เครื่องควบคุมพลังงานไฟฟ้าภายในอาคารด้วย พีซี

POWER CONSUMPTION CONTROL Used PC

โดย

นายเศกสรรค์	ป๊อกคำอู๋	36.013215
นายสมศักดิ์	วงษ์ศรี	36.013218
นายสมศักดิ์	เชาวน์ชะตา	36.013220

วัน เดือน ปี 30 ก.ค. 2540
เลขทะเบียน 036935
เลขเรียกหนังสือ T38028 ก'846 ค

ปริญญานิพนธ์นี้ เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต
สาขาวิชาเทคโนโลยีอิเล็กทรอนิกส์
ภาควิชาเทคนิคอุตสาหกรรม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2538

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

036935

หัวข้อปริญญานิพนธ์	**เครื่องควบคุมพลังงานไฟฟ้าภายในอาคารด้วย พีซี**		
จัดทำโดย	นายเศกสรรค์	ป๊อกคำอู่	36.013215
	นายสมศักดิ์	วงษ์ศรี	36.013218
	นายสมศักดิ์	เชาวน์ชะตา	36.013220
สาขาวิชา	เทคโนโลยีอิเล็กทรอนิกส์อุตสาหกรรม		
อาจารย์ที่ปรึกษา	อาจารย์ไพศาล สิทธิโยภาสกุล		
	อาจารย์บุญชัย โชติวิริยวาณิชย์		
ปีการศึกษา	2538		

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้นับปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาอุตสาหกรรมศาสตร์บัณฑิต

คณะกรรมการสอบปริญญานิพนธ์

_____ ประธานกรรมการ
( )

_____ กรรมการ
( )

_____ กรรมการ
( )

_____ กรรมการ
( )

_____ กรรมการ
( )

_____
( )
หัวหน้าภาควิชา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องควบคุมพลังงานไฟฟ้าภายในอาคารด้วย พีซี

ผู้ร่วมงาน

นายเศกสรรค์	ป๊อกคำอู๋	36.013215
นายสมศักดิ์	วงษ์ศรี	36.013218
นายสมศักดิ์	เชาวน์ชะตา	36.013220

อาจารย์ที่ปรึกษา

อาจารย์ไพศาล สิทธิโยภาสกุล
อาจารย์บุญชัย โชติวิริยวาณิชย์

ปีการศึกษา 2538

บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้กล่าวถึง โครงงานที่เกี่ยวกับ การควบคุมพลังงานไฟฟ้าภายในอาคารด้วย เครื่องคอมพิวเตอร์พีซี (PC) โดยหลักในการควบคุมนี้ จะใช้เครื่องคอมพิวเตอร์ (Computer) เป็นตัวควบคุมแบบ Manual และใช้ตัวคอนโทรลเลอร์ (Controller) เป็นการควบคุมแบบ ออโต้เมติก (Automatic) โดยการควบคุมนี้จะควบคุมในระยะไกล ๆได้ โดยมีเครื่องส่ง ทรานสมิทเตอร์ (Transmitter) อยู่ส่วนหนึ่ง ทำหน้าที่ส่งสัญญาณควบคุม และจะมีเครื่องรับ รีซีฟเวอร์ (Receiver) หลาย ๆ ชุดต่อในระบบ เพื่อควบคุมการปิดเปิดเครื่องใช้ไฟฟ้าที่นำมาต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Power Consumption Control Used PC

Mr.Saksun	Pokcomou	36.013215
Mr.Somsak	Vongsri	36.013218
Mr.Somsak	Choachata	36.013220

Advisor

Mr.Paishan	Sitiyopasakul
Mr.Boonchai	Chotiviriyavanich

1995

ABSTRACT

In this project. We present. the power consumption control in building. by using PC. This equipment was control by controller for automatic control and also by Personal Computer for directed control. It has mamy receiver in system was control by controller in remote. The receiver directly control electric equipment by receive the command from the controller.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การจัดทำปริญาญานิพนธ์ฉบับนี้ ได้รับความกรุณาเอื้อเฟื้อจากบุคคลต่าง ๆ ในการให้คำแนะนำ ให้ข้อมูลและอำนวยความสะดวก ซึ่งได้แก่ อาจารย์ไพศาล สิทธิโยภาสกุล อาจารย์ประจำคณะวิศวกรรมศาสตร์ และอาจารย์ที่ปรึกษาในการจัดทำปริญญานิพนธ์ อาจารย์บุญชัย โชติวิริยวาณิชย์ อาจารย์ประจำคณะวิศวกรรมศาสตร์ และอาจารย์ที่ปรึกษาในการจัดทำปริญญานิพนธ์ คุณศักพงษ์ จันทร์ใบ วิศวกรฝ่าย Development แผนก NPDD บริษัท Centure Electronic and System จำกัด ช่วยให้คำแนะนำในการเขียนโปรแกรมบนคอมพิวเตอร์ คุณสุรชัย หาญอนุพงศ์ วิศวกรฝ่าย Development แผนก NPDD บริษัท Centure Electronic and System จำกัด ช่วยให้คำแนะนำทางด้านการสร้างวงจร

อนึ่งคุณความดีใด ๆ ที่เกิดจากปริญญานิพนธ์ฉบับนี้ ขอมอบให้แด่ บิดามารดา และครูบาอาจารย์ที่ได้ประสิทธิ์ประสาทวิชาความรู้ให้

กลุ่มผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	**1**
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	**2**
2.1 ทฤษฎีของส่วนควบคุม บอร์ด CP-Z84C11 PLUS	2
2.1.1 ซีพียู-Z84C11	2
2.1.2 ข้อมูลของ บอร์ด CP -Z84C11 PLUS	2
2.2 ทฤษฎีส่วนการใช้คำสั่งควบคุมผลการแสดงบน แอลซีดี (LCD)	14
2.3 ทฤษฎีส่วนที่ทำหน้าที่เข้ารหัส (ENCODER) และถอดรหัส (DECODER)	22
2.3.1 ส่วนวงจร เข้ารหัส (ENCODER)	22
2.3.2 ส่วนวงจร ถอดรหัส (DECODER)	25
2.4 ทฤษฎีส่วนของ แกล (GAL)	29
บทที่ 3 ขั้นตอนออกแบบและการสร้าง	**50**
3.1 ส่วนของ วงจร	50
3.1.1 ส่วนควบคุมอัตโนมัติ	50
3.1.2 ส่วนควบคุมปลายทาง	53
3.2 ส่วนของ โปรแกรม	57
3.2.1 การทำงานของโปรแกรมบนบอร์ด CP-Z84C11 PLUS	57
3.2.2 การทำงานของโปรแกรมติดต่อกับเครื่องคอมพิวเตอร์ พีซี	58
3.2.3 การใช้งานโปรแกรมติอต่อกับเครื่องคอมพิวเตอร์ พีซี	59
บทที่ 4 การทดลองและผลการทดลอง	**61**
4.1 ส่วนควบคุมปลายทาง	62
4.2 ส่วนควบคุมอัตโนมัติ	62
4.3 สรุปผลการทดลอง	71
ภาคผนวก	**72**
หนังสืออ้างอิง	**115**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ปัจจุบันเทคโนโลยีอิเล็กทรอนิกส์ ได้เข้ามามีบทบาทในการอำนวยความสดวกนานัปการให้กับมนุษย์ภายในห้องชุดพักอาศัยทั่วไป เช่น อพาร์ตเมนต์ คอนโดมิเนียม หอพัก โรงแรม และตามสถานที่ต่าง ๆ ก็มักมีการให้บริการต่าง ๆ กันไป ตั้งแต่มีจุดต่อสัญญาณโทรทัศน์ วิทยุ หรือทีวีวงจรปิด เสียงตามสายที่มีให้เลือกหลายช่อง และเครื่องปรับอากาศ โดยปริญญานิพนธ์นี้จึงถือกำเนิดขึ้นเพื่ออำนวยความสะดวกในการควบคุมซึ่งในส่วนของอุปกรณ์เครื่องใช้ไฟฟ้า หรืออุปกรณ์ภายในอาคารจะมีการเปิดปิดไฟฟ้าของอุปกรณ์แต่ละตัวด้วยมือ หรือบุคคล ซึ่งทำให้เกิดความไม่แน่นอน และความไม่สะดวกในการเปิดและปิดอุปกรณ์เหล่านั้นทุกอัน เช่น บางครั้งก็ลืมปิดอุปกรณ์ไฟฟ้าบางตัวหลังจากเลิกการทำงาน ซึ่งจะทำให้ต้องเสียค่าใช้จ่ายในด้านค่าไฟฟ้าโดยใช่เหตุ ส่วนในเรื่องของความสะดวก การเปิดและปิดสวิทช์ธรรมดา ดูแล้วก็ไม่เห็นว่าจะน่ารำคาญสักเท่าไร แต่ถ้าเราลดภาระส่วนนี้ไปได้ก็จะดี โดยผลักภาระนี้ไปให้กับโปรแกรม และฮาร์ดแวร์ที่ทำขึ้นเป็นผู้รับผิดชอบแทน โดยในส่วนของฮาร์ดแวร์นั้นจะประกอบด้วยส่วนใหญ่ ๆอยู่ 3 ส่วนด้วยกันคือ ส่วนของเครื่องคอมพิวเตอร์ พีซี (PC) ส่วนควบคุมอัตโนมัติ และส่วนของแบล็คบล็อค (Black Box) ซึ่งเป็นส่วนควบคุมปลายทาง ส่วนในเรื่องของการเลือกใช้ ส่วนประกอบของฮาร์ดแวร์ ในงานควบคุมชุดนี้ทางกลุ่มได้เลือกใช้ ไมโครโปรเซสเซอร์ # Z84C11 ซึ่งเป็นบอร์ดสำเร็จรูป และเหตุผลที่ต้องทำเป็น แบล็คบล๊อค แทนที่จะทำเป็นการ์ดเสียบลงไปในเครื่องคอมพิวเตอร์ พีซี ก็เพราะว่าถ้าทำเป็นการ์ดอินเตอร์เฟส (Interface) กับเครื่องคอมพิวเตอร์ พีซี โดยตรงจะทำให้ทำได้แค่สั่งงานหรือทำงานได้เฉพาะช่วงที่ เปิดใช้งาน เครื่องคอมพิวเตอร์ พีซี ไว้เท่านั้น แต่ถ้าทำเป็น แบล็คบล๊อค เมื่อสั่งจากเครื่องคอมพิวเตอร์ พีซี เสร็จแล้วก็สามารถทำงานได้ตามที่โปรแกรมกำหนดไว้ แม้ว่าจะปิด เครื่องคอมพิวเตอร์ พีซี แล้วก็ตาม เหตุผลอีกประการหนึ่งในการเลือกใช้ ไมโครโปรเซสเซอร์ (Microprocessor) # Z84C11 เพราะ ไมโครโปรเซสเซอร์ เบอร์นี้จะมีคำสั่ง และการใช้งานเหมือนกับ Z-80 ทุกประการ ซึ่งจะเป็นข้อดีที่ว่าถ้าจะเปลี่ยนไปใช้เป็น Z-80 เลยก็ได้โดยไม่ต้องเปลี่ยนซอร์ฟแวร์ (Software) เลย สาเหตุที่เลือกใช้ตระกูล Z-80 ก็จะเป็นตัวที่ได้รับความนิยมอย่างมากเหมือนกันเพราะมันมีราคาถูกกว่า ไมโครโปรเซสเซอร์ เบอร์อื่น ซึ่งจะเป็นตัวที่ทำให้ต้นทุนในการผลิตลดลง และระบบเครื่องควบคุมพลังงานไฟฟ้าภายในอาคารด้วยเครื่องคอมพิวเตอร์ พีซี นี้ เครื่องควบคุมไฟฟ้า จะทำหน้าที่ควบคุมอุปกรณ์ไฟฟ้าต่าง ๆ ภายในอาคารห้องพัก และรับสัญญาณจากอุปกรณ์ตรวจจับ พร้อมกับส่งสัญญาณไปควบคุมอุปกรณ์นั้น ๆ ส่วนสถานะของอุปกรณ์จะถูกแสดงที่ เครื่อวคอมพิวเตอร์ พีซี เพื่อแสดงสถานะต่าง ๆ ของอุปกรณ์ต่าง ๆ ที่ต่อใช้งานอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีที่เกี่ยวข้อง

2.1 ทฤษฎีของส่วนควบคุม บอร์ด CP-Z84C11 PLUS

2.1.1 ซีพียู-Z84C11

บอร์ด CP-Z84C11 PLUS เพิ่มในส่วนของวงจร RTC (Real Time Clock) เข้าไปอีก เป็นบอร์ดที่ใช้ CPU Z84C11 ของบริษัท ZILOG มาเป็น CPU ประจำบอร์ด RUN ที่ SPEED 10 MHz CPU Z84C11 นี้ก็คือ CPU ที่รวบรวมเอาชิพ วงจรต่าง ๆ ของ ZILOG เข้าด้วยกันคือ:-

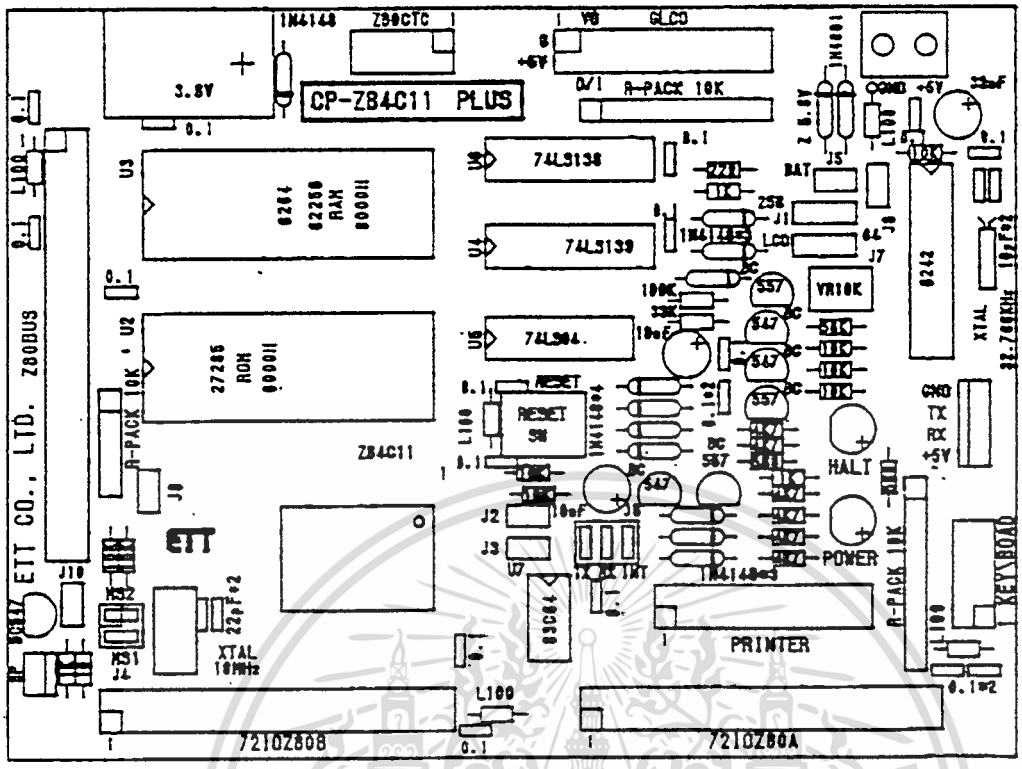
Z84C00	เป็น CPU Z80 แบบ CMOS RUN ที่ 10 MHz
Z84C30	เป็น Z80 CTC แบบ CMOS RUN ที่ 10 MHz
CGC	เป็น CLOCK GENERATOR CONTROL CIRCUIT
WDT	เป็น วงจร WATCH DOG TIMER
POWER ON RESET	เป็น วงจร RESET CPU เมื่อ VOIT VCC ต่ำกว่า 2.2V
40 BIT PARALLEL PORT	เป็น PORT 8 BIT จำนวน 5 PORT ใช้งาน

2.1.2 ข้อมูลของ บอร์ด CP-Z84C11 PLUS

ซีพียู-Z84C11

Z84C11 ของบริษัท ZILOG โดยเป็น HIGH SPEED OPERATION โดยมีให้เลือก 2 รุ่น คือ RUN ที่ความถี่ 6 MHz ,และที่ RUN 10 MHz ในบอร์ดนี้เราใช้ CPU รุ่น RUN 10 MHz CPU เบอร์นี้ของ ZILOG จะเป็นแบบ CMOS ทำให้กินกระแสต่ำมากในขณะที่เรา SET ให้อยู่ใน STOP MODE จะกินกระแสเพียง 50 μA ข้อดีอีกอย่างหนึ่งของ CPU ตัวนี้ก็คือ ถึงแม้จะ RUN 10 MHz ก็ตามเราก็สามารถ SET ให้ RUN เพียงครึ่งเดียวก็ได้คือ 5 MHz ในกรณีที่ใช้ RAM หรือ ROM ที่มี ACCESS TIME ต่ำๆ โดยเมื่อ POWER ON แล้ว CPU จะ RUN เพียง 5 MHz เท่านั้นตอนเริ่มต้นและเมื่อเราต้องการ RUN 10 MHz ก็สามารถ SET ได้ด้วยโปรแกรม รูปแบบลักษณะของ CPU จะเป็น PACKAGE แบบ 100 PIN QFP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1 แสดงส่วนของ บอร์ด CP-Z84C11 PLUS

หน่วยความจำ (MEMORY)

CP-Z84C11 สามารถใส่หน่วยความจำได้สูงสุด 64 KBYTE โดยแบ่งเป็น

SOCKET 1 EPROM สามารถใส่ EPROM ขนาด 32 KBYTE โดยแบ่งเป็นเบอร์ 27256 มีหน่วยความจำเริ่มจาก ADDRESS 0000H ถึง ADDRESS 7FFFFH

SOCKET 2 RAM สามารถใส่ RAMขนาด 8-32 KBYTE ได้โดยเป็นเบอร์ 6264 หรือ 62256 มีหน่วยความจำเริ่มจาก ADDRESS 80000H ถึง FFFFH โดย SET

ตำแหน่ง JUMPER ในการกำหนดเบอร์ไอซี ดังรูป :-

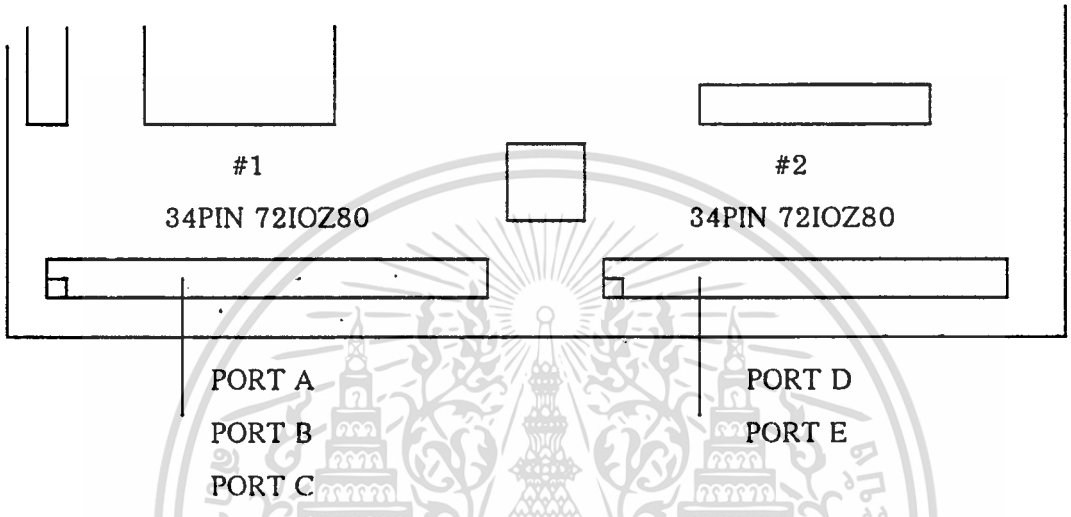
รูปที่ 2 แสดงการเลือก หน่วยความจำ ของ Z84C11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำ RAM ส่วนนี้เราสามารถต่อ BATTERY 3.6V ใช้ BACKUP ข้อมูลของหน่วยความจำนี้ได้ด้วย

พอร์ท (PORT)

CP-Z84C11 จะมี PORT ใช้งาน 40 BIT I/O หรือ 5 PORT ด้วยกัน โดย 5 PORT นี้จะเป็น PORT ในตัว CPU โดยจะมีตำแหน่งต่อออกมาดังรูป :-



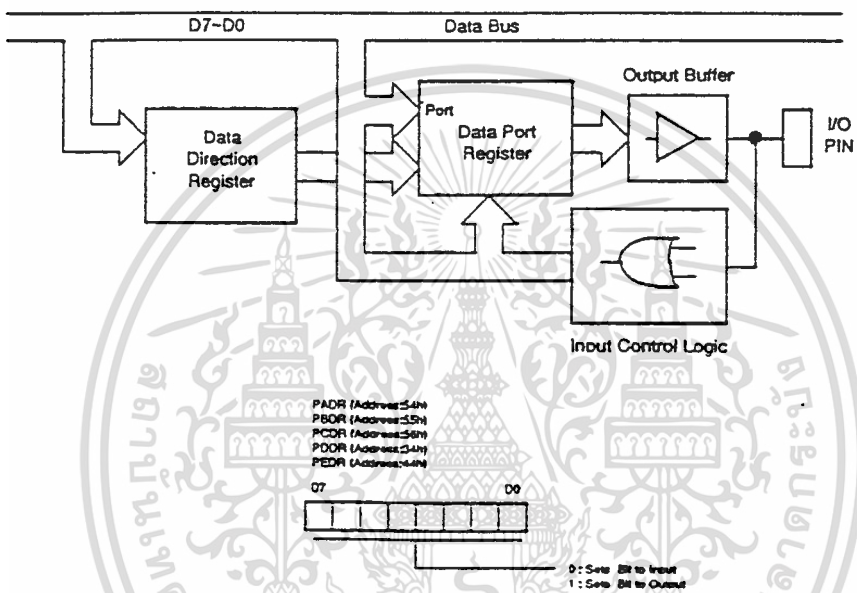
รูปที่ 3 แสดง PORT ใช้งานของ Z84C11

PORT ที่ต่อออกมานี้จะอยู่เป็น CONNECTOR 34 PIN (72IOZ80) สามารถต่อร่วมใช้กับอุปกรณ์สนับสนุนต่าง ๆ ของทาง อีทีที ได้มากมาย เช่น ชุด ET-SSRAC,ชุด ET-AD เป็นต้น

CHANNEL PORT	ADDRESS PORT
PORT A DATA PORT	50 H
PORT B DATA PORT	51 H
PORT C DATA PORT	52 H
PORT D DATA PORT	30 H
PORT E DATA PORT	40 H
CHANNEL CONTROL PORT	**ADDRESS PORT**
PORT A DATA DIRECTION REGISTER	54 H
PORT B DATA DIRECTION REGISTER	55 H
PORT C DATA DIRECTION REGISTER	56 H
PORT D DATA DIRECTION REGISTER	34 H
PORT E DATA DIRECTION REGISTER	44 H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถสั่งให้ PORT ของ Z84C11 เป็น INPUT หรือ OUTPUT ได้โดยอิสระ BIT ต่อ BIT โดยถ้าเรา SET ค่าออกที่ PORT DATA DIRECTION REGISTER ถ้าให้ BIT ใดเป็น "1" ก็คือให้ PORT DATA ของ BIT นั้น ๆ เป็น OUTPUT PORT (ถ้า SET PORT นั้นเป็น OUTPUT PORT: ค่าเริ่มต้นจะเป็นศูนย์) และถ้าเราต้องการให้เป็น INPUT PORT ก็ SET ค่าออกที่ PORT DATA DIRECTION REGISTER เป็นค่า " 0 "

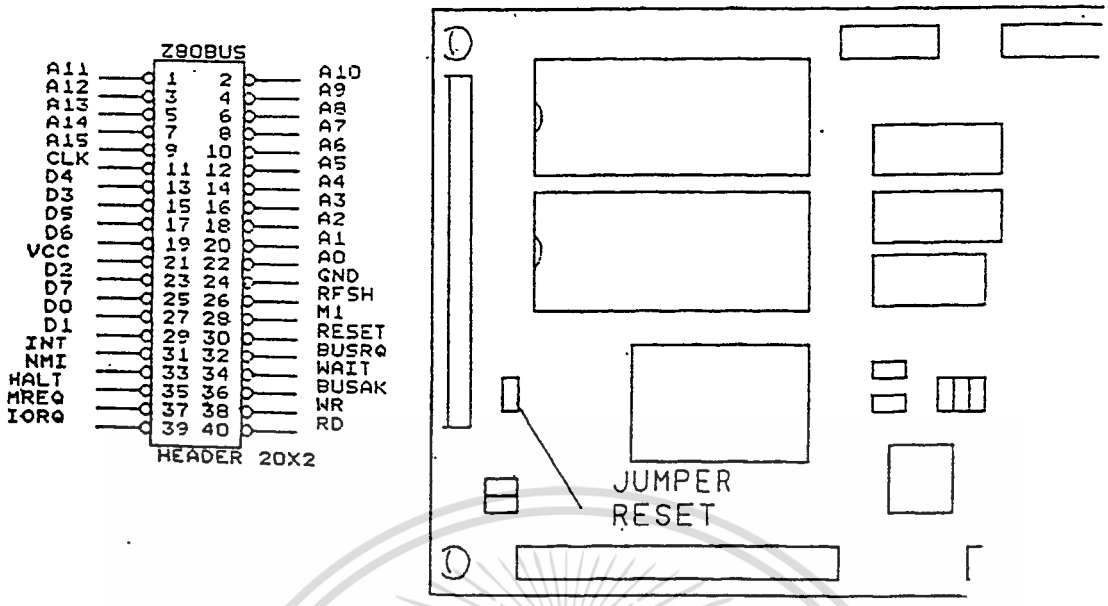


รูปที่ 4 แสดง PORT DIRECTION REGISTER

Z80 BUS

CP-Z84C11 จะมีส่วนขยายระบบได้ทาง Z80 BUS โดยมีลักษณะขาเหมือนกัน CPU Z80 40PIN ทำให้เราสามารถต่อขยายบอร์ดได้เช่น ต่อขยาย PORT 8255 ได้อีกโดยใช้บอร์ด 72IOZ80 เป็นต้น มีข้อพิเศษเนื่องจาก CPU Z84C11 นี้มีส่วนวงจร POWER ON RESET ในตัวอยู่แล้วไม่จำเป็นต้องใช้วงจร POWERON RESET ต่อจากภายนอก เช่น CPU เบอร์อื่น ๆ ฉะนั้นถ้าเราไปต่อร่วมกัน Z80 BUS อื่น ๆ ที่มี R,C ต่ออยู่แล้วอาจจะทำให้ CPU Z84C11 นี้อยู่ในสภาพ RESET ตลอดเวลาได้บอร์ด CP-Z84C11 จึงมี JUMPER ในการเลือกไม่ให้ต่อ RESET จากขาภายนอกได้ ดังรูป โดยถ้ามีการต่อ R , C ที่ RESET PIN ภายนอกก็ให้ถอด JUMPER RESET ออก

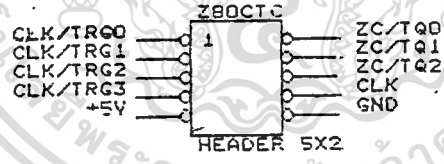
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

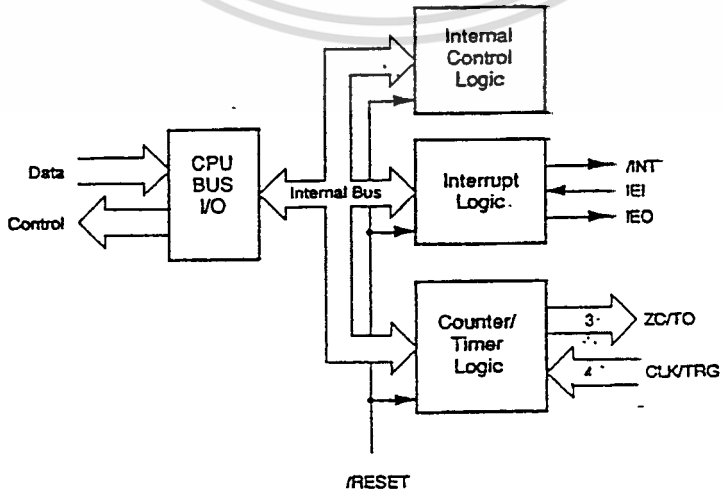


รูปที่ 5 แสดง Z80 BUS

COUNTER TIMER CHIP (CTC)

CP-Z84C11 จะมีวงจร Z84C30 (Z80 CTC) ต่อร่วมกันอยู่ในตัวเรียบร้อยแล้วโดยบอร์ด CP-Z84C11 จะต่อขาใช้งานของ CTC ออกมาที่ CONNECTOR 10 PIN แล้วดังรูป ส่วนขา INT ของ CTC นั้นจะต่อกับขา INT ของ Z80 ในตัวโดยเป็นแบบ WIRED-OR เรียบร้อยในตัวแล้ว รายละเอียดของ CTC ในบอร์ดนี้จะเหมือนกับ Z80 CTC ทั่วไป





รูปที่ 6 แสดง CTC BLOCK DIAGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CHANNEL	CTC ADDRESS PORT ADDRESS PORT
CH 0	10 H
CH 1	11 H
CH 2	12 H
CH 3	13 H

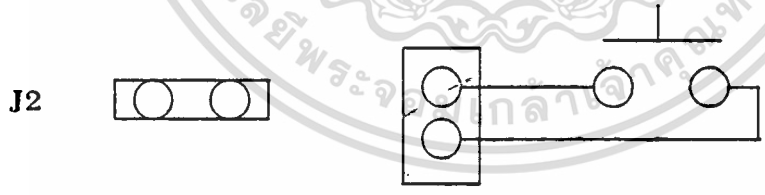
MS , MS2 (J4)

MS1 , MS2 จะเป็นขา INPUT ของ CPU Z84C11 โดยเป็นการ SET ให้ CPU ถ้าอยู่ในคำสั่ง HALT แล้วจะให้อยู่ในสภาพใด (RUN, IDLE1, IDEL2, STOP)

MS1	MS2	HALT STATE
1	1	RUN MODE
0	0	IDEL1 MODE
0	1	IDEL2 MODE
1	0	STOP MODE

RESET (J2)

เป็น INPUT ต่อเข้ากับขา RESET CPU Z84C11 ห้ามต่อกับวงจรประเภท R , C RESET



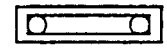
รูปที่ 7 แสดงการ RESET ของ Z84C11

WATCH DOG TIMER OUTPUT (WDTOUT J3)

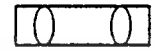
เป็น PIN จาก PORT PE BIT 7 ซึ่งทำ หน้าที่ 2 อย่างคือเป็น DATA PORT ทั่ว ๆ ไป แล้วก็ยังเป็น WATCH DOG TIMER OUTPUT ด้วย โดยจะเป็น OPEN-DRAIN I/O เมื่อถูก SET ให้เป็น WDTOUTโดย PIN นี้จะถูกต่อโดยตรงเข้ากับ PIN RESET ของ CPU โดยปกติถ้าเราไม่มี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การ SET WDTOUT แล้วให้กด JUMPER ออกห้ามต่อถ้าไม่มีการใช้WDTOUT เพราะเมื่อเราใช้ PORT PE 7 จะทำให้ CPU ถูก RESET ตามคำสั่ง OUT PORT PE 7 นั้นด้วย

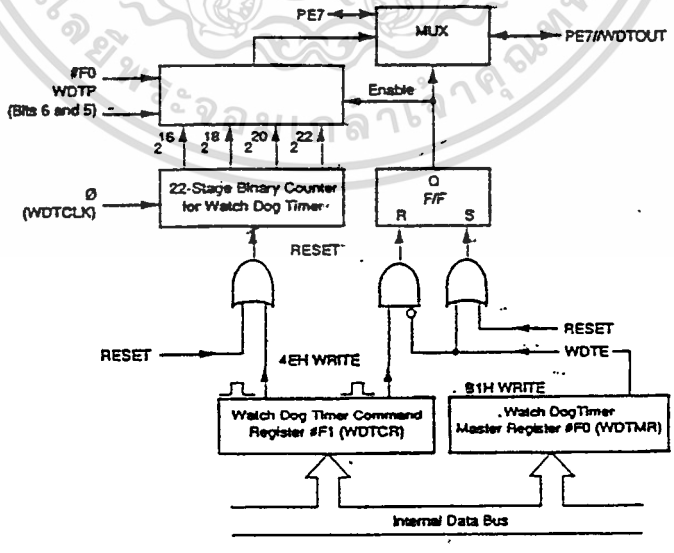

SET USER WDOUT

J3 WDTOUT


SET USER PORT

รูปที่ 8 แสดง WATCH DOG TIMER OUTPUT (WDTOUT)

(* WATCH DOG เป็นลักษณะวงจรที่จะทำการ RESET CPU อยู่เสมอตามค่าเวลาที่เรากำหนด ซึ่งถ้าเราไม่ทำการ DISABLE WATCH DOG ภายในเวลากำหนด CPU นั้นก็จะถูก RESET เช่นในโปรแกรมทำงานปกติเราจะ CALL DISABLE WATCH DOG อยู่เสมอ แต่ถ้า CPU กำลัง RUN อยู่นั้นเกิดมีสัญญาณรบกวนขึ้น ทำให้ไม่อาจสามารถมา RUN โปรแกรมปกติที่มีการเรียกใช้โปรแกรม CALL DISABLE WATCH DOG ได้ CPU ก็จะเกิดการ RESET ขึ้นทันทีเพื่อให้กลับไปเริ่ม RUN ใหม่อีก)

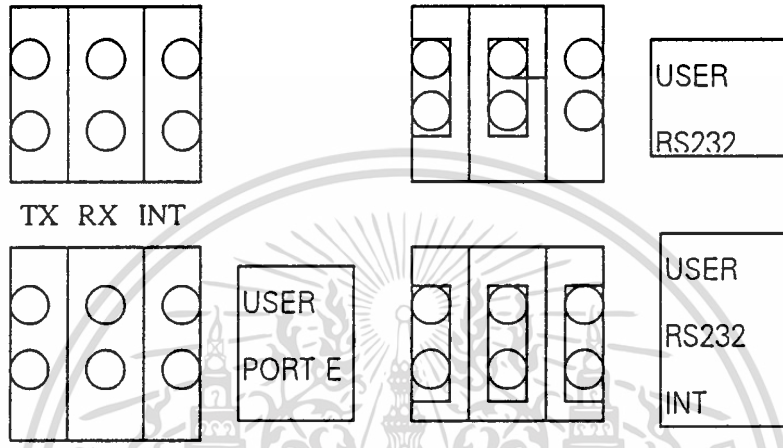


รูปที่ 9 แสดง WATCH DOG TIMER BLOCK DIAGRAM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TX , RX , INT (J6)

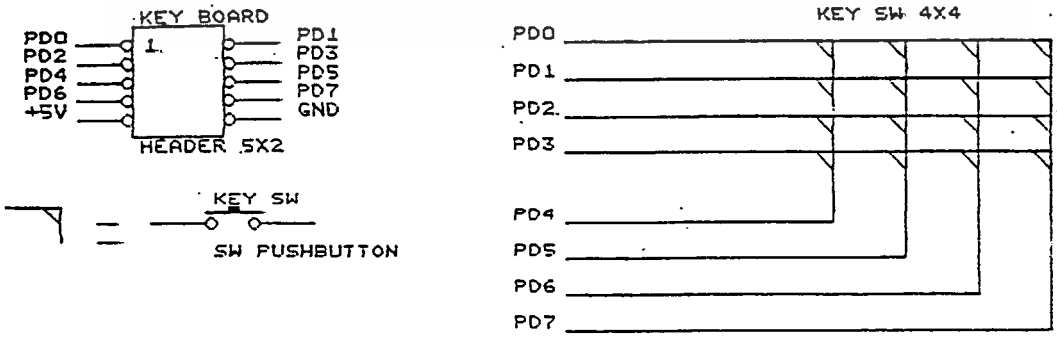
เป็น CONNECTOR 6 PIN ที่ต่อจากวงจร RS-232 ที่ใช้เปลี่ยนระดับสัญญาณต่อเข้าPORT PE 6 , PE 5 ,และเข้าขา INT ของ Z84C11 โดยถ้าเราไม่ใช้ PORT RS-232 ก็ใช้ถอด JUMPER ออก เพื่อจะใช้ PE6, PE5 ได้อย่างอิสระ และอีกส่วนหนึ่งคือ JUMPER ของ INT นั้นเราสามารถ SET ให้ PORT RS-232 รับข้อมูลในรูปแบบของขบวนการ INTERRUPT เช่นเดียวกับการ INTERRUPT จาก RS-232 PORT ได้



รูปที่ 10 แสดงการเลือกใช้งาน RS-232

KEYBOARD

เป็น CONNECTOR ขนาด 10 PIN ต่อจาก PORT PD โดยมี R-PULL-UP 10K ต่ออยู่ด้วยโดย CONNECTOR 10 PIN นี้จะสามารถต่อเป็นรูปแบบ KEY ขนาด 4X4 คือ 16 KEY โดยเขียนโปรแกรม SCAN KEY ขึ้นหรือจะนำไปใช้เป็น PORT อิสระในงานอื่น ๆ ก็ได้



รูปที่ 11 แสดงลักษณะของ KEYBOARD

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BAT ON/OFF (J5)

เป็น CONNECTOR 2 PIN โดยเราใช้ JUMPER ในการปิด ,เปิดไฟจาก BATTERY 3.6 V ถ้ามีการต่อใช้งาน BATTERY

64 , 256 (J1)

เป็น CONNECTOR 3 PIN ใช้ JUM เลือกว่าหน่วยความจำ RAM ของเราจะใช้หน่วยความจำไอซีบอร์ดอะไร 6264 (8 KBYTE) หรือ 62256 (32 KBYTE)

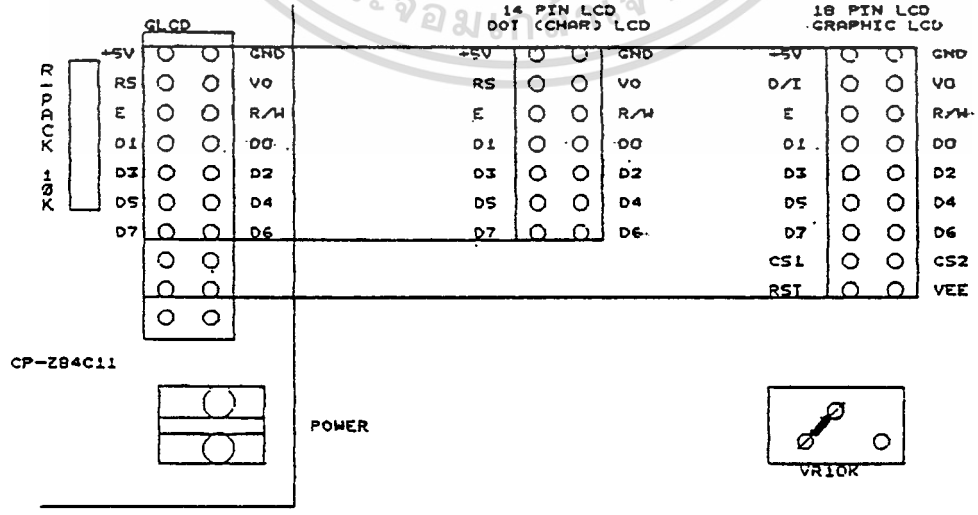
LCD J7

เป็น CONNECTOR 3 PIN ใช้เลือกว่าเราจะต่อ LCD MODULE ประเภทใดคือ LCD MODULE แบบตัวอักษรหรือ LCD MODULE แบบ GRAPHIC

รูปที่ 12 แสดงการเลือกใช้ LCD

GLCD

เป็น CONNECTOR ขนาด 20 PIN โดยสามารถต่อ LCD ได้ 2 แบบคือ DOT TYPE หรือ GRAPHIC TYPE โดยเราเลือกต่อได้ถ้าเป็น DOT TYPE คือ ตัวอักษรก็จะใช้ 14 PIN ต่อได้ดังรูป หรือถ้าจะเลือกต่อแบบ GRAPHIC TYPE ก็จะต่อใช้งาน 18 PIN



รูปที่ 13 แสดงการเลือกใช้ GLCD

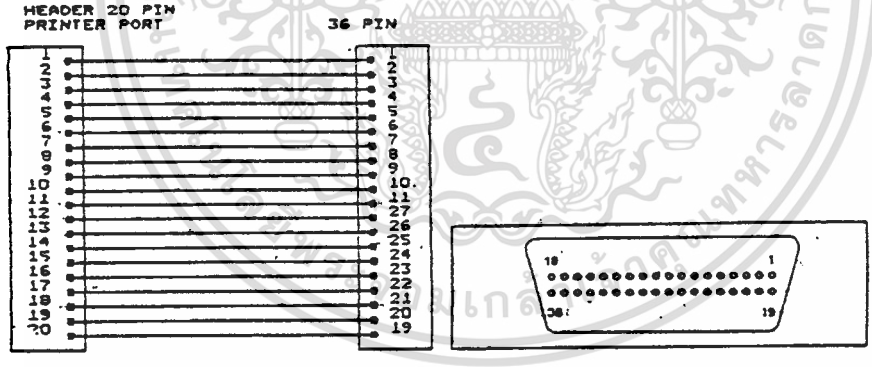
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เราสามารถปรับความคมชัดของ LCD ได้ด้วยการปรับ VR 10 K ในบอร์ดหรือถ้าต้องการให้ชัดสุดก็ต่อ JUM VR ตามรูปเลยก็ได้

CHANNEL PORT	ADDRESS PORT
WRITE DATA INSTRUCTION	80 H
WRITE DATA TO GO OR DD RAM	82 H
READ BUSY FLAG AND ADDRESS	84 H

PRINTER PORT

เป็น CONNECTOR 20 PIN ใช้ต่อเข้ากับเครื่องพิมพ์ แบบ CENTRONICS แบบธรรมดาทั่ว ๆ ไปได้ โดย PRINTER PORT นี้จะต่อจาก PORT PE 3-4 และ PORT PD 0-8

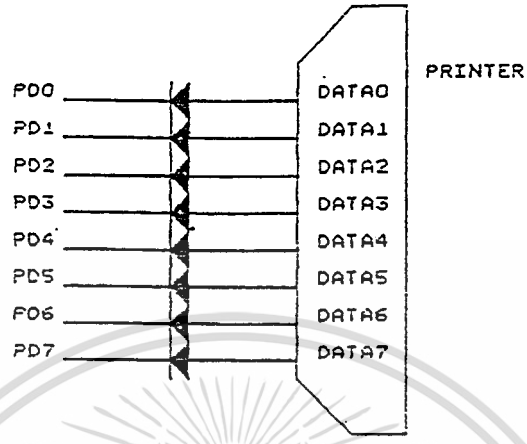


รูปที่ 14 แสดงลักษณะของ PRINTER PORT

PRINTER ADDRESS PORT	
CHANNEL	**ADDRESS PORT**
DATA (PD 0-8)	30 H
STB (PE 4)	40 H (BIT 4)
BUSY (PE 3)	40 H (BIT 3)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกรณีที่ต่อไช้ KEYBOARD ด้วยให้เราต่อเพิ่ม DIODE เข้ากับสัญญาณ PD1-PD7 ที่จะเข้าไปยังเครื่องพิมพ์ด้วยดังรูปเพิ่มไม่ให้สัญญาณจากเครื่องพิมพ์รบกวน KEYBOARD ที่เราต่อใช้



