ปีการศึกษา 2533

ปริญญานิพนธ์

เรื่อง

เครื่องโปรแกรม PAL

(PAL PROGRAMMABLE CARD)

โดย

นาย ยศพงศ์ เตชะพรสิน 31.3418

อาจารย์ที่ปรึกษา

อาจารย์ อุทัย ศรีธีระวิโรจน์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

028740

12 มี.ค. 2534

ปริญญานิพนธ์ปีการศึกษา 2533

ภาควิชาเทคนิคอุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องโปรแกรม PAL (PAL PROGRAMMABLE CARD)

โดย

1. นายยศพงศ์ เตชะพรสิน 31.3418

.....อาจารย์ที่ปรึกษา

(อาจารย์ อุทัย ศรีธีระวิโรจน์)

.....อาจารย์ที่ปรึกษา

.....กรรมการ

.....กรรมการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องโปรแกรม PAL

ยศพงศ์ เตชะพรสิน

อาจารย์ที่ปรึกษา

อ.อุทัย ศรีธีระวิโรจน์

บทคัดย่อ

ปัจจุบัน PAL เริ่มมีบทบาทในการออกแบบวงจรอีเล็คทรอนิสค์มากขึ้น เนื่องจากใน PAL สามารถออกแบบให้ทำงานได้ตามต้องการโดยยุบ Gate ที่ใช้ในวงจรลงใน PAL ซึ่งมีราคาไม่สูงนัก และสามารถหาได้ง่าย การโปรแกรม PAL ที่ออกแบบจะถูกโปรแกรมด้วยเครื่องมือที่ใช้โปรแกรมเฉพาะซึ่งมีราคาแพง และด้วยเหตุนี้เองที่เป็นแรงจูงใจในการทำโครงการนี้ขึ้น เพื่อศึกษาถึงโครงสร้างภายในรวมทั้งกรรมวิธีการโปรแกรมที่ซับซ้อนของ PAL เพื่อที่จะทำการอ่าน การเขียนตลอดจนการรักษาความลับของข้อมูลที่สำคัญ

โครงการนี้เหมาะกับผู้ที่ต้องการออกแบบวงจรโดยใช้ PAL สามารถที่จะทำการโปรแกรมได้เองซึ่งสามารถเพิ่มขอบเขตของ PAL ให้ใช้งานได้กว้างขวางยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PAL PROGRAMMABLE CARD

YOSAPONG TACHAPORNSIN

ABSTACT

Nowadays, PAL become a great role for design electronic circuits because it's was generally operated, inexpansive and facilitate. PAL able to operate by setting data with specify card through to PAL chip but the card is expensive. It's the reason of this project for study internal structure and complex operating of PAL to read, write, investigate and secure important data for create PAL Programmable card.

This project appropriate for the person who want PAL in his circuit. It's substitute expensive card and provide by himself, thus PAL will increase of application and enhance useful.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
บทที่ 1	บทนำ	1
บทที่ 2	หลักการและทฤษฎีการทำงานของ PAL	5
2.1	ประเภทของ PROGRAMMABLE LOGIC DEVICE (PLD)	5
2.2	คุณสมบัติของ PAL	8
2.3	สัญญลักษณ์ทาง LOGIC ภายใน PAL	9
2.4	วิธีการดูเบอร์ของ PAL	11
2.5	วงจรภายใน PAL ชนิดต่าง ๆ	12
2.6	กรรมวิธีการโปรแกรม	15
บทที่ 3	การออกแบบและการสร้าง PAL PROGRAMMABLE CARD	20
3.1	การ INTERFACE กับ IBM PC	20
3.2	BUS RECIEVER AND ADDRESS DECODER	31
3.3	การใช้งาน MC 6821 (PIA)	32
3.4	VOLTAGE DRIVER FOR INPUT PIN	34
3.5	VOLTAGE DRIVER FOR OUTPUT PIN	35
3.6	MONOSTABLE MULTIVIBRATOR	37
3.7	SWITCHING POWER SUPPLY	38
3.8	PROGRAMMABLE VOLTAGE POWER SUPPLY	39
3.9	SOFTWARE	43
บทที่ 4	การทดลองและผลการทดลอง	78
บทที่ 5	สรุปและวิจารณ์	82
ภาคผนวก		
กิตติกรรมประกาศ		
หนังสืออ้างอิง		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

นับตั้งแต่ไอซีประเภทดิจิตอลได้ถูกผลิตขึ้นมาเมื่อประมาณ 20 กว่าปีแล้ว ถือได้ว่าเป็นความสำเร็จขั้นสุดยอดที่ทำให้วงการอิเล็กทรอนิกส์ทั่วโลกในยุคนั้น เริ่มเปลี่ยนโฉมเข้าสู่ยุคการพัฒนาอย่างรวดเร็ว ประกอบกับเทคโนโลยีการผลิตที่ช่วยให้ขนาดของตัวไอซีเล็กลงจนสามารถนำไอซีนับร้อยนับพันตัวมารวมอยู่บน "ชิพ" เดียวกันได้ และเราก็ได้รู้จักดีในชื่อของ "ไมโครโปรเซสเซอร์" หรือวงจรอิเล็กทรอนิกส์ที่ทำงานด้วยโปรแกรม

แต่ถึงจะก้าวล้ำนำหน้าไปขนาดไหนก็ตาม ไอซีดิจิตอลในยุคต้นๆ ก็ยังคงเป็นที่นิยมกันอยู่ในปัจจุบัน ด้วยเหตุผลที่ว่า โครงสร้างที่ไม่ซับซ้อนและเข้าใจการทำงานได้ง่ายสามารถนำมาต่อร่วมกันหลายๆ ตัวเพื่อให้ทำงานตามระบบที่ออกแบบไว้โดยไม่มีขีดจำกัด จึงช่วยให้นักอิเล็กทรอนิกส์สมัครเล่นหรือผู้ผลิตรายย่อยๆ มีโอกาสสร้างสิ่งประดิษฐ์หรือสินค้าออกมาใช้งานหรือจำหน่ายได้

โครงการที่จะกล่าวถึงกันนี้ เป็นการแนะนำไอซีดิจิตอลชนิดใหม่ที่จะทำให้ความยุ่งยากซับซ้อนของการต่อไอซีดิจิตอลหลายๆ ตัว ถูกรวบหัวรวบหางมาไว้อยู่บนไอซีชนิดใหม่นี้เพียงตัวเดียว โดยที่ยังมีคุณสมบัติการทำงานเหมือนเดิม ถึงแม้ว่าวงจรดิจิตอลจะเป็นแบบไหนก็ตาม

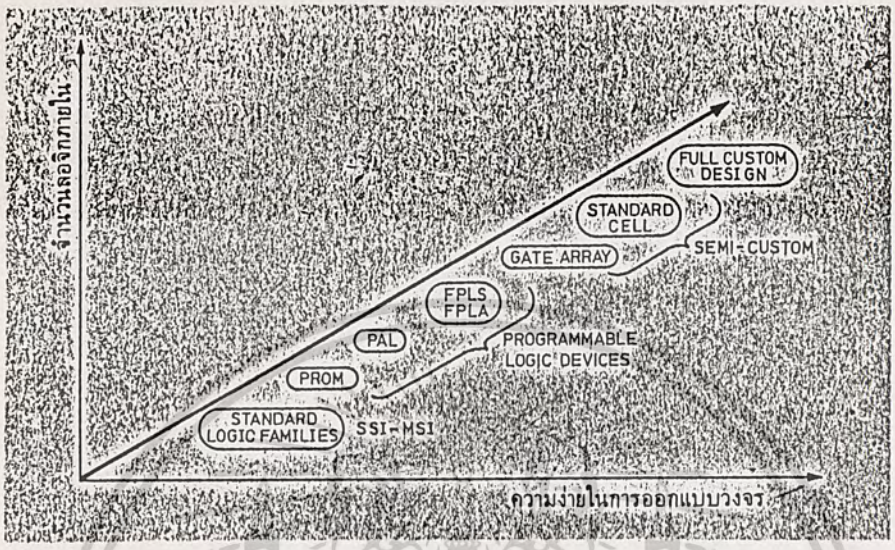
ใช่แล้ว เรากำลังหมายถึงไอซีที่สามารถโปรแกรมรูปแบบและลักษณะการทำงานทางลอจิกของมันได้ตามความต้องการ

ในการที่จะเลือกใช้ไอซีลอจิกต่างๆในปัจจุบัน มีหนทางเลือกอยู่มากมาย ดังแสดงประเภทของไอซีลอจิกไว้ในรูปที่ 1 แสดงให้เห็นว่า ยิ่งจำนวนลอจิกเกตภายในตัวไอซีมากขึ้นเท่าใด ก็จะทำให้การออกแบบวงจรทำได้ง่ายขึ้นตามไปด้วย

ในยุคเริ่มต้นของไอซีที่ผลิตออกมา ในลักษณะการค้าประเภท SSI (Small Scale Integrated) เช่น ไอซี TTL ตระกูล 7400 หรือ CMOS ตระกูล 4000 ซึ่งเป็นลอจิกเกตง่ายๆ ต่อมาเทคโนโลยีการผลิตสูงขึ้นสามารถเพิ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำนวนลอจิกเกตลงในชิพได้มากขึ้น เลยผลิตออกมาเป็น MSI (Medium Scale Integrated) เช่น เคาน์เตอร์หรือไอซีทำหน้าที่คำนวณ ซึ่งไอซีทั้ง 2 ชนิดนี้มีราคาถูกมากและนักอิเล็กทรอนิกส์ทั่วไปมีโอกาสได้ใช้มาเป็นเวลานานแล้ว



รูปที่ 1 แนวทางการเลือกไอซีออกแบบวงจร

สำหรับประเภท Full Custom เป็นไอซีประเภทที่ผลิตตามความประสงค์ของลูกค้าโดยเฉพาะซึ่งเป็นไอซีเฉพาะงาน ไม่สามารถนำไปประยุกต์ให้กับงานทั่วไปได้ จะมีก็แต่บริษัทยักษ์ใหญ่ระดับโลกเท่านั้นที่มีโอกาสจะนำไปใช้กับงานของตนเองได้ และการออกแบบชิพประเภทนี้ต้องใช้เครื่องมือที่ทันสมัย อีกทั้งราคาจะแพงจนไม่กล้าซื้อและการสั่งทำจากโรงงานต้องสั่งครั้งละจำนวนมากหรือมีใช้กับงานทางทหารเท่านั้น

ไอซีดิจิตอลประเภท Semi-Custom แบ่งออกได้เป็น 2 แบบ ตามการออกแบบและผลิต แบบแรกเป็นการออกแบบได้เองด้วยคอมพิวเตอร์ประเภท work station ซึ่งเป็นเครื่องมืออันทันสมัยมีใช้เฉพาะในประเทศที่มีเทคโนโลยีก้าวหน้าเท่านั้น โดยผู้ออกแบบสามารถเรียกไอซีหรือเกตชนิดที่ถูกแบบมาก่อนแล้ว แล้วนำมาต่อรวมกันบนจอคอมพิวเตอร์ เสร็จแล้วก็ส่งให้โรงงานผลิตออกมาเป็นชิพสำเร็จรูป แบบนี้ไม่จำเป็นต้องสั่งทีละมากๆ ก็ได้ เพราะขบวนการผลิตทำได้ง่ายกว่าและต้นทุนไม่สูงมากนัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบที่สองของประเภท Semi-Custom ทางโรงงานผู้ผลิตจะผลิตชิพออกมาในลักษณะกึ่งสำเร็จที่เรียกว่า ULA โดยบนชิพจะมีเกตต่างๆ จัดเรียงกันมากมาย ผู้ออกแบบเพียงแต่ระบุหน้าที่ของเกตต่างๆ และการต่อเกตเข้าด้วยกัน หลังจากนั้นโรงงานที่ผลิตจะรวบรวมการสั่งผลิตจากลูกค้าหลายๆ รายแล้วผลิตพร้อมกันทีเดียวด้วยขั้นตอนสุดท้ายอีกขั้นตอนเดียว

PLD (Programmable Logic Devices) คือ ไอซีที่เรามีโอกาสจะได้สัมผัสได้ ซึ่งแบ่งออกได้เป็น PROM, PAL, PLA, EPLS ซึ่งปัจจุบันได้มีการผลิตออกมาจำหน่ายกันอย่างแพร่หลาย และผู้ใช้สามารถนำไปโปรแกรมได้เองด้วยขบวนการทางไฟฟ้าซึ่งยุ่งยาก ผู้ใช้สามารถออกแบบและกำหนดเลือกชนิดของเกตและการต่อเกตได้ตามความต้องการแต่ก็มีขีดจำกัดที่จำนวนเกตและความซับซ้อนของการต่อวงจร

ข้อดีข้อเสียของ PLD

ข้อดีของการใช้ PLD แทนไอซีดิจิตอลประเภทเกตธรรมดามีอยู่หลายข้อด้วยกัน ยกเว้นในเรื่องของราคาและต้นทุนการผลิตเนื่องจากว่าราคาของมันยังสูงอยู่ เผลอๆ อาจจะสูงกว่าการใช้ไอซีธรรมดาหลายๆ ตัวมาต่อกันด้วยซ้ำไป

อย่างไรก็ตาม ด้วยต้นทุนการออกแบบที่ลดลงซึ่งสามารถลดเวลาและขั้นตอนการออกแบบได้มาก รวมกับขนาดของแผ่นวงจรพิมพ์ที่เล็กลง รวมไปถึงขั้นตอนการผลิตที่จะลดทุนลงได้ จึงทำให้ PLD มีโอกาสที่จะเข้ามาแทนไอซีธรรมดาได้

ในเรื่องของคุณภาพ PLD จะได้เปรียบกว่ามากในแง่ของความเชื่อถือได้ของวงจร เพราะ จำนวนไอซีและจุดบัดกรีจะน้อยลงเป็นการลดปัญหาที่จะเกิดขึ้นภายหลัง และการสิ้นเปลืองกระแสไฟฟ้าก็จะน้อยกว่าด้วย ทำให้ระบบระบายความร้อนมีขนาดเล็กลงตามไปด้วย

ข้อดีอีกประการหนึ่ง ซึ่งอาจจะมองไม่เห็นได้ในขณะนี้ ก็คือจำนวนสต็อคหรือเบอร์ไอซีสำหรับร้านค้าจะลดน้อยลงไปด้วย เพียง PLD แค่ 20-30 เบอร์ สามารถที่จะแทนหน้าที่แบบเดียวกันกับไอซี SSI หรือ LSI ได้ถึงหลายๆร้อยเบอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในอนาคตร้านค้าอาจใช้วิธีโปรแกรมหน้าที่การทำงาน PLD ให้ลูกค้าได้เลยโดยเพียงแต่เสียบแผ่นดิสค์เข้ากับเครื่องคอมพิวเตอร์แล้วกดคีย์เลือกเท่านั้น

ในเรื่องของการรักษาความลับหรือป้องกันการก๊อปปี้ ก็เป็นส่วนดีประการหนึ่งสำหรับผู้ออกแบบและผู้ผลิต เพราะ ไม่สามารถตรวจหาข้อมูลที่โปรแกรมเอาไว้ได้ถ้าขาดรายละเอียดในการออกแบบ การก๊อบปี้วงจรกันจึงไม่สามารถทำได้ง่าย ยังมี PAL บางเบอร์ที่ไม่สามารถตรวจหาข้อมูลได้เลยหลังจากที่โปรแกรมลงไปแล้ว

แม้ว่าการรักษาความลับจะเป็นข้อดีแต่ก็มีข้อเสียแอบแฝงอยู่ สำหรับช่างที่ทำการตรวจซ่อมไม่อาจตรวจสอบการทำงานได้ด้วยลอจิกโพรบหรือเครื่องลอจิกอะนาไลเซอร์ร่วมกับข้อมูลของไอซีเบอร์นั้นๆ เหมือนกับไอซีดิจิตอลทั่วๆ ไป เมื่อวงจรเกิดปัญหาขึ้นมาและไม่ทราบหน้าที่ของ PLD แล้ว จึงเป็นสิ่งลึกลับซ่อนเร้นที่น่ากลัวทีเดียว

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

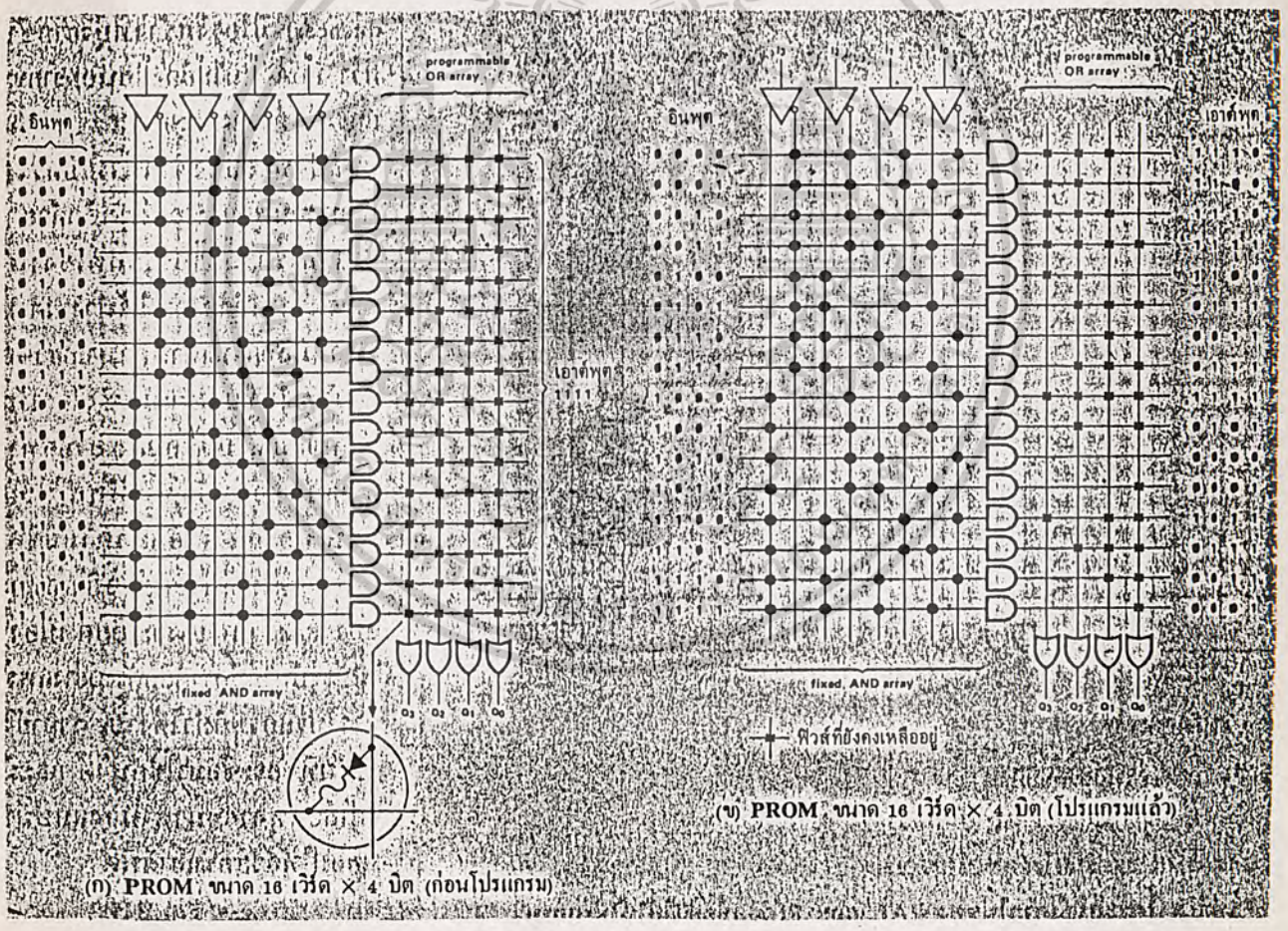
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

หลักการและทฤษฎีการทำงานของ PAL

2.1 ประเภทของ PROGRAMMABLE LOGIC DEVICE (PLD)

จุดเริ่มต้นของ PLDเริ่มขึ้นด้วย PROM ก่อนแล้วจึงมีบริษัท Monolithic Memories ซึ่งเป็นผู้คิดค้น PAL (Programmable Array Logic) และเรียกชื่อว่า PAL ในทางการค้า อีกบริษัทหนึ่งคือ Signetics/Mullard ซึ่งเป็นผู้คิดค้น FPLA (Field Programmable Logic Arrays) หรือเรียกกันสั้นๆ ว่า PLA ทั้งหมดนี้ไม่ใช่ของใหม่โดยเฉพาะอย่างยิ่ง PROM เราก็ได้รู้จักมันเป็นอย่างดีแล้ว



รูปที่ 2 โครงสร้างภายในของ PROM ขนาด 4 อินพุต 4 เอาท์พุต

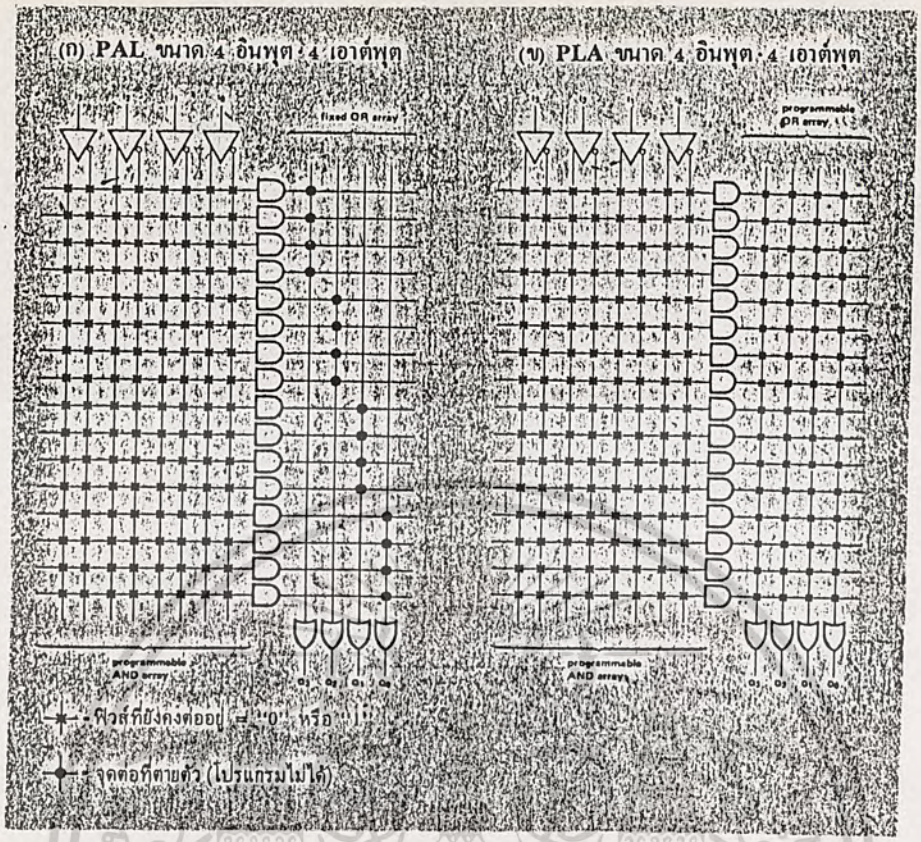
เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ ห้ามมิให้ผู้ใดนำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่โดยหลักของ PAL, PLA, PLS และ FROM เหมือนกัน แต่โครงสร้างภายในเท่านั้นที่แตกต่างกัน พิจารณากันที่ PROM ก่อนในรูปที่ 2 ซึ่งเรารู้จักมันดีในรูปของตัวถอดรหัส โดยรูปที่ 2 ก. คือ โครงสร้างภายในก่อนถูกโปรแกรม สังเกตว่า PROM นี้เป็นการจัดเรียงลอจิกเกตในลักษณะแมทริกซ์โดยที่อินพุตของ AND จะถูกป้อนด้วยลอจิกคงที่จากสัญญาณอินพุต (I_0 -I_3) ส่วนอินพุตของ OR จะถูกโปรแกรมได้ ตัวอย่างในรูปที่ 2 ข. คือ PROM ที่ถูกโปรแกรมแล้ว สังเกตว่าจำนวนขาอินพุตของ OR แต่ละตัวจะเท่ากับจำนวนขาเอาต์พุตของ AND คือ 16 ขา การเลือกว่าจะให้ต่อกันจะแทนด้วยสัญลักษณ์เครื่องหมายกากบาท

PAL มีลักษณะโครงสร้างภายในตรงกันข้ามกับ PROM คือ อินพุตของ AND เป็นแบบโปรแกรมได้ ส่วนอินพุตของ OR จะคงที่ดังตัวอย่างในรูปที่ 3 ก. ซึ่งแสดงโครงสร้างภายในของ PAL ขนาด 4 อินพุต -4 เอาต์พุต จากตัวอย่างนี้สังเกตว่าสัญญาณอินพุต I_0-I_3 จะส่งผ่านบัฟเฟอร์ และอินเวอร์เตอร์ ออกมาเป็นสัญญาณ 8 เส้น (I_0-I_3, I_0-I_3) ป้อนเข้า AND ทุกตัว ซึ่ง AND ทุกตัวจะต้องมีอินพุตอยู่ 8 ขา โดยที่สามารถโปรแกรมได้ว่าจะต่องเข้ากับ I_0-I_3, I_0-I_3 เส้นใด ส่วนเอาต์พุตของ AND จะถูกจัดแบ่งเป็น 4 ชุด ป้อนเข้า OR อย่างตายตัว คือ เอาต์พุต 4 ตัวบนจะป้อนเข้า OR ตัวแรก ถัดลงมา 4 ตัวจะป้อนเข้า OR ต้วที่สอง เป็นเช่นนี้ลงมาเรื่อยๆ ในกรณีนี้หมายถึง OR จะต้องมีอินพุตอยู่ 4 ขา เท่านั้น

PLA จะมีโครงสร้างที่พิเศษกว่า PROM และ PAL คือ อินพุตของทั้ง AND และ OR สามารถโปรแกรมได้ แสดงว่าโครงสร้างภายในของ PLA จะยุ่งยากซับซ้อนกว่าและมีความสามารถสูงกว่า อีกทั้งยังช่วยให้การคำนวณลอจิกเพื่อกำหนดหน้าที่ของมันทำได้หลายรูปแบบมากกว่าเนื่องจากว่าสามารถโปรแกรมได้ 2 ทาง แต่ก็ใช่ว่าจะดีกว่าไปเสียหมด เพราะ ราคาของมันย่อมแพงกว่ารวมทั้งการทำความเข้าใจก็ทำได้ยากกว่า และเครื่องมือที่ใช้โปรแกรมก็ต้องพิเศษมากกว่า ตัวอย่างโครงสร้างภายในของ PLA แสดงไว้ในรูปที่ 3ข.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 โครงสร้างภายในของ PAL และ PLA เมื่อยังไม่ได้โปรแกรม

PLS (Programmable Logic Sequencer) มีโครงสร้างที่ยุ่งยากซับซ้อนกว่า PLA ขึ้นไปอีก คือ นอกจากจะโปรแกรมได้ทั้ง AND และ OR แล้ว ยังมีฟลิป-ฟลอป ซึ่งทำหน้าที่เป็นรีจิสเตอร์ ทั้งทางอินพุตและเอาต์พุตอีกด้วย ดังนั้น จึงสามารถนำมาประยุกต์ใช้งานได้กับวงจรดิจิตอลที่ต้องการให้ทำงานตามลำดับ เวลาที่มีสัญญาณนาฬิกามาเกี่ยวข้องด้วย เช่นตัวนับขึ้น/ลงแบบซิงโครนัส, ตัวเลื่อนข้อมูล, บัสข้อมูลสองทิศทาง, ตัวควบคุมระบบ, ตัวเข้ารหัสสำหรับตรวจสอบ และอื่นๆ อีกมากไม่มีขีดจำกัด

แต่ในโครงการที่ทำนี้จะกล่าวถึงเฉพาะเรื่องของ PAL กันอย่างละเอียด เนื่องจากว่า ในปัจจุบันได้มีการผลิตกันอย่างแพร่หลาย, การโปรแกรมก็ทำได้ง่าย, ราคายังอยู่ในระดับพอจะซื้อมาใช้กันได้ และมีขนาดให้เลือกหลายขนาด บางแบบมีฟลิปฟลอปให้ด้วยทางเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 คุณสมบัติของ PAL

ตารางที่ 1 จะเป็นรายละเอียดแสดงคุณสมบัติของ PAL เบอร์ต่างๆ ทั้ง 20 PIN และ 24 PIN ที่มีอยู่ในปัจจุบัน

Std (35 ns)	High Speed (25 ns)	Pkg Pin	Description
10H8	10H8A	20	10 input, 8 output AND-OR array
12H6	12H6A	20	12 input, 6 output AND-OR array
14H4	14H4A	20	14 input, 4 output AND-OR array
16H2	16H2A	20	16 input, 2 output AND-OR array
10L8	10L8A	20	10 input, 8 output AND-OR array
12L6	12L6A	20	12 input, 6 output AND-OR array
14L4	14L4A	20	14 input, 4 output AND-OR array
16L2	16L2A	20	16 input, 2 output AND-OR array
16C1	16C1A	20	16 input, 1 output AND-OR/NOR array
16L8	16L8A	20	16 input, 8 output AND-OR-invert array
16R8	16R8A	20	16 input, 8 output AND-OR-register array
16R6	16R6A	20	16 input, 6 output AND-OR-register array
16R4	16R4A	20	16 input, 4 output AND-OR-register array
16X4		20	16 input, 4 output AND-OR-XOR-register array
16A4		20	16 input, 4 output AND-CARRY-OR-XOR register
12L10		24	12 input, 10 output AND-OR invert array
14L8		24	14 input, 8 output AND-OR-invert array
16L6		24	16 input, 6 output AND-OR-invert array
18L4		24	18 input, 4 output AND-OR-invert array
20L2		24	20 input, 2 output AND-OR-invert array
20C1		24	20 input, 1 output AND-OR/NOR array
20L10		24	20 input, 10 output AND-OR-invert array
20X10		24	20 input, 10 output AND-OR-XOR-register array
20X8		24	20 input, 8 output AND-OR-XOR-register array
20X4		24	20 input, 4 output AND-OR-XOR-register array

ตารางที่ 1 รายละเอียดแสดงคุณสมบัติของ PAL เบอร์ต่างๆ

PAL ตัวใหม่ๆ ที่เพิ่งจะผลิตออกมาจากโรงงานหรือยังไม่เคยถูกโปรแกรมมาก่อนเลย ขาสัญญาณทางอินพุตทุกขาจะต้องเข้ากับขาสัญญาณของ AND โดยผ่านบัฟเฟอร์และอินเวอร์เตอร์โดยแทนด้วยเครื่องหมายกากบาท จุดที่เป็นกากบาทก็คือ สัญญาณทางแนวตั้งและแนวนอนจะต่อเข้าด้วยกันโดยผ่านตัวไอโอดและฟิวส์ (ดูรูปที่ 2ก.) หมายถึง การจัดเรียงขาทางอินพุตของ AND จะอยู่ในลักษณะแมทริกซ์โดยมีไดโอดและฟิวส์ต่อคั่นอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

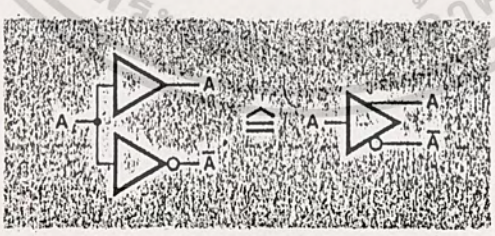
ฟิวส์ที่ต่อคั่นอยู่นี้ สามารถที่จะถูกทำลายทิ้งไปได้โดยการป้อนแรงดันสูงๆ ชั่วขณะเพื่อให้กระแสไหลผ่านตัวมันสูงเกินอัตราจนเกิดความร้อนสูง จนขาดวงจร เหมือนกับว่าฟิวส์ถูกระเบิด ดังนั้นการโปรแกรม PAL ก็คือ การเลือกตำแหน่งที่ต้องการทำลายฟิวส์ภายในส่วนอินพุตของ AND

หลังจากที่ฟิวส์ถูกทำลายไปแล้ว จะไม่สามารถทำให้มันฟื้นหรือต่อวงจรได้ตามเดิม คือ ขาดแล้วขาดเลย แต่สำหรับตำแหน่งที่ยังไม่ถูกทำลายสามารถนำมาโปรแกรมซ้ำให้ขาดได้

สรุปแล้วการโปรแกรม PAL ก็คือ การเลือกลอจิกทางอินพุตของ AND ให้ขึ้นอยู่กับสัญญาณอินพุตขาใดๆ อย่างไร โดยฟิวส์ที่ยังคงอยู่ (เครื่องหมายกากบาท) อาจจะแทนลอจิก "0" หรือ "1" ก็ได้ แต่ตำแหน่งที่ฟิวส์ถูกทำลายไปแล้ว (ไม่มีเครื่องหมายกากบาท) จะแทนลอจิกตรงกันข้าม

2.3 สัญลักษณ์ของลอจิกภายใน PAL

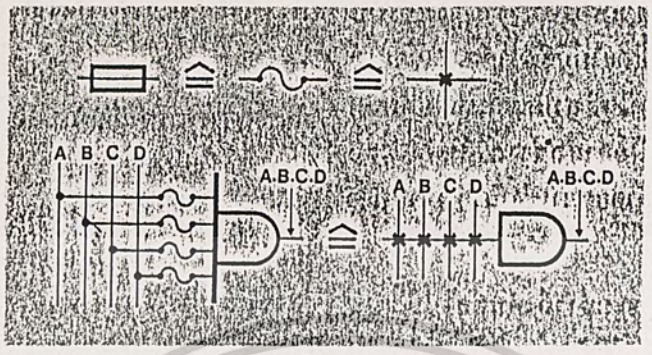
เนื่องจากวงจรลอจิกภายในตัว PAL ค่อนข้างซับซ้อนและมีเป็นจำนวนมาก การเขียนสัญลักษณ์แบบมาตรฐานทั่วไปเพื่อแสดงวงจรภายใน PAL จึงทำให้ดูสับสนวุ่นวายและต้องเปลืองเนื้อที่มาก โรงงานผู้ผลิตส่วนใหญ่จึงนิยมใช้สัญลักษณ์ที่กำหนดขึ้นมาใหม่ เพื่อให้เขียนแสดงวงจรภายในได้สะดวกขึ้นและอ่านความหมายจากวงจรได้ง่ายขึ้น ดังนี้



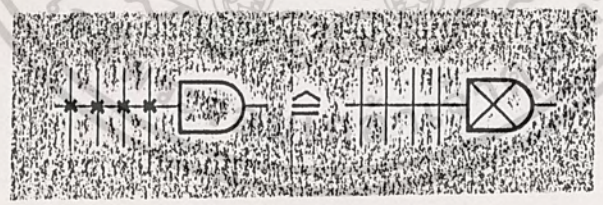
สัญญาณจากขาอินพุตมักจะต้องป้อนเข้าบัฟเฟอร์ 2 ตัวก่อนเพื่อให้ได้ 2 สัญญาณที่ตรงกันข้ามกัน คือ บัฟเฟอร์ตัวหนึ่งจะไม่กลับลอจิกแต่อีกตัวหนึ่งจะกลับลอจิก เพื่อให้การเขียนวงจรลอจิกภายใน PAL ทำได้ง่ายขึ้นแทนที่จะต้องเขียนสัญลักษณ์ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
028740

บัฟเฟอร์ทั้ง 2 ตัว จึงรวมมาเขียนเป็นตัวเดียวกันเลยโดยมีเอาต์พุต 2 สัญญาณดังรูปที่แสดงได้ข้างบนนี้



ขาอินพุตของเกตมักจะมีขาอินพุตอยู่หลายสัญญาณและจะต่อเข้ากับฟิวส์ในลักษณะแมทริกซ์ เพื่อให้เขียนวงจรได้ง่ายขึ้นจึงใช้เครื่องหมายกากบาทแสดงถึงว่ามีฟิวส์ต่ออยู่ระหว่างขาอินพุตของเกตกับขาสัญญาณทางแนวตั้งตามตำแหน่งของเครื่องหมายกากบาทนั้นๆ จึงทำให้เขียนขาสัญญาณอินพุตของเกตเพียงเส้นเดียว แสดงจำนวนขาสัญญาณอินพุตของเกตหลายสัญญาณได้ (จำนวนอินพุตของเกต = จำนวนขาสัญญาณอินพุตทางแนวตั้ง)



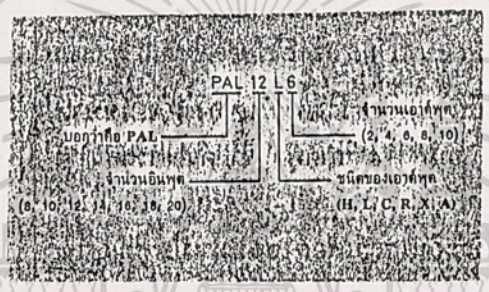
ในกรณีที่ฟิวส์ทุกตัวที่ต่ออยู่เข้ากับอินพุตของเกตไม่ถูกระเบิดทิ้งไปเลยแม้แต่ตัวเดียว จะเขียนเครื่องหมายกากบาทเพียงตัวเดียวเข้าที่สัญลักษณะของเกตเลย ในกรณีเช่นนี้ยังแสดงถึงว่าอินพุตของเกตตัวนั้นจะเป็น "0" เสมอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 วิธีการดูเบอร์ของ PAL

PAL ที่ได้ผลิตออกมาจำหน่ายแล้ว มีอยู่มากมายหลายขนาดโดยจะกำหนดเป็นเบอร์ต่างๆ ที่สอดคล้องกับโครงสร้างภายในของมัน คุณสมบัติของ PAL ที่พอจะพบเห็นได้ดังนี้

- จำนวนอินพุต - 8, 10, 12, 14, 16, 18 หรือ 20 ขา
- จำนวนเอาต์พุต - 2, 4, 6, 8, หรือ 10 ขา
- บัฟเฟอร์ทางเอาต์พุต - สามารถจะป้อนกลับเข้ามาเป็นอินพุตได้
- โปรแกรมได้ทั้งทางอินพุตและเอาต์พุต
- ทำหน้าที่คำนวณทางคณิตศาสตร์ได้



รูปที่ 4 การอ่านความหมายจากเบอร์ PAL

การกำหนดเบอร์ของ PAL จะแสดงให้ถึงจำนวนอินพุต, เอาต์พุต และชนิดของเอาต์พุต ดังแสดงความหมายของเบอร์เอาไว้ในรูปที่ 4 ตัวอักษรซึ่งบอกชนิดของเอาต์พุตมีรายละเอียดดังนี้

