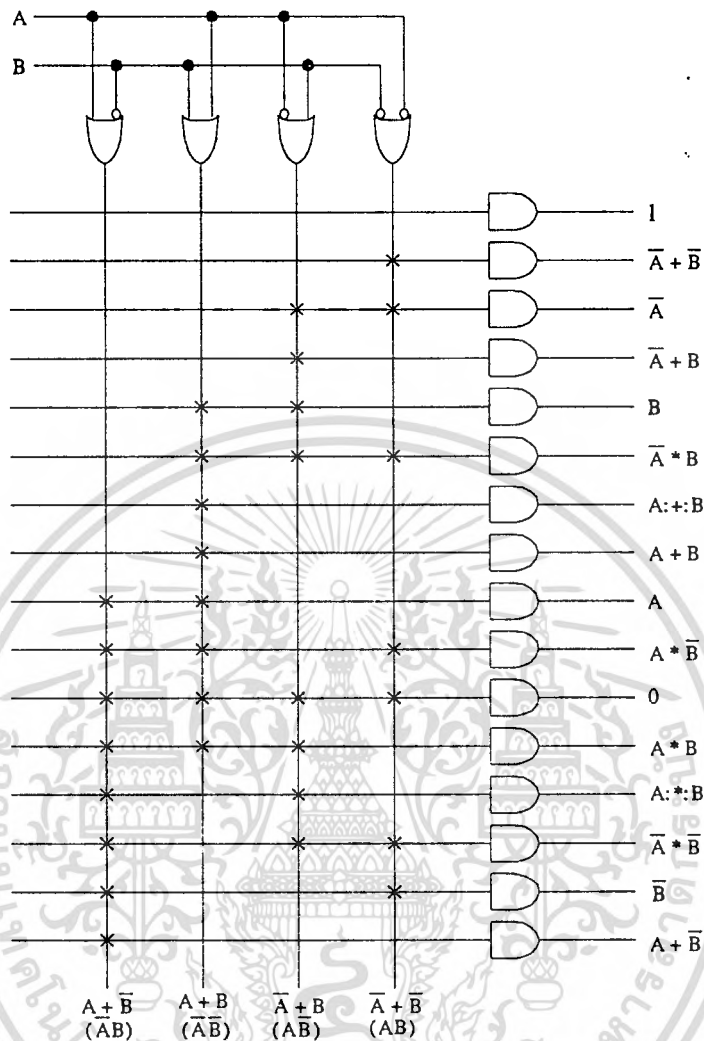


รูปที่ 2.15 วงจรทางด้านเอาต์พุตของ PAL ชนิด A

$(\bar{A} + B), (\bar{A} + \bar{B})$
 $(A + \bar{B}), (A + B)$

	--	-X	XX	X-
--	1	$\bar{A} + \bar{B}$	$\bar{A}$	$\bar{A} + B$
-X	$A + B$	$A + B$	$\bar{A} * B$	B
XX	A	$A * \bar{B}$	0	$A * B$
X-	$A + \bar{B}$	$\bar{B}$	$\bar{A} * \bar{B}$	$A * B$

รูปที่ 2.16 แผนผังคานอร์ของสมการที่เกิดจากการป้อนกลับ



รูปที่ 2.17 ลักษณะวงจรที่ PAL สามารถนำค่ามาโปรแกรมใหม่ 16 สมการ

การพัฒนาของ PAL

ในปี 1985 เบอร์ PAL ได้มีการประดิษฐ์ขึ้นมาใหม่ ซึ่งได้รวบรวมลงในตระกูล PAL มีส่วนประกอบ คือ

- รีจิสเตอร์ ทำหน้าที่เป็นโหลดชุดแรก ใช้สำหรับตรวจสอบการทำงาน
- เอาต์พุตที่สามารถโปรแกรมให้ทำงานที่ลอจิก "1" หรือ "0"
- จำนวนของเส้นอินพุตสามารถเปลี่ยนโดยผู้ใช้
- รีจิสเตอร์ใช้รวมข้อมูลทางเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- สัญญาณอะซิงโครนัส (Asynchronous) , เซต (Set) , รีเซต (Reset) , และเอาต์พุต
อินาเบิล (Output enable)

เทคโนโลยีของ PAL

วงจร PAL เป็นสิ่งประดิษฐ์ซึ่งให้หลักการทดลองแบบ ทีทีแอล ช็อตกี 2 ขั้ว (TTL Schottky bipolar) ซึ่งจะมีกรรมวิธีในการโปรแกรมฟิวส์ที่เชื่อมต่อใน PROM วงจรคอมมอนคอลเลคเตอร์ (Emitter follower) ของชนิด NPN ใช้ในการทำแอนด์ ส่วนชนิด PNP ใช้ทำเป็นอินพุตเพราะมีความต้านทานสูง โดยทั่วไป PAL จะมีโปรเพเกชันดีเลย์ไทม์ (propagation Delay time) น้อยกว่า 25 ns.

การเก็บรักษาข้อมูลใน PAL

วงจรที่ใช้ในการโปรแกรม และส่วนที่ใช้การทดสอบจะแสดงตามโครงสร้างของ PAL ส่วนที่ใช้ในการป้องกันจะอยู่ที่ฟิวส์ตัวสุดท้าย ซึ่งสามารถระเบิดฟิวส์ เพื่อต้องการไม่ให้สามารถดึงข้อมูลขึ้นมาได้ การกระทำอย่างนี้สามารถใช้งาน เพื่อป้องกันการลอกเลียนแบบโปรแกรมที่ออกแบบไว้ใน PAL

ประโยชน์ของการใช้ PAL

PAL สามารถจะนำลอจิกที่ได้ออกแบบมาแล้วมาทำการโปรแกรมลงในตัวซึ่งสามารถเลือกใช้งานตามความต้องการตามเบอร์ที่มีอยู่ในตระกูล PAL

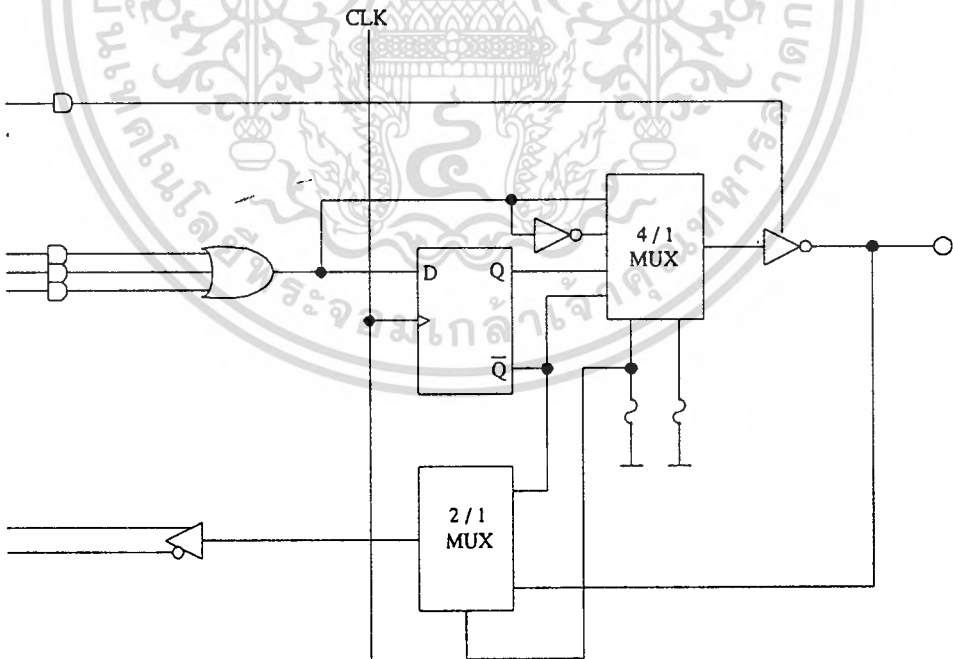
- สามารถโปรแกรมแทนการเปลี่ยนแปลงลอจิก
- สามารถลดจำนวนไอซีที่จะใช้งานได้ และสามารถจำลองการทำงานได้
- สามารถลดจำนวนไอซีลงได้น้อยถึง 4 ต่อ 1 ของวงจรเก่า
- ประหยัดพื้นที่ในวงจร
- มีความเร็วสูง มีเวลาในการหน่วงเพียง 15 ns โดยทั่วไป
- การโปรแกรมเหมือนมาตรฐานของ PROM
- ส่วนเอาต์พุตมีสถานะแบบ 3 สถานะ

สำหรับอุปกรณ์ PAL ที่ได้พัฒนาต่อมาจะมีโครงสร้างของเอาต์พุต เป็นแบบแมโครเซลล์ (Macrocell) เช่น AmPAL22V10 ที่มี 24 ขา ดังแสดงในรูปที่ 2.18 โครงสร้าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของ PAL แบบนี้ประกอบด้วย 22 อินพุต และ 10 อินพุต ซึ่งที่เอาต์พุตนี้จะเลือกให้ โปรแกรม ให้เป็นแบบที่มีรีจิสเตอร์แอกทีฟต่ำหรือสูงได้ หรือเลือกให้เป็นแบบเชิงผลสมแอกทีฟต่ำ หรือ สูง นั่นคือ ทำให้ผู้ใช้สามารถเลือกที่จะใช้แทน PAL ได้หลายชนิดโดยใช้ PAL22V10 เพียง ตัวเดียวมีผลทำให้การออกแบบสะดวกสำหรับผู้ใช้และทำให้การออกแบบวงจรไม่ยุ่งยาก ซับซ้อน

ในปัจจุบันนี้ ได้มีการนำเอาอุปกรณ์ PAL มาใช้งานแทนวงจรลอจิก อย่างกว้างขวาง เนื่องจากว่าภายในโครงสร้างของตัว PAL จะประกอบไปด้วยเกทหลายตัวอยู่ใน อุปกรณ์ PAL เพราะฉะนั้นจึงเป็นการลดจำนวนเกทที่ใช้ลงไปได้มาก ทำให้วงจรที่ออกแบบมี ขนาดเล็ก กะทัดรัด ซึ่งเป็นที่นิยมมากสำหรับนักออกแบบวงจรทางด้านเชิงเลข ต่อมาก็ได้มีการผลิต PAL รุ่นใหม่ที่สามารถทำการลบได้ คือ ทำการโปรแกรมซ้ำได้หลายๆ ครั้ง รวมทั้งยังมี ซอฟต์แวร์ช่วยสนับสนุนในการออกแบบอีกด้วย ในการโปรแกรม PAL จะใช้เครื่องโปรแกรม (PAL Programmer) ซึ่งเครื่องนี้จะทำการต่อพิวส์และการทำให้พิวส์ขาด ตามข้อมูลที่เป็นรูป แบบของ การโปรแกรม



รูปที่ 2.18 รูปโครงสร้างภายในของ PAL ที่เป็นแบบแมโครเซลล์

วงจร PAL สำหรับการใช้งาน

ไอซีในตระกูล PAL และคุณสมบัติของแต่ละเบอร์ได้แสดงในตารางที่ 2.2 ซึ่งจะช่วยให้ผู้ออกแบบสามารถเลือกใช้งานได้ตามความสะดวก เพื่อจะเลือก PAL เบอร์ที่ดีที่สุดในการนำไปประยุกต์ใช้งาน

PART No.	INPUT	OUTPUT	PROG. I/O'S	FEEDBACK REGISTER	OUTPUT POLARITY	FUNCTION	PERFORMANCE			
							STD	A	-2	A-2
10H8	10	8			AND-OR	AND-OR Gate Array	X		X	
12H6	12	6			AND-OR	AND-OR Gate Array	X		X	
14H4	14	4			AND-OR	AND-OR Gate Array	X		X	
16H2	16	2			AND-OR	AND-OR Gate Array	X		X	
16C1	16	2			BOTH ¹	AND-OR/ NOR Gate Array	X		X	
10L8	10	8			AND-NOR	AND-OR Invert Gate Array	X		X	
12L6	12	6			AND-NOR	AND-OR Invert Gate Array	X		X	
14L4	14	4			AND-NOR	AND-OR Invert Gate Array	X		X	
16L2	16	2			AND-NOR	AND-OR Invert Gate Array	X			
12L10	12	10			AND-NOR	AND-OR Invert Gate Array	X			
14L8	14	8			AND-NOR	AND-OR Invert Gate Array	X			
16L6	16	6			AND-NOR	AND-OR Invert Gate Array	X			
18L4	18	4			AND-NOR	AND-OR Invert Gate Array	X			
20L2	20	2			AND-NOR	AND-OR Invert Gate Array	X			
20C1	20	2			BOTH ¹	AND-OR/ NOR Gate Array	X			
16L8	10	2	6		AND-NOR	AND-OR Invert Gate Array		X		X X
20L8	14	2	6		AND-NOR	AND-OR Invert Gate Array		X		
20L10	12	2	8		AND-NOR	AND-OR Invert Gate Array	X			
16R8	8	8		8	AND-NOR	AND-OR Invert Gate Array w/ Regs		X		X X
16R6	8	6	2	6	AND-NOR	AND-OR Invert Gate Array w/ Regs		X		X X
16R4	8	4	4	4	AND-NOR	AND-OR Invert Gate Array w/ Regs		X		X X
20R8	12	8		8	AND-NOR	AND-OR Invert w/ Rege		X		
20R6	12	6	2	6	AND-NOR	AND-OR Invert w/ Rege		X		
20R4	12	4	4	4	AND-NOR	AND-OR Invert w/ Rege		X		
20X10	10	10		10	AND-NOR	AND-OR-XOR Invert w /R ege	X			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับตารางที่ 2.2 แสดงคุณสมบัติของ PAL แต่ละเบอร์ที่ใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PART No.	INPUT	OUTPUT	PROG. I/O'S	FEEDBACK REGISTER	OUTPUT POLARITY	FUNCTIONS	PERFORMANCE				
							A-0	A-1	A-2	A-3	A-4
20x8	10	8	2	8	AND-NOR	AND-OR-XOR Invert w /R ege	X				
20X4	10	4	6	4	AND-NOR	AND-OR-XOR Invert w / Rege	X				
16X4	8	4	4	4	AND-NOR	AND-OR-XOR Invert w / Rege	X				
16A4	8	4	4	4	AND-NOR	AND-CARRY-OR-XOR Invert w / Rege	X				
16P8	10	2	6		PROG ²	AND-OR Gate Array		X			
16RP8	8	8		8	PROG ²	AND-OR Gate Array w/ Regs		X			
16RP6	8	6	2	6	PROG ²	AND-OR Gate Array w/ Regs		X			
16RP4	8	4	4	4	PROG ²	AND-OR Gate Array w/ Regs		X			
20RA10	10		10 ³	10 ³	PROG ²	Asynchronous Gate Array	X				
20RS10	10				PROG ²	AND-OR Gate Array w/ Regs		X			
20RS8	10		2	10	PROG ²	AND-OR Gate Array w/ Regs		X			
20RS4	10		6	8	PROG ²	AND-OR Gate Array w/ Regs		X			
20S10	10		10	4	PROG ²	AND-OR Gate Array		X			
32R16	16	6 ³		6 ³	PROG ²	AND-OR Gate Array w/ Regs		X			
64R32	32	32 ³		32 ³	PROG ²	AND-OR Gate Array w/ Regs		X			

ตารางที่ 2.2 คุณสมบัติของ PAL แต่ละเบอร์ (ต่อ)

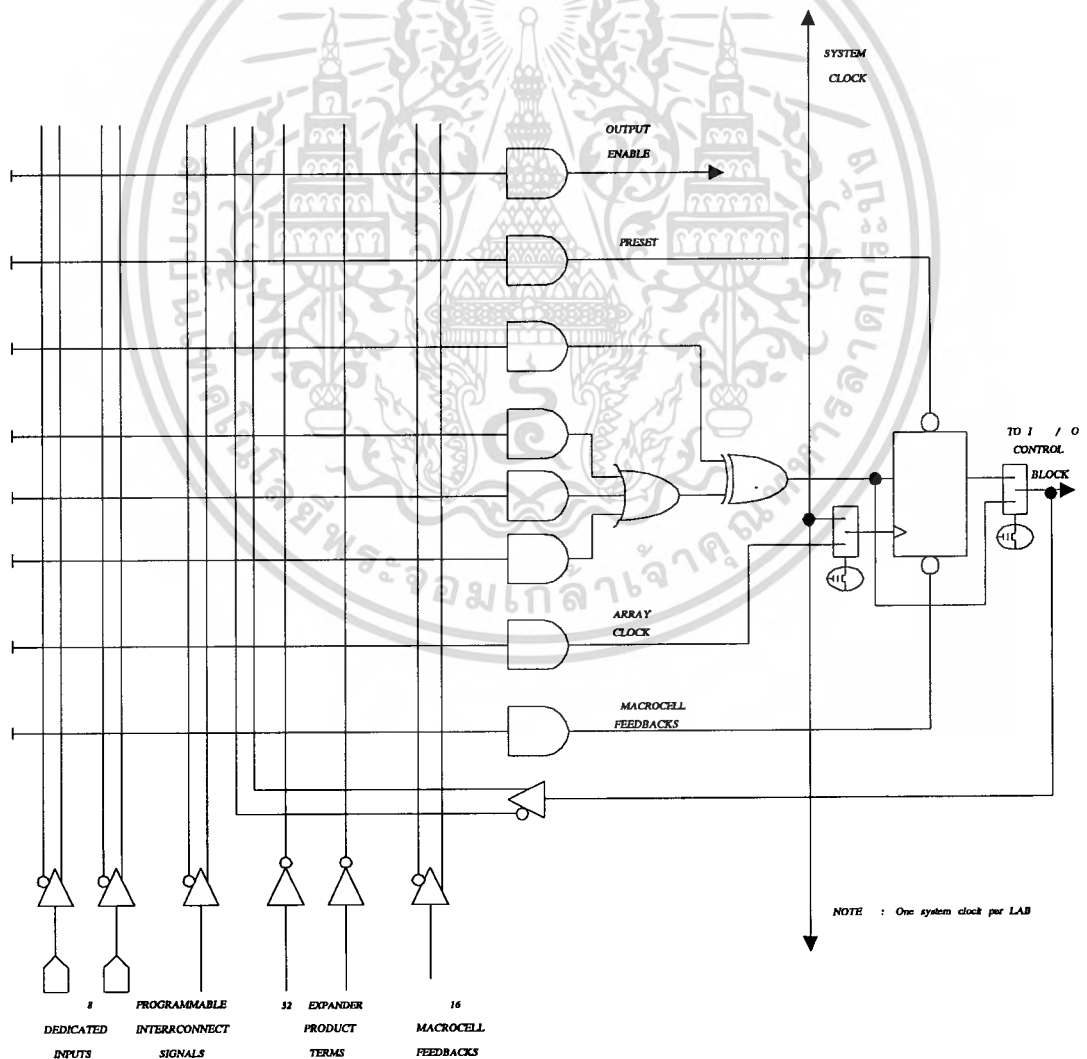
- หมายเหตุ 1 หมายถึง การจำลองเอาต์พุต แอนด์-ออร์ และ แอนด์-นอร์
- 2 หมายถึง การโปรแกรมแอกทีฟ “1” หรือ แอกทีฟ “0” ของ แอนด์-ออร์ หรือ แอนด์-นอร์
- 3 หมายถึง เอาต์พุตมีรีจิสเตอร์หรือไม่

2.3.2 อุปกรณ์ EPLD

อุปกรณ์ PLD ในยุคแรกๆ ตั้งแต่ PLD เบอร์แรกเมื่อ 20 ปีมาแล้ว มีการโปรแกรมโดยตัดฟิวส์บางส่วนที่ไม่ต้องการออกจากฟิวส์อารีย์ ในตัวอุปกรณ์ ทำให้การโปรแกรม PLD ทำได้เพียงครั้งเดียว ซึ่งพบว่าวงจรที่โปรแกรมแล้วใช้งานไม่ได้ตามต้องการ เนื่องจากออกแบบผิดพลาดไป จะต้องโปรแกรมอุปกรณ์ตัวใหม่เหมือนกับการใช้ PROM (Programmable Read Only Memory) คือแต่ละตัวสามารถโปรแกรมข้อมูลได้เพียงครั้งเดียว

เอกสาร EPLD หรือ Erasable Programmable Logic Device เป็นอุปกรณ์สามารถโปรแกรมได้ที่ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สามารถลบและกระทำการโปรแกรมใหม่ได้ EPLD เป็นอุปกรณ์ที่โปรแกรมทางด้านแอนด์ ผู้ใช้สามารถลบโปรแกรมได้โดยใช้แสงอุลตราไวโอเลตเหมือนกับ EPROM บริษัท Altera Corporation ประเทศสหรัฐอเมริกา ได้เริ่มทำการผลิต EPLD ขึ้นในปี 1984 เป็น PLD ที่ใช้เหมือนกับ CMOS EPROM คือ ใช้มอสทรานซิสเตอร์ (MOS Transister) เชื่อมต่อระหว่าง สัญญาณอินพุต กับจุดที่ต้องการแทนการใช้ฟิวส์ วงจรภายในของอุปกรณ์ EPLD แสดงใน รูปที่ 2.19 ทำให้สามารถโปรแกรมการต่อวงจรภายในอุปกรณ์ด้วยการจ่ายกระแสไฟฟ้าตาม ขนาดที่กำหนดเข้าไปในตัวอุปกรณ์และลบได้โดยใช้แสงอุลตราไวโอเลตฉายผ่านช่อง หน้าต่างกระจก เข้าไปกระทบในตัวชิพของอุปกรณ์ ทำให้มอสทรานซิสเตอร์นำกระแสได้อีก ครั้งเพื่อให้สามารถโปรแกรมใหม่ได้



รูปที่ 2.19 วงจรภายในอุปกรณ์ EPLD ใช้มอสทรานซิสเตอร์แทนฟิวส์

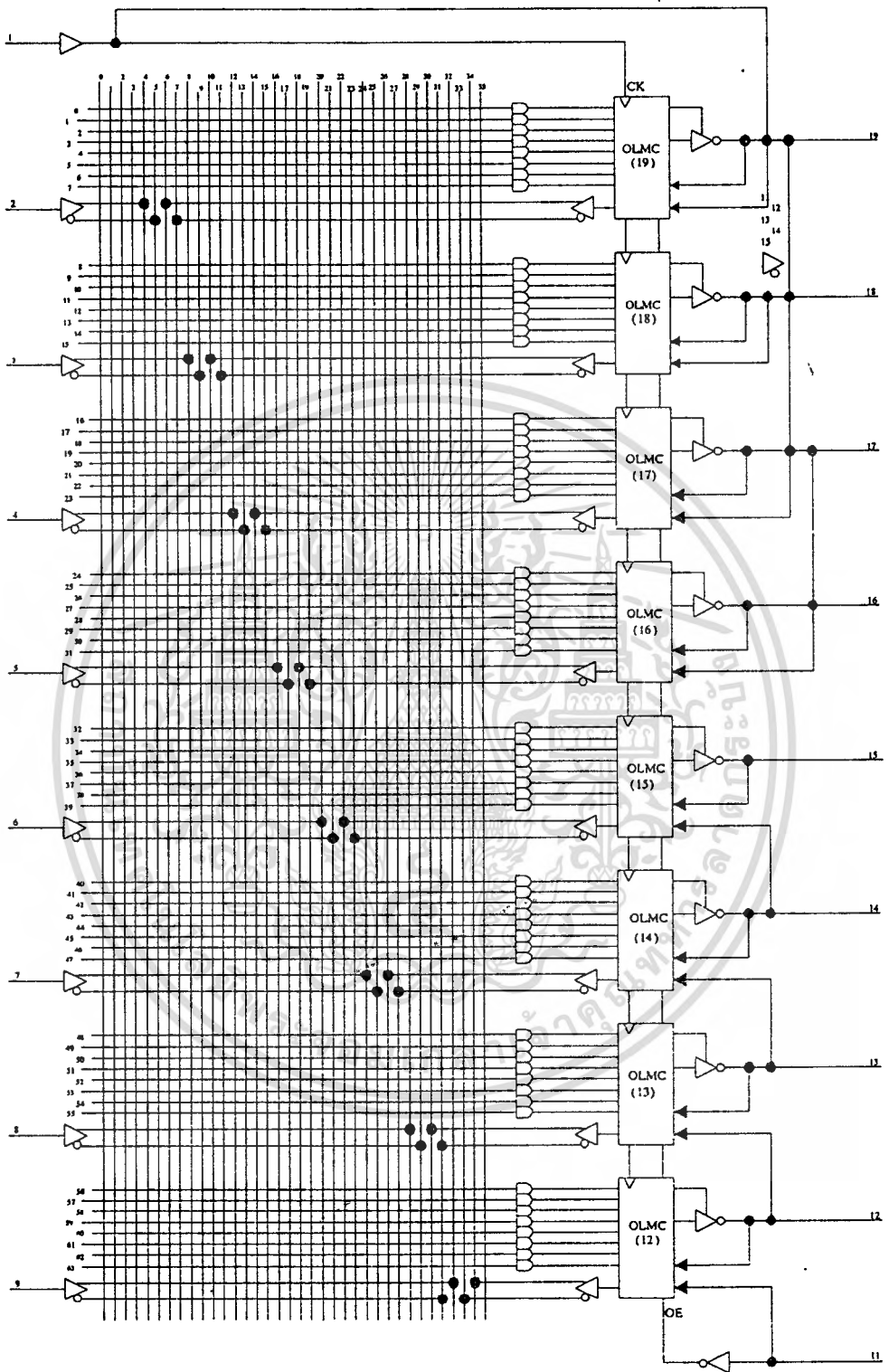
อุปกรณ์ EPLD นอกจากจะลบด้วยแสงอัลตราไวโอเลตได้แล้วยังมีรุ่นที่สามารถลบ และ โปรแกรมด้วยกระแสไฟฟ้า เรียกว่า EEPLD เช่น GAL16V8 , GAL20V8 ของบริษัท Lattice Semiconductor เป็น PLD ลบได้ที่มีฟังก์ชันใกล้เคียงกับ 16V8 (16อินพุต/ 8 เอาต์พุต) และ 20V8 ซึ่งเป็นแมโครเซลล์ PEEL 18CV8 ของบริษัท International CMOS Technology

2.3.3 อุปกรณ์ GAL

GAL EECMOS เป็นอุปกรณ์ที่รวมคุณสมบัติเด่นของ CMOS พร้อมด้วยเทคโนโลยีการลบ และ การโปรแกรมเทคโนโลยีการโปรแกรมนี้นี้ มีประโยชน์ต่อกระบวนการจัดเตรียมลอจิก ในการ ออกแบบ และ การสร้างลอจิกใหม่ การใช้ทั้ง 2 แบบจะใช้เครื่องหมายบอกแสดงความต้องการใช้งาน GAL 16V8 แบบ 20 ขา มีขาสำหรับโปรแกรม เอาต์พุตลอจิกแมโครเซลล์ 8 ขา และแต่ละขามีคุณสมบัติตามการใช้งาน GAL 16V8 มีความสามารถเหมือนกับการสร้างแบบจำลอง , มีฟังก์ชัน ภายใน/ ฟีลด์แมป/ ต่อเข้ากับ อุปกรณ์ , มีโครงสร้างเหมือนกับ PAL 20 ขา (ยกเว้นชนิดเอกซ์คลูซีฟออร์)

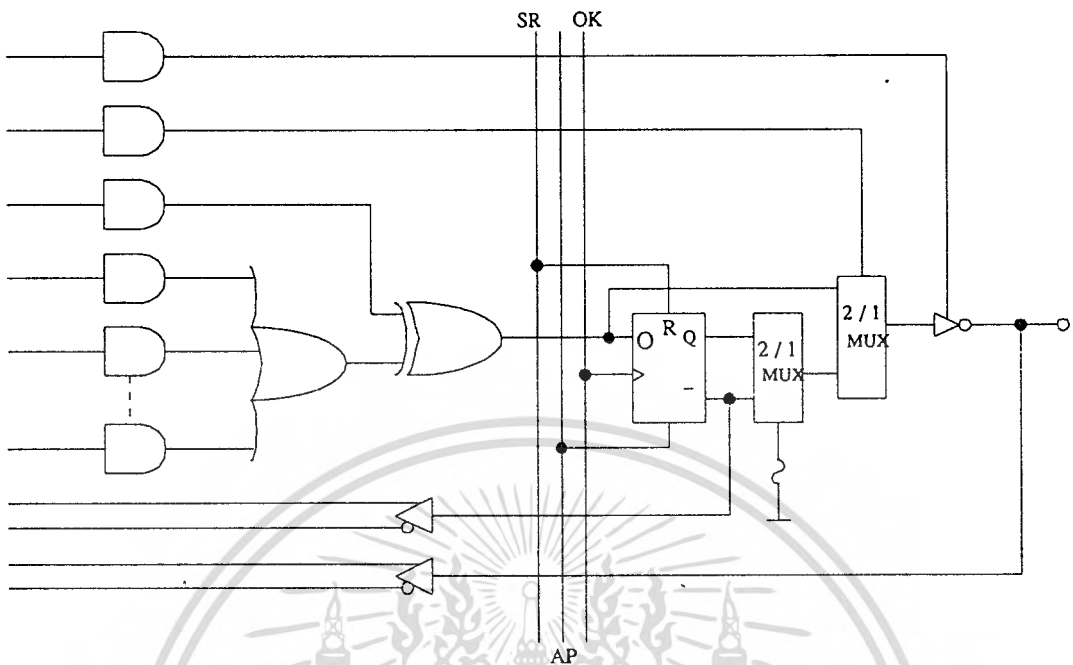
เอาต์พุตลอจิกแมโครเซลล์ (OUTPUT LOGIC MACROCELL : OLMC)

พิจารณาเกี่ยวกับรูปร่างลักษณะเอาต์พุตลอจิกเซลล์ เอาต์พุตของแอนดออาร์ยเป็นสนามของเอาต์พุตลอจิกเซลล์ แต่ละเอาต์พุตจะแตกต่างกันในการเซตให้แอกทีฟ “1” หรือ แอกทีฟ “0” และ เซตลักษณะว่าเป็น การรวมกัน (อะซิงโครนัส) หรือ เป็นรีจิสเตอร์ (ซิงโครนัส) คอมมอนเอาต์พุตอินาเบิล (Common output enable) สามารถต่อร่วมกับเอาต์พุตทั้งหมด หรือ แยกอินพุต หรือ แบ่งเป็นกลุ่มตามลักษณะการใช้งานที่แตกต่างกันแล้วนำมาควบคุมเอาต์พุตอินาเบิล เอาต์พุตแมโครเซลล์ ถูกออกแบบให้สามารถดัดแปลงเอาต์พุตตามสัญญาณที่ต้องการ และ ฟังก์ชันมีความแน่นอน เช่นเดียวกับ PAL 20 ขา



รูปที่ 2.20 โครงสร้างภายในของ GAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.21 แมโครเซลล์ของ GAL

2.2.4 อุปกรณ์ PLD เบอร์ HY16CV8

HY18CV8 เป็นวงจรรวมประเภท EEPLD (MOS Electrically Erasable Programmable Logic Device) มีความสามารถสูง, กินไฟต่ำ, สามารถลบ / โปรแกรมใหม่ได้ และมีโครงสร้างที่ยืดหยุ่น กล่าวคือ ผู้ใช้สามารถเลือกรูปแบบโครงสร้างหนึ่งในหลายๆ แบบ ที่ HY18CV8 ได้เตรียมไว้ HY18CV8 ถูกออกแบบมาจากเทคโนโลยีของซีมอส EEPROM ทำให้ใช้พลังงานน้อยกว่าในขณะที่มีความเร็วเทียบเท่ากับ PLD ประเภทไบโพลาร์ คุณสมบัติการลบ / โปรแกรมใหม่ ได้ทำให้ HY18CV8 มีความสามารถเหนือกว่า PAL ซึ่งเป็นวงจรรวมประเภทที่สามารถโปรแกรมได้เพียงครั้งเดียว คุณสมบัติการเลือกลักษณะเอาต์พุตได้หลายแบบในแต่ละขาเอาต์พุต ทำให้ HY18CV8 มีความสามารถเหนือกว่า GAL ซึ่งเป็นวงจรรวมประเภท ลบ / โปรแกรมใหม่ได้ เช่นกันแต่รูปแบบเอาต์พุตของ GAL ถูกแบ่งออกเป็น โหมด ทำให้มีความยืดหยุ่นน้อยกว่า

สถาปัตยกรรมพื้นฐานของ HY18CV8 คล้ายกับ PLD ทั่วไป คือ ออกแบบมาให้ใช้ประโยชน์จากฟังก์ชันสมการประเภท ผลบวกของผลคูณ (เป็นการออกแบบชนิดที่มีเทอมย่อยเป็นแอนด์ และ รวมเทอมย่อยแต่ละเทอมด้วยออร์ เช่น $Y = AB + CD + \overline{ACD}$ เป็นเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทอมย่อย ที่รวมกันด้วยแอนด์ และ Y คือฟังก์ชันประเภทยผลบวกของผลคูณ) โครงสร้างแบบนี้อาจเรียกได้ว่า โครงสร้างแอนด์กำหนดคอร์ด จากลักษณะโครงสร้างนี้ทำให้ผู้ใช้สามารถสร้างฟังก์ชันเอาต์พุต โดยการเลือกจุดต่อของสัญญาณอินพุต ในอาร์เรย์ ที่ HY18CV8 เตรียมไว้ จากโครงสร้างนี้ทำให้สามารถสร้างฟังก์ชันเอาต์พุต ได้หลากหลาย ขึ้นอยู่กับความต้องการของผู้ใช้แต่ละคน

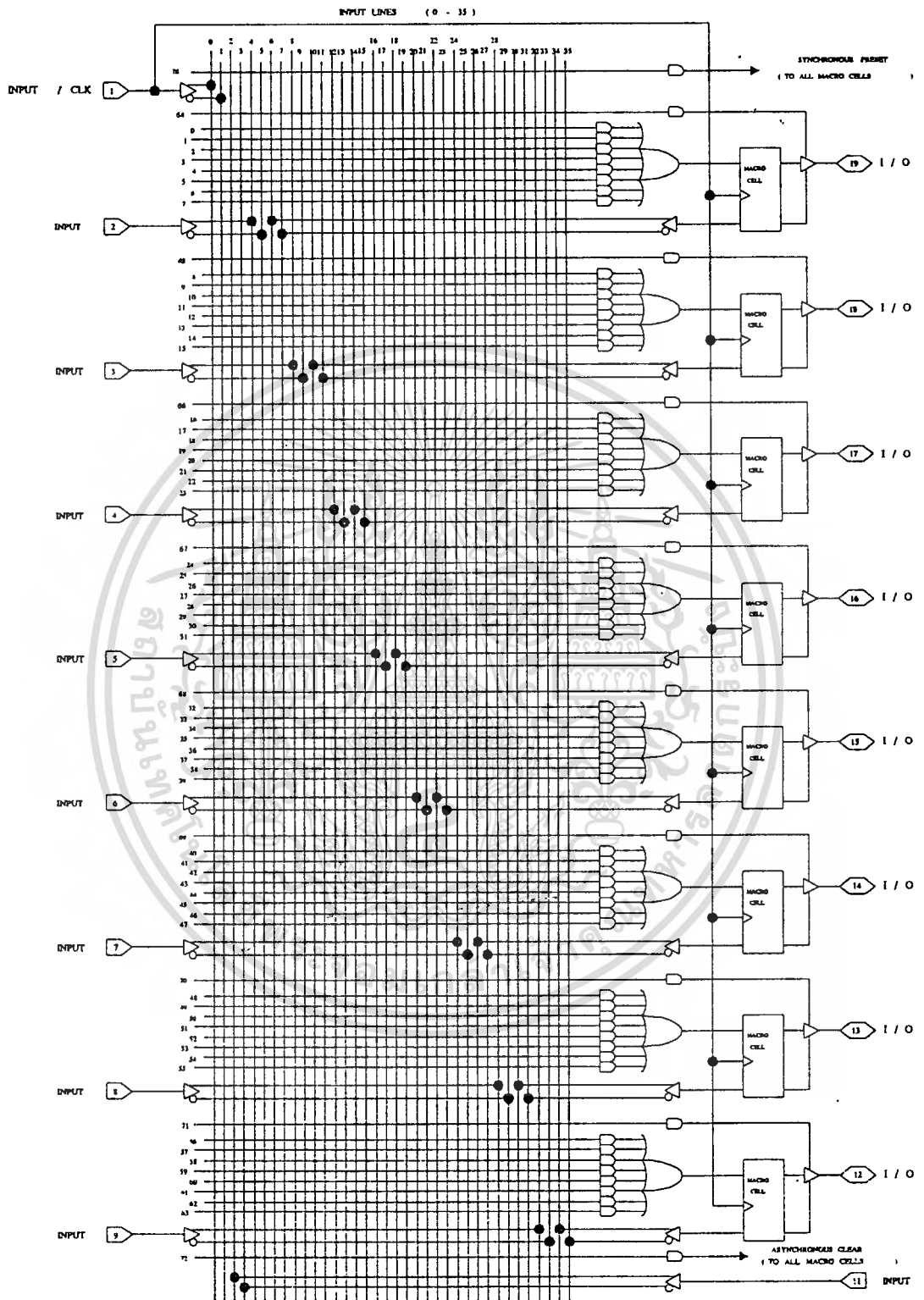
ส่วนประกอบในสถาปัตยกรรมของ HY18CV8 มีจำนวนอินพุตได้ มากที่สุด 18 อินพุต และจำนวนเอาต์พุตมีได้มากที่สุด 18 เอาต์พุต โครงสร้างแอนด์อาร์เรย์ ซึ่งสามารถลบ / โปรแกรมได้ด้วยไฟฟ้า เกิดจาก อินพุต 36 เส้น (อยู่ในแนวตั้งซึ่งได้มาจากสัญญาณอินพุต และ คอมพลิเมนต์ (Complement) ของอินพุต ทั้ง 18) กับ ผลคูณ 74 เทอม (อยู่ในแนวนอน ซึ่งได้มาจาก 1 เทอมของซิงโครนัสพรีเซต (synchronous preset), 1 เทอมของอะซิงโครนัสเคลียร์ (asynchronous clear), 8 เทอมของเอาต์พุตอินาเบิล และอีก 64 เทอม ที่มาจาก 8 เอาต์พุต มี 8 เทอมแอนด์ และแต่ละเทอมแอนด์ต่อเข้ากับฟังก์ชันออร์)

เอาต์พุตจากฟังก์ชันออร์ ถูกต่อเข้ากับแมโครเซลล์ และ ขาอินพุต/เอาต์พุต (I/O) โดยภายในแมโครเซลล์แต่ละตัวสามารถโปรแกรมรูปแบบอิสระต่อกัน (แมโครเซลล์แต่ละชุดไม่จำเป็นต้องโปรแกรมรูปแบบเดียวกัน) รูปแบบต่างๆ ที่ HY18CV8 เตรียมไว้ทั้งหมด 12 รูปแบบ

เอาต์พุตของแมโครเซลล์ สามารถป้อนกลับไปยังโครงสร้างอะเรย์ หรือ เป็นสัญญาณเอาต์พุตส่งออกไปทางขาอินพุต/เอาต์พุต ก็ได้รูปแบบในแมโครเซลล์ ถูกแบ่งออกเป็น 3 แบบใหญ่ ๆ คือ

- อินพุต/เอาต์พุต 2 ทิศทาง (สัญญาณป้อนกลับส่งมาจากขาเอาต์พุตโดยตรง)
- การรวมสัญญาณป้อนกลับ (สัญญาณป้อนกลับส่งมาจากเอาต์พุตของเทอมออร์)
- รีจิสเตอร์ป้อนกลับ (สัญญาณป้อนกลับส่งมาจากเอาต์พุตของ ดีฟลิปฟลอป)

ให้สังเกตว่าเมื่ออินพุต/เอาต์พุตใดเลือกรูปแบบของแมโครเซลล์เป็นแบบอินพุต/เอาต์พุต 2 ทิศทาง ขาอินพุต/เอาต์พุตนั้นสามารถทำหน้าที่เป็นทางเข้าของสัญญาณอินพุตจากภายนอกได้โดยการโปรแกรมให้ค่าลอจิกของเทอมเอาต์พุตอินาเบิล ที่สัมพันธ์กับแมโครเซลล์มีค่าเป็น "0" ลักษณะเช่นนี้ทำให้สามารถมีอินพุตได้มากขึ้น (สามารถใช้ขาอินพุต/เอาต์พุต เป็นอินพุตได้)



รูปที่ 2.22 โครงสร้างภายในของ HY18CV8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดโครงสร้างของ แอนด์อาร์รี่ แสดงไว้ในรูปที่ 2.22 จากรูปจะเห็นได้ว่า HY18CV8 ประกอบได้ด้วย

อินพุต 36 เส้น :

10 สัญญาณอินพุต และ 10 สัญญาณคอมพลีเมนต์อินพุต

8 สัญญาณอินพุตป้อนกลับ และ 8 สัญญาณคอมพลีเมนต์อินพุตป้อนกลับ

74 เทอมผลคูณ (Product term) :

64 เทอม (8×8 เทอมของผลบวกของผลคูณ)

8 ฟังก์ชันเอาต์พุตอีนาเบิล (1 เทอม/ 1 เอาต์พุต)

1 สัญญาณซิงโครไนส์ฟรีเซต

1 สัญญาณอะซิงโครไนส์เคลียร์

จากรูป อินพุต 36 เส้น (0-35) จะอยู่ในแวนนอน และ ผลคูณ 74 เทอม (0-37) จะอยู่ในแวนดั่ง ในแต่ละจุดที่เส้นอินพุตตัดกับเทอมผลคูณ เป็นจุดที่มีเซลล์หน่วยความจำประเภท EEPROM (หน่วยความจำแบบที่สามารถลบ / โปรแกรมได้ด้วยไฟฟ้า) ซึ่งเป็นตัวกำหนดว่าจะให้สัญญาณที่จุดตัดนั้นต่อเข้าด้วยกันหรือไม่ การต่อกันของจุดตัดเปรียบเสมือนการนำเส้นอินพุต นั้นมาเข้าลอจิกแอนด์ กับ เส้นอินพุต อื่น ๆ ที่ตัดกับ เทอมผลคูณ นั้น ๆ สรุปราย ๆ คือ เทอมผลคูณ เปรียบเสมือนกับแอนด์เกตขนาด 36 อินพุตนั่นเอง

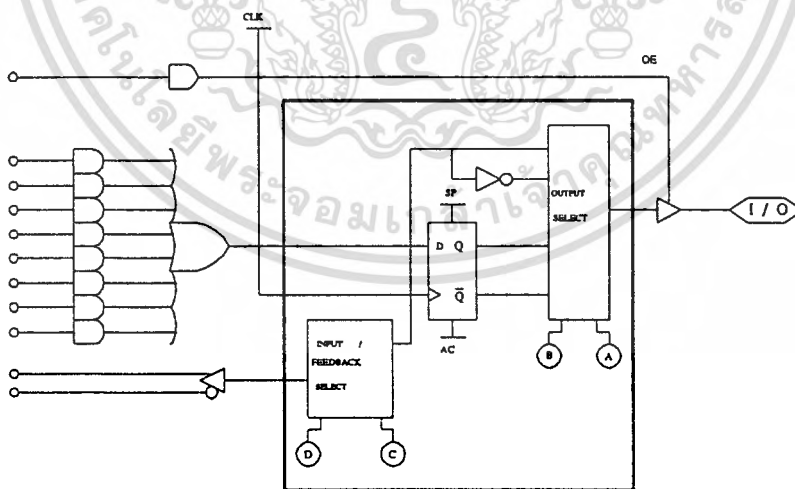
ระหว่างการโปรแกรม HY18CV8 จากเครื่องโปรแกรม อันดันดับแรกสุดเซลล์หน่วยความจำ ชนิด EEPROM ทั้งหมดจะถูกลบเพื่อเปิดจุดตัดทั้งหมดของเส้นอินพุต และ เทอมผลคูณ เฉพาะจุดตัดบางจุดเท่านั้นจะถูกโปรแกรมให้ต่อกันโดยจุดเหล่านั้นผู้ใช้ (ผู้ออกแบบ) เป็นคนกำหนด

จากรูปที่ 2.22 จะเห็นได้ว่ามีลอจิกอาร์รี่ อยู่ 64 เทอมผลคูณ มาจากกลุ่มของ 8 เทอมผลคูณ ที่ป้อนเข้าออร์เกต ซึ่งมีทั้งหมด 8 ตัว ในแต่ละเอาต์พุตของออร์เกต ถูกต่อเข้ากับวงจรรวมที่เรียกว่า แมโครเซลล์ ทำให้สามารถสร้างรูปแบบต่างๆ กันได้ และในแต่ละขาอินพุต/เอาต์พุต ยังมีเทอมที่สามารถควบคุมการอีนาเบิล ให้สัญญาณเอาต์พุตส่งออกมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อินพุต/เอาต์พุต แมโครเซลล์ และ การทำงานของเอาต์พุตอื่นาเบิล

คุณสมบัติที่ยึดหยุ่นของ HY18CV8 ส่วนหนึ่งมาจากแมโครเซลล์ และสัญญาณควบคุม เอาต์พุตที่อิสระต่อกัน รูปร่างภายในแมโครเซลล์ แสดงไว้ในรูปที่ 2.23 (ภายในกรอบสี่เหลี่ยม คือ แมโครเซลล์) ภายในแมโครเซลล์ ประกอบด้วย ดีฟลิปฟลอป และ มัลติเพล็กซ์เซอร์ (multiplexer) อีก 2 ตัว ดีฟลิปฟลอป มีการทำงานเหมือนกับ ดีฟลิปฟลอปใน TTL มาตรฐานทั่วไป คือสัญญาณอินพุตที่ขา D จะถูกเก็บสถานะ (latch) ในจังหวะขอบขาขึ้นของ สัญญาณนาฬิกา (การเปลี่ยนสถานะลอจิกจาก “0” เป็น “1”) เอาต์พุตที่ Q และ $\bar{Q}$ ได้เช่นกัน เมื่อเอาต์พุตของเทอมอะซิงโครนัสเคลียร์ มีค่าลอจิก “1” เอาต์พุต Q จะมีค่าลอจิกเป็น “0” ในทันทีทันใด โดยไม่สนใจสัญญาณนาฬิกา และเมื่อเอาต์พุตของเทอมซิงโครนัส 프리เซต มีค่า ลอจิก “1” เอาต์พุต Q จะถูกเซตเป็น “1” ในจังหวะขอบขาขึ้นของสัญญาณนาฬิกาถ้าสัญญาณ ทั้งคู่เกิดมีค่าลอจิกเป็น “1” พร้อมกัน ลำดับความสำคัญของสัญญาณอะซิงโครนัสเคลียร์ จะมีค่าสูงกว่าในขณะที่ เริ่มป้อนไฟเลี้ยงให้ HY18CV8 ฟังก์ชันอะซิงโครนัสเคลียร์จะ ทำงานโดยอัตโนมัติ เพื่อให้ค่าแรกเริ่มของ Q ในทุก ๆ แมโครเซลล์ มีค่าลอจิกเป็น โลว์มัลติเพล็กซ์ 2 ตัว



รูปที่ 2.23 แมโครเซลล์ของ HY18CV8

ในแต่ละแมโครเซลล์ ถูกควบคุมโดยบิตโปรแกรมของ EEPROM (A, B, C และ D) เพื่อกำหนดรูปแบบใดรูปแบบหนึ่งของแมโครเซลล์จากรูปแบบที่เป็นไปได้ทั้งหมด 12 รูปแบบ ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งบิตโปรแกรมของ EEPROM ในแต่ละเซลล์จะเป็นอิสระต่อกัน ทำให้สามารถโปรแกรมรูปแบบของแต่ละเมโมรีเซลล์ได้โดยอิสระต่อกัน

การป้องกัน

HY18CV8 ได้เตรียมบิต EEPROM พิเศษขึ้นสำหรับใช้ในการรักษาความลับของวงจรภายใน คุณสมบัตินี้ทำให้ป้องกันการลอกเลียนแบบได้ เมื่อบิตนี้ถูกเซต (เซตโดยเครื่องโปรแกรม)จะมีผลทำให้ไม่สามารถอ่านโครงสร้างภายในที่ถูกโปรแกรมไว้จนกว่าจะทำการลบวงจรรวมตัวนั้นอีกครั้ง

เทคโนโลยีมอส EEPROM

HY18CV8 ถูกออกแบบมาให้ สามารถลบ / โปรแกรมได้ มากถึง 1000 ครั้ง และโครงสร้างอะเรย์ที่โปรแกรมไว้สามารถอยู่ได้นานถึง 10 ปี เมื่อนำไปใช้งานในช่วงอุณหภูมิที่เหมาะสม

HY18CV8 สามารถใช้แทน PAL ได้เกือบทุกเบอร์ (เนื่องจาก PAL บางเบอร์มีขนาด 24 ขา จึงทำให้ไม่สามารถแทนได้ทั้งหมด) ดังแสดงไว้ในตารางที่ 2.3

OUTPUT TYPE	PART NUMBER AND I/O CAPACITY					
Combinatorial - High	10H8	12H6	14H4	16H2	16H8	16HD8
Combinatorial - Low	10L8	12L6	14L4	16L2	16L8	16LD8
Combinatorial -polarity					16P8	18P8
Registered - Low				16R4	16R6	16R8
Registered - Polarity				16RP4	16RP6	16RP8

ตารางที่ 2.3 คุณสมบัติของ HY18CV8 ที่สามารถใช้แทน PAL

2.4 การโปรแกรมทางด้านออร์

2.4.1 อุปกรณ์ PROM

รูปที่ 2.24 แสดงโครงสร้างภายในของ PROM ที่มีขนาดเล็ก มีอินพุต 3 อินพุต คือ I_0 , I_1 และ I_2 และมีเอาต์พุต 4 เอาต์พุต คือ O_0 , O_1 , O_2 และ O_3 โดยที่สัญลักษณ์ที่เป็นจุดดำเป็นการกำหนดที่ถาวร สัญลักษณ์วงกลม คือ สามารถทำการ โปรแกรมได้ ถ้าต้องการต่อฟิวส์หรือไม่ต้องการต่อฟิวส์ ตามการออกแบบวงจรที่ได้ออกแบบไว้ในรูปที่ 2.24 ตัว PROM มีอินพุตที่เป็นแอนด์เกต 3 อินพุตถูกกำหนดด้วยค่าที่แน่นอน คือ ค่าตั้งแต่ $I_2I_1I_0$ ถึง $I_2I_1I_0$ ส่วนที่ทำการโปรแกรมได้ คือ เอาต์พุตที่มีออร์เกต ซึ่งเราจะได้ว่า

$$O_3(H) = (I_2I_1I_0 + I_2I_1I_0 + I_2I_1I_0)(H) = (I_2I_1I_0 + I_2I_1I_0)(H) \quad (\text{สมการที่ 2.1})$$

$$O_2(H) = (I_2I_1I_0 + I_2I_1I_0 + I_2I_1I_0)(H) = (I_2I_1I_0 + I_1I_0)(H) \quad (\text{สมการที่ 2.2})$$

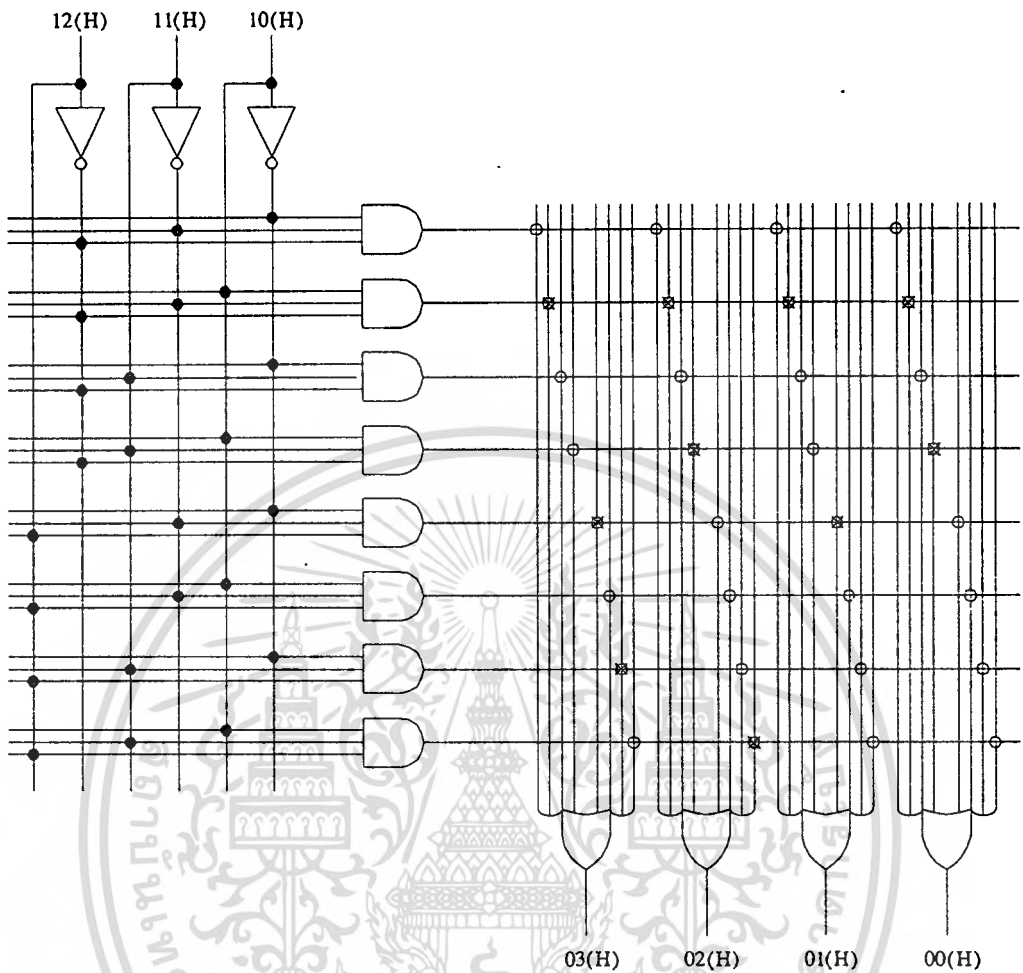
$$O_1(H) = (I_2I_1I_0 + I_2I_1I_0)(H) \quad (\text{สมการที่ 2.3})$$

$$O_0(H) = (I_2I_1I_0 + I_2I_1I_0)(H) \quad (\text{สมการที่ 2.4})$$

PROM คือ อุปกรณ์ที่เป็นหน่วยความจำอีกรูปแบบหนึ่งที่มี การกำหนดค่าตำแหน่งที่แน่นอนในรูปแบบของเลขไบนารีไม่สามารถทำการเปลี่ยนค่าได้ เช่น การกำหนดค่าตามตาราง ที่ 2.4 ค่า $I_2I_1I_0$ ถูกกำหนดค่าดังแสดง ในตารางและนำไปแทนค่า $I_2I_1I_0$ ในสมการที่ 2.1 - สมการที่ 2.4 จากข้างต้นจะได้ค่าเอาต์พุต $O_0O_1O_2O_3$ ดังแสดงในตาราง

Adresses (Input) : I	Contents (Output) : O
000	0000
001	1110
010	0000
011	0100
100	1011
101	0000
110	1000
111	0100

