ปีการศึกษา 2531

เครื่องโปรแกรมไบโพลาร์ พรอม

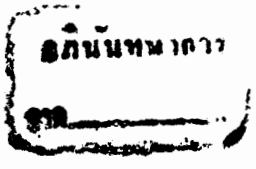
(BIPOLAR PROM PROGRAMMER CARD)

นายชาลี โพธิ์เงินนาค 28.6705

นายมาโนช คชาใย 28.6715

อาจารย์ที่ปรึกษา

รศ. กิตติ ตรีเศรษฐ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

023102 -8.ส.ค. 2532

ปริญญานิพนธ์ปีการศึกษา 2531

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เครื่องโปรแกรมไบโพลาร์ พรอม

ผู้จัดทำ

1. นายชาลี โพธิ์เงินนาค 28.6705
2. นายมาโนช คชาใย 28.6715

..... อาจารย์ที่ปรึกษา

(รศ. กิตติ ตรีเศรษฐ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

023102

เครื่องโปรแกรมไบโพลาร์ พรอม

ชาลี โพธิ์เงินนาค
มาโนช คชาใย

รศ. กิตติ ตรีเศรษฐ อาจารย์ที่ปรึกษา
ปีการศึกษา 2531

บทคัดย่อ

โครงงานนี้ เป็นโครงงานทางด้านอิเล็กทรอนิกส์และคอมพิวเตอร์ ที่เป็นการประยุกต์ใช้งานในลักษณะการอินเตอร์เฟส เครื่องคอมพิวเตอร์ ไอบีเอ็ม พีซี เพื่อใช้เป็นเครื่องโปรแกรม หน่วยความจำชนิดหนึ่ง ที่ใช้กันมากในวงการธุรกิจอุตสาหกรรม ซึ่งมีการนำเอาอุปกรณ์ประเภทหน่วยความจำชนิดนี้ ไปใช้ร่วมกับงานต่างๆ อย่างมาก หน่วยความจำชนิดนี้ก็คือ ไบโพลาร์ พรอม (Bipolar Prom) ซึ่งเป็นหน่วยความจำชนิดที่มีราคาถูก และคุณภาพดีเหมาะสำหรับที่จะนำไปใช้งาน เพราะผู้ใช้สามารถที่จะทำการโปรแกรมข้อมูลเองได้ ซึ่งโครงสร้างภายในจะเป็นฟิวส์ (Fuse) ซึ่งทำมาจากโลหะผสมระหว่างนิเกิลกับโครเมียม (NiCr) วิธีการโปรแกรม สามารถทำการโปรแกรมข้อมูลเข้าไปในตัว ไบโพลาร์ พรอม ได้ครั้งละหนึ่งบิท โดยการทำให้ระดับแรงดันที่ขาข้อมูล (Data) เปลี่ยนจาก 5 โวลท์ เป็น 17.5 โวลท์ และเปลี่ยนระดับแรงดันขา Vcc จาก 5 โวลท์เป็น Vccp ที่ 8.75 โวลท์ ป้อนพัลซ์ (pulse) เข้าที่ขา $\overline{CE}$ ด้วยระดับสัญญาณ ลอจิก "0" ในช่วง 10-25 u sec. ก็จะสามารถทำให้ฟิวส์ในตำแหน่งบิทที่ต้องการโปรแกรมหลอมละลายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูปภาพ

		หน้า
รูปที่ 2.1	โครงสร้างภายในของแรม (RAM)	5
รูปที่ 2.2	แสดงการใช้งานขาข้อมูลเข้าออกร่วมกัน	6
รูปที่ 2.3	โครงสร้างภายในของรอม (ROM)	8
รูปที่ 2.4	วงจรภายในของภาคเข้ารหัสของรอม (ROM)	10
รูปที่ 2.5	วงจรภายในบางส่วนของพรอม (PROM) เบอร์ N82S23/123 ของบริษัท ซิกเนติกส์	11
รูปที่ 2.6	โครงสร้างของมอสเฟท (MOS FET)	12
รูปที่ 2.7	บล็อกไดอะแกรมของซีสเต็มบอร์ด (SYSTEM BOARD)	19
รูปที่ 2.8	แสดงสัญญาณต่างๆที่ขาสล็อต (SLOT) ของ ไอบีเอ็ม พีซี	23
รูปที่ 2.9	การใช้แอดเดรสบิทต่างๆในการอ้างแอดเดรสของพอร์ท ในไอบีเอ็ม พีซี	36
รูปที่ 2.10	การใช้งานแอดเดรสของพอร์ทบน ไอบีเอ็ม พีซี	36
รูปที่ 2.11	การใช้งานแอดเดรสต่างๆสำหรับพอร์ท ไอโอ (I/O) ของไอบีเอ็ม พีซี	37
รูปที่ 2.12	การใช้แอดเดรสสำหรับพอร์ทไอโอ บนการ์ดต่างๆ	39
รูปที่ 2.13	ตัวอย่างวงจรดีโค้ดแอดเดรส (DECODE ADDRESS) แบบฟิกซ์ (FIXED)	40
รูปที่ 2.14	ตัวอย่างวงจรดีโค้ดโดยใช้สวิทช์เลือก	43
รูปที่ 3.1	แสดงช่วงเวลาของแรมดันขณะทำการโปรแกรม	46
รูปที่ 3.2	บล็อกไดอะแกรมของเครื่องโปรแกรม พรอม	47
รูปที่ 3.3	วงจรบัฟเฟอร์ และไอโอ ดีโค้ดเดอร์	49
รูปที่ 3.4	วงจร 8255 PPI ไอโอพอร์ท	50

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.5	วงจรขับรีเลย์ และควบคุมการอ่าน/เขียนข้อมูล	52
รูปที่ 4.1	แสดงการทำงานของโปรแกรมหลัก	54
รูปที่ 4.2	แสดงการอ่านข้อมูล	55
รูปที่ 4.3	แสดงการเขียนหรือโปรแกรมข้อมูล	56
รูปที่ 4.4	แสดงการตรวจสอบข้อมูล	57
รูปที่ 4.5	แสดงการตรวจสอบสภาพ พรอม	58
รูปที่ 4.6	แสดงฟังชั่นการเปลี่ยนชนิดของ พรอม	59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

ในงานอิเล็กทรอนิกส์ปัจจุบัน ที่มีความยุ่งยากซับซ้อนมากๆ นั้นมักจะต้องมีการนำเอาอุปกรณ์ประเภทหน่วยความจำ(Memory)มาใช้งานร่วมด้วยเสมอ อาจใช้งานในลักษณะของตัวเข้ารหัส(Encoder) หรือตัวถอดรหัส(Decoder)ก็ตาม ดังนั้นความจำเป็นในการที่จะต้องนำอุปกรณ์หน่วยความจำมาใช้งาน จึงนับวันจะมี มากขึ้นเรื่อยๆ อย่างไรก็ดี อุปกรณ์หน่วยความจำนั้นมีอยู่หลายชนิดหลายประเภทและราคาก็สูงต่ำแตกต่างกันไปตามชนิด คุณภาพ และความยุ่งยากซับซ้อนในการผลิต สำหรับงานวิจัยนี้ จะทำการวิจัยหน่วยความจำชนิดไบโพลาร์พรอม(Bipolar PROM)ซึ่งเป็นหน่วยความจำที่มีราคาไม่สูงนัก และผู้ใช้สามารถโปรแกรมเพื่อใช้งานเองได้แต่เนื่องจากเครื่องมือในการโปรแกรมหน่วยความจำชนิดนี้มีราคาแพงมาก ทำให้หน่วยความจำชนิดนี้ไม่แพร่หลาย หรือเป็นที่นิยมเท่าที่ควร ดังนั้นงานวิจัยนี้จึงเสนองานที่เป็นลักษณะการออกแบบ (Design) เครื่องโปรแกรมพรอมขึ้น โดยมีจุดประสงค์ คือ เพื่อให้การใช้งานของหน่วยความจำชนิดนี้ เป็นที่นิยมแพร่หลายอย่างกว้างขวางยิ่งขึ้น และเพื่อให้มีเครื่องโปรแกรมพรอมราคาถูกที่สามารถผลิตขึ้นใช้เองได้ นอกจากนั้น งานวิจัยนี้ยังเป็นแนวทางในการศึกษาค้นคว้าเพื่อการพัฒนาไปสู่การออกแบบเครื่อง โปรแกรมตัวอื่นๆต่อไป

เครื่องโปรแกรมไบโพลาร์พรอมนี้ มีลักษณะเป็นการ์ด(Card)สำหรับทำหน้าที่โปรแกรมพรอมเบอร์ต่างๆ โดยใช้เครื่องคอมพิวเตอร์ ไอบีเอ็ม/พีซี (IBM/PC)เป็นหน่วยควบคุมการทำงาน ตามคำสั่งของซอฟแวร์ ซึ่งเขียนด้วยภาษาแอสแซมบลี้ 8088 ขีดความสามารถของการ์ดถูกออกแบบให้มีความสามารถดังนี้ คือ

1.1 แสดงรายการเมนู (Menu)ออกหน้าจอเพื่อบอกหน้าที่การทำงานของการ์ด ในรูปแบบต่างๆได้

1.2 สามารถอ่านข้อมูลจากพรอม นำไปเก็บไว้ในที่พักข้อมูล(Buffer)ได้

1.3 สามารถโปรแกรมหรือเขียน(Write) ข้อมูลลงในตำแหน่งต่างๆของตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พรอมได้ตามต้องการ

1.4 สามารถตรวจสอบ (Check) สภาพของพรอมได้ว่า เคยถูกใช้มาแล้วหรือยัง (ดี หรือ เสีย)

1.5 สามารถแก้ไขเปลี่ยนแปลงข้อมูลในที่พักข้อมูลได้ เพื่อทำการโปรแกรม

1.6 สามารถเก็บข้อมูลจากที่พักข้อมูลลงแผ่นดิสค์ (Disk) หรือนำข้อมูลจากแผ่นดิสค์เข้าไปเก็บยังที่พักข้อมูลได้

1.7 สามารถเลือกชนิดและเบอร์ของพรอม ที่ต้องการอ่านหรือเขียนได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

หลักการทั่วไปของหน่วยความจำและสัญญาณควบคุมของไมโครคอมพิวเตอร์

ในการออกแบบงานใดๆก็ตาม ที่ต้องใช้งานร่วมกับระบบคอมพิวเตอร์ จำเป็นอย่างยิ่งที่จะต้องศึกษา และทำความเข้าใจกับเรื่องราวต่างๆ ที่เกี่ยวข้องกับระบบให้ละเอียดลึกซึ้งในบทนี้จะกล่าวถึงหลักการทั่วไปในบางส่วนของระบบคอมพิวเตอร์ เช่น หน่วยความจำ การอินเตอร์เฟส (Interfaceing) กับสัญญาณต่างๆจากช่องสล๊อต (Slot) ของเครื่องคอมพิวเตอร์ ไอบีเอ็ม พีซี (IBM PC) เพื่อเป็นแนวทางในการศึกษาและทำความเข้าใจกับโครงงานนี้ได้ดียิ่งขึ้น

2.1 ระบบหน่วยความจำ (Memory Systems)

ตามปกติ ไมโครโปรเซสเซอร์เป็นหัวใจของระบบไมโครคอมพิวเตอร์ทั่วๆไป และหน่วยความจำเปรียบเสมือนสมองของระบบ ซึ่งมีความสำคัญไม่ยิ่งหย่อนไปกว่าตัวไมโครโปรเซสเซอร์เอง ที่จริงแล้วหน่วยความจำ ยังเป็นส่วนที่มีราคาแพงมากในระบบอีกด้วยดังนั้นจึงควรได้รับความสนใจอย่างจริงจัง ระบบความจำก็คือระบบบันทึกข้อมูลในปัจจุบันและรักษาไว้ให้อยู่ต่อไปในอนาคต หน่วยความจำเป็นที่เก็บรักษาข้อมูลที่จะนำมาใช้ได้ในภายหลัง ระบบความจำจะประกอบด้วยหน่วยความจำ (Memory elements) มาประกอบกันอยู่ในลักษณะที่เหมาะสม พร้อมกับอุปกรณ์อื่นๆที่จำเป็น วิธีการประกอบกัน ของหน่วยความจำ คือข้อมูลที่เคลื่อนย้ายเข้าและออก จากหน่วยความจำ แต่ละครั้ง จะเป็นจำนวนบิท (bit) ที่แน่นอนจำนวนหนึ่ง เราเรียกว่า คำ หรือเวิร์ด(Word) เราอาจจะมองระบบความจำว่าหมายถึง สถานที่หรือที่เก็บข้อมูลหลายๆตำแหน่งซึ่งเราสามารถอ้างอิงถึงได้โดยตำแหน่งของคำ (Word Position) หรือ (Word Location) หรือแอดเดรส (address) การนำข้อมูลจากภายนอกเข้าไปเก็บไว้ในระบบความจำที่ตำแหน่งใดตำแหน่งหนึ่งเรียกว่าการเขียน (memory write operation) ในทางตรงกันข้าม การนำข้อมูลที่ถูกเก็บไว้ในระบบความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ออกมาภายนอกเรียกว่าการอ่าน (Memory Read Operation) การเลือกตำแหน่งที่จะนำข้อมูลไปเก็บหรือตำแหน่งที่จะนำข้อมูลออกมาจากหน่วยความจำ ทำได้หลายวิธี วิธีการเลือกหรือเข้าไปหาข้อมูลตำแหน่งนั้นๆเราเรียกว่าแอคเซสส์ (Access) หรือการเข้าถึงข้อมูล และเราพอจะแบ่งลักษณะของการเข้าถึงข้อมูลนี้ได้ ๒ แบบ คือ เข้าแบบแรนด้อม (Random Access) และเข้าแบบเป็นลำดับหรือซีเควนเชียล (Sequential Access) ระบบความจำแบบแรนด้อม หมายถึงระบบความจำที่เราสามารถเลือกตำแหน่งของข้อมูลตำแหน่งต่างๆได้โดยใช้เวลาเท่ากัน นั่นคือเราจะใช้เวลาเท่ากันเสมอไม่ว่าจะเลือกอ่านหรือเขียนข้อมูลตำแหน่งไหนในระบบ ในทางตรงกันข้าม ระบบความจำแบบซีเควนเวียล หมายถึงระบบความจำที่เวลาในการเข้าถึงข้อมูลที่ตำแหน่งต่างๆ ไม่เท่ากัน ขึ้นอยู่กับตำแหน่งของข้อมูลนั้นๆ จะเห็นได้ว่า หน้าที่ที่สำคัญของหน่วยความจำ ในระบบไมโครคอมพิวเตอร์ก็คือ เป็นหน่วยเก็บโปรแกรมคำสั่งที่จะให้ ซีพียู (CPU) ทำงานตามความต้องการและเป็นหน่วยเก็บข้อมูลชั่วคราว ซึ่งหมายถึงข้อมูลสามารถเก็บเข้าไปในหน่วยความจำ หรือเอาออกจากหน่วยความจำได้ตามความต้องการในขณะใช้งานหรือไม่ใช้งานก็ได้

จากหน้าที่ทั้งสองของหน่วยความจำ จะเห็นว่า ในบางครั้งสิ่งที่ถูกเก็บไว้อาจเป็นในลักษณะถาวร นั่นคือไม่ต้องเปลี่ยนแปลงอีกต่อไป หลักจากเก็บไว้แล้ว เช่นเป็นโปรแกรมสั่งงานของซีพียูในงานประจำ แต่ในบางครั้งอาจมีการ เก็บข้อมูลในลักษณะชั่วคราว เช่น เป็นตัวเลขจากการคำนวณและพร้อมจะนำไปใช้งานต่อไปจากความจำเป็นนี้ทำให้ต้องมีหน่วยความจำมากกว่า 1 ประเภท เช่นอาจเป็นหน่วยความจำถาวรและหน่วยความจำชั่วคราว ที่สามารถเก็บและเรียกออกมาใช้งานได้ทุกเมื่อ สำหรับในที่นี้จะแบ่งออกเป็น 2 ตระกูล คือตระกูลรอม (ROM หรือ Read-Only Memory) และตระกูลแรม (RAM หรือ Random Access Memory) ที่บางคนรู้จักในชื่อ อาร์ดับบลิวเอ็ม (RWM หรือ Read/Write Memory) ส่วนตระกูลซีเควนเชียล เช่นพวกชิฟท์รีจิสเตอร์ เป็นต้น จะไม่ขอกล่าวถึงในที่นี้

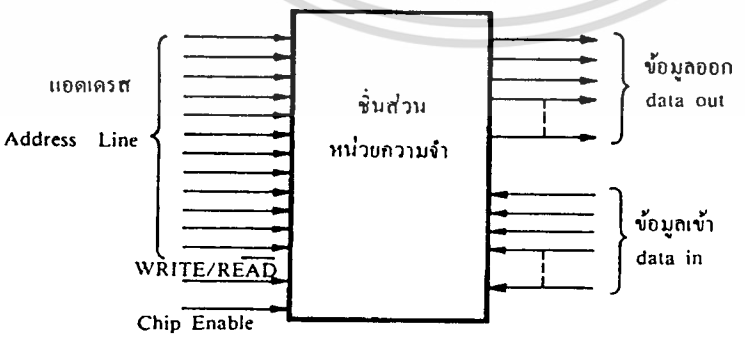
คุณสมบัติที่น่าสนใจ 2 ประการของหน่วยความจำ ได้แก่ การอ่านข้อมูลออกมาแล้วหน่วยความจำยังจำข้อมูลนั้นได้หรือไม่ (Destructiveness) และการที่เมื่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แหล่งจ่ายไฟดับลงข้อมูลในหน่วยความจำจะสูญหายหรือไม่ (Volatility) ซึ่งขึ้นอยู่กับชนิดและตระกูลของหน่วยความจำนั้น เช่น รอม มักจะเป็นพวก นอนโวลาไทล์ (Nonvolatile) ส่วน แรมมักเป็นพวก โวลาไทล์ (Volatile) จึงต้องได้รับการเอาใจใส่เป็นพิเศษ โดยเฉพาะในแง่ของแหล่งจ่ายไฟเลี้ยงวงจร

2.1.1 หน่วยความจำตระกูลแรม

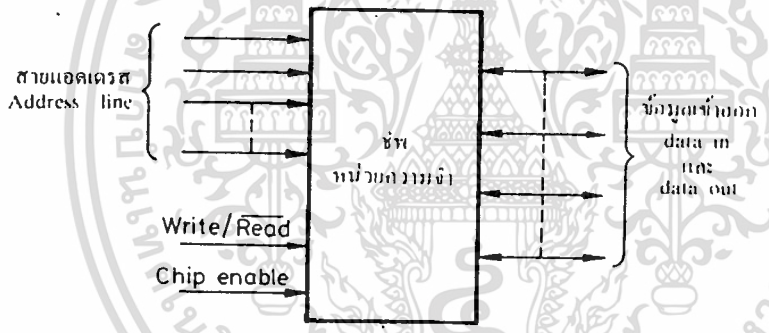
ดังได้กล่าวมาแล้วว่าระบบเข้าถึงแบบแรนด้อม ตำแหน่งที่เก็บข้อมูลแต่ละคำเราสามารถอ้างถึงได้โดยตรงทุกตำแหน่ง ทำให้เวลาในการเข้าถึงข้อมูลนั้นเท่ากันเสมอและแต่ละตำแหน่งจะมีแอดเดรสซึ่งเป็นตัวเลขไบนารี่กำกับอยู่ ระบบความจำจะประกอบด้วย หน่วยความจำย่อย ๆ ซึ่งแต่ละหน่วยก็ต้องมีคุณสมบัติในการเข้าถึงข้อมูลแบบ แรนด้อมเช่นเดียวกันด้วย ในระบบอิเล็กทรอนิกส์หน่วยความจำย่อยๆ เป็นวงจร อินทริเกรดอยู่ในชิพแผ่นเดียว ความจุของระบบความจำทั้งหมดก็คือความจุของแต่ละหน่วยความจำย่อยๆรวมกันนั่นเอง ชิพหน่วยความจำต้องสามารถนำมาต่อร่วมกันเพื่อประกอบเป็นระบบใหญ่ โดยการเชื่อมโยงสายควบคุม และข้อมูลเข้าด้วยกัน ในลักษณะของบัส โดยทั่วๆไป ชิพแต่ละตัวจะมีลักษณะโครงสร้างและสัญญาณต่างๆที่เกี่ยวข้องดังต่อไปนี้



รูปที่ 2.1 แสดงโครงสร้างของแรม (RAM)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามรูป 2.1 จะประกอบด้วยตัวชิพและขาต่อสัญญาณภายนอกคือ ขาข้อมูลออก (Data OUT) เป็นข้อมูลออกจากตัวชิพ เมื่อทำการอ่าน ขาข้อมูลเข้า (Data IN) เป็นข้อมูลที่จะถูกนำเข้าไปเก็บไว้ในชิพเมื่อทำการเขียน สายแอดเดรส (Addr. Line) เป็นสัญญาณเลือกตำแหน่งที่จะเขียนหรืออ่าน ขาเขียน/อ่าน (Write/Read) คือสัญญาณควบคุมแสดงขบวนการที่จะทำคือ เขียนข้อมูลหรืออ่านข้อมูล ขาชิพอีนาเบิล (Chip Enable) นอกจากจะต้องเลือกตำแหน่งด้วย สายแอดเดรสและเลือกขบวนการ (อ่านหรือเขียน) แล้วชิพอีนาเบิลเป็นสัญญาณบังคับ ให้หน่วยความจำอันนี้ต้องรับสัญญาณหรือรับรู้สัญญาณควบคุมตัวอื่นๆทั้งหมด ในบางครั้งเส้นทางนำข้อมูลเข้าหรือออกจากชิพยังเป็นเส้นทางเดียวกัน จะทำให้ลักษณะของชิพเป็นดังนี้



รูปที่ 2.2 แสดงการใช้ขาข้อมูลเข้าออกร่วมกัน

ลักษณะข้อมูลเข้าและข้อมูลออกใช้เส้นทางร่วมกัน เมื่อนำไปต่อร่วมกันกับชิพตัวอื่นและต่อเข้ากับระบบบัสในลักษณะทางเดินของบัสสองทิศทาง (Bidirectional Bus) หน้าที่ของไอซีหน่วยความจำ คือ เก็บข้อมูลไว้ หรือให้ข้อมูลที่เก็บไว้เมื่อทำขบวนการอ่าน หน่วยความจำแบบนี้อาจเรียกว่าแบบการเขียนและอ่านได้ในความเป็นจริงแล้วยังมีหน่วยความจำแบบเขียนข้อมูลเข้าไปได้ครั้งเดียวและข้อมูลจะ อยู่ตลอดไปเมื่อใช้งานจริงเราจะใช้อ่านข้อมูลออกมาอย่างเดียว เราเรียกหน่วยความจำแบบนี้ว่า รอมในไอซีที่เราใช้งานจริงๆ เราอาจแยกไอซีที่ใช้เป็นแรมออกได้ตามเทคนิคของวงจรเก็บความจำได้เป็น ๒ พวก คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1.1 หน่วยความจำแบบสแตติค (Static Memory) เป็นหน่วยความจำที่สามารถเก็บข้อมูลได้ตลอดไปตราบเท่าที่ยังมีไฟจ่ายเลี้ยงวงจรอยู่ หน่วยความจำสแตติค ยังแบ่งออกเป็นชนิดต่างๆเช่นทีทีแอล (Transistor Transistor Logic Static Memory) เป็นไอซีตระกูลหนึ่งซึ่งประกอบด้วยทรานซิสเตอร์และตัวต้านทานต่ออยู่ภายใน เนื่องจากทรานซิสเตอร์เป็นสารกึ่งตัวนำ ที่การทำงานภายในตัวอาศัยประจุทั้งบวกและลบเราจึงเรียกว่าเป็นแบบสองขั้ว หรือไบโพลาร์นอกจากนี้ยังมีหน่วยความจำสแตติกแบบมอส (Metal Oxide Semicondutor หรือ MOS) ซึ่งมีรูปร่างของวงจรกับความจำเหมือนกับพวกไบโพลาร์นั่นเอง แต่แทนที่จะใช้ทรานซิสเตอร์ชนิดไปโพลาร์ต่อเป็นฟลิบฟลอบ ก็ใช้ มอสเฟท (MOSFET) แทนไม่ว่าจะเป็นแบบชนิดเอ็น (N-Type) หรือชนิดพี (P-Type) ก็ตามที่จริงแล้ววงจรหน่วยความจำสแตติกก็คือวงจรฟลิปฟลอปนั่นเอง ซึ่งสามารถคงสถานะใดสถานะหนึ่งในสองสถานะได้ตลอดไปถ้ามีไฟเลี้ยงวงจรอยู่

2.1.1.2 หน่วยความจำแบบไดนามิค (Dynamic Memory) เป็นหน่วยความจำที่สามารถเก็บข้อมูลของตัวเองได้ในระยะสั้น ข้อมูลจะสูญหายไปถึงแม้จะมีไฟเลี้ยงวงจรอยู่ก็ตาม จึงต้องมีขบวนการรีเฟรช เพื่อซ้ำข้อมูลของตัวเองอยู่ทุกๆ ระยะไม่ให้สูญหายไป ถึงแม้ว่าหน่วยความจำแบบไดนามิคจะต้องอาศัยขบวนการรีเฟรชซึ่งยุ่งยาก แต่ก็ยังคุ้มค่า เพราะลักษณะวงจรทำให้สามารถสร้างหน่วยความจำที่มีความจุสูงๆ ได้

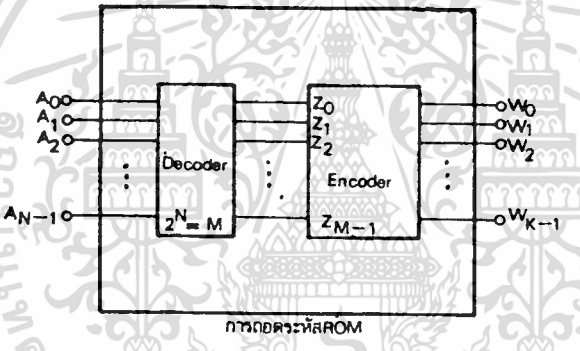
2.1.2 หน่วยความจำตระกูลรอม

รอม เป็นหน่วยความจำที่ใช้เก็บข้อมูลแบบถาวร หรือกึ่งถาวร นั่นคือข้อมูลต้องถูกเขียนไว้ในหน่วยความจำตั้งแต่ต้น หลังจากนั้นก็เป็นการอ่านข้อมูลออกมาเท่านั้น (Read Only) การเขียนข้อมูลทำได้ 2 วิธี คือ การเขียนข้อมูลมาจากโรงงานผู้ผลิต ตามความต้องการของลูกค้า เช่น เป็นชนิด มาสค์โปรแกรม (Mask Programed) ส่วนอีกแบบหนึ่ง คือผู้ใช้สามารถโปรแกรมเองได้ เรียกว่าเป็นชนิดพรอม (PROM หรือ Programmable Read-Only Memory) และก็มีบางประเภท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ผู้ใช้สามารถเปลี่ยนข้อมูลในหน่วยความจำได้ แต่ในการเขียนข้อมูลแต่ละครั้งต้องใช้เวลามากกว่าเวลาที่ใช้ในการอ่านข้อมูล จากหน่วยความจำมาก ประเภทนี้เป็นชนิดรีโปรแกรมเมเบิล รอม (Reprogrammable ROM) ข้อดีอย่างหนึ่งของหน่วยความจำตระกูลนี้คือ ข้อมูลจะไม่สูญหาย ถึงแม้ไฟฟ้าจะดับก็ตาม เราเรียกลักษณะอย่างนี้ว่า นอนโวลาไทล์ ซึ่งต่างไปจากหน่ายความจำตระกูลแรมที่เป็น ชนิด โวลาไทล์

เป็นที่ทราบกันดีแล้วว่า รอมนั้น ทำหน้าที่เป็นหน่วยความจำที่เก็บข้อมูลที่ค่อนข้างถาวร เช่น เก็บโปรแกรมที่ใช้บ่อย หรือสม่ำเสมอ แต่เราอาจมองรอมว่าทำหน้าที่แปลงระหัสจากชนิดหนึ่งไปเป็นอีกชนิดหนึ่งก็ได้ ในกรณีนี้รอมทำงานเป็นวงจรแปลงระหัส



รูปที่ 2.3 โครงสร้างภายในของรอม

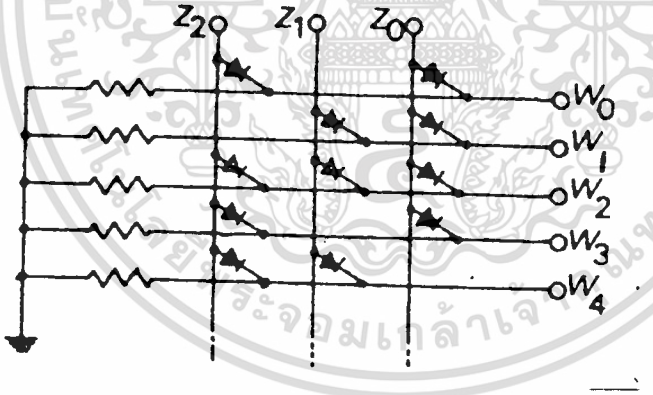
จากรูปที่ 2.3 แสดงให้เห็นโครงสร้างภายในของรอมทั่วไป Ao ถึง An-1 เป็นแอดเดรสของข้อมูลที่ต้องการอ่าน ซึ่งหลักจากผ่านวงจรถอดระหัส จะมีเพียงสายจากจำนวนทั้งหมด 2^n = M สายที่ถูกกระตุ้นและหลังจากผ่านวงจรเข้าระหัสแล้วข้อมูลจะมาปรากฏที่ขั้วออก Wo ถึง Wk-1 จะเห็นว่าสัญญาณเข้าที่มี N บิทและทำหน้าที่เสมือนเป็นแอดเดรส จะถูกแปลงเป็นสัญญาณออก K บิท ซึ่งก็คือ ข้อมูลที่เราต้องการโดยไม่สนใจว่า N จะมากกว่าหรือเท่ากับ หรือน้อยกว่า k ในที่นี้จะบอกได้ว่าหน่วยความจำตระกูลรอมนี้มีขนาดเป็น M x K บิทโดยที่ M คือจำนวนเวิร์ดทั้งหมดในหน่วยความจำ และ K คือจำนวนบิทของแต่ละเวิร์ด รอมมีอยู่หลายชนิดด้วยกันคือ

2.1.2.2 มาสค์โปรแกรมรอม (Mask-Programmed ROM)

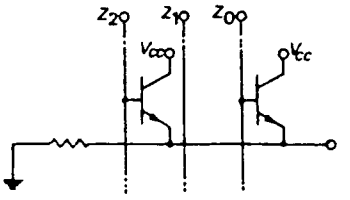
เป็นหน่วยความจำที่โปรแกรมไว้อย่างถาวรในระหว่างขบวนการผลิตโดยการทำ มาสค์ บางอันเฉพาะสำหรับงานนั้นๆ รอมประเภทนี้สามารถทำได้จากสิ่งประดิษฐ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำพวกไดโอด ทรานซิสเตอร์ (BJT) หรือมอสเฟท (MOSFET) ก็ได้ดังในรูปที่ 2.4 ปัญหาใหญ่ของรอมที่ทำจากไดโอด (รูปที่ 2.4 ก.) ไดโอดเป็นสิ่งประดิษฐ์ชนิดพาสซีฟ ดังนั้นมันจึงไม่สามารถจ่ายโหลดได้มากนัก เพราะขั้วออกต่อกับขั้วเข้าผ่านไดโอดโดยไม่มีการขยายกระแสเลย ปัญหานี้สามารถแก้ได้ โดยใช้พวกสิ่งประดิษฐ์ชนิดแอคทีฟ (รูปที่ 2.4 ข. และ ค.) ในทางปฏิบัติ ไบโพลาร์จังชั่นทรานซิสเตอร์ มักต้องการเนื้อที่ในชิ้นส่วนสารกึ่งตัวนำมากกว่า มอสเฟท อีกทั้งมีการ สูญเสียกำลังค่อนข้างสูง รอมที่ทำจากไบโพลาร์จังชั่นทรานซิสเตอร์จึงมีจำนวนบิทน้อยกว่ารอมที่ทำจากมอสเฟท บนชิ้นสารกึ่งตัวนำที่มีพื้นที่เท่ากัน แต่ข้อเสียของมอสเฟทก็มีได้แก่ความเร็วในการตอบสนอง (Access Time) ซึ่งหมายถึงช่วงเวลาที่ต้องรอหลังจากให้แอดเดรสแก่หน่วยความจำจนกระทั่งได้ข้อมูลจากแอดเดรสนั้นๆ มาปรากฏที่ขั้วออกของหน่วยความจำ

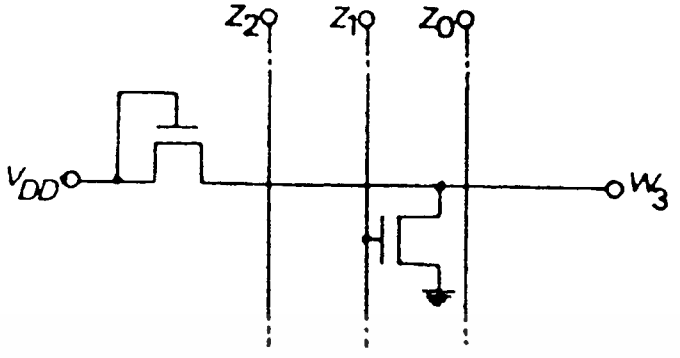


ก) ไดโอด



ข) ทรานซิสเตอร์ (BJT)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
023102



ค) MOSFET

รูปที่ 2.4 วงจรภายในของภาควงจรเข้ารหัสของรอม

จากรูปที่ 2.4 ซึ่งแสดงเฉพาะภาควงจรเข้ารหัสของรอมว่าสายแอดเดรสและสายสัญญาณออกทั้งหมด จะประกอบกันเป็น ตารางกริด ที่จุดตัดจะมีสิ่งประดิษฐ์ซึ่งอาจเป็นไดโอดหรือบีเจที หรือมอสเฟทก็ได้ ต่อเชื่อมโยงระหว่างสายแอดเดรสกับสายสัญญาณออกหรือไม่นั้น ขึ้นอยู่กับว่าที่บิทนั้นของแอดเดรสนั้นๆ ต้องการให้เป็น 1 หรือ 0

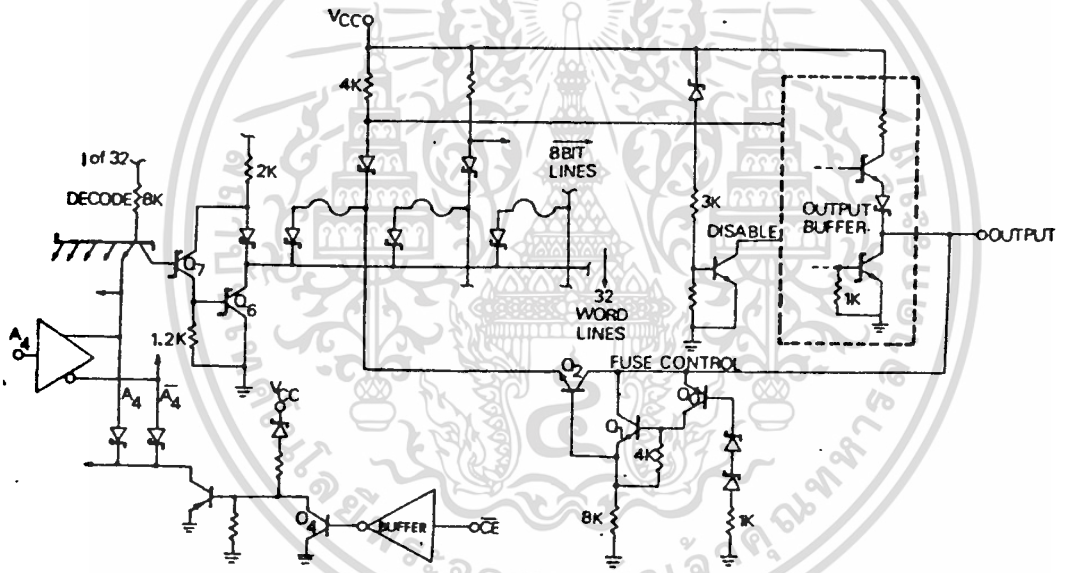
คุณสมบัติของรอมชนิดนี้มีดังต่อไปนี้คือ เก็บข้อมูลได้ตลอดกาล แม้จะมีไฟเลี้ยงวงจรหรือไม่ก็ตาม ผู้ผลิตเป็นผู้โปรแกรมข้อมูลในหน่วยความจำ และเราไม่สามารถเปลี่ยนแปลงข้อมูลได้ข้อดีข้อเสียก็คล้ายคลึงกับของรอมชนิดอื่นแต่ต่างกันบ้างตรงขนาดของหน่วยความจำ (จำนวนเวิร์ด) คูณกับจำนวนบิทต่อ 1 เวิร์ดอาจเล็กหรือใหญ่ตามต้องการได้ และจำนวนบิทต่อ 1 หน่วยพื้นที่ค่อนข้างสูงแต่กระนั้นก็ดีราคาต่อ 1 หน่วยค่อนข้างสูงด้วยเนื่องมาจากค่าใช้จ่ายในการทำ มาสค์ ดังนั้นรอมชนิดนี้จึงมีประโยชน์เฉพาะกรณีที่มีความต้องการสูง มิฉะนั้นจะไม่คุ้มกับต้นทุนในการผลิต

2.1.2.2 โปรแกรมเมเบิลรอม (Programmable ROM หรือ PROM)

ที่จริงแล้วพรอม (PROM) นั้น มีลักษณะวงจรภายในคล้ายคลึงกับพวกมาสค์-โปรแกรม รอม และตามปกติมักจะทำเป็นตัวไอซีที่มีขาอย่างเดียวกับมาสค์-โปรแกรม รอม ที่เป็นคู่ของมันด้วย (pin-for-pin Compatible) ในมาสค์-โปรแกรม รอม ทางผู้ผลิตได้ต่อหรือไม่ต่อเชื่อมโยงสายแอดเดรสกับสายสัญญาณออก ด้วยสิ่งประดิษฐ์สารกึ่งตัวนำที่จุดตัดในตารางกริด ตามที่เราต้องการให้เป็น "1" หรือ "0"ไว้เรียบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ร้อย ส่วนในพรอม ที่จุดตัดทุกจุดในตารางกริด จะมีสิ่งประดิษฐ์สารกึ่งตัวนำต่ออยู่ แต่การโปรแกรมให้ข้อมูลของแต่ละบิทเป็น "0" หรือ "1" นั้น ใช้วิธีปล่อยกระแสสูงๆ ไหลผ่านฟิวส์ที่มีต่อร่วมกับสิ่งประดิษฐ์เพื่อให้ฟิวส์หลอมขาดตัดการต่อเชื่อมโยงที่บิทนั้นๆ หลังจากโปรแกรมเสร็จ ก็จะเหลือเฉพาะบิทที่ต้องการ ให้มีการต่อเชื่อมโยงด้วยสิ่งประดิษฐ์ ลักษณะอย่างนี้ทำให้มีอีกชื่อหนึ่งว่า ฟิลด์-โปรมแกรมเมเบิล รอม (Field-Programmable ROM) ดังรูปที่ 2.5 แสดงวงจรภายในบางส่วนของพรอม เบอร์ # N82S23/123 ของบริษัทซิกเนติคส์



รูปที่ 2.5 วงจรภายในบางส่วนของพรอม เบอร์ N82S23/123 ของบริษัทซิกเนติคส์

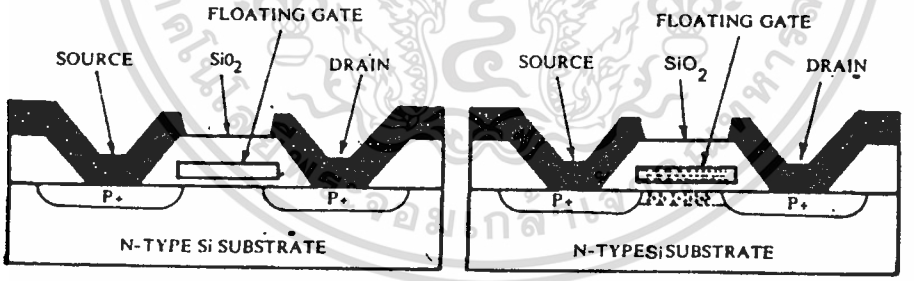
คุณสมบัติของพรอมก็คือ เป็นชนิด นอนโวลาไทล์ ซึ่งโปรแกรมได้แต่ไม่สามารถเปลี่ยนแปลงข้อมูลหลังจากที่โปรแกรมไปแล้วได้เพราะฟิวส์ที่เคยต่ออยู่ได้ขาดไปแล้วและไม่มีทางต่อกลับถึงกันอย่างเดิมได้อีก ขนาดความจุของพรอมไม่ค่อยสูงนัก อันเนื่องมาจากการมีฟิวส์ประกอบร่วมอยู่ด้วย และสาเหตุข้อนี้ด้วย ที่ทำให้จำนวนบิท ต่อ 1 หน่วยพื้นที่ต่ำกว่าของ มาสค์-โปรแกรม รอม ราคาต่อหน่วยไม่สูงเหมือน มาสค์-โปรแกรม รอม เพราะปริมาณที่ผลิตมีมากกว่า แต่สิ่งหนึ่งที่ผู้ใช้จำเป็นต้องมีก็คือเครื่องโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูล (Programmer) ซึ่งก็มีราคาไม่แพงนักด้วย จึงเป็นที่นิยมของผู้ที่กำลังพัฒนาโปรแกรมหรือในกรณีที่ต้องการผลิตอุปกรณ์จำนวนน้อยๆอย่างไรก็ดีนับว่าพรอมสามารถเข้ามาทดแทน มาสค์-โปรแกรม รอม มาก อีกทั้งราคาไม่แพง และโปรแกรมได้ง่าย

2.1.2.3 รีโปรแกรมเมเบิล รอม (Reprogrammable) ROM)

ลักษณะโครงสร้างภายใน และแนวความคิดของรอมชนิดนี้แตกต่างจาก พรอม มาก ตามปกติรีโปรแกรมเมเบิล รอม ทำจากมอสเฟท โดยที่สภาวะ "0" หรือ "1" แทนการนำกระแสหรือไม่นำกระแสของ มอสเฟทซึ่งการนำกระแสหรือไม่ของมอสเฟทนั้น ถูกควบคุมด้วยแรงดันที่ขาเกทเมื่อพิจารณาโครงสร้างของมอสเฟทแล้วเราจะเห็นว่าขาเกทถูกแยกจากส่วนที่นำกระแสอย่างเด็ดขาดดังรูปที่ 2.6 ดังนั้นแรงดันที่ขาเกท (ลักษณะโครงสร้างของขาเกทกับส่วนนำกระแสของมอสเฟทนั้น เปรียบเสมือนตัวเก็บประจุที่มีชั้น ออกไซด์เป็นตัวกลาง ไดอิเลคตริค) นี่เองเป็นแนวความคิดในการพัฒนารีโปรแกรมเมเบิล รอม ขึ้นมาโดยวิธีการดักประจุไว้ที่ขาเกท และประจุนั้นสามารถค้างอยู่ที่ขาเกทได้เป็นเวลานาน



รูปที่ 2.6 โครงสร้างของ มอสเฟท

แนวความคิดดังกล่าวยังแยกออกได้เป็น 2 วิธี ทั้งนี้ขึ้นอยู่กับผลของการดักประจุในขาเกท วิธีแรกคือการดักประจุไว้ที่ขาเกท ซึ่งถูกแยกจากส่วนนำกระแสของมอสเฟท ด้วยชั้นของซิลิกอนไดออกไซด์ ทำให้มอสเฟทนำกระแสได้ วิธีการดักประจุสามารถทำได้โดยการป้อนแรงดันไฟสูงเข้าที่มอสเฟท ทำให้พาหะประจุทะลุผ่านชั้นออกไซด์เข้าไปค้างอยู่ที่ขาเกทได้ ส่วนวิธีการเปลี่ยนแปลงข้อมูลที่ถูกโปรแกรมไปแล้วทำได้วิธีเดียวคือการลบ โดยฉายด้วยรังสีอุลตร้าไวโอเล็ตเพื่อให้พาหะประจุที่ถูกดักอยู่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีพลังงานพอเพียงที่จะหลุดรอดออกมาได้ แต่วิธีการลบข้อมูลทั้งหมดของ รอม เราไม่สามารถเลือกลบข้อมูลเฉพาะบางบิทได้ หน้าต่างที่ปล่อยให้รังสีเข้าไป ต้องเป็นพวกควอทซ์หรือวัสดุที่ยอมให้รังสีผ่านได้ ทำให้มีราคาสูงขึ้นเราเรียกรอมชนิดนี้ว่า อีพรอม (EPROM หรือ Erasable Programmable ROM)

วิธีที่สอง ใช้สิ่งประดิษฐ์สารกึ่งตัวนำจำพวก MNOS (Metal Nitride Oxide Semiconductor) FET การมีชั้นไนไตรด์ อีกชั้นหนึ่งเป็นการเพิ่มฉนวนอีกชั้นหนึ่งต่างหากจากชั้นออกไซด์ การดักประจุไว้ที่ผิวสัมผัส ระหว่างฉนวนทั้งสองจะเปลี่ยนแปลงแรงดันวิกฤตที่ขาเกทเทียบกับซอท (Threshold Voltage)ของเอฟอีที (เป็นแรงดันซึ่งพอดีทำให้เอฟอีที นำกระแสได้) การอ่านข้อมูลออกมาสามารถทำได้โดยป้อนแรงดันไฟสูงพอที่จะทำให้ เอฟอีที ที่มีการดักประจุอยู่นำกระแสได้แต่ต่ำพอที่จะไม่ทำให้ เอฟอีที ที่ไม่มีการดักประจุนำกระแสการดักประจุสามารถทำได้ โดยการป้อนแรงดันไฟสูงเข้าที่ มอสเฟท ทำให้ประจุทะลุผ่านออกไซด์ ไปค้างอยู่ที่ผิวสัมผัสระหว่างชั้นออกไซด์กับชั้นไนไตรด์ ซึ่งเป็นขบวนการคล้ายๆ กับของอีพรอม แต่วิธีการลบข้อมูลแตกต่างกันออกไป การลบข้อมูลของรอมชนิดหลังนี้ทำได้โดยป้อนแรงดันไฟสูงกลับทางกับขณะโปรแกรม ก็จะทำให้พาหะประจุที่ถูกดักสามารถทะลุ ผ่านชั้นออกไซด์กลับมาได้ แทนที่จะต้องใช้รังสีอุลตราไวโอเลตแบบเดียวกับของอีพรอม ทำให้เราสามารถลบข้อมูลที่บิทใดๆที่เราต้องการลบได้เราเรียกรอมชนิดนี้ว่าอีเอรอม (EAROM หรือ Electrically Alterable ROM) ปัญหาที่อีเอรอมยังไม่แพร่หลายอยู่ที่ขบวนการผลิต ซึ่งยังได้ผลไม่เป็นที่น่าพอใจและพื้นที่ต่อจำนวนบิทเท่าๆกันของอีเอรอม มากกว่าอีพรอม

ข้อดีของรีโปรแกรมเมเบิล รอม อยู่ที่ความสามารถในการเปลี่ยนแปลงข้อมูลในหน่วยความจำได้ ซึ่งสะดวกมากสำหรับการพัฒนาทางซอฟแวร์ แต่ในการให้ได้ประโยชน์เช่นนี้มา ก็ต้องลงทุนในด้านเครื่องโปรแกรม รอม ซึ่งมีราคาค่อนข้างสูง

2.2 SYSTEM UNIT

เนื่องจากอุปกรณ์ที่เป็นหัวใจของส่วนประกอบทางอิเลคทรอนิคทุกชิ้นที่ประกอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กันขึ้นเป็นเครื่อง IBM PC คือ System Board ดังนั้นควรจะทำความเข้าใจลักษณะการทำงานและองค์ประกอบของซีสเต็มบอร์ด (System Board)

2.2.1 ลักษณะการทำงานของ System Board

เนื้อหาในที่นี้จะกล่าวถึงลักษณะการทำงานและความสามารถต่างๆ ที่มีอยู่ใน System Board อย่างย่อๆ เมื่อเราเข้าใจลักษณะการทำงานและความสามารถของ System Board การทำการอินเทอร์เฟสอุปกรณ์ต่างๆ เข้ากับเครื่อง IBM PC ก็จะทำได้อย่างถูกต้องและสะดวกขึ้น