- H - แอคทีฟที่ "1"
- L - แอคทีฟที่ "0"
- C - มีให้เลือกทั้งแอคทีฟที่ "1" และ "0"
- R - เป็นรีจิสเตอร์ หมายถึง รักษาสถานะเอาต์พุตให้ค้างเอาไว้ได้ด้วย ฟลิป-ฟลอปและเอาต์พุตยังป้อนย้อนกลับมาที่อินพุตของ AND ได้ (โปรแกรมได้ด้วย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

. X - เป็นรีจิสเตอร์แบบ EX-OR

. A - เป็นรีจิสเตอร์คำนวณทางคณิตศาสตร์ได้

นอกจากนี้ยังจะมีตัวอักษรตามท้ายตัวเลขซึ่งบอกจำนวนเอาต์พุต โดยจะแสดงถึงความเร็วหรือกระแสใช้งาน, ช่วงอุณหภูมิและลักษณะรูปร่าง

2.5 วงจรภายในของ PAL ชนิดต่างๆ

ตารางที่ 1 เป็นตัวอย่างของเบอร์ต่างๆ และคุณสมบัติของ PAL ซึ่งสัญลักษณ์แสดงโครงสร้างภายในจะแสดงไว้ในรูปที่ 5 สำหรับวงจรทางเอาต์พุตของชนิดต่างๆ แสดงได้ดังรูปที่ 6

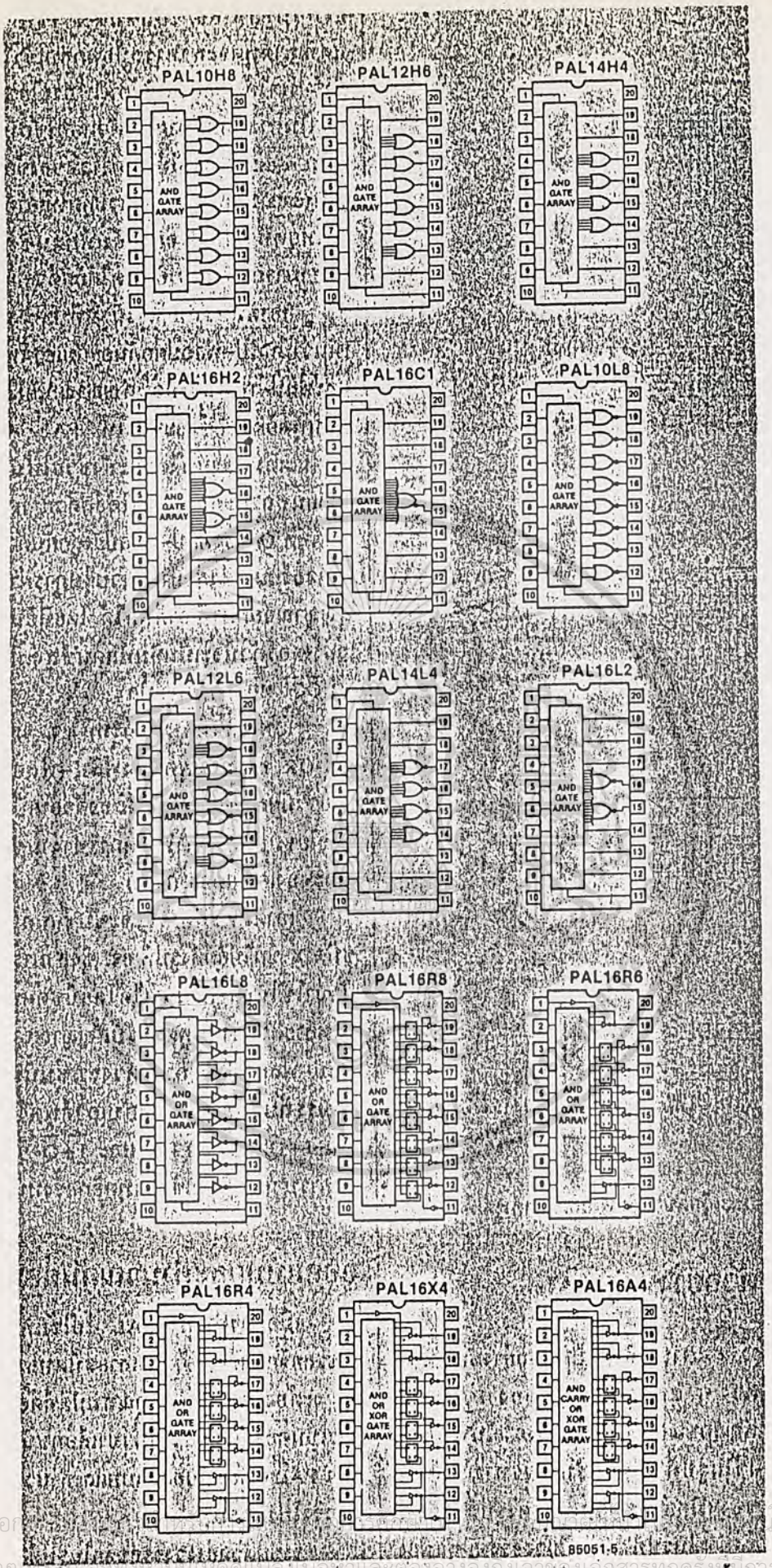
ในรูปที่ 6ก. เป็นวงจรง่ายๆ เพียงอินพุตเดียวและเอาต์พุตเดียวของชนิด L ส่วนรูปที่ 6ข. เป็นชนิด L เช่นกัน แต่ที่เอาต์พุตจะสามารถป้อนกลับเข้าไปยังอินพุตได้ คุณสมบัติข้อนี้ให้ประโยชน์มากในการออกแบบเป็นตัวเลื่อนข้อมูล (shift register) หรือให้เกิดการวนของข้อมูล (data loop) และเมื่อควบคุมให้เอาต์พุตหยุดทำงานหรือเป็นอิมพีแดนซ์สูง จะสามารถใช้เป็นขาอินพุตได้อีกด้วย

เอาต์พุตชนิด R ในรูปที่ 6ค. จะเห็นว่ามีฟลิป-ฟลอปต่อคั่นอยู่กับเอาต์พุตและฟลิป-ฟลอปยังป้อนสัญญาณกลับไปที่อินพุตอีกด้วยซึ่งจะช่วยให้ PAL จำสถานะเดิมของมันได้และยังช่วยให้มันเปลี่ยนสถานะกลับไปมาได้อีกด้วย ทางเอาต์พุต Q ของฟลิป-ฟลอปยังถูกคั่นด้วยอินเวอร์เตอร์ซึ่งสามารถควบคุมการผ่านสัญญาณออกมาที่เอาต์พุตได้ โดยที่อินเวอร์เตอร์ตัวนี้จะเป็นอิมพีแดนซ์สูงเมื่อขา OC หยุดการอีนาเบิ้ล

เอาต์พุตชนิด X ในรูปที่ 6ง. จะมี EX-OR อยู่ที่อินพุตของฟลิป-ฟลอปเพื่อรวมผลของ OR ทั้งสองตัวอันเป็นประโยชน์สำหรับการหยุดการทำงานชั่วขณะของเคาน์เตอร์ (hold)

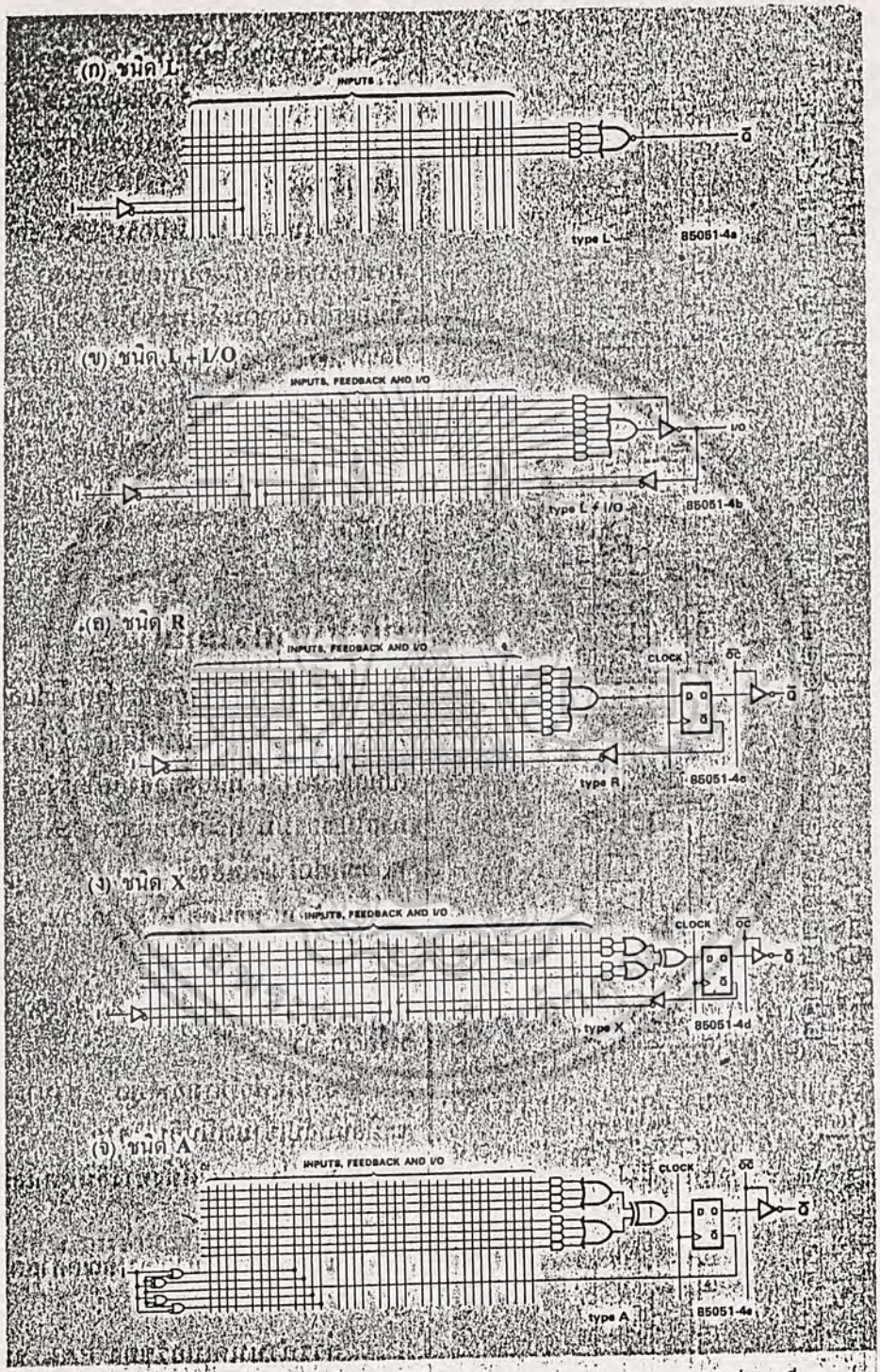
เอาต์พุตชนิด A จะซับซ้อนกว่าชนิด X ขึ้นไปอีกดังรูปที่ 6จ. ซึ่งสามารถนำมาใช้ในการคำนวณได้โดยมีชุดเกตอยู่หลายตัวทางอินพุตซึ่งรับสัญญาณจากฟลิป-ฟลอปด้วย การจัดวงจรเช่นนี้จึงสามารถบวกอินพุตเข้ากับเอาต์พุตได้ เช่น I+Q, I+Q, I+Q และ I+Q ซึ่งผลการบวกจะป้อนเข้าอินพุตอีกครั้งหนึ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6 วงจรภายใน PAL แบบคร่าว ๆ ที่ช่วยให้ตัดสินใจเลือกใช้เบอร์ต่างได้ง่ายขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเอกสารนี้ทุกครั้งที่มีการนำไปใช้

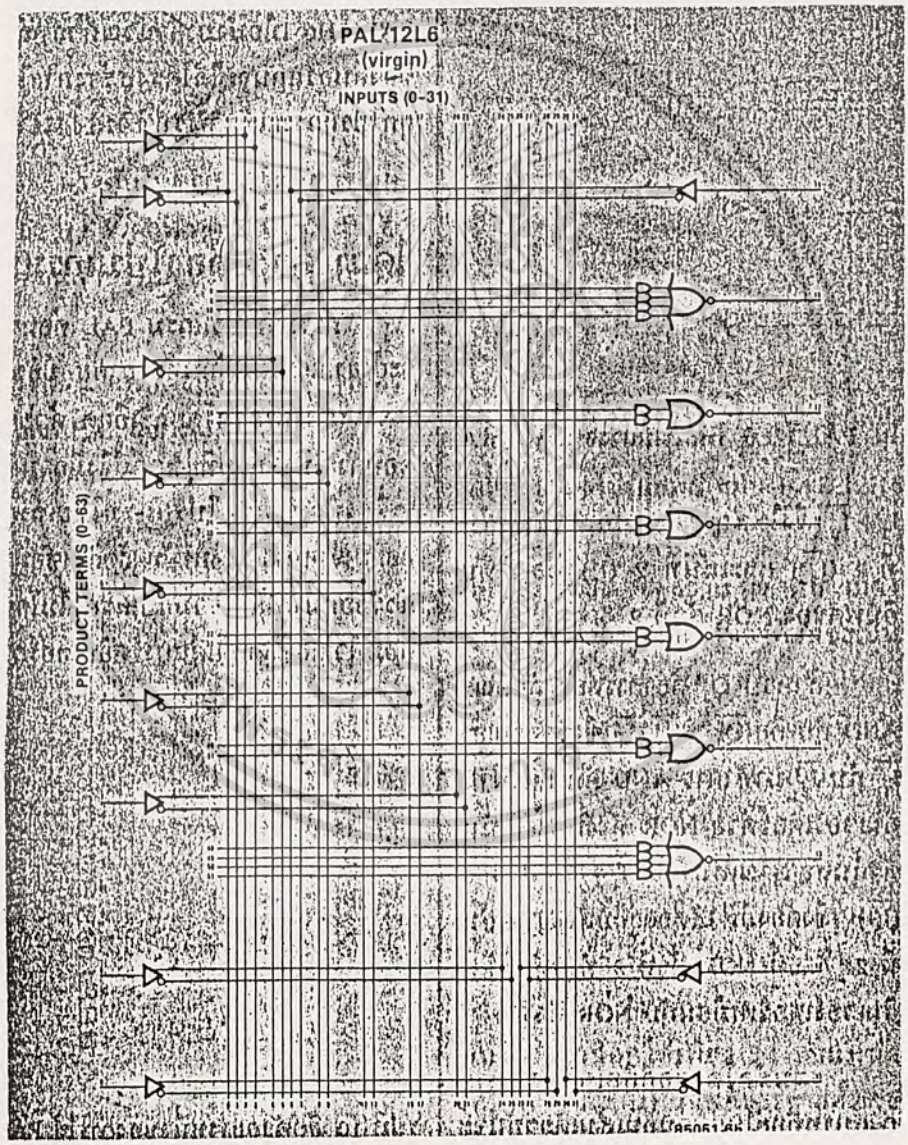


รูปที่ 8 วงจรภายใน PAL ชนิดต่าง ๆ โดยละเอียดของอินพุตและเอาต์พุตเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6 กรรมวิธีการโปรแกรม PAL

กรรมวิธีโปรแกรม PAL ค่อนข้างจะยุ่งยากและซับซ้อนต้องอาศัยคู่มือของ PAL นำมาศึกษาโดยละเอียด ด้วยขั้นตอนการโปรแกรมที่ยุ่งยากนี้จึงมีการผลิตเครื่องมือสำหรับโปรแกรม โดยเฉพาะออกมาจำหน่าย บางบริษัทผลิตเป็นซอฟต์แวร์ออกมาเพื่อใช้ร่วมกับเครื่องไมโครคอมพิวเตอร์ โดยมีแผ่นการ์ดอินเตอร์เฟสเข้ามาช่วย และการโปรแกรมจะเขียนด้วยภาษาที่ง่ายขึ้นเช่นเพียงแต่บอกความสัมพันธ์ของอินพุต-เอาต์พุต ในรูปแบบของทฤษฎีดีมอร์แกนเท่านั้น



รูปที่ 7 วงจรโลจิกภายใน PAL 12L6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่ก็ใช่ว่า เราจะไม่สามารถโปรแกรมได้ด้วยตนเอง ความจริงแล้วเราสามารถออกแบบเครื่องมือสำหรับ PAL ได้โดยการป้อนแรงดันที่เหมาะสมซึ่งมีหลายระดับ (0, 5, 6 และ 11.5 V สำหรับโปรแกรม) เข้าไปยังขาต่างๆ ของ PAL ให้ถูกต้อง และเป็นไปตามลำดับเวลาเพื่อเลือกตำแหน่งที่ต้องการระเบิดฟิวส์แล้วเปลี่ยนตำแหน่งไปเรื่อยๆ จนครบตามที่ได้ออกแบบและคำนวณไว้

ในที่นี้จึงจะกล่าวถึงวิธีการโปรแกรม PAL โดยย่อให้เป็นที่พอเข้าใจเพื่อการออกแบบเครื่องมือหรือเพื่อการทดลองโปรแกรม

กรรมวิธีที่ซับซ้อน

แรงดันในขณะที่โปรแกรมจะมีค่าสูงถึง 11.5 V ผิดพลาดได้ในช่วง $\pm$0.5 V แต่พัลส์สำหรับป้อนเข้าเพื่อโปรแกรมจะแคบมากเพียง 10-50 us เท่านั้น เนื่องจากการจัดเรียงตำแหน่งของฟิวส์จะเป็นแบบแมทริซ์ ดังนั้นการเลือกตำแหน่งฟิวส์ที่ต้องการระเบิดจึงแบ่งออกเป็น 2 กลุ่ม คือ ทางแนวตั้งและแนวนอน

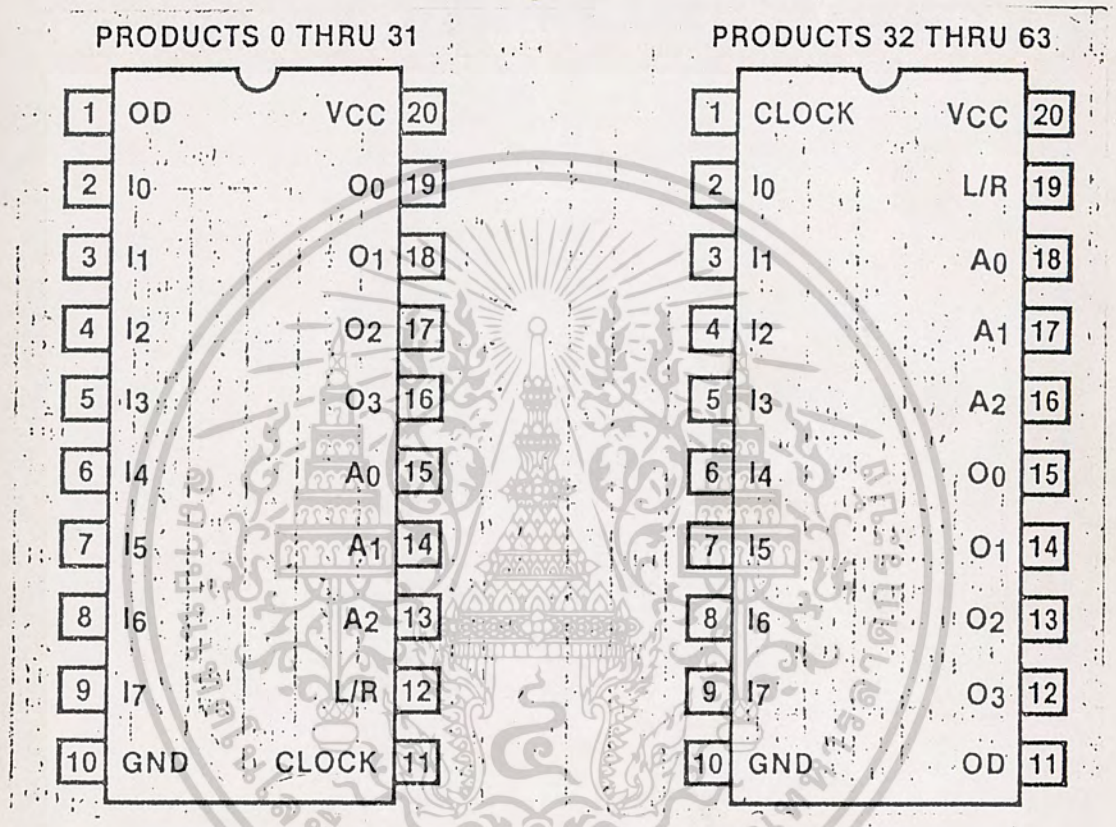
ในการเลือกตำแหน่งทางแนวตั้ง มีวิธีการทำได้ง่ายกว่าทางแนวนอน คือมีตำแหน่งให้เลือกตั้งแต่ 0-31 ตัวเลขนี้สังเกตได้ว่ามีให้เลือกได้ทั้งสิ้น 32 ตำแหน่ง แต่สำหรับ PAL ขนาด 12 อินพุต จะเลือกได้เพียง 24 ตำแหน่ง (ดูรูปที่ 7 ซึ่งจะใช้วิธีข้ามตัวเลขไปบางช่วง เช่นจากตัวอย่างในรูปที่ 7 ตัวเลขที่ขาดหายไปคือ 10, 11, 14, 15, 18, 19, 22 และ 23

การเลือกตำแหน่งทางแนวตั้ง (input line number) จะถูกกำหนดด้วยขาสัญญาณ I_0 -I_7 และ L/R โดยต้องป้อนแรงดันให้กับขาเหล่านี้ตามตารางที่ 2 ซึ่งมีทั้งลอจิก "ต่ำ" (L), "สูง" (H), 11.5 V (HH) และแรงดัน 5 V ที่ผ่านตัวต้านทานพูลอัพ (R)

ขาสัญญาณ I_0 -I_7 และ L/R จะถูกกำหนดขึ้นใหม่ตามรูปที่ 8 ก. ซึ่งจะใกล้เคียงกับการใช้งานในขณะปกติหลังจากที่โปรแกรมไปแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเลือกตำแหน่งทางแนวนอน (product line number) มีวิธีการที่ซับซ้อนขึ้นไอีก เนื่องจากมีตำแหน่งให้เลือกตั้งแต่ 0-63 หรือเลือกได้ทั้งสิ้น 64 ตำแหน่ง แต่มีขาสัญญาณให้เลือกเพียง 7 ขาเท่านั้น คือ A_0 -A_2 และ O_0 -O_3 น้อยกว่าขาสัญญาณที่เลือกทางแนวตั้งเสียอีก ตัวเลข 0-63 นี้ อาจมีตัวเลขบางช่วงที่ถูกข้ามไปบ้างเช่นเดียวกัน ต้องเปิดดูโครงสร้างภายในของ PAL แต่ละเบอร์



ก. แนวนอนช่วงต่ำ ข. แนวนอนช่วงสูง

รูปที่ 8 ขาสัญญาณต่างๆ ตามแนวตั้งและแนวนอน

เนื่องจากมีขาสัญญาณเพียง 7 ขา แต่ต้องเลือกถึง 64 ตำแหน่ง ผู้ออกแบบ PAL จึงต้องใช้วิธีแบ่งออกเป็น 2 ช่วง คือ ช่วงต่ำ (0-31) และช่วงสูง (32-63) การแบ่งวิธีนี้จึงต้องกำหนดขาสัญญาณ A_0 -A_2 และ O_0 -O_3 ให้แตกต่างกันในแต่ละช่วงดังรูปที่ 8 นอกจากนี้ขาสัญญาณที่ใช้ควบคุมโปรแกรม (OD, CLOCK, L/R) จะถูกเปลี่ยนตามไปด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 8 ก. เป็นการจัดขาในช่วงต่ำ และในรูปที่ 8 ข. เป็นการจัดขาในช่วงสูง ส่วนการเลือกตำแหน่งทางแนวนอนในแต่ละช่วงจะเป็นดังตารางที่ 3 ซึ่งต้องป้อนพลัส์ 11.5 V เป็นช่วงเวลา 10-50 us (V_{PH}) ด้วย

Input Line Number	Pin Identification								
	I_7	I_6	I_5	I_4	I_3	I_2	I_1	I_0	L/R
0	HH	HH	HH	HH	HH	HH	HH	L	Z
1	HH	HH	HH	HH	HH	HH	HH	H	Z
2	HH	HH	HH	HH	HH	HH	HH	L	HH
3	HH	HH	HH	HH	HH	HH	HH	H	HH
4	HH	HH	HH	HH	HH	HH	L	HH	Z
5	HH	HH	HH	HH	HH	HH	H	HH	Z
6	HH	HH	HH	HH	HH	HH	L	HH	HH
7	HH	HH	HH	HH	HH	HH	H	HH	HH
8	HH	HH	HH	HH	HH	L	HH	HH	Z
9	HH	HH	HH	HH	HH	H	HH	HH	Z
10	HH	HH	HH	HH	HH	L	HH	HH	HH
11	HH	HH	HH	HH	HH	H	HH	HH	HH
12	HH	HH	HH	HH	L	HH	HH	HH	Z
13	HH	HH	HH	HH	H	HH	HH	HH	Z
14	HH	HH	HH	HH	L	HH	HH	HH	HH
15	HH	HH	HH	HH	H	HH	HH	HH	HH
16	HH	HH	HH	L	HH	HH	HH	HH	Z
17	HH	HH	HH	H	HH	HH	HH	HH	Z
18	HH	HH	HH	L	HH	HH	HH	HH	HH
19	HH	HH	HH	H	HH	HH	HH	HH	HH
20	HH	HH	L	HH	HH	HH	HH	HH	Z
21	HH	HH	H	HH	HH	HH	HH	HH	Z
22	HH	HH	L	HH	HH	HH	HH	HH	HH
23	HH	HH	H	HH	HH	HH	HH	HH	HH
24	HH	L	HH	HH	HH	HH	HH	HH	R
25	HH	H	HH	HH	HH	HH	HH	HH	R
26	HH	L	HH	HH	HH	HH	HH	HH	HH
27	HH	H	HH	HH	HH	HH	HH	HH	HH
28	L	HH	HH	HH	HH	HH	HH	HH	Z
29	H	HH	HH	HH	HH	HH	HH	HH	Z
30	L	HH	HH	HH	HH	HH	HH	HH	HH
31	H	HH	HH	HH	HH	HH	HH	HH	HH

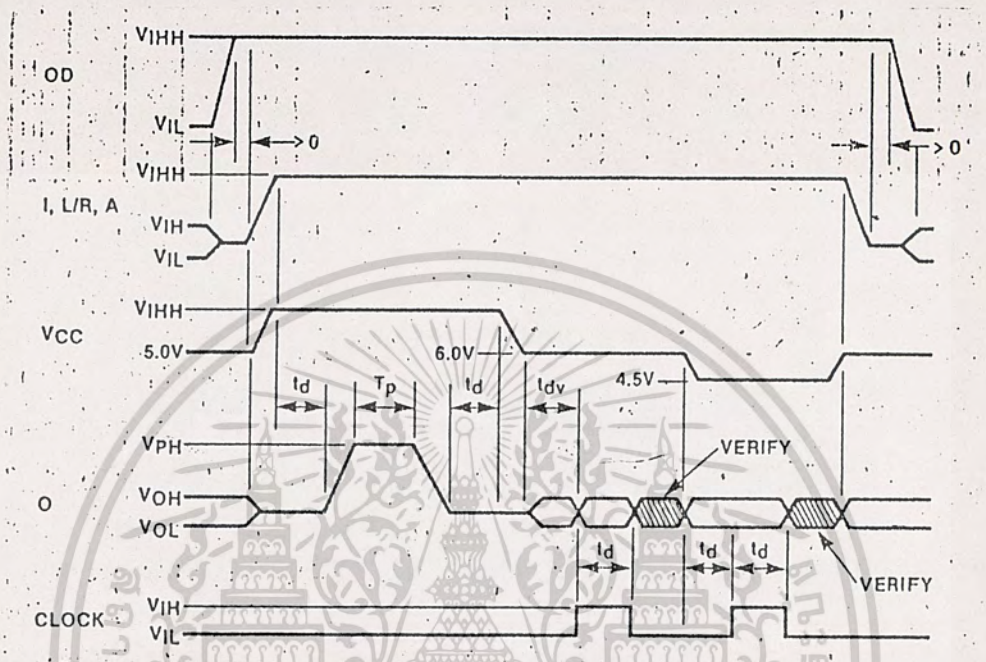
ตารางที่ 2 INPUT LINE SELECT

Product Line Number	Pin Identification						
	O_3	O_2	O_1	O_0	A_2	A_1	A_0
0,32	Z	Z	Z	HH	Z	Z	Z
1,33	Z	Z	Z	HH	Z	Z	HH
2,34	Z	Z	Z	HH	Z	HH	Z
3,35	Z	Z	Z	HH	Z	HH	HH
4,36	Z	Z	Z	HH	HH	Z	Z
5,37	Z	Z	Z	HH	HH	Z	HH
6,38	Z	Z	Z	HH	HH	HH	Z
7,39	Z	Z	Z	HH	HH	HH	HH
8,40	Z	Z	HH	Z	Z	Z	Z
9,41	Z	Z	HH	Z	Z	Z	HH
10,42	Z	Z	HH	Z	Z	HH	Z
11,43	Z	Z	HH	Z	Z	HH	HH
12,44	Z	Z	HH	Z	HH	Z	Z
13,45	Z	Z	HH	Z	HH	Z	HH
14,46	Z	Z	HH	Z	HH	HH	Z
15,47	Z	Z	HH	Z	HH	HH	HH
16,48	Z	HH	Z	Z	Z	Z	Z
17,49	Z	HH	Z	Z	Z	Z	HH
18,50	Z	HH	Z	Z	Z	HH	Z
19,51	Z	HH	Z	Z	Z	HH	HH
20,52	Z	HH	Z	Z	HH	Z	Z
21,53	Z	HH	Z	Z	HH	Z	HH
22,54	Z	HH	Z	Z	HH	HH	Z
23,55	Z	HH	Z	Z	HH	HH	HH
24,56	HH	Z	Z	Z	Z	Z	Z
25,57	HH	Z	Z	Z	Z	Z	HH
26,58	HH	Z	Z	Z	Z	HH	Z
27,59	HH	Z	Z	Z	Z	HH	HH
28,60	HH	Z	Z	Z	HH	Z	Z
29,61	HH	Z	Z	Z	HH	Z	HH
30,62	HH	Z	Z	Z	HH	HH	Z
31,63	HH	Z	Z	Z	HH	HH	HH

ตารางที่ 3 PRODUCT LINE SELECT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตำแหน่งที่ถูกเลือกทางแนวตั้งและแนวนอนแล้วจะไปกำหนดตำแหน่งที่ต้องการระเบิดฟิวส์ โดยจะต้องป้อนแรงดันที่ขาสัญญาณจ่างๆ ให้เป็นไปตามแผนผังเวลาในรูปที่ 9 ซึ่งแสดงช่วงเวลาการตรวจสอบ (verify) เอาไว้ให้ด้วย อันเป็นประโยชน์ในการโปรแกรมซ้ำเข้าไปอีก ถ้าหากการโปรแกรมในครั้งแรกไม่สำเร็จ



รูปที่ 9 PROGRAMMING WAVEFORMS

จากขั้นตอนที่ยุ่งยากซับซ้อนเช่นนี้ เครื่องมือที่ใช้ในการโปรแกรมจะต้องสามารถจ่ายแรงดันให้แต่ละขาสัญญาณได้หลายระดับ และการควบคุมจังหวะเวลาให้เหมาะสม กรรมวิธีเช่นนี้มีระบบควบคุมที่ทำงานด้วยไมโครโปรเซสเซอร์เท่านั้นที่จะช่วยให้การโปรแกรมทำได้ง่ายขึ้น และยังสามารถเก็บรูปแบบการโปรแกรมเอาได้ได้อีกด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการสร้างเครื่องโปรแกรม PAL

3.1 การ INTERFACE กับ IBM PC

สัญญานต่าง ๆ บน SLOT ของ IBM/PC

ภายใน IBM/PC ได้มีการออกแบบให้สามารถเพิ่มเติมวงจรอินเตอร์เฟสเข้าไปในภายหลังได้ โดยผ่านทางสล็อตที่มีอยู่บนเมนบอร์ด (MAIN BOARD) สำหรับบนเมนบอร์ดนี้มีจำนวน 5 สล็อต (สำหรับใน IBM PC/XT จะมี 8 สล็อต) ซึ่งแต่ละสล็อตจะมีจำนวนขาทั้งสิ้น 62 ขา แบ่งออกเป็น 2 ข้าง ๆ ละ 31 ขาส่วนการเรียกตำแหน่งขาของสล็อต จะขึ้นอยู่กับว่าขานั้นอยู่ข้างใด (ซ้ายหรือขวา) ของสล็อตโดยขาทางด้านซ้ายของสล็อตจะเรียกโดยใช้อักษร "B" นำหน้าเลขตำแหน่งของขาเช่น ขา B16 ก็คือ ขาทางด้านซ้ายของสล็อตขาที่ 16 ส่วนขาทางด้านขวาของสล็อตจะเรียกโดยใช้อักษร "A" นำหน้า แต่ละขาของสล็อตนั้นจะเชื่อมกับเส้นสัญญาณต่างๆบนเมนบอร์ด ทำให้การต่อวงจรอินเตอร์เฟส.กับ IBM/PC สามารถทำได้โดยสะดวก ซึ่งสัญญาณที่เชื่อมต่อกับขาของสล็อตเหล่านี้จะประกอบไปด้วยสัญญาณของบัสแอดเดรส (ADDRESS BUS),บัสข้อมูล (DATA BUS), บัสควบคุมสำหรับการเขียน/อ่านข้อมูลจากหน่วยความจำหรือ พอร์ตI/O, เส้นสัญญาณสำหรับการขออินเตอร์รัพของวงจรอินเตอร์เฟส,เส้นสัญญาณสำหรับการขอ DMA, สัญญาณฐานเวลา (TIMING SIGNAL) ต่างๆ ที่ใช้ในระบบ ,เส้นสัญญาณแสดงการ รีเฟรชหน่วยความจำ และ สัญญาณสำหรับการตรวจสอบความผิดพลาด (I/O CHECK)

นอกจากสัญญาณเหล่านี้แล้วสล็อตบนเมนบอร์ดยังเชื่อมต่อกับแหล่งจ่ายไฟต่างๆที่ใช้ในระบบอีกด้วย คือ +5 VDC, -5 VDC, +12 VDC และ -12 VDC

รายละเอียดของสัญญานต่าง ๆ บน SLOT

(I), (O) และ (I/O) หมายถึง ทิศทางของขาสัญญาณเมื่อเทียบกับเมนบอร์ด

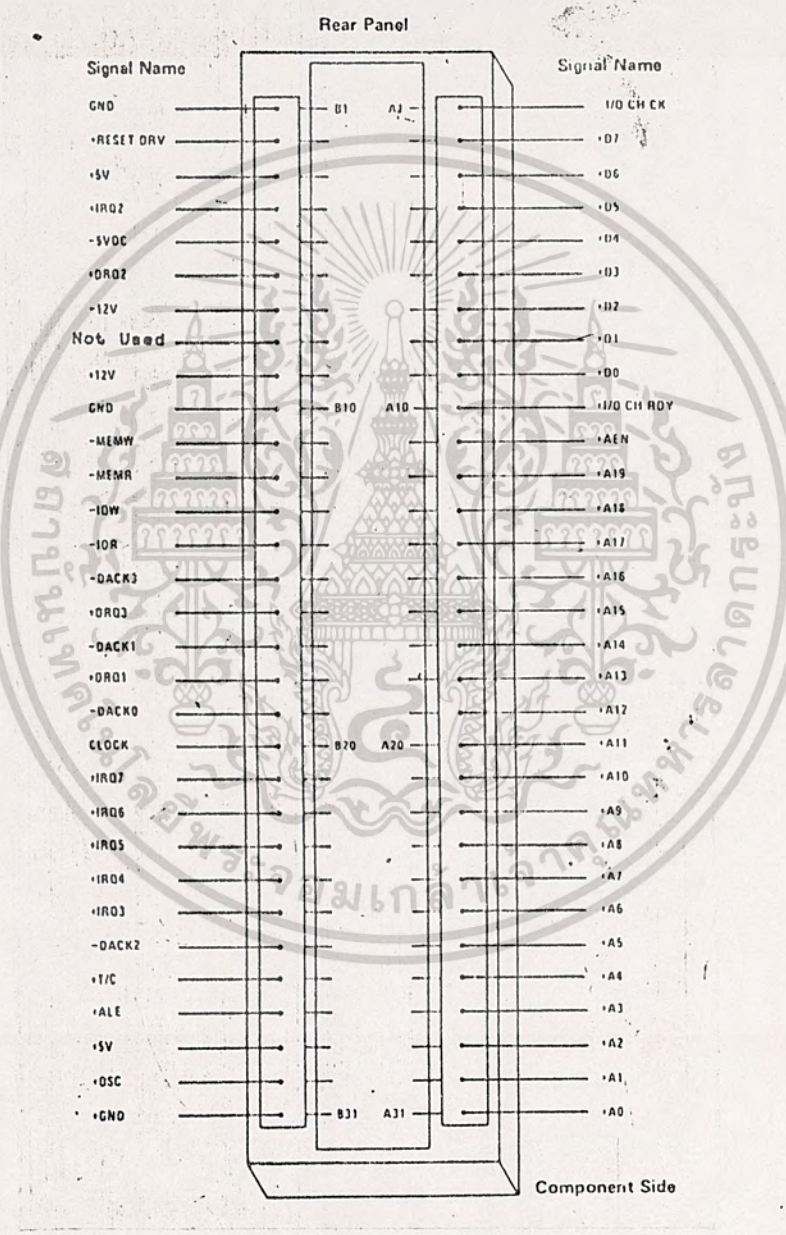
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่ (I) หมายถึง ขาสัญญาณอินพุต

(O) หมายถึง ขาสัญญาณเอาต์พุต

(I/O) หมายถึง ขาสัญญาณที่เป็นได้ทั้งอินพุตและเอาต์พุต

(*I/O) หมายถึงในช่วงการทำงานปกติเป็นขาสัญญาณเอาต์พุต แต่จะเป็นอินพุตในช่วงที่เกิดขบวนการ DMA



รูปที่ 10 ขาสัญญาณที่ SLOT ของ IBM PC/XT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับขาสัญญาณที่มีเครื่องหมายลบน่าหน้าหมายถึง ขาสัญญาณที่แอคตีฟที่ลอจิก "0" และขาสัญญาณที่ไม่มีหรือมีเครื่องหมายบวกน่าหน้าอยู่จะหมายถึง ขาสัญญาณที่แอคตีฟที่ลอจิก "1" สัญญาณที่ต่ออยู่บนสล็อตนี้ใช้ขับไอซีทีทีแอลชนิดโลว์เพาเวอร์ได้สองตัวโดยไม่เกิดการโหลด หรือการเพี้ยนของสัญญาณ ขาสัญญาณต่างๆ บนสลอ็ตของ XT และ AT สามารถแบ่งออกเป็นกลุ่ม ๆ ได้ดังนี้