รูปที่ 15 แสดงการแก้สัญญาณรบกวนจากเครื่องพิมพ์ ไม่ให้รบกวน KEYBOARD

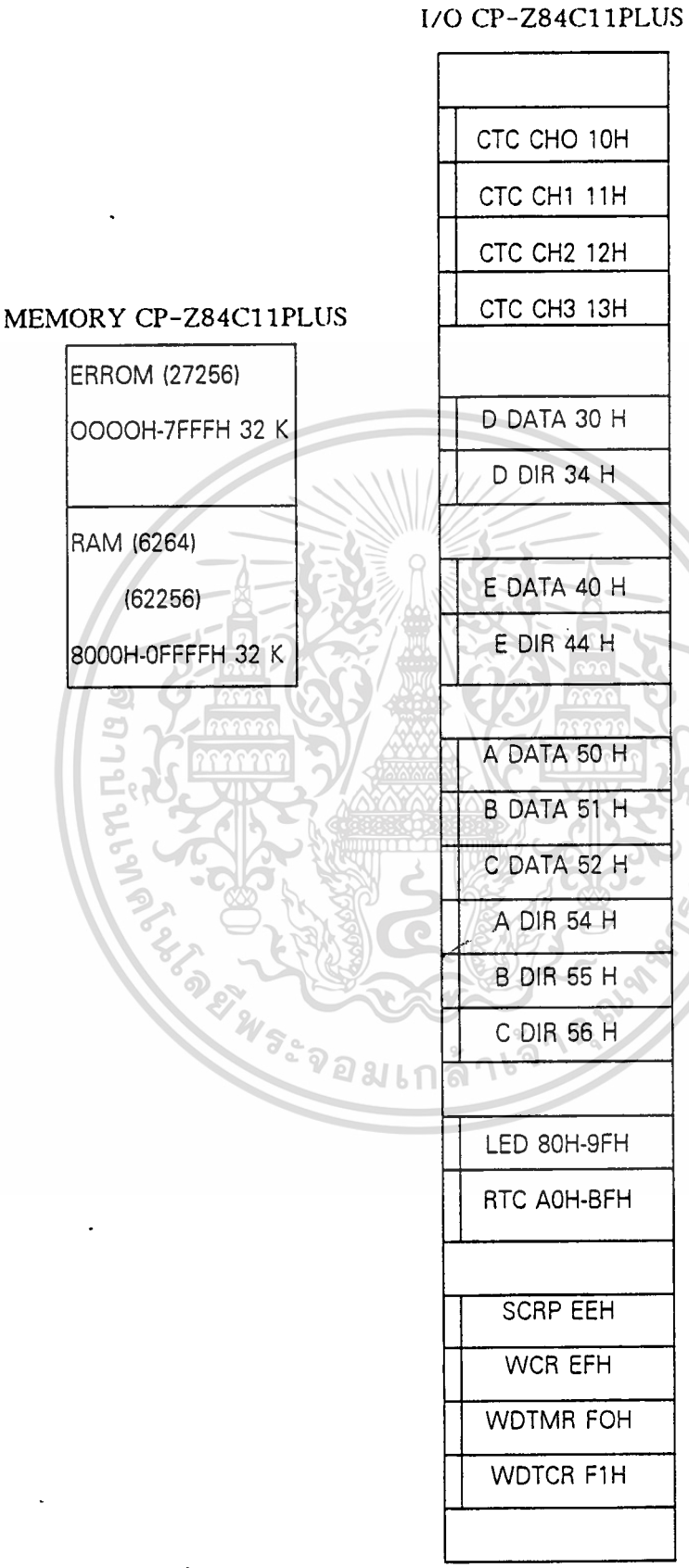
SPEAKER (SP)

เราสามารถต่อลำโพงเข้ากับชั้น SP นี้ได้ด้วยโดยสัญญาณจะมาจาก PE1 โดย SET JUMPER J10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MEMORY MAP ADDRESS

CP-Z84C11 จัด MEMORY ออกเป็น 2 ส่วน คือ



รูปที่ 16 แสดง MEMORY MAP ADDRESS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 ทฤษฎีส่วนการใช้คำสั่งควบคุมผลการแสดงบน แอลซีดี (LCD)

รายละเอียดของคำสั่ง HD44780

2.2.1 CLEAR DISPLAY

Clear display

	RS	R/W	DB7							DB0
Code	0	0	0	0	0	*0	0	0	0	1

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

2.2.2 RETURN HOME

RETURN HOME

	RS	R/W	DB_7							DB_0
Code	0	0	0	0	0	0	0	0	1	*

* No effect

คำสั่งนี้จะทำการ SET DD RAM ADDRESSER เป็นศูนย์ ตัว CURSOR จะกลับไปอยู่ตำแหน่งบนสุดซ้ายมือของจอภาพข้อมูลในจอภาพไม่เปลี่ยน

2.2.3 ENTRY MODE SET

Entry Mode set

	RS	R/W	DB_7							DB_0
Code	0	0	0	0	0	0	0	1	I/D	S

BIT I/D : โดยจะเป็นตัวกำหนดให้ว่าเมื่อเขียนหรืออ่านข้อมูลแล้วจะทำให้ DD RAM ADDREESS เพิ่มขึ้นหนึ่งหรือลดลงหนึ่งโดย 1 = เพิ่ม 0 = ลดลงหนึ่ง

BIT S : เป็นตัวกำหนดแสดงผลโดยถ้า S = 1 จะเป็นการใส่ข้อมูลแล้วตัว CURSOR อยู่กับที่ข้อมูลจะถูกดันไปทางซ้าย ถ้า S = 0 ข้อมูลจะอยู่กับที่ตัว CURSOR จะถูกดันไปทางขวามือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.4 DISPLAY ON/OFF CONTROL

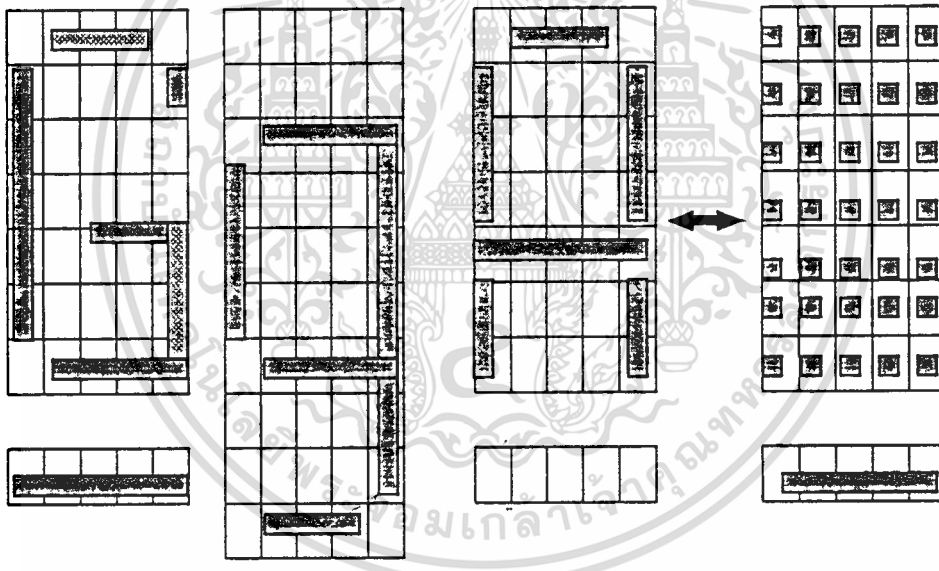
Display ON/OFF control

	RS	R/W	DB_7						DB_0
Code	0	0	0	0	0	0	1	D	C	B

BIT D : เป็น BIT ให้เปิดปิดหน้าจอภาพโดยถ้า D = 1 จะ ON และ D = 0 จะ OFF

BIT C : จะให้แสดง CURSOR ให้ BIT C = 1 และถ้าไม่ต้องการแสดง CURSOR BIT C = 0 โดยตัว CURSOR จะอยู่ที่ LINE ที่ 8 ในแบบ 5 × 7 DOT และจะอยู่ LINE ที่ 11 ในแบบ 5 × 10 DOT

BIT B : เป็น BIT SET การกระพริบของ CURSOR โดย B = 1 มีการกระพริบ B = 0 ไม่มีการกระพริบ โดยมีระยะเวลาการกระพริบประมาณ 379.2 ms



Cursor

Alternating display

5 × 7 dot character font

5 × 10 dot character font

(a) Cursor Display Example (b) Bink Display Example

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.5 CURSOR OR DISPLAY SHIFT

Cursor or display shift

	RS	R/W	DB_7							DB_0
Code	0	0	0	0	0	1	S/C	R/L	*	*

* No effect

เป็นคำสั่งกำหนดให้ตำแหน่ง CURSOR หรือข้อมูลไปเกิดทางซ้ายหรือทางขวาโดยไม่ต้องใช้คำสั่งเขียนหรืออ่าน โดย

S/C	R/L	
0	0	ทำการย้าย CURSOR ไปจากตำแหน่งเดิมไปซ้ายมือ 1 ตำแหน่ง
0	1	ทำการย้าย CURSOR ไปจากตำแหน่งเดิมไปขวามือ 1 ตำแหน่ง
1	0	เป็นการเลื่อนตัวอักษรที่เกิดไปทางซ้าย
1	1	เป็นการเลื่อนตัวอักษรที่เกิดไปทางขวามือ

2.2.6 FUNCTION SET

Function set

	RS	R/W	DB_7							DB_0
Code	0	0	0	0	1	DL	N	F	*	*

* No effect

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

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

F : เป็นการ SET ขนาด DOT การแสดงผล 5 × 7 หรือ 5 × 10 โดย F = 0 เป็นแบบ 5 × 7 และ F = 1 เป็นแบบ 5 × 10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

N	F	No.of display lines	Character font	Duty factor	Remarks
0	0	1	5 × 7 dots	1/8	
0	1	1	5 × 10 dots	1/11	
1	*	2	5 × 7 dots	1/16	cannot display 2 lines witch 5 × 10 dot character font

* No effect

2.2.7 SET CG RAM ADDRESS

Set CG RAM address

	RS	R/W	DB_7							DB_0
Code	0	0	0	1	A	A	A	A	A	A

Higher Order Bits ← Lower Order Bits →

ใน HD44780 นั้นจะมีหน่วยความจำอยู่ 2 ชุด คือ DISPLAY DATA RAM (DD RAM) จำนวน 80×8 และ CHARACTER GENERATOR ROM CG RAM จำนวน 512 BIT และ 7200 BIT คำสั่งนี้จะเป็นการ SET ADDRESS ใน CG RAM โดยต้องทำการ SET ADDRESS ก่อนเขียนหรืออ่านข้อมูลจาก CG RAM ด้วย

2.2.8 SET DD RAM ADDRESS

Set DD RAM address

	RS	R/W	DB_7							DB_0
Code	0	0	1	A	A	A	A	A	A	A

Higher Order Bits ← Lower Order Bits →

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

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

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

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

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

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

display position

DD RAM address

HDM-16416H

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1-line	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
2-line	40	41	42	43	44	45	46	47	48	49	4A	4B	4C	4D	4E	4F
3-line	14	15	16	17	18	19	1A	1B	1C	1D	1E	1F	20	21	22	23
4-line	54	55	56	57	58	59	5A	5B	5C	5D	5E	5F	60	61	62	63

display position

DD RAM address

HDM-20216H

	17	18	19	20
1-line	10	11	12	13
2-line	50	51	52	53
3-line	24	25	26	27
4-line	64	65	66	67

display position

DD RAM address

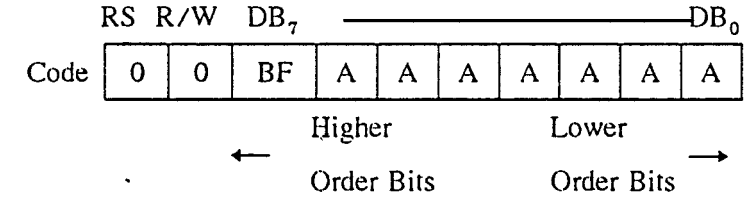
HDM-20216H (CONTINUE)

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.9 READ BUSY FLAG AND ADDRESS

Read busy flag & address



เป็นคำสั่งอ่านค่า BUSY FLAG ซึ่งจะเป็นตัวบอกว่าตัว HD44780 นี้อยู่ในขบวนการทำงานภายในอยู่หรืออยู่ในสภาพพร้อมจะรับข้อมูล โดย

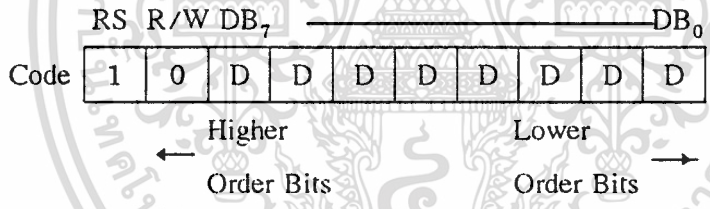
BF = 1 อยู่ในขบวนการทำงานภายในไม่พร้อมจะรับข้อมูลหรือคำสั่ง

BF = 0 พร้อมจะรับข้อมูลหรือคำสั่งได้

และนอกจากนี้ยังเป็นคำสั่งอ่านค่าข้อมูล ADDRESS ของ CG RAM หรือ DD RAM ด้วย

2.2.10 WRITE DATA TO CG หรือ DD RAM

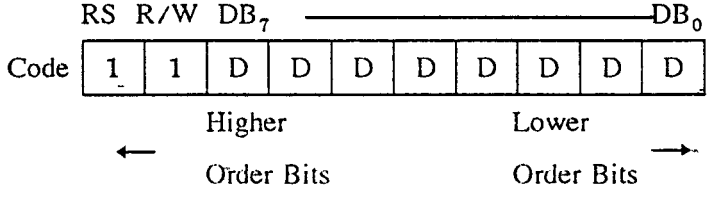
Write data to CG or DD RAM



เป็นคำสั่งเขียนข้อมูลเข้าไปใน CG หรือ DD RAM โดยเมื่อเขียนข้อมูลและ ADDRESS จะเพิ่มหรือลดโดยอัตโนมัติตามคำสั่งที่ SET ใน ENTRY MODE ข้อกำหนดที่จะรู้ว่าเป็นการเขียนข้อมูลของ CG RAM หรือ DD RAM ทำได้โดยการ SET ADDRESS ของ CG RAM หรือ DD RAM ขึ้นมาก่อนจะเขียนข้อมูล

2.2.11 READ DATA FROM CG OR DD RAM

Read data from CG or DD RAM

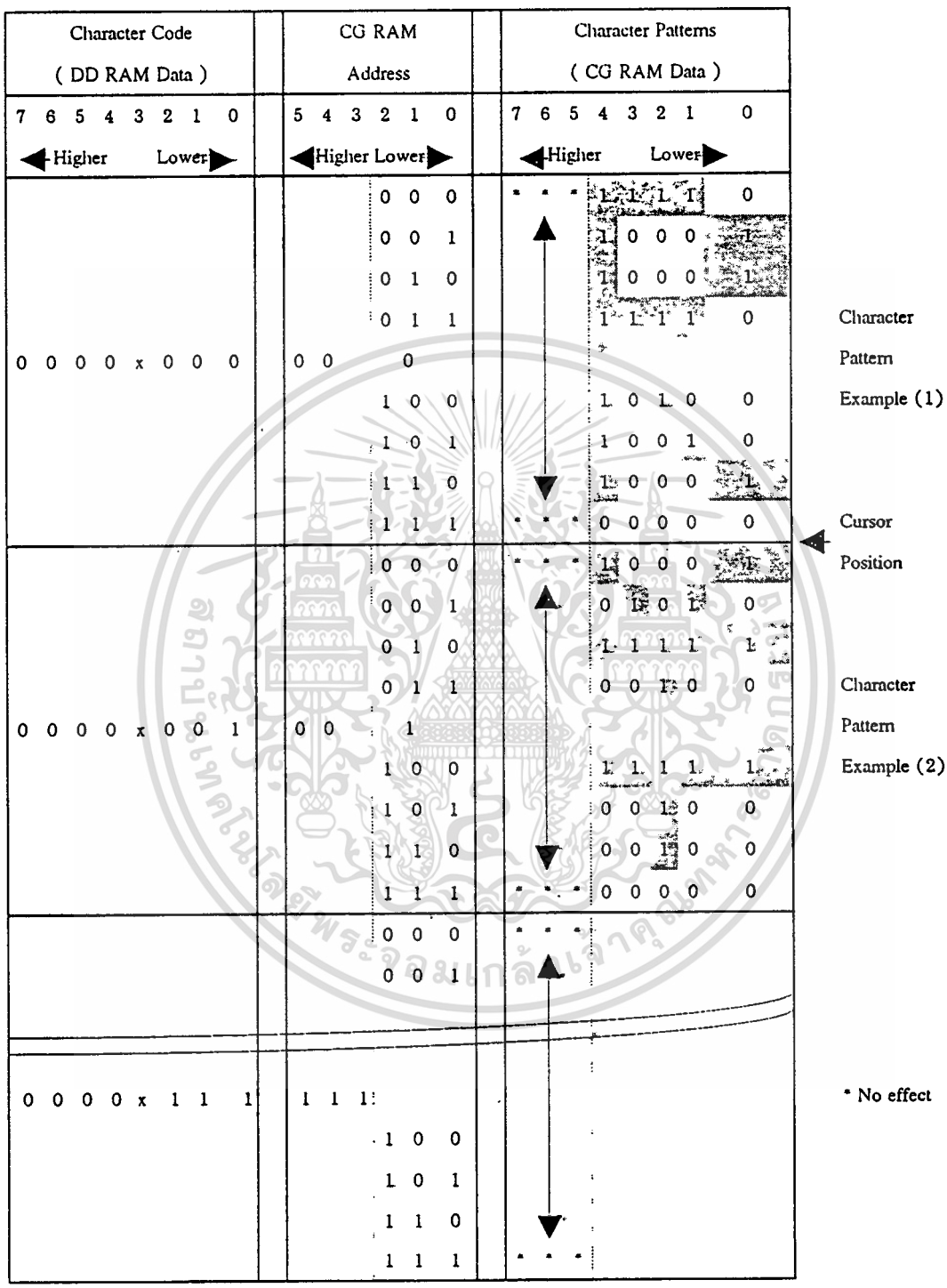


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
036935

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

จากตารางการทำงานจะเห็นว่าการใช้งาน LCD MODULE นั้นง่ายเพียงแต่เราส่งคำสั่งเริ่มแรกและ SET ความต้องการในขนาดตัวอักษร, CURSOR หลังจากนั้นเราก็สามารถเขียนตัวอักษรเข้าไปใน DD RAM ตามตารางตัวอักษรที่ให้มานั้นก็จะเกิดอักษรในจอภาพ LCD เรายังสามารถกำหนดตำแหน่งตัวอักษรที่จะให้เกิดบนจอได้โดยการ SET DD RAM ADDRESS ตามตารางที่ให้มาในหัวข้อ SET DD RAM ADDRESS ขอให้ทดสอบทำความเข้าใจกับตัวโปรแกรมที่ใช้กับ ET-BOARD V3.0 นี้ที่ให้มาจะเห็นว่าจะมีส่วนเริ่มต้นก็คือ ส่วนการ INITIAL LCD เพื่อกำหนดหน้าที่การทำงานต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



(Note) 5 x 7 dot character pattern

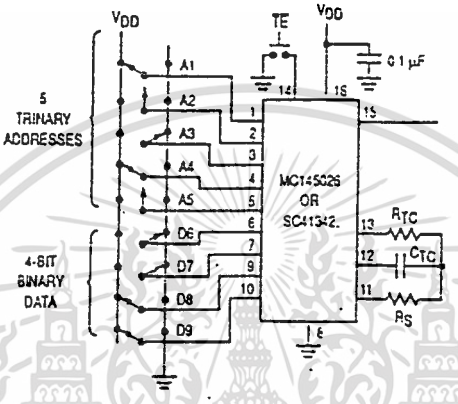
ตารางที่ 1 แสดงการใช้งาน LCD MODULE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

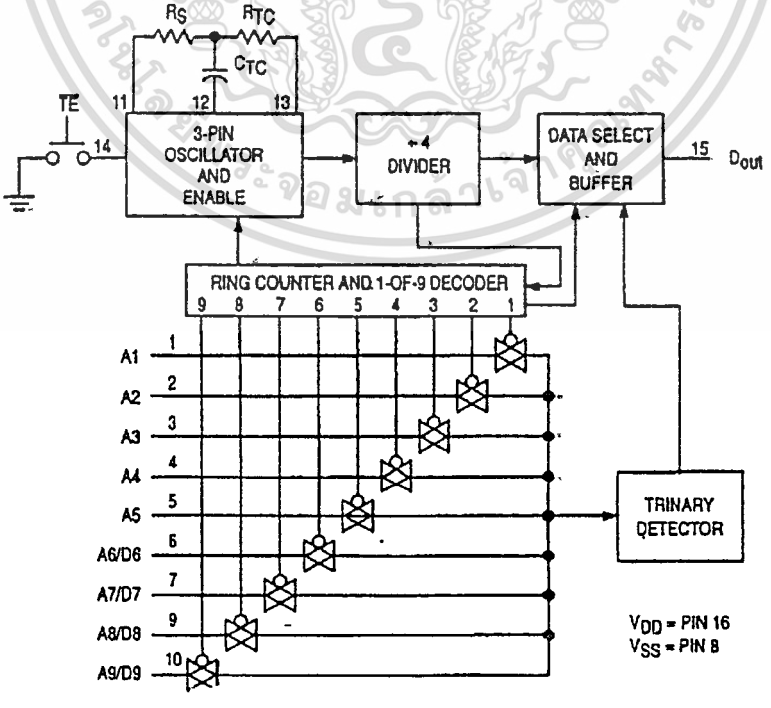
2.3 ทฤษฎีส่วนที่ทำหน้าที่เข้ารหัส (ENCODER) และถอดรหัส (DECODER)

2.3.1 ส่วนวงจร เข้ารหัส (ENCODER)

MC 145026 ทำหน้าที่ Encoder คือทำหน้าที่เข้ารับสัญญาณแบบ 9 บิทของข่าวสาร และจะส่งข่าวสารนี้เป็นแบบอนุกรม โดยข่าวสารจะสามารถส่งออกไปได้ทันทีที่สวิทช์ Transmit Enable (TE) ถูกกด ซึ่งจะทำงานที่สภาวะ “Low” เท่านั้น



รูปที่ 17 แสดงลักษณะของ IC เบอร์ MC 145026 (Encoder)



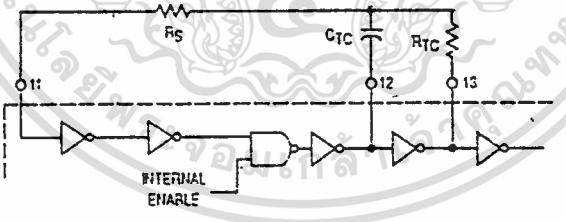
รูปที่ 18 แสดง MC 145026 Encoder Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คุณสมบัติของ IC MC 145026

- Address สามารถเป็นไปได้ทั้ง Binary หรือ Trimary
- Code ของ Address ที่สูงที่สุดคือ Trinary
- ใช้ในการ Interfaces กับ RF, Ultrasonic
- จะส่ง 2 Data เพื่อสำหรับ Error Checking
- จะใช้ไฟตั้งแต่ 4.5-18 Volts
- ใน chip นั้นจะมี R/C Oscillator
- Input และ Output เป็นแบบอนุกรมมาตราฐาน

ส่วนของ Encoder นี้สามารถทำการเข้ารหัสได้มาก โดยขึ้นอยู่กับขา Input ทั้ง 9 บิท ดังนั้นเราจะกำหนดสถานะขา Input ของ A1~A9 ขาเหล่านี้สามารถเป็นไปได้ทั้ง 3 สถานะ คือ Trinary จึงอาจจะเป็น (0, 1, Open) สถานะใดสถานะหนึ่งก็ได้โดย Codes ที่สามารถทำการเข้ารหัสที่เป็นไปได้ซึ่งมีความแตกต่างกันสูงสุดถึง 3^9 = 19683 ซึ่งลำดับในการสั่งนั้นจะเริ่มต้นที่ "Low Level" ของขา Input TE และ VDD ส่วนมากจะใช้เป็น "Positive Supply" และ VSS ส่วนมากจะใช้เป็น "Negative Supply" (GND) แล้วในส่วนของ Encoder นี้ยังมี RS, RTC, CTC ขาเหล่านี้จะเป็นของ Oscillator ของการเข้ารหัส ถ้าแหล่งจ่ายสัญญาณภายนอกใช้แทน Oscillator ภายในมันจะต่อไปที่ขา RS, RTC, CTC จะเป็น "Leftopen" ซึ่งจะเห็นได้ว่า Oscillator นี้จะทำงานที่ความถี่กำหนดโดย RC Network ภายนอก



รูปที่ 19 แสดง Encoder Oscillator Information

RC Network ภายนอกสามารถคำนวณได้จากสูตร

$$F = \frac{1}{(2.3 \times RTC \times CTC)} \quad (Hz)$$

For 1 Khz <= F <= 400 KHz

เมื่อ CTC = CTC + Clayout + 12 PF

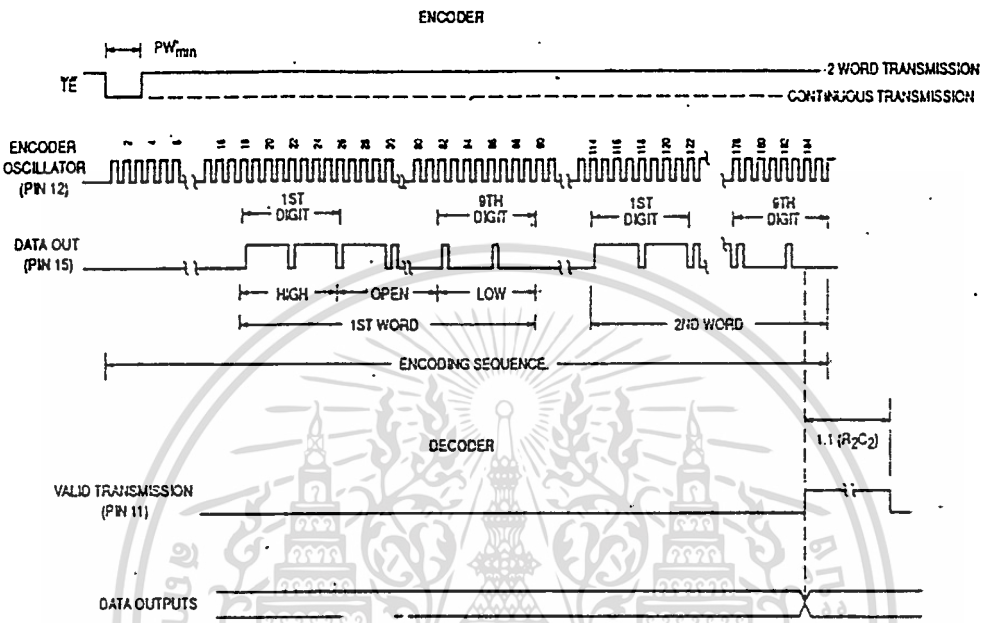
RS ≈ 2RTC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RS ≥ 20 K

RS ≥ 10 K

400 Pf < CTC < 15 μF



รูปที่ 20 แสดง Timing Diagram

ดังนั้น Data Out ที่ขา 15 ของ MC 145026 ที่ส่งออกไปจะส่งไปในลักษณะอนุกรมต่อกันไปขนาด 9 บิท ซึ่งในแต่ละบิทของ Data ที่ส่งออกไปจะมีสภาวะเป็น 0 , 1 , OPEN สภาวะใดก็ได้ ซึ่งก็จะมีสัญญาณเป็นลักษณะ Pulse ที่เข้ารหัสเป็นแบบอนุกรมแล้ว ดังรูปที่แสดงลักษณะการเข้ารหัสของ Data Bit

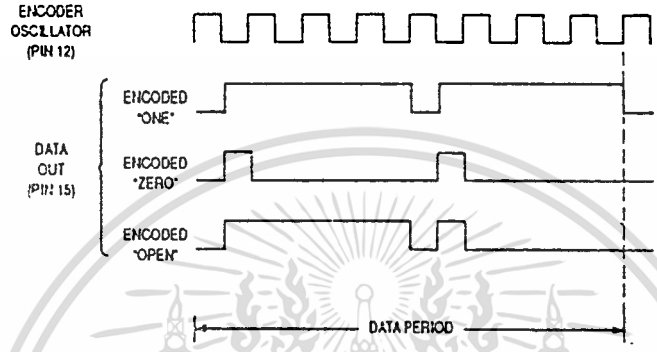
ข้อมูลต่าง ๆ จะเริ่มมีการส่งข้อมูลแบบอนุกรมต่อเนื่องกันไป ก็ต่อเมื่อทันทีที่ Switch TE ถูกกด (ที่ขา TE ได้รับ Active Low) ซึ่งข้อมูลดังกล่าวจะถูกส่งออกไปมีลักษณะเป็น Word โดยจะถูกส่งออกไปเป็นจำนวน 2 Data Words ด้วยกัน ซึ่งหากทางภาครับในส่วนของ Decoder รับได้ Data Words ทั้งสองนี้ได้เหมือนกันแสดงว่าการส่งและการรับข้อมูลเป็นไปอย่างถูกต้องสมบูรณ์ที่สุด จะทำให้ภาครับในส่วนของ Decoder ผลิตสัญญาณ VT (Valid Transmission) ออกมาเป็น Logic “1” ทุกครั้ง

ในการส่งแต่ละครั้ง Data Bit จะถูกเข้ารหัสแบบ 2 Data Pulse โดยที่ Logic “0” จะมีลักษณะเป็น Pulse สั้น ๆ 2 Pulse ต่อเนื่องกันไป และ Logic “1” จะมีลักษณะเป็นแบบ Pulse

เอกสารนี้ถือเป็นลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปใช้ในการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ยาว ๆ 2 Pulse ต่อเนื่องกัน ส่วนสำหรับสภาวะ “OPEN” จะมีการเข้ารหัสเป็นแบบ Pulse ยาว 1 ครั้ง แล้วตามด้วย Pulse สั้น ๆ อีก 1 Pulse ดังแสดงการเข้ารหัสของสภาวะต่าง ๆ ดังกล่าว (0 , 1 , OPEN) ดังรูป

ENCODER DATA WAVE FORMS (MC 145026)



รูปที่ 21 แสดงลักษณะการเข้ารหัสของ Data Bit

ดังนั้น Data Out ที่ขา 15 ของ MC 145026 ขนาด 9 บิท ซึ่งในแต่ละบิทของ Data ที่ส่งออกไปจะมีสภาวะเป็น 0 , 1 , OPEN สภาวะใดก็ได้ ซึ่งจะมีสัญญาณเป็นลักษณะ Pulse ส่งไปให้ภาค Decoder ทำการถอดรหัสต่อไป

2.3.2 ส่วนวงจร ถอดรหัส (DECODER)

MC 145027 ทำหน้าที่ Decoder โดยทำหน้าที่รับสัญญาณที่ส่งมาจากส่วนของ Encoder ที่ส่งเข้ามาแล้วทำการถอดรหัสดูว่าตรงกับค่า Address ที่ Set ไว้หรือไม่ ซึ่งก็คือ หลักในการรับส่งข้อมูลระหว่างส่วนของ Encoder และ Decoder นั้นข้อมูลที่ส่งมาจะรับได้ก็ต่อเมื่อค่าของ Address ของ Encoder และ Decoder ต้องตรงกันเท่านั้นจะทำให้ได้ว่า Output ออกมา แล้วนำค่า Output ที่ได้นำไป Control Load อีกที ลักษณะของ MC 145027 ตามรูป

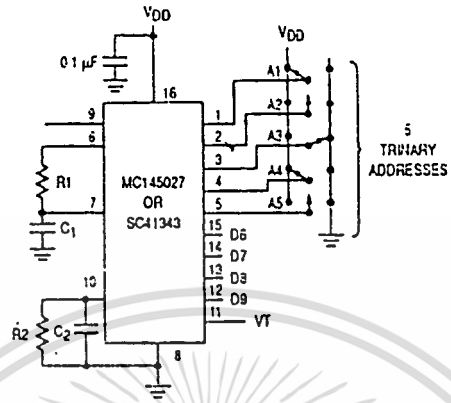
A1-A5

เป็นส่วนของ local address inputs ซึ่งในส่วนของ Decoder นี้ต้องทำการ Set ให้ตรงกับค่า Address ของ Encoder เพื่อที่จะได้ค่าของ Output ออกมา และทำให้ค่า VT เป็น Logic “1” ซึ่งส่วนของ Local Address Inputs นี้สามารถเป็นได้ทั้ง Trinary หรือ Binary

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D6-D9

เป็นส่วนของค่า Output ปัจจุบันนั้นโดยมีค่าเป็น Binary



รูปที่ 22 แสดง IC MC 145027 DECODER

R1, C1 (Pin 6,7)

ขาทั้งสองนี้จะต่อกับ Registor และ Capacitor เพื่อใช้เป็นตัวกำหนดความแคบ หรือความกว้างของ Pulse ที่เป็นรหัสค่า Time Constant ค่าของ R1xC1 จะถูกตั้งไว้ 1.72 ของ Encoder Clock Periode ดังนั้นค่าของ R1C1 สามารถกำหนดได้ดังนี้

R1C1 = 3.95 RTC CTC

R2, C2 (Pin 10)

ขานี้จะมี Registor และ Capacitor ต่อขนานกัน โดยต่อขานี้เข้ากับขา VSS เพื่อเป็นการแยกการส่งครั้งสุดท้ายกับครั้งใหม่ ซึ่งมีค่า Time Constant R2xC2 จะเป็น 33.5 ของคาบเวลาการส่ง (4 Data Bit Periods) โดยค่าของ R2C2 กำหนดโดย

R2C2 = 77 RTC CTC

VT (Pin 11) Valid Transmission

ขานี้จะให้ Output เป็น high ก็ต่อเมื่อหลังจากได้รับสัญญาณจากตัวส่งคือ Encoder ซึ่งมีลักษณะเป็นแบบ 2 Data Pulse ติดต่อกันมา โดยมีเงื่อนไขดังต่อไปนี้

- การรับ Address ของตัวรับต้องตรงกับ Address ของตัวส่ง
- การรับ Data Bits คือ Bit แรก และ Bit ที่สองต้องเหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

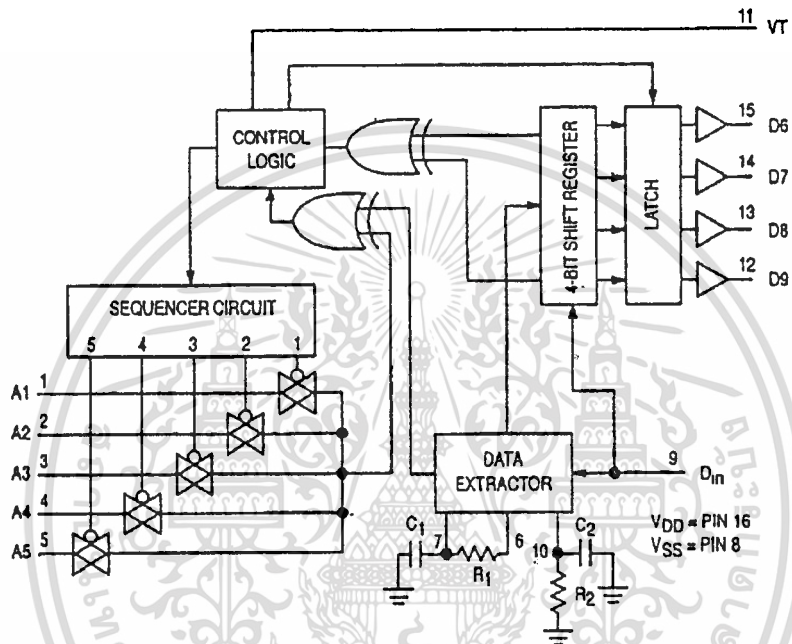
- ขา VT จะเป็น “high” จนกระทั่งการรับไม่ตรงตามเงื่อนไป หรือไม่มีสัญญาณที่รับเข้ามาเป็นเวลา 4 Data Bit

Vss (Pin 8)

ขาแหล่งจ่ายไฟบวก โดยใช้ร่วมกับขา ground

VDD (Pin 16)

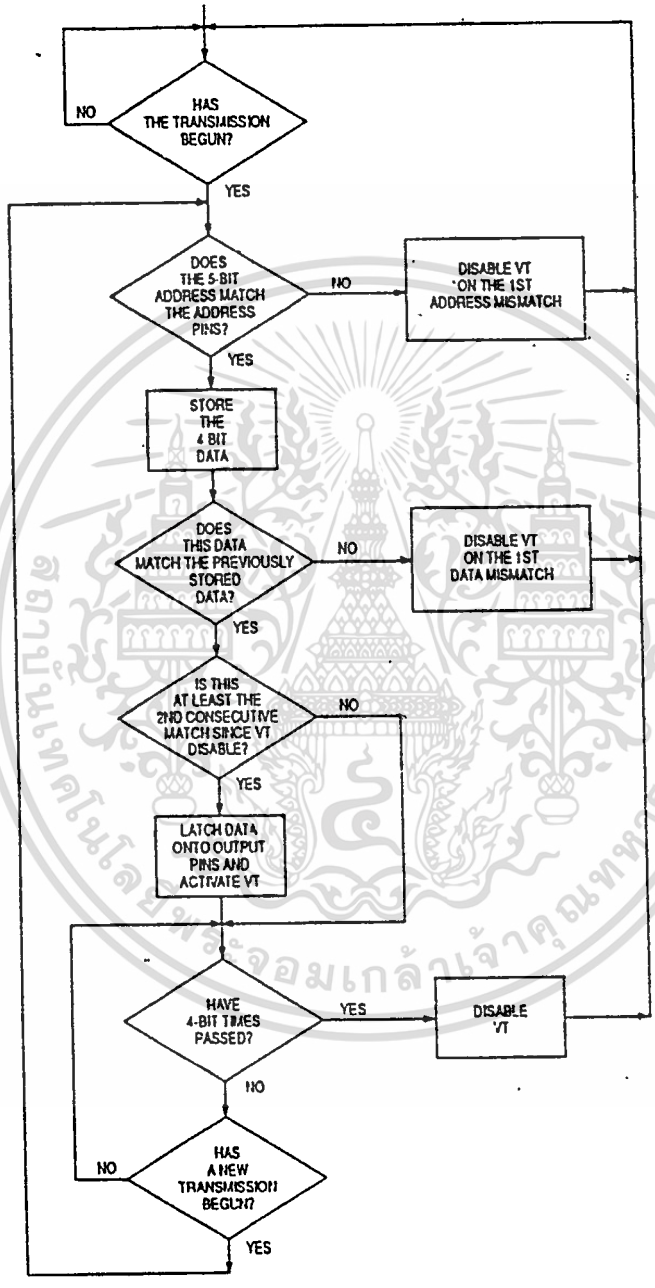
ขาแหล่งจ่ายไฟลบ (ส่วนใหญ่ให้เป็น Ground)



รูปที่ 23แสดง IC MC 145027 Decoder Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลักการทำงานของ IC MC 145027 เป็นไปตาม Flowchart ของรูป ดังนี้



รูปที่ 24 แสดง MC 145027 Flowchart

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 ทฤษฎีส่วนของ แกล (GAL)

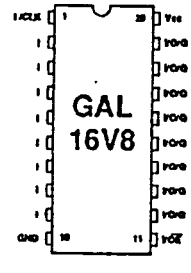
จากอุปรณ์ไอีตระกูล PAL (Programable Array Logic) ซึ่งเป็นอุปกรณ์ที่ทางโรงงานผู้ผลิตได้บรรจุวงจรเกตพื้นฐานแบบต่าง ๆ เช่น AND เกต, OR เกต, อินเวอร์เตอร์รวมไว้ในตัวเดียวกันผู้ใช้สามารถออกแบบเชื่อมโยงวงจรเกตเหล่านี้เข้าด้วยกันได้ เพื่อให้วงจรดิจิตอลลอจิกมีฟังก์ชันการทำงานตามความต้องการ อุประกรณ์ที่สามารถโปรกรมการทำงานได้นี้มีชื่อเรียกรวมกันว่า PLD (Programable Logic Device) ซึ่งก็มีอยู่หลายชนิดด้วยกับและ PAL ก็เป็นอุปรณ์ตัวหนึ่งในจำพวก PLD

ปัจจุบันได้มีการผลิตอุประกรณ์ PLD ชนิดไหม่โดยมีชื่อว่า GAL (Generic Array logic) เจ้า GAL (อ่านว่า “แกล”)

2.4.1 GAL (Generic Array logic)

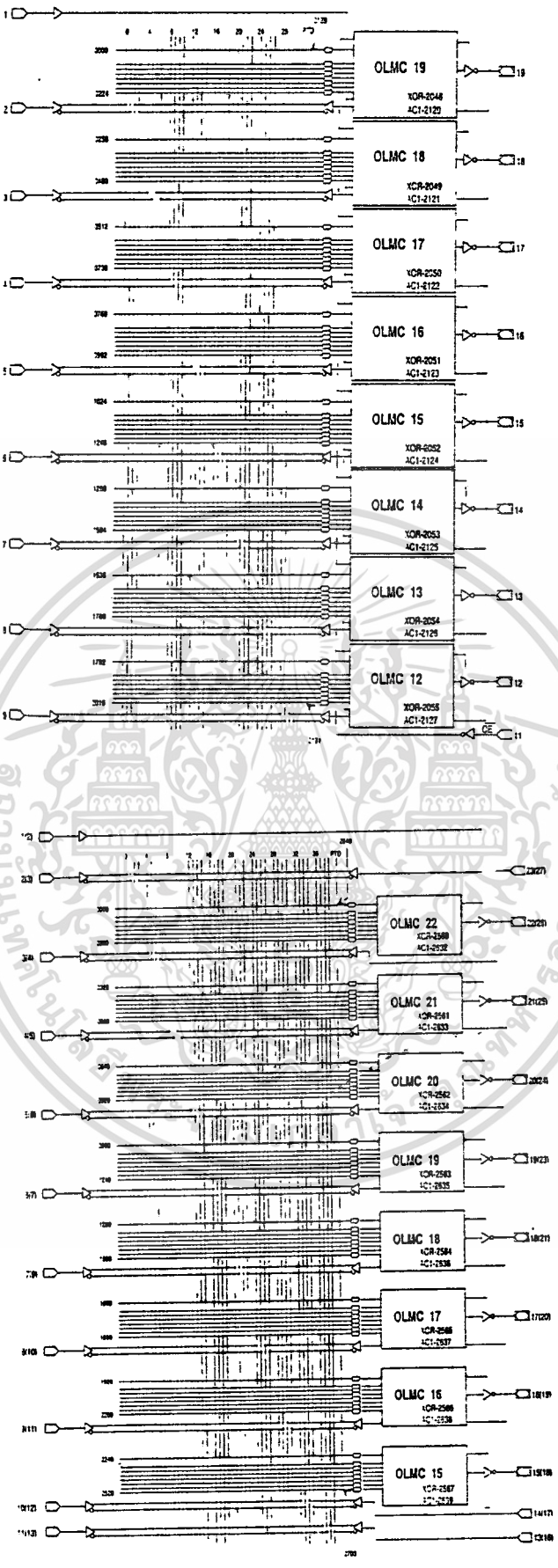
จากไอซี PAL ที่สามารถเลือกชนิดการทำงานของเอาต์พุตแบบต่าง ๆ กันได้ เช่น ทำงานที่ลอจิก “H” (active high) หรือ “L” (active low) หรือเป็นแบบรีจิสเตอร์ที่สามารถออกแบบวงจรซีเควนเชียล (sequential) ได้ ทำให้มีการพัฒนามาเป็น GAL โดยที่ GAL 1 ตัวสามารถใช้แทน PAL แบบต่าง ๆได้มากมาย