ตารางที่ 2.4 แถวลำดับของรีจิสเตอร์



รูปที่ 2.24 โครงสร้างภายในของ PROM

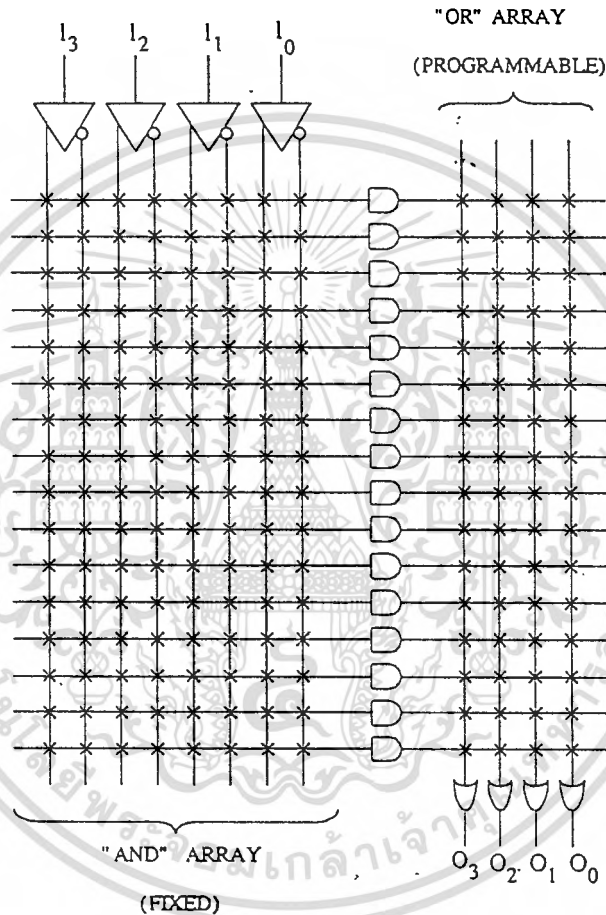
2.5 การโปรแกรมทั้งด้านแอนด์และออร์

2.5.1 อุปกรณ์ FPLA

FPLA ย่อมาจาก Field Programmable Logic Array ถูกคิดค้นขึ้นในปี ค.ศ 1975 โดย บริษัท Signetics/Mullard ได้นำข้อดีของ PROM และ PAL มารวมกัน ทำให้ FPLA เป็นอุปกรณ์ที่สามารถโปรแกรมได้ทั้งส่วนแอนด์อาร์เรย์ และส่วนออร์อาร์เรย์ จึงทำให้ผู้ใช้สามารถที่จะกำหนดจำนวนของเทอมผลคูณของสมการบูลีนเอาต์พุตได้ โดยการโปรแกรม ซึ่งต่างจาก PAL ที่ไม่สามารถกำหนดได้ เนื่องจาก PAL จะคงที่ส่วนออร์อาร์เรย์ รูปที่ 2.25 แสดงให้เห็นถึงตัวอย่างของ FPLA ที่มีขนาด 4 อินพุต 4 เอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเลือกใช้งาน FPLA จำเป็นอย่างยิ่งที่จะต้องเปิดคู่มือเพื่อดูรายละเอียดการใช้งานต่าง ๆ ตลอดจนถึงเบอร์ที่จะใช้งานให้ละเอียดเสียก่อน เนื่องจาก FPLA มีลักษณะที่แตกต่างกัน และมีความซับซ้อนมาก



รูปที่ 2.25 โครงสร้างภายในของ FPLA

สำหรับการกำหนดขนาดของ FPLA นั้นจะนิยมบอกขนาดเป็น $i * p * o$ หรือ จำนวนอินพุตคูณจำนวนเทอมของผลคูณ คูณจำนวนเอาต์พุต

เมื่อ i : จำนวนอินพุตสูงสุด

p : จำนวนเทอมของผลคูณสูงสุด แต่ใน FPLA บางเบอร์เทอมผลคูณจะถูกควบคุมด้วยอุปกรณ์ 3 สถานะ

o : จำนวนเอาต์พุตสูงสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับสถานะการทำงานของเอาต์พุตนั้น ต้องเปิดคู่มือการใช้งานของ FPLA เช่น เบอร์ PLS153 จะมีขนาดเท่ากับ $18 \times 42 \times 10$ หมายถึงจะมีจำนวนอินพุตสูงสุด 18 อินพุต , มีเทอมผลคูณ สูงสุด 42 เทอม และมีจำนวนเอาต์พุตสูงสุด 10 เอาต์พุต

2.6 ข้อดีข้อเสียของการใช้ PLD

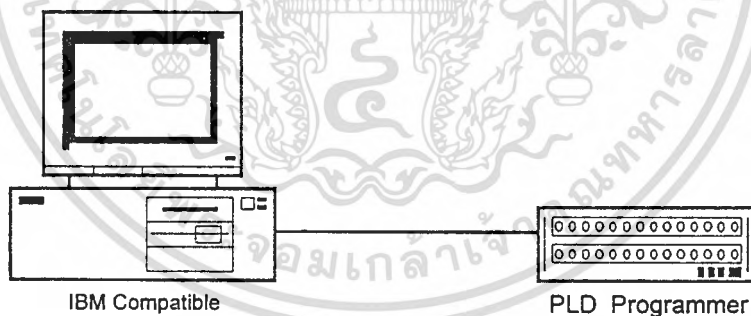
ในสมัยก่อนนั้นเมื่อมีการออกแบบวงจรทางด้านดิจิทัลหรือวงจรอินเทอร์เฟส กับ เครื่องไมโครคอมพิวเตอร์ จะต้องเสียเวลาเปิดคู่มือไอซี เพื่อหาอุปกรณ์ต่าง ๆ และรายละเอียด การทำงานมาออกแบบวงจร ซึ่งทำให้เกิดความยุ่งยากมาก บางครั้งหาซื้ออุปกรณ์ ไม่ได้ ไอซี ที่ให้ออกแบบมีมากเกินไปทำให้ออกแบบลายวงจรลำบากบ้าง พลอยทำให้สิ้นเปลืองค่าใช้จ่าย และเสียเวลา ปัจจุบันนี้อุปกรณ์ PLD สามารถที่จะช่วยให้หมดปัญหาเหล่านี้ไปได้ บางครั้ง ไอซีประเภท SSI หรือ MSI 15 ถึง 20 ตัวก็สามารถบรรจุอยู่ในอุปกรณ์ PLD เพียงตัว เดียวได้ และยังสามารถทำได้ง่าย ทำให้สามารถประหยัดเวลาและมีปัญหาน้อยกว่า เนื่องจาก อุปกรณ์น้อยชิ้น แต่มีข้อเสียในกรณีที่ผู้อื่นไม่ได้ออกแบบวงจรภายในอุปกรณ์ PLD ตัวนั้น ขึ้น จะไม่สามารถตรวจสอบแก้ไข หรือซ่อมแซมเมื่อเกิดการชำรุดได้ แต่ถ้ามองกลับกันก็จะ เป็นประโยชน์ที่ผู้ออกแบบวงจร สามารถป้องกันการลอกเลียนแบบ เป็นการสงวนสิทธิวงจร ไปด้วย

2.7 การโปรแกรมระเบิดฟิวส์

การโปรแกรมอุปกรณ์ PLD นั้น จะเป็นการระเบิดฟิวส์ ในส่วนแอนด์อาร์เรย์ หรือ ออร์อาร์เรย์ ขึ้นอยู่กับชนิดของอุปกรณ์ PLD นั้น ๆ เพื่อเลือกสัญญาณเอาต์พุตเฉพาะสัญญาณ ใด สัญญาณหนึ่งไปป้อนให้กับอินพุตของแอนด์เกต และออร์เกต ในส่วนของ แอนด์และส่วน ของออร์ตามลำดับการระเบิดฟิวส์จะใช้กระแสไฟฟ้าหรือแสงอุลตราไวโอเลตทำการ โปรแกรม ขึ้นอยู่กับกระบวนการผลิตอุปกรณ์ PLD ชนิดนั้น ๆ แต่โดยส่วนใหญ่จะใช้ กระแสไฟฟ้าค่าสูงๆ เนื่องจากมีความสะดวกมากกว่า เมื่อทำการระเบิดฟิวส์เรียบร้อยแล้วฟิวส์ นั้นจะขาดถาวรไม่สามารถที่จะต่อได้อีก ดังนั้นการโปรแกรมใช้งานจะต้องยึดความถูกต้องใน การออกแบบสูงมาก เพื่อไม่ให้เกิดความผิดพลาดขึ้น แต่บางครั้งอาจจะนำมาโปรแกรมใหม่ ได้อีก ซึ่งจะโปรแกรมเฉพาะในส่วนฟิวส์ที่ยังไม่ถูกระเบิดเท่านั้น

ก่อนจะทำการโปรแกรมอุปกรณ์ PLD นั้น จะต้องทำการกำหนดรูปแบบการทำงานที่ต้องการให้กับอุปกรณ์ PLD ก่อน การกำหนดรูปแบบการทำงานต้องอาศัยสมการบูลีนช่วยในการออกแบบเป็นสำคัญ เมื่อได้สมการบูลีนหรือรูปแบบการทำงานที่ต้องการแล้ว จะต้องนำมาสร้างแบบข้อมูล (Data Pattern) โดยใช้โปรแกรมสำเร็จรูปที่ออกแบบมาใช้เฉพาะกับอุปกรณ์ PLD ชนิดนั้น ๆ จากนั้นจึงนำแบบข้อมูลที่ได้ไปทำการโปรแกรมอุปกรณ์ PLD ชนิดที่ต้องการซึ่งจะต้องใช้เครื่องมือไมโครคอมพิวเตอร์ร่วมกับอุปกรณ์โปรแกรม (Programmable Device) เพื่อทำการระบิตฟิวส์ที่ต้องการ สรุปแล้วขั้นตอนการโปรแกรมระบิตฟิวส์จะต้องใช้อุปกรณ์ดังนี้

- ซอฟต์แวร์สำเร็จรูป สำหรับสร้างแบบข้อมูลตามลักษณะของวงจร หรือ อุปกรณ์ที่ต้องการจะออกแบบ
- อุปกรณ์โปรแกรม PLD
- เครื่องไมโครคอมพิวเตอร์เพื่อใช้สำหรับติดต่อกับอุปกรณ์โปรแกรม และให้ซอฟต์แวร์สำเร็จรูปทำงาน



รูปที่ 2.26 การใช้ไมโครคอมพิวเตอร์ร่วมกับเครื่องโปรแกรมอุปกรณ์ PLD

สำหรับในปริญญานิพนธ์ฉบับนี้จะกล่าวถึงการโปรแกรมอุปกรณ์ PLD ประเภท PAL ซึ่งจะได้อธิบายถึงรายละเอียดในหัวข้อถัดไป

2.8 การโปรแกรม PAL

กรรมวิธีโปรแกรม PAL ก่อนข้างจะยุ่งยากและซับซ้อนต้องอาศัยคู่มือของ PAL นำมาศึกษาโดยละเอียด ด้วยขั้นตอนการโปรแกรมที่ยุ่งยากนี้จึงมีการผลิตเครื่องมือสำหรับการโปรแกรมโดยเฉพาะออกมาจำหน่าย บางบริษัทผลิตเป็นซอฟต์แวร์ออกมาเพื่อใช้ร่วมกับเครื่องไมโครคอมพิวเตอร์โดยมีแผ่นการ์ดอินเตอร์เฟซเข้ามาช่วย และการโปรแกรมจะเขียนด้วยภาษาที่ง่ายขึ้น เช่น เพียงแต่บอกความสัมพันธ์ของอินพุต - เอาต์พุต ในรูปแบบของทฤษฎีคิมอร์แกนเท่านั้น แต่โช้วเราจะไม่สามารถโปรแกรมด้วยตนเอง ความจริงแล้วเราสามารถออกแบบ เครื่องมือสำหรับโปรแกรม PAL ได้ด้วยการป้อนแรงดันที่เหมาะสมซึ่งมีหลายระดับ (0, 5, 6, และ 11.5 V. สำหรับโปรแกรม) เข้าไปยังขาต่าง ๆ ของ PAL ให้ถูกต้องและเป็นไปตามลำดับเวลาเพื่อเลือกตำแหน่งที่ต้องการระบิดฟิวส์แล้วเปลี่ยนตำแหน่งไปเรื่อยๆ จนครบตามที่ได้ออกแบบ และคำนวณไว้แรงดันในขณะโปรแกรมจะมีค่าสูงถึง 11.5 V. ผิดพลาดได้ในช่วง ± 0.5 V. แต่ละพัลส์สำหรับการโปรแกรมจะแคบมากเพียง 10 - 50 μ S เท่านั้น เนื่องจากการจัดเรียงตำแหน่งของฟิวส์จะเป็นแบบเมทริกซ์ ดังนั้นการเลือกตำแหน่งฟิวส์ที่ต้องการระบิดแบ่งออกเป็น 2 กลุ่ม คือทางแนวตั้งและแนวนอน ในการเลือกตำแหน่งทางแนวตั้งมีวิธีการทำได้ง่ายกว่าทางแนวนอนคือมีตำแหน่งให้เลือกตั้งแต่ 0-31 ตัวเลขนี้สังเกตได้ว่ามีให้เลือกทั้งสิ้น 32 ตำแหน่ง แต่สำหรับ PAL ขนาด 12 อินพุตจะเลือกได้เพียง 24 ตำแหน่ง ซึ่งจะใช้วิธีข้ามตัวเลขไปบางส่วน

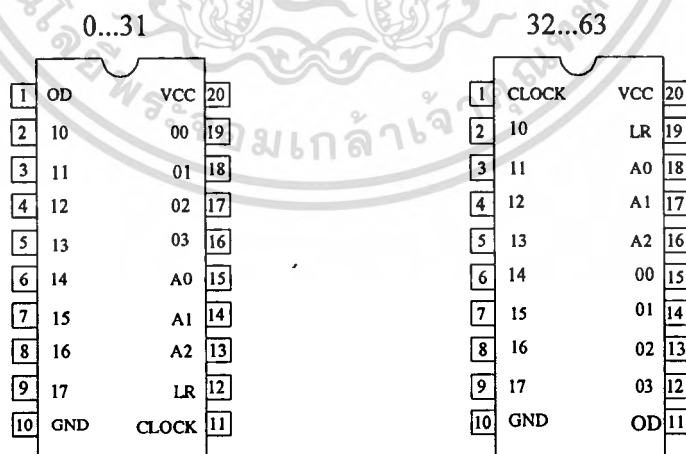
การเลือกตำแหน่งในแนวตั้ง (Input Line Number) จะถูกกำหนดด้วยขาสัญญาณ I_0 - I_7 และ L/R โดยต้องป้อนแรงดันให้กับขาเหล่านี้ตามตารางที่ 2.5 ซึ่งมีทั้งลอจิกต่ำ (L) , สูง (H) , 11.5 V. (HH) และแรงดัน 5 V. ที่ผ่านตัวต้านทานพูลอัพ (R) ขาสัญญาณ I_0 - I_7 และ L/R จะถูกกำหนดขึ้นใหม่ตามรูปที่ 2.27 ก. ซึ่งใกล้เคียงกับการใช้งานปกติ หลังจากโปรแกรมไปแล้ว

การเลือกตำแหน่งทางแนวนอน (Product Line Number) มีวิธีการที่ซับซ้อนขึ้นไปอีกเนื่องจากมีตำแหน่งให้เลือกตั้งแต่ 0-63 หรือเลือกได้ทั้งสิ้น 64 ตำแหน่ง แต่มีขาสัญญาณให้เลือกเพียง 7 ขาเท่านั้น คือ A_0 - A_2 และ O_0 - O_3 น้อยกว่าขาสัญญาณที่เลือกทางแนวนอน ตัวเลข 0-63 นี้ อาจมีตัวเลขบางช่วงที่ถูกข้ามไปบ้างเช่นเดียวกัน ต้องเปิดดูโครงสร้าง ภายในของ PAL แต่ละเบอร์

เนื่องจากมีขาสัญญาณเพียง 7 ขา แต่ต้องเลือกถึง 64 ตำแหน่ง ผู้ออกแบบ PAL ซึ่งต้องใช้วิธีแบ่งออกเป็น 2 ช่วง คือ ช่วงต่ำ (0-31) และช่วงสูง (32-63) การแบ่งวิธีนี้ต้องกำหนดขาสัญญาณ A_0 - A_2 และ O_0 - O_3 ให้แตกต่างกันในแต่ละช่วงดังรูปที่ 2.27 นอกจากนี้ขาสัญญาณที่ใช้ควบคุมการโปรแกรม (OD, CLOCK, L/R) จะถูกเปลี่ยนตามไปด้วยในรูปที่ 2.27 (ก) เป็นการจัดขาในช่วงต่ำ และรูปที่ 2.27 (ข) เป็นการจัดขาในช่วงสูง ส่วนการเลือกตำแหน่งทางแนวตั้ง และแนวนอนในแต่ละช่วงจะเป็นดังตารางที่ 2.6 ซึ่งต้องป้อนพัลส์ 11.5 V. เป็นช่วงเวลา 10-50 μ s (V_{PH}) ด้วย

ตำแหน่งที่ถูกเลือกทางแนวตั้ง และแนวนอนแล้วจะไปกำหนดตำแหน่งที่ต้องการระเบิดฟิวส์ โดยจะต้องป้อนแรงดันที่ขาสัญญาณต่าง ๆ ให้เป็นไปตามแผนผังเวลาในรูปที่ 2.28 ซึ่งแสดงการตรวจสอบ (Verify) เอาไว้ให้ด้วย อันเป็นประโยชน์ในการโปรแกรมซ้ำเข้าไปอีกถ้าหากการโปรแกรมครั้งแรกไม่สำเร็จ

จากขั้นตอนที่ยู้งยากเช่นนี้ เครื่องมือที่ใช้ในการโปรแกรมจะต้องสามารถจ่ายแรงดันให้แต่ละขาสัญญาณได้หลายระดับและควบคุมจังหวะเวลาให้เหมาะสม กรรมวิธีเช่นนี้มีระบบควบคุมที่ทำงานด้วยไมโครโปรเซสเซอร์เท่านั้น ที่จะช่วยให้การโปรแกรมทำได้ง่ายขึ้น และยังสามารถเก็บรูปแบบการโปรแกรมเอาไว้ได้ด้วย ผังเวลาในการโปรแกรมอุปกรณ์ PAL ได้แสดงไว้ในรูปที่ 2.28



(ก) แนวตั้ง, แนวนอนช่วงต่ำ

(ข) แนวนอนช่วงสูง

รูปที่ 2.27 PAL เมื่ออยู่ในช่วงของการโปรแกรมขาสัญญาณต่าง ๆ จะถูกเปลี่ยนไปขึ้น

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์อยู่กับการเลือกตำแหน่งทางแนวตั้งและแนวนอน ดึงนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เส้นตำแหน่ง ทางแนวตั้ง	ขาสัญญาณที่ถูกกำหนดขึ้นใหม่								
	I ₇	I ₆	I ₅	I ₄	I ₃	I ₂	I ₁	I ₀	L/R
0	HH	HH	HH	HH	HH	HH	HH	L	Z
1	HH	HH	HH	HH	HH	HH	HH	H	Z
2	HH	HH	HH	HH	HH	HH	HH	L	HH
3	HH	HH	HH	HH	HH	HH	HH	H	HH
4	HH	HH	HH	HH	HH	HH	L	HH	Z
5	HH	HH	HH	HH	HH	HH	H	HH	Z
6	HH	HH	HH	HH	HH	HH	L	HH	HH
7	HH	HH	HH	HH	HH	HH	H	HH	HH
8	HH	HH	HH	HH	HH	L	HH	HH	Z
9	HH	HH	HH	HH	HH	H	HH	HH	Z
10	HH	HH	HH	HH	HH	L	HH	HH	HH
11	HH	HH	HH	HH	HH	H	HH	HH	HH
12	HH	HH	HH	HH	L	HH	HH	HH	Z
13	HH	HH	HH	HH	H	HH	HH	HH	Z
14	HH	HH	HH	HH	L	HH	HH	HH	HH
15	HH	HH	HH	HH	H	HH	HH	HH	HH
16	HH	HH	HH	L	HH	HH	HH	HH	Z
17	HH	HH	HH	H	HH	HH	HH	HH	Z
18	HH	HH	HH	L	HH	HH	HH	HH	HH
19	HH	HH	HH	H	HH	HH	HH	HH	HH
20	HH	HH	L	HH	HH	HH	HH	HH	Z
21	HH	HH	H	HH	HH	HH	HH	HH	Z
22	HH	HH	L	HH	HH	HH	HH	HH	HH
23	HH	HH	H	HH	HH	HH	HH	HH	HH
24	HH	L	HH	HH	HH	HH	HH	HH	Z
25	HH	H	HH	HH	HH	HH	HH	HH	Z
26	HH	L	HH	HH	HH	HH	HH	HH	HH
27	HH	H	HH	HH	HH	HH	HH	HH	HH
28	L	HH	HH	HH	HH	HH	HH	HH	Z
29	H	HH	HH	HH	HH	HH	HH	HH	Z
30	L	HH	HH	HH	HH	HH	HH	HH	HH
31	H	HH	HH	HH	HH	HH	HH	HH	HH

ตารางที่ 2.5 การป้อนสัญญาณลอจิกต่างๆ เข้าที่ขา PAL ในการเลือกตำแหน่งที่ต้องการ
โปรแกรมทางแนวตั้ง

เส้นตำแหน่ง ทางแนวนอน	ขาสัญญาณที่ถูกกำหนดขึ้น						
	O ₃	O ₂	O ₁	O ₀	A ₂	A ₁	A ₀
0 , 32	Z	Z	Z	HH	Z	Z	Z
1 , 33	Z	Z	Z	HH	Z	Z	HH
2 , 34	Z	Z	Z	HH	Z	HH	Z
3 , 35	Z	Z	Z	HH	Z	HH	HH
4 , 36	Z	Z	Z	HH	HH	Z	Z
5 , 37	Z	Z	Z	HH	HH	Z	HH
6 , 38	Z	Z	Z	HH	HH	HH	Z
7 , 39	Z	Z	Z	HH	HH	HH	HH
8 , 40	Z	Z	HH	Z	Z	Z	Z
9 , 41	Z	Z	HH	Z	Z	Z	HH
10 , 42	Z	Z	HH	Z	Z	HH	Z
11 , 43	Z	Z	HH	Z	Z	HH	HH
12 , 44	Z	Z	HH	Z	HH	Z	Z
13 , 45	Z	Z	HH	Z	HH	Z	HH
14 , 46	Z	Z	HH	Z	HH	HH	Z
15 , 47	Z	Z	HH	Z	HH	HH	HH
16 , 48	Z	HH	Z	Z	Z	Z	Z
17 , 49	Z	HH	Z	Z	Z	Z	HH
18 , 50	Z	HH	Z	Z	Z	HH	Z
19 , 51	Z	HH	Z	Z	Z	HH	HH
20 , 52	Z	HH	Z	Z	HH	Z	Z
21 , 53	Z	HH	Z	Z	HH	Z	HH
22 , 54	Z	HH	Z	Z	HH	HH	Z
23 , 55	Z	HH	Z	Z	HH	HH	HH
24 , 56	HH	Z	Z	Z	Z	Z	Z
25 , 57	HH	Z	Z	Z	Z	Z	HH
26 , 58	HH	Z	Z	Z	Z	HH	Z
27 , 59	HH	Z	Z	Z	Z	HH	HH
28 , 60	HH	Z	Z	Z	HH	Z	Z
29 , 61	HH	Z	Z	Z	HH	Z	HH
30 , 62	HH	Z	Z	Z	HH	HH	Z
31 , 63	HH	Z	Z	Z	HH	HH	HH

ตารางที่ 2.6 การป้อนสัญญาณลอจิก และพัลส์โปรแกรมเพื่อกำหนดตำแหน่ง
ทางแนวนอนทั้งช่วงต่ำ และ ช่วงสูง

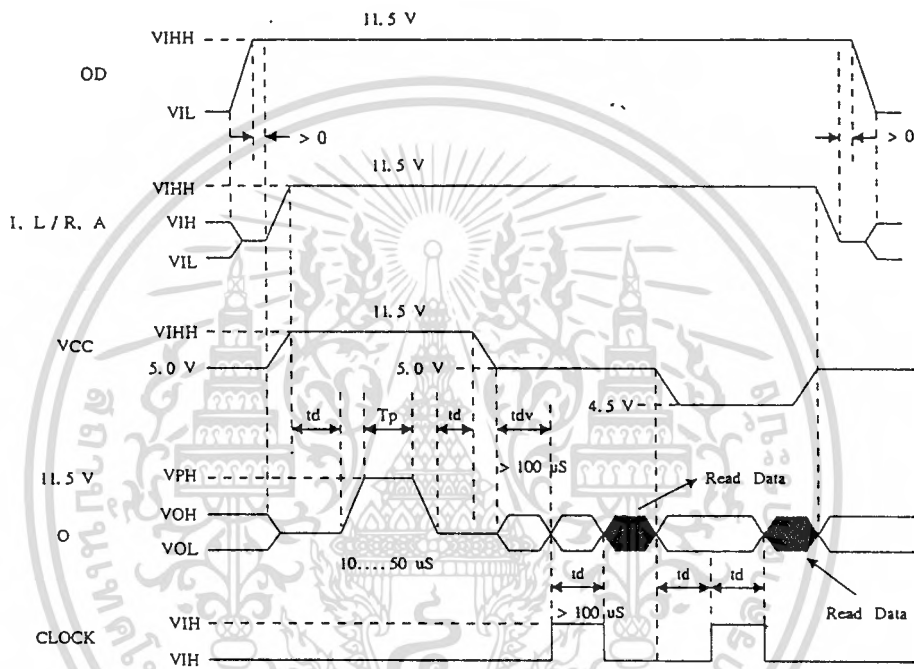
หมายเหตุ

L = สัญญาณอินพุตลอจิก “0” (V_L)

H = สัญญาณอินพุตลอจิก “1” (H_{IH})

HH = แรงดันอินพุตสูงสำหรับการโปรแกรม (V_{HH})

Z = ต่อผ่านตัวต้านทาน 10 K Ω ไปเข้า +5 V.



รูปที่ 2.28 แผนผังเวลาในการ โปรแกรมและตรวจสอบ PAL

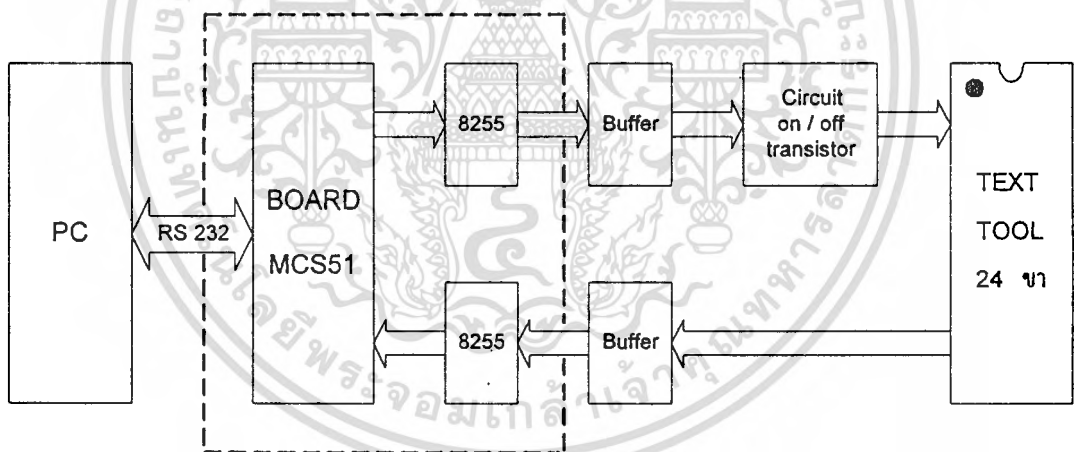
บทที่ 3

การออกแบบและการสร้าง

จากการที่เราได้ทราบถึงรายละเอียดต่าง ๆ ในบทที่ 2 ไปแล้วว่าเครื่องโปรแกรม PAL จะต้องประกอบด้วย ใ้พระดับต่าง ๆ ที่ใช้ในโปรแกรมและทำการอ่านข้อมูลในตัว PAL นั้น เราสามารถออกแบบผังการทำงาน (Block Diagram) ของเครื่องโปรแกรมได้ดังนี้

การออกแบบและการทำงาน

ในการออกแบบนั้นเราได้แบ่งการทำงานออกเป็น 8 ภาคด้วยกัน แสดงได้ดังรูปที่ 3.1



รูปที่ 3.1 ผังการทำงานของเครื่องโปรแกรม PAL

3.1 คุณสมบัติของบอร์ด MCS - 51

ไมโครคอนโทรลเลอร์ : 80C31

หน่วยความจำ : EPROM 2764 (8 กิโลไบต์) ใช้งานที่แอดเดรส 0000H-1FFFH

RAM 6264 (8 กิโลไบต์) ใช้งานที่แอดเดรส 0000H-1FFFH

RAM 6264 (8 กิโลไบต์) ใช้งานที่แอดเดรส 8000H-9FFFH

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RAM ภายใน 128 ไบต์ ใช้เป็นรีจิสเตอร์

พอร์ตอินพุต/เอาต์พุต : ขนาด 48 บิต (ใช้ชิพ 8255 2 ตัว)

8255 ตัวที่ 1 ใช้งานที่แอดเดรส (F800H-F9FFH)

8255 ตัวที่ 2 ใช้งานที่แอดเดรส (FC00H-FDFFH)

LCD BUS ใช้งานที่แอดเดรส (FA00H-FBFFH)

LCD

: ใช้แบบ DOT Matrix LCD ขนาด 20 ตัว * 2 แถว

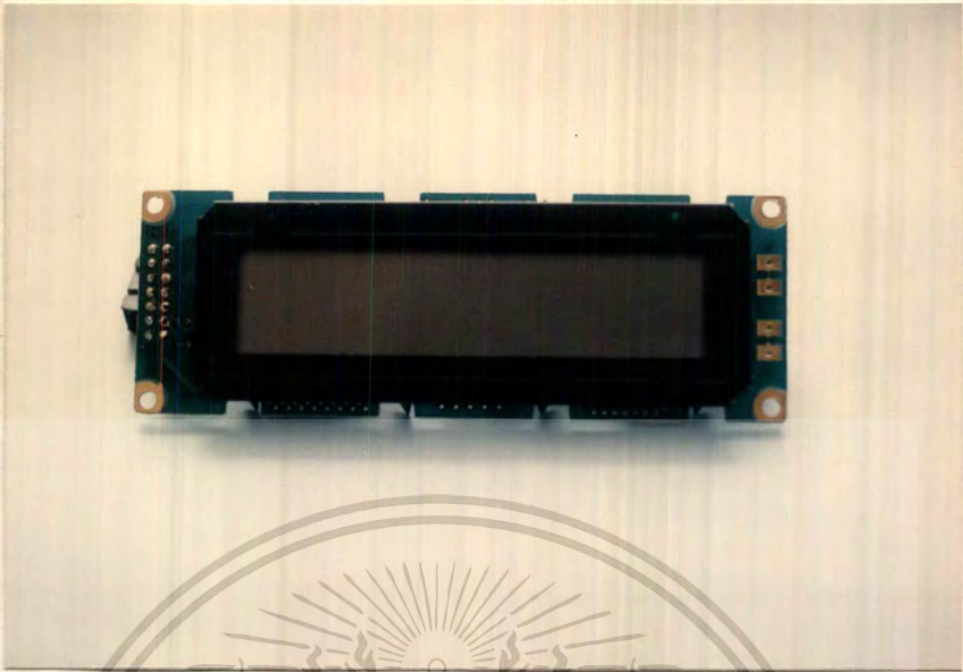
หมายเหตุ

คุณสมบัติบอร์ดดังกล่าวสามารถหาซื้อได้ โดยใช้บอร์ดของบริษัทสิลารีเสิร์ช
จำกัดรุ่น ANT - 32 V3.0



รูปที่ 3.2 บอร์ด MCS-51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 ชุดแสดงผล DOT Matrix LCD ขนาด 20 ตัว * 20 แถว

3.2 การจัดหน่วยความจำบอร์ด MCS - 51 (ANT - 32 V3.0)

การจัดหน่วยความจำแสดงดังรูปที่ 3.4

FFFFH	NOT USER	
FE00H	8255 ตัวที่ 2	
FC00H	ใช้กับ LED	
FA00H	8255 ตัวที่ 1	
F800 H		
A000 H	U4 RAM CODE & DATA MEMORY	
8000 H	NON USER	
2000 H	U2 EPROM CODE MEMORY	U3 EPROM DATA MEMORY
0000 H		

รูปที่ 3.4 การจัดหน่วยความจำของระบบ

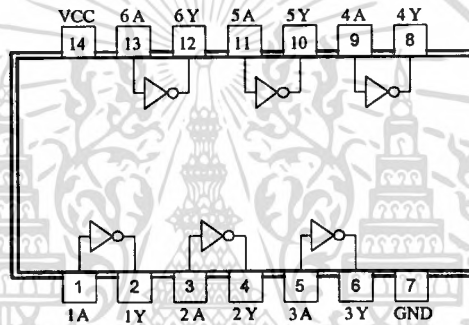
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่ออนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ รายละเอียดการ JUMPER ต่าง ๆ สามารถดูได้จากภาคผนวก

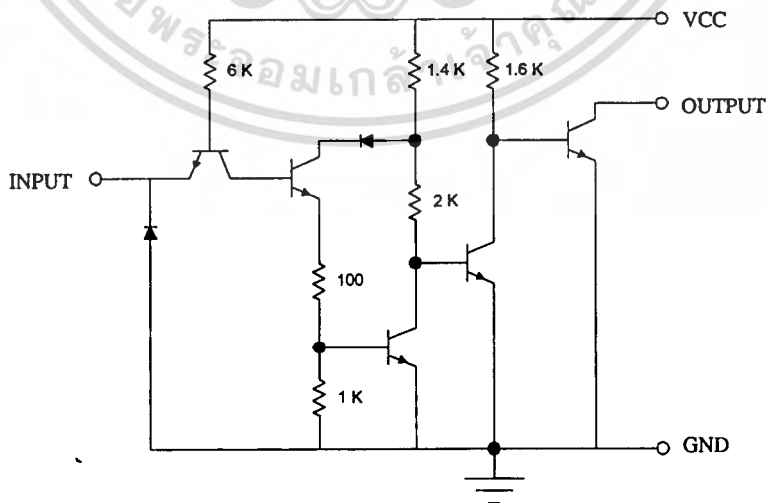
3.3 การเลือกใช้ไอซีบัฟเฟอร์ (IC Buffer)

3.3.1 ไอซีบัฟเฟอร์ที่ใช้กับวงจรเปิด / ปิด (ON/OFF)

ไอซีบัฟเฟอร์ที่ใช้กับวงจรเปิด / ปิด ทรานซิสเตอร์ (ON/OFF Transistor) นั้นเป็นแบบอินเวอร์เตอร์ (Inverter Buffer) ที่สามารถทนแรงดันเอาต์พุตได้สูง (Driver with Open-Collector High Voltage Output) คุณสมบัติดังกล่าวได้แก่ไอซี TTL เบอร์ 7406 มีลักษณะดังรูปที่ 3.5 และรูปที่ 3.6



รูปที่ 3.5 โครงสร้างภายในของไอซีเบอร์ 7406

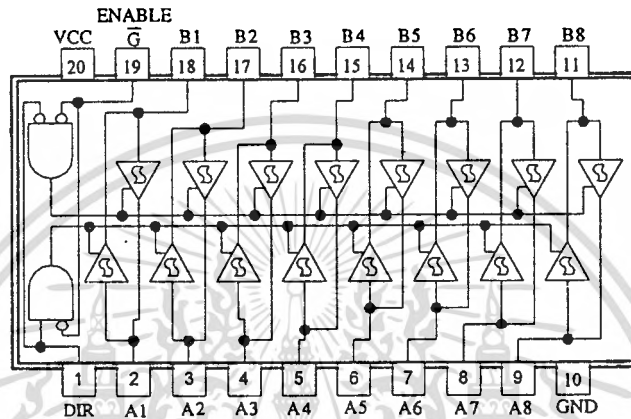


รูปที่ 3.6 โครงสร้างภายใน 1 เกตของไอซี 7406

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 ไอซีบัฟเฟอร์ที่ใช้ในการอ่านอุปกรณ์ PAL

ไอซีบัฟเฟอร์ที่ใช้ในการอ่านอุปกรณ์ PAL นั้นใช้ไอซี TTL เบอร์ 74245 (Octal Bus Transceivers with 3-state Output) ซึ่งมีลักษณะดังรูปที่ 3.7



รูปที่ 3.7 โครงสร้างภายในของไอซีเบอร์ 74245

3.4 การออกแบบวงจรเปิด / ปิด ทรานซิสเตอร์

3.4.1 การออกแบบวงจรตามตารางความจริงของตำแหน่งอินพุตทางแนวตั้ง

จากตารางความจริงของตำแหน่งอินพุตในบทที่ 2 ตารางที่ 2.5 หน้าที่ 40 โดยตัดตารางช่อง L/R เอาไปไว้กับตารางความจริงของการกำหนดตำแหน่งทางแนวนอน คือ ตารางที่ 2.6 เหลือแต่ช่วง I_0-I_7 จากตารางดังกล่าวจะเห็นว่ามีความระดับแรงดันไฟ 3 ระดับ ดังนี้

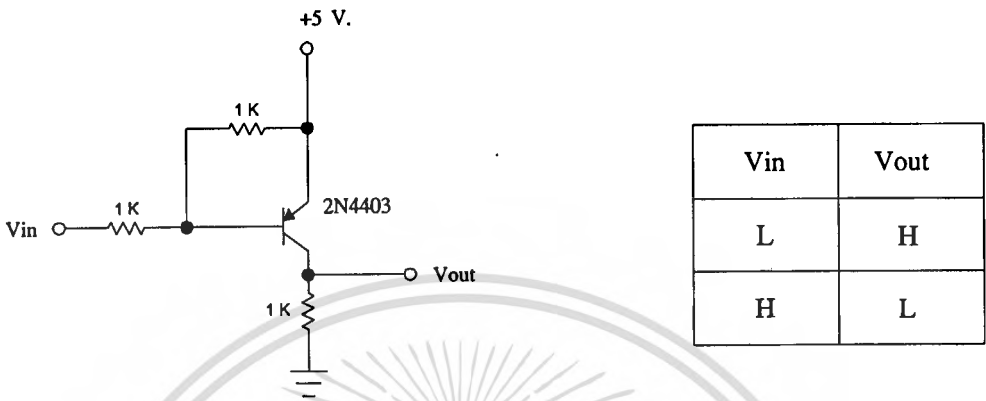
- HH = High - level program voltage, V_{HH} (12 V.)
- H = High - level input voltage, V_{IH} (5 V.)
- L = Low - level input voltage, V_{IL} (0 V.)

ลำดับขั้นตอนออกแบบวงจร

1. ออกแบบวงจรควบคุมระดับแรงดัน H กับ L
2. ออกแบบวงจรควบคุมระดับแรงดัน HH กับ L
3. นำวงจรที่ 1 และ 2 มาทำการรวมกัน

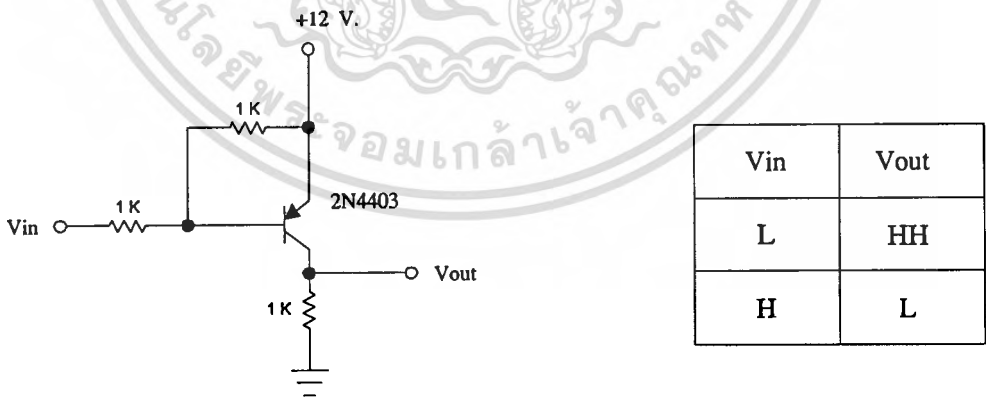
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนที่ 1

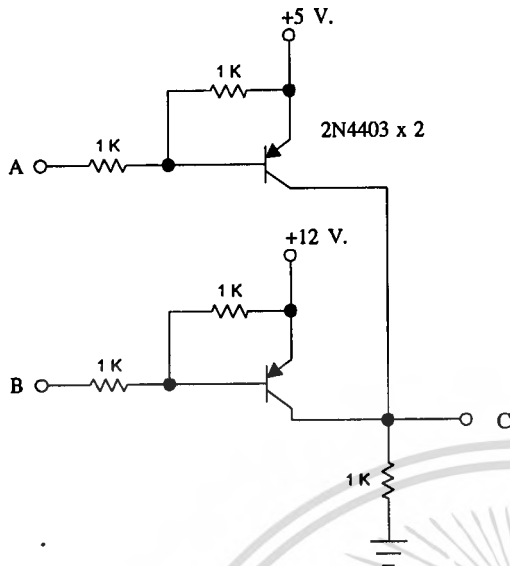


รูปที่ 3.8 วงจรควบคุมระดับแรงดัน H กับ L และ ตารางการทำงาน

หมายเหตุ ที่ Vin มีค่า H กับ L ก็ือค่า Logic High , Low (1 , 0)



รูปที่ 3.9 วงจรควบคุมระดับแรงดัน HH, L และตารางการทำงาน



Input		Output
A	B	C
L	L	HH
L	H	H
H	L	HH
H	H	L

รูปที่ 3.10 วงจรควบคุมระดับแรงดัน HH, H, L และตารางการทำงาน

ในการเลือกค่าความต้านทานนั้นเราใช้กระแสในคอนโปรแกรมอุปกรณ์ PAL ที่ $I_0 - I_7$ ประมาณ 5 - 15 mA ดังนั้นกระแสที่ได้จากการออกแบบวงจรข้างบน คือ มีค่าประมาณ 5 - 12 mA ได้มาจาก

กำหนดให้ $V_{CE} = 0$ V. ขณะ ON, $V_{CC} = 5$ V.

$$R_C = V_{CC} / I_{CQ}$$

$$R_C = 5 \text{ V} / 5 \text{ mA}$$

$$R_C = 1 \text{ K}\Omega$$

และที่แหล่งจ่ายไฟ = 12 V. จะได้ I_{CQ} ประมาณ

$$I_{CQ} = V_{CC} / R_C$$

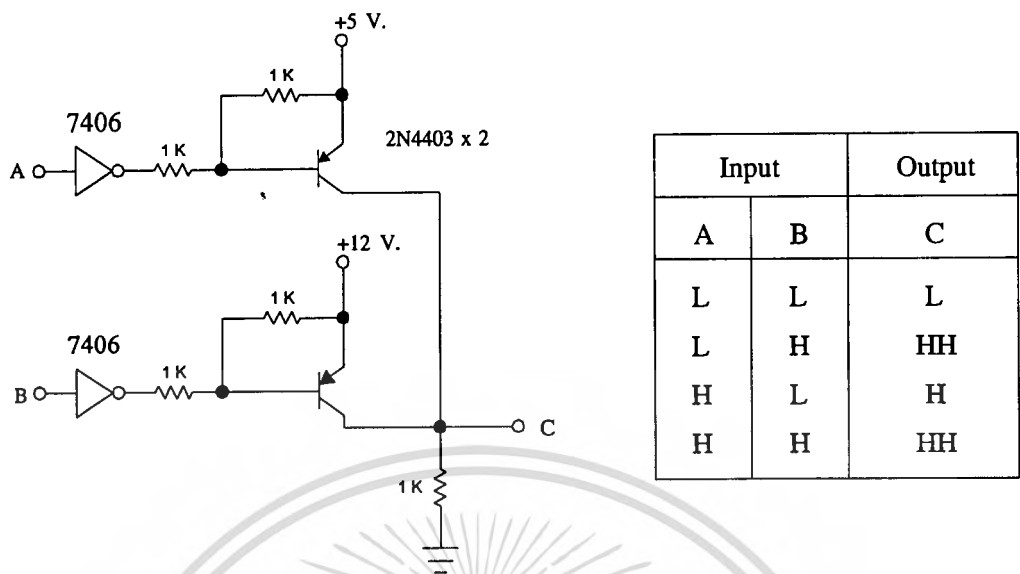
$$I_{CQ} = 12 \text{ V} / 1 \text{ K}$$

$$I_{CQ} = 12 \text{ mA}$$

3.4.2 การนำวงจรบัฟเฟอร์มาต่อกับวงจรเปิด / ปิด ทรานซิสเตอร์

เพื่อให้ง่ายแก่การควบคุมการเปิด / ปิดของทรานซิสเตอร์ จึงนำไอซี TTL เบอร์ 7406 มาต่อร่วมกับขาเบสของทรานซิสเตอร์ ของวงจรควบคุมระดับแรงดัน มีลักษณะการต่อใช้งาน ดังรูปที่ 3.11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



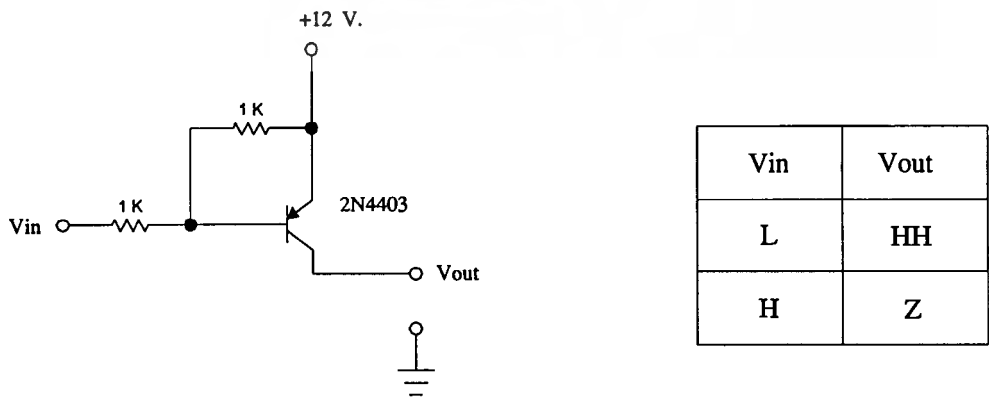
รูปที่ 3.11 วงจรควบคุมระดับแรงดัน HH, H, L ที่มี 7406 ต่อร่วม และตารางการทำงาน

3.4.3 การออกแบบวงจรตามตารางความจริงของตำแหน่งอินพุตทางแวนอน

จากตารางความจริงของการกำหนดตำแหน่งอินพุตทางแวนอน ในบทที่ 2 หน้า 40 ตารางที่ 2.6 โดยนำเอา L/R มาทำการต่อร่วมด้วย จะเห็นว่ามีระดับแรงดันไปอยู่ 2 ระดับ ดังนี้

- HH = High - level program voltage , V_{iHH} (12 V)
- Z = High - impedance (e.g , 10 K Ω to 5.0 V)

เราสามารถออกแบบวงจรได้โดยตัดค่าความต้านทานที่ขาคอลเลกเตอร์ออกมีลักษณะการต่อดังรูปที่ 3.12

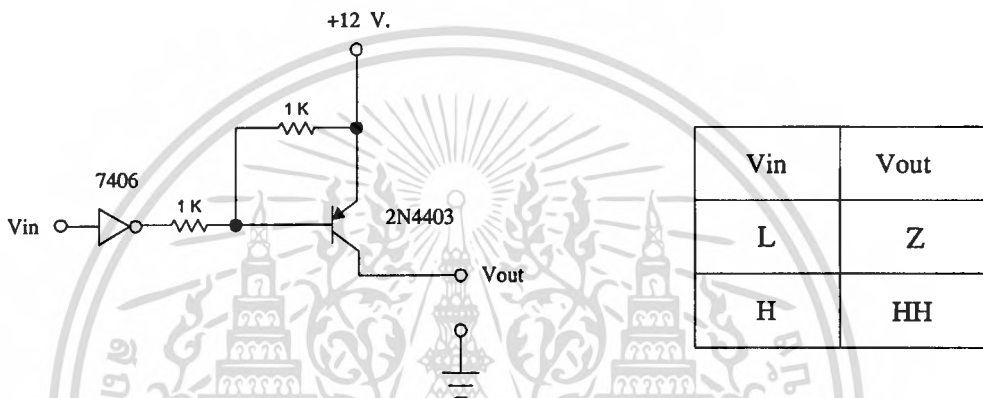


รูปที่ 3.12 วงจรควบคุมระดับแรงดัน HH, Z และตารางการทำงาน

หมายเหตุ แรงดันมากกว่า 1.2 V. แต่ไม่เกิน 2.2 V. คือ สภาวะ High impedance

3.4.4 การนำวงจรบัฟเฟอร์มาต่อกับวงจร เปิด / ปิด ทรานซิสเตอร์

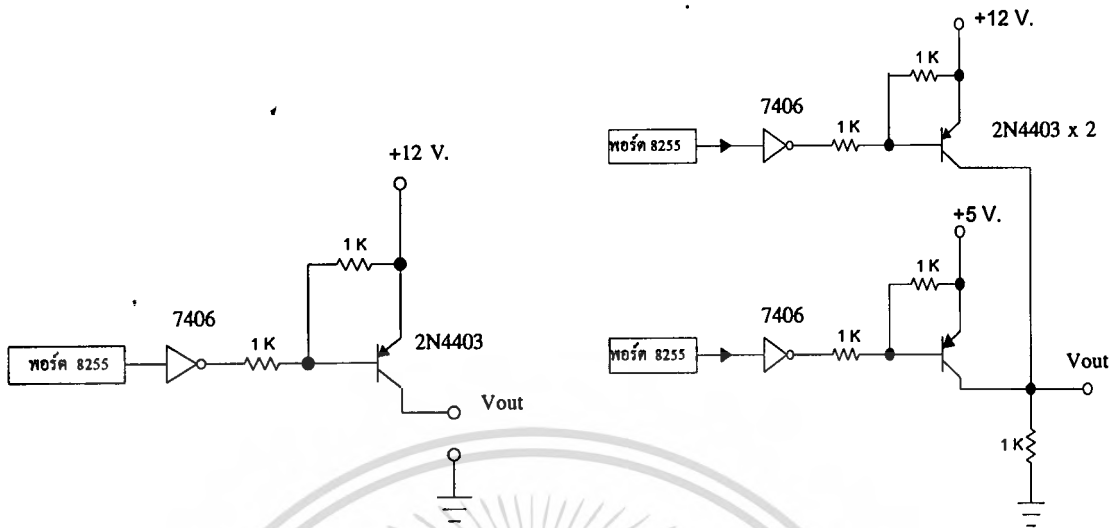
เพื่อให้ง่ายแก่การควบคุมการ เปิด / ปิด ของทรานซิสเตอร์ จึงนำไอซี TTL เบอร์ 7406 มาต่อร่วมกับขาเบสของทรานซิสเตอร์ ของวงจรควบคุมระดับแรงดัน มีลักษณะการทำงาน ดังรูปที่ 3.12



รูปที่ 3.13 วงจรควบคุมระดับแรงดัน HH, Z ที่มี 7406 ต่อร่วมและตารางการทำงาน

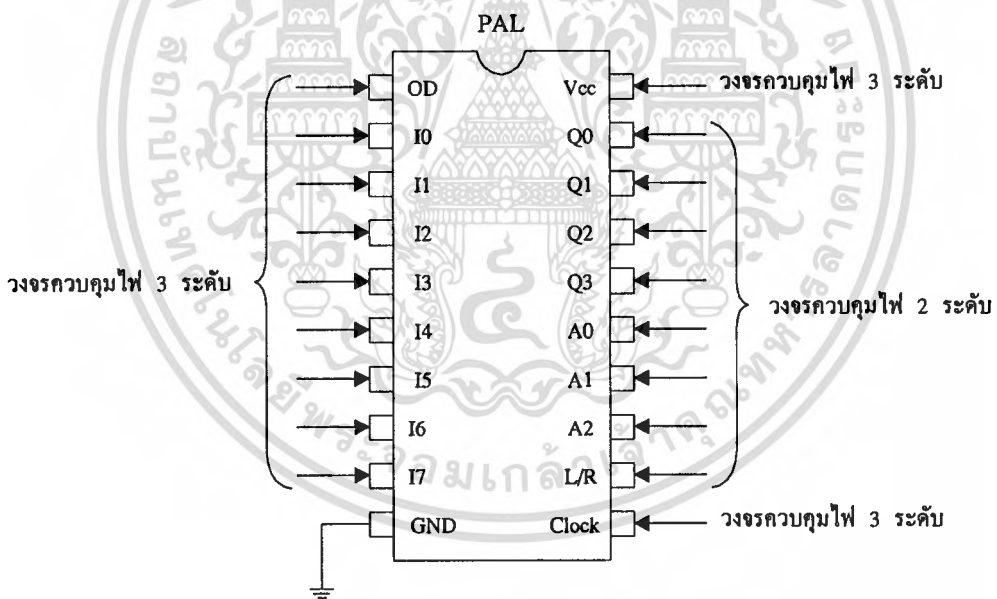
3.5 การรวมวงจรทรานซิสเตอร์

ในหัวข้อนี้จะเป็นการนำเอาวงจรทรานซิสเตอร์ที่ออกแบบไว้ในแต่ละส่วน มารวมเป็นวงจรขนาดใหญ่เข้าด้วยกัน ซึ่งเป็นวงจรที่ใช้ในการโปรแกรมตัวอุปกรณ์ PAL มีลักษณะการทำงาน ดังรูปที่ 3.14



(ก) วงจรควบคุมไฟ 2 ระดับ

(ข) วงจรควบคุมไฟ 3 ระดับ



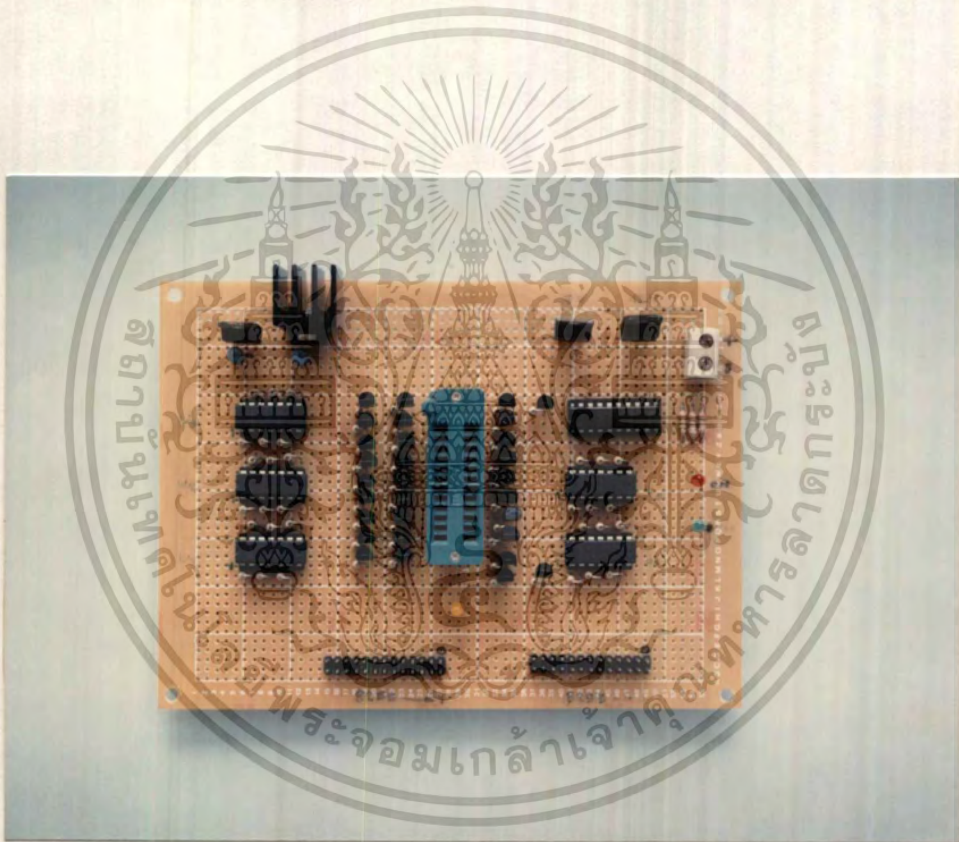
(ค) วงจรที่ใช้โปรแกรมอุปกรณ์ PAL

รูปที่ 3.14 การต่อวงจรที่ใช้โปรแกรมอุปกรณ์ PAL

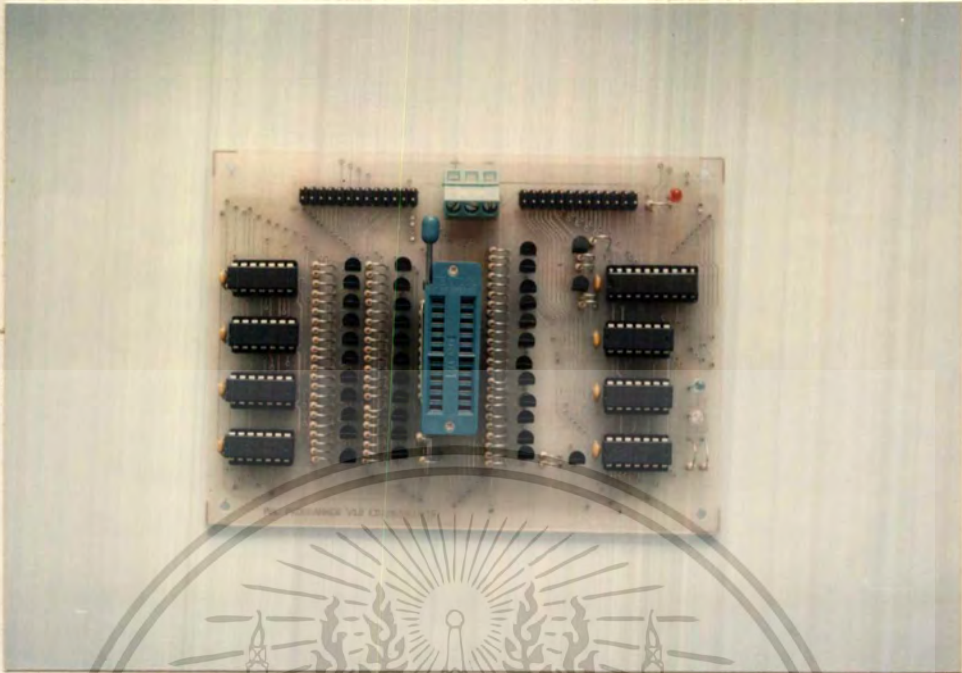
จากรูปจะเห็นว่าขา OD , I₀-I₇ , Clock , Vcc นั้นจะประกอบไปด้วยทรานซิสเตอร์ 2 ตัว ต่อร่วมกัน เพื่อทำหน้าที่สร้างไฟ 3 ระดับ ในการเข้าถึงตำแหน่ง PAL ในขณะที่โปรแกรมตำแหน่งพินส์ ส่วน Q₀-Q₃ , A₀-A₂ , L/R นั้นประกอบไปด้วยทรานซิสเตอร์ตัวเดียว ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สร้างไฟได้ 2 ระดับ คือ 0 V. กับ 12 V. ทำหน้าที่เลือกเส้นทางแวนอน ที่ต้องการโปรแกรม