2.2.2 ไมโครโปรเซสเซอร์ 8088

ถ้าเราสมมุติให้ System Board เป็นหัวใจของอุปกรณ์อิเลคทรอนิคส์ต่างๆ ในเครื่อง PC ไมโครโปรเซสเซอร์ 8088 ก็จะเปรียบเสมือนการเต้นของหัวใจ ไมโครโปรเซสเซอร์ 8088 เป็นไมโครโปรเซสเซอร์ขนาด 16 บิทที่มีบัสข้อมูลเพียง 8 เส้น แต่เราสามารถใช้ส่งข้อมูลขนาด 16 บิทได้ โดยข้อมูลและคำสั่งจะถูกเฟทช์ (fetch) และเขียน (write) ลงในหน่วยความจำทีละ 8 บิทไมโครโปรเซสเซอร์ 8088 สามารถอินเทอร์เฟสกับหน่วยความจำ ซึ่งใช้เก็บข้อมูล และโปรแกรมที่มีขนาดใหญ่ถึง 1 เมกะไบท์ได้

นอกจากนี้ ไมโครโปรเซสเซอร์ 8088 สามารถใช้ร่วมกับไอซีตัวอื่นเพื่อทำขบวนการDirect Memory Access (DMA) หรือการอินเทอร์รัฟท์ ในการทำขบวนการดังกล่าวกับเครื่อง IBM PC ค่าต่างๆที่ต้องใช้จะถูกกำหนดไว้โดยเฉพาะเสียก่อน

2.2.3 วงจรคล็อก

ในเครื่อง PC สัญญาณฐานเวลาที่ใช้กับไมโครโปรเซสเซอร์ 8088 มีค่า 4.77 MHz ซึ่งมีคาบเวลาประมาณ 210 นาโนวินาทีเนื่องจากว่า bus cycle ส่วนใหญ่ช่วงเวลาที่ใช้ในการทำงานจะใช้ประมาณ 4 คล็อก ดังนั้นใน memory cycle หนึ่งๆ จะใช้เวลาประมาณ 840 นาโนวินาที

สัญญาณฐานเวลาที่ไมโครโปรเซสเซอร์ 8088 และอุปกรณ์ประมวลผลอื่นๆได้รับนั้นส่งมาจากระบบออสซิเลเตอร์ ความถี่เอาท์พุทของวงจรออสซิเลเตอร์ถูกควบคุมโดยคริสตอลขนาด 14.31818 MHz สัญญาณความถี่นี้จะถูกหารสามโดยไอซี เบอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8284A (Clock Generstor and Driver) ซึ่งความถี่ที่ได้จะมีค่าเท่ากับ 4.77 MHz ซึ่งจะถูกป้อนให้แก่ไมโครโปรเซสเซอร์ 8088 สัญญาณฐานเวลาความถี่ 4.77 MHz และ 14.31818 MHz ทั้งสองความถี่นี้เราสามารถดึงมาใช้จากสล็อตต่างๆ ได้

สำหรับวงจร timer-counter นั้นจะใช้ความถี่ประมาณ 1.19 MHz ในการทำงานซึ่งความถี่นี้ได้มาจากการหารสี่สัญญาณคล็อกความถี่ 4.77 MHz

2.2.4 ระบบบัสของ System Board

ระบบบัสของ System Board นั้นประกอบด้วยระบบบัส 2 ระบบด้วยกัน คือ local bus และ system bus อุปกรณ์หลักต่างๆที่อยู่บน System Board จะเชื่อมต่อเข้ากับไมโครโปรเซสเซอร์ 8088 ผ่านทาง System Bus, System Bus ประกอบด้วยบัสที่ใช้ส่งสัญญาณต่างๆ ดังนี้ บัสข้อมูล, บัสแอดเดรส, บัสที่ใช้ส่งสัญญาณควบคุม,สัญญาณฐานเวลา,การขออินเทอร์รัพท์,การควบคุม การทำ DMA (Direct Memory Access)ส่วนของ local bus นั้นจะเริ่มจากขาของไมโครโปรเซสเซอร์ 8088 สัญญาณที่ส่งออกมาเป็นสัญญาณที่ถูกเข้ารหัสและถูกมัลติเพล็กซ์ ซึ่งจะถูกต่อเข้ากับอุปกรณ์ต่อไปนี้คือ

2.4.1 ซ๊อกเก็ตของตัวช่วยประมวลผลซึ่งใช้เสียบ numerical processor เบอร์ 8087 ซึ่งผลิตโดยบริษัท Intel

2.4.2 ตัวควบคุมการอินเทอร์รัพท์เบอร์ 8259A

2.4.3 ตัวควบคุมบัสเบอร์ 8288

2.4.4 วงจรปรับระดับกำลังของสัญญาณที่ส่งในบัสนั้นๆ(bus repowering) และดีมัลติเพล็กซ์เอาท์พุทของตัวควบคุมบัส (bus controller)และตัวดิมัลติเพล็กซ์และปรับระดับกำลังจะทำให้สัญญาณใน local bus เปลี่ยนไปเป็นสัญญาณชุดใหม่ซึ่งเป็นสัญญาณที่ใช้ใน system bus

สำหรับอุปกรณ์ที่ถูกต่อเข้ากับ system bus บน system board ได้แก่ อุปกรณ์สนับสนุนการประมวลผล ซึ่งรวม DMA และ timer/counter เข้าไปด้วย, อุปกรณ์ ถอดรหัสพอร์ท อินพุท เอาท์พุท และแอดเดรสของหน่วยความจำ ROM,RAM, สวิทช์ที่ใช้ระบุลักษณะของระบบ, I/O Adapter และ สล็อตต่างๆ ที่มีอยู่ทั้งหมด 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สล็อต

2.2.5 ROM บน System Board

จากหัวข้อก่อนเราทราบแล้วว่าอุปกรณ์พวกหนึ่งต่อกับบัสของ System Board ได้แก่ ROM ROM ที่ใช้ในเครื่อง IBM PC มีขนาด 40 KB สำหรับตำแหน่งแอดเดรสจำนวน 40 KB ของ ROM ชุดนี้อยู่ในช่วงบนสุดของจำนวน ตำแหน่งแอดเดรสทั้งหมด 1 MB ที่ใช้ไมโครโปรเซสเซอร์ 8088 สามารถระบุแอดเดรสได้ ตำแหน่งแอดเดรสนี้เริ่มต้นจากตำแหน่งที่ F6000 (เลขฐานสิบหก) ถึงตำแหน่งที่ FFFFF อย่างไรก็ตามบน System Board ยังมีซ็อกเก็ตว่างอีก 1 ซ็อกเก็ต ซึ่งสามารถเสียบ ROM หรือ EPROM ขนาด 8 KB ได้อีก 1 ตัว แอดเดรสของซ็อกเก็ตที่ว่างนี้อยู่ที่ ตำแหน่ง F4000 (เลขฐานสิบหก) ถึง F5FFF อนึ่ง System Board ถูกออกแบบให้มีหน่วยความจำอยู่บริเวณหนึ่งซึ่งมีขนาดเท่ากับ 16 KB หน่วยความจำบริเวณนี้ ไม่สามารถเรียกใช้งานได้เพราะไม่มีซ็อกเก็ตบนบอร์ดประมวลผลสำหรับตำแหน่งของหน่วยความจำนี้เริ่มจาก F0000 ถึง F3FFF สำหรับหน้าที่ของ ROM ขนาด 40 KBบน System Board นั้นใช้เป็นที่เก็บ Microcode ซึ่งใช้สนับสนุนการทำงานต่อไปนี้

- System initialization
- การวิเคราะห์ระบบและการตรวจสอบระบบเมื่อเปิดเครื่อง
- การกำหนดคุณลักษณะของระบบ
- ไดร์วเวอร์สำหรับอุปกรณ์ I/O ซึ่งปกติเราเรียกว่า BIOS (Basic Input/Output System)
- Diskette bootstrap loader
- รูปแบบของตัวพิมพ์สำหรับคาร์แรกเตอร์จำนวน 128 ตัวแรกของคาร์แรกเตอร์เซ็ททั้งหมด 256 ตัว(ใช้หน่วยความจำใน ROM ไป 1024 ไบท์)

โปรแกรมการทำงานทั้งหมดที่กล่าวมาถูกเก็บไว้ใน ROM ขนาด 8 KB ROM ตัวนี้มีตำแหน่งแอดเดรสเริ่มต้นที่ FE000 (เลขฐานสิบหก) ถึง FFFFF (เลขฐานสิบหก)รายละเอียดของโปรแกรมสำหรับการทำงานแต่ละหน้าที่หาดูได้จากหนังสือ "IBM Tecnical Reference Manual"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ ROM ที่เหลืออีก 4 ตัว ซึ่งมีหน่วยความจำเท่ากับ 32 KB นั้นใช้เก็บ Microsoft BASIC Interpreter

2.2.6 RAM บน System Board

ขนาดของ RAM บน System Board นั้นมีขนาดเริ่มต้นจาก 16 KB ถึง 64 KB การเพิ่มขนาดหน่วยความจำแบบนี้จะเพิ่มได้ทีละ 16 KB RAM ขนาด 16 KB ตัวแรกจะถูกบัดกรีติดกับ System Board และส่วนที่เหลืออีก 48 KB สามารถขยายเพิ่มได้โดยเสียบลงบนซ็อกเก็ตที่เตรียมไว้ ตำแหน่งของ RAM บน System Board จะเริ่มจากแอดเดรสที่ 00000 (เลขฐานสิบหก) ถึง OFFFF ซึ่งเป็นกรณี ที่เราใช้ RAM ทั้งหมด 64 KB สำหรับในกรณีที่ใช้หน่วยความจำไม่ครบ 64 KBเราไม่สามารถเพิ่มหน่วยความจำที่ขาดไปนี้ในการ์ดอื่นๆ ได้เพราะว่าบัสที่เดินไปยังสล็อตต่างๆ ไม่สามารถบ่งตำแหน่งหน่วยความจำที่มีค่าต่ำกว่า 64K ได้ การขยายหน่วยความจำให้มีขนาดใหญ่กว่า 64 KB โดยอาศัยการ์ดต่างๆนั้น ควรจะทำก็ต่อเมื่อหน่วยความจำบน System Board ถูกขยายจนเต็ม 64 KB แล้ว เนื่องจากถ้าเราขยายโดยไม่ได้ใช้หน่วยความจำบน System Board เต็ม 64 KB จะทำให้การบ่งตำแหน่งแอดเดรสของหน่วยความจำเป็นไปอย่างไม่ต่อเนื่องจะมีบางช่วงที่ขาดไป ซึ่งปกติซอฟท์แวร์ที่ใช้กับเครื่อง IBM PC จะเขียนขึ้นเพื่อใช้กับแอดเดรสที่ต่อเนื่องเท่านั้น

หน่วยความจำบน System Board จริงๆ แล้วจะมีขนาด 9 บิท โดยบิทที่ 9 เป็นบิทพาริตี้ค่าของบิทพาริตี้จะถูกสร้างขึ้นทุกครั้งที่เกิดขบวนการเขียนข้อมูล (write cycle) ขึ้น ค่านี้จะถูกเก็บไว้ในบิทที่ 9 ในทุกๆขบวนการอ่านข้อมูล (read cycle) ค่าของบิท พาริตี้ จะถูกสร้างขึ้นใหม่ จากบิทข้อมูลทั้ง 8 บิท ค่าใหม่นี้จะถูกส่งกลับไปเปรียบเทียบกับค่าของบิท พาริตี้ที่ถูกเก็บไว้ก่อนแล้วในขณะที่เกิดขบวนการเขียนข้อมูล ถ้าค่าของบิท พาริตี้ของทั้งสองขบวนการตรงกันแสดงให้รู้ว่า ข้อมูลที่เขียนเข้าไปและอ่านกลับออกมาเป็นข้อมูลที่เหมือนกัน แต่ถ้าตรวจสอบแล้วบิท พาริตี้ไม่ตรงกัน จะเกิดขบวนการ parity check ขึ้น และมีการส่งสัญญาณไปยังระบบโดยการอินเทอร์รัพท์ไมโครโปรเซสเซอร์ 8088 หลังจากนั้นระบบจะส่งข้อความไปยังอุปกรณ์แสดงผลโดยอาศัยซอฟท์แวร์ของระบบ และการประมวลผลจะถูกระงับไว้ก่อน (Halt)

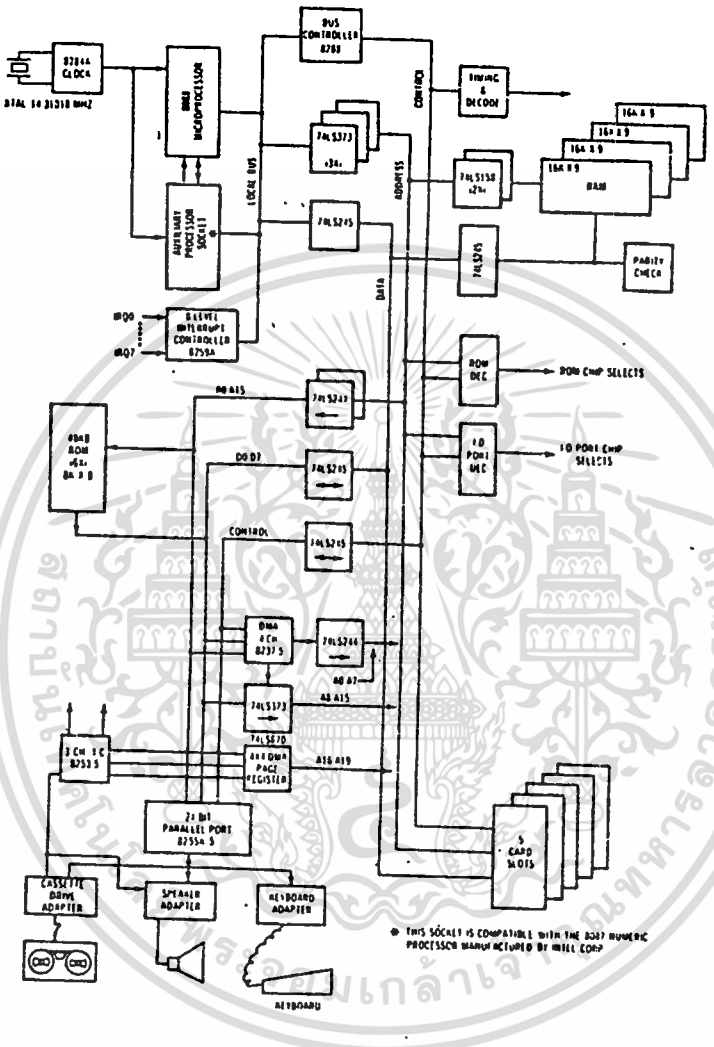
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน่วยความจำที่ใช้ใน System Board เป็นหน่วยความจำแบบไดนามิกแรม ขนาด 16Kx1 ซึ่งมี access time เท่ากับ 250 นาโนวินาที สำหรับระดับแรงดันที่ใช้กับไอซีแบบนี้ใช้ระดับสามระดับด้วยกันคือ +5 โวลท์, -5 โวลท์ และ +12 โวลท์

2.2.7 ส่วนของ Timer/Counter ของ System Board

อุปกรณ์ที่ใช้สนับสนุนการนับ (counting function) และฐานเวลาของเครื่อง PC คือ ไอซีเบอร์ 8253-5 timer/counter ซึ่งมีส่วนของเคาน์เตอร์ขนาด 16 บิท อยู่ 3 ตัว ภายในชิพตัวเดียวกัน เคาน์เตอร์ทั้งสามตัวนี้ถูกต่อเข้ากับ System Bus สัญญาณคล็อกที่ป้อนให้แก่เคาน์เตอร์ทั้งสามนี้มีค่าเท่ากับ 1.19 MHzเอาท์พุทของเคาน์เตอร์ แชนแนลที่ 0 ถูกต่อเข้ากับ อินเทอร์รัพท์ระดับ 0 และถูกโปรแกรมให้ทำการอินเทอร์รัพท์ทุกๆ 54.925 มิลลิวินาที คือจะมีการ อินเทอร์รัพท์ประมาณ 18.2 ครั้งต่อวินาที สัญญาณเวลานี้จะถูกใช้งานโดย System I/O Routine และส่วนแสดงเวลาของระบบ ส่วนเอาท์พุทของเคาน์เตอร์แชนแนลที่ 1 ใช้ในการขอทำ DMA ของ DMA แชนแนลที่ 0 ซึ่งถูกใช้ในการรีเฟรชหน่วยความจำแบบไดนามิกของระบบ โดยการอ่านข้อมูล จากหน่วยความจำทุกๆ 72 processor clock หรือ ทุกๆ 15.12 ไมโครวินาที ส่วนเอาท์พุทของเคาน์เตอร์แชนแนลที่ 2 ใช้ในการขับลำโพงและใช้ส่งข้อมูลไปยังพอร์ท เครื่องเล่นเทปที่มีอยู่บน System Board

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 บล็อกไดอะแกรมของ System Board

2.2.8 ส่วนของ DMA บน System board

การส่งข้อมูลของอุปกรณ์อินพุท/เอาท์พุทบางตัว เช่น ดิสเก็ตไดรฟ์นั้นมีอัตราการส่งข้อมูลสูงกว่าความสามารถของตัวประมวลผลที่จะนำข้อมูลเหล่านั้นไปใช้ในการประมวลผลมาก ดังนั้นถ้าเราต้องโอนย้ายข้อมูลผ่านตัวไมโครโปรเซสเซอร์ 8088 จะทำให้เสียเวลามาก ปัญหาข้อนี้สามารถแก้ไขได้โดยใช้ความสามารถ ของอุปกรณ์อินพุท/เอาท์พุทข้อนี้ทำการโอนย้ายข้อมูลจากอุปกรณ์ภายนอกหรือ Adapter กับหน่วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความจำโดยไม่ต้องยุ่งเกี่ยวกับตัวไมโครโปรเซสเซอร์ 8088 ซึ่งช่วยให้การโอนย้ายข้อมูลทำได้อย่างรวดเร็ว การทำงานในลักษณะนี้ อาศัยชิพตัวหนึ่งที่เรียกว่า DMA Controller สำหรับในเครื่อง PC เราใช้ชิพเบอร์ 8237-5 DMA Controller ชิพตัวนี้จะสนับสนุนการโอนย้ายข้อมูลจาก Adapter ทั้ง 4 กับหน่วยความจำหรือจากหน่วยความจำไปยัง Adapter ทั้ง 4 เมื่อ Adapter หรืออุปกรณ์ใดๆต้องบการโอนย้ายข้อมูลมันจะส่งสัญญาณบอกให้ 8237-5 ทราบเพื่อทำ DMA 8237-5 จะส่งสัญญาณไปบอกให้ไมโครโปรเซสเซอร์ 8088 ตัดตัวเองออกจาก System Bus หลังจากนั้น DMA Controller จะควบคุม System Bus แทนและทำการโอนย้ายข้อมูลต่อไป อนึ่ง DMA Controller ตัวหนึ้สามารถระบุค่าแอดเดรสได้สูงสุด 64 KB เท่านั้นแต่เนื่องจากหน่วยความจำทั้งหมดของเครื่อง PC สามารถมีได้ถึง 1 MB เราจึงต้องเพิ่มบิทแอดเดรสพิเศษเข้าไปเพื่อช่วยให้ DMA Controller สามารถระบุแอดเดรสได้ทั้งหมด บิทของแอดเดรสที่เพิ่มเข้าไปนี้เพิ่มจาก page registers ขนาด 4 บิท จำนวน 4 ตัว โดยในแต่ละ DMA Channel จะมี page registers ขนาด 4 บิท อยู่ 1 ตัว บิททั้ง 4 ของ page register แต่ละตัวนี้ ถูกใช้เป็นแอดเดรสบิทสูงๆ ของการระบุตำแหน่งหน่วยความจำในระหว่างการทำ DMA Cycle แชนแนล 0 ของ DMA ถูกใช้ในการรีเฟรชหน่วยความจำแบบไดนามิกโดย System Board ส่วนแชนแนลที่เหลืออีก 3 แชนแนลนั้นต่อเข้ากับ System Bus จึงสามารถเรียกใช้ผ่านทางสล็อตทั้ง 5 บน System Board ได้

2.2.9 การอินเทอร์รัพท์ที่เกิดขึ้นบน System Board

การอินเทอร์รัพท์ไมโครโปรเซสเซอร์ 8088 มีอยู่ด้วยกัน 2 แบบ คือ แบบ Maskable และ Nonmaskable แต่เนื่องจากเราต้องการจัดลำดับความสำคัญของอุปกรณ์ต่างๆ ที่ขอทำการอินเทอร์รัพท์ ว่าอุปกรณ์ตัวใดมีความสำคัญต่อระบบมากที่สุด ดังนั้นจึงมีการใช้ชิพเบอร์ 8259A Interrupt Controller ช่วยใน การจัดระดับความสำคัญ อุปกรณ์ตัวนี้ช่วยขยายการอินเทอร์รัพท์แบบ Maskable ที่ส่งไปให้ 8088 ออกเป็น 8 ระดับ 8259A นั้นถูกต่อเข้ากับ local bus ซึ่งต่อเข้า กับไมโครโปรเซสเซอร์ 8088 โดยตรง จากระดับความสำคัญทั้ง 8 ระดับนี้ สองระดับถูกใช้โดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

System Board ที่เหลืออีก 6 ระดับ สามารถเรียกใช้จาก System Bus ได้ ซึ่งก็คือมันถูกต่อเข้ากับสล็อตทั้ง 5 ที่มีอยู่บน System Board ระดับความสำคัญสองระดับที่ใช้โดย System Board นั้นใช้ทำหน้าที่ต่อไปนี้คือระดับ 0 ใช้ในการรับการอินเทอร์รัพท์จากเอาท์พุทแชนแนลที่ 0 ของ temer/counter ส่วนระดับที่ 1 ใช้ในการรับการอินเทอร์รัพท์จากคีย์บอร์ดของเครื่อง IBM PC

2.2.10 I/O Adapter ที่มีอยู่บน System Board

Adapter ต่างๆ ที่มีอยู่บน System Board นั้นถูกใช้งานร่วมกับคีย์บอร์ด, ลำโพง และพอร์ทเครื่องเล่นเทปอุปกรณ์เหล่านี้ถูกอินเทอร์เฟสเข้ากับ System Bus ผ่านทางชิพรีจิสเตอร์แบบขนานชิพที่ถูกใช้ในกรณีนี้ก็คือชีพเบอร์ 8255A-5 Programmable Peripheral Interface ซึ่งมีจำนวนบิทที่ใช้ในการส่งและรับข้อมูลอยู่ 24 บิท ที่เราสามารถโปรแกรมสั่งให้ส่งหรือรับ (input/output) ข้อมูลได้

พอร์ทขนาด 8 บิท พอร์ทหนึ่งของ 8255A-5 จะถูกใช้สร้าง sub-busซึ่งใช้ในการอ่านข้อมูลจากดิพสวิทช์ (dip switchs) เพื่อจะได้ทราบตำแหน่งของสวิทช์แต่ละตัวหรือใช้ในการอ่าน scan code จาก input shift register ของคีย์-บอร์ด ส่วนบิทที่เหลือ จะถูกใช้ในการขับลำโพง, ใช้อ่านข้อมูลแบบอนุกรม ที่ส่งมาจากเครื่องเล่นเทป, ใช้ควบคุมการทำงาน ของมอเตอร์ของเครื่องเล่นเทป, ใช้ในการมอดดูเลตสัญญาณคล็อก ที่ได้จากแชนแนลที่สองของ timer counter, ใช้ตรวจสอบแล็ทช์และสภาพของดิพสวิทช์ และใช้ในการอินเทอร์เฟสแบบอนุกรมกับคีย์บอร์ด

2.3 สัญญาณต่าง ๆ บนสล็อตของ IBM/PC

ภายใน IBM/PC ได้มีการออกแบบให้สามารถที่จะเพิ่มเติมวงจรอินเทอร์เฟสเข้าไปในภายหลังได้ โดยผ่านทางสล็อตที่อยู่บนเมนบอร์ด (Main Board) สำหรับสล็อตบนเมนบอร์ดนี้จะมีจำนวน 5 สล็อต (สำหรับใน IBM PC/XT จะมี 8 สล็อต) ซึ่งแต่ละสล็ต จะมีจำนวนขาทั้งสิ้น 62 ขา แบ่งออกเป็น 2 ข้างๆ ละ 31 ขา ส่วนการเรียกตำแหน่งขาของสล็อตเหล่านี้ จะขึ้นอยู่กับว่าขานั้นอยู่ข้างใด (ซ้ายหรือขวา) ของสล็อต โดยขาที่อยู่ทางด้านซ้ายของสล็อตจะเรียกโดยใช้อักษร "B" นำหน้าเลขตำแหน่งของขา เช่นขา B16 ก็คือ ขาทางด้านซ้ายของสล็อตขาที่ 16 (นับจากทาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้านท้ายของเครื่อง) ส่วนขาที่อยู่ทางด้านขวาของสล็อต จะเรียกโดยใช้อักษร "A" นำหน้าเลขตำแหน่งของขา เช่น ขา A24 ก็คือขาทางด้านขวาของสล็อต ขาที่ 24 (นับจากทางด้านท้ายของเครื่อง)

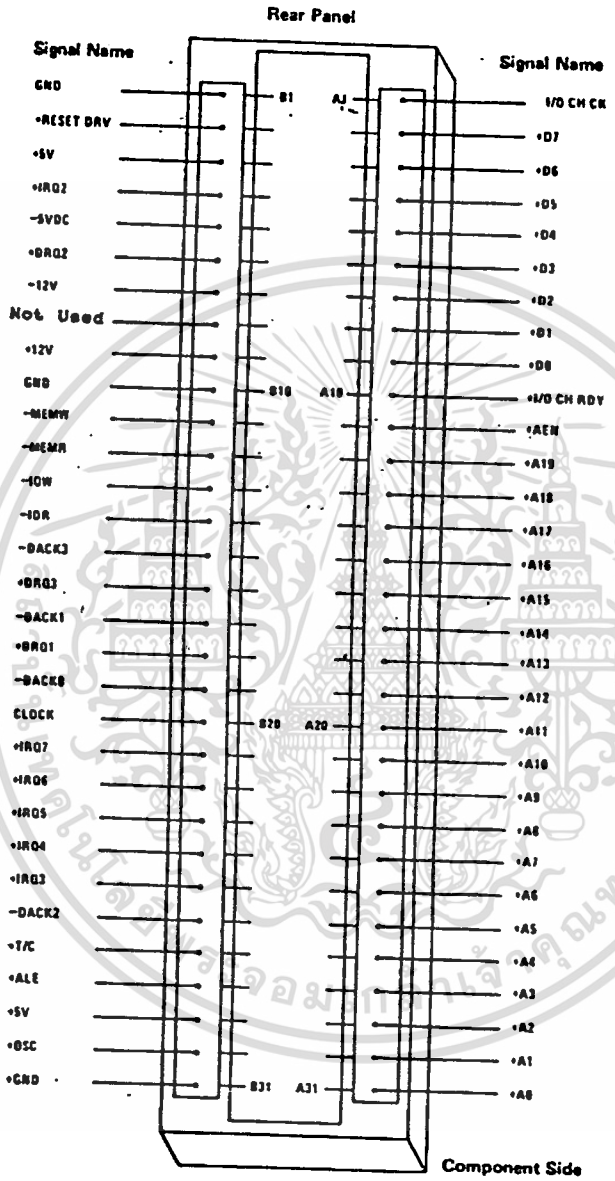
แต่ละขาของสล็อตเหล่านี้จะเชื่อมต่อกับเส้นสัญญาณต่างๆ บนเมนบอร์ดทำให้การสร้างวงจรอินเทอร์เฟสกับ IBM/PC สามารถทำได้โดยสะดวก ซึ่งเส้นสัญญาณที่เชื่อมต่อกับขา ของสล็อตเหล่านี้ จะประกอบไปด้วย เส้นสัญญาณของบัสแอดเดรส (Address Bus), บัสข้อมูล (Data Bus), บัสควบคุมสำหรับการเขียน/อ่านข้อมูลจากหน่วยความจำ หรือพอร์ท I/O, เส้นสัญญาณสำหรับการขออินเทอร์รัพท์ของวงจรอินเทอร์เฟส, เส้นสัญญาณสำหรับการขอ DMA, สัญญาณฐานเวลา (Timing Signal) ต่างๆ ที่ใช้ในระบบ, เส้นสัญญาณแสดงการรีเฟรชหน่วยความจำ และสัญญาณสำหรับการตรวจสอบความผิดพลาด (I/O CHCK) นอกจากเส้นสัญญาณเหล่านี้แล้ว สล็อตบนเมนบอร์ดยังเชื่อมต่อกับแหล่งจ่ายไฟต่างๆที่ใช้ในระบบอีกด้วย คือ +5Vdc, -5Vdc, +12Vdc และ -12Vdc

2.3.1 รายละเอียดเกี่ยวกับสัญญาณต่าง ๆ

2.3.1.1 OSC (Oscillator; ขา B30) :

ขานี้เป็นเอาท์พุทที่เชื่อมต่อกับสัญญาณคล็อกที่มีค่าความถี่สูงสุดบนเมนบอร์ด คือ 14.31818 MHz ซึ่งมีคาบเวลาประมาณ 70 nanosec. และมี Duty Cycle (ช่วงเวลาใน 1 คาบที่สัญญาณคล็อกมีลอจิกเป็น "1" หารด้วยคาบเวลาทั้งหมด) ประมาณ 50% สัญญาณคล็อกอื่นๆ ของระบบ เช่น คล็อกที่ป้อนให้กับ 8088 หรือชิพซัพพอร์ทต่างๆ นั้นจะถูกสร้างขึ้นโดยการหารสัญญาณคล็อกนี้อย่างไรก็ตามสิ่งหนึ่งที่จะต้องคำนึงถึงในการใช้งานสัญญาณ OSC ก็คือ สัญญาณนี้จะไม่ Synchronize กับสัญญาณอื่นๆ บนบัสของระบบ ดังนั้น จึงไม่ควรที่จะนำสัญญาณจากขา OSC นี้ไปใช้เป็นสัญญาณคล็อกสำหรับวงจรภายนอกอื่นๆ ที่ทำงานร่วมกับระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 แสดงตำแหน่งของขาของสลอ็ต

2.3.1.2 CLK (Clock; ขา B20) :

ขาสัญญาณนี้เป็นเอาท์พุทซึ่งต่อกับสัญญาณคล็อกที่ถูกสร้างขึ้นโดยการหารสัญญาณ OSC ด้วย 3 ทำให้ได้ความถี่ประมาณ 4.77 MHz (14,31818 MHz/3) หรือมีช่วง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เวลาใน 1 คาบ (ช่วงเวลาของคล็อก 1 ลูก) เท่ากับ 210 nanosec. (1/4.77 MHz) สำหรับค่า Duty Cycle ของสัญญาณนี้จะมีค่าประมาณ 1/3 คือใน 1 คาบจะมีช่วงเวลาที่เป็นลอจิก "1" เท่ากับ 1/3 ของคาบเวลาทั้งหมด หรือ ประมาณ 70 nanosec. และช่วงเวลาที่เป็นลอจิก "0" เท่ากับ 2/3 ของคาบเวลาทั้งหมด หรือประมาณ 140 nanosec.

สัญญาณนี้เป็นสัญญาณที่ถูกใช้เป็นคล็อกของระบบ

2.3.1.3 RESET DRV (ขา B2) :

ขาสัญญาณนี้เป็นเอาท์พุทซึ่งจะแอคทีฟ (ลอจิก "1") ในช่วงที่เราเริ่มจ่ายไฟให้กับระบบ และจะยังคงแอคทีฟไปจนกว่าระบบต่างๆ ภายใน IBM/PC พร้อมที่จะทำงานได้ จากนั้นสัญญาณนี้ก็จะเปลี่ยนกลับเป็นลอจิก "0" นอกจากนี้ ในระหว่างการทำงานของ IBM/PC ถ้าระดับแรงดันของแหล่งจ่ายไฟตกลงสัญญาณนี้ก็จะถูกทำให้แอคทีฟเช่น โดยทั่วไปแล้วสัญญาณนี้จะถูกนำไปใช้ในการรีเซ็ทวงจร อินเทอร์เฟสหรืออุปกรณ์ I/O ต่างๆ ในช่วงที่เริ่มจ่ายไฟให้กับระบบซึ่งจะเป็นการทำให้วงจรหรืออุปกรณ์เหล่านั้นถูกปรับให้อยู่ในสภาวะที่แน่นอน ก่อนที่จะเริ่มต้นการทำงานในระบบ (สภาวะนี้เป็นสภาวะที่เราทราบ และต้องการให้วงจรทำงานในขณะที่ระบบถูกรีเซ็ท)

2.3.1.4 A0-A19 (Address Bus; ขา A31-A12) :

ขาสัญญาณทั้ง 20 ขานี้เป็นเอาท์พุท ซึ่งใช้สำหรับกำหนดแอดเดรสของหน่วยความจำหรืออุปกรณ์ I/O ที่ 8088 ต้องการติดต่อด้วยโดยที่สัญญาณ A0 จะมีนัยสำคัญต่ำสุด (Least Significant Bit) และ A19 จะมีนัยสำคัญสูงสุด (Most Significant Bit) สำหรับค่าแอดเดรสบนบัสแอดเดรส A0-A19 นี้จะถูกกำหนดโดย 8088 ในระหว่างขบวนการอ่าน/เขียนข้อมูลลงในหน่วยความจำหรืออุปกรณ์ I/O แต่ในช่วงของขบวนการ DMA นั้น DMA-Controller จะเป็นผู้กำหนดค่าแอดเดรสบนบัสแอดเดรสเอง (ในระหว่างนี้ 8088 จะถูกตัดออกจากระบบ) จะเห็นได้ว่าจำนวนเส้นแอดเดรสนี้จะมีอยู่ 20 เส้นซึ่งสามารถที่จะอ้างแอดเดรสของหน่วยความจำได้ถึง 1 Mbyte แต่อย่างไรก็ตามจะมีแอดเดรสบางแอดเดรสที่ถูกใช้งานโดย IBM/PC อยู่ก่อนแล้ว คือแอดเดรสของหน่วยความจำ RAM บนเมนบอร์ดที่ถูกใช้โดยระบบ จำนวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

64 Kbyte (สำหรับ IBM PC/XT จะเป็นจำนวน 256 Kbyte) และ แอดเดรสสำหรับหน่วยความจำ ROM อีก 48 Kbyte ซึ่งถูกจัดในช่วงของแอดเดรสบนสุดใน 1 M byte คือ OFCOOH จนถึง OFFFFFH (สำหรับ IBM PC/XT จะเป็น 64Kbyte)

สำหรับการอ้างแอดเดรสของพอร์ท I/O นั้น จะใช้เส้นแอดเดรสเพียง 16 เส้น คือ A0-A15 ซึ่งจะทำให้อ้างแอดเดรสของพอร์ทได้ 64K พอร์ท โดยผ่านทางชุดคำสั่ง IN และ OUT ส่วนเส้นแอดเดรสที่เหลือคือ A16-A19 นั้นจะไม่ถูกใช้งาน อย่างไรก็ตาม ภายใน IBM/PC จะใช้เส้นแอดเดรส ในการอ้างแอดเดรสของพอร์ทเพียง 10 เส้น คือจาก A0-A9 และค่าแอดเดรสที่ใช้งาน จะต้องอยู่ในช่วง 0200H จนถึง 03FFH เท่านั้น

2.3.1.5 D0-D7 (Data Bus; ขา A9-A2):

ขาสัญญาณนี้ จะเป็นแบบ Bi-Directional ซึ่งต่อกับบัสข้อมูลของระบบเพื่อทำหน้าที่ในการส่งผ่านข้อมูล ระหว่างพอร์ท I/O กับ IBM/PC โดยบิท D0 จะมีนัยสำคัญต่ำสุด และบิท D7 จะมีนัยสำคัญสูงสุด

สำหรับในบัสไซเคิลของการเขียนข้อมูบที่สร้างขึ้นโดย 8088 นั้น ข้อมูลจะถูกส่งออกมาบนบัสข้อมูล ก่อนที่สัญญาณ $\overline{IOW}$ (ในกรณีที่ต้องการส่งข้อมูลให้กับพอร์ท)หรือ $\overline{MEMW}$ (ในกรณีที่ต้องการส่งข้อมูลให้กับหน่วยความจำ) จะเปลี่ยนจากลอจิก "0"เป็นลอจิก "1" (ขอบขาขึ้น) ซึ่งโดยทั่วไปขอบขาขึ้นของสัญญาณ $\overline{IOW}$ หรือ $\overline{MEMW}$ นี้ จะถูกใช้เพื่อสั่งให้พอร์ท I/O หรือหน่วยความจำที่มีแอดเดรส ตรงกับค่าแอดเดรสบนบัสแอดเดรสนั้นรับข้อมูลไปเก็บไว้

สำหรับในบัสไซเคิลของการอ่านข้อมูลที่สร้างขึ้นโดย 8088 นั้น พอร์ท I/O หรือหน่วยความจำที่ถูกอ้างถึง จะต้องส่งข้อมูลออกมาบนบัสข้อมูล ก่อนที่สัญญาณ $\overline{IOR}$ (ในกรณีที่ต้องการอ่านข้อมูลจากพอร์ท) หรือ $\overline{MEMR}$ (ในกรณีที่ต้องการอ่านข้อมูลจากหน่วยความจำ) จะเปลี่ยนจากลอจิก "0" เป็นลอจิก "1" (ขอบขาขึ้น)

2.3.1.6 ALE (Address Latch Enable; ขา B28):

ขาสัญญาณนี้เป็นสัญญาณเอาท์พุทที่ 8288 Bus Controller สร้างขึ้นเพื่อใช้สำหรับแสดงการเริ่มต้นของบัสไซเคิล และแสดงให้อุปกรณ์ภายนอกทราบว่าแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ 8088 ต้องการจะติดต่อด้วยนั้นถูกส่งออกมาบนบัสแอดเดรสแล้ว โดยที่สัญญาณ ALE นี้จะเปลี่ยนจากลอจิก "1" เป็น "0" เมื่อค่าแอดเดรสที่ถูกต้องถูกส่งออกมาบนบัสข้อมูลเรียบร้อยแล้วดังนั้นขอบขาลงของสัญญาณ ALE นี้จะถูกใช้ในการแลทช์ค่าแอดเดรสจากบัสแอดเดรส/ข้อมูล (Address/Data Bus; AD0-AD7) ของ 8088 ทำให้สามารถแยกค่าแอดเดรส (A0-A19) และข้อมูล (A0-A7) ออกจากกันได้ อย่างไรก็ตาม สัญญาณ ALE จะแอดทีฟเฉพาะในบัสไซเคิลที่สร้างขึ้นโดย 8088 เท่านั้น โดยจะไม่แอดทีฟในระหว่างขบวนการ DMA

- 2.3.1.7 $\overline{\text{I/O CHCK}}$ (I/O Channel Check; ขา A1):

ขาสัญญาณนี้เป็นอินพุทที่ใช้ในการแสดงความผิดพลาดเกี่ยวกับพาริตี้ ที่เกิดขึ้นในการทำงานของวงจรอินเทอร์เฟสหรืออุปกรณ์ I/O เมื่อขาสัญญาณนี้ได้รับลอจิก"0" จะทำให้ 8088 ถูกอินเทอร์รัพท์แบบ Non-Maskable (NMI) อย่างไรก็ตาม เราสามารถที่จะกำหนดให้วงจรภายในของ IBM/PC ทำการขออินเทอร์รัพท์ (เมื่อได้รับสัญญาณ $\overline{\text{I/O CHCK}}$) หรือไม่ก็ได้โดยการกำหนดลอจิกของบิทข้อมูลของพอร์ทที่ควบคุมการขอ อินเทอร์รัพท์แบบ NMI คือบิท D7 ของพอร์ท 00A0H ในกรณีที่ บิท D7 ของพอร์ท 00A0H ถูกเซ็ทเป็น "1" ก็จะทำให้วงจรภายนอกขออินเทอร์รัพท์แบบ NMI ได้ (Enable) แต่ถ้าบิท D7 ของพอร์ท 00A0H ถูกเซ็ทเป็น "0" จะเป็นการดิสเอเบิล (Disable) การขออินเทอร์รัพท์แบบ NMI ดังนี้

Enable : ใช้คำสั่ง OUT ส่งข้อมูล 80H ไปยังพอร์ท 00A0H

Disable : ใช้คำสั่ง OUT ส่งข้อมูล 00H ไปยังพอร์ท 00A0H

และเนื่องจากยังมีอุปกรณ์อื่นที่สามารถขออินเทอร์รัพท์แบบ NMI ได้อีกดังนั้นซอฟท์แวร์ที่ใช้งานจะต้องสามารถตรวจสอบว่าการขออินเทอร์รัพท์นั้นเกิดขึ้นจากแหล่งใดได้ด้วย

2.3.1.8 I/O CHRDY (I/O Channel Ready; ขา A10):

ขาสัญญาณนี้เป็นอินพุทที่ใช้เพิ่มช่วงเวลาในบัสไซเคิลในกรณีที่อุปกรณ์ I/Oหรือหน่วยความจำที่เกี่ยวข้องกับขบวนการในบัสไซเคิลที่เกิดขึ้นนั้น ไม่สามารถทำงานทัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตามช่วงเวลาปกติของบัสไซเคิลนั้นๆ ได้ (ช่วงเวลาของบัสไซเคิลที่เกี่ยวกับหน่วยความจำใช้ช่วงเวลาเท่ากับช่วงเวลาของคล็อก 4 ลูก หรือ 840 nanosec.ในขณะที่บัสไซเคิลที่เกี่ยวกับ I/O จะใช้ช่วงเวลา เท่ากับช่วงเวลาของคล็อก 5 ลูก หรือ 1.05 usec.)

เมื่ออุปกรณ์ I/O หรือหน่วยความจำต้องการที่จะเพิ่มช่วงเวลาในบัสไซเคิลให้นานขึ้นอีกนั้น จะสามารถทำได้โดยการป้อนลอจิก "0" ให้กับขา I/O CHRDY ในช่วงเวลาที่ I/O หรือ หน่วยความจำที่ถูกกำหนดนั้น ได้รับสัญญาณจากการดีโค้ดแอดเดรส และสัญญาณ $\overline{\text{MEMR}}$, $\overline{\text{MEMW}}$, $\overline{\text{IOR}}$ หรือ $\overline{\text{IOW}}$ แอคทีฟ

2.3.1.9 IRQ2-IRQ7 (Interrupt Request 2 Through 7; ขา B4 และ B25-B21) :

ขาสัญญาณทั้ง 6 นี้เป็นขาอินพุทที่ใช้สำหรับการขออินเทอร์รัพท์จาก 8088 โดยสัญญาณเหล่านี้จะต่อเข้ากับ 8259A บนเมนบอร์ดโดยตรงโปรแกรมในส่วน BIOS ของ IBM/PC จะทำการโปรแกรม 8259A ให้ IRQ2 มีลำดับความสำคัญสูงสุด (Highest Priority) และ IRQ7 มีลำดับความสำคัญต่ำสุด ในกรณีที่มีการขออินเทอร์รัพท์เกิดขึ้น คือระดับลอจิกที่ขา IRQ ขาใดขาหนึ่งถูกเปลี่ยนจากลอจิก "0" เป็นลอจิก "1" (ขอบขาขึ้น) 8259A ก็จะทำการส่งสัญญาณ INT ให้กับ 8088 เพื่อทำการขออินเทอร์รัพท์

สิ่งสำคัญในการขออินเทอร์รัพท์โดยผ่านทาง IRQ2-IRQ7 นี้ ก็คืออุปกรณ์ที่ทำการขออินเทอร์รัพท์โดยผ่านทาง IRQ ขาใดก็จะต้องรักษาระดับสัญญาณที่ขา IRQ นั้นให้แอคทีฟ (ลอจิก "1") อยู่จนกว่าจะได้รับสัญญาณ INTA (Interrupt Acknowledge) จาก 8088 เสียก่อน ถ้าไม่เช่นนั้นการขออินเทอร์รัพท์จะถูกยกเลิก และอินเทอร์รัพท์ Level 7 (IRQ7) ก็จะถูกสร้างขึ้น
โดยอัตโนมัติ ไม่ว่าการขออินเทอร์รัพท์ที่ถูกยกเลิกนั้น จะเป็นการขออินเทอร์รัพท์ใน Level หรือขา ใด

แต่อย่างใดก็ตามสัญญาณ INTA นี้จะไม่ถูกต่อออกมาที่ขาของสล็อตด้วย ดังนั้นโปรแกรมที่ทำการตอบสนอง ต่อการขออินเทอร์รัพท์ (Interrupt Service

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Routine) จะต้องทำการรีเซ็ทสัญญาณ IRQ เอง โดยใช้คำสั่ง OUT ไปยังพอร์ท I/O ที่เกี่ยวข้อง

2.3.1.10 $\overline{\text{IOR}}$ (I/O Read; ขา B14) :

ขาสัญญาณนี้เป็นเอาท์พุท แอคทีฟที่ลอจิก "0" ที่สร้างขึ้นโดย 8288 Bus Controller เพื่อใช้ในการแสดงว่าบัสไซเคิบที่เกิดขึ้นนี้ เป็นบัสไซเคิลของการอ่านข้อมูลจากพอร์ท I/O เพื่อให้พอร์ท I/Oที่มีแอดเดรสตรงกับแอดเดรสบนบัสแอดเดรสนั้นส่งข้อมูลออกมาบนบัสข้อมูล โดยข้อมูลจะต้องถูกส่งออกมาบนบัสข้อมูลก่อนขอบขาขึ้นของสัญญาณ $\overline{\text{IOR}}$ ประมาณ 30 nanosec. เพื่อให้มั่นใจได้ว่า 8088 สามารถรับข้อมูลได้ถูกต้อง สำหรับในขบวนการ DMA 8237A-5 DMA Controller จะทำการสร้างสัญญาณ $\overline{\text{IOR}}$ เองโดยที่ค่าแอดเดรสที่อยู่บนบัสแอดเดรสจะเป็นค่าแอดเดรสของหน่วยความจำ(แทนที่จะเป็นแอดเดรสของพอร์ท I/O) ที่พอร์ท I/O ที่ขอ DMA ต้องการจะนำข้อมูลไปเก็บ การที่พอร์ทใดจะส่งข้อมูลออกมาบนบัสข้อมูลนั้นจะอาศัยสัญญาณ DACK จาก DMA Controller เป็นตัวกำหนดเช่นกรณีที่สัญญาณ DACK1 แอคทีฟก็จะแสดงว่าพอร์ท I/O ที่จะต้องส่งข้อมูลออกมาบนบัสข้อมูลก็คือพอร์ท I/O ที่ขอ DMA ผ่านทางแชนแนลที่ 1 (DRQ1) เป็นต้น

2.3.1.11 $\overline{\text{IOW}}$ (I/O Write; ขา B13) :

ขาสัญญาณนี้เป็นเอาท์พุทแอคทีฟที่ลอจิก "0" ซึ่งถูกสร้างขึ้นโดย 8288 Bus Controller เพื่อใช้แสดงว่าบัสไซเคิลที่เกิดขึ้นนี้ เป็นบัสไซเคิลของการเขียนข้อมูลลงบนพอร์ท I/O เพื่อให้พอร์ท I/O ที่มีแอดเดรสตรงกับแอดเดรสบนบัสแอดเดรสนั้นรับข้อมูบที่อยู่บนบัสข้อมูลไปเก็บไว้ อย่างไรก็ตามเนื่องจากในช่วงเวลาที่สัญญาณ $\overline{\text{IOW}}$ นี้แอคทีฟ(ลอจิก "0") นั้นข้อมูลบนบัสข้อมูลอาจจะยังไม่สมบรูณ์ ดังนั้นในการออกแบบจึงควรใช้ขอบขาขึ้นของสัญญาณ $\overline{\text{IOW}}$ แทนขอบขาลงในการทำให้พอร์ท I/O ที่เกี่ยวข้องรับข้อมูลไปเก็บไว้ เพื่อให้ข้อมูลบนบัสข้อมูลสมบูรณ์เสียก่อน สำหรับในขบวนการ DMA นั้น DMA-Controller จะทำการสร้างสัญญาณ $\overline{\text{IOW}}$ เอง โดยที่ค่าแอดเดรสที่อยู่บนบัสแอดเดรสจะเป็นค่าแอดเดรสของหน่วยความจำที่พอร์ท I/O ที่ขอ DMA ต้องการจะอ่านข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.12 $\overline{\text{MEMW}}$ (Memory Write; ขา Bill) :

ขานี้เป็นเอาท์พุทแอคทีฟที่ลอจิก "0" ซึ่ง 8288 Bus Controller สร้างขึ้นในระหว่างบัสไซเคิล ในการเขียนข้อมูลลงในหน่วยความจำของ 8088 สัญญาณ $\overline{\text{MEMW}}$ นี้จะถูกส่งออกมาเพื่อให้หน่วยความจำที่แอดเดรสตรงกับค่าแอดเดรสบนบัสแอดเดรสนั้น ทำการรับข้อมูลที่อยู่บนบัสข้อมูลไปเก็บไว้โดยทั่วไปหน่วยความจำจะรับข้อมูลในช่วงขอบขาขึ้นของสัญญาณ $\overline{\text{MEMW}}$

สำหรับในระหว่างขบวนการ DMA นั้น 8237A-5 DMA-Controller จะทำการควบคุมบัสต่างๆ ของระบบแทน 8088 และสัญญาณ $\overline{\text{MEMW}}$ จะถูกใช้ในบัสไซเคิลของการเขียนข้อมูลลงในหน่วยความจำ (ข้อมูลถูกส่งจากอุปกรณ์ I/O ไปให้กับหน่วยความจำ)