GAL ที่จะพูดถึงในส่วนนี้ก็คือ GAL16V8 และ GAL20V8 ของบริษัท Lattice Semiconductor



รูปที่ 25 การจัดขาและลักษณะตัวถังของ GAL16V8 และ GAL20V8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 26 ลอจิกไดอะแกรมของ GAL16V8 และ GAL20V8

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยทั้งสองตัวมีคุณสมบัติต่าง ๆ โดยสังเขปดังนี้

☐ ใช้เทคโนโลยี E^2CMOS (Electrically Erasable CMOS) ในการผลิต มีข้อดีคือ
- เวลาหน่วง (Propagation delay) สูงสุด 15 นาโนวินาที
- ใช้งานความถี่สูงสุด 50 Mhz
- เวลาที่ใช้สูงสุดจากคล็อกถึงการเปลี่ยนแปลงที่เอาต์พุต 12 นาโนวินาที
- คุณสมบัติทางเอาต์พุตเหมือนกับไอซีทีทีแอล (รับกระแสได้ 24mA)

☐ ใช้พลังงานเพียงครึ่งเดียวของอุปกรณ์ไบโพลาร์ทั่วไปคือ กินกระแสสูงสุด (I_{CC}) เพียง 45 มิลลิแอมป์เท่านั้น

☐ โครงสร้างเป็น E^2CELL ทำให้
- สามารถกำหนดลอจิกการทำงานใหม่ได้
- ผ่านการทดสอบและรับประกัน 100 เปอร์เซ็นต์เต็ม
- ใช้ความเร็วในการลบโปรแกรมด้วยไฟฟ้าน้อยกว่า 50 มิลลิวินาที
- เก็บรักษาข้อมูลที่โปรแกรมได้นานถึง 20 ปี

☐ มีเอาต์พุตลอจิกมาโครเซลล์หรือ OLMC (Output Logic Macrocell)8 เอาต์พุต ทำให้
- สะดวกต่อการออกแบบวงจรลอจิกที่ซับซ้อนมากได้
- สามารถโปรแกรมเอาต์พุตให้ทำงานที่ลอจิก “H” หรือ “L” ได้
- GAL16V8 สามารถนำมาใช้งานแทน PAL ขนาด 20 ขาได้หลายชนิดได้แบบเต็มฟังก์ชันการทำงาน
- GAL20V8 สามารถจำลองใช้แทน PAL ขนาด 24 ขา

☐ มีการปรีโหลดและเพาเวอร์ออนรีเซต (power-on reset) ของทุกรีจิสเตอร์

ความสามารถในการใช้ GAL16V8 และ GAL20V8 แทนอุปกรณ์ PAL ต่าง ๆ ได้สรุปไว้แล้วในตารางที่ 2 โดยจะเห็นได้ว่า GAL 1 ตัว สามารถแทน PAL ได้ถึง 21 เบอร์ และด้วยเทคโนโลยี E^2CMOS ในการผลิตจึงสามารถลบและเขียนโปรแกรมใหม่ได้ถึง 100 ครั้ง โดยจะใช้กระแสไฟฟ้าในการลบโปรแกรมนอกจากนี้ GAL ยังกินกระแสน้อยเพียงครึ่งหนึ่งของแบบไบโพลาร์ธรรมดา (50% หรือ 75 mA ในแบบ half power) และยังสามารถเลือกแบบกินกระแสเพียง 1/4 เท่าได้ด้วย (40 mA ในรุ่น quarter power)

จากบล็อกไดอะแกรมของ GAL ในรูปที่ 26 จะเห็นว่ามีวงจรแปลก ๆ ที่ชื่อ OLMC ซึ่งเจ้า OLMC หรือเอาต์พุตลอจิกมาโครเซลล์นี้จะเป็นหัวใจในการที่ทำให้ GAL นี้สามารถใช้แทน PAL ได้หลาย ๆ เบอร์

2.4.2เอาต์พุตลอจิกมาโครเซลล์ (OLMC)

บล็อกไดอะแกรมของเอาต์พุตลอจิกมาโครเซลล์ หรือ OLMC แสดงในรูปที่ 3 เอาต์พุตของแอนด์เกตในส่วนแอนด์อะเรย์จะป้อนเข้ามายัง OLMC ได้ 3 โหมดเพื่อให้ได้เอาต์พุตลอจิกแบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่าง ๆ ตามต้องการ ซึ่งการโปรแกรมเลือกโหมดการทำงานนี้จะถูกทำโดยซอฟต์แวร์หรือฮาร์ดแวร์ที่ใช้ในการออกแบบ GAL เองโดยอัตโนมัติ ซึ่งผู้ใช้งานไม่จำเป็นต้องไปสนใจวิธีการโปรแกรมมันเลย

ตารางที่ 2 แสดงไอซี PAL เบอร์ต่าง ๆ ที่สามารถใช้ GAL20V8 และ GAL16V8 แทนได้

GAL20V8 PAL Archltecture Emulation	GAL16V8 PAL Archltecture Emulation
20L8	16L8
20H8	16H8
20R8	16R8
20R6	16R6
20R4	16R4
20P8	16P8
20RP8	16RP8
20RP6	16RP6
20RP4	16RP4
14L8	10L8
16L6	12L6
18L4	14L4
20L2	16L2
14H8	10H8
16H6	12H6
18H4	14H4
20H2	16H2
14P8	10P8
16P6	12P6
18P4	14P4
20P2	16P2

โหมดการทำงานของ OLMC มีด้วยกัน 3 โหมด คือ ชิมเปิลโหมด (simple mode) คอมเพล็กซ์โหมด (complex mode) และรีจิสเตอร์โหมด (registered mode)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 3 แสดงแนวทางในการเลือกใช้ GAL อย่างคร่าวๆ ตามความเร็วในการทำงานและการสิ้นเปลืองพลังงาน

GAL 16V8(20-PIN) PLD SERIES	
GAL20V8-15L	15ns tpd,Half Power
GAL20V8-20L	20ns tpd,Half Power
GAL20V8-20LM	20ns tpd,Half Power (Military Temp.)
GAL20V8-20Q	20ns tpd,Quarter Power
GAL20V8-20QM	20ns tpd,Quarter Power (Military Temp.)
GAL20V8-25L	25ns tpd,Half Power
GAL20V8-25Q	20ns tpd,Quarter Power

ULTRA HIGH SPEED GAL 16V8A (20-PIN) PLD SERIES	
GAL16V8A-10L	10ns tpd,Half Power
GAL16V8A-12L	12ns tpd,Half Power

GAL 20V8 (24-PIN) PLD SERIES	
GAL16V8-15L	15ns tpd,Half Power
GAL16V8-20L	20ns tpd,Half Power
GAL16V8-20LM	20ns tpd,Half Power (Military Temp.)
GAL16V8-20Q	20ns tpd,Quarter Power
GAL16V8-20QM	20ns tpd,Quarter Power (Military Temp.)
GAL16V8-25L	25ns tpd,Half Power
GAL16V8-25Q	20ns tpd,Quarter Power

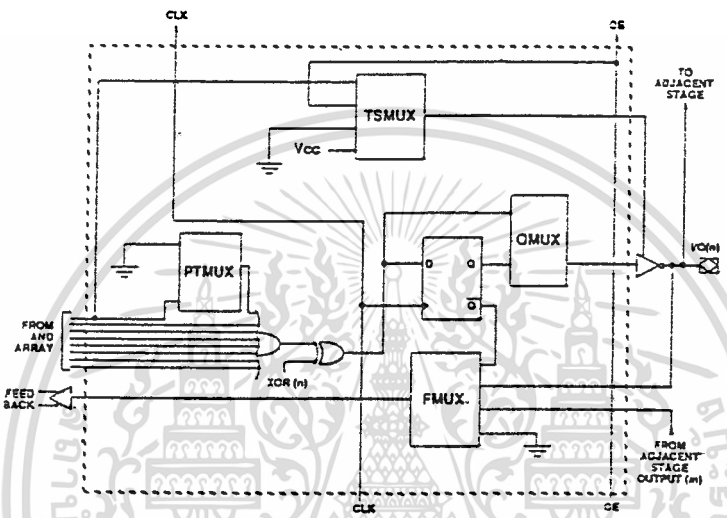
2.4.3 ซิมเปิลโหมด

ในโหมดนี้ OLMC จะมีลักษณะการต่อวงจรดังรูปที่ 28 ในแต่ละขาของ GAL จะทำหน้าที่เป็นอินพุตหรือเอาต์พุตได้เพียงอย่างเดียว

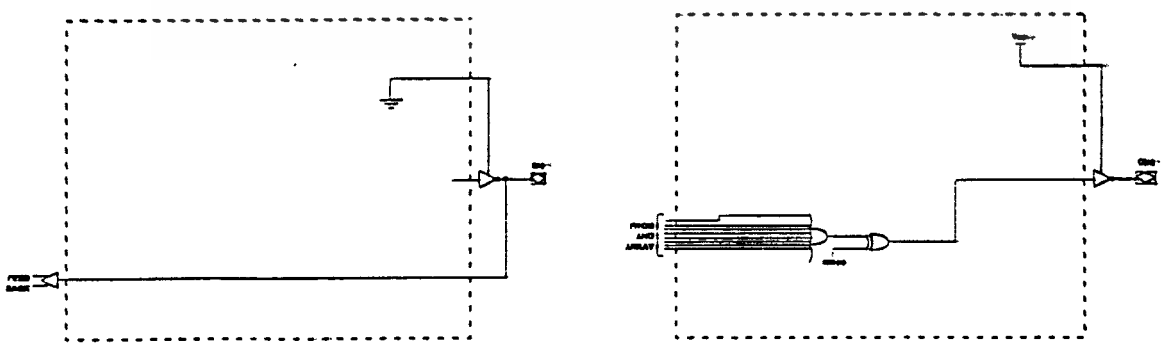
โครงสร้างเอาต์พุตจะเหมือนกับของ PAL10L8 18H4 และ 16P6 แต่สามารถโปรแกรมอินพุตและเอาต์พุตให้ทำงานที่ลอจิก “L” หรือ “H” ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอาต์พุตทั้งหมดจะเป็นของข้อมูลจากโปรดักเทอม (จากเอาต์พุตทั้ง 6 ของ AND เกต) มาเข้า OR เกต โดยแต่ละเอาต์พุตสามารถโปรแกรมให้ทำงานที่ลอจิก “L” หรือ “H” ได้ (programable polarity)



รูปที่ 27 บล็อกไดอะแกรมของเอาต์พุตลอจิกมาโครเซลล์ (OLMC)

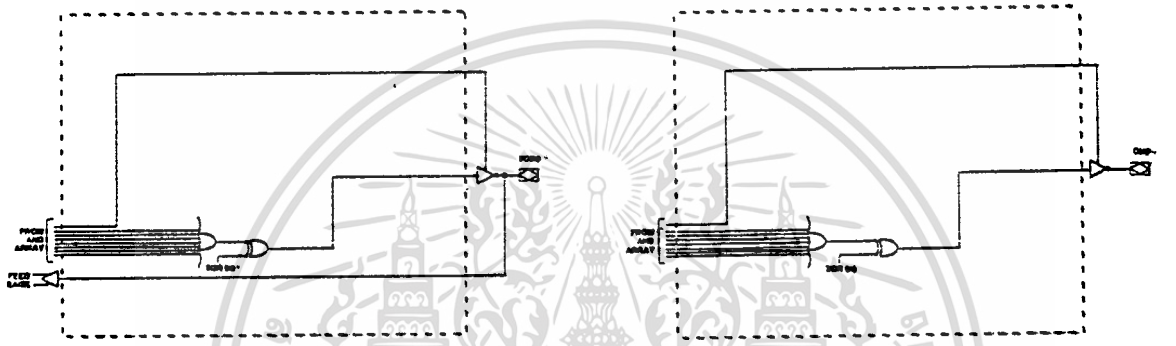


รูปที่ 28 ลักษณะการต่อวงจรภายในของ OLMC ในซิมเปิลโหมด

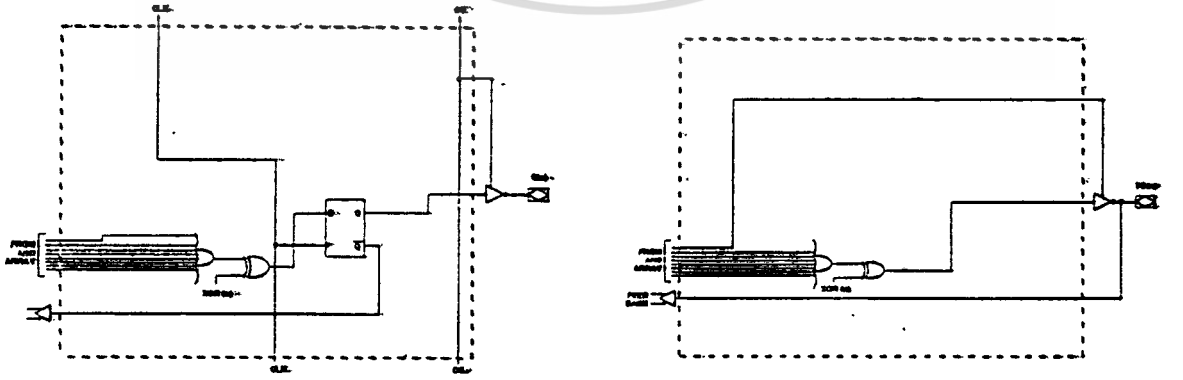
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในโหมดนี้ขา OE และ CLK (ขา 1 และ 11 ของ GAL16V8, ขา 1 และ 13 ของ 20V8) จะไม่ได้ใช้ ทำให้ใช้เป็นขาอินพุตป้อนสัญญาณเข้า AND อะเรย์ได้

แต่ OLMC ตัวที่อยู่ตรงกลางทั้งสองตัว (ขา 15 และ 16 ของ GAL 16V8, ขา 18 และ 19 ของ 20V8) จะไม่สามารถใช้เป็นขาอินพุตได้



รูปที่ 29 ลักษณะการต่อวงจรภายในของ OLMC ในคอมเพล็กซ์โหมด



รูปที่ 30 ลักษณะการต่อวงจรภายในของ OLMC ในรีจิสเตอร์โหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.4 คอมเพล็กซ์โหมด

ในโหมดนี้ OLMC จะมีลักษณะการต่อวงจรดังรูปที่ 29 โดยจะต่อเป็นเอาต์พุตอย่างเดียว หรือต่อเป็นอินพุต/เอาต์พุต (I/O function) ก็ได้เหมือนกับ PAL16L8, 20L8 และ 16P8

การจัดโครงสร้างในลักษณะนี้จะด้วยการโปรแกรมโพลาริตี้ลงในแต่ละ OLMC

ในโหมดนี้จะมีขาที่เป็นอินพุต/เอาต์พุตได้สูงสุด 6 ตัว (ขา) โดย OLMC อีก 2 ตัว (ขา 12 และ 19 ของ 16V8, ขา 15 และ 22 ของ 20V8) จะเป็นเอาต์พุตได้อย่างเดียว สัญญาณอินพุตหรือเอาต์พุตจะสามารถใช้เป็นส่วนหนึ่งของอินพุต/เอาต์พุตได้

เอาต์พุตของ OLMC ทั้งหมดจะได้จากโปรดักเทอม 7 เทอม ต่อ 1 เอาต์พุต โดยจะเสียโปรดักเทอม 1 เทอม จะใช้ของ GAL16V8, ขา 1 และ ขา 13 ของในการโปรแกรมสัญญาณ OE ซึ่งจะเหมือนกับชิมเปิลโหมดที่ขา 1 และ 11 20V8 สามารถใช้เป็นขาอินพุตป้อนข้อมูลไปยัง AND อะเรย์ได้

2.4.5 จิสเตอร์โหมด

ในโหมดนี้ OLMC จมีลักษณะการต่อดังรูปที่ 30 เอาต์พุตของ OLMC จะสามารถต่อเป็นรีจิตเตอร์เอาต์พุตหรือเป็นอินพุต/เอาต์พุตฟังก์ชันได้ จากลักษณะของโหมดนี้จะคล้ายกับ PAL16R8, 20R8 และ 16RP4

ในโหมดนี้ OLMC ทั้งหมดจะใช้สัญญาณ OE และ CLK ร่วมกัน OLMC แต่ละตัวจะสามารถทำงานเป็นรีจิตเตอร์หรืออินพุต/เอาต์พุตได้ โดยสามารถมีได้สูงสุด 8 ตัว สัญญาณอินพุตหรือเอาต์พุตสามารถนำมาใช้เป็นส่วนหนึ่งของอินพุต/เอาต์พุตได้

ถ้าต่อเป็นรีจิตเตอร์เอาต์พุต เอาต์พุตที่ได้จะเป็นของ 8 โปรดักเทอมต่อ 1 เอาต์พุต ถ้าต่อเป็นอินพุต/เอาต์พุตจะมี 7 โปรดักเทอมต่อ 1 เอาต์พุต

2.4.6 อิเล็กทรอนิกส์ซิกเนเจอร์

อิเล็กทรอนิกส์ซิกเนเจอร์หรือ ES (Electronic Signature) จะมีอยู่ใน GAL16V8 และ GAL20V8 ทุกตัว ES นี้จะเป็นหน่วยความจำที่สามารถโปรแกรมใหม่ได้ มีขนาด 64 บิท เพื่อให้ผู้ใช้สามารถป้อนข้อมูลของแต่ละคนได้ ES นี้จะไม่เกี่ยวข้องกับสถานะของซีเคียวริตี้เซลล์ (security cell)

ภายใน ES จะมีข้อมูลของ checksum ไว้ตรวจสอบความถูกต้องของข้อมูลเมื่อเปลี่ยนข้อมูลของ ES จะทำให้ checksum เปลี่ยนค่าได้

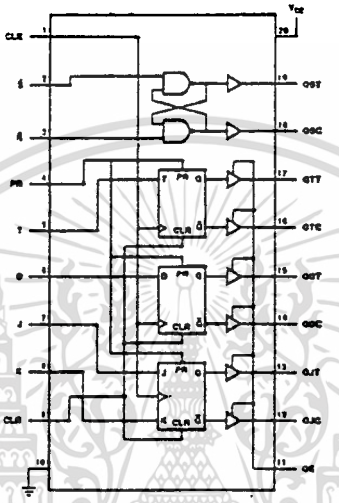
2.4.7 ซีเคียวริตี้เซลล์

ซีเคียวริตี้เซลล์ (security cell) มีไว้เพื่อป้องกันไม่ให้ผู้อื่นมาลอกเลียนแบบ (copy) ข้อมูลภายในตัว GAL ของเราได้ หลังจากที่เราโปรแกรมเซลล์นี้แล้ว จะทำให้ไม่สามารถอ่านข้อมูลภายใน AND อะเรย์ว่ามีการต่ออย่างไร เซลล์ชนิดนี้จะถูกลบได้อย่างเดียวด้วยการลบแบบบัลค์ (bulk erase cycle) ซึ่งข้อมูลทั้งหมดก็จะถูกลบไปพร้อมกันด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.8 อินพุตบัฟเฟอร์

อินพุตบัฟเฟอร์ของ GAL ทั้งสองเบอร์นี้ออกแบบเพื่อให้ใช้กับระดับแรงดันลอจิกของทีทีแอลได้ โดยจะมีอิมพีแดนซ์สูงทำให้กินกระแสจากวงจรที่นำมาต่อกับมันน้อย นอกจากนี้เอาต์พุตยังสามารถนำไปต่อขับอุปกรณ์ลอจิกอื่นๆ ได้มากกว่าทีทีแอลแบบไบโพลาร์ทั่วไปด้วย



รูปที่ 31 แสดงการจัดขาของ GAL16V8 เมื่อใช้ฟลิปฟลอบแบบต่าง ๆ

ที่อินพุทแต่ละตัวไม่มีการต่อพูลอัพภายใน ดังนั้นขาอินพุตและขาไอโอแบบสามสถานะที่ไม่ได้ใช้ควรจะนำไปต่อกับอินพุตอื่นที่ใช้งาน หรือต่อไฟเลี้ยงหรือกราวด์เพื่อลดระดับสัญญาณรบกวนและช่วยให้มีการกินกระแส (I_{CC})น้อยลง

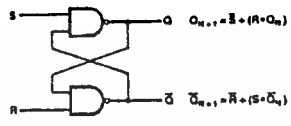
2.4.9 เอาต์พุตรีจิสเตอร์ปรีโหลด (output register preload)

ในการทดสอบการทำงานในซีเควนเชียลโหมด จะสามารถตั้งค่าเอาต์พุตของรีจิสเตอร์ได้ว่าต้องการให้เป็น "L" หรือ "H" ได้ เพื่อกำหนดสถานะปัจจุบัน (present state) ได้ แล้วจึงป้อนอินพุตเพื่อทดสอบสถานะถัดไป (next state) ว่าจะเป็นไปตามค่าที่ต้องการหรือไม่ และยังสามารถทดสอบการใช้งานตามความเป็นจริงคือ หลังจากตั้งค่าเอาต์พุตของรีจิสเตอร์แล้วจะหยุดสัญญาณที่ป้อนกลับแล้วป้อนสัญญาณที่ผิดปกติ (ช่วงเปิดเครื่อง (power up), ไฟเลี้ยงที่ใช้มีกลิตช์แฝงมา ฯลฯ) เข้าไปยังรีจิสเตอร์แล้วตรวจสอบดูว่าเอาต์พุตของวงจรที่ทดสอบจะยังสามารถให้เอาต์พุตในสถานะถัดไปที่ถูกต้องได้หรือไม่

ารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

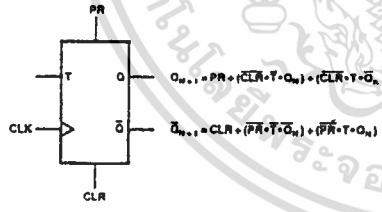
2.4.10 การลบข้อมูลแบบบัลค์โหมด (bulk erase mode)

ก่อนการโปรแกรมข้อมูล (pattern) จะต้องมีการลบข้อมูลเก่าทิ้งไปก่อน ซึ่งการลบข้อมูลนี้จะเป็นไปโดยอัตโนมัติ จากเครื่องที่ใช้โปรแกรม โดยจะเป็นส่วนหนึ่งของไซเกิลการเขียนข้อมูล ซึ่งใช้เวลาในการลบทั้งหมด 50 มิลลิวินาที



S	R	Q_N		Q_{N+1}	$\bar{Q}_{N+1}$	COMMENTS
0	0	0		1	1	INVALID
0	0	1		1	1	INVALID
0	1	0		1	0	SET
0	1	1		1	0	SET
1	0	0		0	1	RESET
1	0	1		0	1	RESET
1	1	0		0	1	LATCH
1	1	1		1	0	LATCH

รูปที่ 32 ตารางการทำงาน (truth table) และสมการของ RS-latch



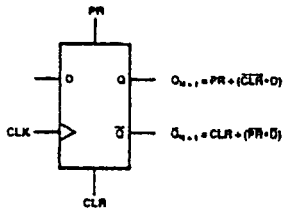
PR	CLR	T	Q_N		Q_{N+1}	$\bar{Q}_{N+1}$	COMMENT
1	1	X	X		1	1	INVALID
1	0	X	X		1	0	PRESET
0	1	X	X		0	1	CLEAR
0	0	0	0		0	1	HOLD
0	0	0	1		1	0	
0	0	1	0		1	0	TOGGLE
0	0	1	1		0	1	

รูปที่ 33 T-ฟลิปฟลอป

2.4.11 เพาเวอร์อัพรีเซต

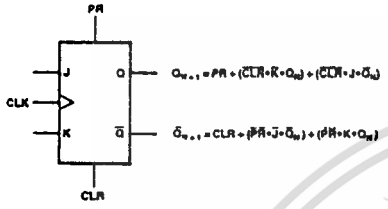
หลังจากที่ป้อนไฟเลี้ยงแล้วจะมีสัญญาณไปรีเซตรีจิสเตอร์ทุกตัว เมื่อเวลาผ่านไป (t_{RESET} สูงสุด 45 μS) เอาต์พุต Q ของรีจิสเตอร์ทุกตัวจะถูกตั้งให้เป็น "L" ส่งผลให้ขาเอาต์พุต (ถ้ามีสัญญาณ $\overline{OE}$ มากระตุ้นให้ทำงาน) จะมีลอจิกเป็น "H" โดยไม่ขึ้นกับการกำหนดโพลาริตี้ ความสามารถอันนี้จะทำให้การออกแบบวงจรซีเควนเชียลได้ง่ายขึ้น โดยการกำหนดสถานะช่วงเริ่มต้นของไซเกิลการทำงานให้ปรากฏที่ช่วงเพาเวอร์อัพ

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



PR	CLR	D	Q_N		Q_{N+1}	$\overline{Q}_{N+1}$	COMMENT
1	1	X	X		1	1	INVALID
1	0	X	X		1	0	PRESET
0	1	X	X		0	1	CLEAR
0	0	0	0		0	1	RESET
0	0	0	1		0	1	RESET
0	0	1	0		1	0	SET
0	0	1	1		1	0	SET

รูปที่ 34 D-ฟลิปฟลอบ



PR	CLR	J	K	Q_N		Q_{N+1}	$\overline{Q}_{N+1}$	COMMENT
1	1	X	X	X		1	1	INVALID
1	0	X	X	X		1	0	PRESET
0	1	X	X	X		0	1	CLEAR
0	0	0	0	0		0	1	HOLD
0	0	0	0	1		1	0	HOLD
0	0	0	1	0		0	1	RESET
0	0	0	1	1		0	1	RESET
0	0	1	0	0		1	0	SET
0	0	1	0	1		1	0	SET
0	0	1	1	0		1	0	TOGGLE
0	0	1	1	1		0	1	TOGGLE

รูปที่ 35 JK-ฟลิปฟลอบ

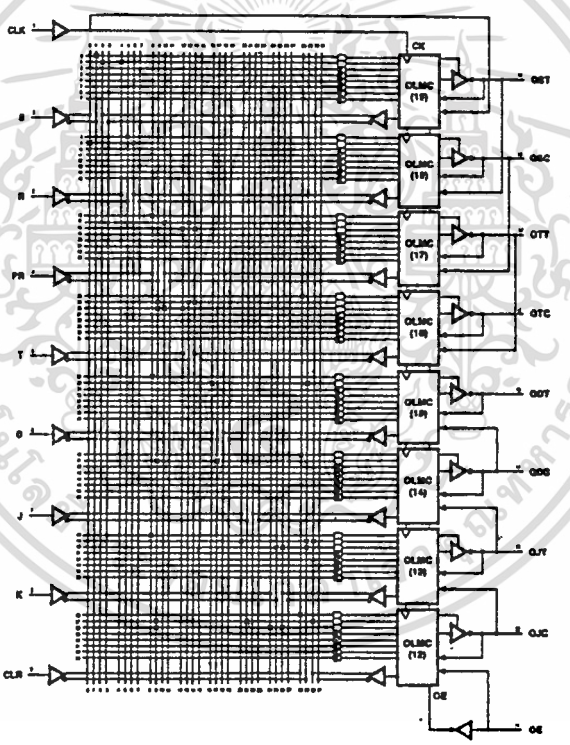
```
/*************************************************************/
/*                                                           */
/*                    CUPL INPUT FILE                        */
/*     Flip-flops and latches implemented in a GAL16V8       */
/*                                                           */
/*************************************************************/
      PARTNO          456STX;
      NAME            FLIPFLOP;
      REV             1;
      DATE            4/11/86;
      DESIGNER        Jerry;
      COMPANY         Lattice Semiconductor;
      ASSEMBLY        Clock Board;
      LOCATION        U23B;
/* RS latch */
      pin [2,3,19,18] = [S,R,QST,QSC];
/* T flip-flop */
      pin [5,17,16] = [T,QTT,QTC];
/* D flip-flop */
      pin [6,15,14] = [D,QDT,QDC];
/* JK flip-flop */
      pin [7,8,13,12] = [J,K,QJT,QJC];
/* control */
      pin [1,4,9,11] = [CLK,PR,CLR,OE];
/* logic equations */
      /* RS latch */
            QST = !S # (R & QST);
            QSC = !R # (S & QSC);
      /* T flip-flop */
            QTT.D = PR # (!CLR & !T & QTT) # (!CLR & T & QTC);
            QTC.D = CLR # (!PR & !T & QTC) # (!PR & T & QTT);
      /* D flip-flop */
            QDT.D = PR # (D & !CLR);
            QDC.D = CLR # (!D & !PR);
      /* JK flip-flop */
            QJT.D = PR # (J & QJC & !CLR) # (!K & QJT & !CLR);
            QJC.D = CLR # (!J & QJC & !PR) # (K & QJT & !PR);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 36 ข้อมูลที่ป้อนให้ CUPL

2.4.12 ตัวอย่างการออกแบบ GAL ไปประยุกต์ใช้ในงาน

ใช้เป็นฟลิปฟลอป (FF) พื้นฐานแบบต่าง ๆ เป็นการนำ GAL16V8 ไปใช้งานเป็นฟลิบฟลอบพื้นฐานทั้ง 4 แบบที่ได้แก่ RS-latch T-ฟลิบฟลอบ,D-ฟลิบฟลอบ,และ JK-ฟลิบฟลอบ ซึ่งมีการจัดขาดังในรูปที่ 31 และแสดงตารางทำงาน (truth table) และสมการของฟลิบฟลอบแต่ละตัวได้ดังรูปที่ 32-35 ตามลำดับ เป็นที่น่าสังเกตว่าฟลิบฟลอบในรูปที่ 32-35 จะใช้สัญญาณ clock, preset และ clear เหมือนกัน ส่วน RS-latch จะไม่ใช้ ซึ่งเป็นการแสดงความสามารถพิเศษของ GAL16V8 ที่สามารถใช้งานได้กว้างขวางมาก



รูปที่ 37 ผลลัพธ์ของการทดสอบการทำงานของ CUPPL

อินพุตไพล์ที่ป้อนให้ CUPL แสดงได้ในรูปที่ 36 และชิมมูเลชันไพล์แสดงได้ในรูปที่ 37 ในส่วนของเอาต์พุตที่ต้องใช้สัญญาณนาฬิกาควบคุมในการป้อนสมการอินพุตและเอาต์พุต จะต้องเขียน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องหมาย D ต่อท้ายตัวแปร เช่น เอาต์พุตของ T-ฟลิบฟลอบ จะเขียนเป็น QTT.D รูปที่ 38 จะแสดงแผนผังของการระเบิดฟิวส์ในตัว GAL

ใช้เป็นวงจรขยายสัญญาณคล็อก (clock stretcher) วงจรนี้จะใช้เชื่อมต่อระหว่างไมโครโปรเซสเซอร์กับอุปกรณ์อินพุต, เอาต์พุตที่เป็นแบบ memory map ที่มีการทำงานช้า โดยจะสามารถถอดรหัสแอดเดรสสำหรับอุปกรณ์ได้ 5 ตัวด้วยกัน

```
/*************************************************************/
/*                                                           */
/*                  CUPL SIMULATION FILE                     */
/*       Flip-flops and latches implemented in a GAL16V8     */
/*                                                           */
/*************************************************************/

        PARTNO          456STX;
        NAME            FLIPFLOPS
        REV             1;
        DATE            4/11/86;
        DESIGNER        Jerry;
        COMPANY         Lattice Semiconductor;
        ASSEMBLY        Clock Board;
        LOCATION        U238;

/* The Order statement specifies the layout of the vector table.
        %n = n spaces inserted between variables.                 */

order:  OE,%1,CLK,%2,S,R,%1,QST,QSC,%2,PR,%1,CLR,%2,
        T,%1,QTT,QTC,%2,D,%1,QDT,QDC,%2,J,K,%1,QJT,QJC;

vectors:
/*          RS-latch          T-FF          D-FF          JK-FF     */
/*OE CLK  SR QSTQSC  PR CLR  T QTTQTC  D QDTQDC  JK QJTQJC */
   0  X   01  H  L    X  X    X  X  X    X  X  X    XX  X  X  /* set */
   0  X   10  L  H    X  X    X  X  X    X  X  X    XX  X  X  /* reset */
   0  X   11  L  H    X  X    X  X  X    X  X  X    XX  X  X  /* latch */
   0  X   10  L  H    X  X    X  X  X    X  X  X    XX  X  X  /* reset */
   0  X   01  H  L    X  X    X  X  X    X  X  X    XX  X  X  /* set */
   0  X   11  H  L    X  X    X  X  X    X  X  X    XX  X  X  /* latch */

   0  C   11  H  H    1  0    X  H  L    X  H  H    XX  H  H  /* preset */
   0  C   11  H  H    0  1    X  L  H    X  H  H    XX  H  H  /* clear */
   0  C   11  H  H    0  0    0  L  H    X  X  X    XX  X  X  /* hold */
   0  C   11  H  H    0  0    1  H  L    X  X  X    XX  X  X  /* toggle */
   0  C   11  H  H    0  0    0  H  L    X  X  X    XX  X  X  /* hold */
   0  C   11  H  H    0  0    1  L  H    X  X  X    XX  X  X  /* toggle */
   0  C   11  H  H    0  0    1  H  L    X  X  X    XX  X  X  /* toggle */

   0  C   11  H  H    1  0    X  H  H    X  H  L    XX  H  H  /* preset */
   0  C   11  H  H    0  1    X  H  H    X  L  H    XX  H  H  /* clear */
   0  C   11  H  H    0  0    X  X  X    0  L  H    XX  X  X
   0  C   11  H  H    0  0    X  X  X    1  H  L    XX  X  X  /* test */
   0  C   11  H  H    0  0    X  X  X    1  H  L    XX  X  X
   0  C   11  H  H    0  0    X  X  X    0  L  H    XX  X  X

   0  C   11  H  H    1  0    X  H  H    X  H  H    XX  H  L  /* preset */
   0  C   11  H  H    0  1    X  H  H    X  H  H    XX  L  H  /* clear */
   0  C   11  H  H    0  0    X  X  X    X  X  X    01  L  H
   0  C   11  H  H    0  0    X  X  X    X  X  X    00  L  H  /* hold */
   0  C   11  H  H    0  0    X  X  X    X  X  X    11  H  L  /* toggle */
   0  C   11  H  H    0  0    X  X  X    X  X  X    10  H  L
   0  C   11  H  H    0  0    X  X  X    X  X  X    00  H  L  /* hold */
   0  C   11  H  H    0  0    X  X  X    X  X  X    11  L  H  /* toggle */
   0  C   11  H  H    0  0    X  X  X    X  X  X    10  H  L
   0  C   11  H  H    0  0    X  X  X    X  X  X    01  L  H
```

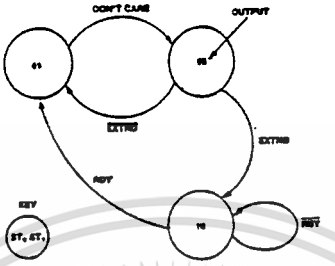
รูปที่ 38 แผนผังการระเบิดฟิวส์ของ GAL16V8

จาก state diagram การทำงานในรูปที่ 39 จะมีการทำงานอยู่ใน 3 สถานะ ถ้าไม่ได้เพิ่มตัวแปร EXTND GAL16V8 จะทำงานอยู่ระหว่างสถานะ “01” และ “00” ถ้าเพิ่มเข้ามาในช่วงที่มีการติดต่อกับอุปกรณ์ที่ทำงานช้า GAL16V8 จะไปทำงานที่สถานะ “10” และรอจนกว่าจะได้รับสัญญาณ RDY จึงจะกลับไปที่สถานะ “01” และทำงานในโหมดปกติ

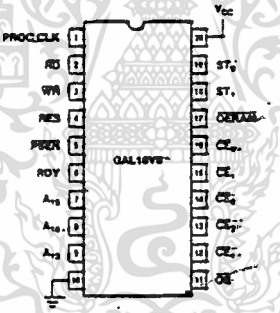
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปการจัดขาของวงจรนี้แสดงได้ในรูปที่ 40 วงจรนี้สัญญาณนาฬิกาที่ใช้ต้องมีความถี่เป็น 2 เท่าของสัญญาณคล็อกของ CPU ที่ใช้ เช่น CPU 12 Mhz ต้องใช้คล็อก 24 Mhz ซึ่งยังไม่เกินความสามารถของ GAL16V8

อินพุตของซอฟต์แวร์ CUPL แสดงได้ดังรูปที่ 41 และจำลองผลลัพธ์ของเอาต์พุตได้ในรูปที่ 42



รูปที่ 39 สเตตไดอะแกรมการทำงาน



รูปที่ 40 ลักษณะการจัดขา

2.4.13 อุปกรณ์ช่วยออกแบบ

ในอดีตเครื่องมือที่ช่วยในการออกแบบและโปรแกรมอุปกรณ์พวก PLD ยังไม่มี (GAL เป็นอุปกรณ์ประเภทหนึ่งของ PLD) ผู้ออกแบบจะต้องออกแบบลดทอนฟังก์ชันกันเอง แล้วจึงนำไปเขียนในแผนที่ตำแหน่งฟิวส์ (fuse Map) ที่มีจำนวน 1,000 ถึง 3,000 ตำแหน่งจากนั้นจึงป้อนเข้าเครื่องโปรแกรมทำการระเบิดฟิวส์ทีละตำแหน่ง เพื่อให้ได้ลอจิกฟังก์ชันตามที่ต้องการ

วิธีการดังกล่าวจะทำให้เกิดเกิดความผิดผลาดมาก บางทีก็หลงลืมระเบิดฟิวส์ได้ไม่ครบหรือเกินตำแหน่งไป ทำให้ PLD ที่โปรแกรมใช้งานไม่ได้ (เสีย) เป็นจำนวนมาก

มาในสมัยปัจจุบันจะมีซอฟต์แวร์คอมพิวเตอร์และฮาร์ดแวร์ (ตัว programmer ที่ใช้โปรแกรมข้อมูลลงในอุปกรณ์) ซึ่งการออกแบบอุปกรณ์ PLD จะมีขั้นตอนดังแสดงในรูปที่ 43

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.14 ซอฟต์แวร์

ในปัจจุบันมีซอฟต์แวร์ที่ใช้ช่วยออกแบบหลายชนิดด้วยกัน เริ่มจากระดับง่าย ๆ ที่เรียกว่าแอสเซมเบลอร์ (assembler) ซึ่งไม่ค่อยจะมีประสิทธิภาพมากนัก มาจนถึงซอฟต์แวร์ระดับสูง ๆที่เรียกว่าคอมไพเลอร์ (complier) ที่มีคุณสมบัติพิเศษช่วยทำให้การออกแบบได้ง่ายมากขึ้น

```
/*************************************************************/
/*                                                           */
/*                    CUPL INPUT FILE                        */
/*          DESIGN INPUT FOR CLOCK STRETCHER CIRCUIT         */
/*                                                           */
/*************************************************************/
/*          Allowable Target Device Types: GAL16V8           */
/*************************************************************/

                PARTNO      CLST1 ;
                NAME        CLOCK STRETCHER ;
                DATE        10/16/85 ;
                REV         01 ;
                DESIGNER    Jerry Greiner;
                COMPANY     Lattice Semiconductor;
                ASSEMBLY    ONE;
                LOCATION    U15;

/** Inputs **/

PIN 1           = PROC_CLK ;
PIN 2           = !RD ;
PIN 3           = !WR ;
PIN 4           = RES ;
PIN 5           = !PSEN ;
PIN 6           = RDY ;
PIN [7..9]      = [A15..13] ;
PIN 11          = !OE ;

/** Outputs **/

PIN 19          = ST0 ;
PIN 18          = ST1 ;
PIN 17          = !OERAM ;
PIN 16          = !CE0 ;
PIN 15          = !CE1 ;
PIN 14          = !CE2 ;
PIN 13          = !CE3 ;
PIN 12          = !CE4 ;

/** Declarations and Intermediate Variable Definitions **/

EXTND = (RD # WR) & !A15 & !A14 & !A13 ;
ENCE = (RD # WR # PSEN) ;
OERAM = RD # PSEN ;
CE0 = ENCE & !A15 & !A14 & !A13 ;
CE1 = ENCE & !A15 & !A14 & A13 ;
CE2 = ENCE & !A15 & A14 & !A13 ;
CE3 = ENCE & !A15 & A14 & A13 ;
CE4 = ENCE & A15 & !A14 & !A13 ;
ST0.D = (!ST1 & !ST0 & EXTND & !RES) # (!ST1 & ST0 & !RDY & !RES) ;
ST1.D = (!ST1 & !ST0 & !RES & !EXTND) # (!ST1 & ST0 & RDY & !RES) ;
```

รูปที่ 41 ข้อมูลที่ป้อนให้ CUPL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*************************************************************/
/*                                                           */
/*                  CUPL SIMULATION FILE                     */
/*              Simulation for Clock Stretcher               */
/*                                                           */
/*************************************************************/
/*          Allowable Target Device Types: GAL16V8           */
/*************************************************************/

               PARTNO     CLST1 ;
               NAME       CLOCK STRETCHER ;
               DATE       10/16/85 ;
               REV        01 ;
               DESIGNER   Jerry Greiner;
               COMPANY    Lattice Semiconductor;
               ASSEMBLY   ONE;
               LOCATION   U15;

ORDER:
PROC_CLK,%2,RES,%2,!RD,!WR,!PSEN,RDY,%2,A15,A14,A13,%2,!OE,
%4,ST0,ST1,%2,!OERAM,%2,!CE0,!CE1,!CE2,!CE3,!CE4 ;

VECTORS:
$msg"                              !          ";
$msg"                              O          ";
$msg"                              E  !!!!!   ";
$msg" C  R  !!SR  AAA  !           R  CCCCC   ";
$msg" L  E  RWED  111  O    SS    A  ZZZZZ   ";
$msg" K  S  DRNY  543  E    01    H  01234   ";
$msg"-------------------------------------------";

     0  1  1110  000  0     XX    H  HHHHH /* Initialization.     */
     C  1  1110  000  0     LL    H  HHHHH /* Toggle between 00,  */
     0  0  1110  000  0     LL    H  HHHHH /* 01 for normal mode  */
     C  0  1110  000  0     LH    H  HHHHH
     0  0  1110  000  0     LH    H  HHHHH
     C  0  1110  000  0     LL    H  HHHHH
     0  0  1110  000  0     LL    H  HHHHH
     C  0  1110  000  0     LH    H  HHHHH
     0  0  0110  000  0     LH    L  LHHHH /* READ active, Clock  */
     C  0  0110  000  0     LL    L  LHHHH /* Stretch required    */
     0  0  0110  000  0     LL    L  LHHHH
     C  0  0110  000  0     HL    L  LHHHH /* Wait in state 10    */
     0  0  0110  000  0     HL    L  LHHHH /* until RDY           */
     C  0  0110  000  0     HL    L  LHHHH

     0  0  0111  000  0     HL    L  LHHHH /* RDY active, resume  */
     C  0  0111  000  0     LH    L  LHHHH /* normal operation    */
     0  0  0110  000  0     LH    L  LHHHH
     C  0  0110  000  0     LL    L  LHHHH
     0  0  1110  000  0     LL    H  HHHHH
     C  0  1110  000  0     LH    H  HHHHH
     0  0  1010  001  0     LH    H  HLHHH /* WRITE (!WR) active  */
     C  0  1010  001  0     LL    H  HLHHH /* No Clock Stretch    */
     0  0  1010  001  0     LL    H  HLHHH /* required this cycle */
     C  0  1010  001  0     LH    H  HLHHH
     0  0  1010  001  0     LH    H  HLHHH
     C  0  1010  001  0     LL    H  HLHHH
     0  0  1110  000  0     LL    H  HHHHH /* !WR inactive,       */
     C  0  1110  000  0     LH    H  HHHHH /* resume normal mode. */
```