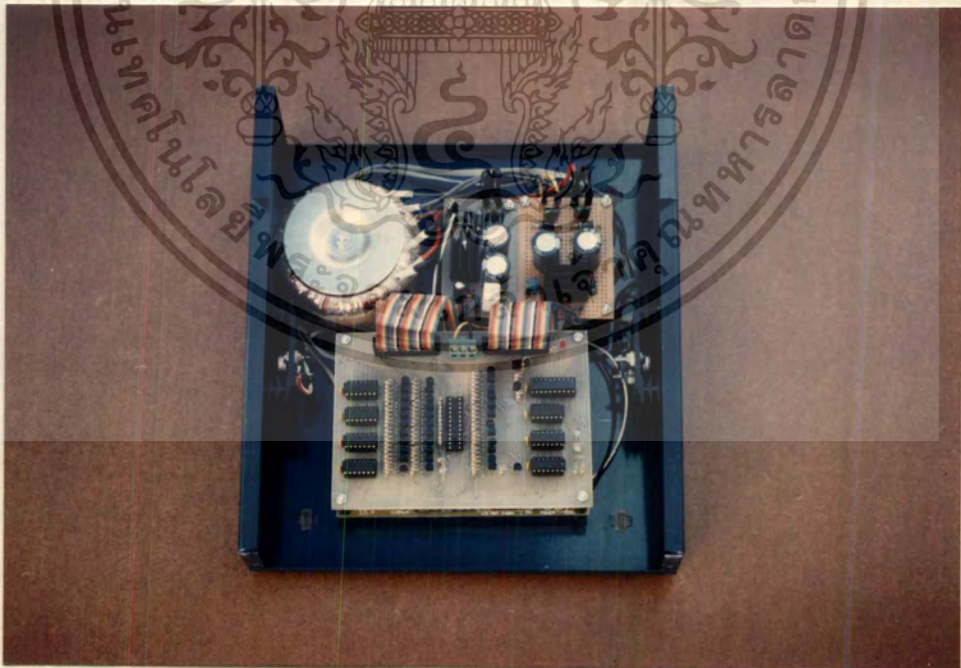
หมายเหตุ ไฟ 3 ระดับ สามารถดูรายละเอียดได้จากบทที่ 2 หน้า 40-42 การเข้าถึงตำแหน่งแอดเดรสพินแต่ละจุดนั้น สามารถดูได้จากตารางหน้า 40-41 และสร้างลอจิกควบคุมทรานซิสเตอร์ให้มีการทำงานตามตารางความจริง ที่เราต้องการเข้าถึงตำแหน่งนั้น ๆ



รูปที่ 3.15 บอร์ดที่ใช้ในการทดลอง

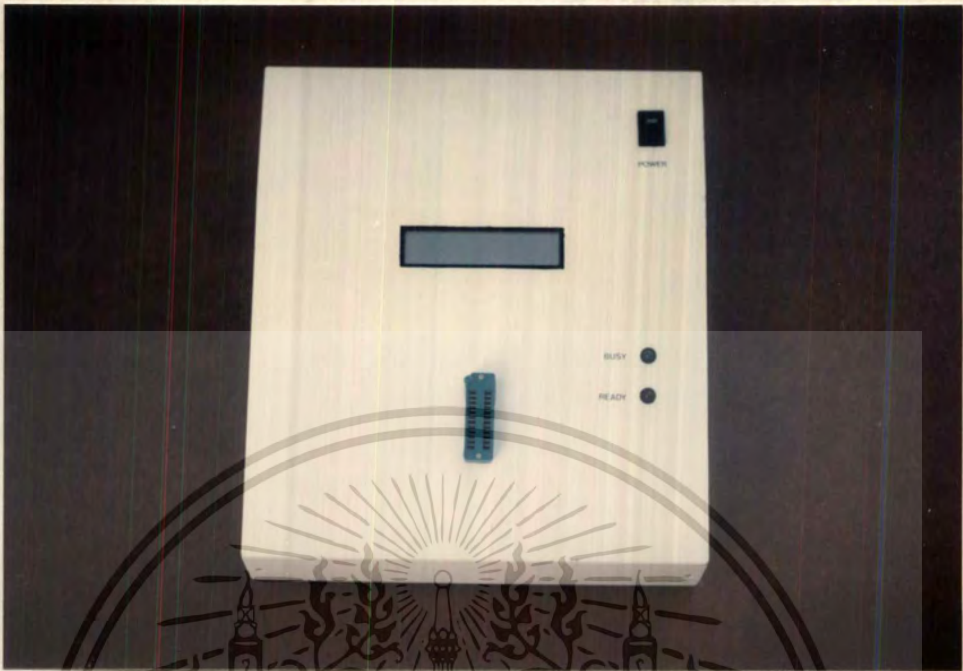


รูปที่ 3.16 บอร์ดที่ออกแบบใช้งานจริง



รูปที่ 3.17 ตำแหน่งการวางอุปกรณ์ภายในเครื่องโปรแกรม PAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.18 เครื่องโปรแกรม PAL

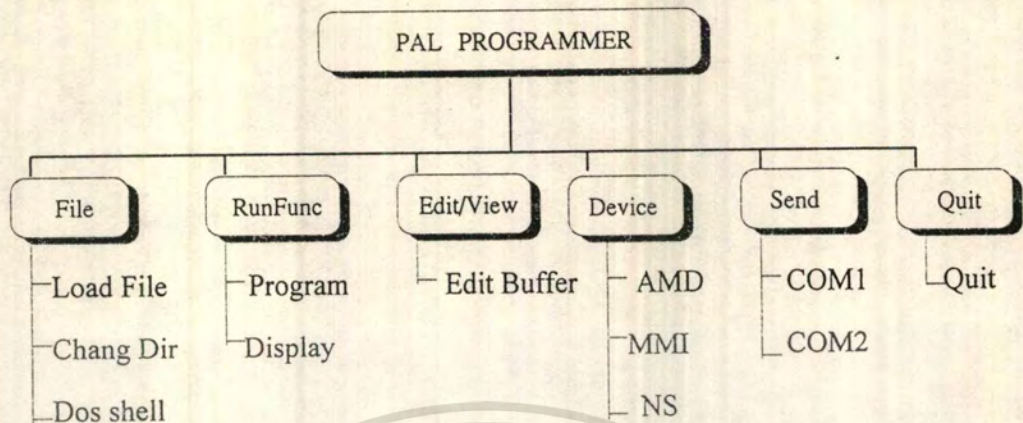
3.6 การออกแบบสร้างโปรแกรม

การออกแบบสร้างโปรแกรมนั้นแบ่งเป็น 2 ส่วนใหญ่ ๆ

1. ส่วนของภาษาซี
2. ส่วนของภาษาแอสเซมบลีของ MCS-51

3.6.1 ส่วนของภาษาซี

ทำหน้าที่จัดการส่งข้อมูล JEDEC FILE ที่ได้จากการออกแบบวงจรโดยใช้โปรแกรมจาก ORCAD , OPAL และทำหน้าที่เลือกเบอร์ต่าง ๆ ของ PAL ที่นำมาทำการโปรแกรมเป็นวงจร โดยส่งข้อมูลให้บอร์ด MCS-51 รับทราบเพื่อให้ข้อมูลที่ใช้ในการโปรแกรมตรงกันทั้ง 2 ฝ่าย ลักษณะโครงสร้างของซอฟต์แวร์แสดงดังรูปที่ 3.19



รูปที่ 3.19 โครงสร้างของซอฟต์แวร์

ซอฟต์แวร์ที่เขียนขึ้นเป็นส่วนที่ติดต่อกับผู้ใช้ โดยที่ผู้ใช้สามารถเลือกฟังก์ชันต่าง ๆ ผ่านทางคีย์บอร์ด แล้วโปรแกรมก็จะทำตามฟังก์ชันที่เลือกไปจนเสร็จแล้วจะกลับมารอรับคำสั่งใหม่ สำหรับรายละเอียดของเมนูต่าง ๆ นั้นสามารถอธิบายได้ดังนี้

1. เมนู “File”

- 1.1 ฟังก์ชัน “Load File” จะทำการโหลดเพิ่มข้อมูล JEDEC ไฟล์ ไปเก็บไว้ในบัฟเฟอร์ และ ก็จะส่งข้อมูลชุดเดียวกันนี้ไปให้เครื่องโปรแกรม เพื่อรอที่จะทำการโปรแกรม
- 1.2 ฟังก์ชัน “Chang Dir” จะทำการเปลี่ยนไดเรกทอรีปัจจุบัน เพื่อที่จะทำการอ่าน JEDEC ไฟล์ ที่อยู่ในไดเรกทอรีอื่น
- 1.3 ฟังก์ชัน “Dos shell” กลับเข้าสู่คอสชักราว

2. เมนู “RunFunc”

- 2.1 ฟังก์ชัน “Program” ทำหน้าที่โปรแกรมอุปกรณ์ PLD ตามข้อมูลของ JEDEC ไฟล์ ที่อยู่ในบัฟเฟอร์
- 2.2 ฟังก์ชัน “Display Device” ทำหน้าที่แสดงผลสถานะของฟิวส์ที่อ่านมา

3. เมนู “Edit/View”

- 3.1 ฟังก์ชัน “Edit/Buffer” จะทำหน้าที่แสดงข้อมูลที่อยู่ในบัฟเฟอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. เมนู “ Device ”

เมนูนี้ใช้ในการเลือกบริษัทผู้ผลิต และ เบอร์ ของอุปกรณ์ PLD ที่เราจะใช้ในการ
โปรแกรม บริษัทผู้ผลิตจะมีให้เลือกอยู่ 3 บริษัทคือ

1. AMD
2. MMI
3. NS

เบอร์ของอุปกรณ์ PLD มีให้เลือกดังนี้

1. บริษัท AMD มี

- 16L8
- 16R4
- 16R6
- 16R8

2. บริษัท MMI มี

- 16R4
- 16R6
- 16L8
- 16R8
- 20R4
- 20R6
- 20L8
- 20R8

3. บริษัท NS มี

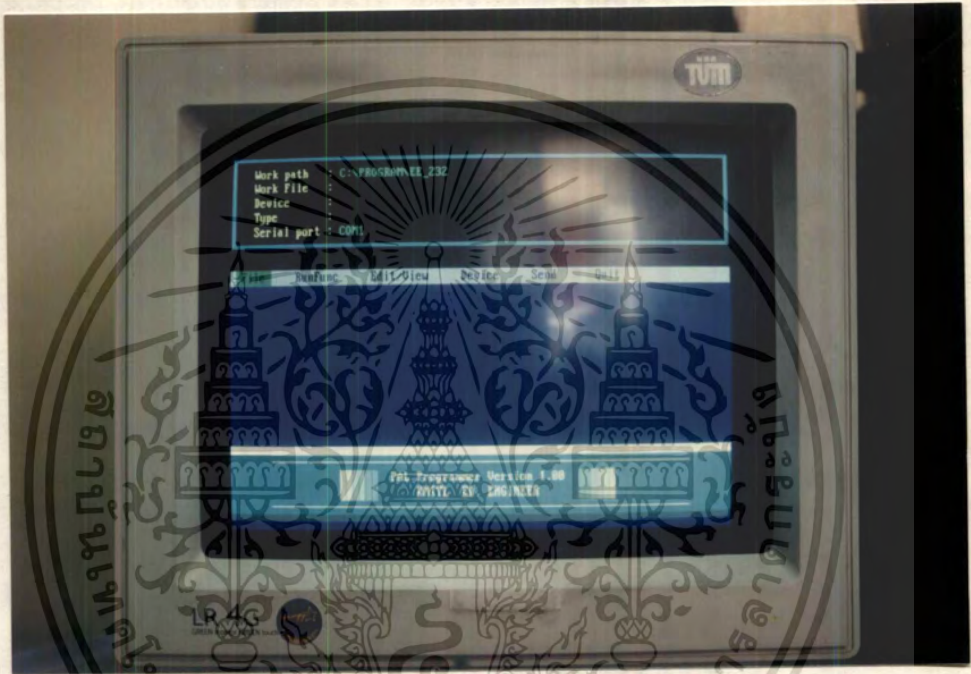
- 16R4
- 16R6
- 16L8
- 16R8
- 20R4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 20L8

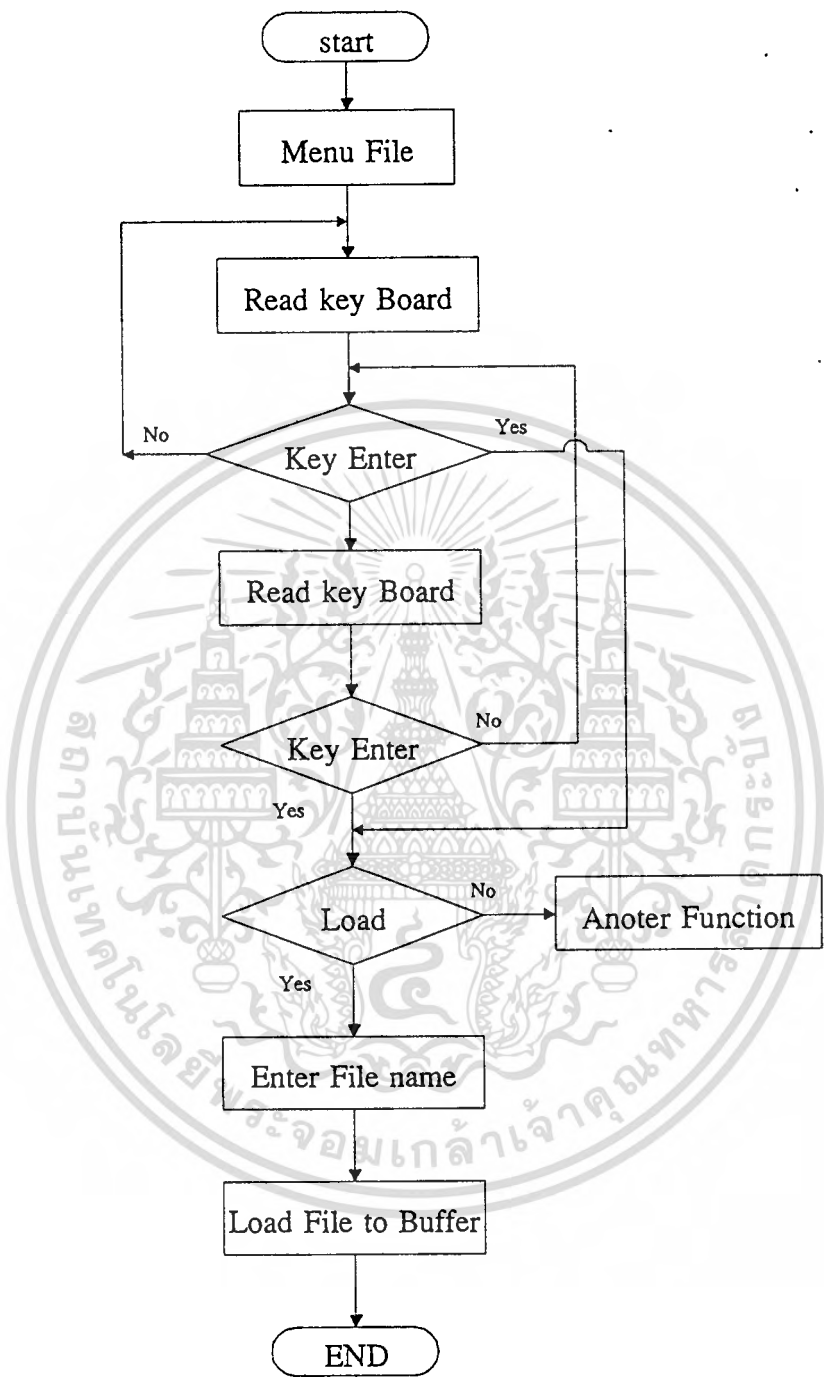
- 20R8

สำหรับโปรแกรมในส่วนนี้ เป็นโปรแกรมทำหน้าที่เพื่อติดต่อกับผู้ใช้ได้ง่ายและสะดวก
ขึ้นโดยมีลักษณะที่หน้าจอเป็นดังรูปที่ 3.20



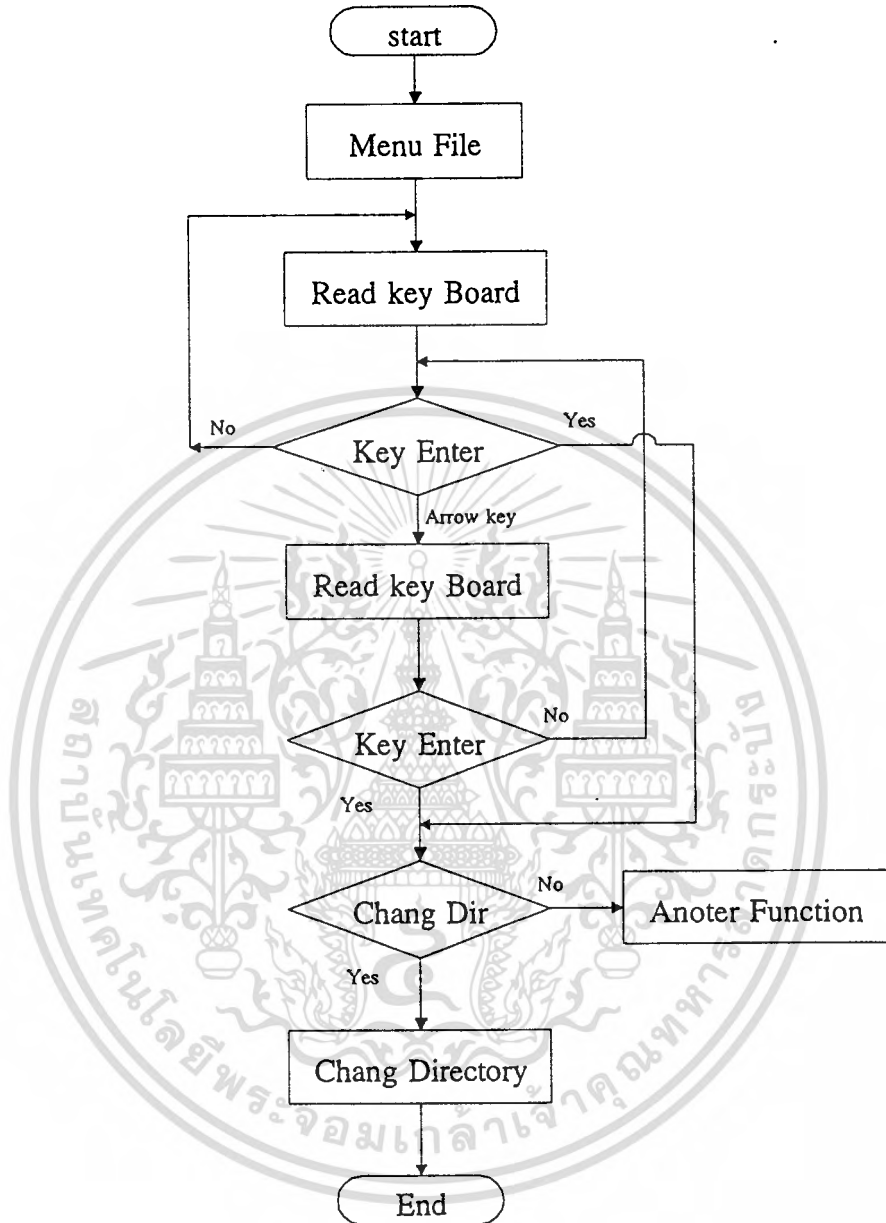
รูปที่ 3.20 ลักษณะของหน้าจอที่ติดต่อกับผู้ใช้งาน

ในการติดต่อกับผู้ใช้งาน ผู้ใช้สามารถเลือกการทำงานของฟังก์ชันต่างๆ ผ่านทางคีย์บอร์ด
ซึ่งในการเขียนโปรแกรมแต่ละฟังก์ชันสามารถอธิบายด้วยผังงานได้ดังนี้



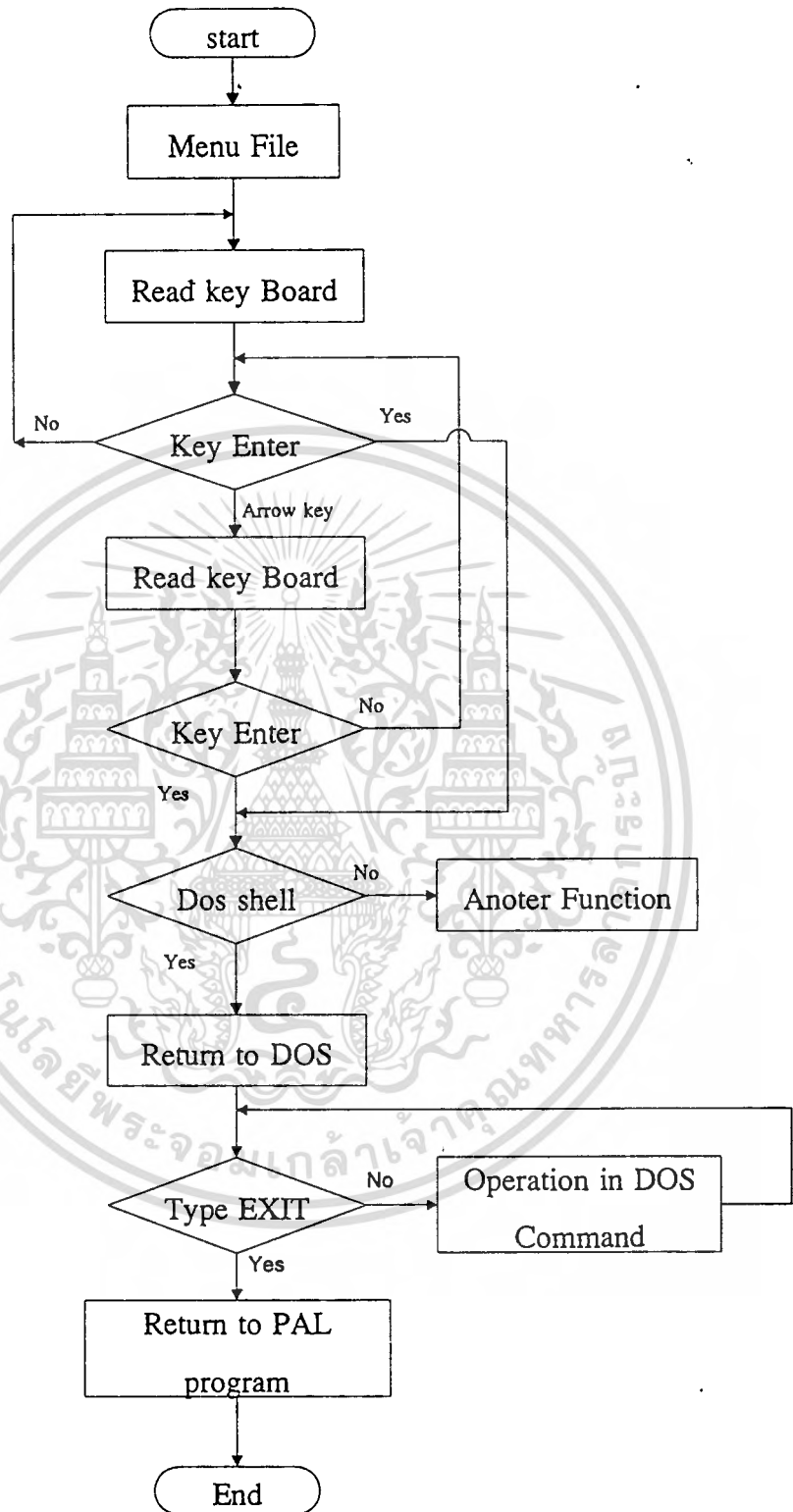
รูปที่ 3.21 ผังงานฟังก์ชันการทำงานของ “Load”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



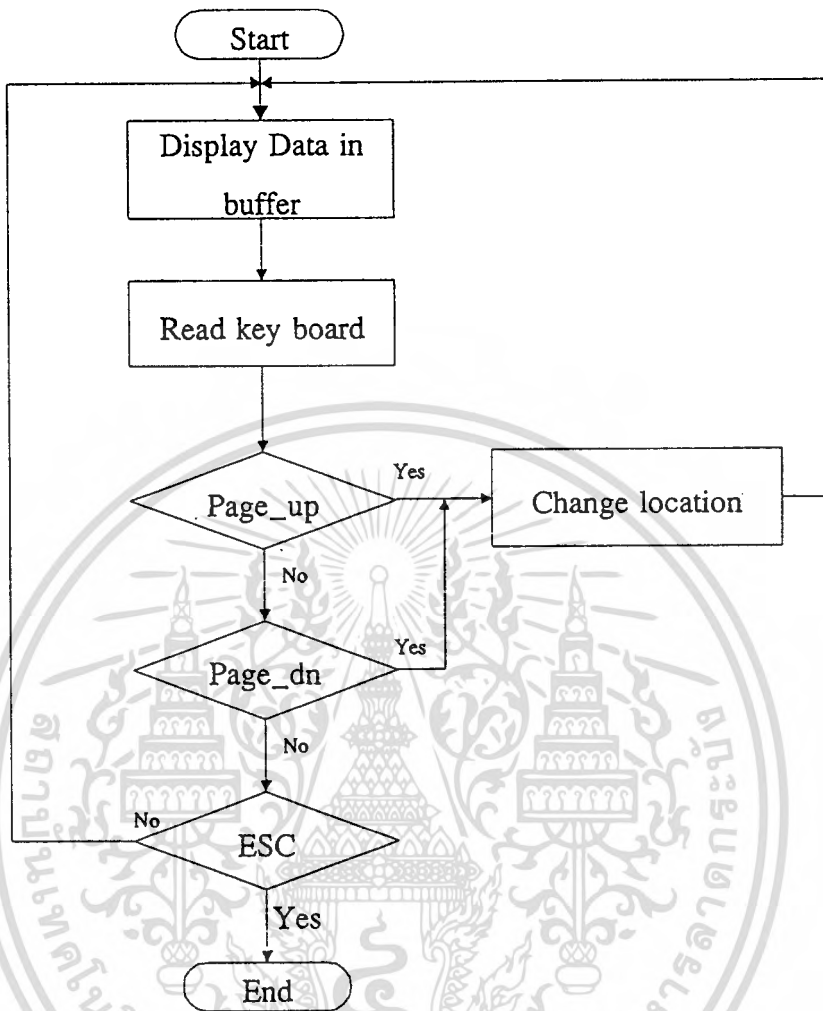
รูปที่ 3.22 ฟังก์ชันการทำงานของฟังก์ชัน “Chang Dir”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.23 ผังงานการทำงานของฟังก์ชัน “ Dos shell ”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.24 ผังงานของการทำงานของฟังก์ชัน “Edit/Buffer”

สำหรับฟังก์ชันอื่น ๆ มีลักษณะการทำงานที่ไม่สลับซับซ้อนมากนัก ทางคณะผู้จัดทำจึงไม่ขออธิบายในส่วนนี้ และรายละเอียดของโปรแกรมดูได้จากภาคผนวก

3.6.2 ส่วนของภาษาแอสเซมบลี

ส่วนของภาษาแอสเซมบลีของ MCS-51 นั้น ประกอบไปด้วยส่วนย่อย ๆ หลักของโปรแกรมได้ 4 ส่วน

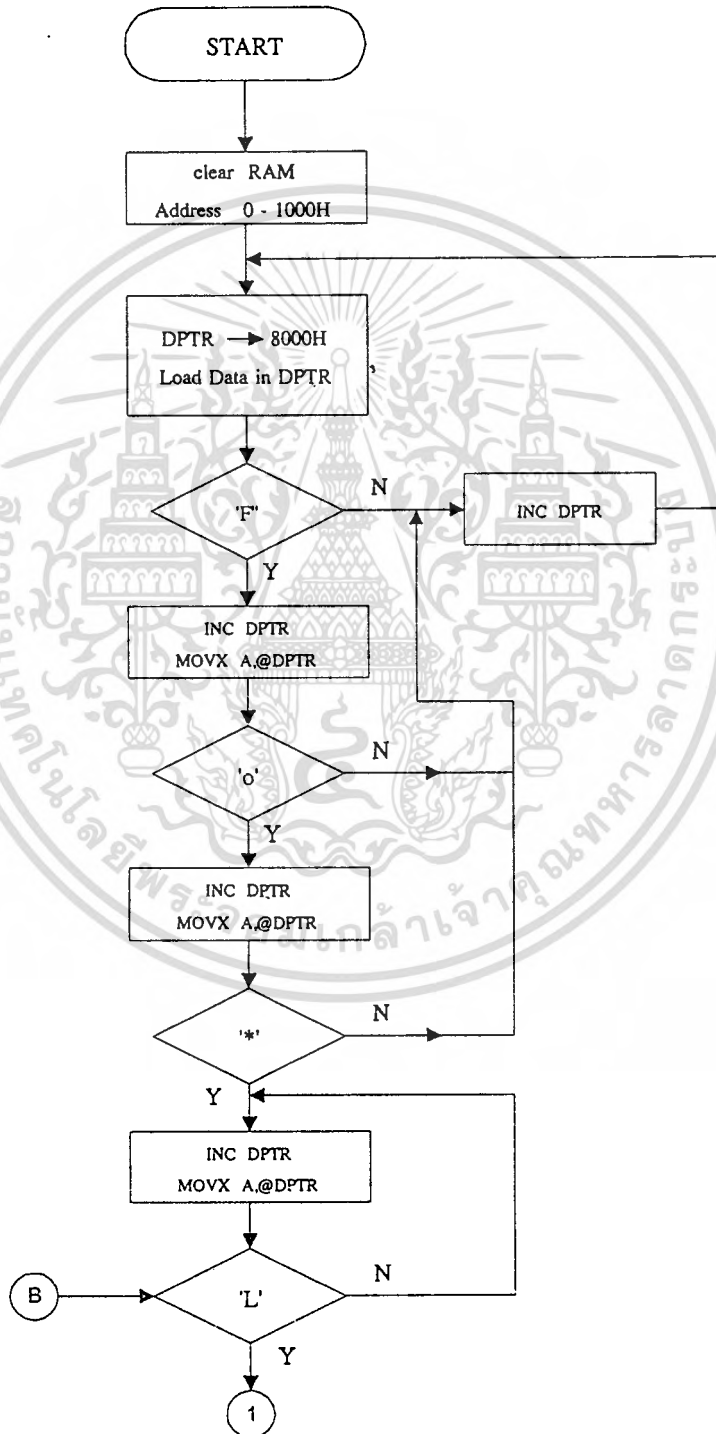
1. การจัดเก็บข้อมูล JEDEC FILE ลงในแรม (RAM)
2. การแสดงผลของ JEDEC ที่จัดเก็บไปแล้ว

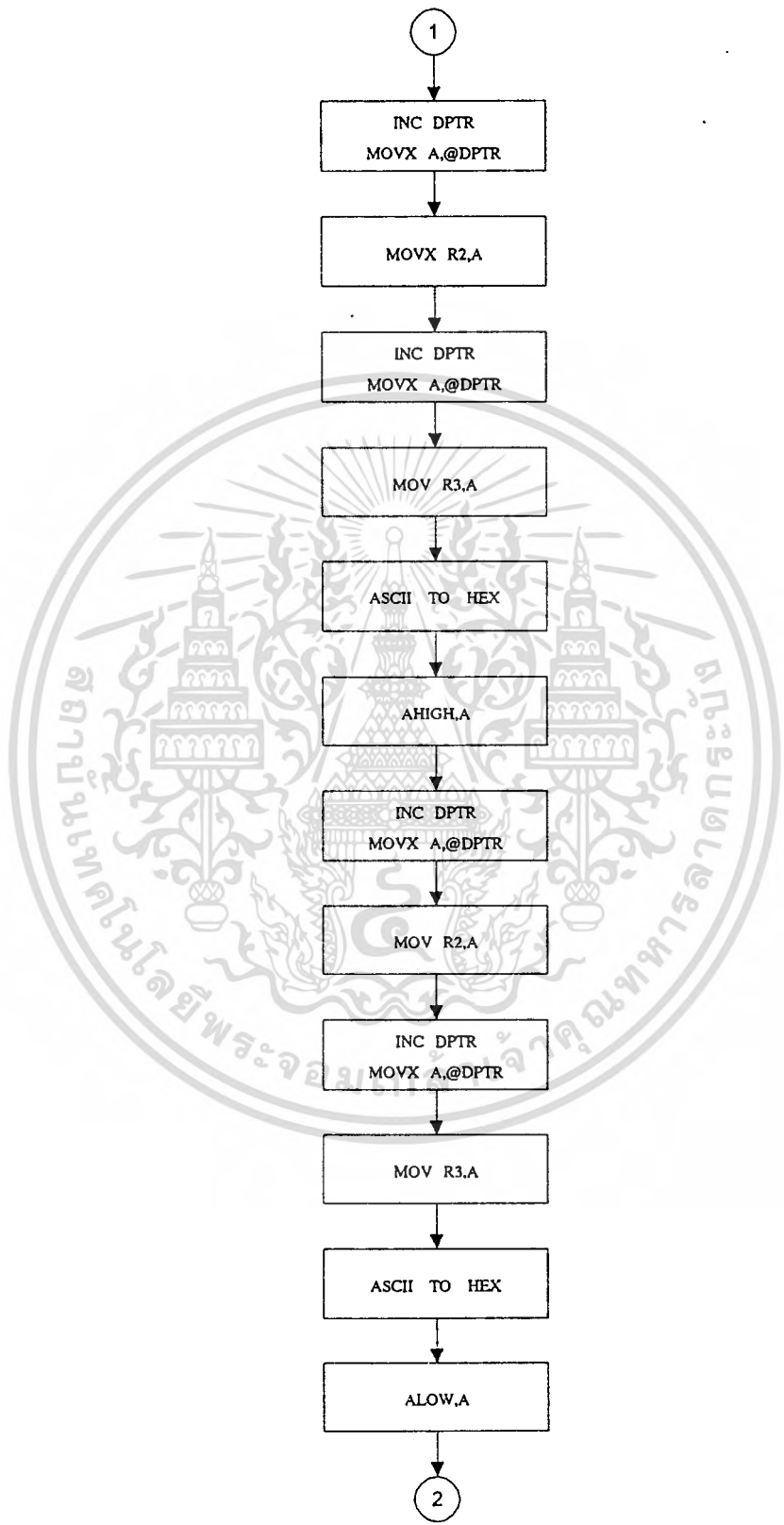
3. การโปรแกรมระเบิดฟิวส์ 1 ตำแหน่ง

เอกสารนี้เป็นเอกสารของบริษัทเอกชนที่ออกให้เพื่อการใช้งานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

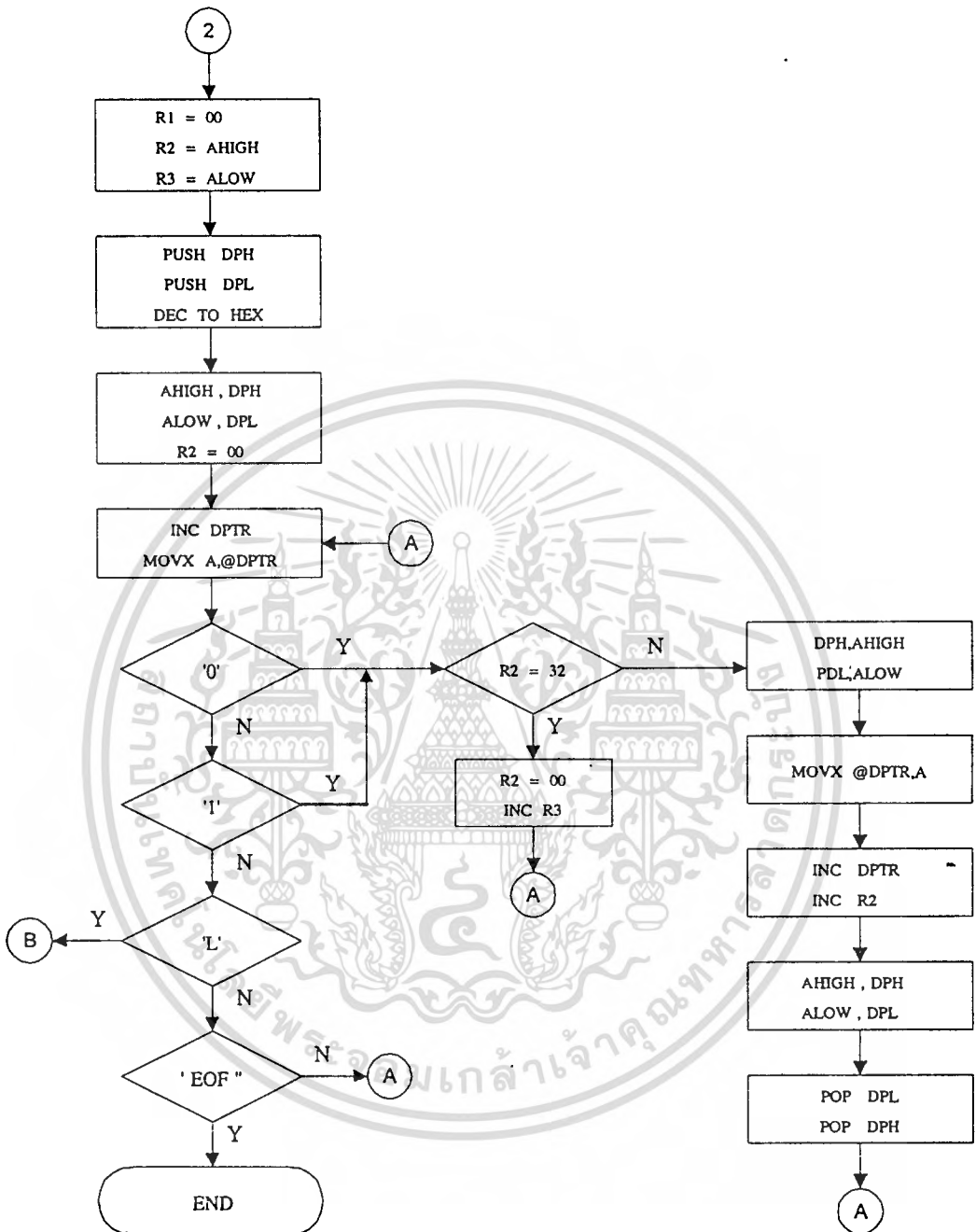
4. การโปรแกรมพีวส์แบบชุดทั้งตัว

ขั้นตอนการทำงานแต่ละส่วนได้แสดงการทำงานเป็นผังงานดังนี้

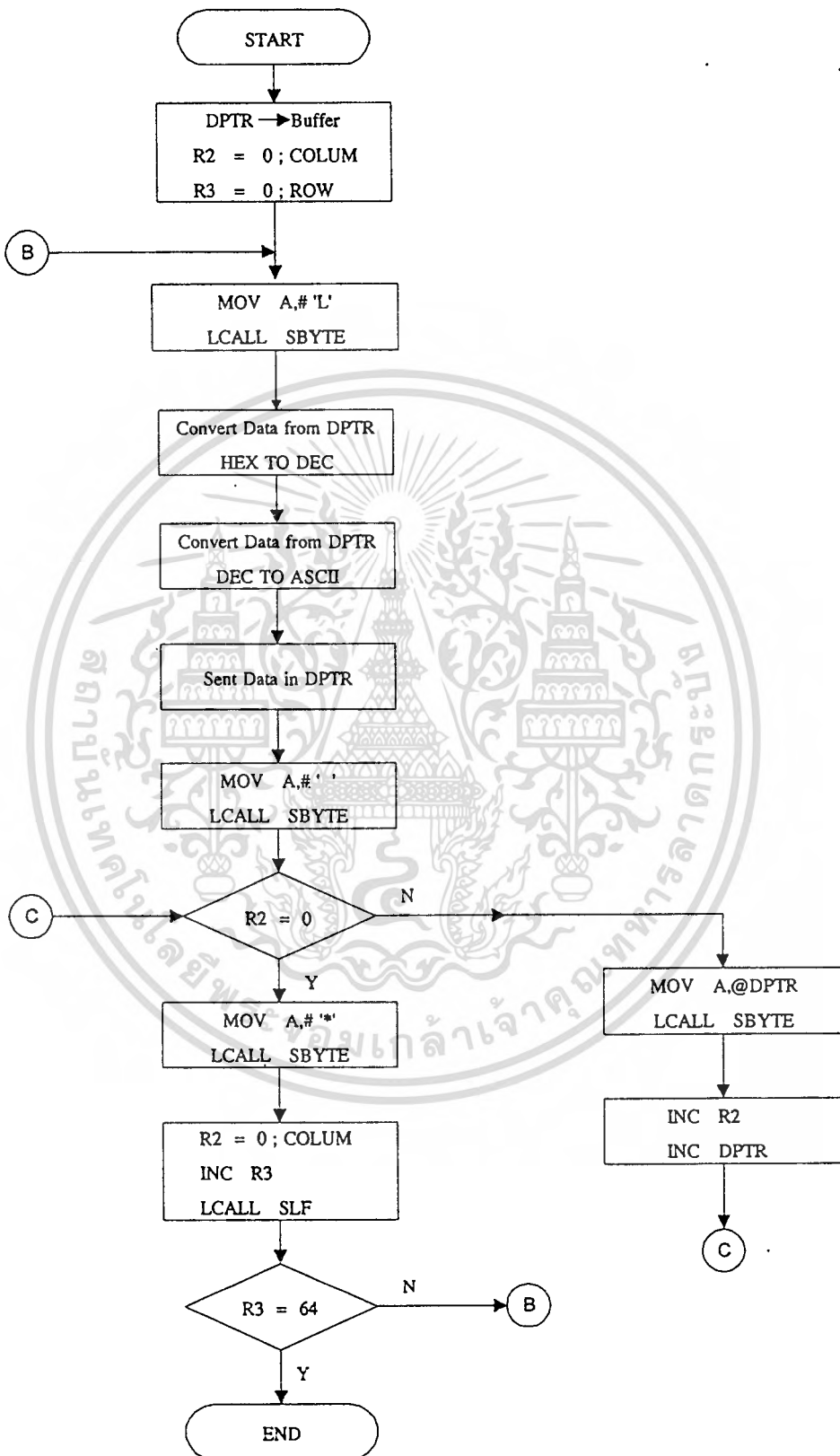


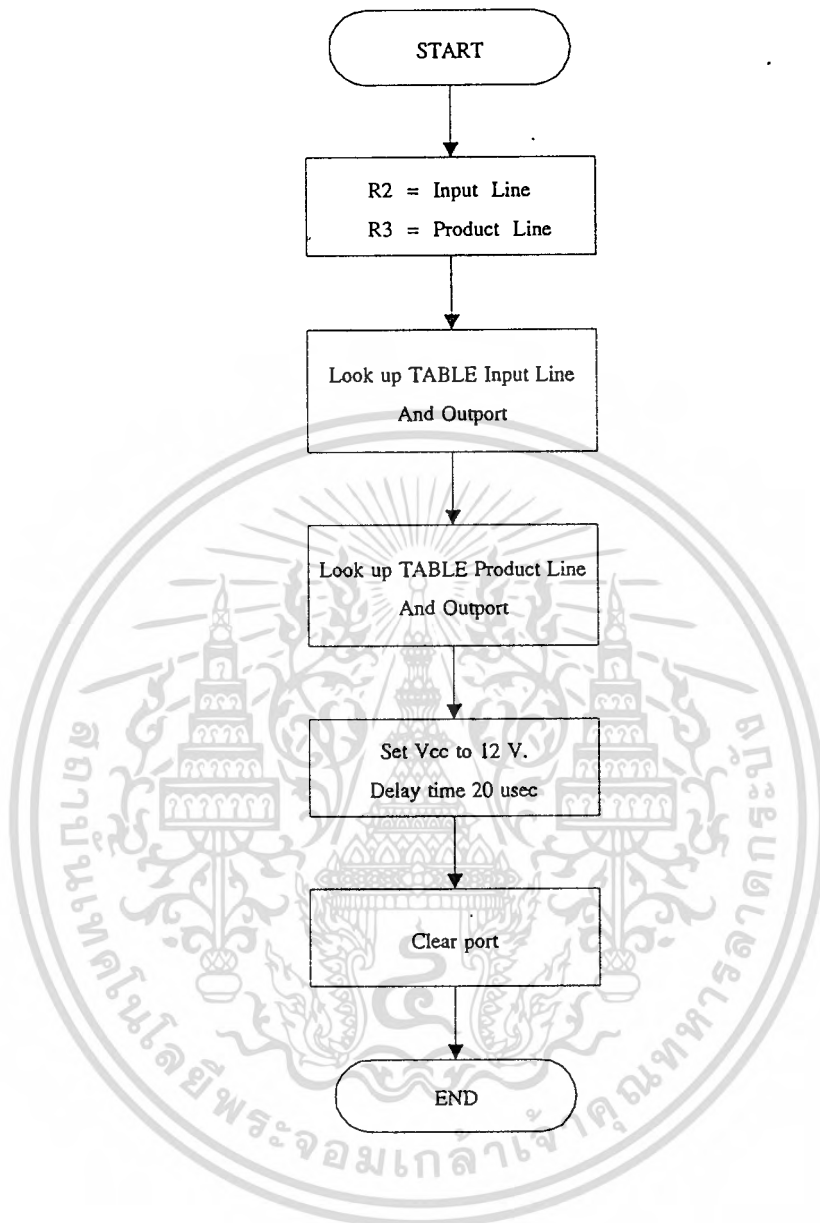


รูปที่ 3.25 ผังงานการเก็บข้อมูล JEDEC FILE ลงในแรม (ต่อ)

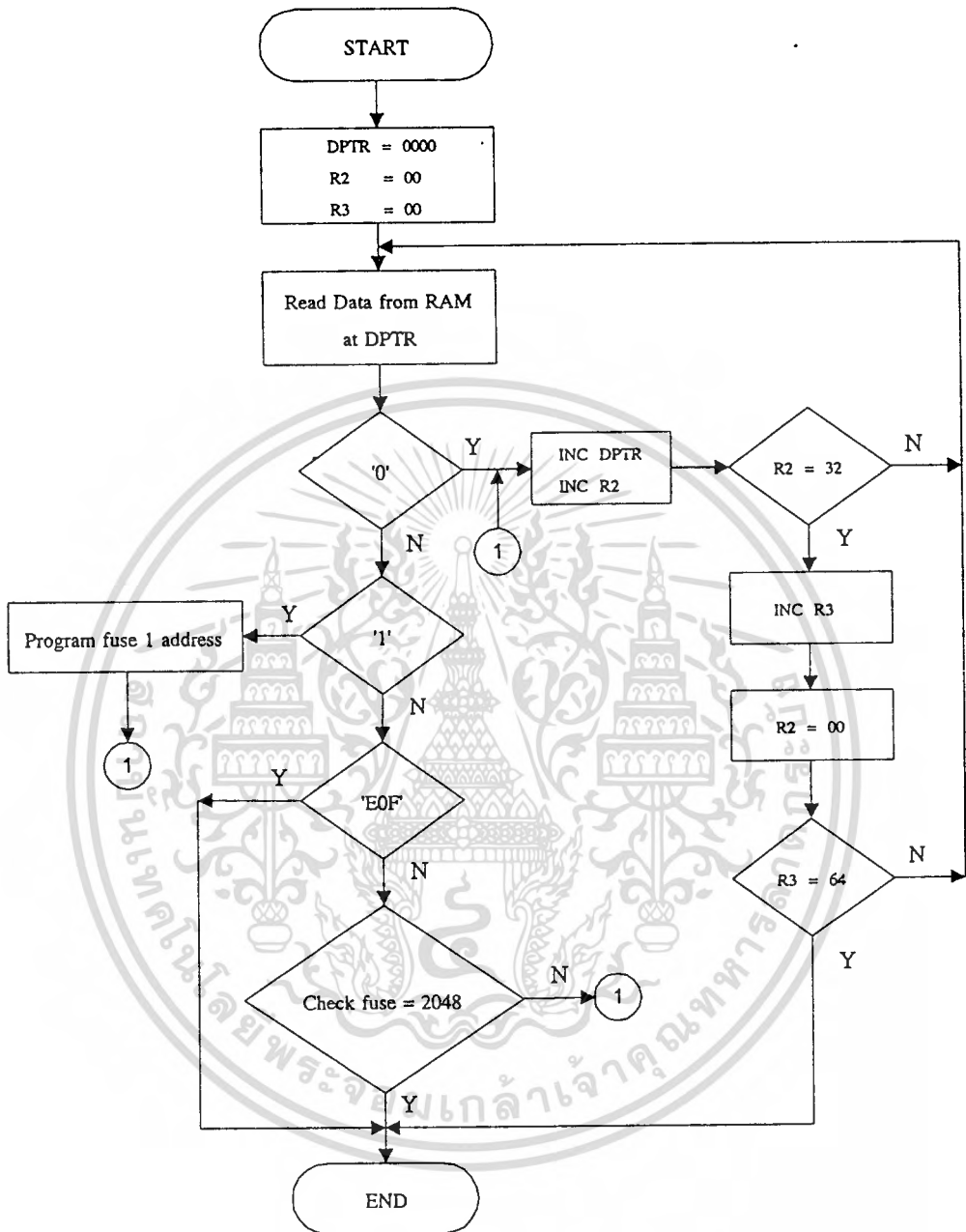


รูปที่ 3.25 ฟังงานการเก็บข้อมูล JEDEC FILE ลงในแรม (ต่อ)





รูปที่ 3.27 ผังงานการโปรแกรมระเบิดฟิวส์ขนาด 1 ตำแหน่ง



รูปที่ 3.28 ฟังก์ชันการโปรแกรมระเบิดฟิวส์แบบชุดทั้งตัว

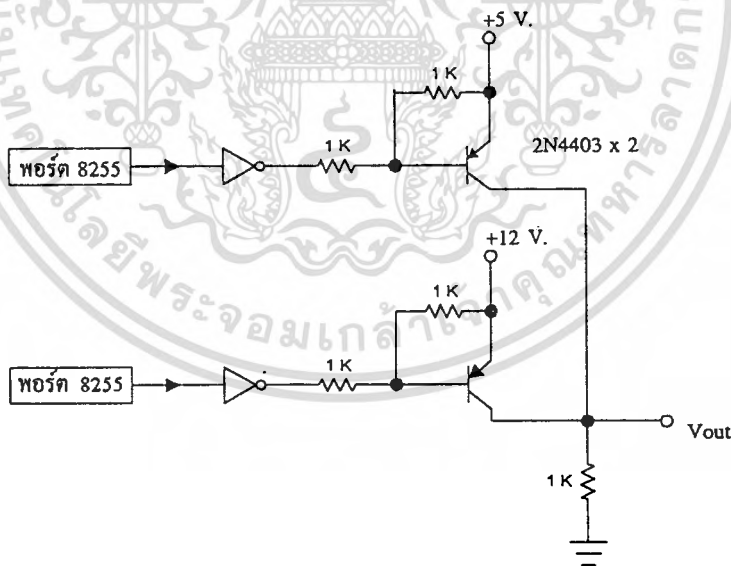
บทที่ 4

ผลการทดลองและทดสอบ

ในบทนี้จะเป็นการกล่าวถึงการทดลองการทำงาน ของวงจรในส่วนต่าง ๆ ว่าผลของการทดลองที่ได้เป็นไปตามที่ได้ออกแบบไว้หรือไม่ ซึ่งสามารถแบ่งออกเป็น การทดลองและผลของการทดลองในแต่ละส่วนได้เป็นดังรูปที่ 4.1

4.1 การทดลองวงจรทรานซิสเตอร์ควบคุมระดับแรงไฟ

ในการทดลองนี้ได้นำวงจรที่ออกแบบไว้ในบทที่ 3 มาทดสอบว่าได้ค่าแรงดันทางด้านเอาต์พุต ตามต้องการหรือไม่ โดยมีลักษณะวงจรดังรูปที่ 4.1 และ รูปที่ 4.2



รูปที่ 4.1 วงจรไฟ 3 ระดับ

ทำการป้อนแรงดันไฟค่าต่าง ๆ ตามตาราง

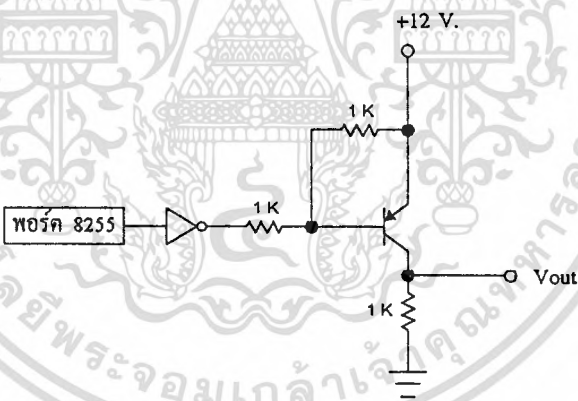
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

Input		Output
A	B	C
L	L	0.45 V.
L	H	12.01 V.
H	L	4.98 V.
H	H	11.8 V.

ตารางที่ 4.1 ผลการทดลองวงจรไฟ 3 ระดับ

สรุป ผลการทดลองวงจรไฟ 3 ระดับเป็นไปตามที่ได้ออกแบบไว้



รูปที่ 4.2 วงจรไฟ 2 ระดับ

ผลการทดลอง

Vin	Vout
L	12 V.
H	1.3 V.