2.3.1.13 $\overline{\text{MEMR}}$ (Memory Read; ขา B12) :

ขานี้เป็นเอาท์พุทจาก 8288 ซึ่งสัญญาณนี้จะแอคทีฟ (ลอจิก "0") ในระหว่างบัสไซเคิลของการอ่านข้อมูล จากหน่วยความจำของ 8088 เพื่อให้หน่วยความจำ ที่มีแอดเดรสตรงกับค่าแอดเดรสบนบัสแอดเดรสนั้น ทำการส่งข้อมูลออก มาบนบัสข้อมูล โดยหน่วยความจำนั้นจะต้องส่งข้อมูลออกมาในช่วงเวลา 30 nanosec.ก่อนที่สัญญาณ $\overline{\text{MEMW}}$ จะกลับเป็นลอจิก "1" ทั้งนี้ก็เพื่อให้ 8088 ได้รับข้อมูลที่ถูกต้อง

สำหรับในระหว่างขบวนการ DMAนั้น DMA-Controller จะควบคุมบัสต่างๆ ของระบบแทน 8088 และสัญญาณ $\overline{\text{MEMR}}$ จะถูกใช้ในบัสไซเคิลของการอ่านข้อมูลจากหน่วยความจำ (ข้อมูลถูกส่งจากหน่วยความจำไปให้กับอุปกรณ์ I/O)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.14 DRQ1-DRQ3 (DMA Request1-3;ขา B18,B6และขา B16): ขาสัญญาณทั้งสามนี้ เป็นสัญญาณอินพุทแอคทีฟที่ลอจิก "1" ซึ่งอุปกรณ์ภายนอกสามารถใช้ในการขอ DMA จากระบบ โดยการป้อนระดับสัญญาณลอจิก "1" ให้กับขา DRQ ขาใดขาหนึ่ง (ขา DRQ ทั้งสามนี้จะต่อเข้ากับ DRQ1-DRQ3 ของ 8237A-5)

เมื่อ 8237A-5ได้รับสัญญาณนี้แล้วก็จะตรวจสอบว่ามีการขอ DMA จาก 8088 และตอบรับการขอ DMA จากอุปกรณ์ภายนอก (สัญญาณ $\overline{\text{DACK}}$ ของแชนแนลที่ขอ DMA จะแอคทีฟ)แต่ถ้ามี 8237A-5 ก็จะทำการขอ DMA ให้กับแชนแนลที่มีลำดับความสำคัญสูงกว่าก่อนแล้วจึงทำการขอ DMA ให้กับแชนแนลที่มีลำดับความต่ำกว่า ภายใน ROM BIOSของ IBM/PC จะโปรแกรม 8237A-5 ให้ DRQ1 มีลำดับความสำคัญสูงสุดและ DRQ3 มีลำดับความสำคัญต่ำสุด ดังนั้นถ้ามีการขอ DMA ของอุปกรณ์ภายนอกผ่านทางแชนแนลที่ (DRQ1) และแชนแนลที่ 2 (DRQ2) 8237A-5 ก็จะทำการขอ DMA ให้กับแชนแนลที่ 1 ก่อน จากนั้นเมื่อเสร็จจากขบวนการ DMA ของแชนแนลที่ 1 แล้วจึงจะทำการขอ DMA ให้กับแชนแนลที่ 2

อย่างไรก็ตาม 8237A-5 ยังมีแชนแนลสำหรับการขอ DMAอยู่อีก 1 แชนแนล คือแชนแนลที่ 0 (DRQ0) ซึ่งในความเป็นจริงแล้วแชนแนลนี้จะมีลำดับความสำคัญที่สูงกว่าแชนแนลที่ 1 แต่จะไม่ถูกต่อออกมายังขาของสล็อต เนื่องจาก IBM/PC จะใช้แชนแนลที่ 0 นี้ในการรีเฟรชหน่วยความจำที่เป็น Dynamic RAM

ในการขอ DMA นั้นสัญญาณ DRQ นี้ จะต้องแอคทีฟอยู่ในช่วงระยะเวลาหนึ่งเท่านั้น ถ้าสัญญาณนี้แอคทีฟอยู่นานเกินไป จะทำให้เกิดขบวนการ DMA ขึ้นมากกว่า 1 ขบวนการได้สำหรับวงจรที่ขอ DMA โดยทั่วไปแล้วจะใช้สัญญาณตอบรับการขอ DMA หรือสัญญาณ $\overline{\text{DACK}}$ ของแชนแนลที่ขอ DMA นั้น ในการรีเซ็ทสัญญาณ DRQ เช่นอุปกรณ์ภายนอกที่ขอ DMA ผ่านทางแชนแนลที่ 1 (DRQ1) ก็จะคอยตรวจสอบการตอบรับการขอ DMA จากสัญญาณ $\overline{\text{DACK}}$ ของแชนแนลที่ 1 ($\overline{\text{DACK1}}$) เมื่อได้รับสัญญาณจาก $\overline{\text{DACK1}}$ แล้ว ก็จะรีเซ็ทสัญญาณ DRQ1 (เปลี่ยนจากลอจิก "1" เป็น "0")

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.1.15 $\overline{\text{DACK0}}$-$\overline{\text{DACK3}}$ (DMA Acknowleadge 0-3; ขา B19, B17, B26 และ B15) :

สัญญาณทั้ง 4 นี้เป็นเอาท์พุทแอคทีฟที่ลอจิก "0" ซึ่ง 8237A-5 สร้างขึ้นเพื่อแสดงให้วงจรภายนอกที่ขอ DMA ทราบว่าการขอ DMA นั้นได้รับการตอบสนองแล้ว และ 8237A-5 จะเข้าสู่ขบวนการ DMAเพื่อให้การส่งผ่านข้อมูลระหว่าง อุปกรณ์ I/Oที่ขอ DMA กับหน่วยความจำเกิดขึ้นได้โดยตรง(คือไม่ต้องผ่าน 8088)โดยสัญญาณ DACK นี้จะแอคทีฟในแชนแนลใดก็ขึ้นอยู่กับว่าขบวนการ DMA ที่จะเกิดขึ้นนั้น เป็นการตอบสนองต่อการขอ DMA ในแชนแนลใด เช่นถ้าขบวนการ DMA ที่จะเกิดขึ้นนั้นเป็นการตอบสนองต่อการขอ DMA ในแชนแนลที่ 2 (DRQ2) สัญญาณ $\overline{\text{DACK2}}$ ก็จะแอคทีฟเป็นต้น

ดังที่ได้กล่าวแล้วว่าสัญญาณ DRQ0 นั้นจะไม่ถูกต่อออกมายังขาของสล็อตดังนั้นวงจรอินเทอร์เฟสจึงไม่สามารถจะขอ DMA ผ่านทางแชนแนล 0ได้แต่สัญญาณ $\overline{\text{DACK0}}$ จะถูกต่อออกมายังสล็อตด้วย(ขา B19)ทั้งนี้ก็เพื่อที่จะแสดงให้วงจรอินเทอร์เฟสต่างๆ ทราบว่า ขบวนการ DMA ที่เกิดขึ้นในช่วงเวลาที่ $\overline{\text{DACK0}}$ แอคทีฟนั้น เป็นขบวนการที่ใช้สำหรับการริเฟรชหน่วยความจำที่เป็น Dynamic RAM ซึ่งวงจรอินเทอร์เฟสที่ใช้หน่วยความจำประเภทนี้สามารถจะนำไปใช้ในการรีเฟรช Dynamic RAM ที่อยู่ในวงจรได้

โดยที่การรีเฟรชหน่วยความจำนั้นจะเกิดขึ้นในทุกๆ 15.12 usec. หรือทุกๆ 72 คล็อก ดังนั้นสัญญาณ $\overline{\text{DACK0}}$ นี้ก็จะแอคทีฟในทุกๆ 15.12 usec. ด้วย

2.3.1.16 AEN (Address Enable; ขา A11)

สัญญาณนี้ เป็นเอาท์พุทที่ใช้ในการแสดงว่า บัสไซเคิลที่เกิดขึ้น ในช่วงเวลาที่สัญญาณ AEN แอคทีฟ (ลอจิก "1") นั้น เป็นบัสไซเคิลของขบวนการ DMA

สำหรับบนเมนบอร์ดของ IBM/PC นั้น จะใช้สัญญาณนี้ ในการดิสเอเบิล (Disable) 8288 Bus Controller และจะใช้ดิสเอเบิลพอร์ท I/O ต่างๆ ที่ไม่เกี่ยวข้องกับขบวนการ DMA ที่เกิดขึ้นนี้ที่จำเป็นต้องทำเช่นนี้ก็เพราะในระหว่างขบวนการ DMA นั้น 8237A-5 จะส่งแอดเดรสของหน่วยความจำออกมาบนบัสแอดเดรส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และจะทำให้สัญญาณ $\overline{IOR}$ หรือ $\overline{IOW}$ แอคทีฟด้วย ดังนั้นถ้าไม่ทำการดิสเอเบิลพอร์ท I/O ที่ไม่เกี่ยวข้องไว้ ก็อาจจะทำให้พอร์ท I/O ที่มีแอดเดรสตรงกับค่าแอดเดรสบนบัสแอดเดรส (ซึ่งเป็นแอดเดรสของหน่วยความจำ) นั้นทำการอ่านหรือส่งข้อมูลออกมาบนบัสข้อมูลทำให้เกิดความผิดพลาดขึ้นได้

2.3.1.17 T/C (Terminal Count; ขา B27) :

สัญญาณนี้ถูกสร้างขึ้นจากการนำเอาสัญญาณเอาท์พุทที่ขา EOP ของ 8237A มากลับลอจิก (โดยใช้เกท Inverter) ทำให้สัญญาณ T/C นี้แอคทีฟที่ลอจิก "1"

สำหรับสัญญาณนี้จะแอคทีฟ เมื่อจำนวนไบท์ในการส่งผ่านข้อมูลของขบวนการ DMA ในแชนแนลใดแชนแนลหนึ่ง ครบตามจำนวนที่กำหนดไว้ โดยทั่วไปแล้วสัญญาณที่จะถูกใช้ในการสิ้นสุดขบวนการ DMA ที่ทำการส่งผ่านข้อมูลเป็นบล็อกเนื่องจากสัญญาณนี้จะแอคทีฟโดยไม่แสดงว่าเป็นสัญญาณของแชนแนลใด ดังนั้นจึงต้องทำการนำสัญญาณ T/C นี้ผ่านเกท Inverterแล้วนำไป OR กับสัญญาณ DACK เพื่อให้สามารถทราบได้ว่า สัญญาณ T/C ที่เกิดขึ้นนั้นเป็นสัญญาณของแชนแนลใด สำหรับในแชนแนลที่ 0 นั้น สัญญาณ T/C จะแอคทีฟในช่วงเวลาที่คงที่คือ ทุกๆ 990.804 millisec. ซึ่งก็คือช่วงเวลาที่ใช้ในการรีเฟรชหน่วยความจำขนาด 64 Kbyte นั่นเอง

2.3.2 บัสของแหล่งจ่ายไฟของระบบ

+5Vdc (ขา B3 และ B29) :

ขาทั้งสองนี้ต่อกับแหล่งจ่ายไฟ DC +5V ของระบบ โดยจะมีค่าความเที่ยงตรง(Regulated) $\pm$ 5% คืออยู่ในช่วง +4.75 ถึง +5.25 Vdc

+12Vdc (ขา B9) :

ขานี้จะต่อกับแหล่งจ่ายไฟ DC +12V ของระบบ โดยจะมีค่าความเที่ยงตรง (Regulated) $\pm$ 5% คืออยู่ในช่วง +11.4 ถึง +12.6 Vdc

-5Vdc (ขา B5) :

ขานี้จะต่อกับแหล่งจ่ายไฟ DC -5V ของระบบ โดยจะมีค่าความเที่ยงตรง (Regulated) $\pm$ 10% คืออยู่ในช่วง -5.5 ถึง -4.5 Vdc

-12Vdc (ขา B7) :

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขานี้จะต่อกับแหล่งจ่ายไฟ DC -12V ของระบบ โดยจะมีค่าความเที่ยงตรง (Regulated) $\pm$ 10% คืออยู่ในช่วง -13.2 ถึง -10.8 Vdc

GND (ขา B1, B10 และ B31) :

ขาทั้งสามนี้จะต่อเข้ากับกราวด์ (Ground) ของระบบ

2.3.3 การจัดสัญญาณบนสล็อตของ IBM PC/XT

สำหรับใน IBM PC/XT นั้นจะมีสล็อตสำหรับเชื่อมต่อกับวงจรภายนอกได้มากขึ้น คือใน IBM PC/XT จะทำการเพิ่มจำนวนสล็อตบนเมนบอร์ดขึ้นเป็น 8 สล็อตจากเดิมที่มีอยู่เพียง 5 สล็อตบน IBM PC โดยการจัดสัญญาณต่างๆ ในทั้ง 8 สล็อตจะยังคงเหมือนกับใน IBM PC เพียงแต่สัญญาณต่างๆ ที่จะถูกส่งออกมายังขาของสล็อตที่ 8 นั้น จะถูกต่อผ่านวงจรขับกระแส (Buffer) ก่อนและในสล็อตที่ 8 นี้ ขา B8 จะถูกใช้งานด้วย โดยจะถูกใช้เป็นขา $\overline{\text{CARD SLCTD}}$ (หรือ Card Selected) ซึ่งขาสัญญาณนี้จะเป็นสัญญาณอินพุทจากวงจรภายนอกที่เสียบอยู่บนสล็อตที่ 8 เพื่อให้วงจรบนเมนบอร์ดทราบว่า คาร์ดที่อยู่บนสล็อตนี้ถูกเลือกใช้งานอยู่ ซึ่งจะทำให้ Driver บนเมนบอร์ดทำการอ่านหรือส่งข้อมูลไปยังสล็อตที่ 8

2.4 การจัดแอดเดรสสำหรับหน่วยความจำและ I/O

ในหัวข้อนี้จะกล่าวถึงการจัดแอดเดรส สำหรับหน่วยความจำและพอร์ท I/O ต่างๆ ภายใน IBM/PC ซึ่งจะแสดงถึงแอดเดรสต่างๆ ที่ถูกใช้งานโดยพอร์ท I/O (Input/Output Port) และหน่วยความจำ นอกจากนี้จะได้กล่าวถึงเทคนิคการดิโค้ด (decode) แอดเดรสในแบบต่างๆ ด้วย

2.4.1 การจัดแอดเดรสสำหรับพอร์ท I/O ใน IBM/PC

ในหัวข้อนี้จะกล่าวถึงวิธีการอ้างและใช้งานแอดเดรสต่างๆ ของพอร์ท I/O ที่ใช้งานอยู่ใน IBM/PC

2.4.1.1 การอ้างแอดเดรสของพอร์ท I/O

ในการควบคุม และตรวจสอบสภาวะการทำงาน รวมทั้งการอ่านข้อมูลจากอุปกรณ์ที่เป็นชิพซัพพอร์ทหรือคาร์ดต่างๆ ที่ใช้ในระบบของ IBM/PC นั้นจะกระทำโดย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านทางพอร์ท I/O ของระบบ ดังนั้นในการที่จะใช้งาน หรือควบคุมการทำงานของอุปกรณ์เหล่านี้ จึงจำเป็นต้องศึกษาถึงวิธีการควบคุมพอร์ท I/O ต่างๆ ของระบบด้วย และเนื่องจากการควบคุมหรือติดต่อกับพอร์ทเหล่านี้ ต้องทำโดยการอ้างถึง แอดเดรสของพอร์ท I/O เหล่านั้นโดยตรง เราจึงจำเป็นต้องศึกษา ถึงหลักการอ้างแอดเดรสของ 8088 ใน IBM/PC ด้วย

สำหรับแอดเดรสของพอร์ท I/O ต่างๆ นั้นจะเป็นแอดเดรสที่ถูกสร้างขึ้นโดย 8088 ซึ่งแอดเดรสเหล่านี้เป็นแอดเดรสที่จัดไว้สำหรับพอร์ท I/O โดยเฉพาะคือแยกจากแอดเดรสของหน่วยความจำโดยเด็ดขาดส่วนการส่งข้อมูลให้กับพอร์ทเหล่านี้จะทำได้โดยการใช้คำสั่ง OUT ของ 8088 ส่งข้อมูลนั้นไปยังแอดเดรสของพอร์ทที่ต้องการ และสำหรับการตรวจสอบหรือการอ่านข้อมูลขสกพอร์ท ก็จะทำได้โดยการใช้คำสั่ง IN ของ 8088 อ่านข้อมูลจากแอดเดรสของพอร์ทที่ต้องการเช่นกัน

ภายในไมโครโปรเซสเซอร์ เบอร์ 8088 นี้จะมีแอดเดรสสำหรับใช้กับพอร์ท I/O อยู่ทั้งสิ้น 65,536 หรือ 64K แอดเดรส(ในขณะที่มีแอดเดรสสำหรับหน่วยความจำอยู่ 1 Mbyte) ซึ่งทำให้การอ้างแอดเดรสของพอร์ I/O ที่ทำงานร่วมกับ 8088 นั้นต้องใช้จำนวนเส้นแอดเดรสในบัสแอดเดรสทั้งสิ้น 16 เส้นคือ A0-A15 แต่สำหรับใน IBM/PC นี้ถูกออกแบบมาให้ใช้เส้นแอดเดรสเฉพาะ 10 เส้นล่าง คือ A0-A9 เท่านั้นดังนั้นในการอ้างถึงแอดเดรสของพอร์ทของอุปกรณ์หรือชิพซัพพอร์ทใดๆที่ใช้ร่วมกับ IBM/PC จึงใช้จำนวนเส้นแอดเดรสเพียง 10 เส้นด้วยโดยเส้นแอดเดรสที่เหลือคือ A10-A15 นั้นจะไม่ถูกนำไปใช้งานอย่างไรก็ตาม แม้ว่าเส้นแอดเดรส A10-A15 นี้จะไม่ถูกนำไปใช้งาน แต่ค่าแอดเดรสบนเส้นแอดเดรสเหล่านี้ยังคงเปลี่ยนแปลงตามค่าแอดเดรสของพอร์ทที่กำหนดไว้ในคำสั่ง OUT หรือ IN อยู่ด้วยเพียงแต่ไม่ได้ถูกนำมาดีโค้ดร่วมกับแอดเดรส A0-A9 เท่านั้นตัวอย่างเช่นในการใช้คำสั่ง OUT ส่งข้อมูลไปยังพอร์ทที่ตรงกับแอดเดรส 0010H นั้นจะให้ผลเหมือนกับการส่งข้อมูลไปยังพอร์ทที่ตรงกับแอดเดรส 0410H, 0810H, 0C10H ทั้งนี้เนื่องจากแอดเดรส 6 บิทบนไม่ได้ถูกใช้งานจึงทำให้การเปลี่ยนแปลงค่าแอดเดรสบนเส้นแอดเดรส A10-A15 นั้นไม่ทำให้เกิดความแตกต่างใดๆขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

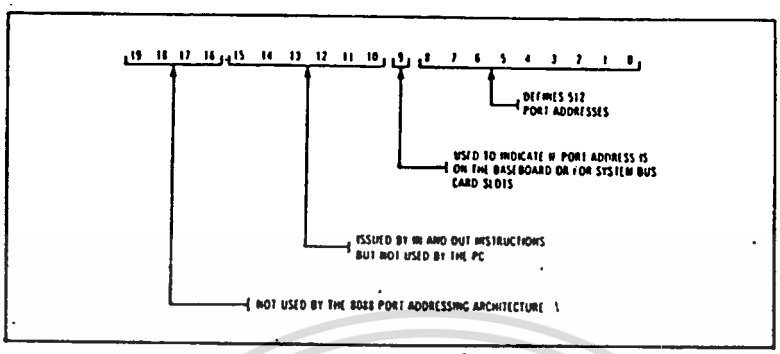
เนื่องจากใน IBM/PC ได้ใช้งานเส้นแอดเดรสเพียง 10 เส้น (คือ A0-A9) ดังนั้นจึงสามารถที่จะอ้างแอดเดรสของพอร์ทได้สูงสุดเพียง 1024 พอร์ท(จากจำนวน 64K พอร์ท) เท่านั้นนอกจากนี้ในกรณีที่เป็นการอ่านข้อมูลจากพอร์ทของ IBM/PC ข้อมูลในบิท A9 จะถูกจัดให้มีหน้าที่ในการแบ่งพอร์ททั้ง 1024 พอร์ทออกเป็น 2 ส่วน (ส่วนละ 512 พอร์ท) อีกดวัยกล่าวคือถ้าข้อมูลในบิท A9 เป็น "0" แล้ว เราจะทำการอ่านข้อมูลได้เฉพาะจากพอร์ทของอุปกรณ์ หรือชิพซัพพอร์ทต่างๆที่อยู่บนเมนบอร์ด (Main Board) ของ IBM/PC เช่น 8253-5, 8237-5 หรือ 8259A เท่านั้น แต่ถ้าข้อมูลในบิท A9 นี้เป็น "1" ก็จะทำการอ่านข้อมูลได้เฉพาะจากพอร์ทที่อยู่บนการ์ดต่างๆเท่านั้น

จากที่ได้กล่าวมานั้นจะสรุปได้ว่าพอร์ทบน IBM/PC ทั้ง 1024 พอร์ทถูกแบ่งออกเป็น 2 กลุ่ม โดยที่กลุ่มแรกเป็นกลุ่มของพอร์ทที่อยู่บนเมนบอร์ดและกลุ่มที่สองเป็นกลุ่มที่จัดเตรียมไว้สำหรับพอร์ทที่อยู่บนการ์ดต่างๆ

สำหรับในกรณีของการส่งข้อมูลให้กับพอร์ททั้ง 1024 พอร์ท เราสามารถที่จะเลือกส่งไปยังพอร์ทใดๆใน IBM/PCได้ ดังนั้นการเลือกแอดเดรสสำหรับพอร์ทที่อยู่บนการ์ดจึงสามารถทำได้โดยสะดวก แต่อย่างไรก็ตามสิ่งหนึ่งที่จะต้องคำนึงถึงก็คือ ถ้าแอดเดรสที่เราเลือกให้กับพอร์ทนี้ ตรงกับค่าแอดเดรสเดิมที่มีอยู่บนเมนบอร์ดแล้วเมื่อเราทำการส่งข้อมูลให้กับพอร์ทที่อยู่ในตำแหน่งแอดเดรสนี้ก็จะเท่ากับเป็นการส่งข้อมูลให้กับทั้งพอร์ทที่อยู่บนเมนบอร์ดและพอร์ทที่อยู่บนการ์ดด้วย ซึ่งในกรณีเช่นนี้ อาจจะก่อให้เกิดความผิดพลาดขึ้นได้เช่นกัน ดังนั้นในการกำหนดค่าแอดเดรส ให้กับพอร์ทที่ถูกสร้างขึ้นบนการ์ดต่างๆ จึงควรจะใช้ค่าแอดเดรสที่แอดเดรสบิท A9 มีค่าเป็น "1" คือแอดเดรส OFE00H จนถึง OFFFFH เท่านั้น (แอดเดรสบิท A10-A15 ไม่ถูกใช้ในการดีโค้ด แต่เพื่อความสะดวก จึงกำหนดให้มีค่าเป็น "1" ในฐานสองทั้งหมดแต่ในการใช้งานจริง อาจเปลี่ยนให้แอดเดรส A10-A15 แต่ละบิทมีค่าเป็น "1" หรือ "0" ก็ได้)

สำหรับรูปที่ 2.9 นี้ จะแสดงถึงการใช้งานแอดเดรสบิทต่างๆในการอ้างแอดเดรสของพอร์ทใน IBM/PC

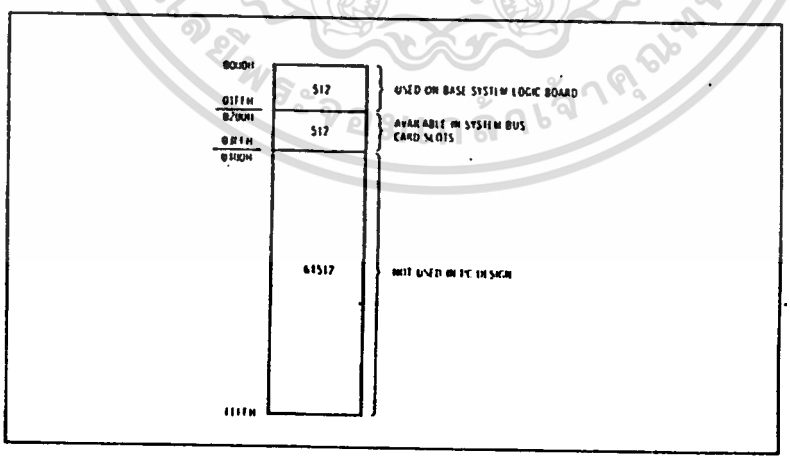
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 การใช้แอดเดรสบิทต่างๆในการอ้างแอดเดรสของพอร์ทใน IBM/PC

2.4.1.2 การใช้งานแอดเดรสสำหรับพอร์ท I/O ใน IBM/PC

จากที่ได้กล่าวไว้ในหัวข้อที่ผ่านมานั้น พอร์ท I/O ทั้ง 1024 พอร์ทใน IBM/PC จะถูกแบ่งออกเป็น 2 กลุ่มๆละ 512 พอร์ท สำหรับในหัวข้อนี้จะกล่าวถึงการใช้งานพอร์ทต่างๆเหล่านี้โดยจะแบ่งออกเป็น 2 กลุ่มตามที่ได้อธิบายไว้ในหัวข้อที่ผ่านมาดังนี้

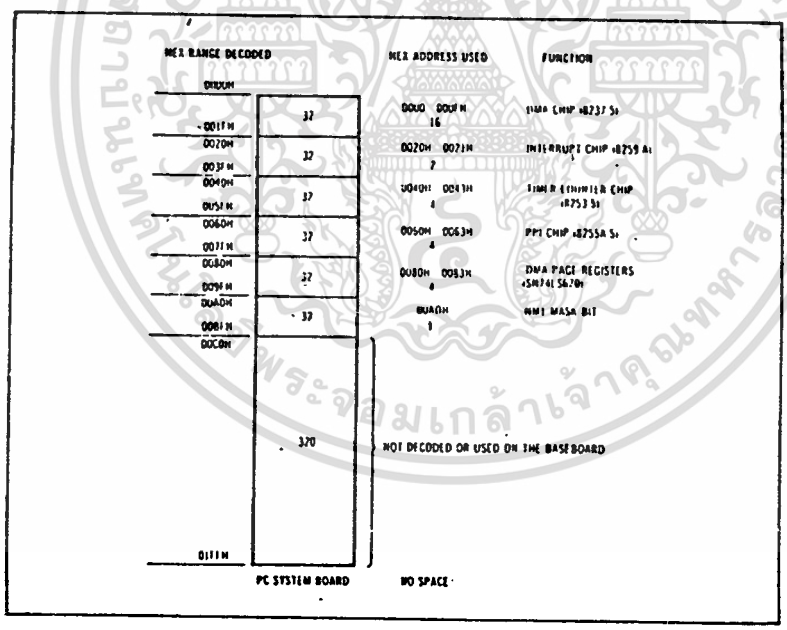


รูปที่ 2.10 การใช้งานแอดเดรสของพอร์ทบน IBM/PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในกลุ่มแรกนี้เป็นกลุ่มของพอร์ท I/O ที่อยู่บนเมนบอร์ดของIBM/PC ซึ่งจะมีแอดเดรสอยู่ในตำแหน่ง 0000H จนถึง 01FFH (ขอให้ระลึกอยู่เสมอว่า A10-A15 นั้นไม่ถูกใช้งาน) หรือ แอดเดรสที่มีบิท A9 เป็น "0" นั่นเอง

สำหรับแอดเดรสของพอร์ท I/O ในกลุ่มนี้จะถูกใช้ในการอ้างแอดเดรสของชิพซัพพอร์ท และอุปกรณ์ที่เป็น I/O ต่างๆบนเมนบอร์ดของ IBM/PC เช่นแอดเดรส 0000H จนถึง 000FH จะถูกใช้เป็นแอดเดรสสำหรับ 8237-5 DMA Controller เป็นต้น ในรูปที่ 2.11 จะแสดงถึงการใช้งานแอดเดรสต่างๆตั้งแต่ 0000H จนถึง 01FFH ในการอ้างแอดเดรสของชิพซัพพอร์ท และอุปกรณ์ต่างๆที่ทำหน้าที่เป็น I/O บนเมนบอร์ดของ IBM/PC



รูปที่ 2.11 การใช้งานแอดเดรสต่างๆสำหรับพอร์ท I/O ของ IBM/PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปข้างต้นจะเห็นว่าแอดเดรส 00C0H จนถึงแอดเดรส 01FFH นั้นไม่ได้ถูกใช้งานบนเมนบอร์ดของ IBM/PC ดังนั้นในกรณีนี้ก็จึงสามารถที่จะใช้งานแอดเดรสต่างๆเหล่านี้ได้ แต่อย่างไรก็ตามแอดเดรสเหล่านี้ยังคงถูกดีโค้ดให้เป็นแอดเดรสที่ใช้ในการอ่านข้อมูลจากพอร์ท I/O บนเมนบอร์ดเท่านั้น ดังนั้น การใช้ค่าแอดเดรส 00C0H-01FFH กับพอร์ท I/O บนการ์ดหรือวงจรอินเทอร์เฟสที่เราสร้างขึ้นนั้น ต้องเป็นพอร์ทเอาท์พุทพอร์ทเพียงชนิดเดียวเท่านั้น กล่าวคือจะทำการอ่านข้อมูลจากพอร์ท I/O (ที่ไม่ได้อยู่บนเมนบอร์ด) ที่มีค่าแอดเดรสอยู่ในช่วง 00C0H-01FFH ไม่ได้

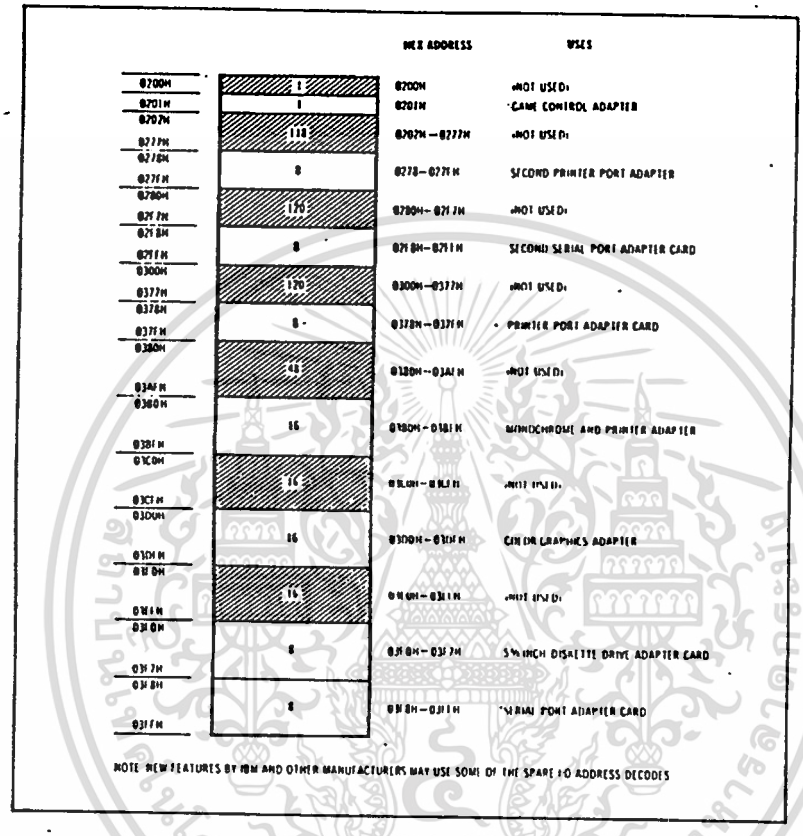
ในกลุ่มที่สองนี้จะเป็นกลุ่มของพอร์ท I/O ที่ถูกใช้งานอยู่บนการ์ดที่ใช้เสียบบนสล็อตต่างๆของ IBM/PC สำหรับแอดเดรสของพอร์ทเหล่านี้จะเริ่มต้นจากแอดเดรส 0200H จนถึง 03FFH ซึ่งก็คือแอดเดรสที่มีบิท A9 เป็น "1" นั่นเอง สำหรับการใช้งานแอดเดรสของพอร์ท I/O ในกลุ่มนี้จะแสดงได้ดังรูป 2.12

อย่างไรก็ตามการใช้งานแอดเดรสกลุ่มนี้อาจจะเปลี่ยนแปลงได้ ทั้งนี้ขึ้นอยู่กับการใช้งานการ์ดต่างๆร่วมกับ IBM/PC โดยการ์ดที่ถูกออกแบบผลิตขึ้นใหม่นั้นอาจใช้ค่าแอดเดรสต่างๆที่เหลืออยู่นี้ได้ ดังนั้นก่อนที่จะทำการออกแบบวงจรอินเทอร์เฟสที่จำเป็นต้องใช้ค่าแอดเดรสสำหรับพอร์ท I/O จึงควรจะตรวจสอบดูก่อนว่าการ์ดต่างๆ ที่ใช้อยู่ในระบบของ IBM/PC ที่เราใช้งานอยู่นั้นมีการ์ดใดบ้าง และการ์ดเหล่านั้นใช้งานแอดเดรสใดบ้าง จากนั้นจึงทำการออกแบบวงจรอินเทอร์เฟสโดยเลือกใช้เฉพาะแอดเดรสที่ยังไม่ถูกใช้งาน

2.4.2 เทคนิคในการดีโค้ดแอดแดรสสำหรับพอร์ท I/O

ในหัวข้อต่างๆที่ผ่านมาข้างต้นนั้นได้กล่าวถึงการอ้างแอดเดรสและการใช้งานแอดเดรสต่างๆของพอร์ท I/O ใน IBM/PC สำหรับในหัวข้อนี้จะกล่าวถึงวิธีการต่างๆที่ใช้ในการดีโค้ดแอดเดรสต่างๆให้เป็นไปตามที่เราต้องการ

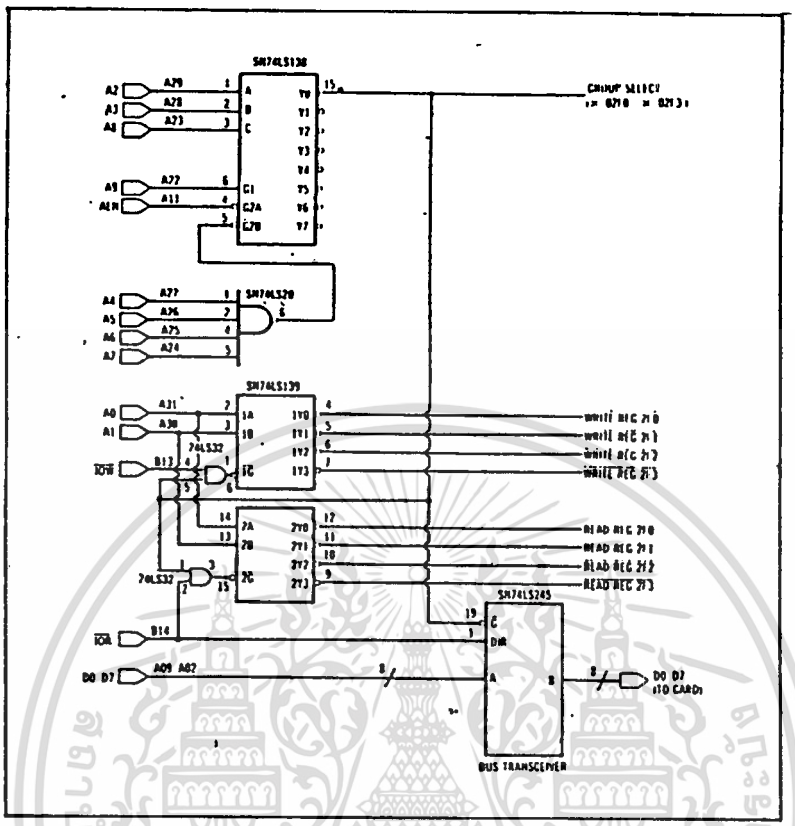
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 การใช้งานแอดเดรสสำหรับพอร์ท I/O บนการ์ดต่างๆ

การดีโค้ดแบบ Fixed วิธีการดีโค้ดแบบนี้เป็นวิธีการที่ง่ายและสะดวกในการดีโค้ดแอดเดรสหรือกลุ่มของแอดเดรสของพอร์ท I/O ซึ่งวิธีนี้เป็นการกำหนดจำนวนของแอดเดรสที่เราต้องการใช้จากนั้นจึงทำการเลือกบล็อกของแอดเดรสที่ยังไม่ถูกใช้งานโดยการ์ดหรือวงจรอินเทอร์เฟสอื่นๆ (บล็อกของแอดเดรสที่เลือก ต้องมีจำนวนแอดเดรสเพียงพอกับจำนวนแอดเดรสที่เราต้องการใช้งาน) แล้วจึงออกแบบวงจรที่ทำการดีโค้ดแอดเดรสที่เราต้องการ สำหรับตัวอย่างวงจรที่ใช้ในการดีโค้ดแอดเดรสในแบบนี้จะแสดงได้ดังรูป 2.13

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 ตัวอย่างวงจรดีโค้ดแอดเดรสแบบ Fixed

จากรูปจะเห็นได้ว่า วงจรที่ใช้นี้เป็นวงจรที่สามารถทำการดีโค้ดแอดเดรสได้ 8 กลุ่ม โดยแต่ละกลุ่มจะมีจำนวนแอดเดรส 4 แอดเดรส ซึ่งแอดเดรสทั้ง 8 กลุ่มจะแสดงได้ดังตารางข้างล่าง

กลุ่ม	แอดเดรส
0 (Y0)	02F0H-02F3H
1 (Y1)	02F4H-02F7H
2 (Y2)	02F8H-02FBH
3 (Y3)	02FCH-02FFH
4 (Y4)	03F0H-03F3H
5 (Y5)	03F4H-03F7H
6 (Y6)	03F8H-03FBH
7 (Y7)	03FCH-03FFH

ตารางที่ 2.1 แสดงการดีโค้ดแอดเดรส 8 กลุ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับในตัวอย่างนี้จะเลือกใช้การดีโค้ดแอดเดรสในกลุ่ม 0 (เริ่มจากแอดเดรส 02F0H จนถึง 02F3H) คือ ใช้สัญญาณเอาท์พุท (สัญญาณ $\overline{\text{GROUPSELECT}}$) จากขา Y0 (ขา15) ของ 74LS138 ไปทำการ OR กับสัญญาณ $\overline{\text{IOR}}$ และ $\overline{\text{IOW}}$ เพื่อสร้างเป็นสัญญาณอีนาเบิลวงจรดีโค้ด (74LS139) แอดเดรสอีก 4 แอดเดรส ซึ่งแบ่งเป็น 2 ชุด คือ ชุดที่เป็น $\overline{\text{WRITE REG}}$ ซึ่งจะแอคตีฟ(ลอจิก "0") เมื่อ CPU ต้องการจะส่งข้อมูลให้กับวงจรภายนอก(สัญญาณ $\overline{\text{IOW}}$ แอคตีฟ) และ ชุดที่เป็น $\overline{\text{READ REG}}$ ซึ่งจะแอคตีฟ เมื่อ CPU ต้องการจะอ่านข้อมูลจากวงจรภายนอก (สัญญาณ $\overline{\text{IOR}}$ แอคตีฟ) สัญญาณ $\overline{\text{WRITE REG}}$ และ $\overline{\text{READ REG}}$ นี้ โดยทั่วไปจะนำไปเป็นสัญญาณสโตรป(Strobe)ให้กับวงจรภายนอกที่เกี่ยวข้อง เพื่อให้สามารถส่งหรือรับข้อมูลจาก CPU ได้ในช่วงเวลาที่เหมาะสม นอกจากนี้สัญญาณ $\overline{\text{GROUPSELECT}}$ ยังถูกนำไปใช้ในการอีนาเบิล บัฟเฟอร์ 74LS245 ด้วย เพื่อให้ CPU สามารถส่งหรือรับข้อมูลจากอุปกรณ์ภายนอก ได้เมื่อแอดเดรสในกลุ่มนี้ถูกเลือก สำหรับทิศทางของข้อมูลจะถูกควบคุมโดยสัญญาณ $\overline{\text{IOR}}$ ส่วนสัญญาณ AEN จะถูกนำมาใช้ในการดีสเอเบิลวงจรดีโค้ดโดย ถ้าสัญญาณ AEN เป็น "1" ซึ่งเป็นช่วงเวลาของขบวนการ DMA นั้น 74LS138 จะถูกดีสเอเบิล ทันที ทั้งนี้ก็เพื่อป้องกันความผิดพลาดที่อาจเกิดขึ้น เนื่องจากการดีโค้ดแอดเดรสของพอร์ทในระหว่างขบวนการ DMA นั่นเอง (ในระหว่างนี้แอดเดรสบนบัสแอดเดรสจะเป็นแอดเดรสของหน่วยความจำคือสัญญาณ $\overline{\text{MEMW}}$ หรือ $\overline{\text{MEMR}}$ จะแอคตีฟ แต่ในขณะเดียวกันสัญญาณ$\overline{\text{IOR}}$ หรือ $\overline{\text{IOW}}$ ก็จะแอคตีฟด้วย ดังนั้นถ้าไม่ดีสเอเบิลวงจรดีโค้ดไว้แล้ว อาจทำให้วงจรดีโค้ดคิดว่าแอดเดรสบนบัสแอดเดรส เป็นแอดเดรสของพอร์ท I/O ก็ได้)

ในการดีโค้ดแอดเดรสของพอร์ท I/O เราจะต้องคำนึงถึงช่วงเวลา ของสัญญาณที่เกิดขึ้นในขบวนการอ่านหรือเขียนข้อมูลลงบนพอร์ท I/O ดังนี้

2.4.2.1 ในช่วงเริ่มต้นของบัสไซเคิลที่เกี่ยวกับพอร์ท I/O นั้น ถ้าสัญญาณจากวงจรดีโค้ดมีการหน่วงเวลา (Delay) มากเกินไป อาจจะทำให้สัญญาณดีโค้ดนี้เกิดขึ้นหลังจากที่สัญญาณ $\overline{\text{IOR}}$ หรือ $\overline{\text{IOW}}$ แอคตีฟ และเนื่องจากค่าแอดเดรสบนบัสแอดเดรสนั้นเปลี่ยนแปลงได้ตลอดเวลา ดังนั้นก่อนที่ค่าแอดเดรสที่ถูกต้องจะถูกส่งออก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาบนบัสแอดเดรสนั้น วงจรดีโค้ดจะได้รับค่าแอดเดรสอื่นๆอยู่ ซึ่งถ้าหากวงจรดีโค้ดมีการหน่วงเวลามากเกินไปแล้ว สัญญาณดีโค้ดแอดเดรสที่ไม่ถูกต้องนี้ อาจจะถูกหน่วงเวลาจนเกิดขึ้นในช่วงเวลาที่สัญญาณ $\overline{IOR}$ หรือ $\overline{IOW}$ เกิดขึ้นแล้วก็ได้ทำให้ข้อมูลบนบัสข้อมูลนั้นถูกส่งไปยังพอร์ทที่ไม่ถูกต้องสำหรับในIBM/PC จะถูกออกแบบให้การหน่วงเวลาในวงจรดีโค้ดนั้นมีค่าไม่เกิน 92 nanosec.

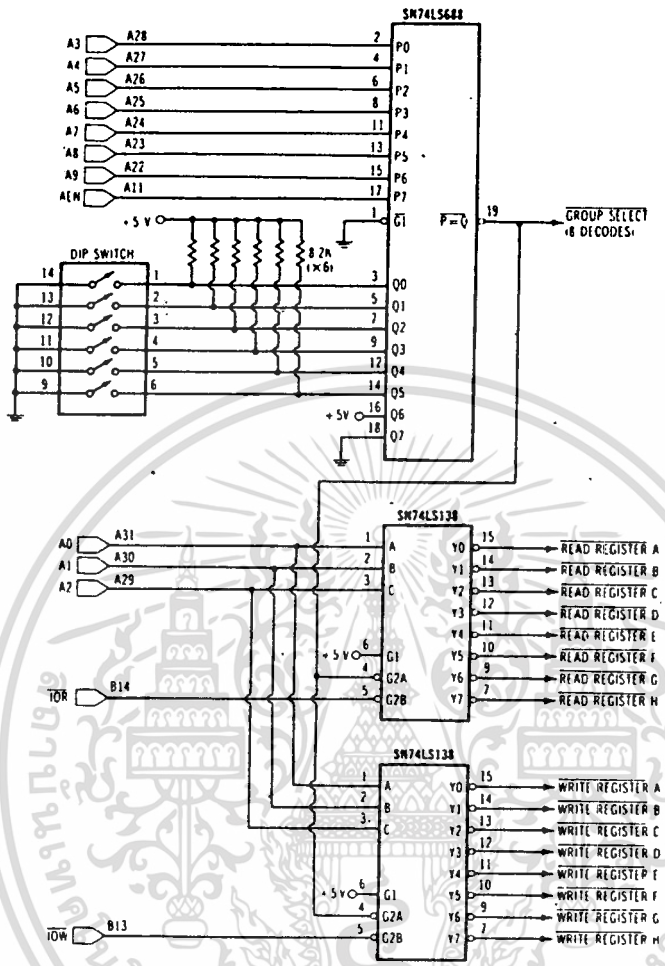
2.4.2.2 ในช่วงท้ายของบัสไซเคิลในการเขียนข้อมูลลงบนพอร์ท I/O นั้น ถ้าสัญญาณ $\overline{IOW}$ มีการหน่วงเวลาออกไป และวงจรดีโค้ดมีความเร็วในการทำงานสูงแล้ว อาจจะทำให้ข้อมูลในบัสไซเคิลนี้ถูกส่งให้กับพอร์ท I/O ที่มีแอดเดรสตรงกับค่าแอดเดรสในบัสไซเคิลต่อไปก็ได้ สำหรับใน IBM/PC สัญญาณ $\overline{IOW}$ จะมีการหน่วงเวลาไปไม่เกิน 200 nanosec.