รูปที่ 42 ผลลัพธ์ของการทดสอบการทำงานของ CUPL

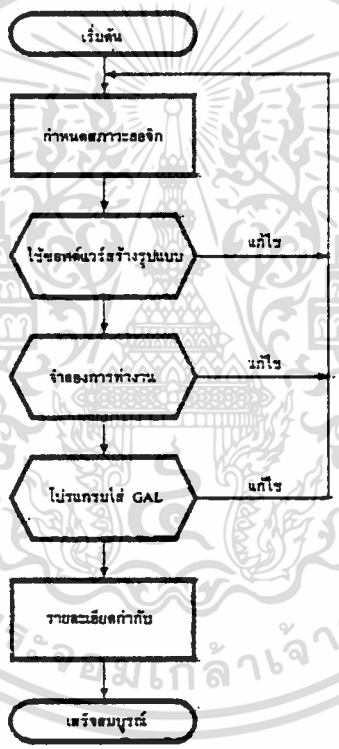
ซอฟต์แวร์คอมไพเลอร์ที่เป็นที่ยอดนิยมชื่อ CUPL และ ABEL ของบริษัท Assisted Technology และ Data I/O Corp ตามลำดับ โดยจะสามารถใช้ออกแบบอุปกรณ์ PLD ได้ทุกชนิด และยังใช้งานได้ง่ายด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.15 ฮาร์ดแวร์

การโปรแกรมหรือแพตเทอร์นิ่งอุปกรณ์ PLD เป็นขั้นตอนในการนำเอาข้อมูลไปโปรแกรมให้ PLD ทำงานตามฟังก์ชันที่เราต้องการ โดยการป้อนข้อมูลจะต้องป้อนพัลส์ที่มีคาบเวลาและระดับแรงดันที่เหมาะสมกับอุปกรณ์แต่ละตัวฮาร์ดแวร์ดังกล่าวจะมีจำหน่ายให้เลือกกันหลายแบบ ฮาร์ดแวร์ที่ใช้ควรเลือกแบบที่ใช้งานได้กว้าง ๆ (universal) และสามารถใช้โปรแกรม PLD ได้หลายชนิด

ข้อมูลที่ใช้ในการโปรแกรมจะได้จากอุปกรณ์ที่ผ่านการโปรแกรมแล้ว (master) หรือได้จากคอมพิวเตอร์ ข้อมูลที่ใช้กับเครื่องโปรแกรมส่วนใหญ่จะเป็นข้อมูลตามมาตราฐานของ JEDEC



รูปที่ 43 แสดงขั้นตอนในการออกแบบอุปกรณ์ PLD

การโปรแกรมข้อมูลของ GAL จะใช้การโปรแกรมแบบขนาน ทำให้เวลาที่ใช้ในการโปรแกรมรวดเร็วมาก (ไม่ถึงหนึ่งวินาที) หลังจากที่ผ่านการโปรแกรมแล้ว ขั้นสุดท้ายของทุก ๆ เซลล์ของ GAL จะถูกโปรแกรมแบบอนาลอก เพื่อทดสอบให้แน่ใจว่าทุกเซลล์ถูกโปรแกรมแล้วและสามารถใช้งานได้ตามที่ต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.16 ขั้นตอนการออกแบบ

ตัวอย่างที่ใช้อธิบายเป็นตัวอย่างที่ใช้ซอฟต์แวร์ CUPL ออกแบบ ส่วนซอฟ์แวร์ตัวอื่นๆ เช่น ABEL ก็จะมีโครงสร้างภาษา (syntax) และฟังก์ชันการทำงานที่ใกล้เคียงกัน จากตัวอย่างจะแสดงข้อมูลที่ป้อน และการทำงานของเครื่องโปรแกรมบนจอมอนิเตอร์ของเครื่องคอมพิวเตอร์พีซี (PC) ที่ใช้รัน CUPL ซึ่งรายละเอียดทั้งหมดจะต้องไปหาดูจากคู่มือของซอฟต์แวร์ CUPL เอง

เริ่มต้นส่วนแรกจะเป็นส่วนที่ผู้ใช้ต้องป้อนข้อมูลอุปกรณ์และชื่อของไพล์ที่ใช้ แล้วมาเป็นข้อมูลส่วนตัว เช่น ชื่อบริษัท ชื่อผู้ออกแบบ ซึ่งจะแสดงได้ดังนี้

```
/*************************************/
/*                                   */
/*     Tutorial Using a GAL16V8      */
/*                                   */
/*     Source File (431.PLD)         */
/*************************************/

     PARTNO 99;
     NAME CHAP4;
     REV 1;
     DATE 4/16/86;
     DESIGNER Dean Suhr;
     COMPANY Lattice Semiconductor;
     ASSEMBLY n/a;
     LOCATION n/a;
```

ต่อมาจะต้องกำหนดชื่อขาของอุปกรณ์ ชื่อขาที่ใช้นี้ควรจะตั้งให้มีความหมายเพื่อให้เข้าใจและจำได้ง่ายโดยตัวซอฟต์แวร์จะไม่สนใจว่าจะตั้งชื่ออะไร ยาวแค่ไหน และมีความหมายอย่างไร แสดงตัวอย่างดังนี้

```
/* inputs */
pin [1,2] = [A,!B];

/* outputs */
pin [18,19] = [Y,!Z];
```

จากตัวอย่างข้างบนจะเห็นได้ว่าจะกำหนดขาของอุปกรณ์ตามสถานะการทำงานจริงคือสัญญาณที่ขา A และ Y จะทำงานที่ลอจิก “H” ขา B และ Z ทำงานที่ลอจิก “L” การกำหนดให้ซอฟต์แวร์รับรู้จะต้องใส่เครื่องหมายตกใจ (!) หน้าชื่อตัวแปรที่ต้องการ เช่น !B และ !Z ถ้าไม่ใส่มันจะเข้าใจว่าจะทำงานที่ลอจิก “H”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนต่อมาป้อนลอจิกฟังก์ชันที่ต้องการลงไป ในอดีตเราจะต้องป้อนเป็นสมการบูลีน (Boolean equation) แต่ในซอฟต์แวร์สมัยใหม่นี้สามารถป้อนเป็นตารางการทำงาน (truth table), สเตตแมชีน (state machine) หรือการต่อเกตต่าง ๆ ก็ได้

ในตัวอย่างนี้จะป้อนเป็นสมการบูลลีน สมการที่ใช้จะมี AND ฟังก์ชันที่เอาต์พุต Y (ขา 18) และ XOR ฟังก์ชันที่เอาต์พุต Z (ขา 19) แต่เอาต์พุต Z นี้กำหนดให้ทำงานที่ลอจิก "L" ทำให้เอาต์พุตที่ 19 จะเป็นเอาต์พุตของ XNOR

```
/* logic equations */
Y = A & B;
Z = A & B  #  !A & !B;
```

ตัวโอเปอเรเตอร์ (operators) ที่ใช้ใน CUPL จะมี "!"สำหรับอินเวอร์เตอร์ และ "&" สำหรับ AND ฟังก์ชัน "#" สำหรับ OR ฟังก์ชัน สมการที่ป้อนให้สามารถป้อนได้โดยตรงเลย

ในการอินเวอร์เตอร์ให้ทำงานที่ลอจิก "L" ที่ขาอินพุตและเอาต์พุตนั้นตัวซอฟต์แวร์จะเป็นตัวแก้สมการให้ได้ตามที่ต้องการ และถ้าสมการที่ป้อนให้สามารถลดทอนลงได้ ตัวซอฟต์แวร์ที่ใช้จะลดทอนสมการให้และใช้อุปกรณ์ให้น้อยที่สุดได้

การทำงานในขั้นต่อไป CUPL จะแปลงข้อความและสามารถที่ป้อนให้เป็นข้อมูลที่สามารถใช้งานได้ โดยจะบอกชนิดของอุปกรณ์ที่ใช้ (G16V8 ในตัวอย่าง) ชื่อของแฟ้มข้อมูล ("401" ในตัวอย่าง) และฟังก์ชันที่เพิ่มเข้าไปซึ่งผู้ใช้จำเป็นต้องใช้ และจำนวนแฟลก (flag) ที่ใช้ ("-jlfxs") และเวลาที่ใช้ในแปลข้อมูล ซึ่งจะแสดงได้ดังนี้

```
CUPL -jlfxs G16V8 401

CUPL Version 2.10B1 Copyright (c)
1983,84,85 Assisted Technology,
Inc.

cuplx
time: 3 secs

cupla
time: 19 secs

cuplb
time: 7 secs

cuplm
time: 4 secs

cuplc
time: 15 secs

csima
time: 24 secs

total time: 73 secs
```

ขั้นสุดทาย้ CUPL จะพิมพ์รายงานผลลัพธ์ที่เรียกว่า doccumentation ไพล์ที่ประกอบด้วยสมการในขั้นสุดท้าย (ที่ผ่านการลดทอนแล้ว) แผนผังของเซลล์ (cell map) หรือฟิวส์พล็อต (fuse plot) ซึ่งแสดงตำแหน่งที่ต้องระเบิดฟิวส์และตำแหน่งของขาอุปกรณ์ (ถ้าต้องการ) ดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Device G16V8s  Library DLIB-d-55-8
Created        Wed Apr 16 04:52:24 1986

--------------------------------------------
          Expanded Product Terms
--------------------------------------------

Y => A & B

Z => A & B   #   !A & !B

--------------------------------------------
                Symbol Table
--------------------------------------------

Pin Variable          Pterms   Max
Pol   Name    Pin     Used     Pterms
--- --------  ---     -----    ------
    A          1        -        -
 !  B          2        -        -
    Y         18        1        8
 !  Z         19        2        8

--------------------------------------------
                  Fuse Plot
--------------------------------------------

Syn   2192 - Ac0   2193 x

Pin #19   2048   Pol x   2120   Ac1 x

 0000 -xx--------------------------------
 0032 x--x--------------------------------
 0064 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0096 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0128 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0160 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0192 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0224 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

Pin #18   2049   Pol -   2121   Ac1 x
 0256 -xx--------------------------------
 0288 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0320 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0352 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0384 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0416 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0448 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
 0480 xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

      (continued ...)

LEGEND    X : fuse not blown
          - : fuse blown
```

ข้อมูลที่ใช้ในการโปรแกรม PLD ที่ใช้กันมากจะต้องเป็นไปตามมาตรฐาน JEDEC จากข้อมูลข้างต้นเมื่อนำมาแปลงเป็น JEDEC file จะมีข้อมูลดังรูปข้างล่างนี้ ข้อมูลของ JEDEC นี้จะมีข้อมูลของบิตควบคุม ซึ่งแสดงการทำงานของซีเคียวริตี้เซลล์และมี checksum ที่ใช้ตรวจสอบความถูกต้องของข้อมูล

```
*QP20
*QF2194
*G0
*F0
*L0000 10011111111111111111111111111111
*L0032 01101111111111111111111111111111
*L0256 10011111111111111111111111111111
*L2048 01000000000000000000000000000000
*L2112 00000000001111111111111111111111
*L2144 11111111111111111111111111111111
*L2176 11111111111111110
*C1404
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เทสเวกเตอร์จะแสดงถึงเอาต์พุตที่ได้จากฟังก์ชันที่ป้อนให้โปรแกรมซอฟต์แวร์ใน source file โดยจะทดลองป้อนอินพุตที่เป็นไปได้ แล้วแสดงเอาต์พุตของฟังก์ชันที่ป้อนให้เพื่อให้ผู้ออกแบบตรวจสอบว่าผลลัพธ์ที่ได้ตรงตามความต้องการหรือไม่ ข้อมูลที่ได้จะเป็นดังนี้

```
/* The Order Statement specifies the
   layout of the function vector
   table.

   %n = n spaces inserted between
   variables.  */

   order: A,B,%1,Y,%1,!2;

   vectors:

/* AB Y Z */

   00 L X /* test AND gate */
   01 L X
   10 L X
   11 H 0
   00 X 1 /* test XNOR gate */
   01 X 0
   10 X 0
   11 X 1
```