สรุป ผลการทดลองวงจรไฟ 2 ระดับ เป็นไปตามที่ออกแบบไว้

หมายเหตุ

แรงดันไฟประมาณ 0 - 1.2 V. คือ ระดับลอจิก L

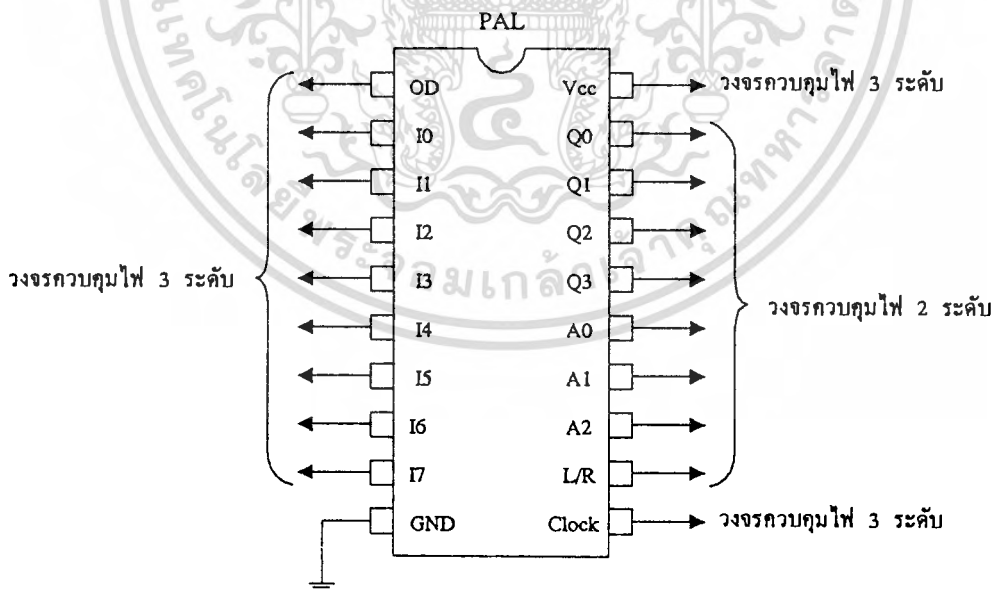
แรงดันไฟประมาณ 2.4 - 5 V. คือ ระดับลอจิก H

แรงดันไฟประมาณ 11 - 12 V. คือ ระดับลอจิก HH

แรงดันไฟประมาณ 1.3 - 2.3 V. คือ ระดับลอจิก Z

4.2 การทดลองวงจรร่วมกันขนาดใหญ่

การทดลองวงจรในส่วนนี้ เนื่องจากมีขนาดใหญ่ จึงไม่สามารถทดลองด้วยมือได้ จำเป็นต้องใช้ซอฟต์แวร์ช่วยในการทดสอบ โดยโปรแกรมนั้นทำหน้าที่ ตรวจสอบแรงดันไฟ ที่ขาต่าง ๆ ที่ละจุดว่าเป็นไปตามที่ออกแบบหรือเปล่า โดยใช้มัลติมิเตอร์วัดเปรียบเทียบกับค่าที่พิมพ์หน้าจอ จนครบทุกขา โปรแกรมที่กล่าวถึงนี้ชื่อ TEST.ASM สามารถดูรายละเอียดได้จากภาคผนวก โดยรูปวงจรที่ใช้ทดลองมีดังรูปที่ 4.3



รูปที่ 4.3 วงจรร่วมขนาดใหญ่ที่ใช้ในการทดลอง

ตำแหน่งขา	ค่าแรงดันที่ปรากฏหน้าจอ (V)	ค่าแรงดันที่วัดได้ (V)
3	0	0
	5	4.8
	12	11.9
4	0	0
	5	4.8
	12	11.9
5	0	0
	5	4.8
	12	11.9
6	0	0
	5	4.8
	12	11.9
7	0	0
	5	4.8
	12	11.9
8	0	0
	5	4.8
	12	11.9
9	0	0
	5	4.8
	12	11.9
10	0	0
	5	4.8
	12	11.9
11	0	0
	5	4.8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

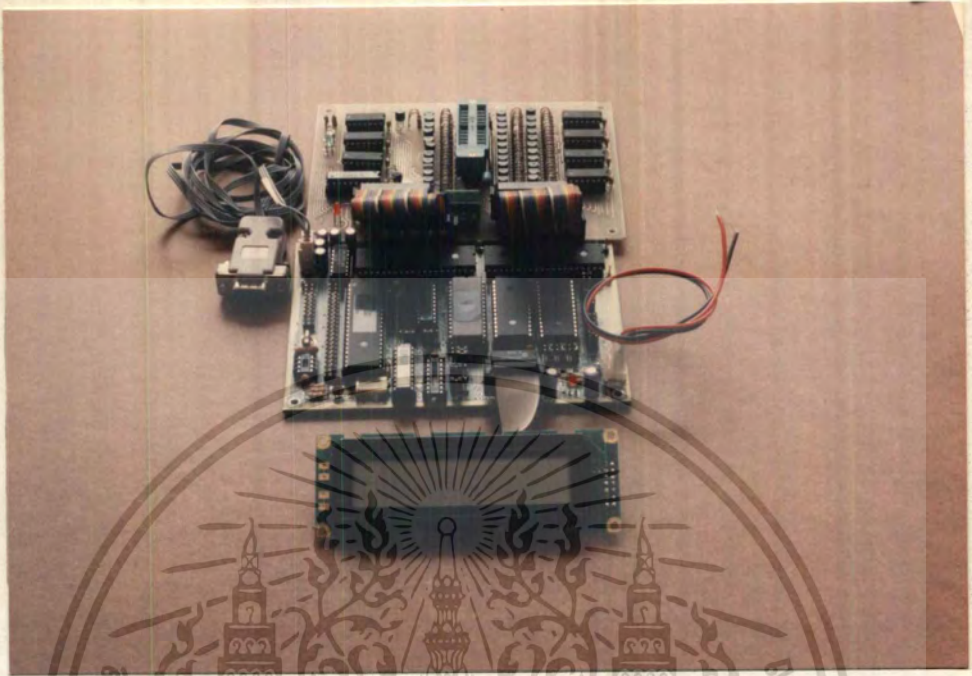
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งขา	ค่าแรงดันที่ปรากฏหน้าจอ (V)	ค่าแรงดันที่วัดได้ (V)
11	12	11.9
13	0	0
	5	4.8
	12	11.9
14	0	0
	12	11.9
15	0	0
	12	11.9
16	0	0
	12	11.9
17	0	0
	12	11.9
18	0	0
	12	11.9
19	0	0
	12	11.9
20	0	0
	12	11.9
21	0	0
	12	11.9
22	0	0
	5	4.8
	12	11.9

ตารางที่ 4.3 ผลการทดสอบบอร์ดโดยใช้โปรแกรม TEST.ASM เทียบกับค่าที่วัดได้

หมายเหตุ

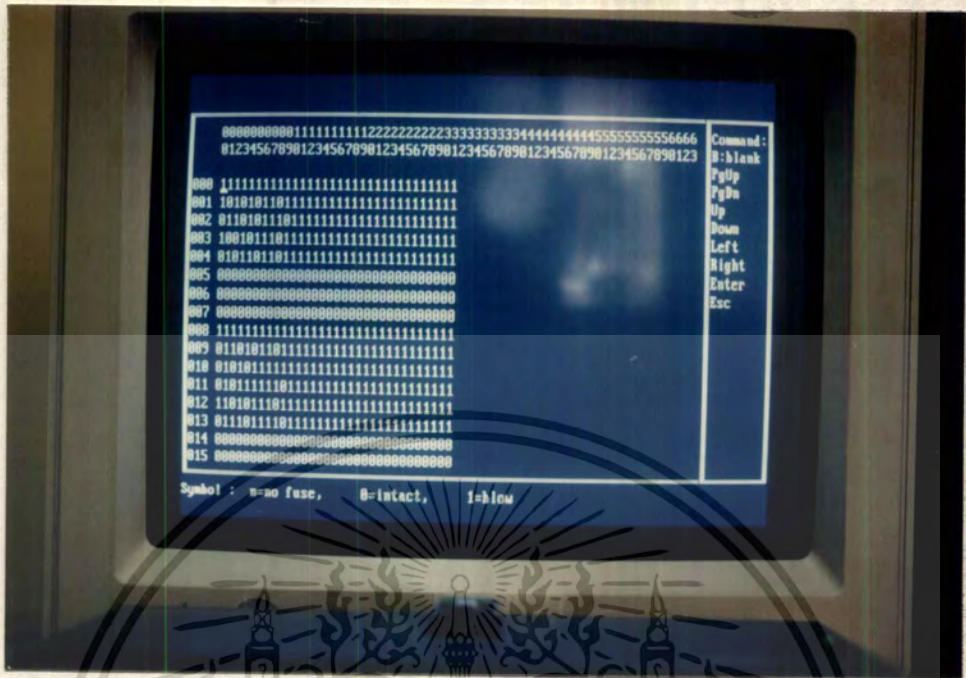
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้ภายในเท่านั้น ไม่ให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.4 บอร์ดวงจรรวมขนาดใหญ่ที่ใช้ในการทดลอง

4.3 ทดสอบการระเบิดฟิวส์ครึ่งละ 1 ตำแหน่ง

การเข้าถึงตำแหน่งฟิวส์ของไอซี PAL โดยดูได้จากตารางที่ 2.5-2.6 หน้าที่ 39 - 40 และนำค่านั้นมาคิดเป็นค่าที่จะทำการควบคุม และทำการส่งออกพอร์ต ไปควบคุมวงจรทรานซิสเตอร์โดยค่าต่าง ๆ ที่ใช้คิดในการส่งออกพอร์ตนั้น สามารถดูได้จากท้ายโปรแกรม ประกอบกับตารางบทที่ 2 และ การทดลองนี้จะทดสอบการโปรแกรมฟิวส์แบบสุ่มตำแหน่ง ว่าตำแหน่งนั้นถูกระเบิดจริงหรือไม่ การทดลองระเบิดฟิวส์จะสุ่มประมาณ 5 ตำแหน่ง เพื่อทดสอบความถูกต้องในการคิดค่าที่ส่งออกจากพอร์ต ในการทดสอบว่าการระเบิดตำแหน่งฟิวส์นั้นถูกต้องหรือไม่ สามารถทดสอบได้จากการนำไปทดสอบกับเครื่องมาตรฐาน คือ EXPRO - 60 ได้ผลการทดลองดังรูปที่ 4.6



รูปที่ 4.5 การทดสอบอุปกรณ์ PAL ที่ผ่านการโปรแกรมโดยใช้เครื่อง EXPRO-60



เอกสารนี้เป็นเอกสารที่สงวนไว้รูปที่ 4.6 ผลการทดสอบการระเบิดฟิวส์ 5 ตำแหน่งใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ทดสอบการระเบิดฟิวส์ครั้งละ 1 เอาต์พุต

ในหัวข้อนี้จะเป็นการออกแบบทดลองการระเบิดฟิวส์ทีละ 1 เอาต์พุตโดยให้อาต์พุตนั้น เป็นเกตพื้นฐาน เช่น แอนด์เกต ออร์เกต แนนด์เกต โดยการออกแบบสร้างวงจร JEDEC ไฟล์จากโปรแกรม OrCAD ในส่วน PLD และทำการโหลด JEDEC ไฟล์ ที่สร้างลงสู่บอร์ดไมโคร MCS-51 และทำการระเบิดฟิวส์ตามลักษณะของ JEDEC ไฟล์ ที่เขียนสร้างวงจรเกตจาก OrCAD ในส่วนของ PLD มีการเขียนดังนี้

ตัวอย่างโปรแกรม GATES.PLD ที่ใช้ทดสอบ

```
PAL 16L8      1:A, 2:B, 3:C, 4:D, 5:E, 6:F, 17..19:Q[3..1]
|  Q1 = A & B      : And
|  Q2 = C # D      : Or
|  Q3 = (E & F)'    : Nand
```

โปรแกรม GATES.PLD ที่ผ่านการแปลเป็นไฟล์ JEDEC หรือ GATES.JED

```
QP20* QF2048* QV1024*
F0*
L0000  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 *
L0032  11 10 11 11 11 11 11 11 11 11 11 11 11 11 11 11 *
L0064  10 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 *
L0256  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 *
L0288  11 11 10 11 10 11 11 11 11 11 11 11 11 11 11 11 *
L0512  11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 *
L0544  11 11 11 11 11 11 01 11 01 11 11 11 11 11 11 11 *
C1BA7*
```

โปรแกรมตรวจสอบการทำงานวงจร GATES.PLD

OrCAD TEST VECTOR GENERATOR V1.20E 6/6/90

| { Display Q[3~1], " -> ",A,B,C,D,E,F

| Test A,B,C,D,E,F

100 -> 0 0 0 0 0 0

100 -> 0 0 0 0 0 1

100 -> 0 0 0 0 1 0

000 -> 0 0 0 0 1 1

110 -> 0 0 0 1 0 0

110 -> 0 0 0 1 0 1

110 -> 0 0 0 1 1 0

010 -> 0 0 0 1 1 1

110 -> 0 0 1 0 0 0

110 -> 0 0 1 0 0 1

110 -> 0 0 1 0 1 0

010 -> 0 0 1 0 1 1

110 -> 0 0 1 1 0 0

110 -> 0 0 1 1 0 1

110 -> 0 0 1 1 1 0

010 -> 0 0 1 1 1 1

100 -> 0 1 0 0 0 0

100 -> 0 1 0 0 0 1

100 -> 0 1 0 0 1 0

000 -> 0 1 0 0 1 1

110 -> 0 1 0 1 0 0

110 -> 0 1 0 1 0 1

110 -> 0 1 0 1 1 0

010 -> 010111

110 -> 011000

110 -> 011001

110 -> 011010

010 -> 011011

110 -> 011100

110 -> 011101

110 -> 011110

010 -> 011111

100 -> 100000

100 -> 100001

100 -> 100010

000 -> 100011

110 -> 100100

110 -> 100101

110 -> 100110

010 -> 100111

110 -> 101000

110 -> 101001

110 -> 101010

010 -> 101011

110 -> 101100

110 -> 101101

110 -> 101110

010 -> 101111

101 -> 110000

101 -> 110001

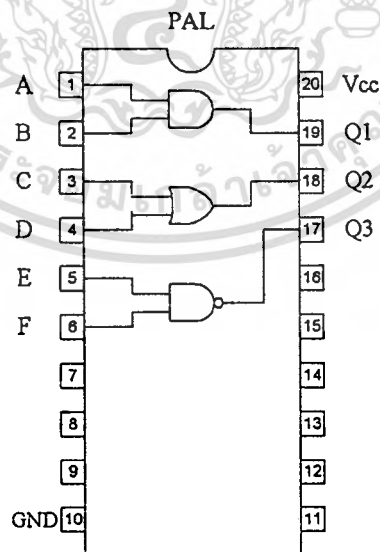
101 -> 110010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

001 -> 1 1 0 0 1 1
111 -> 1 1 0 1 0 0
111 -> 1 1 0 1 0 1
111 -> 1 1 0 1 1 0
011 -> 1 1 0 1 1 1
111 -> 1 1 1 0 0 0
111 -> 1 1 1 0 0 1
111 -> 1 1 1 0 1 0
011 -> 1 1 1 0 1 1
111 -> 1 1 1 1 0 0
111 -> 1 1 1 1 0 1
111 -> 1 1 1 1 1 0
011 -> 1 1 1 1 1 1
| End }

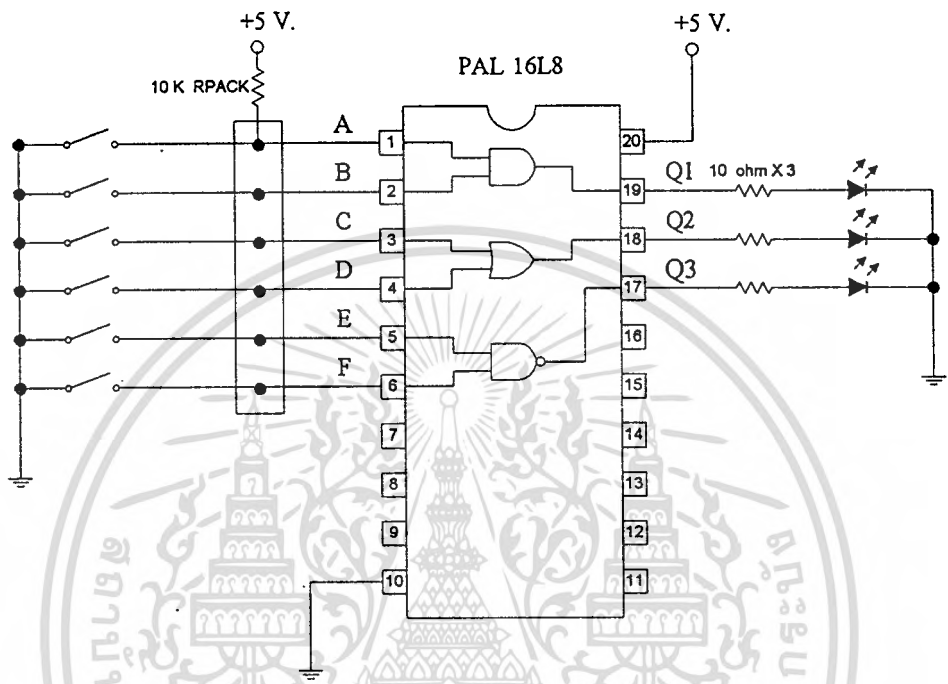
```



รูปที่ 4.7 ลักษณะโครงสร้างภายในของอุปกรณ์ PAL เมื่อผ่านการโปรแกรมจากไฟล์ GATES.JED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากการโปรแกรมไฟล์ GATES.JED สามารถทดสอบวงจรได้ดังรูปที่ 4.8



รูปที่ 4.8 วงจรทดสอบหลังการโปรแกรมไฟล์ GATES.JED

INPUT		OUTPUT
A	B	Q1
0	0	0
0	1	0
1	0	0
1	1	1

INPUT		OUTPUT
C	D	Q2
0	0	0
0	1	1
1	0	1
1	1	1

INPUT		OUTPUT
E	F	Q3
0	0	1
0	1	0
1	0	0
1	1	0

ตารางที่ 4.4 ผลการทดสอบอุปกรณ์ PAL หลังการโปรแกรมไฟล์ GATES.JED

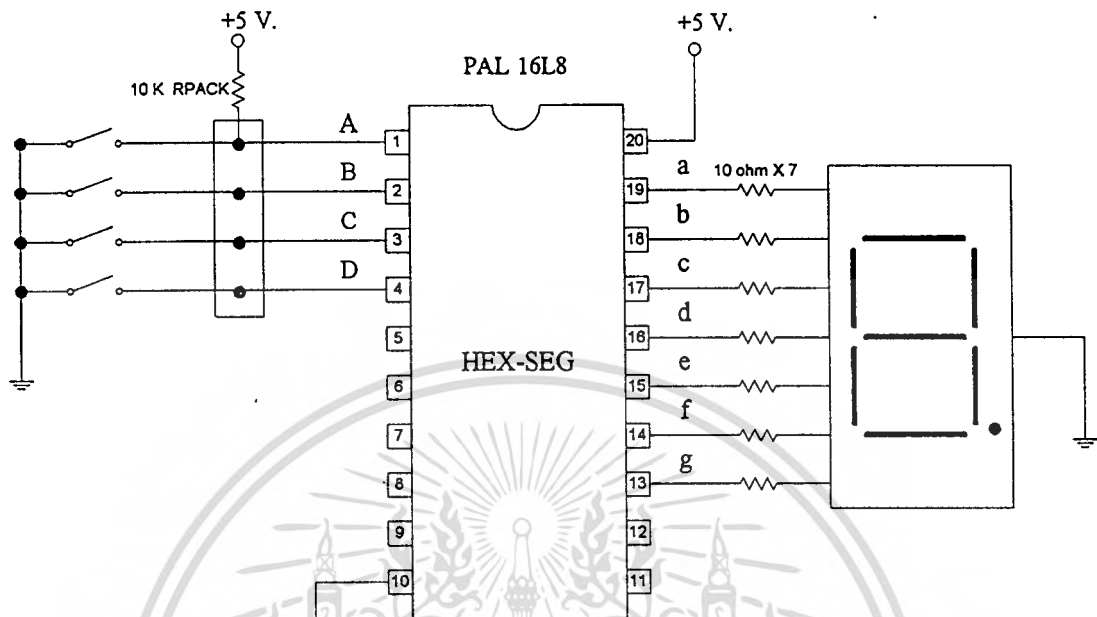
4.4.1 วงจรถอดรหัสฐานสิบหกแสดงผลออก 7 เซกเมนต์

การสร้างวงจรถอดรหัสฐานสิบหกออก 7 เซกเมนต์ สามารถออกแบบได้โดยการเขียนโปรแกรม HEX-SEG.PLD หลังจากการโปรแกรมไฟล์ HEX-SEG.JED สามารถทดสอบวงจรได้ดังรูปที่ 4.9

โปรแกรม HEX-SEG.PLD

```
[PAL16L8 1:A,2:B,3:C,4:D,19:Qa,18:Qb,17:Qc,16:Qd,15:Qe,14:Qf,13:Qg
|TABLE:A,B,C,D -> Qg,Qf,Qe,Qd,Qc,Qb,Qa
|{ 0000B -> 0111111B
| 0001B -> 0000110B
| 0010B -> 1011011B
| 0011B -> 1001111B
| 0100B -> 1100110B
| 0101B -> 1101101B
| 0110B -> 1111101B
| 0111B -> 0000111B
| 1000B -> 1111111B
| 1001B -> 1101111B
| 1010B -> 1110111B
| 1011B -> 1111100B
| 1100B -> 0111001B
| 1101B -> 1011110B
| 1110B -> 1111001B
| 1111B -> 1110001B }
|VECTORS:
|{ DISPLAY A,B,C,D," ->", Qg,Qf,Qe,Qd,Qc,Qb,Qa
| Test A,B,C,D
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
| END }
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 วงจรทดสอบอุปกรณ์ PAL หลังจากการโปรแกรมไฟล์ HEX-SEG.JED

HEX	INPUT				OUTPUT							DATA
	A	B	C	D	g	f	e	d	c	b	a	
0	0	0	0	0	0	1	1	1	1	1	1	3FH
1	0	0	0	1	0	0	0	0	1	1	0	06H
2	0	0	1	0	1	0	1	1	0	1	1	5BH
3	0	0	1	1	1	0	0	1	1	1	1	4FH
4	0	1	0	0	1	1	0	0	1	1	0	66H
5	0	1	0	1	1	1	0	1	1	0	1	6DH

ตารางที่ 4.5 ผลการทดลองอุปกรณ์หลังจากโปรแกรมไฟล์ HEX-SEG.JED

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HEX	INPUT				OUTPUT							DATA
	A	B	C	D	g	f	e	d	c	b	a	
7	0	1	1	1	0	0	0	0	1	1	1	07H
8	1	0	0	0	1	1	1	1	1	1	1	7FH
9	1	0	0	1	1	1	0	1	1	1	1	6FH
A	1	0	1	0	1	1	1	0	1	1	1	77H
b	1	0	1	1	1	1	1	1	1	0	0	7CH
C	1	1	0	0	0	1	1	1	0	0	1	39H
d	1	1	0	1	1	0	1	1	1	1	0	5EH
E	1	1	1	0	1	1	1	1	0	0	1	79H
F	1	1	1	1	1	1	1	0	0	0	1	71H

ตารางที่ 4.5 ผลการทดสอบอุปกรณ์หลังจากโปรแกรมไฟล์ HEX-SEG.JED (ต่อ)

4.4.2 วงจรถอดรหัส

การสร้างวงจรถอดรหัสสามารถออกแบบได้ โดยเขียนโปรแกรม DECODER.PLD และทำการทดสอบวงจรหลังจากการโปรแกรมได้ดังรูปที่ 4.10

โปรแกรม DECODER.PLD

```
[pal1618 1..3:a[1..3],19..12:q[1..8]
|table: a[1..3] -> q[1..8]
|{ 000b -> 00000001b
| 001b -> 00000010b
| 010b -> 00000100b
| 011b -> 00001000b
| 100b -> 00010000b
```

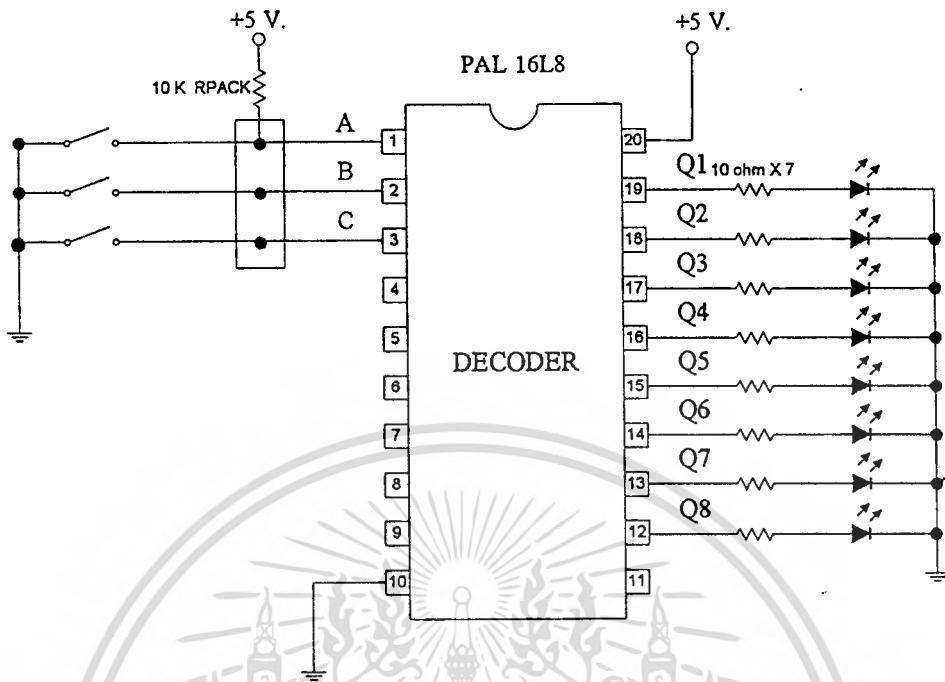
```
| 101b -> 001000000b
| 110b -> 010000000b
| 111b -> 100000000b }
```

```
|vectors:
```

```
|{ display a[1..3],q[1..8]
| test a[1..3]
| end}
```

OrCAD TEST VECTOR GENERATOR V1.20E 6/6/90

```
|{ display a[1..3],q[1..8]
| test a[1..3]
000 00000001
001 00000010
010 00000100
011 00001000
100 00010000
101 00100000
110 01000000
111 10000000
| end}
```



รูปที่ 4.10 วงจรทดสอบอุปกรณ์ PAL หลังการโปรแกรมไฟล์ DECODER.PLD

INPUT			OUTPUT							
A	B	C	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

ตารางที่ 4.6 ผลการทดสอบอุปกรณ์หลังการโปรแกรมไฟล์ DECODER.PLD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองครั้งแรกปรากฏว่าอุปกรณ์ PAL ที่ออกแบบวงจรไว้แล้วไม่สามารถใช้งานจริงได้ ถึงแม้ว่าผ่านการทดสอบจากเครื่องมาตรฐานแล้วก็ตาม ดังนั้นจึงได้ทำการเปลี่ยนแปลงเวลาที่โปรแกรมตามตารางที่ 4.7 จากเดิมใช้เวลาในการโปรแกรม 60 uSec

ครั้งที่	เวลาในการทดลอง	ผลการทดลองหลังโปรแกรม
1	80 uSec	- ใช้งานไม่ได้
2	120 uSec	- ใช้งานไม่ได้
3	50 uSec	- ใช้งานไม่ได้
4	40 uSec	- ใช้งานได้เป็นประมาณ 1 : 3 คือ 4.1 ทดลองครั้งที่ 1 ใช้งานได้ 4.2 ทดลองครั้งที่ 2 ใช้งานไม่ได้ 4.3 ทดลองครั้งที่ 3 ใช้งานไม่ได้
5	30 uSec	- ใช้งานได้เป็นประมาณ 2 : 3 คือ 5.1 ทดลองครั้งที่ 1 ใช้งานได้ 5.2 ทดลองครั้งที่ 2 ใช้งานไม่ได้ 5.3 ทดลองครั้งที่ 3 ใช้งานได้
6	20 uSec	- ใช้งานได้เป็นประมาณ 3 : 3 คือ 6.1 ทดลองครั้งที่ 1 ใช้งานได้ 6.2 ทดลองครั้งที่ 2 ใช้งานได้ 6.3 ทดลองครั้งที่ 3 ใช้งานได้

ตารางที่ 4.7 ผลการทดสอบการใช้งาน

จากการทดสอบวงจรจากชุดทดลอง ปรากฏว่าอุปกรณ์ที่เราต้องการออกแบบสามารถใช้งานได้จริง

บทที่ 5

สรุปอภิปรายและข้อเสนอแนะ

5.1 สรุป

จากการที่ได้ศึกษาถึงลักษณะโครงสร้างภายในของอุปกรณ์ PAL และได้เริ่มทำการออกแบบทดลองสร้างเครื่องโปรแกรมอุปกรณ์ PAL ขึ้นมาพร้อมทั้งได้เขียนโปรแกรมควบคุมการทำงานของเครื่องโปรแกรมอุปกรณ์ PAL จนสามารถที่จะใช้งานได้จริงตามขอบเขตของงานที่คณะผู้จัดทำได้วางเอาไว้ โดยที่ชุดโปรแกรมอุปกรณ์ PAL ประกอบด้วยส่วนต่างๆ ดังนี้

1. ซอฟต์แวร์ภาษาแอสเซมบลี MCS-51 ซึ่งเป็นส่วนของโปรแกรมหลักที่ใช้ควบคุมการส่งข้อมูลไปโปรแกรมอุปกรณ์ PAL
2. ซอฟต์แวร์ภาษา C เป็นโปรแกรมส่วนที่ติดต่อกับผู้ใช้งาน เพื่อที่จะให้ผู้ใช้งานสามารถโปรแกรมอุปกรณ์ PAL ได้อย่างสะดวก รวดเร็ว โดยโปรแกรมในส่วนนี้จะมีหน้าที่หลักในการส่งผ่านข้อมูล JEDEC ไฟล์ จากคอมพิวเตอร์ผ่านทาง RS-232 ไปให้กับเครื่องโปรแกรม PAL
3. เครื่องโปรแกรม PAL ซึ่งประกอบด้วยฮาร์ดแวร์ 2 ส่วนคือ
 - 3.1 บอร์ด MCS-51 ทำหน้าที่ควบคุมการส่งข้อมูลที่ใช้ในการโปรแกรมอุปกรณ์ PAL
 - 3.2 บอร์ดชุดโปรแกรมอุปกรณ์ PAL ทำหน้าที่รับข้อมูลจากบอร์ด MCS-51 แล้ว ปรับระดับแรงดันไฟฟ้าให้เหมาะสมกับการโปรแกรมอุปกรณ์ PAL

จากการที่ได้ศึกษาและได้ทดลองปรับปรุงทั้งทางด้านฮาร์ดแวร์และซอฟต์แวร์ และได้ทำการทดสอบโดยการออกแบบวงจร ลอจิกเกต แบบพื้นฐานจากโปรแกรม OrCAD และได้ทำการแปลงให้เป็น JEDEC ไฟล์ แล้วนำข้อมูลที่ได้นำไปทำการโปรแกรมอุปกรณ์ PAL กับเครื่องโปรแกรมอุปกรณ์ PAL ที่ได้สร้างขึ้นมา เมื่อทำการโปรแกรมเสร็จแล้วก็นำไปทดสอบ ผลปรากฏว่าอุปกรณ์ PAL สามารถทำงานได้ตรงกับที่ได้ออกแบบเอาไว้

5.2 คุณสมบัติของเครื่องโปรแกรมอุปกรณ์ PAL

1. ใช้กับระบบไฟฟ้า 220 V. 50 Hz
2. สามารถติดต่อกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม
3. สามารถใช้โปรแกรมได้กับอุปกรณ์ PAL เพียงอย่างเดียว โดยมีข้อกำหนดดังนี้
 - เป็นผลิตภัณฑ์ PAL ของบริษัท MMI NS และ AMD
 - เบอร์ที่สามารถโปรแกรมได้คือ 16L8 , 16R4 , 16R6 , 16R8 , 20R4 , 20R6 , 20L8 , 20R8
 - สามารถใช้โปรแกรมได้กับ PAL ที่สามารถโปรแกรมได้ครั้งเดียว
5. สามารถโปรแกรมได้ครั้งละ 1 ตัว
6. สามารถรับ JEDEC ไฟล์ ที่ใช้ในการโปรแกรมจาก OrCAD/PLD และ OPAL
7. มี LCD แสดงสถานะการทำงาน
8. สามารถใช้ได้กับเครื่อง IBM Compatible

5.3 ปัญหา และ แนวทางแก้ไข

ปัญหาที่ 1. ในขณะที่ทำการระเบิดฟิวส์แต่ละตำแหน่งเดียวนั้นจะพบปัญหาคือ เวลาที่ใช้ในการระเบิดฟิวส์นั้นนานเกินไปทำให้อุปกรณ์ PAL เสียหาย

แนวทางแก้ไข ทำการสุ่มเพื่อหาค่าเวลาที่เหมาะสม

ปัญหาที่ 2. เพาเวอร์ซัพพลาย (Power supply) ที่ใช้งานในครั้งแรกนั้นจ่ายกระแสได้น้อยเกินไปทำให้ฟิวส์ไม่ระเบิด

แนวทางแก้ไข สั่งทำหม้อแปลงขนาดตามที่ต้องการคือ 3 แอมป์

ปัญหาที่ 3. การต่อสาย RS-232 ในครั้งแรกนั้นจะต่อเฉพาะสายที่ใช้งานจริงเท่านั้น ทำให้การรับส่งข้อมูลเกิดการผิดพลาด

แนวทางแก้ไข คู่มือมาตรฐานการต่อสาย RS-232 ได้จากภาคผนวก

5.4 ข้อเสนอแนะและแนวทางการพัฒนา

คณะผู้จัดทำได้พยายามที่จะปรับปรุงเครื่องโปรแกรมอุปกรณ์ PAL ให้มีประสิทธิภาพมากที่สุดเท่าที่จะทำได้ ในระยะเวลาและงบประมาณที่จำกัด จึงสมควรที่จะต้องมีการปรับปรุงพัฒนาให้มีประสิทธิภาพดีขึ้นกว่านี้ จุดที่ควรจะมีการพัฒนาต่อไปนั้นมีดังนี้

1. เนื่องจากเครื่องโปรแกรมอุปกรณ์ PAL มีขีดความสามารถจำกัดคือ สามารถทำการโปรแกรมอุปกรณ์ PAL ได้เพียงอย่างเดียว ดังนั้นจึงควรพัฒนาให้สามารถที่จะตรวจสอบผลของการโปรแกรมโดยให้มีการอ่านอุปกรณ์ PAL ได้
2. ให้สามารถโปรแกรมอุปกรณ์ PLD ชนิดอื่นๆ ได้ เช่น GAL , EPLD และ PROM





ภาคผนวก ก.

คู่มือการใช้งานเครื่องโปรแกรม PAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คู่มือการใช้งานเครื่องโปรแกรม PAL

ความนำ

เครื่องโปรแกรม PAL นี้ เป็นเครื่องที่สร้างขึ้นให้ทำการโปรแกรมอุปกรณ์ PLD สำหรับความสามารถของเครื่องโปรแกรม PAL นี้จะสามารถที่จะโปรแกรมอุปกรณ์ PAL ได้ 3 บริษัท คือ AMD MMI และ NS ซึ่งจะมีไว้ให้เลือกในโปรแกรมการส่งข้อมูลระหว่างไมโครคอมพิวเตอร์กับเครื่องโปรแกรม PAL จะส่งผ่านทางพอร์ทอนุกรม RS-232

การติดตั้งเครื่องโปรแกรม

1. ทำการต่อสายคอนเนคเตอร์ เข้าที่พอร์ทอนุกรม
2. เปิดสวิทช์เครื่องโปรแกรม PAL
3. ทำการ Execute โปรแกรม PAL

วิธีการใช้โปรแกรม

1. ทำการ load โปรแกรมเพื่อเข้าสู่เมนูหลักโดยจะต้องคีย์คำสั่งดังนี้
`<Drive> PALPGM <Enter>`

ที่จอภาพจะปรากฏดังรูป

2. ขั้นตอนการใช้โปรแกรม
 - 2.1 ทำการโหลดโปรแกรมโดยเลื่อนแถบสว่างไปที่ File แล้วกด Enter แล้วเลื่อนแถบสว่างไปที่ Load กด Enter แล้วทำการใส่ชื่อ JEDEC File หรือกด Enter ก็จะแสดงรายชื่อ JEDEC File ให้เลือก เมื่อเลือกแล้วข้อมูลก็จะถูกส่งไปที่เครื่องโปรแกรม PAL ในขณะนี้เครื่องเตรียมพร้อมที่จะทำการโปรแกรม
 - 2.2 เลือกแถบสว่างไปที่ Device ทำการเลือกบริษัทผู้ผลิต PAL และเบอร์ของ PAL ที่ต้องการจะทำการโปรแกรม
 - 2.3 เลือกแถบสว่างไปที่ Send เพื่อเลือกพอร์ทให้ตรงกับ Connector ที่ต่ออยู่ซึ่งมี 2 พอร์ท คือ COM1 และ COM2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4. เลือกแถบสว่างไปที่เมนู RunFunc แล้วทำการเลื่อนแถบสว่างไปที่ Program กด Enter เมื่อทำการโปรแกรมเสร็จก็จะมีเสียงดังที่ลำโพง



ลักษณะหน้าจอที่ติดต่อกับผู้ใช้งาน

รายละเอียดและหน้าที่ของฟังก์ชันต่าง ๆ ที่ใช้งาน

1. Load File

ทำการโหลดเพิ่มข้อมูลที่เป็นฟิวส์แมป (JEDEC File) ไปเก็บไว้ในหน่วยความจำ Buffer ของเครื่อง PC และจะส่งข้อมูลไปยังเครื่องโปรแกรม PAL ด้วย

2. Chang Dir

ทำการย้ายไดเรกทอรีปัจจุบันเพื่อใช้ในการหา JEDEC File ที่ต้องการ

3. Os shell

ออกจาก Dos ชั่วคราว

4. Program Device

ทำการเขียนข้อมูลของฟิวส์แมปจากหน่วยความจำลงไปที่อุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. Display Device

ทำการแสดงค่าของพีวส์แมปที่อ่านมาจากเครื่องโปรแกรม PAL

6. Edit / View

ทำการแสดงค่าพีวส์แมปที่จะนำไปใช้ในการโปรแกรม

7. COM1 COM2

พอร์ทอนุกรมที่ใช้ในการติดต่อกับเครื่องโปรแกรม PAL

8. Quit

ออกจากโปรแกรมสู่ DOS

9. Device

ให้เลือกบริษัทและเบอร์ ไอซีที่ต้องการจะโปรแกรม





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PLD SOFTWARE

โปรแกรม OrCAD / PLD

โปรแกรมที่ใช้สำหรับการออกแบบเพื่อแปลงค่าของลอจิกที่มีการกระทำทางลอจิก การกระทำทางคณิตศาสตร์ หรือ ทฤษฎีของบูลีน จะทำการแปลงจากโปรแกรม PLD ออกมาเป็น JEDEC ไฟล์ การเขียนโปรแกรม PLD ของ OrCAD/PLD จะมีรูปแบบการเขียนที่เข้าใจง่ายสามารถออกแบบวงจรลอจิกได้ตามต้องการ และยังมีฟังก์ชันที่ใช้ในการทำงานของวงจรดิจิทัล ซีควেনเชียล , วงจรนับ และมีโปรแกรม ที่ใช้ในการตรวจสอบเวกเตอร์เพื่อตรวจสอบโปรแกรมว่าได้ผลตามที่ต้องการหรือไม่ ลักษณะการรายงานผลจะมีทั้งทางจอภาพและเครื่องพิมพ์ สิ่งที่สำคัญคือ OrCAD/PLD สามารถโปรแกรมอุปกรณ์ PLD ได้ทุกเบอร์ ซึ่งสะดวกแก่การใช้งานมาก

สัญลักษณ์ของตัวกระทำ

- & แทนการ AND กัน เช่น $A \& B$
- # แทนการ OR กัน เช่น $A \# B$
- ' แทนการ NOT กัน เช่น A'
- ## แทนการ XOR กัน เช่น $A \# \# B$
- &' แทนการ NAND กัน เช่น $(A \& B)'$
- #' แทนการ NOR กัน เช่น $(A \# B)'$
- ##' แทนการ XNOR กัน เช่น $(A \# \# B)'$
- = แทนการกำหนดค่า เช่น $F = A \# B$
- = = แทนการกำหนดค่าในเงื่อนไข เช่น $F = = 1$
- /= แทนเครื่องหมาย $\neq$
- <= มากกว่าหรือเท่ากับ
- < น้อยกว่า
- >= มากกว่าหรือเท่ากับ
- > มากกว่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- + แทนการบวก เช่น $F = A + B$
- แทนการลบ เช่น $F = A - B$
- * แทนการคูณ เช่น $F = A * B$
- / แทนการหาร เช่น $F = A / B$
- \ แทนการหารแบบเอาเศษ $F = A \setminus B$
- > แทนการส่งสถานะปัจจุบันไปสถานะถัดไป เช่น $ABC \rightarrow DEF$
- ~ แทนการกำหนดช่วง เช่น $A[0 \sim 1]$ หมายถึง A_0, A_1

การกำหนดค่า

- In = ขาอินพุต เช่น In : (A,B) หรือ In : A[0~1]
- Io = ขาที่เป็นทั้งอินพุตและเอาต์พุต เช่น Io : (A,B)
- OUT = ขาเป็นเอาต์พุตเพียงอย่างเดียว เช่น OUT : (A,B)
- Clock = ขาสัญญาณนาฬิกา เช่น Clock : clk
- Enable = ขาที่ทำหน้าที่เป็นตัวเปิด ปิดการทำงานของเกต

คำสั่งที่ใช้ในการเขียนโปรแกรม PLD

Table เป็นตัวกำหนดค่าตารางที่ต้องการจะให้ได้อาต์พุตตามที่ต้องการ
ตัวอย่างการใช้งาน

```
| Table : D,C,B,A -> W,X,Y,Z | (D,C,B,A) = Preent |
| {                               | (W,X,Y,Z) = Next
|   0 -> 0011b
|   1 -> 0001b
|   2 -> 0011b
|   .
|   .
|   15 -> 0011b
| }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Map ใช้กำหนดค่าในรูปแบบของวงจรซีเควนเซียล.

ตัวอย่างในการใช้งาน

| Map : Q[3~0] -> Q[3~0]

| { n -> n+1 , n < 9

| n -> , n >= 9

| }

จากบรรทัดที่ 1 เป็นการกำหนดค่า Present State -> Next State

จากบรรทัดที่ 2 กำหนดให้ค่า n มีค่าเพิ่มขึ้นทีละ 1 ไปเรื่อยๆ ภายใต้งเงื่อนไขที่กำหนดคือน้อยกว่า 9

จากบรรทัดที่ 3 จะทำต่อจากบรรทัดที่ 2 เมื่อค่าที่เพิ่มขึ้นมากกว่าหรือเท่ากับ 9 ก็จะทำให้กลับเป็น 0 คือเริ่มนับใหม่

* จะเห็นได้ว่านั่นคือวงจรนับ 9 นั่นเอง คือ นับจาก 0 ถึง 8

Procedure เป็นคำสั่งกำหนดการทำงานแบบให้กระโดดไปหา Label หรือ Function ต่างๆ ที่ได้สร้างขึ้น

ตัวอย่างใช้งาน

| Procedure : Reset , Q[1,0]

| { Equal. Z1 = 1

| x1 & x2 ' ? -> Greater

| 'x1' & x2 ? -> Less

| -> Equal

| Greater. Z2 = 1

| -> Greater

| Less. Z3 = 1

| }

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากโปรแกรมที่ Lable Equal. กำหนดให้ $Z1 = 1$ หรือ เป็น “1” หลังจากนั้นเปรียบเทียบ ลอจิก ถ้าเป็นจริงให้กระโดดไปที่ Lable Greater ถ้าไม่ใช้มาทำบรรทัดถัดไป คือ Less ถ้าจริงไปทำที่ Less ถ้าไม่ตรงตามเงื่อนไขทั้งหมดก็วนไปทำโปรแกรมใหม่ คือ เริ่มให้ $Z1 = 1$ อีก ส่วน Lable อื่น เป็น การเปรียบเทียบและทำงานก็เหมือนกับ Lable Equal

* หลัง Lable ต้องมีเครื่องหมาย .

* ? หมายถึง Then

* -> หมายถึง Jump หรือ Goto

Vectors เป็นคำสั่งที่ใช้ในการทดสอบโปรแกรมที่เขียนขึ้นมาว่าถูกต้องหรือไม่

ตัวอย่างการใช้งาน

| Vectors :

| { Display Clock ,A,B

| Test A=0 ; B=1 ; Clock= 0,1

| Test A=1 ; B=1 ; Clock 10[0,1]

| END }

จากโปรแกรม

Display เป็นตัวบอกว่าจะให้แสดงค่าใดออกหน้าจอ

Test เป็นตัวบอกถึงสัญญาณกระตุ้นที่ต้องการทดสอบค่าใดบ้าง เป็นจำลองใน Orcad/VST

* Display clock มีหลายแบบ

(clock) d จะแสดงเป็นเลขฐาน 10

(clock) b จะแสดงเป็นเลขฐาน 2

(clock) c จะแสดงเป็นรูปสัญญาณนาฬิกา

(clock) o จะแสดงเป็นเลขฐาน 8

* คำสั่งที่ต้องการสั่งให้เป็น A=0 สามารถใช้คำสั่ง Clear A

* คำสั่งที่ต้องการสั่งให้เป็น A=1 สามารถใช้คำสั่ง Set A

ลำดับการเขียนโปรแกรม

- | Title : รายละเอียดอธิบายโปรแกรม
- | Active - High : อินพุตหรือเอาต์พุตใดต้องการให้ทำงานที่ลอจิก 1
- | Active - Low : อินพุตหรือเอาต์พุตใดต้องการให้ทำงานที่ลอจิก 0
- | Register : { กรณี อุปกรณ์ PLD มี Register ภายในตัว }
- | ส่วนของโปรแกรม
- | Vectors

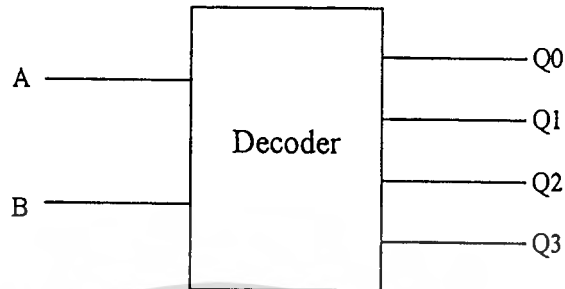
ตัวอย่างการเขียนโปรแกรม PLD

ตัวอย่างที่ 1 ออกแบบ PLD ให้มี AND GATE, NOT GATE , OR GATE , E-OR GATE , NAND GATE มีสมการดังนี้ $Q0 = A \& B$, $Q1 = A + D$, $Q2 = B$, $Q3 = C + B$, $Q4 = C \cdot A$

- | Title : “ This program have 5 gates in 1 PLD “
- | PAL16L8 : In : (A,B,C,D) ; Io : Q[4~0]
- | Active High : Q[4~3]
- | Active Low : Q[2~0]
- | $Q0 = A \& B$
- | $Q1 = A \# B$
- | $Q2 = B'$
- | $Q3 = C \# \# D$
- | $Q4 = C \& 'A'$
- | Vector : {
- | Display (Q[4~3]d, (Q[2])C , Q1 , (Q0)0 , (A)C , B , C , d
- | Set A
- | Set B
- | Clear C
- | Test D = 10(0,1)
- | END}

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวอย่าง 2 ออกแบบ PLD เป็น Decoder เข้า 2 ออก 4



| PAL 16L8 2:A, 3:B, 15~18 : Q[3~0]

| Active Low "ALL"

| Active High A,B

| $Q0 = A' \& B'$

| $Q1 = A \& B'$

| $Q2 = A' \& B$

| $Q3 = A \& B$

| Vector : {

| Display Q[3~0]C , A,B

| Test A=10(0,1)

| Test B = 1

| Test B = 10[1,0]

| END

| }

ตัวอย่างที่ 3 ออกแบบวงจรนับ 0-13

| Pal 16R8 In : (Reset , En}, OUT : Q[3~0] , Clock : Ck

| Map Q[3~0] -> Q[3~0] {

| $n \rightarrow n+1, n < 13 \& \text{Reset} \& \text{En}'$

| $n \rightarrow 0, n \geq 13 \& \text{Reset} \& \text{En}'$

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของสถาบันเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

| }
| Vectors :
|
| {Display Q[3~0]C , (clk)C , En
| Test Reset = 0 ; En = 0 , clk 0,1,0
| Set Reset
| Test En = 0 , clock 10[0,1]
| END
| }

```



ภาคผนวก ค.