เพาว์เวอร์ซัพพลาย

Ground	ขาสัญญาณนี้ต่ออยู่กับกราวด์ของระบบเรกูเลเตอร์
+ 5V	ขาสัญญาณนี้ต่ออยู่กับไฟ DC เรกูเลเตอร์ +5 โวลท์
- 5V	ขาสัญญาณนี้ต่ออยู่กับไฟ DC เรกูเลเตอร์ -5 โวลท์
+ 12V	ขาสัญญาณนี้ต่ออยู่กับไฟ DC เรกูเลเตอร์ +12 โวลท์
- 12V	ขาสัญญาณนี้ต่ออยู่กับไฟ DC เรกูเลเตอร์ -12 โวลท์

แอดเดรสบัส และ สัญญานต่างๆที่เกี่ยวข้อง

SA0-SA19	เป็นแอดเดรสบิตที่ 0 ถึง 19 โดยที่ SA0 มีนัยสำคัญต่ำที่สุด ขาสัญญาณนี้จะแอคตีฟ เมื่อขาสัญญาณ BALE มีสถานะเป็น "1" และจะถูกแลตช์ไว้ตอนขอบขาลงของขาสัญญาณ BALE แอดเดรสทั้ง 20 บิตนี้ สามารถอ้างหน่วยความจำได้ถึง 1 เมกกะไบต์ XT และสำหรับ AT เมื่อใช้ร่วมกับ LA17-LA23 จะอ้างได้ถึง 16 เมกกะไบต์
LA17-LA23 (*I/O)	(เฉพาะรุ่น AT) ขาสัญญาณนี้จะแอดเดรสเมื่อขาสัญญาณ BALE มีสถานะเป็นลอจิก "1" แต่จะไม่มีการแลตช์ไว้ ตอนขอบขาลงของขาสัญญาณ BALE ดังนั้นถ้าอุปกรณ์ I/O ไม่มีการอ้างแอดเดรสเกิน 1 เมกกะไบต์ ขาสัญญาณนี้ก็ไม่จำเป็นต้องใช้ แต่ถ้ามีการอ้างแอดเดรสเกินอุปกรณ์ I/O จะต้องมีการแลตช์สัญญาณนี้ โดยใช้ขอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาลงของขาสัญญาณ BALE ร่วมกับขาสัญญาณ -MEMW และ -MEMR

AEN (O) (Address Enable) ขาสัญญาณนี้จะแอคตีฟเมื่อตัวควบคุม DMA ได้ทำการควบคุมบัสต่าง ๆ ของระบบแล้ว ดังนั้นการอ้างพอร์ตของสัญญาณ I/O จะต้องใช้สัญญาณนี้ ในการดีโค้ดด้วย เพื่อที่จะไม่ทำให้เกิดการติดต่อระหว่างระบบกับอุปกรณ์ I/O ตัวอื่นยกเว้นตัวที่กำลังทำขบวนการ DMA อยู่ BALE (Address Latch Enable) (O) ขาสัญญาณนี้ใช้ในการแสดงการเริ่มต้นของขบวนการต่าง ๆ ที่มีการติดต่อกับหน่วยความจำโดยจะแอคตีฟเมื่อค่าแอดเดรสที่ CPU ต้องการติดต่อด้วยอยู่บนแอดเดรสบัส เรียบร้อยแล้ว ตามปกติขอบขาลงของสัญญาณนี้จะทำให้เกิดการแลตช์สัญญาณ SA0-SA19 และถ้ามีการอ้างแอดเดรสเกิน 1 เมกกะไบต์ใน AT จะใช้ขอบขาลงสัญญาณนี้ในการแลตช์สัญญาณ LA17- LA23 ด้วยเช่นกัน แต่สำหรับในขบวนการ DMA สัญญาณนี้จะมีสถานะเป็น "1" ตลอด

SBHE (*I/O) (เฉพาะรุ่น AT) (Bus High Enable) เป็นขาสัญญาณที่ใช้แสดงว่ามีการรับส่งข้อมูลในบิตที่ SD8-SD15

ดาต้าบัส

SD0-SD7 สำหรับรุ่น AT จะมี SD0-SD15 เพิ่มขึ้นมาด้วยคือ ดาต้าบิต 0 ถึง 7 สำหรับ XT และสำหรับรุ่น AT คือ ดาต้าบิต 0 ถึง 15 โดยที่ SD0 มีนัยสำคัญต่ำสุด สำหรับ AT ถ้ามีการติดต่อกับบิตที่ SD8- SD15 สามารถตรวจสอบได้จากขาสัญญาณ SBHE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญานอินเตอร์รันต์

IRQ2-IRQ7 (Interrupt Request) (สำหรับรุ่น AT จะเป็น IRQ3-7,9-12, 14,1 5) เป็นขาสัญญาณอินเตอร์รัพต์ CPU สำหรับ AT ลำดับความสำคัญของสัญญาณ IRQ เป็นดังนี้คือ 9,10, 11, 12, 14, 15, 3, 4, 5, 6และ 7 โดย IRQ9 มีลำดับความสำคัญมากที่สุด และ IRQ7 มีลำดับความสำคัญน้อยที่สุด สำหรับ XT IRQ2 จะมีลำดับความสำคัญมากที่สุด รอง ๆ ลงไปคือ IRQ 3,4, 5, 6, 7 สำหรับรายละเอียดในการนำไปใช้งานแต่ละอินเตอร์รัพต์ ให้ดูจากตารางการจัดลำดับอินเตอร์รัพต์ในหัวข้อการอินเตอร์รัพต์ โดยปกติสัญญาณนี้จะมีสถานะเป็น "0" เสมอ ถ้าต้องการอินเตอร์รัพต์ CPU ให้ส่งพัลส์ที่เป็นลอจิก "1" ให้กับมัน โดยที่ไม่จำเป็นต้องคำนึงถึงคาบเวลาของพัลส์ทั้งนี้เพราะระบบของ IBM ตัวอินเตอร์รัพต์คอนโทรลเลอร์ (8259 Interrupt Controller) จะถูกโปรแกรมให้ทำการตรวจสอบสัญญาณอินเตอร์รัพต์โดยใช้ขาขอบลงของสัญญาณนี้

I/O CH CK (I) I/O Channel Check) เป็นขาสัญญาณที่บอกถึงความผิดพลาดในการรับส่งข้อมูลที่ตรวจสอบจากพาริตี้บิต ถ้าพาริตี้บิตที่อ่านจากหน่วยความจำ กับพาริตี้บิตที่สร้างขึ้นจากขบวนการรับส่งข้อมูลมีค่าไม่เท่ากัน แสดงว่าเกิดความผิดพลาดในการรับส่งข้อมูล สัญญาณนี้จะทำให้เกิดการอินเตอร์รัพต์ CPU แบบ NMI เพื่อบอกให้ CPU ทราบว่าเกิด Parity Error ขึ้น CPU จะแสดงข้อความบอกความผิดพลาดขึ้น และจะหยุดการทำงาน (Halt) เพื่อให้ผู้ใช้ตรวจสอบหาสาเหตุความผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณที่ใช้ในขบวนการ DMA

DRQ1-DRQ3 (DMA Reques) (สำหรับรุ่น AT จะเป็น DRQ0-3,5-7)

(I) เป็นขาสัญญาณใช้ในการขอทำขบวนการ DMA โดยที่ DRQ0 มีลำดับความสำคัญมากที่สุด และ DRQ3 มีลำดับความสำคัญน้อยที่สุดสำหรับรุ่น XT และสำหรับรุ่น AT ขา DRQ7 จะมีลำดับความสำคัญน้อยที่สุดบน XT DRQ0 ใช้สำหรับการรีเฟรชหน่วยความจำแบบไดนามิคจึงไม่มีขาสัญญาณ DRQ0 ต่อออกมาที่สล็อต แต่สำหรับ AT แล้วจะมีวงจรโดยเฉพาะสำหรับใช้ในการรีเฟรช ในหน่วยความจำแบบไดนามิคอยู่แล้วดังนั้นขา DRQ0 จึงว่างลงและนำมาต่อที่สล็อตเพื่อให้อุปกรณ์ได้ใช้งานได้ บางครั้งอาจจะเรียก DRQ0 เป็น DRQ4 ก็ได้ เพื่อป้องกันการสับสนกับส่วนที่ใช้รีเฟรชไดนามิคแรมบน XT (เช่นเดียวกัน ก็จะเรียก DACK0 เป็น DACK4)

การขอทำ DMA ทำได้โดยทำให้ขานี้มีสถานะเป็น "1" แล้วรอจนกระทั่งได้รับการตอบสนองการทำ DMA จาก CPU โดยการตรวจสอบสัญญาณ DACK ที่ส่งออกมา DACK0-3(Acknowledge)(สำหรับรุ่น AT จะเป็น DACK0-3,5-7) เป็น (O)สัญญาณตอบสนองการขอทำ DMA ของอุปกรณ์ I/O เพื่อให้อุปกรณ์ I/O ทราบว่าการขอทำขบวนการ DMA นั้นได้รับการตอบสนองแล้ว เช่นถ้ามีการขอทำ DMA ผ่านทาง DRQ2 และเมื่อ CPU รับรู้แล้ว จะทำให้สัญญาณ DACK2 แอคตีฟ

ถึงแม้ว่าบน XT จะมีการนำเอา DRQ0 ไปใช้ในการรีเฟรชไดนามิคแรมก็ตาม แต่สัญญาณ DACK0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ก็จะถูกต่อออกมาที่สลอ็ตด้วย เพื่อแสดงถึงขบวนการรีเฟรชไดนามิคแรมและอุปกรณ์ I/O สามารถนำสัญญาณนี้ใช้ในการรีเฟรชหน่วยความจำแบบไดนามิคที่อยู่ในตัวมันได้

Refres (O) (เฉพาะรุ่น AT)(Memory Refresh) มีหน้าที่เหมือนกับขาสัญญาณ DACKO ในรุ่น XT คือ ใช้แสดงขบวนการรีเฟรชหน่วยความจำเพราะว่าในรุ่น AT จะมีวงจรที่ใช้ในการรีเฟรชหน่วยความจำโดยตรงอยู่แล้ว ดังนั้นจึงไม่จำเป็นต้องใช้ขาสัญญาณ DRQ0 และ DACKO

Master (I) (เฉพาะรุ่น AT)(Master) ขาสัญญาณนี้จะใช้ร่วมกับ DMA Request ในการเข้าระบบบัสในขบวนการ DMA โดยที่ตัว DMA คอนโทรลเลอร์จะส่งสัญญาณ DMA Request แล้วรอจนกระทั่งได้รับการตอบสนองโดยสัญญาณ DACk เกิดการแอคตีฟขึ้น แล้วจึงจะส่งสัญญาณนี้ให้กับ CPU จะทำให้แอดเดรสบัส ดาต้าบัส และคอนโทรลบัส เข้าสู่สถานะไตรสเตด หรือ ไฮอิมพีแดนซ์ หลังจากนั้น DMA คอนโทรลเลอร์จะต้องรออีกหนึ่งคาบสัญญาณคลอ็ก ก่อนที่จะเข้าควบคุมบัสต่าง ๆ และจะต้องรออีก 2 ไซเคิล ก่อนที่จะทำการอ่าน หรือเขียนข้อมูล ช่วงเวลาที่สัญญาณนี้แอคตีฟไม่ควรเกิน 15 ไมโครวินาที มิฉะนั้นข้อมูลภายในหน่วยความจำจะสูญหายไปได้ เนื่องจากขาดสัญญาณรีเฟรชหน่วยความจำ

T/C (O) (Terminal Count) เป็นขาสัญญาณที่บอกอุปกรณ์ I/O ที่ทำ DMA ให้ทราบว่าจำนวนข้อมูลที่รับส่งขบวนการ DMA นี้ครบจำนวนแล้ว โดยจะส่งสัญญาณเป็นพัลส์ให้อุปกรณ์ I/O

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญานควบคุมต่าง

MEMR (*I/O) (Memory Read)(สำหรับรุ่น AT คือขาสัญญาณ SMEMR (System Memory Read)) ขาสัญญาณนี้จะเป็นตัวบอกให้หน่วยความจำส่งข้อมูลออกมาที่ดาต้าบัส แต่สำหรับ AT สัญญาณ SMEMR จะแอคตีฟ เมื่อเกิดการอ่านข้อมูลจากหน่วยความจำที่อยู่ภายใน 1 เมกกะไบต์แรกเท่านั้น MEMR (เฉพาะรุ่น AT)(Memory Read) ขาสัญญาณนี้ไม่ใช่สัญญาณเดียวกัน (O)กับสัญญาณ MEMR ใน XT มันจะแอ็คตีฟก็ในทุก ๆ ขบวนการอ่านข้อมูลที่เกิดขึ้น ไม่ว่าอยู่ในช่วงหน่วยความจำ 1 เมกกะไบต์แรกหรือไม่ MEMW(Memory Write)(สำหรับรุ่น AT คือ ขาสัญญาณ SMEMW (*I/O) (System Memory Write)) ขาสัญญาณนี้จะเป็นตัวบอกให้หน่วยความจำเก็บข้อมูลจากดาต้าบัสแต่สำหรับ AT สัญญาณ SMEMW จะ แอคตีฟ เมื่อเกิดการเก็บข้อมูลจากหน่วยความจำที่อยู่ภายใน 1 เมกกะไบต์แรกเท่านั้น MEMW(เฉพาะรุ่น AT)(Memory write) ขาสัญญาณนี้ไม่ใช่สัญญาณ (O)เดียวกันกับสัญญาณ MEMW ใน XT มันจะแอคตีฟก็ในทุก ๆ ขบวน การเก็บข้อมูลที่เกิดขึ้น ไม่ว่าอยู่ในช่วงหน่วยความจำ 1 เมกกะไบต์แรกหรือไม่

IOR (*I/O) (I/O Read) เป็นขาสัญญาณที่บอกให้อุปกรณ์ I/O ที่ต่ออยู่ ทำการส่งข้อมูลลงมาที่ดาต้าบัส (I/O Write) เป็นขาสัญญาณที่บอก IOW (*I/O)ให้อุปกรณ์ I/O ที่ต่ออยู่ ทำการเก็บข้อมูลจากดาต้าบัสเข้าไป RESET DRV(Reset Driver) เป็นขาสัญญาณที่แอคตีฟตอนช่วงที่เราเริ่มจ่าย (O)ไฟให้กับระบบเพื่อใช้ในการรีเซต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CPU และ อุปกรณ์ต่าง ๆ ใน ระบบคอมพิวเตอร์ รวมทั้ง อุปกรณ์ I/O ที่ต่ออยู่ด้วย MEM CS16(เฉพาะรุ่น AT) (Memory 16 Chip Select) เป็นขาสัญญาณที่ (I) ใช้บอกระบบให้ทราบว่าต้องการรับข้อมูลกับหน่วยความจำ ทีละ 16 บิต ถ้าไม่ป้อนสัญญาณนี้ การรับส่งข้อมูลจะทำเหมือนกับ XT คือ การรับส่งข้อมูลทีละ 8 บิต สองครั้ง เพื่อให้ได้ข้อมูลขนาด 16 บิต

I/O CS16 (I) (เฉพาะรุ่น AT)(Memory 16 Chip Select) เป็นขาสัญญาณที่ใช้บอกระบบให้ทราบว่า ต้องการรับส่งข้อมูลกับอุปกรณ์ I/O ทีละ 16 บิต ถ้าไม่ป้อนสัญญาณนี้ การรับส่งข้อมูลจะทำเหมือนกับ XT คือ ทำการรับส่งข้อมูลทีละ 8 บิต สองครั้ง เพื่อให้ได้ข้อมูลขนาด 16 บิต

สัญญาณที่ใช้สร้าง WAIT STATE

I/O Ch RDY (I) (I/O Channel Ready) ขาสัญญาณนี้จะทำให้ถูกแอคตีฟโดยอุปกรณ์ I/O หรือหน่วยความจำที่ไม่สามารถทำงานได้ทันกับระบบ ดังนั้น จะต้องทำการหน่วงระบบให้ทำงานช้าลง ด้วยการเพิ่ม Wait States โดยการทำให้สัญญาณนี้แอคตีฟในช่วงเวลา I/O ได้รับสัญญาณจากการดีโค้ดแอดเดรส,สัญญาณ MEMR, สัญญาณ MEMW, สัญญาณ IOR,สัญญาณ IOW

OWS (I) (เฉพาะรุ่น AT)(Zero Wait State) การแอคตีฟของขาสัญญาณนี้จะบังคับไม่ให้เกิดการสร้าง Wait State โดยอัตโนมัตินั่นคือ การที่จะเกิด Wait State ขึ้นได้จะต้องขึ้นอยู่กับสัญญาณนี้ เช่น การทำงานในขบวนการอ่านเขียนข้อมูลขนาด 16 บิต โดยไม่ใช้ Wait

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

State ทำได้โดยการสร้างสัญญาณ OWS จากสัญญาณการดีโค้ดแอดเดรส และสัญญาณที่ใช้ในการอ่าน หรือเขียน หรือการลด Wait State ในขบวนการอ่านเขียนข้อมูลขนาด 8 บิต ให้เหลือเพียง 2 Wait State ทำได้โดยให้สัญญาณ OWS แอคตีฟหลังจากสัญญาณอ่าน หรือ เขียนไปแล้ว 1 คลอ็ก โดยปกติ การขับสัญญาณนี้ควรใช้เกตที่มีเอ้าท์พุตเป็นแบบ Open Collector ที่ทนกระแสได้ 20 mA(Sinking Current)

<u>สัญญานนาฬิกา</u>

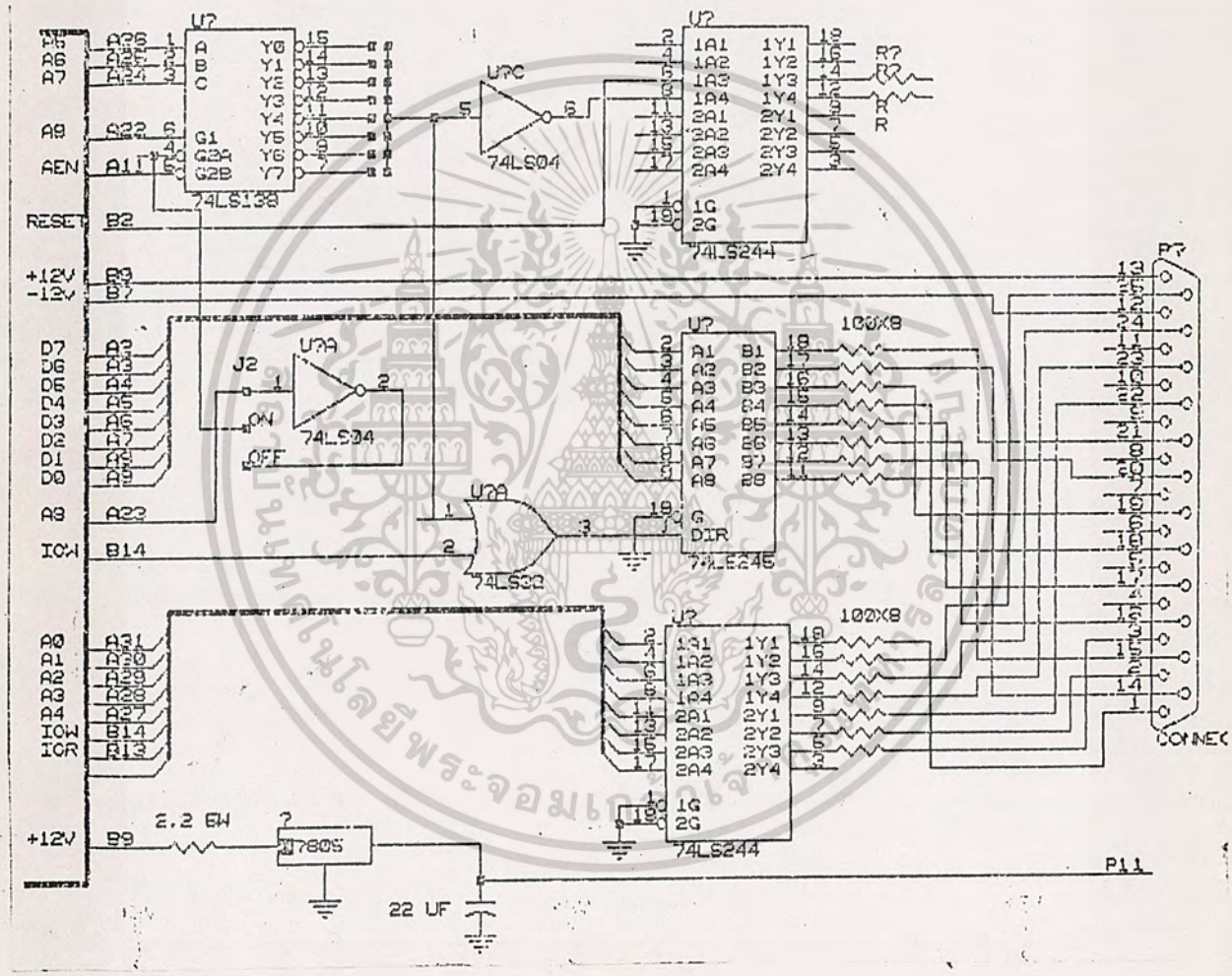
CLK (O) (System Clock) สัญญาณ XT ขาสัญญาณนี้จะมีความถี่ประมาณ 4.77 MHz หรืออาจจะสูงกว่านี้ก็ได้สำหรับรุ่นใหม่ ๆ และสำหรับ AT จะมีความถี่ประมาณ 6 MHz หรือ ในรุ่นใหม่ ๆ อาจจะมีความถี่สูงถึง 15 MHz โดยปกติ ขาสัญญาณ นี้มีดิวตี้ไซเคิล 50% สำหรับ CPU เบอร์ 80286 ตัวกำเนิดสัญญาณนาฬิกาที่ป้อนให้จะมีความถี่เป็น 2 เท่าของความถี่ที่ CPU ทำงาน แต่ขาของสัญญาณนี้ก็จะยังคงมีความถี่เป็น 2 เท่า ของความถี่ที่ CPU ทำงานแต่ขาสัญญาณที่ก็จะยังคงมีความถี่เท่ากับความถี่ที่ CPU ทำงานอยู่เสมอ

OSC (O) (Oscillator)เป็นขาสัญญาณที่มีความถี่สูงคือ 14.31818 MHz ความถี่ของสัญญาณนี้จะคงที่เสมอ และจะไม่ซิงโครนัสกับสัญญาณอื่น ๆในระบบ ดังนั้นจึงไม่ควรนำสัญญาณนี้ไปใช้กับสัญญาณคลอ็กของ อุปกรณ์ I/O ที่ต่ออยู่กับระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IBM PC INTERFACE ของ PAL PROGRAMMABLE CARD

PAL PROGRAMMABLE CARD ได้ออกแบบส่วน INTERFACE IBM PC ดังวงจรรูปที่ 11 โดย LS138 จะทำหน้าที่ DECODE ADDRESS จาก PIN A5, A6, A7, A8, A9 ส่วน LS245 จะทำหน้าที่เป็น BUFFER DATA DO-D7 ไปยัง DB25 และ LS244 จะเป็น BUFFER สัญญาณต่างๆจาก SLOT ที่นำไปใช้ใน PAL PROGRAMMABLE CARD

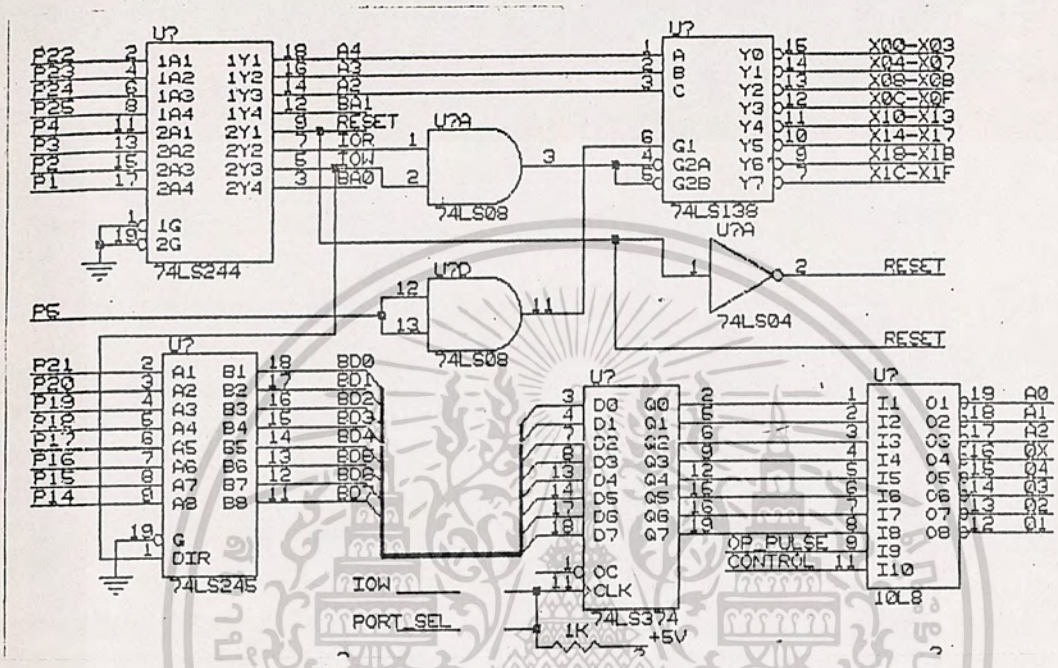


รูปที่ 11 วงจร INTERFACE กับ IBM PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 BUS RECIEVER AND ADDRESS DECODER

วงจรรูปที่ 12 จะเป็นส่วนที่ขยายสัญญาณจาก DB25 เพื่อนำไปใช้ในส่วนต่างๆของวงจร และ LS138 จะทำหน้าที่ DECODE PORT ที่ใช้เป็น CHIP SELECT สำหรับอุปกรณ์ต่างๆ ตามตาราง



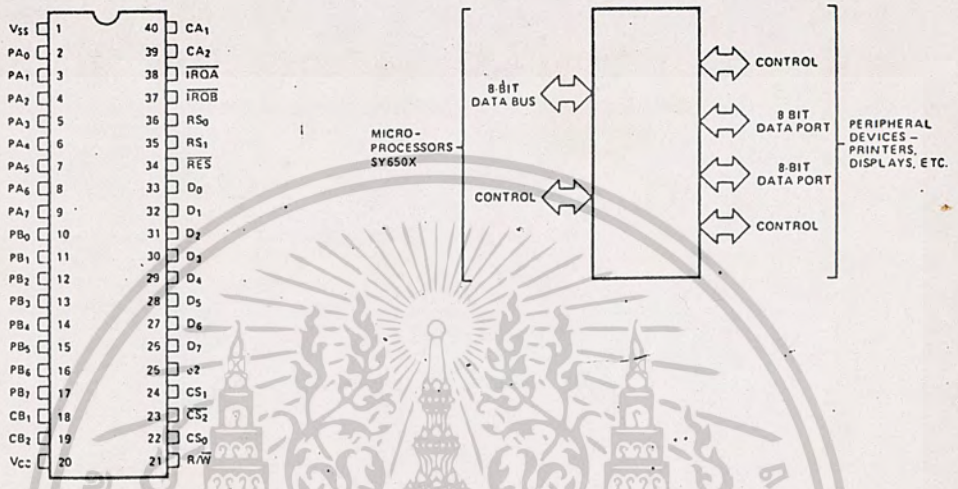
รูปที่ 12 วงจร BUS RECIEVER AND ADDRESS DECODER

PORT NUMBER	DEVICE SELECT
280-283	MC 6821 #1
284-287	MC 6821 #2
288-28B	MC 6821 #3
28C-28F	74LS374

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การใช้งาน MC 6821 (PIA)

MC 6821 Peripheral Interface Adapter (PIA) จะมี PORT 8 bit Bi-directional 2 I/O Port และสามารถทำการ Handshake ในขณะที่เคลื่อนย้ายข้อมูล โดยมีตำแหน่งขาและการ Interface เบื้องต้นดังรูปที่ 13



รูปที่ 13 Pin Configuration and Basic interface

รายละเอียดขาต่างๆของ MC6821

- RES (Reset) เป็นสัญญาณที่ใช้ Initialize PIA ถ้าขานี้เป็น LOW Register ภายในจะถูก clear

2 (Input Clock) เป็น clock ของระบบ และใช้ triggerในการเคลื่อนย้ายข้อมูลระหว่าง microprocessor กับ PIA

R/-W (Read/Write) สัญญาณนี้เกิดจาก Microprocessor ใช้สำหรับควบคุมทิศทางของข้อมูล

-IRQA,-IRQB (Interrupt Requests) เป็นขา Output ซึ่ง PIA เป็นตัวก่าเนิดขึ้นในกรณีที่ต้องการ Interrupt โดยจะ Active low

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D0-D7 (Data Bus) เป็น Bi-directional Data Bus ใช้สำหรับ transfer ข้อมูลระหว่าง Microprocessor กับ PIA

CS0,CS1,-CS2 (Chip Selects) PIA จะถูกเลือกให้ทำงานเมื่อ CS0 และ CS1 เป็น high และ -CS2 เป็น low

RS0,RS1 (Register Selects) เป็นสัญญาณที่ใช้เลือก Register ภายใน PIA

รูปที่ 14 จะแสดง Block Diagram ของ MC6821 โดยจะประกอบด้วย Control Register (CRA,CRB), Data Direction Register (DDRA, DDRB),Output Register (ORA,ORB),Interrupt Status Control และ Peripheral Interface Buffer

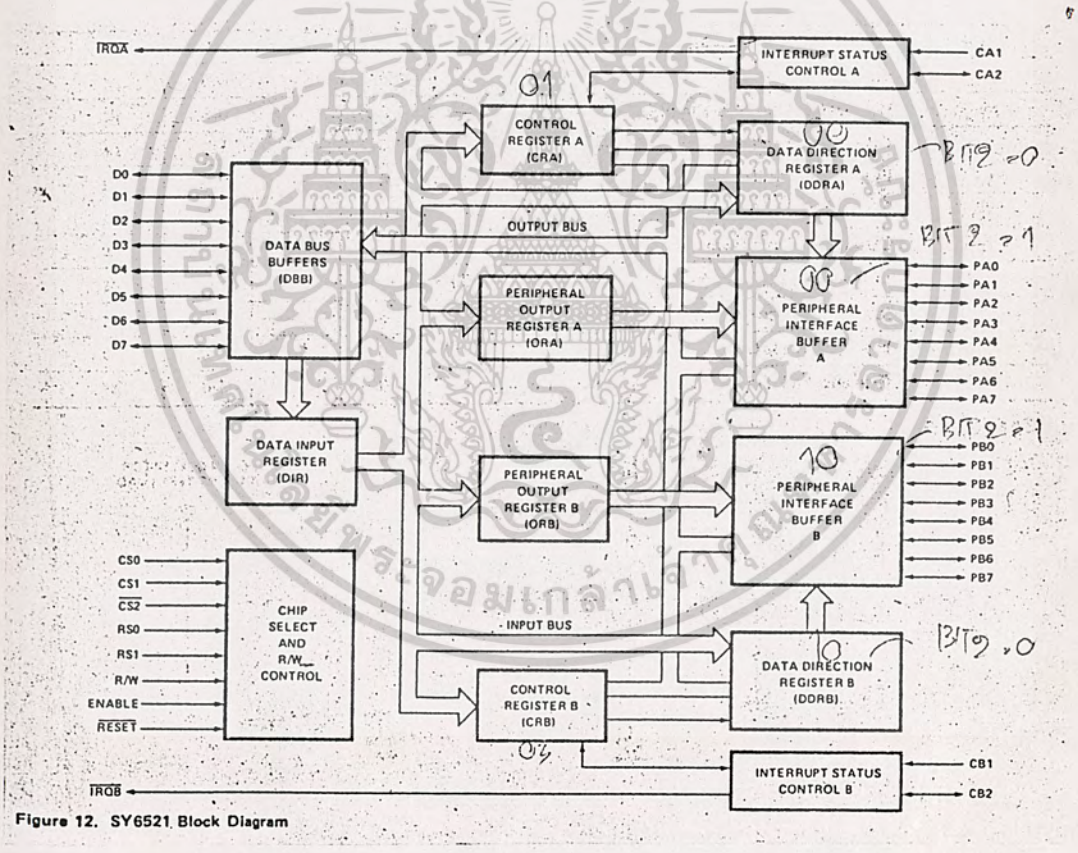


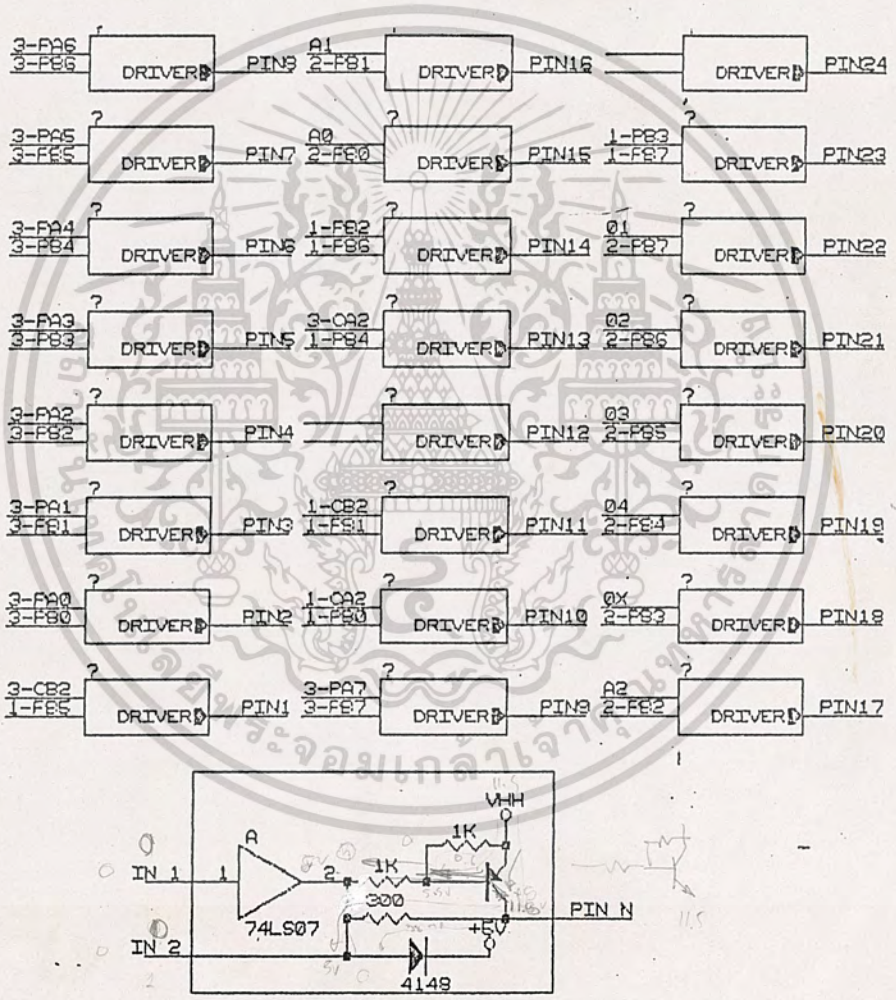
Figure 12. SY6521 Block Diagram

รูปที่ 14 Block Diagram ของ MC 6821

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 VOLTAGE DRIVER FOR INPUT PIN

VOLTAGE DRIVER ดังรูปที่ 15 เป็นส่วนที่ต้องสร้างแรงดันให้ได้ 3 ระดับคือ VL(0V),VH(5V) และ VHH(11.5V) ซึ่งในแต่ละขา จะต้องใช้ DATA 2 BIT ในการ CONTROL โดยสร้างจาก OUTPUT PORT ของ MC6821#3 ทั้ง PA และ PB ซึ่งจะต้องให้ได้ DATA ตามตารางข้างล่าง เพื่อให้ได้แรงดันตามต้องการ



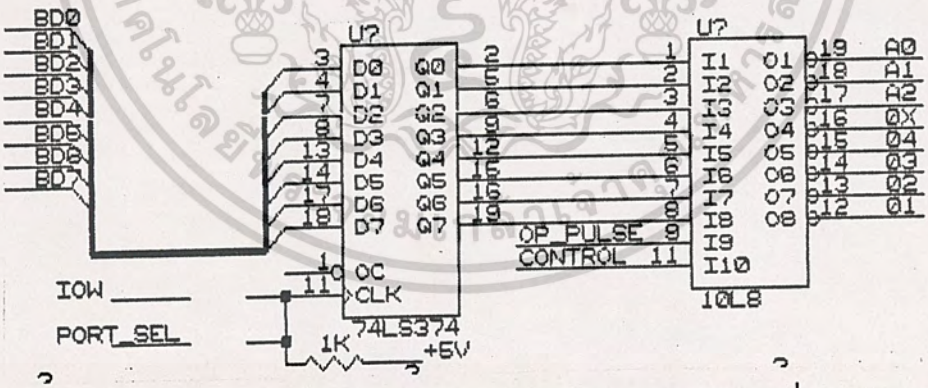
รูปที่ 15 วงจร PIN Driver และ Output Port ของแต่ละ Pin

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PA	PB	O/P PIN
0	0	0 V
0	1	5 V
1	0	11.5 V
1	1	11.5 V