ขั้นตอนการซิมมูเลตจะทดลองนำอินพุตที่เป็นไปได้ป้อนลงในฟังก์ชันที่ผ่านการลดทอนโดยใช้ซอฟต์แวร์ เพื่อนำผลที่ได้ไปเปรียบเทียบกับฟังก์ชันเดิมที่ยังไม่ลดทอน ซึ่งจะแสดงผลลัพธ์ของเอาต์พุตได้ดังนี้ (เอาต์พุตที่ได้ถ้าเป็น X จะหมายถึงไม่สนใจ (don't care) ถ้าเป็น N จะหมายถึง ไม่ได้ทดสอบ (not test)

```
==========================
Simulation Results
==========================
     0001: 00 L N
     0002: 01 L N
     0003: 10 L N
     0004: 11 H N
     0005: 00 N H
     0006: 01 N L
     0007: 10 N L
     0008: 11 N H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
ขั้นตอนการออกแบบและการสร้าง

3.1 ส่วนของ วงจร

เนื่องจากส่วนวงจรที่จะต้องสร้างมี 2 ส่วนใหญ่ ๆ เราจึงแยกที่จะกล่าวทีละส่วนดังนี้

3.1.1 ส่วนควบคุมอัตโนมัติ ซึ่งสามารถแบ่งการทำงานออกเป็น 2 ส่วนย่อย
- วงจรเข้ารหัส (ENCODER)
- วงจรถอดรหัส (DECODER)

3.1.2 ส่วนควบคุมปลายทาง ซึ่งสามารถแบ่งการทำงานออกเป็นแต่ละส่วนย่อยดังนี้
- ส่วนเข้ารหัส (ENCODER)
- ส่วนถอดรหัส (DECODER)
- ปุ่มสวิทช์ (MANUAL SWITCII)
- ส่วนขับรีเลย์
- ชุดสวิทช์ฉุกเฉิน

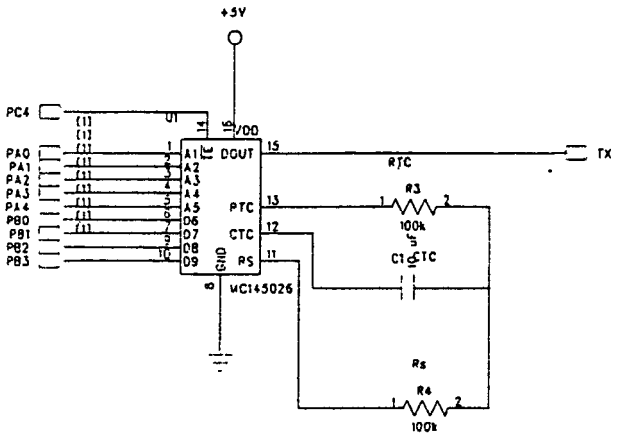
การทำงานของแต่ละส่วนมีดังนี้

3.1.1 ส่วนควบคุมอัตโนมัติ

วงจรส่วนนี้จะวางอยู่ในกล่องเดียวกับชุดไมโครโปรเซสเซอร์ ซึ่งวงจรส่วนนี้จะต่ออยู่กับแผงวงจรชุดไมโครโปรเซสเซอร์ ต่อทางพอร์ท (PORT) อินพุท (INPUT) เอาท์พุท (OUTPUT) ของไมโครโปรเซสเซอร์ Z84C11 ใช้ 3 พอร์ท คือ Pa,Pb,Pc ซึ่งกำหนดให้พอร์ท Pa0 - Pa7 เป็นเอาท์พุท Pb0 - Pb3 เป็นเอาท์พุท Pb4 - Pb7 เป็นอินพุท และ Pc0 - Pc3 เป็นเอาท์พุท Pc4- Pc7 เป็นอินพุท การทำงานมี 2 ส่วนดังนี้

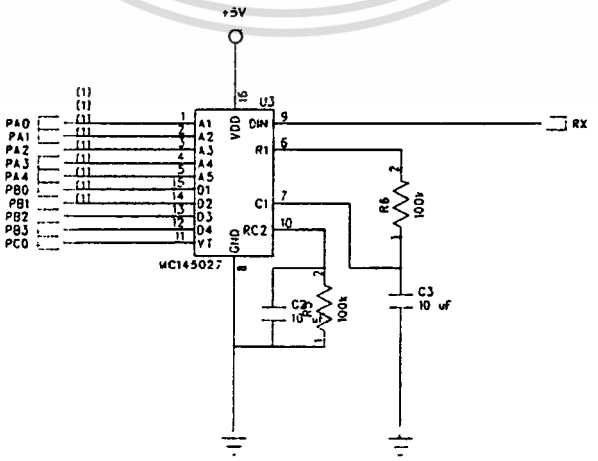
วงจรเข้ารหัส แสดงในรูปที่ 44 จะใช้ไอซีเบอร์ MC145026 และตัวความต้านทาน,ตัวเก็บประจุ ส่วนนี้จะนำข้อมูลทาง ไอโอพอร์ท (I/O Port) ของ Z84C11 มาเป็นข้อมูลในการควบคุมอุปกรณ์ไฟฟ้าและสวิทช์ในเซ็นเซอร์ มีสายสัญญาณรวม 9 เส้น เราจึงใช้ตัวเข้ารหัสคือ MC145026 ซึ่งจะแปลงข้อมูล 9 บิท (A1 - A5, D6 -D9) ขนานมาเป็นข้อมูลแแบอนุกรมแล้วส่งออกไปตามสายซึ่งจะเป็นแบบมัลติพอยต์ (Multipoint) ซึ่งส่งข้อมูลไปให้ส่วนควบคุมอุปกรณ์ปลายทาง สังเกตว่า 5 บิทแรกคือ Pa0 - Pa4 จะเป็น A1 - A5 ซึ่งมีความแตกต่างถึง 32 ค่าเราจึงกำหนดให้แต่ละค่าเป็นแต่ละห้องคือควบคุมได้ 32 ห้อง และ Pb0 - Pb4 จะเป็น D6 - D9 จะมีค่าแตกต่างถึง 16 ค่าแต่เราจะใช้ค่า 0-7 ในการควบคุมอุปกรณ์ไฟฟ้า ของในแต่ละห้องและ 08-0B ควบคุมสวิทช์ฉุกเฉินของแต่ละห้อง อีก 4 ค่าคือ 0C-0F จะใช้ควบคุมการเช็คสถานะอุปกรณ์ปลายทางกลับ ซึ่งจะทำให้ควบคุมอุปกรณ์ไฟฟ้า และสวิทช์ฉุกเฉินรวม 32 ห้องคืออุปกรณ์ไฟฟ้า 32 x 8 = 256 จุดและสวิทช์ฉุกเฉิน 32 x 4 = 128 จุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



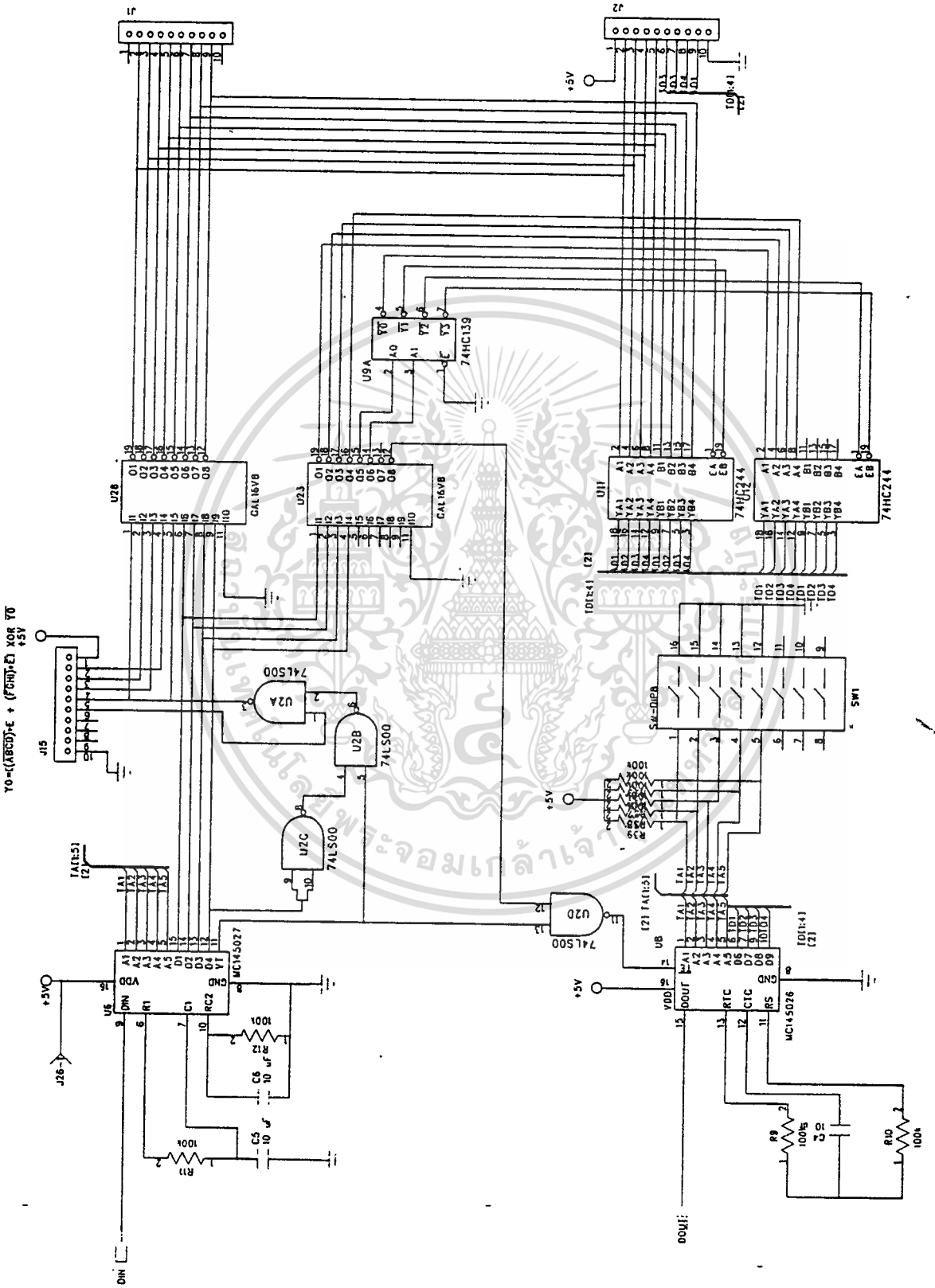
รูปที่ 44 แสดงส่วนของ วงจรเข้ารหัส (ENCODER)

วงจรถอดรหัส แสดงในรูป 45 จะใช้ไอซีเบอร์ MC145027 1 ตัว ร่วมกับความต้านทานและตัวเก็บประจุ ซึ่งส่วนนี้จะนำข้อมูลออกทาง ไอโอพอร์ทของ Z84C11 มาทำการถอดรหัส โดย A1 - A5 จะต่ออยู่กับ Pa0 - Pa4 จะเป็นตัวเลือกว่าจะรับข้อมูลจากห้องใด โดย D6 - D9 จะต่ออยู่กับพอร์ท Pb4 - Pb7 จะเป็นข้อมูลบอกสถานะปลายทางว่าอุปกรณ์กำลังทำงานอยู่หรือไม่ ส่วนขา VT ของ MC145027 จะตัวบอกว่าได้รับข้อมูลแล้วซึ่งถ้าเป็นระดับลอจิก "1" แสดงว่า MC145027 สามารถถอดรหัสได้แล้ว และให้ CPU นำเข้าไปประมวลผลการทำงานคือ CPU จะนำค่าของห้อง (A1 - A5) แล้วส่งรหัส 0E มาที่ขา (Pb0 - Pb3) D6 - D9 เพื่อที่จะทำการดึงข้อมูลของอุปกรณ์ 4 ตัวแรก (0 - 3) แล้วรอรับสถานะของอุปกรณ์แต่ละตัวตามลำดับ เมื่อได้รับสถานะของอุปกรณ์ 4 ตัวแรกแล้ว CPU จะส่งรหัส 0C ที่ขา D6 - D7 เพื่อเช็คสถานะของอุปกรณ์ 4 ตัวต่อมา(4 - 7) และส่งรหัส 0E ตามด้วย 0D เพื่อสถานะของสวิทช์เซ็นเซอร์ว่ามีการทำงานอยู่หรือไม่ แล้ว CPU ส่งรหัส 0C อีกเพื่อเช็คสถานะว่าสวิทช์ฉุกเฉินตัวที่กำลังทำงานอยู่มีการตรวจรับสิ่งที่ต้องการตรวจรับได้หรือไม่



รูปที่ 45 แสดง ส่วนของวงจรถอดรหัส (DECODER)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 46 แสดง ส่วนของวงจรควบคุมอุปกรณ์ปลายทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 ส่วนควบคุมปลายทาง

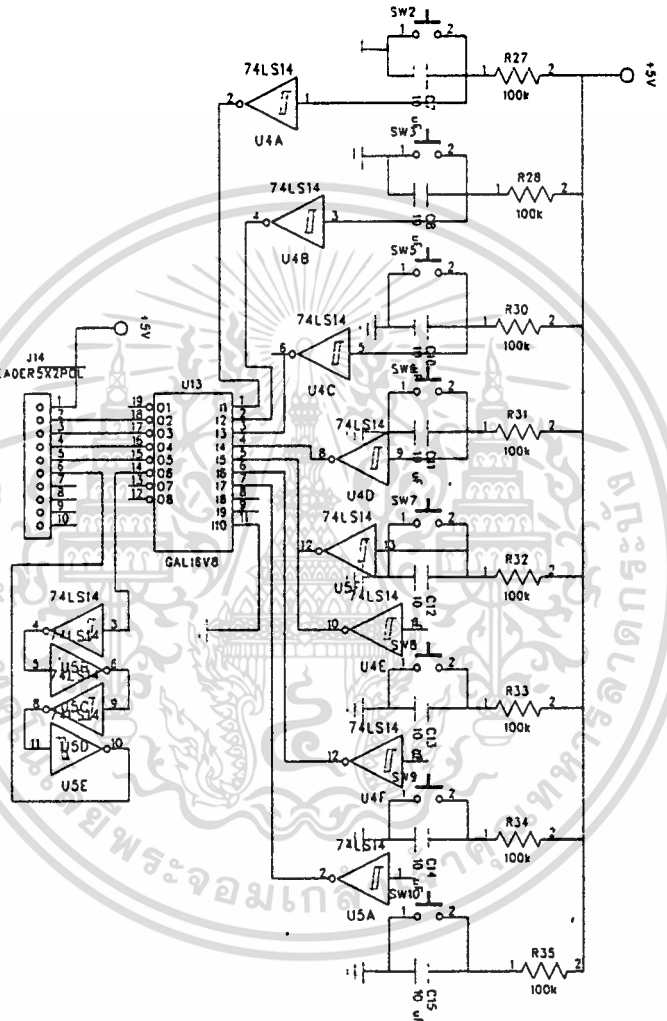
ส่วนนี้แสดงในรูปที่ 46 ซึ่งจะทำหน้าที่แปลงข้อมูลอนุกรมที่รับมาจากสายส่งมาเป็นข้อมูลแบบขนาน โดยปลายทางแต่ละจุดจะมีตำแหน่ง (A1 - A5) ที่แน่นอนแล้วคอยรับข้อมูล D6 - D9 เพื่อส่งให้อุปกรณ์และสวิทช์ฉุกเฉินทำงานหรือไม่ทำงานและส่งสถานะกลับของอุปกรณ์และสวิทช์ฉุกเฉินและการตรวจจับแบ่งออกเป็นดังนี้

ส่วนถอดรหัส ตามรูปที่ 46 ที่ขา A1 - A5 ของ MC145027 จะต่ออยู่กับดิฟสวิทช์คงที่อยู่ซึ่งเป็นค่าที่ใช้กำหนดตำแหน่งห้อง ส่วน D6 - D7 จะต่อเข้ากับอินพุทของไอซีเกทแอเรย์ลอจิกหรือไอซีแกล (GAL = Gate Array Logic) ที่ขา 6,7,8,9 ทั้ง 2 ตัวของไอซีแกล และสัญญาน VT จาก MC145027 จะเป็นสัญญาณที่ใช้เป็นอินพุทของไอซีแกลทั้ง 2 ตัว ซึ่งไอซีแกลทั้ง 2 ตัวเป็นตัวดีมัลติเพล็กซ์ เอาเอาท์พุทไปขับเข้าวงจรขับโหลด (Load) แต่ในส่วนอินพุทของแกลตัวที่หนึ่งจะรับอินพุทจากคีย์สวิทช์ด้วยและจะมีความสำคัญของคีย์สวิทช์สูงกว่าสัญญาณที่มาจากชุดควบคุมอัตโนมัติ ส่วนเอาท์พุทแกลตัวทีสองที่เอาท์พุทล่าง (ขา 15,14,13,12) จะเป็นชุดควบคุมในการสร้างสถานะของอุปกรณ์และสวิทช์ฉุกเฉินและการเซ็นเซอร์กลับ

วงจรเข้ารหัส ตามรูปที่ 46 ส่วนนี้จะทำหน้าที่ส่งข้อมูลสถานะของอุปกรณ์ สวิทช์ฉุกเฉินและเซ็นเซอร์กลับไปให้ส่วนควบคุมอัตโนมัติ ต่อให้บอร์ด CPU อ่านค่าไป แต่ในส่วนนี้ สามารถส่งสถานะกลับได้ครั้งละ 4 ช่วงเท่านั้นการเลือกว่าจะส่งสถานะช่วงใดบ้างจะบังคับ โดยไอซีเบอร์ 74LS139 จะเป็นตัวต่อให้ไอซีเบอร์ 74LS244 ทำงานเพื่อดึงสถานะกลับ.

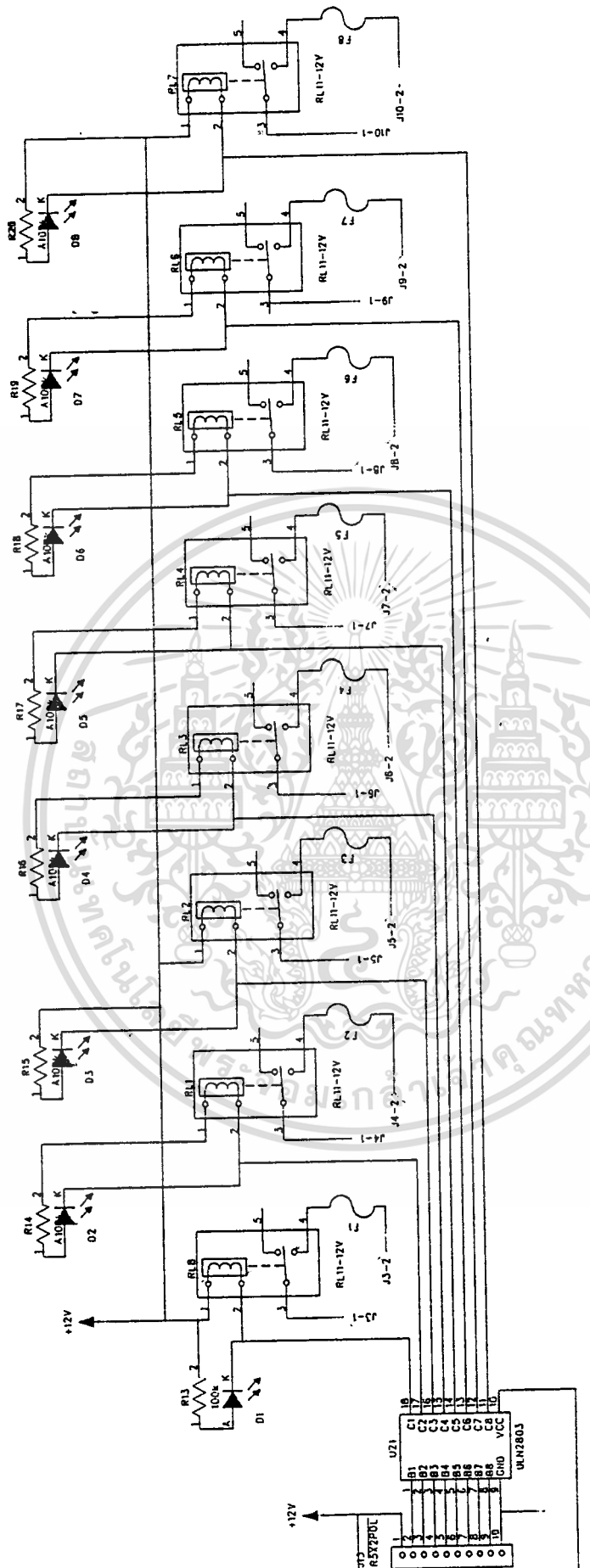
วงจรคีย์สวิทช์ (Manual Switch) ตามรูปที่ 47 จะทำหน้าที่รับการกดสวิทช์จากผู้ใช้ว่าต้องการจะเปิดหรือปิดอุปกรณ์ไฟฟ้าตัวใดซึ่งวงจรนี้จะเป็นวงจรมัลติเพล็กซ์ 8 อินพุท 3 เอาท์พุท สัญญาณเอาท์พุทในวงจรนี้จะไปต่อเป็นอินพุทของไอซีแกลตัวที่หนึ่ง วงจรรับคีย์สวิทช์จะเป็น R และ C เพื่อป้องกันสัญญาณรบกวนจากการกดคีย์เอาสัญญาณทั้ง 8 ช่องไปเข้าเป็นอินพุทของไอซีแกลตัวที่สาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 47 แสดง ส่วนของวงจรคีย์สวิทช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

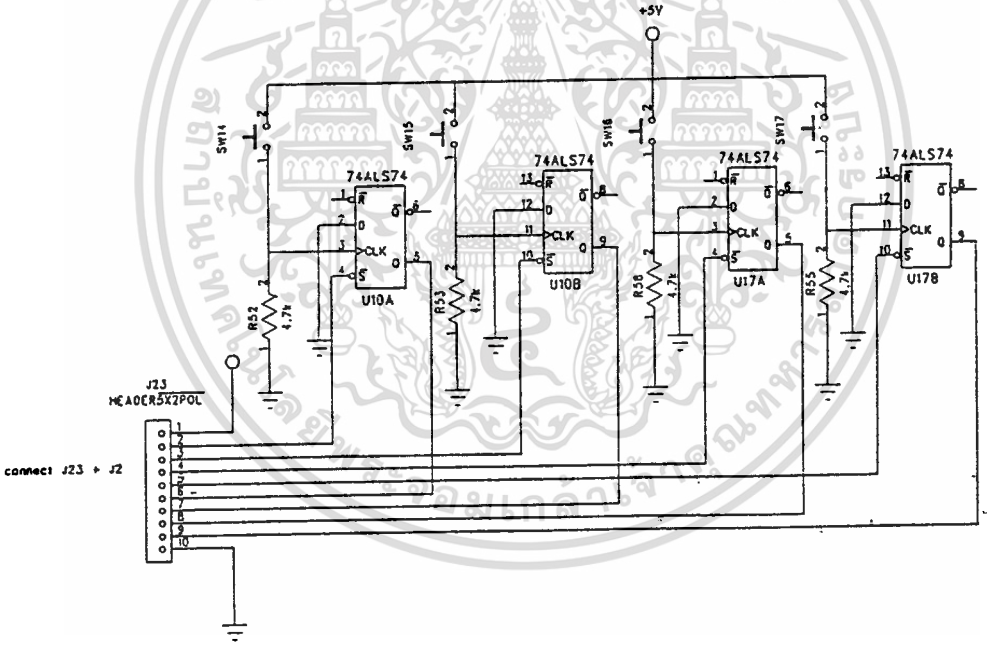


รูปที่ 48 แสดงส่วนของ วงจรขับโหลด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรขับโหลด(Load Driver) ตามรูปที่ 48 โหลดจะเป็นรีเลย์ 8 ตัว ซึ่งรีเลย์แต่ละตัวจะกินกระแสประมาณ 30 mA ซึ่งรีเลย์ 8 ตัวจะกินกระแสประมาณ 240 mAเราจึงใช้ไอซีเบอร์ UNL2803 เป็นตัวขับรีเลย์ซึ่งไอซีตัวนี้เป็นไอซีที่ใช้ขับรีเลย์ได้โดยตรงโดยไม่ต้องต่ออุปกรณ์อื่นเพิ่มเติมและเบอร์นี้มีชุดขับรีเลย์ได้ 8 ตัวพอดี ซึ่งไอซีเบอร์นี้รับแรงดัน Vcc ได้ 30 โวลท์ และทนกระแสได้ 500 mA

วงจรสวิทช์ฉุกเฉิน ตามรูปที่ 49 เราได้ตั้งเงื่อนไขการทำงานของวงจรนี้ไว้ว่า หน้าสัมผัสของสวิทช์ฉุกเฉินนี้แตะกันเมื่อไร ถึงแม้ว่าหน้าสัมผัสจะแยกจากกันแล้วก็ตามเอาท์พุทของวงจรนี้ที่เป็นการทำงานทันที เราจึงออกแบบโดยใช้ ดีฟลิปฟลอป(D-Flip Flop) คือเมื่อหน้าสัมผัสแตะกันจะเป็นการจ่ายสัญญาณรอรับลอจิก "1" ให้กับขาคลอ็ก(Clock)ไน ดีฟลิปฟลอป ทำให้เอาท์พุทของฟลิปฟลอปเป็น "0" ถึงแม้ว่าสวิทช์หน้าสัมจะแยกจาก กันแล้วก็ตาม ถ้าจะทำให้สวิทช์ฉุกเฉินเริ่มทำงานใหม่อีกครั้งต้องสั่งงานทางคอมพิวเตอร์สวิทช์ฉุกเฉินจึงจะมีการเริ่มทำงานใหม่

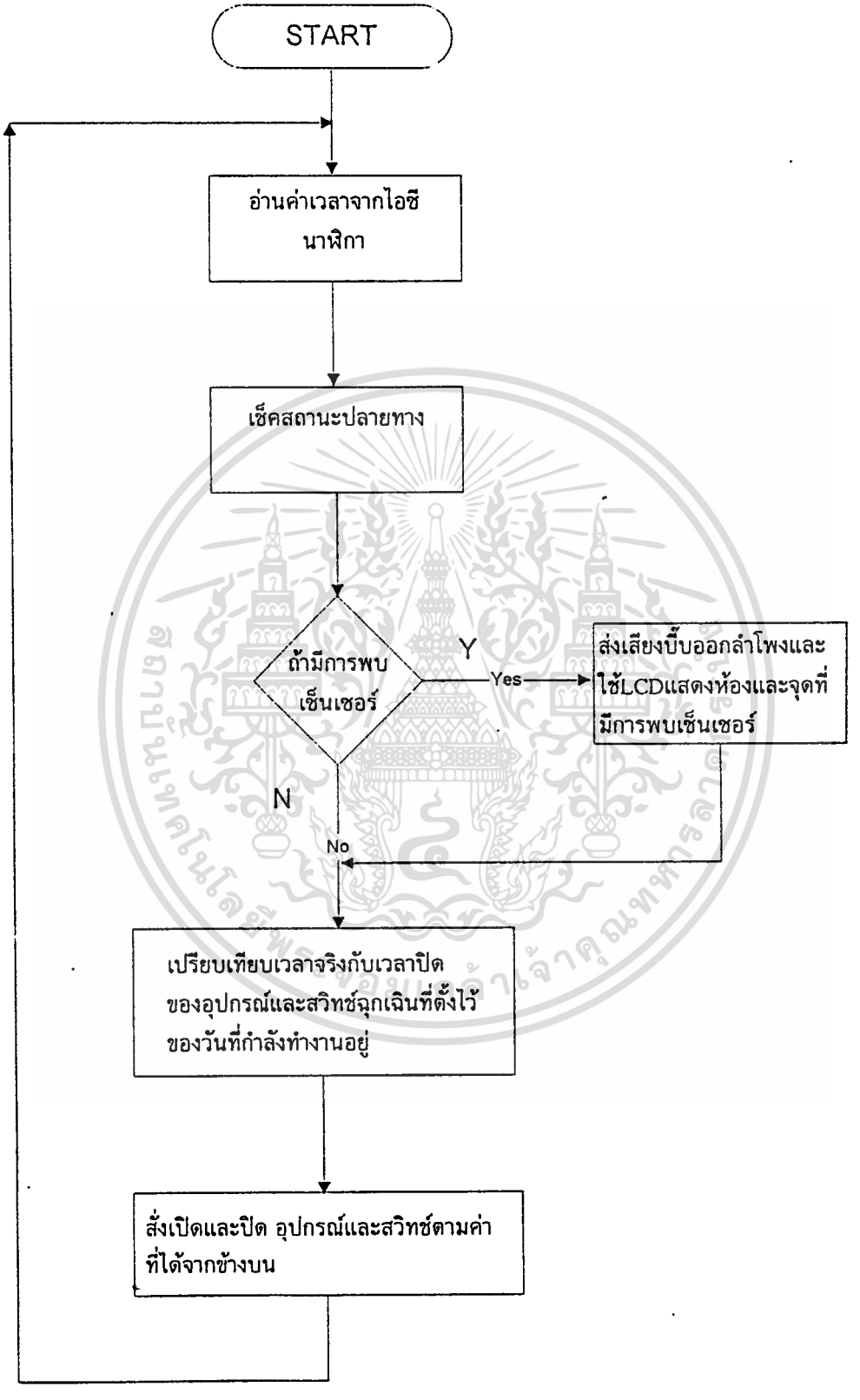


รูปที่ 49 แสดงส่วนของ วงจรสวิทช์ฉุกเฉิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ส่วนของ โปรแกรม

3.2.1 การทำงานของโปรแกรมบนบอร์ด CP-Z84C11 PLUS

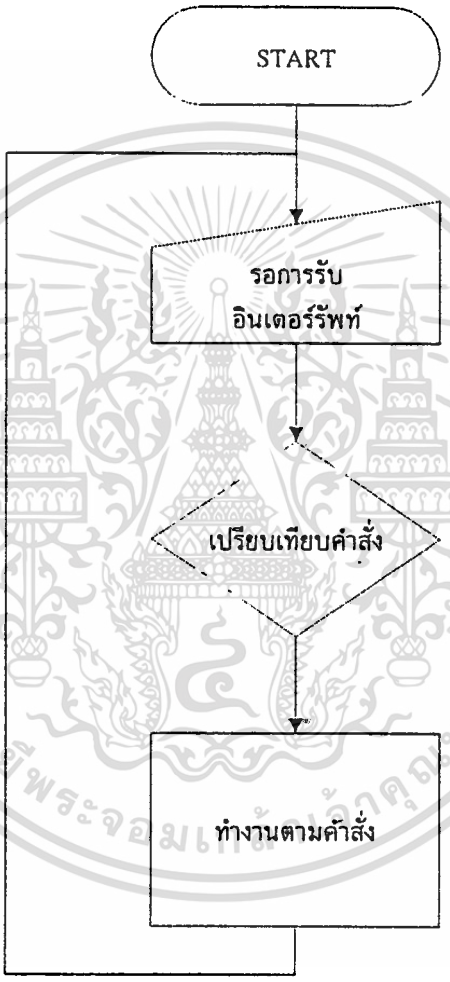


รูปที่ 50 แสดง โพล์ชารท์ (flow chart) การทำงานโปรแกรมบนบอร์ด CP-Z84C11 PLUS

Note : ส่วนของโปรแกรมบนบอร์ด Z84C11 จะอยู่ในส่วนของภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 การทำงานของโปรแกรมติดต่อกับเครื่องคอมพิวเตอร์ พีซี



รูปที่ 51 แสดง โฟลว์ ชาร์ต (Flow Chart) ของโปรแกรมที่จะติดต่อกับเครื่องคอมพิวเตอร์ พีซี

Note : การทำงานของโปรแกรมบนบอร์ด Z84C11 ใน 1 รอบจะใช้เวลาประมาณ 1 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 การใช้งานโปรแกรมติดต่อกับเครื่องคอมพิวเตอร์ พีซี

ส่วนของ โปรแกรม ที่ใช้ติดต่อกับเครื่องคอมพิวเตอร์พีซี (PC) ได้แก่โปรแกรมพีซีพลัสซ์ (PCPLUS) ซึ่งการใช้โปรแกรมเมนูบนพีซีพลัสซ์ (PCPLUS) มีดังนี้

ขั้นตอนแรกให้ต่อวงจรให้เสร็จเรียบร้อยก่อน หลังจากนั้นให้เรียกใช้โปรแกรม โปรคอมพลัสซ์(Procom plus) ซึ่งเป็นโปรแกรมที่ใช้ในการติดต่อสื่อสารระหว่าง เครื่องคอมพิวเตอร์ พีซี กับตัว
ควบคุมอัตโนมัติโดยผ่านทาง พอร์ทอนุกรม (RS232) หน้าจอแรกที่จะปรากฏคือ เมนูหลักตามที่ปรากฏอยู่ข้างล่างนี้

********** W E L C O M E **********

A - SHOW STATUS
B - SET TIME
C - SET TIME ON AND OFF FOR EACH ROOM
D - DIRECT CONTROL EQUIPMENT
E - CLEAR ALARM SENSOR SWITCH
F - CLEAR SCREEN LCD PANEL
G - SETUP TIMER SWITCH SENSOR(S)
H - DIRECT CONTROL SWITCH SENSOR

SELECT> _

เมื่อปรากฏหน้าจอตามข้างบนแล้ว โปรแกรมจะรอรับการเลือก โดยจะแสดงเครื่องหมายพร้อมคือ "SELECT>_" ถ้าต้องการที่จะเลือกเมนูอันไหนก็สามารถเลือกได้โดยการกดตัวอักษรตัวที่ตรงกับตัวอักษรกำกับเมนู จากรูปจะเห็นว่ามี เมนูให้เลือกทั้งหมด 8 เมนู ซึ่งแต่ละเมนูสามารถอธิบายได้ดังนี้

A - SHOW STATUS

เลือกโดยการกด ตัวอักษร A หน้าที่ของเมนูนี้คือ แสดงค่าสถานะของ อุปกรณ์, และสวิทช์ฉุกเฉินที่จะควบคุม และการแสดงอลาม (ALARM) ของสวิทช์ฉุกเฉิน

B - SET TIME

เลือกโดยการกดตัวอักษร B หน้าที่ของเมนูนี้คือตั้งเวลานาฬิกาจริง (Real Time) ของระบบ

C - SET TIME ON AND OFF FOR EACH ROOM

เลือกโดยการกดตัวอักษร C หน้าที่ของเมนูนี้คือ ตั้งเวลาเปิดและปิดของอุปกรณ์ที่จะทำการควบคุมในแต่ละห้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D - DIRECT CONTROL EQUIPMENT

เลือกโดยการกดตัวอักษร D หน้าที่ของเมนูนี้คือ การควบคุมอุปกรณ์โดยตรงโดยไม่ต้องตั้งเวลา

E - CLEAR ALARM SENSOR SWITCH

เลือกโดยการกดตัวอักษร E หน้าที่ของเมนูนี้คือ ทำการ เคลียร์(CLEAR)อลาม ให้สวิทช์ฉุกเฉินมีการทำงานตามปกติ

F - CLEAR SCREEN LCD PANEL

เลือกโดยการกดตัวอักษร F หน้าที่ของเมนูนี้คือ เคลียร์ หน้าจอภาพ แอลซีดี(LCD)

G - SETUP TIMER SWITCH SENSOR(S)

เลือกโดยการกดตัวอักษร G หน้าที่ของเมนูนี้คือตั้งเวลาของสวิทช์ฉุกเฉิน

H - DIRECT CONTROL SWITCH SENSOR

เลือกโดยการกดตัวอักษร H หน้าที่ของเมนูนี้คือ ทำการควบคุมการทำงานของสวิทช์ฉุกเฉิน โดยตรง

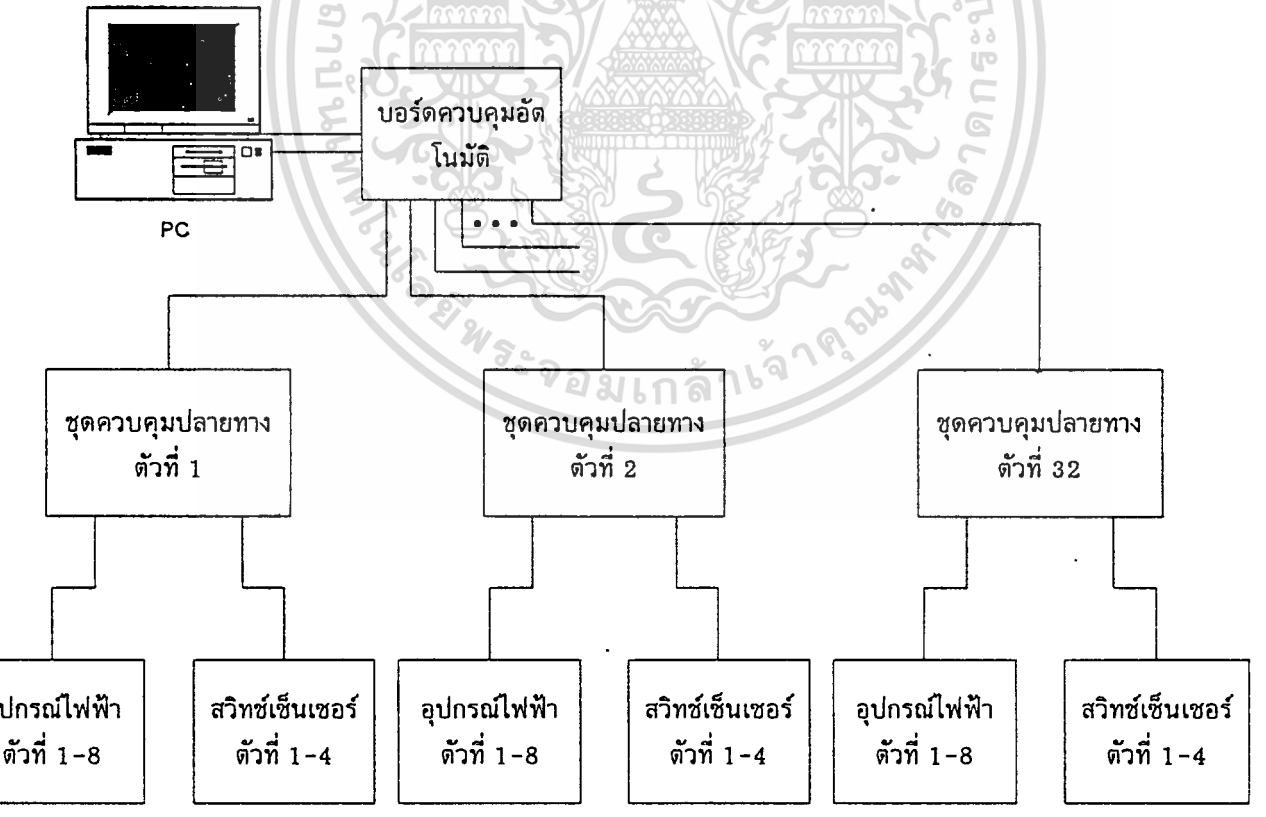
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
การทดลองและผลการทดลอง

วัตถุประสงค์

1. เพื่อทำการทดสอบการสั่ง เปิด ปิด อุปกรณ์ ไฟฟ้าจากส่วนควบคุมปลายทางได้
2. เพื่อทำการทดสอบการสั่ง เปิด ปิด อุปกรณ์ ไฟฟ้าจากส่วนควบคุมอัตโนมัติได้
3. เพื่อทำการทดสอบสวิทช์ฉุกเฉินให้ทำงานได้ตามที่ต้องการ
4. เพื่อทำการทดสอบการตั้งเวลา เปิด ปิด อุปกรณ์ไฟฟ้าจากส่วนควบคุอัตโนมัติได
5. เพื่อทำการทดสอบการตั้งเวลา ให้สวิทช์ฉุกเฉินทำงานจากส่วนควบคุมอัติโนมัติได้
6. เพื่อทำการทดสอบ การแสดงสถานะของอุปกรณ์ต่าง ๆ
7. เพื่อทำการแก้ไขส่วนของวงจร และโปรแกรม ให้ทำงานได้ตามที่ต้องการ

ดังนั้นเพื่อให้ได้ผลการทดลองสำเร็จตามวัตถุประสงค์ที่ตั้งไว้ จะแบ่งการทดลองทั้งหมดออกเป็น 2 ส่วนคือ ส่วนควบคุมปลายทาง และส่วนควบคุมอัตโนมัติ โดยการทดลองทั้งสองส่วนจะต่ออุปรณ์การทดลองตามรูปที่ 52 ดังนี้



รูปที่ 52 แสดง ส่วนของการต่ออุปกรณ์เพื่อทำการทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 ส่วนควบคุมปลายทาง

การทดลองในส่วนนี้จะประกอบด้วยส่วนควบคุมปลายทางอยู่สองตัว โดยเราสามารถที่จะทำการสั่ง เปิดและปิด อุปกรณ์ไฟฟ้าได้โดยตรงจากส่วนนี้ และสามารถทำการทดลองได้ ดังนี้

ทดลองที่ชุดควบคุมปลายทางตัวที่ 1 กดปุ่มสวิทช์ตัวที่ 1 ทำให้หลอดไฟติด กดอีกครั้งหนึ่งทำให้หลอดไฟดับ เสร็จแล้วทำการทดสอบปุ่มสวิทช์ต่อ ผลที่ออกมาก็จะเหมือนกับครั้งแรก ทำอย่างเดียวกันจนครบสวิทช์ทั้ง 8 ตัว และได้ผลตามต้องการ

ทดลองที่ชุดควบคุมปลายทาง ตัวที่ 2 จะทำเหมือนกับชุดควบคุมปลายทางตัวที่ 1 ได้ผลถูกต้องตามต้องการ

ผลการทดลอง

จากการทดลองส่วนควบคุมปลายทางตัวที 1 และตัวที่ 2 สามารถทำการสั่ง เปิดและปิดอุปกรณ์ไฟฟ้าได้ตามที่ต้องการและสถานะที่แสดงการเปิดและปิด ของอุปกรณ์ไฟฟ้านั้นก็แสดงถูกต้อง ดังนั้นส่วนนี้เราสามารถที่จะนำไปต่อกับส่วนควบคุมอัตโนมัติได้ เพื่อทำการทดลองต่อไป

4.2 ส่วนควบคุมอัตโนมัติ

การทดลองในส่วนนี้จะประกอบด้วย ส่วนของเครื่องคอมพิวเตอร์พีซี ,ส่วนควบคุมอัตโนมัติ, ส่วนควบคุมปลายทาง,ส่วนของอุปกรณ์ไฟฟ้า,ส่วนของสวิทช์เซ็นเซอร์

การทดลองนี้จะใช้การทดลองตามการเรียกใช้โปรแกรมเมนูบน พีซีพลัสซ์ (PCPLUS) โดยขั้นตอนแรกให้ต่ออุปกรณ์ตามรูปที่ 52 ให้เสร็จเรียบร้อยก่อน หลังจากนั้นให้เรียกใช้โปรแกรม
โปรคอมพลัสซ์ (Procom plus) ซึ่งเป็นโปรแกรมที่ใช้ในการติดต่อสื่อสารระหว่างเครื่อง พีซี กับตัวควบคุมอัตโนมัติโดยผ่านทาง พอร์ทอนุกรม(RS232) หน้าจอแรกที่จะปรากฏคือ เมนูหลักตามที่ปรากฏอยู่ข้างล่างนี้

********** W E L C O M E **********

A - SHOW STATUS
B - SET TIME
C - SET TIME ON AND OFF FOR EACH ROOM
D - DIRECT CONTROL EQUIPMENT
E - CLEAR ALARM SENSOR SWITCH
F - CLEAR SCREEN LCD PANEL
G - SETUP TIMER SWITCH SENSOR(S)
H - DIRECT CONTROL SWITCH SENSOR

SELECT> _

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นเราสามารถที่จะทำการทดลองตามเมนูที่ปรากฏขึ้นดังนี้

ทำการทดลอง การแสดงค่าสถานะ ของอุปกรณ์, และเซ็นเซอร์สวิทช์ฉุกเฉินที่จะควบคุมและการอลาม (ALARM) ของสวิทช์ฉุกเฉิน จากเมนูทำการทดลองโดยการกด ตัวอักษร A ซึ่งจะได้ผลแสดงดังนี้

********** W E L C O M E **********

A - SHOW STATUS

B - SET TIME

C - SET TIME ON AND OFF FOR EACH ROOM

D - DIRECT CONTROL EQUIPMENT

E - CLEAR ALARM SENSOR SWITCH

F - CLEAR SCREEN LCD PANEL

G - SETUP TIMER SWITCH SENSOR(S)

H - DIRECT CONTROL SWITCH SENSOR

SELECT> A

[Pxx = POINT] [Sxx = SENSOR SW.] [Axx = ALARM]

ADDRESS P01 P02 P03 P04 P05 P06 P07 P08 S01 S02 S03 S04 A01 A02 A03 A04

00 NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT

01 NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT

02 NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT

03 NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT

04 NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT

05 NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT

06 NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT

07 NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT

08 NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT

09 NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT

10 NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT

11 NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT

12 NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT

13 NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT

14 NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT

15 NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT

16 NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT NOT

PLEASE ANY KEY TO CONTINUE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ADDRESS	P01	P02	P03	P04	P05	P06	P07	P08	S01	S02	S03	S04	A01	A02	A03	A04
17	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT
18	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT
19	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT
20	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT
21	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT
22	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT
23	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT
24	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT
25	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT
26	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT
27	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT
28	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT
29	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT
30	OFF	OFF	OFF	OFF	ON	ON	ON	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF	OFF
31	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT	NOT

ผลการทดลอง

จากการทดลองสามารถที่จะแสดงตำแหน่งของส่วนปลายทางได้ 32 แอดเดรส หรือ 32 ห้อง คือตั้งแต่แอดเดรสที่ 00 จนถึงแอดเดรสที่ 31 แสดงส่วนของอุปกรณ์ไฟฟ้าที่จะนำมาต่อใช้งานได้ 8 ตัว คือตั้งแต่ P01 - P08 แสดงส่วนสวิทช์ฉุกเฉินที่จะควบคุมได้ 4 ตัว คือ S01 - S04 แสดงส่วนของการอลาม (ALARM) ของสวิทช์ฉุกเฉินได้ 4 ตัวคือ A01 - A04 โดยที่ NOT คือ ไม่ได้ต่อใช้งาน, ON คือ แสดงสถานะอุปกรณ์ที่นำมาต่อกำลังทำงานอยู่, OFF คือ แสดงสถานะอุปกรณ์ที่นำมาต่อยังไม่ได้ทำงาน ซึ่งในการทดลองนี้จะใช้ส่วนควบคุมปลายทางอยู่ 2 ตัว โดยทำการทดลองตั้งแต่แอดเดรสที่ 00 จนถึงเอดเดรสที่ 31 โดยได้ทำการทดลองร่วมกับเมนูส่วนต่าง ๆ ซึ่งได้ผลการทดลองที่ถูกต้องตามความต้องการทุกประการ

ทำการทดลองการตั้งเวลานาฬิกาจริง(Real Time) ของระบบ จากเมนูทำการทดลองโดยการกดตัวอักษร B ซึ่งจะได้ผลแสดงดังนี้

********** W E ŁC O M E **********

A - SHOW STATUS

B - SET TIME

C - SET TIME ON AND OFF FOR EACH ROOM

D - DIRECT CONTROL EQUIPMENT

E - CLEAR ALARM SENSOR SWITCH

F - CLEAR SCREEN LCD PANEL

G - SETUP TIMER SWITCH SENSOR(S)

H - DIRECT CONTROL SWITCH SENSOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SELECT> B

0 = SUN

1 = MON

2 = TUE

3 = WEN

4 = THU

5 = FRI

6 = SAT

BREACK PROGRAM BY KEY SPACE MANY

Current Day and Time is : WEN 23:58:56

Enter new Day and Time : 3 18:58:00

WEN 18:58:00

ผลการทดลอง

จากการทดลองเราสามารถที่จะทำการตั้งเวลานาฬิกาจริง ได้โดยที่ 0=วันอาทิตย์,
1=วันจันทร์,2=วันอังคาร,3=วันพุธ,4=วันพฤหัสบดี,5=วันศุกร์,6=วันเสาร์ และสามารถตั้งเวลาตามรูปแบบที่กำหนดขึ้นซึ่งผลที่ได้ถูกต้องตามที่ต้องการทุกประการ

ทำการทดลองการตั้งเวลาเปิดและปิด ของอุปกรณ์ที่จะทำการควบคุมในแต่ละห้อง จากเมนูทำการทดลองโดยการ กดตัวอักษร C ได้ผลตามที่ปรากฏอยู่ข้างล่างนี้

********** W E L C O M E **********

A - SHOW STATUS

B - SET TIME

C - SET TIME ON AND OFF FOR EACH ROOM

D - DIRECT CONTROL EQUIPMENT

E - CLEAR ALARM SENSOR SWITCH

F - CLEAR SCREEN LCD PANEL

G - SETUP TIMER SWITCH SENSOR(S)

H - DIRECT CONTROL SWITCH SENSOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SELECT> C

0 = SUN

1 = MON

2 = TUE

3 = WEN

4 = THU

5 = FRI

6 = SAT

BREACK PROGRAM BY KEY SPACE MANY

ENTER CODE DAY FOR SET TIME :WEN

Enter number room (00-31) : 30

POINT : 1 Current time ON 55:55 Current time OFF 00:00
Edit new time ON 10:00 Edit new time OFF 10:01

POINT : 2 Current time ON 55:55 Current time OFF 00:01
Edit new time ON 10:02 Edit new time OFF 10:03

POINT : 3 Current time ON 57:55 Current time OFF 00:02
Edit new time ON 10:04 Edit new time OFF 10:05

POINT : 4 Current time ON 55:55 Current time OFF 00:03
Edit new time ON 10:06 Edit new time OFF 10:07

POINT : 5 Current time ON 00:20 Current time OFF 00:04
Edit new time ON 10:08 Edit new time OFF 10:09

POINT : 6 Current time ON 55:55 Current time OFF 00:05
Edit new time ON 10:10 Edit new time OFF 10:11

POINT : 7 Current time ON 55:55 Current time OFF 00:06
Edit new time ON 10:12 Edit new time OFF 10:13

POINT : 8 Current time ON 00:05 Current time OFF 00:08
Edit new time ON 10:14 Edit new time OFF 10:15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลการทดลอง

จากการทดลองเราสามารถที่จะตั้งเวลาเปิดและปิดอุปกรณ์ที่จะทำการควบคุม ในแต่ละห้องได้ครั้งละ 1 อาทิตย์ โดยทำการกำหนดวันที่จะทดลองคือวันพุธ กำหนดห้องที่จะควบคุมคือห้องที่ 30 และทำการกำหนดเวลาการเปิดและปิดอุปกรณ์ที่จะทำการควบคุมในจุดที่ 1 ถึงจุดที่ 8 โดยกำหนดให้เวลาเปิดและปิดห่างกันจุดละ 1 นาที และทำการทดลองการตั้งเวลาให้ทำงานเปิดและปิดอุปกรณ์ที่ทำการทดลองในเวลา 1 อาทิตย์ โดยทำการเปิดและปิดอุปกรณ์ที่ทำการทดลองในแต่ละวัน ผลที่ได้สามารถที่จะทำการควบคุมการเปิดและปิดอุปกรณ์ที่ทำการทดลอง ได้ผลตามที่ต้องการทุกประการ

ทำการทดลองการควบคุมอุปกรณ์โดยตรง โดยไม่ต้องตั้งเวลา จากเมนูทำการทดลองโดยการกดตัวอักษร D ได้ผลตามที่ปรากฏอยู่ข้างล่างนี้

********** W E L C O M E **********

A - SHOW STATUS
B - SET TIME
C - SET TIME ON AND OFF FOR EACH ROOM
D - DIRECT CONTROL EQUIPMENT
E - CLEAR ALARM SENSOR SWITCH
F - CLEAR SCREEN LCD PANEL
G - SETUP TIMER SWITCH SENSOR(S)
H - DIRECT CONTROL SWITCH SENSOR

SELECT> D
SELECT> This is control OPEN and CLOSE to point
Enter number room (00-31) : 30
Enter number point (1 - 8): 1
PRESENT STATUS :OFF
Enter (ON = 1) OR (OFF = 0) :1

ผลการทดลอง

จากการทดลองทำการควบคุมอุปกรณ์โดยตรง โดยไม่ต้องตั้งเวลา โดยกำหนดห้องที่ 30 จุดที่ 1 ซึ่งตอนนี้โชว์สถานะปิดอยู่ถ้าต้องการเปิดก็เลือก 1 และถ้าต้องการปิดก็เลือก 0 ซึ่งผลที่ได้สามารถทำการเปิดและปิดของอุปกรณ์จุดที่ 1 ได้และทำการทดลอง จากจุดที่ 2 ถึงจุดที่ 8 ในลักษณะเดียวกันนี้ ผลปรากฏว่าสามารถที่จะทำการเปิดและปิดได้ทุกจุดและสถานะของอุปกรณ์ตาม

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จุดต่างๆ โซว์ถูกต้องทุกประการ และทำการทดลองกับห้องอื่น ๆ ผลที่ได้ตรงกับความต้องการทุกประการ

ทำการทดลองเคลียร์ (CLEAR) อลาม โดยทำให้เกิดการอลามขึ้น จากเมนูทำการทดลองโดยการกดตัวอักษร E ได้ผลตามที่ปรากฏอยู่ข้างล่างนี้

********** W E L C O M E **********

A - SHOW STATUS
B - SET TIME
C - SET TIME ON AND OFF FOR EACH ROOM
D - DIRECT CONTROL EQUIPMENT
E - CLEAR ALARM SENSOR SWITCH
F - CLEAR SCREEN LCD PANEL
G - SETUP TIMER SWITCH SENSOR(S)
H - DIRECT CONTROL SWITCH SENSOR

SELECT> E
PLEASE WAIT ... CLEAR ALARM
OK !

ผลการทดลอง

จากการทดลองเมื่อเกิดอลามขึ้นสามารถที่จะทำการเคลียร์อลามได้ตามที่ต้องการ

ทำการทดลอง เคลียร์หน้าจอ แอลซีดี (LCD) จากเมนูทำการทดลองโดยการการกดตัวอักษร F ได้ผลตามที่ปรากฏอยู่ข้างล่างนี้

********** W E L C O M E **********
A - SHOW STATUS
B - SET TIME
C - SET TIME ON AND OFF FOR EACH ROOM
D - DIRECT CONTROL EQUIPMENT
E - CLEAR ALARM SENSOR SWITCH
F - CLEAR SCREEN LCD PANEL
G - SETUP TIMER SWITCH SENSOR(S)
H - DIRECT CONTROL SWITCH SENSOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SELECT> F

ผลการทดลอง

สามารถที่จะทำการเคลียร์หน้าจอ แอลซีดี(LCD) ได้ตามความต้องการ

การทดลองการตั้งเวลาของสวิทช์ฉุกเฉิน จากเมนูทำการทดลองโดยการการกดตัวอักษร G ได้ผลตามที่ปรากฏอยู่ข้างล่างนี้

********** W E L C O M E **********

A - SHOW STATUS
B - SET TIME
C - SET TIME ON AND OFF FOR EACH ROOM
D - DIRECT CONTROL EQUIPMENT
E - CLEAR ALARM SENSOR SWITCH
F - CLEAR SCREEN LCD PANEL
G - SETUP TIMER SWITCH SENSOR(S)
H - DIRECT CONTROL SWITCH SENSOR

SELECT> G

SELECT>
0 = SUN
1 = MON
2 = TUE
3 = WEN
4 = THU
5 = FRI
6 = SAT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

BREACK PROGRAM BY KEY SPACE MANY

ENTER CODE DAY FOR SET TIME :WEN

Enter number room (00-31) : 30

POINT : 1 Current time ON 00:05 Current time OFF 00:07
Edit new time ON **10:00** Edit new time OFF **10:05**

POINT : 2 Current time ON 00:06 Current time OFF 00:08
Edit new time ON **10:00** Edit new time OFF **10:05**

POINT : 3 Current time ON 00:07 Current time OFF 00:09
Edit new time ON **10:00** Edit new time OFF **10:05**

POINT : 4 Current time ON 00:08 Current time OFF 00:10
Edit new time ON **10:00** Edit new time OFF **10:05**

ผลการทดลอง

จ ากการทดลองทำการทดลองการตั้งเวลาของห้องที่ 30 โดยทำการตั้งเวลาตั้งแต่จุดที่ 1 ถึงจุดที่ 4 ให้มีระยะเวลาจุดละ 5 นาที เวลาเดียวกัน ผลปรากฏว่าถ้าสวิทช์ฉุกเฉินจุดไหน ทำงานในช่วงเวลาที่กำหนดขึ้นจะทำให้เกิดอลามขึ้นแสดงที่จอแอลซีดี (LCD) โดยถ้าเราไม่ทำการเคลียร์อลามจะเกิดอลามโชว์ที่จอแอลซีดี (LCD) ตลอด แต่ถ้าทำการเคลียร์อลามไปแล้วส่วนของสวิทช์ฉุกเฉินจะทำการทำงานต่อไปตามที่ได้ตั้งเวลาไว้ และได้ทำการทดลองกับห้องอื่น ๆ ซึ่งผลที่ได้จากการทดลองนี้จะได้ตามที่ต้องการ

การทดลองการควบคุมการทำงานของสวิทช์ฉุกเฉินโดยตรง จากเมนูทำการทดลองโดยการการกดตัวอักษร H ได้ผลตามที่ปรากฏอยู่ข้างล่างนี้

********** W E L C O M E **********

A - SHOW STATUS

B - SET TIME

C - SET TIME ON AND OFF FOR EACH ROOM

D - DIRECT CONTROL EQUIPMENT

E - CLEAR ALARM SENSOR SWITCH

F - CLEAR SCREEN LCD PANEL

G - SETUP TIMER SWITCH SENSOR(S)

H - DIRECT CONTROL SWITCH SENSOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SELECT> H

SELECT> H This is control OPEN and CLOSE to point

Enter number room (00-31) : 30

Enter number point (1-4): 1

PRESENT STATUS :OFF

Enter (ON = 1) OR (OFF = 0) : 1

ผลการทดลอง

จากการทดลองควบคุมการทำงานของสวิทช์ฉุกเฉินโดยตรง โดยไม่ต้องตั้งเวลา โดยกำหนดห้องที่ 30 จุดที่ 1 ซึ่งตอนนี้โชว์สถานะไม่ทำงานอยู่ถ้าต้องการให้ทำงานก็เลือก 1 และถ้าต้องการไม่ให้ทำงานก็เลือก 0 ซึ่งผลที่ได้สามารถทำการควบคุมการทำงานของสวิทช์ฉุกเฉินโดยตรงได้ โดยทำการทดลองจุดที่ 1 จนถึงจุดที่ 4 ในลักษณะเดียวกันนี้ และทำการทดลองกับห้องอื่นๆ ผลปรากฏว่าสามารถที่จะควบคุมได้ถูกต้องทุกจุดตามที่ต้องการ

4.3สรุปผลการทดลอง

1. การทดลองต่อวงจรใช้งานจริงโดยเอาตัวส่งต่ออยู่ที่ต้นทางและตัวรับอยู่ที่ปลายทางและใช้สายยาวประมาณ 100 เมตร ตัวรับ 2 ตัวสามารถรับคำสั่งได้ถูกต้อง ตามความต้องการทุกอย่าง

2. จากการทดลองใช้ในการส่งจากต้นทางไปหาปลายทางอย่างเดียวใช้สายส่งระยะ 400 เมตร ก็ไม่ได้มีปัญหาแต่อย่างไรเลย

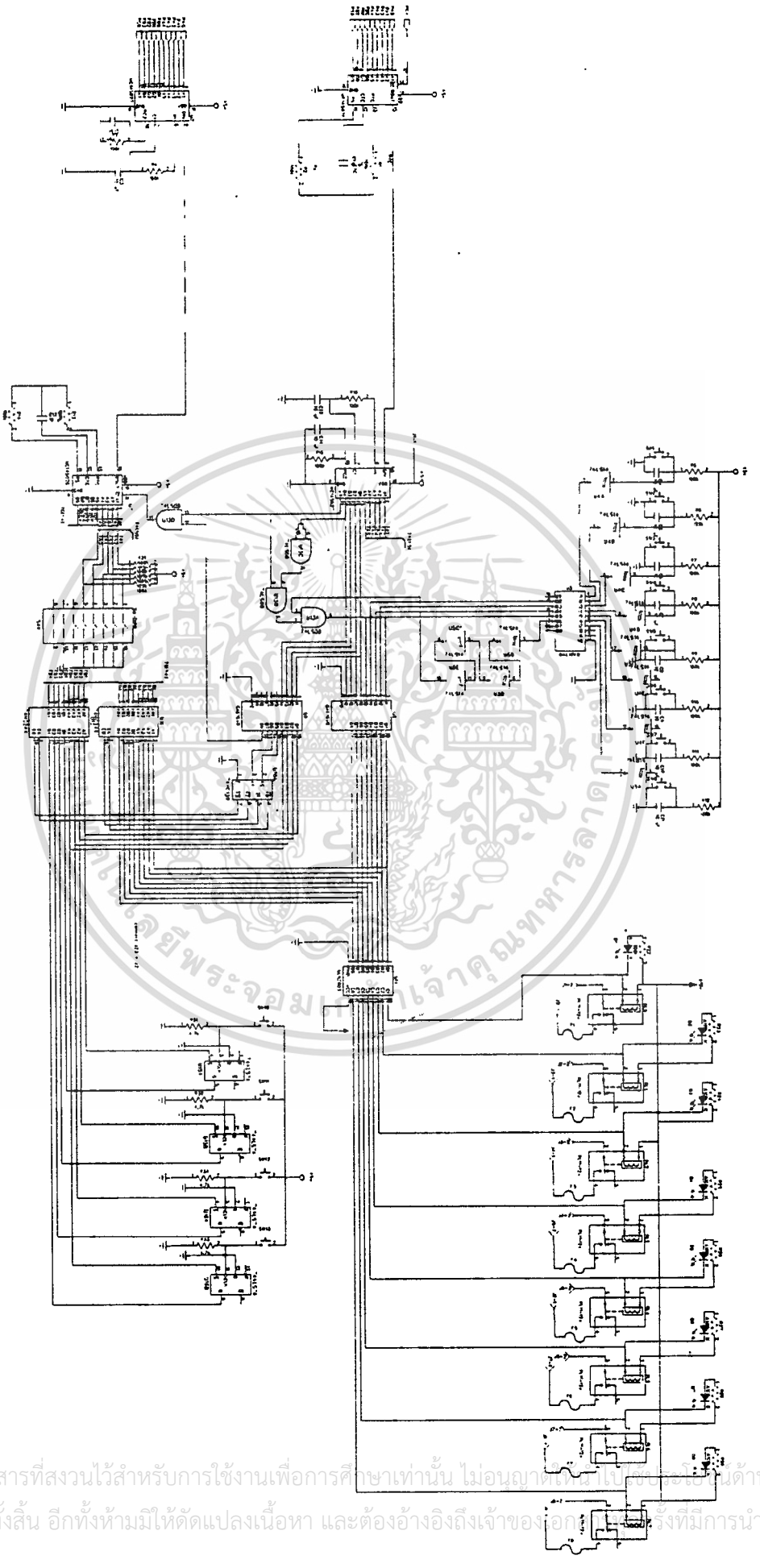
3. ถ้าเครื่องต้นทางมีการติดต่อกับคอมพิวเตอร์บ่อย ๆ จะทำให้เครื่องต้นทางเสียเวลาไปและอาจข้ามการปิดเปิดสวิทช์บางจุดไปได้

4. จากผลการทดลองทั้งหมดสามารถที่จะทำงานได้ตามวัตถุประสงค์ที่ตั้งไว้ทุกประการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

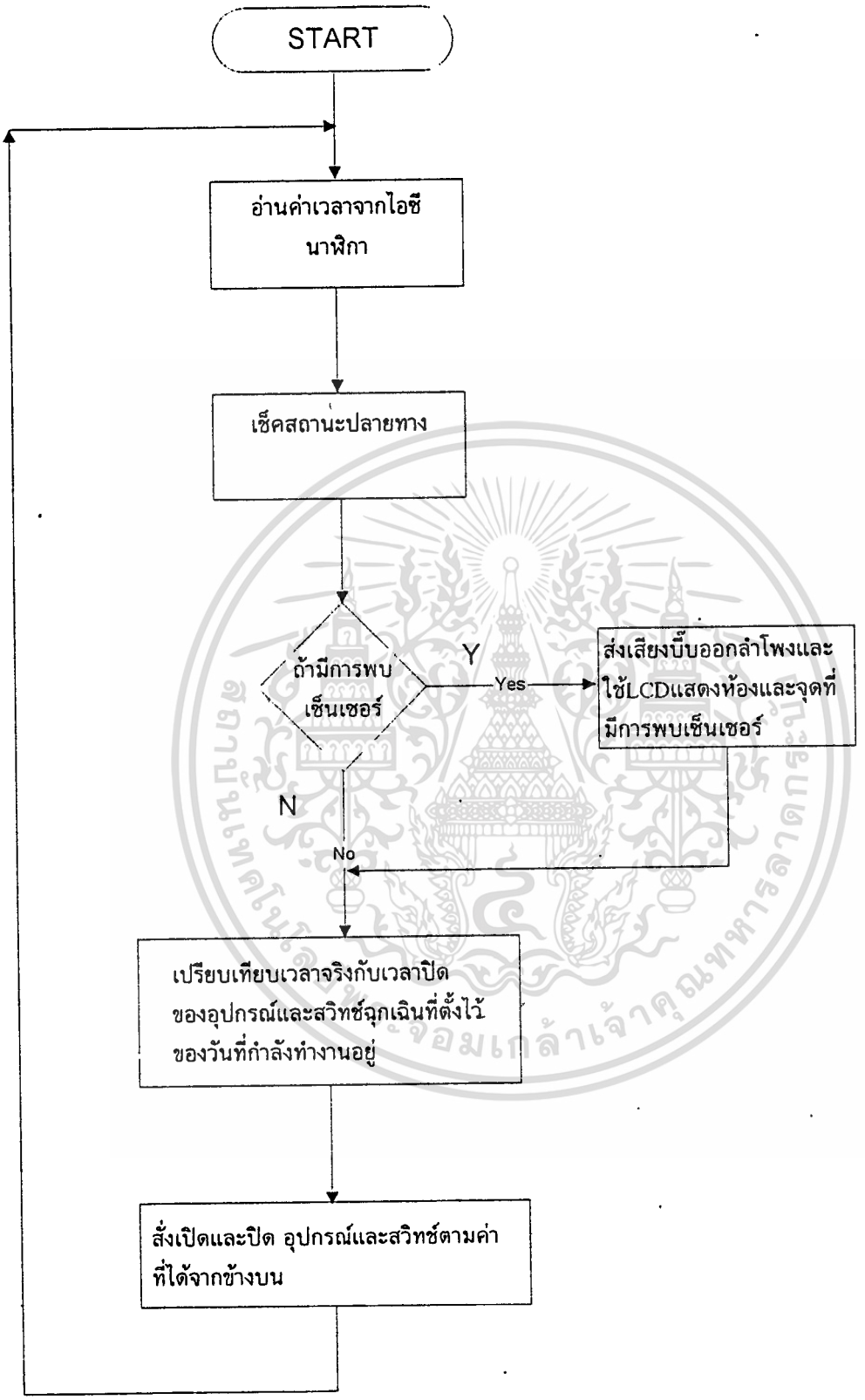
ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



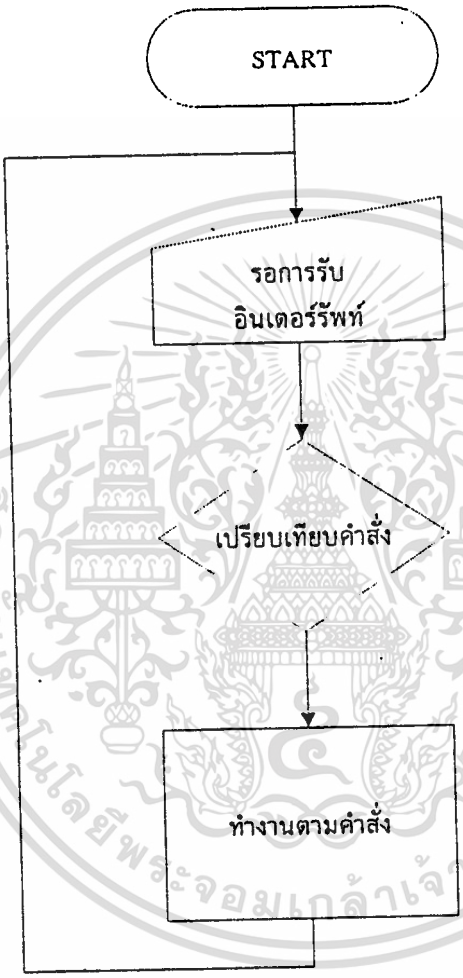
รูปที่ ง (ต่อ) วงจรแสดงการต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดง โฟลว์ชาร์ต (Flow chart) การทำงานโปรแกรมชนบอร์ด Z84c11

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปแสดง โฟลว์ชาร์ต ของโปรแกรมที่ติดต่อกับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของ โปรแกรม แกล # U28 ที่ใช้ในส่วนของวงจรควบคุมอุปกรณ์ปลายทาง

```
title    U28
pattern
revision A
date     05.10.94

chip u28 GAL16V8

; pin  1    2    3    4    5    6    7    8    9    10
     clk  b    c    a    kf   d1   d2   d3   d4   gnd

; pin 11   12   13   14   15   16   17   18   19   20
     /oe  o8   o7   o6   o5   o4   o3   o2   o1   vcc


equations

;
; assignment operator ":=" are registered equations.
;

o1     :=    /kf * /d4 * /d3 * /d2 * /d1 * /o1
           + /kf * d3 * o1
           + /kf * d2 * o1
           + /kf * d1 * o1
           + kf * /a * /b * /c * /o1
           + kf * a * o1
           + kf * b * o1
           + kf * c * o1

o2     :=    /kf * /d4 * /d3 * /d2 * d1 * /o2
           + /kf * d3 * o2
           + /kf * d2 * o2
           + /kf * /d1 * o2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          + kf * a * /b * /c * /o2
          + kf * /a * o2
          + kf * b * o2
          + kf * c * o2

o3    :=   /kf * /d4 * /d3 * d2 * /d1 * /o3
          + /kf * d3 * o3
          + /kf * /d2 * o3
          + /kf * d1 * o3
          + kf * /a * b * /c * /o3
          + kf * a * o3
          + kf * /b * o3
          + kf * c * o3

o4    :=   /kf * /d4 * /d3 * d2 * d1 * /o4
          + /kf * d3 * o4
          + /kf * /d2 * o4
          + /kf * /d1 * o4
          + kf * a * b * /c * /o4
          + kf * /a * o4
          + kf * /b * o4
          + kf * c * o4

o5    :=   /kf * /d4 * d3 * /d2 * /d1 * /o5
          + /kf * /d3 * o5
          + /kf * d2 * o5
          + /kf * d1 * o5
          + kf * /a * /b * c * /o5
          + kf * a * o5
          + kf * b * o5
          + kf * /c * o5

o6    :=   /kf * /d4 * d3 * /d2 * d1 * /o6
          + /kf * /d3 * o6
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          + /kf * d2 * o6
          + /kf * /d1 * o6
          + kf * a * /b * c * /o6
          + kf * /a * o6
          + kf * b * o6
          + kf * /c * o6

o7    :=   /kf * /d4 * d3 * d2 * /d1 * /o7
          + /kf * /d3 * o7
          + /kf * /d2 * o7
          + /kf * d1 * o7
          + kf * /a * b * c * /o7
          + kf * a * o7
          + kf * /b * o7
          + kf * /c * o7

o8    :=   /kf * /d4 * d3 * d2 * d1 * /o8
          + /kf * /d3 * o8
          + /kf * /d2 * o8
          + /kf * /d1 * o8
          + kf * a * b * c * /o8
          + kf * /a * o8
          + kf * /b * o8
          + kf * /c * o8
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของ โปรแกรม แกล # U23 ที่ใช้ในส่วนของวงจรควบคุมอุปกรณ์ปลายทาง

```
title    U23
revision A
date     05.10.94
chip u23 GAL16V8

; pin  1    2    3    4    5    6    7    8    9    10
     clk  d2   d3   d4   d1   nc   nc   nc   nc   gnd

; pin 11   12   13   14   15   16   17   18   19   20
     /oe  o8   o5   o6   o7   o4   o3   o2   o1   vcc


equations

;
; assignment operator ":=" are registered equations.
;

o1      :=    d4 * /d3 * /d2 * /d1 * /o1
            + d4 * d3 * o1
            + d4 * d2 * o1
            + d4 * d1 * o1
            + /d4 * o1

o2      :=    d4 * /d3 * /d2 * d1 * /o2
            + d4 * d3 * o2
            + d4 * d2 * o2
            + d4 * /d1 * o2
            + /d4 * o2

o3      :=    d4 * /d3 * d2 * /d1 * /o3
            + d4 * d3 * o3
            + d4 * /d2 * o3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        + d4 * d1 * o3
        + /d4 * o3

o4   :=   d4 * /d3 * d2 * d1 * /o4
        + d4 * d3 * o4
        + d4 * /d2 * o4
        + d4 * /d1 * o4
        + /d4 * o4

o5   :=   d4 * d3 * /d2 * /d1 * /o7
        + d4 * d3 * /d2 * d1 * o5
        + d4 * d3 * d2 * d1 * o5

o6   :=   d4 * d3 * /d2 * d1 * /o7
        + d4 * d3 * /d2 * /d1 * o6
        + d4 * d3 * d2 * d1 * o6

o7   =    d4 * d3 * d2 * /d1

o8   =    d4 * d3 * d2 * d1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของ โปรแกรม แกล # U13 ที่ใช้ในส่วนของวงจรควบคุมอุปกรณ์ปลายทาง (ส่วนของวงจรคีย์สวิทช์)