โปรแกรมที่ใช้ร่วมกับเครื่องโปรแกรม PAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรม PALPGM.C

```
#include <conio.h>
#include <dir.h>
#include <dos.h>
#include <mem.h>
#include <stdio.h>
#include <string.h>
#include <alloc.h>
#include <bios.h>
#include "pal.h"

char menu_1 [4][13] = { " Load File ",
                        " Change Dir ",
                        " Dos Shell " };

char menu_2 [7][18] = { " Program      ",
                        " Display device " };

char menu_3 [1][14] = { " Edit buffer "};

char menu_4 [3][7] = { " AMD ",
                      " MMI ",
                      " NS  " };

char menu_5 [2][7] = { " COM1 ",
                      " COM2 " };

char menu_6 [1][7] = { " Quit " };

char menu_4_1 [4][7] = { " 16L8 ",
                        " 16R4 ",
                        " 16R6 ",
                        " 16R8 "};

char menu_4_2 [8][7] = { " 16R4 ",
                        " 16R6 ",
                        " 16L8 ",
                        " 16R8 ",
                        " 20R4 ",
                        " 20R6 ",
                        " 20L8 ",
                        " 20R8 "};
```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ 2006 การใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        " 20L8 ",
        " 20R8 " );

char str[200],str_f[20],ed_sw,*buffer,*curloc,*endloc,*curmark,*mark,*endcur;
char x,y,s_ok,PORT=0;
unsigned int row,col,sft;

```

```

/*-----*/

```

```

void printxy(x,y,ascii,attrib)

```

```

unsigned char ascii;

```

```

char attrib;

```

```

int x,y;

```

```

{

```

```

    int address;

```

```

    char far *v_ram,far *crt;

```

```

    crt = (char far *) 0x00400049;

```

```

    if (*crt == 3)

```

```

        v_ram = (char far *) 0xB8000000;

```

```

    else

```

```

        v_ram = (char far *) 0xB0000000;

```

```

    address = ((y-1)*160)+((x-1)*2);

```

```

    *(v_ram+address) = ascii;

```

```

    *(v_ram+address+1) = attrib;

```

```

}

```

```

void printstrxy(x,y,string,color)

```

```

int x,y;

```

```

unsigned char *string;

```

```

char color;

```

```

{

```

```

    int count=0;

```

```

    while (*(string+count) != '\0')

```

```

    {

```

```

        printxy(x+count,y,*(string+count),color);

```

```

        count++;

```

```

    }

```

```

}

```

```

void under (x3,y3,x4,y4,color)

```

```

int x3,y3,x4,y4;

```

```

char color;

```

```

{

```

```

int row,col,count;
for (row=y3;row<=y4;row++)
{
    for (col=x3;col<=x4;col++)
        printxy(col,row,' ',color);
}
}

void table (x1,y1,x2,y2,type,str,color)
int x1,y1,x2,y2;
char type,*str,color;
{
    int x,y,i;
    char *p;
    char option [4][8] = {
        { 218,196,191,179,217,196,192,179 },
        { 214,196,183,186,189,196,211,186 },
        { 213,205,184,179,190,205,212,179 },
        { 201,205,187,186,188,205,200,186 }
    };
    type--;
    for (y=y1+1;y<y2;y++)
    {
        for (x=x1+1;x<x2;x++)
            printxy(x,y,' ',color);
    }
    for (x=x1;x<=x2;x++)
    {
        printxy(x,y1,option[type][1],color);
        printxy(x,y2,option[type][5],color);
    }
    for (y=y1;y<=y2;y++)
    {
        printxy(x1,y,option[type][3],color);
        printxy(x2,y,option[type][7],color);
    }
    printxy(x1,y1,option[type][0],color);
    printxy(x2,y1,option[type][2],color);
    printxy(x2,y2,option[type][4],color);
    printxy(x1,y2,option[type][6],color);
}

```

```

if (strlen(str)) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

x = (x2-x1-5);
p = str;
while (*p != 0) {
    if (*p == 47) *p = 92; p++;    /* 47 = '/' 92 = '\' */
}
if (y > x) {
    y = strlen (str) / 2;
    x = (x2-x1+1) / 2;
    i = (y-x-1)*2;
    i += 12;
    printstrxy (x1+2,y1," ...",color);
    printstrxy (x1+6,y1,(str+i-1),color);
    i = strlen(str) - i;
    printxy (x1+i+7,y1,' ',color);
} else {
    x = (x2-x1+1)/2;
    y = strlen (str) / 2;
    i = x1+x-y;
    printxy (i-1,y1,' ',color);
    printstrxy (i,y1,str,color);
    i = strlen (str) + i;
    printxy (i,y1,' ',color);
}
p = str;
while (*p != 0) {
    if (*p == 92) *p = 47; p++;
}
}
}

```

```

void title(void)

```

```

{
    char x,y;
    for (x=1;x<=80;x++)
    {
        printxy(x,9,' ',0x70);
        printxy(x,8,' ',0);
    }
    table(1,1,80,7,4,"",LIGHTCYAN);

```

```

    printstrxy(4,2," Work path  :",0x0E);

```

```

    printstrxy(4,3," Work File  :",0x0E);

```

```

printstrxy(4,4," Device      :",0x0E);
printstrxy(4,5," Type       :",0x0E);
printstrxy(4,6," Serial port  :",0x0E);
printstrxy(18,6," COM1",0x0A);
curdirectory();
printstrxy(2,9," File ",NOM_COLOR1);
printstrxy(11,9," RunFunc ",NOM_COLOR1);
printstrxy(23,9," Edit/View ",NOM_COLOR1);
printstrxy(37,9," Device ",NOM_COLOR1);
printstrxy(48,9," Send ",NOM_COLOR1);
printstrxy(58,9," Quit ",NOM_COLOR1);
printstrxy(1,20, " ",0x1F);
printstrxy(1,21," ",REV_COLOR);
printstrxy(1,22," PAL Programmer Version 1.00",REV_COLOR);
printstrxy(1,23," KMITL ED. ENGINEER",REV_COLOR);
printstrxy(1,24," ",REV_COLOR);
printstrxy(1,25, " ",0x18);
}

void cur_on(void)
{
    union REGS reg;
    reg.x.ax = 0x0100;
    reg.x.cx = 0x0B0C;
    int86(0x10,&reg,&reg);
    return;
}

void cur_off(void)
{
    union REGS reg;
    reg.x.ax = 0x0100;
    reg.x.cx = 0x2000;
    int86(0x10,&reg,&reg);
    return;
}

void disp_menu1(b)
char b;
{
    char c;
    printstrxy(2,9," File ",MENU_BAR_COLOR);

```

```

char b;
{
    char c;
    printstrxy(2,9," File ",MENU_BAR_COLOR);
    for (c=0;c<3;c++)
        printstrxy(3,11+c,menu_1[c],NOM_COLOR);
    printstrxy(3,11+b,menu_1[b],MENU_BAR_COLOR);
}

void disp_menu2(b)
char b;
{
    char c;
    printstrxy(11,9," RunFunc ",MENU_BAR_COLOR);
    for (c=0;c<2;c++)
        printstrxy(12,11+c,menu_2[c],NOM_COLOR);
    printstrxy(12,11+b,menu_2[b],MENU_BAR_COLOR);
}

void disp_menu3(b)
char b;
{
    printstrxy(23,9," Edit/View ",MENU_BAR_COLOR);
    printstrxy(24,11+b,menu_3[b],MENU_BAR_COLOR);
}

void disp_menu4(b)
char b;
{
    char c;
    printstrxy(37,9," Device ",MENU_BAR_COLOR);
    for (c=0;c<3;c++)
        printstrxy(38,11+c,menu_4[c],NOM_COLOR);
    printstrxy(38,11+b,menu_4[b],MENU_BAR_COLOR);
}

void disp_menu4_1(b)
char b;
{
    char c;

```

```

for (c=0;c<4;c++)
    printstrxy(43,12+c,menu_4_1[c],NOM_COLOR);
printstrxy(43,12+b,menu_4_1[b],MENU_BAR_COLOR);
}

```

```

void disp_menu4_2(b)
char b;
{
    char c;
    for (c=0;c<8;c++)
        printstrxy(43,13+c,menu_4_2[c],NOM_COLOR);
    printstrxy(43,13+b,menu_4_2[b],MENU_BAR_COLOR);
}

```

```

void disp_menu5(b)
char b;
{
    char c;
    printstrxy(48,9," Send ",MENU_BAR_COLOR);
    for (c=0;c<2;c++)
        printstrxy(49,11+c,menu_5[c],NOM_COLOR);
    printstrxy(49,11+b,menu_5[b],MENU_BAR_COLOR);
}

```

```

void disp_menu6(b)
char b;
{
    printstrxy(58,9," Quit ",MENU_BAR_COLOR);
    table(58,10,65,12,2,"",NOM_COLOR);
    printstrxy(59,11+b,menu_6[b],MENU_BAR_COLOR);
}

```

```

void menu_file(key,v_pos)
char key[],*v_pos;
{
    char *buf_file,*buf_1;
    char done;
    buf_file = malloc(15*6*2);

```

```

    *v_pos = 0;

```

```

    gettex(2,10,16,15,buf_file);

```

```

under(3,11,16,15,0x00);
table(2,10,15,14,2,"",NOM_COLOR);
disp_menu1(*v_pos);
do {
    do {
        key[0] = getch();
        if (key[0] == 0)
        {
            key[1] = getch();
            switch (key[1])
            {
                case U_ARRO :
                    if (*v_pos > 0)
                        (*v_pos)--;
                    else
                        (*v_pos)= 2;
                    disp_menu1(*v_pos);
                    break;
                case D_ARRO :
                    if (*v_pos < 2)
                        (*v_pos)++;
                    else
                        (*v_pos)=0;
                    disp_menu1(*v_pos);
                    break;
            }
        } else key[1]=0;
    } while (key[0]!=13 && key[0] !=27 && key[1]!=75 && key[1]!=77);
    if (key[0]==ENTER)
    {
        switch(*v_pos)
        {
            case 0:
                if (readstr (12,11,30,str,200,"Load File Name"))
                {
                    if (check_str(str)==0)
                    {
                        if (load_file(str))
                        {
                            load(str);
                        }
                    }
                }
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการ load(str); เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

load_micro(str);
str_file(str_f,str);
x=STARTX;
y=STARTY;
sft = 0;
col = 1;
row = 1;
printstrxy (19,3,"",0x0A);
printstrxy (19,3,str_f,0x0A);
}
} else drive_error();
} else key[0]=77;
break;
case 1:
/*change dir*/
if (readstr (12,13,20,str,200,"New Directory"))
{
done=ch_dir(str);
if (done==0 && check_drive(str[0]) == 0)
{
printstrxy (19,2,"",0x0A);
printstrxy (19,2,str,0x0A);
}
else
{
path();
}
} else key[0]=77;
break;
case 2:
/*dos shell*/
os_shell();
break;
}
}
} while (key[0]!=27 && key[1]!=75 && key[1]!=77);
puttext(2,10,16,15,buf_file);
/* puttext(3,11,16,16,buf_1);*/
/*printstrxy(2,9," File ",REV_COLOR);*/
free(buf_file);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* free(buf_1);*/
}

void menu_runfunc(key,v_pos)
char key[],*v_pos;
{
    unsigned char *buf_runfunc,*buf_2;
    buf_runfunc = malloc(20*10*2);
    /*buf_2 = malloc(19*5*2);*/
    *v_pos = 0;
    /*gettext(12,11,30,19,buf_2);*/
    gettext(11,10,30,14,buf_runfunc);
    under(12,11,30,14,0x00);
    table(11,10,29,13,2,"",NOM_COLOR);
    disp_menu2(*v_pos);
    do {
        do
        {
            key[0] = getch();
            if (key[0] == 0)
            {
                key[1] = getch();
                switch (key[1])
                {
                    case U_ARRO :
                        if (*v_pos > 0)
                            (*v_pos)--;
                        else
                            (*v_pos)= 1;
                        disp_menu2(*v_pos);
                        break;
                    case D_ARRO :
                        if (*v_pos < 1)
                            (*v_pos)++;
                        else
                            (*v_pos)=0;
                        disp_menu2(*v_pos);
                        break;
                }
            }
        } while (key[0] != 0);
    } while (key[0] != 0);
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    } else key[1]=0;
} while (key[0]!=13 && key[0] !=27 && key[1]!=75 && key[1]!=77);

if (key[0]==ENTER)
{
    switch(*v_pos)
    {
        case 0:
            reayl_process(v_pos);
            /* key[3]=0;*/
            break;
        case 1:
            /*display device*/
            /*key[3]=0;*/
            dump0;
            break;
    }
}

while (key[0]!=27 && key[1]!=75 && key[1]!=77 && key[3]!=0);
puttext(11,10,30,14,buf_runfunc);
/*puttext(12,11,30,19,buf_2);*/
/*printstrxy(11,9," RunFunc ",REV_COLOR);*/
free(buf_runfunc);

/*free(buf_2);*/
}

void menu_edit_view(key,v_pos)
char key[],*v_pos;
{
    char *buf_edit,*buf_3,*save_show;
    buf_edit = malloc(16*4*2);
    /*buf_3 = malloc(25*3*2);*/

    *v_pos = 0;
    /*gettext(24,11,38,13,buf_3);*/
    gettext(23,10,38,13,buf_edit);
    under(24,11,38,13,0x00);
    table(23,10,37,12,2,"",NOM_COLOR);
    disp_menu3(*v_pos);
}

```

```

do
{
    key[0] = getch();
    if (key[0] == 0)
    {
        key[1] = getch();
        switch (key[1])
        {
            case U_ARRO :
                (*v_pos) = 0;
                disp_menu3(*v_pos);
                break;
            case D_ARRO :
                (*v_pos) = 0;
                disp_menu3(*v_pos);
                break;
        }
    } else key[1]=0;
} while (key[0]!=13 && key[0] !=27 && key[1]!=75 && key[1]!=77);
if (key[0]==ENTER)
{
    switch(*v_pos)
    {
        case 0:
            save_show = malloc(80*25*2);
            gettext(1,1,80,25,save_show);
            textcolor(WHITE);
            textbackground(BLUE);
            clrscr();
            /*list();*/
            edit_buf();
            puttext(1,1,80,25,save_show);
            free(save_show);
            break;
    }
}

} while (key[0]!=27 && key[1]!=75 && key[1]!=77);
puttext(23,10,38,13,buf_edit);
/*puttext(24,11,38,13,buf_3);*/
/*printstrxy(23,9," Edit/View " ,REV_COLOR);*/

```

```
free(buf_edit);
```

```
/*free(buf_3);*/
```

```
}
```

```
void menu_amd(key,v1_pos)
```

```
char key[],*v1_pos;
```

```
{
```

```
char *buf_amd,*buf_4;
```

```
buf_amd = malloc(9*7*2);
```

```
/*buf_4 = malloc(8*6*2);*/
```

```
*v1_pos = 0;
```

```
/*gettext(43,12,50,17,buf_4);*/
```

```
gettext(42,11,50,17,buf_amd);
```

```
under(43,12,50,17,0x00);
```

```
table(42,11,49,16,2,"",NOM_COLOR);
```

```
disp_menu4_1(*v1_pos);
```

```
do{
```

```
do
```

```
{
```

```
key[0] = getch();
```

```
if (key[0] == 0)
```

```
{
```

```
key[1] = getch();
```

```
switch (key[1])
```

```
{
```

```
case U_ARRO :
```

```
if (*v1_pos > 0)
```

```
(*v1_pos)--;
```

```
else
```

```
(*v1_pos)= 3;
```

```
disp_menu4_1(*v1_pos);
```

```
break;
```

```
case D_ARRO :
```

```
if (*v1_pos < 3)
```

```
(*v1_pos)++;
```

```
else
```

```
(*v1_pos)=0;
```

```
disp_menu4_1(*v1_pos);
```

```
break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    } else key[1]=0;
} while (key[0]!=13 && key[0]!=27);/* && key[1]!=75 && key[1]!=77);*/
if (key[0]==ENTER)
{
    switch(*v1_pos)
    {
        case 0:
            key[3]=0;
            break;
        case 1:
            key[3]=0;
            break;
        case 2:
            key[3]=0;
            break;
        case 3:
            key[3]=0;
            break;
    }
}

} while (key[0]!=27 && key[0]!=13 && key[1]!=75 && key[1]!=77 && key[3]!=0);
if (key[3]==0)
{
    printstrxy(18,5,menu_4_1[*v1_pos],0x0A);
    switch(*v1_pos)
    {
        case 0:
            sport(PORT,0x82);
            break;
        case 1:
            sport(PORT,0x80);
            break;
        case 2:
            sport(PORT,0x81);
            break;
        case 3:
            sport(PORT,0x83);
            break;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }

    }

    puttext(42,11,50,17,buf_amd);
    /*puttext(43,12,50,17,buf_4);*/
    free(buf_amd);

/*  free(buf_4);*/
}

void menu_nmi(key,vl_pos)
char key[],*vl_pos;
{
    char *buf_nmi,*buf_5;
    buf_nmi = malloc(9*11*2);
    /*buf_5 = malloc(8*10*2);*/
    *vl_pos = 0;
    /*gettext(43,13,50,22,buf_5);*/
    gettext(42,12,50,22,buf_nmi);
    under(43,13,50,22,0x00);
    table(42,12,49,21,2,"",NOM_COLOR);
    disp_menu4_2(*vl_pos);
    do{
    do
    {
        key[0] = getch();
        if (key[0] == 0)
        {
            key[1] = getch();
            switch (key[1])
            {
                case U_ARRO :
                    if (*vl_pos > 0)
                        (*vl_pos)--;
                    else
                        (*vl_pos)= 7;
                    disp_menu4_2(*vl_pos);
                    break;
                case D_ARRO :
                    if (*vl_pos < 7)
                        (*vl_pos)++;
            }
        }
    }
    while (key[0] != 0);
}

```

```

else
    (*v1_pos)=0;
    disp_menu4_2(*v1_pos);
    break;
}
} else key[1]=0;
} while (key[0]!=13 && key[0]!=27);/* && key[1]!=75 && key[1]!=77;*/
if (key[0]==ENTER)
{
    switch(*v1_pos)
    {
        case 0:
            key[3]=0;
            break;
        case 1:
            key[3]=0;
            break;
        case 2:
            key[3]=0;
            break;
        case 3:
            key[3]=0;
            break;
        case 4:
            key[3]=0;
            break;
        case 5:
            key[3]=0;
            break;
        case 6:
            key[3]=0;
            break;
        case 7:
            key[3]=0;
            break;
    }
}
}

```

```

} while (key[0]!=27 && key[0]!=13 && key[1]!=75 && key[1]!=77 && key[3]!=0);

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    printstrxy(18,5,menu_4_2[*v1_pos],0x0A);
    switch(*v1_pos)
    {
        case 0:
            sport(PORT,0x80);
            break;
        case 1:
            sport(PORT,0x81);
            break;
        case 2:
            sport(PORT,0x82);
            break;
        case 3:
            sport(PORT,0x83);
            break;
        case 4:
            sport(PORT,0x84);
            break;
        case 5:
            sport(PORT,0x85);
            break;
        case 6:
            sport(PORT,0x86);
            break;
        case 7:
            sport(PORT,0x87);
            break;
    }
}

puttext(42,12,50,22,buf_nmi);
/*puttext(43,13,50,22,buf_5);*/
free(buf_nmi);

/*free(buf_5);*/
}

```

```

void menu_ns(key,v1_pos)

```

```

char key[],*v1_pos;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

char *buf_ns,*buf_6;
buf_ns = malloc(9*11*2);
/*buf_6 = malloc(8*10*2);*/
*v1_pos = 0;
/*gettext(43,13,50,22,buf_6);*/
gettext(42,12,50,22,buf_ns);
under(43,13,50,22,0x00);
table(42,12,49,21,2,"",NOM_COLOR);
disp_menu4_2(*v1_pos);
do{
do
{
key[0] = getch();
if (key[0] == 0)
{
key[1] = getch();
switch (key[1])
{
case U_ARRO :
if (*v1_pos > 0)
(*v1_pos)--;
else
(*v1_pos)= 7;
disp_menu4_2(*v1_pos);
break;
case D_ARRO :
if (*v1_pos < 7)
(*v1_pos)++;
else
(*v1_pos)=0;
disp_menu4_2(*v1_pos);
break;
}
} else key[1]=0;
} while (key[0]!=13 && key[0]!=27);/* && key[1]!=75 && key[1]!=77);*/
if (key[0]==ENTER)
{
switch(*v1_pos)
{
case 0:

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        key[3]=0;
    break;
    case 1:
        key[3]=0;
    break;
    case 2:
        key[3]=0;
    break;
    case 3:
        key[3]=0;
    break;
    case 4:
        key[3]=0;
    break;
    case 5:
        key[3]=0;
    break;
    case 6:
        key[3]=0;
    break;
    case 7:
        key[3]=0;
    break;
}

} while (key[0]!=27 && key[0]!=13 && key[1]!=75 && key[1]!=77 && key[3]!=0);
if (key[3]==0)
{
    printstrxy(18,5,menu_4_2[*v1_pos],0x0A);
    switch(*v1_pos)
    {
        case 0:
            sport(PORT,0x80);
            break;
        case 1:
            sport(PORT,0x81);
            break;
        case 2:
            sport(PORT,0x82);

```

```

        break;
    case 3:
        sport(PORT,0x83);
        break;
    case 4:
        sport(PORT,0x84);
        break;
    case 5:
        sport(PORT,0x85);
        break;
    case 6:
        sport(PORT,0x86);
        break;
    case 7:
        sport(PORT,0x87);
        break;
    }
}

puttext(42,12,50,22,buf_ns);
free(buf_ns);
/* puttext(43,13,50,22,buf_6);*/
/*free(buf_6);*/
}

void menu_device(key,v_pos)
char key[],*v_pos;
{
    char v1_pos;
    char *buf_device,*buf_7;
    buf_device = malloc(9*6*2);
    /*buf_7 = malloc(8*5*2);*/
    *v_pos = 0;
    /*getttext(38,11,45,15,buf_7);*/
    getttext(37,10,45,15,buf_device);
    under(38,11,45,15,0x00);
    table(37,10,44,14,2,"",NOM_COLOR);
    disp_menu4(*v_pos);
    do

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    key[0] = getch();
    if (key[0] == 0)
    {
        key[1] = getch();
        switch (key[1])
        {
            case U_ARRO :
                if (*v_pos > 0)
                    (*v_pos)--;
                else
                    (*v_pos)= 2;
                disp_menu4(*v_pos);
                break;
            case D_ARRO :
                if (*v_pos < 2)
                    (*v_pos)++;
                else
                    (*v_pos)=0;
                disp_menu4(*v_pos);
                break;
        }
    } else key[1]=0;
} while (key[0]!=13 && key[0] !=27 && key[1]!=75 && key[1]!=77);
if (key[0]==ENTER)
{
    switch(*v_pos)
    {
        case 0:
            menu_amd(key,&v1_pos);
            key[0]=0;
            break;
        case 1:
            menu_nmi(key,&v1_pos);
            key[0]=0;
            break;
        case 2:
            menu_ns(key,&v1_pos);
            key[0]=0;
            break;
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
}

} while (key[0] !=27 && key[1]!=75 && key[1]!=77 && key[3]!=0);
if (key[3]==0)
{
    printstrxy(18,4,menu_4[*v_pos],0x0A);
    switch(*v_pos)
    {
        case 0:
            sport(PORT,0xA0);
            break;
        case 1:
            sport(PORT,0xA1);
            break;
        case 2:
            sport(PORT,0xA2);
            break;
    }
}

puttext(37,10,45,15,buf_device);
/*puttext(38,11,45,15,buf_7);*/
/*printstrxy(37,9," Device ",REV_COLOR);*/
free(buf_device);

/*free(buf_7);*/
}

void menu_send(key,v_pos)
char key[],*v_pos;
{
    char *buf_send,*buf_8;
    buf_send = malloc(9*5*2);
    /*buf_8 = malloc(8*4*2);*/
    *v_pos = 0;
    /*gettext(49,11,56,14,buf_8);*/
    gettext(48,10,56,14,buf_send);
    under(49,11,56,14,0x00);
    table(48,10,55,13,2,"",NOM_COLOR);
    disp_menu5(*v_pos);
    do {

```

```

do
{
    key[0] = getch();
    if (key[0] == 0)
    {
        key[1] = getch();
        switch (key[1])
        {
            case U_ARRO :
                if (*v_pos > 0)
                    (*v_pos)--;
                else
                    (*v_pos)= 1;
                disp_menu5(*v_pos);
                break;
            case D_ARRO :
                if (*v_pos < 1)
                    (*v_pos)++;
                else
                    (*v_pos)=0;
                disp_menu5(*v_pos);
                break;
        }
    } else key[1]=0;
} while (key[0]!=13 && key[0] !=27 && key[1]!=75 && key[1]!=77);
if (key[0]==ENTER)
{
    switch(*v_pos)
    {
        case 0:
            key[3]=0;
            PORT=0;
            break;
        case 1:
            key[3]=0;
            PORT=1;
            break;
    }
}
}

```

```

if (key[3]==0)
{
    port_init(PORT,0xE3);
    printstrxy(18,6,menu_5[*v_pos],0x0A);
}

puttext(48,10,56,14,buf_send);
/*puttext(49,11,56,14,buf_8);*/
/*printstrxy(47,9," Send ",REV_COLOR);*/
free(buf_send);

/*free(buf_8);*/
}

```

```

void menu_quit(key,v_pos)
char key[],*v_pos;
{
    char *buf_quit,*p,*buf_9,*p1;
    char ans;
    p= malloc(32*5*2);
    /*p1=malloc(31*4*2);*/
    buf_quit = malloc(9*4*2);
    /*buf_9 = malloc (8*3*2);*/
    *v_pos = 0;
    /*gettext(59,11,66,13,buf_9);*/
    gettext(58,10,66,13,buf_quit);
    under(59,11,66,13,0x00);
    disp_menu6(*v_pos);
    do {
        do
        {
            key[0] = getch();
            if (key[0] == 0)
            {
                key[1] = getch();
                switch (key[1])
                {
                    case U_ARRO :
                        (*v_pos) = 0;
                        disp_menu6(*v_pos);

```

```
break;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case D_ARRO :
    (*v_pos) = 0;
    disp_menu6(*v_pos);
    break;
}
} else key[1]=0;
} while (key[0]!=13 && key[0] !=27 && key[1]!=75 && key[1]!=77);
if (key[0]==ENTER)
{
    switch(*v_pos)
    {
        case 0:
            /*gettext(26,16,56,19,p1);*/
            gettext(25,15,56,19,p);
            under(26,16,56,19,0x00);
            table(25,15,55,18,4,"",0x4F);
            printstrxy(26,16," Are you sure you want exit? ",0x4F);
            printstrxy(26,17,"      (Yes/NO)      ",0x4F);
            do {
                ans=getch();
            } while (toupper(ans)!='Y' && toupper(ans)!='N');
            if (toupper(ans)=='Y')
            {
                key[4]=0;
                sport(PORT,0xFA); /* retrun loop micro*/
            }
            puttext(25,15,56,19,p);
            free(p);
            /*puttext(26,16,56,19,p1);*/
            /*free(p1);*/
            break;
        }
    }
} while (key[0] !=27 && key[1]!=75 && key[1]!=77 && key[4]!=0);
puttext(58,10,66,13,buf_quit);
/*puttext(59,11,66,13,buf_9);*/
/*printstrxy(58,9," Quit ",NOM_COLOR);*/
free(buf_quit);

```

```

}
```

```

void menu(void)
```

```

{
```

```

    char key[5],v_pos,m,*p;
```

```

    m = 0;
```

```

    title();
```

```

    do
```

```

    {
```

```

        do
```

```

        {
```

```

            switch(m)
```

```

            {
```

```

                case 0:
```

```

                    printstrxy(11,9," RunFunc ",NOM_COLOR1);
```

```

                    printstrxy(58,9," Quit ",NOM_COLOR1);
```

```

                    printstrxy(2,9," File ",MENU_BAR_COLOR);
```

```

                    break;
```

```

                case 1:
```

```

                    printstrxy(2,9," File ",NOM_COLOR1);
```

```

                    printstrxy(23,9," Edit/View ",NOM_COLOR1);
```

```

                    printstrxy(11,9," RunFunc ",MENU_BAR_COLOR);
```

```

                    break;
```

```

                case 2:
```

```

                    printstrxy(11,9," RunFunc ",NOM_COLOR1);
```

```

                    printstrxy(37,9," Device ",NOM_COLOR1);
```

```

                    printstrxy(23,9," Edit/View ",MENU_BAR_COLOR);
```

```

                    break;
```

```

                case 3:
```

```

                    printstrxy(23,9," Edit/View ",NOM_COLOR1);
```

```

                    printstrxy(48,9," Send ",NOM_COLOR1);
```

```

                    printstrxy(37,9," Device ",MENU_BAR_COLOR);
```

```

                    break;
```

```

                case 4:
```

```

                    printstrxy(37,9," Device ",NOM_COLOR1);
```

```

                    printstrxy(58,9," Quit ",NOM_COLOR1);
```

```

                    printstrxy(48,9," Send ",MENU_BAR_COLOR);
```

```

                    break;
```

```

                case 5:
```

```

                    printstrxy(11,9," RunFunc ",NOM_COLOR1);
```

```

                    printstrxy(58,9," Quit ",NOM_COLOR1);
```

```

                    printstrxy(2,9," File ",MENU_BAR_COLOR);
```

```

                    break;
```

```

            }
```

```

        }
```

```

    }
```

```

        printstrxy(48,9," Send ",NOM_COLOR1);
        printstrxy(2,9," File ",NOM_COLOR1);
        printstrxy(58,9," Quit ",MENU_BAR_COLOR);
        break;
    }
    key[0]=getch();
    if (key[0]==0)
        key[1]=getch();
    else
        key[1]=0;
    switch(key[1])
    {
        case R_ARRO:
            if (m<5)
                m++;
            else
                m=0;
            break;
        case L_ARRO:
            if (m>0)
                m--;
            else
                m=5;
            break;
    }

    } while (key[0]!=ENTER);

do
{
    do
    {
        switch(m)
        {
            case 0:
                menu_file(key,&v_pos);
                printstrxy(2,9," File ",MENU_BAR_COLOR);

                switch(key[1])

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        m = 1;
        printstrxy(2,9," File ",NOM_COLOR1);
        break;
    case L_ARRO:
        m = 5;
        printstrxy(2,9," File ",NOM_COLOR1);
        break;
    }
    break;

case 1:
    menu_runfunc(key,&v_pos);
    printstrxy(11,9," RunFunc ",MENU_BAR_COLOR);
    switch(key[1])
    {
        case R_ARRO:
            m = 2;
            printstrxy(11,9," RunFunc ",NOM_COLOR1);
            break;
        case L_ARRO:
            m = 0;
            printstrxy(11,9," RunFunc ",NOM_COLOR1);
            break;
    }
    break;

case 2:
    menu_edit_view(key,&v_pos);
    printstrxy(23,9," Edit/View ",MENU_BAR_COLOR);
    switch(key[1])
    {
        case R_ARRO:
            m = 3;
            printstrxy(23,9," Edit/View ",NOM_COLOR1);
            break;
        case L_ARRO:
            m = 1;
            printstrxy(23,9," Edit/View ",NOM_COLOR1);
            break;
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

break;

case 3:
    menu_device(key,&v_pos);
    printstrxy(37,9," Device ",MENU_BAR_COLOR);
    switch(key[1])
    {
        case R_ARRO:
            m = 4;
            printstrxy(37,9," Device ",NOM_COLOR1);
            break;
        case L_ARRO:
            m = 2;
            printstrxy(37,9," Device ",NOM_COLOR1);
            break;
    }
    break;

case 4:
    menu_send(key,&v_pos);
    printstrxy(48,9," Send ",MENU_BAR_COLOR);
    switch(key[1])
    {
        case R_ARRO:
            m = 5;
            printstrxy(48,9," Send ",NOM_COLOR1);
            break;
        case L_ARRO:
            m = 3;
            printstrxy(48,9," Send ",NOM_COLOR1);
            break;
    }
    break;

case 5:
    menu_quit(key,&v_pos);
    printstrxy(58,9," Quit ",MENU_BAR_COLOR);
    switch(key[1])
    {

```

```

        m = 0;
        printstrxy(58,9," Quit ",NOM_COLOR1);
        break;
    case L_ARRO:
        m = 4;
        printstrxy(58,9," Quit ",NOM_COLOR1);
        break;
    }
    break;
}

} while(key[0]!=27 && key[0]!=ENTER && key[3]!=0 && key[4]!=0);

} while(key[0]!=ESC && key[3]!=0 && key[4]!=0);
key[3]=1;
} while (key[4]!=0);
}
/*-----End Menu-----*/

void load_f (void)
{
    if (readstr (10,5,30,str,200,"Load File Name")) {
        check_str (str);
        if (load_file (str)) {
            load (str);
            str_file (str_f,str);
            x = STARTX;
            y = STARTY;-
            sft = 0;
            col = 1;
            row = 1;
            ed_sw = OFF;
            s_ok = ON;
            str_file (str_f,str);
            printstrxy (STARTX+60,STARTY-1,"          ",NOM_COLOR);
            printstrxy (STARTX+60,STARTY-1,str_f,NOM_COLOR);
        }
    }
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void s_file (void)
{
    if (!s_ok) {
        if (readstr (10,5,30,str,200,"Rename File")) {
            if (w_file (1)) s_ok = ON;
        }
    } else save (str);
}

w_file (pass_ok)
char pass_ok;
{
    char ch,*p,k[20],ok;
    ok = ON;
    if (pass_ok || readstr (10,5,30,str,200,"New Name")) {
        check_str (str);
        str_file (k,str);
        switch (check_file(str)) {
            case 0 :
                save (str);
                str_file (str_f,str);
                break;
            case 1 :
                p = malloc (((29+strlen(k))-10+1)*(12-10+1)*2);
                gettext (10,10,(29+strlen(k)),12,p);
                table (10,10,29+strlen(k),12,1,"Verify",WRITE_T_COLOR);
                gotoxy (11,11); printf ("%s %s ? <Y/N>","Overwrite",k);
                do {
                    ch = getch ();
                } while (ch != 'Y' && ch != 'y' && ch != 'N' && ch != 'n');
                if (ch == 'Y' || ch == 'y') {
                    save (str);
                    str_file (str_f,str);
                } else ok = OFF;
                puttext (10,10,(29+strlen(k)),12,p);
                free (p);
                break;
            case 2 :
                if (load_file (str)) {
                    save (str);

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        str_file (str_f,str);

    }

    break;

}

printstrxy (STARTX+60,STARTY-1,"          ",NOM_COLOR);
printstrxy (STARTX+60,STARTY-1,str_f,NOM_COLOR);
}

return (ok);
}

```

```

check_file (str)

```

```

char *str;

```

```

{
    struct fblk blk;
    char done,ok;
    done = findfirst (str,&blk,0xf7);
    if (!done) {
        if (!findnext (&blk)) ok = 2;      /* MORE THAN ONCE */
        else ok = 1;                      /* ONLY */
    } else ok = 0;                        /* NONE */
    return (ok);
}

```

```

/*-----Line Editor-----*/

```

```

find_end (x,end,curloc,endloc)

```

```

char x,end,*curloc,*endloc;

```

```

{
    while (x <= end) {
        if (curloc != endloc) {
            x++;
            curloc++;
        } else {
            x = end + 1;
            end = 0;
        }
    }
}

return (end);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void re_line (x,y,end,str)
char x,y,end,*str;
{
    while (x < end) {
        if (*str != 0) {
            printxy (x,y,*str,LINE_C_COLOR);
            str++;
        } else printxy (x,y,' ',LINE_C_COLOR);
        x++;
    }
}

readstr (x1,y1,len_t,str,len_s,head_str) /* len_t = lenght of table */
char x1,y1,len_t,*str; /* len_s = lenght of string */
int len_s; /* x1,y1 = top-left point */
char *head_str;
{
    char key[2],col,x2,y2,*buf_N;
    char *curloc,*endloc,*p;
    buf_N = malloc ((len_t+1) * 3);
    gettext (x1,y1,x1+len_t,y1+2,buf_N);
    curloc = endloc = str;
    *endloc = 0;
    col = x1 + 2;
    x2 = x1 + len_t;
    y2 = y1 + 2;
    table (x1,y1,x2,y2,1,head_str,LINE_T_COLOR);
    cur_on 0;
    gotoxy (col,y1+1);
    do {
        key[0] = getch ();
        if (key[0] == 0) {
            key[1] = getch ();
            switch (key[1]) {
                case 75 : /* Lt */
                    if (curloc != str) {
                        col--;
                        curloc--;
                    }
                    if (col < x1+2) {
                        col++;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        re_line (col,y1+1,x2-1,curloc);
        printxy (x2-1,y1+1,16,LINE_C_COLOR);
    }
    gotoxy (col,y1+1);
    if (!find_end (col,x2-2,curloc,endloc))
        printxy (x2-1,y1+1,'',LINE_C_COLOR);
}
break;
case 77 : /* Rt */
    if (curloc != endloc) {
        col++;
        curloc++;
        if (col > x2-2) {
            col--;
            re_line (x1+2,y1+1,x2-1,curloc-len_t+4);
            printxy (x1+1,y1+1,17,LINE_C_COLOR);
        }
        gotoxy (col,y1+1);
    }
    break;
case 83 : /* Del */
    if (curloc != endloc) {
        memmove (curloc,curloc+1,endloc-curloc);
        re_line (col,y1+1,x2-1,curloc);
        endloc--;
        if (!find_end (col,x2-2,curloc,endloc))
            printxy (x2-1,y1+1,'',LINE_C_COLOR);
    }
    break;
}
} else {
    if (key[0] != 8) { /* Not Backspace */
        if (key[0] != 13 && key[0] != 27 && endloc < str+len_s) {
            memmove (curloc+1,curloc,endloc-curloc+1);
            *curloc = key[0];
            if (col < x2-2) {
                re_line (col,y1+1,x2-1,curloc);
                col++;
            } else {
                re_line (x1+2,y1+1,x2-2,curloc-len_t+5);
            }
        }
    }
}

```

```

        printxy (x1+1,y1+1,17,LINE_C_COLOR);
    }

    curloc++;
    endloc++;
    gotoxy (col,y1+1);
    if (find_end (col,x2-2,curloc,endloc))
        printxy (x2-1,y1+1,16,LINE_C_COLOR);
}

} else {
    if (curloc != str) {
        memmove (curloc-1,curloc,endloc-curloc+1);
        col--;
        endloc--;
        curloc--;
        if (col < x1+2) {
            col++;
            printxy (x2-1,y1+1,16,LINE_C_COLOR);
        }
        re_line (col,y1+1,x2-1,curloc);
        gotoxy (col,y1+1);
        if (!find_end (col,x2-2,curloc,endloc))
            printxy (x2-1,y1+1,' ',LINE_C_COLOR);
    }
}

}

if (curloc < str+1) printxy (x1+1,y1+1,' ',LINE_C_COLOR);
if (curloc > endloc-1) printxy (x2-1,y1+1,' ',LINE_C_COLOR);
} while (key[0] != 13 && key[0] != 27); /* Not Enter and Esc */
puttext (x1,y1,x1+len_t,y1+2,buf_N);
cur_off 0;
strupr (str);
p = str;
while (*p == 32) p++; /* 32 = space bar */
memmove (str,p,strlen(p)+1);
free (buf_N);
if (key[0] == 27) key[0] = 0; else key[0] = 1;
return (key[0]);
}

```

```

int check_str (str)
char *str;
{
    char n_drive,*cur_dir,*buff,*p;
    if (check_drive(str[0]) == 0)
        return(0);
    if (check_drive(str[0]) == 0x80)
        return(1);
    cur_dir = malloc (200);
    buff = malloc (200);
    if (*(str+1) == ':' && *(str+2) == 92 && !str'en(str+3)) strcpy ((str+3),NOM_FILE);
    if (!strlen (str) || *str == ':') strcpy (str,NOM_FILE);
    if (*(str+2) != 92 && *(str+1) != ':' && *str != 92) { /* 92 = '\ ' */
        n_drive = getdisk();
        getcurdir (n_drive+1,cur_dir);
        p = buff;
        *p = n_drive + 65; p++; /* 65 = 'A' */
        *p = ':'; p++;
        *p = 92; p++; *p = 0;
        if (strlen (cur_dir)) {
            strcat (p,cur_dir);
            while (*p != 0) p++;
            *p = 92; p++; *p = 0;
        }
        strcat (p,str);
        strcpy (str,buff);
    } else {
        if (*str == 92) {
            if (!strlen (str+1)) strcpy (str+1,NOM_FILE);
            n_drive = getdisk();
            p = buff;
            *p = n_drive + 65; p++;
            *p = ':'; p++; *p = 0;
            strcat (p,str);
            strcpy (str,buff);
        } else {
            if (*(str+1) == ':' && *(str+2) != 92) {
                if (!strlen(str+2)) strcpy (str+2,NOM_FILE);
                n_drive = *str-65;
            }
        }
    }
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

getcurdir (n_drive+1,cur_dir);

p = buff;
*p = *str; p++;
*p = *(str+1); p++;
*p = 92; p++; *p = 0;
strcat (p,cur_dir);
while (*p != 0) p++;
if (*(p-1) != 92) {
    *p = 92; p++;
}
*p = 0;
strcat (p,str+2);
strcpy (str,buff);
}
}
}
free (cur_dir);
free (buff);
}

```

```

void str_file (str,str1) /* str = destination , str1 = source */
char *str,*str1;
{
    char *s,*q, p[3];
    s = malloc (200);
    strcpy (s,str1);
    q = s;
    while (*q != 0) {
        if (*q == 47) *q = 92;
        q++;
    }
    q = s;
    check_str (q);
    while (*q != 0) q++;
    while (*q != 92) q--;
    q++;
    strcpy (str,q);
    free (s);
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*-----File-----*/
```

```
void load (str)
char str[];
{
    FILE *f;
    char k,i,*p;
    p = (char *)buffer;
    curloc = curmark = mark = (char *)buffer;
    if ((f = fopen (str,"rt")) != NULL) {
        while ((k = getc (f)) != EOF && p != endcur) {
            if (k == '\n') k = ENTER;
            if (k == 9) {
                for (i = 1; i < 9; i++) {
                    k = ' ';
                    *p = k;
                    p++;
                }
            } else {
                *p = k;
                p++;
            }
        }
        endloc = p;
        *endloc = END_BUF;
        fclose (f);
    }
}
```

```
void load_micro(str)
char str[];
{
    FILE *jedecfile;
    char ch,*ppp;
    /*if (check_drive(str[0]) == 0)
    { */
    ppp= malloc(25*80*2);
    gettext(1,1,80,25,ppp);
    clrscr();
```

```

delay(50);
if((jedecfile = fopen(str,"rt"))==NULL)
{
    printf("ERROR");
    getch();
    exit(1);/* Send errorcode 1 to DOS */
}
/* Repeat until no data to read */
while(!feof(jedecfile))
{
    ch = fgetc(jedecfile);
    if(ch >= NULL){
        putchar(ch);
        sport(PORT,ch); /* Character code */
    }
    else if (ch==10)
    {
        printf("\n");
        sport(PORT,ch); /* Character code */
    }
}
sport(PORT,0x03);

fclose(jedecfile); /*Close file */
puttext(1;1,80,25,ppp);
free(ppp);
/*Put newline character to screen*/
/* }
else
    drive_error();*/
}

```

void list 0

```

{
    char k,i,*p;
    p = (char *)buffer;
    while (p <= endloc) {
        if (*p != ENTER)
            printf("%c",*p);
        else printf("\n");
    }
}

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    p++;
}
printf("\n Please any key to continous");
getch();
}

```

```

read_file (str,buf,buf_size,ct)
char *str,*buf;
long buf_size;
int *ct;
{
    struct ffbk blk;
    char *p,*b,*d,done,buf_str[13];
    unsigned int i;
    i = 0;
    *ct = 0;
    p = str;
    while (*p != 0) {
        if (*p == 92) *p = 47; /* 92 = '\,47 = '/' */
        p++;
    }
    d = str;
    while (*d != 0) d++;
    if (d != str) d--;
    while (*d != '/' && *d != '.' && d != str) d--;
    if (d != str) d++;
    strcpy (buf_str,d);
    *d = 0;
    strcat (str,"*.");
    p = buf;
    done = findfirst (str,&blk,0xf7);
    while (!done && i < buf_size-1) {
        if (strcmp (blk.ff_name,".") && (blk.ff_attrib & 16) == 16) {
            b = blk.ff_name;
            while (*b != 0) {
                *p = *b; p++; b++; i++;
            }
            *p = 92; p++; i++; *p = 0; p++; i++;(*ct)++; /* 92 = '\ */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
*d = 0;
strcat (str,buf_str);
done = findfirst (str,&blk,0xe7);
while (!done && i < buf_size-1) {
    b = blk.ff_name;
    while (*b != 0) {
        *p = *b; p++; b++; i++;
    }
    *p = 0; p++; i++; (*ct)++;
    done = findnext (&blk);
}
*p = END_BUF;
if (p == buf) i = 0; else i = 1;
return (i);
}

void show_file (x,y,mark,cur)
char x,y,*mark,*cur;
{
    char x1,y1,i,color,sw;
    x1 = x + 2;
    y1 = y + 1;
    sw = 0;
    color = FILE_T_COLOR;
    while (y1 < y+10) {
        if (mark == cur) {
            sw = 1;
            color = FILE_T_B_COLOR;
        }
        if (x1 < x+58) {
            i = 0;
            while (i < 14) { /* space of file = 14 column */
                if (*mark != 0 && *mark != END_BUF && i > 0) {
                    printxy (x1,y1,*mark,color);
                    mark++;
                } else printxy (x1,y1,' ',color);
                i++;
                x1++;
            }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if (sw && *mark == 0) color = FILE_T_COLOR;
    if (*mark != END_BUF && i > 0) mark++;
} else {
    y1++;
    x1 = x + 2;
}
}
}

```

```

key_enter (str,cur)
char *str,*cur;
{
    char *p,ret,b_str[13];
    p = str;
    ret = 0;
    while (*p != 0) p++; p--;
    while (*p != '/') p--; p++;
    strcpy (b_str,p);
    if (!strcmp (cur,".\\\"")) {
        p -= 2;
        while (*p != '/') p--;
        p++; *p = 0;
        strcat (str,b_str);
    } else {
        *p = 0;
        p = cur;
        while (*p != 92 && *p != 0) p++; /* 92 = '\' */
        if (*p != 0) {
            strcat (str,cur);
            strcat (str,b_str);
        } else {
            strcat (str,cur);
            ret = 1;
        }
    }
    return (ret);
}

```

load_file (str)

char *str;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    char *buf,*endbuf,*menu_file_buf,*mark,*cur,x,y;
    char ch,i,ok_exit,ok;
    int ct,ct_file,ct_scn;
    buf = malloc (BUF_FILE);
    menu_file_buf = malloc (61*11);
    mark = buf;
    cur = buf;
    endbuf = buf;
    x = 10;
    y = 10;
    ct_file = 1;
    ct_scn = 1;
    ok_exit = 0;
    ok = 1;
    gettext (x,y,x+60,y+10,menu_file_buf); /* get MENU_FILE on screen */
    if (!read_file (str,buf,BUF_FILE,&ct)).{
        strcpy (buf,"Not found");
        *(buf+strlen (buf)+1) = END_BUF;
        ok = 0;
    }
    while (*endbuf != END_BUF) endbuf++;
    if (endbuf != buf) {
        endbuf -= 2; \
        while (*endbuf != 0) endbuf--;
        if (endbuf != buf) endbuf++;
    }
    table (x,y,x+60,y+10,1,str,FILE_T_COLOR);
    show_file (x,y,mark,cur);
    do {
        ch = getch ();
        if (ch == 0) {
            ch = getch ();
            swinch (ch) {
                case U_ARRO:
                    if (cur > buf && ct_scn-4 >= 1) {
                        ct_file -= 4;
                        ct_scn -= 4;
                        cur -= 2;
                    }
                    for (i = 1; i <= 4; i++) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        while (*cur != 0 && cur != buf) cur--;
        if (cur != buf) cur--;
    }
    if (cur != buf) cur += 2;
    if (ct_file < 1) {
mark -= 2;
        for (i = 1; i <= 4; i++) {
            while (*mark != 0 && mark != buf) mark--;
            if (mark != buf) mark--;
            ct_file++;
        }
        if (mark != buf) mark += 2;
    }
    show_file (x,y,mark,cur);
}
break;
case D_ARRO :
    if (cur < endbuf && ct_scn+4 <= ct) {
        ct_file += 4;
        ct_scn += 4;
        for (i = 1; i <= 4; i++) {
            while (*cur != 0) cur++; cur++;
        }
        if (ct_file > FILE_SCN) {
            for (i = 1; i <= 4; i++) {
                while (*mark != 0) mark++; mark++;
                ct_file--;
            }
        }
        show_file (x,y,mark,cur);
    }
    break;
case L_ARRO :
    if (cur != buf) {
        cur -= 2;
        while (*cur != 0 && cur != buf) cur--;
        if (cur != buf) cur++;
        ct_file--;
        ct_scn--;
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ if (ct_file < 1) เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        mark -= 2;
    for (i = 1; i <= 4; i++) {
        while (*mark != 0 && mark != buf) mark--;
        if (mark != buf) mark--;
        ct_file++;
    }
    if (mark != buf) mark += 2;
}
show_file (x,y,mark,cur);
}
break;
case R_ARRO :
    if (cur < endbuf) {
        while (*cur != 0) cur++; cur++;
        ct_file++;
        ct_scn++;
        if (ct_file > FILE_SCN) {
            for (i = 1; i <= 4; i++) {
                while (*mark != 0 && mark != endbuf) mark++;
                mark++;
                ct_file--;
            }
            show_file (x,y,mark,cur);
        }
        break;
    }
} else {
    switch (ch) {
        case ENTER :
            if (!key_enter (str,cur)) {
                mark = buf;
                cur = buf;
                endbuf = buf;
                ct_file = 1;
                ct_scn = 1;
                read_file (str,buf,BUF_FILE,&ct);
                while (*endbuf != END_BUF) endbuf++;
                if (endbuf != buf) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ endbuf = 2; เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        while (*endbuf != 0) endbuf--;
        if (endbuf != buf) endbuf++;
    }
    table (x,y,x+60,y+10,1,str,FILE_T_COLOR);
    show_file (x,y,mark,cur);
} else ok_exit = 1;
break;

case ESC :
    ok_exit = 1;
    break;

}
}
} while (!ok_exit);
puttext (x,y,x+60,y+10,menu_file_buf); /* put MENU_FILE on screen */
free (buf);
free (menu_file_buf);
if (ch == ESC || !ok) ok_exit = 0;
return (ok_exit);
}

void os_shell(void)
{
    char *save_scr;
    save_scr = malloc(4000);
    gettext(1,1,80,25,save_scr);
    textbackground(BLACK);
    textcolor(WHITE);
    clrscr();
    cur_on();
    printf("Type EXIT to return to Pal Program. . .");
    system("");
    cur_off();
    puttext(1,1,80,25,save_scr);
    free(save_scr);
}

void setmode(mode)
int mode;
{
    union REGS regs;

```

```

regs.h.ah = 0;
regs.h.al = mode;
int86(0x10,&regs,&regs);
}

int chang_drive(int drive)
{
    union REGS regs;
    regs.h.ah = 0x0E;
    regs.h.dl = toupper(drive) - 'A';
    intdos(&regs,&regs);
    return(regs.h.al);
}

int ch_dir(str)
char *str;
{
    char ok;
    if (check_drive(str[0]) == 0x80)
    {
        drive_error();
        return(0);
    }
    while (check_drive(str[0]) == 0)
    {
        chang_drive(str[0]);
        ok=chdir(str);
        return(ok);
    }
}

```

```

void realy_process(v_pos)
char *v_pos;
{
    char *qqq;
    char *p2;
    char ans,r;
    char rx;
    qqq=malloc(26*3*2);

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการเรียนการสอนเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

p2 = malloc (35*11*2);
gettext(23,10,57,20,p2);
table(23,10,57,20,4,"",0x70);
cur_on();
switch(*v_pos)
{
    case 0:
        gotoxy(51,11);
        printstrxy(24,11," Ready to program (Y/<CR>)?"",0x70);
        break;
}

cur_on();
do {
    ans = getch();
} while (toupper(ans)!='Y' && ans !=ENTER);
cur_off();
if (toupper(ans)=='Y')
{
    switch(*v_pos)
    {
        case 0: /* PROGRAM */
            printstrxy(24,12," Programming now... ",0xAF);
            port_init(PORT,0xE3);
            sport(PORT,'P');
            do{
                rx = rport(PORT);

            }while(rx != 3);
            do {
                rx = rport(PORT);
            } while(rx != 0);
            printstrxy(24,12," Complete ! ",0x70);
            printstrxy(24,13," OK ! ",0x70);
            printf("\a");
            break;
    }
}

```

```

    free(p2);
}

void frist_title(void)
{
    int chk_port;
    char run[37] = { " PAL PROGRAMMER V1.0 ED. ENGINEER 16 " };
    char z,y=1;
    int i,b;
    unsigned int c;
    textbackground(BLUE);
    clrscr();
    do {
        for (z=0;z<y;z++)
        {
            /*gotoxy(72-(y-z),17);
            printf("%c",run[z]);*/
            printxy(59-(y-z),17,run[z],0x3E);
        }
        for (i=0;i<60;i++)
        {
            for (b=0;b<1000;b++)
            {
                c=i*b;
            }
        }
        y++;
    } while (y<=37);
    for (i=0;i<200;i++)
    {
        for (b=0;b<1000;b++)
        {
            c=i*b;
        }
    }
}

```

```

/* ***** micro ***** */

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void port_init(int port,unsigned code)
```

```
{
    union REGS r;
    r.x.dx = port; /* Number port */
    r.h.ah = 0;
    r.h.al = code;
    int86(0x14,&r,&r);
}
```

```
void sport(int port,char c)
```

```
{
    union REGS r;
    r.x.dx = port;
    r.h.ah = 1;
    r.h.al = c;
    int86(0x14,&r,&r);
    if(r.h.ah & 128) {
        printf(" Send error detected in serial port ");
        exit(1);
    }
}
```

```
void sport2(int port2,char c)
```

```
{
    union REGS r;
    r.x.dx = port2;
    r.h.ah = 1;
    r.h.al = c;
    int86(0x14,&r,&r);
}
```

```
rport(int port)
```

```
{
    union REGS r;
    while(! (check_stat(PORT) & 0x100))
        if( kbhit() ) {
            getch();
            exit(1);
        }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

r.x.dx = port;    /* Number port */
r.h.ah = 2;       /* Revc    */
int86(0x14,&r,&r);
if(r.h.ah & 128)
    printf("read error detected in serial port");
return(r.h.al);
}

```

```

rport2(int port2)

```

```

{
    union REGS r;
    r.x.dx = port2;    /* Number port */
    r.h.ah = 2;       /* Revc    */
    int86(0x14,&r,&r);
    return(r.h.al);
}

```

```

check_stat(port)

```

```

int port;
{
    union REGS r;

    r.x.dx = port;
    r.h.ah = 3;
    int86(0x14,&r,&r);
    return r.x.ax;
}

```

```

void dump(void)

```

```

{
    char *du;
    unsigned char rx;
    du=malloc(80*25*2);
    gettext(1,1,80,25,du);
    clrscr();
    printf("Display device control by JEDEC format\n");
    sport(PORT,0xB1); /* Display Device */
    do {

```

```

        rx=rport(PORT);

```

```

    }while (rx!=0);

```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

do {
    rx = rport(PORT);
    if (rx >= SPACE) {
        printf("%c",rx);
        sport(PORT,rx);
    }
    else
    {
        if(rx == NEWLINE){
            printf("\n");
            sport(PORT,NEWLINE);
        }
    }

}while(rx != 03);
printf("\nPress any key to continue");
getch();
puttext(1,1,80,25,du);
free(du);
}

void edit_buf(void)
{
    char ch=255;

    table(1,2,71,22,4,"",0x1F);
    table(71,2,80,22,4,"",0x1F);
    printstrxy(6,3,"0000000000111111111122222222223333333333444444444455555555556666",0x1F);
    printstrxy(6,4,"0123456789012345678901234567890123456789012345678901234567890123",0x1F);
    printstrxy(72,3,"Command:",0x1F);
    printstrxy(72,4,"PgUp",0x1F);
    printstrxy(72,5,"PgDn",0x1F);
    printstrxy(72,6,"ESC",0x1F);
    printstrxy(1,23," Symbol : n=no fuse, 0=intact, 1=blow",0x1F);
    disp_scn (STARTX,STARTY,ENDX,ENDY,1,buffer,ON);
    while (ch!=ESC)
    {
        ch=getch();
        if (ch==0)
            ch=getch();
    }
}

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อการเรียนการสอนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

switch(ch)
{
    case PG_UP :
        pg_up();
        break;
    case PG_DN :
        pg_dn();
        break;
}
}

```

```

void change_mark (direct)
char direct;
{
    if (direct == 'D') {
        while (*mark != ENTER && mark < endloc) mark++;
        if (*mark == ENTER) mark++;
    } else {
        if (mark > buffer) mark --;
        if (mark > buffer) {
            mark --;
            while (*mark != ENTER && mark > buffer) mark--;
            if (*mark == ENTER) mark++;
        }
    }
}

```

```

void pg_up (void)
{
    unsigned int ct;
    int s;
    for(s=0;s<15;s++){
        if (row > 1) {
            ct = 1;
            if (*curloc == ENTER && curloc > buffer) curloc--;
            while (*curloc != ENTER && curloc > buffer) curloc--;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while (*curloc != ENTER && curloc > buffer) curloc--;
if (*curloc == ENTER) curloc++;
}
curmark = curloc;
while (ct < col) {
    ct++;
    if (*curloc != ENTER && curloc < endloc) {
        curloc++;
        curmark++;
    } else curmark++;
}
row--;
if (y > STARTY) y--;
else {
    change_mark ('U');
/*    page_dn (y,ENDY);*/
    if (sft)* disp_scn (STARTX,STARTY,ENDX,ENDY,sft+1,mark,ON);*/
    else re_line_ed3 (STARTX,mark);
}
}
}
gotoxy (x,y);
disp_scn (STARTX,STARTY,ENDX,ENDY,sft+1,mark,ON);
}

void pg_dn (void)
{
    char *p,*d;
    int s;
    unsigned int ct;
    ct = 1;

    for(s=0;s<15;s++){
        d = curloc;
        while (*curloc != ENTER && curloc < endloc) curloc++;
        if (*curloc == ENTER) {
            curloc++;
            curmark = p = curloc;
            while (ct < col) {
                ct++;

```

```

if (*curloc != ENTER && curloc < endloc-1) {
    curloc++;
    curmark++;
} else {
    curmark++;
}
}
row++;
y++;
if (y > ENDY) {
    y--;
    change_mark ('D');
/* page_up (STARTY,y);*/
if (sft)/* disp_scn (STARTX,y,ENDX,y,sft+1,p,ON);*/;
else re_line_ed3 (STARTX,p);
}
} else curloc = d;
}
gotoxy (x,y);
disp_scn (STARTX,STARTY,ENDX,ENDY,sft+1,p,ON);
}

/*-----Key of Editor-----*/

void up (void)
{
    unsigned int ct;
    if (row > 1) {
        ct = 1;
        if (*curloc == ENTER && curloc > buffer) curloc--;
        while (*curloc != ENTER && curloc > buffer) curloc--;
        if (curloc > buffer) {
            curloc--;
            while (*curloc != ENTER && curloc > buffer) curloc--;
            if (*curloc == ENTER) curloc++;
        }
        curmark = curloc;
        while (ct < col) {
            ct++;
            if (*curloc != ENTER && curloc < endloc) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        curloc++;
        curmark++;
    } else curmark++;
}
row--;
if (y > STARTY) y--;
else {
    change_mark ('U');
    page_dn (y,ENDY);
    if (sft) disp_scn (STARTX,y,ENDX,y,sft+1,mark,ON);
    else re_line_ed (STARTX,y,mark);
}
gotoxy (x,y);
}
}

void dn (void)
{
    char *p,*d;
    unsigned int ct;
    ct = 1;
    d = curloc;
    while (*curloc != ENTER && curloc < endloc) curloc++;
    if (*curloc == ENTER) {
        curloc++;
        curmark = p = curloc;
        while (ct < col) {
            ct++;
            if (*curloc != ENTER && curloc < endloc) {
                curloc++;
                curmark++;
            } else {
                curmark++;
            }
        }
        row++;
        y++;
        if (y > ENDY) {
            y--;
            change_mark ('D');

```

```

    page_up (STARTY,y);
    if (sft) disp_scn (STARTX,y,ENDX,y,sft+1,p,ON);
    else re_line_ed (STARTX,y,p);
}
gotoxy (x,y);
} else curloc = d;
}

```

```
void re_line_ed (x,y,cur)
```

```
char x,y,*cur;
```

```

{
while (x <= ENDX) {
    if (*cur != ENTER && *cur != END_BUF) {
        printxy (x,y,*cur,0x1F);
        cur++;
    } else printxy (x,y,' ',0x1F);
    x++;
}
}

```

```
void disp_scn (x,y,xs,ys,col,cur,all)
```

```
char x,y,xs,ys;
```

```
unsigned int col;
```

```
char *cur,all;
```

```

{
char sw;
unsigned int ct;
ct = 1;
sw = ON;
while (y <= ys) {
    if (x <= xs) {
        while (ct < col && *cur != ENTER && *cur != END_BUF) {
            cur++;
            ct++;
        }
        if (*cur != ENTER && *cur != END_BUF) {
            printxy (x,y,*cur,0x1F);
            cur++;
        } else {
            if (sw) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        printxy(x,y,'',0x1F);
        if (!all) sw = OFF;
    }

}

x++;
} else {
    while (*cur != ENTER && *cur != END_BUF) cur++;
    if (*cur == ENTER) cur++;
    x = STARTX;
    sw = ON;
    ct = 1;
    y++;
}
}
}

```

```
void page_up (y1,y2)
```

```
char y1,y2;
```

```

{
    union REGS r;
    r.h.ah = 6;
    r.h.al = 1;
    r.h.bh = 0x1F;
    r.h.cl = STARTX - 1;
    r.h.ch = y1 - 1;
    r.h.dl = ENDX - 1;
    r.h.dh = y2 - 1;
    int86 (0x10,&r,&r);
}

```

```
void page_dn (y1,y2)
```

```
char y1,y2;
```

```

{
    union REGS r;
    r.h.ah = 7;
    r.h.al = 1;
    r.h.bh = 0x1F;
    r.h.cl = STARTX-1;
    r.h.ch = y1 - 1 ;
    r.h.dl = ENDX - 1;

```