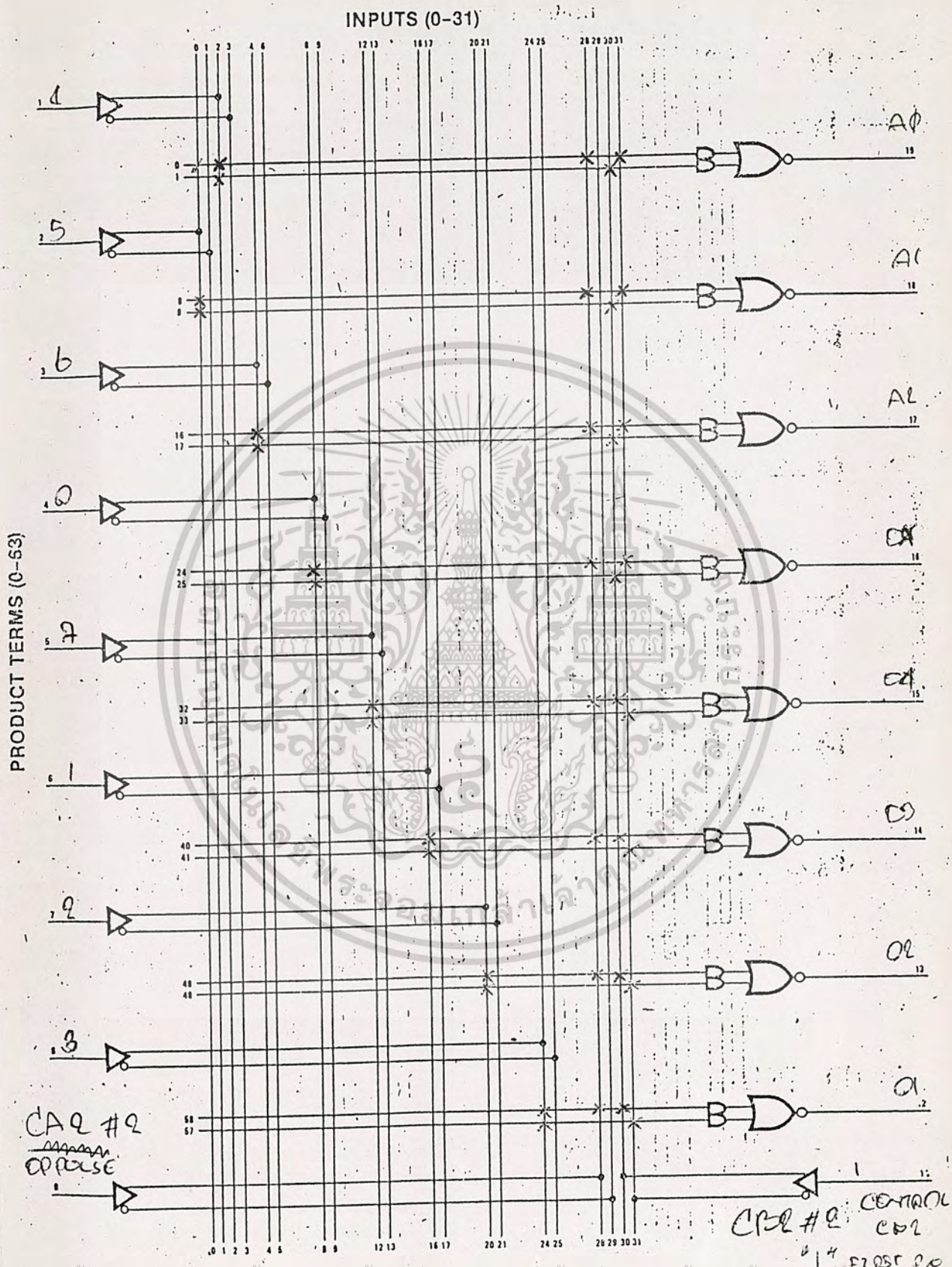
3.5 VOLTAGE DRIVER FOR OUTPUT PIN

VOLTAGE DRIVER FOR OUTPUT PIN จะมีหลักการทำงานเหมือนในส่วนของ INPUT PIN แต่ในส่วนของ OUTPUT PIN จะมี PAL 10L8 ช่วยในการ control ดังรูปที่ 16 เนื่องจาก OUTPUT PIN อยู่ในส่วนของ PRODUCT LINE ซึ่งมี 64 ตำแหน่งและการ BLOW ต้องแบ่งเป็น 2 ช่วงคือ ช่วงต่ำ (0-31) และช่วงสูง (31-63) และยังต้องใช้ MONOSTABLE ป้อน PLUSE ขนาด 20-50 uS ด้วย



ในส่วนของ PAL10L8 จะมีการโปรแกรมภายในดังรูปที่ 17 โดยออกแบบให้ PIN 11 เป็นขา CONTROL สำหรับ SELECT PAGE ของPRODUCT LINE โดยถ้าเป็น "0" จะเป็น LO PAGE และถ้าเป็น "1" จะเป็น HI PAGE ส่วน PIN 9 จะทำหน้าที่เป็น OP PULSE ซึ่งเป็น PULSE 20-50 uS โดยกำเนิดจาก IC 555 monostable สำหรับ BLOW FUSE ใน PAL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.6 MONOSTABLE โดยใช้ IC 555

วงจร MONOSTABLE ช่วงเวลาของพัลส์จะถูกกำหนดด้วย เวลาที่ใช้ในการประจุตัวเก็บประจุจาก 0 V ถึง 2/3 V ของแรงดันไฟเลี้ยง (แรงดันเทรสโฮลด์) โดยทั่วไปแรงดันประจุ C (Vc) จะถูกประจุผ่าน R จากแหล่งจ่ายไฟ VCC จนมีแรงดันเท่ากับ 2/3 VCC

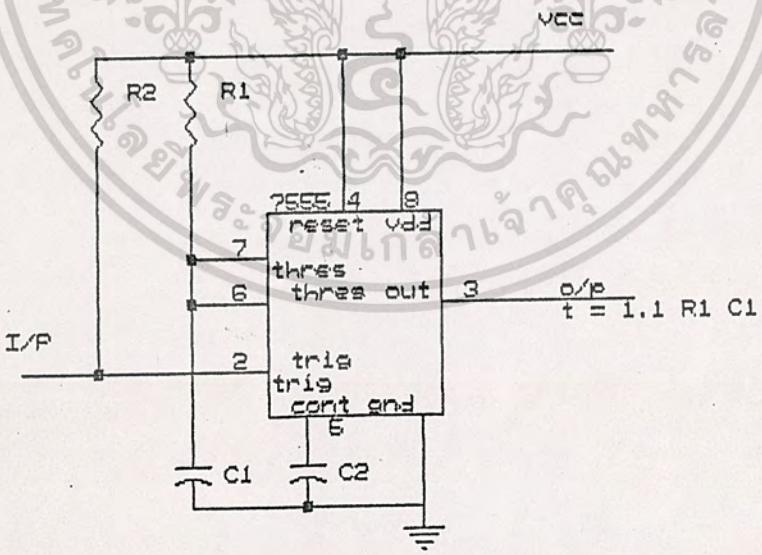
$$\text{เมื่อ} \quad Vc\ (t) = VCC\ (1-e^{-t/RC})$$

$$t = (-LOG\ e^{1/3})\ RC$$

$$= 1.1\ RC$$

แรงดันที่ถูกประจุนี้เป็นตัวกำหนด MONOTIME และ PLUSE ที่มากระตุ้นก็ต้องสั้นกว่า MONOTIME

ข้อดีของ IC 555 ในการทำวงจร MONOSTABLE คือ ช่วงเวลาของ PULSE จะไม่ขึ้นอยู่กับแรงดัน VCC ซึ่งเมื่อ VCC ลดลง จะทำให้แรงดัน TRIGGER และ THESHOLD ลดลงตามมาด้วย ดังนั้นช่วงเวลาของ PLUSE ก็จะไม่เปลี่ยนตามแรงดัน VCC

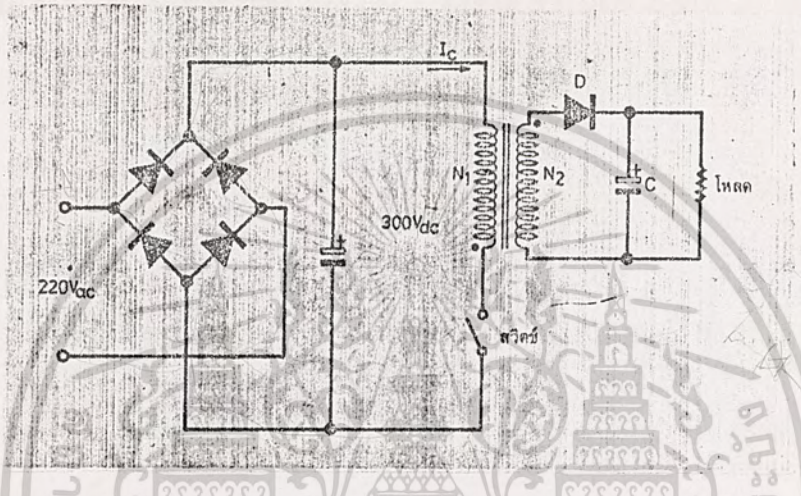


รูปที่ 18 วงจร Monostable ที่กำเนิด Pulse ขนาด 30 uS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.7 SWITCHING POWER SUPPLY

SWITCHING POWER SUPPLY จะสวิตช์ไฟจาก 220 V เลย ซึ่งเรียกย่อๆว่า SMPS (Switch Mode Power Supply) ข้อดีของ SMPS คือ มีประสิทธิภาพสูง ขนาดเล็ก และ น้ำหนักเบา ยิ่งใช้ความถี่ในการสวิตช์สูงเท่าใดก็มีขนาดเล็กลงเท่านั้น รูปแบบของ SMPS มีหลายแบบเช่น flyback, forward, half bridge และ full bridge



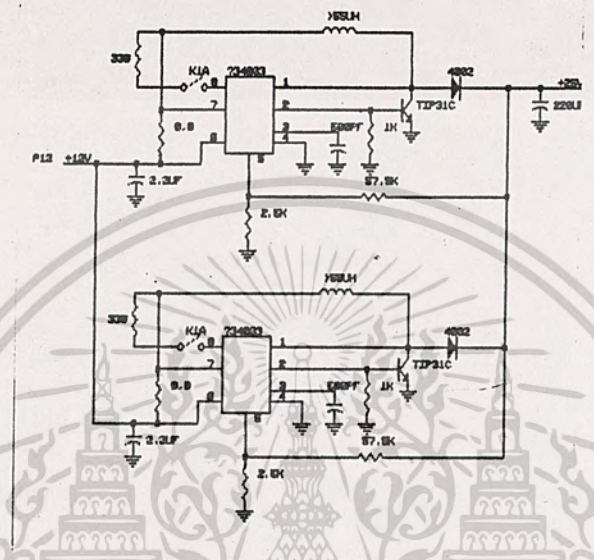
รูปที่ 19 หลักการเบื้องต้น Switching Power Supply

SMPS แบบ fly back มีหลักการทำงานคือ จากรูปที่ 19 ในขณะที่สวิตช์ ON พลังงานจะถูกเก็บไว้ในขดลวด โดยกระแส Ic จะค่อยๆเพิ่มขึ้น โปรดสังเกตุที่ขั้วของขดลวดด้วยเพราะขณะที่สวิตช์ ON แรงดันทาง N2 จะไบอัสกลับแก่ไดโอด D ทำให้ไม่มีกระแสไหล ไม่มีพลังงานผ่านไปที่ตัวเก็บประจุ C และโหลดในช่วงเวลานี้ เมื่อสวิตช์ OFF ขดลวดจะคายพลังงานกลับออกมา แรงดันที่ขดทั้งสองจะกลับขั้วกัน ทำให้กระแสไหลผ่านไดโอด D ไปประจุ C และผ่านโหลด ส่วนที่ N1 แรงดันจากขด N1 จะเสริมกับแรงดัน 300 Vdc เดิม ทำให้แรงดันที่ตกคร่อมสวิตช์มีค่าสูงมาก ประมาณ 700-800 V พลังงานที่ส่งไปจากการ ON และ OFF แต่ละครั้งขึ้นอยู่กับค่าเหนี่ยวนำของขดลวด และ Ic ซึ่งโหลดจะได้กำลังงานมากน้อย ขึ้นอยู่กับจำนวนครั้งในการส่งพลังงานต่อเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่าสวิตช์ตัวนี้ที่จริงก็คือ TRANSISTOR นั่นเองแต่ต้องเป็นสวิตชิงที่ทนแรงดันได้สูง และต้องทำงานได้เร็วมาก

ใน PAL PROGRAMMABLE CARD ได้ออกแบบวงจร SWITCHING ดังรูปที่ 20 ซึ่งจะ Switch ไฟจาก 12V เป็น 25V 1A



รูปที่ 20 Switching Power Supply

3.8 PROGRAMMABLE VOLTAGE POWER SUPPLY

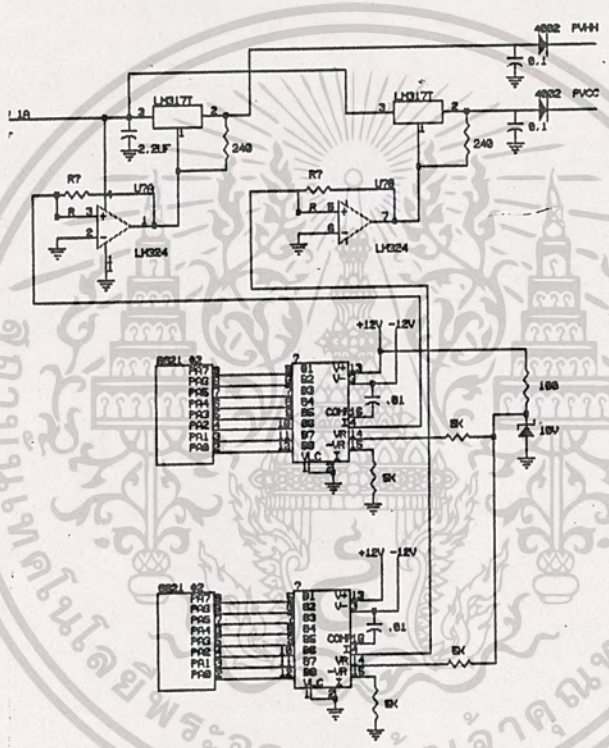
PROGRAMMABLE VOLTAGE จะกำเนิดแรงดัน 2 ค่า คือ PVHH และ PVCC โดย PVHH ใช้จ่ายแรงดันให้ขา E ของ TRANSISTOR ซึ่งต้องปรับค่าได้ เนื่องจาก PAL ของแต่ละบริษัทจะใช้แรงดันไม่เหมือนกันเช่น NS ใช้ 11.5V ส่วน TI ใช้ 12V ส่วน PVCC จะเป็นแรงดัน VCC ซึ่งต้องปรับได้จาก 11.5V เป็น 6V 5V, 4.5V เพื่อที่จะใช้ในการ VERIFY

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การปรับค่าแรงดันนี้จะใช้ PORT PA ของ MC 6821 #1 และ #2 แล้วใช้ DAC แปลงสัญญาณ DIGITAL เป็น ANALOG ผ่าน LM324 แล้วใช้ LM317T เป็นตัว REGULATOR โดยสามารถปรับแรงดันได้จากสูตร

$$Vout = 1.25\ V\ (\ 1+R1/R2\ )$$

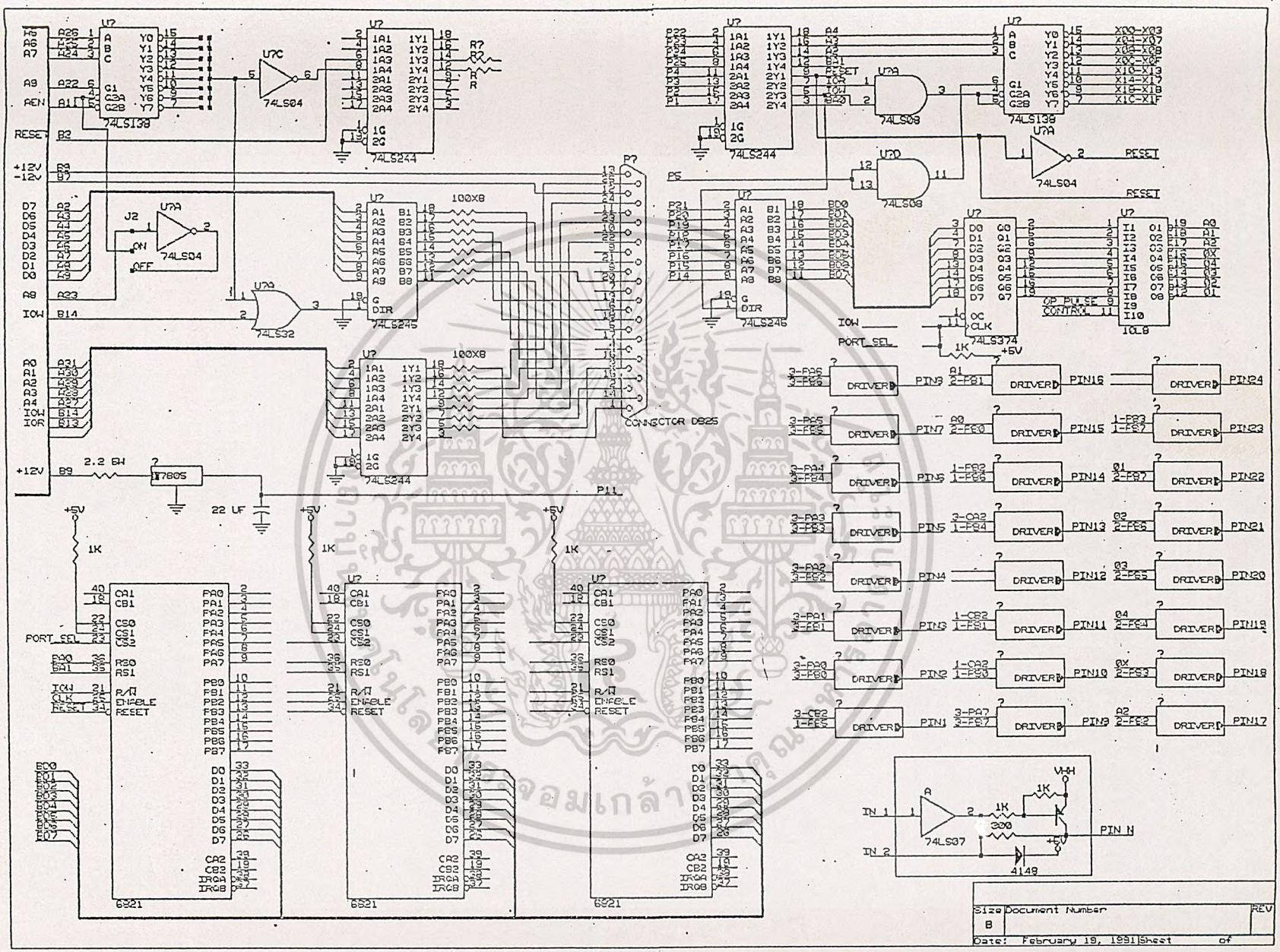
รูปที่ 21 จะกำเนิดไฟ PVHH กับ PVCC ซึ่งสามารถโปรแกรมให้ได้แรงดันตามต้องการ โดยส่งข้อมูลทาง Port PA ของ MC6821 ผ่าน DAC 0800 ซึ่งแปลงสัญญาณ Digital เป็น Analog

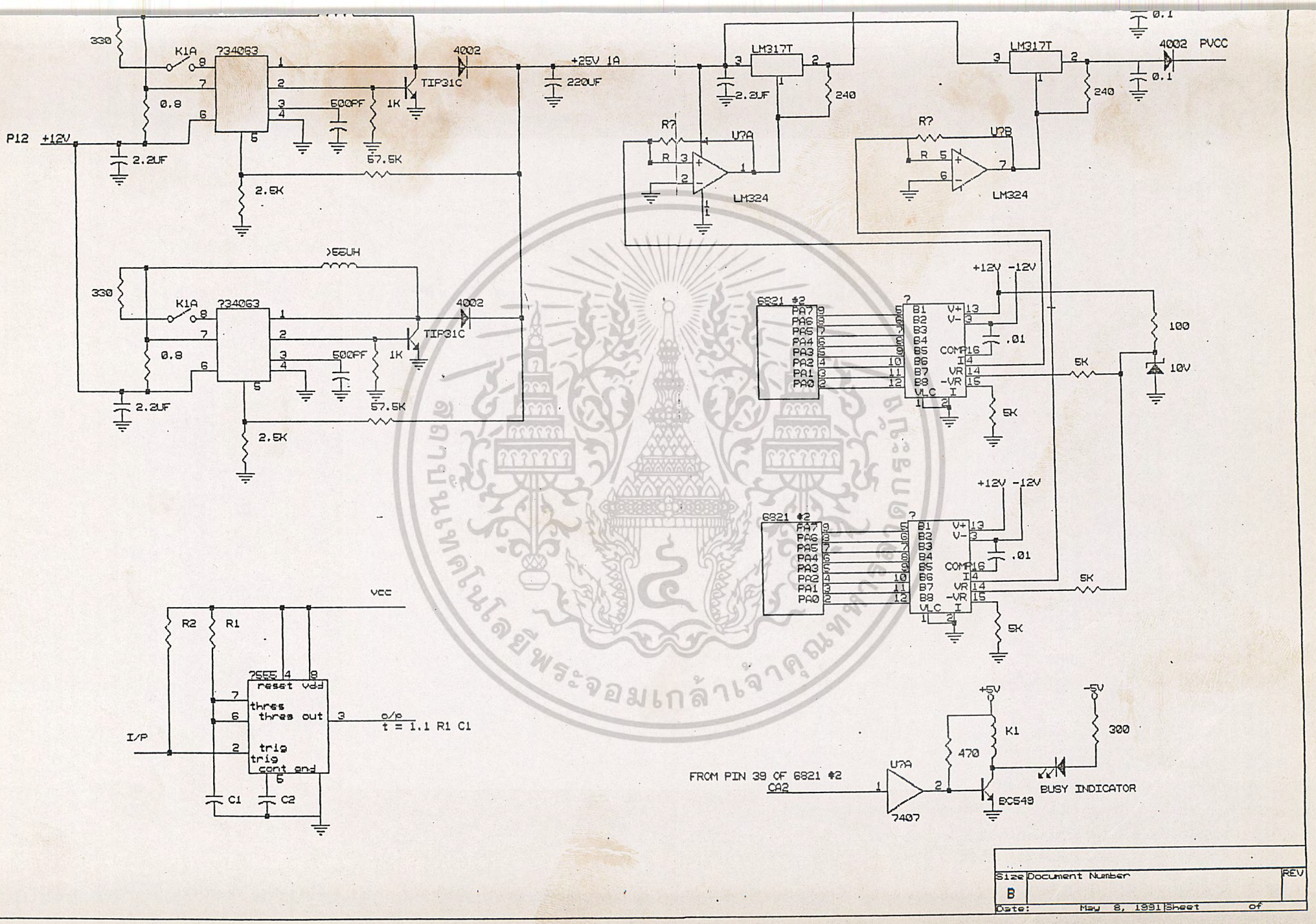


รูปที่ 21 Programmable Voltage

รูปที่ 22 และ รูปที่ 23 จะแสดงรายละเอียดของวงจรทั้งหมดใน PAL Programmable Card

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


74LS138
74LS04
74LS244
74LS245
74LS32
74LS08
74LS374
74LS37
7805
2.2 KH
22 UF
100X8
CONNECTOR DB25
DRIVER
1K
+5V
+12V
-12V
RESET
RESET1
PORT_SEL
IOW
IOR
6821



ภาควิชาเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

Size	Document Number	REV
B		
Date:	May 8, 1991	Sheet of

```
;-------------------------------------------------;
;         PROGRAM OF PAL PROGRAMMABLE CARD        ;
;                                                 ;
;       PRODUCE BY  Mr.YOSAPONG TACHAPORNSIN      ;
;             FACULTY OF ENGINEERING              ;
; KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG ;
;-------------------------------------------------;

DATA          SEGMENT word public 'DATA'

EX00        DB   0
EX01        DB   0
MFG_TYPE    DB   0
CUR_INPLIN  DB   0
CUR_PROLIN  DB   0
PALNUM      DB   0
PAL_PG      DB   0
EX07        DB   0
EX08        DB   0
EX09        DB   0
EX0A        DW   0
EX0D        DW   0
CURPRO_LIN  DB   0
JOB_STATUS  DB   0
PAL_TYPE    DB   0
PG_PROLIN   DB   0
EX14        DB   0
EX15        DW   0
TATINP_LIN  DB   0
EX18        DW   0
EX1A        DB   0
EX1B        DB   0
EX1C        DB   0
PAL_FUSE    DB   3200  dup (?)
            ORG  0CECh
DAT_BUF     DB   0CECh dup (?)

DATA        ENDS

CODE            SEGMENT byte public 'CODE'
                ASSUME  CS:CODE,DS:DATA

                ORG     100h

MAIN_PRO      PROC        NEAR
      PUSH    DS
      MOV     AX,0000h
      PUSH    AX
      MOV     AX,DATA
      MOV     DS,AX
      MOV     ES,AX
      CALL    WR_INITIAL
SET_TYPE:
      CALL    SET_ZERO
      MOV     CX,0000h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CALL    CLRSCR
        MOV     SI,OFFSET PAGE1U
        CALL    WRITE_ST
        MOV     SI,OFFSET PAGE1D
        CALL    WRITE_ST
        CALL    SOUND
PAL_NUM:
        MOV     AH,01h
        INT     21h
        CMP     AL,31h                  ;'1'
        JL      N_PAL
        CMP     AL,3Ah                  ;':'
        JL      Y_PAL
        AND     AL,5Fh                  ;'_'
        SUB     AL,07h
        CMP     AL,3Ah                  ;':'
        JL      N_PAL
        CMP     AL,4Eh                  ;'N'
        JL      Y_PAL
N_PAL:
        MOV     AL,08h
        CALL    WRITE_CHAR
        CALL    SOUND
        JMP     PAL_NUM
Y_PAL:
        SUB     AL,30h                  ;'0'
        DEC     AL
        MOV     [PALNUM],AL
        MOV     Byte Ptr [PAL_TYPE],00
        CMP     AL,0Fh
        JL      HEX
        INC     Byte Ptr [PAL_TYPE]
HEX:
        MOV     Byte Ptr [EX14],08h
        MOV     Word Ptr [EX15],0004h
        MOV     Byte Ptr [TATINP_LIN],20h       ;' '
        MOV     Word Ptr [EX18],007Ch
        CMP     Byte Ptr [PAL_TYPE],00h
        JZ      RD_CURSOR
        MOV     Byte Ptr [EX14],0Ah
        MOV     Word Ptr [EX15],0005h
        MOV     Byte Ptr [TATINP_LIN],28h       ;'('
        MOV     Word Ptr [EX18],00C3h
RD_CURSOR:
        MOV     AH,03h
        MOV     BH,00h
        INT     10h
        PUSH    DX
        MOV     DX,0140h
        CALL    GOTO_XY
        MOV     AL,[PALNUM]
        MOV     AH,00
        MOV     CL,0Fh
        MUL     CL
        MOV     BX,AX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     SI,OFFSET DISP_NUM
        ADD     SI,BX
        CALL    WRITE_ST
        POP     DX
        CALL    GOTO_XY
        MOV     SI,OFFSET SEL_MFG
        CALL    WRITE_ST
GET_MFG:
        MOV     AH,01h
        INT     21h
        CMP     AL,31h                  ;'1'
        JL      N_MFG
        CMP     AL,34h                  ;'4'
        JGE     N_MFG
        SUB     AL,31h                  ;'1'
        MOV     [MFG_TYPE],AL
       ; CALL    INIT_MFG
        JMP     Y_MFG
        NOP
N_MFG:
        MOV     AL,08h
        CALL    WRITE_CHAR
        JMP     GET_MFG
Y_MFG:
        MOV     AH,03h
        MOV     BH,00h
        INT     10h
        MOV     DX,0040h
        CALL    GOTO_XY
        MOV     BL,[MFG_TYPE]
        SHL     BL,1
        SHL     BL,1
        MOV     SI,OFFSET DISP_MFG
        MOV     BH,00
        ADD     SI,BX
        CALL    WRITE_ST
CLR05:
        MOV     CX,0500h
        CALL    CLRSCR
        MOV     SI,OFFSET DISP_MENU
        CALL    WRITE_ST
        CALL    CHECK_SUM
        JMP     INIT_M
TEST_KEY:
        CALL    READ_KEY
        PUSH    AX
        CALL    CLR_LINE
        POP     AX
WRITE_:
        CMP     AL,57h                  ;'W'
        JNZ     READ_
        CALL    MAIN_WRITE
        JMP     CLR11
READ_:
COMMENT*
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CMP     AL,52h                  ;'R'
        JNZ     SECURITY
        CALL    MAIN_READ
        JB      N_SEC
        MOV     SI,0AD5
        CALL    WRITE_ST
        CALL    CHECK_SUM
        JMP     INIT_M
SECURITY:
        CMP     AL,53h                  ;'S'
        JNZ     COMPARE
        MOV     SI,07B2h
        CALL    WRITE_ST
        CALL    READ_KEY
        CMP     AL,59h                  ;'Y'
        JNZ     N_SEC
        CALL    MAIN_SECURITY
N_SEC:
        JMP     INIT_M
COMPARE:
        CMP     AL,43h                  ;'C'
        JNZ     BLANK
        CALL    MAIN_COMPARE
        JMP     INIT_M
BLANK:
        CMP     AL,42h                  ;'B'
        JNZ     INPUT
        CALL    MAIN_BLANK
        JMP     INIT_M
INPUT:
        CMP     AL,49h                  ;'I'
        JNZ     OUTPUT
        CALL    MAIN_INPUT
        CALL    CHECK_SUM
        JMP     INIT_M
OUTPUT:
        CMP     AL,4Fh                  ;'O'
        JNZ     EDIT
        CALL    MAIN_OUTPUT
        JMP     INIT_M
EDIT:
        CMP     AL,45h                  ;'E'
        JNZ     TYPE_PAL
        MOV     CX,1500
        CALL    CLRSCR
        CALL    CHECK
        JB      INIT_M
        CALL    MAIN_EDIT
        JMP     CLR05
TYPE_PAL:
        CMP     AL,54h                  ;'T'
        JNZ     N_TYPE
        JMP     SET_TYPE
N_TYPE:
        CMP     AL,00h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
     JNZ     QUIT
     CALL    READ_KEY
     JMP     CLR11
QUIT:
     CMP     AL,51h                    ;'Q'
     JNZ     CLR11
     MOV     CX,0000
     CALL    CLRSCR
     MOV     AH,4Ch
     INT     21h
CLR11:
     MOV     CX,1100h
     CALL    CLRSCR
INIT_M:
     MOV     DX,1100h
     CALL    GOTO_XY
     MOV     SI,OFFSET SEL_MENU
     CALL    WRITE_ST
     MOV     AL,20h                    ;' '
     CALL    WRITE_CHAR
     MOV     AL,08h
     CALL    WRITE_CHAR
     JMP     TEST_KEY
MAIN_PRO     ENDP

CLR_LINE     PROC     NEAR
        MOV        CX,1100h
        INC        CH
        CALL       CLRSCR
        RET
CLR_LINE     ENDP

PAGE1U  DB  "PAL PROGRAMMABLE CARD !   FOR  20/24 SERIES  "
        DB  "       ¦    MFG. : ",13,10
        DB  "              BY  KMIT'L                    "
        DB  "       ¦ PAL TYPE : ",13,10
        DB  "          FACULTY OF ENGINEERING            "
        DB  "       ¦ FUSE MAP : ",13,10,10
        DB  "¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦"
        DB  "¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦¦",13,10,00

PAGE1D  DB  ' 20 PIN PAL SERIES ',13,10
        DB  ' (1) = 10H8          (2) = 12H6          (3) = 14H4 ',13,10
        DB  ' (4) = 16H2          (5) = 16C1          (6) = 10L8 ',13,10
        DB  ' (7) = 12L6          (8) = 14L4          (9) = 16L2 ',13,10
        DB  ' (A) = 16L8          (B) = 16R8          (C) = 16R6 ',13,10
        DB  ' (D) = 16R4          (E) = 16X4          (F) = 16A4 ',13,10,10
        DB  ' 24 PIN PAL SERIES ',13,10
        DB  ' (G) = 12L10         (H) = 14L8          (I) = 16L6 ',13,10
        DB  ' (J) = 18L4          (K) = 20L2          (L) = 20C1 ',13,10
        DB  ' (M) = 20L10         (N) = 20X10         (O) = 20X8 ',13,10
        DB  ' (P) = 20X4          (Q) = 20L8          (R) = 20R8 ',13,10
        DB  ' (S) = 20R6          (T) = 20R4 ',13,10,10
        DB  ' PAL  TYPE  = ?   ',00
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DISP_NUM   DB  '10H8  (20 PIN)',00,'12H6  (20 PIN)',00
           DB  '14H4  (20 PIN)',00,'16H2  (20 PIN)',00
           DB  '16C1  (20 PIN)',00,'10L8  (20 PIN)',00
           DB  '12L6  (20 PIN)',00,'14L4  (20 PIN)',00
           DB  '16L2  (20 PIN)',00,'16L8  (20 PIN)',00
           DB  '16R8  (20 PIN)',00,'16R6  (20 PIN)',00
           DB  '16R4  (20 PIN)',00,'16X4  (20 PIN)',00
           DB  '16A4  (20 PIN)',00
           DB  '12L10 (24 PIN)',00,'14L8  (24 PIN)',00
           DB  '16L6  (24 PIN)',00,'18L4  (24 PIN)',00
           DB  '20L2  (24 PIN)',00,'20C1  (24 PIN)',00
           DB  '20L10 (24 PIN)',00,'20X10 (24 PIN)',00
           DB  '20X8  (24 PIN)',00,'20X4  (24 PIN)',00
           DB  '20L8  (24 PIN)',00,'20R8  (24 PIN)',00
           DB  '20R6  (24 PIN)',00,'20R4  (24 PIN)',00

SEL_MFG    DB  13,10,' PAL MFG. (1.=MMI,2.=NS,3.=TI) ',13,10
           DB  ' MFG. = ? ',00

DISP_MFG   DB  'MMI',00,'NS ',00,'TI ',00

SUM_ST     DB  'CHECK SUM : ',00

CK_READY       DB  'ARE YOU READY ? <Y/N> ',00
CK_WR_SEC      DB  13,10,'WRITE SECURITY FUSE ? <Y/N> ',00
WR_CON_MSG     DB  'WRITE CONTINUE ? <Y/N> ',00
WR_ERR_MSG     DB  'PAL WRITE ERROR !!! ',00
WR_NEXT_MSG    DB  'WRITE NEXT PAL ? <Y/N> ',00
COMPLETE_MSG   DB  'WRITE COMPLETE !! ',00
CKBLANK_MSG    DB  'PAL BLANK CHECK ',00
OK_MSG         DB  'OK ! ',00
ERROR_MSG      DB  'ERROR ! ',00
COMPARE_MSG    DB  'PAL COMPARE ',00
OUTPUT_MSG     DB  'OUTPUT BUFFER TO DISK FILE',00
OUTERR_MSG     DB  'OUTPUT ERROR ! ',00
OUTOK_MSG      DB  'OUTPUT OK ! ',00
FILEERR_MSG    DB  'CANNOT OPEN FILE ! ',00
FORMATFI_MSG   DB  'FUSE MAP PRODUCED BY KMITL',13,10,'L0000',13,10,00
INPUT_MSG      DB  'INPUT DISK FILE TO BUFFER',00
RDERR_MSG      DB  'READ DISK FILE ERROR !',00
RDOK_MSG       DB  'READ DISK FILE OK !',00
FIERR_MSG      DB  'CANNOT OPEN FILE !',00
GETNAME_MSG    DB  13,10,'FILE NAME = ',00

DISP_MENU  DB  '***************MAIN  MENU************  ',13,10
           DB  '  W   :-  WRITE                       ',13,10
           DB  '  R   :-  READ                        ',13,10
           DB  '  S   :-  SECURITY                    ',13,10
           DB  '  C   :-  COMPARE                     ',13,10
           DB  '  B   :-  BLANK                       ',13,10
           DB  '  I   :-  INPUT                       ',13,10
           DB  '  O   :-  OUTPUT                      ',13,10
           DB  '  E   :-  EDIT                        ',13,10
           DB  '  T   :-  TYPE                        ',13,10
           DB  '  Q   :-  QUIT                        ',13,10,00
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SEL_MENU   DB  '  SELECT  MENU   := ',00

BUF_EMPTY  DB  ' Buffer empty ! ',00

DISP_INP_LINE DB '   0000 0000 0011 1111 1111 2222 2222 2233 3333 3333',13,10
              DB '   0123 4567 8901 2345 6789 0123 4567 8901 2345 6789',00

DISP_TITLE_E  DB 'FUSE MAP BUFFER EDITER',00

DISP_MENU_E   DB '    x   :=  UNBLOWN ',01
              DB '    -   :=  BLOWN ',01,01
              DB '<PgUp>  :=  Page Up',01
              DB '<PgDn>  :=  Page Down',01
              DB '<UpKey> :=  Move Up',01
              DB '<DnKey> :=  Move Down',01
              DB '<RtKey> :=  Move Right',01
              DB '<LtKey> :=  Move Left',01
              DB '<ESC>   :=  Main Menu',00

PRO_LINE      DB  00h,01h,02h,03h,04h,05h,06h,07h,08h,09h
              DB  10h,11h,12h,13h,14h,15h,16h,17h,18h,19h
              DB  20h,21h,22h,23h,24h,25h,26h,27h,28h,29h
              DB  30h,31h,32h,33h,34h,35h,36h,37h,38h,39h
              DB  40h,41h,42h,43h,44h,45h,46h,47h,48h,49h
              DB  50h,51h,52h,53h,54h,55h,56h,57h,58h,59h
              DB  60h,61h,62h,63h,64h,65h,66h,67h,68h,69h
              DB  70h,71h,72h,73h,74h,75h,76h,77h,78h,79h

SEND_CRLF  DB   10,13,00

PROD_PATTERN  DB  03h,03h,03h,03h,03h,03h,03h,03h,00h,00h
              DB  00h,0Fh,03h,03h,03h,03h,0Fh,00h,00h,00h
              DB  00h,00h,0Fh,0Fh,0Fh,0Fh,00h,00h,00h,00h
              DB  00h,00h,00h,0FFh,0FFh,00h,00h,00h,00h,00h
              DB  00h,00h,00h,0FFh,0FFh,00h,00h,00h,00h,00h
              DB  03h,03h,03h,03h,03h,03h,03h,03h,00h,00h
              DB  00h,0Fh,03h,03h,03h,03h,0Fh,00h,00h,00h
              DB  00h,00h,0Fh,0Fh,0Fh,0Fh,00h,00h,00h,00h
              DB  00h,00h,00h,0FFh,0FFh,00h,00h,00h,00h,00h
              DB  0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,00h,00h
              DB  0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,00h,00h
              DB  0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,00h,00h
              DB  0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,00h,00h
              DB  0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,00h,00h
              DB  0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,00h,00h
              DB  03h,03h,03h,03h,03h,03h,03h,03h,03h,03h
              DB  00h,0Fh,03h,03h,03h,03h,03h,03h,0Fh,00h
              DB  00h,00h,0Fh,0Fh,03h,03h,0Fh,0Fh,00h,00h
              DB  00h,00h,00h,3Fh,0Fh,0Fh,3Fh,00h,00h,00h
              DB  00h,00h,00h,00h,0FFh,0FFh,00h,00h,00h,00h
              DB  00h,00h,00h,00h,0FFh,0FFh,00h,00h,00h,00h
              DB  0Fh,0Fh,0Fh,0Fh,0Fh,0Fh,0Fh,0Fh,0Fh,0Fh
              DB  0Fh,0Fh,0Fh,0Fh,0Fh,0Fh,0Fh,0Fh,0Fh,0Fh
              DB  0Fh,0Fh,0Fh,0Fh,0Fh,0Fh,0Fh,0Fh,0Fh,0Fh
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DB  0Fh,0Fh,0Fh,0Fh,0Fh,0Fh,0Fh,0Fh,0Fh,0Fh
                DB  00h,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,00h
                DB  00h,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,00h
                DB  00h,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,00h
                DB  00h,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,00h

INP_PATTERN     DB   3Fh, 33h, 33h,0F3h, 00h,0FFh, 33h, 33h,0FFh, 00h
                DB  0FFh, 3Fh,0F3h,0FFh, 00h,0FFh,0FFh,0FFh,0FFh, 00h
                DB  0FFh,0FFh,0FFh,0FFh, 00h, 3Fh, 33h, 33h,0F3h, 00h
                DB  0FFh, 33h, 33h,0FFh, 00h,0FFh, 3Fh,0F3h,0FFh, 00h
                DB  0FFh,0FFh,0FFh,0FFh, 00h,0FFh,0FFh,0FFh,0FFh, 00h
                DB  0FFh,0FFh,0FFh,0FFh, 00h,0FFh,0FFh,0FFh,0FFh, 00h
                DB  0FFh,0FFh,0FFh,0FFh, 00h,0FFh,0FFh,0FFh,0FFh, 00h
                DB  0FFh,0FFh,0FFh,0FFh, 00h, 3Fh, 33h, 33h, 33h,0F3h
                DB  0FFh, 33h, 33h, 33h,0FFh,0FFh, 3Fh, 33h,0F3h,0FFh
                DB  0FFh,0FFh, 33h,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh
                DB  0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh
                DB  0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh
                DB  0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh,0FFh
                DB  0FFh,0FFh,0FFh,0FFh,0FFh

COMMENT$
INIT_PORT       PROC        NEAR
                IN          AL,21h
                AND         AL,0FDh
                OUT         21h,AL
                MOV         DX,0293h
                MOV         AL,83h
                OUT         DX,AL
                CALL        INIT_MFG
                RET
INIT_PORT       ENDP

INIT_MFG        PROC        NEAR
                MOV         AL,0CFh
                MOV         DX,0290h
                OUT         DX,AL
                MOV         DX,0292h
                MOV         AL,10h
                CMP         Byte Ptr [PAL_TYPE],00
                JZ          PAL20PIN
                MOV         AL,00
PAL20PIN:
                OUT         DX,AL
                MOV         DX,0298h
                MOV         AL,00
                OUT         DX,AL
                MOV         Byte Ptr [EX00],00
                MOV         DX,0288h
                MOV         AL,0F7h
                OUT         DX,AL
                MOV         [EX01],AL
                MOV         DX,0288h
                MOV         BL,[MFG_TYPE]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     BH,00
        MOV     SI,0DBFh
        MOV     AL,CS:[BX+SI]
        MOV     AH,[EX01]
        AND     AH,0CFh
        OR      AL,AH
        OUT     DX,AL
        MOV     [EX01],AL
        RET
INIT_MFG        ENDP

CHECK_SUM   PROC    NEAR
        MOV     DX,0335h
        CALL    GOTO_XY
        MOV     SI,OFFSET SUM_ST
        CALL    WRITE_ST
        MOV     AH,31h
        CALL    SUMSUB1
        MOV     AX,BX
        CALL    CHAR_WRITE_INT
        RET
CHECK_SUM   ENDP

SUMSUB1     PROC    NEAR
        MOV     AH,31
        CALL    SUM_SUB2
        RET
SUMSUB1     ENDP

SUM_SUB2    PROC    NEAR
        MOV     CX,0800h
        CMP     Byte Ptr [PAL_TYPE],00
        JZ      EQUS
        MOV     CX,0C80h
EQUS:
        MOV     SI,OFFSET PAL_FUSE
        MOV     BX,0000
LOOP_SUM:
        MOV     AL,[SI]
        CMP     AL,AH
        JNZ     N_ZERO
        INC     BX
N_ZERO:
        INC     SI
        LOOP    LOOP_SUM
        RET
SUM_SUB2    ENDP

CHECK       PROC        NEAR
        MOV     AH,00
        CALL    SUM_SUB2
        MOV     AX,0800h
        CMP     Byte Ptr [PAL_TYPE],00
        JZ      JUMP_A
        MOV     AX,0C80h
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งไม่อนุญาตให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
JUMP_A:
        CMP         BX,AX
        JNZ         CLEAR_CARRY
        MOV         SI,OFFSET BUF_EMPTY
        CALL        WRITE_ST
        CALL        SOUND
        STC
        RET
CLEAR_CARRY:
        CLC
        RET
CHECK           ENDP

CLRSCR          PROC            NEAR
        PUSH        CX
        MOV         AX,0600h
        MOV         DX,184Fh
        MOV         BH,07
        INT         10h
        POP         DX
        CALL        GOTO_XY
        RET
CLRSCR          ENDP

GOTO_XY         PROC            NEAR
        MOV         AH,02
        MOV         BH,00
        INT         10h
        RET
GOTO_XY         ENDP

WRITE_ST        PROC            NEAR
        PUSH        DX
LOOP_ST:
        MOV         DL,CS:[SI]
        CMP         DL,00h
        JZ          NO_CHAR
        MOV         AH,02h
        INT         21h
        INC         SI
        JMP         LOOP_ST
NO_CHAR:
        POP         DX
        RET
WRITE_ST        ENDP

SOUND           PROC            NEAR
        PUSH        AX
        PUSH        CX
        IN          AL,61h                  ;'a'
        AND         AL,0FCh
        MOV         CX,0100h
FREQUENCY:
        XOR         AL,02
        OUT         61h,AL                  ;'a'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CALL    DELAY
        XOR     AL,02
        OUT     61h,AL                  ;'a'
        CALL    DELAY
        LOOP    FREQUENCY
        POP     CX
        POP     AX
        RET
SOUND   ENDP

DELAY   PROC    NEAR
        PUSH    CX
        MOV     CX,0099h
LOOP_DELAY:
        LOOP    LOOP_DELAY
        POP     CX
        RET
DELAY   ENDP

READ_KEY    PROC    NEAR
READ_BEGIN:
        IN      AL,21h                  ;'!'
        AND     AL,0FDh
        OUT     21h,AL                  ;'!'
        MOV     AH,01
        INT     16h
        JZ      NO_KEY
        MOV     AH,01
        INT     21h
        JMP     READ_BEGIN
NO_KEY:
        MOV     AH,01
        INT     21h
        AND     AL,5Fh                  ;'_'
        RET
READ_KEY    ENDP

WRITE_CHAR  PROC    NEAR
        PUSH    DX
        PUSH    BX
        MOV     DL,AL
        MOV     AH,02
        INT     21h
        POP     BX
        POP     DX
        RET
WRITE_CHAR  ENDP

GET_KEY     PROC    NEAR
        MOV     AH,01
        INT     21h
        RET
GET_KEY     ENDP

WR_INITIAL  PROC    NEAR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     DX,281h         ;MC 6821 #1
        MOV     AL,0FBh
        OUT     DX,AL
        MOV     DX,280h
        MOV     AL,0FFh
        OUT     DX,AL
        MOV     DX,281h
        OUT     DX,AL
        MOV     DX,283h
        MOV     AL,0FBh
        OUT     DX,AL
        MOV     DX,282h
        MOV     AL,0FFh
        OUT     DX,AL
        MOV     DX,283h
        OUT     DX,AL
        MOV     DX,285h         ;MC 6821 #2
        MOV     AL,0FBh
        OUT     DX,AL
        MOV     DX,284h
        MOV     AL,0FFh
        OUT     DX,AL
        MOV     DX,285h
        OUT     DX,AL
        MOV     DX,287h
        MOV     AL,0FBh
        OUT     DX,AL
        MOV     DX,286h
        MOV     AL,0FFh
        OUT     DX,AL
        MOV     DX,287h
        OUT     DX,AL
        MOV     DX,289h         ;MC 6821 #3
        MOV     AL,0FBh
        OUT     DX,AL
        MOV     DX,288h
        MOV     AL,0FFh
        OUT     DX,AL
        MOV     DX,289h
        OUT     DX,AL
        MOV     DX,28Bh
        MOV     AL,0FBh
        OUT     DX,AL
        MOV     DX,28Ah
        MOV     AL,0FFh
        OUT     DX,AL
        MOV     DX,28Bh
        OUT     DX,AL
        RET
WR_INITIAL      ENDP

SET_PZERO       PROC        NEAR
        MOV     DX,280h
        MOV     AL,00h
        OUT     DX,AL
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV         DX,282h
        MOV         AL,00h
        OUT         DX,AL
        MOV         DX,284h
        MOV         AL,00h
        OUT         DX,AL
        MOV         DX,286h
        MOV         AL,00h
        OUT         DX,AL
        MOV         DX,288h
        MOV         AL,00h
        OUT         DX,AL
        MOV         DX,28Ah
        MOV         AL,00h
        OUT         DX,AL
        RET
SET_PZERO       ENDP

SET_OD          PROC        NEAR
        MOV         DX,280h
        MOV         AL,0B0h
        OUT         DX,AL
        OUT         DX,AL
        CMP         Byte Ptr [PG_PRDLIN],00h
        JZ          SECOND_PG
        MOV         DX,28Bh
        MOV         AL,0FFh
        OUT         DX,AL
        OUT         DX,AL
        MOV         DX,282h
        MOV         AL,10h
        OUT         DX,AL
        OUT         DX,AL
        RET
SECOND_PG:
        MOV         DX,289h
        MOV         AL,0FFh
        OUT         DX,AL
        OUT         DX,AL
        MOV         DX,282h
        MOV         AL,20h
        OUT         DX,AL
        OUT         DX,AL
        RET
SET_OD          ENDP

SET_INP         PROC            NEAR
        MOV         AL,[CUR_INPLIN]
        MOV         CL,02h
        SHR         AL,CL
        MOV         CL,AL
        MOV         AL,01h
        SHL         AL,CL
        XOR         AL,0FFh
        MOV         CL,AL
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV         DX,288h
        MOV         AL,CL
        OUT         DX,AL
        OUT         DX,AL
        MOV         DX,28Ah
        MOV         AL,[CUR_INPLIN]
        AND         AL,01h
        JZ          INP_EVEN
        MOV         CL,0FFh
INP_EVEN:
        MOV         AL,CL
        OUT         DX,AL
        OUT         DX,AL
        RET
SET_INP         ENDP

SET_L_R         PROC        NEAR
        MOV         AL,[CUR_INPLIN]
        MOV         CL,00h
        AND         AL,02h
        JZ          OUT_PORT
        MOV         CL,08h
OUT_PORT:
        CMP         Byte Ptr [PG_PROLIN],01h
        JZ          FIRST_PG
        SHL         CL,01h
FIRST_PG:
        MOV         DX,28Ch
        MOV         AL,CL
        OUT         DX,AL
        MOV         DX,286h
        MOV         AL,CL
        OUT         DX,AL
        RET
SET_L_R         ENDP

SET_A0_A2       PROC            NEAR
        MOV         AL,[CUR_PROLIN]
        AND         AL,07h
        CMP         Byte Ptr [PG_PROLIN],00h
        JNZ         PG_ST
        MOV         AL,[CUR_PROLIN]
        AND         AL,0E0h
PG_ST:
        MOV         AH,AL
        MOV         DX,28Ch
        MOV         AL,AH
        OUT         DX,AL
        MOV         DX,286h
        MOV         AL,AH
        OUT         DX,AL
        RET
SET_A0_A2       ENDP

SET_00_03       PROC            NEAR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV             AL,[CUR_PROLIN]
        MOV             CL,03h
        SHR             AL,CL
        MOV             CL,AL
        MOV             AL,10h
        CMP             Byte Ptr [PG_PROLIN],00h
        JNZ             LO_PAGE
        MOV             AL,01h
LO_PAGE:
        SHL             AL,CL
        MOV             CL,AL
        MOV             DX,28Ch
        MOV             AL,CL
        OUT             DX,AL
        MOV             DX,286h
        MOV             AL,CL
        OUT             DX,AL
        RET
SET_00_03       ENDP

SET_10L8        PROC        NEAR
        MOV             DX,287h
        MOV             AL,0FFh
        CMP             Byte Ptr [PG_PROLIN],00h
        JNZ             LOW_PAGE
        MOV             AL,0F7h
LOW_PAGE:
        OUT             DX,AL
        RET
SET_10L8        ENDP

TRIG_MONO       PROC        NEAR
        MOV             DX,285h
        MOV             AL,0F7h
        OUT             DX,AL
        MOV             DX,285h
        MOV             AL,0FFh
        OUT             DX,AL
        RET
TRIG_MONO       ENDP

FUSE_POINT      PROC        NEAR
        PUSH            AX
        PUSH            BX
        MOV             CL,20h
        MOV             BX,0400h
        CMP             Byte Ptr [PAL_TYPE],00h
        JZ              PAL20P
        MOV             CL,28h
        MOV             BX,0640h
PAL20P:
        MOV             AL,[CUR_PROLIN]
        MUL             CL
        MOV             CL,[CUR_INPLIN]
        MOV             CH,00h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ADD         AX,CX
        MOV         SI,OFFSET PAL_FUSE
        ADD         SI,AX
        CMP         Byte Ptr [PG_PROLIN],00h
        JNZ         FIRST_PGPG
        ADD         SI,BX
FIRST_PGPG:
        POP         BX
        POP         AX
        RET
FUSE_POINT      ENDP

MAIN_WRITE      PROC        NEAR
        MOV         SI,OFFSET CK_READY
        CALL        WRITE_ST
        CALL        READ_KEY
        CMP         AL,59h                          ;'Y'
        JZ          Y_READY
        RET
Y_READY:
        MOV         Byte Ptr [JOB_STATUS],00h
        MOV         SI,OFFSET CK_WR_SEC
        CALL        WRITE_ST
        CALL        READ_KEY
        CMP         AL,59h                          ;'Y'
        JNZ         N_WR_SEC
        INC         Byte Ptr [JOB_STATUS]
N_WR_SEC:
        MOV         CX,1100h
        ADD         CH,04h
        CALL        CLRSCR
       ; CALL         MAIN_BLANK
        JMP         BLANK_OK
        MOV         SI,OFFSET WR_CON_MSG
        CALL        WRITE_ST
        CALL        READ_KEY
        CMP         AL,59h                          ;'Y'
        JNZ         WR_ERROR
BLANK_OK:
        CALL        WRITE_TO_PAL
      ;  CALL         MAIN_COMPARE
        JMP         CMP_SEC
WR_ERROR:
        MOV         SI,OFFSET WR_ERR_MSG
        CALL        WRITE_ST
        JMP         END_WRITE
        NOP
CMP_SEC:
        CMP         Byte Ptr [JOB_STATUS],00h
        JZ          END_WRITE
        CALL        MAIN_SECURITY
END_WRITE:
        MOV         SI,OFFSET WR_NEXT_MSG
        CALL        WRITE_ST
        CALL        READ_KEY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CMP         AL,59h                      ;'Y'
        JZ          N_WR_SEC
        RET
MAIN_WRITE      ENDP

WRITE_TO_PAL    PROC      NEAR
        CALL        WR_INITIAL
        MOV         Byte Ptr [CUR_INPLIN],00
        MOV         Byte Ptr [CUR_PROLIN],00
        MOV         Byte Ptr [PG_PROLIN],01h
BEGIN_WRITE:
        CALL        SET_PZERO
        MOV         BL,[CUR_INPLIN]
        CALL        GET_INPLIN
        JNB         NOT_BLOW
        MOV         BL,[CUR_PROLIN]
        CMP         Byte Ptr [PG_PROLIN],00h
        JNZ         FIRST_PGS
        ADD         BL,20h
        CMP         Byte Ptr [PAL_TYPE],00h
        JZ          FIRST_PGS
        ADD         BL,08h
FIRST_PGS:
        CALL        GET_PROLIN
        JNB         NOT_BLOW
        CALL        FUSE_POINT
        MOV         AL,[SI]
        CMP         AL,30h
        JZ          NOT_BLOW
        CALL        SET_OD
        CALL        SET_INP
        CALL        SET_L_R
        CALL        SET_A0_A2
        CALL        SET_O0_O3
        CALL        SET_10L8
        CALL        TRIG_MONO
NOT_BLOW:
        INC         Byte Ptr [CUR_PROLIN]
        MOV         AL,20h
        CMP         Byte Ptr [PAL_TYPE],00h
        JZ          PAL20_PIN
        MOV         AL,28h
PAL20_PIN:
        CMP         [CUR_PROLIN],AL
        JL          WR_NEXT_FUSE
        INC         Byte Ptr [CUR_INPLIN]
        MOV         AL,20h
        CMP         Byte Ptr [PAL_TYPE],00h
        JZ          PTYPE_20
        MOV         AL,28h
PTYPE_20:
        CMP         [CUR_INPLIN],AL
        JL          WR_NEXT_FUSE
        XOR         Byte Ptr [PG_PROLIN],01h
       ; CMP          Byte Ptr [CUR_INPLIN],00h
```

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ; JNZ          WR_NEXT_FUSE
         CMP          Byte Ptr [PG_PROLIN],01h
         JNZ          WR_NEXT_FUSE
         MOV          CX,1100h
         ADD          CH,04h
         CALL         CLRSCR
         MOV          SI,OFFSET COMPLETE_MSG
         CALL         WRITE_ST
         RET
WR_NEXT_FUSE:
         JMP          BEGIN_WRITE
WRITE_TO_PAL    ENDP

MAIN_SECURITY   PROC      NEAR
      ;  CALL          1134
      ;  CALL          1312
      ;  CALL          1242
        MOV           DX,0288h
        MOV           AL,[EX01]
        OR            AL,40h                ;'@'
        OUT           DX,AL
        MOV           [EX01],AL
      ;  CALL          12CC
        MOV           DX,0290h
        IN            AL,DX
        AND           AL,0F7h
        OUT           DX,AL
      ;  CALL          1D2B
        MOV           DX,0290h
        IN            AL,DX
        OR            AL,08h
        OUT           DX,AL
        MOV           DX,0292h
        IN            AL,DX
        AND           AL,0DFh
        OUT           DX,AL
        MOV           DX,0290h
        IN            AL,DX
        AND           AL,0F7h
        OUT           DX,AL
      ;  CALL          1D2B
      ;  CALL          1134
      ;  CALL          16CB
        RET
MAIN_SECURITY   ENDP

MAIN_BLANK      PROC       NEAR
          MOV       SI,OFFSET CKBLANK_MSG
          CALL      WRITE_ST
          MOV       Byte Ptr [EX07],05h
          MOV       Byte Ptr [EX08],00
      ;    CALL      14B5
          CMP       Byte Ptr [EX08],00
          JNZ       CKBLANK_ERR
          MOV       SI,OFFSET OK_MSG
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CALL    WRITE_ST
        CLC
        RET
CKBLANK_ERR:
        MOV     SI,OFFSET ERROR_MSG
        CALL    WRITE_ST
        CALL    SOUND
        STC
        RET
MAIN_BLANK      ENDP

MAIN_COMPARE    PROC    NEAR
        MOV     CX,1500h
        CALL    CLRSCR
        CALL    CHECK
        JNB     COMPARE_CK
        RET
COMPARE_CK:
        MOV     SI,OFFSET COMPARE_MSG
        CALL    WRITE_ST
        MOV     Byte Ptr [EX07],01
        MOV     Byte Ptr [EX08],00
    ;   CALL    14B5
        CMP     Byte Ptr [EX08],00
        JNZ     COMPARE_ERR
        MOV     SI,OFFSET OK_MSG
        CALL    WRITE_ST
        CLC
        RET
COMPARE_ERR:
        MOV     SI,OFFSET ERROR_MSG
        CALL    WRITE_ST
        CALL    SOUND
        STC
        RET
MAIN_COMPARE    ENDP

MAIN_OUTPUT     PROC    NEAR
        MOV     SI,OFFSET OUTPUT_MSG
        CALL    WRITE_ST
        CALL    FORMAT_FILE
        MOV     AL,01h
        CALL    OPEN_FILE
        JB      FILE_ERR
        MOV     AH,40h                  ;'@'
        MOV     CX,0CECh
        SUB     DI,CX
        MOV     CX,DI
        MOV     DX,OFFSET DAT_BUF
        INT     21h
        JNB     OUT_OK
        MOV     SI,OFFSET OUTERR_MSG
        CALL    WRITE_ST
        CALL    SOUND
        RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
OUT_OK:
        CALL    CLOSE_FILE
        MOV     SI,OFFSET OUTOK_MSG
        CALL    WRITE_ST
        RET
FILE_ERR:
        MOV     SI,OFFSET FILEERR_MSG
        CALL    WRITE_ST
        RET
MAIN_OUTPUT     ENDP

FORMAT_FILE     PROC        NEAR
        MOV     DI,OFFSET DAT_BUF
        MOV     AL,[PALNUM]
        MOV     AH,00h
        MOV     CL,0Fh
        MUL     CL
        MOV     BX,AX
        MOV     SI,OFFSET DISP_NUM
        ADD     SI,BX
NUM_LOOP:
        MOV     AL,CS:[SI]
        CMP     AL,20h                  ;' '
        JZ      MOVNUM_OK
        MOV     [DI],AL
        INC     DI
        INC     SI
        JMP     NUM_LOOP
MOVNUM_OK:
        MOV     AL,0Dh
        MOV     [DI],AL
        INC     DI
        MOV     AL,0Ah
        MOV     [DI],AL
        INC     DI
        MOV     SI,OFFSET FORMATFI_MSG
MSG_LOOP:
        MOV     AL,CS:[SI]
        CMP     AL,00h
        JZ      MOVMSG_OK
        MOV     [DI],AL
        INC     DI
        INC     SI
        JMP     MSG_LOOP
MOVMSG_OK:
        MOV     Byte Ptr [PAL_PG],00
        MOV     Byte Ptr [TATINP_LIN],20h       ;' '
        CMP     Byte Ptr [PAL_TYPE],00h
        JZ      DATA_BUF
        MOV     Byte Ptr [TATINP_LIN],28h
DATA_BUF:
        CALL    RD_BUFFER
        CALL    PAGEDOWN
        JB      DATA_BUF
        MOV     AL,1Ah
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MOV     [DI],AL
        INC     DI
        RET
FORMAT_FILE     ENDP

RD_BUFFER       PROC        NEAR
        PUSH    SI
        PUSH    BX
        PUSH    DX
        PUSH    CX
        MOV     BH,02h
        MOV     BL,00h
GET_DATA:
        CALL    DATA_BUFFER
        INC     BL
        CMP     BL,08h
        JL      GET_DATA
        MOV     BL,00h
        INC     Byte Ptr [PAL_PG]
        DEC     BH
        JNZ     GET_DATA
        DEC     Byte Ptr [PAL_PG]
        DEC     Byte Ptr [PAL_PG]
        POP     CX
        POP     DX
        POP     BX
        POP     SI
        RET
RD_BUFFER       ENDP

DATA_BUFFER     PROC        NEAR
        PUSH    BX
        MOV     AL,[PAL_PG]
        MOV     CL,08h
        MUL     CL
        ADD     AL,BL
        MOV     [CURPRO_LIN],AL
        MOV     CL,[TATINP_LIN]
        MUL     CL
        MOV     SI,OFFSET PAL_FUSE
        ADD     SI,AX
        PUSH    SI
        MOV     BL,[CURPRO_LIN]
        CALL    GET_PROLIN
        POP     SI
        JNB     ENDOF_DATA
        MOV     CX,0000h
BEGIN_LOOP:
        PUSH    CX
        PUSH    SI
        PUSH    BX
        MOV     BL,CL
        CALL    GET_INPLIN
        POP     BX
        POP     SI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        POP     CX
        JNB     ENDDAT_INP
        MOV     AL,[SI]
        MOV     [DI],AL
        INC     DI
ENDDAT_INP:
        INC     SI
        INC     CX
        CMP     CL,[TATINP_LIN]
        JL      BEGIN_LOOP
        MOV     AL,0Dh
        MOV     [DI],AL
        MOV     AL,0Ah
        INC     DI
        MOV     [DI],AL
        INC     DI
ENDOF_DATA:
        POP     BX
        RET
DATA_BUFFER     ENDP

MAIN_INPUT      PROC    NEAR
        MOV     SI,OFFSET INPUT_MSG
        CALL    WRITE_ST
        MOV     AL,00
        CALL    OPEN_FILE
        JB      RD_ERROR
        MOV     SI,DX
        MOV     AH,03h
        MOV     BH,00h
        INT     10h
        PUSH    DX
        MOV     DX,0240h
        CALL    GOTO_XY
ST_LOOP:
        MOV     AL,[SI]
        INC     SI
        CALL    WRITE_CHAR
        CMP     AL,00
        JNZ     ST_LOOP
        POP     DX
        CALL    GOTO_XY
        MOV     AH,3Fh                  ;'?'
        MOV     CX,0CECh
        MOV     DX,OFFSET DAT_BUF
        INT     21h
        JNB     RD_OK
        CALL    CLOSE_FILE
RD_ERROR:
        MOV     SI,OFFSET RDERR_MSG
        CALL    WRITE_ST
        CALL    SOUND
        RET
RD_OK:
        MOV     SI,OFFSET RDOK_MSG
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CALL        WRITE_ST
        CALL        CLOSE_FILE
        MOV         DI,OFFSET DAT_BUF
        MOV         CX,00FFh
AA:
        MOV         AL,[DI]
        INC         DI
        CMP         AL,4Ch                  ;'L'
        JZ          BB
        LOOP        AA
BB:
        MOV         AL,[DI]
        INC         DI
        CMP         AL,30h                  ;'0'
        JNZ         AA
        MOV         CX,0003h
CC:
        MOV         AL,[DI]
        INC         DI
        CMP         AL,30h                  ;'0'
        JNZ         END_INPUT
        LOOP        CC
DDDDA:
        MOV         AL,[DI]
        INC         DI
        CMP         AL,30h                  ;'0'
        JL          DDDDA
        DEC         DI
        MOV         Byte Ptr [PAL_PG],00
        MOV         Byte Ptr [TATINP_LIN],20h       ;' '
        CMP         Byte Ptr [PAL_TYPE],00
        JZ          DISKTO_BUF
        MOV         Byte Ptr [TATINP LIN],28h       ;'('
DISKTO_BUF:
        CALL        CKPG_FORLD
        CALL        PAGEDOWN
        JB          DISKTO_BUF
END_INPUT:
        RET
MAIN_INPUT      ENDP

CKPG_FORLD      PROC        NEAR
        PUSH        SI
        PUSH        BX
        PUSH        DX
        PUSH        CX
        MOV         BH,02h
        MOV         BL,00
LD_LOOP:
        CALL        LDTO_BUFFER
        INC         BL
        CMP         BL,08h
        JL          LD_LOOP
        MOV         BL,00h
        INC         Byte Ptr [PAL_PG]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            DEC         BH
            JNZ         LD_LOOP
            DEC         Byte Ptr [PAL_PG]
            DEC         Byte Ptr [PAL_PG]
            POP         CX
            POP         DX
            POP         BX
            POP         SI
            RET
CKPG_FORLD          ENDP

LDTO_BUFFER         PROC        NEAR
            PUSH        BX
            MOV         AL,[PAL_PG]
            MOV         CL,08h
            MUL         CL
            ADD         AL,BL
            MOV         [CURPRO_LIN],AL
            MOV         CL,[TATINP_LIN]
            MUL         CL
            MOV         SI,OFFSET PAL_FUSE
            ADD         SI,AX
            PUSH        SI
            MOV         BL,[CURPRO_LIN]
            CALL        GET_PROLIN
            POP         SI
            JNB         END_LOAD
            MOV         CX,0000h
INPLIN_LOOP:
            PUSH        CX
            PUSH        SI
            PUSH        BX
            MOV         BL,CL
            CALL        GET_INPLIN
            POP         BX
            POP         SI
            POP         CX
            JNB         NO_VALUE
            MOV         AL,[DI]
            MOV         [SI],AL
            INC         DI
NO_VALUE:
            INC         SI
            INC         CX
            CMP         CL,[TATINP_LIN]
            JL          INPLIN_LOOP
            INC         DI
            INC         DI
END_LOAD:
            POP         BX
            RET
LDTO_BUFFER         ENDP

OPEN_FILE           PROC        NEAR
            MOV         BP,AX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CALL    NAME_FILE
        MOV     AX,BP
        MOV     AH,3Dh                  ;'='
        INT     21h
        JNB     OPENFI_OK
        MOV     AX,BP
        CMP     AL,01h
        JNZ     OPEN_ERROR
        MOV     AH,3Ch                  ;'<'
        MOV     CX,0000h
        INT     21h
        JNB     OPENFI_OK
OPEN_ERROR:
        MOV     SI,OFFSET FIERR_MSG
        CALL    WRITE_ST
        STC
        RET
OPENFI_OK:
        MOV     BX,AX
        MOV     [EXOD],AX
        CLC
        RET
OPEN_FILE       ENDP

CLOSE_FILE      PROC    NEAR
        PUSH    AX
        MOV     AH,3Eh                  ;'>'
        MOV     BX,[EXOD]
        INT     21h
        POP     AX
        RET
CLOSE_FILE      ENDP

NAME_FILE       PROC    NEAR
        MOV     SI,OFFSET GETNAME_MSG
        CALL    WRITE_ST
        MOV     AL,4Fh                  ;'O'
        MOV     [EX1A],AL
        MOV     DX,OFFSET EX1A
        MOV     AH,0Ah
        INT     21h
        MOV     SI,OFFSET SEND_CRLF
        CALL    WRITE_ST
        MOV     BL,[EX1B]
        MOV     BH,00h
        MOV     Byte Ptr [BX+EX1C],00
        MOV     DX,OFFSET EX1C
        RET
NAME_FILE       ENDP


MAIN_EDIT   PROC    NEAR
BEGIN_EDIT:
        MOV     Byte Ptr [PAL_PG],00
        MOV     Byte Ptr [TATINP_LIN],20h       ;' '
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CMP     Byte Ptr [PAL_TYPE],00
        JZ      PAL20S
        MOV     Byte Ptr [TATINP_LIN],28h          ;'('
PAL20S:
        MOV     CX,0500h
        CALL    CLRSCR
        CALL    TITLE_EDIT
CLR_EDIT:
        MOV     CX,0703h
        MOV     CL,00
        MOV     DX,1733h
        MOV     AX,0600h
        MOV     BH,07h
        INT     10h
        CALL    DISP_FUSE_MAP
        CALL    MOVE_HOME
RD_EDIT_KEY:
        MOV     AH,08h
        INT     21h
BLOWN_KEY:
        CMP     AL,2Dh                             ;'-'
        JNZ     UNBLOWN1
        CALL    BLOWN
        JMP     RD_EDIT_KEY
UNBLOWN1:
        CMP     AL,58h                             ;'X'
        JNZ     UNBLOWN2
        CALL    UNBLOWN
        JMP     RD_EDIT_KEY
UNBLOWN2:
        CMP     AL,78h                             ;'x'
        JNZ     ESC_RET
        CALL    UNBLOWN
        JMP     RD_EDIT_KEY
ESC_RET:
        CMP     AL,1Bh
        JNZ     BLANK_EDIT
        MOV     DX,0300h
        CALL    GOTO_XY
        MOV     CX,0020h
WR_LOOP:
        MOV     AL,20h                             ;' '
        CALL    WRITE_CHAR
        LOOP    WR_LOOP
        RET
BLANK_EDIT:
        CMP     AL,42h                             ;'B'
        JZ      BLANK_CK
        CMP     AL,62h                             ;'b'
        JNZ     NO_B
BLANK_CK:
        CALL    SET_ZERO
        CALL    CHECK_SUM
        CALL    WRITE_BLANK
        JMP     BEGIN_EDIT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
NO_B:
        CMP     AL,00
        JNZ     RD_EDIT_KEY
        MOV     AH,08h
        INT     21h
        CMP     AL,47h                  ;'G'
        JNZ     UP_KEY
        CALL    MOVE_HOME
        JMP     RD_EDIT_KEY
UP_KEY:
        CMP     AL,48h
        JNZ     PGUP_KEY
        CALL    MOVE_UP
        JMP     RD_EDIT_KEY
PGUP_KEY:
        CMP     AL,49h                  ;'I'
        JNZ     LT_KEY
        CALL    PAGEUP
        JNB     NEW_READ
        JMP     CLR_EDIT
LT_KEY:
        CMP     AL,4Bh                  ;'K'
        JNZ     RT_KEY
        CALL    MOVE_LT
        JMP     RD_EDIT_KEY
RT_KEY:
        CMP     AL,4Dh                  ;'M'
        JNZ     END_KEY
        CALL    MOVE_RT
        JMP     RD_EDIT_KEY
END_KEY:
        CMP     AL,4Fh                  ;'O'
        JNZ     DN_KEY
        CALL    MOVE_END
        JMP     RD_EDIT_KEY
DN_KEY:
        CMP     AL,50h                  ;'P'
        JNZ     PGDN_KEY
        CALL    MOVE_DN
        JMP     RD_EDIT_KEY
PGDN_KEY:
        CMP     AL,51h                  ;'Q'
        JNZ     NEW_READ
        CALL    PAGEDOWN
        JNB     NEW_READ
        JMP     CLR_EDIT
NEW_READ:
        JMP     RD_EDIT_KEY
MAIN_EDIT       ENDP

SET_ZERO        PROC        NEAR
        MOV     SI,OFFSET PAL_FUSE
        MOV     CX,0C80h
        MOV     AL,30h
LOOP_SET:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MOVE_RT         PROC          NEAR
          MOV         AL,[EX09]
          ADD         AL,01h
          CMP         AL,[TATINP_LIN]
          JL          N_LAST_RT
          RET
N_LAST_RT:
          INC         DL
          INC         Byte Ptr [EX09]
          DEC         CH
          JNZ         GROUP_RT
          MOV         CH,04h
          INC         DL
GROUP_RT:
          INC         SI
          CALL        GOTO_XY
          RET
MOVE_RT         ENDP

MOVE_LT         PROC          NEAR
          CMP         Byte Ptr [EX09],00
          JNZ         N_LAST_LEFT
          RET
N_LAST_LEFT:
          DEC         DL
          DEC         Byte Ptr [EX09]
          INC         CH
          CMP         CH,05h
          JL          GROUP_COL
          MOV         CH,01h
          MOV         AL,08h
          CALL        WRITE_CHAR
          DEC         DL
GROUP_COL:
          DEC         SI
          MOV         AL,08h
          CALL        WRITE_CHAR
          RET
MOVE_LT         ENDP

MOVE_UP         PROC          NEAR
          CMP         Word Ptr [EX0A],+00
          JNZ         N_LAST_TOP
          RET
N_LAST_TOP:
          DEC         Word Ptr [EX0A]
          DEC         DH
          CMP         Word Ptr [EX0A],+07
          JNZ         N_CENTER
          DEC         DH
N_CENTER:
          CALL        GOTO_XY
          MOV         AL,[TATINP_LIN]
          MOV         AH,00h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          SUB       SI,AX
          RET
MOVE_UP   ENDP

MOVE_DN   PROC      NEAR
          CMP       Word Ptr [EX0A],+0Fh
          JL        N_LAST_BOTTOM
          RET
N_LAST_BOTTOM:
          INC       Word Ptr [EX0A]
          INC       DH
          CMP       Word Ptr [EX0A],+08
          JNZ       CENTER
          INC       DH
CENTER:
          CALL      GOTO_XY
          MOV       AL,[TATINP_LIN]
          MOV       AH,00h
          ADD       SI,AX
          RET
MOVE_DN   ENDP

PAGEDOWN  PROC      NEAR
          MOV       AL,06h
          CMP       Byte Ptr [PAL_TYPE],00
          JZ        PAL20
          MOV       AL,08h
PAL20:
          CMP       [PAL_PG],AL
          JGE       LAST_PG
          ADD       Byte Ptr [PAL_PG],02
          STC
          RET
LAST_PG:
          CLC
          RET
PAGEDOWN  ENDP

PAGEUP    PROC      NEAR
          CMP       Byte Ptr [PAL_PG],00
          JZ        TOP_PG
          SUB       Byte Ptr [PAL_PG],02
          STC
          RET
TOP_PG:
          CLC
          RET
PAGEUP    ENDP

MOVE_END  PROC      NEAR
BEGIN_PROC:
          CMP       Word Ptr [EX0A],+0Fh
          JL        NOT_END
          RET
NOT_END:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        PUSH    BX
        MOV     SI,OFFSET SEND_CRLF
        CALL    WRITE_ST
        MOV     AL,[PAL_PG]
        MOV     CL,08h
        MUL     CL
        ADD     AL,BL
        MOV     [CURPRO_LIN],AL
        MOV     CL,[TATINP_LIN]
        MUL     CL
        MOV     SI,OFFSET PAL_FUSE
        ADD     SI,AX
        PUSH    BX
        MOV     BL,[CURPRO_LIN]
        MOV     BH,00h
        MOV     AL,CS:[BX+OFFSET PRO_LINE]
        MOV     AH,AL
        AND     AL,0F0h
        MOV     CL,04h
        SHR     AL,CL
        OR      AL,30h                  ;'0'
        CALL    WRITE_CHAR
        MOV     AL,CS:[BX+OFFSET PRO_LINE]
        AND     AL,0Fh
        OR      AL,30h                  ;'0'
        CALL    WRITE_CHAR
        PUSH    SI
        MOV     BL,[CURPRO_LIN]
        CALL    GET_PROLIN
        POP     SI
        POP     BX
        JNB     END_PROC
        MOV     AL,20h                  ;' '
        CALL    WRITE_CHAR
        MOV     DL,04h
        MOV     CX,0000h
LOOP_INP:
        PUSH    CX
        PUSH    SI
        PUSH    BX
        MOV     BL,CL
        CALL    GET_INPLIN
        POP     BX
        POP     SI
        POP     CX
        JB      N_WR_BLANK
        MOV     AL,20h                  ;' '
        CALL    WRITE_CHAR
        JMP     CUR_INP
        NOP
N_WR_BLANK:
        MOV     AL,[SI]
        CMP     AL,00
        JZ      CUR_INP
        CMP     AL,30h                  ;'0'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            JNZ         STAT_BLOWN
            MOV         AL,78h                  ;'x'
            JMP         WR_STAT_FUSE
STAT_BLOWN:
            MOV         AL,2Dh
WR_STAT_FUSE:
            CALL        WRITE_CHAR
CUR_INP:
            INC         SI
            DEC         DL
            JNZ         CK_INP
            MOV         AL,20h                  ;' '
            CALL        WRITE_CHAR
            MOV         DL,04h
CK_INP:
            INC         CX
            CMP         CL,[TATINP_LIN]
            JL          LOOP_INP
END_PROC:
            POP         BX
            RET
            PUSH        CX
            MOV         CX,0032h
WR_LOOPS:
            MOV         AL,20h                  ;' '
            CALL        WRITE_CHAR
            LOOP        WR_LOOPS
            POP         CX
            JMP         END_PROC
DISP_FUSE_LINE   ENDP

CHAR_WRITE_INT   PROC    NEAR
            MOV         BL,AH
            MOV         BH,AL
            CALL        WRITE_INT
            MOV         BL,BH
            CALL        WRITE_INT
            RET
CHAR_WRITE_INT   ENDP