```
title    U13
revision A
date     05.10.94
chip u13 GAL16V8

; pin  1    2    3    4    5    6    7    8    9    10
     clk  s0   s1   s2   s3   s4   s5   s6   s7   gnd

; pin 11   12   13   14   15   16   17   18   19   20
     /oe  o7   o6   o5   o4   o3   o2   o1   o0   vcc


equations

;
; assignment operator "=" are combination equations.
;

o1     =   /s7 * /s6 * /s5 * /s4 * /s3 * /s2 * s1 * /s0
         + /s7 * /s6 * /s5 * /s4 * s3 * /s2 * /s1 * /s0
         + /s7 * /s6 * s5 * /s4 * /s3 * /s2 * /s1 * /s0
         +s7 * /s6 * /s5 * /s4 * /s3 * /s2 * /s1 * /s0

o2     =   /s7 * /s6 * /s5 * /s4 * /s3 * s2 * /s1 * /s0
         + /s7 * /s6 * /s5 * /s4 * s3 * /s2 * /s1 * /s0
         + /s7 * s6 * /s5 * /s4 * /s3 * /s2 * /s1 * /s0
         +s7 * /s6 * /s5 * /s4 * /s3 * /s2 * /s1 * /s0

o3     =   /s7 * /s6 * /s5 * s4 * /s3 * /s2 * /s1 * /s0
         + /s7 * /s6 * s5 * /s4 * /s3 * /s2 * /s1 * /s0
         + /s7 * s6 * /s5 * /s4 * /s3 * /s2 * /s1 * /s0
         +s7 * /s6 * /s5 * /s4 * /s3 * /s2 * /s1 * /s0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
o4      =   /o5 ;out put active '1'