```

r.h.dh = y2 - 1;
int86 (0x10,&r,&r);
}

void re_line_ed3 (x,cur)
char x,*cur;
{
while (x <= ENDX) {
    if (*cur != ENTER && *cur != END_BUF)
        cur++;
    else
        x++;
}
}

void path(void)
{
char ch,*pass_ok;
pass_ok = malloc(31*2*3);
gettext(30,12,60,14,pass_ok);
table(30,12,60,14,4,"",0x4F);
printstrxy(32,13,"Path not found : Please ESC",0x4F);
do {
    ch=getch();
}while (ch!=ESC);
puttext(30,12,60,14,pass_ok);
free(pass_ok);
}

void drive_error(void)
{
char ch,*error;
error = malloc(32*2*3);
gettext(30,12,61,14,error);
table(30,12,61,14,4,"",0x4F);
printstrxy(32,13,"Drive not ready : Please ESC",0x4F);
do {
    ch=getch();
}while (ch!=ESC);
puttext(30,12,61,14,error);
}

```

MONITOR PAL PROGRAMMER V1.0

Binary 8051 Cross-Assembler (1.3) (C) 1987, 1989 Technology

palpgm.asm

```

1
FC00= 2      P15V      EQU      0FC00H
FC01= 3      P112V     EQU      0FC01H
FC02= 4      PDATA     EQU      0FC02H
FC03= 5      CONF_1    EQU      0FC03H      ;8255/1
6
F800= 7      PQA       EQU      0F800H
F801= 8      PGM       EQU      0F801H
F802= 9      P_CON     EQU      0F802H
F803= 10     CONF_2    EQU      0F803H      ;8255/2
11
FA00= 12     LCDWRC     EQU      0FA00H      ;LCD WRITE CONTROL
FA01= 13     LCDRDC     EQU      0FA01H      ;LCD READ CONTROL (BUSY CHECK)
FA02= 14     LCDWRD     EQU      0FA02H      ;LCD WRITE DATA
FA03= 15     LCDRDD     EQU      0FA03H      ;LCD READ DATA
9000= 16     LCDBUF     EQU      9000H      ;LCD RAM-BU
17
0000= 18     BUFFER     EQU      0000H
8000= 19     BUFFER2    EQU      8000H
001E= 20     DELO       EQU      30
21
22      ; ***** WORKING RAM *****
23
0020 24      ORG 20H
25
0020 26      LR:         DS      1
0021 27      AHIGH:      DS      1
0022 28      ALOW:       DS      1
0023 29      RHIGH:      DS      1
0024 30      RLOW:       DS      1
0025 31      CODEFUSE:   DS      1
0026 32      ADDIXX:     DS      2
0028 33      ADDIYY:     DS      2
002A 34      X:          DS      1

```

```

002B      35      Y:      DS      1
          36
0060      37      ORG 60H
          38
0060      39      STACK: DS      20
          40
          41      ;***** MAIN PROGRAM *****
0000      42      ORG 0000H
          43
0000 020100 44      LJMP RES
          45
          46      ;***** RESET SYSTEM *****
0100      47      ORG 0100H
          48
0100 7A80   49      RES:   MOV    R2,#80H
0102 7B00   50      RES1:  MOV    R3,#0
0104 DBFE   51      DJNZ   R3,$
0106 DAFA   52      DJNZ   R2,RES1 ;DELAY SYSTEM
          53
          54
0108 758160 55      MOV    SP,#STACK
010B 759852 56      MOV    SCON,#52H
010E 758920 57      MOV    TMOD,#20H ;SET SERIAL PORT
0111 758DFD 58      MOV    TH1,#0FDH ;BUAD RATE 9600
0114 D28E   59      SETB   TR1
          60
          61
0116 7489   62      MOV    A,#89H ;| SET SYSTEM
0118 90FC03 63      MOV    DPTR,#CONF_1 ;A = B = OUT
011B F0     64      MOVX   @DPTR,A ;C = IN
          65
011C 7480   66      MOV    A,#80H ;A = B = C = OUT
011E 90F803 67      MOV    DPTR,#CONF_2
0121 F0     68      MOVX   @DPTR,A
          69
0122 120967 70      LCALL  CLRPGM ;| END SET SYSTEM
          71
0125 7438   72      MOV    A,#00111000B ;FUNCTION SET
0127 120A77 73      LCALL  LCDWI
012A 740C   74      MOV    A,#00001100B ;DISPLAY ON/OFF

```

```

012C 120A77    75          LCALL  LCDWI
012F 7401      76          MOV   A,#01H          ;CLEAR
0131 120A77    77          LCALL  LCDWI
                                78
                                79
0134 900E6D    80      MAIN4:  MOV   DPTR,#MENU4T  ;FIRST DISPLAY
0137 120A9F    81          LCALL  LCDLDP
013A 7A0A      82          MOV   R2,#10
013C 12089F    83          LCALL  DTSEC
                                84
013F 7814      85          MOV   R0,#20
0141 7418      86      SHIFT:  MOV   A,#00011000B
0143 120A77    87          LCALL  LCDWI
0146 7A03      88          MOV   R2,#3
0148 12089F    89          LCALL  DTSEC
014B D8F4      90          DJNZ  R0,SHIFT
                                91
014D 740C      92      LOOP:   MOV   A,#00001100B  ;DISPLAY ON/OFF
014F 120A77    93          LCALL  LCDWI
0152 7401      94          MOV   A,#01H          ;CLEAR
0154 120A77    95          LCALL  LCDWI
0157 7480      96          MOV   A,#80H          ;LINE 1
0159 120A77    97          LCALL  LCDWI
                                98
015C 900E95    99          MOV   DPTR,#NUM_BER
015F 7814     100          MOV   R0,#20
0161 E4        101      LINE1:  CLR   A
0162 93        102          MOVC  A,@A+DPTR
0163 120A8B    103          LCALL  LCDWD
0166 A3        104          INC   DPTR
0167 D8F8     105          DJNZ  R0,LINE1
                                106
0169 74AF     107      LOOP2:  MOV   A,#0AFH
016B 1209BD    108          LCALL  SBYTE
016E 3098F8    109          JNB   RI,LOOP2
                                110
0171 1209E5    111          LCALL  RBYTE
0174 B4AFF2    112          CJNE  A,#0AFH,LOOP2
                                113
0177 740F     114          MOV   A,#00001111B

```

0179	120A77	115		LCALL	LCDWT	
017C	74C0	116		MOV	A,#0C0H	
017E	120A77	117		LCALL	LCDWT	;X=1,Y=2
0181	743A	118		MOV	A,#:'	
0183	120A8B	119		LCALL	LCDWD	
0186	743E	120		MOV	A,#>'	
0188	120A8B	121		LCALL	LCDWD	
		122				
018B	20B303	123	WAIT:	JB	INT1,\$+6	
018E	7590C0	124		MOV	P1,#0C0H	;ERROR
0191	3098F7	125		JNB	RI,WAIT	
		126				
0194	1209E5	127	W:	LCALL	RBYTE	
0197	B42006	128		CJNE	A,#20H,CODE_L	
019A	1203FA	129		LCALL	MENU	
019D	02018B	130		LJMP	WAIT	
		131				
01A0	B4B02D	132	CODE_L:	CJNE	A,#0B0H,CODE_D	;LOAD
01A3	752A03	133		MOV	X,#3	
01A6	752B02	134		MOV	Y,#2	
01A9	120ADC	135		LCALL	GOTOXY	
		136				
01AC	900EFC	137		MOV	DPTR,#DLOAD	
01AF	120AD0	138		LCALL	PRINTF	
		139				
01B2	120438	140		LCALL	LOAD	
		141				
01B5	752A03	142		MOV	X,#3	
01B8	752B02	143		MOV	Y,#2	
01BB	120ADC	144		LCALL	GOTOXY	
		145				
01BE	900F35	146		MOV	DPTR,#DCLS	
01C1	120AD0	147		LCALL	PRINTF	
		148				
01C4	752A03	149		MOV	X,#3	
01C7	752B02	150		MOV	Y,#2	
01CA	120ADC	151		LCALL	GOTOXY	
		152				
01CD	02018B	153		LJMP	WAIT	

01CD 02018B 153 LJMP WAIT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

154

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

01D0 B4B132	155	CODE_D:	CJNE	A,#0B1H,CODE_3 ;DISPLAY DEVICE
	156			
01D3 752A03	157		MOV	X,#3
01D6 752B02	158		MOV	Y,#2
01D9 120ADC	159		LCALL	GOTOXY
	160			
01DC 900EE9	161		MOV	DPTR,#DDISPLAY
01DF 120AD0	162		LCALL	PRINTF
	163			
01E2 7400	164		MOV	A,#00
01E4 1209BD	165		LCALL	SBYTE
	166			
01E7 1204F6	167		LCALL	DISP
	168			
01EA 752A03	169		MOV	X,#3
01ED 752B02	170		MOV	Y,#2
01F0 120ADC	171		LCALL	GOTOXY
	172			
01F3 900F35	173		MOV	DPTR,#DCLS
01F6 120AD0	174		LCALL	PRINTF
	175			
01F9 752A03	176		MOV	X,#3
01FC 752B02	177		MOV	Y,#2
01FF 120ADC	178		LCALL	GOTOXY
	179			
0202 02018B	180		LJMP	WAIT
	181			
0205 B43303	182	CODE_3:	CJNE	A,#'3',CODE_4
0208 02083A	183		LJMP	DUMP
	184			
020B B43403	185	CODE_4:	CJNE	A,#'4',CODE_5
020E 020816	186		LJMP	DUMPRAM
	187			
0211 B43909	188	CODE_5:	CJNE	A,#'9',CODE_P
0214 12085E	189		LCALL	CLRRAM
0217 120872	190		LCALL	CLRRAM2
021A 02018B	191		LJMP	WAIT
	192			
021D B4503B	193	CODE_P:	CJNE	A,#'P',CODE_R
0220 752A03	194		MOV	X,#3

0223	752B02	195	MOV	Y,#2
0226	120ADC	196	LCALL	GOTOXY
		197		
0229	900F22	198	MOV	DPTR,#DPROGRAM
022C	120AD0	199	LCALL	PRINTF
		200		
022F	12057C	201	LCALL	PALPGM
0232	1209B2	202	LCALL	SLF
0235	7403	203	MOV	A,#"
0237	1209BD	204	LCALL	SBYTE
023A	12091B	205	LCALL	D60USEC
023D	759055	206	MOV	P1,#55H
		207		
0240	752A03	208	MOV	X,#3
0243	752B02	209	MOV	Y,#2
0246	120ADC	210	LCALL	GOTOXY
		211		
0249	900F35	212	MOV	DPTR,#DCLS
024C	120AD0	213	LCALL	PRINTF
		214		
024F	752A03	215	MOV	X,#3
0252	752B02	216	MOV	Y,#2
0255	120ADC	217	LCALL	GOTOXY
		218		
0258	02018B	219	LJMP	WAIT
		220		
025B	B4523E	221	CODE_R: CJNE	A,#'R',CODE_A
		222		
025E	752A03	223	MOV	X,#3
0261	752B02	224	MOV	Y,#2
0264	120ADC	225	LCALL	GOTOXY
		226		
0267	900F0F	227	MOV	DPTR,#DREAD
026A	120AD0	228	LCALL	PRINTF
		229		
026D	120954	230	LCALL	D500USEC
0270	1209B2	231	LCALL	SLF
0273	120676	232	LCALL	READ_FUSE
		233		
0276	7403	233	MOV	A,#"
0278	1209BD	234	LCALL	SBYTE

027B	12091B	235		LCALL	D60USEC
027E	759055	236		MOV	P1,#55H
		237			
0281	752A03	238		MOV	X,#3
0284	752B02	239		MOV	Y,#2
0287	120ADC	240		LCALL	GOTOXY
		241			
028A	900F35	242		MOV	DPTR,#DCLS
028D	120AD0	243		LCALL	PRINTF
		244			
0290	752A03	245		MOV	X,#3
0293	752B02	246		MOV	Y,#2
0296	120ADC	247		LCALL	GOTOXY
		248			
0299	02018B	249		LJMP	WAIT
		250			
029C	B4A01B	251	CODE_A:	CJNE	A,#0A0H,CODE_M
		252	;	MOV	A,#10000110B
		253	;	LCALL	LCDWI
029F	752A06	254		MOV	X,#6
02A2	752B01	255		MOV	Y,#1
02A5	120ADC	256		LCALL	GOTOXY
		257			
02A8	900EAA	258		MOV	DPTR,#DAMD
02AB	120AD0	259		LCALL	PRINTF
		260			
02AE	752A03	261		MOV	X,#3
02B1	752B02	262		MOV	Y,#2
02B4	120ADC	263		LCALL	GOTOXY
		264			
02B7	02018B	265		LJMP	WAIT
		266			
02BA	B4A11B	267	CODE_M:	CJNE	A,#0A1H,CODE_N
02BD	752A06	268		MOV	X,#6
02C0	752B01	269		MOV	Y,#1
02C3	120ADC	270		LCALL	GOTOXY
		271			
02C6	900EAF	272		MOV	DPTR,#DMMI
02C9	120AD0	273		LCALL	PRINTF
		274			

02CC	752A03	275		MOV	X,#3
02CF	752B02	276		MOV	Y,#2
02D2	120ADC	277		LCALL	GOTOXY
		278			
02D5	02018B	279		LJMP	WAIT
		280			
		281			
02D8	B4A21B	282	CODE_N:	CJNE	A,#0A2H,C_16R4
02DB	752A06	283		MOV	X,#6
02DE	752B01	284		MOV	Y,#1
02E1	120ADC	285		LCALL	GOTOXY
		286			
02E4	900EB4	287		MOV	DPTR,#DNS
02E7	120AD0	288		LCALL	PRINTF
		289			
02EA	752A03	290		MOV	X,#3
02ED	752B02	291		MOV	Y,#2
02F0	120ADC	292		LCALL	GOTOXY
		293			
02F3	02018B	294		LJMP	WAIT
		295			
02F6	B4801B	296	C_16R4:	CJNE	A,#80H,C_16R6
		297	;	MOV	A,#10010000B
		298	;	LCALL	LCDWI ;GOTOXY(17,1);
02F9	752A10	299		MOV	X,#16
02FC	752B01	300		MOV	Y,#1
02FF	120ADC	301		LCALL	GOTOXY
		302			
0302	900EB9	303		MOV	DPTR,#D16R4
0305	120AD0	304		LCALL	PRINTF
		305			
0308	752A03	306		MOV	X,#3
030B	752B02	307		MOV	Y,#2
030E	120ADC	308		LCALL	GOTOXY
		309			
0311	02018B	310		LJMP	WAIT
		311			
0314	B4811B	312	C_16R6:	CJNE	A,#81H,C_16L8
0317	752A10	313		MOV	X,#16
031A	752B01	314		MOV	Y,#1

031D	120ADC	315		LCALL	GOTOXY
		316			
0320	900EBF	317		MOV	DPTR,#D16R6
0323	120AD0	318		LCALL	PRINTF
0326	752A03	319		MOV	X,#3
0329	752B02	320		MOV	Y,#2
032C	120ADC	321		LCALL	GOTOXY
		322			
032F	02018B	323		LJMP	WAIT
		324			
0332	B4821B	325	C_16L8:	CJNE	A,#82H,C_16R8
0335	752A10	326		MOV	X,#16
0338	752B01	327		MOV	Y,#1
033B	120ADC	328		LCALL	GOTOXY
		329			
033E	900EC5	330		MOV	DPTR,#D16L8
0341	120AD0	331		LCALL	PRINTF
		332			
0344	752A03	333		MOV	X,#3
0347	752B02	334		MOV	Y,#2
034A	120ADC	335		LCALL	GOTOXY
		336			
034D	02018B	337		LJMP	WAIT
		338			
0350	B4831B	339	C_16R8:	CJNE	A,#83H,C_20R4
0353	752A10	340		MOV	X,#16
0356	752B01	341		MOV	Y,#1
0359	120ADC	342		LCALL	GOTOXY
		343			
035C	900ECB	344		MOV	DPTR,#D16R8
035F	120AD0	345		LCALL	PRINTF
0362	752A03	346		MOV	X,#3
0365	752B02	347		MOV	Y,#2
0368	120ADC	348		LCALL	GOTOXY
		349			
036B	02018B	350		LJMP	WAIT
		351			
036E	B4841B	352	C_20R4:	CJNE	A,#84H,C_20R6
0371	752A10	353		MOV	X,#16
0374	752B01	354		MOV	Y,#1

0377	120ADC	355		LCALL	GOTOXY
		356			
037A	900ED1	357		MOV	DPTR,#D20R4
037D	120AD0	358		LCALL	PRINTF
0380	752A03	359		MOV	X,#3
0383	752B02	360		MOV	Y,#2
0386	120ADC	361		LCALL	GOTOXY
		362			
0389	02018B	363		LJMP	WAIT
		364			
038C	B4851B	365	C_20R6:	CJNE	A,#85H,C_20L8
038F	752A10	366		MOV	X,#16
0392	752B01	367		MOV	Y,#1
0395	120ADC	368		LCALL	GOTOXY
		369			
0398	900ED7	370		MOV	DPTR,#D20R6
039B	120AD0	371		LCALL	PRINTF
039E	752A03	372		MOV	X,#3
03A1	752B02	373		MOV	Y,#2
03A4	120ADC	374		LCALL	GOTOXY
		375			
03A7	02018B	376		LJMP	WAIT
		377			
03AA	B4861B	378	C_20L8:	CJNE	A,#86H,C_20R8
03AD	752A10	379		MOV	X,#16
03B0	752B01	380		MOV	Y,#1
03B3	120ADC	381		LCALL	GOTOXY
		382			
		383			
03B6	900EDD	384		MOV	DPTR,#D20L8
03B9	120AD0	385		LCALL	PRINTF
03BC	752A03	386		MOV	X,#3
03BF	752B02	387		MOV	Y,#2
03C2	120ADC	388		LCALL	GOTOXY
		389			
03C5	02018B	390		LJMP	WAIT
		391			
03C8	B4871B	392	C_20R8:	CJNE	A,#87H,L_FA
03CB	752A10	393		MOV	X,#16
03CE	752B01	394		MOV	Y,#1

03D1	120ADC	395		LCALL	GOTOXY
		396			
03D4	900EE3	397		MOV	DPTR,#D20R8
03D7	120AD0	398		LCALL	PRINTF
		399			
03DA	752A03	400		MOV	X,#3
03DD	752B02	401		MOV	Y,#2
03E0	120ADC	402		LCALL	GOTOXY
		403			
03E3	02018B	404		LJMP	WAIT
		405			
03E6	B4FA0E	406	L_FA:	CJNE	A,#0FAH,L_W
03E9	74C0	407		MOV	A,#0C0H
03EB	120A77	408		LCALL	LCDWI
03EE	900F35	409		MOV	DPTR,#DCLS
03F1	120AD0	410		LCALL	PRINTF
03F4	02014D	411		LJMP	LOOP
		412			
03F7	02018B	413	L_W:	LJMP	WAIT
		414			
		415			
		416			
		417		; ***** MONI SUB *****	
		418		; DISPLAY TO MONITOR	
		419		; IN = NO REG	
		420		; OUT = NO REG ;DISPLAY TO MONITOR	
		421		; REG = A,DPTR	
		422			
03FA	C083	423	MENU:	PUSH	DPH
03FC	C082	424		PUSH	DPL
		425			
03FE	1209B2	426		LCALL	SLF
0401	1209B2	427		LCALL	SLF
0404	1208F5	428		LCALL	D40USEC
		429			
0407	900AED	430		MOV	DPTR,#LO01
040A	120981	431		LCALL	SBLOCK
040D	1208F5	432		LCALL	D40USEC
		433			
0410	900B02	434		MOV	DPTR,#LO02

0413	120981	435	LCALL	SBLOCK	
0416	1208F5	436	LCALL	D40USEC	
		437			
0419	900B19	438	MOV	DPTR,#LO03	
041C	120981	439	LCALL	SBLOCK	
041F	1208F5	440	LCALL	D40USEC	
0422	1209B2	441	LCALL	SLF	
0425	1209B2	442	LCALL	SLF	
0428	1208F5	443	LCALL	D40USEC	
		444			
042B	742D	445	MOV	A,#'-'	
042D	1209BD	446	LCALL	SBYTE	
0430	1208F5	447	LCALL	D40USEC	
0433	D082	448	POP	DPL	
0435	D083	449	POP	DPH	
0437	22	450	RET		
		451			
		452			
		453	:***** LOAD SUB *****		
		454	; LOAD DATA JEDEC FROM PC TO BOARD		
		455	; IN = DATA JEDEC		
		456	; OUT = DATA MEMORY RAM		
		457	; REG = A,R2,R3,DPTR		
		458			
0438	1209B2	459	LOAD:	LCALL	SLF
043B	1209B2	460		LCALL	SLF
043E	900C8F	461		MOV	DPTR,#D_LOAD1
0441	120981	462		LCALL	SBLOCK
		463			
0444	900CA8	464		MOV	DPTR,#D_LOAD2
0447	120981	465		LCALL	SBLOCK
044A	120954	466		LCALL	D500USEC
		467			
044D	908000	468		MOV	DPTR,#BUFFER2
0450	20B303	469	RECV:	JB	INT1,RECV01
0453	7590C0	470		MOV	P1,#0C0H ;ERROR
0456	3098F7	471	RECV01:	JNB	RI,RECV
		472			
0459	1209E5	473	RECV02:	LCALL	RBYTE
045C	B40307	474		CJNE	A,#",LOAD01

เอกสารนี้เป็นทรัพย์สินของหน่วยงานราชการสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

045F 120954	475		LCALL	D500USEC
0462 F0	476		MOVX	@DPTR,A
0463 02046F	477		LJMP	LOAD02
	478			
0466 F0	479	LOAD01:	MOVX	@DPTR,A
0467 A3	480		INC	DPTR
0468 020450	481		LJMP	RECV
	482			
046B A3	483	LOAD04:	INC	DPTR
046C 02047A	484		LJMP	LOAD03
	485			
046F 7A0A	486	LOAD02:	MOV	R2,#10 ;25
0471 120896	487		LCALL	DMSEC
0474 12085E	488		LCALL	CLRRAM
0477 908000	489		MOV	DPTR,#BUFFER2
	490			
047A E0	491	LOAD03:	MOVX	A,@DPTR
047B B446ED	492		CJNE	A,#F,LOAD04
047E A3	493		INC	DPTR
047F E0	494		MOVX	A,@DPTR
	495			
0480 B430E8	496		CJNE	A,#0,LOAD04
0483 A3	497		INC	DPTR
0484 E0	498		MOVX	A,@DPTR
	499			
0485 B42AE3	500		CJNE	A,#*,LOAD04
0488 A3	501	LOAD2:	INC	DPTR
0489 E0	502		MOVX	A,@DPTR
	503			
048A B44CFB	504	LOAD21:	CJNE	A,#L,LOAD2
048D A3	505		INC	DPTR
048E E0	506		MOVX	A,@DPTR
048F FA	507		MOV	R2,A
	508			
0490 A3	509		INC	DPTR
0491 E0	510		MOVX	A,@DPTR
0492 FB	511		MOV	R3,A
	512			
0493 1209FB	513		LCALL	ATOH

	515		
0498 A3	516	INC	DPTR
0499 E0	517	MOVX	A,@DPTR
049A FA	518	MOV	R2,A
	519		
049B A3	520	INC	DPTR
049C E0	521	MOVX	A,@DPTR
049D FB	522	MOV	R3,A
	523		
049E 1209FB	524	LCALL	ATOH
04A1 F524	525	MOV	RLOW,A
	526		
04A3 7900	527	MOV	R1,#0
04A5 AA23	528	MOV	R2,RHIGH
04A7 AB24	529	MOV	R3,RLOW
	530		
04A9 C083	531	PUSH	DPH
04AB C082	532	PUSH	DPL
	533		
04AD 120A2E	534	LCALL	DTOH
04B0 858321	535	MOV	AHIGH,DPH
04B3 858222	536	MOV	ALOW,DPL
04B6 D082	537	POP	DPL
04B8 D083	538	POP	DPH
	539		
04BA A3	540	LOAD3: INC	DPTR
04BB E0	541	MOVX	A,@DPTR
	542		
04BC B43006	543	CJNE	A,#'0',LOAD4
04BF 1204DF	544	LCALL	S_JEDEC
04C2 0204BA	545	LJMP	LOAD3
	546		
04C5 B43106	547	LOAD4: CJNE	A,#'1',LOAD5
04C8 1204DF	548	LCALL	S_JEDEC
04CB 0204BA	549	LJMP	LOAD3
	550		
04CE B44C03	551	LOAD5: CJNE	A,#'L',END_LOAD
04D1 02048A	552	LJMP	LOAD21
	553		
04D4 B40304	554	END_LOAD: CJNE	A,#'',END_LOAD2

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับ END_LOAD: ที่ CINE ศึกษา A, #', END_LOAD2 ถูกนำมาใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

04D7 12091B      555          LCALL  D60USEC
04DA 22          556          RET
                    557
04DB B443DC      558      END_LOAD2: CJNE  A,#'C',LOAD3
04DE 22          559          RET
                    560
                    561
04DF C083        562      S_JEDEC:  PUSH   DPH
04E1 C082        563          PUSH   DPL
                    564
04E3 852183      565          MOV    DPH,AHIGH
04E6 852282      566          MOV    DPL,ALOW
                    567
04E9 F0          568          MOVX   @DPTR,A
04EA A3          569          INC     DPTR
                    570
04EB 858321      571          MOV    AHIGH,DPH
04EE 858222      572          MOV    ALOW,DPL
                    573
04F1 D082        574          POP    DPL
04F3 D083        575          POP    DPH
                    576
04F5 22          577          RET
                    578
                    579      ; ***** DISP SUB *****
                    580      ; DISPLAY BUFFER JEDEC FILE
                    581      ; IN  = DPTR ==> BUFFER
                    582      ; OUT = MONITOR
                    583      ; REG = A,
                    584
                    585
04F6            586      DISP:      ; MOV    DPTR,#LOGODISP
                    587                  ; LCALL  SBLOCK
                    588
                    589                  ; LCALL  SLF
04F6 900000      590          MOV    DPTR,#BUFFER
                    591
04F9 7900        592          MOV    R1,#0
04FB 7A00        593          MOV    R2,#0
04FD 7B00        594          MOV    R3,#0

```

04FF 120954	595	LCALL D500USEC	
0502 120954	596	LCALL D500USEC	
	597		
0505 1209B2	598	DISP01: LCALL SLF	
	599		
0508 1209E5	600	LCALL RBYTE	
	601		
050B 7F0C	602	MOV R7,#12	;12,10,11,24
050D 120954	603	LCALL D500USEC	
0510 DFFB	604	DJNZ R7,\$-3	
	605		
0512 744C	606	MOV A,#'L'	
0514 1209BD	607	LCALL SBYTE	
	608		
0517 1209E5	609	LCALL RBYTE	;CHECK SUM
	610		
051A C083	611	PUSH DPH	
051C C082	612	PUSH DPL	
051E C001	613	PUSH 01	
0520 C002	614	PUSH 02	
0522 C003	615	PUSH 03	
	616		
0524 120A58	617	LCALL HTOD	;CONVERT DPTR ==> DECIMAL
0527 8A21	618	MOV AHIGH,R2	
0529 8B22	619	MOV ALOW,R3	
	620		
052B E521	621	MOV A,AHIGH	
052D 1209D3	622	LCALL SBYTEHC	;SEND 2 BYTE AHIGH
	623		
0530 E522	624	MOV A,ALOW	
0532 1209D3	625	LCALL SBYTEHC	;SEND 2 BYTE ALOW
	626		
0535 D003	627	POP 03	
0537 D002	628	POP 02	
0539 D001	629	POP 01	
053B D082	630	POP DPL	
053D D083	631	POP DPH	
	632		
053F 7420	633	MOV A,#'	
0541 1209BD	634	LCALL SBYTE	

```

635
0544 1209E5 636          LCALL  RBYTE          ;CHECK SUM
0547 1208CF 637          LCALL  D20USEC
638
054A BA2014 639  DIS02:  CJNE   R2,#32,DISP03
054D 1208CF 640          LCALL  D20USEC
0550 742A   641          MOV   A,#"*
0552 1209BD 642          LCALL  SBYTE
643
0555 1209E5 644          LCALL  RBYTE          ;CHECK SUM
0558 7A00   645          MOV   R2,#0
055A 0B     646          INC   R3
055B BB40A7 647          CJNE   R3,#64,DISP01
055E 02056D 648          LJMP   DIS04
649
0561 E0     650  DISP03:  MOVX   A,@DPTR
0562 1209BD 651          LCALL  SBYTE
652
653
0565 1209E5 654          LCALL  RBYTE          ;CHECK SUM
0568 0A     655          INC   R2
0569 A3     656          INC   DPTR
056A 02054A 657          LJMP   DIS02
658
056D 1209B2 659  DIS04:  LCALL  SLF
0570 7403   660          MOV   A,#"
0572 1209BD 661          LCALL  SBYTE          ;DISPLAY OK
0575 1209E5 662          LCALL  RBYTE          ;CHECK SUM
0578 12091B 663          LCALL  D60USEC
057B 22     664          RET
665
666          ; ***** PALPGM SUB *****
667          ; PROGRAMMER PAL DEVICE
668          ; IN = DATA FILE JEDEC
669          ; OUT = PAL DEVICE
670          ; REG =
671
057C 1209B2 672  PALPGM:  LCALL  SLF
673
057F 900CF2 674          MOV   DPTR,#PGMFUSE

```

0582	120981	675	LCALL	SBLOCK	
0585	2092E	676	LCALL	D80USEC	
		677			
0588	900D1B	678	MOV	DPTR,#NUMBER	
058B	120981	679	LCALL	SBLOCK	
058E	12092E	680	LCALL	D80USEC	
		681			
0591	900D44	682	MOV	DPTR,#IN_LINE	
0594	120981	683	LCALL	SBLOCK	
0597	12092E	684	LCALL	D80USEC	
059A	1209B2	685	LCALL	SLF	
		686			
059D	900000	687	MOV	DPTR,#0000	
05A0	7A00	688	MOV	R2,#00	;INPUT LINE
05A2	7B00	689	MOV	R3,#00	;PRODUCT LINE
		690			
05A4	E0	691	READFUSE:	MOVX	A,@DPTR
		692			
05A5	B4301F	693	FUSE0:	CJNE	A,#0',FUSE1
05A8	1209BD	694		LCALL	SBYTE
05AB	1208F5	695		LCALL	D40USEC
		696			
05AE	0A	697	INCFUSE:	INC	R2
05AF	A3	698		INC	DPTR
05B0	BA20F1	699		CJNE	R2,#32,READFUSE
		700			
05B3	7A00	701		MOV	R2,#0
05B5	0B	702		INC	R3
05B6	1209B2	703		LCALL	SLF
05B9	BB40E8	704		CJNE	R3,#64,READFUSE
05BC	7403	705		MOV	A,#"
05BE	1209BD	706		LCALL	SBYTE
05C1	7400	707		MOV	A,#00
05C3	1209BD	708		LCALL	SBYTE
05C6	22	709		RET	
		710			
05C7	E0	711	FUSE1:	MOVX	A,@DPTR
05C8	B43138	712		CJNE	A,#'1',FUSE_EOF
05CB	BB2000	713		CJNE	R3,#32,\$+3
05CE	4019	714		JC	FUSE

```

715
05D0 C083 716          PUSH   DPH
05D2 C082 717          PUSH   DPL
05D4 7432 718          MOV     A,#2'
05D6 1209BD 719        LCALL   SBYTE
05D9 1208F5 720        LCALL   D40USEC
05DC 120641 721        LCALL   PGM32_63
05DF 1208F5 722        LCALL   D40USEC
05E2 D082 723          POP     DPL
05E4 D083 724          POP     DPH
725
05E6 0205AE 726        LJMP    INCFUSE
727
05E9 C083 728  FUSE:    PUSH   DPH
05EB C082 729          PUSH   DPL
05ED 1209BD 730        LCALL   SBYTE
05F0 1208F5 731        LCALL   D40USEC
05F3 12060C 732        LCALL   PGM0_31
05F6 120941 733        LCALL   D100USEC ;60USEC
05F9 120941 734        LCALL   D100USEC ;60USEC
735
05FC D082 736          POP     DPL
05FE D083 737          POP     DPH
738
0600 0205AE 739        LJMP    INCFUSE
740
0603 B403A8 741  FUSE_EOF: CINE   A,#",INCFUSE
0606 7403 742          MOV     A,#"
0608 1209BD 743        LCALL   SBYTE
060B 22 744          RET
745
746      ; ***** PGM0_31 SUB *****
747      ; PROGRAMMER PAL DEVICE
748      ; PRODUCT LINE 0 - 31
749      ; IN = R2 ==> LINE INPUT
750      ;   = R3 ==> PRODUCT LINE
751      ; OUT = ADDRESS FUSE PAL DEVICE = (32*R3) + R2
752      ; REG = A,
753
060C 754  PGM0_31: ; MOV     A,#10000110B ;OD = 12V

```

	755	; MOV DPTR,#PGM	;CLOCK = 0V
	756	; MOVX @DPTR,A	;VCC = 5V.
	757		
060C EA	758	MOV A,R2	
060D 900D6D	759	MOV DPTR,#IPIN5V	;OPEN TABLE LINE INPUT
0610 93	760	MOVC A,@A+DPTR	;TPIN5V
0611 90FC00	761	MOV DPTR,#PI5V	
0614 F0	762	MOVX @DPTR,A	;OUT PORT PI5V
	763		
0615 EA	764	MOV A,R2	
0616 900D8D	765	MOV DPTR,#TPIN12V	;OPEN TABLE LINE INPUT
0619 93	766	MOVC A,@A+DPTR	;TPIN12V
061A 90FC01	767	MOV DPTR,#PI12V	
061D F0	768	MOVX @DPTR,A	;OUT PORT PI12V
	769		
061E EA	770	MOV A,R2	;OPEN TABLE LINE INPUT
061F 900DAD	771	MOV DPTR,#TL_R	;L/R
0622 93	772	MOVC A,@A+DPTR	
0623 F520	773	MOV LR,A	;SAVE L/R
0625 90F800	774	MOV DPTR,#PQA	
0628 F0	775	MOVX @DPTR,A	;OUT L/R TO PQA.7
	776		
0629 748A	777	MOV A,#10001010B	;OD = 12V
062B 90F801	778	MOV DPTR,#PGM	;CLOCK = 0V
062E F0	779	MOVX @DPTR,A	;VCC = 12V
	780		
062F EB	781	MOV A,R3	;OPEN TABLE PRODUCT LINE
0630 900DED	782	MOV DPTR,#PRODUCT	
0633 93	783	MOVC A,@A+DPTR	
0634 4520	784	ORL A,LR	
0636 90F800	785	MOV DPTR,#PQA	
0639 F0	786	MOVX @DPTR,A	;OUT PORT PQA
	787		
063A 1208CF	788	LCALL D20USEC	;DELAY PGM = 20 uSEC
	789		
	790		
063D 120967	791	LCALL CLRPGM	;CLEAR PORT AFTER PROGRAM
	792		
0640 22	793	RET	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

795 ; ***** PGM32_63 SUB *****
796 ; PROGRAMMER PAL DEVICE
797 ; PRODUCT LINE 32 - 63
798 ; IN = R2 ==> LINE INPUT
799 ; = R3 ==> PRODUCT LINE
800 ; OUT = ADDRESS FUSE PAL DEVICE = (32*R3) + R2
801 ; REG = A,
802
0641 803 PGM32_63: ; MOV A,#10100100B ;OD = 12V
804 x ; MOV DPTR,#PGM ;CLOCK = 0V
805 ; MOVX @DPTR,A ;VCC = 5V
806
0641 EA 807 MOV A,R2
0642 900D6D 808 MOV DPTR,#TPIN5V ;OPEN TABLE LINE INPUT
0645 93 809 MOVC A,@A+DPTR ;TPIN5V
0646 90FC00 810 MOV DPTR,#PI5V
0649 F0 811 MOVX @DPTR,A ;OUT PORT PI5V
812
064A EA 813 MOV A,R2
064B 900D8D 814 MOV DPTR,#TPIN12V ;OPEN TABLE LINE INPUT
064E 93 815 MOVC A,@A+DPTR ;TPIN12V
064F 90FC01 816 MOV DPTR,#PI12V
0652 F0 817 MOVX @DPTR,A ;OUT PORT PI12V
818
0653 EA 819 MOV A,R2 ;OPEN TABLE LINE INPUT
0654 900DCD 820 MOV DPTR,#TL_R2 ;L/R
0657 93 821 MOVC A,@A+DPTR
0658 F520 822 MOV LR,A ;SAVE L/R
065A 90F800 823 MOV DPTR,#PQA
065D F0 824 MOVX @DPTR,A ;OUT L/R TO PQA.1
825
065E 74A8 826 MOV A,#10101000B ;OD = 12V
0660 90F801 827 MOV DPTR,#PGM ;CLOCK = 0V
0663 F0 828 MOVX @DPTR,A ;VCC = 12V
829
0664 EB 830 MOV A,R3 ;OPEN TABLE PRODUCT LINE
0665 900DED 831 MOV DPTR,#PRODUCT
0668 93 832 MOVC A,@A+DPTR
0669 4520 833 ORL A,LR
066B 90F800 834 MOV DPTR,#PQA

```

```

066E F0      835      MOVX   @DPTR,A      ;OUT PORT PQA
                        836
066F 1208CF  837      LCALL  D20USEC      ;DELAY PGM = 20 uSEC
                        838
0672 120967  839      LCALL  CLRPGM      ;CLEAR PORT AFTER PROGRAM
                        840
0675 22      841      RET
                        842
                        843      ; ***** READ_FUSE SUB *****
                        844      ; READ FUSE PAL DEVICE
                        845      ; IN  = R2,R3 ADDRESS FUSE (R3*32)+R2
                        846      ; OUT = CODE_FUSE
                        847      ; REG = A,R2,R3,DPTR
                        848
0676 900000  849      READ_FUSE: MOV    DPTR,#BUFFER
0679 7A00     850      MOV    R2,#0      ;INPUT LINE
067B 7B00     851      MOV    R3,#0      ;PRODUCT LINE
                        852
067D BB2000  853      READQ0_7: CJNE   R3,#32,$+3
0680 4006     854      JC     RF0_31
                        855
0682 120727  856      LCALL  READ32_63
0685 02068B  857      LJMP   F_Q0
                        858
0688 1206D1  859      RF0_31:  LCALL  READ0_31
                        860
068B BB0800  861      F_Q0:    CJNE   R3,#8,$+3
068E 4029     862      JC     WORK_Q0
                        863
0690 BB1000  864      F_Q1:    CJNE   R3,#16,$+3
0693 4027     865      JC     WORK_Q1
                        866
0695 BB1800  867      F_Q2:    CJNE   R3,#24,$+3
0698 4025     868      JC     WORK_Q2
                        869
069A BB2000  870      F_Q3:    CJNE   R3,#32,$+3
069D 4023     871      JC     WORK_Q3
                        872
069F BB2800  873      F_Q4:    CJNE   R3,#40,$+3
06A2 4021     874      JC     WORK_Q4

```

เอกสารนี้เป็นทรัพย์สินของสำนักงานส่งเสริมการค้าในต่างประเทศ ณ นครเชียงใหม่ ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

875
06A4 BB3000 876 F_Q5: CJNE R3,#48,$+3
06A7 401F 877 JC WORK_Q5
878
06A9 BB3800 879 F_Q6: CJNE R3,#56,$+3
06AC 401D 880 JC WORK_Q6
881
06AE BB4000 882 F_Q7: CJNE R3,#64,$+3
06B1 401B 883 JC WORK_Q7
884
06B3 742D 885 MOV A,#'
06B5 1209BD 886 LCALL SBYTE
06B8 22 887 RET
888
06B9 02077D 889 WORK_Q0: LJMP FUSE_Q0
06BC 02078A 890 WORK_Q1: LJMP FUSE_Q1
06BF 020797 891 WORK_Q2: LJMP FUSE_Q2
06C2 0207A4 892 WORK_Q3: LJMP FUSE_Q3
06C5 0207B1 893 WORK_Q4: LJMP FUSE_Q4
06C8 0207BE 894 WORK_Q5: LJMP FUSE_Q5
06CB 0207CB 895 WORK_Q6: LJMP FUSE_Q6
06CE 0207D8 896 WORK_Q7: LJMP FUSE_Q7
897
898 ; ***** READ0_31 SUB *****
899 ; READ FUSE PAL DEVICE
900 ; IN = R2,R3 ADDRESS FUSE (R3*32)+R2
901 ; OUT = CODE_FUSE
902 ; REG = A,R2,R3,DPTR
903
06D1 C083 904 READ0_31: PUSH DPH
06D3 C082 905 PUSH DPL
906
06D5 7496 907 MOV A,#10010110B ;VCC = 5 V
06D7 90F801 908 MOV DPTR,#PGM ;CLOCK = 5 V
06DA F0 909 MOVX @DPTR,A ;OD = 12 V
910
06DB EA 911 MOV A,R2 ;OPEN
06DC 900D6D 912 MOV DPTR,#TPIN5V ;TABLE
06DF 93 913 MOVC A,@A+DPTR ;INPUT LINE
06E0 90FC00 914 MOV DPTR,#PI5V ;OUT PORT PI5V

```

06E3 F0	915	MOVX	@DPTR,A	
	916			
06E4 EA	917	MOV	A,R2	;OPEN
06E5 900D8D	918	MOV	DPTR,#TPIN12V	;TABLE
06E8 93	919	MOVC	A,@A+DPTR	;INPUT LINE
06E9 90FC01	920	MOV	DPTR,#PI12V	;OUT PORT PI12V
06EC F0	921	MOVX	@DPTR,A	
	922			
06ED EA	923	MOV	A,R2	;OPEN TABLE INPUT LINE
06EE 900DAD	924	MOV	DPTR,#TL_R	;L/R
06F1 93	925	MOVC	A,@A+DPTR	
06F2 F520	926	MOV	LR,A	;SAVE L/R
06F4 90F800	927	MOV	DPTR,#PQA	
06F7 F0	928	MOVX	@DPTR,A	;OUT PORT L/R
	929			
06F8 EB	930	MOV	A,R3	;OPEN
06F9 900E2D	931	MOV	DPTR,#RPRODUCT	;TABLE
06FC 93	932	MOVC	A,@A+DPTR	;PRODUCT
06FD 4520	933	ORL	A,LR	
06FF 90F800	934	MOV	DPTR,#PQA	;OUT PORT PRODUCT
0702 F0	935	MOVX	@DPTR,A	
	936			
	937	; LCALL	D100USEC	
0703 120908	938	LCALL	D50USEC	
	939			
0706 7486	940	MOV	A,#10000110B	;OD = 12V
0708 90F801	941	MOV	DPTR,#PGM	;CLOCK = 0V
070B F0	942	MOVX	@DPTR,A	;VCC = 5V
	943			
070C 7400	944	MOV	A,#00H	
070E 90F802	945	MOV	DPTR,#P_CON	
0711 F0	946	MOVX	@DPTR,A	
	947			
	948	; LCALL	D100USEC	
0712 120908	949	LCALL	D50USEC	
	950			
0715 90FC02	951	MOV	DPTR,#PDATA	
0718 7F0A	952	MOV	R7,#10	
071A E0	953	MOVX	A,@DPTR	
	954	DJNZ	R7,S-1	

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

071D F525      955      MOV      CODEFUSE,A
071F 120967    956      LCALL   CLRPGM
                957
0722 D082      958      POP      DPL
0724 D083      959      POP      DPH
0726 22        960      RET
                961
                962      ; ***** READ32_63 SUB *****
                963      ; READ FUSE PAL DEVICE
                964      ; IN  = R2,R3 ADDRESS FUSE (R3*32)+R2
                965      ; OUT = CODE_FUSE
                966      ; REG = A,R2,R3,DPTR
                967
0727 C083      968      READ32_63: PUSH   DPH
0729 C082      969      PUSH   DPL
                970
072B 74A5      971      MOV     A,#10100101B ;VCC = 5 V
072D 90F801    972      MOV     DPTR,#PGM ;CLOCK = 5 V
0730 F0        973      MOVX    @DPTR,A ;OD = 12 V
                974
0731 EA        975      MOV     A,R2 ;OPEN
0732 900D6D    976      MOV     DPTR,#TPIN5V ;TABLE
0735 93        977      MOVC    A,@A+DPTR ;INPUT LINE
0736 90FC00    978      MOV     DPTR,#PI5V ;OUT PORT PI5V
0739 F0        979      MOVX    @DPTR,A
                980
073A EA        981      MOV     A,R2 ;OPEN
073B 900D8D    982      MOV     DPTR,#TPIN12V ;TABLE
073E 93        983      MOVC    A,@A+DPTR ;INPUT LINE
073F 90FC01    984      MOV     DPTR,#PI12V ;OUT PORT PI12V
0742 F0        985      MOVX    @DPTR,A
                986
0743 EA        987      MOV     A,R2 ;OPEN TABLE INPUT LINE
0744 900DAD    988      MOV     DPTR,#TL_R ;L/R
0747 93        989      MOVC    A,@A+DPTR
0748 F520      990      MOV     LR,A ;SAVE L/R
074A 90F800    991      MOV     DPTR,#PQA
074D F0        992      MOVX    @DPTR,A ;OUT PORT L/R
                993
074E EB        994      MOV     A,R3 ;OPEN

```

```

074F 900E2D      995          MOV    DPTR,#RPRODUCT ;TABLE
0752 93          996          MOVC   A,@A+DPTR      ;PRODUCT
0753 4520        997          ORL    A,LR
0755 90F800      998          MOV    DPTR,#PQA        ;OUT PORT  PRODUCT
0758 F0          999          MOVX   @DPTR,A
                                1000
                                1001          ; LCALL  D100USEC
0759 120908      1002          LCALL  D50USEC
                                1003
075C 74A4        1004          MOV    A,#10100100B      ;OD   = 12V
075E 90F801      1005          MOV    DPTR,#PGM        ;CLOCK = 0V
0761 F0          1006          MOVX   @DPTR,A        ;VCC  = 5V
                                1007
0762 7400        1008          MOV    A,#00H
0764 90F802      1009          MOV    DPTR,#P_CON
0767 F0          1010          MOVX   @DPTR,A
                                1011
                                1012          ; LCALL  D100USEC
0768 120908      1013          LCALL  D50USEC
                                1014
076B 90FC02      1015          MOV    DPTR,#PDATA
076E 7F0A        1016          MOV    R7,#10
0770 E0          1017          MOVX   A,@DPTR
0771 DFFD        1018          DJNZ   R7,$-1
0773 F525        1019          MOV    CODEFUSE,A
0775 120967      1020          LCALL  CLRPGM
                                1021
0778 D082        1022          POP    DPL
077A D083        1023          POP    DPH
077C 22          1024          RET
                                1025
                                1026          ; ***** FUSE_Q0 SUB *****
                                1027          ; READ FUSE PAL DEVICE
                                1028          ; IN  = CODEFUSE
                                1029          ; OUT = A
                                1030          ; REG = CODEFUSE,A
                                1031
077D E525        1032          FUSE_Q0:  MOV    A,CODEFUSE
077F 5401        1033          ANL    A,#01H

```

```

0781 B40103      1035      CJNE    A,#01H,$+6
0784 0207E5      1036      LJMP     SAVE_FUSE0      ;READ O/P = 1 IN JEDEC = 0
0787 0207EE      1037      LJMP     SAVE_FUSE1      ;READ O/P = 0 IN JEDEC = 1
                  1038
                  1039      ; ***** FUSE_Q1 SUB *****
                  1040      ; READ FUSE PAL DEVICE
                  1041      ; IN = CODEFUSE
                  1042      ; OUT = A
                  1043      ; REG = CODEFUSE,A
                  1044
078A E525         1045      FUSE_Q1:  MOV     A,CODEFUSE
078C 5402         1046      ANL      A,#02H
                  1047
                  1048      CJNE    A,#02H,$+6
078E B40203      1049      LJMP     SAVE_FUSE0      ;READ O/P = 1 IN JEDEC = 0
0791 0207E5      1050      LJMP     SAVE_FUSE1      ;READ O/P = 0 IN JEDEC = 1
0794 0207EE      1051
                  1052      ; ***** FUSE_Q2 SUB *****
                  1053      ; READ FUSE PAL DEVICE
                  1054      ; IN = CODEFUSE
                  1055      ; OUT = A
                  1056      ; REG = CODEFUSE,A
                  1057
0797 E525         1058      FUSE_Q2:  MOV     A,CODEFUSE
0799 5404         1059      ANL      A,#04H
                  1060
                  1061      CJNE    A,#04H,$+6
079B B40403      1062      LJMP     SAVE_FUSE0      ;READ O/P = 1 IN JEDEC = 0
079E 0207E5      1063      LJMP     SAVE_FUSE1      ;READ O/P = 0 IN JEDEC = 1
07A1 0207EE      1064
                  1065      ; ***** FUSE_Q3 SUB *****
                  1066      ; READ FUSE PAL DEVICE
                  1067      ; IN = CODEFUSE
                  1068      ; OUT = A
                  1069      ; REG = CODEFUSE,A
                  1070
07A4 E525         1071      FUSE_Q3:  MOV     A,CODEFUSE
07A6 5408         1072      ANL      A,#08H
                  1073

```

```

07AB 0207E5      1075      L JMP    SAVE_FUSE0      ;READ O/P = 1 IN JEDEC = 0
07AE 0207EE      1076      L JMP    SAVE_FUSE1      ;READ O/P = 0 IN JEDEC = 1
1077
1078      ; ***** FUSE_Q4 SUB *****
1079      ; READ FUSE PAL DEVICE
1080      ; IN = CODEFUSE
1081      ; OUT = A
1082      ; REG = CODEFUSE,A
1083
07B1 E525      1084      FUSE_Q4:  MOV    A,CODEFUSE
07B3 5410      1085      ANL    A,#10H
1086
1087      CJNE    A,#10H,$+6
07B5 B41003      1088      L JMP    SAVE_FUSE0      ;READ O/P = 1 IN JEDEC = 0
07B8 0207E5      1089      L JMP    SAVE_FUSE1      ;READ O/P = 0 IN JEDEC = 1
07BB 0207EE      1090
1091      ; ***** FUSE_Q5 SUB *****
1092      ; READ FUSE PAL DEVICE
1093      ; IN = CODEFUSE
1094      ; OUT = A
1095      ; REG = CODEFUSE,A
1096
07BE E525      1097      FUSE_Q5:  MOV    A,CODEFUSE
07C0 5420      1098      ANL    A,#20H
1099
1100      CJNE    A,#20H,$+6
07C2 B42003      1101      L JMP    SAVE_FUSE0      ;READ O/P = 1 IN JEDEC = 0
07C5 0207E5      1102      L JMP    SAVE_FUSE1      ;READ O/P = 0 IN JEDEC = 1
07C8 0207EE      1103
1104      ; ***** FUSE_Q6 SUB *****
1105      ; READ FUSE PAL DEVICE
1106      ; IN = CODEFUSE
1107      ; OUT = A
1108      ; REG = CODEFUSE,A
1109
07CB E525      1110      FUSE_Q6:  MOV    A,CODEFUSE
07CD 5440      1111      ANL    A,#40H
1112
1113      CJNE    A,#40H,$+6
07CF B44003      1114      L JMP    SAVE_FUSE0      ;READ O/P = 1 IN JEDEC = 0
07D2 0207E5

```

```

07D5 0207EE      1115          LJMP    SAVE_FUSE1      ;READ O/P = 0 IN JEDEC = 1
                  1116
                  1117      ; ***** FUSE_Q7 SUB *****
                  1118      ; READ FUSE PAL DEVICE
                  1119      ; IN = CODEFUSE
                  1120      ; OUT = A
                  1121      ; REG = CODEFUSE,A
                  1122
07D8 E525        1123      FUSE_Q7:  MOV    A,CODEFUSE
07DA 5480         1124          ANL    A,#80H
                  1125
07DC B48003      1126          CJNE   A,#80H,$+6
07DF 0207E5      1127          LJMP    SAVE_FUSE0      ;READ O/P = 1 IN JEDEC = 0
07E2 0207EE      1128          LJMP    SAVE_FUSE1      ;READ O/P = 0 IN JEDEC = 1
                  1129
                  1130      ; ***** SAVE_FUSE0 SUB *****
                  1131      ; SAVE FUSE FROM READ PAL DEVICE
                  1132      ; IN = A,R2,R3
                  1133      ; OUT = SAVE BUFFER,R2+1,R3+1,DPTR+1
                  1134      ; REG = A,R2,R3,DDPTR
                  1135
07E5 7430        1136      SAVE_FUSE0: MOV    A,#0'
07E7 F0          1137          MOVX   @DPTR,A
                  1138
07E8 1209BD      1139          LCALL   SBYTE
                  1140      ; LCALL   D10USEC
07EB 0207F7      1141          LJMP    EXIT_SAVE
                  1142
                  1143      ; ***** SAVE_FUSE1 SUB *****
                  1144      ; SAVE FUSE FROM READ PAL DEVICE
                  1145      ; IN = A,R2,R3
                  1146      ; OUT = SAVE BUFFER,R2+1,R3+1,DPTR+1
                  1147      ; REG = A,R2,R3,DDPTR
                  1148
07EE 7431        1149      SAVE_FUSE1: MOV    A,#1'
07F0 F0          1150          MOVX   @DPTR,A
                  1151
07F1 1209BD      1152          LCALL   SBYTE
                  1153      ; LCALL   D20USEC
07F4 0207F7      1154          LJMP    EXIT_SAVE

```

```

1155
07F7 A3      1156  EXIT_SAVE: INC    DPTR
07F8 0A      1157                INC    R2
07F9 BA2017  1158                CJNE   R2,#32,END_SAVE
07FC 1208CF  1159                LCALL  D20USEC
07FF 1209B2  1160                LCALL  SLF
0802 7A00    1161                MOV    R2,#0
0804 0B      1162                INC    R3
0805 BB400B  1163                CJNE   R3,#64,END_SAVE ;#64
0808 7403    1164                MOV    A,#"
080A 1209BD  1165                LCALL  SBYTE
080D 7400    1166                MOV    A,#00
080F 1209BD  1167                LCALL  SBYTE
0812 22      1168                RET
1169
0813 02067D  1170  END_SAVE: LJMP   READQ0_7
1171
1172 ; ***** BLANKCHECK SUB *****
1173 ; BLANKCHECK PAL DEVICE = 0
1174 ; IN  = R2,R3 ADDRESS FUSE (R3*32)+R2
1175 ; OUT = CODE_FUSE
1176 ; REG = A,R2,R3,DPTR
1177
1178 ; *****
1179
0816 1209B2  1180  DUMPRAM: LCALL  SLF
0819 908000  1181                MOV    DPTR,#BUFFER2
1182
081C E0      1183  PP:      MOVX   A,@DPTR
081D B40313  1184                CJNE   A,#",S
0820 1209BD  1185                LCALL  SBYTE
0823 1209B2  1186                LCALL  SLF
0826 7A3C    1187                MOV    R2,#60
0828 120896  1188                LCALL  DMSEC
082B 742D    1189                MOV    A,#"
082D 1209BD  1190                LCALL  SBYTE
0830 02018B  1191                LJMP   WAIT
1192
0833 1209BD  1193  S:      LCALL  SBYTE
1194                LCALL  D60USEC