WRITE_INT      PROC      NEAR
            MOV         AL,BL
            MOV         CL,04h
            SHR         AL,CL
            CALL        WRITE_HEX
            MOV         AL,BL
            AND         AL,0Fh
            CALL        WRITE_HEX
            RET
WRITE_INT      ENDP

WRITE_HEX      PROC      NEAR
            ADD         AL,30h                  ;'0'
            CMP         AL,3Ah                  ;':'
            JL          LE_NINE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ADD     AL,07h
LE_NINE:
        MOV     AH,0Eh
        INT     10h
        RET
WRITE_HEX   ENDP

GET_PROLIN  PROC    NEAR
        MOV     AL,[PALNUM]
        MOV     CL,0Ah
        XOR     AH,AH
        MUL     CL
        MOV     SI,OFFSET PROD_PATTERN
        ADD     SI,AX
FORM_SUB:
        MOV     CL,08h
        MOV     AL,BL
        MOV     AH,00
        DIV     CL
        MOV     CL,AH
        MOV     AH,00
        ADD     SI,AX
        MOV     AL,CS:[SI]
        INC     CL
        SHR     AL,CL
        RET
GET_PROLIN  ENDP

GET_INPLIN  PROC    NEAR
        MOV     AL,[PALNUM]
        MOV     CL,05h
        MOV     AH,00h
        MUL     CL
        MOV     SI,OFFSET INP_PATTERN
        ADD     SI,AX
        JMP     FORM_SUB
GET_INPLIN  ENDP

TITLE_EDIT  PROC    NEAR
        MOV     DX,0300h
        CALL    GOTO_XY
        MOV     SI,OFFSET DISP_TITLE_E
        CALL    WRITE_ST
        MOV     SI,OFFSET DISP_MENU_E
        MOV     DX,0839h
LOOP_LF:
        CALL    GOTO_XY
LOOP_CH:
        MOV     AL,CS:[SI]
        INC     SI
        CMP     AL,00h
        JZ      END_TITLE_EDIT
        CMP     AL,01h
        JZ      LINEFEED
        CALL    WRITE_CHAR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JMP         LOOP_CH
LINEFEED:
        INC         DH
        JMP         LOOP_LF
END_TITLE_EDIT:
        RET
TITLE_EDIT      ENDP

SET_SP          PROC        NEAR
        PUSH        AX
        PUSH        BX
        PUSH        CX
        PUSH        DX
        MOV         CX,0004h
        MOV         AX,SP
        PUSH        SP
        POP         BX
        CMP         AX,BX
        JNZ         NOP_LOOP
        MOV         CX,0014h
NOP_LOOP:
        NOP
        NOP
        LOOP        NOP_LOOP
        POP         DX
        POP         CX
        POP         BX
        POP         AX
        RET
SET_SP          ENDP

WRITE_BLANK     PROC    NEAR
        MOV         AH,03h
        MOV         BH,00h
        INT         10h
        PUSH        DX
        MOV         DX,0240h
        CALL        GOTO_XY
        MOV         CX,000Fh
BLANK_ASCII:
        MOV         AL,20h
        CALL        WRITE_CHAR
        LOOP        BLANK_ASCII
        POP         DX
        CALL        GOTO_XY
        RET
WRITE_BLANK     ENDP