o5      =    /s7 * /s6 * /s5 * /s4 * /s3 * /s2 * /s1 * /s0 ;out put
active '0'
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของ โปรแกรม ที่ใช้ควบคุมบอร์ด Z84C11-PLUS

```
        ORG   0000H
ORG_00      XOR   A
ORG_01      DEC   A
        NOP
        JR    NZ, ORG_01
        NOP
        IM    1
        EI
        JP    MAIN_PROGRAM
        NOP
        JP    ORG_00

        ORG   0038H
ORG_38H     CALL  RX_INT
        JP    MAIN_PROGRAM
        JP    ORG_00
        NOP
        NOP
        NOP

        ORG   0066H
ORG_66H     CALL  RX_INT_N
        JP    MAIN_PROGRAM
        JP    ORG_00
        NOP
        NOP
        ORG   0100H

;*****************************************************************
MAIN_PROGRAM:

SUN_ON      EQU   0A000H
SUN_OFF     EQU   0A200H

CTC0        EQU   10H
CTC1        EQU   11H
CTC2        EQU   12H
CTC3        EQU   13H
;
PA          EQU   50H
PB          EQU   51H
PC          EQU   52H
PD          EQU   30H
PE          EQU   40H
;
PAC         EQU   54H
PBC         EQU   55H
PCC         EQU   56H
PDC         EQU   34H
PEC         EQU   44H
;
WDTMAS      EQU   0F0H
WDTCON      EQU   0F1H
MISREG      EQU   0EEH
MISCON      EQU   0EFH
;
; I/O LCD PORT
; I/O LCD PORT
PDATA_LCD   EQU   80H
PSING_LCD   EQU   82H
PREAD_LCD   EQU   84H
;
B12         EQU   00H   ;POWER ON RUN 10M NOT SET REG
B24         EQU   81H
B48         EQU   3DH
B96         EQU   1EH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
B192      EQU   0DH

BUT       EQU   B96

;*************************************************************************
;INITIAL RTC
        XOR   A
        OUT   (0ADH),A    ;CLEAR
        LD    A,01H       ;set 24h or 12h
        OUT   (0AFH),A
        LD    A,04H       ;set 24h
        OUT   (0AFH),A

;*************************************************************************
INITIAL_PORT  LD    A, 0FFH
        OUT   (054H), A
        LD    A, 0FH
        OUT   (055H), A
        LD    A, 0F0H
        OUT   (056H), A
        LD    A, 0A2H
        OUT   (044H),A
        CALL  INITLCD

START_PROGRAM  LD    HL, 8E00H
        CALL  READ_RTC_HEX
        CALL  CHK_STATUS
        CALL  SOUND_SENSOR
        CALL  CHK_TIME_ON
        CALL  CHK_TIME_OFF
        CALL  OUT_PORT_CL_OP
        CALL  OUTPORT_REAL
        CALL  SOUND_SENSOR
        CALL  CHK_SENS_ON
        CALL  CHK_SENS_OFF
        CALL  OUT_SENS_CL_OP
        CALL  OUT_SEN_REAL
        CALL  SOUND_SENSOR
        JP    START_PROGRAM
        RET
;*************************************************************************
MENU_MAIN_PCP  PUSH  DE
CONTINUE_PCP   CALL  PUTCH
        LD    D, 8
        CALL  PUTCH
        POP   DE

CONTINUE     LD    A, D
        CP    'A'
        JP    Z, MENU_1
        CP    'B'
        JP    Z, MENU_2
        CP    'C'
        JP    Z, MENU_3
        CP    'D'
        JP    Z, MENU_4
        CP    'E'
        JP    Z, MENU_5
        CP    'F'
        JP    Z, MENU_6
        CP    'G'
        JP    Z, MENU_7
        CP    'H'
        JP    Z, MENU_8

        CP    20H
        JP    Z, SHOW_MENU_CRT
        CP    13
        JP    Z, SHOW_MENU_CRT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RET
;*****************************************************************

MENU_1      LD    IX, MASS_MENU1
        CALL  WRLINE
        CALL  SHOW_STATUS
        CALL  CLRSCR_LCD
        RET
MENU_2      LD    IX, MASS_MENU2
        CALL  WRLINE
        CALL  SET_TIME
        CALL  CLRSCR_LCD
        RET
MENU_3      LD    IX, MASS_MENU3
        CALL  WRLINE
        CALL  TIME_OPEN_CLOSE
        CALL  CLRSCR_LCD
        RET
MENU_4      LD    IX, MASS_MENU4
        CALL  WRLINE
        CALL  ON_LINE
        CALL  CLRSCR_LCD
        RET
MENU_5      LD    IX, MASS_MENU5
        CALL  WRLINE
        CALL  RES_SENS
        CALL  CLRSCR_LCD
        RET
MENU_6      CALL  CLRSCR_LCD
        RET
MENU_7      LD    IX, MASS_MENU7
        CALL  WRLINE
        CALL  SETUP_TIME_SENSOR
        CALL  CLRSCR_LCD
        RET
MENU_8      LD    IX, MASS_MENU8
        CALL  WRLINE
        CALL  DIRECT_SENSOR_CONTROL
        CALL  CLRSCR_LCD
        RET

;*****************************************************************
CHK_STATUS:
        LD    HL, 8000H     ;CHECK STATUS OF EQUIPMENT
        LD    B, 32
        LD    C, 0

CHK_STATUS_1  LD    A, C          ;IN BIT D0-D3
        OUT   (050H), A     ;OUT ADDRESS 5 BIT
        LD    A, 0EH
        OUT   (051H), A
        LD    A, 0EFH
        OUT   (052H), A
        LD    A, 0FFH
        OUT   (052H), A
        CALL  DELAY_SENS
        CALL  CHK_STATUS_IO  ;OUT PORT 51H WITH 4 BIT AND IN PORT D
                    ;GETCH STATUS BIT 0-3
        LD    A, C          ;IN BIT D4-D7
        OUT   (050H), A
        LD    A, 0CH        ;PROVICE 01 IS WAIT IN D4-D7
        OUT   (051H), A
        LD    A, 0EFH
        OUT   (052H), A
        LD    A, 0FFH
        OUT   (052H), A
        CALL  DELAY_SENS
        CALL  CHK_STATUS_IO
        CALL  DELAY_SENS
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CHK_STATUS_SEN:
        LD    A, C          ;IN BIT D0-D3
        OUT   (050H), A     ;OUT ADDRESS 5 BIT
        LD    A, 0EH        ;CLEAR REGITER AT SEND STATUS RETURE P
        OUT   (051H),A
        LD    A, 0EFH
        OUT   (052H), A
        LD    A, 0FFH
        OUT   (052H), A
        CALL  DELAY_SENS

        LD    A, C          ;CHECK STATUS WITH SWITCH SENSOR ON_OFF
        OUT   (050H), A
        LD    A, 0DH
        OUT   (051H), A
        LD    A, 0EFH
        OUT   (052H), A
        LD    A, 0FFH
        OUT   (052H), A
        CALL  DELAY_SENS
        CALL  CHK_STATUS_IO  ;OUT PORT 51H WITH 4 BIT AND IN PORT D

        LD    A, C          ;CHECK ALARM SENSOR
        OUT   (050H), A
        LD    A, 0CH
        OUT   (051H), A
        LD    A, 0EFH
        OUT   (052H), A
        LD    A, 0FFH
        OUT   (052H), A
        CALL  DELAY_SENS
        CALL  CHK_STATUS_IO_BACK  ;OUT PORT 51H WITH 4 BIT AND IN PORT D
        CALL  DELAY_SENS


        INC   C
        DJNZ  CHK_STATUS_1
        CALL  CHK_STATUS_BMP

        RET
;*******************************************************************************

CHK_STATUS_IO:

        LD    A, C          ;TRIG TERMINAL IS SEND RETURE
        OUT   (50H), A
        LD    A, 0FH
        OUT   (051H), A
        LD    A, 0EFH
        OUT   (052H), A
        LD    A, 0FFH
        OUT   (052H), A

        PUSH  AF
        PUSH  BC
        LD    BC, 0FFFH
DELAY_IN    DEC   BC
        IN    A, (052H)
        BIT   0, A
        JR    NZ, IN_BIT_STATUS   ;HARDWARE ACTIVE HIGH
        LD    A, B
        OR    C
        JR    NZ, DELAY_IN
        JR    HIGH_IMP

IN_BIT_STATUS  POP   BC
        POP   AF
        IN    A, (051H)
IN_BIT_4    BIT   4, A
        JR    NZ, IN_BIT_ON4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IN_BIT_OFF4   LD    (HL), 30H
        JR    IN_BIT_5
IN_BIT_ON4    LD    (HL), 31H

IN_BIT_5      INC   HL
        BIT   5, A
        JR    NZ, IN_BIT_ON5
IN_BIT_OFF5   LD    (HL), 30H
        JR    IN_BIT_6
IN_BIT_ON5    LD    (HL), 31H

IN_BIT_6      INC   HL
        BIT   6, A
        JR    NZ, IN_BIT_ON6
IN_BIT_OFF6   LD    (HL), 30H
        JR    IN_BIT_7
IN_BIT_ON6    LD    (HL), 31H

IN_BIT_7      INC   HL
        BIT   7, A
        JR    NZ, IN_BIT_ON7
IN_BIT_OFF7   LD    (HL), 30H
        JR    IN_BIT_END
IN_BIT_ON7    LD    (HL), 31H
        JR    IN_BIT_END

HIGH_IMP      POP   BC
        POP   AF
        LD    (HL), 32H
        INC   HL
        LD    (HL), 32H
        INC   HL
        LD    (HL), 32H
        INC   HL
        LD    (HL), 32H
IN_BIT_END    INC   HL
        RET
;*****************************************************************

CHK_STATUS_IO_BACK:

        LD    A, C          ;TRIG TERMINAL IS SEND RETURN
        OUT   (50H), A
        LD    A, 0FH
        OUT   (051H), A
        LD    A, 0EFH
        OUT   (052H), A
        LD    A, 0FFH
        OUT   (052H), A

        PUSH  AF
        PUSH  BC
        LD    BC, 0FFFH
DELAY_IN_BACK  DEC   BC
        IN    A, (052H)
        BIT   0, A
        JR    NZ, IN_BIT_STATUS_BACK    ;HARDWARE ACTIVE HIGH
        LD    A, B
        OR    C
        JR    NZ, DELAY_IN_BACK
        JR    HIGH_IMP_BACK

IN_BIT_STATUS_BACK
          POP   BC
          POP   AF
          IN    A, (051H)
IN_BIT_4_BACK      BIT   4, A
          JR    Z, IN_BIT_ON4_BACK
IN_BIT_OFF4_BACK   LD    (HL), 30H
          JR    IN_BIT_5_BACK
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
IN_BIT_ON4_BACK    LD    (HL), 31H

IN_BIT_5_BACK      INC   HL
                   BIT   5, A
                   JR    Z, IN_BIT_ON5_BACK
IN_BIT_OFF5_BACK   LD    (HL), 30H
                   JR    IN_BIT_6_BACK
IN_BIT_ON5_BACK    LD    (HL), 31H

IN_BIT_6_BACK      INC   HL
                   BIT   6, A
                   JR    Z, IN_BIT_ON6_BACK
IN_BIT_OFF6_BACK   LD    (HL), 30H
                   JR    IN_BIT_7_BACK
IN_BIT_ON6_BACK    LD    (HL), 31H

IN_BIT_7_BACK      INC   HL
                   BIT   7, A
                   JR    Z, IN_BIT_ON7_BACK
IN_BIT_OFF7_BACK   LD    (HL), 30H
                   JR    IN_BIT_END_BACK
IN_BIT_ON7_BACK    LD    (HL), 31H
                   JR    IN_BIT_END_BACK

HIGH_IMP_BACK      POP   BC
                   POP   AF
                   LD    (HL), 32H
                   INC   HL
                   LD    (HL), 32H
                   INC   HL
                   LD    (HL), 32H
                   INC   HL
                   LD    (HL), 32H
IN_BIT_END_BACK    INC   HL
                   RET

;*****************************************************************
CHK_STATUS_BMP:
        LD    HL, 9000H-1
        LD    IX, 8000H-1   ;OUT PORT ON FOR BMP
        LD    C, 32
CHK_STATUS_BMP1 LD    B, 8
        INC   IX
CHK_STATUS_BMP0 INC   HL
        LD    A, (HL)
        SUB   31H
        CALL  BIT_MAP_STA
        DJNZ  CHK_STATUS_BMP0

        PUSH  AF
        PUSH  DE
        XOR   A
        LD    DE, 08
        ADC   HL, DE
        POP   DE
        POP   AF

        DEC   C
        JR    NZ,CHK_STATUS_BMP1

        LD    HL, 9008H-1
        LD    IX, 8400H-1          ;STATUS BMP SENSOR
        LD    C, 32

CHK_STATUS_BMP2 LD    B, 8
        INC   IX
CHK_STATUS_BMP3 INC   HL
        LD    A, (HL)
        SUB   31H
        CALL  BIT_MAP_STA
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DJNZ   CHK_STATUS_BMP3

        PUSH   AF
        PUSH   DE
        XOR    A
        LD     DE, 08
        ADC    HL, DE
        POP    DE
        POP    AF

        DEC    C
        JR     NZ,CHK_STATUS_BMP2

        RET
;*************************************************************************
BIT_MAP_STA   JR     Z, BIT_MAP_STA1
        JR     NZ, BIT_MAP_STA2
        RET

BIT_MAP_STA1  PUSH   AF
        SET    0,(IX)
        RRC    (IX)
        POP    AF
        RET
BIT_MAP_STA2  PUSH   AF
        RES    0,(IX)
        RRC    (IX)
        POP    AF
        RET

DELAY_SENS    PUSH   AF
        PUSH   BC
        LD     BC, 0FFFH
DELAY_SENS_1  DEC    BC
        LD     A, B
        OR     C
        JR     NZ, DELAY_SENS_1
        POP    BC
        POP    AF
        RET

;*************************************************************************
CHK_TIME_ON:
        CALL   TIME_FULL       ;'DE' IS STROE VALUE HOUR AND MINUTE
        LD     A, (9E00H)      ;CHKDAY
        SLA    A
        SLA    A
CHK_TIME_ON0  LD     HL, SUN_ON
        ADD    A, H
        LD     H, A
        PUSH   HL
        POP    IX
        LD     IY, 8040H      ;STORE START TIMMER ON PORM BMP
        LD     C, 32
CHK_TIME_ON1  LD     B, 8
CHK_TIME_ON2  LD     H, (IX)
        LD     L, (IX+1)
        SBC    HL, DE
        CALL   BITMAP_TIME
        INC    IX
        INC    IX
        DJNZ   CHK_TIME_ON2
        INC    IY
        DEC    C
        JR     NZ, CHK_TIME_ON1
        RET
;*************************************************************************
CHK_TIME_OFF:
        CALL   TIME_FULL       ;'DE' IS STROE VALUE HOUR AND MINUTE
        LD     A, (9E00H)      ;CHKDAY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        SLA   A
        SLA - A
CHK_TIME_OFF0  LD    HL, SUN_OFF
        ADD   A, H
        LD    H, A
        PUSH  HL
        POP   IX
        LD    IY, 8100H     ;STORE START TIMMER OFF FORM BMP
        LD    C, 32
CHK_TIME_OFF1  LD    B, 8
CHK_TIME_OFF2  LD    H, (IX)
        LD    L, (IX+1)
        SBC   HL, DE
        CALL  BITMAP_TIME
        INC   IX
        INC   IX
        DJNZ  CHK_TIME_OFF2
        INC   IY
        DEC   C
        JR    NZ, CHK_TIME_OFF1
        RET
;*************************************************************************
BITMAP_TIME   LD    A,H
        OR    L
        CALL  Z,BMP_TIME1   ;CALL BMP1 IS Z=1
        CALL  NZ,BMP_TIME2  ;CALL BMP2 IS Z=0
        RET
BMP_TIME1     PUSH  AF
        SET   0,(IY)
        RRC   (IY)
        POP   AF
        RET
BMP_TIME2     PUSH  AF
        RES   0,(IY)
        RRC   (IY)
        POP   AF
        RET
;*************************************************************************
;SEND VALUE BY '(HL)'
READ_RTC_HEX  IN    A, (0ADH)     ;GET CONTROL REG.
        OR    1             ;SET HOLD BIT
        OUT   (0ADH), A     ;WRITE TO RTC
RTC_BUSY_HEX  IN    A, (0ADH)     ;IS BUSY BIT EQUAL TO ZERO?
        BIT   1, A          ;
        JR    NZ, RTC_BUSY_HEX; NO, WAIT
        IN    A, (0ACH)         ; YES, READ DATA
        AND   0FH           ;MASK BIT D4-D7
        LD    (HL), A
        INC   HL
        PUSH  BC
        LD    C, 0A5H
        LD    B, 6
READ_LOOP_HEX  IN    A, (C)
        AND   0FH
        LD    (HL), A
        INC   HL
        DEC   C
        DJNZ  READ_LOOP_HEX
        POP   BC
        PUSH  AF
        IN    A, (0ADH)     ;GET CONTROL REG.
        AND   0FEH          ;RESET HOLD BIT
        OUT   (0ADH), A     ;WRITE TO RTC
        POP   AF
        RET
;*************************************************************************
;SEND TIME BY 'DE' D-TIME HOUR E-TIME MINUTE
TIME_FULL     LD    A, (9E01H)
        CALL  SLA_A
        LD    D, A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LD    A, (8E02H)
        AND   0FH
        OR    D
        LD    D, A
        LD    A, (8E03H)
        CALL  SLA_A
        LD    E, A
        LD    A, (8E04H)
        AND   0FH
        OR    E
        LD    E, A
        RET

;*************************************************************************
SRL_A   SRL   A
        SRL   A
        SRL   A
        SRL   A
        RET
SLA_A   SLA   A
        SLA   A
        SLA   A
        SLA   A
        RET
;*************************************************************************
OUT_PORT_CL_OP:
        LD    IX, 8000H     ;BMP STROGE STATUS
        LD    IY, 8040H     ;BMP OUTPORT ON
        LD    HL, 8080H     ;VALUE OUTPORT ON
        LD    B, 32
VALUE_ON    LD    A, (IX)       ;[IX AND IY] XOR [IY]
        AND   (IY)
        XOR   (IY)
        LD    (HL), A
        INC   IX
        INC   IY
        INC   HL
        DJNZ  VALUE_ON
        LD    IX, 8000H
        LD    IY, 8100H     ;BMP OUTPORT OFF
        LD    HL, 8140H     ;VALUE OUTPORT OFF
        LD    B, 32
VALUE_OFF   LD    A, (IX)
        AND   (IY)
        LD    (HL), A
        INC   IX
        INC   IY
        INC   HL
        DJNZ  VALUE_OFF
        LD    IX, 8080H
        LD    IY, 8140H
        LD    HL, 8200H
        LD    B, 32
TOTAL_OUTPORT  LD    A, (IX)
        OR    (IY)
        LD    (HL), A
        INC   IX
        INC   IY
        INC   HL
        DJNZ  TOTAL_OUTPORT
        RET
;*************************************************************************
OUTPORT_REAL  LD    IY, 8200H
        LD    L, 0          ;ADDRESS FOR OUT PORT
        LD    B, 32
OUT3_REAL     LD    H, 0          ;DATA FOR OUT PORT
OUT2_REAL     LD    E, 8
OUT1_REAL     RRC   (IY)
        JR    NC, OUT0_REAL
        LD    C, 50H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        OUT   (C), L   ,  ;OUT PORT WITH ADDRESS
        LD    C, 51H
        OUT   (C), H      ;OUT PORT WITH DATA
        LD    A, 0FFH     ;TRIG TO SEND
        LD    C, 52H
        OUT   (C), A
        LD    A, 0FFH
        OUT   (052H), A
        CALL  DELAY_SENS

OUT0_REAL   INC   H
        DEC   E
        JR    NZ, OUT1_REAL
        INC   IY
        INC   L
        DJNZ  OUT3_REAL
        RET
;*************************************************************************
;TRANSFER VALUE BY '(HL) FORM ASCII
PUTS      LD    A, (HL)
        OR    A
        JR    Z, DONE_PUTS
        LD    D, A
        CALL  PUTCH
        INC   HL
        JR    PUTS
DONE_PUTS   RET

PUTCH     PUSH  AF
        PUSH  BC
        LD    A, 80H      ;SEND START BIT
        OUT   (40H), A
        CALL  DELAY_BAUDRATE    ;CALL 37F3
        LD    B, 08H
SEND_CH   XOR   A
        RRC   D
        RRA
        RRA
        RRA
        SET   7, A
        OUT   (40H), A
        CALL  DELAY_BAUDRATE
        DJNZ  SEND_CH
        LD    A, 0A0H     ;SEND STOP BIT
        OUT   (40H), A
        CALL  DELAY_BAUDRATE
        POP   BC
        POP   AF
        RET
;*************************************************************************
KBHIT     IN    A, (40H)
        BIT   6, A
        JR    NZ,KBHIT
        RET
;*************************************************************************
GETCH     IN    A, (40H)
        BIT   6, A
        JR    NZ, GETCH
        NOP
        CALL  DELAY_BAUDRATE
        PUSH  BC
        LD    D, 00H
        LD    B, 08H
RX_DATA   IN    A, (40H)
        RLA
        RLA
        RR    D
        CALL  DELAY_BAUDRATE
        DJNZ  RX_DATA
        POP   BC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        RET

DELAY_BAUDRATE LD    A, 01EH
        SUB   02H
DELAY_BAUD_R  DEC   A
        JR    NZ, DELAY_BAUD_R
        RET

;*************************************************************************

GETS      IN    A, (40H)          ;STORAGE BY (HL)
        BIT   6, A
        JR    NZ, GETCH
        NOP
        CALL  DELAY_BAUDRATE
        PUSH  BC
        LD    D, 00H
        LD    B, 08H
RX_DATAS    IN    A, (40H)
        RLA
        RLA
        RR    D
        CALL  DELAY_BAUDRATE
        DJNZ  RX_DATA
        POP   BC
        LD    (HL), D
        LD    A, D
        CP    10              ;END BYTE IS ENTER '10'
        JR    NZ,GETS
        RET
;*************************************************************************
SEND_HEX_2DIG:
        LD    A, (HL)
        CALL  SRL_A
        ADD   A, 30H
        LD    D, A
        CALL  PUTCH
        LD    A, (HL)
        AND   0FH
        ADD   A, 30H
        LD    D, A
        CALL  PUTCH
        RET
;*************************************************************************
ASCII_TO_HEX:
        CALL  GETCH             ;SEND VALUE BY 'A'
        CALL  PUTCH
        LD    A, D
        CP    '0'               ;LOWER THAN '0' ?
        RET   C                 ;YES, ERROR RETURN
        CP    '9'+1             ;0-9 ?
        JR    C,ASCII_TO_HEX0
        CP    'A'
        RET   C
        CP    'F'+1
        CCF
        RET   C
        ADD   A, 09H
ASCII_TO_HEX0  AND   0FH
        RET
;*************************************************************************
SOUND_SENSOR:
        PUSH  IY
        PUSH  DE
        PUSH  BC
        LD    IY,8000H-16
        LD    DE,16
        LD    B,32
        LD    C,0
SOUND_SENSOR1:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LD    C,0
        ADD   IY,DE
        LD    A,(IY+0)
        CP    31H
        CALL  Z,MAKE_ALARM
        INC   C
        LD    A,(IY+1)
        CP    31H
        CALL  Z,MAKE_ALARM
        INC   C
        LD    A,(IY+2)
        CP    31H
        CALL  Z,MAKE_ALARM
        INC   C
        LD    A,(IY+3)
        CP    31H
        CALL  Z,MAKE_ALARM
        DJNZ  SOUND_SENSOR1
        POP   BC
        POP   DE
        POP   IY
        RET
;
MAKE_ALARM:
;
        CALL  START_LCD
        PUSH  AF
        PUSH  BC
        LD    A, 3FH
        OUT   (44H), A
        LD    BC, 0FFFH
MAKE_SOUND1:
        IN    A,(40H)
        SET   1,A
        OUT   (40H), A
        CALL  SOUND_DELAY
        IN    A,(40H)
        RES   1,A
        OUT   (40H), A
        CALL  SOUND_DELAY
        DEC   BC
        LD    A, B
        OR    C
        JR    NZ, MAKE_SOUND1
        POP   BC
        POP   AF
        RET

SOUND_DELAY   PUSH  AF
        PUSH  BC
        LD    BC, 02FH
SOUND_DELAY0  DEC   BC
        LD    A,B
        OR    C
        JR    NZ, SOUND_DELAY0
        POP   BC
        POP   AF
        RET
;*************************************************************************


;*******************************
;*  TEST LCD                   *
;*  16 CHARACTERS * 1 LINE     *
;*******************************
;

START_LCD:
WR_SENSOR     PUSH  AF
        PUSH  BC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        PUSH   IX
        PUSH   HL
        LD     A, 00H              ;SET CURSOR LINE '1'
        CALL   GOTO
        LD     IX, ACTIVE_SENSOR
        CALL   WRLINE
        LD     A,B
        CPL    A
        INC    A
        ADD    A,32
        LD     HL,BCD_TABLE
        ADD    A,A
        ADD    A,L
        LD     L,A
        JR     NC,$+3
        INC    H
        LD     A,(HL)
        CALL   WRBYTE
        INC    HL
        LD     A,(HL)
        CALL   WRBYTE
        LD     IX, LCD_SENSOR_POINT
        CALL   WRLINE
        LD     A,C
        ADD    A,'1'
        CALL   WRBYTE
        POP    HL
        POP    IX
        POP    BC
        POP    AF
        RET

BCD_TABLE     DB     '00','01','02','03','04','05','06','07','08','09'
        DB     '10','11','12','13','14','15','16','17','18','19'
        DB     '20','21','22','23','24','25','26','27','28','29'
        DB     '30','31'


ACTIVE_SENSOR         DB     'SWITCH SENSOR ACTIVE ROOM :', 0
LCD_SENSOR_POINT      DB     ' POINT : ', 0

GOTO        SET    7, A             ;SET CURSOR LEFT MOST
        OUT    (PDATA_LCD), A
        CALL   READ_LCD
        RET
;
READ_LCD      IN     A, (PREAD_LCD)
        BIT    7, A
        JR     NZ, READ_LCD
        RET

WRLINE        PUSH   IX               ;TRANFER VALUE BY "(IX)"
TEST_LCD      LD     A, (IX)
        CP     0
        JR     Z, TEST_LCD_END
        CALL   WRBYTE
        INC    IX
        JR     TEST_LCD
TEST_LCD_END  POP    IX
        RET

;***** WRITE BYTE SUB ****
WRBYTE        OUT    (PSINO_LCD), A       ;TRANFER VALUE BY REGISTER "A"          CALL   READ_LCD
        CALL   READ_LCD
        RET

;************************
;    INITIAL LCD    *
;************************
INITLCD       LD     A, 00111000B   ;function set  38H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ;DL=1 8 bit,N=1 1/16 duty,F=0 5x7
        OUT   (PDATA_LCD), A
        CALL  DELAY_LCD1
        CALL  DELAY_LCD1      ;DELAY_LCD > 4.1 ms
        LD    A, 00001111B    ;display on/off control
                ;D=1 off,C=1 cursor on,B=1 blink
        OUT   (PDATA_LCD), A
        CALL  DELAY_LCD1

        LD    A, 00000110B    ;entry mode set
                ;I/D=1 increment,S=0 right
        OUT   (PDATA_LCD), A
        CALL  DELAY_LCD1

CLRSCR_LCD  LD    A, 00000001B    ;clear all display
        OUT   (PDATA_LCD), A
        CALL  DELAY_LCD1
        RET

;***** READ_LCDY SUB *****

DELAY_LCD1  PUSH  BC
        PUSH  AF
        LD    B, 80H
DELAY_LCD12
        DJNZ  DELAY_LCD12
        POP   AF
        POP   BC
        RET

;T1      HALT
;       JR    T1
;
;*********************************
; RX BYTE SUB  8BIT 1 STOP 1 START
; OUTPUT REG=D
;
PCLBY     DI
        EXX
        EX    AF,AF'
        PUSH  IY
        PUSH  IX
        CALL  PCLDLY
        PUSH  BC
        LD    D,0
        LD    B,8
PCLBY12   IN    A,(PB)
        RLA
        RLA
        RR    D
        CALL  PCLDLY
        DJNZ  PCLBY12
        POP   BC
        CALL  PCLDLY
        CALL  PCSBY

        CALL  MENU_MAIN_PCP

        POP   IX
        POP   IY
        EX    AF,AF'
        EXX
        EI
        RETI
;
PCLDLY    LD    A,BUT-4
PCLDLY1   DEC   A
        JR    NZ,PCLDLY1
        RET
;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;**********************************
;
; TX BYTE SUB
;   INPUT DATA REG=D
;
PCSBY       LD    A,10000000B   ;START BIT
            OUT   (PE),A
            CALL  PCSBYD
            PUSH  BC
            LD    B,8           ;8 BIT LOOP
PCSBY11     XOR   A
            RRC   D             ;DATA TO CY
            RRA               ;SHIFT 2 BIT TO PE BIT 5
            RRA
            RRA
            SET   7,A           ;SET PE BIT 7 WDT
            OUT   (PE),A
            CALL  PCSBYD
            DJNZ  PCSBY11
            POP   BC
            LD    A,10100000B   ;STOP BIT
            OUT   (PE),A
            CALL  PCSBYD
            RET
;
PCSBYD      LD    A,BUT
            SUB   2
PCSBYD1     DEC   A
            JR    NZ,PCSBYD1
            RET
;*************************************************************************
MASS_MENU1    DB    'CHECK TOTAL STATUS ',0
MASS_MENU2    DB    'SETUP REAL TIMES',0
MASS_MENU3    DB    'EDIT TIME SET EACH ROOM',0
MASS_MENU4    DB    'DIRECT CONTROL FOR EACH POINT',0
MASS_MENU5    DB    'CLEAR ALARM SENSOR SWITCH',0
MASS_MENU6    DB    'CLEAR SCREEN',0
MASS_MENU7    DB    'SET UP TIMER SWITCH SENSOR',0
MASS_MENU8    DB    'DIRECT CONTROL SWITCH SENSOR',0

;*************************************************************************
CHK_SENS_ON:
            CALL  TIME_FULL       ;'DE' IS STROE VALUE HOUR AND MINUTE
            LD    A, (9E00H)      ;CHKDAY
            SLA   A
CHK_SENS_ON0  LD    HL, SUN_ON + 2000H
            ADD   A, H
            LD    H, A
            PUSH  HL
            POP   IX
            LD    IY, 8500H     ;STORE START TIMMER ON FORM BMP
            LD    C, 32
CHK_SENS_ON1  LD    B, 4
CHK_SENS_ON2  LD    H, (IX)
            LD    L, (IX+1)
            SBC   HL, DE
            CALL  BITMAP_SENS
            INC   IX
            INC   IX
            DJNZ  CHK_SENS_ON2
            SRL   (IY)
            SRL   (IY)
            SRL   (IY)
            SRL   (IY)
            INC   IY
            DEC   C
            JR    NZ, CHK_SENS_ON1
            RET
;*************************************************************************
CHK_SENS_OFF:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CALL   TIME_FULL       ;'DE' IS STROE VALUE HOUR AND MINUTE
        LD     A, (8E00H)      ;CHKDAY
        SLA    A
CHK_SENS_OFF0  LD     HL, SUN_ON + 2100H
        ADD    A, H
        LD     H, A
        PUSH   HL
        POP    IX
        LD     IY, 8600H       ;STORE START TIMMER OFF FORM BMP
        LD     C, 32
CHK_SENS_OFF1  LD     B, 4
CHK_SENS_OFF2  LD     H, (IX)
        LD     L, (IX+1)
        SBC    HL, DE
        CALL   BITMAP_SENS
        INC    IX
        INC    IX
        DJNZ   CHK_SENS_OFF2
        SRL    (IY)
        SRL    (IY)
        SRL    (IY)
        SRL    (IY)
        INC    IY
        DEC    C
        JR     NZ, CHK_SENS_OFF1
        RET

;*************************************************************************
BITMAP_SENS    LD     A,H
        OR     L
        CALL   Z,BMP_SENS1    ;CALL BMP1 IS Z=1
        CALL   NZ,BMP_SENS2   ;CALL BMP2 IS Z=0
        RET
BMP_SENS1      PUSH   AF
        SET    0,(IY)
        RRC    (IY)
        POP    AF
        RET
BMP_SENS2      PUSH   AF
        RES    0,(IY)
        RRC    (IY)
        POP    AF
        RET
;*************************************************************************
OUT_SEN_REAL:
        LD     IY, 8800H
        LD     L, 0          ;ADDRESS FOR OUT PORT
        LD     B, 32
OUT3_REAL_SEN  LD     H, 0          ;DATA FOR OUT PORT
OUT2_REAL_SEN  LD     E, 4
OUT1_REAL_SEN  RRC    (IY)
        JR     NC, OUT0_REAL_SEN
        LD     A, L
        OUT    (050H), A        ;OUT PORT WITH ADDRESS
        LD     A, H
        ADD    A, 08
        OUT    (051H), A        ;OUT PORT WITH DATA
        LD     A, 0FFH       ;TRIG TO SEND '11101111' WILL MAY DELAY
        OUT    (052H), A
        LD     A, 0FFH
        OUT    (052H), A
        CALL   DELAY_SENS

OUT0_REAL_SEN  INC    H
        DEC    E
        JR     NZ, OUT1_REAL_SEN
        INC    IY
        INC    L
        DJNZ   OUT3_REAL_SEN
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;*************************************************************************
OUT_SENS_CL_OP:
        LD    IX, 8400H     ;VALUE STATUS ON FORM BMP
        LD    IY, 8500H     ;VALUE TIMES ON FORM BMP
        LD    HL, 8700H     ;REAL VALUE OUT PORT FORM BMP ON AND OFF

        LD    B, 32
VALUE_ON_SENS  LD    A, (IX)       ;[IX AND IY] XOR [IY]
        AND   (IY)
        XOR   (IY)
        LD    (HL), A
        INC   IX
        INC   IY
        INC   HL
        DJNZ  VALUE_ON_SENS
        LD    IX, 8400H
        LD    IY, 8600H     ;BMP OUTPORT OFF
        LD    HL, 8750H     ;VALUE OUTPORT OFF
        LD    B, 32
VALUE_OFF_SENS LD    A, (IX)
        AND   (IY)
        LD    (HL), A
        INC   IX
        INC   IY
        INC   HL
        DJNZ  VALUE_OFF_SENS
        LD    IX, 8700H
        LD    IY, 8750H
        LD    HL, 8800H
        LD    B, 32
TOTAL_OUT_SENS LD    A, (IX)
        OR    (IY)
        LD    (HL), A
        INC   IX
        INC   IY
        INC   HL
        DJNZ  TOTAL_OUT_SENS
        RET

;*************************************************************************
WELCOME     DB    '********** W E L C O M E **********', 13, 10, 10, 0
MENU_PCP    DB    'A - SHOW STATUS', 13, 10
        DB    'B - SET TIME', 13, 10
        DB    'C - SET TIME ON AND OFF FOR EACH ROOM', 13, 10
        DB    'D - DIRECT CONTROL EQUIPMENT', 13, 10
        DB    'E - CLEAR ALARM SENSOR SWITCH', 13, 10
        DB    'F - CLEAR SCREEN LCD PANEL', 13, 10
        DB    'G - SETUP TIMER SWITCH SENSOR(S)', 13,10
        DB    'H - DIRECT CONTROL SWITCH SENSOR', 13,10
        DB    10
        DB    'SELECT> '
        DB    0

CLRSCR      LD    HL, CLRSCR_STR
        CALL  PUTS
        RET
CLRSCR_STR    DB    27, '[2J', 0

;*************************************************************************
SHOW_STATUS   CALL  INC_LINE
        LD    HL, MEAN_XX
        CALL  PUTS
        CALL  INC_LINE
        LD    HL, POINT_DATA
        CALL  PUTS
        CALL  INC_LINE
        LD    IX, 8000H
        LD    B, 32
SHOW_STATUS_L4 LD    HL, SPACE5
        CALL  PUTS
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LD    A, 15
        CP    E
        JR    NZ, SHOW_STATUS_L5
        CALL  INC_LINE
        LD    HL, MAS_SHOW_STATUS
        CALL  PUTS
        CALL  KBHIT
        CALL  INC_LINE
        LD    HL, POINT_DATA
        CALL  PUTS
        CALL  INC_LINE
        LD    HL, SPACE5
        CALL  PUTS
SHOW_STATUS_L5 PUSH  DE
        LD    HL, ADDRESS_DATA
        LD    D, 0
        LD    A, 32
        SUB   E
        SLA   A
        LD    E, A
        ADD   HL, DE
        POP   DE
        LD    D, (HL)
        CALL  PUTCH
        INC   HL
        LD    D, (HL)
        CALL  PUTCH
        LD    D, ' '
        CALL  PUTCH
;-------------------------------------------------------
SHOW_STATUS_L1 LD    B, 16

SHOW_STATUS_L2 LD    HL, STATUS_OFF
        LD    A, (IX)
        CP    '0'
        JR    Z, SHOW_STATUS_L0

        LD    HL, STATUS_ON
        LD    A, (IX)
        CP    '1'
        JR    Z, SHOW_STATUS_L0

        LD    HL, STATUS_NOT
        LD    A, (IX)
        CP    '2'
        JR    Z, SHOW_STATUS_L0

SHOW_STATUS_L0 LD    D, (HL)
        CALL  PUTS
        INC   IX
        DJNZ  SHOW_STATUS_L2
        CALL  INC_LINE
        DEC   E
        JR    NZ,SHOW_STATUS_L4
        RET                   ;PATCH
;*****************************************************************
MAS_SHOW_STATUS DB    'PLEASE ANY KEY TO CONTINUE',0
STATUS_ON     DB    'ON ',0
STATUS_OFF    DB    'OFF ',0
STATUS_NOT    DB    'NOT ',0
POINT_DATA    DB    'ADDRESS P01 P02 P03 P04 P05 P06 P07 P08 S01 S02 S03 S04 A01 A02 A03 A04',0
ADDRESS_DATA  DB    '0001020304050607080910111213141516171818202122232425262728293031'
SPACE5        DB    '    ',0
MEAN_XX       DB    ' [ Pxx = POINT ] [ Sxx = SENSOR SW. ] [ Axx = ALARM ]',0

;*****************************************************************

SHIRI4        PUSH  BC
        PUSH  AF
        LD    B, 4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SHR_LOOP    SRL   (HL)    ;SHIFT [0-->D7--->D0-->CY] OF (HL)
        DJNZ  SHR_LOOP
        POP   AF
        POP   BC
        RET

;*************************************************************************
INC_LINE    LD    D,13
        CALL  PUTCH
        LD    D,10
        CALL  PUTCH
        RET
;*************************************************************************
SET_TIME    CALL  INC_LINE
        LD    HL, FORMAT_DAY
        CALL  PUTS
        CALL  INC_LINE
        CALL  INC_LINE
        LD    HL, MAS_CUR_TIME
        CALL  PUTS
        CALL  SHOW_CUR_DAY
        LD    D, 20H
        CALL  PUTCH
        CALL  SHOW_CUR_TIME
        CALL  INC_LINE
        LD    HL, MAS_NEW_TIME
        CALL  PUTS
        JP    RX_WRITE_RTC
WRITE_NEW_RTC  LD    IX, 9E10H
        CALL  WRITE_RTC
        CALL  INC_LINE
NOT_SET_TIME   CALL  SHOW_CUR_DAY

        LD    D, 20H
        CALL  PUTCH
        CALL  SHOW_CUR_TIME
        LD    D, 13
        CALL  PUTCH
        CALL  INC_LINE
        RET

SHOW_CUR_DAY   LD    HL, 9E20H     ;STORE VALUE OF TIME
        CALL  READ_RTC
        LD    A, (9E20H)
        SLA   A
        SLA   A
        LD    E, A
        LD    D, 0
        LD    HL, DAYNAME
        ADD   HL, DE
        CALL  PUTS
        RET


SHOW_CUR_TIME  LD    IX, 9E20H
        LD    D, (IX+1)
        CALL  PUTCH
        LD    D, (IX+2)
        CALL  PUTCH
        LD    D, 3AH
        CALL  PUTCH
        LD    D, (IX+3)
        CALL  PUTCH
        LD    D, (IX+4)
        CALL  PUTCH
        LD    D, 3AH
        CALL  PUTCH
        LD    D, (IX+5)
        CALL  PUTCH
        LD    D, (IX+6)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CALL   PUTCH
        ;LD    D, 13        ;START OF LINE
        ;CALL  PUTCH
        ;LD    D, 10        ;INC LINE
        ;CALL  PUTCH
        RET

; READ REAL-TIME CLOCK CHIP
;
; INPUT:     REGISTER C    PORT NUMBER
; OUTPUT:    REGISTER A    DATA READ
;*************************************************************************
READ_RTC    IN    A, (0ADH)    ;GET CONTROL REG.
        OR    1            ;SET HOLD BIT
        OUT   (0ADH), A    ;WRITE TO RTC
RTC_BUSY    IN    A, (0ADH)    ;IS BUSY BIT EQUAL TO ZERO?
        BIT   1, A         ;
        JR    NZ, RTC_BUSY  ; NO, WAIT
        IN    A, (0ACH)      ; YES, READ DATA
        AND   0FH          ;MASK BIT D4-D7
        LD    (HL), A
        INC   HL
        PUSH  BC
        LD    C, 0A5H
        LD    B, 6
READ_LOOP   IN    A, (C)
        AND   0FH
        ADD   A, 30H
        LD    (HL), A
        INC   HL
        DEC   C
        DJNZ  READ_LOOP
        POP   BC
        PUSH  AF
        IN    A, (0ADH)    ;GET CONTROL REG.
        AND   0FEH         ;RESET HOLD BIT
        OUT   (0ADH), A    ;WRITE TO RTC
        POP   AF
        RET
;*************************************************************************
WRITE_RTC   EXX
        LD    D, (IX)
        IN    A, (0ADH)    ;GET CONTROL REG.
        OR    1            ;SET HOLD BIT
        OUT   (0ADH), A    ;WRITE TO RTC
RTC_BUSY_W  IN    A, (0ADH)    ;IS BUSY BIT EQUAL TO ZERO?
        BIT   1, A         ;
        JR    NZ, RTC_BUSY_W ; NO, WAIT
        LD    C, 0ACH
        OUT   (C), D       ; YES, WRITE DATA
        LD    B, 6
        LD    C, 0A5H
WR_RTC_LOOP  INC   IX
        LD    D, (IX)
        OUT   (C), D
        DEC   C
        DJNZ  WR_RTC_LOOP
        AND   0FH          ;MASK BIT D4-D7
        IN    A, (0ADH)    ;GET CONTROL REG.
        AND   0FEH         ;RESET HOLD BIT
        OUT   (0ADH), A    ;WRITE TO RTC
        EXX
        RET
;*************************************************************************
;WRITE RTC
RX_WRITE_RTC  LD    IX, 9E10H
        CALL  GETCH
        LD    A, D         ;CHK SPACE = OUT ROUTINE
        CP    20H
        JP    Z,NOT_SET_TIME
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LD    A, D          ;CHK ENTER = OUT ROUTINE
        CP    13
        JP    Z,NOT_SET_TIME
        LD    A, D
        SUB   30H
        LD    (IX), A
        CALL  PUTCH
        LD    HL, SPACE3
        CALL  PUTS
        CALL  GETCH
        LD    A, D
        SUB   30H
        LD    (IX+1), A
        CALL  PUTCH
        CALL  GETCH
        LD    A, D
        SUB   30H
        LD    (IX+2), A
        CALL  PUTCH
        LD    D, 3AH
        CALL  PUTCH
        CALL  GETCH
        LD    A, D
        SUB   30H
        LD    (IX+3), A
        CALL  PUTCH
        CALL  GETCH
        LD    A, D
        SUB   30H
        LD    (IX+4), A
        CALL  PUTCH
        LD    D, 3AH
        CALL  PUTCH
        CALL  GETCH
        LD    A, D
        SUB   30H
        LD    (IX+5), A
        CALL  PUTCH
        CALL  GETCH
        LD    A, D
        SUB   30H
        LD    (IX+6), A
        CALL  PUTCH
        JP    WRITE_NEW_RTC
        RET
;*************************************************************************

DAYNAME     DB    'SUN',0
        DB    'MON',0
        DB    'TUE',0
        DB    'WEN',0
        DB    'THU',0
        DB    'FRI',0
        DB    'SAT',0.

MAS_CUR_TIME   DB    'Current Day and Time is : ',0
MAS_NEW_TIME   DB    'Enter new Day and Time  : ',0
FORMAT_DAY     DB    '0 = SUN',13,10
        DB    '1 = MON',13,10
        DB    '2 = TUE',13,10
        DB    '3 = WEN',13,10
        DB    '4 = THU',13,10
        DB    '5 = FRI',13,10
        DB    '6 = SAT',13,10
        DB    13,10
        DB    'BREACK PROGRAM BY KEY SPACE MANY',0
SPACE3      DB    ' ',0

;*************************************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
TIME_OPEN_CLOSE CALL   INC_LINE
        LD     HL, FORMAT_DAY
        CALL   PUTS
        CALL   INC_LINE
        LD     HL, MAS_TIME_INDAY
        CALL   PUTS
        CALL   GETCH

        LD     A, D          ;CHK SPACE = OUT ROUTINE
        CP     20H
        JP     Z, END_EDIT_ROOM
        LD     A, D          ;CHK ENTER = OUT ROUTINE
        CP     13
        JP     Z, END_EDIT_ROOM

        LD     A, D
        SUB    30H
        SLA    A
        SLA    A
        LD     E, A
        LD     D, 0
        LD     HL, DAYNAME
        ADD    HL, DE
        CALL   PUTS
        LD     HL, SUN_ON
        LD     A, E
        ADD    A, H
        LD     H, A
        CALL   INC_LINE
        PUSH   HL
        LD     HL, NUMBER_ROOMS
        CALL   PUTS
        POP    HL
        CALL   RX_2BIT_D_TO_H      ;DEC TO HEX
        LD     A, E
        CALL   SRL_A
        LD     D, A
        LD     A, E
        CALL   SLA_A
        LD     E, A
        ADD    HL, DE        ;HL INDEX DAY_OPEN
        CALL   INC_LINE
;*****************************************************************
        PUSH   HL
        POP    IX
        LD     B, 8
        LD     C, 31H
POINT_LOOP    LD     HL, SHOW_POINT_CUR
        CALL   PUTS
        LD     D, C
        CALL   PUTCH
        LD     HL, SPACE3
        CALL   PUTS
        LD     HL, TIME_OLD_OPEN
        CALL   PUTS
        PUSH   IX
SHOW_TIME_OPEN  LD     A, (IX)
        CALL   SRL_A
        ADD    A, 30H
        LD     D, A
        CALL   PUTCH              ;SEND H10
        LD     A, (IX)
        AND    0FH
        ADD    A, 30H
        LD     D, A
        CALL   PUTCH              ;SEND H1
        LD     D, 3AH
        CALL   PUTCH              ;SEND :
        INC    IX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LD    A, (IX)
        CALL  SRL_A
        ADD   A, 30H
        LD    D, A
        CALL  PUTCH             ;SEND M10
        LD    A, (IX)
        AND   0FH
        ADD   A, 30H
        LD    D, A              ;SEND M1
        CALL  PUTCH
        LD    HL, SPACE5
        CALL  PUTS
SHOW_TIME_CLOSE POP   HL
        LD    A, H
        ADD   A, 02H
        LD    H, A
        PUSH  HL
        POP   IY
        LD    HL, TIME_OLD_CLOSE
        CALL  PUTS
        LD    A, (IY)
        CALL  SRL_A
        ADD   A, 30H
        LD    D, A
        CALL  PUTCH             ;SEND H10
        LD    A, (IY)
        AND   0FH
        ADD   A, 30H
        LD    D, A
        CALL  PUTCH             ;SEND H1
        LD    D, 3AH
        CALL  PUTCH             ;SEND :
        INC   IY
        LD    A, (IY)
        CALL  SRL_A
        ADD   A, 30H
        LD    D, A
        CALL  PUTCH             ;SEND M10
        LD    A, (IY)
        AND   0FH
        ADD   A, 30H
        LD    D, A              ;SEND M1
        CALL  PUTCH
        CALL  INC_LINE
;-------------------------------------------------------------------------
EDIT_NEW_T_OP  DEC   IX
        DEC   IY
        LD    HL, TIME_NEW_OPEN
        CALL  PUTS
        CALL  GETCH
        LD    A, D              ;KEY Esc OUT PROGRAM
        CP    1BH
        JP    Z, END_EDIT_ROOM
        LD    A, D              ;PASS EACH POINT