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0836 A3	1195		INC	DPTR	
0837 02081C	1196		LJMP	PP	
	1197				
083A 1209B2	1198	DUMP:	LCALL	SLF	
083D 900000	1199		MOV	DPTR,#BUFFER	
	1200				
0840 E0	1201	PP2:	MOVX	A,@DPTR	
0841 B40313	1202		CJNE	A,#",S2	
0844 1209BD	1203		LCALL	SBYTE	
0847 1209B2	1204		LCALL	SLF	
084A 7A14	1205		MOV	R2,#20	
084C 12091B	1206		LCALL	D60USEC	
084F 742D	1207		MOV	A,#-'	
0851 1209BD	1208		LCALL	SBYTE	
0854 02018B	1209		LJMP	WAIT	
	1210				
0857 1209BD	1211	S2:	LCALL	SBYTE	
085A A3	1212		INC	DPTR	
085B 020840	1213		LJMP	PP2	
	1214				
	1215				
	1216		; ***** SUB ROUTINE *****		
	1217		; ***** CLRRAM SUB *****		
	1218		; IN = NO REG		
	1219		; OUT = RAM BUFFER		
	1220		; REG = A,R2,R3,DPTR		
	1221				
085E 900000	1222	CLRRAM:	MOV	DPTR,#BUFFER	;CLEAR PROCESS
	1223				
0861 7A17	1224		MOV	R2,#17H	
0863 7BFE	1225	CLRRAM01:	MOV	R3,#0FEH	
	1226				
0865 7430	1227	CLRRAM02:	MOV	A,#30H	; A = '0'
0867 F0	1228		MOVX	@DPTR,A	
0868 120886	1229		LCALL	DPCOM	
086B A3	1230		INC	DPTR	
086C 70F7	1231		JNZ	CLRRAM02	
086E 7403	1232		MOV	A,#"	
0870 F0	1233		MOVX	@DPTR,A	
0871 22	1234		RET		

```

1235
1236 ; ***** CLRRAM2 SUB *****
1237 ; IN = NO REG
1238 ; OUT = RAM BUFFER
1239 ; REG = A,R2,R3,DPTR
1240
0872 908000 1241 CLRRAM2: MOV DPTR,#BUFFER2 ;CLEAR PROCESS
1242
0875 7A87 1243 MOV R2,#87H
0877 7BFE 1244 CLRRAM21: MOV R3,#0FEH
1245
0879 7430 1246 CLRRAM22: MOV A,#30H ; A = '0'
087B F0 1247 MOVX @DPTR,A
087C 120886 1248 LCALL DPCOM
087F A3 1249 INC DPTR
0880 70F7 1250 JNZ CLRRAM22
0882 7403 1251 MOV A,#"
0884 F0 1252 MOVX @DPTR,A
0885 22 1253 RET
1254
1255 ; ***** DPCOM SUM *****
1256 ; COMPARE WORD (16 BIT) DPTR <> R2,R3
1257 ; IN = DPTR,R2,R3
1258 ; OUT = CY,A
1259 ; REG = A
1260
0886 C082 1261 DPCOM: PUSH DPL
0888 C3 1262 CLR C
0889 E582 1263 MOV A,DPL
088B 9B 1264 SUBB A,R3
088C F582 1265 MOV DPL,A
088E E583 1266 MOV A,DPH
0890 9A 1267 SUBB A,R2
0891 4582 1268 ORL A,DPL
0893 D082 1269 POP DPL
0895 22 1270 RET
1271
1272 ; ***** DMSEC SUB *****

```

เอกสารนี้เป็นเอกสารที่สงวนไว้ 1273 รับกร; DELAY 1/1000 SECOND เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	1275	; RESGISTOR R2,R3		
	1276			
0896 7BE6	1277	DMSEC:	MOV	R3,#230
0898 00	1278	DMSEC1:	NOP	
0899 00	1279		NOP	
089A DBFC	1280		DJNZ	R3,DMSEC1
089C DAF8	1281		DJNZ	R2,DMSEC
089E 22	1282		RET	
	1283			
	1284	; ***** DTSEC SUB *****		
	1285	; DELAY 1/10 SECOND		
	1286	; IN = R2		
	1287	; REG = R2,R3,R4		
	1288			
089F 7BB3	1289	DTSEC:	MOV	R3,#179
08A1 7C00	1290	DTSEC1:	MOV	R4,#0
08A3 DCFE	1291		DJNZ	R4,\$
08A5 00	1292		NOP	
08A6 00	1293		NOP	
08A7 DBF8	1294		DJNZ	R3,DTSEC1
08A9 DAF4	1295		DJNZ	R2,DTSEC
08AB 22	1296		RET	
	1297			
	1298	; ***** DSEC SUB *****		
	1299	; DELAY SECOND		
	1300	; IN = R1		
	1301	; REG = R1,R2,R3,R4		
	1302			
08AC 7A0A	1303	DSEC:	MOV	R2,#10
08AE 12089F	1304		LCALL	DTSEC
08B1 D9F9	1305		DJNZ	R1,DSEC
08B3 22	1306		RET	
	1307			
	1308	; ***** DMIN SUB *****		
	1309	; DELAY MINUTE		
	1310	; IN = R5		
	1311	; REG = R1,R2,R3,R4,R5		
	1312			
08B4 793C	1313	DMIN:	MOV	R1,#60

เอกสารนี้เป็นเอกสารต้นฉบับที่จัดทำขึ้นเพื่อใช้ในการปฏิบัติงานเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

08B9	DDF9	1315	DJNZ	R5,DMIN	
08BB	22	1316	RET		
		1317			
		1318	; ***** 10USEC SUB *****		
		1319	; DELAY 10 uSEC		
		1320	; NO REG		
		1321	; REG TMOD TH0 TL0		
		1322			
08BC	758921	1323	D10USEC:	MOV	TMOD,#21H
08BF	758CFF	1324		MOV	TH0,#0FFH
08C2	758AF7	1325		MOV	TL0,#0F7H ;F7
08C5	D28C	1326		SETB	TR0
08C7	308DFD	1327		JNB	TF0,\$
08CA	C28C	1328		CLR	TR0
08CC	C28D	1329		CLR	TF0
08CE	22	1330		RET	
		1331			
		1332	; ***** 20USEC SUB *****		
		1333	; DELAY 20 uSEC		
		1334	; NO REG		
		1335	; REG TMOD TH0 TL0		
		1336			
08CF	758921	1337	D20USEC:	MOV	TMOD,#21H
08D2	758CFF	1338		MOV	TH0,#0FFH
08D5	758AEE	1339		MOV	TL0,#0EEH ;EEH
08D8	D28C	1340		SETB	TR0
08DA	308DFD	1341		JNB	TF0,\$
08DD	C28C	1342		CLR	TR0
08DF	C28D	1343		CLR	TF0
08E1	22	1344		RET	
		1345			
		1346	; ***** 30USEC SUB *****		
		1347	; DELAY 30 uSEC		
		1348	; NO REG		
		1349	; REG TMOD TH0 TL0		
		1350			
08E2	758921	1351	D30USEC:	MOV	TMOD,#21H
08E5	758CFF	1352		MOV	TH0,#0FFH
08E8	758AE5	1353		MOV	TL0,#0E5H ;0E5H
08EB	D28C	1354		SETB	TR0

```

08ED 308DFD      1355      JNB      TF0,$
08F0 C28C        1356      CLR      TR0
08F2 C28D        1357      CLR      TF0
08F4 22          1358      RET
                  1359
                  1360      ; ***** 40USEC SUB *****
                  1361      ; DELAY 40 uSEC
                  1362      ; NO REG
                  1363      ; REG TMOD TH0 TL0
                  1364
08F5 758921      1365      D40USEC:  MOV     TMOD,#21H
08F8 758CFF      1366      MOV     TH0,#0FFH
08FB 758ADC      1367      MOV     TL0,#0DCH      ;0DCH
08FE D28C        1368      SETB    TR0
0900 308DFD      1369      JNB      TF0,$
0903 C28C        1370      CLR      TR0
0905 C28D        1371      CLR      TF0
0907 22          1372      RET
                  1373
                  1374      ; ***** 50USEC SUB *****
                  1375      ; DELAY 50 uSEC
                  1376      ; NO REG
                  1377      ; REG TMOD TH0 TL0
                  1378
0908 758921      1379      D50USEC:  MOV     TMOD,#21H
090B 758CFF      1380      MOV     TH0,#0FFH
090E 758AD2      1381      MOV     TL0,#0D2H      ;0D2H
0911 D28C        1382      SETB    TR0
0913 308DFD      1383      JNB      TF0,$
0916 C28C        1384      CLR      TR0
0918 C28D        1385      CLR      TF0
091A 22          1386      RET
                  1387
                  1388      ; ***** 60USEC SUB *****
                  1389      ; DELAY 60 uSEC
                  1390      ; NO REG
                  1391      ; REG TMOD TH0 TL0
                  1392

```

091B 758921 1393 D60USEC: MOV TMOD,#21H

091E 758CFF 1394 MOV TH0,#0FFH

0921	758AC9	1395	MOV	TL0,#0C9H	;C9H
0924	D28C	1396	SETB	TR0	
0926	308DFD	1397	JNB	TF0,\$	
0929	C28C	1398	CLR	TR0	
092B	C28D	1399	CLR	TF0	
092D	22	1400	RET		

1401

1402 ; ***** 80USEC SUB *****

1403 ; DELAY 80 uSEC

1404 ; NO REG

1405 ; REG TMOD TH0 TL0

1406

092E	758921	1407	D80USEC:	MOV	TMOD,#21H
0931	758CFF	1408		MOV	TH0,#0FFH
0934	758AB7	1409		MOV	TL0,#0B7H ;0B7H
0937	D28C	1410		SETB	TR0
0939	308DFD	1411		JNB	TF0,\$
093C	C28C	1412		CLR	TR0
093E	C28D	1413		CLR	TF0
0940	22	1414		RET	

1415

1416 ; ***** 100USEC SUB *****

1417 ; DELAY 100 uSEC

1418 ; NO REG

1419 ; REG TMOD TH0 TL0

1420

0941	758921	1421	D100USEC:	MOV	TMOD,#21H
0944	758CFF	1422		MOV	TH0,#0FFH
0947	758AA4	1423		MOV	TL0,#0A4H
094A	D28C	1424		SETB	TR0
094C	308DFD	1425		JNB	TF0,\$
094F	C28C	1426		CLR	TR0
0951	C28D	1427		CLR	TF0
0953	22	1428		RET	

1429

1430

1431 ; ***** 500USEC SUB *****

1432 ; DELAY 500 uSEC

1433 ; NO REG

1434 ; REG TMOD TH0 TL0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับกรใช้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0954 758921	1435	D500USEC:	MOV	TMOD,#21H
0957 758CFE	1436		MOV	TH0,#0FEH
095A 758A34	1437		MOV	TL0,#34H ;34H
095D D28C	1438		SETB	TR0
095F 308DFD	1439		JNB	TF0,\$
0962 C28C	1440		CLR	TR0
0964 C28D	1441		CLR	TF0
0966 22	1442		RET	
	1443			
	1444			
	1445	; ***** CLRPGM SUM *****		
	1446	; CLEAR PORT AFTER PROGRAM DEVICE		
	1447	; IN NO REG		
	1448	; REG A,DPTR		
	1449			
0967 7480	1450	CLRPGM:	MOV	A,#80H
0969 90F802	1451		MOV	DPTR,#P_CON
096C F0	1452		MOVX	@DPTR,A
	1453			
096D 74C0	1454		MOV	A,#0C0H
096F 90F801	1455		MOV	DPTR,#PGM
0972 F0	1456		MOVX	@DPTR,A
	1457			
0973 E4	1458		CLR	A
0974 90F800	1459		MOV	DPTR,#PQA
0977 F0	1460		MOVX	@DPTR,A
	1461			
0978 90FC00	1462		MOV	DPTR,#PI5V
097B F0	1463		MOVX	@DPTR,A
	1464			
097C 90FC01	1465		MOV	DPTR,#PI12V
097F F0	1466		MOVX	@DPTR,A
0980 22	1467		RET	
	1468			
	1469	; ***** SBLOCK SUB *****		
	1470	; SEND BLOCK		
	1471	; IN = DPTR ROM-ADDRESS (END BY 0 OR 0DH)		
	1472	; OUT = DPTR (NEXT)		
	1473	; REG = A,DPTR		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

0981 E4      1475  SBLOCK:  CLR A
0982 93      1476           MOV  A,@A+DPTR
0983 B40002   1477           CJNE  A,#0,SBLOCK1
0986 A3      1478           INC   DPTR           ;NEXT ADDRESS
0987 22      1479           RET             ;EXIT BY 0
0988 B40D05   1480  SBLOCK1: CJNE  A,#0DH,SBLOCK2
098B 1209B2   1481           LCALL  SLF           ;CR/LF
098E A3      1482           INC   DPTR           ;NEXT ADDRESS
098F 22      1483           RET             ;EXIT BY 0DH
0990 1209BD   1484  SBLOCK2: LCALL  SBYTE
0993 A3      1485           INC   DPTR
0994 80EB     1486           SJMP  SBLOCK
1487
1488 ; ***** SBLANK SUB *****
1489 ; SEND BLANK
1490 ; IN = R2 (NO OF BLANK)
1491 ; REG = A,R2
1492
0996 7420     1493  SBLANK:  MOV   A,#'
0998 1209BD   1494           LCALL  SBYTE
099B DAF9     1495           DJNZ  R2,SBLANK
099D 22      1496           RET
1497
1498 ; ***** SPACE SUB *****
1499 ; PRINT CR/LF = R1
1500 ; IN = NO REG
1501 ; OUT = TO MONITOR
1502 ; REG = R1,DPTR
1503
099E C083     1504  SPACE:  PUSH  DPH
09A0 C082     1505           PUSH  DPL
09A2 C001     1506           PUSH  01
09A4 7912     1507           MOV   R1,#18
09A6 1209B2   1508  F:      LCALL  SLF
09A9 D9FB     1509           DJNZ  R1,F
09AB D001     1510           POP   01
09AD D082     1511           POP   DPL
09AF D083     1512           POP   DPH
09B1 22      1513           RET
1514

```

```

1515 ; ***** SLF SUB *****
1516 ; PRINT CR/LF
1517 ; REG A
1518
09B2 740D 1519 SLF:      MOV    A,#0DH
09B4 1209BD 1520          LCALL   SBYTE
09B7 740A 1521          MOV    A,#0AH
09B9 1209BD 1522          LCALL   SBYTE
09BC 22 1523          RET
1524
1525
1526 ; ***** SBYTE SUB *****
1527 ; SEND BYTE
1528 ; IN = A
1529 ; REG = NO
1530
09BD 3099FD 1531 SBYTE:  JNB     TI,$
09C0 C299 1532          CLR     TI
09C2 F599 1533          MOV     SBUF,A
09C4 F590 1534          MOV     PI,A
09C6 22 1535          RET
1536
1537 ; ***** SBYTEH SUB *****
1538 ; SEND 2 BYTE HEX FROM A
1539 ; IN = A
1540 ; REG = A,R2,R3
1541
09C7 120A11 1542 SBYTEH: LCALL   HTOA
09CA EA 1543          MOV     A,R2
09CB 1209BD 1544          LCALL   SBYTE
09CE EB 1545          MOV     A,R3
09CF 1209BD 1546          LCALL   SBYTE
09D2 22 1547          RET
1548
1549 ; ***** SBYTEHC SUB *****
1550 ; SEND 2 BYTE HEX FROM A AND CHECK SUM
1551 ; IN = A
1552 ; REG = A,R2,R3
1553
09D3 120A11 1554 SBYTEHC: LCALL   HTOA

```

09D6 EA	1555	MOV	A,R2
09D7 1209BD	1556	LCALL	SBYTE
	1557		
09DA 1209E5	1558	LCALL	RBYTE
	1559		
09DD EB	1560	MOV	A,R3
09DE 1209BD	1561	LCALL	SBYTE
	1562		
09E1 1209E5	1563	LCALL	RBYTE
09E4 22	1564	RET	
	1565		
	1566	; ***** RBYTE SUB *****	
	1567	; OUT = A	
	1568	; REG = A	
	1569		
09E5 3098FD	1570	RBYTE:	JNB RI,\$
09E8 C298	1571	CLR	RI
09EA E599	1572	MOV	A,SBUF
09EC F590	1573	MOV	P1,A
09EE 22	1574	RET	
	1575		
	1576	; ***** RBYTEH SUB *****	
	1577	; READ HEX-ASCII FROM SERIAL (2 BYTE)	
	1578	; IN = SERIAL PORT RS 232C	
	1579	; OUT = A	
	1580	; REG = A,R2,R3	
	1581		
09EF 1209E5	1582	RBYTEH:	LCALL RBYTE
09F2 FA	1583	MOV	R2,A
09F3 1209E5	1584	LCALL	RBYTE
09F6 FB	1585	MOV	R3,A
09F7 1209FB	1586	LCALL	ATOH
09FA 22	1587	RET	
	1588		
	1589	; ***** ATOH SUB *****	
	1590	; ASCII TO HEX CONVERT	
	1591	; IN = R2,R3 30H,41H	
	1592	; OUT = A 0AH	
	1593	; REG = A,R2	

09FB EA	1595	ATOH :	MOV	A,R2
09FC 120A07	1596		LCALL	ATOHS
09FF C4	1597		SWAP	A
0A00 FA	1598		MOV	R2,A
0A01 EB	1599		MOV	A,R3
0A02 120A07	1600		LCALL	ATOHS
0A05 4A	1601		ORL	A,R2
0A06 22	1602		RET	
	1603			
0A07 B44100	1604	ATOHS:	CJNE	A,#A',\$,+3
0A0A 4002	1605		JC	ATOHS1
0A0C 2409	1606		ADD	A,#9
0A0E 540F	1607	ATOHS1:	ANL	A,#0FH
0A10 22	1608		RET	
	1609			
	1610		; ***** HTOA SUB *****	
	1611		; CONVERT HEX TO ASCII	
	1612		; IN = A	
	1613		; OUT = R2,R3	
	1614		; REG = A,R2,R3	
	1615			
0A11 C0E0	1616	HTOA:	PUSH	ACC
0A13 C4	1617		SWAP	A
0A14 120A1F	1618		LCALL	HTOAS
0A17 FA	1619		MOV	R2,A
0A18 D0E0	1620		POP	ACC
0A1A 120A1F	1621		LCALL	HTOAS
0A1D FB	1622		MOV	R3,A
0A1E 22	1623		RET	
	1624			
0A1F 540F	1625	HTOAS:	ANL	A,#0FH
0A21 B40A00	1626		CJNE	A,#0AH,\$+3
0A24 5003	1627		JNC	HTOAS1
0A26 4430	1628		ORL	A,#30H
0A28 22	1629		RET	
	1630			
0A29 9409	1631	HTOAS1:	SUBB	A,#9
0A2B 4440	1632		ORL	A,#40H
0A2D 22	1633		RET	

```

1635 ; ***** DTOH SUB *****
1636 ; DECIMAL TO HEX
1637 ; IN = R1,R2,R3
1638 ; OUT = DPTR
1639 ; REG = A,R0,R1,R2,R3,R4,R5,DPTR
1640
0A2E 7C10 1641 DTOH: MOV R4,#16
0A30 7D03 1642 DTOH1: MOV R5,#3
0A32 7801 1643 MOV R0,#1
0A34 C3 1644 CLR C
0A35 E6 1645 DTOH2: MOV A,@R0
0A36 13 1646 RRC A
0A37 C0D0 1647 PUSH PSW
0A39 30E703 1648 JNB ACC.7,DTOH3
0A3C C3 1649 CLR C
0A3D 9430 1650 SUBB A,#30H
1651
0A3F 30E303 1652 DTOH3: JNB ACC.3,DTOH4
0A42 C3 1653 CLR C
0A43 9403 1654 SUBB A,#03H
1655
0A45 F6 1656 DTOH4: MOV @R0,A
0A46 08 1657 INC R0
0A47 D0D0 1658 POP PSW
0A49 DDEA 1659 DJNZ R5,DTOH2
1660
0A4B E583 1661 MOV A,DPH
0A4D 13 1662 RRC A
0A4E F583 1663 MOV DPH,A
0A50 E582 1664 MOV A,DPL
0A52 13 1665 RRC A
0A53 F582 1666 MOV DPL,A
0A55 DCD9 1667 DJNZ R4,DTOH1
0A57 22 1668 RET
1669
1670 ; ***** HTOD SUB *****
1671 ; HEX TO DECIMAL
1672 ; IN = DPTR
1673 ; OUT = R1,R2,R3

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ไว้ 1674 หรือหาก REG = A,R0,R1,R2,R3,R4,R5,DPTR ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1675
0A58 E4      1676      HTOD:      CLR      A              ;CLEAR OUTPUT
0A59 F9      1677              MOV      R1,A
0A5A FA      1678              MOV      R2,A
0A5B FB      1679              MOV      R3,A
1680
0A5C 7C10    1681              MOV      R4,#16          ;SHIFT 16 BIT
0A5E E582    1682      HTOD1:      MOV      A,DPL
0A60 33      1683              RLC      A
0A61 F582    1684              MOV      DPL,A
0A63 E583    1685              MOV      A,DPH
0A65 33      1686              RLC      A
0A66 F583    1687              MOV      DPH,A
1688
0A68 7D03    1689              MOV      R5,#3          ;ADD DECIMAL
0A6A 7803    1690              MOV      R0,#3          ;INDEX TO R3
0A6C E6      1691      HTOD2:      MOV      A,@R0
0A6D 35E0    1692              ADDC     A,ACC
0A6F D4      1693              DA      A
0A70 F6      1694              MOV      @R0,A
0A71 18      1695              DEC      R0
0A72 DDF8    1696              DJNZ     R5,HTOD2
0A74 DCE8    1697              DJNZ     R4,HTOD1
0A76 22      1698              RET
1699
1700      ; ***** LCDWI SUB *****
1701      ; LCD WRITE INSTRUCTION (RS=0)
1702      ; IN = A
1703      ; REG = A,DPTR
1704
0A77 C083    1705      LCDWI:      PUSH     DPH
0A79 C082    1706              PUSH     DPL
0A7B 90FA00  1707              MOV      DPTR,#LCDWRC
0A7E F0      1708              MOVX     @DPTR,A
0A7F 90FA01  1709              MOV      DPTR,#LCDRDC
0A82 E0      1710      LCDWI1:      MOVX     A,@DPTR          ;WAIT FOR BF=0
0A83 20E7FC  1711              JB      ACC.7,LCDWI1
0A86 D082    1712              POP      DPL
0A88 D083    1713              POP      DPH
0A8A 22      1714              RET

```

```

1715
1716 ; ***** LCDWD SUB *****
1717 ; LCD WRITE DATA (RS=1)
1718 ; IN = A
1719 ; REG = A,DPTR
1720
0A8B C083 1721 LCDWD:  PUSH  DPH
0A8D C082 1722          PUSH  DPL
0A8F 90FA02 1723          MOV   DPTR,#LCDWRD
0A92 F0 1724          MOVX  @DPTR,A
0A93 90FA01 1725          MOV   DPTR,#LCDRDC
0A96 E0 1726 LCDWD1:  MOVX  A,@DPTR      ;WAIT FOR BF=0
0A97 20E7FC 1727          JB    ACC.7,LCDWD1
0A9A D082 1728          POP   DPL
0A9C D083 1729          POP   DPH
0A9E 22 1730          RET
1731
1732 ; ***** LCDLDP SUB *****
1733 ; LOAD PMEM TO LCD-MODULE (DMC202)
1734 ; IN = DPTR START BLOCK
1735 ; REG = A,R2,DPTR
1736
0A9F 7480 1737 LCDLDP:  MOV   A,#80H      ;SET ADDRESS LINE 1
0AA1 120AAA 1738          LCALL  LCDLDPS
0AA4 74C0 1739          MOV   A,#0C0H    ;SET ADDRESS LINE 2
0AA6 120AAA 1740          LCALL  LCDLDPS
0AA9 22 1741          RET
1742
0AAA 120A77 1743 LCDLDPS:  LCALL  LCDWI      ;LOAD ONE LINE
0AAD 7A14 1744          MOV   R2,#20     ;20 CHAR.
0AAF E4 1745 LCDLDPS1: CLR   A
0AB0 93 1746          MOVC  A,@A+DPTR   ;PMEM=MOVC
0AB1 120A8B 1747          LCALL  LCDWD      ;WRITE DATA
0AB4 A3 1748          INC   DPTR
0AB5 DAF8 1749          DJNZ  R2,LCDLDPS1
0AB7 22 1750          RET
1751
1752 ; ***** LCDLDD SUB *****
1753 ; LOAD DMEM TO LCD-MODULE (DMC202)
1754 ; IN = DPTR START BLOCK

```

```

1755      ; REG = A,R2,DPTR
1756
0AB8 7480      1757      LCDLDD:  MOV    A,#80H          ;SET ADDRESS LINE 1
0ABA 120AC3     1758                  LCALL   LCDLDDS
0ABD 74C0      1759                  MOV    A,#0C0H          ;SET ADDRESS LINE 2
0ABF 120AC3     1760                  LCALL   LCDLDDS
0AC2 22        1761                  RET
1762
0AC3 120A77     1763      LCDLDDS:  LCALL   LCDWI          ;LOAD ONE LINE
0AC6 7A14      1764                  MOV    R2,#20          ;20 CHAR.
0AC8 E0        1765      LCDLDDS1: MOVX   A,@DPTR          ;DMEM=MOVX
0AC9 120A8B     1766                  LCALL   LCDWD          ;WRITE DATA
0ACC A3        1767                  INC     DPTR
0ACD DAF9      1768                  DJNZ   R2,LCDLDDS1
0ACF 22        1769                  RET
1770
1771      ; ***** PRINTF SUB *****
1772      ; PRINT DATA ON LCD
1773      ; IN  = DPTR
1774      ; OUT = ON LCD
1775      ; REG = A,DPTR
1776
0AD0 E4        1777      PRINTF:  CLR     A
0AD1 93        1778                  MOVC   A,@A+DPTR
0AD2 120A8B     1779      P_TYPE:  LCALL   LCDWD
0AD5 A3        1780                  INC     DPTR
0AD6 E4        1781                  CLR     A
0AD7 93        1782                  MOVC   A,@A+DPTR
0AD8 B40DF7     1783                  CJNE   A,#0DH,P_TYPE
0ADB 22        1784                  RET
1785
1786      ; ***** GOTOXY SUB *****
1787      ; GOTOXY ON LCD
1788      ; IN  = X,Y
1789      ; OUT = A,(TO X,Y ON LCD)
1790      ; REG = A,R0,X,Y,DPTR
1791
0ADC E52B      1792      GOTOXY:  MOV     A,Y
0ADE B40202     1793                  CJNE   A,#02H,Y_1
0AE1 4440      1794                  ORL     A,#40H

```

	1795		
0AE3 4480	1796	Y_1:	ORL A,#80H
0AE5 782A	1797		MOV R0,#X
0AE7 16	1798		DEC @R0
0AE8 46	1799		ORL A,@R0
0AE9 120A77	1800		LCALL LCDWI
0AEC 22	1801		RET
	1802		
	1803		
0AED 50414C20	1804	LO01:	DB "PAL Programmer V1.0 ",0DH
0AF1 50726F67			
0AF5 72616D6D			
0AF9 65722056			
0AFD 312E3020			
0B01 0D			
0B02 4B4D4954	1805	LO02:	DB "KMITL ED.Engineer 16.",0DH
0B06 274C2045			
0B0A 442E456E			
0B0E 67696E65			
0B12 65722031			
0B16 362E0D			
0B19 31393935	1806	LO03:	DB "1995 Buad Rate :9600 ",0
0B1D 20427561			
0B21 64205261			
0B25 7465203A			
0B29 39363030			
0B2D 2000			
	1807		
	1808		
0B2F 20200D	1809	ME01:	DB ' ',0DH
0B32 2D2D2D2D	1810	ME02:	DB '----- MAIN MENU ----- ',0DH
0B36 2D2D2D2D			
0B3A 2D2D2D2D			
0B3E 204D4149			
0B42 4E204D45			
0B46 4E55202D			
0B4A 2D2D2D2D			
0B4E 2D2D2D2D			
0B52 2D2D2D2D			
0B56 200D			

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0B58 412E203D 1811 ME03: DB 'A. ==> Automatic (B&P&V&S) ',0DH

0B5C 3D3E2020

0B60 4175746F

0B64 6D617469

0B68 63202842

0B6C 26502656

0B70 26532920

0B74 0D

0B75 422E203D 1812 ME04: DB "B. ==> Blank check ",0DH

0B79 3D3E2020

0B7D 426C616E

0B81 6B206368

0B85 65636B20

0B89 0D

0B8A 442E203D 1813 ME05: DB "D. ==> Display buffer ",0DH

0B8E 3D3E2020

0B92 44697370

0B96 6C617920

0B9A 62756666

0B9E 6572200D

0BA2 4C2E203D 1814 ME06: DB "L. ==> Load JEDEC file to Buffer ",0DH

0BA6 3D3E2020

0BAA 4C6F6164

0BAE 204A4544

0BB2 45432066

0BB6 696C6520

0BBA 746F2042

0BBE 75666665

0BC2 72200D

0BC5 4D2E203D 1815 ME07: DB "M. ==> Mfr. select ",0DH

0BC9 3D3E2020

0BCD 4D66722E

0BD1 2073656C

0BD5 65637420

0BD9 0D

0BDA 502E203D 1816 ME08: DB "P. ==> Program PAL Device ",0DH

0BDE 3D3E2020

0BE2 50726F67

0BE6 72616D20

0BEA 50414C20

0BEE 44657669
 0BF2 6365200D
 0BF6 522E203D 1817 ME09: DB "R. ==> Read PAL Device ",0DH
 0BFA 3D3E2020
 0BFE 52656164
 0C02 2050414C
 0C06 20446576
 0C0A 69636520
 0C0E 0D
 0C0F 532E203D 1818 ME10: DB "S. ==> Security fuse blow ",0DH
 0C13 3D3E2020
 0C17 53656375
 0C1B 72697479
 0C1F 20667573
 0C23 6520626C
 0C27 6F77200D
 0C2B 542E203D 1819 ME11: DB "T. ==> Type Select ",0DH
 0C2F 3D3E2020
 0C33 54797065
 0C37 2053656C
 0C3B 65637420
 0C3F 0D
 0C40 562E203D 1820 ME12: DB "V. ==> Verify ",0DH
 0C44 3D3E2020
 0C48 56657269
 0C4C 6679200D
 0C50 2D2D2D2D 1821 ME13: DB "-----",0DH
 0C54 2D2D2D2D
 0C58 2D2D2D2D
 0C5C 2D2D2D2D
 0C60 2D2D2D2D
 0C64 2D2D2D2D
 0C68 2D2D2D2D
 0C6C 2D2D2D2D
 0C70 2D2D2D2D
 0C74 200D
 0C76 53656C65 1822 ME14: DB "Select which function ? ",0
 0C7A 63742077
 0C7E 68696368
 0C82 2066756E

0C86 6374696F
 0C8A 6E203F20
 0C8E 00
 1823
 0C8F 446F776E 1824 D_LOAD1: DB "Download JEDEC FILE ... ",0DH
 0C93 6C6F6164
 0C97 204A4544
 0C9B 45432046
 0C9F 494C4520
 0CA3 2E2E2E20
 0CA7 0D
 0CA8 55736520 1825 D_LOAD2: DB "Use ^A and Command to send File ",0DH
 0CAC 5E412061
 0CB0 6E642043
 0CB4 6F6D6D61
 0CB8 6E642074
 0CBC 6F207365
 0CC0 6E642046
 0CC4 696C6520
 0CC8 0D
 1826
 0CC9 44697370 1827 LOGODISP: DB "Display device control by JEDEC format ",0
 0CCD 6C617920
 0CD1 64657669
 0CD5 63652063
 0CD9 6F6E7472
 0CDD 6F6C2062
 0CE1 7920204A
 0CE5 45444543
 0CE9 20666F72
 0CED 6D617420
 0CF1 00
 0CF2 44415441 1828 PGMFUSE: DB "DATA PROGRAM FUSE FROM JEDEC FILE ",0DH
 0CF6 2050524F
 0CFA 4752414D
 0CFE 20465553
 0D02 45204652
 0D06 4F4D204A
 0D0A 45444543
 0D0E 2046494C

0D12 45202020
0D16 20202020
0D1A 0D
0D1B 30303030 1829 NUMBER: DB "00000000001111111112222222223333333333",0DH
0D1F 30303030
0D23 30303131
0D27 31313131
0D2B 31313131
0D2F 32323232
0D33 32323232
0D37 32323333
0D3B 33333333
0D3F 33333333
0D43 0D
0D44 30313233 1830 IN_LINE: DB "0123456789012345678901234567890123456789",0DH
0D48 34353637
0D4C 38393031
0D50 32333435
0D54 36373839
0D58 30313233
0D5C 34353637
0D60 38393031
0D64 32333435
0D68 36373839
0D6C 0D

1831
1832
1833 ;ME01: DB "*****",0
1834 ;ME02: DB " ! HELLO I AM A PAL PROGRAMER V 1.0 ",0
1835 ;ME03: DB " ",0
1836 ;ME04: DB " 1. Load JEDEC file to Buffer ",0
1837 ;ME05: DB " 2. Display buffer ",0
1838 ;ME06: DB " 3. Program PAL Device ",0
1839 ;ME07: DB " 4. Read PAL Device ",0
1840 ;ME08: DB " 5. Select PAL type ",0
1841 ;ME09: DB " 6. Mfr. select ",0
1842 ;ME10: DB " ",0
1843 ;ME11: DB " You Select Number ",0
1844 ;ME12: DB "*****",0
1845

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0D6D 00010001	1846	TPIN5V:	DB 00H,01H,00H,01H,00H,02H,00H,02H
0D71 00020002			
0D75 00040004	1847		DB 00H,04H,00H,04H,00H,08H,00H,08H
0D79 00080008			
0D7D 00100010	1848		DB 00H,10H,00H,10H,00H,20H,00H,20H
0D81 00200020			
0D85 00400040	1849		DB 00H,40H,00H,40H,00H,80H,00H,80H
0D89 00800080			
	1850		
0D8D FEFEFEFE	1851	TPIN12V:	DB 0FEH,0FEH,0FEH,0FEH,0FDH,0FDH,0FDH,0FDH
0D91 FDFDFDFD			
0D95 FBFBFBFB	1852		DB 0FBH,0FBH,0FBH,0FBH,0F7H,0F7H,0F7H,0F7H
0D99 F7F7F7F7			
0D9D EFEFEFEF	1853		DB 0EFH,0EFH,0EFH,0EFH,0DFH,0DFH,0DFH,0DFH
0DA1 DFD FDFD			
0DA5 BFBFBFBF	1854		DB 0BFH,0BFH,0BFH,0BFH, 7FH, 7FH, 7FH, 7FH
0DA9 7F7F7F7F			
	1855		
0DAD 00008080	1856	TL_R:	DB 00H,00H,80H,80H,00H,00H,80H,80H
0DB1 00008080			
0DB5 00008080	1857		DB 00H,00H,80H,80H,00H,00H,80H,80H
0DB9 00008080			
0DBD 00008080	1858		DB 00H,00H,80H,80H,00H,00H,80H,80H
0DC1 00008080			
0DC5 00008080	1859		DB 00H,00H,80H,80H,00H,00H,80H,80H
0DC9 00008080			
	1860		
0DCD 00000101	1861	TL_R2:	DB 00H,00H,01H,01H,00H,00H,01H,01H
0DD1 00000101			
0DD5 00000101	1862		DB 00H,00H,01H,01H,00H,00H,01H,01H
0DD9 00000101			
0DDD 00000101	1863		DB 00H,00H,01H,01H,00H,00H,01H,01H
0DE1 00000101			
0DE5 00000101	1864		DB 00H,00H,01H,01H,00H,00H,01H,01H
0DE9 00000101			
	1865		
0DED 01112131	1866	PRODUCT:	DB 01H,11H,21H,31H,41H,51H,61H,71H
0DF1 41516171			
0DF5 02122232	1867		DB 02H,12H,22H,32H,42H,52H,62H,72H
0DF9 42526272			

เอกสารนี้เป็นเอกสารต้นฉบับที่จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0DFD 04142434	1868		DB 04H,14H,24H,34H,44H,54H,64H,74H
0E01 44546474			
0E05 08182838	1869		DB 08H,18H,28H,38H,48H,58H,68H,78H
0E09 48586878			
	1870		
0E0D 10121416	1871		DB 10H,12H,14H,16H,18H,1AH,1CH,1EH
0E11 181A1C1E			
0E15 20222426	1872		DB 20H,22H,24H,26H,28H,2AH,2CH,2EH
0E19 282A2C2E			
0E1D 40424446	1873		DB 40H,42H,44H,46H,48H,4AH,4CH,4EH
0E21 484A4C4E			
0E25 80828486	1874		DB 80H,82H,84H,86H,88H,8AH,8CH,8EH
0E29 888A8C8E			
	1875		
0E2D 00102030	1876	RPRODUCT:	DB 00H,10H,20H,30H,40H,50H,60H,70H
0E31 40506070			
0E35 00102030	1877		DB 00H,10H,20H,30H,40H,50H,60H,70H
0E39 40506070			
0E3D 00102030	1878		DB 00H,10H,20H,30H,40H,50H,60H,70H
0E41 40506070			
0E45 00102030	1879		DB 00H,10H,20H,30H,40H,50H,60H,70H
0E49 40506070			
	1880		
0E4D 00020406	1881		DB 00H,02H,04H,06H,08H,0AH,0CH,0EH
0E51 080A0C0E			
0E55 00020406	1882		DB 00H,02H,04H,06H,08H,0AH,0CH,0EH
0E59 080A0C0E			
0E5D 00020406	1883		DB 00H,02H,04H,06H,08H,0AH,0CH,0EH
0E61 080A0C0E			
0E65 00020406	1884		DB 00H,02H,04H,06H,08H,0AH,0CH,0EH
0E69 080A0C0E			
	1885		
	1886		;0000000001111111112
	1887		;12345678901234567890
0E6D 50414C20	1888	MENU4T:	DB 'PAL PROGRAMMER V1.0'
0E71 50524F47			
0E75 52414D4D			
0E79 45522056			
0E7D 312E3020			
0E81 20204544	1889		DB ED. Engineer 16

0E85 2E20456E
0E89 67696E65
0E8D 65722031
0E91 36202020
1890
0E95 2A4D6672 1891 NUM_BER: DB '*Mfr. *Type ',0DH
0E99 2E202020
0E9D 20202A54
0EA1 79706520
0EA5 20202020
0EA9 0D
0EAA 20414D44 1892 DAMD: DB 'AMD',0DH
0EAE 0D
0EAF 204D4D49 1893 DMMI: DB 'MMT',0DH
0EB3 0D
0EB4 204E5320 1894 DNS: DB 'NS',0DH
0EB8 0D
0EB9 20313652 1895 D16R4: DB '16R4',0DH
0EBD 340D
0EBF 20313652 1896 D16R6: DB '16R6',0DH
0EC3 360D
0EC5 2031364C 1897 D16L8: DB '16L8',0DH
0EC9 380D
0ECB 20313652 1898 D16R8: DB '16R8',0DH
0ECF 380D
0ED1 20323052 1899 D20R4: DB '20R4',0DH
0ED5 340D
0ED7 20323052 1900 D20R6: DB '20R6',0DH
0EDB 360D
0EDD 2032304C 1901 D20L8: DB '20L8',0DH
0EE1 380D
0EE3 20323052 1902 D20R8: DB '20R8',0DH
0EE7 380D
1903 ;000000001111111112
1904 ;12345678901234567890
0EE9 20446973 1905 DDISPLAY: DB 'Display device ',0DH
0EED 706C6179
0EF1 20646576
0EF5 69636520
0EF9 20200D

0EF0 204C6F61 1906 DLOAD: DB ' Load JEDEC File ',0DH
0F00 64204A45
0F04 44454320
0F08 46696C65
0F0C 20200D
0F0F 20526561 1907 DREAD: DB ' Read device ',0DH
0F13 64206465
0F17 76696365
0F1B 20202020
0F1F 20200D
0F22 2050726F 1908 DPROGRAM: DB ' Program device ',0DH
0F26 6772616D
0F2A 20646576
0F2E 69636520
0F32 20200D
0F35 20202020 1909 DCLS: DB ' ',0DH
0F39 20202020
0F3D 20202020
0F41 20202020
0F45 20200D
1910
1911
1912
1913
1914
1915
0000= 1916 END

addixx = 0026	addiyy = 0028	ahigh = 0021	alow = 0022	atoh = 09FB
atohs = 0A07	atohs1 = 0A0E	buffer = 0000	buffer2 = 8000	c_16l8 = 0332
c_16r4 = 02F6	c_16r6 = 0314	c_16r8 = 0350	c_20l8 = 03AA	c_20r4 = 036E
c_20r6 = 038C	c_20r8 = 03C8	clrpgrm = 0967	clrram = 085E	clrram01 = 0863
clrram02 = 0865	clrram2 = 0872	clrram21 = 0877	clrram22 = 0879	code_3 = 0205
code_4 = 020B	code_5 = 0211	code_a = 029C	code_d = 01D0	code_l = 01A0
code_m = 02BA	code_n = 02D8	code_p = 021D	code_r = 025B	codefuse = 0025
conf_1 = FC03	conf_2 = F803	d100usec = 0941	d10usec = 08BC	d16l8 = 0EC5
d16r4 = 0EB9	d16r6 = 0EBF	d16r8 = 0ECB	d20l8 = 0EDD	d20r4 = 0ED1
d20r6 = 0ED7	d20r8 = 0EE3	d20usec = 08CF	d30usec = 08E2	d40usec = 08F5
d50usec = 0954	d50usec = 0908	d60usec = 091B	d80usec = 092E	d_load1 = 0C8F

d_load2 = 0CA8	damd = 0EAA	dcls = 0F35	ddisplay = 0EE9	delo = 001E
dis02 = 054A	dis04 = 056D	disp = 04F6	disp01 = 0505	disp03 = 0561
dload = 0EFC	dmin = 08B4	drmi = 0EAF	dmsec = 0896	dmsec1 = 0898
dns = 0EB4	dpcom = 0886	dprogram = 0F22	dread = 0F0F	dsec = 08AC
dtoh = 0A2E	dtoh1 = 0A30	dtoh2 = 0A35	dtoh3 = 0A3F	dtoh4 = 0A45
dtsec = 089F	dtsec1 = 08A1	dump = 083A	dumpram = 0816	end_load = 04D4
end_load2 = 04DB	end_save = 0813	exit_save = 07F7	f = 09A6	f_q0 = 068B
f_q1 = 0690	f_q2 = 0695	f_q3 = 069A	f_q4 = 069F	f_q5 = 06A4
f_q6 = 06A9	f_q7 = 06AE	fuse = 05E9	fuse0 = 05A5	fuse1 = 05C7
fuse_eof = 0603	fuse_q0 = 077D	fuse_q1 = 078A	fuse_q2 = 0797	fuse_q3 = 07A4
fuse_q4 = 07B1	fuse_q5 = 07BE	fuse_q6 = 07CB	fuse_q7 = 07D8	gotoxy = 0ADC
htoa = 0A11	htoas = 0A1F	htoas1 = 0A29	htod = 0A58	htod1 = 0A5E
htod2 = 0A6C	in_line = 0D44	incfuse = 05AE	l_fa = 03E6	l_w = 03F7
lcdbuf = 9000	lcdldd = 0AB8	lcdldds = 0AC3	lcdldds1 = 0AC8	lcdldp = 0A9F
lcdldps = 0AAA	lcdldps1 = 0AAF	lcdrdc = FA01	lcdrdd = FA03	lcdwd = 0A8B
lcdwd1 = 0A96	lcdwi = 0A77	lcdwi1 = 0A82	lcdwrc = FA00	lcdwrd = FA02
line1 = 0161	lo01 = 0AED	lo02 = 0B02	lo03 = 0B19	load = 0438
load01 = 0466	load02 = 046F	load03 = 047A	load04 = 046B	load2 = 0488
load21 = 048A	load3 = 04BA	load4 = 04C5	load5 = 04CE	logodisp = 0CC9
loop = 014D	loop2 = 0169	lr = 0020	main4 = 0134	me01 = 0B2F
me02 = 0B32	me03 = 0B58	me04 = 0B75	me05 = 0B8A	me06 = 0BA2
me07 = 0BC5	me08 = 0BDA	me09 = 0BF6	me10 = 0C0F	me11 = 0C2B
me12 = 0C40	me13 = 0C50	me14 = 0C76	menu = 03FA	menu4t = 0E6D
num_ber = 0E95	number = 0D1B	p_con = F802	p_type = 0AD2	palpgm = 057C
pdata = FC02	pgm = F801	pgm0_31 = 060C	pgm32_63 = 0641	pgmfuse = 0CF2
pi12v = FC01	pi5v = FC00	pp = 081C	pp2 = 0840	pqa = F800
printf = 0AD0	product = 0DED	rbyte = 09E5	rbyteh = 09EF	read0_31 = 06D1
read32_63 = 0727	read_fuse = 0676	readfuse = 05A4	readq0_7 = 067D	recv = 0450
recv01 = 0456	recv02 = 0459	res = 0100	res1 = 0102	rfd_31 = 0688
rhig = 0023	rlow = 0024	rproduct = 0E2D	s = 0833	s2 = 0857
s_jedec = 04DF	save_fuse0 = 07E5	save_fuse1 = 07EE	sblank = 0996	sblock = 0981
sblock1 = 0988	sblock2 = 0990	sbyte = 09BD	sbyteh = 09C7	sbytehc = 09D3
shift = 0141	slf = 09B2	space = 099E	stack = 0060	tl_r = 0DAD
tl_r2 = 0DCD	tpin12v = 0D8D	tpin5v = 0D6D	w = 0194	wait = 018B
work_q0 = 06B9	work_q1 = 06BC	work_q2 = 06BF	work_q3 = 06C2	work_q4 = 06C5
work_q5 = 06C8	work_q6 = 06CB	work_q7 = 06CE	x = 002A	y = 002B
y_1 = 0AE3				

โปรแกรมทดสอบบอร์ด

8051 Cross-Assembler (1.3) (C) 1987, 1989 Binary Technology

TEST.ASM

```
1
FC00= 2      PI5V      EQU    0FC00H
FC01= 3      PI12V     EQU    0FC01H
FC02= 4      PDATA     EQU    0FC02H
FC03= 5      CONF_1    EQU    0FC03H      ;8255/1
6
F800= 7      PQA        EQU    0F800H
F801= 8      PGM        EQU    0F801H
F802= 9      P_CON      EQU    0F802H
F803= 10     CONF_2     EQU    0F803H      ;8255/2
11
0000= 12     BUFFER     EQU    0000H
8000= 13     BUFFER2    EQU    8000H
14
15     ; ***** WORKING RAM *****
16
0020 17     ORG 20H
18
0020 19     LR:          DS      1
0021 20     AHIGH:       DS      1
0022 21     ALOW:        DS      1
0023 22     RHIGH:       DS      1
0024 23     RLOW:        DS      1
0025 24     CODEFUSE:    DS      1
0026 25     ADDIXX:       DS      2
0028 26     ADDIYY:       DS      2
002A 27     X:           DS      1
002B 28     Y:           DS      1
29
0060 30     ORG 60H
```

```

34 ;***** MAIN PROGRAM *****
0000 35      .   ORG 0000H
36
0000 020100 37      LJMP  RES
38
39 ;***** RESET SYSTEM *****
0100 40      ORG 0100H
41
0100 7A80 42 RES:    MOV  R2,#80H
0102 7B00 43 RES1:   MOV  R3,#0
0104 DBFE 44          DJNZ R3,$
0106 DAFA 45          DJNZ R2,RES1      ;DELAY SYSTEM
46
47
0108 758180 48          MOV  SP,#STACK
010B 759852 49          MOV  SCON,#52H
010E 758920 50          MOV  TMOD,#20H      ;SET SERIAL PORT
0111 758DFD 51          MOV  TH1,#0FDH      ;BUAD RATE 9600
0114 D28E 52          SETB  TR1
53
0116 7489 54          MOV  A,#89H      ;{ SET SYSTEM
0118 90FC03 55          MOV  DPTR,#CONF_1  ;A = B = OUT
011B F0 56          MOVX  @DPTR,A    ;C = IN
57
011C 7480 58          MOV  A,#80H      ;A = B = C = OUT
011E 90F803 59          MOV  DPTR,#CONF_2
0121 F0 60          MOVX  @DPTR,A
61
0122 120680 62          LCALL CLRPGM      ;} END SET SYSTEM
63
0125 20B303 64 WAIT:   JB   INT1,$+6
0128 7590C0 65          MOV  P1,#0C0H      ;ERROR
012B 3098F7 66          JNB  RI,WAIT
67
012E 1206E0 68          LCALL RBYTE
0131 B42006 69          CJNE  A,#20H,CODE_1
0134 12014F 70          LCALL MENU
0137 020125 71          LJMP  WAIT

```

```

72
013A B43103 73 CODE_1: CJNE A,#'1',CODE_2
013D 020125 74 LJMP WAIT
75
0140 B43203 76 CODE_2: CJNE A,#'2',CODE_3
0143 020125 77 LJMP WAIT
78
0146 B433DC 79 CODE_3: CJNE A,#'3',WAIT
0149 120178 80 LCALL TPIN
014C 020125 81 LJMP WAIT
82
83
84
85 ; ***** MENU SUB *****
86 ; DISPLAY TO MONITOR
87 ; IN = NO REG
88 ; OUT = NO REG ; DISPLAY TO MONITOR
89 ; REG = A,DPTR
90
014F C083 91 MENU: PUSH DPH
0151 C082 92 PUSH DPL
93
0153 1206CB 94 LCALL SLF
0156 90072C 95 MOV DPTR,#DTEST
0159 12069A 96 LCALL SBLOCK
.97
015C 900751 98 MOV DPTR,#DTEST1
015F 12069A 99 LCALL SBLOCK
100
0162 90076F 101 MOV DPTR,#DTEST2
0165 12069A 102 LCALL SBLOCK
103
0168 90078D 104 MOV DPTR,#DTEST3
016B 12069A 105 LCALL SBLOCK
106
016E 742D 107 MOV A,#'-'
0170 1206D6 108 LCALL SBYTE

```

```

0173 D082      110      POP    DPL
0175 D083      111      POP    DPH
0177 22        112      RET
113
114 ; ***** TPIN SUB *****
115 ; DISPLAY GND TO PIN XX VCC = X V
116 ; IN = KEY '3'
117 ; OUT = NO REG ;DISPLAY TO MONITOR
118 ; REG = A,DPTR
119
120
121 TPIN:      ;***** PIN 3 = 0 V *****
0178 1206CB    122      LCALL SLF
017B 9007A2    123      MOV    DPTR,#DPT3_0V
017E 12069A    124      LCALL SBLOCK
0181 E4        125      CLR    A
0182 90F801    126      MOV    DPTR,#PGM
0185 F0        127      MOVX   @DPTR,A
128
0186 900C89    129      MOV    DPTR,#DGETCH
0189 12069A    130      LCALL SBLOCK
018C 1206E0    131      LCALL RBYTE
132
133      ;***** PIN 3 = 5 V *****
018F 1206CB    134      LCALL SLF
0192 9007BA    135      MOV    DPTR,#DPT3_5V
0195 12069A    136      LCALL SBLOCK
0198 74C1      137      MOV    A,#11000001B
019A 90F801    138      MOV    DPTR,#PGM
019D F0        139      MOVX   @DPTR,A
140
019E 900C89    141      MOV    DPTR,#DGETCH
01A1 12069A    142      LCALL SBLOCK
01A4 1206E0    143      LCALL RBYTE
144
145      ;***** PIN 3 = 12 V *****
01A7 1206CB    146      LCALL SLF
01AA 9007D2    147      MOV    DPTR,#DPT3_12V

```

01AD 12069A	148	LCALL SBLOCK
01B0 74C2	149	MOV A,#11000010B
01B2 90F801	150	MOV DPTR,#PGM
01B5 F0	151	MOVX @DPTR,A
	152	
01B6 900C69	153	MOV DPTR,#DGETCH
01B9 12069A	154	LCALL SBLOCK
01BC 1206E0	155	LCALL RBYTE
	156	
01BF 74C0	157	MOV A,#11000000B
01C1 90F801	158	MOV DPTR,#PGM
01C4 F0	159	MOVX @DPTR,A
	160	
	161	;***** PIN 4 = 0 V *****
01C5 1206CB	162	LCALL SLF
01C8 9007EB	163	MOV DPTR,#DPT4_0V
01CB 12069A	164	LCALL SBLOCK
01CE E4	165	CLR A
01CF 90FC00	166	MOV DPTR,#PI5V
01D2 F0	167	MOVX @DPTR,A
	168	
01D3 90FC01	169	MOV DPTR,#PI12V
01D6 F0	170	MOVX @DPTR,A
	171	
01D7 900C69	172	MOV DPTR,#DGETCH
01DA 12069A	173	LCALL SBLOCK
01DD 1206E0	174	LCALL RBYTE
	175	
	176	;***** PIN 4 = 5 V *****
01E0 1206CB	177	LCALL SLF
01E3 900803	178	MOV DPTR,#DPT4_5V
01E6 12069A	179	LCALL SBLOCK
	180	
01E9 7401	181	MOV A,#01H
01EB 90FC00	182	MOV DPTR,#PI5V
01EE F0	183	MOVX @DPTR,A
	184	
01EF E4	185	CLR A

01F0 90FC01	186	MOV DPTR,#PI12V
01F3 F0	187	MOVX @DPTR,A
	188	
01F4 900C89	189	MOV DPTR,#DGETCH
01F7 12069A	190	LCALL SBLOCK
01FA 1206E0	191	LCALL RBYTE
	192	
	193	;***** PIN 4 = 12 V *****
01FD 1206CB	194	LCALL SLF
0200 90081B	195	MOV DPTR,#DPT4_12V
0203 12069A	196	LCALL SBLOCK
0206 E4	197	CLR A
0207 90FC00	198	MOV DPTR,#PI5V
020A F0	199	MOVX @DPTR,A
	200	
020B 7401	201	MOV A,#01H
020D 90FC01	202	MOV DPTR,#PI12V
0210 F0	203	MOVX @DPTR,A
	204	
0211 900C89	205	MOV DPTR,#DGETCH
0214 12069A	206	LCALL SBLOCK
0217 1206E0	207	LCALL RBYTE
	208	
	209	;***** PIN 5 = 0 V *****
021A 1206CB	210	LCALL SLF
021D 900834	211	MOV DPTR,#DPT5_0V
0220 12069A	212	LCALL SBLOCK
0223 E4	213	CLR A
0224 90FC00	214	MOV DPTR,#PI5V
0227 F0	215	MOVX @DPTR,A
	216	
0228 90FC01	217	MOV DPTR,#PI12V
022B F0	218	MOVX @DPTR,A
	219	
022C 900C89	220	MOV DPTR,#DGETCH
022F 12069A	221	LCALL SBLOCK
0232 1206E0	222	LCALL BYTE

	224	;***** PIN 5 = 5 V *****
0235 1206CB	225	LCALL SLF
0238 90084C	225	MOV DPTR,#DPT5_5V
023B 12069A	227	LCALL SBLOCK
	228	
023E 7402	229	MOV A,#02H
0240 90FC00	230	MOV DPTR,#PI5V
0243 F0	231	MOVX @DPTR,A
	232	
0244 E4	233	CLR A
0245 90FC01	234	MOV DPTR,#PI12V
0248 F0	235	MOVX @DPTR,A
	236	
0249 900C69	237	MOV DPTR,#DGETCH
024C 12069A	238	LCALL SBLOCK
024F 1206E0	239	LCALL RBYTE
	240	
	241	;***** PIN 5 = 12 V *****
0252 1206CB	242	LCALL SLF
0255 900864	243	MOV DPTR,#DPT5_12V
0258 12069A	244	LCALL SBLOCK
025B E4	245	CLR A
025C 90FC00	246	MOV DPTR,#PI5V
025F F0	247	MOVX @DPTR,A
	248	
0260 7402	249	MOV A,#02H
0262 90FC01	250	MOV DPTR,#PI12V
0265 F0	251	MOVX @DPTR,A
	252	
0266 900C69	253	MOV DPTR,#DGETCH
0269 12069A	254	LCALL SBLOCK
026C 1206E0	255	LCALL RBYTE
	256	
	257	;***** PIN 6 = 0 V *****
026F 1206CB	258	LCALL SLF
0272 90087D	259	MOV DPTR,#DPT6_0V
0275 12069A	260	LCALL SBLOCK
0278 E4	261	CLR A

0279 90FC00	262	MOV DPTR,#PI5V
027C F0	263	MOVX @DPTR,A
	264	
027D 90FC01	265	MOV DPTR,#PI12V
0280 F0	266	MOVX @DPTR,A
	267	
0281 900C69	268	MOV DPTR,#DGETCH
0284 12069A	269	LCALL SBLOCK
0287 1206E0	270	LCALL RBYTE
	271	
	272	;***** PIN 6 = 5 V *****
028A 1206CB	273	LCALL SLF
028D 900895	274	MOV DPTR,#DPT6_5V
0290 12069A	275	LCALL SBLOCK
	276	
0293 7404	277	MOV A,#04H
0295 90FC00	278	MOV DPTR,#PI5V
0298 F0	279	MOVX @DPTR,A
	280	
0299 E4	281	CLR A
029A 90FC01	282	MOV DPTR,#PI12V
029D F0	283	MOVX @DPTR,A
	284	
029E 900C69	285	MOV DPTR,#DGETCH
02A1 12069A	286	LCALL SBLOCK
02A4 1206E0	287	LCALL RBYTE
	288	
	289	;***** PIN 6 = 12 V *****
02A7 1206CB	290	LCALL SLF
02AA 9008AD	291	MOV DPTR,#DPT6_12V
02AD 12069A	292	LCALL SBLOCK
02B0 E4	293	CLR A
02B1 90FC00	294	MOV DPTR,#PI5V
02B4 F0	295	MOVX @DPTR,A
	296	
02B5 7404	297	MOV A,#04H
02B7 90FC01	298	MOV DPTR,#PI12V
02BA F0	299	MOVX @DPTR,A