อย่างไรก็ตาม ช่วงเวลาที่ต้องสนใจมากอีกช่วงเวลาหนึ่ง ก็คือ ช่วงเวลาระหว่างขอบขาขึ้นของสัญญาณ $\overline{IOW}$ กับช่วงเวลาที่ข้อมูลที่ถูกต้องถูกส่งออกมาบนบัสข้อมูล ถ้าสัญญาณ $\overline{IOW}$ ถูกหน่วงเวลาไปเกินกว่า 120 nanosec. แล้ว อาจจะทำให้พอร์ท I/O ได้รับข้อมูลที่ไม่ถูกต้องก็ได้ และสำหรับสัญญาณ $\overline{IOR}$ นั้น ถ้ามีการหน่วงเวลาเกิดขึ้นแล้ว ก็จะทำให้ความเร็วในการอ่านข้อมูลถูกลดลง

2.4.3 การดีโค้ดโดยใช้สวิทซ์เลือก

การดีโค้ดในแบบ Fixed ที่ได้กล่าวไว้ในหัวข้อที่ผ่านมานั้น มีข้อเสียอยู่บางประการคือแอดเดรสที่เราเลือกใช้งานไว้นั้นอาจจะซ้ำกับแอดเดรสของการ์ดอื่นที่เรานำมาเพิ่มเข้าไปในระบบในภายหลังก็ได้ ซึ่งกรณีเช่นนี้ เราต้องแก้ไขวงจรเพื่อหลีกเลี่ยงไปใช้แอดเดรสอื่นที่ยังว่างอยู่ และไม่ถูกใช้งานโดยการ์ดที่จะเพิ่มเข้าไปใหม่ซึ่งยุ่งยากและต้องเสียเวลามากขึ้น ปัญหาเช่นนี้เราสามารถแก้ไขได้ โดยใช้วงจรดีโค้ดที่สามารถเปลี่ยนแปลงค่าแอดเดรสได้ โดยเพียงแต่เปลี่ยนตำแหน่งของสวิทซ์ (ในที่นี้คือ DIP Switch) ที่เซ็ทไว้ในวงจรเท่านั้น ดังรูปที่ 2.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 ตัวอย่างวงจรดีโค้ดโดยใช้สวิทช์เลือก

จากรูป เป็นวงจรที่ทำการดีโค้ดกลุ่มแอดเดรสขนาด 8 แอดเดรส ซึ่งการเลือกกลุ่ม แอดเดรสที่จะทำการดีโค้ดนี้ จะทำได้โดยการเซ็ท DIP Switch ที่ขา Q0-Q5 ของ 74LS688 สำหรับหน้าที่ของ 74LS688 นี้ จะทำการเปรียบเทียบค่าของอินพุท 2 ชุดที่ถูกส่งเข้ามาทางขา P0-P7 และขา Q0-Q7 ถ้าอินพุททั้ง 2 ชุดนี้เท่ากันแล้วเอาท์พุทที่ขา $\overline{P=Q}$ จะให้เอาท์พุทเป็นลอจิก "0" จากในวงจรขา P0-P6 ของ 74LS688 ต่อกับแอดเดรสบิท A3-A9 ในขณะที่ขา Q0-Q5 ต่อกับความต้านทานที่ทำหน้าที่เป็น Pull Up (รักษาระดับแรงดันให้เป็นลอจิก "1" ไว้ ในกรณีที่ไม่มีอินพุทใดๆเข้ามา) และขา Q0-Q5 นี้ จะต่อกับปลายข้างหนึ่งของ DIP Switch นั้นจะต่อลง Ground (ลอจิก "0") ไว้ดังนั้นถ้าเราทำการ "ON" DIP Switch ที่ต่อ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กับขาใดขานั้นจะได้รับลอจิก "0" ในขณะที่ DIP Switch ที่ต่อกับขาใดถูก "OFF" ขานั้นจะได้รับลอจิก "1" และเนื่องจากอินพุทที่ขา P0-P5(แอดเดรส A3-A9) ต้องเท่ากับอินพุทที่ขา Q0-Q5 ดังนั้นถ้าเราเปลี่ยนแปลงการเซ็ท DIP Switch เหล่านี้ก็จะทำให้แอดเดรสบิท A3-A5 ซึ่งต่อกับขา P0-P5 นั้น ต้องเปลี่ยนแปลงตามไปด้วย จึงจะทำให้เอาท์พุทของ 74LS688 แอคตีฟได้ ทำให้เราสามารถเปลี่ยนค่าแอดเดรสที่ต้องการจะดีโค้ดได้ง่ายกว่า วิธีการดีโค้ดแบบ Fixed สำหรับขา Q6 นั้น จะต่อกับลอจิก "1"(+5 V.)และขา P6 ต่อกับแอดเดรสบิท A9 ในกรณีเช่นนี้ จึงเท่ากับเป็นการบังคับ ให้แอดเดรสที่จะทำการดีโค้ดได้นั้น จะต้องมีแอดเดรสบิท A9 เป็น "1" เท่านั้น ส่วนขา P7 จะต่อกับสัญญาณ AEN โดยมีขา Q7 ต่อกับลอจิก "0" การต่อในลักษณะนี้เพื่อป้องกันไม่ให้ 74LS688 ทำการดีโค้ดในระหว่างขบวนการ DMA นั่นเอง เอาท์พุทจากขา $\overline{P=Q}$ ของ 74LS688 นี้จะถูกนำไปใช้ในการอีนาเบิล 74LS138 ซึ่งทำหน้าที่ในการดีโค้ดแอดเดรส 8 แอดเดรสของกลุ่มแอดเดรสที่เราเลือก

วงจรในลักษณะนี้ เราสามารถจะนำไปใช้เป็นการดีโค้ดในแบบ Fixed ได้ โดยการนำเอา DIP Switch ออก จากนั้นถ้าอินพุทใดต้องการลอจิก "0" จึงจะใช้ตัวนำเชื่อมต่อระหว่างขั้วทั้งสอง แทนการเซ็ท DIP Switch ให้ "ON" แต่ถ้าอินพุทใดต้องการลอจิก "1" ก็ปล่อยขั้วทั้งสองนั้นไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบฮาร์ดแวร์ (HARDWARE DESIGN)

อุปกรณ์หน่วยความจำที่ให้ข้อมูลของโปรแกรมอ่านออกมาอย่างเดียว (READ-ONLY MEMORY) ชนิด ทีทีเอล (TTL) จะมีข้อแตกต่างจากอุปกรณ์ชนิด มอส (MOS) ที่เป็น พรอม (P-ROM)· นั่นก็คือข้อมูลที่ได้ถูกโปรแกรมเข้าไปในอุปกรณ์นี้แล้ว จะไม่สามารถที่จะลบออกและเขียนเข้าไปใหม่ได้ มีโครงสร้างภายนอกเป็นแบบ ดิพ (DIP หรือ Dual Inline Package) ข้อดีก็คือจะให้ความเร็วในการทำงานสูง ตัวอย่างการนำไปใช้งานของอุปกรณ์หน่วยความจำชนิดที่มีปริมาณความจำน้อยได้แก่การทำเป็นดีโค้ดเดอร์ (Decoder) เอ็นโค้ดเดอร์ (Encoder) หรือ ทำโค้ดคอนเวอร์เตอร์ (Code Converter) การนำไปใช้งานของอุปกรณ์ หน่วยความจำชนิดที่มีปริมาณการจำมาก ก็ได้แก่อุปกรณ์ความจำของข้อมูลของโปรแกรม ที่ใช้ในไมโครคอมพิวเตอร์

3.1 หลักการโปรแกรม พรอม

การโปรแกรม พรอม นั้นจำเป็นอย่างยิ่งที่เราจะต้องทราบรายละเอียดต่างๆ ของ พรอม เบอร์ที่ต้องการโปรแกรม เช่น คุณสมบัติทางไฟฟ้าของ พรอม ช่วงเวลาสำหรับการโปรแกรม (Timing Diagram For Programing) รายละเอียดต่างๆ เหล่านี้หาดูได้จากคู่มือ (Data Manual) ของ พรอม เบอร์ต่างๆที่บริษัทผู้ผลิตพิมพ์ขึ้นประกอบการใช้งานในที่นี้จะยกตัวอย่างไอซีเบอร์ N82S23/123 ซึ่งเป็นไบโพลาร์ พรอม ขนาด 32 X 8 บิทซึ่งเป็นชนิด (Junction Isolated Schottky PROM) ในการโปรแกรม พรอม เพื่อใช้งานนั้น มีขั้นตอนต่างๆพอสรุปได้ดังนี้คือ

3.1.1 ให้ค่าแอดเดรส (Address) ตำแหน่งที่จะทำการโปรแกรมแก่พรอม

3.1.2 ตรวจสอบสถานะบิทข้อมูล (Data Bit) ที่จะทำการโปรแกรม

3.1.3 ลดระดับแรงดัน Vcc จาก 5 โวลท์ เป็น 0 โวลท์

3.1.4 สำหรับตำแหน่งบิทข้อมูลที่ไม่ต้องการโปรแกรม ให้ข้ามไปทำสเต็ปที่ 3.1.10 ได้เลย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 เพิ่มระดับแรงดัน Vcc จากเดิม 5 โวลท์ เป็นระดับ Vccp ที่แรงดัน 8.75 $\pm$ .25 โวลท์

3.1.6 ให้สถานะที่ขา $\overline{S}$, $\overline{E}$ หรือ $\overline{G}$ เป็น ลอจิก "1" หรือ (Disable)

3.1.7 ป้อนสถานะบิทข้อมูลเข้าที่ขา Data ของ พรอม ด้วยระดับแรงดัน 17.5 โวลท์ ยกเว้นบิทที่ถูกโปรแกรมไปแล้ว ในการโปรแกรมแต่ละครั้ง สามารถโปรแกรมได้ครั้งละหนึ่งบิทเท่านั้น

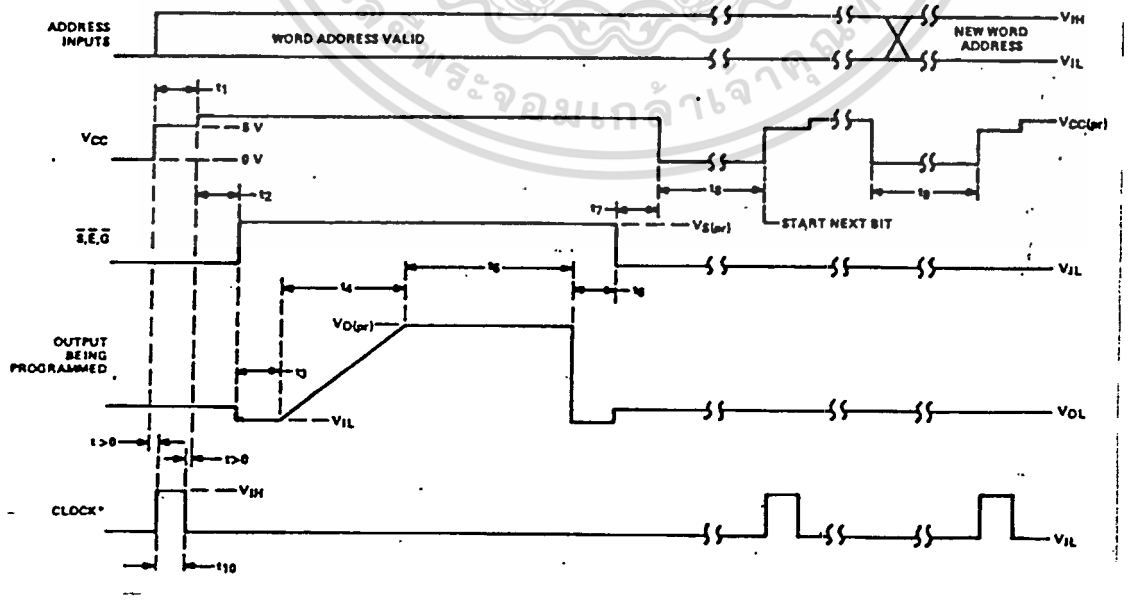
3.1.8 ป้อนพัลซ์เข้าที่ขา $\overline{S}$, $\overline{E}$ หรือ $\overline{G}$ ขณะโปรแกรมด้วยพัลซ์ช่วงเวลา 10-25 usec.

3.1.9 ลดระดับแรงดัน Vccp จาก 8.75 $\pm$ .25 โวลท์ เป็น 0 โวลท์

3.1.10 กลับไปทำในสเต็ปที่ 3.1.4 จนกระทั่งข้อมูลที่ทำการโปรแกรมชุดแรก ถูกโปรแกรมเสร็จ

3.1.11 ให้ทำซ้ำตามสเต็ปที่ 3.1.2 ถึง 3.1.9 จนกระทั่งหมดข้อมูลในเมมโมรี่ที่จะโปรแกรม

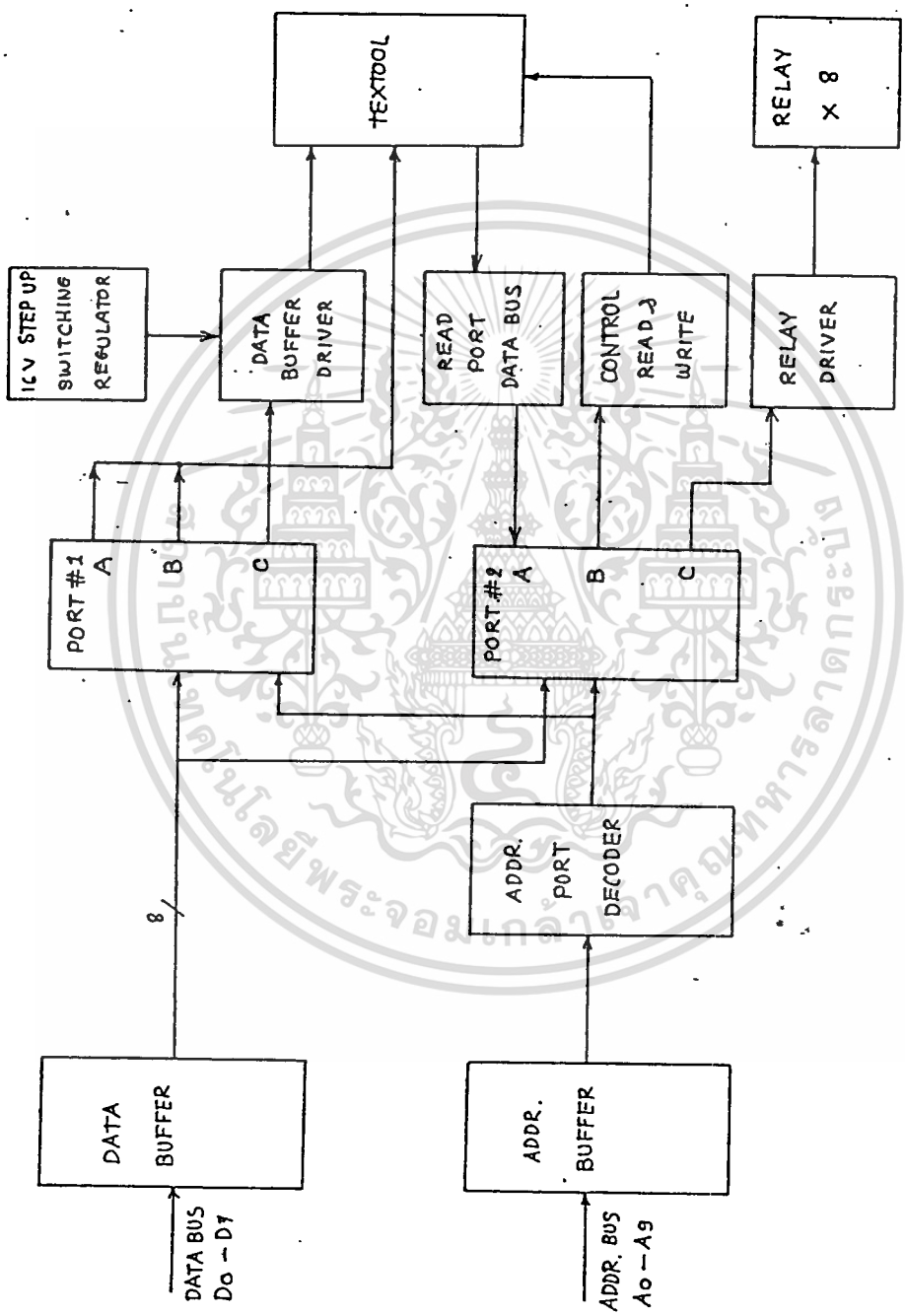
3.1.12 ให้ทำการตรวจสอบข้อมูลที่โปรแกรม กับข้อมูลที่อยู่ใน พรอม ว่าถูกต้องหรือไม่



รูปที่ 3.1 แสดงช่วงเวลาของแรงดันขณะทำการโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 บล็อกไดอะแกรมของเครื่องโปรแกรม พรอม



รูปที่ 3.2 บล็อกไดอะแกรม ของเครื่องโปรแกรม พรอม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของบล็อกไดอะแกรม (Block Diagram) ตามรูปที่ 3.2 เส้นทางเดินข้อมูล (Data Bus) D0-D7 จากสล็อตของ IBM PC จะส่งมายังบัฟเฟอร์ข้อมูล และส่งต่อไปยัง ไอโอ พอร์ท ตัวที่ 1 และตัวที่ 2 ส่วนสายแอดเดรส A0-A10 จากสล็อต ก็จะส่งไปยัง บัฟเฟอร์แอดเดรส เพื่อทำการดีโคดแอดเดรสของพอร์ท ตัวที่ 1 และตัวที่ 2 โดยสัญญาณเอ้าท์พุตที่ พอร์ท A และ B ของตัวที่ 1 จะถูกกำหนดให้เป็นค่าแอดเดรสของพรอม คือ A0-A10 เพื่อส่งไปยัง Textools ส่วนพอร์ท C จะถูกกำหนดให้เป็นค่าข้อมูลสำหรับโปรแกรม พรอม โดยระดับแรงดันที่ขาข้อมูลจะถูกยกระดับขึ้นเป็น +17.5 โวลท์ กระแส 250 mA. ได้มาจากวงจร Step up Switching Regulator แล้วส่งไปยัง Textools

สำหรับสัญญาณที่พอร์ท A ของตัวที่ 2 จะถูกกำหนดให้เป็นอินพุตพอร์ท ทำหน้าที่รับค่าข้อมูลที่อ่านมาจาก พรอม บน Textools แล้วส่งไปยัง CPU ส่วน พอร์ท B ทำหน้าที่ส่งสัญญาณควบคุมการเลือกอ่าน/เขียน พรอม และพอร์ท C เป็นตัวส่งสัญญาณควบคุม Relay Driverเพื่อบังคับสวิทช์เลือกตำแหน่งการอ่าน/เขียนตามต้องการ

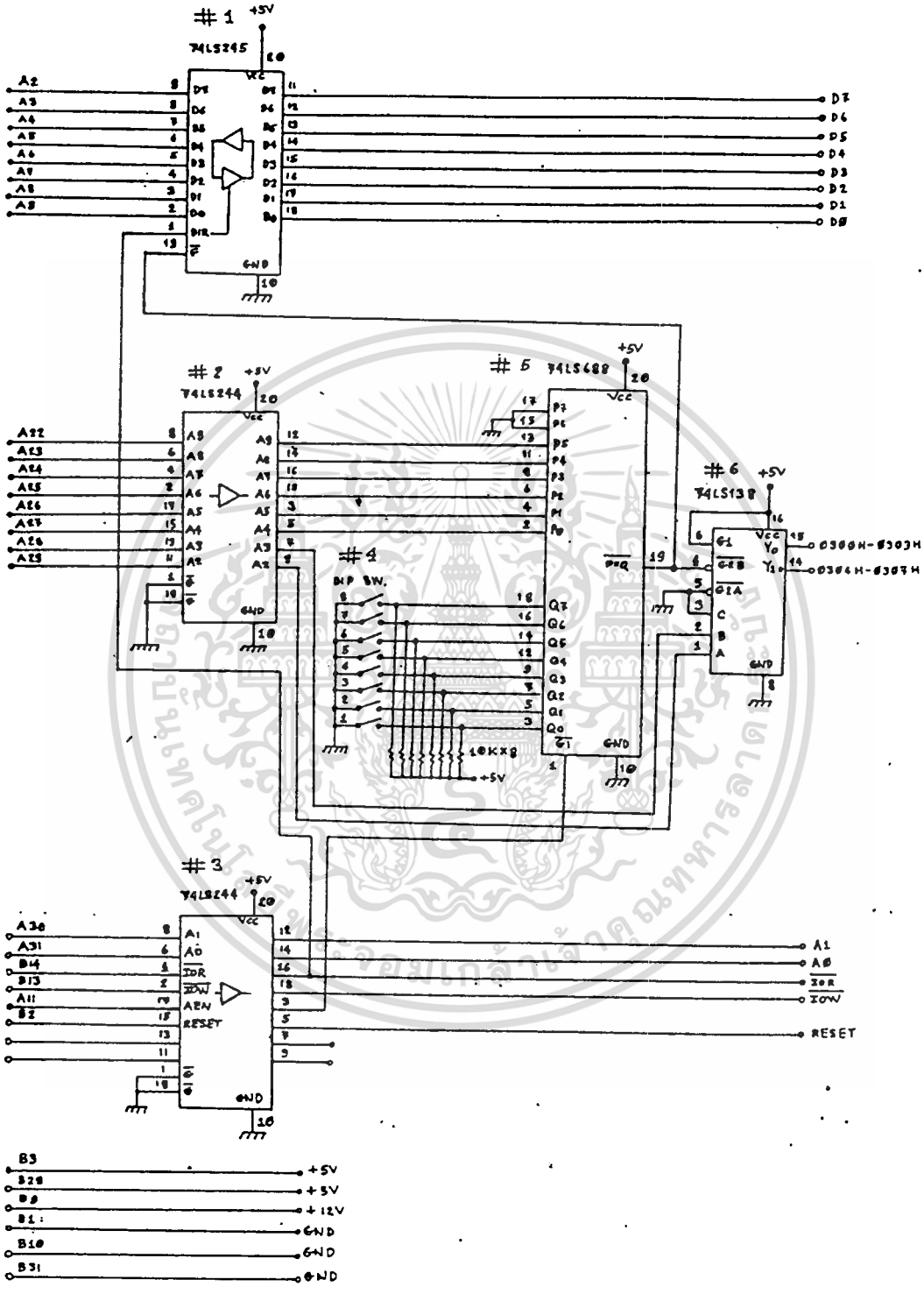
3.3 วงจรและการทำงานของเครื่องโปรแกรม พรอม

สำหรับการทำงานของวงจรของเครื่องโปรแกรม พรอม นี้จะแยกอธิบายออกเป็นส่วนๆ เพื่อให้ง่ายแก่การเข้าใจ โดยแบ่งออกเป็น 3 ส่วนด้วยกัน คือ

3.3.1 วงจร บัฟเฟอร์ และ ไอโอ ดิโค้ดเดอร์

จากวงจร ในรูปที่ 3.3 IC#1 (74LS245) เป็นบัฟเฟอร์แบบสองทิศทาง (Bi-Directional Buffer) ซึ่งทำหน้าที่เป็นตัวเชื่อมโยงของบัสข้อมูล (D0-D7) ที่รับมาจากช่องสล็อต IC#2 (74LS244) เป็นบัฟเฟอร์แบบทิศทางเดียว ซึ่งเชื่อมต่อกับบัสแอดเดรส (A2-A9) ของช่องสล็อตเช่นกัน IC#5 (74LS688) ทำหน้าที่เป็นตัวเปรียบเทียบ (Comparator) ค่าของอินพุทสองชุด ที่ถูกส่งเข้ามาทางขา P0-P7 เป็นค่าแอดเดรสของ A4-A9 ที่ส่งมาจาก IC#2 กับขา Q0-Q7 ซึ่งเป็นค่าแอดเดรสที่ถูกเลือกไว้ด้วย DIP SWITCH โดยวงจร ในรูปที่ 3.3 นี้ได้ถูกออกแบบให้ DIP SWITCH สามารถเลือกพอร์ท 8255 ได้สองพอร์ทด้วย IC#6 (74LS138) ที่ขาของ $\overline{Y0}$ คือ พอร์ท 0300-0303H และที่ขา $\overline{Y1}$ คือ พอร์ท 0304-0307H

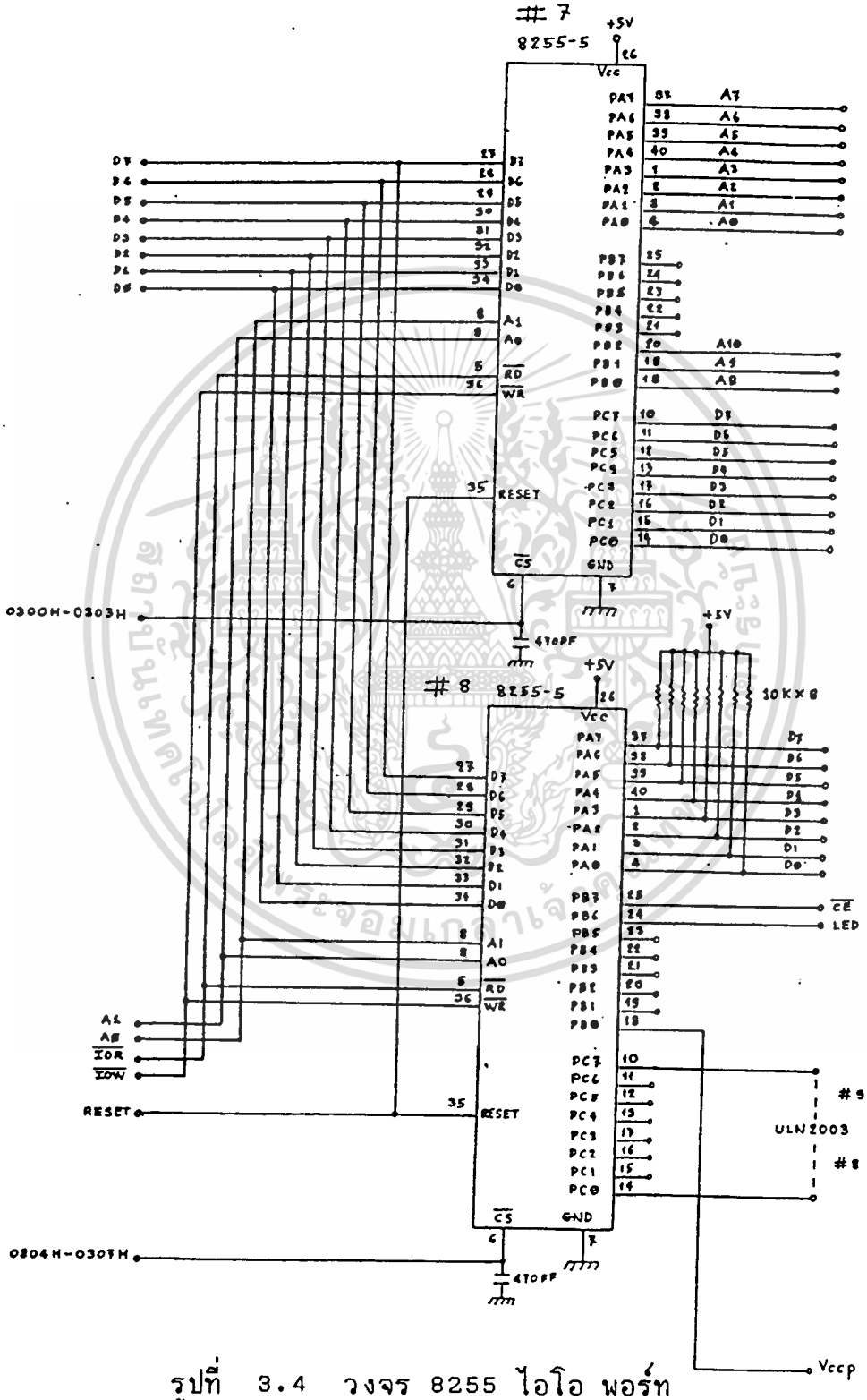
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 วงจรบัฟเฟอร์ และ ไอโอ ดีโค้ดเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 วงจร 8255 ไอโอ พอร์ท



รูปที่ 3.4 วงจร 8255 ไอโอ พอร์ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากวงจร ในรูปที่ 3.4 IC#7 (8255 PPI) ซึ่งถูกกำหนดให้อยู่ที่ตำแหน่งพอร์ทแอดเดรส 0300-0303H โดยที่ พอร์ท A และพอร์ท B ทำหน้าที่เป็นเอ้าท์พุทพอร์ท เพื่อใช้กำหนดตำแหน่งแอดเดรสให้แก่ ไบโพลาร์ พรอม เมื่อต้องการอ่านและเขียนข้อมูลกับ พรอม ส่วนพอร์ท C ทำหน้าที่เป็นเอ้าท์พุทพอร์ทของข้อมูล สำหรับทำการเขียนข้อมูลให้แก่ พรอม

ส่วน IC#8 (8255 PPI) ถูกกำหนดให้อยู่ที่ตำแหน่งพอร์ทแอดเดรส 0304-0307H โดยที่พอร์ท A กำหนดให้เป็นอินพุทพอร์ท สำหรับอ่านข้อมูลจากตัวพรอม ผ่านบัสข้อมูล มาเก็บไว้ยัง Memory Buffer ส่วนพอร์ท B และพอร์ท C ทำหน้าที่เป็นเอ้าท์พุทพอร์ท โดยจะส่งพัลซ์ ช่วงเวลา 10-25 u sec ในขณะทำการเขียนข้อมูลพร้อมทั้งส่งสัญญาณ BUSY ซึ่งแสดงผลด้วย LED ออกทางพอร์ท B สำหรับพอร์ท C ทำหน้าที่เป็นตัวขับรีเลย์ เพื่อเปลี่ยนทางเดินข้อมูลขณะทำการอ่านข้อมูล จากพรอม

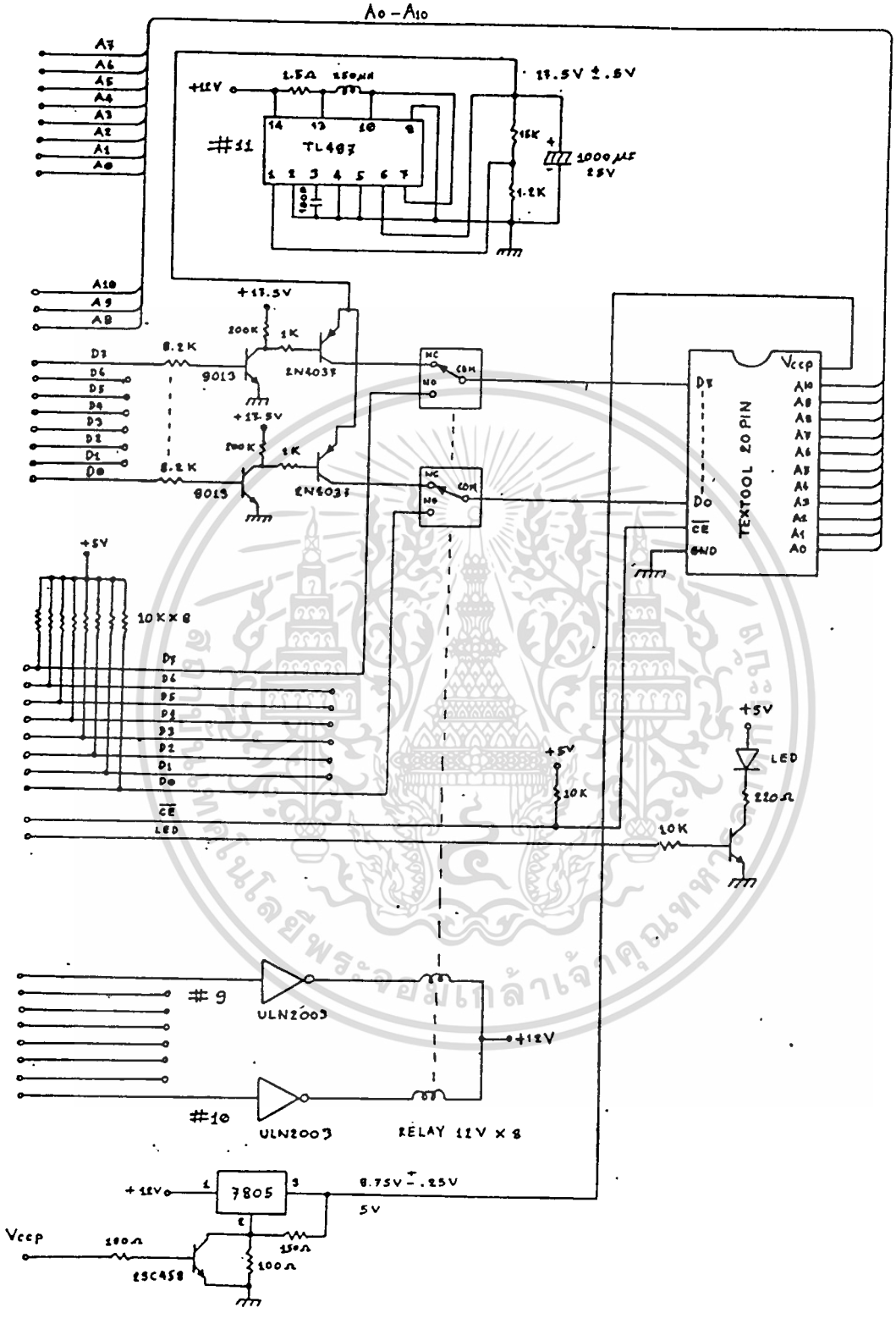
3.3.3 วงจรขับรีเลย์ และควบคุมการอ่าน/เขียนข้อมูล

จากวงจร ในรูปที่ 3.5 IC#11 (TL497) Switching Voltage Reg. ทำหน้าที่ยกระดับแรงดันจาก +12 Volts เป็น +17.5 Volts ที่กระแส 250 mA. เพื่อจ่ายให้แก่ขาข้อมูล (D0-D7) ของ พรอม ในขณะทำการเขียนข้อมูล โดยการใช้ทรานซิสเตอร์เบอร์ #9013 และเบอร์ #2N4037 เป็นตัวขับแรงดัน +17.5 Volts ที่รับมาจาก IC#11 ผ่านหน้าสัมผัสขั้ว NC ของรีเลย์ไปยังขาข้อมูลของ พรอม

ส่วน IC#9 และ IC#10 (ULN3003) ทำหน้าที่เป็นตัวขับรีเลย์ โดยการรับสัญญาณควบคุม รีเลย์ จากพอร์ท C ของ IC#8 และจ่ายแรงดัน +12 Volts ให้แก่ขดลวดของรีเลย์จำนวน 8 ตัว เพื่อบังคับให้หน้าสัมผัสของรีเลย์ เปลี่ยนตำแหน่งจากขั้ว NC เป็นขั้ว NO ซึ่งเป็นการเปลี่ยนหน้าที่จากการเขียนข้อมูล มาสู่การอ่านข้อมูล

สำหรับค่าแรงดัน Vccp ได้มาจากการเปลี่ยนแรงดัน +12 Volts มาเป็นแรงดัน +8.75 Volts โดยใช้ IC เบอร์ 7805 และทรานซิสเตอร์เบอร์ 2SC458 ทำหน้าที่เป็นตัวเปลี่ยนระดับแรงดัน เพื่อจ่ายให้กับขา Vcc ของพรอมใน Textools โดยอาศัยสัญญาณควบคุมจากพอร์ท B ของ IC#8 และในขณะเดียวกันที่ขา A0-A10 ของ พรอม จะได้รับการกำหนดค่าแอดเดรส จากพอร์ท A และพอร์ท B ของ IC#7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 วงจรขับรีเลย์ และควบคุมการอ่าน/เขียนข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ซอฟแวร์ (SOFT WARE)

4.1 การกำหนดหน้าที่การควบคุมด้วยซอฟแวร์

ในส่วนของฮาร์ดแวร์ที่ออกแบบสร้างไว้นั้น จะต้องอาศัย โปรแกรมซอฟแวร์เป็นตัวควบคุมการทำงาน ให้เป็นไปตามหน้าที่ (Function) ต่างๆที่กำหนด ซึ่งส่วนของซอฟแวร์นี้เขียนด้วยภาษาแอสแซมบลี้ 8086/8088 ทั้งหมดโดยให้มีความสามารถทำหน้าที่ในการควบคุมการทำงานของฮาร์ดแวร์ ดังนี้ คือ

[D] = DIR .....Directory

[R] = READ .....Into Buffer

[W] = WRITE .....Bipolar PROM

[B] = BLANK .....Check

[V] = VERIFY .....With Buffer

[T] = TYPE .....of Bipolar PROM

[L] = LOAD .....Disk File to Buffer

[S] = SAVE .....Buffer to Disk File

[O] = OS .....Shell

[M] = Modify .....Buffer

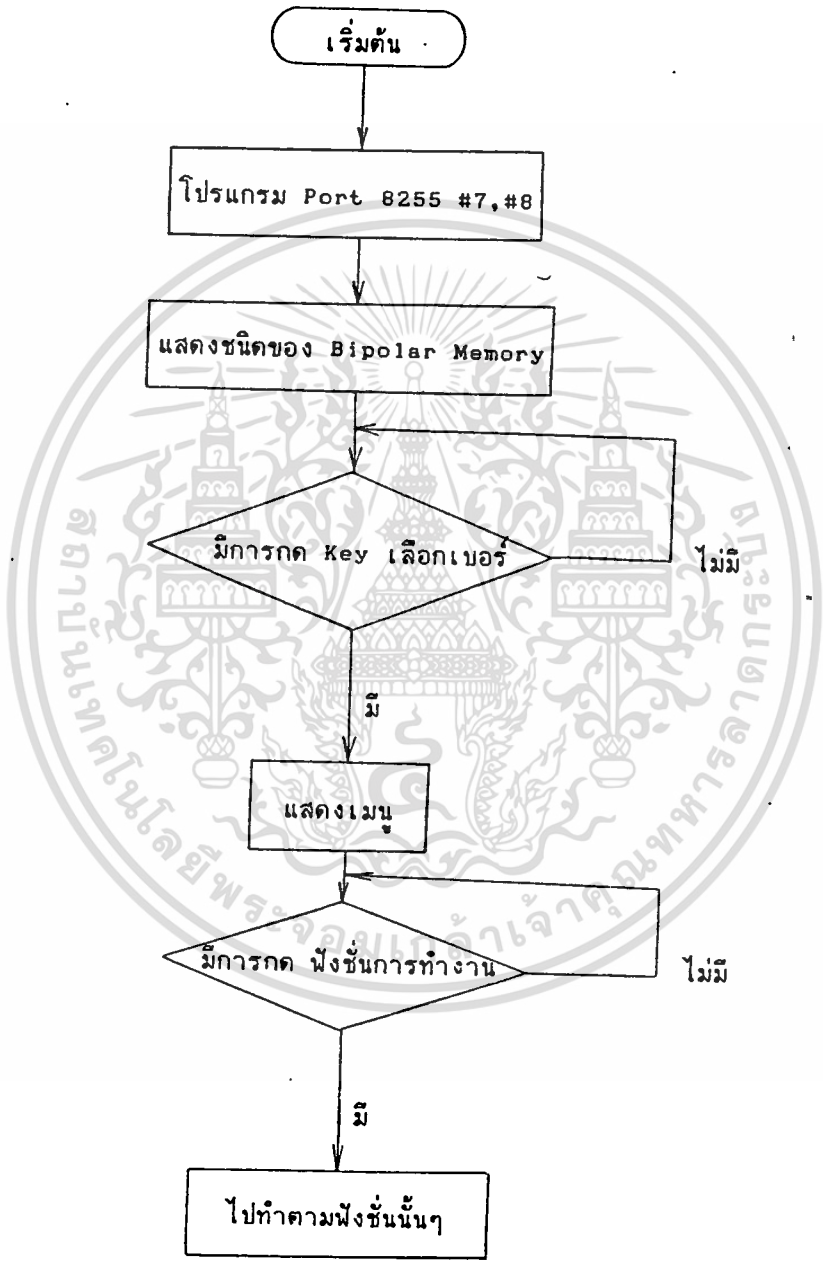
[Q] = Quit .....MS-DOS

4.2 การทำงานของซอฟแวร์

เนื่องจาก ซอฟแวร์ ที่เขียนไว้ด้วยภาษาแอสแซมบลี้ของ 8086/8088 นั้นถูกกำหนดให้ทำงานได้หลายหน้าที่ด้วยกัน และเพื่อให้เกิดความเข้าใจได้ง่ายขึ้น จึงขอแสดงรายระเอียดโดยแยกการอธิบายไว้เป็นส่วนๆ ด้วยโฟล์วชาร์ต (Flow Chart) ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

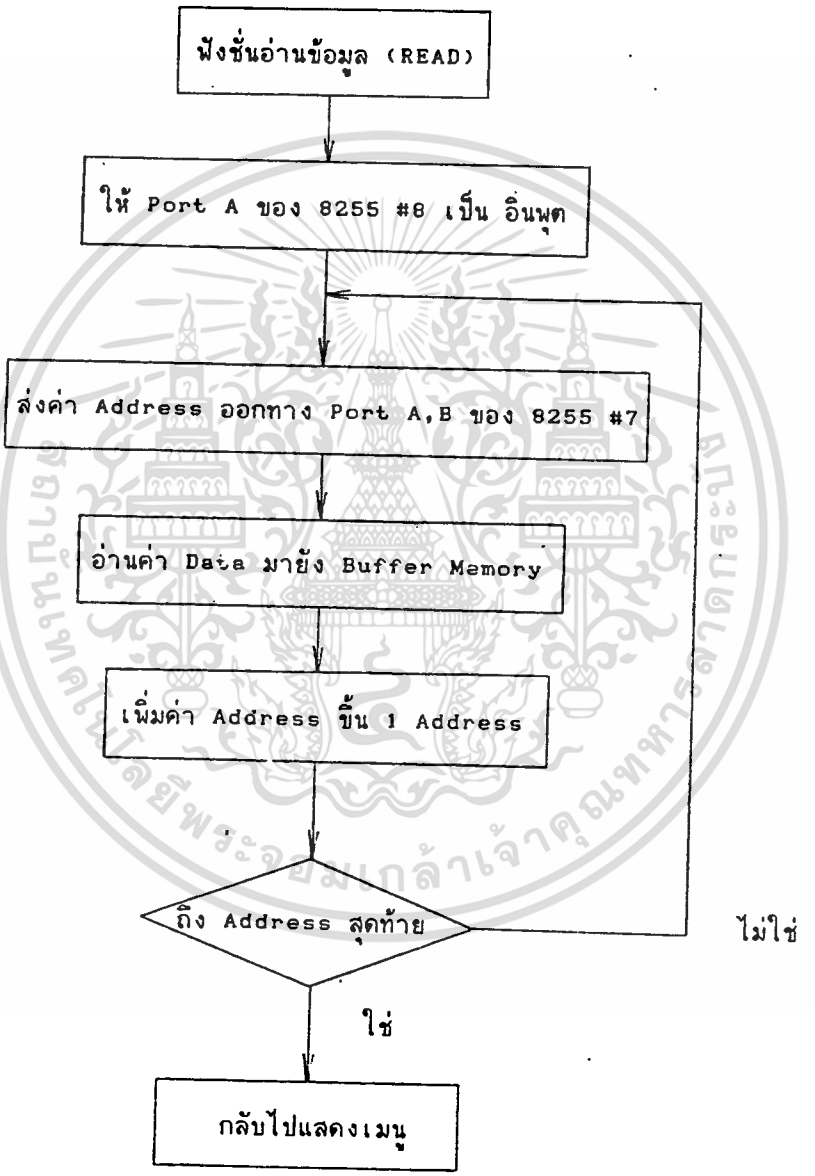
4.2.1 โฟล์วชาร์ตแสดงการทำงานของโปรแกรมหลัก (Main Program)



รูปที่ 4.1 แสดงการทำงานของโปรแกรมหลัก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

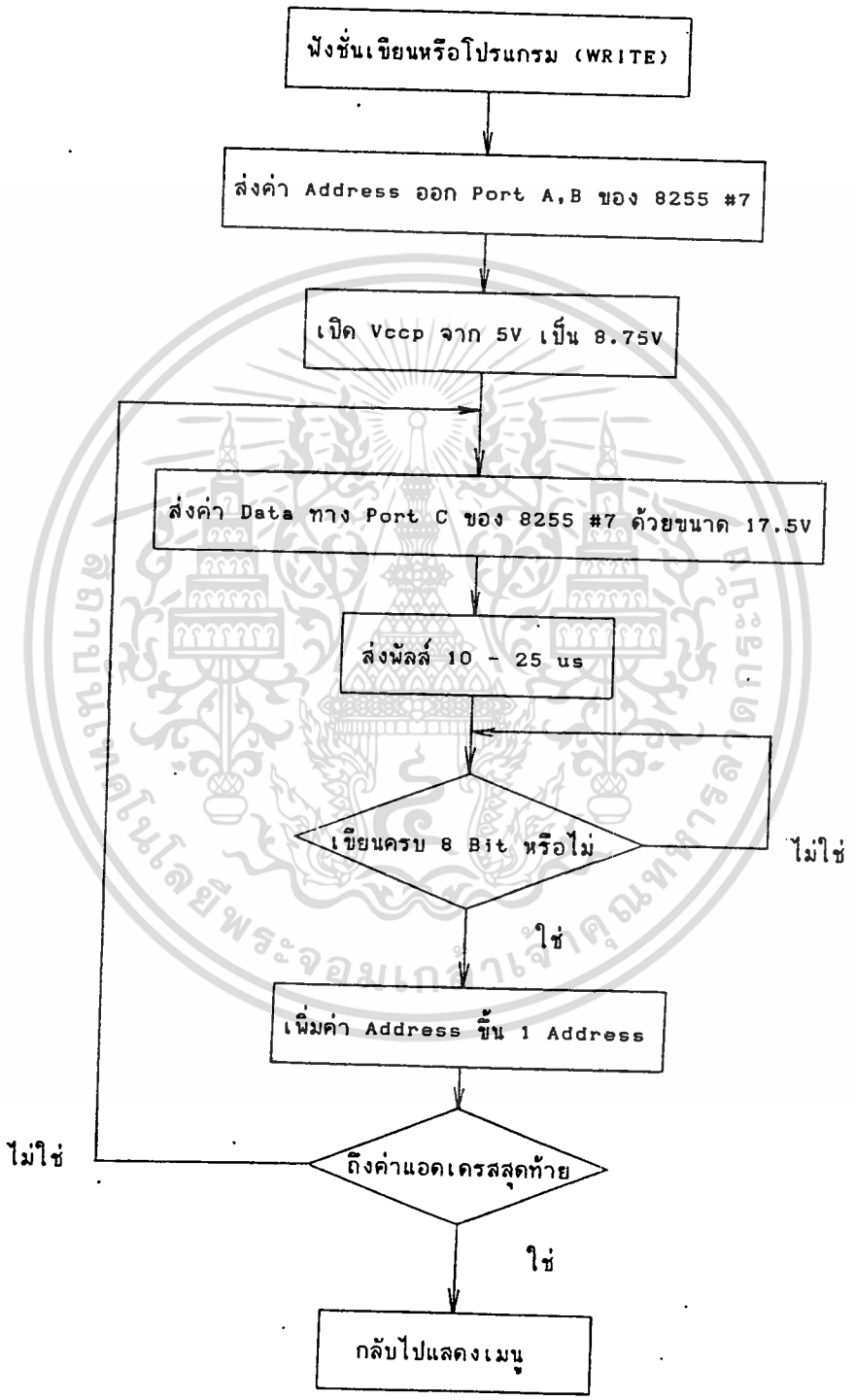
4.2.2 โฟล์วชาร์ตแสดงการอ่านข้อมูล (Read Data)



รูปที่ 4.2 แสดงการอ่านข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

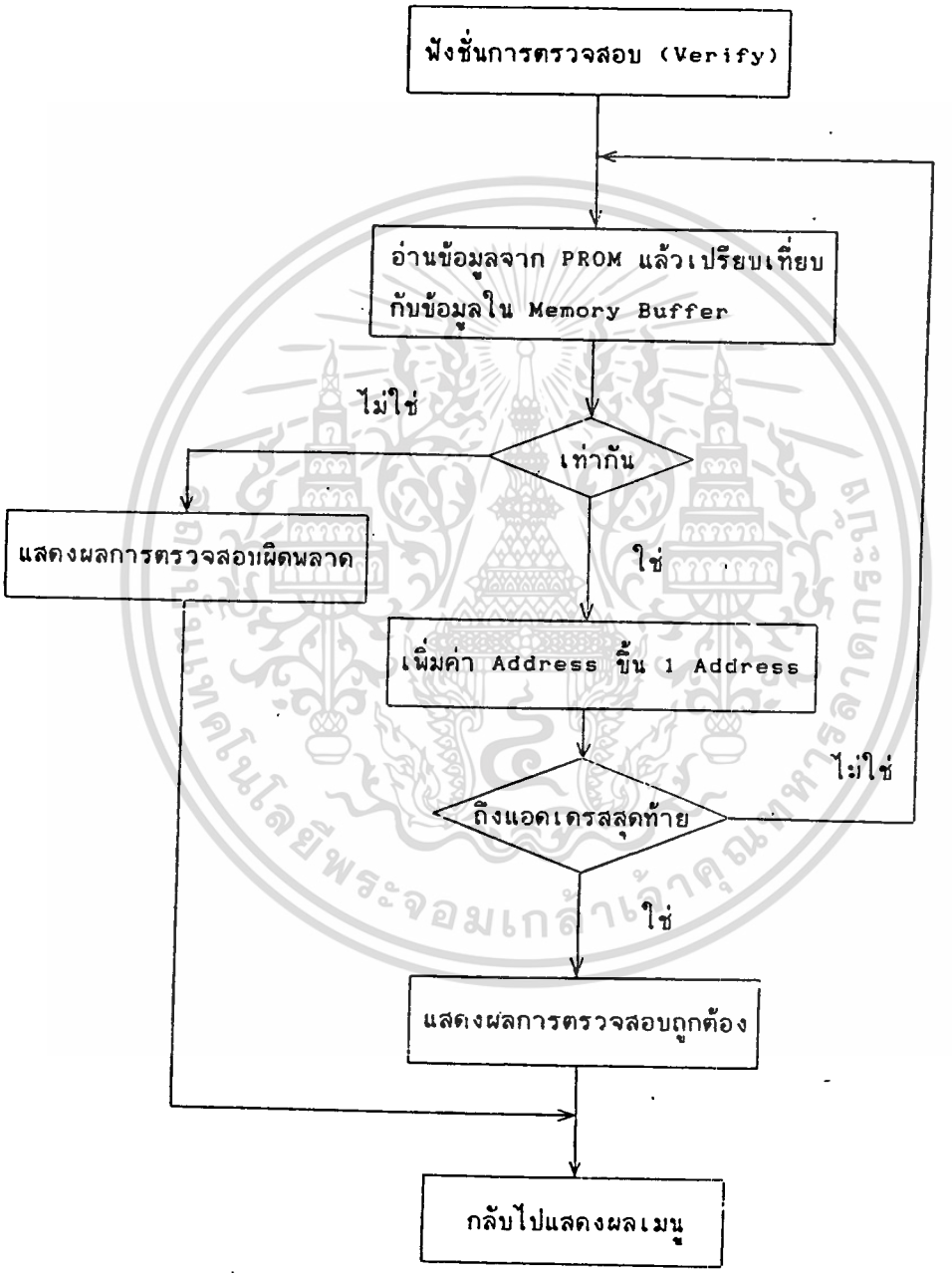
4.2.3 โฟล์วชาร์ตแสดงการเขียนหรือโปรแกรมข้อมูล (Write Data)



รูปที่ 4.3 แสดงการเขียนหรือโปรแกรมข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