CODE            ENDS

   END          MAIN_PRO
```

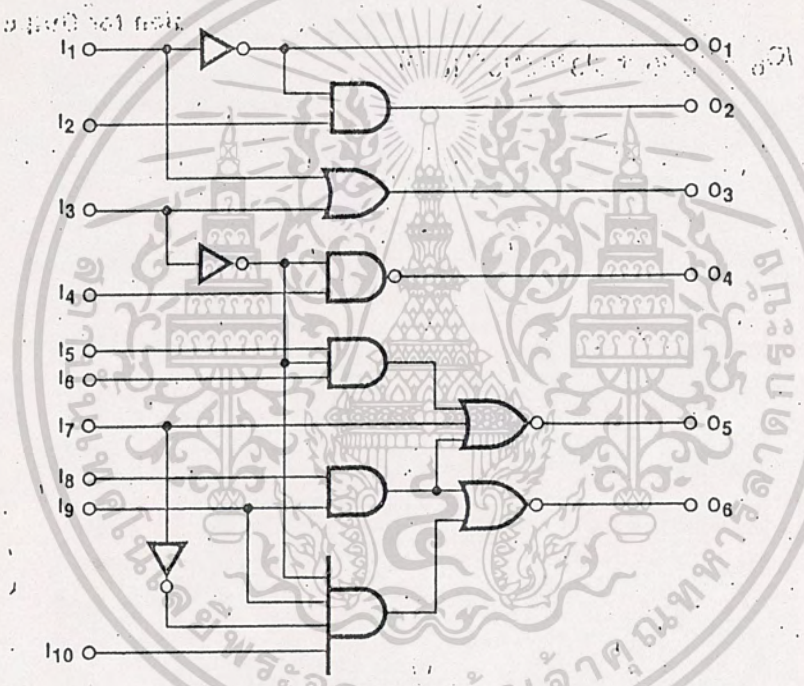
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

เมื่อต้องการนำ PAL ไปใช้แทนวงจรลอจิก ซึ่งต้องใช้เกตหลายชนิดหลายตัว จะต้องรู้ว่าวงจรและคุณสมบัติที่ต้องการเป็นอย่างไร แล้วจึงนำมาเลือกชนิดของ PAL พร้อมกับออกแบบและคำนวณแล้วจึงโปรแกรม

จะยกตัวอย่างวงจรลอจิกในรูปที่ 24 ซึ่งถ้าใช้ไอซีเกตธรรมดามาต่อกันต้องใช้จำนวนไม่น้อยกว่า 5 ตัว แต่เราจะออกแบบและใช้ PAL ขนาด 20 ขา มาทำงานแทนเพียงตัวเดียว



รูปที่ 24 Design Example , Logic Diagram

เริ่มต้นต้องเลือเบอร์หรือชนิดของ PAL กันก่อน พิจารณาดูที่เอาต์พุตของรูปที่ 24 จะเห็นว่าเอาต์พุต (Q_1-Q_6)มากกว่าครึ่งถูกอินเวอร์หรือแอคทีฟที่ลอจิก "0" แสดงว่าถ้าใช้ชนิด L จะสะดวกกว่า จากนั้นมาดูที่จำนวนอินพุตเอาต์พุต จะเห็นวฝ่าต้องการ 10 อินพุต กับ 6 เอาต์พุต ดังนั้นจึงน่าจะใช้ PAL เบอร์ 10L8 ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แต่เมื่อพลิกกลับมาดูโครงสร้างของ PAL 10L8 จะเห็นว่าเอาต์พุตซึ่งเป็น NOR ทั้ง 8 ตัวไม่มีตัวใดเป็นแบบ 3 อินพุตเลย (แต่ Q_5 ต้องการ) จึงเปลี่ยนไปใช้เบอร์อื่น

เมื่อดูที่เบอร์ 12L6 ซึ่งมี NOR อยู่ 2 ตัว ซึ่งมี 4 อินพุต เนื่องจากเอาต์พุต Q_5 ในรูปที่ 24 เป็น NOR เพียง 3 อินพุต PAL เบอร์ 12L6 จึงนำมาใช้ได้

หลังจากที่เลือกเบอร์ได้ตามต้องการแล้ว ต่อมาจึงคำนวณความสัมพันธ์ของอินพุต-เอาต์พุต ด้วยทฤษฎีของดีมอร์แกน ตามนี้

$Q_1 = I_1$ หรือ $Q_1 = I_1$

$Q_2 = I_1 . I_2$ หรือ $Q_2 = I_1 + I_2$

$Q_3 = I_1 + I_3$ หรือ $Q_3 = I_1 . I_3$

$Q_4 = I_3 . I_4$ หรือ $Q_4 = I_3 . I_4$

$Q_5 = I_3 . I_5 . I_6 + I_7 + I_8 . I_9$ หรือ

$Q_5 = I_3 . I_5 . I_6 + I_7 + I_8 . I_9$

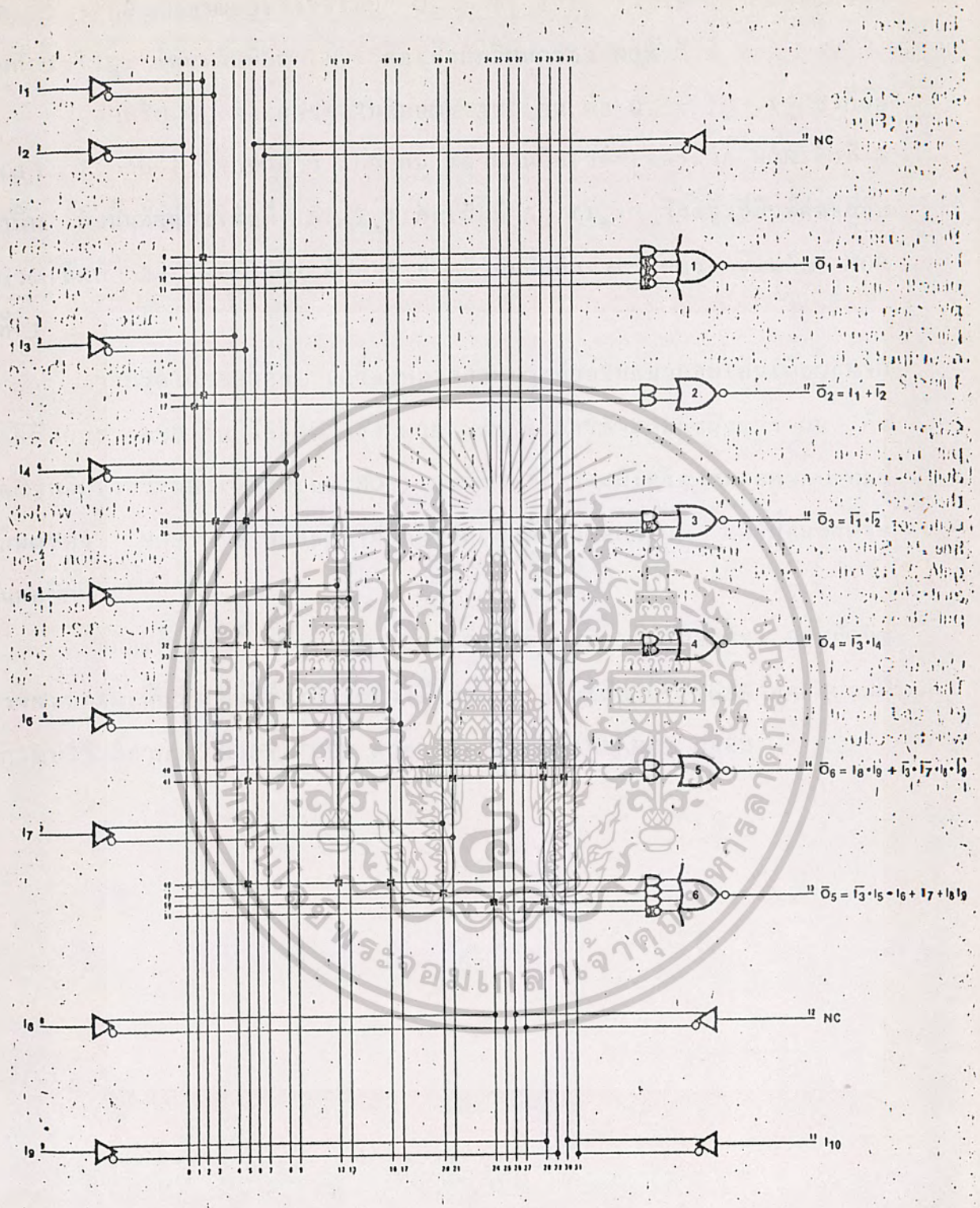
$Q_6 = I_8 . I_9 + I_3 . I_7 . I_9 . I_{10}$ หรือ

$Q_6 = I_8 . I_9 + I_3 . I_7 . I_9 . I_{10}$

เมื่อนำเอาที่คำนวณได้มากำหนดการโปรแกรม โครงสร้างภายในของตัว PAL จะเปลี่ยนเป็นดังรูปที่ 25 จะเห็นว่าเครื่องหมายกากบาทคือ ฟิวส์ที่ยังคงเหลืออยู่และเป็นจุดต่อระหว่างอินพุตกับ NOR

จากความต้องการให้ $Q_1 = I_1$ หรือ $Q_1 = I_1$ ในรูปที่ 25 ใช้ NOR ตัวที่ 1 ซึ่งมี 4 อินพุต ดังนั้นอีก 3 อินพุตของ NOR ที่ไม่ได้ใช้จึงต้องให้เป็นลอจิก "0" ด้วยวิธีนี้ฟิวส์ที่สายสัญญาณเส้นที่ 9, 10, 11 ตามแนวนอน จะยังต้องคงอยู่ โดยเขียนแทนด้วยเครื่องหมายกากบาทที่หน้าอินพุตของ NOR ตัวที่ 1 ก็พอ เพื่อลดจำนวนเครื่องหมายกากบาท แต่สัญญาณเส้นที่ 8 แนวนอน จะต่อเข้ากับสัญญาณเส้นที่ 2 แนวตั้ง (I_1) จึงมีฟิวส์ที่ตำแหน่งนี้เพียงตัวเดียว นอกนั้นถูกระเบิดหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์

อุปกรณ์ PLD ประเภท PAL ที่กล่าวถึงนี้ อาจจะเป็นของใหม่สำหรับบ้านเราถึงแม้ว่าจะได้มีผลิตออกมาจำหน่ายเป็นเวลานานเกือบสิบปีแล้ว แต่ด้วยเทคโนโลยีที่ไม่เคยหยุดอยู่กับที่ จึงมีการคิดค้นพัฒนา PLD ใหม่ๆ ออกมาอยู่เรื่อยๆ

เทคโนโลยีในการผลิตที่ได้ทำการพัฒนามาอย่างต่อเนื่อง ทำให้ราคาของ PLD มีราคาถูกลงในขณะที่ความเร็วในการทำงานจะสูงขึ้น, กินกำลังน้อยลง และใช้เทคโนโลยีของการผลิตหน่วยความจำเข้ามาเปลี่ยนแปลงโครงสร้างภายในดังที่มีการเรียกชื่อตามแต่ละชนิดดังนี้

- TTL-PLD โครงสร้างภายใน และสัญญาณลอจิกเป็น TTL แต่มีให้เลือกความเร็วถึง 4 ระดับ คือ ความเร็วมาตรฐาน, A, B และ D และยังมีรุ่นให้เลือกขนาดกินกำลัง 1/2 เท่า และ 1/4 เท่าอีกด้วย สำหรับรุ่นที่มีความเร็วจะมีชื่อเรียกทางการค้าว่า FPAL, FPLA, FPLS และ FPLD เป็นต้น

- COMS-PLD โครงสร้างภายใน และสัญญาณลอจิกเป็น CMOS จึงกินกำลังงานต่ำมากๆ และยังมีรุ่นที่เมื่ออยู่ในโหมดรอการทำงานจะไม่กินกระแสไฟเลย (zero-standby-current CMOS)

- ECL-PLD โครงสร้างภายในเป็น ECL ใช้สำหรับงานความถี่สูงมากๆ

ทั้ง 3 ชนิดที่กล่าวมานี้ยังคงใช้การระเบิดฟิวส์ ซึ่งเมื่อโปรแกรมไปแล้วจะไม่สามารถเปลี่ยนแปลงแก้ไขได้อีก คือ โปรแกรมได้เพียงครั้งเดียวแล้วจะอยู่เช่นนั้นตลอดไป แต่ PLD รุ่นใหม่ๆ จะใช้เทคโนโลยีของการผลิตหน่วยความจำเข้ามาช่วย ทำให้สามารถเปลี่ยนแปลงแก้ไขโปรแรมได้ หรือเรียกกันง่ายๆ ว่า PLD ที่ลบได้เช่นเดียวกับ EPROM, EEPROM หรือ RAM โดยโครงสร้างภายในจะไม่ใช้ฟิวส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการออกแบบสร้าง PAL PROGRAMMABLE CARD นี้ได้ประสบปัญหาหลายประการ ซึ่งผู้จัดทำได้แก้ไขจนสามารถทำงานได้ตามจุดมุ่งหมาย จึงขอเสนอปัญหาที่เกิดขึ้นเป็นข้อๆ ดังนี้คือ

1. ในส่วนของ CLOCK ซึ่ง MC6821 ต้องการอย่างต่ำ 1MHz ซึ่งตอนแรกออกแบบจาก OSC 2 MHz แล้วใช้ LS74 หาร 2 เหลือ 1MHz ปรากฏว่า DATA ที่ OUT ออกทาง OUTPUT PORT จะได้ไม่แน่นอน ซึ่งเกิดขึ้นจาก CLOCK ที่ได้ไม่ตรงตาม TIMING DIAGRAM ของ MC6821 ซึ่งผู้จัดทำได้แก้ไขโดยการนำ CLOCK 14.318 MHz จาก SLOT ของ IBM หาร 10 ซึ่งผลที่ได้ จะได้ DATA ตามต้องการ

2. ในส่วน DECODE OUTPUT PORT ควรทำการ FULL DECODEและควรทำการเช็ค TECHNICAL REFERENCE ของ IBM PC ดูว่า PORT ที่จะใช้ว่างหรือไม่

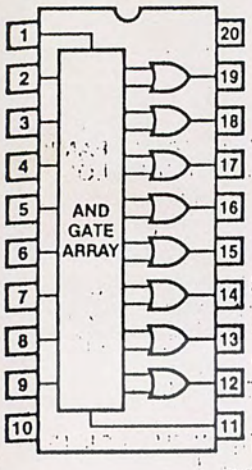
3. ในส่วนแผ่นวงจรพิมพ์ ถ้าเป็นไปได้ควรใช้ PLATED THROUGH HOLE เพราะ แผ่นวงจรพิมพ์สองหน้ามีปัญหาเรื่องจุดต่ออย่างมาก

4. เนื่องจาก TEXTOOL 24 PIN ไม่มีจำหน่ายจึงทำให้ไม่สามารถใช้กับ PAL 24 PIN ได้แต่ผู้จัดทำได้ออกแบบส่วนนี้ไว้แล้ว

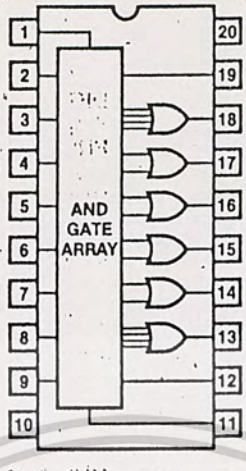
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PAL Logic Symbols — Series 20

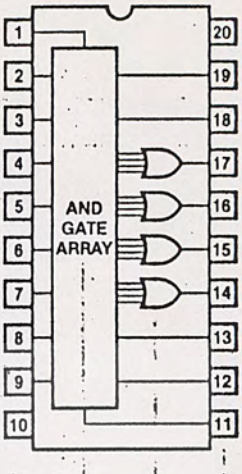
DMPAL10H8



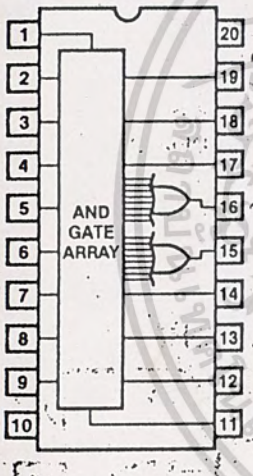
DMPAL12H6



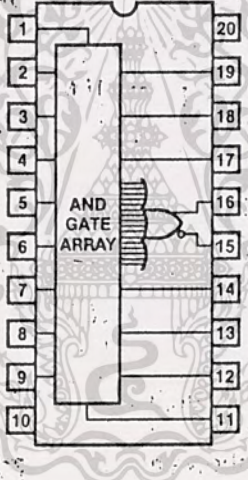
DMPAL14H4



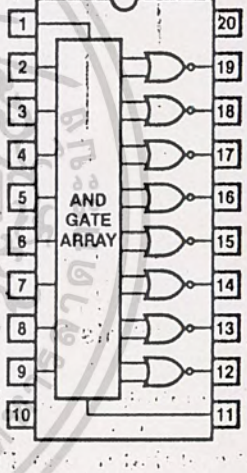
DMPAL16H2



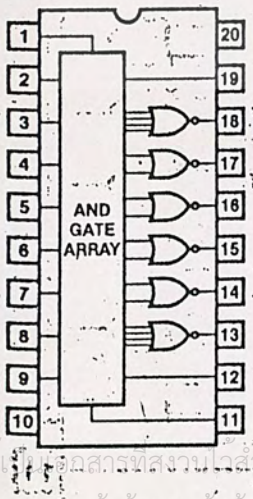
DMPAL16C1



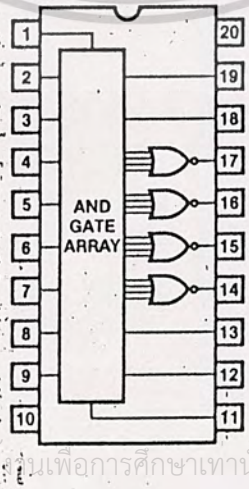
DMPAL10L8



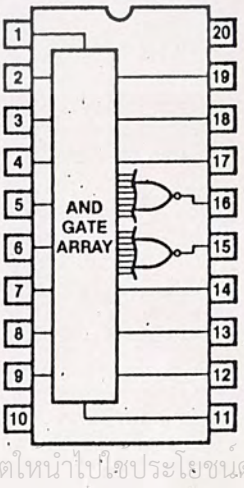
DMPAL12L6



DMPAL14L4



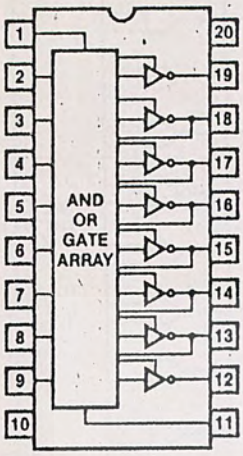
DMPAL16L2



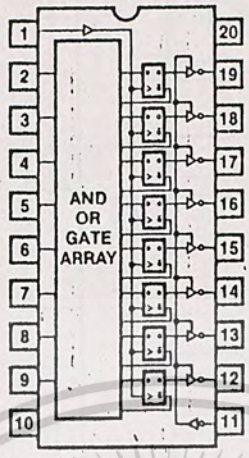
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PAL Logic Symbols - Series 20 (Contd.)

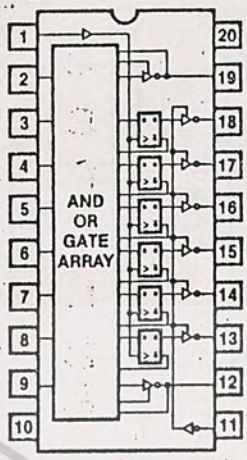
DMPAL16L8



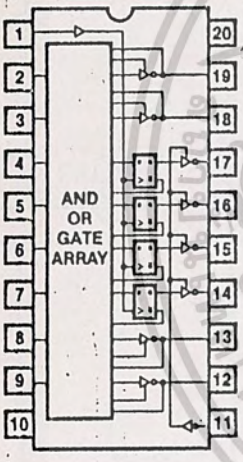
DMPAL16R8



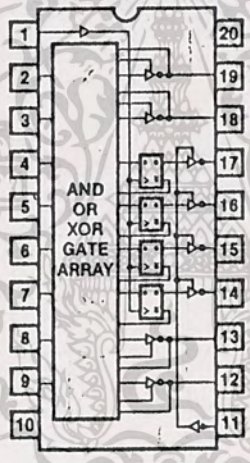
DMPAL16R6



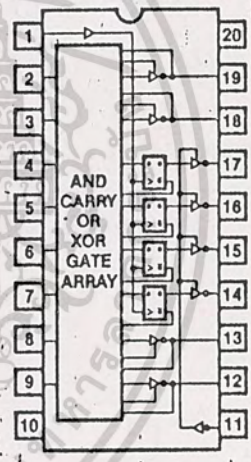
DMPAL16R4



DMPAL16X4

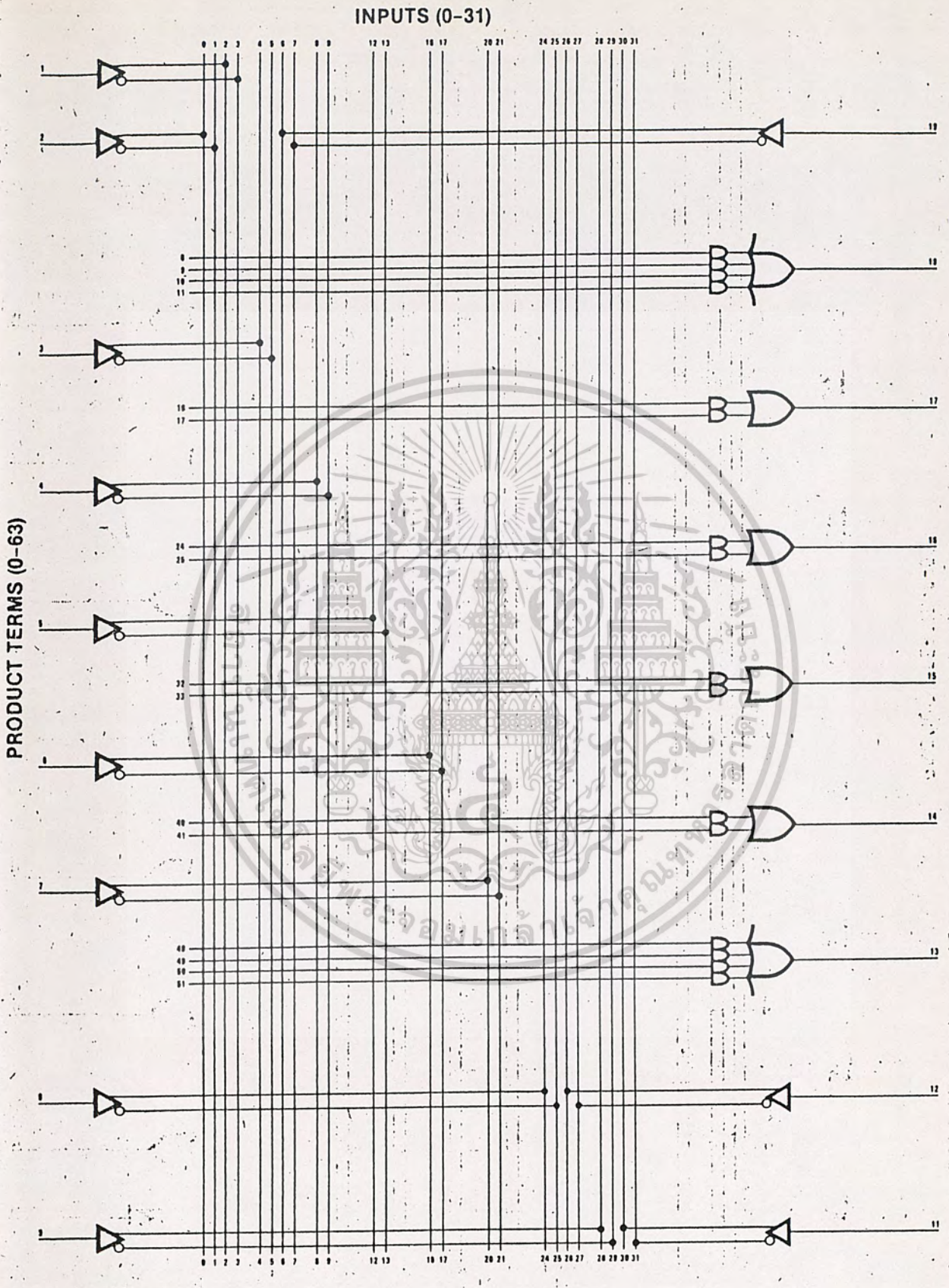


DMPAL16A4



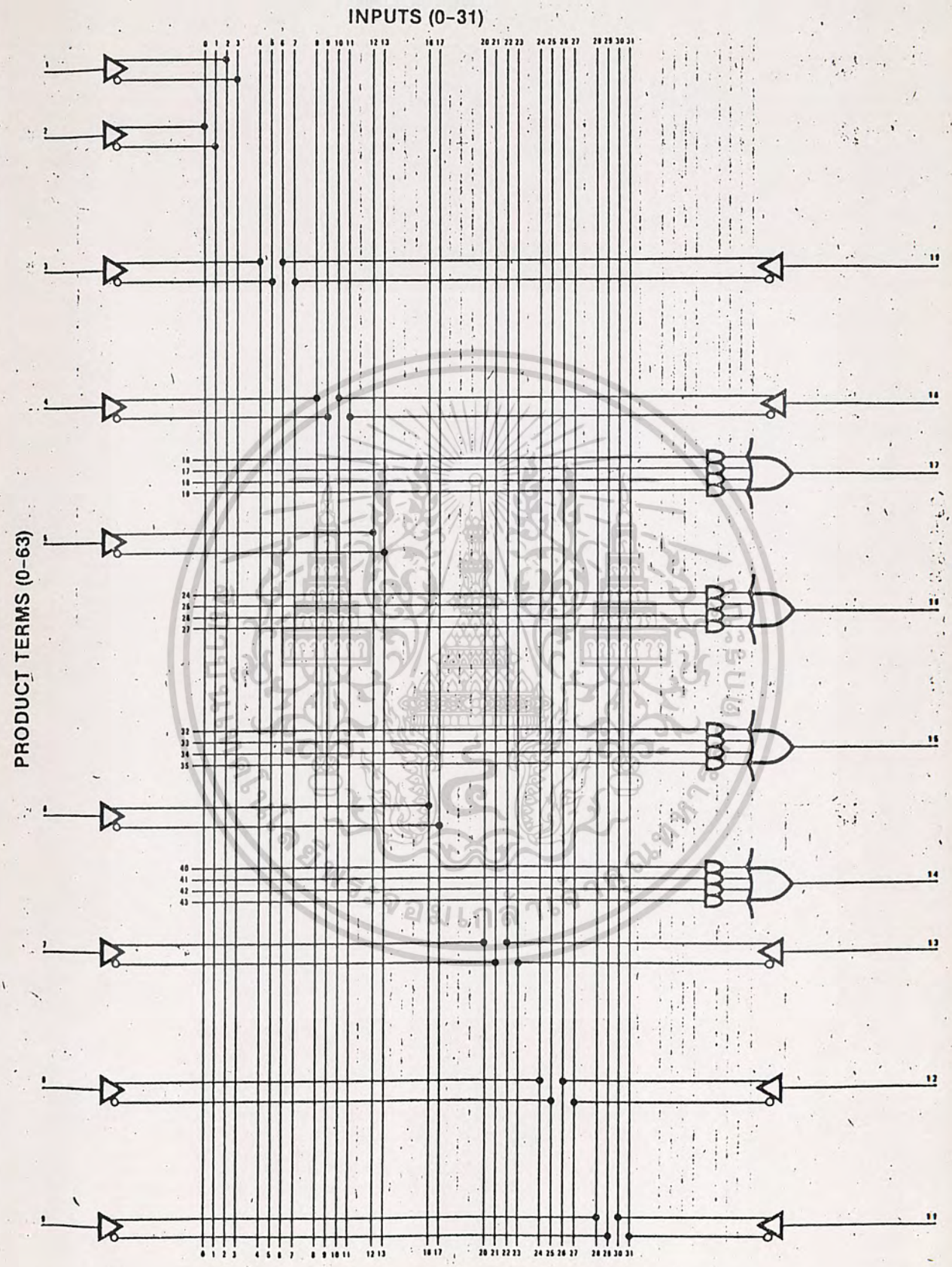
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Logic Diagram PAL12H6

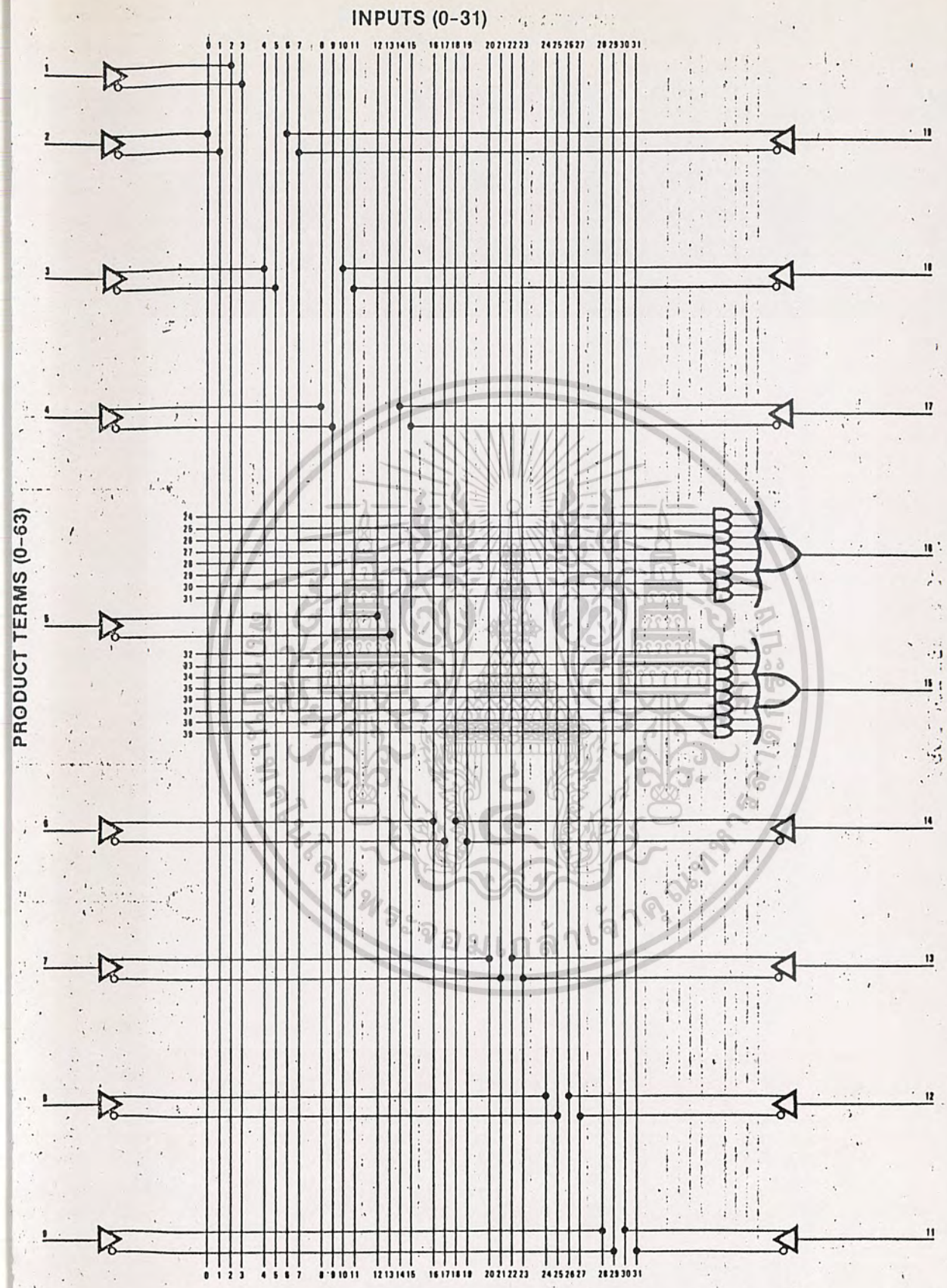


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Logic Diagram PAL14H4

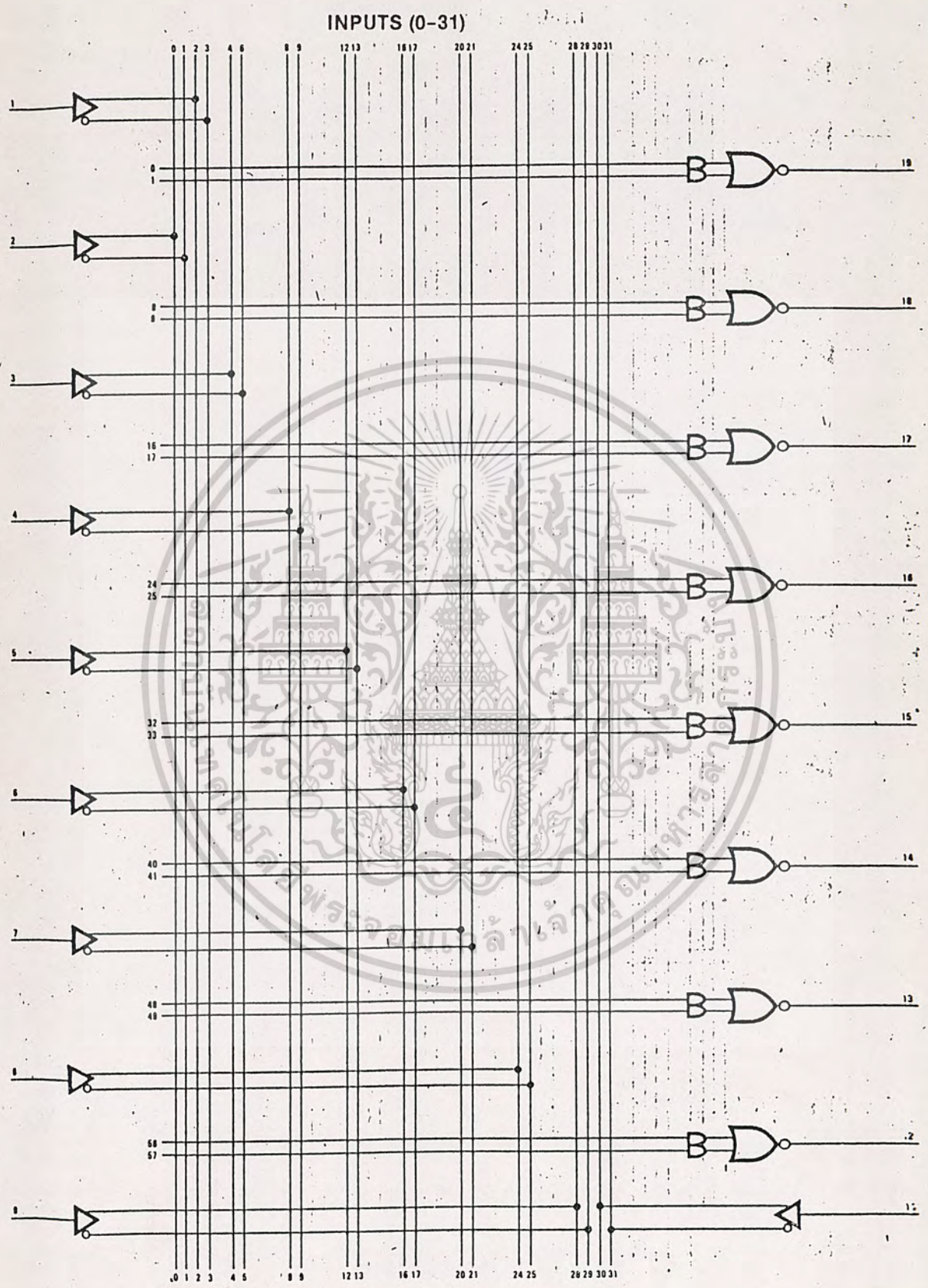


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

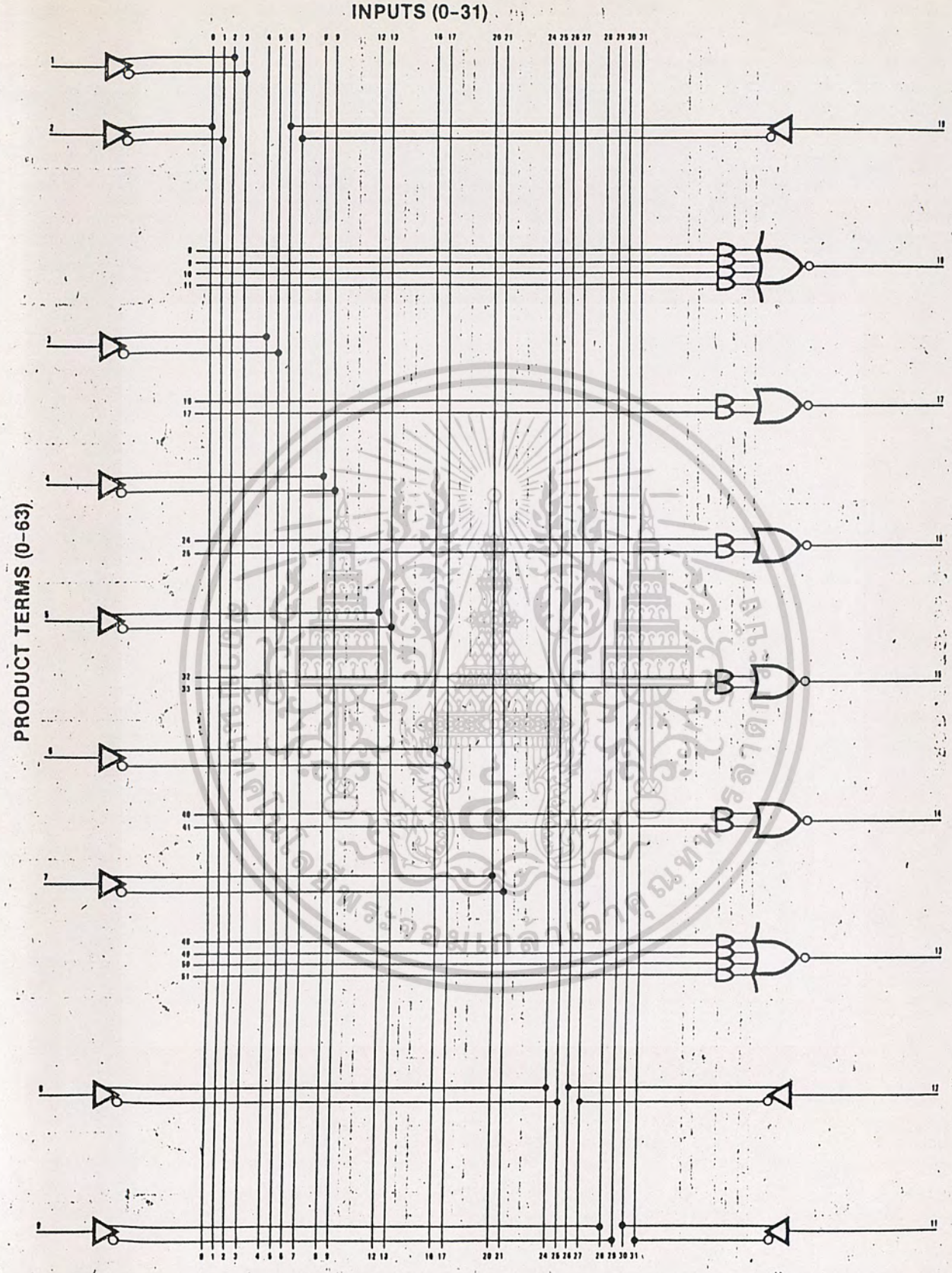


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Logic Diagram PAL10L8

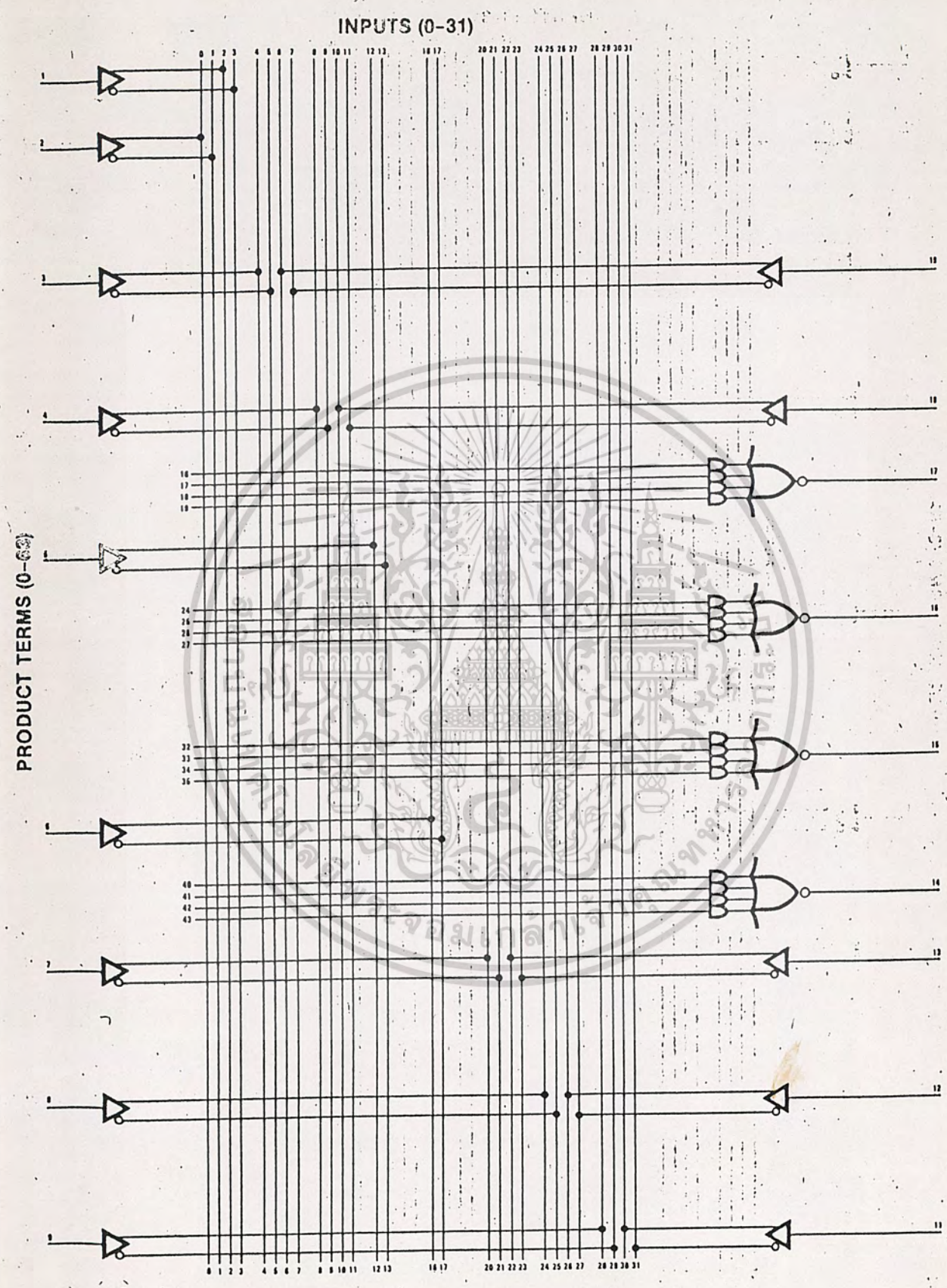


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

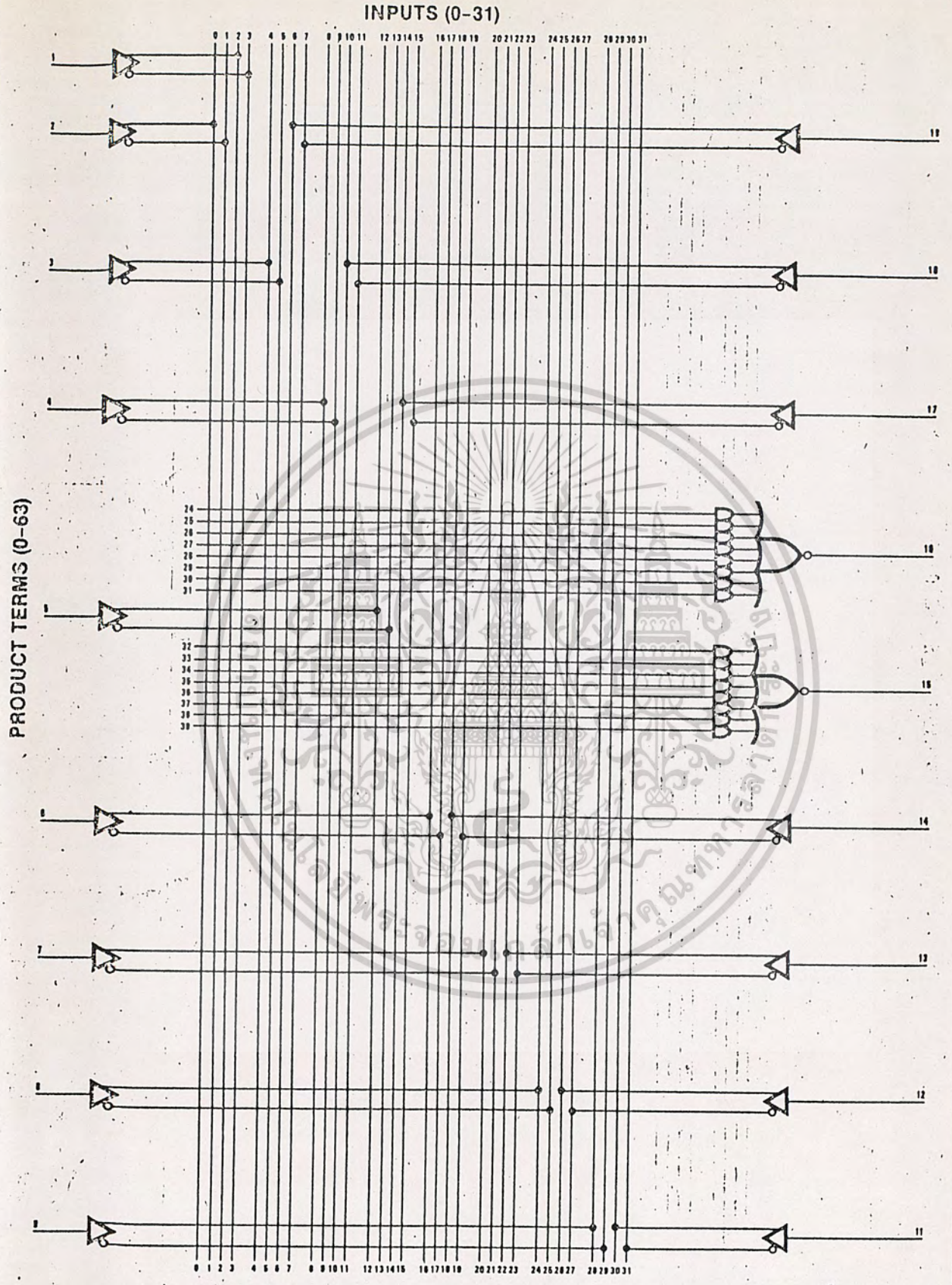


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Logic Diagram PAL14L4

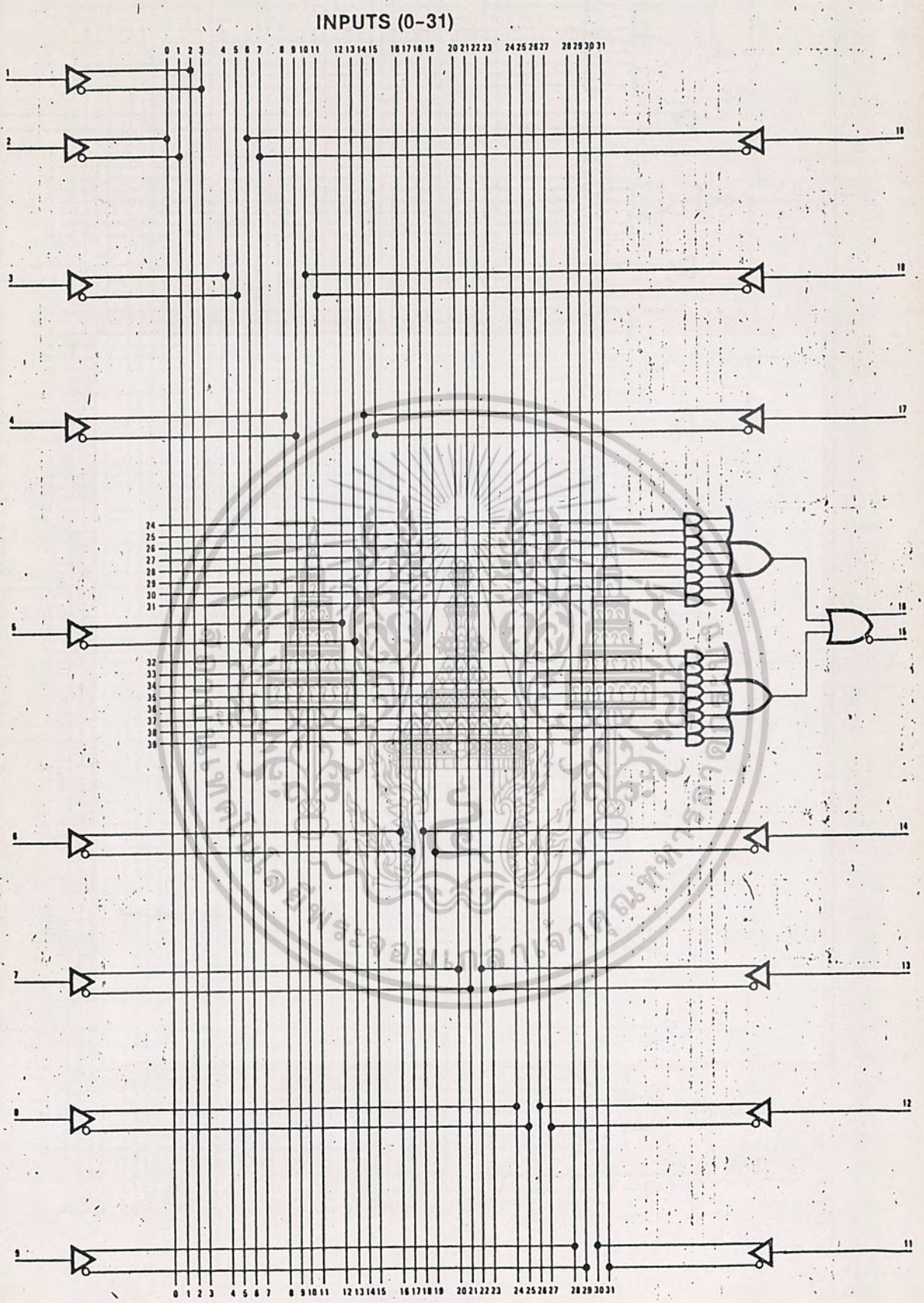


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



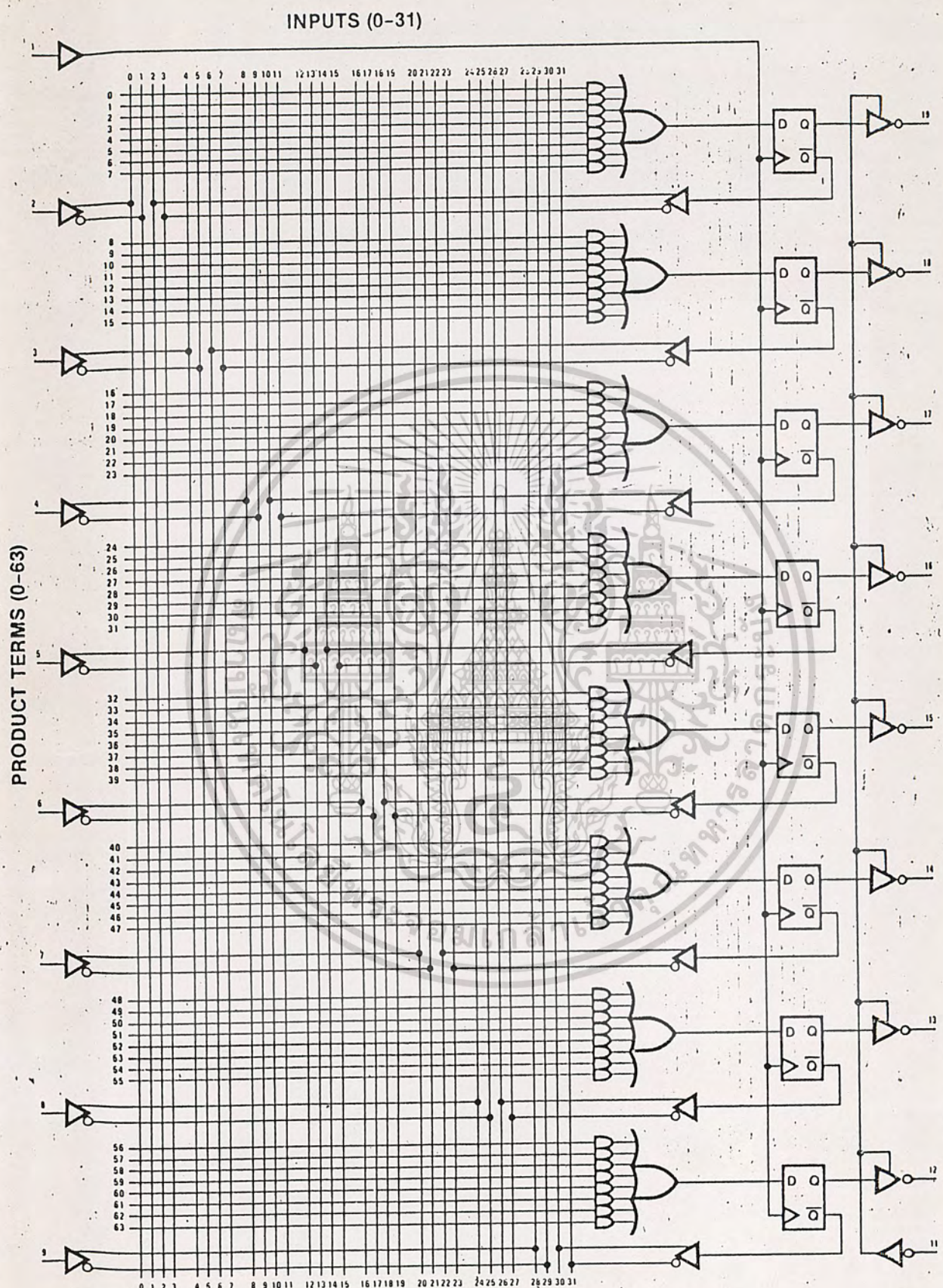
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Logic Diagram PAL16C1

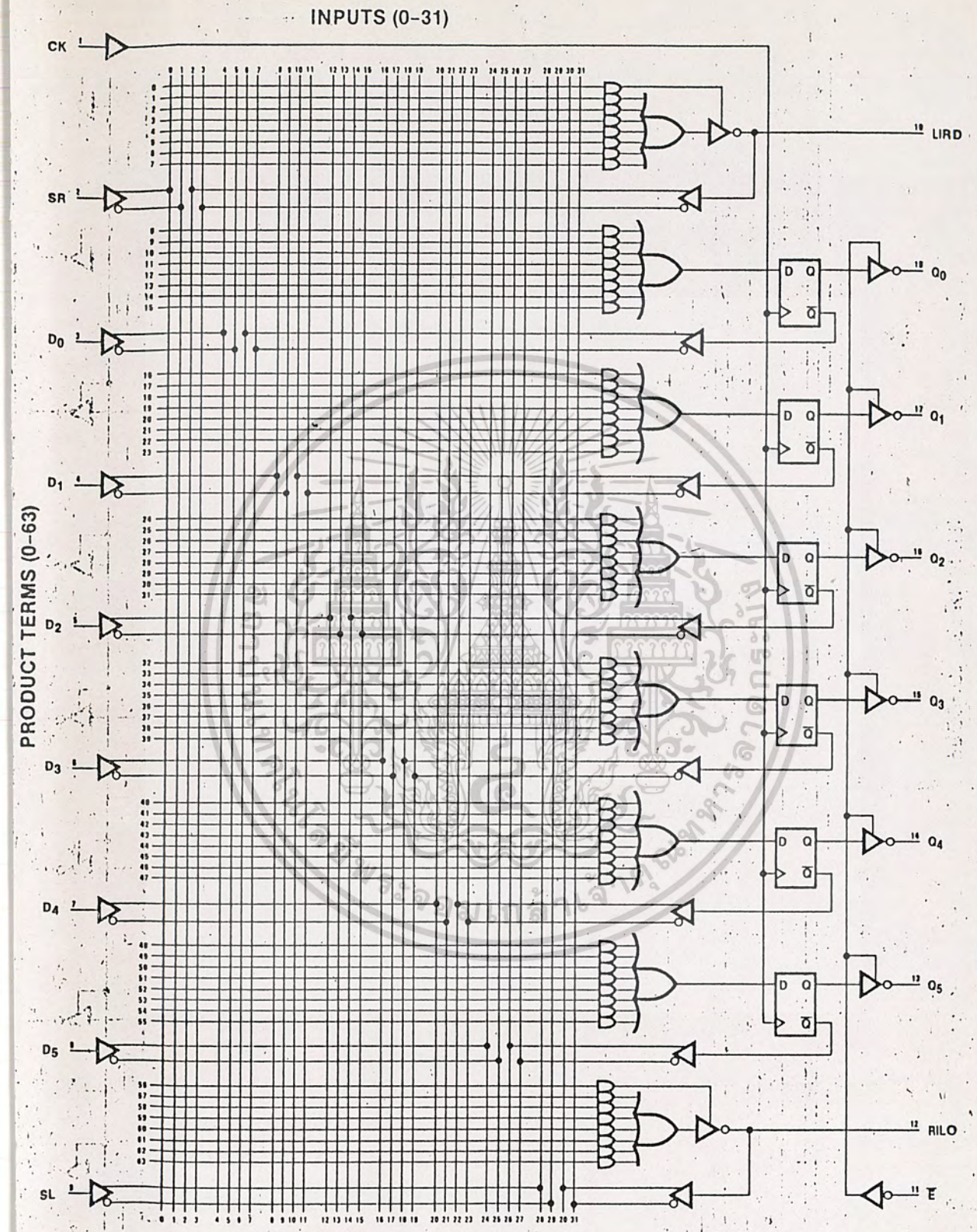


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Logic Diagram PAL16R8

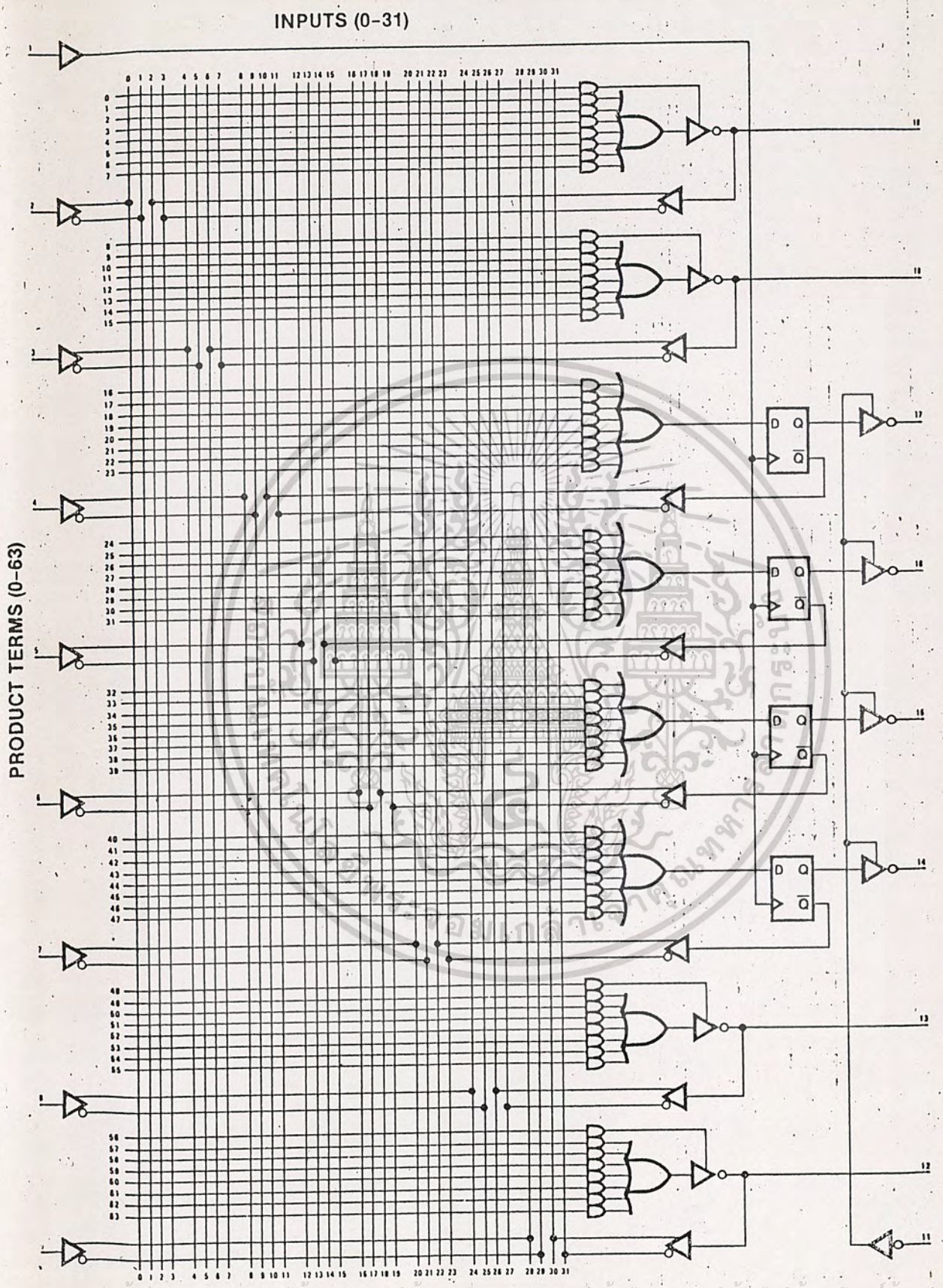


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



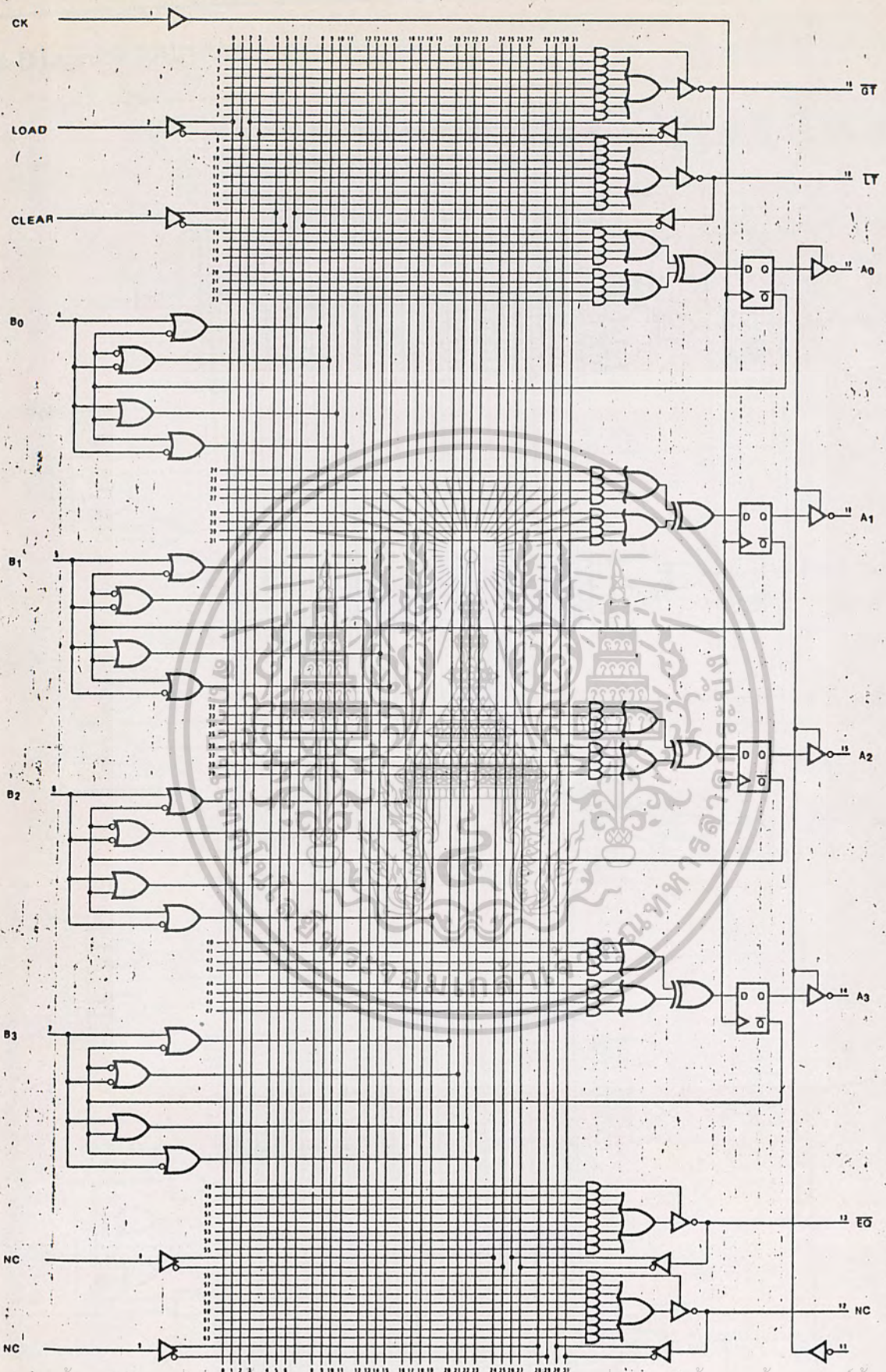
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Logic Diagram PAL16R4



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Logic Diagram PAL16X4



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Synertek

SY6521/SY6821
SY6521A/SY68B21

Peripheral Interface Adapter (PIA)

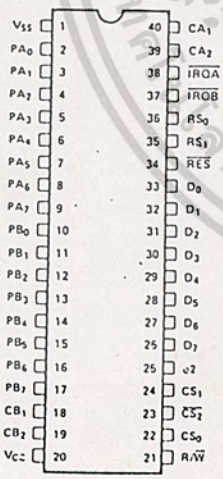
Features

- Extended Performance Version of SY6520
- Single +5V Power Supply
- Two 8-bit Bi-directional I/O Ports with Individual Data Direction Control
- CMOS-Compatible Peripheral Port A lines
- Automatic "Handshake" Control of Data Transfers
- Programmable Interrupt Capability
- Automatic Initialization on Power Up
- 1 and 2 MHz Versions
- Direct Replacement for MC6821

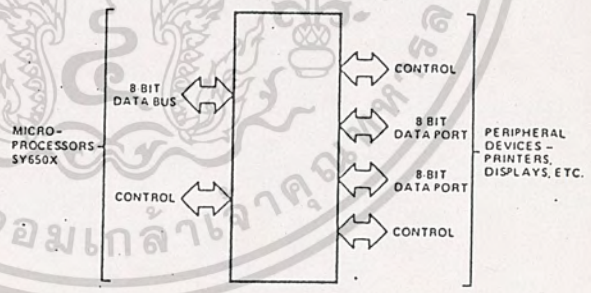
Description

The SY6521 Peripheral Interface Adapter (PIA) is designed to provide a broad range of peripheral control to microcomputer systems. It is functionally compatible with the SY6520, but with more drive capability and improved performance. Control of peripheral devices is accomplished through two 8-bit bi-directional I/O ports. Each I/O line may be programmed to be either an input or an output. In addition, four peripheral control lines are provided to perform "handshaking" during data transfers.

Pin Configuration



Basic SY6521 Interface Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings*

RATING	SYMBOL	VALUE	UNIT
Supply Voltage	V_{CC}	-0.3 to +7.0	V
Input Voltage	V_{IN}	-0.3 to +7.0	V
Operating Temperature Range	T_A	0 to +70	°C
Storage Temperature Range	T_{STG}	-55 to +150	°C

Comment*

This device contains circuitry to protect the inputs against damage due to high static voltages, however, it is advised that normal precautions be taken to avoid application of any voltage higher than maximum rated voltages to this circuit.

D.C. Characteristics

(V_{CC} = 5.0V ± 5%, V_{SS} = 0, T_A = 0°C to 70°C unless otherwise noted)

Symbol	Characteristic	Min.	Max.	Unit
V_{IH}	Input High Voltage	+2.0	V_{CC}	V
V_{IL}	Input Low Voltage	-0.3	+0.8	V
I_{IN}	Input Leakage Current V_{IN} = 0 to 5.0V R/$\overline{W}$, $\overline{Reset}$, RS_0, RS_1, CS_0, CS_1, $\overline{CS_2}$, CA_1, CB_1, ϕ_2	–	±2.5	μA
I_{TSI}	Three-State (Off-State Input Current) (V_{IN} = 0.4 to 2.4V, V_{CC} = max), D_0-D_7, PB_0-PB_7, CB_2	–	±10	μA
I_{IH}	Input High Current (V_{IH} = 2.4V), PA_0-PA_7, CA_2	-200	–	μA
I_{IL}	Input Low Current (V_{IL} = 0.4V), PA_0-PA_7, CA_2	–	2.4	mA
V_{OL}	Output Low Voltage (I_L = 3.2 mA), $\overline{IRQA}$, $\overline{IRQB}$	–	0.4	V
V_{OH}	Output High Voltage (I_L = -205 μA), D_0-D_7	2.4	–	V
V_{OL}	Output Low Voltage (I_L = 3.2mA), PA_0-PA_7, PB_0-PB_7, CA_2, CB_2	–	0.4	V
V_{OH}	Output High Voltage (I_H = -200μA), PA_0-PA_7, PB_0-PB_7, CA_2, CB_2	2.4	–	V
I_{OH}	Output High Current (Direct Transistor Drive Outputs) (V_{OUT} = 1.5V), PB_0-PB_7, CB_2	-1.0	-10.0	mA
I_{OFF}	Output Leakage Current (Off-State), $\overline{IRQA}$, $\overline{IRQB}$	–	10	μA
P_D	Power Dissipation (V_{CC} = 5.25V)	–	500	mW
C_{IN}	Input Capacitance (V_{IN} = 0, T_A = 25°C, f = 1.0 MHz) D_0-D_7, PA_0-PA_7, PB_0-PB_7, CA_2, CB_2 R/W, $\overline{Reset}$, RS_0, RS_1, CS_0, CS_1, $\overline{CS_2}$, CA_1, CB_1, ϕ_2	 – – –	 10 7.0 20	 pF pF pF
C_{OUT}	Output Capacitance (V_{IN} = 0, T_A = 25°C, f = 1.0 MHz)	–	10	pF

Note: Negative sign indicates outward current flow, positive indicates inward flow.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

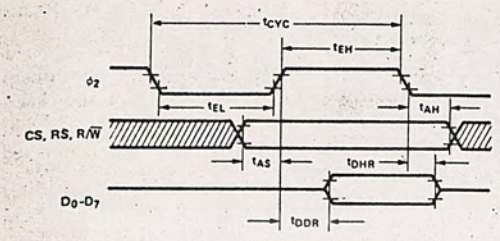


Figure 1. Read Timing Characteristics

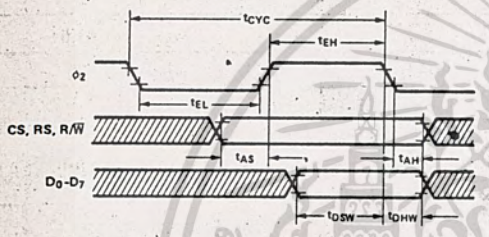


Figure 2. Write Timing Characteristics

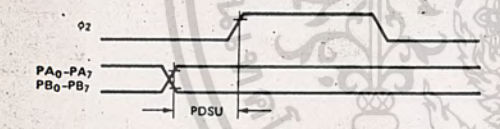


Figure 3. Peripheral Data Setup Time

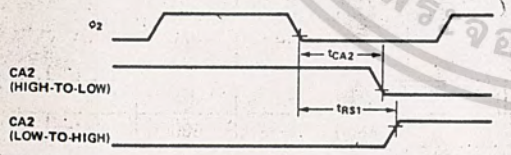


Figure 4. CA_2 Timing

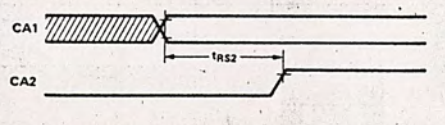


Figure 5. CA_1/CA_2 Timing

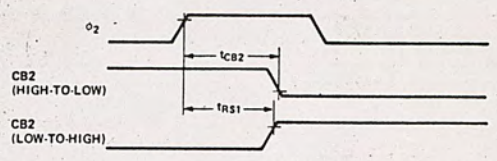


Figure 6. CB_2 Timing

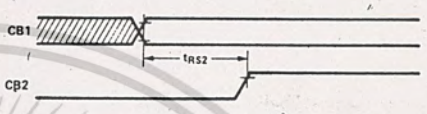


Figure 7. CB_1/CB_2 Handshake Timing

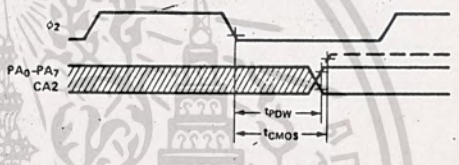


Figure 8. PA Port Delay Time

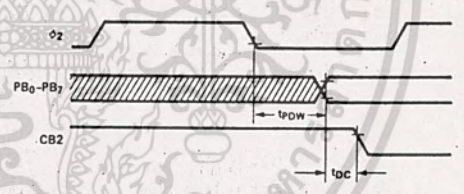


Figure 9. PB Port Delay Time

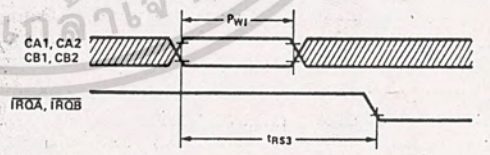


Figure 10. Interrupt Timing

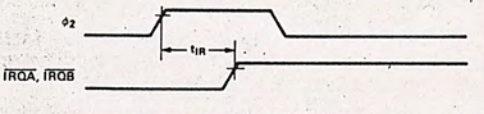


Figure 11. Interrupt Clear Timing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Processor Interface Timing

(V_{CC} = 5V ± 5%, T_A = 0°C to 70°C unless otherwise noted)

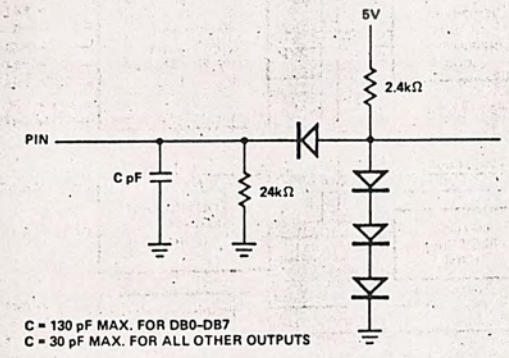
Symbol	Parameter	SY6521		SY6521A		Unit
		Min.	Max.	Min.	Max.	
t_{CY}	Cycle Time	1000	–	500	–	ns
t_{EH}	ϕ_2 Pulse Width	440	–	200	–	ns
t_{EL}	ϕ_2 Pulse Delay	430	–	210	–	ns
t_{AS}	CS, RS, R/$\overline{W}$ Setup Time	160	–	70	–	ns
t_{AH}	CS, RS, R/$\overline{W}$ Hold Time	10	–	10	–	ns
t_{DDR}	Data Delay Time, Read Cycle	–	320	–	180	ns
t_{DHR}	Data Hold Time, Read Cycle	10	–	10	–	ns
t_{DSW}	Data Setup Time, Write Cycle	195	–	60	–	ns
t_{DHW}	Data Hold Time, Write Cycle	10	–	10	–	ns

Peripheral Interface Timing

(V_{CC} = 5V ± 5%, T_A = 0°C to 70°C unless otherwise noted)

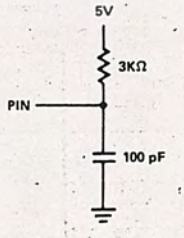
Symbol	Parameter	SY6521		SY6521A		Unit
		Min.	Max.	Min.	Max.	
t_{PDSU}	Peripheral Data Setup Time	200	–	100	–	ns
t_{CA2}	CA_2 Delay Time, High-to-Low	–	1.0	–	0.5	μs
t_{RS1}	CA_2 Delay Time, Low-to-High	–	1.0	–	0.5	μs
t_{RS2}	CA_2 Delay Time, Handshake Mode	–	2.0	–	1.0	μs
t_{CB2}	CB_2 Delay Time, High-to-Low	–	1.0	–	0.5	μs
t_{RS1}	CB_2 Delay Time, Low-to-High	–	1.0	–	0.5	μs
t_{RS2}	CB_2 Delay Time, Handshake Mode	–	2.0	–	1.0	μs
t_{PDW}	Peripheral Port Delay Time	–	1.0	–	0.5	μs
t_{CMOS}	Peripheral Port Delay Time (CMOS)	–	2.0	–	1.0	μs
t_{DC}	CB_2 Delay Time from Data Valid	20	–	20	–	ns
P_{WI}	Interrupt Input Pulse Width	500	–	500	–	ns
t_{RS3}	Interrupt Response Time	–	1.0	–	1.0	μs
t_{IR}	Interrupt Clear Delay	–	1.6	–	0.85	μs
t_R, t_F	Rise and Fall Times – CA_1, CA_2, CB_1, CB_2	–	1.0	–	1.0	μs

Test Load



C = 130 pF MAX. FOR DB0–DB7
C = 30 pF MAX. FOR ALL OTHER OUTPUTS

OPEN COLLECTOR OUTPUT TEST LOAD



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Interface Signal Description

$\overline{RES}$ (Reset)

This signal is used to initialize the PIA. A low signal on the $\overline{RES}$ input causes all internal registers to be cleared.

ϕ_2 (Input Clock)

This input is the system ϕ_2 clock and is used to trigger all data transfers between the microprocessor and the PIA.

R/$\overline{W}$ (Read/Write)

This signal is generated by the microprocessor and is used to control the direction of data transfers. A high on the R/$\overline{W}$ signal permits the processor to read data supplied by the PIA; a low on the R/$\overline{W}$ signal permits the processor to Write into the PIA.

$\overline{IRQA}$, $\overline{IRQB}$ (Interrupt Requests)

$\overline{IRQA}$ and $\overline{IRQB}$ are interrupt lines generated by the PIA for ports A and B respectively. These signals are active low signals and have open-drain outputs, thus allowing multiple IRQ signals from multiple PIA's to be wire-ORed together before connecting to the processor IRQ signal input.

D_0-D_7 (Data Bus)

These eight data bus lines are used to transfer data information between the processor and the PIA. These signals are bi-directional and are normally high-impedance except when selected for a read operation.

CS0, CS1, $\overline{CS2}$ (Chip Selects)

The PIA is selected when CS0 and CS1 are high and $\overline{CS2}$ is low. These three chip select lines are normally connected to the processor address lines either directly or through external decoder circuits.

RS0, RS1 (Register Selects)

These two signals are used to select the various registers inside the PIA.

Internal Architecture

The SY6520 is organized into two independent sections referred to as the "A Side" and the "B Side." Each section consists of a Control Register (CRA, CRB), Data Direction Register (DDRA, DDRB), Output Register (ORA, ORB), Interrupt Status Control and the buffers necessary to drive the Peripheral Interface buses. Figure 12 is a block diagram of the SY6521.

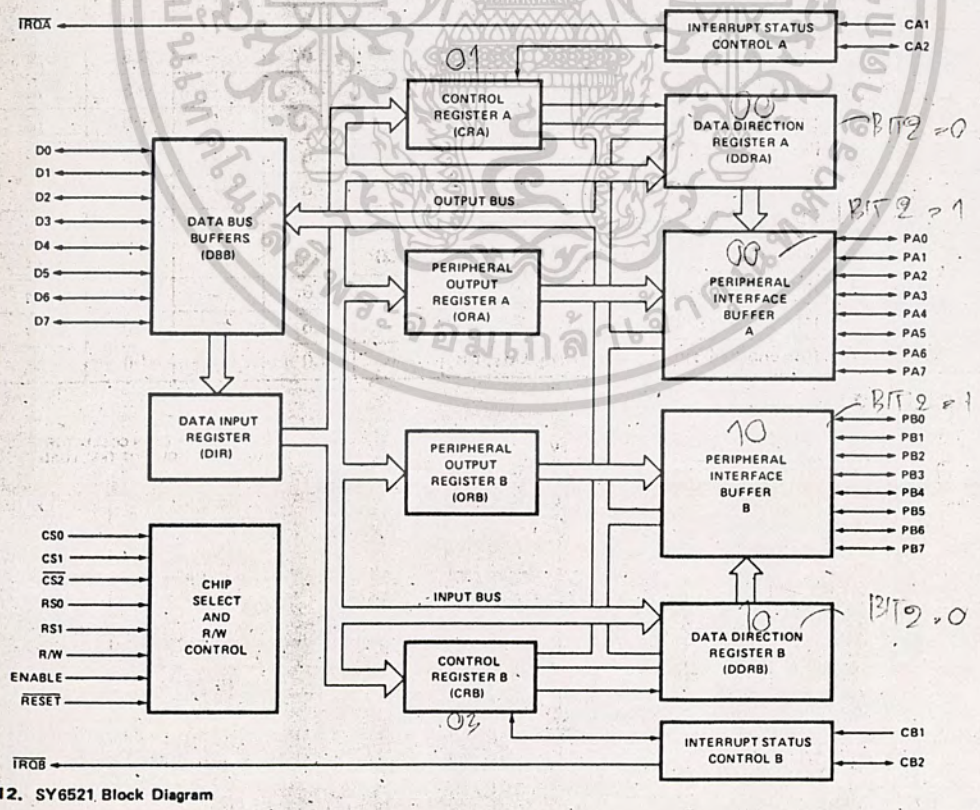


Figure 12. SY6521 Block Diagram

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	7	6	5 4 3	2	1 0
CRA	IRQA1	IRQA2	CA2 Control	DDRA Access	CA1 Control
	7	6	5 4 3	2	1 0
CRB	IRQB1	IRQB2	CB2 Control	DDRB Access	CB1 Control

Figure 13. Control Registers

Data Input Register

When the microprocessor writes data into the SY6521 the data which appears on the data bus during the Phase Two clock pulse is latched into the Data Input Register. It is then transferred into one of six internal registers of the SY6521 after the trailing edge of Phase Two. This assures that the data on the peripheral output lines will make smooth transitions from high to low or from low to high and the voltage will remain stable except when it is going to the opposite polarity.

Control Registers (CRA and CRB)

Figure 4 illustrates the bit designation and functions in the Control Registers. The Control Registers allow the microprocessor to control the operation of the Interrupt Control inputs (CA1, CA2, CB1, CB2), and Peripheral Control outputs (CA2, CB2). Bit 2 in each register controls the addressing of the Data Direction Registers (DDRA, DDRB) and the Output Registers (ORA, ORB). In addition, two bits (bit 6 and 7) are provided in each control register to indicate the status of the interrupt input lines (CA1, CA2, CB1, CB2). These interrupt status bits (IRQA1, IRQB1) are normally interrogated by the microprocessor during the interrupt service routine to determine the source of an active interrupt. These are the interrupt lines which drive the interrupt input ($\overline{IRQ}$, $\overline{NMI}$) of the microprocessor.

Data Direction Registers (DDRA, DDRB)

The Data Direction Registers allow the processor to program each line in the 8-bit Peripheral I/O port to be either an input or an output. Each bit in DDRA controls the corresponding line in the Peripheral A port and each bit in DDRB controls the corresponding line in the Peripheral B port. Placing a "0" in a bit position in the Data Direction Register causes the corresponding Peripheral I/O line to act as an input; a "1" causes it to act as an output.

Peripheral Output Registers (ORA, ORB)

The Peripheral Output Registers store the output data which appears on the Peripheral I/O port. Writing a "0" into a bit in ORA causes the corresponding line on the Peripheral A port to go low ($< 0.4V$) if that line is programmed to act as an output. A "1" causes the corresponding output to go high. The lines of the Peripheral B port are controlled by ORB in the same manner.

Interrupt Status Control

The four interrupt/peripheral control lines (CA1, CA2, CB1, CB2) are controlled by the Interrupt Status Control logic (A, B). This logic interprets the contents of the corresponding Control Register, detects active transitions on the interrupt inputs and performs those operations necessary to assure proper operation of these four peripheral interface lines.

Peripheral Interface Buffers (A, B) and Data Bus Buffers (DBB)

These Buffers provide the necessary current and voltage drive on the peripheral I/O ports and data bus to assure proper system operation and to meet the device specifications.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Functional Description

Bit 2 (DDR) in each Control Register (CRA and CRB) controls the accessing to the Data Direction Register or the Peripheral interface. If bit 2 is a "1", a Peripheral Output register (ORA, ORB) is selected, and if bit 2 is a "0", a Data Direction Register (DDRA, DDRB) is selected. The Data Direction Register Access Control bit, together with the Register Select lines (RS0, RS1) selects the various internal registers as shown in Figure 14.

In order to write data into DDRA, ORA, DDRB, or ORB registers, bit 2 in the proper Control Register must first be set. The desired register may then be accessed with the address determined by the address interconnect technique used.

Register Select Lines (RS0), (RS1)

These two register select lines are used to select the various registers inside the SY6521. These input lines are used in conjunction with internal control registers to select a particular register that is to be accessed by the microprocessor. These lines are normally connected to microprocessor address output lines. These lines operate in conjunction with the chip-select inputs to allow the microprocessor to address a single 8-bit register within the microprocessor address space. This register may be an internal register (CRA, ORA, etc.) or it may be a Peripheral I/O port.

The processor may write directly into the Control Registers (CRA, CRB), the Data Direction Registers (DDRA, DDRB) and the Peripheral Output Registers (ORA, ORB). In addition, the processor may directly read the contents of the Control Registers and the Data Direction Registers. Accessing the Peripheral Output Register for the purpose of reading data back into the processor operates differently on the ORA and the ORB registers and therefore are discussed separately below.

Reading the Peripheral A I/O Port

The Peripheral A I/O port consists of 8 lines which can be programmed to act as inputs or outputs. When programmed to act as outputs, each line reflects the contents of the corresponding bit in the Peripheral Output Register. When programmed to act as inputs, these lines will go high or low depending on the input data. The Peripheral Output Register (ORA) has no effect on those lines programmed to act as inputs. The eight lines of the Peripheral A I/O port therefore contain either input or output data depending on whether the line is programmed to act as an input or an output.

Performing a Read operation with RS1 = 0, RS0 = 0 and the Data Direction Register Access Control bit (CRA-2) = 1, directly transfers the data on the Peripheral A I/O lines into the processor (via the data bus). This will contain both the input and output data. The processor must be programmed to recognize and interpret only those bits which are important to the particular peripheral operation being performed.

Since the processor always reads the Peripheral A I/O port pins instead of the actual Peripheral Output Register (ORA), it is possible for the data read into the processor to differ from the contents of the Peripheral Output Register for an output line. This is true when the I/O pin is not allowed to go to a full +2.4V DC when the Peripheral Output register contains a logic 1. In this case, the processor will read a 0 from the Peripheral A pin, even though the corresponding bit in the Peripheral Output register is a 1.