```

300
02BB 900C89 301      MOV  DPTR,#DGETCH
02BE 12069A 302      LCALL SBLOCK
02C1 1206E0 303      LCALL RBYTE
304
305      ;***** PIN 7 = 0 V *****
02C4 1206CB 306      LCALL SLF
02C7 9008C6 307      MOV  DPTR,#DPT7_0V
02CA 12069A 308      LCALL SBLOCK
02CD E4      309      CLR   A
02CE 90FC00 310      MOV  DPTR,#PI5V
02D1 F0      311      MOVX  @DPTR,A
312
02D2 90FC01 313      MOV  DPTR,#PI12V
02D5 F0      314      MOVX  @DPTR,A
315
02D6 900C89 316      MOV  DPTR,#DGETCH
02D9 12069A 317      LCALL SBLOCK
02DC 1206E0 318      LCALL RBYTE
319
320      ;***** PIN 7 = 5 V *****
02DF 1206CB 321      LCALL SLF
02E2 9008DE 322      MOV  DPTR,#DPT7_5V
02E5 12069A 323      LCALL SBLOCK
324
02E8 7408    325      MOV  A,#08H
02EA 90FC00 326      MOV  DPTR,#PI5V
02ED F0      327      MOVX  @DPTR,A
328
02EE E4      329      CLR   A
02EF 90FC01 330      MOV  DPTR,#PI12V
02F2 F0      331      MOVX  @DPTR,A
332
02F3 900C89 333      MOV  DPTR,#DGETCH
02F6 12069A 334      LCALL SBLOCK
02F9 1206E0 335      LCALL RBYTE

```

02FC 1206CB	338	LCALL	SLF
02FF 9008F6	339	MOV	DPTR,#DPT7_12V
0302 12069A	340	LCALL	SBLOCK
0305 E4	341	CLR	A
0306 90FC00	342	MOV	DPTR,#PI5V
0309 F0	343	MOVX	@DPTR,A
	344		
030A 7408	345	MOV	A,#08H
030C 90FC01	346	MOV	DPTR,#PI12V
030F F0	347	MOVX	@DPTR,A
	348		
0310 900C69	349	MOV	DPTR,#DGETCH
0313 12069A	350	LCALL	SBLOCK
0316 1206E0	351	LCALL	RBYTE
	352		
	353	;***** PIN 8 = 0 V *****	
0319 1206CB	354	LCALL	SLF
031C 90090F	355	MOV	DPTR,#DPT8_0V
031F 12069A	356	LCALL	BLOCK
0322 E4	357	CLR	A
0323 90FC00	358	MOV	DPTR,#PI5V
0326 F0	359	MOVX	@DPTR,A
	360		
0327 90FC01	361	MOV	DPTR,#PI12V
032A F0	362	MOVX	@DPTR,A
	363		
032B 900C69	364	MOV	DPTR,#DGETCH
032E 12069A	365	LCALL	SBLOCK
0331 1206E0	366	LCALL	RBYTE
	367		
	368	;***** PIN 8 = 5 V *****	
0334 1206CB	369	LCALL	SLF
0337 900927	370	MOV	DPTR,#DPT8_5V
033A 12069A	371	LCALL	SBLOCK
	372		
033D 7410	373	MOV	A,#10H
033F 90FC00	374	MOV	DPTR,#PI5V
0342 F0	375	MOVX	@DPTR,A

```

376
0343 E4      377      CLR      A
0344 90FC01   378      MOV      DPTR,#PI12V
0347 F0      379      MOVX     @DPTR,A
380
0348 900C89   381      MOV      DPTR,#DGETCH
034B 12069A   382      LCALL     SBLOCK
034E 1206E0   383      LCALL     RBYTE
384
385           ;***** PIN 8 = 12 V *****
0351 1206CB   386      LCALL     SLF
0354 90093F   387      MOV      DPTR,#DPT8_12V
0357 12069A   388      LCALL     SBLOCK
035A E4      389      CLR      A
035B 90FC00   390      MOV      DPTR,#PI5V
035E F0      391      MOVX     @DPTR,A
392
035F 7410     393      MOV      A,#10H
0361 90FC01   394      MOV      DPTR,#PI12V
0364 F0      395      MOVX     @DPTR,A
396
0365 900C89   397      MOV      DPTR,#DGETCH
0368 12069A   398      LCALL     SBLOCK
036B 1206E0   399      LCALL     RBYTE
400
401           ;***** PIN 9 = 0 V *****
036E 1206CB   402      LCALL     SLF
0371 900958   403      MOV      DPTR,#DPT9_0V
0374 12069A   404      LCALL     SBLOCK
0377 E4      405      CLR      A
0378 90FC00   406      MOV      DPTR,#PI5V
037B F0      407      MOVX     @DPTR,A
408
037C 90FC01   409      MOV      DPTR,#PI12V
037F F0      410      MOVX     @DPTR,A
411
0380 900C89   412      MOV      DPTR,#DGETCH
0383 12069A   413      LCALL     SBLOCK

```

```

0386 1206E0    414      LCALL  RBYTE
                                415
                                416      ;***** PIN 9 = 5 V *****
0389 1206CB    417      LCALL  SLF
038C 900970    418      MOV    DPTR,#DPT9_5V
038F 12069A    419      LCALL  SBLOCK
                                420
0392 7420      421      MOV    A,#20H
0394 90FC00    422      MOV    DPTR,#PI5V
0397 F0        423      MOVX   @DPTR,A
                                424
0398 E4        425      CLR    A
0399 90FC01    426      MOV    DPTR,#PI12V
039C F0        427      MOVX   @DPTR,A
                                428
039D 900C69    429      MOV    DPTR,#DGETCH
03A0 12069A    430      LCALL  SBLOCK
03A3 1206E0    431      LCALL  RBYTE
                                432
                                433      ;***** PIN 9 = 12 V *****
03A6 1206CB    434      LCALL  SLF
03A9 900988    435      MOV    DPTR,#DPT9_12V
03AC 12069A    436      LCALL  SBLOCK
03AF E4        437      CLR    A
03B0 90FC00    438      MOV    DPTR,#PI5V
03B3 F0        439      MOVX   @DPTR,A
                                440
03B4 7420      441      MOV    A,#20H
03B6 90FC01    442      MOV    DPTR,#PI12V
03B9 F0        443      MOVX   @DPTR,A
                                444
03BA 900C69    445      MOV    DPTR,#DGETCH
03BD 12069A    446      LCALL  SBLOCK
03C0 1206E0    447      LCALL  RBYTE
                                448
                                449      ;***** PIN 10 = 0 V *****

```

03C3 1206CB

450

LCALL SLF

03C6 9009A1

451

MOV DPTR,#DPT10_0V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

03C9 12069A	452	LCALL	SBLOCK
03CC E4	453	CLR	A
03CD 90FC00	454	MOV	DPTR,#PI5V
03D0 F0	455	MOVX	@DPTR,A
	456		
03D1 90FC01	457	MOV	DPTR,#PI12V
03D4 F0	458	MOVX	@DPTR,A
	459		
03D5 900C69	460	MOV	DPTR,#DGETCH
03D8 12069A	461	LCALL	SBLOCK
03DB 1206E0	462	LCALL	RBYTE
	463		
	464	;***** PIN 10 = 5 V *****	
03DE 1206CB	465	LCALL	SLF
03E1 9009BA	466	MOV	DPTR,#DPT10_5V
03E4 12069A	467	LCALL	SBLOCK
	468		
03E7 7440	469	MOV	A,#40H
03E9 90FC00	470	MOV	DPTR,#PI5V
03EC F0	471	MOVX	@DPTR,A
	472		
03ED E4	473	CLR	A
03EE 90FC01	474	MOV	DPTR,#PI12V
03F1 F0	475	MOVX	@DPTR,A
	476		
03F2 900C69	477	MOV	DPTR,#DGETCH
03F5 12069A	478	LCALL	SBLOCK
03F8 1206E0	479	LCALL	RBYTE
	480		
	481	;***** PIN 10 = 12 V *****	
03FB 1206CB	482	LCALL	SLF
03FE 9009D3	483	MOV	DPTR,#DPT10_12V
0401 12069A	484	LCALL	SBLOCK
0404 E4	485	CLR	A
0405 90FC00	486	MOV	DPTR,#PI5V
0408 F0	487	MOVX	@DPTR,A
	488		
0409 7440	489	MOV	A,#40H

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

040B 90FC01    490      MOV    DPTR,#PI12V
040E F0        491      MOVX   @DPTR,A
                                492
040F 900C69    493      MOV    DPTR,#DGETCH
0412 12069A    494      LCALL  SBLOCK
0415 1206E0    495      LCALL  RBYTE
                                496
                                497      ;***** PIN 11 = 0 V *****
0418 1206CB    498      LCALL  SLF
041B 9009ED    499      MOV    DPTR,#DPT11_0V
041E 12069A    500      LCALL  SBLOCK
0421 E4        501      CLR    A
0422 90FC00    502      MOV    DPTR,#PI5V
0425 F0        503      MOVX   @DPTR,A
                                504
0426 90FC01    505      MOV    DPTR,#PI12V
0429 F0        506      MOVX   @DPTR,A
                                507
042A 900C69    508      MOV    DPTR,#DGETCH
042D 12069A    509      LCALL  SBLOCK
0430 1206E0    510      LCALL  RBYTE
                                511
                                512      ;***** PIN 11 = 5 V *****
0433 1206CB    513      LCALL  SLF
0436 900A06    514      MOV    DPTR,#DPT11_5V
0439 12069A    515      LCALL  SBLOCK
                                516
043C 7480      517      MOV    A,#80H
043E 90FC00    518      MOV    DPTR,#PI5V
0441 F0        519      MOVX   @DPTR,A
                                520
0442 E4        521      CLR    A
0443 90FC01    522      MOV    DPTR,#PI12V
0446 F0        523      MOVX   @DPTR,A
                                524
0447 900C69    525      MOV    DPTR,#DGETCH
044A 12069A    526      LCALL  SBLOCK
044D 1206E0    527      LCALL  RBYTE

```

```

528
529      ;***** PIN 11 - 12 V *****
0450 1206CB 530      LCALL    SLF
0453 900A1F 531      MOV     DPTR,#DPT11_12V
0456 12069A 532      LCALL    SBLOCK
0459 E4      533      CLR     A
045A 90FC00 534      MOV     DPTR,#PI5V
045D F0      535      MOVX    @DPTR,A
536
045E 7480    537      MOV     A,#80H
0460 90FC01 538      MOV     DPTR,#PI12V
0463 F0      539      MOVX    @DPTR,A
540
0464 900C69 541      MOV     DPTR,#DGETCH
0467 12069A 542      LCALL    SBLOCK
046A 1206E0 543      LCALL    RBYTE
046D 120680 544      LCALL    CLRPGM
545
546      ;***** PIN 13 - 0 V *****
0470 1206CB 547      LCALL    SLF
0473 900A39 548      MOV     DPTR,#DPT13_0V
0476 12069A 549      LCALL    SBLOCK
0479 E4      550      CLR     A
047A 90F801 551      MOV     DPTR,#PGM
047D F0      552      MOVX    @DPTR,A
553
047E 900C69 554      MOV     DPTR,#DGETCH
0481 12069A 555      LCALL    SBLOCK
0484 1206E0 556      LCALL    RBYTE
557
558      ;***** PIN 13 - 5 V *****
0487 1206CB 559      LCALL    SLF
048A 900A52 560      MOV     DPTR,#DPT13_5V
048D 12069A 561      LCALL    SBLOCK
0490 74D0    562      MOV     A,#11010000B
0492 90F801 563      MOV     DPTR,#PGM
0495 F0      564      MOVX    @DPTR,A
565

```

```

0496 900C69      566      MOV     DPTR,#DGETCH
0499 12069A      567      LCALL   SBLOCK
049C 1206E0      568      LCALL   RBYTE
                    569
                    570      ;***** PIN 13 = 12 V ***** CLOCK,OD
049F 1206CB      571      LCALL   SLF
04A2 900A6B      572      MOV     DPTR,#DPT13_12V
04A5 12069A      573      LCALL   SBLOCK
04A8 74E0        574      MOV     A,#11100000B
04AA 90F801      575      MOV     DPTR,#PGM
04AD F0          576      MOVX    @DPTR,A
                    577
04AE 900C69      578      MOV     DPTR,#DGETCH
04B1 12069A      579      LCALL   SBLOCK
04B4 1206E0      580      LCALL   RBYTE
04B7 120680      581      LCALL   CLRPGM
                    582
                    583      ;***** PIN 14 = 0 V ***** L/R
04BA 1206CB      584      LCALL   SLF
04BD 900A85      585      MOV     DPTR,#DPT14_0V
04C0 12069A      586      LCALL   SBLOCK
                    587
04C3 E4          588      CLR     A
04C4 90F800      589      MOV     DPTR,#PQA
04C7 F0          590      MOVX    @DPTR,A
                    591
04C8 900C69      592      MOV     DPTR,#DGETCH
04CB 12069A      593      LCALL   SBLOCK
04CE 1206E0      594      LCALL   RBYTE
                    595
                    596      ;***** PIN 14 = 12 V *****
04D1 1206CB      597      LCALL   SLF
04D4 900A9E      598      MOV     DPTR,#DPT14_12V
04D7 12069A      599      LCALL   SBLOCK
                    600
04DA 7480        601      MOV     A,#80H
04DC 90F800      602      MOV     DPTR,#PQA
04DF F0          603      MOVX    @DPTR,A

```

```

604
04E0 900C69 605      MOV    DPTR,#DGETCH
04E3 12069A 606      LCALL  SBLOCK
04E6 1206E0 607      LCALL  RBYTE
608
609      ;***** PIN 15 = 0 V *****
04E9 1206CB 610      LCALL  SLF
04EC 900AB8 611      MOV    DPTR,#DPT15_0V
04EF 12069A 612      LCALL  SBLOCK
613
04F2 E4      614      CLR    A
04F3 90F800 615      MOV    DPTR,#PQA
04F6 F0      616      MOVX   @DPTR,A
617
04F7 900C69 618      MOV    DPTR,#DGETCH
04FA 12069A 619      LCALL  SBLOCK
04FD 1206E0 620      LCALL  RBYTE
621
622      ;***** PIN 15 = 12 V *****
0500 1206CB 623      LCALL  SLF
0503 900AD1 624      MOV    DPTR,#DPT15_12V
0506 12069A 625      LCALL  SBLOCK
626
0509 7440    627      MOV    A,#40H
050B 90F800 628      MOV    DPTR,#PQA
050E F0      629      MOVX   @DPTR,A
630
050F 900C69 631      MOV    DPTR,#DGETCH
0512 12069A 632      LCALL  SBLOCK
0515 1206E0 633      LCALL  RBYTE
634
635      ;***** PIN 16 = 0 V *****
0518 1206CB 636      LCALL  SLF
051B 900AEB 637      MOV    DPTR,#DPT16_0V
051E 12069A 638      LCALL  SBLOCK
639
0521 E4      640      CLR    A
0522 90F800 641      MOV    DPTR,#PQA

```

0525 F0	642	MOVX @DPTR,A
	643	
0526 900C89	644	MOV DPTR,#DGETCH
0529 12069A	645	LCALL SBLOCK
052C 1206E0	646	LCALL RBYTE
	647	
	648	;***** PIN 16 = 12 V *****
052F 1206CB	649	LCALL SLF
0532 900B04	650	MOV DPTR,#DPT16_12V
0535 12069A	651	LCALL SBLOCK
	652	
0538 7420	653	MOV A,#20H
053A 90F800	654	MOV DPTR,#PQA
053D F0	655	MOVX @DPTR,A
	656	
053E 900C89	657	MOV DPTR,#DGETCH
0541 12069A	658	LCALL SBLOCK
0544 1206E0	659	LCALL RBYTE
	660	
	661	;***** PIN 17 = 0 V *****
0547 1206CB	662	LCALL SLF
054A 900B1E	663	MOV DPTR,#DPT17_0V
054D 12069A	664	LCALL SBLOCK
	665	
0550 E4	666	CLR A
0551 90F800	667	MOV DPTR,#PQA
0554 F0	668	MOVX @DPTR,A
	669	
0555 900C89	670	MOV DPTR,#DGETCH
0558 12069A	671	LCALL SBLOCK
055B 1206E0	672	LCALL RBYTE
	673	
	674	;***** PIN 17 = 12 V *****
055E 1206CB	675	LCALL SLF
0561 900B37	676	MOV DPTR,#DPT17_12V
0564 12069A	677	LCALL SBLOCK

0569 90F800	680	MOV	DPTR,#PQA
056C F0	681	MOVX	@DPTR,A
	682		
056D 900C69	683	MOV	DPTR,#DGETCH
0570 12069A	684	LCALL	SBLOCK
0573 1206E0	685	LCALL	RBYTE
	686		
	687	;***** PIN 18 = 0 V *****	
0576 1206CB	688	LCALL	SLF
0579 900B51	689	MOV	DPTR,#DPT18_0V
057C 12069A	690	LCALL	SBLOCK
	691		
057F E4	692	CLR	A
0580 90F800	693	MOV	DPTR,#PQA
0583 F0	694	MOVX	@DPTR,A
	695		
0584 900C69	696	MOV	DPTR,#DGETCH
0587 12069A	697	LCALL	SBLOCK
058A 1206E0	698	LCALL	RBYTE
	699		
	700	;***** PIN 18 = 12 V *****	
058D 1206CB	701	LCALL	SLF
0590 900B6A	702	MOV	DPTR,#DPT18_12V
0593 12069A	703	LCALL	SBLOCK
	704		
0596 7408	705	MOV	A,#08H
0598 90F800	706	MOV	DPTR,#PQA
059B F0	707	MOVX	@DPTR,A
	708		
059C 900C69	709	MOV	DPTR,#DGETCH
059F 12069A	710	LCALL	SBLOCK
05A2 1206E0	711	LCALL	RBYTE
	712		
	713	;***** PIN 19 = 0 V *****	
05A5 1206CB	714	LCALL	SLF
05A8 900B84	715	MOV	DPTR,#DPT19_0V
05AB 12069A	716	LCALL	SBLOCK

05AE E4	718	CLR	A
05AF 90F800	719	MOV	DPTR,#PQA
05B2 F0	720	MOVX	@DPTR,A
	721		
05B3 900C69	722	MOV	DPTR,#DGETCH
05B6 12069A	723	LCALL	SBLOCK
05B9 1206E0	724	LCALL	RBYTE
	725		
	726	;***** PIN 19 = 12 V *****	
05BC 1206CB	727	LCALL	SLF
05BF 900B9D	728	MOV	DPTR,#DPT19_12V
05C2 12069A	729	LCALL	SBLOCK
	730		
05C5 7404	731	MOV	A,#04H
05C7 90F800	732	MOV	DPTR,#PQA
05CA F0	733	MOVX	@DPTR,A
	734		
05CB 900C69	735	MOV	DPTR,#DGETCH
05CE 12069A	736	LCALL	SBLOCK
05D1 1206E0	737	LCALL	RBYTE
	738		
	739	;***** PIN 20 = 0 V *****	
05D4 1206CB	740	LCALL	SLF
05D7 900BB7	741	MOV	DPTR,#DPT20_0V
05DA 12069A	742	LCALL	SBLOCK
	743		
05DD E4	744	CLR	A
05DE 90F800	745	MOV	DPTR,#PQA
05E1 F0	746	MOVX	@DPTR,A
	747		
05E2 900C69	748	MOV	DPTR,#DGETCH
05E5 12069A	749	LCALL	SBLOCK
05E8 1206E0	750	LCALL	RBYTE
	751		
	752	;***** PIN 20 = 12 V *****	
05EB 1206CB	753	LCALL	SLF
05EE 900BD0	754	MOV	DPTR,#DPT20_12V
05F1 12069A	755	LCALL	SBLOCK

	756	
05F4 7402	757	MOV A,#02H
05F6 90F800	758	MOV DPTR,#PQA
05F9 F0	759	MOVBX @DPTR,A
	760	
05FA 900C69	761	MOV DPTR,#DGETCH
05FD 12069A	762	LCALL SBLOCK
0600 1206E0	763	LCALL RBYTE
	764	
	765	;***** PIN 21 = 0 V *****
0603 1206CB	766	LCALL SLF
0606 900BEA	767	MOV DPTR,#DPT21_0V
0609 12069A	768	LCALL SBLOCK
	769	
060C E4	770	CLR A
060D 90F800	771	MOV DPTR,#PQA
0610 F0	772	MOVBX @DPTR,A
	773	
0611 900C69	774	MOV DPTR,#DGETCH
0614 12069A	775	LCALL SBLOCK
0617 1206E0	776	LCALL RBYTE
	777	
	778	;***** PIN 21 = 12 V *****
061A 1206CB	779	LCALL SLF
061D 900C03	780	MOV DPTR,#DPT21_12V
0620 12069A	781	LCALL SBLOCK
	782	
0623 7401	783	MOV A,#01H
0625 90F800	784	MOV DPTR,#PQA
0628 F0	785	MOVBX @DPTR,A
	786	
0629 900C69	787	MOV DPTR,#DGETCH
062C 12069A	788	LCALL SBLOCK
062F 1206E0	789	LCALL RBYTE
	790	
	791	;***** PIN 22 = 0 V *****

0632 1206CB

792

LCALL SLF

0635 900C1D

793

MOV DPTR,#DPT22_0V

0638 12069A	794	LCALL	SBLOCK
063B E4	795	CLR	A
063C 90F801	796	MOV	DPTR,#PGM
063F F0	797	MOVX	@DPTR,A
	798		
0640 900C69	799	MOV	DPTR,#DGETCH
0643 12069A	800	LCALL	SBLOCK
0646 1206E0	801	LCALL	RBYTE
	802		
	803	:***** PIN 22 = 5 V *****	
0649 1206CB	804	LCALL	SLF
064C 900C36	805	MOV	DPTR,#DPT22_5V
064F 12069A	806	LCALL	SBLOCK
0652 74C4	807	MOV	A,#11000100B
0654 90F801	808	MOV	DPTR,#PGM
0657 F0	809	MOVX	@DPTR,A
	810		
0658 900C69	811	MOV	DPTR,#DGETCH
065B 12069A	812	LCALL	SBLOCK
065E 1206E0	813	LCALL	RBYTE
	814		
	815	:***** PIN 22 = 12 V *****	
0661 1206CB	816	LCALL	SLF
0664 900C4F	817	MOV	DPTR,#DPT22_12V
0667 12069A	818	LCALL	SBLOCK
066A 74C8	819	MOV	A,#11001000B
066C 90F801	820	MOV	DPTR,#PGM
066F F0	821	MOVX	@DPTR,A
	822		
0670 900C69	823	MOV	DPTR,#DGETCH
0673 12069A	824	LCALL	SBLOCK
0676 1206E0	825	LCALL	RBYTE
	826		
0679 74C0	827	MOV	A,#11000000B
067B 90F801	828	MOV	DPTR,#PGM
067E F0	829	MOVX	@DPTR,A
067F 22	830	RET	
	831		

```

832: ***** CLRPGM SUM *****
833: CLEAR PORT AFTER PROGRAM DEVICE
834: IN NO REG
835: REG A,DPTR
836
0680 7480      837  CLRPGM: MOV    A,#80H
0682 90F802     838          MOV    DPTR,#P_CON
0685 F0         839          MOVX   @DPTR,A
840
0686 74C0      841          MOV    A,#0C0H
0688 90F801     842          MOV    DPTR,#PGM
068B F0         843          MOVX   @DPTR,A
844
068C E4        845          CLR    A
068D 90F800     846          MOV    DPTR,#PQA
0690 F0         847          MOVX   @DPTR,A
848
0691 90FC00     849          MOV    DPTR,#PI5V
0694 F0         850          MOVX   @DPTR,A
851
0695 90FC01     852          MOV    DPTR,#PI12V
0698 F0         853          MOVX   @DPTR,A
0699 22        854          RET
855
856
857 : ***** SBLOCK SUB *****
858 : SEND BLOCK
859 : IN  = DPTR ROM-ADDRESS (END BY 0 OR 0DH)
860 : OUT = DPTR (NEXT)
861 : REG = A,DPTR
862
069A E4        863  SBLOCK: CLR A
069B 93        864          MOVC   A,@A+DPTR
069C B40002     865          CJNE   A,#0,SBLOCK1
069F A3        866          INC    DPTR          ;NEXT ADDRESS
06A0 22        867          RET          ;EXIT BY 0
06A1 B40D05     868  SBLOCK1: CJNE   A,#0DH,SBLOCK2
06A4 1206CB     869          LCALL   SLF          ;CRLF

```

```

06A7 A3      870      INC      DPTR      ;NEXT ADDRESS
06A8 22      871      RET              ;EXIT BY ODH
06A9 1206D6  872  SBLOCK2: LCALL  SBYTE
06AC A3      873      INC      DPTR
06AD 80EB    874      SJMP     SBLOCK
875
876      ; ***** SBLANK SUB *****
877      ; SEND BLANK
878      ; IN  = R2 (NO OF BLANK)
879      ; REG = A,R2
880
06AF 7420    881  SBLANK: MOV     A,#' '
06B1 1206D6  882      LCALL  SBYTE
06B4 DAF9    883      DJNZ   R2,SBLANK
06B6 22      884      RET
885
886; ***** SPACE SUB *****
887; PRINT CR/LF = R1
888; IN  = NO REG
889; OUT = TO MONITOR
890; REG = R1,DPTR
891
06B7 C083    892  SPACE:  PUSH   DPH
06B9 C082    893      PUSH   DPL
06BB C001    894      PUSH   01
06BD 7912    895      MOV    R1,#18
06BF 1206CB  896  F:      LCALL  SLF
06C2 D9FB    897      DJNZ   R1,F
06C4 D001    898      POP     01
06C6 D082    899      POP     DPL
06C8 D083    900      POP     DPH
06CA 22      901      RET
902
903; ***** SLF SUB *****
904; PRINT CR/LF
905; REG A
906
06CB 740D    907  SLF:   MOV     A,#ODH

```

```

06CD 1206D6      908      LCALL  SBYTE
06D0 740A        909      MOV    A,#0AH
06D2 1206D6      910      LCALL  SBYTE
06D5 22          911      RET
                  912
913; ***** SBYTE SUB *****
914; SEND BYTE
915; IN = A
916; REG = NO
917
06D6 3099FD      918  SBYTE:  JNB    TI,$
06D9 C299        919      CLR    TI
06DB F599        920      MOV    SBUF,A
06DD F590        921      MOV    P1,A
06DF 22          922      RET
                  923
924 ; ***** RBYTE SUB *****
925 ; OUT = A
926 ; REG = A
927
06E0 3098FD      928  RBYTE:  JNB    RI,$
06E3 C298        929      CLR    RI
06E5 E599        930      MOV    A,SBUF
06E7 F590        931      MOV    P1,A
06E9 22          932      RET
                  933
06EA 50414C20    934  LO01:  DB "PAL Programmer V1.0 ",0DH
06EE 50726F67
06F2 72616D6D
06F6 65722056
06FA 312E3020
06FE 0D
06FF 4B4D4954    935  LO02:  DB "KMIT'L ED.Engineer 16.",0DH
0703 274C2045
0707 442E456E
070B 67696E65
070F 65722031
0713 362E0D

```

0716 31393935 936 LO03: DB "1995 Buad Rate :9600 ",ODH
 071A 20427561
 071E 64205261
 0722 7465203A
 0726 39363030
 072A 200D
 937
 072C 50726F67 938 DTEST: DB "Program test board PAL Program V1.0 ",ODH
 0730 72616D20
 0734 74657374
 0738 20626F61
 073C 72642050
 0740 414C2050
 0744 726F6772
 0748 616D2056
 074C 312E3020
 0750 0D
 0751 312E5445 939 DTEST1: DB "1.TEST RAM US 0000H - 0FFFFH ",ODH
 0755 53542052
 0759 414D2055
 075D 33203030
 0761 30304820
 0765 2D203046
 0769 46464648
 076D 200D
 076F 322E5445 940 DTEST2: DB "2.TEST RAM U4 8000H - 0FFFFH ",ODH
 0773 53542052
 0777 414D2055
 077B 34203830
 077F 30304820
 0783 2D203046
 0787 46464648
 078B 200D
 078D 332E5445 941 DTEST3: DB "3.TEST Vcc on borad ",ODH
 0791 53542056
 0795 6363206F
 0799 6E20626F
 079D 72616420

07A1 0D

942

07A2 474E4420 943 DPT3_0V: DB "GND TO PIN 3 VCC = 0 V ",0DH

07A6 544F2050

07AA 494E2033

07AE 20564343

07B2 203D2030

07B6 2056200D

07BA 474E4420 944 DPT3_5V: DB "GND TO PIN 3 VCC = 5 V ",0DH

07BE 544F2050

07C2 494E2033

07C6 20564343

07CA 203D2035

07CE 2056200D

07D2 474E4420 945 DPT3_12V: DB "GND TO PIN 3 VCC = 12 V ",0DH

07D6 544F2050

07DA 494E2033

07DE 20564343

07E2 203D2031

07E6 32205620

07EA 0D

946

07EB 474E4420 947 DPT4_0V: DB "GND TO PIN 4 VCC = 0 V ",0DH

07EF 544F2050

07F3 494E2034

07F7 20564343

07FB 203D2030

07FF 2056200D

0803 474E4420 948 DPT4_5V: DB "GND TO PIN 4 VCC = 5 V ",0DH

0807 544F2050

080B 494E2034

080F 20564343

0813 203D2035

0817 2056200D

081B 474E4420 949 DPT4_12V: DB "GND TO PIN 4 VCC = 12 V ",0DH

081F 544F2050

0823 494E2034

0827 20564343

082B 203D2031
082F 32205620
0833 0D

950

0834 474E4420 951 DPT5_0V: DB "GND TO PIN 5 VCC = 0 V ",0DH

0838 544F2050
083C 494E2035
0840 20564343

0844 203D2030
0848 2056200D

084C 474E4420 952 DPT5_5V: DB "GND TO PIN 5 VCC = 5 V ",0DH

0850 544F2050
0854 494E2035
0858 20564343

085C 203D2035
0860 2056200D

0864 474E4420 953 DPT5_12V: DB "GND TO PIN 5 VCC = 12 V ",0DH

0868 544F2050
086C 494E2035
0870 20564343

0874 203D2031
0878 32205620
087C 0D

954

087D 474E4420 955 DPT6_0V: DB "GND TO PIN 6 VCC = 0 V ",0DH

0881 544F2050
0885 494E2036
0889 20564343

088D 203D2030
0891 2056200D

0895 474E4420 956 DPT6_5V: DB "GND TO PIN 6 VCC = 5 V ",0DH

0899 544F2050
089D 494E2036
08A1 20564343

08A5 203D2035
08A9 2056200D

08AD 474E4420 957 DPT6_12V: DB "GND TO PIN 6 VCC = 12 V ",0DH

08B1 544F2050

08B5 494E2036

08B9 20564343

08BD 203D2031

08C1 32205620

08C5 0D

958

08C6 474E4420 959 DPT7_0V: DB "GND TO PIN 7 VCC = 0 V ",0DH

08CA 544F2050

08CE 494E2037

08D2 20564343

08D6 203D2030

08DA 2056200D

08DE 474E4420 960 DPT7_5V: DB "GND TO PIN 7 VCC = 5 V ",0DH

08E2 544F2050

08E6 494E2037

08EA 20564343

08EE 203D2035

08F2 2056200D

08F6 474E4420 961 DPT7_12V: DB "GND TO PIN 7 VCC = 12 V ",0DH

08FA 544F2050

08FE 494E2037

0902 20564343

0906 203D2031

090A 32205620

090E 0D

962

090F 474E4420 963 DPT8_0V: DB "GND TO PIN 8 VCC = 0 V ",0DH

0913 544F2050

0917 494E2038

091B 20564343

091F 203D2030

0923 2056200D

0927 474E4420 964 DPT8_5V: DB "GND TO PIN 8 VCC = 5 V ",0DH

092B 544F2050

092F 494E2038

0933 20564343

0937 203D2035

093B 2056200D

093F 474E4420 965 DPT8_12V: DB "GND TO PIN 8 VCC = -12 V ",0DH
 0943 544F2050
 0947 494E2038
 094B 20564343
 094F 203D2031
 0953 32205620
 0957 0D

966

0958 474E4420 967 DPT9_0V: DB "GND TO PIN 9 VCC = 0 V ",0DH
 095C 544F2050
 0960 494E2039
 0964 20564343
 0968 203D2030
 096C 2056200D

0970 474E4420 968 DPT9_5V: DB "GND TO PIN 9 VCC = 5 V ",0DH
 0974 544F2050
 0978 494E2039
 097C 20564343
 0980 203D2035
 0984 2056200D

0988 474E4420 969 DPT9_12V: DB "GND TO PIN 9 VCC = 12 V ",0DH
 098C 544F2050
 0990 494E2039
 0994 20564343
 0998 203D2031
 099C 32205620
 09A0 0D

970

09A1 474E4420 971 DPT10_0V: DB "GND TO PIN 10 VCC = 0 V ",0DH
 09A5 544F2050
 09A9 494E2031
 09AD 30205643
 09B1 43203D20
 09B5 30205620
 09B9 0D

09BA 474E4420 972 DPT10_5V: DB "GND TO PIN 10 VCC = 5 V ",0DH

09BE 544F2050

09C2 494E2031

09C6 30205643

09CA 43203D20

09CE 35205620

09D2 0D

09D3 474E4420 973 DPT10_12V: DB "GND TO PIN 10 VCC = 12 V ",0DH

09D7 544F2050

09DB 494E2031

09DF 30205643

09E3 43203D20

09E7 31322056

09EB 200D

974

09ED 474E4420 975 DPT11_0V: DB "GND TO PIN 11 VCC = 0 V ",0DH

09F1 544F2050

09F5 494E2031

09F9 31205643

09FD 43203D20

0A01 30205620

0A05 0D

0A06 474E4420 976 DPT11_5V: DB "GND TO PIN 11 VCC = 5 V ",0DH

0A0A 544F2050

0A0E 494E2031

0A12 31205643

0A16 43203D20

0A1A 35205620

0A1E 0D

0A1F 474E4420 977 DPT11_12V: DB "GND TO PIN 11 VCC = 12 V ",0DH

0A23 544F2050

0A27 494E2031

0A2B 31205643

0A2F 43203D20

0A33 31322056

0A37 200D

978

0A39 474E4420 979 DPT13_0V: DB "GND TO PIN 13 VCC = 0 V ",0DH

0A3D 544F2050

0A41 494E2031

0A45 33205643

0A49 43203D20

0A4D 30205620

0A51 0D

0A52 474E4420 980 DPT13_5V: DB "GND TO PIN 13 VCC = 5 V ",0DH

0A56 544F2050

0A5A 494E2031

0A5E 33205643

0A62 43203D20

0A66 35205620

0A6A 0D

0A6B 474E4420 981 DPT13_12V: DB "GND TO PIN 13 VCC = 12 V ",0DH

0A6F 544F2050

0A73 494E2031

0A77 33205643

0A7B 43203D20

0A7F 31322056

0A83 200D

982

0A85 474E4420 983 DPT14_0V: DB "GND TO PIN 14 VCC = 0 V ",0DH ;Q7

0A89 544F2050

0A8D 494E2031

0A91 34205643

0A95 43203D20

0A99 30205620

0A9D 0D

0A9E 474E4420 984 DPT14_12V: DB "GND TO PIN 14 VCC = 12 V ",0DH

0AA2 544F2050

0AA6 494E2031

0AAA 34205643

0AAE 43203D20

0AB2 31322056

0AB6 200D

985

0AB8 474E4420 986 DPT15_0V: DB "GND TO PIN 15 VCC = 0 V ",0DH ;Q6

0ABC 544F2050

0AC0 494E2031

0AC4 35205643

0AC8 43203D20

0ACC 30205620

0AD0 0D

0AD1 474E4420 987 DPT15_12V: DB "GND TO PIN 15 VCC - 12 V ",0DH

0AD5 544F2050

0AD9 494E2031

0ADD 35205643

0AE1 43203D20

0AE5 31322056

0AE9 200D

988

0AEB 474E4420 989 DPT16_0V: DB "GND TO PIN 16 VCC - 0 V ",0DH ;Q5

0AEF 544F2050

0AF3 494E2031

0AF7 36205643

0AFB 43203D20

0AFF 30205620

0B03 0D

0B04 474E4420 990 DPT16_12V: DB "GND TO PIN 16 VCC - 12 V ",0DH

0B08 544F2050

0B0C 494E2031

0B10 36205643

0B14 43203D20

0B18 31322056

0B1C 200D

991

0B1E 474E4420 992 DPT17_0V: DB "GND TO PIN 17 VCC - 0 V ",0DH ;Q4

0B22 544F2050

0B26 494E2031

0B2A 37205643

0B2E 43203D20

0B32 30205620

0B36 0D

0B37 474E4420 993 DPT17_12V: DB "GND TO PIN 17 VCC - 12 V ",0DH

0B3B 544F2050

0B3F 494E2031

0B43 37205643

0B47 43203D20

0B4B 31322056

OB4F 200D

994

OB51 474E4420 995 DPT18_0V: DB "GND TO PIN 18 VCC = 0 V ",0DH ;Q3

OB55 544F2050

OB59 494E2031

OB5D 38205643

OB61 43203D20

OB65 30205620

OB69 0D

OB6A 474E4420 996 DPT18_12V: DB "GND TO PIN 18 VCC = 12 V ",0DH

OB6E 544F2050

OB72 494E2031

OB76 38205643

OB7A 43203D20

OB7E 31322056

OB82 200D

997

OB84 474E4420 998 DPT19_0V: DB "GND TO PIN 19 VCC = 0 V ",0DH ;Q2

OB88 544F2050

OB8C 494E2031

OB90 39205643

OB94 43203D20

OB98 30205620

OB9C 0D

OB9D 474E4420 999 DPT19_12V: DB "GND TO PIN 19 VCC = 12 V ",0DH

OBA1 544F2050

OBA5 494E2031

OBA9 39205643

OBAD 43203D20

OBB1 31322056

OBB5 200D

1000

OBB7 474E4420 1001 DPT20_0V: DB "GND TO PIN 20 VCC = 0 V ",0DH ;Q1

OBBB 544F2050

OBBF 494E2032

OBC3 30205643

OBC7 43203D20

OBCB 30205620

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

OBCF 0D

OBD0 474E4420 1002 DPT20_12V: DB "GND TO PIN 20 VCC = 12 V ",0DH

OBD4 544F2050

OBD8 494E2032

OBDC 30205643

OBE0 43203D20

OBE4 31322056

OBE8 200D

1003

OBEA 474E4420 1004 DPT21_0V: DB "GND TO PIN 21 VCC = 0 V ",0DH ;Q0

OBEE 544F2050

OBF2 494E2032

OBF6 31205643

OBFA 43203D20

OBFE 30205620

OC02 0D

OC03 474E4420 1005 DPT21_12V: DB "GND TO PIN 21 VCC = 12 V ",0DH

OC07 544F2050

OC0B 494E2032

OC0F 31205643

OC13 43203D20

OC17 31322056

OC1B 200D

1006

OC1D 474E4420 1007 DPT22_0V: DB "GND TO PIN 22 VCC = 0 V ",0DH ;VCC OF IC

OC21 544F2050

OC25 494E2032

OC29 32205643

OC2D 43203D20

OC31 30205620

OC35 0D

OC36 474E4420 1008 DPT22_5V: DB "GND TO PIN 22 VCC = 5 V ",0DH ;VCC OF IC

OC3A 544F2050

OC3E 494E2032

OC42 32205643

OC46 43203D20

OC4A 35205620

เอกสารนี้เป็นเอกสารตัวอย่างไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

0C4F 474E4420 1009 DPT22_12V: DB "GND TO PIN 22 VCC = 12 V ",0DH ;VCC OF IC

0C53 544F2050

0C57 494E2032

0C5B 32205643

0C5F 43203D20

0C63 31322056

0C67 200D

1010

0C69 50726573 1011 DGETCH: DB "Press any key ",0DH

0C6D 7320616E

0C71 79206B65

0C75 79200D

1012

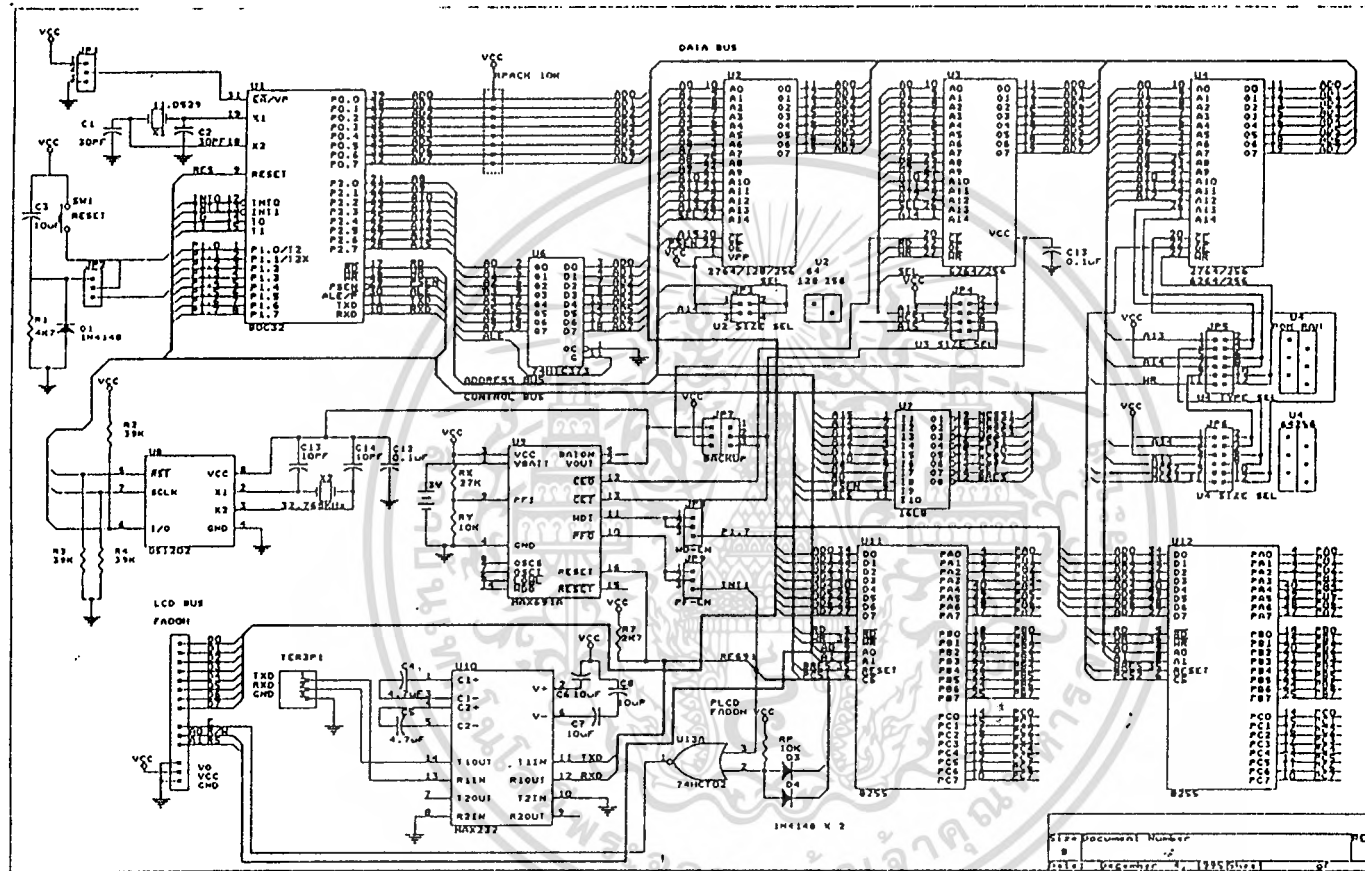
0000=

1013 END

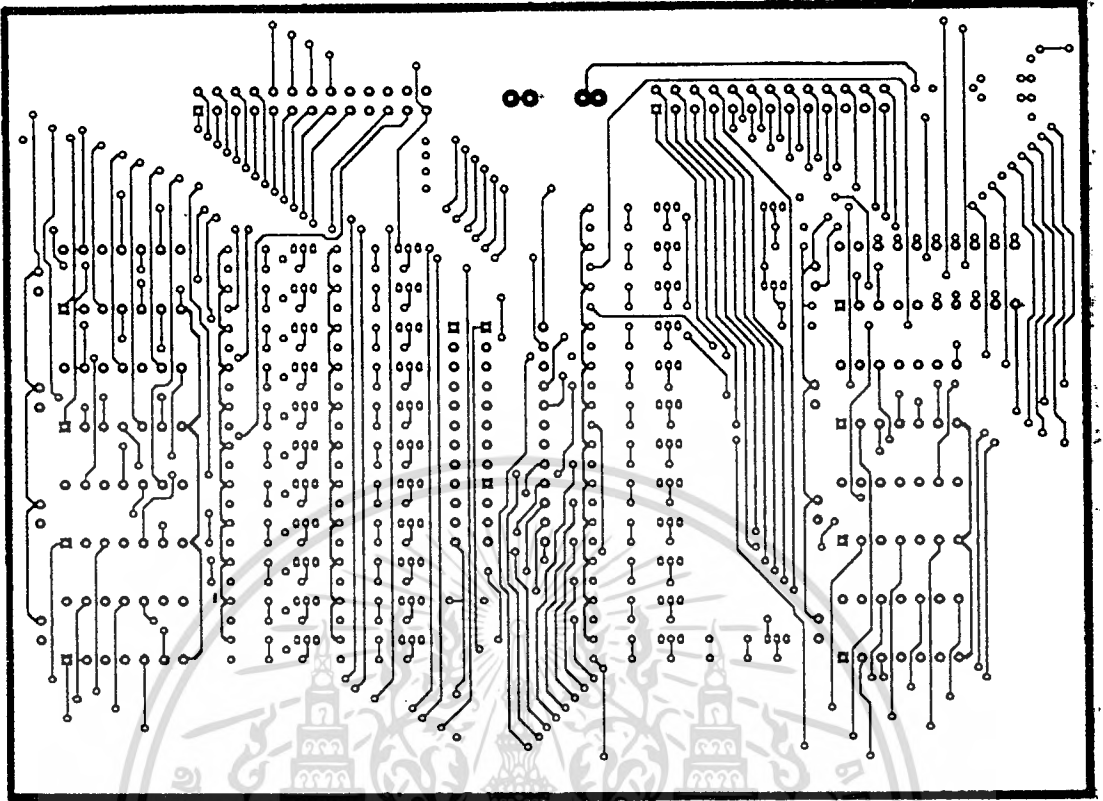
addixx = 0026	addiyy = 0028	ahigh = 0021	alow = 0022	buffer = 0000
buffer2 = 8000	clrpgrm = 0680	code_1 = 013A	code_2 = 0140	code_3 = 0146
codefuse = 0025	conf_1 = FC03	conf_2 = F803	dgetch = 0C69	dpt10_0v = 09A1
pt10_12v = 09D3	dpt10_5v = 09BA	dpt11_0v = 09ED	dpt11_12v = 0A1F	dpt11_5v = 0A06
dpt13_0v = 0A39	dpt13_12v = 0A6B	dpt13_5v = 0A52	dpt14_0v = 0A85	dpt14_12v = 0A9E
dpt15_0v = 0AB8	dpt15_12v = 0AD1	dpt16_0v = 0AEB	dpt16_12v = 0B04	dpt17_0v = 0B1E
dpt17_12v = 0B37	dpt18_0v = 0B51	dpt18_12v = 0B6A	dpt19_0v = 0B84	dpt19_12v = 0B9D
dpt20_0v = 0BB7	dpt20_12v = 0BD0	dpt21_0v = 0BEA	dpt21_12v = 0C03	dpt22_0v = 0C1D
dpt22_12v = 0C4F	dpt22_5v = 0C36	dpt3_0v = 07A2	dpt3_12v = 07D2	dpt3_5v = 07BA
dpt4_0v = 07EB	dpt4_12v = 081B	dpt4_5v = 0803	dpt5_0v = 0834	dpt5_12v = 0864
dpt5_5v = 084C	dpt6_0v = 087D	dpt6_12v = 08AD	dpt6_5v = 0895	dpt7_0v = 08C6
dpt7_12v = 08F6	dpt7_5v = 08DE	dpt8_0v = 090F	dpt8_12v = 093F	dpt8_5v = 0927
dpt9_0v = 0958	dpt9_12v = 0988	dpt9_5v = 0970	dtest = 072C	dtest1 = 0751
dtest2 = 076F	dtest3 = 078D	f = 06BF	lo01 = 06EA	lo02 = 06FF
lo03 = 0716	lr = 0020	menu = 014F	p_con = F802	pdata = FC02
pgm = F801	pi12v = FC01	pi5v = FC00	pqa = F800	rbyte = 06E0
res = 0100	res1 = 0102	rhigh = 0023	rlow = 0024	sblank = 06AF
sblock = 069A	sblock1 = 06A1	sblock2 = 06A9	sbyte = 06D6	slf = 06CB
space = 06B7	stack = 0060	tpin = 0178	wait = 0125	x = 002A
y = 002B				



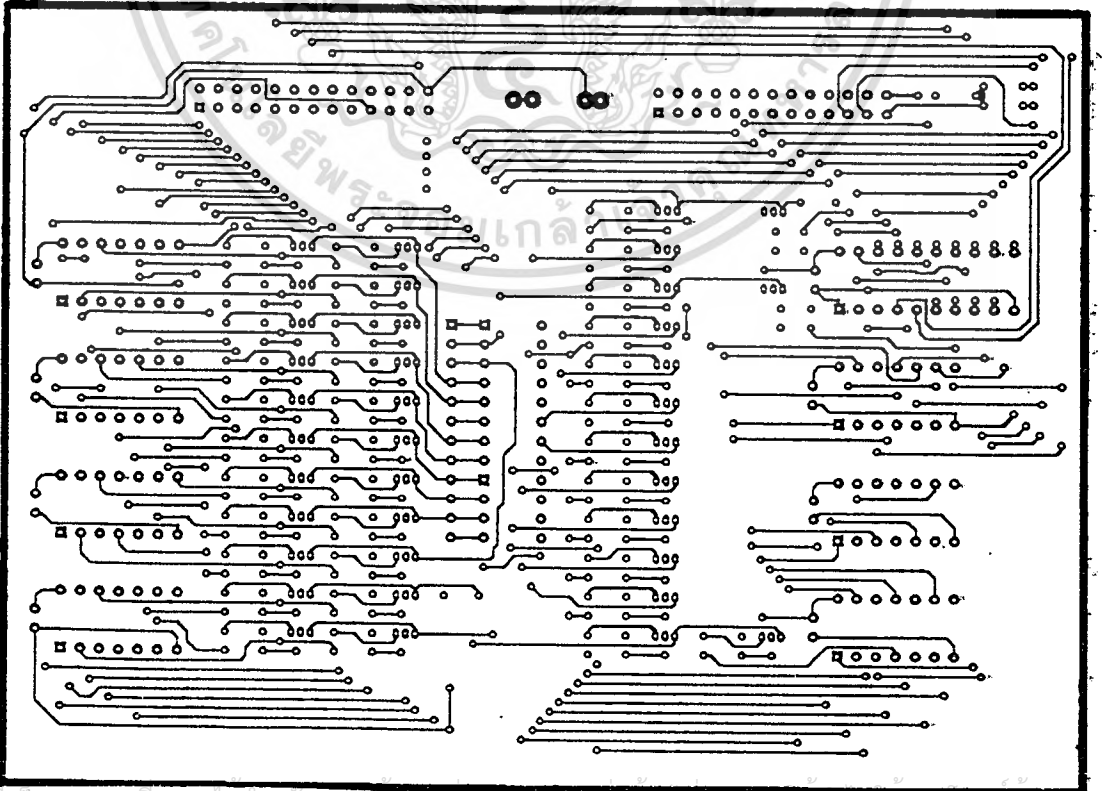
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปวงจรบอร์ดควบคุมเครื่องโปรแกรม PAL



รูปลายวงจรพิมพ์เครื่องโปรแกรม PAL ด้านบน



รูปลายวงจรพิมพ์เครื่องโปรแกรม PAL ด้านล่าง



รูปการวางอุปกรณ์เครื่องโปรแกรม PAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม



เจนวิทย์ เหลืองอร่าม, พ.อ. การใช้ Turbo C++ เขียนโปรแกรมด้วยภาษา C .
กรุงเทพมหานคร : สำนักพิมพ์สุขภาพใจ .

บริษัทซิลาร์เสิร์ช จำกัด . DMCxxx USER'S MANUAL DOT MATRIX LCD
MODULE . กรุงเทพมหานคร .

บริษัทซิลาร์เสิร์ช จำกัด . ANT-32 VERSION 3.0 EMBEDDED CONTROL
BOARD . กรุงเทพมหานคร .

บวร ปัทสราทร, ประเสริฐ คันธมานนท์ และ สุเมธ อังคะศิริกุล . เทคโนโลยีการ
ออกแบบวงจรรวม . ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ ,
พิมพ์ครั้งที่ 1, 2533

วิภา แสงพิสิตี . “ระบบการพัฒนา PAL.” วิทยานิพนธ์ปริญญาโท สาขา
วิศวกรรมไฟฟ้า บัณฑิตวิทยาลัย จุฬาลงกรณ์มหาวิทยาลัย, 2537

สมพันธ์ -ชาณูศิลป์ . ภาษาซี . หน่วยงานสารบรรณงานบริหารและธุรการ คณะ
วิศวกรรมศาสตร์ มหาวิทยาลัยขอนแก่น , พิมพ์ครั้งที่ 3 , 2537

สัมฤทธิ์ แสนธิ . “อุปกรณ์ที่สามารถโปรแกรมได้.” อิเล็กทรอนิกส์เวิลด์ , ฉบับที่ 141
(2536), หน้า 78-96 .

“PAL ลอจิกเกตสารพัดนึก ” วารสารเซมิคอนดักเตอร์อิเล็กทรอนิกส์ , ฉบับที่ 83
(ธันวาคม 2530 - มกราคม 2531), หน้า 150 - 161 .

“Programmable Logic Device” อิเล็กทรอนิกส์เวิลด์ , ฉบับที่ 147 (2538), หน้า 56-65

ADVANCED MICRO DEVICES ,“Data Book .” PAL Device ,1992

Bostock Geoff. Programmable logic handbook ,2nd ,1993



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้