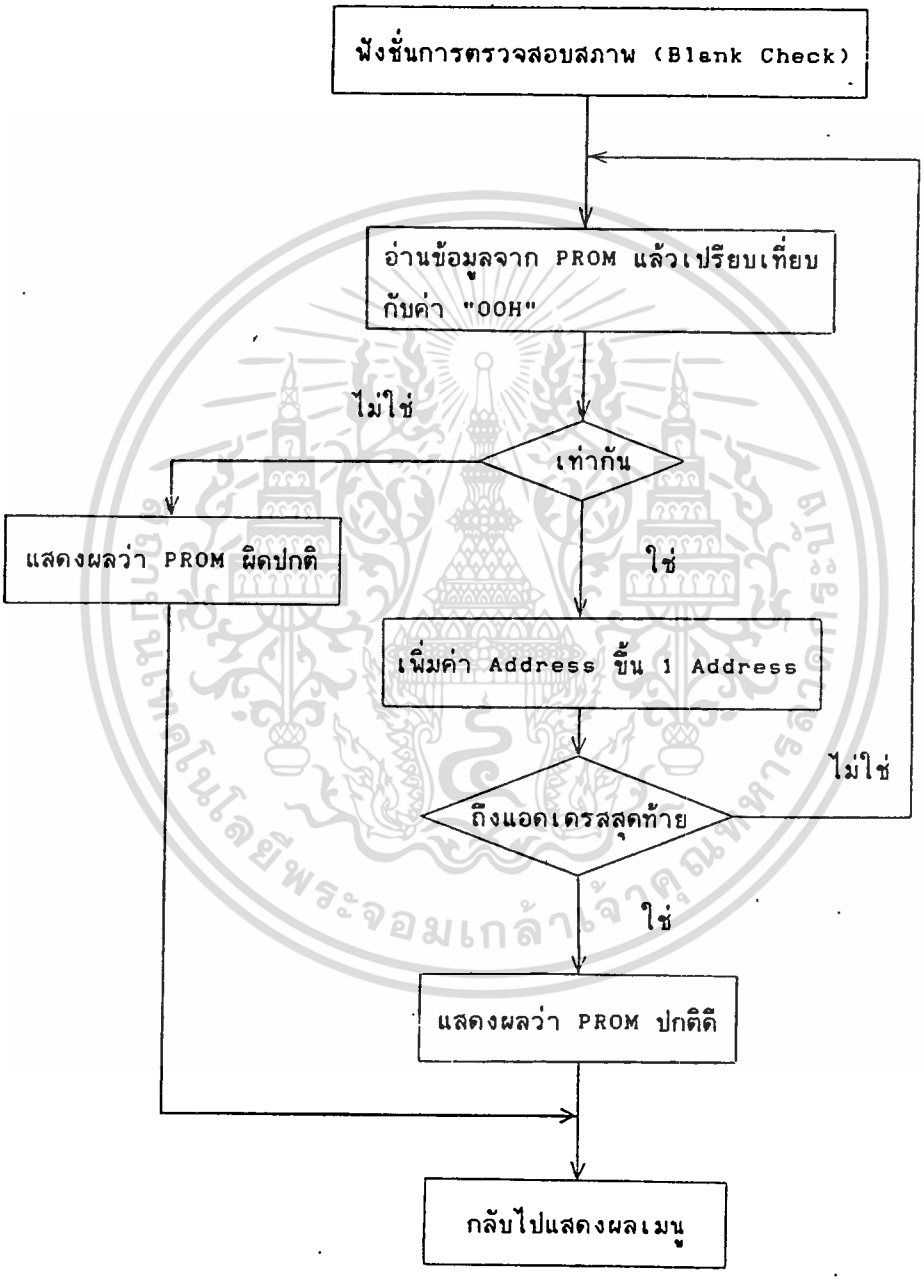
4.2.4 โฟล์วชาร์ตแสดงการตรวจสอบข้อมูล (Verify Data)



รูปที่ 4.4 แสดงการตรวจสอบข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

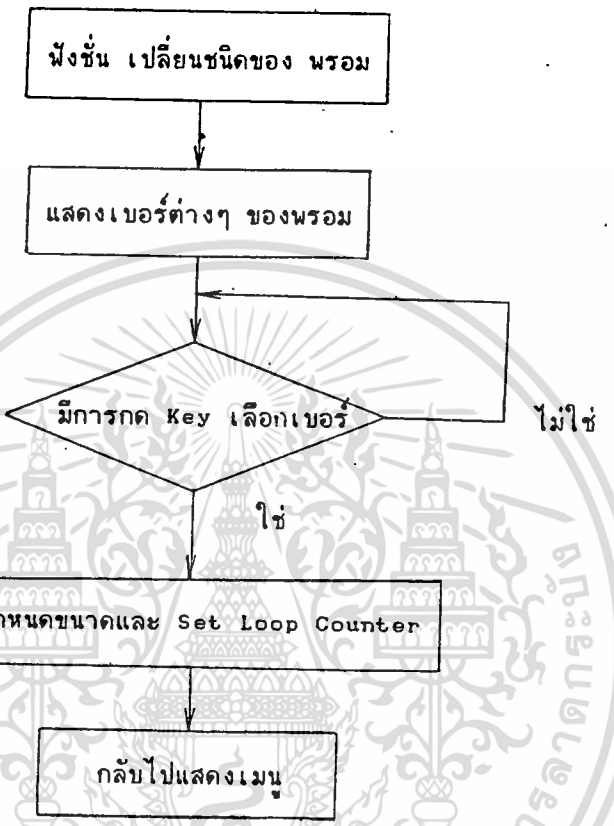
4.2.5 โฟล์วชาร์ตแสดงการตรวจสภาพของ พรอม (Blank Check)



รูปที่ 4.5 แสดงการตรวจสภาพ พรอม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2.6 โฟล์วชาร์ทแสดงฟังชั่นการเปลี่ยนชนิดของ พรอม