Reading the Peripheral B I/O Port

Reading the Peripheral B I/O port yields a combination of input and output data in a manner similar to the Peripheral A port. However, data is read directly from the Peripheral B Output Register (ORB) for

Register Select Pin		Data Direction Register Access Control Bit		Register Selected
RS1	RS0	CRA-2	CRB-2	
0	0	1	–	Peripheral Interface A
0	0	0	–	Data Direction Register A
0	1	–	–	Control Register A
1	0	–	1	Peripheral Interface B
1	0	–	0	Data Direction Register B
1	1	–	–	Control Register B

Figure 14. Register Addressing

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

those lines programmed to act as outputs. It is therefore possible to load down the Peripheral B Output lines without causing incorrect data to be transferred back into the processor on a Read operation.

Interrupt Request Lines ($\overline{IRQA}$, $\overline{IRQB}$)

The active low Interrupt Request lines ($\overline{IRQA}$ and $\overline{IRQB}$) act to interrupt the microprocessor either directly or through external interrupt priority circuitry. These lines are "open drain" and are capable of sinking 1.6 milliamps from an external source. This permits all interrupt request lines to be tied together in a "wired-OR" configuration. The "A" and "B" in the titles of these lines correspond to the "A" peripheral port and the "B" peripheral port. Hence each interrupt request line services one peripheral data port.

Each Interrupt Request line has two interrupt flag bits which can cause the Interrupt Request line to go low. These flags are bits 6 and 7 in the two Control Registers. These flags act as the link between the peripheral interrupt signals and the microprocessor interrupt inputs. Each flag has a corresponding interrupt disable bit which allows the processor to enable or disable the interrupt from each of the four interrupt inputs (CA1, CA2, CB1, CB2).

The four interrupt flags are set by active transitions of the signal on the interrupt input (CA1, CA2, CB1, CB2).

Control of $\overline{IRQA}$

Control Register A bit 7 is always set by an active transition of the CA1 interrupt input signal. Interrupting from this flag can be disabled by setting bit 0 in the Control Register A (CRA) to a logic 0. Likewise, Control Register A bit 6 can be set by an active transition of the CA2 interrupt input signal. Interrupting from this flag can be disabled by setting bit 3 in the Control Register to a logic 0.

Both bit 6 and bit 7 in CRA are reset by a "Read Peripheral Output Register A" operation. This is defined as an operation in which the proper chip-select and register-select signals are provided to allow the processor to read the Peripheral A I/O port.

Control of $\overline{IRQB}$

Control of $\overline{IRQB}$ is performed in exactly the same manner as that described above for $\overline{IRQA}$. Bit 7 in CRB is set by an active transition on CB1; interrupting from this flag is controlled by CRB bit 0. Likewise, bit 6 in CRB is set by an active transition on CB2; interrupting from this flag is controlled by CRB bit 3.

Also, both bit 6 and bit 7 are reset by a "Read Peripheral B Output Register" operation.

SUMMARY:
$\overline{IRQA}$ goes low when CRA-7 = 1 and CRA-0 = 1 or when CRA-6 = 1 and CRA-3 = 1
$\overline{IRQB}$ goes low when CRB-7 = 1 and CRB-0 = 1 or when CRB-6 = 1 and CRB-3 = 1

It should be stressed at this point that the flags act as the link between the peripheral interrupt signals and the processor interrupt inputs. The interrupt disable bits allow the processor to control the interrupt function.

Interface Between SY6521 and Peripheral Devices

The SY6521 provides two 8-bit bi-directional ports and 4 interrupt/control lines for interfacing to peripheral devices. These ports and the associated interrupt/control lines are referred to as the "A" side and the "B" side. Each side has its own unique characteristics and will therefore be discussed separately below.

Peripheral I/O Ports

The Peripheral A and Peripheral B I/O ports allow the microprocessor to interface to the input lines on the peripheral device by loading data into the Peripheral Output Register. They also allow the processor to interface with the peripheral device output lines by reading the data on the Peripheral Port input lines directly onto the data bus and into the internal registers of the processor.

Peripheral A I/O Port (PA0-PA7)

Each of the Peripheral I/O lines can be programmed to act as an input or an output. This is accomplished by setting a "1" in the corresponding bit in the Data Direction Register for those lines which are to act as outputs. A "0" in a bit of the Data Direction Register causes the corresponding Peripheral I/O lines to act as an input.

The buffers which drive the Peripheral A I/O lines contain "passive" pull-ups as shown in Figure 15. These pull-up devices are resistive in nature and therefore allow the output voltage to go to V_{CC} for a logic 1. The switches can sink a full 1.6mA, making these buffers capable of driving one standard TTL load.

In the input mode, the pull-up devices shown in Figure 15 are still connected to the I/O pin and still supply current to this pin. For this reason, these lines represent one standard TTL load in the input mode.

Peripheral B I/O Port (PB0-PB7)

The Peripheral B I/O port duplicates many of the functions of the Peripheral A port. The process of programming these lines to act as an input or an out-

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

put has been discussed previously. Likewise, the effect of reading or writing this port has been discussed. However, there are several characteristics of the buffers driving these lines which affect their use in peripheral interfacing.

The Peripheral B I/O port buffers are push-pull devices as shown in Figure 16. The pull-up devices are switched "OFF" in the "0" state and "ON" for a logic 1. Since these pull-ups are active devices, the logic "1" voltage is not guaranteed to go higher than +2.4V. They are TTL compatible but are not CMOS compatible.

However, the active pull-up devices can source up to 1mA at 1.5V. This current drive capability is provided to allow direct connection to Darlington transistor switches. This allows very simple control of relays, lamps, etc.

Because these outputs are designed to drive transistors directly, the output data is read directly from the Peripheral Output Register for those lines programmed to act as inputs.

The final characteristic which is a function of the Peripheral B push-pull buffers is the high-impedance input state. When the Peripheral B I/O lines are programmed to act as inputs, the output buffer enters the high impedance state.

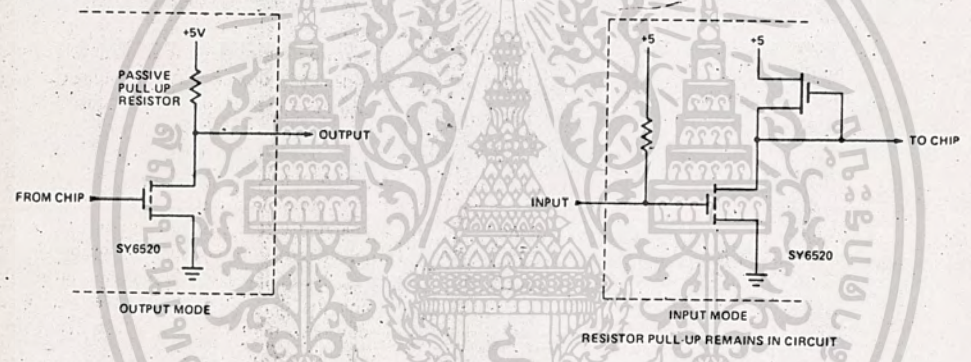


Figure 15. Port A Buffer Circuit (PA_0-PA_7)

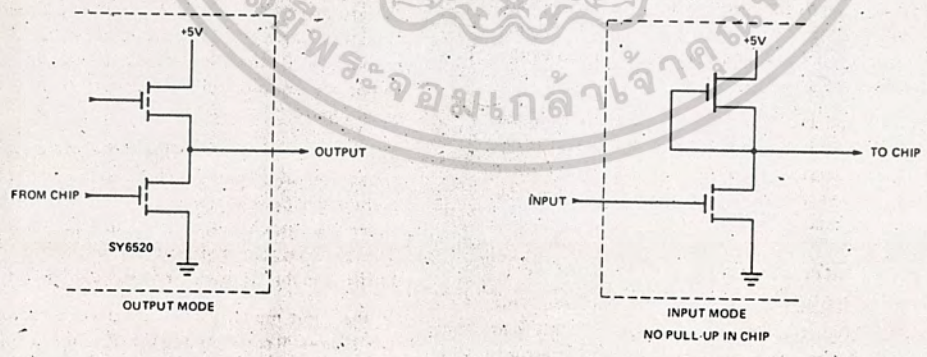


Figure 16. Port B Buffer Circuit (PB_0-PB_7)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Interrupt Input/Peripheral Control Lines (CA1, CA2, CB1, CB2)

The four interrupt input/peripheral control lines provide a number of special peripheral control functions. These lines greatly enhance the power of the two general purpose interface ports (PA0-PA7, PB0-PB7). Figure 17 summarizes the operation of these control lines.

Peripheral A Interrupt Input/Peripheral Control Lines (CA1, CA2)

CA1 is an interrupt input only. An active transition of the signal on this input will set bit 7 of the Control Register A to a logic 1. The active transition can be programmed by setting a "0" in bit 1 of the CRA if the interrupt flag (bit 7 of CRA) is to be set on a negative transition of the CA1 signal or a "1" if it is to be set on a positive transition.

NOTE: A negative transition is defined as a transition from a high to a low, and a positive transition is defined as a transition from a low to a high voltage.

Setting the interrupt flag will interrupt the processor through $\overline{IRQA}$ if bit 0 of CRA is a 1 as described previously.

CA2 can act as a totally independent interrupt input or as a peripheral control output. As an input (CRA, bit 5 = 0) it acts to set the interrupt flag, bit 6 of CRA, to a logic 1 on the active transition selected by bit 4 of CRA.

These control register bits and interrupt inputs serve the same basic function as that described above for CA1. The input signal sets the interrupt flag which serves as the link between the peripheral device and the processor interrupt structure. The interrupt disable bit allows the processor to exercise control over the system interrupts.

In the Output mode (CRA, bit 5 = 1), CA2 can operate independently to generate a simple pulse each time the microprocessor reads the data on the Peripheral A I/O port. This mode is selected by setting CRA, bit 4 to a "0" and CRA, bit 3 to a "1". This pulse output can be used to control the counters, shift registers, etc. which make sequential data available on the Peripheral input lines.

A second output mode allows CA2 to be used in conjunction with CA1 to "handshake" between the processor and the peripheral device. On the A side, this technique allows positive control of data transfers from the peripheral device into the microprocessor. The CA1 input signals the processor that data is available by interrupting the processor. The processor reads the data and sets CA2 low. This signals the peripheral device that it can make new data available.

The final output mode can be selected by setting bit 4 of CRA to a 1. In this mode, CA2 is a simple peripheral control output which can be set high or low by setting bit 3 of CRA to a 1 or a 0 respectively.

Peripheral B Interrupt Input/Peripheral Control Lines (CB1, CB2)

CB1 operates as an interrupt input only in the same manner as CA1. Bit 7 of CRB is set by the active transition selected by bit 0 of CRB. Likewise, the CB2 input mode operates exactly the same as the CA2 input modes. The CB2 output modes, CRB bit 5 = 1, differ somewhat from those of CA2. The pulse output occurs when the processor writes data into the Peripheral B Output Register. Also, the "handshaking" operates on data transfers from the processor into the peripheral device.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CA1/CB1 Control

CRA (CRB) Bit 1	CRA (CRB) Bit 0	Active Transition of Input Signal*	IRQA (IRQB) Interrupt Outputs
0	0	Negative	Disable – remain high
0	1	Negative	Enable – goes low when bit 7 in CRA (CRB) is set by active transition of signal on CA1 (CB1)
1	0	Positive	Disable – remain high
1	1	Positive	Enable – as explained above

*Note: Bit 7 of CRA (CRB) will be set to a logic 1 by an active transition of the CA1 (CB1) signal. This is independent of the state of Bit 0 in CRA (CRB).

CA2/CB2 Input Modes

CRA (CRB) Bit 5	CRA (CRB) Bit 4	CRA (CRB) Bit 3	Active Transition of Input Signal*	IRQA (IRQB) Interrupt Outputs
0	0	0	Negative	Disable – remains high
0	0	1	Negative	Enable – goes low when bit 6 in CRA (CRB) is set by active transition of signal on CA2 (CB2)
0	1	0	Positive	Disable – remains high
0	1	1	Positive	Enable – as explained above

*Note: Bit 6 of CRA (CRB) will be set to a logic 1 by an active transition of the CA2 (CB2) signal. This is independent of the state of Bit 3 in CRA (CRB).

CA2 Output Modes

CRA Bit 5	CRA Bit 4	CRA Bit 3	Mode	Description
1	0	0	"Handshake" on Read	CA2 is set high on an active transition of the CA1 interrupt input signal and set low by a microprocessor "Read A Data" operation. This allows positive control of data transfers from the peripheral device to the microprocessor.
1	0	1	Pulse Output	CA2 goes low for one cycle after a "Read A Data" operation. This pulse can be used to signal the peripheral device that data was taken.
1	1	0	Manual Output	CA2 set low
1	1	1	Manual Output	CA2 set high

CB2 Output Modes

CRB Bit 5	CRB Bit 4	CRB Bit 3	Mode	Description
1	0	0	"Handshake" on Write	CB2 is set low on microprocessor "Write B Data" operation and is set high by an active transition of the CB1 interrupt input signal. This allows positive control of data transfers from the microprocessor to the peripheral device.
1	0	1	Pulse Output	CB2 goes low for one cycle after a microprocessor "Write B Data" operation. This can be used to signal the peripheral device that data is available.
1	1	0	Manual Output	CB2 set low
1	1	1	Manual Output	CB2 set high

Figure 17. Summary of Operation of Control Lines

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

54138/74138 3-Line-to-8-Line Decoder

	Schottky TTL					High-Speed TTL					Low-Power Schottky TTL					Standard TTL					Low-Power TTL				
	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF	Device Type	Package C	P	M	CF
T.I.	SN54S138	J⑯			W⑯						SN54LS138	J⑯			W⑯										
	SN74S138	J⑯	N⑯								SN74LS138	J⑯	N⑯												
FAIRCHILD	/FM93S138	D⑯									FM54LS138/FM93LS138	D⑯			F⑯										
	FC74S138/FC93S138	D⑯									FC74LS138/FC93LS138	D⑯	P⑯		F⑯										
MOTOROLA											SN74LS138		P⑯												
N.S.C.	DM74S138		⑯								DM74LS138		⑯												
											DM54LS138		⑯												
PHILIPS	N74S138		⑯								N74LS138		⑯												
SIGNETICS	S54S138	F⑯	B⑯		W⑯																				
	N74S138	F⑯	B⑯								N74LS138		A⑯												
SIEMENS																									
FUJITSU											74LS138		M⑯												
HITACHI											HD74LS138		P⑯												
MITSUBISHI	M74S138		P⑯								M74LS138		P⑯												
NEC											74LS138		C⑯												
TOSHIBA																									

Electrical Characteristics SN54LS138/SN74LS138

absolute maximum ratings over operating free-air temperature range

Supply voltage, V_{CC}	7V	Operating free-air temperature range	SN54LS	−55°C to 125°C
Input voltage	7V		SN74LS	0°C to 70°C
		Storage temperature range		−65°C to 150°C

recommended operarating conditions

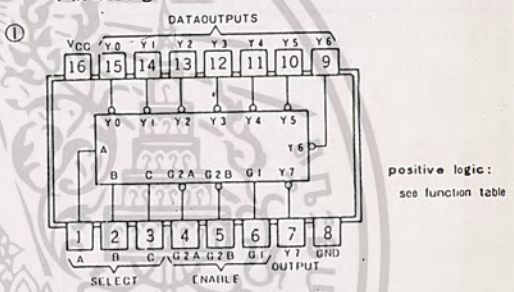
	SN54LS138 MIN	NOM	MAX	SN74LS138 MIN	NOM	MAX	UNIT
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			400			400	μA
Low-level output current, I_{OL}			4			8	mA
Operating free-air temperature, T_A	55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range

PARAMETER ◆		TEST CONDITIONS†		MIN	TYP‡	MAX	UNIT
V_{IH}	High-level input voltage			2			V
V_{IL}	Low-level input voltage					0.8	V
V_I	Input clamp voltage	V_{CC} = MIN, I_I = −18mA				1.5	V
V_{OH}	High-level output voltage	V_{CC} = MIN, V_{IH} = 2V, V_{IL} = 0.8V, I_{OH} = 400μA	SN54LS	2.5	3.4		V
			SN74LS	2.7	3.4		
V_{OL}	Low-level output voltage	V_{CC} = MIN, V_{IH} = 2V, V_{IL} = 0.8V, I_{OL} = 8mA			0.35	0.5	V
I_I	Input current at maximum input voltage	V_{CC} = MAX, V_I = 7V				0.1	mA
I_{IH}	High-level input current	V_{CC} = MAX, V_I = 2.7V				20	μA
I_{IL}	Low-level input current	V_{CC} = MAX, V_I = 0.4V				0.4	mA
I_{OS}	Short-circuit output current ◆	V_{CC} = MAX		−20		−100	mA
I_{CC}	Supply current	V_{CC} = MAX, Outputs enabled and open			6.3	10	mA
t_{PLH}	from Binary select to Any output	V_{CC} = 5V, T_A = 25°C, C_L = 15pF, R_L = 2kΩ, levels of delay	2		13	20	ns
t_{PHL}					27	41	
t_{PLH}			3		18	27	ns
t_{PHL}					26	39	
t_{PLH}	from Enable to Any output		2		12	18	ns
t_{PHL}					21	32	
t_{PLH}			3		17	26	ns
t_{PHL}					25	38	

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable device type.
‡ All typical values are at V_{CC} = 5V, T_A = 25°C.
◆ Not more than one output should be shorted at a time, and duration of the short-circuit test should not exceed one second.
■ t_{PLH} = propagation delay time, low-to-high-level output
t_{PHL} = propagation delay time, high-to-low-level output

Pin Assignment (Top View)



positive logic: see function table

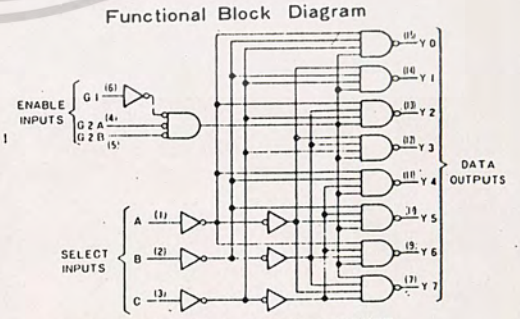
Function Table

'S138 'LS138

INPUTS ENABLE G1	G2*	SELECT C	B	A	OUTPUTS Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7
X	H	X	X	X	H	H	H	H	H	H	H	H
L	X	X	X	X	H	H	H	H	H	H	H	H
H	L	L	L	L	L	H	H	H	H	H	H	H
H	L	L	L	H	H	L	H	H	H	H	H	H
H	L	L	H	L	H	H	L	H	H	H	H	H
H	L	L	H	H	H	H	H	L	H	H	H	H
H	L	H	L	L	H	H	H	H	L	H	H	H
H	L	H	L	H	H	H	H	H	H	L	H	H
H	L	H	H	L	H	H	H	H	H	H	L	H
H	L	H	H	H	H	H	H	H	H	H	H	L

◆ G2 = G2A + G2B
H = high level, L = low level, X = irrelevant

Functional Block Diagram



'S138 'LS138 DECODER/DEMULTIPLEXER

เอกสารสำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

54245/74245 Octal Bus Transceivers with 3-state Outputs

	Schottky TTL				High-Speed TTL				Low-Power Schottky TTL					Standard TTL				Low-Power TTL			
	Device Type	Package			Device Type	Package			Device Type	Package				Device Type	Package			Device Type	Package		
		C	P	M CF		C	P	M CF		C	P	M	CF		C	P	M CF		C	P	M CF
									SN54LS245	J		W	W								
									SN74LS245	J	N										
I.																					
AIRCHILD																					
OTOROLA																					
S. C.																					
HILIPS																					
IGNETICS																					
EMENS																					
UJITSU																					
ITACHI																					
ITSUBISHI																					
EC																					
OSHIBA																					

Electrical Characteristics SN54LS245/SN74LS245

absolute maximum ratings over operating free-air temperature range

Supply voltage, VCC	7 V	Operating free-air temperature range	SN54LS	55°C to 125°C
Input voltage	7 V		SN74LS	0°C to 70°C
		Storage temperature range		65°C to 150°C

recommended operating conditions

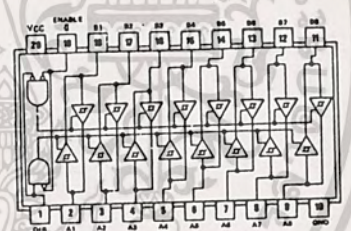
	SN54LS245			SN74LS245			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, VCC	4.5	5	5.5	4.75	5	5.25	V
High-level output current, IOH			12			15	mA
Low-level output current, IOL			12			24	mA
Operating free-air temperature, TA	55		125	0		70	°C

electrical characteristics over recommended operating free-air temperature range

PARAMETER			TEST CONDITIONS†		SN74LS245 MIN	TYP‡	MAX	UNIT
VIH	High-level input voltage				2			V
VIL	Low-level input voltage						0.8	V
VIK	Input clamp voltage		VCC = MIN, II = −18mA				−1.5	V
	Hysteresis (VT+ − VT−)	A or B input	VCC = MIN			0.2	0.4	V
VOH	High-level output voltage		VCC = MIN, VIH = 2V, VIL = VIL max	IOH = −3mA	2.4	3.4		V
				IOH = MAX	2			
VOL	Low-level output voltage		VCC = MIN, VIH = 2V, VIL = VIL max	IOL = 12mA			0.4	V
				IOL = 24mA			0.5	
IOZH	Off-state output current, high-level voltage applied		VCC = MAX, G at 2V	VO = 2.7 V			10	μA
IOZL	Off-state output current, low-level voltage applied			VO = 0.4V			−200	
II	Input current at maximum input voltage	A or B	VCC = MAX,	VI = 5.5V			0.1	mA
		DIR or G		VI = 7V			0.1	
IIH	High-level input current		VCC = MAX, VIH = 2.7V				20	μA
IIL	Low-level input current		VCC = MAX, VIL = 0.4V				−0.2	mA
IOS	Short-circuit output current		VCC = MAX		−40		−225	mA
ICC	Supply current	Total, outputs high	VCC = MAX, Outputs open			48	70	mA
		Total, outputs low				62	90	
		Outputs at Hi-Z				64	95	

switching characteristics, VCC 5V, TA 25°C

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
tPLH	Propagation delay time, low-to-high-level output	CL = 45pF, RL = 667Ω, See Note 2		8	12	ns
tPHL	Propagation delay time, high-to-low-level output			8	12	ns
tPZL	Output enable time to low level			27	40	ns
tPZL	Output enable time to high level			25	40	ns
tPLZ	Output disable time from low level	CL = 5pF, RL = 667Ω, See Note 2		15	25	ns
tPHZ	Output disable time from high level			15	25	ns



SN54LS245 (J) SN74LS245 (J, N)

†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡All typical values are at VCC = 5V, TA = 25°C.
*Not more than one output should be shorted at a time, and duration of the short-short should not not exceed one second.

ไม่วากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงการ PAL PROGRAMMABLE CARD นี้ ประสพปัญหาในการปฏิบัติงานพอสมควร ทั้งในด้านทฤษฎีและด้านปฏิบัติ แต่ก็ได้แก้ไขจนประสพความสำเร็จด้วยดี เนื่องจากได้รับการอนุเคราะห์จากท่านอาจารย์ ดร.ไพศาล นาคพิพัฒน์ กับท่านอาจารย์ อุทัย ศรีธีระวิโรจน์ ซึ่งท่านได้ให้คำแนะนำทางด้านวิชาการ และการปฏิบัติงาน ตลอดจนการวางแผนงาน จนกระทั่งงานที่ออกมาประสพผลสำเร็จ

กระผมจึงขอขอบพระคุณท่านอาจารย์ ดร.ไพศาล นาคพิพัฒน์ กับท่านอาจารย์ อุทัย ศรีธีระวิโรจน์ ไว้ ณ.ที่นี้ และถ้าท่านผู้อ่านพบข้อผิดพลาดประการใดในโครงการนี้ กระผมต้องขออภัย และขอความกรุณาแนะนำข้อผิดพลาดที่ท่านพบเพื่อแก้ไขให้สมบูรณ์ จักเป็นพระคุณอย่างยิ่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. IBM PC/XT Technical Reference
2. Programmable Array Logic Data Book , National Semiconductor Corporation.
3. Programmable Array Logic Data Book ,Taxas Instrument
4. Turbo Assembly User's Guide, Borland International.
5. Turbo Debugger User's Guide, Borland International.
6. The 8086 Book, Russell Rector - George Alexy
7. BYTE Magazine, January 1987, Robert A. Freedman
8. Interfacing Techniques, Ronald L. Krutz, Ph.D.
9. Peter Norton's Assembly Language Book for the IBM PC Peter Norton and John Socha
10. Advanced MS-DOS, Ray Duncan , Microsoft Corporation
11. การอินเทอร์เฟส IBM PC ของ ธานินทร์ ถาวรศาสนวงศ์ และ ทินกร
12. หลักการเขียนภาษาแอสเซมบลี้ 8088 ของ ดร.ศิริวรรณ ฉันทาดิศัย
13. คู่มือชิพซับพอร์ทและหน่วยความจำ ของ บ.ซีเอ็ดยูเคชั่น
14. คู่มือ/เทียบเบอร์ ไอซี TTL ของ บ.ซีเอ็ดยูเคชั่น
15. คู่มือดูขาไอซี ของ บ.ซีเอ็ดยูเคชั่น
16. PAL Logic Gate สารพัดนึก เซมิคอนดัคเตอร์อิเลคทรอนิคส์ ฉบับที่ 83

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้