        CP    20H
        JR    NZ, PASS_NEW_T_OP
        LD    HL, SPACE5
        CALL  PUTS
        INC   IX
        INC   IX
        JR    EDIT_NEW_T_CL
 PASS_NEW_T_OP  CALL  PUTCH
        CALL  SLA_A
        LD    E, A
        CALL  GETCH
        CALL  PUTCH
        LD    A, D
        AND   0FH
        OR    E
        LD    (IX), A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        INC   IX
        LD    D, 3AH
        CALL  PUTCH
        CALL  GETCH
        CALL  PUTCH
        LD    A, D
        CALL  SLA_A
        LD    E, A
        CALL  GETCH
        CALL  PUTCH
        LD    A, D
        AND   0FH
        OR    E
        LD    (IX), A
        INC   IX
EDIT_NEW_T_CL  LD    HL, TIME_NEW_CLOSE    ;DIVISION OPEN AND CLOSE
        CALL  PUTS
        CALL  GETCH
        LD    A, D
        CP    20H
        JR    Z, END_EDIT_OP_CL
        CALL  PUTCH
        CALL  SLA_A
        LD    E, A
        CALL  GETCH
        CALL  PUTCH
        LD    A, D
        AND   0FH
        OR    E
        LD    (IY), A
        LD    D, 3AH
        CALL  PUTCH
        INC   IY
        CALL  GETCH
        CALL  PUTCH
        LD'   A, D
        CALL  SLA_A
        LD    E, A
        CALL  GETCH
        CALL  PUTCH
        LD    A, D
        AND   0FH
        OR    E
        LD    (IY), A
        INC   IY
END_EDIT_OP_CL CALL  INC_LINE
        CALL  INC_LINE
        INC   IY
        INC   IY
        INC   C
        DEC   B
        JP    NZ, POINT_LOOP
        CALL  INC_LINE
END_EDIT_ROOM  RET
;*****************************************************************************

MAS_TIME_INDAY  DB    'ENTER CODE DAY FOR SET TIME :',0
DAY_SET_TIME    DB    'SET TIME FOR DAY : ',0
NUMBER_ROOMS    DB    'Enter number room (00-31) : ',0
SHOW_POINT_CUR  DB    'POINT : ',0
TIME_OLD_OPEN   DB    'Current time ON  ',0
TIME_OLD_CLOSE  DB    'Current time OFF  ',0
TIME_NEW_OPEN   DB    '          Edit new time ON ',0
TIME_NEW_CLOSE  DB    '    Edit new time OFF  ',0
ON_LINE_POINT   DB    'Enter number point (1 - 8): ', 0
MAS_ON_LINE     DB    'This is control OPEN and CLOSE to point', 13, 10, 0
STATUS_ON_OFF   DB    'Enter ( ON = 1) OR ( OFF = 0) : ', 0
PRESENT_STA_ON  DB    'PRESENT STATUS          : ON',0
PRESENT_STA_OFF DB    'PRESENT STATUS          :OFF',0
PRESENT_STA_NOT DB    'PRESENT STATUS          :not',0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
ON_LINE_CONTROL DB     'PLEASE ENTER ( 0 = ON )( 1 = OFF ) : ',0

;*************************************************************************
:PROGRAM RECIVE 2 DIGIT FORM DECIMAL TO HEXAGON TRANFER BY REGISTER 'E'
RX_2BIT_D_TO_H CALL    GETCH          ; GET DATA FIRST BYTE TO REG D
         LD     E, D          ; TEMP SAVE
         CALL   PUTCH         ; ECHO
         LD     A, E          ; SAVE TO REG A
         SUB    30H           ; CHANGE ASCII TO HEX
         CALL   SLA_A
         LD     E, A
         CALL   GETCH         ; GET DATA SECOND BYTE
         CALL   PUTCH         ; ECHO
         LD     A, D
         SUB    30H
         OR     E
         LD     E, A
         AND    0F0H
         JR     Z,NOCHANG_DEC_HEX
         CALL   SRL_A
         LD     B, A
         LD     A, 0
MUL_FLOW_INPUT ADD    A, 6
         DJNZ   MUL_FLOW_INPUT
         LD     D, A
         LD     A, E
         SUB    D
         LD     E, A           ;VALUE TRANFER DECIMAL TO HEX
NOCHANG_DEC_HEX RET
;*************************************************************************
ON_LINE      LD     HL, MAS_ON_LINE
         CALL   PUTS
         LD     HL, NUMBER_ROOMS
         CALL   PUTS
         CALL   RX_2BIT_D_TO_H      ;STORAGE 'E'
         CALL   INC_LINE
         LD     HL, ON_LINE_POINT
         CALL   PUTS
         CALL   GETCH
         CALL   PUTCH
         LD     A, D
         SUB    30H
         LD     D, A
         PUSH   BC                  ; SAVE DATA
         PUSH   HL
         LD     L,E
         LD     H,0
         SLA    L
         SLA    L
         SLA    L
         SLA    L
         RL     H                   ; SHIFT WITH CARRY
         LD     A,D                 ; GET POSITION
         DEC    A                   ; DECREASE BY 1
         ADD    A,L
         LD     L,A
         JR     NC,$+3
         INC    H
         PUSH   DE
         LD     DE,8000H
         ADD    HL,DE
         LD     A,(HL)
         PUSH   HL
         AND    00000011B
         LD     HL,STATUS_MAS
         SLA    A
         SLA    A
         SLA    A
         SLA    A
         SLA    A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ADD   A,L
        LD    L,A
        JR    NC,$+3
        INC   H
        CALL  INC_LINE              ; MAKE LINEFEED
        CALL  PUTS                  ; SHOW STATUS
        CALL  INC_LINE
        LD    HL,STATUS_ON_OFF      ;
        CALL  PUTS
        CALL  GETCH
        CALL  PUTCH
        POP   HL
        LD    A,(HL)
        XOR   D                     ; SAME AS PREVIOUS
        POP   DE
        JR    Z,ON_LINE_RETURN
        DEC   D
        LD    (8800H),DE
        LD    A, E
        OUT   (050H), A
        LD    A, D
        OUT   (051H), A
        LD    A, 0FFH
        OUT   (052H), A
        LD    A; 0FFH
        OUT   (052H), A
        CALL  DELAY_SENS

ON_LINE_RETURN POP   HL
        POP   BC
        RST

STATUS_MAS   DB   'PRESENT STATUS      :OFF',0
        DB    'PRESENT STATUS       : ON',0
        DB    'PRESENT STATUS       :not',0
;*************************************************************************
; RESET SENSOR ROUTINE
;
RST_SENSOR   LD   HL, MAS_RST_SENSOR
        CALL  PUTS
        LD    HL, NUMBER_ROOMS
        CALL  PUTS
        CALL  RX_2BIT_D_TO_H        ;STORAGE 'E'
        CALL  INC_LINE
        LD    HL, SENSOR_POINT
        CALL  PUTS
        CALL  GETCH
        CALL  PUTCH
        LD    A, D
        SUB   30H
        ADD   A,7                   ; 1-4 -> 8-B
        LD    D, A

        LD    A, E
        OUT   (050H), A
        LD    A, D
        OUT   (051H), A
        LD    A, 0FFH
        OUT   (052H), A
        LD    A, 0FFH
        OUT   (052H), A
        CALL  DELAY_SENS

        LD    A, E
        OUT   (050H), A
        LD    A, D
        OUT   (051H), A
        LD    A, 0FFH
        OUT   (052H), A
        LD    A, 0FFH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        OUT   (052H), A
        CALL  DELAY_SENS
        RET

MAS_RST_SENSOR  DB    'RESET SENSOR ',0
SENSOR_POINT    DB    'Enter number point (1-4): ',0

;*****************************************************************
SETUP_TIME_SENSOR:
        CALL  INC_LINE
        LD    HL, FORMAT_DAY
        CALL  PUTS
        CALL  INC_LINE
        LD    HL, MAS_TIME_INDAY
        CALL  PUTS
        CALL  GETCH

        LD    A, D          ;CHK SPACE = OUT ROUTINE
        CP    20H
        JP    Z, END_SETUP_SEN
        LD    A, D          ;CHK ENTER = OUT ROUTINE
        CP    13
        JP    Z, END_SETUP_SEN

        LD    A, D
        SUB   30H
        SLA   A
        PUSH  AF
        SLA   A
        LD    E, A
        LD    D, 0
        LD    HL, DAYNAME
        ADD   HL, DE
        CALL  PUTS
        LD    HL, SUN_ON+2000H
        POP   AF
        ADD   A, H
        LD    H, A
        CALL  INC_LINE
        PUSH  HL            ;STORG INITIAL OF EACH DAY
        LD    HL, NUMBER_ROOMS
        CALL  PUTS
        POP   HL
        CALL  RX_2BIT_D_TO_H      ;DEC TO HEX
        SLA   E
        SLA   E
        SLA   E             ;VALUE ADDRESS OF POINT X 8
;       LD    A, E
;       CALL  SRL_A
;       LD    D, A
;       LD    A, E
;       CALL  SLA_A
;       LD    E, A
        LD    D, 0
        ADD   HL, DE        ;HL INDEX DAY_OPEN
        CALL  INC_LINE
;*****************************************************************
        PUSH  HL            ;VALUE ADDRESS OF TIME ON POINT OF DAY
        POP   IX
        LD    B, 4
        LD    C, 31H
POINT_LOOP_SEN
        LD    HL, SHOW_POINT_CUR
        CALL  PUTS
        LD    D, C
        CALL  PUTCH
        LD    HL, SPACE3
        CALL  PUTS
        LD    HL, TIME_OLD_OPEN
        CALL  PUTS
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        PUSH   IX
SHOW_TIME_OPEN_SEN
        LD     A, (IX)
        CALL   SRL_A
        ADD    A, 30H
        LD     D, A
        CALL   PUTCH            ;SEND H10
        LD     A, (IX)
        AND    0FH
        ADD    A, 30H
        LD     D, A
        CALL   PUTCH            ;SEND H1
        LD     D, 3AH
        CALL   PUTCH            ;SEND :
        INC    IX
        LD     A, (IX)
        CALL   SRL_A
        ADD    A, 30H
        LD     D, A
        CALL   PUTCH            ;SEND M10
        LD     A, (IX)
        AND    0FH
        ADD    A, 30H
        LD     D, A             ;SEND M1
        CALL   PUTCH
        LD     HL, SPACE5
        CALL   PUTS
SHOW_TIME_CLOSE_SEN
        POP    HL
        LD     A, H
        ADD    A, 01H
        LD     H, A
        PUSH   HL
        POP    IY               ;VALUE OF ADDRESS OFF EACH POINT
        LD     HL, TIME_OLD_CLOSE
        CALL   PUTS
        LD     A, (IY)
        CALL   SRL_A
        ADD    A, 30H.
        LD     D, A
        CALL   PUTCH            ;SEND H10
        LD     A, (IY)
        AND    0FH
        ADD    A, 30H
        LD     D, A
        CALL   PUTCH            ;SEND H1
        LD     D, 3AH
        CALL   PUTCH            ;SEND :
        INC    IY
        LD     A, (IY)
        CALL   SRL_A
        ADD    A, 30H
        LD     D, A
        CALL   PUTCH            ;SEND M10
        LD     A, (IY)
        AND    0FH
        ADD    A, 30H
        LD     D, A             ;SEND M1
        CALL   PUTCH
        CALL   INC_LINE
;-------------------------------------------------------------------------------
EDIT_NEW_T_OP_SEN
        DEC    IX
        DEC    IY
        LD     HL, TIME_NEW_OPEN
        CALL   PUTS
        CALL   GETCH
        LD     A, D             ;KEY Esc OUT PROGRAM
        CP     1BH
        JP     Z, END_SETUP_SEN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LD    A, D              ;PASS EACH POINT
        CP    20H
        JR    NZ,PASS_NEW_T_OP_SEN
        LD    HL, SPACE8
        CALL  PUTS
        INC   IX
        INC   IX
        JR    EDIT_NEW_T_CL_SEN
PASS_NEW_T_OP_SEN
        CALL  PUTCH
        CALL  SLA_A
        LD    E, A
        CALL  GETCH
        CALL  PUTCH
        LD    A, D
        AND   0FH
        OR    E
        LD    (IX), A
        INC   IX
        LD    D, 3AH
        CALL  PUTCH
        CALL  GETCH
        CALL  PUTCH
        LD    A, D
        CALL  SLA_A
        LD    E, A
        CALL  GETCH
        CALL  PUTCH
        LD    A, D
        AND   0FH
        OR    E
        LD    (IX), A
        INC   IX
EDIT_NEW_T_CL_SEN
        LD    HL, TIME_NEW_CLOSE    ;DIVISION OPEN AND CLOSE
        CALL  PUTS
        CALL  GETCH
        LD    A, D
        CP    20H
        JR    Z, END_EDIT_OP_CL_SEN
        CALL  PUTCH
        CALL  SLA_A
        LD    E, A
        CALL  GETCH
        CALL  PUTCH
        LD    A, D
        AND   0FH
        OR    E
        LD    (IY), A
        LD    D, 3AH
        CALL  PUTCH
        INC   IY
        CALL  GETCH
        CALL  PUTCH
        LD    A, D
        CALL  SLA_A
        LD    E, A
        CALL  GETCH
        CALL  PUTCH
        LD    A, D
        AND   0FH
        OR    E
        LD    (IY), A
        INC   IY
END_EDIT_OP_CL_SEN
        CALL  INC_LINE
        CALL  INC_LINE
        INC   IY
        INC   IY
        INC   C
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DEC   B
        JP    NZ, POINT_LOOP_SEN
        CALL  INC_LINE
END_SETUP_SEN  RET
;*****************************************************************************

;
DIRECT_SENSOR_CONTROL:
        LD    HL, MAS_ON_LINE
        CALL  PUTS
        LD    HL, NUMBER_ROOMS
        CALL  PUTS
        CALL  RX_2BIT_D_TO_H      ;STORAGE 'E'
        CALL  INC_LINE
        LD    HL, SENSOR_POINT
        CALL  PUTS
        CALL  GETCH
        CALL  PUTCH
        INC   D
        LD    A, D
        SUB   30H-7
        LD    D, A
        PUSH  BC                  ; SAVE DATA
        PUSH  HL
        LD    L,E
        LD    H,0
        SLA   L
        SLA   L
        SLA   L
        SLA   L
        RL    H                   ; SHIFT WITH CARRY
        LD    A,D                 ; GET POSITION
        DEC   A                   ; DECREASE BY 1
        ADD   A,L
        LD    L,A
        JR    NC,$+3
        INC   H
        PUSH  DE
        LD    DE,8000H
        ADD   HL,DE
        LD    A,(HL)
        PUSH  HL
        AND   00000011B
        LD    HL,STATUS_MAS
        SLA   A
        SLA   A
        SLA   A
        SLA   A
        SLA   A
        ADD   A,L
        LD    L,A
        JR    NC,$+3
        INC   H
        CALL  INC_LINE            ; MAKE LINEFEED
        CALL  PUTS                ; SHOW STATUS
        CALL  INC_LINE
        LD    HL,STATUS_ON_OFF    ;
        CALL  PUTS
        CALL  GETCH
        CALL  PUTCH
        POP   HL
        LD    A,(HL)
        XOR   D                   ; SAME AS PREVIOUS
        POP   DE
        JR    Z,DIRECT_RETURN
        DEC   D
        LD    A, E
        OUT   (050H), A
        LD    A, D
        OUT   (051H), A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        LD    A, 0EFH
        OUT   (052H), A
        LD    A, 0FFH
        OUT   (052H), A
        CALL  DELAY_SENS

DIRECT_RETURN  POP   HL
        POP   BC
        RET

;*************************************************************************

SHOW_MENU_CRT  CALL  INC_LINE
        LD    HL, WELCOME
        CALL  PUTS
        LD    HL, MENU_PCP
        CALL  PUTS
        RET
;*************************************************************************
;RESET SENSOR SWITCH INITIAL
RES_SENS:    PUSH  AF
        PUSH  BC
        PUSH  DE
        PUSH  HL

        CALL  INC_LINE
        LD    HL, MAS_RES_1
        CALL  PUTS
        LD    HL, 900CH-1
        LD    C, 32
        LD    D, 0

RES_SENS_1   LD    B, 4
        LD    E, 7

RES_SENS_2   INC   HL
        INC   E
        LD    A, (HL)
        SUB   31H
        JR    NZ, RES_SENS_3

        LD    A, D
        OUT   (050H), A
        LD    A, E
        OUT   (051H), A
        LD    A, 0EFH
        OUT   (052H), A
        LD    A, 0FFH
        OUT   (052H), A
        CALL  DELAY_SENS

        LD    A, D
        OUT   (050H), A
        LD    A, E
        OUT   (051H), A
        LD    A, 0EFH
        OUT   (052H), A
        LD    A, 0FFH
        OUT   (052H), A
        CALL  DELAY_SENS

RES_SENS_3   DJNZ  RES_SENS_2
        XOR   A
        PUSH  DE
        LD    DE, 0CH
        ADC   HL, DE
        POP   DE
        INC   D
        DEC   C
RES_SENS_4   JR    NZ, RES_SENS_1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CALL   INC_LINE
        LD     HL, MAS_RES_2
        CALL   PUTS

        POP    HL
        POP    DE
        POP    BC
        POP    AF
        RET

MAS_RES_1     DB     'PLEASE WAIT ... CLEAR ALARM ',0
MAS_RES_2     DB     'OK !',0

;*************************************************************************

RX_INT        DI

        EXX
        EX     AF,AF'
        PUSH   IX
        PUSH   IY

        CALL   GETCH
        CALL   MENU_MAIN_PCP

        POP    IY
        POP    IX
        EX     AF,AF'
        EXX

        EI

        RETI

RX_INT_N      DI

        EXX
        EX     AF,AF'
        PUSH   IX
        PUSH   IY

        LD     IX, MASS_HALT
        CALL   WRLINE

        POP    IY
        POP    IX
        EX     AF,AF'
        EXX

        EI

        RETN
MASS_HALT     DB     'HALT SYSTEM YOU SHOULD BE RESET PROGRAM',0

;*************************************************************************
        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. อ.อรรถสิทธ์ หง่าสกุล, ไมโครโปรเซสเซอร์พื้นฐาน,ตำราชุด วิศวกรรมศาสตร์ คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
2. บริษัท อีทีที จำกัด,Dot Matrix LCD Module
3. บริษัท อีทีที จำกัด,CP-Z84C11 PLUS
4. วิบูลย์ ชื่นแขก, 2532 , ไมโครโปรเซสเซอร์พื้นฐาน,สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ
5. โอภาส ศิริครรชิตถาวร , GAL เทคโนโลยีล่าสุดของอุปกรณ์ลอจิกอะเรย์ เซมิคอนดัคเตอร์ อิเลคทรอนิกส์ ฉบับที่ 113 มกราคม 2535 บริษัท ซีเอ็ดยูเคชั่น จำกัด
6. Ridnay Zaks, 1980, Edition,Programming the Z80,SYBEX,United State of America

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA
SEMICONDUCTOR
TECHNICAL DATA

MOTOROLA SC (TELECOM)

MC145026
MC145027
MC145028
SC41343
SC41344

Encoder and Decoder Pairs
CMOS

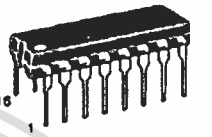
These devices are designed to be used as encoder/decoder pairs in remote control applications.

The MC145026 encodes nine lines of information and serially sends this information upon receipt of a transmit enable ($\overline{TE}$) signal. The nine lines may be encoded with trinary data (low, high, or open) or binary data (low or high). The words are transmitted twice per encoding sequence to increase security.

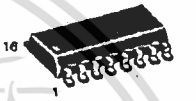
The MC145027 decoder receives the serial stream and interprets five of the trinary digits as an address code. Thus, 243 addresses are possible. If binary data is used at the encoder, 32 addresses are possible. The remaining serial information is interpreted as four bits of binary data. The valid transmission (VT) output goes high on the MC145027 when two conditions are met. First, two addresses must be consecutively received (in one encoding sequence) which both match the local address. Second, the 4 bits of data must match the last valid data received. The active VT indicates that the information at the Data output pins has been updated.

The MC145028 decoder treats all nine trinary digits as an address which allows 19,683 codes. If binary data is encoded, 512 codes are possible. The VT output goes high on the MC145028 when two addresses are consecutively received (in one encoding sequence) which both match the local address.

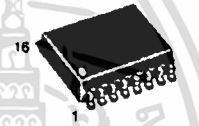
- Operating Temperature Range: −40 to +85°C
- Very-Low Standby Current for the Encoder: 300 nA Maximum @ 25°C
- Interfaces with RF, Ultrasonic, or Infrared Modulators and Demodulators
- RC Oscillator, No Crystal Required
- High External Component Tolerance; Can Use ±5% Components
- Internal Power-On Reset Forces All Decoder Outputs Low
- For Infrared Applications, See Applications Notes AN1016 and AN1126
- Operating Voltage Range: MC145026 = 2.5 to 18 V*
  MC145027, MC145028 = 4.5 to 18 V
- Low-Voltage Versions Available:
  SC41343 = 2.8 to 10 V Version of the MC145027
  SC41344 = 2.8 to 10 V Version of the MC145028



P SUFFIX
PLASTIC DIP
CASE 648



D SUFFIX
SOG
CASE 751B



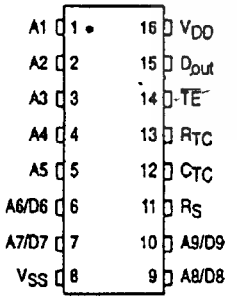
DW SUFFIX
SOG
CASE 751G

ORDERING INFORMATION

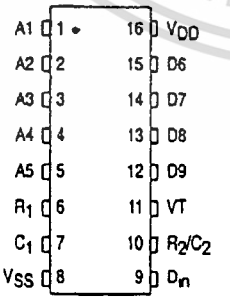
MC145026P	Plastic DIP
MC145026D	SOG
MC145027P, SC41343P	Plastic DIP
MC145027DW, SC41343DW	SOG
MC145028P, SC41344P	Plastic DIP
MC145028DW, SC41344DW	SOG

PIN ASSIGNMENTS

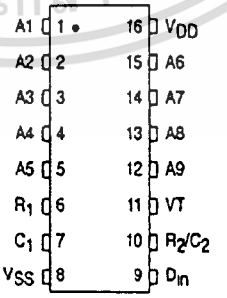
MC145026
ENCODER



MC145027/SC41343
DECODERS



MC145028/SC41344
DECODERS



*All MC145026 devices manufactured after date code 9314 or 314 are guaranteed over this wider voltage range. All previous designs using the low-voltage SC41342 should convert to the MC145026, which is a drop-in replacement. The SC41342 part number will be discontinued.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ted From CAPS XPert Version 1.2 This Material Copyrighted By Its Respective Manufacturer.

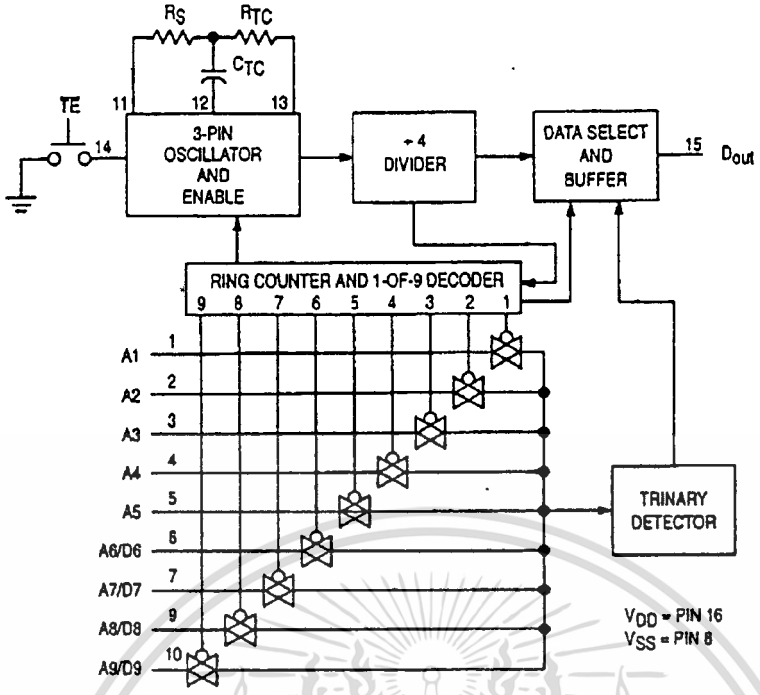


Figure 1. MC145026 Encoder Block Diagram

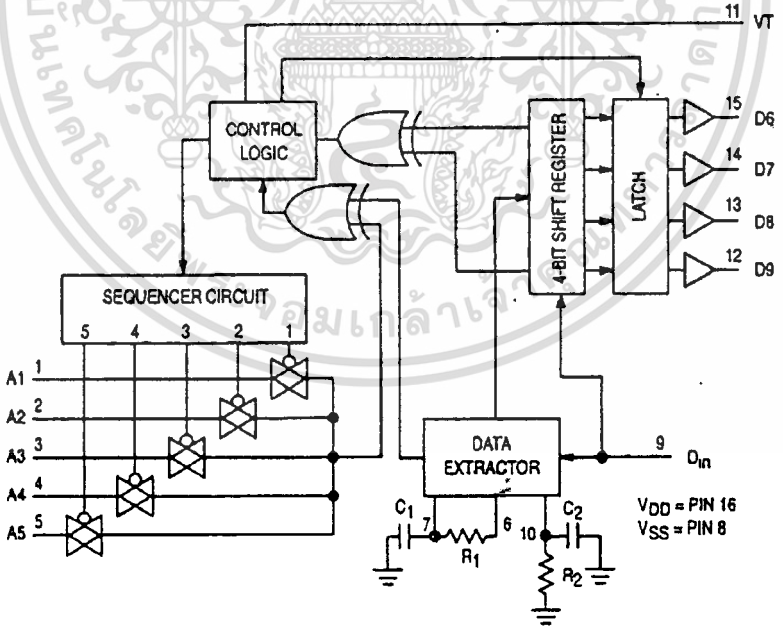


Figure 2. MC145027 Decoder Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
ted From CAPS XPert Version 1.2 This Material Copyrighted By Its Respective Manufacturer.

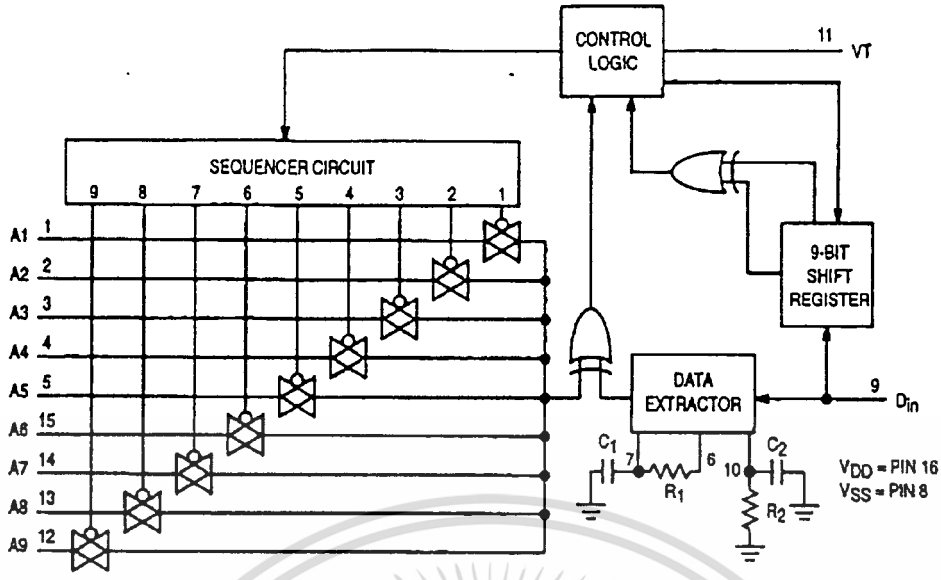


Figure 3. MC145028 Decoder Block Diagram

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

Symbol	Parameter	Value	Unit
V_{DD}	DC Supply Voltage (except SC41343, SC41344)	−0.5 to +18	V
V_{DD}	DC Supply Voltage (SC41343, SC41344 only)	−0.5 to +10	V
V_{in}	DC Input Voltage	−0.5 to V_{DD} +0.5	V
V_{out}	DC Output Voltage	−0.5 to V_{DD} +0.5	V
I_{in}	DC Input Current, per Pin	±10	mA
I_{out}	DC Output Current, per Pin	±10	mA
P_D	Power Dissipation, per Package	500	mW
T_{stg}	Storage Temperature	−65 to +150	°C
T_L	Lead Temperature, 1 mm from Case for 10 Seconds	260	°C

* Maximum Ratings are those values beyond which damage to the device may occur. Functional operation should be restricted to the limits in the Electrical Characteristics tables or Pin Descriptions section.

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
ted From CAPS XPert Version 1.2 This Material Copyrighted By Its Respective Manufacturer.

ELECTRICAL CHARACTERISTICS — MC145026*, MC145027, and MC145028 (Voltage Referenced to V_{SS})

Symbol	Characteristic	V_{DD} V	Guaranteed Limit						Unit
			−40°C		25°C		+85°C		
			Min	Max	Min	Max	Min	Max	
V_{OL}	Low-Level Output Voltage (V_{in} = V_{DD} or 0)	5.0	—	0.05	—	0.05	—	0.05	V
		10	—	0.05	—	0.05	—	0.05	
		15	—	0.05	—	0.05	—	0.05	
V_{OH}	High-Level Output Voltage (V_{in} = 0 or V_{DD})	5.0	4.95	—	4.95	—	4.95	—	V
		10	9.95	—	9.95	—	9.95	—	
		15	14.95	—	14.95	—	14.95	—	
V_{IL}	Low-Level Input Voltage								V
	(V_{out} = 4.5 or 0.5 V)	5.0	—	1.5	—	1.5	—	1.5	
	(V_{out} = 9.0 or 1.0 V)	10	—	3.0	—	3.0	—	3.0	
	(V_{out} = 13.5 or 1.5 V)	15	—	4.0	—	4.0	—	4.0	
V_{IH}	High-Level Input Voltage								V
	(V_{out} = 0.5 or 4.5 V)	5.0	3.5	—	3.5	—	3.5	—	
	(V_{out} = 1.0 or 9.0 V)	10	7.0	—	7.0	—	7.0	—	
	(V_{out} = 1.5 or 13.5 V)	15	11	—	11	—	11	—	
I_{OH}	High-Level Output Current								mA
	(V_{out} = 2.5 V)	5.0	−2.5	—	−2.1	—	−1.7	—	
	(V_{out} = 4.6 V)	5.0	−0.52	—	−0.44	—	−0.36	—	
	(V_{out} = 9.5 V)	10	−1.3	—	−1.1	—	−0.9	—	
	(V_{out} = 13.5 V)	15	−3.6	—	−3.0	—	−2.4	—	
I_{OL}	Low-Level Output Current								mA
	(V_{out} = 0.4 V)	5.0	0.52	—	0.44	—	0.36	—	
	(V_{out} = 0.5 V)	10	1.3	—	1.1	—	0.9	—	
	(V_{out} = 1.5 V)	15	3.6	—	3.0	—	2.4	—	
I_{in}	Input Current — $\overline{TE}$ (MC145026, Pull-up Device)	5.0	—	—	3.0	11	—	—	µA
		10	—	—	16	60	—	—	
		15	—	—	35	120	—	—	
I_{in}	Input Current R_S (MC145026), D_{in} (MC145027, MC145028)	15	—	±0.3	—	±0.3	—	±1.0	µA
I_{in}	Input Current								µA
	A1–A5, A6/D6–A9/D9 (MC145026),	5.0	—	—	—	±110	—	—	
	A1–A5 (MC145027),	10	—	—	—	±500	—	—	
	A1–A9 (MC145028)	15	—	—	—	±1000	—	—	
C_{in}	Input Capacitance (V_{in} = 0)	—	—	—	—	7.5	—	—	pF
I_{DD}	Quiescent Current — MC145026	5.0	—	—	—	0.1	—	—	µA
		10	—	—	—	0.2	—	—	
		15	—	—	—	0.3	—	—	
I_{DD}	Quiescent Current — MC145027, MC145028	5.0	—	—	—	50	—	—	µA
		10	—	—	—	100	—	—	
		15	—	—	—	150	—	—	
I_{dd}	Dynamic Supply Current — MC145026 (f_c = 20 kHz)	5.0	—	—	—	200	—	—	µA
		10	—	—	—	400	—	—	
		15	—	—	—	600	—	—	
I_{dd}	Dynamic Supply Current — MC145027, MC145028 (f_c = 20 kHz)	5.0	—	—	—	400	—	—	µA
		10	—	—	—	800	—	—	
		15	—	—	—	1200	—	—	

*Also see next Electrical Characteristics table for 2.5 V specifications.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
rom CAPS XPert Version 1.2 This Material Copyrighted By Its Respective Manufacturer.

ELECTRICAL CHARACTERISTICS — MC145026 (Voltage Referenced to V_{SS})

Symbol	Characteristic	V_{DD} V	Guaranteed Limit						Unit
			−40°C		25°C		+85°C		
			Min	Max	Min	Max	Min	Max	
V_{OL}	Low-Level Output Voltage (V_{in} = 0 V or V_{DD})	2.5	—	0.05	—	0.05	—	0.05	V
V_{OH}	High-Level Output Voltage (V_{in} = 0 V or V_{DD})	2.5	2.45	—	2.45	—	2.45	—	V
V_{IL}	Low-Level Input Voltage (V_{out} = 0.5 V or 2.0 V)	2.5	—	0.3	—	0.3	—	0.3	V
V_{IH}	High-Level Input Voltage (V_{out} = 0.5 V or 2.0 V)	2.5	2.2	—	2.2	—	2.2	—	V
I_{OH}	High-Level Output Current (V_{out} = 1.25 V)	2.5	0.28	—	0.25	—	0.2	—	mA
I_{OL}	Low-Level Output Current (V_{out} = 0.4 V)	2.5	0.22	—	0.2	—	0.16	—	mA
I_{in}	Input Current ($\overline{TE}$ — Pull-Up Device)	2.5	—	—	0.09	1.8	—	—	µA
I_{in}	Input Current (A1–A5, A6/D6–A9/D9)	2.5	—	—	—	±25	—	—	µA
I_{DD}	Quiescent Current	2.5	—	—	—	0.05	—	—	µA
I_{dd}	Dynamic Supply Current (f_c = 20 kHz)	2.5	—	—	—	40	—	—	µA

ELECTRICAL CHARACTERISTICS — SC41343 and SC41344 (Voltage Referenced to V_{SS})

Symbol	Characteristic	V_{DD} V	Guaranteed Limit						Unit
			−40°C		25°C		+85°C		
			Min	Max	Min	Max	Min	Max	
V_{OL}	Low-Level Output Voltage (V_{in} = 0 V or V_{DD})	2.8	—	0.05	—	0.05	—	0.05	V
		5.0	—	0.05	—	0.05	—	0.05	
		10	—	0.05	—	0.05	—	0.05	
V_{OH}	High-Level Output Voltage (V_{in} = 0 V or V_{DD})	2.8	2.75	—	2.75	—	2.75	—	V
		5.0	4.95	—	4.95	—	4.95	—	
		10	9.95	—	9.95	—	9.95	—	
V_{IL}	Low-Level Input Voltage								V
	(V_{out} = 2.3 V or 0.5 V)	2.8	—	0.84	—	0.84	—	0.84	
	(V_{out} = 4.5 V or 0.5 V)	5.0	—	1.5	—	1.5	—	1.5	
	(V_{out} = 9.0 V or 1.0 V)	10	—	3.0	—	3.0	—	3.0	
V_{IH}	High-Level Input Voltage								V
	(V_{out} = 0.5 V or 2.3 V)	2.8	1.96	—	1.96	—	1.96	—	
	(V_{out} = 0.5 V or 4.5 V)	5.0	3.5	—	3.5	—	3.5	—	
	(V_{out} = 1.0 V or 9.0 V)	10	7.0	—	7.0	—	7.0	—	
I_{OH}	High-Level Output Current								mA
	(V_{out} = 1.4 V)	2.8	−0.73	—	−0.7	—	−0.55	—	
	(V_{out} = 4.5 V)	5.0	−0.59	—	−0.5	—	−0.41	—	
	(V_{out} = 9.0 V)	10	−1.3	—	−1.1	—	−0.9	—	
I_{OL}	Low-Level Output Current								mA
	(V_{out} = 0.4 V)	2.8	0.35	—	0.3	—	0.24	—	
	(V_{out} = 0.5 V)	5.0	0.8	—	0.6	—	0.4	—	
	(V_{out} = 1.0 V)	10	3.5	—	2.9	—	2.3	—	
I_{in}	Input Current — D_{in}	10	—	±0.3	—	±0.3	—	±1.0	µA
I_{in}	Input Current	2.8	—	—	—	±30	—	—	µA
	A1–A5 (SC41343)	5.0	—	—	—	±140	—	—	
	A1–A9 (SC41344)	10	—	—	—	±600	—	—	
C_{in}	Input Capacitance (V_{in} = 0)	—	—	—	—	7.5	—	—	pF
I_{DD}	Quiescent Current	2.8	—	—	—	60	—	—	µA
		5.0	—	—	—	75	—	—	
		10	—	—	—	150	—	—	
I_{dd}	Dynamic Supply Current (f_c = 20 kHz)	2.8	—	—	—	300	—	—	µA
		5.0	—	—	—	500	—	—	
		10	—	—	—	1000	—	—	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
Scanned From CAPS XPert Version 1.2 This Material Copyrighted By Its Respective Manufacturer.

SWITCHING CHARACTERISTICS — MC145026*, MC145027, and MC145028 (C_L = 50 pF, T_A = 25°C)

Symbol	Characteristic	Figure #	V_{DD}	Guaranteed Limit Min	Guaranteed Limit Max	Unit
t_{TLH}, t_{THL}	Output Transition Time	4, 8	5.0 10 15	— — —	200 100 80	ns
t_r	D_{in} Rise Time — Decoders	5	5.0 10 15	— — —	15 15 15	µs
t_f	D_{in} Fall Time — Decoders	5	5.0 10 15	— — —	15 5.0 4.0	µs
f_{osc}	Encoder Clock Frequency	6	5.0 10 15	0.001 0.001 0.001	2.0 5.0 10	MHz
f	Decoder Frequency — Referenced to Encoder Clock	12	5.0 10 15	1.0 1.0 1.0	240 410 450	kHz
t_W	$\overline{TE}$ Pulse Width — Encoders	7	5.0 10 15	65 30 20	— — —	ns

*Also see next Switching Characteristics table for 2.5 V specifications.

SWITCHING CHARACTERISTICS — MC145026 (C_L = 50 pF, T_A = 25°C)

Symbol	Characteristic	Figure #	V_{DD}	Guaranteed Limit Min	Guaranteed Limit Max	Unit
t_{TLH}, t_{THL}	Output Transition Time	4, 8	2.5	—	450	ns
f_{osc}	Encoder Clock Frequency	6	2.5	1.0	250	kHz
t_W	$\overline{TE}$ Pulse Width	7	2.5	1.5	—	µs

SWITCHING CHARACTERISTICS — SC41343 and SC41344 (C_L = 50 pF, T_A = 25°C)

Symbol	Characteristic	Figure #	V_{DD}	Guaranteed Limit Min	Guaranteed Limit Max	Unit
t_{TLH}, t_{THL}	Output Transition Time	4, 8	2.8 5.0 10	— — —	320 200 100	ns
t_r	D_{in} Rise Time	5	2.8 5.0 10	— — —	15 15 15	µs
t_f	D_{in} Fall Time	5	2.8 5.0 10	— — —	15 15 5.0	µs
f	Decoder Frequency — Referenced to Encoder Clock	12	2.8 5.0 10	1.0 1.0 1.0	100 240 410	kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
ted From CAPS XPert Version 1.2 This Material Copyrighted By Its Respective Manufacturer.

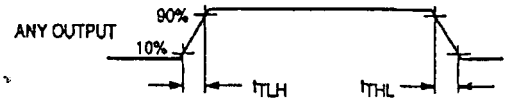


Figure 4.

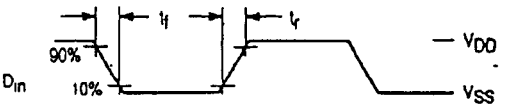


Figure 5.

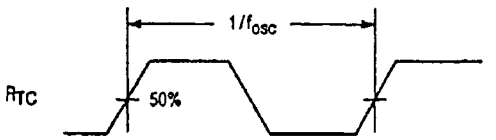


Figure 6.

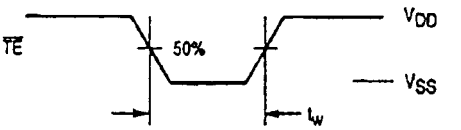


Figure 7.

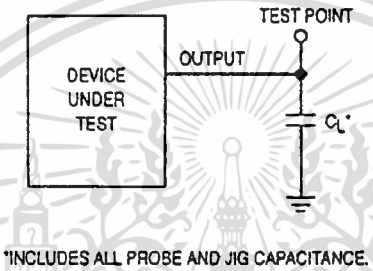


Figure 8. Test Circuit

OPERATING CHARACTERISTICS

MC145026

The encoder serially transmits trinary data as defined by the state of the A1–A5 and A6/D6–A9/D9 input pins. These pins may be in either of three states (low, high, or open) allowing 19,683 possible codes. The transmit sequence is initiated by a low level on the $\overline{TE}$ input pin. Upon power up, the MC145026 can continuously transmit as long as $\overline{TE}$ remains low (also, the device can transmit two-word sequences by pulsing $\overline{TE}$ low). However, no MC145026 application should be designed to rely upon the first data word transmitted immediately after power up because this word may be invalid. Between the two data words, no signal is sent for three data periods (see Figure 10).

Each transmitted trinary digit is encoded into pulses (see Figure 11). A logic 0 (low) is encoded as two consecutive short pulses, a logic 1 (high) as two consecutive long pulses, and an open (high impedance) as a long pulse followed by a short pulse. The input state is determined by using a weak "output" device to try to force each input high then low. If only a high state results from the two tests, the input is assumed to be hardwired to V_{DD}. If only a low state is obtained, the input is assumed to be hardwired to V_{SS}. If both a high and a low can be forced at an input, an open is assumed and is encoded as such. The "high" and "low" levels are 70% and 30% of the supply voltage as shown in the Electrical Characteristics Table. The weak "output" device sinks/sources up to 110 µA at a 5-V supply level, 500 µA at 10 V, and 1 mA at 15 V.

The $\overline{TE}$ input has an internal pull-up device so that a simple switch may be used to force the input low. While $\overline{TE}$ is high, the encoder is completely disabled, the oscillator is inhibited, and the current drain is reduced to quiescent current. When $\overline{TE}$ is brought low, the oscillator is started and the transmit sequence begins. The inputs are then sequentially selected, and determinations are made as to the input logic states. This information is serially transmitted via the D_{out} pin.

MC145027

This decoder receives the serial data from the encoder and outputs the data, if it is valid. The transmitted data, consisting of two identical words, is examined bit by bit during reception. The first five trinary digits are assumed to be the address. If the received address matches the local address, the next four (data) bits are internally stored, but are not transferred to the output data latch. As the second encoded word is received, the address must again match. If a match occurs, the new data bits are checked against the previously stored data bits. If the two nibbles of data (four bits each) match, the data is transferred to the output data latch by VT and remains until new data replaces it. At the same time, the VT output pin is brought high and remains high until an error is received or until no input signal is received for four data periods (see Figure 10).

Although the address information may be encoded in trinary, the data information must be either a 1 or 0. A trinary (open) data line is decoded as a logic 1.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
From CAPS XPert Version 1.2 This Material Copyrighted By Its Respective Manufacturer.

OLMC Configuration Details (Continued)

As mentioned in the Functional Description, the OLMC is responsible for selecting input and/or output paths, registered vs. combinatorial outputs, active-high or low polarity, and common vs. locally-controlled TRI-STATE control. Additionally, the OLMCs select between alternate logic array input paths to maintain JEDEC cell-map compatibility with either "small-PAL" or "medium-PAL" logic arrays.

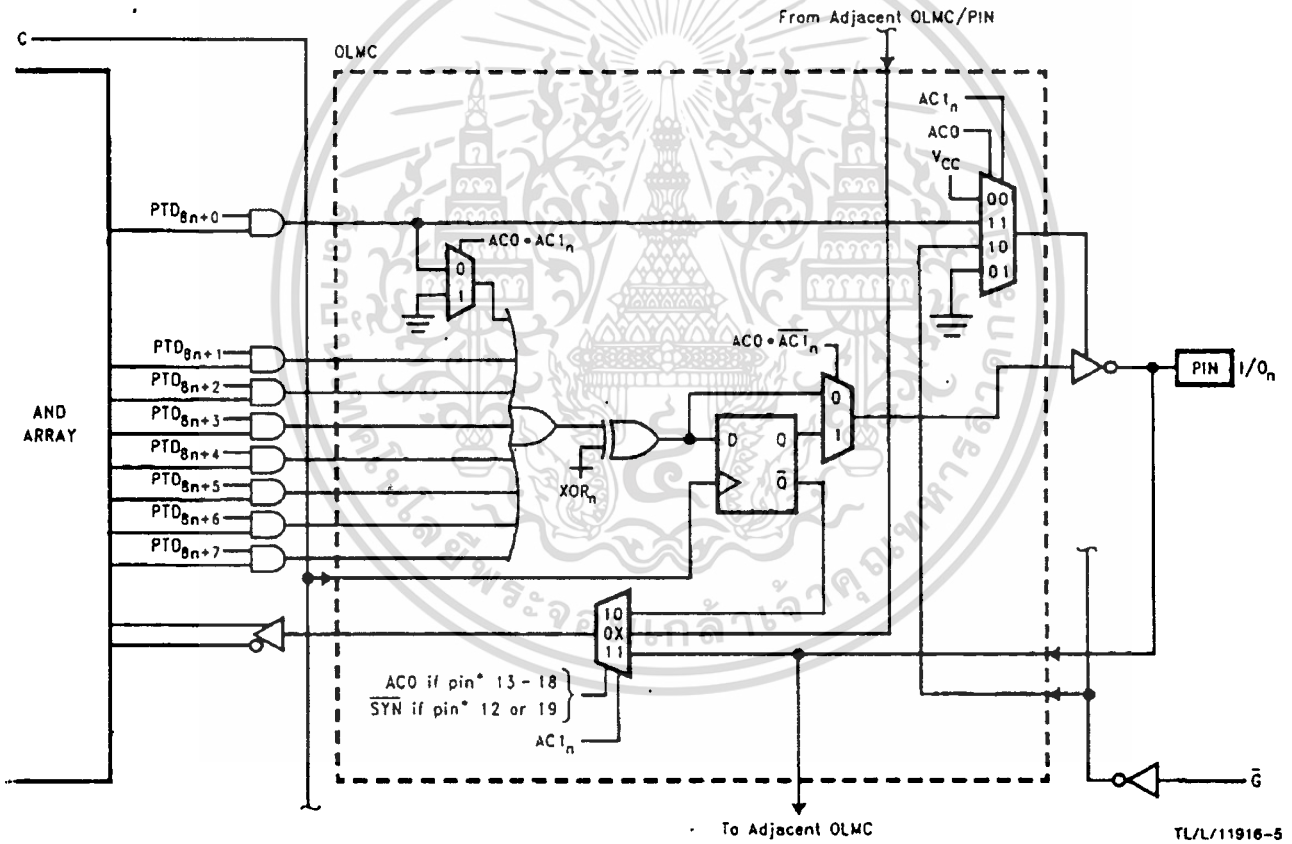
The various configurations of the OLMCs are controlled by a set of programmable "architecture" cells, separate from the logic-defining array cells. Each GAL device contains two "global" architecture cells, "SYN" and "AC0", which affect all OLMCs. Each of the device's eight OLMCs also contains two "local" cells, "AC1" and "XOR". The OLMC Logic Diagram in *Figure 4* shows how the architecture cells select the different paths through the OLMC.

The SYN bit controls whether a device will have any registered outputs (SYN = 0) or will be purely combinatorial (SYN = 1). The SYN bit determines whether device pins* 1 and 11 are used as the clock and global TRI-STATE control inputs (SYN = 0) or whether they are ordinary inputs (SYN = 1). The AC0 bit selects between the "Small-PAL" mode and the "Medium/Registered-PAL" modes. The function of the AC1 bits depend on the state of the AC0 bit. In "Small-PAL" mode (AC0 = 0), the AC1 bit in each OLMC determines whether the associated device pin is an output (AC1 = 0) or an input (AC1 = 1). In "Registered-PAL" mode (AC0 = 1), the AC1 bit determines whether each OLMC is registered (AC1 = 0) or combinatorial (AC1 = 1). In "Medium-PAL" mode (AC0 = 1), the AC1 bits in all OLMCs must be set to 1 (combinatorial). All of the valid architecture bit configurations are shown in the OLMC Architecture table (Table I).

Independent of SYN, AC0 and the AC1 bits, the XOR bit in each OLMC selects between active-low (XOR = 0) or active-high (XOR = 1) output polarity.

*Applies to both 20-pin DIP and 20-lead PLCC packages for GAL16V8QS.

OLMC Logic Diagram



*Applies to 20-pin DIP, 20-pin SOIC and 20-lead PLCC packages for GAL16V8QS.

FIGURE 4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CAPS XPert Version 1.2 This Material Copyrighted By Its Respective Manufacturer.

Input Schematic

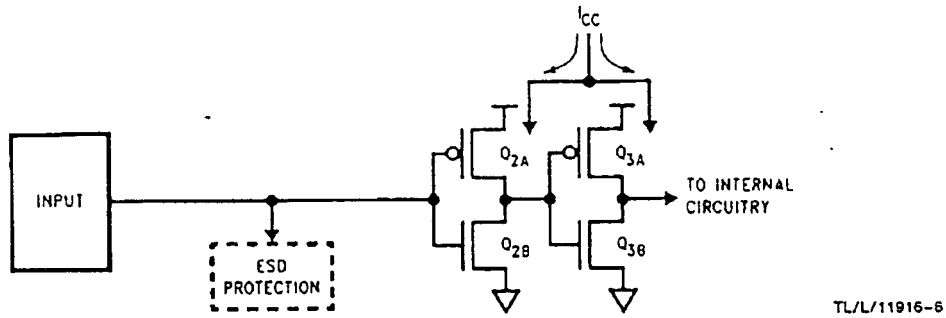


TABLE I. OLMC Architecture Configuration

Pin Number	"Small PAL" Mode		"Registered PAL" Mode		"Medium PAL" Mode
	Function		Function		Function
1	INPUT	INPUT	CLOCK	CLOCK	INPUT
19***	I/O	INPUT	REGISTER	I/O	TRI-STATE**
18***	I/O	INPUT	REGISTER	I/O	I/O
17***	I/O	INPUT	REGISTER	I/O	I/O
16***	OUTPUT*	NC	REGISTER	I/O	I/O
15***	OUTPUT*	NC	REGISTER	I/O	I/O
14***	I/O	INPUT	REGISTER	I/O	I/O
13***	I/O	INPUT	REGISTER	I/O	I/O
12***	I/O	INPUT	REGISTER	I/O	TRI-STATE**
11	INPUT	INPUT	$\overline{G}$	$\overline{G}$	INPUT
Architecture Bits Configuration	$AC1_n = 0$	$AC1_n = 1$	$AC1_n = 0$	$AC1_n = 1$	$AC1_n = 1$
	SYN = 1, AC0 = 0		SYN = 0, AC0 = 0		SYN = 1, AC0 = 0
	All outputs are combinatorial and always active		At least one output is registered		All I/O pins are combinatorial

Note: Pin numbers above apply to 20-pin DIP, 20-pin SOIC and 20-lead PLCC packages for GAL16V8.

* Active combinatorial output

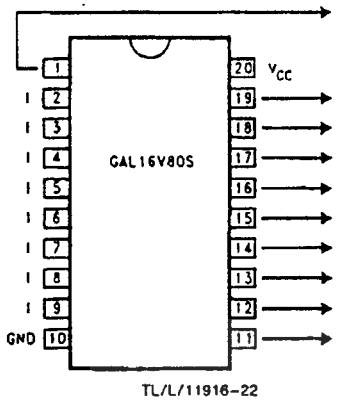
** TRI-STATE combinatorial output

*** $AC1_n$ applies to these I/O pins only

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

d From CAPS XPert Version 1.2 This Material Copyrighted By Its Respective Manufacturer.

PAL Replacement Configurations



TL/L/11916–22

TABLE II. OLMC Architecture Configuration

	"Small PAL" Mode				"Registered-PAL" Mode			"Medium-PAL" Mode
	INPUT	INPUT	INPUT	INPUT	CLOCK	CLOCK	CLOCK	INPUT
	OUTPUT*	INPUT	INPUT	INPUT	REGISTER	I/O	I/O	TRI-STATE**
	OUTPUT*	OUTPUT*	INPUT	INPUT	REGISTER	REGISTER	I/O	I/O
	OUTPUT*	OUTPUT*	OUTPUT*	INPUT	REGISTER	REGISTER	REGISTER	I/O
	OUTPUT*	OUTPUT*	OUTPUT*	OUTPUT*	REGISTER	REGISTER	REGISTER	I/O
	OUTPUT*	OUTPUT*	OUTPUT*	OUTPUT*	REGISTER	REGISTER	REGISTER	I/O
	OUTPUT*	OUTPUT*	OUTPUT*	INPUT	REGISTER	REGISTER	REGISTER	I/O
	OUTPUT*	OUTPUT*	INPUT	INPUT	REGISTER	REGISTER	I/O	I/O
	OUTPUT*	INPUT	INPUT	INPUT	REGISTER	I/O	I/O	TRI-STATE**
	INPUT	INPUT	INPUT	INPUT	$\overline{G}$	$\overline{G}$	$\overline{G}$	INPUT
	10L8	12L6	14L4	16L2	16R8	16R6	16R4	16L8
EMULATED	10H8	12H6	14H4	16H2	16RP8	16RP6	16RP4	16H8
PAL PRODUCTS	10P8	12P6	14P4	16P2				16P8

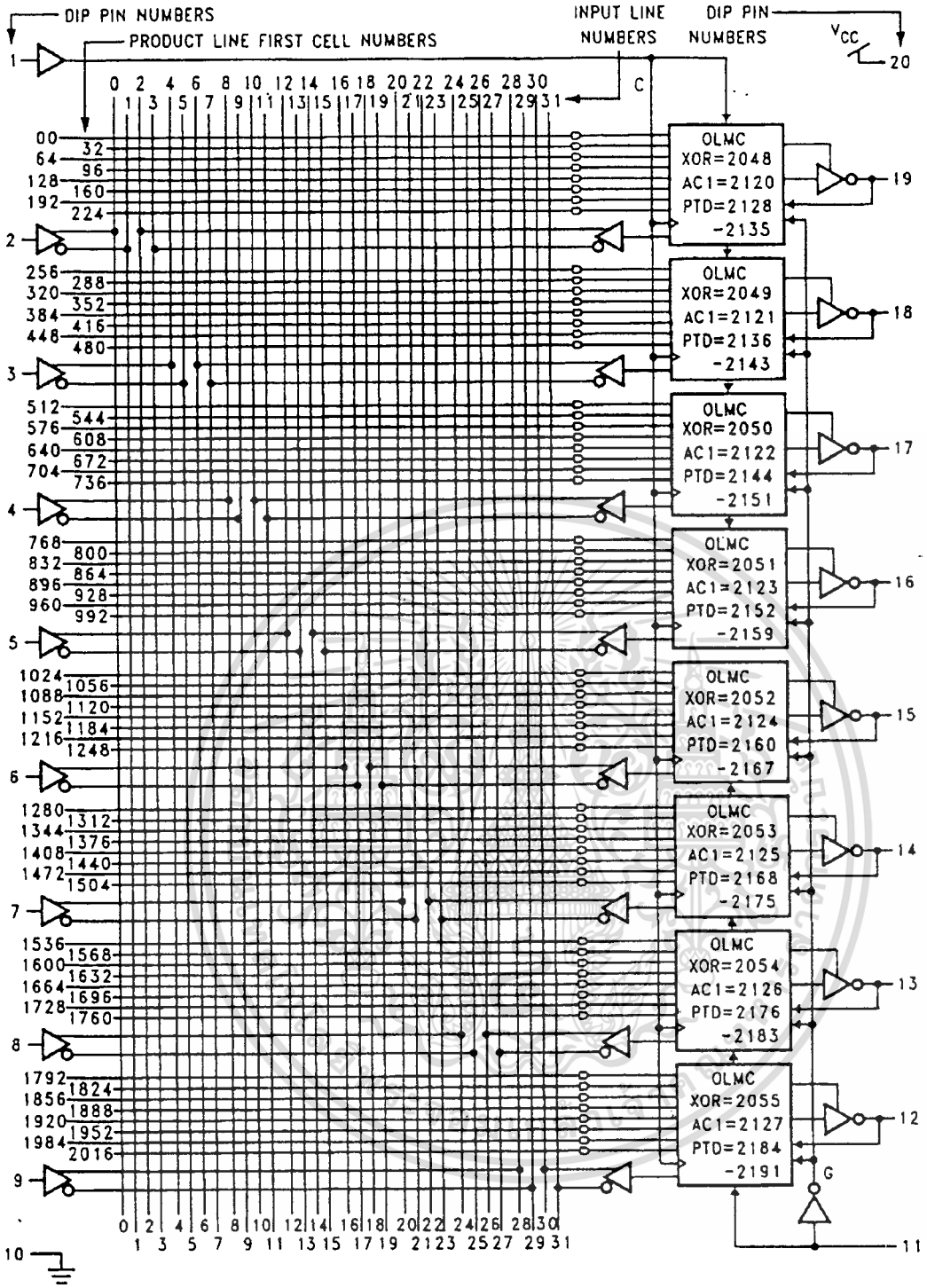
*Active combinatorial output.

**TRI-STATE combinatorial output.

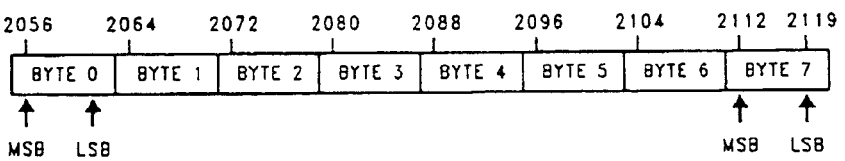
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CAPS XPert Version 1.2 This Material Copyrighted By Its Respective Manufacturer.

L16V8QS Logic Diagram



USER ELECTRONIC SIGNATURE WORD:



SYN=2192
ACO=2193

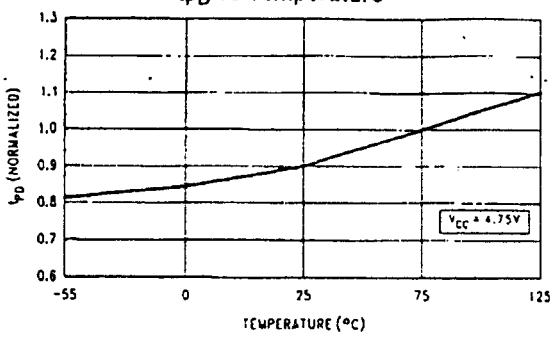
TL/L/11916-14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PS XPert Version 1.2 This Material Copyrighted By Its Respective Manufacturer.

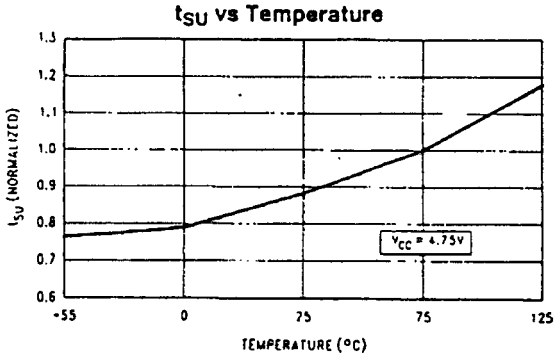
Typical Performance Characteristics

t_{PD} vs Temperature



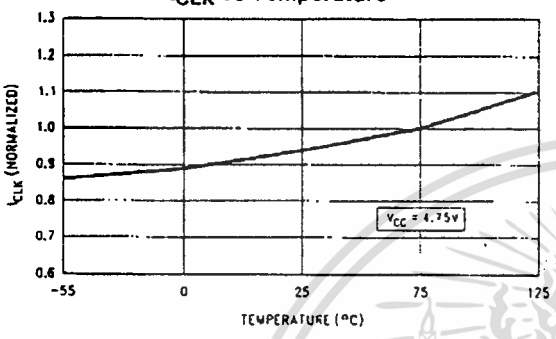
TL/L/11916-13

t_{SU} vs Temperature



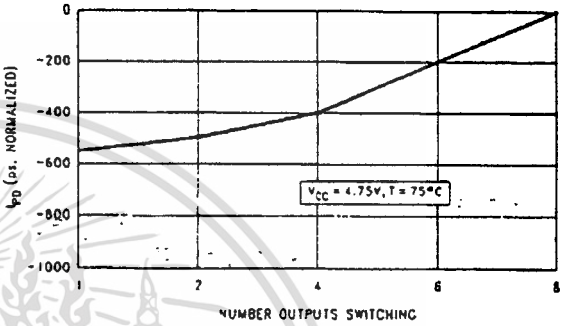
TL/L/11916-24

t_{CLK} vs Temperature



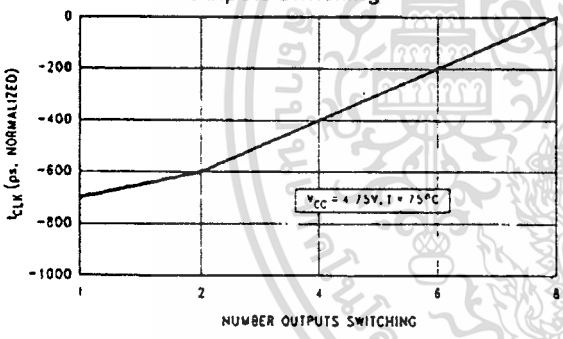
TL/L/11916-25

t_{PD} vs Number of Outputs Switching



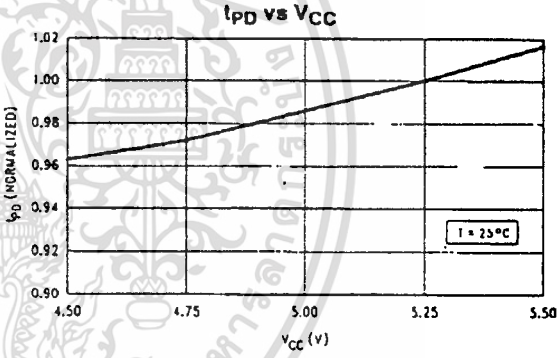
TL/L/11916-26

t_{CLK} vs Number of Outputs Switching



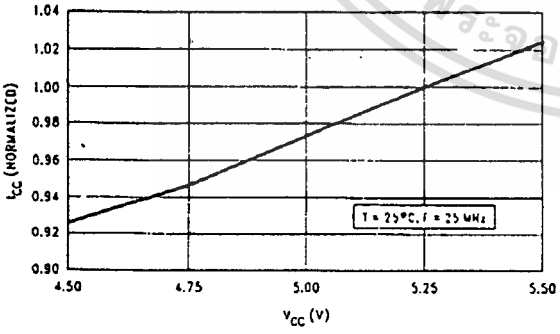
TL/L/11916-27

t_{PD} vs V_{CC}



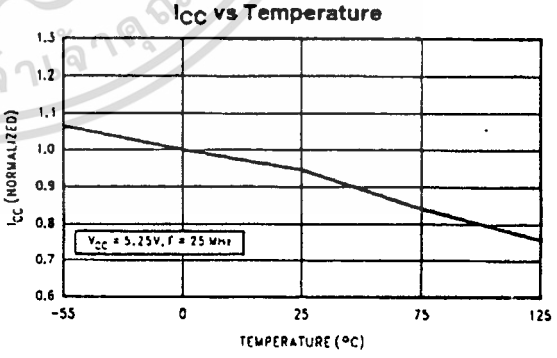
TL/L/11916-28

I_{CC} vs V_{CC}



TL/L/11916-29

I_{CC} vs Temperature



TL/L/11916-30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

om CAPS XPert Version 1.2 This Material Copyrighted By Its Respective Manufacturer.

Ordering Information†

Commercial Devices

t_{PD} (ns)	t_{SU} (ns)	t_{CLK} (ns)	I_{CC} (mA)	QS	Part Number	Package
10	5	7.5	115	Y	GAL16V8QS-10LNC	PDIP
10	5	7.5	115	Y	GAL16V8QS-10LVC	PLCC
10	5	7.5	115	Y	GAL16V8QS-10LMC	SOIC
15	7	9	90	Y	GAL16V8QS-15LNC	PDIP
15	7	9	90	Y	GAL16V8QS-15LVC	PLCC
15	7	9	90	Y	GAL16V8QS-15LMC	SOIC

Industrial Devices

t_{PD} (ns)	t_{SU} (ns)	t_{CLK} (ns)	I_{CC} (mA)	QS	Part Number	Package
10	5	7.5	130	Y	GAL16V8QS-10LNI	PDIP
10	5	7.5	130	Y	GAL16V8QS-10LVI	PLCC
10	5	7.5	130	Y	GAL16V8QS-10LMI	SOIC
15	7	9	130	Y	GAL16V8QS-15LNI	PDIP
15	7	9	130	Y	GAL16V8QS-15LVI	PLCC
15	7	9	130	Y	GAL16V8QS-15LMI	SOIC

Military Devices

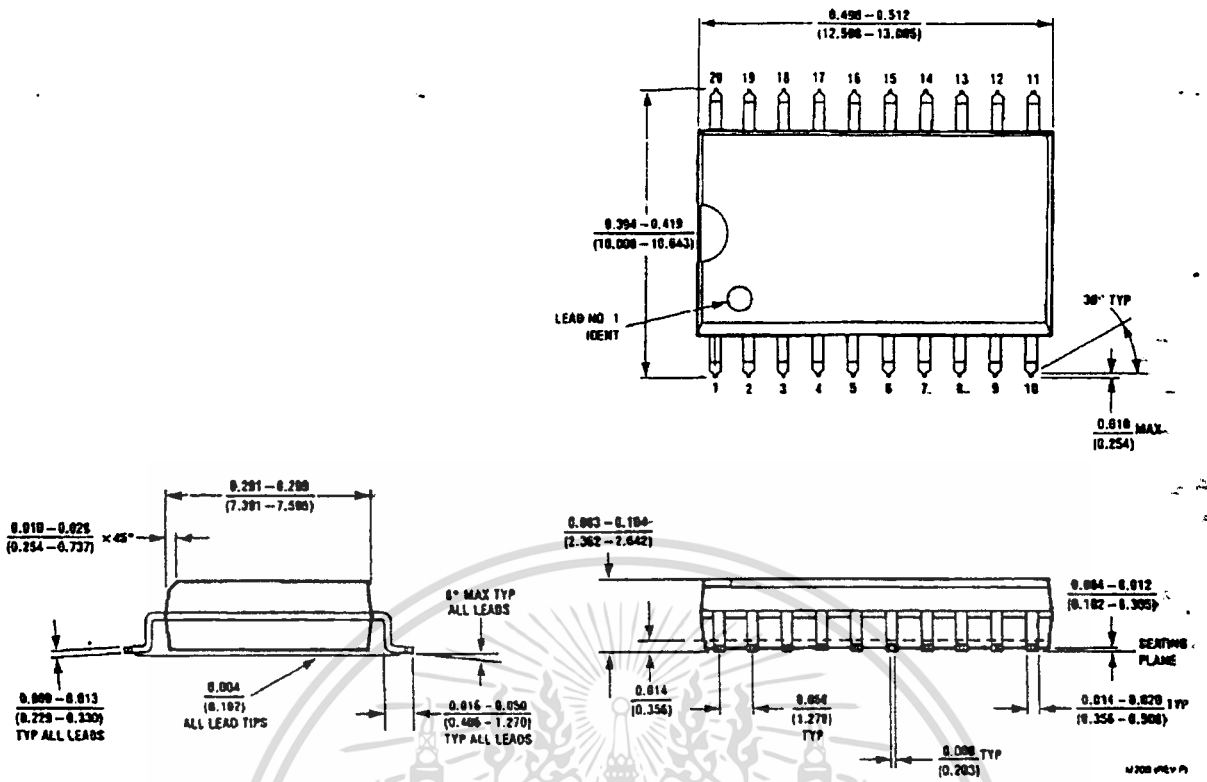
Contact your local National Semiconductor sales representative for availability of military grade GAL16V8QS devices.

†**Quiet Series devices** (GAL16V8QS) recommended for new designs. Refer to 1993 Programmable Logic Devices Databook and Design Guide (Lit # 400081) for quarter power GAL16V8A specifications.

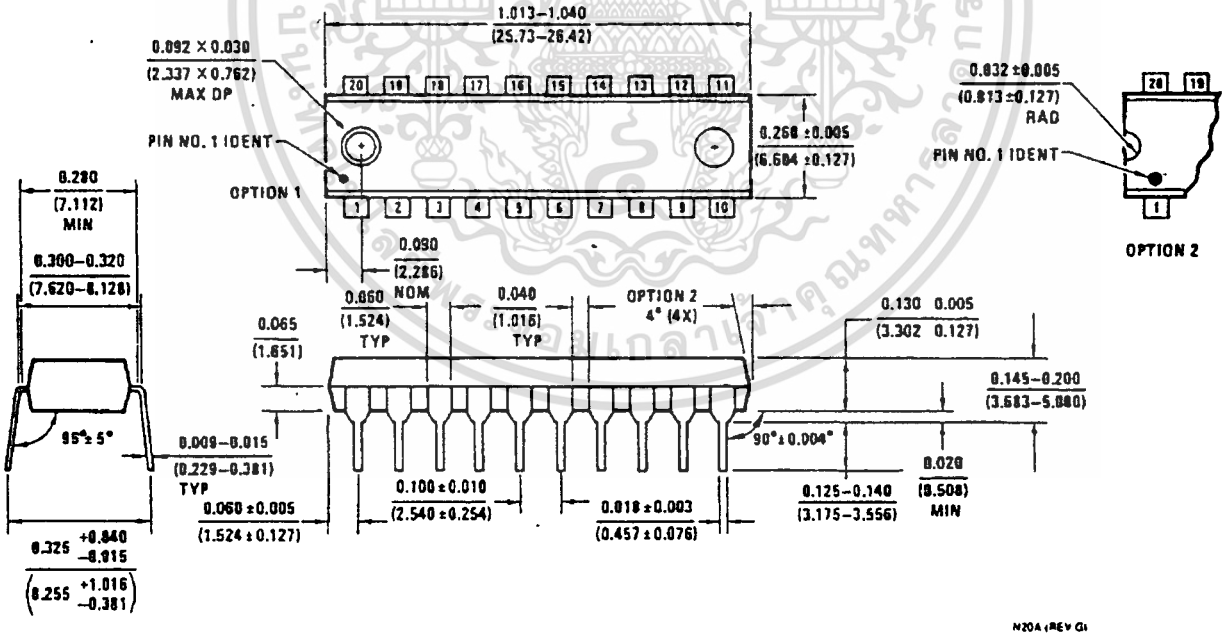
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ed From CAPS XPert Version 1.2 This Material Copyrighted By Its Respective Manufacturer.

Physical Dimensions inches (millimeters)



20-Lead (0.300" Wide) Molded Small Outline Package, JEDEC
NS Package Number M20B



20-Lead (0.300" Wide) Molded Dual-In-Line Package
NS Package Number N20A

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

n CAPS XPert Version 1;2 This Material Copyrighted By Its Respective Manufacturer.

National Semiconductor

November 1993

GAL16V8QS-10L, -15L
20-Pin 0.8μ EECMOS PLDs

General Description

The EECMOS GAL16V8QS devices are fabricated using National's CS80BEV 0.8μ Electrically Erasable CMOS process. This advanced process makes National's GAL16V8QS extremely fast, allowing controlled output edge rates which dramatically reduce noise. Low noise is actually specified and guaranteed with National's GAL16V8QS Quiet Series™ devices.

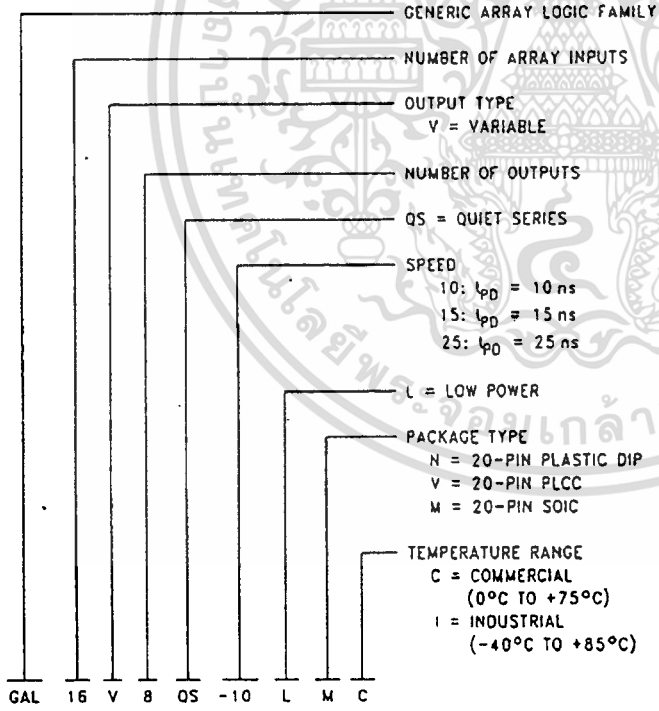
National's fast programming algorithm allows the GAL16V8QS to be programmed significantly faster than similar devices using industry standard programmers. Fast programming reduces the cost of programming by greatly increasing programming throughput. National guarantees a minimum of 100 erase/write cycles.

Unique test circuitry and reprogrammable cells allow complete AC, DC, cell, and functionality testing during manufacture. Therefore, National guarantees 100% field programmability and functionality of GAL® devices. In addition, a security circuit is built-in, providing proprietary designs with copy protection.

Features

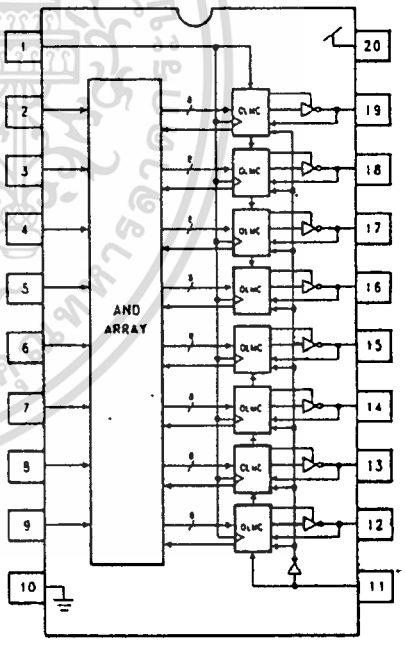
- High performance 0.8 μ EECMOS technology
  - 10 ns maximum propagation delay
  - 5 ns setup time delay
  - 7.5 ns clock to registered output delay
  - f_{MAX} = 80 MHz
  - Reduced ground bounce
  - 2000V ESD protection
- Reduced power
  - I_{CC} max = 90 mA @ 25 MHz
- Electrically erasable cell technology
  - 100% tested at manufacture
- Fast programming algorithm
  - Reduces programming cost, increases throughput
- Emulates popular PAL® devices
- Fully supported by National's OPAL™ and OPAL jr software as well as 3rd-party PLD development software
- Commercial and industrial grades

Ordering Information



TL/L/11916-23

Block Diagram



TL/L/11916-1

TRI-STATE® is a registered trademark of National Semiconductor Corporation.
OPAL™ and Quiet Series™ are trademarks of National Semiconductor Corporation.
GAL® is a registered trademark of Lattice Semiconductor.
PAL® is a registered trademark of and used under license from Advanced Micro Devices, Inc.

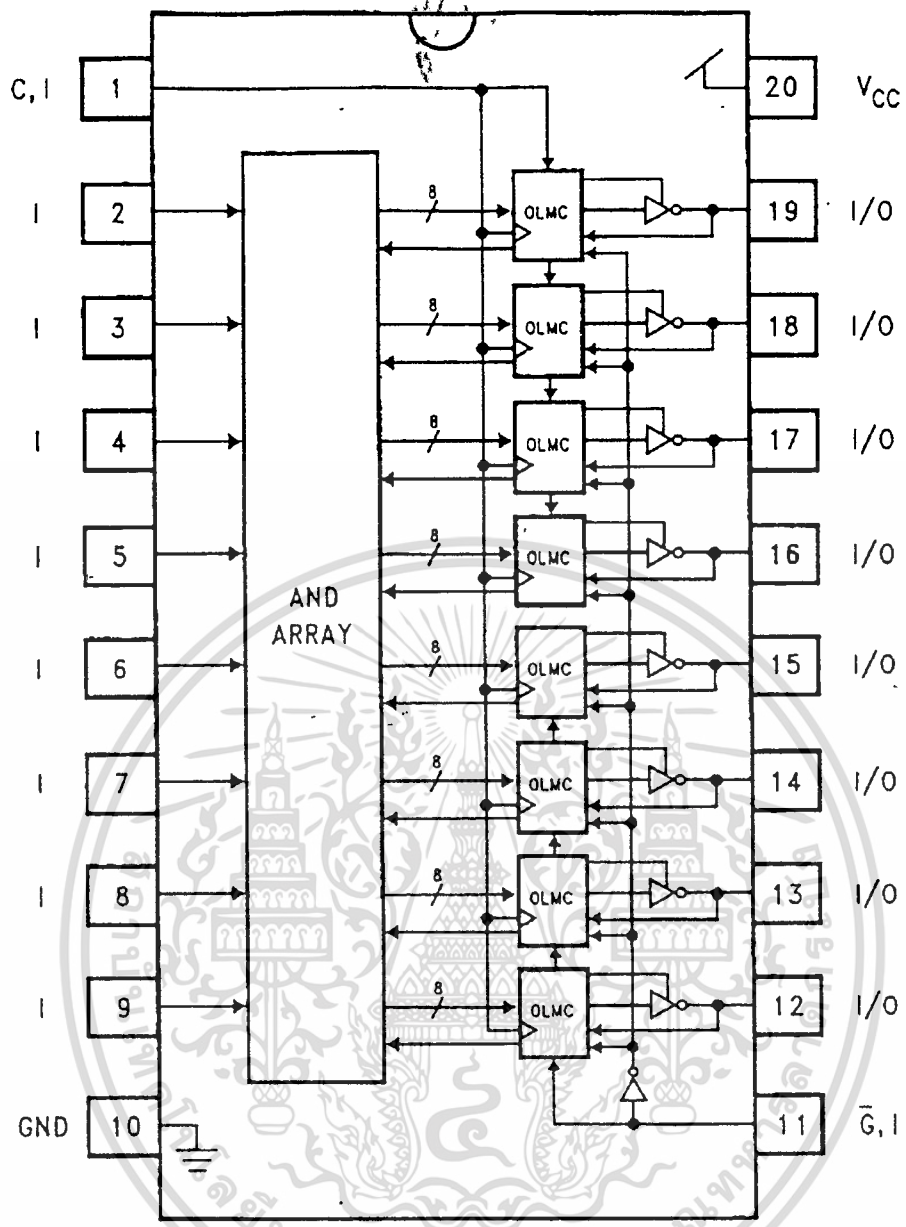
©1993 National Semiconductor Corporation

RRD-B20M113/Printed in U. S. A.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

d From CAPS XPert Version 1.2 This Material Copyrighted By Its Respective Manufacturer.

GAL16V8QS Block Diagram/DIP and SOIC Connections



TL/L/11916-7

FIGURE 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

om CAPS XPert Version 1.2 This Material Copyrighted By Its Respective Manufacturer.