รูปที่ 4.6 แสดงฟังชั่นการเปลี่ยนชนิดของ พรอม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ซอฟแวร์ภาษาแอสแซมบลี้ 8086/8088 ของเครื่องโปรแกรม พรอม

```
TITLE   BIPOLAR PROM PROGRAMMER CARD ON IBM PC
;Industrial Technology Computer of KMITL
;=============================================================
display         MACRO     string
                MOV       AH,09H
                MOV       DX,offset string
                INT       21H
                ENDM
check_kbd_status    MACRO
                MOV       AH,0BH
                INT       21H
                ENDM
serch_first MACRO     fcb
                MOV       AH,11H
                MOV       DX,offset fcb
                INT       21H
                ENDM
serch_next  MACRO     fcb
                MOV       AH,12H
                MOV       DX,offset fcb
                INT       21H
                ENDM
current_disk MACRO
                MOV      AH,19H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            INT     21H
            ENDM
set_dta     MACRO   buffer
            MOV     AH,1AH
            MOV     DX,offset buffer
            INT     21H
            ENDM
GotoXY      MACRO   Row,Column
            PUSH    AX
            PUSH    BX
            PUSH    DX
            PUSH    CX
            MOV     AH,02
            MOV     BH,00
            MOV     DH,Row
            MOV     DL,Column
            INT     10H
            POP     CX
            POP     DX
            POP     BX
            POP     AX
            ENDM
convert     MACRO   value,base,destination    ;Convert
            local   table,start
            JMP     start
table       DB      '0123456789ABCDEF'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
start:      MOV     AL,value
            XOR     AH,AH
            XOR     BX,BX
            DIV     base
            MOV     BL,AL
            MOV     AL,CS:table[BX]
            MOV     destination,AL
            MOV     BL,AH
            MOV     AL,CS:table[BX]
            MOV     destination[1],AL
            ENDM
;-------------------------------------------------------------
            CODE    SEGMENT  PAGE
            ASSUME  CS:CODE,DS:DATA,SS:STACK
            MOV     AX,DATA              ;Set DS Register
            MOV     DS,AX
            MOV     AX,STACK             ;Set SS Register
            MOV     SS,AX
            MOV     SP,OFFSET TOP        ;Set SP Register
;-------------------------------------------------------------
;The Dummy of Extra_Segment
            Z_seg   SEGMENT
            Z_seg   ENDS
;-------------------------------------------------------------
Extra_buf   EQU     3000H
Create_file EQU     3CH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Open_file    EQU       3DH
Close_file   EQU       3EH
Read_file    EQU       3FH
Write_file   EQU       40H
;-------------------------------------------------------------
;Set 8255 PPI control word
             CALL      Ctrl_word
             CALL      Led_off          ;Set Textool socket free
;Clear all of screen and set cursor at Row 0,Col 0
             CALL      ClrScr1
;Display Header of Main Project
             Display   Header1
             Display   Header2
;Set cursor postion at Row 5,Col 0
             GotoXY    5,0
;Display prom type & choice
             CALL      Prom_type
;Display main menu choice
menu:        CALL      ClrScr2
             CALL      Disp_menu
;Check Key_pressed
_RD_Key:     CALL      Selection
             JMP       _RD_Key
;Display Directory in current drive.
Files_dirw:  PUSH      CX
             Current_Disk                ;Get Current function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CMP     AL,00H
            JNE     Disk_1
            GotoXY  05,00
            Display Drive1          ;Drive A
            JMP     fun_12H
Disk_1:     CMP     AL,01H
            JNE     Disk_2
            GotoXY  05,00
            Display Drive2          ;Drive B
            JMP     fun_12H
Disk_2:     CMP     AL,03H
            JNE     Disk_3
            GotoXY  05,00
Disk_3:     Display Drive3          ;Drive C
fun_12H:    set_dta buffer          ;function 1AH
            current_disk
            INC     AL
            MOV     BX,offset fcb
            MOV     [BX],AL         ;current dir
            serch_first fcb         ;function 11H
            CMP     AL,0FFH         ;directory entry found
            JE      allDone         ;no,no files on disk
            CALL    dispfiles       ;display any files
            INC     filecount       ;file(s) counter
            INC     files           ;yes increment file count
serch_dir:  serch_next fcb          ;This function
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CMP     AL,0FFH         ;directory entry found
                JE      done            ;yes
                CALL    dispfiles       ;display any files
                INC     filecount       ;file(s) counter
                INC     files           ;yes,increment file count
                JMP     serch_dir       ;check again
done:           Convert files,ten,messege
                Display Crlf
AllDone:        Display blank1
                Display messege         ;function 09H
all_done:       Display key_press
                CALL    cursor_out
_wait:          CALL    read_key        ;press any key to cont.
                MOV     AL,00
                MOV     [files],AL
                POP     CX
                JMP     dirw_back
dispfiles:      PUSH    DX
                PUSH    CX
                PUSH    BX
                PUSH    AX
                MOV     BX,offset buffer+1 ;index to filename
                MOV     CX,11           ;number of filename&ext.
bufinx:         MOV     DL,[BX]         ;send data in DL
                MOV     AH,02H          ;function
                INT     21H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                INC     BX              ;increment character
                CMP     CX,4            ;check first character
                JNE     cont            ;No, so write next char.
                CALL    DispDot         ;write dot between finame
cont:           LOOP    bufinx
                CMP     filecount,4     ;write filename&ext.
                JE      _5files         ;yes
                display blank           ;write blank
_5files:        MOV     filecount,0
                POP     AX
                POP     BX
                POP     CX
                POP     DX
                RET
DispDot:        PUSH    DX
                PUSH    AX
                MOV     DL,'.'
                MOV     AH,02H
                INT     21H
                POP     AX
                POP     DX
                RET
;---------------------------------------------------------------
Selection       PROC    NEAR
Read:           CALL    read_key
                CMP     AL,54H          ;key T is prom type
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JNE     check1
                CALL    ClrScr2
                CALL    Prom_type
                JMP     menu
check1:         CMP     AL,44H          ;key D is files Directory
                JNE     check2
                CALL    ClrScr2
                JMP     Files_dirw
dirw_back:      JMP     menu
check2:         CMP     AL,42H          ;key B is blank check
                JNE     check3
                MOV     AL,_size
                CMP     AL,04H
                JNE     K1
                CALL    Blank_ck4
                JMP     Read
K1:             CALL    Blank_ck8
                JMP     Read
check3:         CMP     AL,52H          ;key R is read prom
                JNE     check4
                CALL    _Read_prom
                CALL    ClrScr4
                JMP     Read
check4:         CMP     AL,53H          ;key S is Save file
                JNE     check5
                CALL    Clrscr2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CALL    _out_file
                JMP     menu
check5:         CMP     AL,4CH          ;key L is Load file
                JNE     check6
                CALL    Clrscr2
                CALL    _In_file
                JMP     menu
check6:         CMP     AL,56H          ;key V is Verify
                JNE     check7
                CALL    _Verify
                CALL    ClrScr4
                JMP     Read
check7:         CMP     AL,57H          ;key W is Write prom
                JNE     check8
                MOV     AL,_size
                CMP     AL,04H
                JNE     Q1
                CALL    _Prom_WR4
                CALL    _Ver_Prog
                CALL    ClrScr4
                JMP     Read
Q1:             CALL    _Prom_WR8
                CALL    _Ver_Prog
                CALL    ClrScr4
                JMP     Read
check8:         CMP     AL,4DH          ;key M is Modify buffer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JNE     check9
                CALL    Move_Extra
                CALL    Modify
                CALL    Extra_Main
                CALL    Disp_Save
                JMP     Read
check9:         CMP     AL,4FH          ;key O is OS shell
                JNE     check10
                CALL    Shell
                CALL    Disp_Save
                JMP     _RD
check10:        CMP     AL,51H          ;key Q is quit to MS-DOS
                JNE     _RD
                CALL    EXIT
_RD:            RET
Selection       ENDP
;-----------------------------------------------------------
EXIT            PROC    NEAR
;If [AL] = 51H then Quit
;Set cursor at row 20 col 00
Quit:           GotoXY  20,00
                MOV     AH,4CH          ;Terminate process
                INT     21H
                RET
EXIT            ENDP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------------------
Shell       PROC    NEAR
            PUSH    ES
            PUSH    BX
            CALL    ClrScr1
            Display _Shell_msg
            GotoXY  01,00
;Modify allocate for free Memory
            MOV     AH,4AH
            MOV     BX,1000H
            INT     21H
;Point to Parameter Block
            MOV     AX,Z_seg
            MOV     ES,AX
            MOV     PR1,AX
            MOV     PR2,AX
            MOV     PR3,AX
;Point to file name String
            MOV     DX,offset Filename1
            MOV     BX,offset Para_block
;Save Stack Pointor
            MOV     Keep_ss,SS
            MOV     Keep_sp,SP
;Load file COMMAND.COM
            MOV     AH,4BH
            MOV     AL,00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                INT         21H
;Affterwords, Restore Register
                MOV         AX,Data
                MOV         DS,AX
                MOV         SS,Keep_ss
                MOV         SP,Keep_sp
                POP         BX
                POP         ES
                RET
                Shell       ENDP
;-------------------------------------------------------------
Modify          PROC        NEAR
                PUSH        ES
                PUSH        DX
                PUSH        CX
                PUSH        BX
                CALL        ClrScr1
                Display     Intro
                GotoXY      02,00
;Modify allocate for free Memory
                MOV         AH,4AH
                MOV         BX,1000H
                INT         21H
;Point to Parameter block
                MOV         AX,Z_seg
                MOV         ES,AX
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     PR1,AX
                MOV     PR2,AX
                MOV     PR3,AX
 ;Point to File name String
                MOV     DX,offset Filename2
                MOV     BX,offset Para_block
 ;Save Stack Pointor
                MOV     Keep_ss,SS
                MOV     Keep_sp,SP
 ;Load The Program and Execute
                MOV     AH,4BH
                MOV     AL,00H
                INT     21H
  ;Affterwords, Restore Registers
                MOV     AX,Data
                MOV     DS,AX
                MOV     SS,Keep_ss
                MOV     SP,Keep_sp
                POP     BX
                POP     CX
                POP     DX
                POP     ES
                RET
Modify          ENDP
;---------------------------------------------------------------
Move_Extra      PROC    NEAR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;Memory allocate for free
            PUSH    ES
            PUSH    BX
            PUSH    CX
            MOV     BX,0800H
            MOV     AH,4AH
            INT     21H
            MOV     CX,0800H
            MOV     BX,0000H
            MOV     DI,0000H
;Point to Extra buffer
            MOV     AX,Extra_buf
            MOV     ES,AX
_Get1:      MOV     AL,[HEX_buffer+BX]
            MOV     ES:[DI],AL
            INC     BX
            INC     DI
            LOOP    _Get1
            POP     CX
            POP     BX
            POP     ES
            RET
Move_Extra  ENDP
;-------------------------------------------------------------
Extra_Main  PROC    NEAR
;Memory allocate for free
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                PUSH    ES
                PUSH    BX
                PUSH    CX
                MOV     BX,0800H
                MOV     AH,4AH
                INT     21H
                MOV     CX,0800H
                MOV     BX,0000H
                MOV     DI,0000H
;Point to Extra buffer
                MOV     AX,Extra_buf
                MOV     ES,AX
_Get2:          MOV     AL,ES:[DI]
                MOV     [HEX_buffer+BX],AL
                INC     BX
                INC     DI
                LOOP    _Get2
                POP     CX
                POP     BX
                POP     ES
                RET
Extra_Main      ENDP
;---------------------------------------------------------------
Disp_Save       PROC    NEAR
;--Disp message Current type =
;--Disp message Use Card NO.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;--buffer length =      byte
            CALL      ClrScr1
            Display   Header1
            Display   Header2
            MOV       AL,_Prom_type
DP0:        CMP       AL,01H
            JNE       DP1
            GotoXY    03,17
            Display   Prom1
            JMP       _Next
DP1:        CMP       AL,02H
            JNE       DP2
            GotoXY    03,17
            Display   Prom2
            JMP       _Next
DP2:        CMP       AL,03H
            JNE       DP3
            GotoXY    03,17
            Display   Prom3
            JMP       _Next
DP3:        CMP       AL,04H
            JNE       DP4
            GotoXY    03,17
            Display   Prom4
            JMP       _Next
DP4:        CMP       AL,05H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        JNE     DP5
        GotoXY  03,17
        Display Prom5
        JMP     _Next
DP5:    CMP     AL,06H
        JNE     DP6
        GotoXY  03,17
        Display Prom6
        JMP     _Next
DP6:    CMP     AL,07H
        JNE     DP7
        GotoXY  03,17
        Display Prom7
        JMP     _Next
DP7:    CMP     AL,08H
        JNE     DP8
        GotoXY  03,17
        Display Prom8
        JMP     _Next
DP8:    CMP     AL,09H
        JNE     _Next
        GotoXY  03,17
        Display Prom9
_Next:  MOV     AL,_Contain
        CMP     AL,00H
        JNE     DL0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                GotoXY    03,65
                Display   length_000
                JMP       Empty
DL0:            MOV       AL,_length
                CMP       AL,01H
                JNE       DL1
                GotoXY    03,65
                Display   length_123
                JMP       Empty
DL1:            CMP       AL,02H
                JNE       DL2
                GotoXY    03,65
                Display   length_129
                JMP       Empty
DL2:            CMP       AL,03H
                JNE       DL3
                GotoXY    03,65
                Display   length_131
                JMP       Empty
DL3:            CMP       AL,04H
                JNE       Empty
                GotoXY    03,65
                Display   length_185
Empty:          GotoXY    05,00
                Display   main_menu
                GotoXY    19,14
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            RET
Disp_Save   ENDP
;-------------------------------------------------------------
cursor_out  PROC      NEAR
;Set cursor out of screen at Row 25,Col 0
            GotoXY    25,00
            RET
cursor_out  ENDP
;-------------------------------------------------------------
;Output buffer to disk file type file handle
_out_file   PROC      NEAR
            PUSH      CX
;Display output buffer data to disk file
            Display   _output_msg
;Set cursor at row 10 col 22
            GotoXY    10,22
;Enter file_name with function AH = 0AH
            MOV       AH,0AH
            MOV       DX,offset _filename
            INT       21H
            CALL      Cursor_out
;Insert zero into _filename to ASCIIZ string
            MOV       BL,_filename+1
            MOV       BH,00
            MOV       [_filename+BX+2],0
;Create file handle
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     DX,offset _filename+2
                MOV     AH,Create_file
                MOV     CX,00H
                INT     21H
                MOV     _handle,AX      ;Save handle value
                JC      _error2         ;if function failed
;Open file handle
                MOV     AH,Open_file
                MOV     AL,02H          ;R/W file
                MOV     DX,offset _filename+2
                INT     21H
                JC      _error2         ;if function failed
;Write file handle
_write:         MOV     BX,_handle      ;get handle back to BX
                MOV     DX,offset HEX_buffer  ;ASCII buffer area
                MOV     CX,0800H        ;data = 2048 byte
                MOV     AH,Write_file
                INT     21H
                JC      _error2         ;if function failed
                CMP     AX,0800H        ;Check No. to write
                JNE     _error2
;Close file handle
                MOV     BX,_handle      ;get handle back to BX
                MOV     AH,Close_file
                JC      _error2         ;if function failed
;Display message save file - ok -
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;and set cursor at row 11 col 22
            GotoXY    11,22
            Display   _save_ok
            CALL      Cursor_out
            CALL      Read_key
            JMP       _out1
;Display message save file error
_error2:    CALL      Beep
;Set cursor at row 11 col 22
            GotoXY    11,22
            Display   _Disk_fail1
            CALL      Cursor_out
            CALL      Read_key
_out1:      POP       CX
            RET
_out_file   ENDP
;-------------------------------------------------------------
;Intput file to memory buffer type file handle
_In_file    PROC      NEAR
            PUSH      CX
;Display Input disk file to buffer
            Display   _input_msg
;Set cursor at row 10 col 22
            GotoXY    10,22
;Enter file_name with function AH = 0AH
            MOV       AH,0AH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     DX,offset _filename
                INT     21H
                CALL    Cursor_out
;Insert zero into filename to ASCIIZ string
                MOV     BL,_filename+1
                MOV     BH,00
                MOV     [_filename+BX+2],0
;Open file handle
                MOV     AH,Open_file
                MOV     AL,00H          ;Open file for reading
                MOV     DX,offset _filename+2
                INT     21H
                MOV     _handle,AX      ;save handle value
                JC      _error3         ;if function failed
;Read file handle
_Read:          MOV     AH,Read_file
                MOV     BX,_handle      ;get handle back to BX
                MOV     CX,0800H        ;byte to read = 2048
                MOV     DX,offset HEX_buffer    ;ASCII buffer
                INT     21H
                JC      _error3         ;if function failed
                CMP     AX,00
                JE      _Close          ;if equal is end of file
;Close file handle
_Close:         MOV     BX,_handle      ;get handle back to BX
                MOV     AH,Close_file
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JC          _error3          ;if function failed
;Display message load file - ok -
;and set cursor at row 11 col 22
                GotoXY      11,22
                Display     _load_ok
                CALL        Cursor_out
                CALL        Read_key
                JMP         _out2
;Display message load file error
_error3:        CALL        Beep
;Set cursor at row 11 col 22
                GotoXY      11,22
                Display     _Disk_fail2
                CALL        Cursor_out
                CALL        Read_key
_out2:          POP         CX
                RET
_In_file        ENDP
;-------------------------------------------------------------
;Read PROM data into memory buffer
_Read_prom      PROC        NEAR
                PUSH        CX
                MOV         CX,_Counter
                GotoXY      18,00
                Display     _data_msg1
                CALL        Cursor_out
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CALL    Relay_on
        CALL    CE_ready
Y3:     CALL    read_key
        CMP     AL,0DH          ;check return key
        JNE     Y3
        CALL    led_on
        MOV     DX,00
        MOV     BX,00
J1:     MOV     AL,DL
        CALL    Addr_LSB
        MOV     AL,DH
        CALL    Addr_MSB
        CALL    read_data
        MOV     [HEX_buffer+BX],AL   ;Put data
        INC     DX
        INC     BX
        CALL    Delay1
        LOOP    J1
        CALL    led_off
        MOV     _Contain,0FFH
        CALL    Disp_length
        GotoXY  18,00
        Display _data_msg2
        CALL    Cursor_out
        CALL    read_key
        CALL    Relay_off
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            POP     CX
            RET
_Read_prom  ENDP
;-------------------------------------------------------------
;Verify prom with memory buffer when Prog Ready
_Ver_Prog   PROC    NEAR
            PUSH    CX
            MOV     CX,_Counter
            CALL    Relay_on
            CALL    Delay2
            CALL    CE_ready
            MOV     BX,00
            MOV     DX,00
R2:         MOV     AL,DL
            CALL    Addr_LSB
            MOV     AL,DH
            CALL    Addr_MSB
            CALL    read_data
            MOV     _Result,AL
            MOV     AL,[HEX_buffer+BX]
            CMP     _Result,AL
            JNE     _Prog_err
            INC     DX
            INC     BX
            LOOP    R2
            CALL    led_off
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                GotoXY    18,00
                Display   _data_msg7
                JMP       _back1
_Prog_err:      CALL      Beep
                CALL      led_off
                GotoXY    18,00
                Display   _data_msg8
_back1:         CALL      Cursor_out
                CALL      read_key
                CALL      Relay_off
                POP       CX
                RET
_Ver_Prog       ENDP
;-------------------------------------------------------------
;Program bipolar prom
_Prom_WR4       PROC      NEAR
                PUSH      CX
                MOV       CX,_Counter
                GotoXY    18,00
                Display   _Data_msg6
                CALL      Cursor_out
                CALL      Init_prom
K3:             CALL      read_key
                CMP       AL,0DH          ;check return key
                JNE       K3
                GotoXY    19,00
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                Display  Programing
                GotoXY   19,26
                CALL     led_on
                MOV      BX,00
                MOV      DX,00
WR4:            MOV      AL,DL
                CALL     Addr_LSB
                MOV      AL,DH
                CALL     Addr_MSB
                MOV      AL,[HEX_buffer+BX]
                MOV      _Save,AL
                CALL     Bit_test4
                INC      BX
                INC      DX
                LOOP     WR4
                POP      CX
                RET
_Prom_WR4       ENDP
;-----------------------------------------------------------------
Bit_test4       PROC     NEAR
_Bit0:          TEST     AL,00000001B    ;AL = 01H
                JZ       _Bit1           ;if bit 0 <> 1 then
                AND      AL,00000001B
                CALL     _Data_Prog      ;Sent data to Prog
                CALL     Vccp_on
                CALL     _CE_Pulse       ;Pulse 10us
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CALL    Vccp_off
                CALL    _Data_Free
_Bit1:          MOV     AL,_Save
                TEST    AL,00000010B    ;AL = 02H
                JZ      _Bit2           ;if bit 1 <> 1 then
                AND     AL,00000010B
                CALL    _Data_Prog      ;Sent data to Prog
                CALL    Vccp_on
                CALL    _CE_Pulse       ;Pulse 10us
                CALL    Vccp_off
                CALL    _Data_Free
_Bit2:          MOV     AL,_Save
                TEST    AL,00000100B    ;AL = 04H
                JZ      _Bit3           ;if bit 2 <> 1 then
                AND     AL,00000100B
                CALL    _Data_Prog      ;Sent data to Prog
                CALL    Vccp_on
                CALL    _CE_Pulse       ;Pulse 10us
                CALL    Vccp_off
                CALL    _Data_Free
_Bit3:          MOV     AL,_Save
                TEST    AL,00001000B    ;AL = 08H
                JZ      _Next_1         ;if bit 3 <> 1 then
                AND     AL,00001000B
                CALL    _Data_Prog      ;Sent data to Prog
                CALL    Vccp_on
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CALL    _CE_Pulse       ;Pulse 10us
                CALL    Vccp_off
                CALL    _Data_Free
_Next_1:        RET
Bit_test4       ENDP
;-------------------------------------------------------------
;Program bipolar prom
_Prom_WR8       PROC    NEAR
                PUSH    CX
                MOV     CX,_Counter
                GotoXY  18,00
                Display _data_msg6
                CALL    Cursor_out
                CALL    Init_prom
P3:             CALL    read_key
                CMP     AL,0DH          ;check return key
                JNE     P3
                GotoXY  19,00
                Display Programing
                GotoXY  19,26
                CALL    led_on
                MOV     BX,00
                MOV     DX,00
WR8:            MOV     AL,DL
                CALL    Addr_LSB
                MOV     AL,DH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CALL    Addr_MSB
                MOV     AL,[HEX_buffer+BX]
                MOV     _Save,AL
                CALL    Bit_test8
                INC     BX
                INC     DX
                LOOP    WR8
                POP     CX
                RET
_Prom_WR8       ENDP
;-------------------------------------------------------------------
Bit_test8       PROC    NEAR
Bit0:           TEST    AL,00000001B    ;AL = 01H
                JZ      Bit1            ;if bit 0 <> 1 then
                AND     AL,00000001B
                CALL    _Data_Prog      ;Sent data to Prog
                CALL    Vccp_on
                CALL    _CE_Pulse       ;Pulse 10us
                CALL    Vccp_off
                CALL    _Data_Free
Bit1:           MOV     AL,_Save
                TEST    AL,00000010B    ;AL = 02H
                JZ      Bit2            ;if bit 1 <> 1 then
                AND     AL,00000010B
                CALL    _Data_Prog      ;Sent data to Prog
                CALL    Vccp_on
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        CALL    _CE_Pulse       ;Pulse 10us
        CALL    Vccp_off
        CALL    _Data_Free
Bit2:   MOV     AL,_Save
        TEST    AL,00000100B    ;AL = 04H
        JZ      Bit3            ;if bit 2 <> 1 then
        AND     AL,00000100B
        CALL    _Data_Prog      ;Sent data to Prog
        CALL    Vccp_on
        CALL    _CE_Pulse       ;Pulse 10us
        CALL    Vccp_off
        CALL    _Data_Free
Bit3:   MOV     AL,_Save
        TEST    AL,00001000B    ;AL = 08H
        JZ      Bit4            ;if bit 3 <> 1 then
        AND     AL,00001000B
        CALL    _Data_Prog      ;Sent data to Prog
        CALL    Vccp_on
        CALL    _CE_Pulse       ;Pulse 10us
        CALL    Vccp_off
        CALL    _Data_Free
        JMP     Bit4
_WR8:   JMP     WR8
Bit4:   MOV     AL,_Save
        TEST    AL,00010000B    ;AL = 10H
        JZ      Bit5            ;if bit 4 <> 1 then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        AND     AL,00010000B
        CALL    _Data_Prog      ;Sent data to Prog
        CALL    Vccp_on
        CALL    _CE_Pulse       ;Pulse 10us
        CALL    Vccp_off
        CALL    _Data_Free
Bit5:   MOV     AL,_Save
        TEST    AL,00100000B    ;AL = 20H
        JZ      Bit6            ;if bit 5 <> 1 then
        AND     AL,00100000B
        CALL    _Data_Prog      ;Sent data to Prog
        CALL    Vccp_on
        CALL    _CE_Pulse       ;Pulse 10us
        CALL    Vccp_off
        CALL    _Data_Free
Bit6:   MOV     AL,_Save
        TEST    AL,01000000B    ;AL = 40H
        JZ      Bit7            ;if bit 6 <> 1 then
        AND     AL,01000000B
        CALL    _Data_Prog      ;Sent data to Prog
        CALL    Vccp_on
        CALL    _CE_Pulse       ;Pulse 10us
        CALL    Vccp_off
        CALL    _Data_Free
Bit7:   MOV     AL,_Save
        TEST    AL,10000000B    ;AL = 80H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JZ      _Next_2         ;if bit 7 <> 1 then
                AND     AL,10000000B
                CALL    _Data_Prog      ;Sent data to Prog
                CALL    Vccp_on
                CALL    _CE_Pulse       ;Pulse 10us
                CALL    Vccp_off
                CALL    _Data_Free
_Next_2:        RET
Bit_test8       ENDP
;-------------------------------------------------------------------
_Data_Prog      PROC    NEAR
                PUSH    DX
                MOV     DX,0302H
                OUT     DX,AL
                POP     DX
                RET
_Data_Prog      ENDP
;-------------------------------------------------------------------
_CE_Pulse       PROC    NEAR
                PUSH    DX
;Set CE Active low
                MOV     AL,01000000B    ;AL   = 40H
                MOV     DX,0305H        ;Led  = on
                OUT     DX,AL           ;Vccp = 8.75V
                                        ;CE   = 0
                CALL    Time_10us
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;Set CE Active hight
                MOV     AL,11000000B    ;AL   = C0H
                OUT     DX,AL           ;Led  = on
                                        ;Vccp = 8.75V
                                        ;CE   = 1
                POP     DX
                RET
_CE_Pulse       ENDP
;-------------------------------------------------------------
Vccp_off        PROC    NEAR
                PUSH    DX
                MOV     AL,11000001B    ;AL  = C1H
                MOV     DX,0305H        ;Led = on,Vccp = 5V
                OUT     DX,AL           ;CE  = 1
                POP     DX
                RET
Vccp_off        ENDP
;-------------------------------------------------------------
Time_10us       PROC    NEAR
                RET
Time_10us       ENDP
;-------------------------------------------------------------
_Data_Free      PROC    NEAR
                PUSH    DX
                MOV     AL,00
                MOV     DX,0302H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                OUT     DX,AL
                POP     DX
                RET
_Data_Free      ENDP
;-------------------------------------------------------------
Vccp_on         PROC    NEAR
                PUSH    DX
                MOV     AL,11000000B    ;AL   = C0H
                MOV     DX,0305H        ;CE   = 1
                OUT     DX,AL           ;Led  = on
                POP     DX              ;Vccp = 8.75V
                RET
Vccp_on         ENDP
;-------------------------------------------------------------
Init_prom       PROC    NEAR
                PUSH    DX
                MOV     AL,10000001B    ;AL   = 81H
                MOV     DX,0305H        ;Vccp = 5V
                OUT     DX,AL           ;CE   = 1
                POP     DX              ;Led  = off
                RET
Init_prom       ENDP
;-------------------------------------------------------------
Disp_length     PROC    NEAR
                CALL    Row3_Col65
                MOV     AL,_length
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CMP     AL,01H
                JNE     Jump1
                Display length_123
                JMP     Jump4
Jump1:          CMP     AL,02H
                JNE     Jump2
                Display length_129
                JMP     Jump4
Jump2:          CMP     AL,03H
                JNE     Jump3
                Display length_131
                JMP     Jump4
Jump3:          CMP     AL,04H
                JNE     Jump4
                Display length_185
Jump4:          RET
Disp_length ENDP
;-------------------------------------------------------------
;Verify prom with memory buffer
_Verify         PROC    NEAR
                PUSH    CX
                MOV     CX,_Counter
                GotoXY  18,00
                Display _data_msg3
                CALL    Cursor_out
                CALL    Relay_on
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            CALL    CE_ready
Z3:         CALL    read_key
            CMP     AL,0DH          ;check return key
            JNE     Z3
            CALL    led_on
            MOV     BX,00
            MOV     DX,00
R1:         MOV     AL,DL
            CALL    Addr_LSB
            MOV     AL,DH
            CALL    Addr_MSB
            CALL    read_data
            MOV     _Result,AL
            MOV     AL,[HEX_buffer+BX]
            CMP     _Result,AL
            JNE     _error5
            INC     DX
            INC     BX
            CALL    Delay1
            LOOP    R1
            CALL    led_off
            GotoXY  18,00
            Display _data_msg4
            JMP     back1
_error5:    CALL    Beep
            CALL    led_off
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                GotoXY    18,00
                Display   _data_msg5
back1:          CALL      Cursor_out
                CALL      read_key
                CALL      Relay_off
                POP       CX
                RET
_Verify         ENDP
;-------------------------------------------------------------
prom_pos        PROC      NEAR
;Set cursor position at Current type =
;at row 03 col 17
                GotoXY    03,17
                RET
prom_pos        ENDP
;-------------------------------------------------------------
disp_menu       PROC      NEAR
                Display   main_menu
;Set cursor position at Command ?
;at row 19 col 14
                GotoXY    19,14
                RET
disp_menu       ENDP
;-------------------------------------------------------------
ctrl_word       PROC      NEAR
                MOV       AL,80H          ;Set 8255 #7
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     DX,0303H        ;Port A,B,C is all O/P
                OUT     DX,AL
                MOV     AL,90H          ;Set 8255 #8
                MOV     DX,0307H        ;Port A is I/P and
                OUT     DX,AL           ;Port B,C is O/P
                RET
ctrl_word       ENDP
;-------------------------------------------------------------
Relay_on        PROC    NEAR
                MOV     AL,11111111B    ;AL = FFH
                MOV     DX,0306H
                OUT     DX,AL
                RET
Relay_on        ENDP
;-------------------------------------------------------------
Relay_off       PROC    NEAR
                MOV     AL,00000000B    ;AL = 00H
                MOV     DX,0306H
                OUT     DX,AL
                RET
Relay_off       ENDP
;-------------------------------------------------------------
Led_on          PROC    NEAR
                MOV     AL,01000001B    ;AL  = 41H
                MOV     DX,0305H        ;Vccp = 5V
                OUT     DX,AL           ;CE   = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                RET                             ;Led   = on
Led_on          ENDP
;-------------------------------------------------------------
Led_off         PROC    NEAR
                MOV     AL,10000001B    ;AL    = 81H
                MOV     DX,0305H        ;Vccp  = 5V
                OUT     DX,AL           ;CE    = 1
                RET                     ;Led   = off
Led_off         ENDP
;-------------------------------------------------------------
CE_low          PROC    NEAR
                MOV     AL,01000001B    ;AL    = 41H
                MOV     DX,0305H        ;Vccp  = 5V
                OUT     DX,AL           ;CE    = 0
                RET                     ;Led   = on
CE_low          ENDP
;-------------------------------------------------------------
Addr_LSB        PROC    NEAR
                PUSH    DX
                MOV     DX,0300H
                OUT     DX,AL
                POP     DX
                RET
Addr_LSB        ENDP
;-------------------------------------------------------------
Addr_MSB        PROC    NEAR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                PUSH    DX
                MOV     DX,0301H
                OUT     DX,AL
                POP     DX
                RET
Addr_MSB        ENDP
;-------------------------------------------------------------
Read_data       PROC    NEAR
                PUSH    DX
                MOV     DX,0304H
                IN      AL,DX
                POP     DX
                RET
Read_data       ENDP
;-------------------------------------------------------------
;Generate sound Beep 1 time
beep            PROC    NEAR
                PUSH    AX
                PUSH    DX
                MOV     AH,02
                MOV     DL,07
                INT     21H
                POP     DX
                POP     AX
                RET
beep            ENDP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------------------
Delay1      PROC    NEAR
            PUSH    CX
            MOV     CX,0080H
LP:         LOOP    LP
            POP     CX
            RET
Delay1      ENDP
;-------------------------------------------------------------
Delay2      PROC    NEAR
            PUSH    CX
            PUSH    BX
            MOV     BX,0010H
LP2:        MOV     CX,0FFFH
LP1:        LOOP    LP1
            DEC     BX
            JNZ     LP2
            POP     BX
            POP     CX
            RET
Delay2      ENDP
;-------------------------------------------------------------
CE_Ready    PROC    NEAR
            PUSH    DX
            MOV     AL,00000001B    ;AL   = 01H
            MOV     DX,0305H        ;Vccp = 5V
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                OUT     DX,AL           ;CE   = 0
                POP     DX              ;Led  = off
                RET
CE_Ready        ENDP
;-----------------------------------------------------------------
;Blank Prom Checker
Blank_ck4       PROC    NEAR
                PUSH    CX
                GotoXY  18,00
;Display insert prom message
                Display blank_msg1
                CALL    cursor_out
                CALL    Relay_on
                CALL    CE_Ready
yet1:           CALL    read_key
                CMP     AL,0DH          ;key RETURN
                JNE     yet1
                MOV     CX,_Counter     ;Loop Counter
                MOV     BX,0000H        ;Base Address
                CALL    Led_on
next1:          MOV     AL,BL
                CALL    Addr_LSB
                MOV     AL,BH
                CALL    Addr_MSB
                CALL    read_data
                CMP     AL,11110000B    ;if AL=F0H mean blank
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JNE     _error1
                INC     BX
                CALL    Delay1
                LOOP    next1
_ok1:           CALL    blank_ck1       ;check blank ok
                JMP     next2
_error1:        CALL    blank_ck2       ;check blank error
next2:          POP     CX
                RET
Blank_ck4       ENDP
;-------------------------------------------------------------
;Blank Prom Checker
Blank_ck8       PROC    NEAR
                PUSH    CX
                GotoXY  18,00
;Display insert prom message
                Display blank_msg1
                CALL    cursor_out
                CALL    Relay_on
                CALL    CE_Ready
yet2:           CALL    read_key
                CMP     AL,0DH          ;key RETURN
                JNE     yet2
                MOV     CX,_Counter     ;Loop Counter
                MOV     BX,0000H        ;Base Address
                CALL    Led_on
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
next3:      MOV     AL,BL
            CALL    Addr_LSB
            MOV     AL,BH
            CALL    Addr_MSB
            CALL    read_data
            CMP     AL,00000000B    ;if AL=00H mean blank
            JNE     _error4
            INC     BX
            CALL    Delay1
            LOOP    next3
_ok2:       CALL    blank_ck1       ;check blank ok
            JMP     next4
_error4:    CALL    blank_ck2       ;check blank error
next4:      POP     CX
            RET
Blank_ck8   ENDP
;-------------------------------------------------------------
blank_ck1   PROC    NEAR
;Blank Check 'OK'
            GotoXY  18,00
            CALL    Led_off
            Display blank_msg2
            CALL    Cursor_out
            CALL    read_key
            CALL    Relay_off
            CALL    ClrScr4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            RET
blank_ck1   ENDP
;-------------------------------------------------------------
blank_ck2   PROC    NEAR
;Blank Check 'ERROR'
            CALL    beep
            GotoXY  18,00
            CALL    Led_off
            Display blank_msg3
            CALL    Cursor_out
            CALL    read_key
            CALL    Relay_off
            CALL    ClrScr4
            RET
blank_ck2   ENDP
;-------------------------------------------------------------
prom_type   PROC    NEAR
;Display type of bipolar prom
            Display prom_select
;Set cursor position at Select type = ?
;at row 16 col 20
            GotoXY  16,20
Accept:     CALL    read_key
            CMP     AL,31H          ;key NO. 1
            JNE     Accept1
            CALL    prom_pos
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Display  prom1
            CALL     Row3_Col65
            Display  Length_000
            MOV      _Contain,00H
            MOV      _Prom_type,01H
            MOV      _length,01H
            MOV      _Counter,0020H
            MOV      _size,08H
            JMP      finish
Accept1:    CMP      AL,32H          ;key NO. 2
            JNE      Accept2
            CALL     prom_pos
            Display  prom2
            CALL     Row3_Col65
            Display  Length_000
            MOV      _Contain,00H
            MOV      _Prom_type,02H
            MOV      _length,01H
            MOV      _Counter,0020H
            MOV      _size,08H
            JMP      finish
Accept2:    CMP      AL,33H          ;key NO. 3
            JNE      Accept3
            CALL     prom_pos
            Display  prom3
            CALL     Row3_Col65
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            Display  Length_000
            MOV      _Contain,00H
            MOV      _Prom_type,03H
            MOV      _length,02H
            MOV      _Counter,0100H
            MOV      _size,04H
            JMP      finish
Accept3:    CMP      AL,34H          ;key NO. 4
            JNE      Accept4
            CALL     prom_pos
            Display  prom4
            CALL     Row3_Col65
            Display  Length_000
            MOV      _Contain,00H
            MOV      _Prom_type,04H
            MOV      _length,02H
            MOV      _Counter,0100H
            MOV      _size,04H
            JMP      finish
Accept4:    CMP      AL,35H          ;key NO. 5
            JNE      Accept5
            CALL     prom_pos
            Display  prom5
            CALL     Row3_Col65
            Display  Length_000
            MOV      _Contain,00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     _Prom_type,05H
                MOV     _length,03H
                MOV     _Counter,0200H
                MOV     _size,04H
                JMP     finish
Accept5:        CMP     AL,36H          ;key NO. 6
                JNE     Accept6
                CALL    prom_pos
                Display prom6
                CALL    Row3_Col65
                Display Length_000
                MOV     _Contain,00H
                MOV     _Prom_type,06H
                MOV     _length,03H
                MOV     _Counter,0200H
                MOV     _size,04H
                JMP     finish
Accept6:        CMP     AL,37H          ;key NO. 7
                JNE     Accept7
                CALL    prom_pos
                Display prom7
                CALL    Row3_Col65
                Display Length_000
                MOV     _Contain,00H
                MOV     _Prom_type,07H
                MOV     _length,02H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     _Counter,0100H
                MOV     _size,08H
                JMP     finish
Accept7:        CMP     AL,38H          ;key NO. 8
                JNE     Accept8
                CALL    prom_pos
                Display prom8
                CALL    Row3_Col65
                Display Length_000
                MOV     _Contain,00H
                MOV     _Prom_type,08H
                MOV     _length,03H
                MOV     _Counter,0200H
                MOV     _size,08H
                JMP     finish
Accept8:        CMP     AL,39H          ;key NO. 9
                JNE     Accept9
                CALL    prom_pos
                Display prom9
                CALL    Row3_Col65
                Display Length_000
                MOV     _Contain,00H
                MOV     _Prom_type,09H
                MOV     _length,04H
                MOV     _Counter,0800H
                MOV     _size,04H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                JMP     finish
Accept9:        JMP     Accept
finish:         RET
prom_type       ENDP
;-----------------------------------------------------------
;Set cursor at row 03 col 65
Row3_Col65      PROC    NEAR
                GotoXY  03,65
                RET
Row3_Col65      ENDP
;-----------------------------------------------------------
ClrScr1         PROC    NEAR
;Clear all of the screen & set cursor position at (R0 C0)
                MOV     AH,6            ;Scroll page up function
                MOV     AL,0            ;Number of line
                MOV     CH,0            ;Row NO.0
                MOV     CL,0            ;Col NO.0
                                        ;Upper left
                MOV     DH,24           ;Row NO.24
                MOV     DL,79           ;Col NO.79
                                        ;Lower right
                MOV     BH,7            ;Attribute normal
                INT     10H             ;Call Bios
                MOV     AH,2            ;Set cursor position
                MOV     DH,0            ;Row NO.0
                MOV     DL,0            ;Col NO.0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                MOV     BH,0            ;Page 0
                INT     10H             ;Call Bios
                RET                     ;Return back
ClrScr1         ENDP                    ;End of subroutine
;-------------------------------------------------------------
ClrScr2         PROC    NEAR
;Clear_Screen under head of MAIN MENU:
                MOV     AH,6            ;Scroll page up function
                MOV     AL,0            ;Number of line
                MOV     CH,5            ;Row NO.5
                MOV     CL,0            ;Col NO.0
                                        ;Upper left
                MOV     DH,24           ;Row NO.24
                MOV     DL,79           ;Col NO.79
                                        ;Lower right
                MOV     BH,7            ;Attribute normal
                INT     10H             ;Call Bios
                MOV     AH,2            ;Set cursor position
                MOV     DH,5            ;Row NO.5
                MOV     DL,0            ;Col NO.0
                MOV     BH,0            ;Page 0
                INT     10H             ;Call bIOS
                RET                     ;Return back
ClrScr2         ENDP                    ;End of subroutine
;-------------------------------------------------------------
Clrscr3         PROC    NEAR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;Clear screen under Command ?
            MOV     AH,06
            MOV     AL,00
            MOV     CH,21           ;Row 21
            MOV     CL,00           ;Col 00
                                    ;Upper left
            MOV     DH,24           ;Row 24
            MOV     DL,79           ;Col 79
                                    ;Lower right
            MOV     BH,07           ;Attribute normal
            INT     10H
;Set cursor position at row 18,col 14
            GotoXY  18,14
            RET
ClrScr3     ENDP
;-------------------------------------------------------------------
Clrscr4     PROC    NEAR
            MOV     AH,06
            MOV     AL,00
            MOV     CH,18           ;Row 18
            MOV     CL,00           ;Col 00
                                    ;Upper left
            MOV     DH,24           ;Row 24
            MOV     DL,79           ;Col 79
                                    ;Lower right
            MOV     BH,07           ;Attribute normal
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                INT       10H
 ;Set cursor position at row 19,col 00
                GotoXY    19,00
                Display   Command
 ;Set cursor position at row 19,col 14
                GotoXY    19,14
                CALL      Selection
                RET
 ClrScr4        ENDP
 ;-------------------------------------------------------------
 ;Read the character from keyboard buffer
 read_key       PROC      NEAR
                MOV       AH,0              ;Read keyboard function
                INT       16H               ;Call Bios
 ;Value in AL is a ASCII code
 ;If no character in buffer'this function is wait until keypressed
 ;Convert any low case to upper case
                CMP       AL,'a'
                JB        _not              ;Not Convert
                CMP       AL,'z'
                JA        _not              ;Not Convert
                SUB       AL,'a'-'A'        ;Convert to upper case
 _not:          RET                         ;Return back
 Read_key       ENDP
 CODE           ENDS
 ;-------------------------------------------------------------
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DATA        SEGMENT
HEX_buffer  DB    4096 DUP (?)
_Save       DB    ?
_Result     DB    ?
_Prom_type  DB    ?                    ;type of bi_prom
_length     DB    ?                    ;length of bi_prom 2 byte
_Contain    DB    ?
_handle     DW    ?
_Counter    DW    ?                    ;load CX to _Counter
_size       DB    ?                    ;Prom size 4 - 8 bit
_filename   DB    15,17 DUP (0)
messege     DB    'NO File(s)','$'
files       DB    0
filecount   DB    0
ten         DB    10
fcb         DB    1,'???????????'      ;drive#,wile card
            DB    25  DUP (?)
buffer      DB    128 DUP (?)
Crlf        DB    13,10,'$'
blank       DB    '     $'
blank1      DB    '         $'
Command     DB    '     Command ?$'
Drive1      DB    'Volume in drive A is BPROM_PROG',13,10
            DB    'Directory of  A:\',13,10
            DB    '    ',13,10,'$'
Drive2      DB    'Volume in drive B is BPROM_PROG',13,10
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DB      'Directory of  B:\',13,10
                DB      '    ',13,10,'$'
Drive3          DB      'Volume in drive C is BPROM_PROG',13,10
                DB      'Directory of  C:\',13,10
                DB      '    ',13,10,'$'
key_press       DB      '         ',13,10,13,10
                DB      'Type any key to continue !','$'
blank_msg1      DB      '                    ',13,10
                DB      'Insert PROM into socket then press (RETURN)'
blank_msg2      DB      '                    ',13,10
                DB      ' Result:  BLANK CHECK      - OK -    ',13,10
                DB      '          ',13,10
                DB      '    Type any key to continue !','$'
blank_msg3      DB      '                    ',13,10
                DB      'Result:  BLANK CHECK    - ERROR -  ',13,10
                DB      '          ',13,10
                DB      '    Type any key to continue !','$'
_data_msg1      DB      '                    ',13,10
                DB      'Insert original PROM and then press (RETURN)'
_data_msg2      DB      '                    ',13,10
                DB      'Result:  READ PROM DATA  - OK -  ',13,10
                DB      '    ',13,10
                DB      '    Type any key to continue !','$'
_data_msg3      DB      '                    ',13,10
                DB      'Insert PROM VERIFY and then press (RETURN)'
_data_msg4      DB      '                    ',13,10
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DB      'Result: PROM VERIFY    - OK -    ',13,10
                DB      '       ',13,10
                DB      '  Type any key to continue !','$'
                _data_msg5  DB      '                      ',13,10
                DB      'Result: PROM VERIFY  - ERROR -   ',13,10
                DB      '       ',13,10
                DB      '    Type any key to continue !','$'
_data_msg6      DB      '                         ',13,10
                DB      'Insert PROM to socket then press (RETURN)'
_data_msg7      DB      '                         ',13,10
                DB      'Result: PROM PROGRAM    - OK -    ',13,10
                DB      '       ',13,10
                DB      '    Type any key to continue !','$'
_data_msg8      DB      '                         ',13,10
                DB      'Result: PROM PROGRAM  - ERROR -  ',13,10
                DB      '       ',13,10
                DB      '    Type any key to continue !','$'
Programing      DB      'Now !   Programing...            ','$'
_output_msg     DB      '       ',13,10
                DB      ' SAVE BUFFER DATA TO DISK FILE',13,10
                DB      '       ',13,10
                DB      'BUFFER STARTING AT = (3000:0000)',13,10
                DB      '       ',13,10
                DB      '    ENTER FILE NAME = ','$'
                _input_msg  DB      '       ',13,10
                DB      '    LOAD DISK FILE TO BUFFER',13,10
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DB      '     ',13,10
                DB      'BUFFER STRATING AT = (3000:0000)',13,10
                DB      '     ',13,10
                DB      '     ENTER FILE NAME = ','$'
_Disk_fail1     DB      '     ',13,10
                DB      ' Result:  CANNOT SAVE FILE ',13,10
                DB      '     ',13,10
                DB      ' Type any key to continue !','$'
_Disk_fail2     DB      '     ',13,10
                DB      '     Result:  FIND NOT FOUND    ',13,10
                DB      '     ',13,10
                DB      '  Type any key to continue !','$'
_load_ok        DB      '     ',13,10
                DB      'Result:  LOAD FILE  - OK -',13,10
                DB      '     ',13,10
                DB      '  Type any key to continue !','$'
_save_ok        DB      '     ',13,10
                DB      '  Result:  SAVE FILE  - OK - ',13,10
                DB      '     ',13,10
                DB      'Type any key to continue !','$'
Intro           DB      'Memory buffer starting address at 3000:0000'
                DB      '-----------------------------------------'
_Shell_msg      DB      'Type EXIT to return to BPROM Programing... '
Filename1       DB      'COMMAND.COM',0
Filename2       DB      'DEBUG.COM',0
Para_block      Label Word
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DW      0
                DW      offset Cmd_buf
PR1             DW      ?
                DW      5CH
PR2             DW      ?
                DW      6CH
PR3             DW      ?
Cmd_buf         DB      ?
                DB      ' '
Keep_ss         DW      0
Keep_sp         DW      0
header1         DB      'THE INDUSTRIAL TECHNOLOGY COMPUTER OF KMITL'
                DB      'BIPOLAR PROM PROGRAMMER CARD ON THE IBM PC '
                DB      '          ',13,10,'$'
header2         DB      'Current type =   Use Card NO.   buffer length ='
                DB      '-----------------------------------------------'
main_menu       DB      '          > > > >   MAIN MENU   < < < <',13,10
                DB      '    ',13,10
                DB      '[D] = DIR -----Directory',13,10
                DB      '[R] = READ ----into buffer',13,10
                DB      '[W] = WRITE----bipolar prom',13,10
                DB      '[B] = BLANK ---check',13,10
                DB      '[V] = VERIFY --with buffer',13,10
                DB      '[T] = TYPE ----of bipolar prom',13,10
                DB      '[L] = LOAD ----Disk file to buffer',13,10
                DB      '[S] = SAVE ----buffer to Disk file',13,10
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                DB      '[O] = OS ------Shell',13,10
                DB      '[M] = MODIFY --buffer',13,10
                DB      '[Q] = QUIT ----to MS-DOS',13,10
                DB      '       ',13,10
                DB      '       Command ?','$'
prom_select     DB      '         ',13,10
                DB      '[1] = N82S23F   ( 0032 X 8 bit )',13,10
                DB      '[2] = N82S123F  ( 0032 X 8 bit )',13,10
                DB      '[3] = N82S126F  ( 0256 X 4 bit )',13,10
                DB      '[4] = N82S129F  ( 0256 X 4 bit )',13,10
                DB      '[5] = N82S130F  ( 0512 X 4 bit )',13,10
                DB      '[6] = N82S131F  ( 0512 X 4 bit )',13,10
                DB      '[7] = N82S135F  ( 0256 X 8 bit )',13,10
                DB      '[8] = N82S147F  ( 0512 X 8 bit )',13,10
                DB      '[9] = N82S185F  ( 2048 X 4 bit )',13,10
                DB      '          ',13,10
                DB      '  Select type = ?','$'
prom1           DB      'N82S23F      Use Card NO.6','$'
prom2           DB      'N82S123F     Use Card NO.6','$'
prom3           DB      'N82S126F     Use Card NO.1','$'
prom4           DB      'N82S129F     Use Card NO.1','$'
prom5           DB      'N82S130F     Use Card NO.5','$'
prom6           DB      'N82S131F     Use Card NO.5','$'
prom7           DB      'N82S135F     Use Card NO.3','$'
prom8           DB      'N82S147F     Use Card NO.2','$'
prom9           DB      'N82S185F     Use Card NO.4','$'
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Length_000   DB     '0000','$'
Length_123   DB     '0032','$'
Length_129   DB     '0256','$'
Length_131   DB     '0512','$'
Length_185   DB     '2048','$'
DATA         ENDS
;-------------------------------------------------------------
STACK        SEGMENT  STACK
DW        100 DUP(?)
TOP          DW       0
STACK        ENDS
END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

วิจารณ์และสรุป

ในบทนี้จะกล่าวถึงผลงาน และประสิทธิภาพของเครื่องโปรแกรมพรอม ปัญหาจากการออกแบบและทดลอง ตลอดจนแนวทางในการที่จะพัฒนาให้ดีขึ้นต่อไป ซึ่งจะแยกกล่าวโดยสรุปเป็นเรื่องๆ ดังนี้

5.1 ขีดความสามารถของเครื่องโปรแกรมพรอม

สามารถอ่านข้อมูลจากตัวพรอม และเขียนข้อมูลเข้าตัวพรอมขนาดต่างๆได้ตั้งแต่ 32 x 8 บิท จนถึงขนาด 2048 x 4 บิท ซึ่งสามารถเลือก ชนิดและเบอร์ต่างๆ ได้มากมายหลายเบอร์ จนสามารถกล่าวได้ว่าครอบคลุมสำหรับการใช้งานทั่วไปอย่างพอเพียง และจากการทดสอบการใช้งานของเครื่องโปรแกรมปรากฏว่า ทำงานได้ผลดีตามเป้าหมายที่กำหนด

5.2 ปัญหา และ อุปสรรค

ในการทำงานใดๆก็ตาม ย่อมต้องพบกับปัญหาและอุปสรรคบ้าง เว้นแต่จะมากหรือน้อยเท่านั้น การดำเนินการของงานวิจัยนี้ก็เช่นกัน แต่ก็ได้พยายามแก้ไขให้ผ่านอุปสรรคต่างๆไปสู่ความสำเร็จได้ด้วยดี ซึ่งปัญหาและอุปสรรคนั้น พอจะยกมากล่าวโดยสรุปได้ คือ

5.2.1 ปัญหาที่เกี่ยวกับข้อมูลของพรอม เนื่องจากข้อมูลต่างๆของพรอมที่มีอยู่ตามแหล่งวิชาการและท้องตลาด ยังมีอยู่น้อยมากไม่เพียงพอที่จะนำมาประกอบการออกแบบวงจรทางด้านฮาร์ดแวร์ได้ จึงแก้ปัญหาโดยการขอข้อมูลต่างๆ เหล่านั้นจากเจ้าหน้าที่ของบริษัทผู้ผลิต เช่น คู่มือของไบโพลาร์พรอม จากบริษัทซิกเนติกส์ ซึ่งก็ได้รับความร่วมมือและให้การอนุเคราะห์เป็นอย่างดี ทำให้ได้ข้อมูลเพิ่มเติมมากพอที่จะค้นคว้าและดำเนินงานต่อไปได้

5.2.2 เนื่องจากลักษณะทางกายภาพของพรอม มีความแตกต่างกันมากเช่น จำนวนขนาดความจุ ตำแหน่งของขาต่างๆ ตลอดจนรูปร่าง(Package) ของแต่ละ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บริษัทหรือ ของพรอมแต่ละเบอร์ ก็แตกต่างกันไป ซึ่งเป็นปัญหาทำให้เกิดความยุ่งยากในการออกแบบ จึงต้องแก้ไขด้วยการออกแบบโมดุลการ์ด (Module Card) เพื่อทำการสลับขาข้อมูลและตำแหน่งต่างๆที่ป้อนให้แก่พรอมให้ถูกต้องตามคู่มือของบริษัทผู้ผลิต

5.2.3 ในการออกแบบลายแผ่นวงจรพิมพ์ของเท็กทูลส์ (Textools) ด้วยสมาร์ทเวิร์ค ปรากฏว่าขาของดีบี-25 (DB-25) ไม่เข้ากับมาตรฐานของขาไอซี จึงไม่สามารถใช้สมาร์ทเวิร์คออกแบบได้ ต้องแก้ไขโดยการออกแบบด้วยมือในส่วนที่เป็นขาของดีบี-25

5.2.4 การพิมพ์ลายวงจรลงกระดาษปริ้นท์เตอร์ด้วยสมาร์ทเวิร์ค เพื่อถ่ายฟิล์ม ก่อนทำปริ้นท์ ปรากฏว่ากระดาษปริ้นท์เตอร์ ยืดและหดตัวตามอุณหภูมิที่เปลี่ยนแปลงได้ เป็นผลให้ตำแหน่งของโดนัท (Donut) ของปริ้นท์ทั้งสองด้าน ไม่ตรงกันจึงต้องแก้ไขด้วยการใช้กระดาษถ่ายเอกสาร เบอร์ เอ.3 แทนกระดาษปริ้นท์เตอร์

5.2.5 ในภาคแหล่งจ่ายไฟตรง(D.C.Supply)ของไอบีเอ็ม/พีซี ไม่มีแรงดัน 16 โวลท์ ซึ่งจำเป็นต้องใช้ในช่วง Vccp ของการโปรแกรมพรอม จึงต้องออกแบบสร้างวงจรยกระดับแรงดัน (Step Up Switching Regulator) ขึ้นเองโดยใช้ไอซี เบอร์ ทีแอล. 497 ทำการรักษาระดับแรงดัน 16 โวลท์ ที่ 250 มิลลิแอมป์

5.3 แนวทางในการพัฒนาการสร้างเครื่องโปรแกรม

5.3.1 งานวิจัยนี้สามารถใช้เป็นแนวในการพัฒนาการสร้างเครื่องโปรแกรมพรอม ที่มีขนาดและจำนวนบิทที่สูงมากขึ้น ในโอกาสต่อไป

5.3.2 ข้อมูลต่างๆจากงานวิจัยนี้ สามารถนำไปใช้เป็นพื้นฐานในการศึกษาค้นคว้าเพิ่มเติม เพื่อการออกแบบสร้างเครื่องโปรแกรมอุปกรณ์หน่วยความจำชนิดอื่นๆ ที่มีความยุ่งยากซับซ้อนมากขึ้นได้ เช่น การออกแบบสร้างเครื่องอ่านและเขียน PAL ซึ่งกำลังเริ่มมีบทบาท และได้รับความนิยมอย่างแพร่หลายในปัจจุบัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Signetics

82S23
82S123
256-Bit TTL Bipolar PROM

Product Specification

Bipolar Memory Products

DESCRIPTION

The 82S23 and 82S123 are field programmable, which means that custom patterns are immediately available by following the Signetics Generic I fusing procedure. The 82S23 and 82S123 devices are supplied with all outputs at logical Low. Outputs are programmed to a logic High level at any specified address by fusing a Ni-Cr link matrix.

These devices include on-chip decoding and 1 chip enable input for memory expansion. They feature either Open collector or Three-state outputs for optimization of word expansion in bused organizations.

Ordering information can be found on the following pages.

The 82S23 and 82S2123 devices are also processed to military requirements for operation over the military temperature range. For specifications and ordering information consult the Signetics Military Data Book.

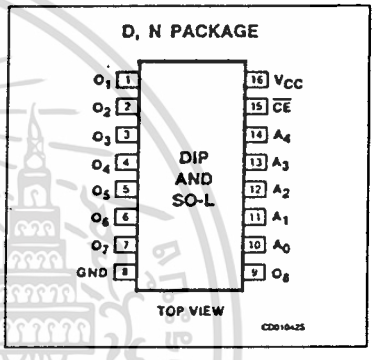
FEATURES

- Address access time: 50ns max
- Power dissipation: 1.3mW/bit typ
- Input loading: −100μA max
- On-chip address decoding
- One chip enable input
- Output options:
  - N82S23: Open collector
  - N82S123: Three-state
- No separate fusing pins
- Unprogrammed outputs are Low level
- Fully TTL compatible

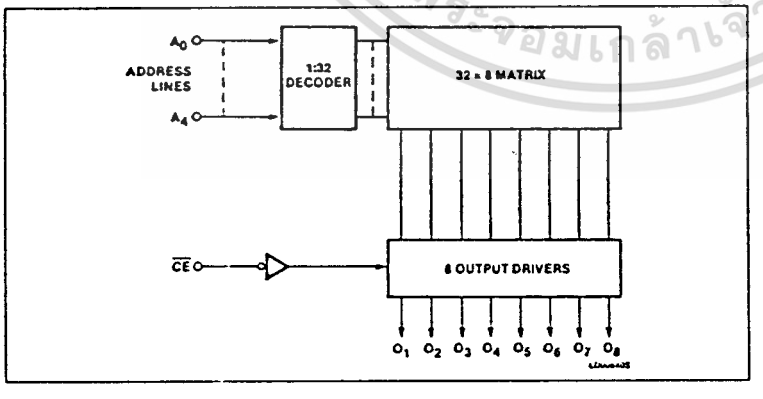
APPLICATIONS

- Prototyping/volume production
- Sequential controllers
- Format conversion
- Hardwired algorithms
- Random logic
- Code conversion

PIN CONFIGURATION



BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ORDERING CODE

DESCRIPTION	ORDER CODE
Plastic Dual Inline 300mil wide 16-pin	N82S23 N • N82S123 N
Plastic Small Outline 300mil wide 16-pin	N82S23 D • N82S123 D

ABSOLUTE MAXIMUM RATINGS

	PARAMETER	RATING	UNIT
V_{CC}	Supply voltage	+7	V_{dc}
V_{IN}	Input voltage	+5.5	V_{dc}
V_{OH} V_O	Output voltage High (82S23) Off-state (82S123)	+5.5	V_{dc}
T_A T_{STG}	Temperature range Operating Storage	0 to +75 −65 to +150	°C

DC ELECTRICAL CHARACTERISTICS $0°C \leq T_A \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

PARAMETER		TEST CONDITIONS[1,2]	LIMITS			UNIT
			Min	Typ[5]	Max	
Input voltage						
V_{IL}	Low	V_{CC} = 4.75V			0.8	V
V_{IH}	High	V_{CC} = 5.25V	2.0			
V_{IC}	Clamp	I_{IN} = −12mA			−1.2	
Output voltage		$\overline{CE}$ = Low				
V_{OL}	Low	I_{OUT} = 16mA			0.45	V
V_{OH}	High	I_{OUT} = −2mA	2.4			
Input current						
I_{IL}	Low	V_{IN} = 0.45V			−100	μA
I_{IH}	High	V_{IN} = 5.5V			50	
Output current						
I_{OLK}	Leakage (82S23)	$\overline{CE}$ = High, V_{OUT} = 5.5V			40	μA
I_{OZ}	Hi-Z State (82S123)	$\overline{CE}$ = High, V_{OUT} = 5.5V			40	
		$\overline{CE}$ = High, V_{OUT} = 0.5V			−40	
I_{OS}	Short circuit (82S123)[3]	$\overline{CE}$ = Low, V_{OUT} = 0V, High stored	−15		−90	mA
Supply current						
I_{CC}		V_{CC} = 5.25V			96	mA
Capacitance		$\overline{CE}$ = High, V_{CC} = 5.0V				
C_{IN}	Input	V_{IN} = 2.0V		5		pF
C_{OUT}	Output	V_{OUT} = 2.0V		8		

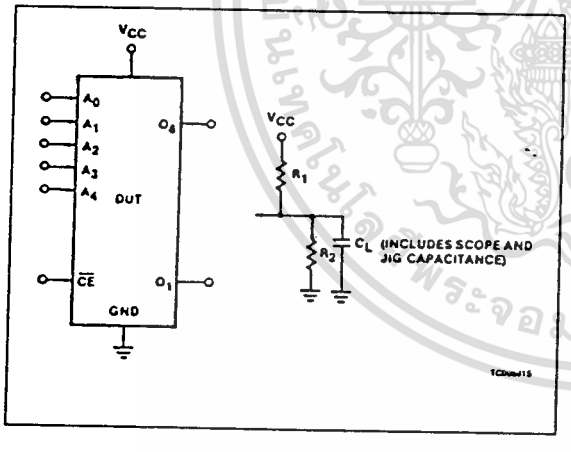
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC ELECTRICAL CHARACTERISTICS $R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$, $0°C \leq T_A \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

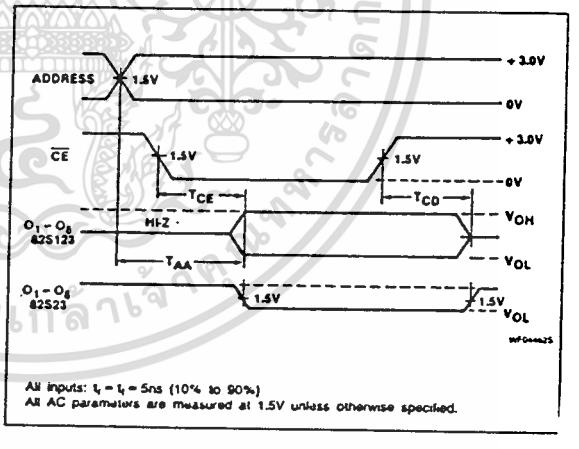
PARAMETER	TO	FROM	LIMITS			UNIT
			Min	Typ[5]	Max	
Access time[4]						
T_{AA}	Output	Address		45	50	ns
T_{CE}	Output	Chip enable			35	
Disable time[6]						
T_{CD}	Output	Chip enable			35	ns

NOTES:
1. Positive current is defined as into the terminal referenced.
2. All voltages with respect to network ground terminal.
3. Duration of short circuit should not exceed 1 second.
4. Tested at an address cycle time of 1μsec.
5. Typical values are at $V_{CC} = 5V$, $T_A = 25°C$.
6. Measured at a delta of 0.5V from Logic Level with $R_1 = 750\Omega$, $R_2 = 750\Omega$ and $C_L = 5pF$.

TEST LOAD CIRCUIT



VOLTAGE WAVEFORM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Signetics

82S126
82S129
1K-Bit TTL Bipolar PROM

Product Specification

Bipolar Memory Products

DESCRIPTION

The 82S126 and 82S129 are field programmable, which means that custom patterns are immediately available by following the Signetics Generic I fusing procedure. The 82S126 and 82S129 devices are supplied with all outputs at logical Low. Outputs are programmed to a logic High level at any specified address by fusing the Ni-Cr link matrix.

These devices include on-chip decoding and 2 chip enable inputs for ease of memory expansion. They feature either Open collector or Three-state outputs for optimization of word expansion in bused organizations.

Ordering information can be found on the following page.

The 82S126 and 82S129 devices are also processed to military requirements for operation over the military temperature range. For specifications and ordering information consult the Signetics Military Data Book.

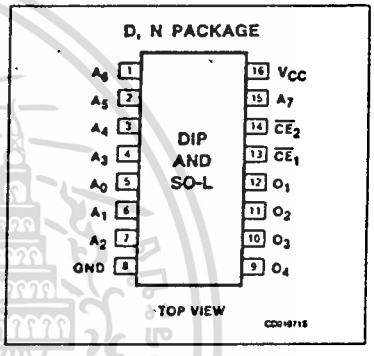
FEATURES

- Address access time: 50ns max
- Power dissipation: 0.5mW/bit typ
- Input loading: −100μA max
- On-chip address decoding
- Two chip enable inputs
- Output options:
  - N82S126: Open collector
  - N82S129: Three-state
- No separate fusing pins
- Unprogrammed outputs are Low level
- Fully TTL compatible

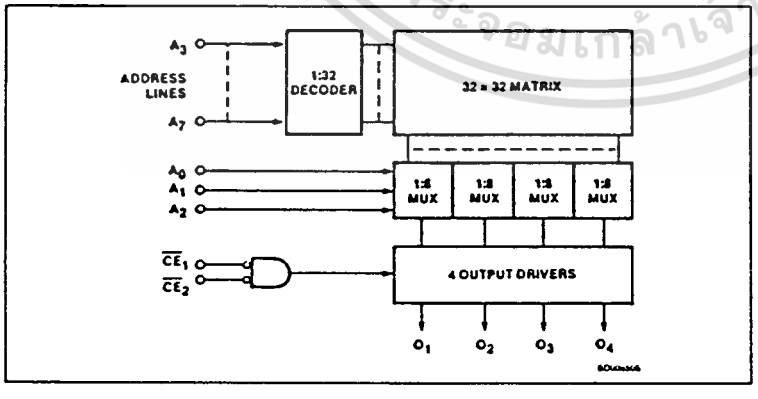
APPLICATIONS

- Prototyping/volume production
- Sequential controllers
- Microprogramming
- Hardwired algorithms
- Control store
- Random logic
- Code conversion

PIN CONFIGURATION



BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ORDERING CODE

DESCRIPTION	ORDER CODE
Plastic Dual Inline 300mil wide 16-pin	N82S126 N • N82S129 N
Plastic Small Outline 300mil wide 16-pin	N82S126 D • N82S129 D

ABSOLUTE MAXIMUM RATINGS

	PARAMETER	RATING	UNIT
V_{CC}	Supply voltage	+7	V_{dc}
V_{IN}	Input voltage	+5.5	V_{dc}
	Output voltage		
V_{OH}	High (82S126)	+5.5	V_{dc}
V_O	Off-state (82S129)	+5.5	
	Temperature range		
T_A	Operating	0 to +75	°C
T_{STG}	Storage	−65 to +150	

DC ELECTRICAL CHARACTERISTICS $0°C \leq T_A \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

PARAMETER		TEST CONDITIONS[1,2]	LIMITS			UNIT
			Min	Typ[5]	Max	
Input voltage						
V_{IL}	Low				0.8	V
V_{IH}	High		2.0			
V_{IC}	Clamp	$I_{IN} = -12mA$			−1.2	
Output voltage		$\overline{CE}_{1,2}$ = Low				
V_{OL}	Low	$I_{OUT} = 16mA$			0.45	V
V_{OH}	High (82S129)	$I_{OUT} = -2.0mA$	2.4			
Input current						
I_{IL}	Low	$V_{IN} = 0.45V$			−100	μA
I_{IH}	High	$V_{IN} = 5.5V$			40	
Output current						
I_{OLK}	Leakage (82S126)	$\overline{CE}_1$ or $\overline{CE}_2$ = High, $V_{OUT} = 5.5V$			40	μA
I_{OZ}	Hi-Z State (82S129)	$\overline{CE}_1$ or $\overline{CE}_2$ = High, $V_{OUT} = 5.5V$			40	
		$\overline{CE}_1$ or $\overline{CE}_2$ = High, $V_{OUT} = 0.5V$			−40	
I_{OS}	Short circuit (82S129)[3]	$\overline{CE}_{1,2}$ = Low, $V_{OUT} = 0V$, Stored High	−15		−70	mA
Supply current						
I_{CC}		$V_{CC} = 5.25V$			120	mA
Capacitance		$\overline{CE}_1$ or $\overline{CE}_2$ = High, $V_{CC} = 5.0V$				
C_{IN}	Input	$V_{IN} = 2.0V$		5		pF
C_{OUT}	Output	$V_{OUT} = 2.0V$		8		

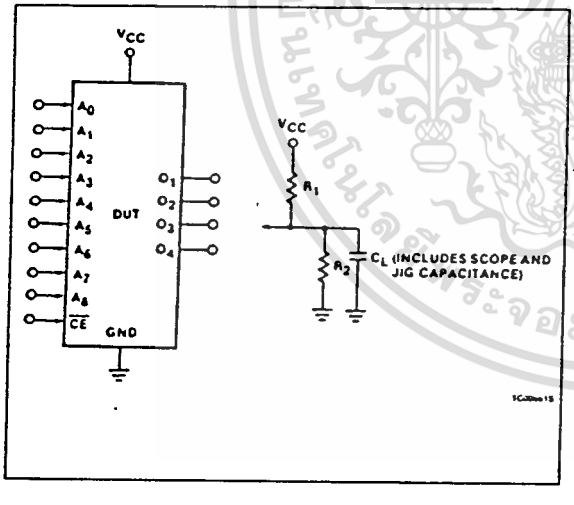
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC ELECTRICAL CHARACTERISTICS $R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$, $0°C \leq T_A \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

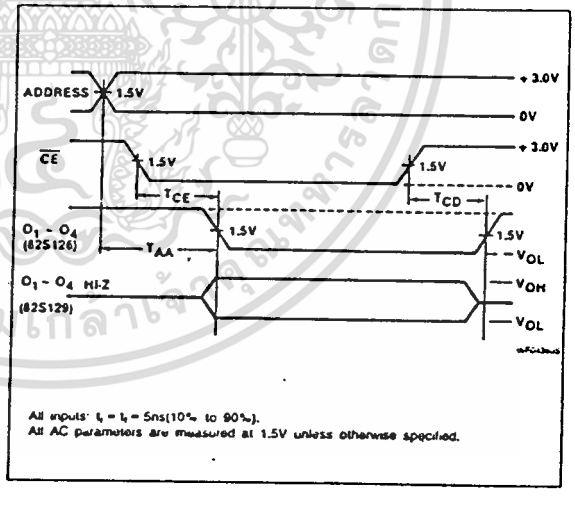
PARAMETER	TO	FROM	MIN	TYP[5]	MAX	UNIT
Access time[4]						
T_{AA}	Output	Address		40	50	ns
T_{CE}	Output	Chip enable			25	
Disable time[6]						
T_{CD}	Output	Chip disable			25	ns

NOTES:
1. Positive current is defined as into the terminal referenced.
2. All voltages with respect to network ground.
3. Duration of short circuit should not exceed 1 second.
4. Tested at an address cycle time of 1μsec.
5. Typical values are at $V_{CC} = 5V$, $T_A = 25°C$.
6. Measured at a delta of 0.5V from Logic Level with $R_1 = 750\Omega$, $R_2 = 750\Omega$ and $C_L = 5pF$

TEST LOAD CIRCUIT



VOLTAGE WAVEFORM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Signetics

82S130
82S131
2K-Bit TTL Bipolar PROM

Product Specification

Bipolar Memory Products

DESCRIPTION

The 82S130 and 82S131 are field programmable, which means that custom patterns are immediately available by following the Signetics Generic I fusing procedure. The standard 82S130 and 82S131 are supplied with all outputs at logical Low. Outputs are programmed to a logic High level at any specified address by fusing the Ni-Cr link matrix.

These devices include on-chip decoding and 1 chip enable input for ease of memory expansion. They feature either Open collector or Three-state outputs for optimization of word expansion in bused organizations.

Ordering information can be found on the folowing page.

The 82S130 and 82S131 devices are also processed to military requirements for operation over the military temperature range. For specifications and ordering information consult the Signetics Military Data Book.

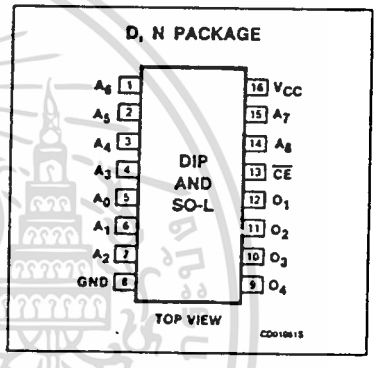
FEATURES

- Address access time: 50ns max
- Power dissipation: 0.3mW/bit typ
- Input loading: −100μA max
- On-chip address decoding
- One chip enable input
- Output options:
  - N82S130: Open collector
  - N82S131: Three-state
- No separate fusing-pins
- Unprogrammed outputs are Low level
- Fully TTL compatible

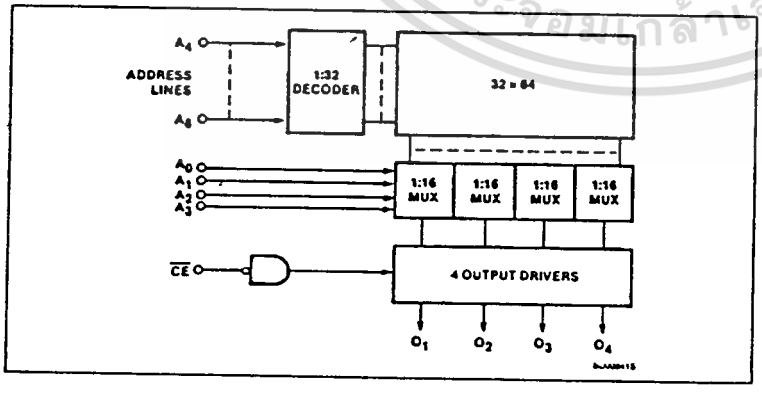
APPLICATIONS

- Prototyping/volume production
- Sequential controllers
- Microprogramming
- Hardwired algorithms
- Control store
- Random logic
- Code conversion

PIN CONFIGURATION



BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ORDERING CODE

DESCRIPTION	ORDER CODE
Plastic Dual Inline 300mil wide 16-pin	N82S130 N • N82S131 N
Plastic Small Outline 300mil wide 16-pin	N82S130 D • N82S131 D

ABSOLUTE MAXIMUM RATINGS

PARAMETER		RATING	UNIT
V_{CC}	Supply voltage	+7	V_{dc}
V_{IN}	Input voltage	+5.5	V_{dc}
	Output voltage		
V_{OH}	High (82S130)	+5.5	V_{dc}
V_O	Off-state (83S131)	+5.5	
	Temperature range		
T_A	Operating	0 to +75	°C
T_{STG}	Storage	−65 to +150	

DC ELECTRICAL CHARACTERISTICS $0°C \leq T_A \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

PARAMETER		TEST CONDITIONS[1,2]	LIMITS			UNIT
			Min	Typ[5]	Max	
Input voltage						
V_{IL}	Low			0.8		V
V_{IH}	High		2.0			
V_{IC}	Clamp	$I_{IN} = -12mA$			−1.2	
Output voltage		$\overline{CE}$ = Low				
V_{OL}	Low	$I_{OUT} = 16mA$			0.45	V
V_{OH}	High (82S131)	$I_{OUT} = -2mA$	2.4			
Input current						
I_{IL}	Low	$V_{IN} = 0.45V$			−100	μA
I_{IH}	High	$V_{IN} = 5.5V$			40	
Output current						
I_{OLK}	Leakage (82S130)	$\overline{CE}$ = High, $V_{OUT} = 5.5V$,			40	μA
I_{OZ}	Hi-Z State (82S131)	$\overline{CE}$ = High, $V_{OUT} = 5.5V$			40	
		$\overline{CE}$ = High, $V_{OUT} = 0.5V$			−40	
I_{OS}	Short circuit (82S131)[3]	$\overline{CE}$ = Low, V_{OUT} = 0V, Stored High	−15		−70	mA
Supply current						
I_{CC}		$V_{CC} = 5.25V$			140	mA
Capacitance		$\overline{CE}$ = High, $V_{CC} = 5.0V$				
C_{IN}	Input	$V_{IN} = 2.0V$		5		pF
C_{OUT}	Output	$V_{OUT} = 2.0V$		8		

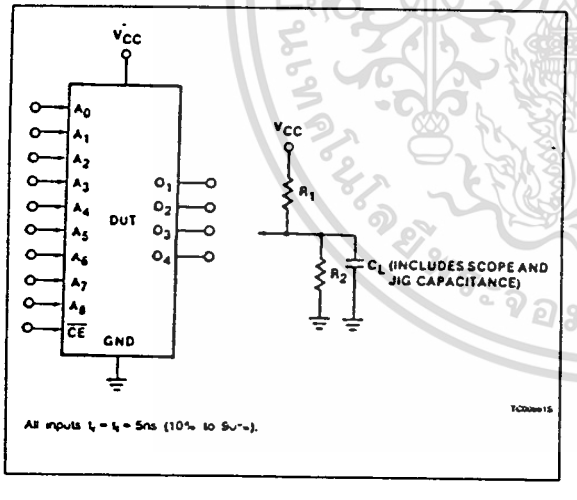
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC ELECTRICAL CHARACTERISTICS $R_1 = 470\Omega$, $R_2 = 1k\Omega$, $C_L = 30pF$, $0°C \leq T_A \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

PARAMETER	TO	FROM	LIMITS			UNIT
			Min	Typ[5]	Max	
Access time[4]						
T_{AA}	Output	Address			50	ns
T_{CE}	Output	Chip enable			30	
Disable time[6]						
T_{CD}	Output	Chip disable			30	ns

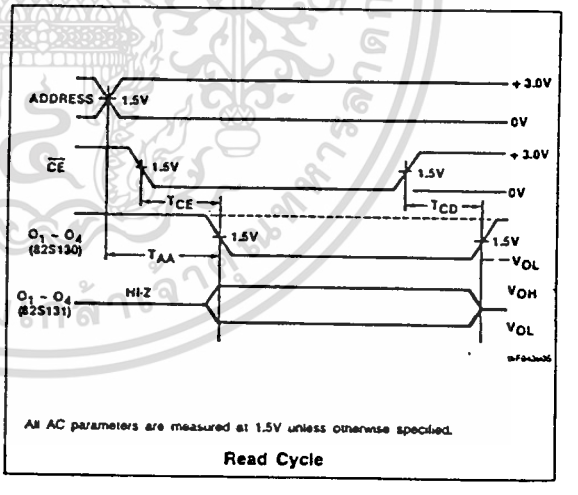
NOTES:
1. Positive current is defined as into the terminal referenced.
2. All voltages with respect to network ground.
3. Duration of short circuit should not exceed 1 second.
4. Tested at an address cycle time of 1 μsec.
5. Typical values are at $V_{CC} = 5V$, $T_A = 25°C$.
6. Measured at a delta of 0.5V from Logic Level with $R_1 = 750\Omega$, $R_2 = 750\Omega$ and $C_L = 5pF$

TEST LOAD CIRCUIT



All inputs $t_r = t_f = 5ns$ (10% to 90%).

VOLTAGE WAVEFORM



All AC parameters are measured at 1.5V unless otherwise specified.

Read Cycle

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Signetics

82S135
2K-Bit TTL Bipolar PROM

Product Specification

Bipolar Memory Products

DESCRIPTION

The 82S135 is field-programmable, which means that custom patterns are immediately available by following the Signetics Generic I fusing procedure. The standard devices are supplied with all outputs at logical Low. Outputs are programmed to a logic High level at any specified address by fusing the Ni-Cr link matrix.

The 82S135 includes on-chip decoding and two chip enable inputs for ease of memory expansion, and features Three-state outputs for optimization of word expansion in bused organizations.

Ordering information can be found on the following page.

The 82S135 devices are also processed to military requirements for operation over the military temperature range. For specifications and ordering information consult the Signetics Military Data Book.

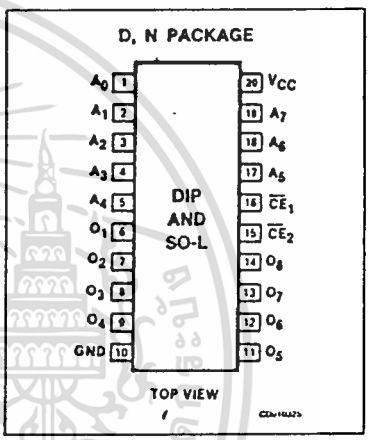
FEATURES

- Address access time: 45ns max
- Power dissipation: 329μW/bit typ
- Input loading: −100μA max
- Two chip enable inputs
- On-chip address decoding
- No separate fusing pins
- Fully TTL compatible
- Outputs: Three-state
- Unprogrammed outputs are Low level

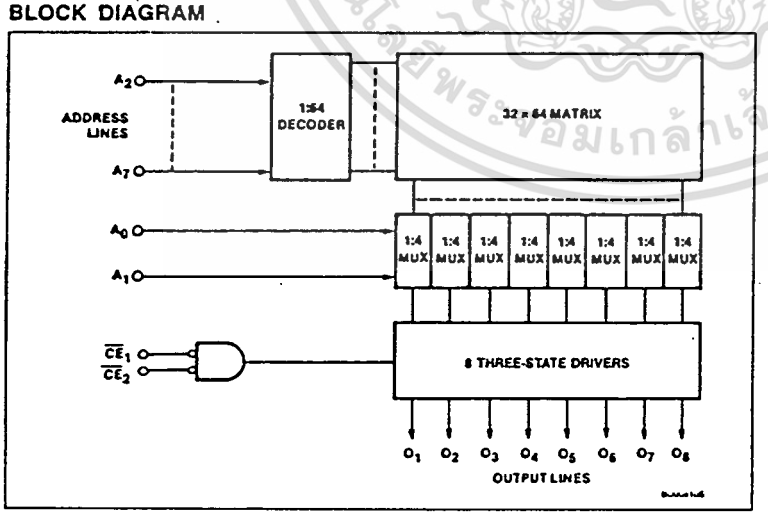
APPLICATIONS

- Prototyping/volume production
- Sequential controllers
- Microprogramming
- Hardwired algorithms
- Control store
- Random logic
- Code conversion

PIN CONFIGURATION



BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2K-Bit TTL Bipolar PROM (256 x 8) — 82S135

ORDERING CODE

DESCRIPTION	ORDER CODE
Plastic Dual Inline 300mil wide 20-pin	N82S135 N
Plastic Small Outline 300mil wide 20-pin	N82S135 D

ABSOLUTE MAXIMUM RATINGS

	PARAMETER	RATING	UNIT
V_{CC}	Power supply voltage	+7	V_{dc}
V_{IN}	Input voltage	+5.5	V_{dc}
V_O	Output voltage Off-state	+5.5	V_{dc}
	Temperature range		
T_A	Operating	0 to +75	°C
T_{STG}	Storage	−65 to +150	

DC ELECTRICAL CHARACTERISTICS $0°C \leq T_A \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

PARAMETER	TEST CONDITIONS[1,2]	LIMITS Min	LIMITS Typ[5]	LIMITS Max	UNIT
Input voltage					
V_{IL} Low	V_{CC} = 4.75V			.80	V
V_{IH} High	V_{CC} = 5.25V	2.0			
V_{IC} Clamp	I_{IN} = −12mA			−1.2	
Output voltage					
V_{OL} Low	I_{OUT} = 9.6mA			.50	V
V_{OH} High	$\overline{CE}_1, \overline{CE}_2$ = Low, I_{OUT} = −2mA, High stored	2.4			
Input current					
I_{IL} Low	V_{IN} = 0.45V			100	μA
I_{IH} High	V_{IN} = 5.5V			40	
Output current					
I_{OZ} Hi-Z State	$\overline{CE}_1, \overline{CE}_2$ = High, V_{OUT} = 0.5V			−40	μA
	$\overline{CE}_1, \overline{CE}_2$ = High, V_{OUT} = 5.5V			40	
I_{OS} Short circuit[3]	$\overline{CE}_1, \overline{CE}_2$ = Low, V_{OUT} = 0V, One Stored	−15		−75	mA
Supply current					
I_{CC}	V_{CC} = 5.25V		135	150	mA
Capacitance	V_{CC} = 5.0V, $\overline{CE}$ = High				
C_{IN} Input	V_{IN} = 2.0V		5		pF
C_{OUT} Output	V_{OUT} = 2.0V		8		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

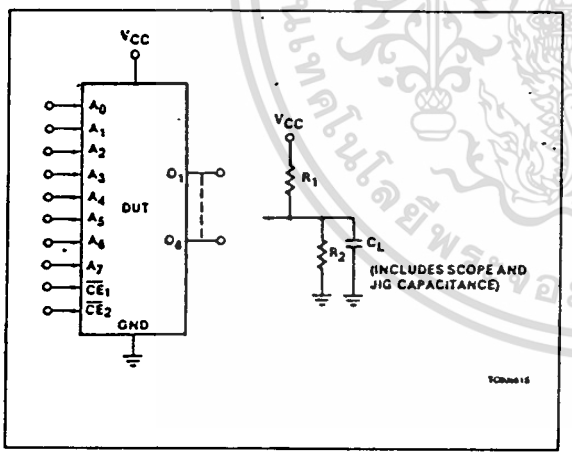
2K-Bit TTL Bipolar PROM (256 x 8) — 82S135

AC ELECTRICAL CHARACTERISTICS $R_1 = 270\Omega$, $R_2 = 600\Omega$, $C_L = 30pF$, $0°C \leq T_A \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

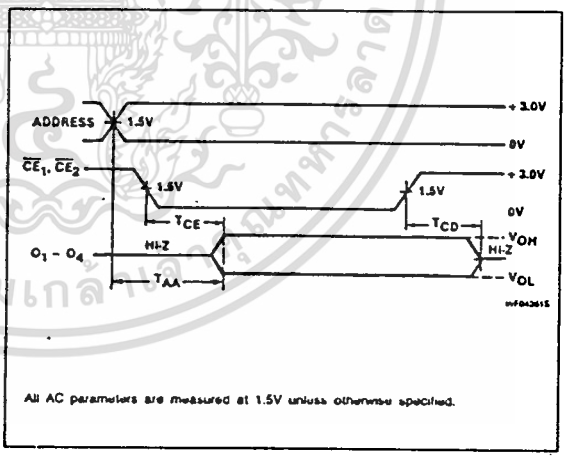
PARAMETER	TO	FROM	LIMITS			UNIT
			Min	Typ[5]	Max	
Access time[4]						
T_{AA}	Output	Address		40	45	ns
T_{CE}	Output	Chip enable		20	25	
Disable time[6]						
T_{CD}	Output	Chip disable		20	35	ns

NOTES:
1. Positive current is defined as into the terminal referenced.
2. All voltages with respect to network ground.
3. Duration of short circuit should not exceed 1 second.
4. Tested at an address cycle time of 1μsec.
5. Typical values are at $V_{CC} = 5V$, $T_A = 25°C$.
6. Measured at a delta of 0.5V from Logic Level with $R_1 = 750\Omega$, $R_2 = 750\Omega$ and $C_L = 5pF$.

TEST LOAD CIRCUIT



VOLTAGE WAVEFORM



All AC parameters are measured at 1.5V unless otherwise specified.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Signetics

82S147
82S147A
4K-Bit TTL Bipolar PROM

Product Specification

Bipolar Memory Products

DESCRIPTION

The 82S147 and 82S147A are field-programmable, which means that custom patterns are immediately available by following the Signetics generic f fusing procedure. The standard devices are supplied with all outputs at logical Low. Outputs are programmed to a logic High level at any specified address by fusing the Ni-Cr link matrix.

The 82S147 and 82S147A includes on-chip decoding and one chip enable input for ease of memory expansion, and features Three-state outputs for optimization of word expansion in bused organizations.

Ordering information can be found on the following page.

The 82S147 and 82S147A devices are also processed to military requirements for operation over the military temperature range. For specifications and ordering information consult the Signetics Military Data Book.

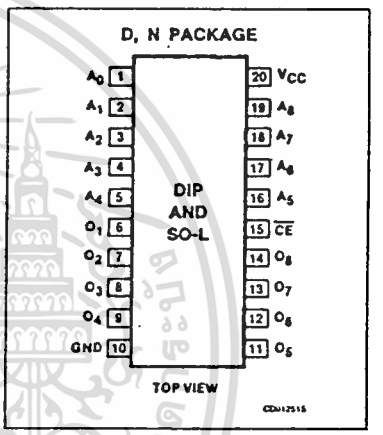
FEATURES

- Address access time:
  - N82S147: 60ns max
  - N82S147A: 45ns max
- Power dissipation: 625mW typ
- Input loading: −100μA max
- One chip enable input
- On-chip address decoding
- No separate fusing pins
- Fully TTL compatible
- One chip enable input
- Outputs: Three-state
- Unprogrammed outputs are Low level

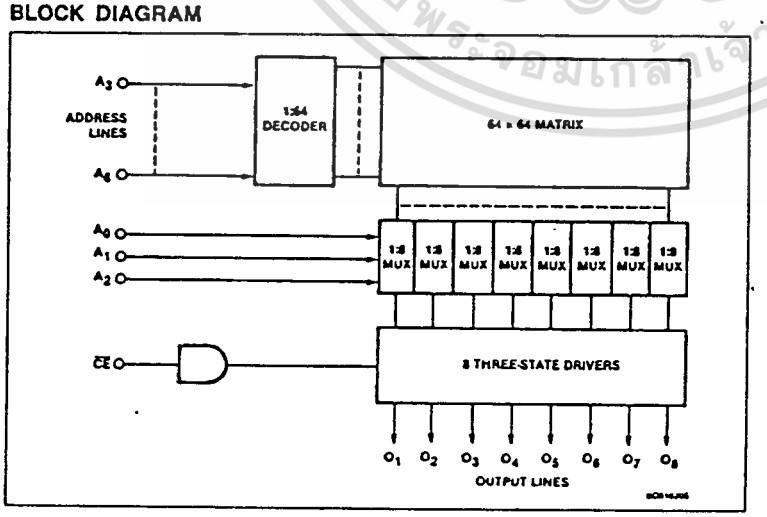
APPLICATIONS

- Prototyping/volume production
- Sequential controllers
- Microprogramming
- Hardwired algorithms
- Control store
- Random logic
- Code conversion

PIN CONFIGURATION



BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ORDERING CODE

PACKAGES	ORDER CODES
Plastic Dual Inline 300mil wide 20-pin	N82S147 N • N82S147A N
Plastic Small Outline 300mil wide 20-pin	N82S147 D • N82S147A D

ABSOLUTE MAXIMUM RATINGS

	PARAMETER	RATING	UNIT
V_{CC}	Power supply voltage	+7	V_{dc}
V_{IN}	Input voltage	+5.5	V_{dc}
V_O	Output voltage Off-state	+5.5	V_{dc}
	Temperature range		
T_A	Operating	0 to +75	°C
T_{STG}	Storage	−65 to +150	

DC ELECTRICAL CHARACTERISTICS $0°C \leq T_A \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

PARAMETER		TEST CONDITIONS[1,2]	LIMITS			UNIT
			Min	Typ[5]	Max	
Input voltage						
V_{IL}	Low				0.8	V
V_{IH}	High		2.0			
V_{IC}	Clamp	$I_{IN} = -12mA$		−0.8	−1.2	
Output voltage		$\overline{CE}$ = Low				
V_{OL}	Low	$I_{OUT} = 9.6mA$			0.45	V
V_{OH}	High	$I_{OUT} = -2mA$	2.4			
Input current						
I_{IL}	Low	$V_{IN} = 0.45V$			−100	μA
I_{IH}	High	$V_{IN} = 5.5V$			40	
Output current						
I_{OZ}	Hi-Z State	$\overline{CE}$ = High, $V_{OUT} = 5.5V$			40	μA
		$\overline{CE}$ = High, $V_{OUT} = 0.5V$			−40	
I_{OS}	Short-circuit[3]	$\overline{CE}$ = Low, $V_{OUT} = 0V$	−15		−70	mA
Supply current						
I_{CC}		$V_{CC} = 5.25V$		125	155	mA
Capacitance		$\overline{CE}$ = High, $V_{CC} = 5.0V$				
C_{IN}	Input	$V_{IN} = 2.0V$		5		pF
C_{OUT}	Output	$V_{OUT} = 2.0V$		8		

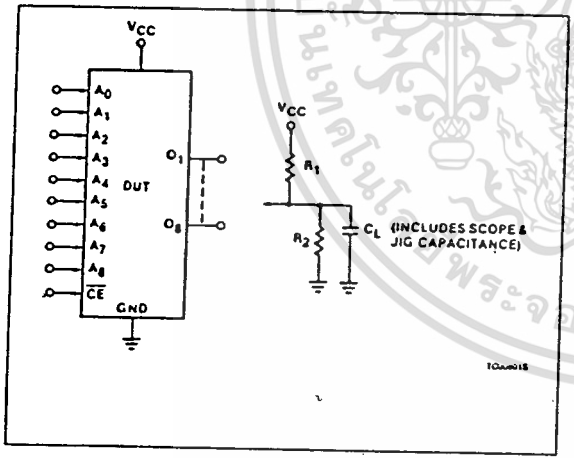
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC ELECTRICAL CHARACTERISTICS $R_1 = 270\Omega$, $R_2 = 600\Omega$, $C_L = 30pF$, $0°C \leq T_A \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

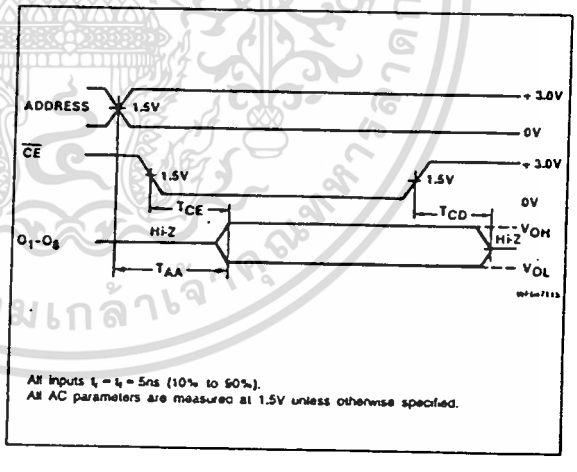
PARAMETER	TO	FROM	N82S147			N82S147A			UNIT
			Min	Typ[5]	Max	Min	Typ[5]	Max	
Access time[4]									
T_{AA}	Output	Address		45	60		40	45	ns
T_{CE}	Output	Chip Enable		20	35		20	30	
Disable time[6]									
T_{CD}	Output	Chip disable		20	35		20	30	ns

NOTES:
1. All voltage values are with respect to network ground terminal.
2. Positive current is defined as into the terminal referenced.
3. Duration of the short circuit should not exceed 1 second.
4. Tested at an address cycle time of 1μsec.
5. Typical values are at $V_{CC} = 5V$, $T_A = 25°C$.
6. Measured at a delta of 0.5V from Logic Level with $R_1 = 750\Omega$, $R_2 = 750\Omega$ and $C_L = 5pF$.

TEST LOAD CIRCUIT



VOLTAGE WAVEFORM



All inputs $t_r = t_f = 5ns$ (10% to 90%).
All AC parameters are measured at 1.5V unless otherwise specified.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Signetics

82S185
8K-Bit TTL Bipolar PROM

Product Specification

Bipolar Memory Products

DESCRIPTION

The 82S185 is field programmable, which means that custom patterns are immediately available by following the Signetics Generic I fusing procedure. The standard 82S185 is supplied with all outputs at logical Low. Outputs are programmed to a logic High level at any specified address by fusing the Ni-Cr link matrix.

This device includes on-chip decoding and 1 chip enable input for memory expansion. It features Three-state outputs for optimization of word expansion in bused organizations.

Ordering information can be found on the following page.

The 82S185 devices are also processed to military requirements for operation over the military temperature range. For specifications and ordering information consult the Signetics Military Data Book.

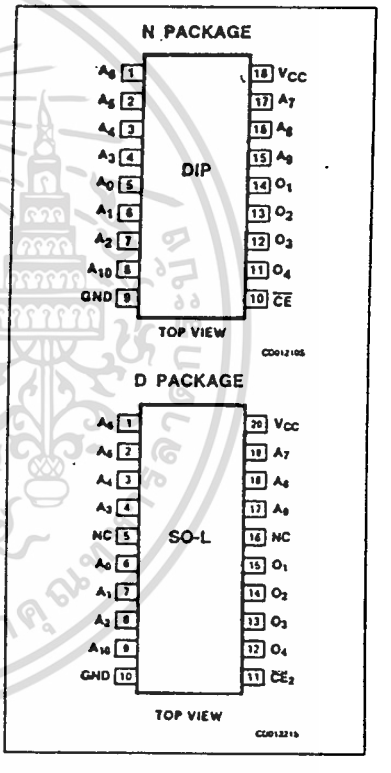
FEATURES

- Low power dissipation: 50μW/bit typ
- Address access time: 100ns max
- Input loading: −100μA max
- On-chip address decoding
- No separate fusing pins
- Unprogrammed outputs are Low level
- Fully TTL compatible
- One chip enable input
- Outputs: Three-state

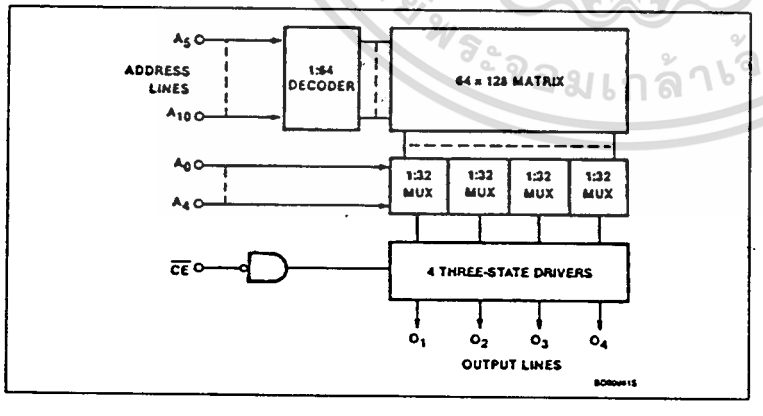
APPLICATIONS

- Sequential controllers
- Control store
- Random logic
- Code conversion

PIN CONFIGURATION



BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8K-Bit TTL Bipolar PROM (2048 x 4) — 82S185

ORDERING CODE

DESCRIPTION	ORDER CODE
Plastic Dual Inline 300mil wide 18-pin	N82S185 N
Plastic Small Outline 300mil wide 20-pin	N82S185 D

ABSOLUTE MAXIMUM RATINGS

	PARAMETER	RATING	UNIT
V_{CC}	Supply voltage	+7	V_{dc}
V_{IN}	Input voltage	+5.5	V_{dc}
V_O	Output voltage Off-state	+5.5	V_{dc}
T_A	Temperature range Operating	0 to +75	°C
T_{STG}	Storage	−65 to +150	

DC ELECTRICAL CHARACTERISTICS $0°C \leq T_A \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

PARAMETER	TEST CONDITIONS[1,2]	LIMITS Min	LIMITS Typ[5]	LIMITS Max	UNIT
Input voltage[1]					
V_{IL} Low				0.8	V
V_{IH} High		2.0			
V_{IC} Clamp	$I_{IN} = -12mA$		−0.8	−1.2	
Output voltage[1]	$\overline{CE}$ = Low				
V_{OL} Low	$I_{OUT} = 16mA$			0.45	V
V_{OH} High	$I_{OUT} = -2mA$	2.4			
Input current[2]					
I_{IL} Low	$V_{IN} = 0.45V$			−100	μA
I_{IH} High	$V_{IN} = 5.5V$			40	
Output current					
I_{OZ} Hi-Z State	$\overline{CE}$ = High, $V_{OUT} = 0.5V$,			−40	μA
	$\overline{CE}$ = High, $V_{OUT} = 5.5V$			40	
I_{OS} Short circuit[3]	$\overline{CE}$ = Low, V_{OUT} = 0V High stored	−15		−70	mA
Supply current					
I_{CC}	$V_{CC} = 5.25V$		90	120	mA
Capacitance	$\overline{CE}$ = High, $V_{CC} = 5.0V$				
C_{IN} Input	$V_{IN} = 2.0V$		5		pF
C_{OUT} Output	$V_{OUT} = 2.0V$		8		

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

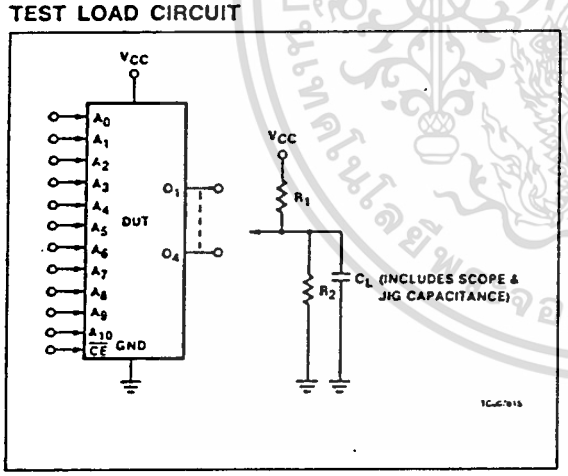
8K-Bit TTL Bipolar PROM (2048 x 4) — 82S185

AC ELECTRICAL CHARACTERISTICS $R_1 = 270\Omega$, $R_2 = 600\Omega$, $C_L = 30pF$, $0°C \leq T_A \leq +75°C$, $4.75V \leq V_{CC} \leq 5.25V$

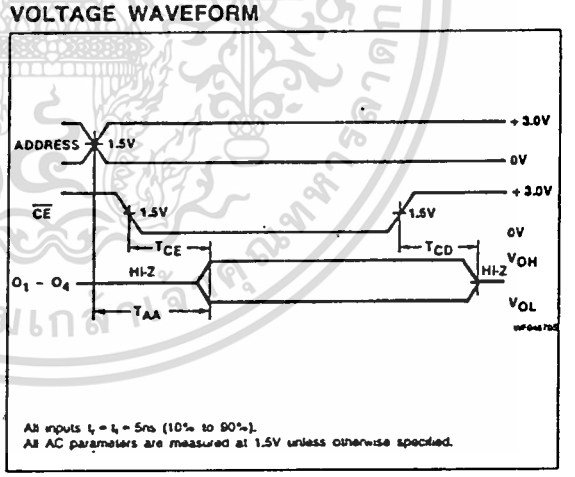
PARAMETER	TO	FROM	LIMITS			UNIT
			Min	Typ[5]	Max	
Access time[4]						
T_{AA}	Output	Address		70	100	ns
T_{CE}	Output	Chip Enable		30	40	
Disable time[6]						
T_{CD}	Output	Chip disable		30	40	ns

NOTES:
1. All voltage values are with respect to network ground terminal.
2. Positive current is defined as into the terminal referenced.
3. Duration of the short circuit should not exceed 1 second.
4. Tested at an address cycle time of 1μsec.
5. All typical values are at $V_{CC} = 5V$, $T_A = 25°C$.
6. Measured at a delta of 0.5V from Logic Level with $R_1 = 750\Omega$, $R_2 = 750\Omega$ and $C_L = 5pF$.

TEST LOAD CIRCUIT



VOLTAGE WAVEFORM



All inputs $t_r = t_f = 5ns$ (10% to 90%).
All AC parameters are measured at 1.5V unless otherwise specified.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

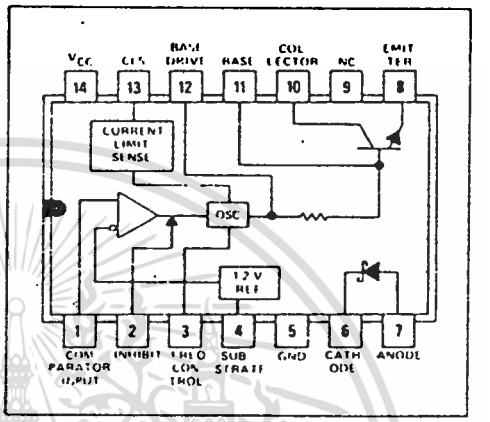
LINEAR INTEGRATED CIRCUITS

TYPES TL497AM, TL497AI, TL497AC SWITCHING VOLTAGE REGULATORS

BULLETIN NO. DL-S 12422, JUNE 1976 REVISED SEPTEMBER 1977

- All Monolithic
- High Efficiency . . . 60% or Greater
- Output Current . . . 500 mA
- Input Current Limit Protection
- TTL Compatible Inhibit
- Adjustable Output Voltage
- Input Regulation . . . 0.2% Typ
- Output Regulation . . . 0.4% Typ
- Soft Start-up Capability

TL497AM . . . . J
TL497AI, TL497AC . . . . J OR N
DUAL-IN-LINE PACKAGE (TOP VIEW)



NC – No internal connection

description

The TL497A incorporates on a single monolithic chip all the active functions required in the construction of a switching voltage regulator. It can also be used as the control element to drive external components for high-power-output applications. The TL497A was designed for ease of use in step-up, step-down, or voltage inversion applications requiring high efficiency.

A block diagram of the TL497A is shown in the above pinout. The TL497A is a fixed-on-time variable-frequency switching voltage regulator control circuit. The on time is programmed by a single external capacitor connected between the frequency control pin and ground. This capacitor, C_T, is charged by an internal constant-current generator to a predetermined threshold. The charging current and the threshold vary proportionally with V_{CC}, thus the on time remains constant over the specified range of input voltage (5 to 12 volts). Typical on times for various values of C_T are as follows.

TIMING CAPACITOR, C_T (pF)	100	150	200	250	350	400	500	750	1000	1500	2000
ON-TIME (μs)	11	15	19	22	26	32	44	56	80	120	180

The output voltage is programmed by an external resistor ladder network (R1 and R2 in Figures 1, 2, and 3) that attenuates the desired output voltage to 1.2 volts. This feedback voltage is compared to the 1.2-volt reference by the high-gain comparator. When the output voltage decays below the programmed voltage, the comparator enables the oscillator circuit, which charges and discharges C_T as described above. The internal pass transistor is driven on during the charging of C_T. The internal transistor may be used directly for switching currents up to 500 milliampers. Its collector and emitter are uncommitted and it is current driven to allow operation from the positive supply voltage or ground. An internal Schottky diode matched to the current characteristics of the internal transistor is also available for blocking or commutating purposes. The TL479A also has on-chip current-limit circuitry that senses the peak currents in the switching regulator and protects the inductor against saturation and the pass transistor against overstress. The current limit is adjustable and is programmed by a single sense resistor, R_{CL}, connected between pin 14 and pin 13. The current-limit circuitry is activated when 0.7 volt is developed across R_{CL}. External gating is provided by the inhibit input. When the inhibit input is high, the output is turned off.

Simplicity of design is a primary feature of the TL497A. With only six external components (three resistors, two capacitors, and one inductor), the TL497A will operate in numerous voltage conversion applications (step-up, step-down, invert) with as much as 85% of the source power delivered to the load. The TL497A replaces the TL497 in all applications.

The TL497AM is characterized for operation over the full military temperature range of −55°C to 125°C, the TL497AI is characterized for operation from −25°C to 85°C, and the TL497AC from 0°C to 70°C.

TEXAS INSTRUMENTS
INCORPORATED
POST OFFICE BOX 9[illegible] • DALLAS, TEXAS 75222

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPES TL497AM, TL497AI, TL497AC
SWITCHING VOLTAGE REGULATORS

absolute maximum ratings over operating free-air temperature range (unless otherwise noted)

Input voltage, V_{CC} (see Note 1)	15 V
Output voltage	35 V
Comparator input voltage	5 V
Inhibit input voltage	5 V
Diode reverse voltage	35 V
Power switch current	750 mA
Diode forward current	750 mA
Continuous total dissipation at (or below) 25°C free-air temperature (see Note 2)	1000 mW
Operating free-air temperature range: TL497AM	−55°C to 125°C
TL497AI	−25°C to 85°C
TL497AC	0°C to 70°C
Storage temperature range	−65°C to 150°C
Lead temperature 1/16 inch from case for 60 seconds: J package	300°C
Lead temperature 1/16 inch from case for 10 seconds: N package	260°C

NOTES: 1. All voltage values except diode voltages are with respect to network ground terminal.
2. For operation above 25°C free-air temperature, refer to Dissipation Derating Curves, Figure 1 and Figure 11, page 10.

recommended operating conditions

	MIN	MAX	UNIT
Input voltage, V_I	4.5	12	V
Output voltage: step-up configuration (see Figure 2)	$V_I + 2$	30	V
step-down configuration (see Figure 3)	V_{ref}	V_I − 1	V
negative regulator (see Figure 4)	$-V_{ref}$	−25	V
Power switch current		500	mA
Diode forward current		500	mA

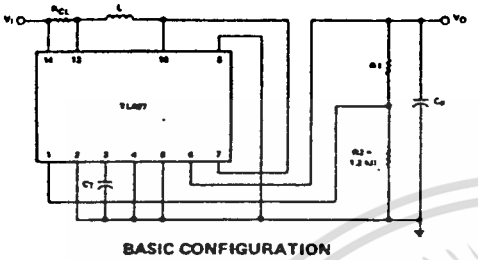
electrical characteristics at specified free-air temperature, V_I = 6 V (unless otherwise noted)

PARAMETER	TEST CONDITIONS†		TL497AM, TL497AI			TL497AC			UNIT
			MIN	TYP‡	MAX	MIN	TYP‡	MAX	
High-level inhibit input voltage		25°C	3			2.5			V
Low-level inhibit input voltage		25°C			0.6			0.8	V
High-level inhibit input current	$V_{I(I)}$ = 5 V	Full range		0.8	1.5		0.8	1.5	mA
Low-level inhibit input current	$V_{I(I)}$ = 0 V	Full range		5	20		5	10	μA
Comparator reference voltage	V_I = 4.5 V to 6 V	Full range	1.14	1.20	1.26	1.08	1.20	1.32	V
Comparator input bias current	V_I = 6 V	Full range		40	100		40	100	μA
Switch on-state voltage	V_I = 4.5 V, I_O = 100 mA	25°C		0.13	0.2		0.13	0.2	V
	I_O = 500 mA	Full range			1			0.85	
Switch off-state current	V_I = 4.5 V, V_O = 30 V	25°C		10	50		10	50	μA
		Full range			500			200	
Current-limit sense voltage	V_{CC} = 6 V	25°C	0.45		1	0.45		1	V
Diode forward voltage	I_O = 10 mA	Full range		0.75	0.95		0.75	0.85	V
	I_O = 100 mA	Full range		0.9	1.1		0.9	1	
	I_O = 500 mA	Full range		1.33	1.75		1.33	1.55	
Diode reverse voltage	I_O = 500 μA	Full range	30						V
	I_O = 200 μA	Full range				30			
On-state supply current		25°C		11	14		11	14	mA
		Full range			16			15	
Off-state supply current		25°C		6	9		6	9	mA
		Full range			11			10	

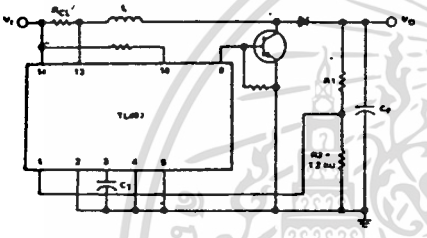
†Full range for TL497AM is −55°C to 125°C, for TL497AI is −25°C to 85°C, and for TL497AC is 0°C to 70°C.
‡All typical values are at T_A = 25°C.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL APPLICATION DATA



BASIC CONFIGURATION
(I_{PK} < 500 mA)



EXTENDED POWER CONFIGURATION
(USING EXTERNAL TRANSISTOR)

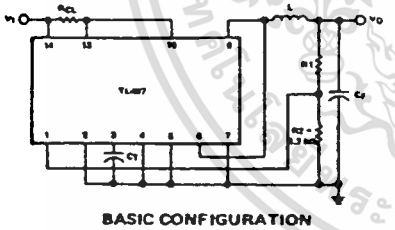
DESIGN EQUATIONS

- $I_{PK} = 2\, I_{LOAD\ max} \left[\frac{V_I - V_O}{V_I} + 1\right]$
- $L\ (\mu H) = \frac{V_I}{I_{PK}}\, t_{on}(\mu s)$

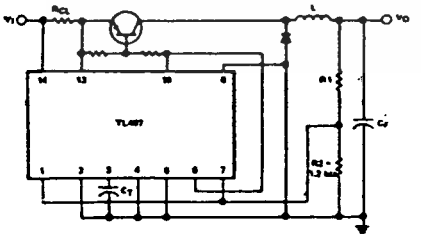
Choose L (50 to 500 μH), calculate t_{on} (25 to 150 μs)

- $C_T(pF) \approx 12\, t_{on}(\mu s)$
- $R1 = (V_O - 1.2)\ k\Omega$
- $R_{CL} = \frac{0.5\ V}{I_{PK}}$
- $C_F(\mu F) \approx \frac{t_{on}\left[\frac{V_I}{V_O} I_{PK} + I_{LOAD}\right]}{V_{RIPPLE\ (PK)}}$

FIGURE 1—POSITIVE REGULATOR, STEP-UP CONFIGURATIONS



BASIC CONFIGURATION
I_{PK} < 500 mA



EXTENDED POWER CONFIGURATION
(USING EXTERNAL TRANSISTOR)

DESIGN EQUATIONS

- $I_{PK} = 2\, I_{LOAD\ max}$
- $L\ (\mu H) = \frac{V_I - V_O}{I_{PK}}\, t_{on}(\mu s)$

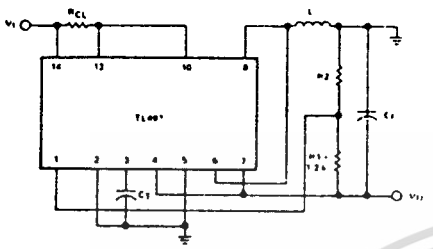
Choose L (50 to 500 μH), calculate t_{on} (10 to 150 μs)

- $C_T(pF) \approx 12\, t_{on}(\mu s)$
- $R1 = (V_O - 1.2)\ k\Omega$
- $R_{CL} = \frac{0.5\ V}{I_{PK}}$
- $C_F(\mu F) \approx \frac{t_{on}\left[\frac{V_I - V_O}{V_O} I_{PK} + I_{LOAD}\right]}{V_{RIPPLE(PK)}}$

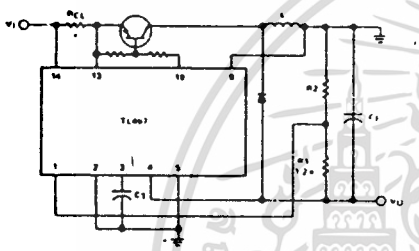
FIGURE 2—POSITIVE REGULATOR, STEP-DOWN CONFIGURATIONS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TYPICAL APPLICATION DATA



BASIC CONFIGURATION
(I_{PK} < 500 mA)



EXTENDED POWER CONFIGURATION
(USING EXTERNAL TRANSISTOR)

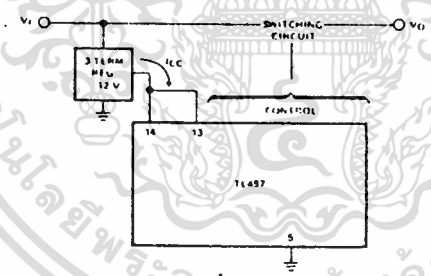
DESIGN EQUATIONS

- $I_{PK} \geq 2\, I_{LOAD(max)} \left[1 + \frac{|V_O|}{V_I}\right]$
- $L\ (\mu H) \geq \frac{V_I}{I_{PK}}\, t_{on}(\mu s)$

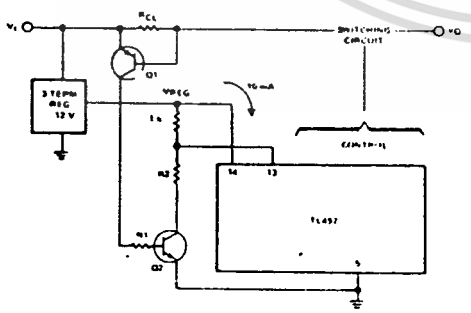
Choose L (50 to 500 μH), calculate t_{on} (25 to 150 μs)

- $C_T\ (pF) \approx 12\, t_{on}(\mu s)$
- $R2 = (V_O - 1.2)\ k\Omega$
- $R_{CL} \geq \frac{0.5\ V}{I_{PK}}$
- $C_F\ (\mu F) \geq t_{on} \frac{\left[\frac{V_I}{|V_O|}\, I_{PK} + I_{LOAD}\right]}{V_{RIPPLE(PK)}}$

FIGURE 3–INVERTING APPLICATIONS



EXTENDED INPUT CONFIGURATION WITHOUT CURRENT LIMIT



CURRENT LIMIT FOR EXTENDED INPUT CONFIGURATION

DESIGN EQUATIONS

$$R_{CL} = \frac{V_{BE(Q1)}}{I_{LIMIT(PK)}}$$

$$R1 = \frac{V_I}{I_{B(Q2)}}$$

$R2 = (V_{REG} - 1)\ 10\ k\Omega$

FIGURE 4–EXTENDED INPUT VOLTAGE RANGE (V_I > 15 V)

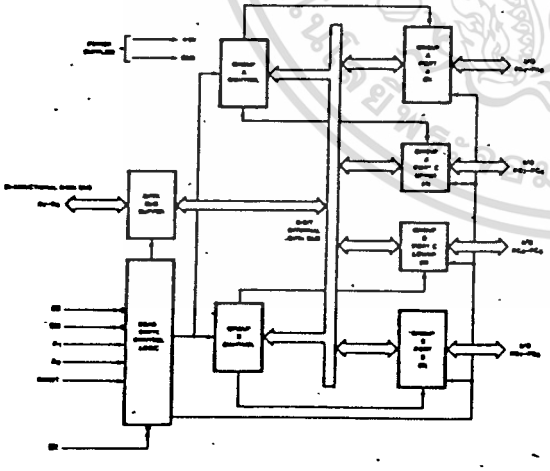
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

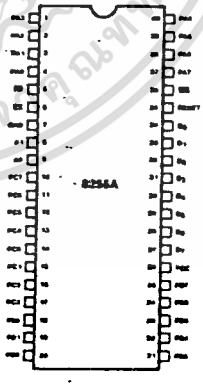
intel

8255A/8255A-5
PROGRAMMABLE PERIPHERAL INTERFACE

- MCS-85™ Compatible 8255A-5
- 24 Programmable I/O Pins
- Completely TTL Compatible
- Fully Compatible with Intel® Microprocessor Families
- Improved Timing Characteristics
- Direct Bit Set/Reset Capability Easing Control Application Interface
- Reduces System Package Count
- Improved DC Driving Capability
- Available in EXPRESS
  - —Standard Temperature Range
  - —Extended Temperature Range

The Intel® 8255A is a general purpose programmable I/O device designed for use with Intel® microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8255A FUNCTIONAL DESCRIPTION

General

The 8255A is a programmable peripheral interface (PPI) device designed for use in Intel® microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 8255A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 8255A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

($\overline{CS}$)

Chip Select. A "low" on this input pin enables the communiction between the 8255A and the CPU.

($\overline{RD}$)

Read. A "low" on this input pin enables the 8255A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 8255A.

($\overline{WR}$)

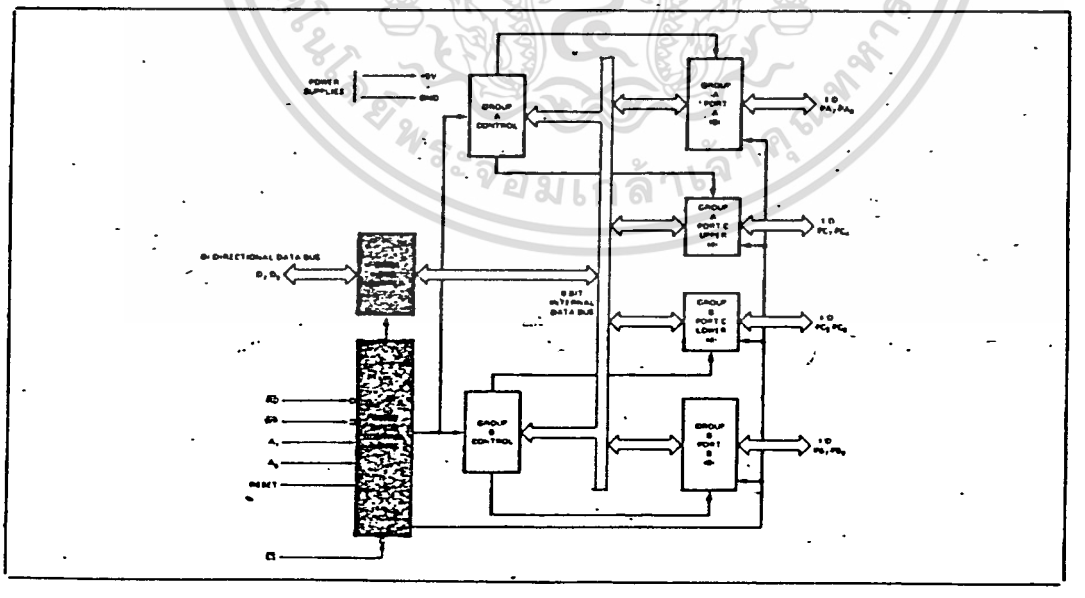
Write. A "low" on this input pin enables the CPU to write data or control words into the 8255A.

(A_0 and A_1)

Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A_0 and A_1).

8255A BASIC OPERATION

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	INPUT OPERATION (READ)
0	0	0	1	0	PORT A – DATA BUS
0	1	0	1	0	PORT B – DATA BUS
1	0	0	1	0	PORT C – DATA BUS
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	DATA BUS – PORT A
0	1	1	0	0	DATA BUS – PORT B
1	0	1	0	0	DATA BUS – PORT C
1	1	1	0	0	DATA BUS – CONTROL
					DISABLE FUNCTION
X	X	X	X	1	DATA BUS – 3-STATE
1	1	0	1	0	ILLEGAL CONDITION
X	X	1	1	0	DATA BUS – 3-STATE



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(RESET)

Reset. A "high" on this input clears the control register and all ports (A, B, C) are set to the input mode.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 8255A.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A – Port A and Port C upper (C7-C4)
Control Group B – Port B and Port C lower (C3-C0)

The Control Word Register can Only be written into. No Read operation of the Control Word Register is allowed.

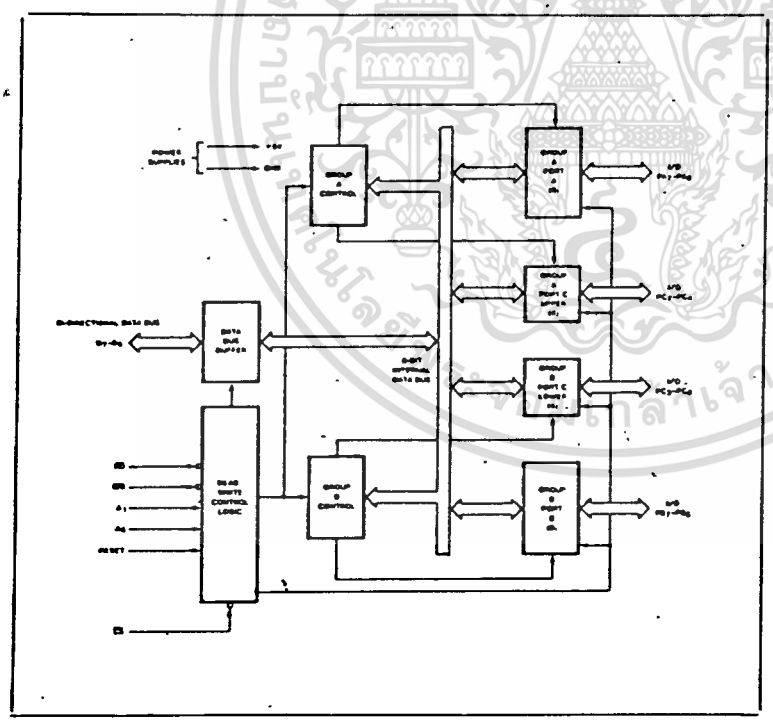
Ports A, B, and C

The 8255A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255A.

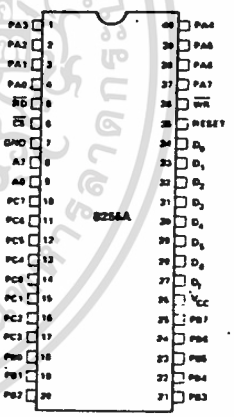
Port A. One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B. One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.



PIN CONFIGURATION



PIN NAMES

D_7–D_0	DATA BUS (BI-DIRECTIONAL)
RESET	RESET INPUT
$\overline{CS}$	CHIP SELECT
$\overline{RD}$	READ INPUT
$\overline{WR}$	WRITE INPUT
A0, A1	PORT ADDRESS
PA7-PA0	PORT A (BIT)
PB7-PB0	PORT B (BIT)
PC7-PC0	PORT C (BIT)
V_{CC}	+5 VOLTS
GND	0 VOLTS

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8255A OPERATIONAL DESCRIPTION

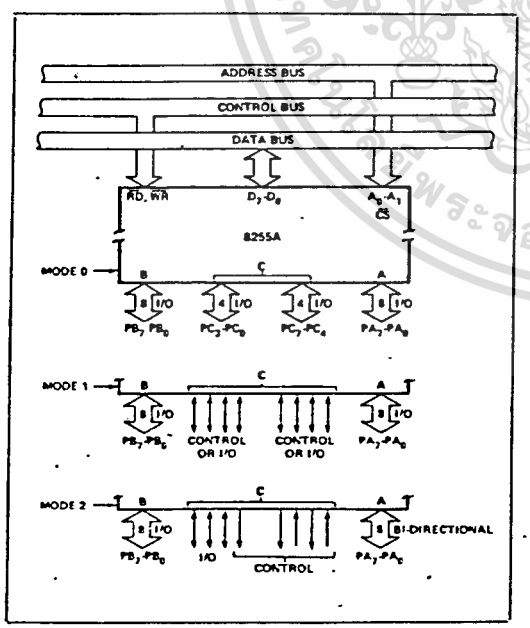
Mode Selection

There are three basic modes of operation that can be selected by the system software:

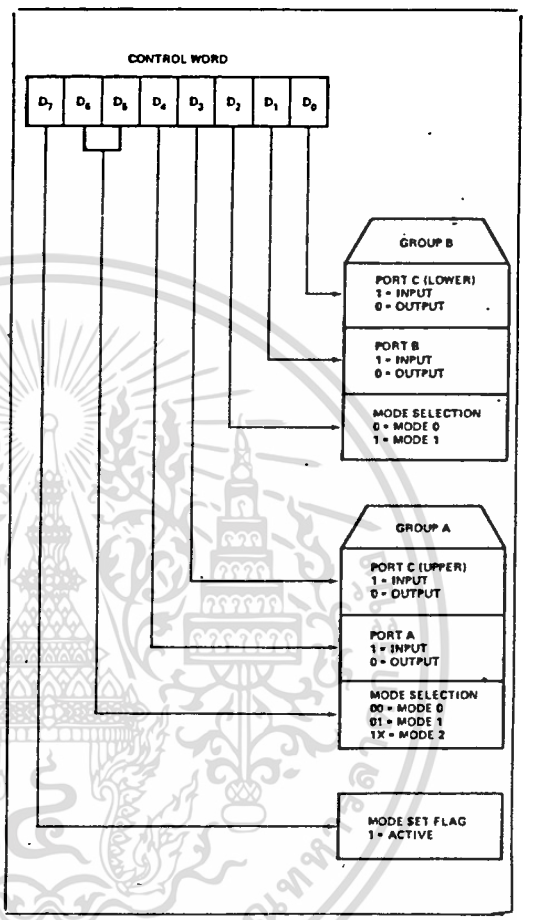
Mode 0 – Basic Input/Output
Mode 1 – Strobed Input/Output
Mode 2 – Bi-Directional Bus

When the reset input goes "high" all ports will be set to the input mode (i.e., all 24 lines will be in the high impedance state). After the reset is removed the 8255A can remain in the input mode with no additional initialization required. During the execution of the system program any of the other modes may be selected using a single output instruction. This allows a single 8255A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.



MODE 0 (Basic Output)



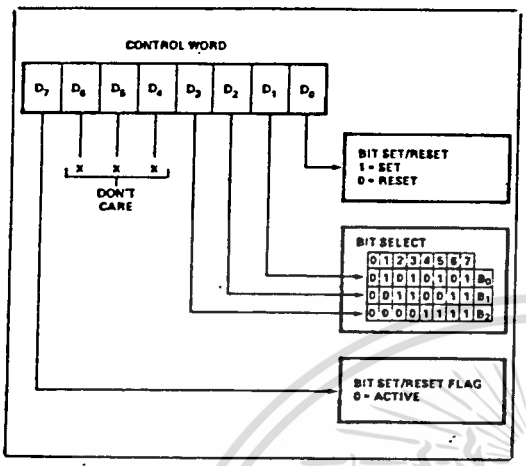
Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 8255A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Bit Set/Reset Format

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the 8255A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET) — INTE is SET — Interrupt enable
(BIT-RESET) — INTE is RESET — Interrupt disable

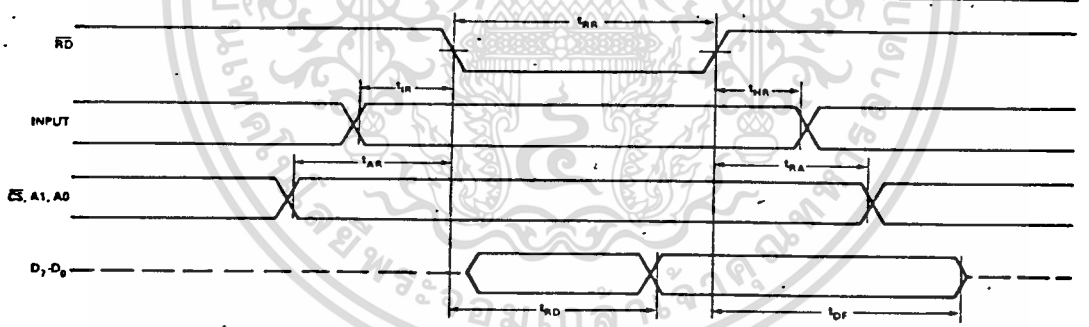
Note: All Mask flip-flops are automatically reset during mode selection and device Reset.

Operating Modes

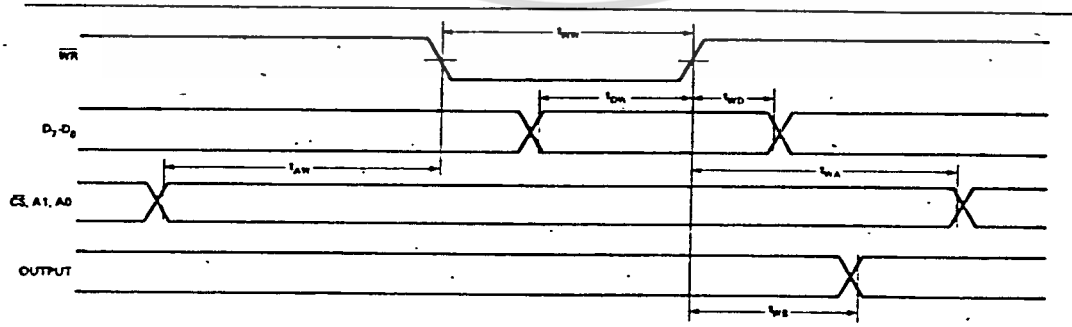
MODE 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.



MODE 0 (Basic Input)

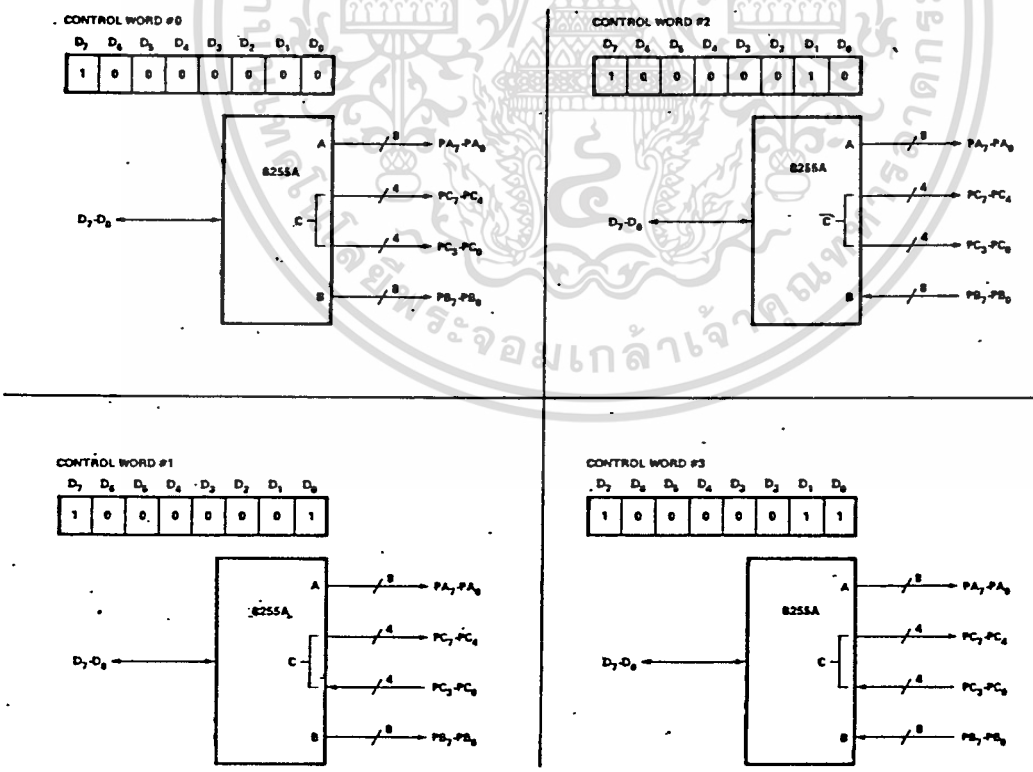


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODE 0 Port Definition

A		B		GROUP A			GROUP B	
D_4	D_3	D_1	D_0	PORT A	PORT C (UPPER)	#	PORT B	PORT C (LOWER)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

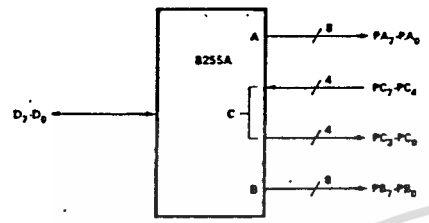
MODE 0 Configurations



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

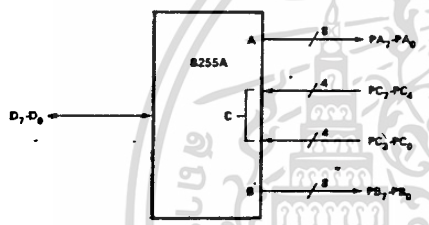
CONTROL WORD #4

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	0



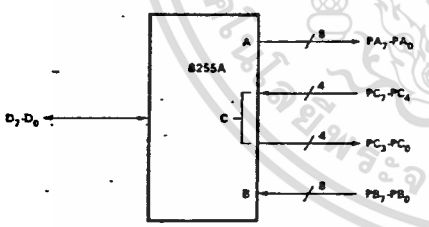
CONTROL WORD #5

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	1



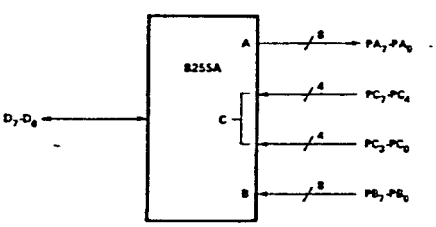
CONTROL WORD #6

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	0



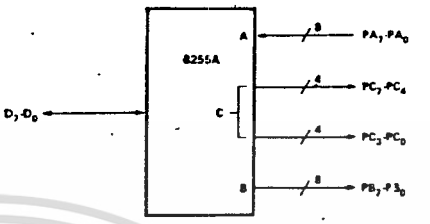
CONTROL WORD #7

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	1



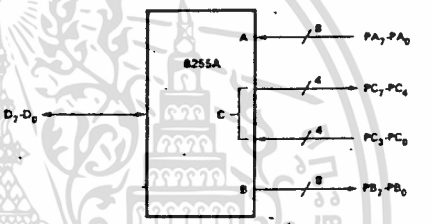
CONTROL WORD #8

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	0



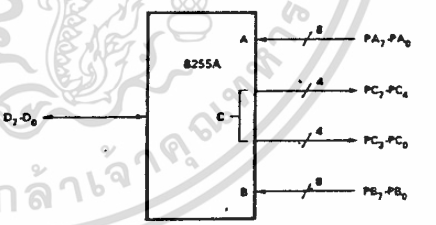
CONTROL WORD #9

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	1



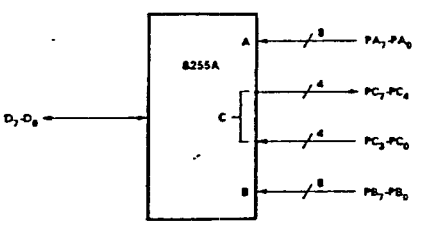
CONTROL WORD #10

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	0

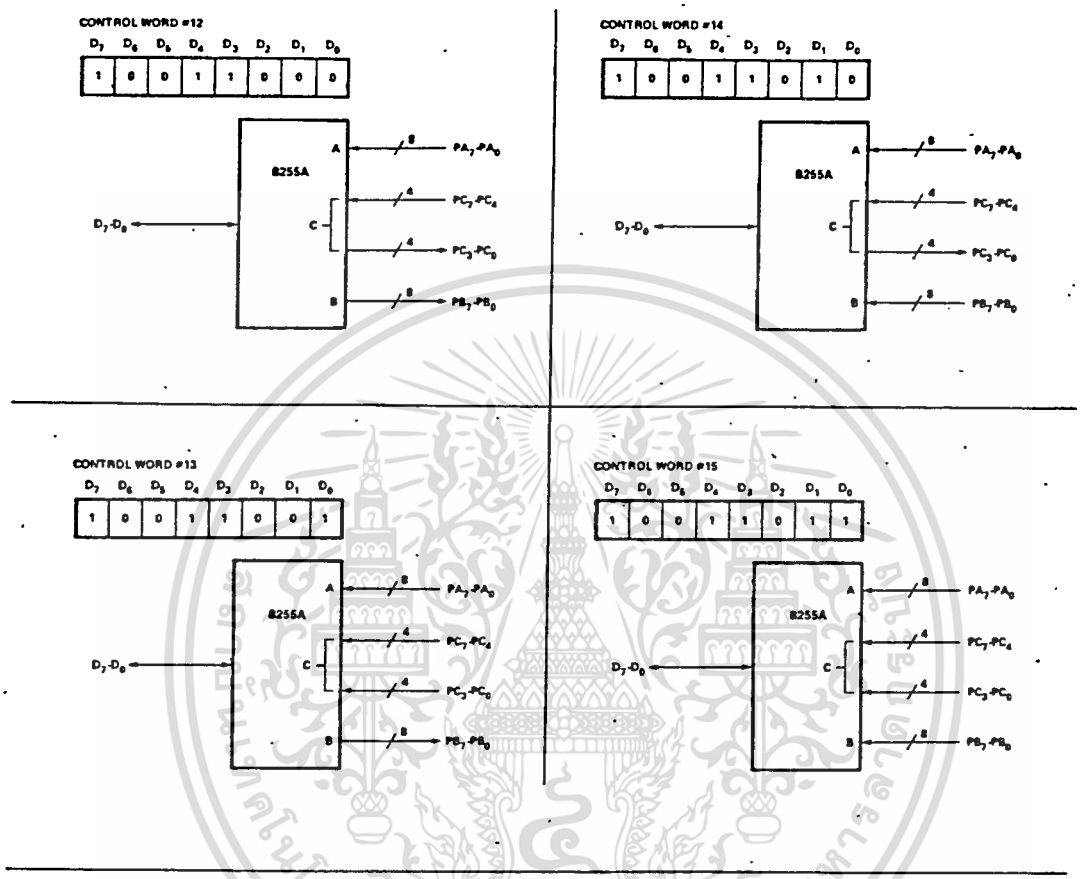


CONTROL WORD #11

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Operating Modes

MODE 1 (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In mode 1, port A and Port B use the lines on port C to generate or accept these "handshaking" signals.

Mode 1 Basic Functional Definitions:

- Two Groups (Group A and Group B)
- Each group contains one 8-bit data port and one 4-bit control/data port.
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Input Control Signal Definition

$\overline{STB}$ (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by STB input being low and is reset by the rising edge of the RD input.

INTR (Interrupt Request)

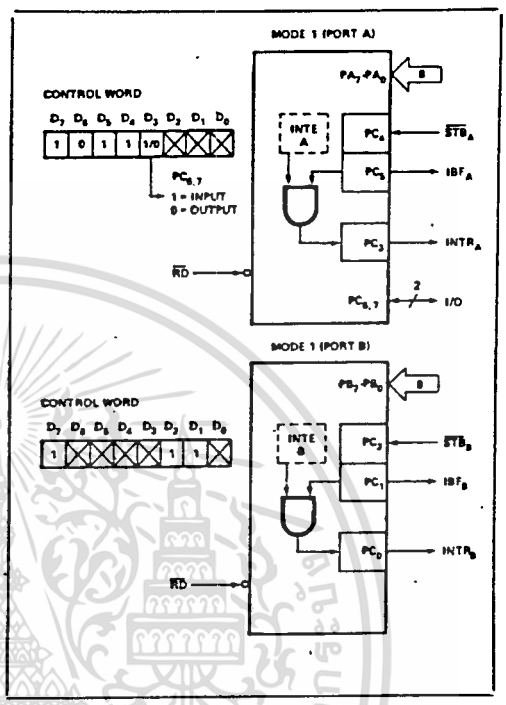
A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the $\overline{STB}$ is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{RD}$. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A

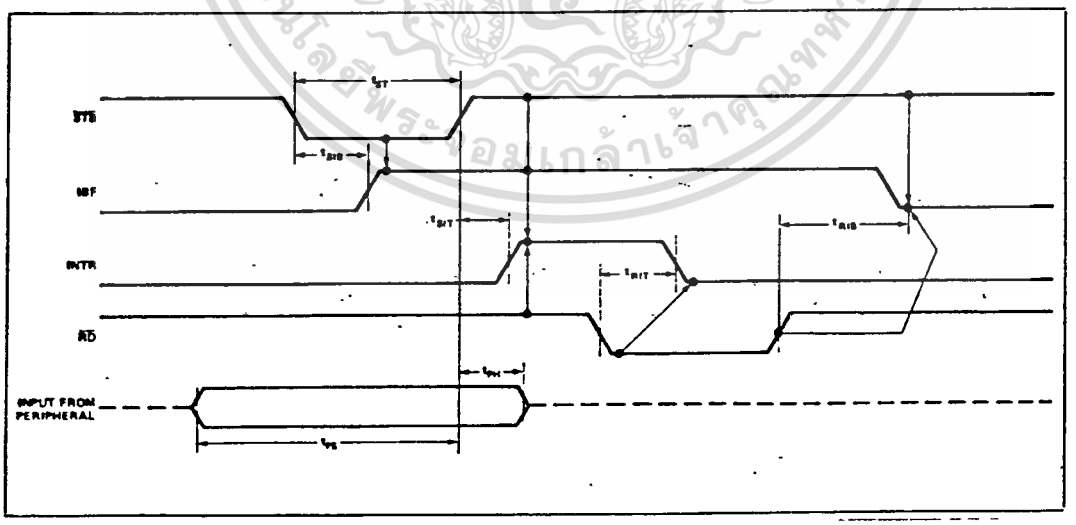
Controlled by bit set/reset of PC_4.

INTE B

Controlled by bit set/reset of PC_2.



MODE 1 Input



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Output Control Signal Definition

$\overline{OBF}$ (Output Buffer Full F/F). The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to the specified port. The $\overline{OBF}$ F/F will be set by the rising edge of the $\overline{WR}$ input and reset by $\overline{ACK}$ input being low.

$\overline{ACK}$ (Acknowledge Input). A "low" on this input informs the 8255A that the data from port A or port B has been accepted. In essence, a response from the peripheral device indicating that it has recieved the data output by the CPU.

INTR (Interrupt Request). A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when $\overline{ACK}$ is a "one", $\overline{OBF}$ is a "one", and INTE is a "one". It is reset by the falling edge of $\overline{WR}$.

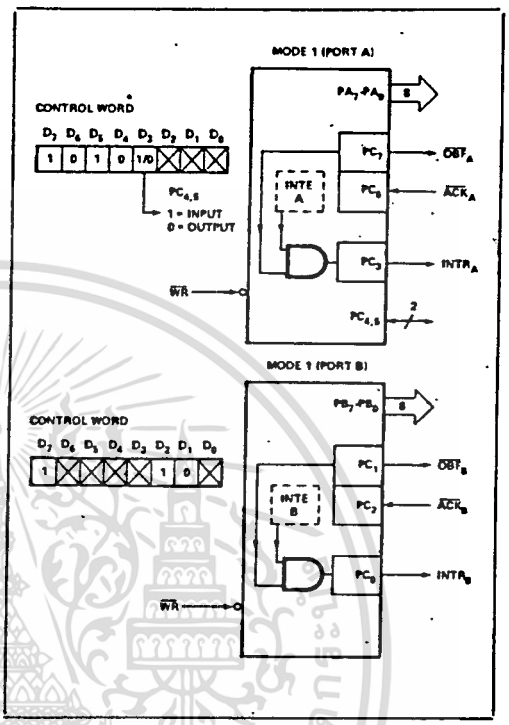
INTR (Interrupt Request). A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when $\overline{ACK}$ is a "one", $\overline{OBF}$ is a "one", and INTE is a "one". It is reset by the falling edge of WR.

INTE A

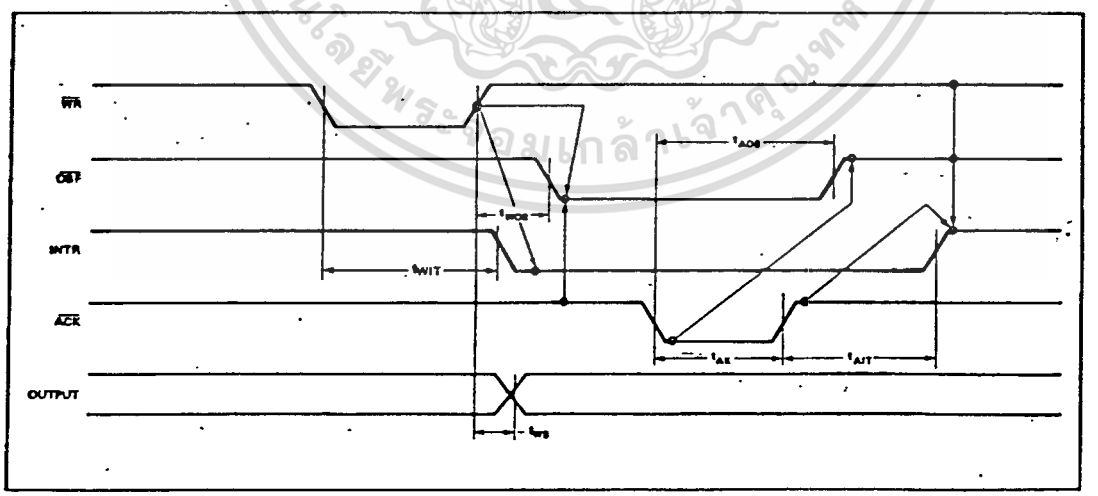
Controlled by bit set/reset of PC_6.

INTE B

Controlled by bit set/reset of PC_2.



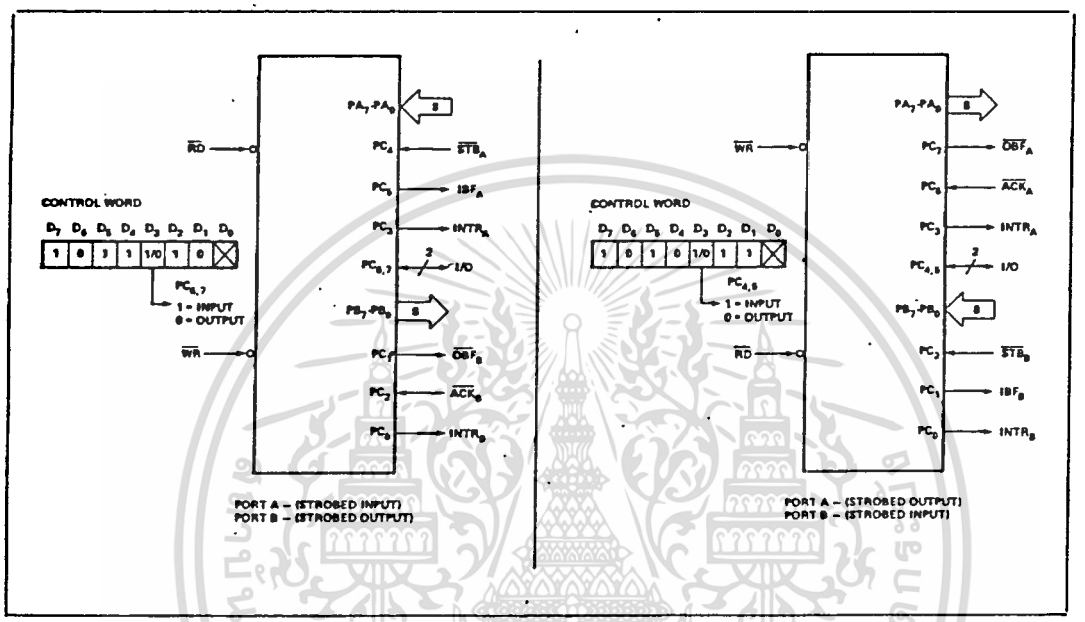
MODE 1 Output



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Combinations of MODE 1

Port A and Port B can be individually defined as input or output in Mode 1 to support a wide variety of strobed I/O applications.



Combinations of MODE 1

Operating Modes

MODE 2 (Strobed Bidirectional Bus I/O). This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to MODE 1. Interrupt generation and enable/disable functions are also available.

MODE 2 Basic Functional Definitions:

- Used in Group A only.
- One 8-bit, bi-directional bus Port (Port A) and a 5-bit control Port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Bidirectional Bus I/O Control Signal Definition

INTR (Interrupt Request). A high on this output can be used to interrupt the CPU for both input or output operations.

Output Operations

$\overline{OBF}$ (Output Buffer Full). The $\overline{OBF}$ output will go "low" to indicate that the CPU has written data out to port A.

$\overline{ACK}$ (Acknowledge). A "low" on this input enables the tri-state output buffer of port A to send out the data. Otherwise, the output buffer will be in the high impedance state.

INTE 1 (The INTE Flip-Flop Associated with OBF). Controlled by bit set/reset of PC_6.

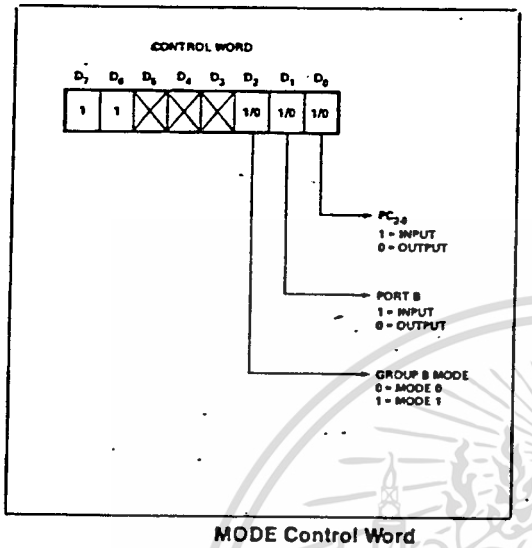
Input Operations

$\overline{STB}$ (Strobe Input). A "low" on this input loads data into the input latch.

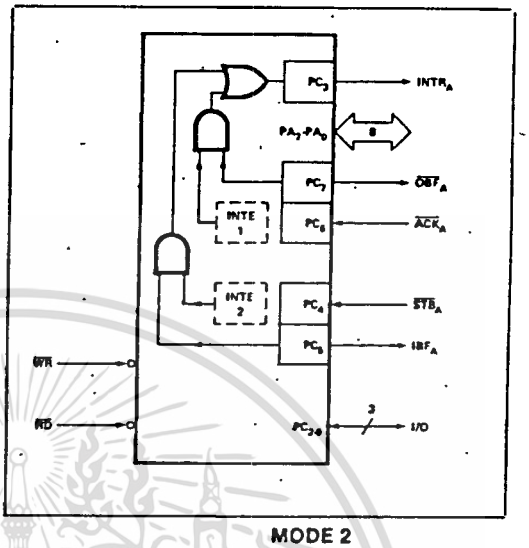
IBF (Input Buffer Full F/F). A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE Flip-Flop Associated with IBF). Controlled by bit set/reset of PC_4.

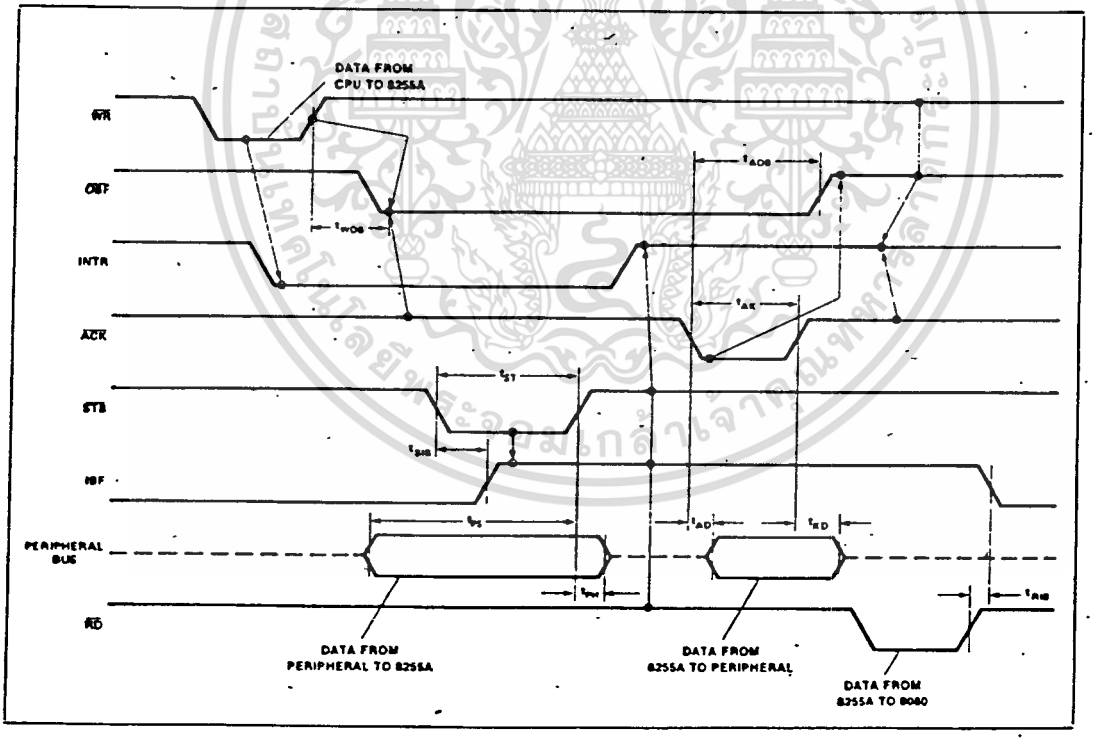
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



MODE Control Word

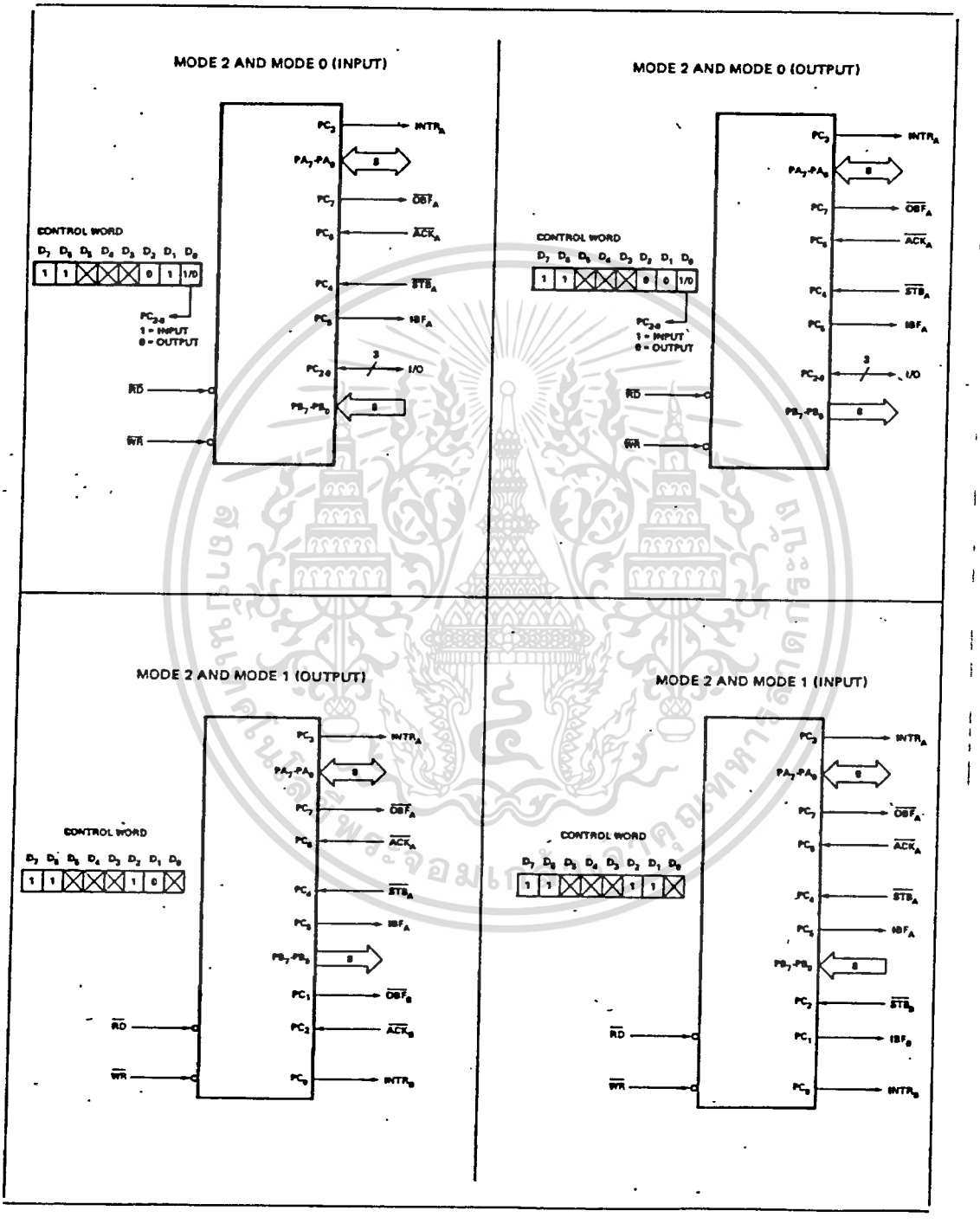


MODE 2



NOTE: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
(INTR = IBF • $\overline{MASK}$ • $\overline{STB}$ • $\overline{RD}$ + $\overline{OBF}$ • $\overline{MASK}$ • $\overline{ACK}$ • $\overline{WR}$)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Mode Definition Summary

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	
PA_0	IN	OUT	IN	OUT	↔	
PA_1	IN	OUT	IN	OUT	↔	
PA_2	IN	OUT	IN	OUT	↔	
PA_3	IN	OUT	IN	OUT	↔	
PA_4	IN	OUT	IN	OUT	↔	
PA_5	IN	OUT	IN	OUT	↔	
PA_6	IN	OUT	IN	OUT	↔	
PA_7	IN	OUT	IN	OUT	↔	
PB_0	IN	OUT	IN	OUT	—	MODE 0 OR MODE 1 ONLY
PB_1	IN	OUT	IN	OUT	—	
PB_2	IN	OUT	IN	OUT	—	
PB_3	IN	OUT	IN	OUT	—	
PB_4	IN	OUT	IN	OUT	—	
PB_5	IN	OUT	IN	OUT	—	
PB_6	IN	OUT	IN	OUT	—	
PB_7	IN	OUT	IN	OUT	—	
PC_0	IN	OUT	$INTR_B$	$INTR_B$	I/O	
PC_1	IN	OUT	IBF_B	$\overline{OBF}_B$	I/O	
PC_2	IN	OUT	$\overline{STB}_B$	$\overline{ACK}_B$	I/O	
PC_3	IN	OUT	$INTR_A$	$INTR_A$	$INTR_A$	
PC_4	IN	OUT	$\overline{STB}_A$	I/O	$\overline{STB}_A$	
PC_5	IN	OUT	IBF_A	I/O	IBF_A	
PC_6	IN	OUT	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$	
PC_7	IN	OUT	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$	

Special Mode Combination Considerations

There are several combinations of modes when not all of the bits in Port C are used for control or status. The remaining bits can be used as follows:

If Programmed as Inputs –
All input lines can be accessed during a normal Port C read.

If Programmed as Outputs –
Bits in C upper (PC_7-PC_4) must be individually accessed using the bit set/reset function.

Bits in C lower (PC_3-PC_0) can be accessed using the bit set/reset function or accessed as a threesome by writing into Port C.

Source Current Capability on Port B and Port C

Any set of eight output buffers, selected randomly from Ports B and C can source 1mA at 1.5 volts. This feature allows the 8255 to directly drive Darlington type drivers and high-voltage displays that require such source current.

Reading Port C Status

In Mode 0, Port C transfers data to or from the peripheral device. When the 8255 is programmed to function in Modes 1 or 2, Port C generates or accepts "hand-shaking" signals with the peripheral device. Reading the contents of Port C allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.

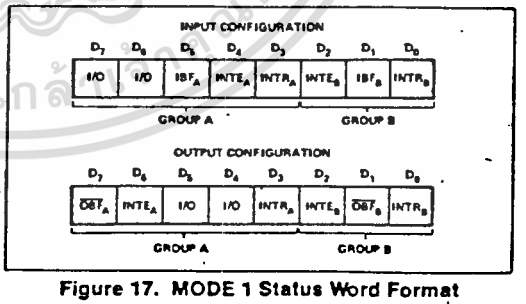
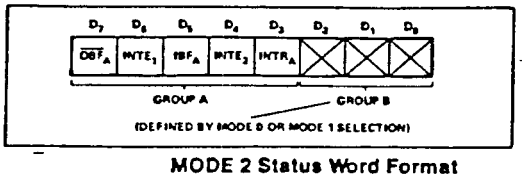


Figure 17. MODE 1 Status Word Format



MODE 2 Status Word Format

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

APPLICATIONS OF THE 8255A

The 8255A is a very powerful tool for interfacing peripheral equipment to the microcomputer system. It represents the optimum use of available pins and is flexible enough to interface almost any I/O device without the need for additional external logic.

Each peripheral device in a microcomputer system usually has a "service routine" associated with it. The routine manages the software interface between the device and the CPU. The functional definition of the 8255A is programmed by the I/O service routine and becomes an extension of the system software. By examining the I/O devices interface characteristics for both data transfer and timing, and matching this information to the examples and tables in the detailed operational description, a control word can easily be developed to initialize the 8255A to exactly "fit" the application. Figures 19 through 25 present a few examples of typical applications of the 8255A.

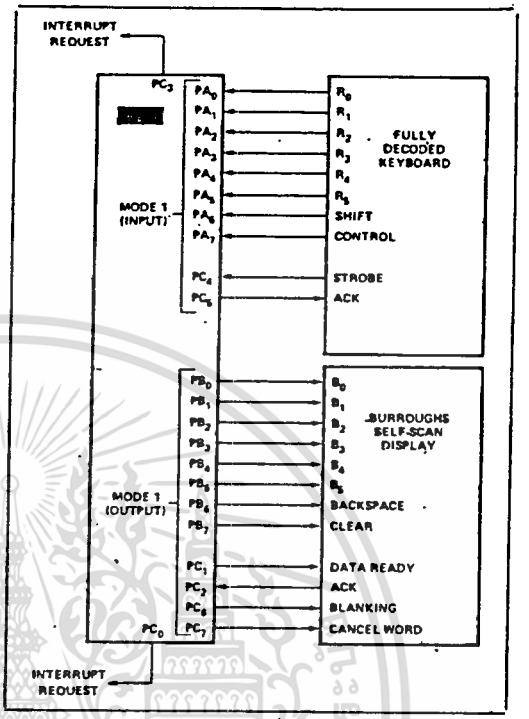
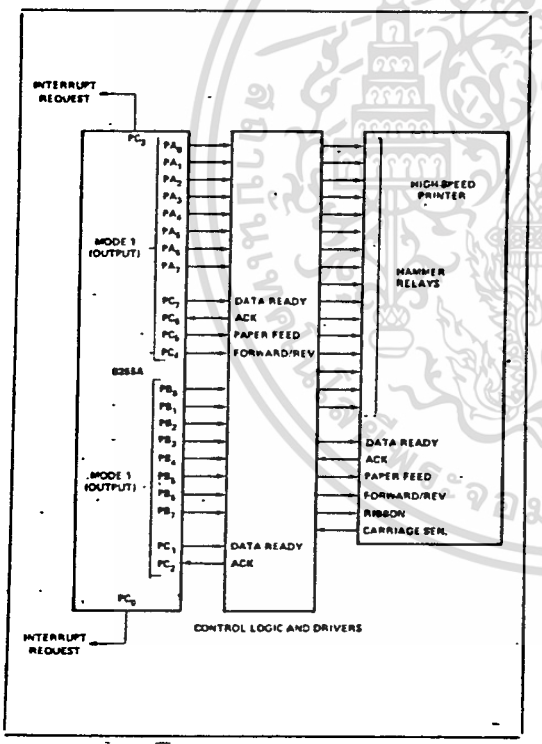
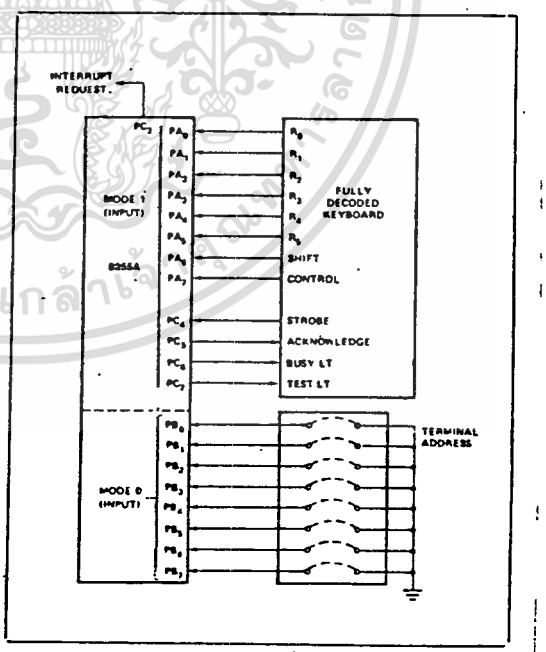


Figure 20. Keyboard and Display Interface



Printer Interface



Keyboard and Terminal Address Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

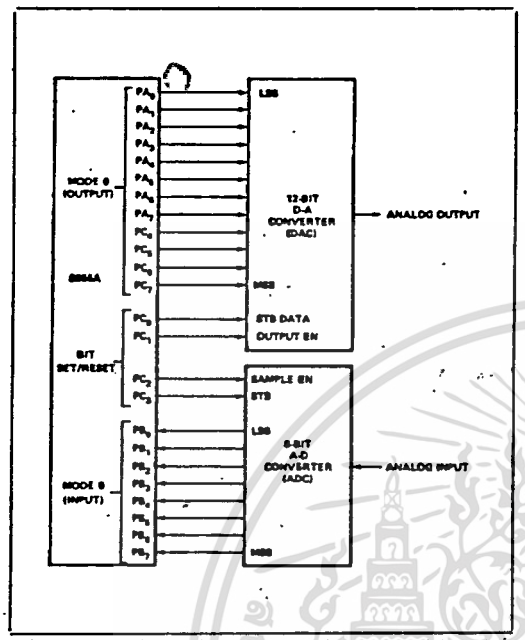


Figure 22. Digital to Analog, Analog to Digital

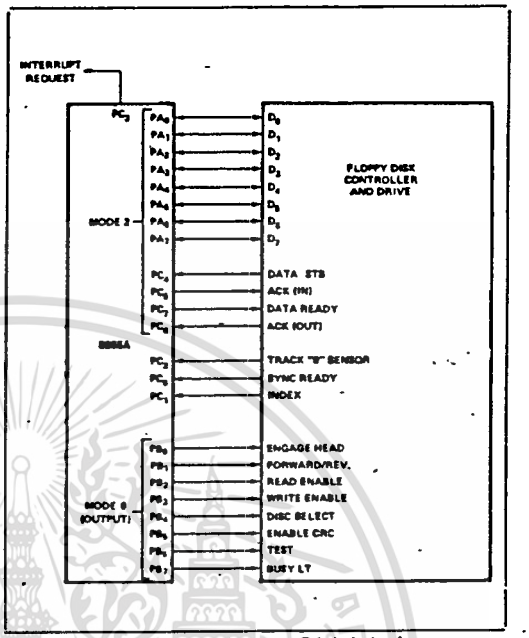
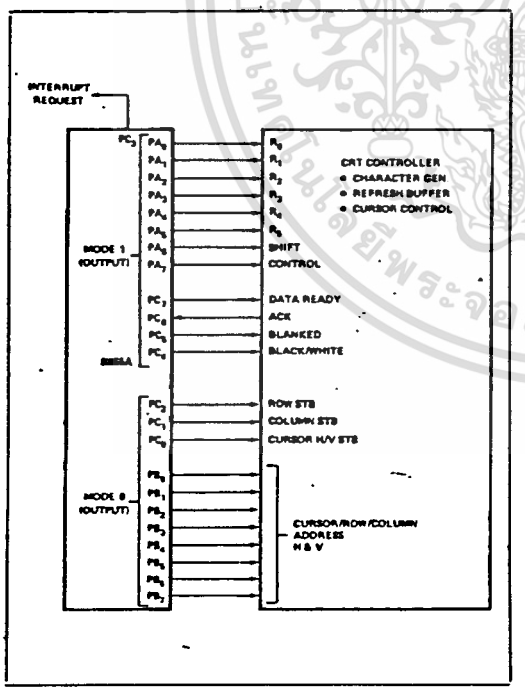
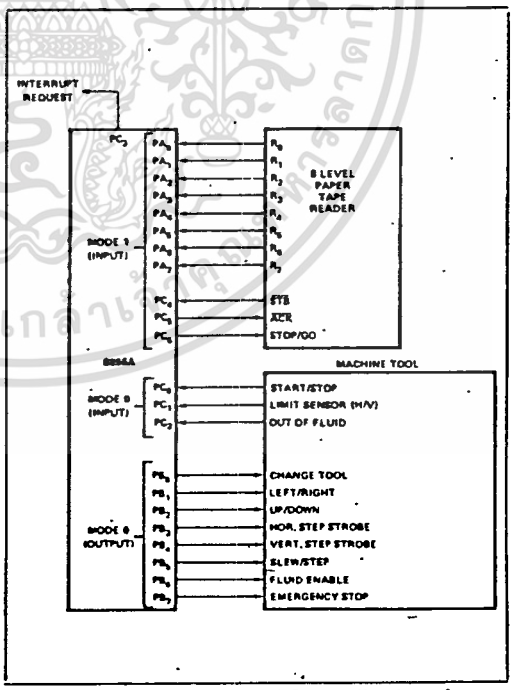


Figure 23. Basic Floppy Disk Interface



Basic CRT Controller Interface



Machine Tool Controller Interface

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias. . . . . 0°C to 70°C
Storage Temperature . . . . . −65°C to +150°C
Voltage on Any Pin
With Respect to Ground . . . . . −0.5V to +7V
Power Dissipation . . . . . 1 Watt

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

D.C. CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = +5V ±10%, GND = 0V)*

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	V	
V_{OL} (DB)	Output Low Voltage (Data Bus)		0.45*	V	I_{OL} = 2.5mA
V_{OL}(PER)	Output Low Voltage (Peripheral Port)		0.45*	V	I_{OL} = 1.7mA
V_{OH}(DB)	Output High Voltage (Data Bus)	2.4		V	I_{OH} = -400μA
V_{OH}(PER)	Output High Voltage (Peripheral Port)	2.4		V	I_{OH} = -200μA
I_{DAR}[1]	Darlington Drive Current	-1.0	-4.0	mA	R_{EXT} = 750Ω; V_{EXT} = 1.5V
I_{CC}	Power Supply Current		120	mA	
I_{IL}	Input Load Current		±10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} to .45V

NOTE:
1. Available on any 8 pins from Port B and C.

CAPACITANCE (T_A = 25°C, V_{CC} = GND = 0V)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance			10	pF	fc = 1MHz
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to GND

A.C. CHARACTERISTICS (T_A = 0°C to 70°C, V_{CC} = +5V ±10%, GND = 0V)*

Bus Parameters

READ

Symbol	Parameter	8255A		8255A-5		Unit
		Min.	Max.	Min.	Max.	
t_{AR}	Address Stable Before READ	0		0		ns
t_{RA}	Address Stable After READ	0		0		ns
t_{RR}	READ Pulse Width	300		300		ns
t_{RD}	Data Valid From READ[1]		250		200	ns
t_{DF}	Data Float After READ	10	150	10	100	ns
t_{RV}	Time Between READs and/or WRITEs	850		850		ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A.C. CHARACTERISTICS (Continued)

WRITE

Symbol	Parameter	8255A		8255A-5		Unit
		Min.	Max.	Min.	Max.	
t_{AW}	Address Stable Before WRITE	0		0		ns
t_{WA}	Address Stable After WRITE	20		20		ns
t_{WW}	WRITE Pulse Width	400		300		ns
t_{DW}	Data Valid to WRITE (T.E.)	100		100		ns
t_{WD}	Data Valid After WRITE	30		30		ns

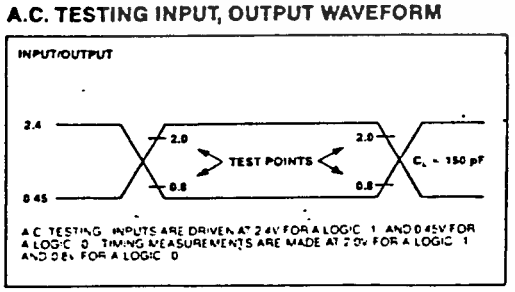
OTHER TIMINGS

Symbol	Parameter	8255A		8255A-5		Unit
		Min.	Max.	Min.	Max.	
t_{WB}	WR = 1 to Output[1]		350		350	ns
t_{IR}	Peripheral Data Before RD	0		0		ns
t_{HR}	Peripheral Data After RD	0		0		ns
t_{AK}	ACK Pulse Width	300		300		ns
t_{ST}	STB Pulse Width	500		500		ns
t_{PS}	Per. Data Before T.E. of STB	0		0		ns
t_{PH}	Per. Data After T.E. of STB	180		180		ns
t_{AD}	ACK = 0 to Output[1]		300		300	ns
t_{KD}	ACK = 1 to Output Float	20	250	20	250	ns
t_{WOB}	WR = 1 to OBF = 0[1]		650		650	ns
t_{AOB}	ACK = 0 to OBF = 1[1]		350		350	ns
t_{SIB}	STB = 0 to IBF = 1[1]		300		300	ns
t_{RIB}	RD = 1 to IBF = 0[1]		300		300	ns
t_{RIT}	RD = 0 to INTR = 0[1]		400		400	ns
t_{SIT}	STB = 1 to INTR = 1[1]		300		300	ns
t_{AIT}	ACK = 1 to INTR = 1[1]		350		350	ns
t_{WIT}	WR = 0 to INTR = 0[1,3]		450		450	ns

NOTES:
1. Test Conditions: C_L = 150 pF.
2. Period of Reset pulse must be at least 50μs during or after power on. Subsequent Reset pulse can be 500 ns min.
3. INTR↑ may occur as early as $\overline{WR}$↓.

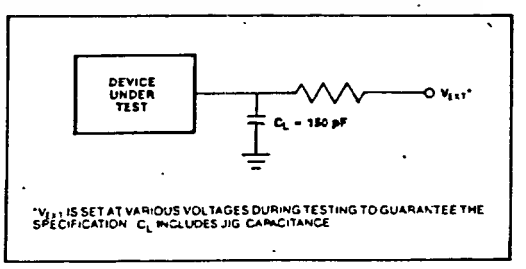
* For Extended Temperature EXPRESS, use M8255A electrical parameters.

A.C. TESTING INPUT, OUTPUT WAVEFORM



A.C. TESTING: INPUTS ARE DRIVEN AT 2.4V FOR A LOGIC "1" AND 0.45V FOR A LOGIC "0". TIMING MEASUREMENTS ARE MADE AT 2.0V FOR A LOGIC "1" AND 0.8V FOR A LOGIC "0".

A.C. TESTING LOAD CIRCUIT

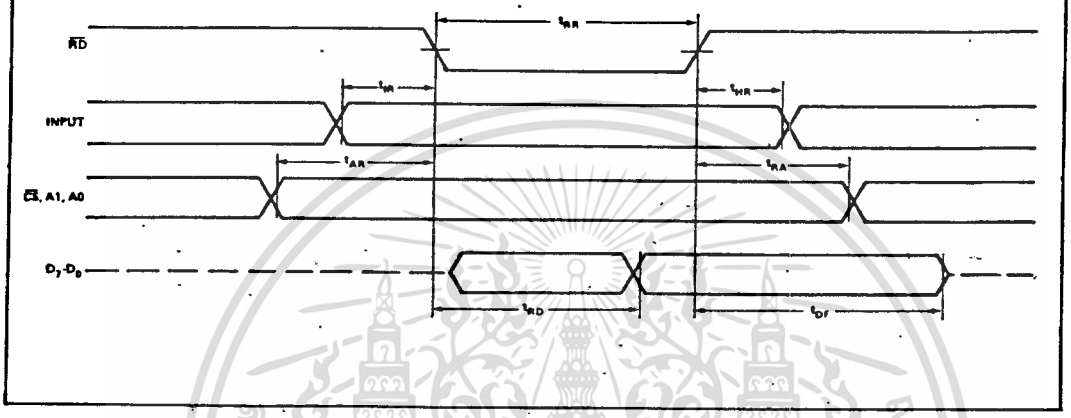


*V_{CC} IS SET AT VARIOUS VOLTAGES DURING TESTING TO GUARANTEE THE SPECIFICATION. C_L INCLUDES JIG CAPACITANCE.

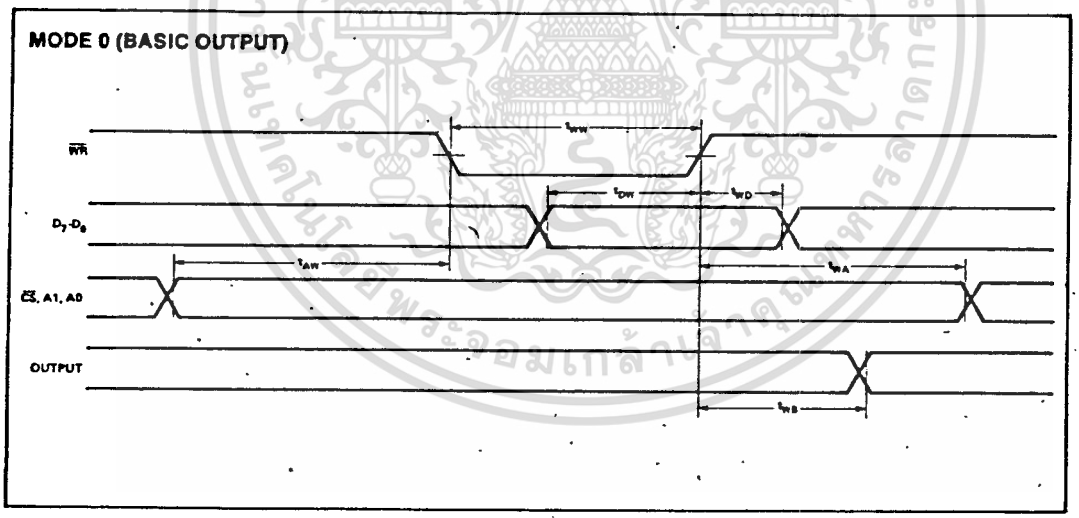
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAVEFORMS

MODE 0 (BASIC INPUT)



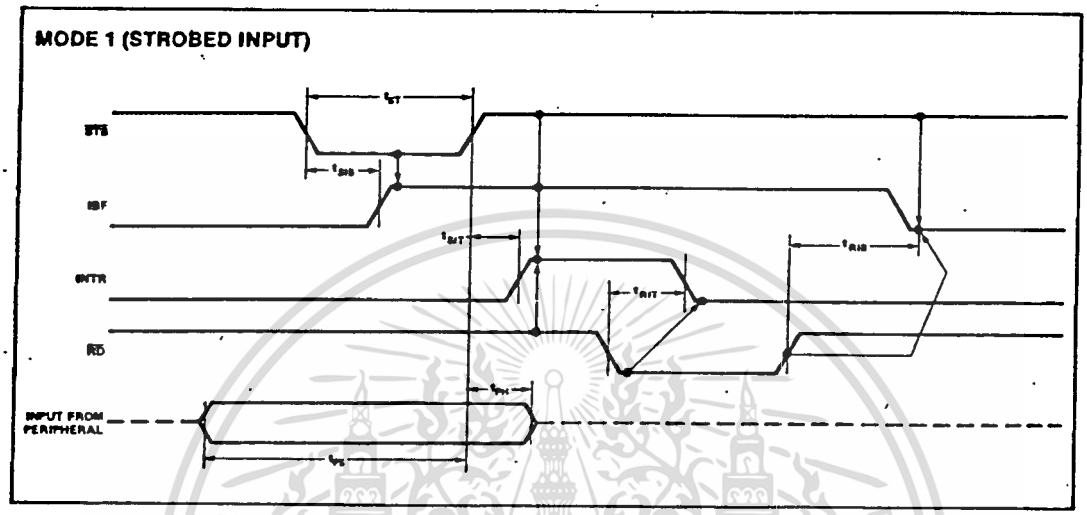
MODE 0 (BASIC OUTPUT)



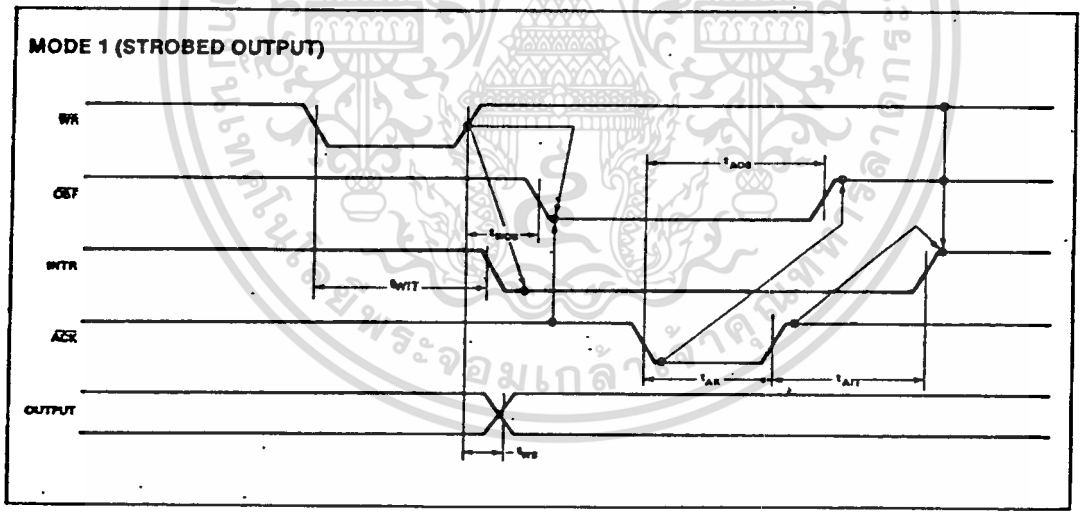
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

WAVEFORMS (Continued)

MODE 1 (STROBED INPUT)



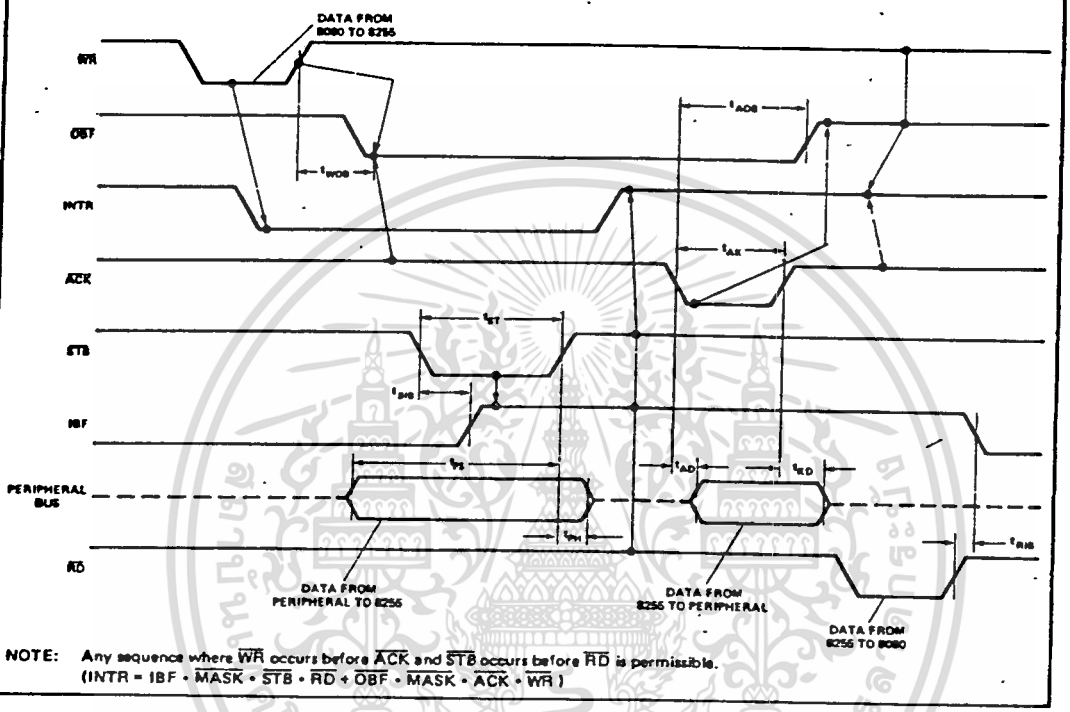
MODE 1 (STROBED OUTPUT)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

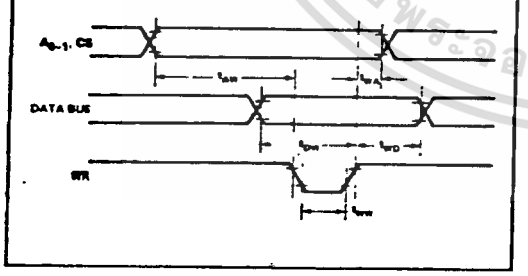
WAVEFORMS (Continued)

MODE 2 (BIDIRECTIONAL)

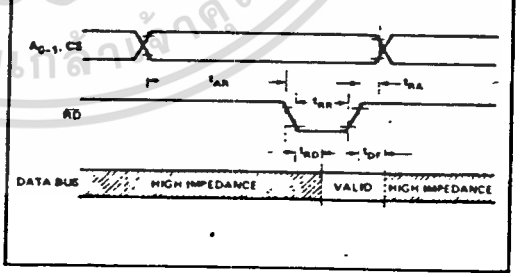


NOTE: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
($INTR = IBF \cdot \overline{MASK} \cdot \overline{STB} \cdot \overline{RD} + \overline{OBF} \cdot \overline{MASK} \cdot \overline{ACK} \cdot \overline{WR}$)

WRITE TIMING



READ TIMING



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

1. มนัส สังวรศิลป์, "สร้างเครื่องเขียน ทีทีแอล พรอม", วารสารอิเล็กทรอนิคส์ เวิลด์, ปีที่ 9, ฉบับที่ 92, 2528, หน้า 72-76.
2. ยืน ภู่วรวรรณ, "ไมโครโปรเซสเซอร์ 8088", ซีเอ็ดยูเคชั่น, 363 หน้า, 2522
3. "Bipolar Memory Data Manual 1986", Signetic, 175 p., 1986.
4. Eggbrecht, "IBM Interfacing", Howard W. Sam & Co., 246 p., 1983.
5. "IBM Technical ReferenceManual", IBM Company, 536 p,
6. John Angermeyer and Kevin Jaeger, "MS-DOS Devloper's Guide", Howard W. Sam & Co., 440 p.,1986.
7. "Microsoft Marco Assembler", Microsoft, 272 p., 1985.
8. Ray Duncan, "Advance MS-DOS" , Microsoft Press, 486 p., 1986.
9. Robert Jourdain, "Programmer's Problem Solver For The IBM PC, XT & AT", Brady Books, 473 p. 1986.
10. Russell Rector and George Alexy, "The 8086 Book", McGraw Hill, 602 p.,1980.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้